

專題二 結果報告

1. 開發平台:MATLAB
2. 如何執行: 一鍵執行
3. 程式運行流程及核心程式碼說明:

3.1 Joint move (joint.m)

將三個點的 Homogenous transformation 透過 Project 1 的逆向運動學程式，轉換為各軸的角度 (選擇其中一組未超出範圍的即可)，如圖一。

```

1  clc;clear;
2  A=[-100.4577  70.6108  48.3997      0  60.9896  29.2746]; %%利用 project1 輸入齊次矩陣 求得逆向運動學的解
3  B=[ -52.1158  -1.4358  30.2060  58.6166  11.4781  3.2859]; %%利用 project1 輸入齊次矩陣 求得逆向運動學的解
4  C=[  0.0955  65.7969  14.3196  15.3377  -20.1730  29.5160]; %%利用 project1 輸入齊次矩陣 求得逆向運動學的解
5  %%%初始化
6  theta=zeros([6,501]);
7  v=zeros([6,501]);
8  a=zeros([6,501]);
9  t1=0:0.002:0.3;
10 t2=0.3:0.002:0.7;
11 t3=0.7:0.002:1.0;
12 %%%

```

圖一

接著針對每個軸進行一次式線段及四次式線段做軌跡規劃(詳細公式推導在後方)，如圖二、圖三、圖四。

```

14 %%% 第一段直線段 t= 0.0~0.3
15 for t=t1
16     for i=1:6%六軸
17         time=int32(500*t+1); %% matlab index 由 1 開始且為整數 因此將 t= 0.0~0.3 映射到 time = 1~151
18
19         theta(i,time)=(B(i)-A(i))*t/0.5+A(i); % 推導出的角度公式
20         v(i,time)=(B(i)-A(i))/0.5;% 推導出的速度公式
21         a(i,time)=0; % 推導出的加速度公式，角度 為1次式 故為0
22     end
23 end

```

圖二 、 t=0.0~0.3 時的一次式線段軌跡規劃

```

26 for t=t2 %第二段曲線段 t= 0.3~0.7
27     for i=1:6%六軸
28         time=int32(500*t+1); % t= 0.0~0.3 映射到 time = 151~351
29
30         %推導出來的公式 詳情請看 pdf 檔
31         h=(t-0.5+0.2)/0.4;
32         a0=(B(i)-A(i))*0.3/0.5+A(i);
33         a1=(B(i)-a0)/0.5;
34         a3=2*(a0-B(i))+0.8*(C(i)-B(i));
35         a4=-0.5*a3;
36         theta(i,time)=a0+a1*h+a3*h^3+a4*h^4;
37         v(i,time)=(a1+3*a3*h^2+4*a4*h^3)/0.4;
38         a(i,time)=(6*a3*h+12*a4*h^2)/0.16;
39         %%%
40     end
41 end

```

圖三 、 t=0.3~0.7 時的四次式線段軌跡規劃

```

45 for t=t3 %第三段直線段 t= 0.7~1.0
46     for i=1:6 %六軸
47         time=int32(500*t+1); % t= 0.7~1.0 映射到 time = 351~501
48
49         theta(i,time)=B(i)+(C(i)-B(i))*(t-0.5)/0.5;% 推導出的角度公式
50         v(i,time)=(C(i)-B(i))/0.5;% 推導出的速度公式
51         a(i,time)=0;% 推導出的加速度(位置)公式，角度為1次式故為0
52     end
53 end

```

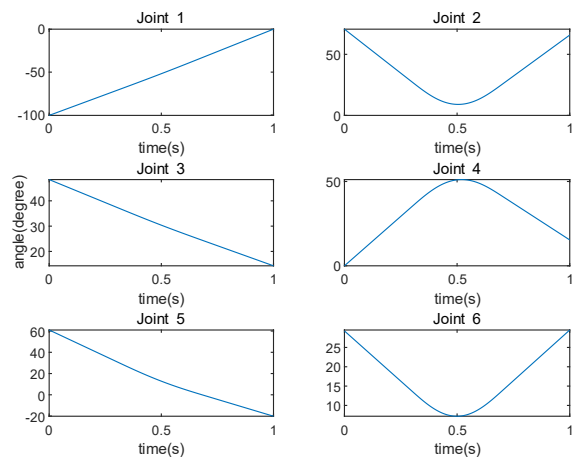
圖四、t=0.7~1.0時的一次式線段軌跡規劃

畫出規劃後的各軸角度、速度、加速度變化曲線，如圖五、圖六、圖七。

```

55 figure(1) %各軸的角度變化曲線
56 for i=1:6
57     subplot(3,2,i);
58     txt = ['Joint ' int2str(i)];
59     plot(0:0.002:1,theta(i,:))
60     subtitle(txt);
61     if i== 3
62         ylabel('angle(degree)')
63     end
64     xlabel('time(s)')
65 end

```

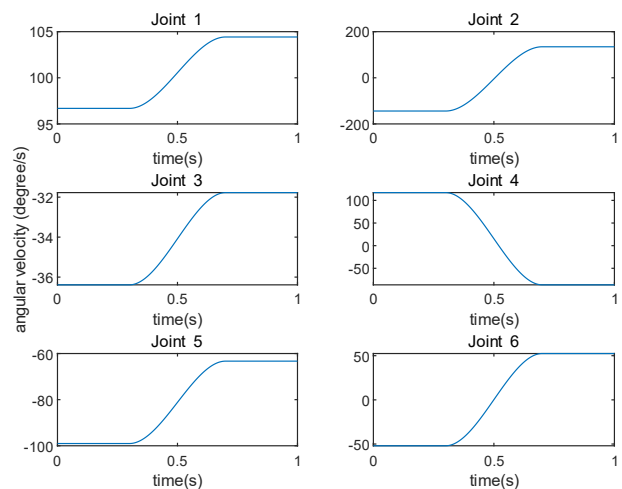


圖五、各軸角度 程式（左）與結果（右）

```

67 figure(2) %各軸的速度變化曲線
68 for i=1:6
69     subplot(3,2,i);
70     txt = ['Joint ' int2str(i)];
71     plot(0:0.002:1,v(i,:))
72     subtitle(txt);
73     if i== 3
74         ylabel('angular velocity (degree/s)')
75     end
76     xlabel('time(s)')
77 end

```

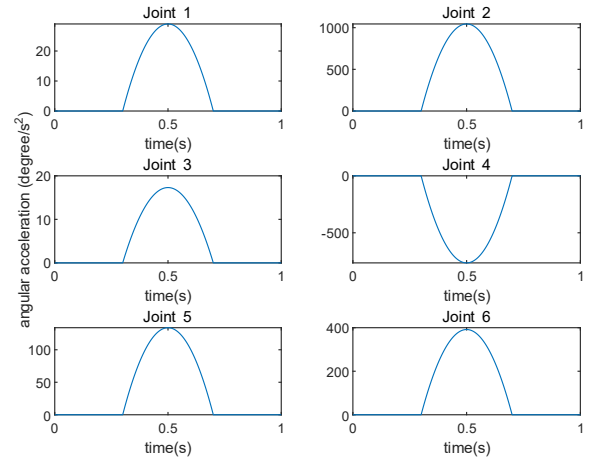


圖六、各軸速度 程式（左）與結果（右）

```

79 figure(3)%各軸的加速度變化曲線
80 for i=1:6
81     subplot(3,2,i);
82     txt = ['Joint ' int2str(i)];
83     plot(0:0.002:1,a(i,:))
84     subtitle(txt);
85     xlabel('time(s)')
86     if i== 3
87         ylabel('angular acceleration (degree/s^2)')
88     end
89 end

```



圖七 、 各軸加速度 程式（左）與結果（右）

建立 DH-table 並初始化參數，如圖八。

```

93 %將剛才計算出的各軸的角度變化 透過順向運動學轉換成 位置變化與俯仰角
94 % DH-table
95 d1=0.0; d2=0; d3=14.9; d4=43.3; d5=0; d6=0;
96 a1=0.0; a2=43.2; a3=-2; a4=0; a5=0; a6=0;
97 alpha1=-90; alpha2=0; alpha3=90; alpha4=-90; alpha5=90; alpha6=0;
98 d= [d1;d2;d3;d4;d5;d6];
99 a=[a1;a2;a3;a4;a5;a6];
100 alpha=[alpha1;alpha2;alpha3;alpha4;alpha5;alpha6];
101 % 參數初始化
102 x=zeros(1,501);
103 y=zeros(1,501);
104 z=zeros(1,501);
105 %approach vector
106 xaf=zeros(1,501);
107 yaf=zeros(1,501);
108 zaf=zeros(1,501);

```

圖八 、 建立 DH-table 並初始化參數

透過順項運動學計算各時間的手臂末端的位置與方向，如圖九。

```

118 for i = 1:501 %求得各個時間的齊次矩陣 並得知末端執行器的位置及姿態
119     An=zeros([4 4 6]);
120     T=eye(4);
121     for j=1:6
122         An(:,j)=DH(theta(j,i),d(j),a(j),alpha(j));
123         T=T*An(:,j);
124     end
125     %n o a p 的 p (位置)
126     x(i)=T(1,4);
127     y(i)=T(2,4);
128     z(i)=T(3,4);
129     %n o a p 的 approach (只顯示這組 不然畫面太亂了)
130     xaf(i)=T(1,3);%approach vector X分量
131     yaf(i)=T(2,3);%approach vector Y分量
132     zaf(i)=T(3,3);%approach vector Z分量
133 end

```

圖九 、 各時間的手臂末端的位置與方向

最後畫出卡式空間下的 3D 軌跡圖，如圖十。

```

143 %畫出手臂末端在空間中的軌跡與姿態
144 figure(4)
145 hold on ;
146 plot3(5,-55,-60,'o');%點A
147 text(5,-55,-60,'A (5,-55,-60)')
148 plot3(50,-40,40,'o');%點B
149 text(50,-40,40,'B (50,-40,40)')
150 plot3(60,15,-30,'o');%點C
151 text(60,15,-30,'C (5,-55,-60)')
152
153 plot3(x,y,z);% 軌跡
154 %quiver3(x,y,z,xnf,yzf,'r','LineWidth',2)
155 %quiver3(x,y,z,xof,yof,zof,'g','LineWidth',2)
156 quiver3(x,y,z,xaf,yaf,zaf,'LineWidth',2) % 末端點方向 approach vector
157
158 xlabel('x (cm)')
159 ylabel('y (cm)')
160 zlabel('z (cm)')
161 view([-135,15]) % 圖的視角
162 grid on;

```

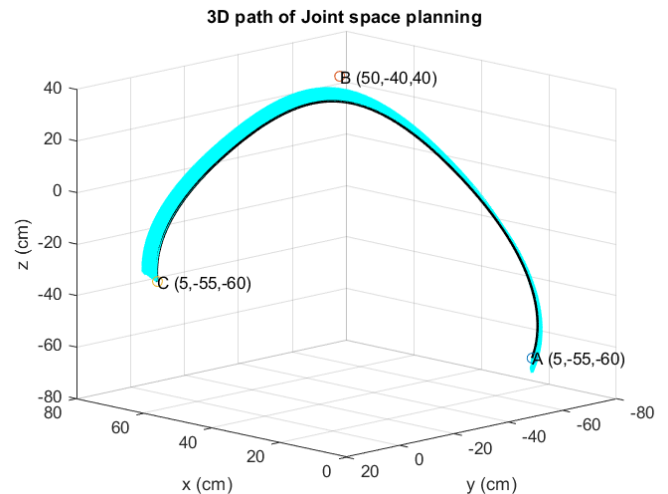


圖 十 、 程式（左）與卡式空間下的 3D 軌跡圖（右）

3.2 Cartesian move (Cartesian.m)

透過 `rotm2eul` 計算 ZYZ 的 Euler angle，並列出三點相對應的 X、Y、Z、phi、theta、psi，並初始化，如圖十一。

```

1 clc;clear;
2 %透過 rotm2eul 計算 ZYZ 的euler angle
3 AA=rotm2eul([0.64 0.77 0;0.77 -0.64 0;0 0 -1;],'ZYZ')/pi*180;
4 BB=rotm2eul([0.87 -0.1 0.48;0.29 0.9 -0.34;-0.4 0.43 0.81;],'ZYZ')/pi*180;
5 CC=rotm2eul([0.41 -0.29 0.87;0.69 0.71 -0.09;-0.6 0.64 0.49;],'ZYZ')/pi*180;
6 %
7 % 初始化位置與 euler angle
8 A=[5 -55 -60 AA];
9 B=[50 -40 40 BB];
10 C=[60 15 -30 CC];
11 %
12 %初始化參數儲存空間
13 postion=zeros([501,6]); % 位置與 euler angle
14 v=zeros([501,6]);% 位置與 euler angle 的速度
15 a=zeros([501,6]);% 位置與 euler angle 的加速度
16 xaf=zeros(1,501);%approach vector X分量
17 yaf=zeros(1,501);%approach vector Y分量
18 zaf=zeros(1,501);%approach vector Z分量
19
20 % 時間
21 t1=0:0.002:0.3;
22 t2=0.3:0.002:0.7;
23 t3=0.7:0.002:1.0;

```

圖 十一

接著針對 X、Y、Z、phi、theta、psi 進行一次式線段及四次式線段做軌跡規劃(詳細公式推導在後方)，如圖十二、圖十三、圖十四。

```

27 for t=t1 %第一段直線段 t= 0.0~0.3
28     for i=1:6 % X Y Z phi theta psi 規劃
29         time=int32(500*t+1);%% matlab index 由 1 開始且為整數 因此將 t= 0.0~0.3 映射到 time = 1~151
30
31         postion(time,i)=(B(i)-A(i))*t/0.5+A(i);% 推導出的位置與角度公式
32         v(time,i)=(B(i)-A(i))/0.5;% 推導出的速度公式
33         a(time,i)=0;% 推導出的加速度公式，角度與位置 為1次式 故為0
34     end
35 end

```

圖 十二 、 t=0.0~0.3 時的一次式線段軌跡規劃

```

39 for i=1:6 % X Y Z phi theta psi 規劃
40     time=int32(500*t+1); % t= 0.0~0.3 映射到 time = 151~351
41     %推導出來的公式 詳情請看 pdf 檔
42     h=(t-0.5+0.2)/0.4;
43     a0=(B(i)-A(i))*0.3/0.5+A(i);
44     a1=(B(i)-a0)/0.5;
45     a3=2*(a0-B(i))+0.8*(C(i)-B(i));
46     a4=-0.5*a3;
47     postion(time,i)=a0+a1*h+a3*h^3+a4*h^4;
48     v(time,i)=(a1+3*a3*h^2+4*a4*h^3)/0.4;
49     a(time,i)=(6*a3*h+12*a4*h^2)/0.16;
50     %%%
51 end
52 end

```

圖 十三 、 t=0.3~0.7 時的四次式線段軌跡規劃

```

56 for t=t3 %第三段直線段 t= 0.7~1.0
57     for i=1:6 % X Y Z phi theta psi 規劃
58         time=int32(500*t+1); % t= 0.7~1.0 映射到 time = 351~501
59
60         postion(time,i)=B(i)+(C(i)-B(i))*(t-0.5)/0.5;%推導出的角度與位置公式
61         v(time,i)=(C(i)-B(i))/0.5;% 推導出的速度公式
62         a(time,i)=0;% 推導出的加速度公式，角度與位置 為1次式 故為0
63     end
64
65 end

```

圖 十四 、 t=0.7~1.0 時的一次式線段軌跡規劃

透過 `eul2rotm` 將 ZYZ 的 Euler angle 轉換為齊次矩陣，並取得 approach vector
如圖十五。

```

69     phi=postion(:,4);
70     theta=postion(:,5);
71     psi=postion(:,6);
72
73     for i = 1:501
74         eul = [phi(i) theta(i) psi(i)]/180*pi;
75         rotm = eul2rotm(eul,'ZYZ'); %透過 eul2rotm 將 ZYZ 的euler angle 轉換成齊次矩陣
76         xaf(i) = rotm(1,3); %存取approach vector X分量
77         yaf(i) = rotm(2,3); %存取approach vector Y分量
78         zaf(i) = rotm(3,3); %存取approach vector Z分量
79     end

```

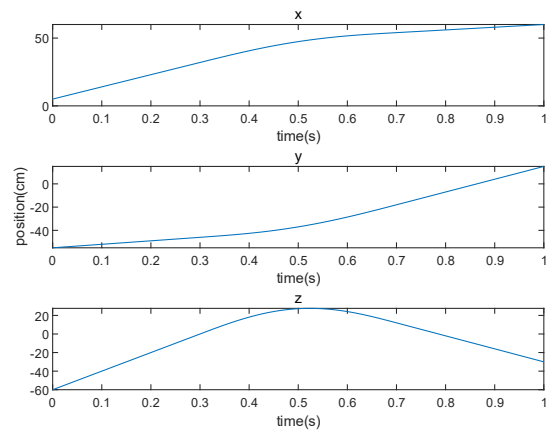
圖 十五

畫出規劃後的空間中位置、速度、加速度 變化曲線， 如圖十六、圖十七、圖十八。

```

83     figure(1)%圖1 空間中位置變化曲線
84     for i=1:3
85         subplot(3,1,i);
86         plot(0:0.002:1,postion(:,i))
87         subtitle(txt(i));
88         if i== 2
89             ylabel('position(cm)')
90         end
91         xlabel('time(s)')
92     end

```

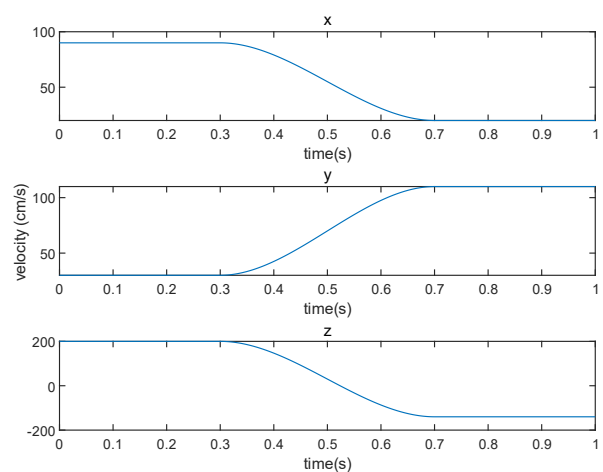


圖十六 、 空間中的位置 程式(左)與結果(右)

```

94     figure(2)%圖2 速度變化曲線
95     for i=1:3
96         subplot(3,1,i);
97         plot(0:0.002:1,v(:,i))
98         subtitle(txt(i));
99         if i== 2
100             ylabel('velocity (cm/s)')
101         end
102         xlabel('time(s)')
103     end

```

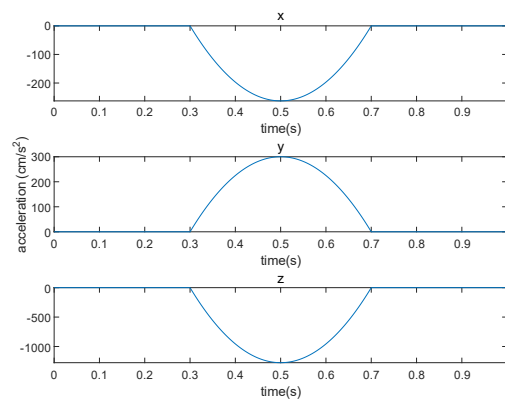


圖十七 、 空間中的速度 程式(左)與結果(右)


```

105 figure(3)%圖3 加速度變化曲線
106 for i=1:3
107     subplot(3,1,i);
108     plot(0:0.002:1,a(:,i))
109     subtitle(txt(i));
110     xlabel('time(s)')
111     if i==2
112         ylabel('acceleration (cm/s^2)')
113     end
114 end

```



圖十八、空間中的加速度 程式(左)與結果(右)

最後畫出手臂末端在空間中的軌跡與方向，如圖十九。

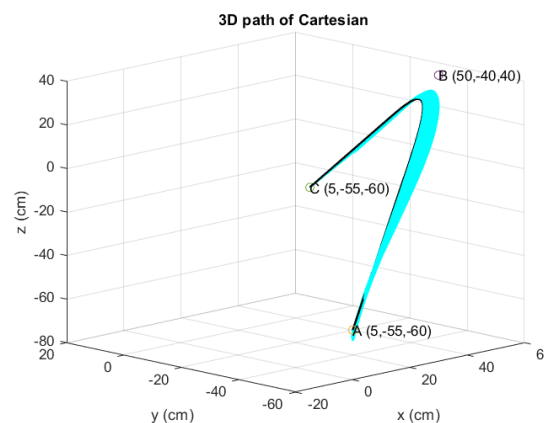
```

figure(4)%畫出手臂末端在空間中的軌跡與姿態
hold on;
plot3(position(:,1),position(:,2),position(:,3)); % 軌跡
quiver3(position(:,1),position(:,2),position(:,3),xaf,yaf,zaf,'LineWidth',2)% approach vector 方向

plot3(5,-55,-60,'o'); %點A
text(5,-55,-60,'A (5,-55,-60)')
plot3(50,-40,40,'o'); %點B
text(50,-40,40,'B (50,-40,40)')
plot3(60,15,-30,'o'); %點C
text(60,15,-30,'C (60,15,-30)')

xlabel('x (cm)')
ylabel('y (cm)')
zlabel('z (cm)')
view([-45,15]) %視角
axis([-20 60 -60 20 -80 40]) % 3D圖範圍

```



圖十九、程式(左)與卡式空間下的 3D 軌跡圖(右)

4. 數學運算說明

數學運算主要內容(雖有些差異但概念一致)，為方便計算先將 t 正規化成 h ，如圖二十。

5.2 Polynomial Path Planning

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

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

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

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

Where $h = \frac{t+t_{acc}}{2t_{acc}}$ for $-t_{acc} \leq t \leq t_{acc}$

5.2 Polynomial Path Planning

□ For linear portion

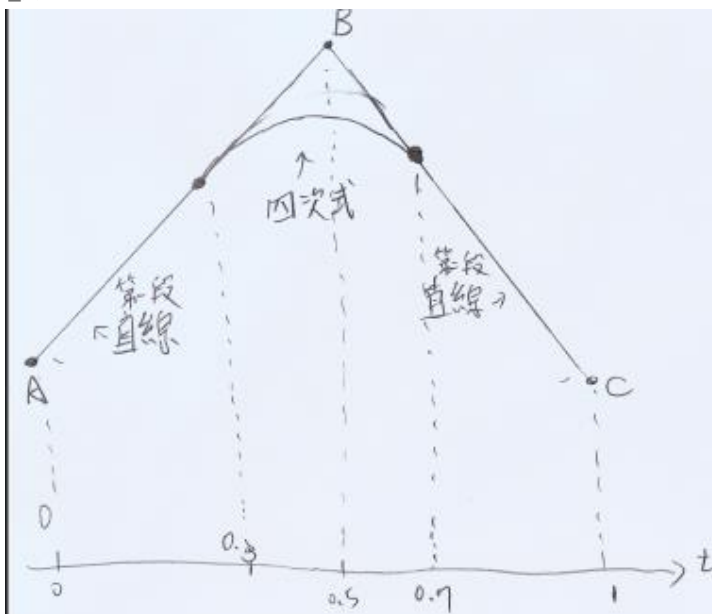
$$\begin{cases} q = \Delta C \cdot h + B \\ \dot{q} = \frac{\Delta C}{T} \\ \ddot{q} = 0 \end{cases} \quad h = \frac{t}{T}, t_{acc} \leq t \leq T - t_{acc}$$

□ Transition of different segments

$$\begin{aligned} T &\leftarrow T_{new} \\ A, B, C &\leftarrow B, C, D \\ \Delta B, \Delta C &\leftarrow \Delta C, \Delta D \end{aligned}$$

圖二十、軌跡規劃投影片

而詳細推導如圖二十一，由於起始加速度與中止加速度為 0，因此在 transition portion 的位置(角度)多項式可簡化成四次式，並且為了方便計算 t 正規化成 h 。



第一段直線

$$t = 0.0 \sim 0.3$$

$$q(t) = \frac{B-A}{0.5} \cdot t + A$$

$$\dot{q}(t) = \frac{B-A}{0.5}$$

$$\ddot{q}(t) = 0$$

第二段直線

$$t = 0.7 \sim 1.0$$

$$q(t) = B + \frac{C-B}{0.5} \cdot t$$

$$\dot{q}(t) = \frac{C-B}{0.5}$$

$$\ddot{q}(t) = 0$$

transition portion: $t = 0.3 \sim 0.7 \Rightarrow h = 0 \sim 1$

$$h = \frac{t-0.3}{0.4} \Rightarrow dh = \frac{dt}{0.4}$$

$$q(h) = a_0 + a_1 h + a_2 h^2 + a_3 h^3 + a_4 h^4$$

$$\dot{q}(h) = \frac{dq(h)}{dh} = a_1 + 2a_2 h + 3a_3 h^2 + 4a_4 h^3$$

$$\ddot{q}(h) = \frac{d\dot{q}(h)}{dh} = 2a_2 + 6a_3 h + 12a_4 h^2$$

$$\Rightarrow q(0) = \frac{B-A}{0.5} \cdot (0.3) + A = a_0 \quad \begin{matrix} t=0 \Rightarrow h=0 \\ t=0.5 \Rightarrow h=0.5 \end{matrix} \Rightarrow \dot{q}(0) = \frac{B-A}{0.5 \cdot 0.4}$$

$$\dot{q}(0) = \frac{B-A}{1.25} = \frac{0.4}{0.5} (B-A) = \frac{B-a_0}{0.5} = a_1$$

$$\ddot{q}(0) = 0 = 2a_2$$

$$q(1) = a_0 + a_1 + a_2 + a_3 + a_4 = B + \frac{C-B}{0.5} \cdot (0.7)$$

$$\dot{q}(1) = a_1 + 2a_2 + 3a_3 + 4a_4 = \frac{C-B}{1.25} \quad -①$$

$$\ddot{q}(1) = 0 = 2a_2 + 6a_3 + 12a_4 \Rightarrow a_4 = -\frac{1}{2}a_3 \quad -②$$

$$\textcircled{1} \wedge \textcircled{2} \Rightarrow a_3 + a_1 = (C-B) \cdot 0.8$$

$$\Rightarrow a_3 = -a_1 + (C-B) \cdot 0.8$$

$$= \frac{a_0-B}{0.5} + (C-B) \cdot 0.8$$

$$q(t) = a_0 + a_1 h + a_2 h^2 + a_3 h^3 + a_4 h^4$$

$$\Rightarrow \dot{q}(t) = (a_1 + 2a_2 h + 3a_3 h^2 + 4a_4 h^3) \cdot \frac{1}{0.4}$$

$$\ddot{q}(t) = (2a_2 + 6a_3 h + 12a_4 h^2) \cdot \frac{1}{0.4^2}$$

$$\text{where } h = \frac{t-0.3}{0.4}, t = 0.3 \sim 0.7$$

$$a_0 = \frac{B-A}{0.5} \cdot 0.3 + A \quad a_1 = \frac{B-a_0}{0.5}, a_2 = 0$$

$$a_3 = \frac{a_0-B}{0.5} + (C-B) \cdot 0.8, a_4 = -\frac{1}{2}a_3$$

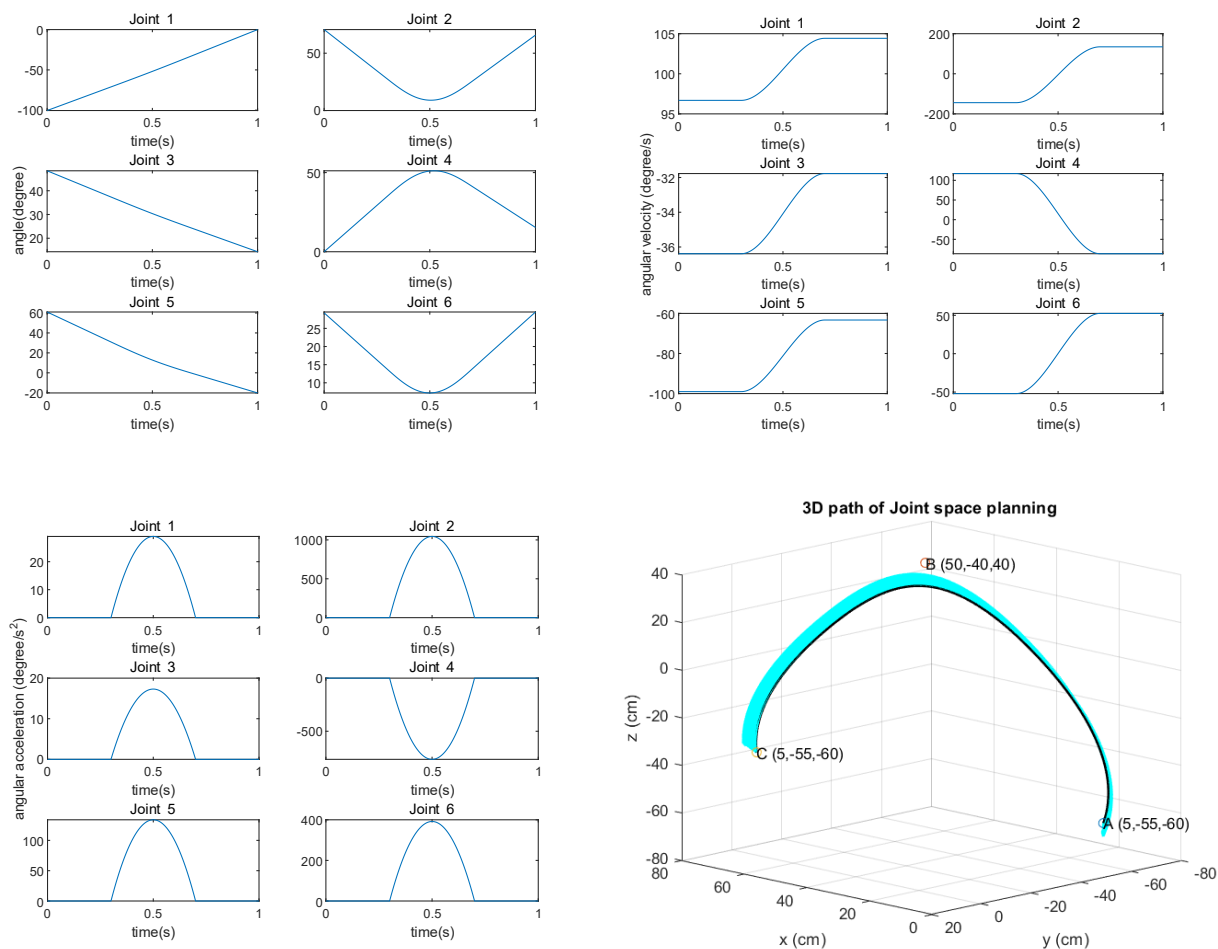
圖二十一、公式推導

在 Joint move 中 透過圖二十一公式規劃手臂第一軸到第六軸的轉動角度、速度、加速度，而在 Cartesian move 中 透過圖二十一公式規劃 手臂末端在卡式空間中的軌跡以及 Euler angle 的變化。

5. 結果示意圖

5.1 Joint move

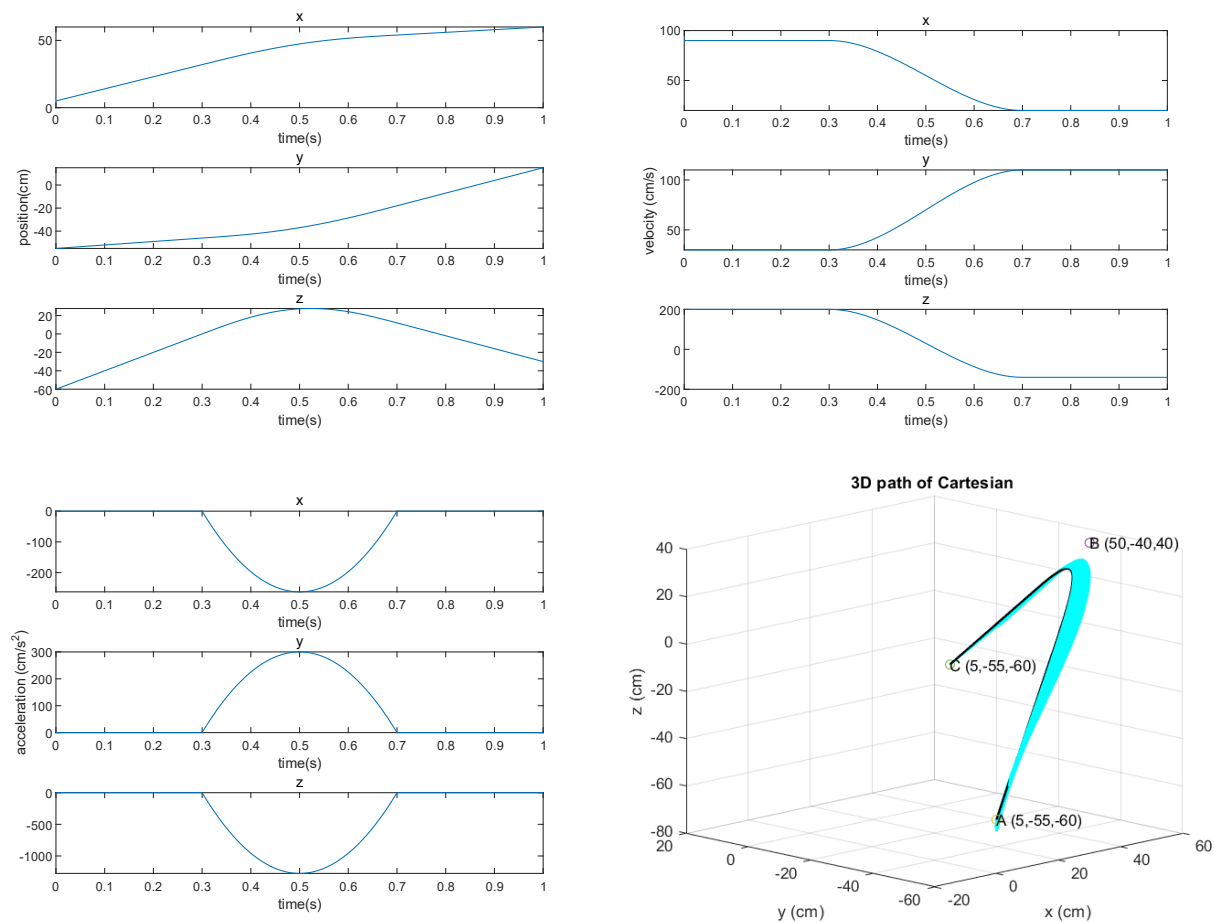
Joint move 執行結果如圖二十二。



圖二十二、各軸角度變化(左上)、速度變化(右上)、加速度變化(左下)、卡式空間中的軌跡與方向(右下)

5.2 Cartesian move

Cartesian move 執行結果如圖二十三。

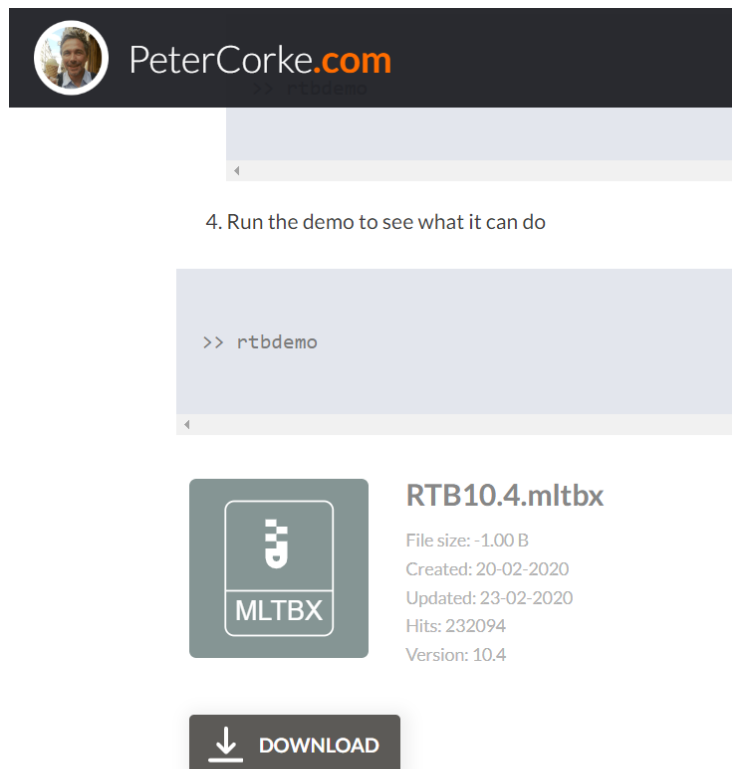


圖二十三、空間中位置變化(左上)、速度變化(右上)、加速度變化(左下)、卡式空間中的軌跡與方向(右下)

6. 加分題:

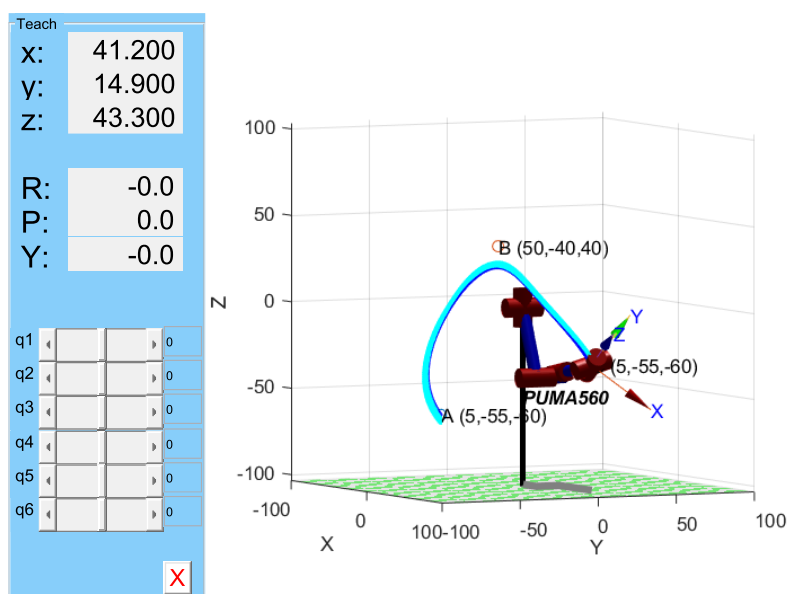
6.1 六軸手臂的動畫 (Joint_robot.m)

首先前往 <https://petercorke.com/toolboxes/robotics-toolbox/> 下載ROBOTICS TOOLBOX(非MATLAB內建的ROBOTIC SYSTEM TOOLBOX)，如圖二十四。



圖二十四 、 下載檔

執行 Joint_robot.m 即可顯示 六軸手臂在 Joint Move 的動畫 (影片連結: <https://www.youtube.com/watch?v=KqzGMOJ37Gk> 。動畫速度稍慢請耐心等待)，如圖二十五。



圖二十五 、 六軸機械手臂動畫

軌跡規劃的部分與程式joint.m一致，因此只講解不同之處(動畫)，首先根據DH-Table 建立手臂模型及關節限制，如圖二十六。

```
figure(5)
% %參數  theta    d      a      alpha  offset : 將某變數 變為可調
L1=Link([ 0      0      0      0      -pi/2      0],'standard');
L2=Link([ 0      0      43.2    0      0      0],'standard');
L3=Link([ 0      14.9     -2      pi/2    0      0],'standard');
L4=Link([ 0      43.3     0      -pi/2    0      0],'standard');
L5=Link([ 0      0      0      pi/2    0      0],'standard');
L6=Link([ 0      0      0      0      0      0],'standard');
robot1=SerialLink([L1 L2 L3 L4 L5 L6 ], 'name', 'PUMA560');
robot1.qlim=[-160*pi/180 160*pi/180; -125*pi/180 125*pi/180; -135*pi 135*pi; -140*pi/180 140*pi/180; -100*pi/180 100*pi/180; -260*pi 260*pi];
robot1.plot( [0 0 0 0 0], 'workspace', [-100 100 -100 100 -50 100]);
robot1.teach
robot1.display();
```

圖二十六 、 建立手臂模型

接著利用內建函數fkine進行順向運動學並畫出軌跡，並輸入每一時間點的關節角度至robot1.plot以完成動畫，如圖二十七。

```
129      atj2=zeros(4,4);
130      for i=1:length(0:0.002:1)
131          atj2=robot1.fkine(theta(:,i)*pi/180); % 順向運動學 取得末端的noap
132          JTA2(i,:)=transl(atj2); % 取得 XYZ
133          jta2=JTA2;
134          plot2(jta2(i,:), 'b.') % 劃出軌跡 ( 藍點 )
135          %robot1.plot(theta(:,i)*pi/180 ) % 軌跡動畫
136          hold on
137      end
138      robot1.plot(theta()*pi/180 ) % 軌跡動畫
139      hold on
```

圖二十七 、 動畫顯示

6.2 討論兩種軌跡規劃的優缺點、遇到奇異點如何處理

軌跡規劃分為Joint空間軌跡規劃及Cartesian空間軌跡規劃，Joint空間軌跡規劃描述各軸關節在末端執行器由起點到終點時的軌跡規劃，而Cartesian空間只考慮末端作用器由起點到終點時的軌跡規劃。因此Cartesian空間軌跡在視覺及直覺上較於簡單，但每個點都須進行逆向運動學，因此計算十分複雜。而Joint空間軌跡較不平滑且有時會有規劃的軌跡有過多轉動的問題，但因為是對各軸關節進行規劃，只會在起點、中繼點、終點使用到反向運動學，因此計算相對簡單。

在奇異點方面Joint空間軌跡規劃除非給定的起點或中繼點或終點為奇異點，否則不會遇到奇異點的問題，Cartesian空間則有可能遇到奇異點的問題，此時可以透過加入其他中繼點去繞開奇異點，來重新規劃路徑。