质心辨识

Diablo 的质心辨识

基于A1机器人进行参数辨识，辨识的参数为机器人的质心位置 ${}^RP_{com}$ ，其中右上角R代表为机器人的坐标系，com为质心，$$P$$代表绝对位置，

通过仿真可以近似获得机器人直驱电机下关节的力矩输出为：

$$^R\tau_i = {}^Rr_{comi} \times {}^R F$$

而 ${}^Rr_{comi} = {}^R p_{com} - {}^R p_i$ ，其中 ${}^Rp_i$ 为关节位置，例如hip关节，knee关节，wheel关节

而 ${}^Rp_i$ 可以由正运动学算法得到，即： ${}^Rp_i = FK(q)$ 。

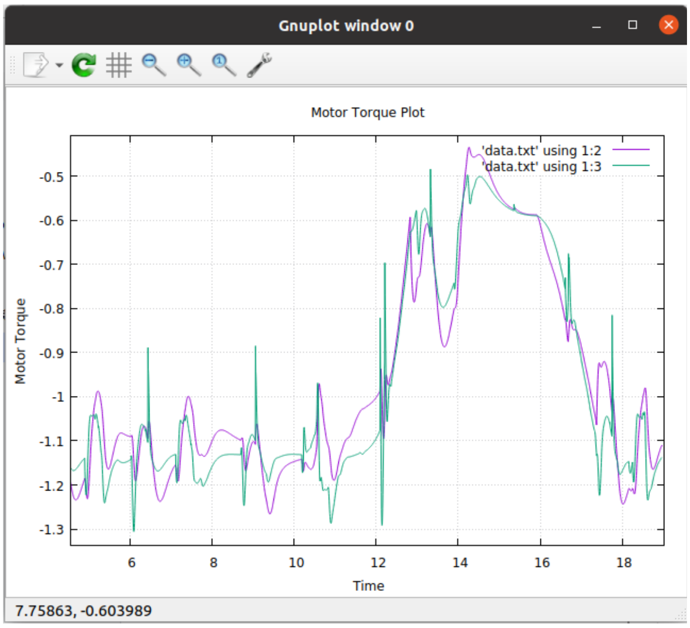
进而，知道关节力矩后，可以得到 ${}^Rr_{com}$ ，正运动学得到 ${}^Rp_i$ ，从而得到 ${}^Rp_{com}$ 。

实例化

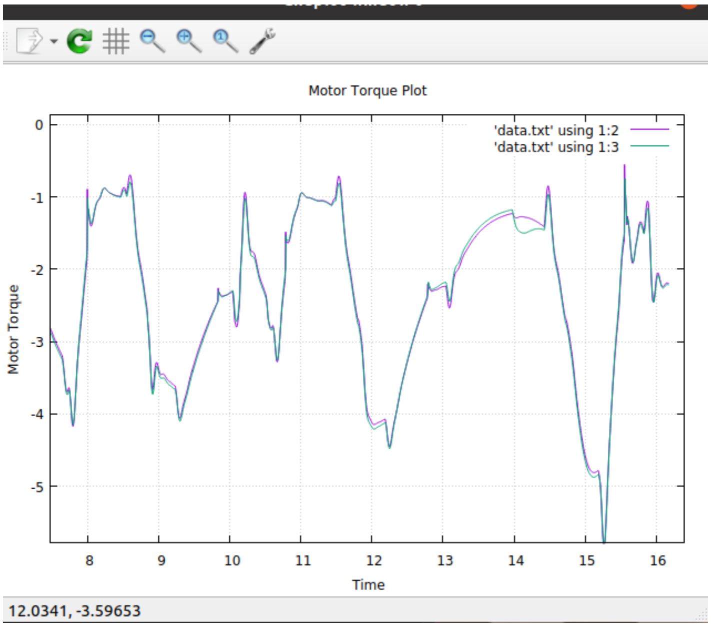
对于diablo机器人，实际上，有（静力学）：

$$^R\tau_1 = {}^Rr_{com1} \times {}^R F$$

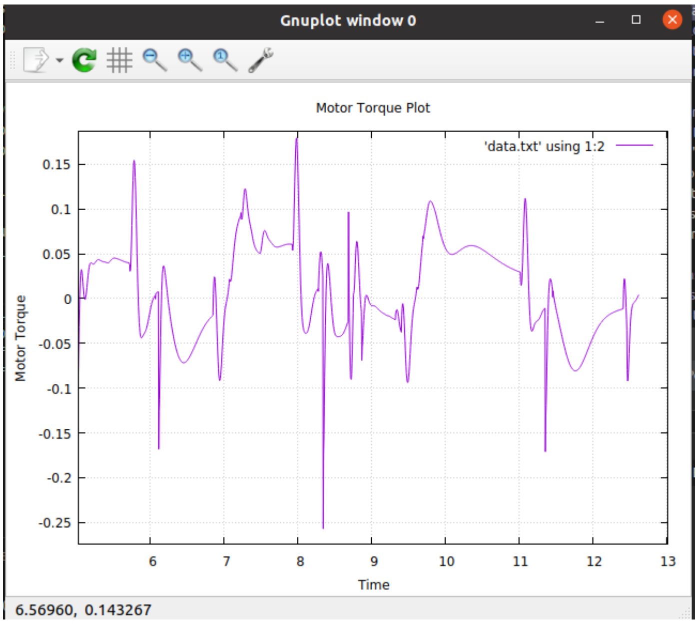
$$^R\tau_2 = {}^Rr_{com2} \times {}^R F - {}^R \tau_1 \text{（可能是由于非直驱导致）}$$

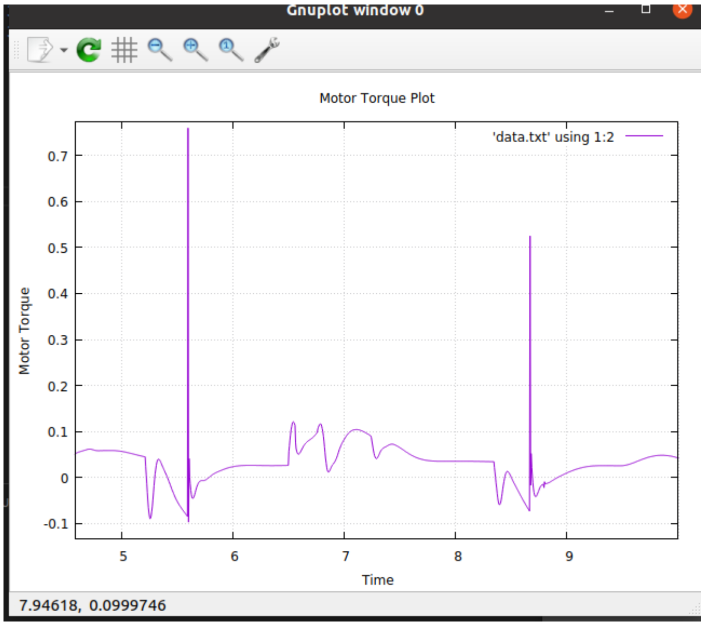


hip计算和实际（下图为两者差）



knee计算和实际（下图为两者差）





由 ${}^R r_{comi} = {}^R p_{com} - {}^R p_i$，可以计算得到：

$$ {}^R\tau_1 + {}^R p_1 \times {}^R F = {}^R p_{com} \times {}^R F $$

$$ ({}^R\tau_1 + {}^R\tau_2) + {}^R p_2 \times {}^R F = {}^R p_{com} \times {}^R F $$

力矩为绕y轴的力矩，所以：${}^R p_1 \times F = -(p_1 f_3 - p_3 f_1)$

可以有：${}^R p_{com} \times {}^R F = \begin{bmatrix} -f_3 & f_1 \end{bmatrix} \begin{bmatrix} p_{com1} \\ p_{com3} \end{bmatrix} = b$

求解方法：

https://zhuanlan.zhihu.com/p/503664717

线性方程组的最小二乘解和最小范数解

对线性方程组 Ax=b，如果未知量个数 n 和方程个数 m 数量相等，有成熟的线性方程组求解方法，比如克莱姆法则、矩阵消元法等。这里我们只考虑方程组的未知量个数…

- 当线性方程组$Ax = b$未知量个数不大于方程个数时，存在最小二乘解：$x^* = (A^\top A)^{-1} A^\top b$。
- 当线性方程组$Ax = b$未知量个数不小于方程个数时，存在最小范数解：$x^* = A^\top (AA^\top)^{-1} b$。

最小范数解

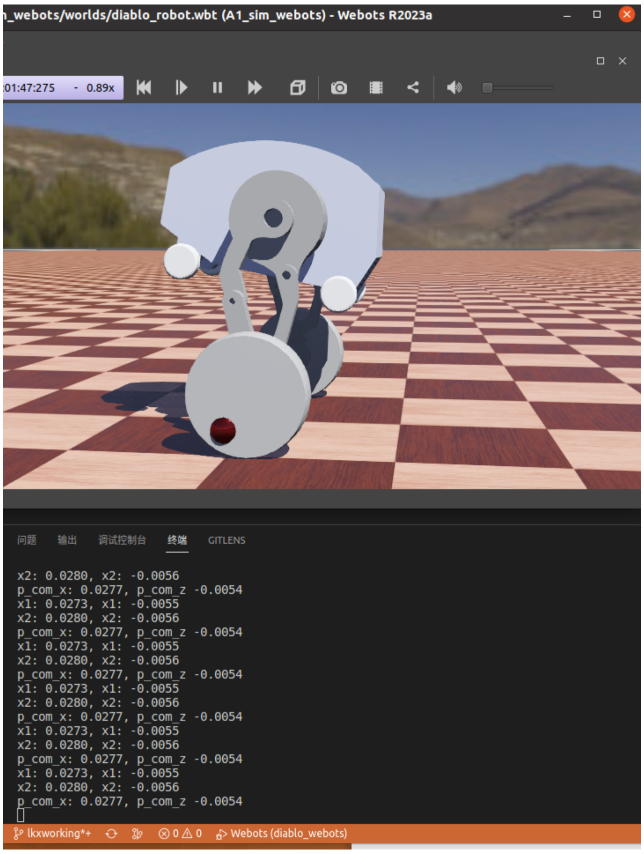
解1：将两个方程单独求解，代码如下

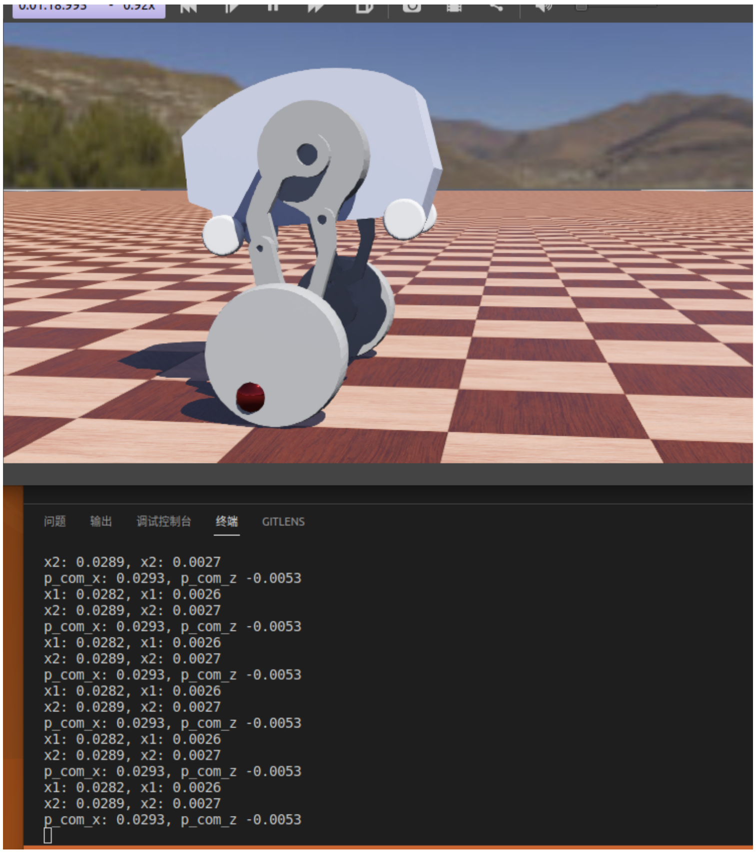
```
float b1 = tau1 - ((p_hip_l[0] + p_hip_r[0]) / 2 * accl[2] - (p_hip_l[2] +
```

```
    p_hip_r[2]) / 2 * accl[0]) * m;
    float b2 = tau1 + tau2 - ((p_knee_l[0] + p_knee_r[0]) / 2 * accl[2] -
(p_knee_l[2] + p_knee_r[2]) / 2 * accl[0]) * m;// tau2 方向不对
    float x1[2], x2[2];
    x1[0] = -b1 * (accl[2])/ (accl[0] * accl[0] + accl[2] * accl[2]) / m;
    x1[1] = b1 * (accl[0])/ (accl[0] * accl[0] + accl[2] * accl[2]) / m;
    x2[0] = -b2 * (accl[2])/ (accl[0] * accl[0] + accl[2] * accl[2]) / m;
    x2[1] = b2 * (accl[0])/ (accl[0] * accl[0] + accl[2] * accl[2]) / m;
    printf("x1: %.4f, x1: %.4f\n", x1[0], x1[1]);
    printf("x2: %.4f, x2: %.4f\n", x2[0], x2[1]);
```

求解结果中包含了机器人质量m，因此要提前知道机器人的上半身质量，得到的结果分别为x1和x2.

结果中当机器人质量准确时，辨识在特定位置ok，但两组解之间存在一定的误差。在抬头低头至其他位置误差较大（下图中p_com为webots模型计算得到的实际质心在机器人基坐标系下的位置，x1和x2为两个方程的解），同时一旦质量改变计算质心位置便不准确！





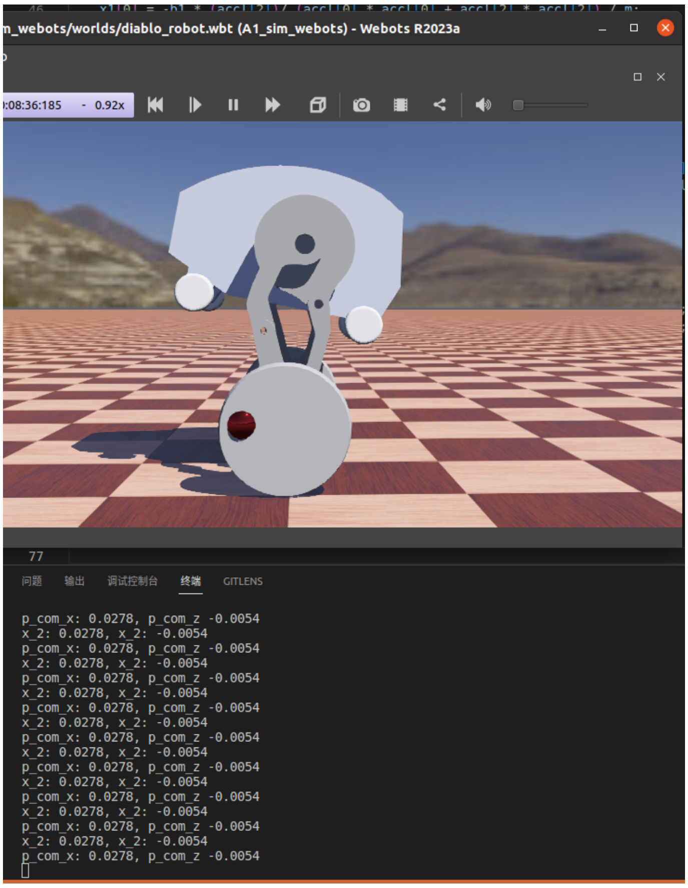
解2：将两个方程结合来求解

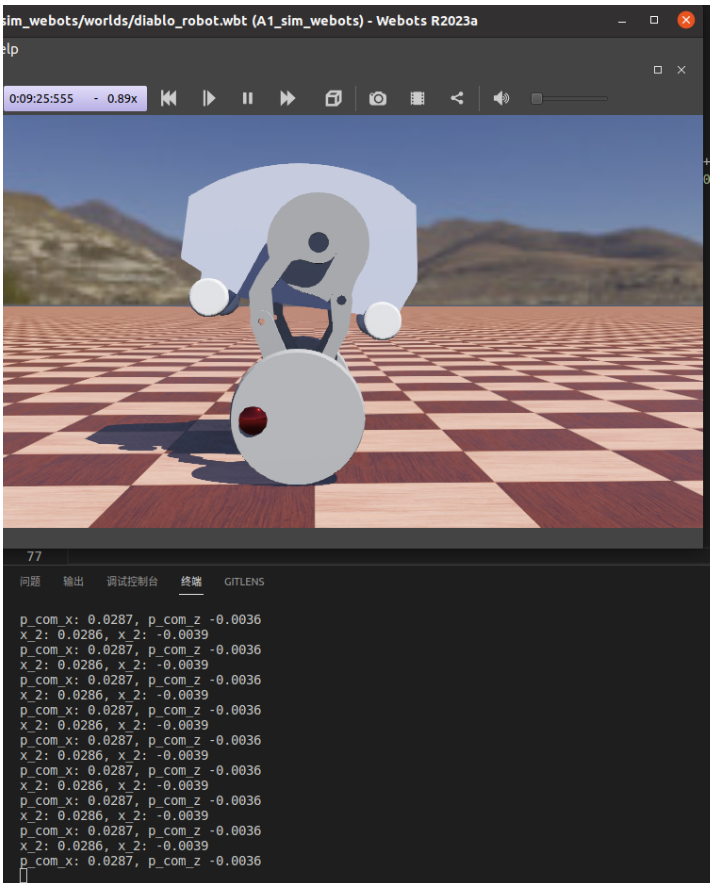
```
    float k = tau1/(tau1+tau2);
    float b12 = accl[0] * (p_hip_l[2] + p_hip_r[2]) / 2 - accl[2] *
(p_hip_l[0] + p_hip_r[0]) / 2;
    float b22 = accl[0] * (p_knee_l[2] + p_knee_r[2]) / 2 - accl[2] *
(p_knee_l[0] + p_knee_r[0]) / 2;

```

```
    float A_2[2],x_2[2];
    w[0] = 1; w[1] = 1;
    A_2[0] = (1 - k) * accl[2];
    A_2[1] = -(1 - k) * accl[0];
    float b_2 = k * b22 - b12;
    x_2[0] = A_2[0]*b_2*w[0]/(A_2[0]*A_2[0]*w[0]*w[0]+A_2[1]*A_2[1]*w[1]*w[1]);
    x_2[1] = A_2[1]*b_2*w[1]/(A_2[0]*A_2[0]*w[0]*w[0]+A_2[1]*A_2[1]*w[1]*w[1]);
```

求解方程中，去除机器人上半身质量对质心位置的影响，求解结果大概效果和上述类似：即在特定位置准确，而在抬头低头至其他角度误差较大，但由于方程中去除了质量的影响，所以当上半身质量改变时，机器人在特定位置仍然可以计算出质心位置。（左图为原质量，右图为增大5kg质量）





结论：由于求解方法为最小范数法，即方程个数小于未知数个数，所以很显然会出现如上问题，但是仍在特定位置结果准确，也证明了算法的方向大致正确，下一步需要找一下是否可以再补充一个方程来得到准确的质心位置。