

## 课程作业 2

潘奔 201820501031 机械工程 2 班

### 一、实验简介

以 Arduino 为硬件平台，利用 PC 端模拟电机控制系统，并且与 Arduino 进行串口通讯，具体要求如下：

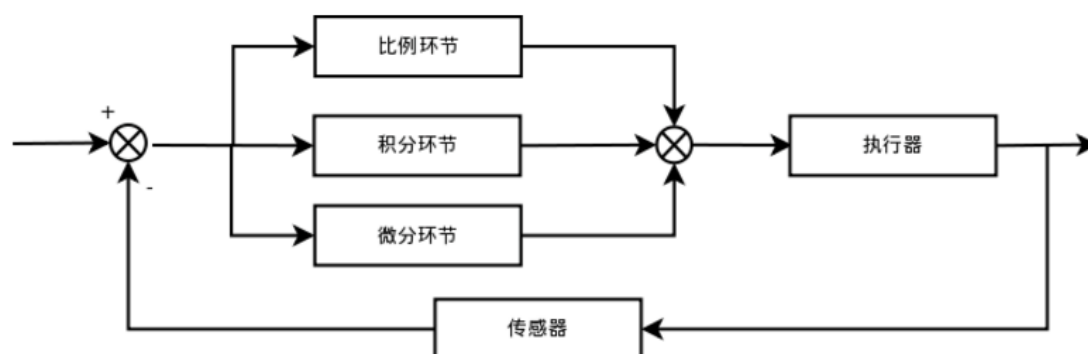
(1) 在 PC 中编写软件，运行一仿真程序，模拟一系统（例如温度控制系统，电机控制系统等），要求包含该系统的模型，以及控制接口。该控制接口能够接收来自串口的控制指令。

(2) 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。

(3) 分别运行上述实验系统，在 PC 端记录控制系统的状态曲线，绘制该曲线并进行说明。

### 二、实验原理

#### 2.1. Arduino——PID



本设计将 Arduino UNO 开发板充当 PID 调节器，外部数据从 PC 端通过串口输入到 Arduino 板子上，进行 PID 计算，将 PID 的输出数据再通过串口输出到 PC 端。

PID 算法的形式有多种，如 PID 离散化、位置型 PID、增量型 PID 等等，PID 控制实质就是对偏差的控制过程，如果偏差为 0，则比例环节不起作用，只有存在偏差时，比例环节才起作用。积分环节主要是用来消除静差，所谓静差，就是系统稳定后输出值和设定值之间的差值，积分环节实际上就是偏差累计的过程，把累计的误差加到原有系统上以抵消系统造成的静差。而微分信号则反应了偏差信号的变化规律，或者说是变化趋势，根据偏差信号的变化趋势来进行超前调节，从而增加了系统的快速性。

本设计选用增量型 PID 作为 PID 环节计算。增量型 PID 控制是通过对控制量的增量（本次控制量和上次控制量的差值）进行 PID 控制的一种控制算法。主要是通过求出增量，将原先的积分环节的累积作用进行了替换，避免积分环节占用大量计算性能和存储空间。增量式 PID 控制的主要优点为：

①算式中不需要累加。控制增量  $\Delta u(k)$  的确定仅与最近 3 次的采样值有关，容易通过加权处理获得比较好的控制效果；

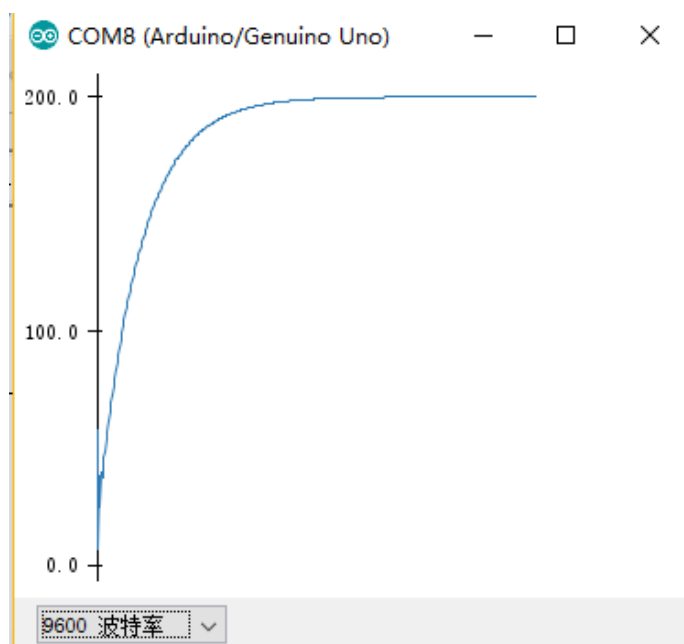
②计算机每次只输出控制增量，即对应执行机构位置的变化量，故机器发生故障时影响范围小、不会严重影响生产过程；

③手动—自动切换时冲击小。当控制从手动向自动切换时，可以作到无扰动切换。

增量型 PID 数学模型如下所示：

$$\Delta u[n] = K_p \{e[n] - e[n-1]\} + K_i e[n] + K_d \{e[n] - 2e[n-1] + e[n-2]\}$$

PID 调节能够使信号稳定调节到所设置的目标值，波形如下图所示：



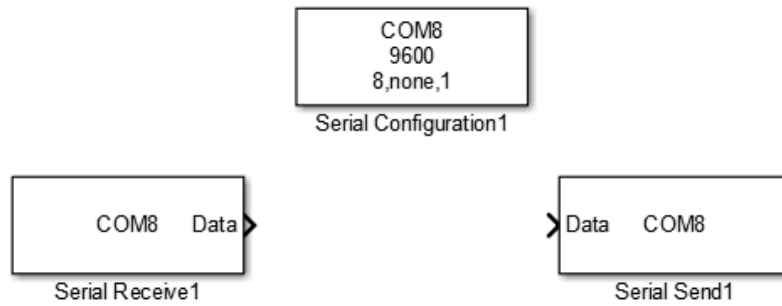
## 2.2. 仿真模型

本设计的仿真模型选用直流电机控制系统模型，利用 MATLAB 的 Simulink 仿真系统设计出直流电机的控制系统模型，并与 Arduino UNO 开发板进行串口通讯。

在 Arduino 上设定目标 PWM 值为 200（PWM 值范围为 0-255），系统初始化后经过 Arduino PID 计算，将 PID 输出值作为 PWM 占空比数值通过串口发送到 Simulink 模型中，直流电机控制系统模型根据接收到的 PWM 占空比，首先产生确定的 PWM 脉冲，再将 PWM 脉冲转化为电压值输入到 H 电桥模型中，从而驱动直流电机模型的运行，最后通过编码器模型测出直流电机的转速并通过串口返回到 Arduino 中，进行 PID 循环计算，直到电机转速稳定，系统稳定运行。

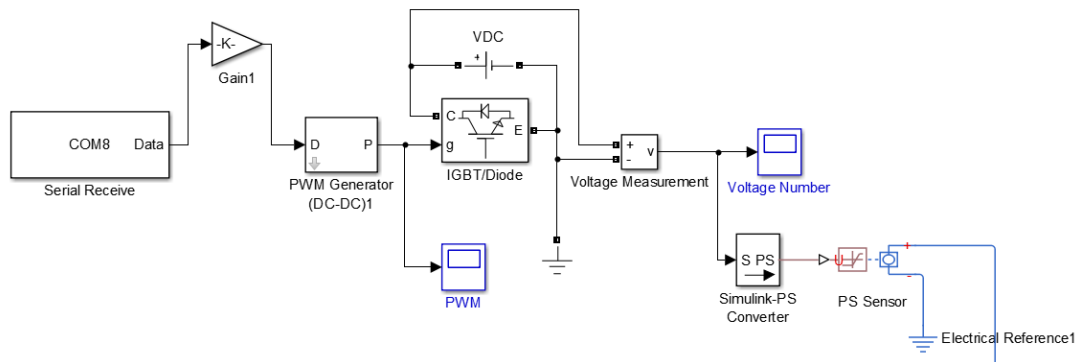
### 1) 串口通讯配置

Simulink 仿真模型与 Arduino 串口通信，需要在 Simulink 中配置串口信息，必须先将 Arduino 板子连接电脑再打开 MATLAB Simulink，在模型中需要用到串口配置，串口发送，串口接收三个模块：



## 2) PWM 占空比转化模型

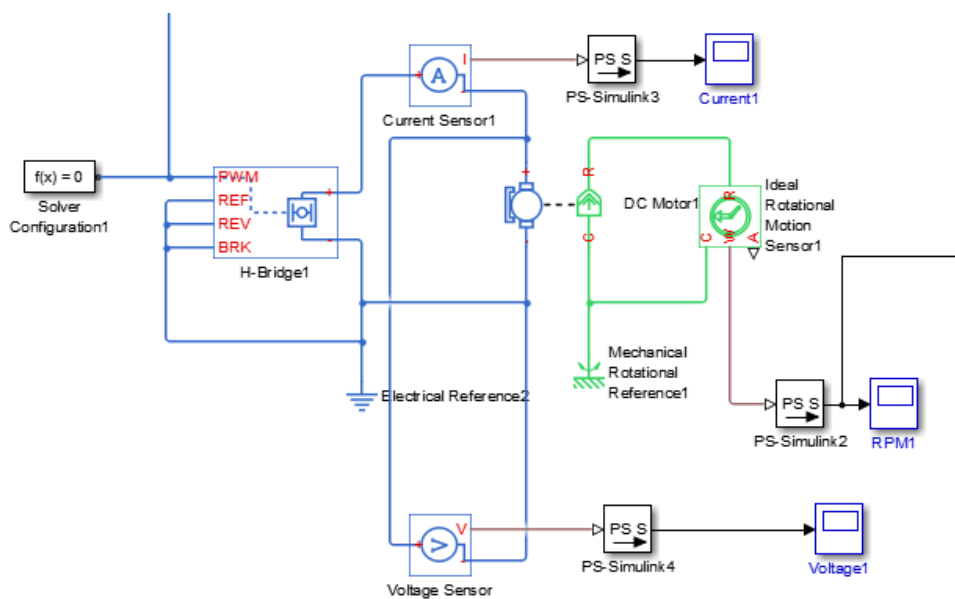
Arduino 可以通过串口将 PWM 脉冲波形、PWM 占空比数值与周期等信息发送到 PC 端，为了计算方便和程序解析方便，选取将 PWM 的占空比数值发送到 Simulink 上，直接在模型中将占空比进行转化为电机驱动所需要的电压值，PWM 的脉冲频率在 Simulink 在模块中设置。PWM 占空比转化模型如下所示：



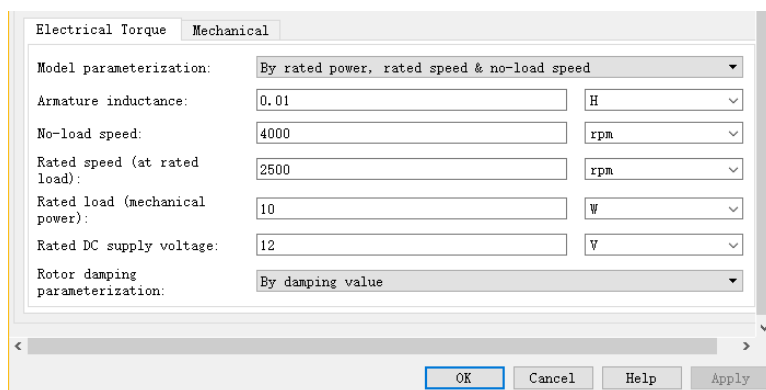
Simulink 中串口接收到的 uint8 数据（范围为 0-255）通过比例增益转化为范围为 0-1 的占空比，通过 PWM Generator 设置 PWM 频率为 1000Hz（与 Arduino 默认 PWM 频率一致）后产生 PWM 脉冲波形。PWM 脉冲通过 IGBT 功率管转化为实际产生的电压值。

## 3) 直流电机控制模型

本设计模型的直流电机采用 H 电桥驱动，通过 H 电桥输出电压值到直流电机上，驱动直流电机能够调速运行和正反运行。H 电桥以 0-5V 电压输入，0-12V 输出驱动直流电机。并且在 Simulink 上能够显示出直流电机的电流值、电压值与转速。

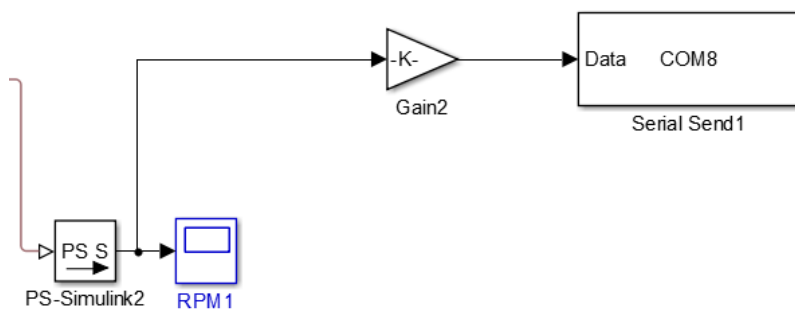


本设计选取的直流电机参数为额定电压 12V，无负载最高转速 4000rpm



#### 4) 转速输出模型

直流电机的转速需要通过串口反馈到 Arduino 上进行下一步的 PID 计算，为简化 Arduino 程序，直流电机输出转速范围 0-4000 通过比例增益转换为范围为 0-1 的数值后再通过 Serial.Send 发送相应的 uint8 到 Arduino 上。



### 三、程序代码与仿真模型

Arduino—PID 程序代码：

```
#define PIN_INPUT 0
#define PIN_OUTPUT 3

//PID 参数结构体
struct _pid{
    float SetSpeed;
    float ActualSpeed;
    float err;
    float err_next;
    float err_last;
    float Kp,Ki,Kd;
}pid;

//PID 初始化，配置初始参数
void PID_init(void) {
    pid.SetSpeed = 0.0;
    pid.ActualSpeed = 0.0;
    pid.err = 0.0;
    pid.err_last = 0.0;
    pid.err_next = 0.0;
    pid.Kp = 0.2;
    pid.Ki = 0.03;
    pid.Kd = 0.2;
}

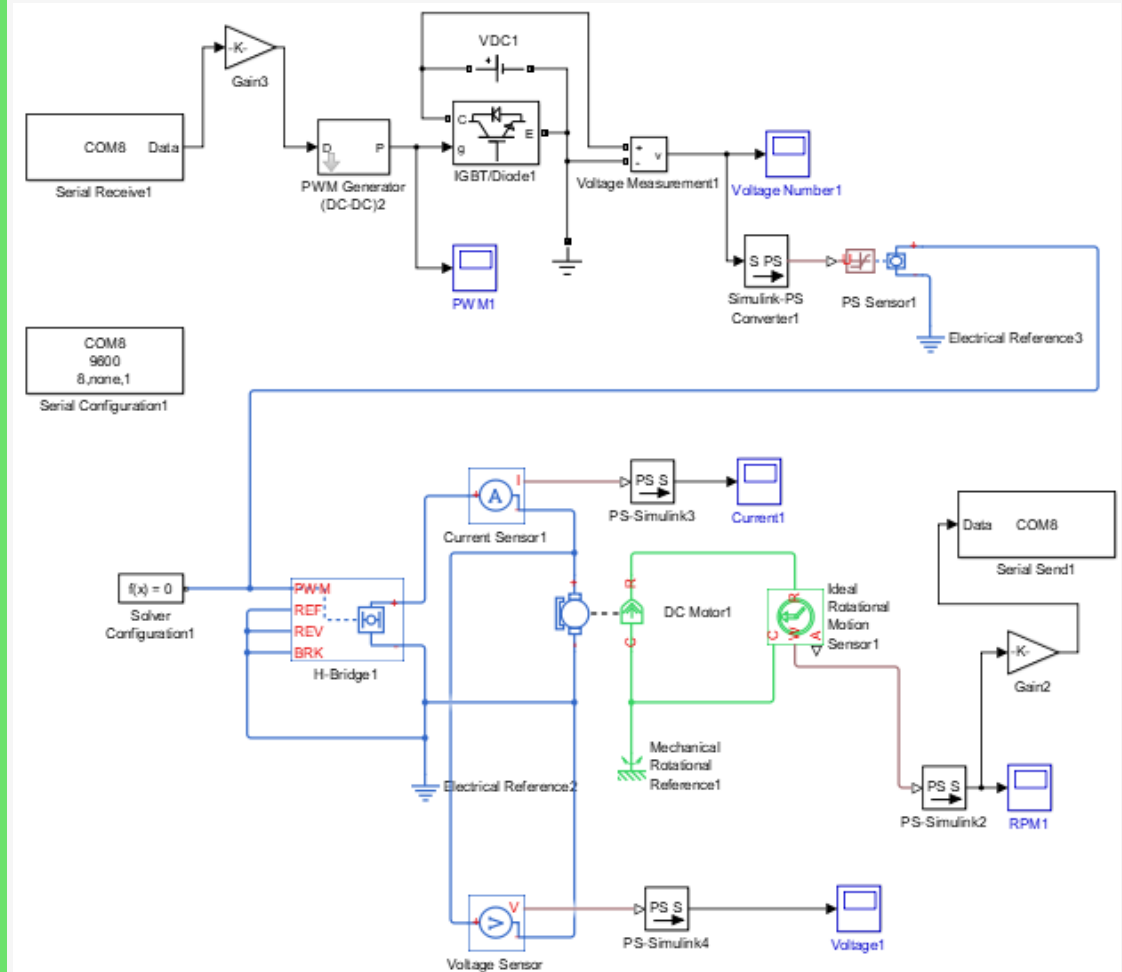
//PID 计算函数，输入要设定的目标值，输出当前计算后的值
float PID_realize(float sp) {
    pid.SetSpeed = sp;
    if(Serial.available()>0)
    {
        pid.err = pid.SetSpeed - Serial.read();
    }
    else pid.err = pid.SetSpeed - pid.ActualSpeed;
//PID 计算公式
    float incrementSpeed = pid.Kp*(pid.err - pid.err_next)+
pid.Ki*pid.err+pid.Kd*(pid.err - 2*pid.err_next + pid.err_last);
    pid.err_last = pid.err_next;
    pid.err_next = pid.err;
    pid.ActualSpeed += incrementSpeed;
    return pid.ActualSpeed;
```

```
}

//Arduino 初始化函数
void setup()
{
    Serial.begin(9600);
    pinMode(PIN_OUTPUT, OUTPUT);
    pinMode(PIN_INPUT, INPUT);
    PID_init();
}

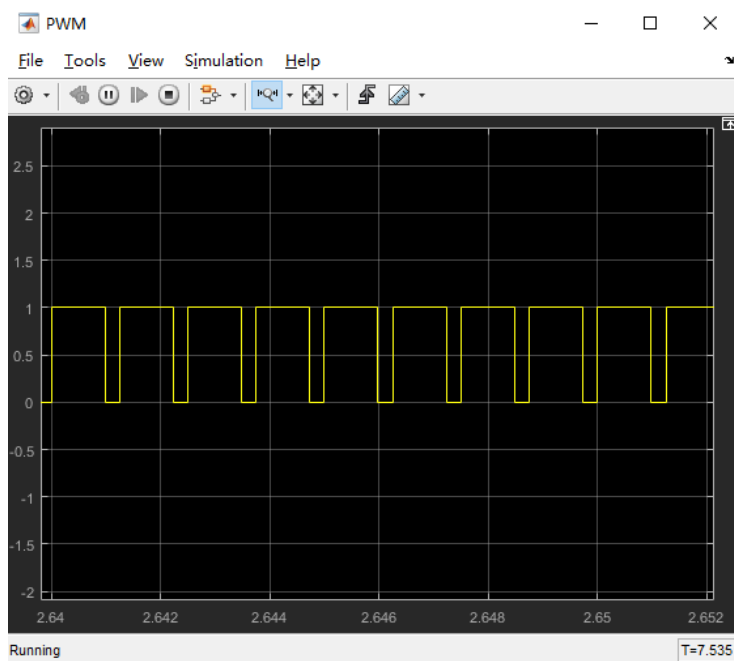
//Arduino 循环执行函数
void loop()
{
    int count = 0;
    while(count<100)
    {
        float pwm = PID_realize(200);
        count++;
        Serial.println(pwm);//发送 PWM 值（0-255）到串口
        delay(200);
    }
}
```

## Simulink 直流电机控制系统仿真模型：

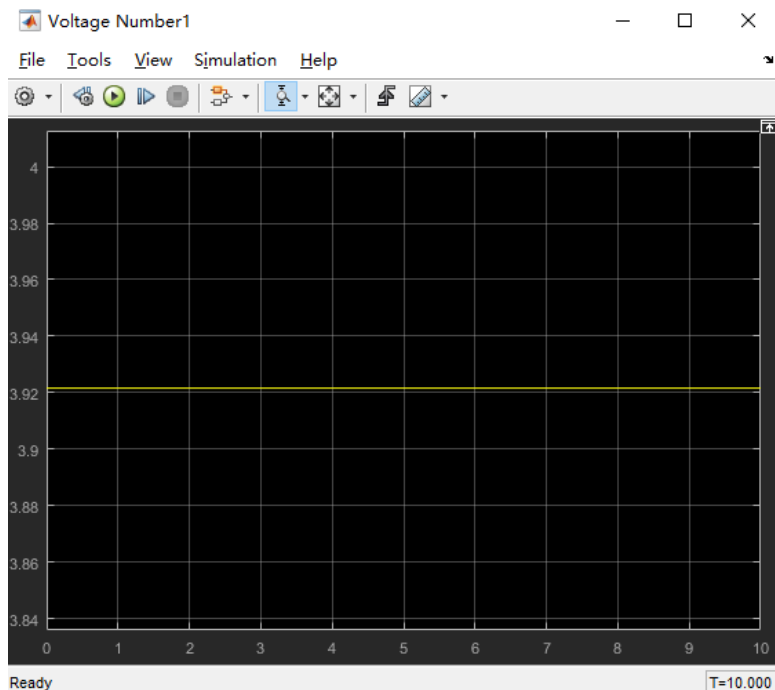


#### 四、实验结果

设定目标值为 200，即最后稳定时，PWM 输出脉冲为频率 1000Hz，占空比为 0.784，通过 Simulink 示波器模块 PWM 得到 PWM 脉冲波形：

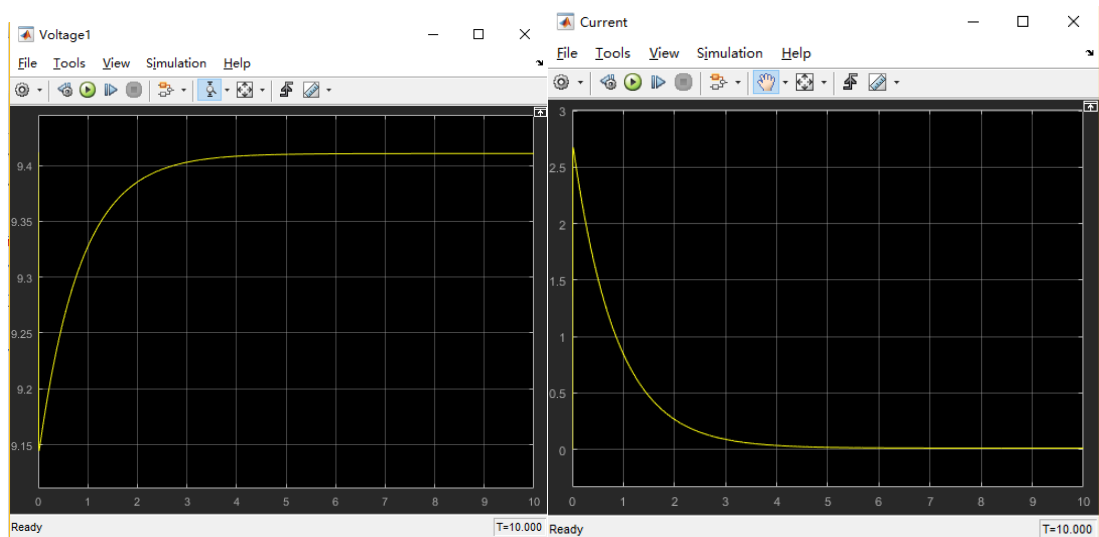


通过 Simulink 示波器模块 Voltage Number1 得到此时以 5V 为参考电压，PWM 脉冲等价电压为 3.92V：



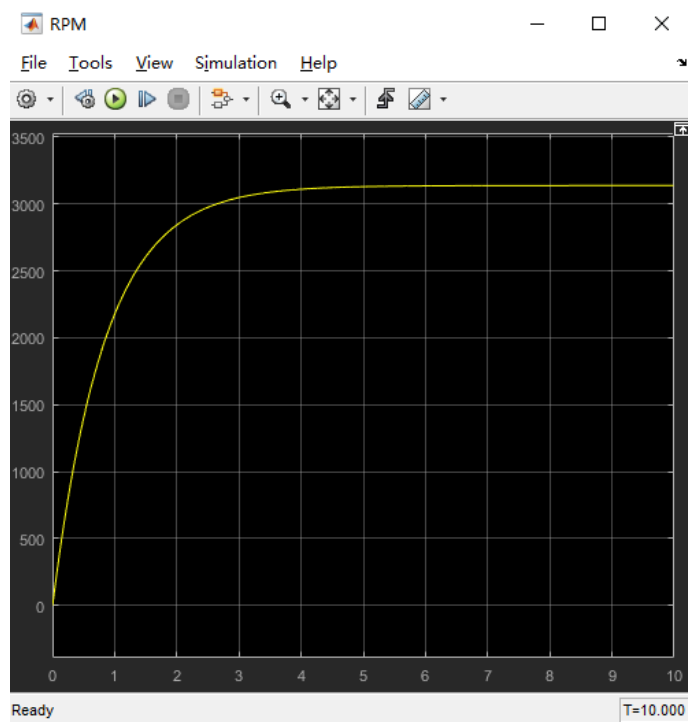
即 H 电桥以输入电压为 3.92V 控制直流电机，通过 Voltage Sensor1 电压表与 Current Sensor 电流表，测出 H 电桥的输出电压值与输出电流值，即直流电机运行的电压值与电流值，通过示波器模型得到波形：





电压值逐步上升，最后稳定在 9.4V 左右，即 H 电桥提供 9.4V 的电压驱动直流电机。直流电机的启动电流一般较大，所以开始时，测得的电流值会很高，由于本设计的直流电机空载运行，所以最终电流值会趋向于 0A。

最后通过示波器模型 RPM，测得本设计空载直流电机的转速如下所示，最终稳定在 3137rpm。



Arduino PID 调节 PWM 占空比数值的波形为

