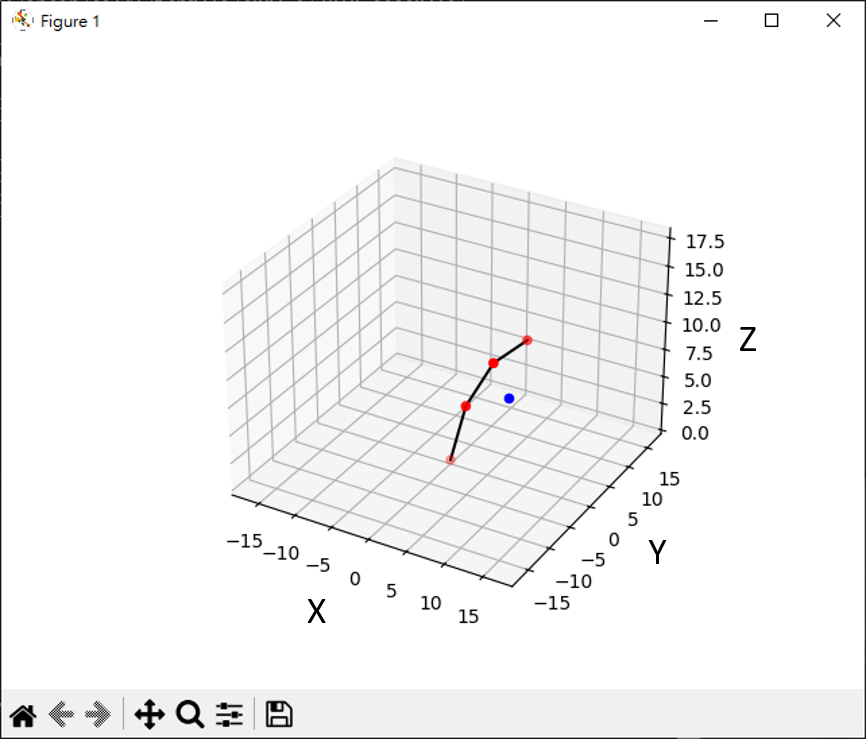
1. **順向運動學**



計算公式:

1. 計算x, z分量

令力臂角度為 (z軸為0度)，長度為，則頂點座標為：

1. 計算底盤旋轉，修正x, y分量

令底盤旋轉角度為θ3

可得力臂頂點為(x\_fixed, y\_fixed, z)

程式實作：

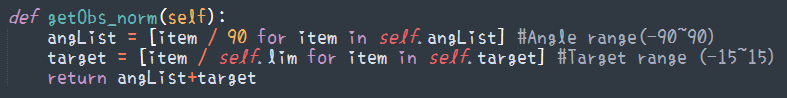


1. **取得環境觀察值(Observation)**

令力臂角度為，長度為，目標點座標為 (θ3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：

程式實作：



1. **Q\_net與PSO**

令Q\_net為(11, 64, 1)的MLP

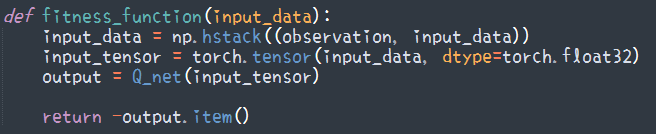
[11]->ReLu->[64]->[1]

令PSO矩陣輸出大小為4，範圍[-1, 1]

swarmsize=10, max\_iter=100

固為:

程式實作：



1. **Reward function**
   1. 角度變化量(即為PSO輸出值)

* 1. 頂點與目標距離
  2. 加權

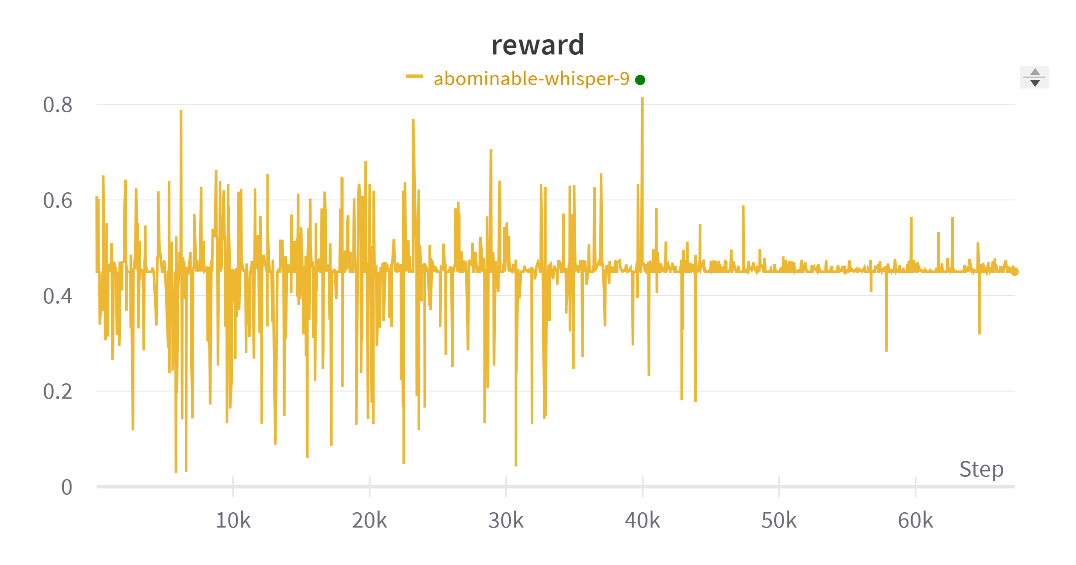
1. **Loss & Optimizer**

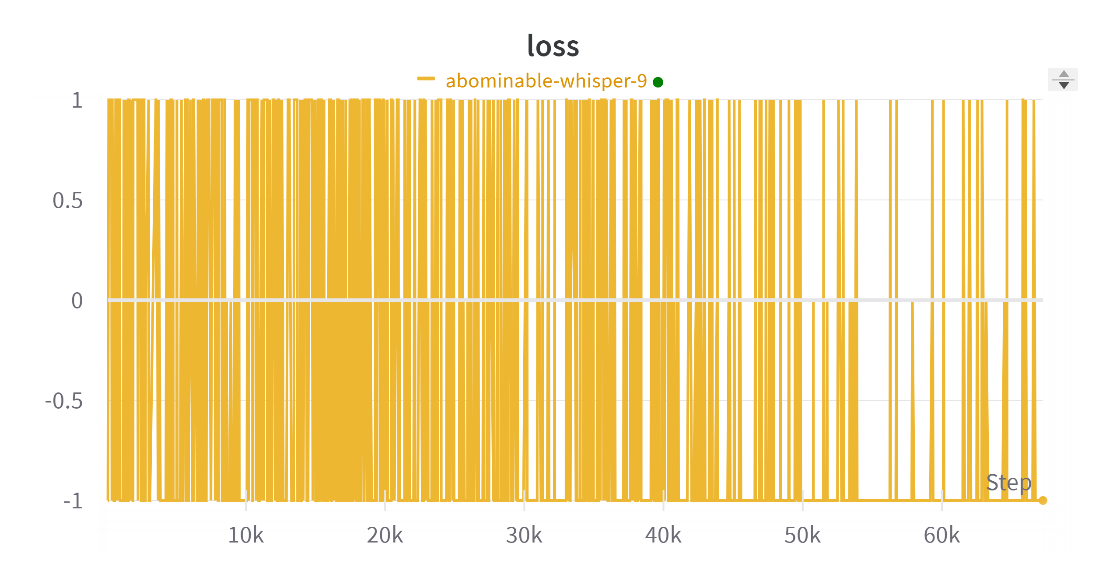
Loss =

R為Reward，Q為Action的Q\_value

Rprev, Qprev 為上一步的Reward與Q值。

1. **訓練流程(Action Received)**
2. 取得觀察值
3. 使用PSO或random取得action(epsilon greedy)
4. 環境更新(關節旋轉△θ\*180)
5. 取得Reward
6. 倒傳遞
7. **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