

1. 系统建模

对于机器人的平衡，纵向以及转向运动，可以将模型简化为三维空间中的类一阶倒立摆模型，即忽略腿长的变化，转向运动由两个驱动轮差速运动控制。

1.1 模型定义

称机器人机体质心到两个驱动轮转轴中心的连线为摆杆，得到如图 1 所示平衡车模型。

该模型的变量与参数定义如表 1、表 2 所示。

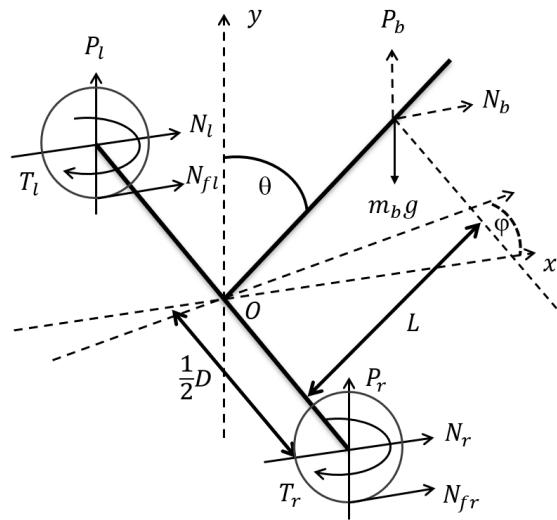


图 1 平衡车模型

表 1 平衡车模型变量定义

符号	含义	正方向	单位
θ	摆杆与竖直方向夹角	图示为正方向	rad
φ	摆杆偏航角	图示为正方向	rad
x_i	驱动轮位移	X 轴正方向	m
x_b	机体位移	X 轴正方向	m
P_i	驱动轮对摆杆力的竖直分量	箭头所示	N
P_b	摆杆受到的竖直方向力	箭头所示	N
N_i	驱动轮对摆杆力的水平分量	箭头所示	N
N_b	摆杆受到的水平方向力	箭头所示	N
T_i	驱动轮输出力矩	箭头所示	N·m

表 2 平衡车模型参数定义

符号	单位	含义
D	m	车体转向直径
R	m	驱动轮半径
L	m	摆杆质心到驱动轮转轴中心的直线长度
m_w	kg	驱动轮质量
m_b	kg	摆杆质量
I_b	$kg \cdot m^2$	摆杆绕质心关于 Z 轴的转动惯量
I_w	$kg \cdot m^2$	驱动轮转子转动惯量
I_z	$kg \cdot m^2$	整车质心关于 Y 轴的转动惯量

1.2 经典力学分析

对驱动轮，有：

$$m_w \ddot{x}_l = N_{fi} - N_i \quad (1)$$

$$I_w \frac{\ddot{x}_l}{R} = T_i - N_{fi} R \quad (2)$$

对摆杆，有：

$$m_b \ddot{x}_b = N_{fl} + N_{fr} \quad (3)$$

$$m_b \ddot{y}_b = P_r + P_l - m_b g \quad (4)$$

$$I_b \ddot{\theta}_b = (P_r + P_l) L \sin \theta_b - (N_r + N_l) L \cos \theta_b - (T_r + T_l) \quad (5)$$

对整车，有：

$$I_z \ddot{\phi} = \frac{D}{2} (N_l - N_r) \quad (6)$$

其中：

$$x_b = x + L \sin \theta_b \quad (7)$$

$$y_b = L \cos \theta_b \quad (8)$$

1.3 状态空间模型

定义非线性模型：

$$\dot{x} = f(x, T_l, T_r)$$

利用 *MATLAB* 符号运算工具，联立式 (1) 到式 (8)，通过函数 *solve* 对方程组消去中间变量，最终得到非线性表达式 (9)，式 (10)，和式 (11)

$$2m_w \ddot{x} + m_b \left(\ddot{x} - L \left(\sin(\theta_b) \dot{\theta}_b^2 - \dot{\theta}_b \cos(\theta_b) \right) \right) - \frac{T_l}{R} - \frac{T_r}{R} + \frac{2I_w \ddot{x}}{R^2} = 0 \quad (9)$$

$$T_l + T_r + I_b \ddot{\theta}_b - L \sin(\theta_b) \left(g m_b - L m_b \left(\cos(\theta_b) \dot{\theta}_b^2 + \ddot{\theta}_b \sin(\theta_b) \right) \right) - 2L m_w \ddot{x} \cos(\theta_b) + \frac{L T_l \cos(\theta_b)}{R} + \frac{L T_r \cos(\theta_b)}{R} - \frac{2I_w L \ddot{x} \cos(\theta_b)}{R^2} = 0 \quad (10)$$

$$D T_l R = m_w \ddot{\phi}_b D^2 R^2 + I_w \ddot{\phi}_b D^2 + T_r D R + 2I_z \ddot{\phi}_b R^2 \quad (11)$$

定义状态向量 x 与控制向量 u 分别为：

$$x = \begin{bmatrix} x \\ \dot{x} \\ \theta_b \\ \dot{\theta}_b \\ \phi \\ \dot{\phi} \end{bmatrix}, u = \begin{bmatrix} T_l \\ T_r \end{bmatrix}$$

根据式状态向量 x 与控制向量 u ，求非线性模型在平衡点处的雅可比矩阵并对其线性化，即将 $\theta = 0$ ，则 $\sin \theta = \theta, \cos \theta = 1$ 代入式 (9) 到 (11) 后，求：

$$A = \frac{\partial f}{\partial x}(x, u), \quad B = \frac{\partial f}{\partial u}(x, u)$$

其中， x 、 u 是系统平衡点，即方程 $f(x, u) = 0$ 的解：

$$x = \begin{bmatrix} x_b \\ 0 \\ 0 \\ 0 \\ \phi_b \\ 0 \end{bmatrix}, u = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

最终求得的状态方程为：

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & A_1 & A_2 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & A_3 & A_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 & 0 \\ B_1 & B_2 \\ 0 & 0 \\ B_3 & B_4 \\ 0 & 0 \\ B_5 & B_6 \end{bmatrix} u \quad (12)$$

由于表达式略为复杂，此处用符号代替，具体内容可通过运行 MATLAB 代码在工作空间中查看。

所有变量均可通过平衡车上各个传感器（如驱动轮编码器，IMU 模块等）直接测量或通过数据融合解算得到，故可得系统输出 $y = I_6 x$ ，其中 I_6 为单位阵。

代入平衡车实车模型参数后可以得到该状态空间模型的状态矩阵 A 和控制矩阵 B ，在 MATLAB 中使用 `rank` 函数求得其可控矩阵满秩，系统可控。系统输出矩阵 C 为单位矩阵，系统可观。

1.4 仿真验证

将 Solidwork 中搭建的模型分模块导入到仿真软件 Webots 中，通过对关节电机设置其允许位移为 0，得到一个定腿长的平衡车模型。

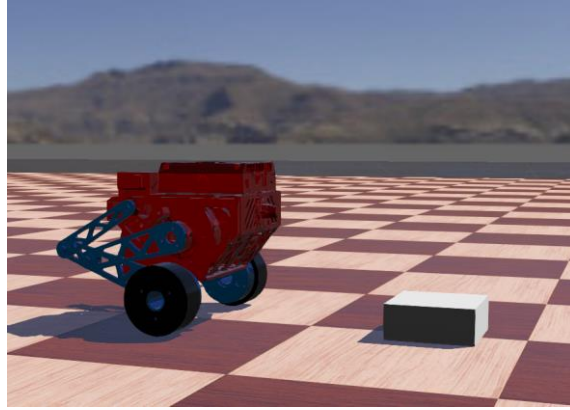


图 2 仿真环境中的整车模型

通过使用 MATLAB 中的 `lqr` 函数计算得到反馈增益矩阵 K 。因为在仿真中主要目的是验证模型建立是否正确，用于计算 K 矩阵的 Q 和 R 权矩阵的数值选取只关注机器人俯仰角的稳定。经检验，得到模型正确。

2 控制器设计

基于本项目的平衡车构型，除了机器人的平移、转向和保持平衡的控制外，还需要控制平衡车双腿的长度及其横滚角。

2.1 LQR 控制器

基于上述求得的状态空间方程，通过一个 LQR 控制器可以较好地对机器人平移、旋转以及机体俯仰角进行控制。

设计控制律为系统状态的线性组合：

$$u = -Kx = - \begin{bmatrix} K_{11} & K_{12} & K_{13} & K_{14} & K_{15} & K_{16} \\ K_{21} & K_{22} & K_{23} & K_{24} & K_{25} & K_{26} \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \theta_b \\ \dot{\theta}_b \\ \phi \\ \dot{\phi} \end{bmatrix}$$

通过线性二次型调节器计算反馈矩阵，定义代价函数为

$$J = \int_0^{\infty} (x^T Q x + u^T R u) dt$$

其中：

- x 是系统状态向量；
- u 是控制输入向量；
- Q 是对状态变量的加权矩阵，通常是半正定矩阵，反映了对状态偏离的惩罚；
- R 是对控制输入的加权矩阵，通常是正定矩阵，反映了对控制能量的惩罚。

最优控制律：

为了使代价函数 J 达到最小，控制输入 u 应满足以下关系：

$$u = -R^{-1}B^T Px$$

其中：

- P 是代数 Riccati 方程的解。

反馈增益矩阵：

通过上述控制律，反馈增益矩阵 K 可以表示为：

$$K = R^{-1}B^T P$$

代数 Riccati 方程：

矩阵 P 需要满足以下代数 Riccati 方程：

$$A^T P + PA - PBR^{-1}B^T P + Q = 0$$

其中：

- A 和 B 是系统状态空间方程中的系统矩阵和输入矩阵。

为了可以控制平衡车的移动和转向，还需要实现对位移和偏航角两个状态变量的轨迹跟踪。在系统输入中加入参考输入，即状态期望：

$$u = -K(x - x_d)$$

其中 x_d 为：

$$x_d = \begin{bmatrix} x_d \\ 0 \\ 0 \\ 0 \\ \phi_d \\ 0 \end{bmatrix}$$

x_d 为期望位移， ϕ_d 为期望偏航角，两者均由其期望一阶微分量对时间积分得到。

2.2 双环 PID 控制器

基于该项目平衡车的四连杆腿部构型,可以通过关节电机的输出来调整机器人的腿部长;同时,为了让机器人在进行视觉识别时机体保持横滚角水平,还需要根据机器人实时的横滚角值来调整两边腿的长度。

为了让机器人的腿长控制有更快的响应速度和鲁棒性,引入两个双环 PID 控制器分别对机器人的腿长和横滚角进行控制,最后并行输出到电机控制电流值。

腿长的控制框图如图 3 所示:

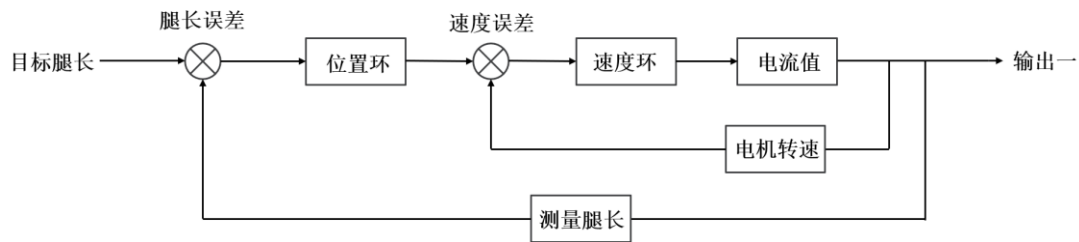


图 3 机器人腿长双环 PID 控制框图

横滚角的控制框图如图 4 所示:

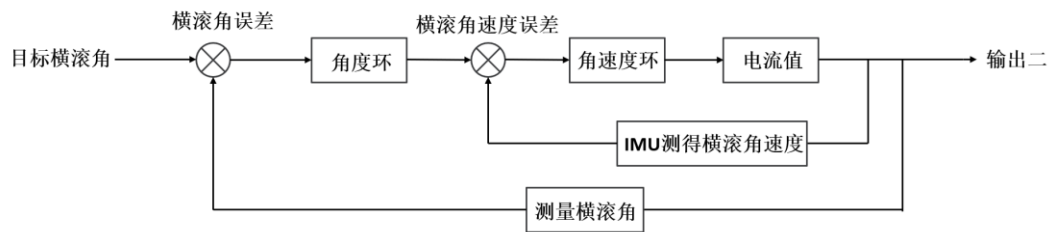


图 4 机器人横滚角双环 PID 控制框图

最终发送的电机电流值为两个控制器的输出之和。

2.3 增益矩阵 K 的参数拟合

在建模阶段,我们将倒立摆的杆长假设为定值,但实际上,该值会随着机器人腿长的变化而改变,其对应的模型也会有差异。由于求得解析解较为困难,我们将机器人腿长作为拟合的自变量,对增益矩阵 K 进行四阶多项式拟合。

由于实车环境下我们无法直接测量机器人实时腿长,需要由一些角度关系推算得到。将模型简化至如图 5 所示。

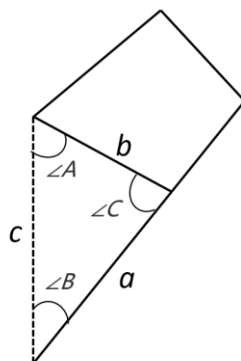


图 5 腿部四连杆结构简化模型

其中，三角形边 a 、边 b 可以通过在机械图纸中测量得到，角 A 可以由电机编码器求得，边 c 则是期望求得的机器人腿长。由余弦定理得式 (13)：

$$a^2 - b^2 - c^2 + 2bc \cos(A) = 0 \quad (13)$$

通过 MATLAB 的 solve 函数求解（排除负数解）得式 (14)：

$$c = b \cos(A) + \sqrt{a^2 + b^2 \cos^2(A) - b^2} \quad (14)$$

经测量，实车边 $b = 0.11291m$ ，边 $a = 0.11461m$ ，将数据代入并由 0 至 90 度遍历角 A 可以得到角 A 与腿长的关系曲线如图 6 所示。

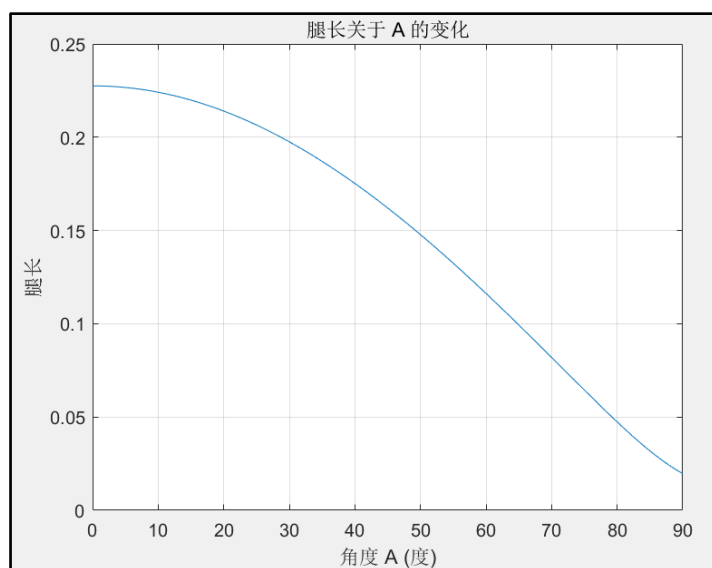


图 6 机器人腿长与角度 A 的关系曲线

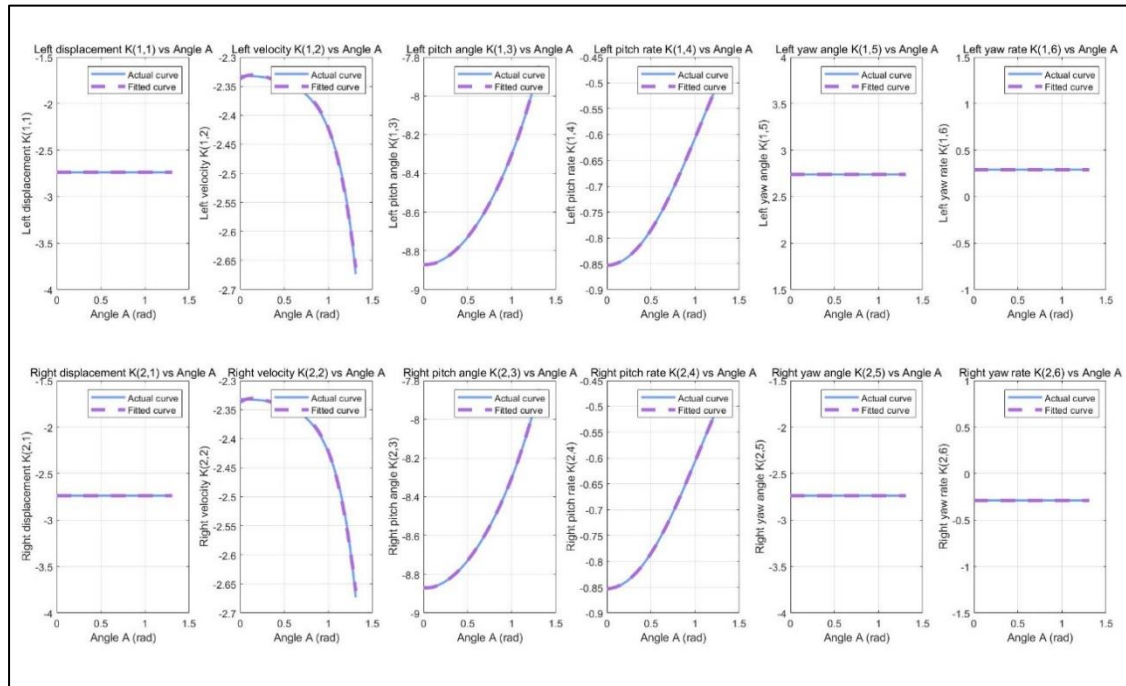
在 MATLAB 中遍历腿长，计算不同腿长值下的增益矩阵 K ，并对 K 矩阵每个元素进行关于角度 A 的多项式拟合，求得拟合系数并存到 EXCEL 表格中，以便将它们添加进实车代码中进行增益矩阵的实时计算。

拟合公式如式 (15)：

$$y = a_4 x^4 + a_3 x^3 + a_2 x^2 + a_1 x + a_0$$

其中 y 对应 K 矩阵的各个元素， x 为测得角度 A 的值。

最终的拟合效果如图 7 所示，蓝色曲线为各个角度 A 值下的实际 K 矩阵值，紫色曲线为拟合值。

图 7 增益矩阵 K 的实际曲线与拟合曲线对比

通过对比, 可知拟合效果满足实车控制需求。

3. 实车硬件选型及代码框架介绍

基于项目需求及材料储备, 本次选取的电控硬件型号如表 3 所示。

表 3 平衡车实车主要电控硬件表

名称	部分技术参数	功能	数量
Robomaster 开发板 C 型	输入电压: 8-28 V IMU+E-compass: 1 个	搭载的芯片提供足够高的运算能力, 让机器人控制频率保持	1

	CAN 总线接口: CAN1: 2 个; CAN2: 2 个 UART 接口: 2 个 SPI 接口: 1 个	1000Hz。电机通讯、遥控器信息接收以及视觉电脑通讯	
迈克 mc6cmini 遥控器航模接收器接收机	通道数量: 6 频率范围: 2401-2478 Mhz 控制范围: >800 m	发送摇杆值, 发送机器人平移、旋转、跳跃等指令	1
瓴控 MG6012E-i8v3 伺服电机	额定电压: 48 V 空载转速: 310 rpm 额定扭矩: 6 Nm 峰值扭矩: 16 Nm	机器人关节电机, 控制机器人的腿长以及机体横滚角	2
Robomater GM6020 电机	额定电压: 24 V 空载转速: 320 rpm 额定扭矩: 1.2 Nm 峰值扭矩: 1.4 Nm	机器人驱动轮电机, 执行平移、旋转、维持平衡等动作	2
经纬 M600 系列 TB47S 智能飞行电池	容量: 4500 mAh 电压: 22.2 V	整套机器人电控系统的供电	1
DC-DC 可调电源模块	输入电压: 7-32 V 输出电压: 0.8-28 V 恒流范围: 0.2-12 A	为视觉电脑提供适配的工作电压及电流	1

4. 实车测试结果与数据分析

对整车多个控制器的控制效果进行分析, 除了腿长控制器和横滚角控制器的测试, 所有数据测量均在固定腿长的实车状态下测得, 对应的增益矩阵 K 如下所示。数据均在 J-Scope 中测得, 在 MATLAB 中绘制。

$$K = \begin{bmatrix} -3.693 & -3.287 & -8.559 & -0.591 & 3.693 & 0.369 \\ -3.693 & -3.287 & -8.559 & -0.591 & -3.693 & -0.369 \end{bmatrix}$$

4.1 平衡车位移、偏航角静态响应及俯仰角变化

在静止状态下, 给机器人位移期望输入幅值为0.4的阶跃信号, 其位移响应及俯仰角变化如图 8 所示。

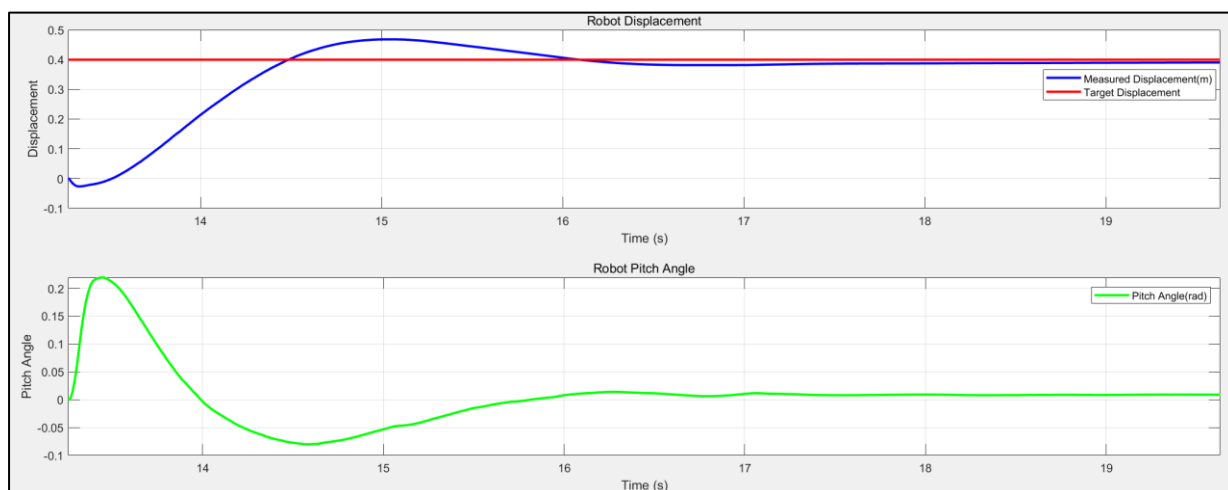


图 8 机器人位移阶跃响应及其过程俯仰角变化

经测量，位移响应的峰值时间 T_p 为：

$$T_p = 1.731s$$

超调量 $\sigma\%$ 为：

$$\sigma\% = \frac{A_{\max} - A_{\text{steady}}}{A_{\text{steady}}} \times 100\% = \frac{0.468 - 0.400}{0.4} \times 100\% = 17\%$$

调节时间 T_s 为（5%误差下）：

$$T_s = 2.542s$$

在位移状态量的轨迹跟踪过程中，测得机器人机体最大俯仰角为 0.219 rad ，即约 12.549 度。

在静止状态下，给机器人偏航角期望输入幅值为 1.579 （约 90 度）的阶跃信号，其偏航角响应及俯仰角变化如图 9 所示。

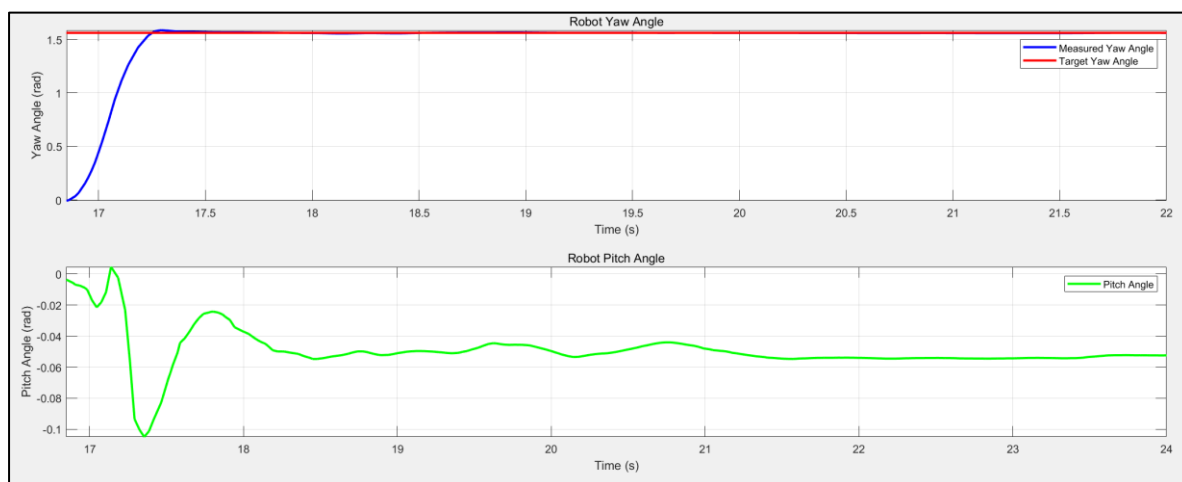


图 9 机器人偏航角阶跃响应及其过程俯仰角变化

经测量，位移响应的峰值时间 T_P 为：

$$T_P = 0.477s$$

超调量 $\sigma\%$ 为：

$$\sigma\% = \frac{A_{\max} - A_{\text{steady}}}{A_{\text{steady}}} \times 100\% = \frac{1.585 - 1.560}{1.560} \times 100\% = 1.6\%$$

调节时间 T_S 为（5%误差下）：

$$T_S = 0.385s$$

在偏航角状态量的轨迹跟踪过程中，测得机器人机体俯仰角在转动过程中会在 0.104 rad 内产生小幅度震荡。

综上，机器人位移响应速度较慢，考虑到测量过程中没有加入机器人速度状态的轨迹跟踪，且响应过程中机体倾角较小，实际响应效果可以满足项目需求。偏航角的响应十分迅速，且超调量小，对机体的俯仰角影响十分有限，也能满足项目需求。

4.2 平衡车腿长及机体横滚角响应

对于机器人腿长的控制，我们采用了双环 PID 控制器。对腿长目标值输入幅值为 0.3 的阶跃信号。其位置外环及速度内环的响应曲线如图 10 所示。

其中，该 PID 控制器的内外环参数如下所示：

位置环外环： $K_p = 100$, $K_i = 0.5$, $Integral_{\max} = 10$, $Out_{\max} = 30$;

速度环外环： $K_p = 0.4$, $K_i = 0.05$, $Integral_{\max} = 5$, $Out_{\max} = 8$;

基于关节电机的扭矩参数，额定扭矩为 $6N \cdot m$ ，峰值扭矩为 $16N \cdot m$ ，设置常态下速度环的最大速出为 8，即 $8N \cdot m$ 。

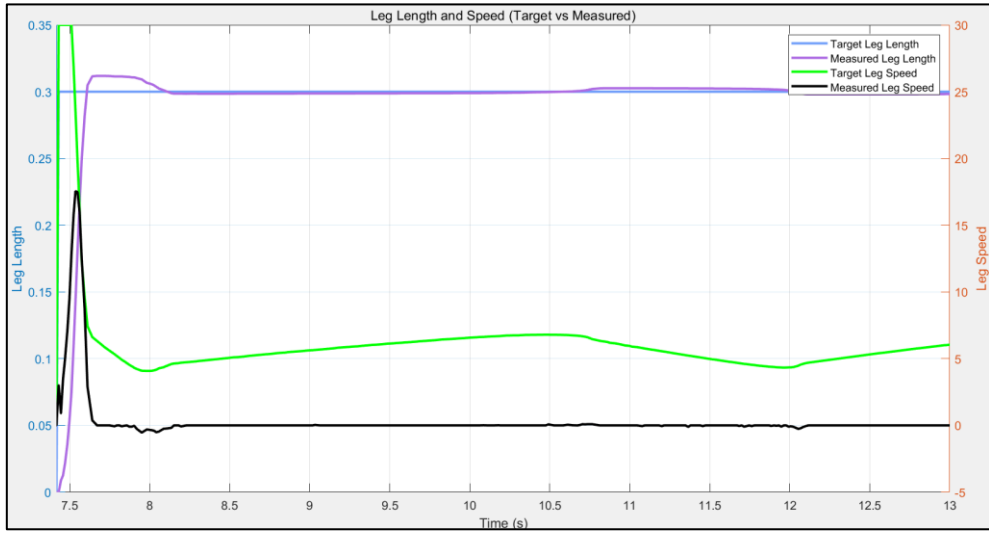


图 10 机器人腿长双环控制器的内外环响应曲线

由图 10 可以看到，速度响应在初始阶段有一段锯齿波形，这是由于机械的腿部涉及有一些干涉导致的。再次输入阶跃信号，其幅值为机器人的最高腿长，其响应曲线如图 11 所示。

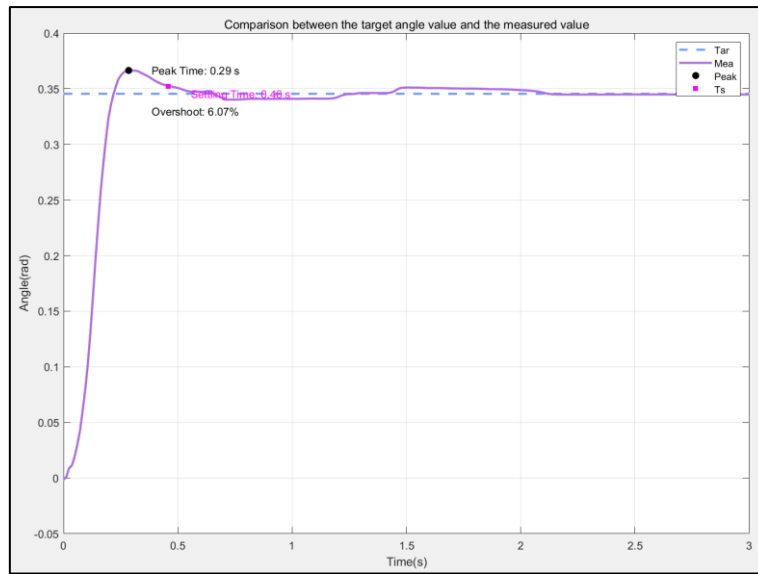


图 11 机器人腿长最大阶跃响应曲线

经测量，腿长响应的峰值时间 T_p 为：

$$T_p = 0.29s$$

超调量 $\sigma\%$ 为：

$$\sigma\% = \frac{A_{\max} - A_{\text{steady}}}{A_{\text{steady}}} \times 100\% = \frac{0.366 - 0.350}{0.350} \times 100\% = 6.07\%$$

调节时间 T_s 为（2%误差下）：

$$T_s = 0.46s$$

4.3 离地检测与跳跃动作

在机器人跳跃过程中,机器人整体完全离开地面,此时其模型与在地面时有差异,这将会导致系统发散。故高效的离地检测与离地处理十分重要。理论上,严谨的离地处理需要通过测得的关节输出扭矩来解算机器人实时受到的支持力,这将需要一定的推导计算。为了项目进度,我们将该流程简化,直接用关节输出扭矩来判断机器人是否离地。图 12 为机器人两次离地落地的关节扭矩输出,我们选取了离地阈值为 0.5Nm ,经多次测试,该方式检测离地较为可靠与高效。关于离地后的处理,我们将驱动轮的输出置零,并将位移和偏航角两个状态变量置零,以使得机器人在落地后保持初始状态。

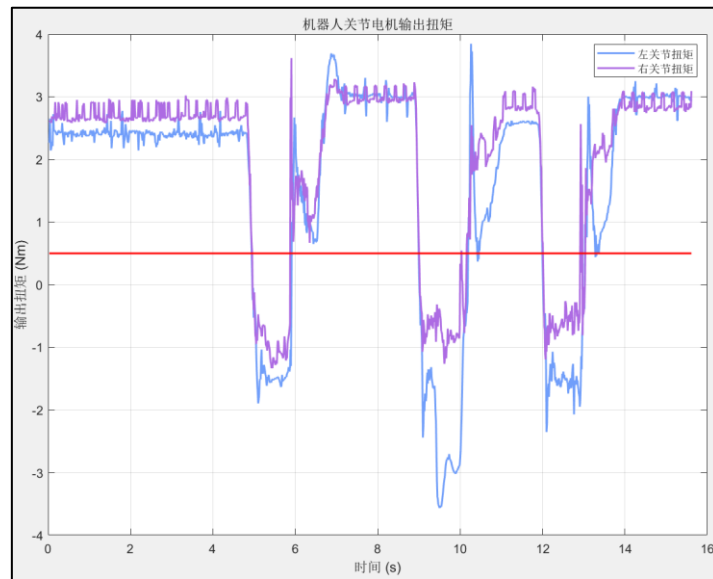


图 12 机器人两次离地落地时关节电机输出扭矩曲线

对于机器人跳跃动作的设计,我们通过给腿长期望输入一个幅值较大的脉冲信号实现。实车跳跃过程如图 13 所示,多次测试测得的平均跳跃高度约为 0.1m 。受限于机器人较大的机体重量,以目前使用的关节电机输出性能难以实现较高的跳跃高度,其跳跃性能有待提高。图 14 为机器人在执行两次跳跃动作时腿长控制器双环的输出曲线

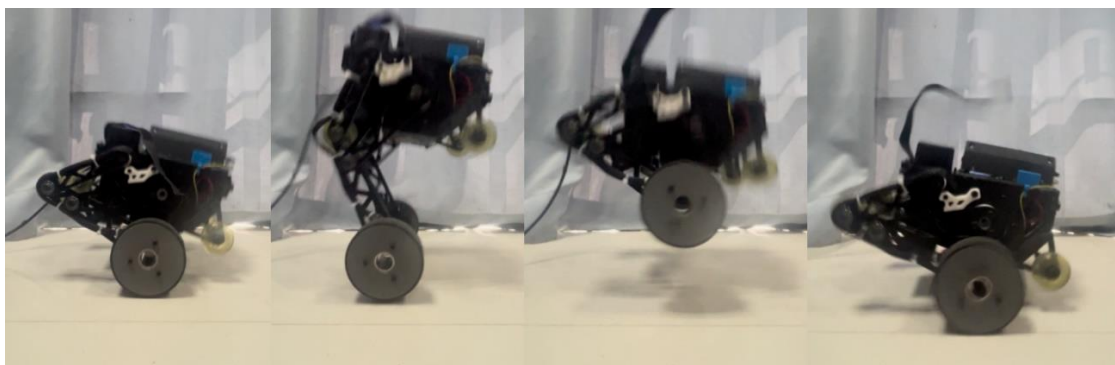


图 13 机器人跳跃过程

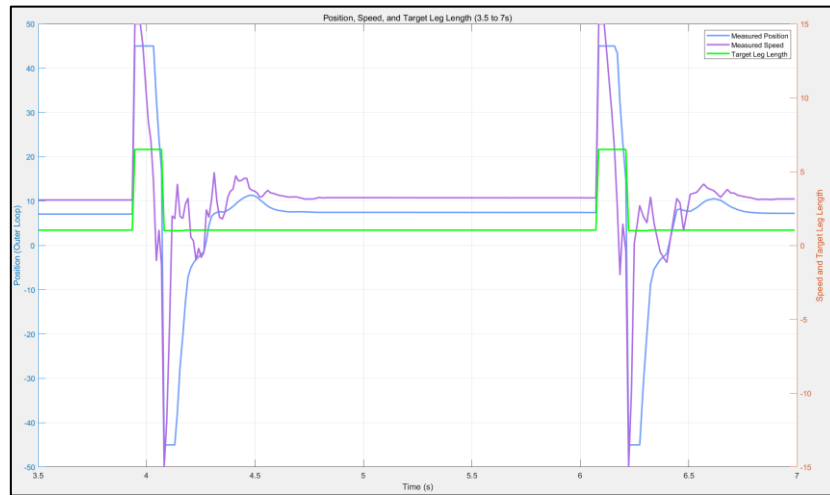


图 14 机器人两次跳跃动作过程内外环控制器的输出曲线