数字电阻 PID 控制及其实现

杜佳颖

July 29, 2024

1 数字电阻 PID 部分

在本项目中，我们使用 PID 控制算法来调节电机的转速。PID 控制器的设置需要考虑多个方面，包括参数设置、采样处理和防抖等。

1.1 PID 参数设置

我们首先使用一般的 PID 控制算法，通过不断调整 P、I、D 参数，得到了一个较优的 PID 参数组合。但是，即使在最优参数下，电机的转速仍然无法跟上手动操纵杆的速度。因此，我们在控制逻辑上进行了优化：当电机位置接近设定目标时，使用 PID 控制；在其他情况下，使用 PWM 最大值 255 进行控制。这样的分段控制方法可以确保电机在大幅度调整时快速响应，同时在接近目标时精确调整。

1.2 采样处理

为了确保 PID 控制的精度，我们在代码中引入了采样处理。通过读取传感器的值，计算出当前电机位置与设定值之间的误差，并根据误差调整控制信号。每次更新控制信号时，我们都会记录当前时间，并在下一次计算时使用时间差来调整积分和微分项，确保控制信号的稳定性和准确性。

1.3 防抖处理

在实际应用中，传感器读数可能会因为环境噪声或机械振动而产生抖动。为了避免这些干扰影响控制效果，我们在计算误差的过程中加入了防抖处理。具体做法是，当误差的变化量小于一定阈值时，不更新微分项，从而减少因抖动引起的控制信号波动。

以下是主要代码段的概括：- 初始化 PID 参数和各类引脚。- 在主循环中，不断读取传感器和设定值，计算误差并根据误差大小选择 PID 控制或 PWM 最大值控制。- 通过串口监视器进行实时参数调整，方便调试和优化。

2 Python GUI 部分

为了方便用户调整 PID 参数和监控电机状态，我们设计了一个基于 Python 的 GUI 界面。通过串口通讯，GUI 界面可以实时发送参数调整命令，并接收电机

的运行状态。

2.1 界面设计

GUI 界面使用 Tkinter 库实现，包含以下功能模块：- 串口选择和连接按钮。- 用于调整 Kp、Ki、Kd 参数的滑动条和设置按钮。- 设定值输入框和设置按钮。- 显示当前电机位置和设定值的标签。- LED 状态显示。

2.2 串口通讯

GUI 通过串口与电机控制器进行通讯。用户在 GUI 上调整参数时，相应命令会通过串口发送到控制器。控制器实时反馈电机位置和状态，GUI 界面相应更新显示。为了确保通讯的可靠性，我们在程序中加入了多线程处理，保证主界面不会因为串口读写操作而卡顿。

3 发挥部分 EXE 方案

由于时间和资源限制，目前我们还没有开发独立的 EXE 应用程序。未来计划将 Python GUI 部分打包为独立的 EXE 文件，方便用户在无需安装 Python 环境的情况下直接使用。