

# 第五組 - 實驗四

## 二連桿寫字機器人

教師：黃漢邦、林沛群 教授

R09522849 陳品存

R10522801 吳政彥

R10522814 陳政豪

R10522834 戴承寧



# 機構

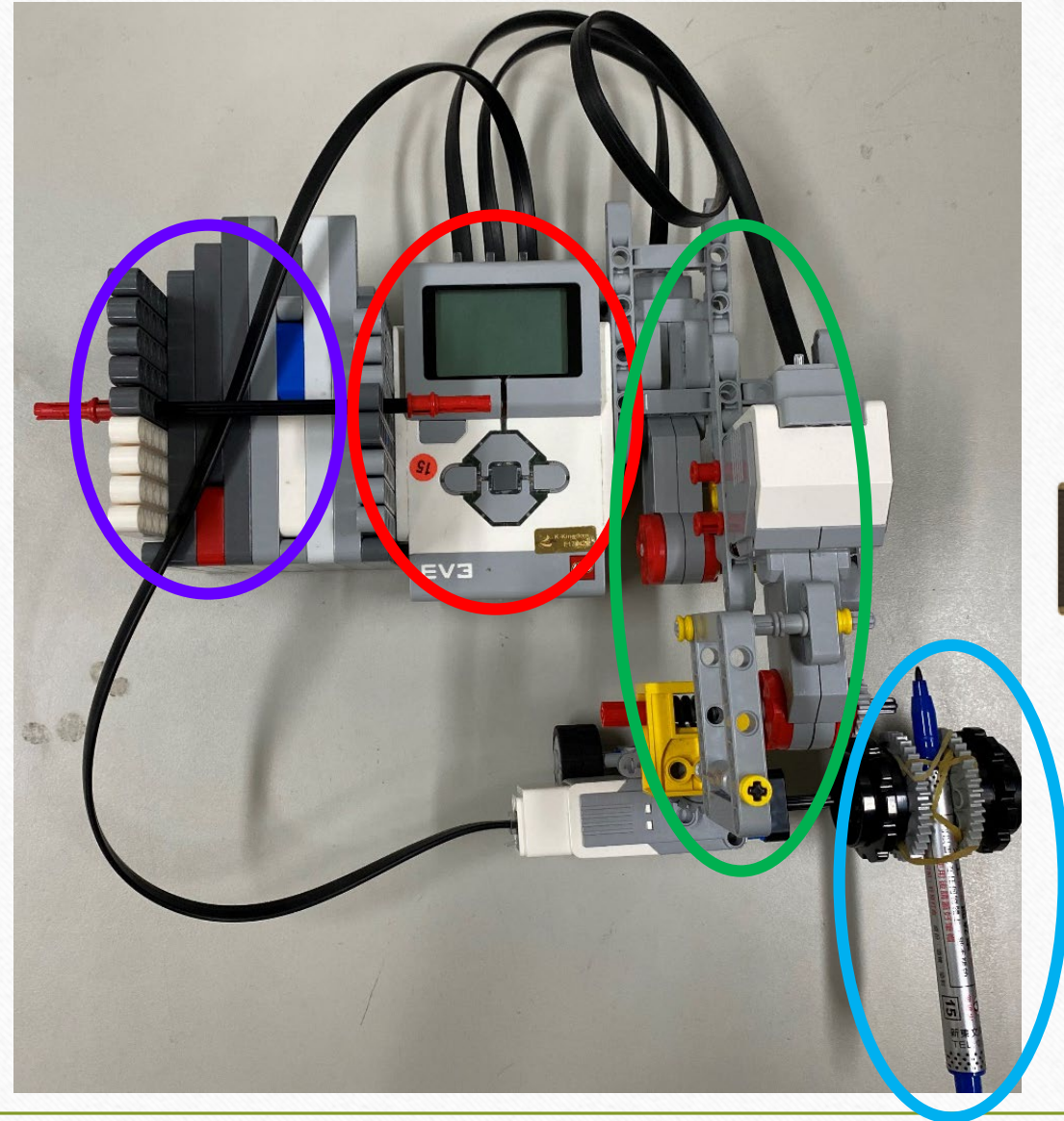


# 機構 - 上視圖

從上視圖可以看出我們機構。紫色框的部份是我們的配重，主要是因為我們二連桿以及夾筆機構類似於懸臂樑，所以主機會微微翹起，於是我們就加一個配重的機構。

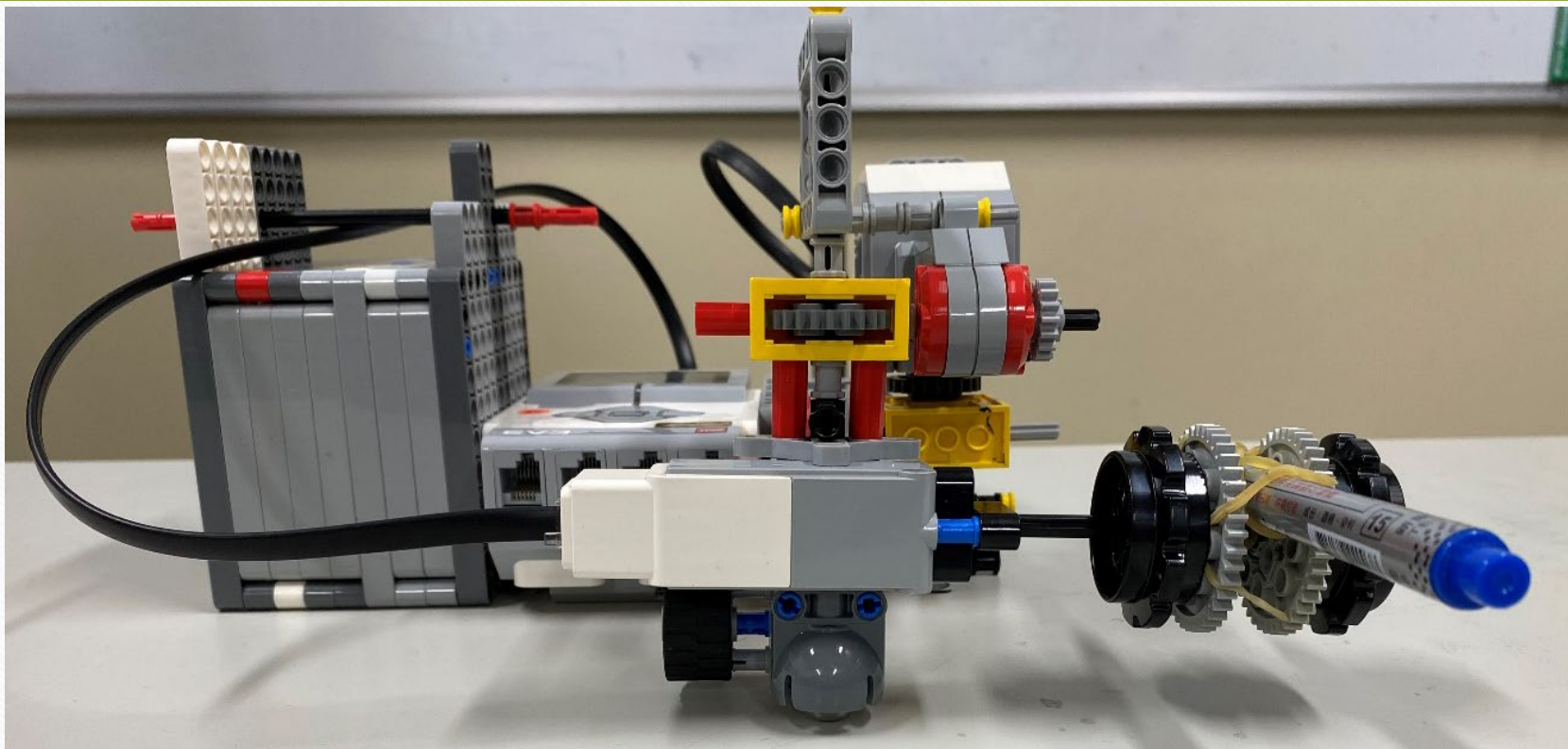
紅色框是EV3主機。

綠色框是二連桿機構，藍色框是夾筆機構，後面會對這兩個機構作詳細說明。



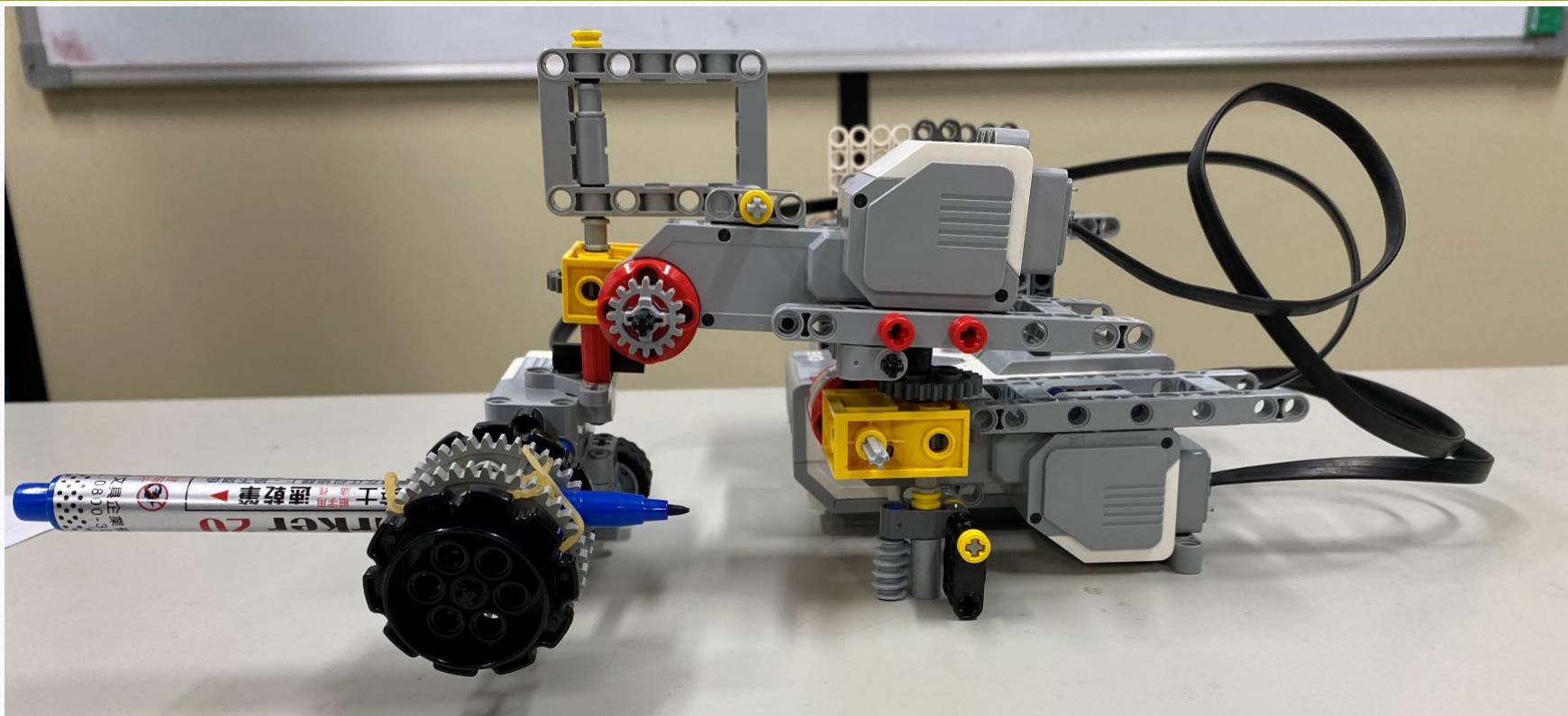


## 機構 - 側視圖 (1/2)





## 機構 - 側視圖 (2/2)

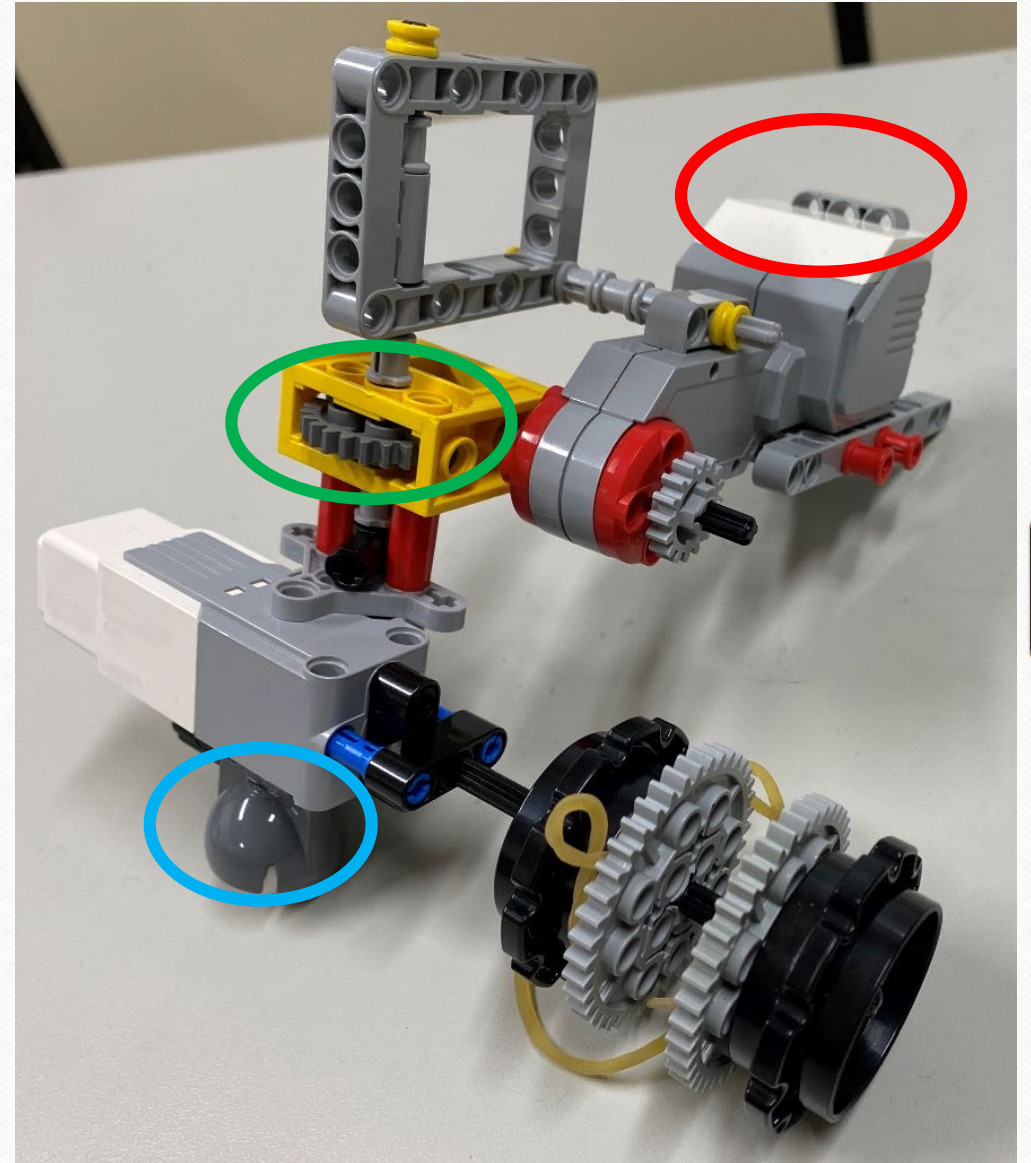




# 機構 - 二連桿

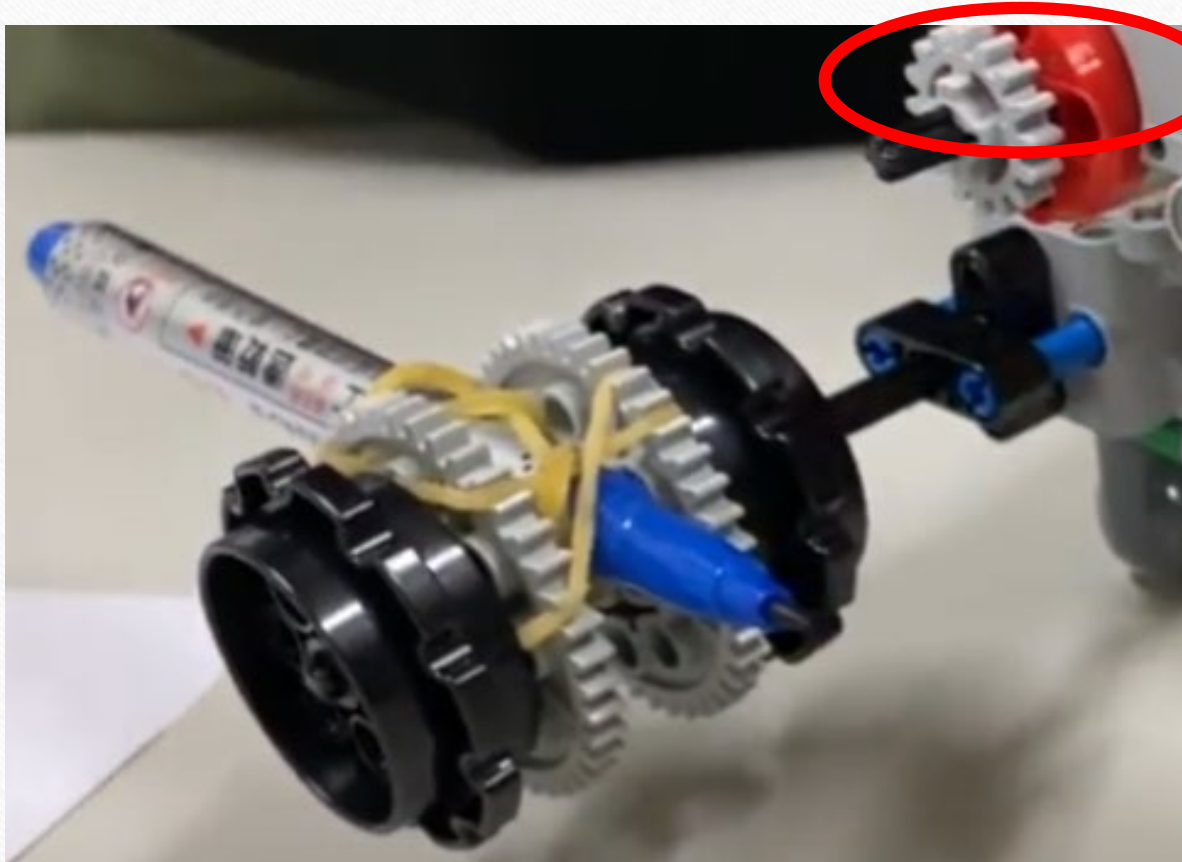
首先，在平面上若有兩個Revolute Joint，在沒有singular的狀況下，即可到達workspace上的任一點，我們二連桿機構的第一個R joint是紅色框的部分，與主機是直接相連。第二個R joint是綠色框的部分。

再來因為二連桿機構往往都是從主機作延伸出去的機構，所以會形成懸臂樑的狀況，懸臂樑是個不穩定的機構，因為尾端會因為重力而有一個上下方向的偏移量，因此我們在二連桿最後一個R joint下加上一個萬象輪（藍色框），作一個支撐。





# 機構-夾筆機構



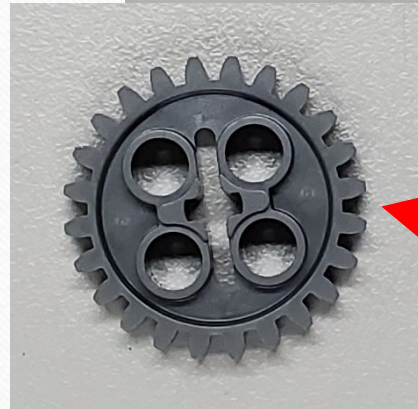
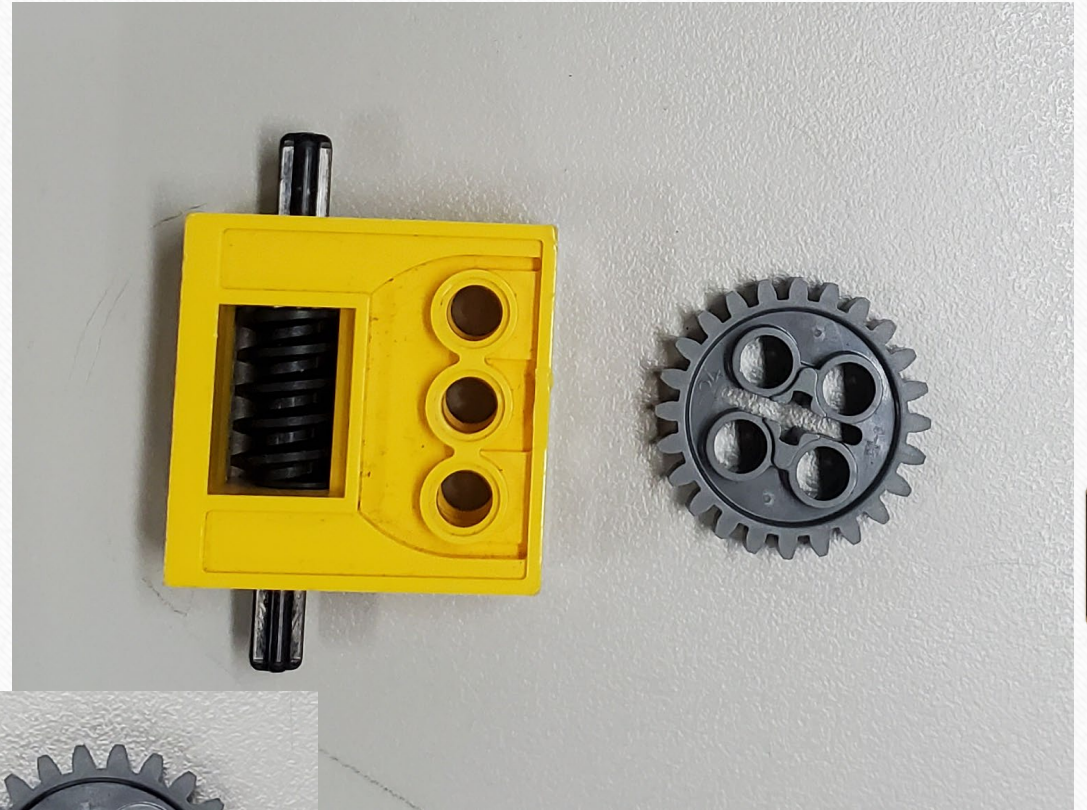
因為我們在移動連桿的時候，不希望筆是垂下來跟紙有接觸（會有筆跡），因此在移動的時候我們讓筆是平行與紙面的。當到達想要寫字的軌跡時，我們在利用馬達（紅色框）轉動夾筆機構使筆可以照著軌跡寫字，但有一個小問題是，因為我們是讓夾筆機構轉，讓筆跟紙有接觸，所以在一開始的軌跡之前會有多餘的筆跡。



# 機構 - 減速機 (1/2)

由於解完IK後，會有許多小角度的問題，但Lego硬體上無法做到小角度（單次轉動角度不到一度）的運動，因此需要使用減速機構使其轉動角度減小。

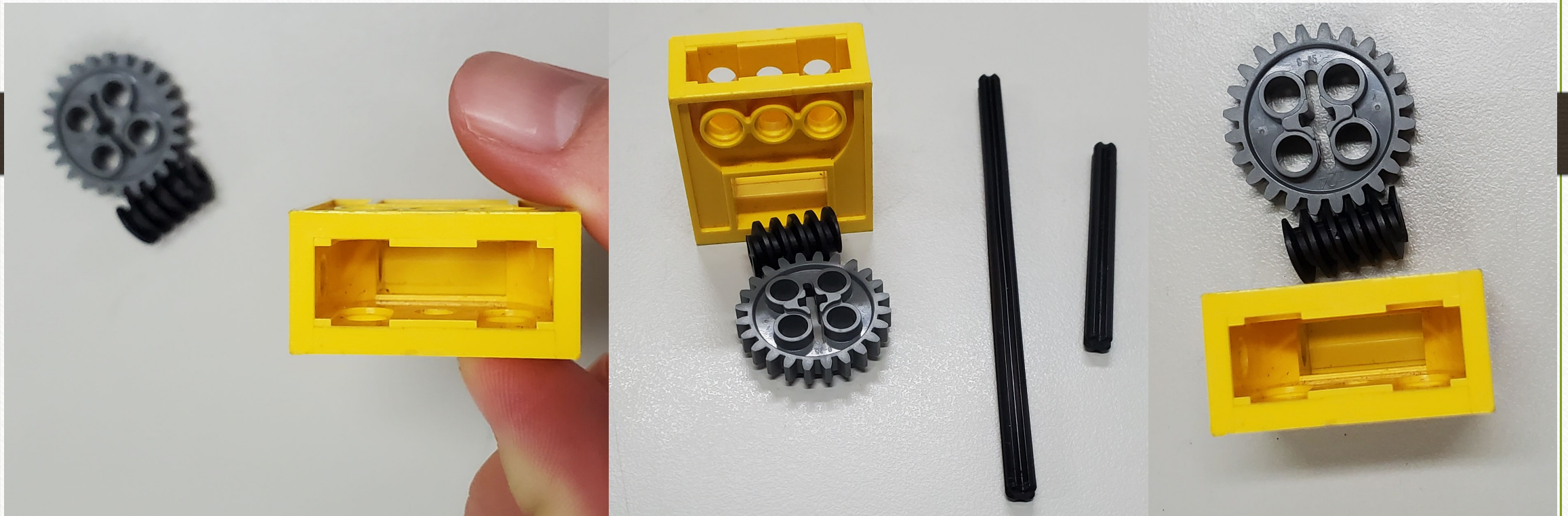
我們採用Lego配件－ $1/24$ 減速機，在理想狀態（齒輪完全密合）下，當馬達旋轉一度時，其輸出可以達到驚人的 $1/24 = 0.00416$ 度。



24齒。



## 機構 - 減速機 (2/2)



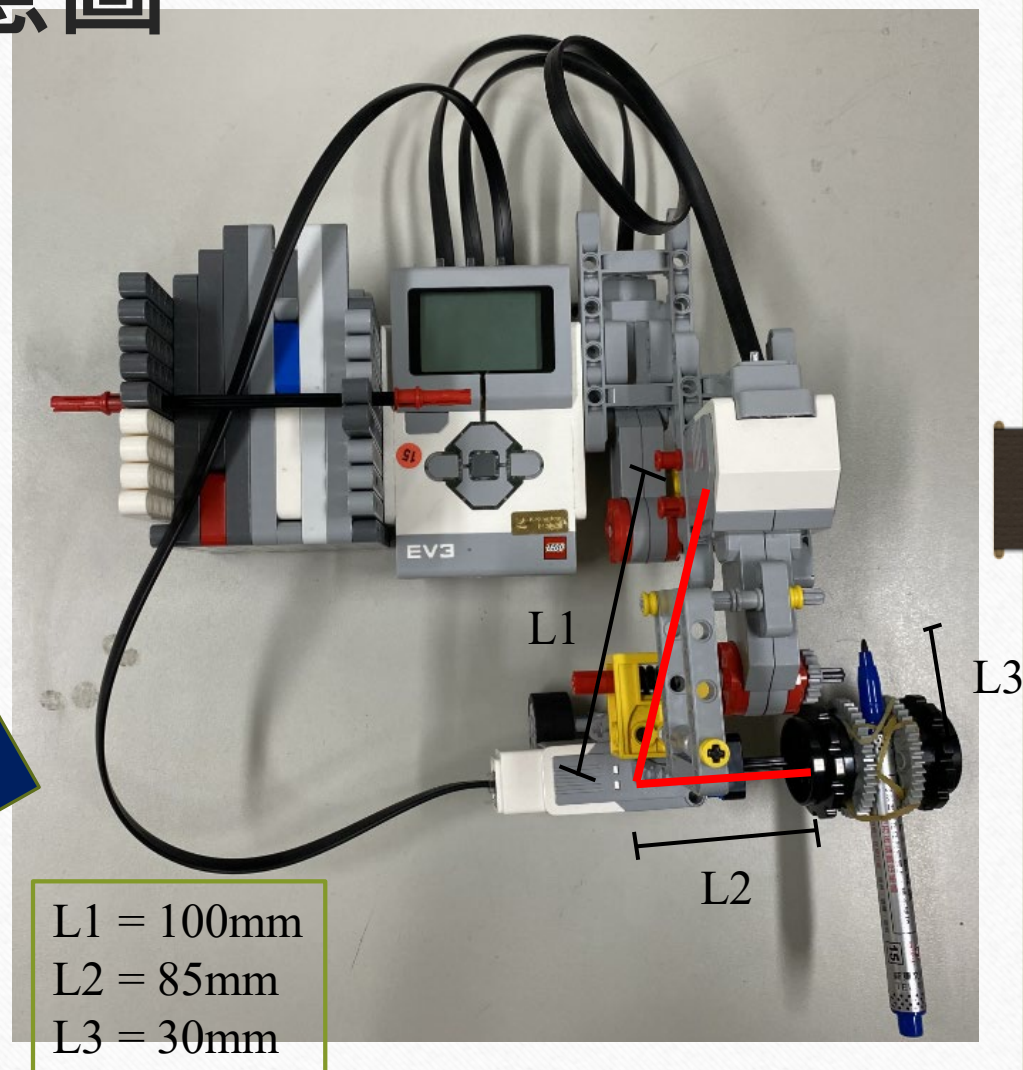
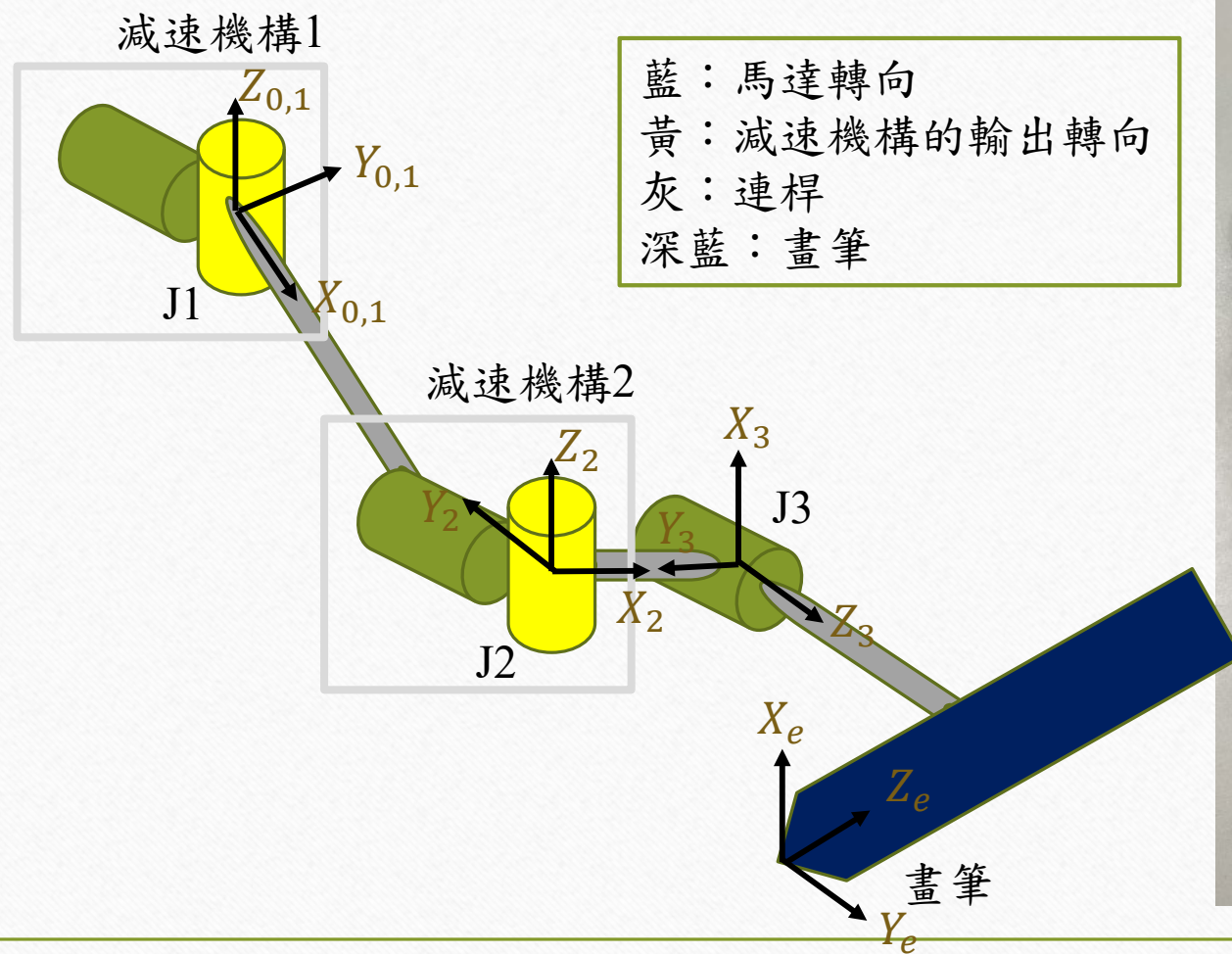


# 詳述 DH Table

---



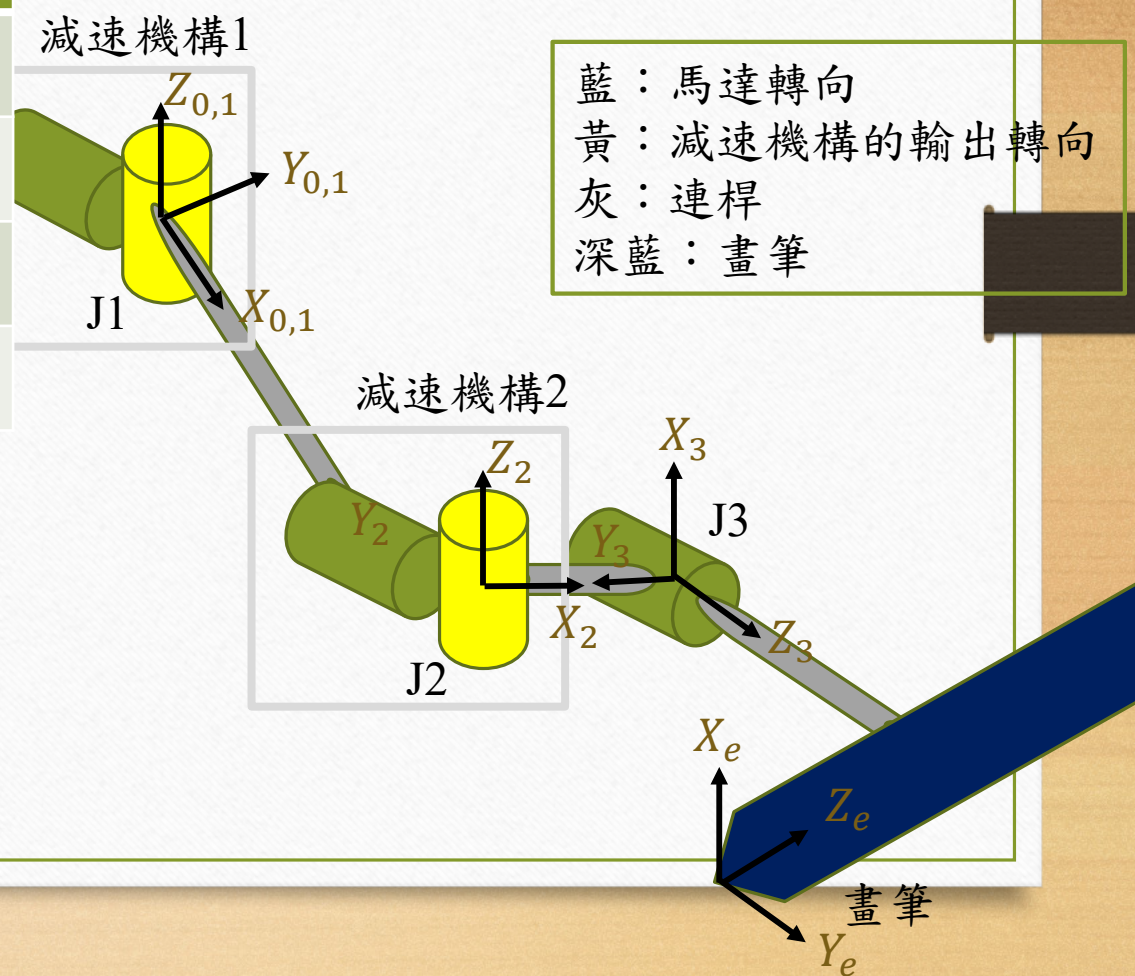
# 機構示意圖





# DH 表

Joint $i$	$\alpha_{i-1}$ (deg)	$\theta_i$ (deg)	$a_{i-1}$ (mm)	$d_i$ (mm)
1	0	$\theta_1$	0	0
2	0	$\theta_2$	100	0
3	90	$90 + \theta_3$	10	0
4	90	0	85	0



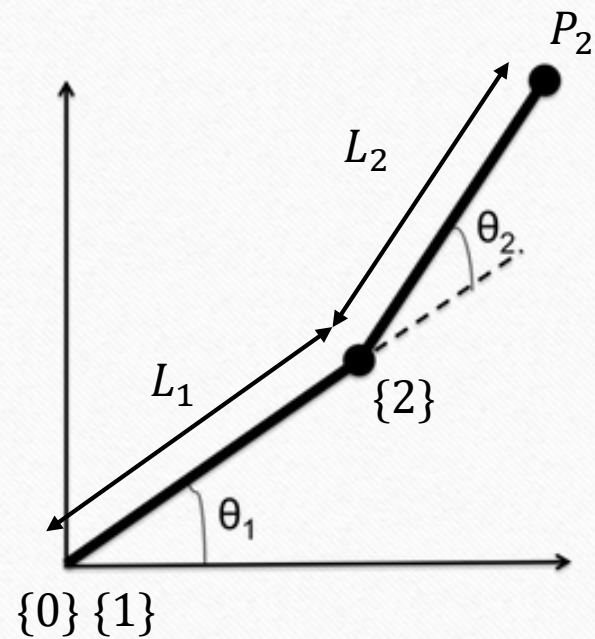


# 程式解説

---



# DH Table of Two Link Manipulator



Craig method

Link#	$\alpha_{i-1}$	$a_{i-1}$	$d_i$	$\theta_i$
1	0	0	0	$\theta_1$
2	0	$L_1$	0	$\theta_2$

$$T_2^0 = \begin{bmatrix} \cos(\theta_1 + \theta_2) & -\sin(\theta_1 + \theta_2) & 0 & L_1 \cos(\theta_1) \\ \sin(\theta_1 + \theta_2) & \cos(\theta_1 + \theta_2) & 0 & L_1 \sin(\theta_1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad P_2^2 = \begin{bmatrix} L_2 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$



# Forward Kinematics of Two Link Manipulator

---

$${}^0P_2 = T_2^0 P_2^2 = \begin{bmatrix} L_1 \cos(\theta_1) + L_2 \cos(\theta_1 + \theta_2) \\ L_1 \sin(\theta_1) + L_2 \sin(\theta_1 + \theta_2) \\ 0 \\ 1 \end{bmatrix}$$

```
def FK(self, theta1, theta2):  
    J1_loc = self.J0_loc + [self.L1 * np.cos(theta1), self.L1 * np.sin(theta1)]  
    J2_loc = J1_loc + [self.L2 * np.cos(theta1 + theta2), self.L2 * np.sin(theta1 + theta2)]  
    return J1_loc, J2_loc
```

給兩軸的旋轉角度，FK函式會返回轉軸1與轉軸2末端的XY座標



# Inverse Kinematics of Two Link Manipulator

---

$$\theta_2 = \cos^{-1}\left(\frac{x^2 + y^2 - L_1^2 - L_2^2}{2L_1L_2}\right)$$

$$\theta_1 = \tan^{-1}\left(\frac{y}{x}\right) - \tan^{-1}\left(\frac{L_2 \sin(\theta_2)}{L_1 + L_2 \cos(\theta_2)}\right)$$

```
def IK(self, x, y):  
    theta2 = np.arccos((x ** 2 + y ** 2 - self.L1 ** 2 - self.L2 ** 2) / (2 * self.L1 * self.L2))  
    theta1 = np.arctan2(y, x) - np.arctan2(self.L2 * np.sin(theta2), self.L2 * np.cos(theta2) + self.L1)  
    return self.dr * np.degrees(theta1), self.dr * np.degrees(theta2)
```

給末端的XY座標，IK函式會返回轉軸1與轉軸2的旋轉角度



# Synchronize Two Motors

```
class MotorControl(Thread):
    def __init__(self, port, rot_dir):
        Thread.__init__(self)
        self.m = LargeMotor(port)
        self.m.polarity = rot_dir

    def run(self, speed, theta):
        self.m.on_for_degrees(speed, theta)

    def motorInfo(self):
        print(self.m.state)

    def reset(self):
        self.m.reset()

    def stop(self):
        self.m.stop()
```

馬達Class

```
def move(self, theta1, theta2):
    t12 = abs(theta1) + abs(theta2)
    maxspeed = 30
    if theta1 != 0 or theta2 != 0:
        p1 = maxspeed * abs(theta1) / t12
        p2 = maxspeed * abs(theta2) / t12
        t1 = Thread(target=m1.run, args=(p1, theta1,))
        t2 = Thread(target=m2.run, args=(p2, theta2,))
        t1.start()
        t2.start()
        t1.join()
        t2.join()
```

使用兩個thread

為了讓兩個馬達能同動，因此使用兩條執行緒來執行，並定義最大速度，接著依照馬達旋轉角度的大小來配速

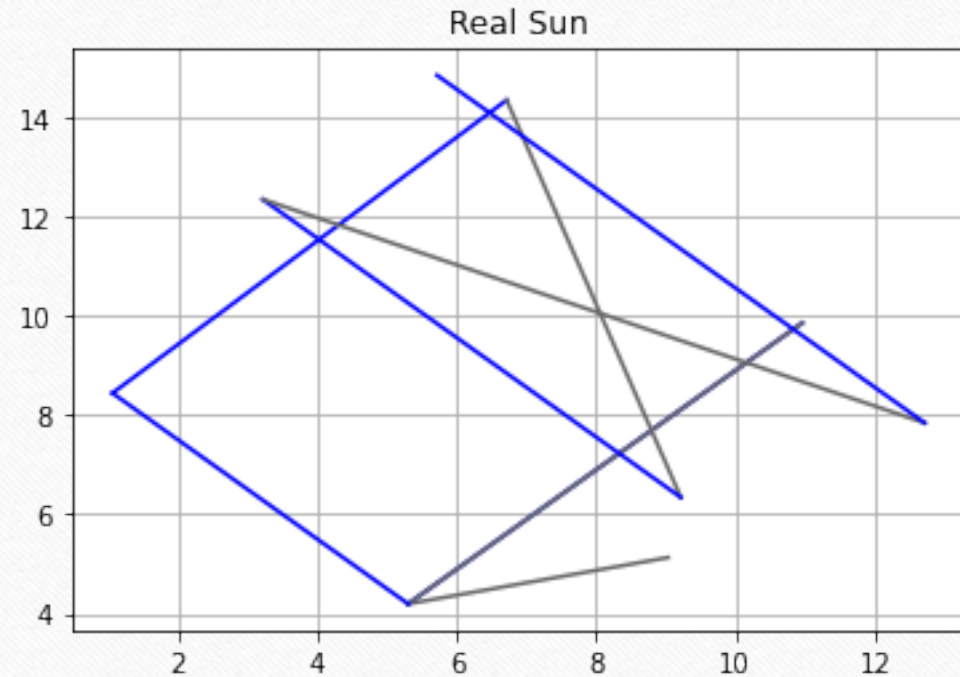
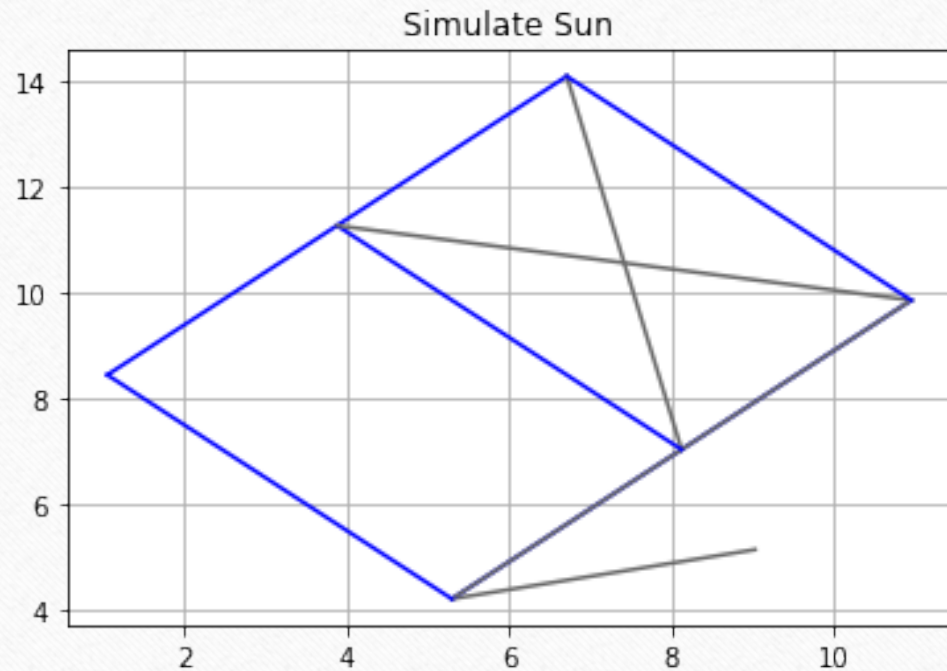


# Trajectory Planner

- 透過程式先將「日」、「春」的軌跡先規劃出來，每兩點之間會用線性內插的方式補10個點，接著將每個點的XY座標來解二連桿的逆運動學，但照著模擬的軌跡無法將字寫完整，因為實際會有馬達誤差累積、筆與紙的摩擦力等因素，因此有些部分會故意將筆跡寫超出預期的目標點來補足誤差。
- 由於將字正著寫會有一邊轉軸角度大另一轉軸角度小的問題，因此將字斜著寫。

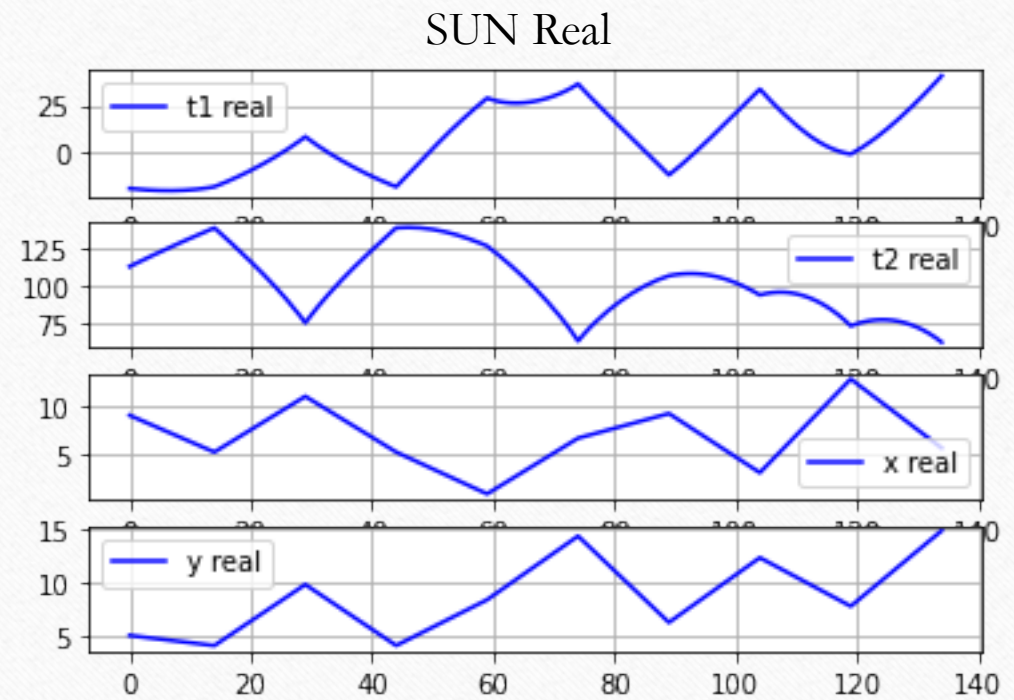
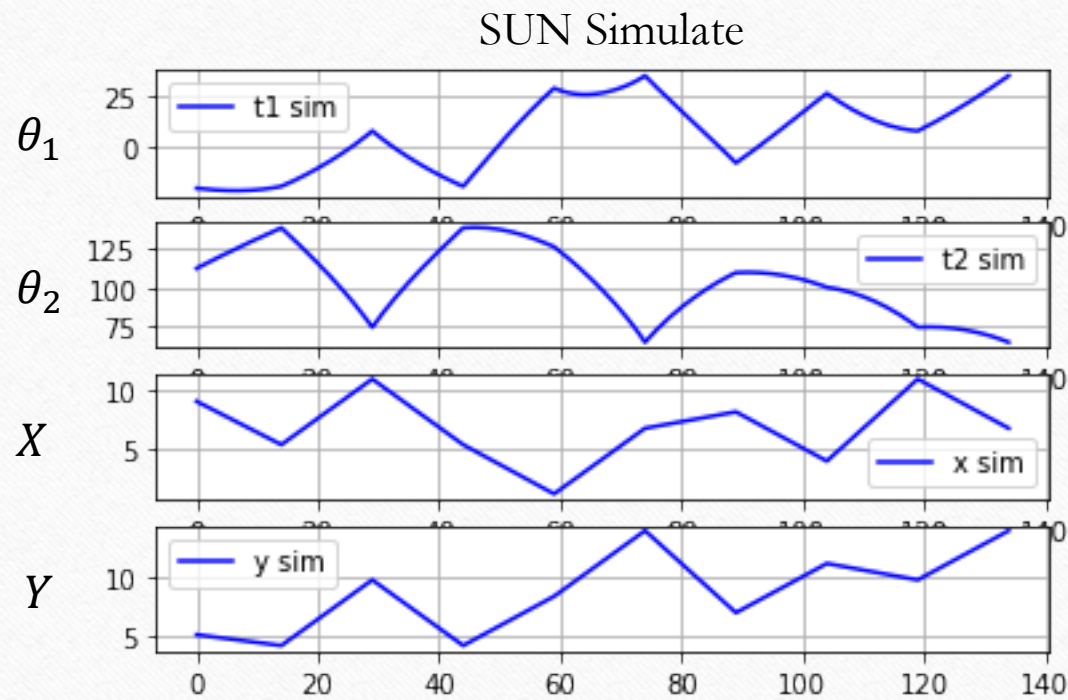


# Simulate and Real Trajectory of **SUN**





# Simulate and Real Trajectory of **SUN**

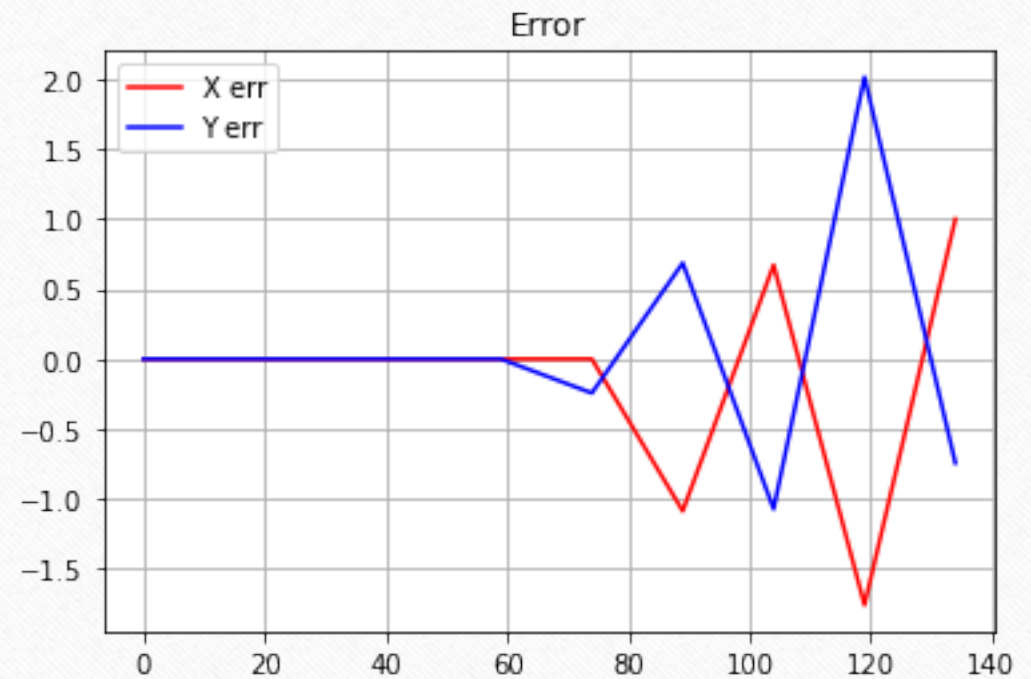




# SUN Trajectory Mean Absolute Error



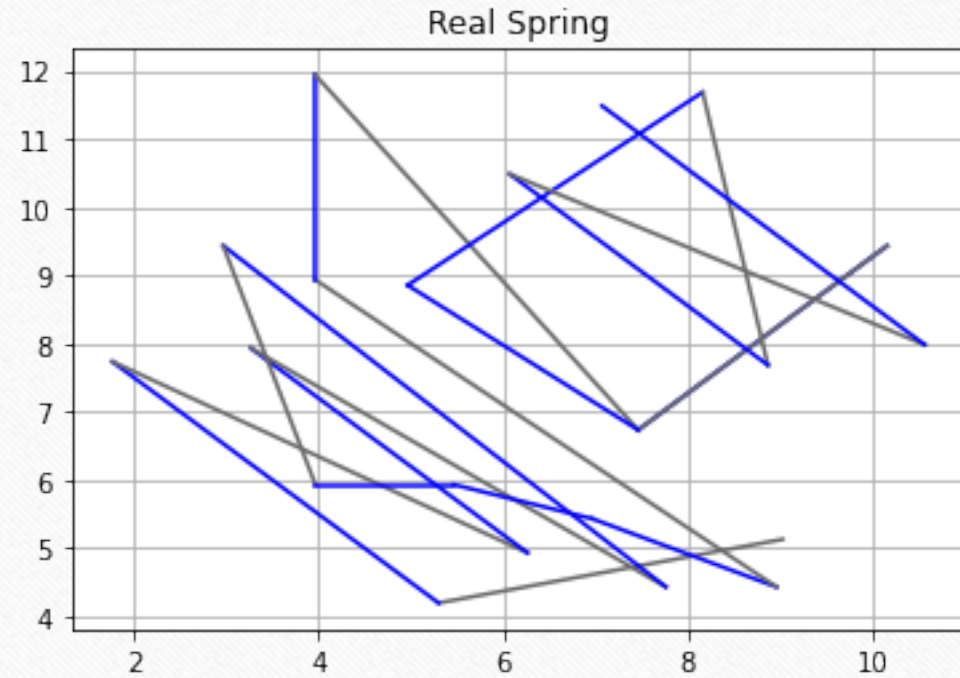
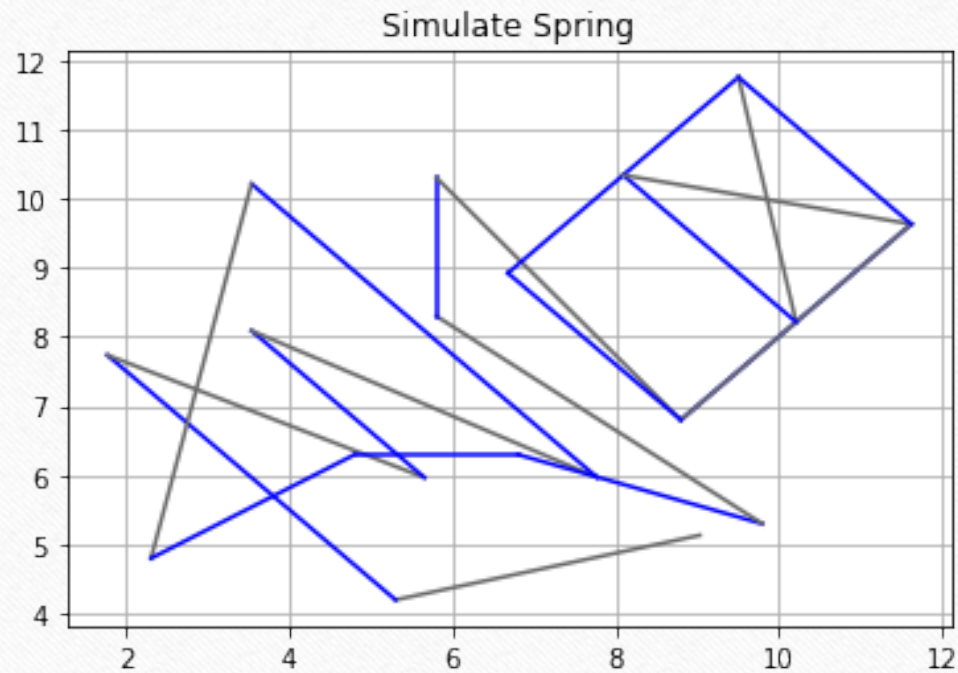
Theta1 err = 0.68682      Theta2 err = 0.71271



X err = 0.18203      Y err = 0.10988

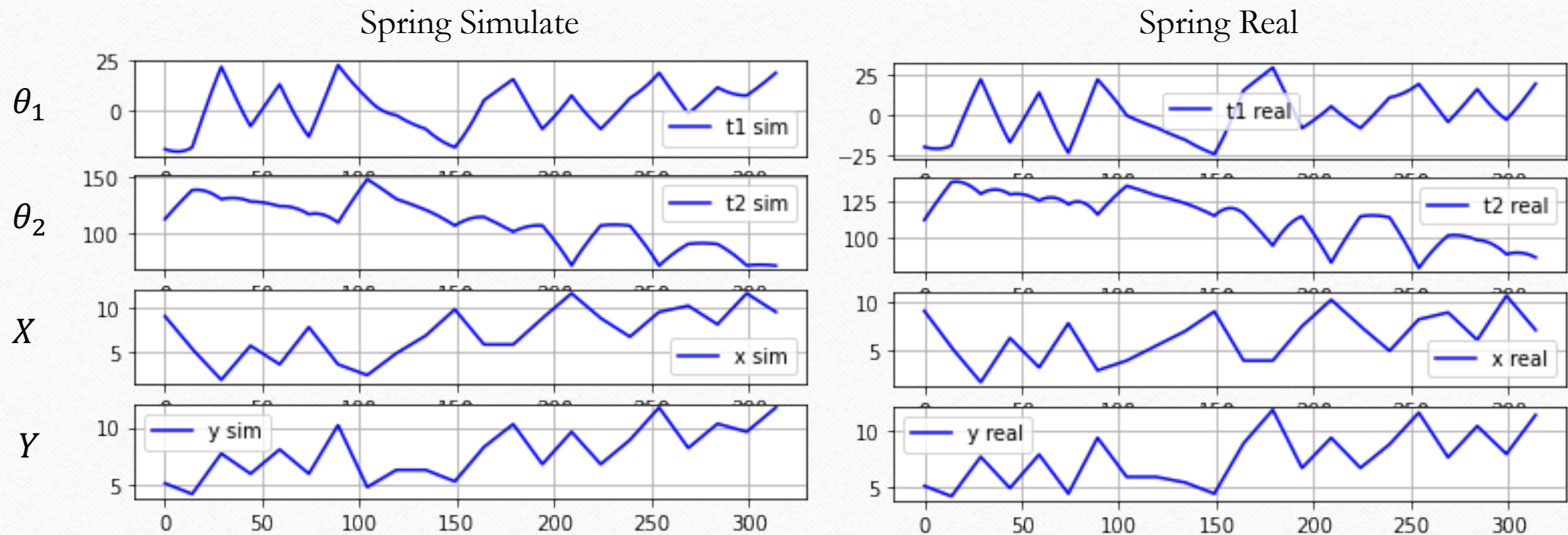


# Simulate and Real Trajectory of **SPRING**





# Simulate and Real Trajectory of **SPRING**





# SPRING Trajectory Mean Absolute Error



Theta1 err = 0.90538      Theta2 err = 5.22962



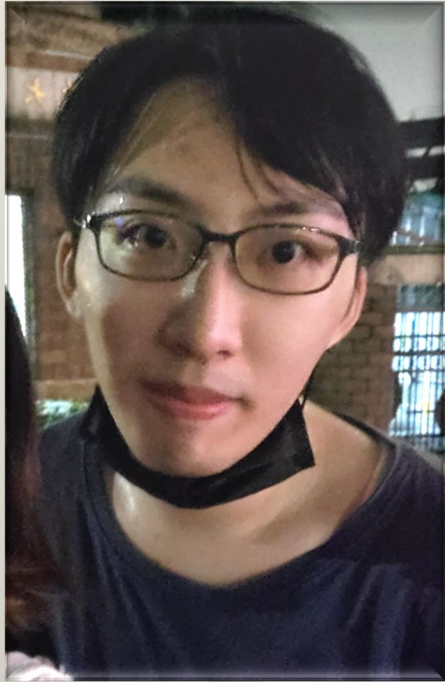
X err = 0.72188      Y err = 0.22829



# 組員心得

---





陳品存 (R09522849) 心得：

此次實驗大概是本學期所有實驗以來，機構改變最多的一次，從一開始我們如同市售印表機的X-Y平面型設計，再到二連桿機構卻發現因為沒有加上減速機而導致解完IK後依舊無法寫出漂亮的字，直到最後設計出二連桿搭配減速機構才算完成此次實驗的基本功能。其中減速機構的設計也是一大學問，我們試了三種以上的方法，後來才在Lego官方找到適合的減速機構。也因此，我們前期花了很大的時間在機構上的設計與改良，所以程式的部份也一再修改，負責的Teammates真的辛苦了！

吳政彥 (R10522801) 心得：

透過這次的實驗了解了軌跡規劃的重要性，一開始使用二軸XY卡氏座標機構，只需要用順運動學即可完成，且寫直線部分十分輕鬆，但之後助教說只能用二連桿機構，就必須計算二連桿的逆運動學，由於將兩個點透過線性內插的方式來產生十個點，但因此移動的角度過小，馬達精度無法達到，因此使用了24倍的減速機構。雖然這次實驗我們這組是直接透過XY座標來規劃軌跡，並沒有使用三次方程或者Parabolic Blends，因此在經過轉折點時會發現誤差較大，且機構前端夾筆部分會晃動，使得每次下筆都會多劃一筆，使得寫出來字多了幾分恐怖感，希望這次學到種種教訓能在期末專題的機械手臂上改善。







陳政豪 (R10522814) 心得：

如果沒有這次的實驗，那這堂課肯定是平淡無奇的，在做這個實驗時，我發現這個實驗不僅連結上課內容，也讓我們夥伴間的羈絆更加堅定不移，真是一堂好課，但我不想再修一次！

戴承寧 (R10522834) 心得：

這次的實驗也是命運多舛，一開始我們以為可以用xy平台實現，因此花了許多時間組裝類似印表機的xy平台，機構相較於二連桿更為複雜，但是不須解IK。二連桿實現上遇到最大的問題是小角度問題以及結構剛性，這兩者都是硬體本身的限制，從中學習到最多的就是如何調適不夠好的硬體(例如配重)來設法達到更好的軌跡表現。





# 小組分工表

---



## 小組分工表

姓名	負責項目與內容
陳品存	機構設計（減速機）與PPT統整
吳政彥	程式設計與軌跡規劃
陳政豪	機構設計（二連桿）
戴承寧	機構設計（夾筆）與DH-Table