

底盘控制模型 V1.4 (2020/04/26)

马伟明

#说明：该文章仅个人看法，难免有谬误之处，欢迎指出和纠正。

Log:

- V1.0
 - 1. 分成三个主题：模型分类、机械模式、陀螺仪模式
 - 2. 模型分类里面对机械模式和陀螺仪模式的简介
 - 3. 陀螺仪模式中对坐标系转换、串级 PID、旋转因子、常规跟随、小陀螺、扭腰六个子标题的阐述
- V1.1
 - 1. 稍微修改和补充了 V1.0 的部分阐述，将 C 代码片段的插入用 <http://www.planetb.ca/syntax-highlight-word> 生成的格式替代
 - 2. 陀螺仪模式中加入底盘卡死处理的初步思路
 - 3. 新建主题：功率控制，完成子标题：功率计算公式推导、功率、扭矩和转速之间的关系讨论
- V1.2
 - 1. 功率控制章节新增子标题：启动特性分析
 - 2. 新增目录，便于检索
- V1.3
 - 1. 新增麦克纳姆轮底盘模型章节
 - 2. 将原来的模型分类修改成模式分类
- V1.4
 - 1. 修改对于底盘姿态的 agl_z 的公式替换
 - 2. 补充功率限制章节，包括对电机性能曲线图的解读

目录

- 一、 麦克纳姆轮底盘模型..... 4
 - 1. 左旋轮与右旋轮..... 4
 - 2. 力学特性 4
 - 3. 麦轮安装方式 5
 - 4. 底盘逆运动学模型..... 7
 - 1) 问题描述 7
 - 2) 问题求解 7
 - 3) 结论扩展 9
 - 4) 坐标系转换..... 10
 - 5. 底盘正运动学模型 11
 - 1) 问题描述 11
 - 2) 问题求解 11
 - 3) 结论扩展 12
 - 4) 坐标系转换..... 15
- 二、 模式分类 17
 - 1. 机械模式 17
 - 2. 陀螺仪模式..... 17
- 三、 机械模式 18
 - 1. 手动爬坡 18
- 四、 陀螺仪模式..... 18
 - 1. 坐标系转换..... 18
 - 2. 串级 PID 18
 - 3. 旋转因子 20
 - 4. 常规跟随 21
 - 5. 小陀螺..... 21
 - 1) 串级 PID 21
 - 2) 单级 PID 23
 - 6. 扭腰..... 23
 - 7. 底盘卡死处理 23
 - 1) 判断底盘被卡死的方式..... 23

2)	底盘被卡死后的处理方式	23
五、	功率限制	24
1.	功率计算公式推导	24
2.	功率、扭矩和转速之间的关系讨论	25
3.	启动特性分析	26
4.	实时功率与缓冲能量	27
5.	功率限制算法	27
6.	优化算法	27

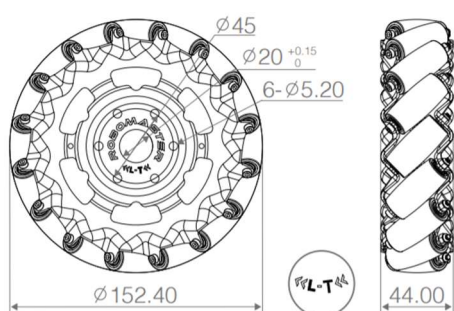
一、 麦克纳姆轮底盘模型

参考资料：

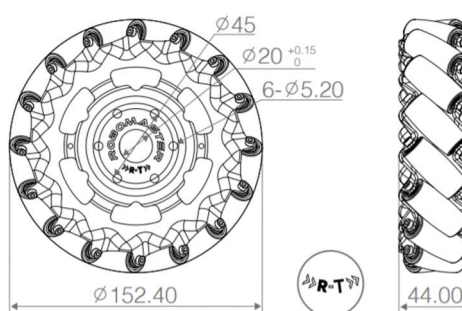
1. 麦克纳姆轮浅谈 <https://zhuanlan.zhihu.com/p/20282234>
2. Kinematic Analysis of Four-Wheel Mecanum Vehicle Rev 05/31/10 © Ether
3. 【干货】麦轮正逆向运动学建模 <https://bbs.robomaster.com/thread-3960-1-1.html>

1. 左旋轮与右旋轮

Left-Threaded Mecanum Wheel



Right-Threaded Mecanum Wheel



图片来源：DJI 麦克纳姆轮说明手册

2. 力学特性



		Standard	Omni	Mecanum
kinematics	V_f	$\omega \cdot r$	$\omega \cdot r \cdot \sqrt{2}$	$\omega \cdot r$
	V_r	-	$\omega \cdot r \cdot \sqrt{2}$	$\omega \cdot r$
	V_d	-	$\omega \cdot r$	$\omega \cdot r / \sqrt{2}$
force	F_f	$4\tau / r$	$4\tau / (r\sqrt{2})$	$4\tau / r$
	F_r	-	$4\tau / (r\sqrt{2})$	$4\tau / r$
	F_d	-	$2\tau / r$	$2\tau\sqrt{2} / r$

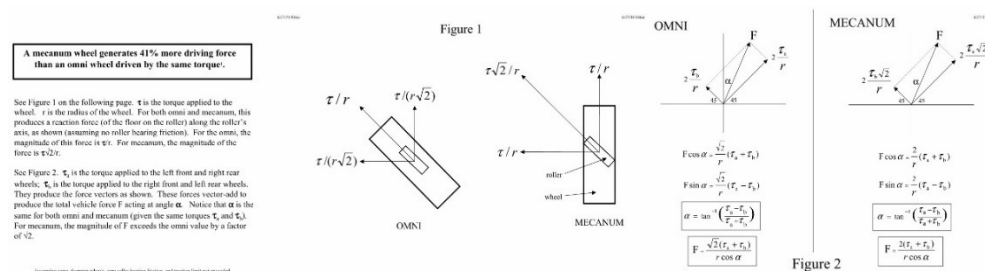
The three columns are for standard, omni, and mecanum 4-wheeled vehicles, respectively. The omni vehicle's wheels are mounted at 45 degrees. All wheels same diameter. The first three rows are vehicle velocity: forward, strafe, and diagonal, for a given wheel speed ω (radians/sec)¹. The second three rows are vehicle total pushing force: forward, strafe, and diagonal, for a given wheel torque τ ¹. These last three rows assume a) frictionless mecanum and omni roller bearings, and b) sufficient traction to support the floor reaction forces.

*Bottom line: for the same wheel speeds, omni vehicle goes 41% faster than mecanum; for the same wheel torque, mecanum vehicle has 41% more pushing force than omni*²

¹ only two wheels are powered for pure diagonal
² the 41% ratio applies to all directions, not just the fwd/strafe/diagonal shown in chart

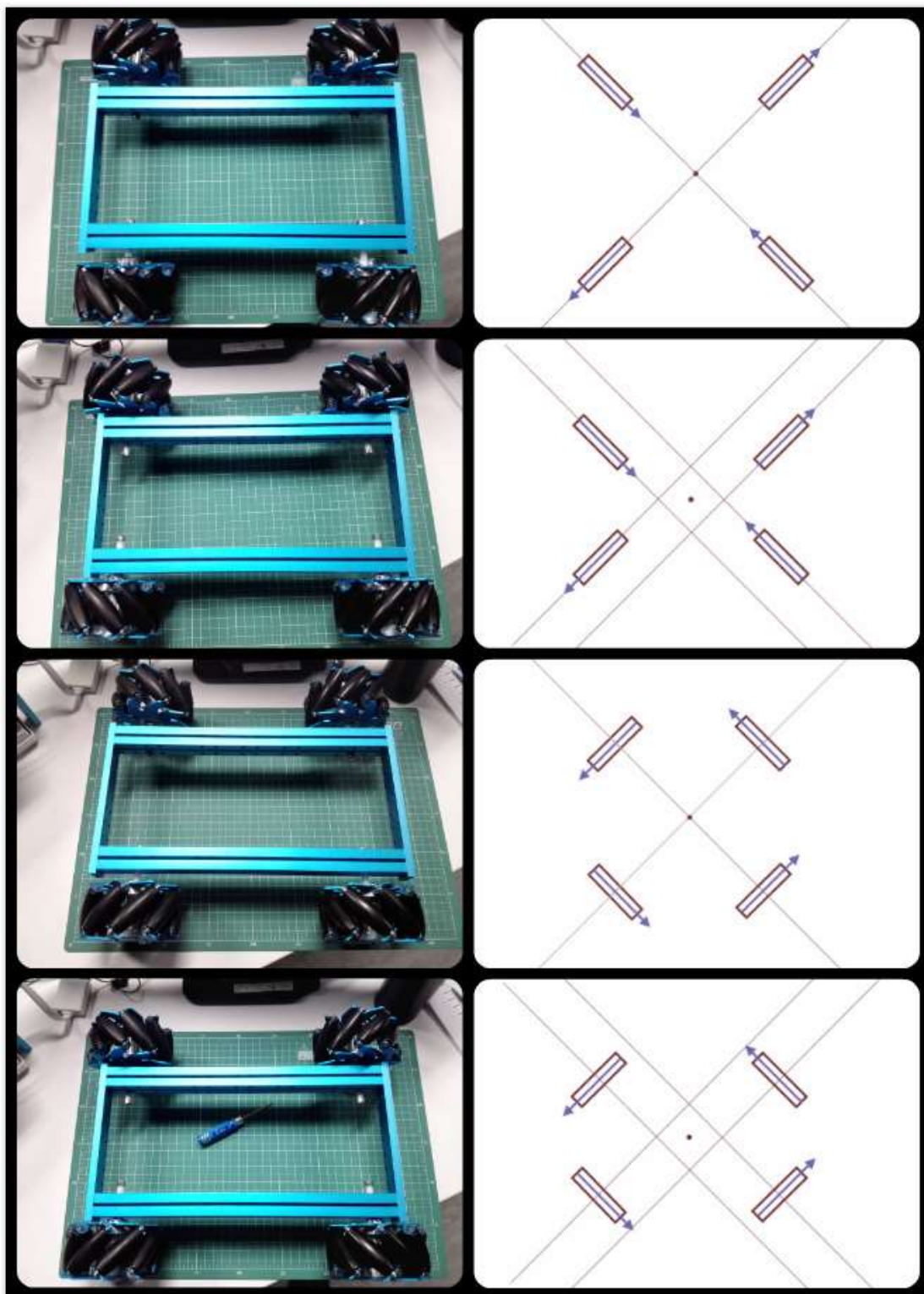
Standard、Omni 和 Mecanum 分别指标准轮、全向轮和麦克纳姆轮。

- 1) 前三行表示速度： V_f 指的是直行速度（前后）， V_r 指的是平移速度（左右）， V_d 指的是对角速度。 ω 指的是轮子转速（rad/s）
- 2) 后三行表示合力。 τ 指的是轮子的扭矩。在这里做了两个假设：①麦克纳姆轮和全向轮的轮子轴承无摩擦。②有足够的牵引力来支持地板的反作用力。
- 3) 在这里全向轮的角度被安装成 45° 偏置，四轮的全向轮底盘则是以“X”这种形式安装的。



3. 麦轮安装方式

麦轮有两种形态，呈现手性对称，分别为左旋轮和右旋轮。想要制作能全向运动的底盘，对四个麦轮的安装也是有讲究的。常见的安装方式有如下四种：



注：上图的长方形为轮子底部接触地面的辊子。紫色箭头为辊子受到的摩擦力。

1) X-正方形

辊子受到的摩擦力经过转轴中心，产生的扭矩为 0，因此底盘的 yaw 轴无法主动旋转。一般不会采用这种安装方式。

2) X-长方形

轮子的运动组合可以产生 yaw 轴旋转的效果，但是转动力矩的力臂较短。采

用这种安装方式的情况也不多。

3) O-正方形

四个轮子位于正方形的四个顶点，平移和旋转都没有问题。只是考虑到实际的机器人底盘形状、尺寸等因素，这种安装方式可遇不可求。

4) O-长方形

轮子的运动组合可以产生 yaw 轴旋转的效果，而且转动力矩的力臂较长。因此，这是**最常见的安装方式**。我们战队的机器人底盘也是默认以这种方式安装的。

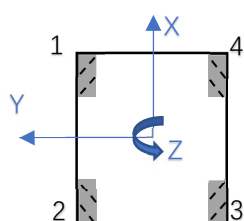
4. 底盘逆运动学模型

1) 问题描述

已知底盘运动速度，求解四个轮子的转速。

2) 问题求解

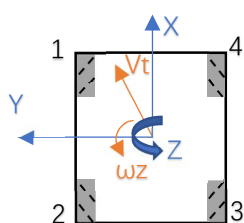
a) 构建坐标系和总方程



坐标系如左所示：

X,Y,Z 满足右手法则。

- 麦轮（灰色矩形）
- 与地面接触的自由辊子（黑色虚线）
- 规定的正方向如图所示，方向相反时取个“-”号。X 表示前后，Y 表示左右，Z 表示 YAW。
- 规定 x_n, y_n 均为正数，符号位单独提出，用以表示轮毂中心位置，如 $(-x_2, +y_2)$
- 麦轮编号如图所示，左上角为①轮



我们用一个矩阵[V]来表示底盘的运动

$$[V] = [V_{tx} \ V_{ty} \ \omega_z]^T$$

用另一个矩阵[Ω]来表示四个轮子的转速

$$[\Omega] = [\omega_1 \ \omega_2 \ \omega_3 \ \omega_4]^T$$

利用公式 $\omega = v/r$ 构建方程

$$[\Omega] = 1/r * [R] * [V] \quad (1)$$

$\omega_1, \omega_2, \omega_3, \omega_4$ 是四个麦轮的角速度，单位为 rad/s。

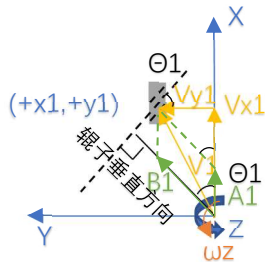
V_{tx}, V_{ty} 是底盘的平移速度在 x, y 方向上的分量，单位为 mm/s。 ω_z 是底盘的旋转速度，单位为 rad/s。r 为麦轮半径，单位为 mm。[R]为要求解的系数矩阵（转移矩阵）。

b) 分析左旋轮

$$\vec{Vt} = \vec{Vtx} + \vec{Vty}$$

$$V_{x1} = V_{tx} - \omega_z \cdot y_1 \quad (2)$$

$$V_{y1} = V_{ty} + \omega_z \cdot x_1 \quad (3)$$



$$\vec{V1} = Vx1\vec{i} + Vy1\vec{j}$$

$$\vec{V1} = A1\vec{i} + B1\vec{u}$$

$$A1 = Vx1 - \frac{Vy1}{\tan(\theta1)} = \omega1 \cdot r, \text{ 形式对 3 号轮同样适用}$$

A1 - 轮毂中心(线)速度

B1 - 触地自由辊子中心(线)速度

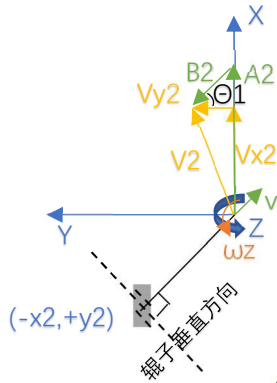
\vec{u} - 垂直于辊子轴线的单位向量

$$\omega1 = \frac{1}{r} \cdot (Vx1 - Vy1/\tan(\theta1)) \quad (4)$$

$$\omega1 = \frac{1}{r} \cdot ((Vtx - \omega z \cdot y1) - (Vty + \omega z \cdot x1)/\tan(\theta1))$$

$$\omega1 = \frac{1}{r} \cdot \left(Vtx - \frac{1}{\tan(\theta1)} \cdot Vty + \left(-\frac{x1}{\tan(\theta1)} - y1 \right) \cdot \omega z \right)$$

c) 分析右旋轮



$$\vec{Vt} = \vec{Vtx} + \vec{Vty}$$

$$Vx2 = Vtx - \omega z \cdot y2$$

$$Vy2 = Vty - \omega z \cdot x2$$

$$\vec{V2} = Vx2\vec{i} + Vy2\vec{j}$$

$$\vec{V2} = A2\vec{i} - B2\vec{v}$$

$$A2 = Vx2 + Vy2 \cdot \tan(\theta2) = \omega2 \cdot r, \text{ 形式对 4 号轮同样适用}$$

A2- 轮毂中心(线)速度

B2 - 触地自由辊子中心(线)速度

\vec{v} - 垂直于辊子轴线的单位向量

$$\omega2 = \frac{1}{r} \cdot (Vx1 + Vy1 \cdot \tan(\theta2)) \quad (4)$$

$$\omega2 = \frac{1}{r} \cdot ((Vtx - \omega z \cdot y2) + (Vty - \omega z \cdot x2) \cdot \tan(\theta2))$$

$$\omega2 = \frac{1}{r} \cdot (Vtx + \tan(\theta2) \cdot Vty + (-x2 \cdot \tan(\theta2) - y2) \cdot \omega z)$$

d) 扩展到四个轮子

$$\omega_2 = \frac{1}{r} \cdot (Vtx + \tan(\theta_2) \cdot Vty + (-x_2 * \tan(\theta_2) - y_2) \cdot \omega_z)$$

$$\omega_3 = \frac{1}{r} \cdot (Vtx - 1/\tan(\theta_3) \cdot Vty + (x_3/\tan(\theta_4) + y_3) \cdot \omega_z)$$

$$\omega_4 = \frac{1}{r} \cdot (Vtx + \tan(\theta_4) \cdot Vty + (x_4 * \tan(\theta_4) + y_4) \cdot \omega_z)$$

$$\text{从而得到 } [R] = \begin{bmatrix} 1 & -1/\tan(\theta_1) & x_1*(-1)/\tan(\theta_1)-y_1 \\ 1 & \tan(\theta_2) & -x_2*\tan(\theta_2)-y_2 \\ 1 & -1/\tan(\theta_3) & -x_3*(-1)/\tan(\theta_3)+y_3 \\ 1 & \tan(\theta_4) & x_4*\tan(\theta_4)+y_4 \end{bmatrix}$$

因为麦轮的辊子排列方向为 45° ，所以 $\theta_1 = \theta_2 = \theta_3 = \theta_4 = 45^\circ$

$$[R] = \begin{bmatrix} 1 & -1 & -x_1-y_1 \\ 1 & 1 & -x_2-y_2 \\ 1 & -1 & x_3+y_3 \\ 1 & 1 & x_4+y_4 \end{bmatrix}$$

$$\omega_1 = \frac{1}{r} \cdot (Vtx - Vty - (x_1 + y_1) \cdot \omega_z)$$

$$\omega_2 = \frac{1}{r} \cdot (Vtx + Vty - (x_2 + y_2) \cdot \omega_z)$$

$$\omega_3 = \frac{1}{r} \cdot (Vtx - Vty + (x_3 + y_3) \cdot \omega_z)$$

$$\omega_4 = \frac{1}{r} \cdot (Vtx + Vty + (x_4 + y_4) \cdot \omega_z)$$

3) 结论扩展

a) $\omega \rightarrow n$ 的转换

$$\omega = \text{rad/s}$$

$$n = \text{r/min}$$

$$n * 2\pi = \text{rad/min}$$

$$n * 2\pi/60 = \omega$$

b) 旋转中心调节

假设重心不在 (0,0)，而在 (x', y')，注意这里的 x' 和 y' 带符号。

$$\omega_1 = \frac{1}{r} \cdot (Vtx - Vty - (x_1 + y_1 - x' - y') \cdot \omega_z)$$

$$\omega_2 = \frac{1}{r} \cdot (Vtx + Vty - (x_2 + y_2 + x' - y') \cdot \omega_z)$$

$$\omega_3 = \frac{1}{r} \cdot (Vtx - Vty + (x_3 + y_3 + x' + y') \cdot \omega_z)$$

$$\omega_4 = \frac{1}{r} \cdot (Vtx + Vty + (x_4 + y_4 - x' + y') \cdot \omega_z)$$

$$\pi d = 2\pi r = \text{麦轮周长}$$

定子转速:

$$n1' = \frac{60}{\pi d} \cdot (Vtx - Vty - (x1 + y1 - x' - y') \cdot \omega z)$$

$$n2' = \frac{60}{\pi d} \cdot (Vtx + Vty - (x2 + y2 + x' - y') \cdot \omega z)$$

$$n3' = \frac{60}{\pi d} \cdot (Vtx - Vty + (x3 + y3 + x' + y') \cdot \omega z)$$

$$n4' = \frac{60}{\pi d} \cdot (Vtx + Vty + (x4 + y4 - x' + y') \cdot \omega z)$$

M3508 电机的减速比是 19:1, 则转子转速:

$$n1 = 19 \cdot \frac{60}{\pi d} \cdot (Vtx - Vty - (x1 + y1 - x' - y') \cdot \omega z)$$

$$n2 = 19 \cdot \frac{60}{\pi d} \cdot (Vtx + Vty - (x2 + y2 + x' - y') \cdot \omega z)$$

$$n3 = 19 \cdot \frac{60}{\pi d} \cdot (Vtx - Vty + (x3 + y3 + x' + y') \cdot \omega z)$$

$$n4 = 19 \cdot \frac{60}{\pi d} \cdot (Vtx + Vty + (x4 + y4 - x' + y') \cdot \omega z)$$

c) 公式修正

在这里再考虑电机正转时候的正方向, 则修改后的公式如下。(判断方法: 正视电机输出转轴, 逆时针转动方向为正输出值的方向。3,4 号公式取“-”)

$$n1 = 19 \cdot \frac{60}{\pi d} \cdot (Vtx - Vty - (x1 + y1 - x' - y') \cdot \omega z)$$

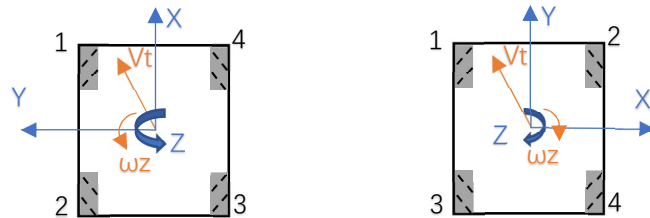
$$n2 = 19 \cdot \frac{60}{\pi d} \cdot (Vtx + Vty - (x2 + y2 + x' - y') \cdot \omega z)$$

$$n3 = -19 \cdot \frac{60}{\pi d} \cdot (Vtx - Vty + (x3 + y3 + x' + y') \cdot \omega z)$$

$$n4 = -19 \cdot \frac{60}{\pi d} \cdot (Vtx + Vty + (x4 + y4 - x' + y') \cdot \omega z)$$

4) 坐标系转换

推导公式的时候, 坐标系与我代码中定义的不一样, 以及对麦轮的编号也不一样。因此这里的公式并不能直接生成对应的代码, 而是需要转换。



上面左图是推导公式的时候所规定的坐标系, 右图是我的代码中所规定的的坐标系。

因此得到替换规则如下:

a) X -> Y

- b) $Y \rightarrow -X$
 c) $Z \rightarrow -Z$ (实际上是 $\omega z \rightarrow -\omega z$)
 d) 下标① \rightarrow ①; ② \rightarrow ③; ③ \rightarrow ④; ④ \rightarrow ②

替换之前的公式如下:

$$\begin{aligned}n1 &= 19 \cdot \frac{60}{\pi d} \cdot (Vtx - Vty - (x1 + y1 - x' - y') \cdot \omega z) \\n2 &= 19 \cdot \frac{60}{\pi d} \cdot (Vtx + Vty - (x2 + y2 + x' - y') \cdot \omega z) \\n3 &= -19 \cdot \frac{60}{\pi d} \cdot (Vtx - Vty + (x3 + y3 + x' + y') \cdot \omega z) \\n4 &= -19 \cdot \frac{60}{\pi d} \cdot (Vtx + Vty + (x4 + y4 - x' + y') \cdot \omega z)\end{aligned}$$

替换之后的公式如下:

$$\begin{aligned}n1 &= 19 \cdot \frac{60}{\pi d} \cdot (Vty + Vtx + (y1 - x1 - y' + x') \cdot \omega z) \\n3 &= 19 \cdot \frac{60}{\pi d} \cdot (Vty - Vtx + (y3 - x3 + y' + x') \cdot \omega z) \\n4 &= -19 \cdot \frac{60}{\pi d} \cdot (Vty + Vtx - (y4 - x4 + y' - x') \cdot \omega z) \\n2 &= -19 \cdot \frac{60}{\pi d} \cdot (Vty - Vtx - (y2 - x2 - y' - x') \cdot \omega z)\end{aligned}$$

5. 底盘正运动学模型

1) 问题描述

已知四个轮子的转速, 求解底盘速度。

2) 问题求解

求解正向运动学模型就是要找到一个 3×4 矩阵 $[F]$ 使其满足:

$$[F] \cdot [\Omega] \cdot r = [V]$$

即给定四个轮子的转速 $[\Omega]$, 需要找到底盘的速度 $[V]$ 。

这个问题通常来说无解, 因为它是一个超定的线性方程组。这个的物理意义是: 如果给定四个轮子的转速, 有可能会有轮子打滑而底盘没有运动的情况出现。

(比如, 前面两个轮子向后转, 后面两个轮子向前转, 此时会因为轮子的速度方向冲突而发生打滑现象)。不过我们还是能通过最小二乘法来提供最优解的矩阵 $[F]$ 。

① 以逆动力学方程开始

$$[\Omega] \cdot r = [R] \cdot [V]$$

② 方程两边同时乘以 $[R]'$ ($[R]$ 的转置矩阵)

$$[R]' \cdot [\Omega] \cdot r = [R]' \cdot [R] \cdot [V]$$

③ 方程两边同时乘以 $([R]' \cdot [R])^{-1}$

$$([R]' \cdot [R])^{-1} \cdot [R]' \cdot [\Omega] \cdot r = ([R]' \cdot [R])^{-1} \cdot [R]' \cdot [R] \cdot [V]$$

$$([R]' \cdot [R])^{-1} \cdot [R]' \cdot [\Omega] \cdot r = [V]$$

④ 令 $[F] = ([R]' \cdot [R])^{-1} \cdot [R]'$

$$[F] \cdot [\Omega] \cdot r = [V]$$

$$[\Omega] \cdot r = [R] \cdot [V]$$

使用简化后的逆矩阵[R]，正运动学中的矩阵(伪逆矩阵)[F]可以被求解出来
(这部分跟超定方程组的求解问题有关，涉及数学，有兴趣的请自行搜索，这里直接引用别人的结论)：

$$[R] = \begin{bmatrix} 1 & -1 & -K \\ 1 & 1 & -K \\ 1 & -1 & K \\ 1 & 1 & K \end{bmatrix}$$

$$[F] = \begin{bmatrix} 1/4 & 1/4 & 1/4 & 1/4 \\ -1/4 & 1/4 & -1/4 & 1/4 \\ -1/(4K) & -1/(4K) & 1/(4K) & 1/(4K) \end{bmatrix}$$

3) 结论扩展

a) n->ω 的转换

$$\omega = rad/s$$

$$n = r/min$$

$$\omega/2\pi = r/s$$

$$\omega/2\pi * 60 = n$$

b) 旋转中心调节

$$r1 = -(x1 + y1 - x' - y')$$

$$r2 = -(x2 + y2 + x' - y')$$

$$r3 = (x3 + y3 + x' + y')$$

$$r4 = (x4 + y4 - x' + y')$$

$$[R] = \begin{bmatrix} 1 & -1 & -r1 \\ 1 & 1 & -r2 \\ 1 & -1 & r3 \\ 1 & 1 & r4 \end{bmatrix}$$

$$[F] = \begin{bmatrix} 1/4 & 1/4 & 1/4 & 1/4 \\ -1/4 & 1/4 & -1/4 & 1/4 \\ -1/(4 \cdot r1) & -1/(4 \cdot r2) & 1/(4 \cdot r3) & 1/(4 \cdot r4) \end{bmatrix}$$

c) 底盘姿态解算之 Vx、Vy、ωz

- n'为定子转速(rpm)，n 为转子转速(rpm)，1/19 为 M3508 电机的减速比
- 若麦轮周长和 x, y 的单位均为 mm，则 Vx, Vy 的单位为 mm/s，ωz 的单位为 rad/s

i. Vx

$$Vx = (\frac{1}{4} \cdot \omega1 + \frac{1}{4} \cdot \omega2 + \frac{1}{4} \cdot \omega3 + \frac{1}{4} \cdot \omega4) \cdot r$$

$$Vx = \frac{2\pi r}{60} \cdot (\frac{1}{4} \cdot n1' + \frac{1}{4} \cdot n2' + \frac{1}{4} \cdot n3' + \frac{1}{4} \cdot n4')$$

$$Vx = \frac{\text{麦轮周长}}{60} \cdot (\frac{1}{4} \cdot n1' + \frac{1}{4} \cdot n2' + \frac{1}{4} \cdot n3' + \frac{1}{4} \cdot n4')$$

$$Vx = \frac{1}{19} \cdot \frac{\text{麦轮周长}}{4 \cdot 60} \cdot (n1 + n2 + n3 + n4)$$

ii. Vy

$$Vy = (-\frac{1}{4} \cdot \omega1 + \frac{1}{4} \cdot \omega2 - \frac{1}{4} \cdot \omega3 + \frac{1}{4} \cdot \omega4) \cdot r$$

... ↓ (推导方式同 Vx)

$$Vy = \frac{1}{19} \cdot \frac{\text{麦轮周长}}{4 \cdot 60} \cdot (-n1 + n2 - n3 + n4)$$

iii. Ωz

$$\omega z = (-\frac{1}{4 \cdot r1} \cdot \omega1 - \frac{1}{4 \cdot r2} \cdot \omega2 + \frac{1}{4 \cdot r3} \cdot \omega3 + \frac{1}{4 \cdot r4} \cdot \omega4) \cdot r$$

... ↓ (推导方式同 Vx)

$$\omega z = \frac{1}{19} \cdot \frac{\text{麦轮周长}}{4 \cdot 60} \cdot (-\frac{n1}{r1} - \frac{n2}{r2} + \frac{n3}{r3} + \frac{n4}{r4})$$

iv. 公式修正

考虑电机正转时反馈的转速符号，上述公式修改后如下：

$$Vx = \frac{1}{19} \cdot \frac{\text{麦轮周长}}{4 \cdot 60} \cdot (n1 + n2 - n3 - n4)$$

$$Vy = \frac{1}{19} \cdot \frac{\text{麦轮周长}}{4 \cdot 60} \cdot (-n1 + n2 + n3 - n4)$$

$$\omega z = \frac{1}{19} \cdot \frac{\text{麦轮周长}}{4 \cdot 60} \cdot (-\frac{n1}{r1} - \frac{n2}{r2} - \frac{n3}{r3} - \frac{n4}{r4})$$

d) 底盘姿态解算之 Pos_x 、 Pos_y 、 Agl_z

- $Mn.agl_sum$ 表示 n 号轮累计转过的机械角度， $Mn.agl_dlt$ 表示 n 号轮在 Δt 时间内转过的机械角度。这里的机械角度由 M3508 电机反馈的角度 (0~8191) 计算所得。

i. Pos_x

$$Vx = (\frac{1}{4} \cdot \omega1 + \frac{1}{4} \cdot \omega2 + \frac{1}{4} \cdot \omega3 + \frac{1}{4} \cdot \omega4) \cdot r$$

两边对时间积分

$$Pos_x = (\frac{1}{4} \cdot \int \omega1 dt + \frac{1}{4} \cdot \int \omega2 dt + \frac{1}{4} \cdot \int \omega3 dt + \frac{1}{4} \cdot \int \omega4 dt) \cdot r$$

$$Pos_x = \frac{1}{4} (\alpha1 + \alpha2 + \alpha3 + \alpha4) \cdot r$$

$$Pos_x = \frac{1}{4} \cdot 2\pi r \cdot (\frac{\alpha1}{2\pi} + \frac{\alpha2}{2\pi} + \frac{\alpha3}{2\pi} + \frac{\alpha4}{2\pi})$$

$$\frac{\alpha1}{2\pi} = \frac{M1.agl_sum}{8192} \cdot \frac{1}{19} = 1 \text{ 号轮转过的圈数, 对 2、3、4 号轮也适用}$$

$$Pos_x = \frac{1}{4} \cdot \text{麦轮周长} \cdot \left(\frac{M1.agl_sum}{8192 \cdot 19} + \frac{M2.agl_sum}{8192 \cdot 19} + \frac{M3.agl_sum}{8192 \cdot 19} + \frac{M4.agl_sum}{8192 \cdot 19} \right)$$

$$Pos_x = \frac{1}{19} \cdot \frac{\text{麦轮周长}}{4 \cdot 8192} \cdot (M1.agl_sum + M2.agl_sum + M3.agl_sum + M4.agl_sum)$$

对于离散系统来说，可以周期执行下列表达式来计算位移 Pos_x

$$Pos_x += \frac{1}{19} \cdot \frac{\text{麦轮周长}}{4 \cdot 8192} \cdot (M1.agl_dlt + M2.agl_dlt + M3.agl_dlt + M4.agl_dlt)$$

ii. Pos_y

$$Vy = \left(-\frac{1}{4} \cdot \omega_1 + \frac{1}{4} \cdot \omega_2 - \frac{1}{4} \cdot \omega_3 + \frac{1}{4} \cdot \omega_4 \right) \cdot r$$

... ↓ (推导方式同 Pos_x)

$$Pos_y = \frac{1}{19} \cdot \frac{\text{麦轮周长}}{4 \cdot 8192} \cdot (-M1.agl_sum + M2.agl_sum - M3.agl_sum + M4.agl_sum)$$

对于离散系统来说，可以周期执行下列表达式来计算位移 Pos_y

$$Pos_y += \frac{1}{19} \cdot \frac{\text{麦轮周长}}{4 \cdot 8192} \cdot (-M1.agl_dlt + M2.agl_dlt - M3.agl_dlt + M4.agl_dlt)$$

iii. Agl_z

$$\omega_z = \left(-\frac{1}{4 \cdot r_1} \cdot \omega_1 - \frac{1}{4 \cdot r_2} \cdot \omega_2 + \frac{1}{4 \cdot r_3} \cdot \omega_3 + \frac{1}{4 \cdot r_4} \cdot \omega_4 \right) \cdot r$$

... ↓ (推导方式同 Pos_x)

$$Rad_z = \frac{1}{19} \cdot \frac{\text{麦轮周长}}{4 \cdot 8192} \cdot \left(-\frac{M1.agl_sum}{r_1} - \frac{M2.agl_sum}{r_2} + \frac{M3.agl_sum}{r_3} + \frac{M4.agl_sum}{r_4} \right)$$

$$Agl_z = \frac{1}{19} \cdot \frac{\text{麦轮周长}}{4 \cdot 8192} \cdot \left(-\frac{M1.agl_sum}{r1} - \frac{M2.agl_sum}{r2} + \frac{M3.agl_sum}{r3} + \frac{M4.agl_sum}{r4} \right) \cdot 57.3 \text{ } ^\circ/\text{rad}$$

iv. 公式修正

考虑电机正转时反馈的转速符号，上述公式修改后如下：

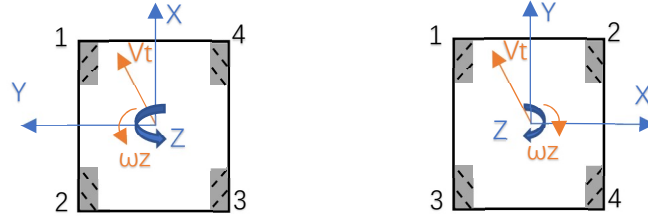
$$Pos_{x+} = \frac{1}{19} \cdot \frac{\text{麦轮周长}}{4 \cdot 8192} \cdot (M1.agl_dlt + M2.agl_dlt - M3.agl_dlt - M4.agl_dlt)$$

$$Pos_{y+} = \frac{1}{19} \cdot \frac{\text{麦轮周长}}{4 \cdot 8192} \cdot (-M1.agl_dlt + M2.agl_dlt + M3.agl_dlt - M4.agl_dlt)$$

$$Agl_z = \frac{1}{19} \cdot \frac{\text{麦轮周长}}{4 \cdot 8192} \cdot \left(-\frac{M1.agl_sum}{r1} - \frac{M2.agl_sum}{r2} - \frac{M3.agl_sum}{r3} - \frac{M4.agl_sum}{r4} \right) \cdot 57.3 \text{ } ^\circ/\text{rad}$$

4) 坐标系转换

推导公式的时候，坐标系与我代码中定义的不一样，以及对麦轮的编号也不一样。因此这里的公式并不能直接生成对应的代码，而是需要转换。



上面左图是推导公式的时候所规定的坐标系，右图是我的代码中所规定的的坐标系。

因此得到**替换规则**如下：

- e) X -> Y
- f) Y -> -X
- g) Z -> -Z (实际上是 $\omega_z \rightarrow -\omega_z$)
- h) 下标①->①; ②->③; ③->④; ④->②

替换之前的公式如下：

$$Vx = \frac{1}{19} \cdot \frac{\text{麦轮周长}}{4 \cdot 60} \cdot (n1 + n2 - n3 - n4)$$

$$Vy = \frac{1}{19} \cdot \frac{\text{麦轮周长}}{4 \cdot 60} \cdot (-n1 + n2 + n3 - n4)$$

$$\begin{aligned}
\omega_z &= \frac{1}{19} \cdot \frac{\text{麦轮周长}}{4 \cdot 60} \cdot \left(-\frac{n_1}{r_1} - \frac{n_2}{r_2} - \frac{n_3}{r_3} - \frac{n_4}{r_4} \right) \\
Pos_x &+= \frac{1}{19} \cdot \frac{\text{麦轮周长}}{4 \cdot 8192} \cdot (M1.agl_dlt + M2.agl_dlt - M3.agl_dlt \\
&\quad - M4.agl_dlt) \\
Pos_y &+= \frac{1}{19} \cdot \frac{\text{麦轮周长}}{4 \cdot 8192} \cdot (-M1.agl_dlt + M2.agl_dlt + M3.agl_dlt \\
&\quad - M4.agl_dlt) \\
Agl_z &= \frac{1}{19} \cdot \frac{\text{麦轮周长}}{4 \cdot 8192} \\
&\quad \cdot \left(-\frac{M1.agl_sum}{r_1} - \frac{M2.agl_sum}{r_2} - \frac{M3.agl_sum}{r_3} \right. \\
&\quad \left. - \frac{M4.agl_sum}{r_4} \right) \cdot 57.3^\circ / rad \\
r_1 &= -(x_1 + y_1 - x' - y') \\
r_2 &= -(x_2 + y_2 + x' - y') \\
r_3 &= (x_3 + y_3 + x' + y') \\
r_4 &= (x_4 + y_4 - x' + y')
\end{aligned}$$

替换之后的公式如下：

$$\begin{aligned}
V_y &= \frac{1}{19} \cdot \frac{\text{麦轮周长}}{4 \cdot 60} \cdot (n_1 + n_3 - n_4 - n_2) \\
V_x &= -\frac{1}{19} \cdot \frac{\text{麦轮周长}}{4 \cdot 60} \cdot (-n_1 + n_3 + n_4 - n_2) \\
\omega_z &= \frac{1}{19} \cdot \frac{\text{麦轮周长}}{4 \cdot 60} \cdot \left(\frac{n_1}{r_1} + \frac{n_3}{r_3} + \frac{n_4}{r_4} + \frac{n_2}{r_2} \right) \\
Pos_y &+= \frac{1}{19} \cdot \frac{\text{麦轮周长}}{4 \cdot 8192} \cdot (M1.agl_dlt + M3.agl_dlt - M4.agl_dlt \\
&\quad - M2.agl_dlt) \\
Pos_x &+= -\frac{1}{19} \cdot \frac{\text{麦轮周长}}{4 \cdot 8192} \cdot (-M1.agl_dlt + M3.agl_dlt \\
&\quad + M4.agl_dlt - M2.agl_dlt) \\
Agl_z &= \frac{1}{19} \cdot \frac{\text{麦轮周长}}{4 \cdot 8192} \\
&\quad \cdot \left(\frac{M1.agl_sum}{r_1} + \frac{M3.agl_sum}{r_3} + \frac{M4.agl_sum}{r_4} \right. \\
&\quad \left. + \frac{M2.agl_sum}{r_2} \right) \cdot 57.3^\circ / rad
\end{aligned}$$

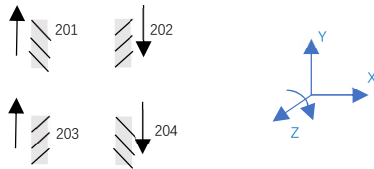
这里特别说明一下：在旋转中心调节这一部分规定了——(x_n, y_n)均为正数，而(x', y')本身是带符号的，因此替换的时候 x_n, y_n 只需修改下标，而 x', y' 则与坐标轴的方向有关，表示在当前坐标系下重心的位置。r 表达式中括号外最前面的符号与 ω_z 的方向有关系，当 ω_z 方向取反时表达式整个取反。综上所述得到下面替换后的公式。

$$\begin{aligned}
r_1 &= (x_1 + y_1 + x' - y') \\
r_3 &= (x_3 + y_3 + x' + y')
\end{aligned}$$

$$r4 = (x4 + y4 - x' + y')$$

$$r2 = (x2 + y2 - x' - y')$$

二、 模式分类



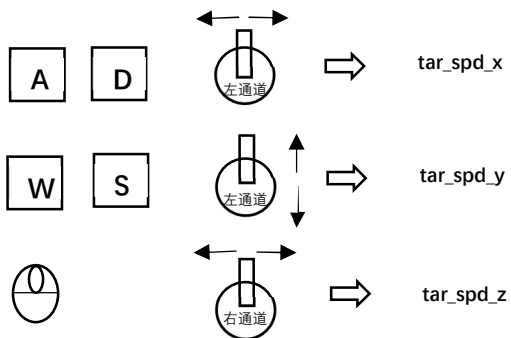
底盘控制模型，四个电调输出值均为正的时候，四个电机的转动方向。
XYZ 的正方向如图所示。

底盘全向分配算法：

1. M201.tar_spd = +tar_spd_x +tar_spd_y + tar_spd_z
2. M202.tar_spd = +tar_spd_x -tar_spd_y + tar_spd_z
3. M203.tar_spd = -tar_spd_x +tar_spd_y + tar_spd_z
4. M204.tar_spd = -tar_spd_x -tar_spd_y + tar_spd_z

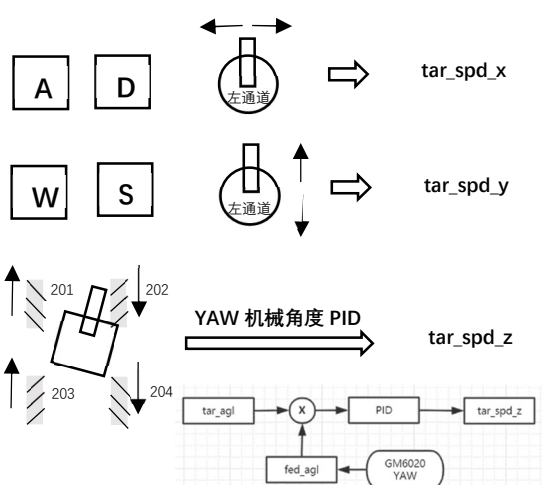
1. 机械模式

主动-底盘，从动-云台。（头跟底盘）



2. 陀螺仪模式

主动-云台，从动-底盘。（底盘跟头）



三、机械模式

1. 手动爬坡

在功率限制下，爬坡比较费力。因此，往往需要借助超级电容才能够快速地爬坡。在这里主要讨论的是手动爬坡，即不借助超级电容下的爬坡模式。

根据前辈们的经验，有两种优化的方式。

1) 四个麦轮的目标速度设置成一个固定值。

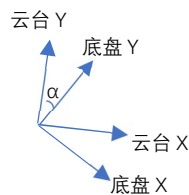
2) 前轮最大输出值缩小，后轮最大输出值保持。

上坡的时候由于重心后移，前轮打滑会浪费功率。因此，在功率限制的情况下我们可以将前轮的部分功率分配给后轮。

四、陀螺仪模式

1. 坐标系转换

当云台和底盘分开控制时，我们存在两个坐标系：分别为云台坐标系（操作手坐标系）和底盘坐标系。定义 Y 为前进正方向，X 为右平移正方向。定义 α 角为两个坐标系的夹角，范围 $[0,360^\circ]$ 。



利用旋转矩阵，我们可以得到两个坐标系的转换关系：

```
1. /* 旋转矩阵 将云台坐标系的期望速度 转换到 底盘坐标系的期望速度 */
2. chas_tar_spd[ x ] = cos(α) * tar_spd_x - sin(α) * tar_spd_y;
3. chas_tar_spd[ y ] = sin(α) * tar_spd_x + sin(α) * tar_spd_y;
```

tar_spd_x 为用户期望的左右平移速度（带符号）。

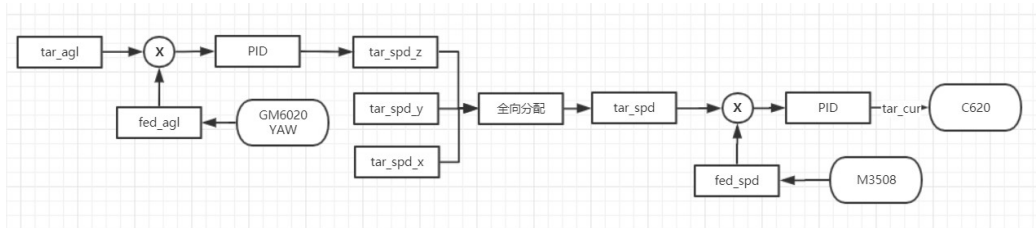
tar_spd_y 为用户期望的前后平移速度（带符号）。

α 的计算其实有两种等效方式，一种是将机械中值作为 0° 轴，然后以底盘 z 运动正方向作为角度增加正方向，计算出来的 α 范围在 $[0,360^\circ]$ 。另一种是直接利用 YAW 轴电机反馈机械角度-机械中值，计算出来的 α 有正负。这里的机械中值指的是云台的规定指向或者所谓的初始角度。

在陀螺仪模式下一直会进行坐标系的转换。这样能有效地将操作手坐标系中下达的移动指令变换到底盘坐标系中，使底盘的速度分配（大小和方向）满足第一视角的操作。

2. 串级 PID

YAW 机械角度为外环角度环，输出 tar_spd_z。底盘电机转速为内环速度环，输出 tar_cur。期望速度利用全向分配算法计算得到。



下面是 Z 方向的外环角度环示例代码。

```

1.  /**
2.   * @brief  底盘陀螺仪模式加上 Z 方向的速度环(实际上是位置环)
3.   * @note
4.   *       将当前 YAW 机械角和中值 YAW 机械角的差值作为误差送进 Z_PID 控制器。
5.   *       反馈期望的角速度 tar_spd_z
6.   */
7. float CHASSIS_Z_Angle_PidCalc(PID_Object_t *pid, float kp)
8. {
9.     pid->kp = kp;
10.    pid->erro = pid->target - pid->feedback;
11.
12.    /* 临界处理 */
13.
14.    /*
15.
16.        0 - 8191
17.
18.        误差=期望-反馈
19.
20.        误差大于 4096
21.
22.        反馈落后于期望 -> 确定期望的方向 0->8191 --》 反馈大于期望
23.
24.        实际误差 = - (8191 - 期望 + 反馈) = -8191 + 误差
25.
26.        误差小于-4096
27.
28.        反馈落后于期望 -> 确定期望方向 8191->0 --》 期望大于反馈
29.
30.        实际误差 = 8191 + 期望 -反馈 = 8191 + (期望 - 反馈) = 8191 + 误差
31.    */
32.
33.
34.    if(pid->erro >= 4096) {

```

```

35.         pid->erro = -8192 + pid->erro;
36.     }
37.     else if(pid->erro <= -4096) {
38.         pid->erro = 8192 + pid->erro;
39.     }
40.
41.     pid->erro = KalmanFilter(&Chassis_Kalman_Error, pid->erro);
42.
43.     /* 底盘处于普通跟随模式 */
44.     if(CHASSIS_IfTopGyroOpen() == false) {
45.         // 死区控制算法
46.         if(abs(pid->erro) < 15) {
47.             pid->erro = 0;
48.         }
49.     }
50.     /* 底盘处于小陀螺模式 */
51.     else {
52.         //..不做死区处理
53.     }
54.
55.     /* Pout */
56.     pid->pout = pid->kp * pid->erro;
57.     if(abs(pid->erro) > SPIN_ANGLE) {
58.         /* Dout*/
59.         pid->dout = pid->kd * (pid->erro - pid->last_erro)/0.002f;
60.     } else {
61.         pid->dout = 0;
62.     }
63.
64.     /* Record Last Error */
65.     pid->last_erro = pid->erro;
66.
67.     /* Total PID Output*/
68.     pid->out = pid->pout + pid->dout;
69.     /* Total PID Output Limits */
70.     pid->out = constrain(pid->out, -
        Chas_Spin_Move_Max, Chas_Spin_Move_Max);
71.     return pid->out;
72. }

```

3. 旋转因子

在常规跟随/小陀螺/扭腰的情况下平移时，我们需要权衡好旋转和平移的分配量。

因此，我们可以设置一个旋转因子来自动调节他们之间的关系。这个旋转因子的核心思想是：当底盘旋转得越快，平移量越小。相反，底盘旋转得越慢，平移量越大。

```
1.  /* 计算旋转因子(通过调整进入条件来设置旋转和平移的分配比例) */
2.  if(abs(Chas_Target_Speed[ Z ]) > 800) {
3.      // 扭头速度越快，平移速度越慢
4.      k_rc_z = ((Chas_Spin_Move_Max - (abs(Chas_Target_Speed[ Z ]) - 800)) * (
        Chas_Spin_Move_Max - (abs(Chas_Target_Speed[ Z ]) - 800)))
5.          / (Chas_Spin_Move_Max * Chas_Spin_Move_Max);
6.
7.      k_rc_z = constrain(k_rc_z, 0.f, 1.f);
8.  } else {
9.      k_rc_z = 1.f;
10. }
11.
12. /* 期望速度计算 */
13. Chas_Target_Speed[ Y ] = Chas_Target_Speed[ Y ] * k_rc_z;
14. Chas_Target_Speed[ X ] = Chas_Target_Speed[ X ] * k_rc_z;
```

4. 常规跟随

利用上面的串级 PID 思路，将外环的期望角度设定为 YAW 机械中值，即可实现最基本的底盘跟随效果。

5. 小陀螺

小陀螺的实现目前有两种思路：串级 PID，单级 PID。

1) 串级 PID

我们套用常规跟随的串级 PID 控制模型，只不过在小陀螺里面，我们将外环的期望角度进行变化，而不再是固定的机械中值。这样，底盘会随着外环期望角度的变化而旋转起来。(目前采用这种方式)

① 匀速小陀螺

期望角度以斜坡方式变化，从而实现匀速小陀螺。通过调节斜坡步长，我们可以调节匀速小陀螺的速度。

② 非匀速小陀螺

期望角度以非斜坡方式变化，如正弦方式变化，则我们可以实现正弦小陀螺。

注意

- 期望角度在边界上的处理方式，即 target: 8191->0 或者 0->8191，保证 target 在合理的范围内变化。
- 云台与底盘需要独立控制。云台的运动会导致底盘角度环反馈的变化，因此我们可以利用陀螺仪数据来做补偿，很大程度上可以解决云台在转动时对底盘小陀螺速度的影响。

```

1. yaw[NOW] = Mpu_Info.yaw;
2. yaw[DELTA] = yaw[NOW] - yaw[PREV];
3. if(yaw[DELTA] > +180.f) {
4.     yaw[DELTA] = -360.f + yaw[DELTA];
5. } else if(yaw[DELTA] < -180.f) {
6.     yaw[DELTA] = +360.f + yaw[DELTA];
7. }
8. yaw[PREV] = yaw[NOW];
9.
10. /* 利用陀螺仪数据作扭头补偿(使得云台与底盘独立运动) */
11. if(CHASSIS_IfTopGyroOpen() == true) {
12.     Chassis_Z_PID.Angle.target = CHASSIS_MECH_YawBoundaryProc(&Chassis_Z_PID, yaw[DELTA] * 8192 / 360.f);
13. }

```

```

1. /**
2.  * @brief 机械模式下云台 YAW 期望值边界处理
3.  */
4. float CHASSIS_MECH_YawBoundaryProc(Chassis_Z_PID_t *pid, float delta_target)
5. {
6.     float target;
7.     target = pid->Angle.target + delta_target;
8.
9.     if(target > 8191.f) {
10.         target = (target - 8191.f);
11.     } else if(target < 0.f) {
12.         target = (target + 8191.f);
13.     }
14.
15.     return target;
16. }

```

之所以目前采用这种方式，原因如下。

首先这个控制模型与常规跟随的控制模型是一样的，兼容性更好，因此小陀螺和常规跟随的切换代码十分简单，而且小陀螺->常规控制时会根据 PID 控制器中对误差的处理方式就近回去。其次，我们可以很方便地设定底盘逻辑，从而实现扭头就跑（头调转 180°，底盘逻辑调转）。

需要注意，当底盘刚被卡住的时候，角度环的误差会随之增大，此时输出的期望旋转速度也会增大。当误差较大时，此时容易造成底盘平移不顺畅的感觉，即旋转量分配较大，而平移量分配较小。此处可结合后面的底盘堵转处理做特殊优化。

2) 单级 PID

不经过外环 PID 计算直接给定 tar_spd_z。这样的话云台和底盘的运动完全独立，同样可以通过调整给定的 tar_spd_z 来调整小陀螺的速度。

6. 扭腰

扭腰可以看成是小陀螺的一种特例，分析方法类似。

7. 底盘卡死处理

在实际比赛中，机器人之间难免有碰撞和围堵，此时底盘电机容易出现堵转的状态。为防止底盘电机持续高功率输出，我们需要考虑如何做底盘卡死处理，以及此时底盘与云台之间的配合关系。

1) 判断底盘被卡死的方式

① 模仿拨盘电机的卡弹判断思路

判断四个底盘电机速度环的 pid 输出值以及速度反馈。

```
if 速度环 pid 输出值较大 and 电机转子反馈速度较小 then
    if ++stuck_cnt>500 then
        底盘卡死
    endif
else
    stuck_cnt=0
    底盘正常
endif
```

只有四个轮子均满足上述的判断条件才认为底盘被卡死。

2) 底盘被卡死后的处理方式

① 尝试挣脱

```
if 底盘卡死 then
    云台灵敏度↑
    if state==小陀螺 then
        if ++try_cnt>500 then
            限制底盘最大输出
        endif
        if stuck_cnt>500 then
            stuck_cnt=0
            小陀螺反向
        endif
    elif state==扭腰 then
        if ++try_cnt>500 then
```

```

限制底盘最大输出
endif
if stuck_cnt>500 then
    stuck_cnt=0
    扭腰反向
endif
elif state==常规跟随 then
    限制底盘最大输出
endif
else
    try_cnt=0
    云台灵敏度恢复
endif
endif

```

云台灵敏度增大是方便操作手在这种情况下查看围堵的敌方，易于快速反击。而限制底盘最大输出是防止底盘长时间堵转，大功率输出。

② 其他

五、 功率限制

1. 功率计算公式推导

参考资料：

1. RM2019 总决赛 第 68 场 哈工大 VS 港科 技术复盘中有简单提及功率控制

- 功率(W) = 扭力(N) * 速度(m/s)
 $P = F * V$
- 扭矩(N*m) = 扭力(N) * 力臂(m)
 $T = F * R$
- 功率(W) = 扭矩(N*m) * 转速(rpm) * $\pi/30$
 $P = T * n * \pi/30$

推导过程：

$$P = T/R * V$$

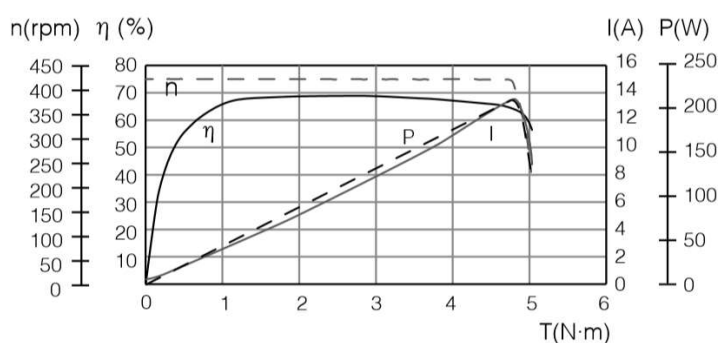
$$V = \text{周长(m)} * \text{转/秒}$$

$$= \text{周长(m)} * \text{转/分钟} / 60$$

$$= 2\pi * R * n / 60 \quad n \text{ 的单位为 rpm}$$

下图摘自 RM C620 无刷电机调速器使用说明 V1.01

搭配 M3508 直流无刷减速电机时的电机性能曲线



P- 输出功率，I- 电流， η - 效率，T- 扭矩，n- 转速

2. 功率、扭矩和转速之间的关系讨论

参考资料：

1. 纯干货！谈谈马力、扭矩和转速那点事

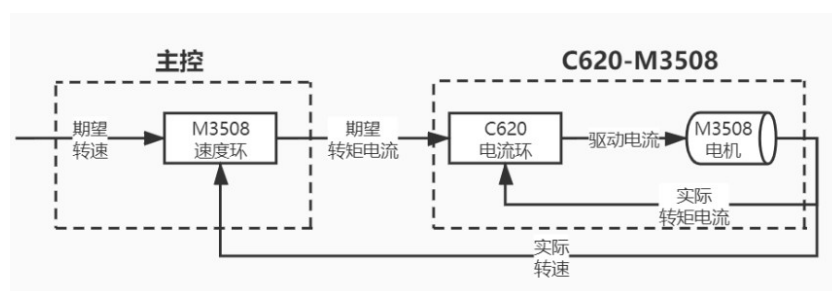
<http://www.360che.com/tech/190625/113224.html>

2. 直流电机之损耗与效率

<https://max.book118.com/html/2019/0114/5212210244002001.shtm>

扭矩决定（角）加速度，而电机转速决定速度。**功率限制**意味着功率存在一个最大上限，也就是**限制了步兵的加速度和速度**。

通常，我们底盘电机使用的是速度环 PID，也就是说我们的控制对象是底盘电机的转速，从而控制整个底盘的运动速度。要想了解功率、扭矩和转速之间的关系，我们可以先回顾一下刚进队调单个电机的速度环 PID 时，将期望速度设定为 0，用手去扭动电机转子部分，我们会感受到电机的阻力，而将期望速度设定为某一定值时再用手去尝试轻轻“握”住正在转动的电机转子部分，我们会感受到电机的“脾气”。那么这个力气是怎么来的呢？实际上，我们存在如下的控制链路：



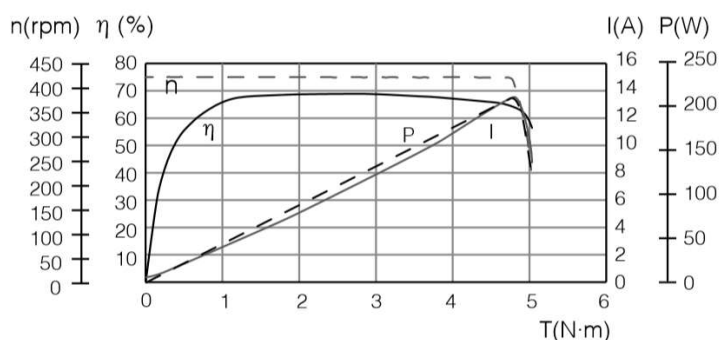
当底盘电机因为阻力导致转速下降时，速度环 PID 计算得到的期望转矩电流会增大，再经过电流环 PID 计算输出的驱动电流也会增大，从而 3508 电机会输出更大的扭矩来抵消我们的阻力来维持我们的转速，这里的抵消并非完全抵消。

同样地，对于整个底盘来说，当我们用脚去推底盘的时候，底盘速度环 PID 同样会为了维持速度而输出更大的扭矩来抵消阻力。或者这样理解，当我们用脚去推动底盘的

时候，底盘电机因为这个外力产生扭矩，而扭矩会产生角加速度，即底盘有加速的趋势，但此时我们要控制底盘维持原来的转速（静止状态下转速约等于 0），则电机输出扭矩来抵抗这个外力。

通过上面这个例子，简要地说明了速度环 PID 下扭矩与转速之间的关系。那扭矩与功率，转速与功率之间又是什么关系呢？我们再来看看手册里面的这张电机性能曲线图。

搭配 M3508 直流无刷减速电机时的电机性能曲线



P- 输出功率，I- 电流， η - 效率，T- 扭矩，n- 转速

该图是 M3508 电机搭配 C620 电调做速度闭环控制时的曲线。从图上我们可以看出，在转速稳定的时候，T-扭矩与 P-输出功率呈正比关系，简单来说就是电机的输出扭矩越大，输出功率越大。而电机的输出扭矩又是由电调的输出转矩电流来控制的。

这张图上有一条曲线表示效率 η （电调和电机构成的系统），效率的计算法则是：效率 η =输出功率 P_{out} /输入功率 P_{in} $\times 100\%$ 。由于存在损耗，输出功率恒小于输入功率，即效率一定小于 100%。对于电机来说，输出功率（机械功率）可由 $P=T*\omega$ 求得（请回看 1、功率计算公式推导部分）。另外，请注意图上的电流指的是输入电流 I_{in} ，如果认为输入电压恒定为 24V，则可粗略计算输入功率 P_{in} =输入电压 U_{in} \times 输入电流 I_{in} 。

举个例子，我们在比赛检录区检测功率限制的做法是将机器人顶住一面墙之后开全速前进，检测功率曲线是否超过上限功率（当然也还有缓冲能量机制）。同样地，当机器人顶住墙之后麦轮还要往前转的时候所需扭矩会增大，即输出功率会增大，如果此时我们还要保持最大转速的话，肯定会超功率。为防止超功率扣血，功率限制算法闪亮登场。想想我们的祖传功率限制算法，我们的做法是判断裁判系统发送过来的剩余缓冲能量，并根据剩余缓冲能量求出一个缩放系数，来对我们速度环的 pid 输出值做限制，此时电机的转速会下降，但带来的是我们扭矩的增大。（T 扭矩 = P 功率/ ω 角速度）。

3. 启动特性分析

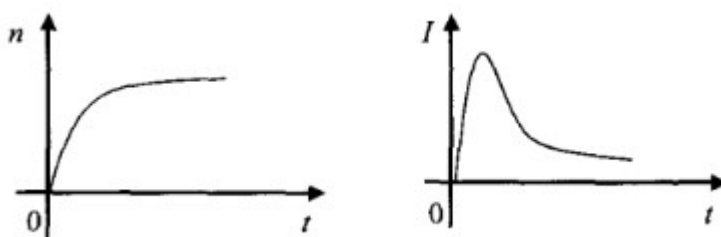
参考资料：

1. 电控大礼包-学习资料-功率控制
2. 无刷直流电机启动特性分析-江国栋

“无刷直流电机启动特性。启动过程是指电机转速从零上升到稳定值的过程，电机启动瞬间转速和每相绕组反电动势均为零。启动过程中转速和电枢电流的变化曲线便是启

动特性。

启动时电枢电流迅速增大，使电磁转矩远大于负载转矩，转速迅速增加；转速增加引起反向电动势增大，将抑制电枢电流增长，使之增长变缓直至达到最大值，然后开始减小。电枢电流减小导致电磁转矩减小，又造成转速上升变缓直至达到最大值，然后开始减小。当电磁转矩和负载转矩达到动态平衡时，转速和电枢电流稳定。无刷直流电机启动特性如图所示。



启动电流比稳定运行电流要大得多，大约是稳定运行时电流的 6 到 7 倍。对于频繁停、启应用场合，会严重影响电机的效能。”

这告诉我们一个信息，底盘在起步的时候启动电流会瞬间增大，从而会有有一个较大的启动功率。常规的处理方式就是通过缓慢加速使得启动功率变小。而功率控制的好坏体现在战车启动加速的过程时间长短。功率控制主要是把 60J 用在快速启动，电机转向突变上。

启动和制动都是比较消耗功率的行为。虽然可以通过缓慢斜坡的方式来使得功率的增加减缓，但会使得启动和制动的耗时变长。如何权衡好他们之间的关系是功率控制中需要考虑的一个重点。

4. 实时功率与缓冲能量

5. 功率限制算法

6. 优化算法

1. 换向

六、 无