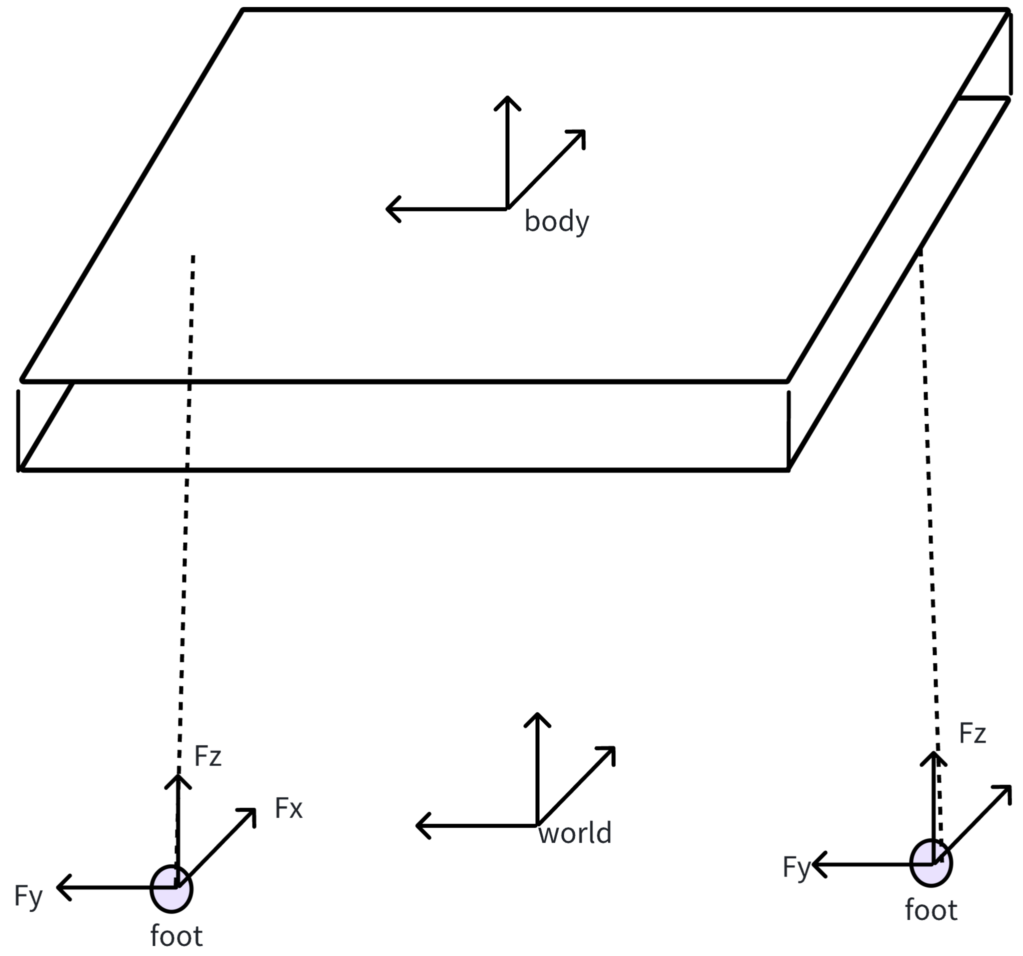
基于单刚体模型的离线优化跳跃

0.单刚体模型的示意图



1.描述单刚体模型六自由度与受力关系的数学公式

假设：将**模型的腿部质量集中于身体两侧**，整体简化成一个单刚体

1.1该模型的动力学模型描述如下，

Eq(1) $$p(\ddot{k}) = \sum_{i=1}^{2} f(k)_i/m + g$$

Eq(2) $$\frac{\mathrm{d}}{\mathrm{d}t}(I_w\omega(k)) = I_w\dot{\omega}(k) + \omega(k-1)\ \times I_w\omega(k-1) = \sum_{i=1}^{2} r(k)_i \times f(k)_i$$

$p(\ddot{k})$： 世界坐标系下xyz方向的加速度，

$f(k)_i$： 世界坐标系下两条支撑腿的支撑力（ground reaction force）

$\dot{\omega}(k)$:世界坐标系下角加速度

$\omega(k-1)$:世界坐标系下角速度

$r(k)_i$:世界坐标系下质心位置到足端的位置矢量

1.2 加速度前向欧拉积分得到速度

Eq(3) $\dot{p}(k) \ = \dot{p}\ (k-1)\ +\ dt * \ddot{p}\ (k)$

Eq(4) $\omega(k)\ = \omega\ (k-1)\ +\ dt * \dot{\omega}(k)$

Eq(5) $\dot{\alpha}(k) = \begin{bmatrix} \dot{\phi}(k) \\ \dot{\theta}(k) \\ \dot{\psi}(k) \end{bmatrix} = \begin{bmatrix} cos(\psi(k))/cos(\theta(k)) & sin(\psi(k))/cos(\theta(k)) & 0 \\ -sin(\psi(k)) & cos(\psi(k)) & 0 \\ cos(\psi(k))tan(\theta(k)) & sin(\psi(k))tan(\theta(k)) & 1 \end{bmatrix} \omega(k)$

采用前向欧拉积分，可以得到下一时刻的质心速度，角速度。欧拉角的微分由欧拉角的角速度共同决定

$p(\dot{k})$:质心速度

$\omega(k)$:角速度

$\dot{\alpha}(k)$:欧拉角的微分，

欧拉角 $\alpha(k) = \begin{bmatrix} \phi(k) \\ \theta(k) \\ \psi(k) \end{bmatrix}$

1.3 速度前向欧拉积分得到位置

采用前向欧拉积分，可以得到下一时刻的质心位置和欧拉角

Eq(6) $p(k)\ = p(k-1)\ +\ dt * \dot{p}\ (k)$

Eq(7) $\alpha(k) = \alpha(k-1) + dt * \dot{\alpha}(k)$

1.4结论

上述7个公式描述了单刚体模型六个自由度在每个时刻随受力和时间的变化关系。

2.单刚体模型的离线轨迹优化

2.0基本原理

四足机器人的mpc和wbc部分均用到了**带有 线性 约束**的二次规划问题的求解：

一个典型的带有线性约束二次规划问题：

$$\begin{array}{ll} \underset{U}{min} & \frac{1}{2}U^T HU + U^T g \\ s.t. & c_l \leq CU \leq c_u \end{array}$$

其中U表示决策变量，CU表示对于决策变量的线性约束，

在单刚体模型的离线优化中我们需要用到**带有 非线性 约束**的二次规划问题,该问题一般采用casadi库进行求解

$$\begin{array}{ll} \underset{U}{min} & \frac{1}{2}U^T HU + U^T g \\ s.t. & c_l \leq C(U) \leq c_u \end{array}$$

2.1目标方程

目标方程中的决策变量为24N*1维矢量

$$[\ddot{X}(1), \dot{X}(1), X(1), F(1), ..., \ddot{X}(k), \dot{X}(k), X(k), F(k), ..., \ddot{X}(N), \dot{X}(N), X(N), F(N)]$$

$$X(k) = \begin{bmatrix} x(k) \\ y(k) \\ z(k) \\ \phi(k) \\ \theta(k) \\ \psi(k) \end{bmatrix} \quad X_d(k) = \begin{bmatrix} x_d(k) \\ y_d(k) \\ z_d(k) \\ \phi_d(k) \\ \theta_d(k) \\ \psi_d(k) \end{bmatrix}$$

$$F(k) = \begin{bmatrix} F_{lx}(k) \\ F_{ly}(k) \\ F_{lz}(k) \\ F_{rx}(k) \\ F_{ry}(k) \\ F_{rz}(k) \end{bmatrix}$$

目标方程如下所示。

$$\sum_{k=1}^{N}(X_d(k) - X(k))^T W_x(k)(X_d(k) - X(k)) + \sum_{k=1}^{N}(\dot{X}_d(k) - \dot{X}(k))^T W_{dx}(k)(\dot{X}_d(k) - \dot{X}(k)) + \sum_{k=1}^{N}(\ddot{X}_d(k) - \ddot{X}(k))^T W_{ddx}(k)(\ddot{X}_d(k) - \ddot{X}(k)) + \sum_{k=1}^{N} F(k)^T W_f(k)F(k)$$

将1中介绍的eq(1)~eq(7)稍加修改，即可将动力学过程转换成每个时刻的约束，其中有线性的也有非线性的，

如果没有旋转运动可以忽略掉姿态角相关的变量

3使用方法

3.1代码仓库

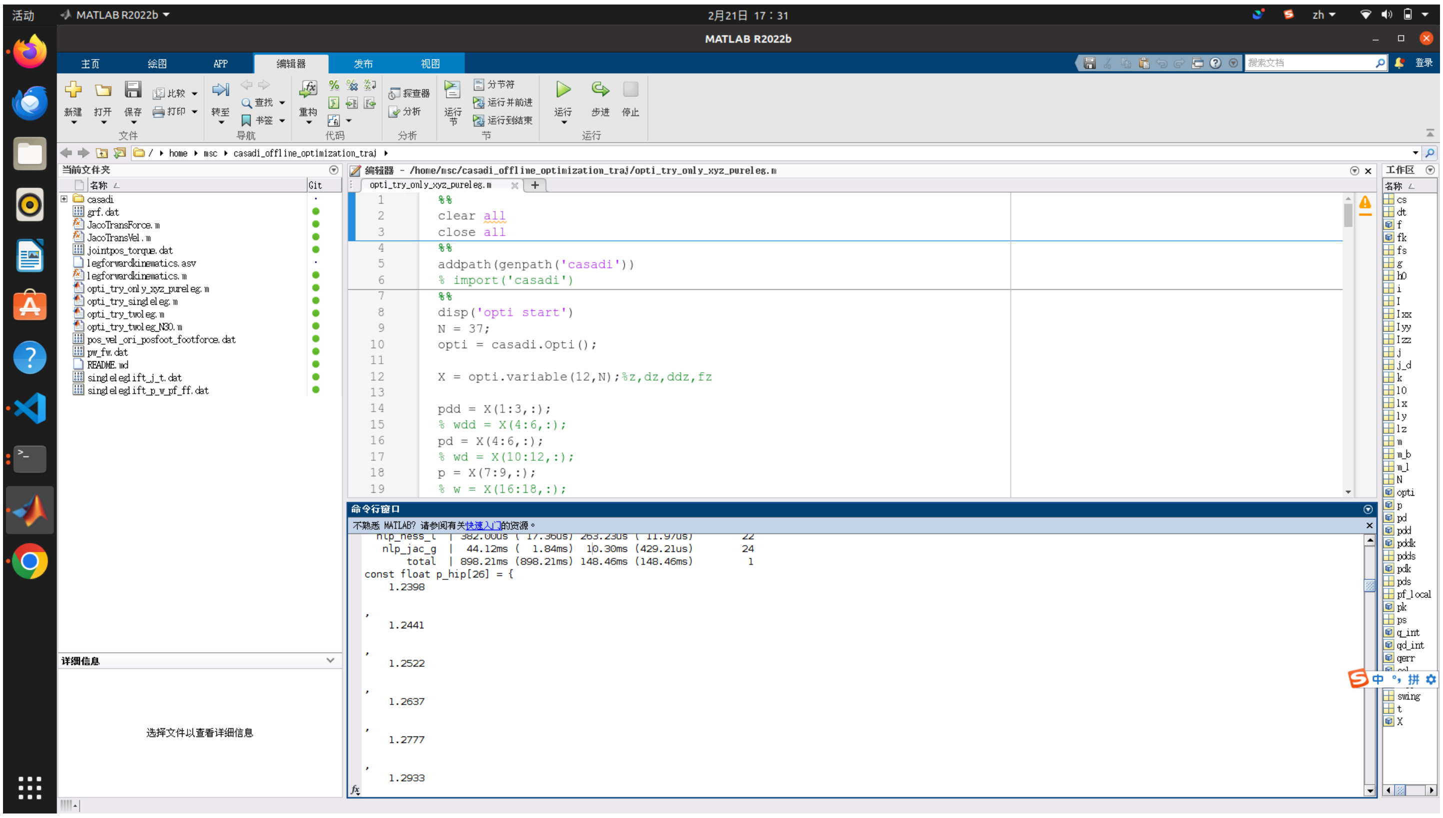
离线优化的代码仓库

https://git.ddt.dev:9281/rbt/alg/casadi_offline_optimization_traj https://git.ddt.dev:9281/rbt/alg/sim-webots/-/tree/dev_offline_opti_front_jump_old?ref_type=heads

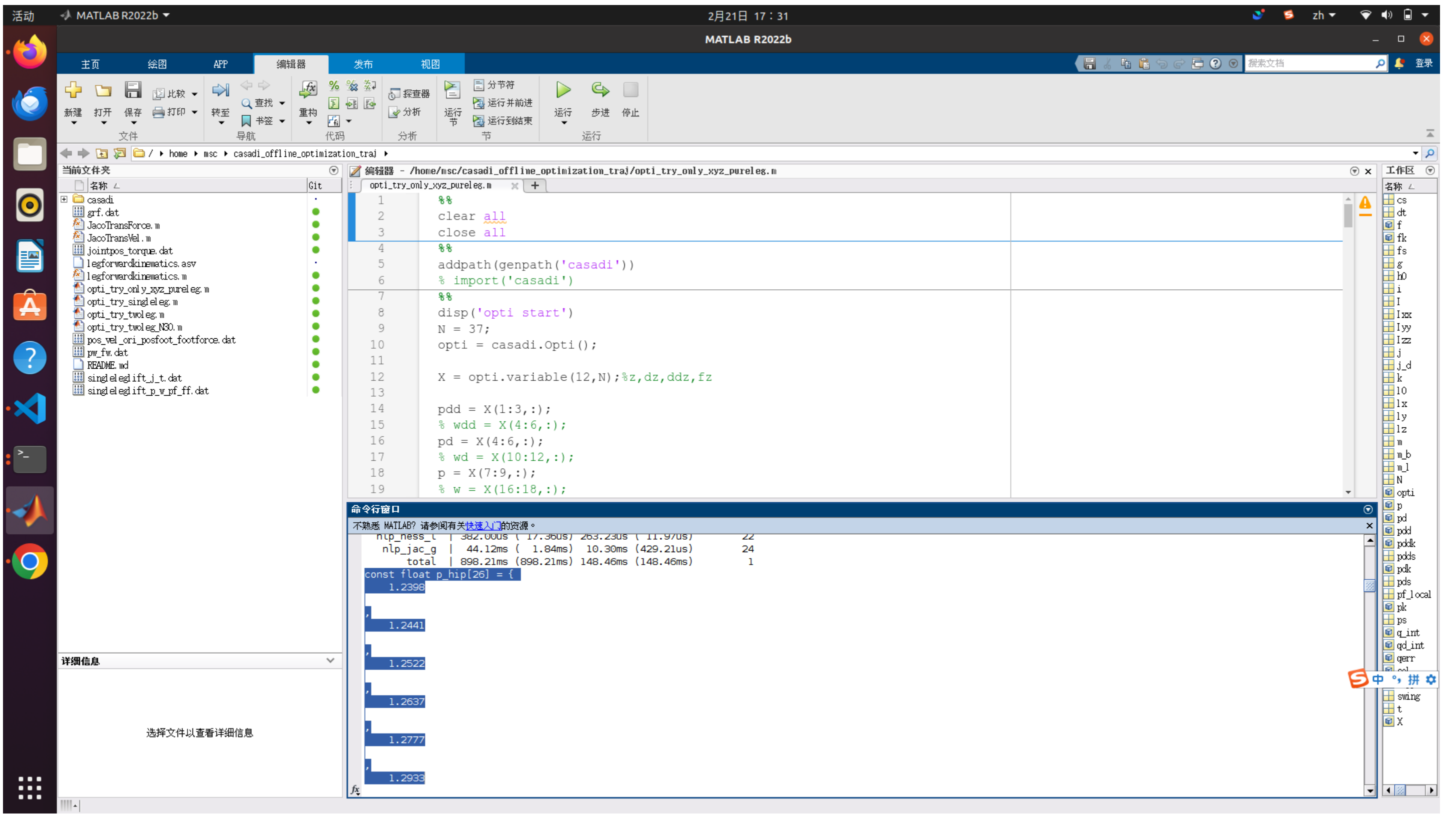
1.用matlab打开 你的路径/casadi_offline_optimization_traj

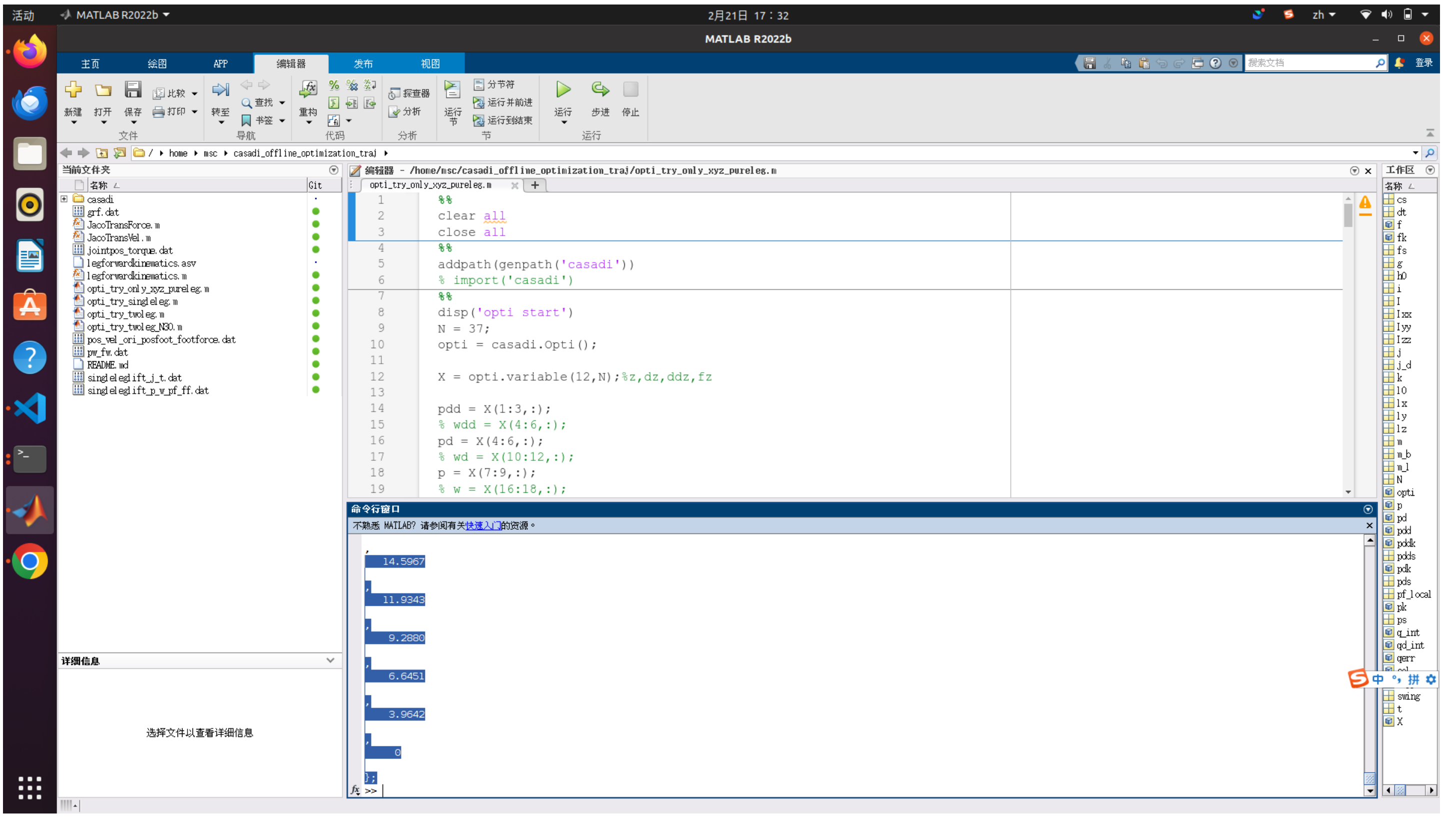
2.打开文件casadi_offline_optimization_traj/opti_try_only_xyz_pureleg.m

3.点击运行



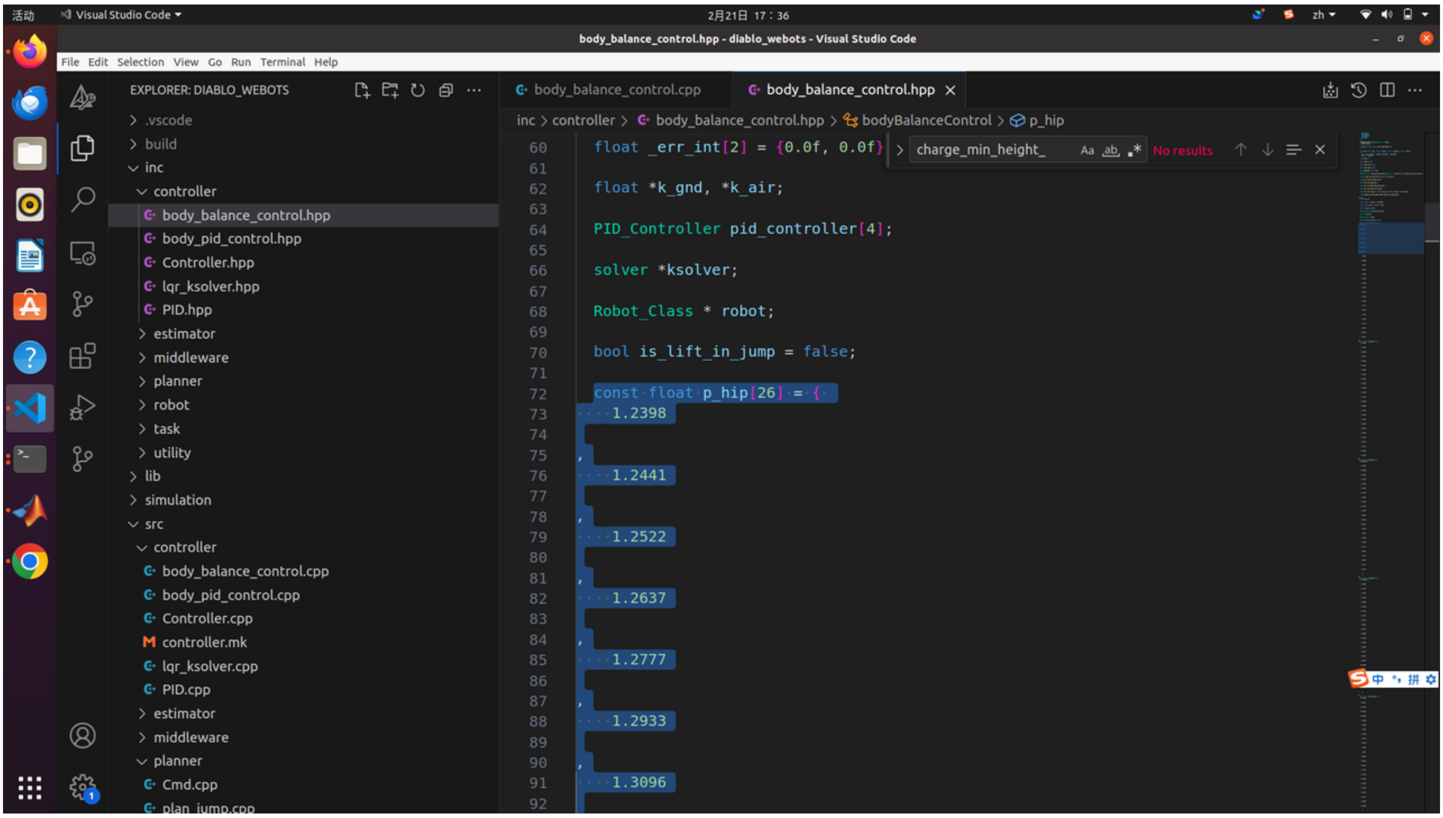
4.复制生成的期望轨迹



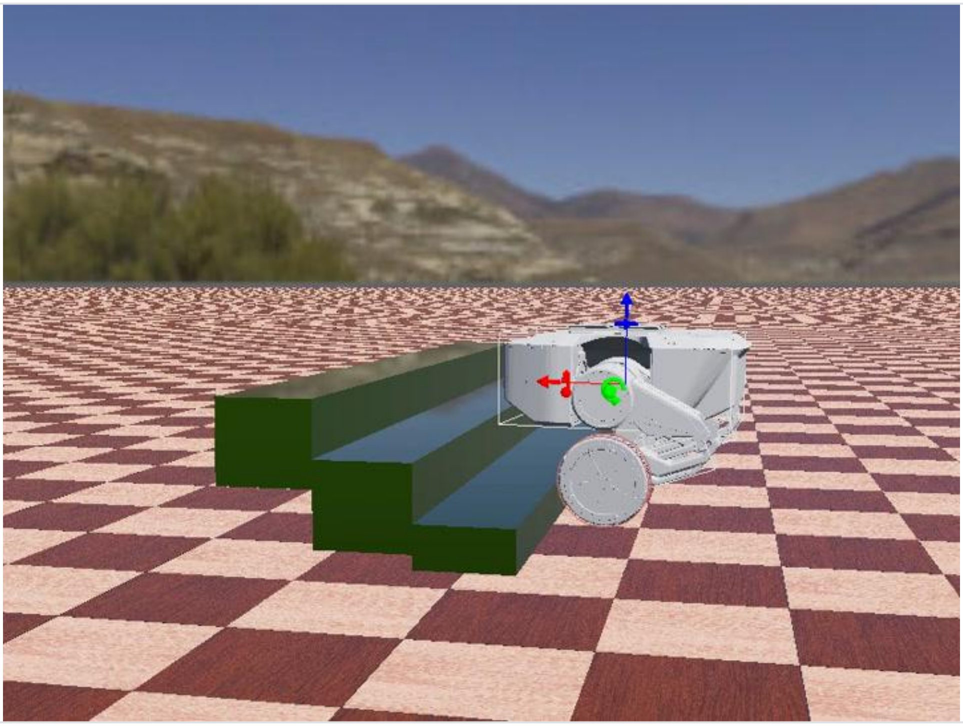


5. 进入sim-webots,切换到dev_offline_opti_front_jump_old分支

6.将matlab命令行复制的内容copy到图下地方



7. 老方法make debug编译,，正常跑仿真，高度设定16cm到20cm之间，摁空格即为立定跳跃



8. 进入dev_offline_opti_front_jump_old，执行git show 0cd5991,可以看到修改跳跃轨迹的过程