



**塔克创新**  
XTARK INNOVATION

# 塔克智能机器人

机器人运动学分析教程

V1.0



更多资讯 · 欢迎关注塔克创新微信公众号

**烟台塔克电子科技有限公司**

## 版权申明

本手册版权归属塔克创新所有，并保留一切权力，受法律保护。非经(书面形式)同意，任何单位及个人不得擅自摘录或修改本手册部分或全部内容，违者我们将追究其法律责任。

## 塔克媒体

塔克官网	<a href="http://www.xtark.cn">www.xtark.cn</a>
淘宝店铺	<a href="https://xtark.taobao.com">https://xtark.taobao.com</a>
塔克哔哩	<a href="https://space.bilibili.com/511052131">https://space.bilibili.com/511052131</a>
销售邮箱	<a href="mailto:sales@xtark.cn">sales@xtark.cn</a>

## 塔克淘宝，微信公众号二维码



更多精品·欢迎关注塔克淘宝店铺



更多资讯·欢迎关注塔克创新微信公众号

机器人底盘按照转向方式的不同，可以分为两轮差速模型、四轮差速模型、阿克曼模型、全向模型等，全向模型包含麦轮，三轮全向，舵轮等。两轮差速模型、四轮差速模型、阿克曼模型也称为运动约束模型，全向模型也称为运动不受约束模型。本教程介绍机器人的几种运动学模型。

假设我们给定了机器人目标速度，这里的速度是矢量速度，包含 X 轴、Y 轴和 Z 轴三个值，其中 Z 轴为机器人旋转。机器人为了实现目标速度运动，需要将机器人的目标速度转换成每个电机的目标速度，并控制电机按照目标速度运行，最终实现机器人目标运动。机器人的目标速度转换成电机的目标速度这个过程叫“运动学分析”，运动学分析又分为前向和逆向。在运动学分析之前先来分别解释一下什么是前向运动学和逆向运动学。

前向运动学：通过机器人的各轮速度求出机器人 X 轴、Y 轴和 Z 轴方向的矢量速度。

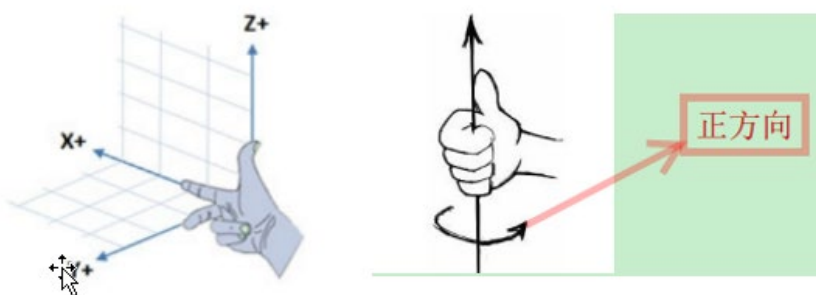
逆向运动学：通过机器人 X 轴、Y 轴和 Z 轴方向的矢量速度分别求出机器人各轮的速度。



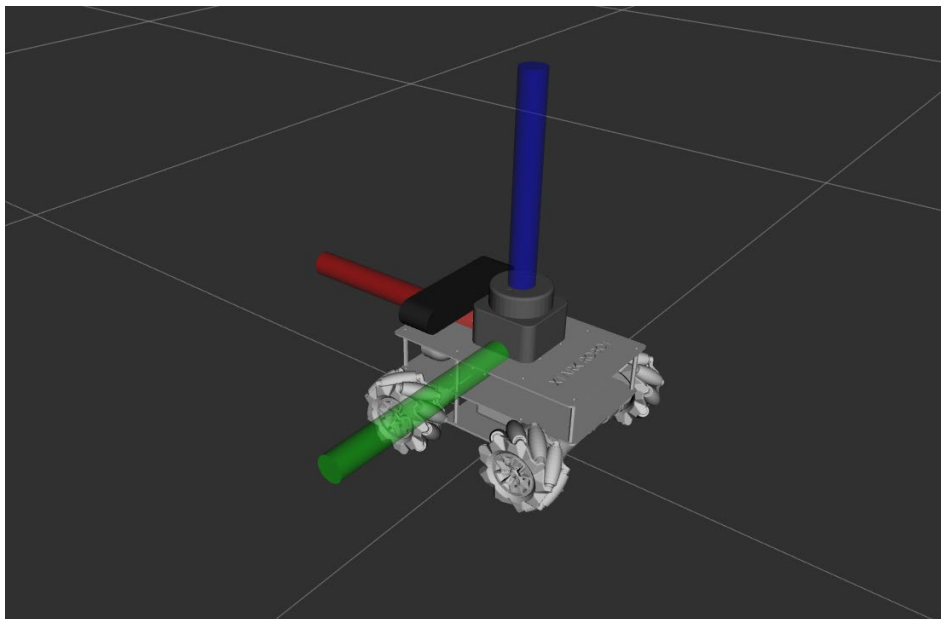
前向运动学解析出机器人实时的矢量速度，再进一步根据矢量速度计算机器人的行驶距离和角度，也就是常说的机器人里程计（Odometry）的计算，也就是航迹推演。

## 1. 预备知识

对于机器人的运动速度及坐标位姿，都需要基于坐标系来进行计算，在 ROS 系统中，坐标系遵循的是右手定则。



即对于机器人来说，X 轴的正方向为机器人正前方，Y 轴正方向为机器人左方，Z 轴正方向为逆时针旋转。其中，对于平面机器人来讲，对于 X 轴以及 Y 轴，仅考虑线速度，不考虑在 X 轴及 Y 轴上的角速度，即不考虑平面机器人的横滚及俯仰变化。对于 Z 轴，只考虑角速度，不考虑 Z 轴线速度变化，即不考虑平面机器人位置高度的变化。塔克机器人坐标定义如下图所示。



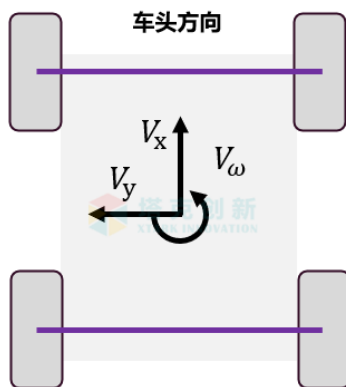
在 ROS 系统中定义了速度及坐标的标准单位，线速度是 m/s，角速度是 rad/s，坐标/距离/长度等单位是 m。

对于平面机器人，矢量速度包含如下 3 个部分。

$V_x$ ：机器人在 O 点的目标前后速度，前进为正，单位：m/s。

$V_y$ ：机器人在 O 点的目标左右速度，左移为正，单位：m/s。

$V_w$ ：机器人绕 O 点的目标旋转速度，逆时针为正，单位：rad/s。



对于两轮差速、四轮差速、阿克曼等运动受约束模型，一般  $V_y$  为 0。对于麦轮，全向轮等全向运动模型，运动不受约束，具有实际的  $V_y$  速度。

## 2. 两轮差速模型

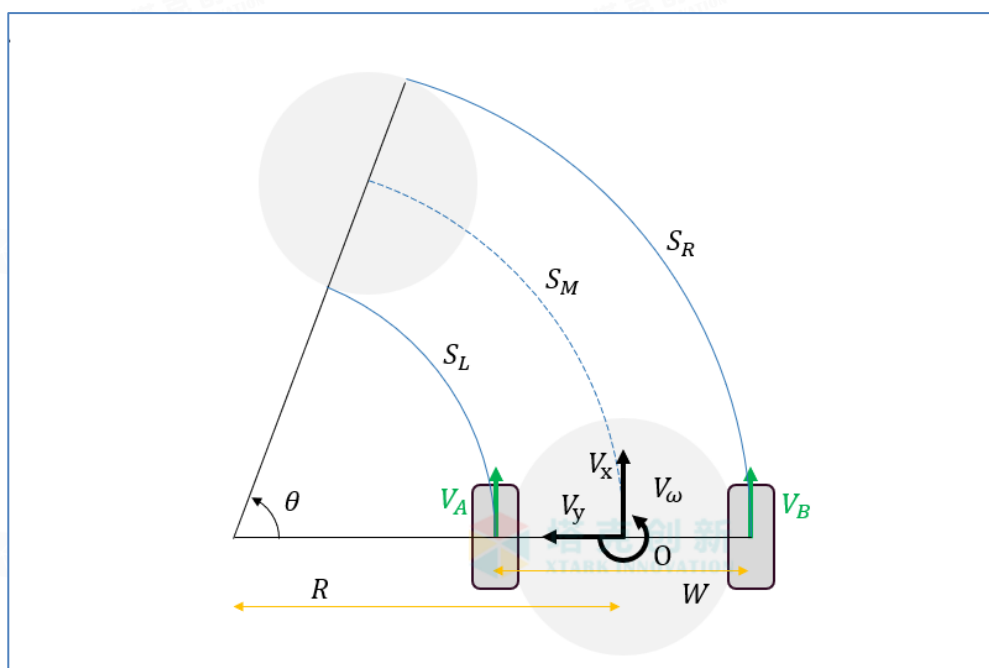
两轮差速模型可以说是最简单的底盘模型，在底盘的左右两边平行安装两个主动轮，考虑到支撑，增加一个或多个万向轮。控制两个主动轮的速度存在差异，即两轮差速，即可控制机器人实现

转向，这种结构可以实现零转弯半径，即原地旋转，旋转时的速度瞬心位于两动力轮轴线中点位置。下图是塔克两款差速机器人。



## 2.1. 运动学分析

下面我们对两轮差速式机器人进行运动学分析，分析图如下所示。



其中参数含义如下。

机器人矢量速度

$V_x$ : 机器人在 O 点的目标前后速度，前进为正，单位：m/s。

$V_y$ : 机器人在 O 点的目标左右速度，左移为正，单位：m/s。两轮忽略。

$V_\omega$ : 机器人绕 O 点的目标旋转速度，逆时针为正，单位：rad/s。

机器人参数

W: 轮距, 两个主动轮的距离, 中心为 O 点, 单位: m。

R: 机器人同时前进和旋转产生的转弯半径 R, 单位: m。

机器人轮子速度

$V_A$ : 机器人左轮速度, 前进方向为正, 单位: m/s。

$V_B$ : 机器人右轮速度, 前进方向为正, 单位: m/s。

路径

$S_L$ : 左轮在一定时间 t 内走过的路径

$S_M$ : 中间 O 点在一定时间 t 内走过的路径

$S_R$ : 右轮在一定时间 t 内走过的路径

$\theta$ : 机器人在一定时间 t 内旋转的角度, 单位: rad。

### 运动学分析

下面讨论它们之间的关系, 并求出两轮差速式机器人的运动学正逆解公式。由速度对时间的积分等于路程得。

$$S_L = V_A * t$$

$$S_M = V_x * t$$

$$S_R = V_B * t$$

由弧长除以半径等于弧度得。

$$\theta = \frac{S_L}{R - \frac{W}{2}} = \frac{S_M}{R} = \frac{S_R}{R + \frac{W}{2}}$$

$$\theta = \frac{V_A * t}{R - \frac{W}{2}} = \frac{V_x * t}{R} = \frac{V_B * t}{R + \frac{W}{2}}$$

公式两边同时除以 t, 即对时间积分得。

$$V_w = \frac{V_A}{R - \frac{W}{2}} = \frac{V_x}{R} = \frac{V_B}{R + \frac{W}{2}}$$

对上式分解, 可以到的如下三个方程:

$$R = \frac{V_x}{V_w}$$

$$\frac{V_x}{R} = \frac{V_A}{R - \frac{W}{2}}$$

$$\frac{V_x}{R} = \frac{V_B}{R + \frac{W}{2}}$$

### 逆向运动学求解

根据上述方程求解逆向运动学方程, 已知机器人目标速度  $V_x$ 、 $V_w$ , 求解驱动轮速度  $V_A$ 、 $V_B$ 。

$$V_A = V_X - \frac{W}{2} * V_w$$

$$V_B = V_X + \frac{W}{2} * V_w$$

### 前向运动学求解

根据逆向运动学方程，求解前向运动学方程。已知驱动轮的当前速度  $V_A$ ,  $V_B$ ，求当前机器人的实时速度  $V_X$ ,  $V_w$ 。

$$V_X = \frac{V_A + V_B}{2}$$

$$V_w = \frac{-V_A + V_B}{W}$$

## 2.2. 代码实现

根据上面逆向运动学求解公式，已知机器人目标速度和机器人参数，可以计算机器人左右轮的目标速度，计算代码如下所示。

```
//运动学逆解析，由机器人目标速度计算电机轮子速度 (m/s)
MOTOR_A.Wheel_TG = RobotV_TG.F_X - RobotV_TG.F_W*R_WHEEL_BASE*0.5;
MOTOR_B.Wheel_TG = RobotV_TG.F_X + RobotV_TG.F_W*R_WHEEL_BASE*0.5;
```

根据上面前向运动学求解公式，已知机器人通过编码器获取的轮子实时转速，可以计算机器人实时的运动速度。

```
//运动学正解析，由机器人轮子速度计算机器人速度，扩大1000倍取整数
RobotV_RT.I_X = ((MOTOR_A.Wheel_RT + MOTOR_B.Wheel_RT)/2)*1000;
RobotV_RT.I_W = ((-MOTOR_A.Wheel_RT + MOTOR_B.Wheel_RT)/R_WHEEL_BASE)*1000;
```

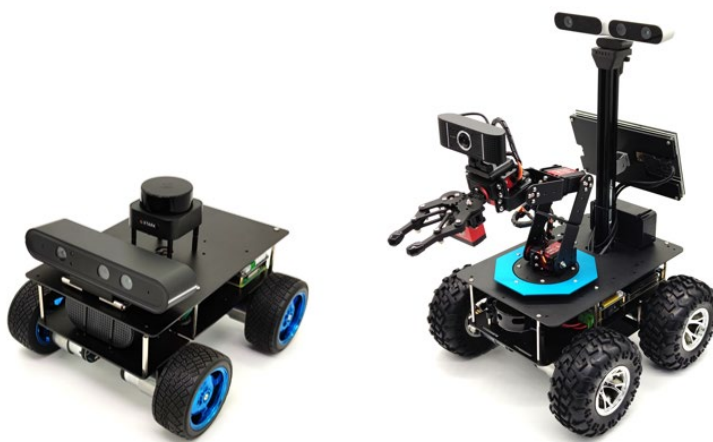
详细代码参考可以参考机器人 STM32 代码部分或者 ROS 底层驱动部分，不同机器人变量名称可能会有变化，计算步骤是一样的，请参考学习。



### 3. 四轮差速模型

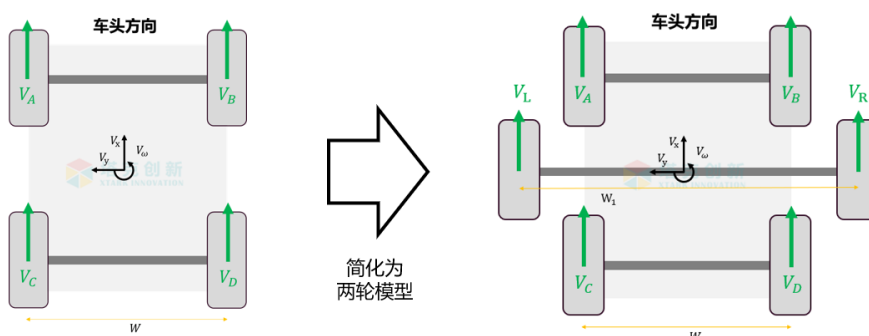
四轮差速模型与两轮差速模型非常相似，并且在载重能力和越野性能上比两轮差速模型更有优势。与两轮差速底盘一样，四轮差速底盘也是靠左右两边轮子的转速差实现转弯，只不过每一边都是两个轮子，同一边的两个轮子的转速是完全一样的，即前轮和后轮的速度是同步的，因此也把四轮差速模型称为同步驱动模型。

四轮差速底盘具有很好的载重和越野性能，但缺点是转弯时轮子会发生一定程度的侧向滑动。在侧向滑动不明显的情况下，可以用两轮差速模型来等效实现。无论如何，这种侧向滑动让底盘控制稳定性、里程计解算、轨迹跟踪等问题变得更加复杂。以下是塔克创新差速机器人。



#### 3.1. 运动学分析

下面我们对四轮差速式机器人进行运动学分析，四轮差速左边两轮子转速一样，右边两轮转速一样，可以简化为两轮差速模型分析。由于四轮差速轮子会发生侧向滑动，简化为两轮时的轮距会与实际四轮轮距不同，我们假设虚拟为两轮差速模型时的轮距为  $W_1$ 。如下分析图所示。



其中参数含义如下。

机器人矢量速度



$V_x$ : 机器人在 O 点的目标前后速度, 前进为正, 单位: m/s。

$V_y$ : 机器人在 O 点的目标左右速度, 左移为正, 单位: m/s。四轮模型忽略。

$V_w$ : 机器人绕 O 点的目标旋转速度, 逆时针为正, 单位: rad/s。

机器人参数

$W$ : 轮距, 机器人左右轮的距离, 单位: m。

机器人轮子速度

$V_A$ : 机器人前左轮速度, 前进方向为正, 单位: m/s。

$V_B$ : 机器人前右轮速度, 前进方向为正, 单位: m/s。

$V_C$ : 机器人后左轮速度, 前进方向为正, 单位: m/s。

$V_D$ : 机器人后右轮速度, 前进方向为正, 单位: m/s。

### 运动学分析

机器人实际轮距为  $W$ , 简化为两轮差速模型后的虚拟轮距为  $W1$ , 我们定义

$$W1 = \gamma * W$$

其中 $\gamma$ 为轮距系数, 是个无量纲参数, 与机器人的总负载、轮胎与地面的相对摩擦系数、转弯半径及质心位置都是有关系, 是一个非常复杂的参数, 所以常用的方法就是做实验, 控制不再改动的机器人在特定地面上做差速转向运动, 采集到多组实验数据后, 拟合估算 $\gamma$ 。塔克四轮差速机器人 $\gamma$ 参数, 可通过源码查看, 不同型号有差异。

下面讨论机器人目标速度  $V_x$ 、 $V_y$ 、 $V_w$ , 轮子的速度  $V_A$ 、 $V_B$ 、 $V_C$ 、 $V_D$ 之间的关系, 当机器人沿着 X 轴前进时:

$$V_A = V_x$$

$$V_B = V_x$$

$$V_C = V_x$$

$$V_D = V_x$$

当机器人绕几何中心自转时:

$$V_A = V_C = V_L = -V_w \left( \frac{W1}{2} \right)$$

$$V_B = V_D = V_R = +V_w \left( \frac{W1}{2} \right)$$

### 逆向运动学求解

根据上述方程求解逆向运动学方程, 已知机器人目标速度  $V_x$ 、 $V_y$ 、 $V_w$ , 求解轮子的速度  $V_A$ 、 $V_B$ 、 $V_C$ 、 $V_D$ 。将以上方程组合并, 即可根据机器人的运动状态解算出四个轮子的转速, 其中  $V_A$ 、 $V_C$  相等,  $V_B$ 、 $V_D$  相等。

$$V_A = V_x - V_w * \left( \frac{W1}{2} \right)$$

$$V_B = V_x + V_w * \left(\frac{W1}{2}\right)$$

$$V_C = V_x - V_w * \left(\frac{W1}{2}\right)$$

$$V_D = V_x + V_w * \left(\frac{W1}{2}\right)$$

### 前向运动学求解

根据逆向运动学方程，求解前向运动学方程。已知驱动轮的当前速度  $V_A$ 、 $V_B$ 、 $V_C$ 、 $V_D$ ，求当前机器人的实时速度  $V_x$ 、 $V_w$ 。

$$V_x = \frac{V_A + V_B + V_C + V_D}{4}$$

$$V_w = \frac{-V_A + V_B - V_C + V_D}{2W1}$$

## 3.2. 代码实现

根据上面逆向运动学求解公式，已知机器人目标速度和机器人参数，可以计算机器人四个轮的目标速度，计算代码如下所示。

```
//运动学逆解析，由机器人目标速度计算机器人轮子速度 (m/s)
MOTOR_A.Wheel_TG = RobotV_TG.F_X - RobotV_TG.F_W*(R_WHEEL_SCALE*R_WHEEL_BASE/2 );
MOTOR_B.Wheel_TG = RobotV_TG.F_X + RobotV_TG.F_W*(R_WHEEL_SCALE*R_WHEEL_BASE/2 );
MOTOR_C.Wheel_TG = RobotV_TG.F_X - RobotV_TG.F_W*(R_WHEEL_SCALE*R_WHEEL_BASE/2 );
MOTOR_D.Wheel_TG = RobotV_TG.F_X + RobotV_TG.F_W*(R_WHEEL_SCALE*R_WHEEL_BASE/2 );
```

根据上面前向运动学求解公式，已知机器人通过编码器获取的轮子实时转速，可以计算机器人实时的运动速度。

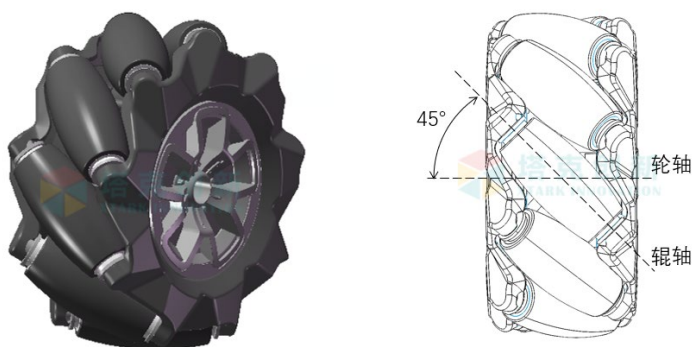
```
//运动学正解析，由机器人轮子速度计算机器人速度
RobotV_RT.I_X = ((MOTOR_A.Wheel_RT + MOTOR_B.Wheel_RT + MOTOR_C.Wheel_RT + MOTOR_D.Wheel_RT)/4)*1000;
RobotV_RT.I_W = ((-MOTOR_A.Wheel_RT + MOTOR_B.Wheel_RT - MOTOR_C.Wheel_RT + MOTOR_D.Wheel_RT)/(2*R_WHEEL_SCALE*R_WHEEL_BASE))*1000;
```

详细代码可以参考机器人 STM32 代码部分或者 ROS 底层驱动部分，不同机器人变量名称可能会有变化，计算步骤是一样的，请参考学习。

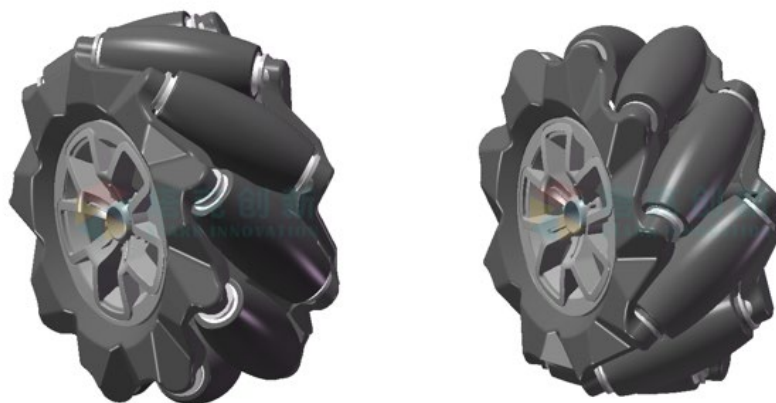
## 4. 麦克纳姆轮全向模型

在竞赛机器人和特殊工种机器人中，全向移动经常是一个必需的功能。全向移动意味着可以在平面内做出任意方向平移同时自转的动作。为了实现全向移动，一般机器人会使用全向轮（Omni Wheel）或麦克纳姆轮（Mecanum Wheel）这两种特殊轮子。

麦克纳姆轮是瑞典麦克纳姆公司的专利。由轮毂和辊子两部分组成，轮毂是整个轮子的主体支架，辊子则是安装在轮毂上的鼓状物，轮毂轴与辊子转轴呈  $45^\circ$ ，如下图所示。



麦克纳姆轮机一般是成对使用的，机器人包含两个左旋轮，两个右旋轮，左旋轮和右旋轮呈手性对称。区别如下图所示。



轮子安装方式如下图，辊子方向呈 X 形状，对角的轮子为相同的轮子。注意如果轮子安装位置错误，会导致不能按照设定的方向移动。

左旋麦轮

右旋麦轮

右旋麦轮

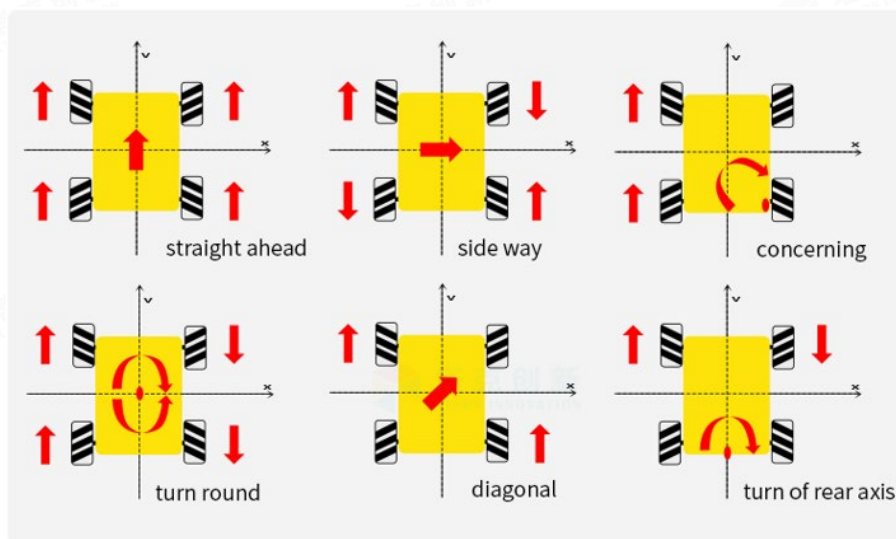
左旋麦轮

由于麦轮的特殊性，无悬挂机器人由于金属板刚性会导致 4 个轮子无法同时着地，或者 4 个轮子受力不均匀，导致轮子打滑，运动精度差，ROS 里程计数据不准确的问题。通过我们摆式悬挂可以很好的解决这个问题。下图是塔克创新两款麦克纳姆轮机器人。

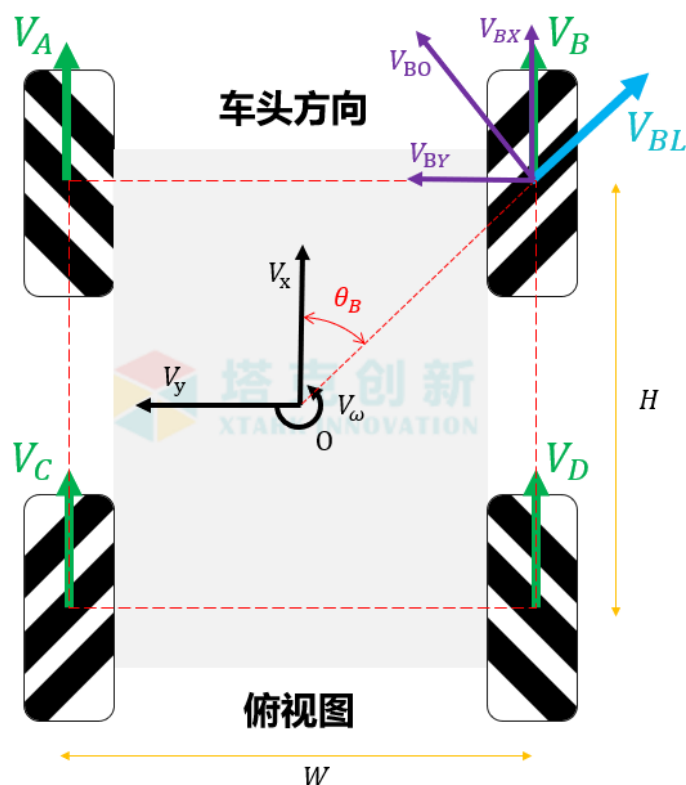


## 4.1. 运动学分析

下面我们对麦轮机器人进行运动学分析，分析前可以先通过下图，感受下机器人速度方向和轮子转动方向的关系，有助于理解下方运动学方程。



麦轮机器人分析图如下所示。



其中参数含义如下。

机器人矢量速度

$V_x$ : 机器人在 O 点的目标前后速度, 前进为正, 单位: m/s。

$V_y$ : 机器人在 O 点的目标左右速度, 左移为正, 单位: m/s。

$V_\omega$ : 机器人绕 O 点的目标旋转速度, 逆时针为正, 单位: rad/s。

机器人参数

$W$ : 轮距, 机器人左右麦轮的距离, 单位: m。

$H$ : 轴距, 机器人前后麦轮的距离, 单位: m。

机器人轮子速度

$V_A$ : 机器人前左轮速度, 前进方向为正, 单位: m/s。

$V_B$ : 机器人前右轮速度, 前进方向为正, 单位: m/s。

$V_C$ : 机器人后左轮速度, 前进方向为正, 单位: m/s。

$V_D$ : 机器人后右轮速度, 前进方向为正, 单位: m/s。

下面是麦轮相关参数, 以 B 轮为例说明, 其它轮子类似。



$V_{BL}$ : 麦轮 B 与地面接触的辊子的线速度, 该速度由麦轮 B 与地面的相对滑动产生, 垂直辊轴向前为正, 单位: m/s。

$V_{BO}$ : 麦轮 B 的质心绕点 O 旋转的线速度, 正方向垂直于麦轮 B 质心与 O 点的连线, 偏逆时针方向。

$V_{BX}$ : 麦轮 B 质心的前后移动速度, 与机器人前后移动速度 $V_x$ 和机器人绕 O 点旋转速度 $V_w$ 相关, 前进为正, 单位: m/s。

$V_{BY}$ : 麦轮 B 质心的左右移动速度, 与机器人左右移动速度 $V_y$ 和机器人绕 O 点旋转速度 $V_w$ 相关, 左移为正, 单位: m/s。

$\theta_B$ : 轮子 B 质心与 O 点的连线与前进方向的夹角, 其值易知为 $\tan^{-1} \frac{W}{H}$ 。

$\beta$ : 轮轴和辊轴之间的夹角, 为  $45^\circ$ 。

### 运动学分析

首先求麦轮 B 的质心前后左右移动速度与机器人整体的矢量速度之间的关系, 假设机器人底盘与四个麦轮质心是一个刚体, 则速度分解可得。

$$V_{BX} = V_x + V_w * \sqrt{\left(\frac{H}{2}\right)^2 + \left(\frac{W}{2}\right)^2} * \sin \theta_B$$

$$V_{BY} = V_y + V_w * \sqrt{\left(\frac{H}{2}\right)^2 + \left(\frac{W}{2}\right)^2} * \cos \theta_B$$

上式说明, 麦轮 B 质心速度与机器人整体速度的关系, 而麦轮 O 质心速度从根本上来说, 还是靠 B 轮线速 $V_B$ 和 B 轮辊子速度 $V_{BL}$ 合并产生的。因为 H,W 为已知参数, 设  $L = \sqrt{\left(\frac{H}{2}\right)^2 + \left(\frac{W}{2}\right)^2}$ , L 物理含义为车轮质心到机器人质心的距离, 则上式变为。

$$V_{BX} = V_x + V_w * L * \sin \theta_B \quad \text{式-1}$$

$$V_{BY} = V_y + V_w * L * \cos \theta_B \quad \text{式-2}$$

B 轮线速 $V_B$ 是由电机转动带动麦轮轮毂产生的。B 轮辊子速度 $V_{BL}$ 是由麦轮 B 与地面的相对滑动产生的,  $V_{BL}$ 速度的大小方向是由麦轮的差速和机械结构决定的, 其动力来源也是电机, 所以当机器人的速度需要 $V_{BL}$ 转动时, 就需要电机产生更大的扭矩, 带动麦轮轮毂的同时带动麦轮辊子。通过速度分解可得。

$$V_{BX} = V_B + V_{BL} * \sin \beta \quad \text{式-3}$$

$$V_{BY} = -V_{BL} * \cos \beta \quad \text{式-4}$$

联立式-1, -2, -3, -4 可得:

$$V_B + V_{BL} * \sin \beta = V_x + V_w * L * \sin \theta_B \quad \text{式-5}$$

$$-V_{BL} * \cos \beta = V_y + V_w * L * \cos \theta_B \quad \text{式-6}$$

式-6 可以变形为:

$$V_{BL} = -\frac{V_y + V_w * L * \cos \theta_B}{\cos \beta} \quad \text{式-7}$$

式-7 代入式-5 可得:

$$\begin{aligned} V_B + \sin \beta \left( -\frac{V_y + V_w * L * \cos \theta_B}{\cos \beta} \right) &= V_x + V_w * \sin \theta_B \\ V_B - V_y * \tan \beta - V_w * L * \cos \theta_B * \tan \beta &= V_x + V_w * \sin \theta_B \\ V_B &= V_x + V_y * \tan \beta + V_w * L * (\sin \theta_B + \cos \theta_B * \tan \beta) \end{aligned}$$

其中 $\beta=45^\circ$ , 简化可得:

$$V_B = V_x + V_y + V_w * L * (\sin \theta_B + \cos \theta_B)$$

又因为

$$\begin{aligned} L * \sin \theta_B &= \frac{W}{2} \\ L * \cos \theta_B &= \frac{H}{2} \end{aligned}$$

则:

$$V_B = V_x + V_y + V_w * L * \left( \frac{H}{2} + \frac{W}{2} \right)$$

类似的推导方式可以求出  $V_A$ 、 $V_C$ 、 $V_D$  与  $V_x$ 、 $V_y$ 、 $V_z$  的关系, 推导时需要注意麦轮与地面接触的辊子的线速度方向, 观察分析图容易知道, 其方向影响的是麦轮 B 质心的左右移动速度。

$$V_{AY} = V_{AL} * \cos \beta$$

$$V_{CY} = -V_{CL} * \cos \beta$$

$$V_{DY} = V_{DL} * \cos \beta$$

### 逆向运动学求解

根据上述方程求解逆向运动学方程, 已知机器人目标速度  $V_x$ 、 $V_y$ 、 $V_w$ , 求解轮子的速度  $V_A$ 、 $V_B$ 、 $V_C$ 、 $V_D$ 。

$$V_A = V_x - V_y - V_w * \left( \frac{H}{2} + \frac{W}{2} \right)$$

$$V_B = V_x + V_y + V_w * \left( \frac{H}{2} + \frac{W}{2} \right)$$

$$V_C = V_x + V_y - V_w * \left( \frac{H}{2} + \frac{W}{2} \right)$$

$$V_D = V_x - V_y + V_w * \left( \frac{H}{2} + \frac{W}{2} \right)$$

### 前向运动学求解

根据逆向运动学方程, 求解前向运动学方程。已知驱动轮的当前速度  $V_A$ 、 $V_B$ 、 $V_C$ 、 $V_D$ , 求当前机器人的实时速度  $V_x$ 、 $V_y$ 、 $V_w$ 。

$$V_x = \frac{V_A + V_B + V_C + V_D}{4}$$



$$V_y = \frac{-V_A + V_B + V_C - V_D}{4}$$

$$V_w = \frac{-V_A + V_B - V_C + V_D}{2H + 2W}$$

## 4.2. 代码实现

根据上面逆向运动学求解公式，已知机器人目标速度和机器人参数，可以计算机器人四个轮的目标速度，计算代码如下所示。

```
//运动学逆解析，由机器人目标速度计算电机轮子速度 (m/s)
MOTOR_A.Wheel_TG = RobotV_TG.F_X - RobotV_TG.F_Y - RobotV_TG.F_W*(R_WHEEL_BASE/2+R_ACLE_BASE/2);
MOTOR_B.Wheel_TG = RobotV_TG.F_X + RobotV_TG.F_Y + RobotV_TG.F_W*(R_WHEEL_BASE/2+R_ACLE_BASE/2);
MOTOR_C.Wheel_TG = RobotV_TG.F_X + RobotV_TG.F_Y - RobotV_TG.F_W*(R_WHEEL_BASE/2+R_ACLE_BASE/2);
MOTOR_D.Wheel_TG = RobotV_TG.F_X - RobotV_TG.F_Y + RobotV_TG.F_W*(R_WHEEL_BASE/2+R_ACLE_BASE/2);
```

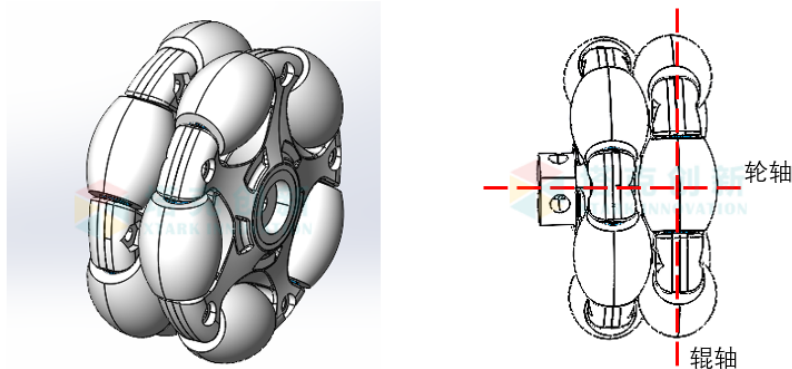
根据上面正向运动学求解公式，已知机器人通过编码器获取的轮子实时转速，可以计算机器人实时的运动速度。

```
//运动学正解析，由机器人轮子速度计算机器人速度
RobotV_RT.I_X = ((MOTOR_A.Wheel_RT + MOTOR_B.Wheel_RT + MOTOR_C.Wheel_RT + MOTOR_D.Wheel_RT)/4)*1000;
RobotV_RT.I_Y = ((-MOTOR_A.Wheel_RT + MOTOR_B.Wheel_RT + MOTOR_C.Wheel_RT - MOTOR_D.Wheel_RT)/4)*1000;
RobotV_RT.I_W = ((-MOTOR_A.Wheel_RT + MOTOR_B.Wheel_RT - MOTOR_C.Wheel_RT + MOTOR_D.Wheel_RT)/4)/(R_WHEEL_BASE/2+R_ACLE_BASE/2)*1000;
```

详细代码可以参考机器人 STM32 代码部分或者 ROS 底层驱动部分，不同机器人变量名称可能会有变化，计算步骤是一样的，请参考学习。

## 5. 三轮全向模型

全向轮 (Omni Wheel) 又叫福马轮, 由轮毂和固定在外周的许多小辊子构成, 轮轴和辊轴之间的夹角为  $90^\circ$ , 如下图所示。



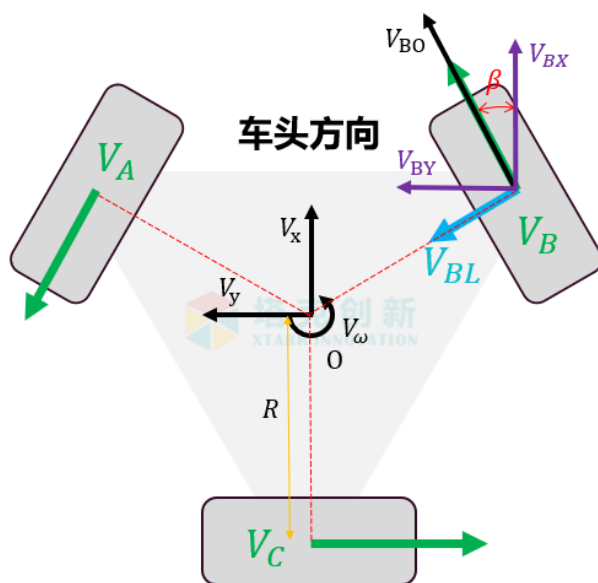
每个轮子具有三个自由度, 分别是绕轮轴转动, 沿垂直于与地面接触的辊子的辊轴方向移动, 绕轮子和地面的接触点转动。

根据机械原理, 机构的原动件数应该等于机构的自由度, 因此, 若要实现三个自由度的控制, 则应该有三个独立的输入, 即 3 个全向轮。下图是塔克创新的三轮全向机器人。



### 5.1. 运动学分析

下面我们对三轮全向机器人进行运动学分析, 分析图如下所示。



其中参数含义如下。

机器人矢量速度

$V_x$ : 机器人在 O 点的目标前后速度, 前进为正, 单位: m/s。

$V_y$ : 机器人在 O 点的目标左右速度, 左移为正, 单位: m/s。

$V_w$ : 机器人绕 O 点的目标旋转速度, 逆时针为正, 单位: rad/s。

机器人参数

R: 旋转半径, 全向轮到机器人中心 O 点的距离, 单位: m。

机器人轮子速度

$V_A$ : 机器人 A 轮线速度, 绕 O 点逆时针为正, 单位: m/s。

$V_B$ : 机器人 B 轮线速度, 绕 O 点逆时针为正, 单位: m/s。

$V_C$ : 机器人 C 轮线速度, 绕 O 点逆时针为正, 单位: m/s。

运动学分析就是机器人矢量速度  $V_x$ ,  $V_y$ ,  $V_w$  与轮子  $V_A$ ,  $V_B$ ,  $V_C$  线速度之前的关系。下面是全向轮相关参数, 以 B 轮为例说明, 其它轮子类似。

$V_{BL}$ : 全向轮 B 与地面接触的辊子的线速度, 该速度由全向轮 B 与地面的相对滑动产生, 垂直辊轴向机器人质心 O 为正, 单位: m/s。

$V_{BO}$ : 全向轮 B 的质心绕点 O 旋转的线速度, 正方向垂直于全向轮 B 质心与 O 点的连线, 偏逆时针方向。

$V_{BX}$ : 全向轮 B 质心的前后移动速度, 与机器人前后移动速度  $V_x$  和机器人绕 O 点旋转速度  $V_w$  相关, 前进为正, 单位: m/s。

$V_{BY}$ : 全向轮 B 质心的左右移动速度, 与机器人左右移动速度 $V_y$ 和机器人绕 O 点旋转速度 $V_w$ 相关, 左移为正, 单位: m/s。

$\beta$ : 全向轮 B 前进方向与机器人前进方向的夹角, 因为全向轮为等边三角形安装, 所以其值为 $30^\circ$ 。

### 运动学分析

参数讲解完毕, 我们现在讲解这些参数之间的关系, 以及如何求出三轮全向轮机器人的运动学正逆解公式。

首先求全向轮 B 的质心前后左右移动速度与机器人整体的矢量速度之间的关系, 假设机器人底盘与三个全向轮质心是一个刚体, 则速度分解可得。

$$V_{BX} = V_x + V_w * R * \cos\beta \quad \text{式-1}$$

$$V_{BY} = V_y + V_w * R * \sin\beta \quad \text{式-2}$$

上式说明, 全向轮 B 质心速度与机器人整体速度的关系, 而全向轮 B 质心速度从根本上来讲, 还是靠 B 轮线速 $V_B$ 和 B 轮辋子速度 $V_{BL}$ 合并产生的。

B 轮线速 $V_B$ 是由电机转动带动全向轮轮毂产生的。B 轮辋子速度 $V_{BL}$ 是由全向轮 B 与地面的相对滑动产生的,  $V_{BL}$ 速度的大小方向是由全向轮的差速和机械结构决定的, 其动力来源也是电机, 所以当机器人的速度需要 $V_{BL}$ 转动时, 就需要电机产生更大的扭矩, 带动全向轮的同时带动全向轮辋子。通过速度分解可得。

$$V_{BX} = -V_{BL} * \sin\beta + V_B * \cos\beta \quad \text{式-3}$$

$$V_{BY} = V_{BL} * \cos\beta + V_B * \sin\beta \quad \text{式-4}$$

联立以上四个式子可得:

$$-V_{BL} * \sin\beta + V_B * \cos\beta = V_x + V_w * R * \cos\beta \quad \text{式-5}$$

$$V_{BL} * \cos\beta + V_B * \sin\beta = V_y + V_w * R * \sin\beta \quad \text{式-6}$$

上述式-6 变形为:

$$V_{BL} = \frac{V_y + V_w * R * \sin\beta - V_B * \sin\beta}{\cos\beta} \quad \text{式-7}$$

式-7 代入公式-5 可得:

$$-\tan\beta(V_y + V_w * R * \sin\beta - V_B * \sin\beta) + V_B * \cos\beta = V_x + V_w * R * \cos\beta$$

其中  $\beta=30^\circ$ , 则:

$$-\frac{\sqrt{3}}{3}V_y - \frac{\sqrt{3}}{6}V_w * R + \frac{\sqrt{3}}{6}V_B + \frac{\sqrt{3}}{2}V_B = V_x + \frac{\sqrt{3}}{2}V_w * R$$

$$\frac{2\sqrt{3}}{3}V_B = V_x + \frac{\sqrt{3}}{3}V_y + \frac{2\sqrt{3}}{3}V_w * R$$

$$V_B = \frac{\sqrt{3}}{2}V_x + \frac{1}{2}V_y + V_w * R$$

类似的推导方式可以求出  $V_A$ 、 $V_C$  与  $V_x$ 、 $V_y$ 、 $V_w$  的关系。

### 逆向运动学求解

根据上述方程求解逆向运动学方程，已知机器人目标速度  $V_x$ 、 $V_y$ 、 $V_w$ ，求解轮子的速度  $V_A$ 、 $V_B$ 、 $V_C$ 。

$$\begin{aligned} V_A &= -\frac{\sqrt{3}}{2}V_x + \frac{1}{2}V_y + V_w * R \\ V_B &= \frac{\sqrt{3}}{2}V_x + \frac{1}{2}V_y + V_w * R \\ V_C &= -V_y + V_w * R \end{aligned}$$

### 前向运动学求解

根据逆向运动学方程，求解前向运动学方程。已知驱动轮的当前速度  $V_A$ 、 $V_B$ 、 $V_C$ ，求当前机器人的实时速度  $V_x$ 、 $V_y$ 、 $V_w$ 。

$$\begin{aligned} V_x &= \frac{-V_A + V_B}{\sqrt{3}} \\ V_y &= \frac{V_A + V_B - 2V_C}{3} \\ V_w &= \frac{V_A + V_B + V_C}{3R} \end{aligned}$$

## 5.2. 代码实现

根据上面逆向运动学求解公式，已知机器人目标速度和机器人参数，可以计算机器人四个轮的目标速度，计算代码如下所示。

```
//运动学逆解析，由机器人目标速度计算电机轮子速度 (m/s)
MOTOR_A.Wheel_TG = -0.5*SQRT3*RobotV_TG.F_X + 0.5*RobotV_TG.F_Y + R_RADIUS*RobotV_TG.F_W;
MOTOR_B.Wheel_TG = 0.5*SQRT3*RobotV_TG.F_X + 0.5*RobotV_TG.F_Y + R_RADIUS*RobotV_TG.F_W;
MOTOR_C.Wheel_TG = -RobotV_TG.F_Y + R_RADIUS*RobotV_TG.F_W;
```

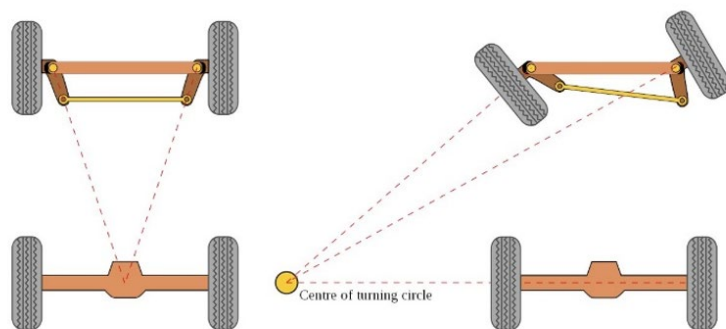
根据上面前向运动学求解公式，已知机器人通过编码器获取的轮子实时转速，可以计算机器人实时的运动速度。

```
//运动学正解析，由机器人轮子速度计算机器人速度
RobotV_RT.I_X = ((-MOTOR_A.Wheel_RT + MOTOR_B.Wheel_RT)/SQRT3)*1000;
RobotV_RT.I_Y = ((MOTOR_A.Wheel_RT + MOTOR_B.Wheel_RT - 2.0*MOTOR_C.Wheel_RT)/3)*1000;
RobotV_RT.I_W = ((MOTOR_A.Wheel_RT + MOTOR_B.Wheel_RT + MOTOR_C.Wheel_RT)/(3*R_RADIUS))*1000;
```

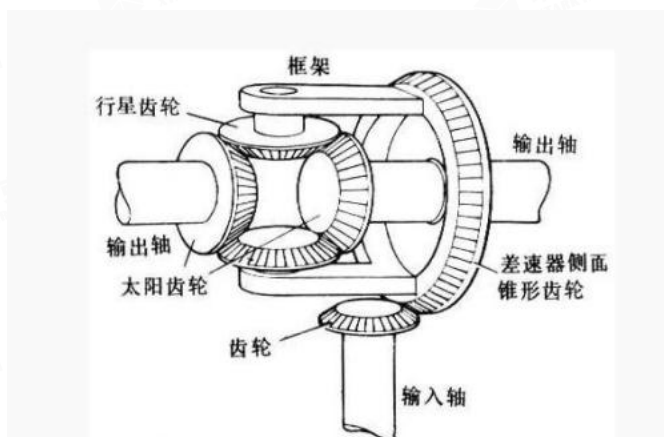
详细代码可以参考机器人 STM32 代码部分或者 ROS 底层驱动部分，不同机器人变量名称可能会有变化，计算步骤是一样的，请参考学习。

## 6. 阿克曼转向模型

阿克曼转向是一种现代汽车的转向方式，在汽车转弯的时候，内外轮转过的角度不一样，内侧轮胎转弯半径小于外侧轮胎。这种转向方式最初是由的德国马车工程师的 Georg Lankensperger 1817 年提出，他的代理商 Rudolph Ackerman 于 1818 年在英国申请专利，所以从今往后这个转向原理就叫阿克曼转向几何了。

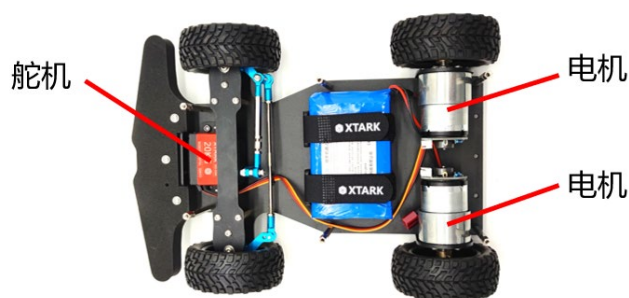


我们知道，汽车是通过方向盘控制车的转向，通过刹车油门控制车的行驶速度。后面两个轮为主动轮，前面为从动轮。后轴上的左右轮共用一个发动机驱动。为了能让汽车在转弯，或曲线行驶时，左右轮的速度能相互协调，这时就需要加入差速器用以调整左右轮的转速差。



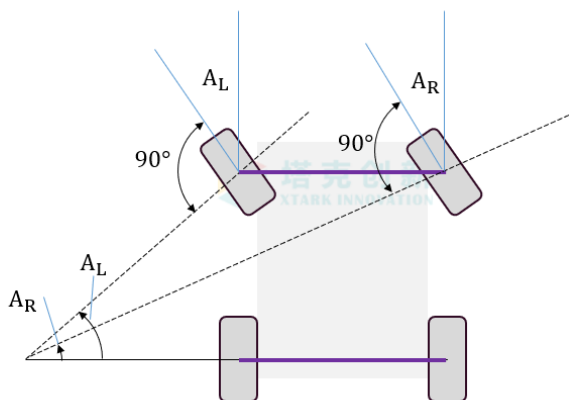
考虑到差速器复杂的齿轮结构设计，机器人底盘通常采用两个独立的电机驱动左右轮，并配合电机驱动板调速程序实现电子差速，控制前轮转向的阿克曼结构由舵机带动来实现转向，如下图所示。因此下面的前向运动学、逆向运动学和里程计相关内容，将基于这种简易阿克曼后驱底盘展开讲解。



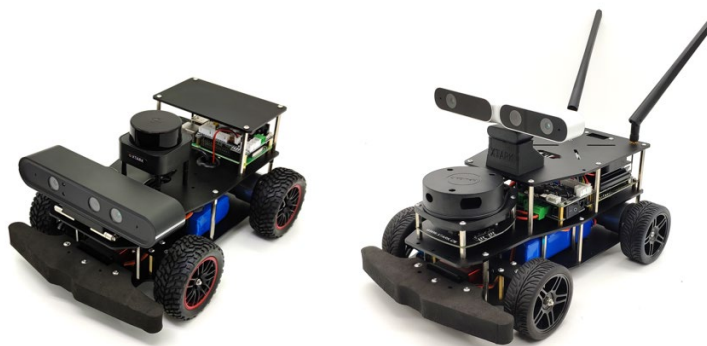


四轮差速底盘具有很好的载重和越野性能，但缺点是转弯时轮子会发生一定程度的侧向滑动，这种侧向滑动让底盘控制稳定性、里程计解算、轨迹跟踪等问题变得更加复杂。阿克曼底盘通过前轮的机械转向，让底盘中的 4 个轮子在基本不发生侧向滑动的情况下能顺畅地转弯。不过也有缺点，阿克曼底盘不能原地旋转，也就是最小转弯半径不为 0，想象一下汽车在狭小的地方倒车入库的场景就不难理解了，车辆需要反复调整方向盘才能将车停入指定车位。

如下图所示，为阿克曼式机器人转向时的状态，要求 4 个轮子运动方向的垂线相交于一点，即 4 个轮子围绕同一个圆心进行旋转。



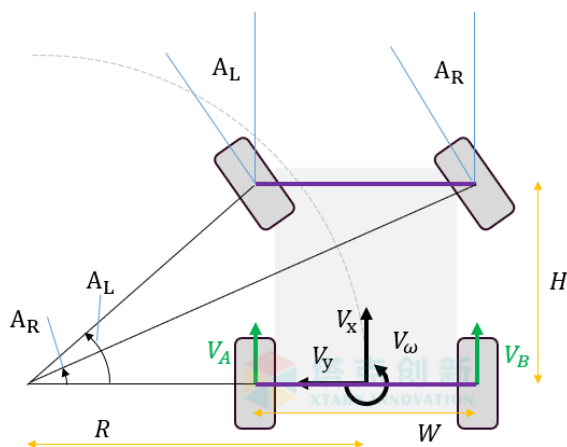
下图是塔克创新的阿克曼机器人。





## 6.1. 运动学分析

下面我们对阿克曼转向机器人进行运动学分析，分析图如下所示。



其中参数含义如下。

机器人矢量速度

$V_x$ : 机器人在 O 点的目标前后速度，前进为正，单位：m/s。

$V_y$ : 机器人在 O 点的目标左右速度，左移为正，单位：m/s。此处阿克曼忽略。

$V_\omega$ : 机器人绕 O 点的目标旋转速度，逆时针为正，单位：rad/s。

机器人参数

W: 轮距，机器人左右轮的距离，单位：m。

H: 轴距，机器人前后轮的距离，单位：m。

机器人轮子速度

$V_A$ : 机器人左轮速度，前进方向为正，单位：m/s。

$V_B$ : 机器人右轮速度，前进方向为正，单位：m/s。

机器人前轮转向偏角

$A_L$ : 机器人左前轮转向偏角，左偏方向为正，单位：rad/s。

$A_R$ : 机器人右前轮转向偏角，左偏方向为正，单位：rad/s。

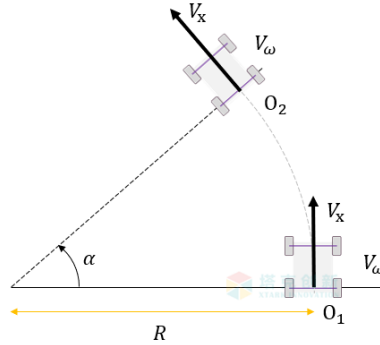
转弯半径

R: 机器人转弯半径，单位：m。

### 运动学分析

阿克曼机器人由于转向方式的不同，无法实现零转弯半径转向，具有最小转弯半径。运动学求解前，必须先确定  $V_x$ 、 $V_\omega$  与转弯半径 R 的关系。

如下图所示，假设机器人以速度  $V_x$ 、 $V_w$  运动了  $t$  秒，则圆弧  $O_1O_2$  的长度为速度  $V_x$  对时间  $t$  的积分： $O_1O_2 = V_t * t$ ；机器人姿态旋转了  $\alpha$ ，为速度  $V_w$  对时间  $t$  的积分： $\theta = V_w * t$ 。



路径为圆形，可得转弯半径  $R$ ：

$$R = \frac{O_1O_2}{\theta} = \frac{V_x * t}{V_w * t} = \frac{V_x}{V_w} \quad \text{式-1}$$

现在已知  $V_x$ 、 $V_w$  与转弯半径  $R$ ，接下来求左右后轮的速度  $V_A$ 、 $V_B$  和左右前轮偏角  $A_L$ 、 $A_R$ 。

已知转向时机器人整体在绕  $O$  点做旋转运动，则容易求得右后轮的速度  $A_L$ 、 $A_R$ ：

$$V_A = V_x \frac{R - 0.5W}{R}$$

$$V_B = V_x \frac{R + 0.5W}{R}$$

由几何关系可得左右轮偏角  $A_L$ 、 $A_R$ ：

$$A_L = \tan^{-1}\left(\frac{H}{R - 0.5W}\right)$$

$$A_R = \tan^{-1}\left(\frac{H}{R + 0.5W}\right)$$

阿克曼逆向运动学公式已经求出，接下来求前向运动学：

$$V_A + V_B = V_x \frac{R - 0.5W}{R} + V_x \frac{R + 0.5W}{R}$$

$$V_A + V_B = V_x \frac{R + R + 0.5W - 0.5W}{R}$$

$$V_A + V_B = 2V_x$$

由上式得：

$$V_x = \frac{V_A + V_B}{2}$$

再求  $V_w$ ：

$$V_B - V_A = V_x \frac{R + 0.5W}{R} - V_x \frac{R - 0.5W}{R}$$

$$V_B - V_A = V_x \frac{R - R + 0.5W + 0.5W}{R}$$

$$R = V_x \frac{W}{V_B - V_A}$$

将式  $R = \frac{V_x}{V_w}$  带入得:

$$\frac{V_x}{V_w} = V_x \frac{W}{V_R - V_L}$$

由上式得:

$$V_w = \frac{V_R - V_L}{W}$$

### 逆向运动学求解

已知机器人目标速度  $V_x$ 、 $V_w$ ，求解后轮速度  $V_A$ 、 $V_B$ ，前轮偏角  $A_L$ 、 $A_R$  的关系。

$$R = \frac{V_x}{V_w}$$

$$V_A = V_x \frac{R - 0.5W}{R}$$

$$V_B = V_x \frac{R + 0.5W}{R}$$

$$A_L = \tan^{-1}\left(\frac{H}{R - 0.5W}\right)$$

$$A_R = \tan^{-1}\left(\frac{H}{R + 0.5W}\right)$$

### 前向运动学求解

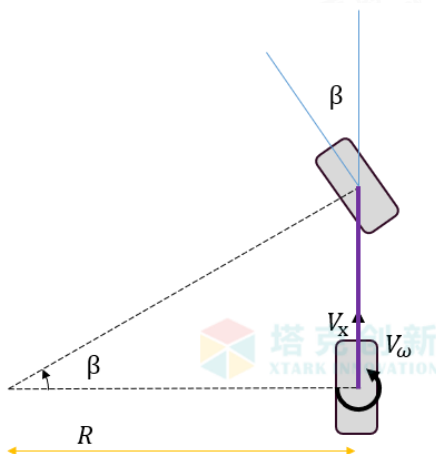
阿克曼由于采用后轮电子差速，前向运动学与两轮差速运动学相同。已知驱动轮的当前速度  $V_A$ 、 $V_B$ ，求当前机器人的实时速度  $V_x$ 、 $V_w$ 。

$$V_x = \frac{V_A + V_B}{2}$$

$$V_w = \frac{V_A - V_B}{W}$$

## 6.2. 简化模型

由于前轮转角装置的原因，前面左右两轮的在同一方向臂控制下，往往形成的转角不同。为了方便分析舵机转接，将阿克曼四轮机器人的模型进一步简化为如下图所示的两轮单车模型。



逆向运动学模型，求解左右轮转角改为求解机器人整体转角 $\beta$ 进一步简化为

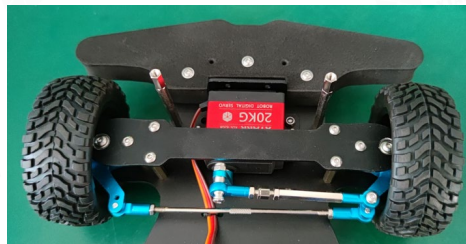
$$\beta = \tan^{-1}\left(\frac{H}{R}\right)$$

虽然求出了转向角 $\beta$ ，但是 $\beta$ 不是最终的控制量。我们还需要将转向角 $\beta$ 映射成舵机的输出舵角，才能最终实现底盘的转向。而转向角 $\beta$ 与舵角之间的映射关系，与舵机及转向悬臂具体安装位置有关系，通常都是通过实验标定法获取到转向角 $\beta$ 与舵角之间的映射表 $\beta$ -Angle。利用映射表 $\beta$ -Angle，就能快速从给定的转向角 $\beta$ 换算出舵机旋转角度，实现最终的转向控制。

### 6.3. 舵机角度

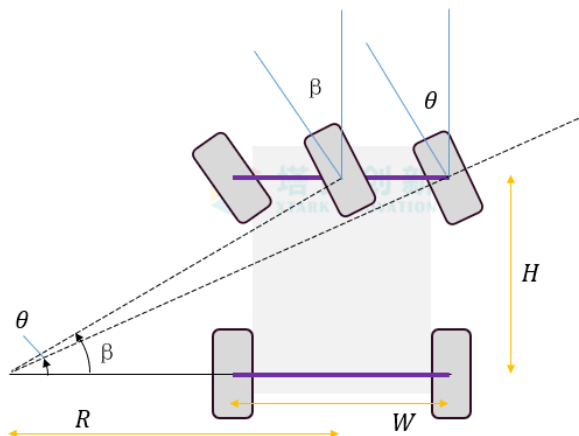
虽然求出了转向角 $\beta$ ，但是 $\beta$ 不是最终的控制量。我们还需要将转向角 $\beta$ 映射成舵机的输出舵角，才能最终实现底盘的转向。而转向角 $\beta$ 与舵角之间的映射关系，与舵机及转向悬臂具体安装位置有关系，通常都是通过实验标定法获取到转向角 $\beta$ 与舵角之间的映射表。利用映射表，就能快速从给定的转向角 $\beta$ 换算出舵机，实现最终的转向控制。

下图所示为基于曲柄摇杆机构的阿克曼转向结构，通过驱动舵机带动舵臂，舵臂再通过机械传动控制两个前轮进行转向。两个前轮偏角大小的差异则由机械设计决定。我们知道使用舵机控制舵臂时，舵臂偏角与右前轮的关系，可以等价为舵机 PWM 控制值与右前轮的关系，因为 PWM 与舵臂偏角是一一对应的关系的。



这个曲柄摇杆的结构分析起来比较复杂，而舵臂偏角与前轮偏角是一一对应的，可以采用曲线拟合模型来近似的获得舵臂偏角与前轮偏角之间的关系公式。

由于 $\beta$ 不能直接测量，首先我们计算出机器人转向角 $\beta$ 与右前轮转向角 $\theta$ 的关系，再通过测量右前轮与舵机输出之间的关系，拟合曲线。



上图根据几何关系，可以计算得到：

$$\theta = \tan^{-1}\left(\frac{H}{R + \frac{W}{2}}\right)$$

进一步计算得：

$$\theta = \tan^{-1}\left(\frac{H}{\frac{H}{\tan\beta} + \frac{W}{2}}\right)$$

测量左前轮与舵机的转角对应关系，最小二乘拟合得到曲线

$$\text{舵机角度} = -0.0041\theta^2 + 1.205\theta$$

控制 PWM 信号，即可控制舵机角度

## 6.4. 代码实现

根据上面运动学求解公式，已知机器人目标速度和机器人参数，首先计算后轮目标转速和机器人前轮转向角度，代码实现如下所示。

```

//判断机器人转向速度是否为0
if(RobotV_TG.I_W != 0)
{
    //判断机器人前进速度是否为0
    if( RobotV_TG.I_X != 0)
    {
        //计算转弯半径
        RobotStr.Radius = RobotV_TG.F_X/RobotV_TG.F_W;

        //阿克曼机器人需要设置最小转弯半径
        //如果目标速度要求的转弯半径小于最小转弯半径,
        //会导致机器人运动摩擦力大大提高,严重影响控制效果

        //转弯半径小于最小转弯
        if(RobotStr.Radius>0 && RobotStr.Radius<R_TURN_R_MINI)
        {
            RobotStr.Radius = R_TURN_R_MINI;
        }

        else if(RobotStr.Radius<0 && RobotStr.Radius>(-R_TURN_R_MINI))
        {
            RobotStr.Radius = -R_TURN_R_MINI;
        }

        //计算机器人前轮转向角度,单位弧度
        RobotStr.Angle = atan(R_ACLE_BASE/(RobotStr.Radius));

        //运动学逆解析,由机器人目标速度计算电机轮子速度 (m/s)
        MOTOR_A.Wheel_TG = RobotV_TG.F_X*(RobotStr.Radius-0.5*R_WHEEL_BASE)/RobotStr.Radius;
        MOTOR_B.Wheel_TG = RobotV_TG.F_X*(RobotStr.Radius+0.5*R_WHEEL_BASE)/RobotStr.Radius;
    }
    else
    {
        MOTOR_A.Wheel_TG = 0;
        MOTOR_B.Wheel_TG = 0;
        RobotStr.Radius = 0;
        RobotStr.Angle = 0;
    }
}
else
{
    MOTOR_A.Wheel_TG = RobotV_TG.F_X;
    MOTOR_B.Wheel_TG = RobotV_TG.F_X;
    RobotStr.Radius = 0;
    RobotStr.Angle = 0;
}
}

```

然后再根据前轮转向角，计算最终舵机的 PWM 输出信号。

```

//根据前轮角度计算右前轮角度
if(RobotStr.Angle !=0 )
{
    RobotStr.RAngle = (atan(R_ACLE_BASE/((R_ACLE_BASE/tan(RobotStr.Angle*0.01745))+0.5*R_WHEEL_BASE)))*(180.0/PI);
}
else
{
    RobotStr.RAngle = 0;
}

//根据右前轮角度，计算舵机转向角度
RobotStr.SAngle = -(0.0041*RobotStr.RAngle*RobotStr.RAngle + 1.2053*RobotStr.RAngle)*180/PI;

//根据舵机转向角度，计算舵机PWM控制量
servo_angle = (RobotStr.SAngle*10 + 900 + ax_servo_offset);

```

根据上面前向运动学求解公式，已知机器人通过编码器获取的轮子实时转速，可以计算机器人实时的运动速度。

//运动学正解析，由机器人轮子速度计算机器人速度

```
RobotV_RT.I_X = ((MOTOR_A.Wheel_RT + MOTOR_B.Wheel_RT)/2)*1000;
```

```
RobotV_RT.I_Y = 0;
```

```
RobotV_RT.I_W = ((-MOTOR_A.Wheel_RT + MOTOR_B.Wheel_RT)/R_WHEEL_BASE)*1000;
```

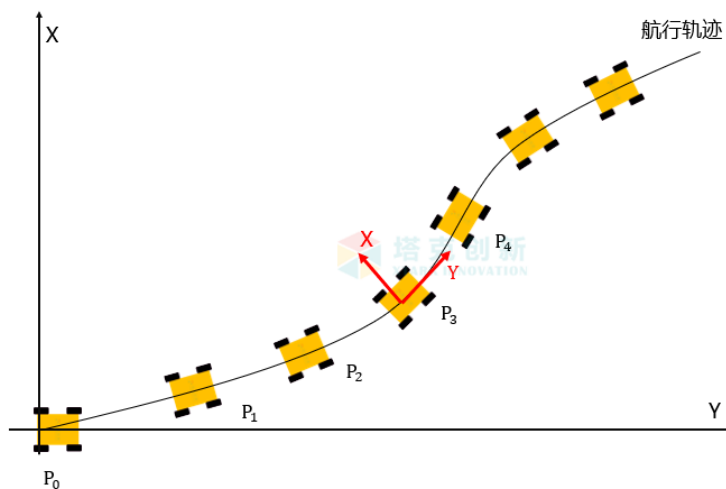
详细代码可以参考机器人 STM32 代码部分或者 ROS 底层驱动部分，不同机器人变量名称可能会有变化，计算步骤是一样的，请参考学习。



## 7. 里程计解算

机器人主机向底盘发送运动控制量  $V_x$ 、 $V_y$ 、 $V_w$ ，经过逆向运动学解算，得到底盘中每个轮子应该运行的目标速度，通过电机控制主板的 PID 算法可以实现每个轮子转速控制。

底盘在运动过程中，轮式编码器可以反馈每个轮子的转速，经过前向运动学解算，得到底盘的整体实时速度  $V_x$ 、 $V_y$ 、 $V_w$ 。基于  $V_x$ 、 $V_y$ 、 $V_w$  速度，就可以由前一时刻底盘的位姿，推算出当前时刻底盘的位姿，这个推算过程也叫“航迹推演算法”，底盘的位姿代表的是底盘速度瞬心在世界坐标系的位姿。



利用轮式编码器反馈，经过航迹推演算法，得到底盘当前时刻位姿，整个过程就是机器人技术中常说的轮式里程计。

很多初学者常常将轮式里程计与编码器、电机控制主板、底盘等概念混为一谈，所以需要特别注意。可以看出，轮式里程计其实是编码器、底盘运动学模型、航迹推演算法等综合作用的产物。也就是说，轮式里程计并不是某种传感器，而是利用轮式编码器计算底盘位姿的一种算法。

多款机器人模型里程计解算相似，下面分析轮式里程计的解算过程，已知机器人实时的矢量速度定义如下。

$V_x$ ：机器人在 O 点的目标前后速度，前进为正，单位：m/s。

$V_y$ ：机器人在 O 点的目标左右速度，左移为正，单位：m/s。

$V_w$ ：机器人绕 O 点的目标旋转速度，逆时针为正，单位：rad/s。

一般在底盘上电时刻，在底盘的位姿处建立里程计坐标系 (odom)，随时间推移，底盘的实时位姿连接起来就形成了底盘的航行轨迹。同时，底盘自身的坐标系建立在其速度瞬心处。

只要建立起机器人坐标系中的速度量  $V_x$ 、 $V_y$ 、 $V_w$  与里程计坐标系中的底盘位姿  $P_k$  之间的关系，就可以解算出轮式里程计。在二维平面中移动的底盘，可以用  $P=[x, y, w]^T$  表示底盘的坐标和航向角，那么位姿  $P$  满足微分方程。

$$\dot{P} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} \cos w & -\sin w & 0 \\ \sin w & \cos w & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} V_x \\ V_y \\ V_w \end{bmatrix}$$

将在时间  $t$  上进行积分，就可以求出底盘实时的位姿，也就是里程计信息。实际情况下，需要在离散时间域上进行计算，由于底盘相邻两位姿相隔时间  $\Delta t$  很小，积分运算可以用下式的累加运算来替代。

$$P_k = P_{k-1} + \dot{P}_{k-1} \Delta t$$

代码实现如下所示，其中 DATA\_PERIOD 表示循环计算时间，一般取 10~50ms，具体可参考机器人源代码。

```
//计算里程计数据
pos_data.pos_x += (vel_data.linear_x*cos(pos_data.angular_z) - vel_data.linear_y*sin(pos_data.angular_z)) * DATA_PERIOD;
pos_data.pos_y += (vel_data.linear_x*sin(pos_data.angular_z) + vel_data.linear_y*cos(pos_data.angular_z)) * DATA_PERIOD;
pos_data.angular_z += vel_data.angular_z * DATA_PERIOD; //绕Z轴的角度位移，单位: rad
```



塔克机器人 淘宝店



塔克创新 微信公众号

了解更多机器人  
[www.xtark.cn](http://www.xtark.cn)