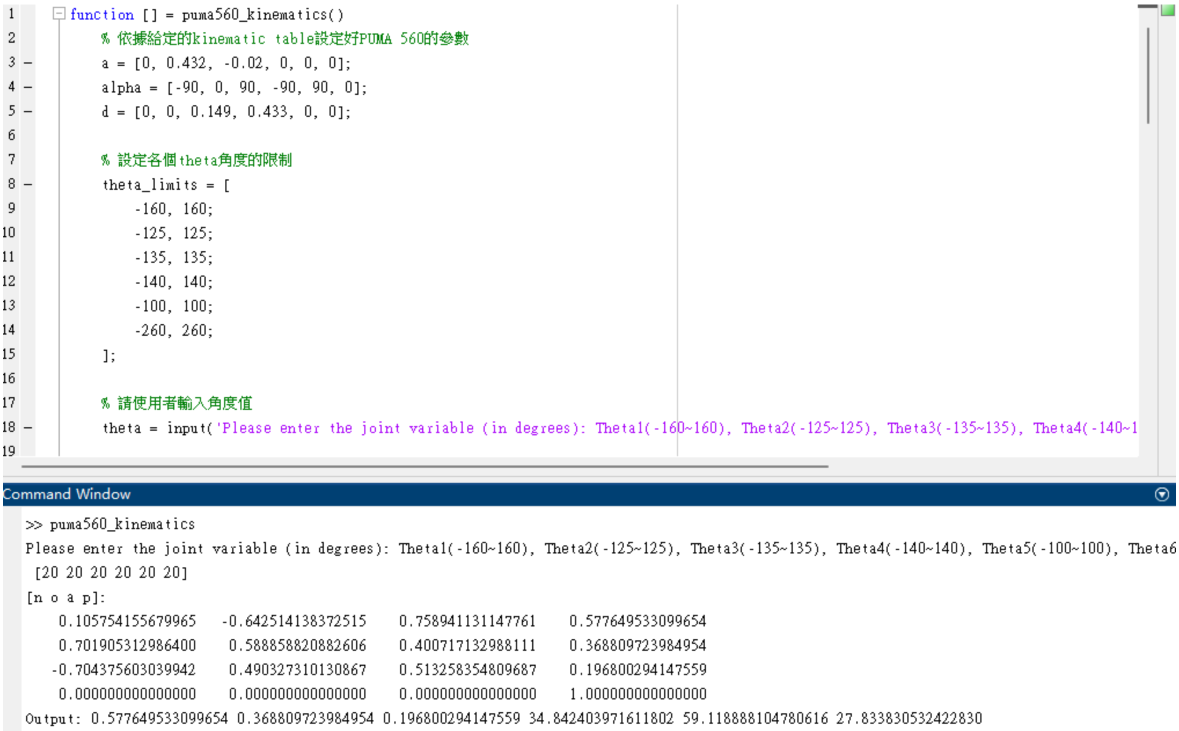
312605015 詹恆瑜 機器人學 Project1

(一)介面說明

我使用 matlab 來做這次的 project，我將 question1(正運動學)、2(逆運動學)分別寫在 puma560_kinematics.m 和 puma560_inverse_kinematics.m 中。

以 puma560_kinematics.m 為例:

點選介面上面的 run 鍵來做執行(如左圖所示)



```
function [] = puma560_kinematics()
    % 依據給定的kinematic table設定好PUMA 560的參數
    a = [0, 0.432, -0.02, 0, 0, 0];
    alpha = [-90, 0, 90, -90, 90, 0];
    d = [0, 0, 0.149, 0.433, 0, 0];

    % 設定各個theta角度的限制
    theta_limits = [
        -160, 160;
        -125, 125;
        -135, 135;
        -140, 140;
        -100, 100;
        -260, 260;
    ];

    % 請使用者輸入角度值
    theta = input('Please enter the joint variable (in degrees): Theta1(-160~160), Theta2(-125~125), Theta3(-135~135), Theta4(-140~1
```

```
Command Window
>> puma560_kinematics
Please enter the joint variable (in degrees): Theta1(-160~160), Theta2(-125~125), Theta3(-135~135), Theta4(-140~140), Theta5(-100~100), Theta6
 [20 20 20 20 20 20]
[n o a p]:
    0.105754155679965   -0.642514138372515    0.758941131147761    0.577649533099654
    0.701905312986400    0.588858820882606    0.400717132988111    0.368809723984954
   -0.704375603039942    0.490327310130867    0.513258354809687    0.196800294147559
    0.000000000000000    0.000000000000000    0.000000000000000    1.000000000000000
Output: 0.577649533099654 0.368809723984954 0.196800294147559 34.842403971611802 59.118888104780616 27.833830532422830
```

之後程式的 command window 會有請我們輸入 the joint variable，如我上圖所示，我將六組數值輸入到[]中括號中進行計算，他就會呈現出[n o a p]的值，以及所要 output 結果。

以 puma560_inverse_kinematics.m 為例:

同樣點選介面上面的 run 鍵來做執行(如左圖所示)

```
Please enter Cartesian point：
[    0.105754155679965    -0.642514138372515    0.758941131147761    0.577649533099654
     0.701905312986400     0.588858820882606    0.400717132988111    0.368809723984954
    -0.704375603039942     0.490327310130867    0.513258354809687    0.196800294147559
     0.000000000000000     0.000000000000000    0.000000000000000    1.000000000000000]
Corresponding variable (theta1, theta2, theta3, theta4, theta5, theta6)

20.0000 20.0000 20.0000 20.0000 20.0000 20.0000

Corresponding variable (theta1, theta2, theta3, theta4, theta5, theta6)
theta2 is out of range!
-134.8863 -127.2131 20.0000 19.1824 50.8830 -166.6086

Corresponding variable (theta1, theta2, theta3, theta4, theta5, theta6)
theta3 is out of range!
theta4 is out of range!
20.0000 -52.7869 165.2892 171.6767 53.9098 -136.1928

Corresponding variable (theta1, theta2, theta3, theta4, theta5, theta6)
theta2 is out of range!
theta3 is out of range!
theta4 is out of range!
-134.8863 -200.0000 165.2892 146.2403 27.3062 56.4786

Corresponding variable (theta1, theta2, theta3, theta4, theta5, theta6)
theta4 is out of range!
20.0000 20.0000 20.0000 -160.0000 -20.0000 -160.0000

Corresponding variable (theta1, theta2, theta3, theta4, theta5, theta6)
theta2 is out of range!
theta4 is out of range!
-134.8863 -127.2131 20.0000 -160.8176 -50.8830 13.3914

Corresponding variable (theta1, theta2, theta3, theta4, theta5, theta6)
theta3 is out of range!
20.0000 -52.7869 165.2892 -8.3233 -53.9098 43.8072

Corresponding variable (theta1, theta2, theta3, theta4, theta5, theta6)
theta2 is out of range!
theta3 is out of range!
-134.8863 -200.0000 165.2892 -33.7597 -27.3062 -123.5214
```

他就會請使用者輸入一個 4*4 的[n o a p]矩陣，且要記得加入中括號，之後就會和上圖一樣將八種解給呈現出來，並把超出範圍不合理的角度呈現出來。

(二)程式架構說明

以 puma560_kinematics.m 為例:

```
function [] = puma560_kinematics()
    % 依據給定的kinematic table設定好PUMA 560的參數
    a = [0, 0.432, -0.02, 0, 0, 0];
    alpha = [-90, 0, 90, -90, 90, 0];
    d = [0, 0, 0.149, 0.433, 0, 0];

    % 設定各個theta角度的限制
    theta_limits = [
        -160, 160;
        -125, 125;
        -135, 135;
        -140, 140;
        -100, 100;
        -260, 260;
    ];

    % 請使用者輸入角度值
    theta = input('Please enter the joint variable (in degrees): Theta1(-160~160), 
Theta2(-125~125), Theta3(-135~135), Theta4(-140~140), Theta5(-100~100), Theta6(-
260~260):\n ');

    % 確認每個角度都在範圍內，如果沒有會顯示出來
    for i = 1:6
        if theta(i) < theta_limits(i, 1) || theta(i) > theta_limits(i, 2)
            fprintf('Theta%d is out of range!\n', i);
        end
    end

    % 計算轉換矩陣
    T = eye(4);
    for i = 1:6
        T = T * dh_transform(d(i), theta(i), a(i), alpha(i));
    end

    % 印出[n o a p]的結果
    fprintf('[n o a p]:\n');
```

```
    for i = 1:size(T,1)
        for j = 1:size(T,2)
            fprintf(' %20.15f', T(i,j));
        end
        fprintf('\n');
    end

    % 從旋轉矩陣中找出 euler_angle
    [phi, theta, psi] = extract_euler_angles(T(1:3, 1:3));

    % 印出位置和方向
    fprintf('Output: %.15f %.15f %.15f %.15f %.15f %.15f\n', T(1, 4), T(2, 4), T(3,
4), phi, theta, psi);
end

function T = dh_transform(d, theta, a, alpha)
    theta = deg2rad(theta); % 轉換為弧度
    alpha = deg2rad(alpha); % 轉換為弧度
    T = [cos(theta), -sin(theta)*cos(alpha),  sin(theta)*sin(alpha), a*cos(theta);
         sin(theta),  cos(theta)*cos(alpha), -cos(theta)*sin(alpha), a*sin(theta);
         0,           sin(alpha),            cos(alpha),             d;
         0,           0,                     0,                      1];
end

function [phi, theta, psi] = extract_euler_angles(R)
    % 利用 ZYZ Euler angles 來做旋轉矩陣 R

    if R(3,3) == 1 || R(3,3) == -1
        % 如果為奇異點，將把其值設為 0
        theta = 0;
        psi = 0; % 可為任意值
        phi = atan2(R(1,2), R(1,1));
    else
        theta = acos(R(3,3));        % 先利用 Z-axis 旋轉
        psi = atan2(R(2,3), R(1,3)); % 再利用 Y-axis 旋轉
        phi = atan2(R(3,2), -R(3,1));% 最後再利用 Z-axis 旋轉
    end
```

```
    % 將弧度再改為角度
    phi = rad2deg(phi);
    theta = rad2deg(theta);
    psi = rad2deg(psi);
end
```

把dh轉換矩陣函式和轉換ZYZ歐拉角的函示給寫好，放在程式碼的最下方方便上面的程式碼可以直接引用，主程式function [] = puma560_kinematics()之後會讀過我們先設好的參數再要求使用者輸入角度值，如果有超過限制角度會呈現出來，之後就是將矩陣做轉換，打印出最後的[n o a p]結果，之後再從其中提取出歐拉角並將計算結果呈現在output結果。

以 puma560_inverse_kinematics.m 為例:

```
% 設定 DH 參數，且設置顯示到小數點後 15 位
format long;
d = [0, 0, 0.149, 0.433, 0, 0];
a = [0, 0.432, -0.02, 0, 0, 0];
alpha = [-90, 0, 90, -90, 90, 0];

% 讀取輸入的矩陣
userInput = input('Please enter Cartesian point：\n');

% 轉為 4*4 矩陣
m = reshape(userInput, 4, 4);


% theta1 兩種可能
theta1_1 = atan2(m(14), m(13)) - atan2(0.149, sqrt(m(13)^2 + m(14)^2
- 0.149^2));
theta1_1 = rad2deg(theta1_1) ;
theta1_2 = atan2(m(14), m(13)) - atan2(0.149, -(sqrt(m(13)^2 +
m(14)^2 - 0.149^2)));
theta1_2 = rad2deg(theta1_2) ;


% theta3 兩種可能
mix = (m(13)^2 + m(14)^2 + m(15)^2 - 0.432^2 - (-0.02)^2 - 0.149^2 -
```

```
0.433^2) / (2*0.432);
theta3_1 = atan2(mix, sqrt(0.02^2 + 0.433^2 - mix^2)) - atan2((-
0.02) , 0.433);
theta3_1 = vpa(theta3_1 * 180/pi);
theta3_2 = atan2(mix, -(sqrt(0.02^2 + 0.433^2 - mix^2))) - atan2((-
0.02) , 0.433);
theta3_2 = vpa(theta3_2 * 180/pi);


%theta2 的四種可能
syms arc_theta2_1 arc_theta2_2 arc_theta2_3 arc_theta2_4
arc_theta1_1 = theta1_1*pi/180;
arc_theta3_1 = theta3_1*pi/180;
arc_theta1_2 = theta1_2*pi/180;
arc_theta3_2 = theta3_2*pi/180;
q = cos(arc_theta1_1)*cos(arc_theta2_1+arc_theta3_1)*m(13) +
sin(arc_theta1_1)*cos(arc_theta2_1+arc_theta3_1)*m(14) -
sin(arc_theta2_1+arc_theta3_1)*m(15) + 0.02 -0.432*cos(arc_theta3_1);
k = cos(arc_theta1_2)*cos(arc_theta2_2+arc_theta3_1)*m(13) +
sin(arc_theta1_2)*cos(arc_theta2_2+arc_theta3_1)*m(14) -
sin(arc_theta2_2+arc_theta3_1)*m(15) + 0.02 -0.432*cos(arc_theta3_1);
i = cos(arc_theta1_1)*cos(arc_theta2_3+arc_theta3_2)*m(13) +
sin(arc_theta1_1)*cos(arc_theta2_3+arc_theta3_2)*m(14) -
sin(arc_theta2_3+arc_theta3_2)*m(15) + 0.02 -0.432*cos(arc_theta3_2);
j = cos(arc_theta1_2)*cos(arc_theta2_4+arc_theta3_2)*m(13) +
sin(arc_theta1_2)*cos(arc_theta2_4+arc_theta3_2)*m(14) -
sin(arc_theta2_4+arc_theta3_2)*m(15) + 0.02 -0.432*cos(arc_theta3_2);
q = matlabFunction(q);
k = matlabFunction(k);
i = matlabFunction(i);
j = matlabFunction(j);
[x1,fval1]=fzero(q,1);
[x2,fval2]=fzero(k,-3);
[x3,fval3]=fzero(i,-1);
[x4,fval4]=fzero(j,-3);
theta2_1 = x1*180/pi ;
theta2_2 = x2*180/pi ;
theta2_3 = x3*180/pi ;
```

```
theta2_4 = x4*180/pi ;
formatSpec = '%.4f\n';
theta2_1 = sprintf(formatSpec, theta2_1);
theta2_2 = sprintf(formatSpec, theta2_2);
theta2_3 = sprintf(formatSpec, theta2_3);
theta2_4 = sprintf(formatSpec, theta2_4);

%theta4 的八種解
T6_3_9 = cosd(theta1_1)*cosd(theta2_1 + theta3_1)*m(9) +
sind(theta1_1)*cosd(theta2_1 + theta3_1)*m(10) -sind(theta2_1 +
theta3_1)*m(11);
T6_3_10 = -sind(theta1_1)*m(9) + cosd(theta1_1)*m(10);
theta4_1 = atan2(T6_3_10, T6_3_9);
theta4_1 = vpa(theta4_1 * 180/pi, 15);

T6_3_9 = cosd(theta1_2)*cosd(theta2_2 + theta3_1)*m(9) +
sind(theta1_2)*cosd(theta2_2 + theta3_1)*m(10) -sind(theta2_2 +
theta3_1)*m(11);
T6_3_10 = -sind(theta1_2)*m(9) + cosd(theta1_2)*m(10);
theta4_2 = atan2(T6_3_10, T6_3_9);
theta4_2 = vpa(theta4_2 * 180/pi, 15);

T6_3_9 = cosd(theta1_1)*cosd(theta2_3 + theta3_2)*m(9) +
sind(theta1_1)*cosd(theta2_3 + theta3_2)*m(10) -sind(theta2_3 +
theta3_2)*m(11);
T6_3_10 = -sind(theta1_1)*m(9) + cosd(theta1_1)*m(10);
theta4_3 = atan2(T6_3_10, T6_3_9);
theta4_3 = vpa(theta4_3 * 180/pi, 15);

T6_3_9 = cosd(theta1_2)*cosd(theta2_4 + theta3_2)*m(9) +
sind(theta1_2)*cosd(theta2_4 + theta3_2)*m(10) -sind(theta2_4 +
theta3_2)*m(11);
T6_3_10 = -sind(theta1_2)*m(9) + cosd(theta1_2)*m(10);
theta4_4 = atan2(T6_3_10, T6_3_9);
theta4_4 = vpa(theta4_4 * 180/pi, 15);

theta4_5 = theta4_1 -180;
theta4_6 = theta4_2 -180;
```

```
theta4_7 = theta4_3 -180;
theta4_8 = theta4_4 -180;

%theta5 八種解
% 定義八組 theta 值
thetas = [
    [theta1_1, theta2_1, theta3_1, theta4_1];
    [theta1_2, theta2_2, theta3_1, theta4_2];
    [theta1_1, theta2_3, theta3_2, theta4_3]
    [theta1_2, theta2_4, theta3_2, theta4_4]
    [theta1_1, theta2_1, theta3_1, theta4_5]
    [theta1_2, theta2_2, theta3_1, theta4_6]
    [theta1_1, theta2_3, theta3_2, theta4_7]
    [theta1_2, theta2_4, theta3_2, theta4_8]
];
d = [0, 0, 0.149, 0.433];
a = [0, 0.432, -0.02, 0];
alpha = [-90, 0, 90, -90];

% 對於每組 theta 值計算轉換矩陣
for set = 1:size(thetas, 1)
    T = eye(4);
    for i = 1:4
        theta = deg2rad(thetas(set, i)); % 轉換為弧度
        a_val = a(i);
        alpha_val = deg2rad(alpha(i)); % 轉換為弧度
        d_val = d(i);
        T_i = [cos(theta), -sin(theta)*cos(alpha_val),
sin(theta)*sin(alpha_val), a_val*cos(theta);
               sin(theta),  cos(theta)*cos(alpha_val), -
cos(theta)*sin(alpha_val), a_val*sin(theta);
               0,           sin(alpha_val),
cos(alpha_val),              d_val;
               0,           0,                           0,
1];
        T = T * T_i;
    end
```

```
    % 根據 set 值將 T 賦值給相應的變數
    switch set
        case 1
            T4_1 = T;
        case 2
            T4_2 = T;
        case 3
            T4_3 = T;
        case 4
            T4_4 = T;
        case 5
            T4_5 = T;
        case 6
            T4_6 = T;
        case 7
            T4_7 = T;
        case 8
            T4_8 = T;
    end
end

Ts = {T4_1, T4_2, T4_3, T4_4, T4_5, T4_6, T4_7, T4_8};
theta5 = zeros(1, 8);
% 計算每個 theta5 值
for i = 1:8
    T6_4 = inv(Ts{i})*m;  % 計算T6_4
    theta5(i) = double(atan2(T6_4(9), -T6_4(10)));  % 使用 atan2 進行
計算並轉換為數值
    theta5(i) = rad2deg(theta5(i));  % 將弧度轉為角度
end

theta5_1 =theta5(1);
theta5_2 =theta5(2);
theta5_3 =theta5(3);
theta5_4 =theta5(4);
theta5_5 =theta5(5);
theta5_6 =theta5(6);
theta5_7 =theta5(7);
```

```
theta5_8 =theta5(8);

%計算theta6值

% 定義八組 theta 值
thetas = [
    [theta1_1, theta2_1, theta3_1];
    [theta1_2, theta2_2, theta3_1];
    [theta1_1, theta2_3, theta3_2]
    [theta1_2, theta2_4, theta3_2]
    [theta1_1, theta2_1, theta3_1]
    [theta1_2, theta2_2, theta3_1]
    [theta1_1, theta2_3, theta3_2]
    [theta1_2, theta2_4, theta3_2]
];
d = [0, 0, 0.149, 0.433];
a = [0, 0.432, -0.02, 0];
alpha = [-90, 0, 90, -90];

% 對於每組 theta 值計算轉換矩陣
for set = 1:size(thetas, 1)
    T = eye(4);
    for i = 1:3
        theta = deg2rad(thetas(set, i)); % 轉換為弧度
        a_val = a(i);
        alpha_val = deg2rad(alpha(i)); % 轉換為弧度
        d_val = d(i);
        T_i = [cos(theta), -sin(theta)*cos(alpha_val),  sin(theta)*sin(alpha_val),
a_val*cos(theta);
               sin(theta),  cos(theta)*cos(alpha_val), -cos(theta)*sin(alpha_val),
a_val*sin(theta);
               0,          sin(alpha_val),              cos(alpha_val),
d_val;
               0,          0,                           0,
1];
        T = T * T_i;
    end
```

```
    % 根據set值將T賦值給相應的變數
    switch set
        case 1
            T3_1 = T;
        case 2
            T3_2 = T;
        case 3
            T3_3 = T;
        case 4
            T3_4 = T;
        case 5
            T3_5 = T;
        case 6
            T3_6 = T;
        case 7
            T3_7 = T;
        case 8
            T3_8 = T;
    end
end
Ts = {T3_1, T3_2, T3_3, T3_4, T3_5, T3_6, T3_7, T3_8};
theta6 = zeros(1, 8);
% 計算每個 theta6 值
for i = 1:8
    T6_3= inv(Ts{i})*m;  % 計算 T6_3
    theta6(i) = double(atan2(T6_3(7), -T6_3(3)));  % 使用 atan2 進行計算並轉換為數
值
    theta6(i) = rad2deg(theta6(i));  % 弧度轉角度
end


theta6_1 =theta6(1);
theta6_2 =theta6(2);
theta6_3 =theta6(3);
theta6_4 =theta6(4);
theta6_5 =theta6(5)-180;
theta6_6 =theta6(6)+180;
theta6_7 =theta6(7)+180;
```

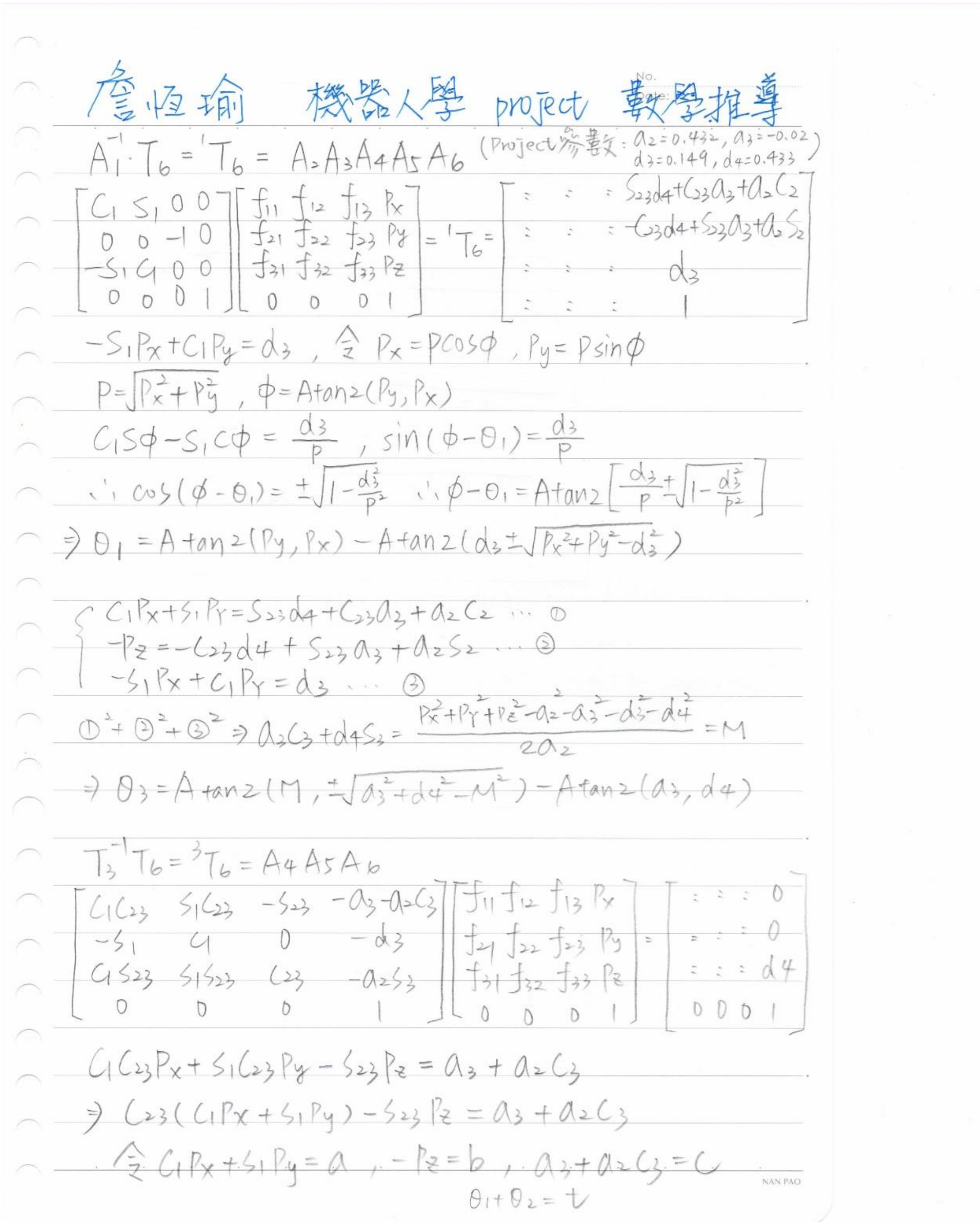
```
theta6_8 =theta6(8)-180;


%呈現最後全部結果
theta1_1 = double(theta1_1);
theta1_2 = double(theta1_2);
theta2_1 = str2double(theta2_1);
theta2_2 = str2double(theta2_2);
theta2_3 = str2double(theta2_3);
theta2_4 = str2double(theta2_4);
theta3_1 = double(theta3_1);
theta3_2 = double(theta3_2);
theta4_1 = double(theta4_1);
theta4_2 = double(theta4_2);
theta4_3 = double(theta4_3);
theta4_4 = double(theta4_4);
theta4_5 = double(theta4_5);
theta4_6 = double(theta4_6);
theta4_7 = double(theta4_7);
theta4_8 = double(theta4_8);

% 定義 theta 值和限制範圍
theta_limits = [-160, 160; -125, 125; -135, 135; -140, 140; -100,
100; -260, 260];
thetas = [
    theta1_1, theta2_1, theta3_1, theta4_1, theta5_1, theta6_1;
    theta1_2, theta2_2, theta3_1, theta4_2, theta5_2, theta6_2;
    theta1_1, theta2_3, theta3_2, theta4_3, theta5_3, theta6_3;
    theta1_2, theta2_4, theta3_2, theta4_4, theta5_4, theta6_4;
    theta1_1, theta2_1, theta3_1, theta4_5, theta5_5, theta6_5;
    theta1_2, theta2_2, theta3_1, theta4_6, theta5_6, theta6_6;
    theta1_1, theta2_3, theta3_2, theta4_7, theta5_7, theta6_7;
    theta1_2, theta2_4, theta3_2, theta4_8, theta5_8, theta6_8
];

% 打印 theta 變數名稱
disp('Corresponding variable (theta1, theta2, theta3, theta4, theta5,
theta6)');
```

```
% 印出每組 theta
for i = 1:size(thetas, 1)
    % 未超出範圍的 theta 值生成警告字符串
    out_of_range_msg = "";
    for j = 1:size(thetas, 2)
        if thetas(i, j) < theta_limits(j, 1) || thetas(i, j) >
theta_limits(j, 2)
            out_of_range_msg = strcat(out_of_range_msg,
sprintf("theta%d is out of range!\n", j));
        end
    end

    % 如果有超出範圍的值，先打印警告
    if out_of_range_msg ~= ""
        fprintf('\n');
        disp('Corresponding variable (theta1, theta2, theta3, theta4,
theta5, theta6)');
        fprintf('%s', out_of_range_msg);
    else
        fprintf('\n');
    end

    fprintf('%.4f %.4f %.4f %.4f %.4f %.4f\n', thetas(i, :));
end
```

首先設定 DH 參數，並要求使用者輸入[n o a p]矩陣，計算細節中的 theta1 利用講義中的推導得到 theta1，將其數學公式輸入來計算出解，接著 theta3 同樣使用講義的公式推導出 theta3 的解，再來就是最複雜的 theta2，我程式中是利用 Matlab 中的 fzero 函數來得到它的解，但我相關公式推導有寫在紙上，因為帶入有些問題，所以暫時用 fzero 迭代的方式來求解，接著 theta4 相對簡單很多，同樣用課本上的公式把前面求出的 theta1、2、3 代入求出 T6_3，如此就可以相關公式求出 Theta4 的 4 個解，後四個解則和前四解互補，所以將前四解減 180 就可以得到後四組解，theta5 和 theta4 前面錯法很像，將所得到的解代求得 T6_4 之後，就可以利用其中的 atan2 關係得到 theta5 的解，最後 theta6 我利用講義上的公式找出兩數值的關係，且同樣使用 atan2 來求出我的值，要注意角度的象限位置，最後再上述所求得的解先轉為數值，之後一一把他們都呈現出來，總共有八組解，同時在打印時也有比較他們的值是否有在我

們的範圍內，沒有的話就會呈現出來。

(三)數學運算說明



詹恆瑜 機器人學 project 數學推導

$A_1^{-1}\cdot T_6 = {}^1T_6 = A_2A_3A_4A_5A_6$ (Project參數: $a_2=0.432$, $a_3=-0.02$, $d_3=0.149$, $d_4=0.433$)

$$\begin{bmatrix} C_1 & S_1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ -S_1 & C_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} f_{11} & f_{12} & f_{13} & P_x \\ f_{21} & f_{22} & f_{23} & P_y \\ f_{31} & f_{32} & f_{33} & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = {}^1T_6 = \begin{bmatrix} : & : & : & S_{23}d_4+C_{23}a_3+a_2C_2 \\ : & : & : & -C_{23}d_4+S_{23}a_3+a_2S_2 \\ : & : & : & d_3 \\ : & : & : & 1 \end{bmatrix}$$

$-S_1P_x+C_1P_y=d_3$，令 $P_x=\rho\cos\phi$，$P_y=\rho\sin\phi$

$\rho=\sqrt{P_x^2+P_y^2}$，$\phi=A\tan2(P_y,P_x)$

$C_1S\phi-S_1C\phi=\frac{d_3}{\rho}$，$\sin(\phi-\theta_1)=\frac{d_3}{\rho}$

$\therefore \cos(\phi-\theta_1)=\pm\sqrt{1-\frac{d_3^2}{\rho^2}}$ $\therefore \phi-\theta_1=A\tan2\left[\frac{d_3}{\rho},\pm\sqrt{1-\frac{d_3^2}{\rho^2}}\right]$

$\Rightarrow \theta_1=A\tan2(P_y,P_x)-A\tan2(d_3,\pm\sqrt{P_x^2+P_y^2-d_3^2})$

$$\begin{cases} C_1P_x+S_1P_y=S_{23}d_4+C_{23}a_3+a_2C_2 & \cdots ① \\ -P_z=-C_{23}d_4+S_{23}a_3+a_2S_2 & \cdots ② \\ -S_1P_x+C_1P_y=d_3 & \cdots ③ \end{cases}$$

$①^2+②^2+③^2 \Rightarrow a_2C_3+d_4S_3=\frac{P_x^2+P_y^2+P_z^2-a_2^2-a_3^2-d_3^2-d_4^2}{2a_2}=M$

$\Rightarrow \theta_3=A\tan2(M,\pm\sqrt{a_3^2+d_4^2-M^2})-A\tan2(a_3,d_4)$

$T_3^{-1}T_6={}^3T_6=A_4A_5A_6$

$$\begin{bmatrix} C_1C_{23} & S_1C_{23} & -S_{23} & -a_3-a_2C_3 \\ -S_1 & C_1 & 0 & -d_3 \\ C_1S_{23} & S_1S_{23} & C_{23} & -a_2S_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} f_{11} & f_{12} & f_{13} & P_x \\ f_{21} & f_{22} & f_{23} & P_y \\ f_{31} & f_{32} & f_{33} & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} : & : & : & 0 \\ : & : & : & 0 \\ : & : & : & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$C_1C_{23}P_x+S_1C_{23}P_y-S_{23}P_z=a_3+a_2C_3$

$\Rightarrow C_{23}(C_1P_x+S_1P_y)-S_{23}P_z=a_3+a_2C_3$

令 $C_1P_x+S_1P_y=a$，$-P_z=b$，$a_3+a_2C_3=c$

$\theta_1+\theta_2=t$

令 $\tan(\frac{t}{2}) = k$, $\cos t = \frac{1-k^2}{1+k^2}$

$\sin t = \frac{2k}{1+k^2}$

No.

Date: / /

$$\Rightarrow a\cos t + b\sin t = c$$

$$\Rightarrow a\frac{1-k^2}{1+k^2} + b\frac{2k}{1+k^2} = c$$

$$\Rightarrow (a+c)k^2 - 2bk + (c-a) = 0$$

$$\Rightarrow k = \frac{2b \pm \sqrt{4b^2 - 4(a+c)(c-a)}}{2(a+c)} = \frac{b \pm \sqrt{b^2+a^2-c^2}}{a+c}$$

$$\Rightarrow t = 2\tan^{-1}\left(\frac{b \pm \sqrt{b^2+a^2-c^2}}{a+c}\right)$$

$$\Rightarrow \theta_2 + \theta_3 = 2\tan^{-1}\left(\frac{-P_z \pm \sqrt{P_z^2 + (C_1P_x + S_1P_y)^2 - (a_3 + a_2C_3)^2}}{C_1P_x + S_1P_y + a_3 + a_2C_3}\right)$$

$$\Rightarrow \theta_2 = 2\tan^{-1}\left(\frac{-P_z \pm \sqrt{P_z^2 + (C_1P_x + S_1P_y)^2 - (a_3 + a_2C_3)^2}}{C_1P_x + S_1P_y + a_3 + a_2C_3}\right) - \theta_3$$

$$^3T_6 = \begin{bmatrix} C_4C_5C_6 - S_4S_6 & -C_4C_5S_6 - S_4C_6 & C_4S_5 & 0 \\ S_4C_5C_6 + C_4S_6 & -S_4C_5S_6 + C_4C_6 & S_4S_5 & 0 \\ -S_5C_6 & S_5S_6 & C_5 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\Rightarrow \tan\theta_4 = \frac{S_4S_5}{C_4S_5} = \frac{S_4}{C_4} \Rightarrow \theta_4 = \tan^{-1}\left(\frac{S_4}{C_4}\right)$$
$$= 2\tan(S_4, C_4)$$

$$^4T_6 = \begin{bmatrix} \vdots & \vdots & S_5 & 0 \\ \vdots & \vdots & -C_5 & 0 \\ \vdots & \vdots & 0 & 0 \\ \vdots & \vdots & 0 & 1 \end{bmatrix}$$

$$\Rightarrow -\tan\theta_5 = \frac{S_5}{C_5} \Rightarrow \theta_5 = \tan^{-1}\left(\frac{-S_5}{C_5}\right) = -\tan^{-1}\left(\frac{S_5}{C_5}\right)$$

$$\begin{cases} -C_4S_6 - S_4C_6 = -S_{46} \cdots ⑤ \\ -S_4S_6 + C_4C_6 = C_{46} \cdots ⑥ \end{cases}$$

$$\frac{⑤}{⑥} = \tan(\theta_4+\theta_6) = \frac{C_4S_6+S_4C_6}{C_4C_6-S_4S_6}$$

$$\Rightarrow \theta_6 = (\theta_4+\theta_6) - \theta_4$$

(四)加分題-兩種逆向運動學的優缺點

代數法

優點:

1. 在複雜且自由度多的機器人結構中，代數法可以有效的處理他們的運動問

題。
2.往往代數法因為透過精確的數學公式才能得到相關的解，所以得到的解會比較為精確。
3.利於我們在程式中直接透過公式輸入做計算，用電腦計算就會快很多。
缺點:
1.越多自由度關節的機器人，他的數學公式就會更加複雜。
2.因為數學公式繁瑣，所以計算量很大，人工會不好處理計算。

幾何法:
優點:
1.相對沒有繁瑣的計算且想法較為直觀。
2.可以簡化一些複雜的模型來計算。
3.不用繁瑣的數學計算也可以得到相關的角度。
缺點:
1.在高度複雜的機器人中會很難使用幾何法，甚至精度會比代數法還差。
2.在每個機器人中的幾何算法可能會不一樣。