



中山大學

SUN YAT-SEN UNIVERSITY

实 验 报 告

课程名称： 电机与拖动技术

姓 名： 方桂安

学 号： 20354027

专业班级： 2020 级智能科学与技术

任课教师： 冯国栋

2022 年 12 月 15 日

实验二 永磁同步电机控制实验

一、 实验目的

1. 通过连接永磁同步电机的三相电线和位置传感器线到试验箱，并运行永磁同步电机程序，实现电机的转动。
2. 通过修改参数 PI_speed: Kp, Ki 来观察实验现象，对电机进行转速外环控制。
3. 通过采集电压，电流，转速等参数并通过 DAC 输出，对电机进行监测和分析。

二、 实验原理

首先将永磁同步电机的三相电线与位置传感器线连接到试验箱，下载永磁同步电机程序并运行，实现电机转动。

代码简介：

```
InitPeripherals(); //Initialize all the Device Peripherals
```

初始化 ADC, DAC 等，后面会用到 ADC 与 DAC

```
InitVariables(); //Initialize Variables
```

初始化参数

需要特别关注其中的：

PI_speed Kp, Ki, 后面需要修改参数，观看实验现象。

程序一直运行 for(;;) 里面的代码执行条件判断

```
if ((pmsm_set.start == 1) && (pmsm_set.run == 0)) //起动电机条件判断
if (pmsm_set.start == 0) //停止电机判断
```

电机控制运行在中断中，中断是每个固定时间执行一次，执行过程中主程序暂停执行。

电机控制程序在下面中断中运行

```
interrupt void EPWM1_Isr(void)
//GpioDataRegs.GPASET.bit.GPIO23 = 1; //LED_Run 点亮 LED 等
ADC_Ctrl(); //ADC 采样采集电流，电压
```

见下面的代码

```
pmsm.udc = ((float) Get_A_Adc(0)) * 0.0048828125;
pmsm.idc = ((float) Get_A_Adc(1)) * 0.00091552734375;
pmsm.iu = -((float) Get_A_Adc(2)) * 0.00152587890625 + 12.5;
pmsm.iv = -((float) Get_A_Adc(3)) * 0.00152587890625 + 12.5;
pmsm.iw = -((float) Get_A_Adc(4)) * 0.00152587890625 + 12.5;
pmsm.exadc = ((float) Get_A_Adc(5)) * 0.00030517578125;
```

电机控制程序（电流内环控制）

```
Motor_Ctrl(&pmsm);
```

电机控制程序（转速外环控制）

PMSMAPP 每隔固定时间运行（比电流控制要慢）

DAC 输出范围是-10V~10V, 所以电压 电流 转速等需要做相应的缩放，使得能够正确的输出到 DAC 上。

```
WriteDAC(DA_ADD1, pmsm.udc * 0.01);
WriteDAC(DA_ADD2, pmsm.speed * 0.0005);
WriteDAC(DA_ADD3, pmsm.idc * 100);
```

注意：

a. 电机运行的数据存在 pmsm 结构里面

b. 控制器参数都存在 PI_Speed, PI_Id, PI_Iq 等结构里面

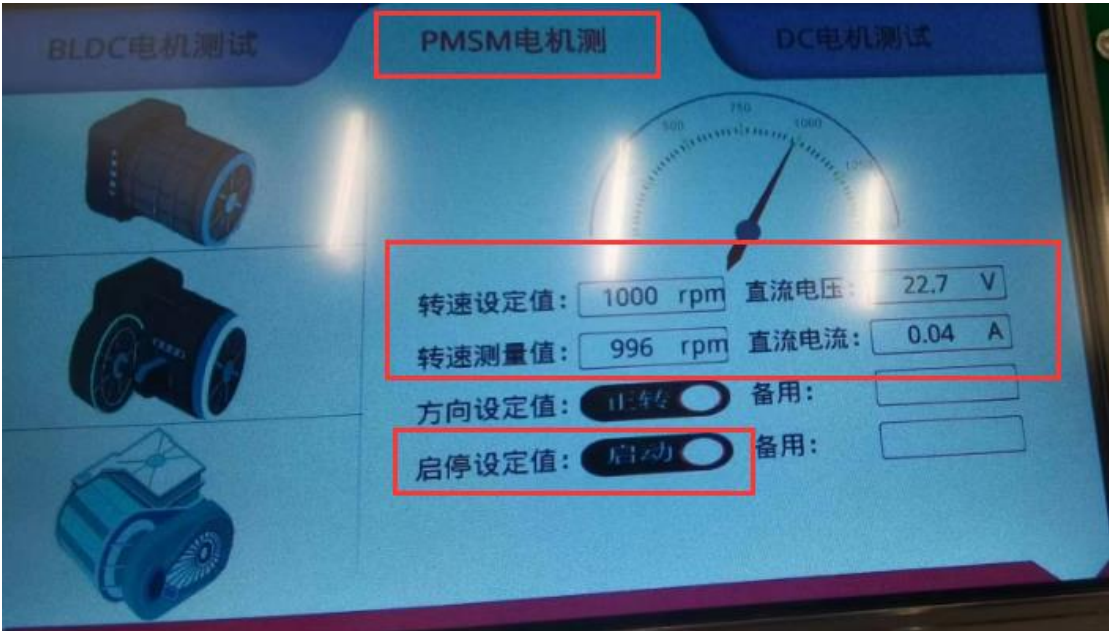
三、实验设备

(1) 硬件：BOX28335 实验平台，仿真器 HDSP-XDS200ISO，相应的配套电源。

(2) 软件：安装了 Windows7/10 和 CCS 软件的 PC。

四、实验过程

1. 启动电机，观察电机运行，包括电机转速，位置，三相电流，直流电压等参数（通过 CCS 设置转速等参数的值，观察上述参数的值与波形）。



正确连接设备之后，选择 PMSM 电机测试并将启停设定值改为启动，屏幕上即可显示当前电机的相关数据：设定转速、实际转速、直流电压、直流电流。

除了实验箱的屏幕，也可以从 CCS 的变量读出各项参数的具体数值：

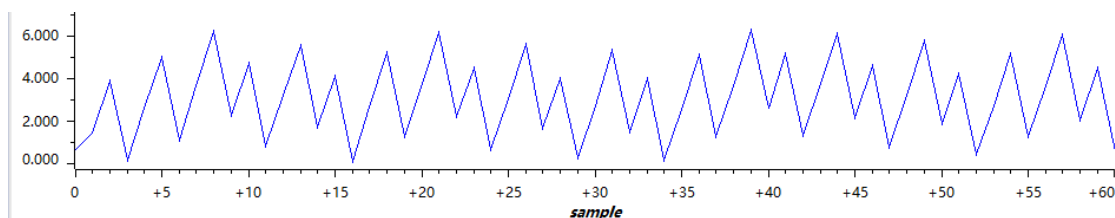
(*) pmsm.position	float	5.30031967	0x0000A496@Data
(*) pmsm.speed	unsigned int	996	0x0000A498@Data
(*) Pl_Speed.SetPoint	float	1000.0	0x0000A468@Data
(*) pmsm.udc	float	22.734375	0x0000A49A@Data
(*) pmsm.idc	float	0.0494384766	0x0000A4A2@Data

上图中展示了电机位置、设定转速、实际转速、直流电压、直流电流。

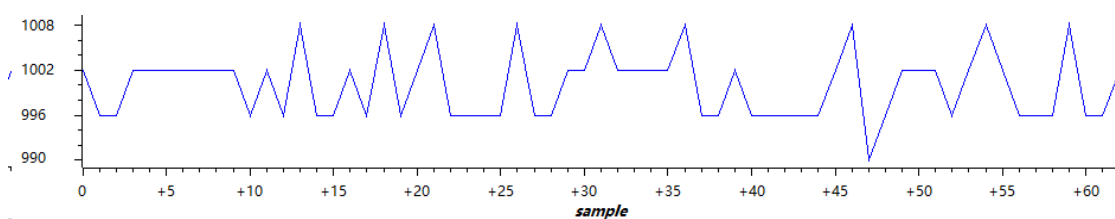
④= pmsm.iu	float	0.00610351563	0x0000A49C@Data
④= pmsm.iv	float	-0.244140625	0x0000A49E@Data
④= pmsm.iw	float	0.190734863	0x0000A4A0@Data

上图中展示了电机的三相电流。

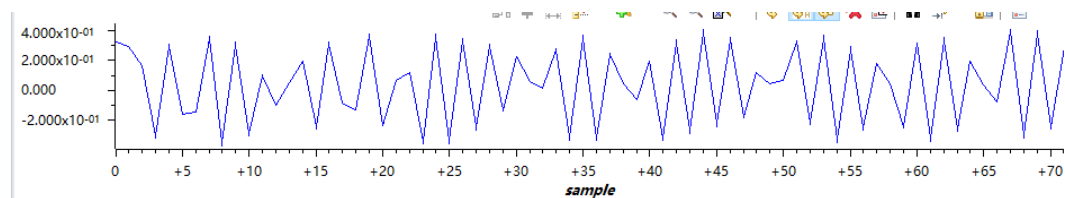
CCS 还可以将参数实时变化可视化，效果如下：



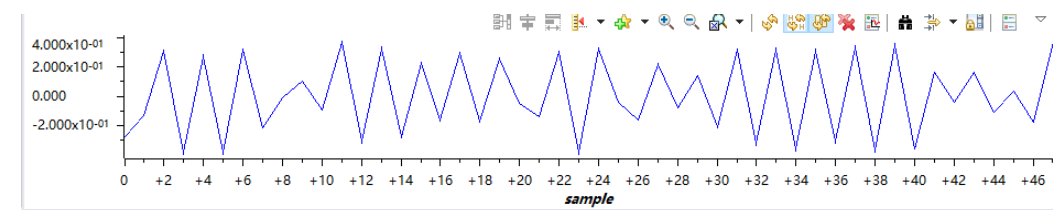
position 变化图



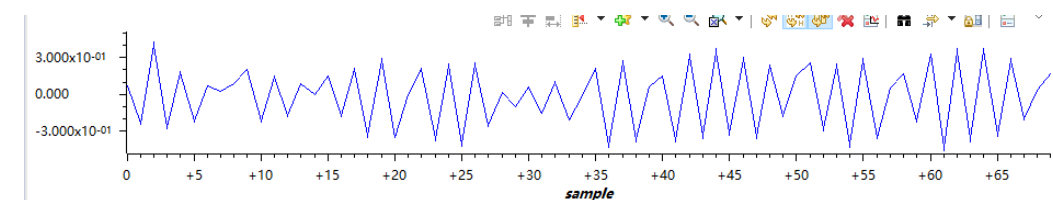
speed 变化图



三相电流 i_u 变化图



三相电流 i_v 变化图



三相电流 i_w 变化图

2. 通过示波器观察 DAC 输出的电机的位置，转速，电流，电压等波形（DAC 没有输出电流、位置等，需要添加代码 `WriteDAC(DA_ADDxx, xxxx);` xx 与 xxxx 需要自己修改。特别的，用示波器同时观察三相电流与位置波形，分析位置一个周期是否是极对数*电流周期。

添加代码如下：

```
157 WriteDAC(DA_ADD0, pmsm.position*0.1);
158 WriteDAC(DA_ADD1, pmsm.udc * 0.01);
159 WriteDAC(DA_ADD2, pmsm.speed * 0.0005);
160 WriteDAC(DA_ADD3, pmsm.idc * 10);
```

上述代码显示 DAC1-ADC，DAC2，DAC3，DAC4 四个通道分别会显示电机的位置、直流电压、转速、直流电流 4 个参数。

阅读 main.c 中的代码 `InitVariables()` 中的参数初始化显示，电机极对数为 4。

```
void InitVariables(void)
{
//Initialize system variables
    MAIN_LOOP.cnt1 = 0;
    MAIN_LOOP.cnt2 = 0;

//Insert user code

    pmsm.idc = 0;
    pmsm.pole = 4; //电机极对数
    pmsm.speed = 0;
    pmsm.udc = 0;
    pmsm.direction = 1;
    pmsm.error = 0;
}
```

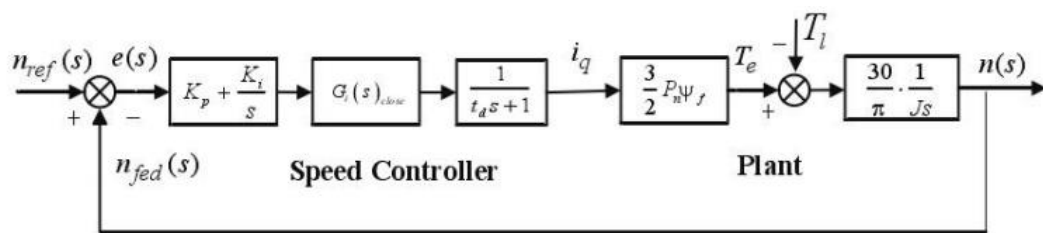
用示波器同时观察三相电流与位置波形，我们初步认为位置周期是电流周期的四倍。继续查阅资料得知：

PMSM（永磁同步电机）的位置周期与电流周期之间的关系取决于电机的构造和控制方式。

在单电机体系中，如果电机的磁链控制方式是通过模拟电流控制电机的转矩来实现的，那么电机的位置周期与电流周期之间的关系就是电流周期的极对数次倍。如果电机的磁链控制方式是通过脉冲宽度调制（PWM）的方式来实现的，那么电机的位置周期与电流周期之间的关系就取决于 PWM 的频率和占空比。

一般编码器得到的位置是机械角度，如果要用到控制中，一般会乘以极对数，从而得到电角度。

3. 分别修改转速 PI 控制的比例与积分参数，观察对转速控制性能的影响（响应时间，超调，控制精度等）



转速环里面包含有电流环在内，当外面给定转速时，系统通过控制 i_q 来控制电磁转矩，从而电机开始转动。电机在转动的过程，传感器检测到机械角度会改变，也就是说电角度也会改变，由电角度在单位时间的变化，就可以求出电角速度，从而就可以得到反馈回来转速参与转速决策，最终可以控制转速稳定。

本次实验中转速的初始值为 1000，我们会将设定值调整到 1500，研究该阶跃响应下 PI 控制的系统动态性能，使用控制变量法对比分析比例与积分参数对控制效果的影响。

利用 CSS 自带的 graph 功能我们可以绘制出对应的阶跃响应曲线，然后在图像上右键选择 data，可以把数据以 csv 格式导出，我在 matlab 中编写了一

个函数利用导出的 csv 数据来计算此时系统的动态性能指标，代码如下：

```
function [ys,tr,ts,tm,ov] = Fun_Step_Performance(t,y,drawflag)
% [ys,tr,ts,ov] = Fun_Step_Performance(t,y) 标准阶跃响应的性能指标求解
% 本程序适用于标准阶跃响应曲线，末尾时间必须已经接近稳态值
% t-y 为阶跃响应的时间-输出配对序列，可由[y,t] = step(sys)求得
% drawflag 为时候作图标志，不输入或输入非 0 值时，默认作图，输入 0 时不做图
% ys 稳态值
% tr 上升时间，默认为 0-90%的上升时间
% ts 调整时间，默认为 2%的调整时间
% tm 为峰值时间
% ov 超调量 %
% e.g.
% sys = tf(1,[1 2*0.5*1 1]);
% [y,t] = step(sys,15);
% [ys,tr,ts,tm,ov] = Fun_Step_Performance(t,y,1);

if nargin == 2
    drawflag = 1; % 默认绘图
end

ys = y(end); % 稳态增益
[ym, ind] = max(y); % 最大输出
ov = 100*(ym-ys)/ys; % 超调量
tm = t(ind); % 峰值时间
ind2 = length(t);
delta = 0.02; % 调整时间默认范围 2%
while t(ind2) > 0
    if abs(y(ind2)-ys) >= delta*ys
        break
    end
    ind2 = ind2-1;
end
ts = t(ind2); % 调整时间
ind3 = 1;
while y(ind3) < 0.9*ys
    ind3 = ind3 + 1;
end
tr = t(ind3); % 上升时间

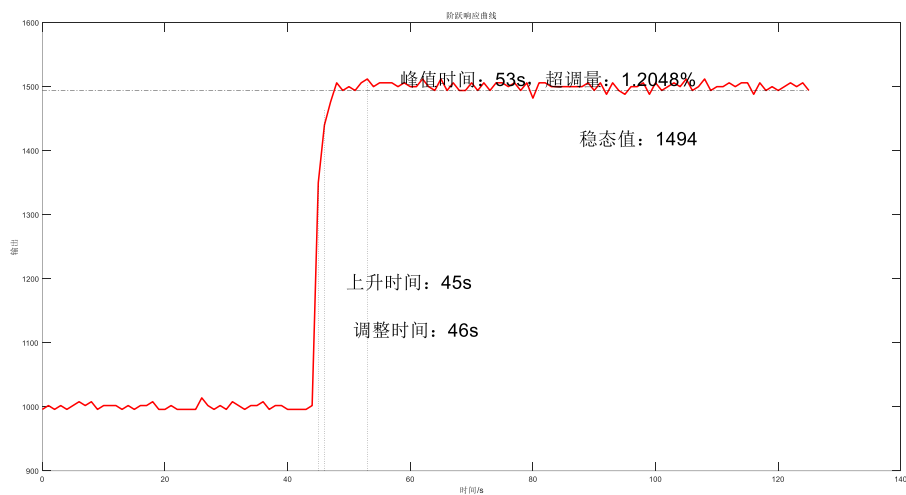
if drawflag ~= 0
    figure
    plot(t,y,'r',LineWidth=2)
    hold on
```



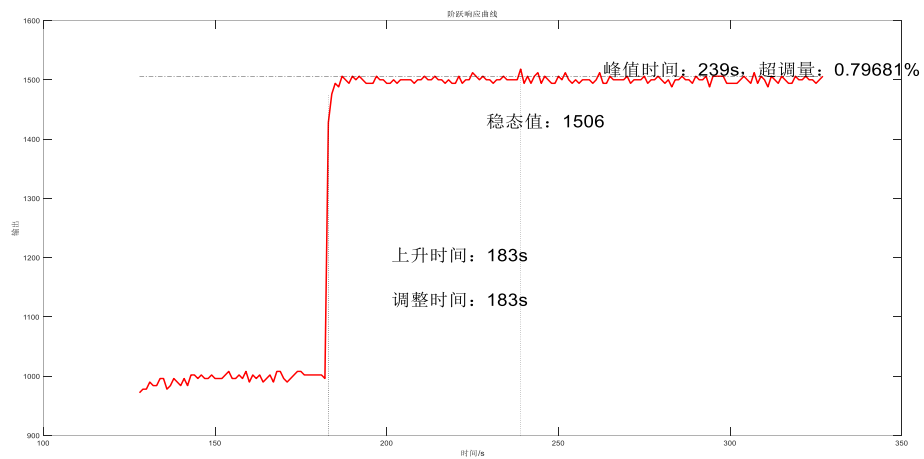
```

plot([tr tr],[0 0.9*ys],'k:')
plot([tm tm],[0 ym],'k:')
plot([ts ts],[0 (1-delta)*ys],'k:')
plot([t(1) t(end)],[ys ys],'k-')
xlabel('时间/s')
ylabel('输出')
title('阶跃响应曲线')
ylim([900,1600])
text(1.1*tr,0.8*ys,['上升时间: ' num2str(tr) 's'])
text(1.1*ts,0.75*ys,['调整时间: ' num2str(ts) 's'])
if abs(tm-t(end)) > 0.1*tm
    text(1.1*tm,1*ym,['峰值时间: ' num2str(tm) 's, 超调量: '
num2str(ov) '%'])
else
    text(0.55*tm,0.8*ym,['峰值时间: ' num2str(tm) 's, 超调量: '
num2str(ov) '%'])
end
text(0.7*t(end),0.95*ys,['稳态值: ' num2str(ys)])
disp('% 阶跃响应指标结果: ')
disp(['上升时间: ' num2str(tr) 's'])
disp(['调整时间: ' num2str(ts) 's'])
disp(['峰值时间: ' num2str(tm) 's, 超调量: ' num2str(ov) '%'])
disp(['稳态值: ' num2str(ys)])
disp('% 阶跃响应指标结果显示结束')
end

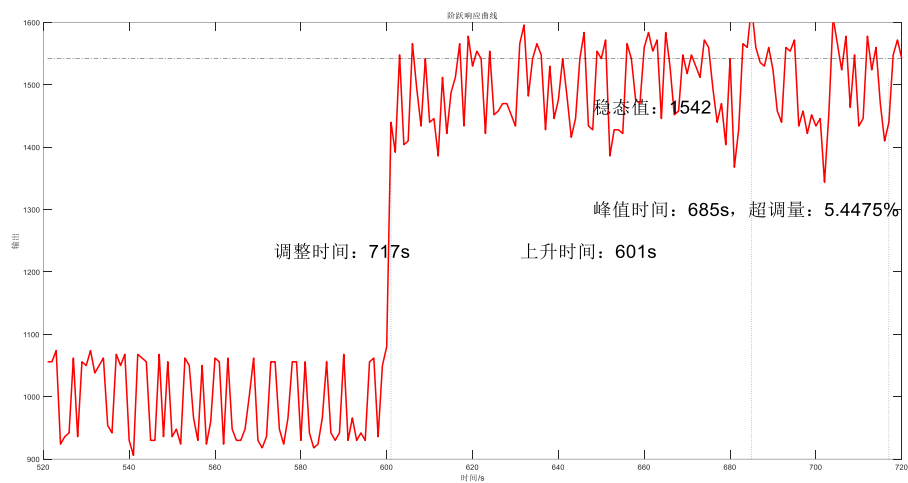
```



$K_p=0.02$, $K_i=0.5$ 的阶跃响应曲线



$K_p=0.04$, $K_i=0.5$ 的阶跃响应曲线



$K_p=0.02$, $K_i=1.0$ 的阶跃响应曲线

动态性能指标表格如下:

参数	稳态 值 $C(\infty)$	超调 量 $\sigma\%$	峰值 时间 t_p	调节 时间 t_s	上升 时间 t_r
$K_p=0.02$ $K_i=0.5$	1494	1.2%	53s	46s	45s
$K_p=0.04$ $K_i=0.5$	1506	0.8%	239s	183s	183s
$K_p=0.02$ $K_i=1.0$	1542	5.4%	685s	717s	601s

比例调节作用：按比例反应系统的偏差,系统一旦出现了偏差,比例调节立即产生调节作用用以减少偏差。比例作用大,可以加快调节,能迅速反应误差,从而减小稳态误差。但是,比例控制不能消除稳态误差。过大的比例,使系统的稳定性下降,甚至造成系统的不稳定

积分调节作用：使系统消除稳态误差,提高无差度。积分控制的作用是,只要系统有误差存在,积分调节就进行,积分控制器就不断地积累,输出控制量,直至无差,积分调节停止,积分调节输出一常值。因而,只要有足够的时间,积分控制将能完全消除误差,使系统误差为零,从而消除稳态误差。积分作用的强弱取决于积分时间常数 T_i , T_i 越小,积分作用就越强,积分作用太强会使系统超调加大,甚至使系统出现振荡,反之 T_i 大则积分作用弱。加入积分调节可使系统稳定性下降,动态响应变慢。

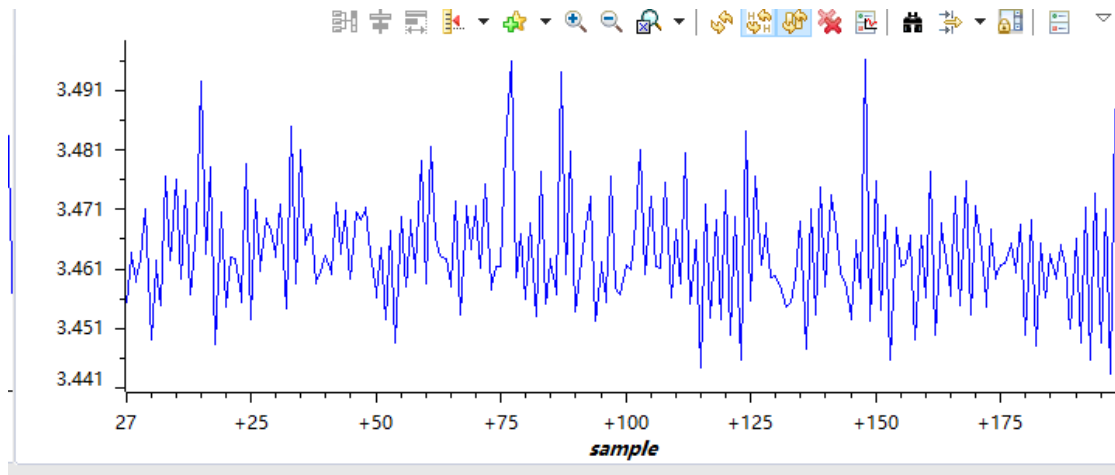
不难看出三组参数中 $K_p=0.04$, $K_i=0.5$ 的效果是最好的,适当增加的 K_p 使得系统更快地调节,减小稳态误差;较小的 K_i 防止系统超调增大,甚至振荡。

4. 研究空载时电压与转速的关系 (电压为 $\sqrt{u_d^2+u_q^2}$) 并画出曲线,其中 u_d 为 PI_{Id} 的输出, u_q 为 PI_{Iq} 的输出,转速为传感器测量,(需要自己添加代码计算 u ,)。记录不同转速对应的电压,离线画出关系曲线,分析规律。

在 `control.c` 中我们可以找到题目中的 u_d 、 u_q , 利用全局变量,在代码中添加以下内容来计算 u :

```
extern float u = 0;  
u = sqrt(TRS_MPR_U.d*TRS_MPR_U.d+TRS_MPR_U.q*TRS_MPR_U.q)
```

利用 CSS 观察不同转速下的 u , 结果如下:



转速设定值为 1000 时，u 的变化曲线

(x) pmsm.speed	unsigned int	1002	0x0000A498@Data
(x) PI_Speed.Kp	float	0.009999999978	0x0000A470@Data
(x) PI_Speed.Ki	float	1.0	0x0000A472@Data
(x) PI_Speed.SetPoint	float	1000.0	0x0000A468@Data
(x) pmsm.position	float	3.61099982	0x0000A496@Data
(x) u	float	3.4498651	0x0000A3F2@Data

通过修改转速设定值，记录不同情况的 u，列表如下：

Speed	500	750	1000	1250	1500
u	1.87076688	2.65094876	3.4498651	4.24564314	5.04943609

观察数据，我们认为转速与电压呈一次函数关系，利用 python 分析数据，得

到 u 与 speed 之间的规律：

```
# 分析二者的函数关系表达式
import plotly.graph_objects as go
import numpy as np
from scipy.optimize import curve_fit

# 电机转速
speed = [500, 750, 1000, 1250, 1500]

# 对应的电压 u
u = [1.87076688, 2.65094876, 3.4498651, 4.24564314, 5.04943609]

# 定义函数
def func(x, a, b):
    return a * x + b

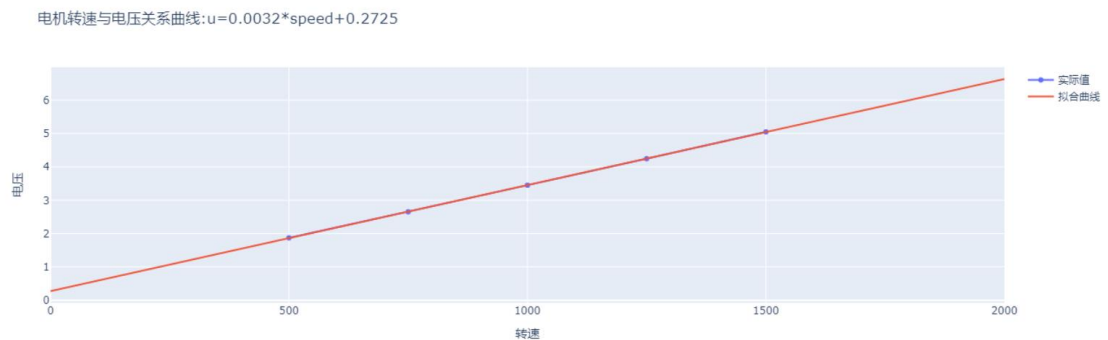
# 拟合
popt, pcov = curve_fit(func, speed, u)

# 绘制拟合曲线
```

```

x = np.linspace(0, 2000, 100)
y = func(x, popt[0], popt[1])
fig = go.Figure()
fig.add_trace(go.Scatter(x=speed, y=u, mode='lines+markers', name='
实际值'))
fig.add_trace(go.Scatter(x=x, y=y, mode='lines', name='拟合曲线'))
fig.update_layout(title='电机转速与电压关系曲
线:'+u='+str(round(popt[0], 4))+ '*speed'+str(round(popt[1], 4)),
axis_title='转速', yaxis_title='电压')
fig.show()

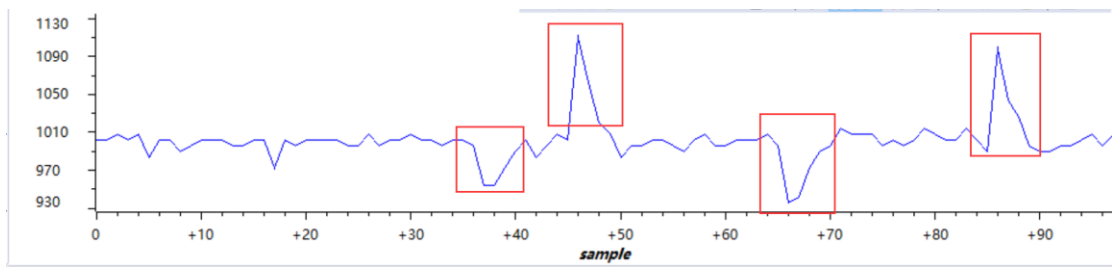
```



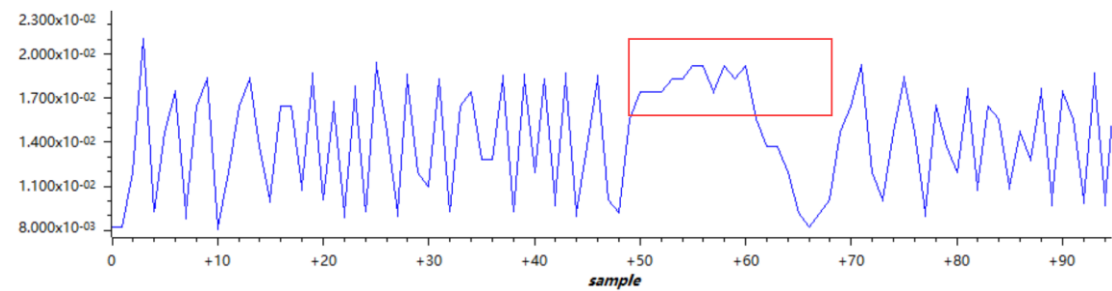
5. 尝试添加负载, 观察电流、转速、转矩的变化, 并记录实验现象 ($T_e=K_1*\text{flux}*I_q$, $\text{voltage}=K_2*\text{flux}*speed$); 观察 q 轴电流与施加负载的关系。

首先捏住电机来添加负载, 使得转矩产生变化





观察转速变化图，原本稳定在 1000 左右的值出现突变。



观察电流变化图，可以发现添加负载之后电流变大了，虽然还是在波动，但波动的范围相较于添加负载前有显著上升。

6. 尝试设置一些条件点亮 LED 灯（如电机启动灯亮，超过多少灯闪等

GpioDataRegs.GPATOGGLE.bit.GPIO23 对应的 LED 可以用）（选做）

思路：在主函数中利用一个 counter, for 循环中对 counter++; 然后判断 counter 小于 500 点亮灯, counter 大于 500 熄灭灯, counter 大于 1000, 重置 counter=0; 我们在电机启动的主函数中增加了一个计数用的变量 counter, 并且会随着主循环的运行不断 counter++, LED 亮灭逻辑如题目中所述。

```

if (counter<500){
    GpioDataRegs.GPASET.bit.GPIO23=1;//RUN
}
else{
    GpioDataRegs.GPASET.bit.GPIO23=0;//RUN
}
if (counter>1000){
    counter=0;
}

```



Expression	Type	Value
(x)= pmsm.idc	float	0.0128173828
(x)= pmsm.speed	unsigned int	1008
(x)= pmsm.iu	float	0.239562988
(x)= pmsm.iv	float	-0.267028809
(x)= pmsm.iv	float	0.0946044922
(x)= counter	float	505.0

效果符合实验预期，实验成功。

五、 实验心得与感受

1. 通过观察电机运行的参数，如转速、位置、三相电流、直流电压等，可以了解电机的运行情况，并通过调整 CCS 设置的参数值，观察这些参数的变化，从而了解电机的性能。
2. 通过观察 DAC 输出的电机参数波形，可以更好地理解电机的运行原理，并分析电机的控制性能。
3. 通过修改转速 PI 控制的比例与积分参数，可以观察对转速控制性能的影响，从而优化电机的控制性能。
4. 通过添加负载，可以了解电机在有负载时的运行情况，并观察电流、转速、转矩的变化，进一步深入了解电机的运行原理。
5. 通过设置一些条件点亮 LED 灯，可以更直观地观察电机的运行情况，并可以根据实际需要进行相应的调整。