

TR-83-008

計算機控制機械手計畫第一年報告

計畫主持人：林 俊 忻

參與人員：

助理研究員：何 建 明

助 理 員：陳燕明、詹德信

研究生助理：李維智、黃英哲、黃嘉麟、曹同慶、

蔡明祺、鍾國亮

計畫編號：NSC 70-0401-E001-01

執行期限：七十一年四月一日至七十二年六月卅日

日 期：中華民國 七十二 年 八 月 一 日

中研院資訊所圖書室



3 0330 03 00041 3

0041

參 考 書  
不 外 借

# 目 錄

	頁數
導言 .....	1
計畫目標 .....	1
工作成果 .....	1
機械手動力分析與運動模擬 .....	3
1 導言 .....	3
2 機械手之座標系統與運動學 .....	3
3 機械手動力學 .....	8
4 動力分析與運動模擬 .....	16
5 結論 .....	28
機械手連續路徑之近似描述 .....	29
1 導論 .....	29
2 X-spline 軌跡 .....	31
3 卡氏座標路徑規畫上之考慮 .....	33
4 結論 .....	34
一個機械手語言的設計 .....	36
1 導言 .....	36
2 RLO 語言 .....	37
3 轉譯器及直譯器 .....	46
4 結論 .....	56
參考書目 .....	59
附錄一 .....	61
附錄二 .....	62
附錄三 .....	65

附錄四 ..... 68

附錄五 ..... 82

三  
二  
一

## 摘 要

本報告敘述「計算機控制機械手」計畫第一年之研究成果，本計畫為期兩年，主要目的在於研究擴展當前有關機械手控制之理論，以期提供對設計高等機械手所需具備之良好根基，本報告首先討論動力學在機械手上之分析與模擬應用，然後討論機械手運動軌跡問題，最後為進一步工作之準備，先試著實作一類似現有AL之語言系統。

## 1 導言

本報告敘述「計算機控制機械手」計畫第一年工作及研究成果。

## 2 計畫目標

本計畫主要目的在於研究並擴展當前有關機械手控制理論，以期提供對設計機械手所需具備之良好根基，計畫研究方向著重下列三個方面：

- (1) 機械手動力學分析與運動模擬。
- (2) 機械手路徑問題。
- (3) 機械手系統與語言。

## 3 工作成果

本計畫第一年工作主要成果包括

### (1) 機械手動力學與運動模擬

我們首先深入探討 Lagrangian 及 Newton-Euler 兩種方式之動力學式子，並使用 FORTRAN 將之寫成程式，基本上這些 FORTRAN 程式適用於一般機械手，只要配合給予機械手之一些特性值及關節關係，這些程式可自動計算運動時關節之施力情形，因此這些程式可為機械手設計上分析及運動模擬之工具。

### (2) 機械手運動軌跡描述

當手接近物體、或為避免碰撞、或為跟隨輸送帶取物、及其他

許多動作，往往其運動非一般「點至點」方式所能勝任，在此狀況之下，我們必須要求機械手之運動經過一系列指定點，以達成工作需求，對此需求，我們採用X-Spline 在關節值上做內插，並考慮在卡氏座標路徑規畫上路徑取點間隔及速度決定之一些原則。

### (3) 機械手系統與語言

我們基於AL 語言系統實作一 RLO 機械手控制語言，這個語言系統可接受語言程式，依指令做必要之路徑規畫，以產生伺服系統所需之關節運動資料。本系統並未真正連結至伺服系統，但試驗顯示轉譯器、直譯器及路徑規畫程式均工作正常。

# 機械手動力分析與運動模擬

## 1 導言

機械手係一高度非線性之結構，因此其系統模式也就非常複雜，使得準確而有效的控制較為困難，過去，對於動力學的探討基本上有 Lagrangian 及 Newton-Euler 兩種運算方式，由於要求的運算太多，至目前直接使用這些方式於控制機械手仍屬不可能或困難。但這兩種運算方式事實上對機械手動力學分析及模擬仍為必需的工具，針對這個需要，我們設計了一套計算機的程式來執行這些運算，這些程式可讀取各種不同機械手之結構資料，計算出機械手在各種運動狀況之施力情形，因此可用為動力分析與運動模擬。

本文首先對機械手運動學及動力學加以探討整理，然後敘述可能的運動模擬，同時列出這些分析與模擬所需要之主要程式及一些應用。

## 2 機械手之座標系統與運動學

機械手的結構是由幾組機械臂 (link) 連接而成的，臂與臂之間的關節 (joint) 連接方式不外乎是迴轉型 (revolute) 或平移角柱型 (prismatic) 兩種。通常，每組關節臂 (joint-link) 形成一個自由度 (degree of freedom 簡稱 d. o. f.)，有  $n$  個自由度的機械手，就須至少有  $n$  個關節臂，為了方便起見，定第 0 個機械臂為附著在基座的 (base) 支持物，在此基座及  $n$  個關節臂上各定義一

參考座標。對於相鄰兩機械臂間的座標關係，如圖 1 所示，可引用 Denavit-Hartenberg 矩陣表示相互間的座標關係 [1]。

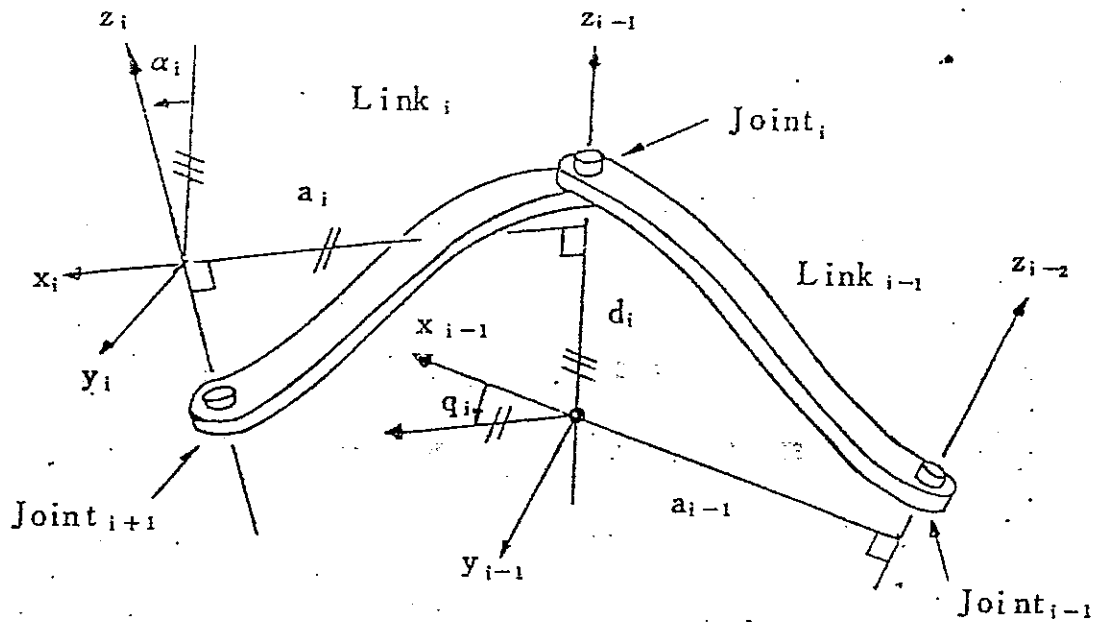


圖 1 機械臂間的座標系參數

現在考慮一含有  $n+1$  組機械臂的機械手，將之編號成 0 到  $n$ ，第 0 個機械臂定為機械手所座落的基座支架 (frame)。兩相鄰機械臂  $i-1$  和  $i$  的交點稱為關節  $i$ ；關節的運動方式分為轉動 (rotational) 及平移 (translational) 兩種。

為描述手臂運動關係，在每一個機械臂  $i$  上設定一單位正交向量 (orthonormal) 座標系  $(x_i, y_i, z_i)$ ，此座標系是附著在機械臂上隨著它運動。對於  $x_i, y_i, z_i$  三個軸的定義 (看圖 1) 如下：

$z_i \triangleq$  若  $i+1$  關節為旋轉式則為  $i+1$  關節的旋轉軸心，如是平移式則為  $i+1$  關節的移動方向。

$x_i \triangleq z_{i-1} \times z_i / \| z_{i-1} \times z_i \|$ ；互與  $z_{i-1}$  及  $z_i$  相垂直的向量，假



如  $z_{i-1} \times z_i = 0$  則任選一  $x_i$  向量使得  $x_i \cdot z_i = 0$ 。

$$y_i \triangleq z_i \times x_i$$

相鄰兩座標系間的轉換矩陣關係可由下列參數 (parameters) 訂定之：

$q_i \triangleq$  沿著  $z_{i-1}$  方向反時鐘衡量， $x_{i-1}$  至  $x_i$  的角度。

$d_i \triangleq$  沿著  $z_{i-1}$  方向，從  $x_{i-1}$  至  $x_i$  的距離。

$\alpha_i \triangleq$  沿著  $x_i$  方向反時鐘衡量， $z_{i-1}$  至  $z_i$  的角度。

$a_i \triangleq$  沿著  $x_i$  方向，從  $z_{i-1}$  至  $z_i$  的距離。

在座標系  $i$  上的一向量  $y_i$  轉換到  $i-1$  座標系，以  $y_{i-1}$  表示，則  $y_i$  及  $y_{i-1}$  有下列關係：

$$y_{i-1} = T_{i-1}^i y_i$$

其中  $T_{i-1}^i$  是轉換矩陣，表示如下：

$$T_{i-1}^i = \begin{pmatrix} \cos q_i - \cos \alpha_i \cdot \sin q_i & \sin \alpha_i \cdot \sin q_i & a_i \cdot \cos q_i & 0 \\ \sin q_i & \cos \alpha_i \cdot \cos q_i - \sin \alpha_i \cdot \cos q_i & a_i \cdot \sin q_i & 0 \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2)$$

因此，對任何座標系  $i$  和  $j$  間的轉換關係都可藉由上述的關係式推導而得。對  $j > i$  而言，可定義成  $T_i^j$  其中  $T_i^j = T_i^{i+1} \cdot T_{i+1}^{i+2} \cdot \dots \cdot T_{j-1}^j$ ； $T_i^j$  為一齊次轉換式 (homogeneous transformation)。 (2) 式中左上端 3X3 子矩陣 (submatrix) 是描述轉動的方位 (orientation) 稱為轉動矩陣，而第四行是表示平移的位置向量。轉動矩陣  $A_{i-1}^i$  可表示成：

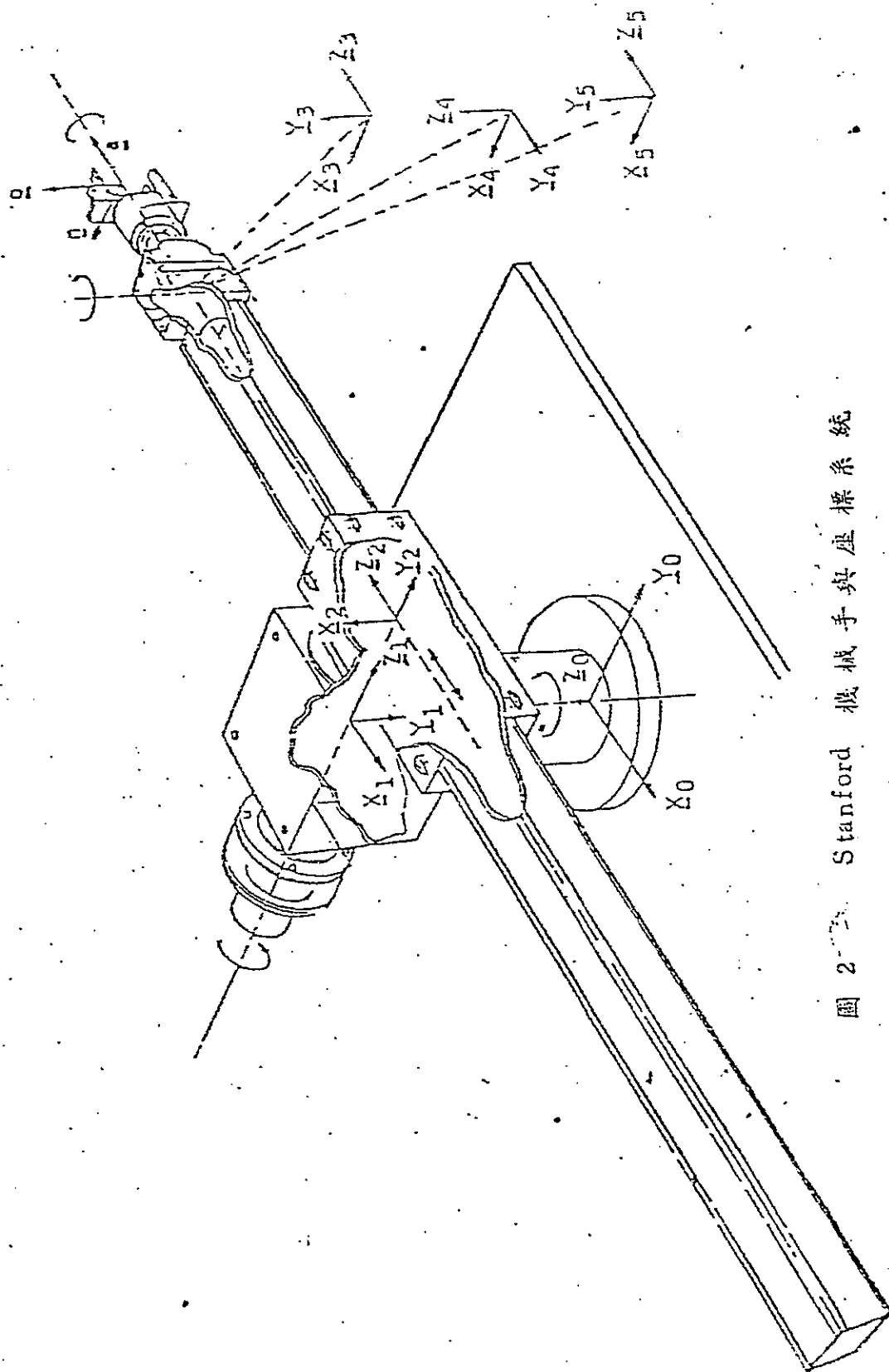


圖 2-3 Stanford 機械手與座標系統

$$A_{i-1}^i = \begin{pmatrix} \cos q_i & -\cos \alpha_i \sin q_i & \sin \alpha_i \sin q_i & \sin q_i \\ \sin q_i & \cos \alpha_i \cos q_i & -\sin \alpha_i \cos q_i & \cos q_i \\ 0 & \sin \alpha_i & \cos \alpha_i & 0 \end{pmatrix} \quad (3)$$

對  $j > i$  而言,  $A_i^j = A_i^{i+1} \cdot A_{i+1}^{i+2} \cdots A_{j-1}^j$ ; 轉動矩陣  $A_i^j$  中的各分量是相互正交的。反矩陣 (inverse matrix) 以  $A_j^i$  表示其等式關如下:

$$A_j^i = (A_i^j)^{-1} = (A_i^j)^T \quad (4)$$

其中  $( )^T$  是表示矩陣轉置 (transpose) 之意思。

圖 2 是 Stanford 機械手的結構外型與座標系, 詳細的運動學參數以及座標轉換矩陣, 列於附錄一中。

### 3. 機械手動力學

機械手的動態運動方程式，可以由物理的運動定理 (Newtonian 和 Lagrangian mechanics) 與量測的實體 (physical) 參數推導出來。一般常以 Lagrangian/Newtonian 公式為基礎推演，應用到開式關節鏈 (open articulate chains)，並以 Denavit-Hartenberg 轉換矩陣方式表示之。Paul [ 2 ]，Bejczy [ 3 ]，Orin [ 4 ] 曾利用上述原理推導簡化六個關節的機械手動態方程式。以下分兩小節簡明地討論 Lagrangian 和 Newtonian 的運動公式。文中黑體鉛字 (boldface) 小寫符號是表示向量。

#### Lagrangian-Euler 公式

今考慮一以齊次座標表示的位置向量， $P(x, y, z, 1)^T$ ，此向量是代表  $i$  機械臂上的小單位質量 (differential mass，簡稱為  $dm$ )，到基準座標的位置向量，其關係式以下列式子說明：

$$P = T_0^i \gamma_i \quad (5)$$

$$\gamma_i = (x_i, y_i, z_i, 1)^T$$

$\gamma_i$ ：機械臂  $i$  上的  $dm$  在  $i$  座標系的位置向量因此  $dm$  對基準座標的相對速度是：

$$V_0^i = \frac{d p}{d t} = \left[ \sum_{j=1}^i \frac{\partial T_0^i}{\partial \dot{q}_j} \dot{q}_j \right] \tau_i \quad (6)$$

$$i = 1, 2, \dots, 6$$

結合成的動能 (associated kinetic energy),  $dk_i$ , 等於式子 (7)。

$$\begin{aligned} dk_i &= \frac{1}{2} \text{TRACE} (V_0^i (V_0^i)^T) dm \\ &= \frac{1}{2} \sum_{j=1}^i \sum_{k=1}^i \text{TRACE} \left\{ \frac{\partial T_0^i}{\partial \dot{q}_j} \tau_i (\tau_i)^T \right. \\ &\quad \left. \cdot \left[ \frac{\partial T_0^i}{\partial \dot{q}_k} \right]^T \dot{q}_j \dot{q}_k \right\} \quad (7) \end{aligned}$$

將每個機械臂上的小單位質量積分起來, 可得式子(8)動能等式。

$$\begin{aligned} k \cdot E &= \sum_{i=1}^6 \int dk_i \\ &= \sum_{i=1}^6 \left\{ \frac{1}{2} \text{TRACE} \left\{ \sum_{j=1}^i \sum_{k=1}^i \frac{\partial T_0^i}{\partial \dot{q}_j} J_i \left[ \frac{\partial T_0^i}{\partial \dot{q}_k} \right]^T \right. \right. \\ &\quad \left. \left. \cdot \dot{q}_j \dot{q}_k \right\} \right\} \quad (8) \end{aligned}$$

$J_i$  :  $i$  機械臂之虛擬慣性矩陣 (pseudo inertia matrix)

; 以矩陣形式表示則成下式:

$$J_i = m_i \begin{pmatrix} \frac{1}{2}(-k_{i11}^2 + k_{i22}^2 + k_{i33}^2) & k_{i12}^2 & k_{i13}^2 & \bar{x}_i \\ k_{i21}^2 & \frac{1}{2}(k_{i11}^2 - k_{i22}^2 + k_{i33}^2) & k_{i23}^2 & \bar{y}_i \\ k_{i31}^2 & k_{i32}^2 & \frac{1}{2}(k_{i11}^2 + k_{i22}^2 - k_{i33}^2) & \bar{z}_i \\ \bar{x}_i & \bar{y}_i & \bar{z}_i & 1 \end{pmatrix} \quad (9)$$

其中

$s_i^* = (\bar{x}_i, \bar{y}_i, \bar{z}_i, 1)^T$ ; 以  $i$  座標系為參考座標，  
 $i$  機械臂的質量中心向量。

$k_{ijk}$  = 以  $i$  座標系為原點， $i$  機械臂之迴轉半徑 (radius of gyration); " $j, k$ " ( $j, k = 1, 2, 3$ )。

同理機械手的總位能 (potential energy) 是各個機械臂的位能總合，以底座座標系為參考座標可得：

$$P.E. = \sum_{i=1}^6 P_i = \sum_{i=1}^6 -m_i g T_0^i s_i^* \quad (10)$$

其中

$$g = (0, 0, g_z, 0); \text{ 重力加速向量} \quad (11)$$

$$g_z = 9.8062 \text{ 公尺 / 秒平方}$$

應用  $L-E$  運動方程式，求  $L = K.E. - P.E.$ ; Lagrangian

函数，可得關節  $i$  需要的供應轉矩  $\tau_i$  (generalized torque / force)；亦即推動  $i$  機械臂運動的力。如下：

$$\begin{aligned} \tau_i &= \frac{d}{dt} \left[ \frac{\partial L}{\partial \dot{q}_i} \right] - \frac{\partial L}{\partial q_i} \\ &= \sum_{k=i}^6 \sum_{j=1}^k \text{TRACE} \left\{ \frac{\partial T_0^k}{\partial q_j} J_k \left[ \frac{\partial T_0^k}{\partial q_i} \right]^T \right\} \ddot{q}_i \\ &\quad + \sum_{m=i}^6 \sum_{j=1}^m \sum_{k=1}^m \text{TRACE} \left\{ \frac{\partial^2 T_0^m}{\partial q_j \partial q_k} J_m \left[ \frac{\partial T_0^m}{\partial q_i} \right]^T \right\} \\ &\quad \cdot \dot{q}_j \dot{q}_k - \sum_{j=i}^6 m_j g \frac{\partial T_0^j}{\partial q_i} s_{j_i}^* \end{aligned} \quad (12)$$

其中  $i = 1, 2, \dots, 6$

(12) 式是以矩陣的結構表示，在控制的觀點是很適用的。

，該式可寫成常微分方程的形式如下：

$$D(q) \ddot{q} + H(q, \dot{q}) + G(q) = \tau \quad (13)$$

其中

$D(q)$ ：6 × 6 慣性加速對稱矩陣。

$$D_{ik}(q) = \sum_{j=\max(i,k)}^6 \text{TRACE} \left\{ \frac{\partial T_0^j}{\partial q_k} J_j \left[ \frac{\partial T_0^j}{\partial q_i} \right]^T \right\} \quad (14)$$

$i = 1, 2, \dots, 6$

$H(q, \dot{q})$ ：6 × 1 非線性 coriolis 和離心力向量。

$$H_{i, km} = \sum_{j=\max(i, k, m)}^6 \text{TRACE} \left\{ \frac{\partial^2 T_0^j}{\partial q_k \partial q_m} J_j \left[ \frac{\partial T_0^j}{\partial q_i} \right]^T \right\} \cdot \dot{q}_k \dot{q}_m \quad (15)$$

$$i, j, k = 1, 2, \dots, 6$$

$G(q)$ : 機械臂的重力負荷向量 ( gravity )。

$$G_i(q) = - \sum_{j=i}^6 m_j g \frac{\partial T_0^j}{\partial q_i} s_i^* \quad (16)$$

$$i = 1, 2, \dots, 6$$

$q = (q_1, q_2, \dots, q_6)^T$ ; 位置向量

$\dot{q} = (\dot{q}_1, \dot{q}_2, \dots, \dot{q}_6)^T$ ; 速度向量

$\ddot{q} = (\ddot{q}_1, \ddot{q}_2, \dots, \ddot{q}_6)^T$ ; 加速度向量

$\tau = (\tau_1, \tau_2, \dots, \tau_6)^T$ ; 供應轉矩向量。

L-E公式的物理意義較為明顯，各關節間的耦合效應可清楚的表示出來，因此在設計控制器時，做非線性效應的補償製作分析比較容易。然而 (12) 式的計算卻是非常的費時，特別是當關節數目增多時，計算變得相當複雜，實際應用會有困難 [5, 6]。

### Newton-Euler 公式

此方法主要的基本觀念是將每個機械臂視為一自由體 ( free body )，然後計算機械臂運動的速度、加速度和角速度、角加速度。以順向遞迴的方式 ( forward recursive ) 由固定的底座往外



推算到手端上，然後再由手端背向遞迴方式 ( backward recursive ) 計算關節力矩到底座。此方法最大的貢獻是節省計算關節力矩的時間，而使應用於即時控制變得可能。本節將扼要地說明 N-E 公式的方法 [ 6 ]。

座標系統在前面已經定義了，底下的一些符號定義，若無特別聲明者，則參考座標系皆以  $i$  座標系為基準而言。

$m_i$  :  $i$  機械臂的總質量

$I_i$  :  $i$  機械臂在質量中心的慣性矩陣

$\omega_i$  :  $i$  機械臂的角速度

$v_i$  :  $i$  機械臂的線性速度

$\dot{v}_i$  :  $i$  機械臂於質量中心點的線性加速度

$F_i$  :  $i$  機械臂上的總外施力向量 ( total external force )

$N_i$  :  $i$  機械臂上的總外施銜量向量 ( external moment )

$s_i^*$  :  $i$  機械臂的質量中心到  $i$  座標系的位置向量

$p_i^*$  :  $i-1$  座標系的原點到  $i$  座標系的向量

$q_i$  :  $i$  關節的位置變量

$f_i$  :  $i-1$  機械臂施於  $i$  機械臂的聯接力

$n_i$  :  $i-1$  機械臂施於  $i$  機械臂的聯接矩力

$\tau_i$  : 施於  $i$  關節上所需的供應轉矩

$A_i^{i-1}$  : 如同上節之定義

[ 6 ] 中所導的N-E運動方程式可分成兩個程序，首先由順向遞回方程式，將機械臂運動的線性速度、加速度、角速度、角加速度和本節臂運動的力、力矩，由底座遞迴傳送到手端上，其式子整理如下：

( I ) 順向方程式 ( forward equation ) :  $i = 1, 2, \dots, n$

$$\dot{w}_i = \begin{cases} A_i^{i-1} (w_{i-1} + z_0 \dot{q}_i) & ; i \text{ 機械臂是迴轉運動} \\ & \text{(以下稱為A情況)} \\ A_i^{i-1} w_{i-1} & ; i \text{ 機械臂是平移運動} \\ & \text{(稱為B情況)} \end{cases} \quad (17)$$

$$\ddot{w}_i = \begin{cases} A_i^{i-1} (\ddot{w}_{i-1} + z_0 \ddot{q}_i + w_{i-1} \times z_0 \dot{q}_i) & ; A \text{ 情況} \\ A_i^{i-1} \ddot{w}_{i-1} & ; B \text{ 情況} \end{cases} \quad (18)$$

$$\dot{v}_i = \begin{cases} w_i \times p_i^* + \dot{w}_i \times (w_i \times p_i^*) + A_i^{i-1} \dot{v}_{i-1} & ; A \text{ 情況} \\ \dot{w}_i \times p_i^* + w_i \times (\dot{w}_i \times p_i^*) + 2 w_i \times A_i^{i-1} z_0 \dot{q}_i \\ + A_i^{i-1} (\dot{v}_{i-1} + z_0 \ddot{q}_i) & ; B \text{ 情況} \end{cases} \quad (19)$$

$$\ddot{v}_i = \ddot{w}_i \times s_i^* + \dot{w}_i \times (\dot{w}_i \times s_i^*) + \ddot{v}_i \quad (20)$$

$$F_i = m_i \ddot{v}_i \quad (21)$$

$$N_i^* = I_i \dot{w}_i + w_i \times I_i w_i \quad (22)$$

其次應用反向遞迴運動方程式，將由後一節臂反作用於本節臂的力量，傳送加至本節單獨作用力而得總共實際之關節力與力矩，此計算由手端背向計算到底座上。如下：

(II) 反向方程式： $i = n+1, n, \dots, 1$

$$f_i = A_i^{i+1} f_{i+1} + F_i \quad (23)$$

$$n_i = A_i^{i+1} n_{i+1} + p_i^* \times f_{i+1} + (p_i^* + s_i^*) \times F_i + N_i \quad (24)$$

$$\tau_i = \begin{cases} n_i^T (A_i^{i-1} z_0) & ; A \text{ 情況} \\ f_i^T (A_i^{i-1} z_0) & ; B \text{ 情況} \end{cases} \quad (25)$$

其中的初值如下：

$$w_0 = 0, \quad \dot{w}_0 = 0, \quad z_0 = (0, 0, 1)^T$$

$$v_0 = g z_0, \quad g = -9.8062 \text{ 公尺 / 秒平方}$$

$f_{n+1}$ ：作用在手端上的外施力

$n_{n+1}$ ：作用在手端上的外施力矩

$n$ ：關節數目

N-E 公式的計算方法，以計算複雜度而言，是和關節數目呈線性增加關係，且  $i$  機械臂的速度、加速度、力及力矩等的計算時間並不受  $i$  機械臂位置影響。對於各種動力學的計算分析，[ 5 ] 有詳細的討論。

#### 4. 動力分析與運動模擬

對於某一機械手，我們可依節 2 所敘述訂定需要之座標系，這些座標系間之彼此關係即可以  $q_i, d_i, \alpha_i, a_i$  來描述，利用這些參數來描述機械手結構，再加上必需的各臂資料如質量、質量中心、轉動慣量等等，我們可將節 3 中之 Lagrangian 及 Newton-Euler 公式寫成一般的計算機程式。程式內容請參見附錄四。

其中主程式中 Data Block 資料係史丹佛手資料，對於其他機械手只要修改這些資料即可，這個程式將計算該設定之特別機械手在不同運動情況下之重力，加速度項及速度項分別造成之力。

我們可以配合一些特定軌跡來計算該種軌跡之動力改變情形。

對機械手系統而言，軌跡的規畫也是相當重要的 [7, 8]。我們採用 5 階多項式軌跡 (5th polynomial trajectory) [9] 為例，此軌跡具有第三次導數連續之特點；軌跡的變化平滑，加速度連續。

對於關節座標系 (joint coordinate) 而言，今考慮機械手從一靜態位置  $q_i = (0, 0, 0.2, 0, 0, 0)^T$ ，各關節同時移動到另一個位置  $q_f = (1.57, 2.36, 1.0, 1.57, 1.57, 1.57)^T$  (這裡我們將機械手機構上的限制也納入考慮，如附錄二)。如設定一秒鐘完成，則軌跡規畫系統將分別產生，如圖 3 所示的位置軌跡 (第一個關節)，及圖 4、圖 5 的速度、加速度軌跡，而各個關節是相似的。

此處，軌跡起點及終點的選擇是經過特殊考慮決定的。當在  $q_i$  位置時，第三個關節所受的重力負載最大，而其他的關節為零；此位置為機械手的歸零位置點 (rest point) 然後一、四、五、六關節分別同時旋轉 90 度，而第二個關節特意讓它旋轉 135 度，使其經過最大的重力負載位置，而第三關節則經過重力零之位置。

為顯示本程式之使用，我們舉例分析 Bejczy 模式 (如附錄三) 之準確性，並探討負載對各關節之影響大小。Bejczy 模式是不考慮耦合及速度效應的影響，因此當速度提高時，所得的供應轉矩與實際所需要的轉矩兩者之間的誤差將變大，圖 6 至圖 11 顯示以下三種情況的轉矩變化及大小：

- (1) 機械手無負載時，N-E 公式所計算得到的供應轉矩。
- (2) Bejczy 模式所計算得到的供應轉矩。
- (3) 在最大負載時，N-E 公式所求得的供應轉矩。

所得的結果顯示，腕部關節在有負載高速運動時，其轉矩誤差是相對的比手臂部份（即前三關節）關節大。其中第四、五關節影響最遠。此現象事實上導致一般採用的定回授增益控制方法，當負載變化時，控制誤差會變大。

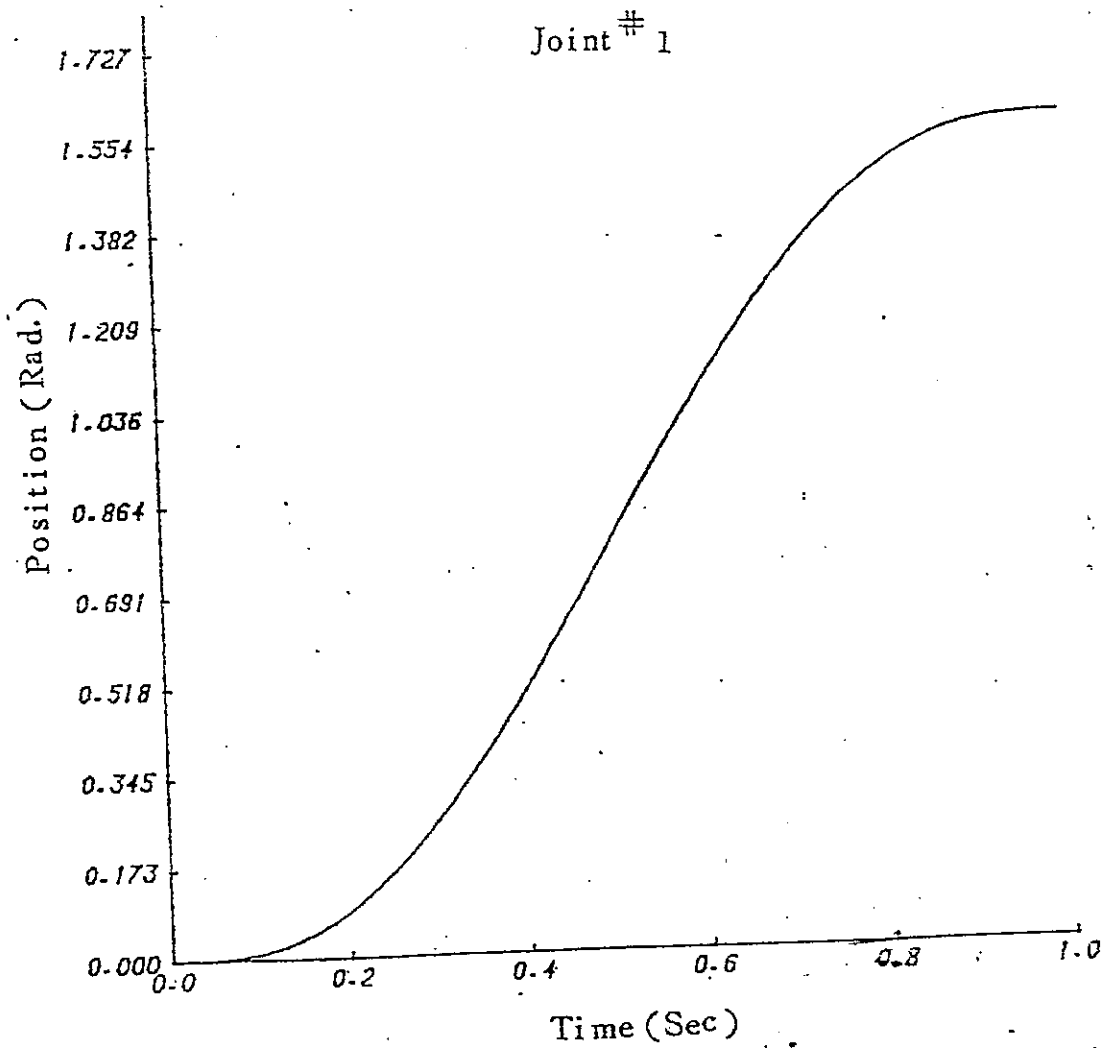


圖 3 位置軌跡

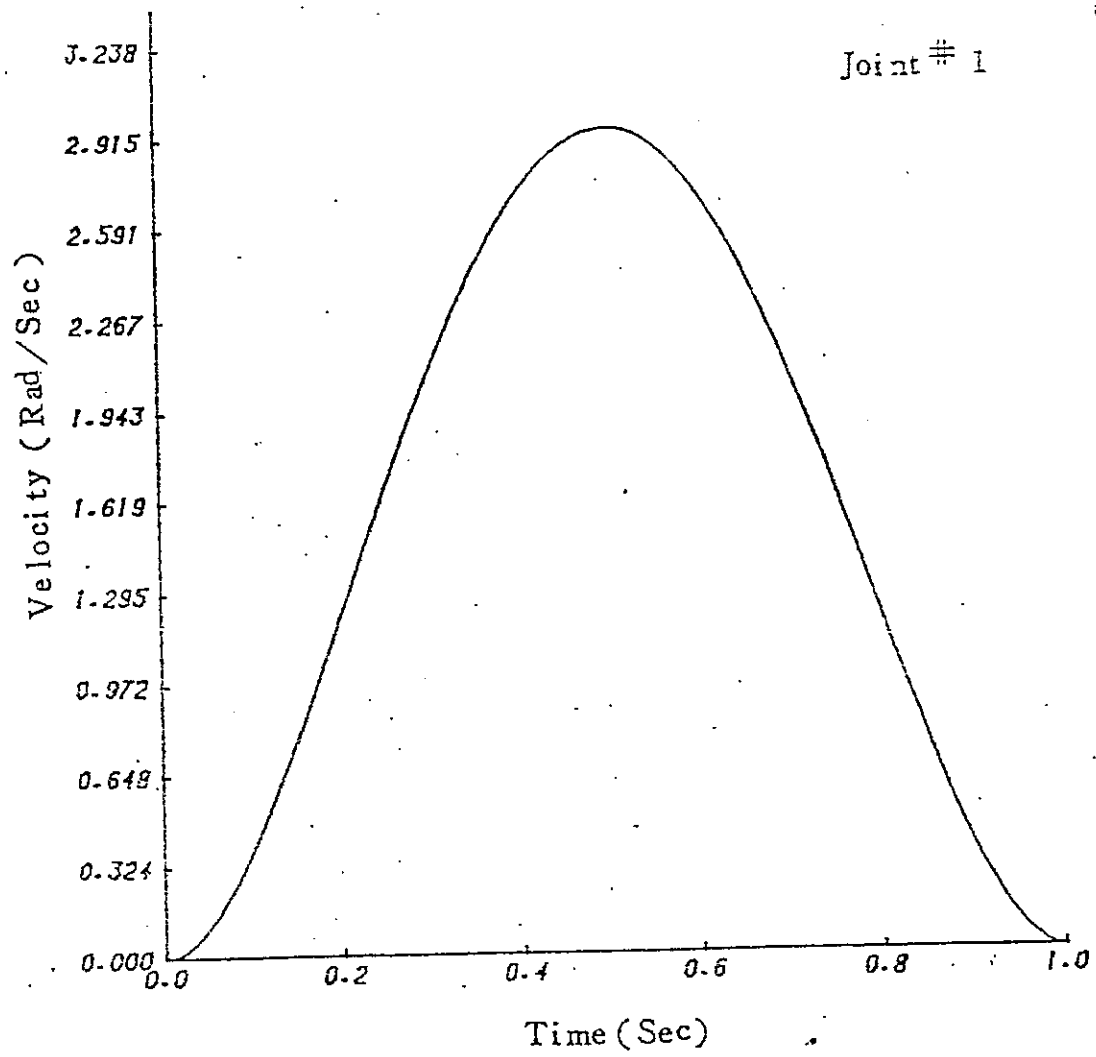


圖 4 速度軌跡



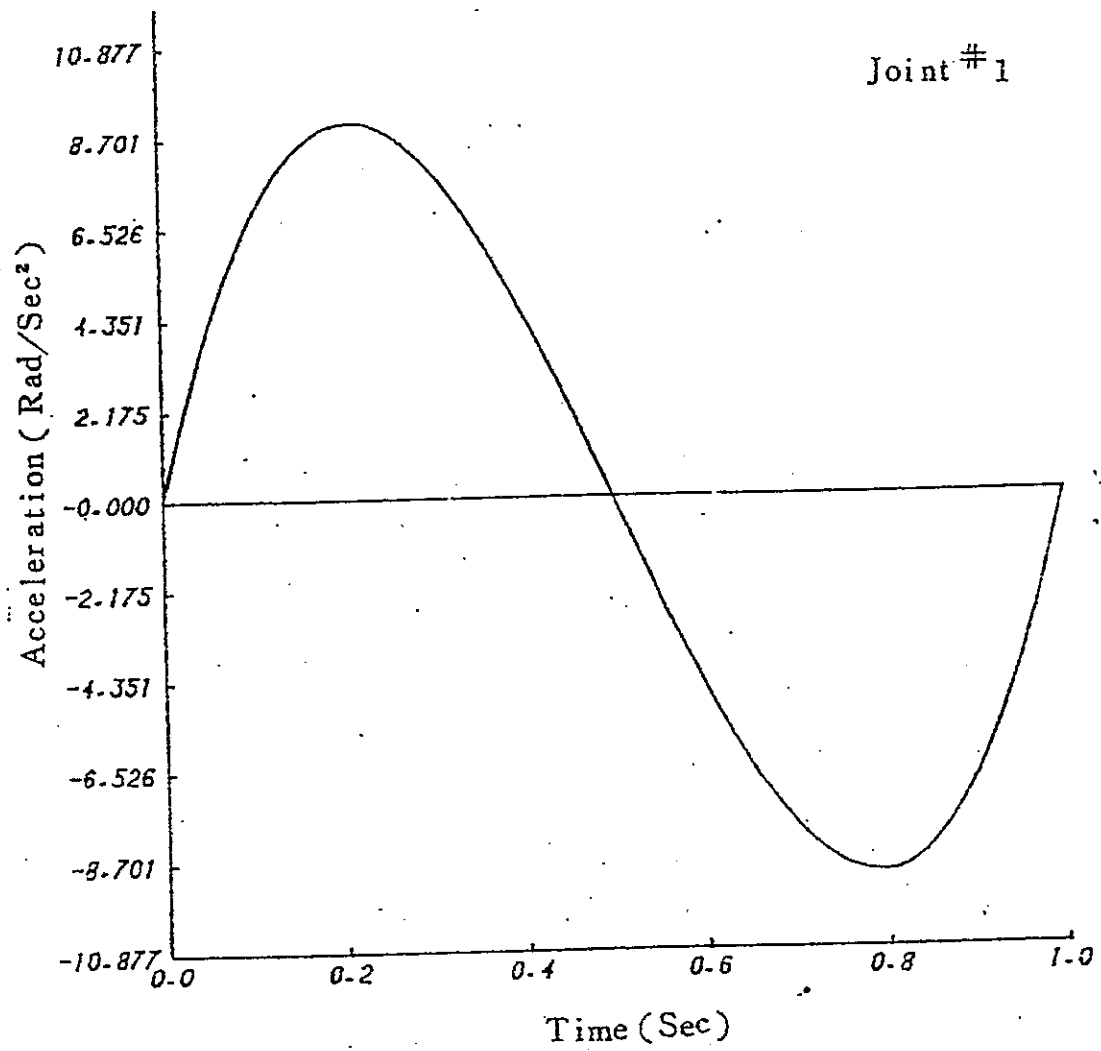
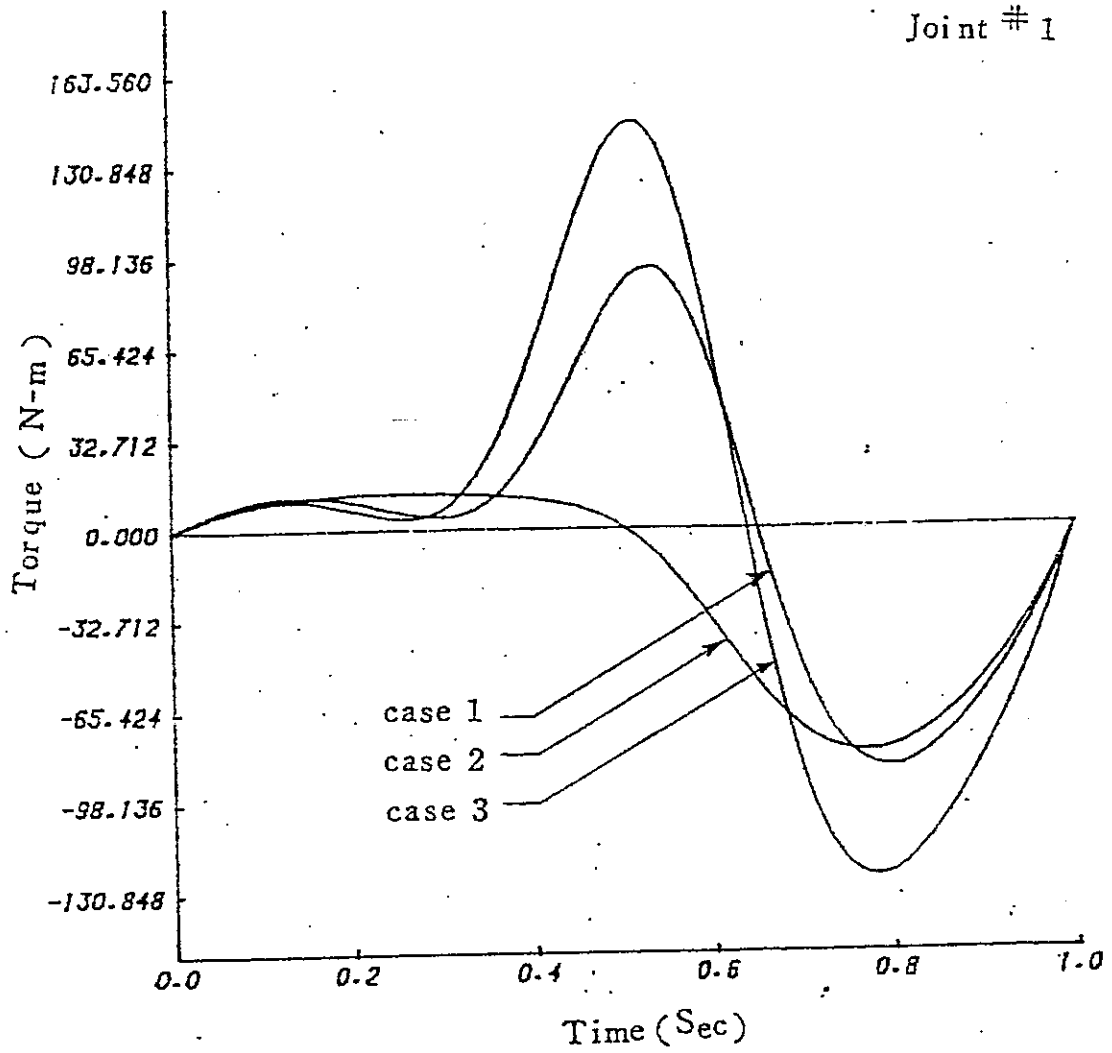
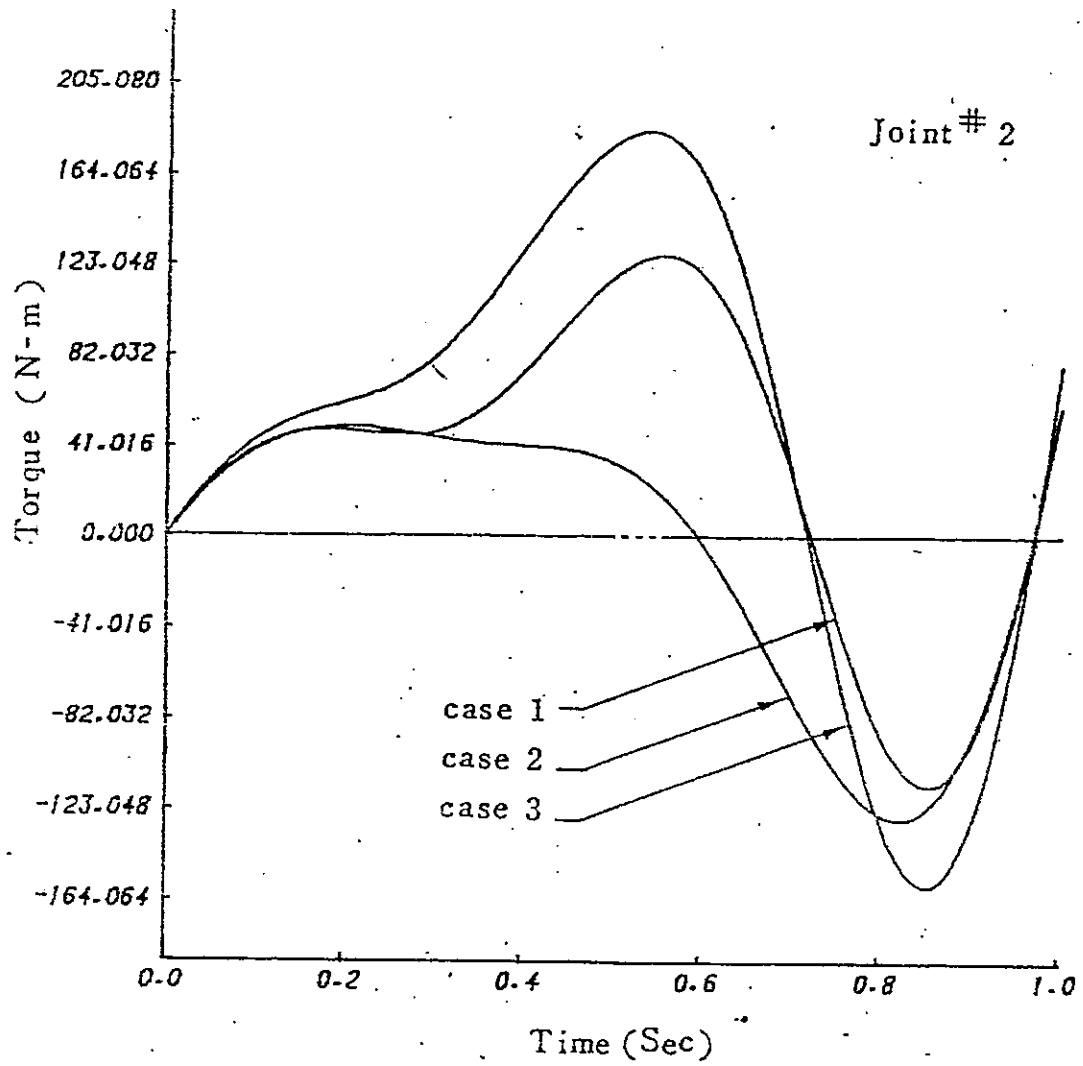


圖 5 加 速 度 軌 跡



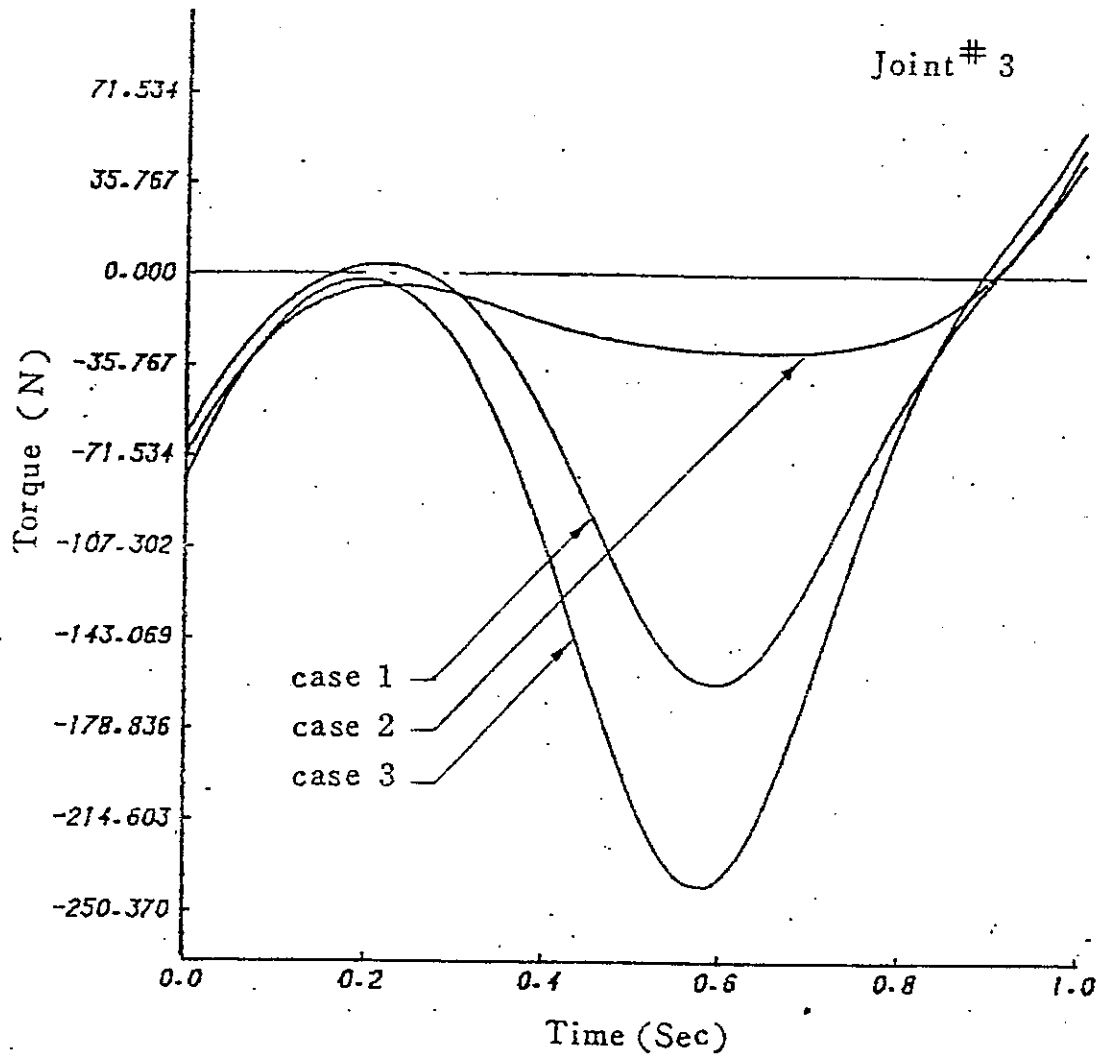
- Case 1 : N - E formulation with no load
- Case 2 : Bejczy model
- Case 3 : N - E formulation with max. load

圖 6 不同情況的转矩軌跡



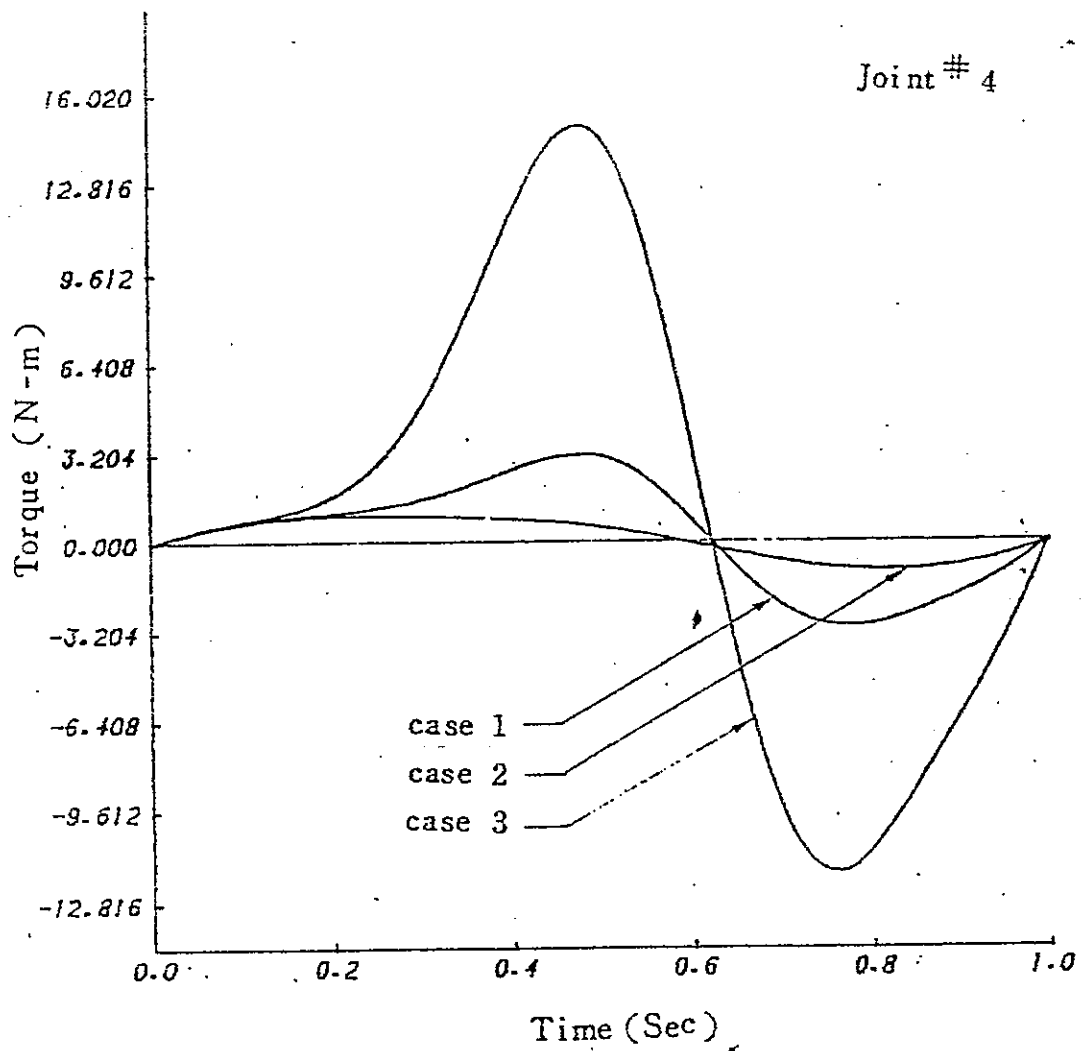
- Case 1 : N-E formulation with no load
- Case 2 : Bejczy model
- Case 3 : N-E formulation with max. load

圖 7 . . 不同情況的轉矩軌跡



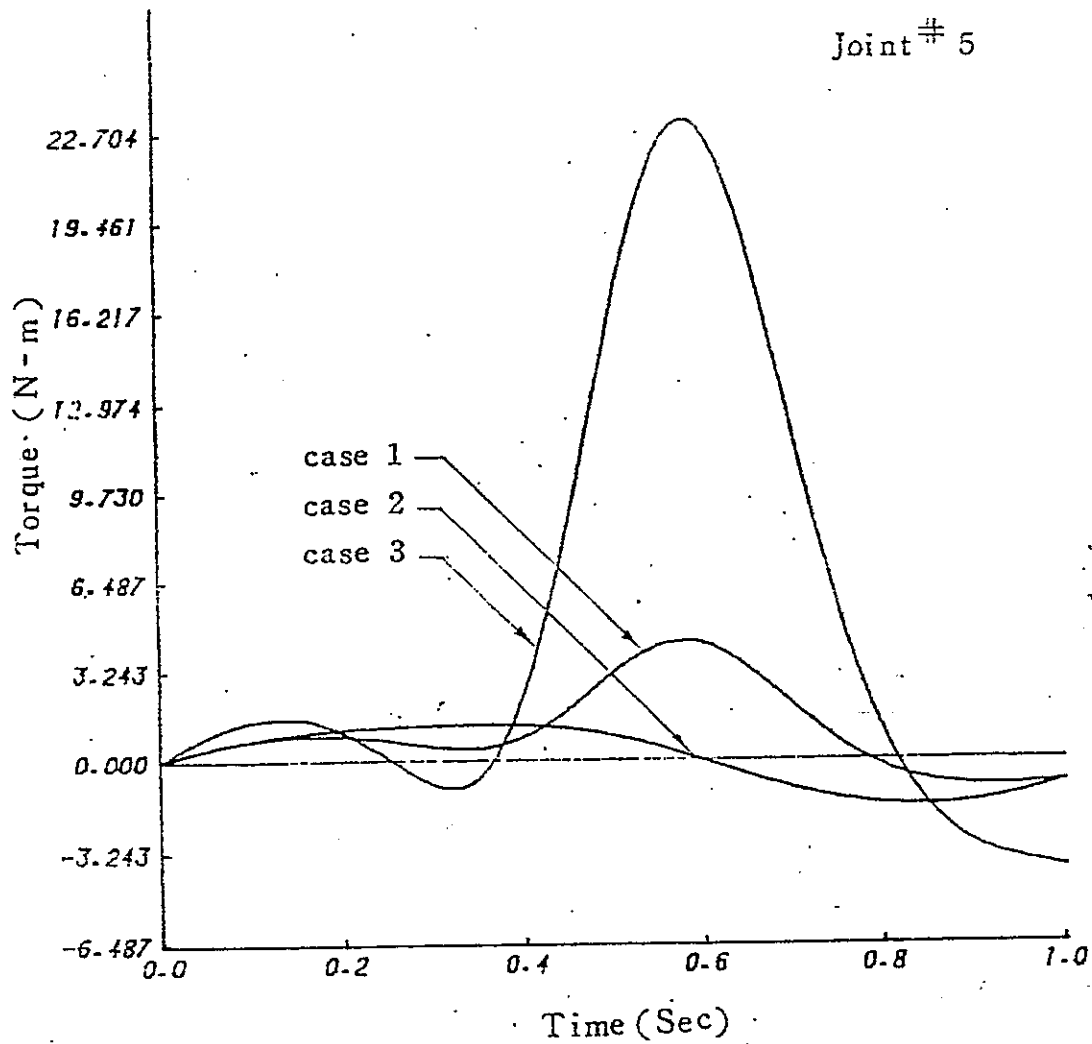
Case 1 : N-E formulation with no load  
 Case 2 : Bejczy model  
 Case 3 : N-E formulation with max. load

圖 8 不同情況的轉矩軌跡



- Case 1 : N-E formulation with no load
- Case 2 : Bejczy model
- Case 3 : N-E formulation with max. load

圖 9 不同情況的转矩軌跡



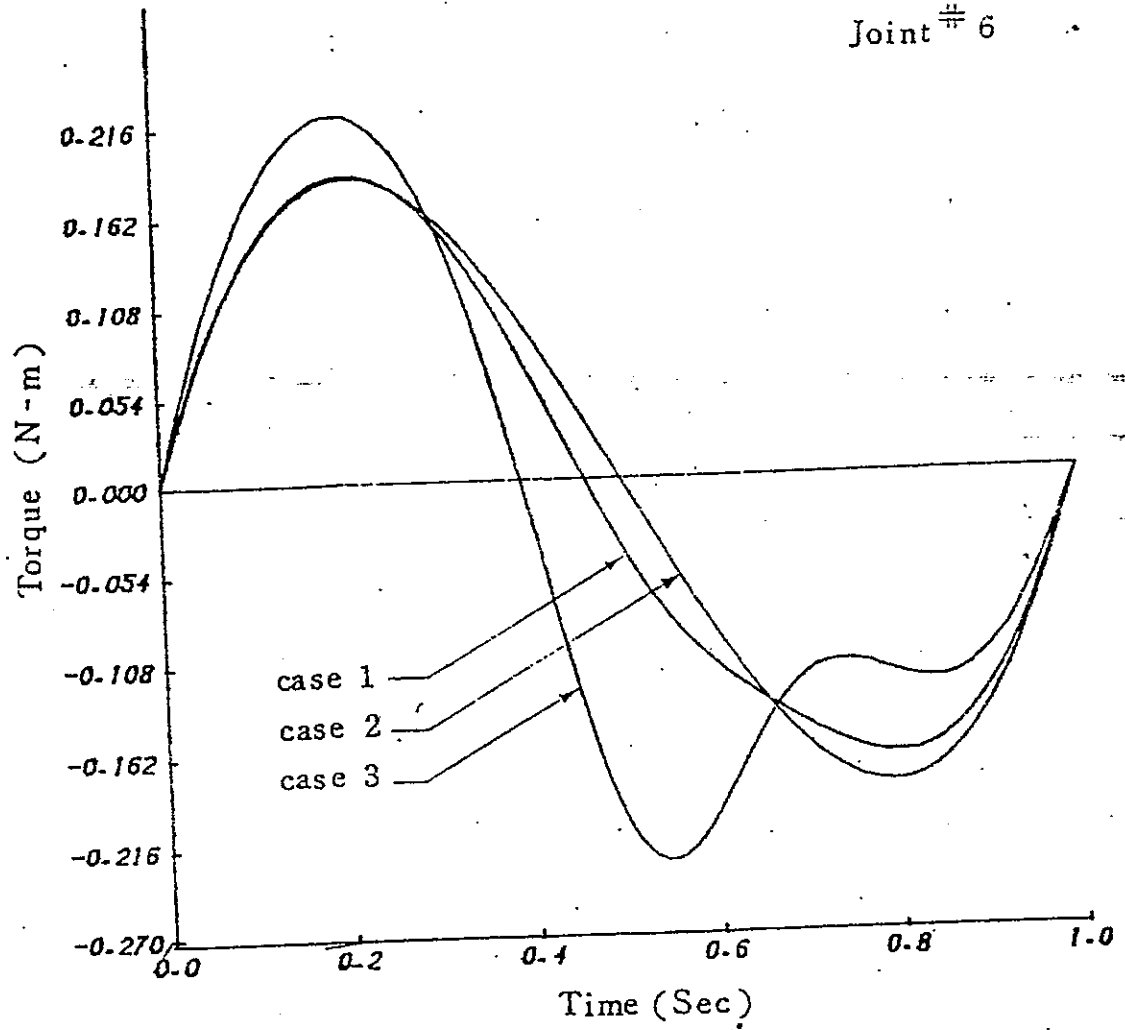
Case 1 : N - E formulation with no load

Case 2 : Bejczy model

Case 3 : N - E formulation with max. load

圖 10 不同情況的轉矩軌跡

Joint # 6



- Case 1 : N-E formulation with no load
- Case 2 : Bejczy model
- Case 3 : N-E formulation with max. load

圖 11 不同情況的轉矩軌跡

至於運動之模擬，可配合 Runge-Kutta-Gill 四階法則，做連續性模式 (continuous model) 的模擬，模擬中最主要的計算乃是由施加之關節力來計算關節運動情形，這個計算，我們可採用同一程式稍加處理即可，附錄五是由施力計算關節加速度之程式可茲利用。

## 5. 結論

機械手係由一系列臂經由關節連結而成，為描述臂間關係，每一節臂必需訂定一個座標系，而 Denavit-Hartenberg 座標訂定法則是一合適方法，其座標間關係可由一些參數表示，至於動力與運動之關係，一般則由 Lagrangian 及 Newton-Euler 公式來運算。本研究綜合這些運動學及動力學建議並設計一套適合及方便一般機械手動力分析及運動模擬使用之程式，該程式讀取關節結構資料可計算關節施力情形，可用於分析及模擬之用，使用者必須根據自己所需要分析或模擬設計部份自己之特殊程式。



## 機械手連續路徑之近似描述

### 1 導論

機械手 (Mechanical Manipulator) 是經由關節組合的一串機械結構，而其每一個關節由激勵器 (actuator) 所驅動。爲了確使工作成功的完成，通常是將這些關節驅動致使手部能被控制去沿著所指定的卡氏路徑 (Cartesian path) 行進。

有許多篇的文章對於上面所提的運動問題已經作了討論。[9] 討論並比較了各種不同關節的運動軌跡，諸如 bang-bang，多項式 (polynomial)，臨界阻尼 (critical damping)，餘弦函數 (cosine) ... 等等軌跡，但都僅包含二點位置，由起始點出發後而至目的地終止。假如要求機械手經過更多的指定位置，則到達每一節點 (knot) 時，均需停止。因此，整個軌跡過程，在時間上相當不經濟，且運動不平滑。此外，有許多篇文章也在卡氏座標運動 (Cartesian Motion) 上下了功夫，[7]、[10]、[11] 於直線路徑上做了一些研究。雖然人們喜歡以卡氏座標爲參考描述運動，但在實際上所控制的力量及轉矩是加在關節上面的，於移動時若以手部的位置描述，則於每個取樣點間均需做關節與手方位的轉換，爲了減少於轉換時之計算上的負荷，於是人們通常考慮使用關節的運動軌跡。

一般而言，手的位置與轉向可以一  $4 \times 4$  矩陣表示之 (看圖 12)

$$H(t) = \begin{bmatrix} n(t) & s(t) & a(t) & p(t) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

其中  $\underline{p}$  是手部位置向量， $\underline{n}$ 、 $\underline{s}$  及  $\underline{a}$  分別是單位法線向量 (unit normal vector)，單位滑行向量 (unit slide vector) 及單位趨近向量 (unit approach vector)。其對應於  $\underline{H}(t)$  的關節值能夠根據機械手的結構解出。

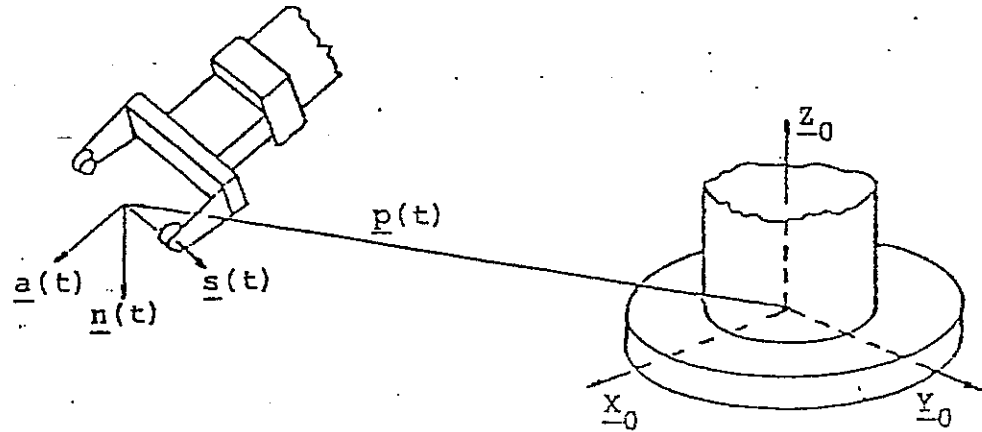


圖 12 手的位置及轉向向量

機械手的移動能夠以一序列卡氏路徑上節點 (Cartesian Knots)  $\{H_1, H_2, \dots, H_n\}$  來表示它的運動路徑。對於一個 6 個關節的機械手而言，這些節點被轉換成六組關節值， $[q_{11}, q_{12}, \dots, q_{1n}]$ 、 $[q_{21}, q_{22}, \dots, q_{2n}]$ ，... 及  $[q_{61}, q_{62}, \dots, q_{6n}]$ ，每一組對應一個關節。近似的關節路徑能夠由這些關節值插值所構成，關節的運動軌跡需一個個關節分別予以規畫，每一次僅考慮一個關節。所以當討論至我們的步驟時，第  $j$  個關節之  $q_{ji}$  是不需要去考慮的，因此，可以  $q_i$  來取代  $q_{ji}$ 。

此篇文章中，我們採用三次楔形函數 (cubic spline function) 來構成關節的運動軌跡。被選擇的  $n$  個卡氏路徑節點再利用片斷的三次多項式 (cubic polynomial) 來做內插。假如所選擇的卡氏

節點被順序的即時產生，例如利用路徑規畫器或是視覺系統產生，則一小段的關節路徑必須於一部份節點已知之時予以決定，在此種情形下，加速度的連續性情況將無法考慮，以完成此一即時的規畫。X-spline 函數 [12] 為一種 cubic spline, 其二次微分不連續，但符合容易計算之要求。可以運用以產生關節運動軌跡，本文研究採用這種關節軌跡，同時建議一應用 X-spline 關節軌跡以達到卡氏座標上路徑規畫之方法。

## 2 X-spline 軌跡

X-spline 由 Glenshaw 及 Negus [13] 於 1978 年研究過，且由 Behforooz, Papamichael, 及 Worsey 加以擴充 [12]。以此種方式，三次楔形函數 (cubic spline) 於各個節點上的二次微分是允許不連續的。

每一關節值可順序的從所選到的卡氏節點中於合理的時間內計算而得。關節本身的運動軌跡可由次二個已知關節值中決定出來。因而不需要在開始運動之前即把路徑上所有的節點都解出來。此種方法是很簡單且適合做即時控制上運用。且在此運用近似的方法上，一直保持著其精確度。二次微分的跳躍值 (jumps) 可利用縮短節點的區間來控制它，使它能保持在可接受的範圍內。

X-spline 的定義已經在 [12] 中作了說明，設  $P_i$  為三次多項式，此多項式為  $q$  在  $t_i, t_{i+1}, t_{i+2}$  及  $t_{i+3}$  所插值的關節運動軌跡函數，且定 4 次多項式  $r_i; i = 1, \dots, n-2,$

$$r_i = p'_i; \quad i = 1, \dots, n-3$$

$$r_{n-2} = p'_{n-3}$$

(26)

### X-spline 的定義

設  $a_i, b_i; i=2, \dots, n-1$  為  $2n-4$  個實數，一片段連續多項式其導數為  $m_i; i=1, \dots, n$  滿足了下列關係

$$m_1 = q'(t_1)$$

$$a_i m_{i-1} + m_i + b_i m_{i+1} = a_i r_{i-1}(t_{i-1}) + r_{i-1}(t_i) + b_i r_{i-1}(t_{i+1})$$

$$i = 2, 3, \dots, n-1$$

$$m_n = q'(t_n) \quad (27)$$

多項式  $r_i$  稱為 X-spline 函數，其參數為  $a_i, b_i; i=2, 3, \dots, n-1$ 。  $r_i$  由方程式(26)所定義

X-spline 函數的唯一存在之充分條件為

$$|a_i| + |b_i| < 1 \quad \text{for } i = 2, \dots, n-1$$

$a_i, b_i$  之一種可能的選擇是

$$a_i = \frac{h_i}{h_{i-1} + h_i}, \quad b_i = 0$$

其中  $h_i$  等於  $t_{i+1} - t_i$  此種選擇使得從方程式(27)中解出  $m_i$  及  $m_{i-1}$  較簡單，事實上，節點的速度  $m_i$  能夠根據節點的位置及原先的速度來明顯的表達，從相等空間的情況  $r_j(t_j), r_j(t_{j+1})$  及  $r_j(t_{j+2})$  能夠從方程式(26)的定義中導出並代入方程式(27)中獲得  $m_i$ ：

$$m_i = (q_{i+1} + 4q_i - 5q_{i-1})/4h - m_{i-1}/2; \quad 2 \leq i \leq n-1$$

注意到  $m_i$  是與  $q_{i+2}$  無關，僅三個關節值及一個關節速度包含於其關係中，一旦  $m_i$  被解出來，則其軌跡  $Q_{i-1}(t)$  能夠根據  $m_{i-1}, m_i$

$q_{i-1}$  及  $q_i$  解得。

### 3 卡氏座標路徑規畫上之考慮

在此我們考慮關節型手臂，我們注意到，由於手的伸長程度不同，擺動同一角度，會導致不同的圓弧，直線軌跡近似的準確性與這圓弧之曲度有密切關係，假如弧度曲度大，則構成軌跡時在卡氏座標的軌跡取樣就需較密。對直線軌跡，我們要求偏差在容忍的範圍內，因此，依下面公式定出兩點取樣間距離為  $\Delta y$ ，兩點間則在關節上做內插近似。

參考圖 13，依三角公式可得：

$$(1/2\Delta y)^2 + (R/2 - 2\Delta x)^2 = (R/2)^2$$

化簡可得

$$\begin{aligned}\Delta y^2 &= 8R\Delta x - 16\Delta x^2 \\ &= 8R\Delta x, \quad \text{當 } \Delta x^2 \text{ 足夠小}\end{aligned}$$

由於允許使用者使用不同速度來走直線軌跡，所以我們以下面方法來考慮，當旋轉同一角度，因臂伸長長度的不同，端點的卡氏速度也不同，由機械手設計的規格中，可以找出手臂的最大伸長度  $R_{MAX}$  及最大  $V_{MAX}$ ，然後以  $v_{lmax} = \left(\frac{r_1}{R_{MAX}}\right) * V_{MAX}$  的公式來估測當手臂伸延長度為  $r_1$  時之速度上限  $v_{lmax}$ 。

根據上述決定卡氏軌跡取樣間隔及速度之方式，由 A 點移動到 B 點的直線軌跡即可以下列步驟完成：

step 1 以機械手基底 (base) 到線段 AB 的最短距離為  $\frac{R}{2}$ ，算出  $\Delta y$  之值，把 AB 切成  $N$  段  $\Delta y$  長度的線段。

$$N = \frac{L(AB)}{\Delta y}; \quad L(AB) \text{ 表 } \overline{(AB)} \text{ 線段的長度}$$

step2 : 以機械手要走的線段AB間最短距離： $r_{min}$

$$\text{求出 } v_{min2} = \left( \frac{r_{min}}{R_{MAX}} \right) * V_{MAX} * \alpha$$

並以  $v_{min2}$  表示在線段AB間的正常速度。

step3 : 依使用者所給速度量不同，在指令中 (speed 的 量) 求出相對應的速度。

$$V_{spd} = (\text{speed}/100) * v_{min2}$$

step4 : 決定了  $V_{spd}$ ，我們便估計走長度為 1 的AB線段中須花費時間。

$$t = 1/V_{spd}$$

step5 : 在線段頭尾加減速段間隔取  $1/2\Delta y$ ，而在中間以  $\Delta y$  作為一段的長度，然後在每段區間以時間  $t$  用 X-spline 去做內插，然後決定每個區間內所須做的伺服次數，在區間內依 X-spline 內插，依伺服週期送出關節值、關節速度、關節加速度。

#### 4. 結論

許多機械手操作及應用均要求路徑的控制，這些路徑一般皆以人們所直接可以見得到的地面座標來描述，但在控制時，控制的量卻是關節值，因此，由手端方位轉換至關節值乃屬必要，而這個轉換卻又相當費時，為求減少運算，在路徑上選取足夠代表之手方位資料，然後再在關節上做內插是一可行方法，X-spline 則是一可以採用之內插函數，其二次微分不要求連續使得加速度之連續性受到損害，但卻可達到使用三個區域性資料來做規畫的

好處，在使用此種內插時，運動速度之限制乃成爲規畫上一必須考慮的因素，我們針對關節型也提出一些這方面可循之基本原則。

1 導言

機械手的發展在近幾年來在工業界及學術研究方面都受到廣泛的注意，在工業方面的應用，經由計算機的控制，機械手能在輸送帶上抓，放物體以及從事裝配、焊接、噴漆以及一些危險性工作，同時，因為加上感應器 (sensor)，使得它對不同工作的適應性更為提高。

概略的講，利用計算機來達到控制目的可分下列幾種可能方式：A. 用教導 (teaching) 的方法來完成。B. 使用高等智慧型的程式法。C. 文章式 (texture) 的程式法。

使用教導 (teaching) 方法完成時，機械手是經由人手或教導盒操作，以教導機械手，教導完成之後，所有的動作資料被存在記憶器中，機械手便可依命令重覆做這一連串動作。這種方法使用非常簡單，一個工人可以很快地學會，但此種方式很難使用感應器的情報以處理意外情況，同時有些工作很難以這種教導方式進行。

在高等智慧型的程式法方面有學者做過一些嘗試，如史坦福人工智慧研究室的“問題解決系統”(problem solving system)

[14]，系統嘗試著自動把一個欲完成的工作，分解成一連串待完成的次工作 (subtask)，然後重覆這種方法，直到問題簡化到可以輕易的解決，其主要觀念在於模仿人類的智慧，但由於問題的解析其整個過程非常複雜，因此不是有效的解決方法。

一個可行的方式，便是介於上述二種方法取折衷的文章式程式法，一個機械手程式語言的系統提供一些基本動作，它們的功



用在幫助程式者避免不必要的細節，同時將軟體工程中結構化的觀念引入程式，這種方式是目前最爲可行而實用的方法，本研究重點也在於此。

在這方面，史坦福大學於1970年左右曾發展出AL語言(Assembly Language)它提供了空間座標系統的使用，其間的座標轉換，及機械臂的不同運動方法。由於該語言對資料結構有完整的規劃，具有豐富的運動方式描述及結構化的程式觀念，使同期如VAL等語言都採用其成功處，RLO的設計也是基於AL語言，基本上這個語言系統可接受語言指令，處理後產生伺服控制部份在控制時所需之軌跡資料。本文即提出此處理系統規劃及實現方法。

## 2 RLO 語言

RLO系統三個主要部份包括語言、轉譯器及直譯器，其間的關係可以圖1表示。使用者的程式，經過轉譯器翻成中間碼，送入直譯器，產生軌跡資料送至伺服控制部份以執行對機械手及周邊器的控制。

RLO語言設計的重點在提供合適的機械手運動方式及適合的資料結構來幫助程式者完成其工作。其指令可分成5類，包括(1)資料形式(data type) (2)始值指令(3)連結架子及脫離的指令(4)工具控制及感應器指令(5)運動的指令。我們分別說明其內容如下：

### (1) 資料形式：

RLO的資料形式可分爲純量(scalar)，座標轉換(trans)及架子(frame)。

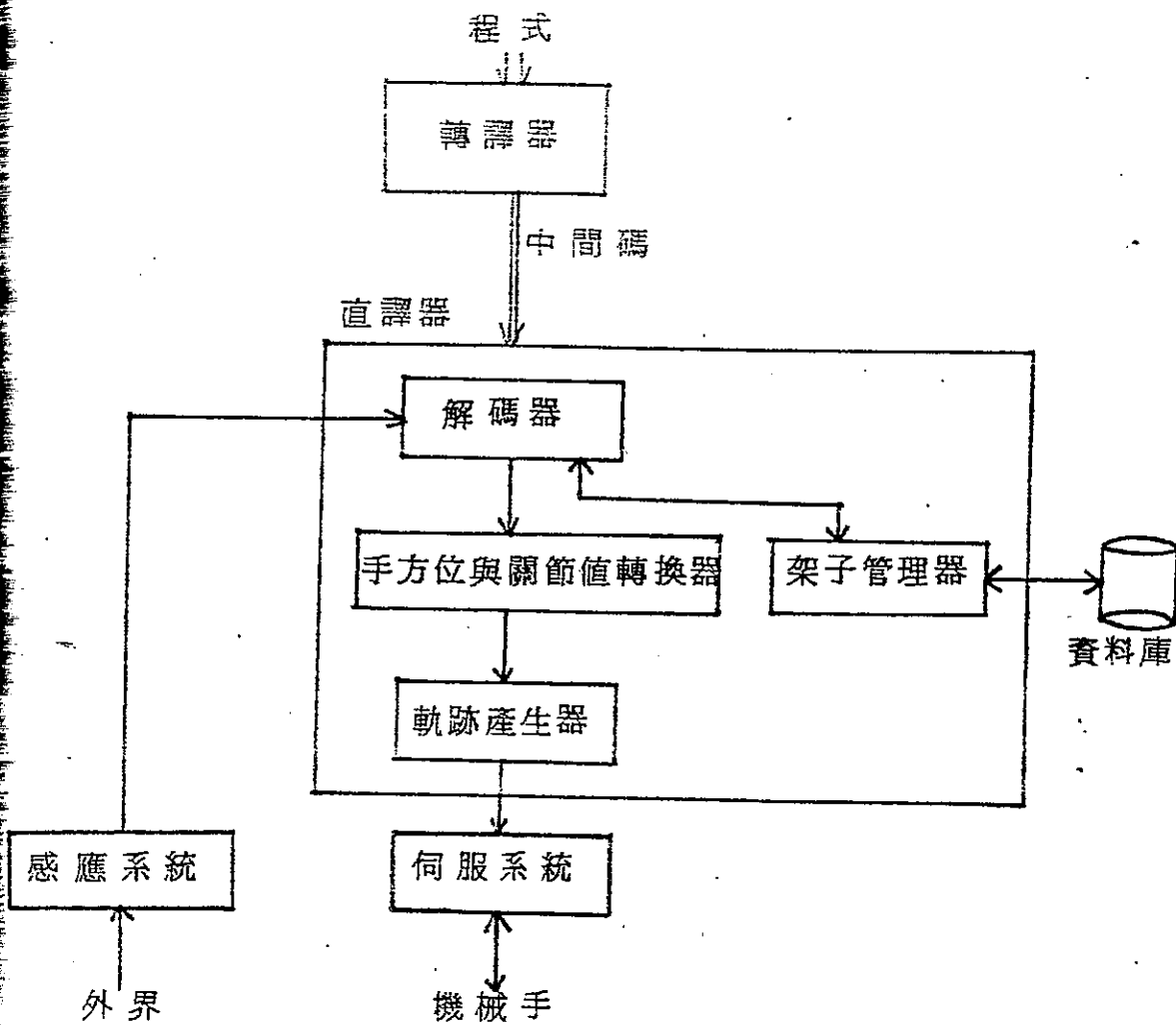


圖14 轉譯器、直譯器之關連

純量：由 1-9 個字元 (character) 組成，可用來代表一維的物理量，如溫度、時間等。

座標轉換：由四個 3 維向量組成，代表座標間的轉換。

架子：在空間的一點，可以給予一組座標，用來表示其位置及轉向 (position & orientation)，圖 2 即顯示兩個架子，即架子 1 及 2，以均勻矩陣表示，則

架子 1 ( $f_1$ ) 可表示為

$$\begin{bmatrix} 1 & 0 & 0 & px_1 \\ 0 & -1 & 0 & py_1 \\ 0 & 0 & -1 & pz_1 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

架子 2 與 架子 1 間的座標轉換  $t_1$  可以表為

$$t_1 = \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

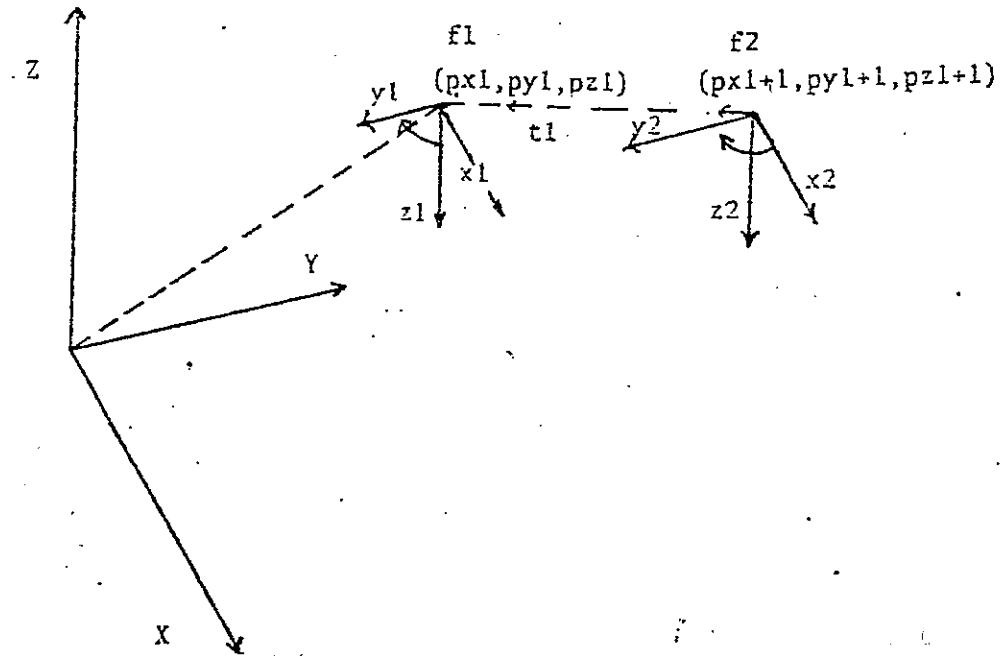


圖 15 空間點的架子表示法

(2) 始值指令：

在程式中，可給架子或座標轉換初值，它將一個 4x4 的矩陣表前三行，放入對應的變數內，例如

```
t1 = (1,0,0,0,1,0,0,0,1,-1,-1,-1);
```

(3) 連結架子及脫離的指令：

由於移動物體會導致多個架子的變動，為了方便使用者，系統提供連結架子的指令 `affix`。"`affix f1 to f2`" 表示 `f1` 與 `f2` 兩個架子有連結關係，當 `f2` 移動了，則系統自動將 `f1` 改變。"`affix f1 to f2`" 表示 `f1` 與 `f2` 兩個架子之間連結關係，是以 `t1` 這個座標轉換表示。系統同時也提供使架子脫離連結關係的指令 `unfix`。"`unfix f1 from f2`" 表示 `f1` 與 `f2` 兩個架子不再有連結關係，而系統將把 `f1` 此時對整個系統基底的座標求出。

(4) 工具控制及感應器指令：

當程式者希望以機械手抓取物體，他可以用 "`open ARM-1`" 來達成，這個指令的效果是使 `ARM-1` 的手指張開，然後再以抓取 "`grasp`" 這個指令使機械手抓住物體。讀取視覺系統資料可以 "`Read visions (f1)`" 來達成，系統啟動視覺系統，將得到的物體方位值，放入架子 `f1` 中，另外 "`Read system (ARM-1)`" 可以將機械手 1 的資料放入 `ARM-1` 中以幫助程式者知道系統的資料；如果機械手前端接著工具，則可以用啟動工具 "`Operate tool1`" 來啟動工具 1。

(5) 運動的指令：

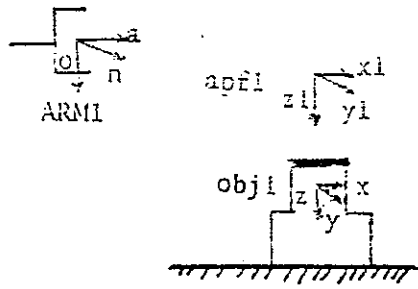
`RLO` 對機械手的運動方式分為細部動作 (`fine motion`) 及粗略

動作 (gross motion) 。細部動作是當機械手接近物體時必須有謹慎的動作方式，以免碰撞四周物體，粗略動作則表示當機械手在相隔較遠物體間移動時，而可以用較粗略的軌跡控制，以求較短的計劃時間。在程式中為避免使用者考慮太多，於是使用 `with approach, with departure` 的運動描述語句來幫助使用者，例如 `move ARM-1 to obj1 with approach = f1;` 則機械手會移動到物體 `obj1` 的逼近點 `f1`，而在 `f1` 與 `obj1` 之間是以直線接近。如此，則不會破壞 `obj1`，離開時亦然。

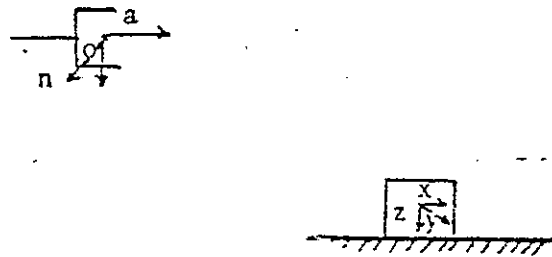
這種處理方式的好處在於程式者可以事先考慮好以何種角度、位置抓取物體，然後再設計逼近點，以確信不會碰到物體。

系統提供的指令，對於機械手的運動速度可以用 `with speed=n1` 來達成 ( $1 < n1 < 120$ ) 當 `n1=120` 表示以機械手容許的最大速度運動，其餘值則成比例的使機械手的速度下降。為了避免碰撞，移動路徑可以指定機械手必須經過某些點；用 `via (f1, f2, ..., fn)` 的運動描述語句完成，機械手在經過 `f1, f2, ..., fn` 的架子時為連續運動，而不在每個架子 `fi` 上停止，再前進。另外，直線運動方式可以用 `straightly` 的描述語句表示，`directly` 則表示不考慮直線或經由某些點或以逼近點的運動方式，直接由起始點到終點的運動方式，但是仍然可以使用 `with speed` 的描述語句，以下以例子說明對運動指令的使用。

例 1：以 ARM-1 抓取 `obj1`，如下圖可以用：`move arm-1 to obj1 with approach = apf1` 來完成。

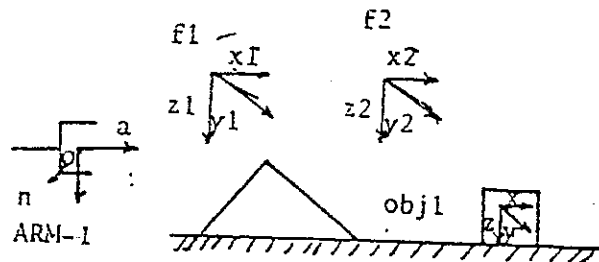


例 2 : ARM-1 由現在位置直接移動到 obj1 的地方，如下圖



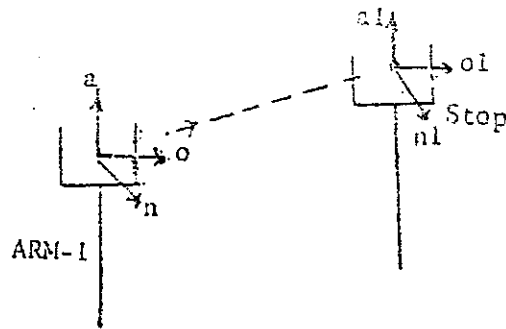
可使用 `move ARM-1 to obj1 directly, with speed = 100`  
 或 `move ARM-1 to obj1 with speed = 80`, 如果沒有 `directly` 或  
 其他有關運動方式，系統假設使用者希望以 `directly` 方式  
 完成。

例 3 : 移動 ARM-1 到 obj1 的位置，而且希望避開 / 或經過某些  
 點，如下圖：



可以 `move ARM-1 to obj1 via (f1, f2)` 來完成。

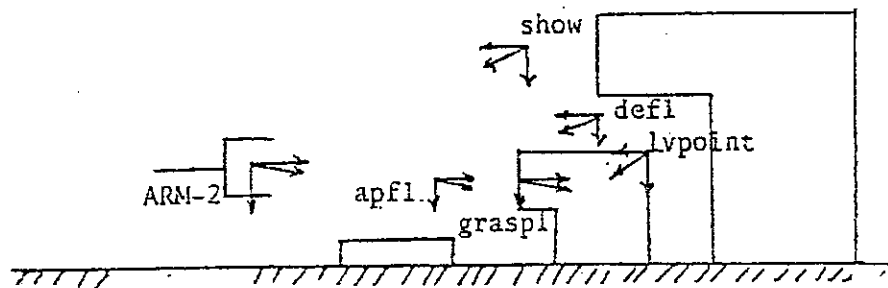
例 4 : 沿著直線移動，如下圖：



move ARM-1 to stop straightly, with speed = 100

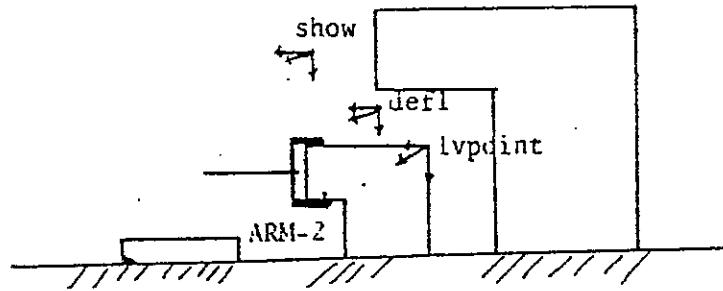
以上的例子，機械手的移動方式都以本身的座標為著眼點，當機械手抓住某個物體後，程式者可能希望以物體本身為參考沿某種軌跡前進，這時系統提供一種運動指令的形式，也就是可以用物體本身座標為參考點，如下面的例子，可以下列一串指令完成。

①



move ARM-2 to grasp 1 with approach = apfl ;

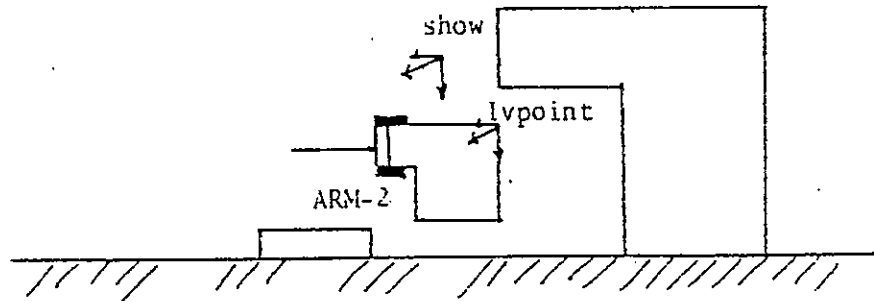
②



move lvpoint to show with departure = def 1 ;



③



④

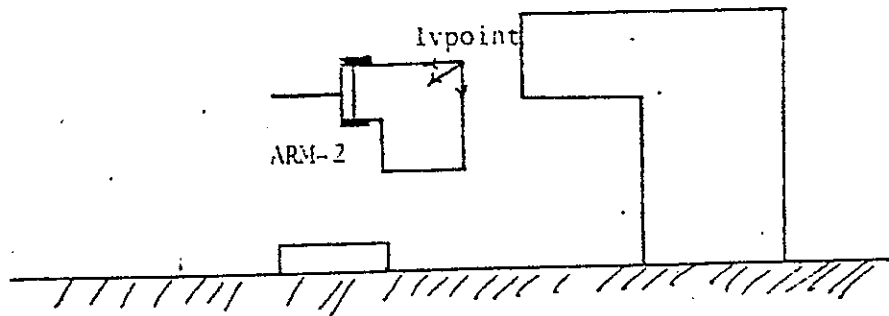


圖 16 "移動" 指令的示意圖

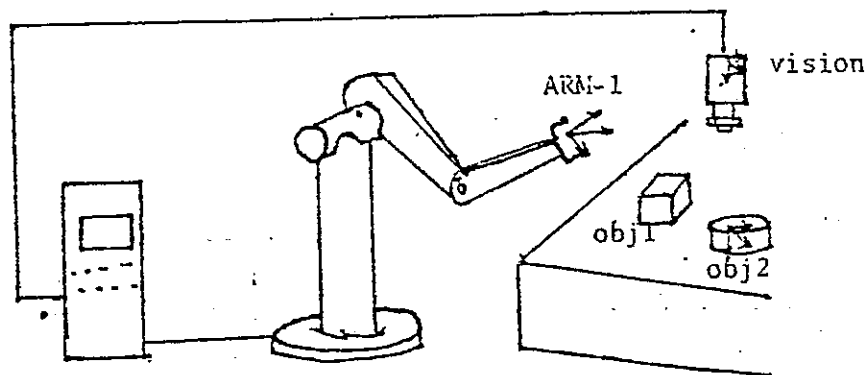


使用這些運動指令程式者可以專心在物體的移動，不必考慮如何控制機械手，系統幫助程式者解決座標轉換的問題。

底下是一個以 RLO 控制語言程式例。

工作：以視覺系統找出物體 1 與物體 2 的位置，然後先抓住物體 1，接著以物體 1 的座標為準，將它疊在物體 2 上，然後放開物體 1。

```
100   VAR   OBJ1, OBJ2, STOP : FRAME ;
200   BEGIN
300       READ VISIONS (OBJ1) ;
400       READ VISIONS (OBJ2) ;
500       MOVE ARM-1 TO OBJ1 ;
600       OPEN ARM_1 ;
700       GRASP ;
800       AFFIX OBJ1 TO ARM_1 ;
900       MOVE OBJ1 TO OBJ2 ;
1000      OPEN ARM_1 ;
1100      UNFIX OBJ1 FROM ARM_1 ;
1200      MOVE ARM_1 TO STOP ;
1300  END.
1400
1500
```



### 3. 轉譯器及直譯器

#### 轉譯器工作原理：

RLO 的轉譯器設計是採用遞迴下降法 (Recursive descent) 其工作原理是由上往下 (top-down)，語法圖 (syntax graph) 即代表剖析語句 (sentence) 的進行方向，每遇到一個非終端符號，就建立一個子轉譯器 (subparser)，每個子轉譯器的任務即辨認這個非終端符號的語句，這個過程重覆下去，直到認出整個語句，然後填入表格內容及產生中間碼。

#### 直譯器 (interpreter) 的設計：

直譯器的功能為執行中間碼所指明各種功能，完成軌跡計劃，座標轉換，架子管理，與外界感應器變通等工作。我們分別說明直譯器中主要組件如下：

#### A. 架子管理器 (frame manager)

架子的基本概念是對於空間中一點，我們可以給予一個  $4 \times 4$  的矩陣以代表其位置與轉向，即  $[\underline{n}, \underline{o}, \underline{a}, \underline{p}]$ ； $\underline{n}, \underline{o}, \underline{a}$  為  $3 \times 1$  向量，合併可代表此點的轉向， $\underline{p}$  為  $3 \times 1$  向量，代表此點的位置。因此，由架子本身與架子之間的連結關係，可以用來做為外界物體的模式，以圖 3 為例我們可使用一個往上指的樹狀結構來表示架子間關係，每個節點代表一個架子，每個分枝代表架子間的連結關係 (affix relation)，樹的根為一個基底架子 (world frame)，指向同一節點的兩個架子，表示兩者有相同的基底，所以在觀念上可以表示如下

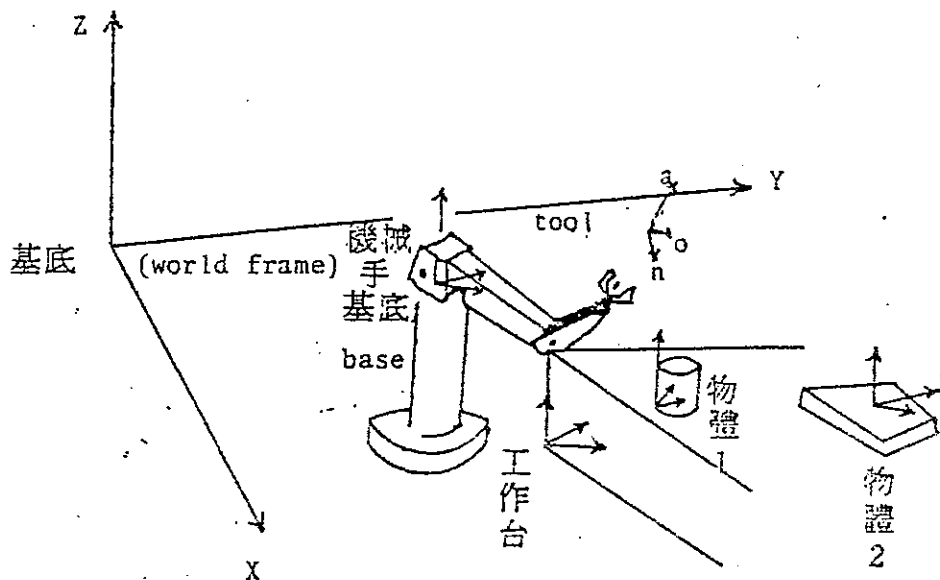
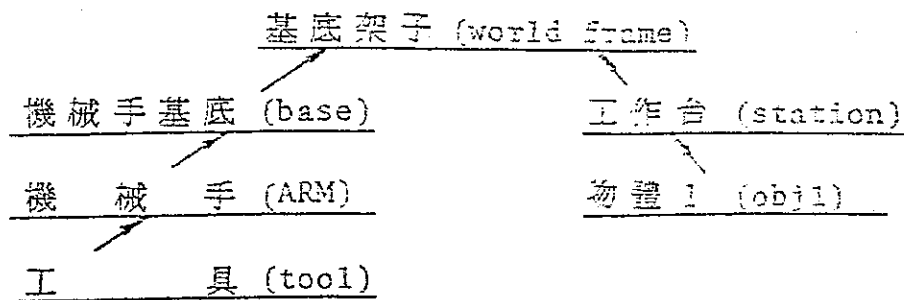


圖 17 外界模型以架子表示的情形

如果，我們要以機械手抓取工作台的物體，可以用下列步驟完成：

- 1 定出物體 1 與工作台的連結關係。
- 2 定出機械手逼近物體 1 的位置與轉向。
- 3 命令機械手經由逼近點到物體 1。
- 4 抓取。

寫為 RLO 程式則為：

```

var  apf1, obj1 : frame ;
      obj2 : frame ;
      t1 : frans ;
  
```

```

begin
    affix obj1 to station ;
    affix apf1 to obj1 at t1 ;
        move ARM-1 to obj1 with approach = apf1 ,
            with speed = 100 ;
    grasp ;
    move obj1 to obj2 directly ;
end.

```

這種模型及指令的優點包括：

1. 如果工作台變動了位置，可以不必修改程式，只須對工作台架子重給新的始值，這增加了程式的可轉移性。
2. 逼近點的選擇是以物體為基準，使程式者方便且精確地定義出其值。
3. 架子 A 與架子 B 因為有了連結 (affix relation) ；因此系統自動對架子作改變，如上圖中將 obj1 移動到 obj2 之後，apf1 與 obj1 關係未變，因此下次要以 apf1 逼近 obj1 時仍然用同樣的指令即可。因此架子管理器中工作包括對 affix 指令及 unfix 指令的處理及對 move 指令後，架子資料的更新。

架子管理器也使得機械手控制系統與 CAD/CAM 系統有一個良好的介面，因為當 CAD 系統完成了設計與測試的工作，可直接將各物體的座標，輸入架子表格內 (frame table) 。

#### B. 手方位與關節轉換器

此轉換器的工作在於空間中一個架子與相對應的軸座標之轉換；它與機械手結構有關 (Robot dependent)。在 RLO 中，它主要作

用有二：1 手方位與關節值間轉換。2 檢查路徑的可行性。分別說明如下：

手方位與關節值間轉換

這種轉換有二個方向：

① 由架子座標 - 6 個軸關節座標

$$f : [ \underline{n}, \underline{o}, \underline{a}, \underline{p} ] \rightarrow [ \theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6 ]$$

② 由 6 個軸關節座標 - 架子座標

$$g : [ \theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6 ] \rightarrow [ \underline{n}, \underline{o}, \underline{a}, \underline{p} ]$$

其中，f 的轉換不一定唯一。

B.2 檢查路徑的可行性：

機械手的機械設計，決定它的工作區間，因此，在每段軌跡的計劃中，都要檢查是否超越它的可達的區間，然後再作轉換工作。

C：運動軌跡：

在 RLO 系統中，我們規劃出直線運動、連續運動及點到點運動 3 種基本的運動軌跡。利用這 3 種基本運動的軌跡，我們可以在下一節中，說明如何組成各種較複雜的運動方式。軌跡的形成乃是依下列步驟達成：

① 在卡式座標的軌跡取樣 (sampling)，轉換成關節值。

② 在關節座標空間中，以內插 (interpolation) 形成軌跡的逼近函數。

③ 依一定的伺服週期，如 1/60sec 或 1/30sec，送出這些函數的值到伺服系統中 (servo systtm)。

RLO 系統使用 X-spline 作關節空間的內插函數 [15]，當我們使用 X-spline 在關節空間作內插時，其步驟如下：

step1：有一組簡單的公式可使用，即當  $[t_i, t_{i+1}]$   $i=0, \dots, n-1$  皆為等間隔時（即  $h=t_{i+1}-t_i$ ），則在時間  $t_i$  點關節之計劃速度  $m_i$  可以下列公式表出：

$$m_i = (q_{i+1} + 4q_i - 5q_{i-1}) / 4h - m_{i-1} / 2 \quad \text{for } i=2, n-1$$

step2：得出  $m_i$  之後，依  $[t_{i-1}, t_i]$  兩端的端點條件：

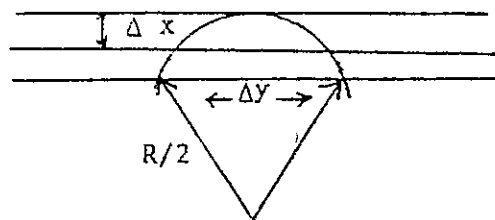
$$q_{i-1}, m_{i-1}, q_i, m_i \quad \text{可唯一決定一 3 次多項式 } S_i。$$

step3：利用  $S_i$  得出  $[t_{i-1}, t_i]$  之間的軌跡函數內插值。

接著，我們分別考慮直線運動、連續運動、點到點運動軌跡的構成。

### C.1 直線運動軌跡：

在此我們考慮關節型手臂，我們注意到，由於手的伸長程度不同，擺動同一角度，會導致不同的圓弧，直線軌跡近似的準確性與這圓弧之曲度有密切關係，假如弧度曲度大，則構成軌跡時在卡氏座標的軌跡取樣就需較密。對直線軌跡，我們要求偏差在容忍的範圍內，因此，依下面公式定出兩點取樣間距離為  $\Delta y$ ，兩點間則在關節上做內插近似。



直線軌跡

- $\Delta x$ ：容許的軌跡誤差
- $R$ ：手臂伸長的長度

圖 18 - 直線軌跡的取樣方式

參考圖18，依三角公式可得：

$$(1/2\Delta y)^2 + (R/2 - 2\Delta x)^2 = (R/2)^2$$

化簡可得

$$\begin{aligned}\Delta y^2 &= 8R\Delta x - 16\Delta x^2 \\ &= 8R\Delta x, \text{ 當 } \Delta x^2 \text{ 足夠小}\end{aligned}$$

由於允許使用者使用不同速度來走直線軌跡，所以我們以下面方法來考慮，當旋轉同一角度，因臂伸長長度的不同，端點的卡氏速度也不同，由機械手設計的規格中，可以找出手臂的最大伸長度  $R_{MAX}$  及最大  $V_{MAX}$ ，然後以  $v_{lmax} = \left(\frac{r_l}{R_{MAX}}\right) * V_{MAX}$  的公式來估測當手臂伸延長度為  $r_l$  時之速度上限  $v_{lmax}$ 。

根據上述決定卡氏軌跡取樣間隔及速度之方式，由 A 點移動到 B 點的直線軌跡即可以下列步驟完成：

step1：以機械手基底 (base) 到線段 AB 的最短距離為  $\frac{R}{2}$ ，算出  $\Delta y$  之值，把 AB 切成 N 段  $\Delta y$  長度的線段。

$$N = \frac{\overline{L(AB)}}{\Delta y}; \quad \overline{L(AB)} \text{ 表 } \overline{AB} \text{ 線段的長度}$$

step2：以機械手要走的線段  $\overline{AB}$  間最短距離： $r_{min}$

$$\text{求出 } v_{min2} = \left(\frac{r_{min}}{R_{MAX}}\right) * V_{MAX} * \alpha$$

並以  $v_{min2}$  表示在線段 AB 間的正常速度。

step3：依使用者所給速度量不同，在指令中 (speed 的量) 求出相對應的速度。

$$V_{spd} = (speed/100) * v_{min2}$$

step4 : 決定了  $V_{spd}$  , 我們便估計走長度為  $l$  的AB 線段中須花費時間。

$$t = l/V_{spd}$$

step5 : 在線段頭尾加減速段間隔取  $1/2\Delta y$  , 而在中間以  $\Delta y$  作為一段的長度 , 然後在每段區間以時間  $t$  用 X-spline 去做內插然後決定每個區間內所須做的伺服次數 , 在區間內依X-spline 內插 , 依伺服週期送出關節值、關節速度、關節加速度。

### C.2 連續運動軌跡 (continuous motion via some intermediate frames)

連續運動是採取 :

- ① 在卡氏座標的二個架子間做線性內插。
- ② 在關節座標內使用 X-spline 為逼近函數 : 如圖 19

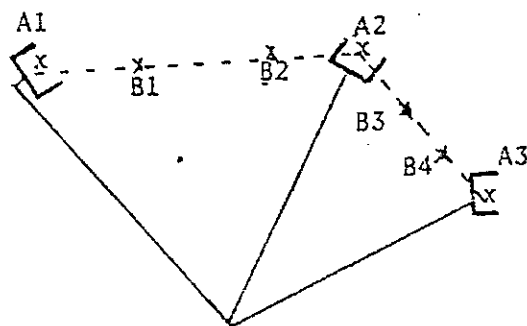


圖 19 機械手連續地經過三點

指令中表明要經過  $A1, A2, A3$  三個架子 , 則我們

- ① 在  $A1A2$  間取出數點 , 譬如  $B1, B2$  。在  $A2A3$  間取出數點 , 譬如  $B3, B4$  。
- ② 以  $A1, B1, B2, A2$  作為起點 , 完成  $A1, B1$  的計算後 , 往右移一位。
- ③ 以  $B1, B2, A2, B3$  為四點 , 完成  $B1, B2$  的軌跡計算。



④繼續②③直到完成。

它的卡氏速度決定，也是以直線軌跡中的速度的決定方式；而由於連續運動軌跡間  $A_1, A_2$  可能距離很大，因此若  $A_1, B_1, B_2, A_2$  之間差值變大，通常爲了不使內插的軌跡發生很大的誤差，對經過每個區間的時間需要加長。

### C.3 點到點運動 (point to point motion)

點對點運動的特徵在其起始及終點的速度與加速度皆爲 0，可以以五次各項式軌跡內插函數表示關節運動，即  $6t^5 - 15t^4 + 10t^3$ ； $0 \leq t \leq 1$  [9]。

### D. 如何完成各種運動形式

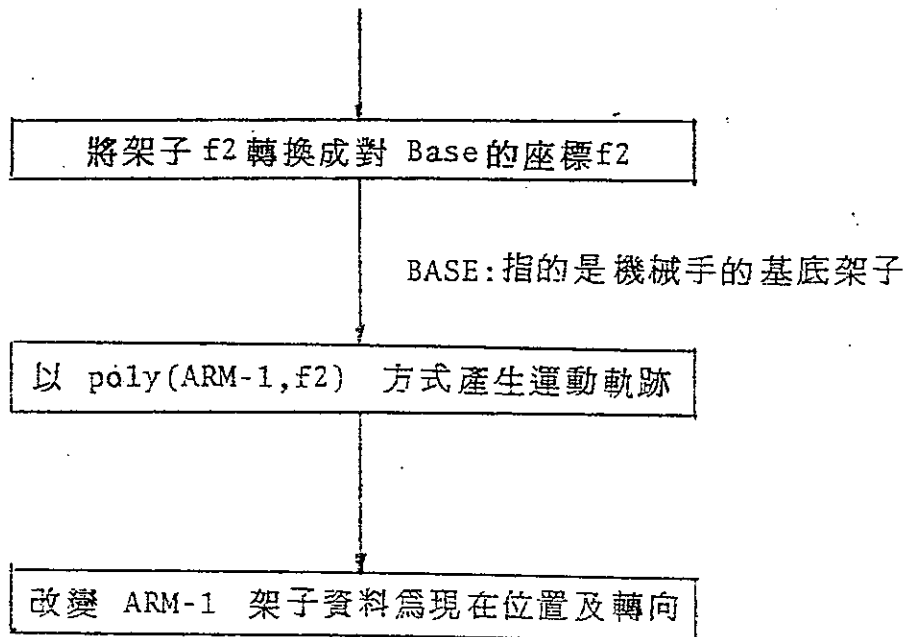
RLO 系統提供的運動指令依上述區分，可以有下例格式：

1. MOVE ARM-1 to f2 directly;
2. MOVE f1 to f2 directly;
3. MOVE ARM-1 to f2 with approach=apf1;
4. MOVE f1 to f2 with approach=apf1;
5. MOVE ARM-1 to f2 with departure=def1;
6. MOVE f1 to f2 with departure=def1;
7. MOVE ARM-1 to f2 with approach=apf1, via (vinf1,...viafn);
8. MOVE f1 to f2 with approach=apf1, via (viaf1,...viafn);
9. MOVE ARM-1 to f2 with departure=def1, via (viaf1,...viafn);
10. MOVE f1 to f2 with departure=def1, via (viaf1,...viafn);
11. MOVE ARM-1 to f2 straightly;
12. MOVE f1 to f2 straightly;

運動形式可以機械手為基點或以物體為基點。而由於不同的運動形式的要求，我們採用上述三種基本運動軌跡來合成它們，同時也要隨時適當地調整架子資料，下面分別說明(1)、(2)、(7)等項運動之構成方法，其餘形式的組成方法則可以予以推知的。

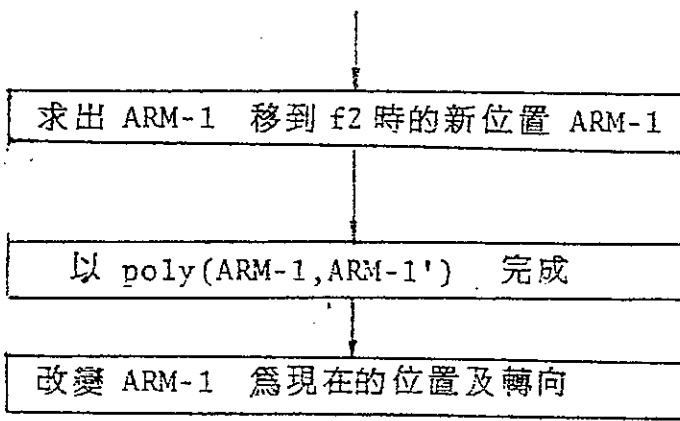
(1) move ARM-1 to f2 directly ;

實現步驟



(2) move f1 to f2 directly ;

實現步驟 :



說明：此類以物體為基底的運動形式中，有 2 點要注意的：

① 在動作開始之間；一定要有 "affix f1 to ARM-1" 的指令，如此才能用上面過程得出正確軌跡。

② 將架子 f1 移到架子 f2 時，ARM-1 的新位置 ARM-1'，依下列公式：

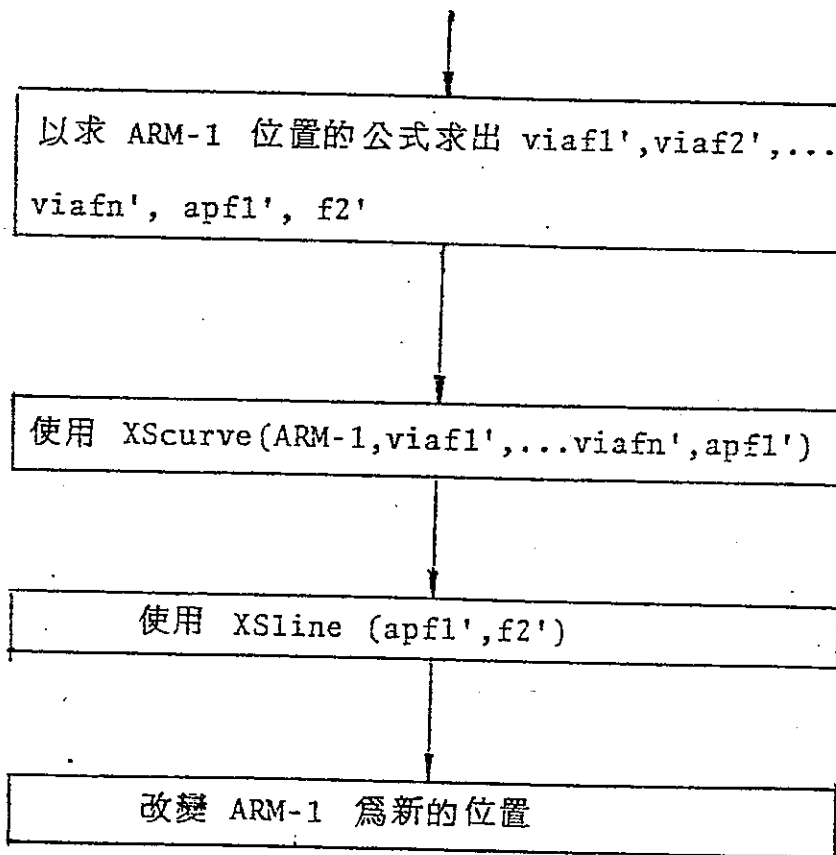
$\text{Base} * \text{ARM-1}' * f1 = f2$ ，其中 \* 表矩陣乘法；

所以，要求出 ARM-1 移動的新位置 ARM-1' 時，須以

$$\text{ARM-1}' = \text{Base}^{-1} * f2 * f1^{-1}$$

(7) move f1 to f2 with approach = apf1,  
via (viaf1', viaf2', ..., viafn') ;

實現方法



上面 XScurve 係利用 X-Spline 選擇  $viaf_1', \dots, viaf_n', apf_1'$  做軌跡形成，XSline 則在  $apf_1', f_2'$  間取足夠點做直線軌跡形成。

#### 4. 結論

由於軌跡演算法的不同，RLO 在路徑計劃上，可動態的接受新計劃出的點。

對於下一點的路徑計劃，只須把點 2、點 3、點 4 往左移一位，把新加入的點放入點 4，再做速度的計算，因此在硬體上以一個環形的佇列 (circle queue) 的結構，來儲存這些點是可行的。而連續軌跡運動中，並未在中間點停留，因此速度可較快，而為了避免碰撞的控制，則可以由經過一些中間點 ( $via p_1, p_2, \dots, p_n$ ) 來完成。其所花的代價為二點間的運動學的解，內插函數以及在關節座標上的 3 次多項函數係數的計算。系統中並提供了局部的座標關係 (經由 with approach, with departure) 使程式者在控制機械手的運動上可以用目標物的座標系統為準，如此則可免除因複雜的幾何關係而導致的程式錯誤。同時，在不同的工作環境下，如 world frame 改變機械手移位，工作台位置改變，或視覺系統移位等，亦不必重新改寫程式，增加了程式，對環境的可移性 (portability) 。

語言方面可以加強的是結構化的控制指令如 if then-else-, while-do 等，同時可增加副程式之使用以提供使用者一些常用的裝配動作。迴路及表列 (loop & array) 的能力也可加強處理重覆性動作的能力。

建立一個即時控制系統是使控制語言更有力的基礎工作，當考慮一個以上的機械手控制系統時，對不同的機械臂需採用不同的程序 (process) 與之對應，程序之間的交通，合作及對系統資源的分享是控制系統老早便存在的問題，如何完美的達成，是須要依應用環境適當調整。對架子表格的共時管理 (concurrent control) 也需考慮設計一個優先權的定序方法，以符合實際狀況需要，是重要而挑戰性的問題。

皮瑞特 (PRATT) [16] 曾經提出一個良好的程式語言所必須具有的 6 項特色，這些準則亦適用於機械手控制語言：

- ① 明確、簡單及單元性的觀念。
- ② 程式結構的明晰性。
- ③ 合用於所使用的領域。
- ④ 容易擴充性。
- ⑤ 易除錯及附帶的軟體支援。
- ⑥ 有效性。

機械手語言的設計除了必須考慮上述六項要求外，還必須考慮未來由於科技進步帶來的變動，如：

- ① 由於複雜的外界模型，使語言處理器及直譯器的工作更重，如何採取一個有效的管理方法。
- ② 在即時控制系統上，因為新加入的元件及控制方法的改變，因而系統必須有模組化的規劃。

綜合以上考慮，導致層次式的控制 (hierarchical control) 以及模組化的規劃 (modularity) 是面對變動環境中必須遵守的原則。

在可預見的未來，因為應用領域的擴大，以及對感應器及系統架構不斷改進的結果，勢必使得機械手控制語言增加更多的特色，進而使得機械手的使用更容易與有效。

## 參 考 資 料

- (1) Denavit, J. and Hartenberg, R. S., "A Kinematic Notation for Lower-pair Mechanisms Based on Matrices," ASME Journal of Applied Mechanics, June 1955, pp. 215-221
- (2) R. C. Paul. "Modelling, Trajectory Calculation and Servoing of a Computer Controlled Arm, " Ph. D. dissertation, Stanford Univ., Stanford, CA, Aug. 1972.
- (3) A. K. Bejczy, "Robot Arm Dynamics and Control," JPL Tech. Memo. 33-669, Feb. 1974.
- (4) D. E. Orin, R. B. McGhee, M. Vukobratovic, and G. Hartoch, "Kinematic and Kinetic Analysis of Openchain Linkages Utilizing Newton - Euler methods," Math. Biosc, Vol. 43, Feb. 1979, pp. 107-130.
- (5) C. S. Lin, "Control and Computation for Mechanical Manipulators," Ph. D. thesis, Purdue University, July 1980.
- (6) J. Y. S. Luh and M. W. Walker, and R. C. Paul, "On-line Computational Scheme for Mechanical Manipulators," J. Dynamic Syst. Meas. and Contr., Trans. ASME, Vol. 102, June 1980, pp. 69-76.
- (7) Luh, J. Y. S. and C. S. Lin, "Optimum Path Planning for Mechanical Manipulators," ASME Trans., J. Dynamic Syst. Meas. and Contr., Vol. 102, No 2, June 1981, pp. 142-151.
- (8) C. S. Lin, P. R. Chang and J. Y. S. Luh, "Formulation and Optimization of Cubic Polynomial Joint Trajectories for Mechanical Manipulators," To appear in IEEE Automatic Control Sept. 1983.

- (9) M. S. Mujtaba, "Discussion of Trajectory Calculation Methods," Ph. D. dissertation, Stanford University, 1980.
- (10) Rosen, C., et al., "Exploratory Research in Advanced Automation," Fifth Report, Stanford Research Institute, January, 1976.
- (11) Taylor, R. H., "Planning and Execution of Straight-Line Manipulator Trajectories," IBM Thomas J. Watson Research Center, RC 6657 (#28633) Research Report, July 1977.
- (12) Behforooz, G. H. et, "A Class of Piecewise-Cubic Interpolatory Polynomials," J. Inst. Maths Applics 25, 1980, pp.53-65.
- (13) Clenshaw, C. W. and Negus, B., "The Cubic X-spline and its Application to interpolation," J. Inst. Maths Applics 22, 1978, pp. 109-119.
- (14) Fahlman, S. E. "A Planning System for Robot Construction Tasks," Artificial Intelligence, Vol.5 No.1 Spring 1974, pp. 1-49.
- (15) Lin, C. S., Chang, P. R. "Joint Trajectories of Mechanical Manipulators for Cartesian Path Approximation," IEEE Trans, on System, Man and Cybernetics, Dec.1983, to appear.
- (16) Pratt, T. W. Programming Languages: Design and Implementation, Englewood Cleffs, N. J. : Prentice-Hall, 1975.



附 錄 一

Stanford 機械手臂間的轉換式

Stanford 機械手，各個機械臂間的幾何轉換參數及轉換式如下：

Parameter Joint	$\alpha$ (deg.)	d (cm)	q (deg.)	a (cm)
1	$-90^\circ$	16.24	$q_1$	0
2	$90^\circ$	6.05	$q_2$	0
3	$0^\circ$	$q_3$	$-90^\circ$	0
4	$-90^\circ$	0	$q_4$	0
5	$90^\circ$	0	$q_5$	0
6	$0^\circ$	10.35	$q_6$	0

表 A-1 機械臂間座標轉換參數位

轉換式：式中  $c_i$  表示  $\cos(q_i)$ ， $s_i$  表示  $\sin(q_i)$

$$T_0^1 = \begin{bmatrix} c_1 & 0 & -s_1 & 0 \\ s_1 & 0 & c_1 & 0 \\ 0 & -1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_1^2 = \begin{bmatrix} c_2 & 0 & s_2 & 0 \\ s_2 & 0 & -c_2 & 0 \\ 0 & 1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_2^3 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & q_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_3^4 = \begin{bmatrix} c_4 & 0 & -s_4 & 0 \\ s_4 & 0 & c_4 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_4^5 = \begin{bmatrix} c_5 & 0 & s_5 & 0 \\ s_5 & 0 & -c_5 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_5^6 = \begin{bmatrix} c_6 & -s_6 & 0 & 0 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

## 附 錄 二

### Stanford 型的機械手有關的參數值

本附錄所列的參數值來源取自參考文獻[3]。以下的慣性張量 (inertia tensors)、質量中心及質量等幾何參數，都以每一個機械臂本身所定的座標系為基準座標而言。相關的單位為MKS制。

幾何參數與物理參數：

(1) 慣性張量矩陣  $I_{ip}$  : 以  $i$  座標系為原點所測量之值 ( $\text{kg}\cdot\text{m}^2$ )。

$$\begin{array}{l}
 I_{1p} = \begin{pmatrix} 0.276 & 0 & 0 \\ 0 & 0.255 & 0 \\ 0 & 0 & 0.071 \end{pmatrix} \quad I_{2p} = \begin{pmatrix} 0.108 & 0 & 0 \\ 0 & 0.018 & 0 \\ 0 & 0 & 0.1 \end{pmatrix} \\
 I_{3p} = \begin{pmatrix} 2.510 & 0 & 0 \\ 0 & 2.510 & 0 \\ 0 & 0 & 0.006 \end{pmatrix} \quad I_{4p} = \begin{pmatrix} 0.002 & 0 & 0 \\ 0 & 0.001 & 0 \\ 0 & 0 & 0.001 \end{pmatrix} \\
 I_{5p} = \begin{pmatrix} 0.003 & 0 & 0 \\ 0 & 0.003 & 0 \\ 0 & 0 & 0.0004 \end{pmatrix} \quad I_{6p} = \begin{pmatrix} 0.005 & 0 & 0 \\ 0 & 0.005 & 0 \\ 0 & 0 & 0.0003 \end{pmatrix}
 \end{array}$$

(2) 質量中心(m) :  $S_i^* = (\bar{x}_i \quad \bar{y}_i \quad \bar{z}_i)^T$

$$\begin{array}{l}
 S_1^* = \begin{pmatrix} 0.0 \\ 0.0175 \\ -0.1105 \end{pmatrix} \\
 S_2^* = \begin{pmatrix} 0.0 \\ -0.1054 \\ -0.0079 \end{pmatrix} \\
 S_3^* = \begin{pmatrix} 0.0 \\ 0.0 \\ -0.6447 \end{pmatrix} \\
 S_4^* = \begin{pmatrix} 0.0 \\ 0.0092 \\ -0.0054 \end{pmatrix} \\
 S_5^* = \begin{pmatrix} 0.0 \\ 0.0003 \\ 0.0566 \end{pmatrix} \\
 S_6^* = \begin{pmatrix} 0.0 \\ 0.0014 \\ -0.0922 \end{pmatrix}
 \end{array}$$

(3) 每組機械臂的重量 (kg) :

$$\begin{array}{lll}
 m_1 = 9.29 & m_2 = 5.505 & m_3 = 4.25 \\
 m_4 = 1.08 & m_5 = 1.08 & m_6 = 0.51
 \end{array}$$

(4) 負載：最大的負載是設為 1.8 kg 重，422 cm<sup>3</sup> 的立方體，質  
重中心置於手座標系的原點，如此第六個機械臂的等效

參數值為：

$${}^L I_{6p} = \begin{pmatrix} 0.006 & 0 & 0 \\ 0 & 0.006 & 0 \\ 0 & 0 & 0.002 \end{pmatrix}$$

$$S_6^* = \begin{pmatrix} 0.0 \\ 0.0014 \\ -0.0203 \end{pmatrix}$$

$$m_6^L = 2.31$$

(5) 馬達轉軸的慣量轉換到負載端的值 (kg-m<sup>2</sup>):

$$J_{m1} = 0.953 \quad J_{m2} = 2.193 \quad J_{m3} = 0.782(\text{kg})$$

$$J_{m4} = 0.106 \quad J_{m5} = 0.097 \quad J_{m6} = 0.02$$

(6) 馬達裝設的最大驅動轉換到負載端的值 (N-m) 設為:

$$T_{m1} = 91.8 \quad T_{m2} = 303 \quad T_{m3} = 198(\text{N})$$

$$T_{m4} = 10.15 \quad T_{m5} = 10.15 \quad T_{m6} = 7.15$$

(7) 各個關節機構的限制如下表:

關節	1	2	3 (cm)	4	5	6
上限	175°	175°	111.76	175°	110°	175°
下限	-175°	-175°	14.	-175°	-110°	175°

表 A-2 關節的機構限制

### 附錄三

#### Bejczy 模式轉矩計算式子

Bejczy [3] 針對 JPL 機械手，將完整的 L-E 動力學公式，簡化成如下的轉矩計算式子：

$$\tau_i = D_{ii}(q) \ddot{q}_i + G_i(q); \quad i = 1, 2, \dots, 6$$

式中慣量  $D_{ii}$  如下：

$$D_{11} = b_1 + (b_2 + b_3 + b_5^L c_5) q_3 + b_4^L q_3^2$$

$$D_{22} = b_{13} + (b_3 + b_5^L c_5) q_3 + b_4^L q_3^2$$

$$D_{33} = b_4^L$$

$$D_{44} = b_{14} + b_{15}^L s_5^2$$

$$D_{55} = b_{17} + b_{18}^L$$

$$D_{66} = b_{19} + b_{20}^L$$

而重力負載項  $G_i$  如下：

$$G_1 = 0$$

$$G_2 = g(k_1 + d_2^L q_3) s_2 + g k_3^L (s_2 c_5 + c_2 s_4 s_5)$$

$$G_3 = -g k_2^L c_2$$

$$G_4 = g k_3^L s_2 c_4 s_5$$

$$G_5 = g k_3^L (s_2 s_4 c_5 + c_2 s_5)$$

$$G_6 = 0.$$

式中的係數值為：

$$b_1 = 1.423 \text{ (kg} \cdot \text{m}^2)$$

$$b_2 = 2.51 \text{ (kg} \cdot \text{m}^2)$$

$$b_3 = -5.49 \text{ (kg} \cdot \text{m)}$$

$$b_4^L = m_3 + m_4 + m_5 + m_6 + J_{m_3} = 7.257 \text{ (kg)}$$

$$b_5^L = 2m_5 \bar{z}_5 + 2m_6 (\bar{z}_6 + d_6) = 0.232 \text{ (kg} \cdot \text{m)}$$

$$b_{13} = 4.72 \text{ (kg} \cdot \text{m}^2)$$

$$b_{14} = 0.107 \text{ (kg} \cdot \text{m}^2)$$

$$b_{15}^L = m_5 k_{511}^2 + m_6 k_{611}^2 + m_6 d_6 (d_6 + 2\bar{z}_6) = 0.015 \text{ (kg} \cdot \text{m}^2)$$

$$b_{17} = 0.099 \text{ (kg} \cdot \text{m}^2)$$

$$b_{18}^L = m_6 k_{611}^2 + m_6 d_6 (d_6 + 2\bar{z}_6) = 0.013 \text{ (kg} \cdot \text{m}^2)$$

$$b_{19} = 0.02 \text{ (kg} \cdot \text{m}^2)$$

$$b_{20}^L = m_6 k_{633}^2 = 0.013 \text{ (kg} \cdot \text{m}^2)$$

$$k_1 = -2.788 \text{ (kg} \cdot \text{m)}$$

$$k_2^L = b_4^2 = 7.257 \text{ (kg)}$$

$$k_3^L = 0.05 b_5^L = 0.116 \text{ (kg} \cdot \text{m)}$$

$$g = 9.80621 \text{ (m/sec}^2)$$

其中  $c_i$  表示  $\cos(q_i)$ ， $s_i$  表示  $\sin(q_i)$ ，而有“ $q_i$ ”註標的係數是表示與手端所取的負載有關，此處的係數值是沒有考慮負載情況所得到的。

附 錄 四

```

PARAMETER N=6
COMMON /ARDATA/AFL(6), RS(6), PJOT(6), AS(6),
*          RMASS(6), RCMX(6), RCMY(6), RCMZ(6), RKX(6), RKY(6),
*          RKZ(6), RMOTJ(6), LINK(6), UP(6), RLOW(6), GRA
DIMENSION Q(6), QD(6), QDD(6), Z(6)
DATA AFL/-90. 0, 90. 0, 0. 0, -90. 0, 90. 0, 0. 0/
*,      RS/0. 3556, 0. 1619, 0. 0. 0, 0. 0, 0. 2477/
*,      PJOT/0. 0, 0. 0, -90. 0, 0. 0, 0. 0, 0/
*,      AS/6*0. 0/
*,      RMASS/9. 29, 5. 505, 4. 25, 1. 08, 0. 63, 0. 51/
*,      RCMX/6*0. 0/
*,      RCMY/0. 0175, -0. 1054, 0. 0, 0. 0092, 0. 0003, 0. 0014/
*,      RCMZ/-0. 1105, -0. 0079, -0. 6447, -0. 0054, 0. 0566, -0. 0922/
DATA RKX/0. 276, 0. 108, 2. 51, 0. 002, 0. 003, 0. 005/
*,      RKY/0. 255, 0. 018, 2. 510, 0. 001, 0. 003, 0. 005/
*,      RKZ/0. 071, 0. 1, 0. 006, 0. 001, 0. 0004, 0. 0003/
*,      RMOTJ/6*0. 0/
*,      LINK/0, 0, 1, 0, 0, 0/
*,      UP/6*0. 0/
*,      RLOW/6*0. 0/
*,      GRA/9. 80621/
DO 110 I=1, N
  XX=3. 141592654/180. 0
  AFL(I)=AFL(I)*XX
  PJOT(I)=PJOT(I)*XX
110  CONTINUE
  TYPE 200
200  FORMAT(1X, 'ENTER: Q(1) TO Q(6)')
  ACCEPT 400, Q
  TYPE 220
220  FORMAT(1X, 'ENTER: QD(1) TO QD(6)')
  ACCEPT 400, QD
  TYPE 240
240  FORMAT(1X, 'ENTER: QDD(1) TO QDD(6)')
  ACCEPT 400, QDD
400  FORMAT(6F10. 5)
WRITE(3, 475)
475  FORMAT(1X, 'INPUT Q, QD, QDD ARE :')
  WRITE(3, 410)Q
  WRITE(3, 410)QD
  WRITE(3, 410)QDD
  WRITE(3, 485)
485  FORMAT(1X, '*****')
  CALL CLAGRA(N, Q, QD, QDD, Z)
  WRITE(3, 410) Z
410  FORMAT(6F10. 5)
  STOP
  END

```



```

C*****
C
C SUBROUTINE GLAGRA
C PURPOSE : GENERATED FORCE BY LAGRANGE FORMULATION; GENERAL FORM
C CALL GLAGRA (NN, Q, DQ, DDQ, F)
C
C NN ==> LINK NUMBER
C Q ==> INPUT 6X1 JOINT POSITIONS
C DQ ==> INPUT 6X1 JOINT VELOCITIES
C DDQ ==> INPUT 6X1 JOINT ACCELERATIONS
C F <== OUTPUT 6X1 TORQUES
C*****
C
C SUBROUTINE GLAGRA (NN, Q, DQ, DDQ, F)
C
C COMMON /ARDATA/AFL(6), RS(6), PJOT(6), AS(6),
C * RMASS(6), RCMX(6), RCMY(6), RCMZ(6), RKX(6), RKY(6),
C * RKZ(6), RMOTJ(6), LINK(6), UP(6), RLOW(6), GRA
C COMMON/GLAG/N, TA(4, 4, 6), H(4, 4, 6), U(4, 4, 6, 6),
C * CU(4, 4, 6, 6, 6)
C DIMENSION Q(1), DQ(1), DDQ(1), G(6), COR(6), F(6), RJ(6, 6)
C *, ACCF(6)
C N=NN
C PI=3.141592654
C
C**** CHECK THE BRAKE OF JOINTS
C**** IF THE BRAKE IS SET ON THEN JOINT VELOCITY TO BE SET ZERO
C
C DO 10 I=1, N
C IF (BRAKE(I).EQ.1) DQ(I)=0.0
C10 CONTINUE
C
C**** CALCULATE THE N-TH TRANSFORMATION MARTIX ; AIJ
C
C CALL CALAI(Q)
C TYPE 111, N
111 FORMAT(1X, 'N=: ', I)
C DO 113 K=1, N
C TYPE 114, K
114 FORMAT(1X, 'K=: ', I)
C TYPE 112, ((TA(I, J, K), J=1, 4), I=1, 4)
112 FORMAT(1X, 4(4F10.5/))
113 CONTINUE
C
C-----
C**** CALCULATE PSEUDO INERTIA MARTIX ; JI
C
C CALL CALHI
C
C DO 300 K=1, 6
C TYPE 301, K
301 FORMAT(1X, 'H=: ', I)
C TYPE 511, ((H(I, J, K), J=1, 4), I=1, 4)
511 FORMAT(1X, 4(4F10.5/))
300 CONTINUE

```

```

C*** CALCULATE ;UIIJ
C
CALL CALUIJ
C
C*** CALCULATE ;UIJK
C
CALL CAUIJK
C
C*** CALCULATE REACTION FORCE INERTIA CONSTANT ;CIK
C
CALL CALCIK(RJ)
TYPE 211, ((RJ(I, J), J=1, 6), I=1, 6)
211 FORMAT(1X, 6(6F10.5))
C
C*** ADD MOTOR SHAFT INTERTIAS
C
DO 20 I=1, N
RJ(I, I)=RJ(I, I)+RMOTJ(I)
20 CONTINUE
C*** TO CALCULATE ACCELE. FORCES
DO 30 I=1, N
BUFF=0.0
DO 40 J=1, N
BUFF=RJ(I, J)*ODQ(J)+BUFF
40 CONTINUE
ACCF(I)=BUFF
30 CONTINUE
C
C*** CALCULATE CORIOLIS AND CENTRIFUGAL FORCES
C
CALL CALCOR(DQ, COR)
C
TYPE 311, COR
311 FORMAT(1X, 'COR=: ', 6F10.5)
C*** CALCULATE GRAVITY FORCES
C
CALL CALCI(G)
TYPE 322, G
322 FORMAT(1X, 'G=: ', 6F10.5)
TYPE 333, ACCF
333 FORMAT(1X, 'ACCF=: ', 6F10.5)
DO 50 I=1, N
F(I)=G(I)+COR(I)+ACCF(I)
50 CONTINUE
TYPE 444, F
444 FORMAT(1X, 'F=: ', 6F10.5)
RETURN
END
C*****
C
C SUBROUTINE CALAI
C
C PURPOSE : TO CALCULATE THE TRANSFORMATION MATRIX AIJ
C
C CALL CALAI(G)
C

```

```

C
SUBROUTINE CALAI(Q)
C
COMMON /ARDATA/AFL(6), RS(6), PJOT(6), AS(6),
*          RMASS(6), RCMX(6), RCMY(6), RCMZ(6), RKX(6), RKY(6),
*          RKZ(6), RMOTJ(6), LINK(6), UP(6), RLOW(6), GRA
COMMON/GLAG/N, TA(4, 4, 6), H(4, 4, 6), U(4, 4, 6, 6),
*          CU(4, 4, 6, 6, 6)
DIMENSION Q(1)
DO 10 I=1, N
  IF(LINK(I).EQ.1) GOTO 20
  B=COS(Q(I))
  C=SIN(Q(I))
  GOTO 25
20  CONTINUE
  B=COS(PJOT(I))
  C=SIN(PJOT(I))
  RS(I)=Q(I)
-25 CONTINUE
  D=COS(AFL(I))
  E=SIN(AFL(I))
  TA(1, 1, I)=B
  TA(2, 1, I)=C
  TA(3, 1, I)=0.
  TA(4, 1, I)=0.
  TA(1, 2, I)=-D*C
  TA(2, 2, I)=B*0
  TA(3, 2, I)=E
  TA(4, 2, I)=0.
  TA(1, 3, I)=E*C
  TA(2, 3, I)=-E*B
  TA(3, 3, I)=0
  TA(4, 3, I)=0.
  TA(1, 4, I)=AS(I)*B
  TA(2, 4, I)=AS(I)*C
  TA(3, 4, I)=RS(I)
  TA(4, 4, I)=1
10  CONTINUE
RETURN
END

```

```

C *****
C
C SUBROUTINE CALHI
C
C PURPOSE : TO CALCULATE PSEUDO INERTIA MATRIX JI
C
C CALL CALHI
C *****
C
C SUBROUTINE CALHI
C
C COMMON /ARDATA/AFL(6), RS(6), PJOT(6), AS(6),
*          RMASS(6), RCMX(6), RCMY(6), RCMZ(6), RKX(6), RKY(6),
*          RKZ(6), RMOTJ(6), LINK(6), UP(6), RLOW(6), GRA
C COMMON/GLAG/N, TA(4, 4, 6), H(4, 4, 6), U(4, 4, 6, 6),
*          CU(4, 4, 6, 6, 6)
C DO 10 I=1, N
C   CALL NULM3 (1, 1, I, H)
C   H(1, 1, I)=0. 5*(-RKX(I)+RKY(I)+RKZ(I))
C   H(4, 1, I)=RMASS(I)*RCMX(I)
C   H(1, 4, I)=H(4, 1, I)
C   H(2, 2, I)=0. 5*(RKX(I)-RKY(I)+RKZ(I))
C   H(4, 2, I)=RMASS(I)*RCMY(I)
C   H(2, 4, I)=H(4, 2, I)
C   H(3, 3, I)=0. 5*(RKX(I)+RKY(I)-RKZ(I))
C   H(4, 3, I)=RMASS(I)*RCMZ(I)
C   H(3, 4, I)=H(4, 3, I)
C   H(4, 4, I)=RMASS(I)
10 CONTINUE
RETURN
END
C *****
C
C SUBROUTINE CALTIJ
C
C PURPOSE : TO CALCULATE TRANSFORMATION MATRIX :
C           TIJ=AI*... *AJ+1
C CALL CALTIJ(I, J, TM)
C *****
C
C SUBROUTINE CALTIJ(I, J, TM)
C
C COMMON/GLAG/N, TA(4, 4, 6), H(4, 4, 6), U(4, 4, 6, 6),
*          CU(4, 4, 6, 6, 6)
C DIMENSION TM(4, 4), TB(4, 4)
C CALL IDENTM(TB)
C IF (J. LE. I) GOTO 10
C   I=I+1
C   DO 20 K=L, J
C     CALL MULT3(1, 1, 1, TB, 1, 1, K, TA, 1, 1, 1, TM)
C     CALL MCOPY(1, 1, 1, TM, TB)
20 CONTINUE
GOTO 15
10 CONTINUE
CALL MCOPY (1, 1, 1, TB, TM)
15 CONTINUE
RETURN
END

```

```

C*****
C
C   SUBROUTINE CALUIJ
C
C   PURPOSE : TO CALCULATE UIJ MATRIX
C
C   CALL CALUIJC
C
C*****
C
C   SUBROUTINE CALUIJ
C
COMMON /ARDATA/AFL(6), RS(6), PJOT(6), AS(6),
*       RMASS(6), RCMX(6), RCMY(6), RCMZ(6), RKX(6), RKY(6),
*       RKZ(6), RMOYJ(6), LINK(6), UP(6), RLOW(6), GRA
COMMON/CLAG/N, TA(4, 4, 6), H(4, 4, 6), U(4, 4, 6, 6),
*       CU(4, 4, 6, 6, 6)
DIMENSION T1(4, 4), T2(4, 4)
DO 10 I=1, N
  DO 20 J=1, N
    IF (I GE J) GOTO 30
    CALL NULM3(1, I, J, U)
    GOTO 35
30    CONTINUE
    K=0
    CALL CALTIJ(K, J-1, T1)
    CALL MULTQK(LINK(J), T1, T2)
    CALL CALTIJ(J-1, I, T1)
    CALL MULT3(1, 1, 1, T2, 1, 1, 1, T1, 1, I, J, U)
35    CONTINUE
20    CONTINUE
10    CONTINUE
    RETURN
    END

```

```

C*****
C
C   SUBROUTINE CAUIJK
C
C   PURPOSE TO CALCULATE UIJK MATRIX
C   CALL CAUIJK
C*****
C
C   SUBROUTINE CAUIJK
C
C   COMMON /ARDATA/AFL(6), RS(6), PJOT(6), AS(6),
*       RMASS(6), RCMX(6), RCMY(6), RCMZ(6), RKX(6), RKY(6),
*       RKZ(6), RMOTJ(6), LINK(6), UP(6), RLOW(6), GRA
COMMON/GLAG/N, TA(4, 4, 6), H(4, 4, 6), U(4, 4, 6, 6),
*       CU(4, 4, 6, 6)
DIMENSION T1(4, 4), T2(4, 4), T3(4, 4)
DO 10 I=1, N
  DO 20 J=1, N
    DO 30 K=1, N
      IF((I.GE.J).AND.(I.GE.K)) GOTO 40
      CALL NULM3(I, J, K, CU)
      GOTO 45
40     CONTINUE
      IF (K.GE.J) GOTO 50
      M=K
      L=J
      GOTO 55
50     CONTINUE
      M=J
      L=K
55     CONTINUE
      KK=0
      CALL CALTIJ(KK, M-1, T1)
      CALL MULTQK(LINK(M), T1, T2)
      CALL CALTIJ(M-1, L-1, T3)
      CALL MULT(T2, T3, T1)
      CALL MULTQK(LINK(L), T1, T3)
      CALL CALTIJ(L-1, L, T2)
      CALL MULT3(L, L, L, T3, 1, L, L, T2, I, J, K, CU)
45     CONTINUE
30     CONTINUE
20     CONTINUE
10     CONTINUE
RETURN
END

```

```

C*****
C
C      SUBROUTINE CALCIK
C
C      SUBROUTINE CALCIK
C
C      PURPOSE : TO CALCULATE REACTION FORCE CONSTANT CIK TERM
C
C      CALL CALCIK (RF)
C
C*****
C
C      SUBROUTINE CALCIK(RF)
C
C      COMMON/GLAG/N, TA(4, 4, 6), H(4, 4, 6), U(4, 4, 6, 6),
*      CU(4, 4, 6, 6)
-----
C      DIMENSION T1(4, 4), T2(4, 4), T3(4, 4), RF(1, 1)
C      DO 10 I=1, N
C        DO 20 K=1, N
C          L=MAXIJ(I, K)
C          D=0.
C          DO 30 J=L, N
C            CALL MULT3(I, J, K, U, 1, 1, J, H, 1, 1, 1, T1)
C            CALL MCOPY(1, J, I, U, T2)
C            CALL TRANS(T2, T3)
C            CALL MULT(T1, T3, T2)
C            B=TRACE(T2)
C            D=D+B
C          30 CONTINUE
C        RF(I, K)=D
C        RF(K, I)=D
C      20 CONTINUE
C    10 CONTINUE
C      RETURN
C      END

```

```

C*****
C
C   SUBROUTINE CALCOR(DQ, COR)
C
C   PURPOSE : TO CALCULATE CORIOLIS AND CENTRIFUGAL FORCES
C
C   CALL CALCOR(DQ, COR)
C*****
C
C   SUBROUTINE CALCOR(DQ, COR)
C
C   COMMON/GLAG/N, TA(4, 4, 6), H(4, 4, 6), U(4, 4, 6, 6),
*   CU(4, 4, 6, 6)
C   DIMENSION COR(1), DQ(1), T1(4, 4), T2(4, 4), T3(4, 4)
C   DO 10 I=1, N
C     B=0.
C     DO 20 K=1, N
C       C=0.
C       LI=MAXIJ(I, K)
C       DO 30 M=L, N
C         D=0.
C         L2=MAXIJ(LI, M)
C         DO 40 J=L2, N
C           CALL MULT3(J, K, M, CU, 1, 1, J, H, 1, 1, 1, T1)
C           CALL MCOPY(1, J, I, U, T2)
C           CALL TRANS(T2, T3)
C           CALL MULT(T1, T3, T2)
C           E=TRACE(T2)
C           D=D+E
C40        CONTINUE
C           E=D*DQ(M)
C           C=C+E
C30        CONTINUE
C           B=B+C*DQ(K)
C20        CONTINUE
C           COR(I)=B
C10        CONTINUE
C   RETURN
C   END

```



```

C*****
C
C   SUBROUTINE CALCI
C
C   PURPOSE : TO CALCULATE GRAVITY FORCE CI
C
C   CALL CALCI(G)
C*****
C
C   SUBROUTINE CALCI(G)
C
C   COMMON /ARDATA/AFL(6), RS(6), PJOT(6), AS(6),
C   *       RMASS(6), RCMX(6), RCMY(6), RCMZ(6), RKX(6), RKY(6),
C   *       RKZ(6), RMOTJ(6), LINK(6), UP(6), RLOW(6), GRA
C   COMMON/GLAG/N, TA(4, 4, 6), H(4, 4, 6), U(4, 4, 6, 6),
C   *       CU(4, 4, 6, 6, 6)
C   DIMENSION G(1), T(4, 4)
C   DO 10 I=1, N
C     B=0.
C     DO 20 J=1, N
C       CALL MCOPY(1, J, I, U, T)
C       BUFF=T(3, 1)*RCMX(J)+T(3, 2)*RCMY(J)+T(3, 3)*RCMZ(J)+T(3, 4)
C       C=GRA*RMASS(J)*BUFF
C       B=B+C
C   20   CONTINUE
C       G(I)=-B
C   10   CONTINUE
C       RETURN
C       END
C*****
C
C   SUBROUTINE MULT
C
C   PURPOSE : MULTIPLY TWO MARTIXS
C
C   CALL MULT (A, B, C)
C*****
C
C   SUBROUTINE MULT (A, B, C)
C
C   DIMENSION A(4, 4), B(4, 4), C(4, 4), D(4, 4)
C   CALL TRANS(A, D)
C   DO 10 I=1, 4
C     DO 20 J=1, 4
C       C(J, I)=D(1, J)*B(1, I)+D(2, J)*B(2, I)+D(3, J)*B(3, I)+
C   *       D(4, J)*B(4, I)
C   20   CONTINUE
C   10   CONTINUE
C       RETURN
C       END

```

```

C *****
C
C SUBROUTINE MCOPY
C
C PURPOSE : COPY A MARTIX
C
C CALL MCOPY(I, J, K, TI, TR)
C
C I ==> NUMBER OF THIRD LEVEL
C J ==> NUMBER OF SECOND LEVEL
C K ==> NUMBER OF FIRST LEVEL
C TI ==> NAME OF INPUT 4X4 MARTIX
C TR <== NAME OF OUTPUT 4X4 MARTIX
C *****
C
C SUBROUTINE MCOPY(I, J, K, TI, TR)
C
C DIMENSION TI(4, 4, 6, 6, 6), TR(4, 4)
C DO 20 I1=1, 4
C DO 30 J1=1, 4
C TR(J1, I1)=TI(J1, I1, K, J, I)
20 CONTINUE
20 CONTINUE
RETURN
END
C *****
C
C SUBROUTINE MULT3
C PURPOSE : MULTPY TWO MARTIXS
C CALL SUBROUTINE (I, J, K, A, I1, J1, K1, B, I2, J2, K2, C)
C
C A ==> FIRST 4X4 INPUT MARTIX
C B ==> SECOND 4X4 INPUT MARTIX
C C <== OUT 4X4 MARTIX
C *****
C
C SUBROUTINE MULT3 (I, J, K, A, I1, J1, K1, B, I2, J2, K2, C)
C
C DIMENSION A(4, 4, 6, 6, 6), B(4, 4, 6, 6, 6), C(4, 4, 6, 6, 6)
C DO 10 II=1, 4
C DO 20 JJ=1, 4
C BUFF=0.0
C DO 30 KK=1, 4
C BUFF=A(JJ, KK, K, J, I)*B(KK, II, K1, J1, I1)+BUFF
30 CONTINUE
C(C(JJ, II, K2, J2, I2)=BUFF
20 CONTINUE
10 CONTINUE
RETURN
END

```

```

C *****
C
C SUBROUTINE NULM3
C PURPOSE : GIVEN A ZERO MARTIX WITH THREE LAYERS
C CALL NULM3 (I, J, K, T)
C
C I ==> THIRD LAYER
C J ==> SECOND LAYER
C K ==> FIRST LAYER
C T ==> OUTPUT 4X4 ZERO MARTIX T(K, J, I)
C *****
C
C SUBROUTINE NULM3(I, J, K, T)
C
C DIMENSION T(4, 4, 6, 6, 6)
C DO 10 KK=1, 4
C   DO 20 LL=1, 4
C     T(LL, KK, K, J, I)=0.0
20 CONTINUE
10 CONTINUE
RETURN
END
C *****
C
C SUBROUTINE IDENTM(T)
C PURPOSE : GIVEN A IDENTITY 4X4 MARTIX
C CALL IDENTM(T)
C
C T <== OUTPUT 4X4 IDENTITY MARTIX
C *****
C
C SUBROUTINE IDENTM(T)
C
C DIMENSION T(4, 4)
C CALL NULM3(1, 1, 1, T)
C DO 10 I=1, 4
C   T(I, I)=1.
10 CONTINUE
RETURN
END

```

```

C *****
C
C SUBROUTINE TRANS
C PURPOSE : TO TRANSPOSE A MARTIX
C CALL TARNM (T, TR)
C
C C ==> NAME OF A 4X4 TO BE TRANSPOSE
C TR <== RESULTED MARTIX
C *****
C
C SUBROUTINE TRANS (T, TR)
C
C DIMENSION T(4, 4), TR(4, 4)
C DO 10 I=1, 4
C   DO 20 J=1, 4
C     TR(I, J)=T(J, I)
20 CONTINUE
10 CONTINUE
RETURN
END
C *****
C
C SUBROUTINE MULTQK
C PURPOSE : MULIPY QK( MARTIX DEPEDENT EITHER ROTATE OR PRISMATIC
C CALL MULTQK(LINK, TI, TR)
C
C LINK ==> 1 : THE JOINT IS PRISMATIC
C        0 : THE LINK IS ROTATE
C TR <== RESULTED 4X4 MARTIX ; TI=TI*QK
C TI ==> INPUT 4X4 MARTIX
C *****
C
C SUBROUTINE MULTQK (LINK, TI, TR)
C
C DIMENSION TI(4, 4), TR(4, 4)
C CALL NULM3(1, 1, 1, TR)
C IF (LINK.EQ. 1) GOTO 10
C   DO 20 J=1, 4
C     TR(J, 1)=TI(J, 2)
C     TR(J, 2)=-TI(J, 1)
20 CONTINUE
C   GOTO 15
10 CONTINUE
C   DO 30 J=1, 4
C     TR(J, 4)=TI(J, 3)
30 CONTINUE
15 CONTINUE
RETURN
END

```

```

C*****
C
C   FUNCTION MAXIJ
C   PURPOSE : TO GET MAX. NUMBER OPERATION
C   MAXIJ (I,C)
C
C*****
C   FUNCTION MAXIJ(I,J)
C
C   IF(I.GE.J) GOTO 10
C       MAXIJ=J
C       GOTO 15
10  CONTINUE
C       MAXIJ=I
15  CONTINUE
C   RETURN
C   END
C*****
C
C   FUNCTION TRACE
C   PURPOSE : TO DO MATRIX TRACE OPERATION
C   TRACE (T)
C
C   T ==> NAME OF INPUT 4X4 MARTIX
C
C*****
C
C   FUNCTION TRACE (T)
C
C   DIMENSION T(4,4)
C   F=0.
C   DO 10 I=1,4
C       F=F+T(I,I)
10  CONTINUE
C   TRACE=F
C   RETURN
C   END

```

附 錄 五

```

0001          SUBROUTINE TOACC(TOU,Q,QD,QDD)

C          THIS SUBROUTINE SOLVES FOR THE JOINT ACCELERATIONS
C          GIVEN THE INPUT GENERALIZED FORCES, THE JOINT VALUES
C          AND THE JOINT VELOCITIES.

C          THE FOLLOWING VARIABLES ARE USED.
C          TOU(I) = THE I-TH JOINT GENERALIZED FORCE.
C          Q(I) = THE I-TH JOINT VALUE.
C          QD(I) = THE VELOCITY OF THE I-TH JOINT VALUE.
C          QDD(I) = THE ACCELERATION OF THE I-TH JOINT VALUE.
C          QZ(.) = AN ARRAY OF ZERO VALUES.
C          GRAVY = THE ACCELERATION DUE TO GRAVITY.
C          AV(.) = A DUMMY ARRAY.
C          A(...)= THE JOINT ACCELERATION COEF. MATRIX.
C          GRAVY = THE ACCELERATION DUE TO GRAVITY.

0002          DIMENSION TOU(6),Q(6),QD(6),QDD(6),QZ(6),AV(6),A(6,6)
0003          DATA QZ/ 0.0,0.0,0.0,0.0,0.0,0.0/

C          SOLVE FOR THE JOINT ACCELERATION COEF. MATRIX

0004          GRAVY=0.0
0005          DO 10 I=1,6
0006          DO 5 J=1,6
0007          QDD(J)=0.0
0008          QDD(I)=1.0
0009          CALL GFORCE(Q,QZ,QDD,GRAVY,AV)
0010          DO 10 J=1,6
0011          A(J,I)=AV(J)

C          SOLVE FOR THE GRAVITY AND VELOCITY COMPONENTS.

0012          GRAVY=9.80621
0013          CALL GFORCE(Q,QD,QZ,GRAVY,AV)
0014          DO 15 I=1,6
0015          TOU(I)=TOU(I)-AV(I)

C          SOLVE FOR THE JOINT ACCELERATIONS.

0016          CALL INVERT(A,QDD,TOU)

0017          RETURN
0018          END

```

註：本程式以 GFORCE 計算關節力矩。