

Arduino 离散 PID 控制 simulink 仿真程序

谢业平

(机械工程 3 班 201810501055)

摘要：在 PC 中编写软件，运行一仿真程序，模拟一系统（例如温度控制系统，电机控制系统等），要求包含该系统的模型，以及控制接口。该控制接口能够接收来自串口的控制指令。在Arduino 中编写控制程序，实现离散PID 控制。要求该程序包括 PID 控制算法以及控制接口实现，该控制接口能够控制 PC 里的模型程序。分别运行上述实验系统，在 PC 端记录控制系统的状态曲线，绘制该曲线并机进行说明。本文采用matlab 软件进行仿真，同时通过串口连接arduino，实现arduino 中编程，进行离散PID 控制。

关键字：仿真；arduino；离散；控制

1、总体方案设计

为了实现仿真程序，采用MATLAB 软件中的simulink 进行仿真设计，并且可以与 arduino 串口相连，实现系统的电路仿真，同时设计对应的MATLAB 程序。之后对设计的arduino 程序，实现电路的离散PID 控制。最后对设计的产品进行调试，解决存在的问题。

1.1 实物设置

Arduino 无法提供电机本身所需的电流强度，因此我们将使用晶体管组成的电源电路。微控制器将向晶体管栅极发送 PWM 信号，因此电机将接收相同的 PWM 但功率信号。Arduino 和电动机等部件连接的实物图如下：

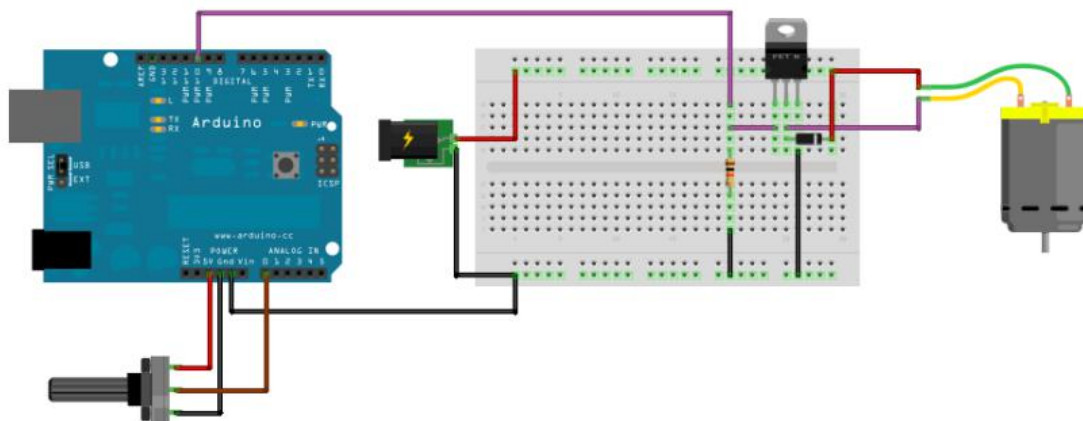


图1arudino 连接实物图

1.2 arduinoI/O 接口设置

首先，Arduino 代码必须能够读取电位器的电压，同时可以通过串口发送数据，也能够接收电机所需的电源数据，并且将数据作为 PWM 信号写入模拟输出。因为转换器返回 0 到 1024 之间的值，因此我们将其映射到 0 和 255，以便将其作为单个无符号位发送。

实现 arduino 的代码如下：

```
int out = 0; byte in = 0; byte pinOut = 10;
void setup()
{
    Serial.begin(9600); // 初始化串口
    pinMode(pinOut, OUTPUT); // 准备输出
}
void loop() {
    out = analogRead(A0); // 从引脚 A0 读取
    out = map(out, 0, 1023, 0, 255); // 缩放以获得 uint8 格式
    Serial.write(out); // 以 ASCII 格式发送基数为 10
    if(Serial.available()) // 如果有数据则从系统中读取
    {
        in = Serial.read();
        analogWrite(pinOut, in); // 写在引脚 10 上
    }
    delay(20); // 等待稳定转换器
```

1.3 simulink 控制循环

另一方面，Simulink 程序必须从中接收这些数据，通过串口接口，将所得的数据传输到指定的部件，同时通过 PID 控制器，确定所需要的控制信号，也就是发动机的发动功率。然后通过串口发送它，格式为 uint8，无符号整数 8 位，取 0 到 255 之间的值，使用 analogWrite()Arduino 函数。

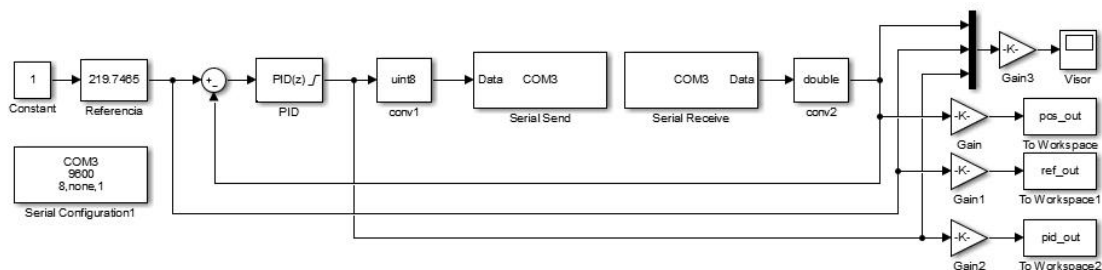


图2仿真电路示意图

在 PID 块内，可以编辑参数 P，I 和 D，这是我们使用 $P=0.26$ ， $I=0.9$ ， $D=0.04$ 和 10ms 的离散化。由于串行传输是以 uint8 格式完成的，但操作是以双格式完成的，因此需要用到转换器。K 增益的值为 5/1024。

1.4 matlab 对收集的数据进行处理

如果我们查看之前的 Simulink 程序，可以看出，不仅三个信号在查看器中呈现，而且它们也存储在 Matlab 工作区中。这允许我们事后处理监测过程的结果。

在运行仿真电路后，运行 matlab 程序，matlab 程序如下：

```
%% 提前各种参数定义
f = figure('Name','Captura');
axis([0 length(ref_out) 0 5.1])
grid on
xlabel('测量 (-)')
ylabel('电压 (V)')
title('使用 arduino 实时捕获电压')
hold all

%% 对参数赋值
pos_out = pos_out(:);
pid_out = pid_out(:);
x = linspace(0,length(ref_out),length(ref_out));

%%绘制图表
cla
plot(x,pid_out,'Color',[0.6,0.6,0.6],'LineWidth',2)
plot(x,ref_out,x,pos_out,'LineWidth',2)
legend('控制','参考','位置','地点','最优解');

%%保存图表
savefigure('resultado','s',[4.5 3],'po','-dpdf')
```

运行 matlab 程序得到如下图所示的结果，通过对比发现 pid 控制使得曲线更加趋向与设定的值，平稳性也有所增加。

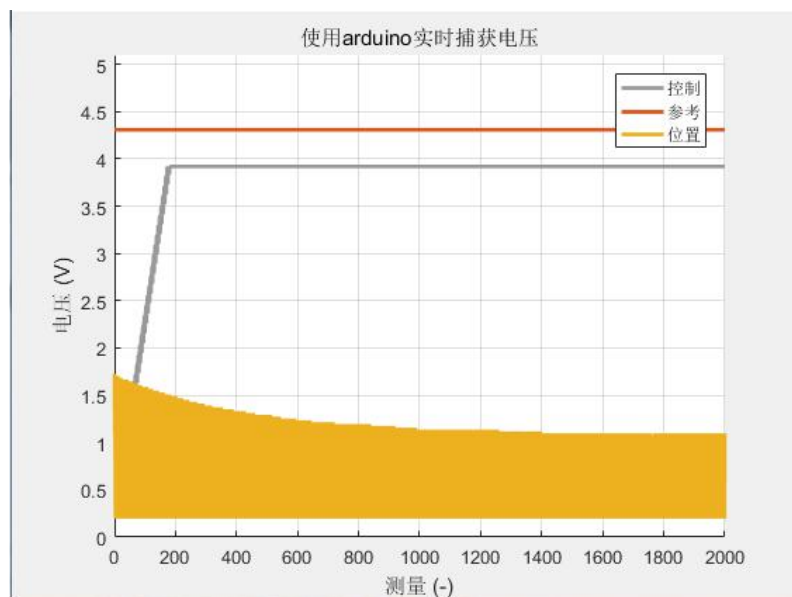


图 3 PID 整定后的阶跃响应曲线

2、方案的调试

在arduino 中运行程序，由于在 arduino 程序中加入了有串口绘图器的功能的代码，可以直接的从绘图器中得到电动机的 pwm 图，具体的图如下所示：

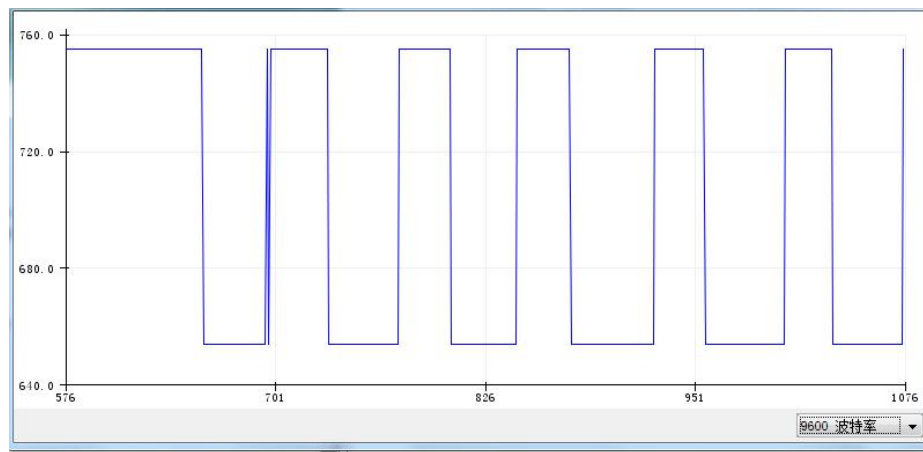


图 4 arduino 绘图器绘制的 pwm 图

3、总结

通过这次课程作业，对于 pid 三个环节的控制有更加深刻的认识，在选定 pid 参数的过程中也发现不少问题，选定的参数基本上难以得到理想的运行的结果，故这方面有待提高。Simulink 的仿真电路图基本是按照实物图来连接的，在设置电器元件参数的过程中，也大量的查找了网上的资源，对自己的能力有很大的提升。Arduino 串口与 simulink 相连，并在 arduino 中运行程序控 simulink 电路图，期间总是出现各种的问题，但还在最终基本上解决了。也是通过这次的作业，不仅仅让我对 arduino 的功能有个更加全面的理解，更是对它兴趣有所增加。也期待自己能在以后的学习的过程中发现更多有意思的东西。