中山大學 SUN YAT-SEN UNIVERSITY

电机拖动技术实验报告

院(系):智能工程学院 学号：22354189 姓名：张瑞程

日期：2024 年 12 月 实验名称：第 2 次实验-永磁同步电机 PMSM

实验二：永磁同步电机 PMSM

目录

实验二：永磁同步电机PMSM

1. 电机启动

1.1. 实验目的

（1）实现电机的转动；
（2）尝试对电机参数的初步观察。

1.2. 电机原理

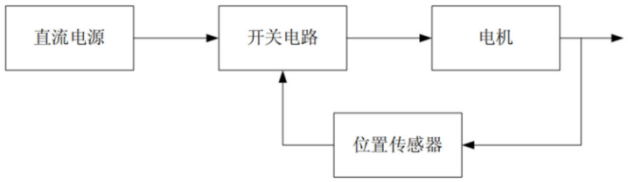
目前，电机可以分为两类：**直流电机**和**交流电机**。

（1）**直流电机**是最早出现的电机，也是最早实现调速的电机，具有良好的线性调速特性、简单的控制性能、高质高效平滑运 转的特性，不过由于电刷和换向器的存在阻碍了它的发展，逐渐被交流电机所取代。

（2）**交流电机**按照转子材料等的不同，可以分为同步电机和异步电机，其中根据感应电动势的不同，同步电机可以分为永磁同步电机和直流无刷电机，永磁同步电机的感应电动势为
正弦波，直流无刷电机的感应电动势为梯形波。

直流无刷电机利用电子换相器件取代了机械电刷和机械换相器，因此不仅保留了直流电机的优点，而且又具有交流电机结构简单、运行可靠、维护方便等特点。直流无刷电机的转子是由永磁材料制成的，具有一定的磁极对数的永磁体。无刷电机的转子磁钢的形状呈弧形，产生梯形波感应电动势。

直流无刷电机的工作离不开电子开关电路，因此由电动机本体、转子位置传感器和电子开关电路三部分组成了直流无刷电机的控制系统，其原理框图如下所示。直流电源通过开关电路向电动机定子绕组供电，位置传感器随时检测到转子所处的位置，并根据位置信号来控制开关管的导通和截至，从而自动地控制了哪些绕组通电，哪些绕组断电，实现了电子换相。



在这里是通过改变DSP 输出方波信号（PWM 信号）的占空比来达到对电机调速的目的。直流无刷电机的 PWM 产生主要是以连续增/减方式，带有死区控制，其具有以下特点：

- 通过改变 PWM 载波频率来改变 PWM 频率；

- 可根据需要改变 PWM 的占空比；
- 可改变 PWM 输出的极性，如高、低、强制高和强制低；
- 可设置死区的大小。

1.3. 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。

1.4. 设备连接与变量观察

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，然后测试电脑与试验箱连接；随后编译并运行程序。

在实验箱触摸屏上选择 pmsm 电机，设定合理的转速，并点击启动。（在实验中我们发现，刚启动的电机无法自己转动，需要用手拨一下）

在弹出菜单中单击 $Add\ Watch\ Expression$，添加需要观察的变量名称（如 pmsm, pmsm_set, PI_Speed, PI_Id, PI_Iq 等结构体），运行程序，实时刷新显示栏观察变量的变化情况。

(x)= run	unsigned int	1	0x0000C143@Data
(x)= start	unsigned int	0	0x0000C144@Data
(x)= stop	unsigned int	0	0x0000C145@Data
(x)= cw_ccw	unsigned int	1	0x0000C146@Data
(x)= control_logic	unsigned int	0	0x0000C147@Data

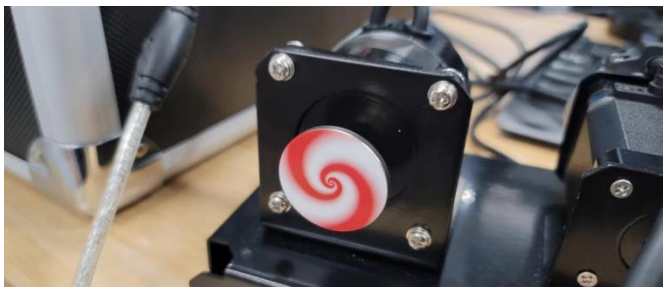
代码中的变量主要有两部分：电机运行参数和控制器参数。

a. 电机运行的实时数据都保存在 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（这三个 PI 控制器里面 ActualPoint 是测量值，SetPoint 是参考值，Kp,Ki 是 PI 控制器参数，另外只有 PI_Speed 的转速参考值 SetPoint 是可设置的）

将 $pmsm_set.start$置 1，电机启动，将 $pmsm_set.stop$置 1，电机停止。$speed$为电机实时转速，$pmsm_set.cw_ccw$为电机选择方向控制，默认为 1，电机逆时针旋转，将其设置为 0，则电机顺时针旋转。观察可以发现，当 $pmsm_set.start$置为 1 的时候，$pmsm_set.run$同时跳变为 1，伴随电机启动。

后续实验我们将会尝试调整其他变量的值，从而对电机 PID 控制进行更全面的理解。



1.5. 电机驱动的深度理解

这里，我们对应 1.2 中的理论分析，在实验中进一步理解电机的驱动原理。本电机由方波信号 pwm 进行驱动。我们将*pwm_duty*添加到*Expression*中，改变*pwm_duty*的值，发现电机转速会发生变化。

run	unsigned int	0	0x0000C143@Data
start	unsigned int	0	0x0000C144@Data
stop	unsigned int	0	0x0000C145@Data
cw_ccw	unsigned int	1	0x0000C146@Data
control_logic	unsigned int	0	0x0000C147@Data
bldcm.speed	unsigned int	321	0x0000C149@Data
pwm_duty	float	0.100000001	0x0000C102@Data
Add new expression			

run	unsigned int	1	0x0000C143@Data
start	unsigned int	0	0x0000C144@Data
stop	unsigned int	0	0x0000C145@Data
cw_ccw	unsigned int	1	0x0000C146@Data
control_logic	unsigned int	0	0x0000C147@Data
bldcm.speed	unsigned int	669	0x0000C149@Data
pwm_duty	float	0.200000003	0x0000C102@Data
Add new expression			

run	unsigned int	1	0x0000C143@Data
start	unsigned int	0	0x0000C144@Data
stop	unsigned int	0	0x0000C145@Data
cw_ccw	unsigned int	1	0x0000C146@Data
control_logic	unsigned int	0	0x0000C147@Data
bldcm.speed	unsigned int	1195	0x0000C149@Data
pwm_duty	float	0.5	0x0000C102@Data
Add new expression			

***pwm_duty*调节电机转速原理：**

pwm_duty 变量代表电机所需要方波信号（PWM 信号）的占空比。

占空比是指在一个周期内，信号处于高电平状态的时间与整个周期的时间之比。通常表示为百分比。例如，占空比为 0.5 意味着信号在一半的时间内处于高电平状态。

电机的转速取决于其接收的电压。占空比决定了电机接收到的平均电压。占空比越高，电机接收到的平均电压越高。

通过调节占空比，可以控制电机的平均电压，从而精确控制电机的转速。占空比增加，电机的转速通常会增加；占空比减少，电机的转速会降低。

2. 永磁同步电机控制实验

2.1. 实验目的

（1）理解和掌握永磁同步电机（PMSM）的控制原理和操作方法。

（2）学习如何使用 CCS（Code Composer Studio）开发环境进行电机控制程序的编写、调试和运行。

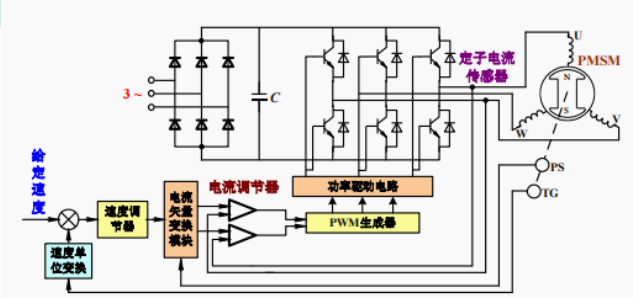
（3）通过实验观察和分析 PMSM 在不同控制参数下的性能，包括启动、运行、速度控制和负载响应。

（4）研究 PMSM 在空载条件下电压与转速的关系，并通过实验数据进行验证。探索在增加负载情况下，电机电流、转速和转矩的变化关系。

（5）学习 LED 灯状态控制的联合调试。

2.2. 实验原理 & 代码详解

2.2.1 电机结构



永磁同步电机的控制系统示意图如图所示。通过精确控制三相电流来实现对电机转速和转矩的控制，由方波信号 pwm 进行驱动，其占空比直接决定了电机的转速。实验中使用的 PMSM 接线包括强电线和位置传感器线，用于监测和控制电机的运行状态。

本次实验就是要编程实现这一控制系统，然后对系统和永磁同步电机本体进行性能分析。

2.2.2 代码详解

在实验之前详细阅读代码可以帮助我们更好地理解电机的控制原理。在 main 函数里面首先调用 *InitPeripherals*( ) 和 *InitVeriable*( ) 两个函数进行初始化。

```
38      InitPeripherals(); //Initialize all the Device Peripherals
39      InitVariables(); //Initialize Variables
```

InitPeripherals( )，用于初始化 ADC,DAC 等，这些参数在数模和模数转换中发挥作用，我们的实验不需要细究。*InitVeriable*( )，对一些需要用到的参数进行初始化，如 PID 参数。

在进行各个参数的初始化以后，就是要执行主循环了。程序一直运行 for(;;) 里面的代码执行条件判断：

```
1. if ((pmsm_set.start == 1) && (pmsm_set.run == 0)) // 启动电机条件判断
```

```
1. if (pmsm_set.start == 0) // 停止电机判断
```

电机控制运行在中断，中断是每个固定时间执行一次，应该是由实验箱控制的，执行过程中主程序暂停执行。

具体功能如下：

（1）

```
1. //GpioDataRegs.GPASET.bit.GPIO23 = 1; //LED_Run
```

这行代码被注释掉了，它原本的作用是设置 GPIO23 为高电平，用于点亮一个指示电机运行状态的 LED，会在 Task 6 中用到。

（2）

```
1. ADC_Ctrl(); //ADC 采样
2. pmsm.udc = ((float) Get_A_Adc(0)) * 0.00488281250;
3. pmsm.idc = ((float) Get_A_Adc(1)) * 0.00091552734375;
4. pmsm.iu = -((float) Get_A_Adc(2)) * 0.00152587890625 + 12.5;
5. pmsm.iv = -((float) Get_A_Adc(3)) * 0.00152587890625 + 12.5;
6. pmsm.iw = -((float) Get_A_Adc(4)) * 0.00152587890625 + 12.5;
```

从 ADC 获取直流电压、直流电流、三相电流的值，并进行转换和偏移处理。

（3）

```
1. u_global = sqrt(PI_Id.out * PI_Id.out + PI_Iq.out * PI_Iq.out);
```

计算全局电压 u_global，它是 d 轴和 q 轴电压的平方和的平方根。该变量会在 Task 4 中用到。

（4）

```
1. WriteDAC(DA_ADD0, pmsm.position * 0.1); // 试验板通道 1
2. WriteDAC(DA_ADD1, pmsm.udc * 0.1);
3. WriteDAC(DA_ADD2, pmsm.speed * 0.001);
4. WriteDAC(DA_ADD3, pmsm.iv * 1);
```

将电机状态值输出到 DAC 通道 1，2，3，4，用于示波器分析。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。因为在 CCS 中观察的值或波形（e.g.三相电流），由于采样频率的原因，无法获得三相正弦波形，需要把对应的值输出的 DAC 通道上，通过示波器才能观察到美观的正弦波形。下图中的四个通道分别对应从右至左 DA_ADD0-通道 1；DA_ADD1-通道 2；DA_ADD2-通道 3；DA_ADD3-通道 4。

2.3. 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 Windows 10 和 CCS 软件的 PC。

2.4. 实验步骤及结果展示

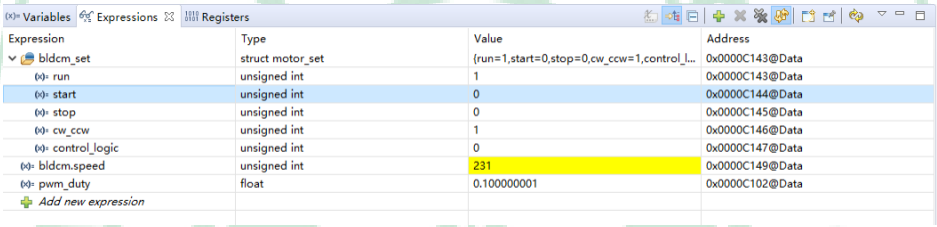
参考基于 CCS 仿真调试和程序下载的实验步骤，开启设备连接 CCS，打开 CCS。

（1）对实验设备进行硬件部分连接，连接好开发板的仿真器并上电（电机实验需打开 24V 电源开关），

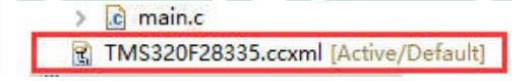
（2）按照工程导入步骤导入光盘资料里的*motor_openloop*工程，

（3）编译工程生成*motor_openloop. out*的可执行程序，

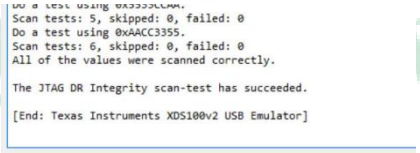
（4）下载程序，添加观察变量。右击变量*bldcm*和*bldcm_set*，在弹出菜单中单击 *Add Watch Expression*，随后运行程序



首先先进行基本的运行，测试设备连接情况，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，参考上图连接。



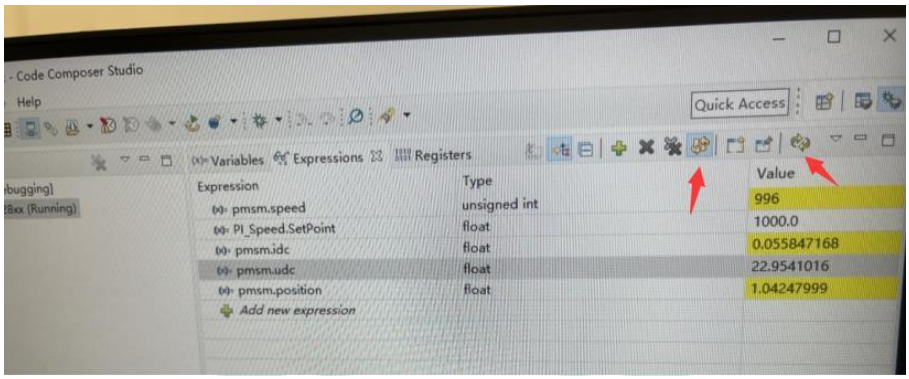
然后运行红框中的程序测试连接，当全部 success 以后就可以开始进行后续的实验了。



点击“小锤子”图标编译程序

点击“小瓢虫”图标下载永磁同步电机程序 TestBoxPMSM 并点击运行按钮

按照图示，按顺序选择 PMSM,点击启动按钮，设置转速，实现电机转动。然后将各个变量添加进 Expression 中进行观察如下图，注意要电机箭头所示的地方，开启“连续刷新”功能。



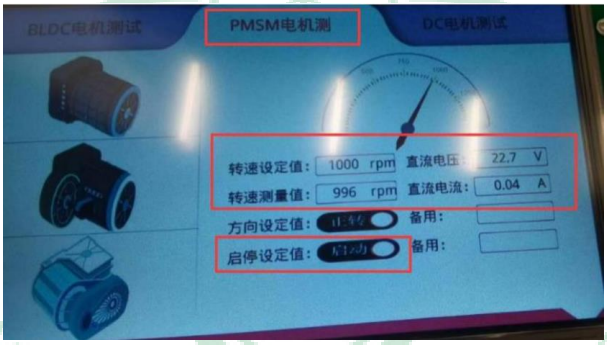
在 Expression 中选择变量，右击选择 Graph，就会出现上图，实时观察数据。

Task1

任务要求：启动电机，通过 CCS 的 Expression 和 Graph 观察电机运行，包括电机转速 speed，位置 position，三相电流 iu,iv,iw，直流电压 udc，idc 等参数（通过 CCS Expression 或屏幕 设置转速 PI_Speed 里面 SetPoint 参数的值，观察上述参数的值与波形）。记录并分析实验结果。

完成过程：

启动电机以后，就直接可以在电机的屏幕上面观察到基本的转速等数据：

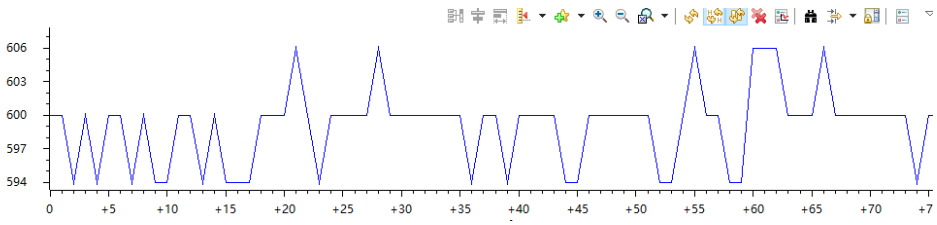


但是显示面板只能观察，不能记录，还是要在 Expression 中的 graph 中进行分析。我们在 CCS 的 Expression 界面加入 Speed、udc、idc 的值，并通过 graph 进行实时记录：

Expressions

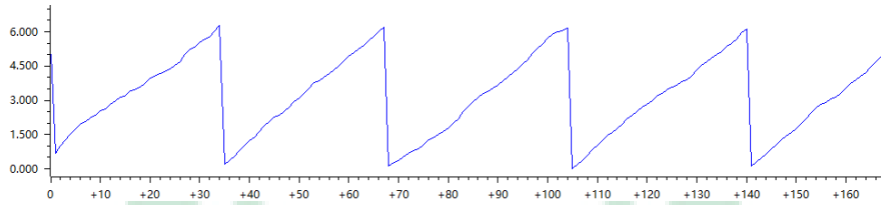
Expression	Type	Value	Address
pmsm.position	float	0.251199991	0x0000A496@Data
pmsm.speed	unsigned int	0	0x0000A498@Data
PI_Speed.SetPoint	float	599.0	0x0000A468@Data
pmsm.udc	float	23.7548828	0x0000A49A@Data
pmsm.idc	float	0.0567626953	0x0000A4A2@Data
Add new expression			

Speed 的变化：



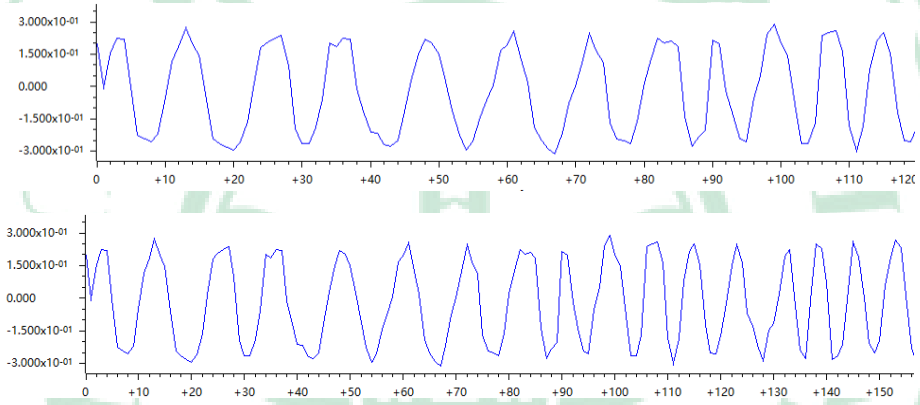
在实验过程中，我们将电机的设定转速调整为 600 转。通过观察记录的数据图表，可以明显看到实际运行时的转速在 594 至 606 转之间有轻微的波动，而这些波动的平均值紧密地围绕在 600 转附近。这表明转速的偏差处于一个可接受的误差范围内，从而证实了电机的运行状态十分稳定，且所采用的控制系统能够有效地维持预定的转速。

Position 的变化：

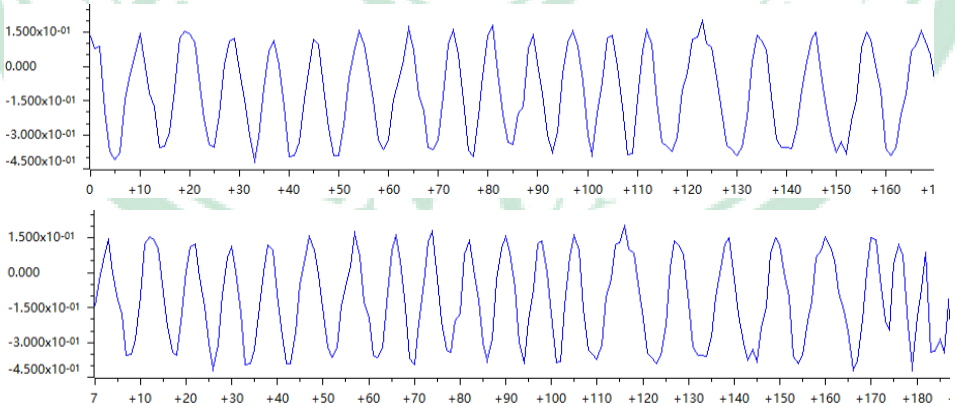


位置变量呈现出周期性的波动，其数值在 500 至 6500 的范围内变化。这些波动的波形表现出高度的一致性，表明系统在运行过程中保持了良好的稳定性和可预测性。这种规律性的波动模式进一步证实了系统的运行平稳，且在各个周期中均能维持相似的动态特性，反映出系统控制的精准性和可靠性。

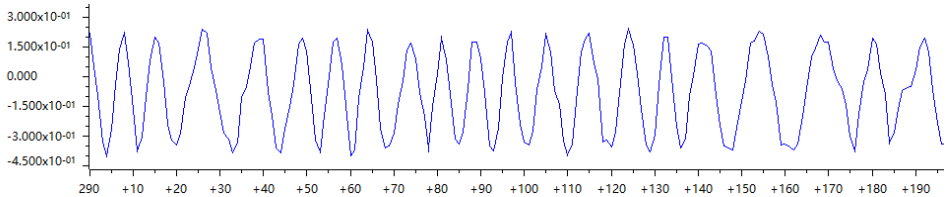
三相电流 Iu 的变化：

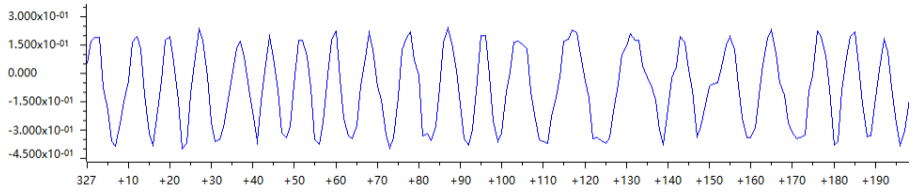


三相电流 Iv 的变化：



三相电流 Iw 的变化：





三相电流在实验中展现出典型的周期性波动特征，每一相电流均呈现出规律性的变化。值得注意的是，这三相电流之间保持着精确的相位差，具体为每相之间相差 120 度（即三分之一个圆周），这是三相系统正常运行的典型标志。然而，在观察中发现，Iv 和 Iw 两相电流的波动中心并非完美地对齐在零值基准线上，而是存在一定的偏差。这种偏差可能源于测量设备的精度限制或系统内部的微小不平衡，尽管如此，整体系统的运行仍保持稳定，且这种偏差对系统的整体性能影响有限。

Task2

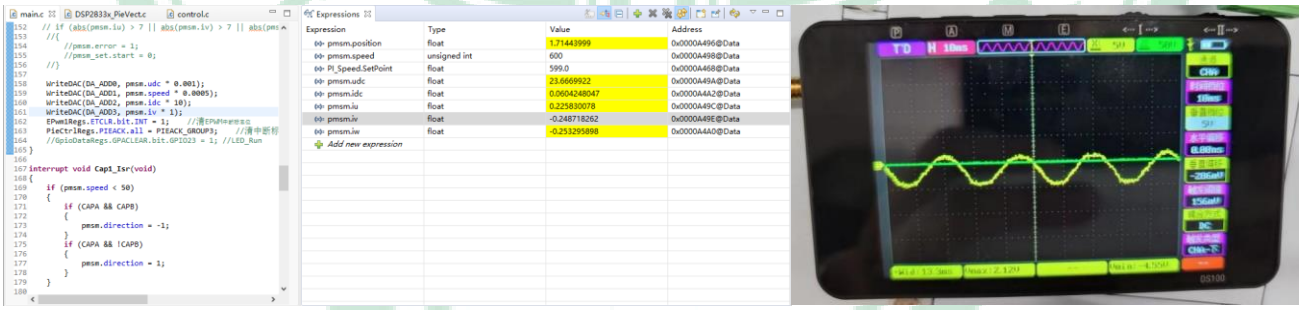
任务要求：通过示波器观察电机(pmsm 里面)的位置 position，转速 speed，电流(iu，iv，iw，idc)，电压等波形，这里需要将电机的位置，转速，电流，电压等波形输出到 DAC 通道（检查当前代码是否包含 WriteDAC 给 DAC 通道输出输出电流、位置等，如果没有需要自己添加相应的代码 WriteDAC(DA_ADDxx，xxxx)；xx（通道）与 xxxx（变量，e.g.位置，电流等）。

特别地，用示波器同时观察三相电流中的一相（e.g. iu）与位置 position 波形，分析位置一个周期是等于极对数*相电流周期。记录不同转速下的实验结果，并分析结果。”

完成步骤：

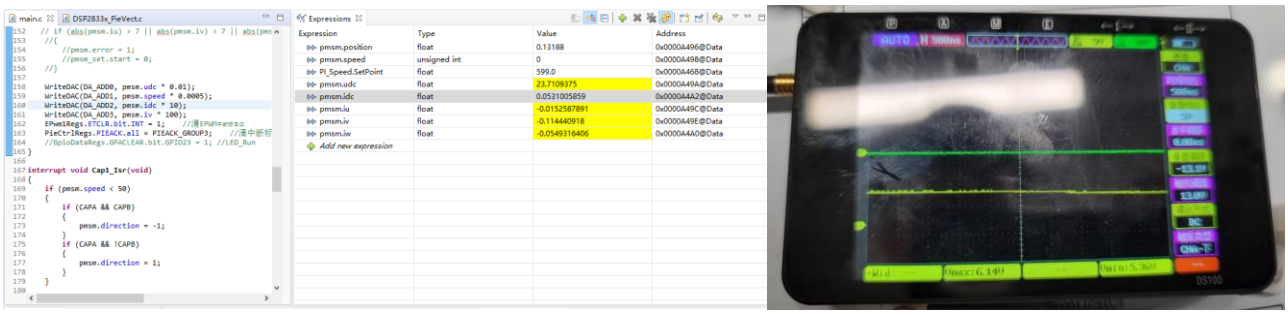
<1> **波形输出与观察：**逐一对各个变量进行输出进行分析，首先是三相电流 iv ，我们将其输出写入通道 3，并通过示波器可视化：

```
1.WriteDAC(DA_ADDR2, pmsm.iv * 1);
```



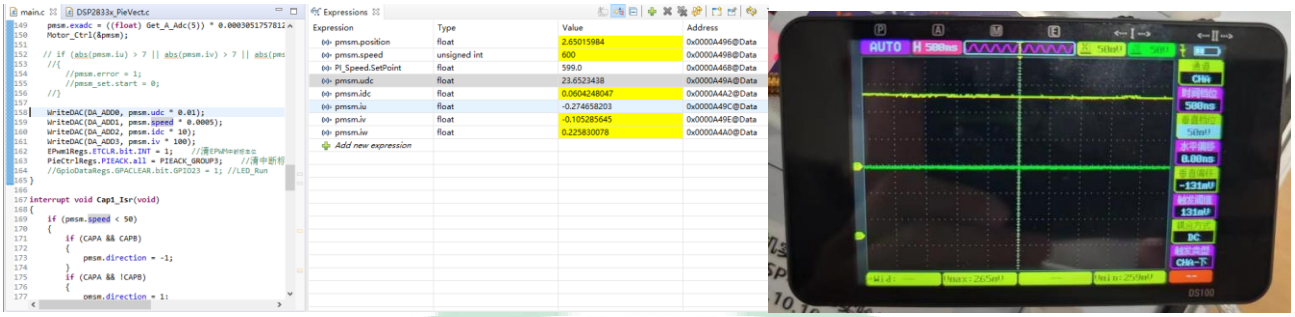
然后是 idc（直流电流）：、

```
1. WriteDAC(DA_ADDR3, pmsm.idc * 100);
```



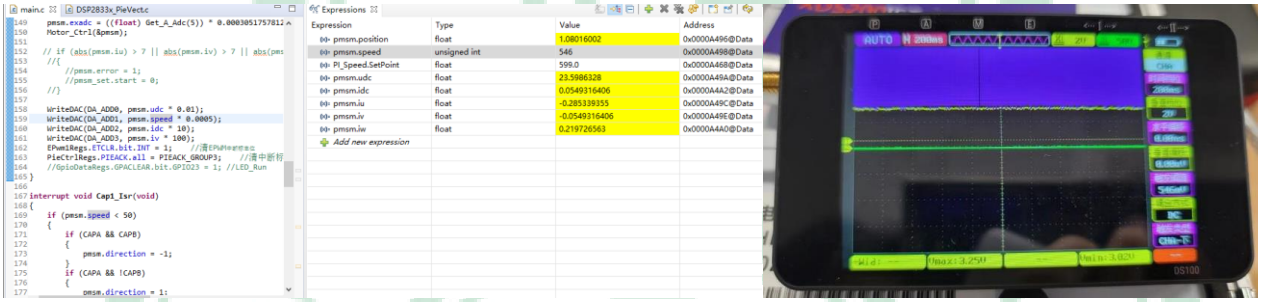
Udc（直流电压）：

```
1. WriteDAC(DA_ADDR3, pmsm.udc * 0.01);
```



Speed（转速实际值）

```
1. WriteDAC(DA_ADDR2, pmsm.speed * 0.005);
```



<2> 探究三相电机的周期与极对数、相电流周期的关系：

首先观察代码可知电机的极对数是 4

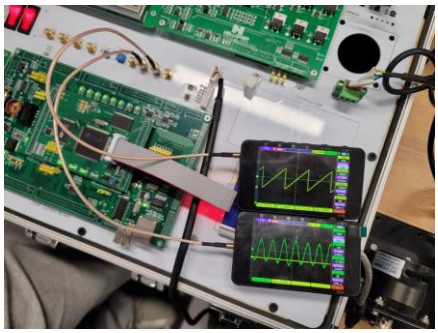
```
223     pmsm.idc = 0;
224     pmsm.pole = 4; //电机极对数
225     pmsm.speed = 0;
226     pmsm.udc = 0;
227     pmsm.direction = 1;
228     pmsm.error = 0;
```

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。

下面我们进行实验证明了我们的猜想：



用示波器同时观察三相电流中的一相（e.g. iu）与位置 position 波形，分析位置一个周期是等于极对数*相电流周期

Expression	Type	Value	Address
pmsm.position	float	0.270040005	0x0000A496@Data
pmsm.iu	float	0.296020508	0x0000A49C@Data
pmsm.idc	float	0.0531005859	0x0000A4A2@Data
Add new expression			

观察可以得出，位置一个周期确实是是等于极对数*相电流周期（66.5=16.7×4）。

Task3

任务要求：分别修改转速 PI_Speed 控制的比列与积分参数（PI_Speed kp, ki,修改范围不要太大，避免失控），通过 CCS Graph 观察对转速控制性能的影响（响应时间，超调，控制精度等,定性分析即可，即定性分析控制参数对上述控制性能的影响，无需定量的数据结果），即设置转速到不同的参考值（阶跃响应）。记录并分析实验结果，PI 参数对控制性能的影响。(不要修改主函数代码里面变量 PI_Speed 的 kp ki 的值，仅在 Expression 中修改 kp ki 的值，这样程序重启后不影响原来控制器参数与控制运行)

完成步骤：

在进行实验数据的分析过程中，我们采用了 matlab 软件来辅助处理和分析数据。首先，在图形界面（Graph）中，右键点击并选择“ExportData”功能，将所需的数据导出为 CSV 文件格式。这一步骤为我们后续的数据处理和分析打下了基础。

由于直接从图形界面上观察数据的细微变化存在一定的局限性，为了能够更精确地计算诸如超调量、调节时间等关键参数，我专门编写了一个 MATLAB 函数。该函数旨在对导出的 CSV 文件中的数据进行深入分析和处理。通过这一自动化的分析过程，我们能够获得更为准确和可靠的结果，从而为实验的评估和结论提供坚实的数据支持。这种方法不仅提高了分析的效率，也增强了结果的准确性和可信度。

以下为辅助分析函数 *Fun_Step_Performance* 代码，用于分析和计算给定时间序列和输出序列的标准阶跃响应的性能指标，包括稳态值、上升时间、调整时间和超调量，并根据需要绘制响应曲线。

```
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 %

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
inds = 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: '])
    else
```

```
51.         text(0.555*tm,0.8*ym,['峰值时间: ' num2str(tm) 's, 超调量: ' num2str(ov)
'%'])
52.     end
53.     text(0.7*t(end),0.95*ys,['稳态值: ' num2str(ys)])
54.     disp('%% 阶跃响应指标结果: ')
55.     disp(['上升时间: ' num2str(tr) 's'])
56.     disp(['调整时间: ' num2str(ts) 's'])
57.     disp(['峰值时间: ' num2str(tm) 's, 超调量: ' num2str(ov) '%'])
58.     disp(['稳态值: ' num2str(ys)])
59.     disp('%% 阶跃响应指标结果显示结束')
60. end
```

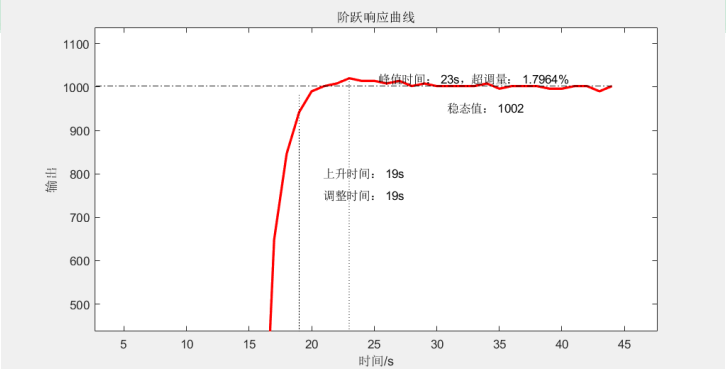
随后我们读取刚才导出的 cvs 文件，调用辅助函数，对系统性能进行分析：

```
1. % Load data from CSV file
2. filename = 'speed5.csv';
3. opts = detectImportOptions(filename);
4. opts.VariableNamingRule = 'preserve'; % Preserve the original variable name
5. data = readtable(filename, opts);
6.
7. % Extract data for 'Sample:float' and 'Data:float'
8. t = data.('Sample:float');
9. y = data.('Data:float');
10.
11. % Set the 'drawflag' parameter
12. drawflag = 1; % or 0 depending on whether you want to draw plots
13.
14. % Call the custom function
15. [ys, tr, ts, tm, ov] = Fun_Step_Performance(t, y, drawflag);
```

以下我们展示了 4 组不同的 PI 参数对模型性能的影响：

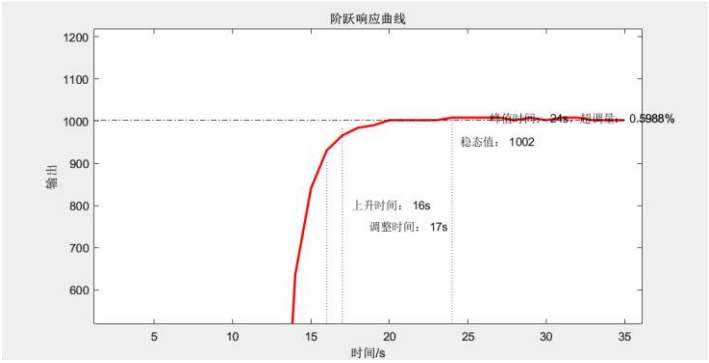
第一组数据

(x)= PI_Speed.Kp	float	0.0199999996	
(x)= PI_Speed.Ki	float	1.0	



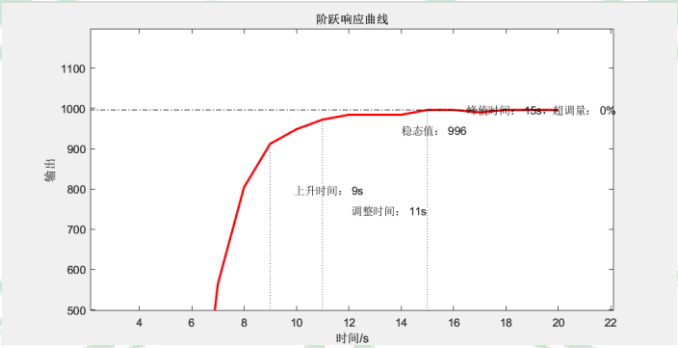
第二组数据

(x)= PI_Speed.Kp	float	0.0199999996	0
(x)= PI_Speed.Ki	float	0.5	0



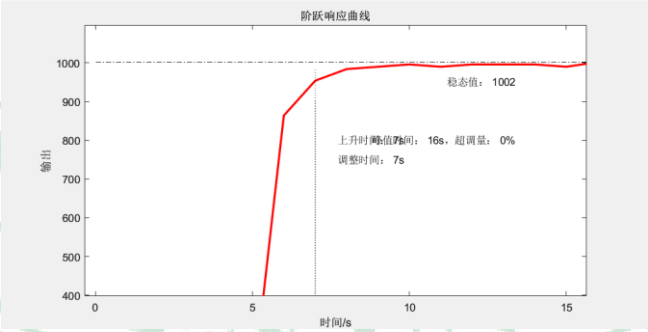
第三组数据

(x)= PI_Speed.Kp	float	0.0199999996
(x)= PI_Speed.Ki	float	0.200000003



第四组数据

(x)= PI_Speed.Kp	float	0.0500000007
(x)= PI_Speed.Ki	float	0.200000003



动态性能指标表格如下：

参数	稳态值	超调量	峰值时间	调节时间	上升时间
$K_p = 0.02;$ $K_i = 1.0$	1000	1.7%	23	19	19
$K_p = 0.02;$ $K_i = 0.5$	1000	0.5%	$24s$	$17s$	$16s$
$K_p = 0.02;$ $K_i = 0.2$	1000	0%	$15s$	$11s$	$9s$
$K_p = 0.05;$ $K_i = 0.2$	1000	0%	$16s$	$7s$	$7s$

随后我们对以上结果进行了原理分析：在 PI 控制中，比例和积分环节都对系统性能产生着一定的影响。

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 Ti 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 Kp 使得系统更快地调节，减小稳态误差；较小的 Ki 防止系统超调增大，甚至振荡。

Task4

任务要求：研究空载时电压与转速的关系（电压为$uglobal = sqrt(ud\,2 + uq\,2)$，需要定义一个全局变量$u_global$，$ud$来自$PI_id$，$uq$来自$PI_iq$），设置不同转速，记录数据并画出曲线，其中$ud$为$PI_Id$的输出，$uq$为$PI_Iq$的输出，转速为传感器测量,（最新程序已经添加代码计算u_global）。通过 CCS Expression 记录不同转速$speed$对应的电压u_global，离线画出关系曲线并分析规律。

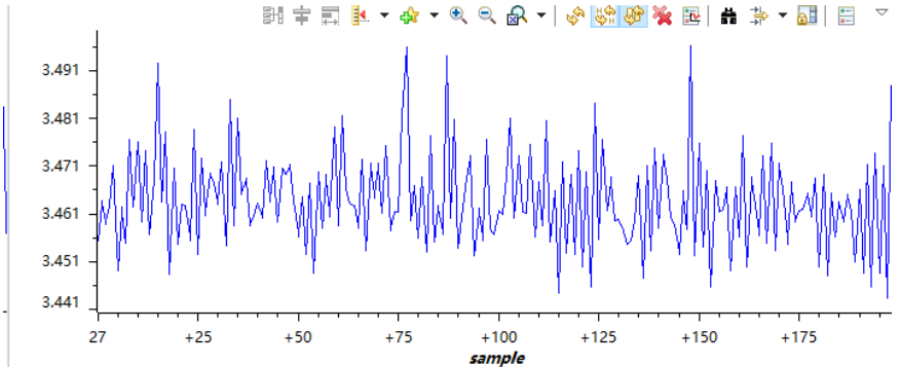
完成步骤： 首先就是要添加定义u_global ：

```
1. float u_global = 0;
```

然后再$EPWM1_Isr()$函数中新增这样一句计算u_global的指令

```
1. u_global = sqrt(PI_Id.out * PI_Id.out + PI_Iq.out * PI_Iq.out);
```

利用 CSS 观察不同转速下的u_global曲线，结果如下（转速设定为 1000 时候的 u_global）：



通过不断修改转速 speed，并且观察 u 的变化，记录数据如下：

Expressions

Expression	Type	Value
pmsm.position	float	1.18691993
pmsm.speed	unsigned int	996
PI_Speed.SetPoint	float	1000.0
pmsm.udc	float	22.8173828
pmsm.idc	float	0.0421142578
pmsm.iv	float	-0.192260742
pmsm.iu	float	0.411987305
pmsm.iw	float	-0.234985352
PI_Speed.Kp	float	0.00400000019
PI_Speed.Ki	float	1.0
u_global	float	3.7188158

Expression	Type	Value
pmsm.position	float	4.50275993
pmsm.speed	unsigned int	900
PI_Speed.SetPoint	float	900.0
pmsm.udc	float	22.7392578
pmsm.idc	float	0.0476074219
pmsm.iv	float	-0.282287598
pmsm.iu	float	0.228881836
pmsm.iw	float	0.108337402
PI_Speed.Kp	float	0.00400000019
PI_Speed.Ki	float	1.0
u_global	float	3.37391472

Expression	Type	Value
pmsm.position	float	3.20907998
pmsm.speed	unsigned int	792
PI_Speed.SetPoint	float	800.0
pmsm.udc	float	22.8173828
pmsm.idc	float	0.0485229492
pmsm.iv	float	-0.120544434
pmsm.iu	float	0.36315918
pmsm.iw	float	-0.219726563
PI_Speed.Kp	float	0.00400000019
PI_Speed.Ki	float	1.0
u_global	float	3.03117967

Expression	Type	Value
pmsm.position	float	4.97375965
pmsm.speed	unsigned int	1104
PI_Speed.SetPoint	float	1100.0
pmsm.udc	float	22.7001953
pmsm.idc	float	0.0393676758
pmsm.iv	float	0.463867188
pmsm.iu	float	-0.311279297
pmsm.iw	float	0.0854492188
PI_Speed.Kp	float	0.00400000019
PI_Speed.Ki	float	1.0
u_global	float	4.06872702

Expression	Type	Value
pmsm.position	float	1.53232002
pmsm.speed	unsigned int	1206
PI_Speed.SetPoint	float	1200.0
pmsm.udc	float	22.6806641
pmsm.idc	float	0.0503540039
pmsm.iv	float	-0.186157227
pmsm.iu	float	0.518798828
pmsm.iw	float	-0.401306152
PI_Speed.Kp	float	0.00400000019
PI_Speed.Ki	float	1.0
u_global	float	4.40641737

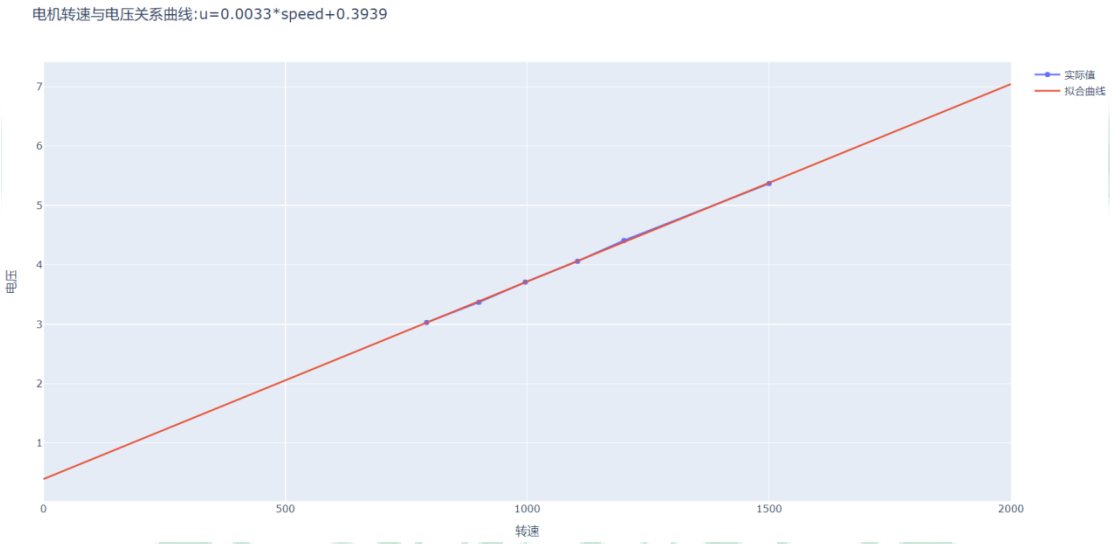
Expression	Type	Value
pmsm.position	float	3.91243982
pmsm.speed	unsigned int	1482
PI_Speed.SetPoint	float	1500.0
pmsm.udc	float	22.7392578
pmsm.idc	float	0.0631713867
pmsm.iv	float	0.12512207
pmsm.iu	float	-0.198364258
pmsm.iw	float	0.566101074
PI_Speed.Kp	float	0.00400000019
PI_Speed.Ki	float	1.0
u_global	float	5.37809086

观察数据，我们认为转速与电压呈一次函数关系，考虑使用 python 进行分析数据，得 u 与 speed 之间的规律，代码如下：

调用了 plotly 这个库进行拟合，需要先执行：*pip install plotly* == 5.18.0

```python
# 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 = [792, 900, 996, 1104, 1200, 1500]
# 对应的电压 u
u = [3.03, 3.37, 3.71, 4.06, 4.41, 5.37]
# 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],
xaxis_title='转速', yaxis_title='电压')
fig.show()
```

拟合效果如下，可以明显地看出一次函数关系和拟合的一次函数曲线。



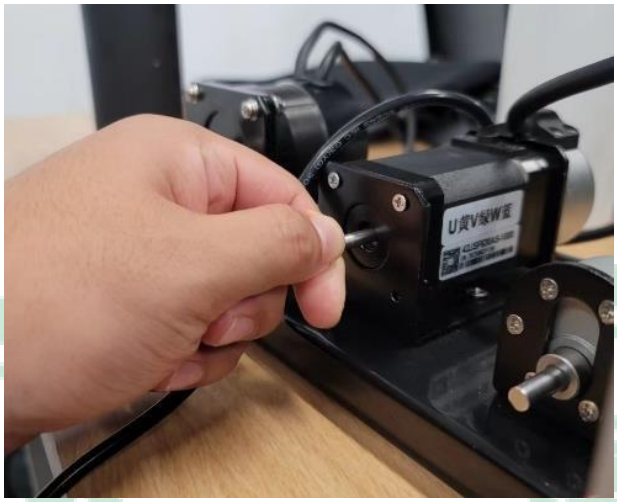
Task5

任务要求：尝试添加负载（手动，放在电机轴上增加阻力），通过 CCS 观察观察电流 PI_Iq 里面的 iq(ActualPoint)、转速 speed、转矩(与 iq 正比，即观察 iq)的变化，并记录实验现象（其中 Te=K1*flux*Iq 与 iq 正比，voltage=K2*flux*speed 与电压正比，观察 iq 等价于观察 Te)；观察并分析 q 轴电流与施加负载的关系（PI_iq 中的 iq（ActualPoint））。（提示：增加负载，由运动学模型知，加速度 dω/dt 小于零，导致转速降低）

完成步骤：这个实验是要探究负载对电机的一些影响，在正常启动电机以后。

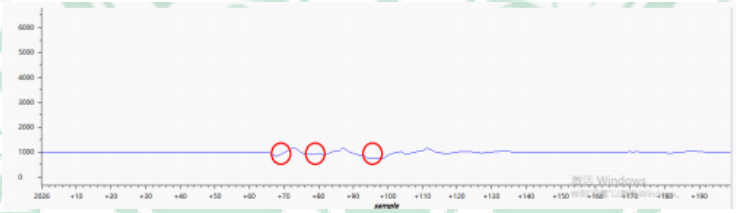
尝试添加负载，观察电流、转速、转矩的变化，并记录实验现象（Te=K1*flux*Iq,voltage=K2*flux*speed）；观察 q 轴电流与施加负载的关系。

<1> 添加负载，通过捏住电机来添加负载，使得转矩产生变化的方式进行。

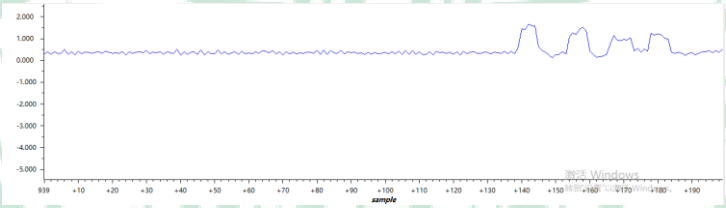


<2> 观察实验现象，则是通过 Expression 中的 graph 功能。

Speed：可以观察到转速图中，原本稳定在 1000 左右的值出现突变。



观察下面的电流变化图，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。



可以分析得到，当电机的负载增加以后，q 轴电流也会相应增加。

这是由于电机在负载增加时需要更多的转矩来维持其转速。在永磁同步电机的向量控制中，q 轴电流（直轴电流）直接与电机的转矩输出相关联。

永磁同步电机的控制通常基于 d-q 轴理论，其中 d 轴和 q 轴是在同步旋转参考框架下的两个正交轴。在这个框架下，d 轴电流主要影响电机的磁通，而 q 轴电流则与产生转矩相关。当电机的负载增加时，为了产生更大的转矩以克服负载，电机控制系统会增加 q 轴电流。

由于负载增加，电机可能需要更多的能量来维持相同的转速和性能，这通常表现为 q 轴电流的增加。这种增加有助于增强电机的磁场，从而产生所需的额外转矩。然而，这也可能导致电机效率的降低和热量的增加，这是在设计和操作永磁同步电机时需要考虑的重要因素。

Task6

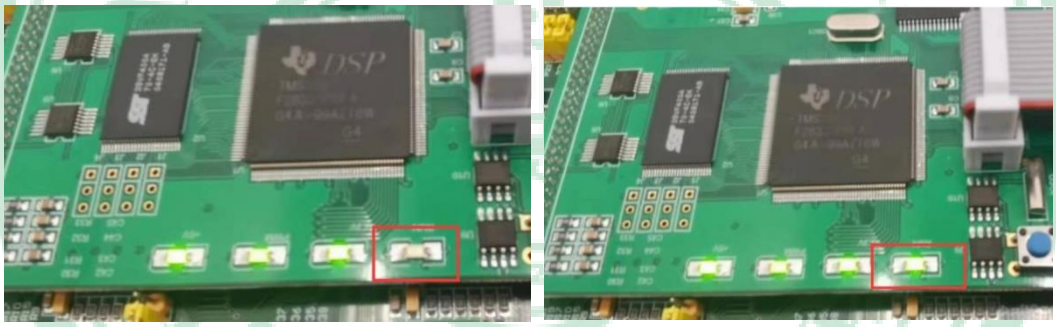
任务要求：尝试设置一些条件点亮 LED 灯（如电机启动灯亮，超过一定转速灯闪等 GpioDataRegs.GPATOGGLE.bit.GPIO23 对应的 LED 可以用），记录实验结果，并分析结果（选做，附加题）（提示：LED 控制程序请加在主函数 main（）的 for/while 循环中，不要添加到中断程序中）

完成步骤：在主函数中利用一个计数变量 counter 控制灯泡亮暗，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;

LED 亮灭逻辑下图代码中所示：

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

运行效果如图：



观察可知，成功实现了随着程序运行不断亮灭的功能。

3. 实验心得

通过本次关于永磁同步电机（PMSM）的控制实验，我获得了宝贵的实践经验和深刻的理论认识。实验不仅加深了我对电机驱动原理的理解，还提高了我使用 CCS（Code Composer Studio）开发环境进行程序编写、调试和运行的能力。

实验前，我对电机的控制原理有了一定的理论了解，但通过实际操作，我更加直观地感受到了理论在实际中的应用。例如，通过调整 PI 控制器的参数，我观察到了电机性能的显著变化，这让我深刻认识到了理论对于实践的指导意义以及参数调整对于系统性能的重要影响。

在实验过程中，我学会了如何使用仿真器和 CCS 软件进行电机编程，这些工具对于电机控制系统的开发至关重要。我通过不断尝试和调整，逐渐掌握了如何

通过软件来监控和调整电机的运行状态。此外，我也提高了我的编程能力，特别是在处理数据和实现控制算法方面。

通过对实验数据的记录和分析，我锻炼了自己的数据分析能力。使用 MATLAB 进行数据处理和图形化展示，让我更加直观地理解了电机运行的特性。例如，通过分析转速与电压的关系，我验证了理论预期，并发现了一些有趣的现象，如负载变化对电流和转矩的影响。

实验中遇到了不少挑战，如电机无法自行启动、参数调整导致的系统不稳定等。面对这些问题，我学会了如何通过查阅资料、分析数据和调整代码来解决。这个过程提高了我的问题解决能力，也让我意识到了在工程实践中遇到问题并寻找解决方案的重要性。

总的来说，这次实验是一次非常宝贵的学习经历。我不仅加深了对永磁同步电机控制的理解，还提升了自己的实践操作能力和问题解决能力。我相信这些经验和技能将在我的未来学习和工作中发挥重要作用。