Robotics_Project2

A081615　黃伯宇

一、介面說明(與 Project1 相同，使用 Python 編寫程式)

本程式碼的編寫環境為 Python，是一種廣泛使用的直譯式、進階編程、通用型程式語言，可在 Windows 與 Linux 等作業系統執行，以 C 語言為開發平台。電腦若無 Python，安裝程式安裝步驟如下：

*若有 python，直接執行最後第 4 步驟，有缺 package 就看步驟 3。

1. 安裝 Python 與 Visual Studio Code
   可以依照教學影片 https://www.youtube.com/watch?v=wqRlKVRUV_k&list=PL-g0fdC5RMboYEyt6QS2iLb_1m7QcgfHk (標題: Python 簡介、安裝、與快速開始)

2. 安裝完成選定執行程式資料夾的位置(點選工作列 File>Open Folder>點選欲執行檔案之資料夾)

3. 安裝欲執行之程式內運算式需用到的 package (在命令列輸入以下四個指令"pip install numpy"、"pip install Pillow"、"pip install math"、"pip install matplotlib")

4. 輸入" Robotics_Project2_A081615 .py"，執行檔案

二、程式架構說明

先導入程式執行需使用到的 package[1]，再定義五個函式[2]，給定 A、B、C 三點的 noap 之後[3]，開始進行以下兩種運算：

(一)、Joint move

將三點的 noap 用逆向運動學計算出八組解，取無超出各軸角度限制的其中一組解，決定各點關節姿態[4]。依著 straight line portion-> transition portion-> straight line portion 將 A、B、C 各點帶入運算式，每 0.002 秒計算一次各軸的角度(or 長度)、角速度(or 速度)與角加速度(or 加速度)[5]。將所得到各軸的角度、角速度與角加速度分別儲存[6]，繪製六軸軸變數圖[7]、軸速度圖[8]、軸加速度圖[9]與 3D 的軸座標軌跡規劃曲線圖[10]，最後在軌跡規劃曲線圖中加入末端點方向與其中一軸持續變化[11]來了解座標是如何位移與旋轉的。

(二)、Cartesian Move

將 Joint move 所得到六軸的解用正向運動學得到 A、B、C 三點的 xyz, Φ, θ, ψ[12]，先判斷：若 $|\psi C - \psi A| > 90^\circ$，則 $\psi A+180^\circ$ 且 $\theta A = -\theta$ [13]。依著 straight line portion => transition portion => straight line portion 將各點帶入運算式，每 0.002 秒計算一次卡式座標末端點的位置、速度與加速度[14]，將所得到卡式座標末端點的位置、速度與加速度分別儲存[15]，繪製末端點位置圖[16]、末端點速度圖[17]、末端點加速度圖[18]與 3D 的軸座標軌跡規劃曲線圖[19]，最後在軌跡規劃曲線圖中加入末端點方向[20]。

變數圖、速度圖與加速度圖要手動關閉後程式才會繼續顯示 3D 圖，3D 圖預設只顯示 1 秒，若要看久一點請將第 566 行的 plt.pause(1)中的 1 改為 10 or 想停留的秒數，秒數過後 3D 圖會自動關閉(若手動直接將 3D 圖關閉，程式會停止執行)，接下來再顯示卡式的部分。一樣地，3D 圖預設只顯示 1 秒，若要看久一點請將第 791 行的 plt.pause(1)中的 1 改為 10 or 想停留的秒數，秒數過後 3D 圖會自動關閉。
總結看圖步驟：程式執行>手動關三張圖>3D 圖等待數秒自動關>手動關三張圖>3D 圖等待數秒自動關>最後一張圖

程式碼簡介：
---- Joint move ----
[1]1~9 行：三角函數運算、print 出 2D 圖表與 3D 空間圖所需函式
[2]11~71 行：函式 - Project1 自己寫的正向運動學
[2]74~164 行：函式 - Project1 自己寫的逆向運動學
[2]167~178 行：函式 - 繪製 3D 空間圖所需的程式碼(from Internet)
[2]181~188 行：函式 - straight line portion
[2]190~200 行：函式 - transition portion
[3]203~244 行：給定三個點 ABC
[4]246~257 行：決定各點關節姿態
[5]260~290 行：每 0.002 秒計算一次各軸的位置、角速度與角加速度
[6]292~342 行：將所得到各軸的角度、角速度與角加速度分別儲存
[7]344~396 行：繪製六軸軸變數圖
[8]398~449 行：繪製六軸軸速度圖
[9]451~502 行：繪製六軸軸加速度圖
[10]504~534 行：繪製軸座標軌跡規劃曲線圖(3D)
[11]536~568 行：繪製末端點方向與其中一軸持續變化圖

---- Cartesian move ----

[12]573~575 行：得到 A、B、C 三點的 xyz, Φ, θ, ψ

[13]577~589 行：判斷若 ｜ψC –ψA | > 90°，ψA+180°, θA = - θA

[14]591~620 行：每 0.002 秒計算一次卡式座標末端點的位置、速度與加速度

[15]622~650 行：將所得到卡式座標末端點的位置、速度與加速度分別儲存

[16]652~676 行：繪製末端點位置圖

[17]679~704 行：繪製末端點速度圖

[18]707~733 行：繪製末端點加速度圖

[19]735~764 行：繪製卡式座標軌跡規劃曲線圖(3D)

[20]766~793 行：繪製卡式座標末端點方向

寫到這發現卡式部分應該不是用這麼簡單的寫法，於是在後面寫一個新的但時間不夠(太晚察覺 QQ)，前面就不拿掉了，在後面補充說明目前重寫的部分，希望助教能看程式碼斟酌給分> <

795~808 行：定義位置 1~3 的 n, o, a, p

810~821 行：定義位置 P1~P3

823~828 行：用 Ch5 第 17 頁定義 D(r)中的 T(r)部分

830~846 行：用 Ch5 第 18 頁定義 D(r)=T(r)*Ra(r)*Ro(r)

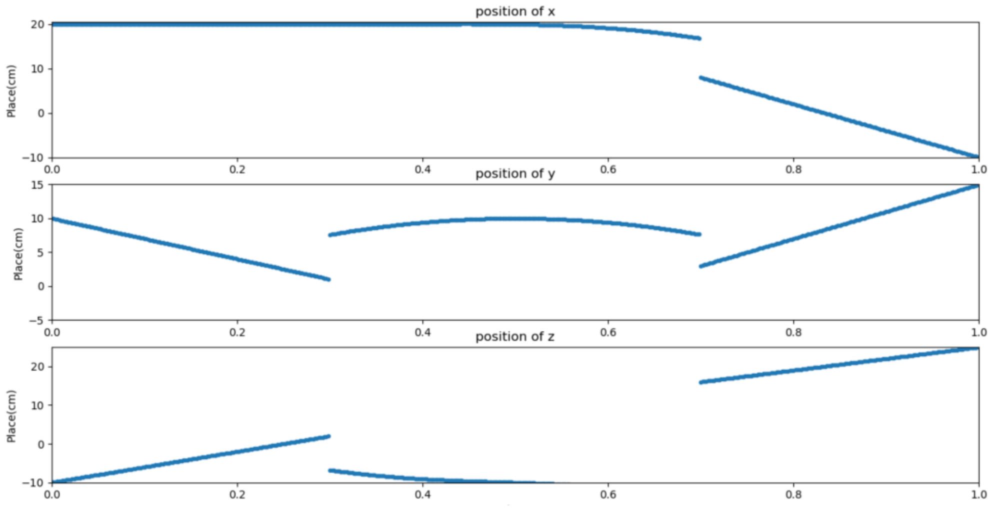
849~852 行、863~866 行：算 D(r)中的 T(r)部分

855~861 行、868~874 行：用 Ch5 第 22、23 頁算 Φ, θ, ψ 變化量

876~884 行：用 Ch5 第 10 頁定義 transition 部分

886~921 行：計算 P1~P3 的 linear > transfer > linear 並儲存

923~948 行：畫出 x, y, z 方向的位置變化圖(線性與轉換的接點未吻合，轉換部分尚須修改)



Linear 部分沒錯，但 transition 部分接不起來尚須修改

三、數學運算說明

Joint Move

將三個位置的矩陣用逆向運動學計算出八組解，取無超出各軸角度限制的其中一組解，決定各點關節姿態，決定 P1 使用第二、P2 使用第二、P3 使用第四個姿態。逆向運動學公式因 project1 有放過，這邊就不再多放了。

straight line portion 是將 Ch5 p.11 的公式做修改(因為只有 3 個點，並非無限多點)，t 的上限 T - t_{acc} 改為 T。

□ For linear portion

$$\begin{cases} q = \Delta C \cdot h + B \\ \dot{q} = \dfrac{\Delta C}{T} \\ \ddot{q} = 0 \end{cases} \qquad h = \frac{t}{T}, t_{acc} \le t \le T - t_{acc}$$

transition portion 是用 Ch5 p.10 的公式，其中 ΔB 的 A-B 並不是 P1 的 A 點，而是從 linear 準備變成 transition 的 A'點，依照題目給的 t_{acc}，實際的 ΔB 為 0.4A - 0.4B，這邊要注意一下。

$$\text{Let } \begin{cases} \Delta C = C - B \\ \Delta B = A - B \end{cases}$$

$$q(h) = [(\Delta C \frac{t_{acc}}{T} + \Delta B)(2-h)h^2 - 2\Delta B]h + B + \Delta B$$

$$\dot{q}(h) = [(\Delta C \frac{t_{acc}}{T} + \Delta B)(1.5-h)2h^2 - \Delta B]\frac{1}{t_{acc}}$$

$$\ddot{q}(h) = [(\Delta C \frac{t_{acc}}{T} + \Delta B)(1-h)]\frac{3h}{t_{acc}^{\ 2}}$$

$$\text{Where } h = \frac{t + t_{acc}}{2t_{acc}} \qquad \text{for } -t_{acc} \le t \le t_{acc}$$

Cartesian Move

這部分較複雜，首先，每點的位置是依照 pos1*D(r) 得出結果，而 D(r) = T_r(r)*Ra(r)*Ro(r)，T_r(r)需要 r, P1, P2, P3 來解，Ra(r)需要 r, θ, ψ 來解，Ro(r)需要 r, Φ 來解，而 θ, ψ 要用 Ch5 p.22 的公式解，Φ 要用 Ch5 p.23 的公式與 θ, ψ 解。文字敘述有點亂，以下用圖示來表明個變數關係

$$D(r) = T_r(r) * Ra(r) * Ro(r)$$

$$T_r(r) = \begin{pmatrix} 1 & 0 & 0 & r_x \\ 0 & 1 & 0 & r_y \\ 0 & 0 & 1 & r_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

p.18

$$Ro(r) = \begin{pmatrix} C(r\phi) & -S(r\phi) & 0 & 0 \\ S(r\phi) & C(r\phi) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

p.20

用P1, P2, P3的noap解

$$x = {}^1n \cdot ({}^2p - {}^1p)$$
$$y = {}^1o \cdot ({}^2p - {}^1p)$$
$$z = {}^1a \cdot ({}^2p - {}^1p)$$

p.21

用θ, ψ解Φ

$$S\phi = -S\psi C\psi V\theta({}^1n \cdot {}^2n) + [(C\psi)^2 V\theta + C\theta]({}^1o \cdot {}^2n) - S\psi S\theta({}^1a \cdot {}^2n)$$
$$C\phi = -S\psi C\psi V\theta({}^1n \cdot {}^2o) + [(C\psi)^2 V\theta + C\theta]({}^1o \cdot {}^2o) - S\psi S\theta({}^1a \cdot {}^2o)$$
$$\therefore \tan\phi = \frac{S\phi}{C\phi}, -\pi \le \phi \le \pi$$

p.23

$$Ra(r) = \begin{pmatrix} S\psi^2 V(r\theta) + C(r\theta) & -S\psi C\psi V(r\theta) & C\psi S(r\theta) & 0 \\ -S\psi C\psi V(r\theta) & C\psi^2 V(r\theta) + C(r\theta) & S\psi S(r\theta) & 0 \\ -C\psi S(r\theta) & -S\psi S(r\theta) & C(r\theta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

where $V(r\theta) = Vers(r\theta) = 1 - \cos(r\theta)$

p.19

用P1, P2, P3的noap解

$$\psi = \tan^{-1}\left(\frac{{}^1o \cdot {}^2a}{{}^1n \cdot {}^2a}\right)$$

Where $\sin\theta > 0$
i.e. $0° \le \theta \le 180°$

p.22

用P1, P2, P3的noap解

$$\tan\theta = \frac{[({}^1n \cdot {}^2a)^2 + ({}^1o \cdot {}^2a)^2]^{\frac{1}{2}}}{{}^1a \cdot {}^2a}, 0^o \le \theta \le 180^o$$

p.22

注意ψ角是否要反轉

If $|\psi_C - \psi_A| > 90°$, then let $\begin{cases} \psi_A = \psi_A + 180^o \\ \theta_A = -\theta_A \end{cases}$

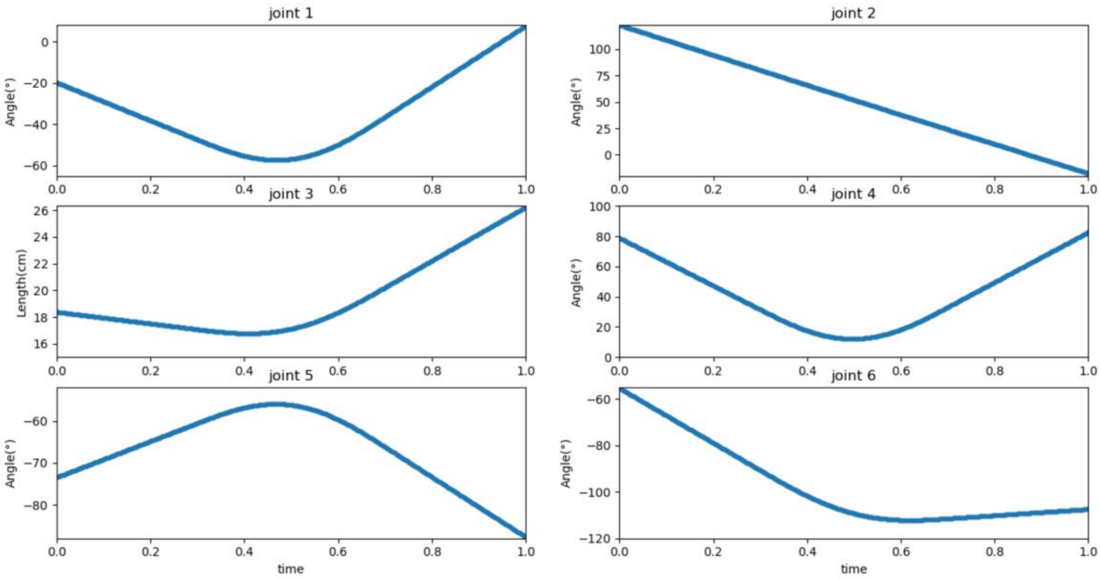
p.25

四、軌跡規劃軌跡圖

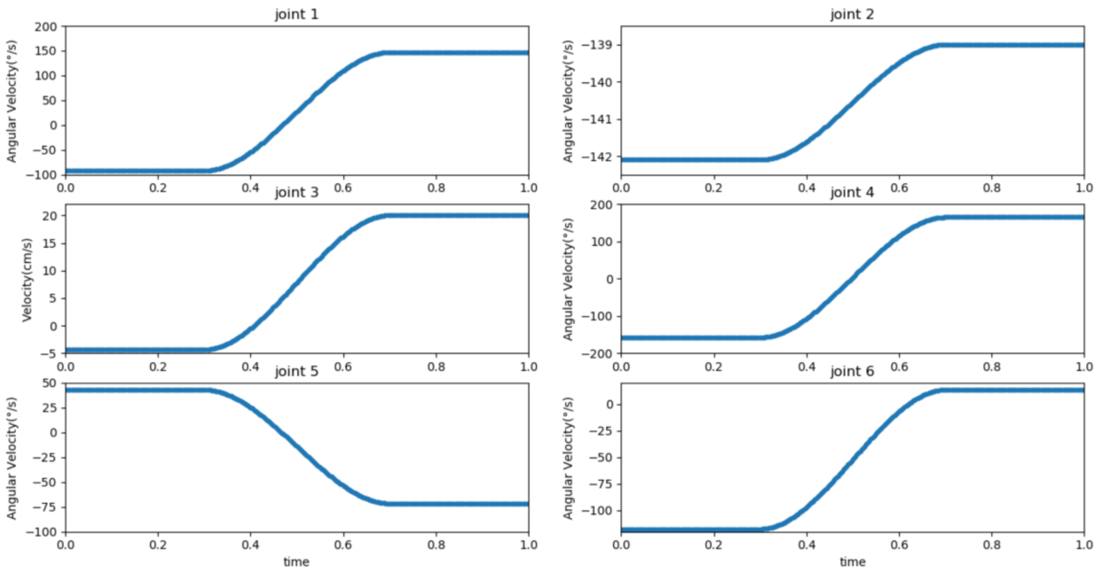
因選擇的 ABC 姿態與助教的那組解相同，若計算無誤，得到的數值與 print 出來的圖會與範例一樣。

Joint Move

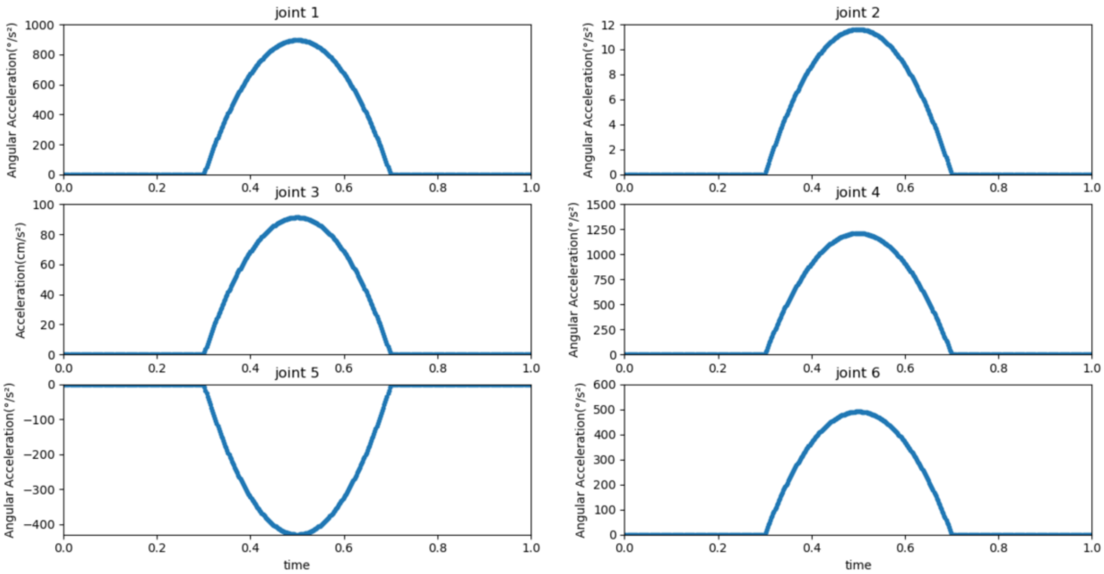
六軸軸變數圖



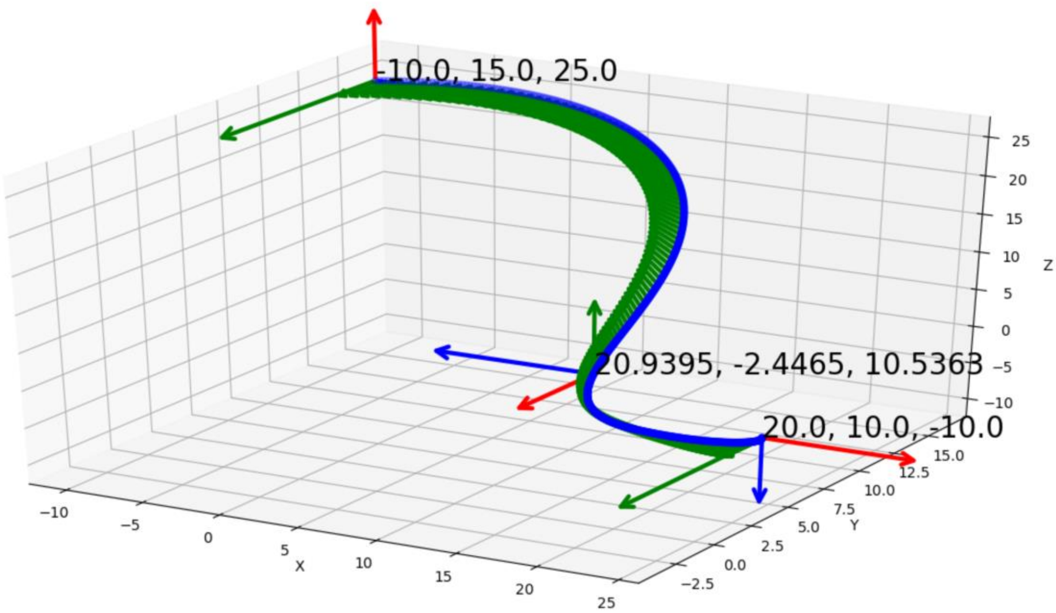
六軸軸速度圖



六軸軸加速度圖

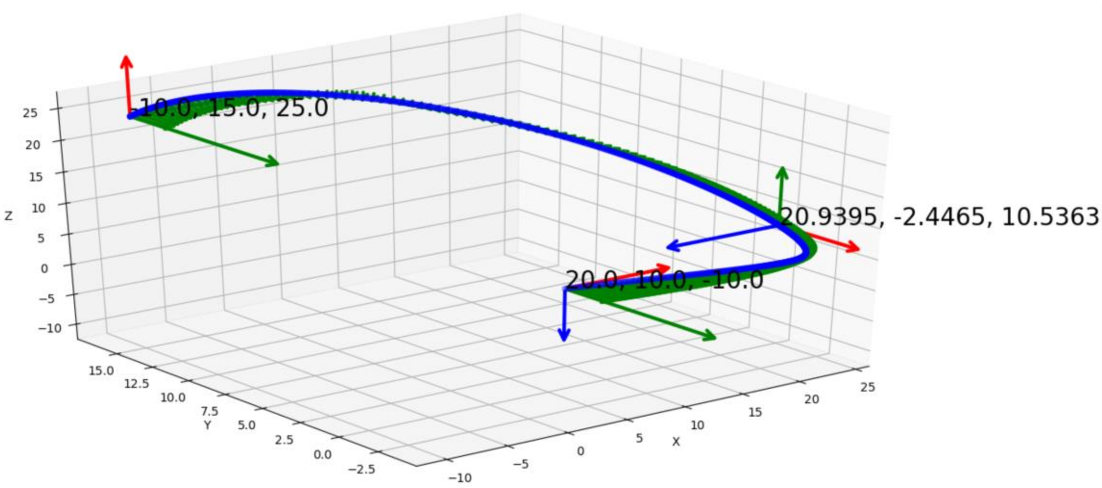


軸座標軌跡規劃曲線圖(左下 X 右下 Y)



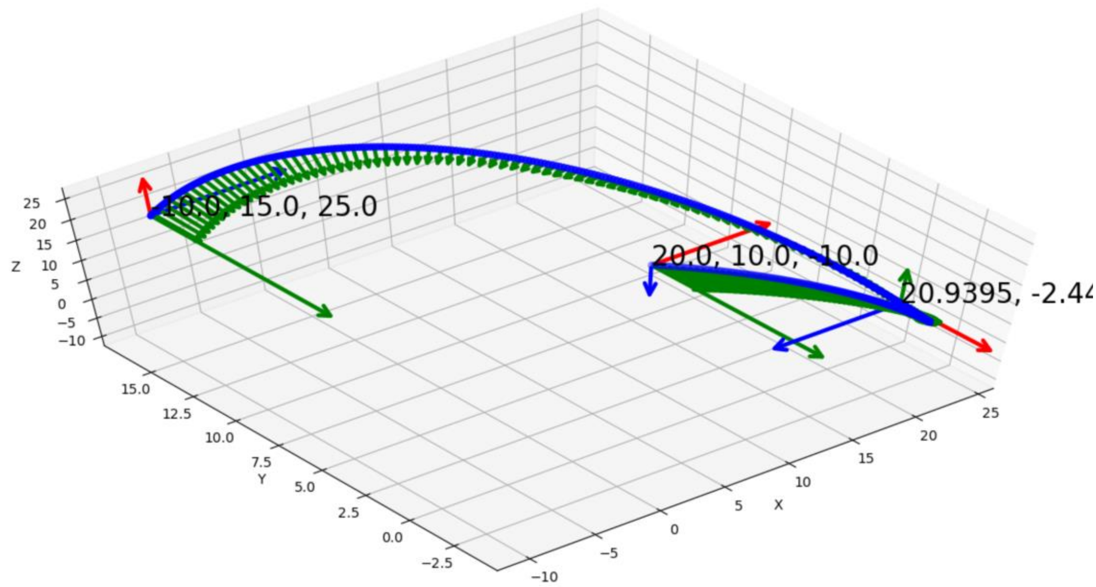
雖然看起來像個 S 型，但換一個視角其實跟助教的答案相同

軸座標軌跡規劃曲線圖(左下 Y 右下 X)



C 點的箭頭被藍線擋住了(向+X 方向)
在 transition 部分，軌跡並沒有通過(20, -5, 10)，
而是從該點附近擦過去(20.94, -2.45, 10.54)。

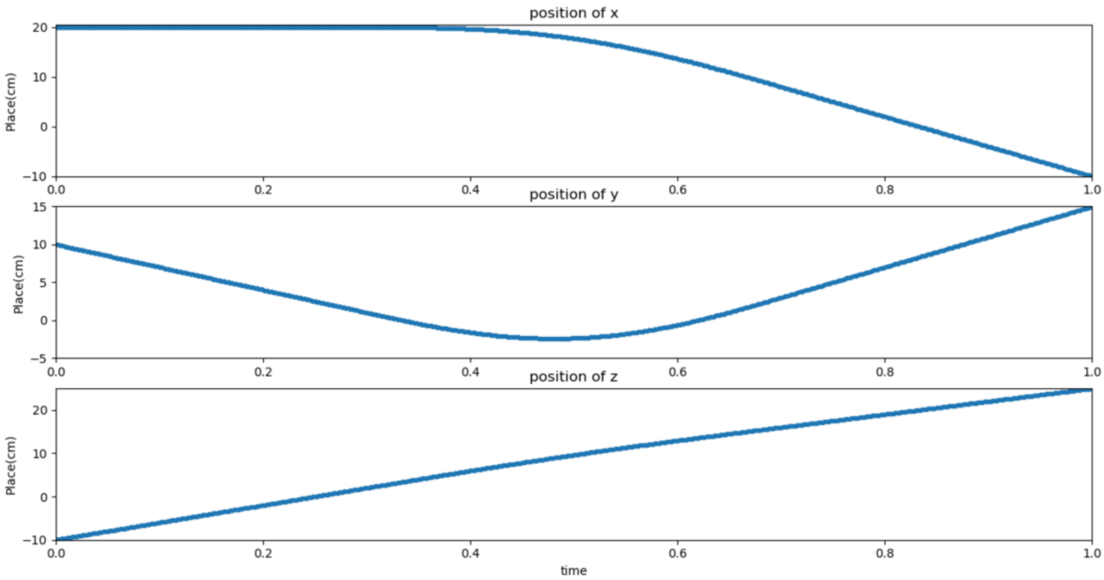
軸座標軌跡規劃曲線圖(左下 Y 右下 X)俯視



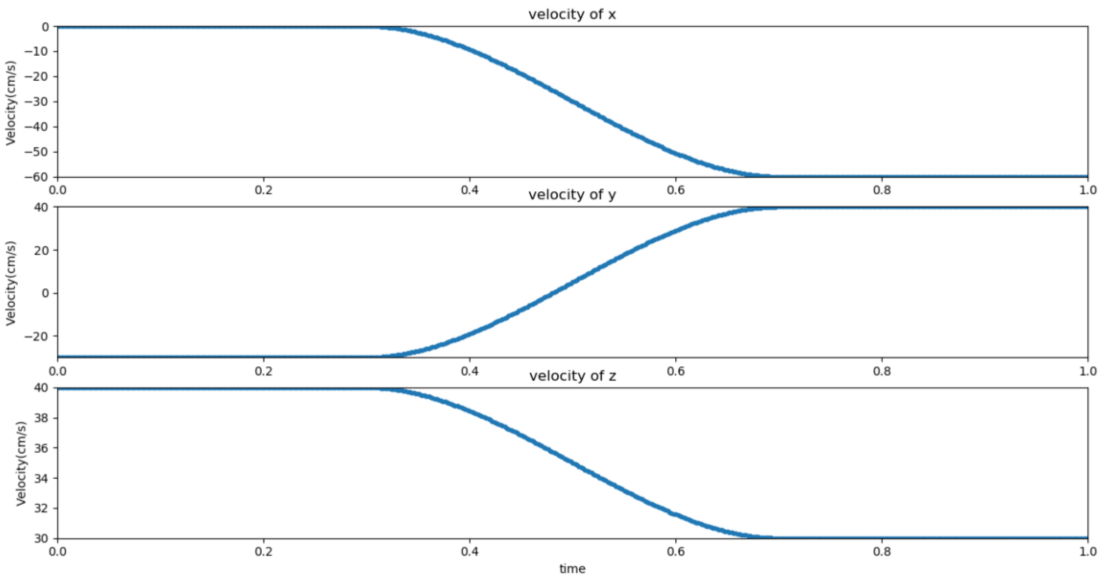
其中一軸的持續變化是以帶狀箭頭表示

Cartesian Move

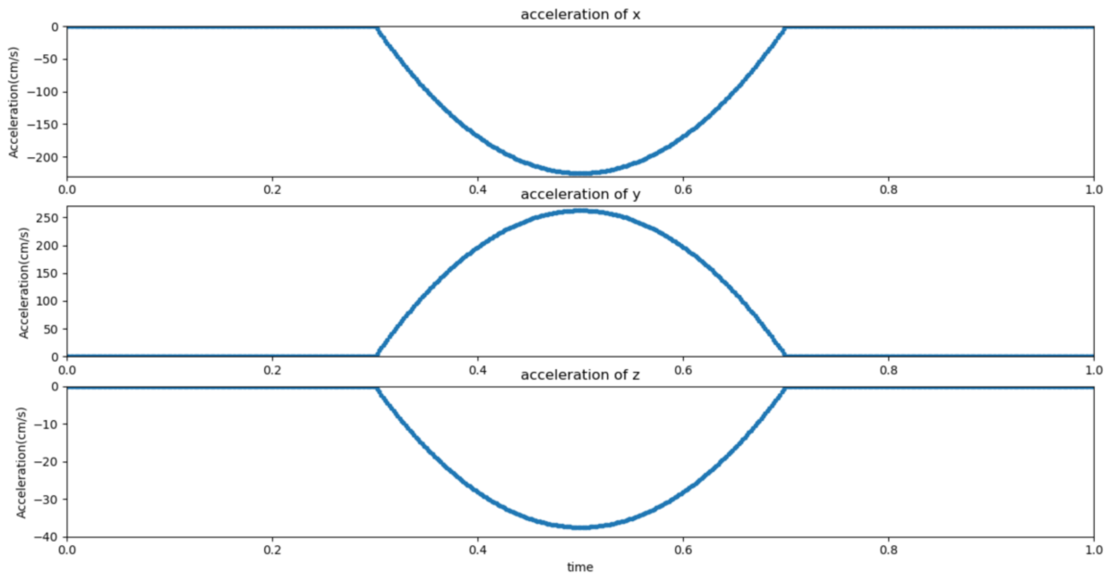
末端點位置圖



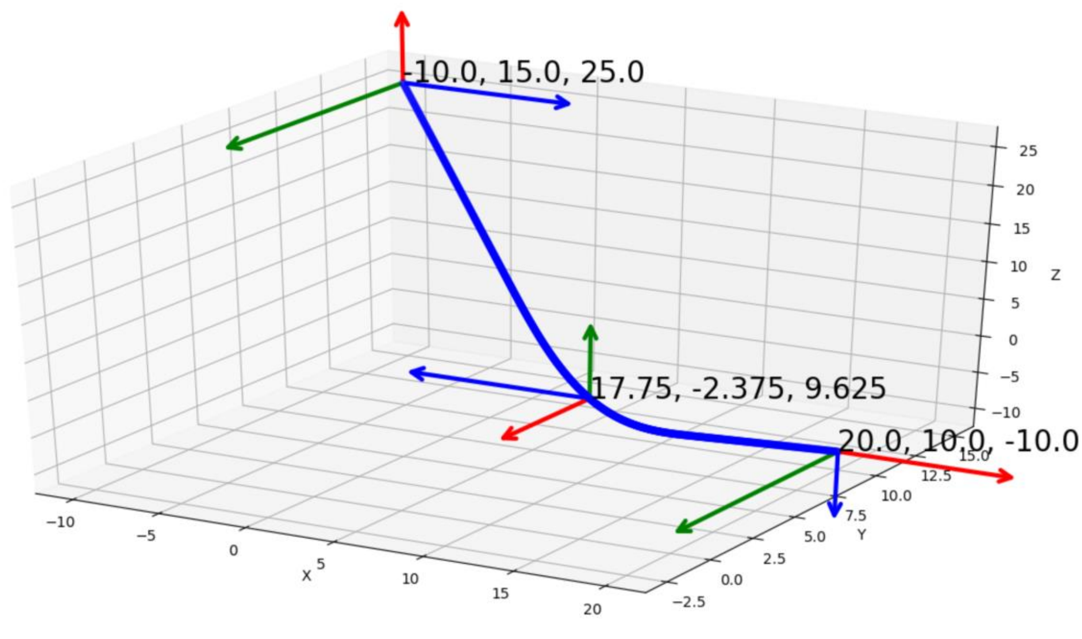
末端點速度圖



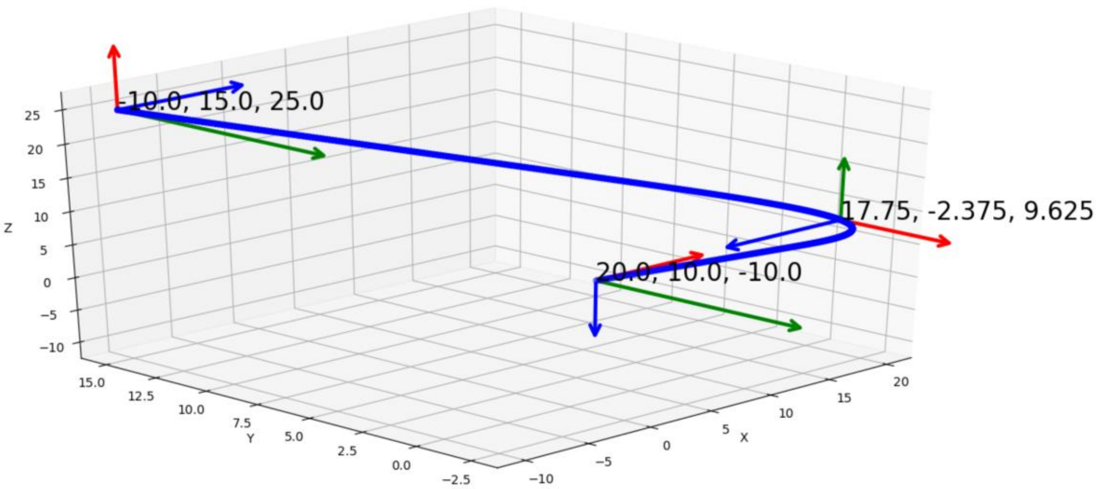
末端點加速度圖



卡式座標軌跡規劃曲線圖(左下 X 右下 Y)



卡式座標軌跡規劃曲線圖(左下 Y 右下 X)



卡式座標其中一軸持續變化尚未完成，這邊就沒顯示出來

五、兩種軌跡規劃的優缺點

首先，Joint Move 的優點是可以明確知道每個軸於每點的狀態，由軸變數圖可知哪個軸角度變化較多、哪個軸幾乎不動；由軸速度圖可知哪個軸轉速快，速度越快慣量越大；由軸加速度圖可知哪個軸受力驅動大，若連接該軸的是長桿件則要避免加速度變化量劇烈起伏，急轉或急停都會對手臂造成巨大負擔。缺點是無法一眼看出機械手臂的末端點位置，要經過換算才能知道 X、Y、Z 實際位置，且要避免路徑走到奇異點。

Cartesian Move 部分，優點是一眼就可知道手臂末端點狀態與座標軸的旋轉角度，由末端點位置圖可知手臂末端點於三維空間的移動路徑，末端點速度圖與末端點加速度圖可知末端點移動狀態，若手臂夾取的是脆弱物品，則要避免巨大的速度與加速度變化。缺點是要隨時注意各軸有無超出角度限制，雖然在起始與終點的手臂姿態符合限制，但隨著路徑移動，各軸角度隨時在變，末端點雖路徑平滑，但其實各軸角度要變化劇烈才能符合此路徑的情況是有可能發生的，若在現實中，沒做好模擬就運行的話是有可能造成手臂損壞。