

Actions

It had 5 motors and a connector at the end to take action

```
Box(low=np.array([-1.0, -1.0,-1.0,-1.0,-1.0,0]), high=np.array([1.0, 1.0, 1.0, 1.0, 1.0]), dtype=np.intc)
```

No.	action	description
0	m0	Increase or decrease or do nothing on base motor(m0) angle by 1 degree ie 0.01745329252 rad To control motor angle.limit controlled by rewards
1	m1	Increase or decrease or do nothing on m1 motor angle by 1 degree ie 0.01745329252 rad
2	m2	Increase or decrease or do nothing on m2 motor angle by 1 degree ie 0.01745329252 rad
3	m3	Increase or decrease or do nothing on m3 motor angle by 1 degree ie 0.01745329252 rad
4	m4	Increase or decrease or do nothing on m4 motor angle by 1 degree ie 0.01745329252 rad
5	c	Lock or unlock connector

For motors

+1 to increase position by 1 degree

0 to do no change

-1 to decrease position

For connector

+1 to turn on connection (if load is in range it will auto connect)

0 to turn of connector (doesn't connect if load is in range)

1 for lock , 0 for unlock

Observations or states

```
Box(low=np.array([0, 0, 0, 0, 0, -0.7, -0.7, -0.7, 0, 0, -1, -1, -1, -1, -1, -1]),  
high=np.array([6.2831853072, 2.0943951024, 2.0943951024,  
2.0943951024, 2.0943951024, 0.7, 0.7, 0.7, 1, 1, 1, 1, 1, 1, 1, 1]), dtype=np.float32)
```

No	Observation	Description
0.	m0	Motor m0 position (Base motor)
1	m1	Motor m1 position
2	m2	Motor m2 position
3	m3	Motor m3 position
4	m4	Motor m4 position
5	Cpx	Connector's current positionX
6	Cpy	Connector's current positionY
7	Cpz	Connector's current positionZ
8	c	Connector state, 1 for lock , 0 for unlock
9	islinkable	Load connector presence
10	Ppx	Load pick up positionX
11	Ppy	Load pick up positionX
12	Ppz	Load pick up positionX
13	Dpx	Load Drop positionX
14	Dpy	Load Drop positionY
15	Dpz	Load Drop positionZ

