取物抛射机器人设计及制作（七）

控制程序设计

完成机器人硬件系统搭建后即可开始进行控制程序设计，类似于赋予机器人灵魂，让机器人按工作要求实现运动与完成任务动作。

1．系统简述

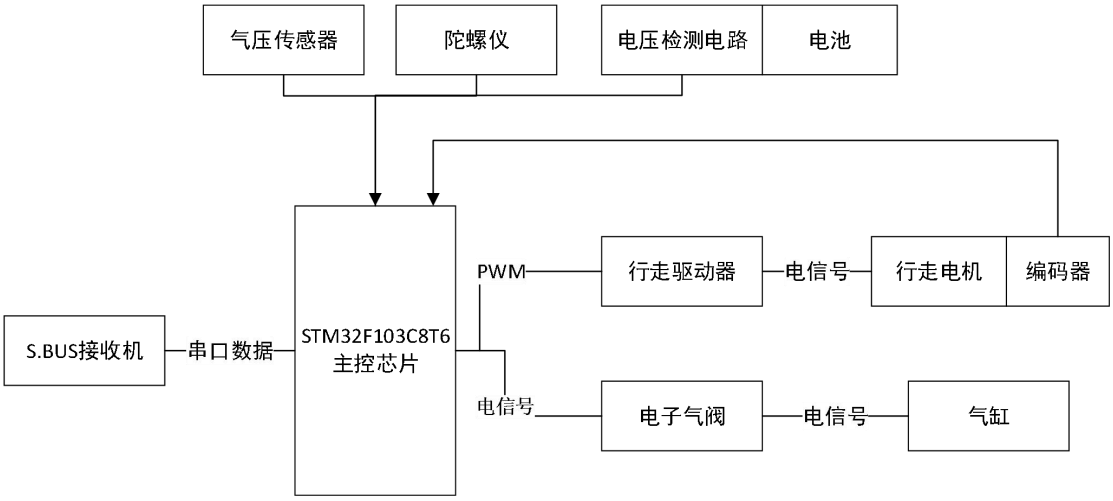

图 1 总体控制结构图

如上图所示，机器人控制系统主要分为四个系统，动作系统，传感器系统，信号接收系统，以及控制处理系统。各个系统紧密相连，动作系统由行走系统和行走手臂组成，用以完成机器人的行走和物体的拾取；传感器系统通过多个传感器收集机体运行数据，让机体的动作变得流畅和精准；信号接收系统是负责控制信号的收取，并将数据传输到主控芯片。机器人采用 STM32F103C8T6 为处理芯片，高达 72MHZ 的时钟频率，能高效而稳定处理所有的数据，完美实现对机体的一切控制。

2.控制程序实现

（1）工具

本机器人基于 c 语言的开发，使用了 STM studio、串口助手等辅助软件。

（2）平台

所有的控制程序都基于 stm32f103c8t6。

（3）算法

本程序使用到了多个算法。最典型的是 PID 控制算法。本算法用于对电机的精确控制。

2．软件流程图

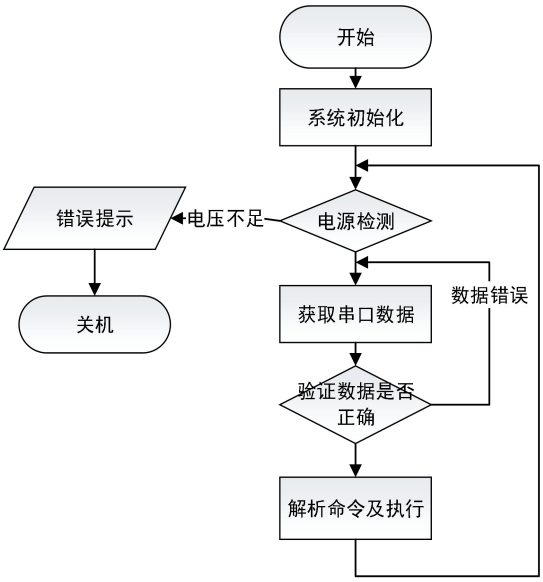

图 2 控制流程图

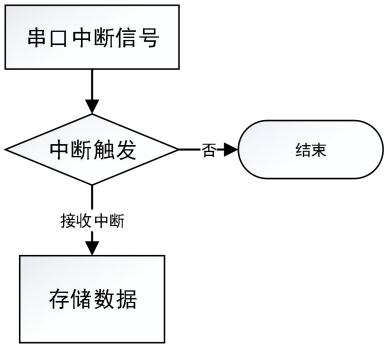

图 3 串口数据接收

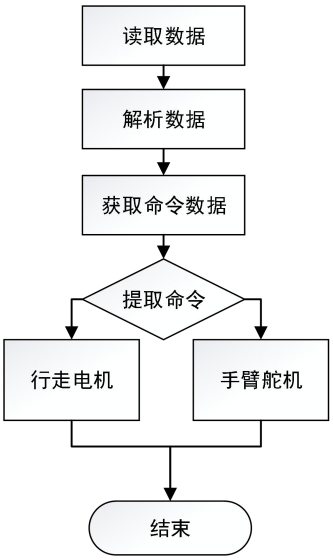

图 4　命令执行流程

4.使用算法

使用 PID 算法本质上是为了让编码器的值（即速度）稳定在一个值。PID 调参变成了如何让编码器的值（速度）保持在一个固定值。

PID 算法是工业应用中最广泛算法之一，在闭环系统的控制中，可自动对控制系统进行准确且迅速的校正。PID 算法已经有 100 多年历史，在四轴飞行器，平衡小车、汽车定速巡航、温度控制器等场景均有应用。

比例 P：e(k) - e(k-1) 此次误差-上次误差

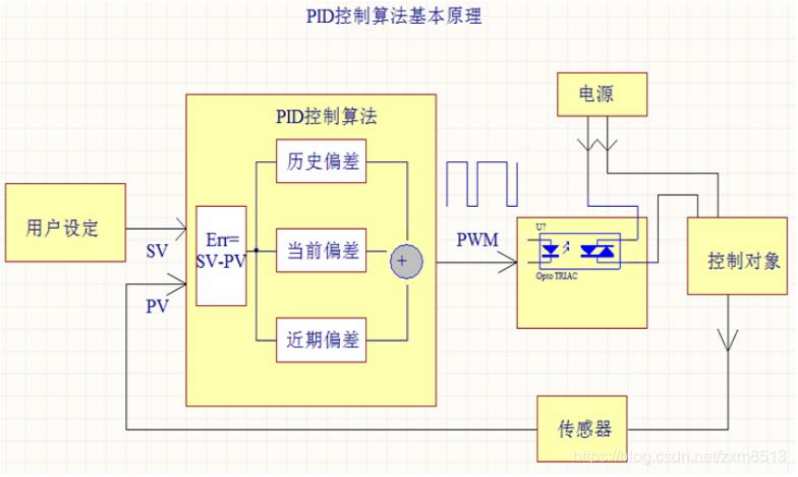

图 5 PID 算法原理图

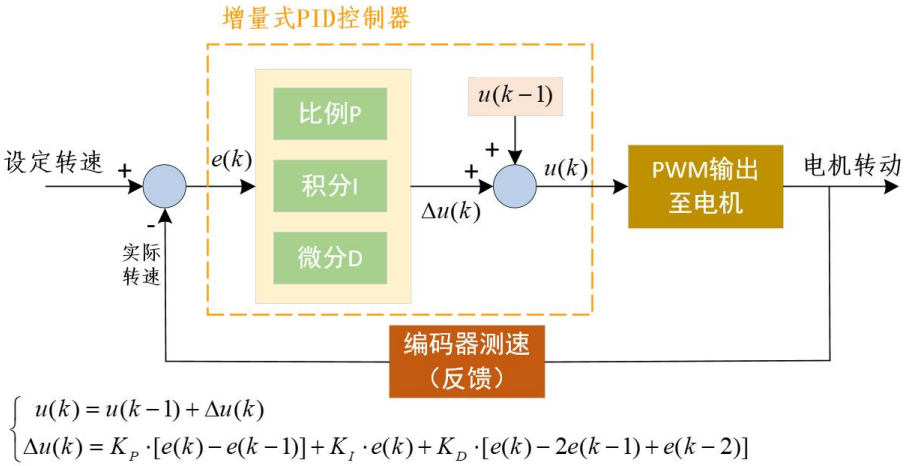

图 6 PID 算法控制框图

积分 I：e(k) 此次误差 d

微分 D：e(k) - 2e(k-1)+e(k-2) 这次误差-2×上次误差+上上次误差

增量式 PID 首先计算的是 Δu(k)，然后与上次的输出相加，才是此次的输出结果。增量式 PID 没有误差累加，控制增量 Δu(k)的确定仅与最近 3 次的采样值有关。

比例项：用于粗调节和快速调节，是必不可少的，是缩短和目标值之间距离的主要部分，着眼于缩小当下的差距，但会有稳态误差

积分项：用于消除稳态误差，借助先前的经验即把之前的一段时间内的各个偏差累加起来，来消除当下的稳态误差

微分项：用于预判以加快响应，同样是借助先前的经验，但是是

通过求先前一段时间内的偏差变化率来预测当下或者下一段时间内偏差的变化趋势，从而早做修正。

```
//计时器计时
void TIM3_IRQHandler (void)
{
  if(TIM_GetITStatus(TIM3,TIM_IT_Update)==SET)
  {
    circle=Read_circle_count();//圈数
    scale= Read_scale();//走过的刻度值
    speed=circle;

      //增量式
      cnt=speed;
      speed_now=cnt;//当前实际速度
      err_now=target-speed_now;//误差速度=期望值-实际值
      jisuan=KP*(err_now-err_last)+KI*err_now+KD*(err_now+err_last-2*err_last);
      out+=jisuan;
      if(out>19000)
      out=19000;
      PWM_SetCompare1(out);
      err_last_last=err_last;
      err_last=err_now;

      printf("%d",speed_now);
  }
    TIM_ClearITPendingBit(TIM3,TIM_IT_Update);
}
```

图 7 算法实现代码事宜

完成控制程序编写后，各工作原件的运动状态达到了稳定期望值，达到了预期设计效果。