

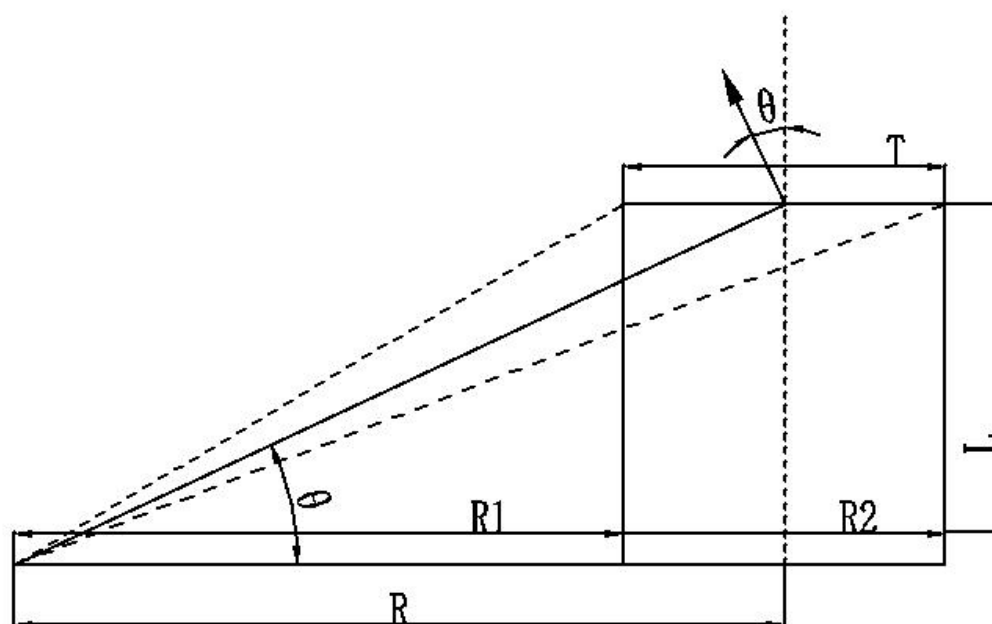
后轮双电机差速小车控制原理

汽车在转弯时，后轮的内轮和外轮行驶距离不同，行驶的时间却一样，因此，后轮之间存在差速的问题。很容易想到，转弯的时候内侧的轮胎要比外侧的轮胎转速慢。传统汽车使用机械差速器完成差速，机械差速器的基本运动规律是：无论转弯或者直行，两侧驱动车轮的转速之和始终等于差速器壳转速的 2 倍。比如差速器壳转速为 V ，内侧后轮转速为 $V-a$ ，则外侧后轮转速为 $V+a$ ；

最近几年以特斯拉为代表的电动汽车企业迅速崛起，在电动汽车中，由于使用两个电机直接驱动后轮，不需要机械差速，使得机械设计更加简单。这样在电控方面就更加复杂一点。我们车模也是类似的原理。

1. 运动学分析

设舵机控制转角为 θ ，车前进速度为 v ，前后轮中心距为 L ，后轮轮距为 T ，后转弯内侧轮速度为 v_1 后 转弯外侧轮速度为 v_2 ；



因为角速度的一致性，根据运动关系分析得知：

$$\begin{aligned}\frac{v}{R} &= \frac{v1}{R1} = \frac{v2}{R2} \\ \tan \theta &= \frac{L}{R} \\ R1 &= R - \frac{1}{2}T \\ R2 &= R + \frac{1}{2}T\end{aligned}$$

由此可得：

$$\begin{aligned}V1 &= \frac{V}{R} * R1 = V * \left(1 - \frac{T \tan \theta}{2L}\right) \\ V2 &= \frac{V}{R} * R2 = V * \left(1 + \frac{T \tan \theta}{2L}\right)\end{aligned}$$

2. C 语言实现

车模上面有 2 个带编码器的电机和一个舵机，我们需要通过 C 语言写出上面的运动关系，然后对电机和舵机进行控制。代码如下：

```
void Kinematic_Analysis(float velocity, float angle)
{
    Target_A = velocity * (1 + T * tan(angle) / 2 / L);
    Target_B = velocity * (1 - T * tan(angle) / 2 / L);    //后轮差速
    Servo = SERVO_INIT + angle * K;                      //舵机转向
}
```

以上语句在已知前轮转向角度和目标速度的前提下，求两个后轮电机的速度大小。velocity 和 angle 分别是由用户输入的速度和前轮转角。K 是小车整体的系数，和舵机的安装、PWM 的初始化都有关系，需要实际测量，主要作用是完成舵机 PWM 控制引脚的脉冲宽度与前轮转向角度之间的转换。具体测试方法：设置 Servo = SERVO_INIT + a 控制舵机输出一个固定的脉宽，然后测量前轮的转角为 α ，容易得到该系数为：

$$K = \frac{a}{\alpha}$$

舵机的控制是比较简单的，因为内部集成了控制电路，把修改后的参数赋值给 STM32 的相关寄存器即可。但是直流电机的控制要麻烦的多，我们上面的运动学分析得到的是电机的目标速度，我们需要把这个目标值送入 PID 控制器进行速

度闭环控制，使得电机的实际输出速度趋近于目标值。

```
/******
```

函数功能：增量 PI 控制器

入口参数：编码器测量值，目标速度

返回 值：电机 PWM

根据增量式离散 PID 公式

$$pwm += Kp[e(k) - e(k-1)] + Ki * e(k) + Kd[e(k) - 2e(k-1) + e(k-2)]$$

$e(k)$ 代表本次偏差

$e(k-1)$ 代表上一次的偏差 以此类推

pwm 代表增量输出

在我们的速度控制闭环系统里面，只使用 PI 控制

$$pwm += Kp[e(k) - e(k-1)] + Ki * e(k)$$

```
*****/
```

```
int Incremental_PI_A (int Encoder,int Target)
```

```
{
```

```
    static int Bias,Pwm,Last_bias;
```

```
    Bias=Target-Encoder;           //计算偏差
```

```
    Pwm+=Velocity_KP*(Bias-Last_bias)+Velocity_KI*Bias;//增量式 PI 控制器
```

```
    Last_bias=Bias;                //保存上一次偏差
```

```
    return Pwm;                    //增量输出
```

```
}
```

具体调用过程如下，通过速度闭环控制计算电机 A 最终 PWM:

```
Motor_A=Incremental_PI_A(Encoder_Left,Target_A);
```

以 100hz 的频率执行相关的代码即可完成速度闭环控制。