中文摘要

多連桿會依連桿數與關節還有尺寸,輸出點的移動路徑會有不同。設計時，會依照我們想要的輸出點以特殊的路徑移動來設計，但過程中常因為尺寸設計不當而得不到我們想要的結果,此時就必須要有可以將輸出點路徑方程式反推成尺寸合成的工具(套件)，如果將路徑設計好，就能將尺寸定下來，下一步就是運動模擬，能夠在虛擬環境中模擬運動成功的話，實體成型成功的機會就會很高，不必浪費實體資源。

我們將從多連桿的輸出點路線的運算到如何從 Pyslvs 套件,設計具行走及跨越障礙的行走機構，對多連桿做機構模擬、尺寸合成，再到 onshape 建立模型匯入 V-Rep 組立與設定，進一步加入程式進行模擬，以此完成一個完整的設計架構。

關鍵字: 多連桿、Jansen linkage、行走機構模擬

ABSTRACT

Multiple links will depend on the number of links and the size of the joints, and the path of each point will be different.When designing, we will design according to a certain point we want to move with a special path,But in the process, because of the improper size design, we do not get the results we want.At this point, there must be a tool that can push the point path equation back into size synthesis,At this point, there must be a tool (kit) that can push the point path equation back into size synthesis. If the path is designed well, the dimensions can be set. The next step is the motion simulation, which can simulate the success of the movement in a virtual environment. The opportunity for successful solid forming will be high, without wasting physical resources.

We will start with multi-link point-path calculations, how to design a walking mechanism with walking and crossing obstacles from the Pyslvs kit, do mechanism simulations on multiple links, and sizing, and then build model into onshape to import V-Rep. Settings, further adding programs to complete a complete design architecture.

Keywords: Multi-link, Jansen linkage, walking mechanism simulation

致 謝

專題研究期間，感謝嚴家銘教授在專題研究方向的指導，並在輔導過程中培養我們解決問題以及應對環境的能力，使我們在研究過程中受益良多。也感謝研究室張元學長，提供有關機構尺寸合成演算法的開發套件並成功整合進研究中，也感謝研究室林祐生學長，處理問題時提供許多有用的建議等等。

目錄

圖表目錄

第一章 簡介

1.1 研究目的

一開始我想設計並實作一個能走動的行走機構,但是在不知道多連桿的設計方法，以及毫無理論的設計，引發諸多問題，尺寸大小造成的組裝問題以及行走時不實際的運動等等...，這些問題讓我知道盲目的設計不但浪費時間，設計出來的東西還會出現很多問題。

也因為這樣，這次我注重於虛擬方面，先理解多連桿店路徑的推導，然後嘗試使用 Pyslvs、Onshape、V-Rep 等套件，建構一個完整的設計流程。

1.2 預期結果

在設計的部分，我會分析行走機構的行走距離以及提腳高度，嘗試讓每一次都是不同尺寸,這樣它就有不同的提腳高度或者是行走距離，最後依良好的提腳高度來選擇它的尺寸。

挑選好的尺寸會進入到 onshape 裡繪製，因為主要以模擬運動為主，在外觀上會選擇簡易清楚明白，繪製好組合完進入 Vrep 模擬。

最後的 Vrep 模擬，是為了能讓我們知道設計好的行走機構如果再地上走會是什麼樣子?走起來的情況如何?

為了呼應前面提腳高度分析，會嘗試加入路障並使之穿越，編寫程式讓它辨識障礙物的高度是否在跨越高度的門檻內,如果會辨識、在門檻內直接跨越成功;門檻外則避開成功，這本次分析才會有意義。

第二章 研究方法與理論

2.1 自由度

連桿的自由度，乃是相對於機架或固定桿而言，確定每一根連桿位置所需的最少**獨立參數(independent parameter)**之數目，對一根不受任何拘束的空間桿件而言，其具有六個自由度，即包含三個座標方向平移和三個座標方向旋轉。若一桿件只在一特定平面上運動，則該桿件具有三個自由度－2 個座標方向平移以及該平面上之旋轉，但當兩連桿以上連接時，因為受限於對偶接頭形狀，往往造成連桿間的自由度損失。如果連桿相互經由對偶接頭連接，而形成一個封閉的的幾何圖形，則稱此運動鏈為**閉迴路運動鏈**。反之，則稱為**開迴路運動鍊**。通常對平面封閉運動鏈的自由度計算，可用**古柏方程式。**

$$F = 3(L - 1) - 2P_1 - P_2$$

F=3(L-1)-2P1-P2

F:平面運動鏈的自由度

L:連桿數總和

P1:具有一個自由度之對偶接頭數目

P2:具有兩個自由度之對偶街頭數目

F=1，則為拘束運動鏈。

F>1，為無拘束運動鏈。

F=0，為呆鏈，也就是架構或結構。

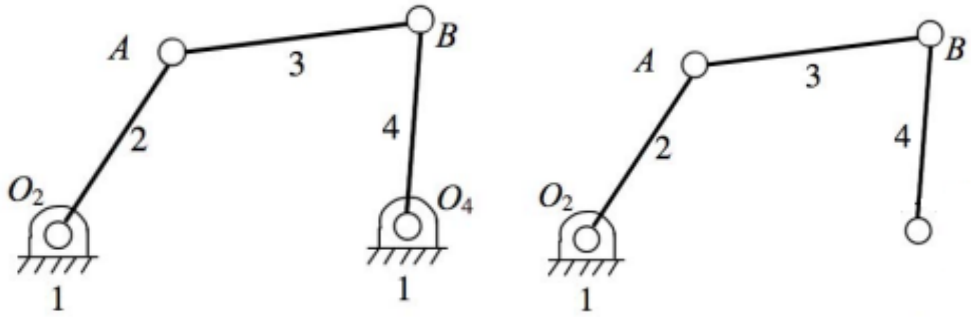


圖2.1　閉迴路、開迴路運動鏈

依前面的古柏方程式，我們可以計算 Jansen Linkage 的自由度

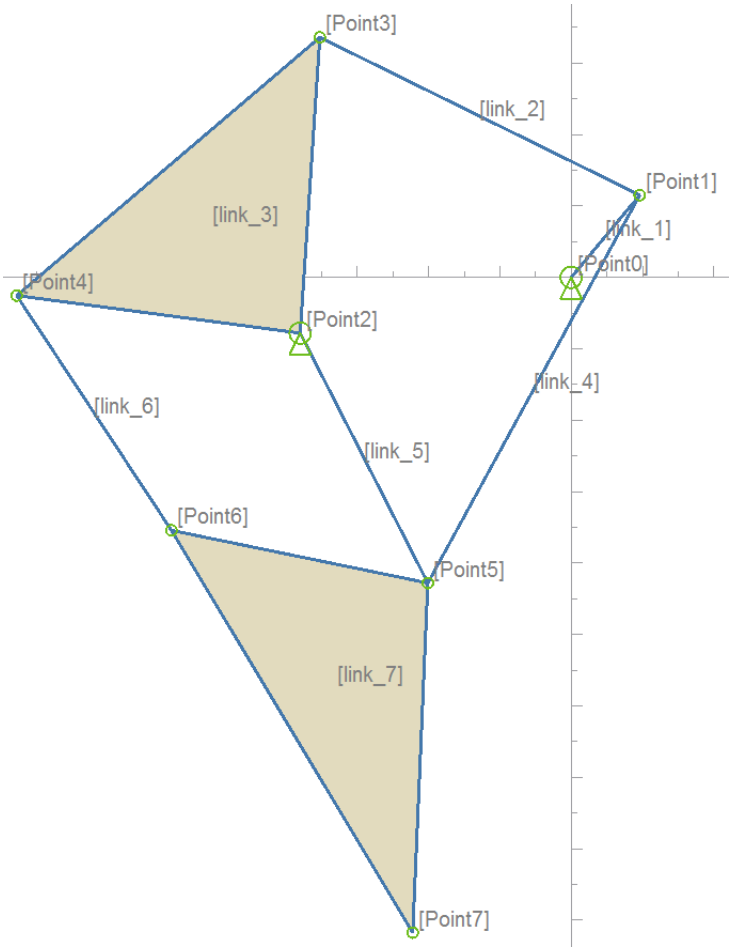


圖2.2　Jansen Linkage

L=8，P1=10，P2=0

F=3(8-1)-210-0=1，為拘束運動鏈

2.2 Triangluar analysis (三角形表示法)

我們在設計多連桿是為了在輸出點找到符合我們要求的運動路徑，但如何求得路徑，必須要先知道路徑怎麼來的。

我們其實可以將多連桿分解成許多三角形，此時我們可以套用一個叫三角形表示法的方法，如下列:

圖2.3　PLAP

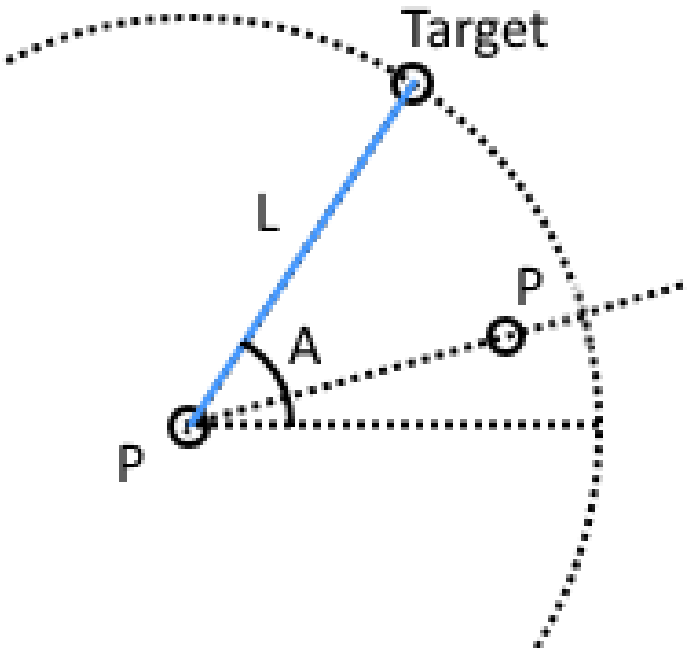
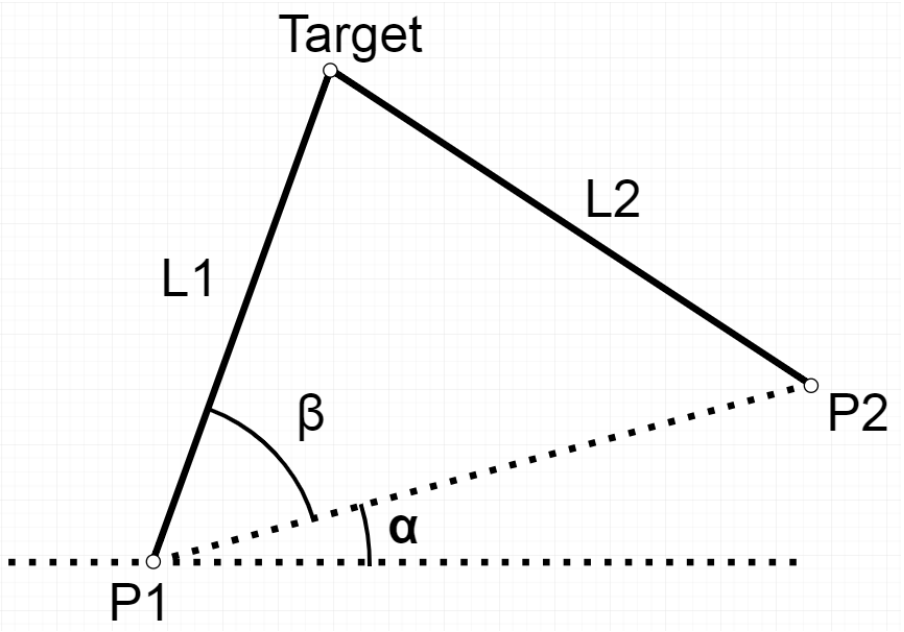


圖2.4　PLLP



PLAP:

透過兩個（或一個）已知點、基準點至目標點距離，及其（與水平線）的夾角，可以求得目標點座標。

$$\alpha = 0{\sim}360$$

$$\mathrm{A} = \tan^{-1}\left(\frac{P_2y - P_1y}{P_2x - P_1x}\right) + \alpha$$

$$T_x = P_1x + Lcos(A)$$

$$T_y = P_1y + Lsin(A)$$

由上列公式可以知道，這是很單純的直角三角形座標求法，加上固定點 P1 的 XY 位置，我們可以求得 Target(x,y)，通常輸入端(motor)會套入這方法運算。

PLLP:

透過兩個已知點、兩基準點至目標點距離，可以求得目標點座標。

$$\mathrm{d} = \sqrt{(P_2x - P_1x)^2 + (P_2y - P_1y)^2}$$

$$\alpha = tan^{-1}(\frac{P_2y - P_1y}{P_2x - P_1x})$$

餘弦定理求角

$$\beta = cos^{-1}(\frac{L_1^2 + d^2 - L_2^2}{2 \cdot L_1 \cdot d})$$

$$T_x = P_1x + Lcos(\alpha + \beta)$$

$$T_y = P_1y + Lsin(\alpha + \beta)$$

若 T 點的位置在 P1P2 之下，兩個角度會變成相減，在程式中可以藉由參數反轉之(順相法)。

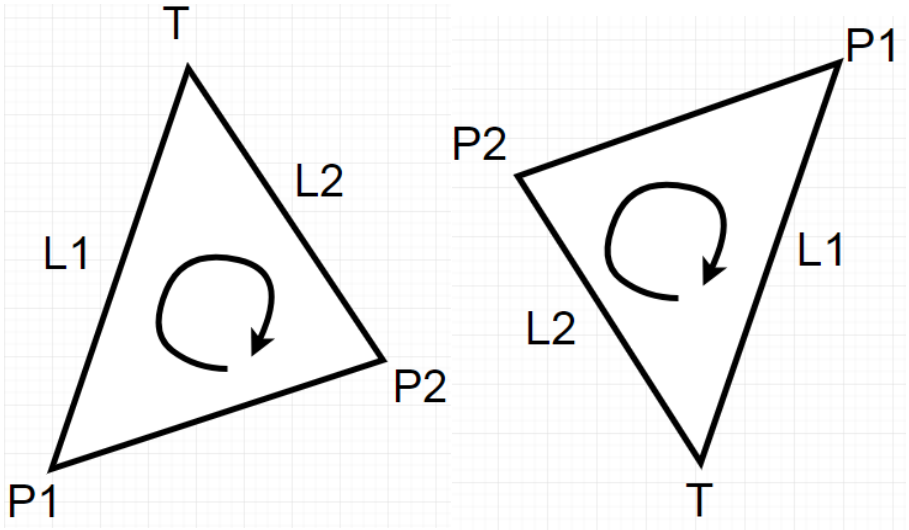


圖 2.5　順向法

第三章 工具介紹

3.1 Pyslvs

一個基於 GUI 的工具用於解決二維鏈接問題

平面連接模擬：使用 Python 軟件包（SWIG）從 Solvespace 獲取內核。

機械合成：

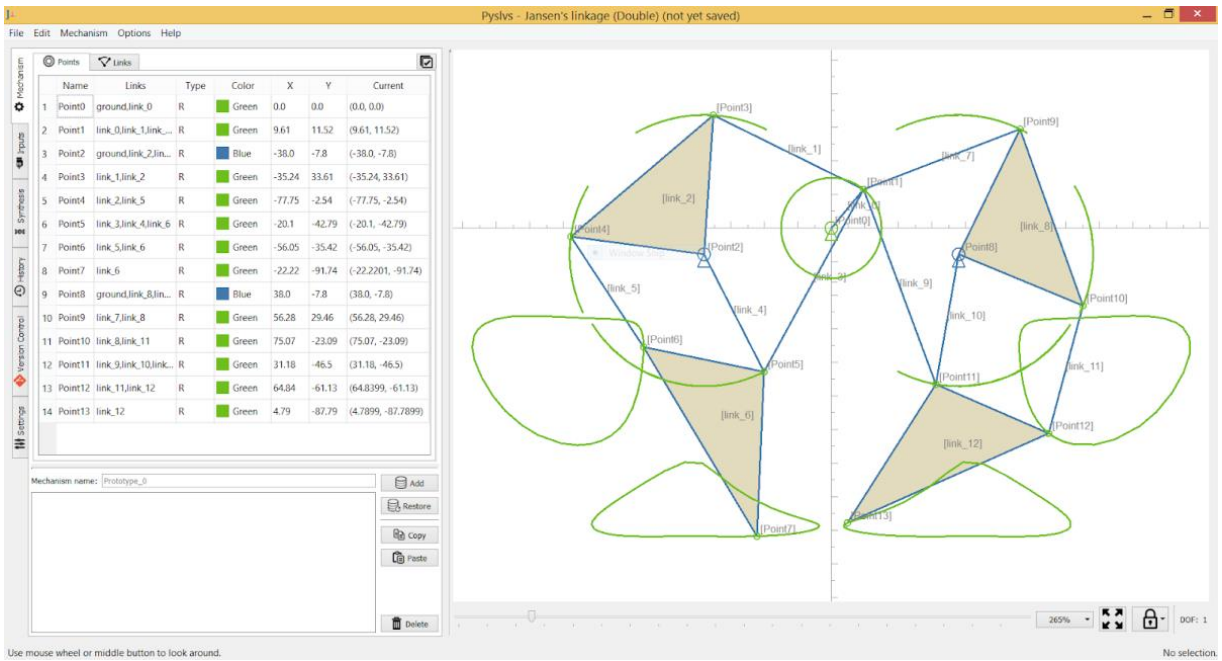
數字和類型綜合：Cython 算法用於找出機制的結構可能性。

Dimensional Synthesis：三種 Cython 算法 API 的內核。

兼容 Python 3.5，PyQt 5.7（用於 PyQtChart）及以上版本。

跨平台開發：Ubuntu 和 Windows（64 位）。

圖 3.1 Window 介面



3.2Onshape

為線上 CAD 設計軟件。

Onshape 的現代 CAD 系統使工程師能夠專注於完成最佳工作。與舊的 CAD 系統不同，Onshape 將建模工具和設計數據管理集成到安全的雲工作空間中，可在任何設備上訪問，從不丟失數據，並消除設計僵局。

圖 3.2　Onshape

1.先進的參數化三維建模

用您需要的一切重新構建 CAD，而不會遇到麻煩。真正的自上而下設計、配置、標準內容庫，多部分建模和上下文編輯。

2.隨時隨地訪問

沒有下載，安裝，許可證密鑰，服務包或兼容性問題 - 每個人都有最新版本。

3.零檔案零麻煩。

通過在一個安全的雲工作區中存儲和共享您的CAD數據，消除登記/退房麻煩並浪費舊產品資料管理(PDM)系統的IT開銷。

4.前所未有的控制和安全性

舊CAD文件中的知識產權很容易失去。使用安全的雲工作空間完全控制您的數據。控制和監視訪問權限，並查看誰更改了什麼時間。

3.3V-Rep

具有集成開發環境的機器人模擬器 V-REP 基於分佈式控制架構：每個對象/模型都可以通過嵌入腳本，插件，ROS 或 BlueZero 節點，遠程 API 客戶端或自定義解。這使得 V-REP 非常靈活，是多機器人應用的理想選擇。控制器可以用 C / C ++，Python，Java，Lua，Matlab 或 Octave 編寫。

V-REP 用於快速算法開發，工廠自動化模擬，快速原型和驗證，機器人相關教育，遠程監控，安全雙重檢查等。可以找到功能概述

圖 3-3　V-REP

以下是 V-REP 的一些應用：

- 工廠自動化系統的仿真
- 遠程監控
- 硬件控制
- 快速原型設計和驗證
- 安全監控
- 快速算法開發
- 機器人相關教育

第四章 虛擬控制

4.1 架構流程

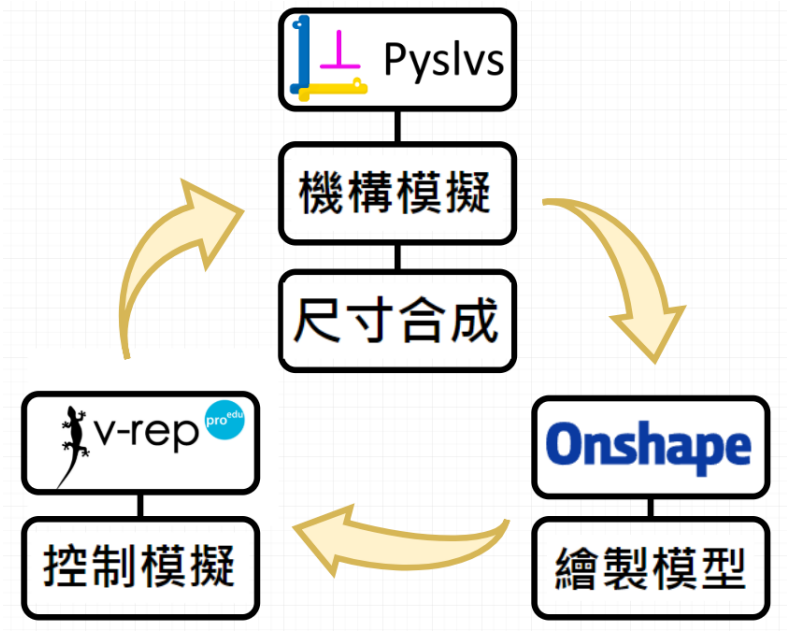


圖 4.1 虛擬控制架構

從架構圖可以知道，這是一種可以循環下去的設計概念，從 Pyslvs:機構模擬→尺寸合成開始，將設計多連桿的路徑方程式解出，這部分是在理解前面三角形表示法的前提下，才能知道 Pyslvs 是用什麼方法得出移動路線。然後再到 Onshape:繪製模型，在前面已經得到已知的重要尺寸，便可以設計具行走功能的行走機構。最後 V-rep:控制模擬，設計出的模型必須匯入到 V-rep 進行所謂的動態模擬，觀察行走步態的分析，進一步在編寫程式控制行走機構的馬達，觀察跨越障礙以及沿著牆壁的模擬，若覺得在尺寸的方面可以再做調整，則回到原本的第一步來做改進。

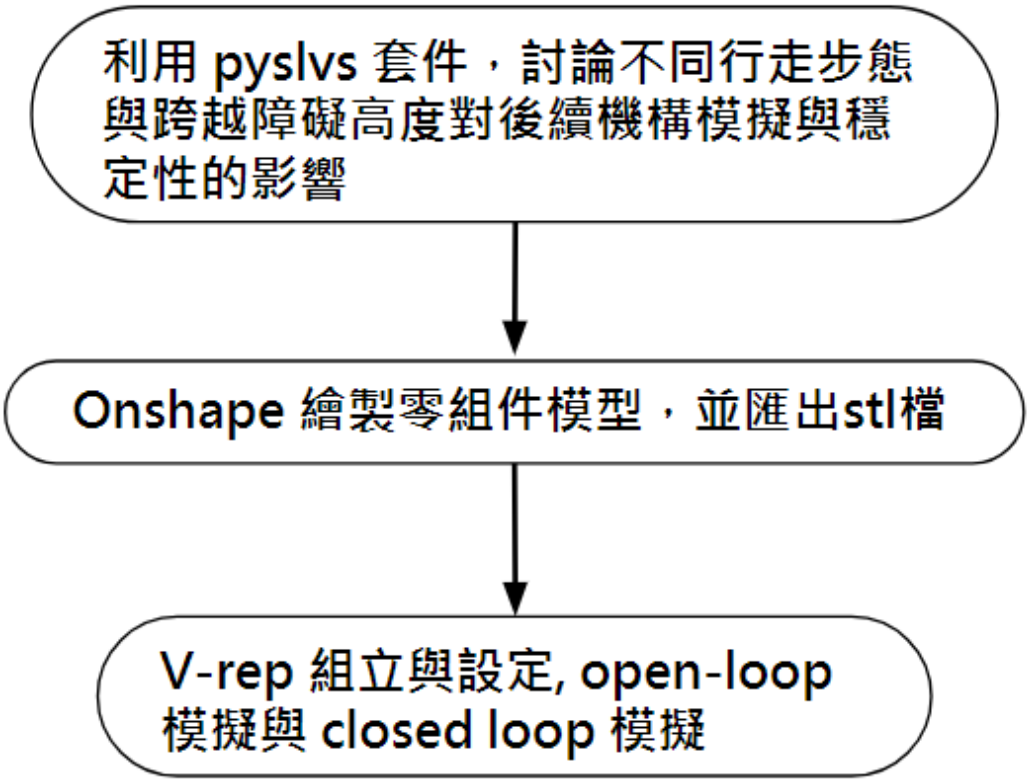


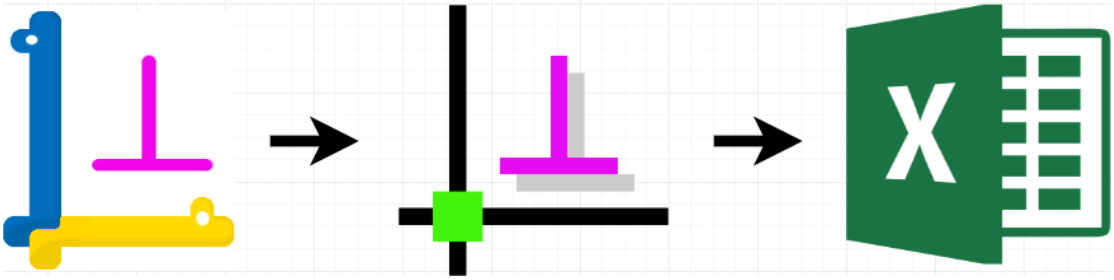
圖 4.2　流程圖

4.2 尺寸合成設計及數據表

我們可以用 Pyslvs 來解多連桿點路徑方程式，或者可以在畫布上繪出不規則曲線及限制固定點範圍與連桿數，則 Pyslvs 會在這些限制條件內反推點位置與連桿的尺寸長度，得到的尺寸長度，便可匯入 Solvespace 得到尺寸合成，並且可以在 Solvespace 裡得到點移動路線的數據檔(csv)，進入 Excel 裡匯出成圖表，可以比較分析在不同尺寸下不同路徑的多連桿，分析它的提腳高度及跨步長度。

基本流程如下：

圖 4.3 Pyslvs 流程圖



1.啟動 pyslvs-18.4.0.mscv1900-amd64

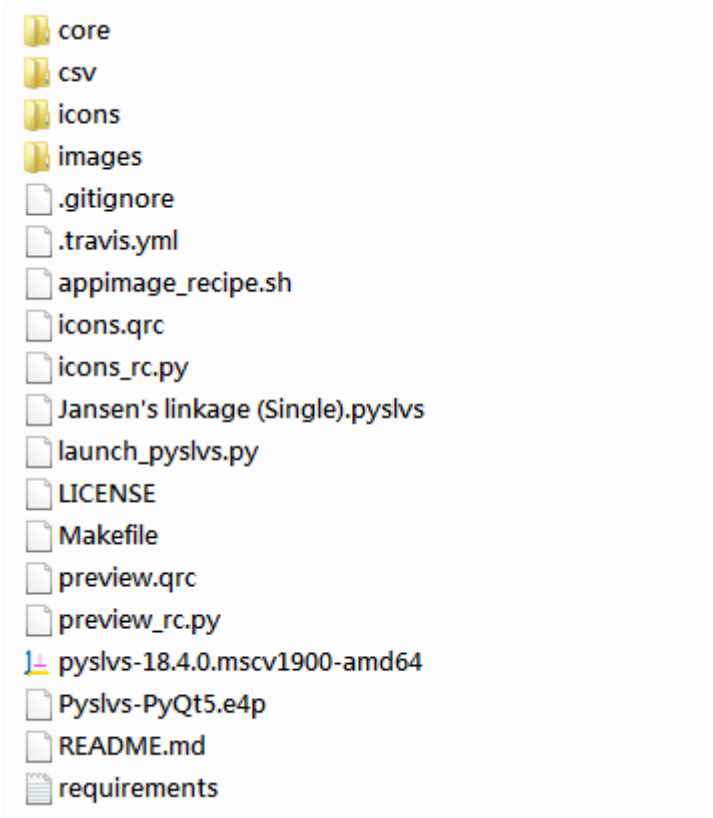
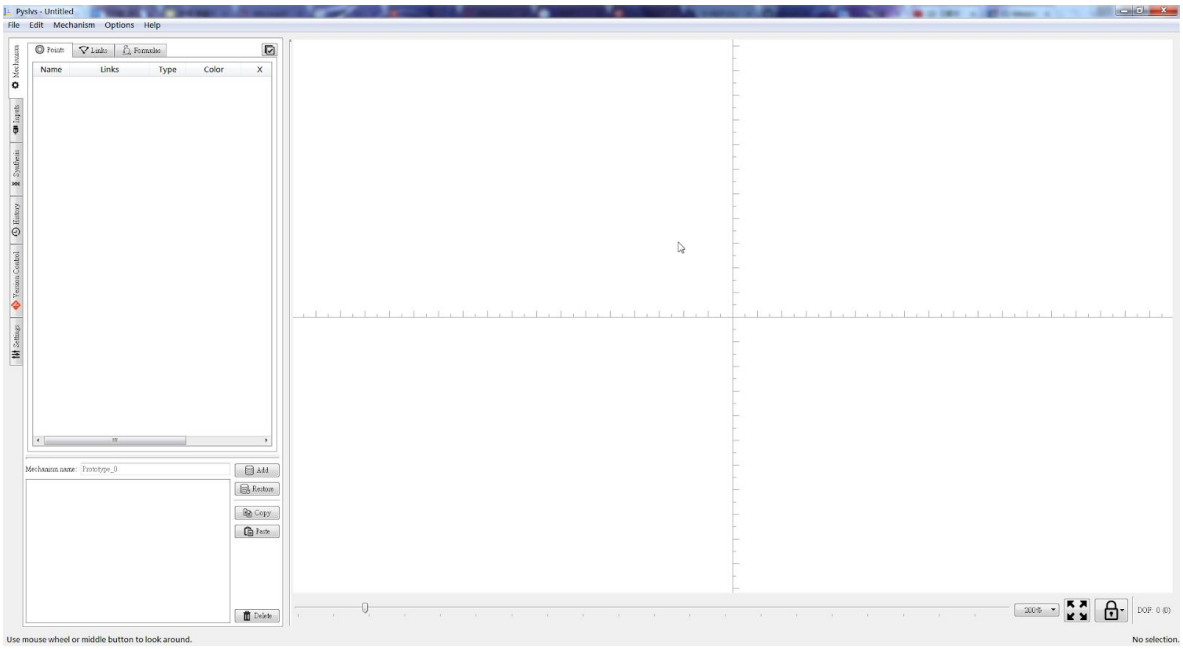
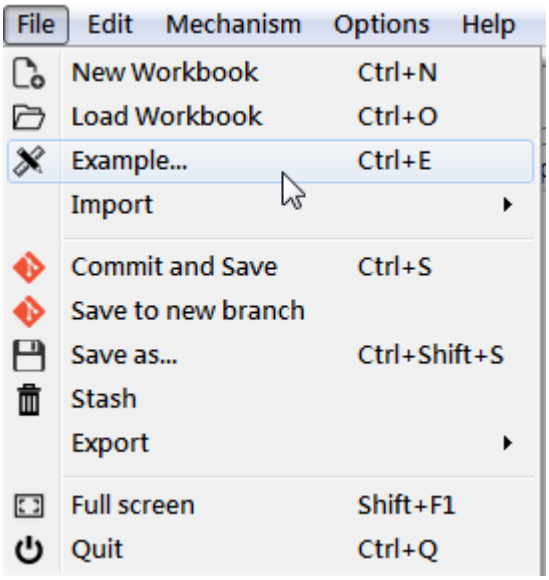
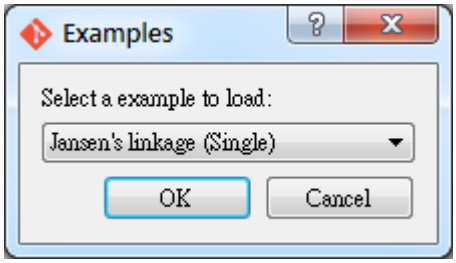


圖 4.4　啟動與介面



2.左擊 File→Example，選擇 Jansen's linkage(Single)





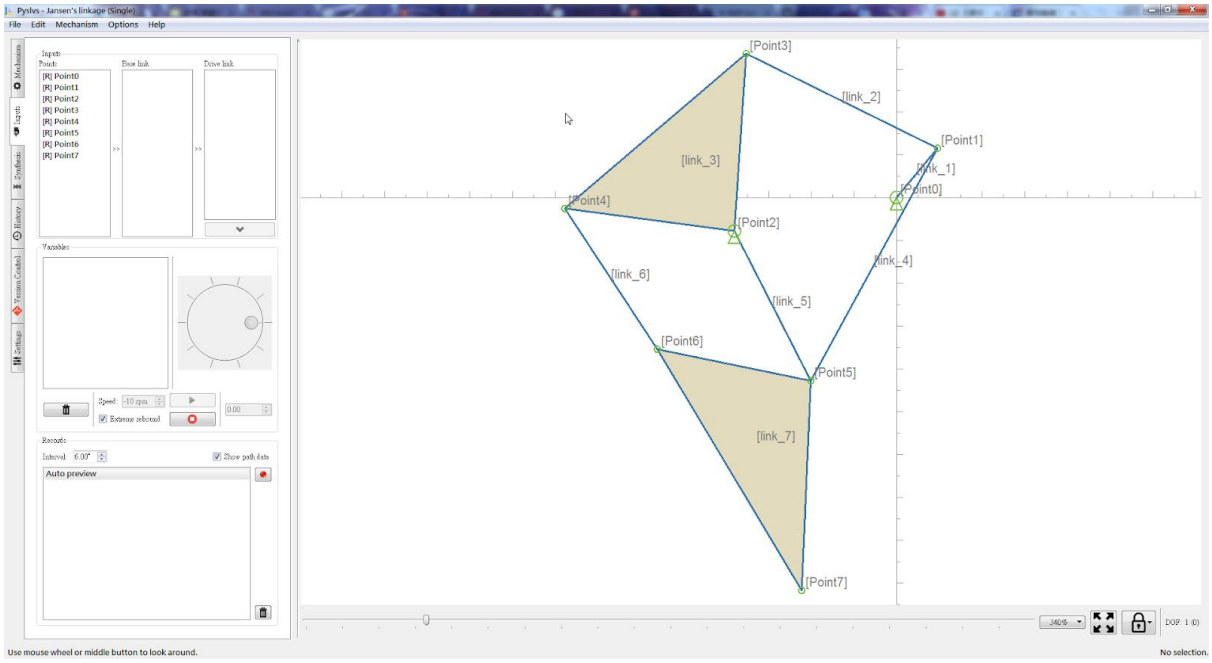
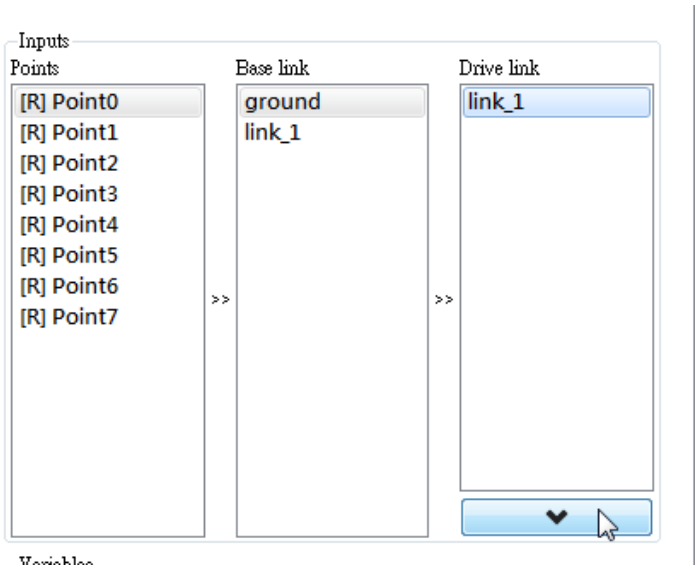
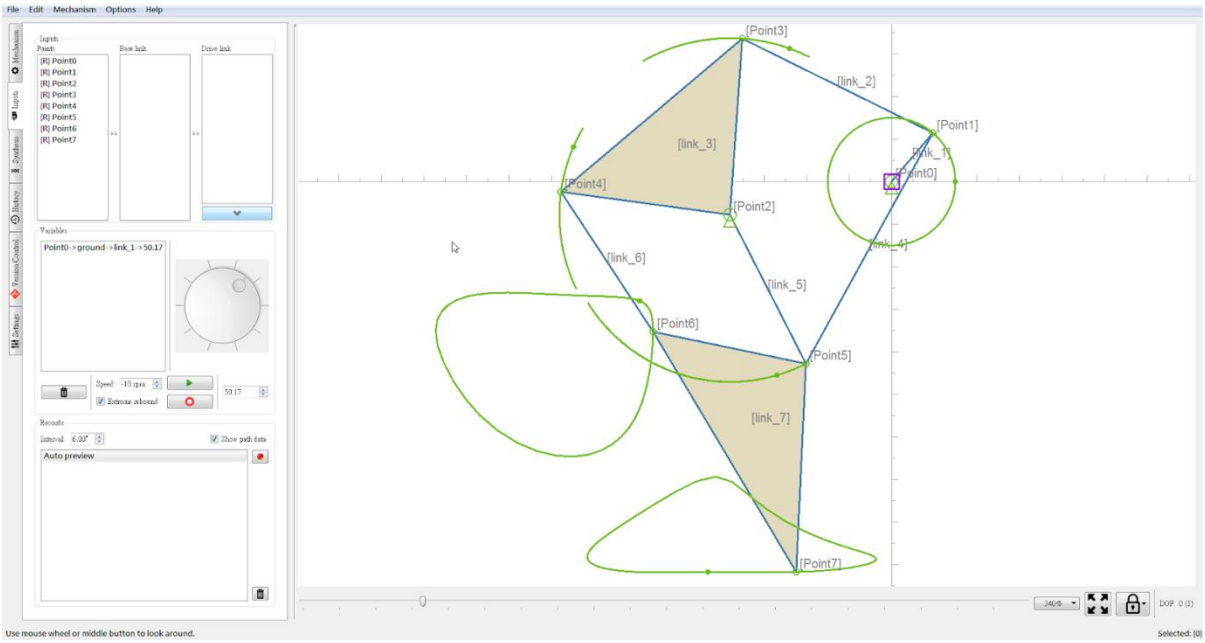


圖 4.5　選擇 Jansen's linkage(Single)

3.左邊的操作視窗可以選擇某點作為 motor 旋轉，點選[R] Point0 →ground→link_1 並確定

圖 4.6　motor 設定介面，與路線顯示





4.左視窗滾輪為手動轉動 motor 功能，也可以設定轉速

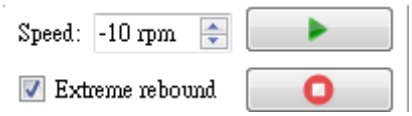


圖 4.7　轉速設定

5.旋轉右下角的鑰匙則可以改變連桿長，或是尋找三角形表示法中 PLLP 的另一解。

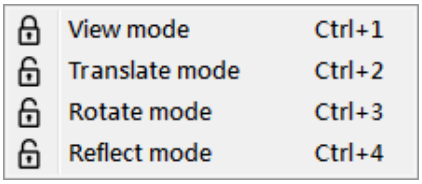
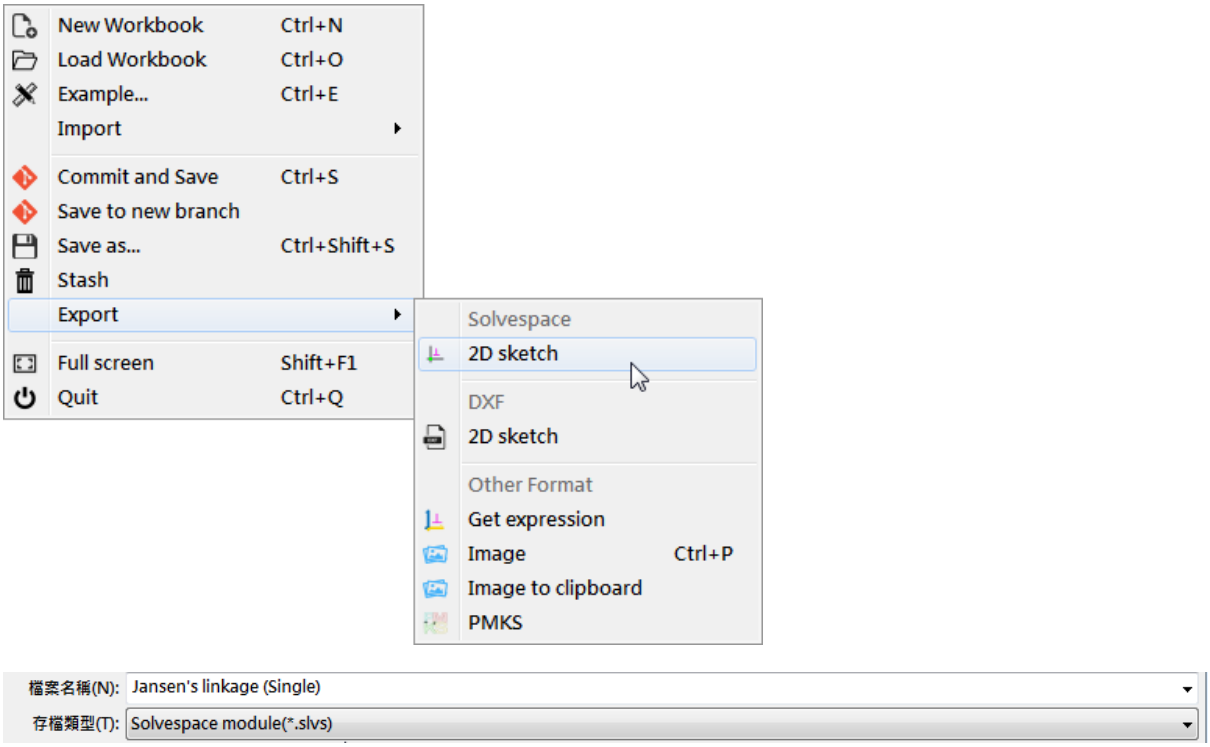


圖 4.8　鎖定設定選項

6.修改連桿長，觀察路線的變化，選擇適當路徑並存成 Slvs 檔

圖 4.9　File→Export→2D sketch



7.啟動 SolveSpace

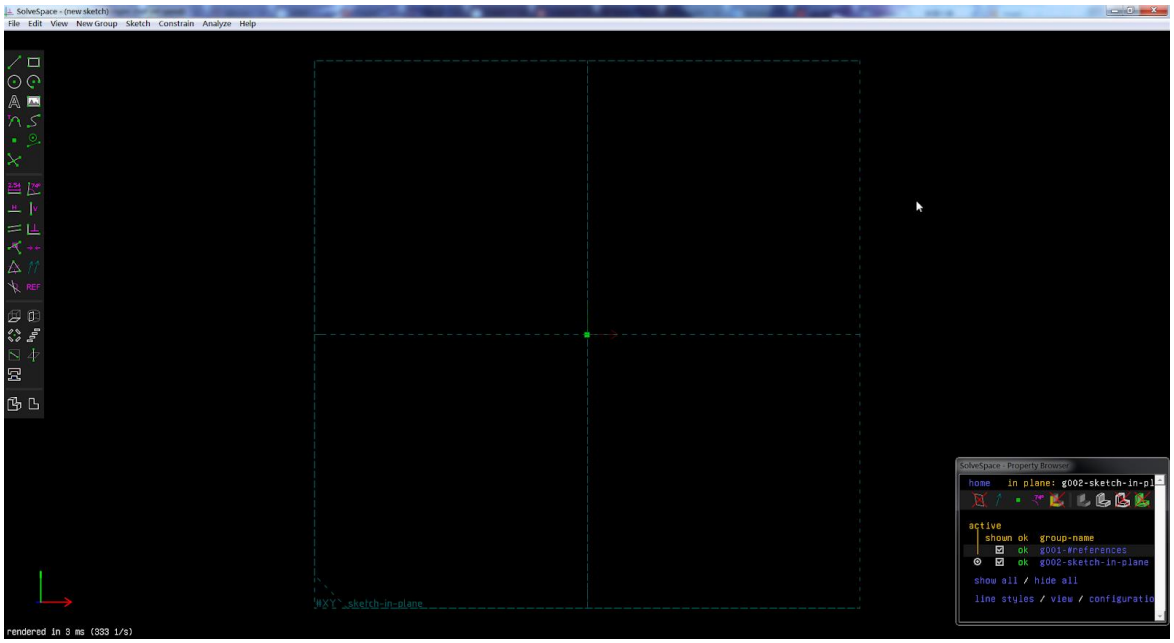


圖 4.10 SolveSpace 介面

8.開啟 slvs 檔

9.開啟後便可得到個連桿的尺寸，用於之後的 Onshape 繪製模型設計用途

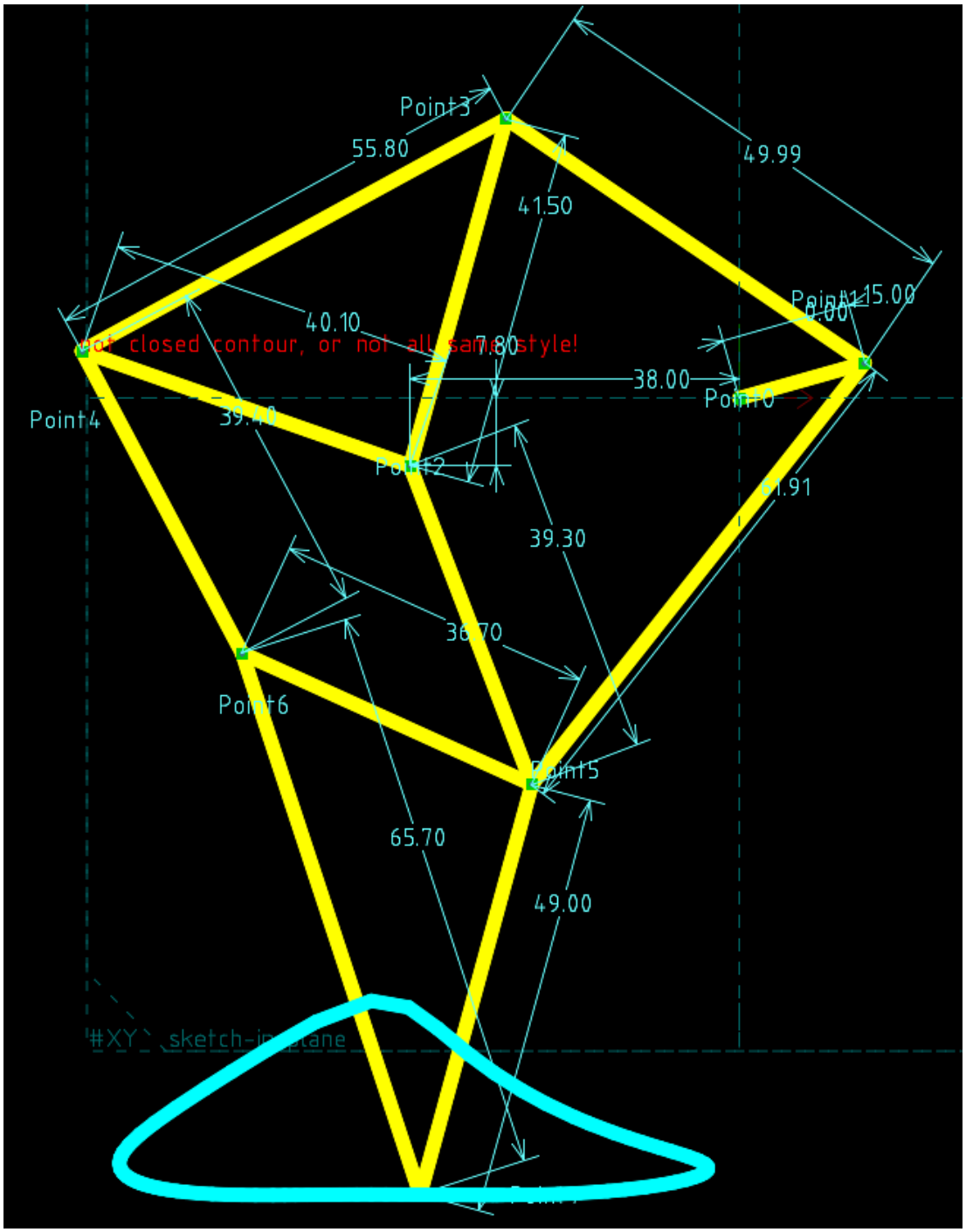


圖 4.11 多連桿尺寸

10.選擇 Point7 點，點選功能 Analyze→Trace Point，顯示點路線移動

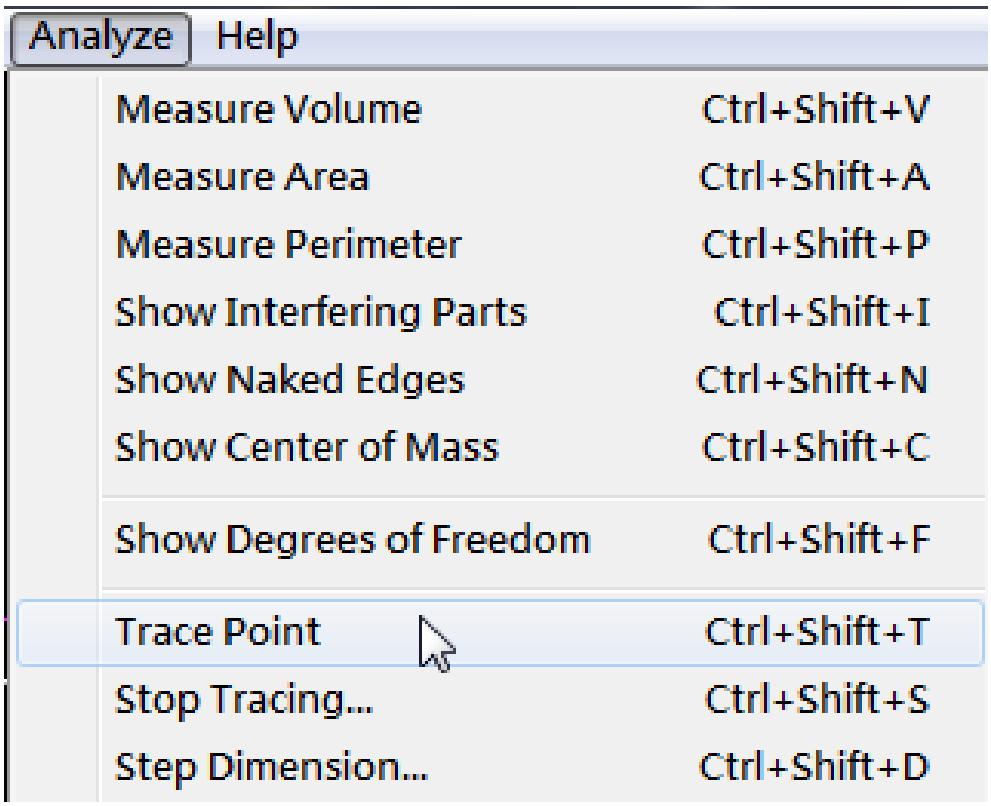


圖 4.12 Trace Point

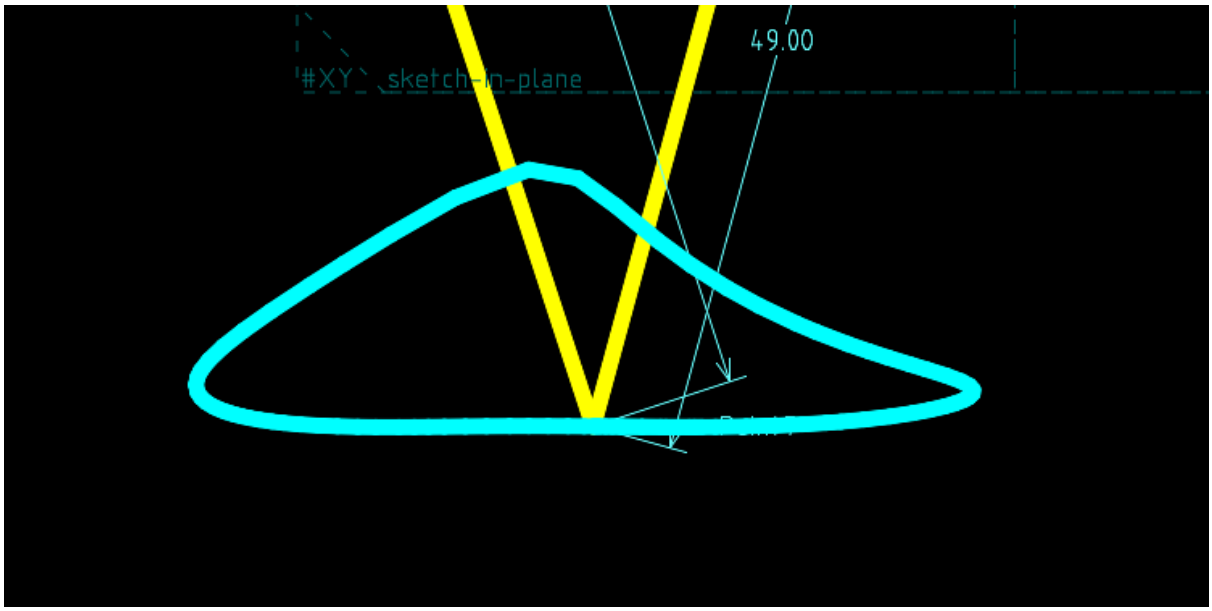


圖 4.13 點路線

11.點選功能 Analyze→Stop Tracing，存成 CSV 數據檔

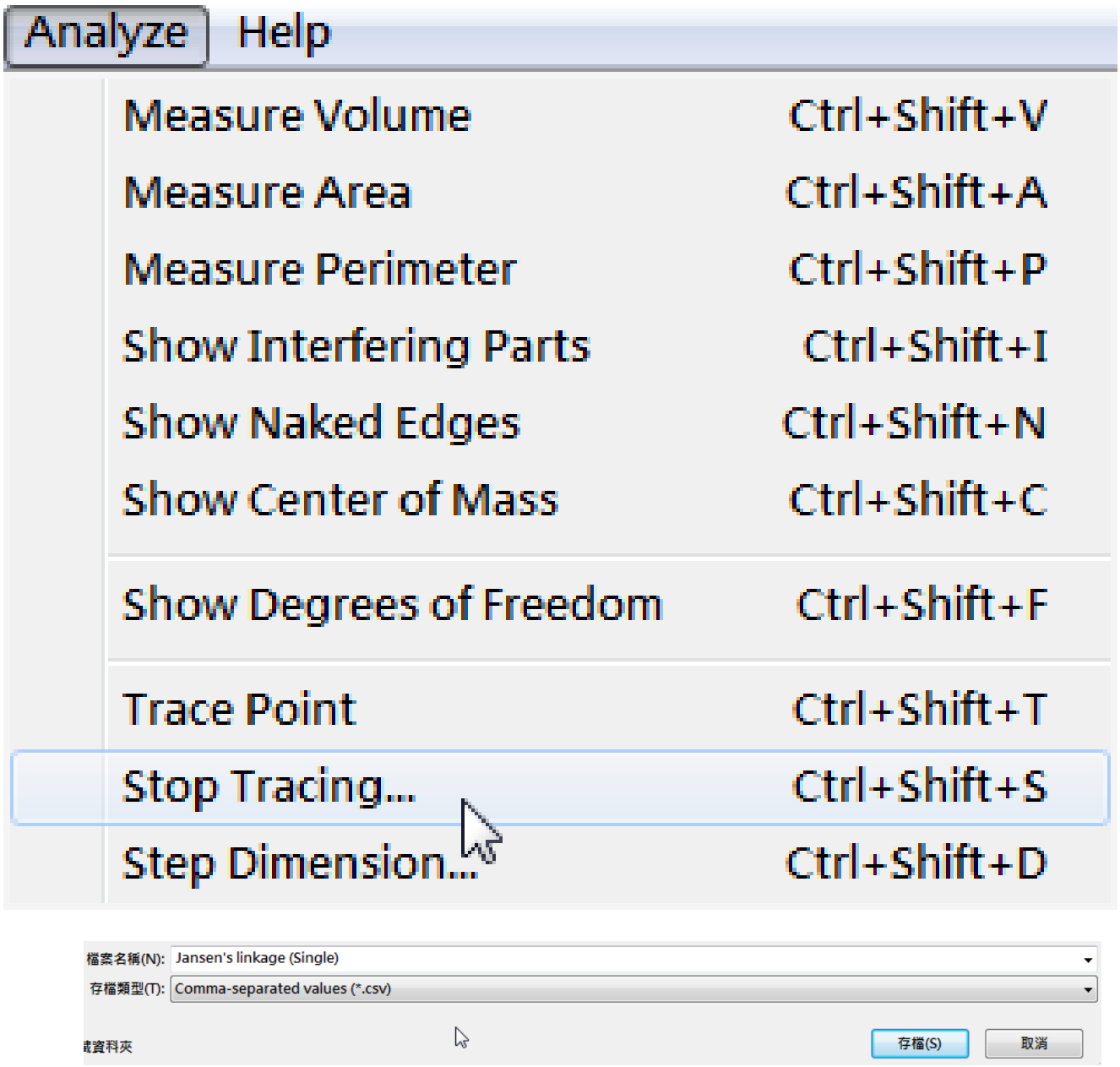
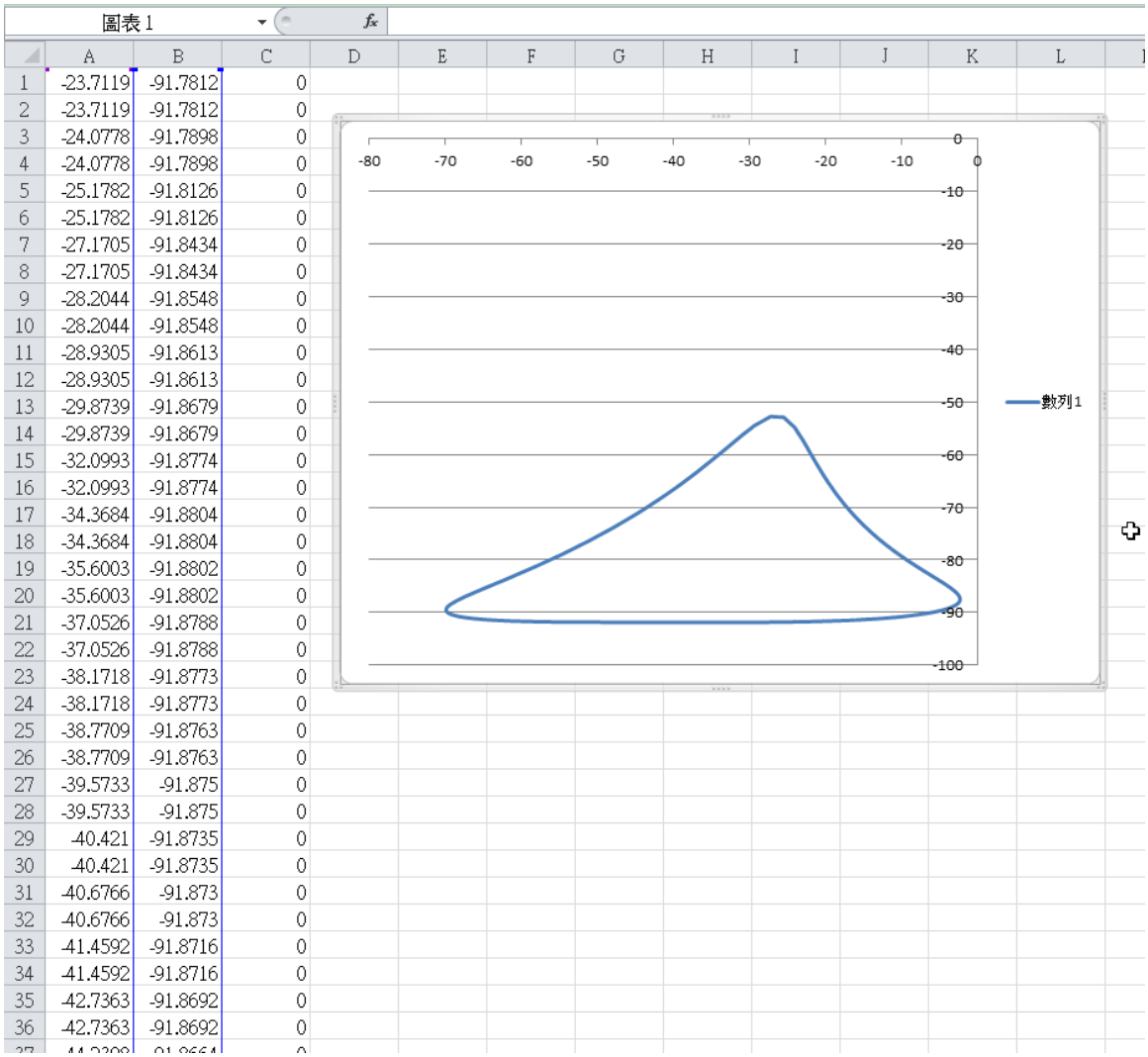


圖 4.14 Stop Tracing

12.啟動 Excel，開啟 CSV 檔案，並插入散佈圖



	A	B	C
1	-23.7119	-91.7812	0
2	-23.7119	-91.7812	0
3	-24.0778	-91.7898	0
4	-24.0778	-91.7898	0
5	-25.1782	-91.8126	0
6	-25.1782	-91.8126	0
7	-27.1705	-91.8434	0
8	-27.1705	-91.8434	0
9	-28.2044	-91.8548	0
10	-28.2044	-91.8548	0
11	-28.9305	-91.8613	0
12	-28.9305	-91.8613	0
13	-29.8739	-91.8679	0
14	-29.8739	-91.8679	0
15	-32.0993	-91.8774	0
16	-32.0993	-91.8774	0
17	-34.3684	-91.8804	0
18	-34.3684	-91.8804	0
19	-35.6003	-91.8802	0
20	-35.6003	-91.8802	0
21	-37.0526	-91.8788	0
22	-37.0526	-91.8788	0
23	-38.1718	-91.8773	0
24	-38.1718	-91.8773	0
25	-38.7709	-91.8763	0
26	-38.7709	-91.8763	0
27	-39.5733	-91.875	0
28	-39.5733	-91.875	0
29	-40.421	-91.8735	0
30	-40.421	-91.8735	0
31	-40.6766	-91.873	0
32	-40.6766	-91.873	0
33	-41.4592	-91.8716	0
34	-41.4592	-91.8716	0
35	-42.7363	-91.8692	0
36	-42.7363	-91.8692	0

圖 4.15 Excel 圖表

4.3 繪製模型

模型的部分，使用 Onshape:線上 CAD 來繪製。

1.進入 https://www.onshape.com/，開啟新的文件

圖 4.16 建立文件

2.建立草圖，繪製設計好的連桿尺寸

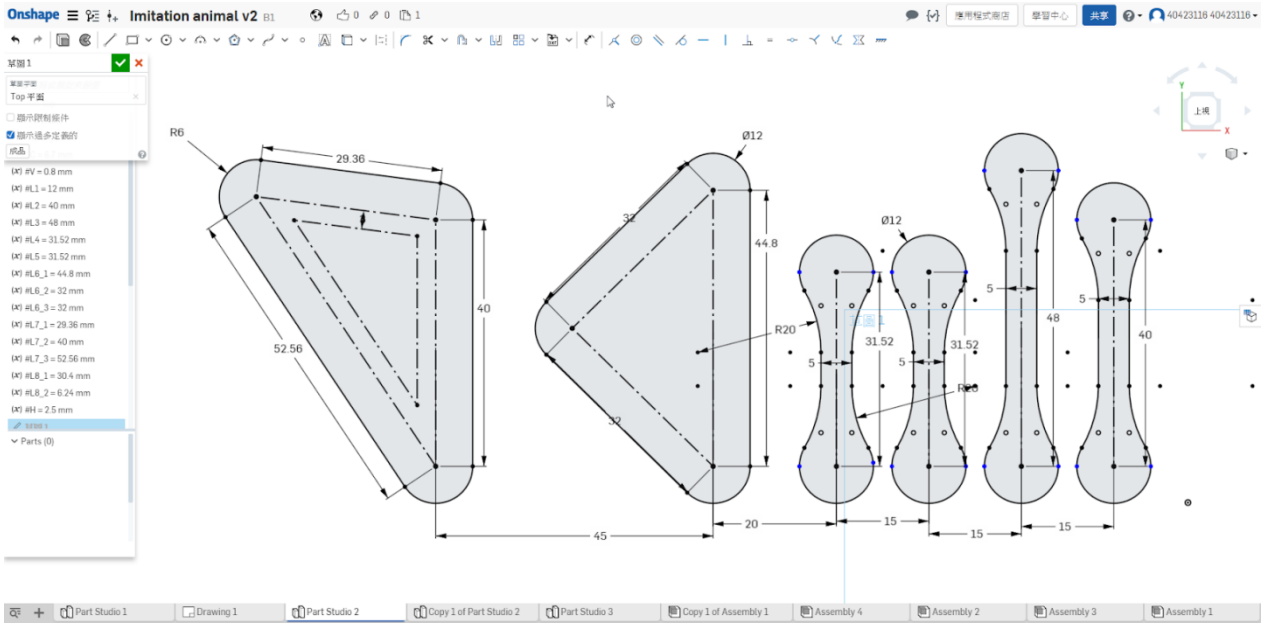


圖 4.17 繪製多連桿尺寸

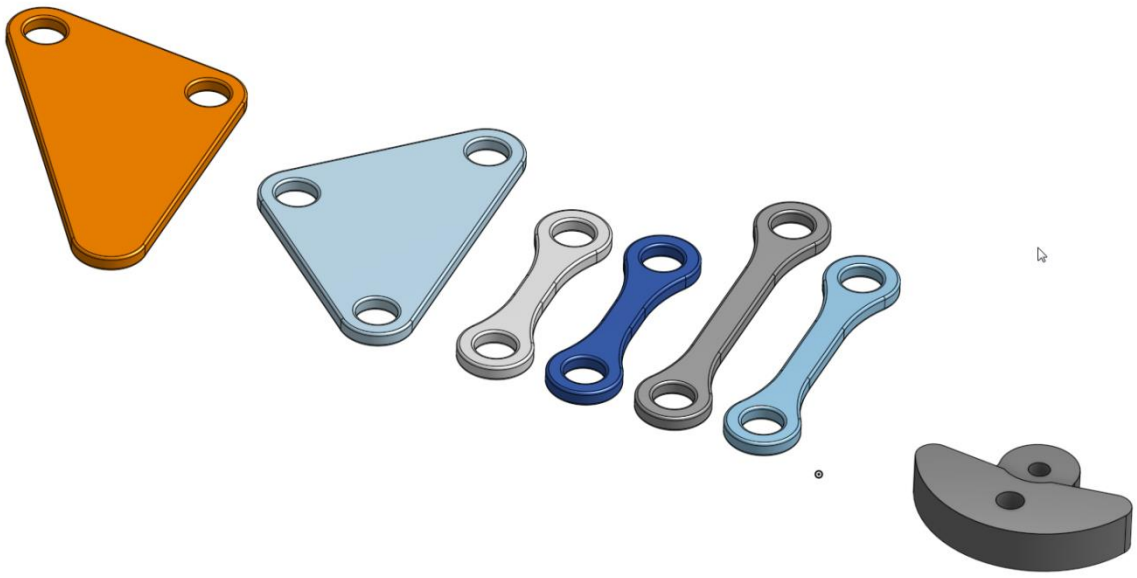


圖 4.18 擠出

3.其他模型如 Base 為自定義設計。

4.組裝

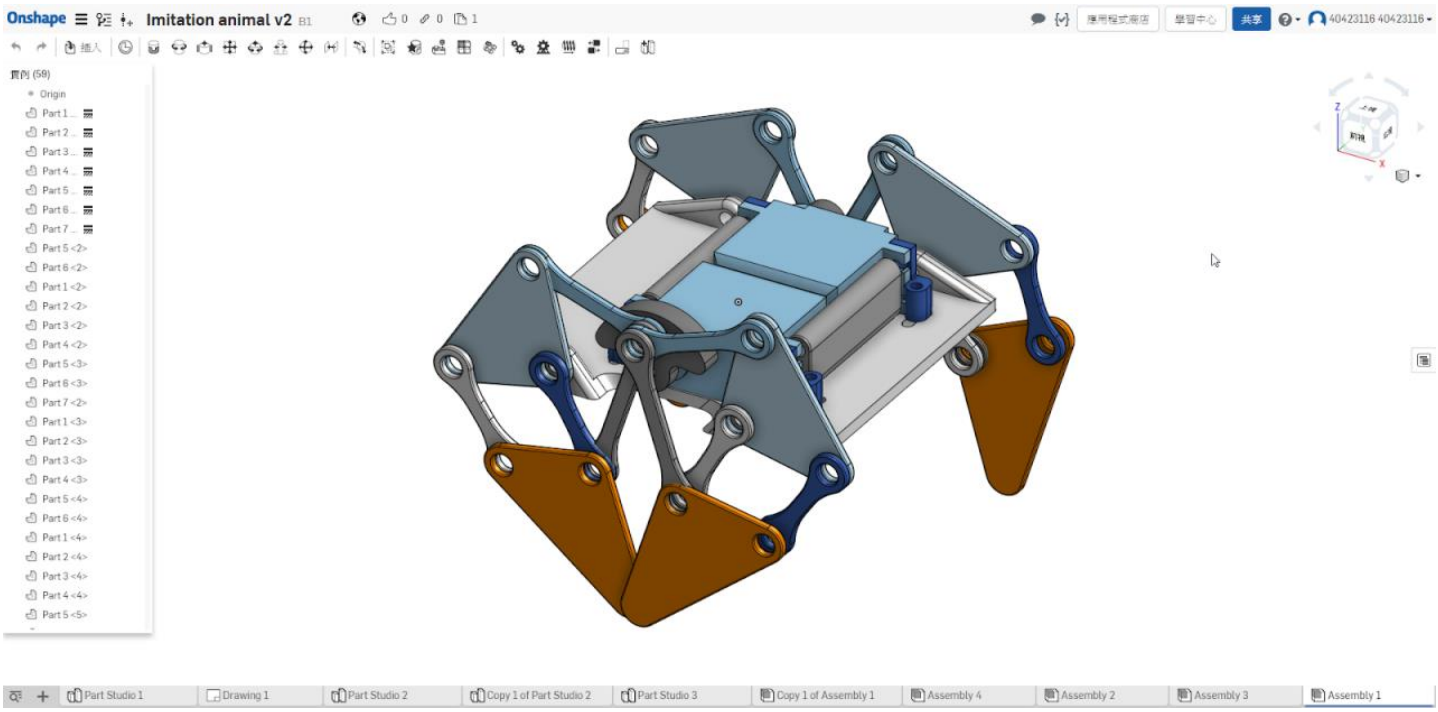


圖 4.19 組合圖

5.組合圖存成 stl 檔

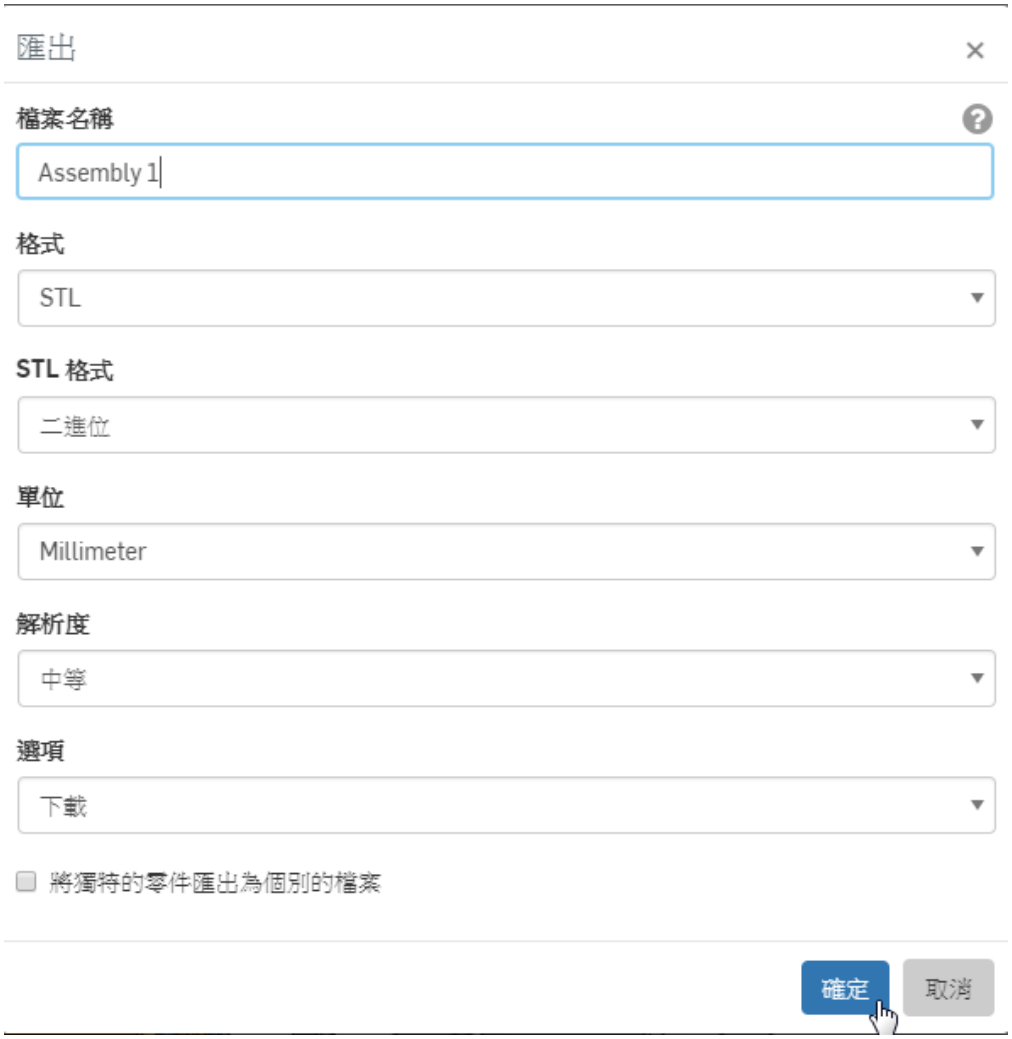


圖 4.20 Stl 檔

4.4 V-REP 組立與設定

使用 V-rep 模擬多連桿行走機構

1.匯入 syl 檔

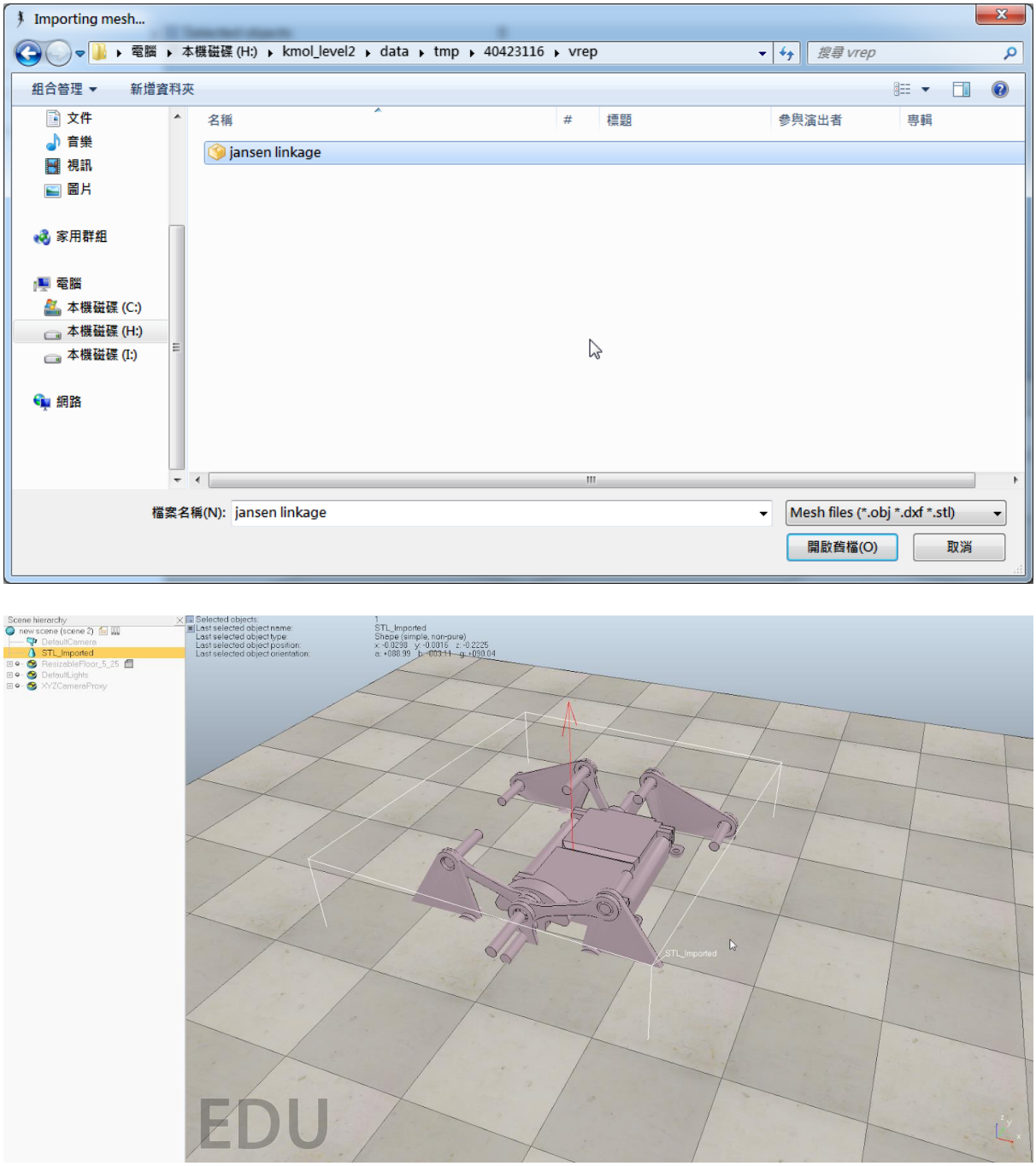


圖 4.21 匯入 stl 檔

2.可以將零件 xyz 軸移動或旋轉，點選功能 Object/item shift，點選 Z 軸上下移動到平台上。

圖 4.22 xyz 軸移動、旋轉

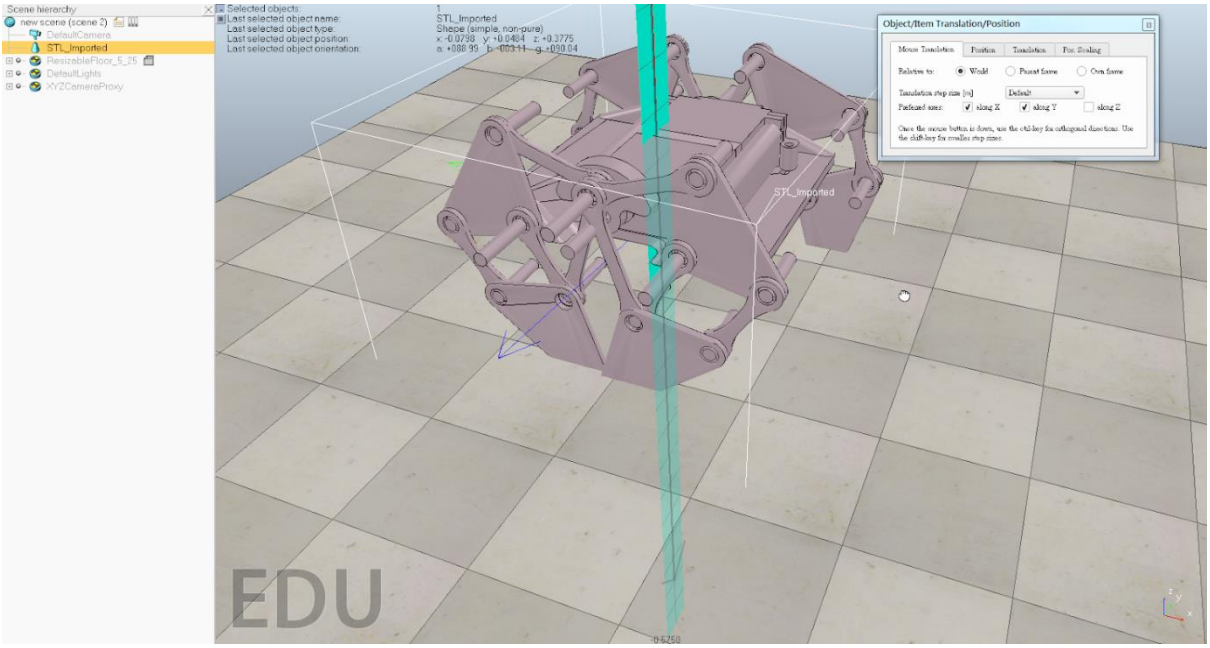


圖 4.23 上下移動

3. 將組合件分解成數個零件。右鍵並點選 Edit → Grouping/MergingDivide selected shapes

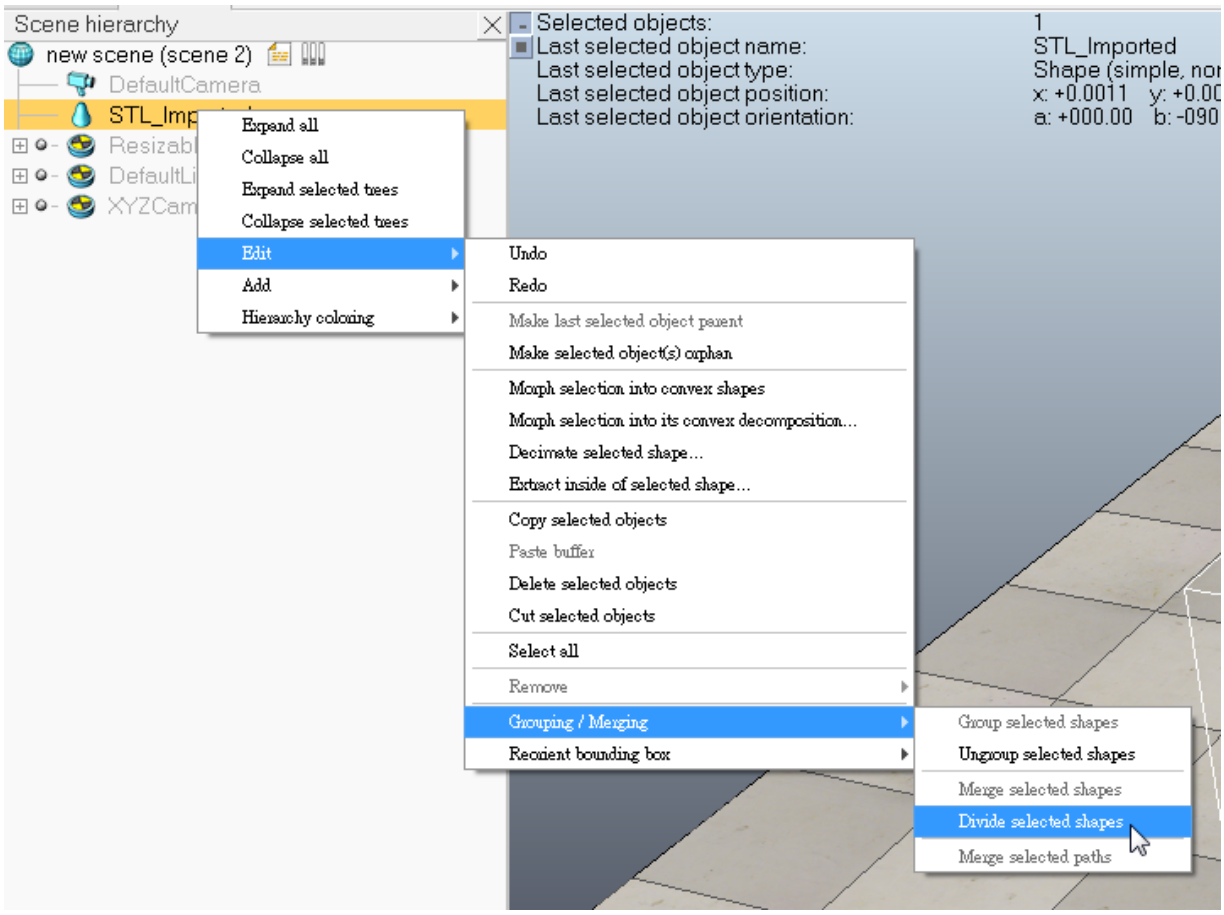


圖 4.24 分解組合圖

4.建立 Joint。右鍵 add→joint→Revolute，設定大小(自定義)。

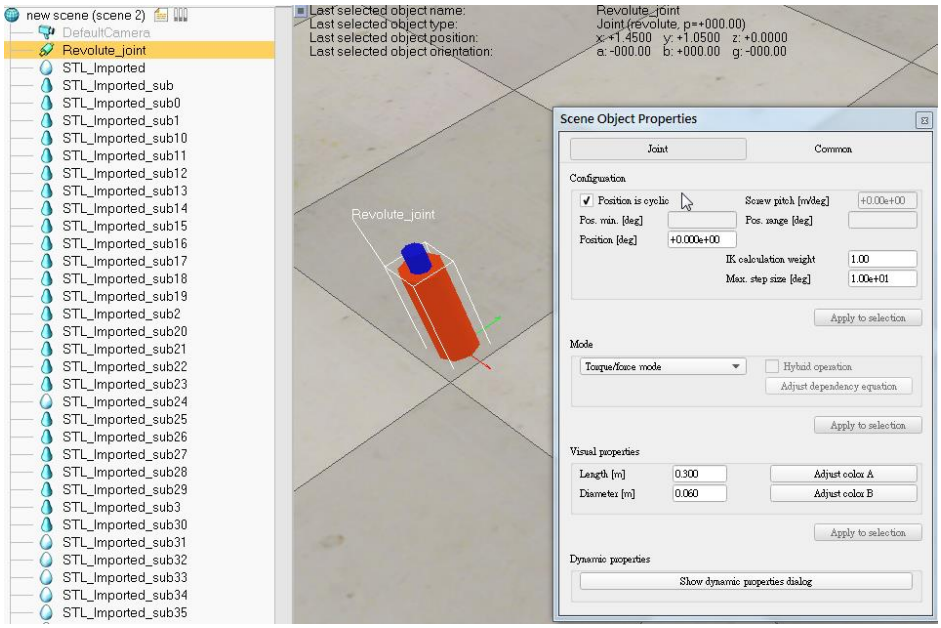


圖 4.25 設定 Joint 大小

5.為方便作業，將其他無關零件隱藏，可按住 Shift 框住欲隱藏的零件，將 Camera visiblity layers 的下排第一個打勾，並套用全部(Apply)

圖 4.26 顯示設定

6.將 Joint，與在 Onshape 建立的圓柱重疊座標，已經重疊旋轉軸。

(1).點選功能 Object/item shift，先選擇 Joint 按住 ctrl 點選另一個圓柱，在 Postition 選項點選 Apply to selection

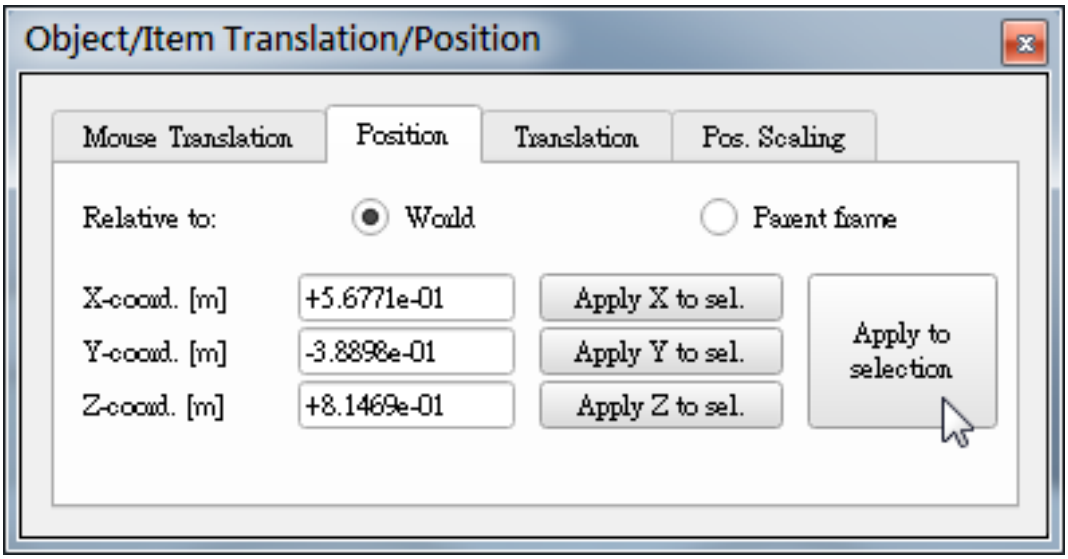


圖 4-27 重疊座標

(2).點選功能 Object/item rotate，先選擇 Joint 按住 ctrl 點選另一個圓柱，在 Orientation 選項點選 Apply to selection

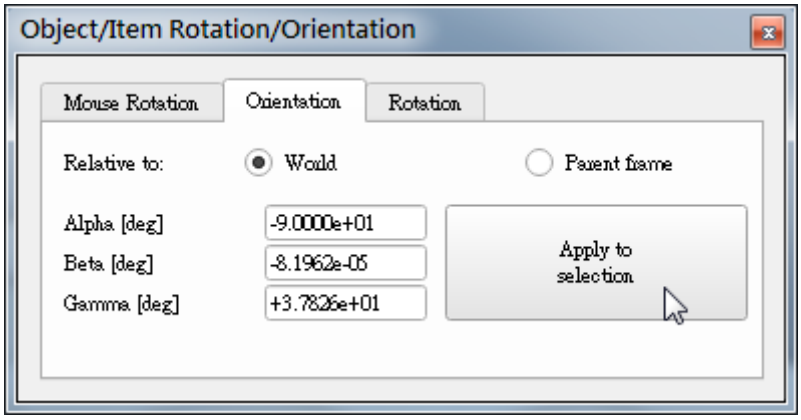


圖 4.28 重疊轉軸

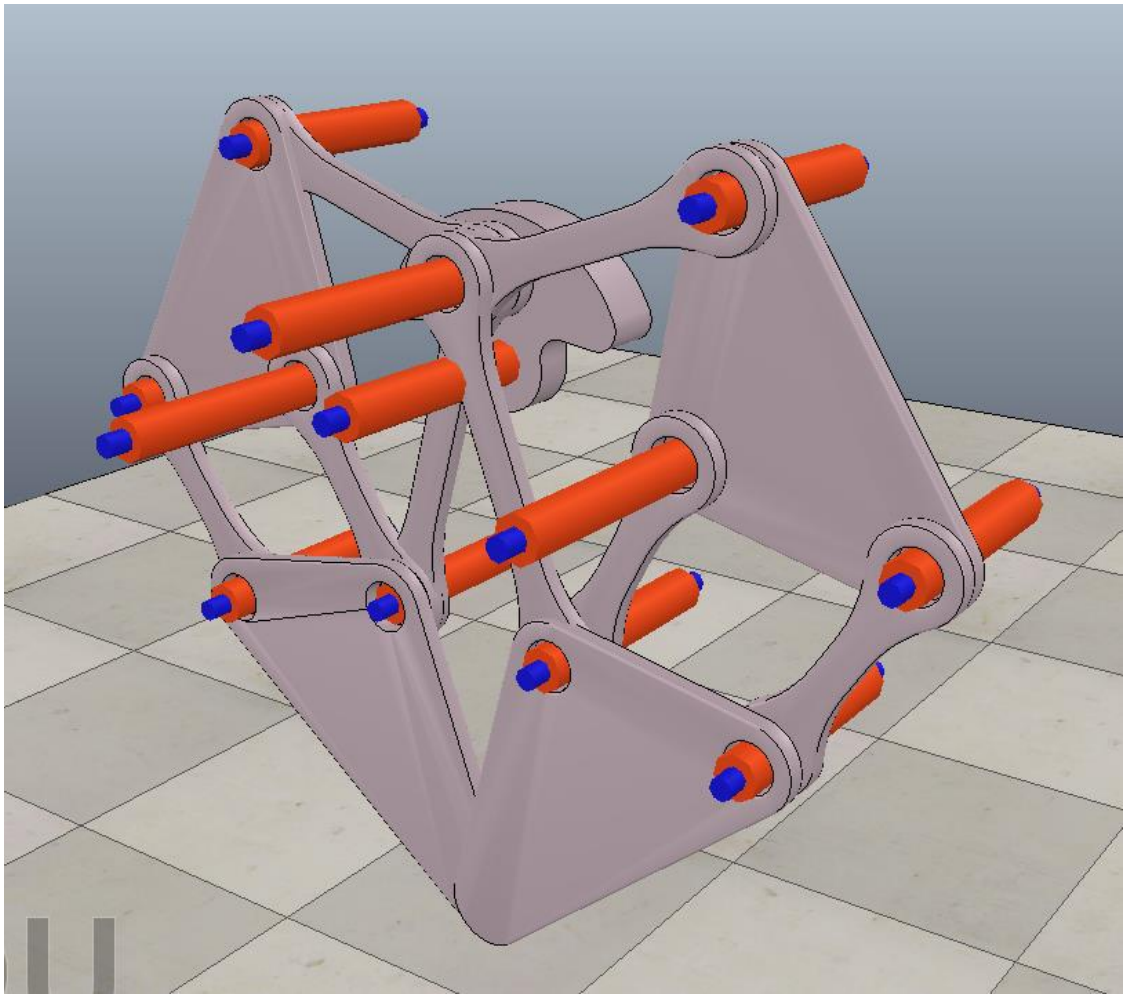


圖 4.29 各關節重疊

7.建立從屬關係。

從屬關係:主動件傳達動力給從動件,兩者的關係便是從屬關係,通常主動件位於從動件的上階。

Dummy:連結點,可以視為一質點當兩質點連結固定,則兩點距離將不會改變,通常遇到三接頭問題會建立兩個 dummy 來做連結,一點在 joint,另一點則在 link 上。

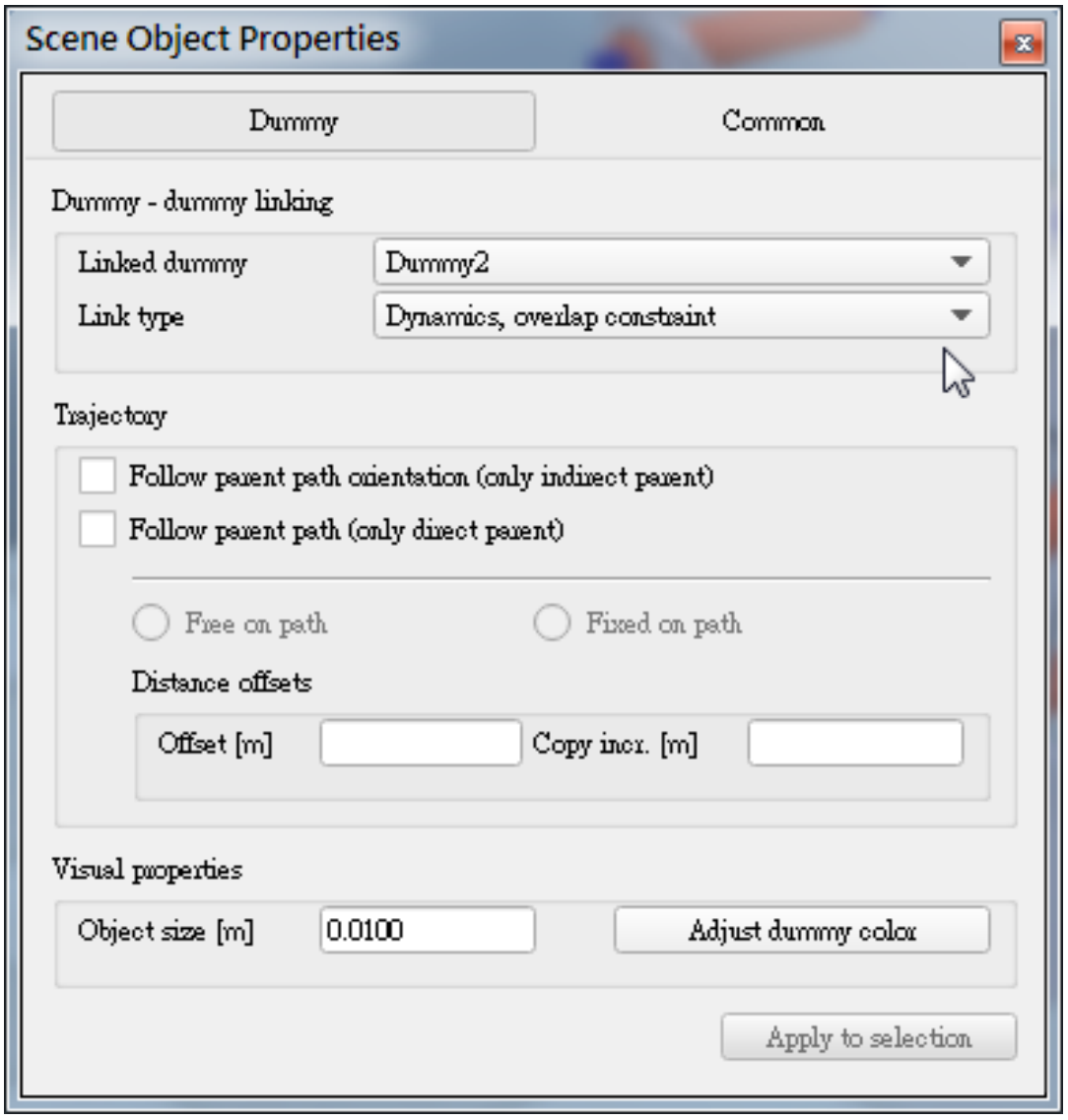


圖 4.30 Dummy 功能

(1)依照主動從動關係來組立，下圖各零件、joint 名稱對照 4-2 章裡 pyslvs 個點及連桿名稱。

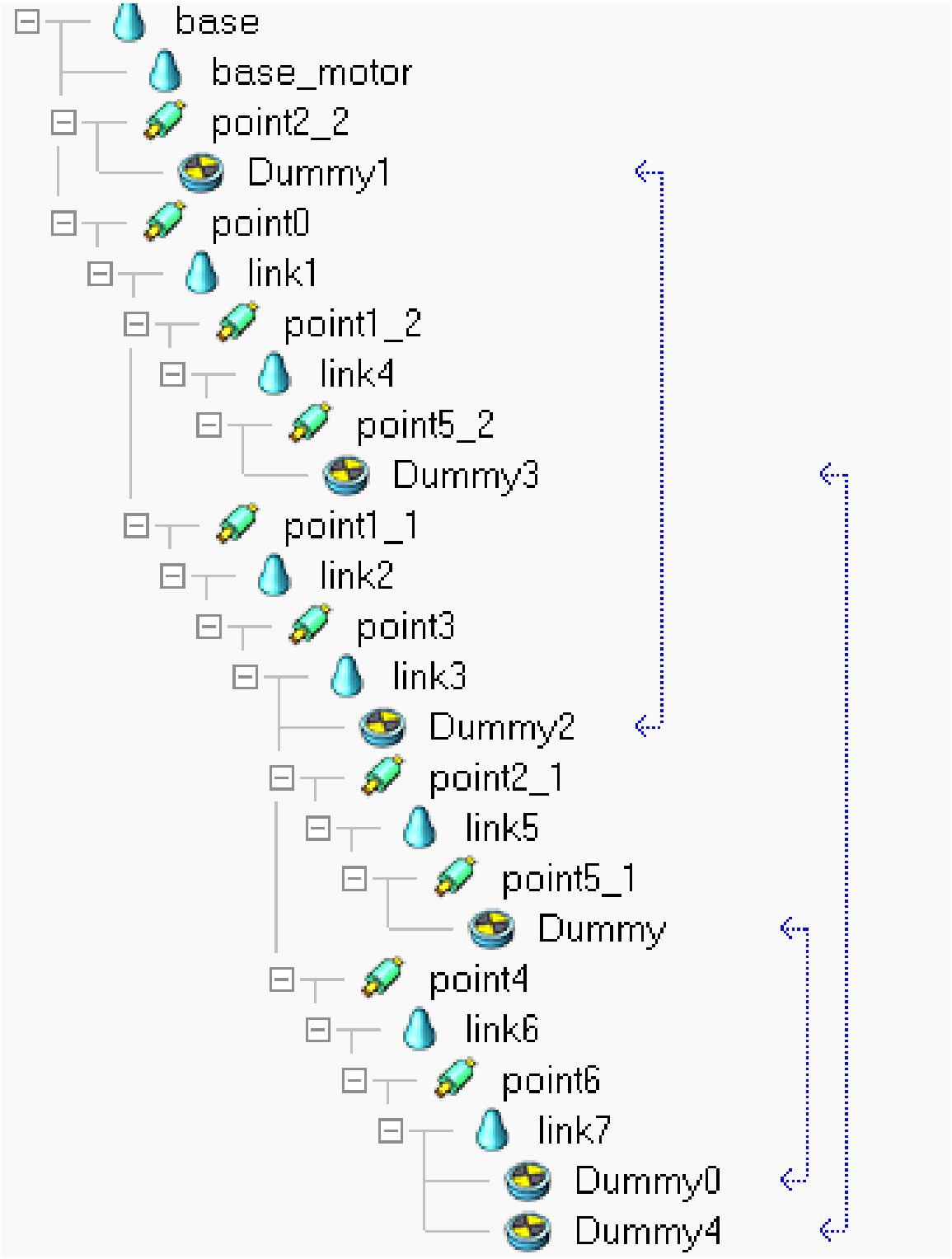


圖 4.31 從屬關係

8.可以設置 motor 轉速測試從屬關係是否有錯。

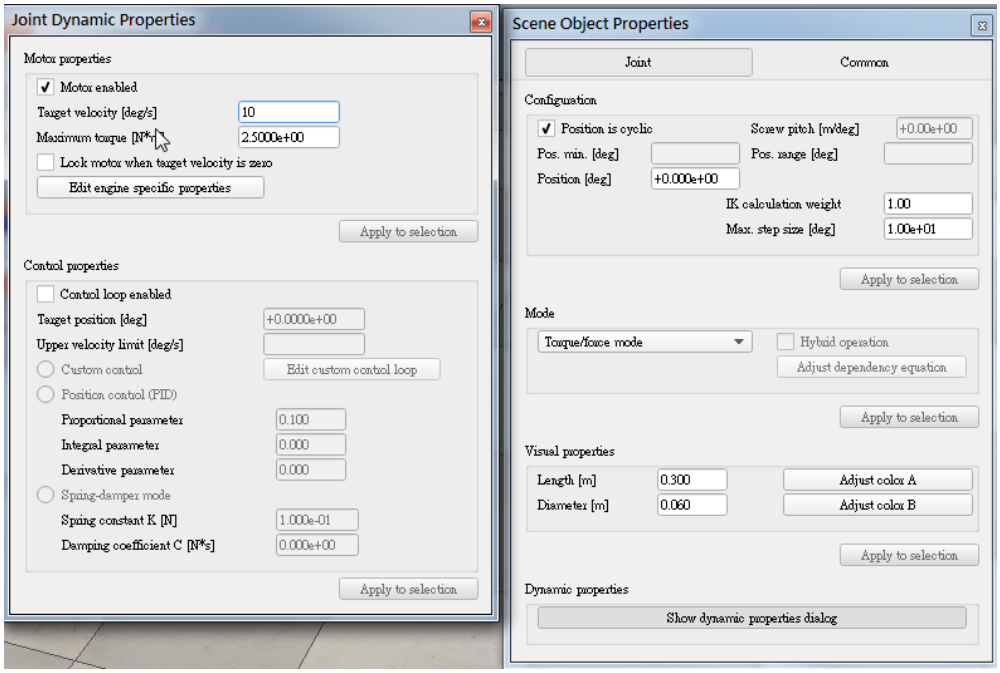


圖 4.32 motor 轉速測試

9.依序上述步驟完成其他多連桿。圓柱在確定完成後清除掉即可。

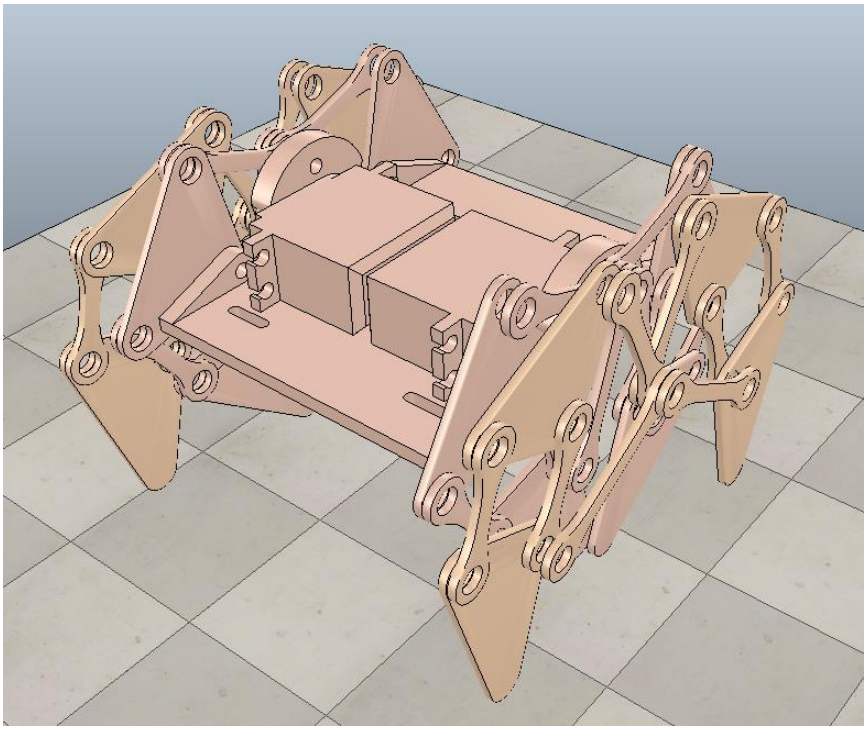


圖 4.33 完成組立

10.設定馬達轉速讓行走機構能在地上行走

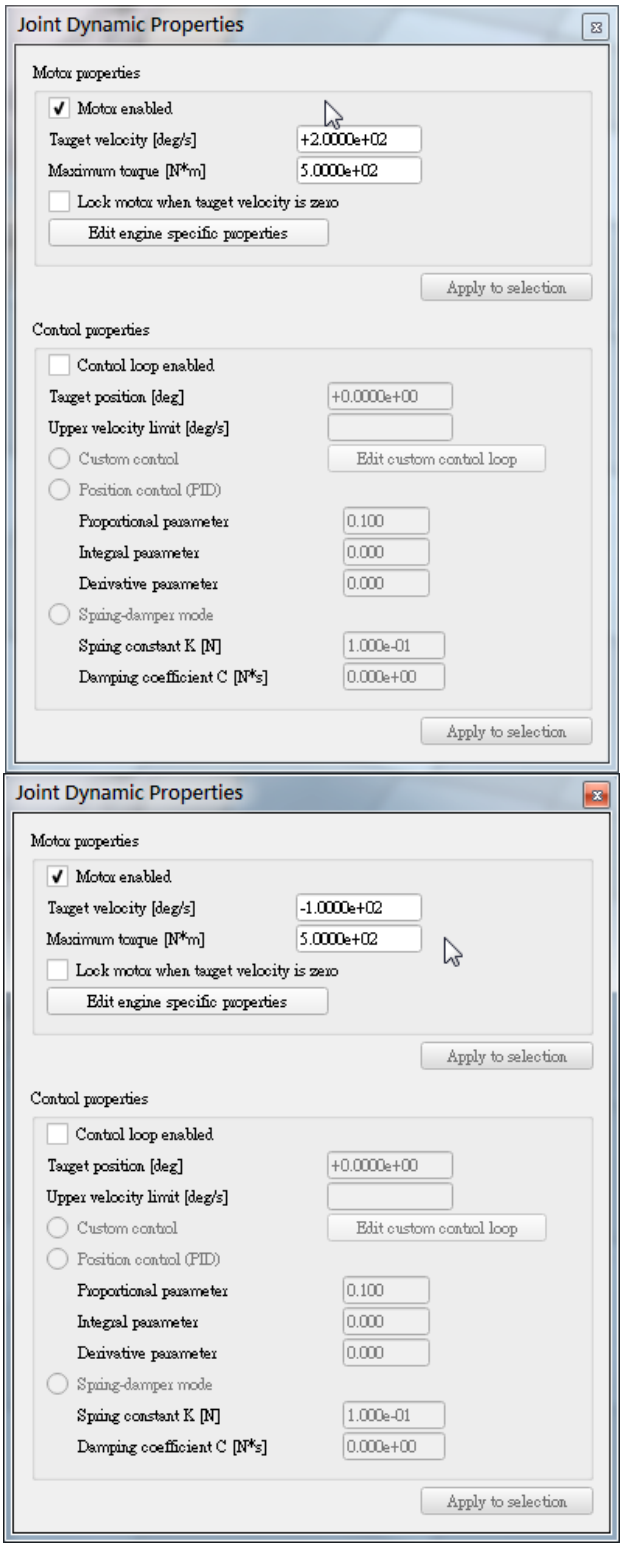


圖 4.34 左馬達及右馬達

4.5 模擬控制

用超音波感測器並編寫 Scripts 內部程式控制仿生獸配合道路轉彎、直走等等..

1.建立 Proximity sensor(超音波感測器)，右鍵點選 add→Proximity sensor

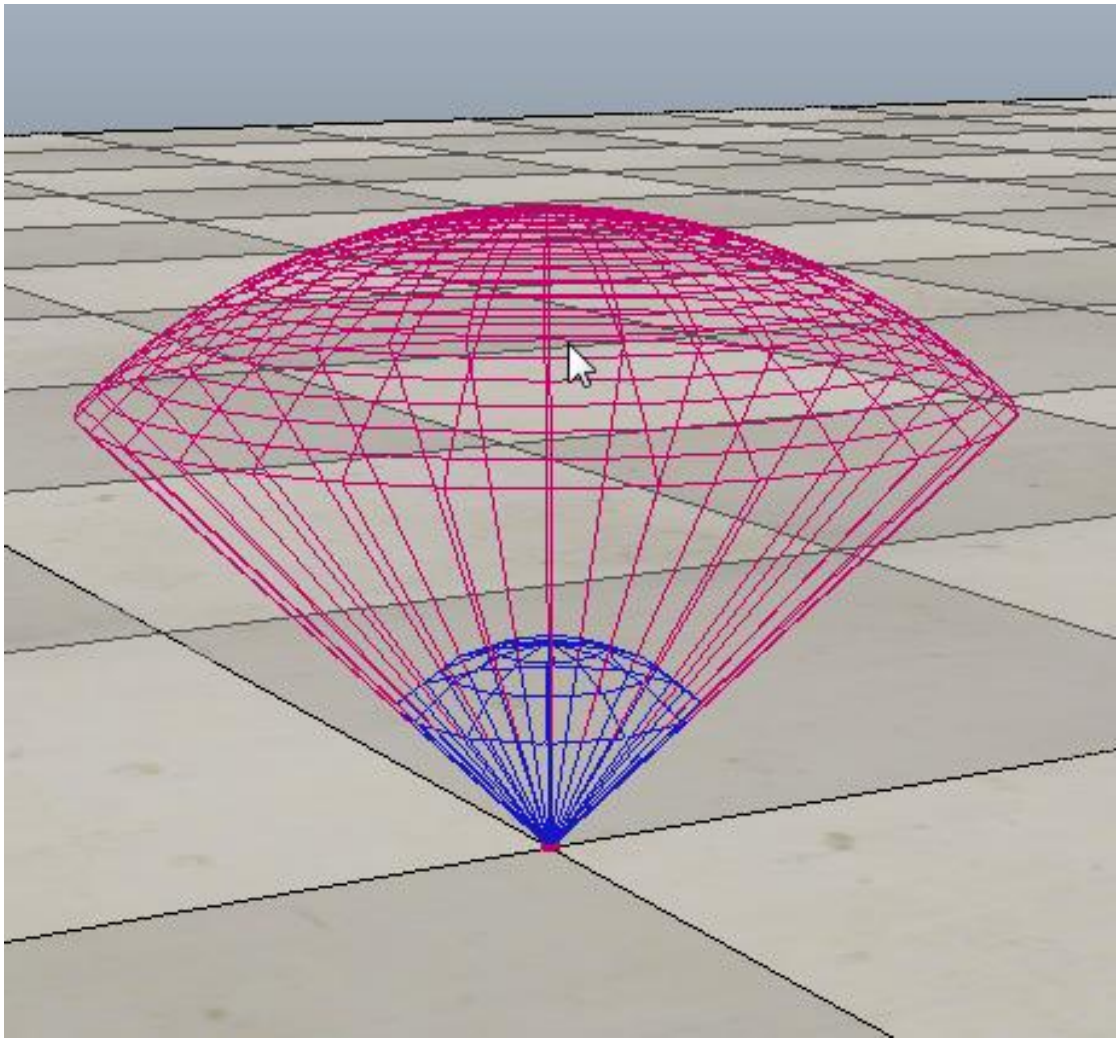
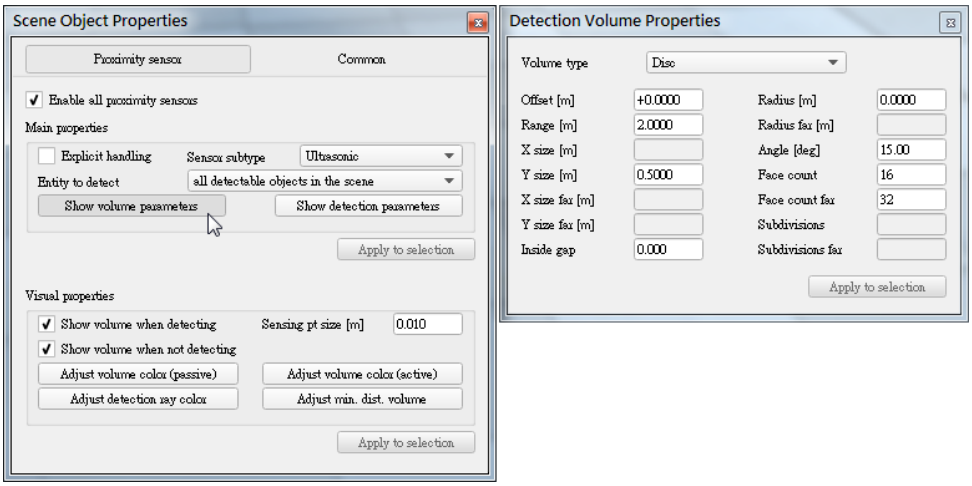


圖 4.35 Proximity sensor

2.Proximity sensor 尺寸設定

圖 4.36 sensor 尺寸設定



3.新增 8 個 Proximity sensor，依序排列好

(1)位置與 Base 重疊座標點、旋轉軸為 Z 軸

(2)第一與最後一個成 180 度，每個間隔為 20 度

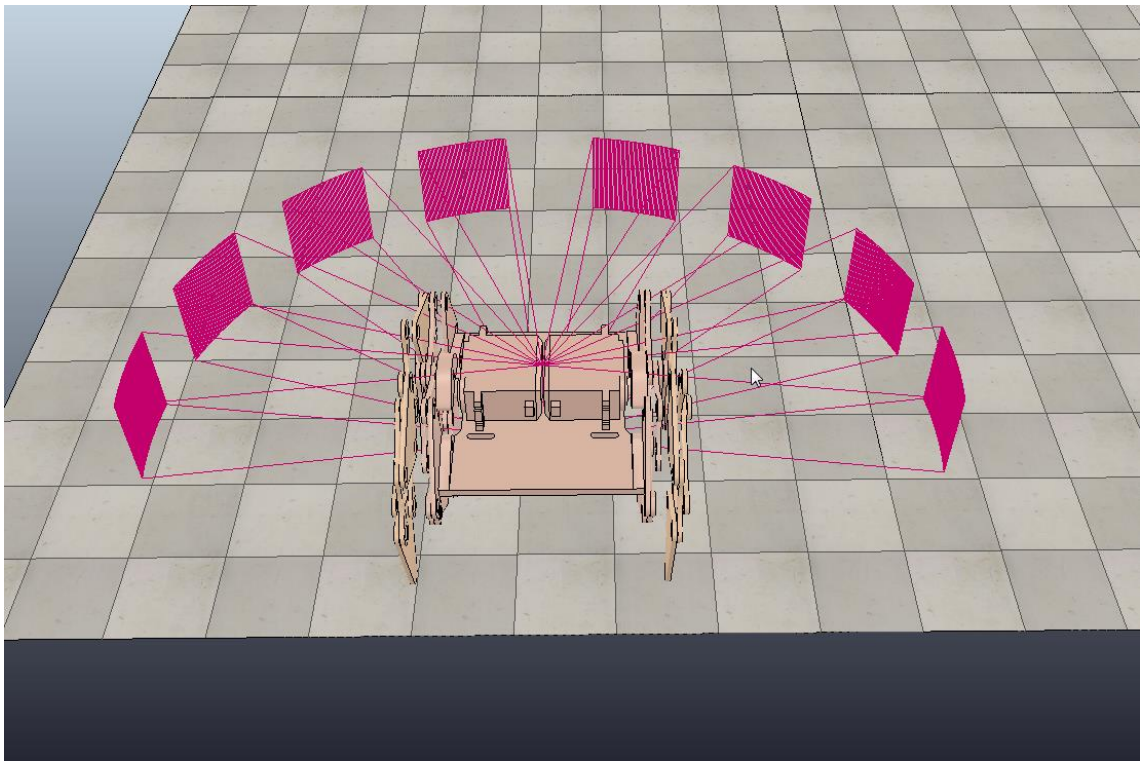


圖 4.37 Proximity sensor 排列

4.建立障礙物，個人自定義設計。本次主題為封閉、曲折、S 型及路障型

5.因為感測器必須要感測到障礙物，所以障礙物模型設定必須將 Detectable(檢測)勾起來，而仿身獸本身不需要被感測，則不設定。

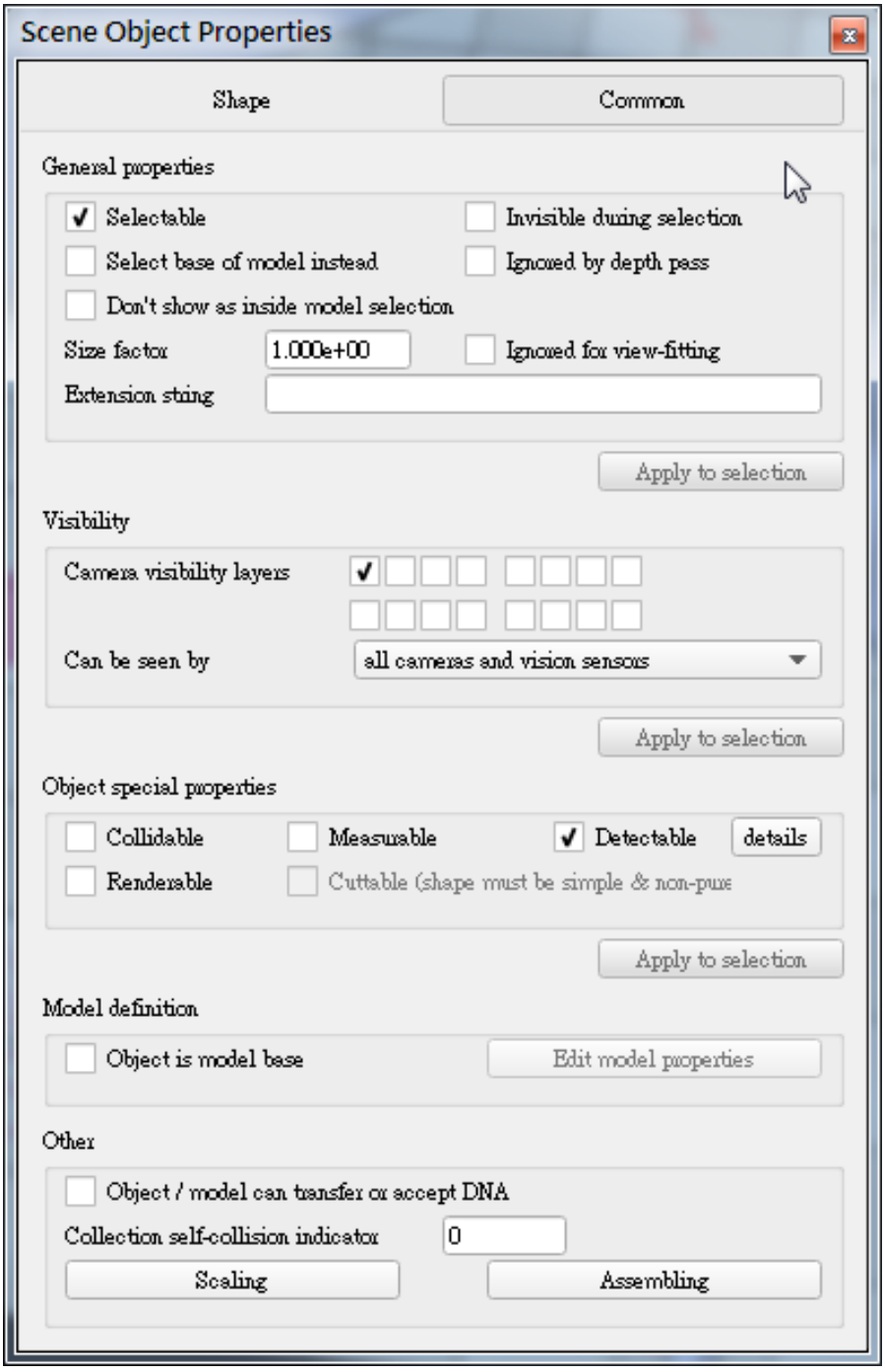


圖 4.38 Detectable(檢測)

6.點選左側功能 Scripts，並建立新文本

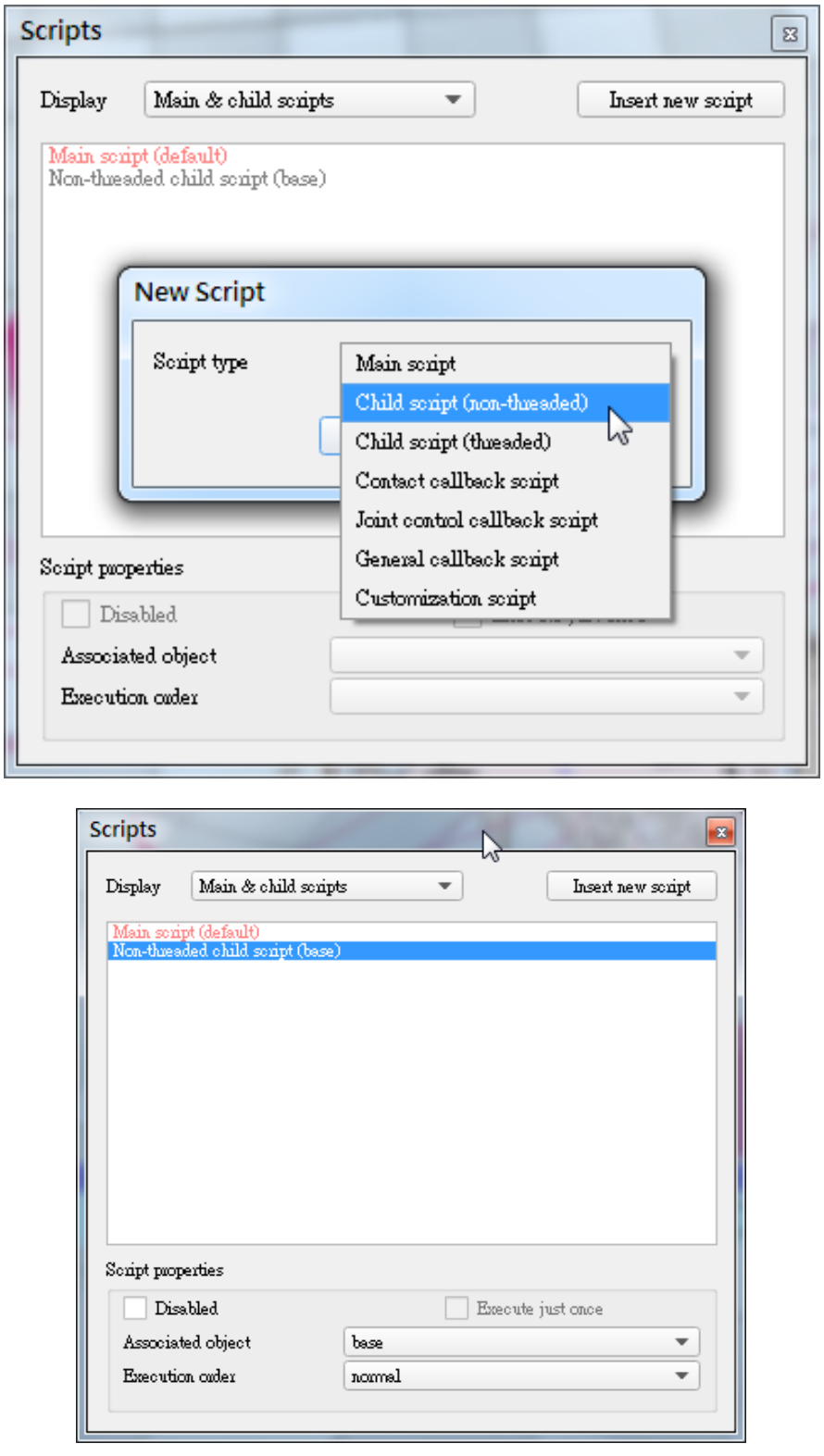


圖 4.39 Script 文本設定

7.編寫程式

圖 4.40 程式碼

Non-threaded child script (base)

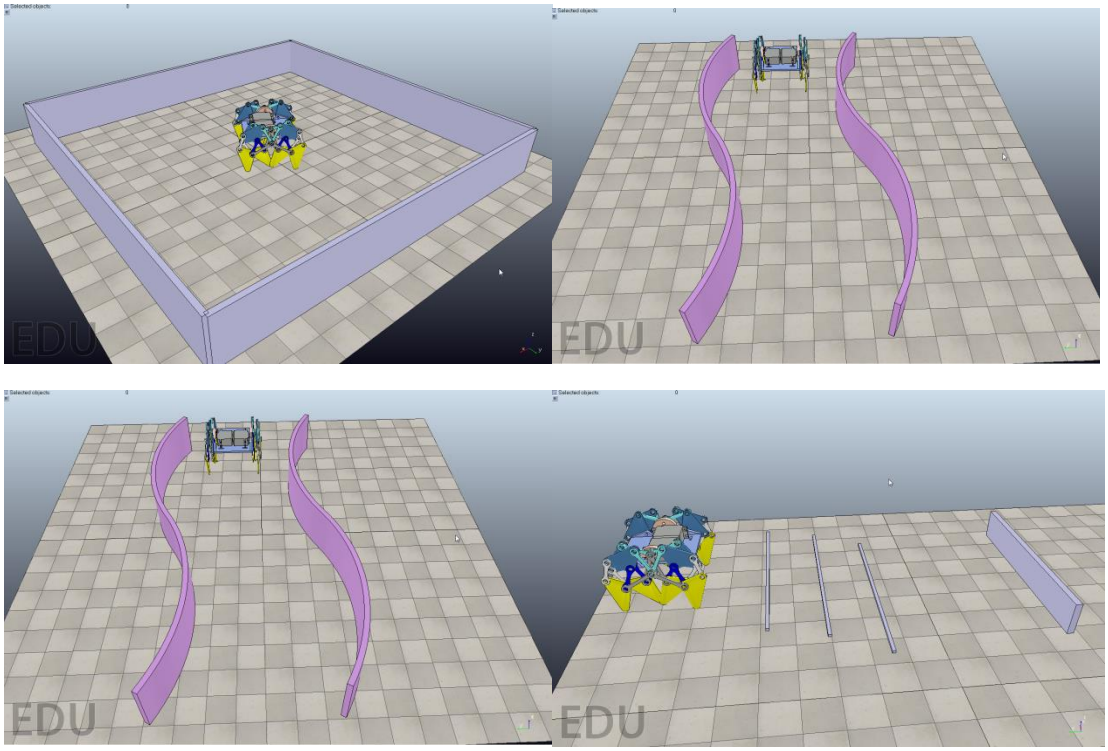
```
if (sim_call_type==sim_childscriptcall_initialization) then
    usensors={-1,-1,-1}
    for i=1,8,1 do
        usensors[i]=simGetObjectHandle("Proximity"..i)
    end
    motorLeft=simGetObjectHandle("left_input")
    motorRight=simGetObjectHandle("right_input")
    noDetectionDist= 2
    maxDetectionDist=0.5
    detect={0,0,0}
    braitenbergL={-0.2,-0.4,-0.6,-0.8,-1,-1.2,-1.4,-1.6}
    braitenbergR={-1.6,-1.4,-1.2,-1,-0.8,-0.6,-0.4,-0.2}
    v0=2.0
end
if (sim_call_type==sim_childscriptcall_actuation) then
    for i=1,8,1 do
        res,dist=simReadProximitySensor(usensors[i])
        if (res>0) and (dist<noDetectionDist) then
            if (dist<maxDetectionDist) then
                dist=maxDetectionDist
            end
            detect[i]=1-((dist-maxDetectionDist)/(noDetectionDist-maxDetectionDist))
        else
            detect[i]=0
        end
    end
    vLeft=v0
    vRight=v0
    for i=1,8,1 do
        vLeft=vLeft+braitenbergL[i]*detect[i]
        vRight=vRight+braitenbergR[i]*detect[i]
    end
    simSetJointTargetVelocity(motorLeft,vLeft)
    simSetJointTargetVelocity(motorRight,vRight)
end
if (sim_call_type==sim_childscriptcall_sensing) then

end
if (sim_call_type==sim_childscriptcall_cleanup) then

end
```

8.模擬封閉、S 型、曲折、路障等等動態模擬。

圖 4.41 動態模擬類型



第五章 分析與結論

5.1 Pyslvs 分析數據

由 4-2 章完成步驟而得到下列數據圖表:

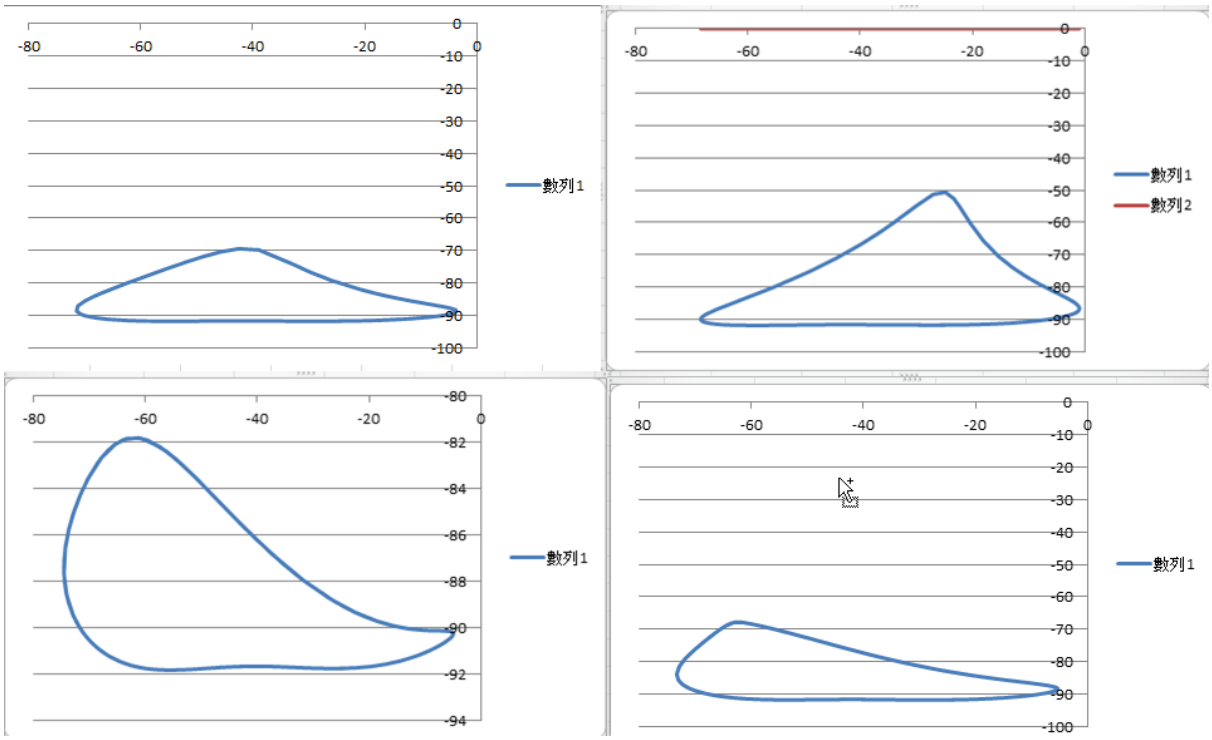
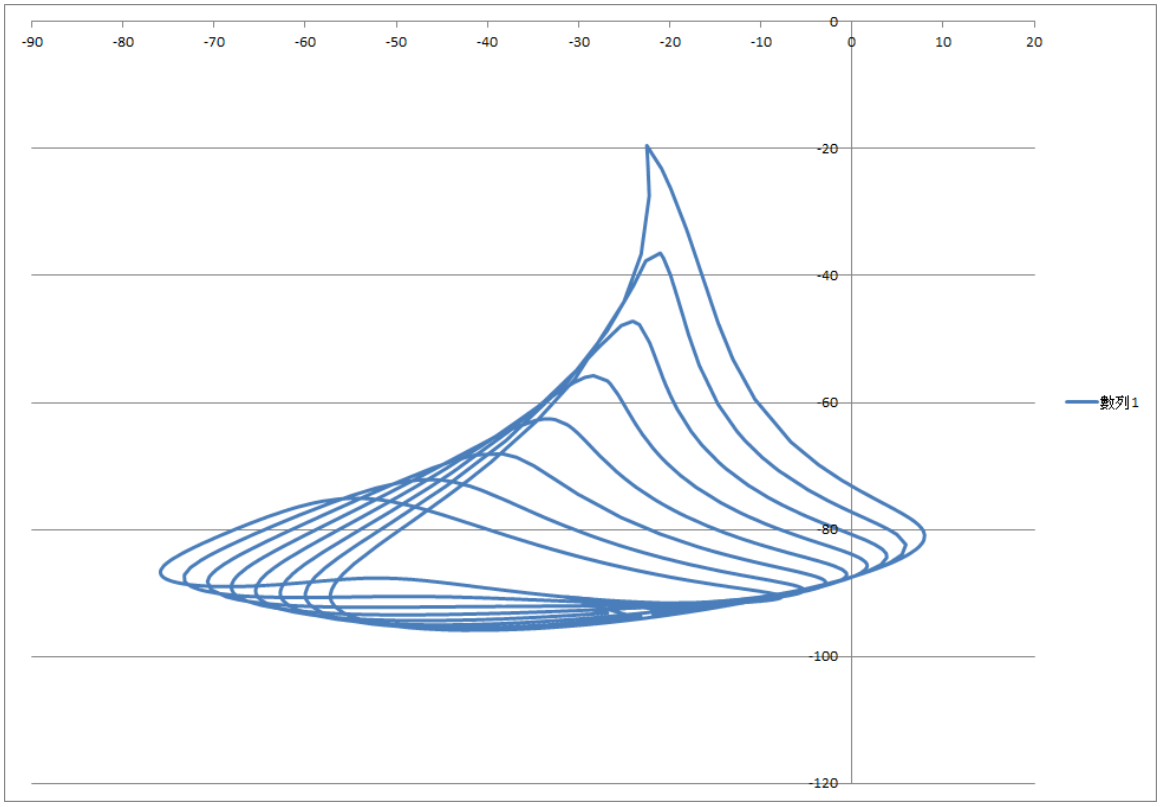


圖 5.1 數據圖表(單位:mm)

左上為基本的 Jansen worker，另外三種是修改版。由圖 5-1 可知，每個圖表中跨步距離幾乎相同，改變的幾乎只有上方曲線高低，當中以左下提腳高度最低，右上提腳高度最高，右下左上跨步高度一樣，但一個最高點偏中間，另一個則偏左邊。如想設計一個可跨越一定高度障礙物的行走機構，那麼右上的設計非常適合。

若想增加比較數據，可以將每個連桿尺寸作為變數，畫出變化圖


-90
-80
-70
-60
-50
-40
-30
-20
-10
0
10
20
0
-20
-40
-60
-80
-100
-120
數列1

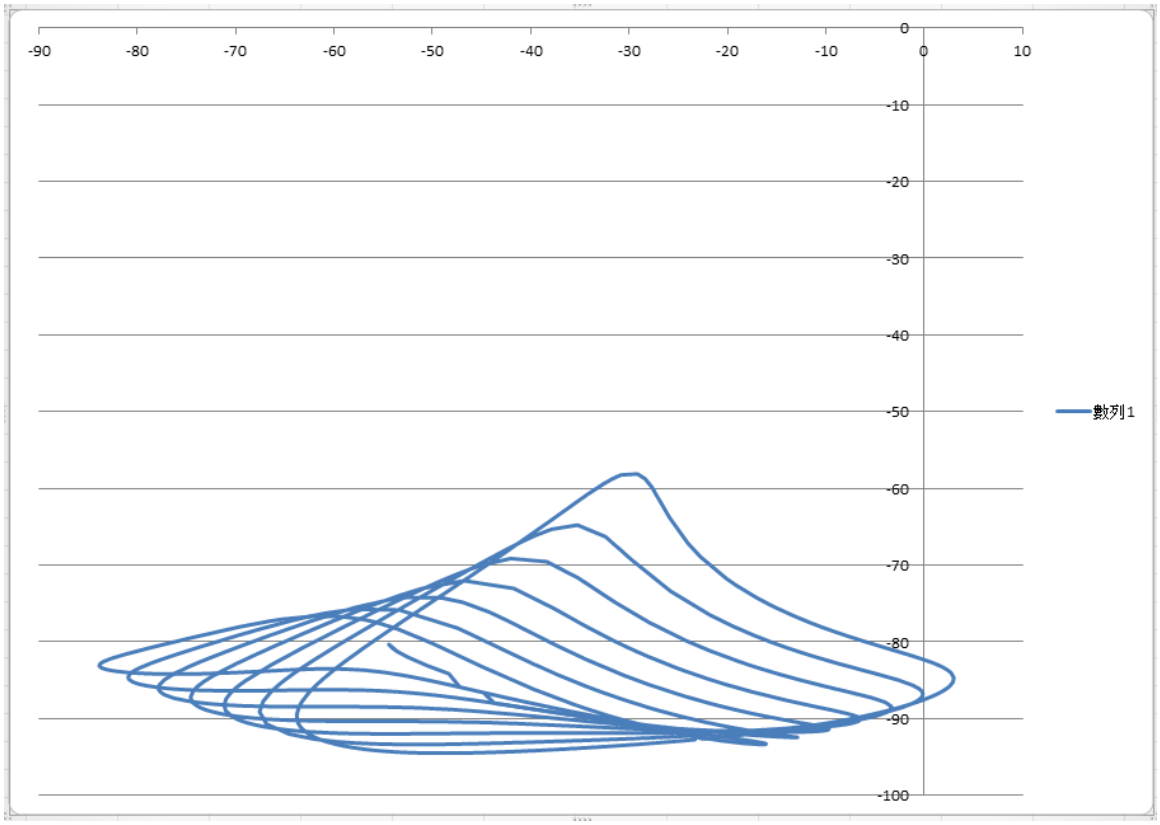

-90
-80
-70
-60
-50
-40
-30
-20
-10
0
10
0
-10
-20
-30
-40
-50
-60
-70
-80
-90
-100
數列1

圖 5.2　變化圖

5.2 延伸問題

在研究過程中，我們發現，以 Jansen worker 為雛形的行走機構設計大部分是難以跨越很高的障礙物的，如圖 5-1 所示，我們將第二張數據表的機構開啟並展示所有輸出點的路線。

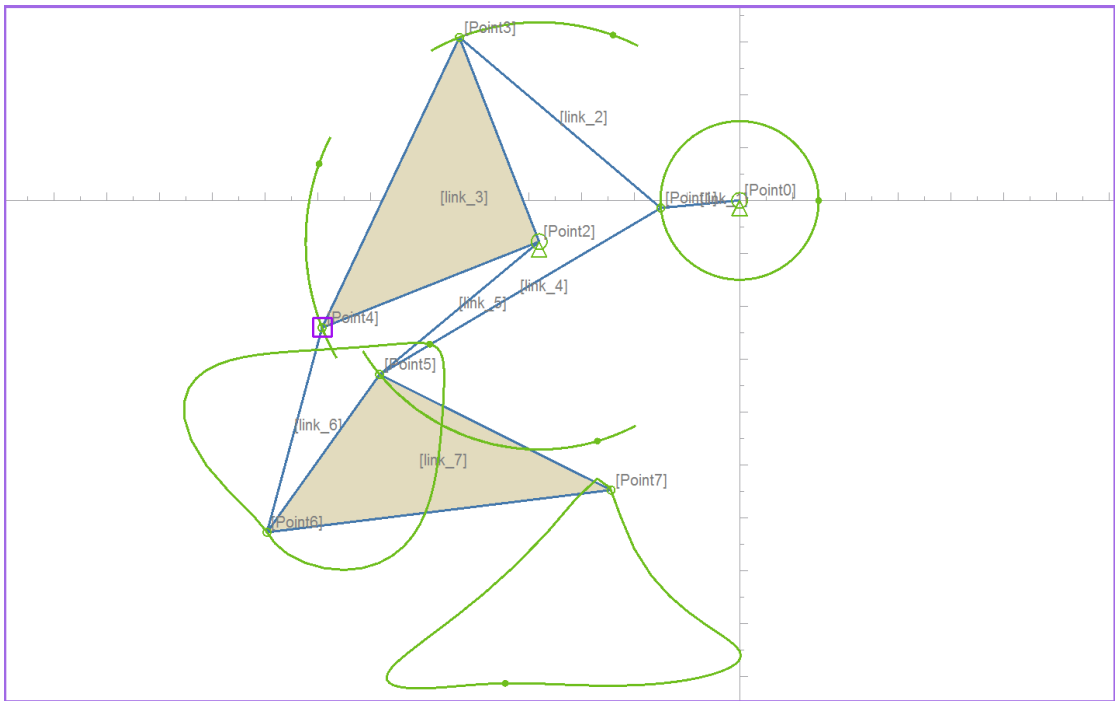


圖 5.3　Pyslvs 分析圖

如圖 5-2 所示，雖然 Point7 可以提到非常高的點，但同時也超過 Point6 最低點的高度，這代表若此機構在行走時，Point6 必定會在某轉角觸地而使得行走機構出問題，所以設計 jansen worker 時，應該取 Point7 能提到最高，同時不會高於 Point6 的最低點。

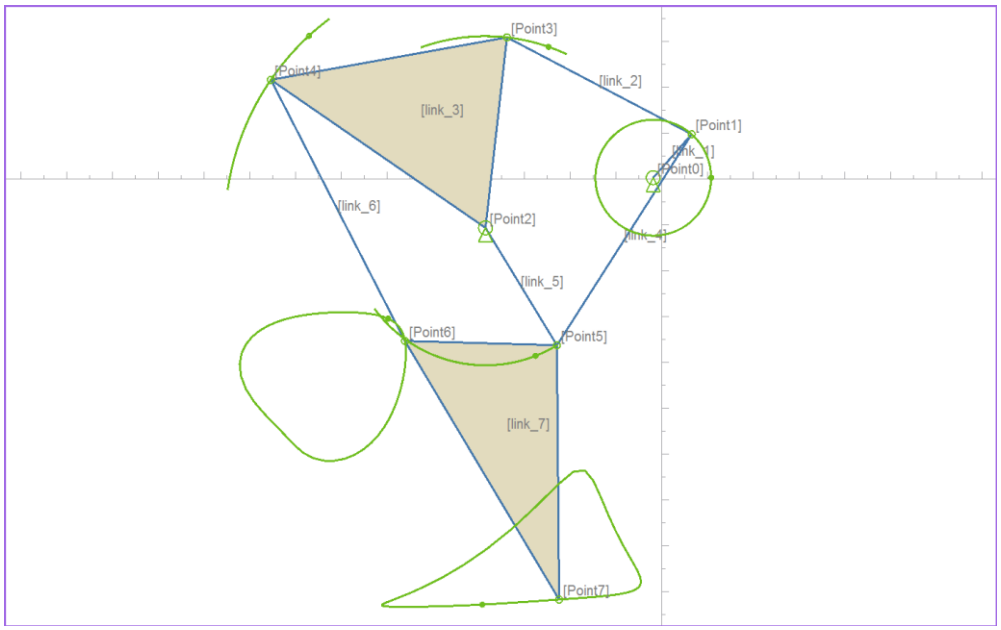


圖 5.4　Pyslvs:Point6 及 Point7 路徑

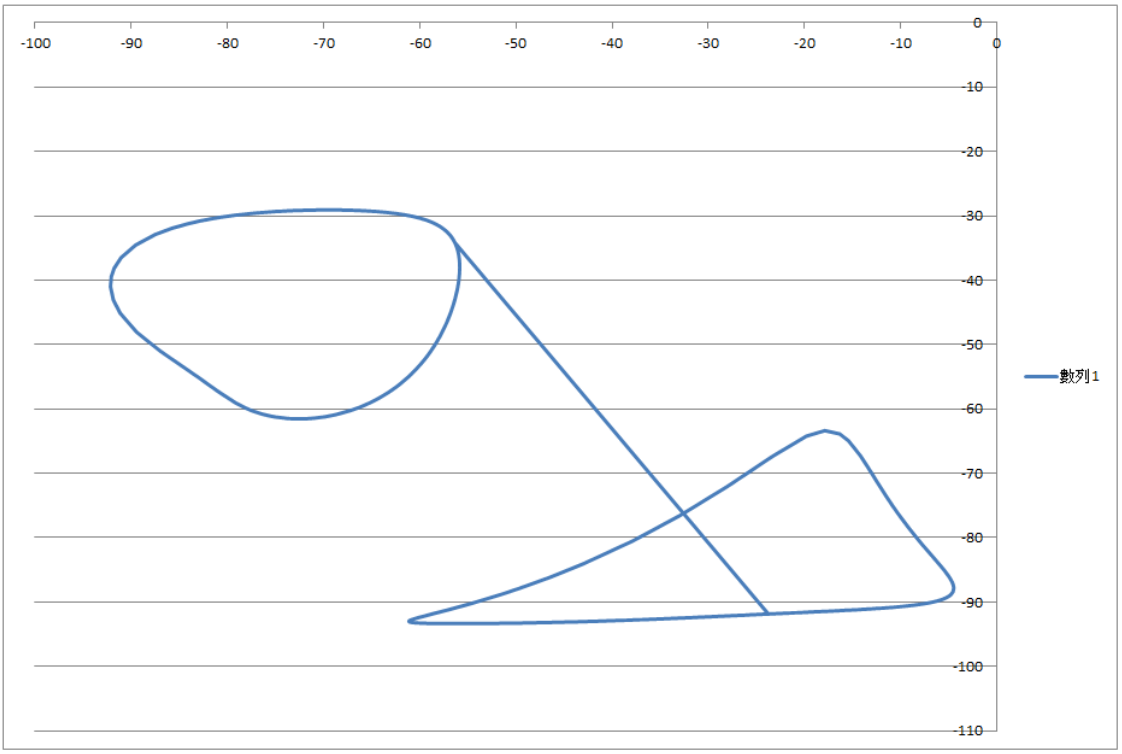


圖 5.5　Point6 及 Point7 路徑

如圖 5-3 所示，針對 Point6 及 Point7 的問題回到 Pyslvs 做修改，得到比雛形高約 10mm 的高度。

其他試著增加提腳高度的方法，已知兩種。

1.比例放大，將原尺寸比例放大。

2.跨步距離短，跨步高度高。

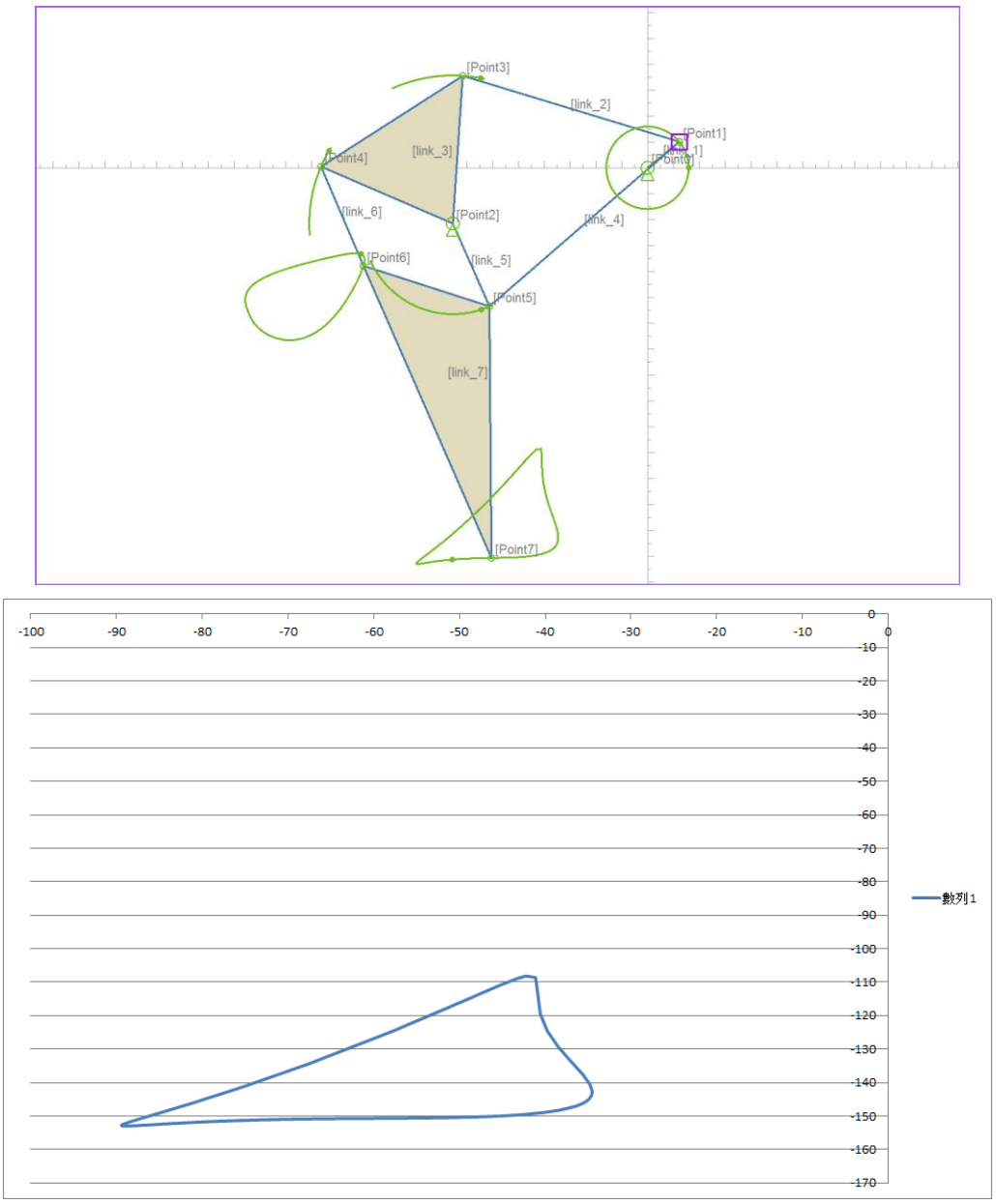


圖 5.6　比例放大，加上距離短，提腳高度高

5.2 總結

在了解輸出點移動路線的方程式推導後，不用再盲目設計，只要使用 Pyslvs 設計特殊路徑的連桿機構，再從 Onshape 建立模型到 V-Rep 模擬控制，就可以讓我們在製作實體前，得到我們想要的答案，節省不必要的實體資源。

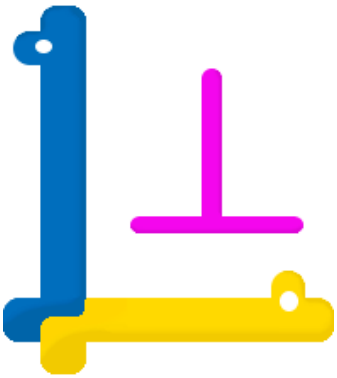


圖 5.7　各軟體圖示

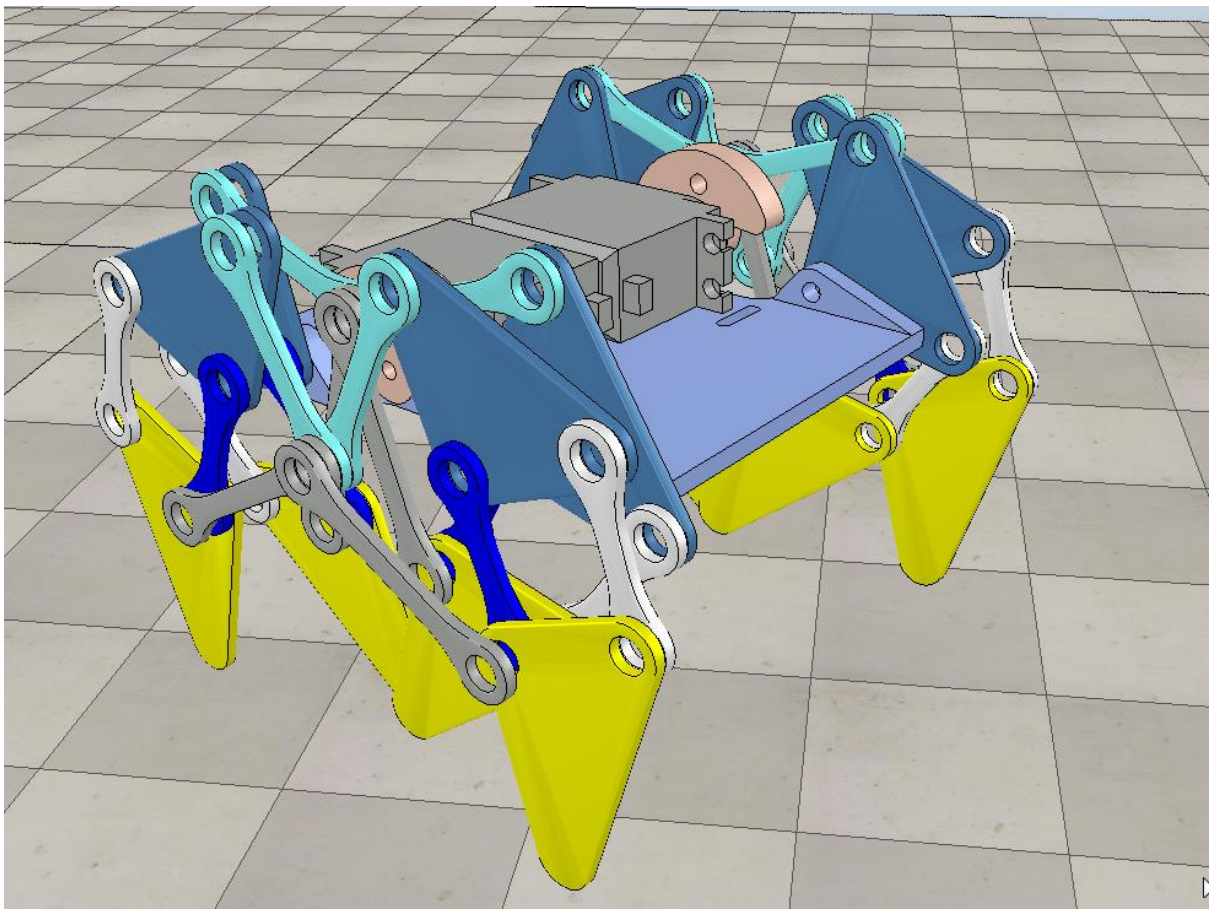


圖 5.8　設計成形

參考文獻

[1] https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6721968&tag=1

Energy Based Position Control of Jansen Walking Robot

[2] http://www.coppeliarobotics.com/

V-Rep

[3] https://github.com/KmolYuan/Pyslvs-PyQt5

Pyslvs-PyQt5

[4] https://www.onshape.com/

Onshape

[5] http://project.mde.tw/blog/pyslvs_triangle_expression.html

Triangluar analysis

[6] https://ieeexplore.ieee.org/document/7030533/

Design and analysis of theo Jansen's mechanism based sports ground (pitch) marking robot Sign In or Purchase

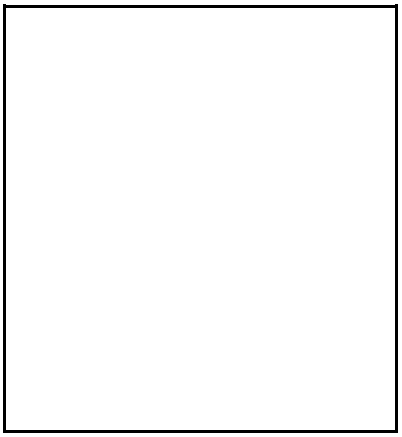
附錄

專題使用研究網站

名稱	網址	說明
專題網誌	http://project.mde.tw	專題研究紀錄存放
goodle word	https://docs.google.com/document/d/1Apoe-almX19GUHsgpsZwCpglGkCjnsuVZIvr_-DzBYo/edit?usp=sharing	專題文書報告(檢視)
google ppt	https://docs.google.com/presentation/d/1NcdLke6KBewAG8bB7Vx11A9tSVW2r2TfuXUBvDORT94/edit?usp=sharing	專題簡報(檢視)
GitHub	https://github.com/40423116	40423116 github(專題存放)
Pyslvs 倉儲	https://github.com/KmolYuan/Pyslvs-PyQt5	Pyslvs 套件說明

Vrep	http://www.coppeliarobotics.com/	Vrep 功能說明
Onshape	https://www.onshape.com/	線上建模教學

作者簡介



姓　　名：李冠辰

學　　號：40423116

畢業學校：國立虎尾科技大學

機械設計工程系