

# 山海机甲平衡步兵软件设计

与其他四轮兵种相比，平衡步兵具有高效率、高地形通过力等优势，但同时平衡步兵也具有远高于四轮兵种的控难度，对我们的控制算法提出了很高的要求。

从直觉上来看，我们很容易可以发现平衡步兵是一个MIMO系统，难以使用经典控制理论中的PID算法进行控制，效果也并不理想，取而代之，我们将采用现代控制理论中的LQR算法对单腿进行动力学建模，然后通过外加综合运动控制算法等将双腿结合起来构成完整的控制系统。

## 1 平衡步兵核心控制算法——LQR

LQR是一种针对线性定常系统的控制理论，现考虑线性定常系统的状态空间方程：

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu} \\ \mathbf{y} = \mathbf{Cx} + \mathbf{Du} \end{cases}$$

我们对此构造代价函数如下：

$$J = \frac{1}{2} \int_0^{\infty} (\mathbf{x}^T \mathbf{Qx} + \mathbf{u}^T \mathbf{Ru}) dt \quad (1-1)$$

一般来说， $\mathbf{Q}$ 阵和 $\mathbf{R}$ 阵为对角阵，分别表示状态向量 $\mathbf{x}$ 和控制向量 $\mathbf{u}$ 的权重。LQR实际上就是通过构造线性反馈器 $\mathbf{u} = -\mathbf{Kx}$ 对代价函数 $J$ 的一种最优控制。下面我们进行推导来说明 $\mathbf{K}$ 取何值时可以实现代价函数 $J$ 最小。

我们先引入一个常量矩阵 $\mathbf{P}$ ，使得：

$$\frac{d}{dt} \mathbf{x}^T \mathbf{Px} = -(\mathbf{x}^T \mathbf{Qx} + \mathbf{u}^T \mathbf{Ru}) \quad (1-2)$$

这一步的理论依据笔者也没有搞懂，貌似与哈密顿方程和变分法有关，不过这并不是我们讨论的重点。我们假设 $\lim_{t \rightarrow \infty} \mathbf{x} = \mathbf{0}$ ，这时我们可以将式(1-2)代入式(1-1)得到：

$$\begin{aligned} J &= \frac{1}{2} \int_0^{\infty} (\mathbf{x}^T \mathbf{Qx} + \mathbf{u}^T \mathbf{Ru}) dt \\ &= -\frac{1}{2} \int_0^{\infty} \left( \frac{d}{dt} \mathbf{x}^T \mathbf{Px} \right) dt \\ &= -\frac{1}{2} (\mathbf{x}^T \mathbf{Px} |_{\infty} - \mathbf{x}^T \mathbf{Px} |_0) \\ &= \frac{1}{2} \mathbf{x}^T(0) \mathbf{Px}(0) \end{aligned}$$

我们发现代价函数 $J$ 的大小仅与初始条件和常量矩阵 $\mathbf{P}$ 有关，只要让 $\mathbf{P}$ 最小即可得到最小的代价，下面推导常量矩阵 $\mathbf{P}$ 的约束条件，从式(1-2)入手代入 $\mathbf{u} = -\mathbf{Kx}$ 进行分析：

$$\begin{aligned} \frac{d}{dt} \mathbf{x}^T \mathbf{Px} + (\mathbf{x}^T \mathbf{Qx} + \mathbf{u}^T \mathbf{Ru}) &= 0 \\ \dot{\mathbf{x}}^T \mathbf{Px} + \mathbf{x}^T \mathbf{P} \dot{\mathbf{x}} + \mathbf{x}^T \mathbf{Qx} + \mathbf{x}^T \mathbf{K}^T \mathbf{R} \mathbf{Kx} &= 0 \end{aligned} \quad (1-3)$$

然后将 $\mathbf{u} = -\mathbf{Kx}$ 代入状态空间方程可以得到：

$$\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu} = \mathbf{Ax} - \mathbf{BKx} = (\mathbf{A} - \mathbf{BK})\mathbf{x} \quad (1-4)$$

将式(1-3)代入式(1-4)得到：

$$\begin{aligned} \dot{\mathbf{x}}^T \mathbf{Px} + \mathbf{x}^T \mathbf{P} \dot{\mathbf{x}} + \mathbf{x}^T \mathbf{Qx} + \mathbf{x}^T \mathbf{K}^T \mathbf{R} \mathbf{Kx} &= 0 \\ ((\mathbf{A} - \mathbf{BK})\mathbf{x})^T \mathbf{Px} + \mathbf{x}^T \mathbf{P}((\mathbf{A} - \mathbf{BK})\mathbf{x}) + \mathbf{x}^T \mathbf{Qx} + \mathbf{x}^T \mathbf{K}^T \mathbf{R} \mathbf{Kx} &= 0 \\ \mathbf{x}^T (\mathbf{A}^T - \mathbf{K}^T \mathbf{B}^T) \mathbf{Px} + \mathbf{x}^T \mathbf{P} (\mathbf{A} - \mathbf{BK}) \mathbf{x} + \mathbf{x}^T \mathbf{Qx} + \mathbf{x}^T \mathbf{K}^T \mathbf{R} \mathbf{Kx} &= 0 \\ \mathbf{x}^T [(\mathbf{A}^T - \mathbf{K}^T \mathbf{B}^T) \mathbf{P} + \mathbf{P} (\mathbf{A} - \mathbf{BK}) + \mathbf{Q} + \mathbf{K}^T \mathbf{R} \mathbf{K}] \mathbf{x} &= 0 \end{aligned}$$

上式对任意状态向量 $\mathbf{x}$ 均成立，于是我们可以得到：

$$\begin{aligned} (\mathbf{A}^T - \mathbf{K}^T \mathbf{B}^T) \mathbf{P} + \mathbf{P} (\mathbf{A} - \mathbf{BK}) + \mathbf{Q} + \mathbf{K}^T \mathbf{R} \mathbf{K} &= 0 \\ \mathbf{A}^T \mathbf{P} + \mathbf{P} \mathbf{A} + \mathbf{Q} + \mathbf{K}^T \mathbf{R} \mathbf{K} - \mathbf{K}^T \mathbf{B}^T \mathbf{P} - \mathbf{P} \mathbf{B} \mathbf{K} &= 0 \end{aligned} \quad (1-5)$$

我们已然得到了常量矩阵 $\mathbf{P}$ 的约束条件式(1-5)，于是问题转化为如何构造反馈矩阵 $\mathbf{K}$ 通过约束条件得到最小的 $\mathbf{P}$ ，通过复杂的数学推导（好像也涉及到哈密顿方程和变分法，笔者也没搞懂），我们将得到反馈矩阵 $\mathbf{K}$ 的方程：

$$\mathbf{K} = \mathbf{R}^{-1} \mathbf{B}^T \mathbf{P} \quad (1-6)$$

将式(1-6)代入式(1-5)得到：

$$\begin{aligned}
& \mathbf{A}^T \mathbf{P} + \mathbf{P} \mathbf{A} + \mathbf{Q} + \mathbf{K}^T \mathbf{R} \mathbf{K} - \mathbf{K}^T \mathbf{B}^T \mathbf{P} - \mathbf{P} \mathbf{B} \mathbf{K} = 0 \\
& \mathbf{A}^T \mathbf{P} + \mathbf{P} \mathbf{A} + \mathbf{Q} + (\mathbf{R}^{-1} \mathbf{B}^T \mathbf{P})^T \mathbf{R} (\mathbf{R}^{-1} \mathbf{B}^T \mathbf{P}) - (\mathbf{R}^{-1} \mathbf{B}^T \mathbf{P})^T \mathbf{B}^T \mathbf{P} - \mathbf{P} \mathbf{B} (\mathbf{R}^{-1} \mathbf{B}^T \mathbf{P}) = 0 \\
& \mathbf{A}^T \mathbf{P} + \mathbf{P} \mathbf{A} + \mathbf{Q} - \mathbf{P} \mathbf{B} \mathbf{R}^{-1} \mathbf{B}^T \mathbf{P} = 0
\end{aligned}$$

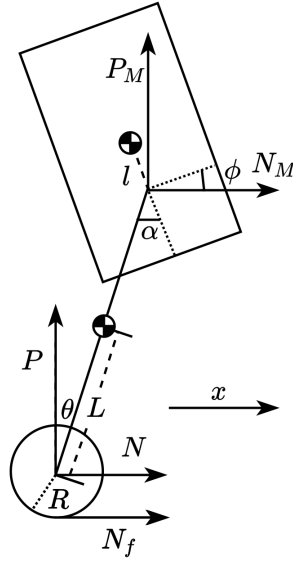
这时我们得到了一个连续时间代数黎卡提方程，通过数值解法即可得到常量矩阵 $\mathbf{P}$ ，再通过式(1-6)即可求解反馈矩阵 $\mathbf{K}$ 。

考虑到我们的控制器部署在嵌入式系统Robomaster C板上，应当使用离散系统无限时域的LQR算法，不过由于我们的步长很小，同时离散系统更加难以分析，我们仍然使用连续系统无限时域的LQR算法对平衡步兵进行控制。

## 2 平衡步兵的动力学建模与控制器设计

本赛季我们采用了哈工程建模的方法将双腿分开建模再通过其他控制器将双腿耦合在一起，现在看来，上交的建模方法可能更好一点。

建模的主要思路就是将每条腿简化为一个二阶倒立摆模型进行分析，如下图：



其中，

符号	含义	正方向	单位	符号	含义	单位
$\theta$	摆杆与竖直方向的夹角	顺时针	rad	$R$	驱动轮半径	m
$x$	驱动轮位移	向右	m	$L$	摆杆重心到驱动轮轴线的距离	m
$\phi$	机体与水平方向的夹角	逆时针	rad	$L_M$	摆杆重心到机体转轴的距离	m
$T$	轮向电机的输出力矩	右手螺旋向内（顺时针）	$N \cdot m$	$l$	机体重心到机体转轴的距离	m
$T_p$	髋关节电机等效输出力矩	右手螺旋向内（顺时针）	$N \cdot m$	$m_w$	驱动轮质量（不含电机）	kg
$N$	驱动轮对摆杆的力的水平分量	向右	N	$m_p$	摆杆质量（含轮向电机不含髋关节电机）	kg
$P$	驱动轮对摆杆的力的竖直分量	向上	N	$M$	机体质量（含髋关节电机）	kg
$N_M$	摆杆对机体的力的水平分量	向右	N	$I_w$	驱动轮转子的转动惯量	$kg \cdot m^2$
$P_M$	摆杆对机体的力的竖直分量	向上	N	$I_p$	摆杆绕质心的转动惯量	$kg \cdot m^2$
$N_f$	地面对驱动轮的摩擦力	向右	N	$I_M$	机体绕质心的转动惯量	$kg \cdot m^2$

### 2.1 经典力学建模分析

#### 2.1.1 驱动轮动力学分析

驱动轮处于惯性系，我们直接基于大地系进行受力分析（牛顿第二定律+转动定律）：

$$\begin{cases} m_w a = m_w \ddot{x} = N_f - N \\ I_w \beta = I_w \frac{\ddot{x}}{R} = T - N_f R \end{cases} \quad (2-1-1)$$

注意，轮向电机的定子安装在摆杆上，转子安装在驱动轮上，当输出力矩为 $T$ 时，产生的效果是驱动轮有顺时针转动趋势，摆杆有逆时针转动趋势。同理，髋关节的定子安装在机体上，转子安装在摆杆上，当等效输出力矩为 $T_p$ 时，产生的效果是摆杆有顺时针转动趋势，机体有逆时针转动趋势。

然后我们通过式(2-1-1)合并消去 $N_f$ 得到：

$$\ddot{x} = \frac{T - NR}{\frac{I_w}{R} + m_w R} \quad (2-1-2)$$

### 2.1.2 摆杆动力学分析

由于摆杆处于非惯性系，我们建立摆杆的质心系进行力学分析，有质心坐标：

$$\begin{cases} x_c = x + L \sin\theta \\ y_c = L \cos\theta \end{cases} \quad (2-1-3)$$

基于式(2-1-3)我们可以写出摆杆基于质心系的牛顿第二定律和转动定律公式：

$$\begin{cases} m_p a_{cx} = m_p \ddot{x}_c = m_p \frac{\partial^2}{\partial t^2} (x + L \sin\theta) = N - N_M \\ m_p a_{cy} = m_p \ddot{y}_c = m_p \frac{\partial^2}{\partial t^2} (L \cos\theta) = P - P_M - m_p g \\ I_p \ddot{\theta} = (PL \sin\theta + P_M L_M \sin\theta) - (NL \cos\theta + N_M L_M \cos\theta) + T_p - T = (PL + P_M L_M) \sin\theta - (NL + N_M L_M) \cos\theta + T_p - T \end{cases} \quad (2-1-4)$$

### 2.1.3 机体动力学分析

机体同样处于非惯性系，我们依然建立机体质心系进行力学分析，有质心坐标：

$$\begin{cases} x_c = x + (L + L_M) \sin\theta - l \sin\phi \\ y_c = (L + L_M) \cos\theta + l \cos\phi \end{cases} \quad (2-1-5)$$

基于式(2-1-5)我们可以写出机体基于质心系的牛顿第二定律和转动定律公式：

$$\begin{cases} M a_{cx} = M \ddot{x}_c = M \frac{\partial^2}{\partial t^2} (x + (L + L_M) \sin\theta - l \sin\phi) = N_M \\ M a_{cy} = M \ddot{y}_c = M \frac{\partial^2}{\partial t^2} ((L + L_M) \cos\theta + l \cos\phi) = P_M - Mg \\ I_M \ddot{\phi} = P_M l \sin\phi + N_M l \cos\phi + T_p \end{cases} \quad (2-1-6)$$

### 2.1.4 动力学模型

整合式(2-1-2)、式(2-1-4)和式(2-1-6)我们就得到了平衡步兵单腿的动力学模型：

$$\left\{ \begin{array}{l} \text{驱动轮:} \quad \ddot{x} = \frac{T - NR}{\frac{I_w}{R} + m_w R} \\ \\ \text{摆杆:} \quad \begin{aligned} N - N_M &= m_p \frac{\partial^2}{\partial t^2} (x + L \sin\theta) \\ P - P_M - m_p g &= m_p \frac{\partial^2}{\partial t^2} (L \cos\theta) \\ I_p \ddot{\theta} &= (PL + P_M L_M) \sin\theta - (NL + N_M L_M) \cos\theta - T + T_p \end{aligned} \\ \\ \text{机体:} \quad \begin{aligned} N_M &= M \frac{\partial^2}{\partial t^2} (x + (L + L_M) \sin\theta - l \sin\phi) \\ P_M - Mg &= M \frac{\partial^2}{\partial t^2} ((L + L_M) \cos\theta + l \cos\phi) \\ I_M \ddot{\phi} &= T_p + N_M l \cos\phi + P_M l \sin\phi \end{aligned} \end{array} \right. \quad (2-1-7)$$

## 2.2 控制器设计

定义状态向量和控制向量为：

$$\mathbf{x} = \begin{bmatrix} \theta \\ \dot{\theta} \\ x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} \quad \mathbf{u} = \begin{bmatrix} T \\ T_p \end{bmatrix}$$

回看式(2-1-7), 有变量  $m_w, m_p, M, I_w, I_p, I_M, R, L, L_M, l, N, N_M, P, P_M, \theta, \dot{\theta}, \ddot{\theta}, \ddot{x}, \phi, \dot{\phi}, \ddot{\phi}, T, T_p$  共23个, 已知变量  $m_w, m_p, M, I_w, I_p, I_M, R, L, L_M, l, \mathbf{x}, \mathbf{u}$  共16个, 共7个未知量  $N, N_M, P, P_M, \ddot{\theta}, \ddot{x}, \ddot{\phi}$ , 同时动力学模型也是7个方程, 可解。其实对于我们构造控制器来说, 只需要解出  $\ddot{\theta}, \ddot{x}, \ddot{\phi}$  这三个变量即可, 借助MATLAB的符号运算, 我们通过动力学模型的第2 3 5 6方程可以得到  $N, N_M, P, P_M$  的表达式:

$$\begin{cases} N = m_p \frac{\partial^2 x}{\partial t^2} + M \left( \frac{\partial^2 x}{\partial t^2} + l \sin\phi \left( \frac{\partial \phi}{\partial t} \right)^2 - l \cos\phi \frac{\partial^2 \phi}{\partial t^2} - \sin\theta \left( \frac{\partial \theta}{\partial t} \right)^2 (L + L_M) + \cos\theta (L + L_M) \frac{\partial^2 \theta}{\partial t^2} \right) - L m_p \sin\theta \left( \frac{\partial \theta}{\partial t} \right)^2 + L m_p \cos\theta \frac{\partial^2 \theta}{\partial t^2} \\ N_M = M \left( \frac{\partial^2 x}{\partial t^2} + l \sin\phi \left( \frac{\partial \phi}{\partial t} \right)^2 - l \cos\phi \frac{\partial^2 \phi}{\partial t^2} - \sin\theta \left( \frac{\partial \theta}{\partial t} \right)^2 (L + L_M) + \cos\theta (L + L_M) \frac{\partial^2 \theta}{\partial t^2} \right) \\ P = Mg + g m_p - M \left( l \cos\phi \left( \frac{\partial \phi}{\partial t} \right)^2 + l \sin\phi \frac{\partial^2 \phi}{\partial t^2} + \cos\theta \left( \frac{\partial \theta}{\partial t} \right)^2 (L + L_M) + \sin\theta (L + L_M) \frac{\partial^2 \theta}{\partial t^2} \right) - L m_p \left( \sin\theta \frac{\partial^2 \theta}{\partial t^2} + \cos\theta \left( \frac{\partial \theta}{\partial t} \right)^2 \right) \\ P_M = Mg - M \left( l \cos\phi \left( \frac{\partial \phi}{\partial t} \right)^2 + l \sin\phi \frac{\partial^2 \phi}{\partial t^2} + \cos\theta \left( \frac{\partial \theta}{\partial t} \right)^2 (L + L_M) + \sin\theta (L + L_M) \frac{\partial^2 \theta}{\partial t^2} \right) \end{cases} \quad (2-2-1)$$

将式(2-2-1)代入动力学模型的第1 4 7方程即可联立求解  $\ddot{\theta}, \ddot{x}, \ddot{\phi}$ , 公式过于冗长, 就不写出来了。我们观察后发现其中存在大量三角函数, 是非线性的, 若要使用LQR, 我们需要在平衡位置附近近似线性化变成线性系统。

然后我们在进行一些简单分析, 在我们的7个方程中, 其实并没有出现  $x, \dot{x}$  这两项, 所以在最后得到的  $\ddot{\theta}, \ddot{x}, \ddot{\phi}$  的公式中也一定不会出现  $x, \dot{x}$ , 即  $\mathbf{A}_{23} = \mathbf{A}_{24} = \mathbf{A}_{43} = \mathbf{A}_{44} = \mathbf{A}_{63} = \mathbf{A}_{64} = 0$ , 在后面的近似线性化中也就没必要考虑  $x, \dot{x}$  了。其实对于  $\mathbf{A}$  阵和  $\mathbf{B}$  阵, 我们还能得到一些信息, 对于第1 3 5行易知:

$$\begin{aligned} \mathbf{A}_{11} = \mathbf{A}_{13} = \mathbf{A}_{14} = \mathbf{A}_{15} = \mathbf{A}_{16} = \mathbf{A}_{31} = \mathbf{A}_{32} = \mathbf{A}_{33} = \mathbf{A}_{35} = \mathbf{A}_{36} = \mathbf{A}_{51} = \mathbf{A}_{52} = \mathbf{A}_{53} = \mathbf{A}_{54} = \mathbf{A}_{55} = 0 \\ \mathbf{A}_{12} = \mathbf{A}_{34} = \mathbf{A}_{56} = 1 \\ \mathbf{B}_{11} = \mathbf{B}_{12} = \mathbf{B}_{31} = \mathbf{B}_{32} = \mathbf{B}_{51} = \mathbf{B}_{52} = 0 \end{aligned}$$

其实, 我们近似线性化的目的只是为了得到  $\mathbf{A}_{21}, \mathbf{A}_{22}, \mathbf{A}_{25}, \mathbf{A}_{26}, \mathbf{A}_{41}, \mathbf{A}_{42}, \mathbf{A}_{45}, \mathbf{A}_{46}, \mathbf{A}_{61}, \mathbf{A}_{62}, \mathbf{A}_{65}, \mathbf{A}_{66}$  和  $\mathbf{B}_{21}, \mathbf{B}_{22}, \mathbf{B}_{41}, \mathbf{B}_{42}, \mathbf{B}_{61}, \mathbf{B}_{62}$ , 下面我们同样借助MATLAB进行符号计算。

我们首先定义  $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$ , 通过上方的求解, 事实上我们已知了  $\mathbf{f}$  的具体形式, 我们可以通过雅可比矩阵进行近似线性化:

$$\mathbf{A} = \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_{\mathbf{x}=\bar{\mathbf{x}}, \mathbf{u}=\bar{\mathbf{u}}} \quad \mathbf{B} = \left. \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \right|_{\mathbf{x}=\bar{\mathbf{x}}, \mathbf{u}=\bar{\mathbf{u}}}$$

其中,

$$\bar{\mathbf{x}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \bar{\mathbf{u}} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

由此我们就可以得到LQR的最后一块拼图:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ A_{21} & 0 & 0 & 0 & A_{25} & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ A_{41} & 0 & 0 & 0 & A_{45} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ A_{61} & 0 & 0 & 0 & A_{65} & 0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 & 0 \\ B_{21} & B_{22} \\ 0 & 0 \\ B_{41} & B_{42} \\ 0 & 0 \\ B_{61} & B_{62} \end{bmatrix}$$

这下我们就可以通过MATLAB里面的LQR函数求解出  $\mathbf{K}$  阵, 进一步通过  $\mathbf{u} = -\mathbf{K}\mathbf{x}$  我们就可以得到在每一个状态下轮向电机和髋关节等效力矩应该是多大, 此时我们已经可以实现将一个二阶倒立摆保持平衡。

但是我们并不是一直都需要  $x = 0$ , 更多时候我们希望  $x$  是可以变化的, 我们可以通过下面的方式实现:

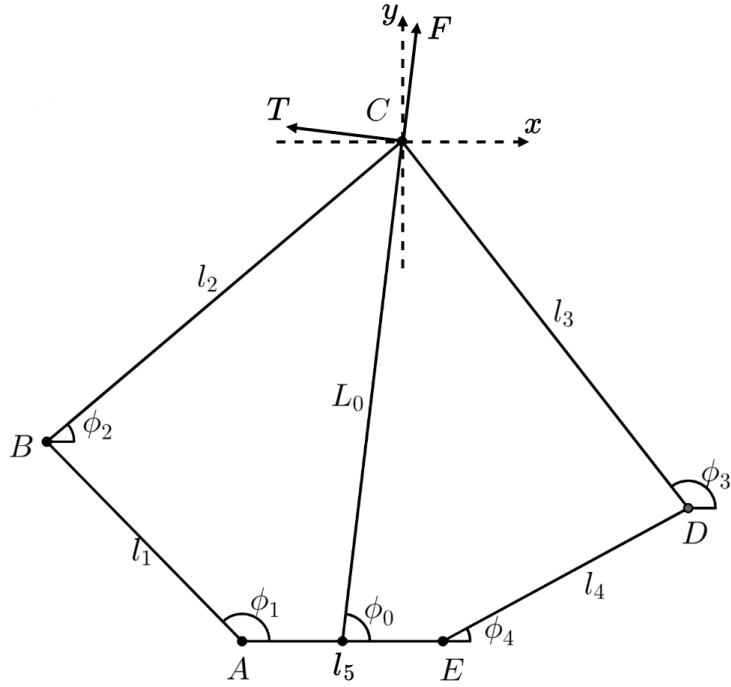
$$\mathbf{x}_d = \begin{bmatrix} 0 \\ 0 \\ x_{target} \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{u} = \mathbf{K}(\mathbf{x}_d - \mathbf{x})$$

最后, 我们还要讨论一个问题, 在前面求解方程的时候, 我们提到了有16个变量是已知的, 但其实其中有一个量  $L_0 = L + L_M$  是一个可知的变化量, 随着  $L_0$  的变化, 事实上  $\mathbf{K}$  阵也会随之变化, 虽然变化不是特别大, 但会有略微影响, 我们可以通过曲线拟合获取  $\mathbf{K}(L_0)$  来解决这个问题, 对此我们有经验公式:

$$\mathbf{K}(L_0) = \mathbf{p}_0 + \mathbf{p}_1 L_0 + \mathbf{p}_2 L_0^2 + \mathbf{p}_3 L_0^3$$

### 3 五连杆机构的运动学与动力学变换

在我们建模的过程中，将五连杆腿部机构简化为了一根摆杆，得到了髋关节等效输出力矩，但是我们目前仍然无法确定两个关节电机的输出力矩分别是多少，于是我们引入了五连杆机构的运动学和动力学变换，运动学变换用来确定 $L_0$   $\phi_0$ 与 $\phi_1$   $\phi_4$ 之间的关系，动力学变换用来确定 $F$   $T_p$ 与 $T_1$   $T_2$ 之间的关系，具体符号说明参见下图（A点为坐标原点），公式推导参考了玺佬的开源：



#### 3.1 运动学变换

##### 3.1.1 正运动学变换

在正问题中，我们已知 $\phi_1$   $\phi_4$ ，通过几何学得C点坐标：

$$\begin{cases} x_C = x_B + l_2 \cos\phi_2 = x_D + l_3 \cos\phi_3 \\ y_C = y_B + l_2 \sin\phi_2 = y_D + l_3 \sin\phi_3 \end{cases} \quad (3-1-1)$$

其中，

$$\begin{cases} x_B = l_1 \cos\phi_1 & x_D = l_5 + l_4 \cos\phi_4 \\ y_B = l_1 \sin\phi_1 & y_D = l_4 \sin\phi_4 \end{cases}$$

在式(3-1-1)中求解出 $\phi_2$   $\phi_3$ ：

$$\begin{aligned} \phi_2 &= 2 \operatorname{arctg} \left( \frac{B_0 + \sqrt{A_0^2 + B_0^2 - C_0^2}}{A_0 + C_0} \right) \\ \phi_3 &= \operatorname{arctg} \left( \frac{y_B - y_D + l_2 \sin\phi_2}{x_B - x_D + l_2 \cos\phi_2} \right) \end{aligned} \quad (3-1-2)$$

其中，

$$\begin{cases} A_0 = 2l_2(x_D - x_B) \\ B_0 = 2l_2(y_D - y_B) \\ C_0 = l_2^2 + l_{BD}^2 - l_3^2 \\ l_{BD}^2 = (x_D - x_B)^2 + (y_D - y_B)^2 \end{cases}$$

将式(3-1-2)中的 $\phi_2$ 代入式(3-1-1)得到 $x_C$ 和 $y_C$ ，然后通过下式即可得到 $L_0$ 和 $\phi_0$ ：

$$\begin{cases} L_0 = \sqrt{\left(x_C - \frac{l_5}{2}\right)^2 + y_C^2} \\ \phi_0 = \operatorname{arctg} \left( \frac{y_C}{x_C - \frac{l_5}{2}} \right) \end{cases} \quad (3-1-3)$$

由于后面观测器设计中还会使用到 $\dot{L}_0$   $\dot{\phi}_0$ ，这里通过MATLAB进行微分：

$$\begin{cases} \dot{L}_0 = \frac{y_C(\dot{y}_B + A_1 \cos\phi_2) + (x_C - \frac{l_5}{2})(\dot{x}_B - A_1 \sin\phi_2)}{L_0} \\ \dot{\phi}_0 = \frac{(x_C - \frac{l_5}{2})(\dot{y}_B + A_1 \cos\phi_2) - y_C(\dot{x}_B - A_1 \sin\phi_2)}{L_0^2} \end{cases} \quad (3-1-4)$$

其中,

$$\begin{cases} A_1 = \frac{l_1\dot{\phi}_1 \sin(\phi_1 - \phi_3) + l_4\dot{\phi}_4 \sin(\phi_3 - \phi_4)}{\sin(\phi_3 - \phi_2)} \\ \dot{x}_B = -l_1\dot{\phi}_1 \sin\phi_1 \\ \dot{y}_B = l_1\dot{\phi}_1 \cos\phi_1 \end{cases}$$

其中,  $\dot{\phi}_1$   $\dot{\phi}_4$ 可以通过髋关节电机编码器读取。

### 3.1.2 逆运动学变换

在逆问题中, 我们已知 $L_0$   $\phi_0$ , 可以得到C点坐标:

$$\begin{cases} x_C = \frac{l_5}{2} + L_0 \cos\phi_0 \\ y_C = L_0 \sin\phi_0 \end{cases}$$

然后依旧通过几何关系求解:

$$\begin{cases} x_C = x_B + l_2 \cos\phi_2 = l_1 \cos\phi_1 + l_2 \cos\phi_2 \\ y_C = y_B + l_2 \sin\phi_2 = l_1 \sin\phi_1 + l_2 \sin\phi_2 \end{cases}$$

两个方程两个未知量, 直接MATLAB暴力求解, 舍弃无效解:

$$\phi_1 = 2 \arctg \left( \frac{2l_1 y_C + \sqrt{2l_1^2 D + 2x_C^2 C - B^2 - C^2}}{A^2 - C} \right)$$

其中,

$$\begin{cases} A = l_1 + x_C \\ B = l_1^2 - x_C^2 \\ C = l_2^2 - y_C^2 \\ D = l_2^2 + y_C^2 \end{cases}$$

同理有:

$$\begin{cases} x_C = x_D + l_3 \cos\phi_3 = l_5 + l_4 \cos\phi_4 + l_3 \cos\phi_3 \\ y_C = y_D + l_3 \sin\phi_3 = l_4 \sin\phi_4 + l_3 \sin\phi_3 \end{cases}$$

依然直接MATLAB暴力求解, 舍弃无效解:

$$\phi_4 = 2 \arctg \left( \frac{2l_4 y_C - \sqrt{EF}}{D^2 + y_C^2 - l_3^2} \right)$$

其中,

$$\begin{cases} A = l_3 + l_4 \\ B = l_5 - x_C \\ C = l_3 - l_4 \\ D = x_C + l_4 - l_5 \\ E = A^2 - B^2 - y_C^2 \\ F = B^2 - C^2 + y_C^2 \end{cases}$$

注意,  $\phi_1$ 和 $\phi_4$ 都需要分别归一化到域值 $[0, 2\pi]$ 和 $[-\pi, \pi]$ 。

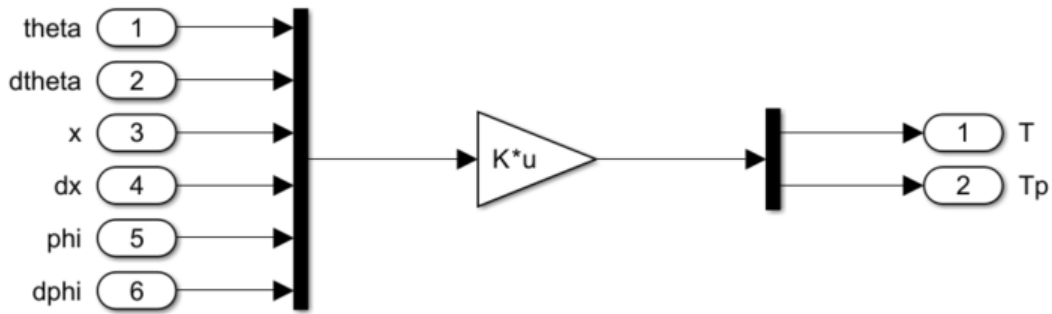
## 3.2 动力学变换 (VMC)

### 3.2.1 正动力学变换

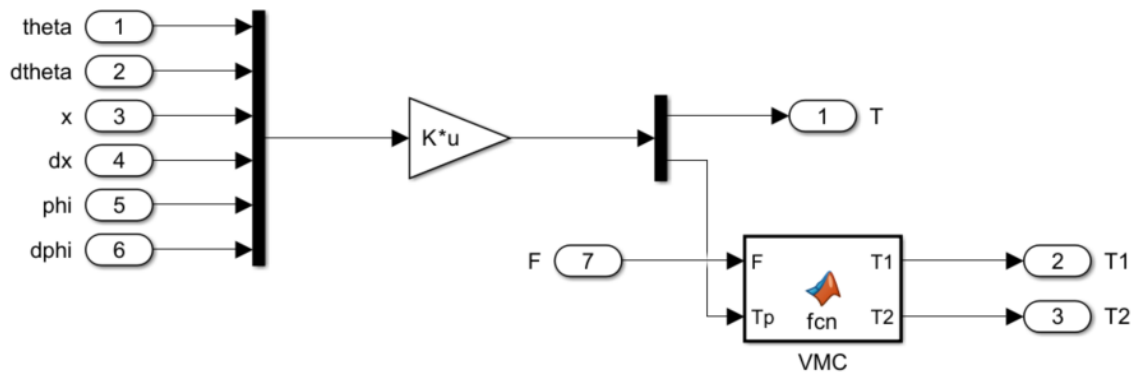
我们观察上图, 其实摆杆就是 $L_0$ 那个杆, 正动力学变换就是已知沿摆杆的推力 $F$ 和垂直于摆杆的力矩 $T_p$  (实际方向是右手螺旋向外), 得到两个髋关节电机分别的输出力矩 $T_1$   $T_2$  (方向是右手螺旋向外)。定义 $\mathbf{x} = [L_0 \ \phi_0]^T$ 和 $\mathbf{q} = [\phi_1 \ \phi_4]^T$ , 同时定义正运动学模型式(3-1-3)为 $\mathbf{x} = f(\mathbf{q})$ , 下面通过虚功原理来求解动力学变换矩阵, 沿摆杆的推力 $F$ 和垂直于摆杆的力矩 $T_p$ 产生的虚功之和应该与两个髋关节电机输出力矩产生的虚功之和相等:

$$\begin{aligned} \delta W_F + \delta W_{T_p} &= \delta W_{T_1} + \delta W_{T_2} \\ F\delta L_0 + T_p\delta\phi_0 &= T_1\delta\phi_1 + T_2\delta\phi_4 \end{aligned} \quad (3-2-1)$$

这里我们发现这里 $T_p$ 的方向与我们建模的时候相反，对于这一点我们进行一定的说明，先说明在我们设计控制器的时候是将状态变量通过增益 $\mathbf{K}$ 再反相，而非直接通过增益 $-\mathbf{K}$ ，如下图所示：



而我们的动力学变换VMC是连接在这个 $T_p$ 后面，如图：



在这里我们就会发现输入到VMC的 $T_p$ 通过反相才会变成控制向量中的 $T_p$ ，于是这里的 $T_p$ 就是与建模时的方向相反。

说明了正方向的问题，我们接着来求解动力学变换矩阵，我们接着定义 $\mathbf{F} = [F \ T_p]^T$ 和 $\mathbf{T} = [T_1 \ T_2]^T$ ，用矩阵描述式(3-2-1)有：

$$\mathbf{F}^T \delta \mathbf{x} = \mathbf{T}^T \delta \mathbf{q} \quad (3-2-2)$$

要得到 $\mathbf{F}$   $\mathbf{T}$ 的变换矩阵，我们就要消除掉 $\delta \mathbf{x}$   $\delta \mathbf{q}$ ，我们通过雅可比矩阵就可以解决这个问题：

$$\delta \mathbf{x} = \mathbf{J} \delta \mathbf{q} = \begin{bmatrix} \frac{\partial f_1}{\partial \phi_1} & \frac{\partial f_1}{\partial \phi_4} \\ \frac{\partial f_2}{\partial \phi_1} & \frac{\partial f_2}{\partial \phi_4} \end{bmatrix} \delta \mathbf{q} \quad (3-2-3)$$

将式(3-2-3)代入式(3-2-2)有：

$$\begin{aligned} \mathbf{F}^T \mathbf{J} \delta \mathbf{q} &= \mathbf{T}^T \delta \mathbf{q} \\ \mathbf{T}^T &= \mathbf{F}^T \mathbf{J} \\ \mathbf{T} &= \mathbf{J}^T \mathbf{F} \end{aligned}$$

我们已然得到了动力学变换矩阵，但通过正运动学来求解 $\mathbf{J}$ 并不容易而且公式冗长，玺佬提出了更好的求解方法，我们知道 $[L_0 \ \phi_0]^T$ 和 $[x_C \ y_C]^T$ 具有变换关系，同时 $[x_C \ y_C]^T$ 与 $[\phi_1 \ \phi_4]^T$ 的变换关系更加简单，使用 $[x_C \ y_C]^T$ 推导动力学变换矩阵将大大简化公式，我们还是先通过虚功原理将 $\mathbf{F}$   $\mathbf{T}$ 联系在一起，与原来不同，对于 $[x_C \ y_C]^T$ ，我们定义 $\mathbf{x}_C = [x_C \ y_C]^T$ ， $\mathbf{F}_{tc} = [F_t \ F_c]^T$ 和 $\mathbf{F}_{xy} = [F_x \ F_y]^T$ ，其中 $F_t$   $F_c$ 分别是摆杆推力 $F$ 和髋关节等效力矩 $T_p$ 产生的垂直摆杆方向的力（向右为正方向）和沿摆杆方向的力（向上为正方向）， $F_x$   $F_y$ 是 $F_t$   $F_c$ 正交分解之后的 $x$ 方向分量和 $y$ 方向分量。通过定义，我们可以得到如下的变换关系：

$$\mathbf{F}_{tc} = \mathbf{M}\mathbf{F} = \begin{bmatrix} 0 & -\frac{1}{L_0} \\ 1 & 0 \end{bmatrix} \mathbf{F} \quad \mathbf{F}_{xy} = \mathbf{R}\mathbf{F}_{tc} = \begin{bmatrix} \cos(\phi_0 - \frac{\pi}{2}) & -\sin(\phi_0 - \frac{\pi}{2}) \\ \sin(\phi_0 - \frac{\pi}{2}) & \cos(\phi_0 - \frac{\pi}{2}) \end{bmatrix} \mathbf{F}_{tc} \quad (3-2-4)$$

下面我们进行虚功原理：

$$\begin{aligned} \delta W_{F_x} + \delta W_{F_y} &= \delta W_{T_1} + \delta W_{T_2} \\ F_x \delta x_C + F_y \delta y_C &= T_1 \delta \phi_1 + T_2 \delta \phi_4 \\ \mathbf{F}_{xy}^T \delta \mathbf{x}_C &= \mathbf{T}^T \delta \mathbf{q} \end{aligned}$$

对于 $\delta \mathbf{x}_C$   $\delta \mathbf{q}$ 的变换矩阵的求解就简单多了，对式(3-1-1)求导有：

$$\begin{cases} \dot{x}_B - l_2 \dot{\phi}_2 \sin \phi_2 = \dot{x}_D - l_3 \dot{\phi}_3 \sin \phi_3 \\ \dot{y}_B + l_2 \dot{\phi}_2 \cos \phi_2 = \dot{y}_D + l_3 \dot{\phi}_3 \cos \phi_3 \end{cases}$$

从中可以解得：

$$\dot{\phi}_2 = \frac{(\dot{x}_D - \dot{x}_B) \cos \phi_3 + (\dot{y}_D - \dot{y}_B) \sin \phi_3}{l_2 \sin(\phi_3 - \phi_2)}$$

其中，

$$\begin{cases} \dot{x}_D = -l_4 \dot{\phi}_4 \sin \phi_4 \\ \dot{y}_D = l_4 \dot{\phi}_4 \cos \phi_4 \end{cases}$$

这下我们就可以得到 $\delta \mathbf{x}_C$ 了：

$$\begin{cases} \dot{x}_C = -l_1 \dot{\phi}_1 \sin \phi_1 - l_2 \left( \frac{(\dot{x}_D - \dot{x}_B) \cos \phi_3 + (\dot{y}_D - \dot{y}_B) \sin \phi_3}{l_2 \sin(\phi_3 - \phi_2)} \right) \sin \phi_2 \\ \dot{y}_C = l_1 \dot{\phi}_1 \cos \phi_1 + l_2 \left( \frac{(\dot{x}_D - \dot{x}_B) \cos \phi_3 + (\dot{y}_D - \dot{y}_B) \sin \phi_3}{l_2 \sin(\phi_3 - \phi_2)} \right) \cos \phi_2 \end{cases}$$

这下我们再进行雅可比矩阵操作，结果相当简洁：

$$\delta \mathbf{x}_C = \mathbf{J}' \delta \mathbf{q} = \begin{bmatrix} \frac{l_1 \sin(\phi_1 - \phi_2) \sin \phi_3}{\sin(\phi_2 - \phi_3)} & \frac{l_4 \sin(\phi_3 - \phi_4) \sin \phi_2}{\sin(\phi_2 - \phi_3)} \\ -\frac{l_1 \sin(\phi_1 - \phi_2) \cos \phi_3}{\sin(\phi_2 - \phi_3)} & -\frac{l_4 \sin(\phi_3 - \phi_4) \cos \phi_2}{\sin(\phi_2 - \phi_3)} \end{bmatrix} \delta \mathbf{q} \quad (3-2-5)$$

其中， $\phi_2$   $\phi_3$ 可由式(3-1-2)得到。将式(3-2-5)代入虚功原理得到：

$$\begin{aligned} \mathbf{F}_{xy}^T \mathbf{J}' \delta \mathbf{q} &= \mathbf{T}^T \delta \mathbf{q} \\ \mathbf{T} &= \mathbf{J}'^T \mathbf{F}_{xy} = \mathbf{J}'^T \mathbf{R} \mathbf{F}_{tc} = \mathbf{J}'^T \mathbf{R} \mathbf{M} \mathbf{F} \end{aligned}$$

于是我们只需要解出 $\mathbf{J}'^T \mathbf{R} \mathbf{M}$ 即可，通过MATLAB得到：

$$\mathbf{V} = \mathbf{J}'^T \mathbf{R} \mathbf{M} = \begin{bmatrix} \frac{l_1 \sin(\phi_0 - \phi_3) \sin(\phi_1 - \phi_2)}{\sin(\phi_3 - \phi_2)} & \frac{l_1 \cos(\phi_0 - \phi_3) \sin(\phi_1 - \phi_2)}{L_0 \sin(\phi_3 - \phi_2)} \\ \frac{l_4 \sin(\phi_0 - \phi_2) \sin(\phi_3 - \phi_4)}{\sin(\phi_3 - \phi_2)} & \frac{l_4 \cos(\phi_0 - \phi_2) \sin(\phi_3 - \phi_4)}{L_0 \sin(\phi_3 - \phi_2)} \end{bmatrix} \quad (3-2-6)$$

### 3.2.2 逆动力学变换

对于逆变换，我们直接通过MATLAB求解 $\mathbf{V}$ 的逆矩阵：

$$inv(\mathbf{V}) = \begin{bmatrix} -\frac{\cos(\phi_0 - \phi_2)}{l_1 \sin(\phi_1 - \phi_2)} & \frac{\cos(\phi_0 - \phi_3)}{l_4 \sin(\phi_3 - \phi_4)} \\ \frac{L_0 \sin(\phi_0 - \phi_2)}{l_1 \sin(\phi_1 - \phi_2)} & -\frac{L_0 \sin(\phi_0 - \phi_3)}{l_4 \sin(\phi_3 - \phi_4)} \end{bmatrix} \quad (3-2-7)$$

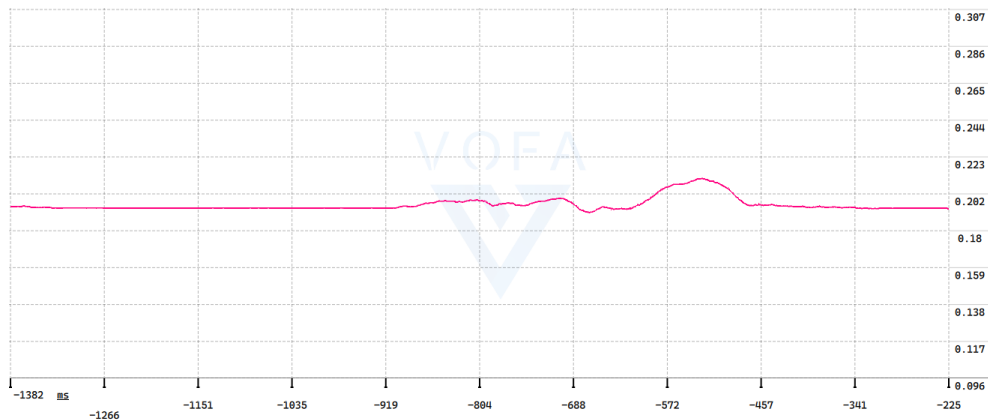
## 4 平衡步兵的综合运动控制

### 4.1 腿长控制

我们的腿长控制分为两种形式，一是用于正常运行二阶倒立摆模型，一是用于倒地自起和一阶倒立摆模型。

用于二阶倒立摆模型的腿长控制采用奎佬的方案，对于 $L_0$ 通过前馈+PD进行控制，输出推力 $F$ 输入到VMC中。其中前馈用于抵抗机体重量，有 $F_{ff} = Mg/\cos\theta$ ，PD控制用来产生阻尼弹簧效果，达到近乎完美的避震器悬挂效果。

实测效果波形，变化幅度仅有 $\pm 1cm$ ：



用于倒地自起和一阶倒立摆模型的腿长控制采用逆运动解算+前馈+LQR进行控制，因为我们在使用这个腿长控制时，是希望 $\phi_0 = \frac{\pi}{2}$ 的，可以直接通过目标腿长进行逆运动解算得到 $\phi_1 \sim \phi_4$ ，然后对髋关节电机角度环进行建模，近似为匀质单摆模型。

假设摆杆与水平夹角为 $\alpha$ ，逆时针为正，髋关节力矩同样逆时针为正，有：

$$\begin{cases} T - mg\frac{l}{2} \cos\alpha = J\ddot{\alpha} \\ J = \frac{1}{3}ml^2 \end{cases}$$

这里，我们取 $l_1$ 或 $l_4$ 那个杆的质量和长度进行近似。

取状态向量 $\mathbf{x} = [\alpha \quad \dot{\alpha}]^T$ ，控制向量 $\mathbf{u} = [T]^T$ ，于是有：

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\alpha} \\ \frac{T - mg\frac{l}{2} \cos\alpha}{J} \end{bmatrix}$$

通过雅可比矩阵近似线性化得到：

$$A_{21} = \left. \frac{\partial f_2}{\partial \alpha} \right|_{\mathbf{x}=[0 \ 0]^T, \mathbf{u}=[0]^T} = \left. \frac{mgl\dot{\alpha} \cos\alpha}{2J} \right|_{\mathbf{x}=[0 \ 0]^T, \mathbf{u}=[0]^T} = 0$$

最后得到状态空间方程：

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ \frac{1}{J} \end{bmatrix} \mathbf{u}$$

然后LQR求解 $\mathbf{K}$ 阵即可，实际上，这样的做法和微分先行的PD控制没有区别，只是参数可能更加合理一点。同样的这个PD控制仍然需要加入前馈用于抵抗机体重量，具体的前馈值可以通过VMC计算，我们直接令 $T_p = 0$ ，然后将推力前馈值输入到VMC就可以得到当前两个关节电机的扭矩前馈了。

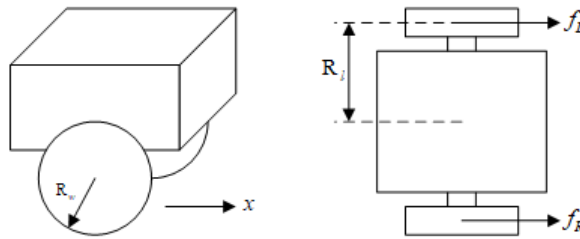
## 4.2 双腿协调

我们在建模时， $\mathbf{x}$ 是单腿驱动轮的位移，但实际使用时我们分别通过左右腿得到机体相对地面的速度取平均得到 $\dot{\mathbf{x}}$ ，然后对时间积分得到 $\mathbf{x}$ ，用机体的位移代替单腿驱动轮的位移，所以 $\mathbf{x}$ 是由两条腿共同得到的，具体方法参照[2.3.4.5 运动估计与打滑补偿](#)，我们设想这样的情景，如果左腿向前行进了10cm，右腿向后行进了10cm，这时 $\mathbf{x} = 0$ 位移项不会对控制器产生效果，这并不合理，于是我们引入了双腿协调。

双腿协调实际上就是一个PD控制器，控制量是左右腿的角度差 $\Delta\theta = \theta_r - \theta_l$ ，然后将输出力矩以相反的符号叠加到左右腿的髋关节等效力矩 $T_p$ 上以保证左右腿的同步。

## 4.3 转向控制

在玺佬的开源中，转向控制是直接使用了PD控制，控制量为偏航角，输出力矩以相反的符号叠加到左右轮向电机的输出力矩上，笔者在测试时参数一直调不好，效果不佳，后来笔者对转向进行了动力学建模，希望通过LQR获取合理的参数。我们首先对模型进行简化，假定机体与摆杆始终保持平稳，近似将二阶倒立摆约化为一阶倒立摆模型：



符号	含义	正方向	单位	符号	含义	单位
$\phi$	偏航角	逆时针	rad	$R_w$	驱动轮半径	m
$x_L \ x_R$	左右驱动轮位移	向右	m	$R_l$	半轮距	m
$f_L \ f_R$	地面对左右驱动轮的摩擦力	向右	N	$I_w$	驱动轮转子的转动惯量	kg·m <sup>2</sup>
$T_L \ T_R$	左右轮向电机的输出力矩	右手螺旋向内（顺时针）	N·m	$I$	机身绕偏航角旋转轴的转动惯量	kg·m <sup>2</sup>

### 4.3.1 单环LQR偏航角控制

建立运动学和动力学方程：

$$\left\{ \begin{array}{l} \text{运动学方程: } \frac{\dot{x}_R - \dot{x}_L}{2} = \dot{\phi} R_l \\ \text{动力学方程: } (f_R - f_L) R_l = I \ddot{\phi} \\ T_R - f_R R_w = I_w \frac{\ddot{x}_R}{R_w} \\ T_L - f_L R_w = I_w \frac{\ddot{x}_L}{R_w} \end{array} \right. \quad (4-3-1)$$

我们发现在上面的方程组中未知量的数量多于方程数，我们并不能解出所有未知量的值，但我们仍然可以得到 $\ddot{\phi}$ 与轮向电机力矩差 $\Delta T = T_R - T_L$ 的关系式。

将式(4-3-1)的第一个方程左右求导有：

$$\frac{\ddot{x}_R - \ddot{x}_L}{2} = \ddot{\phi} R_l$$

代入到式(4-3-1)的第二个方程有：

$$\begin{aligned} (f_R - f_L) R_l &= I \frac{\ddot{x}_R - \ddot{x}_L}{2 R_l} \\ \ddot{x}_R - \ddot{x}_L &= \frac{2(f_R - f_L) R_l^2}{I} \end{aligned}$$

将式(4-3-1)的第三四方程做差得到：

$$\begin{aligned} (T_R - T_L) - (f_R - f_L) R_w &= \frac{I_w}{R_w} (\ddot{x}_R - \ddot{x}_L) \\ (T_R - T_L) - (f_R - f_L) R_w &= \frac{I_w}{R_w} \cdot \frac{2(f_R - f_L) R_l^2}{I} \\ \left( \frac{2R_l^2 I_w}{R_w I} + R_w \right) (f_R - f_L) &= (T_R - T_L) \\ (f_R - f_L) &= \frac{R_w I}{2R_l^2 I_w + R_w^2 I} \Delta T \end{aligned}$$

将上式代入式(4-3-1)的第二个方程有：

$$\begin{aligned} I \ddot{\phi} &= \frac{R_w I}{2R_l^2 I_w + R_w^2 I} \Delta T R_l \\ \ddot{\phi} &= \frac{R_w R_l}{2R_l^2 I_w + R_w^2 I} \Delta T \end{aligned} \quad (4-3-2)$$

此时我们就得到了 $\ddot{\phi}$ 与轮向电机力矩差 $\Delta T = T_R - T_L$ 的关系式，下面构建状态空间方程用以求解 $\mathbf{K}$ 阵。定义状态向量 $\mathbf{x} = [\phi \ \dot{\phi}]^T$ ，控制向量 $\mathbf{u} = [\Delta T]^T$ ，有：

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ \frac{R_w R_l}{2R_l^2 I_w + R_w^2 I} \end{bmatrix} \mathbf{u}$$

### 4.3.2 双环LQR偏航角-偏航角速度控制

在后面章节中会涉及到高速转向处理和功率控制，他们均是对旋转角速度进行的限制，于是我们有必要在转向控制中加入角速度限制，即将单环变成双环同时对角度环进行输出限幅。

不管环数的多少，模型不会变，依旧遵循式(4-3-2)，只不过我们定义的状态向量和控制向量不同。

在外环角度环中，我们定义状态向量 $\mathbf{x} = [\phi]^T$ ，控制向量 $\mathbf{u} = [\dot{\phi}]^T$ ，有：

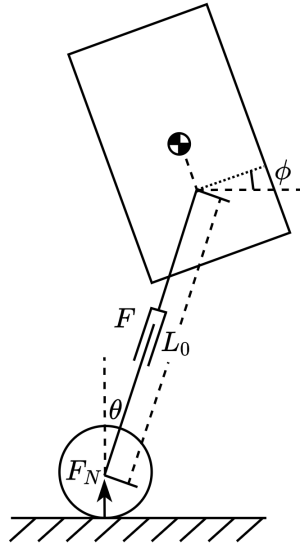
$$\dot{\mathbf{x}} = [0] \mathbf{x} + [1] \mathbf{u}$$

在内环角速度环中，我们定义状态向量 $\mathbf{x} = [\dot{\phi}]^T$ ，控制向量 $\mathbf{u} = [\Delta T]^T$ ，有：

$$\dot{\mathbf{x}} = [0] \mathbf{x} + \left[ \frac{R_w R_l}{2R_l^2 I_w + R_w^2 I} \right] \mathbf{u}$$

## 4.4 离地检测

我们前面章节所讨论的控制算法并不能保证双轮离地时的姿态稳定，因为当我们双轮离地时首先会失去摩擦力，导致动力学模型并不符合，其次我们的推力前馈也会出现问题，原来会以轮子为支撑使得推力前馈可以抵抗机体重量，但双轮离地时，轮子将无法作为支撑，我们的推力前馈会起一个反作用导致双腿伸到最长，对我们的机械上限位产生压力，很容易产生发散。基于此我们需要进行离地检测，还是借鉴玺佬的支持力解算方案，示意图如下：



符号	含义	正方向	单位	符号	含义	单位
$\theta$	摆杆与竖直方向的夹角	顺时针	$rad$	$m_w$	驱动轮质量 (不含电机)	$kg$
$z_w$	驱动轮竖直方向位移	向上	$m$	$L_0$	摆杆长度	$m$
$z_M$	机体竖直方向位移	向上	$m$			
$T_p$	髋关节电机等效输出力矩	右手螺旋向内 (顺时针)	$N \cdot m$			
$P$	驱动轮对摆杆的力的竖直分量	向上	$N$			
$F$	髋关节电机产生的沿摆杆的推力	沿摆杆向下	$N$			
$F_N$	地面对驱动轮的支持力	向上	$N$			

我们对驱动轮进行受力分析:

$$F_N - P - m_w g = m_w \ddot{z}_w$$

$$P \approx F \cos\theta + \frac{T_p}{L_0} \sin\theta$$

这里对 $P$ 进行了近似计算,进行了摆杆对驱动轮的力平衡分析,忽略了驱动轮的竖直加速度,可以接受,力学分解的过程可以参照[2.3.3.2 动力学变换](#)。

其中的 $F$ 和 $T_p$ 可以通过关节电机读取 $T_1$ 和 $T_2$ 然后通过动力学逆解算得到,逆解算矩阵见式(3-2-7)。

然后,我们并不能实际检测出驱动轮的竖直加速度,我们的传感器位于机体之上,我们只能测量出机体的竖直加速度,我们需要对此进行变换:

$$\begin{aligned} \ddot{z}_w &= \frac{\partial^2}{\partial t^2} (z_M - L_0 \cos\theta) \\ &= \frac{\partial}{\partial t} (\dot{z}_M - \dot{L}_0 \cos\theta + L_0 \dot{\theta} \sin\theta) \\ &= \ddot{z}_M - (\ddot{L}_0 \cos\theta - \dot{L}_0 \dot{\theta} \sin\theta) + (\dot{L}_0 \dot{\theta} \sin\theta + L_0 \ddot{\theta} \sin\theta + L_0 \dot{\theta}^2 \cos\theta) \\ &= \ddot{z}_M - \ddot{L}_0 \cos\theta + 2\dot{L}_0 \dot{\theta} \sin\theta + L_0 \ddot{\theta} \sin\theta + L_0 \dot{\theta}^2 \cos\theta \end{aligned}$$

其中,我们注意到有 $\ddot{L}_0$ 和 $\ddot{\theta}$ 两个二阶导项,由于我们无法读取关节电机的 $\dot{\phi}_1$ 和 $\dot{\phi}_4$ ,我们并不能得到 $\ddot{L}_0$ 和 $\ddot{\theta}$ 的精确解,我们只能通过 $\dot{L}_0$ 和 $\dot{\theta}$ 进行差分然后滤波一下,效果略差一点,但可以接受。而且我们也可以发现 $\ddot{z}_M$ 事实上应该是机体相对于大地的竖直加速度,需要将陀螺仪读取的加速度变换到绝对坐标系。

然后通过这个公式我们对本队的平衡步兵进行了下台阶测试:



效果还是可以的。

对离地时的处理还是采用佬佬的方法，将**K**阵中除 $K_{21}$   $K_{22}$ 外全部置零，让平衡步兵在飞坡过程中仅仅保持腿部竖直。

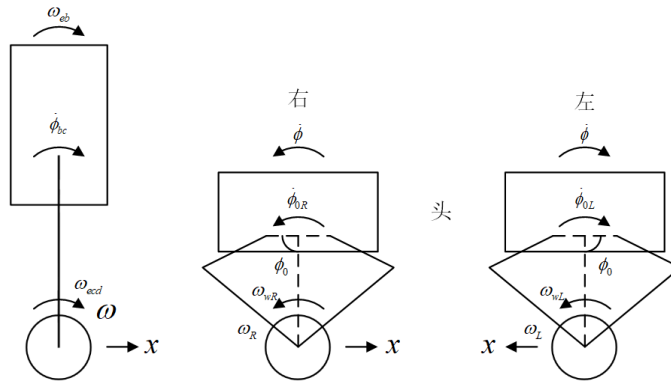
#### 4.5 运动估计与打滑补偿

目前我们还没有对 $\dot{x}$ 和 $\dot{y}$ 进行讨论，其实我们并不能直接对左右轮向电机角速度取平均作为速度，我们还需要考虑定子侧的转动和平动，我们通过如下的算式进行解算（还是佬佬的开源）：

$$w = w_{eb} + \dot{\phi}_{bc} + w_{ecd}$$

符号	含义	正方向	单位
$w$	驱动轮转子相对大地的角速度	顺时针	rad
$w_{eb}$	机体相对大地的角速度	顺时针	rad
$\dot{\phi}_{bc}$	轮向电机定子相对机体的角速度	顺时针	rad
$w_{ecd}$	轮向电机转子相对定子的角速度（可由编码器测得）	顺时针	rad

由于左右腿各变量的符号方向比较乱，下面做示意图：



由此我们可以得到速度解算公式：

$$\begin{cases} w_L = w_{wL} - \dot{\phi}_{0L} - \dot{\phi} \\ w_R = -w_{wR} - \dot{\phi}_{0R} - \dot{\phi} \end{cases}$$

我们再回到单腿上，通过驱动轮转子相对大地的角速度，我们有：

$$\dot{x}_w = wR$$

通过几何关系，我们也有 $x_M = x_w + L_0 \sin\theta$ ，两边求导有：

$$\begin{aligned} \dot{x}_M &= \dot{x}_w + \dot{L}_0 \sin\theta + L_0 \dot{\theta} \cos\theta \\ &= wR + \dot{L}_0 \sin\theta + L_0 \dot{\theta} \cos\theta \end{aligned}$$

于是我们通过算术平均的方法就能得到机体速度的测量值：

$$\begin{cases} \dot{x}_{ML} = w_L R + \dot{L}_{0L} \sin\theta_L + L_{0L} \dot{\theta}_L \cos\theta_L \\ \dot{x}_{MR} = w_R R + \dot{L}_{0R} \sin\theta_R + L_{0R} \dot{\theta}_R \cos\theta_R \end{cases}$$

$$v_{measure} = \frac{\dot{x}_{ML} + \dot{x}_{MR}}{2}$$

将 $v_{measure}$ 对时间积分就能得到机体位移，这样就已经可以用于控制向量的计算了。

但是我们考虑一个问题就是，轮子会出现打滑，导致 $v_{measure}$ 剧烈扰动，在通过小弹丸的时候尤为明显，这个时候我们就可以将机体加速度计数据通过卡尔曼滤波滤除扰动达到一种打滑补偿效果，因为加速度计不会出现打滑。

我们构建状态向量、测量向量、状态转移矩阵和量测矩阵分别为：

$$\mathbf{x} = \begin{bmatrix} v \\ a \end{bmatrix} \quad \mathbf{z} = \begin{bmatrix} v_{measure} \\ a_{measure} \end{bmatrix} \quad \mathbf{F} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \quad \mathbf{H} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

其中 $a_{measure}$ 也应该是机体相对大地的前向加速度，需要将陀螺仪读取的加速度变换到绝对坐标系。

这样我们就会得到滤波后的机体速度 $\dot{x}$ ，通过时间积分就可以得到机体位移 $x$ 。

#### 4.6 倒地自起

倒地自起是平衡的一个重要环节，用于让平衡步兵从任何姿态上电都可以进入满足进入二阶倒立摆模型的条件。我们有两个方案，一种是应对以前赛季平衡倒地z轴方向不竖直的情况，即倒地时俯仰角很大的情况，一种是当前赛季平衡倒地z轴方向竖直的情况，即倒地时俯仰角较小的情况。

第一种方案，先进行用于倒地自起和一阶倒立摆模型的腿长控制（参照[2.3.4.1 腿长控制](#)），可以在任意姿态情况下将腿收起来，等待收腿完成后，进入一阶倒立摆模型，等待机体俯仰角平稳后进入云台底盘偏航角同步，等待同步完成，起身进入二阶倒立摆模型。虽然实现了任意姿态倒地自起，但耗时较长。

第二种方案，同样先进行用于倒地自起和一阶倒立摆模型的腿长控制，等待收腿完成后，直接进入云台底盘偏航角同步，等待同步完成，起身进入二阶倒立摆模型。在我们的测试中，如果第一步收腿时目标腿长过短，可能进入奇异状态，无法起身需要一定的外部扰动。后面通过我们的测试，第二种方案对于第一种情况同样适用。

#### 4.7 失控检测

在平衡步兵平时的运行过程中，可能出现过大扰动、被摔到或是卡墙等情况，都会导致姿态发散出现失控，这时候我们的失控检测就尤为重要，既能保障人身安全又能帮助其恢复正常。

我们的失控检测比较简单粗暴，尽可能对正常运行减少影响（减少误判断），我们仅在俯仰角大于一定的阈值或者当前位移与期望位移差距过大时进入失控状态。

进入失控状态后，机器人会重新进入倒地自起，重新起身恢复正常，但这过程比较耗时而且容易被补刀，同时失控期间可能超功率，所以如果可以不失控，还是不要失控比较好。

#### 4.8 高速转向处理

在高速转向时，会产生较大的向心力，根据高中知识，摩擦力会去提供向心力，内侧轮会比外侧轮提供更多的向心力，有时会导致支持力不足，向外侧翻倒。

我们直到向心力 $F = mvw$ ，所以只要我们保证 $vw$ 在一定范围内就能避免翻车，于是我们得到了角速度限制值：

$$w_{Limit} = \frac{K}{\dot{x}}$$

其中， $K$ 是经验参数，需要通过调参确定，然后将 $w_{Limit}$ 作为转向控制外环的输出限幅值即可（参照[2.3.4.3 转向控制](#)），不过注意在 $\dot{x} = 0$ 时要做额外处理，同时对 $w_{Limit}$ 也要限幅以防止 $w_{Limit}$ 过大。

#### 4.9 功率控制

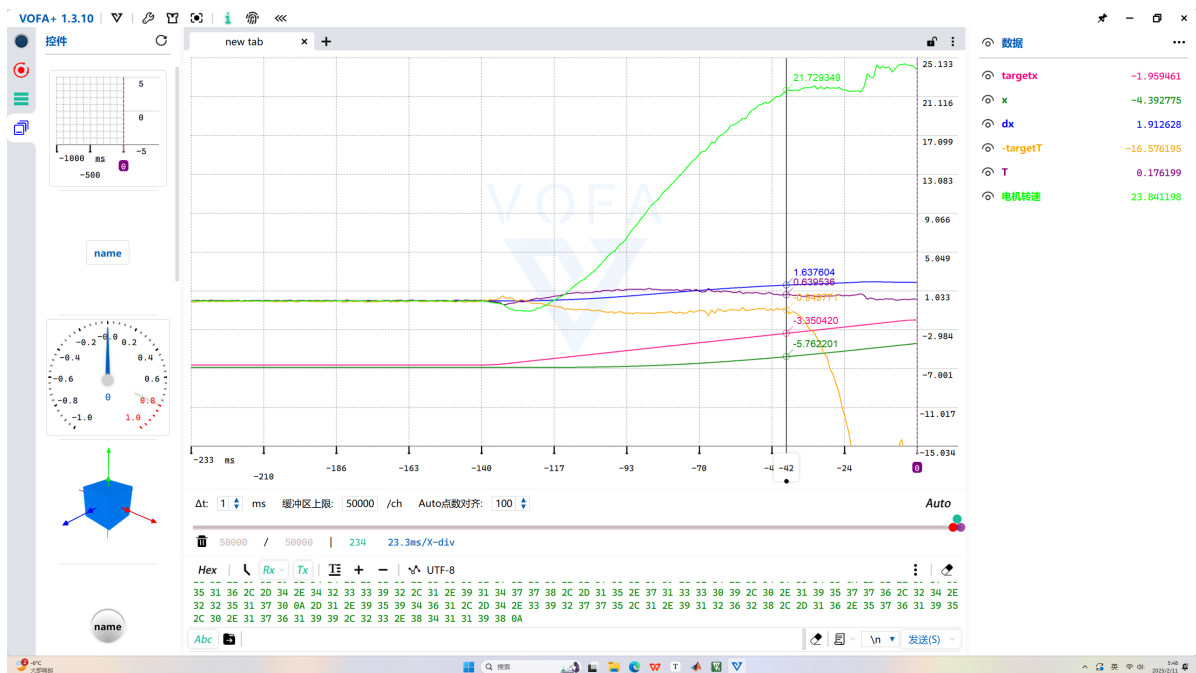
功率控制方面通过与一些队伍的交流，表现出对超电的高度依赖，可惜我们队伍没有超电，思来想去，还是借鉴上交的开源，并将参数调的极为保守。

在上交的开源中，通过计算得到了如下的关系式：

$$v_{Limit} = K\sqrt{P_{ref}}$$

笔者也对于哈工程建模进行了粗略推导，大抵也可以套用这个公式，然后对照裁判系统对 $K$ 进行了调节。

下面，简单谈一下笔者在调功率控制的感受（菜队勿喷），平衡步兵的功率控制与其他底盘构型有本质的区别，我们无法对轮向力矩进行限幅，对轮向力矩进行任何限幅都将导致平衡步兵出现无法平衡的问题，因为力矩的输出达不到模型计算出保持稳定时的力矩，这个问题我们深有体会，我们的平衡步兵在最开始时出现了电机选型的问题，我们选用了MF9025 35T，其电机空载转速只有280rpm，于是在我们企图达到2m/s时由于电机转速已经升到了峰值，根据电机曲线，力矩输出将大大减小，无法维持平衡产生翻倒，下面是我们测试时的曲线：



然后平衡在起步时的功率峰值是恐怖的，也是钳制平衡运行速度的关键，平衡步兵在 $2m/s$ 的起步能达到 $100W$ 左右，但匀速仅有不到 $30W$ ，面对如此大的功率缺额，笔者也只能无奈叹息。平衡匀速状态的低功率也让我们在联赛赛的检录很头疼，因为检录功率时，我只能将拨杆前推一下，如果一直推就会导致位移差过大进入失控，这个时候电机不会空转，导致功率很小，不能通过检录，好在万能的群友提到了可以在此基础上把云台别过去，让底盘运行加上转向，强行提高功率，确实是有效的。最后还要说明一个情况就是我们在测试功率时，笔者狠狠踹了一脚，结果她用了将近 $300W$ 去保持平衡，这是恐怖的，也是笔者无能为力的。

#### 4.10 小陀螺转速处理

小陀螺的转速同样也是受功率控制，更确切的说受起步时的功率尖峰限制，笔者类比了运行速度的功率控制，得出以下关系式：

$$w_{PowerLimit} = K\sqrt{P_{ref}}$$

由于笔者没有完成小陀螺移动，时间太紧了，所以在小陀螺期间完全不用考虑运行速度的影响，但笔者仍然将 $w_{PowerLimit}$ 和 $w_{Limit}$ 进行比较取较小值，以应对高速运行下突然开小陀螺的情况，效果还是不错的，具体效果可以看2025赛季山东站巡天御风VS山海机甲。

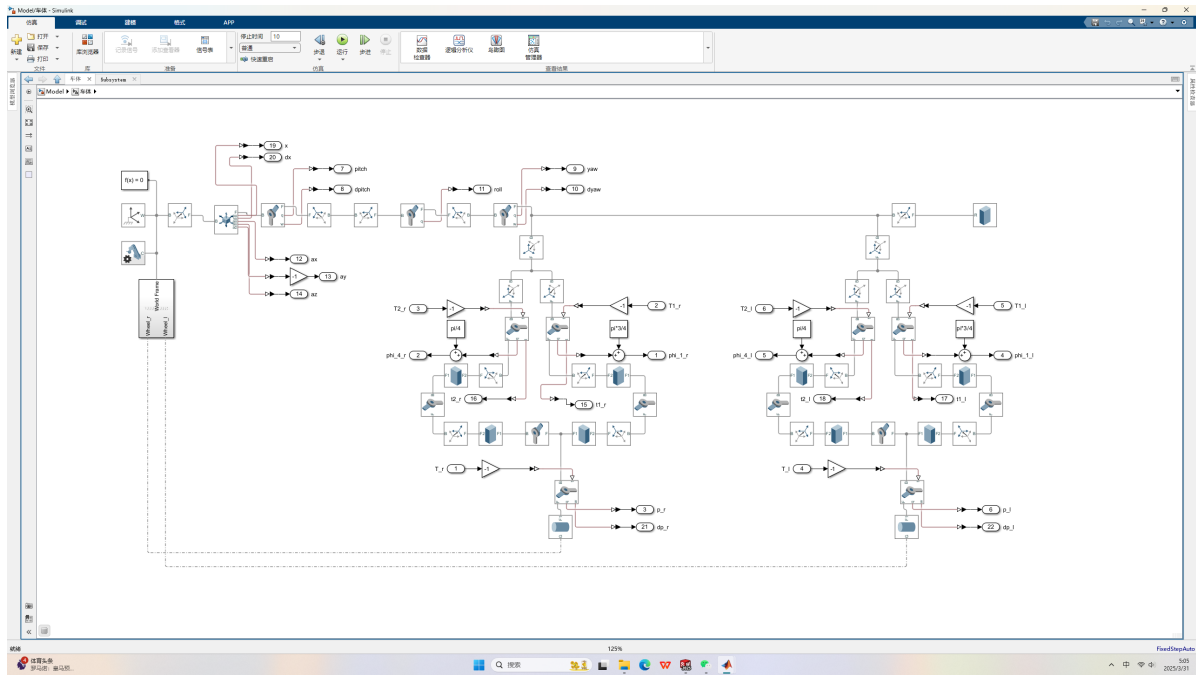
#### 4.11 起跳处理

对于起跳的处理有很多方法，笔者采取了如下流程，先将腿收到近乎机械下限位的位置，以获得最长的加速距离，然后给予一个很大的推力 $F$ ，使机体弹射起来，在空中进行收腿，此时给予一个反推力以更快的收腿，然后等待一段时间，再将腿伸开，在落地时可以腿部阻尼的作用防止落地太硬，经过一段时间的调试我们可以将重 $23.35kg$ 的平衡步兵跳起接近 $220mm$ 。

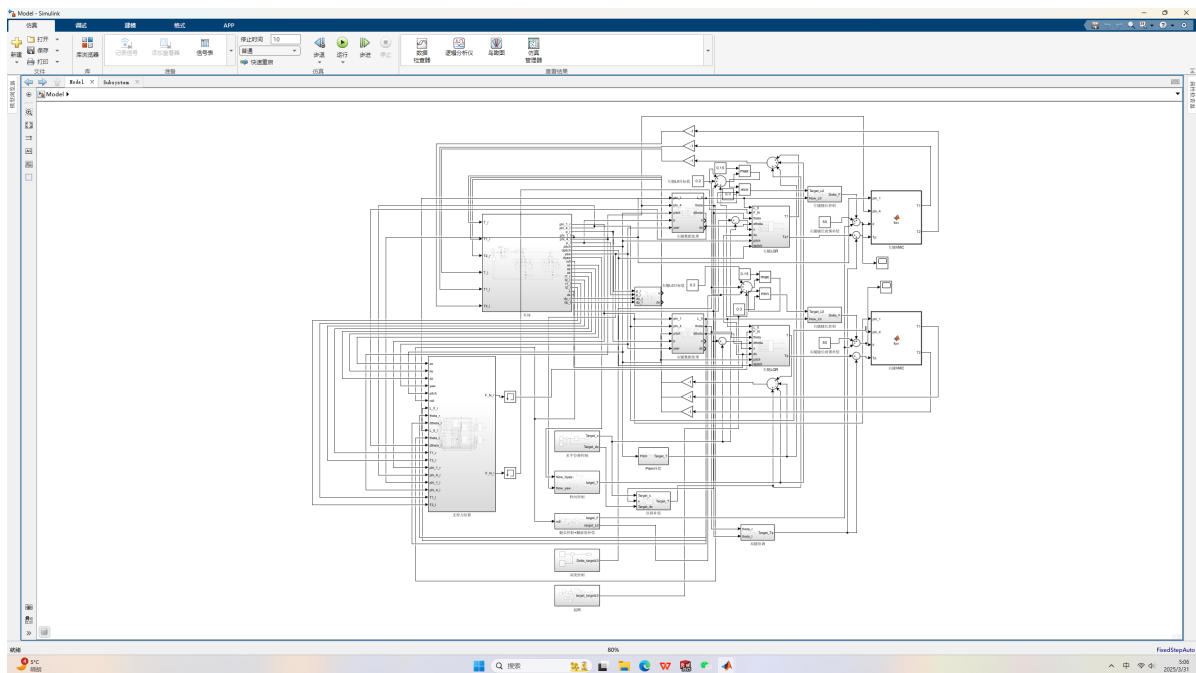


## 5 平衡步兵的Simscape仿真

机器人建模:



控制系统建模 (没有整理, 有点乱):



## 6 平衡步兵的观测器设计

数据准确的观测器是平衡步兵进行后续控制的必要条件, 下面笔者对本队平衡观测器进行一定的介绍。

笔者采用结构体的形式来存储变量:

```
1 typedef struct
2 {
3     float x;          //轮子位移x(m)
4     float Angle;     //轮子角度(rad)
5     float Speed;     //轮子速度(rad/s)
6     float T;         //轮子转矩T(N·m)
7 }Observer_wheelStatus;
8
9 typedef struct
10 {
11     Observer_wheelStatus Wheel; //轮子状态
```

```

12
13 float phi_1;           //五连杆phi1(rad)
14 float phi_4;           //五连杆phi4(rad)
15 float dphi_1;         //五连杆dphi1(rad/s)
16 float dphi_4;         //五连杆dphi4(rad/s)
17
18 float L_0;            //摆杆长度L0(m)
19 float last_dL_0;      //上一次摆杆长度last_L0(m)
20 float dL_0;           //摆杆长度变化率dL0(m/s)
21 float ddL_0;         //摆杆长度二阶变化率ddL0(m/s^2)
22
23 float phi_0;          //摆杆角度phi0(rad)
24 float dphi_0;         //摆杆角速度dphi0(rad/s)
25
26 float T1;            //髋关节1的转矩T1(N·m)
27 float T2;            //髋关节2的转矩T2(N·m)
28
29 float theta;          //摆杆摆角theta(rad)
30 float last_dtheta;    //上一次摆杆摆角last_theta(rad)
31 float dtheta;         //摆杆摆角角速度dtheta(rad/s)
32 float ddtheta;        //摆杆摆角角加速度ddtheta(rad/s^2)
33
34 float F;             //摆杆推力F(N)
35 float Tp;            //摆杆扭矩Tp(N·m)
36 float FN;            //支持力FN(N)
37
38 float J_11;           //VMC雅可比矩阵J元素
39 float J_12;           //VMC雅可比矩阵J元素
40 float J_21;           //VMC雅可比矩阵J元素
41 float J_22;           //VMC雅可比矩阵J元素
42
43 float T_11;           //VMC逆解矩阵T元素
44 float T_12;           //VMC逆解矩阵T元素
45 float T_21;           //VMC逆解矩阵T元素
46 float T_22;           //VMC逆解矩阵T元素
47 }Observer_LegStatus;
48
49 typedef struct
50 {
51     float Yaw;           //偏航角(rad)
52     float GM6020_Yaw;
53     float Pitch;         //俯仰角(rad)
54     float Roll;          //翻滚角(rad)
55
56     float dYaw;          //偏航角速度(rad/s)
57     float dPitch;        //俯仰角速度(rad/s)
58     float dRoll;         //翻滚角速度(rad/s)
59
60     float a_xE;          //世界坐标系x轴加速度(m/s^2)
61     float a_yE;          //世界坐标系y轴加速度(m/s^2)
62     float a_zE;          //世界坐标系z轴加速度(m/s^2)
63
64     float a_xb;          //机体坐标系x轴加速度(m/s^2)
65     float a_yb;          //机体坐标系y轴加速度(m/s^2)
66     float a_zb;          //机体坐标系z轴加速度(m/s^2)
67
68     float x;             //位移x(m)
69     float dx;            //速度dx(m/s)
70     float x_nofilter;    //未运动估计的位移(m)
71     float dx_nofilter;   //未运动估计的速度(m)
72
73     MotionEstimation_Balance MotionEstimation; //运动估计结构体
74 }Observer_BodyStatus;
75
76 typedef struct
77 {
78     float dt;            //闭环周期
79
80     float P;
81
82     Observer_LegStatus LeftLeg; //左腿
83     Observer_LegStatus RightLeg; //右腿
84
85     Observer_BodyStatus Body; //机体

```

下面对一些必要的观测量进行介绍。

首先在Observer\_WheelStatus结构体中我们只关注Speed，直接读取转向电机编码器即可。

Observer\_LegStatus结构体中，对于phi\_1 phi\_4 dphi\_1 dphi\_4 T1 T2，直接读取膝关节电机编码器即可，对于L\_0 dL\_0 phi\_0 dphi\_0直接通过式(3-1-3)和式(3-1-4)即可求解，笔者测试了对dL\_0进行差分和精确求解的比较，如下：

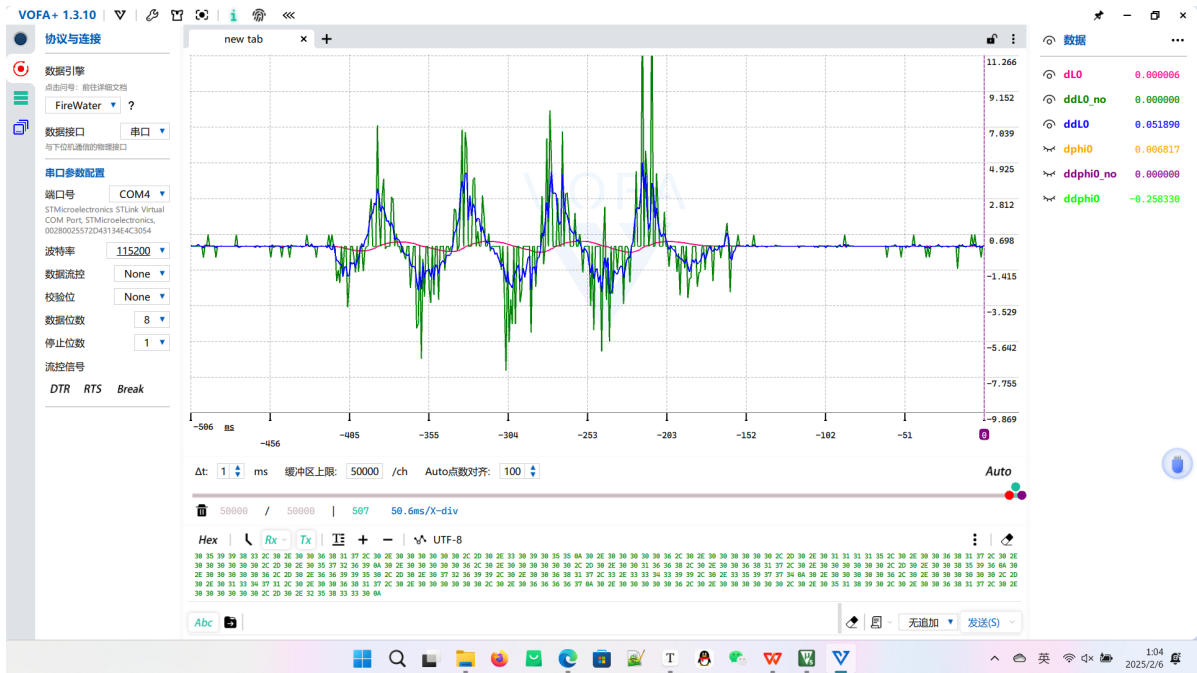


其中红色为差分求解，绿色为精确求解，我们可以发现精确求解相比差分更平滑更连续，效果好很多。然后theta可由几何关系得到，dtheta由其求导而得，关系式如下：

$$\theta = \frac{\pi}{2} - (\phi_0 + \phi)$$

$$\dot{\theta} = -(\dot{\phi}_0 + \dot{\phi})$$

而由于 $\ddot{L}_0$   $\ddot{\phi}_0$ 无法精确求解，ddL\_0和ddtheta只能通过差分+一阶低通滤波求解，笔者也测试了滤波前后的比较：





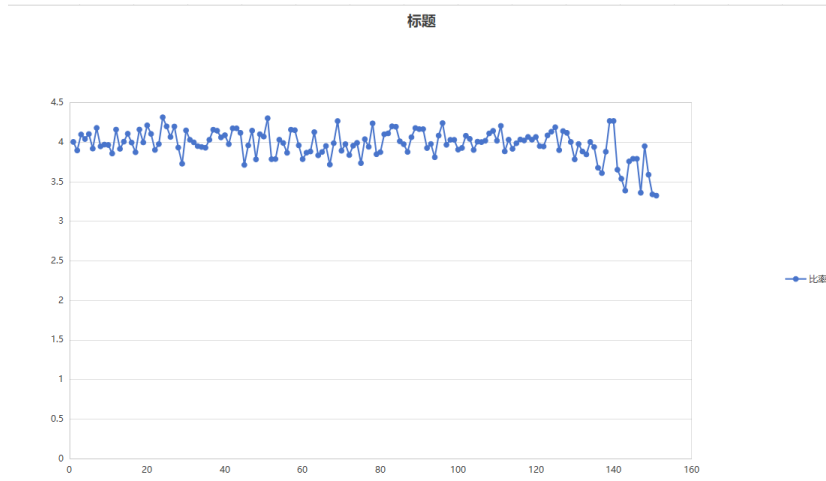
可以看出滤波效果还是不错的，抑制了很多高频噪声，但还达不到精确解那样平滑。再然后J T可通过式(3-2-6)和式(3-2-7)得到，F Tp可通过逆动力学变换矩阵T求解，FN可通过[2.3.4.4 离地检测](#)中的方法求解。

Observer\_BodyStatus结构体中，Yaw Pitch Roll dYaw dPitch dRoll可直接读取陀螺仪通过EKF姿态解算得到，注意这里均是机体坐标系，a\_yE a\_zE可通过加速度计坐标变换转到世界坐标系得到（参照[2.3.4.4 离地检测](#)和[2.3.4.5 运动估计与打滑补偿](#)），GM6020\_Yaw用于云台底盘Yaw同步之后底盘跟随时模拟底盘偏航角，可直接通过Yaw轴电机编码器得到，x dx可通过运动估计得到，参见[2.3.4.5 运动估计与打滑补偿](#)。

Observer\_Balance结构体中，dt默认0.002，即500Hz闭环频率，P是通过软件计算得到的功率，采用的公式是笔者在大二时提出的，效果还不错，不过对于堵转时，效果稍差，公式如下：

$$P = K \cdot \sqrt{3}T\Omega$$

其中，K为经验参数，需要搭配功率计进行调参，Ω为电机转速，单位rad/s。笔者测试下来，M3508电机对应K = 1，MF9025 16T电机对应K = 4，下面是笔者拟合MF9025 16T电机功率参数时的截图：



通过笔者的功率计算公式得到的功率与实际功率的对比如下：



拟合效果还是非常理想的。

## 7 平衡步兵的云台控制

不说别的，麻神NB就完事了。笔者最近也在拜读中科大的教程，确实提供了很多新的思路，后面笔者也打算一一尝试，下面仅仅对笔者本赛季的探索进行介绍（菜队勿喷）。

笔者借鉴LQR通过动力学建模优化PD参数，使我们的云台响应相较于上一个赛季有了质的飞跃（菜队勿喷）。

### 7.1 Pitch角控制

对于Pitch轴，笔者采用了匀质单摆模型，定义状态向量 $\mathbf{x} = [\gamma \quad \dot{\gamma}]^T$ ，控制向量 $\mathbf{u} = [T]^T$ ，参照[2.3.4.1 腿长控制](#)于是有：

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ \frac{1}{J} \end{bmatrix} \mathbf{u}$$

$$J = \frac{1}{3} ml^2$$

除了LQR外，笔者还为其加入了重力前馈 $\frac{1}{2} mgl \cos(\gamma)$ ，说到底笔者采用了前馈+PD+微分先行的策略，通过合理的参数，我们最终的效果达到了稳态误差 $0.5^\circ$ 之内，且响应迅速。

### 7.2 Yaw角控制

对于Yaw轴，笔者采用了柱体沿轴线旋转的模型，定义状态向量 $\mathbf{x} = [\phi \quad \dot{\phi}]^T$ ，控制向量 $\mathbf{u} = [T]^T$ ，有：

$$\begin{cases} T = J\ddot{\phi} \\ J = \frac{1}{2} mR^2 \end{cases}$$

我们很容易得到其状态空间方程：

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ \frac{1}{J} \end{bmatrix} \mathbf{u}$$

## 8 当前问题与未来规划

由于我们是一个新队伍，技术薄弱资金人员不足，笔者还兼具其他兵种的研发，精力有限，对平衡步兵的研究和测试并不深入，本文档也只是笔者个人的肤浅的想法和经验（轻点喷▽▽），不过目前我们也是面对了一些问题：

- 由于在建模的时候进行了近似线性化，导致在摆杆倾角很大的情况下，其实并不能近似线性化，在考虑对此处非线性区域是否有更好的处理，可能可以借鉴iLQR的思想。
- 调的不够好，因为参数众多，互相之间也有耦合关系，调参较为困难，后期可能通过PSO进行优化。
- 还有一个致命的问题，就是LQR自身的鲁棒性并不好，JOHN C. DOYLE在论文Guaranteed margins for LQG regulators中指出LQG具有任意小的裕度。LQR的低鲁棒性直接导致平衡步兵很容易翻倒和姿态发散，使用失控检测来处理这个问题并不是一个好的主意，因为倒地自起需要一定的时间，在这个时间内平衡步兵很容易暴毙掉，后期将考虑外加鲁棒控制方法提高鲁棒性，例如加入MPC。

- 由于笔者使用了位移控制方法，如果平衡步兵撞墙后容易导致位移积分效应使得翻倒和姿态发散，后期可能考虑加入主动柔顺算法。
- 在飞坡的时候，我们经常遇到翻倒的问题，经过慢动作，我们发现有两种情况，一个是坡道撞到腿的情况，一个是在空中因为俯仰角过大触发了失控检测，后面需要多次测试加以改进，又要碎板子了呜呜呜 $\nabla$ 。
- 其实对于高速转向的问题，南科提到了支持力补偿的算法，笔者认为是有一定必要的，后期准备加入到代码中，提高转向速度。
- 由于我们平衡步兵的底盘重心偏后，导致我们的平衡步兵小陀螺时会明显的向一个方向偏，尤其是低速的情况，后面尝试一下能不能通过一些算法去补偿，或者增加配重将重心配平。同时也考虑参考上交的思路研制小陀螺移动。
- 功率问题现在也是很麻烦，用现在的功率控制算法，在匀速阶段并不能充分利用功率，而且现在底盘超功率并不会扣血而是底盘掉电，事实上这对于平衡步兵来说并不友好，尤其是对我们这样没有超电的队伍，这将是毁灭性的打击，笔者和几个其他队伍的平衡电控负责人进行了请教，基本上往年一贯的策略就是用超电补偿，补偿不过来就通过扣血补偿。目前来看，笔者借鉴了上交的开源，基本上正常情况下缓冲能量就足以弥补意外的功率尖峰，但如果出现卡墙或者被撞等大的扰动时，会产生极大的功率峰值，笔者目前对此无能为力，在2025赛季联盟赛上也确实出现了这种情况，我只能等待底盘上电重新自起，后面得考虑对于没有超电队伍更合理的功率控制，也是希望我们的超电组做出超电 $\nabla$ 。

## 9 后记——碎嘴念

笔者与平衡步兵的缘分要从2023赛季说起，当时刚上大学懵懵懂懂，队伍也是刚刚建立，我作为新生一起去了威海，参加了建队以来的第一场联盟赛，当时就被交龙战队的平衡步兵惊艳到了，也想做一个，不过当时过于菜鸡（现在也是），也就忘了。

到了2024赛季，老学长大四了大部分都去忙毕设了，基本上队伍处在了几乎断代的时期，不过在大家燃烧生命下，我们还是从零开始参加了联盟赛，甚至参加了超级对抗赛，当时的激动至今记忆犹新，虽然这几场比赛我们都被打爆了。

超级对抗赛的第一场我们就被西交用平衡步兵五杀了，只记得我操控工程只看了一眼对面平衡，我就被打死了。

2025赛季开始，其实就做不做平衡步兵的问题进行了多次讨论，因为对于我们这样的队伍，缺资金缺技术缺人，做平衡这样的决定很冒险，然后在国庆七天假期，笔者直接燃尽，速通平衡步兵的仿真（在国庆前就学习了相关理论），证明了可行性，最终我们就有了现在的平衡步兵（我垫了5k的助学金什么时候能回来呜呜呜 $\nabla$ ）。

在最初的时候其实并不顺利，因为没有经验和基础，我们出现了限位的问题以及电机选型的问题，在大量测试中还损坏了好几个限位，撞坏了好几块板子，底盘是到处是伤痕。在代码上也是出现过抄错公式的情况，调了一个月才发现，直到春节结束才是真正的把观测器和原地平衡做的没有公式上的问题，总之，调试的过程相当煎熬，常常因为一个问题熬夜通宵好几天，到了功控的部分，硬卡了一个礼拜，始终找不到好的方法，直到现在也没有很好的解决，目前的状态相较于强队还差的太远太远，技术上的差距还是太大了。

在联盟赛的时候，本来就想让平衡打一局热身赛，测试一下，因为当时在场下还不太稳定，时常发生腿部抽搐和莫名其妙翻倒伸腿的情况，到现在还没有发现问题所在，好在场上平衡发挥的相当稳定，几乎是达到了有史以来的最好状态，也是感谢山东理工放水，让平衡还杀了几个，真的打爽了。后面面对北科和山理，他们都采用了哨兵站中心点的策略，虽然笔者自认为全向步兵已经被笔者调到了很高的性能，但是面对对面三辆车，自身自瞄也时有时无的情况下也无力抵抗。最后一场已经注定离开了，就直接还用平衡上场测试一下，只能说，上场还是很争气的。

回想下来，一切也很梦幻，也很有感触，大一的一个空想变成了现实，我们真的有了平衡步兵，我们真的把平衡带上了赛场，也算是RM闭环了，然后就是希望我们今年还能进入超级对抗赛，能不能冲一波春茧 $\nabla$ ，真的很想去，也希望笔者最后能将一台让自己满意的平衡步兵带进7V7的舞台。

## 10 参考文献

- [1] 陈阳,王洪玺,张兰勇.[轮腿式平衡机器人控制](#)[J]
- [2] 韭菜的菜.[五连杆运动学解算与VMC](#)[OL]
- [3] 韭菜的菜.[轮腿倒立摆机器人运动速度估计](#)[OL]
- [4] djiuser\_MzHeMvWw.[【RM2023-平衡步兵控制系统开源】上海交通大学-云汉交龙](#)[OL]
- [5] Alan.[【RM AWARD 2024】电子科技大学中山学院 RoboBraver 柳幸之 部分技术成果说明](#)[OL]
- [6] JOHN C. DOYLE.[Guaranteed margins for LQG regulators](#)[J]
- [7] 偏红苍穹.[【RM2024-技术报告开源】南方科技大学ARTINX战队](#)[OL]

## 11 通讯信息

QQ: 2297651490 (zpclyn)