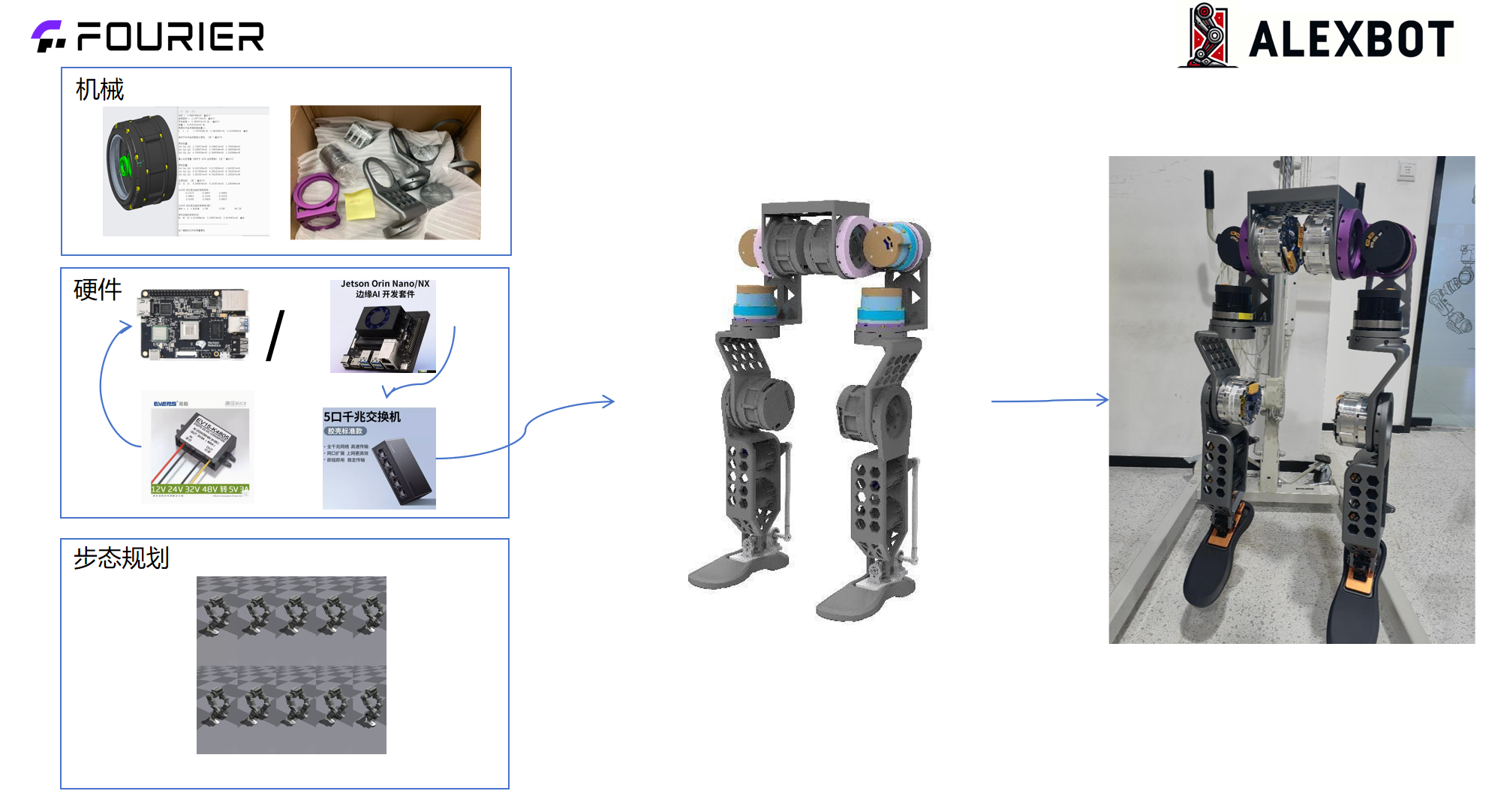
# 双足机器人简易全技能栈

# 暨fftai\_alexbot\_mini说明文档



fftai\_alexbot\_mini是一款拟打算全开源的双足机器人（包含机械，硬件，控制，算法），重点在于研究步态算法以及轻松部署步态降低RealityGap而设计的，整机fftai\_alexbot\_mini下半身长度700mm，上半身长度500mm，整体符合人体比例设计。

<https://github.com/Alexhuge1/fftai_alexbotmini>

**目录**

**[双足机器人简易全技能栈 1](#_Toc11152)**

**[暨fftai\_alexbot\_mini说明文档 1](#_Toc7336)**

**[1.本体整体设计 3](#_Toc19098)**

**[1.1机械设计 3](#_Toc26886)**

**[1.2硬件和驱动架构设计 6](#_Toc23011)**

**[1.2.1硬件架构设计 6](#_Toc6637)**

**[1.2.2电机Ethernet协议说明（用过就回不去了） 6](#_Toc27469)**

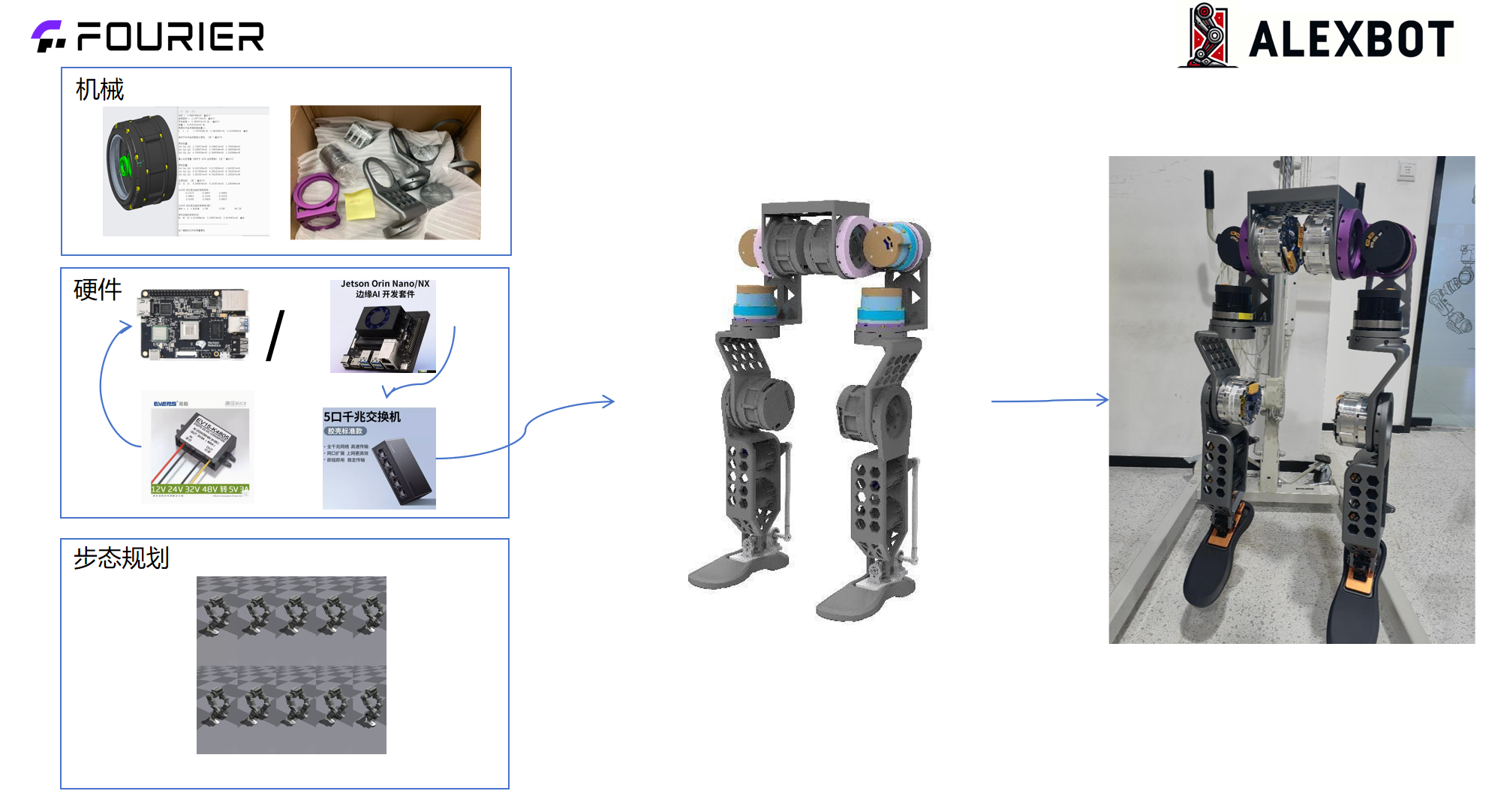
**[2.软件整体设计 8](#_Toc22689)**

**[2.1.0 URDF文件导出注意点 8](#_Toc11167)**

**[2.1.1步态规划设计 9](#_Toc29603)**

**[2.1.2 Simtosim迁移（From isaacgym to mujoco） 11](#_Toc24842)**

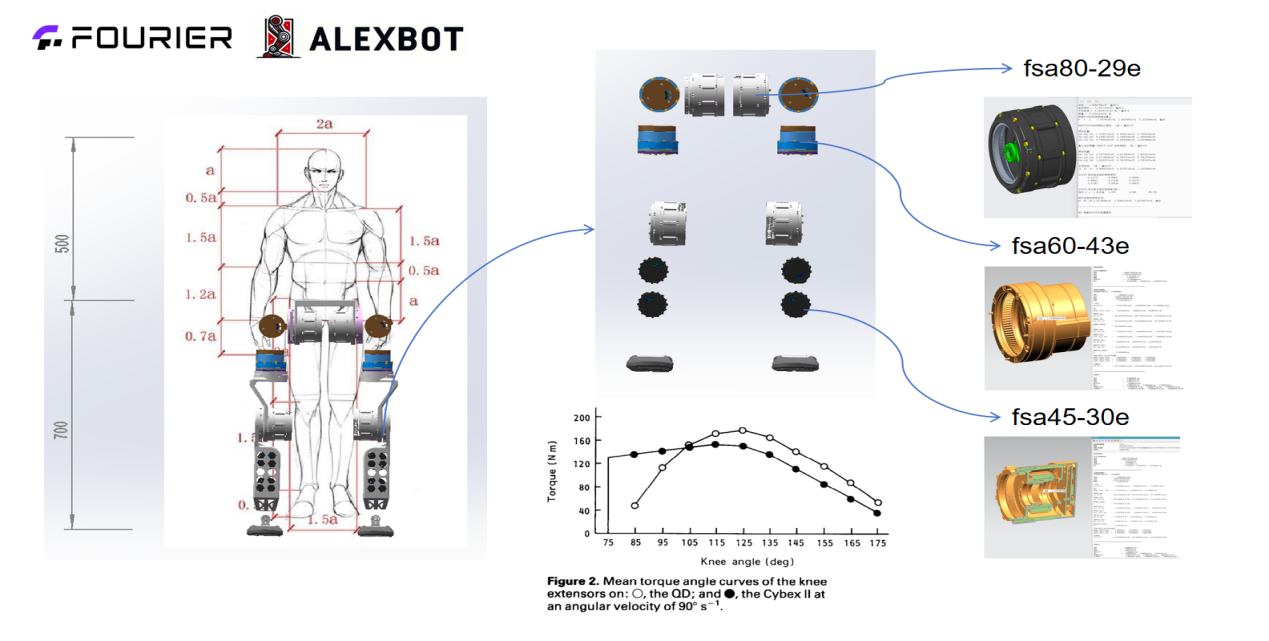
## 1.本体整体设计



## **1.1机械设计**

## （1）双腿结构

fftai\_alexbot\_mini双腿包含12个自由度全长700mm，预计重量<30kg，符合全身人体结构比。



根据人体的生物学结构以及生物力学知识，正常行走时髋关节的动作平衡且有节奏，耗能最低。双髋轮流负重，重心左右来回移动约4.0～4.5 cm。髋关节在步态周期过程中会有两个受力波峰，分别在足后跟着地及趾尖离地时。缓慢行走时，惯性力作用可不计，视与静力学相同。但髋关节在快速运动时，受加速和减速的作用，受力会增加。合力等于体重加惯性力，包括地面反冲力、重力、加速度、肌力等，一般认为是体重的3.9～6.0倍。在走路时(速度为1.5m/s)，髋关节最大受力约为2.5倍体重，而当跑步时(速度为3.5m/s)，关节最大受力约为5～6倍体重。

Stam等（见REF）研究了等速运动对膝伸肌力矩曲线的影响，运用CybexI等速测力仪对20名受试者进行主动和被动测试。发现主动模式下，关节角在119°时产生的平均力矩值为158Nm；被动模式下，关节角度在123°时的平均力矩值为179Nm，因此两种模式下的平均力矩有较大的差异。



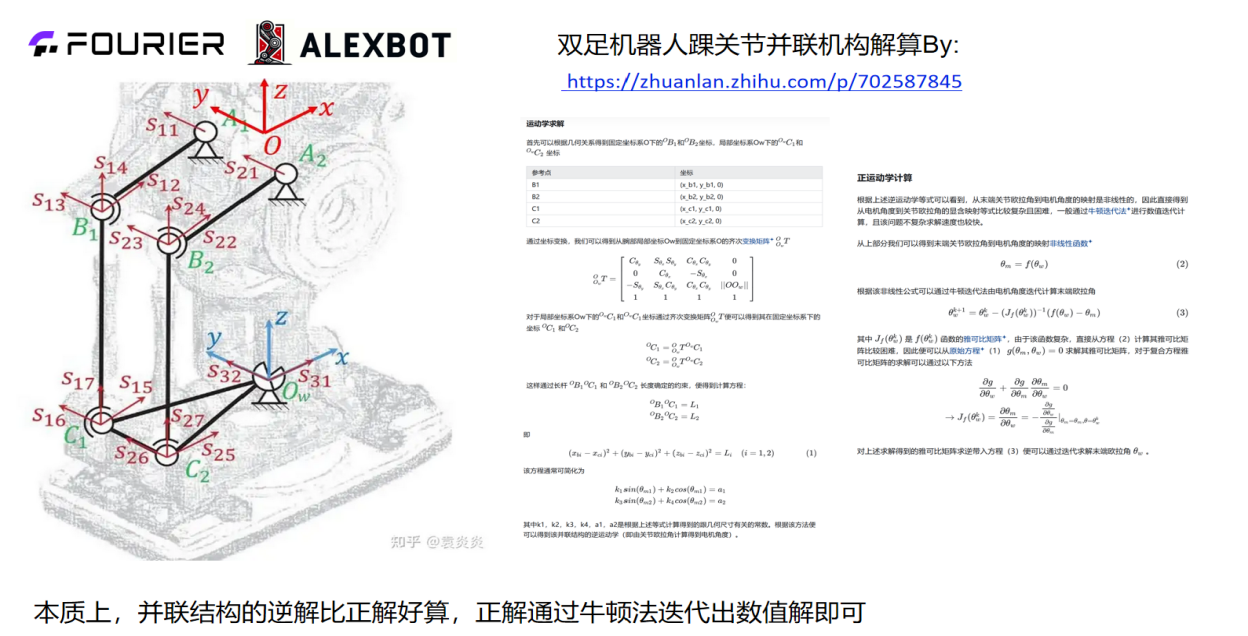
因此，对于本体重量约等于人类1/2，身高约等于人类的2/3的小型双足机器人扭矩要求选用如下：对于扭矩要求最高的胯部和膝关节部分电机选用FSA80-29E，对于大腿部分选用FSA60-43E，对于小腿，为了惯量的设计选用FSA45-30E以及并联结构，采用双电机朝向后方，通过板簧连接至脚底结构。

同时关于电机的串并联的选择，在电机扭矩十分金贵的时候，机器人时常采用并联结构从而降低整体惯量从而降低对于电机的扭矩的要求，并链结构不仅能放大力矩输出，还能通过连杆机构将电机上置，减少驱动器末端负载。

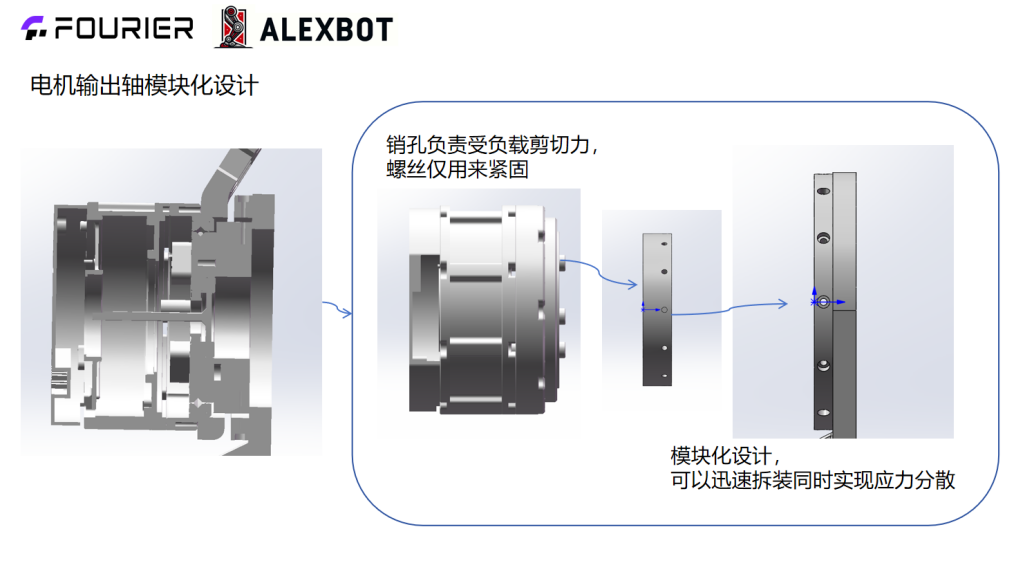
但是机器人结构采用并联往往会带来许多麻烦，首先机械上多了许多层级传动从而会增加机器人背隙，由于齿轮间隙或其他组件间隙造成的非预期运动或自由运动，过大的背隙会导致控制系统的精度下降，影响设备的性能和可靠性。

其次在机械导出URDF文件中会存在并联需要转移输出轴的做法，由于算法上位机解算的回传数据本质是位置环，所以会存在需要姿态逆解算的问题，而这样的并联机构一方面相比于单个平行四边形难以从几何学直接计算，另一方面其电机角度到空间欧拉角度的映射是非线性的，因此通常逆运动学比较好解（空间欧拉角到电机角度），而正运动学（电机角度到空间欧拉角）大多需要数值迭代求解。往往需要通过数值解去逼近。

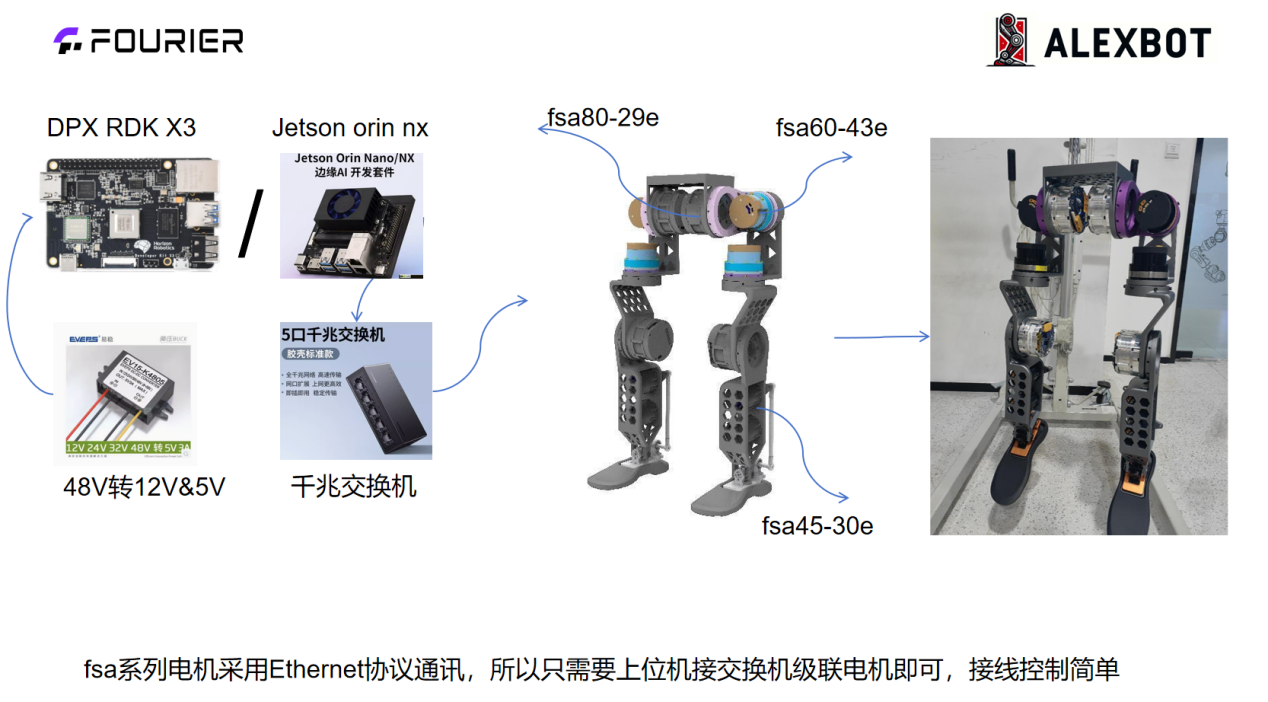
处理方法参考：<https://zhuanlan.zhihu.com/p/702587845>



同时，对于电机的输出侧使用了模块化快拆结构，方便拆卸和组装机器人，采用输出盘的类似设计，使用销结构与螺钉实现快拆，以膝关节为例:



## **1.2硬件和驱动架构设计**



## **1.2.1硬件架构设计**

## 硬件整体比较简单，分为信号层和功率层:

(1)信号层面上：主要包含上位机去推理神经网络，然后通过交换机来进行数据交换以实现一（上位机）拖多（电机）的情况，电机采用固定ip地址，上位机采用交换机进行数据交换，整体线路连接简单。

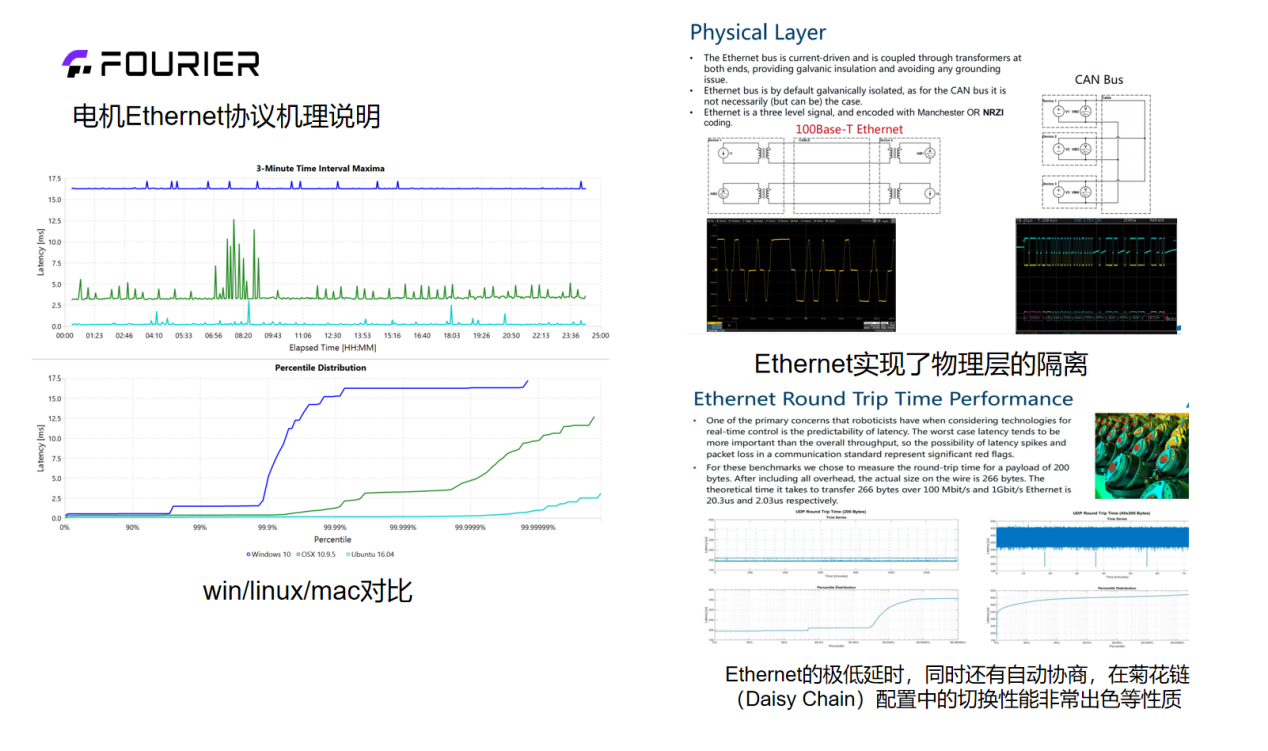
(2)功率层面上：实现硬件隔离，防止电机电流过大反冲烧坏上位机，同时电机采用48v供电，上位机采用12v供电，交换机采用5-60v宽幅输入

## **1.2.2电机Ethernet协议说明（用过就回不去了）**

得益于简易的电机控制框架，电机走的Ethernet协议实现了对于硬件层的封装，实现了简易的操作和快速上手的体验同时降低了电机的lateancy，如下方论文所述。

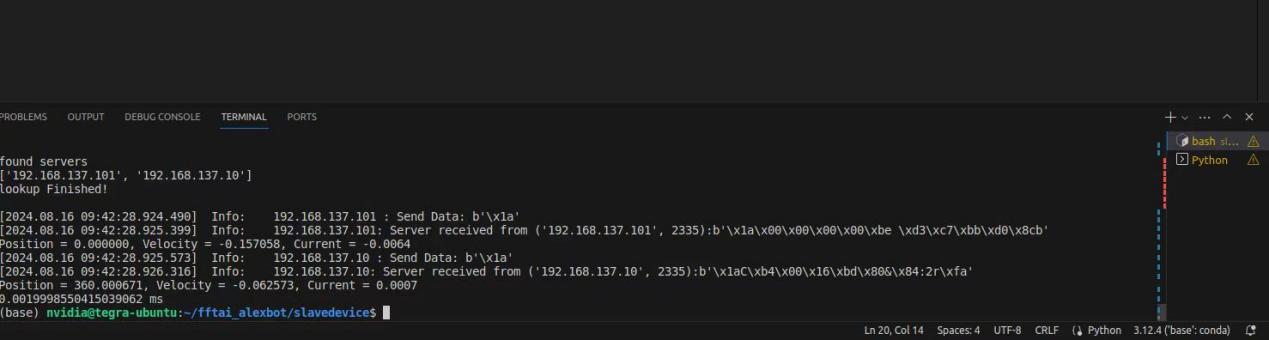
左侧图片都包含两个图表。上部显示在给定时间间隔窗口内发生的最严重故障，即第一个数据点显示前 3 分钟内发生的最严重故障，下一个数据点显示后 3 分钟内发生的最严重故障。下图显示了整个持续时间内所有测量值的百分位数。每个 24 小时数据集大约包含 7000-8000 万个样本。以太网还具有诸如低延时，多电机级联，冗余管理链路，无需下位机（现代机器人通常由嵌入式计算机控制以太网端口）。

与CAN总线的数据帧（只有8字节）相比，以太网可以处理更大的数据负载，无需在数据帧中节省字节。同时支持多关节机器人的通信拓扑、网络通信编程的简便性、OTA（Over-The-Air）升级、基于JSON的通信协议，方便阅读等等。总之是一个好的协议以及电机控制路线。

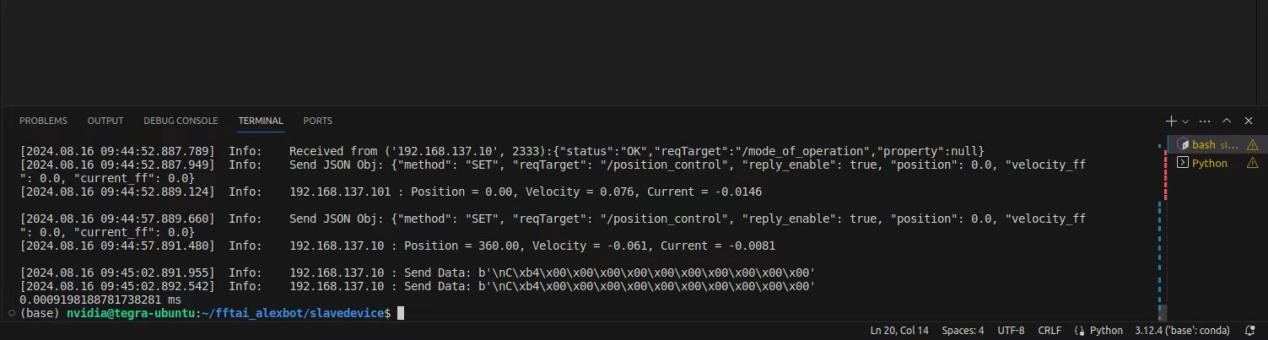


<https://ennerf.github.io/2016/09/20/A-Practical-Look-at-Latency-in-Robotics-The-Importance-of-Metrics-and-Operating-Systems.html>

框架主要是修改了FFTAI的电机sdk：https://gitee.com/FourierIntelligence/wiki-fsa，整体主要采用了 fi\_fsa\_v2.fast\_set\_position\_control()和fi\_fsa\_v2.fast\_get\_pvc();



fi\_fsa\_v2.fast\_set\_position\_control()的系统延时



fi\_fsa\_v2.fast\_get\_pvc()的系统延时

可以看到，整体的延时还是十分低的，对于RL等对控制频率不那么敏感的使用场景来说，我期望的控制频率在50hz左右，整体间隔2ms，这样的控制频率加上电机和imu还有模型推理的时间是足够的。

更进一步的框架和代码见2.2 Simtoreal的部分

## **2.软件整体设计**

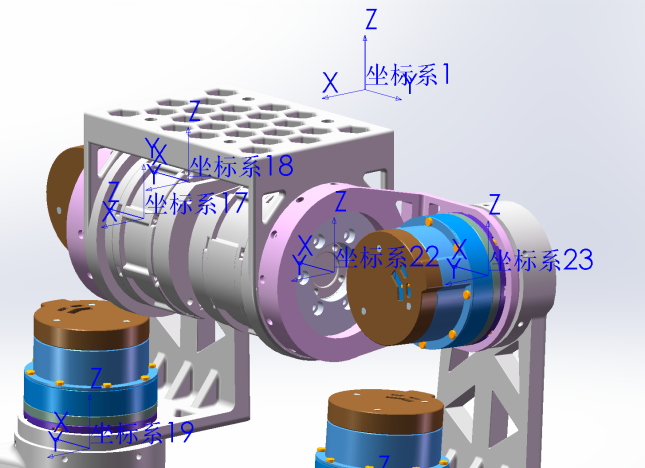
软件整体包含

1. 步态设计
2. Simtosim
3. 3.Simtoreal的部分

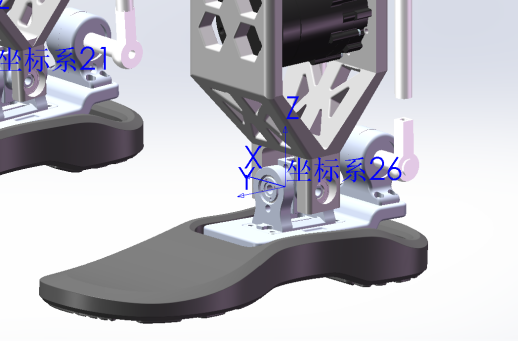
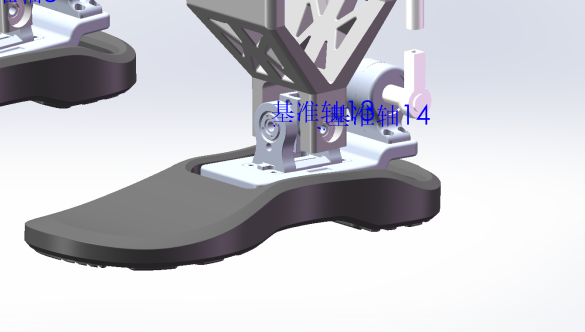
**软件设计框图**

**2.1.1 URDF文件导出注意点**

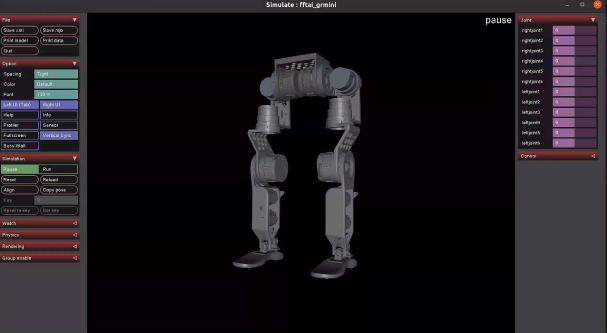
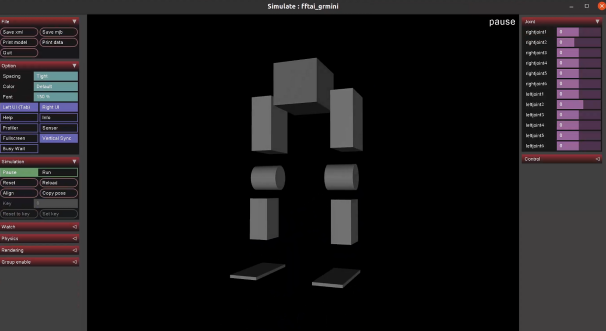
（1）注意导出的baselink方向，右手系以及x轴朝前，为前进方向。然后baselink的坐标系所在点为imu所在点！



（2）导出脚踝的urdf的时候采用虚拟轴设计，使用同一个坐标系，然后双轴，一个pitch，一个roll，这里导出的是leftlink5是小的中间的那个十字轴，leftlink6是脚底板。

（3）urdf在导入仿真环境的时候先使用mujoco的simulate检查一下，mujoco的物理引擎比较真实。如果模型文件的stl文件面数过多，使用meshlab简化一下到15000面即可；同时模型urdf文件的collision文件需要做简化，降低无用的以及自我的碰撞同时并不影响实际输出和simtoreal部署。

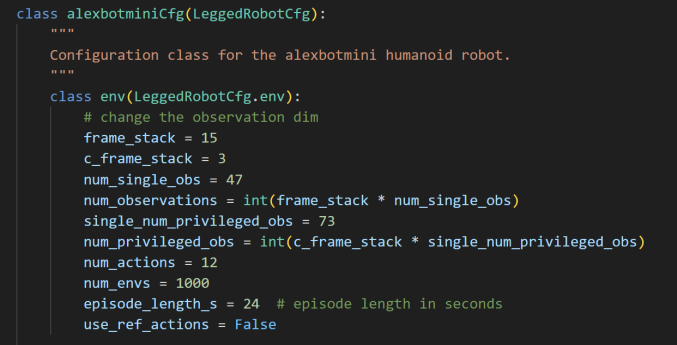
 

## **2.1.2步态规划设计**

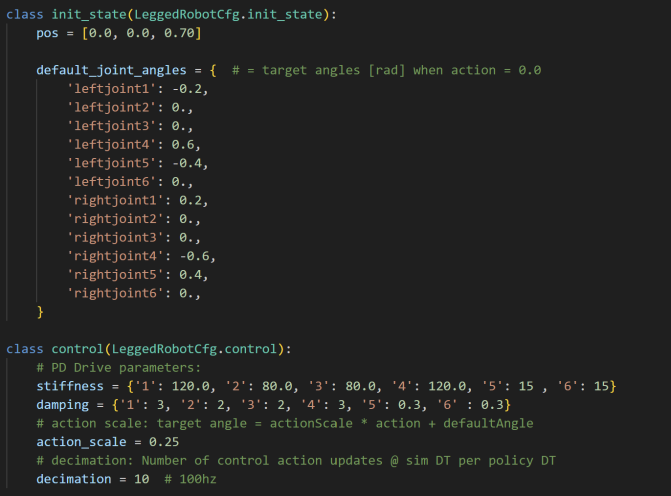
步态规划采用humanoidgym：https://github.com/roboterax/humanoid-gym，实现的步态效果如下（可以见ppt/github/知乎文档）：

其中主要的config设计大体上基于XBot的example，有几个比较重要的参数需要修改一下

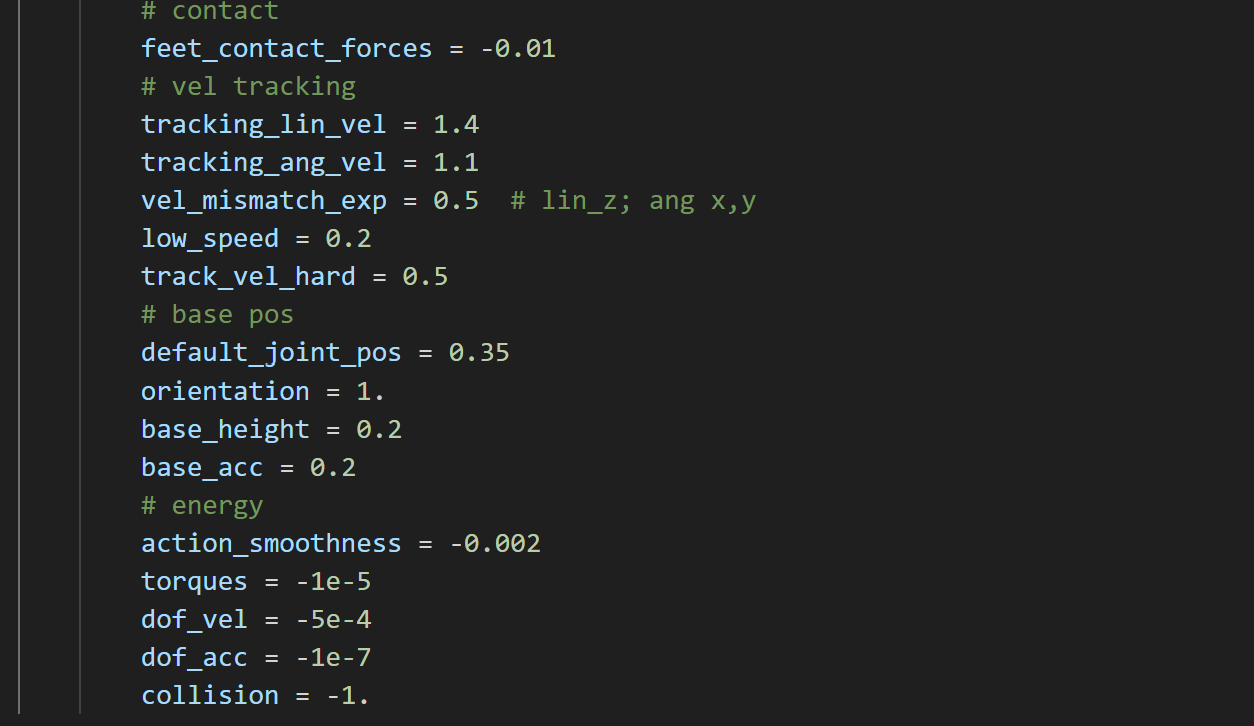
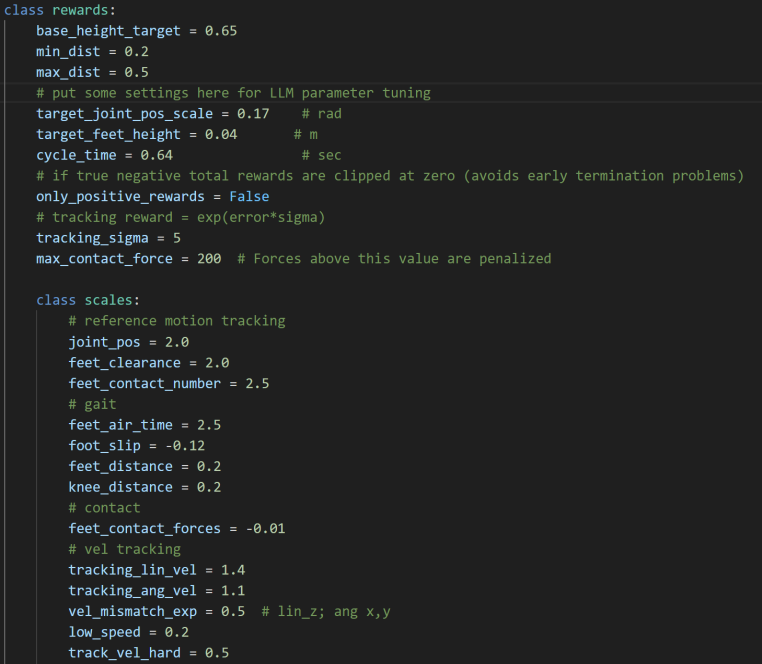
（1）这个是机器人的参数设定，包含了维数以及训练参数



（2）这个是机器人的初始化设定，尽可能使用微微下蹲作为初始状态



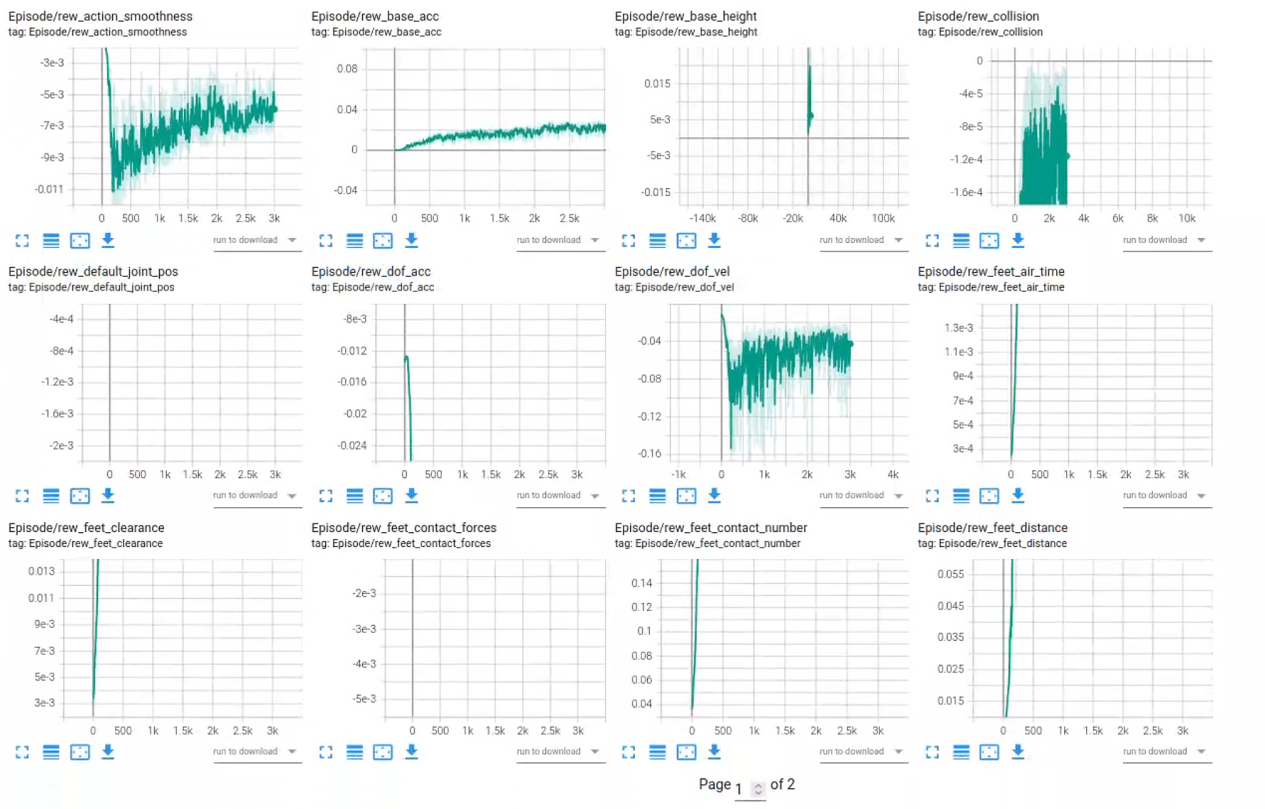
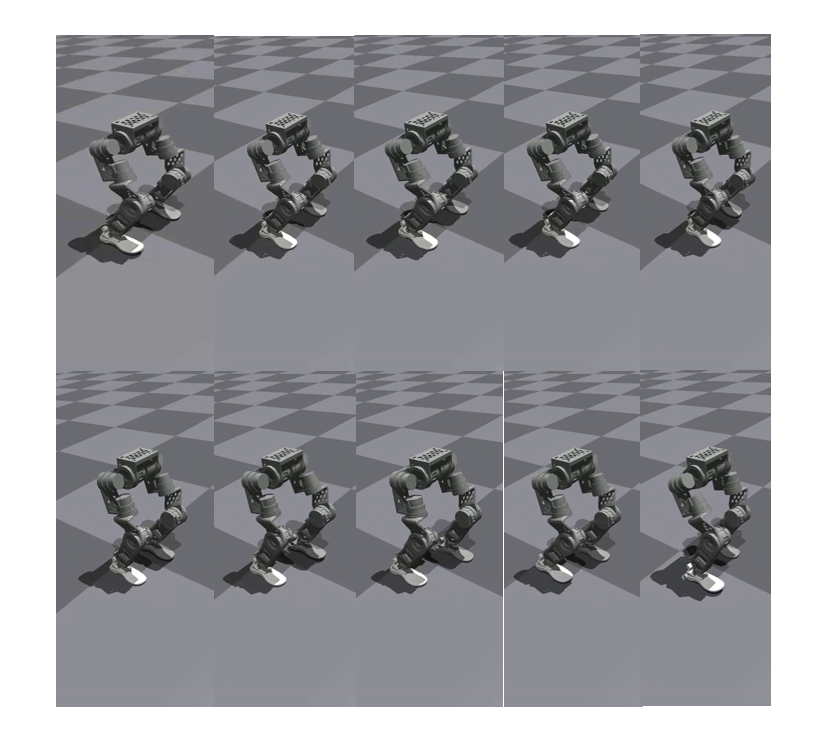
（3）这个是机器人的reward，细节reward代表什么可以看env的reward的class







## **2.1.2步态规划效果**



## **2.1.3 Simtosim迁移（From isaacgym to mujoco）**

对于模型参数文件（.pt）的迁移来说，模型的Reality gap极为重要，通过不同仿真软件的迁移可以起到模型验证以及降低gap的作用，其中isaacgym所使用的physicx是支持并行运算的，非常适合模型训练，然后在simtosim迁移到mujoco中，如下图可以看到mujoco与实际环境的gap相对isaacgym小（参考：[[2404.05695] Humanoid-Gym: Reinforcement Learning for Humanoid Robot with Zero-Shot Sim2Real Transfer (arxiv.org)](https://arxiv.org/abs/2404.05695)）

