HXZP 木子羽
1324146673@qq.com

倒立摆仿真

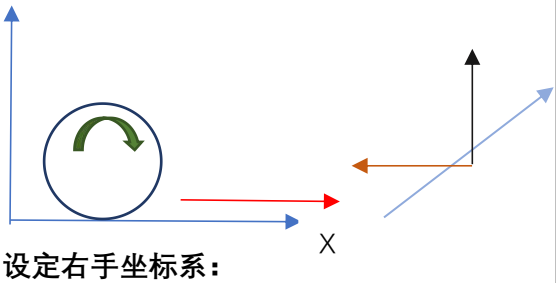
目录

一、二维动力学建模
二、LQR 算法
三、SIMSCAPE 仿真
四、三维动力学建模
（代码模型见附件）

一、 动力学建模

先进行二维倒立摆建模，假设二维平面内存在一个倒立摆，由轮子和杆组成，如 1-1，轮子中有一电机作为驱动，与轮子同轴。地面存在摩擦力。假设不打滑。

对轮子进行坐标建立：



设定右手坐标系：
横向为 X
纵向为 Z
垂直向里为 Y

设置参数：
r：轮子半径
T：电机输出扭矩（顺时针为正）
f：轮子与地面摩擦力
N：轮子与杆在 X 轴的相互作用力
P：轮子与杆在 Z 轴的相互作用力
F：为扭矩在轮子边缘产生的力
x：轮子的位移
v：轮子的绝对速度
a：轮子的加速度
α：轮子的转速加速度
I：轮子的 Y 轴的转动惯量
g：重力加速度

Y 方向力矩平衡方程：

$$T - f * r = I * \alpha \quad (1.1)$$

X 方向力平衡方程：

$$f - N = a * m \quad (1.2)$$

Y 方向上转动惯量：

$$I = \frac{1}{2} * m * r^2 \quad (1.3)$$

为方便 simscape 仿真，倒立摆等效为滑块和杆。设定 F 作为滑块输入，沿 X 轴正方向；Ti 作为杆输入，顺时针为正方向，此时 Ti 为电机实际输出扭矩 T 的反作用扭矩，倒立摆类比修改后，简图为 1-2

对于滑块：

$$F = \frac{T}{r} \quad (1.4)$$

HXZP 木子羽
1324146673@qq.com

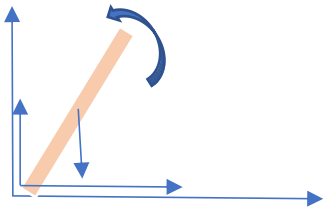
对于杆：

$$F = -\frac{Ti}{r} \qquad (1.5)$$

联立（1.1）（1.2）（1.3）（1.4）
得到：

$$F * r^2 - N * r^2 = (I + m * r^2) * a \quad (a)$$

对杆进行坐标建立：



Vx：X 方向绝对速度
Vz：Z 方向绝对速度
ax：X 方向加速度
az：Z 方向加速度
M：杆的质量
d：链接点到质心的距离
Tp：P 产生的扭矩
Tn：N 产生的扭矩
Jy：Y 轴的转动惯量
θ：偏转角
β：偏转角角速度
γ：偏转角加速度（顺时针为正

X 方向力平衡方程：

$$N = M * ax \qquad (2.1)$$

Z 方向力平衡方程：

$$P = M * az + Mg \qquad (2.2)$$

杆所处坐标系为非惯性坐标系，合成绝对速度为：
X 方向绝对速度：

$$Vx = V + \beta * d * cos\theta \qquad (2.3)$$

Z 方向绝对速度：

$$Vz = -\beta * d * sin\theta \qquad (2.4)$$

对（2.3）（2.4）求导得：

$$ax = a + \gamma * d * cos\theta - \beta^2 * d * sin\theta \qquad (2.5)$$
$$az = -\gamma * d * sin\theta - \beta^2 * d * cos\theta \qquad (2.6)$$

Y 方向上，N 得 P 产生的扭矩：

$$P * d * sin\theta = Tp \qquad (2.7)$$
$$N * d * cos\theta = Tn \qquad (2.8)$$

Y 方向上，扭矩平衡方程：

$$Jy * \gamma = Tp - Tn + T \qquad (2.9)$$

联立(2.9) (2.7) (2.8) (2.5) (2.6) (2.1) (2.2) (1.5) 并且使用线性化

$$(M * d^2 + Jy) * \gamma = M * g * d * \theta - M * d * a - F * r \qquad (b)$$

联立(a) (2.1) (2.5)

$$\mathbf{F}*\mathbf{r}^2-\mathbf{M}*\mathbf{r}^2*\mathbf{d}*\boldsymbol{\gamma}=\left(\mathbf{I}+\mathbf{m}*\mathbf{r}^2+\mathbf{M}*\mathbf{r}^2\right)*\mathbf{a} \quad \text{(c)}$$

对(a) (b)进行重整理得到两条分别关于 a 和γ的微分方程式，可用 matlab 进行整理:

*假设：m=4KG M=1.6KG r=0.1m d=0.2m J=0.064KG*m^2 g=9,81m/s^2*

$$\boldsymbol{\gamma}=(\mathbf{225/224}*\boldsymbol{F})-(\mathbf{6213/280}*\boldsymbol{\theta}) \quad \text{(x)}$$

$$\boldsymbol{a}=\left(\frac{5}{56}*\boldsymbol{F}\right)-(\mathbf{327/350}*\boldsymbol{\theta}) \quad \text{(y)}$$

二、 LQR 算法

将(x) (y)为状态空间方程形式

选取状态变量 z = [x v a θ β γ] 输出 u = F

设定：
a = 43.6852
b = -1.9775
c = -1.8394
e = 0.2148

矩阵 A：

$$\begin{matrix}0&1&0&0\\0&0&c&0\\0&0&0&1\\0&0&a&0\end{matrix}$$

矩阵 B：

$$\begin{matrix}0\\e\\0\\b\end{matrix}$$

1)求矩阵 A 的特征根：

$$\begin{matrix}\mathbf{0}\\\mathbf{0}\\\mathbf{6.6095}\\\mathbf{-6.6095}\end{matrix}$$

存在特征根大于 0，系统不稳定

2)能控性分析：

矩阵 A 的维度为 4

$$\boldsymbol{C0}=[\boldsymbol{B}\ \ \boldsymbol{A}*\boldsymbol{B}\ \ \boldsymbol{A}^2*\boldsymbol{B}\ \ \boldsymbol{A}^3*\boldsymbol{B}]$$

$$\boldsymbol{Rank}(\boldsymbol{C0})=\mathbf{4}$$

系统可控

3)引入代价函数：

设定 Q 矩阵和 R 矩阵， QR 分别为半正定矩阵和正定矩阵。Q 为 n*n，R 为 p*p；n 为状态变量个数，p 为输入的个数。

Q 矩阵代表对状态变量的约束，R 矩阵代表对输入的约束，例如随着 R 的增大，会逐渐限制输入大小。
由于实际使用中位移不好测量（例如传感器精度或者打滑），因此对位移的约束减少，但不能为 0；
电机的扭矩有最大的输出限制，需要选择一个较为合适的 R 的来限制输出；

$$Q=\begin{matrix}[0.01,0,0,0\\0,1000,0,0\\0,0,1000,0\\0,0,0,100]\end{matrix}$$

$$R=0.1$$

4)使用 matlab 的 LQR 工具进行计算：

```
K = lqr(A,B,Q,R);
```

得到输出

```
K =

   -0.3162 -100.2122 -217.3449  -47.4633
```

输出：

$$\boldsymbol{u} = -\boldsymbol{Kz}(\boldsymbol{t})$$

$$\boldsymbol{F} = -(\boldsymbol{K1} * \boldsymbol{z1} + \boldsymbol{K2} * \boldsymbol{z2} + \boldsymbol{K2} * \boldsymbol{z3} + \boldsymbol{K4} * \boldsymbol{z4} + \boldsymbol{K5} * \boldsymbol{z5} + \boldsymbol{K6} * \boldsymbol{z6})$$

F 作用的目标是将状态变量收敛为 0，假设需要将速度维持在 2，则需要偏置 z2，也就是说：

$$\boldsymbol{z2} = \boldsymbol{z2'} - \boldsymbol{2}$$

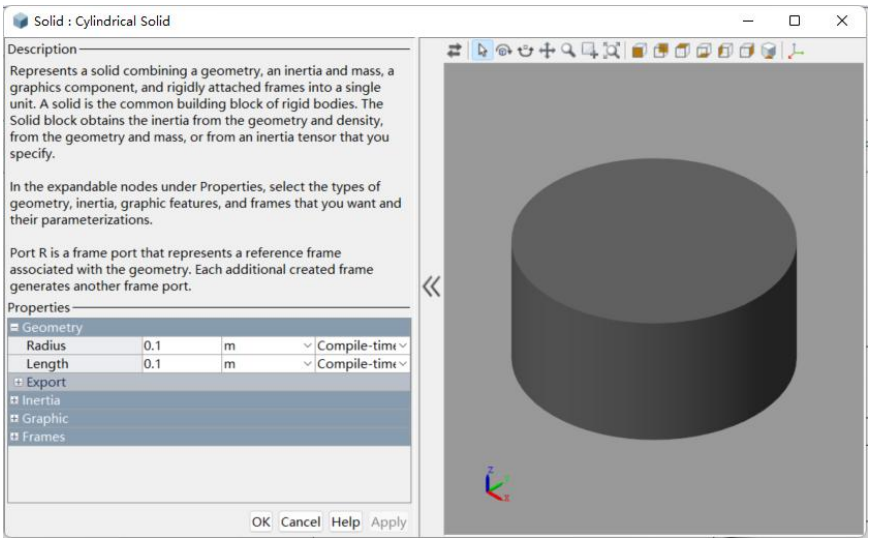
z2'为真实的速度状态，这样当 z2 收敛为 0 的时候：

$$\boldsymbol{z2'} = \boldsymbol{2}$$

三、 SIMSCAPE 仿真

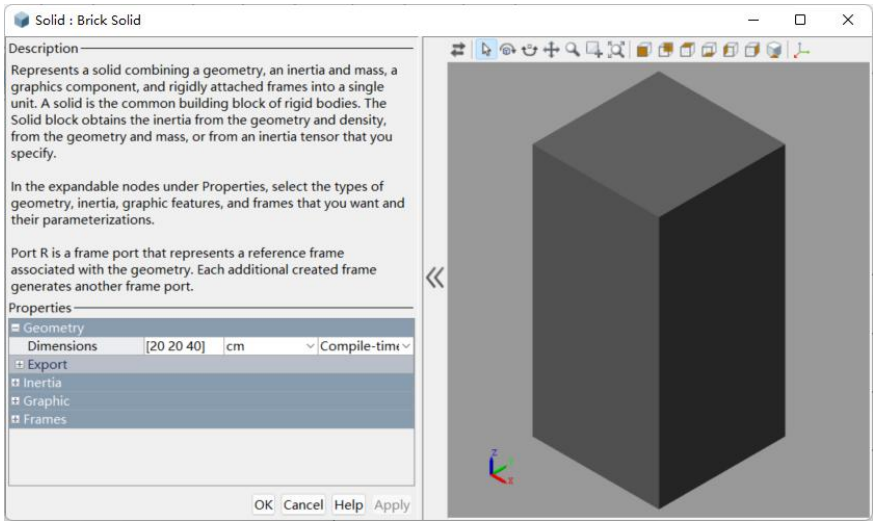
模型搭建：

圆柱模块作为轮子：设定半径、高度，密度修改为 1g/m^3



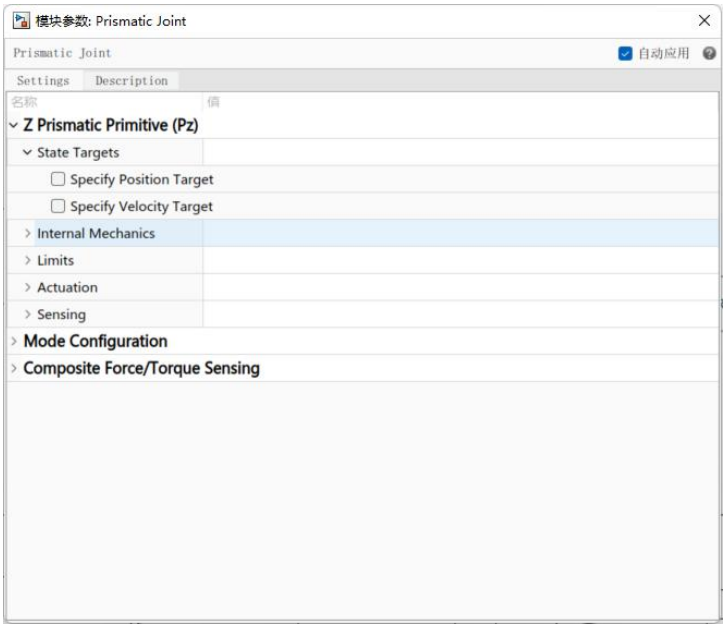
inertia 是指其惯性相关特性

砖块模块作为车体：设定长宽高，密度修改为 0.1g/m^3

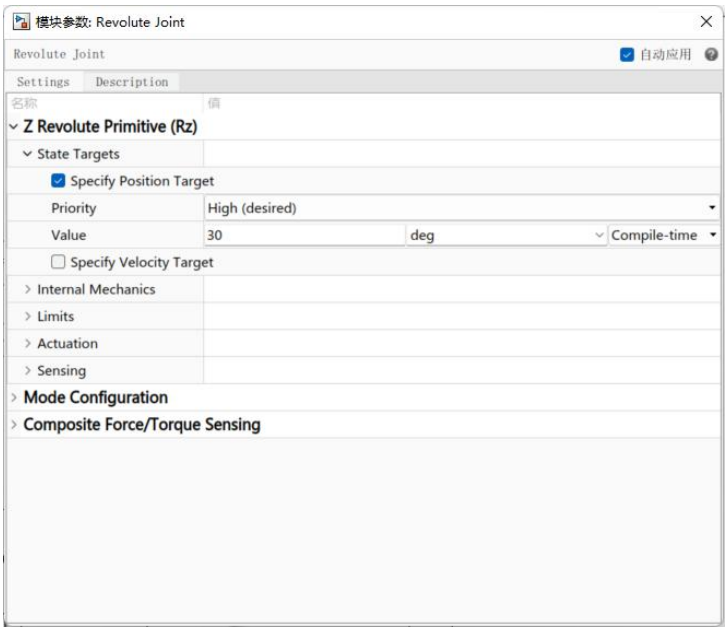


HXZP 木子羽
1324146673@qq.com

使用滑块作移动动作：沿着+X 轴方向为正
State Targets 是初始角度或者速度
Sensing 为传感器配置
internal Mechanics 可以配置其内部物理属性
actuation 配置驱动力方法

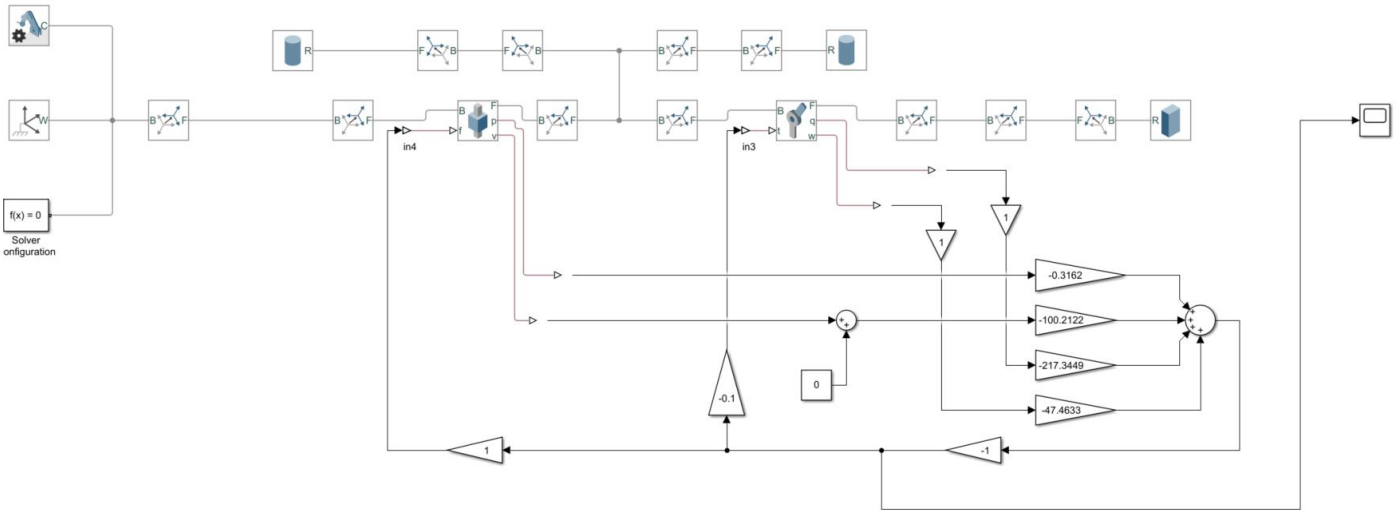


使用转动副作旋转动作：设定初始角度为 30，沿着+Y 方向顺时针为正

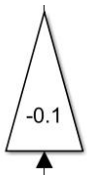


HXZP 木子羽
1324146673@qq.com

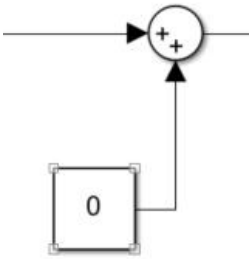
总拓扑：



由于输出为力 F，在传入转动副中需要叉乘半径，并且根据关系（1.5）

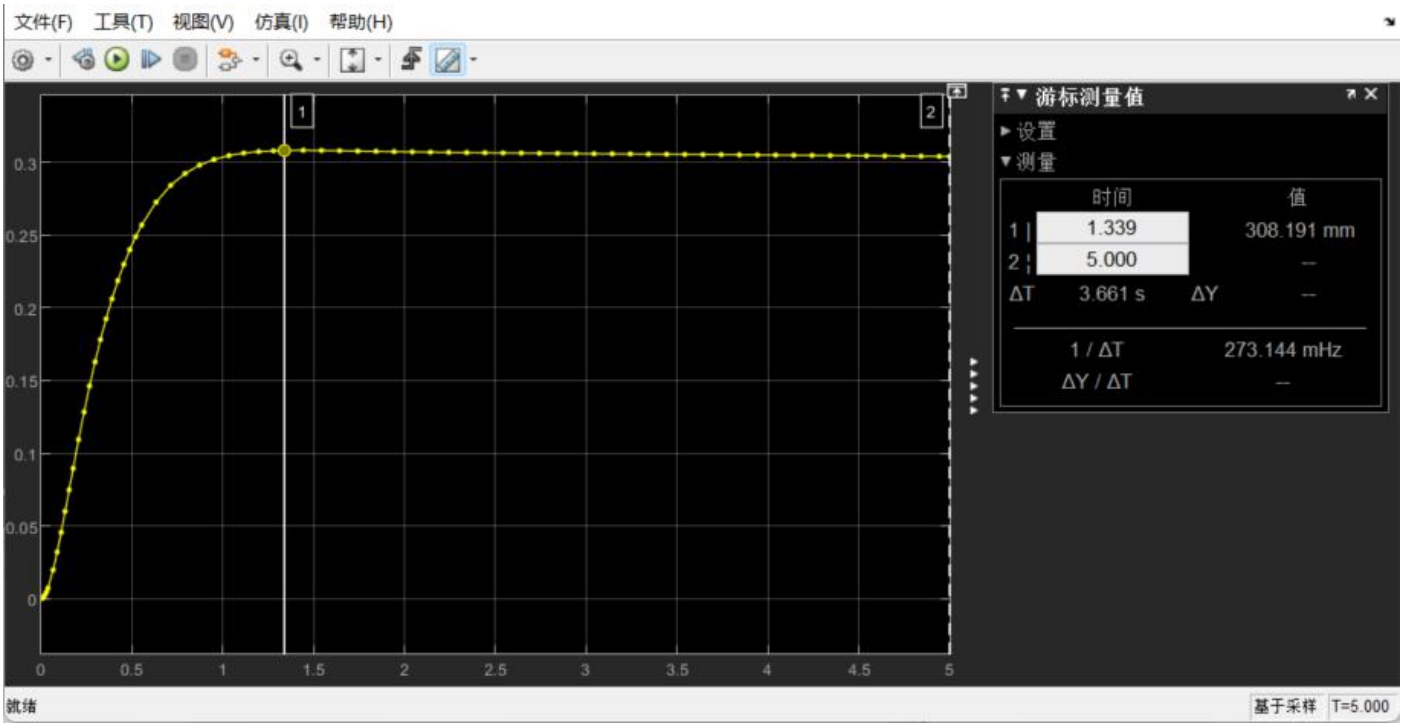


通过偏执修改目标速度：



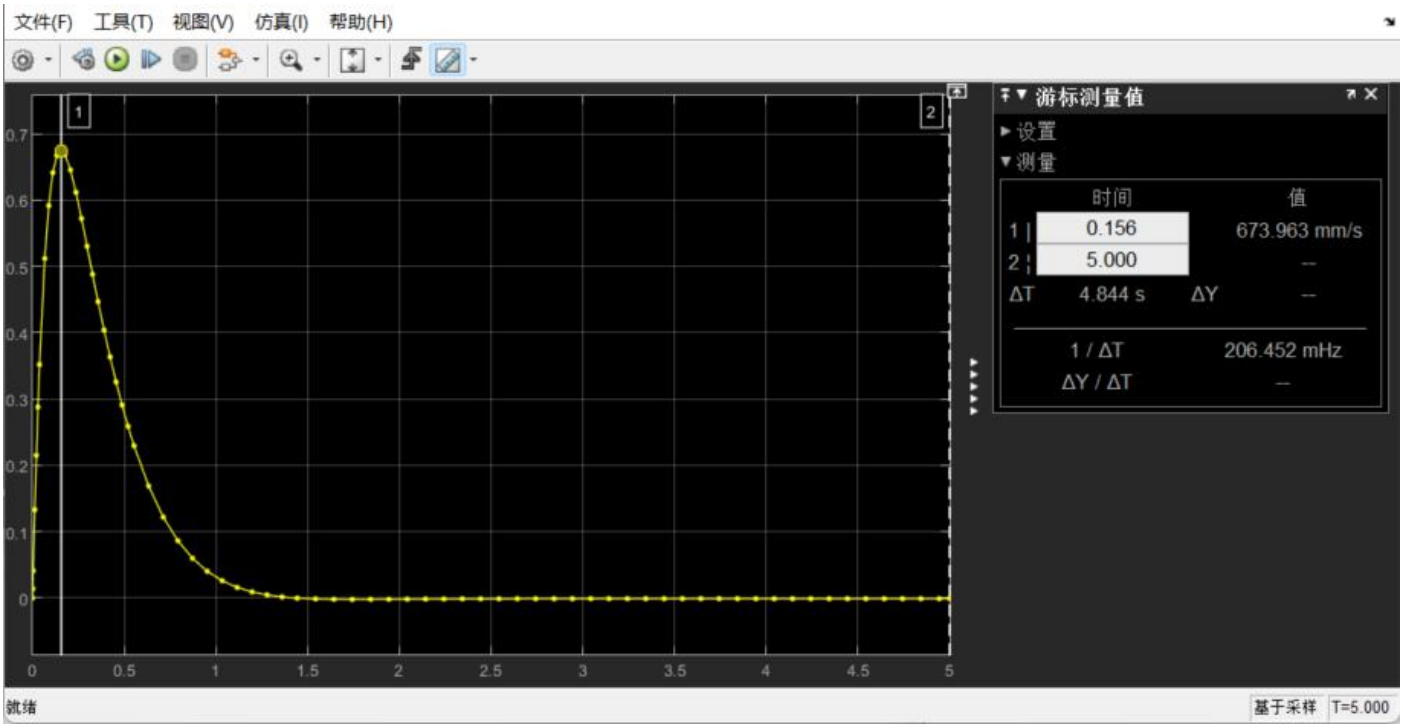
仿真结果分析：

车体最大位移约为 308mm，耗时约为 1.339 秒

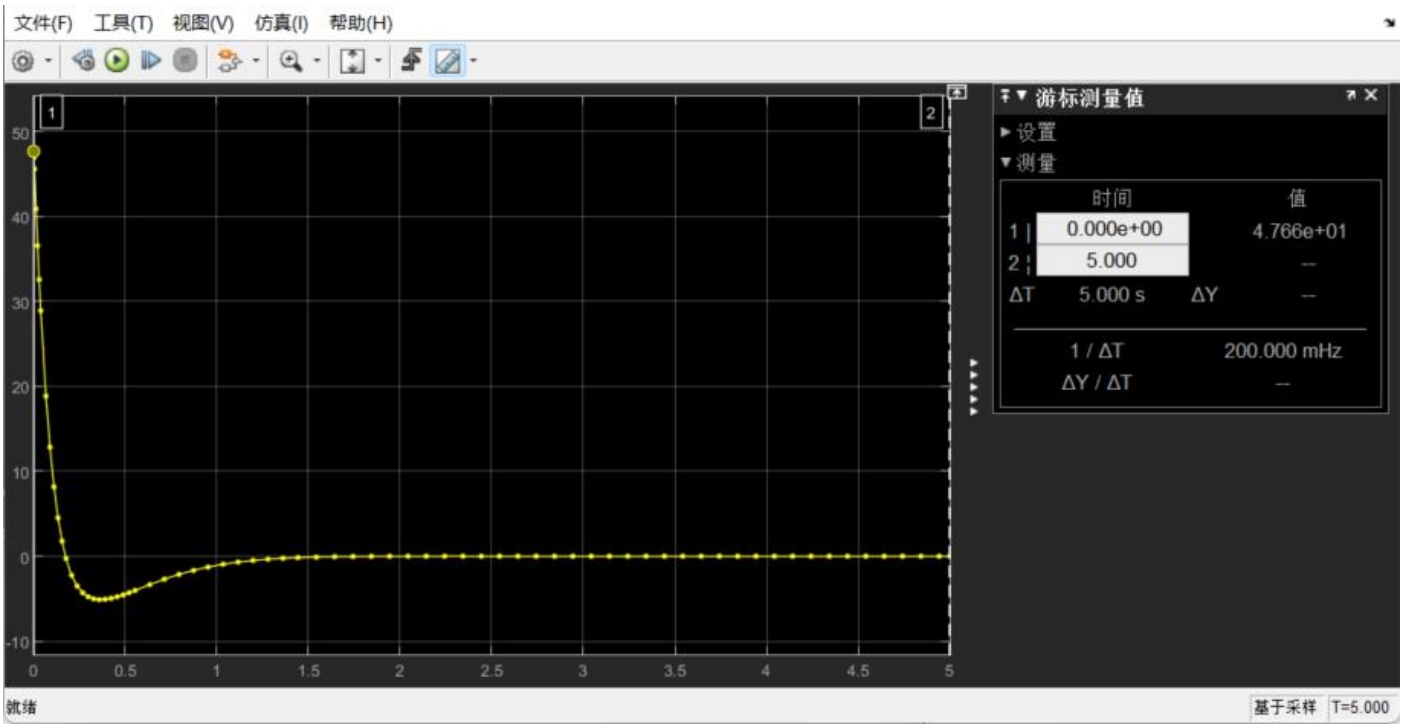


HXZP 木子羽
1324146673@qq.com

车体最大速度为 0.673m/s，推测电机最大转速为 8.91rpm



峰值输出为 47.66N，换算扭矩为 4.766N*m



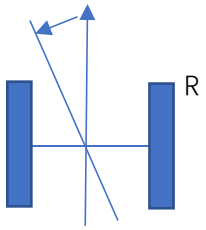
对于上述结果可以进行对电机进行选型，或者分析性能，分析可行性。

HXZP 木子羽
1324146673@qq.com

四、三维拓展

三维与二维主要区别是在 Y 轴上多了一个自由度 Yaw，以及 Roll，这里暂且不分析 Roll 轴情况，假定始终 Roll 为 0，只分析 Yaw；

设定逆时针为正方向



Tr：右轮驱动扭矩
Tl：左轮驱动扭矩
Ω：偏转角
ω：偏转角角速度
o：偏转角角加速度
Jz：z 方向转动惯量
Nr Nl：左右轮与杆水平作用力

根据前文式（a）

$$Tr * r - Nr * r^2 = \left(I + m * r^2\right) * ar \quad (3.1)$$

$$Tl * r - Nl * r^2 = \left(I + m * r^2\right) * al \quad (3.2)$$

Z 方向扭矩平衡方程：

$$(Nr - Nl) * l/2 = Jz * o \quad (3.3)$$

转动速度和轮子加速度关系：

$$o = \frac{ar - al}{l} \quad (3.4)$$

联立上述四条式子：

$$o = \frac{Tr - Tl}{r * \left(2 * \frac{Jz}{l} + l * \frac{m * r^2 + I}{r^2}\right)} \quad (c)$$

对于

$$\left(M * d^2 + Jy\right) * \gamma = M * g * d * \theta - M * d * a - F * r$$
$$\mathrm{F} * \mathrm{r}^2 - \mathrm{M} * \mathrm{r}^2 * \mathrm{d} * \gamma = \left(\mathrm{I} + \mathrm{m} * \mathrm{r}^2 + \mathrm{M} * \mathrm{r}^2\right) * \mathrm{a}$$

X 方向驱动力为左右线性叠加，并且质量发生改变，因此修改后为：

$$\left(M * d^2 + Jy\right) * \gamma = M * g * d * \theta - M * d * a - F * r$$
$$\mathrm{F} * \mathrm{r}^2 - \mathrm{M} * \mathrm{r}^2 * \mathrm{d} * \gamma = \left(2 * \mathrm{I} + 2 * \mathrm{m} * \mathrm{r}^2 + \mathrm{M} * \mathrm{r}^2\right) * \mathrm{a}$$

取输入

$$F = (Tr + Tl) * r$$
$$T = Tr - Tl$$

可以推算出左右轮扭矩分别为

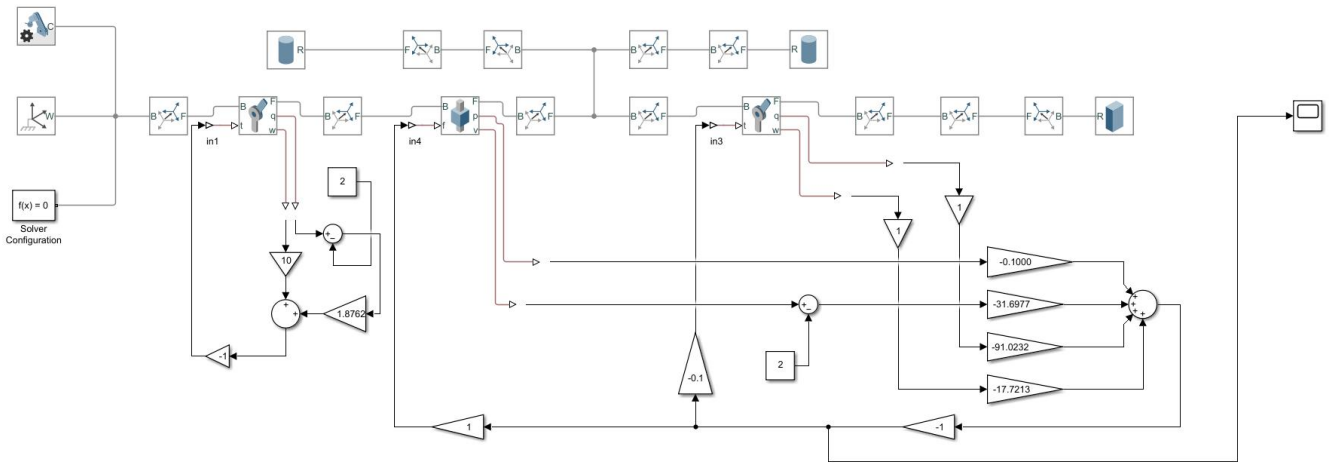
$$\mathrm{Tr} = \frac{1}{2} * \left(\frac{\mathrm{F}}{\mathrm{r}} + \mathrm{T}\right)$$
$$\mathrm{Tl} = \frac{1}{2} * \left(\frac{\mathrm{F}}{\mathrm{r}} + \mathrm{T}\right)$$

按照前文所述对（a）（b）（c）进行建立状态空间方程，使用 lqr 工具得到 K

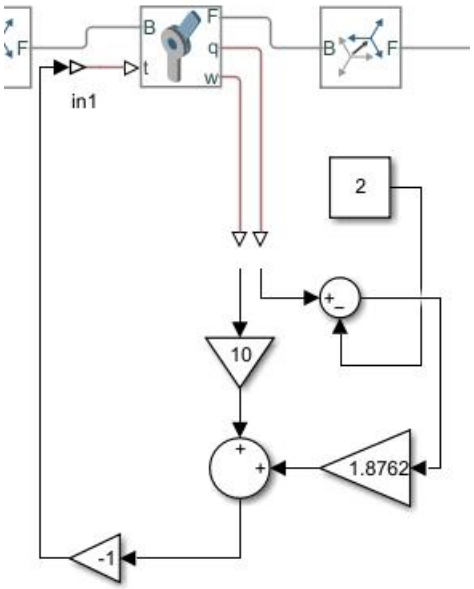
```
K =

   -0.1000  -31.6977  -91.0232  -17.7213   -0.0000   -0.0000
   -0.0000   -0.0000   -0.0000   -0.0000   10.0000    1.8762
```

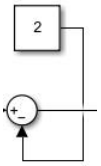
总拓扑：

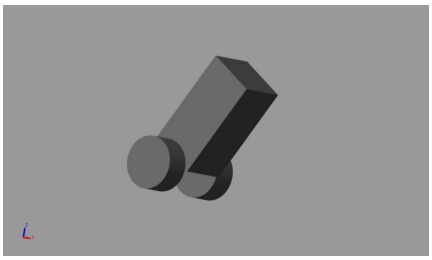


与二维建模区别在于 z 轴上添加了一个转动副代替转向：



控制块





精彩瞬间

在仿真界面里可以安全快速的模拟调试，Matlab 的模型较为简单，后续可以使用 urdf 文件或者 stl 文件在 ros、coppeliasim、webots 等里导入更复杂的模型，获得更好的仿真效果和控制体验。当然仿真和建模是在各种“理想条件”下实现的，因此不要单单侥幸与仿真的成功，路漫漫其修远兮。