

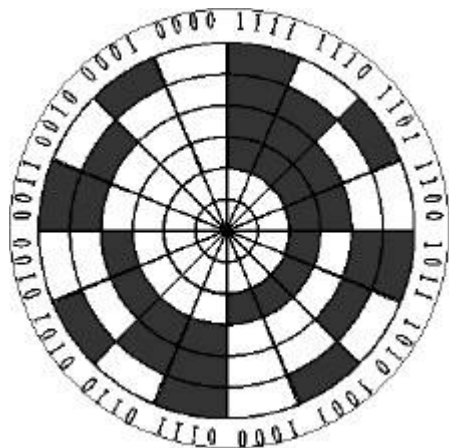
- 1.编码器基础
 - 1.1绝对式编码器
 - 1.2增量式编码器
 - 1.3 4倍频
 - 1.4 测试代码
- 2.PID控制
 - 2.1PID控制的原理
 - 比例（P）控制
 - 积分（I）控制
 - 微分（D）控制
 - 2.2实际模型
 - 2.3具体实现

1.编码器基础

对机器人实现位置和速度的控制需要使用传感器获取机器人运动的信息，编码器是常用的方式。常见的编码器有增量式编码器和绝对式编码器。

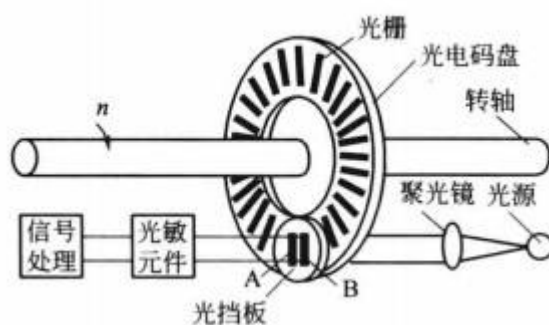
1.1绝对式编码器

绝对式编码器通过对码盘上的各个位置设计特定的编码，可以输出转动轴的绝对位置信息。



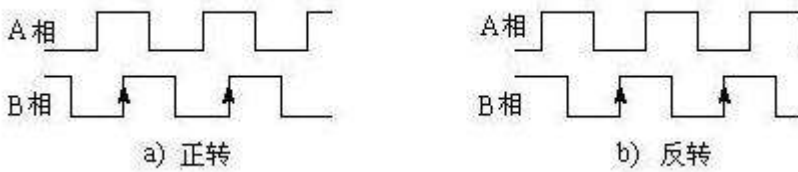
1.2增量式编码器

增量式编码器无法直接得到轴的绝对位置信息。对于轴的每一圈转动，增量式编码器提供一定数量的脉冲（通常编码器参数中用 n 线/ n 脉冲来表示）。通过测量脉冲的数量可以得到旋转的角度，或者测量单位时间内的脉冲



数可以得到轴的转速。

但这里涉及到方向的问题，单纯从一相的脉冲输出无法判断轴的旋转方向，所以通常增量式编码器通过输出AB两相脉冲实现方向的测量。AB两相脉冲之间相差90°。

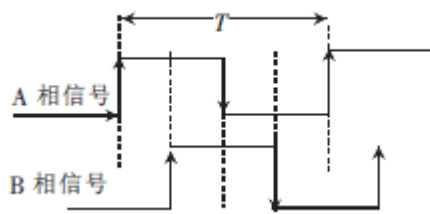


例如上图中的增量式编码器脉冲，当编码器正转时，B相脉冲的上升沿对应A相高电平，而当编码器反转时，B相上升沿对应A向低电平。如此判断旋转方向。

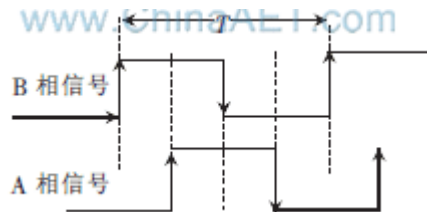
旋转编码器很重要的参数是其每圈输出的脉冲数，以400线的编码器为例，每转动一圈，AB相各输出400个脉冲。通过4倍频可以实现更高的测量精度。

1.3 4倍频

所谓4倍频就是对AB相的上升沿和下降沿均作检测，这样在一个周期内有四种状态，如下：



旋转方向为正向时，A相上升沿对应B相低电平 -> B相上升沿对应A相高电平 -> A相下降沿对应B相高电平 -> B



相下降沿对应A相低电平

旋转方向为反向时，B相上升沿对应A相低电平 -> A相上升沿对应B相高电平 -> B相下降沿对应A相高电平 -> A相下降沿对应B相低电平

1.4 测试代码

具体代码实现可以通过Arduino中的中断判断CHANGE很容易实现4倍频。

```
#define Phase_A 2 //定义A相引脚
#define Phase_B 3 //定义B相引脚

unsigned long time;
long Position = 0;

boolean A = false;
boolean B = false;
```

```
void setup()
{
  pinMode(Phase_A, INPUT_PULLUP); //内部上拉, 防止信号干扰
  pinMode(Phase_B, INPUT_PULLUP);

  attachInterrupt(0, Interrupt_A, CHANGE); //检测上升沿、下降沿
  attachInterrupt(1, Interrupt_B, CHANGE); //检测上升沿、下降沿

  Serial.begin(115200); //初始化Arduino串口
}

void loop(){
  time = millis();
  if(time%30 == 0){
    Serial.println(Position,DEC);
  }
}

void Interrupt_A(){
  if(A == false){
    if( B == false){Position = Position + 1;A == true;}
    else{Position = Position - 1;A == true;}
  }
  else{
    if( B == false){Position = Position - 1;A == false;}
    else{Position = Position + 1;A == false;}
  }
}

void Interrupt_B(){
  if(B == false){
    if( A == false){Position = Position - 1;B == true;}
    else{Position = Position + 1;A == true;}
  }
  else{
    if( A == false){Position = Position + 1;B == false;}
    else{Position = Position - 1;B == false;}
  }
}
```

2.PID控制

2.1PID控制的原理

在工程实际中，PID控制是应用最为广泛控制方式。PID为比例、积分、微分的缩写。当被控制的系统的结构和参数不能完全掌握，或得不到精确的数学模型时，控制理论的其它技术难以采用时，系统控制器的结构和参数必须依靠经验和现场调试来确定，这时应用PID控制技术最为方便。即当我们不完全了解一个系统和被控对象，或不能通过有效的测量手段来获得系统参数时，最适合用PID控制技术。PID控制，实际中也有PI和PD控制。PID控制器就是根据系统的误差，利用比例、积分、微分计算出控制量来对系统进行控制。

比例（P）控制

比例控制是一种最简单的控制方式。其控制器的输出与输入误差信号成比例关系。当仅有比例控制时系统输出存在**稳态误差**（Steady-state error）。

积分（I）控制

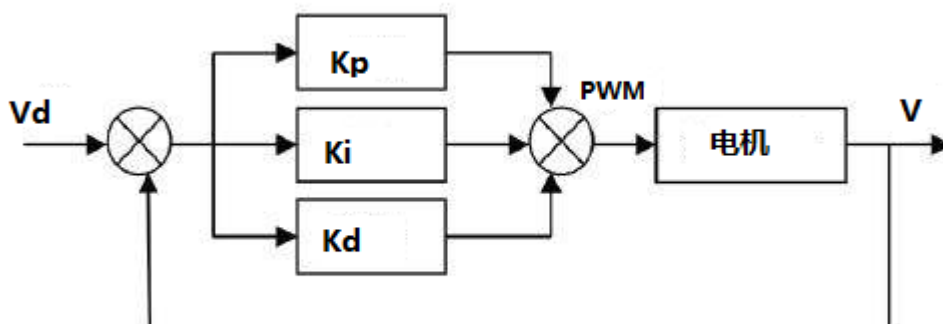
在积分控制中，控制器的输出与输入误差信号的积分成正比关系。对于一个自动控制系统，如果在进入稳态后存在稳态误差，则称这个控制系统是有稳态误差的或简称有差系统（System with Steady-state Error）。为了消除稳态误差，在控制器中必须引入“积分项”。积分项对误差取决于时间的积分，随着时间的增加，积分项会增大。这样，即便误差很小，积分项也会随着时间的增加而加大，它推动控制器的输出增大，使稳态误差进一步减小，直到等于零。因此，比例+积分(PI)控制器，可以使系统在进入稳态后无稳态误差。

微分（D）控制

在微分控制中，控制器的输出与输入误差信号的微分（即误差的变化率）成正比关系。自动控制系统在克服误差的调节过程中可能会出现振荡甚至失稳。其原因是由于存在有较大惯性组件（环节）或有滞后(delay)组件，具有抑制误差的作用，其变化总是落后于误差的变化。解决的办法是使抑制误差的作用的变化“超前”，即在误差接近零时，抑制误差的作用就应该是零。这就是说，在控制器中仅引入“比例”项往往是不够的，比例项的作用仅是放大误差的幅值，而目前需要增加的是“微分项”，它能预测误差变化的趋势，这样，具有比例+微分的控制器，就能够提前使抑制误差的控制作用等于零，甚至为负值，从而避免了被控量的严重超调。所以对有较大惯性或滞后的被控对象，比例+微分(PD)控制器能改善系统在调节过程中的动态特性。

2.2实际模型

对于3WD全向小车，在对整体运动进行解算后得到各个轮子的期望转速 V_d ，以这个转速作为PID控制的输入，输出为控制电机的PWM值，反馈则为通过编码器反馈值计算出的实际速度 V 。闭环系统如下：



在实际操作中，由于是通程序进行PID的计算，无法达到连续的积分和微分计算，所以用离散处理。给定PID控制频率，周期性地对PID控制计算。

主要参数：

- pid_interval 控制周期 ms
- kp 比例系数
- ki 积分系数

- kd 微分系数
- ko 统一放大缩小比例

2.3具体实现

见下PIBOT Firmware代码分析