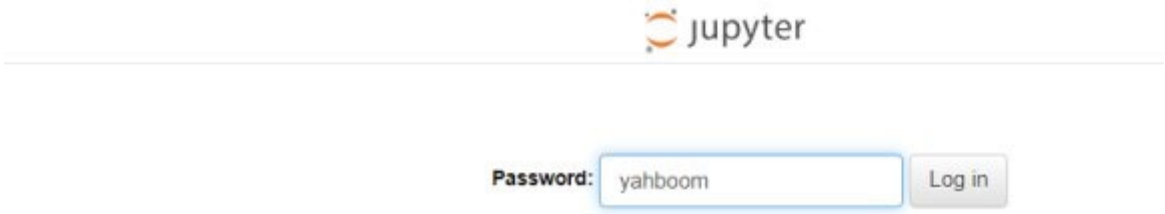
Robotic arm control

1. Purpose

Learning to control the robot arm of a robot dog

2. Experimental path source code

Enter the robot dog system, end the robot dog program, enter "ip (ip is the robot dog's ip): 8888" in the browser, and enter the password "yahboom"



Then log in and enter the path of **cd ~/DOGZILLA_Lite_class/3.Dog Base Control/08.Puppy robotic arm control and run robotic_arm_control.ipynb** .

3. Experimental Phenomenon

After running the source code, you can make an assignment according to the control you want to control the robotic arm on the robot dog.

4. Main source code analysis

1. motor(motor_id, data): API function for positive calculation, which inputs the id number (51\52\53) of the robot arm servo and the corresponding servo angle to control the rotation of the servo

Parameter name	scope	illustrate
motor_id	[51,52,53]	51, 52, 53 are the gripper, small arm, and large arm servos respectively
data	51: [-65, 65] 52: [-115, 70] 53: [-85, 100]	This parameter represents the angular position of the servo, in degrees
It is recommended to use the claw interface to control the gripper.		

2. claw (pos): function to control the claw individually pos: opening and closing position control of the claw 0 corresponds to fully open, 255 corresponds to fully closed 127: middle of the

claw, initial position

3. arm(arm_x, arm_z) Set the end position of the robot arm (inverse solution)

Parameter name	scope	illustrate
arm_x	[-80, 155]	Unit: mm
arm_z	[-95, 155]	Unit: mm

Note: x and z here are coordinates relative to the base of the robot, in millimeters. When setting a value that exceeds the robot's workspace, the robot will maintain the posture corresponding to the last valid value. For example, (155,0) corresponds to the maximum forward extension, (0,155) corresponds to the maximum upward extension, and (155,155) is the maximum diagonal upward extension. However, if the robot cannot reach this position, it will maintain the valid position sent last time.

4. arm_mode (switch) switch: the self-stabilization mode switch of the robotic arm, 0 is off and 1 is on . After turning it on, the end of the robotic arm will not translate with the translation of the body (translation refers to the movement of the torso while standing on all fours, rather than moving forward, backward, left, or right).

The effect of turning on:

```
def dog_arm(x,z):
    dog.arm(x,z)

interact(dog_arm,\
        x = widgets.IntSlider(min=-80,max=155,step=1,value=-25),\
        z = widgets.IntSlider(min=-95,max=155,step=1,value=40))
```

x −25

z 40

[3]: <function __main__.dog_arm(x, z)>

[2]:

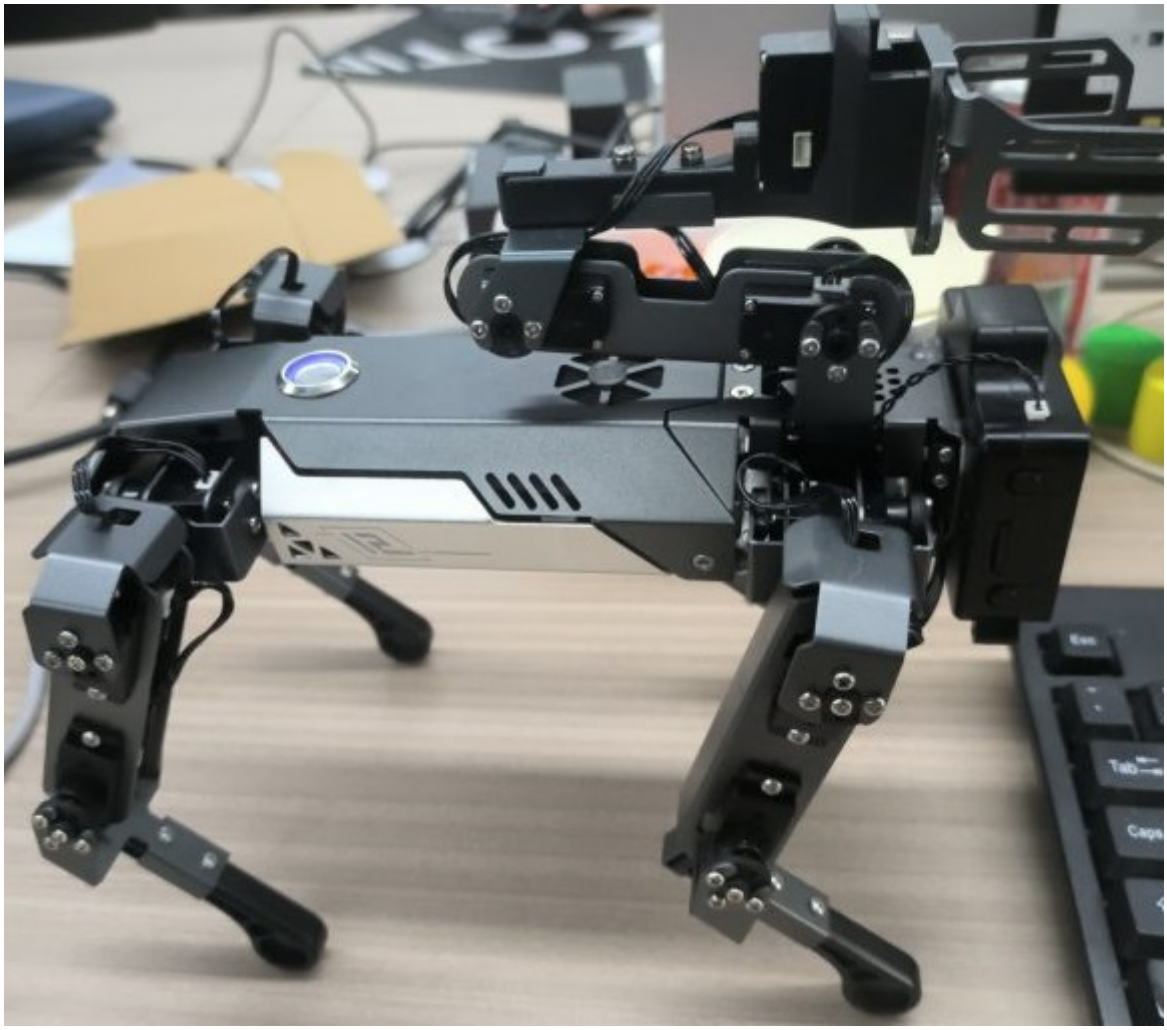
```
#机械臂自稳 -这个要用机械臂坐标控制
def dog_arm_mode(switch):
    dog.arm_mode(switch)
interact(dog_arm_mode,\
        switch = widgets.IntSlider(min=0,max=1,step=1,value=0))

def dog_translation(valuex,valuey,valuez):
    dog.translation(['x','y','z'],[valuex,valuey,valuez])

interact(dog_translation,\
        valuex = widgets.IntSlider(min=-35,max=35,step=1,value=0),\
        valuey = widgets.IntSlider(min=-18,max=18,step=1,value=0),\
        valuez = widgets.IntSlider(min=75,max=115,step=1,value=90))
```

switch 1

valuex 0

valuey 1

valuez 114



robotic_arm_control.ipynb

Code

Python 3 (ipykernel)

```
def dog_arm(x,z):
    dog.arm(x,z)

interact(dog_arm,\
        x = widgets.IntSlider(min=-80,max=155,step=1,value=-25),\
        z = widgets.IntSlider(min=-95,max=155,step=1,value=40))
```

x −25

z 40

[3]: <function __main__.dog_arm(x, z)>

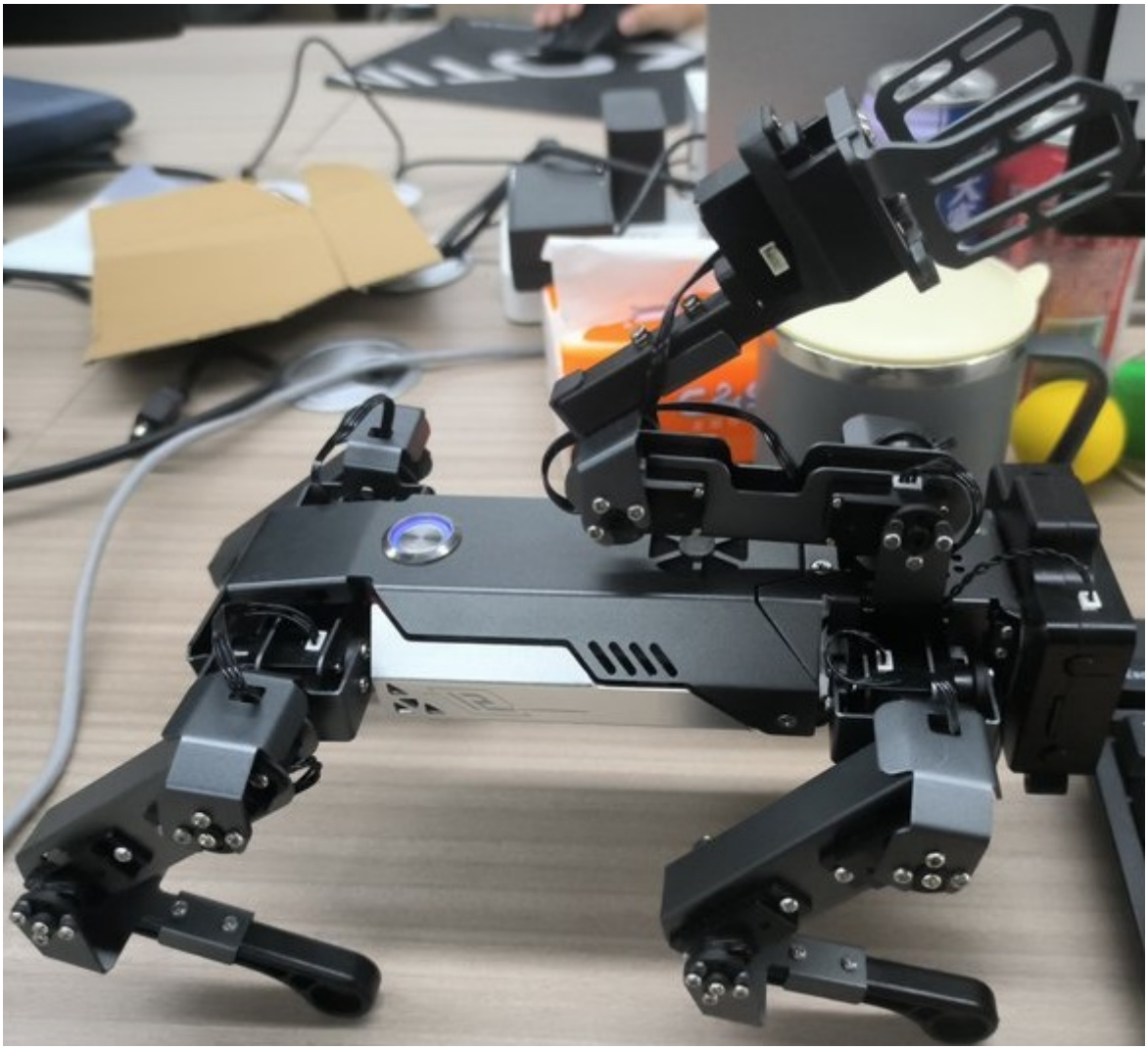
[2]:
```
#机械臂自稳 -这个要用机械臂坐标控制
def dog_arm_mode(switch):
    dog.arm_mode(switch)
interact(dog_arm_mode,\
        switch = widgets.IntSlider(min=0,max=1,step=1,value=0))

def dog_translation(valuex,valuey,valuez):
    dog.translation(['x','y','z'],[valuex,valuey,valuez])

interact(dog_translation,\
        valuex = widgets.IntSlider(min=-35,max=35,step=1,value=0),\
        valuey = widgets.IntSlider(min=-18,max=18,step=1,value=0),\
        valuez = widgets.IntSlider(min=75,max=115,step=1,value=90))
```

switch 1

valuex 0

valuey 1

valuez 75



The height of the end of the robotic arm is always basically the same from the ground.

The effect of not enabling:

robotic_arm_control.ipynb

Code

Python 3 (ipykernel)

```
def dog_arm(x,z):
    dog.arm(x,z)

interact(dog_arm,\
        x = widgets.IntSlider(min=-80,max=155,step=1,value=-25),\
        z = widgets.IntSlider(min=-95,max=155,step=1,value=40))
```

x −25

z 40

[3]: <function __main__.dog_arm(x, z)>

[2]:
```
#机械臂自稳 -这个要用机械臂坐标控制
def dog_arm_mode(switch):
    dog.arm_mode(switch)
interact(dog_arm_mode,\
        switch = widgets.IntSlider(min=0,max=1,step=1,value=0))

def dog_translation(valuex,valuey,valuez):
    dog.translation(['x','y','z'],[valuex,valuey,valuez])

interact(dog_translation,\
        valuex = widgets.IntSlider(min=-35,max=35,step=1,value=0),\
        valuey = widgets.IntSlider(min=-18,max=18,step=1,value=0),\
        valuez = widgets.IntSlider(min=75,max=115,step=1,value=90))
```

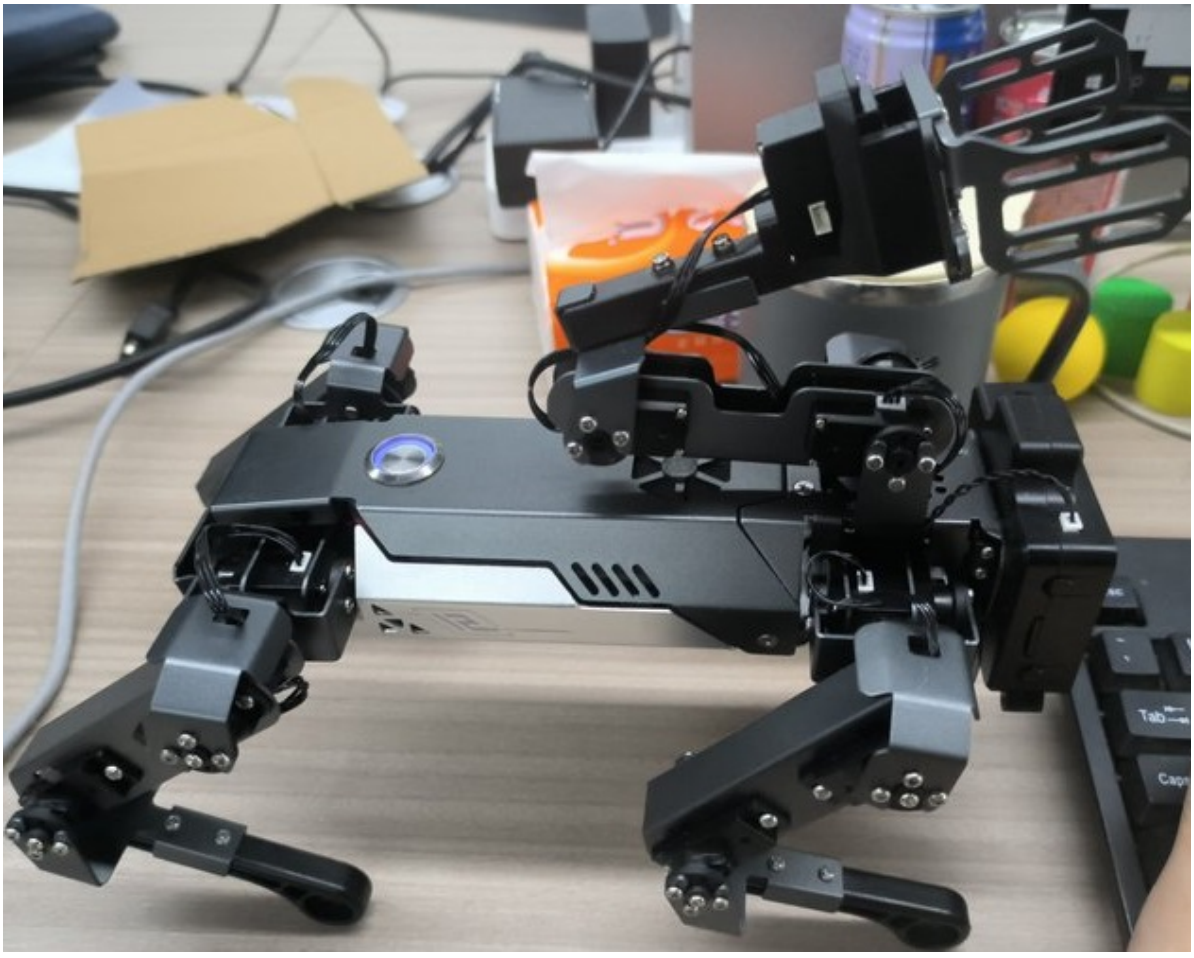
switch 0

valuex 0

valuey 1

valuez 75

[2]: <function __main__.dog_translation(valuex, valuey, valuez)>



robotic_arm_control.ipynb

Code

Python 3 (ipykernel)

```
def dog_arm(x,z):
    dog.arm(x,z)

interact(dog_arm,\
        x = widgets.IntSlider(min=-80,max=155,step=1,value=-25),\
        z = widgets.IntSlider(min=-95,max=155,step=1,value=40))
```

x −25

z 40

[3]: <function __main__.dog_arm(x, z)>

[2]:

```
#机械臂自稳 -这个要用机械臂坐标控制
def dog_arm_mode(switch):
    dog.arm_mode(switch)
interact(dog_arm_mode,\
        switch = widgets.IntSlider(min=0,max=1,step=1,value=0))

def dog_translation(valuex,valuey,valuez):
    dog.translation(['x','y','z'],[valuex,valuey,valuez])

interact(dog_translation,\
        valuex = widgets.IntSlider(min=-35,max=35,step=1,value=0),\
        valuey = widgets.IntSlider(min=-18,max=18,step=1,value=0),\
        valuez = widgets.IntSlider(min=75,max=115,step=1,value=90))
```

switch 0

valuex 0

valuey 1

valuez 115