

## Project 2 洪廷維 L091074

### 一、介面說明

1. 開發環境：MATLAB
2. 如何執行：開啟 main.m 並執行

### 二、程式架構說明

1. The project contains : 1 main program (main) , 2 classes (PATH, RVM2) , 7 functions (config2noap, joint2noap, noap2config, noap2joint, rad2deg0, numercial\_diff, Print)

2. main.m

主程式，定義題目的參數，控制專題流程

3. PATH.m

該類別可用於產生 robot 類別之軸、末端點軌跡，詳細說明於程式碼中，以下會著重介紹

- (1) Arguments of constructor

共五個，依序為：

- a. Desired points

n 個 noap matrix，本專題即{P1, P2, P3}

- b. Desired points' time

設定 Desired points 之抵達時間，本專題即 [0, 0.5, 1]

- c. Size of transition portion

$t_{acc}$ ，本專題即 [0, 0.2, 0]

- d. robot 類別

本專題即 RVM2

- e. method

用於決定使用 Joint move method 或 Cartesian move method

- (2) properties

- a. point

為一結構，用以儲存 Desired points 之相關資料

- b. path

為一結構，用以儲存規劃之軌跡，共 7 種，如下表

path(A).B			
A \ B	joint	config	noap
	$(\theta_1 \sim \theta_6)$	$(x, y, z, \phi, \theta, \psi)$	
1 (position)	O	O	O
2 (velocity)	O	O	X
3 (acceleration)	O	O	X

- (3) methods

- a. PATH

PATH.m 的 constructor

b. path\_planning

即講義 Polynomial Path Planning 方法，但沒有限定 $\psi$ 只在 transition portion 改變

4. RVM2.m

定義 RVM2 相關之函數、參數，包含專題一之 forward kinematic 和 inverse kinematic，透過更改參數可拓展至任意的、基於 D-H model 的 n link robot

5. functions

(1) config2noap, joint2noap, noap2config, noap2joint, rad2deg0

用於 joint, config, noap 之間的轉換

(2) numercial\_diff

數值微分，用來計算軌跡之速度和加速度

(3) Print

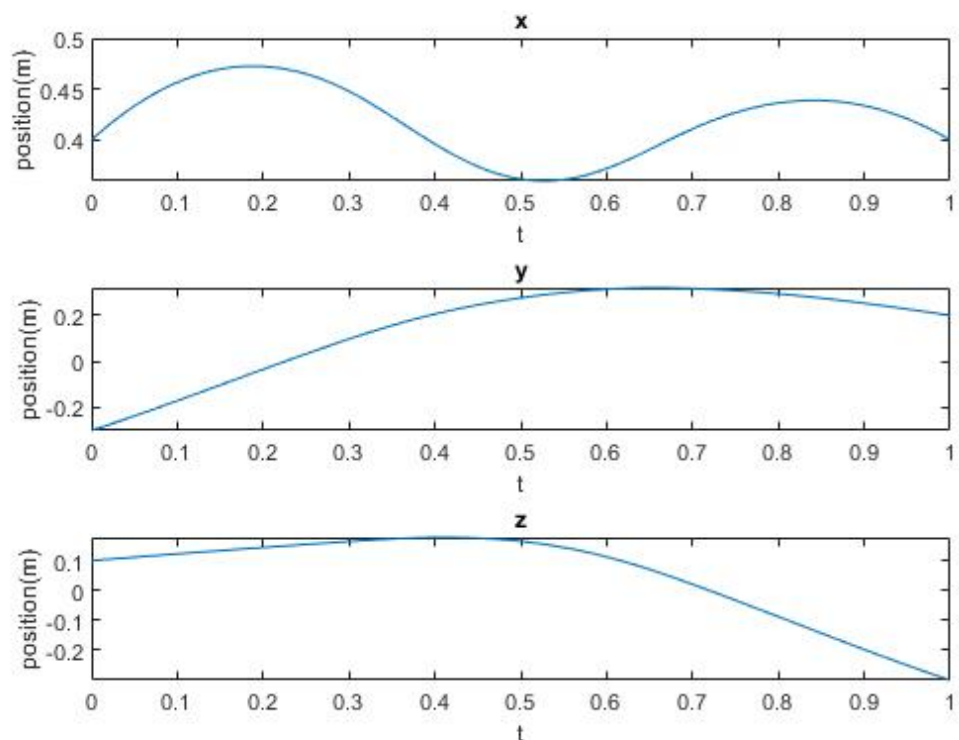
輸出結果圖

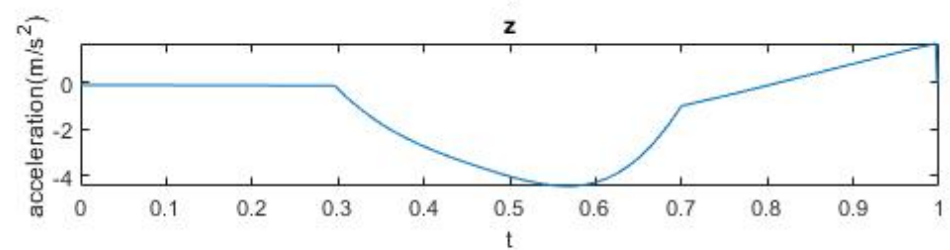
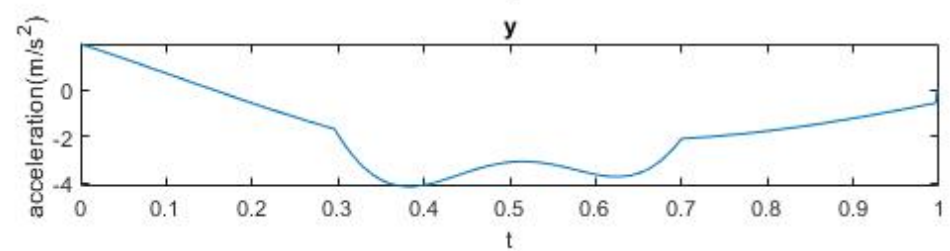
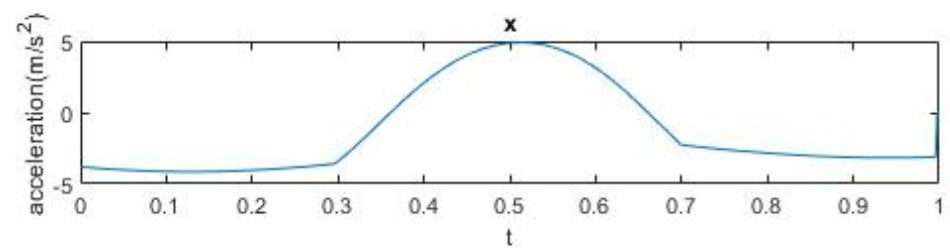
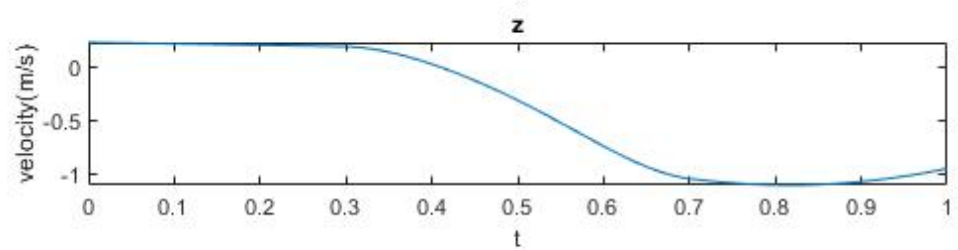
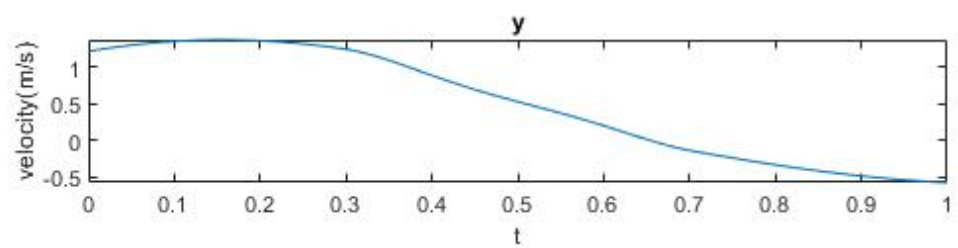
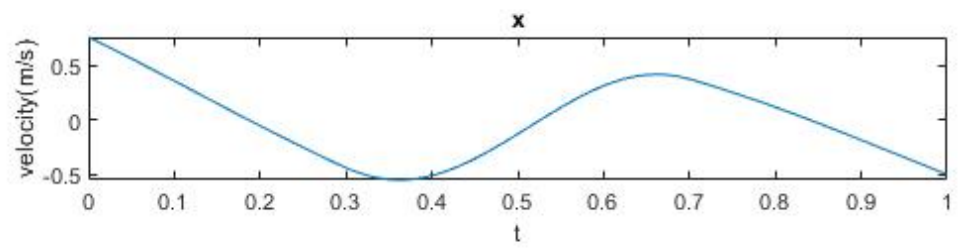
三、數學運算說明

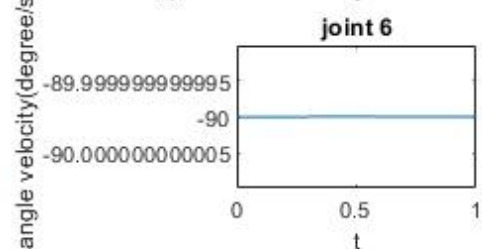
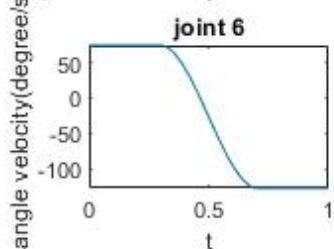
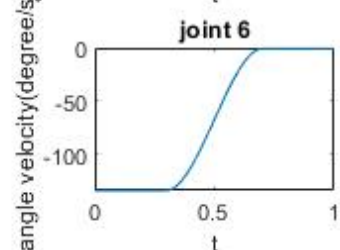
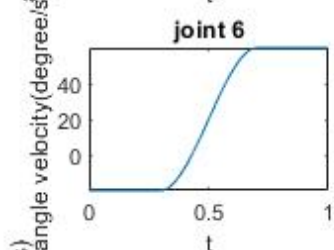
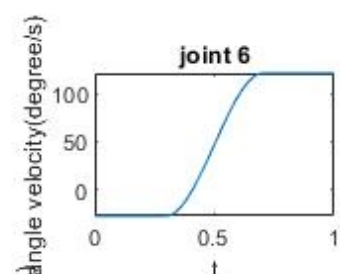
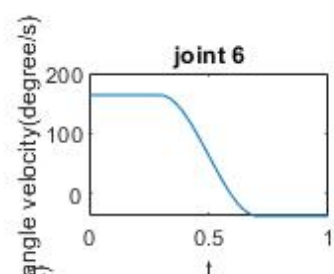
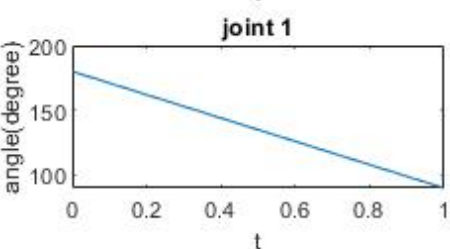
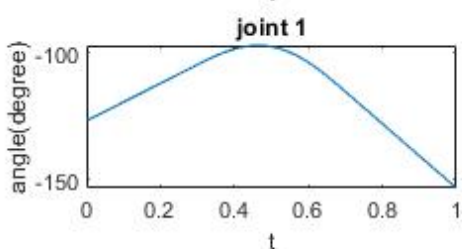
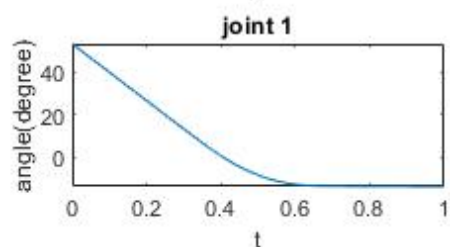
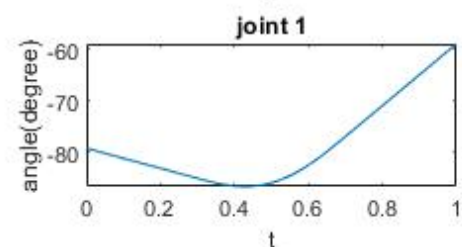
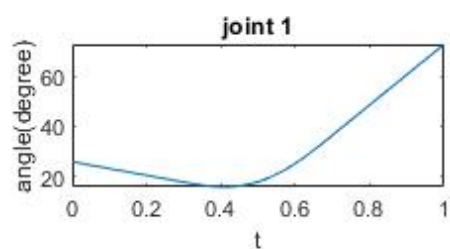
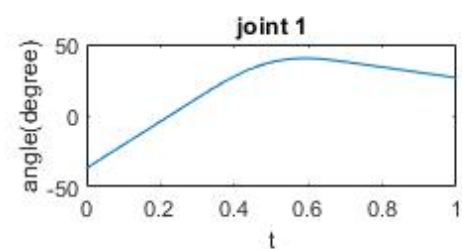
本專題主要之數學運算為 Polynomial Path Planning 之推導，因 HW4 的(a)推導過，故省略

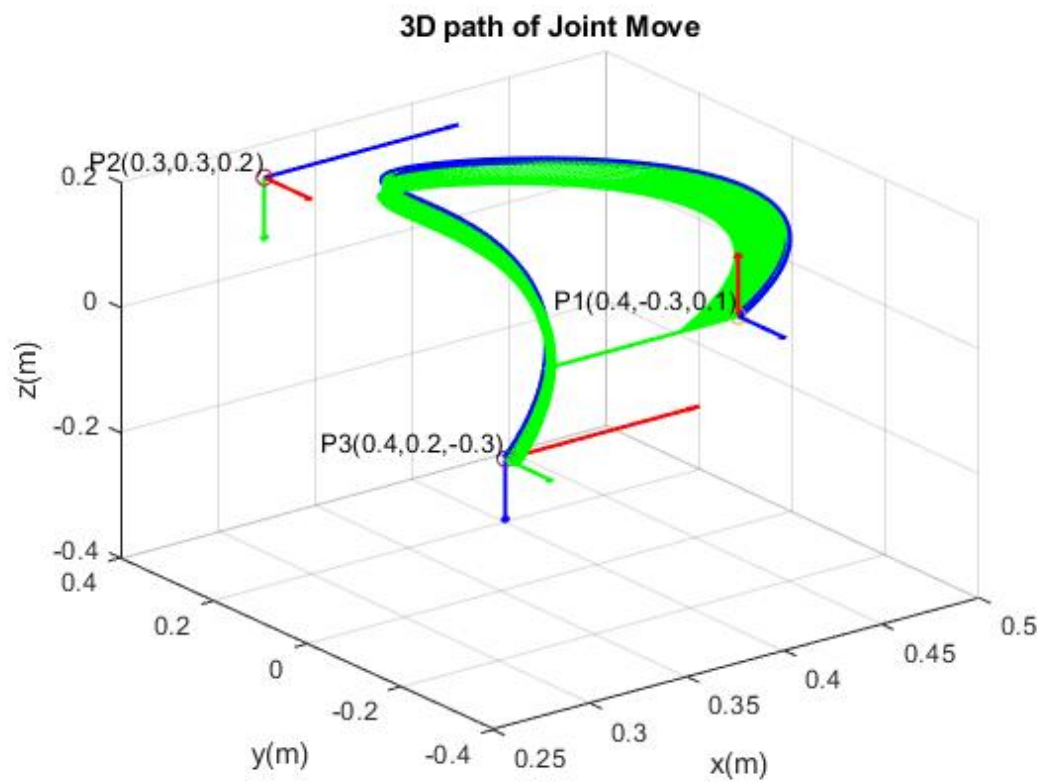
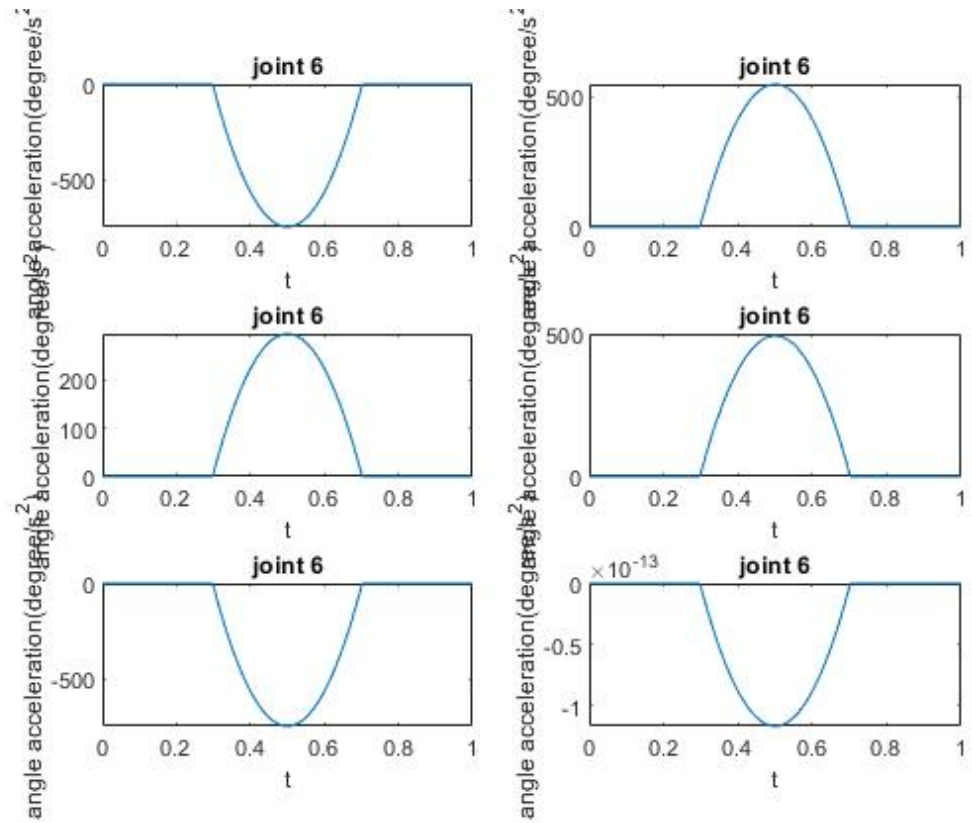
四、軌跡規劃曲線圖結果

1. Joint move

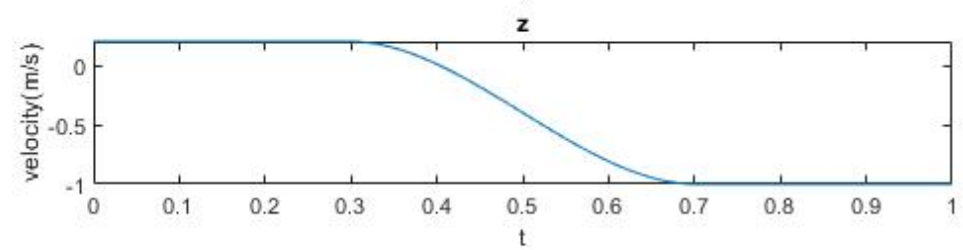
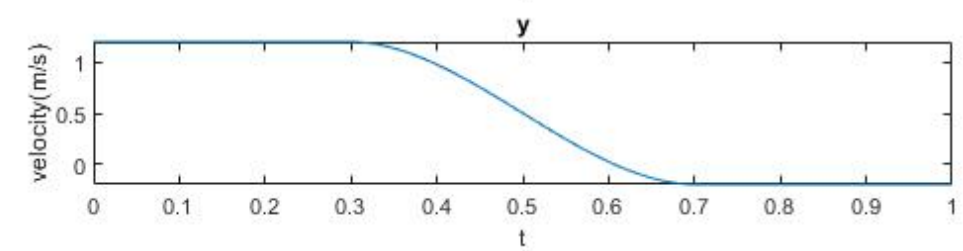
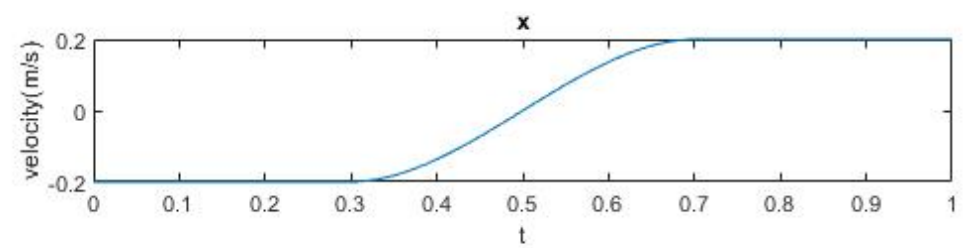
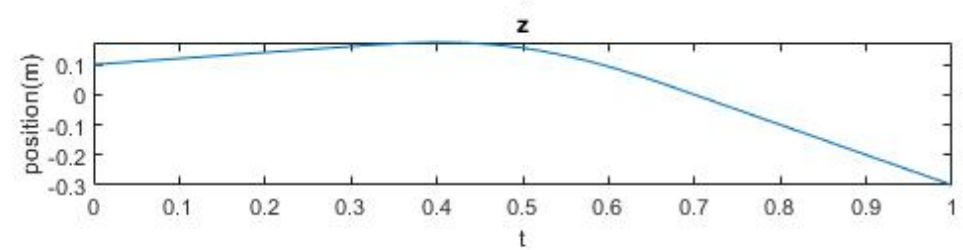
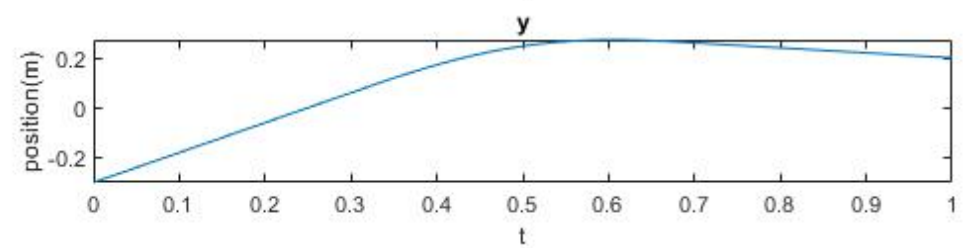
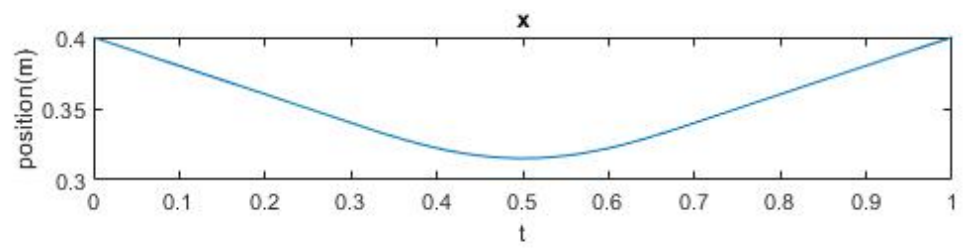


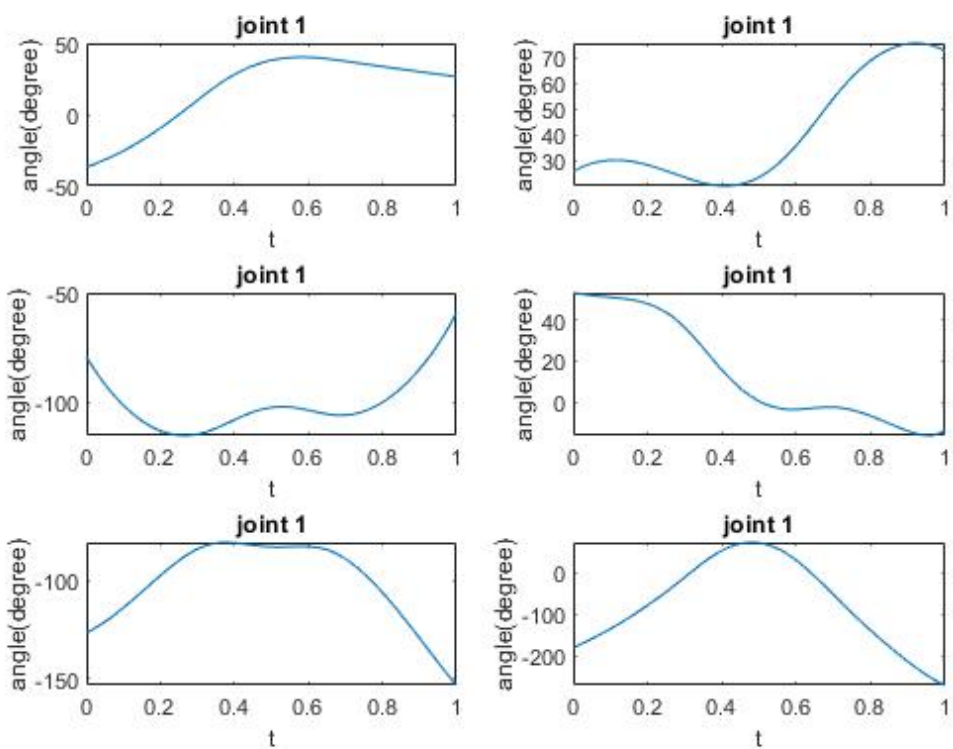
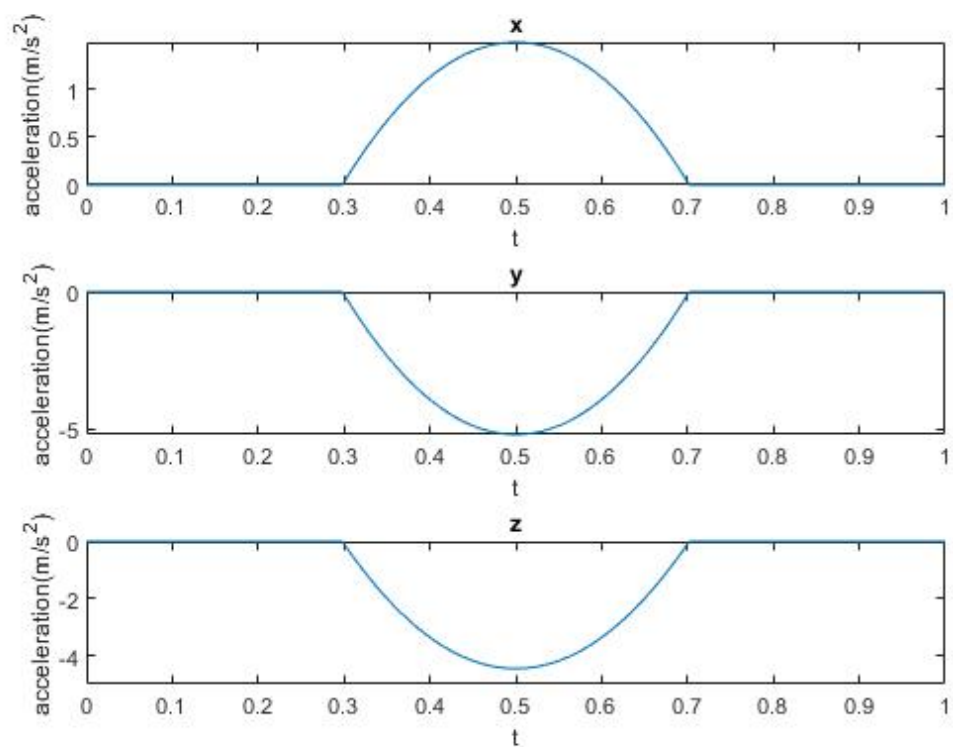




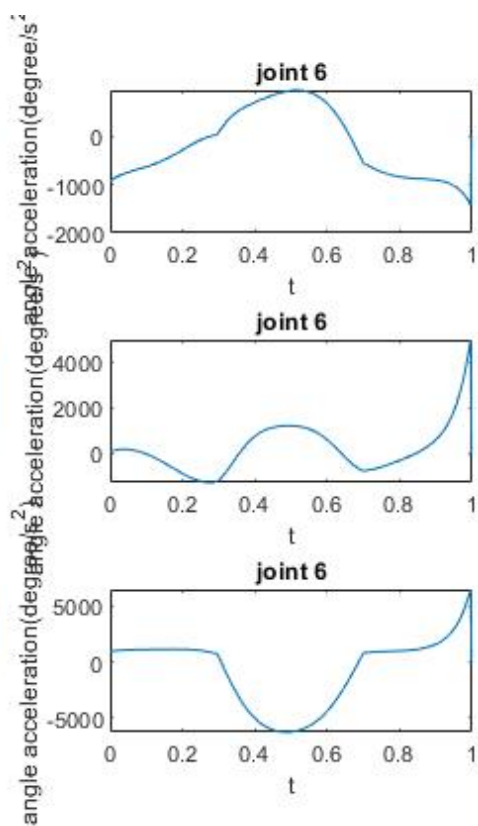
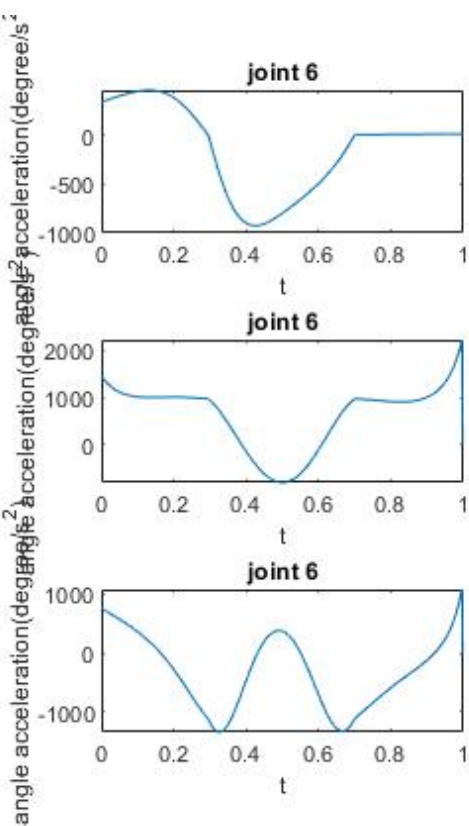
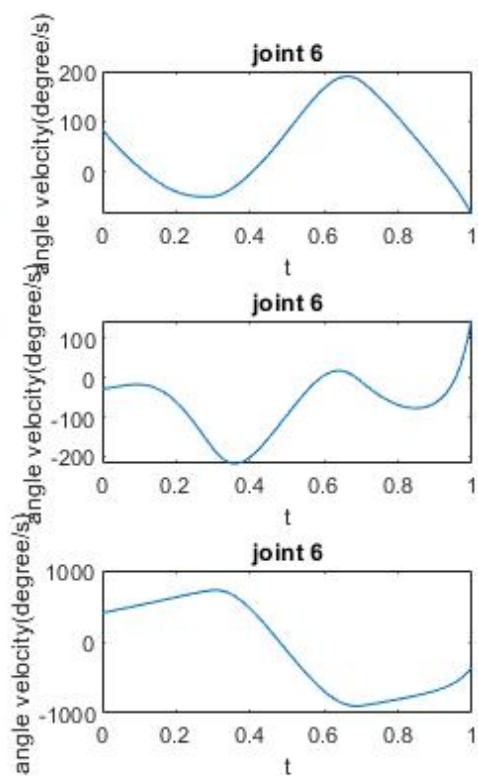
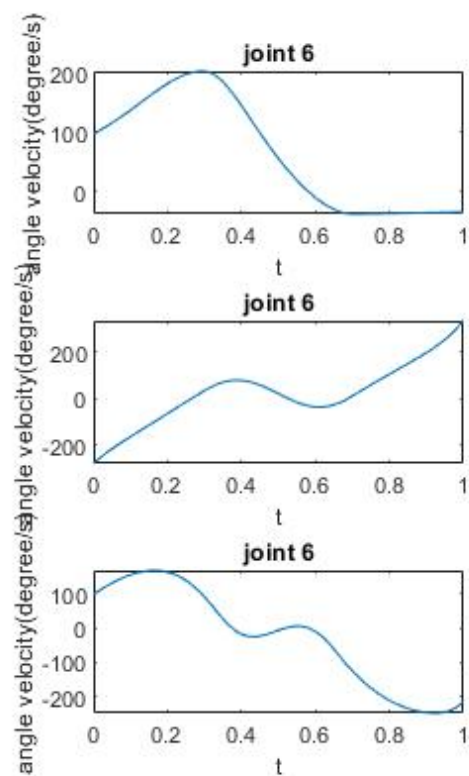


2. Cartesian move

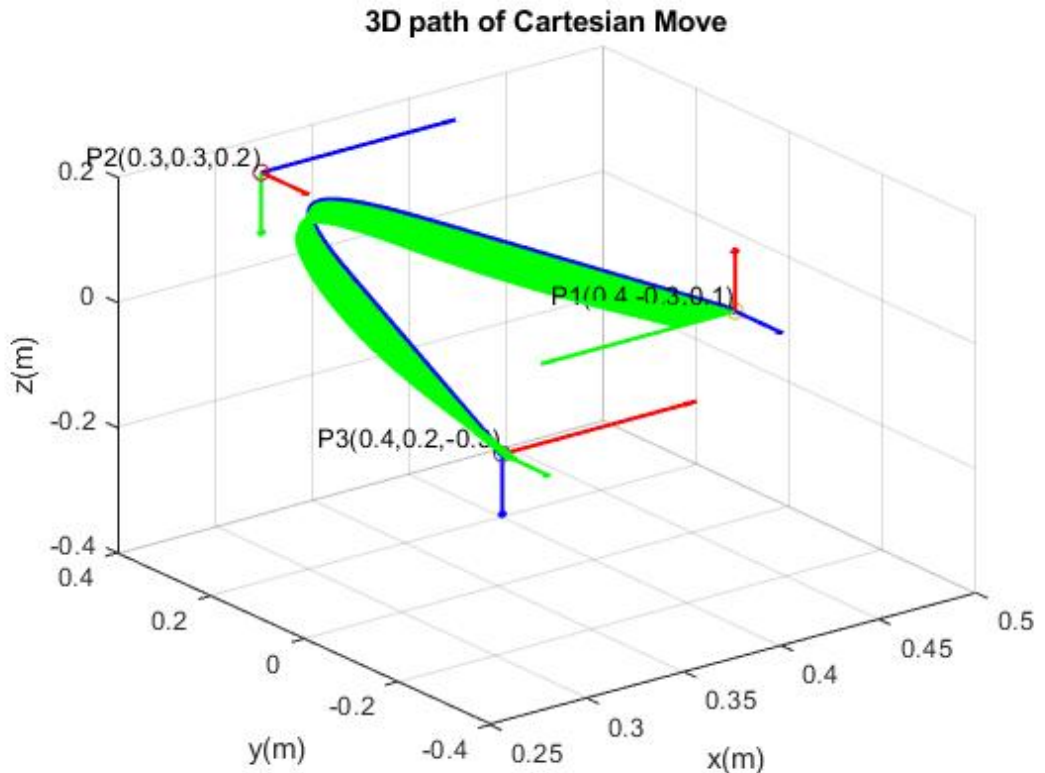












### 3. 說明

- (1) Joint move method 是對軸軌跡作規劃再透過 forward kinematic 計算末端點之軌跡，雖然計算較簡單且不會有奇異點問題，但因為我們給予機械手臂任務時，通常所關心的是末端點的軌跡，而該方法無法精確地給出點之間（例如 P1 和 P2 之間）的軌跡，因此對於末端點軌跡之精確度有要求的任務將不適用，相反地，若只是想將機械臂末端點的 configuration 移至特定點，該方法將優於 Cartesian move method。
- (2) Cartesian move method 是對末端點作軌跡規劃再透過 inverse kinematic 計算軸軌跡，雖計算較複雜、軸軌跡可能不連續且具有奇異點問題，但該方法可以直接對末端點軌跡作設計（可從 2.之 3D path 結果圖看出 Joint move method 跟 Cartesian move method 之差異），因此更能夠勝任對末端點軌跡之精確度有要求的任務。
- (3) 本專題之軌跡於  $t = 0.3$  時發生奇異點，實作中是將  $t_{acc}$  增加一個 time step，使軌跡於  $t = 0.298$  時就開始轉彎（進入 transition portion）以避開奇異點。