

第四章、以靜力學討論三軸機器手臂之路徑規劃問題

本章我們將討論具三個自由度的三軸舉重機器人，並僅先考慮靜力問題。在 4-2 節中，我們根據 Dijkstra 演算法來作三軸舉重機器人的路徑規劃。在 4-3 節中，我們將修正 Dijkstra 演算法，來節省搜尋時間並提高搜尋結果的精密度。最後在 4-4 節中，我們利用 B-spline Curve 的理論，以求得參數化且較平順的結果路徑。

4.1 網格的劃分與起點終點限制的修正

在前一章中，雖然我們僅討論兩軸舉重機器手臂的靜力問題，不過我們透過觀察角度與力矩關係曲面，得到目標函數與次方係數設定的方法與結果。在本章中，我們將根據上一章所使用的方法和得到結果，來作三軸舉重機器手臂的路徑規劃。我們同樣使用上一章所討論的三軸舉重機器手臂，相關資料如圖 3.1 以及圖 3.2 所示。在本章中，我們仍然僅討論靜力問題，因此只需要用到靜力方程式，則三軸力矩的計算分別如式(3-1)、(3-2)、(3-3)所示。

若節點在每一維座標的移動方向有三種：前進、後退、停留，則對於 n 維座標系統，節點共有 3^n 種移動方式。不過要扣掉每一維都停留的情況(即為原節點本身)，所以每一個節點應有 $3^n - 1$ 的子節點。所以對於二維座標系

統，每一個節點應有 8 個子節點。而對於三維座標系統，每一個節點應有 26 個子節點。

在第二章中所介紹的 Dijkstra 演算法，起點與終點都必須位於已劃分好的格點上，這對網格的劃分十分不方便；尤其是搜尋的維數愈多，對網格的劃分就愈困難。所以我們對此缺點做出修正，一開始網格的劃分僅考慮網格的大小，而起點與終點必然將位於某兩個網格內；對於三維座標系統，每個網格應該是由八個節點組成。原本 Dijkstra 演算法一開始只將起點加入 open 表中，現在我們修正這個步驟，改為先計算構成起點所在的網格之八個節點的目標函數值，並將之通通加入 open 表中，以這八個節點開始 Dijkstra 演算法的路徑搜尋，而這八個節點的父節點即均為起點。類似的做法，原本 Dijkstra 演算法的終止條件是搜尋到終點為止，現在我們改為搜尋到構成終點所在的網格之八個節點之一即終止搜尋，而其子節點則為終點。如此起點和終點的設定則可以不受限制。修正後的程式流程圖如圖 4.1 所示。

4.2 目標函數的設計與搜尋的結果

在上一章我們對兩軸機器手臂設計了目標函數，在本章中我們將沿用相同的目標函數來做路徑規劃。其方程式如(4-1)式：

$$\text{Minimise } f = \sum_{i=1}^n \left(|\tau_1|^\omega + |\tau_2|^\omega + |\tau_3|^\omega \right) \quad (4-1)$$

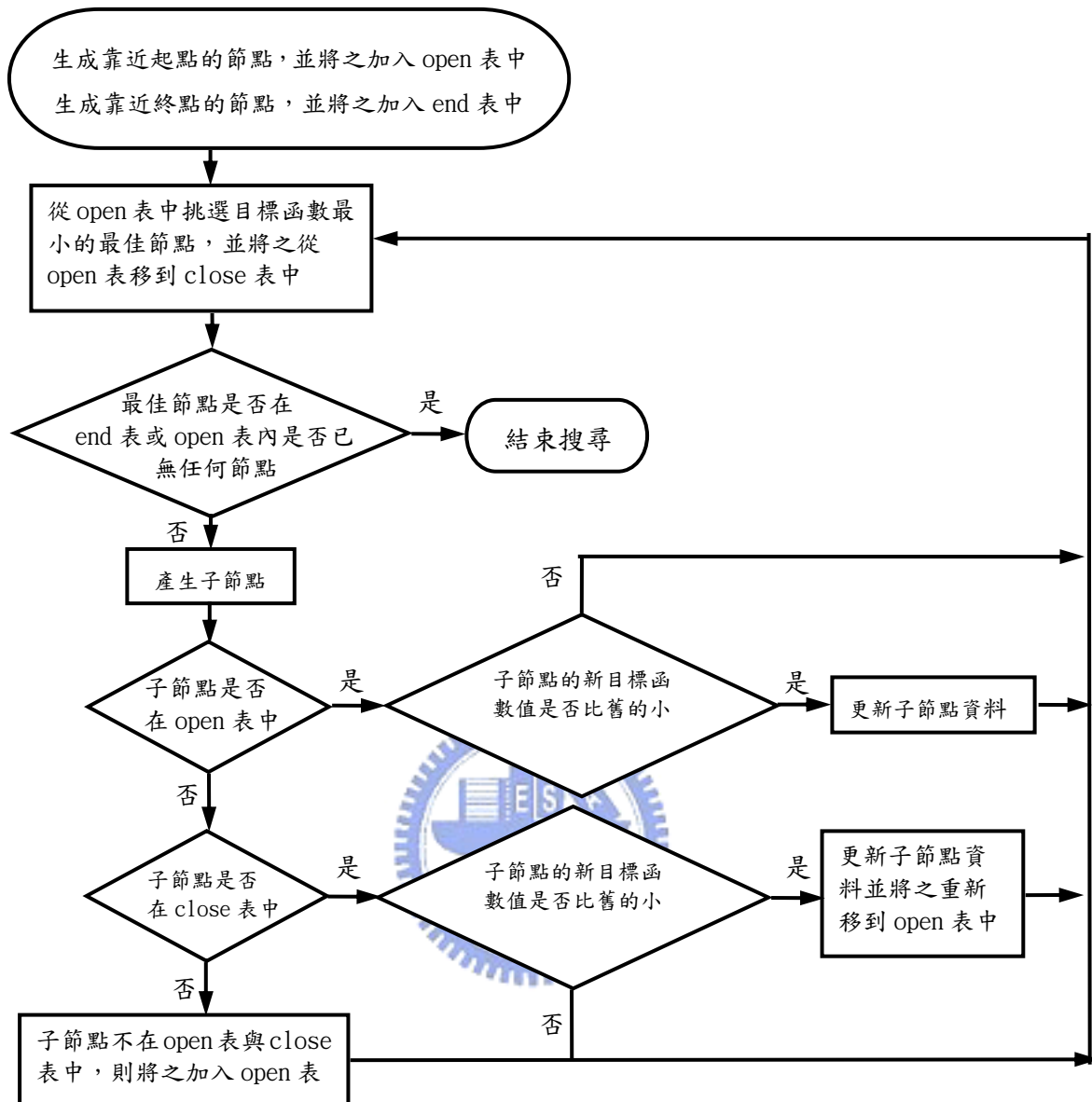


圖 4.1 修正起點與終點限制的三維 Dijkstra 演算法流程圖

此外，我們增加限制條件以符合實際情況，並且可以減少搜尋範圍。限制條件如下兩項：(1) θ_1 、 θ_2 與 θ_3 的範圍必須在正負 150 度之間 (2) 每一軸所能承受的力矩大小在正負 15 Nm 之間。首先我們先將網格以每 8 度劃分，並設定次方係數 ω 為 1。接著我們可以開始利用程式搜尋最佳路徑，其結果如圖 4.2 與圖 4.4 所示：

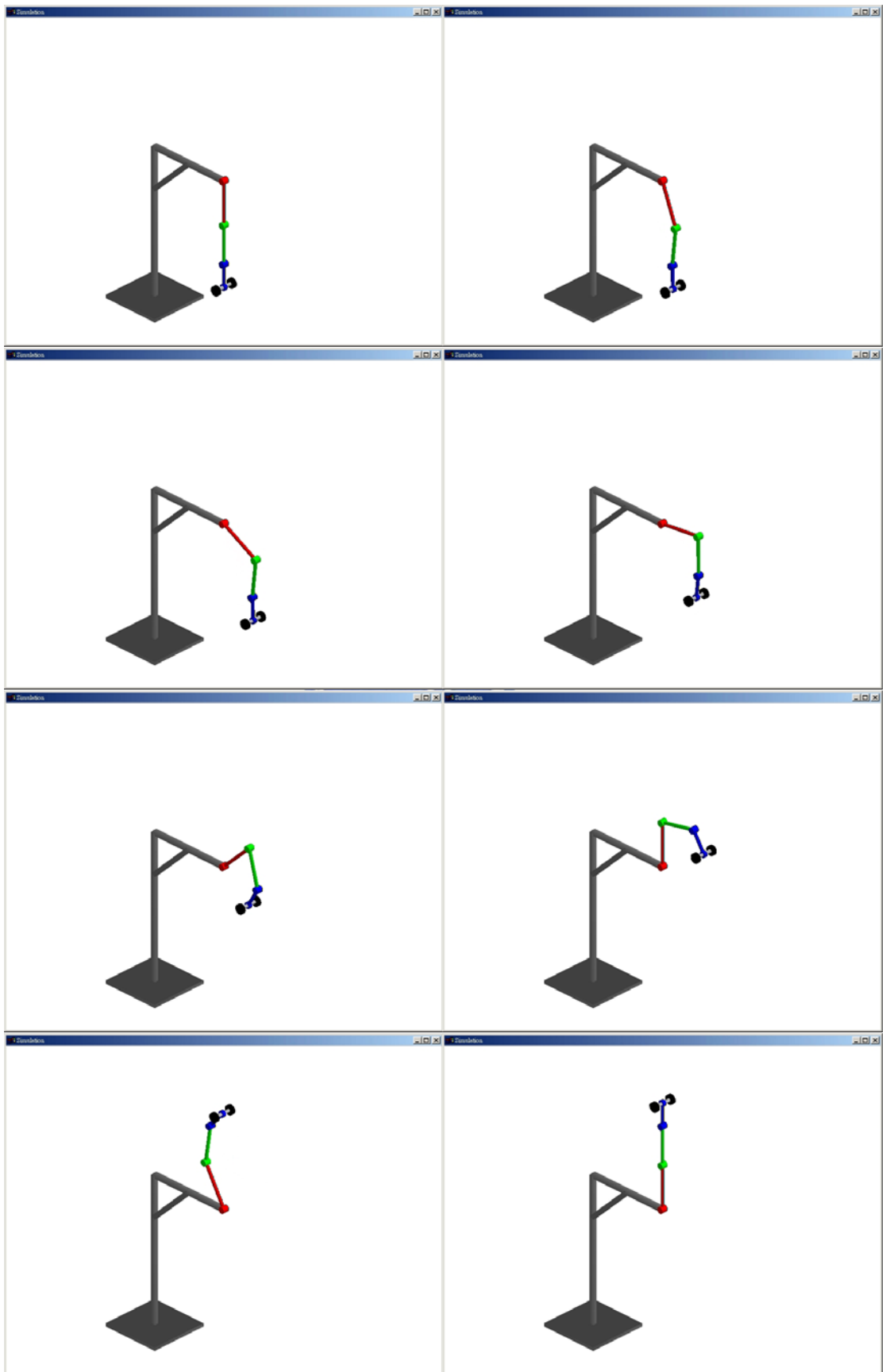


圖 4.2 動態模擬結果(次方係數為 1)

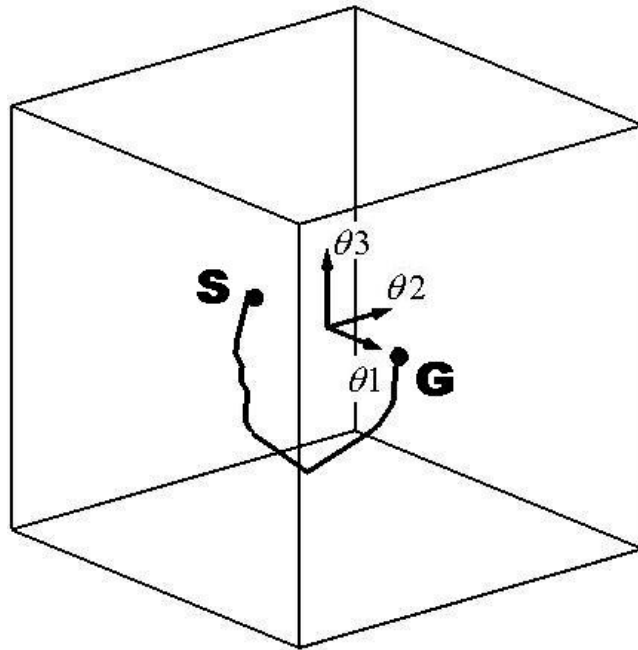


圖 4.3 模擬結果之行進路徑(次方係數為 1)

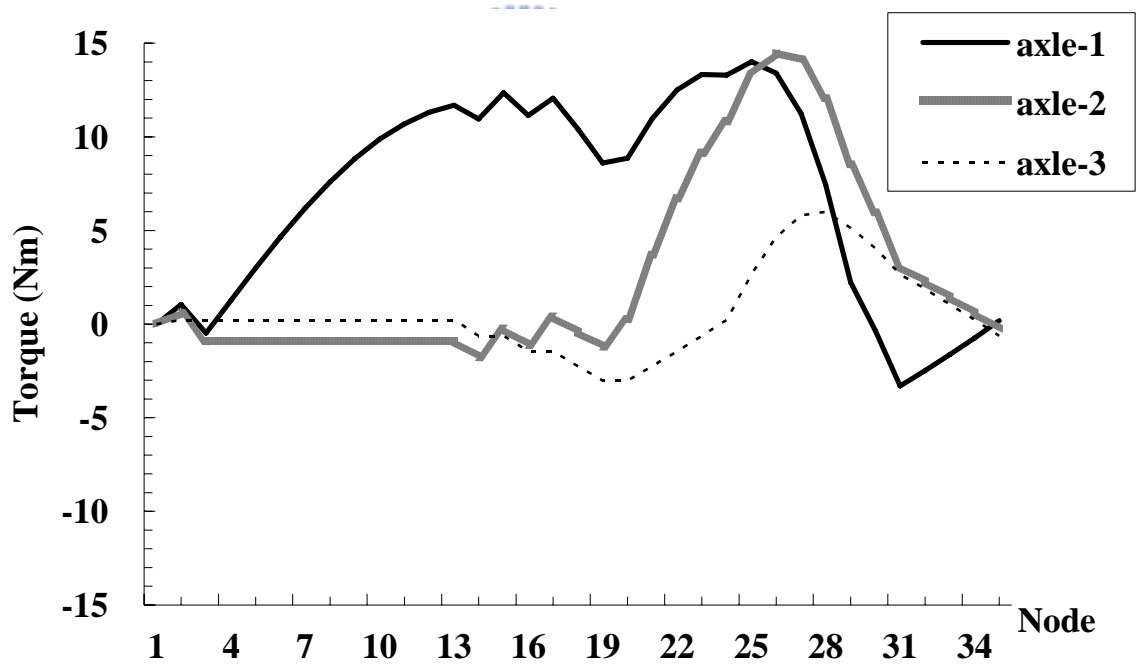


圖 4.4 力矩與路徑節點的關係圖(次方係數為 1)

現在我們將上一個例子的次方係數 ω 改為 2，其它設定不變，再次利用程式搜尋最佳路徑，其搜尋結果如下面圖 4.5 與圖 4.7 所示：

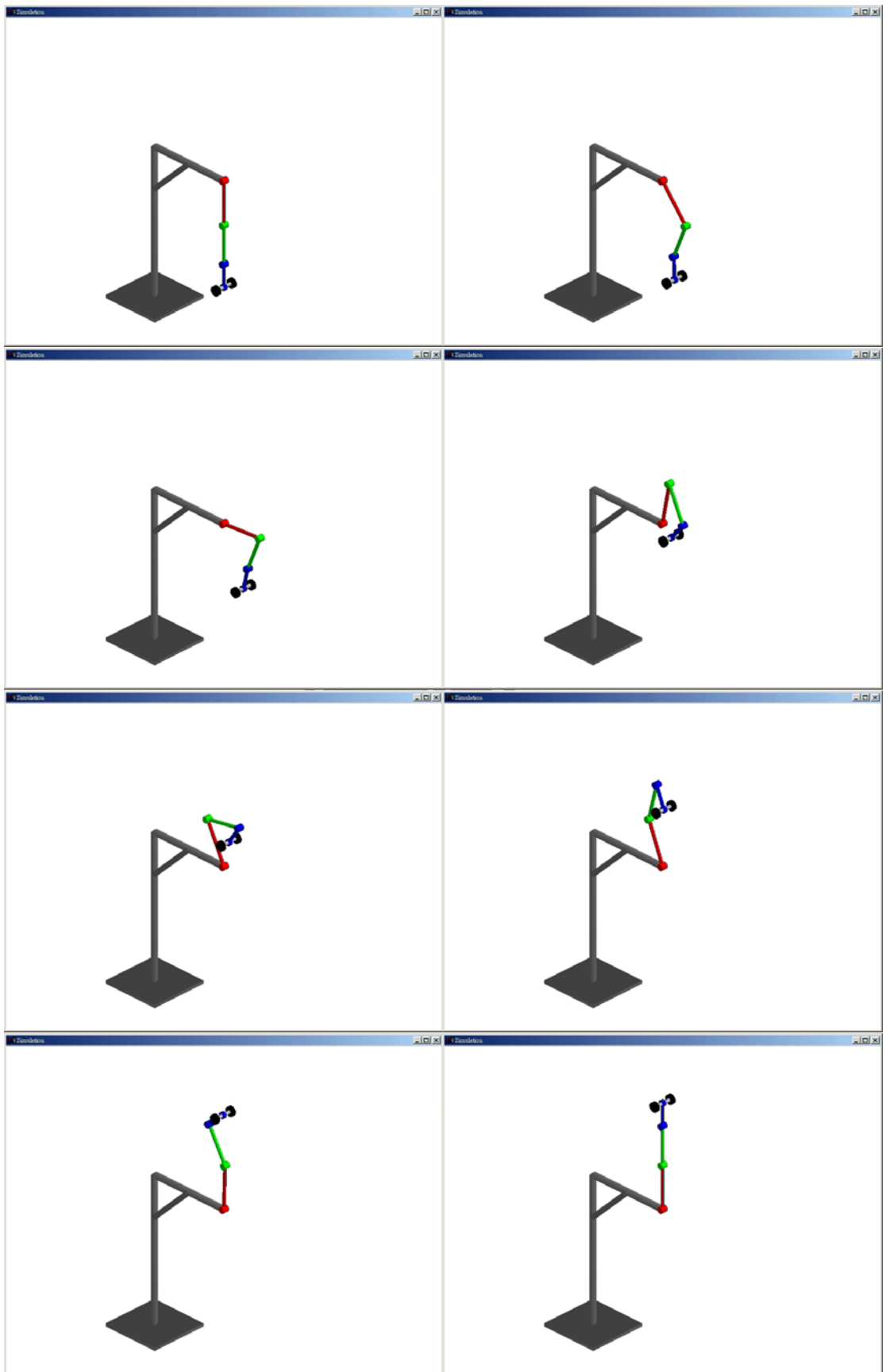


圖 4.5 動態模擬結果(次方係數為 2)

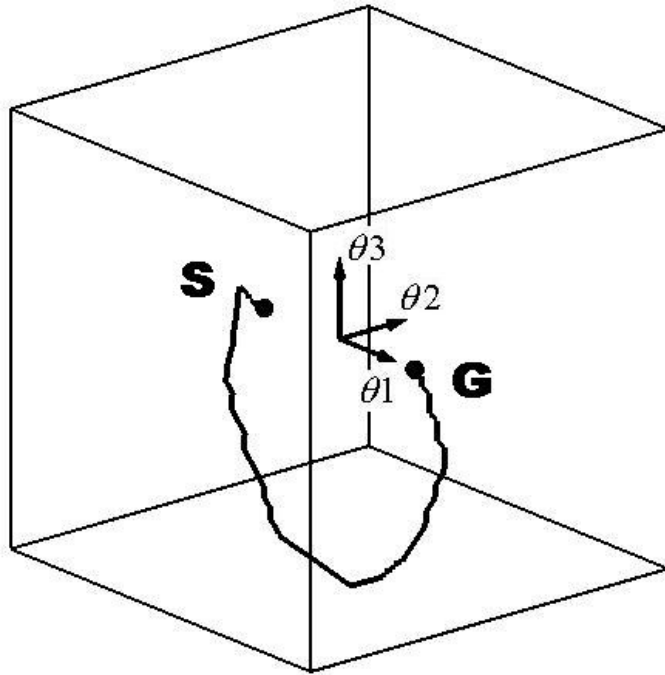


圖 4.6 模擬結果之行進路徑(次方係數為 2)

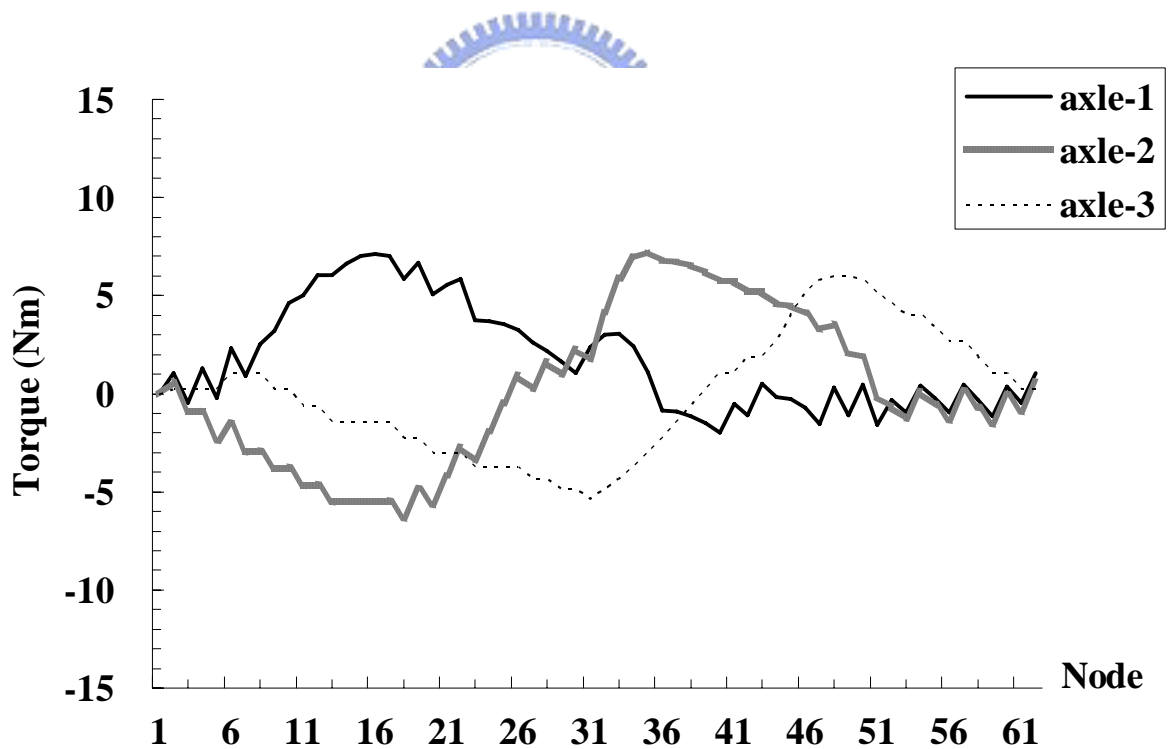


圖 4.7 力矩與路徑節點的關係圖(次方係數為 2)

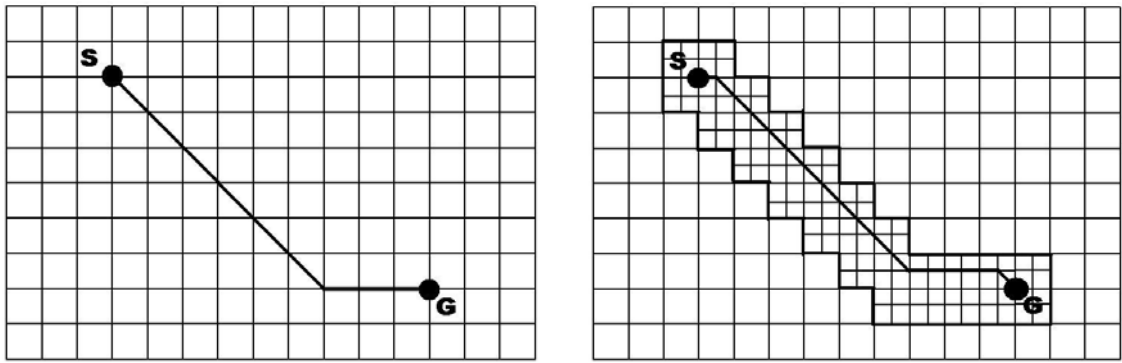
由上面兩個例子的結果，我們可以看出當次方係數為 1 時，搜尋的最佳路

徑比較短，節點數較少，所以總力矩和也比較小，但是所承受的瞬間力矩卻比較大。當次方係數為 2 時，搜尋的最佳路徑比較長，節點數較多，所以總力矩和也比次方係數為 1 時大，但是所承受的瞬間力矩卻比較小。

4.3 Dijkstra 演算法加網格漸細法

雖然上一節我們利用 Dijkstra 演算法成功求得最佳路徑，但搜尋的時間卻花了大約 30 分鐘。若網格尺寸小於 4 度的時候，路徑搜尋的時間則需要花上好幾個小時。為了能快速得到網格尺寸較小的最佳路徑，我們提出了網格漸細法的概念。

Dijkstra 演算法最大的缺點，就是搜尋的廣度很大，因此搜尋的空間和時間也比其他種搜尋法還要很多。然而正因為 Dijkstra 演算法具有很大的搜尋廣度，所以它最佳解的保證性也很高。所以如果我們先以較大的網格尺寸來求得概略的最佳路徑，接著再細分概略最佳路徑附近的網格，以這些被細分的網格為搜尋空間再搜尋精密度更高的最佳路徑。以此類推，不斷細分網格直到網格尺寸為可容許的精密誤差大小，如此我們就可以節省許多時間和空間來求得網格精密度很高的最佳路徑。網格漸細搜尋法的概念圖解如下圖 4.8 所示。



(a)

(b)

圖 4.8 網格漸細法的概念圖 (a)先以較大網格搜尋粗略路徑
(b)細分粗略路徑附近的網格，再搜尋較精密的路徑

接著我們先以每 16 度劃分網格作為初始網格大小作概略最佳路徑規劃；接著再以網格漸細法細分網格尺寸為 8 度，搜尋概略最佳路徑附近的網格。我們同樣加入與上一節相同的限制條件，並設定次方係數 ω 為 1。其結果如下面圖 4.9 與圖 4.11 所示：

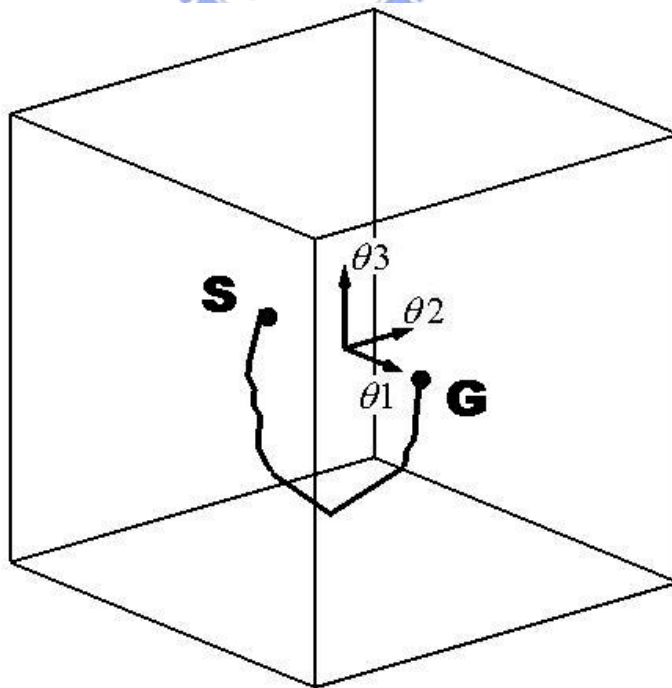


圖 4.9 模擬結果之行進路徑(網格漸細法，次方係數為 1)

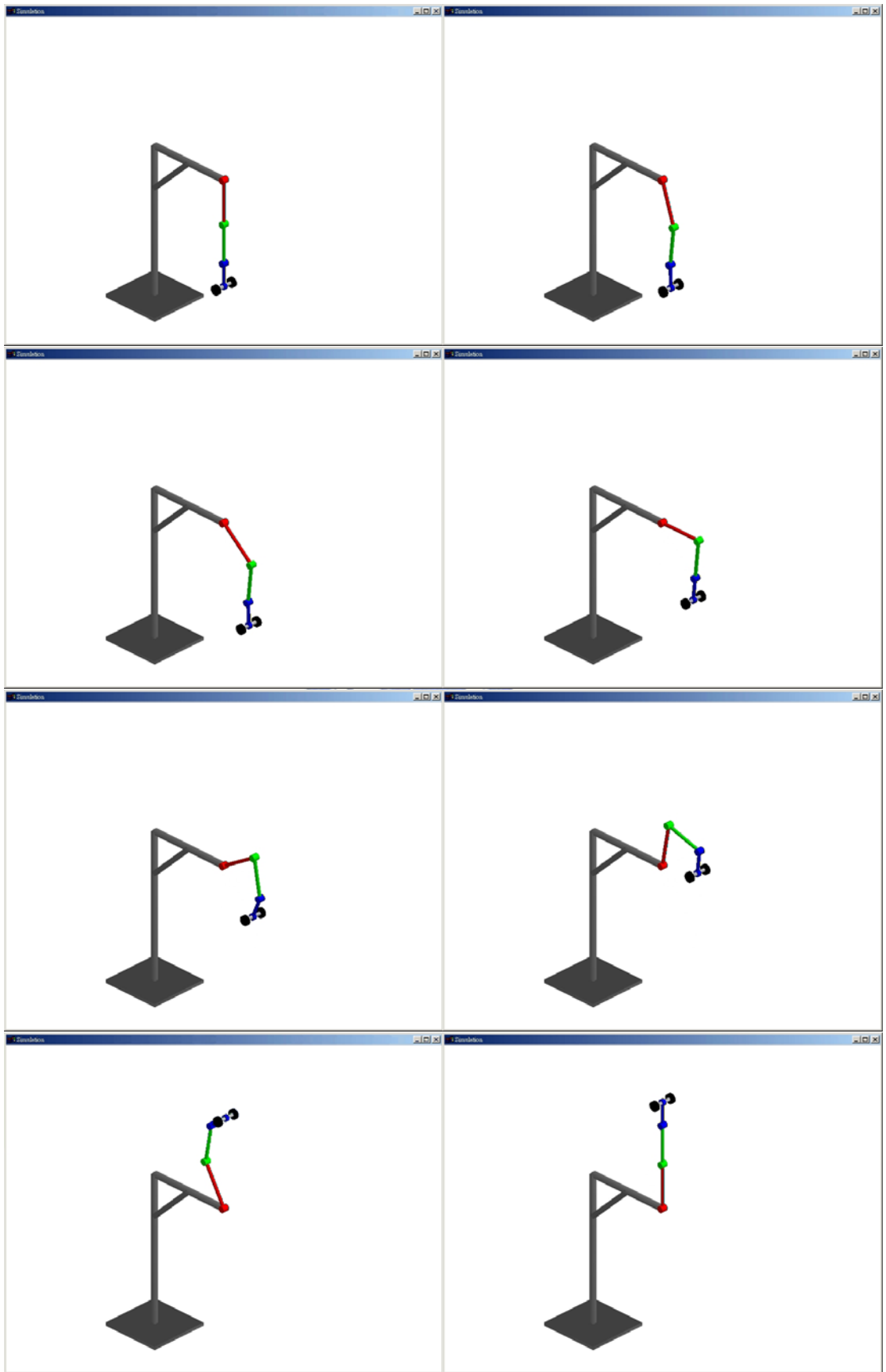


圖 4.10 動態模擬結果(網格漸細法，次方係數為1)

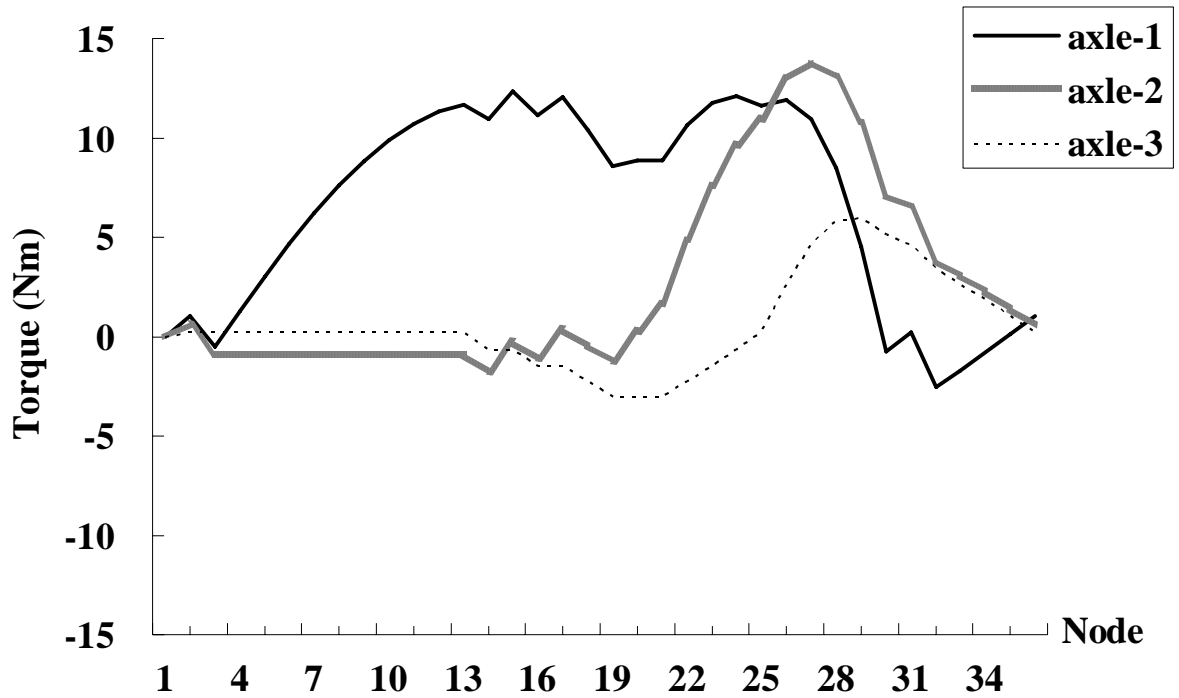


圖 4.11 力矩與路徑節點的關係圖(網格漸細法，次方係數為1)

現在我們將次方係數 ω 改為2，其它設定不變，同樣以網格漸細法搜尋最佳路徑，其結果如下面圖 4.12 與圖 4.14 所示：

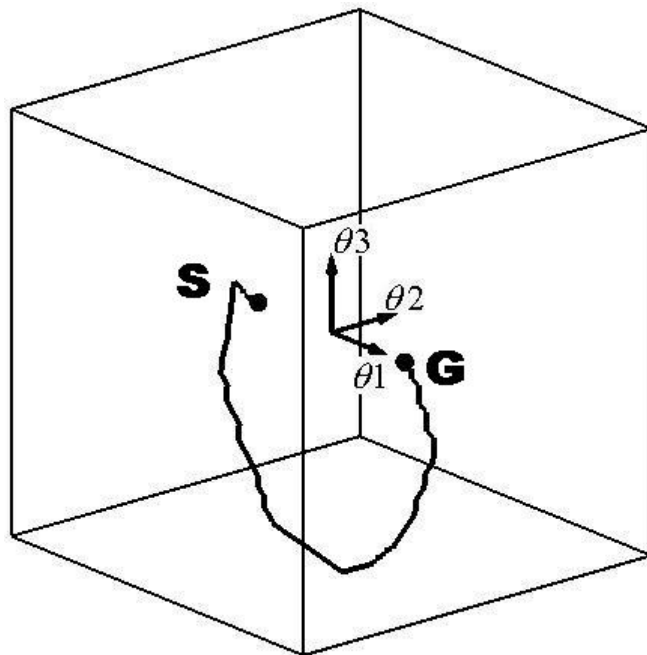


圖 4.12 模擬結果之行進路徑(網格漸細法，次方係數為2)

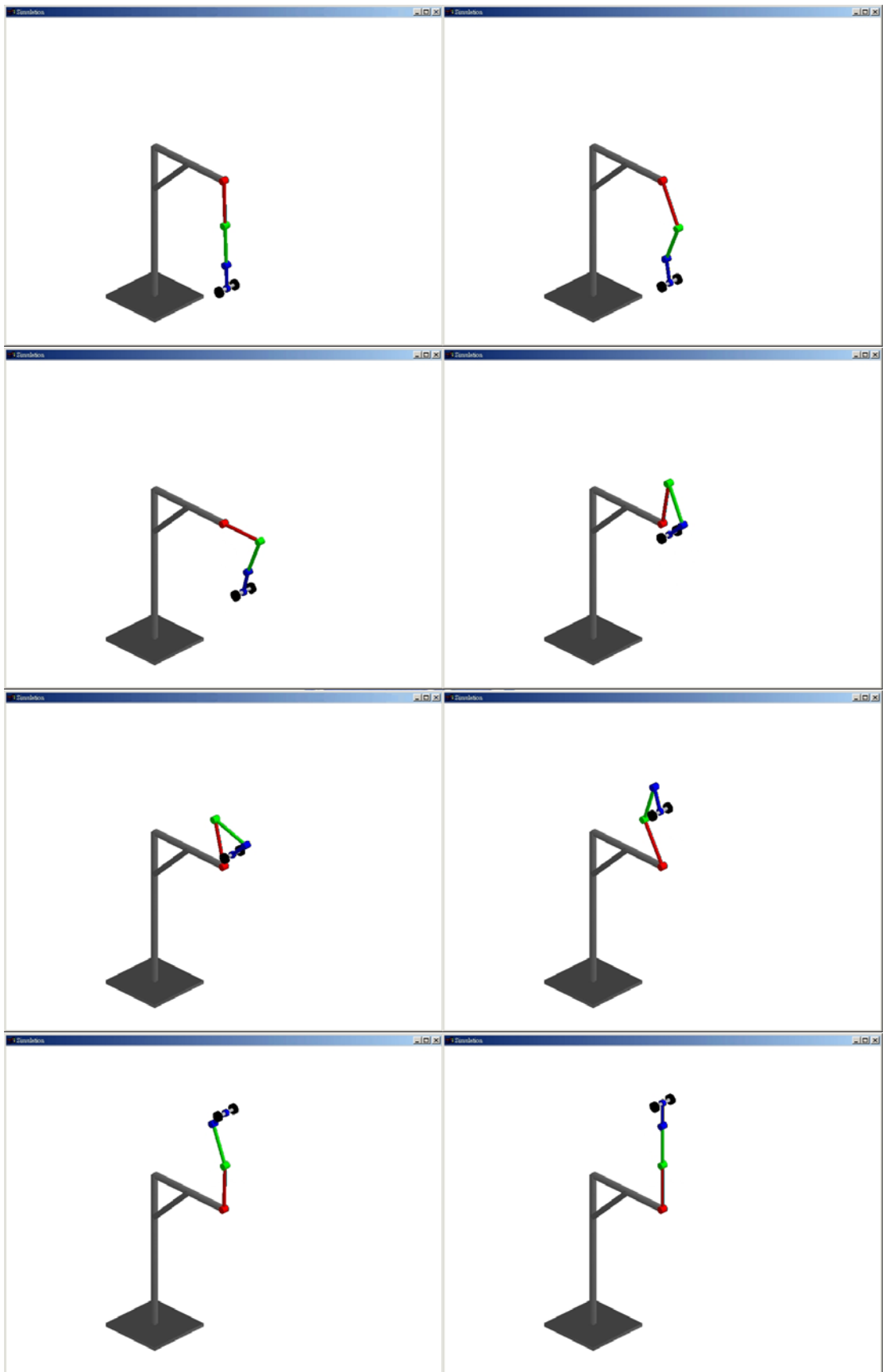


圖 4.13 動態模擬結果(網格漸細法，次方係數為2)

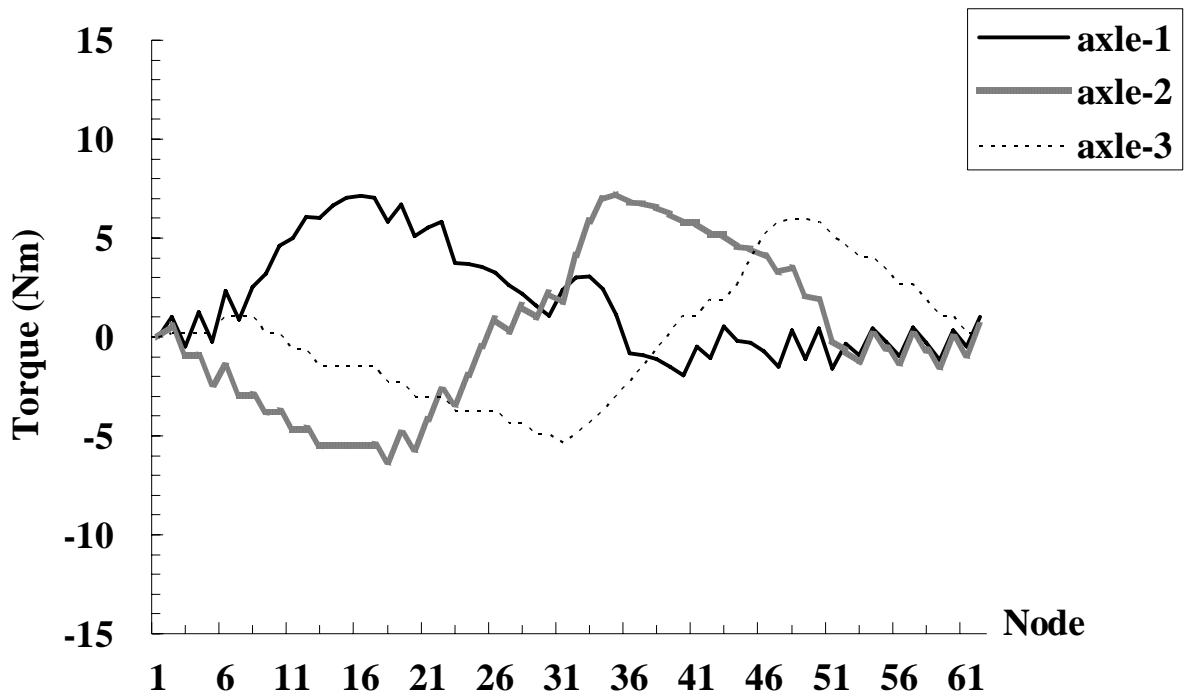


圖 4.14 力矩與路徑節點的關係圖(網格漸細法，次方係數為 2)

由上面兩個例子的結果，我們得到相當好的效果。因為由網格漸細法所得到的路徑與上一節所得到的路徑幾乎一樣，然而搜尋的時間卻大大減少。我們將本節與上一節中的各個例子搜尋路徑所花費的時間整理成表 4.1 表示：

表 4.1 不同搜尋方法與其搜尋時間之比較表

搜尋方法	搜尋時間
Dijkstra 演算法(次方係數為 1)	24 分 25 秒
Dijkstra 演算法(次方係數為 2)	36 分 58 秒
Dijkstra 演算法+網格漸細法(次方係數為 1)	14 秒
Dijkstra 演算法+網格漸細法(次方係數為 2)	15 秒

4.4 B-spline 參數化路徑

在前三節中，我們修正了 Dijkstra 搜尋法，使我們可以快速且正確的找到最佳路徑。然而即使我們可以利用網格漸細法來求得精密度很高的最佳路徑，但其搜尋的時間也相對的增加很多；另外我們使用 Dijkstra 搜尋法所得到的路徑，是由許多的格點所組成的，所以具有不連續且不平順的缺點。根據以上的遺憾，我們可以利用 B-spline Curve 的理論來解決。

在第二章中，我們已經介紹的 B-spline Curve 的理論。現在我們把 4.3 節中所得到的最佳路徑上的節點，當作 B-spline Curve 的控制點，並將它們代入 B-spline Curve 的公式中。如下面(4-2)式，如此我們就可以得到連續且較平滑的路徑。我們取 200 個點重新繪製路徑圖並計算每一點的力矩值，其結果如下圖 4.15 與圖 4.16 所示：

$$\begin{bmatrix} P_{\theta_1}(u) \\ P_{\theta_2}(u) \\ P_{\theta_3}(u) \end{bmatrix} = \sum_{j=0}^n N_{i,k}(u) \cdot \begin{bmatrix} P_{\theta_1,j} \\ P_{\theta_2,j} \\ P_{\theta_3,j} \end{bmatrix} \quad (4-2)$$

其中 $P_{\theta_1}(u)$ 、 $P_{\theta_2}(u)$ 、 $P_{\theta_3}(u)$ 為 B-spline Curve 路徑上節點的三軸角度向量， $P_{\theta_1,j}$ 、 $P_{\theta_2,j}$ 、 $P_{\theta_3,j}$ 為控制點，即為原結果路徑上節點的三軸角度向量， n 為控制點數目，即為原結果路徑的節點總數目。

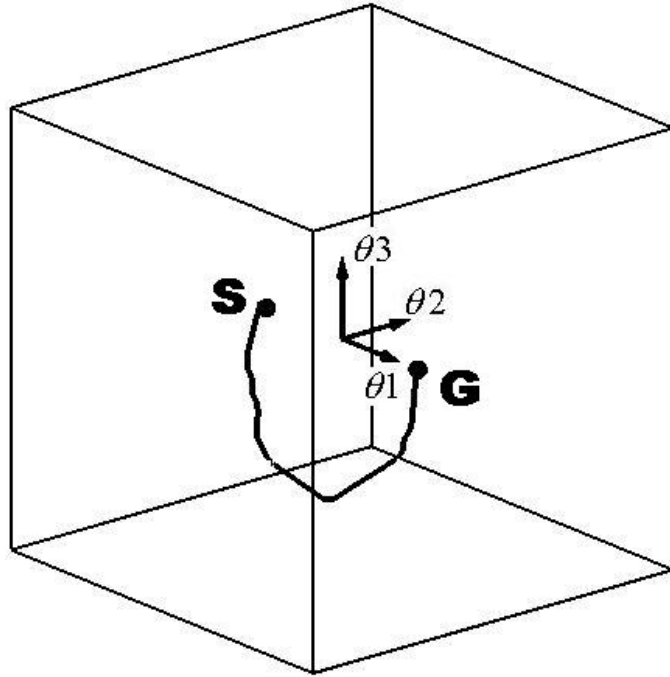


圖 4.15 B-spline Curve 之路徑(網格漸細法，次方係數為 1)

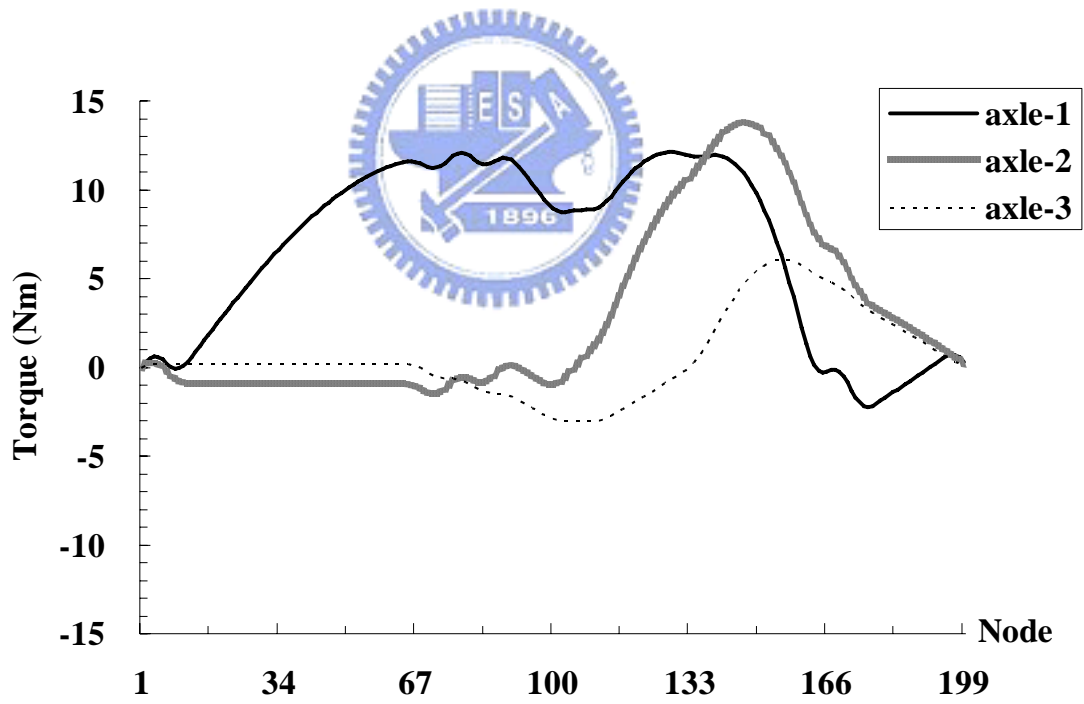


圖 4.16 B-spline Curve 之力矩與路徑節點的關係圖(網格漸細法，次方係數為 1)

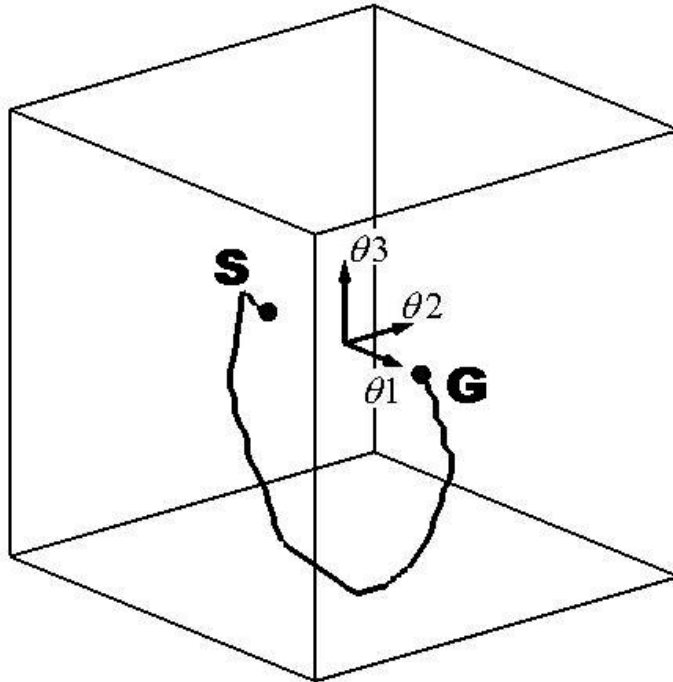


圖 4.17 B-spline Curve 之路徑(網格漸細法，次方係數為 2)

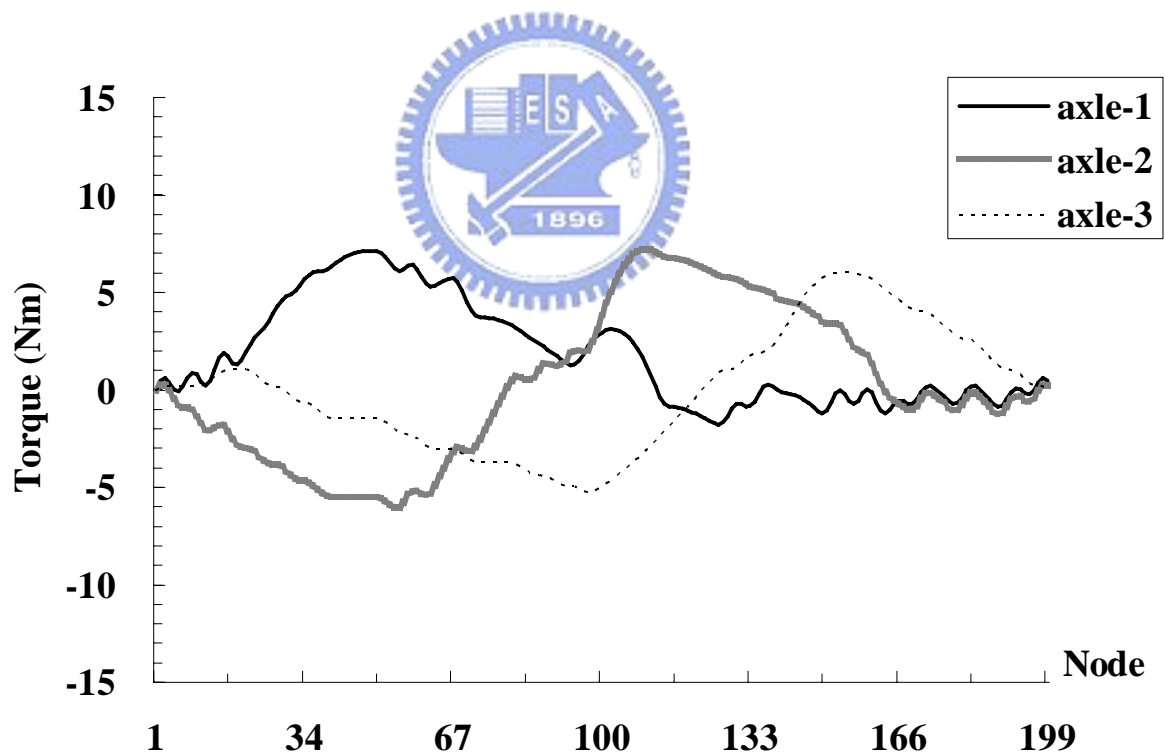


圖 4.18 B-spline Curve 之力矩與路徑節點的關係圖(網格漸細法，次方係數為 2)

由上面的結果看來，將節點當作控制點所建構的 B-spline Curve 與 4.3 節

的結果路徑幾乎相同，其力矩變化也幾乎相同，並且均位於力矩的限制條件內。所以利用 B-spline Curve 來將路徑參數化是可行且成效良好的。

4.5 結果討論

在本章中我們將第三章所設計的目標函數與搜尋路徑的方法，延伸到三軸機器手臂的路徑規劃上。與第三章相同的是，藉由提高力矩的次方係數值(ω)，我們可以得到移動路徑較長、但是三軸的瞬間力矩平均較小的結果，更能有效地減輕驅動器的負擔。另外我們在 4.3 節中提出網格漸細法的觀念，藉著一次一次將網格逐由粗略逐漸細分，來求得較精密的最佳路徑解。如此我們不但可以保持最佳路徑的可靠度，還能兼顧搜尋的速度。最後我們以 B-spline Curve 的理論來求得參數化的路徑，這樣我們可以得到連續且較平順的路徑。雖然僅以靜力學來討論機器手臂的運動情形，於理論上不夠嚴謹。但是對於移動速度緩慢且穩定的機器手臂來說，僅以靜力學來討論是可以被接受的。僅以靜力學討論的路徑規劃問題，其搜尋速度相當快，所以可以應用到需要作即時反應的機器人。