機器人動力與控制

FINAL PROJECT

王奕勝　陳若桐　林昱宏　武敬祥

OUTLINE

- 期末專題任務
- 硬體及軟體
- Flow Chart
- 策略及演算法
- 影像辨識
- Q & A
- 實際運作情形

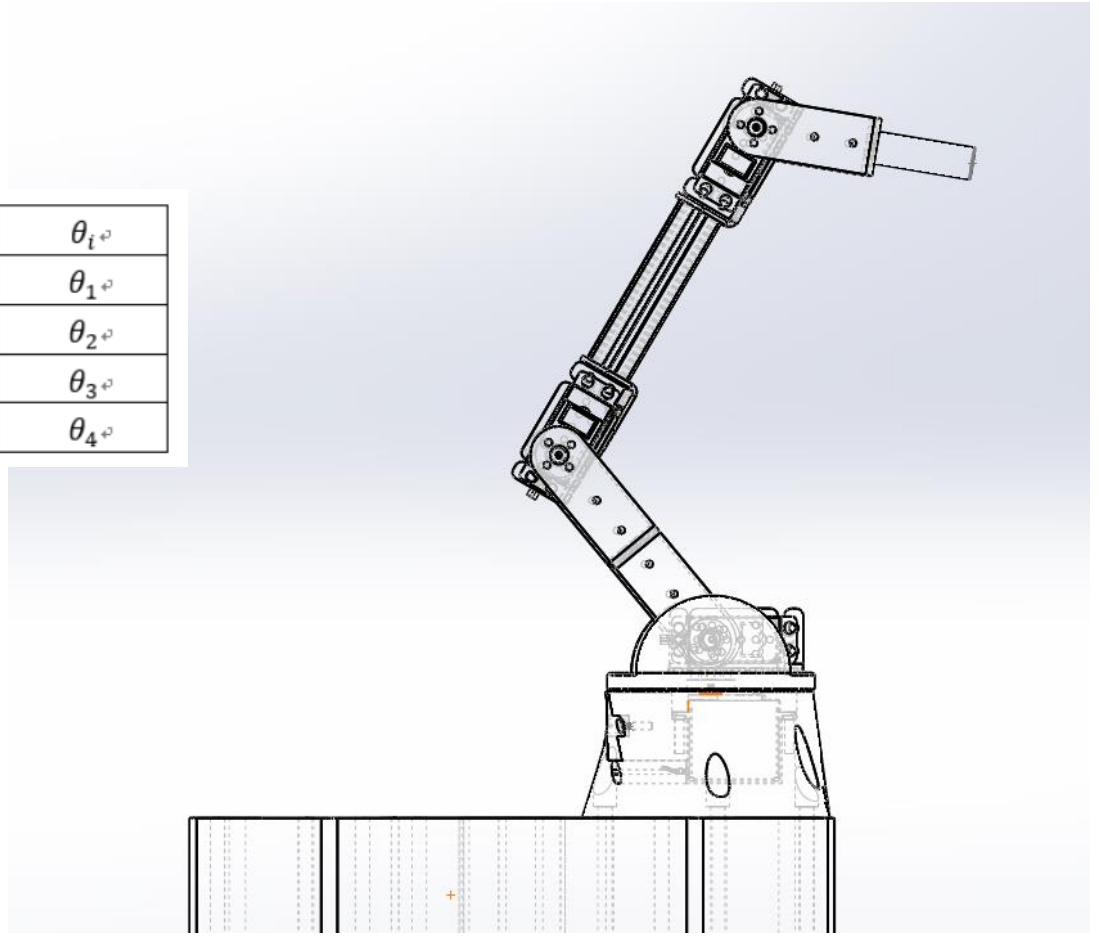
任務及流程

- 讓車體向前移動，同時操控4R手臂使其能在停到目標物前時，讓手臂插進目標物之洞裡
- 車體擺放在某一位置（可看到目標物）→車子移動到目標物旁→手臂移動並依序插進目標物的洞裡（大中小）

硬體—設計和零件組

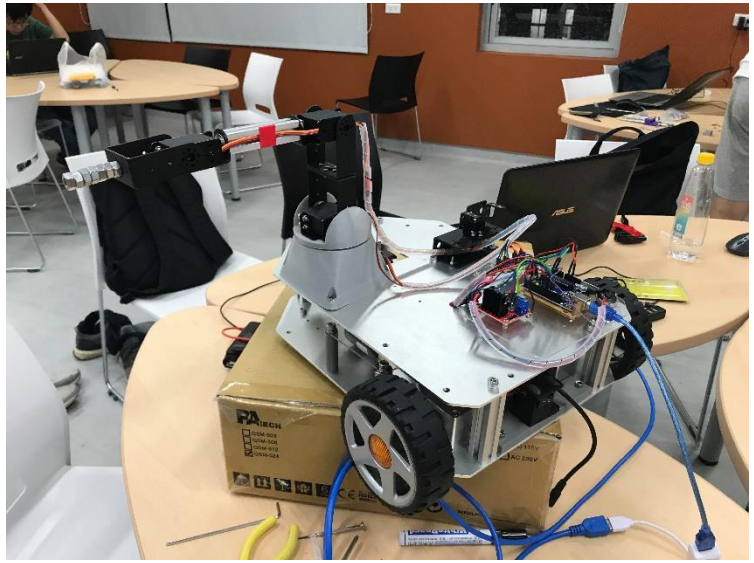
- 車體
- 手臂
- 底座
- 馬達

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	90°	0	23	θ_1
2	0	100.4	0	θ_2
3	0	170	0	θ_3
4	0	95	0	θ_4

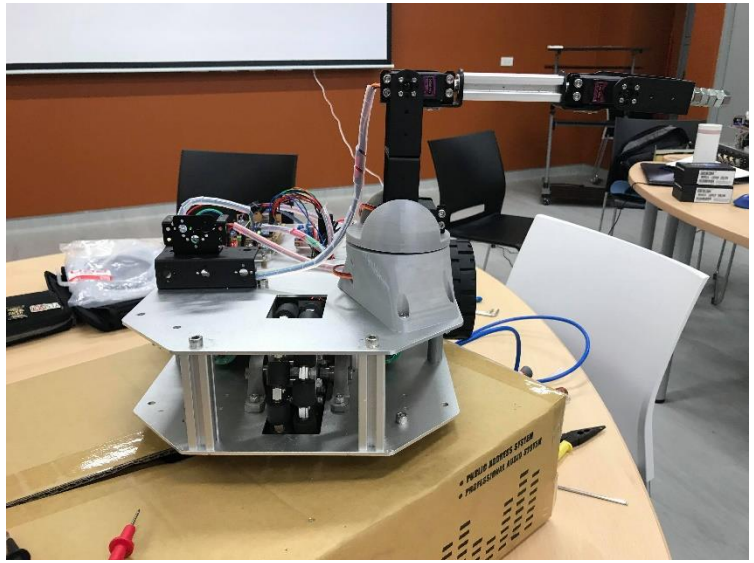


車體

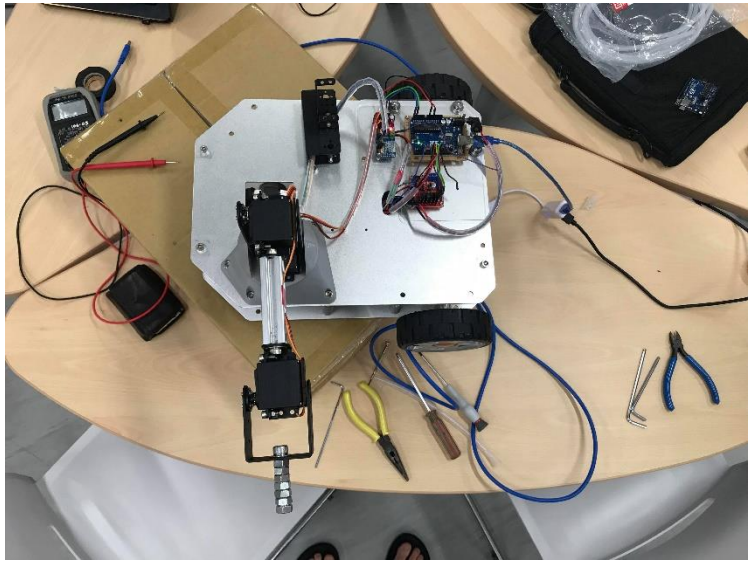
側視圖



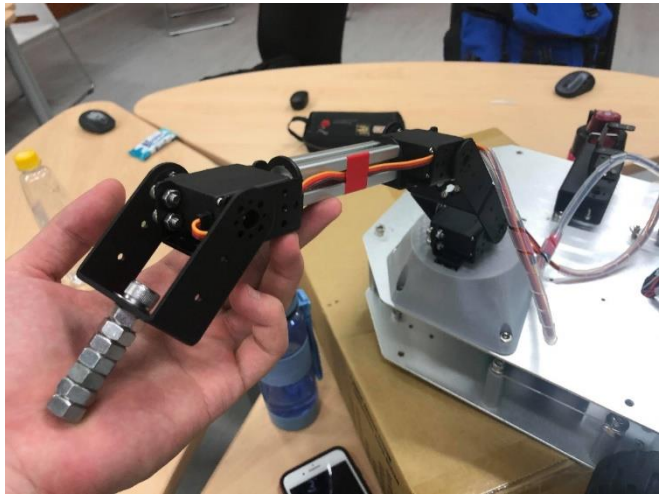
正視圖



上視圖

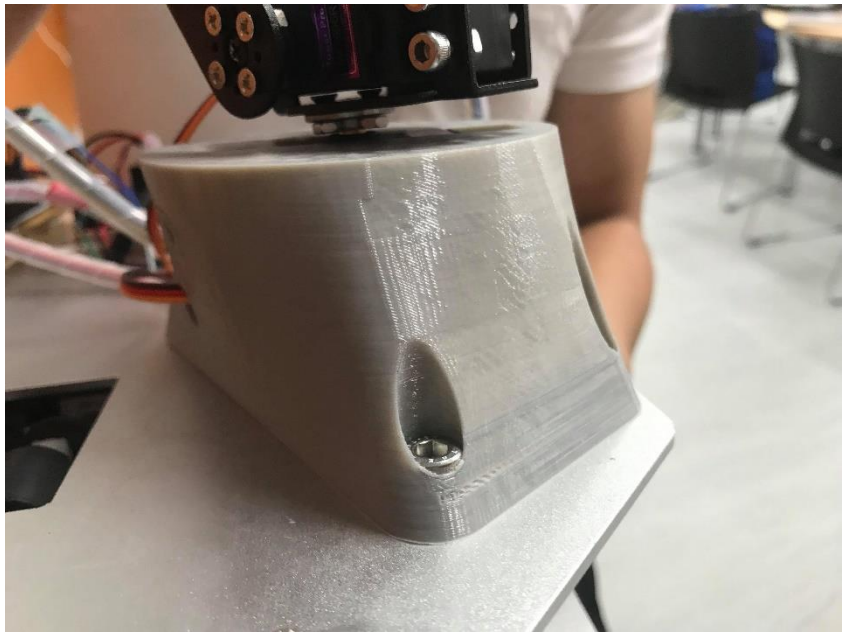


手臂

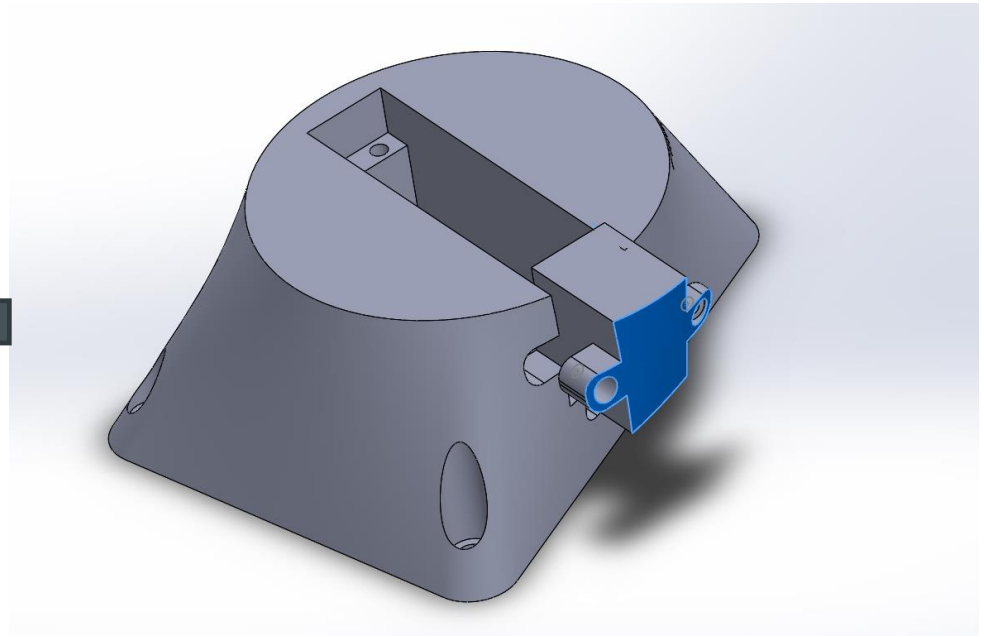


底座

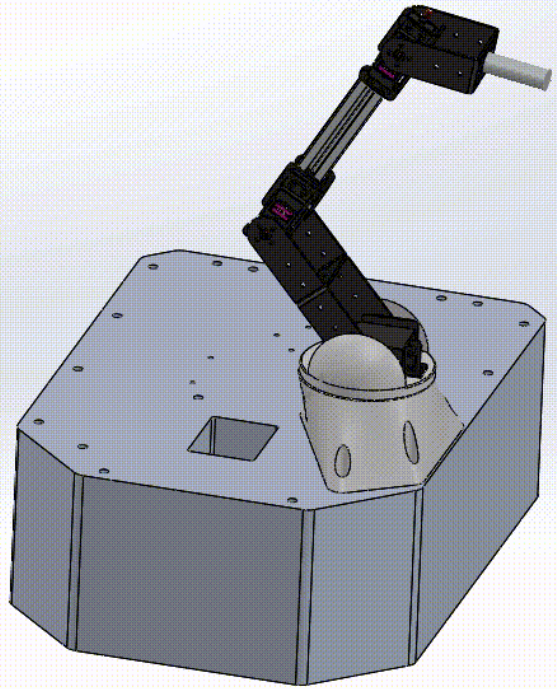
實際圖



模擬圖



馬達– MG996R、24V DC MOTOR



軟體應用

- Python (上層決策)
- Arduino(底層控制)
- Matlab (模擬)

MATLAB

Inverse Kinematics

Inverse kinematics solution

已知 X, Y, Z 和 $\phi = \theta_2 + \theta_3 + \theta_4$

$$\theta_1 = \tan^{-1}\left(\frac{Y}{X}\right)$$

計算

$$A = X - l_4 \cos\theta_1 \cos\phi$$
$$B = Y - l_4 \sin\theta_1 \cos\phi$$
$$C = Z - l_1 - l_4 \sin\phi$$

得出

$$\theta_3 = \cos^{-1}\frac{A^2 + B^2 + C^2 - l_2^2 - l_3^2}{2\, l_2\, l_3}$$

再計算

$$a = l_3 \sin\theta_3$$
$$b = l_2 + l_3 \cos\theta_3$$
$$c = Z - l_1 - l_4 \sin\phi$$
$$r = \sqrt{a^2 + b^2}$$

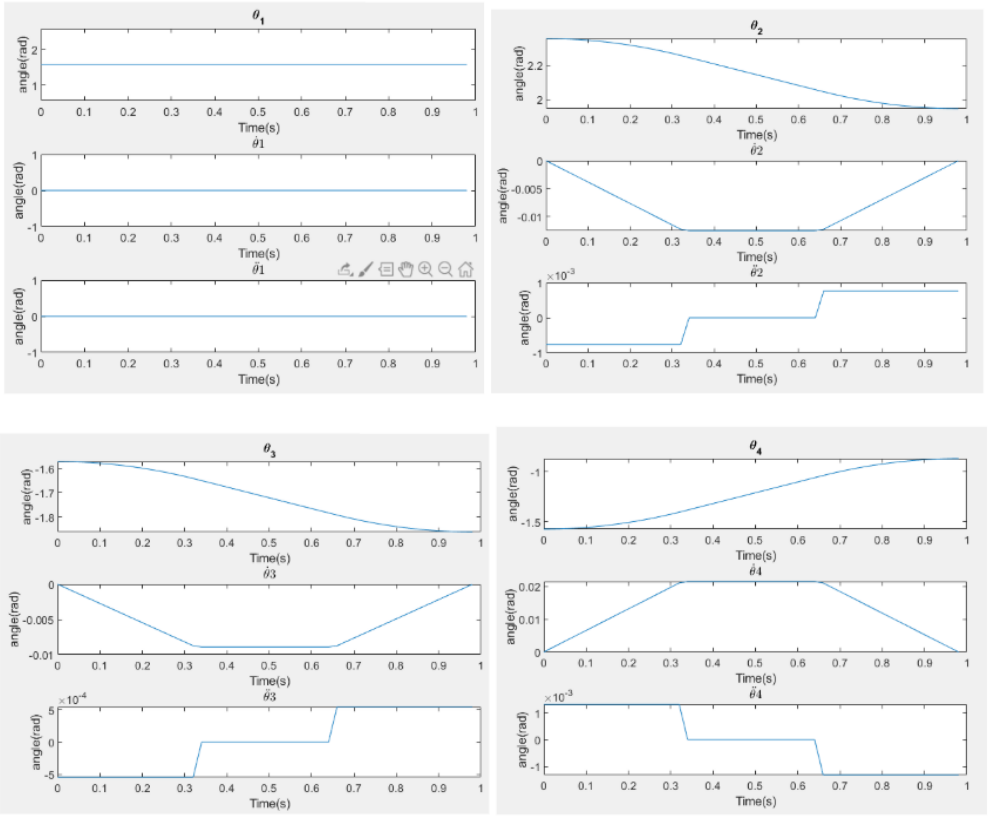
得出

$$\theta_2 = \tan^{-1}\frac{c}{\sqrt{r^2 - c^2}_3} - \tan^{-1}\frac{a}{b}$$

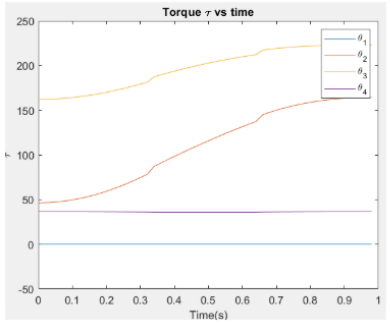
最後可知

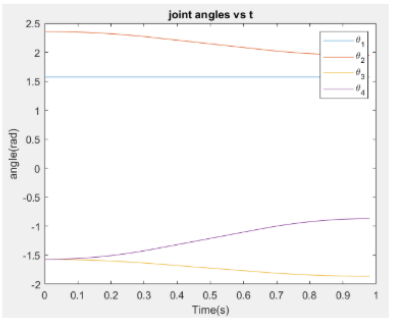
$$\theta_4 = \phi - \theta_2 - \theta_3$$

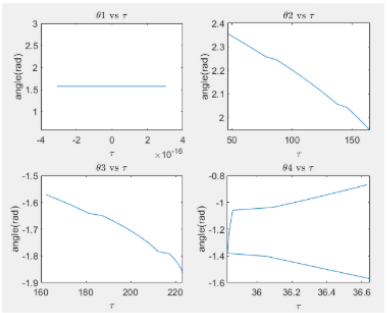
Kinematics simulation



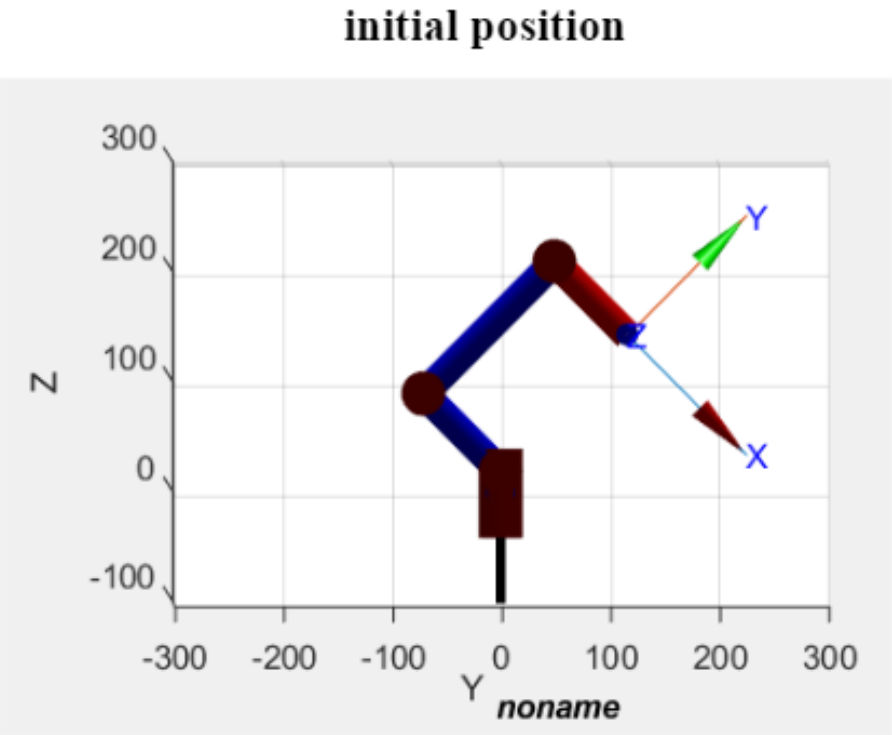
Dynamics Simulation

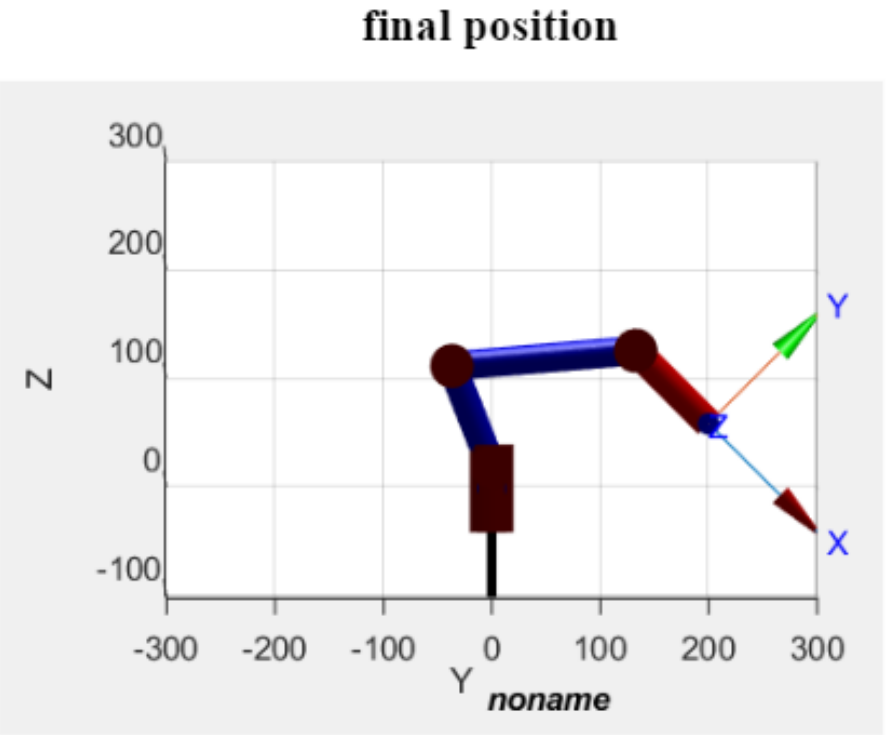






MATLAB


initial position
300
200
100
0
-100
Z
-300
-200
-100
0
100
200
300
Y
noname
Y
X


final position
300
200
100
0
-100
Z
-300
-200
-100
0
100
200
300
Y
noname
Y
X

座標系轉換

Coordinate Transform

像素座標系 → 影像座標系 → 相機坐標系 → 世界座標系

世界座標 → 像素座標

$$\begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = \begin{bmatrix} {}^{W}_{C}R & {}^{W}P_{CORG} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{Z_c}{camera\_fx} & 0 & \frac{-Z_c camera\_cx}{camera\_fx} \\ 0 & \frac{Z_c}{camera\_fy} & \frac{-Z_c camera\_cy}{camera\_fy} \\ 0 & 0 & Z_c \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

外部參數　　內部參數

內部參數(intrinsic parameter):

Realsense d435i

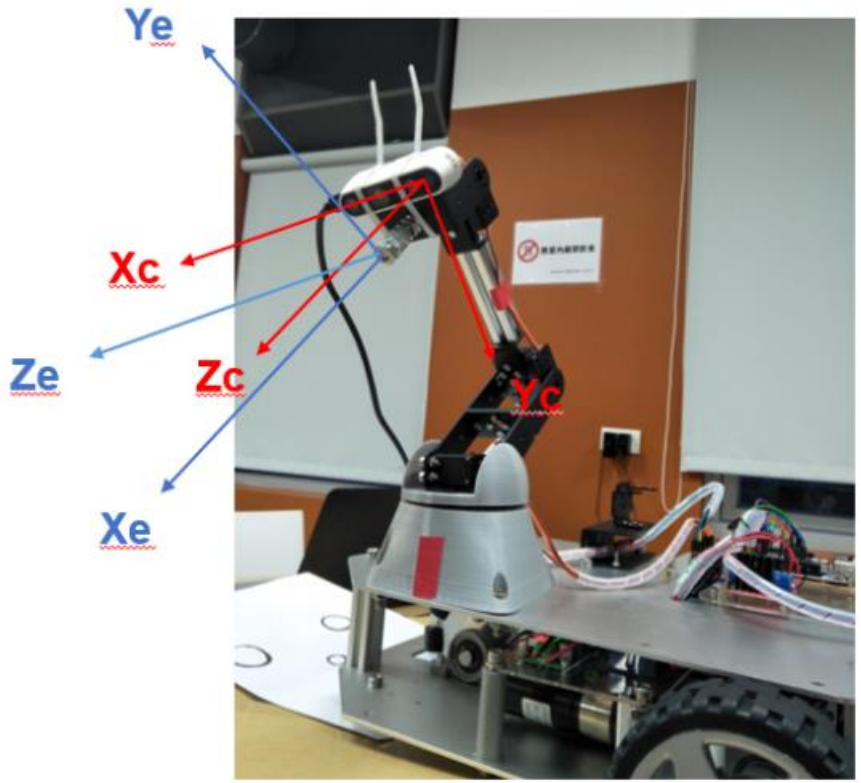
```
camera_cx = 321.798
camera_cy = 239.607
camera_fx = 615.899
camera_fy = 616.468
```

外部參數(extrinsic parameter):

外部參數=手臂順向Transform matrix * Rotation matrix

Rotation matrix:使相機坐標系與手臂end_effeter 的方向一致

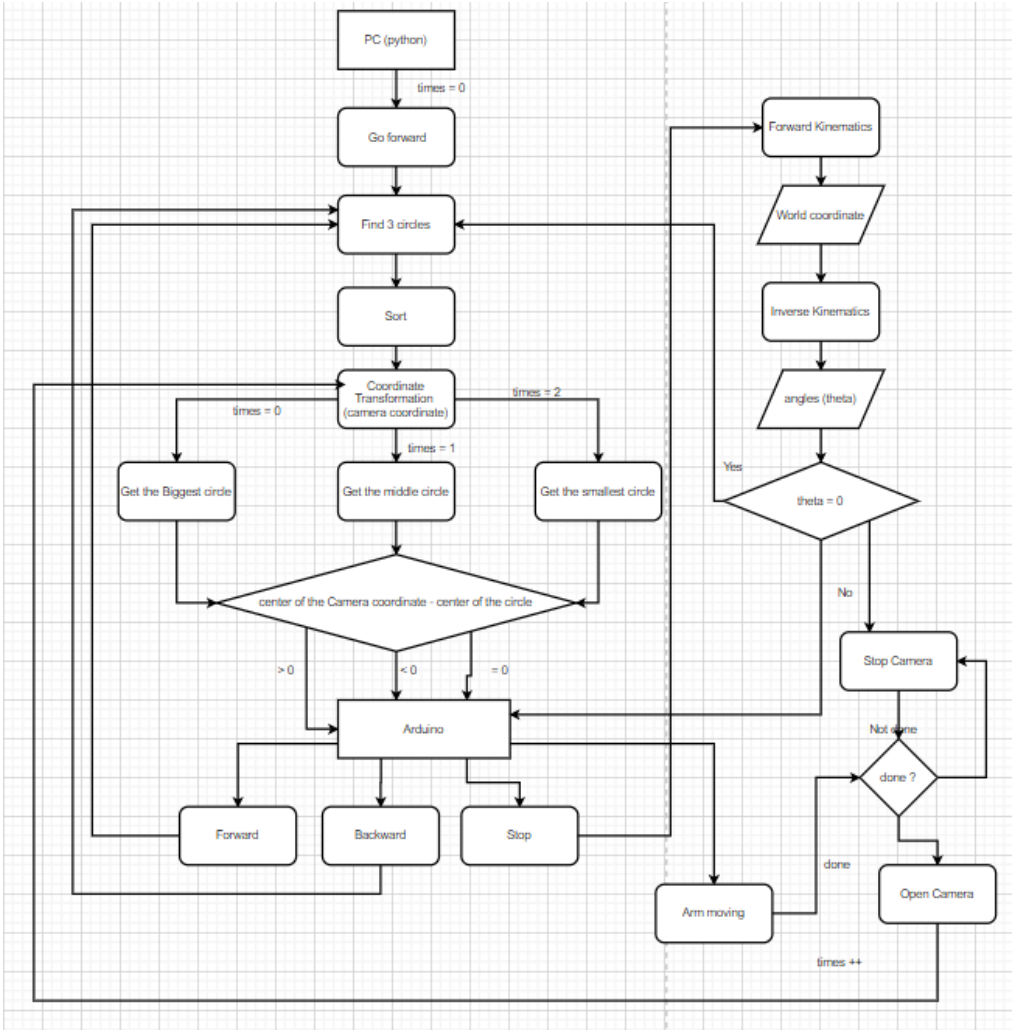
座標系轉換

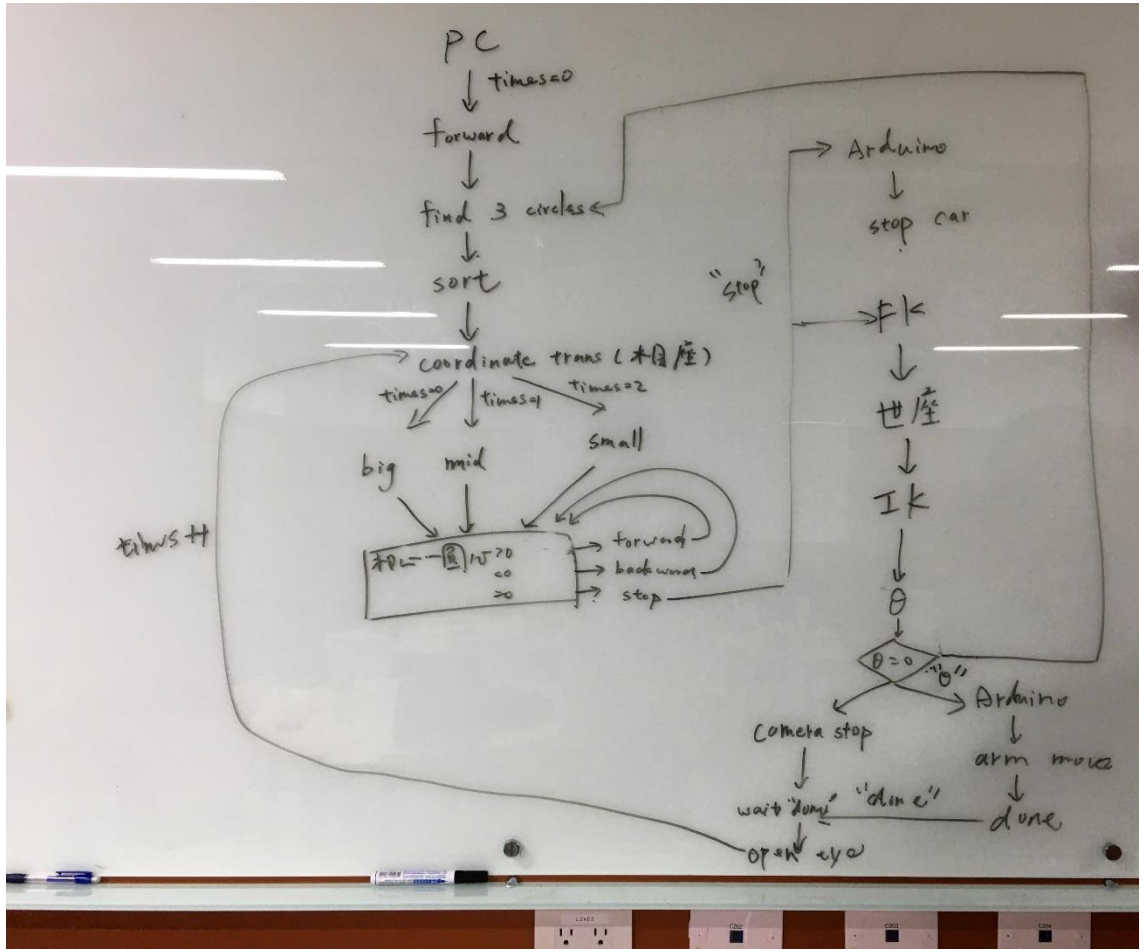


(Xc,Yc,Zc):相機坐標系

(Xe,Ye,Ze):end_effecter坐標系

FLOW CHART


PC (python)
times = 0
Go forward
Find 3 circles
Sort
Coordinate Transformation (camera coordinate)
times = 0
times = 1
times = 2
Get the Biggest circle
Get the middle circle
Get the smallest circle
center of the Camera coordinate - center of the circle
> 0
< 0
= 0
Arduino
Forward
Backward
Stop
Forward Kinematics
World coordinate
Inverse Kinematics
angles (theta)
Yes
theta = 0
No
Stop Camera
Not done
done ?
done
Arm moving
Open Camera
times ++


PC
times=0
forward
find 3 circles
sort
coordinate trans (相機座)
times=0
times=1
times=2
big
mid
small
times++
forward
backward
stop
Arduino
stop car
"stop"
FK
世座
IK
θ
θ=0
"0"
Arduino
arm move
done
Camera stop
wait "done"
"done"
open cv

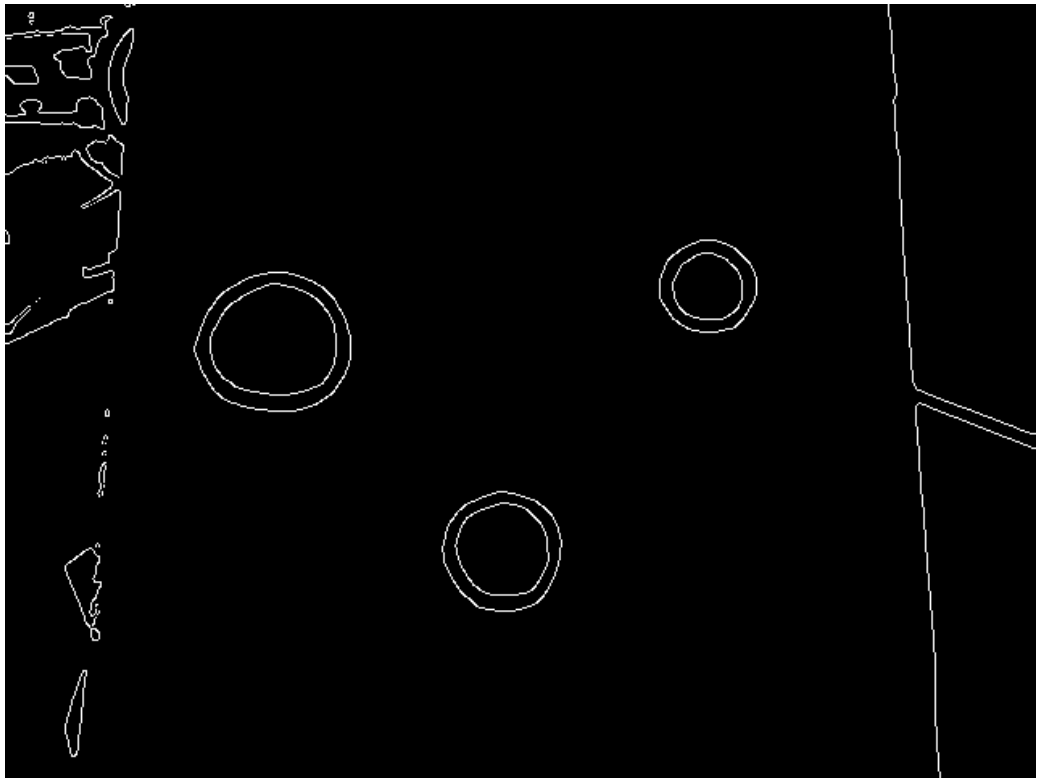
車體控制

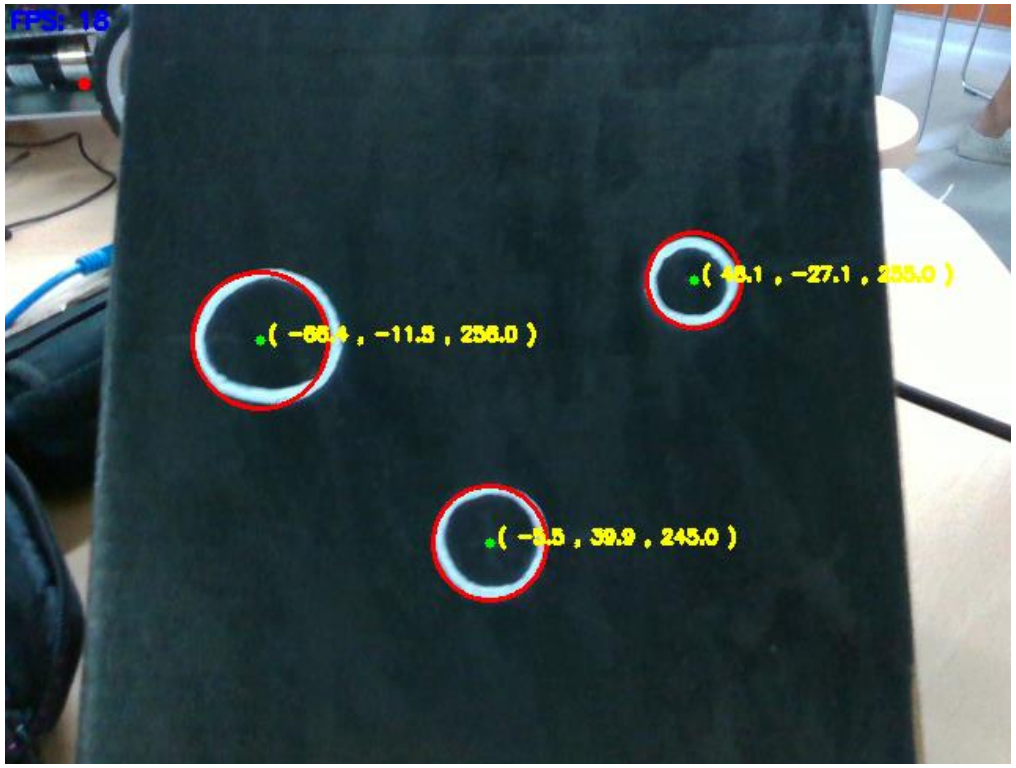
pseudo code:

```
目標物在車體右邊
k = 0.30        #gain
if 找到三個圓:
    將圓從大排到小
    if (相機中心-圓中心) < 30mm:
        velocity = 0
    else :
        velocity = (相機中心-圓中心) * k

    if velocity > 0:
        go backward
    else:
        go forward
else:
    跑很快
```

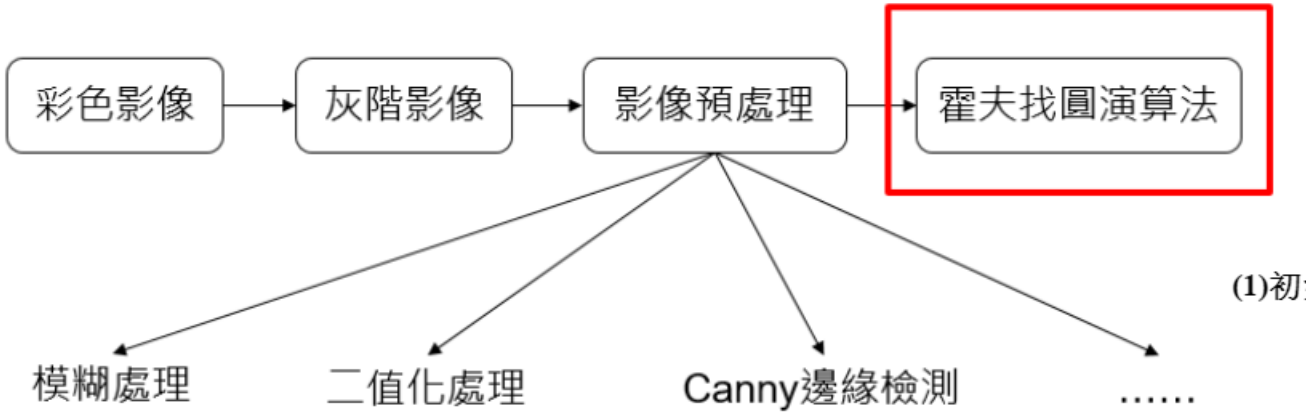
```
k = 0.5
if np.size(coordinate,0) == 3:
    coordinate = rad_sort(coordinate)
    if -coordinate[0][0] < 10:
        vel = 0
    else :
        vel = -coordinate[0][0] * k

    if vel < 0:
        vel = -vel
        velocity = str(vel) + "," + "1\n"
    else:
        velocity = str(vel) + "," + "0\n"

    ser.write(velocity.encode())
else:
    ser.write(("50,0\n").encode())
```

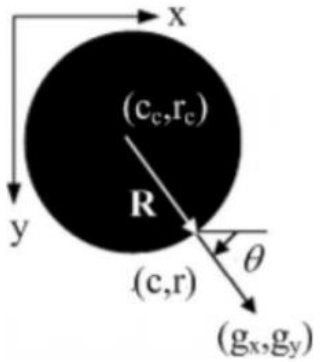
影像辨識





影像辨識演算法





$$(c-c_c)^2+(r-r_c)^2=R^2$$

$$c_c=c-R\cos\theta$$

$$r_c=r-R\sin\theta$$

$$\cos\theta=\frac{g_x}{\sqrt{g_x^2+g_y^2}}$$

$$\sin\theta=\frac{g_y}{\sqrt{g_x^2+g_y^2}}$$

gx : gradient in x axis

gy : gradient in y axis

Implementation

(1)初始化一個三維累加器accu=(X_C , Y_C , R)，分別代表圓心X座標，圓心Y 座標及圓半徑R

(2)利用 sobel filter 取出圓形的gx ,gy

(3)給定圓半徑R的最大Rmax及最小值Rmin

(4)利用下式:

$$X_C = x - Rcos\theta$$

$$Y_C = y - Rsin\theta$$

對每一個R及圓心pixel座標(x,y)求出圓心(Xc,Yc)

(5)將累加器accu(X_C , Y_C , R)+1

(6)得到accu最大的值及求得圓的圓心及半徑

Q&A

實際運作情形

- 請見7/3下午兩點半

THANKS FOR LISTENING