Robotics Project2

0860077 王國倫 電控所碩一

- ## 介面說明

本次專題使用 MATLAB 做開發，使用前須輸入指定的數字(1~3)，即可開始使用相對應的任務。

```
COMMAND WINDOW
1.Joint moove
2.Cartesian move
3.exit
Please input (1~3):
```

- ## 程式架構說明

清除所有資訊，並建立起點、中繼點和終點的資訊，初始化參數，等待使用者輸入相對應的值。

$$A = \begin{bmatrix} 0 & 1 & 0 & 20 \\ -1 & 0 & 0 & 30 \\ 0 & 0 & 1 & 20 \\ 0 & 0 & 0 & 1 \end{bmatrix}, B = \begin{bmatrix} 0 & 0 & -1 & -10 \\ -1 & 0 & 0 & 15 \\ 0 & 1 & 0 & 30 \\ 0 & 0 & 0 & 1 \end{bmatrix}, C = \begin{bmatrix} 1 & 0 & 0 & -25 \\ 0 & -1 & 0 & 10 \\ 0 & 0 & -1 & -20 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

```
% clear all
clc
clear all
close all
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% defind the orientation and position of A,B,C (cm)
A = [ 0  1  0  20
     -1  0  0  30
      0  0  1  20
      0  0  0  1 ];

B = [ 0  0 -1 -10
     -1  0  0  15
      0  1  0  30
      0  0  0  1 ];

C = [ 1  0  0 -25
      0 -1  0  10
      0  0 -1 -20
      0  0  0  1 ];
```

```
% calculate (n, o, a, p) of points (A, B, C)
nA = A(1:3,1);
oA = A(1:3,2);
aA = A(1:3,3);
pA = A(1:3,4);

nB = B(1:3,1);
oB = B(1:3,2);
aB = B(1:3,3);
pB = B(1:3,4);

nC = C(1:3,1);
oC = C(1:3,2);
aC = C(1:3,3);
pC = C(1:3,4);

number = input('1.Joint moove\n2.Cartesian move\n3.exit\nPlease input (1~3):');
```

1.Joint move

當輸入數字為 1 時，執行 Joint move 的計算，計算 A 到 C 各軸之間的角度、角速度和角加速度的值。

```
s = 1;
sampling_rate = 0.002;
d = thetaB-thetaA;
thetaA2 = thetaA+(thetaB-thetaA)/0.5*(0.5-0.2);
dB = thetaA2-thetaB;
dC = thetaC-thetaB;
for t=-0.5:sampling_rate:0.5
    if t<=-0.2     %linear
        dtheta(:,s) = thetaA+d/0.5*(t+0.5);
        domega(:,s) = d/0.5;
        dalpha(:,s) = [0;0;0;0;0;0];
        s = s+1;
    elseif t>=0.2  %polynomial
        h = t/0.5;
        dtheta(:,s) = dC*h+thetaB;
        domega(:,s) = dC/0.5;
        dalpha(:,s) = [0;0;0;0;0;0];
        s = s+1;
    else           %linear
        h = (t+0.2)/0.4;
        dtheta(:,s) = ((dC*0.2/0.5+dB)*(2-h)*h^2-2*dB)*h+dB+thetaB;
        domega(:,s) = ((dC*0.2/0.5+dB)*(1.5-h)*2*h^2-dB)/0.2;
        dalpha(:,s) = (dC*0.2/0.5+dB)*(1-h)*3*h/0.2^2;
        s = s+1;
    end
end
```

從 A 到 C 的各軸角度變化

```
figure(1)
t=-0.5:sampling_rate:0.5;
for i=1:1:6
    theta = dtheta(i,:);
    subplot(3,2,i);
    plot(t,theta);
    grid
    title(sprintf('joint%i',i));
    if i==3
        ylabel({'Angle(°)';' '}, 'FontSize', 12, 'FontWeight', 'bold');
    end
end
```

從 A 到 C 的各軸角速度變化

```
figure(2)
for i=1:1:6
    omega = domega(i,:);
    subplot(3,2,i);
    plot(t,omega);
    grid
    title(sprintf('joint%i',i));
    if i==3
        ylabel({'Angular Velocity(°/s)';' '}, 'FontSize', 12, 'FontWeight', 'bold');
    end
end
```

從 A 到 C 的各軸角加速度變化

```
figure(3)
for i=1:1:6
    alpha = dalpha(i,:);
    subplot(3,2,i);
    plot(t,alpha);
    grid
    title(sprintf('joint%i',i));
    if i==3
        ylabel({'Angular Acceleration(°/s^2)';' '}, 'FontSize', 12, 'FontWeight', 'bold');
    end
end
```

利用 PUMA 560 的 Forward Kinematics 將 Joint Path 轉成 Cartesian Path

```
s = 1;
for t=-0.5:sampling_rate:0.5
    p = forward_kinematics(dtheta(:,s)');
    x(s) = p(1,1);
    y(s) = p(2,1);
    z(s) = p(3,1);
    s = s+1;
end
```

Forward Kinematics → 使用 project 1 的 FK function

```
% forward kinematics
function [p] = forward_kinematics(joint_varaible);
angle_to_rad = pi/180;  %transfer angle to rad
rad_to_angle = 180/pi;  %transfer rad to angle

%kinematic table parameter
a2 = 43.2;
a3 = -2;
d3 = 14.9;
d4 = 43.3;

%Forward Kinematic
joint = joint_varaible;
theta1=joint(1,1)*angle_to_rad;
theta2=joint(1,2)*angle_to_rad;
theta3=joint(1,3)*angle_to_rad;
theta4=joint(1,4)*angle_to_rad;
theta5=joint(1,5)*angle_to_rad;
theta6=joint(1,6)*angle_to_rad;
```

```matlab
%transfermation matrix A1~A6
A1 = [ cos(theta1)   0    -sin(theta1)    0;
       sin(theta1)   0     cos(theta1)    0;
                 0  -1               0    0;
                 0   0               0    1  ];

A2 = [ cos(theta2)   -sin(theta2)    0    a2*cos(theta2);
       sin(theta2)    cos(theta2)    0    a2*sin(theta2);
                 0              0    1                 0;
                 0              0    0                 1  ];

A3 = [ cos(theta3)   0     sin(theta3)   a3*cos(theta3);
       sin(theta3)   0    -cos(theta3)   a3*sin(theta3);
                 0   1               0               d3;
                 0   0               0                1  ];

A4 = [ cos(theta4)     0    -sin(theta4)     0;
       sin(theta4)     0     cos(theta4)     0;
                 0    -1               0    d4;
                 0     0               0     1  ];

A5 = [ cos(theta5)     0     sin(theta5)      0;
       sin(theta5)     0    -cos(theta5)      0;
                 0     1               0      0;
                 0     0               0      1  ];

A6 = [ cos(theta6)   -sin(theta6)     0     0;
       sin(theta6)    cos(theta6)     0     0;
                 0              0     1     0;
                 0              0     0     1  ];
```

```
    % calculate final transfermation matrix
    T6=A1*A2*A3*A4*A5*A6;

    % get n o a p value
    nx = T6(1,1);
    ny = T6(2,1);
    nz = T6(3,1);

    ox = T6(1,2);
    oy = T6(2,2);
    oz = T6(3,2);

    ax = T6(1,3);
    ay = T6(2,3);
    az = T6(3,3);

    px = T6(1,4);
    py = T6(2,4);
    pz = T6(3,4);

    % transfer n o a to phi theta psi
    phi=atan2(ay,ax)*rad_to_angle;
    theta=atan2(sqrt((ax)^2+(ay)^2),az)*rad_to_angle;
    psi=atan2(oz,-nz)*rad_to_angle;

p = [px, py, pz, phi, theta, psi]';
end
```

畫出由 Joint Move 轉換成 Cartesian 座標的 3D 圖形

```
figure(4)
plot3(x,y,z);

xlabel('X-axis(cm)');
ylabel('Y-axis(cm)');
zlabel('Z-axis(cm)');
text(20,30,20,'A(20,30,20)');
text(-10,15,30,'B(-10,15,30)');
text(-25,10,-20,'C(-25,10,-20)');
grid
title('3D path of Joint Motion')
```

畫出軸座標和虛線

```
%畫出ABC三點座標軸
hold on;
plot3([pA(1), pA(1)+nA(1)*5], [pA(2), pA(2)+nA(2)*5], [pA(3), pA(3)+nA(3)*5], 'r');
hold on;
plot3([pA(1), pA(1)+oA(1)*5], [pA(2), pA(2)+oA(2)*5], [pA(3), pA(3)+oA(3)*5], 'g');
hold on;
plot3([pA(1), pA(1)+aA(1)*5], [pA(2), pA(2)+aA(2)*5], [pA(3), pA(3)+aA(3)*5], 'b');
hold on;
plot3([pB(1), pB(1)+nB(1)*5], [pB(2), pB(2)+nB(2)*5], [pB(3), pB(3)+nB(3)*5], 'r');
hold on;
plot3([pB(1), pB(1)+oB(1)*5], [pB(2), pB(2)+oB(2)*5], [pB(3), pB(3)+oB(3)*5], 'g');
hold on;
plot3([pB(1), pB(1)+aB(1)*5], [pB(2), pB(2)+aB(2)*5], [pB(3), pB(3)+aB(3)*5], 'b');
hold on;
plot3([pC(1), pC(1)+nC(1)*5], [pC(2), pC(2)+nC(2)*5], [pC(3), pC(3)+nC(3)*5], 'r');
hold on;
plot3([pC(1), pC(1)+oC(1)*5], [pC(2), pC(2)+oC(2)*5], [pC(3), pC(3)+oC(3)*5], 'g');
hold on;
plot3([pC(1), pC(1)+aC(1)*5], [pC(2), pC(2)+aC(2)*5], [pC(3), pC(3)+aC(3)*5], 'b');
%畫出虛線
hold on;
plot3([pA(1),pB(1)], [pA(2),pB(2)], [pA(3),pB(3)], ':');
hold on;
plot3([pC(1),pB(1)], [pC(2),pB(2)], [pC(3),pB(3)], ':');
```

2. Cartesian move

當輸入數字為 2 時，執行 Cartesian move 的計算，先計算初始的參數。

```
sampling_rate = 0.002;
r = 1;
x = dot(nA, (pB - pA));
y = dot(oA, (pB - pA));
z = dot(aA, (pB - pA));
psi = atan2(dot(oA,aB), dot(nA, aB));
temp = sqrt(dot(nA, aB)^2 + dot(oA, aB)^2);
theta = atan2(temp, dot(aA, aB));
V_r_theta = 1-cos(r*theta);
sin_phi = -sin(psi)*cos(psi)*V_r_theta*dot(nA, nB) + (cos(psi)^2*V_r_theta+cos(theta))*dot(oA, nB) - sin(psi)*sin(theta)*dot(aA, nB);
cos_phi = -sin(psi)*cos(psi)*V_r_theta*dot(nA, oB) + (cos(psi)^2*V_r_theta+cos(theta))*dot(oA, oB) - sin(psi)*sin(theta)*dot(aA, oB);
phi = atan2(sin_phi, cos_phi);
```

直線部分的路徑規劃 (-0.5s~-0.2s)

```
dataA = 1;                    % the index of the data of the matrix
for t=-0.5:sampling_rate:-0.2
    h=(t+0.5)/0.5;
    dx=x*h;
    dy=y*h;
    dz=z*h;
    dsi=psi;
    dtheta=theta*h;
    dphi=phi*h;

    S_psi=sin(psi);
    C_psi=cos(psi);
    S_theta=sin(dtheta);
    C_theta=cos(dtheta);
    V_theta=1-C_theta;
    S_phi=sin(dphi);
    C_phi=cos(dphi);

    % find Dr with Dr=Tr*Rar*Ror
    Tr = [1  0  0  dx;
          0  1  0  dy;
          0  0  1  dz;
          0  0  0  1];
    Rar = [S_psi^2*V_theta+C_phi, -S_psi*C_psi*V_theta , C_psi*S_phi, 0 ;
           -S_psi*C_psi*V_theta , C_psi^2*V_theta+C_phi, S_psi*S_phi, 0 ;
           -C_psi*S_phi         , -S_psi*S_phi         , C_phi      , 0 ;
           0                    , 0                    , 0          , 1];
    Ror = [C_theta, -S_theta, 0, 0 ;
           S_theta, C_theta , 0, 0 ;
           0      , 0       , 1, 0 ;
           0      , 0       , 0, 1];
    Dr = Tr*Rar*Ror;

    pA_B(:,:,dataA)=A*Dr;
    xA_B(:,dataA)=pA_B(1,4,dataA);
    yA_B(:,dataA)=pA_B(2,4,dataA);
    zA_B(:,dataA)=pA_B(3,4,dataA);
    dataA=dataA+1;
end
```

曲線部分的路徑規劃 (-0.2s~0.2s)

```
A2=pA_B(:,:,dataA-1);                          % A'的位置
nA2=[A2(1,1);A2(2,1);A2(3,1)];
oA2=[A2(1,2);A2(2,2);A2(3,2)];
aA2=[A2(1,3);A2(2,3);A2(3,3)];
pA2=[A2(1,4);A2(2,4);A2(3,4)];

xA=nB'*(pA2-pB);
yA=oB'*(pA2-pB);
zA=aB'*(pA2-pB);
psiA=atan2(oB'*aA2,nB'*aA2);
thetaA=atan2(sqrt((nB'*aA2)^2+(oB'*aA2)^2),aB'*aA2);
SphiA=-sin(psiA)*cos(psiA)*(1-cos(thetaA))*(nB'*nA2)+((cos(psiA))^2*(1-cos(thetaA))+cos(thetaA))*(oB'*nA2)-sin(psiA)*sin(thetaA)*(aB'*nA2);
CphiA=-sin(psiA)*cos(psiA)*(1-cos(thetaA))*(nB'*oA2)+((cos(psiA))^2*(1-cos(thetaA))+cos(thetaA))*(oB'*oA2)-sin(psiA)*sin(thetaA)*(aB'*oA2);
phiA=atan2(SphiA,CphiA);

xC=nB'*(pC-pB);
yC=oB'*(pC-pB);
zC=aB'*(pC-pB);
psiC=atan2(oB'*aC,nB'*aC);
thetaC=atan2(sqrt((nB'*aC)^2+(oB'*aC)^2),aB'*aC);
SphiC=-sin(psiC)*cos(psiC)*(1-cos(thetaC))*(nB'*nC)+((cos(psiC))^2*(1-cos(thetaC))+cos(thetaC))*(oB'*nC)-sin(psiC)*sin(thetaC)*(aB'*nC);
CphiC=-sin(psiC)*cos(psiC)*(1-cos(thetaC))*(nB'*oC)+((cos(psiC))^2*(1-cos(thetaC))+cos(thetaC))*(oB'*oC)-sin(psiC)*sin(thetaC)*(aB'*oC);
phiC=atan2(SphiC,CphiC);

if abs(psiC-psiA)>pi/2
    psiA=psiA+pi;
    thetaA=-thetaA;
end
```

```
dataB = 1;                                      % path planing
for t=(-0.2+sampling_rate):sampling_rate:(0.2-sampling_rate)
    h=(t+0.2)/(0.2+0.2);
    dx_B=((xC*0.2/0.5+xA)*(2-h)*h^2-2*xA)*h+xA;
    dy_B=((yC*0.2/0.5+yA)*(2-h)*h^2-2*yA)*h+yA;
    dz_B=((zC*0.2/0.5+zA)*(2-h)*h^2-2*zA)*h+zA;
    dpsi_B=(psiC-psiA)*h+psiA;
    dtheta_B=((thetaC*0.2/0.5+thetaA)*(2-h)*h^2-2*thetaA)*h+thetaA;
    dphi_B=((phiC*0.2/0.5+phiA)*(2-h)*h^2-2*phiA)*h+phiA;

    S_psi=sin(dpsi_B);
    C_psi=cos(dpsi_B);
    S_theta=sin(dtheta_B);
    C_theta=cos(dtheta_B);
    V_theta=1-C_theta;
    S_phi=sin(dphi_B);
    C_phi=cos(dphi_B);

    Tr = [1 0 0 dx_B ;
          0 1 0 dy_B ;
          0 0 1 dz_B ;
          0 0 0 1    ];
    Rar = [S_psi^2*V_theta+C_phi, -S_psi*C_psi*V_theta , C_psi*S_phi, 0 ;
           -S_psi*C_psi*V_theta , C_psi^2*V_theta+C_phi, S_psi*S_phi, 0 ;
           -C_psi*S_phi         , -S_psi*S_phi         , C_phi      , 0 ;
           0                    , 0                    , 0          , 1];
    Ror = [C_theta, -S_theta, 0, 0 ;
           S_theta, C_theta , 0, 0 ;
           0      , 0       , 1, 0 ;
           0      , 0       , 0, 1];
    Dr_B = Tr*Rar*Ror;

    p_B(:,:,dataB)=B*Dr_B;
    x_B(:,dataB)=p_B(1,4,dataB);
    y_B(:,dataB)=p_B(2,4,dataB);
    z_B(:,dataB)=p_B(3,4,dataB);
    dataB=dataB+1;
end
```

直線部分的路徑規劃 (0.2s~0.5s)

```
dataC = 1;                                          % path planing
for t=0.2:sampling_rate:0.5
    h=t/0.5;
    dx_C=xC*h;
    dy_C=yC*h;
    dz_C=zC*h;
    dpsi_C=psiC;
    dtheta_C=thetaC*h;
    dphi_C=phiC*h;

    S_psi=sin(dpsi_C);
    C_psi=cos(dpsi_C);
    S_theta=sin(dtheta_C);
    C_theta=cos(dtheta_C);
    V_theta=1-C_theta;
    S_phi=sin(dphi_C);
    C_phi=cos(dphi_C);

    % find Dr with Dr=Tr*Rar*Ror
    Tr = [1 0 0 dx_C ;
          0 1 0 dy_C ;
          0 0 1 dz_C ;
          0 0 0 1    ];
    Rar = [S_psi^2*V_theta+C_phi, -S_psi*C_psi*V_theta , C_psi*S_phi, 0 ;
           -S_psi*C_psi*V_theta , C_psi^2*V_theta+C_phi, S_psi*S_phi, 0 ;
           -C_psi*S_phi         , -S_psi*S_phi         , C_phi      , 0 ;
           0                    , 0                    , 0          , 1];
    Ror = [C_theta, -S_theta, 0, 0 ;
           S_theta, C_theta , 0, 0 ;
           0      , 0       , 1, 0 ;
           0      , 0       , 0, 1];
    Dr_C = Tr*Rar*Ror;

    p_C(:,:,dataC)=B*Dr_C;
    x_C(:,dataC)=p_C(1,4,dataC);
    y_C(:,dataC)=p_C(2,4,dataC);
    z_C(:,dataC)=p_C(3,4,dataC);
    dataC=dataC+1;
end
```

從 A 到 C 之 x, y, z 各軸的變化情形

```
X=[xA_B x_B x_C];
Y=[yA_B y_B y_C];
Z=[zA_B z_B z_C];
t=-0.5:sampling_rate:0.5;

figure(2)
subplot(3,1,1);
plot(t,X);
title('position of x');
grid

subplot(3,1,2);
plot(t,Y);
ylabel({'Position(cm)';' '}, 'FontSize', 12, 'FontWeight', 'bold');
title('position of y');
grid

subplot(3,1,3);
plot(t,Z);
title('position of z');
xlabel('Time(s)')
grid
```

從 A 到 C 之 x, y, z 各速度的變化情形

```
dt=t(2:501);
dX=diff(X)/sampling_rate;
dY=diff(Y)/sampling_rate;
dZ=diff(Z)/sampling_rate;

figure(3)
subplot(3,1,1);
plot(dt,dX);
title('velocity of x');
grid

subplot(3,1,2);
plot(dt,dY);
title('velocity of y');
ylabel({'Velocity(cm/s)';' '}, 'FontSize', 12, 'FontWeight', 'bold');
grid

subplot(3,1,3);
plot(dt,dZ);
title('velocity of z');
xlabel('Time(s)')
grid
```

從 A 到 C 之 x, y, z 各加速度的變化情形

```
dt2=t(3:501);
dX2=diff(dX)/sampling_rate;
dY2=diff(dY)/sampling_rate;
dZ2=diff(dZ)/sampling_rate;

figure(4)
subplot(3,1,1);
plot(dt2,dX2);
title('acceleration of x');
grid

subplot(3,1,2);
plot(dt2,dY2);
title('acceleration of y');
ylabel({'Acceleration(cm/s^2)';' '}, 'FontSize', 12, 'FontWeight', 'bold');
grid

subplot(3,1,3);
plot(dt2,dZ2);
title('acceleration of z');
xlabel('Time(s)')
grid
```

從 A 至 C 的 3D 路徑圖

```
figure(1)
plot3(xA_B,yA_B,zA_B,x_B,y_B,z_B,x_C,y_C,z_C);

%畫出ABC三點座標軸
hold on;
plot3([pA(1), pA(1)+nA(1)*5], [pA(2), pA(2)+nA(2)*5], [pA(3), pA(3)+nA(3)*5], 'r');
hold on;
plot3([pA(1), pA(1)+oA(1)*5], [pA(2), pA(2)+oA(2)*5], [pA(3), pA(3)+oA(3)*5], 'g');
hold on;
plot3([pA(1), pA(1)+aA(1)*5], [pA(2), pA(2)+aA(2)*5], [pA(3), pA(3)+aA(3)*5], 'b');
hold on;
plot3([pB(1), pB(1)+nB(1)*5], [pB(2), pB(2)+nB(2)*5], [pB(3), pB(3)+nB(3)*5], 'r');
hold on;
plot3([pB(1), pB(1)+oB(1)*5], [pB(2), pB(2)+oB(2)*5], [pB(3), pB(3)+oB(3)*5], 'g');
hold on;
plot3([pB(1), pB(1)+aB(1)*5], [pB(2), pB(2)+aB(2)*5], [pB(3), pB(3)+aB(3)*5], 'b');
hold on;
plot3([pC(1), pC(1)+nC(1)*5], [pC(2), pC(2)+nC(2)*5], [pC(3), pC(3)+nC(3)*5], 'r');
hold on;
plot3([pC(1), pC(1)+oC(1)*5], [pC(2), pC(2)+oC(2)*5], [pC(3), pC(3)+oC(3)*5], 'g');
hold on;
plot3([pC(1), pC(1)+aC(1)*5], [pC(2), pC(2)+aC(2)*5], [pC(3), pC(3)+aC(3)*5], 'b');
%畫出虛線
hold on;
plot3([xA_B(151),pB(1)], [yA_B(151),pB(2)], [zA_B(151),pB(3)], ':');
hold on;
plot3([x_C(1),pB(1)], [y_C(1),pB(2)], [z_C(1),pB(3)], ':');

xlabel('X-axis(cm)');
ylabel('Y-axis(cm)');
zlabel('Z-axis(cm)');
text(20,10,-10,'A(20, 10, -10)');
text(20,-5,10,'B(20, -5, 10)');
text(-10,15,25,'C(-10, 15, 25)');
grid
title('3D path of Cartesian Motion')
```

- 數學運算說明

1.Joint move

Joint move 於 transition 過程之運動的路徑θ(t),速度ω(t),加速度α(t)分別表示為:

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

$$\dot{\theta}(t)=\omega(t)=\frac{1}{T}\left[(\Delta B+\Delta C)(3\text{-}2h(t))h^2(t)-\Delta B\right]$$

$$\ddot{\theta}(t)=\alpha(t)=\frac{3}{t_{acc}T}\left[(\Delta B+\Delta C)(1\text{-}h(t))h(t)\right]$$

其中 $\Delta B=\theta(t_A)-\theta(t_B)$ ，$\Delta C=\theta(t_C)-\theta(t_B)$

而且 $h(t)=\frac{t+t_{acc}}{2t_{acc}}$ ，$-T\le t\le +T$ ，$t_{acc}=0.2$ ，$T=0.5$

Joint move 於 直線路段

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

2. Cartesian move

Cartesian move 中,將各 drive 變數於 transition 過程之運動路徑θ(t),速度ω(t),加速度α(t)分別表示為:

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

$$\dot{q}(t)=\frac{1}{T}\left[(\Delta B+\Delta C)(3\text{-}2h(t))h^2(t)-\Delta B\right]$$

$$\ddot{q}(t)=\frac{3}{t_{acc}T}\left[(\Delta B+\Delta C)(1\text{-}h(t))h(t)\right]$$

其中 $\Delta B=q(t_A)-q(t_B)$ ，$\Delta C=q(t_C)-q(t_B)$

而且 $q(t_A)=x_A,y_A,z_A,\theta_A,\phi_A$ ，$q(t_B)=B=0$ ，$q(t_C)=x_C,y_C,z_C,\theta_C,\phi_C$

另外 $h(t)=\frac{t+t_{acc}}{2t_{acc}}$ ，$-T\le t\le +T$ ，$t_{acc}=0.2$ ，$T=0.5$

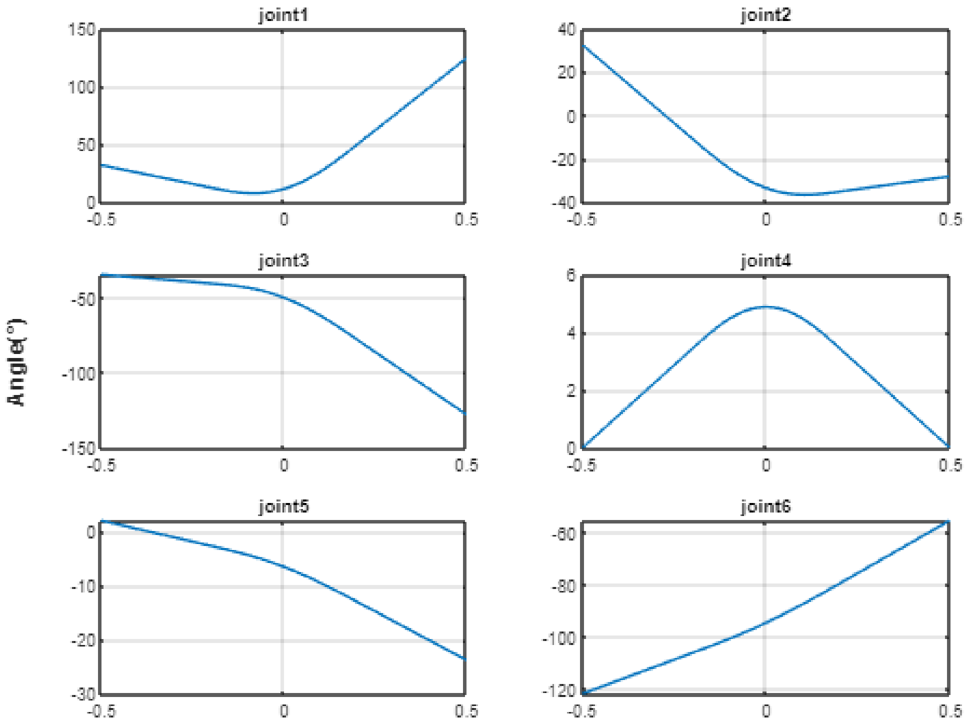
Cartesian move 於 直線路段

$$\begin{cases} q=\Delta C\cdot\dfrac{t}{T} \\ \dot{q}=\dfrac{\Delta C}{T} \\ \ddot{q}=0 \end{cases}$$

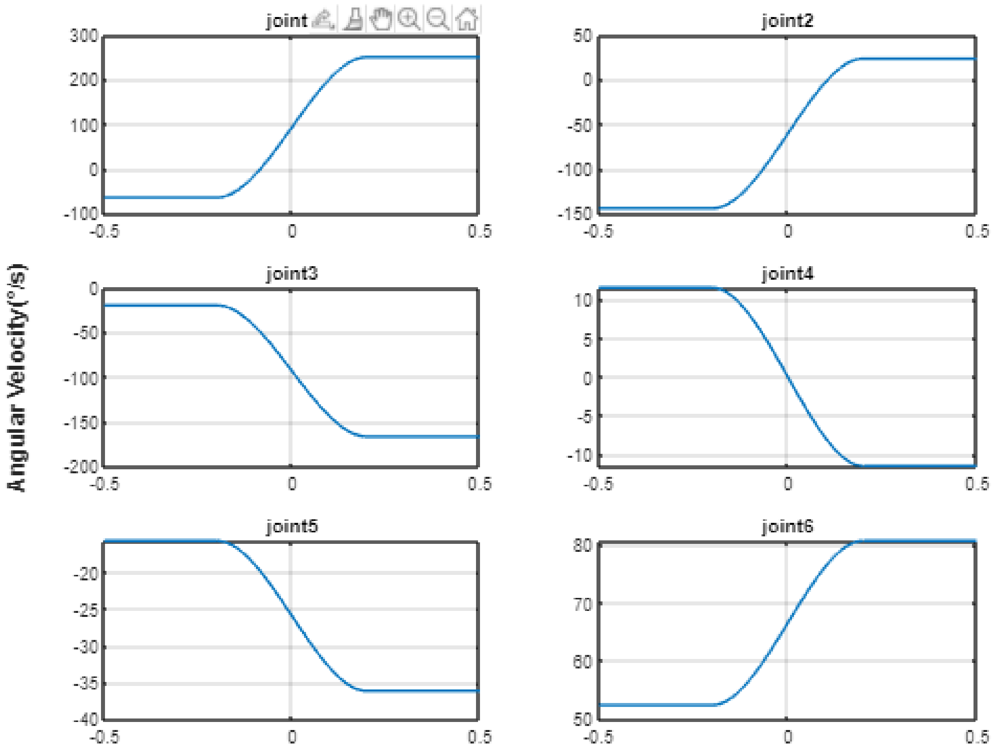
- 軌跡規劃曲線圖結果

1.Joint move

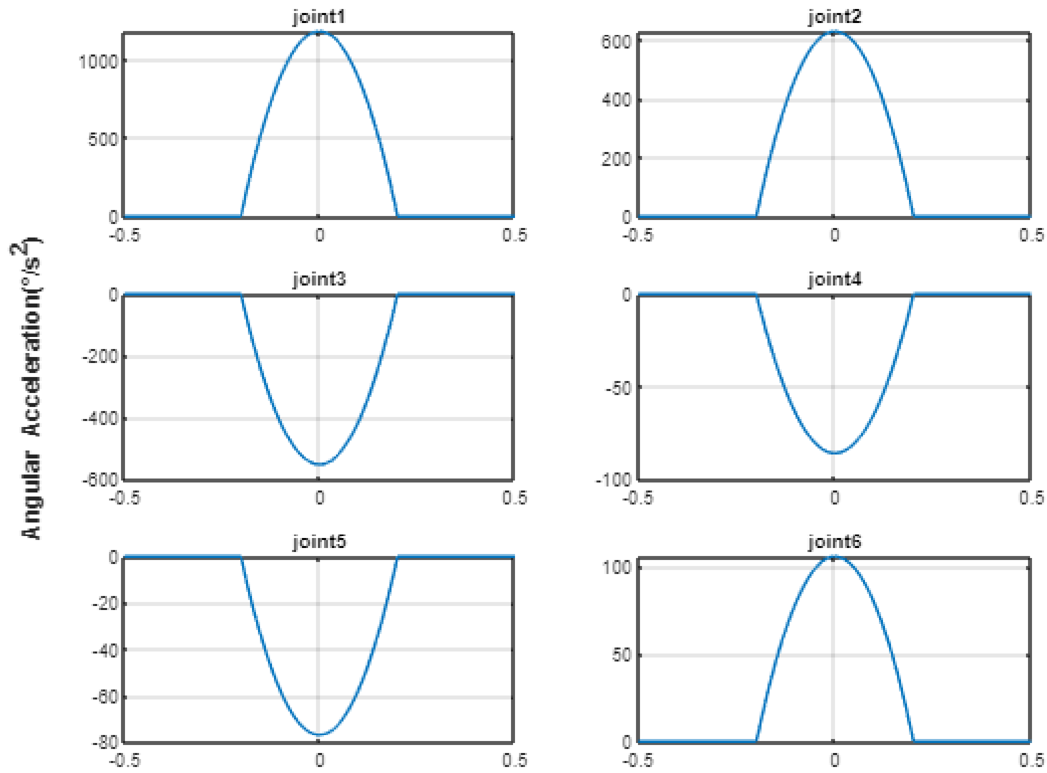
Joint 1~6 的角度變化圖



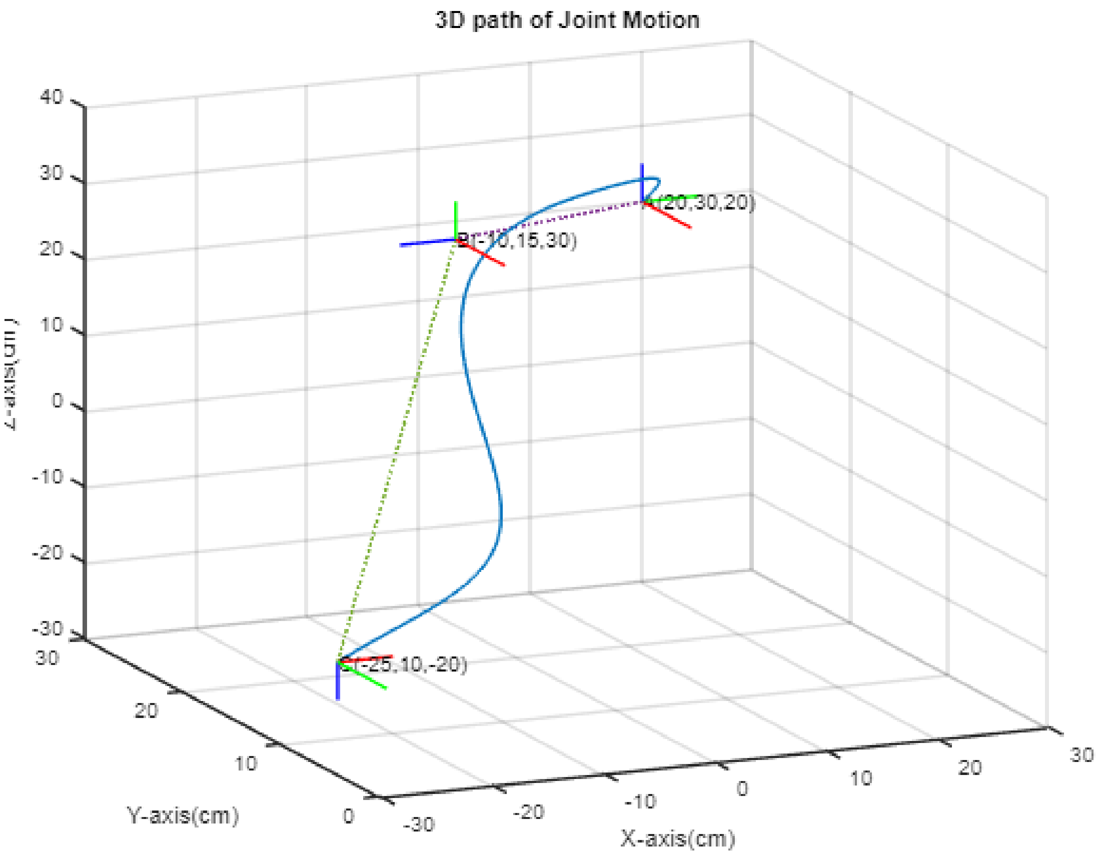
Joint 1~6 的角速度變化圖



Joint 1~6 的角加速度變化圖

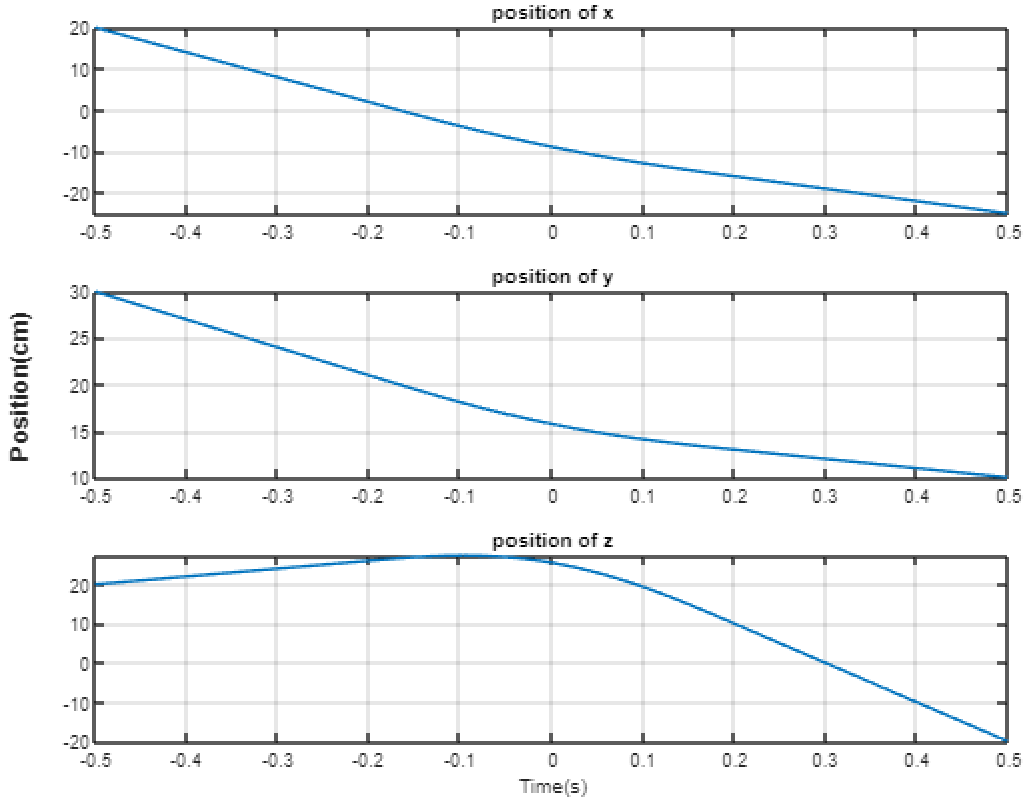


由 Joint Motion 轉換成 Cartesian 座標的 3D 圖形

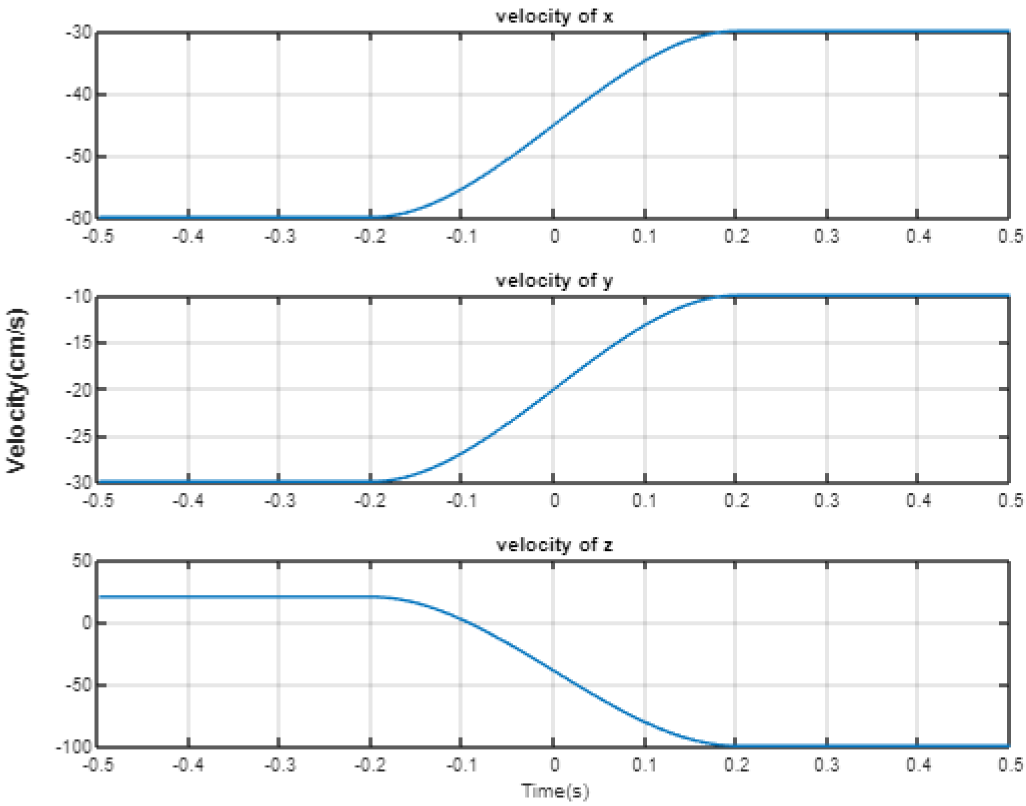


2. Cartesian move

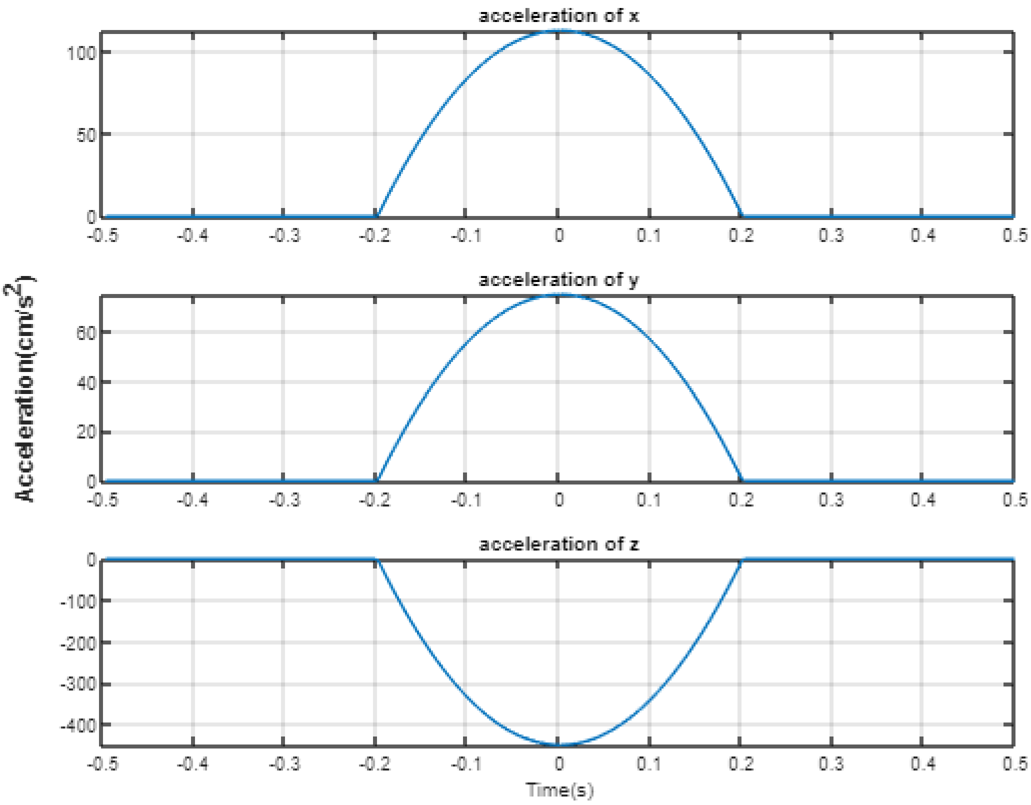
從 A 到 C 之 x, y, z 位置的變化情形



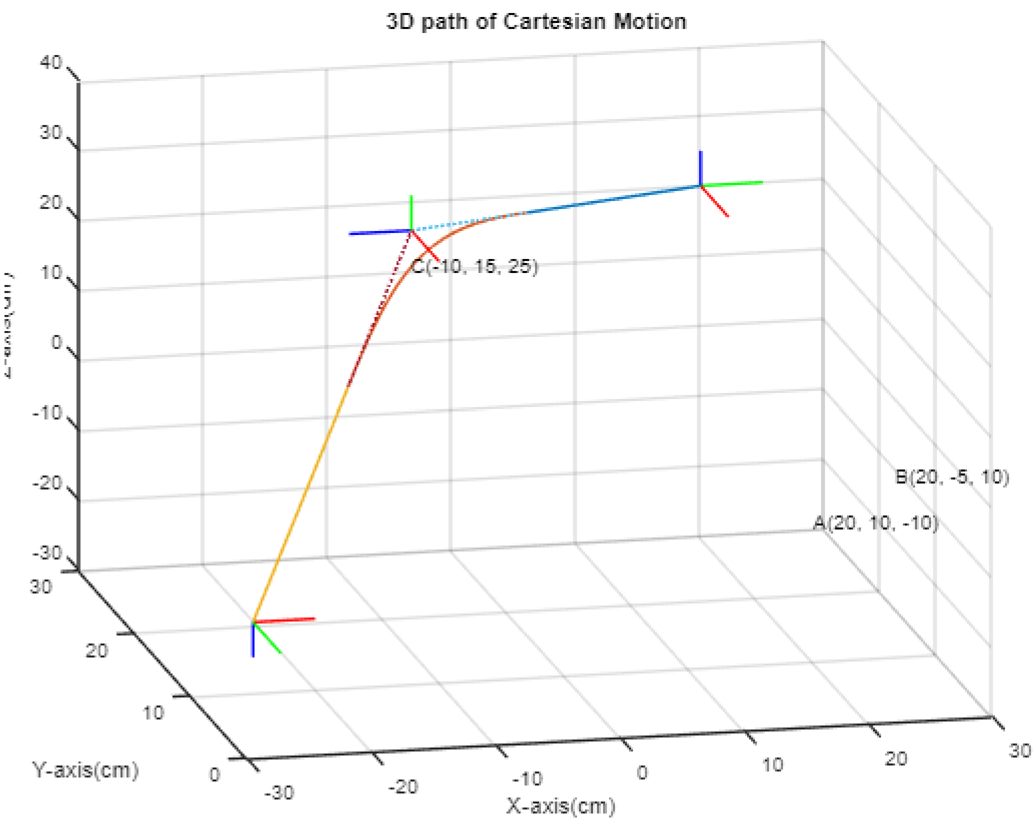
從 A 到 C 之 x, y, z 速度的變化情形



從 A 到 C 之 x, y, z 加速度的變化情形



從 A 到 C 的 3D 路徑圖



- 加分題：討論兩種軌跡規劃的優缺點

1. Joint move
   - Advantage: efficient in computation, no singularity problem, no configuration problem, minimum time planning.
   - Disadvantage: the corresponding Cartesian locations may be complicated.
2. Cartesian move
   - Advantage: motion between path segments and points is well defined. Different constraints, such as smoothness and shortest path, etc., can be imposed upon.
   - Disadvantage:
     - Computational load is high.
     - The motion breaks down when singularity occurs.