

行政院原子能委員會
委託研究計畫研究報告

核設施除役熱切割遙控蛇形機械載具研究開發

**Highly-Redundant Manipulator (Snake-type Robot) Design for
Thermal Cutting Operation in Radioactive Environment**

計畫編號：108A009

受委託機關(構)：元智大學

計畫主持人：陳傳生

聯絡電話：03 4638800 分機 2472

E-mail address：mecch@saturn.yzu.edu.tw

研究期程：中華民國 108 年 5 月至 108 年 12 月

研究經費：新臺幣 67 萬元

核研所聯絡人員：陳建富

報告日期：108 年 12 月 25 日

目 錄

目 錄.....	I
中文摘要.....	1
ABSTRACT	2
壹、計畫緣起與目的.....	4
貳、研究方法與過程.....	6
一、蛇型機械載具設計	6
二、蛇型機械臂運動學分析	7
三、蛇型機械臂電腦模型	10
四、蛇型機械臂逆向機動學	10
五、繩長與角度轉換公式	12
六、機械臂關節控制系統	14
參、主要發現與結論.....	16
肆、參考文獻.....	17

中文摘要

核設施除役工作需要借助機械臂，經過複雜管道，進行環境檢測以及拆解大型核能組件的工作。而目前能穿越彎曲管路的機械臂構型以蛇型或稱高冗餘自由度關節型機械臂(Hyper-Redundant Articulated Manipulator，簡稱 HRAM)為首選。本計畫提案以開發能遙控操作，精確定位、與穿越彎曲管道，進入保護殼內探勘及進行熱切割工作的雛型蛇型機械臂為標的。本期計畫為設計與製作一具擁有 6 節，每節具有兩個自由度，全部至少 12 個自由度的蛇型機械臂及其運動控制軟體。基於實際需求，此蛇型機械臂採中空腔體設計：電源與信號線路可從內部通過，到達前端安裝座。此安裝座可裝置攝影機或切割工具。完成的機械臂可使用視覺回授方式，遙控指揮機械臂身軀通過彎曲管道，同時控制前端工具(安裝座)的指向。機械設計與控制系統的架構具延伸性，可增加節數。

計畫完成預定項目有：

1. 六節蛇型機械臂的 CAD 設計與 3D 印表機製作完成雛型；
2. 該機械臂的正向運動方程式推導；
3. 逆向運動計算方法設計與模擬；
4. 逆向運動，控制纜繩長度計算公式推導；
5. 網路運動控制系統設計與實驗；
6. 兩節機械臂雛型控制實驗

Abstract

Robots or tele-operated manipulators are indispensable tools for the safe removal of contaminated hardware during the decommissioning of a nuclear power plant in order to avoid the radioactive exposure risk on personnel. This goal of this proposal is to design and implement a prototype for a Hyper-Redundant Articulate Manipulator , or snake-type robot , which can launch through through-wall tubes to enter the suspected high level radioactive area.

The proposed manipulator consists of 6 cable-driven segments with 12 degrees of freedom in total. Each segment has two d.o.f. control and a hollow shaft design to provide the passage for front mounted cameras or thermal cutting tools.

The project has implemented the following items ,

1. Designed a six-segment manipulator and built a prototype robot with 3D printer for verification;
2. Derived the forward kinematics equations and build the 3D model for the robot simulation.
3. Developed the program to calculate the inverse kinematics problem for control computer;
4. Derived the formulation for cable length control and simulate the

inverse kinematic simulation for the target manipulator.

5. Implemented the network control system with EtherCAT protocol and verified the control system in a two-segment prototype.

壹、計畫緣起與目的

核能設施除役時，低汙染區域可以用大型的遙控機具或機械人進行拆除的工作；高汙染區域，必須使用遙控機具進入水泥保護殼內的高輻射區域，探勘與進行熱切割工作或清除輻射。大型輻射汙染物切割成小件後，方便移出，並減少長期封存所需的體積[1, 2]。若要不破壞外圍隔離保護殼，便能進入輻射區工作，各國普遍採用蛇型機械臂[3]，穿越通過管線進入的方式進行。例如，英國 OC Robotics 的蛇型機械臂 Laser Snake 2 (圖 1)進入反應爐核心 進行雷射切割[4]；美國卡耐基美隆大學的機械蛇探勘反應爐等[5]。國內核能發電廠除役在即，運用機械臂執行除役工作勢在必行，故提出本計畫以期能掌握相關基礎與應用技術。



圖 1. OC Robotics Laser Snake 2

固定基座式纜繩蛇型機械臂有兩種形式：關節式與連續式 (continuum robot)：前者關節採用萬向接頭(universal joint)，獨立控

制 yaw-pitch；後者採用撓性關節[6]，可以在任意方向彎曲。但是後者受限於撓性，無法承受高酬載，都用於微創手術。工業上仍以關節式為主。

為能在蜿蜒的管線前進或繞過遮蔽物，高冗餘度關節型機械臂或稱蛇型機械臂，具有高於六軸關節型機械臂的自由度。其設計多半採用多段關節設計：每節具有兩個自由度，節間以萬向接頭連接，總節數至少四段。驅動方式則依工作性質而異：探勘型機械蛇，因為酬載攝影機的重量輕，每節都有獨立的驅動器(馬達或是液壓)；而工作型機械臂的酬載高，為避免靠近工作點的遠端驅動器重量對近端驅動器造成過大負荷，多半採用纜繩拉動的方式，所有驅動器裝在基座，透過纜繩拉動遠端的關節，以降低關節的慣性和驅動器的功率。

貳、研究方法與過程

一、蛇型機械載具設計

蛇型機械手臂採用模組化設計，每節身軀的結構型完全相同，由三種模組：身軀(link)，端板(end plate)與球型萬向節(universal joint). 除身軀的長度可依工作需求變更長度外，其餘模組都是相同尺寸，可以互換使用.端板上有通孔，可供纜繩通過.身軀長度為 300mm，直徑為 120mm. 身軀之間以萬向接頭連接，身軀間有兩個旋轉自由度(俯仰/方位)，具有 $\pm 45^\circ$ 運動範圍。

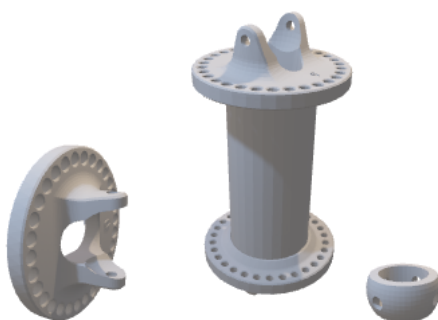


圖 2. 蛇型機械臂模組(端板，組零件和球型萬向接頭)

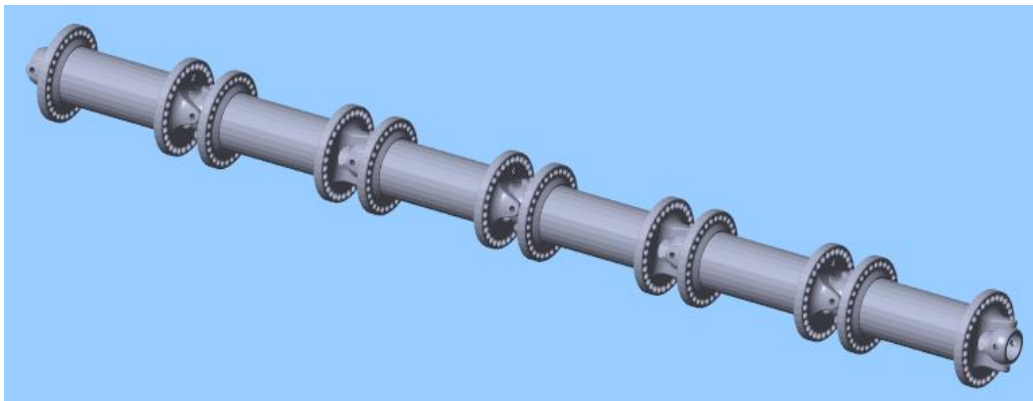


圖 3. 蛇型機械臂設計的組合模型

完成的設計經過3D印表機製作實體模型，以驗證正確性。

二、蛇型機械臂運動學分析

蛇型機械臂的操控是拉動固定在每個端板的纜繩，以改變在每個萬向節頭前後兩個端板的俯仰/方位角度.而末端的工具的空間位置受到全部關節角度地影響.控制蛇行機械臂末端工具的控制位置，必須進行運動學分析，求出關節角度與末端空間位置的數學關係.

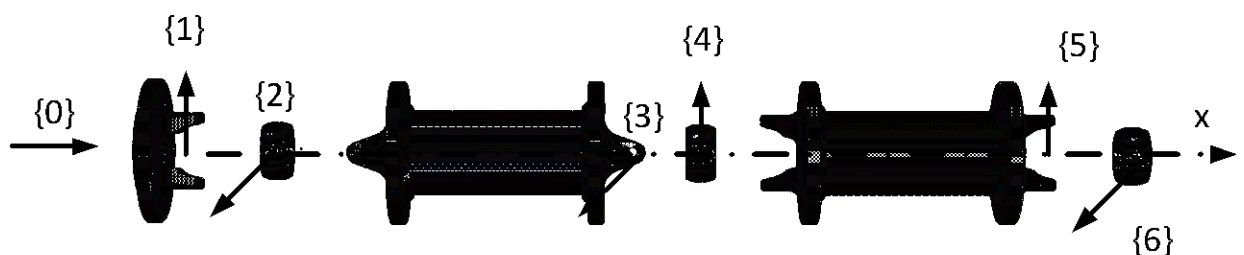


圖 4. 蛇型機械臂的座標系統: {}內是數字是座標系統編號;除底座{0}外，座標系統的 x-軸朝向圖的右側;

表 1: D-H 參數表

Link 編號	角度 θ_i	長 度 d_i	距 離 d_i	扭角 twist α_i
1	θ_1	0	0	$\pi/2$
2	θ_2	0	L_1	0
3	θ_3	0	0	$\pi/2$
4	θ_4	0	L_2	0
5	θ_5	0	0	$\pi/2$
6	θ_6	0	L_3	0
7	θ_7	0	0	$\pi/2$
8	θ_8	0	L_4	0
9	θ_9	0	0	$\pi/2$
10	θ_{10}	0	L_5	0
11	θ_{11}	0	0	$\pi/2$
12	θ_{12}	0	L_6	0

建立數學方程式前，必須先定義機械臂的座標系統，根據標準的 Denavit-Hatenberg 參數定義方法:平移或旋轉軸定義為 z-軸.賦予每節身軀的座標系統如圖 4 所示。從底座(編號 0)開始分別為 link_0，link_1，link_2...，link_12，最末端是工具端。其中，奇數編號

的連桿是萬向接頭;偶數編號是身軀。全部的 D-H 參數臚列在表 1 內。

根據 D-H 的座標計算方式，相鄰坐標系統 (j-1, j) 的轉換矩陣是:

$${}^{j-1}T_j = R_z(\theta_i) \cdot T_z(d_i) \cdot T_x(a_i) \cdot R_x(\alpha_i)$$

其中:

$$R_z(\theta) = \begin{pmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

$$T_z(d) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

$$T_x(a) = \begin{pmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

$$R_x(\alpha) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha & 0 \\ 0 & \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

末端工具的空間位置 $\mathbf{x}_{12} = (x_{12}, y_{12}, z_{12})$ 於底座的座標的計算公式為:

$${}^0\mathbf{x}_{12} = {}^0T_1 \cdot {}^1T_2 \cdot {}^2T_3 \cdot \dots \cdot {}^{11}T_{12} \cdot \mathbf{x}_{12}$$

在蛇型機械臂的設計中，機械臂的角度 $\theta_1, \theta_2, \theta_3, \dots, \theta_{12}$ 是可以控制.所以工具空間座標是所有關節角度的函數:

$${}^0x_{12} = f_{12}(\theta_1, \theta_2, \theta_3, \dots, \theta_{12})$$

由於詳細計算公式請見附件。

三、蛇型機械臂電腦模型

上述機械臂在 Simulink 模擬軟體內建立模型，並透過 robotic toolbox 驗證公式計算的正確性;因為計畫目標是建立控制系統，所以計算必須有公式，才能寫成計算程式。

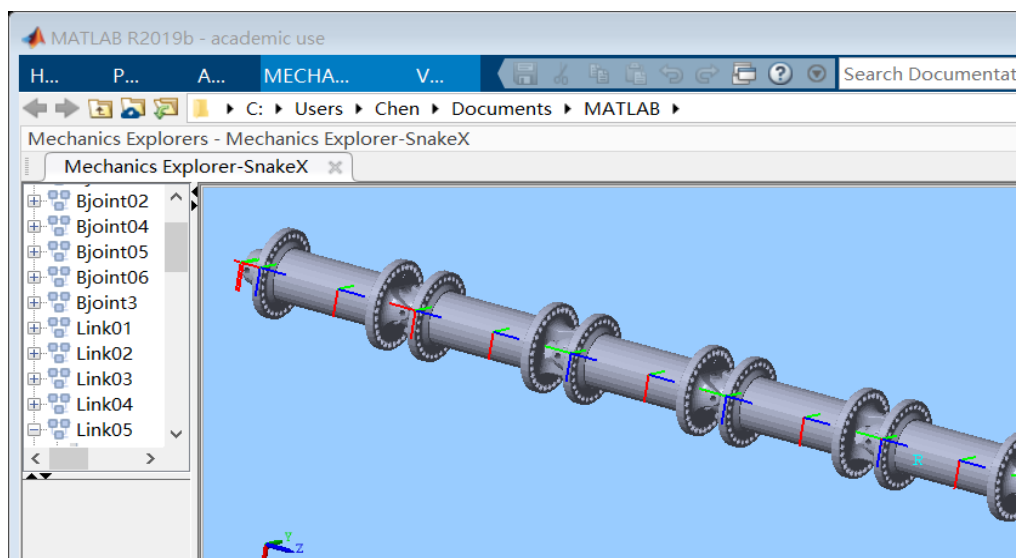


圖 5. 蛇型機械臂電腦模型

四、蛇型機械臂逆向機動學

操作機械臂時主要是控制末端工具的空間位置為準.末端空間

位置是由關節的角度決定，必先從工具位置 x_{12} 推算所需各關節的角度，由於蛇行機械臂的關節數目超過六個(這裡是 $6 \times 2 = 12$ 個)，代表可以到達同一工具位置的關節角度組合 $(\theta_1, \theta_2, \dots, \theta_{12})$ 不只一組。雖然上一節已經推得從角度換算為末端空間位置的正向機動學公式： $x_{12} = f_1(\theta_1, \theta_2, \dots, \theta_{12})$ ，但是逆向運動學的解析解： $(\theta_1, \theta_2, \dots, \theta_{12}) = f_2(x_{12})$ 尚未發現。加上此一函數是非線型函數，目前文獻上處理的方式是透過最佳化的方法求得解答，亦即是設計 cost function: $\min_{\theta} \left| (x_{12} - f_1(\theta_1, \theta_2, \dots, \theta_{12}))^2 \right|$ ，找出使偏離指定末端工具位置的誤差最小的關節角度解答。而常用的最佳化計算方法是使用。下圖是根據我們的機械臂手進行的模擬與計算。同時也計算出控制關節，所需要改變的纜繩長度(請見下節內容)。

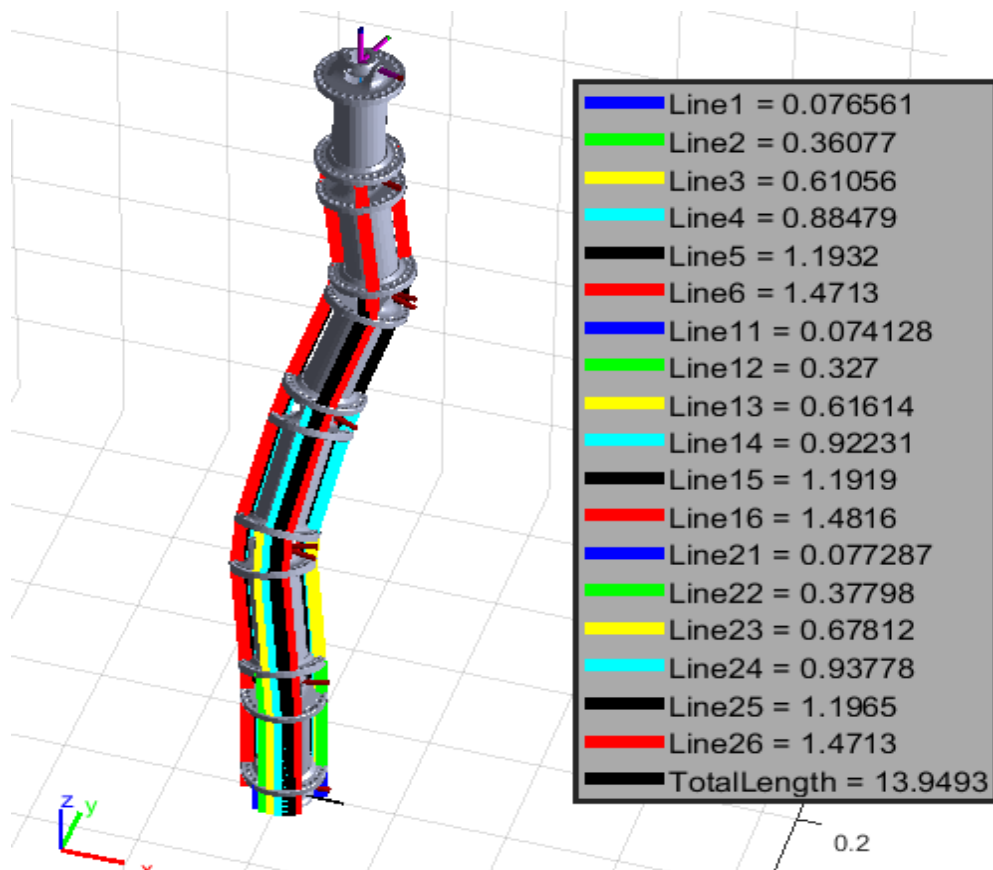


圖 6.關節機械臂進行圓周運動的模擬結果，右邊顯示纜繩長度變化(單位:公尺).

五、繩長與角度轉換公式

由於本提案是針對熱切割工作，需要安裝雷射或電漿工具及夾取工具，故採用纜繩驅動設計增加酬載能力。初期採用 4 節-8 自由度設計進行概念設計，未來可擴充到 12 節-24 自由度節設計。

每節具有兩個自由度，但因為纜繩僅能提供拉力，必須採用三驅動器方式，維持每條纜繩都在拉力狀態，同時可以提高關節的扭力輸出。

在蛇型機械臂的設計中，每個萬向關節具有兩個自由度，從關節相隔距離 d 的前後兩個端板的觀點而言分別是俯仰(pitch, ρ) 與方位角(yaw, ψ).由於每個關節是由等角度分隔 120 度在半徑 r 的圓周的三條纜繩控制，如以纜繩相對於 z -軸的圓周分布角度(δ_i)為準，則該關節的角度(ψ, ρ)，與此三條纜繩存在函數關係，控制纜繩的長度，即可控制此關節的方位與俯仰角度。

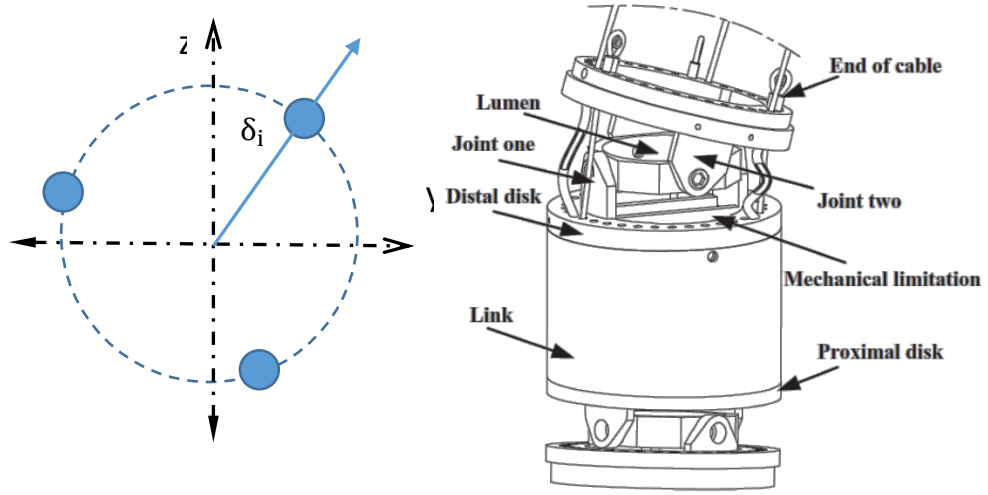


圖 7.關節纜繩通孔的定位方式[7]

根據附件的推導結果，圓周角度位置在 δ 纜繩，因為關節角度改百年(ψ, ρ)導致得變化角度為：

$$l_{c,i}(\delta, \psi, \rho) = \sqrt{(A_1 + A_2 + A_3)}$$

其中：

$$A_1 = \frac{d^2}{2} (1 + \cos \psi \cos \rho)$$

$$A_2 = 2r^2(1 - \cos \rho - \cos \psi \cos \delta^2 + \cos \delta^2 \cos \rho \\ - \cos \delta \sin \psi \sin \delta \sin \rho)$$

$$A_3 = dr(\sin \delta \sin \rho - \cos \delta \sin \psi \\ - \cos \delta \cos \rho \sin \psi + \cos \psi \sin \delta \sin \rho)$$

由於纜繩連接到控制的端板，同時也通過前面所有的關節，例如控制第 4 號關節的三條纜繩也會經過 1，2，3 號關節。所以，第 4 號關節的纜繩的全部改變長度，也會受到 1，2，3 纜繩的影響。亦即是：

$$L_{c,i} = \sum_{i=1}^{i-1} l_{c,\delta} ,$$

其中 $l_{c,\delta}$ 是在 δ 位置的纜繩的變化量。

公式推導過程請參考附件。

六、機械臂關節控制系統

基於減少配線與關節數量擴充的考慮，控制系統採用分散式架構：採用 EtherCAT 通訊協定，採用標準網路線，以 Daisy-Chain 形式，連結所有馬達運動控制驅動器，其總數可達 64 個以上。主控制電腦負責執行規劃運動軌跡與命令下達，關節運動的閉路控制則由驅動器負責圖 8。

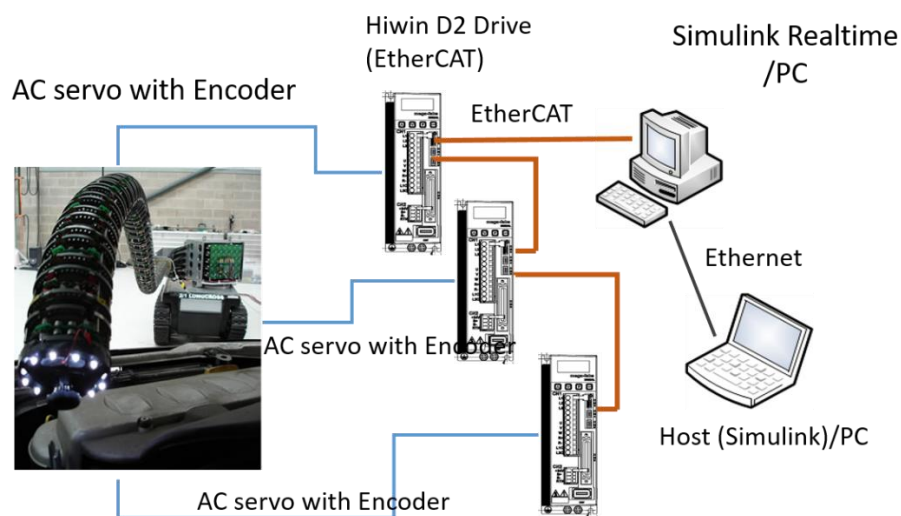


圖 8. EtherCAT 機械臂控制系統

由於馬達數量龐大，控制程式開發我們使用 Simulink Realtime，以方塊圖的方式，開發控制軟體。所見的方塊圖模型可以直接轉換為 C 程式語言在不同電腦系統控制。

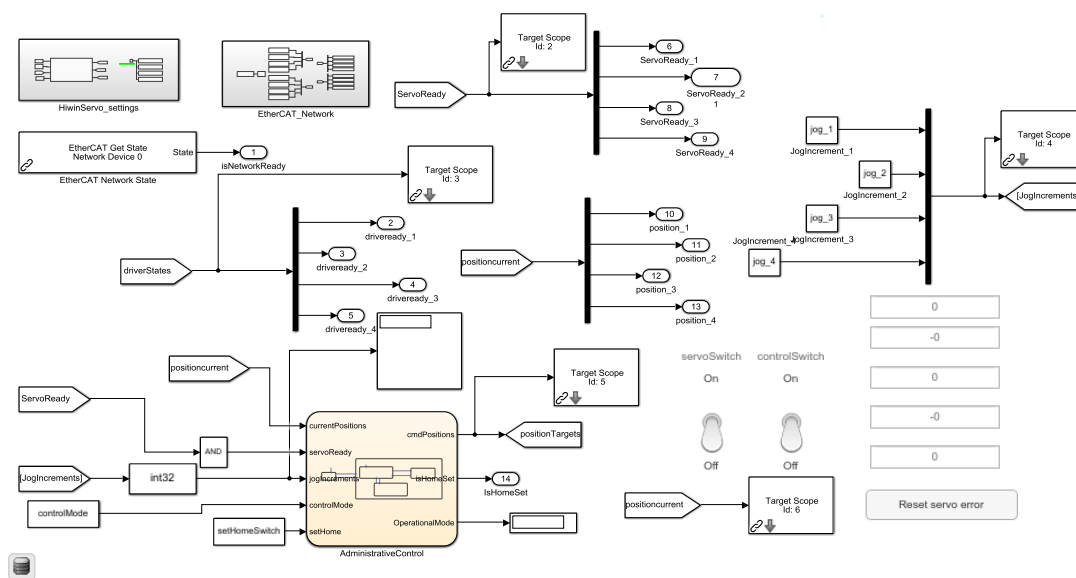


圖 9. EtherCAT 機械臂控制系統

參、主要發現與結論

- 一、蛇型機械臂的設計與包含控制系統的離型系統，以及軟體開發模型，建立後續發展的基礎。
- 二、控制蛇型機械臂所需的正向/逆向運動學計算，包含纜繩長度的計算公式也已經完成。
- 三、蛇型機械臂除機械控制系統之外，還須影像導引 visual servo 能力，可以在下階段進行。

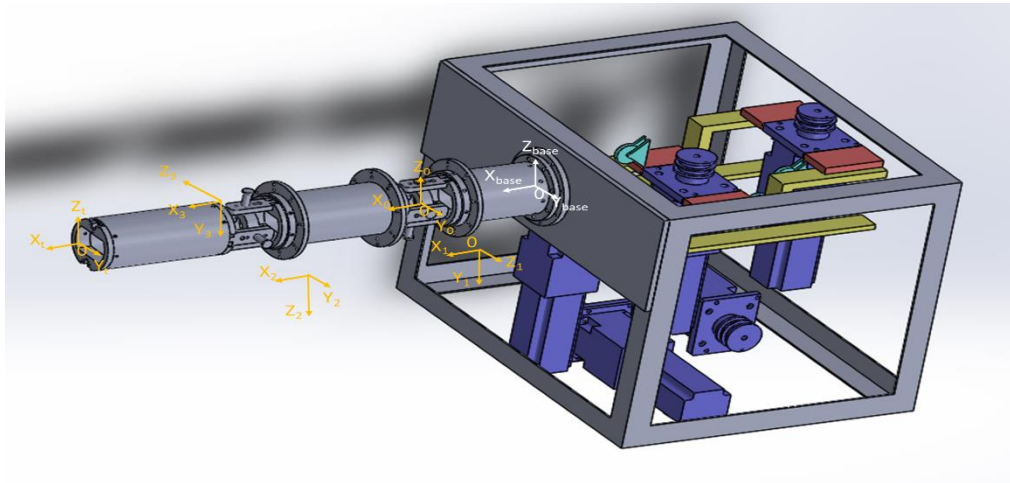
肆、參考文獻

1. R. Bogue , "Robots in the nuclear industry: a review of technologies and applications , " *Industrial Robot: the international journal of robotics research and application* , vol. 38 , no. 2 , pp. 113-118 , 2011. °
2. M. Laraia , *Nuclear decommissioning: Planning , execution and international experience*. 2012 , pp. 1-824 °
3. S. NAKAJIMA , "Japan's advanced snake robot for disaster site inspection , " *Nikkei Asian Review* , July 13 , 2018 2018 °
4. W. N. News. (2018). Robots get smarter to help with decommissioning. Available: <http://www.world-nuclear-news.org/Articles/Robots-get-smarter-to-help-with-decommissioning> °
5. J. Falconer. (2013). CMU's snake robot explores defunct nuclear power plant. Available: <https://newatlas.com/cmu-snake-robot-explores-nuclear-power-plant/28235/> °
6. S. Cobos-Guzman , D. Palmer , and D. Axinte , "Kinematic model to control the end-effector of a continuum robot for multi-axis processing , " *Robotica* , vol. 35 , no. 1 , pp. 224-240 , 2017 °
7. L. Tang , J. Wang , Y. Zheng , G. Gu , L. Zhu , and X. Zhu , "Design of a cable-driven hyper-redundant robot with experimental validation , " *International Journal of Advanced Robotic Systems* , vol. 14 , no. 5 , p. 1729881417734458 , 2017 °
8. M. Xanthidis , K. J. Kyriakopoulos , and I. Rekleitis , "Dynamically

efficient kinematics for hyper-redundant manipulators , " in 2016 24th
Mediterranean Conference on Control and Automation (MED) , 2016 ,
pp. 207-213 .

附件一

Denavit-Hatenberg Parameters for a Ssnake robot (RR....R revolute joints)



the table give the following parameters associated with rotational angle , offset , length , twist angle q, d, a, α in order

$$\begin{bmatrix} \theta_1 & 0 & 0 & -\pi/2 \\ \theta_2 & 0 & L_1 & -\pi/2 \\ \theta_3 & 0 & 0 & -\pi/2 \\ \theta_4 & 0 & L_2 & -\pi/2 \end{bmatrix}$$

```
syms q1 q2 q3 q4 real
syms L1 L2 positive
DH=[q1 0 0 -sym(pi/2); q2 0 L1 -sym(pi/2); q3 0 0 -sym(pi/2); q4 0
L2 -sym(pi/2)]
```

DH =

$$\begin{pmatrix} q_1 & 0 & 0 & -\frac{\pi}{2} \\ q_2 & 0 & L_1 & -\frac{\pi}{2} \\ q_3 & 0 & 0 & -\frac{\pi}{2} \\ q_4 & 0 & L_2 & -\frac{\pi}{2} \end{pmatrix}$$

Homogeneous transformation matrix: ${}^{j-1}\xi_j = R_z(\theta_j) \oplus T_z(d) \oplus T_x(a) \oplus R_x(\alpha)$

$$R_z(\theta) = \begin{pmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, T_z(d) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

$$T_x(a) = \begin{pmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, R_x(\alpha) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha & 0 \\ 0 & \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

```
syms q d alpha real ;sym L positive;
syms Rz(q) Tz(d) Tx(a) Rx(alpha)
Rz(q)=[cos(q), -sin(q), sym(0), sym(0);
       sin(q), cos(q), sym(0), sym(0);
       sym(0), sym(0), sym(1), sym(0);
       sym(0), sym(0), sym(0), sym(1)]
```

Rz(q) =

$$\begin{pmatrix} \cos(q) & -\sin(q) & 0 & 0 \\ \sin(q) & \cos(q) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

```
Tz(d)=sym([eye(3,3),[0;0;d];0 0 0 1]),Tx(a)=sym([eye(3,3),[a;0;0];0
0 0 1]),Tx(a)=sym([eye(3,3),[a;0;0];0 0 0 1]),Rx(alpha)=[1 0 0 0;0
cos(alpha), -sin(alpha) 0; 0 sin(alpha), cos(alpha), 0; 0 0 0 1]
```

Tz(d) =

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Tx(a) =

$$\begin{pmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$T_x(a) =$$

$$\begin{pmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$R_x(\alpha) =$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) & 0 \\ 0 & \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$T_x(a)=\text{sym}([\text{eye}(3,3),[a;0;0];0\ 0\ 0\ 1])$$

$$T_x(a) =$$

$$\begin{pmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$R_x(\alpha)=[1\ 0\ 0\ 0;0\ \cos(\alpha),\ -\sin(\alpha)\ 0; 0\ \sin(\alpha),\ \cos(\alpha)\ 0; 0\ 0\ 0\ 1]$$

$$R_x(\alpha) =$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) & 0 \\ 0 & \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\text{zeta_0_1}=\text{Rz}(\text{DH}(1,1))*\text{Tz}(\text{DH}(1,2))*\text{Tx}(\text{DH}(1,3))*\text{Rx}(\text{DH}(1,4))$$

$$\text{zeta_0_1} =$$

$$\begin{pmatrix} \cos(q_1) & 0 & -\sin(q_1) & 0 \\ \sin(q_1) & 0 & \cos(q_1) & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\text{zeta}_{1_2} = \text{Rz}(\text{DH}(2, 1)) * \text{Tz}(\text{DH}(2, 2)) * \text{Tx}(\text{DH}(2, 3)) * \text{Rx}(\text{DH}(2, 4))$$

$$\text{zeta}_{1_2} =$$

$$\begin{pmatrix} \cos(q_2) & 0 & -\sin(q_2) & L_1 \cos(q_2) \\ \sin(q_2) & 0 & \cos(q_2) & L_1 \sin(q_2) \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\text{zeta}_{2_3} = \text{Rz}(\text{DH}(3, 1)) * \text{Tz}(\text{DH}(3, 2)) * \text{Tx}(\text{DH}(3, 3)) * \text{Rx}(\text{DH}(3, 4))$$

$$\text{zeta}_{2_3} =$$

$$\begin{pmatrix} \cos(q_3) & 0 & -\sin(q_3) & 0 \\ \sin(q_3) & 0 & \cos(q_3) & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\text{zeta}_{3_4} = \text{Rz}(\text{DH}(4, 1)) * \text{Tz}(\text{DH}(4, 2)) * \text{Tx}(\text{DH}(4, 3)) * \text{Rx}(\text{DH}(4, 4))$$

$$\text{zeta}_{3_4} =$$

$$\begin{pmatrix} \cos(q_4) & 0 & -\sin(q_4) & L_2 \cos(q_4) \\ \sin(q_4) & 0 & \cos(q_4) & L_2 \sin(q_4) \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\text{zeta}_{0_4} = \text{zeta}_{0_1} * \text{zeta}_{1_2} * \text{zeta}_{2_3} * \text{zeta}_{3_4}$$

$$\text{zeta}_{0_4} =$$

$$\begin{pmatrix} \cos(q_4) \sigma_2 + \cos(q_1) \sin(q_2) \sin(q_4) & \cos(q_1) \cos(q_2) \sin(q_3) - \cos(q_3) \sin(q_1) & \cos(q_1) \cos(q_4) \sin(q_2) - \sin(q_4) \sigma_2 & L_1 \cos(q_4) \\ \sin(q_1) \sin(q_2) \sin(q_4) - \cos(q_4) \sigma_1 & \cos(q_1) \cos(q_3) + \cos(q_2) \sin(q_1) \sin(q_3) & \sin(q_4) \sigma_1 + \cos(q_4) \sin(q_1) \sin(q_2) & L_1 \cos(q_4) \\ \cos(q_2) \sin(q_4) - \cos(q_3) \cos(q_4) \sin(q_2) & -\sin(q_2) \sin(q_3) & \cos(q_2) \cos(q_4) + \cos(q_3) \sin(q_2) \sin(q_4) & L_2 \cos(q_4) \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

where

$$\sigma_1 = \cos(q_1) \sin(q_3) - \cos(q_2) \cos(q_3) \sin(q_1)$$

$$\sigma_2 = \sin(q_1) \sin(q_3) + \cos(q_1) \cos(q_2) \cos(q_3)$$

`simplify(zeta_0_4)`

`ans =`

$$\begin{pmatrix} \cos(q_4) \sigma_2 + \cos(q_1) \sin(q_2) \sin(q_4) & \cos(q_1) \cos(q_2) \sin(q_3) - \cos(q_3) \sin(q_1) & \cos(q_1) \cos(q_4) \sin(q_2) - \sin(q_4) \sigma_2 & L_1 \cos(q_4) \\ \sin(q_1) \sin(q_2) \sin(q_4) - \cos(q_4) \sigma_1 & \cos(q_1) \cos(q_3) + \cos(q_2) \sin(q_1) \sin(q_3) & \sin(q_4) \sigma_1 + \cos(q_4) \sin(q_1) \sin(q_2) & L_1 \cos(q_4) \\ \cos(q_2) \sin(q_4) - \cos(q_3) \cos(q_4) \sin(q_2) & -\sin(q_2) \sin(q_3) & \cos(q_2) \cos(q_4) + \cos(q_3) \sin(q_2) \sin(q_4) & L_2 \cos(q_4) \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

where

$$\sigma_1 = \cos(q_1) \sin(q_3) - \cos(q_2) \cos(q_3) \sin(q_1)$$

$$\sigma_2 = \sin(q_1) \sin(q_3) + \cos(q_1) \cos(q_2) \cos(q_3)$$

加入底座到 $\{x_0, y_0, z_0\}$ 座標系的轉換: ${}^{\text{base}}\xi_0$ 和末端到工具端的轉換 ${}^4\xi_{\text{tool}}$

```
syms L0 l_t positive
zeta_b_0=Rz(0)*Tz(0)*Tx(L0)*Rx(0) , ...
zeta_4_t=Rz(0)*Tz(0)*Tx(l_t)*Rx(0)
```

`zeta_b_0 =`

$$\begin{pmatrix} 1 & 0 & 0 & L_0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

`zeta_4_t =`

$$\begin{pmatrix} 1 & 0 & 0 & l_t \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

如果要算從 **base** 到工具的轉換矩陣: ${}^b\xi_t = {}^{\text{base}}\xi_0 \oplus {}^0\xi_4 \oplus {}^4\xi_{\text{tool}}$

```
simplify(zeta_b_0*zeta_0_4*zeta_4_t)
```

ans =

$$\begin{pmatrix} \sigma_4 & \cos(q_1) \cos(q_2) \sin(q_3) - \cos(q_3) \sin(q_1) & \cos(q_1) \cos(q_4) \sin(q_2) - \sin(q_4) \sigma_5 & L_0 + l_t \sigma_4 + L_1 \cos(q_1) \cos(q_2) \sin(q_3) \\ \sigma_3 - \cos(q_4) \sigma_1 & \cos(q_1) \cos(q_3) + \cos(q_2) \sin(q_1) \sin(q_3) & \sin(q_4) \sigma_1 + \cos(q_4) \sin(q_1) \sin(q_2) & L_1 \cos(q_2) \sin(q_1) - l_t (\cos(q_4) \sigma_1 + \cos(q_2) \sin(q_1) \sin(q_2)) \\ \sigma_2 & -\sin(q_2) \sin(q_3) & \cos(q_2) \cos(q_4) + \cos(q_3) \sin(q_2) \sin(q_4) & l_t \sigma_2 - L_1 \sin(q_2) + L_2 \cos(q_2) \sin(q_1) \sin(q_2) \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

where

$$\sigma_1 = \cos(q_1) \sin(q_3) - \cos(q_2) \cos(q_3) \sin(q_1)$$

$$\sigma_2 = \cos(q_2) \sin(q_4) - \cos(q_3) \cos(q_4) \sin(q_2)$$

$$\sigma_3 = \sin(q_1) \sin(q_2) \sin(q_4)$$

$$\sigma_4 = \cos(q_4) \sigma_5 + \cos(q_1) \sin(q_2) \sin(q_4)$$

$$\sigma_5 = \sin(q_1) \sin(q_3) + \cos(q_1) \cos(q_2) \cos(q_3)$$

附件二

計算兩節之間的對應孔位的距離(纜繩長度). 假設兩端板 endplate 的距離為 d 。因此萬向節的旋轉中心為均分 $d/2$

```
syms r d positive
syms psi theta rho real %psi , theta 分別是萬向關節的 yaw (z) , pitch
(y) angle; delta_i 是每個端板上纜繩孔的位置.
syms q real
```

定義兩個座標系統:第一個固定在前一節($j-1$)，原點在萬向關節的中心， z 軸在旋轉中心.另一個座標系統附屬於 j 節(是 j 節的 body-fixed 座標系統)。相對的 yaw，pitch 角度就是(ψ ， θ)

```
% convert yaw , pitch to axis-angle
[vector , angle]=EulerAngle2AxisAngle(psi , theta , 0);
vector=simplify(vector)
angle=simplify(angle)
%定義旋轉矩陣: Rz: rotation about z-axis , Ry: rotation about y-axis
Rz(q)=[cos(q) , -sin(q) , sym(0);
       sin(q) , cos(q) , sym(0);
       sym(0) , sym(0) , sym(1)]
Ry(q)=[cos(q) , sym(0) , sin(q);
       sym(0) , sym(1) , sym(0);
       -sin(q) , sym(0) , cos(q);]
```

假設端板上的纜繩孔的位置是在($j-1$)節的端板是 $(x, y)=(r*\cos(\delta_i), r*\sin(\delta_i), -d/2)$;在 j 節的端板上，其對應孔位 $(x, y)=(r*\cos(\delta_i), r*\sin(\delta_i), -d/2)$;

```
upHole=[d/2;r*cos(rho); r*sin(rho)];lowerHole=[-d/2;r*cos(rho);
r*sin(rho)]
```

由於 j 節的(上)端板經過 yaw 和 pitch 的旋轉後，移動到

```
upHoleNew=simplify(Rz(psi)*Ry(theta)*upHole)
```

上下端板對應纜繩孔的距離是:

```

    cableLength=sqrt(simplify(expand((upHoleNew-lowerHole)'*(upHoleN
ew-lowerHole))))
    latex(cableLength)
    subs(cableLength, {psi, theta}, {0, 0})
    matlabFunction(cableLength, 'File', 'CableLength', 'Optimize', true);
    type CableLength.m
    Tmatrix=Rz(psi)*Ry(theta);
    [vTmatrix, dTmatrix]=(eig(Tmatrix));
    lambda1=simplify(dTmatrix(1))
    lambda2=simplify(dTmatrix(2))
    lambda3=simplify(dTmatrix(3))
    % tform2axang(Tmatrix)
    qat=RotationMatrix2Quaternion(Tmatrix);
    [axis, angle]=Quaternion2AxisAngle(qat);
    qat=simplify(qat')
    axis=simplify(axis')
    angle=simplify(angle)
    matlabFunction(Tmatrix, 'File', 'TmatrixFunc', 'Optimize', true);
    type TmatrixFunc.m
    % simulink_model_name="Snake_cable_length";
    % if ~exist(simulink_model_name),
    %     new_system(simulink_model_name);
    % end
    % open_system(simulink_model_name);
    % matlabFunctionBlock(simulink_model_name, cableLength)

```

測試

```

    rr=150;dd=100; low=YZCircle([-dd/2,0,0],rr,50);up=YZCircle([dd/2,
0,0],rr,50);
    upAxis=[0,dd/2;0,0;0,0], lowAxis=[0,-dd/2;0,0;0,0]
    psi_r=pi/6;theta_r=sqrt(pi^2/16-psi_r^2);
    upAxisT=TmatrixFunc(psi_r,theta_r)*upAxis
    upT=TmatrixFunc(psi_r,theta_r)*up;
    plot3(up(1,:),up(2,:),up(3,:), '*'), hold on, plot3(low(1,:),
low(2,:),low(3,:), 'd'), plot3(upT(1,:),upT(2,:),upT(3,:), 'r.'),
line(upAxisT(1,:),upAxisT(2,:),upAxisT(3,:)), line(lowAxis(1,:),
lowAxis(2,:),lowAxis(3,:)), axis equal, hold off

    view([57 -62])
    figure % new figure
    plot3circle(up(1,:),up(2,:),up(3,:), 'red'), hold on,
plot3circle(low(1,:),low(2,:),low(3,:), 'cyan'), plot3circle(upT(1,:),
upT(2,:),upT(3,:), 'green');

```

```

line(upAxisT(1,:), upAxisT(2,:), upAxisT(3,:), 'linewidth', 5,
'color', 'blue'), line(lowAxis(1,:), lowAxis(2,:), lowAxis(3,:),
'linewidth', 5, 'color', 'black'), axis equal, hold off

```

```

xlim([-68 140])
ylim([-150 157])
zlim([-153 150])
view([68 -84])

```

vector =

$$\begin{pmatrix} \frac{2 \sin\left(\frac{\psi}{2}\right) \sin\left(\frac{\theta}{2}\right)}{\sqrt{3 - \cos(\theta) - \cos(\psi) \cos(\theta) - \cos(\psi)}} \\ \frac{\cos\left(\frac{\theta}{2}\right) \sin\left(\frac{\psi}{2}\right)}{\sigma_1} \\ \frac{\cos\left(\frac{\psi}{2}\right) \sin\left(\frac{\theta}{2}\right)}{\sigma_1} \end{pmatrix}$$

where

$$\sigma_1 = \sqrt{1 - \cos\left(\frac{\psi}{2}\right)^2 \cos\left(\frac{\theta}{2}\right)^2}$$

angle =

$$2 \arccos\left(\cos\left(\frac{\psi}{2}\right) \cos\left(\frac{\theta}{2}\right)\right)$$

Rz(q) =

$$\begin{pmatrix} \cos(q) & -\sin(q) & 0 \\ \sin(q) & \cos(q) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Ry(q) =

$$\begin{pmatrix} \cos(q) & 0 & \sin(q) \\ 0 & 1 & 0 \\ -\sin(q) & 0 & \cos(q) \end{pmatrix}$$

lowerHole =

$$\begin{pmatrix} -\frac{d}{2} \\ r \cos(\rho) \\ r \sin(\rho) \end{pmatrix}$$

upHoleNew =

$$\begin{pmatrix} \frac{d \cos(\psi) \cos(\theta)}{2} - r \cos(\rho) \sin(\psi) + r \cos(\psi) \sin(\rho) \sin(\theta) \\ r \cos(\psi) \cos(\rho) + \frac{d \cos(\theta) \sin(\psi)}{2} + r \sin(\psi) \sin(\rho) \sin(\theta) \\ r \cos(\theta) \sin(\rho) - \frac{d \sin(\theta)}{2} \end{pmatrix}$$

cableLength =

$$\sqrt{\frac{d^2}{2} - 2 r^2 \cos(\theta) + 2 r^2 - 2 r^2 \cos(\psi) \cos(\rho)^2 + 2 r^2 \cos(\rho)^2 \cos(\theta) + \frac{d^2 \cos(\psi) \cos(\theta)}{2} + d r \sin(\rho) \sin(\theta) - d r \cos(\rho) \sin(\psi) - 2 r^2 \cos(\theta) \sin(\rho) \sin(\psi)}$$

```
ans = '\sqrt{\frac{d^2}{2}-2\, r^2\, \cos\left(\theta \right)+2\, r^2-2\, r^2\, \cos\left(\psi \right)\cos\left(\rho \right)^2+2\, r^2\, \cos\left(\rho \right)^2\cos\left(\theta \right)+\frac{d^2\cos\left(\psi \right)\cos\left(\theta \right)}{2}+d\, r\sin\left(\rho \right)\sin\left(\theta \right)-d\, r\cos\left(\rho \right)\sin\left(\psi \right)-2\, r^2\cos\left(\theta \right)\sin\left(\rho \right)\sin\left(\psi \right)}'

\cos\left(\psi \right)\, ,\, \{\cos\left(\rho \right)\}^2+2\, ,\, r^2\, ,\, \{\cos\left(\rho \right)\}^2\, ,\, \cos\left(\theta \right)+\frac{d^2}{2}\, ,\, \cos\left(\psi \right)\, ,\, \cos\left(\theta \right)\}^2+d\, ,\, r\, ,\, \sin\left(\rho \right)\, ,\, \sin\left(\theta \right)-d\, ,\, r\, ,\, \cos\left(\rho \right)\, ,\, \sin\left(\psi \right)-2\, ,\, r^2\, ,\, \cos\left(\rho \right)\, ,\, \sin\left(\psi \right)\, ,\, \sin\left(\rho \right)\, ,\, \sin\left(\theta \right)-d\, ,\, r\, ,\, \cos\left(\rho \right)\, ,\, \cos\left(\theta \right)\, ,\, \sin\left(\psi \right)+d\, ,\, r\, ,\, \cos\left(\psi \right)\, ,\, \sin\left(\rho \right)\, ,\, \sin\left(\theta \right)\right)'
```

ans =

d

```
function cableLength = CableLength(d,psi,r,rho,theta)
```

```
%CABLELENGTH
```

```
% CABLELENGTH = CABLELENGTH(D,PSI,R,RHO,THETA)
```

```
% This function was generated by the Symbolic Math Toolbox version 8.4.
```

```

%    26-Dec-2019 13:53:53

t2 = cos(psi);

t3 = cos(rho);

t4 = cos(theta);

t5 = sin(psi);

t6 = sin(rho);

t7 = sin(theta);

t8 = d.^2;

t9 = r.^2;

t10 = t3.^2;

cableLength =

sqrt(t8./2.0+t9.*2.0-t4.*t9.*2.0+(t2.*t4.*t8)./2.0-t2.*t9.*t10.*2.0+t4.*t9.*t10.*2.0-d.*r.*t3.*t5+d.*r.*t6.*t7-d.*r.*t3.*t4.*t5+d.*r.*t2.*t6.*t7-t3.*t5.*t6.*t7.*t9.*2.0);

lambda1 =

```

$$\frac{\cos(\psi)}{3} + \frac{\cos(\theta)}{3} + \frac{\sigma_3}{3} + \frac{2^{2/3} \sigma_1}{6} + \frac{3 \cdot 2^{1/3} \left(\frac{\cos(\psi)^2 \cos(\theta)^2}{9} + \frac{2 \cos(\psi)^2 \cos(\theta)}{9} + \frac{\cos(\psi)^2}{9} + \frac{2 \cos(\psi) \cos(\theta)^2}{9} - \frac{\sigma_3}{9} - \frac{\cos(\psi)}{3} + \frac{\cos(\theta)^2}{9} - \cos(\theta) \right)}{\sigma_1}$$

where

$$\sigma_1 = (2 \sigma_2^3 - 9 \sigma_2^2 + 27 \cos(\psi)^2 \cos(\theta)^2 + 27 \cos(\psi)^2 \sin(\theta)^2 + 27 \cos(\theta)^2 \sin(\psi)^2 + 27 \sin(\psi)^2 \sin(\theta)^2 + 3 \sqrt{3} \sqrt{\cos(\psi) + 1} \sqrt{\cos(\theta) + 1})^{1/3}$$

$$\sigma_2 = \cos(\psi) + \cos(\theta) + \sigma_3$$

$$\sigma_3 = \cos(\psi) \cos(\theta)$$

lambda2 =

0

lambda3 =

0

qat =

$$\begin{pmatrix} \frac{\sigma_2 \sigma_1}{2} \\ -\frac{\sin(\psi) \sin(\theta)}{2 \sigma_2 \sigma_1} \\ \frac{\sin(\theta) \sigma_2}{2 \sigma_1} \\ \frac{\sin(\psi) \sigma_1}{2 \sigma_2} \end{pmatrix}$$

where

$$\sigma_1 = \sqrt{\cos(\theta) + 1}$$

$$\sigma_2 = \sqrt{\cos(\psi) + 1}$$

axis =

$$\begin{pmatrix} -\frac{\sin(\psi) \sin(\theta)}{\sqrt{\cos(\psi) + 1} \sqrt{\cos(\theta) + 1} \sigma_1} \\ \frac{\sin(\theta) \sqrt{\cos(\psi) + 1}}{\sqrt{\cos(\theta) + 1} \sigma_1} \\ \frac{\sin(\psi) \sqrt{\cos(\theta) + 1}}{\sqrt{\cos(\psi) + 1} \sigma_1} \end{pmatrix}$$

where

$$\sigma_1 = \sqrt{3 - \cos(\theta) - \cos(\psi) \cos(\theta) - \cos(\psi)}$$

angle =

$$2 \arccos\left(\frac{\sqrt{\cos(\psi) + 1} \sqrt{\cos(\theta) + 1}}{2}\right)$$

```
function Tmatrix = TmatrixFunc(psi , theta)
```

```
%TMATRIXFUNC
```

```
%    TMATRIX = TMATRIXFUNC(PSI , THETA)
```

```
%    This function was generated by the Symbolic Math Toolbox version 8.4.
```

```
%    26-Dec-2019 13:53:54
```

```
t2 = cos(psi);
```

```
t3 = cos(theta);
```

```
t4 = sin(psi);
```

```
t5 = sin(theta);
```

```
Tmatrix = reshape([t2.*t3 , t3.*t4 , -t5 , -t4 , t2 , 0.0 , t2.*t5 , t4.*t5 , t3] , [3 , 3]);
```

```
upAxis = 3×2
```

```
    0    50
    0     0
    0     0
```

```
lowAxis = 3×2
```

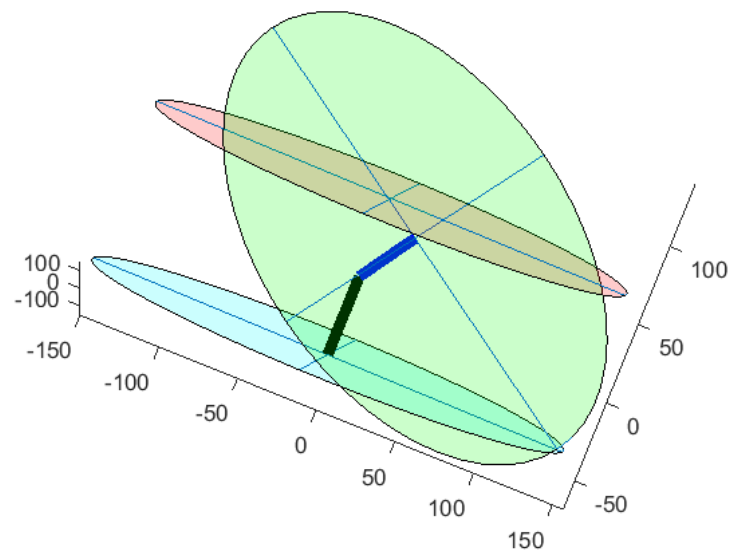
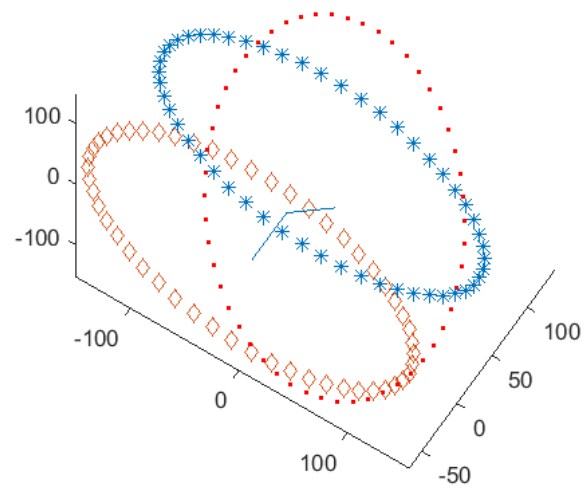
```
    0   -50
    0     0
    0     0
```

```
upAxisT = 3×2
```

```
0    36.0912
```

```
0    20.8373
```

```
0   -27.6267
```



```
% 測試旋轉一圈的結果，
```

```

nSeg=10;nCables=3; holes=nSeg*nCables;
rhos=[0:holes-1]*2*pi/holes;
maxAngle=pi/5;testPoints=20;
testPsi=[0:testPoints]*maxAngle/testPoints;
testTheta=sqrt(maxAngle^2-testPsi.^2);
testResults=zeros(length(testPsi),length(rhos));
for i=1:length(testPsi)
    testResults(i,:)=CableLength(dd,testPsi(i),rjr,rhos,
testTheta(i));
end

```

Not enough input arguments.

Error in rjr (line 14)

[m,n] = size(A);

```
plot(testResults')
```

拉動時各結繩長的計算方式

每個萬向關節包含兩個旋轉角度： θ_i ， θ_{i+1} 。這兩個角度是由三條，通過端板上三個以 120° 等角度分布在圓周半徑 r 的孔洞的纜繩所控制。使用三條纜繩，而非兩條的原因是：只有兩條纜繩無法維持足夠的張力。

產生機械臂末端工具的空間位置是由所有的關節旋轉角度 (θ_i, θ_{i+1}) 決定。透過逆向運動學計算的個別關節角度，可以推算出控制每個關節的三條纜繩所需改變的長度。

如以角度皆為 0 度的纜繩長度為基準點，則角度 (θ_i, θ_{i+1}) 的關係是由：

```
cableLength
```

因為每個萬向節有兩個自由度（ ψ, θ ），計算以每個萬向節為主。每經過一個萬向節，萬向節與連結體的接點會有 90 度的旋轉角度。但是纜繩通過每個連結體的相對角度位置也會有 90 度的改變。

如果纜繩以基座的角度位置 δ_i 為標記，經過一個萬向節後，計算纜繩的計算方法為

根據我們座標系統的定義，

Define a circle points on the Y-Z plane

```
function points=YZCircle(C, r, numPoints)
if mod(numPoints, 4)~=0
    numPoints=ceil(numPoints/4)*4;
end
increment=360.0/numPoints;
angle=0.0;
points=zeros(3, numPoints);
for i=1:numPoints
    points(:, i)=[C(1);C(2)+r*cosd(angle);C(3)+r*sind(angle)];
    angle=angle+increment;
end
end

function plot3circle(x, y, z, color)

numb=length(x); % number of data points
plot3(x, y, z)
fill3(x, y, z, color, 'FaceAlpha', 0.2)
%line(up(1, :), up(2, :), up(3, :))
%plot3(x, y, z)
line([x(1);x(end)], [y(1);y(end)], [z(1);z(end)]);
%line([up(1, 1);up(1, end)], [up(2, 1);up(2, end)], [up(3, 1);up(3,
end)])
half=ceil(numb/2);
quater = ceil(numb/4);
onethird=ceil(3*numb/4);
line([x(half);x(end)], [y(half);y(end)], [z(half);z(end)]);
line([x(quater);x(onethird)], [y(quater);y(onethird)],
[z(quater);z(onethird)]);

%line([up(1, 26);up(1, end)], [up(2, 26);up(2, end)], [up(3, 26);up(3,
end)])
%line([up(1, 13);up(1, 39)], [up(2, 13);up(2, 39)], [up(3, 13);up(3,
39)])
end
```

```

function [vector , angle]=EulerAngle2AxisAngle(yaw , pitch , roll)
c1=cos(yaw/2);
c2=cos(pitch/2);
c3=cos(roll/2);
s1=sin(yaw/2);
s2=sin(pitch/2);
s3=sin(roll/2);
angle = 2 * acos(c1*c2*c3-s1*s2*s3);
vector=[s1*s2*c3+c1*c2*s3;s1*c2*c3-c1*s2*s3;c1*s2*c3-s1*c2*s3];
vector=vector/sqrt(dot(vector , vector));
end

function q=RotationMatrix2Quaternion(RM)

[r , c]=size(RM);
q=sym(zeros(1 , 4));
if ( r ~= 3 | c~=3)
    error('Rotation matrix must be 3X3 matrix!');
else
    qr=sqrt(1+trace(RM))/2;
    qi=(RM(3 , 2)-RM(2 , 3))/4/qr;
    qj=(RM(1 , 3)-RM(3 , 1))/4/qr;
    qk=(RM(2 , 1)-RM(1 , 2))/4/qr;
    q=[qr , qi , qj , qk];
end
end

function [axis , angle]=Quaternion2AxisAngle(q)
axis=sym(zeros(1 , 3));angle=sym(0);
if length(q)~=4
    error('Input is not a quaternion');
else
    angle = 2 * acos(q(1));
    sin_qw=sqrt(1-q(1)*q(1));
    axis=[q(2)/sin_qw , q(3)/sin_qw , q(4)/sin_qw];
end
end
end

```