

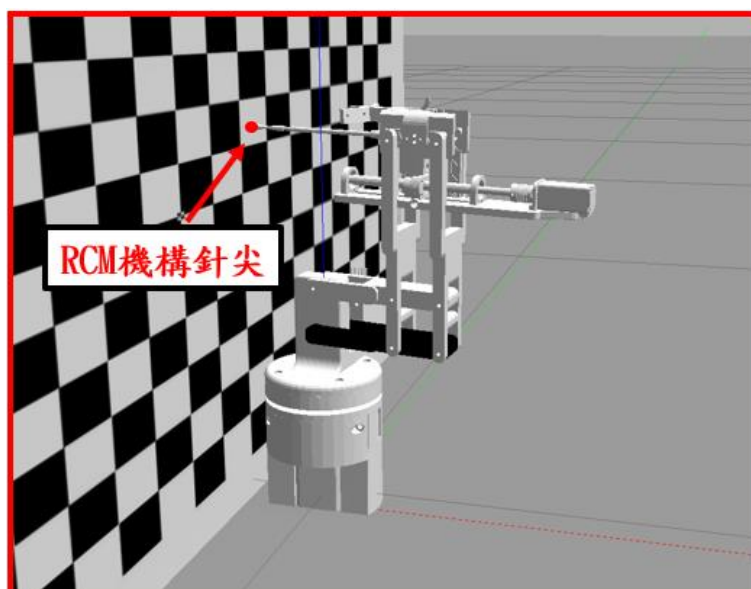
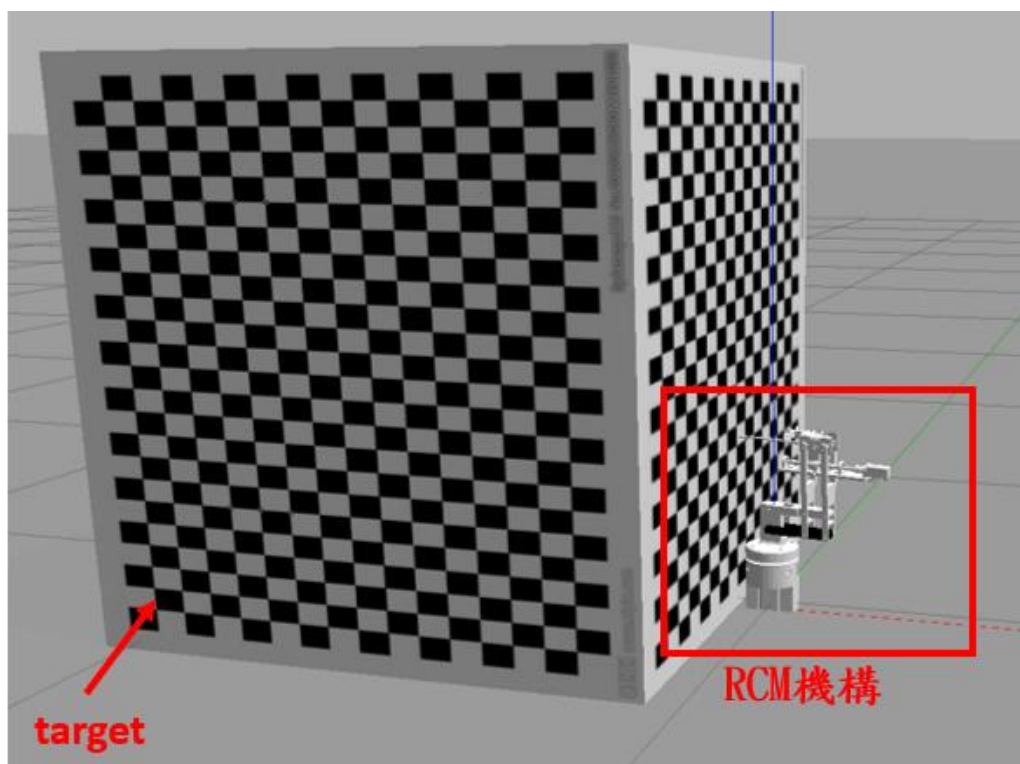
暑期 side project-RCM 機構虛實整合

暑期 SIDE PROJECT-RCM 機構虛實整合	1
專題目標	2
GAZEBO WORLD.....	2
相機畫面	3
介紹	3
信號傳遞架構	3
模擬平台&工具	4
遙控功能	4
機構.....	4
RCM 機構.....	5
線性致動器	5
機器人學	6
FORWARD KINEMATIC:.....	6
INVERSE KINEMATIC:.....	7
相機座標轉換	8
畫素座標	8
影像座標	9
相機座標	9
世界座標	10
控制系統(未完成).....	11
機器學習(未完成).....	11
介面設計(未完成).....	11

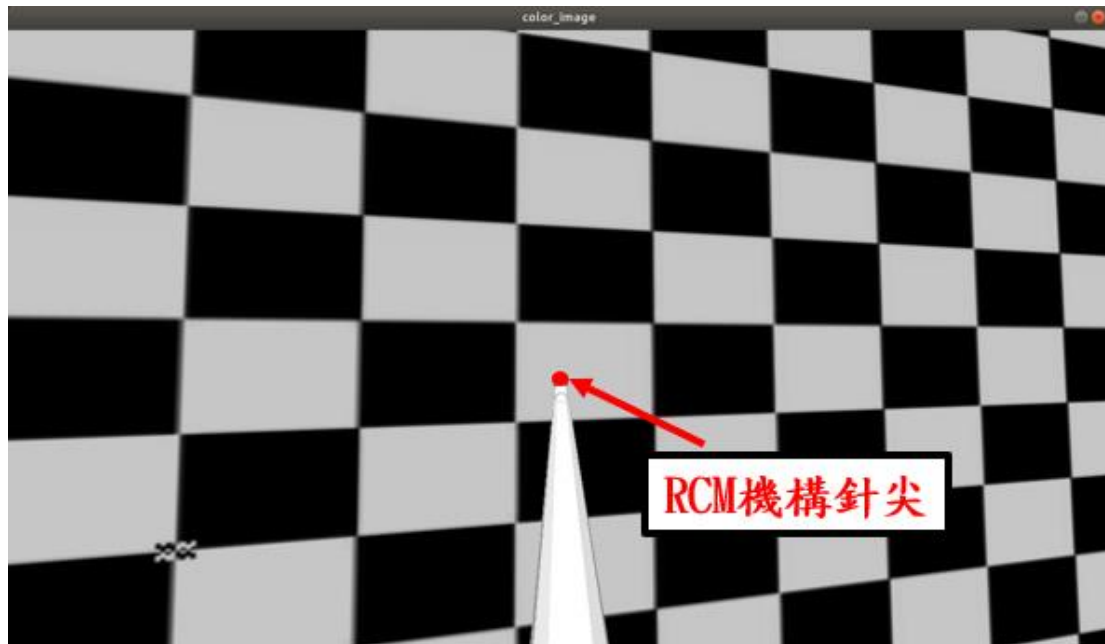
專題目標

點擊相機畫面中的目標物使 RCM 機構針尖可以自動到達點擊的座標位置

GAZEBO world



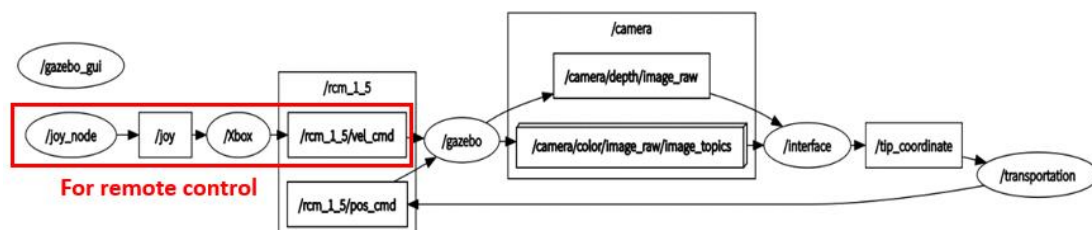
相機畫面



介紹

這是我在大四暑假時自己做的一個小小的 side project，當初做這個 project 的目的是想要利用這個 project 綜合複習大學以來學到的有關機器人的知識，比如 ROS、機器人學(順逆向運動學)、相機座標轉換、控制系統、機器學習、介面設計。並練習使用 GAZEBO 機器人模擬軟體

信號傳遞架構



gazebo：發布相機資訊(深度及彩色)

interface：顯示照片，當滑鼠點擊照片時，計算點擊位址的座標，並發布針尖座標

transportation : 接收針尖座標並計算順逆項運動學，發布機器人姿態。

模擬平台&工具

GAZEBO

ROS1 melodic

Python

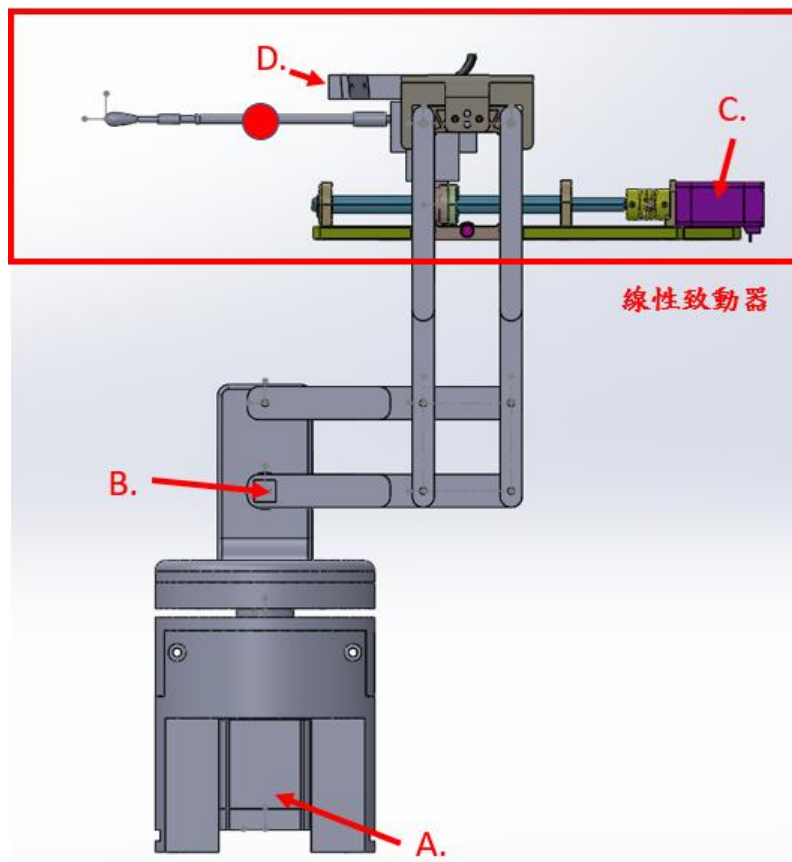
遙控功能

提供利用 Xbox one joysticks 對此機器人進行操作的選項

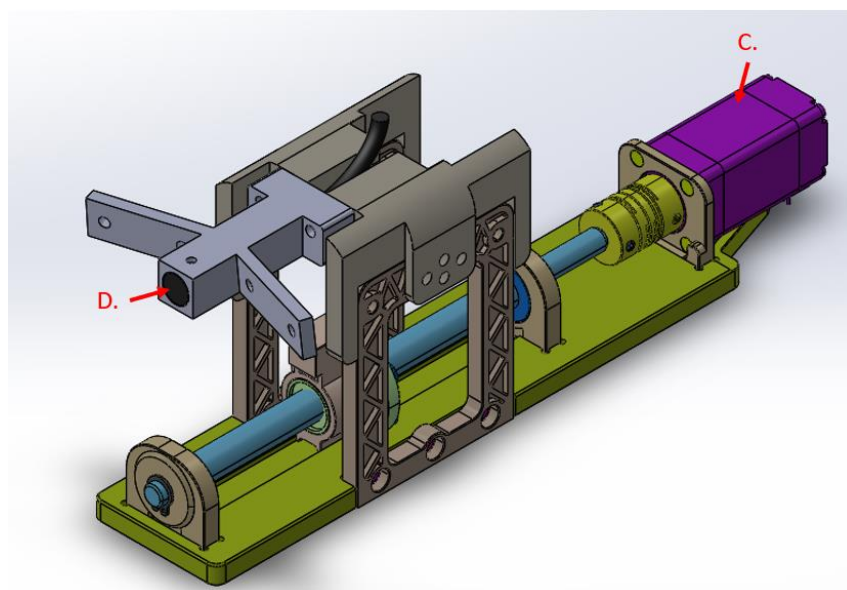
機構

此機器人為一 RCM(remote center motion)機構，此機構特性為有一運動中心(下圖紅點)為靜止不動的，在(A.)、(B.)各有一顆馬達，提供其兩個方向的旋轉自由度。末端有一線性的致動器，(C.)的馬達則提供其線性自由度，此外在此線性致動器末端(D.)處有一相機。

RCM 機構



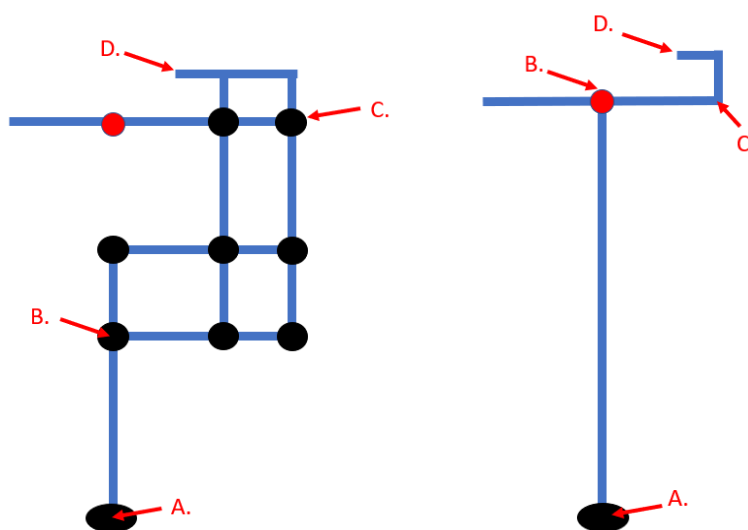
線性致動器



機器人學

Forward kinematic:

實際 RCM 機構簡圖如下圖左，且其運動學模型可經由右下圖的簡化模型較簡易求得



因此在解順向運動學時可將此機構視為一 RRP 三自由度的機器手臂，以簡圖建立座標系後可藉由以下轉換矩陣(fixed angle)，求出順向運動學

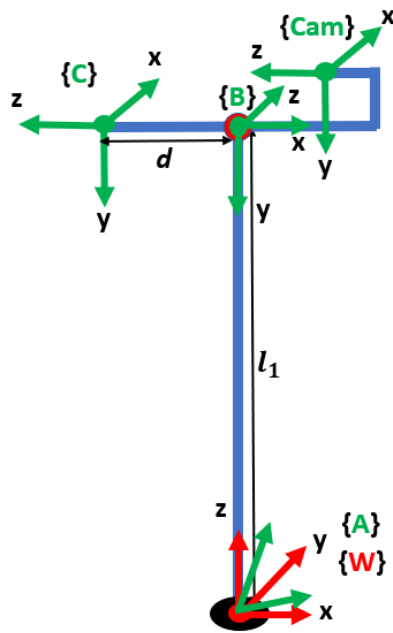
$${}^W T = {}^W T \cdot {}^A T \cdot {}^B T$$

其中

$${}^W T = \begin{bmatrix} \cos \theta_1 & -\sin \theta_1 & 0 & 0 \\ \sin \theta_1 & \cos \theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^A T = \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ -\sin \theta_1 & -\cos \theta_2 & 0 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^B T = \begin{bmatrix} 0 & 0 & -1 & -d \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



θ_1 : A 馬達提供的旋轉角度

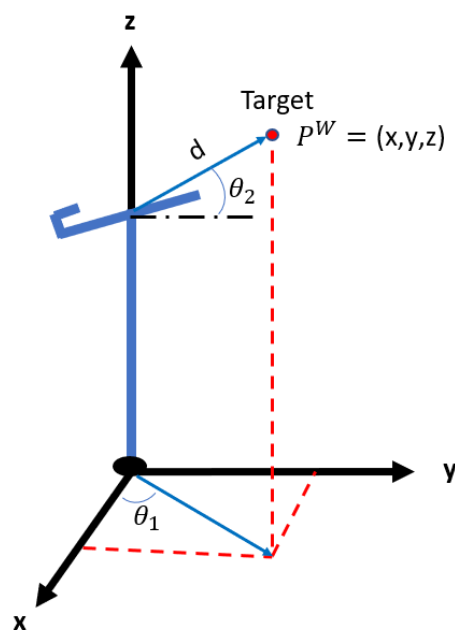
θ_2 : B 馬達提供的旋轉角度

d : C 馬達提供的伸出量(mm)

l_1 : 簡化模型中{W}原點到運動中心(紅點)的距離

Inverse kinematic:

藉由幾何法求出其逆向運動學，假設 $P^W = (x, y, z)$ 為機器人針尖的目標

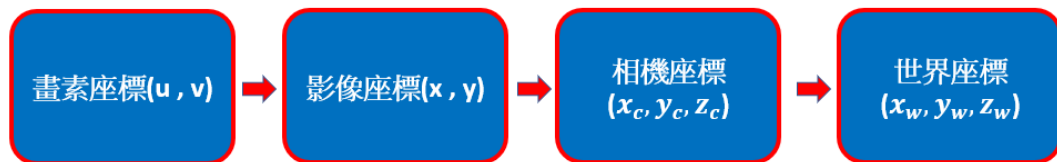


則

$$\theta_1 = \tan^{-1} \frac{y}{x}$$
$$\theta_2 = \tan^{-1} \left(\frac{z - l_1}{\sqrt{x^2 + y^2}} \right)$$
$$d = \sqrt{x^2 + y^2 + (z - l_1)^2}$$

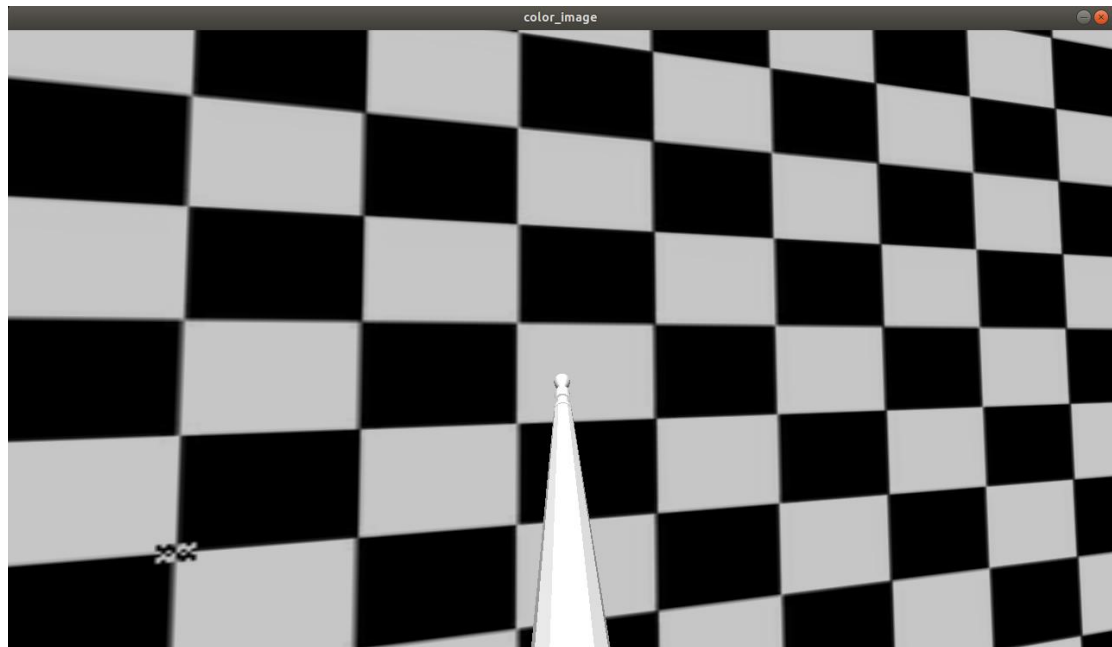
相機座標轉換

不考慮 distortion 的情況下將相機畫面經由以下四個步驟轉換轉成機器人的世界坐標系



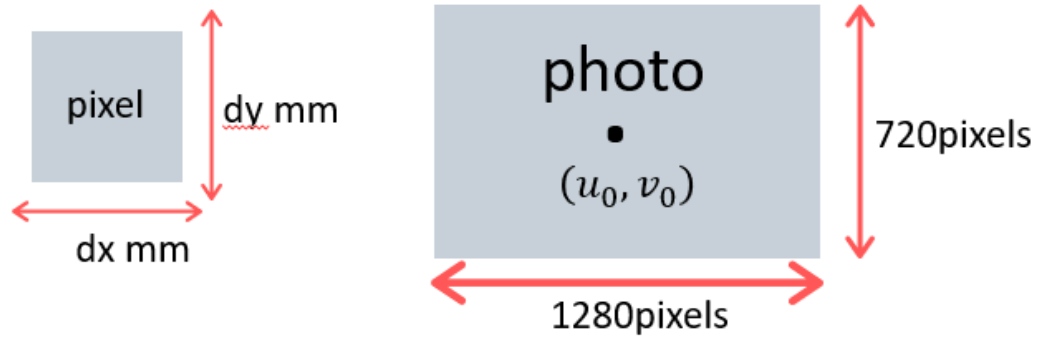
畫素座標

一開始點擊相機畫面即可求出畫素座標(u, v)



影像座標

假設一張照片的 pixel 長寬分別為 dx , dy (如下圖)且令照片原點 (u_0, v_0) 在照片中央,



u, v 可以以下面二式表達 $u = \frac{x}{dx} + u_0$, $v = \frac{y}{dy} + v_0$, 並寫成矩陣的形式

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{dx} & 0 & u_0 \\ 0 & \frac{1}{dy} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad \text{求其逆矩陣即可求得影像座標}(x, y)$$

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{dx} & 0 & u_0 \\ 0 & \frac{1}{dy} & v_0 \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

此次照片畫素為 $(1280*720)$, 因此

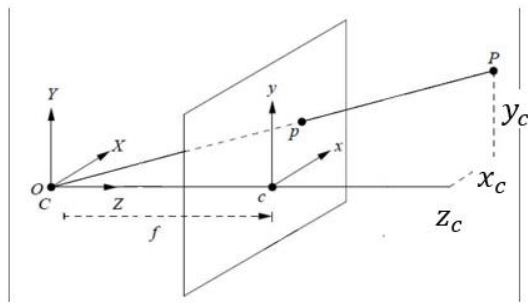
$$(u_0, v_0) = (640, 360)$$

$$\frac{f}{dx}, \frac{f}{dy} = 277(\text{default})$$

f =焦距

相機座標

求出 (x, y) 後, 利用相似三角形原理如下圖可求出下式



$$\frac{f}{z_c} = \frac{x}{x_c} = \frac{y}{y_c}$$

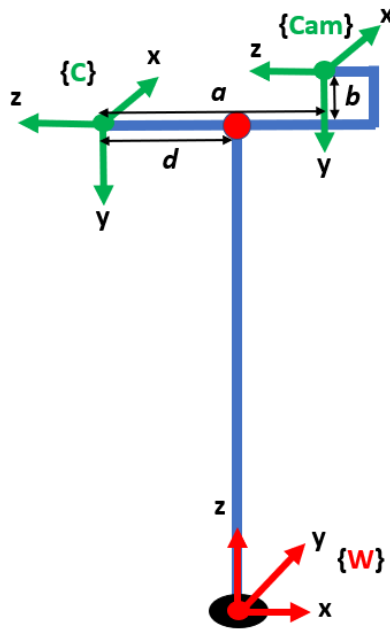
x, y 可以以下面二式表達 $x = \frac{fx_c}{z_c}$, $y = \frac{fy_c}{z_c}$, 並寫成矩陣的形式

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \frac{1}{z_c} \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} \text{ 求其逆矩陣即可求得相機座標 } (x_c, y_c, z_c)$$

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = z_c \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

此次模擬使用的是深度相機，因此 z_c 可直接求出

世界座標



之前利用機器人學已求出機器人的順向運動學 ${}^w_c T$ ，現在只要求出轉換矩陣 ${}^{cam}_c T$ 可利用下式求出世界座標

$$\begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} = {}^w_c T {}^{cam}_c T \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix}$$
$${}^{cam}_c T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -b \\ 0 & 0 & 1 & -a \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

控制系統(未完成)

Gazebo 具備物理引擎可以用來模擬真實世界的物理運動，因此可以撰寫控制器使其符合預計的規格

機器學習(未完成)

藉由正逆向運動學可以求出世界座標對馬達轉角的關西，將世界座標 (x, y, z) 作為輸入，馬達轉角 (θ_1, θ_2, d) 做輸出，訓練一神經網路，用來與逆向運動學的結果做為比較

介面設計(未完成)

預計利用 pyqt 撰寫一符合使用者需求的介面