

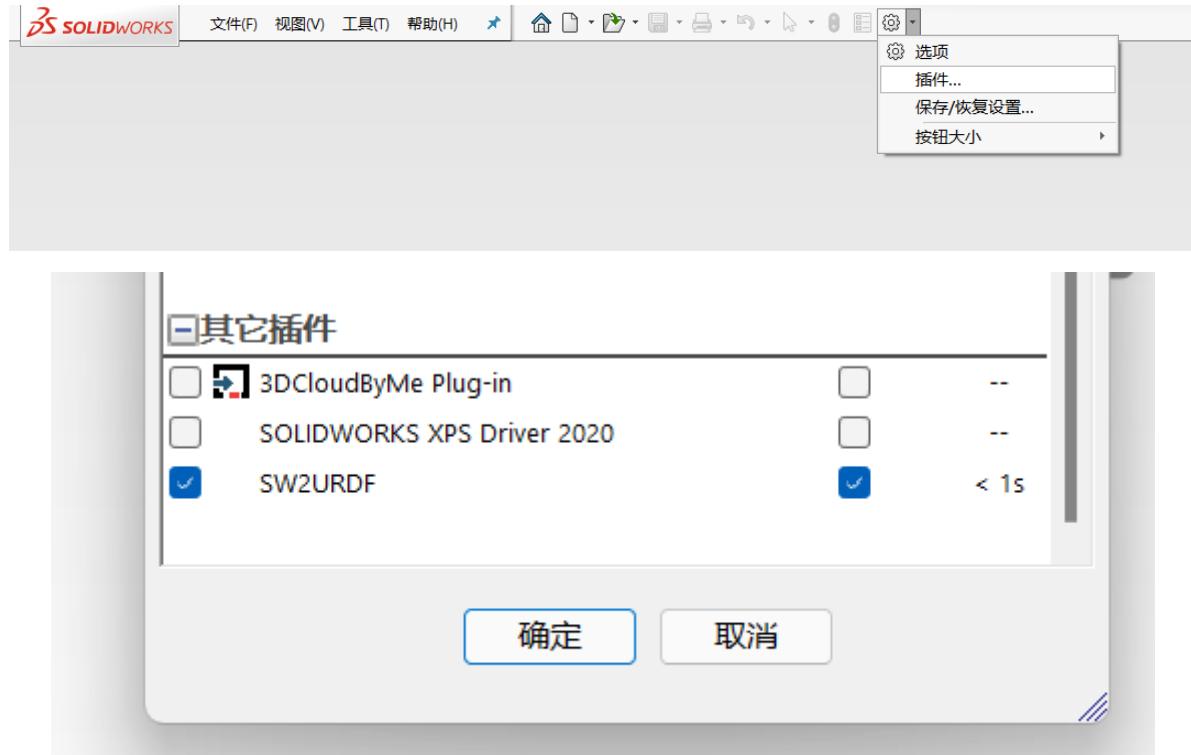
环境说明：solidworks2020, webots

一、插件安装

1、SolidWorks插件solidworks_urdf_exporter

下载[solidworks urdf exporter](#)后安装即可，然后打开SolidWorks。

点击小齿轮旁边的下拉菜单，选择插件，然后拉到插件最底下有个sw2urdf的插件，点击选中，然后保存。



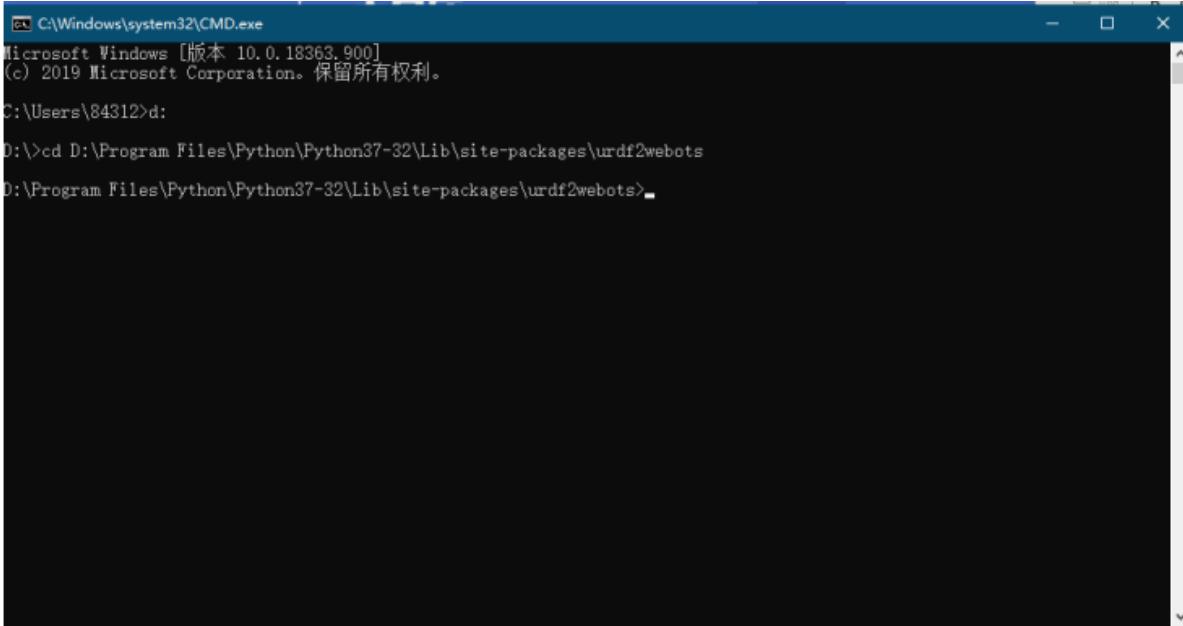
2、Python插件urdf2webots

将压缩文件解压至

```
.....\Python\Python37-32\Lib\site-packages
```

打开CMD切换至文件

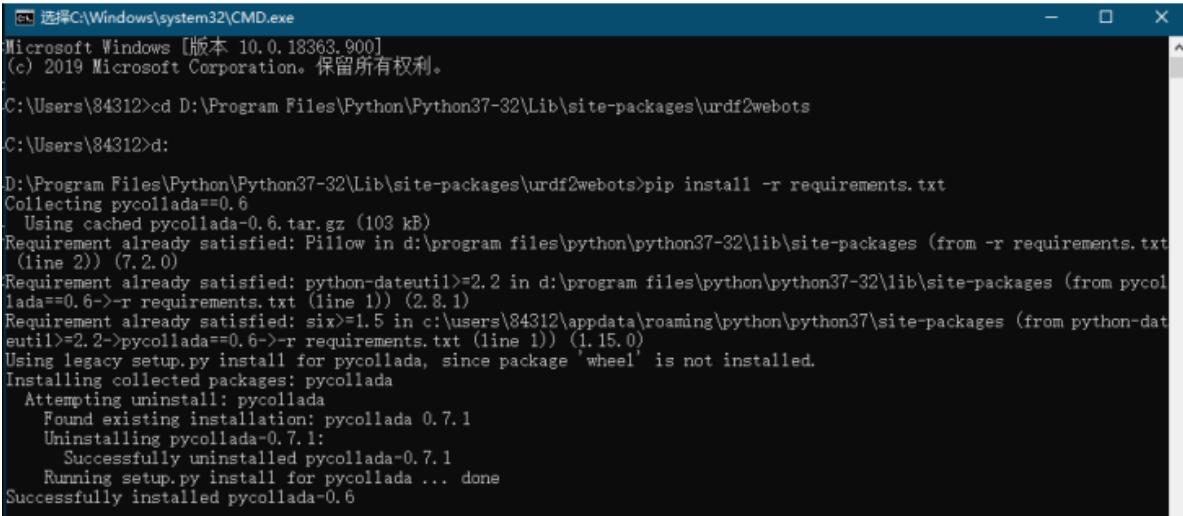
```
.....\Python\Python37-32\Lib\site-packages\urdf2webots
```



```
C:\Windows\system32\cmd.exe
Microsoft Windows [版本 10.0.18363.900]
(c) 2019 Microsoft Corporation. 保留所有权利。
C:\Users\84312>d:
D:>cd D:\Program Files\Python\Python37-32\Lib\site-packages\urdf2webots
D:\Program Files\Python\Python37-32\Lib\site-packages\urdf2webots>
```

输入指令

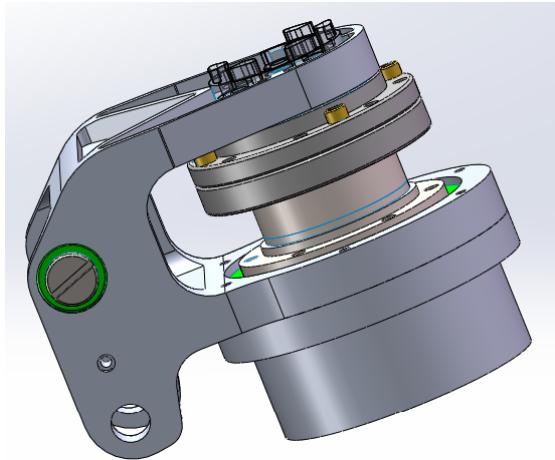
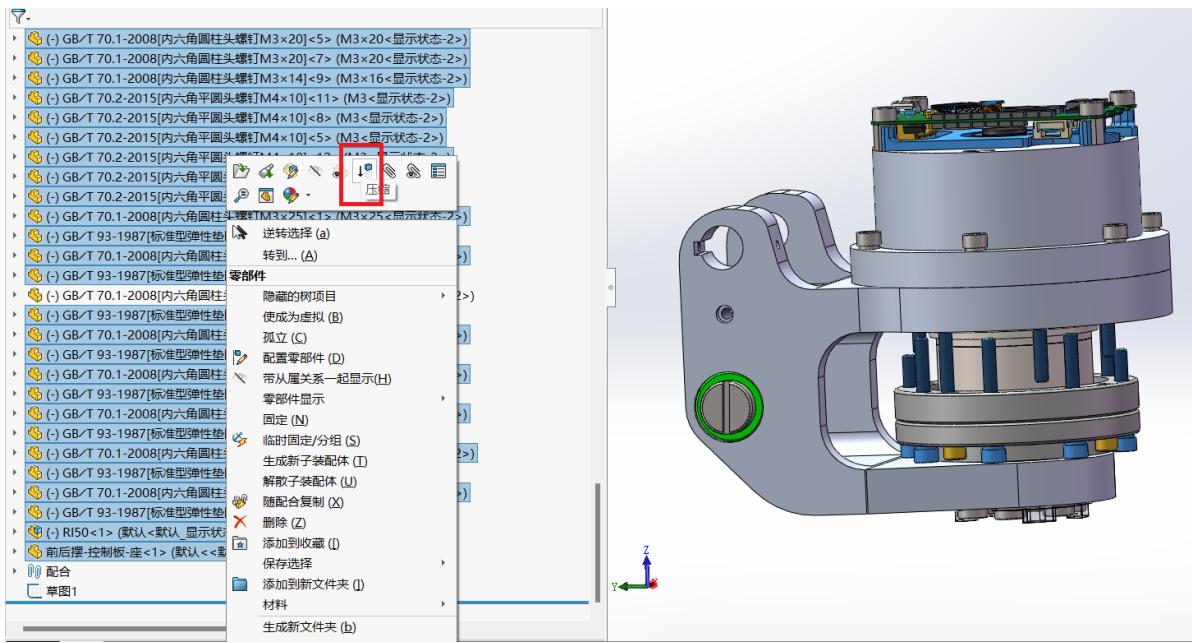
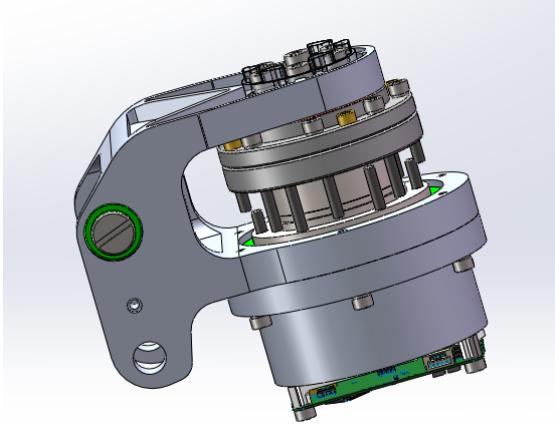
```
pip install -r requirements.txt
```



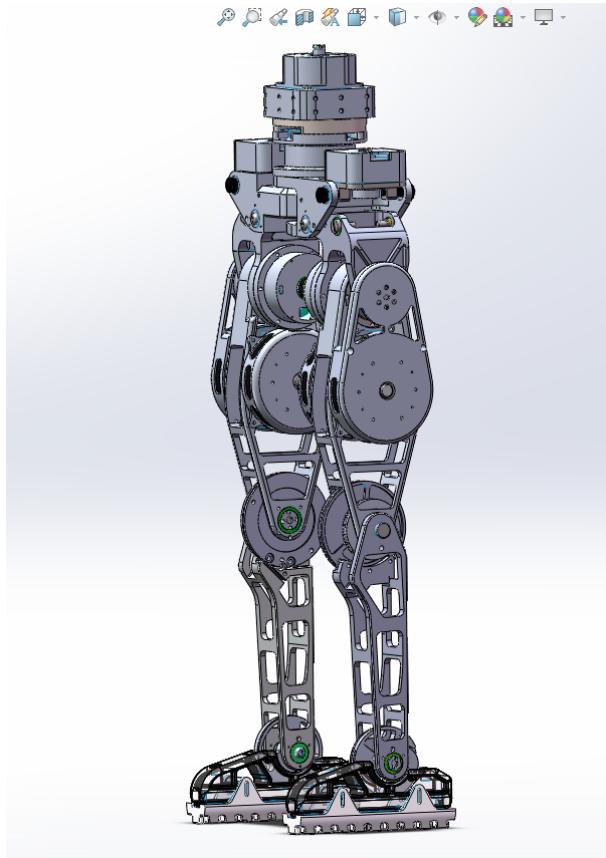
```
选择C:\Windows\system32\cmd.exe
Microsoft Windows [版本 10.0.18363.900]
(c) 2019 Microsoft Corporation. 保留所有权利。
C:\Users\84312>cd D:\Program Files\Python\Python37-32\Lib\site-packages\urdf2webots
C:\Users\84312>d:
D:\Program Files\Python\Python37-32\Lib\site-packages\urdf2webots>pip install -r requirements.txt
Collecting pycollada==0.6
  Using cached pycollada-0.6.tar.gz (103 kB)
Requirement already satisfied: Pillow in d:\program files\python\python37-32\lib\site-packages (from -r requirements.txt (line 2)) (7.2.0)
Requirement already satisfied: python-dateutil>=2.2 in d:\program files\python\python37-32\lib\site-packages (from pycollada==0.6->-r requirements.txt (line 1)) (2.8.1)
Requirement already satisfied: six>=1.5 in c:\users\84312\appdata\roaming\python\python37\site-packages (from python-dateutil>=2.2->pycollada==0.6->-r requirements.txt (line 1)) (1.15.0)
Using legacy setup.py install for pycollada, since package 'wheel' is not installed.
Installing collected packages: pycollada
  Attempting uninstall: pycollada
    Found existing installation: pycollada 0.7.1
    Uninstalling pycollada-0.7.1:
      Successfully uninstalled pycollada-0.7.1
      Running setup.py install for pycollada ... done
      Successfully installed pycollada-0.6
```

二、模型简化和装配

在solidworks中对需要的SLDASM文件格式的模型进行简化，通过压缩来去掉不需要的零件（例如螺钉等不影响外观的零部件），以保留基本外观，并重新进行装配。



总体简化图如下图所示，并约束下半身的位置，使其保持竖直站立。



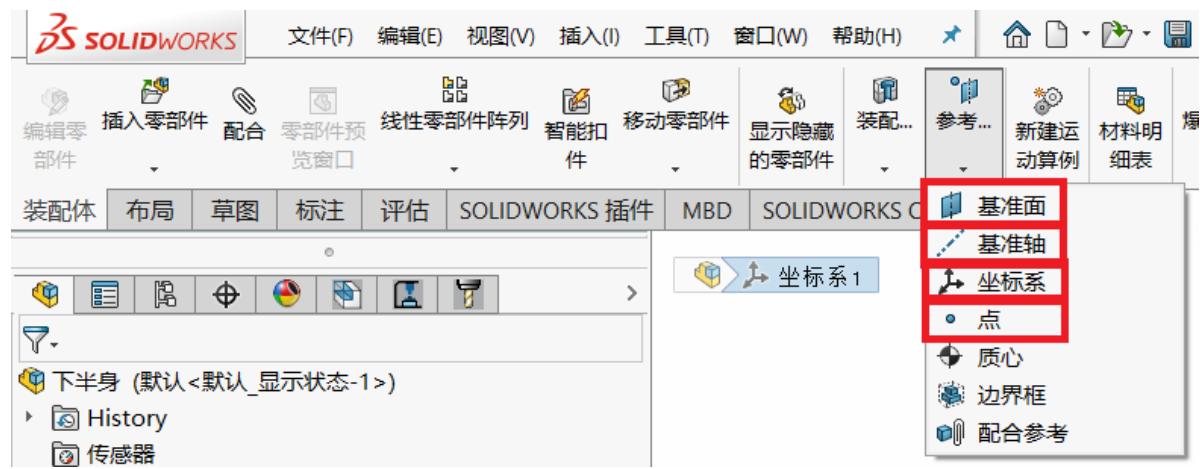
三、建立坐标系

双足的URDF文件转化关键是在建坐标系，双足的每个关节都在围绕自己的独立坐标系运动。

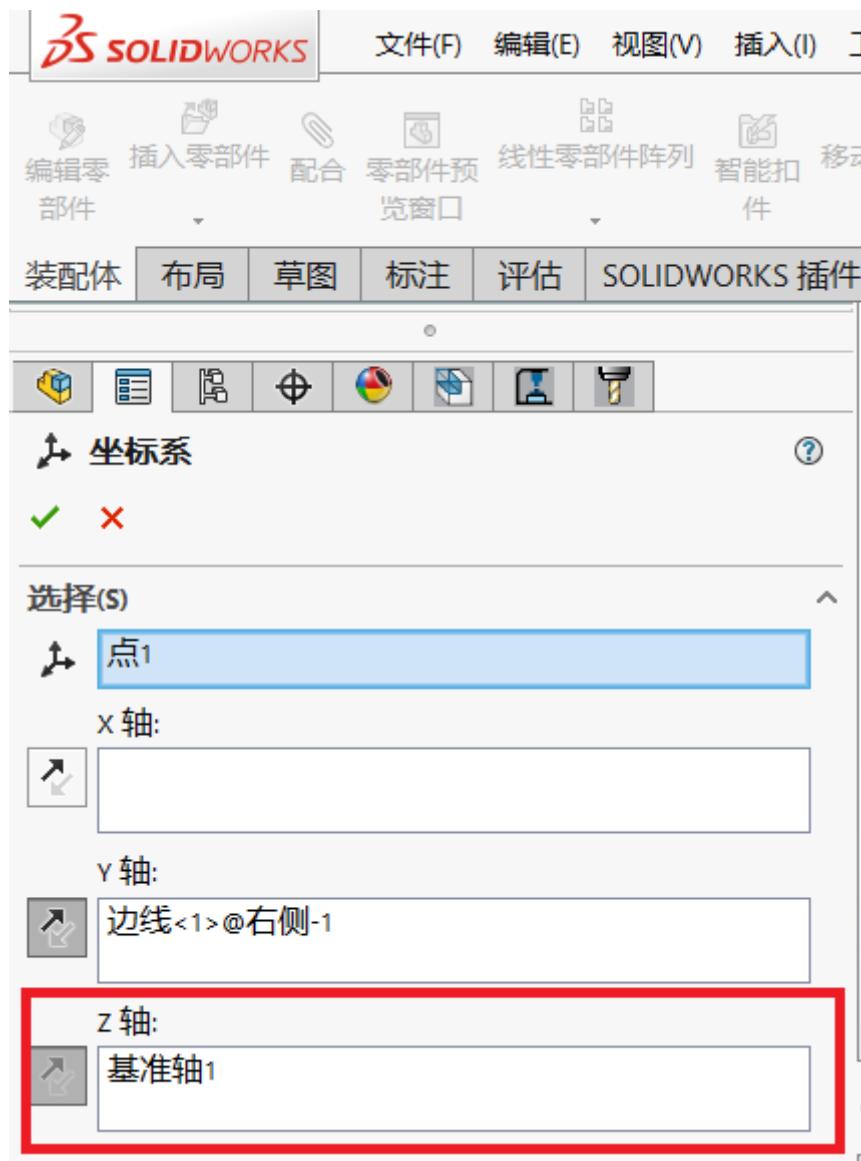
这个双足一共9个关节，我们也就需要建立9个坐标系，还要加上一个初始坐标系，这是用于确定整个双足的位置的。

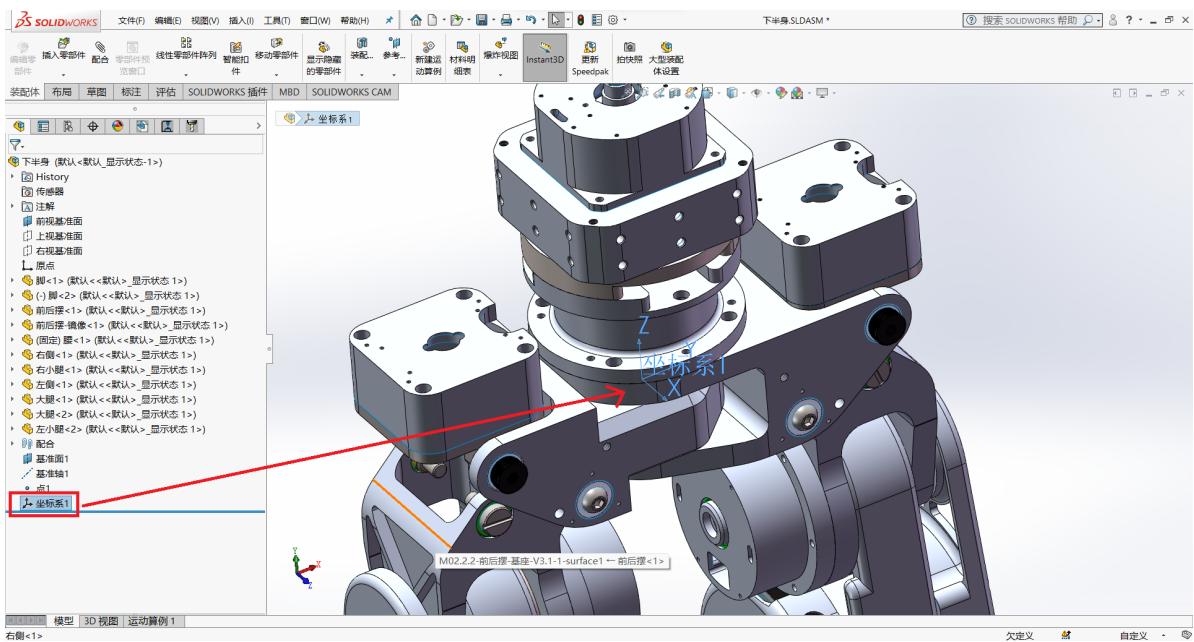
- 1、建立坐标系前明确各部位是如何运动的，绕哪一根轴做怎样的运动，在双足中都是旋转运动。
- 2、所有坐标系都设置成Z轴朝上，X朝前。
- 3、建立坐标系时，只需要确定两根轴就可以，这里建议首先确定Z轴。
- 4、基准轴（关节旋转轴线）的命名需要严格按照命名规则。
- 5、建立坐标系时可以根据自己的习惯，对相应的坐标系合理命名，但是要注意区分左右，根据机器人自身的视角来命名左右。

首先建立基准面和基准轴，根据基准面和基准轴的交点来建立参考点，选取参考点为坐标系，并确定XYZ轴的方向

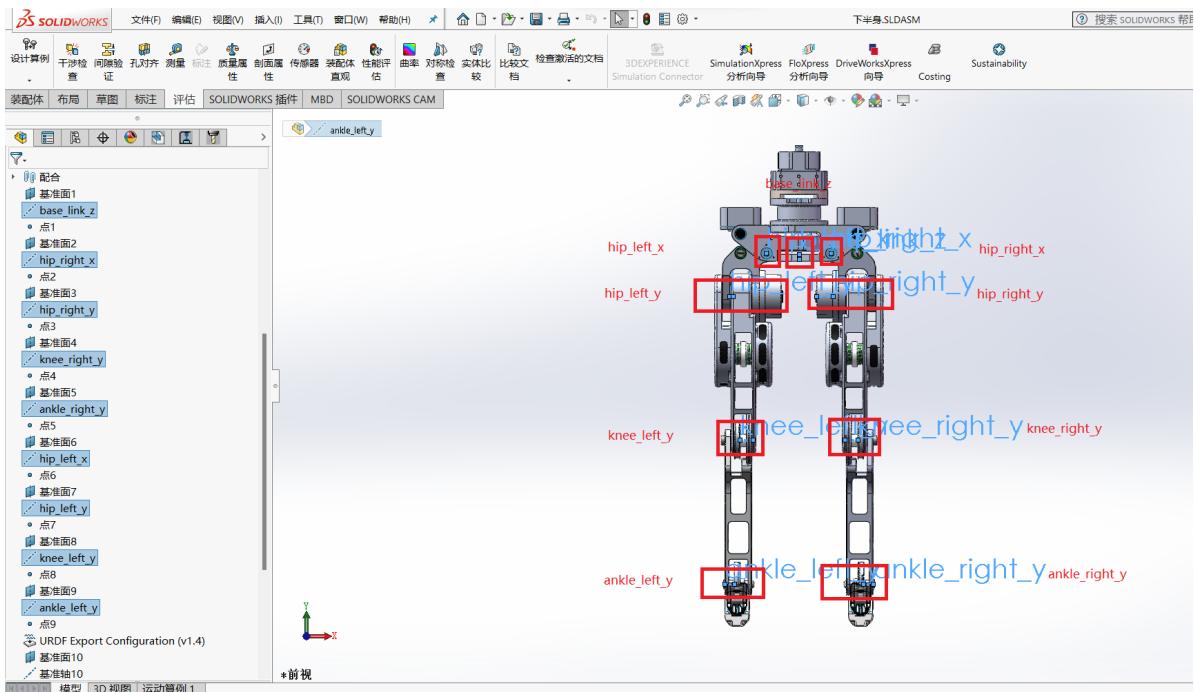


轴的方向可以参考竖直和水平的双足零件的线条来确定，同时注意轴的正方向

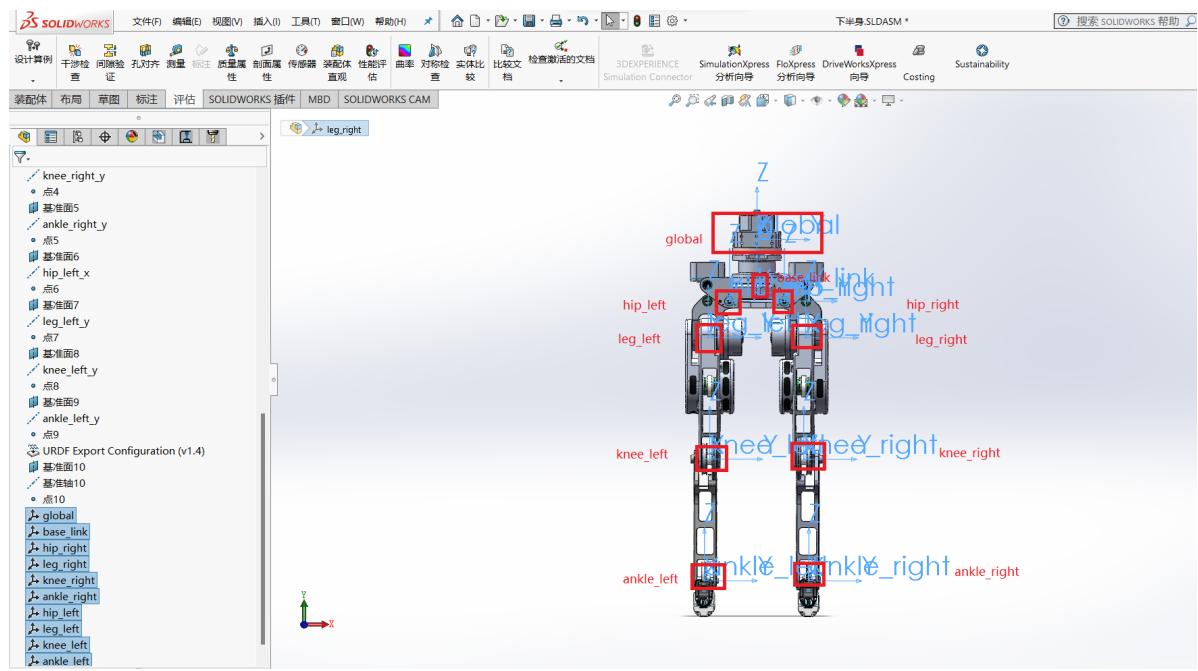




基准轴的命名如下图所示，必须严格按照要求命名，同时注意区分左右（站在机器人的视角）

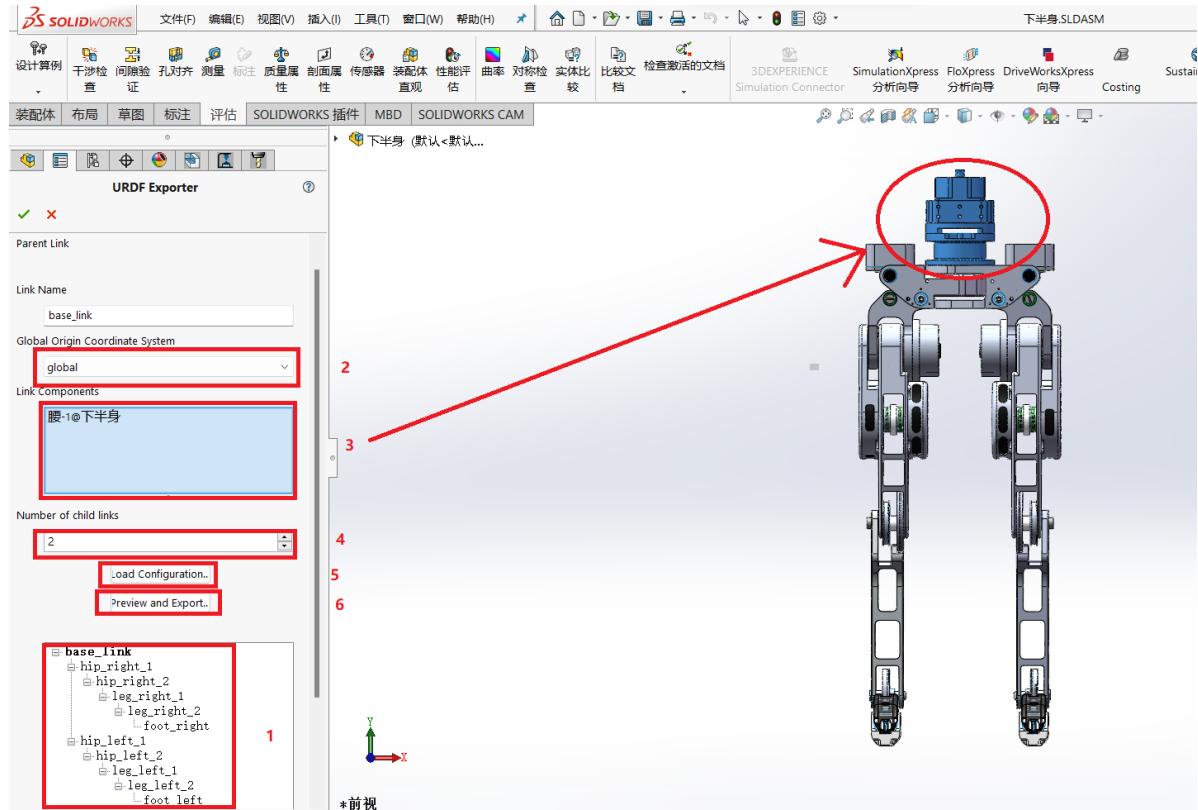


坐标系的命名根据个人习惯，同时注意区分左右（站在机器人的视角）



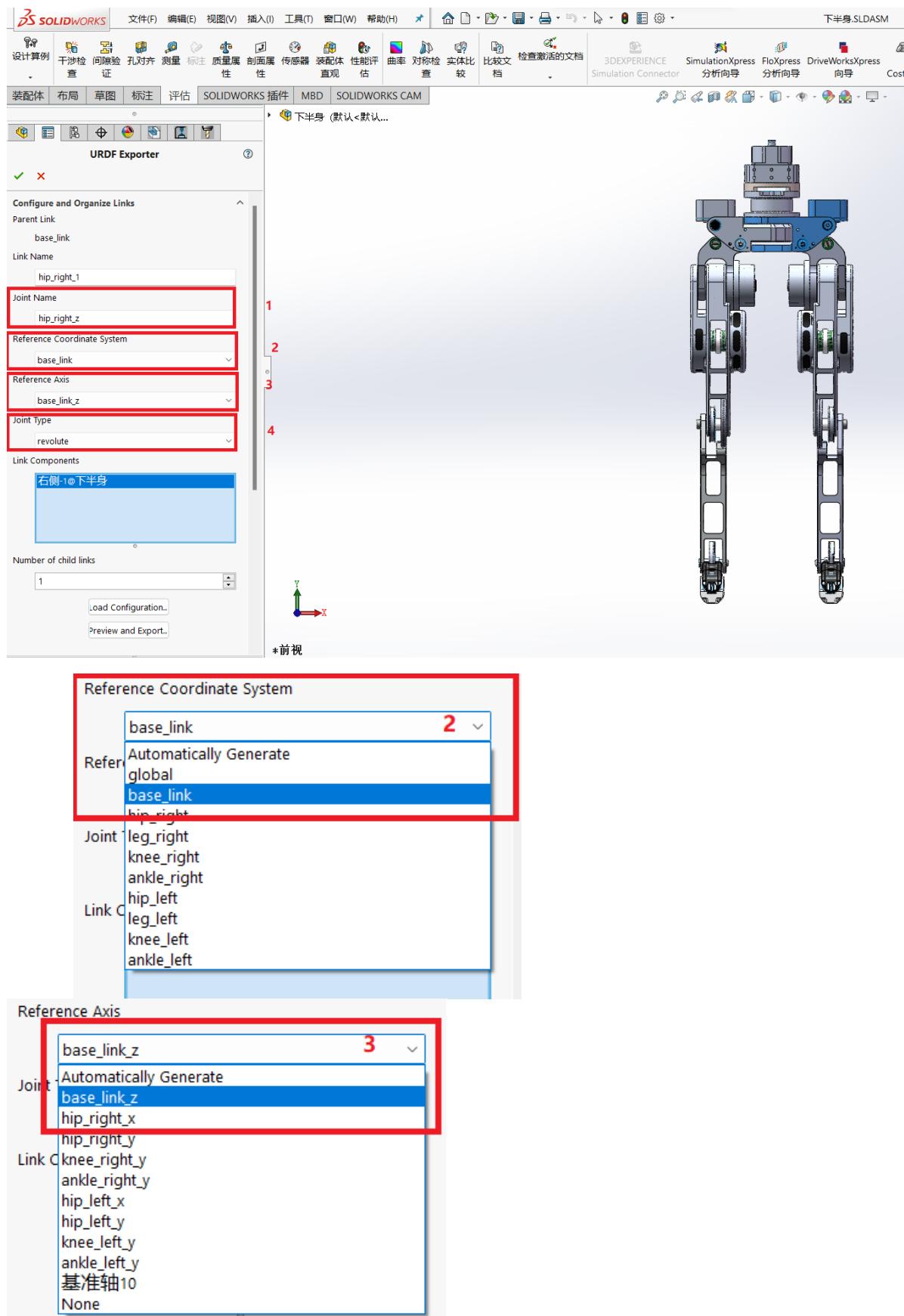
四、输出URDF文件设置

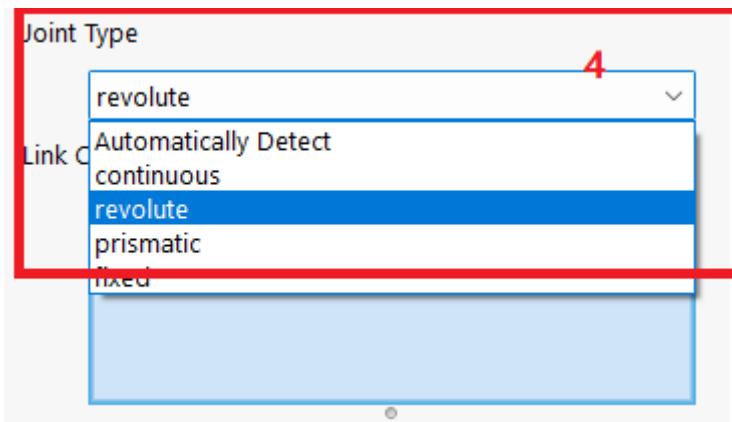
“工具”---“File”---“Export as URDF”，进入设置，具体画面如下图



- 1、这是对应双足的各个关节，在设置时先选择一个关节进行修改，上图是在对基准线进行设置。
- 2、这是选择初始坐标系。
- 3、这一块选择的是在当前坐标系下关联的的部位，即选中的蓝色部分是在当前坐标系下进行运算，在选择这一部分时记住不要和其他坐标系的部位选择重复，否则会在两个坐标系里都出现该部位。
- 4、这是选中的部位的子关节的数目，在这个双足有子关节的情况下，除了腰部的子关节数目是2外，其他子关节数目都是1。
- 5、这是从一个CSV文件中导入相应参数的按钮，我们没有用到。

6、这是查看生成的URDF文件并准备生成一个URDF package的按钮，等各关节的坐标系选择完毕，子关节部位及数目设定完后再点击，进行后面的设置。

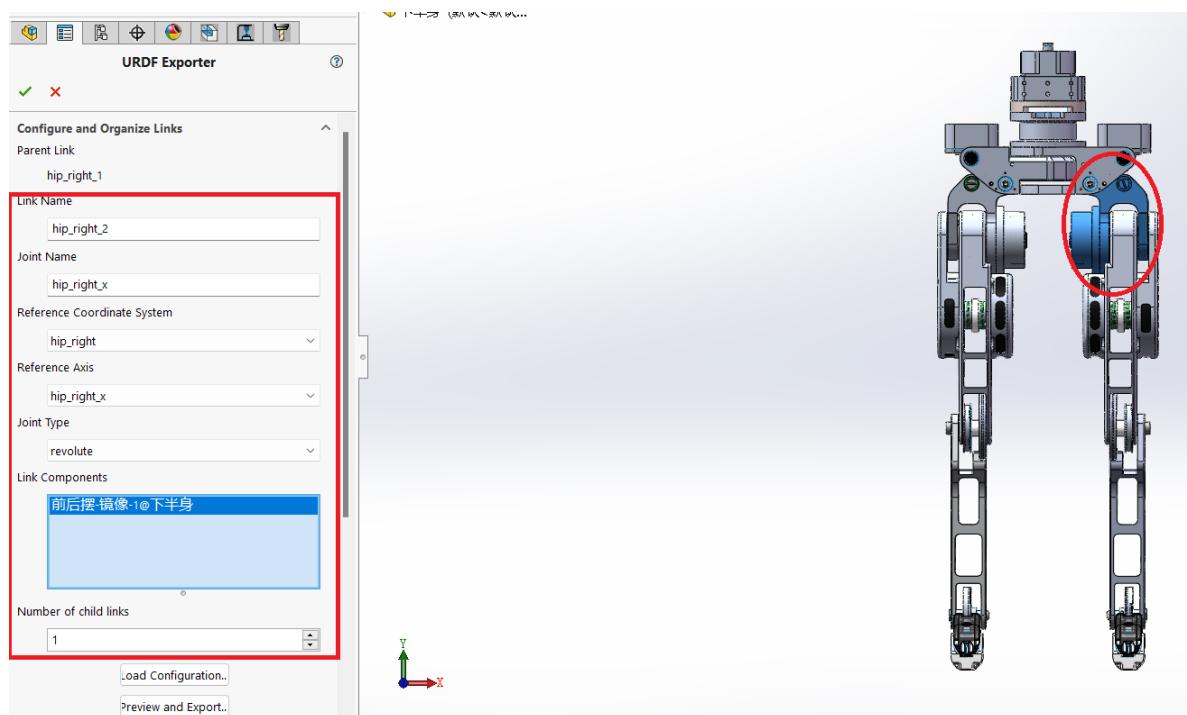


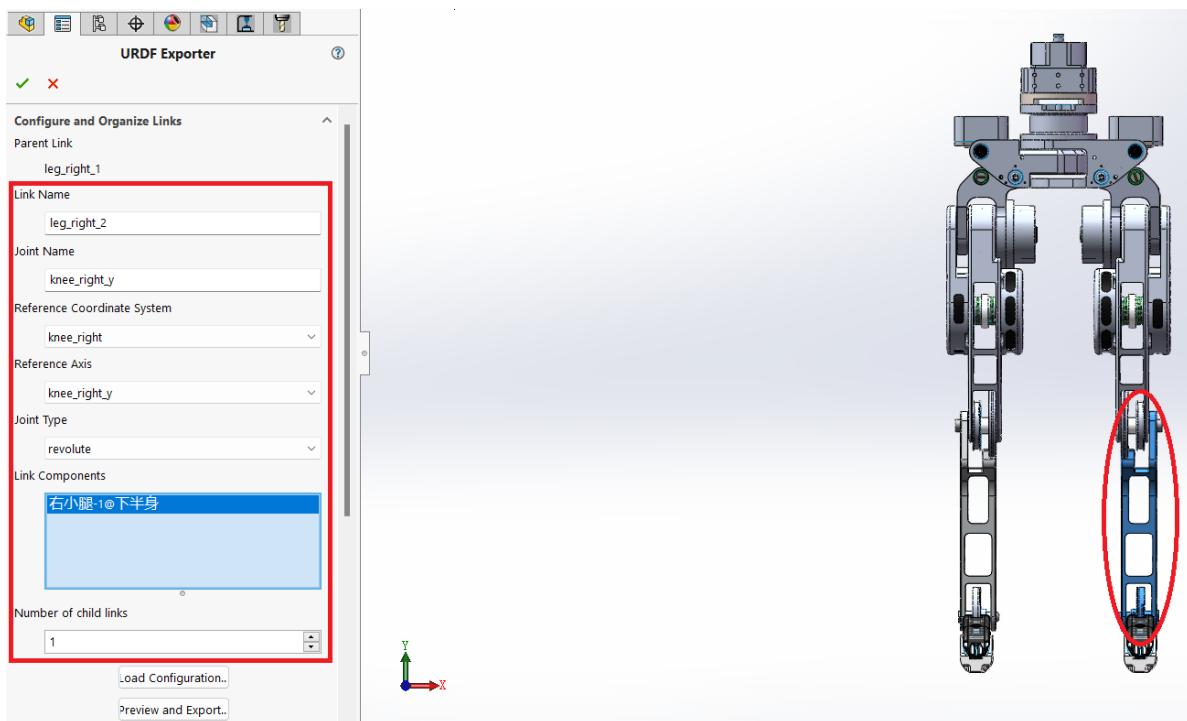
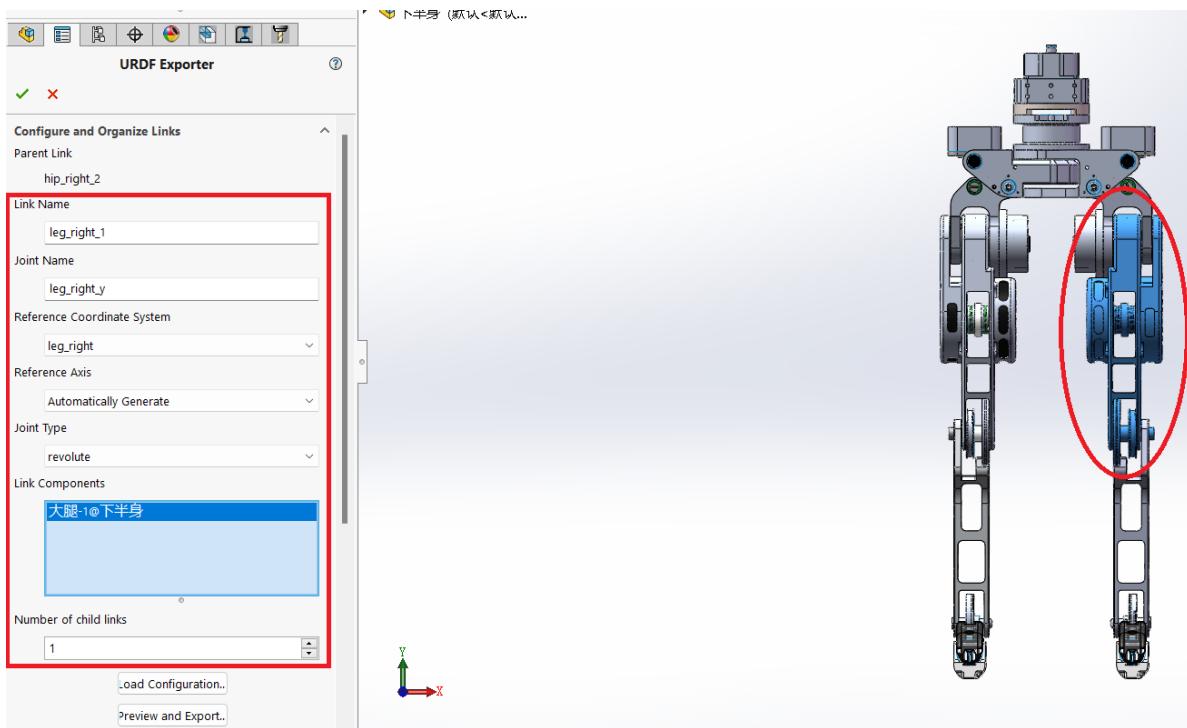


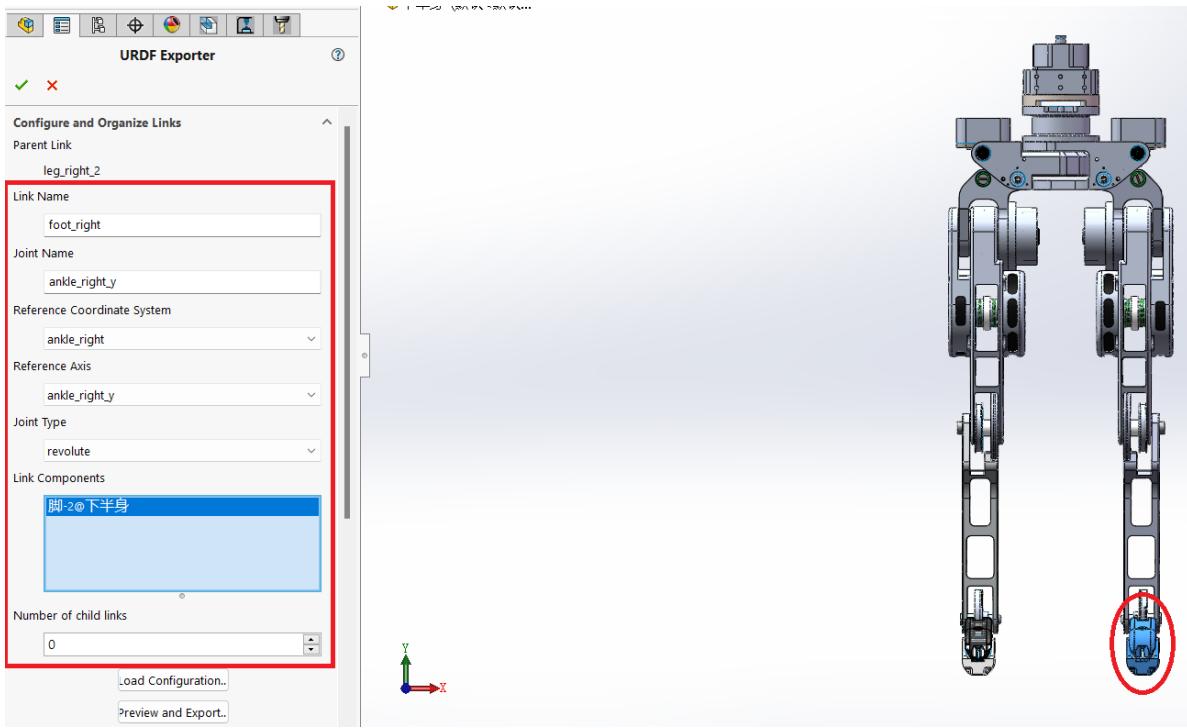
上图是在设置hip_right_1关节

- 1、关节名：hip_right_1
- 2、相匹配的坐标系：base_link （这是前面建立的坐标系）
- 3、与运动相关的轴：base_link_z （前面用于确定Z轴的基准轴，在这里做旋转轴）
- 4、关节类型：revolute （确定关节运动类型，此处是旋转的）

后面几个关节设置相似，在此不再赘述，贴上右腿的设置图片，左腿





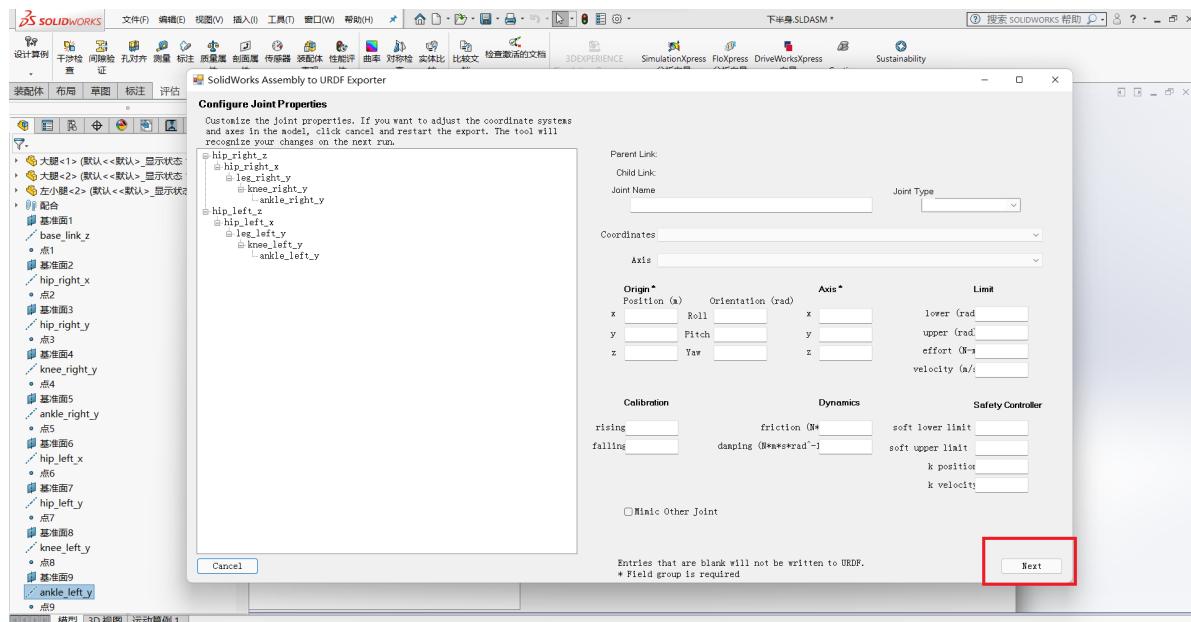


在最后foot后面已经没有子关节了，子关节数目是0

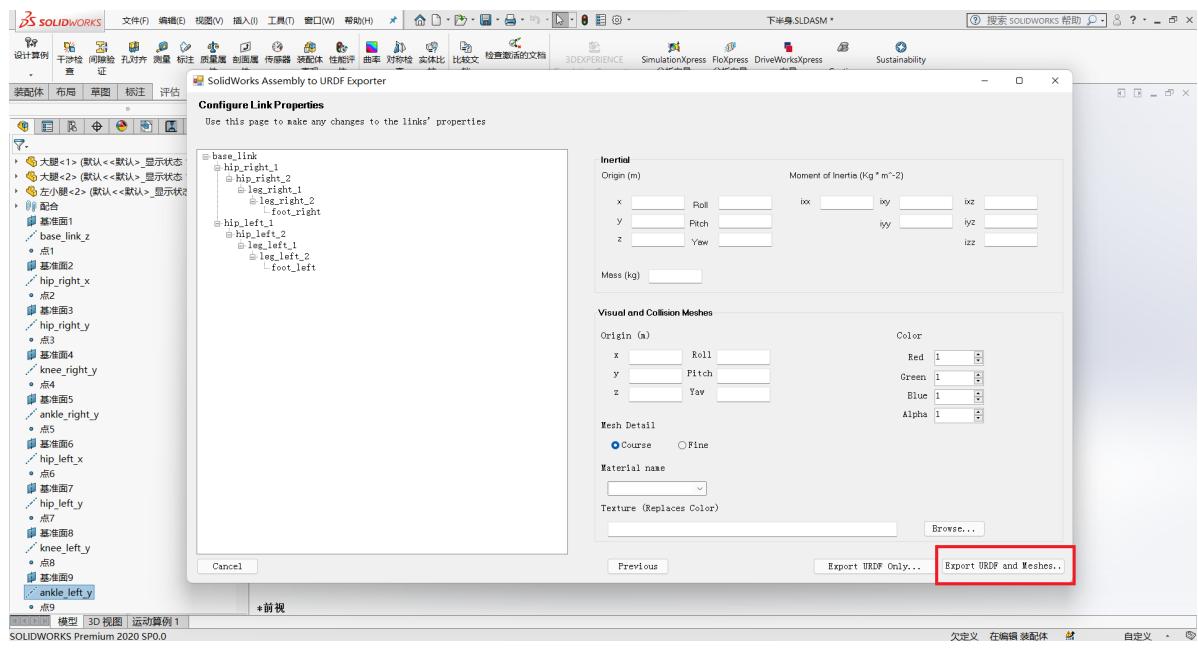
这些设置完成后，点击生成URDF文件并生成一个URDF package的按钮，会弹出一个窗口，我们要确保每个关节的Joint type, Coordinate system, Axis设置正确

Preview and Export..

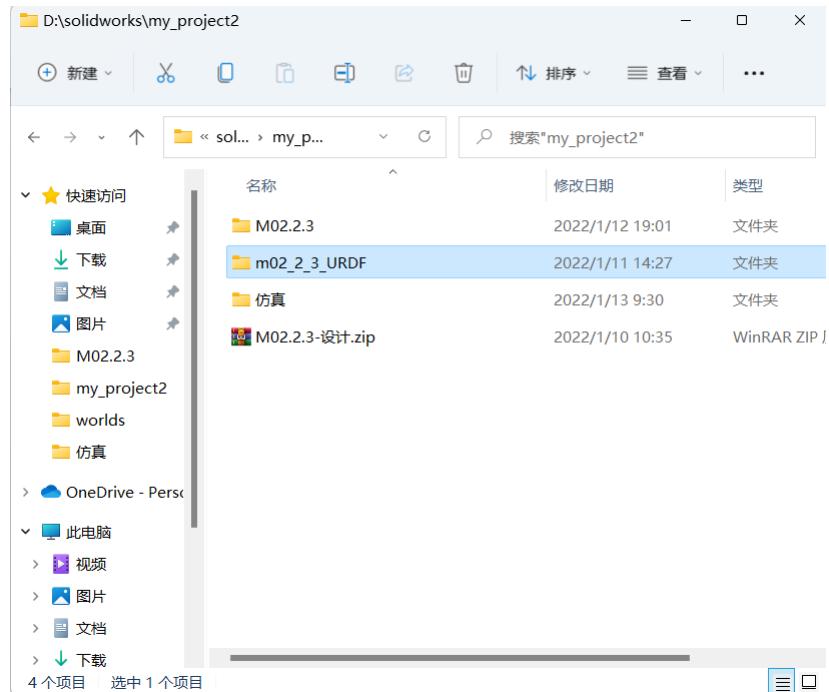
确保正确后点击Next按钮



再点击如下图所示按钮



最后选择存储位置，生成文件m02_2_3.URDF



五、urdf文件转换为webots文件

打开CMD切换文件夹到

```
.....\Python\Python37-32\Lib\site-packages\urdf2webots
```

输入指令

```
python urdf2webots.py --input=yourSelfRobotModel.urdf
```

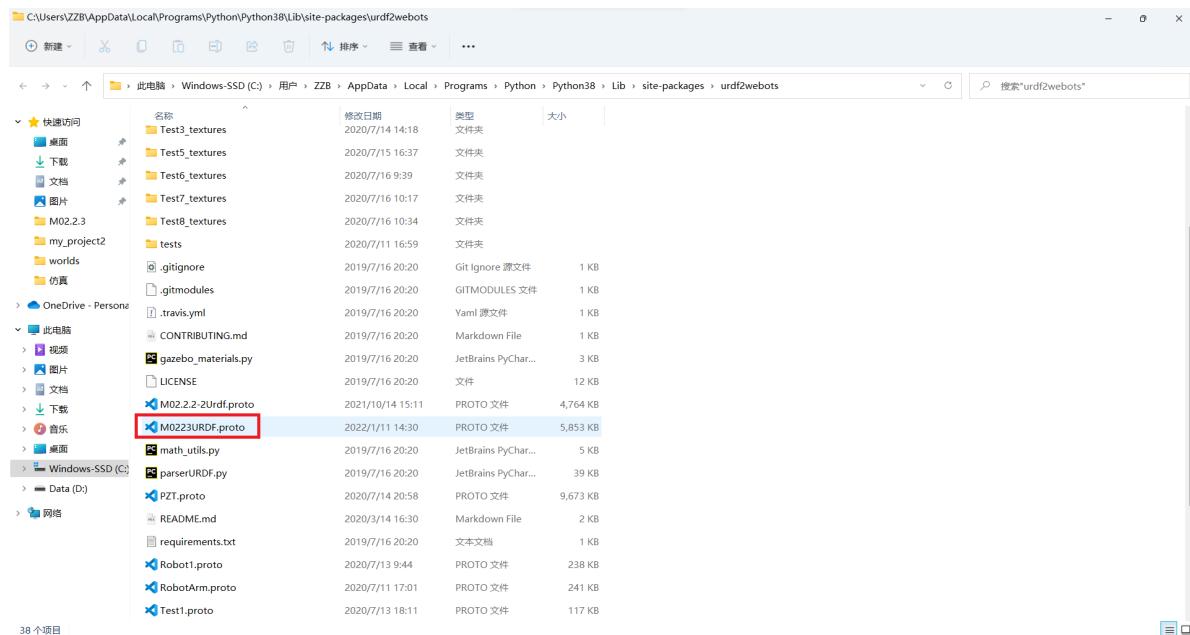
把yourSelfRobotModel.urdf替换为你的urdf文件路径

注：此处一定要将python添加到系统环境变量中，否则无法执行该指令

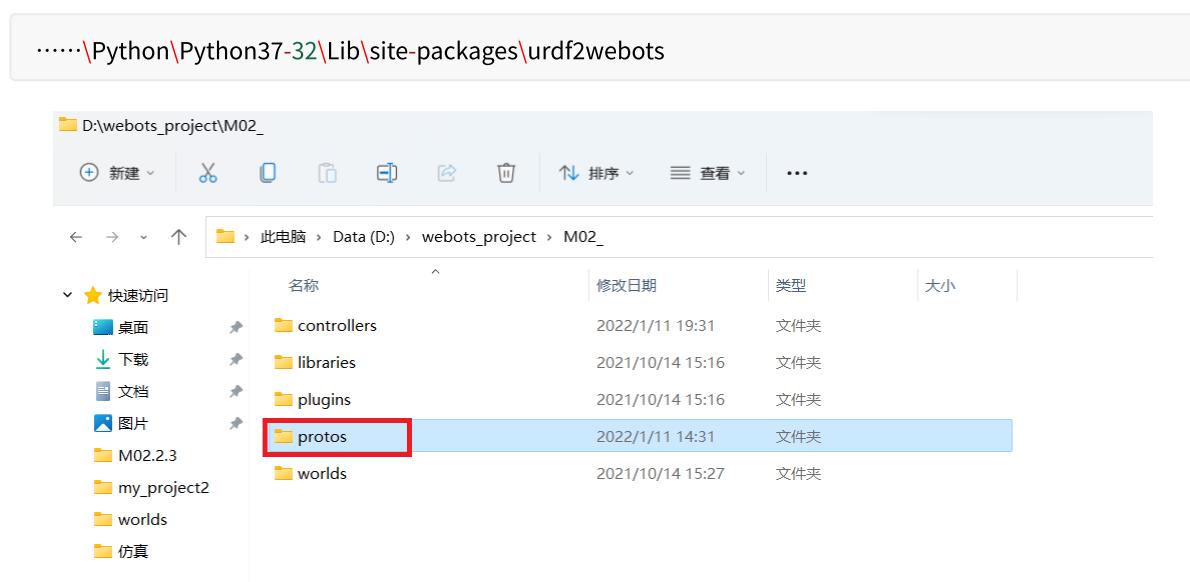
例：

```
python urdf2webots.py --input=D:\solidworks\my_project2\m02_2_3_URDF\urdf\m02_2_3_URDF.urdf
```

转换完成

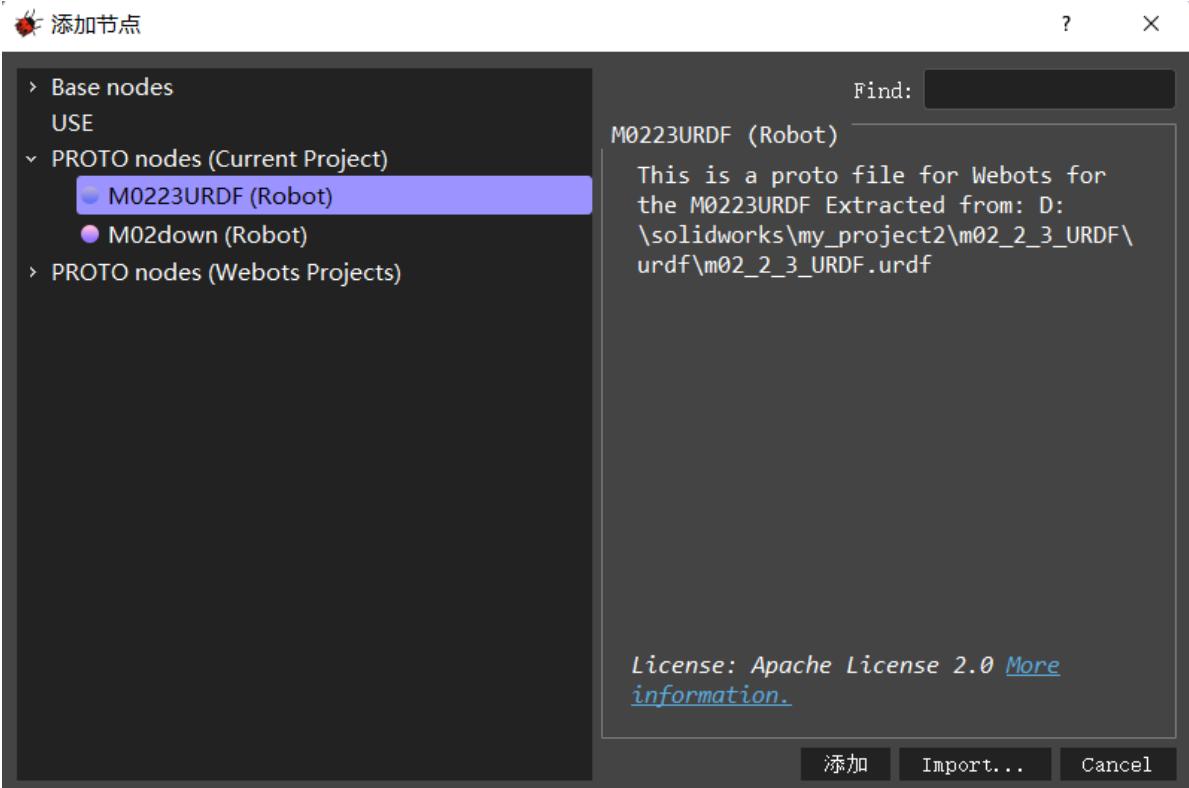


Proto文件在

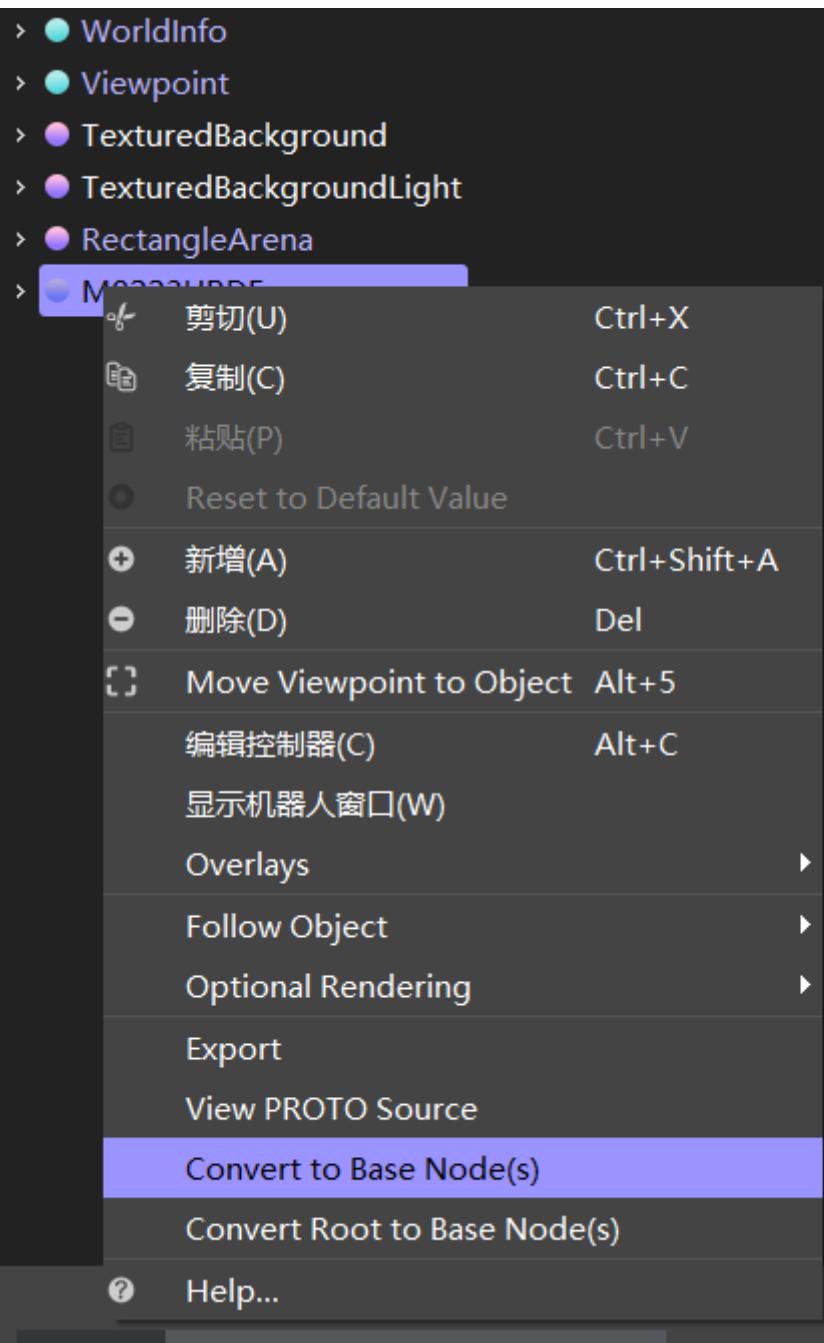


将proto文件复制到webots工程的protos文件夹中

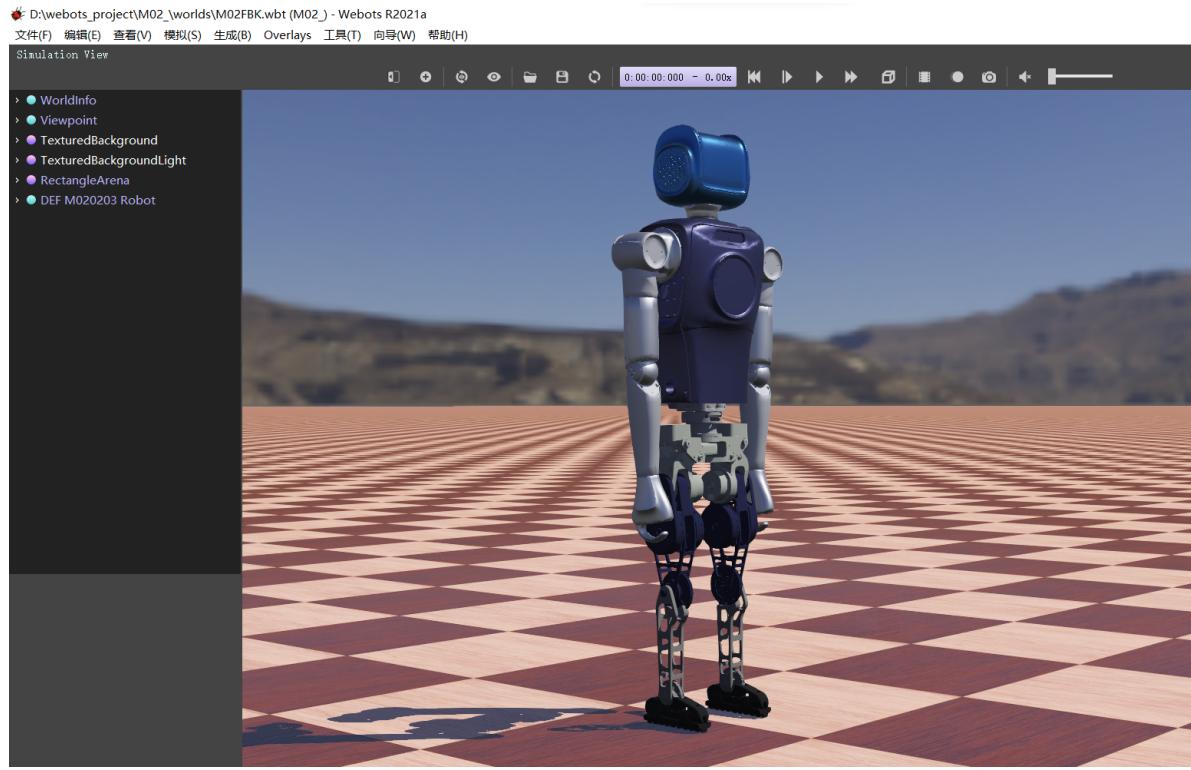
打开工程，导入proto文件



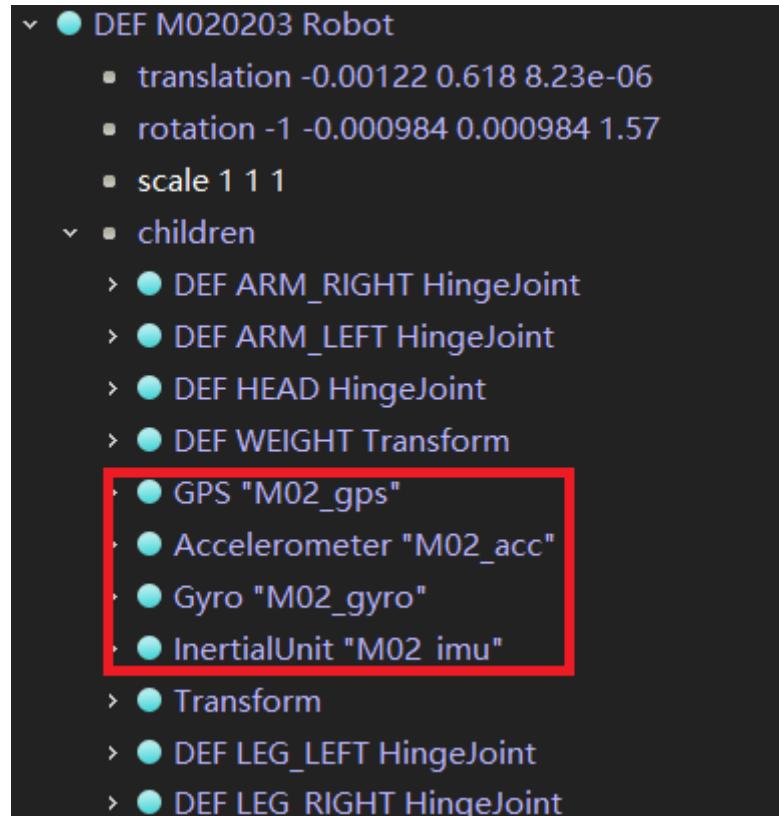
选择转换为基本节点，就可以配置了



六、webots配置



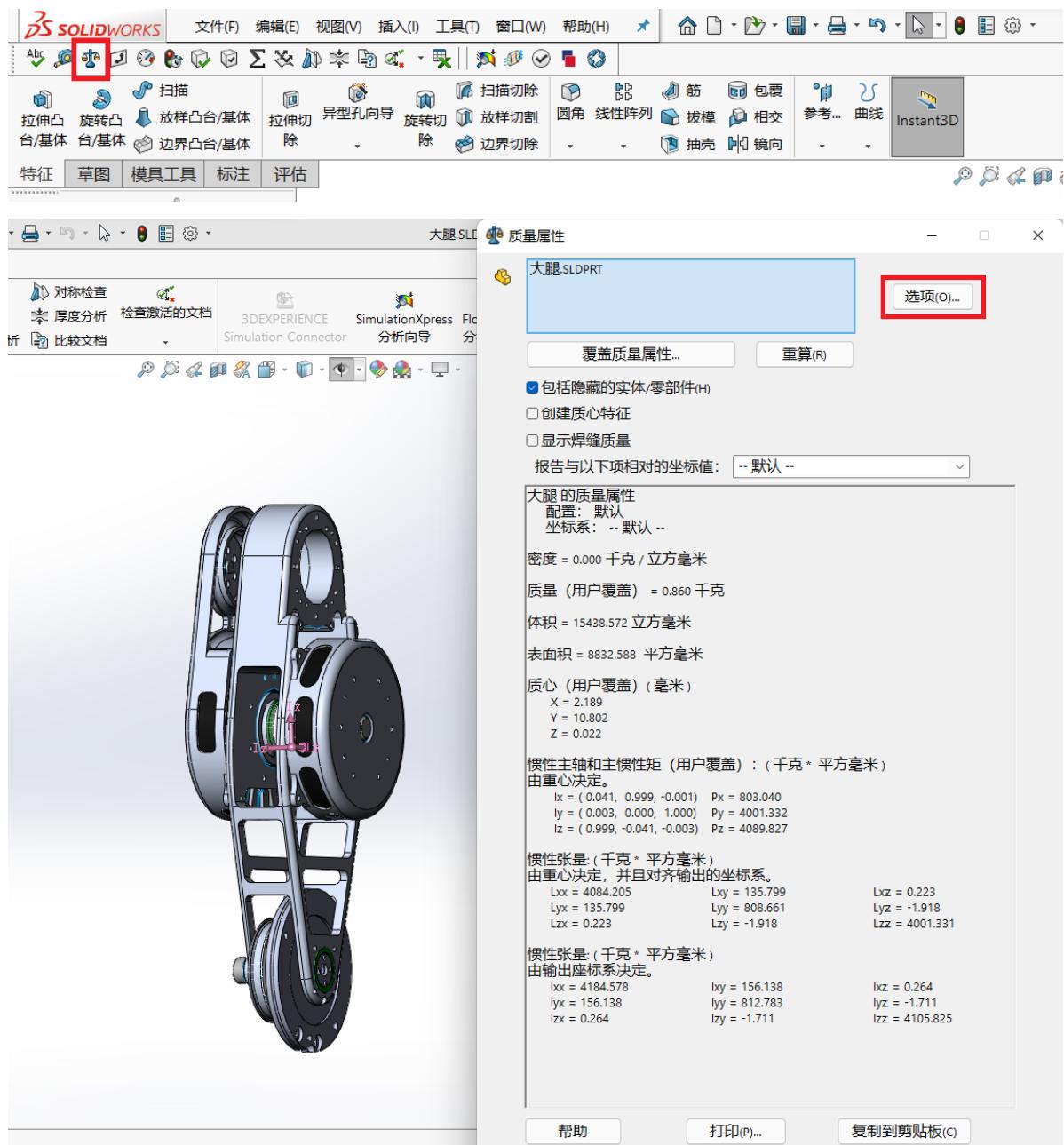
首先添加GPS、Accelerometer、Gyro、InertialUnit传感器



首先通过solidworks获取每个link的相应物理参数，点击特征选项里面的“质量属性”，然后点击“选项”，设置单位。

注：检查零件质量是否满足要求，如不满足需修改

例：



质量/剖面属性选项

X

单位

 科学记号(N) 使用文档设定(O) 使用自定义设定(U)

长度:



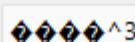
小数位数(C):

6

质量(M):



单位体积(V):



材料属性

密度(D):

0.00005572 kg/mm³

精度水准

低(较快)

高(较慢)

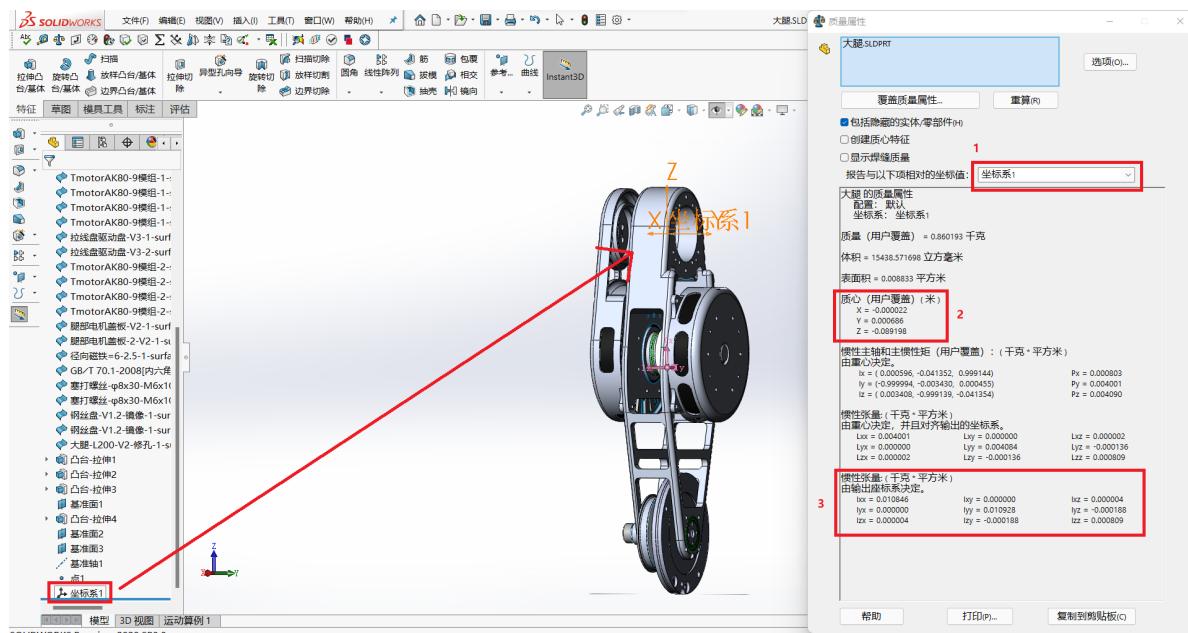
 在窗口的一角显示输出坐标系。

确定

取消

帮助

我们选择“使用自定义设定”---“小数位数”---“长度”---“质量”。小数位数选择6或者5，长度单位选择米，质量单位选择千克。



1、然后，建立坐标系，坐标系的建立和之前建立坐标系的方法和位置一样。建立完之后选择“坐标系1”。

2、确保质心单位:米

3、确保惯性张量单位:千克/平方米

依次获取完每个link的参数之后，分别配置webots中pyhsics的参数：mass、centerOfMass、inertiaMatrix。

inertiaMatrix的第一行对应惯性张量的[lxx,lxy,lzx]

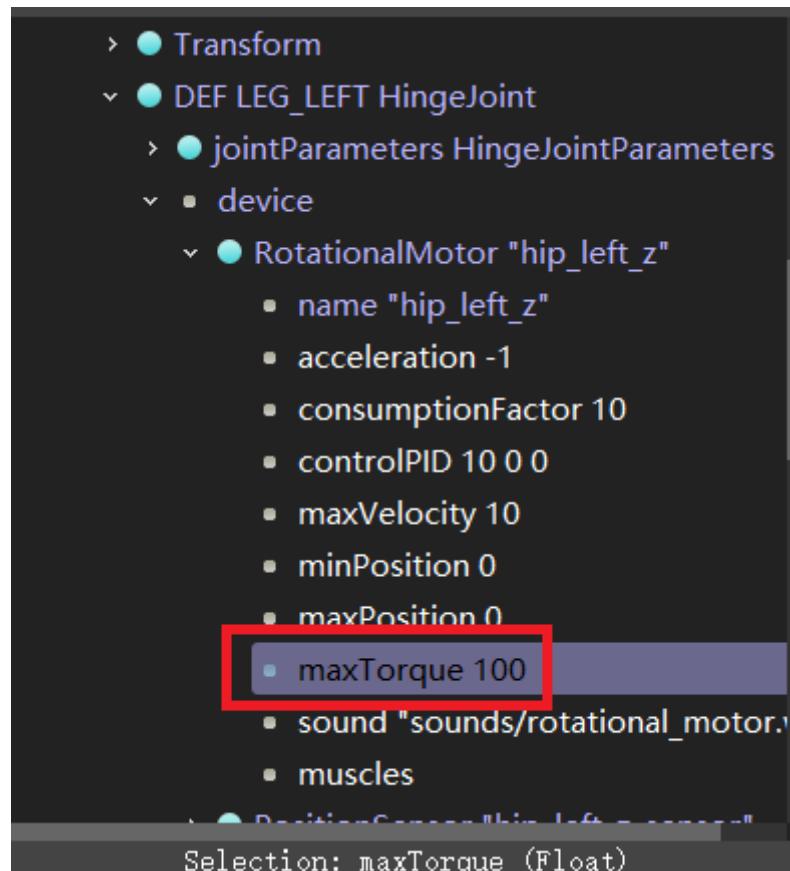
inertiaMatrix的第二行对应惯性张量的[lxy,lxz,lyz]

```

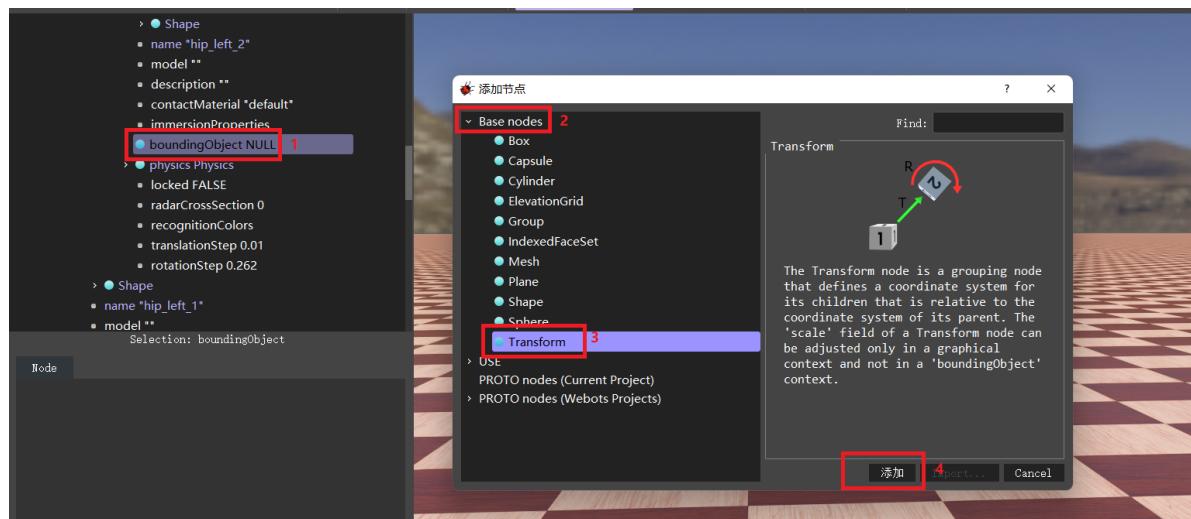
    • name "hip_left_1"
    • model ""
    • description ""
    • contactMaterial "default"
    • immersionProperties
    > ● boundingObject Transform
    > ● physics Physics
        • density -1
        • mass 1
        • centerOfMass
            -0.00104 0.0625 0.0146
        • inertiaMatrix
            0.00136 0.00202 0.00254
            1.8e-05 3.1e-05 0.000274
    • damping NULL
    • locked FALSE
Selection: WorldInfo

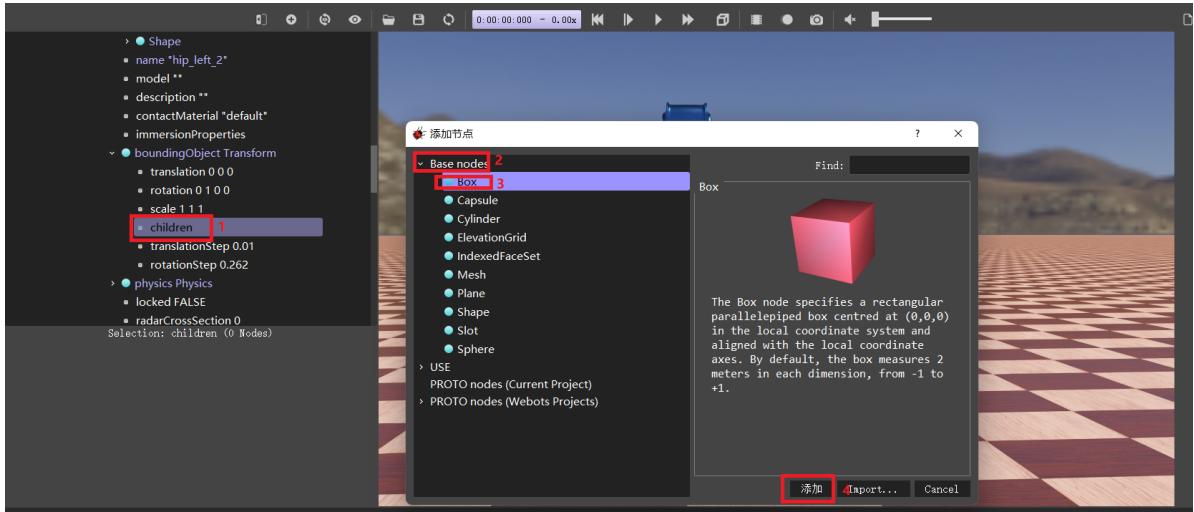
```

同时设置每个RotationalMotor的maxTorque为100



给每个link添加boundingObject，这里以hip为例添加Box





最后进行joints的关节正反转测试

测试要求：以左腿为例

hip_left_z 向外为正

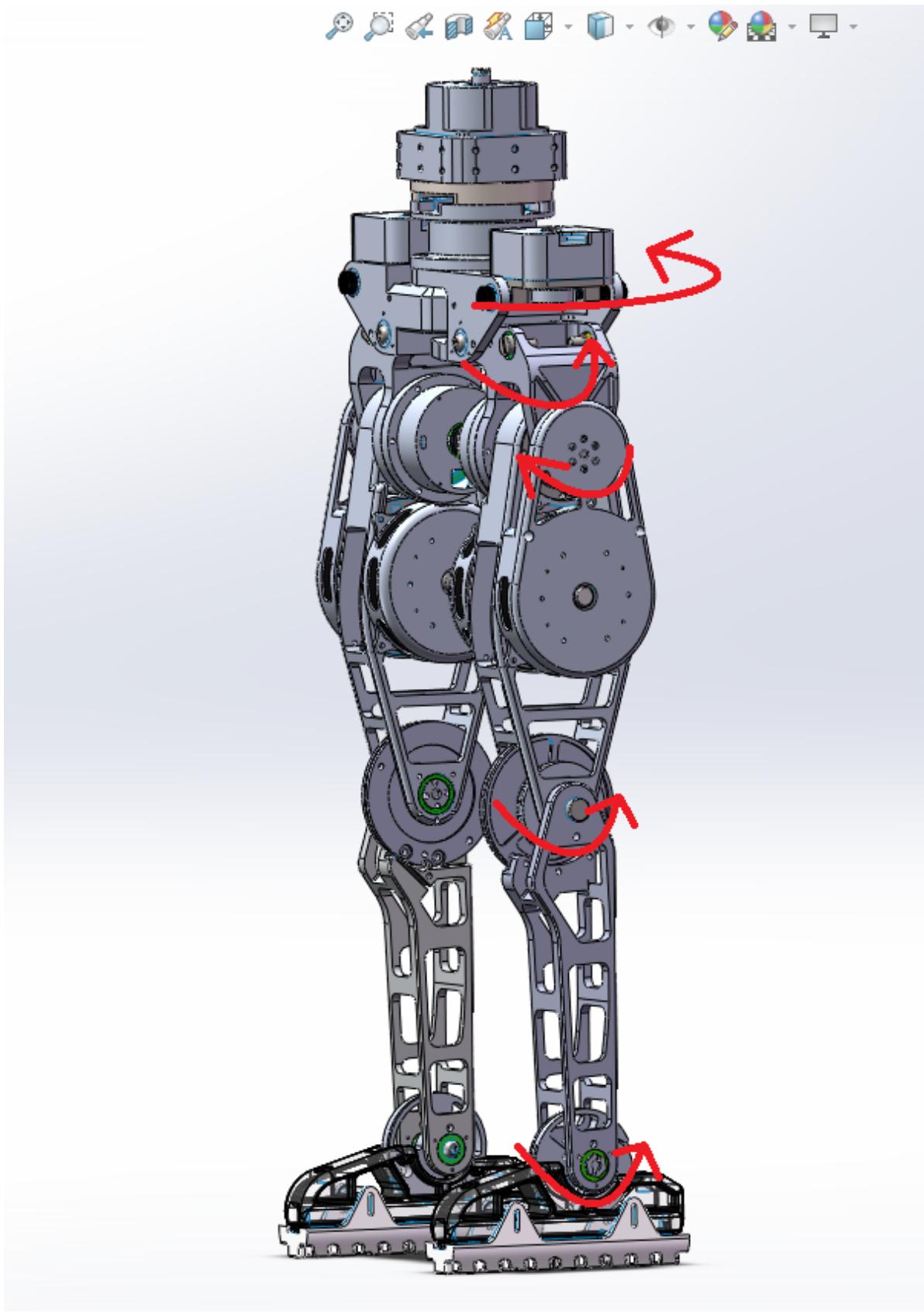
hip_left_x 向外为正

hip_left_y 向外为正

knee_left_y 向内为负

ankle_left_y 向内为正

如下图所示



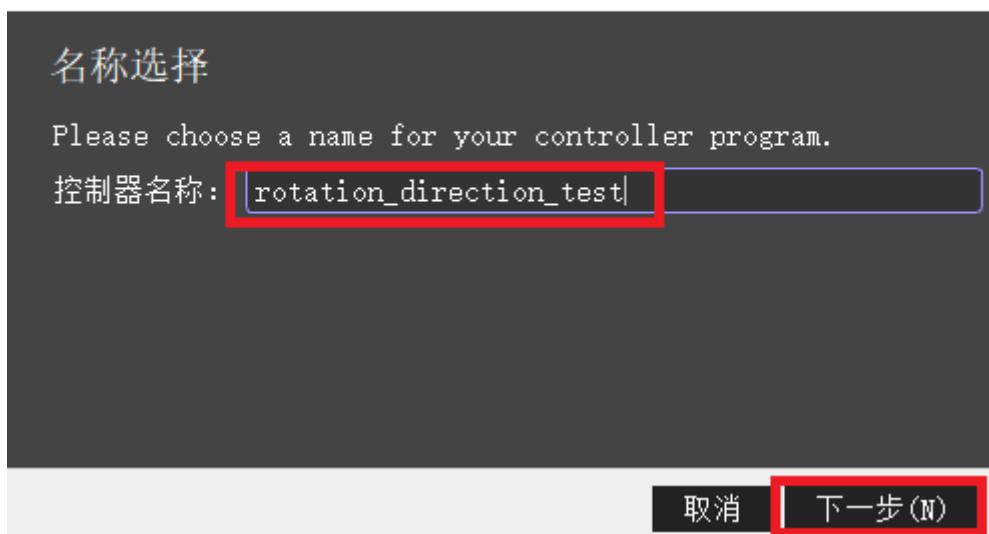
首先编辑控制器







← 创建新机器人控制器。



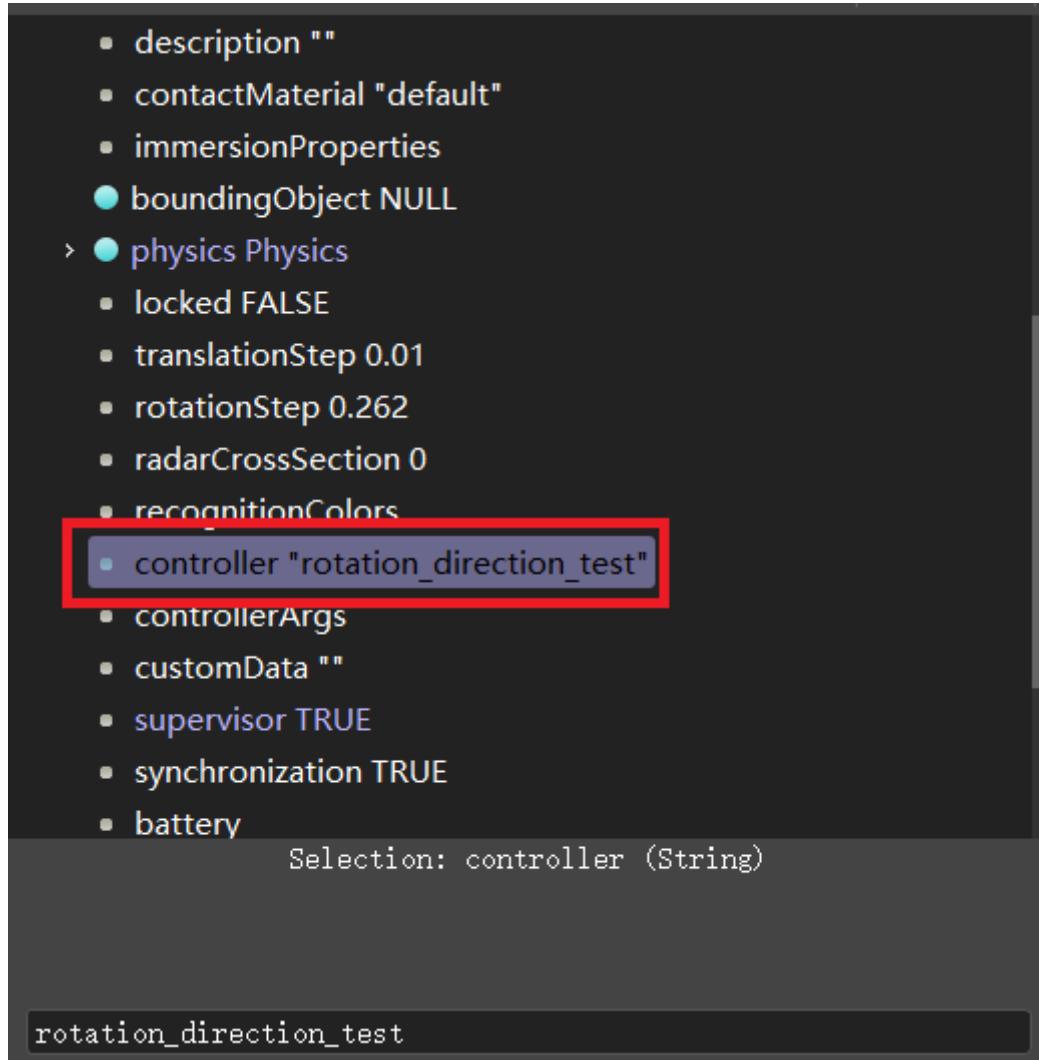
? X

← 创建新机器人控制器。



? X

选择自己创建的控制器



代码编写如下图所示

```

D:\webots_project\W02\worlds\W0223.wbt (W02) - Webots R2021a
文件(F) 编辑(E) 查看(V) 模拟(S) 生成(G) Overlay 工具(T) 向导(W) 帮助(H)
Simulation View 0:00-00:000 - 0.00x
rotation_direction_test.py
1 from controller import Robot
2
3
4
5 robot = Robot()
6 jointsmotor = ["hip_left_x", "hip_left_y", "knee_left_y", "ankle_left_y",
7                 "hip_right_x", "hip_right_y", "knee_right_y", "ankle_right_y"]
8
9 joint = robot.getMotor(jointsmotor[0])
10 joint.setPosition(1)
11 joint.setVelocity(5)
12 timestep = int(robot.getBasicTimeStep())
13
14 while robot.step(timestep) != -1:
15     pass
16

```

