

A1中split和tilt的处理方法

Planner也是一种控制器 —— by 高飞

问题描述

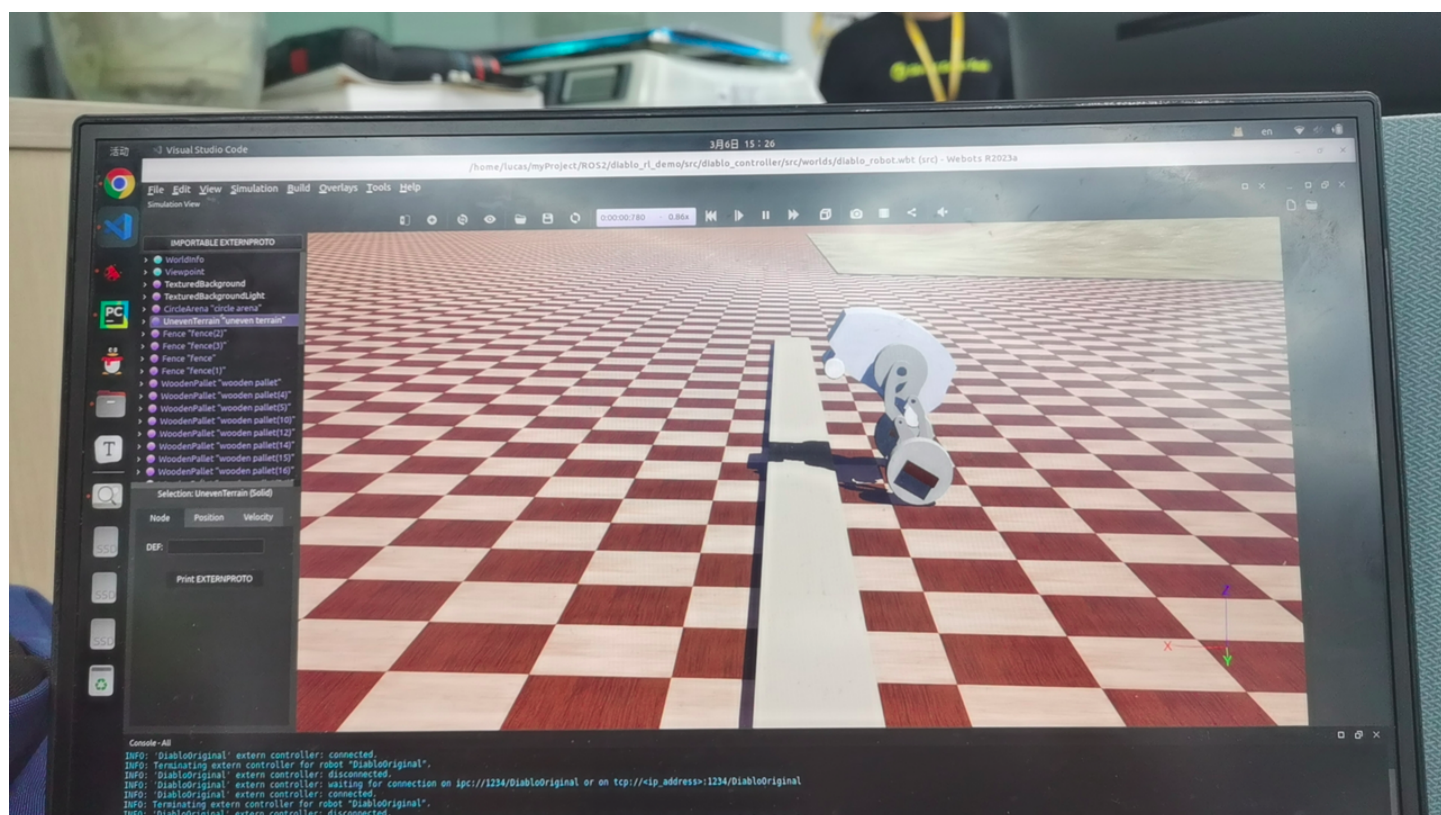
在我们当前的控制器中，采用的是LQR，其本质是一个并行控制器，即多路反馈控制器直接进行叠加。这就可能会出现控制器中出现多种稳态的情况，而这多种稳态背后涉及到系统的极点配置，需要借助根轨迹等手段进行分析。

这里面一个典型的案例就是A1的split控制器和tilt控制器的处理。hip关节电机力矩的输出为：

$$\tau_{hip} = \tau_{split} + \tau_{tilt}$$

其中， τ_{split} 是A1的劈叉角度决定的， τ_{tilt} 是由头部的俯仰角head_pitch决定的。于是我们便发现这是个欠驱动的问题：2个hip关节电机要控制3个状态量，即左腿的split，右腿的split，还有head_pitch，是欠驱系统，最终达到的稳态效果可能并不是我们想要的。

比如，根据实习生反馈，在做强化学习的时候，明明set_pitch一直是0，但莫名其妙地出现了脑袋抬头的情况。



查看控制器会发现，此时

$$\tau_{split} \approx -\tau_{tilt} \neq 0$$

那么

$$\tau_{hip} \approx 0$$

头部控制卡在这个状态出不来了，但确实也达到了稳态。造成这个问题的原因是因为两者的优先级在不同状况下不一样，但LQR中没有进行区分。（Q算是做了下区分，但不全面）

思路盲区

1. 直接加积分解决

split和tilt各自理论上都应该带积分，来保证对应位置的紧缩，因此没办法从根上把握会不会进入到对应的稳态区域。

2. 增加或降低其中一方的增益或者权重

可以解决，但有可能牺牲其他方面的性能。

比如，降低了split的权重，可能导致单脚卡腿的时候机器人没锁好split，进而摔跤。现在A1上在用这种方法，但不太好。

3. 层级控制。但是层级控制好像一般是针对冗余系统，欠驱系统我不会。。。

4. Q根据不同状况自动调整，这个应该也可以，但感觉很麻烦。

解决

引入planner。一种做法是通过split_err，对split的力矩输出进行通道规划，以下是代码：

```
1     float max_split = mabs(split_err * 100.f);
2     bound(max_split,30.f);
3     bound(outL,5.f + max_split);
4     bound(outR,5.f + max_split);
5
6     left_out.split_out = outL;
7     right_out.split_out = outR;
8
```

做了个超级简单的线性规划。其大致思路就是split的输出通道根据split_err的大小会发生相应的变化。这样，在split_err不大时，保证了tilt优先级；在split_err较大时，又保证了split的控制优先级。