上海理工大学机器智能研究院
ISTITUTE OF MACHINE INTELLIGENT, UNIVERSITY OF SHANGHAI FOR SCIENCE AND TECHNOLOGY

openloong运动控制框架实战经验

刘元基　　上海理工大学

2024.10.27

目录

模型适配

openloong-dyn-control框架中使用到的MPC和WBC算法均涉及到动力学和运动学，使用pinocchio库进行计算，因此，需要在模型的链式关系，坐标系，和pinocchio中操作空间关节命名方面进行适配，保证正确的运动学和动力学计算。

踝关节坐标系适配

- 与Azureloong模型中脚部坐标系方向一致

Link链式关系与关节顺序

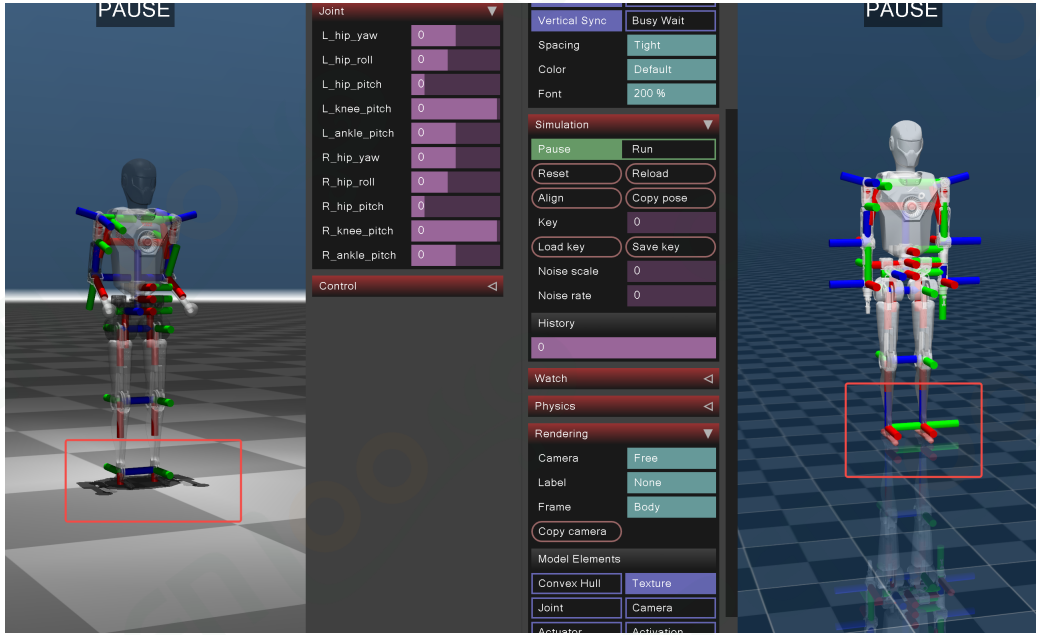
- 确保与AzureLoong中urdf链式关系一致
- 修改关节驱动名称与新模型保持一致

注：

- 即使机器人的关节电机方向不与AzureLonng完全一致，只要在urdf和底层（实物或者xml）定义好电机旋转方向，也能正常控制

踝关节坐标系适配

- 保证pinocchio计算输出的J_r, J_l, 和 fe_l_pos_cur_W数据正确，用于下游规划模块。



Link链式关系与关节顺序

- 统一urdf各link间的链式关系与AzureLoong相同

```
---------- Successfully Parsed XML ---------------
root Link: pelvis has 3 child(ren)
    child(1):  L_hip_yaw_Link
        child(1):  L_hip_roll_Link
            child(1):  L_hip_pitch_Link
                child(1):  L_knee_Link
                    child(1):  L_ankle_pitch_Link
    child(2):  R_hip_yaw_Link
        child(1):  R_hip_roll_Link
            child(1):  R_hip_pitch_Link
                child(1):  R_knee_Link
                    child(1):  R_ankle_pitch_Link
    child(3):  torso_link
        child(1):  L_shoulder_pitch_Link
            child(1):  L_shoulder_roll_Link
                child(1):  L_shoulder_yaw_Link
                    child(1):  L_elbow_Link
        child(2):  R_shoulder_pitch_Link
            child(1):  R_shoulder_roll_Link
                child(1):  R_shoulder_yaw_Link
                    child(1):  R_elbow_Link
```

```
---------- Successfully Parsed XML ---------------
root Link: base_link has 4 child(ren)
    child(1):  Link_arm_l_01
        child(1):  Link_arm_l_02
            child(1):  Link_arm_l_03
                child(1):  Link_arm_l_04
                    child(1):  Link_arm_l_05
                        child(1):  Link_arm_l_06
                            child(1):  Link_arm_l_07
    child(2):  Link_arm_r_01
        child(1):  Link_arm_r_02
            child(1):  Link_arm_r_03
                child(1):  Link_arm_r_04
                    child(1):  Link_arm_r_05
                        child(1):  Link_arm_r_06
                            child(1):  Link_arm_r_07
    child(3):  Link_head_yaw
        child(1):  Link_head_pitch
    child(4):  Link_waist_pitch
        child(1):  Link_waist_roll
            child(1):  Link_waist_yaw
                child(1):  Link_hip_l_roll
                    child(1):  Link_hip_l_yaw
                        child(1):  Link_hip_l_pitch
                            child(1):  Link_knee_l_pitch
                                child(1):  Link_ankle_l_pitch
                                    child(1):  Link_ankle_l_roll
                child(2):  Link_hip_r_roll
                    child(1):  Link_hip_r_yaw
                        child(1):  Link_hip_r_pitch
                            child(1):  Link_knee_r_pitch
                                child(1):  Link_ankle_r_pitch
                                    child(1):  Link_ankle_r_roll
```

Link链式关系与关节顺序

- 建立与AzureLoong一致的motorName容器

Azureloong DOF 数量： 31
X02 DOF 数量：18

- 由于自由度的差异，造成在KinWBC部分需要修改task 代码与模型进行适配

XML ▾ 取消自动换行 | 复制

```
{"J_arm_l_01","J_arm_l_02","J_arm_l_03", "J_arm_l_04", "J_arm_l_05",
 "J_arm_l_06","J_arm_l_07",
 "J_arm_r_01", "J_arm_r_02", "J_arm_r_03","J_arm_r_04","J_arm_r_05",
 "J_arm_r_06", "J_arm_r_07",
"J_head_yaw","J_head_pitch","J_waist_pitch","J_waist_roll", "J_waist_yaw",
"J_hip_l_roll", "J_hip_l_yaw", "J_hip_l_pitch", "J_knee_l_pitch","J_ankle_l_pitch",
"J_ankle_l_roll",
"J_hip_r_roll", "J_hip_r_yaw","J_hip_r_pitch", "J_knee_r_pitch","J_ankle_r_pitch",
"J_ankle_r_roll"};
```

```
{"L_hip_yaw", "L_hip_roll", "L_hip_pitch", "L_knee_pitch", "L_ankle_pitch",
           "R_hip_yaw", "R_hip_roll", "R_hip_pitch", "R_knee_pitch", "R_ankle_pitch",
           "L_shoulder_pitch", "L_shoulder_roll", "L_shoulder_yaw", "L_elbow",
           "R_shoulder_pitch", "R_shoulder_roll", "R_shoulder_yaw", "R_elbow"};
```

底层重构

- 将PVT控制器放置底层与上层程序分离，使用grpc进行通信，使用 request-response模式向底层实时发送期望位置、速度、前馈力矩、和PD参数

- 将Mujoco渲染和状态更新放置底层，加入状态估计算法提供上层程序所需状态输入

目的：与现有实物通信和控制保持一致，方便MPC-WBC算法的迁移和部署

WBC适配

```
///------------ walk --------------
std::vector<std::string> taskOrder_walk;
taskOrder_walk.emplace_back("RedundantJoints");
taskOrder_walk.emplace_back("static_Contact");
taskOrder_walk.emplace_back("SwingLeg");
taskOrder_walk.emplace_back("PosRot");
///-------- stand ------------
std::vector<std::string> taskOrder_stand;
taskOrder_stand.emplace_back("static_Contact");
taskOrder_stand.emplace_back("CoMXY_HipRPY");
taskOrder_stand.emplace_back("Pz");
```

```
///------------ walk --------------
std::vector<std::string> taskOrder_walk;
taskOrder_walk.emplace_back("RedundantJoints");
taskOrder_walk.emplace_back("static_Contact");
taskOrder_walk.emplace_back("SwingLeg");
taskOrder_walk.emplace_back("HandTrackJoints");
///-------- stand ------------
std::vector<std::string> taskOrder_stand;
taskOrder_stand.emplace_back("static_Contact");
taskOrder_stand.emplace_back("CoMXY_HipRPY");
taskOrder_stand.emplace_back("Pz");
taskOrder_stand.emplace_back("HandTrackJoints");
taskOrder_stand.emplace_back("HeadRP");
```

Stand任务适配

- 任务数量简化
- 去掉任务中有关waist关节的代码
- 根据low-damping分支修改kp-kd参数

Walking任务适配

- 任务数量简化
- 去掉任务中有关waist关节的代码
- 根据low-damping分支修改kp-kd参数，
- 修改SwingLeg任务和PosRot任务

相关问题：

- **issue 51：缺少腰关节和ankle roll关节对整体控制的影响**
- **issue 31：通过静力学验证pinocchio动力学数据结算是否正确以及DynWBC优化算法是否收敛**

调试心得

PVT参数整定

1. 禁用MPC和WBC，仅使用PVT控制机器人站立；
2. 整定各关节kp、kd、和力矩上下限参数

PVT+WBC站立测试

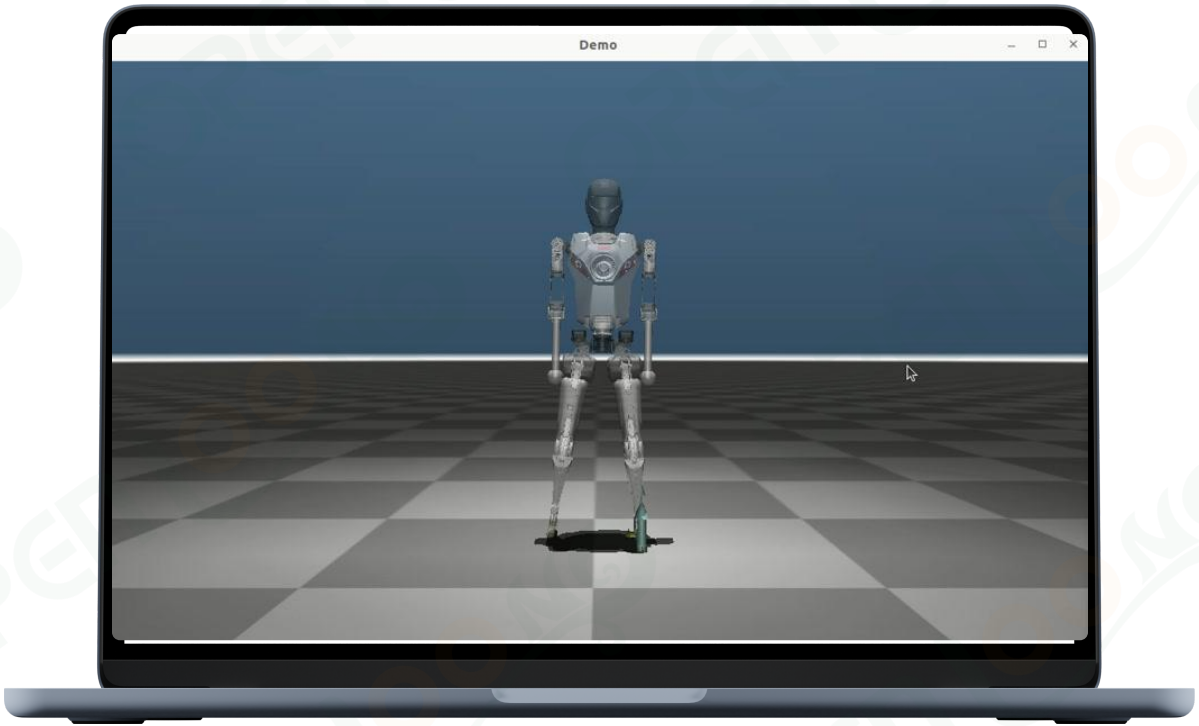
1. 设置状态机一直保持stand状态，禁用mpc，介入WBC；
2. 根据介入WBC后关节力矩曲线的变化情况整定部分关节的PVT参数和KinWBC-Stand各任务的kp-kd参数；

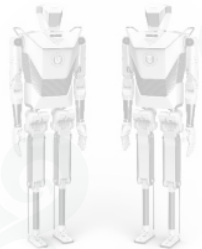
PVT+WBC行走测试

1. 设置状态机正常进入walking，设置期望速度xv_des=0，让机器人步态规划处于原地踏步；
2. 根据左右脚步态规划曲线，整定步态规划模块的kp、kd、和偏移量等参数；
3. 整定KinWBC-Walking各任务的kp-kd参数；
4. 给定期望速度，重复上述整定过程；

PVT+MPC+WBC行走测试

1. 根据不同的机器人设置不同的质量、约束上下限和惯性张量参数；
2. 根据数据和机器人行走现象整定MPC中的权重矩阵；


Demo



感谢聆听