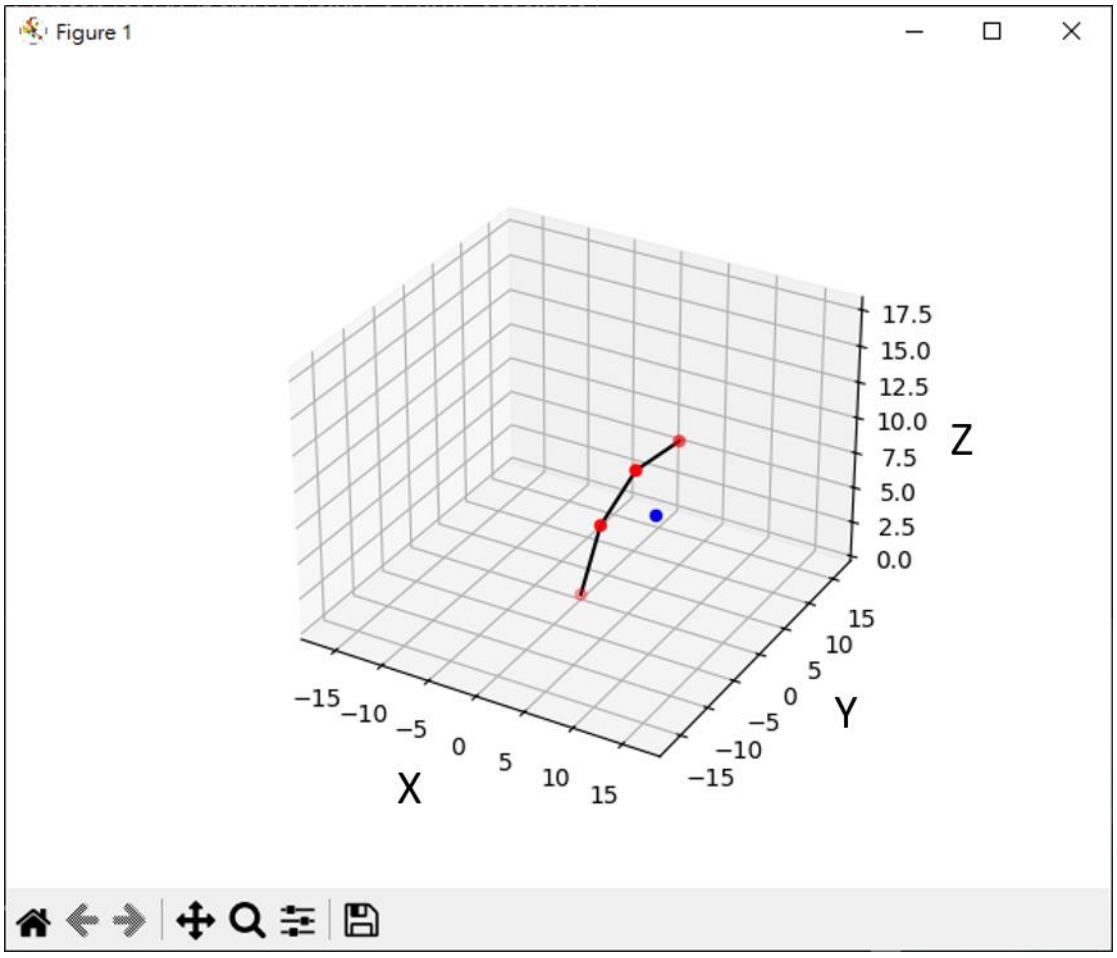
一、 順向運動學



計算公式:

1. 計算 x, z 分量

令力臂角度為$[\theta_0, \theta_1, \theta_2]$（z 軸為 0 度），長度為$[L_0, L_1, L_2]$，則頂點座標為：

$$x = L_0\, sin(\theta_0) + L_1\, sin(\theta_0 + \theta_1) + L_2\, sin(\theta_0 + \theta_1 + \theta_2)$$

$$z = L_0 \cos(\theta_0) + L_1 \cos(\theta_0 + \theta_1) + L_2 \cos(\theta_0 + \theta_1 + \theta_2)$$

2. 計算底盤旋轉，修正 x, y 分量

令底盤旋轉角度為$\theta 3$

$$x\_fixed = x\, sin(\theta_3)$$

$$\mathrm{y}\_fixed = y\, cos(\theta_3)$$

可得力臂頂點為(x_fixed, y_fixed, z)

程式實作：

```python
# 基礎運算 (x, y) = [L*sin(theta), L*cos(theta)]
def compVector(self, a, theta):
    radians = math.radians(theta)
    x = a*math.sin(radians)
    y = a*math.cos(radians)
    return x, y

# 計算x, z方向
def compCord(self):
    top = [0, 0]; theta=0
    cordList=[] # 將手臂的每個關節座標紀錄後，迭代運算出頂點
    for i in range(3):
        theta+=self.angList[i]
        x, z = self.compVector(self.lenList[i], theta)
        top[0] += x
        top[1] += z
        cordList.append([top[0], top[1]])
    return cordList

# 修正x, y
def compCord_y(self):
    cord = self.compCord()
    for i in range(3):
        x, z = cord[i]
        x, y = self.compVector(x, self.angList[3])
        cord[i] = [x, y, z]
    self.cord = cord
```

二、 取得環境觀察值(Observation)

令力臂角度為$[\theta_0,\theta_1,\theta_2,\theta_3]$，長度為$[L_0,L_1,L_2]$，目標點座標為$[x,y,z]$（$\theta 3$ 為底盤旋轉角度）。

由於θ與 x, y, z 在空間中的極限範圍，分別是[-90, 90]與[-L, +L]，故回傳標準化後的觀察值：

$$\left\{\frac{\theta_0}{90},\frac{\theta_1}{90},\frac{\theta_2}{90},\frac{\theta_3}{90},\frac{x}{L},\frac{y}{L},\frac{z}{L}\right\}$$

$$L = L_0 + L_1 + L_2$$

程式實作：

```python
def getObs_norm(self):
    angList = [item / 90 for item in self.angList] #Angle range(-90~90)
    target = [item / self.lim for item in self.target] #Target range (-15~15)
    return angList+target
```

三、 Q_net 與 PSO

令 Q_net 為(11, 64, 1)的 MLP
[11]->ReLu->[64]->[1]

令 PSO 矩陣輸出大小為 4，範圍[-1, 1]
swarmsize=10, max_iter=100

固$fitness(\Delta\theta_0,\Delta\theta_1,\Delta\theta_2,\Delta\theta_3)$為: $Q\_net(stack(Obs,\Delta\theta))$

$$stack(Obs,\Delta\theta)=\left\{\frac{\theta_0}{90},\frac{\theta_1}{90},\frac{\theta_2}{90},\frac{\theta_3}{90},\frac{x}{L},\frac{y}{L},\frac{z}{L},\Delta\theta_0,\Delta\theta_1,\Delta\theta_2,\Delta\theta_3\right\}$$

$$L=L_0+L_1+L_2$$

程式實作：

```
def fitness_function(input_data):
    input_data = np.hstack((observation, input_data))
    input_tensor = torch.tensor(input_data, dtype=torch.float32)
    output = Q_net(input_tensor)

    return -output.item()
```

四、 Reward function

1. 角度變化量(即為 PSO 輸出值)

$$R_{Ang}=1-(\Delta\theta_0,\Delta\theta_1,\Delta\theta_2,\Delta\theta_3)/4$$

2. 頂點與目標距離

$$R_{dis}=\frac{Dis}{L\times 2}$$

$$L=L_0+L_1+L_2$$

$$Dis=\sqrt{(target_x-top_x)^2+\left(target_y-top_y\right)^2+(target_z-top_z)^2}$$

3. 加權

$$Reward=0.1\left(R_{Ang}\right)+0.9(R_{dis})$$

五、 Loss & Optimizer

$$\text{Loss} = sgn\left[\frac{R-R_{prev}}{Q-Q_{prev}}\right]$$

R 為 Reward，Q 為 Action 的 Q_value

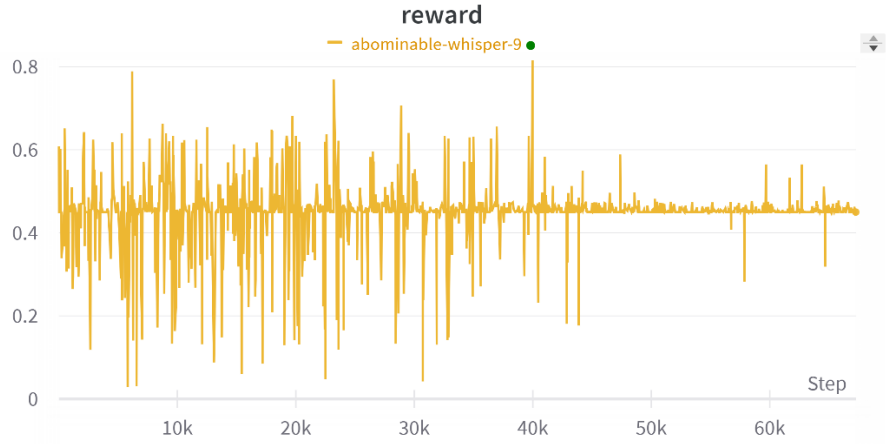
R_{prev}, Q_{prev} 為上一步的 Reward 與 Q 值。

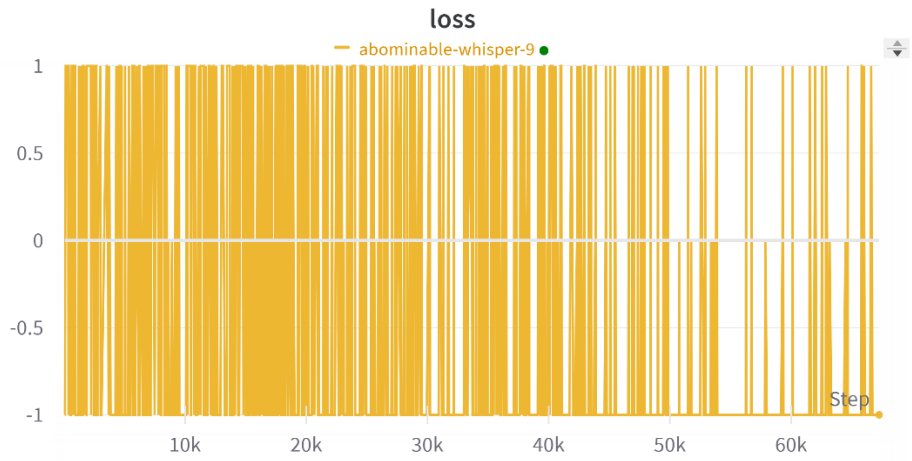
$$\Delta W = \eta * Loss * grad$$

六、 訓練流程(Action Received)

1. 取得觀察值
2. 使用 PSO 或 random 取得 action(epsilon greedy)
3. 環境更新(關節旋轉$\triangle\theta$*180)
4. 取得 Reward
5. 倒傳遞

七、 圖表





*10/25 開會

1. reward 加權
2. 3D 順向運動學(Env)
3. 角度極值(強制設定)
4. 公式寫成文件