## 两轮差速底盘

### 1.1、底盘说明

两轮差速底盘由两个动力轮位于底盘左右两侧，两轮独立控制速度，通过给定不同速度实现底盘转向控制。一般会配有一到两个辅助支撑的万向轮。

**主要底盘有：**

1. Turtlebot；
2. 扫地机器人；
3. 无人仓AGV小车；
4. 轮椅；

**底盘主要特点：**

1. 控制简单、里程计计算简单。
2. 只能给定X方向速度、Z轴方向角速度。

### 1.2、底盘运动学分析

1.2.1、底盘模型



图1.2.1 常见的两轮差速底盘模型

1.2.2、机器人本身约束方程（物理特性）



图1.2.2 a 左侧是车的两个轮子，右侧带箭头的圆圈是运动的方向和运动的圆心



图1.2.2 b 两轮底盘运动解析图

如图1.2.2所示：车体速度为V, 左轮速度为VL ,右轮速度为VR , 车体自传速度为,转弯半径为R，两轮之间距离为D，两轮到车中心的距离为d，右轮到圆心距离为L。

**约束方程（高中知识）：**

①、 、角速度、速度和运动半径之间的物理关系。

②、 左轮速度分解

③、 右轮速度分解

④、 和速度V等于左右轮速度之和的一半。

⑤、由②、③得  得  整车角速度和两轮速度之间的关系

**1.3、运动控制、控制指令分解**

控制指令分解指如何通过控制左右两个轮的独立速度使整个机器人的整体运动既满足前向速度等于V，转动的角速度等于



图1.2.3 运动控制的输入、输出参数说明

如图1.2.3所示，运动控制器输入参数为整车速度Vx和角速度（因为轮子不能横着走所以Vy一直为零），输出参数是左右两轮速度VL 、VR。转弯半径R则由左右轮速度决定。

由约束方程②、③得：

左轮速度为：（d为两轮之间距离的一半）

右轮速度为：

**1.4、运动轨迹、里程计计算**



图1.2.4 上电时刻机器人坐标系和世界坐标系重合

里程计（odom）计算是指以机器人上电时刻为世界坐标系的起点**O（0,0）**(航向为世界坐标系的X轴指向)累积计算任意时刻机器人相对于世界坐标系的位置及航向。

机器人的位置Pose.Xw、Pose.Yw的值可以看成是车体运动方向极小时间内位置增量分解到X、Y方向的积分量。



图1.2.5 机器人在世界坐标系的位置

**此时分两种方式来推算轨迹：**速度推算方式、编码器推算方式

**I、速度推算轨迹：（速度积分累积误差较大，最终精度在10%左右）**

在机器人坐标系下，单位时间∆t（一个控制周期：∆t = ti+1 - ti，通常为10ms、20ms）以速度V移动的距离为∆d = ∆t\*V。将此距离分别分解到世界坐标系的X、Y轴：

∆xw = ∆d \* cos(θ)= ∆t \* V \* cos(θ)

∆yw = ∆d \* sin(θ)= ∆t \* V \* sin(θ)

同时单位时间角度变化为∆θ = \*∆t。

以此方式不断累积，即可实现任意时间的位置解算。

Xw = Xw + ∆xw = Xw + ∆t \* V \* cos(θ)

Yw = Yw + ∆yw = Yw + ∆t \* V \* sin(θ)

θ = θ + \*∆t

**II、编码器推算轨迹：（直接对距离做分解累积误差相对较小，最终精度1%以内，如果做修正，同时航向角较准确的情况下精度可以达到0.1%以内（已经实测））**

编码器每一个脉冲对应实际轮子行走的直线距离系数为：

rate\_encoder = 2𝛑r/sum\_encoders

其中r为车轮半径，sum\_encoders为轮子走动一圈的编码器的脉冲总数。

单位时间∆t内编码的增量为：inc\_encoder = encoder\_now – encoder\_last,(当前编码器值减去上次编码器值)

则单位时间机器人移动的距离为∆d = inc\_encoder \* rate\_encoder

世界坐标系下x、y方向累计里程分别为：

Xw = Xw + ∆xw = Xw + ∆d \* cos(θ)

Yw = Yw + ∆yw = Yw + ∆d \* sin(θ)

**其中航向角θ的获取分两种情况：**

1. **底盘带具有稳定航向角的IMU，此方式可以使得最终解算的位置十分准确**（电子罗盘受电机干扰交大不可用）

θ直接等于IMU的航向角Yaw。（IMU的Yaw上电为0，刚好和车的航向角一致）

1. **依靠底盘两个轮子上精确的编码器推算航向角。**

两轮编码器单位时间内增量分别为 inc\_encoder\_r、inc\_encoder\_l

单位时间内两轮扭动的距离差为 lenth\_error = (inc\_encoder\_r- inc\_encoder\_l) \* rate\_encoder;

由⑤得：由距离差得单位时间内角度差为 anlge\_z\_error = lenth\_error / 2d;

根据编码器累计的角度θ为 anlge\_z += anlge\_z\_error。

此方式非常依赖编码器精度，有累计误差，效果不如直接使用IMU的好。