

# 國立交通大學

資訊工程學系  
碩士論文

利用逆向動力學修正立體視覺  
人體動作原始資料

Inverse Kinematics Refinement of Stereo  
Vision-Based Human Motion Raw Data

研究生：蘇詠迪

指導教授：陳稔博士

中華民國九十四年七月

利用逆向動力學修正立體視覺人體動作原始資料

Inverse Kinematics Refinement of Stereo  
Vision-Based Human Motion Raw Data

研究生：蘇詠迪

Student : Yung-Ti Su

指導教授：陳 稔 博士

Advisor : Dr. Zen Chen



Submitted to Department of Computer Science and Information Engineering  
College of Electrical Engineering and Computer Science  
National Chiao Tung University  
in partial Fulfillment of the Requirements  
for the Degree of  
Master  
in

Computer Science and Information Engineering

July 2005

Hsinchu, Taiwan, Republic of China

中華民國九十四年七月

# 利用逆向動力學修正立體視覺人體動作原始資料

學生：蘇詠迪

指導教授：陳稔 博士

國立交通大學資訊工程學系

## 摘要

本論文的主要目的，是爲了在不貼感應標誌(Marker)的情況下，以逆向動力學修正由立體視覺所得到較粗略的人體動作原始資料，使其肢節長度與內部角度更爲正確可靠。

本論文使用兩部攝影機分別以不同的視角拍攝表演者的動作，並得到兩串立體影像序列，透過影像處理方式得到人體各關節粗略三維座標位置後，以肢節長度正確定義的人體模型爲初解，使用逆向動力學修正人體動作原始資料。

文中對本論文所定的人體模型架構做了完整的介紹，將人體模型定義爲以腳跟爲Root的樹狀架構，其中包含了十一個關節點，共二十一個關節的自由度，再以四個End-effectors在人體動作原始資料中的位置爲逆向動力學的修正目標。並且說明了各種逆向動力學方法的理論根據，Jacobian的定義、pseudo-inverse方法、Damped Least Square方法與Selectively Damped Least Square方法。

實驗中展示了以不同的初解來源進行逆向動力學修正的結果，以不同的逆向動力學解法求解的結果。修正完各個畫格後，以關鍵畫格取代完整畫格的成果。最後以IP爲初解使用的DLS逆向動力學方法通常是在效率與正確性上較好的選擇。

# Inverse Kinematics Refinement of Stereo Vision-Based Human Motion Raw Data

Student: Yung-Ti Su

Advisor: Dr. Zen Chen

Department of Computer Science and Information Engineering  
National Chiao Tung University

## Abstract

The main purpose of this thesis is to refine a sequence of stereo vision-based human motion raw data with a goal that the true limb lengths and proper joint angles can be satisfied.

Two CCD cameras are used to gain a video of human motion from two different view points. 3D joint positions are extracted from these two image sequences through techniques of image-processing and computer vision.

The thesis introduces a hierarchic human body structure, which is a tree structure using the heel as root. It includes twenty-one degrees of freedom in eleven joints. 4 end-effectors are used in inverse kinematics refinement. The overview of Jacobian, pseudo-inverse method, Damped Least Square Method is given.

In the experiments, results of refinement using different inverse kinematics methods are obtained. The analysis and comparisons of the results are given. A typical segment of a smooth motion data is used to show the similarity between the frame-by-frame refined stick figure motion and the key-frame interpolated motion. At the end, the damped least square method using the IP-based solution as the initial solution is generally top-choice solution in terms of efficiency and precision.

# 誌謝

首先感謝 陳稔教授兩年來的指導，使我對研究的方法有了正確的觀念，在研究的精神與態度上得到了良好的訓練。在研究的過程中遭遇困難時，老師往往能夠積極地給予許多的意見，對我幫助莫大。這篇論文得以順利完成，必須要感謝老師的幫忙，在此向老師敬上無盡的謝意。

感謝 PAIS 實驗室的全體成員。特別是方彥翔學長總是不厭其煩地與我討論論文中所遭遇的大小問題，並教授我許多正確的理論與概念；紀文原同學、顧正偉同學、吳昶濤學弟和陳柏聰學弟在論文的實驗拍攝以及理論的推導上，也都給予了莫大的協助。

更要感謝我的父母親，你們二十幾年來對我無怨尤的付出，我才能夠無牽掛地專心投注在研究之上，突破所有難關，完成這篇論文。

我的女朋友雅琴，在我研究遇到瓶頸時，不斷鼓勵我，使我能夠不被困難挫折所打敗，很感謝她陪伴我到現在。

最後感謝交通大學提供了我這麼好的環境與設備，感謝我的親戚朋友們，你們一直都是我的助力；無法一一致謝，僅以本篇論文獻給他們，願能夠表達我的感謝之情。

蘇詠迪 於交大

July, 2005

# 目 錄

摘要	i
Abstract	ii
誌謝	iii
1、 序論 .....	1
1.1. 研究動機 .....	1
1.2. 相關研究 .....	1
1.3. 研究方法 .....	2
1.4. 論文大綱 .....	3
2、 三維座標重建理論基礎 .....	4
2.1. 相機參數校正 .....	4
2.2. 重建立體影像中的點之三維位置座標 .....	5
3、 人體模型定義 .....	6
3.1. 人體各肢節座標系定義 .....	6
3.2. 建立各肢節local座標系間之階層幾何轉換關係 .....	6
3.3. 人體各關節點定義 .....	19
3.4. 人體肢節方向定義 .....	21
3.5. 人體各肢節長度定義 .....	22
3.6. Local角度的表示法 .....	22
3.7. 如何以IP中的資訊做為IK的初解(Initial Solution) .....	23
3.8. 雙Root的角度互換 .....	27
3.9. 逆向動力學中角度變化的演算法 .....	29
4、 逆向動力學 (Inverse Kinematics) .....	31
4.1. Jacobian定義 .....	31
4.2. Least square method (pseudo inverse method) .....	34
4.3. Damped least square method .....	34
4.4. Selectively damped least square method .....	36
4.5. 角度更新方式 .....	37
4.6. 加入各關節角度限制 .....	38
5、 實驗流程與方法 .....	39
5.1. 實驗輸入資料 .....	39
5.1.1. 立體影像資料 .....	39
5.1.2. End-Effectors 3D座標 .....	40
5.2. 實驗流程 .....	43
5.3. 停止條件的選擇 .....	45
6、 實驗數據及結果 .....	47
6.1. 比較有無Constraint的差異 .....	47
6.2. 比較以IP為初解與以上一個frame的解為初解的優劣 .....	52

6.3.	比較CCD與SDLS的差異 .....	70
6.4.	同以IP為初解，比較DLS與SDLS的差異 .....	89
6.5.	DLS與SDLS的演算法複雜度分析.....	107
6.6.	以不同的 $\lambda$ 值，觀察 $\Delta\theta$ 的收斂情形，以選擇比較好的 $\lambda$ 值 .....	110
6.7.	比較IK做每個frame與key-frame內插的結果 .....	115
7、	結論和未來工作 .....	122
7.1.	結論 .....	122
7.2.	未來工作 .....	122
	參考文獻 .....	124



## 圖 目 錄

圖 1-1 研究方法流程圖.....	2
圖 2-1 相機校正用影像View1 相機校正用影像View2.....	4
圖 3-1 $L_0$ 的local座標系(Joint $J_0$ 為其原點, $V_0$ 為其肢節的末端點).....	6
圖 3-2 $L_1$ 的local座標系(Joint $J_1$ 為其原點, $V_1$ 為其肢節的末端點).....	7
圖 3-3 將 $L_1$ local座標系移至上一層父系 $L_0$ 的local座標系.....	7
圖 3-4 將Root的 $L_0$ 移至上一層的global座標系(其child肢節 $L_1$ 隨之一起移).....	8
圖 3-5 未旋轉時, 各肢節的階層性關係圖.....	8
圖 3-6 經過旋轉時, 各肢節的階層性關係圖.....	9
圖 3-7 肢節點 $V_0$ 在其local座標系中旋轉 $\theta_0$ 圖.....	10
圖 3-8 肢節點 $V_1$ 在其local座標系中旋轉 $\theta_1$ 圖.....	11
圖 3-9 以左腳為Root的人體模型自然姿勢.....	14
圖 3-10 以右腳為Root的人體模型自然姿勢.....	15
圖 3-11 3DOF的肢節拆成 3 個 1DOF的肢節之示意圖.....	16
圖 3-12 人體關節點定義.....	20
圖 3-13 人體肢節向量定義圖.....	21
圖 3-14 假設由IP所得到的肢節向量資訊圖.....	23
圖 3-15 自然姿勢(neutral pose)的人體模型.....	23
圖 3-16 選擇不同Root的人體模型流程圖.....	27
圖 3-17 角度變化之流程圖.....	30
圖 5-1 左腳跟(Lheel)3D座標圖.....	40
圖 5-2 左手腕(Lwrist)3D座標圖.....	40
圖 5-3 頭部(Head)3D座標圖.....	41
圖 5-4 右手腕(Rwrist)3D座標圖.....	41
圖 5-5 右腳跟(Rheel)3D座標圖.....	42
圖 5-6 實驗流程圖.....	43
圖 7-1 擴充的左手定則人體肢節樹狀模型圖.....	123



# 1、 序論

## 1.1. 研究動機

由於近年來，電腦繪圖的技術不斷地進步，已經可以達到近乎真實相片的品質了，因此電影、遊戲都引入了越來越多利用電腦動畫所製作的擬真人物，爲了方便在製作動畫時，能夠使動畫裡的人物所做出來的動作更貼近於真實，不可能全部都靠動畫師去憑空想像人物的動作，因此就有了動作捕捉(Motion Capture)的技術。

動作捕捉的原理就是，追蹤影響人物動作的一些關鍵部份，這些關鍵的部份通常就是人物的關節點，因爲人物的動作主要就是靠著關節的轉動才能夠造成人物的動作改變，因此只要有了這些關節點的資訊，包含其三維位置座標及方向，就等於是有了人物動作的初步資訊。

而現今的動作捕捉技術，多半就是利用一些感應標誌(Marker)，並使用專門對應這種感應標誌的攝影機來讀取這些感應標誌的資訊。此方法的缺點就是表演者必須要穿著特殊的服裝並貼上這些感應標誌，然而這些爲了更正確地截取資料，這些服裝常常是很緊身的，因此有時會造成表演者無法很自在地表演。本篇論文就是希望能夠在不使用這些感應標誌的情況下，仍然利用這種動作捕捉技術，來重建出人物的運動，使表演者能夠無拘束地表演，而仍然能捕捉到他正確的動作，並將此動作重現於定義的人體模型上。

應用的範圍可及一些無法使用感應標誌的場合，如百米跑步，人體移動的範圍太大了。或是要分析人體的動作。以及一些微小生物的運動。

## 1.2. 相關研究

目前人體動作捕捉系統大致可以分爲以下三類：

- (1) 機械式[1,2]
- (2) 電磁波式[1,3]
- (3) 光學式[1,4,5,6]

目前逆向動力學則主要有下列方法：

- (1) Jacobian Pseudo-inverse [7]：以 Jacobian Matrix 的 Pseudo-inverse 估算所需角度變化量，優點是估算角度變化量較準確，缺點是當關節點多時，會因矩陣 Rank 較高，計算較複雜；且在接近 singularity 的時候會振盪。
- (2) Jacobian Transpose [8]：以 Jacobian 的 Transpose 代替 Pseudo-inverse 複雜的運算，好處是計算簡單，缺點是估算的角度變化量誤差較大。
- (3) Cyclic Coordinate Decent [8]：逐個以單一轉軸計算所需的角度變化量，優點是速度非常快，缺點則是一次的更新往往比其他方式的更新量更大，可能會過度修正不需修正的關節角度。
- (4) Damped Least Square Method [7]：爲了修正 Pseudo-inverse 在接近 singularity 的時候

角度過度修正造成振盪的問題，加入了 damping constant 值平衡這個修正的角度。

(5) Selectively Damped Least Square Method [7]：爲了讓 Damped Least Square Method 的更新更有效率，因此將 DLS 方法中的 damping constant 改爲動態的值，使其能更縮短逆向動力學 iteration 的次數，但計算複雜度很高。

### 1.3. 研究方法

本篇論文是由兩部攝影機，同時拍攝表演者的運動[9]，然後將這兩部攝影機所拍攝到的影片中，將其所拍攝到的每個 Frame 都取出來，製作成兩段不同視角的影像序列，並將其同步化，以確定相同編號的影像所代表的是同一時間但不同視角的影像；再進行相機校正的處理後，將每個 frame 的立體影像中的每個關節點的三維座標位置求出，但由於這樣的方式所求出的表演者骨架，會因爲拍攝的一些誤差，使得每個 Frame 的肢節長度都不盡相同，因此透過本論文的改良型逆向動力學解法，來重新計算正確的各關節點三維座標位置。流程圖如下：

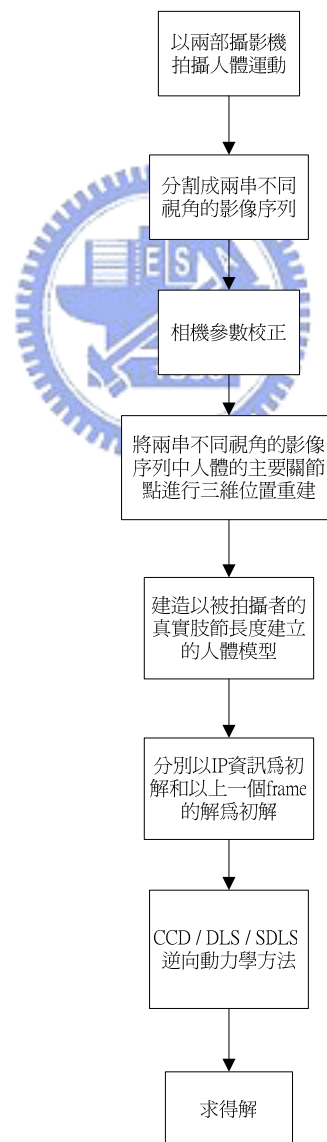


圖 1-1 研究方法流程圖

## 1.4. 論文大綱

本論文除本章為“序論”外，第二章中將討論本論文需要用到的電腦視覺定位理論，包括相機校正的方法、平面投影幾何關係的理論，利用校正後的攝影機參數，將所得的二維影像資料轉換成三維的資訊。

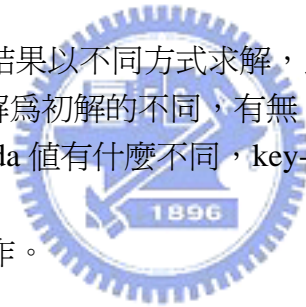
第三章中，介紹本論文所使用的人體模型之詳細定義；包括各關節點在樹狀架構中的關係、Local 座標系的定義、角度的表示及如何使人體模型轉動。

第四章介紹以 Jacobian 為基礎的幾種逆向動力學解法，包含了 Jacobian Matrix 的定義、least square method、damped least square method、selectively damped least square method。並說明 constraint 如何加入計算。

第五章說明完整的實驗流程與方法，包含如何將 IP 取得的資訊套入人體模擬、如何交互使用以左腳與以右腳為 Root 的人體模型、三個自由度的關節點如何實作等等。

第六章將實際拍攝實驗結果以不同方式求解，比較其之間的不同之處；例如以 IP 為初解與以上一個 frame 的解為初解的不同，有無 constraint 的不同，DLS 與 SDLS 的不同，DLS 中以不同的 Lamda 值有什麼不同，key-framing 的結果。

第七章為結論與未來工作。



## 2、 三維座標重建理論基礎

### 2.1.相機參數校正

利用 homogeneous 座標系統，可以求得在世界座標系 (WCS) 中的三維位置座標  $[10] [X_w \ Y_w \ Z_w]^T$  和相對應的影像座標系統 (ICS) 上的二維座標點  $[u_i \ v_i]^T$  的關係式，兩者之間的關係可用數學式表示為：

$$\begin{bmatrix} u_i \cdot w \\ v_i \cdot w \\ w \end{bmatrix} = H \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{14} \\ h_{21} & h_{22} & h_{23} & h_{24} \\ h_{31} & h_{32} & h_{33} & h_{34} \end{bmatrix} \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

其中  $w$  為一應變數， $H$  則是所要求的相機投影矩陣 (Projection matrix) 或稱為相機矩陣 (camera matrix)。

利用最小平方方法 (Least square method) 計算出矩陣  $H$  中的每一個參數，即可求得最近的投影矩陣  $H$ 。



圖 2-1 相機校正用影像 View1



相機校正用影像 View2

## 2.2. 重建立體影像中的點之三維位置座標

已知兩張影像上的點  $a_1 = [u_{a_1} \ v_{a_1}]^T$  ,  $a_2 = [u_{a_2} \ v_{a_2}]^T$  為三度空間中的點

$A = [X_A \ Y_A \ Z_A]^T$  在兩張影像上的投影點，且令兩張影像經由相機校正後求得的  $H$  矩陣各為

$$H' = \begin{bmatrix} h'_{11} & h'_{12} & h'_{13} & h'_{14} \\ h'_{21} & h'_{22} & h'_{23} & h'_{24} \\ h'_{31} & h'_{32} & h'_{33} & h'_{34} \end{bmatrix} \quad H'' = \begin{bmatrix} h''_{11} & h''_{12} & h''_{13} & h''_{14} \\ h''_{21} & h''_{22} & h''_{23} & h''_{24} \\ h''_{31} & h''_{32} & h''_{33} & h''_{34} \end{bmatrix}$$

有了以上的資訊，可以列出下列關係式

$$\begin{bmatrix} u_{a_1} w_1 \\ v_{a_1} w_1 \\ w_1 \end{bmatrix} = \begin{bmatrix} h'_{11} & h'_{12} & h'_{13} & h'_{14} \\ h'_{21} & h'_{22} & h'_{23} & h'_{24} \\ h'_{31} & h'_{32} & h'_{33} & h'_{34} \end{bmatrix} \cdot \begin{bmatrix} X_A \\ Y_A \\ Z_A \\ 1 \end{bmatrix}, \quad \begin{bmatrix} u_{a_2} w_2 \\ v_{a_2} w_2 \\ w_2 \end{bmatrix} = \begin{bmatrix} h''_{11} & h''_{12} & h''_{13} & h''_{14} \\ h''_{21} & h''_{22} & h''_{23} & h''_{24} \\ h''_{31} & h''_{32} & h''_{33} & h''_{34} \end{bmatrix} \cdot \begin{bmatrix} X_A \\ Y_A \\ Z_A \\ 1 \end{bmatrix}$$

將上列四式寫成矩陣方式為

$$\begin{bmatrix} h'_{11} - h'_{31} u_{a_1} & h'_{12} - h'_{32} u_{a_1} & h'_{13} - h'_{33} u_{a_1} \\ h'_{21} - h'_{31} v_{a_1} & h'_{22} - h'_{32} v_{a_1} & h'_{23} - h'_{33} v_{a_1} \\ h''_{11} - h''_{31} u_{a_2} & h''_{12} - h''_{32} u_{a_2} & h''_{13} - h''_{33} u_{a_2} \\ h''_{21} - h''_{31} v_{a_2} & h''_{22} - h''_{32} v_{a_2} & h''_{23} - h''_{33} v_{a_2} \end{bmatrix} \cdot \begin{bmatrix} X_A \\ Y_A \\ Z_A \end{bmatrix} = \begin{bmatrix} h'_{34} u_{a_1} - h'_{14} \\ h'_{34} v_{a_1} - h'_{14} \\ h''_{34} u_{a_2} - h''_{14} \\ h''_{34} v_{a_2} - h''_{14} \end{bmatrix}$$

為了方便，將上列等式表示為

$M_{4 \times 3} \cdot V_{3 \times 1} = b_{4 \times 1}$ ，利用最小平方方法 (least square method)，

$$M^T M V = M^T b$$

$$V = (M^T M)^{-1} M^T b = [X_A \ Y_A \ Z_A]^T$$

### 3、 人體模型定義

#### 3.1.人體各肢節座標系定義

本論文將座標系總共分爲兩種：

(1) World(Global)座標系：相機校正後得到的三維座標系。

(2) Local 座標系：在任何時刻各肢節都存在各自的 Local 座標系，但在  $t=0$  時定義一組特別的 Local 座標系稱爲 Neutral Pose(自然姿勢)，此時所有的 local 角度都=0。

但在  $t=0$  時，仍需依據圖 3-9 定義每個肢節的 local x、y、z 軸。

定義原則依左手定則如下：

x 軸定義在各肢節方向上。

z 軸定義在各關節點的主要轉軸上。

y 軸定義在 local 座標系中的”x 軸”與”z 軸”的 cross product 上。

定義每個肢節的 local x、y、z 軸後，則可以建立各肢節 local 座標系間之階層幾何轉換關係。

#### 3.2.建立各肢節 local 座標系間之階層幾何轉換關係

以兩個肢節  $L_0$ 、 $L_1$  爲例，依圖 3-1、3-2 定義出各肢節(Limb)的 local 座標系，並將各肢節中的所有 Vertex  $V_i$ ，都定義在各自的 local 座標系中；而  $L_0$ 、 $L_1$  local 座標系的原點就是 joint  $J_0$ 、 $J_1$ 。

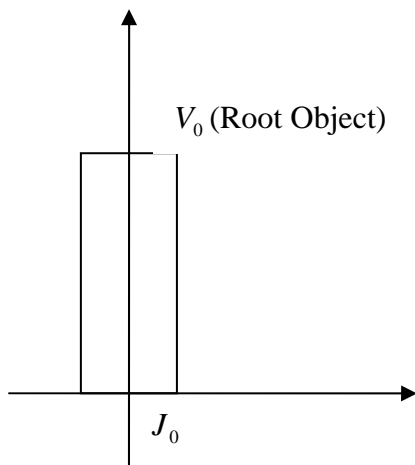


圖 3-1  $L_0$  的 local 座標系(Joint  $J_0$  爲其原點， $V_0$  爲其肢節的末端點)

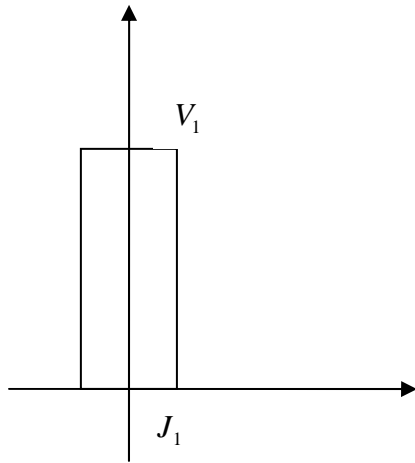


圖 3-2  $L_1$  的 local 座標系(Joint  $J_1$  為其原點， $V_1$  為其肢節的末端點)

定義 Neutral Pose( $t=0$  時)，各肢節 local 座標系間樹狀架構之幾何轉換關係，在每個 parent 肢節與其 child 肢節之間，都各定義一組 constant transformation(可能包含 rotation 及 translation 用來表示 child 肢節的 local 座標系與 parent 肢節的 local 座標系之間的相對關係)；Root 肢節則會與 global 座標系有一組 transformation 用來表示 Root 肢節的 local 座標系與 global 座標系之間的相對關係。如此就將每個 child 肢節都定義在其 parent 肢節上。圖 3-3、3-4 表示如何將兩肢節  $V_0$ 、 $V_1$  依序表示在其 parent 肢節的 local 座標系上：

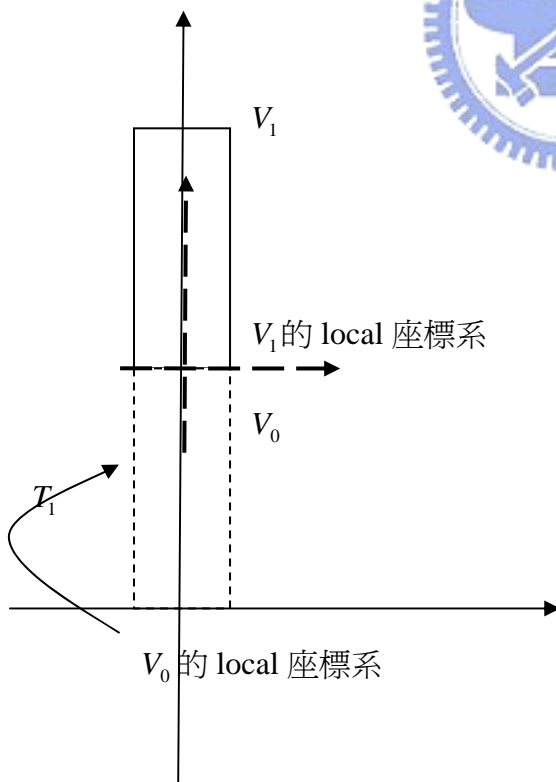


圖 3-3 將  $L_1$  local 座標系移至上一層父系  $L_0$  的 local 座標系

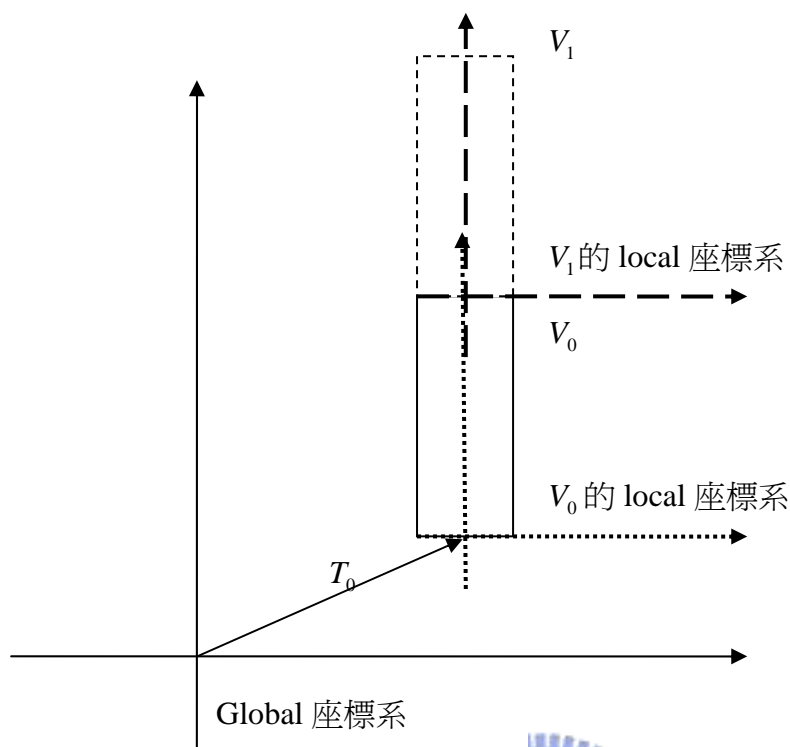


圖 3-4 將 Root 的  $L_0$  移至上一層的 global 座標系(其 child 肢節  $L_1$  隨之一起移)

如圖 3-3、3-4 所示，其肢節向量定義在各自的 local 座標系中分別可表示為  $V_0$ 、 $V_1$ ，令 parent 肢節與其 child 肢節之間的 constant transformation 分別為  $T_0$ 、 $T_1$ 。在未旋轉時，其之間的階層性關係表示如下圖：

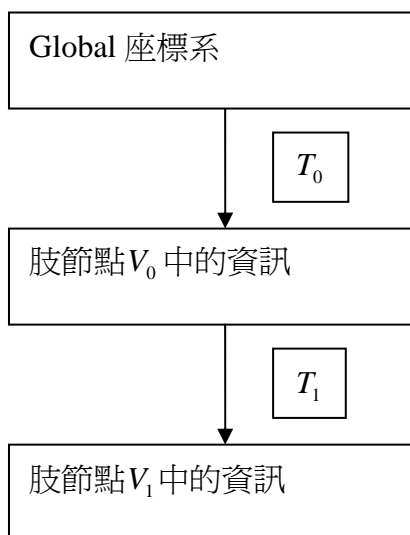


圖 3-5 未旋轉時，各肢節的階層性關係圖



若要將  $V_0$ 、 $V_1$  表示在 global 座標系中，則可寫為：

$$V_0^{global} = T_0 \cdot V_0$$

$$V_1^{global} = T_0 \cdot T_1 \cdot V_1$$

若要對肢節  $V_0$ 、 $V_1$  分別旋轉  $\theta_0$ 、 $\theta_1$  角度，則分別在其各自的 local 座標系中，依 fixed-angle rotation 旋轉  $\theta_0$ 、 $\theta_1$  角度，旋轉時是旋轉 local 座標系中的所有物體，並不改變 local 座標系本身；表示其階層性關係如下圖：

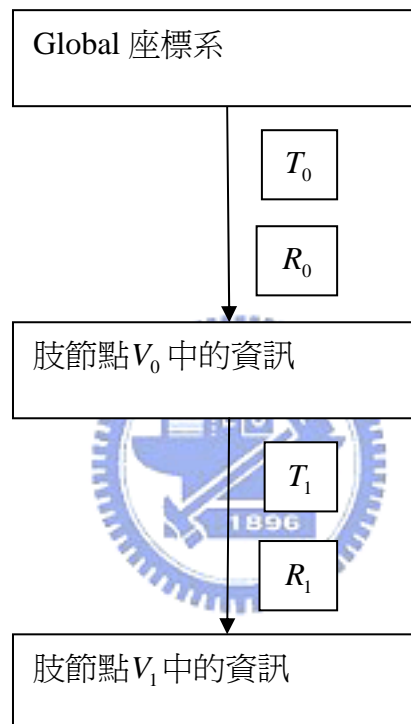


圖 3-6 經過旋轉時，各肢節的階層性關係圖

再以圖 3-7、3-8 說明肢節點  $V_0$ 、 $V_1$  分別旋轉  $\theta_0$ 、 $\theta_1$

先對肢節點 $V_0$ 旋轉 $\theta_0$ ：

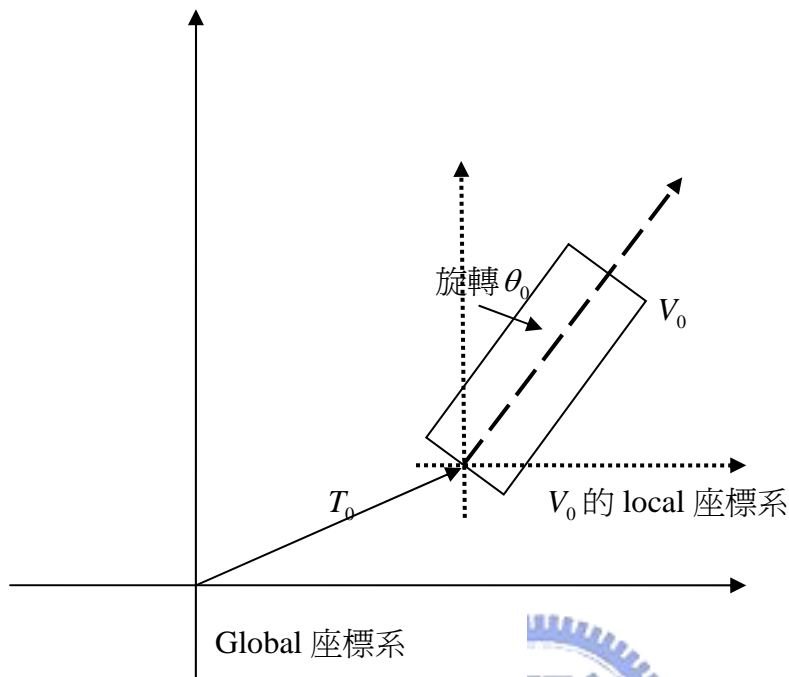


圖 3-7 肢節點 $V_0$ 在其 local 座標系中旋轉 $\theta_0$ 圖

將 $V_0$ 表示在 global 座標系中，則可寫為：

$$V_0^{global} = T_0 \cdot R_0(\theta_0) \cdot V_0$$

其中 $R_0(\theta_0) \cdot V_0$ 表示將肢節點 $V_0$ 在其本身的 local 座標系中做 fixed-angle rotation  $R_0(\theta_0)$ 。因此仍表示在 $V_0$ 本身的 local 座標系中。

再經過 transformation  $T_0$ 將旋轉後的 $V_0$ 表示在 global 座標系中，寫為 $T_0 \cdot R_0(\theta_0) \cdot V_0$ 。

再對肢節點 $V_1$ 旋轉 $\theta_1$ ：

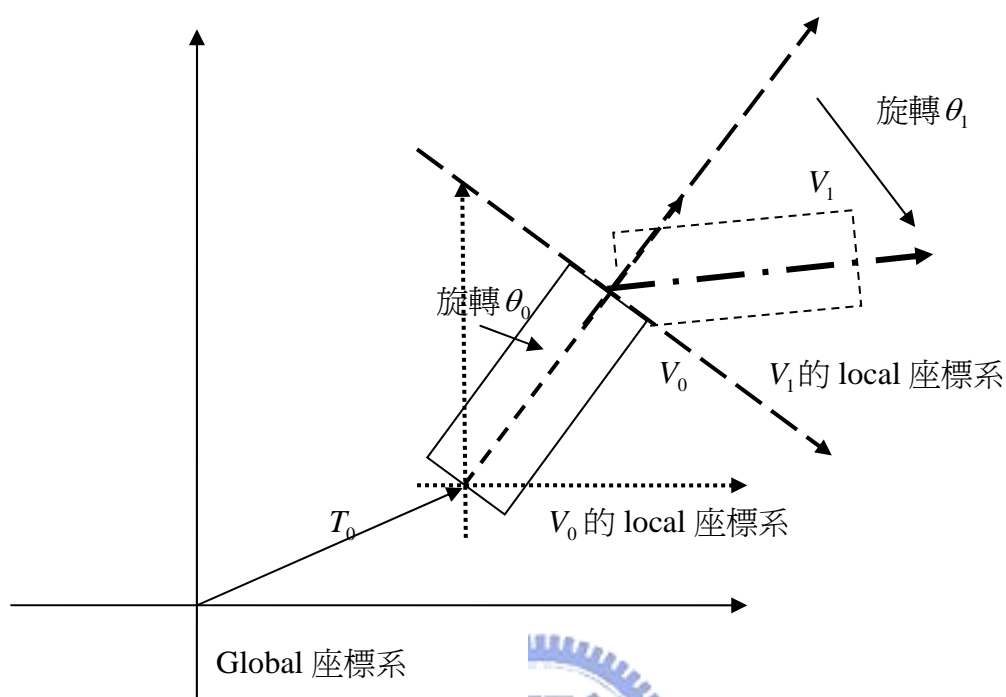


圖 3-8 肢節點 $V_1$ 在其 local 座標系中旋轉 $\theta_1$ 圖

將 $V_1$ 表示在 global 座標系中，則可寫為：

$$V_1^{global} = T_0 \cdot R_0(\theta_0) \cdot T_1 \cdot R_1(\theta_1) \cdot V_1$$

其中 $R_1(\theta_1) \cdot V_1$ 表示將 $V_1$ 在其本身的 local 座標系中做 fixed-angle rotation  $R_1(\theta_1)$ 。因此仍表示在 $V_1$ 本身的 local 座標系中。

再經過 transformation  $T_1$ 將旋轉後的 $V_1$ 表示在 $V_0$ 的 local 座標系中，寫為 $T_1 \cdot R_1(\theta_1) \cdot V_1$ 。

$V_0$ 再對其 local 座標系中的所有物體(包含 $V_1$ )做 fixed-angle rotation  $R_0(\theta_0)$ 。

$$\text{寫為 } R_0(\theta_0) \cdot T_1 \cdot R_1(\theta_1) \cdot V_1$$

再經過 transformation  $T_0$ 將 $V_1$ 表示在 global 座標系中，

$$\text{寫為 } T_0 \cdot R_0(\theta_0) \cdot T_1 \cdot R_1(\theta_1) \cdot V_1。$$

若每個 joints 都是 1-DOF，若肢節  $V_0$ 、 $V_1$ ，再分別旋轉  $\theta'_0$ 、 $\theta'_1$  角度，由於肢節各自的旋轉都在其 local 座標系中進行，肢節的 local 座標系是不會改變的，因此肢節永遠是對同一個轉軸旋轉，

原本已經過  $\theta_0$  的  $V_0$  表示在 global 座標系中，可寫為：

$$V_0^{global} = T_0 \cdot R_0(\theta_0) \cdot V_0$$

其中  $R_0(\theta_0) \cdot V_0$  表示將  $V_0$  在其本身的 local 座標系中做 fixed-angle rotation  $R_0(\theta_0)$ 。因此仍表示在  $V_0$  本身的 local 座標系中。

若再旋轉  $\theta'_0$ ，可表示為：

$$V_0^{global} = T_0 \cdot R_0(\theta'_0) \cdot R_0(\theta_0) \cdot V_0$$

其中  $R_0(\theta'_0) \cdot R_0(\theta_0)$  可合併為  $R_0(\theta'_0 + \theta_0)$

$$\text{寫為 } V_0^{global} = T_0 \cdot R_0(\theta'_0 + \theta_0) \cdot V_0$$

同理，

原本已經過  $\theta_0$ 、 $\theta_1$  旋轉的  $V_1$  表示在 global 座標系中，可寫為：

$$V_1^{global} = T_0 \cdot R_0(\theta_0) \cdot T_1 \cdot R_1(\theta_1) \cdot V_1$$

其中  $R_1(\theta_1) \cdot V_1$  表示將  $V_1$  在其本身的 local 座標系中做 fixed-angle rotation  $R_1(\theta_1)$ ，因此仍表示在  $V_1$  本身的 local 座標系中； $R_0(\theta_0) \cdot (T_1 \cdot R_1(\theta_1) \cdot V_1)$  表示在  $V_0$  本身的 local 座標系中對所有物體做 fixed-angle rotation  $R_0(\theta_0)$ ，因此這個旋轉仍表示在  $V_0$  本身的 local 座標系中。

因此若再旋轉  $\theta'_0$ 、 $\theta'_1$ ，可表示為：

$$V_1^{global} = T_0 \cdot R_0(\theta'_0) \cdot R_0(\theta_0) \cdot T_1 \cdot R_1(\theta'_1) \cdot R_1(\theta_1) \cdot V_1$$

其中  $R_0(\theta'_0) \cdot R_0(\theta_0)$  可合併為  $R_0(\theta'_0 + \theta_0)$ ； $R_1(\theta'_1) \cdot R_1(\theta_1)$  可合併為  $R_1(\theta'_1 + \theta_1)$

$$\text{寫為 } V_1^{global} = T_0 \cdot R_0(\theta'_0 + \theta_0) \cdot T_1 \cdot R_1(\theta'_1 + \theta_1) \cdot V_1$$

由此可見先旋轉  $\theta_0$ 、 $\theta_1$  再旋轉  $\theta'_0$ 、 $\theta'_1$  與一次旋轉  $\theta_0 + \theta'_0$ 、 $\theta_1 + \theta'_1$  是等價的。



欲依據上述方法定義一個完整的人體在 Neutral Pose 下各肢節 local 座標系及樹狀架構。

定義 Neutral Pose 為：

當各個肢節  $t=0$  時未對 joint 做旋轉(即角度為 0)之前的相互幾何關係圖，稱爲人體肢節的 Neutral Pose。由此一圖可看出每個肢節與其上一層次父系座標系的 constant transform matrix 爲何。

再考慮由於資料是經由影像處理所得來，因此資料的準確性是有限的，而身體輪廓較好判斷的點只有頭、手、腳這些端點是較爲準確的，因此把這些端點當做是人體模型中的 End-Effector 或是 Root；而除了這些點以外的點就較難用影像處理來判斷了，因此就取幾個比較重要的關節做爲 Root 到 End-Effector 之間的端點，並有 1DOF 與 3DOF 的關節，以表示人體正確的動作，並使求解的複雜度不至於太高，且認爲內部的關節點較難定位，因此只考慮以人體的端點做 Root，又認爲人體至少需靠單腳站立，而踩地的該腳位置固定，因而制定了如圖 3-9、3-10 分別以左腳和右腳爲 Root 的人體肢節模型，有 11 個關節點共 21 個關節自由度、4 個 End-effectors。

圖 3-9、3-10 就是本論文所定義的自然姿勢(neutral pose)時( $t=0$ )，各肢節的 local 座標系及樹狀架構圖。



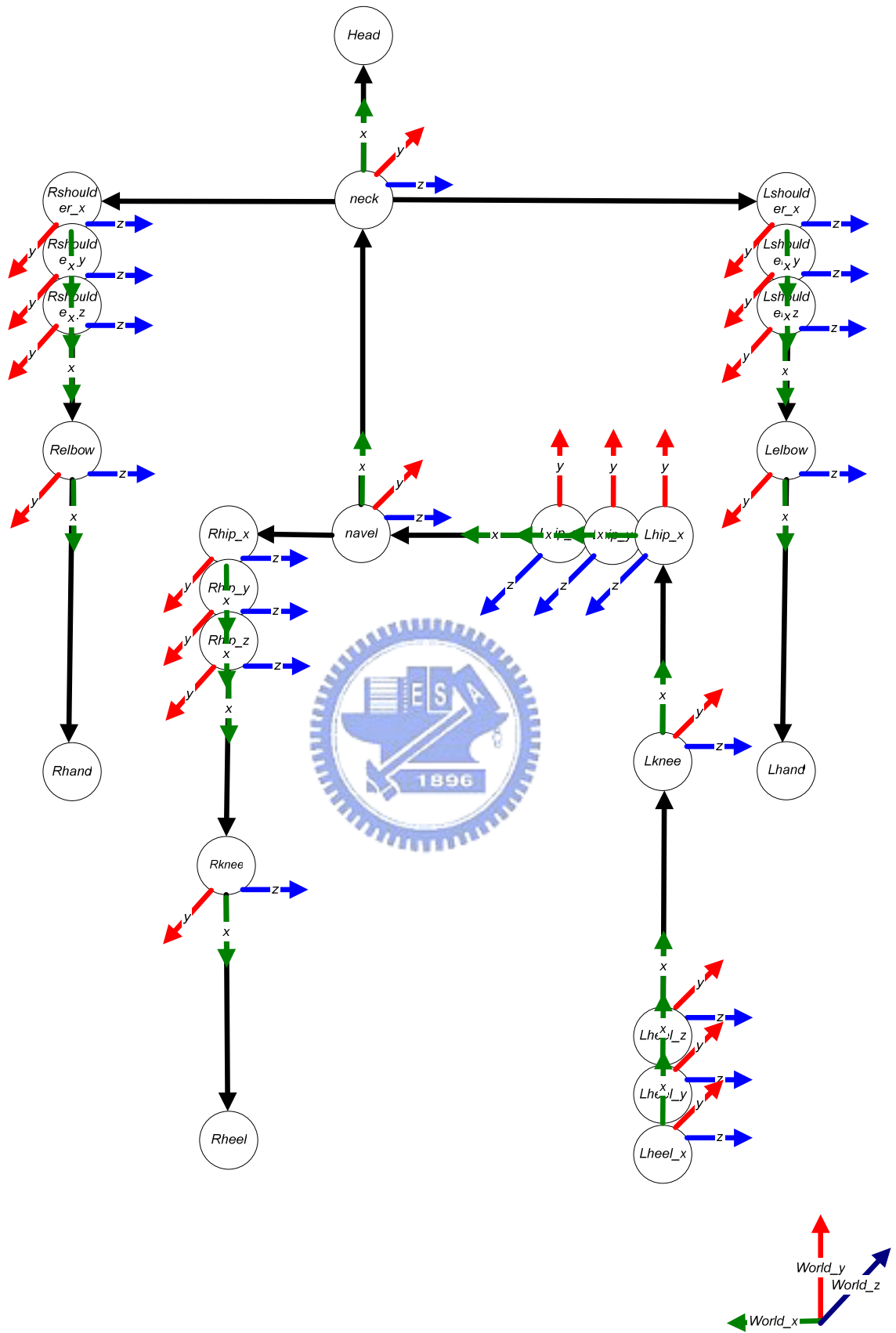


圖 3-9 以左腳為 Root 的人體模型自然姿勢

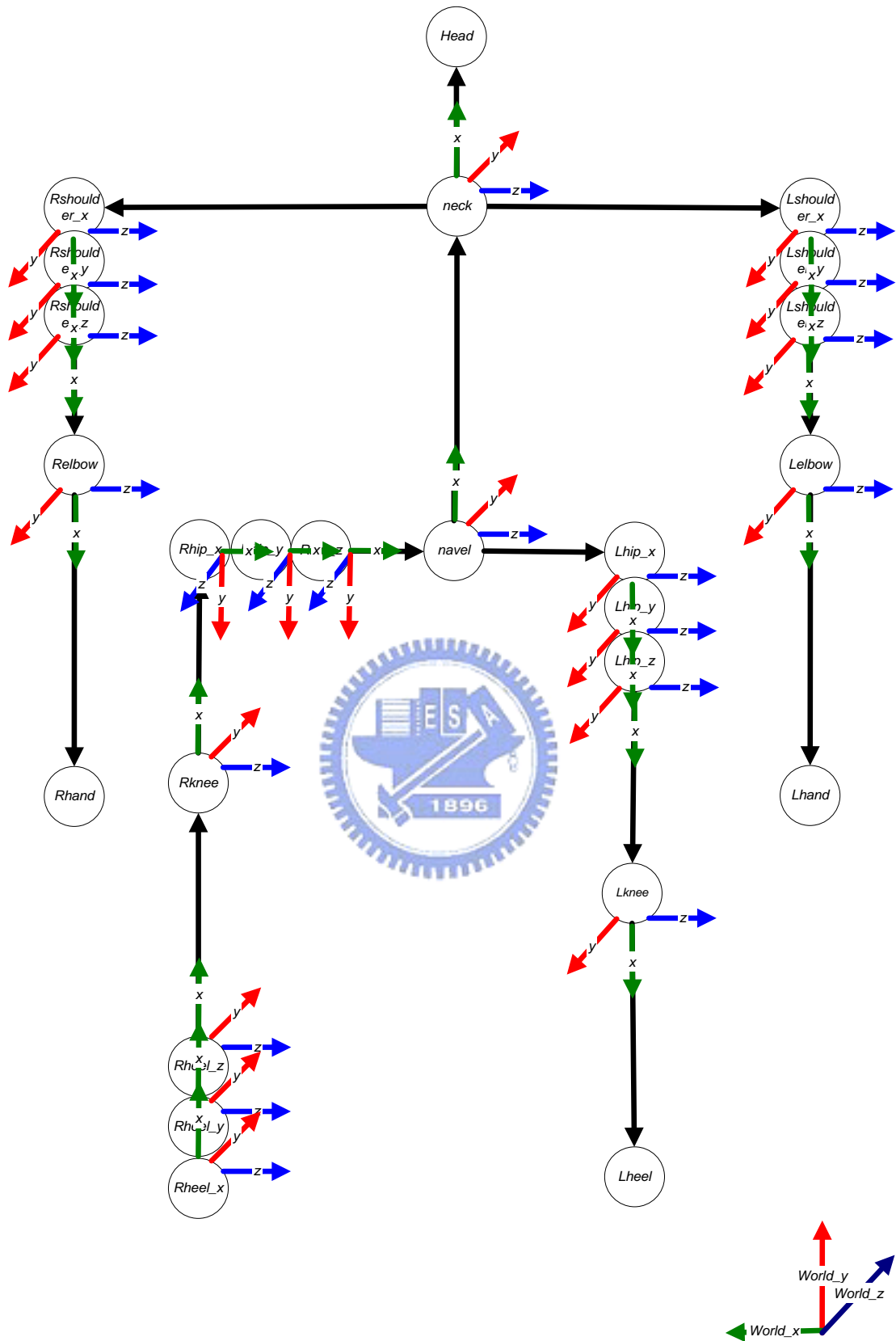


圖 3-10 以右腳為 Root 的人體模型自然姿勢

在關節點的自由度是 3DOF 時，為了配合 Jacobian matrix 中的每個 element 僅考慮一個轉軸對 End-effector 位置的影響，本論文將 3-DOF 拆成 3 個 1-DOF 的座標系及其之間的 constant transform matrix = I

並且將 x-dof 與 y-dof 的肢節向量設為 0

只有 z-dof 的肢節向量是此 3DOF 肢節的肢節向量

以 Lheel 為例；hierarchical tree 順序為 Lheel\_x→Lheel\_y→Lheel\_z

就是分別以 Lheel 肢節 local 座標系中 x、y、z 軸為轉軸的三個肢節。

如圖 3-11 所示：

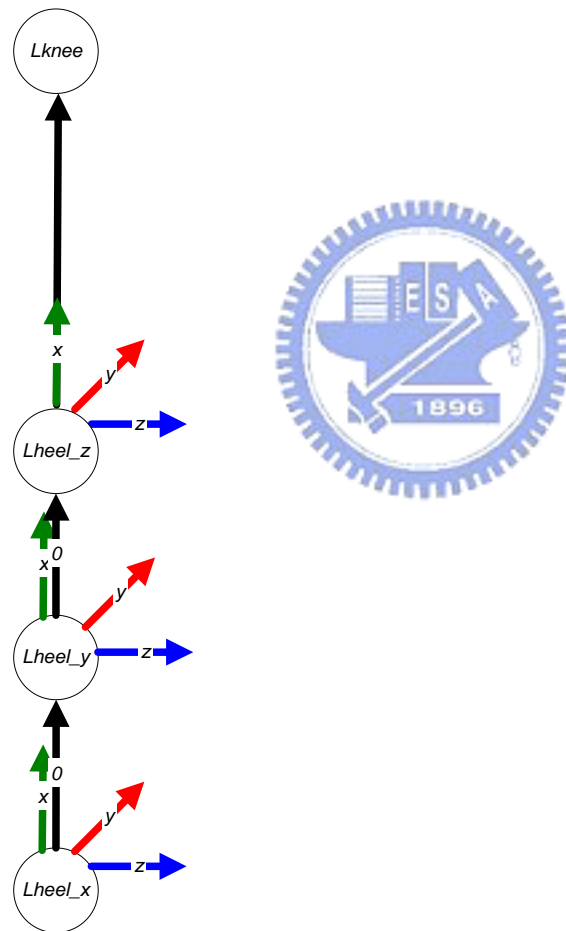


圖 3-11 3DOF 的肢節拆成 3 個 1DOF 的肢節之示意圖



各肢節對其 local 座標系原點的 joint 做 1-DOF 旋轉的表示式如下：

定義：

$node_i$ ：第  $i$  個肢節

$node_i - x / y / z - dof$ ：以  $x$  或  $y$  或  $z$  軸為轉軸的三個 local 座標系

$V_{node_i - x - dof}$ ： $node_i - x - dof$  中的肢節向量表示在本身的 local 座標系中

$[V_{node_i - x - dof}(0)]^{node_{i-1}}$ ：以  $x$  軸為轉軸的 local 座標系在時間  $t = 0$  時的肢節向量表示在上標之  $node_{i-1}$  的 local 座標系中

$T[node_i - x - dof]$ ： $node_i - x - dof$  的 local 座標系與其 parent 的 local 座標系之間的 constant transformation

$T[\cdot]$  為 Transformation， $R(\cdot)$  為 Rotation。

以某一個 3-DOF 肢節  $node_i$  所拆成的三個肢節  $node_i - x - dof$ 、 $node_i - y - dof$ 、 $node_i - z - dof$  為例(階層順序依序為  $node_i - x - dof \rightarrow node_i - y - dof \rightarrow node_i - z - dof$ )：

(1) 在時刻  $t=0$ ，當  $t=0$  時角度都還未轉動時；三個肢節  $node_i - x - dof$ 、 $node_i - y - dof$ 、 $node_i - z - dof$  中的肢節向量表示在其 parent  $node_{i-1}$  的 local 座標系中分別為：

$$[V_{node_i - x - dof}(0)]^{node_{i-1}} = T[node_i - x - dof] \cdot V_{node_i - x - dof}$$

$$[V_{node_i - y - dof}(0)]^{node_{i-1}} = T[node_i - x - dof] \cdot T[node_i - y - dof] \cdot V_{node_i - y - dof}$$

$$[V_{node_i - z - dof}(0)]^{node_{i-1}} = T[node_i - x - dof] \cdot T[node_i - y - dof] \cdot T[node_i - z - dof] \cdot V_{node_i - z - dof}$$

(2) 在時刻  $t=1$ ，三個肢節分別轉動  $\theta_x$ 、 $\theta_y$ 、 $\theta_z$ ；三個肢節  $node_i - x - dof$ 、 $node_i - y - dof$ 、 $node_i - z - dof$  中的肢節向量表示在其 parent  $node_{i-1}$  的 local 座標系中分別為：

$$[V_{node_i - x - dof}(1)]^{node_{i-1}} = T[node_i - x - dof] \cdot R_x(\theta_x) \cdot V_{node_i - x - dof}$$

$$[V_{node_i - y - dof}(1)]^{node_{i-1}} = T[node_i - x - dof] \cdot R_x(\theta_x) \cdot T[node_i - y - dof] \cdot R_y(\theta_y) \cdot V_{node_i - y - dof}$$

$$[V_{node_i - z - dof}(1)]^{node_{i-1}} = T[node_i - x - dof] \cdot R_x(\theta_x) \cdot T[node_i - y - dof] \cdot R_y(\theta_y) \cdot T[node_i - z - dof] \cdot R_z(\theta_z) \cdot V_{node_i - z - dof}$$

又由於本論文將 3-DOF 拆成 3 個 1-DOF 的座標系後，其之間的 constant transform matrix = I；因此  $T[node_i - y - dof]$  與  $T[node_i - z - dof]$  都是等於 I；因此可再改寫為：

$$[V_{node_i-x-dof}(1)]^{node_{i-1}} = T[node_i-x-dof] \cdot R_x(\theta_x) \cdot V_{node_i-x-dof}$$

$$[V_{node_i-y-dof}(1)]^{node_{i-1}} = T[node_i-x-dof] \cdot R_x(\theta_x) \cdot R_y(\theta_y) \cdot V_{node_i-y-dof}$$

$$[V_{node_i-z-dof}(1)]^{node_{i-1}} = T[node_i-x-dof] \cdot R_x(\theta_x) \cdot R_y(\theta_y) \cdot R_z(\theta_z) \cdot V_{node_i-z-dof}$$

之後也可如此改寫，但爲了更清楚表示其階層性架構，仍先不以 I 替換  $T[node_i-y-dof]$  與  $T[node_i-z-dof]$ 。

(3) 在時刻  $t=2$ ，三個肢節分別轉動再  $\theta'_x$ 、 $\theta'_y$ 、 $\theta'_z$ ；三個肢節  $node_i-x-dof$ 、 $node_i-y-dof$ 、 $node_i-z-dof$  中的肢節向量表示在其 parent  $node_{i-1}$  的 local 座標系中分別爲：

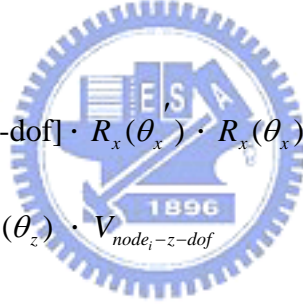
$$[V_{node_i-x-dof}(2)]^{node_{i-1}} = T[node_i-x-dof] \cdot R_x(\theta'_x) \cdot R_x(\theta_x) \cdot V_{node_i-x-dof}$$

$$[V_{node_i-y-dof}(2)]^{node_{i-1}} = T[node_i-x-dof] \cdot R_x(\theta'_x) \cdot R_x(\theta_x) \cdot T[node_i-y-dof] \cdot R_y(\theta'_y) \cdot R_y(\theta_y) \cdot$$

$$V_{node_i-y-dof}$$

$$[V_{node_i-z-dof}(2)]^{node_{i-1}} = T[node_i-x-dof] \cdot R_x(\theta'_x) \cdot R_x(\theta_x) \cdot T[node_i-y-dof] \cdot R_y(\theta'_y) \cdot R_y(\theta_y) \cdot$$

$$T[node_i-z-dof] \cdot R_z(\theta'_z) \cdot R_z(\theta_z) \cdot V_{node_i-z-dof}$$



由於每個肢節都對相同的 local 轉軸旋轉，因此這個旋轉矩陣是可以合併的，上述三式可再寫爲：

$$[V_{node_i-x-dof}(2)]^{node_{i-1}} = T[node_i-x-dof] \cdot R_x(\theta'_x + \theta_x) \cdot V_{node_i-x-dof}$$

$$[V_{node_i-y-dof}(2)]^{node_{i-1}} = T[node_i-x-dof] \cdot R_x(\theta'_x + \theta_x) \cdot T[node_i-y-dof] \cdot R_y(\theta'_y + \theta_y) \cdot V_{node_i-y-dof}$$

$$[V_{node_i-z-dof}(2)]^{node_{i-1}} = T[node_i-x-dof] \cdot R_x(\theta'_x + \theta_x) \cdot T[node_i-y-dof] \cdot R_y(\theta'_y + \theta_y) \cdot$$

$$T[node_i-z-dof] \cdot R_z(\theta'_z + \theta_z) \cdot V_{node_i-z-dof}$$

因此依序轉動  $\theta_x$ 、 $\theta_y$ 、 $\theta_z$  再轉動  $\theta'_x$ 、 $\theta'_y$ 、 $\theta'_z$  與一次轉動  $\theta_x + \theta'_x$ 、 $\theta_y + \theta'_y$ 、 $\theta_z + \theta'_z$  是等價的。故各關節的旋轉角度是可以累算的。

### 3.3. 人體各關節點定義

我們由兩部攝影機分別得到 View1 與 View2 兩組影像序列，並將同一時刻的兩個 View 的影像分別標記我們所需要的關節點(左右腳尖、左右膝蓋、左右髖部、左右肩膀、左右手肘、左右手腕、頭頂)，經過三維重建的處理得到這些關節點在各時刻的三維位置座標。

在這些位置座標中我們較信任端點部份的位置座標(頭頂、左右手腕、左右腳尖)，因此我們把這些點當做是 End-effector 或是 Root。內部的關節點雖不信任其三維位置座標，但仍參考其肢節向量。

因此我們由 IP 得到的資訊如下：

以下位置都是表示在Global座標系中。

左腳尖位置：LheelPos

左膝蓋位置：LkneePos

左髖部位置：LhipPos

左肩膀位置：LshoulderPos

左手肘位置：LelbowPos

左手腕位置：LwristPos

頭頂位置：HeadPos

右腳尖位置：RheelPos

右膝蓋位置：RkneePos

右髖部位置：RhipPos

右肩膀位置：RshoulderPos

右手肘位置：RelbowPos

右手腕位置：RwristPos



再對這些資料做處理：

肚臍的位置為左髖部與右髖部的中點

肚臍位置：NavelPos = (LhipPos + RhipPos)/2

脖子位置為左肩膀與右肩膀的中點

脖子位置：NeckPos = (LshoulderPos + RshoulderPos)/2

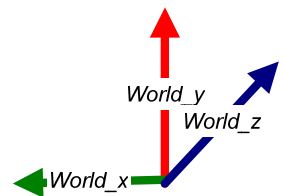
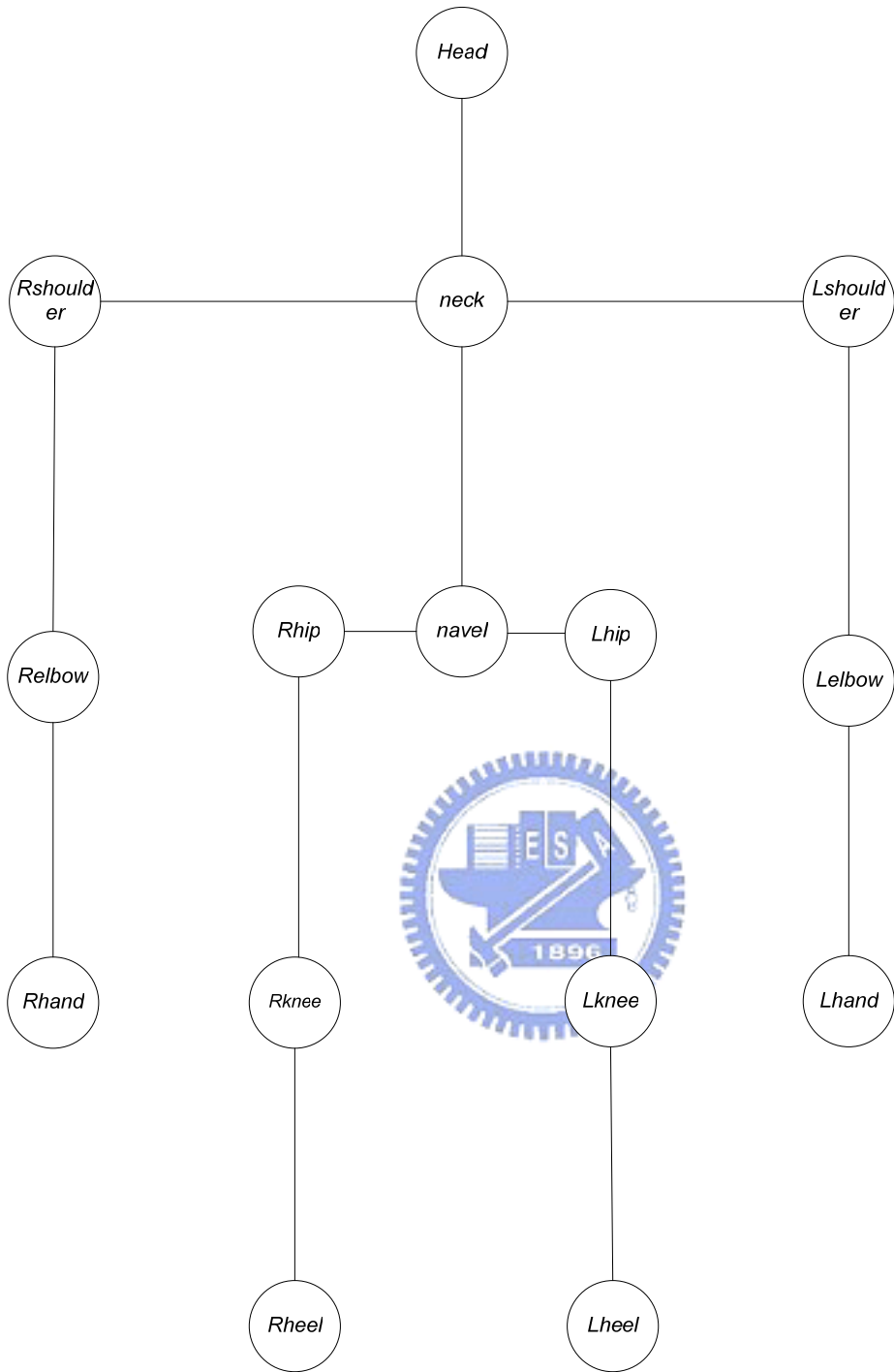


圖 3-12 人體關節點定義

### 3.4. 人體肢節方向定義

由上節中所得到的各關節點的Global位置計算出各肢節的Global向量為：

左小腿向量  $Lh\_Lk = LkneePos - LheelPos$

左大腿向量  $Lk\_Lp = LhipPos - LkneePos$

左右髖向量  $Lp\_Rp = RhipPos - LhipPos$

脊椎向量  $Na\_Ne = NeckPos - NavelPos$

脖子向量  $Ne\_He = HeadPos - NeckPos$

左上臂向量  $Ls\_Le = LelbowPos - LshoulderPos$

左下臂向量  $Le\_Lw = LwristPos - LelbowPos$

右上臂向量  $Rs\_Re = RelbowPos - RshoulderPos$

右下臂向量  $Re\_Rw = RwristPos - RelbowPos$

右大腿向量  $Rp\_Rk = RkneePos - RhipPos$

右小腿向量  $Rk\_Rh = RheelPos - RkneePos$

對應如下圖：

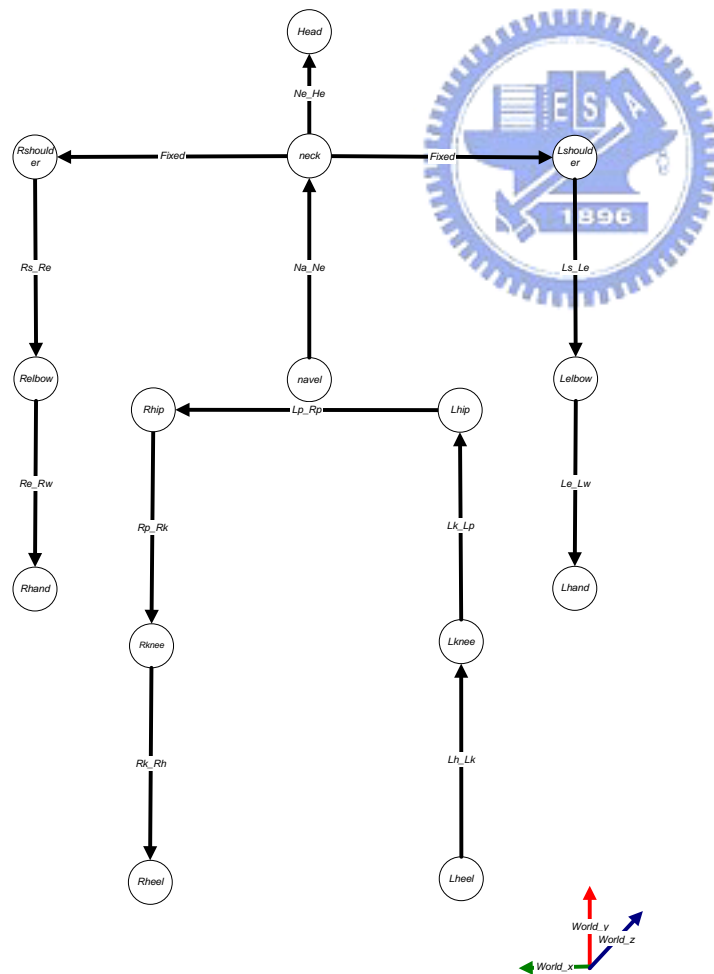


圖 3-13 人體肢節向量定義圖

### 3.5. 人體各肢節長度定義

上一節已經定義了一個人體模型的架構，然而由這個架構只能夠得知每個關節的相連的關係，也就是一個樹狀架構，但是要表示一個人體的活動除了這個架構外，還必須要有關節與關節之間的距離；也就是肢節的長度，將人體的肢節與關節之關係定義如下：

小腿 長度：heel(腳跟)與 knee(膝蓋)之間的距離

大腿 長度：knee(膝蓋)與 hip(臀部)之間的距離

髖部 長度：hip(臀部)與 navel(肚臍)之間的距離

身體 長度：navel(肚臍)與 neck(脖子)之間的距離

頸部 長度：neck(脖子)與 head(頭頂)之間的距離

肩部 長度：neck(脖子)與 shoulder(肩膀)之間的距離

上臂 長度：shoulder(肩膀)與 elbow(手肘)之間的距離

下臂 長度：elbow(手肘)與 hand(手部)之間的距離

有了這個定義後，量測被拍攝者的這些肢體上的長度，套入所定義的這個人體模型，就等於是使人體模型真正對應到被拍攝者的身體資訊。



### 3.6. Local 角度的表示法

在圖 3-9、3-10 中定義了人體的自然姿勢(neutral pose)模型，在本文中定義，在這個參考姿勢的狀態下，人體的每個關節點的 local 角度值都為 0；下一步則要使這個人體的模型能夠旋轉；且必須要能夠記錄其旋轉後的狀態。因此依據每個關節點各自的旋轉軸，順時針旋轉角度增加，逆時針旋轉角度減少。

因為關節點的自由度都是 1DOF，一個關節點只有一個旋轉軸，記錄的角度就是以該旋轉軸在自然姿勢(neutral pose)下的 local 角度為 0，順時針旋轉角度增加，逆時針旋轉角度減少。

### 3.7. 如何以 IP 中的資訊做為 IK 的初解(Initial Solution)

本節目標為求一組 local 角度  $\bar{\theta}$ ，套入 Neutral Pose 後，能使得人體模型的姿勢與 IP 所求得的姿勢相同(在 global 座標系中的肢節向量方向相同)。

以左腳根(Root)到左膝為例，說明如何利用 IP 所得到的 global 座標系中的肢節向量，求出各關節點的 local 角度做為 IK 的初解，由於自然姿勢(neutral pose)的 local 角度定義為 0，因此想求各關節點的 local 角度，就是求把人體模型肢節從自然姿勢(neutral pose)旋轉到與 IP 所得到的肢節向量相同的方向所需旋轉的 local 角度。

假設我們由 IP 所得到的肢節向量資訊如下：

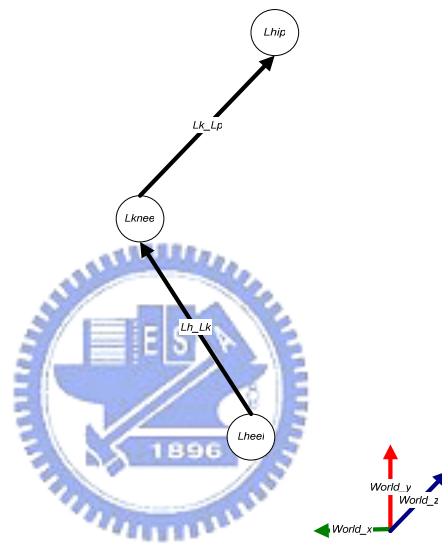


圖 3-14 假設由 IP 所得到的肢節向量資訊圖

自然姿勢(neutral pose)的人體模型如下：

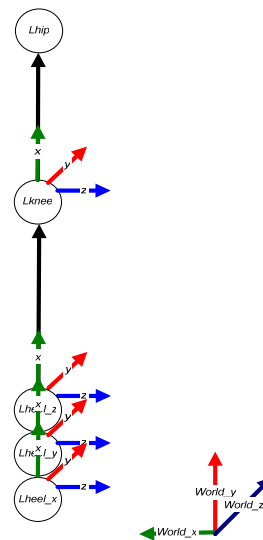


圖 3-15 自然姿勢(neutral pose)的人體模型

由於只是要求 local 座標系中三個座標軸方向表示在 global 座標系上，因此 constant transformation 中的 translation 可以不用考慮，只考慮 rotation 即可。因此我們將 root 的

$$\text{local 座標系與 global coordinate 之間的 constant transformation } T_0 = \begin{pmatrix} R_{11} & R_{12} & R_{13} & t_x \\ R_{21} & R_{22} & R_{23} & t_y \\ R_{31} & R_{32} & R_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \text{ 中}$$

的 rotation 部份表示為  $T_0 \cdot R$ 。

使用立體影像技術取得 Lheel、Lknee、Lhip 在 global 座標系中的三維位置，表示為  $[Lheel]_{IP}$ 、 $[Lknee]_{IP}$ 、 $[Lhip]_{IP}$ 。在 Neutral Pose 時，Lheel 的三個 local 座標系的座標軸方向都是相同的，x 軸方向是從  $[Lheel]_{Local}$  到  $[Lknee]_{Local}$  (i.e.  $[Lh - Lk]_{Local}$ )，z 軸方向是 Lknee 的轉軸方向，因此可以套用 Sec 3.1 中的正向動力學式子。

$$T_0 \cdot R \cdot R(\theta_{Lheel\_x}) \cdot R(\theta_{Lheel\_y}) \cdot R(\theta_{Lheel\_z}) \cdot (1,0,0)^T = \frac{[Lh - Lk]_{IP}}{|[Lh - Lk]_{IP}|}$$

$$T_0 \cdot R \cdot R(\theta_{Lheel\_x}) \cdot R(\theta_{Lheel\_y}) \cdot R(\theta_{Lheel\_z}) \cdot (0,0,1)^T = \frac{-[Lh - Lk]_{IP} \times [Lk - Lp]_{IP}}{|-[Lh - Lk]_{IP} \times [Lk - Lp]_{IP}|}$$

$$T_0 \cdot R \cdot R(\theta_{Lheel\_x}) \cdot R(\theta_{Lheel\_y}) \cdot R(\theta_{Lheel\_z}) \cdot (0,1,0)^T$$

$$= \frac{[Lh - Lk]_{IP} \times (-[Lh - Lk]_{IP} \times [Lk - Lp]_{IP})}{|[Lh - Lk]_{IP} \times (-[Lh - Lk]_{IP} \times [Lk - Lp]_{IP})|}$$

由上述三等式可以求出  $\theta_{Lheel\_x}$ 、 $\theta_{Lheel\_y}$ 、 $\theta_{Lheel\_z}$ ，因為 constant 矩陣  $T_0 \cdot R$  為已知。

因此  $R_x(\theta_{Lheel\_x}) \cdot R_y(\theta_{Lheel\_y}) \cdot R_z(\theta_{Lheel\_z}) = T_0 \cdot R^{-1}$ 。

$$\left[ \begin{pmatrix} [Lh - Lk]_{IP} \\ [Lh - Lk]_{IP} \end{pmatrix}^T \begin{pmatrix} [Lh - Lk]_{IP} \times (-[Lh - Lk]_{IP} \times [Lk - Lp]_{IP}) \\ [Lh - Lk]_{IP} \times (-[Lh - Lk]_{IP} \times [Lk - Lp]_{IP}) \end{pmatrix}^T \begin{pmatrix} -[Lh - Lk]_{IP} \times [Lk - Lp]_{IP} \\ |-[Lh - Lk]_{IP} \times [Lk - Lp]_{IP}| \end{pmatrix}^T \right]$$

= Rotation Matrix M =  $\{m_{ij}\}$

Rotation Matrix M =  $R_x(\theta_{Lheel\_x}) \cdot R_y(\theta_{Lheel\_y}) \cdot R_z(\theta_{Lheel\_z})$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_x) & -\sin(\theta_x) \\ 0 & \sin(\theta_x) & \cos(\theta_x) \end{bmatrix} \begin{bmatrix} \cos(\theta_y) & 0 & \sin(\theta_y) \\ 0 & 1 & 0 \\ -\sin(\theta_y) & 0 & \cos(\theta_y) \end{bmatrix} \begin{bmatrix} \cos(\theta_z) & -\sin(\theta_z) & 0 \\ \sin(\theta_z) & \cos(\theta_z) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

=



$$\begin{bmatrix} \cos(\theta_y)\cos(\theta_z) & -\cos(\theta_y)\sin(\theta_z) & \sin(\theta_y) \\ \sin(\theta_x)\sin(\theta_y)\cos(\theta_z) + \cos(\theta_x)\sin(\theta_z) & -\sin(\theta_x)\sin(\theta_y)\sin(\theta_z) + \cos(\theta_x)\cos(\theta_z) & -\sin(\theta_x)\cos(\theta_y) \\ -\cos(\theta_x)\sin(\theta_y)\cos(\theta_z) + \sin(\theta_x)\sin(\theta_z) & \cos(\theta_x)\sin(\theta_y)\sin(\theta_z) + \sin(\theta_x)\cos(\theta_z) & \cos(\theta_x)\cos(\theta_y) \end{bmatrix}$$

$$\theta_{Lheel\_x} = \text{atan}(-m_{23} / m_{33})$$

$$\theta_{Lheel\_y} = \text{asin}(m_{13})$$

$$\theta_{Lheel\_z} = \text{atan}(-m_{12} / m_{11})$$

$$\text{if } |\theta_{Lheel\_y}| = 90^\circ, \text{ 則 } \theta_{Lheel\_x} = 0^\circ, \theta_{Lheel\_z} = \text{atan}(m_{32} / m_{22})$$

如此依 tree structure 順序對 3 個 1-DOF 的肢節旋轉  $\theta_{Lheel\_x}$ 、 $\theta_{Lheel\_y}$ 、 $\theta_{Lheel\_z}$  就會使 Lheel 的方向與 IP 所求得的 Lh\_Lk 同向。



再來進行的是 Lknee 的旋轉。

在 Neutral Pose 時，Lknee 的三個 local 座標系的座標軸方向都是相同的，x 軸方向是從  $[Lknee]_{Local}$  到  $[Lhip]_{Local}$  (i.e.  $[Lk-Lp]_{Local}$ )，z 軸方向是 Lknee 的轉軸方向，因此可以套用 Sec 3.1 中的正向動力學式子。

$$T_{0-R} \cdot R(\theta_{Lheel\_x}) \cdot R(\theta_{Lheel\_y}) \cdot R(\theta_{Lheel\_z}) \cdot T_{1-R} \cdot R_z(\theta_{Lheel\_z}) \cdot (1,0,0)^T = \frac{[Lk-Lp]_{IP}}{|[Lk-Lp]_{IP}|}$$

$$T_{0-R} \cdot R(\theta_{Lheel\_x}) \cdot R(\theta_{Lheel\_y}) \cdot R(\theta_{Lheel\_z}) \cdot T_{1-R} \cdot R_z(\theta_{Lheel\_z}) \cdot (0,0,1)^T$$

$$= \frac{-[Lh-Lk]_{IP} \times [Lk-Lp]_{IP}}{|-[Lh-Lk]_{IP} \times [Lk-Lp]_{IP}|}$$

$$T_{0-R} \cdot R(\theta_{Lheel\_x}) \cdot R(\theta_{Lheel\_y}) \cdot R(\theta_{Lheel\_z}) \cdot T_{1-R} \cdot R_z(\theta_{Lheel\_z}) \cdot (0,1,0)^T$$

$$= \frac{[Lk-Lp]_{IP} \times (-[Lh-Lk]_{IP} \times [Lk-Lp]_{IP})}{|[Lk-Lp]_{IP} \times (-[Lh-Lk]_{IP} \times [Lk-Lp]_{IP})|}$$

由上述三等式可以求出  $\theta_{Lknee\_z}$ ，因為 constant 矩陣  $T_{0-R}$ 、 $T_{1-R}$  為已知，經過 Lheel 的

旋轉後  $R(\theta_{Lheel\_x}) \cdot R(\theta_{Lheel\_y}) \cdot R(\theta_{Lheel\_z})$  都已知。

因此  $R_z(\theta_{Lknee\_z})$

$$= T_{1-} R^{-1} \cdot R_z^{-1}(\theta_{Lheel\_z}) \cdot R_y^{-1}(\theta_{Lheel\_y}) \cdot R_x^{-1}(\theta_{Lheel\_x}) \cdot T_{0-} R^{-1} \cdot$$

$$\left[ \left( \frac{[Lk-Lp]_{IP}}{|[Lk-Lp]_{IP}|} \right)^T \left( \frac{[Lk-Lp]_{IP} \times (-[Lh-Lk]_{IP} \times [Lk-Lp]_{IP})}{|[Lk-Lp]_{IP} \times (-[Lh-Lk]_{IP} \times [Lk-Lp]_{IP})|} \right)^T \left( \frac{-[Lh-Lk]_{IP} \times [Lk-Lp]_{IP}}{|-[Lh-Lk]_{IP} \times [Lk-Lp]_{IP}|} \right)^T \right]$$

$$R_z(\theta_{Lknee\_z}) = T_{1-} R^{-1} \cdot R_z^{-1}(\theta_{Lheel\_z}) \cdot R_y^{-1}(\theta_{Lheel\_y}) \cdot R_x^{-1}(\theta_{Lheel\_x}) \cdot T_{0-} R^{-1} \cdot$$

knee 旋轉  $\theta_{Lknee\_z}$  後，其肢節方向就會與 IP 求得的肢節方向 Lk\_Lp 同向了。

之後的肢節同以此法類推。



### 3.8. 雙 Root 的角度互換

人類是一種靠雙腳來站立的動物，由於人除了在跳起來的情況下，必定有一隻腳是著地而且固定位置，因此選擇由影像處理所求得的腳部位置較低的那一隻腳當做 Root，另一隻腳則是其中一個 end-effector。由於當人體在運動的時候，可能會用兩隻腳交替做為支撐人體的支點。因此我們將人體模型的定義為，以左腳為 Root 的人體模型和以右腳為 Root 的人體模型是不相同的。

實作流程如下：

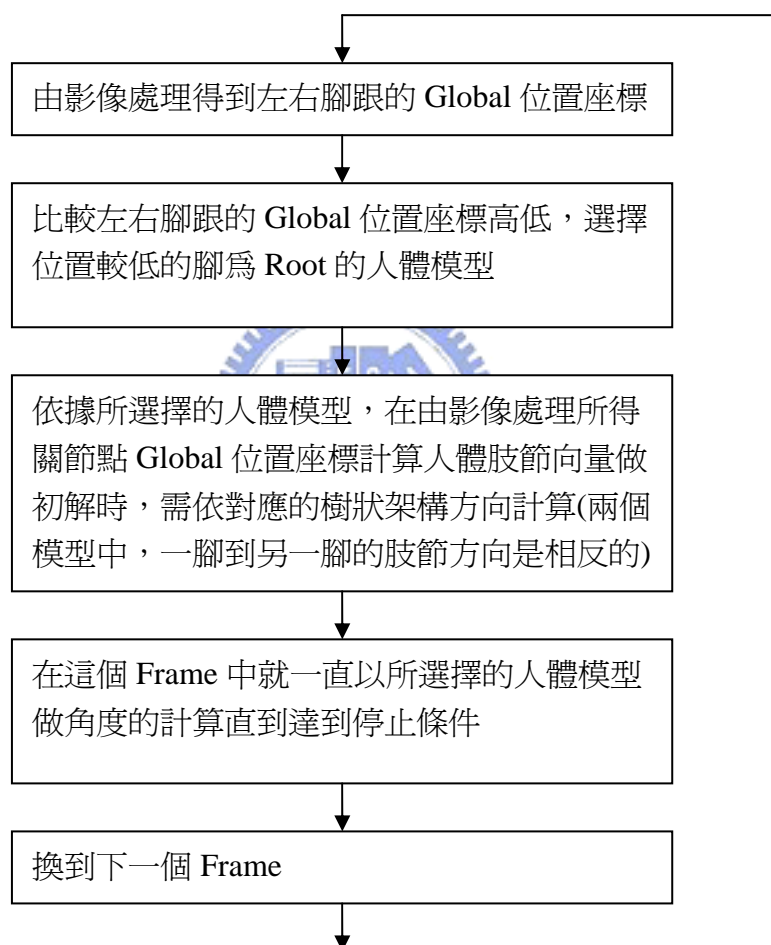


圖 3-16 選擇不同 Root 的人體模型流程圖

若要記錄不同的人體模型下，各關節的旋轉角度，則需先標記是使用那一個人體模型。否則無從知道這組角度是基於那一個人體模型所計算的。

若不以 IP 為初解，而以上一個 frame 的解為初解。

則當上一個 frame 與目前 frame 是使用以不同的腳為 root 的人體模型。

(左右腳高低有變化)

需將上一個 frame 的解(一組 local 角度  $\bar{\theta}$ )，以對應的人體模型利用正向動力學將其表示在 global 座標系中，再將這個人體的姿勢，換成以另一隻腳為 root 的人體模型之 local 角度  $\bar{\theta}'$  來表示。

方法如下：

1. 先以原本的人體模型，求出各 joint 在 global 座標系中的位置。

依 sec3.1 所定義的方式：

將第 i 個 joint 表示在 global 座標系中的位置可寫為：

$$J_i^{global} = T_0 \cdot R_0(\bar{\theta}_0) \cdot T_1 \cdot R_1(\bar{\theta}_1) \cdot \dots \cdot T_i \cdot R_i(\bar{\theta}_i) \cdot (0,0,0)^T$$

以此方式求出所有的 joint 在 global 座標系中的位置。

2. 再來就仿照以 IP 為初解的方式，把這些 joint 在 global 座標系中的位置，當做是由 IP 求得的 joint global 位置，並依對應的樹狀架構方向計算肢節向量(兩個模型中，一腳到另一腳的肢節方向是相反的)，再將以另一腳為 root 的人體模型，旋轉到與這些肢節向量同向，而這個旋轉所需的各關節點 local 角度，即是要求的 local 角度  $\bar{\theta}'$ 。



### 3.9. 逆向動力學中角度變化的演算法

定義：

1. 將某時刻所有關節點的 local angles 表示成一個向量  $\vec{\theta}$   
以這組 local angles 旋轉後的人體模型就表示為：Model( $\vec{\theta}$ )
2. 以 IP 資訊為初解，這組初解就是一組所有關節點的 local angles： $\vec{\theta}_{ip}$

由於每個 frame 都有 IP 資訊為初解，表示為  $\vec{\theta}_{ip}^{(t)}$ ， $t=1\sim m$ 。m 為總 frame 數。

3. 某 frame t 進行 IK iteration 後，每個 iteration k 都會求得一組新的角度變化量  $\Delta\vec{\theta}_k^{(t)}$ ， $k=1\sim r$ 。r 為 frame t 由初解到達停止條件總共所需的 iteration 數。

在 frame t 中，令 iteration k 時，原本所有關節點的 local angles 為  $\dot{\vec{\theta}}_k^{(t)}$ 。

以 iteration k 時，Model( $\dot{\vec{\theta}}_k^{(t)}$ )所求得的角度變化量  $\Delta\vec{\theta}_k^{(t)}$  加到  $\dot{\vec{\theta}}_k^{(t)}$  中，

得到 iteration k 的新的 local angles 為  $\vec{\theta}_k^{(t)} = \dot{\vec{\theta}}_k^{(t)} + \Delta\vec{\theta}_k^{(t)}$ 。

並將  $\vec{\theta}_k^{(t)}$  做為下一個 iteration 的  $\dot{\vec{\theta}}_k^{(t)}$ 。

最後經過 r 次 iteration 達到停止條件後，此時的 local angles 就是本 frame t 的解；

令為  $\vec{\theta}_{sol}^{(t)} = \vec{\theta}_r^{(t)}$ 。

$$\vec{\theta}_r^{(t)} = \vec{\theta}_{ip}^{(t)} + \sum_{k=1}^r \Delta\vec{\theta}_k^{(t)}$$

這就是角度變化量的累計。

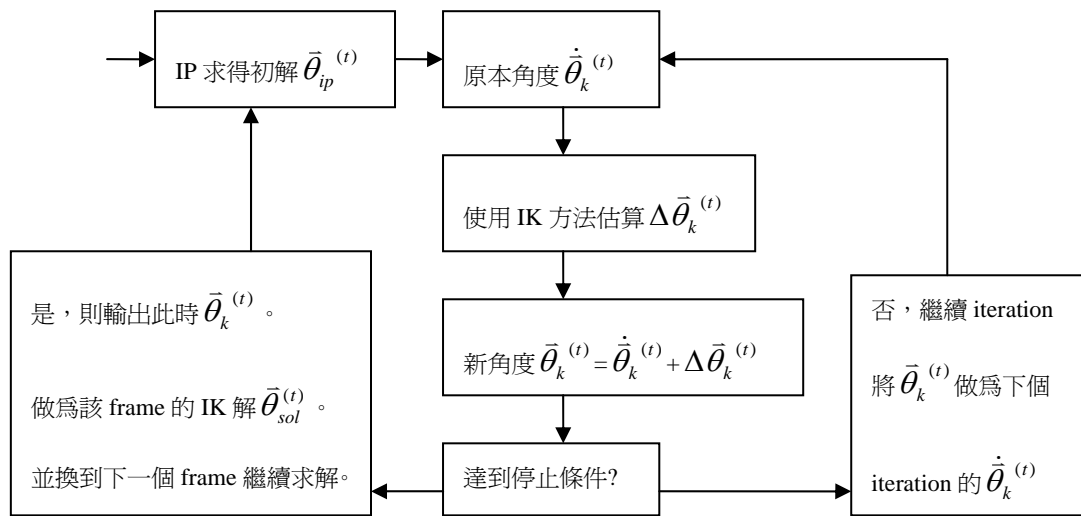


圖 3-17 角度變化之流程圖

演算法：

For each frame  $t$  ( $t = 1 \sim m$ 。  $m$  為總 frame 數。)

{

由 frame  $t$  的 IP 資訊得到一組所有關節點的 local angles 初解： $\bar{\theta}_{ip}^{(t)}$

While(未到停止條件)

{

iteration 1 時，原本所有關節點的 local angles 為  $\dot{\theta}_1^{(t)} = \bar{\theta}_{ip}^{(t)}$

每次 iteration  $k$  時，就以  $\text{Model}(\dot{\theta}_k^{(t)})$  所求得的角度變化量  $\Delta \bar{\theta}_k^{(t)}$  加到  $\dot{\theta}_k^{(t)}$  中

得到 iteration  $k$  的新的 local angles 為  $\bar{\theta}_k^{(t)} = \dot{\theta}_k^{(t)} + \Delta \bar{\theta}_k^{(t)}$ 。

並將  $\bar{\theta}_k^{(t)}$  做為下一個 iteration 的  $\dot{\theta}_k^{(t)}$ 。

}

(做了  $r$  個 iteration 達到停止條件後)

輸出此時的  $\bar{\theta}_k^{(t)} = \bar{\theta}_r^{(t)}$ 。就是這個 frame  $t$  的逆向動力學解了。解： $\bar{\theta}_{sol}^{(t)} = \bar{\theta}_r^{(t)}$ 。

}

## 4、 逆向動力學 ( Inverse Kinematics )

### 4.1. Jacobian 定義

要利用人體的模型中的 Local 角度來使 end-effectors 的位置到達所希望到達的 Target。Local 角度的數量很多，但是 end-effector 比較少；因此這是一個解 non-linear system 的問題。因此採取最普遍用來解 non-linear solution 的方法，也就是 Jacobian。先定義 Jacobian Matrix 如下：

Jacobian Matrix 定義:

$\bar{e}_i = [e_{ix}, e_{iy}, e_{iz}]^T, i = 1, \dots, m$  為共  $m$  (本論文中  $m=4$ ) 個 end-effector 所希望到達的世界座標系上的位置。

$\bar{\theta} = [\theta_1, \theta_2, \dots, \theta_n]^T$  為 root 到 end-effector 所經過的所有轉軸要轉到 target 所需轉的角度。

$$\bar{e}_i = F_i(\bar{\theta}) = F_i(\theta_1, \theta_2, \dots, \theta_n), i = 1, \dots, m$$

假設  $\bar{\theta}_k$  是目前第  $k$  個 iteration 的  $\bar{\theta}$  的近似解，為了求取更好的下一個解  $\bar{x}_{k+1}$ ；

$$\text{令 } \bar{\theta} = \bar{\theta}_k + d\bar{\theta}, \quad d\bar{\theta} = [d\theta_1, d\theta_2, \dots, d\theta_n]^T$$

$$\text{則 } \bar{e}_i = F_i(\bar{\theta}) = F_i(\bar{x}_k + d\bar{\theta}), i = 1, \dots, m$$

由全微分(total differential)可知  $F_i(\bar{\theta}_k + d\bar{\theta}) \approx F_i(\bar{\theta}_k) + dF_i(\bar{\theta})$ ，where

$$dF(\bar{\theta}) = F_{x_1}(\bar{\theta})d\theta_1 + F_{x_2}(\bar{\theta})d\theta_2 + \dots + F_{x_n}(\bar{\theta})d\theta_n$$

$$\rightarrow \bar{e}_i = F_i(\bar{\theta}_k + d\bar{\theta}) = F_i(\bar{\theta}_k) + \frac{\partial F_i(\bar{\theta}_k)}{\partial \theta_1} d\theta_1 + \frac{\partial F_i(\bar{\theta}_k)}{\partial \theta_2} d\theta_2 + \dots + \frac{\partial F_i(\bar{\theta}_k)}{\partial \theta_n} d\theta_n, i = 1, \dots, m$$

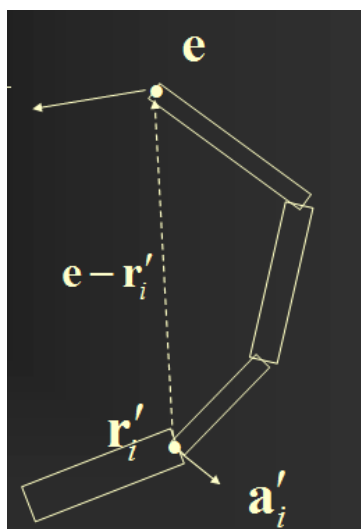
如此即為一個 linear system，其 matrix form 可表示為：

$$\begin{bmatrix} \frac{\partial F_1(\bar{\theta}_k)}{\partial \theta_1} & \frac{\partial F_1(\bar{\theta}_k)}{\partial \theta_2} & \dots & \frac{\partial F_1(\bar{\theta}_k)}{\partial \theta_n} \\ \frac{\partial F_2(\bar{\theta}_k)}{\partial \theta_1} & \frac{\partial F_2(\bar{\theta}_k)}{\partial \theta_2} & \dots & \frac{\partial F_2(\bar{\theta}_k)}{\partial \theta_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial F_m(\bar{\theta}_k)}{\partial \theta_1} & \frac{\partial F_m(\bar{\theta}_k)}{\partial \theta_2} & \dots & \frac{\partial F_m(\bar{\theta}_k)}{\partial \theta_n} \end{bmatrix} \begin{bmatrix} d\theta_1 \\ d\theta_2 \\ \vdots \\ d\theta_n \end{bmatrix} = \begin{bmatrix} \bar{e}_1 \\ \bar{e}_2 \\ \vdots \\ \bar{e}_m \end{bmatrix} - \begin{bmatrix} F_1(\bar{\theta}_k) \\ F_2(\bar{\theta}_k) \\ \vdots \\ F_m(\bar{\theta}_k) \end{bmatrix}$$

，即  $F'(\bar{\theta}_k)d\bar{\theta} = \bar{e} - F(\bar{\theta}_k)$ ，而令  $F'(\bar{\theta}_k) = J$ ， $J$  即為 Jacobian Matrix。

而  $\vec{e}_i = F_i(\vec{\theta}) = F_i(\theta_1, \theta_2, \dots, \theta_n), i = 1, \dots, m$  這個函數  $F$  也就是  $\vec{e}_i$  與  $\theta$  的關係就等於是利用 local 角度 ( $\vec{\theta}$ ) 求出 end-effector 在 global 座標系上的座標值 ( $\vec{e}_i$ ) 的關係。

而由於  $\vec{e}_i = F_i(\vec{\theta}) = F_i(\theta_1, \theta_2, \dots, \theta_n), i = 1, \dots, m$ ；這個函數  $F$  也就是  $Y$  與  $\theta$  的關係就等於是利用 local coordinate 的角度 ( $\vec{\theta}$ ) 求出 end-effector 在 global coordinate 的座標值 ( $\vec{e}$ ) 的關係；因此在求取 Jacobian Matrix 中的各個偏微分項時，就等於是在求取各個轉軸轉動微小的角度量，對 end-effector 之 global position 所造成的位移影響。由圖 4.1 我得到以下結論：



$a_i'$ : 在世界座標系中單位長的旋轉軸  
 $r_i'$ : 關節點在世界座標系中的位置  
 $e$ : end-effector 在世界座標系中的位置

$$\frac{\partial F_i(\theta_k)}{\partial \theta_j} = a_i' \times (e - r_i')$$

圖 4-1 Jacobian 示意圖

將所要偏微分的該轉軸取 unit length，如此一來求 cross product 的時候就只剩下  $e - r_i'$  的部份會影響其 cross product 的長度。



詳細證明：

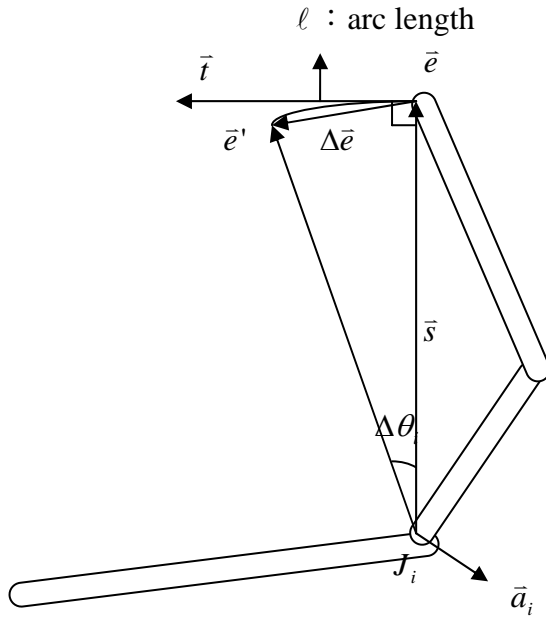


圖 4-2 Jacobian 證明示意圖

Assume there are  $n$  joints  $J_1 \sim J_n$ , whose unit length axis vectors are  $\bar{a}_1 \sim \bar{a}_n$ , pivot positions  $\bar{r}_1 \sim \bar{r}_n$ , rotation angles  $\theta_1 \sim \theta_n$ . ( $\bar{a}_1 \sim \bar{a}_n$ ,  $\bar{r}_1 \sim \bar{r}_n$  are represented in WCS) Then the end-effectors' positions in WCS can be represented as a function of  $x_1 \sim x_n$   
 $\bar{e} = F(\theta_1, \theta_2, \dots, \theta_n)$

Consider if we change the  $i$ -th joint's rotation angle for  $\Delta x_i$ , the end-effector will move to a new position  $\bar{e}'$ ; The arc length from  $\bar{e}$  to  $\bar{e}'$  is  $\ell$ .

Let  $\bar{s} = \bar{e} - \bar{r}_i$  and  $\bar{t} = \bar{a}_i \times \bar{s}$

Since  $|\bar{a}_i| = 1$  ( $\bar{a}_1 \sim \bar{a}_n$  are unit length axis vectors),  $|\bar{t}| = |\bar{a}_i| \cdot |\bar{s}| \cdot \sin \angle(\bar{a}_i, \bar{s}) = |\bar{s}|$ .

When  $\Delta x_i \rightarrow 0$ ,  $|\Delta \bar{e}| \cong \ell$  (長度相似) and  $\frac{\Delta \bar{e}}{|\Delta \bar{e}|} = \frac{\bar{t}}{|\bar{t}|}$  (方向相似)

$$\Rightarrow \Delta \bar{e} = \frac{\bar{t}}{|\bar{t}|} \cdot |\Delta \bar{e}| = \bar{t} \cdot \frac{\ell}{|\bar{s}|} = \bar{t} \cdot \frac{|\bar{s}| \cdot \Delta \theta_i}{|\bar{s}|} = \bar{t} \cdot \Delta \theta_i$$

$$\Rightarrow \lim_{\Delta \theta_i \rightarrow 0} \frac{\Delta \bar{e}}{\Delta \theta_i} = \bar{t} \Rightarrow \frac{\partial \bar{e}}{\partial \theta_i} = \bar{t}, \text{ i.e. } \frac{\partial F(\theta_1, \theta_2, \dots, \theta_n)}{\partial \theta_i} = \bar{a}_i \times (\bar{e} - \bar{r}_i)$$

## 4.2. Least square method (pseudo inverse method)

目的: Minimize  $\|J\Delta\theta - \bar{e}\|$ 。

也就是要 End-effector 最接近 target。

而要 Minimize  $\|J\Delta\theta - \bar{e}\|$  等於是要使  $\|J\Delta\theta - \bar{e}\| = 0$ ,

也就是要  $J\Delta\theta = \bar{e}$ 。

我們想知道  $\Delta\theta$  的量,

但因  $\Delta\theta$  與  $\bar{e}$  的 Rank 不同, 因此不能直接取 inverse,

而要求其 pseudo inverse:  $\Delta\theta = J^\Psi \bar{e}$

而  $J\Delta\theta = \bar{e}$  由矩陣的 normal equation 就可以寫成  $J^T J\Delta\theta = J^T \bar{e}$

由於  $J^T J$  為 non-singular 所以 invertible

因此  $\Delta\theta = (J^T J)^{-1} J^T \bar{e}$

又  $(J^T J)^{-1} J^T = J^{-1} (J^T)^{-1} J^T = J^{-1} = J^T (J^T)^{-1} J^{-1} = J^T (JJ^T)^{-1}$

所以  $\Delta\theta = J^T (JJ^T)^{-1} \bar{e}$

但是此法在接近 Singularity 的時候效果很差, 會振盪。

## 4.3. Damped least square method

目的: Minimize  $\|J\Delta\theta - \bar{e}\|^2 + \lambda^2 \|\Delta\theta\|^2$ ,  $\lambda$  為 damping constant。

而要 Minimize  $\|J\Delta\theta - \bar{e}\|^2 + \lambda^2 \|\Delta\theta\|^2$

也就是等同於要 Minimize  $\left\| \begin{pmatrix} J \\ \lambda I \end{pmatrix} \Delta\theta - \begin{pmatrix} \bar{e} \\ 0 \end{pmatrix} \right\|$

又等同於要使  $\begin{pmatrix} J \\ \lambda I \end{pmatrix} \Delta\theta = \begin{pmatrix} \bar{e} \\ 0 \end{pmatrix}$

而矩陣的 normal equation 就可以寫成  $\begin{pmatrix} J \\ \lambda I \end{pmatrix}^T \begin{pmatrix} J \\ \lambda I \end{pmatrix} \Delta\theta = \begin{pmatrix} J \\ \lambda I \end{pmatrix}^T \begin{pmatrix} \bar{e} \\ 0 \end{pmatrix}$

並可重寫為  $(J^T J + \lambda^2 I) \Delta\theta = J^T \bar{e}$

由於  $J^T J + \lambda^2 I$  為 non-singular 所以 invertible

因此  $\Delta\theta = (J^T J + \lambda^2 I)^{-1} J^T \bar{e}$

又  $(J^T J)^{-1} J^T = J^{-1} (J^T)^{-1} J^T = J^{-1} = J^T (J^T)^{-1} J^{-1} = J^T (JJ^T)^{-1}$

所以  $\Delta\theta = J^T (JJ^T + \lambda^2 I)^{-1} \bar{e}$

而為了使計算這個  $\Delta\theta = J^T (J^T J + \lambda^2 I)^{-1} \bar{e}$  更方便

使用了 Singular Value Decomposition

先將 J 做 SVD 表示為  $J = UDV^T$

又可再進一步寫成

$$J = \sum_{i=1}^m \sigma_i u_i v_i^T = \sum_{i=1}^r \sigma_i u_i v_i^T \quad (\text{The rank of } J \text{ is equal to the largest value } r \text{ such that } \sigma_i \neq 0.)$$

再來 J 的 pseudo-inverse 則可寫為

$$J^\Psi = VD^\Psi U^T$$

由於  $D^\Psi = (d^\Psi_{i,j})$

$$d^\Psi_{i,i} = \begin{cases} 1/d_{i,i}, & d_{i,i} \neq 0 \\ 0, & d_{i,i} = 0 \end{cases}$$

$$\text{因此, } J^\Psi = \sum_{i=1}^r \sigma_i^{-1} v_i u_i^T$$

$$JJ^T + \lambda^2 I = (UDV^T)(VD^T U^T) + \lambda^2 I = U(DD^T + \lambda^2 I)U^T$$

$$J^T (JJ^T + \lambda^2 I)^{-1} = VD^T (DD^T + \lambda^2 I)^{-1} U^T = VEU^T$$

$$E = (e_{i,j}), e_{i,i} = \frac{\sigma_i}{\sigma_i^2 + \lambda^2}.$$

$$J^T (JJ^T + \lambda^2 I)^{-1} = \sum_{i=1}^r \frac{\sigma_i}{\sigma_i^2 + \lambda^2} v_i u_i^T$$

再來計算  $\Delta\theta = J^T (J^T J + \lambda^2 I)^{-1} \bar{e}$  就更方便了。



$\lambda$  : damping constant 的設定會影響  $\Delta\theta$  的量；

由於  $\sigma_i$  在接近 Singularity 的時候會極小，而造成  $\sigma_i^{-1}$  的值很大，使得  $\Delta\theta$  的量過大。

因此加入  $\lambda$  這個 damping constant 來平衡這個不正確的量，

但  $\lambda$  如果設定的過小，那也不足以縮小這個不正確的  $\sigma_i^{-1}$ ；

又  $\lambda$  如果設定的過大，那  $\sigma_i^{-1}$  的量就會變得很小，使得  $\Delta\theta$  很小，那就要收斂很久，也

不是一個好方法；因此  $\lambda$  的設定很重要。

另外，為了防止 target 距離過遠時，人體模型會伸直造成 Singularity 而產生 jitter 的現象，為了避免這個問題，希望 target 能近一點，於是改變了

$\bar{e} = \bar{t}$  (target position) -  $\bar{s}$  (end-effector position) 的設定。而改為

$$e_i = \begin{cases} t_i - s_i, & \text{if } \|t_i - s_i\| \leq D_{\max} \\ D_{\max} \frac{t_i - s_i}{\|t_i - s_i\|}, & \text{otherwise} \end{cases}$$

$D_{\max}$  為從 IK iteration 前到目前的 iteration，end-effectors 移動的總量。

## 4.4. Selectively damped least square method

在上一節中，提到了  $\lambda$  : damping constant 設定的重要性。

因此在 Damped Least Square Method 上，針對每一個 Joint 每一個 iteration 都去更新它的  $\lambda$  : damping constant 值，以達到更有效的  $\Delta\theta$  之 Update。

與 DLS 的方式相同，先計算出目前的 Jacobian Matrix 並對其做 SVD，即  $J = UDV^T$ 。

將 End-effector 希望的改變量  $\bar{e}$  以 U 的 columns 之線性組合表示， $\bar{e} = \sum_i \alpha_i u_i$ ，where

$$\alpha_i = u_i^T \bar{e}。$$

$\Delta\theta$  也如同 DLS 的方式表示成  $\Delta\theta = J^\Psi \bar{e} = \sum_{i=1}^r \tau_i v_i u_i^T \bar{e}$ ，再將  $\alpha_i = u_i^T \bar{e}$  代入，

則  $\Delta\theta = \sum_{i=1}^r \alpha_i \tau_i v_i$ ， $\tau_i$  為 scalar。

SDLS 的目的是要針對每個 Joint 去計算其  $\alpha_i u_i$  對  $\bar{e}$  的影響。因此針對每個 i，去計算 U

的 i-th column 的 Magnitude 的總和  $N_i = \sum_{j=1}^k \|u_{j,i}\|$ ，where  $u_i = (u_{1,i}^T, \dots, u_{k,i}^T)^T$  with each

$$u_{j,i} \in R^3。$$

另外，假設 D 的第 i 個對角項不為 0， $v_{j,i}$  為 V 的第 j 列第 i 行；以及求出對第 j 個 joint

旋轉微小角度時，第  $\ell$  個 end-effector 相對應的改變 Magnitude 量  $\rho_{\ell,j} = \|\partial s_\ell / \partial \theta_j\|$ ，那麼

Local angle 做了改變時，對第  $\ell$  個 end-effector 相對應的移動量就等於是

$M_{i,\ell} = \sigma_i^{-1} \sum_{j=1}^n |v_{j,i}| \rho_{\ell,j}$ 。可以想像以上這個動作就像是 pseudo-inverse method 作用在

$\bar{e} = u_i$  時，其  $\Delta\theta = \sigma_i^{-1} v_{j,i}$ ，可估算其使第  $\ell$  個 end-effector 移動的距離就是  $\sigma_i^{-1} |v_{j,i}| \rho_{\ell,j}$ 。

進而統計對所有的 end-effectors 的位置改變量  $M_i = \sum_\ell M_{i,\ell}$ 。

當  $M_i > N_i$  時代表 joints 的角度變化使得 end-effectors 的位置朝 target 的反方向行走，太多的這種情形會造成 Singularity 的發生，因此設定  $\gamma_i = \min(1, N_i / M_i) \cdot \gamma_{\max}$ ； $\gamma_{\max}$  為一個 iteration 中一個 joint 的 angle 最大允許改變量，如此  $\gamma_i$  就會被控制在 End-effector 希望的

改變量  $\bar{e}$  以 U 的 columns 之線性組合表示， $\bar{e} = \sum_i \alpha_i u_i$  的最大角度改變量中。

$$\text{再令 } \varphi_i = \text{ClampMaxAbs}(\sigma_i^{-1} \alpha_i v_i, \gamma_i) = \begin{cases} \sigma_i^{-1} \alpha_i v_i, & \text{if } |\sigma_i^{-1} \alpha_i v_i| \geq |\gamma_i| \\ \gamma_i, & \text{if } |\sigma_i^{-1} \alpha_i v_i| < |\gamma_i| \end{cases}$$

$$\text{最後 } \Delta\theta = \text{ClampMaxAbs}(\sum_i \varphi_i, \gamma_{\max}) = \begin{cases} \sum_i \varphi_i, & \text{if } |\sum_i \varphi_i| \geq |\gamma_{\max}| \\ \gamma_{\max}, & \text{if } |\sum_i \varphi_i| < |\gamma_{\max}| \end{cases},$$

$\Delta\theta$  就是本次 iteration 要 Update 的 Local angle 量。

#### 4.5. 角度更新方式

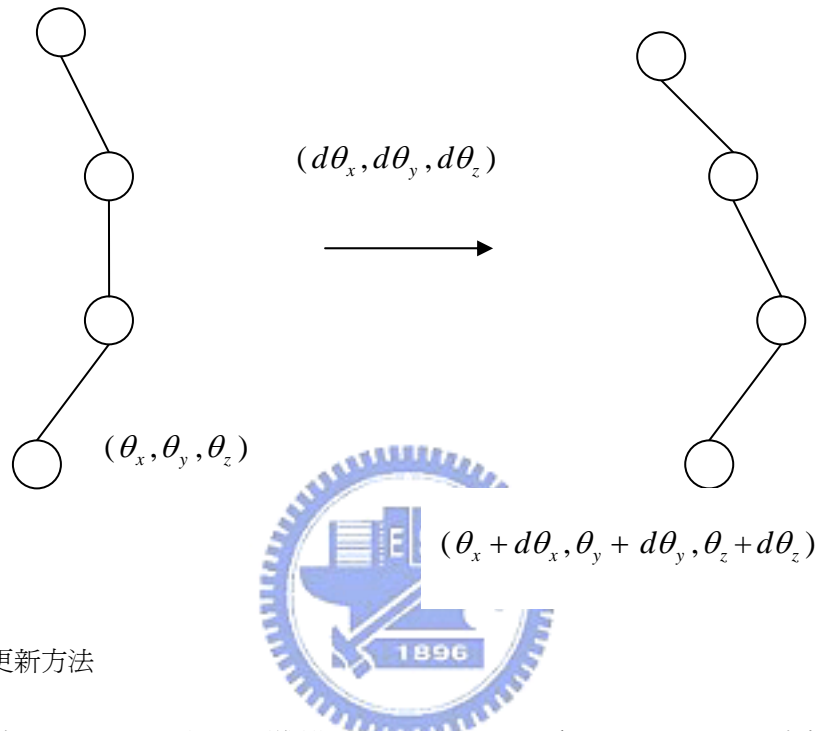


圖 4-3 角度更新方法

如圖 4-3，在 iteration k 時，人體模型以  $(\theta_x, \theta_y, \theta_z)$  為 initial solution 求解逆向動力學。

此時會得到一組解  $(d\theta_x, d\theta_y, d\theta_z)$ ，而這組解的資訊(包含在計算逆向動力學時所要用到的，每個 Joint 的 global position 與每個轉軸的方向)，都是依據人體模型  $(\theta_x, \theta_y, \theta_z)$  所求出來的。

而不是逐個角度更新後，以更新後的人體模型再重新計算  $d\theta$ 。

實際要更新角度的時候，就直接將  $(d\theta_x, d\theta_y, d\theta_z)$  加到  $(\theta_x, \theta_y, \theta_z)$  上，

使 iteration k+1 時，人體模型更新為  $(\theta_x + d\theta_x, \theta_y + d\theta_y, \theta_z + d\theta_z)$

## 4.6. 加入各關節角度限制

將目前的人體模型的 Local 角度定義在 World Coordinate System 的 Reference 座標系所採取的 Fixed Angle Rotation Representation 累算後的值；

而這些 Local 角度卻有可能是人體不可能達到的姿勢，

因此我們分別對這些 Local 角度設定 Constraints。

使這些累算後的 Local 角度不要超過某個範圍。

Constraint 分為上限與下限，分別使得角度不能過大或是過小。

並在 IK iteration 的時候檢查是否在這些限制中，

來幫助我們排除逆向動力學的多重解。

或是檢驗 IP(影像處理)所得到的角度是否有問題。

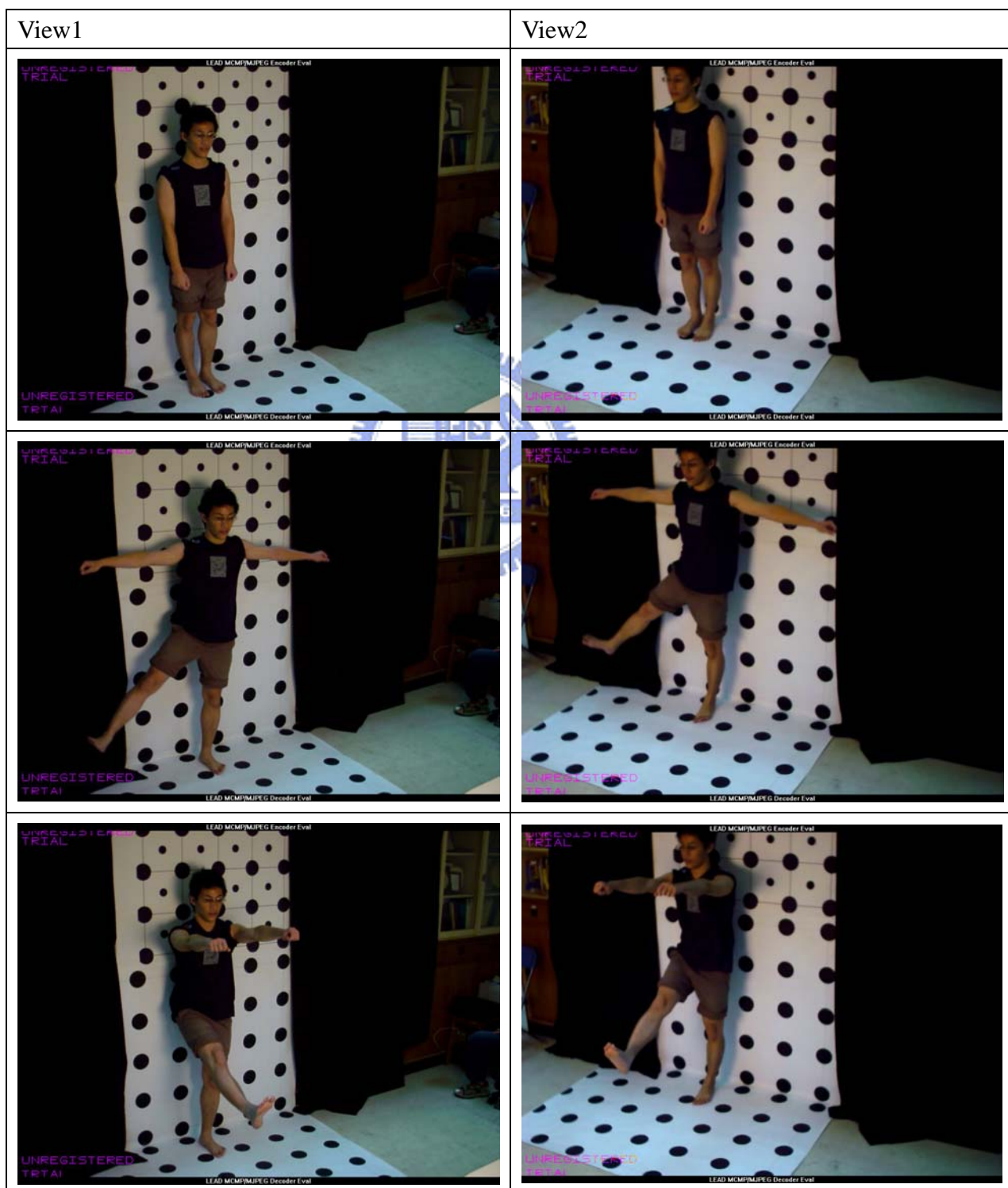
旋轉軸	Lower Bound	Upper Bound
Lheel-x-rot	-30	30
Lheel-y-rot	-10	10
Lheel-z-rot	-30	30
Lknee-rot	0	120
Lhip-x-rot	-10	80
Lhip-y-rot	-20	20
Lhip-z-rot	-10	80
Navel-rot	-80	0
Neck-rot	-30	10
Lshoulder-x-rot	-20	150
Lshoulder-y-rot	-30	30
Lshoulder-z-rot	-180	20
Lelbow-rot	0	100
Rshoulder-x-rot	-20	150
Rshoulder-y-rot	-30	30
Rshoulder-z-rot	-20	180
Relbow-rot	0	100
Rhip-x-rot	-10	80
Rhip-y-rot	-20	20
Rhip-z-rot	-10	80
Rknee-rot	-120	0

表格 1 各關節自由度角度限制表

## 5、 實驗流程與方法

### 5.1.實驗輸入資料

#### 5.1.1. 立體影像資料





## 5.1.2. End-Effectors 3D 座標

由立體影像進行三維座標重建出 End-Effectors 的 3D 座標。

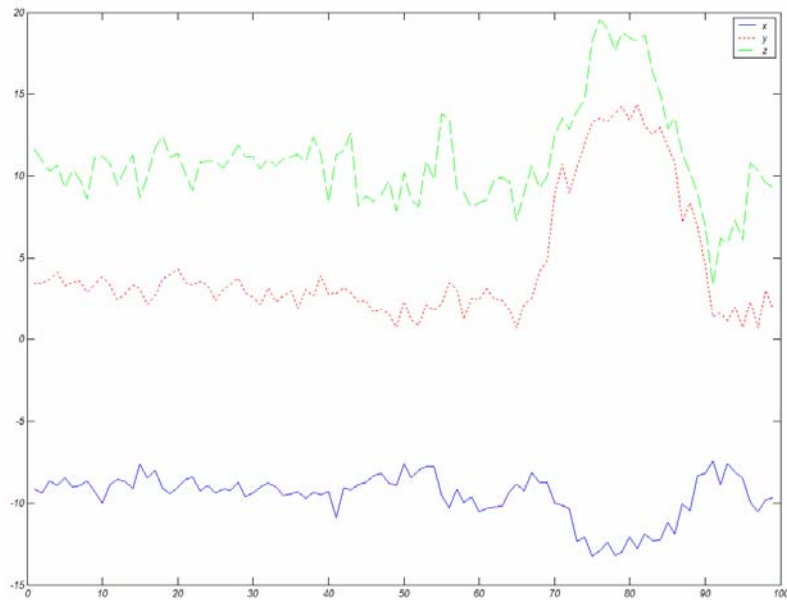


圖 5-1 左腳跟(Lheel)3D 座標圖

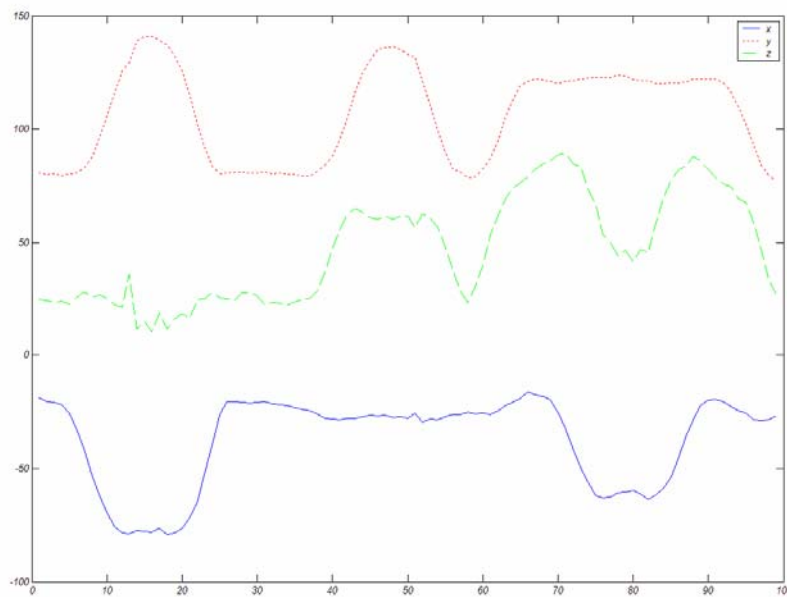


圖 5-2 左手腕(Lwrist)3D 座標圖



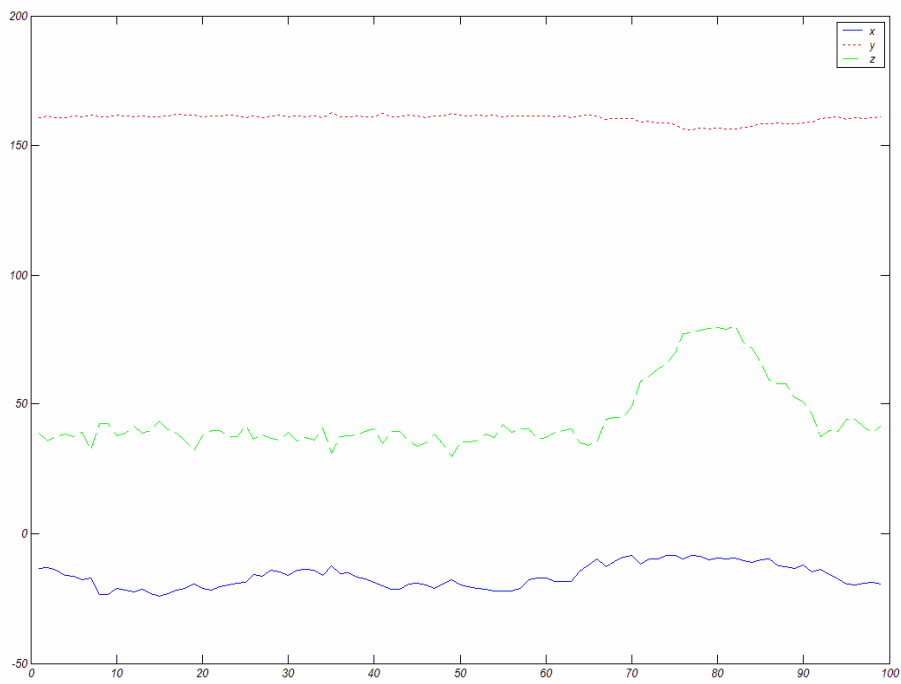


圖 5-3 頭部(Head)3D 座標圖

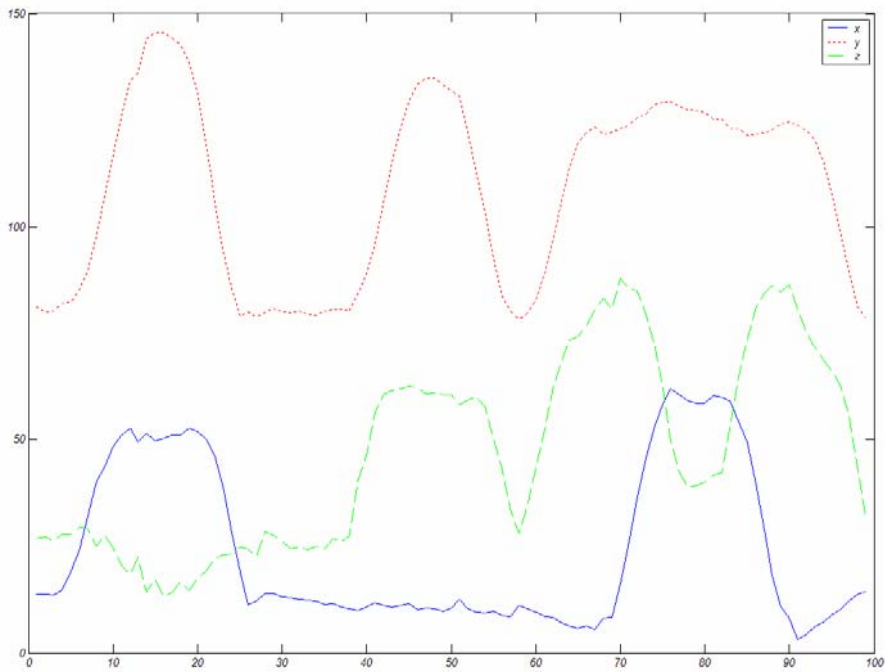


圖 5-4 右手腕(Rwrist)3D 座標圖

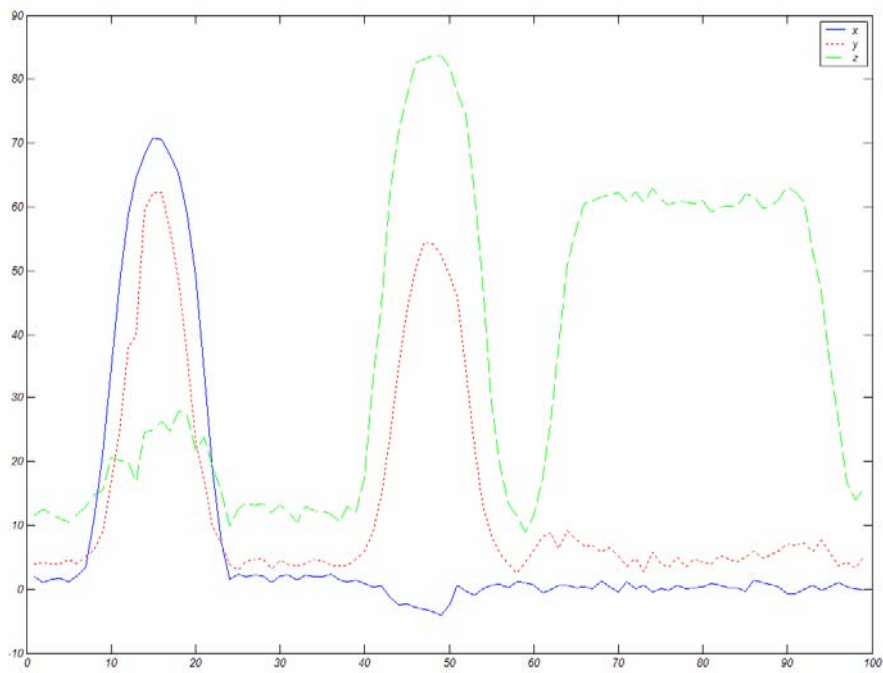


圖 5-5 右腳跟(Rheel)3D 座標圖



## 5.2. 實驗流程

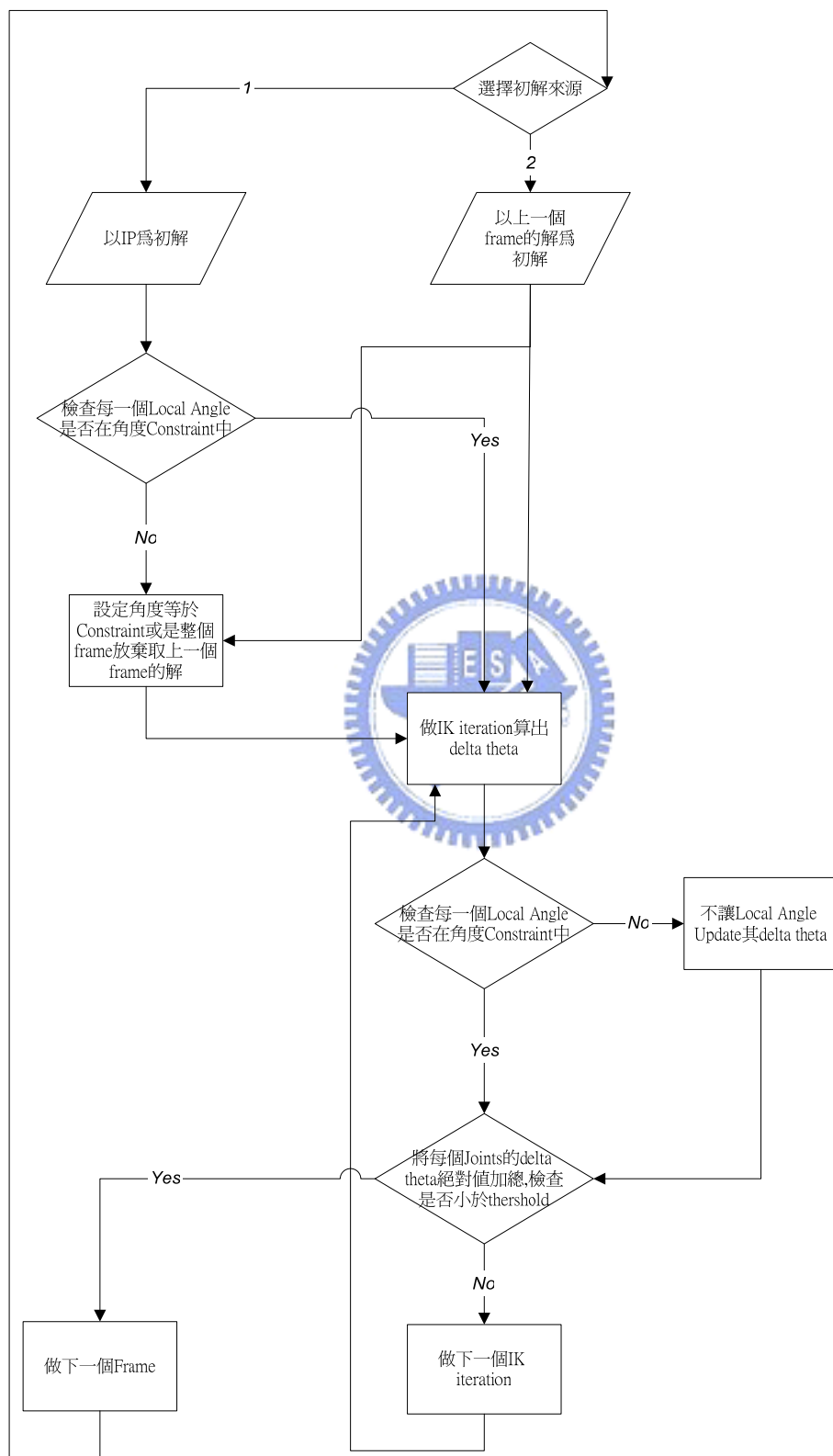


圖 5-6 實驗流程圖

## IP-Based 的 IK 方法

For All Frames

{

取得 IP 所求出的 End-effector 位置與各 Joint Vectors ;  
以這些 Joint Vectors 求出所有的 Joints 的 Local Angles ;

For All Joints

{

if (Joint 的 Local Angle < physical constraint 的下限值)  
此 Joint 的 Local Angle = physical constraint 的下限值 ;  
if (Joint 的 Local Angle > physical constraint 的上限值)  
此 Joint 的 Local Angle = physical constraint 的上限值 ;

}

While(總  $\Delta\theta$  < 門檻值)

{

以這組 Local angles 套入我們定義的人體肢節長度做 FK, 這就是我們的 initial solution ;  
以我們的 initial solution 中各 joint 的旋轉軸單位向量  $\bar{a}$  與各 Joint 到 end-effector 的向量

$(\bar{e} - \bar{r}_i)$ , 求取 Jacobian Matrix  $\frac{\partial F(\theta_1, \theta_2, \dots, \theta_n)}{\partial \theta_i} = \bar{a}_i \times (\bar{e} - \bar{r}_i)$  ;

再以 IP 所求出的 End-effector 位置為 Target, 看目前 initial solution 的狀態下, end-effector 到 target 的向量為多少估出各 Joints 的 Local Angles 應該再 Update 的  $\Delta\theta$  有多少 ;

For All Joints

{

Joint 的 Local Angle New  $\theta =$  Local Angle Original  $\theta +$  要 Update 的  $\Delta\theta$   
if (Joint 的 Local Angle < physical constraint 的下限值)  
此 Joint 的 Local Angle New  $\theta =$  Joint Angle Original  $\theta$  ;  
if (Joint 的 Local Angle > physical constraint 的上限值)  
此 Joint 的 Local Angle New  $\theta =$  Local Angle Original  $\theta$  ;

}

將 All Joints 的  $\Delta\theta$  絕對值加起來, 計算總  $\Delta\theta$  ;

}

}

### 5.3. 停止條件的選擇

在求解 IK 時，是要經過許多的 iteration 才能夠到達目標解的。

可以做為求解停止條件的變數有三種：

1. Position Error 值
2.  $\sum_{j=1 \sim n} |\Delta\theta_j^i|$  值
3. iteration 數

其使用的方式、理由與優劣點如下：

1. Position Error 值

Position Error 值 = |End-effectors Position – Target Position|

也就是以離目標的距離是否小於某個固定的 threshold 做為停止條件。

因為 IK 的目的就是要使 End-effectors 達到 Target，

所以此法最為直覺也最常使用，但必須要保證 End-effectors 一定都可以到位才能使用；

因為此法的缺點就是如果其實 End-effectors 根本就到不了 Target；就會永遠無法停止 IK iteration。



2.  $\sum_{j=1 \sim n} |\Delta\theta_j^i|$  值

令在第 i 個 iteration 的第 j 個 joint 的角度變化量為  $\Delta\theta_j^i$

在一次 iteration i 中的總角度變化量 =  $\sum_{j=1 \sim n} |\Delta\theta_j^i|$ ，n 為 joints 的總數量(本論文的人體模型 n=21)。

就是該次 iteration i 中所有 joints 的 local angles  $\theta_j$  的總變化量，

由於每次的 IK iteration 中，就是在求取各 local angles  $\theta_j$  所需要的變化量  $\Delta\theta_j^i$ ，Update

在原本的 Model 上；因此當 local angles  $\theta_j$  沒有變化的時候，即可停止 IK iteration，

$\sum_{j=1 \sim n} |\Delta\theta_j^i|$  雖不會完全等於 0，但卻會是一個單調的收斂，因此以 local angles  $\theta_j$  的總變化量  $\sum_{j=1 \sim n} |\Delta\theta_j^i|$  是否小於某個固定的 threshold 做為停止條件是合理的。

此法的缺點就是不保證 End-effectors 是否到得了 Target。

但這同時也是他的優點，若是 End-effectors 根本就到不了 Target，也會在 local angles  $\theta_j$

沒有變化的時候停止。

### 3. iteration 次數

所有 iteration 演算法的最基本停止條件。

以 iteration 次數是否到達某個固定值做為停止條件。

非常直觀，但是是最差的一個方法，既不保證 End-effectors 是否到得了 Target；也不管 Model 的 local angles 是否還有在 Update。

唯一的好處就是，次數到了就一定會停止，不會做無窮的 iteration。

結論是：

1. 以 Position Error 值做為停止條件，看不出內部角度的變化，故這只是必要條件，非充分條件。
2. 純看 iteration 數，只是為了不收殮，做不完，並無意義。
3. 以  $\sum_{j=1 \sim n} |\Delta\theta_j^i|$  值做為停止條件，才能充份反映解的變化。

在本論文的實驗中，都採用以  $\sum_{j=1 \sim n} |\Delta\theta_j^i|$  值  $< 1^\circ$  為停止條件。由於在  $\sum_{j=1 \sim n} |\Delta\theta_j^i|$  值  $< 1^\circ$  的時候，在能夠到達 target 的情況下，Position error 值平均已能夠小於 4cm，共有四個 end-effectors，就是平均一個 end-effector 的 position error 已能夠小於 1cm，如此是我們可以接受的範圍。

在以不同  $\lambda$  值比較  $\Delta\theta$  的收斂速度的實驗中，由於只需比較速度，於是採用以  $\sum_{j=1 \sim n} |\Delta\theta_j^i|$  值  $< 1^\circ$  且 position error  $< 3\text{cm}$  為停止條件，共有四個 end-effectors，就是平均一個 end-effector 的 position error 已能夠小於 0.75cm。並選擇能夠到達 target 的情況來做實驗；如此就可在 End-effectors 已經到達一定的精確度，且 Model 的 Local Angles 不再 Update 的情況下，知道所需花的 iteration 次數如何了。

## 6、 實驗數據及結果

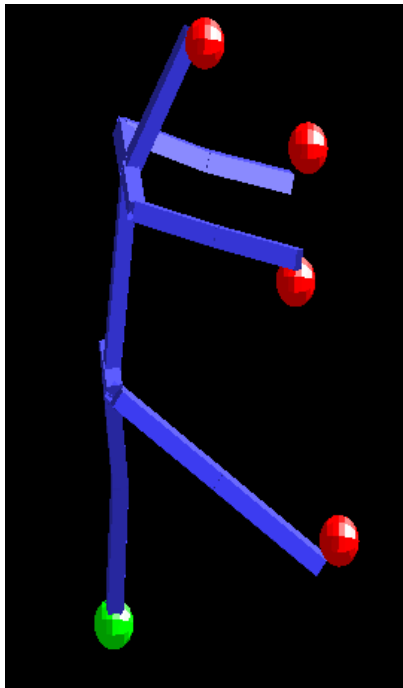
### 6.1.比較有無 **Constraint** 的差異

由於本論文的實驗主題是人體的模型，因此必須要針對人體的特性做設定，人體的關節並非是能完全沒有限制的轉動的，因此必須要設定角度的限制，以防止人體模型做出人體不可能做到的姿勢。若沒有加入角度的限制，逆向動力學只會盲目地將所有的 end-effectors 轉到 target，此時有可能會使一些關節做了做不到的轉動，因此我們比較是否加入角度的限制，是否會造成一些關節做了做不到的轉動。

同樣以 IP 為初解，分別以有 **Constraint** 與無 **Constraint** 的條件下，解決 **IK** 問題；實驗結果如下：



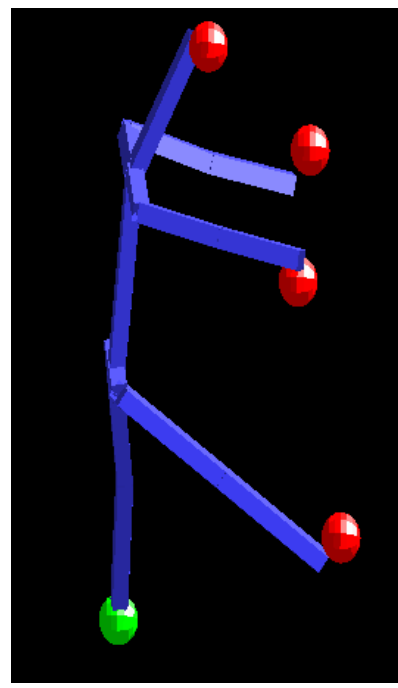
With Constraint



Initial Solution 時各關節角度：

Lheel x-rot	-3.71064
Lheel y-rot	9.99676
Lheel z-rot	2.10219
Lknee -rot	7.59616
Lhip x-rot	-2.64554
Lhip y-rot	0.169677
Lhip z-rot	9.41078
Navel -rot	-9.54453
Neck -rot	-21.9621
Lshoulder x-rot	94.9975
Lshoulder y-rot	22.4495
Lshoulder z-rot	0.0834601
Lelbow -rot	5.02659
Rshoulder x-rot	81.044
Rshoulder y-rot	24.0836
Rshoulder z-rot	36.7118
Relbow -rot	1.46844
Rhip x-rot	45.3035
Rhip y-rot	8.30273
Rhip z-rot	-6.51617
Rknee -rot	-0.136216

Without Constraint

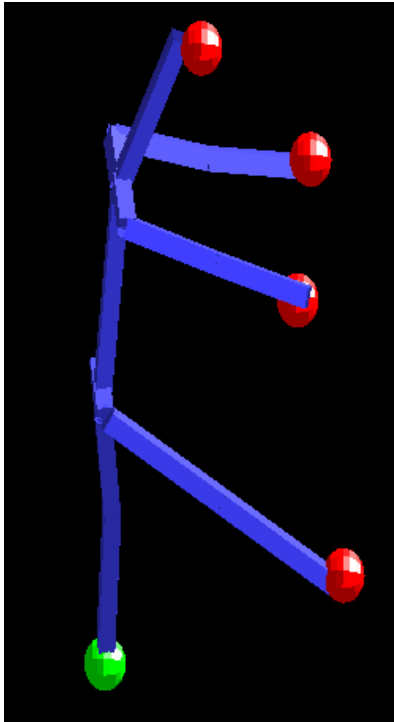


Initial Solution 時各關節角度：

Lheel x-rot	-3.71064
Lheel y-rot	9.99676
Lheel z-rot	2.10219
Lknee -rot	7.59616
Lhip x-rot	-2.64554
Lhip y-rot	0.169677
Lhip z-rot	9.41078
Navel -rot	-9.54453
Neck -rot	-21.9621
Lshoulder x-rot	94.9975
Lshoulder y-rot	22.4495
Lshoulder z-rot	0.0834601
Lelbow -rot	5.02659
Rshoulder x-rot	81.044
Rshoulder y-rot	24.0836
Rshoulder z-rot	36.7118
Relbow -rot	1.46844
Rhip x-rot	45.3035
Rhip y-rot	8.30273
Rhip z-rot	-6.51617
Rknee -rot	-0.136216



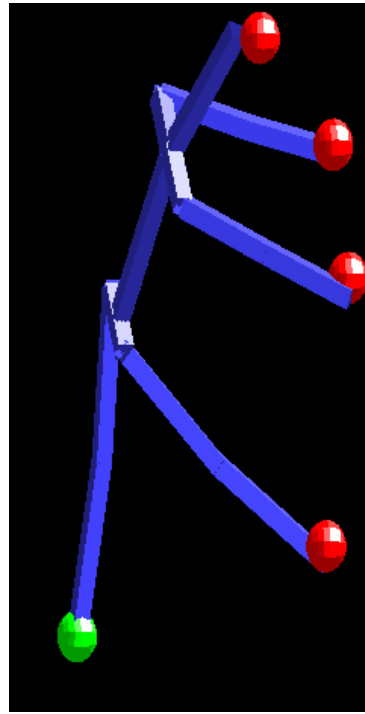
With Constraint



Iteration 2 時各關節角度：

Lheel x-rot	-3.94761
Lheel y-rot	9.54819
Lheel z-rot	3.19316
Lknee -rot	7.72035
Lhip x-rot	-1.94819
Lhip y-rot	-0.254495
Lhip z-rot	6.94209
Navel -rot	-9.54453
Neck -rot	-19.4003
Lshoulder x-rot	100.831
Lshoulder y-rot	22.1705
Lshoulder z-rot	8.35547
Lelbow -rot	6.51765
Rshoulder x-rot	75.9141
Rshoulder y-rot	23.632
Rshoulder z-rot	25.9875
Relbow -rot	1.04869
Rhip x-rot	48.2932
Rhip y-rot	8.01682
Rhip z-rot	-4.5453
Rknee -rot	-0.136216

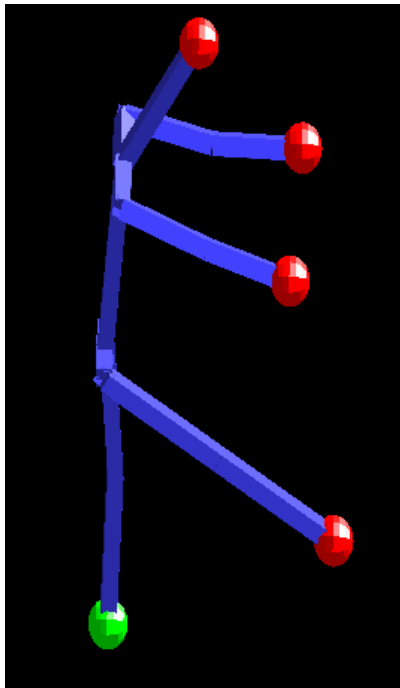
Without Constraint



Iteration 2 時各關節角度：

Lheel x-rot	-6.10507
Lheel y-rot	4.5867
Lheel z-rot	19.0495
Lknee -rot	5.84271
Lhip x-rot	-2.35532
Lhip y-rot	-2.90552
Lhip z-rot	-20.6855
Navel -rot	-5.76083
Neck -rot	-12.6707
Lshoulder x-rot	97.1422
Lshoulder y-rot	20.4664
Lshoulder z-rot	17.9678
Lelbow -rot	3.04326
Rshoulder x-rot	79.4879
Rshoulder y-rot	22.7489
Rshoulder z-rot	35.4489
Relbow -rot	0.821393
Rhip x-rot	45.856
Rhip y-rot	19.148
Rhip z-rot	14.0143
Rknee -rot	7.31749

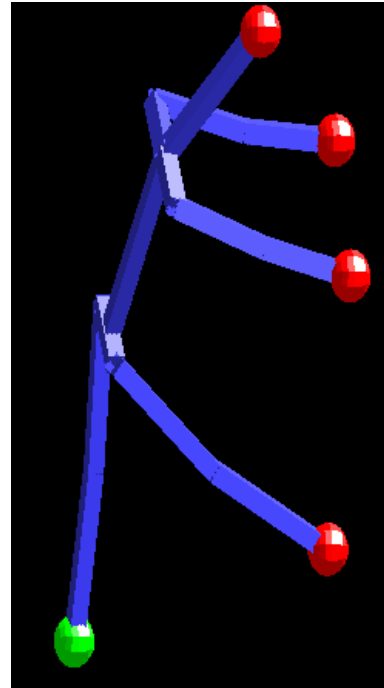
With Constraint



iteration 結束時(26 次)各關節角度：

Lheel x-rot	-4.27656
Lheel y-rot	5.5266
Lheel z-rot	6.92007
Lknee -rot	6.97237
Lhip x-rot	-0.698175
Lhip y-rot	-4.15222
Lhip z-rot	0.811916
Navel -rot	-9.15237
Neck -rot	-26.8961
Lshoulder x-rot	109.724
Lshoulder y-rot	23.39
Lshoulder z-rot	8.69505
Lelbow -rot	10.6767
Rshoulder x-rot	75.2221
Rshoulder y-rot	25.1748
Rshoulder z-rot	15.0103
Relbow -rot	3.64517
Rhip x-rot	51.1499
Rhip y-rot	7.80681
Rhip z-rot	-4.70339
Rknee -rot	-0.00631645

Without Constraint



iteration 結束時(49 次)各關節角度：

Lheel x-rot	-2.25702
Lheel y-rot	5.39286
Lheel z-rot	13.5174
Lknee -rot	3.77401
Lhip x-rot	-3.17852
Lhip y-rot	-3.84535
Lhip z-rot	-12.6408
Navel -rot	-8.05533
Neck -rot	-22.5365
Lshoulder x-rot	106.761
Lshoulder y-rot	24.7862
Lshoulder z-rot	14.081
Lelbow -rot	9.7954
Rshoulder x-rot	79.0067
Rshoulder y-rot	26.7199
Rshoulder z-rot	22.2779
Relbow -rot	5.64652
Rhip x-rot	47.9481
<b>Rhip y-rot</b>	<b>20.6887</b>
Rhip z-rot	8.40203
<b>Rknee -rot</b>	<b>10.6235</b>

實驗結果：

未加入 Constraint 的實驗結果中，有兩個轉軸的角度超過 Constraint 使人體做出超出我們定義的自由度範圍。

分別是

1. Rhip y-rot 20.6887

Lower Bound：-20，Upper Bound：20

其實幾乎沒有超過限制

2. Rknee -rot 10.6235

Lower Bound：-120，Upper Bound：0

嚴重錯誤！右膝蓋往上側翻轉。

結論：

未設定 Constraint 的話的確會造成人體模型做出人體不可能做出的姿勢。



## 6.2. 比較以 IP 為初解與以上一個 frame 的解為初解的優劣

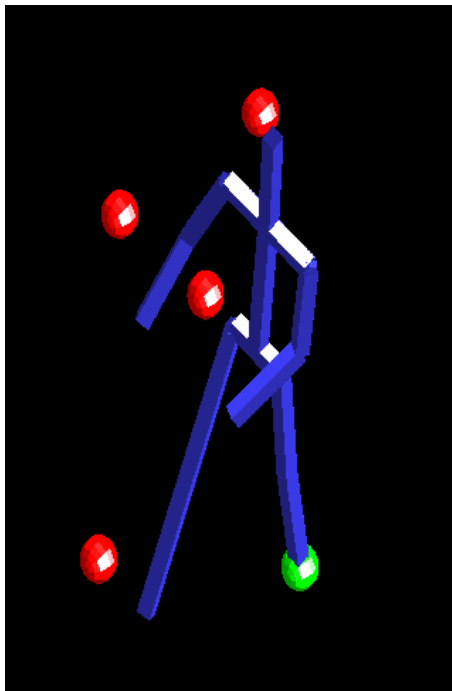
認為使用 IP 得到的粗略的各關節點資訊為初解，能夠比以上一個 frame 的解為初解更能引導逆向動力學往一個比較正確的方向求解。尤其是在 frame rate 很低，上一個 frame 的解與這個 frame 差距很大的時候，以 IP 為初解的效果更顯著。

於是取了一組上一個 frame 的解與這個 frame 差距很大的例子，讓這兩組不同的初解去逼近 target。

實驗結果如下：



原本方法：



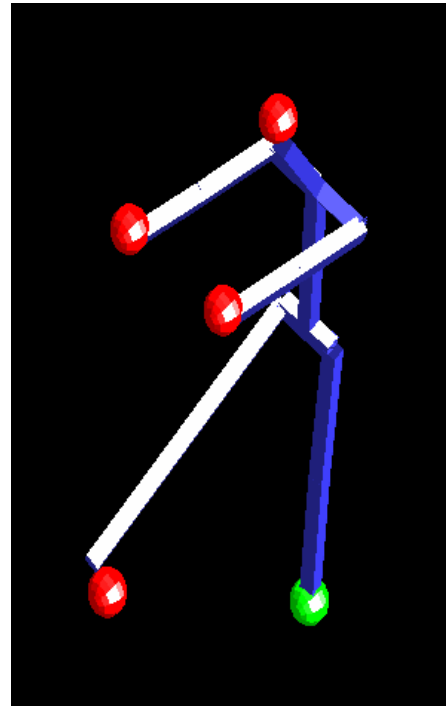
Initial Solution 時各關節角度：

Lheel x-rot	-12.8624
Lheel y-rot	-0.583171
Lheel z-rot	6.33484
Lknee -rot	4.05769
Lhip x-rot	12.7079
Lhip y-rot	4.39689
Lhip z-rot	4.84631
Navel -rot	-9.98634
Neck -rot	-2.05118
Lshoulder x-rot	10.1757
Lshoulder y-rot	-29.9595
Lshoulder z-rot	-8.08949
Lelbow -rot	49.7011
Rshoulder x-rot	8.06549
Rshoulder y-rot	23.1751
Rshoulder z-rot	18.8849
Relbow -rot	25.92
Rhip x-rot	20.606
Rhip y-rot	11.0049
Rhip z-rot	-4.10872
Rknee -rot	-0.390902

位置誤差：

LeftWrist	Position Error = 51.0387
Head	Position Error = 9.74419
RightWrist	Position Error = 45.8481
RightHeel	Position Error = 52.7801
Total Position Error	= 159.411

IP 做 Initial Solution：



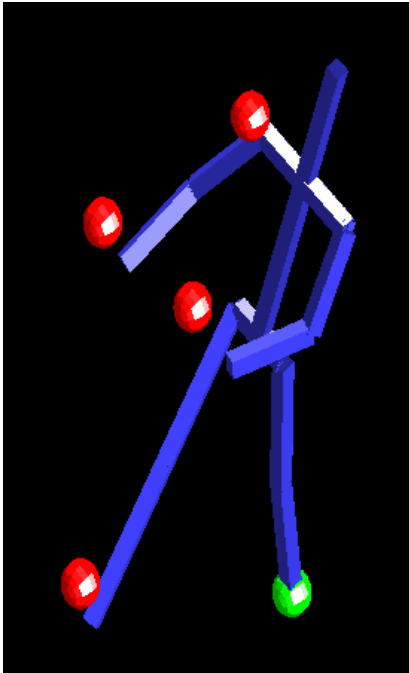
Initial Solution 時各關節角度：

Lheel x-rot	-1.54662
Lheel y-rot	9.20609
Lheel z-rot	6.16833
Lknee -rot	2.64595
Lhip x-rot	-4.64455
Lhip y-rot	-4.91797
Lhip z-rot	2.40443
Navel -rot	-3.02843
Neck -rot	-27.4671
Lshoulder x-rot	129.336
Lshoulder y-rot	13.0242
Lshoulder z-rot	8.68211
Lelbow -rot	0.0223306
Rshoulder x-rot	88.026
Rshoulder y-rot	17.2408
Rshoulder z-rot	15.5562
Relbow -rot	4.60507
Rhip x-rot	70.571
Rhip y-rot	19.0028
Rhip z-rot	3.71546
Rknee -rot	-0.0055452

位置誤差：

LeftWrist	Position Error = 2.37451
Head	Position Error = 5.74403
RightWrist	Position Error = 2.81116
RightHeel	Position Error = 8.30422
Total Position Error	= 19.2339

Iteration No.1



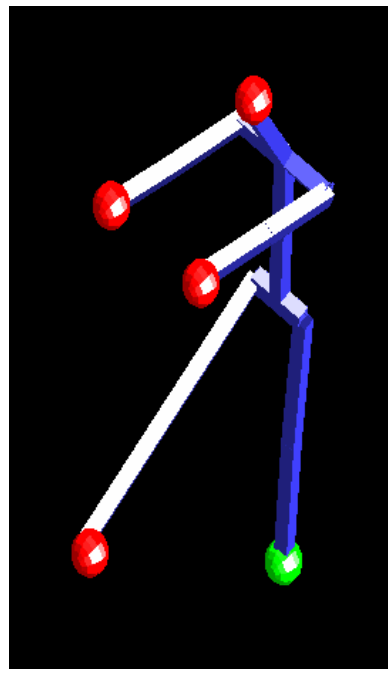
角度變化量：

Lheel x-rot	3.44381
Lheel y-rot	0.62118
Lheel z-rot	-1.91145
Lknee -rot	3.43592
Lhip x-rot	6.79471
Lhip y-rot	0.911608
Lhip z-rot	3.29065
Navel -rot	0
Neck -rot	1.54033
Lshoulder x-rot	11.3385
Lshoulder y-rot	0
Lshoulder z-rot	-1.50849
Lelbow -rot	10.1741
Rshoulder x-rot	9.22996
Rshoulder y-rot	0.564776
Rshoulder z-rot	7.3647
Relbow -rot	5.48392
Rhip x-rot	8.05164
Rhip y-rot	-0.625058
Rhip z-rot	-2.05342
Rknee -rot	0

Iteration 後的 Error：

LeftWrist	Position Error = 34.5647
Head	Position Error = 26.9594
RightWrist	Position Error = 30.0563
RightHeel	Position Error = 29.2979
Total Position Error	= 120.878

Iteration No.1



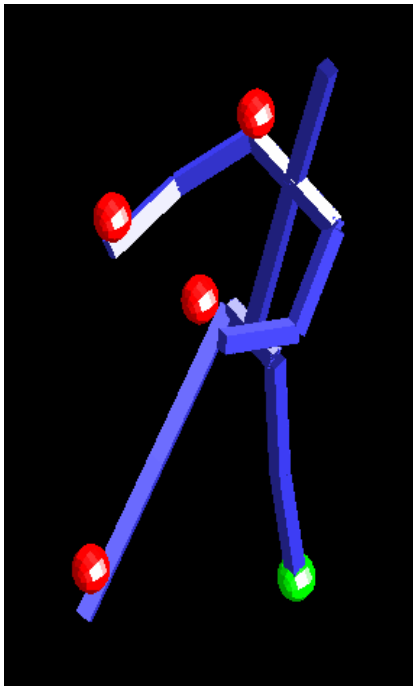
角度變化量：

Lheel x-rot	-0.332067
Lheel y-rot	0
Lheel z-rot	-0.543987
Lknee -rot	0.369082
Lhip x-rot	0.896738
Lhip y-rot	0.99705
Lhip z-rot	-1.82439
Navel -rot	0.644493
Neck -rot	0.187156
Lshoulder x-rot	-0.532279
Lshoulder y-rot	-0.0812514
Lshoulder z-rot	-0.0612277
Lelbow -rot	0
Rshoulder x-rot	0.144434
Rshoulder y-rot	0.00374821
Rshoulder z-rot	0.299319
Relbow -rot	0.0621731
Rhip x-rot	0.139752
Rhip y-rot	0.0630137
Rhip z-rot	-2.39379
Rknee -rot	0

Iteration 後的 Error：

LeftWrist	Position Error = 2.38809
Head	Position Error = 2.20685
RightWrist	Position Error = 2.14207
RightHeel	Position Error = 3.59462
Total Position Error	= 10.3316

Iteration No.2



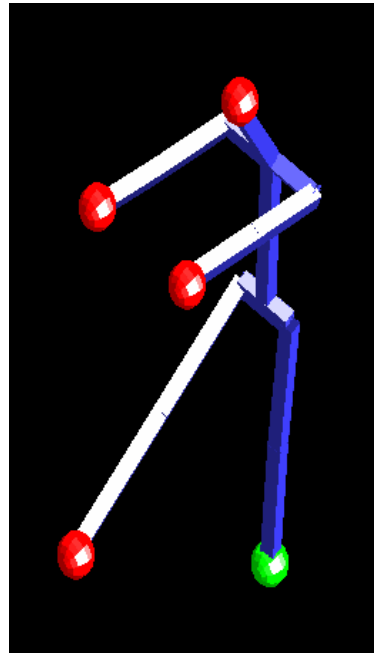
角度變化量：

Lheel x-rot	-2.23151
Lheel y-rot	0.53768
Lheel z-rot	-1.1287
Lknee -rot	-1.14103
Lhip x-rot	3.09798
Lhip y-rot	0.793339
Lhip z-rot	0.523548
Navel -rot	0
Neck -rot	2.43174
Lshoulder x-rot	9.71783
Lshoulder y-rot	0
Lshoulder z-rot	-1.87101
Lelbow -rot	6.08247
Rshoulder x-rot	7.99184
Rshoulder y-rot	1.70735
Rshoulder z-rot	4.89889
Relbow -rot	4.48883
Rhip x-rot	6.44846
Rhip y-rot	-0.723607
Rhip z-rot	-0.679223
Rknee -rot	0

Iteration 後的 Error：

LeftWrist	Position Error = 22.1383
Head	Position Error = 23.8895
RightWrist	Position Error = 19.0023
RightHeel	Position Error = 21.0156
Total	Position Error = 86.0457

Iteration No.2



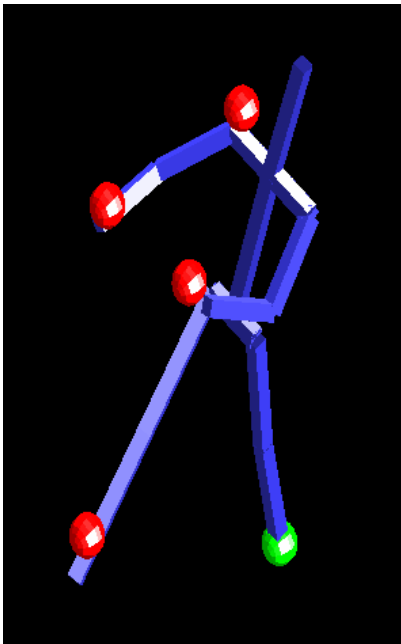
角度變化量：

Lheel x-rot	-0.39599
Lheel y-rot	0.118622
Lheel z-rot	0.341125
Lknee -rot	0.0432126
Lhip x-rot	0.537447
Lhip y-rot	0.191853
Lhip z-rot	-0.558392
Navel -rot	0.367516
Neck -rot	0.107201
Lshoulder x-rot	-0.528455
Lshoulder y-rot	-0.0762212
Lshoulder z-rot	-0.170826
Lelbow -rot	0
Rshoulder x-rot	0.046264
Rshoulder y-rot	0.0162505
Rshoulder z-rot	-0.0190939
Relbow -rot	0.0611072
Rhip x-rot	0.134836
Rhip y-rot	0.0110969
Rhip z-rot	-0.984978
Rknee -rot	0

Iteration 後的 Error：

LeftWrist	Position Error = 2.48343
Head	Position Error = 1.84696
RightWrist	Position Error = 2.05734
RightHeel	Position Error = 1.41781
Total	Position Error = 7.80554

Iteration No.3



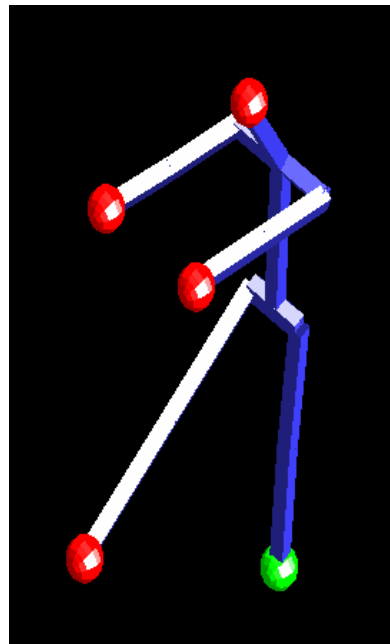
角度變化量：

Lheel x-rot	-0.266987
Lheel y-rot	-0.136543
Lheel z-rot	-0.49986
Lknee -rot	-0.765892
Lhip x-rot	0.619088
Lhip y-rot	-0.23019
Lhip z-rot	-0.0686446
Navel -rot	0
Neck -rot	0.794619
Lshoulder x-rot	7.02534
Lshoulder y-rot	0
Lshoulder z-rot	-1.60779
Lelbow -rot	3.47309
Rshoulder x-rot	5.78182
Rshoulder y-rot	1.87629
Rshoulder z-rot	2.99329
Relbow -rot	3.56174
Rhip x-rot	4.69175
Rhip y-rot	-0.58838
Rhip z-rot	-0.727481
Rknee -rot	0

Iteration 後的 Error：

LeftWrist	Position Error = 15.2985
Head	Position Error = 22.5222
RightWrist	Position Error = 12.5412
RightHeel	Position Error = 16.6103
Total Position Error	= 66.9721

Iteration No.3



角度變化量：

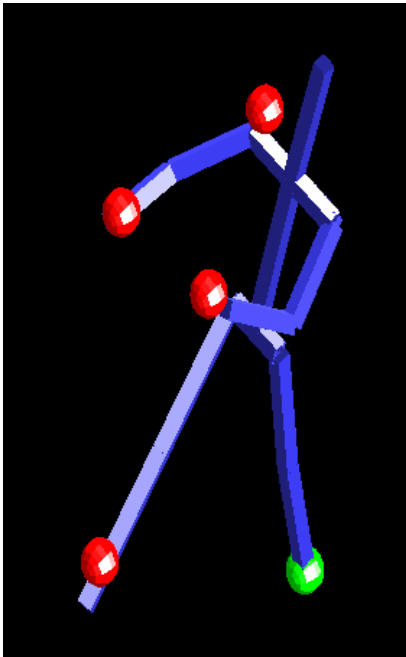
Lheel x-rot	-0.287103
Lheel y-rot	-0.254315
Lheel z-rot	0.245031
Lknee -rot	-0.000926229
Lhip x-rot	0.365533
Lhip y-rot	-0.203897
Lhip z-rot	-0.308209
Navel -rot	0.22949
Neck -rot	0.0834097
Lshoulder x-rot	-0.474158
Lshoulder y-rot	-0.0667774
Lshoulder z-rot	-0.160493
Lelbow -rot	0
Rshoulder x-rot	-0.0216946
Rshoulder y-rot	0.0096029
Rshoulder z-rot	-0.122448
Relbow -rot	0.0403235
Rhip x-rot	0.116654
Rhip y-rot	0.00168476
Rhip z-rot	-0.483852
Rknee -rot	0

Iteration 後的 Error：

LeftWrist	Position Error = 2.27563
Head	Position Error = 1.68083
RightWrist	Position Error = 1.96487
RightHeel	Position Error = 0.934999
Total Position Error	= 6.85633



Iteration No.4



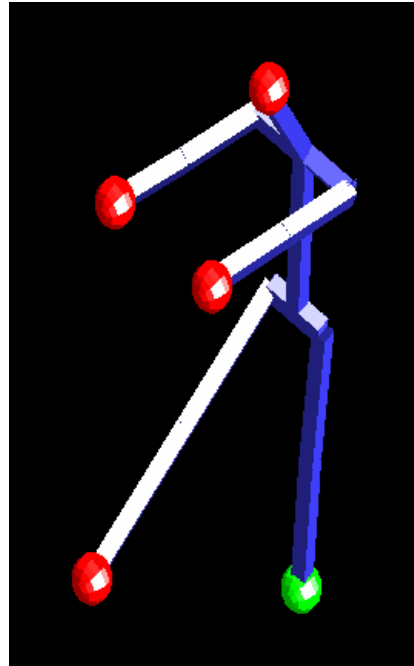
角度變化量：

Lheel x-rot	0.332082
Lheel y-rot	-0.400779
Lheel z-rot	-0.140335
Lknee -rot	-0.725043
Lhip x-rot	-0.561713
Lhip y-rot	-0.567777
Lhip z-rot	-0.254972
Navel -rot	0
Neck -rot	-0.175501
Lshoulder x-rot	5.19691
Lshoulder y-rot	0
Lshoulder z-rot	-1.36847
Lelbow -rot	2.05624
Rshoulder x-rot	4.20956
Rshoulder y-rot	1.79419
Rshoulder z-rot	1.81066
Relbow -rot	2.92409
Rhip x-rot	3.69565
Rhip y-rot	-0.516055
Rhip z-rot	-0.759949
Rknee -rot	0

Iteration 後的 Error：

LeftWrist	Position Error = 11.5232
Head	Position Error = 20.7055
RightWrist	Position Error = 8.51053
RightHeel	Position Error = 14.0444
Total Position Error	= 54.7836

Iteration No.4



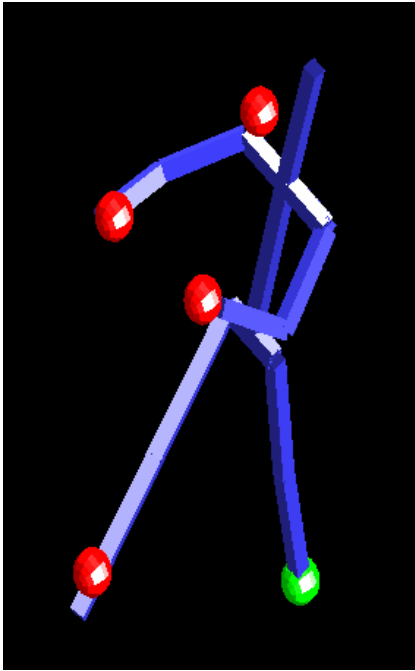
角度變化量：

Lheel x-rot	-0.217568
Lheel y-rot	-0.280301
Lheel z-rot	0.222821
Lknee -rot	-0.0109191
Lhip x-rot	0.276999
Lhip y-rot	-0.239944
Lhip z-rot	-0.248775
Navel -rot	0.16742
Neck -rot	0.0900001
Lshoulder x-rot	-0.407434
Lshoulder y-rot	-0.0550601
Lshoulder z-rot	-0.174632
Lelbow -rot	0
Rshoulder x-rot	-0.05302
Rshoulder y-rot	0.00803783
Rshoulder z-rot	-0.180983
Relbow -rot	0.0312902
Rhip x-rot	0.0943268
Rhip y-rot	-0.000583632
Rhip z-rot	-0.349017
Rknee -rot	0

Iteration 後的 Error：

LeftWrist	Position Error = 2.05501
Head	Position Error = 1.56947
RightWrist	Position Error = 1.84836
RightHeel	Position Error = 0.715575
Total Position Error	= 6.18841

Iteration No.5



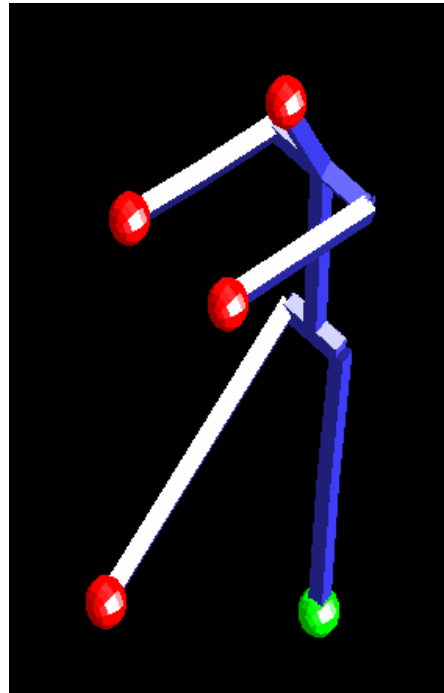
角度變化量：

Lheel x-rot	0.605863
Lheel y-rot	-0.515251
Lheel z-rot	0.00468029
Lknee -rot	-0.642577
Lhip x-rot	-1.07571
Lhip y-rot	-0.692979
Lhip z-rot	-0.299319
Navel -rot	0
Neck -rot	-0.686365
Lshoulder x-rot	4.02841
Lshoulder y-rot	0
Lshoulder z-rot	-1.21002
Lelbow -rot	1.29433
Rshoulder x-rot	3.10458
Rshoulder y-rot	0
Rshoulder z-rot	1.08902
Relbow -rot	2.45628
Rhip x-rot	3.04987
Rhip y-rot	-0.470442
Rhip z-rot	-0.741485
Rknee -rot	0

Iteration 後的 Error：

LeftWrist	Position Error = 9.3335
Head	Position Error = 18.8073
RightWrist	Position Error = 6.67358
RightHeel	Position Error = 12.2024
Total Position Error	= 47.0168

Iteration No.5



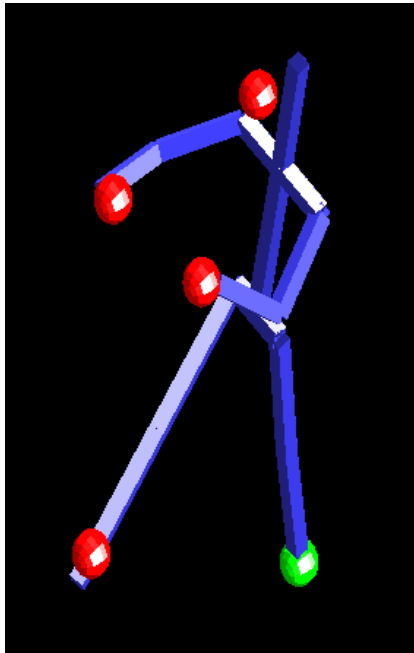
角度變化量：

Lheel x-rot	-0.173859
Lheel y-rot	-0.258671
Lheel z-rot	0.196668
Lknee -rot	-0.0133389
Lhip x-rot	0.22383
Lhip y-rot	-0.225024
Lhip z-rot	-0.225969
Navel -rot	0.131443
Neck -rot	0.0967008
Lshoulder x-rot	-0.347644
Lshoulder y-rot	-0.0448695
Lshoulder z-rot	-0.183439
Lelbow -rot	0
Rshoulder x-rot	-0.0613108
Rshoulder y-rot	0.00934922
Rshoulder z-rot	-0.209268
Relbow -rot	0.0296903
Rhip x-rot	0.0785884
Rhip y-rot	-0.00159838
Rhip z-rot	-0.288466
Rknee -rot	0

Iteration 後的 Error：

LeftWrist	Position Error = 1.86022
Head	Position Error = 1.47943
RightWrist	Position Error = 1.73228
RightHeel	Position Error = 0.573228
Total Position Error	= 5.64515

Iteration No.8



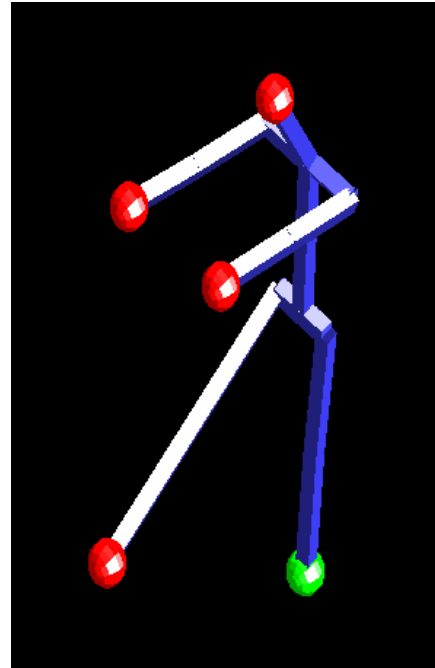
角度變化量：

Lheel x-rot	0.666916
Lheel y-rot	-0.311475
Lheel z-rot	0.186015
Lknee -rot	-0.450624
Lhip x-rot	-1.28473
Lhip y-rot	-0.468564
Lhip z-rot	-0.355663
Navel -rot	0
Neck -rot	-0.896372
Lshoulder x-rot	2.33213
Lshoulder y-rot	0
Lshoulder z-rot	-0.815586
Lelbow -rot	0.422461
Rshoulder x-rot	1.51622
Rshoulder y-rot	0
Rshoulder z-rot	-0.0925186
Relbow -rot	1.81514
Rhip x-rot	1.9355
Rhip y-rot	-0.347517
Rhip z-rot	0
Rknee -rot	0

Iteration 後的 Error：

LeftWrist	Position Error = 6.29158
Head	Position Error = 13.5139
RightWrist	Position Error = 4.39583
RightHeel	Position Error = 8.55529
Total Position Error	= 32.7566

Iteration No.8



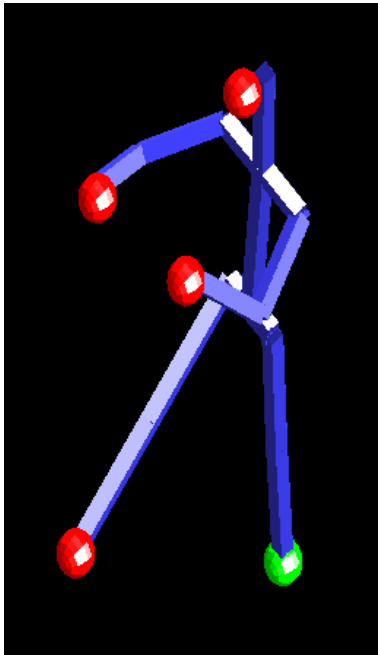
角度變化量：

Lheel x-rot	-0.102103
Lheel y-rot	-0.190944
Lheel z-rot	0.133936
Lknee -rot	-0.0145028
Lhip x-rot	0.137631
Lhip y-rot	-0.169919
Lhip z-rot	-0.174693
Navel -rot	0.0715338
Neck -rot	0.0955559
Lshoulder x-rot	-0.218295
Lshoulder y-rot	-0.0242559
Lshoulder z-rot	-0.181458
Lelbow -rot	0
Rshoulder x-rot	-0.0383825
Rshoulder y-rot	0.016765
Rshoulder z-rot	-0.215992
Relbow -rot	0.0386279
Rhip x-rot	0.0553607
Rhip y-rot	-0.00282875
Rhip z-rot	-0.201816
Rknee -rot	0

Iteration 後的 Error：

LeftWrist	Position Error = 1.41099
Head	Position Error = 1.2786
RightWrist	Position Error = 1.42631
RightHeel	Position Error = 0.328519
Total Position Error	= 4.44442

Iteration No.13



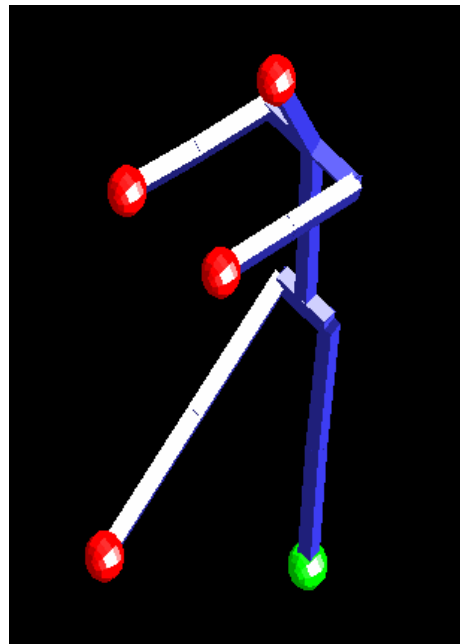
角度變化量：

Lheel x-rot	0.424951
Lheel y-rot	-0.132302
Lheel z-rot	0.163213
Lknee -rot	-0.249479
Lhip x-rot	-0.857566
Lhip y-rot	-0.244292
Lhip z-rot	-0.34394
Navel -rot	0
Neck -rot	-0.465234
Lshoulder x-rot	1.23712
Lshoulder y-rot	0
Lshoulder z-rot	-0.45409
Lelbow -rot	-0.0105983
Rshoulder x-rot	0.726674
Rshoulder y-rot	0
Rshoulder z-rot	-0.291327
Relbow -rot	1.10467
Rhip x-rot	1.05088
Rhip y-rot	-0.192275
Rhip z-rot	0
Rknee -rot	0

Iteration 後的 Error：

LeftWrist	Position Error = 4.2013
Head	Position Error = 7.59228
RightWrist	Position Error = 2.68511
RightHeel	Position Error = 5.15282
Total Position Error	= 19.6315

Iteration No.13



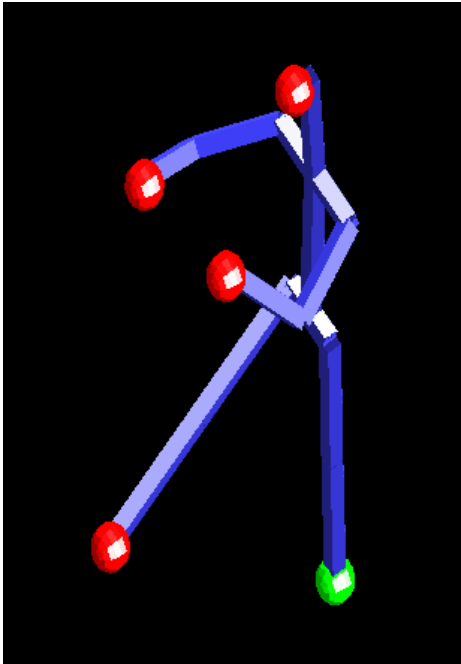
角度變化量：

Lheel x-rot	-0.0477366
Lheel y-rot	-0.123057
Lheel z-rot	0.07478
Lknee -rot	-0.0137638
Lhip x-rot	0.0724559
Lhip y-rot	-0.112633
Lhip z-rot	-0.107321
Navel -rot	0.0242665
Neck -rot	0.0720569
Lshoulder x-rot	-0.103274
Lshoulder y-rot	-0.00858616
Lshoulder z-rot	-0.146695
Lelbow -rot	0
Rshoulder x-rot	0.00163217
Rshoulder y-rot	0.021179
Rshoulder z-rot	-0.167972
Relbow -rot	0.0470786
Rhip x-rot	0.0418492
Rhip y-rot	-0.00342875
Rhip z-rot	-0.139103
Rknee -rot	0

Iteration 後的 Error：

LeftWrist	Position Error = 0.920239
Head	Position Error = 1.09497
RightWrist	Position Error = 1.06348
RightHeel	Position Error = 0.18831
Total Position Error	= 3.267

Iteration No.17



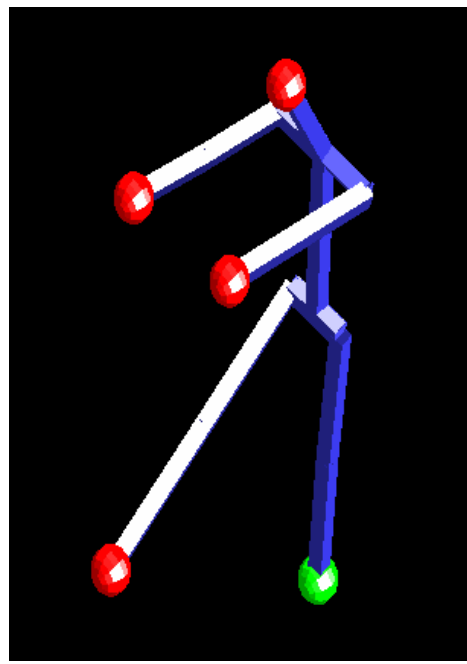
角度變化量：

Lheel x-rot	0.281427
Lheel y-rot	-0.0850479
Lheel z-rot	0.137481
Lknee -rot	-0.16224
Lhip x-rot	-0.582929
Lhip y-rot	-0.158486
Lhip z-rot	-0.27337
Navel -rot	0
Neck -rot	-0.214994
Lshoulder x-rot	0.782488
Lshoulder y-rot	0
Lshoulder z-rot	-0.290875
Lelbow -rot	-0.121572
Rshoulder x-rot	0.467313
Rshoulder y-rot	0
Rshoulder z-rot	-0.168474
Relbow -rot	0.685977
Rhip x-rot	0.682464
Rhip y-rot	-0.127287
Rhip z-rot	0
Rknee -rot	0.373779

Iteration 後的 Error：

LeftWrist	Position Error = 3.13355
Head	Position Error = 4.91492
RightWrist	Position Error = 1.61427
RightHeel	Position Error = 3.4776
Total Position Error	= 13.1403

Iteration No.17



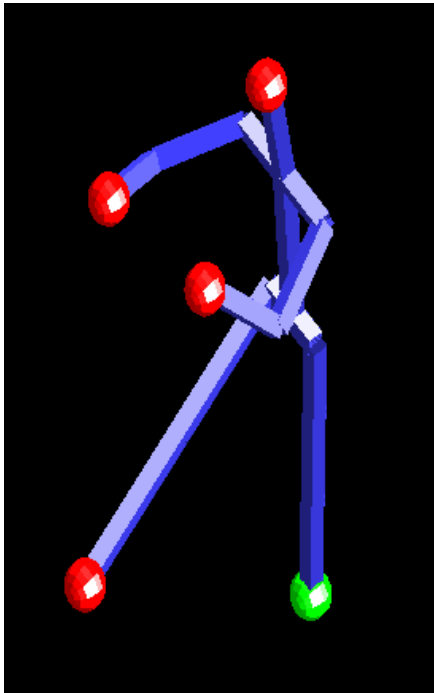
角度變化量：

Lheel x-rot	-0.0269606
Lheel y-rot	-0.0895507
Lheel z-rot	0.0480489
Lknee -rot	-0.0125575
Lhip x-rot	0.0470134
Lhip y-rot	-0.0833801
Lhip z-rot	-0.0707384
Navel -rot	0.00816345
Neck -rot	0.0561584
Lshoulder x-rot	-0.0546906
Lshoulder y-rot	-0.00321115
Lshoulder z-rot	-0.118482
Lelbow -rot	0
Rshoulder x-rot	0.0130837
Rshoulder y-rot	0.0192942
Rshoulder z-rot	-0.131688
Relbow -rot	0.0443873
Rhip x-rot	0.0350287
Rhip y-rot	-0.00345243
Rhip z-rot	-0.109869
Rknee -rot	0

Iteration 後的 Error：

LeftWrist	Position Error = 0.654775
Head	Position Error = 1.02182
RightWrist	Position Error = 0.865641
RightHeel	Position Error = 0.155281
Total Position Error	= 2.69752

修正後：(iteration 36 次)



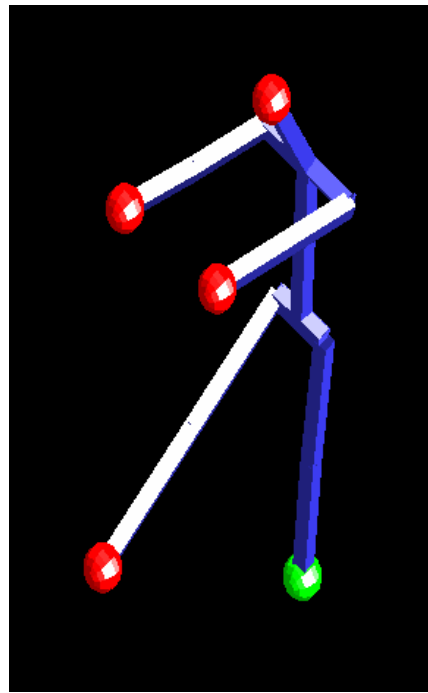
最後結果各關節角度：

Lheel x-rot	-3.11572
Lheel y-rot	-3.76433
Lheel z-rot	5.86796
Lknee -rot	0.026287
Lhip x-rot	4.19132
Lhip y-rot	-0.652905
Lhip z-rot	1.54559
Navel -rot	-9.98634
Neck -rot	-5.23197
Lshoulder x-rot	73.4802
Lshoulder y-rot	-29.9595
Lshoulder z-rot	-24.0625
Lelbow -rot	72.6659
Rshoulder x-rot	55.7393
Rshoulder y-rot	29.9671
Rshoulder z-rot	33.7497
Relbow -rot	66.2338
Rhip x-rot	68.792
Rhip y-rot	4.04939
Rhip z-rot	-9.99676
Rknee -rot	-0.0171234

修正後位置 Error：

LeftWrist	Position Error = 0.993925
Head	Position Error = 1.95037
RightWrist	Position Error = 0.61613
RightHeel	Position Error = 0.744732
Total	Position Error = 4.30516

修正後：(iteration 17 次)

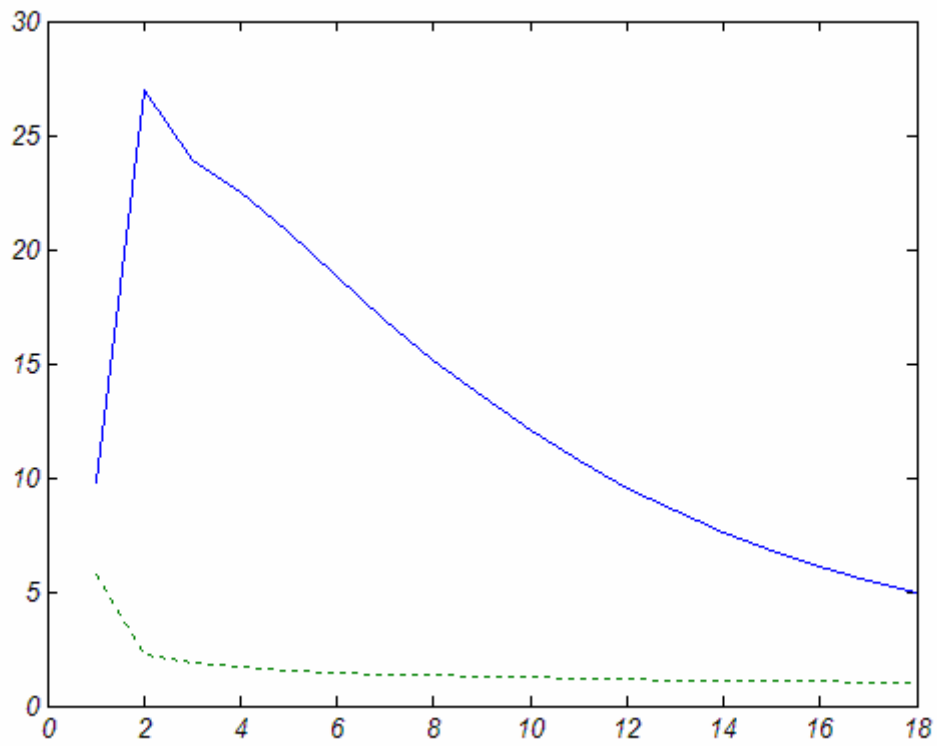


最後結果各關節角度：

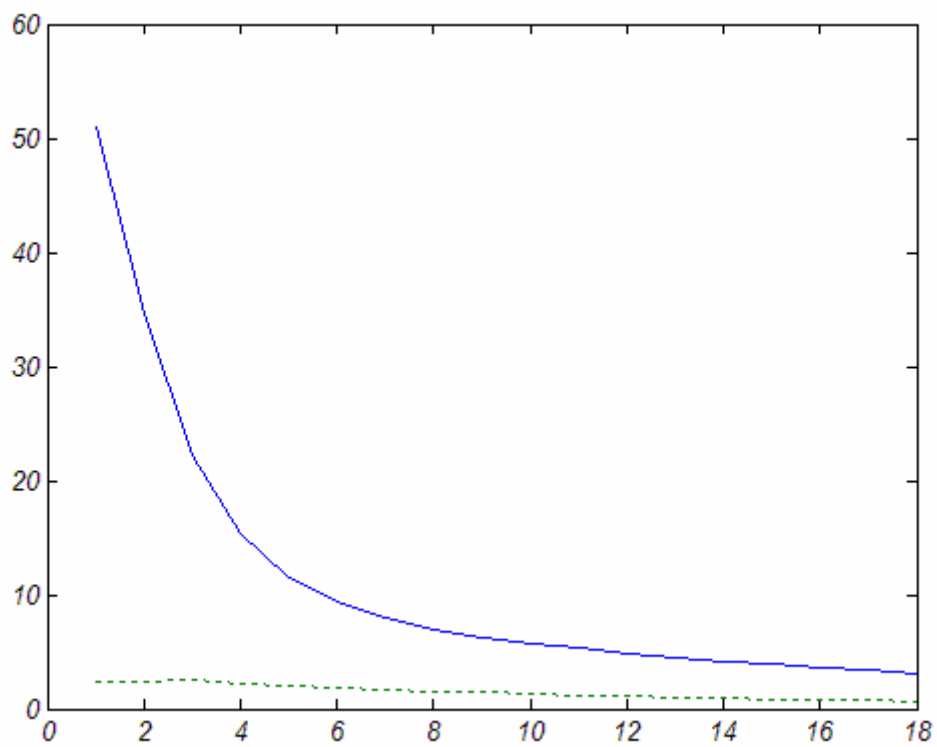
Lheel x-rot	-3.78288
Lheel y-rot	6.7569
Lheel z-rot	7.79291
Lknee -rot	2.86704
Lhip x-rot	-1.16762
Lhip y-rot	-5.99823
Lhip z-rot	-2.32991
Navel -rot	-0.971196
Neck -rot	-25.9552
Lshoulder x-rot	125.279
Lshoulder y-rot	12.5213
Lshoulder z-rot	6.056
Lelbow -rot	0.00223581
Rshoulder x-rot	87.9161
Rshoulder y-rot	17.5131
Rshoulder z-rot	13.1468
Relbow -rot	5.34377
Rhip x-rot	71.7007
Rhip y-rot	19.0387
Rhip z-rot	-2.74211
Rknee -rot	-0.0055452

修正後位置 Error：

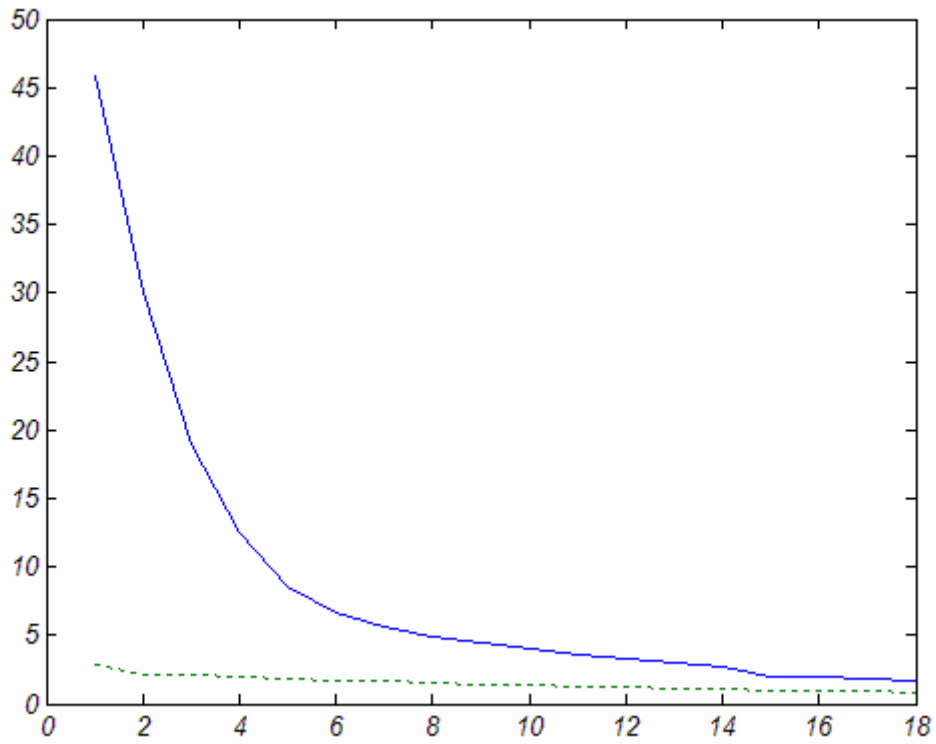
LeftWrist	Position Error = 0.654775
Head	Position Error = 1.02182
RightWrist	Position Error = 0.865641
RightHeel	Position Error = 0.155281
Total	Position Error = 2.69752



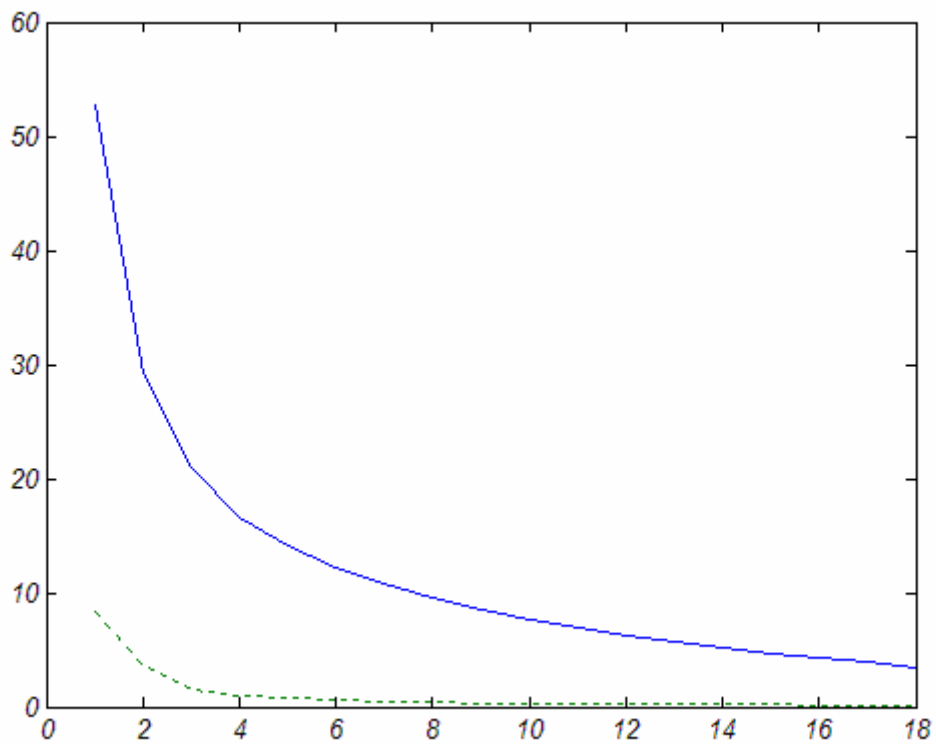
頭部位置誤差收斂比較圖  
 (藍色實線：原本方法；綠色虛線：以 IP 做初解)



左手腕位置誤差收斂比較圖

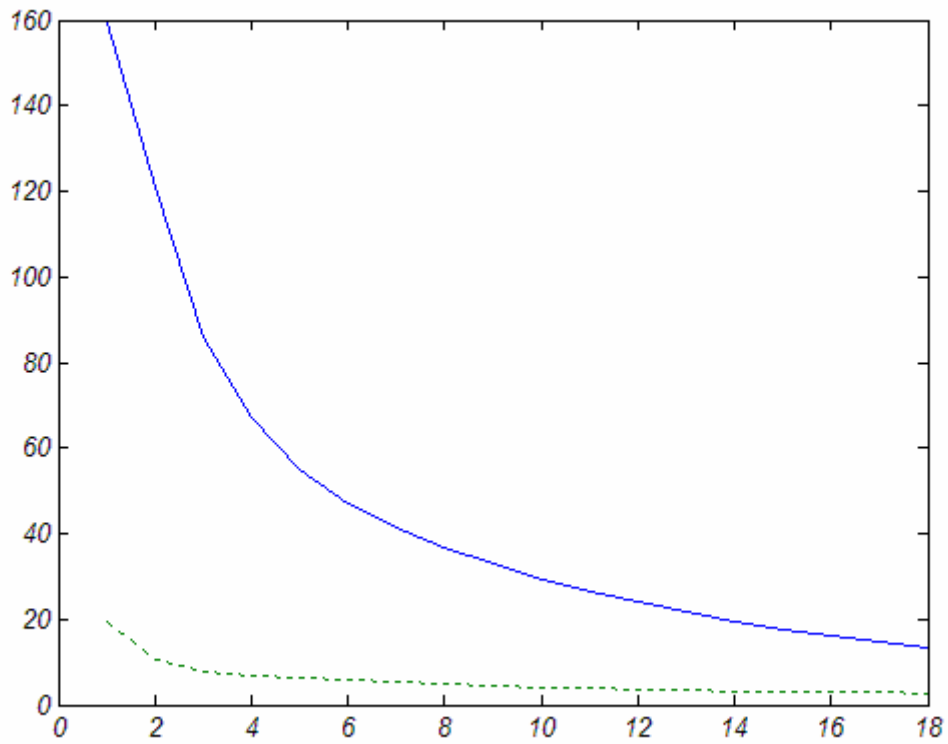


右手腕位置誤差收斂比較圖  
(藍色實線：原本方法；綠色虛線：以 IP 做初解)





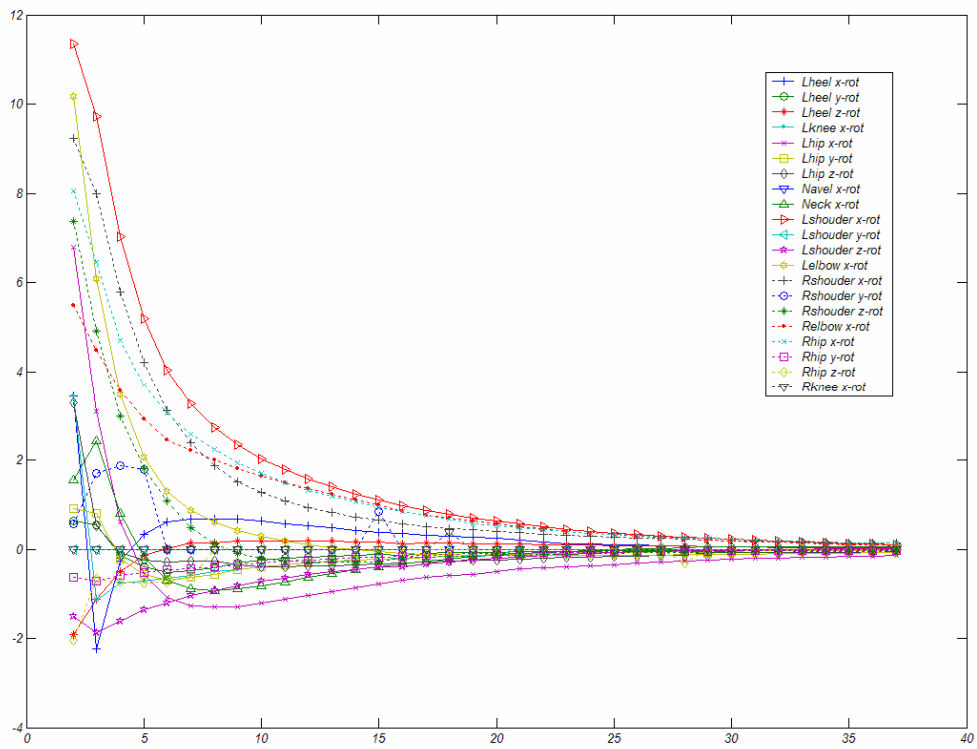
右腳跟位置誤差收斂比較圖



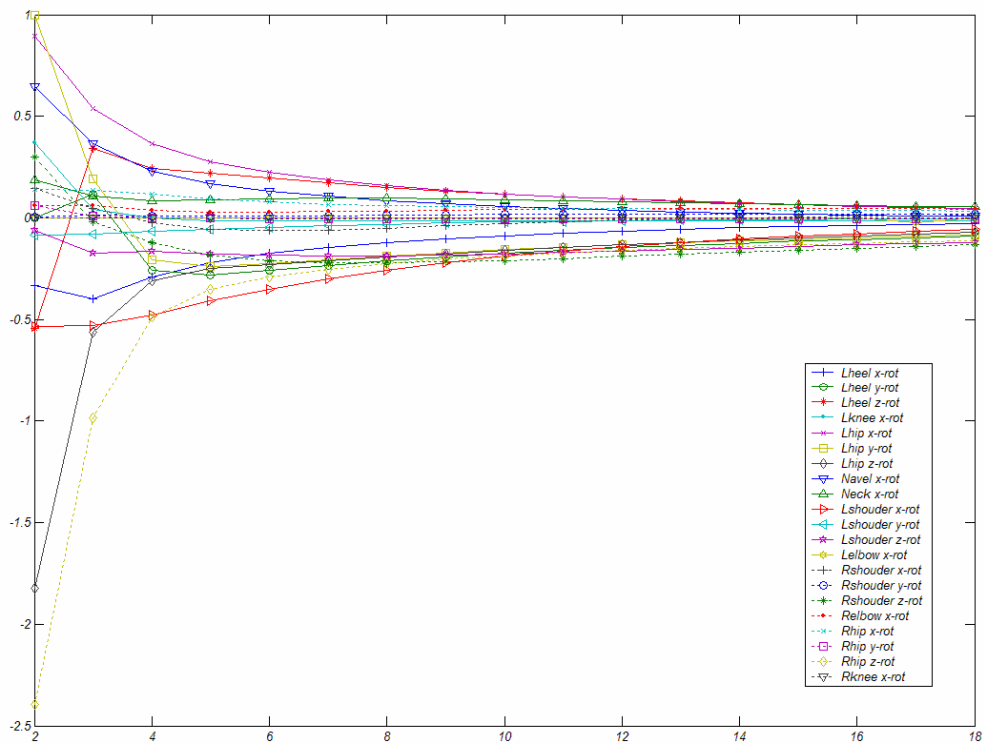
End-Effectors 總和位置誤差收斂比較圖

(藍色實線：原本方法；綠色虛線：以 IP 做初解)

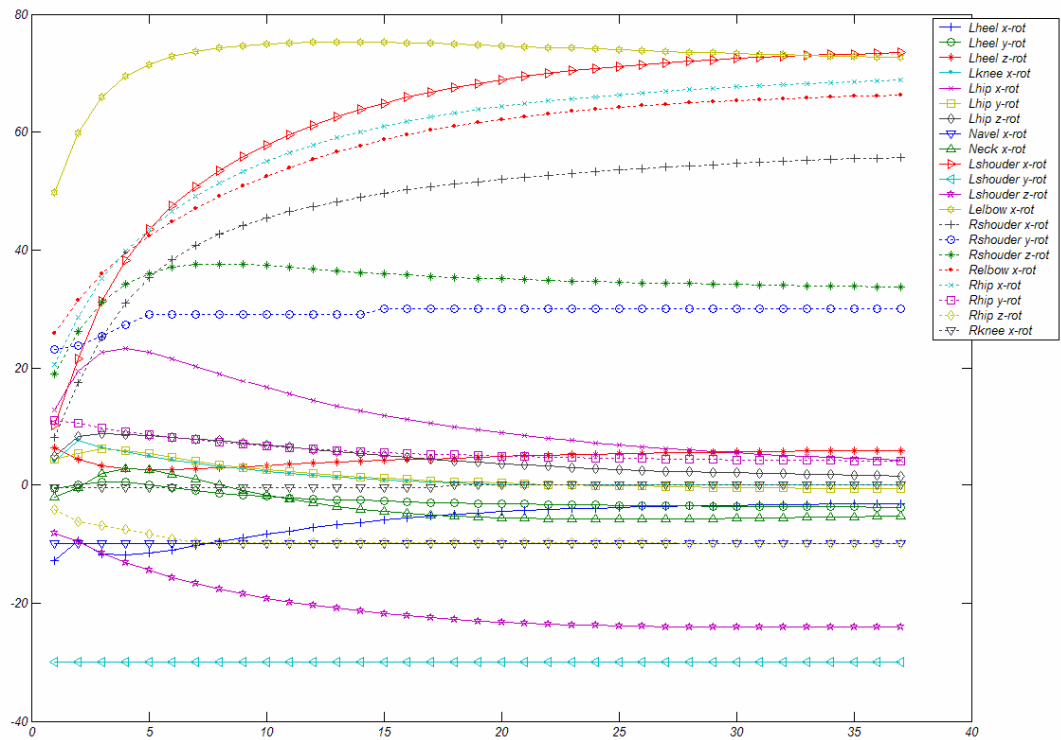
原本方法的  $\Delta\theta$  收斂情形：



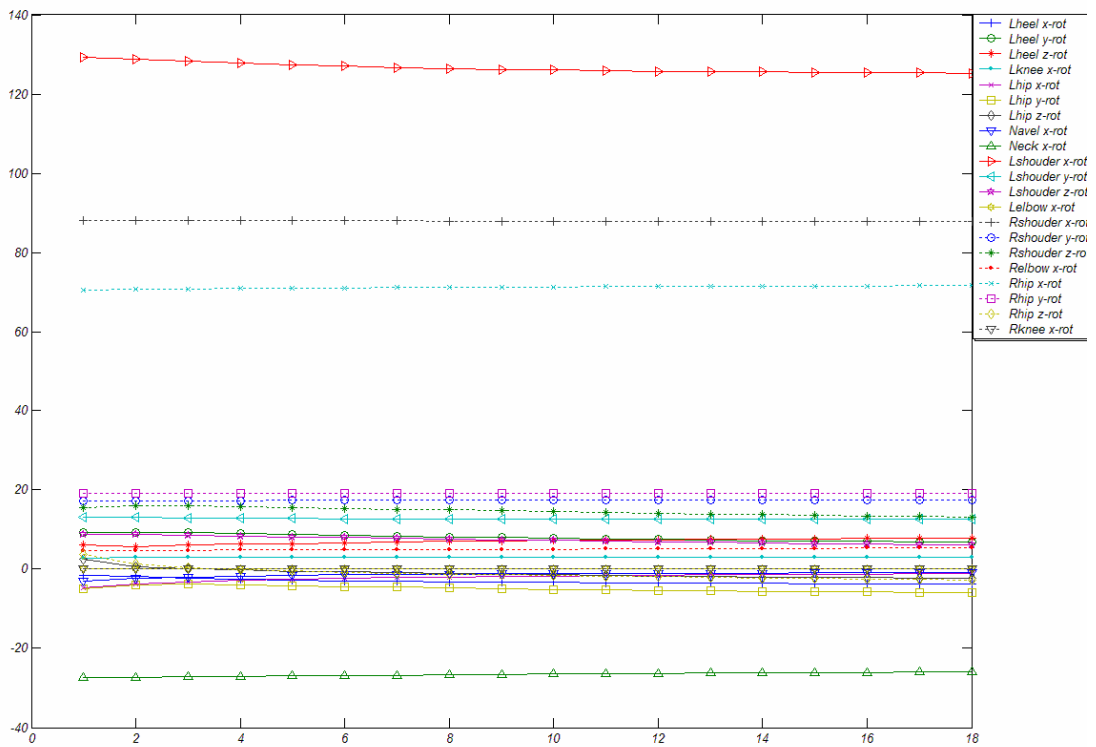
IP 方法的  $\Delta\theta$  收斂情形：



原本方法的 $\theta$ 收敛情形：



IP 方法的 $\theta$ 收敛情形：





View 1 Data Image



View 2 Data Image

實驗結果：

原本方法以上一個 frame 為初解不僅耗時，且 IK iteration 完成後，對照真實拍攝照片結果，左右肩膀、左右手肘的部份姿勢不正確。

以 IP 為初解速度就較快；且對照真實拍攝照片結果，姿勢正確。

結果分析：

原本方法	IP 做初解的方法
Initial Solution End-effectors 距離 Target 較遠	Initial Solution End-effectors 距離 Target 較遠
Initial Solution Local Angle 較不接近真實	Initial Solution Local Angle 接近真實
IK iteration 所需次數較多	IK iteration 所需次數較少
End-Effectors 到達 Target 速度較慢	End-Effectors 到達 Target 速度較快
$\theta$ 收斂情形較慢	$\theta$ 收斂情形較快
$\Delta\theta$ 收斂情形較慢	$\Delta\theta$ 收斂情形較快
最後結果雖然到達 Target，但卻不貼近真實	最後結果到達 Target 且貼近真實

### 6.3. 比較 CCD 與 SDLS 的差異

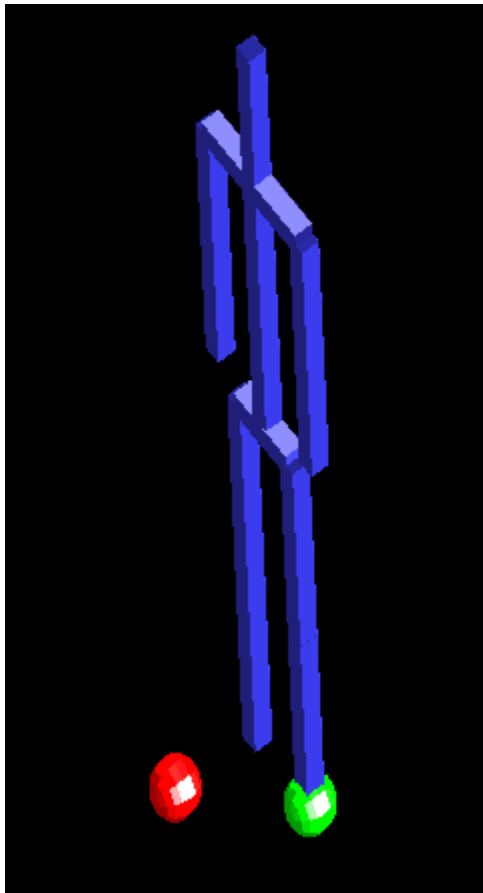
爲了要了解 cyclic coordinate decent 與 selectively damped least square 的差別，分別使用 CCD 與 SDLS 的方法做單一 End-effector 右腳跟 IK iteration 比較。

停止條件爲 Position Error <0.6cm 且  $\sum_{j=1}^n |\Delta\theta_j^i|$  值 <1°。

實驗結果如下：



CCD Method :



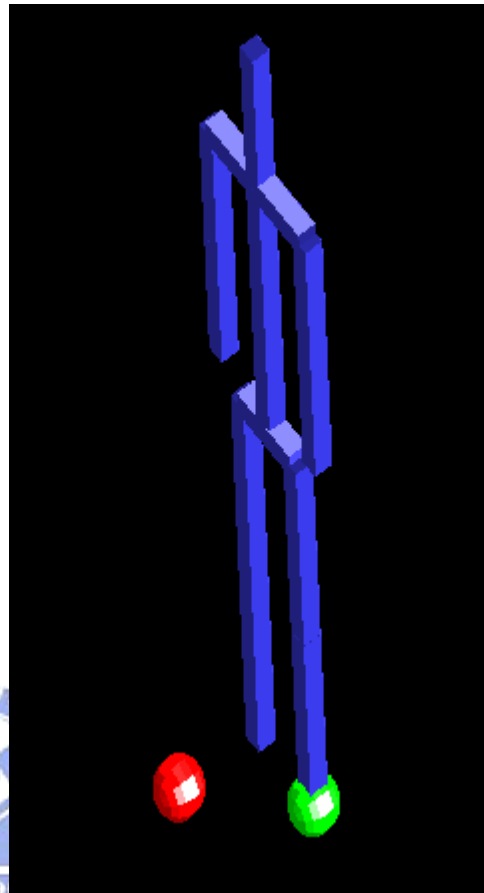
Initial Solution 時各關節角度 :

Lheel x-rot	0
Lheel y-rot	0
Lheel z-rot	0
Lknee -rot	0
Lhip x-rot	0
Lhip y-rot	0
Lhip z-rot	0
Rhip x-rot	0
Rhip y-rot	0
Rhip z-rot	0
Rknee -rot	0

位置誤差 :

RightHeel Position Error = 33.6008

Selectively Damped Least Square :



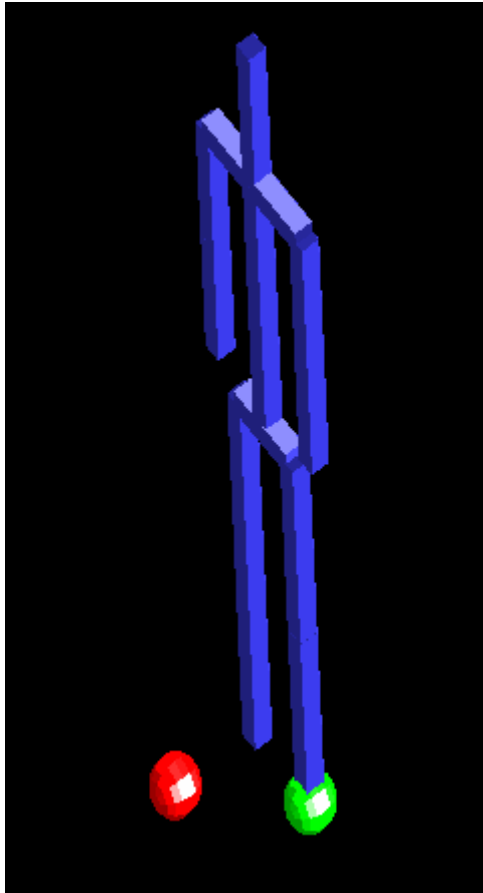
Initial Solution 時各關節角度 :

Lheel x-rot	0
Lheel y-rot	0
Lheel z-rot	0
Lknee -rot	0
Lhip x-rot	0
Lhip y-rot	0
Lhip z-rot	0
Rhip x-rot	0
Rhip y-rot	0
Rhip z-rot	0
Rknee -rot	0

位置誤差 :

RightHeel Position Error = 33.6008

Iteration No.1



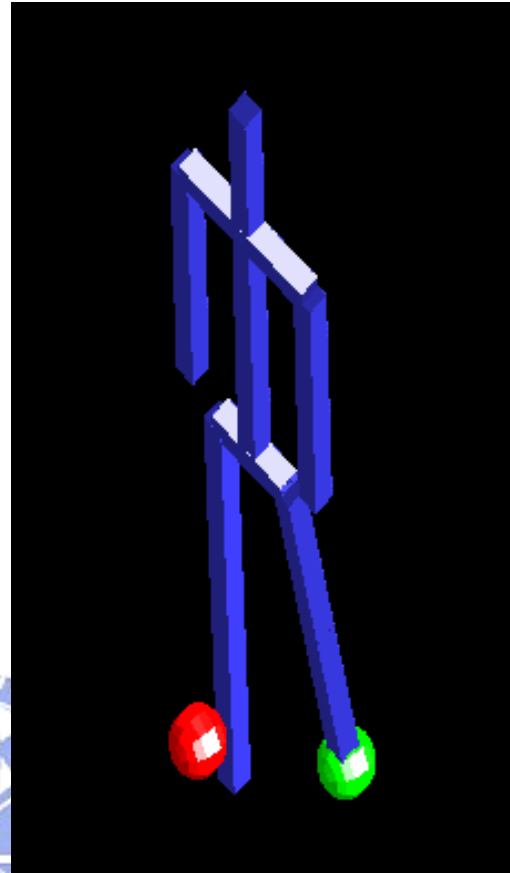
角度變化量：

Rknee -rot 0

位置誤差：

RightHeel Position Error = 33.6008

Iteration No.1



角度變化量：

Lheel x-rot -14.2177

Lheel y-rot 3.41823

Lheel z-rot 7.3766

Lknee -rot 0

Lhip x-rot 7.6152

Lhip y-rot 3.41823

Lhip z-rot 1.51589

Rhip x-rot 4.68349

Rhip y-rot 0

Rhip z-rot -5.61221

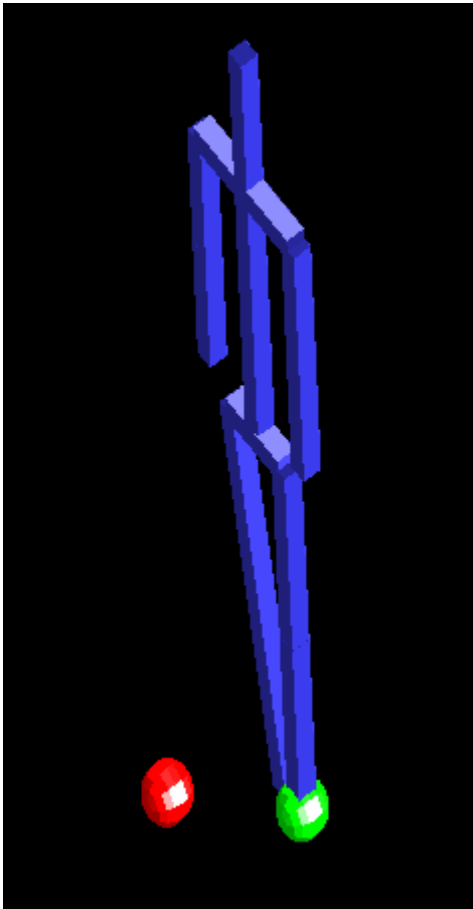
Rknee -rot 0

位置誤差：

RightHeel Position Error = 18.9769



Iteration No.2



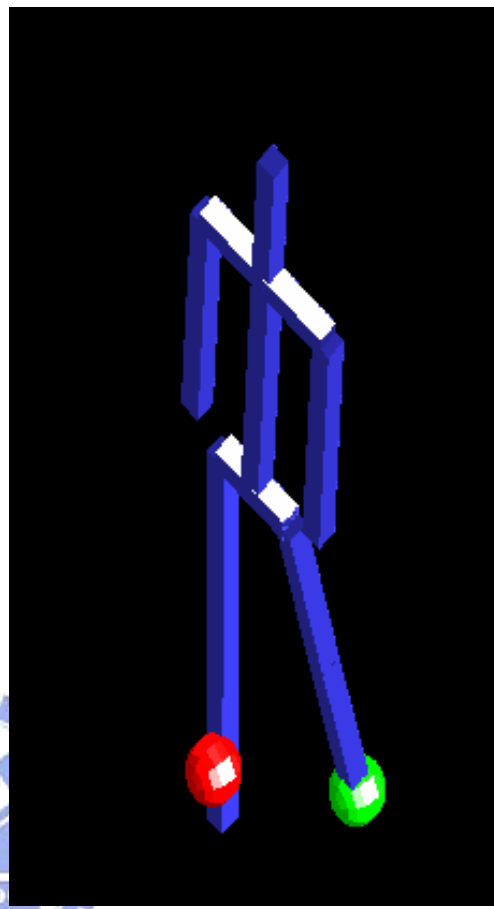
角度變化量：

Rhip z-rot -9.92442

位置誤差：

RightHeel Position Error = 31.6372

Iteration No.2



角度變化量：

Lheel x-rot -1.239

Lheel y-rot 0.322862

Lheel z-rot -1.33463

Lknee -rot 0.564701

Lhip x-rot 2.26187

Lhip y-rot 1.02587

Lhip z-rot 4.25902

Rhip x-rot 1.55885

Rhip y-rot -0.153182

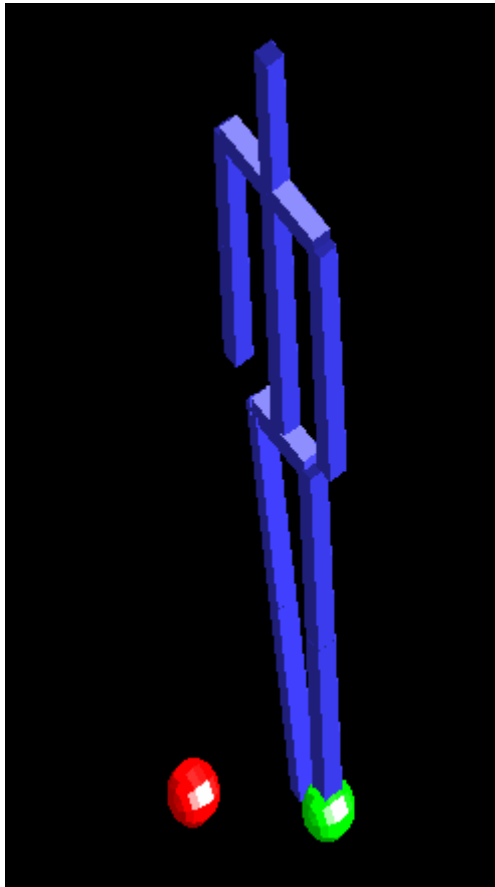
Rhip z-rot -2.95923

Rknee -rot 0

位置誤差：

RightHeel Position Error = 16.1243

Iteration No.3



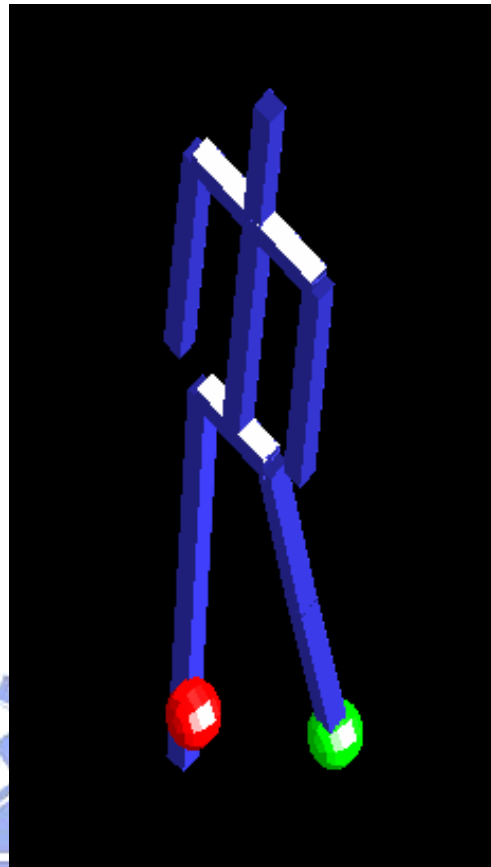
角度變化量：

Rhip y-rot -9.41879

位置誤差：

RightHeel Position Error = 29.7241

Iteration No.3



角度變化量：

Lheel x-rot 0.603826

Lheel y-rot 0.24521

Lheel z-rot -0.72576

Lknee -rot 0.514231

Lhip x-rot 0.602323

Lhip y-rot 0.557451

Lhip z-rot 2.17884

Rhip x-rot 0.922247

Rhip y-rot -0.138064

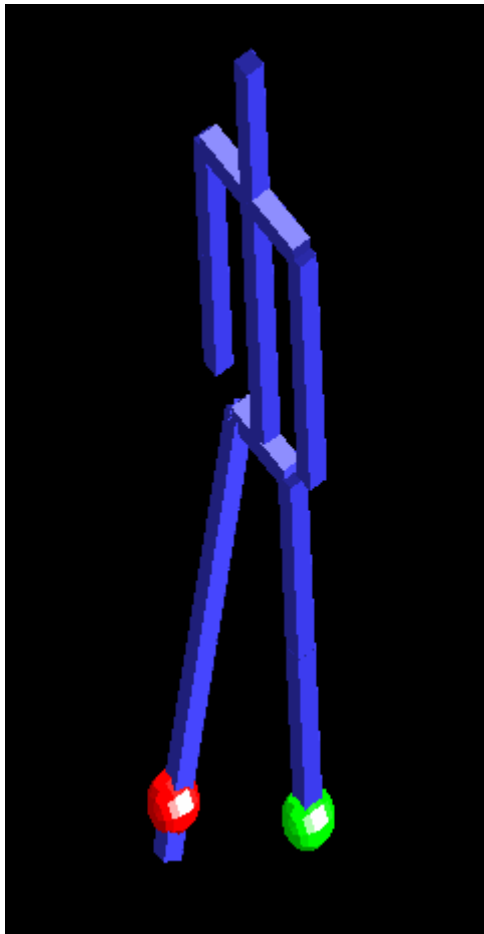
Rhip z-rot 0

Rknee -rot 0

位置誤差：

RightHeel Position Error = 15.287

Iteration No.4



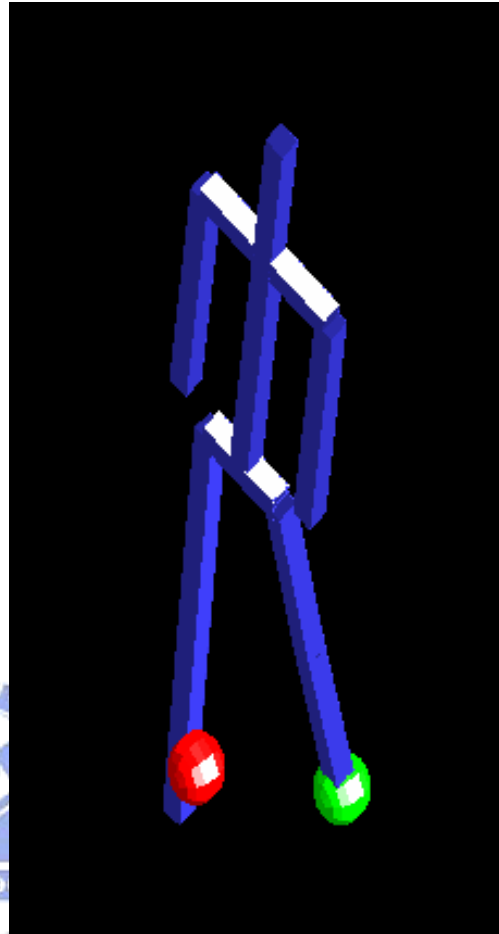
角度變化量：

Rhip x-rot 23.7023

位置誤差：

RightHeel Position Error = 8.75207

Iteration No.4



角度變化量：

Lheel x-rot 0.840829

Lheel y-rot 0.282551

Lheel z-rot -0.222619

Lknee -rot 0.161997

Lhip x-rot -0.169768

Lhip y-rot 0.336034

Lhip z-rot 0.663801

Rhip x-rot 0.771976

Rhip y-rot -0.114476

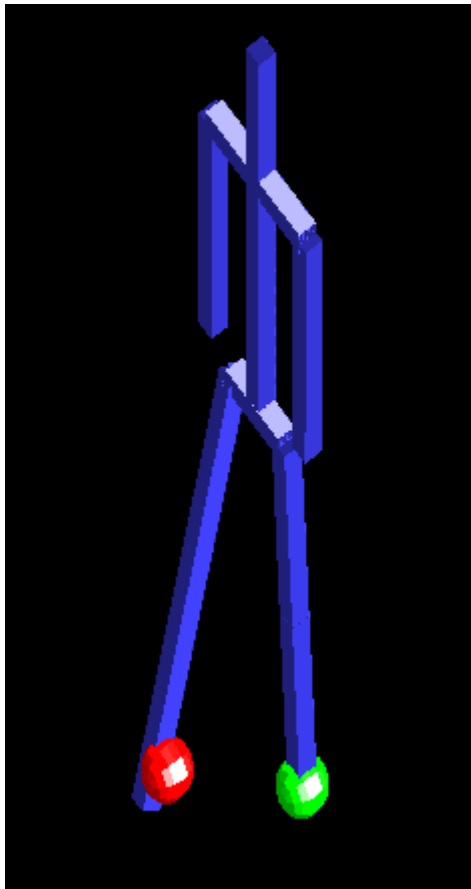
Rhip z-rot 0

Rknee -rot 0

位置誤差：

RightHeel Position Error = 14.8045

Iteration No.5



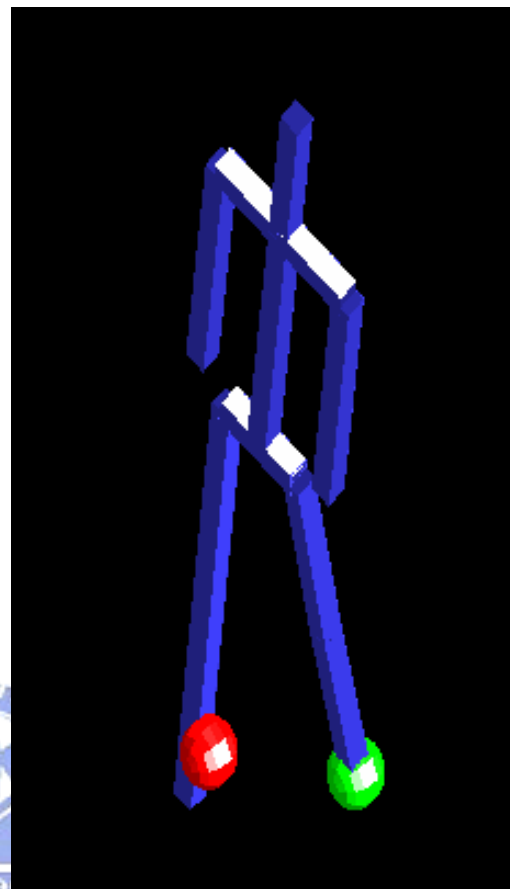
角度變化量：

Lhip z-rot 4.30858

位置誤差：

RightHeel Position Error = 8.12779

Iteration No.5



角度變化量：

Lheel x-rot 0.99132

Lheel y-rot 0.285791

Lheel z-rot -0.0933667

Lknee -rot 0.0428345

Lhip x-rot -0.564178

Lhip y-rot 0.231641

Lhip z-rot 0.119211

Rhip x-rot 0.755768

Rhip y-rot -0.111394

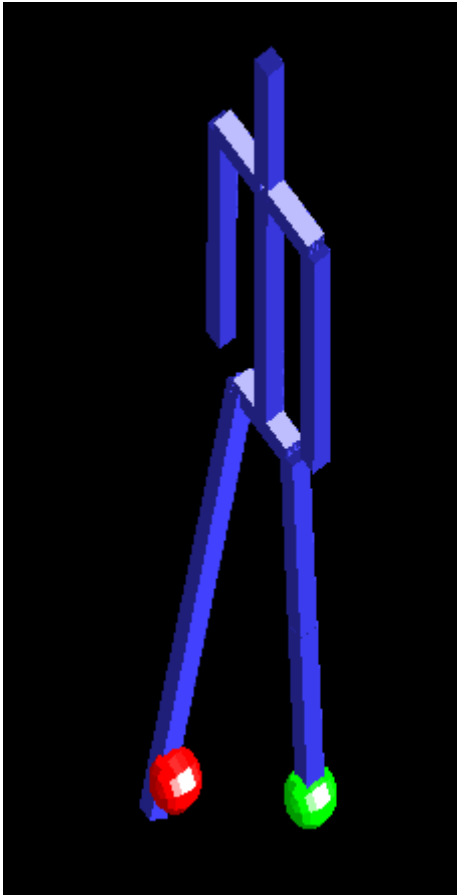
Rhip z-rot 0

Rknee -rot 0

位置誤差：

RightHeel Position Error = 14.2786

Iteration No.6



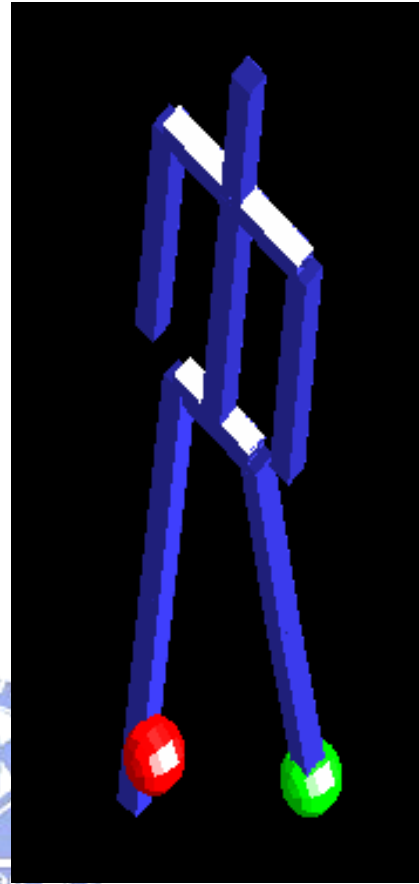
角度變化量：

Lhip y-rot -0.0486329

位置誤差：

RightHeel Position Error = 8.04854

Iteration No.6



角度變化量：

Lheel x-rot 1.04266

Lheel y-rot 0.249066

Lheel z-rot -0.0561243

Lknee -rot 0.0366933

Lhip x-rot -0.659456

Lhip y-rot 0.150806

Lhip z-rot -0.105872

Rhip x-rot 0.757727

Rhip y-rot -0.111233

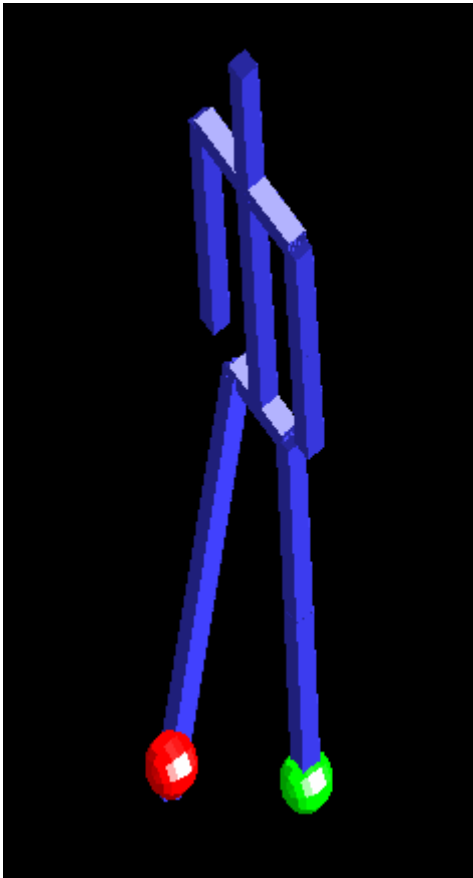
Rhip z-rot 0

Rknee -rot 0

位置誤差：

RightHeel Position Error = 13.7037

Iteration No.7



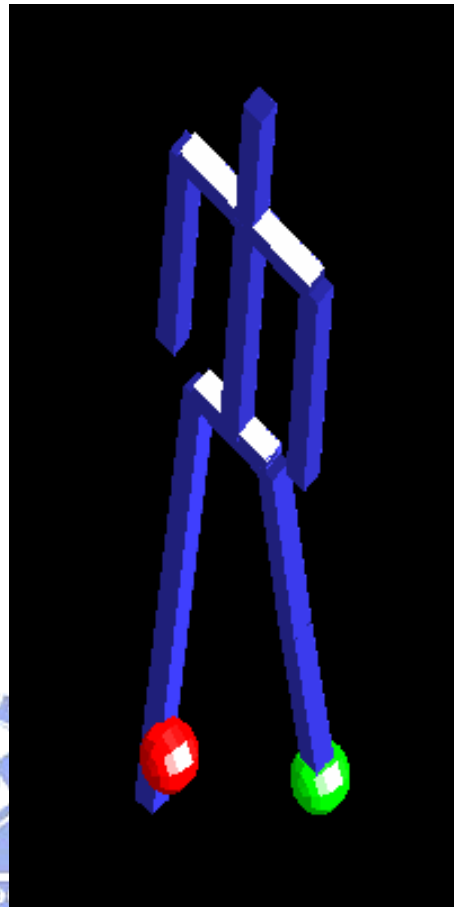
角度變化量：

Lhip x-rot -4.18778

位置誤差：

RightHeel Position Error = 8.01972

Iteration No.7



角度變化量：

Lheel x-rot 1.02434

Lheel y-rot 0.199306

Lheel z-rot -0.0509009

Lknee -rot 0.0420619

Lhip x-rot -0.653654

Lhip y-rot 0.0889665

Lhip z-rot -0.190526

Rhip x-rot 0.756258

Rhip y-rot -0.110686

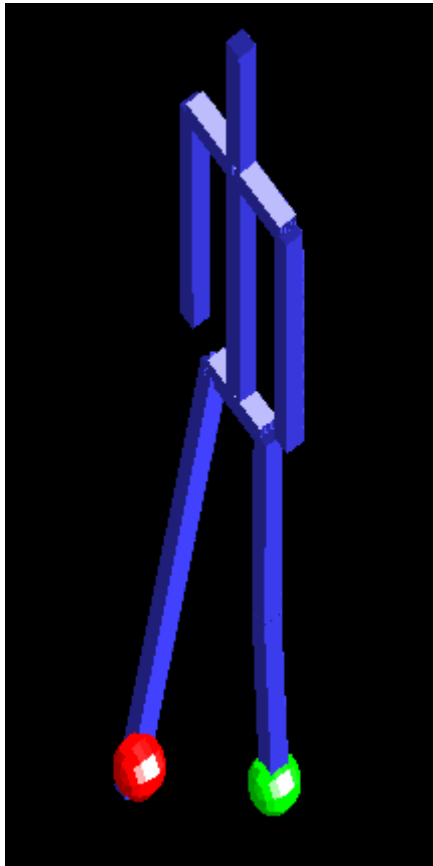
Rhip z-rot 0

Rknee -rot 0

位置誤差：

RightHeel Position Error = 13.1278

Iteration No.8



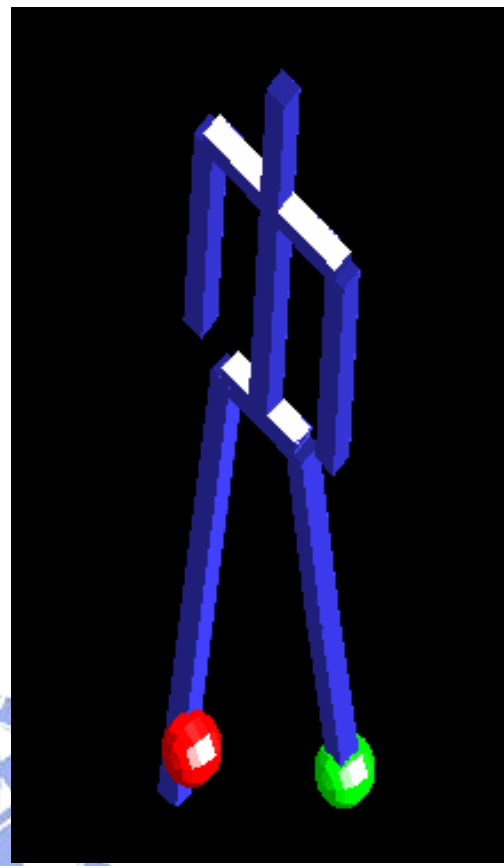
角度變化量：

Lknee -rot 4.87742

位置誤差：

RightHeel Position Error = 6.95053

Iteration No.8



角度變化量：

Lheel x-rot 0.979504

Lheel y-rot 0.147531

Lheel z-rot -0.0557757

Lknee -rot 0.0469089

Lhip x-rot -0.614823

Lhip y-rot 0.0385627

Lhip z-rot -0.212783

Rhip x-rot 0.743622

Rhip y-rot -0.108551

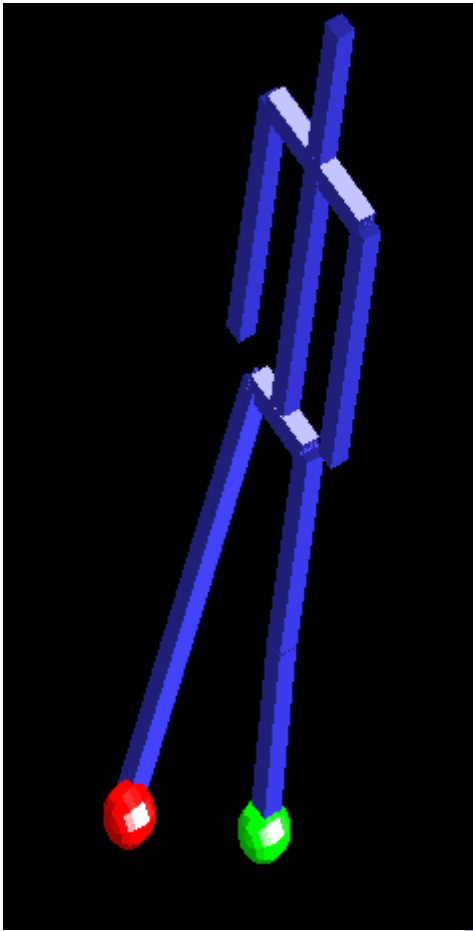
Rhip z-rot 0

Rknee -rot 0

位置誤差：

RightHeel Position Error = 12.5741

Iteration No.11



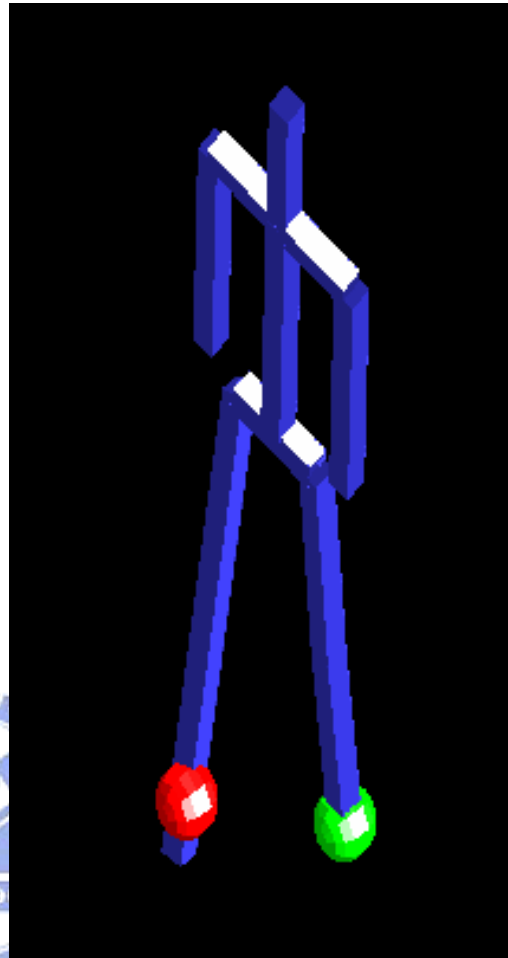
角度變化量：

Lheel x-rot 9.55858

位置誤差：

RightHeel Position Error = 2.11822

Iteration No.11



角度變化量：

Lheel x-rot 0.82358

Lheel y-rot 0.0196092

Lheel z-rot -0.0661409

Lknee -rot 0.0469312

Lhip x-rot -0.483446

Lhip y-rot -0.067093

Lhip z-rot -0.177164

Rhip x-rot 0.658458

Rhip y-rot -0.0953614

Rhip z-rot -1.40849

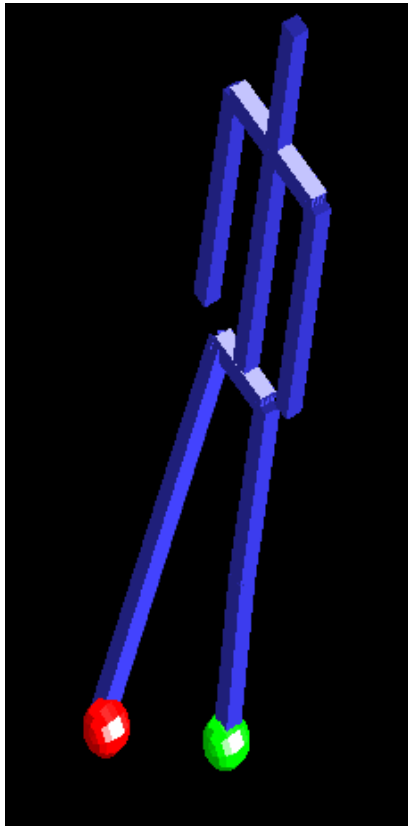
Rknee -rot 0

位置誤差：

RightHeel Position Error = 10.7051



Iteration No.15



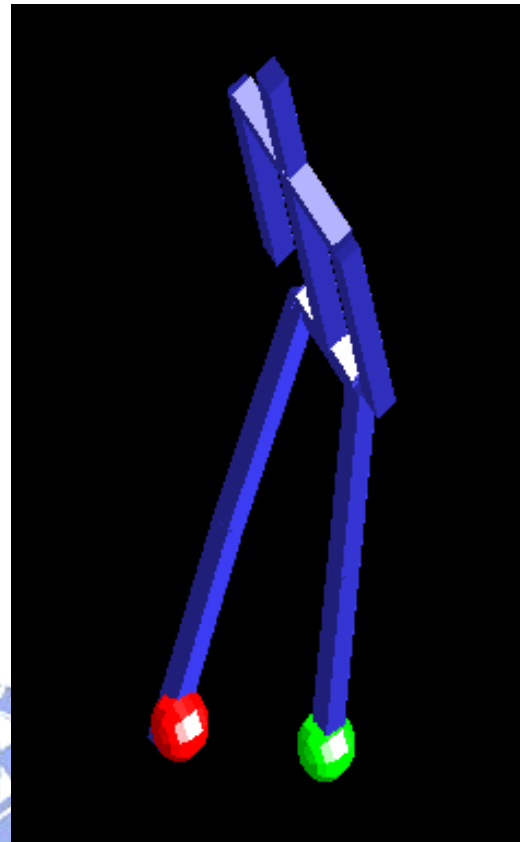
角度變化量：

Rhip x-rot 0.42321

位置誤差：

RightHeel Position Error = 1.96375

Iteration No.133



角度變化量：

Lheel x-rot -0.000651478

Lheel y-rot -0.00806797

Lheel z-rot 0.00472887

Lknee -rot 0

Lhip x-rot 0

Lhip y-rot -0.00708367

Lhip z-rot -0.0038579

Rhip x-rot 0.0408976

Rhip y-rot -0.0203935

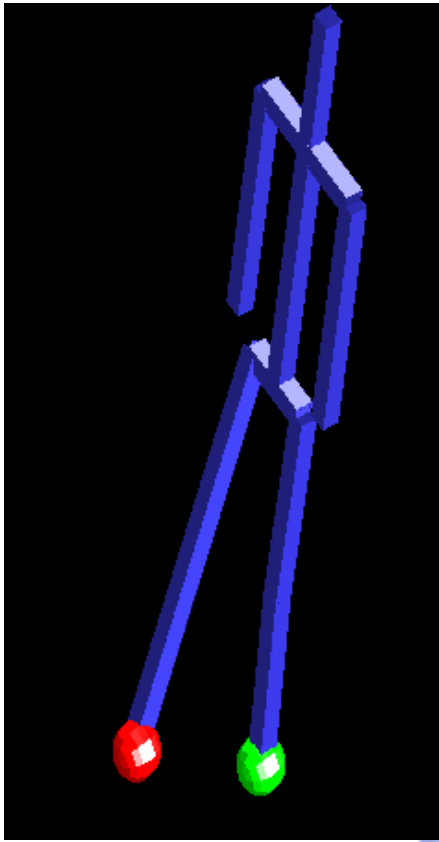
Rhip z-rot 0

Rknee -rot -0.0907866

位置誤差：

RightHeel Position Error = 4.81533

Iteration No.16



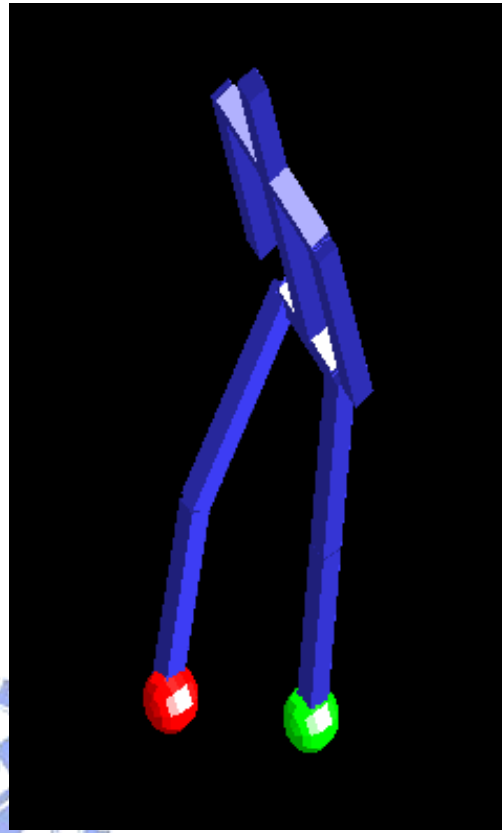
角度變化量：

Lhip z-rot -1.21647

位置誤差：

RightHeel Position Error = 0.887057

Iteration No.225



角度變化量：

Lheel x-rot -0.0134826

Lheel y-rot 0.00197068

Lheel z-rot -0.00972108

Lknee -rot 0

Lhip x-rot 0

Lhip y-rot 0.00104673

Lhip z-rot 0.0146198

Rhip x-rot 0.14414

Rhip y-rot 0

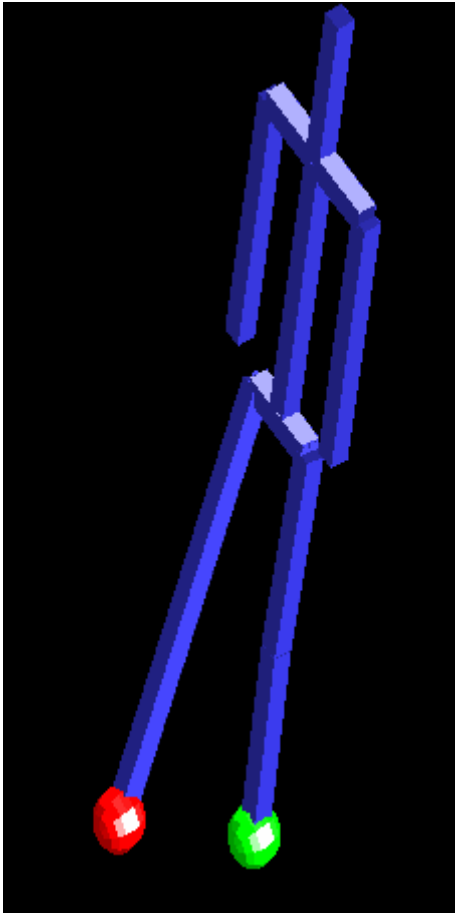
Rhip z-rot 0

Rknee -rot -0.311015

位置誤差：

RightHeel Position Error = 3.12473

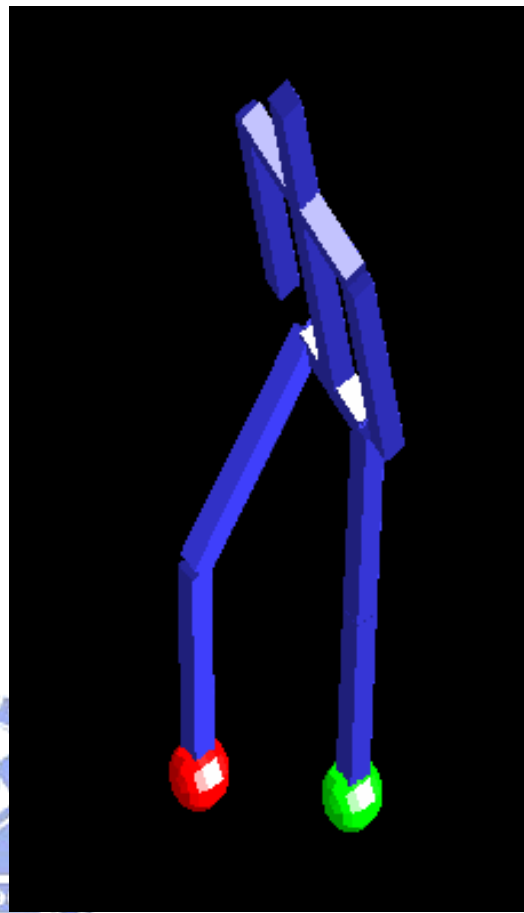
IK iteration 完成後 : (25 次)



Lheel x-rot	10.2497
Lheel y-rot	-0.00693399
Lheel z-rot	3.24742
Lknee -rot	5.01156
Lhip x-rot	-4.39744
Lhip y-rot	-0.107036
Lhip z-rot	3.1022
Rhip x-rot	24.309
Rhip y-rot	-9.44547
Rhip z-rot	-9.92442
Rknee -rot	0

RightHeel Position Error = 0.577684

IK iteration 完成後 : (341 次)



Lheel x-rot	0.794662
Lheel y-rot	-0.476253
Lheel z-rot	5.0969
Lknee -rot	0.289229
Lhip x-rot	-6.71698
Lhip y-rot	-0.406957
Lhip z-rot	6.34895
Rhip x-rot	52.6863
Rhip y-rot	4.79646
Rhip z-rot	-2.04797
Rknee -rot	41.4306

RightHeel Position Error = 0.310719

實際立體影像：



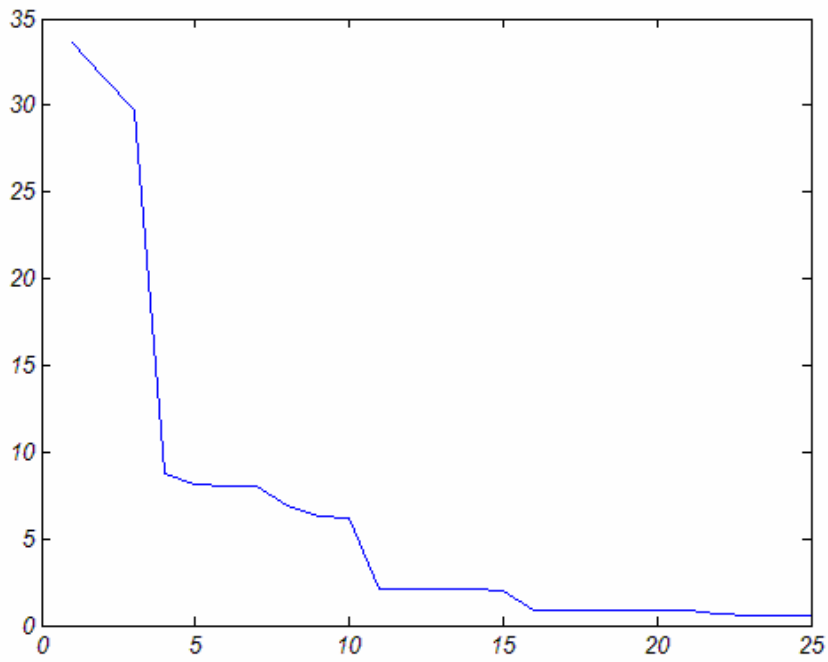
View1



View2

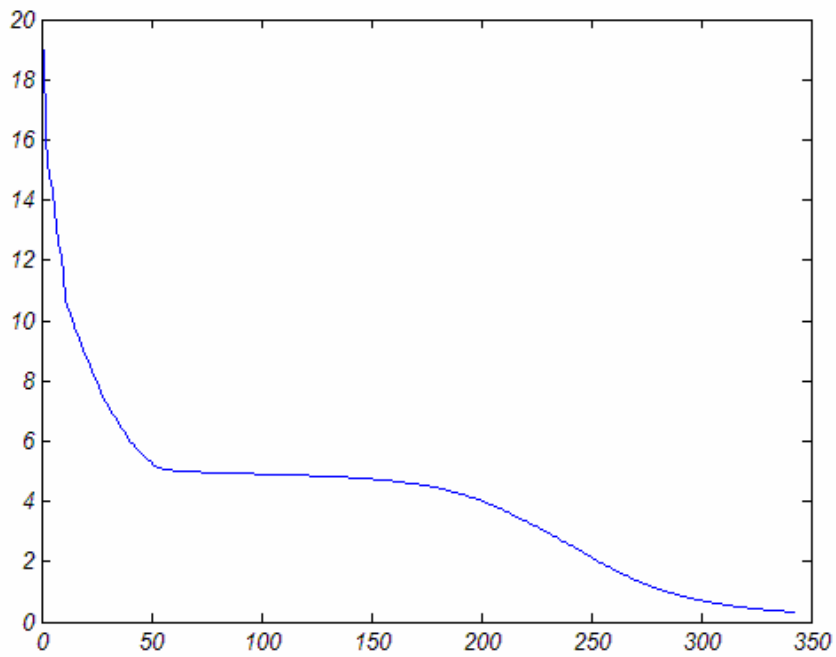


(a1) CCD Method



(橫軸爲 iteration，縱軸爲右腳跟的 position error)

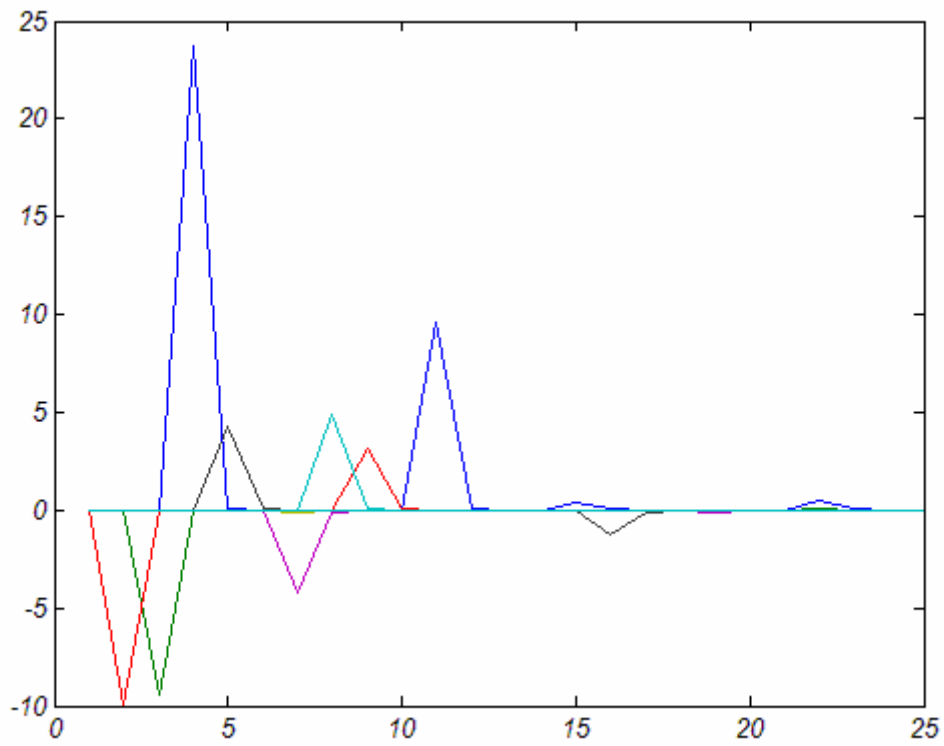
(a2) SDLS Method



(橫軸爲 iteration，縱軸爲右腳跟的 position error)

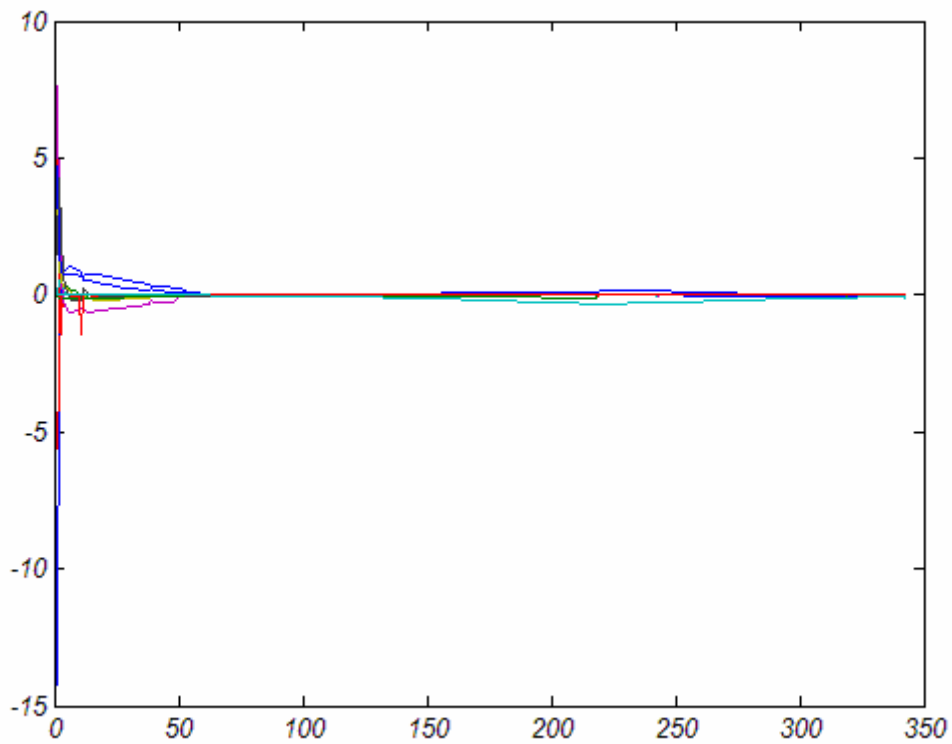
右腳跟位置誤差收斂比較圖

CCD Method 的  $\Delta\theta$  收斂情形：



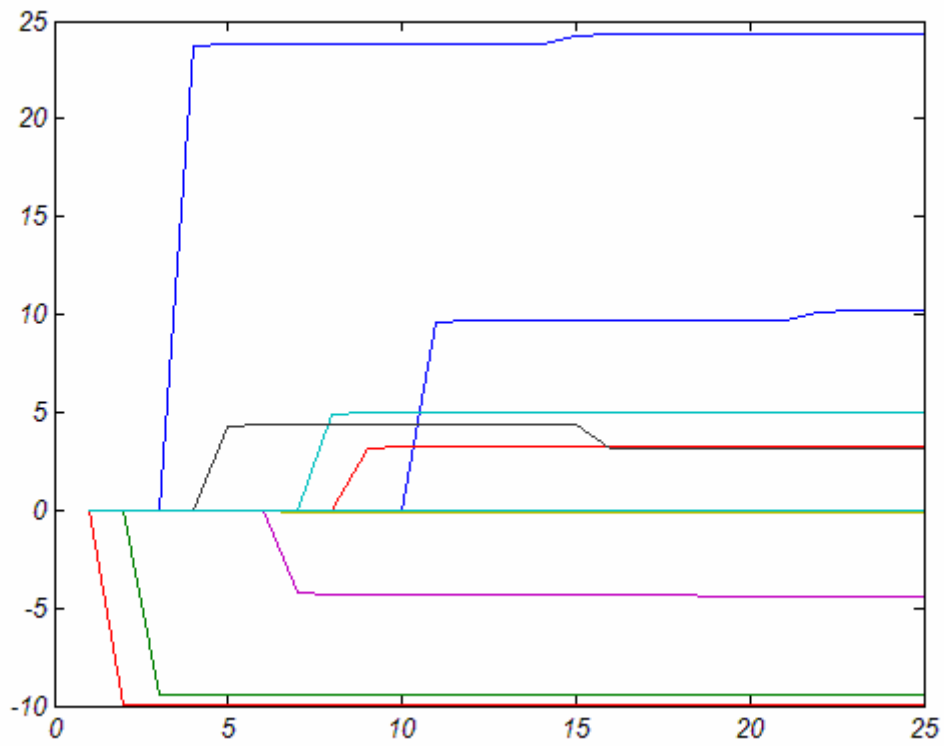
(橫軸為 iteration，縱軸為各 joint 的  $\Delta\theta$  值)

SDLS Method 的  $\Delta\theta$  收斂情形：



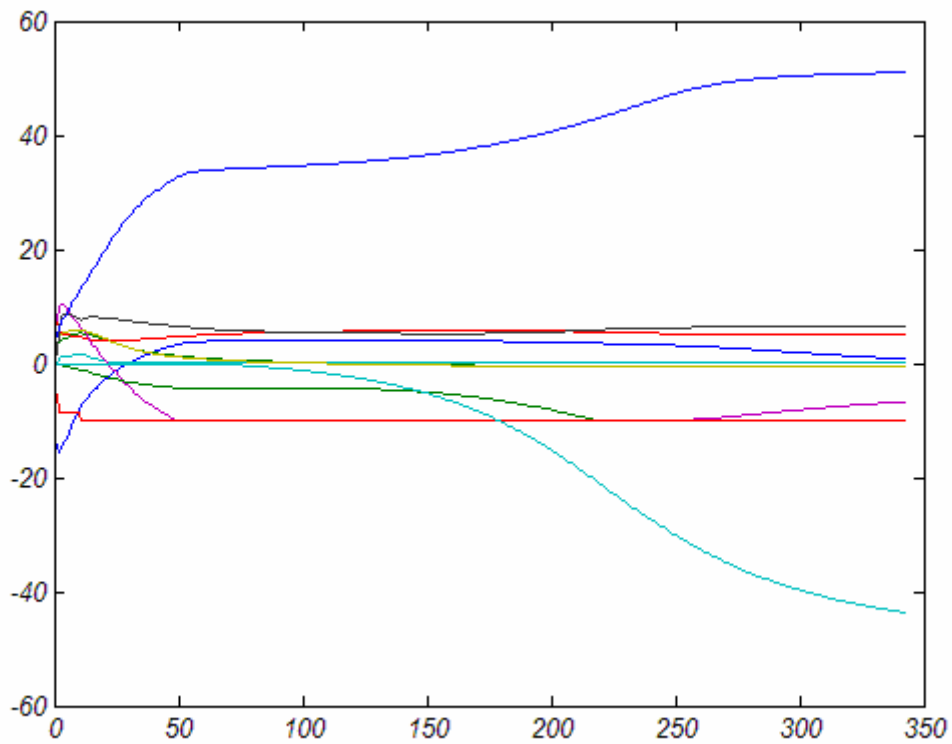
(橫軸為 iteration，縱軸為各 joint 的  $\Delta\theta$  值)

CCD Method 的  $\theta$  收斂情形：



(橫軸為 iteration，縱軸為各 joint 的  $\theta$  值)

SDLS Method 的  $\theta$  收斂情形：



(橫軸為 iteration，縱軸為各 joint 的  $\theta$  值)

實驗結果：

SDLS 所需的 iteration 多很多，是由於右膝蓋的 constraint。

SDLS 最後雖然仍然能夠修正到正確的姿勢。

但仍會因為右膝蓋的 constraint 需花較多的 iteration 修正。

CCD Method	SDLS Method
IK iteration 所需次數較少	IK iteration 所需次數較多
End-Effectors 到達 Target 速度較快	End-Effectors 到達 Target 速度較慢
$\theta$ 的單次的變化較大	$\theta$ 的單次的變化較小
$\Delta\theta$ 收斂情形較快	$\Delta\theta$ 收斂情形較慢
最後結果雖然到達 Target，但卻不貼近真實	最後結果到達 Target 且貼近真實





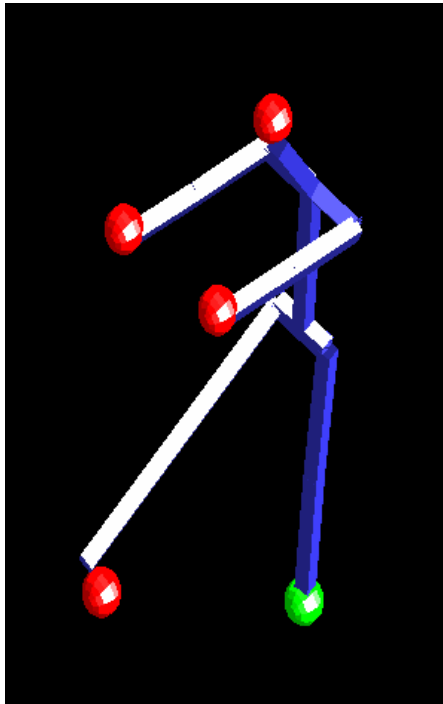
## 6.4. 同以 IP 為初解，比較 DLS 與 SDLS 的差異

爲了要了解 damped least square 與 selectively damped least square 的差別，因此同以 IP 為初解，分別使用 DLS 與 SDLS 的方法做 IK iteration。

實驗結果如下：



Selective Damped Least Square :



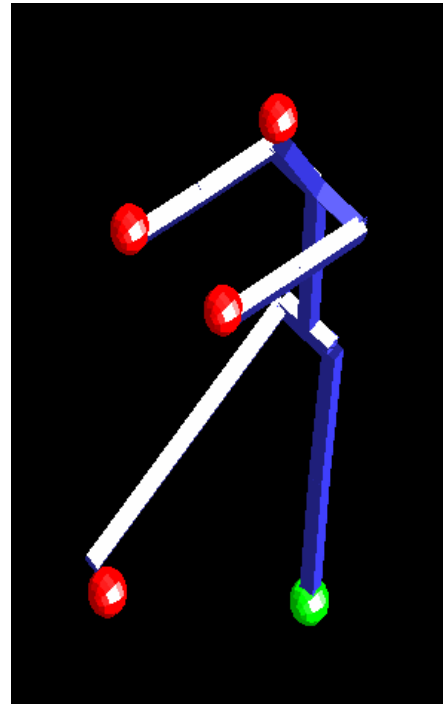
Initial Solution 時各關節角度 :

Lheel x-rot	-1.54662
Lheel y-rot	9.20609
Lheel z-rot	6.16833
Lknee -rot	2.64595
Lhip x-rot	-4.64455
Lhip y-rot	-4.91797
Lhip z-rot	2.40443
Navel -rot	-3.02843
Neck -rot	-27.4671
Lshoulder x-rot	129.336
Lshoulder y-rot	13.0242
Lshoulder z-rot	8.68211
Leibow -rot	0.0223306
Rshoulder x-rot	88.026
Rshoulder y-rot	17.2408
Rshoulder z-rot	15.5562
Relbow -rot	4.60507
Rhip x-rot	70.571
Rhip y-rot	19.0028
Rhip z-rot	3.71546
Rknee -rot	-0.0055452

位置誤差 :

LeftWrist	Position Error = 2.37451
Head	Position Error = 5.74403
RightWrist	Position Error = 2.81116
RightHeel	Position Error = 8.30422
Total Position Error	= 19.2339

Damped Least Square :



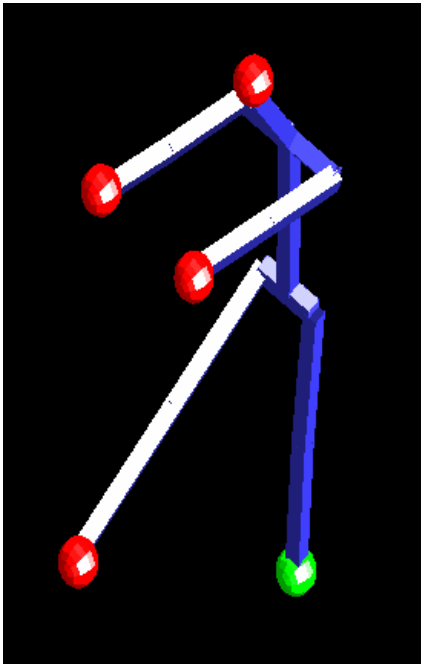
Initial Solution 時各關節角度 :

Lheel x-rot	-1.54662
Lheel y-rot	9.20609
Lheel z-rot	6.16833
Lknee -rot	2.64595
Lhip x-rot	-4.64455
Lhip y-rot	-4.91797
Lhip z-rot	2.40443
Navel -rot	-3.02843
Neck -rot	-27.4671
Lshoulder x-rot	129.336
Lshoulder y-rot	13.0242
Lshoulder z-rot	8.68211
Leibow -rot	0.0223306
Rshoulder x-rot	88.026
Rshoulder y-rot	17.2408
Rshoulder z-rot	15.5562
Relbow -rot	4.60507
Rhip x-rot	70.571
Rhip y-rot	19.0028
Rhip z-rot	3.71546
Rknee -rot	-0.0055452

位置誤差 :

LeftWrist	Position Error = 2.37451
Head	Position Error = 5.74403
RightWrist	Position Error = 2.81116
RightHeel	Position Error = 8.30422
Total Position Error	= 19.2339

Iteration No.1



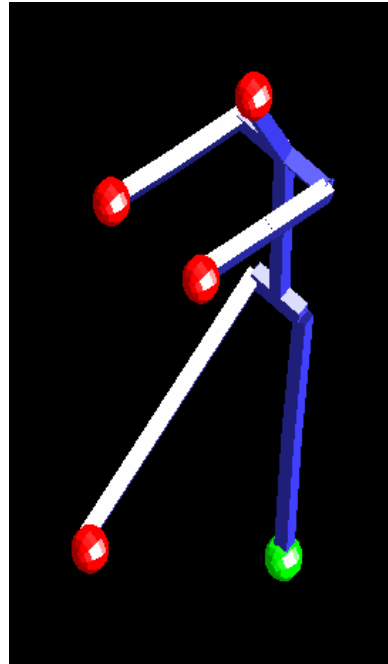
角度變化量：

Lheel x-rot	-0.865957
Lheel y-rot	0.492133
Lheel z-rot	0.689416
Lknee -rot	0.116895
Lhip x-rot	1.17362
Lhip y-rot	0.586004
Lhip z-rot	0.027123
Navel -rot	1.30104
Neck -rot	0.374008
Lshoulder x-rot	-0.427835
Lshoulder y-rot	-0.0507532
Lshoulder z-rot	-0.479033
Lelbow -rot	3.10601e-018
Rshoulder x-rot	0.314963
Rshoulder y-rot	0.0158025
Rshoulder z-rot	0.569135
Relbow -rot	0.0967705
Rhip x-rot	-0.123619
Rhip y-rot	0.00499919
Rhip z-rot	-0.827953
Rknee -rot	0

Iteration 後的 Error：

LeftWrist	Position Error = 2.75829
Head	Position Error = 2.37155
RightWrist	Position Error = 2.21502
RightHeel	Position Error = 2.13941
Total Position Error	= 9.48426

Iteration No.1



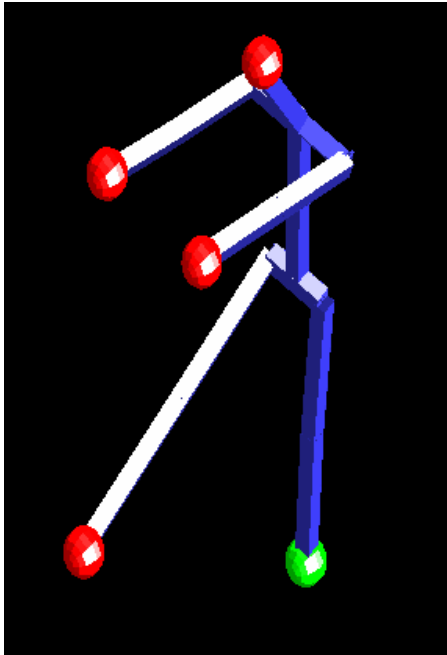
角度變化量：

Lheel x-rot	-0.332067
Lheel y-rot	0
Lheel z-rot	-0.543987
Lknee -rot	0.369082
Lhip x-rot	0.896738
Lhip y-rot	0.99705
Lhip z-rot	-1.82439
Navel -rot	0.644493
Neck -rot	0.187156
Lshoulder x-rot	-0.532279
Lshoulder y-rot	-0.0812514
Lshoulder z-rot	-0.0612277
Lelbow -rot	0
Rshoulder x-rot	0.144434
Rshoulder y-rot	0.00374821
Rshoulder z-rot	0.299319
Relbow -rot	0.0621731
Rhip x-rot	0.139752
Rhip y-rot	0.0630137
Rhip z-rot	-2.39379
Rknee -rot	0

Iteration 後的 Error：

LeftWrist	Position Error = 2.38809
Head	Position Error = 2.20685
RightWrist	Position Error = 2.14207
RightHeel	Position Error = 3.59462
Total Position Error	= 10.3316

Iteration No.2



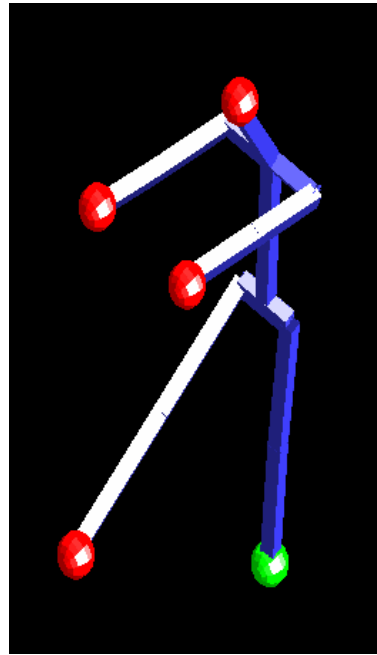
角度變化量：

Lheel x-rot	-0.633606
Lheel y-rot	-0.114958
Lheel z-rot	0.303056
Lknee -rot	-0.0295282
Lhip x-rot	0.646432
Lhip y-rot	-0.0389425
Lhip z-rot	-0.0522016
Navel -rot	0.679171
Neck -rot	0.187071
Lshoulder x-rot	-0.537025
Lshoulder y-rot	-0.0718573
Lshoulder z-rot	-0.258374
Lelbow -rot	2.48481e-017
Rshoulder x-rot	0.0835114
Rshoulder y-rot	-0.0141025
Rshoulder z-rot	0.294645
Relbow -rot	0.01897
Rhip x-rot	-0.034664
Rhip y-rot	0.00210709
Rhip z-rot	-0.454104
Rknee -rot	0

Iteration 後的 Error：

LeftWrist	Position Error = 2.60359
Head	Position Error = 1.88747
RightWrist	Position Error = 2.14104
RightHeel	Position Error = 1.07848
Total Position Error	= 7.71058

Iteration No.2



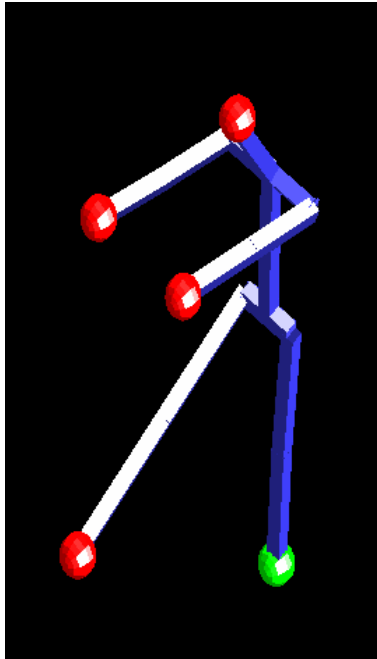
角度變化量：

Lheel x-rot	-0.39599
Lheel y-rot	0.118622
Lheel z-rot	0.341125
Lknee -rot	0.0432126
Lhip x-rot	0.537447
Lhip y-rot	0.191853
Lhip z-rot	-0.558392
Navel -rot	0.367516
Neck -rot	0.107201
Lshoulder x-rot	-0.528455
Lshoulder y-rot	-0.0762212
Lshoulder z-rot	-0.170826
Lelbow -rot	0
Rshoulder x-rot	0.046264
Rshoulder y-rot	0.0162505
Rshoulder z-rot	-0.0190939
Relbow -rot	0.0611072
Rhip x-rot	0.134836
Rhip y-rot	0.0110969
Rhip z-rot	-0.984978
Rknee -rot	0

Iteration 後的 Error：

LeftWrist	Position Error = 2.48343
Head	Position Error = 1.84696
RightWrist	Position Error = 2.05734
RightHeel	Position Error = 1.41781
Total Position Error	= 7.80554

Iteration No.3



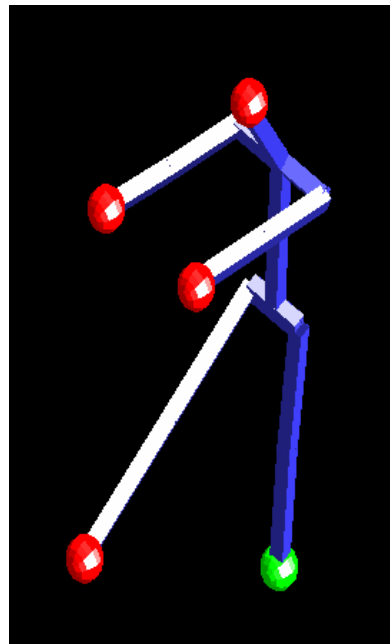
角度變化量：

Lheel x-rot	-0.28023
Lheel y-rot	-0.284875
Lheel z-rot	0.207735
Lknee -rot	-0.037033
Lhip x-rot	0.299277
Lhip y-rot	-0.243053
Lhip z-rot	-0.111674
Navel -rot	0.270962
Neck -rot	0.214569
Lshoulder x-rot	-0.547212
Lshoulder y-rot	-0.0761672
Lshoulder z-rot	-0.103018
Lelbow -rot	0
Rshoulder x-rot	-0.0276229
Rshoulder y-rot	-0.0123013
Rshoulder z-rot	0.0322107
Relbow -rot	0.000612739
Rhip x-rot	0.0225875
Rhip y-rot	0.00156348
Rhip z-rot	-0.405163
Rknee -rot	0

Iteration 後的 Error：

LeftWrist	Position Error = 2.33823
Head	Position Error = 1.6847
RightWrist	Position Error = 2.07555
RightHeel	Position Error = 0.822963
Total Position Error	= 6.92145

Iteration No.3



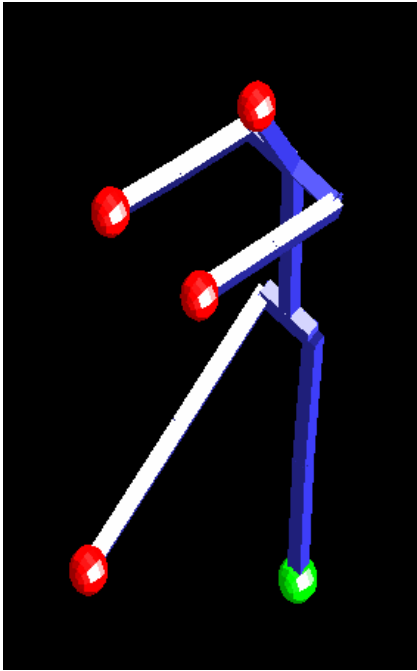
角度變化量：

Lheel x-rot	-0.287103
Lheel y-rot	-0.254315
Lheel z-rot	0.245031
Lknee -rot	-0.000926229
Lhip x-rot	0.365533
Lhip y-rot	-0.203897
Lhip z-rot	-0.308209
Navel -rot	0.22949
Neck -rot	0.0834097
Lshoulder x-rot	-0.474158
Lshoulder y-rot	-0.0667774
Lshoulder z-rot	-0.160493
Lelbow -rot	0
Rshoulder x-rot	-0.0216946
Rshoulder y-rot	0.0096029
Rshoulder z-rot	-0.122448
Relbow -rot	0.0403235
Rhip x-rot	0.116654
Rhip y-rot	0.00168476
Rhip z-rot	-0.483852
Rknee -rot	0

Iteration 後的 Error：

LeftWrist	Position Error = 2.27563
Head	Position Error = 1.68083
RightWrist	Position Error = 1.96487
RightHeel	Position Error = 0.934999
Total Position Error	= 6.85633

Iteration No.4



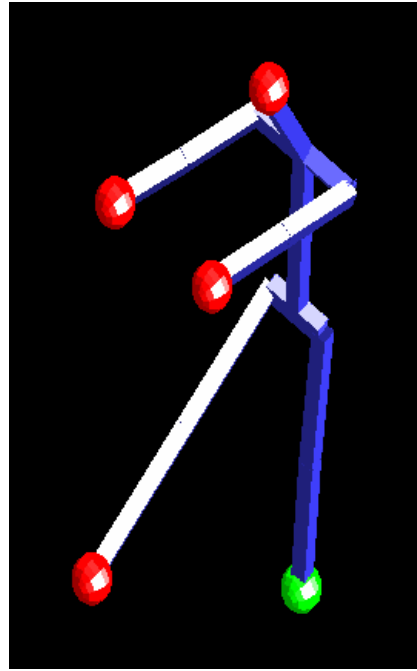
角度變化量：

Lheel x-rot	-0.193084
Lheel y-rot	-0.281883
Lheel z-rot	0.198018
Lknee -rot	-0.0346571
Lhip x-rot	0.220191
Lhip y-rot	-0.248605
Lhip z-rot	-0.134459
Navel -rot	0.154291
Neck -rot	0.259273
Lshoulder x-rot	-0.488109
Lshoulder y-rot	-0.06849
Lshoulder z-rot	-0.0471322
Lelbow -rot	0
Rshoulder x-rot	-0.0859496
Rshoulder y-rot	-0.0131154
Rshoulder z-rot	-0.0921078
Relbow -rot	-0.0109465
Rhip x-rot	0.0586141
Rhip y-rot	0.000470071
Rhip z-rot	-0.379437
Rknee -rot	0

Iteration 後的 Error：

LeftWrist	Position Error = 2.09197
Head	Position Error = 1.52975
RightWrist	Position Error = 1.9898
RightHeel	Position Error = 0.685835
Total Position Error	= 6.29736

Iteration No.4



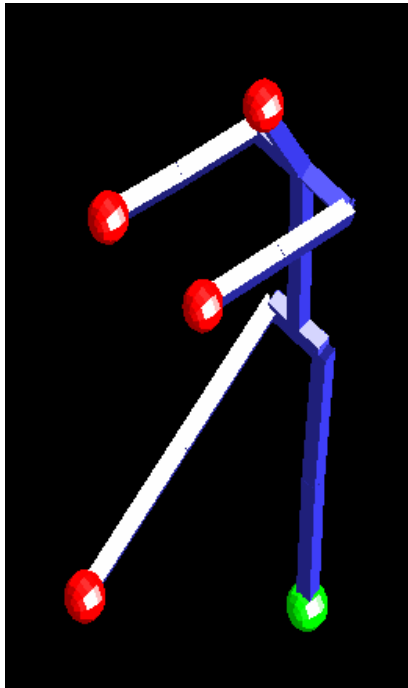
角度變化量：

Lheel x-rot	-0.217568
Lheel y-rot	-0.280301
Lheel z-rot	0.222821
Lknee -rot	-0.0109191
Lhip x-rot	0.276999
Lhip y-rot	-0.239944
Lhip z-rot	-0.248775
Navel -rot	0.16742
Neck -rot	0.0900001
Lshoulder x-rot	-0.407434
Lshoulder y-rot	-0.0550601
Lshoulder z-rot	-0.174632
Lelbow -rot	0
Rshoulder x-rot	-0.05302
Rshoulder y-rot	0.00803783
Rshoulder z-rot	-0.180983
Relbow -rot	0.0312902
Rhip x-rot	0.0943268
Rhip y-rot	-0.000583632
Rhip z-rot	-0.349017
Rknee -rot	0

Iteration 後的 Error：

LeftWrist	Position Error = 2.05501
Head	Position Error = 1.56947
RightWrist	Position Error = 1.84836
RightHeel	Position Error = 0.715575
Total Position Error	= 6.18841

Iteration No.5



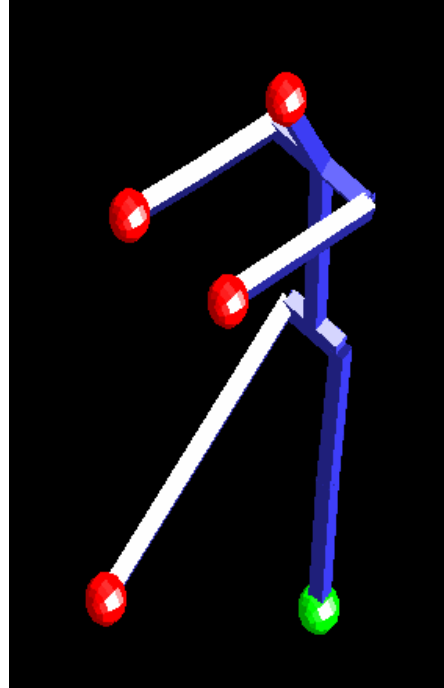
角度變化量：

Lheel x-rot	-0.163811
Lheel y-rot	-0.267266
Lheel z-rot	0.18733
Lknee -rot	-0.0304769
Lhip x-rot	0.196746
Lhip y-rot	-0.237297
Lhip z-rot	-0.143428
Navel -rot	0.113604
Neck -rot	0.269786
Lshoulder x-rot	-0.443478
Lshoulder y-rot	-0.0614706
Lshoulder z-rot	-0.0546444
Lelbow -rot	0
Rshoulder x-rot	-0.105731
Rshoulder y-rot	-0.013086
Rshoulder z-rot	-0.13582
Relbow -rot	-0.0145444
Rhip x-rot	0.0751613
Rhip y-rot	-0.000848261
Rhip z-rot	-0.351829
Rknee -rot	0

Iteration 後的 Error：

LeftWrist	Position Error = 1.87381
Head	Position Error = 1.40063
RightWrist	Position Error = 1.89672
RightHeel	Position Error = 0.575107
Total Position Error	= 5.74627

Iteration No.5



角度變化量：

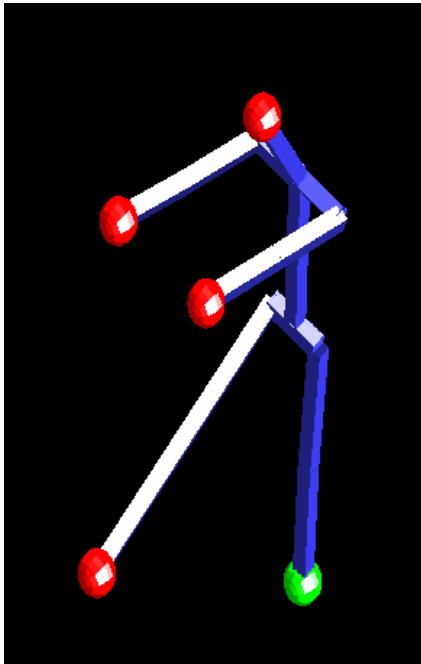
Lheel x-rot	-0.173859
Lheel y-rot	-0.258671
Lheel z-rot	0.196668
Lknee -rot	-0.0133389
Lhip x-rot	0.22383
Lhip y-rot	-0.225024
Lhip z-rot	-0.225969
Navel -rot	0.131443
Neck -rot	0.0967008
Lshoulder x-rot	-0.347644
Lshoulder y-rot	-0.0448695
Lshoulder z-rot	-0.183439
Lelbow -rot	0
Rshoulder x-rot	-0.0613108
Rshoulder y-rot	0.00934922
Rshoulder z-rot	-0.209268
Relbow -rot	0.0296903
Rhip x-rot	0.0785884
Rhip y-rot	-0.00159838
Rhip z-rot	-0.288466
Rknee -rot	0

Iteration 後的 Error：

LeftWrist	Position Error = 1.86022
Head	Position Error = 1.47943
RightWrist	Position Error = 1.73228
RightHeel	Position Error = 0.573228
Total Position Error	= 5.64515



Iteration No.8



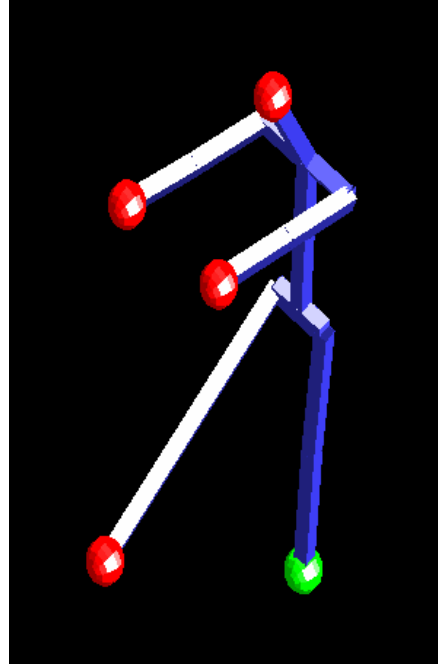
角度變化量：

Lheel x-rot	-0.111258
Lheel y-rot	-0.217883
Lheel z-rot	0.144062
Lknee -rot	-0.0221094
Lhip x-rot	0.148579
Lhip y-rot	-0.195837
Lhip z-rot	-0.148631
Navel -rot	0.0629124
Neck -rot	0.214363
Lshoulder x-rot	-0.323609
Lshoulder y-rot	-0.0408487
Lshoulder z-rot	-0.127709
Lelbow -rot	0
Rshoulder x-rot	-0.0657563
Rshoulder y-rot	0.00613432
Rshoulder z-rot	-0.19391
Relbow -rot	0.0172803
Rhip x-rot	0.0768016
Rhip y-rot	-0.00373486
Rhip z-rot	-0.271162
Rknee -rot	0

Iteration 後的 Error：

LeftWrist	Position Error = 1.34158
Head	Position Error = 1.1538
RightWrist	Position Error = 1.62699
RightHeel	Position Error = 0.326843
Total Position Error	= 4.44921

Iteration No.8



角度變化量：

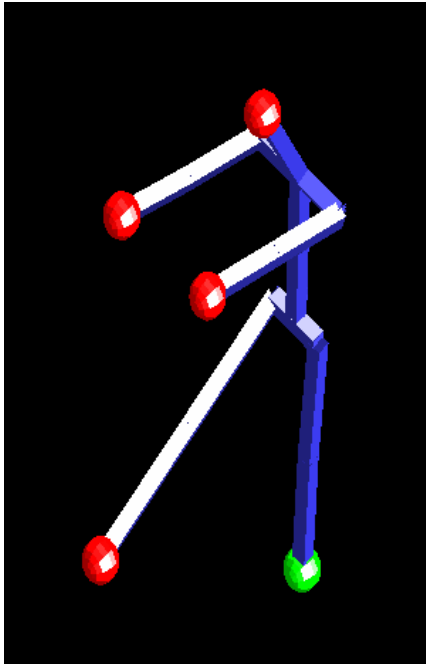
Lheel x-rot	-0.102103
Lheel y-rot	-0.190944
Lheel z-rot	0.133936
Lknee -rot	-0.0145028
Lhip x-rot	0.137631
Lhip y-rot	-0.169919
Lhip z-rot	-0.174693
Navel -rot	0.0715338
Neck -rot	0.0955559
Lshoulder x-rot	-0.218295
Lshoulder y-rot	-0.0242559
Lshoulder z-rot	-0.181458
Lelbow -rot	0
Rshoulder x-rot	-0.0383825
Rshoulder y-rot	0.016765
Rshoulder z-rot	-0.215992
Relbow -rot	0.0386279
Rhip x-rot	0.0553607
Rhip y-rot	-0.00282875
Rhip z-rot	-0.201816
Rknee -rot	0

Iteration 後的 Error：

LeftWrist	Position Error = 1.41099
Head	Position Error = 1.2786
RightWrist	Position Error = 1.42631
RightHeel	Position Error = 0.328519
Total Position Error	= 4.44442



Iteration No.13



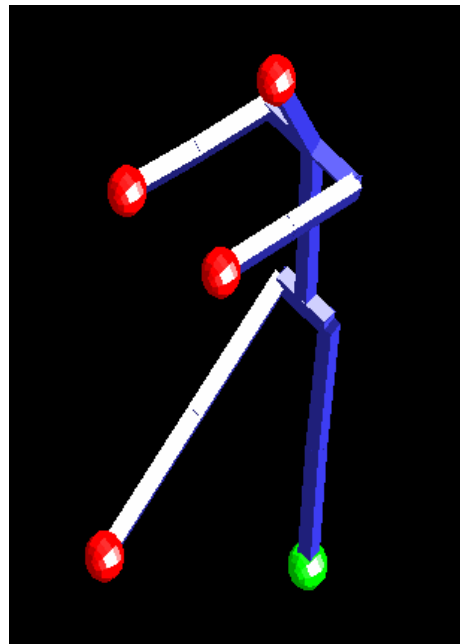
角度變化量：

Lheel x-rot	-0.0464111
Lheel y-rot	-0.157574
Lheel z-rot	0.0982325
Lknee -rot	-0.0163007
Lhip x-rot	0.0804657
Lhip y-rot	-0.146353
Lhip z-rot	-0.154368
Navel -rot	0.0141458
Neck -rot	0.0719657
Lshoulder x-rot	-0.0948053
Lshoulder y-rot	-0.00600287
Lshoulder z-rot	-0.210881
Lelbow -rot	0
Rshoulder x-rot	0.0279569
Rshoulder y-rot	0.0366744
Rshoulder z-rot	-0.238374
Relbow -rot	0.0792107
Rhip x-rot	0.0557655
Rhip y-rot	-0.00483026
Rhip z-rot	-0.172403
Rknee -rot	0

Iteration 後的 Error：

LeftWrist	Position Error = 0.776377
Head	Position Error = 0.984377
RightWrist	Position Error = 1.18822
RightHeel	Position Error = 0.180079
Total Position Error	= 3.12905

Iteration No.13



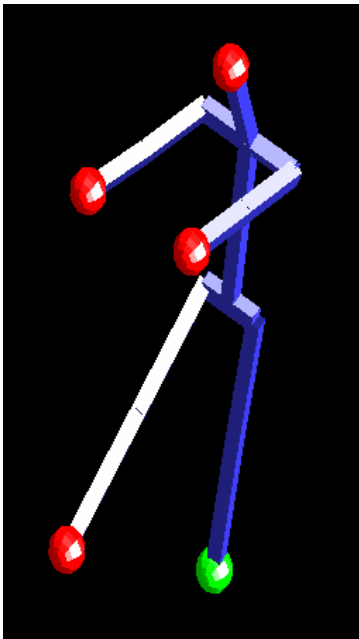
角度變化量：

Lheel x-rot	-0.0477366
Lheel y-rot	-0.123057
Lheel z-rot	0.07478
Lknee -rot	-0.0137638
Lhip x-rot	0.0724559
Lhip y-rot	-0.112633
Lhip z-rot	-0.107321
Navel -rot	0.0242665
Neck -rot	0.0720569
Lshoulder x-rot	-0.103274
Lshoulder y-rot	-0.00858616
Lshoulder z-rot	-0.146695
Lelbow -rot	0
Rshoulder x-rot	0.00163217
Rshoulder y-rot	0.021179
Rshoulder z-rot	-0.167972
Relbow -rot	0.0470786
Rhip x-rot	0.0418492
Rhip y-rot	-0.00342875
Rhip z-rot	-0.139103
Rknee -rot	0

Iteration 後的 Error：

LeftWrist	Position Error = 0.920239
Head	Position Error = 1.09497
RightWrist	Position Error = 1.06348
RightHeel	Position Error = 0.18831
Total Position Error	= 3.267

Iteration No.17



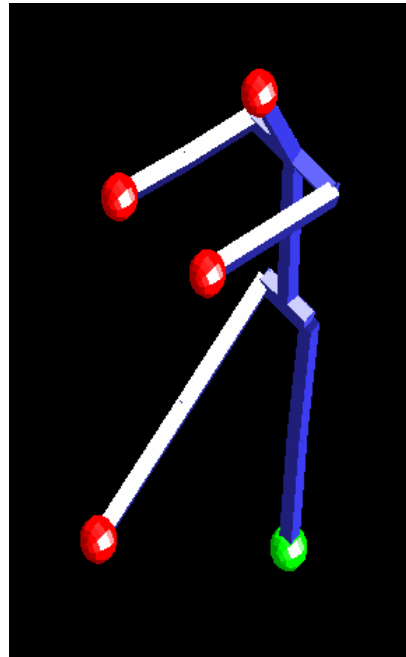
角度變化量：

Lheel x-rot	-0.0367081
Lheel y-rot	-0.149989
Lheel z-rot	0.0941778
Lknee -rot	-0.0167029
Lhip x-rot	0.0668415
Lhip y-rot	-0.140674
Lhip z-rot	-0.154327
Navel -rot	0.00431104
Neck -rot	0.0264593
Lshoulder x-rot	-0.00558947
Lshoulder y-rot	0.0043127
Lshoulder z-rot	-0.210064
Lelbow -rot	0.0203126
Rshoulder x-rot	0.04591
Rshoulder y-rot	0.0387597
Rshoulder z-rot	-0.227476
Relbow -rot	0.0900932
Rhip x-rot	0.0517783
Rhip y-rot	-0.0057799
Rhip z-rot	-0.160693
Rknee -rot	0

修正後位置 Error：

LeftWrist	Position Error = 0.432185
Head	Position Error = 0.933754
RightWrist	Position Error = 0.840133
RightHeel	Position Error = 0.147608
Total Position Error	= 2.35368

Iteration No.17



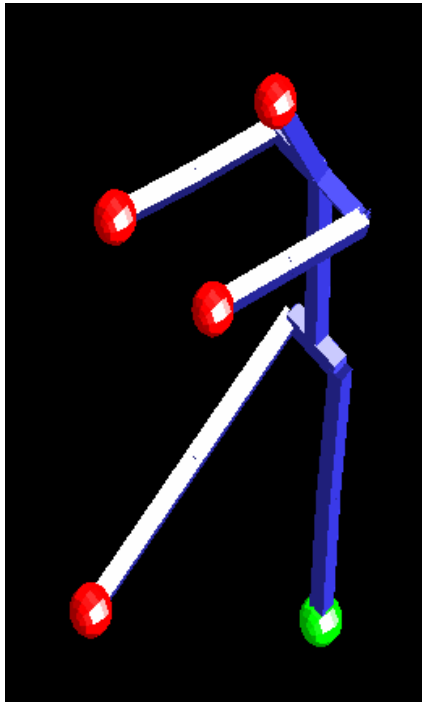
角度變化量：

Lheel x-rot	-0.0269606
Lheel y-rot	-0.0895507
Lheel z-rot	0.0480489
Lknee -rot	-0.0125575
Lhip x-rot	0.0470134
Lhip y-rot	-0.0833801
Lhip z-rot	-0.0707384
Navel -rot	0.00816345
Neck -rot	0.0561584
Lshoulder x-rot	-0.0546906
Lshoulder y-rot	-0.00321115
Lshoulder z-rot	-0.118482
Lelbow -rot	0
Rshoulder x-rot	0.0130837
Rshoulder y-rot	0.0192942
Rshoulder z-rot	-0.131688
Relbow -rot	0.0443873
Rhip x-rot	0.0350287
Rhip y-rot	-0.00345243
Rhip z-rot	-0.109869
Rknee -rot	0

修正後位置 Error：

LeftWrist	Position Error = 0.654775
Head	Position Error = 1.02182
RightWrist	Position Error = 0.865641
RightHeel	Position Error = 0.155281
Total Position Error	= 2.69752

修正後：(iteration 25 次)



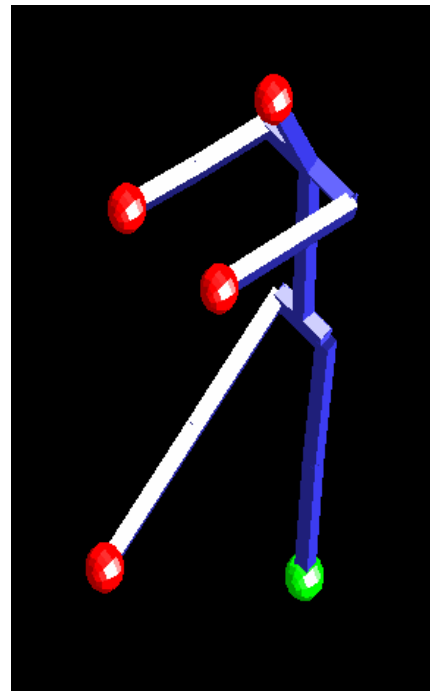
最後結果各關節角度：

Lheel x-rot	-3.98954
Lheel y-rot	5.58325
Lheel z-rot	8.39177
Lknee -rot	2.67047
Lhip x-rot	-0.633344
Lhip y-rot	-7.1514
Lhip z-rot	-3.26126
Navel -rot	-1.14754
Neck -rot	-25.1851
Lshoulder x-rot	124.796
Lshoulder y-rot	12.4884
Lshoulder z-rot	4.36112
Leibow -rot	0.251014
Rshoulder x-rot	88.3514
Rshoulder y-rot	17.8342
Rshoulder z-rot	11.303
Relbow -rot	6.10711
Rhip x-rot	72.1569
Rhip y-rot	19.1193
Rhip z-rot	-4.26445
Rknee -rot	-0.0055452

修正後位置 Error：

LeftWrist	Position Error = 0.144516
Head	Position Error = 0.916711
RightWrist	Position Error = 0.293559
RightHeel	Position Error = 0.0845181
Total Position Error	= 1.4393

修正後：(iteration 17 次)

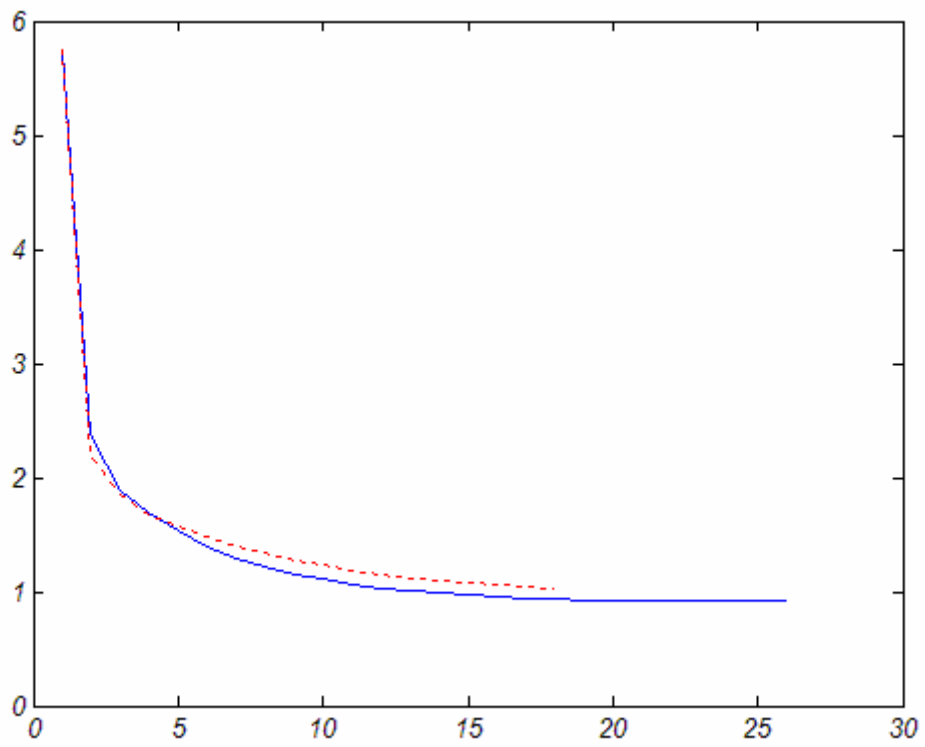


最後結果各關節角度：

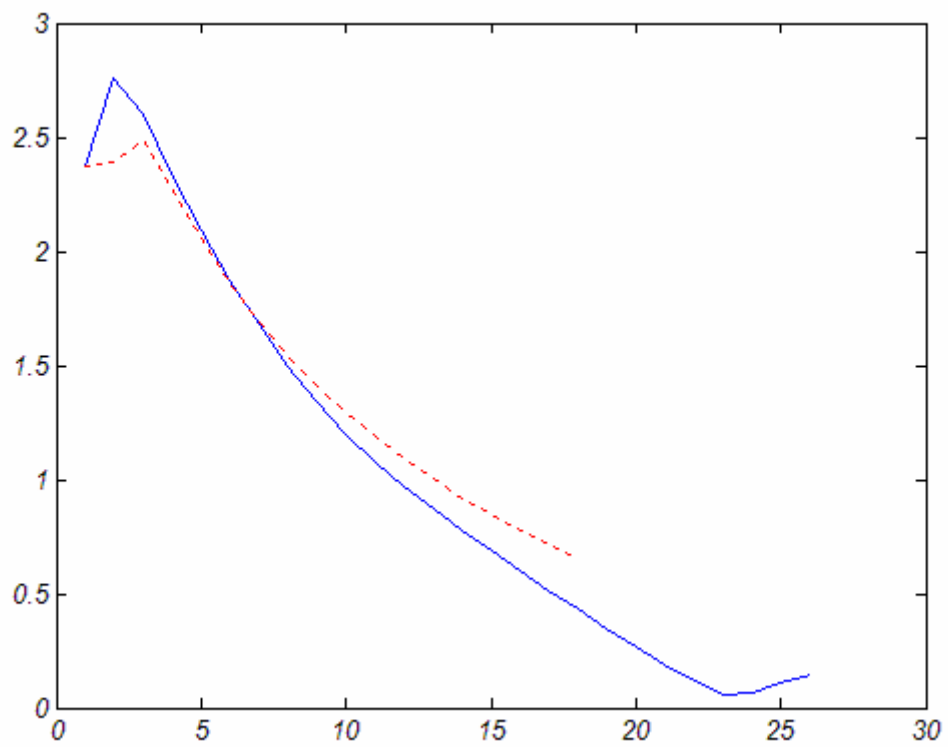
Lheel x-rot	-3.78288
Lheel y-rot	6.7569
Lheel z-rot	7.79291
Lknee -rot	2.86704
Lhip x-rot	-1.16762
Lhip y-rot	-5.99823
Lhip z-rot	-2.32991
Navel -rot	-0.971196
Neck -rot	-25.9552
Lshoulder x-rot	125.279
Lshoulder y-rot	12.5213
Lshoulder z-rot	6.056
Leibow -rot	0.00223581
Rshoulder x-rot	87.9161
Rshoulder y-rot	17.5131
Rshoulder z-rot	13.1468
Relbow -rot	5.34377
Rhip x-rot	71.7007
Rhip y-rot	19.0387
Rhip z-rot	-2.74211
Rknee -rot	-0.0055452

修正後位置 Error：

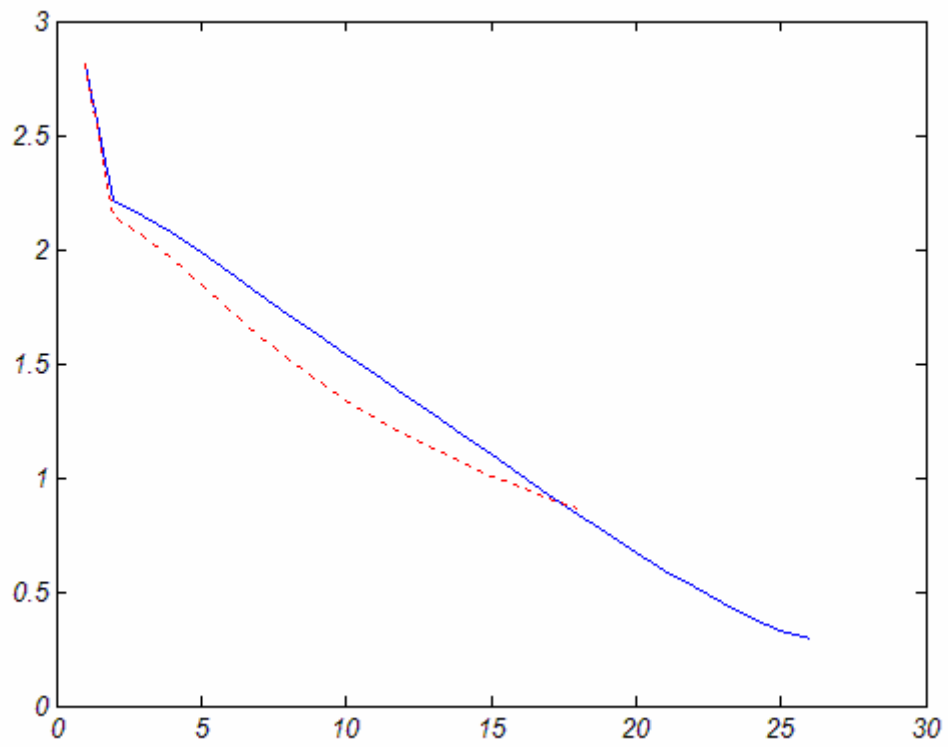
LeftWrist	Position Error = 0.654775
Head	Position Error = 1.02182
RightWrist	Position Error = 0.865641
RightHeel	Position Error = 0.155281
Total Position Error	= 2.69752



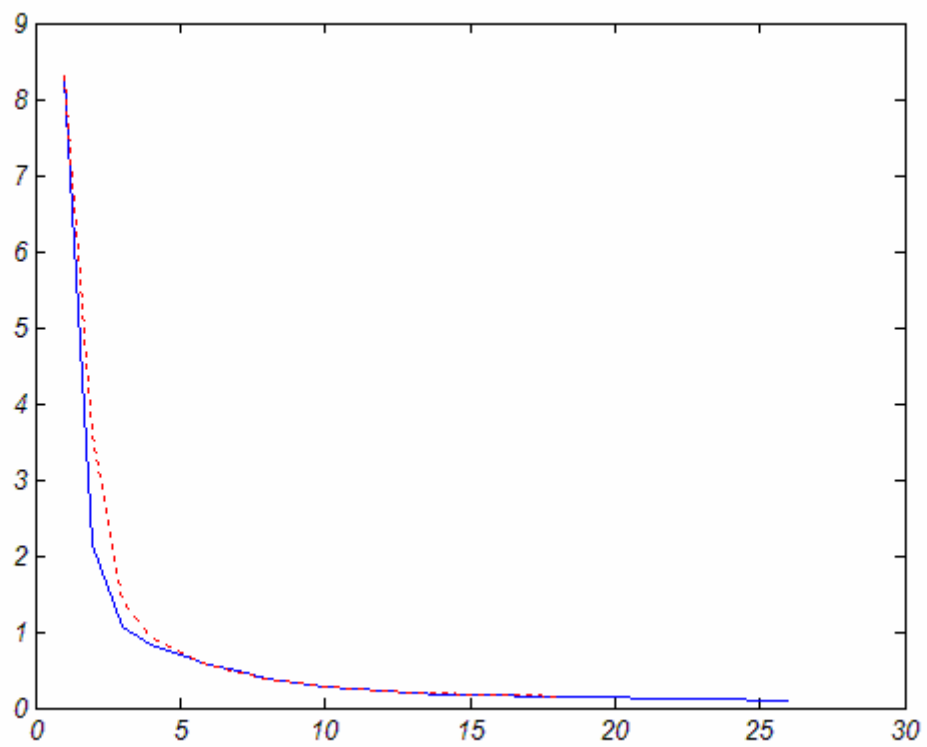
頭部位置誤差收斂比較圖  
 (藍色實線：SDLS Method；紅色虛線：DLS Method)



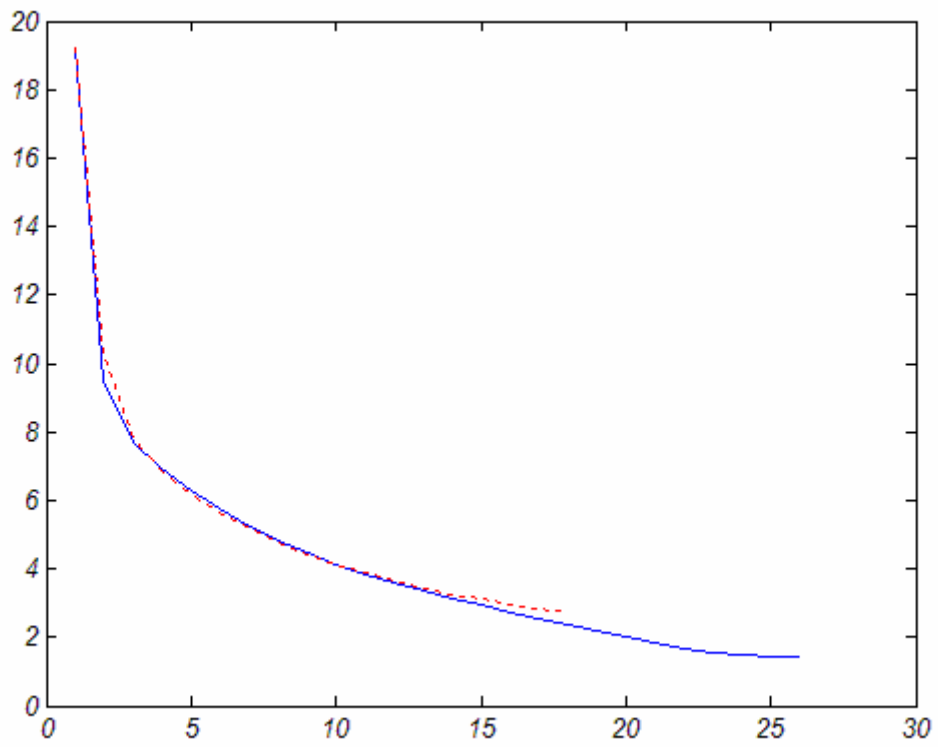
左手腕位置誤差收斂比較圖



右手腕位置誤差收斂比較圖  
 (藍色實線：SDLS Method；紅色虛線：DLS Method)



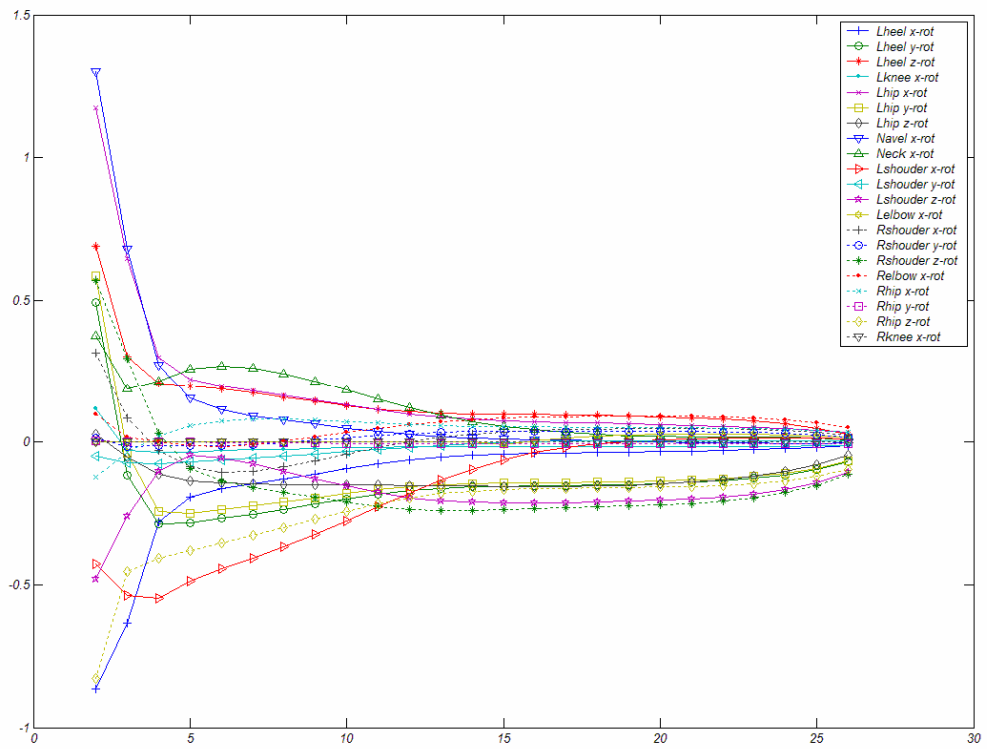
右腳跟位置誤差收斂比較圖



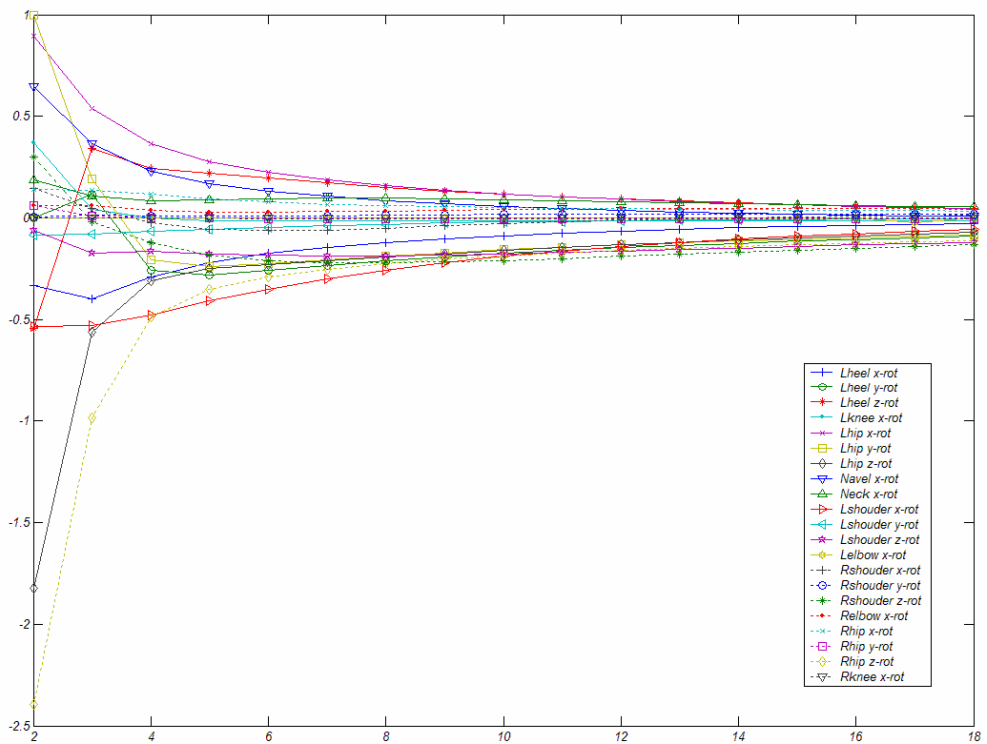
End-Effectors 總和位置誤差收斂比較圖  
(藍色實線：SDLS Method；紅色虛線：DLS Method)



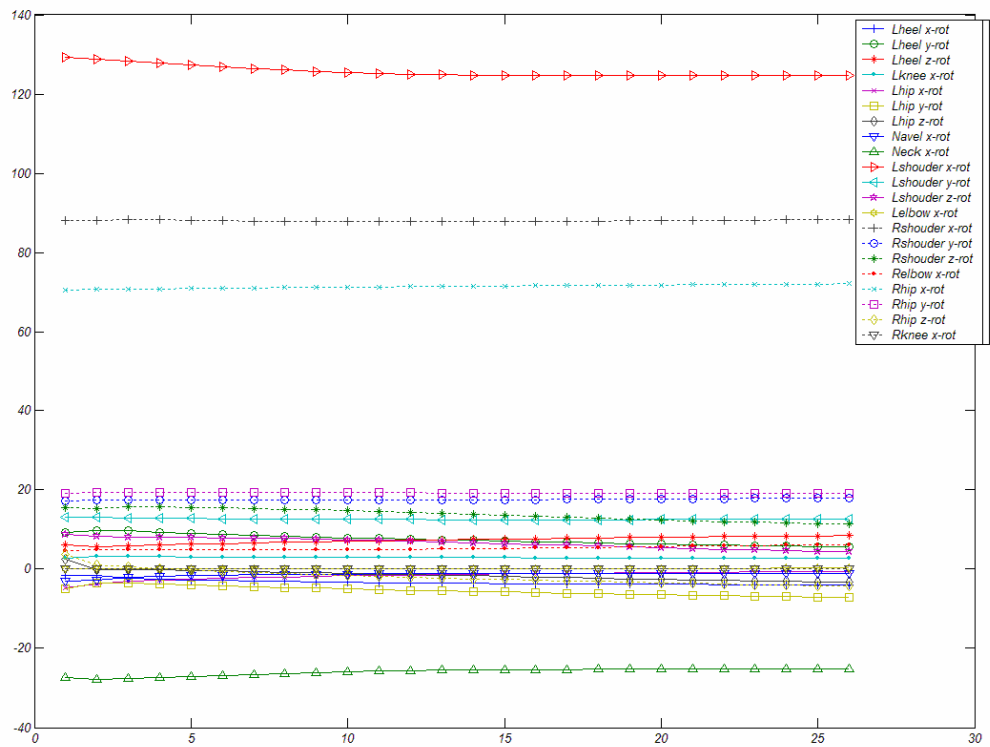
SDLS 的  $\Delta\theta$  收斂情形：



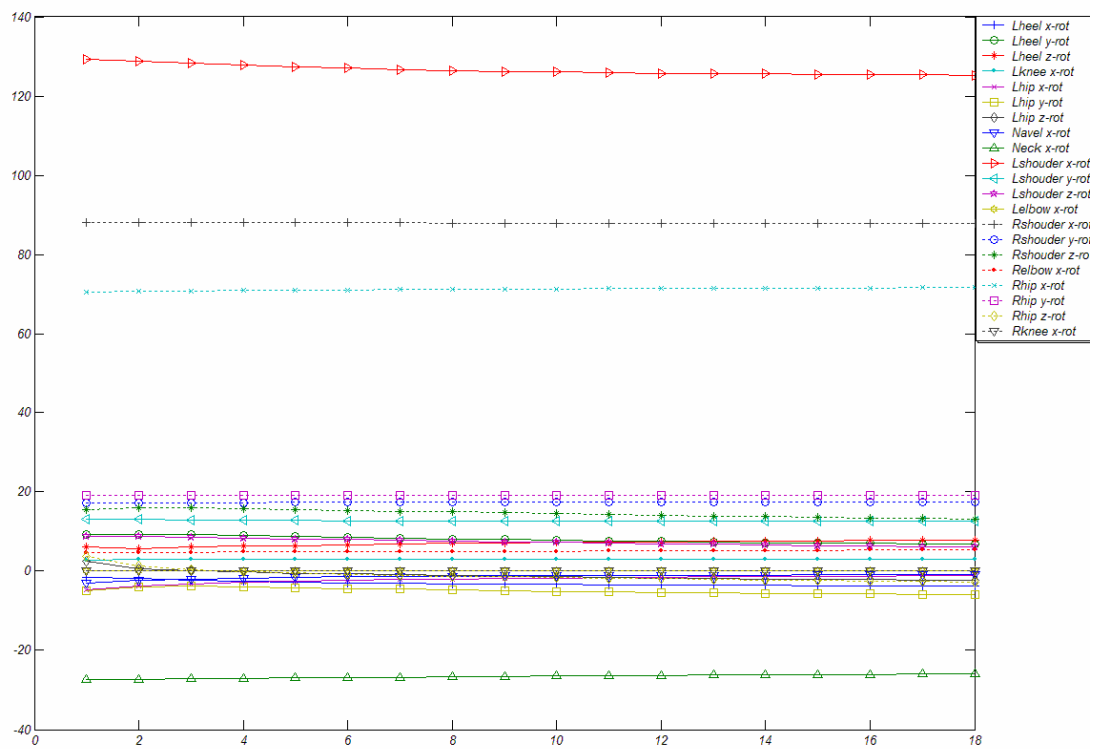
DLS 方法的  $\Delta\theta$  收斂情形：



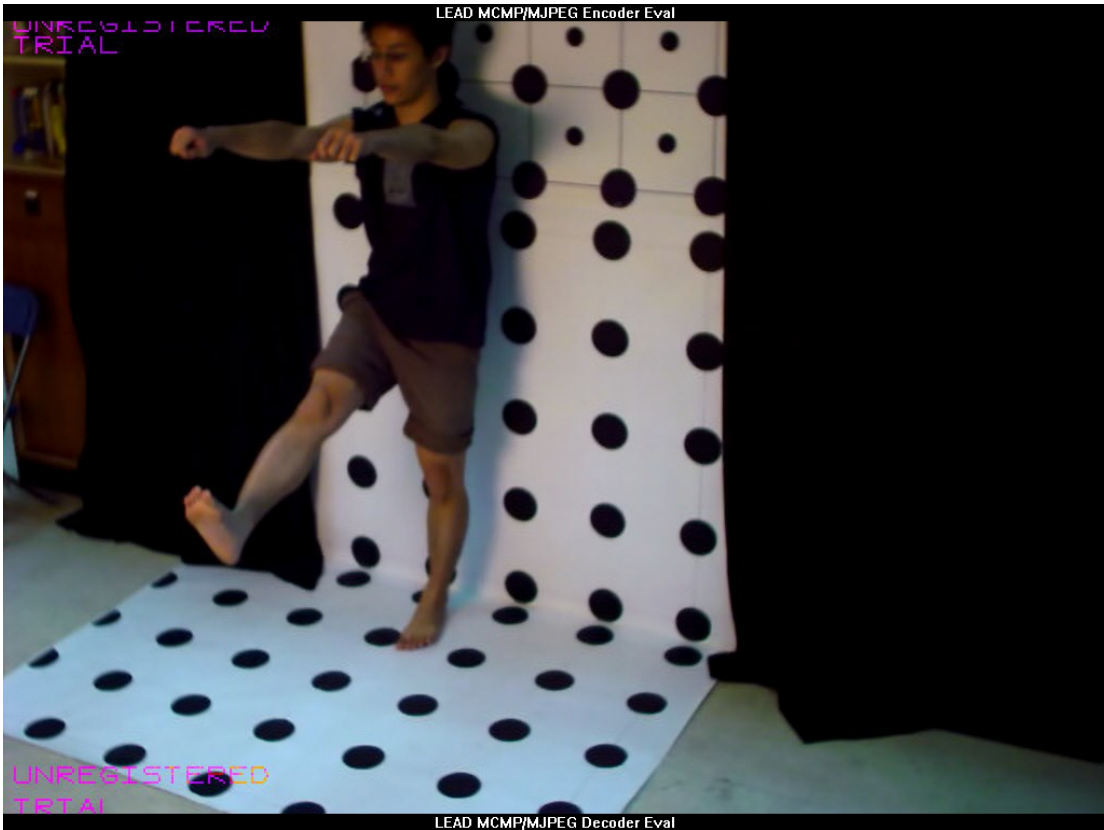
SDLS 方法的  $\theta$  收斂情形：



DLS 方法的  $\theta$  收斂情形：







View 1 Data Image



View 2 Data Image

實驗結論：

兩種方法在實驗的結果中，不論是在 **iteration** 次數上或是 **position error** 上都差不多，然而在速度上我們只考慮 **iteration** 的次數卻沒有考慮單次 **iteration** 所花的時間；然而 **SDLS** 的演算法比 **DLS** 明顯複雜的多，因此在下一節中我們將分析這兩個方法的演算法，並實際量測時間判斷 **DLS** 與 **SDLS** 的執行效率如何。



## 6.5. DLS 與 SDLS 的演算法複雜度分析

DLS Method 複雜度 與 SDLS Method 複雜度 的比較：

在一個 IK iteration 中，流程約分為三部份：

1. 以肢節目前的狀態計算出 Jacobian Matrix
2. 對 Jacobian Matrix 做 SVD
3. 求取  $\Delta\theta$

而 DLS 與 SDLS 唯一的不同只在於“求取  $\Delta\theta$ ”的方式

因此我們分析這兩個方法在求取  $\Delta\theta$  時的複雜度，以比較兩者之差異。

**DLS 求取  $\Delta\theta$  時步驟：**

1. 求  $\sum_{i=1}^r \frac{\sigma_i}{\sigma_i^2 + \lambda^2} v_i u_i^T$ 。

主要計算量為 r 個 dot-product，r 為 Diagonal Matrix 的非零項個數，最大可能值為 Joints 的總個數 J，因此複雜度為 O(J) 個 dot-product。

一個 dot-product 為 U 的 Row 數個乘法與加法；U 的 Row 數 = End-effectors 總個數 k，所以複雜度為 O(J\*k) 個乘法與加法。

2.  $\Delta\theta = \left( \sum_{i=1}^r \frac{\sigma_i}{\sigma_i^2 + \lambda^2} v_i u_i^T \right) \cdot \bar{e}$ 。

一個 dot-product。O(1)。

因此 DLS 的總複雜度為 O(J\*k) 個乘法與加法。

**SDLS 求取  $\Delta\theta$  時步驟：**

For All i (i = 1~k, k 為 End-effectors 總個數)

{

1. 求  $N_i = \sum_{j=1}^k \|u_{j,i}\|$ 。

主要計算量為 k 個取 Norm，k 為 end-effectors 總個數，因此複雜度為 O(k) 個取 Norm。取 Norm 為 U 的 Row 數個乘法與加法；U 的 Row 數 = End-effectors 總個數 k，所以複雜度為 O(k\*k) 個乘法與加法。

2. 求  $M_i = \sum_{\ell} M_{i,\ell}$ ， $M_{i,\ell} = \sigma_i^{-1} \sum_{j=1}^n |v_{j,i}| \rho_{\ell,j}$ 。

主要計算量為計算所有的  $\rho_{\ell,j} = \left\| \partial s_{\ell} / \partial \theta_j \right\|$ ，計算量為 Joints 總個數 J\*End-effectors 總

個數  $k$  個取 Norm，複雜度  $O(J*k)$ 個 3 個乘法和 2 個加法。

以及計算  $\sigma_i^{-1} \sum_{j=1}^n |v_{j,i}| \rho_{\ell,j}$ ， $n$  為  $V$  的 column 數=Joints 總個數  $J$ ，

複雜度為  $O(J)$ 個乘法與加法。

再計算  $M_i = \sum_{\ell} M_{i,\ell}$ ，將不同 all end-effectors 的  $M_{i,\ell}$  加起來，End-effectors 總個數= $k$ ，

計算量為  $O(k)$ 個加法。

$$3. \quad \gamma_i = \min(1, N_i / M_i) \cdot \gamma_{\max} \circ$$

計算量為一個除法和一個比較大小和一個乘法，複雜度  $O(1)$ 。

$$4. \quad \varphi_i = \text{ClampMaxAbs}(\sigma_i^{-1} \alpha_i v_i, \gamma_i) \circ$$

計算  $\varphi_i = \begin{cases} \sigma_i^{-1} \alpha_i v_i, & \text{if } |\sigma_i^{-1} \alpha_i v_i| \geq |\gamma_i| \\ \gamma_i, & \text{if } |\sigma_i^{-1} \alpha_i v_i| < |\gamma_i| \end{cases}$ ，計算量為一個取 1-Norm 和一個比較大小，複雜

度  $O(1)$ 。

在 For Loop 中的所有步驟計算複雜度為  $O(k*k+3*J*k+J)$ 個乘法和  $O(k*k+2*J*k+J+k)$ 個加法；化簡為  $O(k^2 + J*k)$ 個乘法和  $O(k^2 + J*k)$ 個加法。

}

Loop 的總複雜度為； $i$  次 For Loop 中的計算， $i$  為 End-effectors 總個數  $k$ 。

總複雜度為： $O(k^3 + J*k^2)$ 個乘法和  $O(k^3 + J*k^2)$ 個加法。

$$\Delta\theta = \text{ClampMaxAbs}(\sum_i \varphi_i, \gamma_{\max})$$

$$\Delta\theta = \begin{cases} \sum_i \varphi_i, & \text{if } |\sum_i \varphi_i| \geq |\gamma_{\max}| \\ \gamma_{\max}, & \text{if } |\sum_i \varphi_i| < |\gamma_{\max}| \end{cases}$$

計算量為  $i$  個加法和一個比較大小， $i$  為 End-effectors 總個數  $k$ ，複雜度  $O(k)$ 個加法。

因此 SDLS 的總複雜度為  $O(k^3 + J*k^2)$ 個乘法和  $O(k^3 + J*k^2)$ 個加法。

論文中  $k=12, J=16$ ，

DLS：192 個乘法與 192 個加法； $18.5 \mu s$  (參考論文提供數據)

SDLS：4032 個乘法與 4032 個加法； $120 \mu s$  (參考論文提供數據)

我的實驗中  $k=12, J=21$

DLS : 252 個乘法與 252 個加法

SDLS : 4752 個乘法與 4752 個加法

我的實驗量測：

	求 J&SVD	求取 $\Delta\theta$	One-iteration total
DLS method	70 $\mu s$	70~80 $\mu s$	140~150 $\mu s$
SDLS method	70 $\mu s$	1800~1900 $\mu s$	1850~2000 $\mu s$

演算法分析中 SDLS 比 DLS 求取  $\Delta\theta$  要多做約 18.85 倍的計算；

實際 runtime 中 SDLS 比 DLS 求取  $\Delta\theta$  要多費時約 22.5 倍；

顯示分析結果為一個可接受的範圍。

因此我們可知道 SDLS 每個單一 iteration 要比 DLS 耗時得多，因此如果用 DLS 與 SDLS 所需的 iteration 次數差不多時，其實 SDLS 反而耗時，此時可以選擇 DLS 來取代之。



## 6.6. 以不同的 $\lambda$ 值，觀察 $\Delta\theta$ 的收斂情形，以選擇比較好的 $\lambda$ 值

在 Damped Least Square Method 中，以  $\Delta\theta = (\sum_{i=1}^r \sigma_i^{-1} v_i u_i^T) \cdot \bar{e} = (\sum_{i=1}^r \frac{\sigma_i}{\sigma_i^2 + \lambda^2} v_i u_i^T) \cdot \bar{e}$  來計算  $\Delta\theta$  的值，而其中  $\lambda$  為一 damping constant，主要目的在防止  $\sigma_i$  過小的時候，造成  $\Delta\theta$  的值過度上升。但這個 damping constant  $\lambda$  的選定，由於沒辦法為  $\Delta\theta$  值定一個 threshold 說明多大才算是過大，且  $\Delta\theta$  的值也不只是被  $\lambda$  與  $\sigma_i$  影響而已；跟該時刻的  $v_i$ 、 $u_i$ 、 $\bar{e}$  都有關，而  $\lambda$  只是一個 constant，並不會動態地變化，因此並沒有一個標準可以決定多大的  $\lambda$  是絕對的 Optimal，但  $\lambda$  在計算  $\Delta\theta$  時，確實會對其結果有重要影響；因此我們實測分析，在  $\lambda$  設定不同值的時候，其  $\Delta\theta$  的收斂情形為何？

首先先知道  $\sigma_i$  的值域約為  $\sigma_i = 6 \sim 0.01$  之間(根據實際資料)。

但大部分的值是 6~1 之間。

因此若不 Damping(即  $\lambda = 0$ )

$\sigma_i$  的值小時(根據實際資料可能是 0.1~0.01)， $\sigma_i^{-1} = 10 \sim 100$ ；

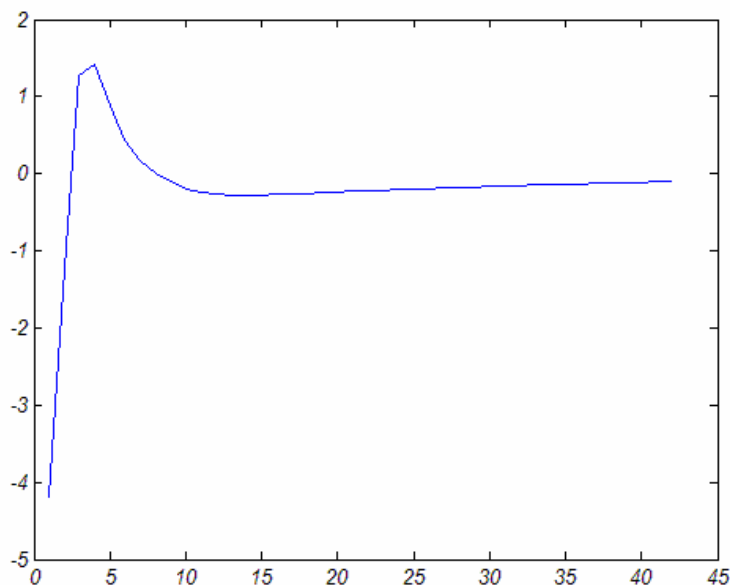
相較於大部分的  $\sigma_i$  值是 6~1 之間， $\sigma_i^{-1} = 1 \sim 0.166$ ；

$\sigma_i$  的值小時確實會使  $\sigma_i^{-1}$  上升很多。

確認有這個現象後，實際比較  $\lambda$  設定不同值的時候，其  $\Delta\theta$  的收斂情形：

$\lambda = 0.1$

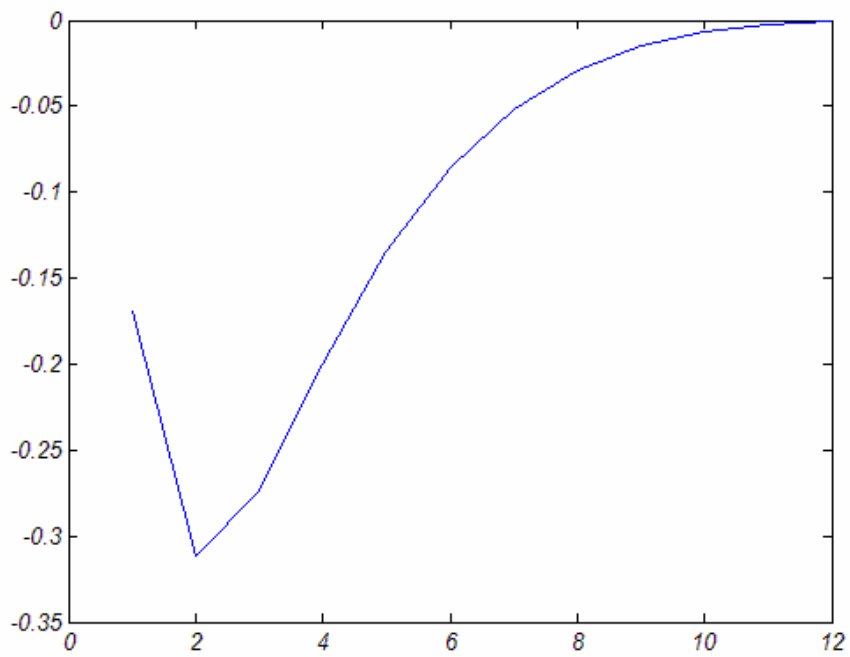
iteration 次數=42



Rshoulder-x-dof 各 iteration 的  $\Delta\theta$  圖

$\lambda=0.7$

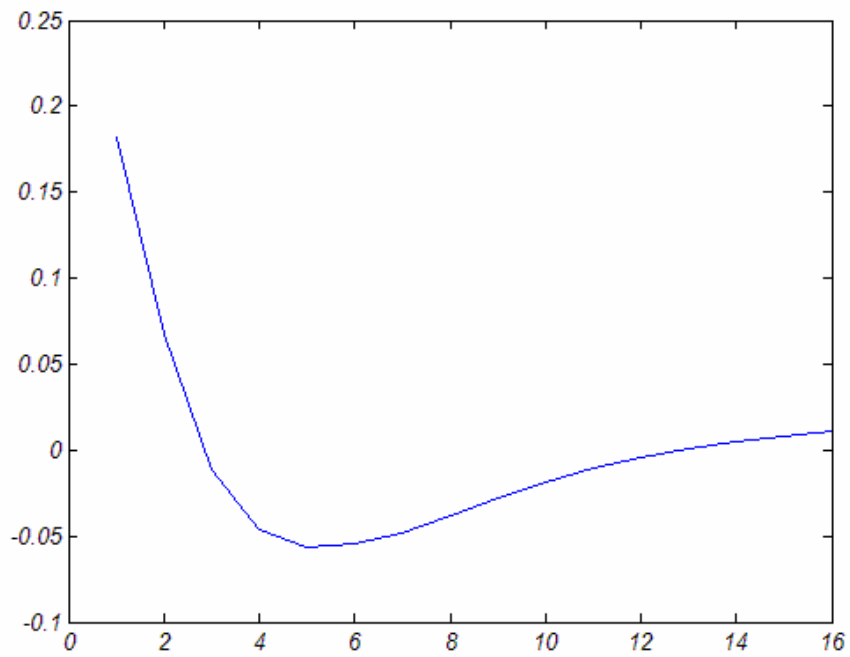
iteration 次數=12



Rshoulder-x-dof 各 iteration 的  $\Delta\theta$  圖

$\lambda=1.1$

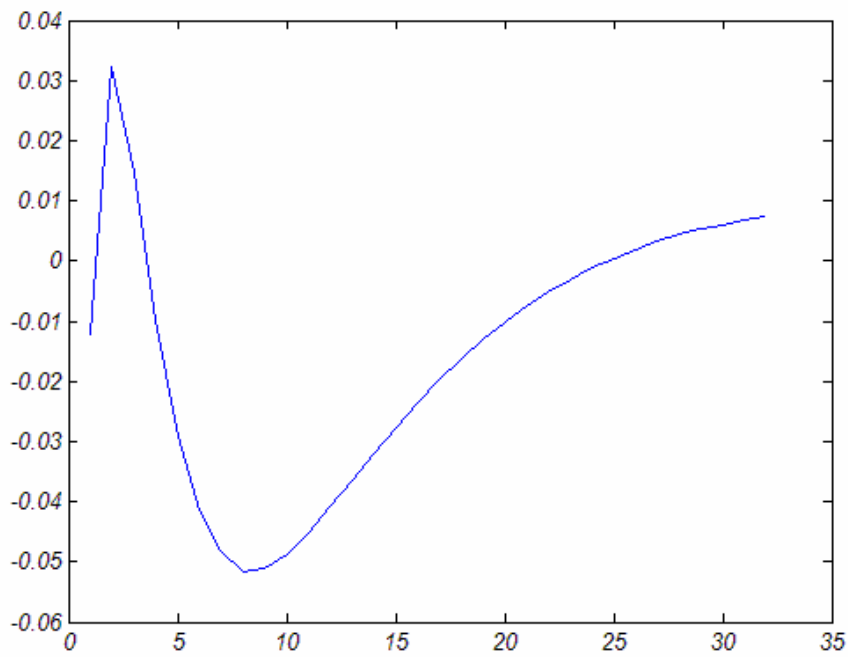
iteration 次數=16



Rshoulder-x-dof 各 iteration 的  $\Delta\theta$  圖

$\lambda=1.6$

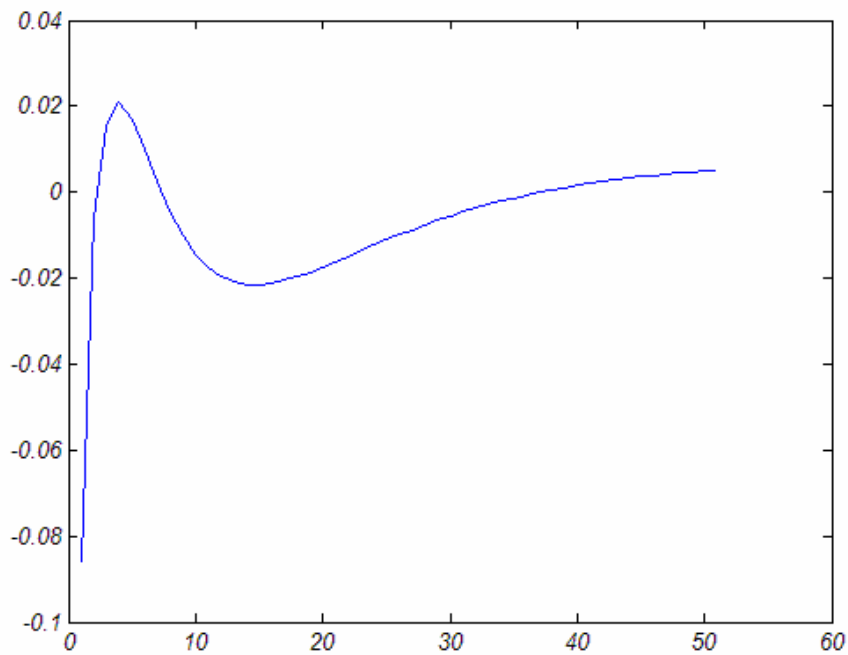
iteration 次數=32



Rshoulder-x-dof 各 iteration 的  $\Delta\theta$  圖

$\lambda=2.1$

iteration 次數=51

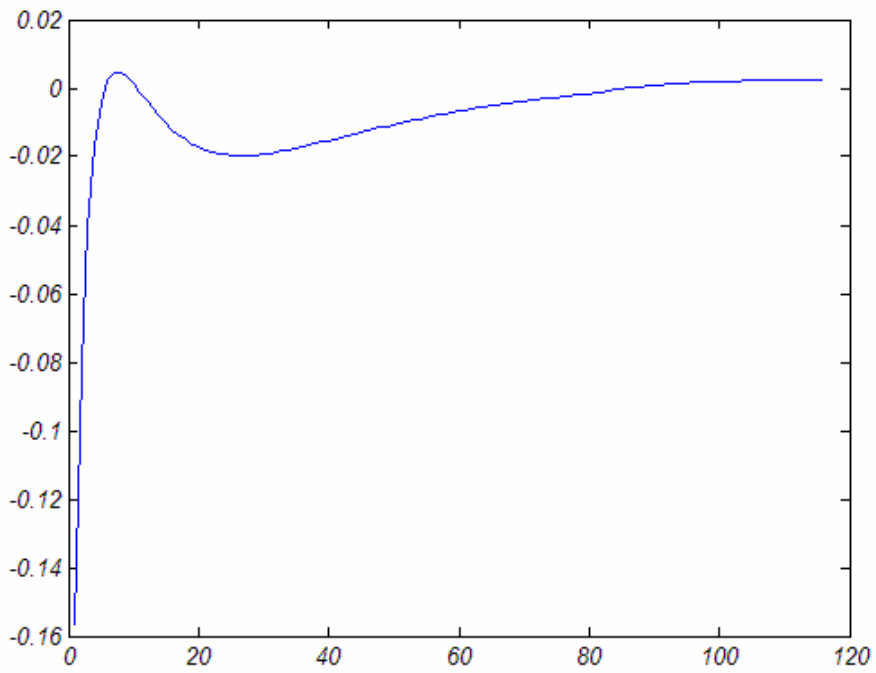


Rshoulder-x-dof 各 iteration 的  $\Delta\theta$  圖

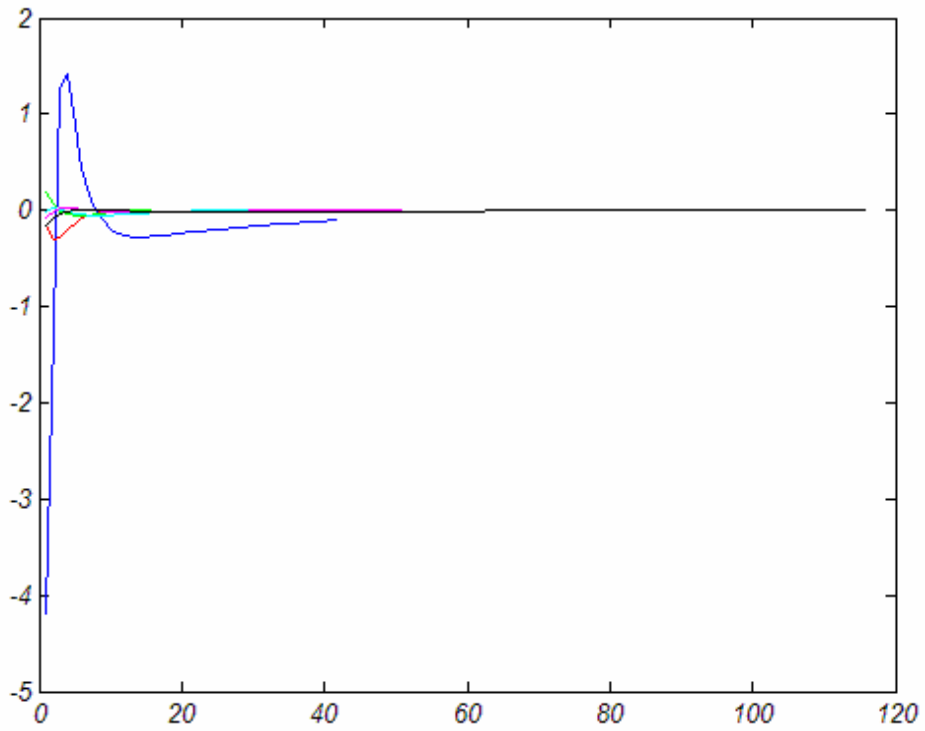


$\lambda=3.1$

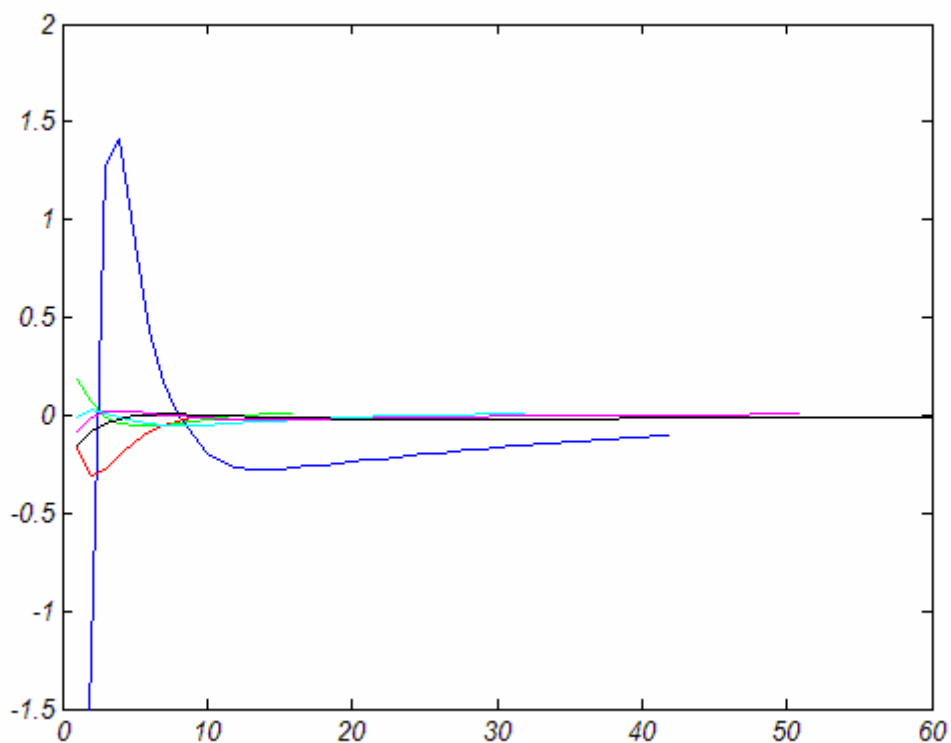
iteration 次數=116



Rshoulder-x-dof 各 iteration 的  $\Delta\theta$  圖



六種不同的  $\lambda$  各 iteration 的  $\Delta\theta$  的比較圖



六種不同的  $\lambda$  各 iteration 的  $\Delta\theta$  的比較放大圖  
 結論：



$\lambda$  過大或過小都會造成所需 iteration 次數的上升，

由圖中我們看出  $\lambda$  過小時會造成  $\Delta\theta$  過大，每次 Update 的  $\Delta\theta$  過大，造成後來要修正前面 Update 錯誤的  $\Delta\theta$  的誤差，造成收斂較久。

由圖中我們看出  $\lambda$  過大時會造成  $\Delta\theta$  過小，每次 Update 的  $\Delta\theta$  過小，造成一次就可以 Update 的  $\Delta\theta$  卻分成了很多次去 Update，造成收斂較久。

根據我實驗的資料顯示  $\lambda$  值在 0.7~1.1 所得的結果較為良好。

## 6.7.比較 IK 做每個 frame 與 key-frame 內插的結果

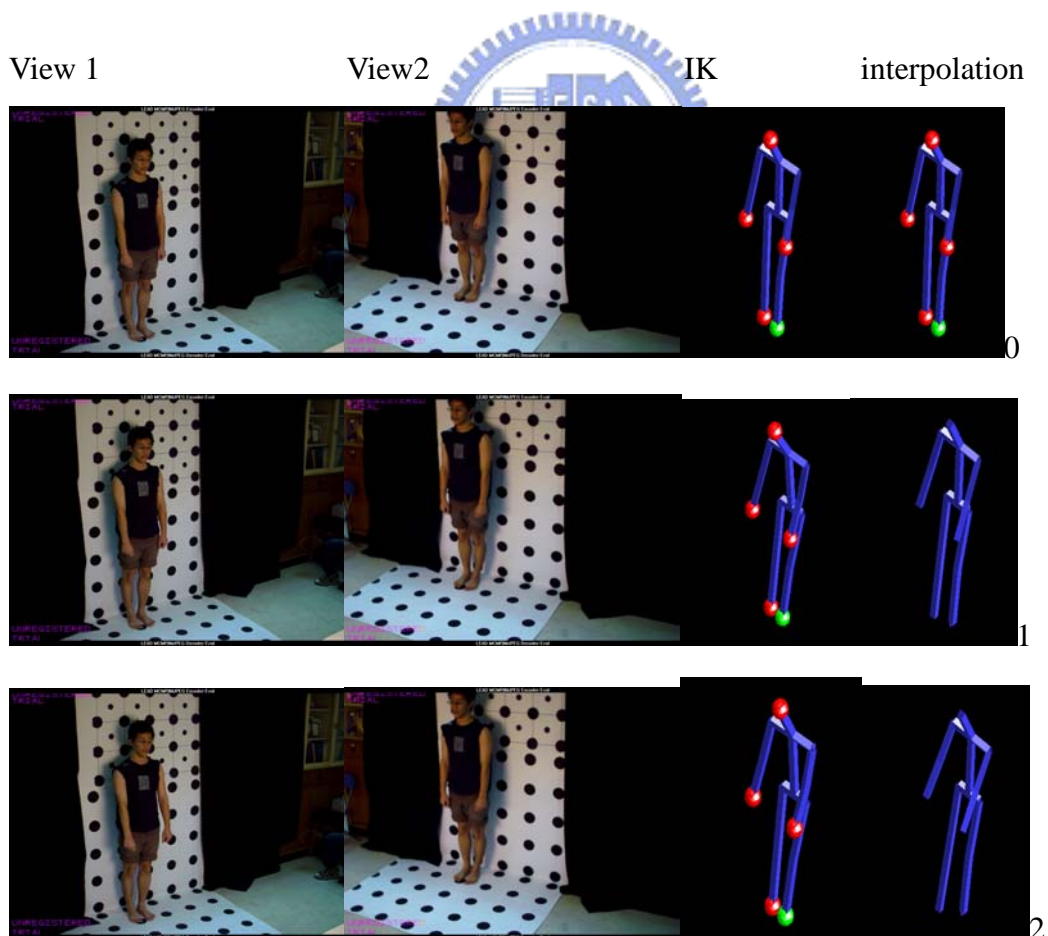
在一連串的动作經過 IK 修正後，可以透過一些演算方式判斷某些 frame 的动作是 key-frame。如下列的动作序列中，我們研判 frame#0 與 frame#12 是此动作序列中的 key-frame，因此我們將 frame#0 與 frame#12 中各個 DOF 的角度都進行線性的內插，等分成爲與原本的 frame 數相同的數量；使內插出來的 frame 能與原本 IK 修出來的 frame 一一對應。並觀察是否兩個序列會很相似，如此即可以僅記錄 key-frame 的部份爲代表，而不需記錄每一個 frame 的結果了。

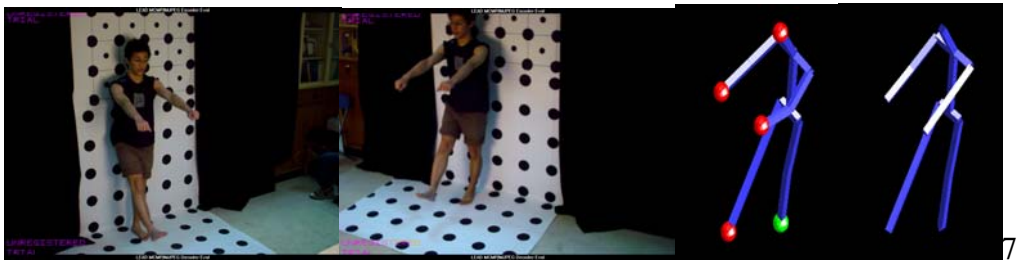
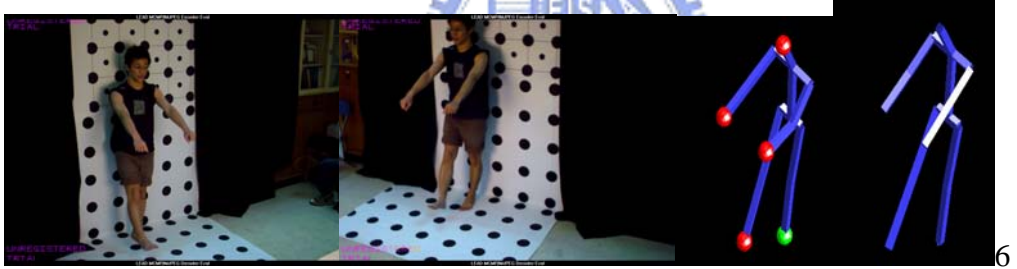
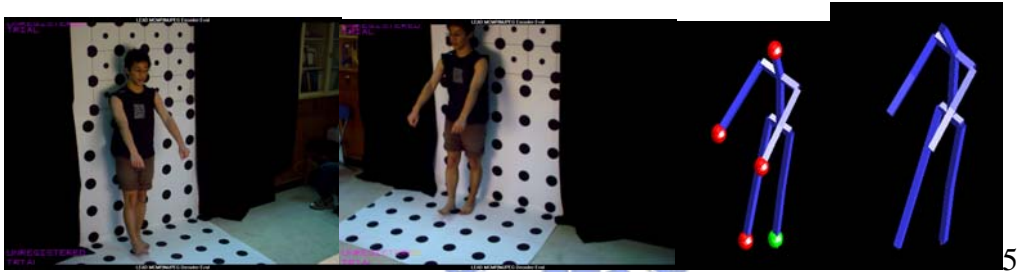
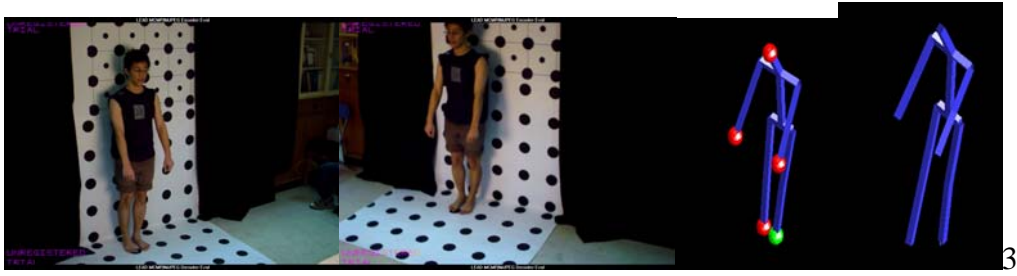
實驗結果如下：

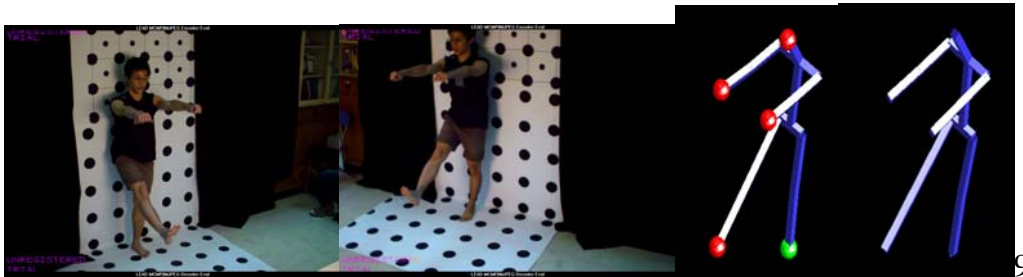
View1、View2 分別是同一時刻不同角度的影像

IK 是 IK 針對每一個 frame 的影像修正完所得到的結果

Interpolation 則是以 frame#0 與 frame#12 爲 key-frame 內插出來的結果



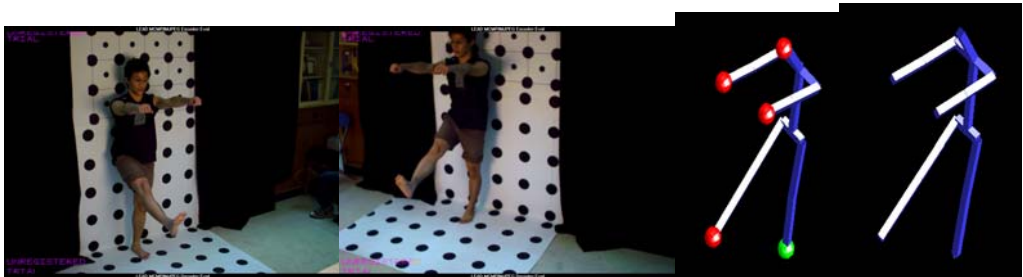




9



10



11



12

IK 結果

內插結果

data_index	No.0	data_index	No.0
Lheel x-rot	-0.628871	Lheel x-rot	-0.628871
Lheel y-rot	0.414916	Lheel y-rot	0.414916
Lheel z-rot	1.98725	Lheel z-rot	1.98725
Lknee -rot	8.76062	Lknee -rot	8.76062
Lhip x-rot	-9.99215	Lhip x-rot	-9.99215
Lhip y-rot	6.94059	Lhip y-rot	6.94059
Lhip z-rot	3.91477	Lhip z-rot	3.91477
Navel -rot	-9.83073	Navel -rot	-9.83073
Neck -rot	-22.2485	Neck -rot	-22.2485
Lshouder x-rot	42.931	Lshouder x-rot	42.931



Lshoulder y-rot 24.3061	Lshoulder y-rot 24.3061
Lshoulder z-rot -1.51413	Lshoulder z-rot -1.51413
Lelbow -rot 7.06775	Lelbow -rot 7.06775
Rshoulder x-rot 18.9633	Rshoulder x-rot 18.9633
Rshoulder y-rot 3.01439	Rshoulder y-rot 3.01439
Rshoulder z-rot 1.15472	Rshoulder z-rot 1.15472
Relbow -rot 1.96698	Relbow -rot 1.96698
Rhip x-rot 4.94616	Rhip x-rot 4.94616
Rhip y-rot -0.766155	Rhip y-rot -0.766155
Rhip z-rot -7.8671	Rhip z-rot -7.8671
Rknee -rot -0.560591	Rknee -rot -0.560591
data_index No.1	data_index No.1
Lheel x-rot 0.409102	Lheel x-rot -0.716848
Lheel y-rot -0.738018	Lheel y-rot 0.720669
Lheel z-rot 6.37664	Lheel z-rot 2.5083
Lknee -rot 7.70709	Lknee -rot 8.30977
Lhip x-rot -9.99215	Lhip x-rot -9.53411
Lhip y-rot 3.6852	Lhip y-rot 6.11024
Lhip z-rot -3.43405	Lhip z-rot 3.37087
Navel -rot -9.99898	Navel -rot -9.13997
Neck -rot -23.5323	Neck -rot -22.8944
Lshoulder x-rot 39.3187	Lshoulder x-rot 50.2478
Lshoulder y-rot 10.0836	Lshoulder y-rot 24.2355
Lshoulder z-rot 3.51785	Lshoulder z-rot -0.776488
Lelbow -rot 15.3832	Lelbow -rot 6.48223
Rshoulder x-rot 19.988	Rshoulder x-rot 24.7771
Rshoulder y-rot 13.7141	Rshoulder y-rot 4.37107
Rshoulder z-rot 6.08119	Rshoulder z-rot 2.34314
Relbow -rot 2.02559	Relbow -rot 2.33066
Rhip x-rot 5.60785	Rhip x-rot 10.64
Rhip y-rot -1.14143	Rhip y-rot 0.81055
Rhip z-rot -1.19435	Rhip z-rot -7.37621
Rknee -rot -0.0998361	Rknee -rot -0.514337
...	...

data_index	No.12	data_index	No.12
Lheel x-rot	-1.68459	Lheel x-rot	-1.68459
Lheel y-rot	4.08395	Lheel y-rot	4.08395
Lheel z-rot	8.23988	Lheel z-rot	8.23988
Lknee -rot	3.35047	Lknee -rot	3.35047
Lhip x-rot	-4.49571	Lhip x-rot	-4.49571
Lhip y-rot	-3.0236	Lhip y-rot	-3.0236
Lhip z-rot	-2.61197	Lhip z-rot	-2.61197
Navel -rot	-1.54165	Navel -rot	-1.54165
Neck -rot	-29.999	Neck -rot	-29.999
Lshouder x-rot	130.733	Lshouder x-rot	130.733
Lshouder y-rot	23.4589	Lshouder y-rot	23.4589
Lshouder z-rot	7.33757	Lshouder z-rot	7.33757
Lelbow -rot	0.0415155	Lelbow -rot	0.0415155
Rshouder x-rot	88.7284	Rshouder x-rot	88.7284
Rshouder y-rot	19.2945	Rshouder y-rot	19.2945
Rshouder z-rot	15.4158	Rshouder z-rot	15.4158
Relbow -rot	6.33115	Relbow -rot	6.33115
Rhip x-rot	73.2728	Rhip x-rot	73.2728
Rhip y-rot	18.1543	Rhip y-rot	18.1543
Rhip z-rot	-1.97639	Rhip z-rot	-1.97639
Rknee -rot	-0.0055452	Rknee -rot	-0.0055452

爲了比較內插的 local 角度與實際每個 frame 的 local 角度差異，計算各別 Frame 中，內插的 local 角度與實際用 IK 做每個 frame 的 local 角度平均一個 joint 的角度差異與其標準差。

令 Frame t 中第 i 個 joint 的內插角度與實際實際用 IK 做每個 frame 的 local 角度的差異爲  $\Delta\theta'_i$ 。

則 Frame t 中內插角度與實際實際用 IK 做每個 frame 的 local 角度的平均差異爲

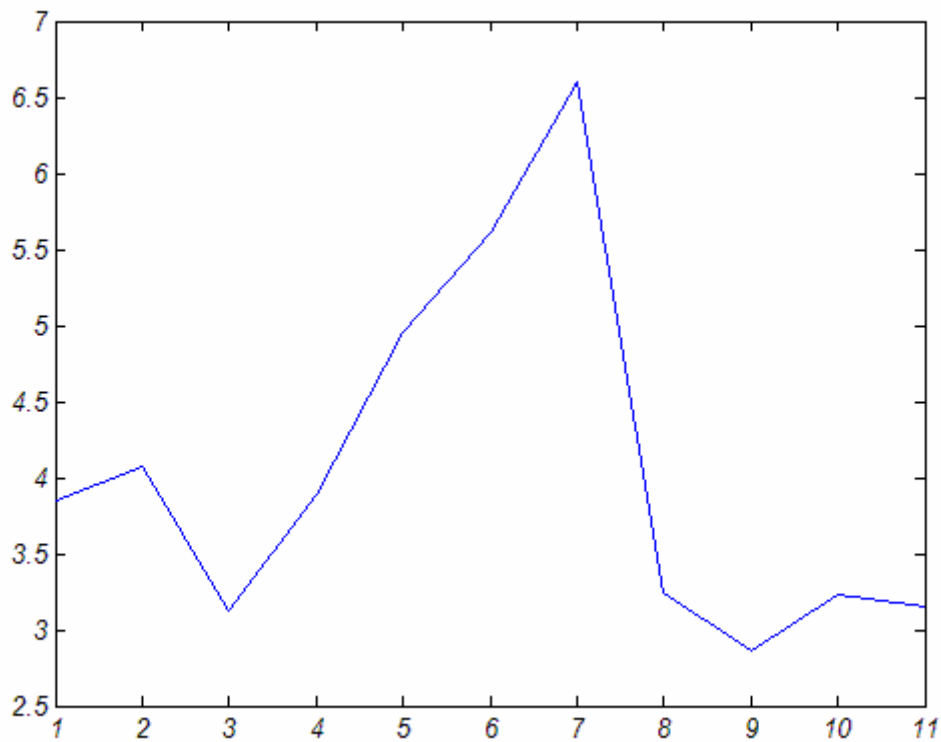
$$\overline{|\Delta\theta'|} = \frac{1}{n} \sum_{i=1}^n |\Delta\theta'_i|, \quad n \text{ 爲 joints 個數。}$$

Frame t 中內插角度與實際實際用 IK 做每個 frame 的 local 角度差異的標準差爲

$$\text{Standard deviation of } |\Delta\theta'| = \sqrt{\frac{1}{n} \sum_{i=1}^n (|\Delta\theta'_i| - \overline{|\Delta\theta'|})^2}$$

frame#1	$\overline{ \Delta\theta^t } : 4.2035^\circ$	Standard deviation of $ \Delta\theta^t  : 3.8447^\circ$
frame#2	$\overline{ \Delta\theta^t } : 3.8272^\circ$	Standard deviation of $ \Delta\theta^t  : 4.0658^\circ$
frame#3	$\overline{ \Delta\theta^t } : 3.7009^\circ$	Standard deviation of $ \Delta\theta^t  : 3.1201^\circ$
frame#4	$\overline{ \Delta\theta^t } : 3.7620^\circ$	Standard deviation of $ \Delta\theta^t  : 3.8868^\circ$
frame#5	$\overline{ \Delta\theta^t } : 4.8400^\circ$	Standard deviation of $ \Delta\theta^t  : 4.9597^\circ$
frame#6	$\overline{ \Delta\theta^t } : 5.4187^\circ$	Standard deviation of $ \Delta\theta^t  : 5.6126^\circ$
frame#7	$\overline{ \Delta\theta^t } : 5.7384^\circ$	Standard deviation of $ \Delta\theta^t  : 6.5933^\circ$
frame#8	$\overline{ \Delta\theta^t } : 3.7321^\circ$	Standard deviation of $ \Delta\theta^t  : 3.2390^\circ$
frame#9	$\overline{ \Delta\theta^t } : 3.6835^\circ$	Standard deviation of $ \Delta\theta^t  : 2.8643^\circ$
frame#10	$\overline{ \Delta\theta^t } : 3.0816^\circ$	Standard deviation of $ \Delta\theta^t  : 3.2287^\circ$
frame#11	$\overline{ \Delta\theta^t } : 3.2236^\circ$	Standard deviation of $ \Delta\theta^t  : 3.1543^\circ$





各 frame t 的  $|\Delta\theta'|$  圖

並統計所有 frame 平均的  $\overline{|\Delta\theta'|}$  與其標準差。

$$\overline{|\Delta\theta'|} = \frac{1}{T} \sum_{t=1}^T |\Delta\theta'|, \quad T \text{ 為總 frame 數。}$$

$$\text{Standard deviation of } \overline{|\Delta\theta'|} = \sqrt{\frac{1}{T} \sum_{t=1}^T (|\Delta\theta'| - \overline{|\Delta\theta'|})^2}$$

求出在此組實驗數據中，

所有的 frame 平均的一個 joint 的角度差異  $\overline{|\Delta\theta'|}$  : 4.1101°

Standard deviation of  $\overline{|\Delta\theta'|}$  : 0.8226°

結論：

實驗結果果然如預期，內插出來的 frame 確實極為相似，不僅在圖形上極為相近，在 local 角度上的差異也極小；各 DOF 的平均角度差異僅約 4°。因此我們確實可以僅記錄 key-frame 的 local 角度資訊來代替所有的 frame 的 local 角度資訊。

## 7、 結論和未來工作

### 7.1. 結論

本論文使用兩部攝影機所拍攝到的立體影像序列，重建了人體關節點粗略的 3D 位置，並相信人體的 End-Effectors 是在影像處理方法中能夠取得較準確的點，其三維位置座標能夠重建的比較精確，相對地其他的內部關節點較不準，因此選擇以 End-Effectors 中的關節點做為 Root，並且因為人體至少會以一腳著地，因而使用腳跟做為人體模型的 Root。

並因為相信人體的 End-Effectors 的三維座標位置是較為精確的，故強調使用逆向動力學修正內部錯誤的關節角度。與原拍攝圖片對照後，確實能夠將表演者的動作重現於程式中的 3D 人體模型。使得本論文在內部的關節點座標品質較差的情況下，能夠成功地使用立體視覺人體動作原始資料，進行修正的程序。

論文實驗中更比較了是否使用 IP 資訊引導逆向動力學之間的差異，不同的逆向動力學演算法之間的不同，演算法複雜度的分析，以及以 key-frame 代替完整影像序列的實驗結果。

實驗的最後結論是，以 IP 資訊引導，並使用 Damped Least Square 逆向動力學，是在效率上與正確性上較好的選擇。

### 7.2. 未來工作

本論文純粹是考慮在立體影像序列中人體關節點的自動截取已經完成的情況下進行修正，因此影像中人體關節點截取的好壞會影響到以 IP 為初解時，這個初解的好壞，如此也會連帶影響以逆向動力學修正後的正確性，因此改進立體影像序列中人體關節點的自動截取將是使本論文的應用更提升的一個重要工作。

若以後想以肚臍為 Root 並再增加腳尖(toe)以及肚臍與頸部的轉動自由度，則可依據圖 7-1 擴充的左手定則人體肢節樹狀模型圖去實施。圖中若肚臍對 x 軸旋轉時，身體則可左右擺動；若肚臍對 y 軸旋轉時，肩膀則可上下擺動；若肚臍對 z 軸旋轉時，身體則可上下擺動。

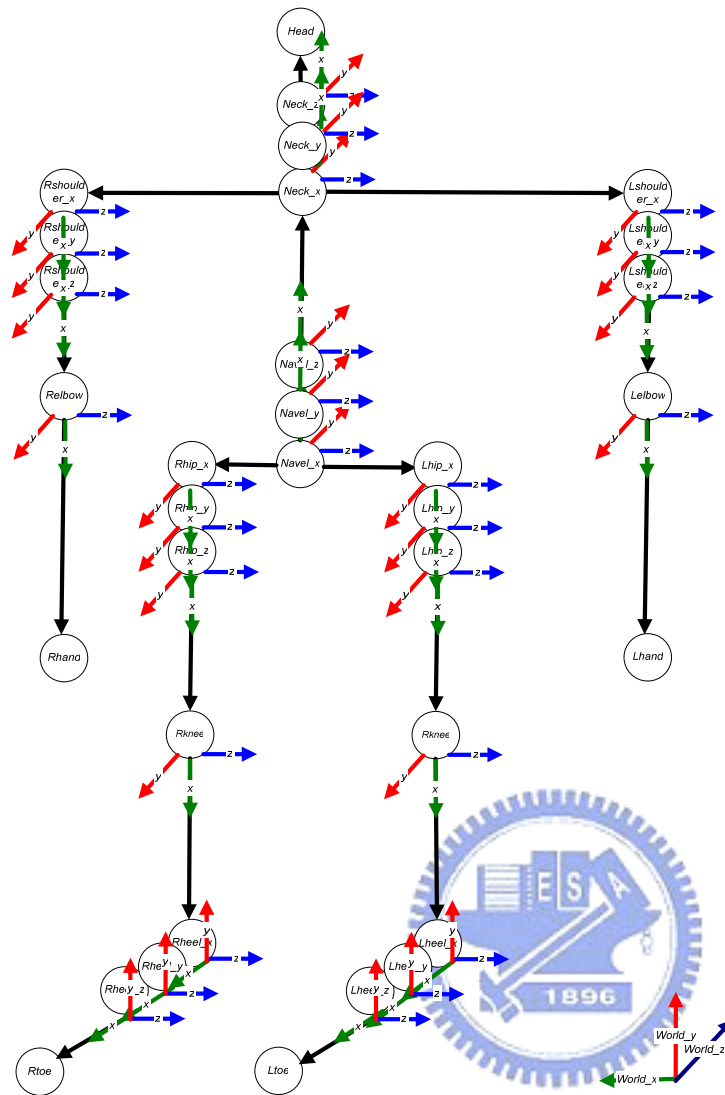


圖 7-1 擴充的左手定則人體肢節樹狀模型圖

在關節的角度限制上，我們僅以各轉軸的轉動上下限來約束人體模型的運動。若能夠再加入自我碰撞的偵測，就能夠使這個人體模型的 **Constraint** 更加地完整。

本論文的人體模型是一個樹狀架構的表示，內部的關節點個數與 **End-effectors** 個數都沒有限制，因此這個模型只要經過適當地設定其關節點與 **End-effectors** 就可以套用到任何的動物上，如馬、狗等等，若再針對不同的動物設定正確的 **Constraint**，就可以將本論文應用到任何的動物上了。

由於人體各肢節的長度因人而異，本論文採取人為量測被拍攝者身體肢節長度的方式，若能夠自動化這個項目，本論文的系統就能更加自動化。

## 參考文獻

1. Grigore Burdea, and Philippe Coiffect, "Virtual Reality Technology", John Wiley & Sons, New Yourk N.Y. 1994
2. Joe Gradecki, "The Virtual Reality Construction Kit", John Wiley & Sons, New York N.Y. 1994
3. Frederick H. Raab, Ernest B. Blood, Terry O. Steiner, and Herbert R. Jones, "Magnetic Position and Orientation Tacking System", IEEE Transcations on Aerospace and Electronic System, vol. AeS-15, no.5, pp.709-717. September 1979
4. Jih-fang Wang, Vernon Chi and Henry Fuchs, "A Real-time Optical 3D Tracker For Head-mounted Display System", Proc. Of IEEE Virtual Reality Annual International Symposium, Seattle, W.A., September 1993.
5. Jun Rekimoto, "A Vision-Based Head Tracker for Fish Tank Virtual Reality-VR without Head Gear", Proc. of 1995 IEEE Virtual Reality Annual International Symposium, Research Triangle Park, N.C., pp.94-100, March 1995.
6. 周宏隆, "虛擬實境中以真人帶領木偶做肢體動作之電腦視覺技術", 國立交通大學, 碩士論文, 民國 85 年
7. Samuel R. Buss and Jin-Su Kim, "Selectively Damped Least Squares for Inverse Kinematics", ACM Journal of Graphics Tools, 2004.
8. 吳政峰, "人體肢節三維動作捕捉之電腦視覺技術", 國立交通大學, 碩士論文, 民國 93 年
9. Naoto Date, Hiromasa Yoshimoto, Daisaku Arita and Rin-ichiro Taniguchi, "Real-time Human Motion Sensing based on Vision-based Inverse Kinematics for Interactive Applications", IEEE ICPR '04, Jan 2004.
10. 謝寧, "融入與互動式虛擬棒球遊戲中真人揮棒打擊", 國立交通大學, 碩士論文, 民國 86 年
11. Q. Cai, A. Mitiche, and J. Aggarwal, "Tracking human motion in an indoor environment", Proc. of International Conference on Image Processing, vol A, 1995, pp.215-218.
12. J. Aloimonos, "Perspective approximation", Image and Vision Computing, vol. 8, no.3, pp.179-192, 1990
13. Li-Chun Tommy Wang and Chih Cheng Chen, "A Combined Optimization Methods for Solving the Inverse Kinematics Problem of Mechanical Manipulators", IEEE Transactions on Robotics and Automation, Vol. 7, NO. 4, August 1991
14. A.Sezgin Abalı, "ANIMATION OF HUMAN OTION WITH INVERSE KINEMATICS USING NONLINEAR PROGRAMMING", b\_ilkent university, master thesis, September, 2001
15. Chris Welman, "INVERSE KINEMATICS AND GEOMETRIC CONSTRAINTS FOR ARTICULATED FIGURE MANIPULATION", Simon Fraser University, master thesis,

September 1993

16. JIANMIN ZHAO and NORMAN I. BADLER, "Inverse Kinematics Positioning Using Nonlinear Programming for Highly Articulated Figures", University of Pennsylvania, ACM Transactions on Graphics, Vol. 13, No.4, October 1994, Pages 313-336
17. Naganand Madhavapeddy and Stuart Ferguson, "Specialized Constraints for an Inverse Kinematics Animation System Applied to Articulated Figures", The Queen's University of Belfast
18. E. Durucan and T. Ebrahimi, "Change Detection and Background Extraction by Linear Algebra," Proceedings of the IEEE, Vol.89, No.10, pp.1368-1381, Oct. 2001.
19. T. Aach, et al., "Statistical model-based change detection in moving video," Signal Processing, Vol.31, pp.165-180, 1993.
20. Elgammal, A., Harwood, D. and Davis L. "Non-parametric Model for Background Subtraction", Proc IEEE FRAME-RATE Workshop,
21. Rafael C. Gonzalez & Richard E. Woods "Digital Image Processing"
22. 張家銘, "三維動畫中動作重置之研究", 國立清華大學, 碩士論文, 民國 90 年
23. 黃聿文, "三維空間多攝影機視覺監控系統", 國立交通大學, 碩士論文, 民國 91 年
24. 謝耀賢, "環場全景製作技術及其在虛擬實境應用", 國立交通大學, 碩士論文, 民國 86 年
25. 張振宇, "利用人臉視覺資訊作為虛擬實境畫面瀏覽之控制", 國立交通大學, 碩士論文, 民國 88 年
26. 陳郁中, "藉由平面投射轉換執行影像嵌合及物體重建", 國立交通大學, 碩士文, 民國 89 年