

电机 PWM 控制与驱动电路实验指导书



哈尔滨工业大学（深圳）
实验与创新实践教育中心

目 录

1. 实验目的	1
2. 实验原理	1
2.1 直流无刷电机六步换相	1
2.1.1 换向逻辑分析	1
2.1.2 程序设计说明	5
2.2 交流伺服电机 Clark/Park 变换	7
2.2.1 Clark/Park 变换思想	7
2.2.2 程序设计说明	10
3. 实验步骤	12
3.1 直流无刷电机六步换相实验步骤	12
3.1.1 参考实验设计部分	12
3.1.2 六步换向操作步骤	14
3.2 交流伺服电机 Clark/Park 变换实验步骤	21
3.2.1 参考实验设计部分	21
3.2.2 Clark 和 Park 变换操作步骤	23
4. 实验分析	31
附录 1 系统硬件介绍	32
1.1 系统构成	32
1.2 DSP 核心板	33
1.2.1 接线定义	33
1.2.2 参数	34
1.2.3 功能特点	34
1.3 直流无刷电机	36
1.3.1 接线定义	36
1.3.2 参数	36
1.3.3 结构特点	37
1.4 直流无刷电机	38
1.4.1 接线定义	38
1.4.2 参数	38
1.4.3 结构特点	39
附录 A CCS 工程导入	40
附录 B 上位机实时内核设置	47

1. 实验目的

- (1) 掌握直流无刷电机的控制原理，理解六步换相法；
- (2) 掌握交流伺服电机的控制原理，理解 clark 变换和 park 变换；
- (3) 了解 Dsp 微控制器 CCS 工程创建导入、编译、信号传输通道连接的过程；

2. 实验原理

2.1 直流无刷电机六步换相

2.1.1 换向逻辑分析

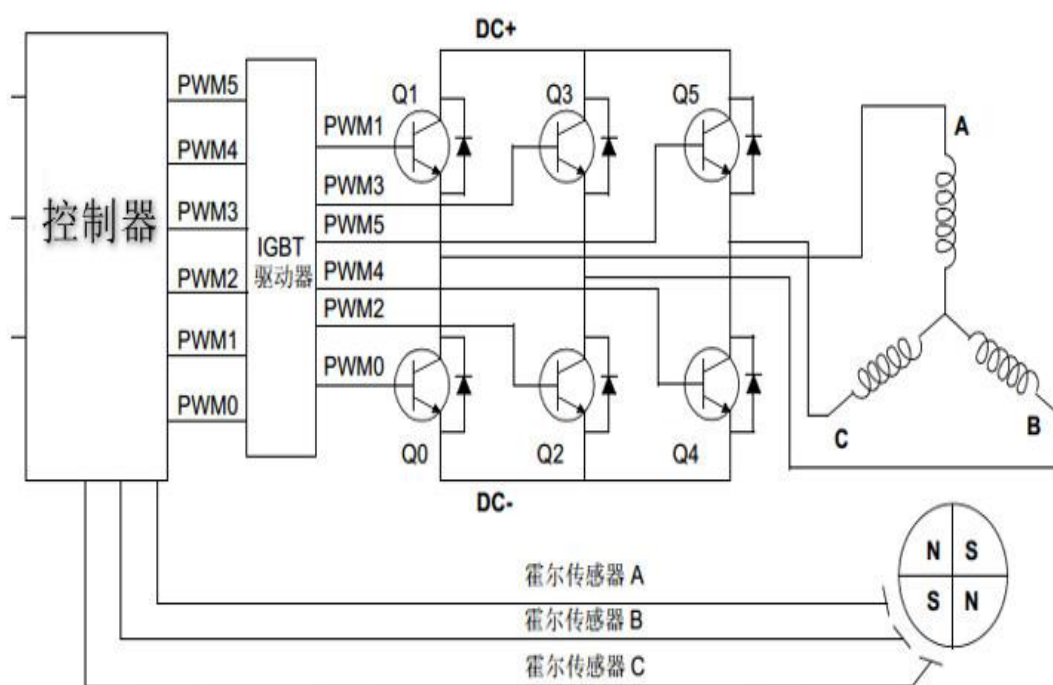


图 2-1-1 有霍尔传感器的 BLDC 控制框图

本控制系统的构成包括：控制器（单片机、DSP、PLC 等）、IGBT 驱动器、桥式电路（全桥和半桥）、无刷直流电机、霍尔传感器等。根据无刷直流电机转子磁极的位置，对定子线圈进行换相通电，形成 6 步的旋转磁场，进而带动转子同步转动。

有传感器的控制方式的关键技术在于：一、如何通过安装在电机上的三个霍尔器件 A、B、C 来准确快速的获取转子磁极位置信息；二、如何通过 6 个功率器件组成的 3 相桥式电路来控制定子线圈的 6 拍通电方式，从而形成一个旋转磁场。6 拍定子线圈方式如图 2-1-2 所示。

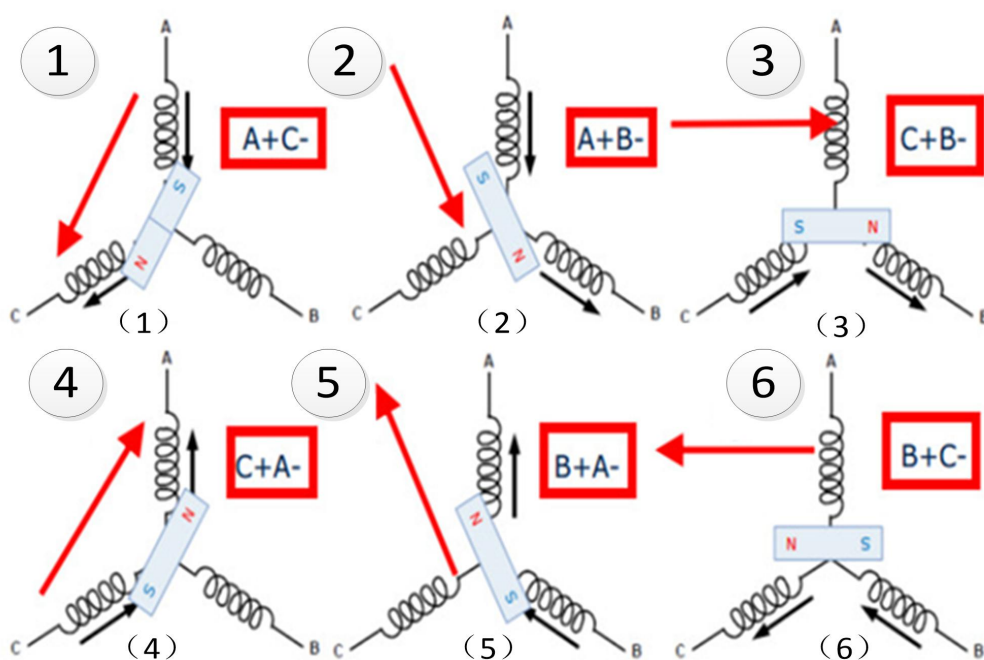


图 2-1-2 6 拍定子线圈通电方式示意图（逆时针旋转）

按照上图 2-1-2 的方式通电时，三相绕组线圈形成的合成磁场会慢慢的逆时针转动，形成一个旋转磁场，从而导致转子的运动。

此外需要说明的是，控制框图里的霍尔传感器是根据霍尔效应制作的一种磁场传感器，它可以有效的反映通过霍尔原件的磁密度，有正向磁场通过霍尔传感器，输出 1，有反向磁场通过霍尔，输出 0。当霍尔传感器在和电机的转子做相对运动时，根据转子下磁场密度的变化，来产生变化的信号。如下图 2-1-3 所示。

如果将一只霍尔传感器安装在靠近转子的位置，当 N 极逐渐靠近霍尔传感器即磁传感器达到一定值时，其输出是导通状态；当 N 极逐渐离开霍尔传感器、磁感应逐渐小时，其输出仍然保持导通状态；只有磁场转变为 S 极并达到一定值时，其输出才翻转为截止状态。在 S 和 N 交替变化下传感器输出波形占高、低电平各占 50%。如果转子是一对极，则电机旋转一周霍尔传感器输出一个周期的电压波形。

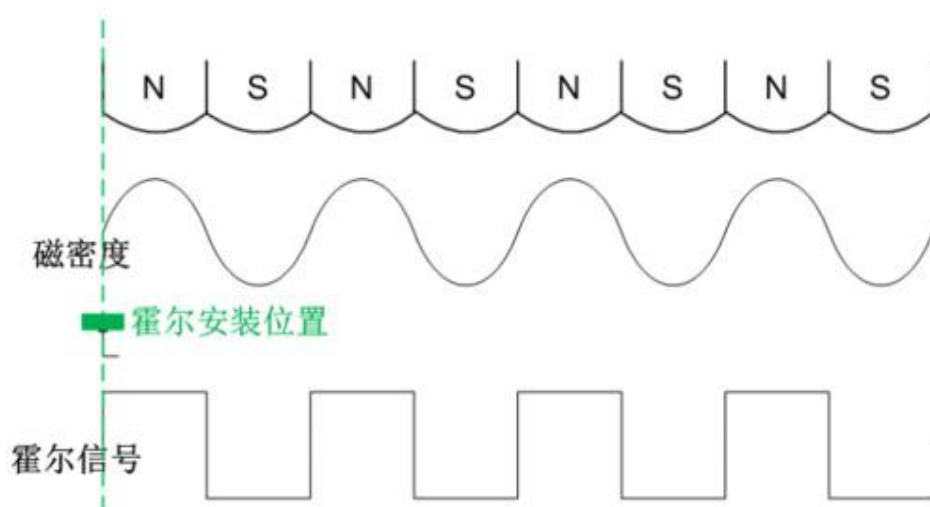


图 2-1-3 霍尔信号变化

当电机按一个（正转或反转）方向转动时，3 个霍尔传感器的输出会按照 6 步的规律变化，如图 2-1-4 所示。BLDC 电机的转子位置检测器件采用的是霍尔传感器，其与微控制器(MCU)连接非常简单。可以使用任一可用的具有电平变化中断功能的 MCU 的引脚作为其接口。

为了驱动电机运转，必须根据电机转子所处的空间位置，按照一定的换向顺序，对定子上的 A、B、C 三个线圈的两相同时进行通电，形成旋转磁场，从而带动转子旋转。一般在其非驱动端上的定子中安装三个霍尔传感器。这样当转子磁极经过霍尔传感器附近时，它们便会发出一个高(低)电平信号，根据这三个霍尔传感器信号的组合，就能得到定子绕组通电换向的精确顺序。

对于典型的三相带传感器的 BLDC 电机，有 6 个不同的工作区间，每个区间中对特定的两相绕组通电。每次换向，都有一个绕组连到电源的正极(电流进入绕组)，第二个绕组连到负极(电流从中流出)，第三个处于失电状态。转矩是由定子线圈产生的磁场和转子之间相互作用产生的。其转矩峰值出现在两个磁场正交时，而在两磁场平行时最弱。为了保持电机转动，由定子绕组产生的磁场应不断变换位置，因为转子会向着与定子磁场平行的方向旋转。BLDC 电机其三个定子绕组每换向一轮(1 个电周期)需要六步，应按照“六步换向”给定子绕组加电。

转子每转过 60° 电角度，定子上其中一个霍尔传感器就会改变状态，因此，完成电周期需要六步。在同步模式下，每转过 60° 电角度相电流切换一次。（一个电周期可能并不对应于完整的转子机械转动周期，完成一圈机械转动要重复的电周期数取决于转子磁极的对数。）每对转子磁极换向需要完成一个电周期。因此，转子的转数=电周期数/转子磁极对数，即提前设定好输出电流或电压信号的电周期数，就能得到电机转子旋转的转数，实现位置控制。

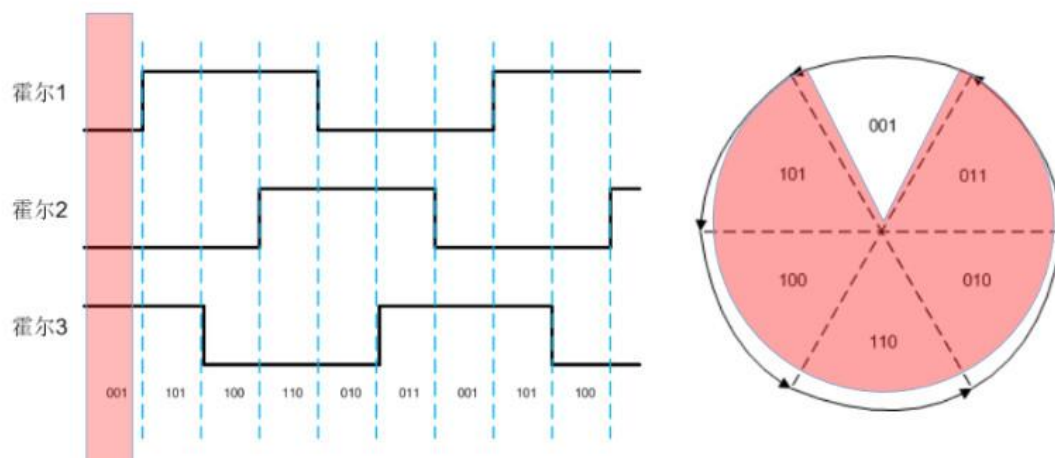


图 2-1-4 霍尔传感器输出规律变化原理图

电机驱动电路如图 2-1-5 所示。

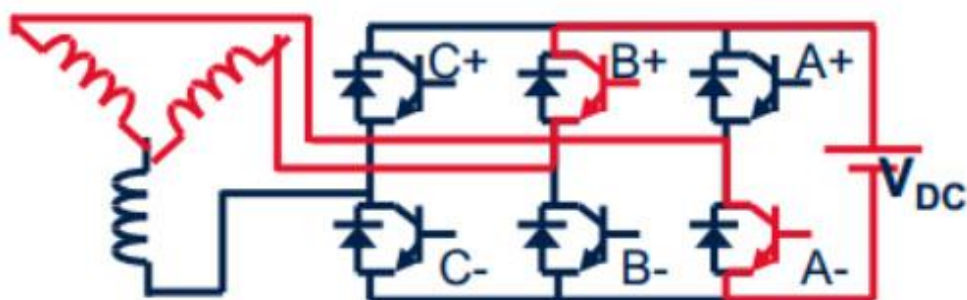


图 2-1-5 驱动电路示意图

以上为电机转子逆时针旋转控制方法，作为已知，见表 2-1-1。如需控制电机转子顺时针旋转，请参考表 2-1-2。

表 2-1-1 (MOTOR_DIR==0) 逆时针旋转控制规律

霍尔传感器输入			相电压			PWM 信号状态
1	2	3	A	B	C	pwm1.CmtnPointer
0	1	0	DC+	关	DC-	0
0	1	1	DC+	DC-	关	5
0	0	1	关	DC-	DC+	4
1	0	1	DC-	关	DC+	3
1	0	0	DC-	DC+	关	2
1	1	0	关	DC+	DC-	1

表 2-1-2 (MOTOR_DIR==1) 顺时针旋转控制规律

霍尔传感器输入			相电压			PWM 信号状态
1	2	3	A	B	C	pwm1.CmtnPointer
1	0	1	DC+	关	DC-	0
0	0	1	关	DC+	DC-	1
0	1	1	DC-	DC+	关	2
0	1	0	DC-	关	DC+	3
1	1	0	关	DC-	DC+	4
1	0	0	DC+	DC-	关	5

2.1.2 程序设计说明

如下图 2-1-6 所示，此为实验上位机 Matlab 中的程序。

程序设计思想：BLDC 电机采用模拟启动，采用一个递减的斜坡模块控制脉冲模块的周期，使频率递增，输入到模拟霍尔信号发生模块，产生一个频率逐渐递增的霍尔信号输入到六步换相模块，电机开始转动。电机模拟启动后，霍尔传感器模块产生一个标志位，切换到真实霍尔传感器输出。

模式 1 上位机给定参考的电流 REF，使用 rmp1 模块产生一个斜坡到参考的目标值，rmp1 模块产生一个标志位切换到电流 PID 闭环。

模式 2 上位机给定参考的速度 REF，使用 rmp1 模块产生一个斜坡到参考的目标值，rmp1 模块产生一个标志位切换到电流速度 PID 闭环。

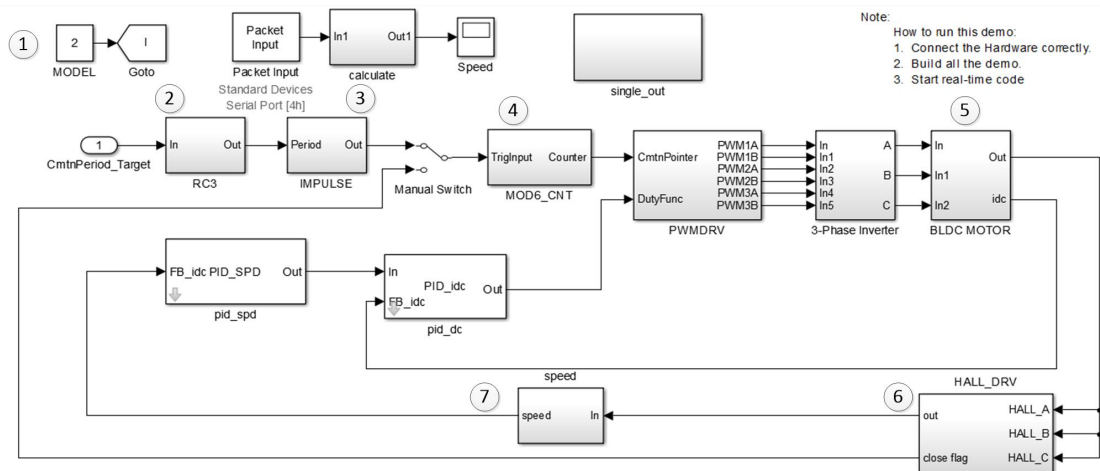


图 2-1-6 直流无刷电机 Matlab 控制程序界面

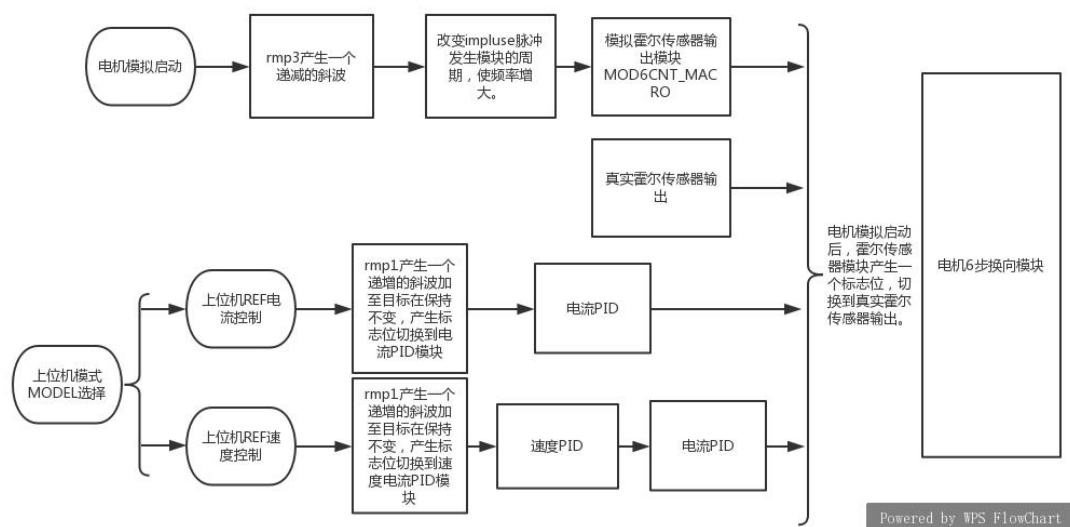


图 2-1-7 程序设计流程图

①模式选择代码块，模式 1 为电流环控制，模式 2 是速度环控制。

```
switch (model)
{
case 1:
    SpeedRef=ref_idc;//电流环控制
    break;
case 2:
    SpeedRef=ref_spd;//速度环控制
    break;
}
```

图 2-1-8 模式选择 CCS 程序

②RC3 斜坡代码块，控制 PWM 波形的占空比，当占空比越来越小时，电机速度加快。

Connect inputs of the RMP3 module and call the Ramp control 3 macro. 下降斜坡

```
rmp3.DesiredInput = CmtnPeriodTarget;
rmp3.Ramp3Delay = RampDelay;
RC3_MACRO(rmp3)
```

图 2-1-9 PC3 斜坡 CCS 程序

③脉冲代码块，是输出一个频率可调的脉冲给 MOD6CNT_MACRO 模块。

Connect inputs of the IMPULSE module and call the Impulse macro. 脉冲

```
impl1.Period = rmp3.Out;
IMPULSE_MACRO(impl1)
```

图 2-1-10 脉冲 CCS 程序

④模拟信号代码块，模拟六步换相信号。


```

#define MOD6CNT_MACRO(v)
{
    if (v.TrigInput > 0)
    {
        if (v.Counter == 0) /* Reset the counter when it is 5 */
            v.Counter = 5;
        else
            v.Counter--; /* Otherwise, increment by 1 */
    }
}

```

图 2-1-11 模拟信号 CCS 程序

⑤采样电流代码块，采集电机电流，做电流闭环。

```

if (pwm1.CmntPointer == 0)
DCbus_current = -(_IQ15toIQ((AdcResult.ADCRESULT1 << 3) - IDC_offset) << 1);

```

图 2-1-12 电机采样电流 CCS 程序

⑥信号输出，采集霍尔传感器信号。

```

#define HALL3_READ_MACRO(v)
v.CmntTrigHall = 0; /* Reset trigger, it only handshakes with calling program. */
if (v.EdgeDebounced == 0) /* Debounce current position. */
{
    HALL3_DEBOUNCE_MACRO(v)
    v.CmntTrigHall = v.EdgeDebounced; /* Set Commutation trigger here */
} else /* If current position is debounced, find match in table */
    HALL3_NEXT_STATE_MACRO(v) /* and return pointer to current state. Ptr
                                to be incremented */
                                /* by MOD6CNT after RET. */
v.EdgeDebounced = 0; /* Reset trigger */

```

图 2-1-13 信号输出 CCS 程序

⑦计算电机速度。

```

#if MOTOR_DIR == 0
    if ((pwm1.CmntPointer == 2) && (PreviousState == 3) &&
        (hall1.CmntTrigHall == 0x7FFF) / mod_dir1.TrigInput hall1.CmntTrigHall == 0x7FFF
        && (mod1.TrigInput) 5 4
    {
        //a=400;
        speed1.TimeStamp = VirtualTimer;
        SPEED_PR_MACRO(speed1)
    }
}

```

图 2-1-14 电机速度计算

2.2 交流伺服电机 Clark/Park 变换

2.2.1 Clark/Park 变换思想

在伺服系统中，直流伺服电机能获得优良的动态与静态性能，其根本原因是被控制量只有电机磁通 Φ 和电枢电流 I_a ，且这两个量是独立的。此外，电磁转矩

($T_m = K_T \Phi I_a$) 与磁通 Φ 和电枢电流 I_a 分别成正比关系。因此，可做线性控制。如果能够模拟直流电机，求出交流电机与之对应的磁场与电枢电流，独立地加以控制，即可使交流电机具有与直流电机近似的优良特性。为此，必须将三相交变量（矢量）转换为与之等效的直流量（标量），建立起交流电机的等效模型，然后按直流电机的控制方法对其进行控制。

在使用矢量控制方法控制电机时，需建立交流伺服电机的等效模型，使用 Clark 和 Park 变换对矢量进行变换并加以控制。涉及到交流伺服电机中的磁场分布如图 2-2-1 所示。交流伺服电机的磁场方向为正弦磁场。由于 U、V、W 三相的电流分别为 i_a 、 i_b 、 i_c 。

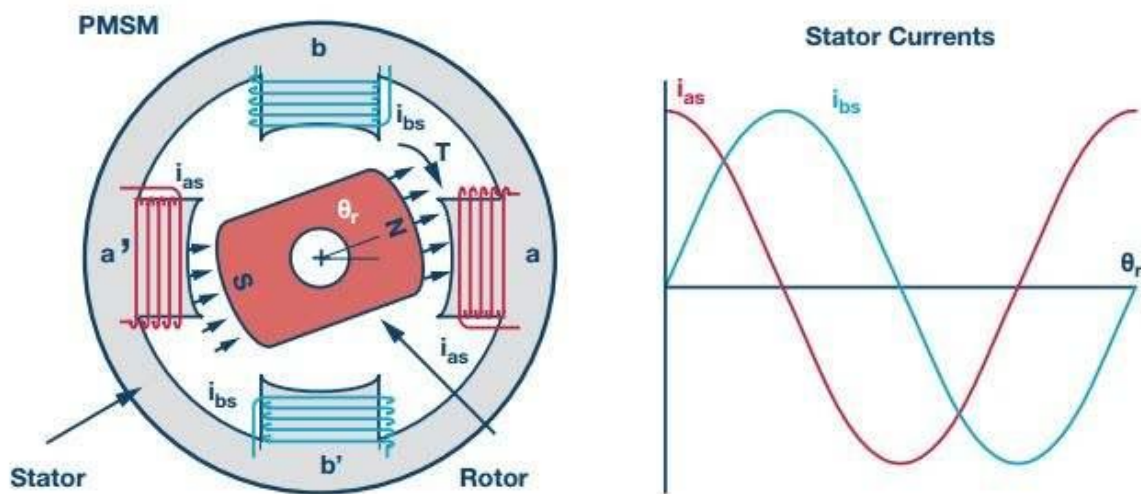


图 2-2-1 交流伺服电机的磁场

又因为在交流伺服电机中，电流的幅值相等，相位角各相差 120° 。所以可得电机在动态的时候各相电流表达式：

$$i_a = i_m \cdot \sin(\omega t) \quad (2-2-1)$$

$$i_b = i_m \cdot \sin\left(\omega t + \frac{2\pi}{3}\right) \quad (2-2-2)$$

$$i_c = i_m \cdot \sin\left(\omega t + \frac{4\pi}{3}\right) \quad (2-2-3)$$

其中， ω 表示电流角频率， i_m 表示电流幅值。这三个电流在空间的矢量分布如图 2-2-2 所示。

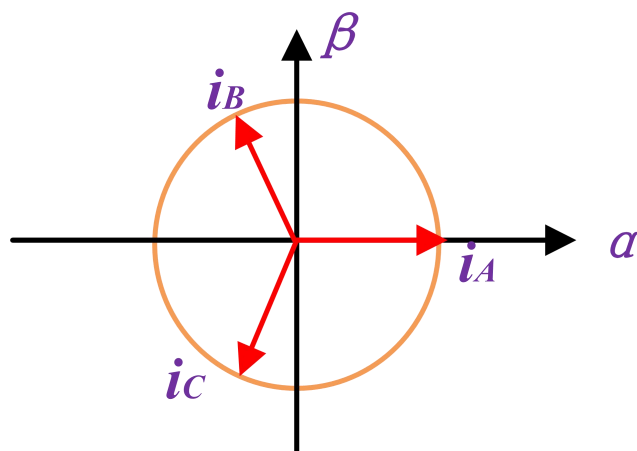


图 2-2-2 平面内电流矢量分布图

把平面内的三个电流矢量经过 Clark 变换（将 abc 变换到静止的 $\alpha\beta$ 坐标系下）

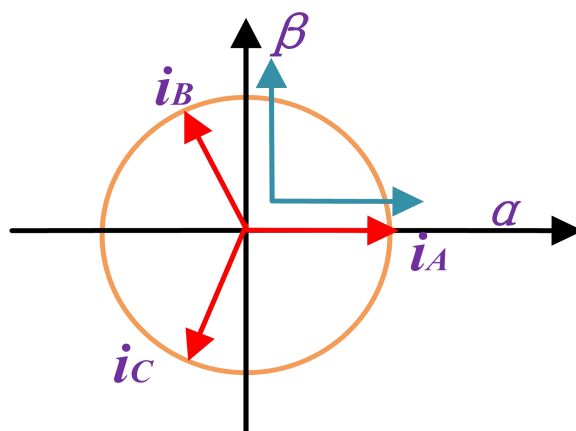


图 2-2-3 电流矢量变换原理图

投影到 α 、 β 轴上可得出：

$$\begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} = \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \cdot \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix}, \quad S_a = \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \quad (2-2-4)$$

由式子（2-2-1）、（2-2-2）、（2-2-3）、（2-2-4）联合可知：

$$\begin{cases} i_\beta = (2i_b + i_a)/\sqrt{3} \\ i_\alpha = i_a \end{cases} \quad (2-2-5)$$

其中 i_α ， i_β 是把平面内的三个矢量用二维坐标来进行表示。为了进一步方便计算，把二维坐标系建在电机转子上，同时由于电机转子在实时转动，则需要一个转化规则，将上述二维坐标系转化为转子上的运动坐标系。随时间变化的绕组电流产生的磁场可等效为旋转磁场，则可以将绕组电流表达在旋转坐标系中。在二维坐标系进行旋转变化时，也就是进行 Park 变换（把 $\alpha\beta$ 坐标系变换到旋转 dq 坐标系下），如图 2-2-4 所示。

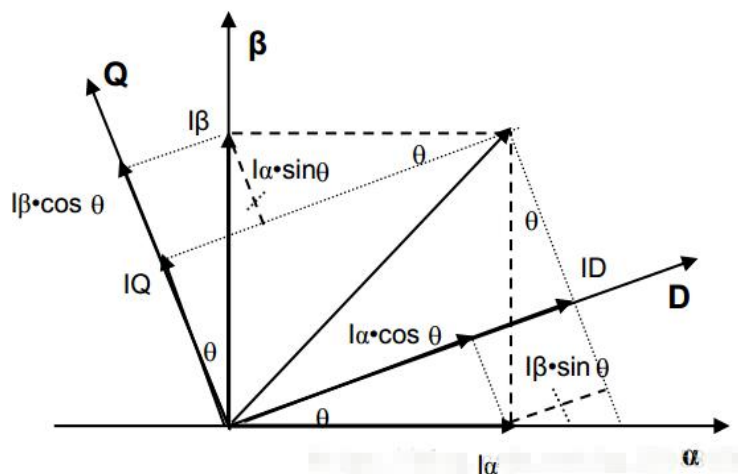


图 2-2-4 dq 变换

经由旋转之后可以得出：

$$\begin{cases} I_D = i_\alpha \cdot \cos \theta + i_\beta \cdot \sin \theta \\ I_Q = -i_\alpha \cdot \sin \theta + i_\beta \cdot \cos \theta \end{cases} \quad (2-2-6)$$

2.2.2 程序设计说明

程序设计思想：电机开始，旋转一周进行寻相，完成寻相后，通过上位机配置参数，选择模式，模式 0 表示电流环设置，双击上位机电流环 PID，设置电流环参数，改变不同的参考电流值 i_{q_ref} 。模式 1 表示速度环设置，双击上位机速度环 PID，设置速度环参数，改变不同的参考速度值 spd_ref 。模式 2 表示位置环设置，双击上位机位置环 PID，设置位置环参数，改变不同的参考位置值 pos_ref ，程序流程框图如 2-2-5 所示。

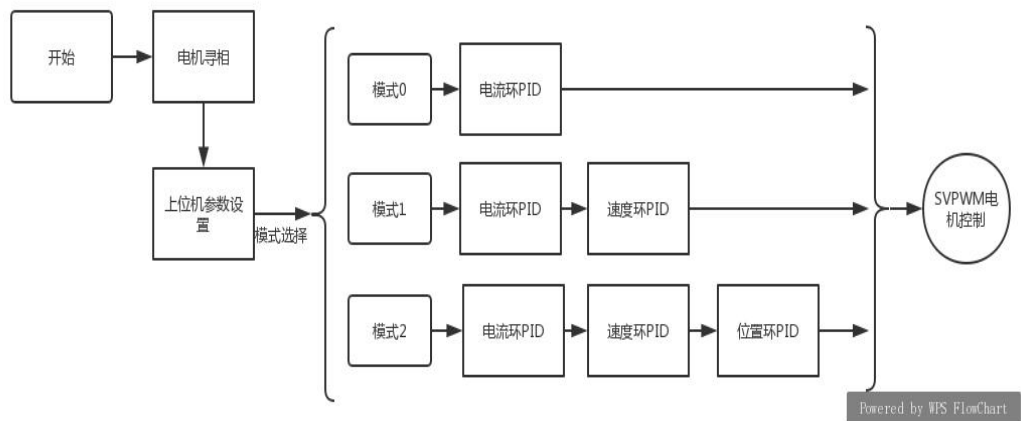


图 2-2-5 程序流程框图

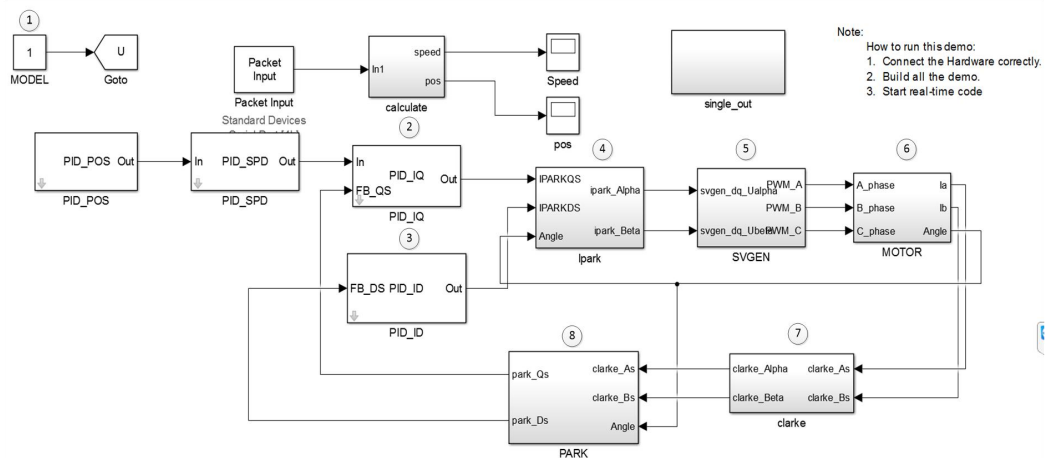


图 2-2-6 交流伺服电机三环控制程序界面

①模式选择，1 为电流环，2 为速度环，3 为位置环。

```

switch (model)//模式一电流环，模式二速度环，模式三位置环
{
    case 1:
    {
        SpeedRef=ref_iq;
        break;
    }
    case 2:
    {
        SpeedRef=ref_spd;
        break;
    }
    case 3:
    {
        SpeedRef=ref_pos;
        break;
    }
}

```

图 2-2-7 模式选择 CCS 程序

②PID 程序，PID 参数设置。

```

pid1_id.Kp = _IQ(recbuff[1] / 100.0);//kp参数设置
if(recbuff[2]==0)pid1_id.Ki=_IQ(0);
else pid1_id.Ki = _IQ(T/(recbuff[2] / 200.0));//ki参数设置
pid1_id.Kd = _IQ((recbuff[3] / 10000.0)/T);//kd参数设置
pid1_id.kh = _IQ(recbuff[4] / 1.0);//开关
ref_id=_IQ(recbuff[5] / 100.0);//ID模拟量

pid1_iq.Kp = _IQ(recbuff[6] / 100.0);//kp参数设置
if(recbuff[7]==0)pid1_iq.Ki=_IQ(0);
else pid1_iq.Ki = _IQ(T/(recbuff[7] / 100.0));//ki参数设置
pid1_iq.Kd = _IQ((recbuff[8] / 10000.0)/T);//kd参数设置
pid1_iq.kh = _IQ(recbuff[9] / 1.0);//ID模拟量
ref_iq=_IQ(recbuff[10] / 100.0);//ID模拟量

```

图 2-2-8 PID 程序

③PID 结构代码块，构建 PID。

```

#define PID_MACRO(v)                                /* PID构造 */
    v.Err = _IQmpy((v.Ref - v.Fdb),v.kh);           /* Compute the error */
    \
    v.Up= _IQmpy(v.Kp,v.Err);                        /* Compute the proportional output */
    \
    v.Ui= v.Ui + _IQmpy(v.Ki,v.Err);                /* Compute the integral output */
    \
    v.Ui= _IQsat(v.Ui, v.OutMax, v.OutMin);
    \
    v.Ud = _IQmpy(v.Kd,(v.Up - v.Up1));             /*wei fen*/
    \
    v.OutPreSat = v.Up + v.Ui + v.Ud;               /* Compute the pre-saturated output */
    \
    v.Out = _IQsat(v.OutPreSat, v.OutMax, v.OutMin);/* Saturate the output */
    \
    v.Up1 = v.Up;                                   /* Update the previous proportional output */
#endif // __PIDREG3_H__

```

图 2-2-9 PID 结构程序

④IPark 模块，Park 变换的逆变换。


```

#define IPARK_MACRO(v)      /*iPARK变换, PARK变换的反变换*/
v.Alpha = _IQmpy(v.Ds,v.Cosine) - _IQmpy(v.Qs,v.Sine);
v.Beta = _IQmpy(v.Qs,v.Cosine) + _IQmpy(v.Ds,v.Sine);

#endif // __IPARK_H__

```

图 2-2-10 CCS 程序中的 IPark 变换

⑤计算 PWM 占空比模块。

```

// -----
// Connect inputs of the SVGEN_DQ module and call the space-vector gen. macro
// -----
svgen_dq1.Ualpha = ipark1.Alpha;
svgen_dq1.Ubeta = ipark1.Beta;
SVGEN_MACRO(svgen_dq1)

```

图 2-2-11 计算 PWM 占空比 CCS 程序

⑥电机三相输入电压。

```

// -----
// Connect inputs of the PWM_DRV module and call the PWM signal generation macro
// -----
pwm1.MfuncC1 = _IQtoQ15(svgen_dq1.Ta);
pwm1.MfuncC2 = _IQtoQ15(svgen_dq1.Tb);
pwm1.MfuncC3 = _IQtoQ15(svgen_dq1.Tc);
PWM_MACRO(pwm1) // Calculate the new PWM compare values

EPwm1Regs.CMPA.half.CMPA=pwm1.PWM1out; // PWM 1A - PhaseA
EPwm2Regs.CMPA.half.CMPA=pwm1.PWM2out; // PWM 2A - PhaseB
EPwm3Regs.CMPA.half.CMPA=pwm1.PWM3out; // PWM 3A - PhaseC

```

图 2-2-12 电机三相输入电压 CCS 程序

3. 实验步骤

3.1 直流无刷电机六步换相实验步骤

3.1.1 参考实验设计部分

1. 首先观察图 3-1-1 所示代码块，变量 pwm1.CmtnPointer 表示当前绕组电流的方向（参数范围 0~5）。

```

pwm1.CmtnPointer = 0;//State s1:
pwm1.CmtnPointer = 1;//State s2:
pwm1.CmtnPointer = 2;//State s3:
pwm1.CmtnPointer = 3;//State s4:
pwm1.CmtnPointer = 4;//State s5:
pwm1.CmtnPointer = 5;//State s6:

```

图 3-1-1 电流方向代码块

2. 观察如图 3-1-2 所示，变量 hall1.HallGpioAccepted 表示此时霍尔传感器的

扇区（参数范围 1~6）。

```
hall1.HallGpioAccepted == 5//当前转子所处位置经由霍尔传感器反馈为第5扇区
hall1.HallGpioAccepted == 1//当前转子所处位置经由霍尔传感器反馈为第1扇区
```

图 3-1-2 当前霍尔传感器扇区

```
#if MOTOR_DIR == 0// Comment the following if-else-if statements in case of
// non-inverted Hall logics for commutation states.
if (hall1.HallGpioAccepted == 2)
    pwm1.CmtnPointer = 0;
else if (hall1.HallGpioAccepted == 6)
    pwm1.CmtnPointer = 1;
else if (hall1.HallGpioAccepted == 4)
    pwm1.CmtnPointer = 2;
else if (hall1.HallGpioAccepted == 5)
    pwm1.CmtnPointer = 3;
else if (hall1.HallGpioAccepted == 1)
    pwm1.CmtnPointer = 4;
else if (hall1.HallGpioAccepted == 3)
    pwm1.CmtnPointer = 5;
```

图 3-1-3 逆时针旋转换向控制代码

在理解上述 3-1-1、3-1-2、3-1-3 所示代码块之后，加上对六步换相原理的分析，在 CCS 中导入工程 BLDC_SD（详见附录 A）后，打开 main.c 文件，在图 3-1-4 所示空白处填写关于六步换相电机反转的核心代码（**项目中可能已经给出**）：


① 通过 Ctrl+F 搜索 MOTOR_DIR 进行定位。参考在项目工程中已经给出的实验正转代码 MOTOR_DIR==0；

② 通过观察给出的 MOTOR_DIR==0 的代码，结合六步换相的原理，推断并且填写出 MOTOR_DIR==1 的电机反转代码，并在头文件更改 MOTOR_DIR 的赋值，实现电机反转。

```
947 白#elif MOTOR_DIR == 1
948 白// Comment the following if-else-if statements in case of
949 白// inverted Hall logics for commutation states.
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964 白#endif
965 白    }
966 白    }
```

图 3-1-4 填写“六步换相”核心代码

3.1.2 六步换向操作步骤

	注意	先上电再用 USB 接口下载代码。
		CN1-CN6 接口不支持带电插拔，请先切断掉电源。

1. 线路连接: CN3 和 CN5 连接、CN4 和 CN6 连接, USB 与上位机 USB 连接。在连接 USB 接口前时,需要先打开**设备管理器**——点击**端口**;然后查看现有端口,再插上 USB 线观察哪一个端口是重新出现的,并做好实验记录,为后续操作做好准备。向上推动断路器开关上电,实验平台面板框架如图 3-1-5 所示;

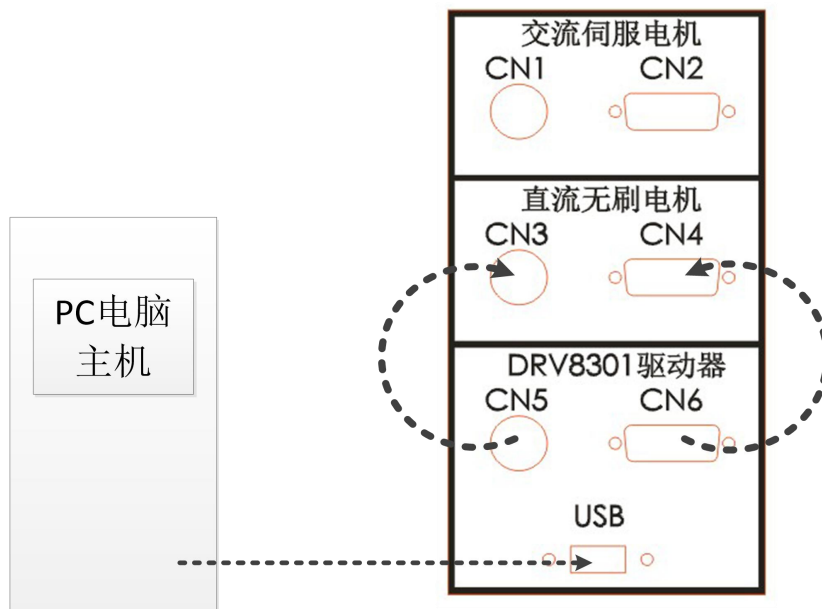


图 3-1-5 硬件接线

2. 打开 CCS, 在目录“C:\ProgramFiles\MATLAB\R2015b\workspace\pwm\BLDC”下导入工程 BLDC, **工程导入详见附录 A**。

编译完成后点击仿真下载,若点击红框的运行按钮,软件开始运行,若表现改图表颜色暗淡,则已经处于运行状态,不必再次点击。如图 3-1-6 所示; **注意:一定要按照附录 A 流程完整的导入 CCS 工程 BLDC,在第三个步骤前保持程序处于持续运行状态。**

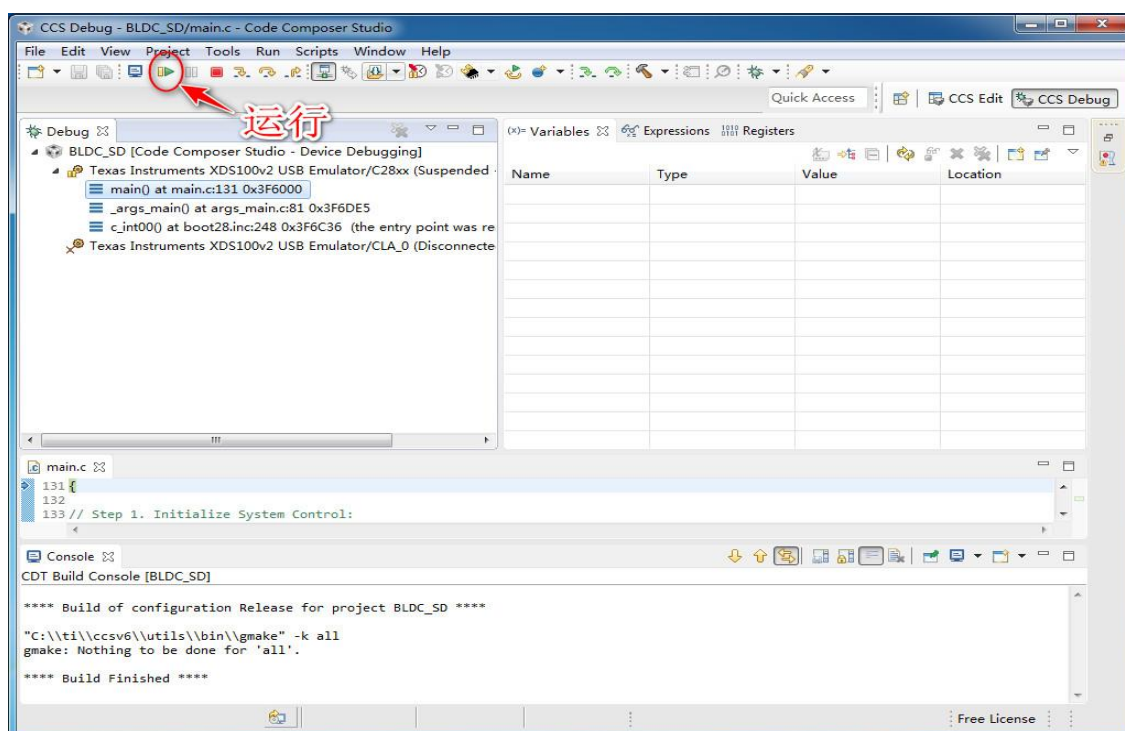


图 3-1-6 仿真界面

3. 打开 Matlab, 打开目录“C:\ProgramFiles\MATLAB\R2015b\workspace\pwm\BLDC”下 speed_control 程序, 如图 3-1-7 所示;

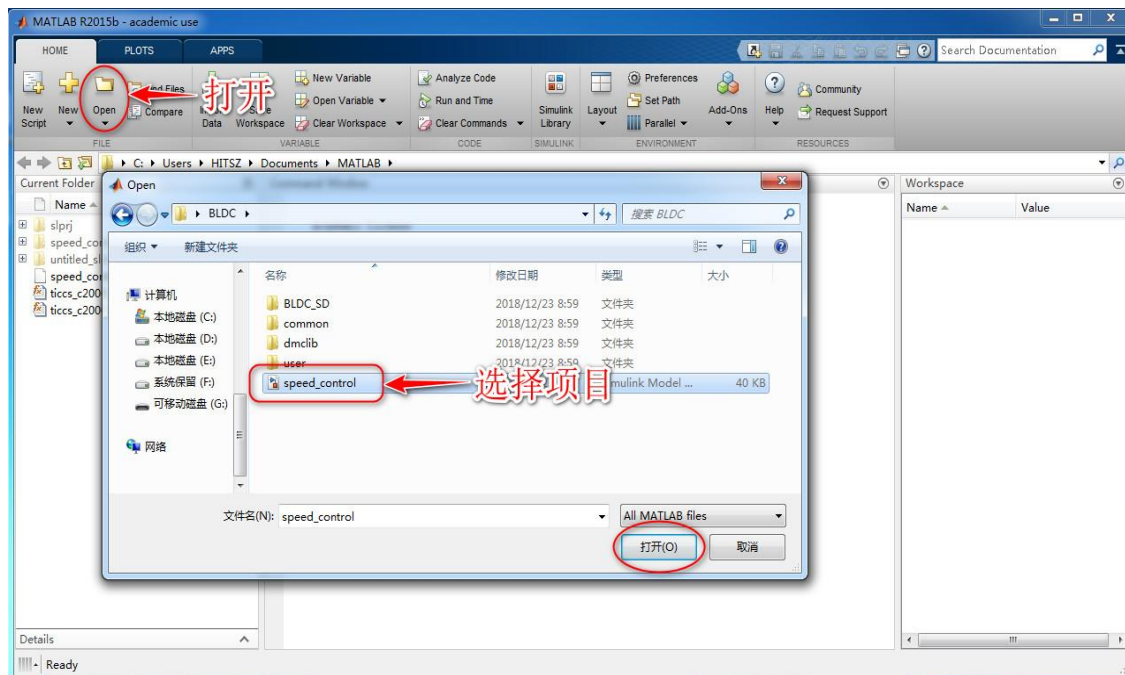


图 3-1-7 Matlab 软件打开项目

4. 打开程序的路径必须是全英文路径, 如图 3-1-8 所示;

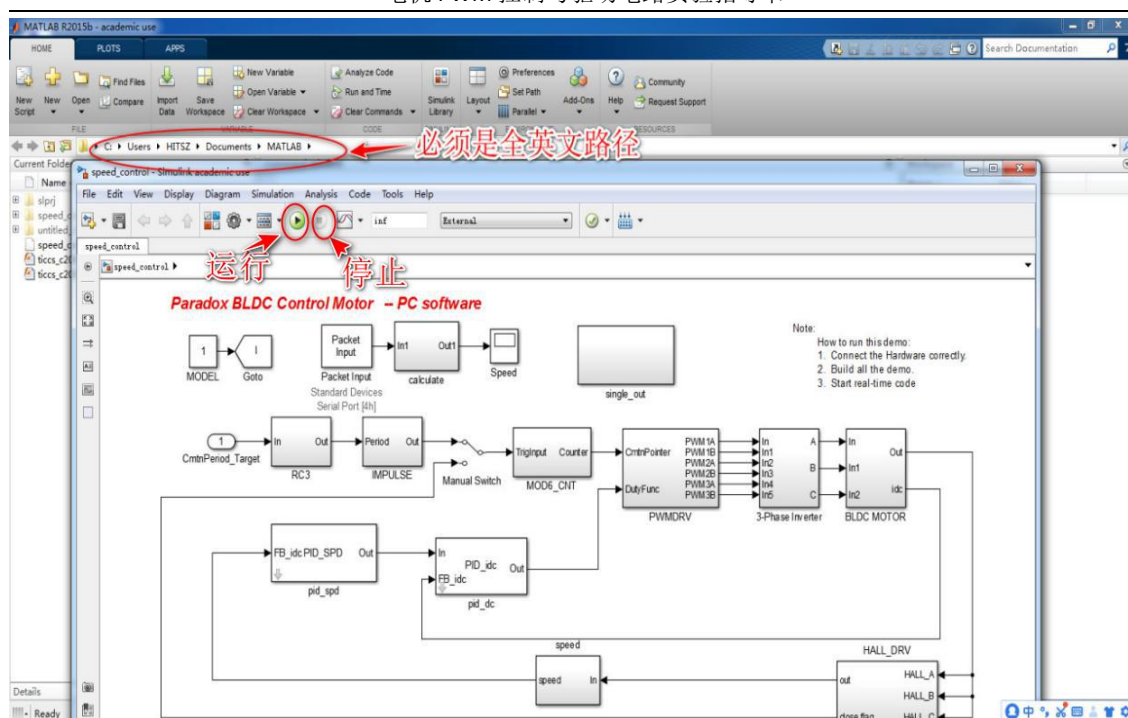


图 3-1-8 全英文程序路径示意图

5. 双击 Packet Input 模块设置 Packet Input 的 COM 口，如图 3-1-9 所示；

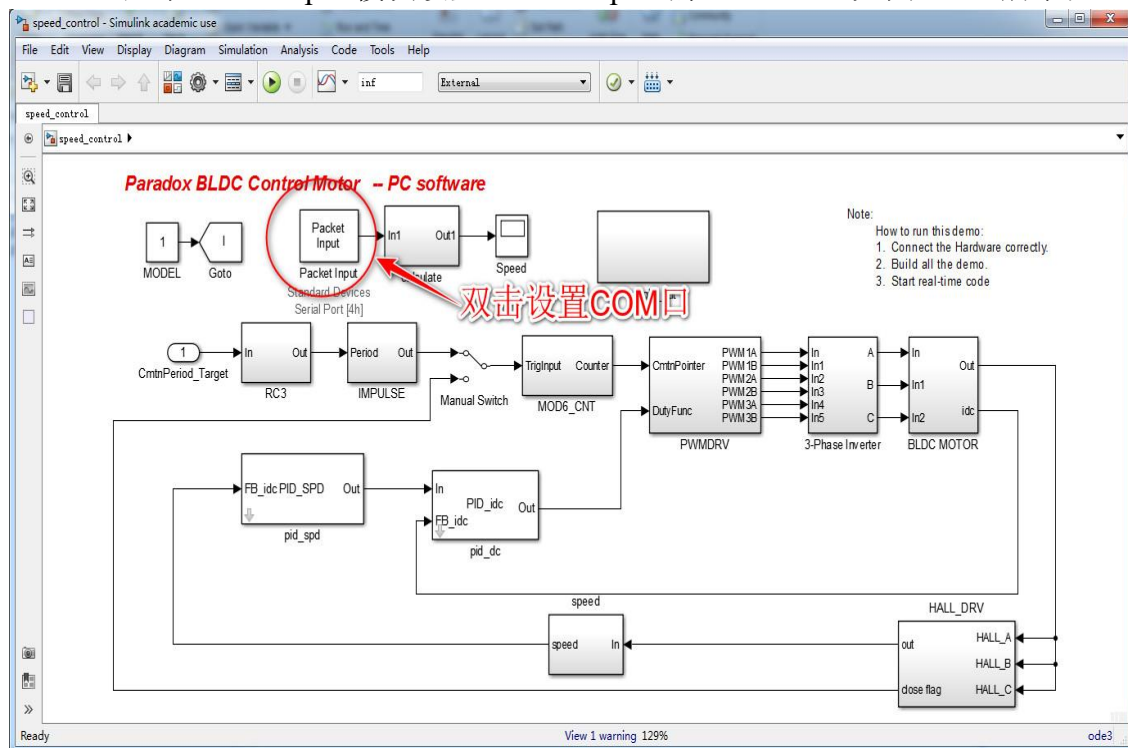


图 3-1-9 双击 Packet Input 模块示意图

6. 在弹出的窗口中选择 Board setup 选项。在弹出的对话框中设置 COM 口参数。在第一个步骤时已经对新出现的端口做了记录，此时的端口需要与记录的端口保持一致，波特率是 115200。点击 Test 以检查系统是否能够正常通信，出现绿色√则表示通信正常。如图 3-1-10 所示；

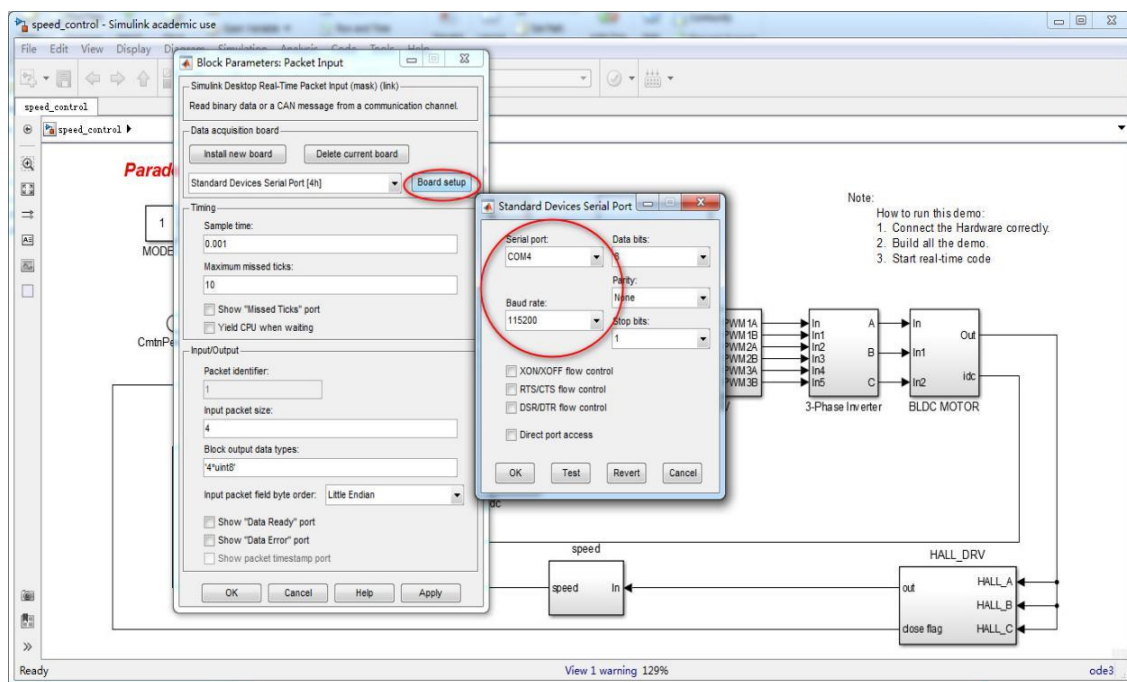


图 3-1-10 COM 口参数设置界面

7. 双击 single-out 模块，如图 3-1-11 所示；

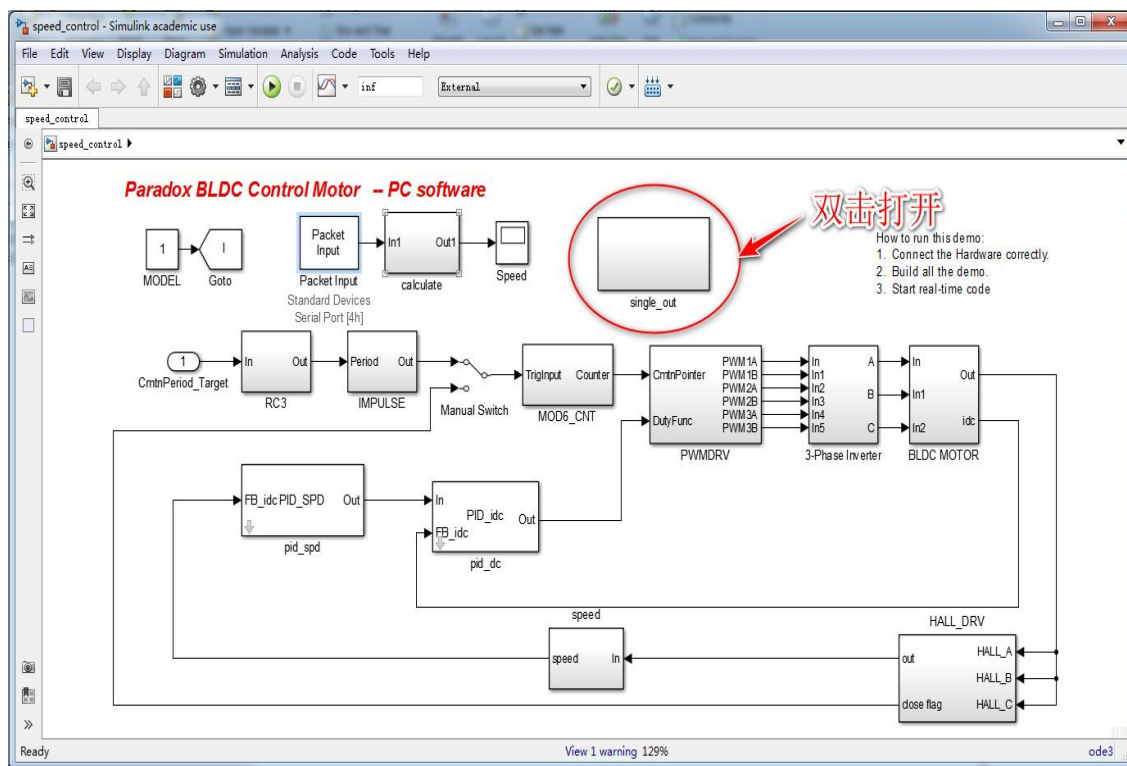


图 3-1-11 双击 single-out 模块示意图

8. 双击 Packet Output 模块设置 Packet Input 的 COM 口，如图 3-1-12 所示；

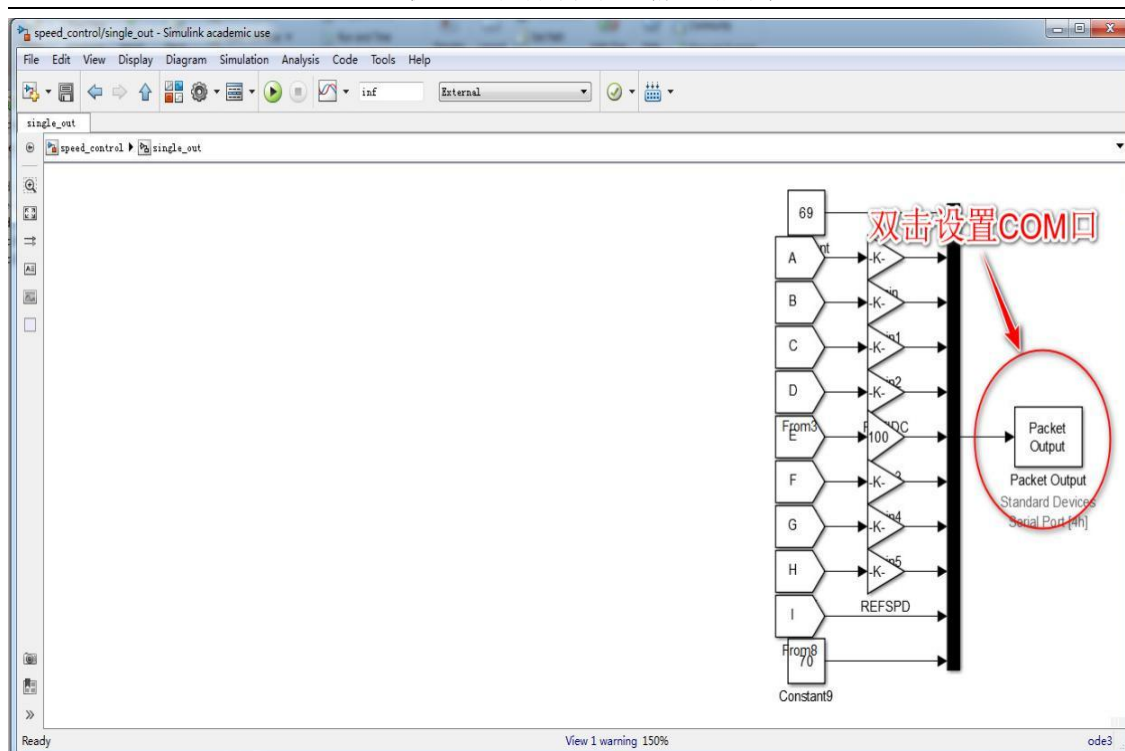


图 3-1-12 双击 Packet Output 模块示意图

9. 在弹出的窗口中选择 **Board setup** 选项，在弹出的对话框中设置 COM 口参数。在第一个步骤时已经对新出现的端口做了记录，此时的端口仍然需要与记录的端口保持一致，波特率是 115200。点击 **Test** 以检查系统是否能够正常通信，出现绿色 \checkmark 则表示通信正常。如图 3-1-13 所示；

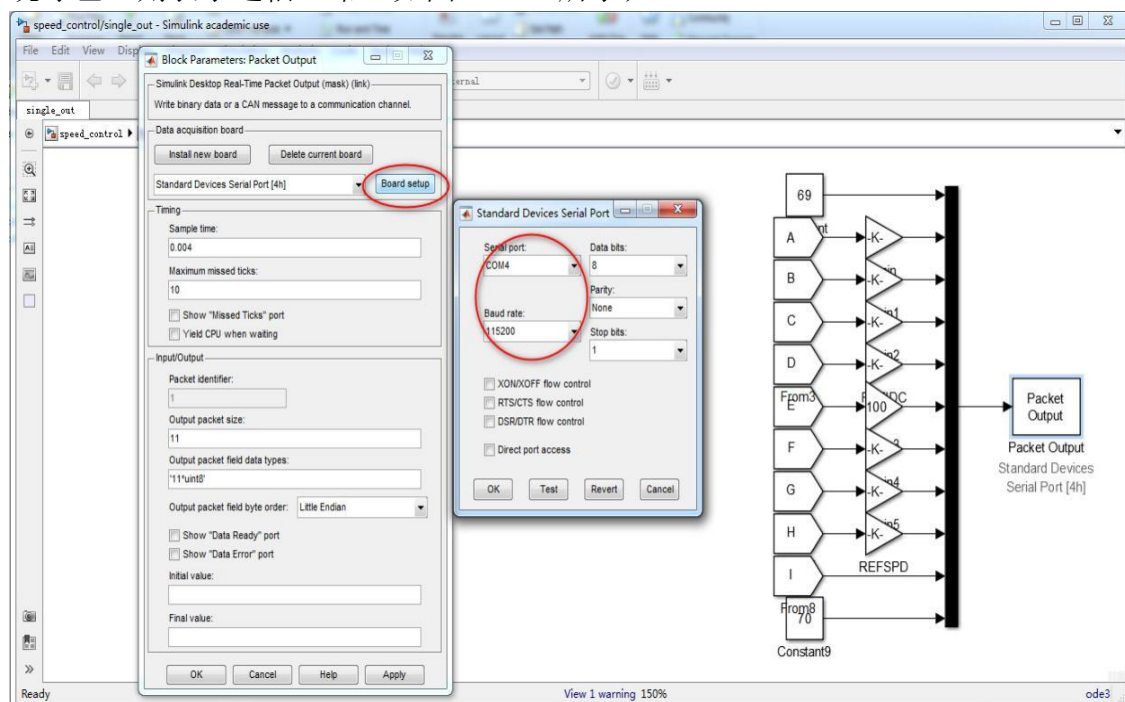


图 3-1-13 设置 COM 口参数界面

10. 把电机模式改为 1（电流环运动模式），双击 **MODEL** 模块，在弹出的对话框中 **Constant Value** 填入 1（电流环运动模式），单击 **OK**，如图 3-1-14 所示；

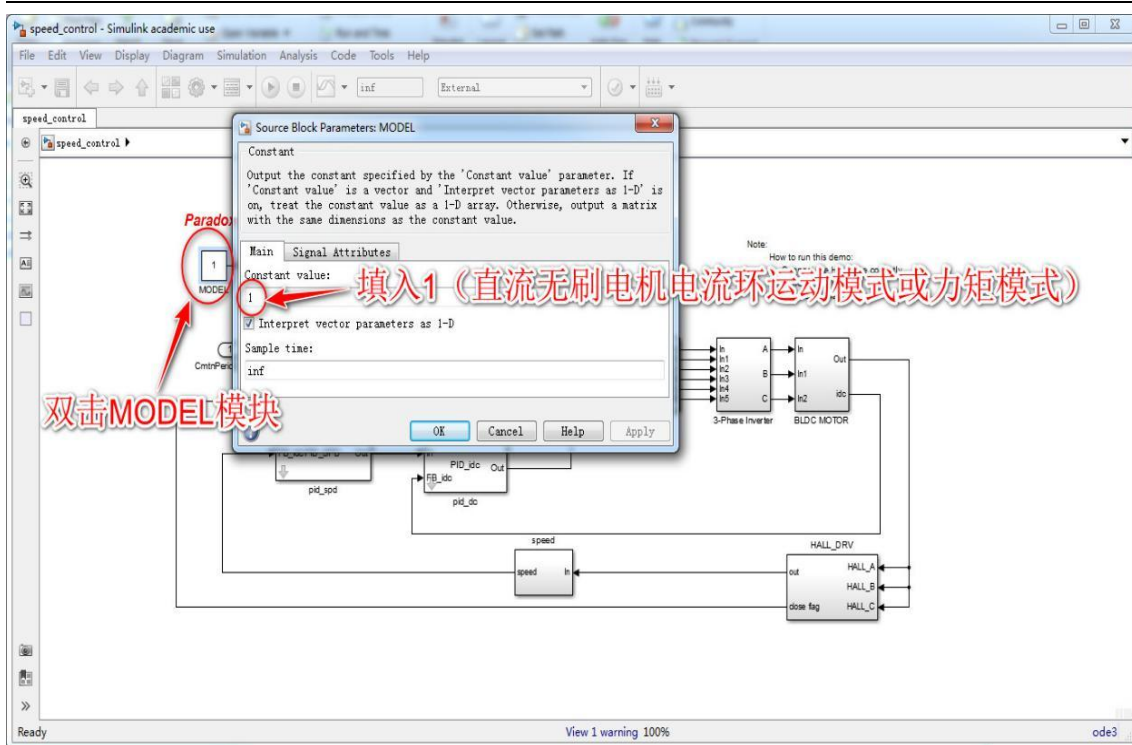


图 3-1-14 设置为电流环运动模式（力矩模式）界面

11. 启动运行程序，打开 PID 模块（力矩模式 PID），如图 3-1-15 所示；

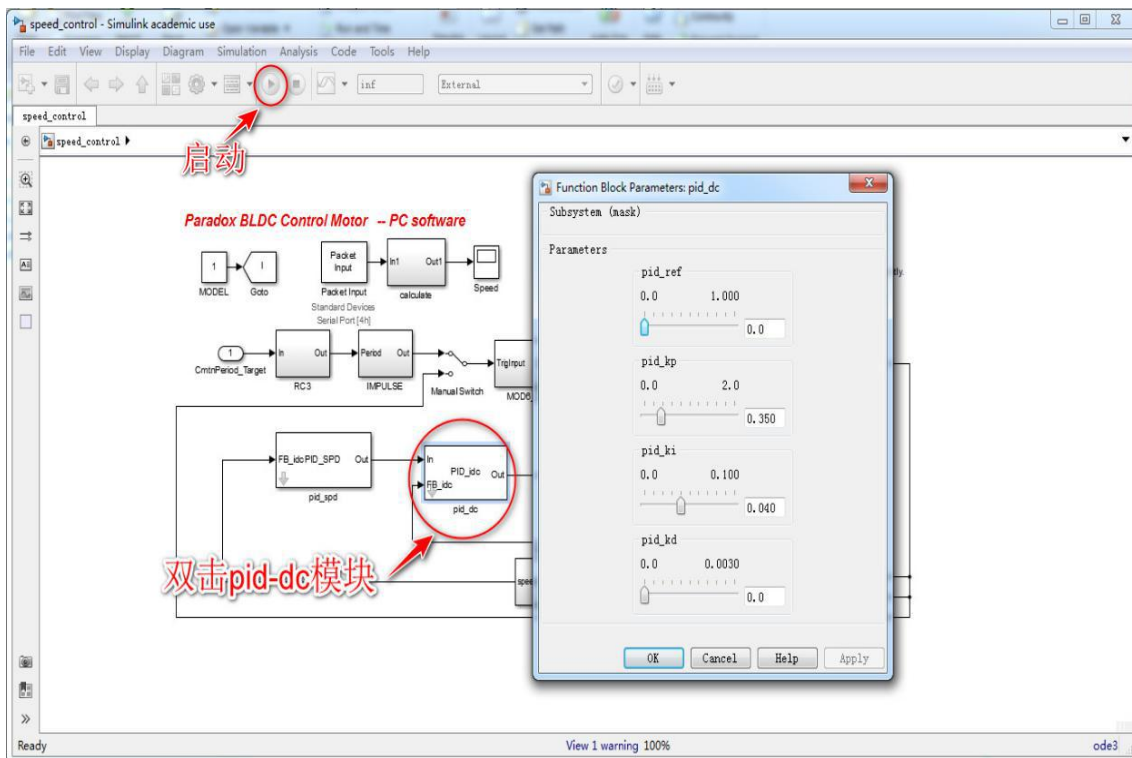


图 3-1-15 力矩模式 PID 模式参数设置面板

12. 速度显示模块会自动弹出，若未出现，则打开 Speed（速度）模块，如图 3-1-16 所示；

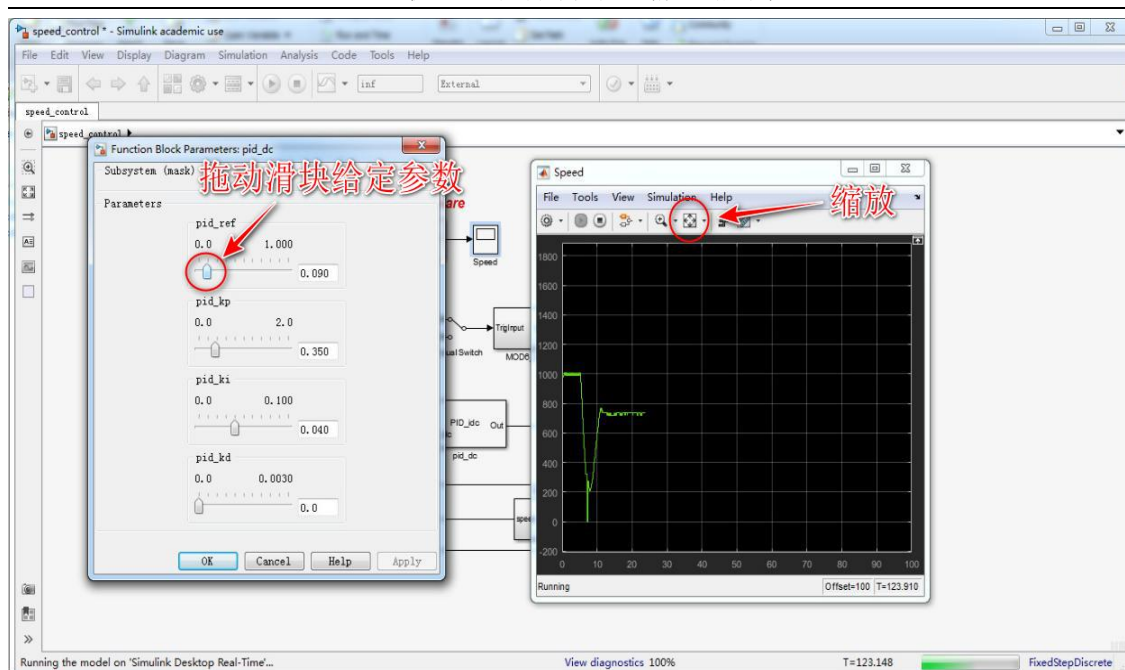


图 3-1-16 力矩模式速度实时监控

13. 在对话框中拖动滑块给定 `pid_ref` 参考电流，拖动滑块给定 PID 参数，电机开始运动（参考电流为 0 时电机不动）。运行结束拖动滑块使参考电流为 0，电机停转，再点击停止按钮，打开 `speed` 示波器模块，点击缩放按钮，可以看到完整的运行过程的曲线图。（注意，第一次滑动滑块时，驱动板上自举电容需要充电，所以电机不会运动，把滑块置零后，再次滑动时电机开始运动）

通过打开 `speed` 或 `pos` 模块观察速度位置变化情况，并保存几张速度或位置变化曲线。其中可以通过 `view-style`-调整图像背景颜色。

如果程序设计有误，电机可能会出现以下现象：

1. 在滑动上位机 PID 模块时，电机开始会动作一下，但不会连续运动；
2. 在 CCS 中编译时程序会报错；
3. 电机完全无动作或有异响。

注意事项：

- (1) 在按下停止按钮前，若没有将 `pid_ref` 滑块置零，电机依然会继续运转，请一定要将 `pid_ref` 置零，再点击停止按钮。
- (2) 程序只需改动关键词 `MOTOR_DIR==1` 处的代码，以及头文件中 `MOTOR_DIR` 的赋值（0 或 1），其余部分勿改动。

3.2 交流伺服电机 Clark/Park 变换实验步骤

3.2.1 参考实验设计部分

1. 根据对 Clark 变换原理的理解，以及给出的变量的意义，设计出 Clark 变换核心代码。

表 3-2-1 关于 Clark 代码的变量含义

变量	含义	注释
v.Alpha	i_α	应用中, v.Alpha 就相当于公式中的 i_α
v.Beta	i_β	应用中, v.Beta 就相当于公式中的 i_β
v.As	i_a	应用中, v.As 就相当于公式中的 i_a
_IQmpy(A,B)	其中, A,B 分别代表两个乘数	_IQmpy(A,B)是一个乘法函数, 表示 A,B 两个乘数相乘
_IQmpy2(A)	其中 A 表示任意一个数	_IQmpy2(A)是一个 2 倍关系, 表示当前 A 的二倍
_IQ		_IQ 表示数据格式, 例如像 int, double 之类的数据格式
v.Bs	i_b	应用中, v.Bs 就相当于公式中的 i_b

2. 打开 CCS，在目录“C:\ProgramFiles\MATLAB\R2015b\workspace\pwm \PMSM”下导入工程 PMSM_SD（详见附录 A），打开 main.c 文件,找到 main.h 按下 Ctrl 再点击 main.h,找到 clarke.h 按下 Ctrl 再点击 clarke.h.,如图 3-2-1 所示,搜索 CLARKE_MACRO 到指定位置。请在下面的空白处,结合对 Clark 变换原理的理解以及应用表 3-2-1 中给出的变量填写核心代码。此实验为验证实验,所以程序中代码已经提前详细给出,直接按后续实验步骤操作即可。（每行代码后都需要加上“\”，不得有空行,否则编译会出错）。

```

37
38 #define CLARKE_MACRO(v)
39
40
41
42
43
44
45 // 1/sqrt(3) = 0.57735026918963
46

```

图 3-2-1 填写 Clark 变换代码

3. 根据对 Park 变换原理的理解，以及给出的变量的意义，设计出 Park 变换的核心代码。项目程序工程中已经默认给出。

表 3-2-2 关于 Park 代码的变量含义

变量	含义	注释
v.Ds	I_D	应用中, v.Ds 相当于公式中的 I_D
v.Qs	I_Q	应用中, v.Qs 相当于公式中的 I_Q
_IQmpy(A,B)	其中, AB 分别代表两个乘数	_IQmpy(A,B)是一个乘法函数, 表示 A,B 两个乘数相乘
v.Alpha	i_α	应用中, v.Alpha 就相当于公式中的 i_α
v.Beta	i_β	应用中, v.Beta 就相当于公式中的 i_β
v.Cosine	$\cos \theta$	应用中, v.Cosine 就相当于公式中的 $\cos \theta$
v.Sine	$\sin \theta$	应用中, v.Sine 就相当于公式中的 $\sin \theta$

4. 打开 main.c 文件，按住键盘上 Ctrl 键，Ctrl+f 搜索 Park_MACRO,按下 Ctrl 使用鼠标点击 PARK_MACRO，进入到 park.h 文件中，如图 3-2-2 所示，请在下面空白处，结合对 Park 变换原理的理解以及应用表 3-2-2 中给出的变量填写所设计的核心代码。（每行代码后都需要加上“\”，不得有空行，否则编译会出错）。（由于为验证实验，程序代码已经给出。）

```


44 #define PARK_MACRO(v)
45
46
47
48
49
50
51 #endif // __PARK_H__
52

```

图 3-2-2 填写 Park 变换代码

5. Clark 和 Park 变换代码设计完成后，点击保存，编译下载代码，并根据后续步骤验证代码。代码设计若无异常，经过实验验证，电机可正常运行，则在上位机中电流环、速度环、位置环都可正常运行。

3.2.2 Clark 和 Park 变换操作步骤

	注意	上电后，用 USB 接口下载代码。
		CN1-CN6 接口不支持带电插拔，必须先切断电源。

1. 线路连接：CN1 和 CN5 连接、CN2 和 CN6 连接，USB 与上位机 USB 连接，连接电源，开关上电，如图 3-2-3 所示；

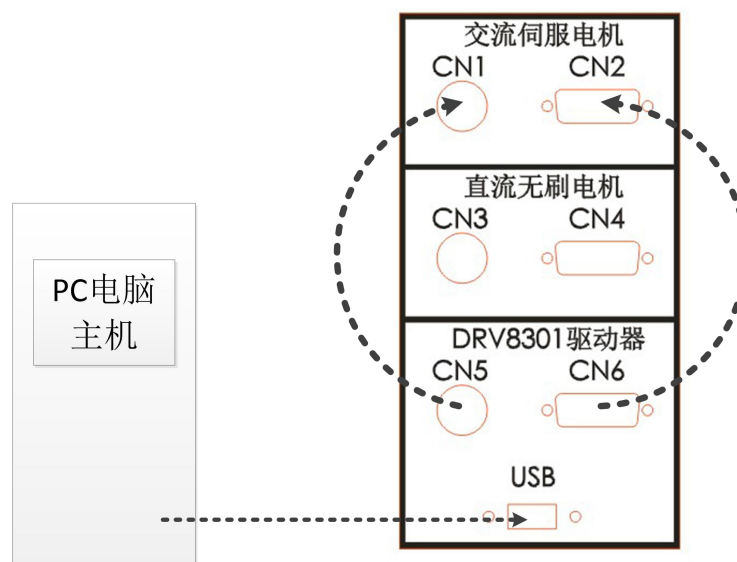


图 3-2-3 硬件接线

2. 打开 CCS，在目录“C:\ProgramFiles\MATLAB\R2015b\workspace\pwm\PMSM”下导入工程 PMSM_SD（工程导入详见附录一）。

在程序编译过程中，出现关键字错误，不存在 kh，打开 main.c，将 kh 所在的代码行前出现红色小×号的那一行代码给注释掉，如果 kh 前不存在红色小×号不要注释掉这一行代码。

顺利编译完成后点击仿真下载，点击运行按钮，同时观察电机圆盘是否旋转一圈寻相完成。如果未出现转一圈自动寻相，则重新启动编译程序直到完成为止，不然后续流程不能完成。如图 3-2-4 所示；

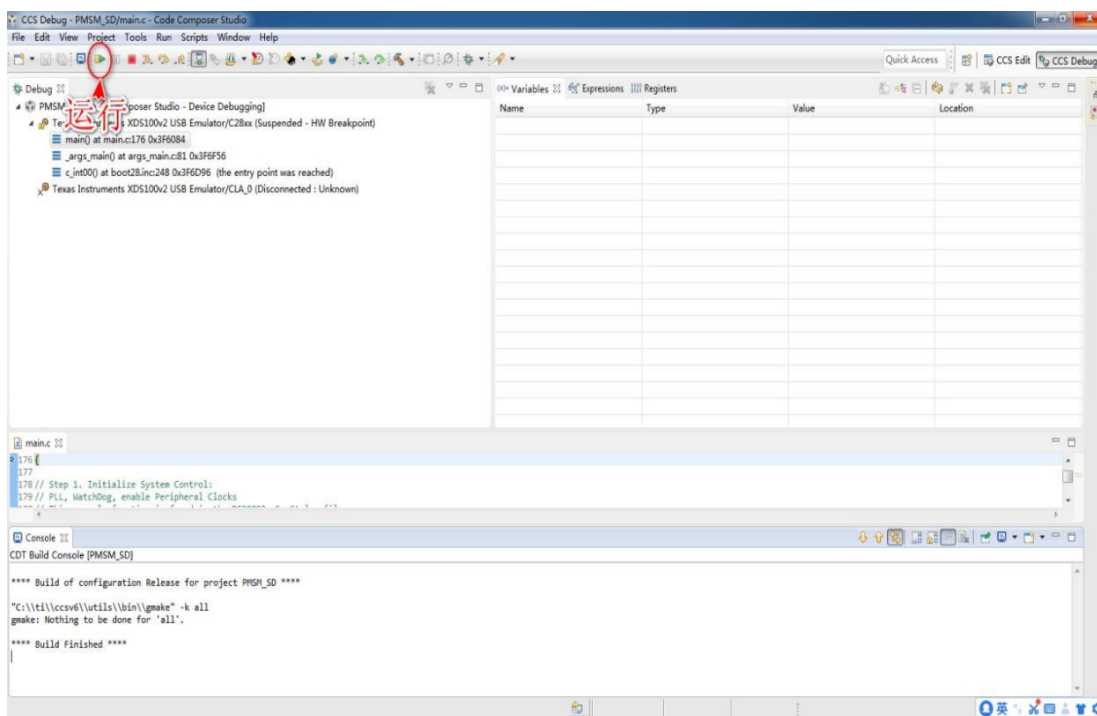


图 3-2-4 仿真启动界面

3. 打开 Matlab 软件，单击 Open，在目录“C:\ProgramFiles\MATLAB\R2015b\workspace\pwm\PMSM”下打开 PMSM 项目，如图 3-2-5 所示；

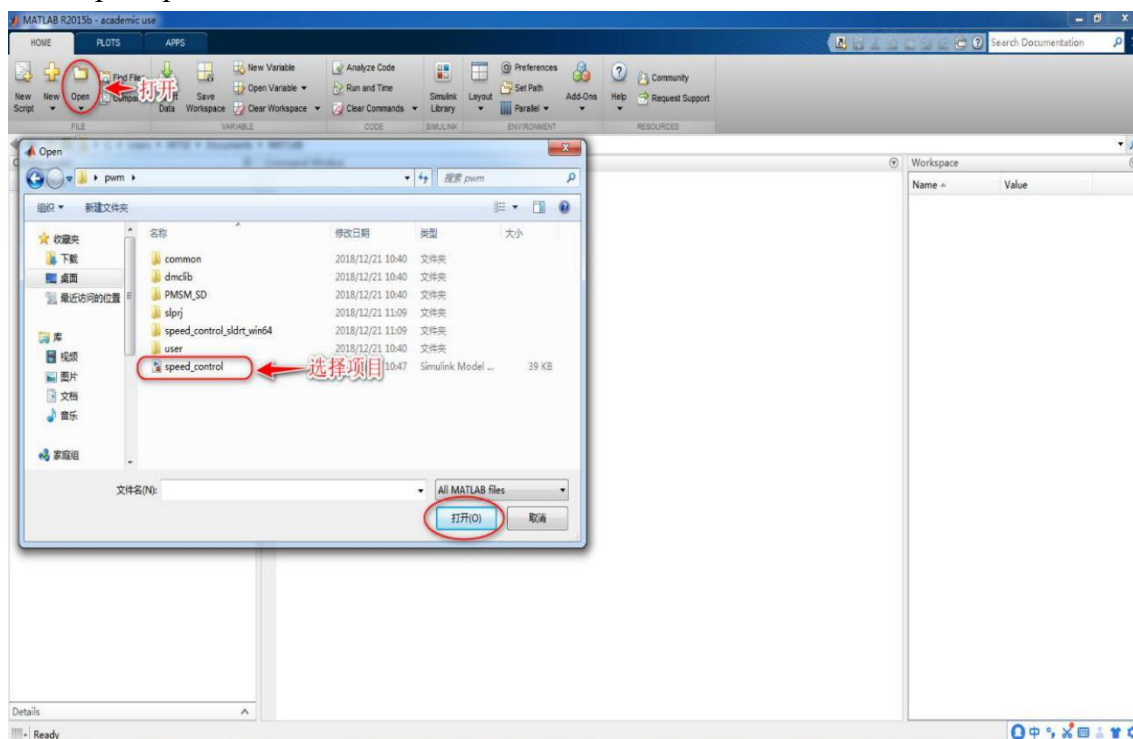


图 3-2-5 Matlab 软件打开项目

4. 打开程序的路径必须是全英文，如图 3-2-6 所示；

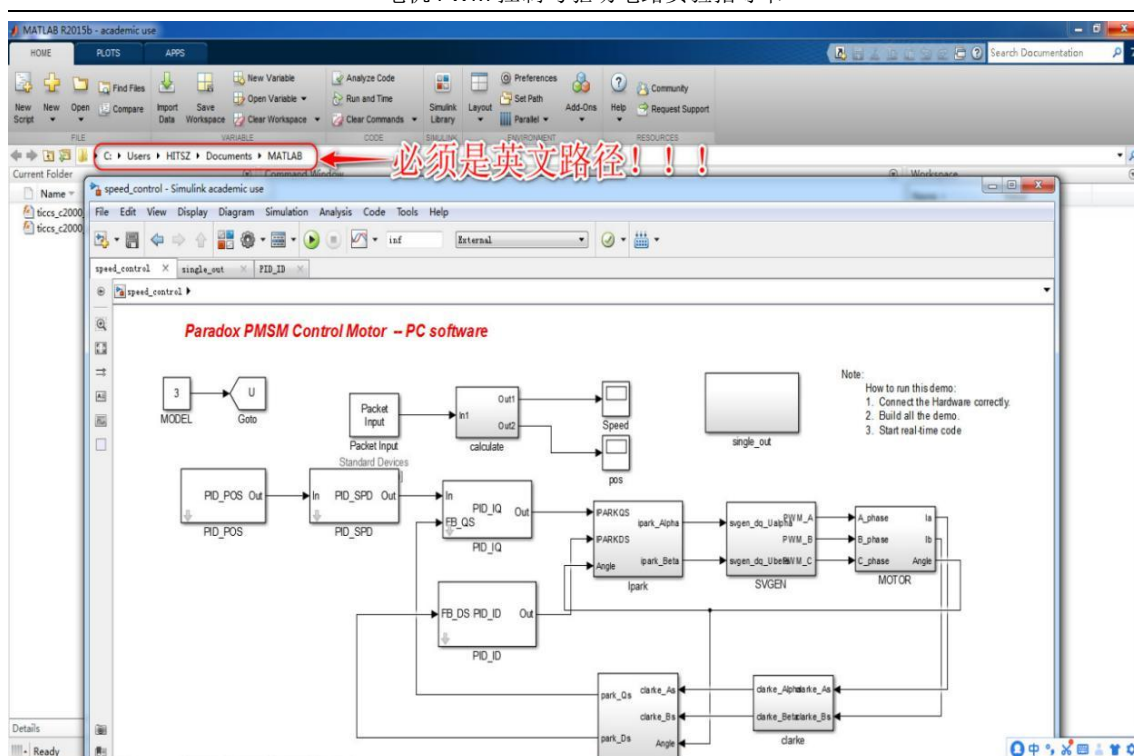


图 3-2-6 全英文程序路径示意图

5. 双击 Packet Input 模块设置 Packet Input 的 COM 口，如图 3-2-7 所示；

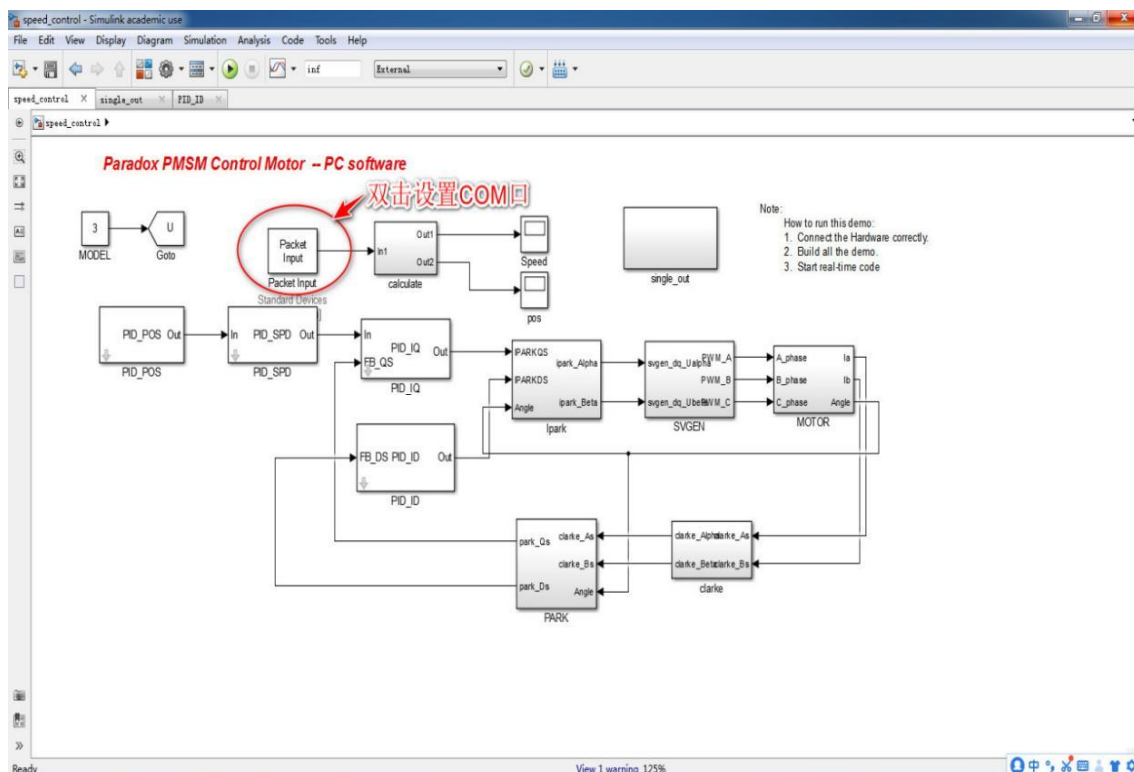


图 3-2-7 双击 Packet Input 模块示意图

6. 在弹出的窗口中电机 Board setup 选项，在弹出的对话框中设置 COM 口参数，如图 3-2-8 所示；

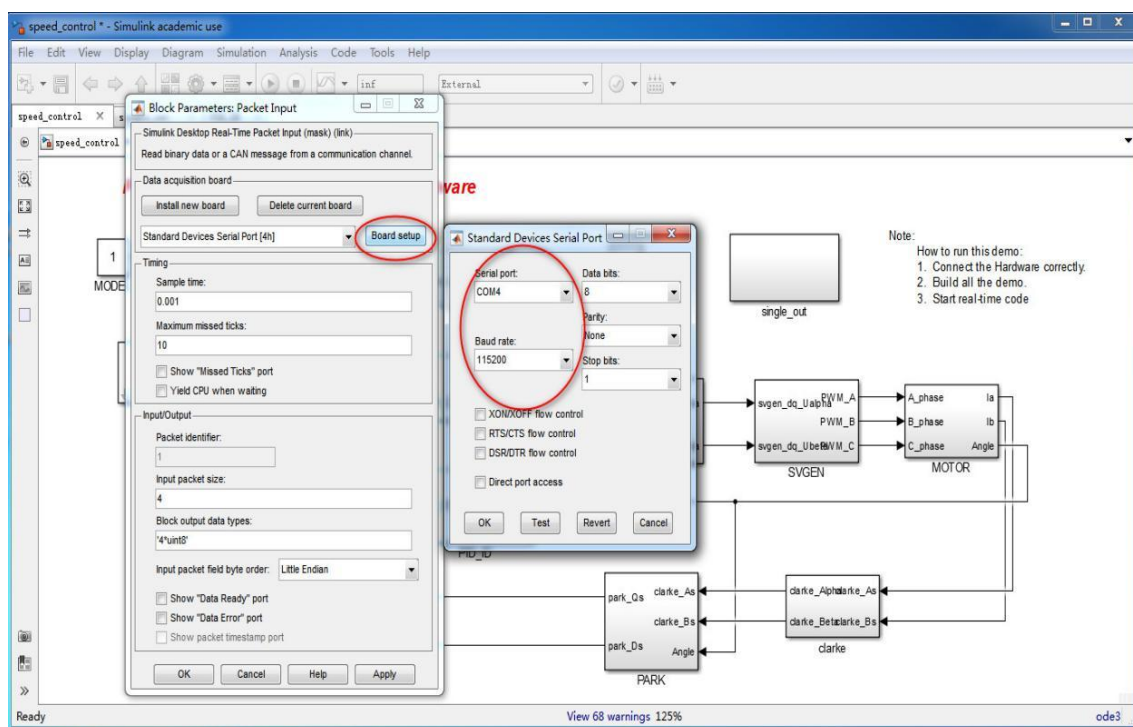


图 3-2-8 设置 COM 口参数

7. 打开 single-out 模块，如图 3-2-9 所示；

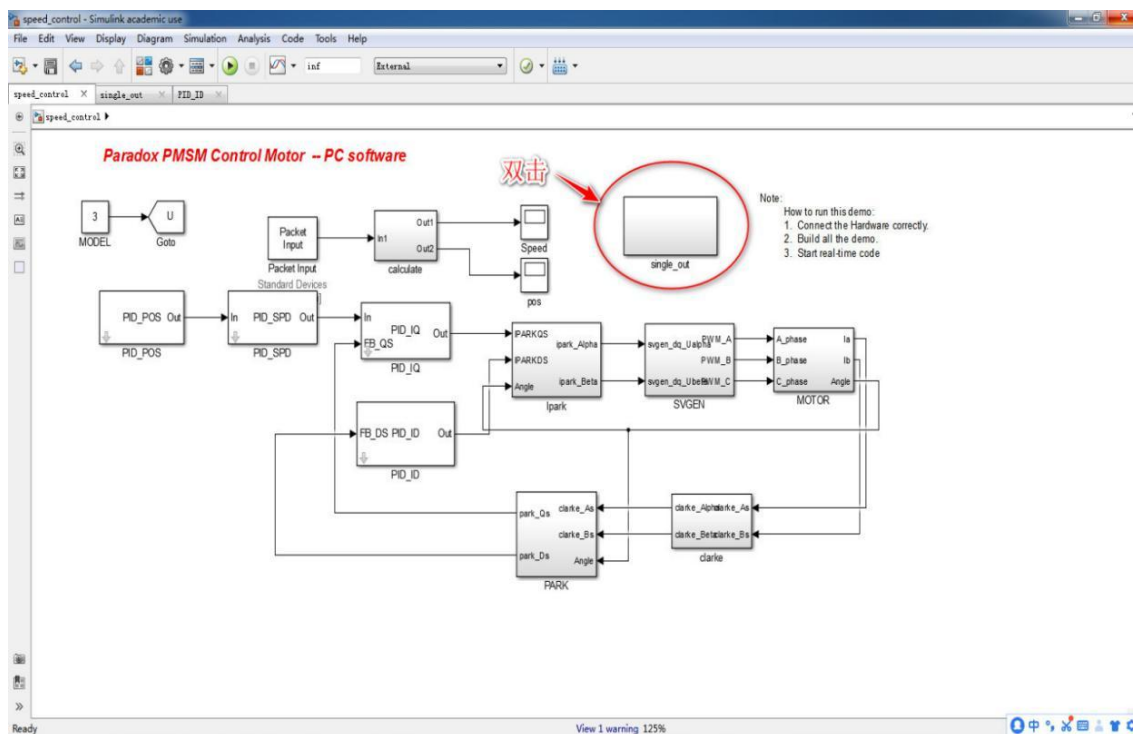


图 3-2-9 打开 single-out 模块

8. 打开 Packet Output 模块，设置 Packet Input 的 COM 口，如图 3-2-10 所示；

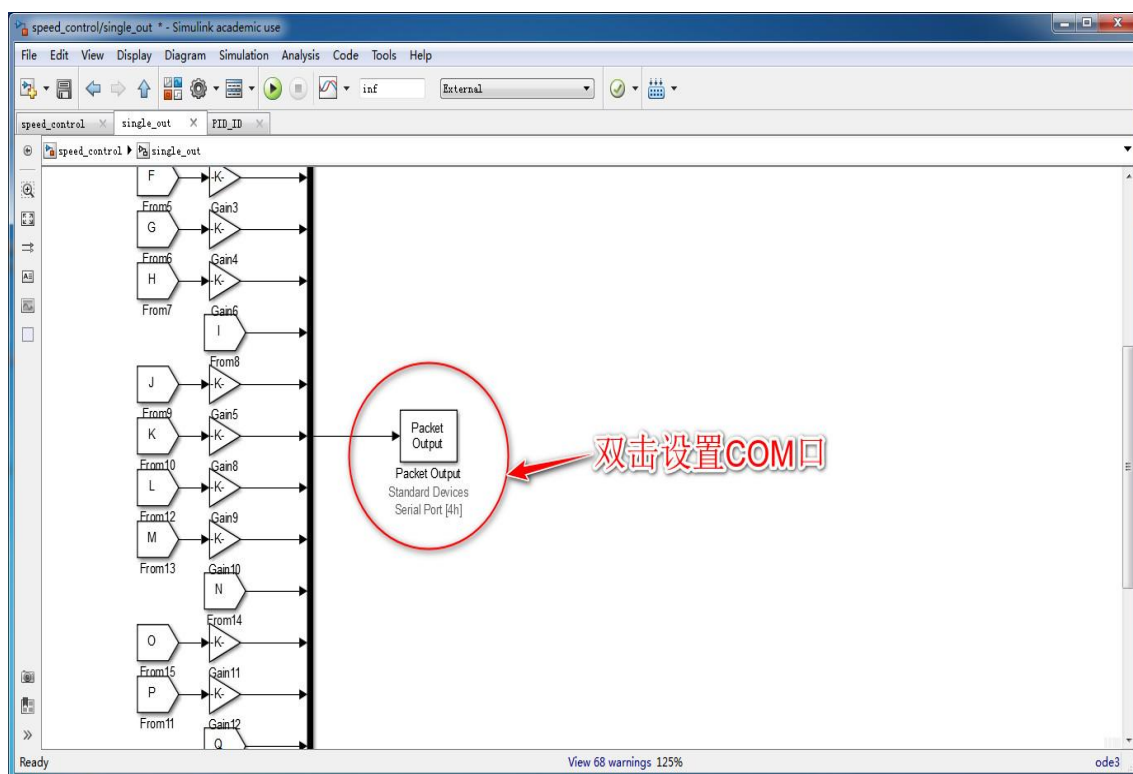


图 3-2-10 双击 Packet Output 模块

9. 在弹出的窗口中电机 Board setup 选项，在弹出的对话框中设置 COM 口参数，如图 3-2-11 所示；

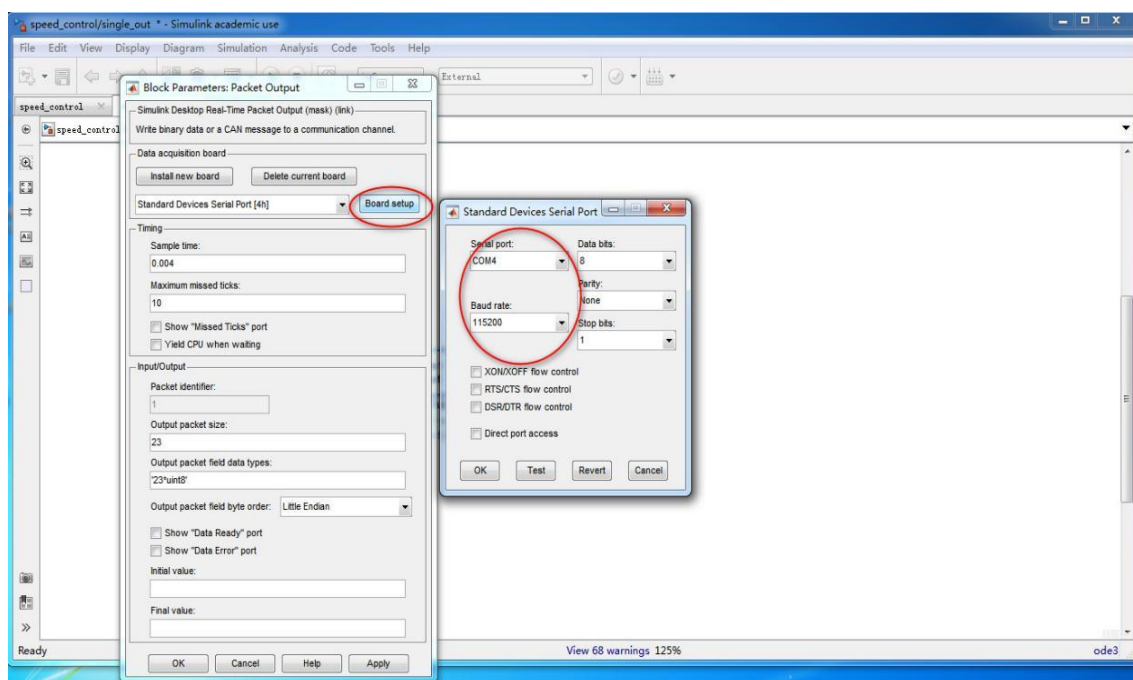


图 3-2-11 设置 COM 口参数

10. 把电机模式改为 1（电流环运动模式），打开 MODEL 模块，在弹出的对话框中 Constant Value 选项中填入 1（电流环运动模式），单击 OK，如图 3-2-12 所示；

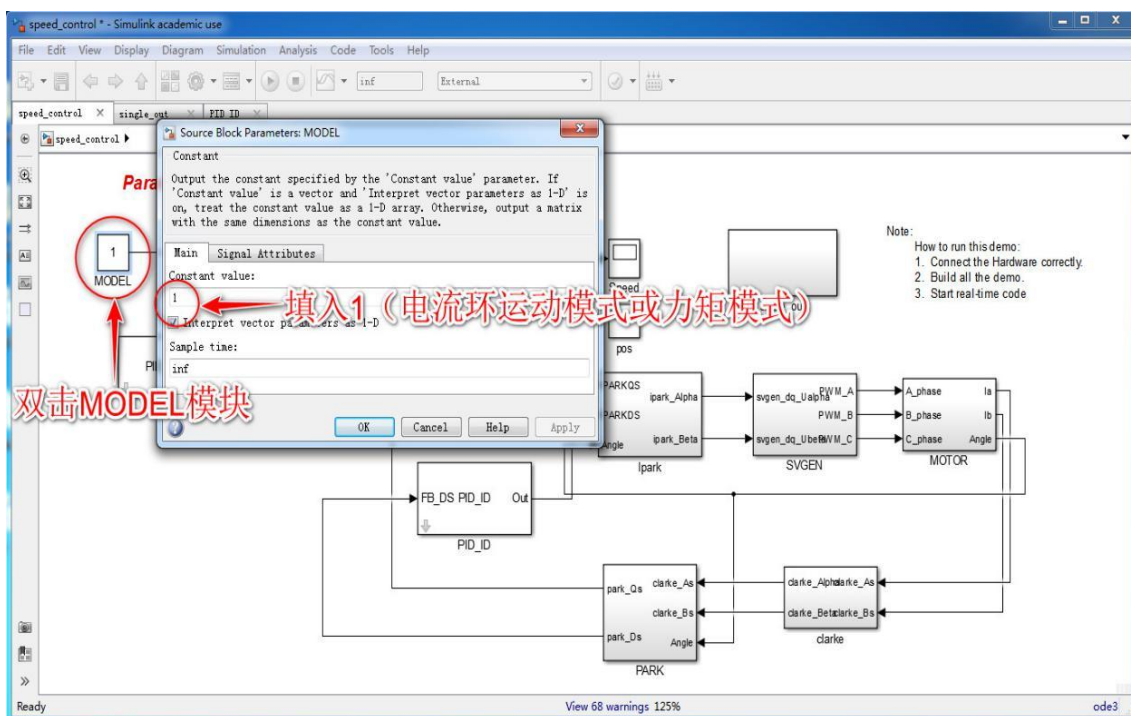


图 3-2-12 设置为电流环模式（力矩模式）示意图

11. 启动运行程序，打开 PID_IQ 模块（力矩模式 PID），如图 3-2-13 所示；

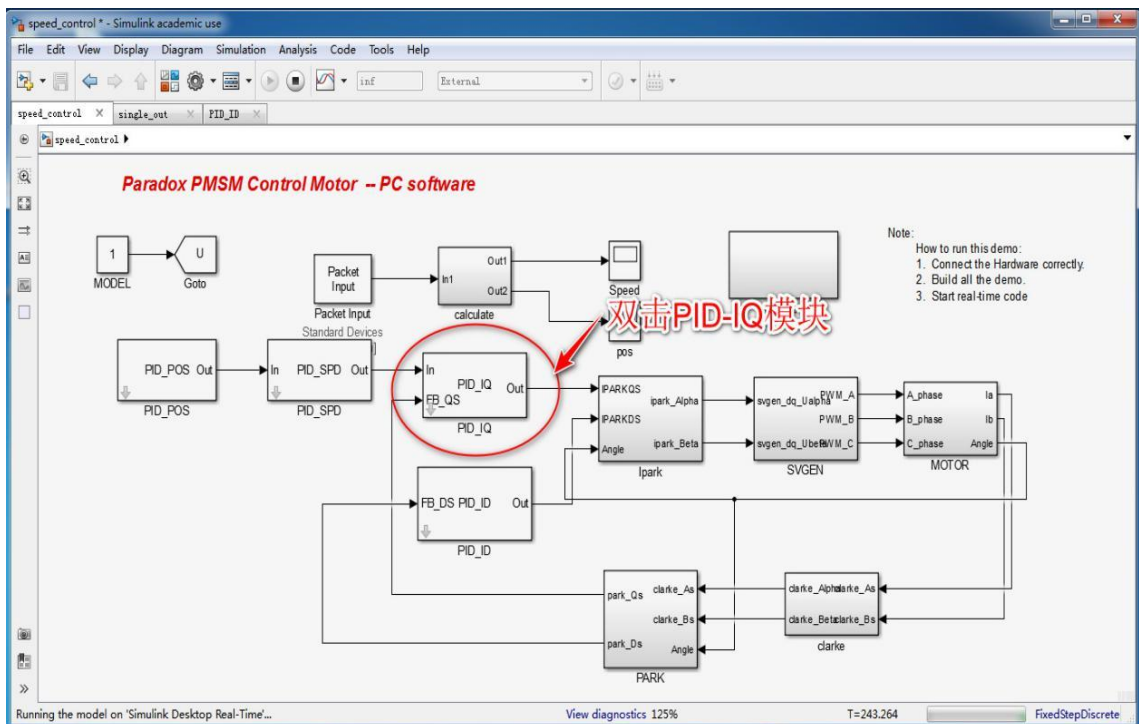


图 3-2-13 力矩模式 PID 模块

12. 在对话框中拖动滑块给定 pid_ref 参考电流，拖动滑块给定 PID 参数，电机开始运动（参考电流为 0 时电机不动）；

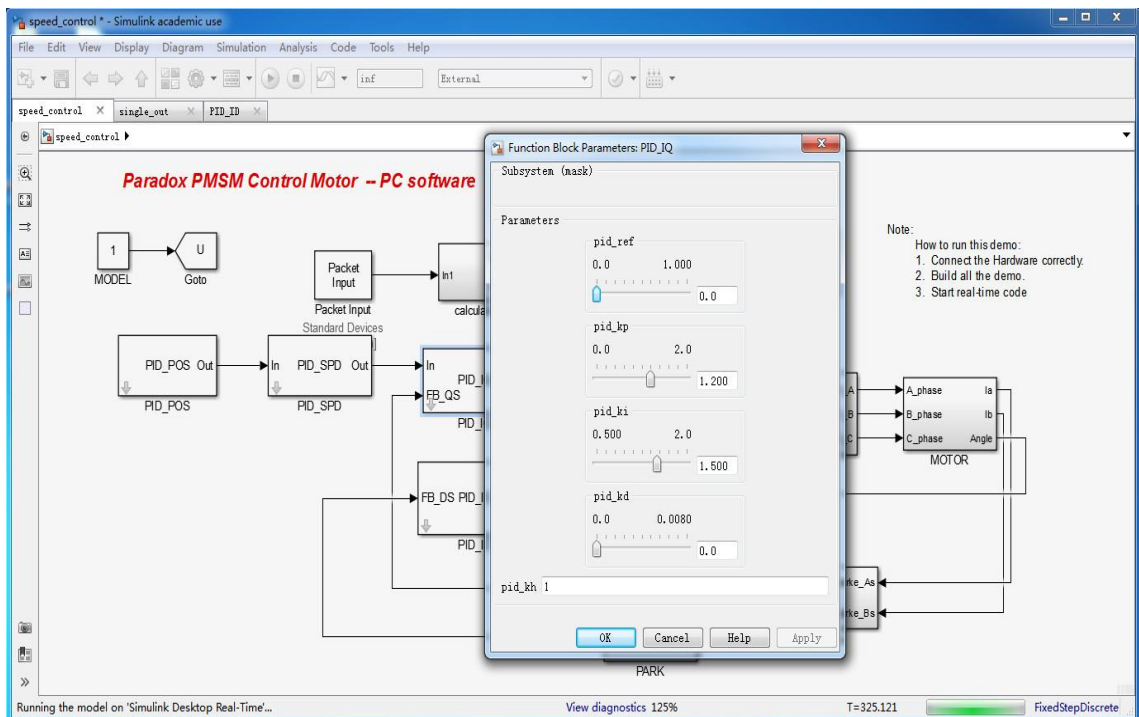


图 3-2-14 力矩模式参数设置面板

13. 打开 Speed 和 pos 模块可以监控力矩模式的速度和位置，如图 3-2-15 所示；

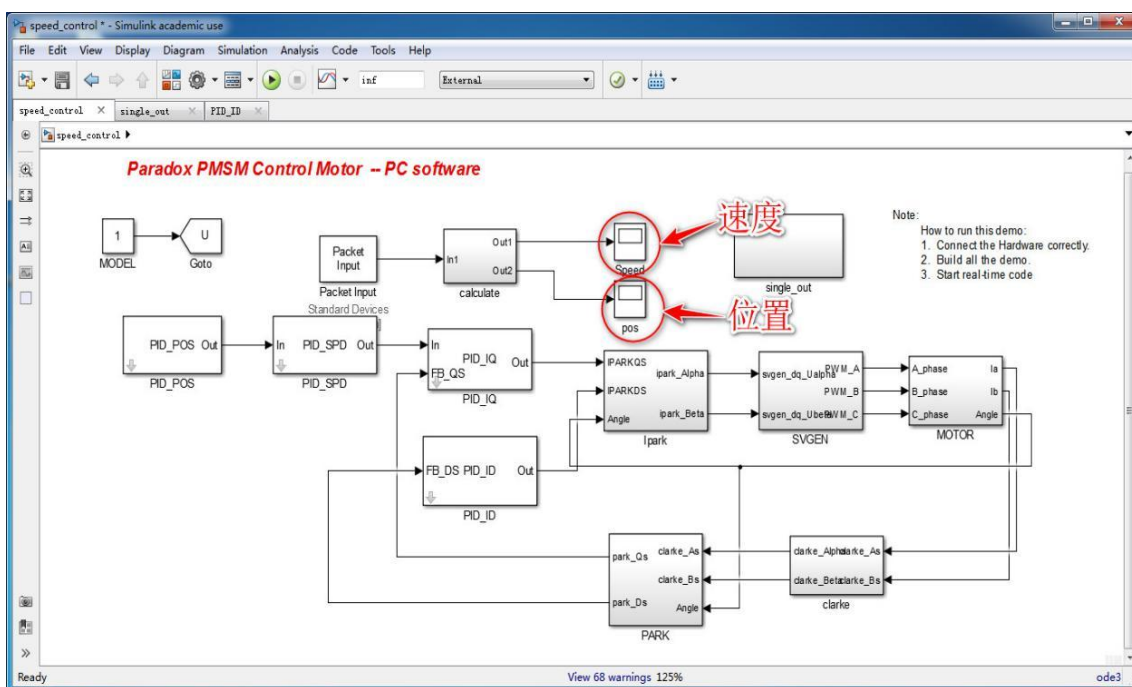


图 3-2-15 力矩模式速度和位置监控

14. 打开 Speed 模块，观察速度曲线，如图 3-2-16 所示。

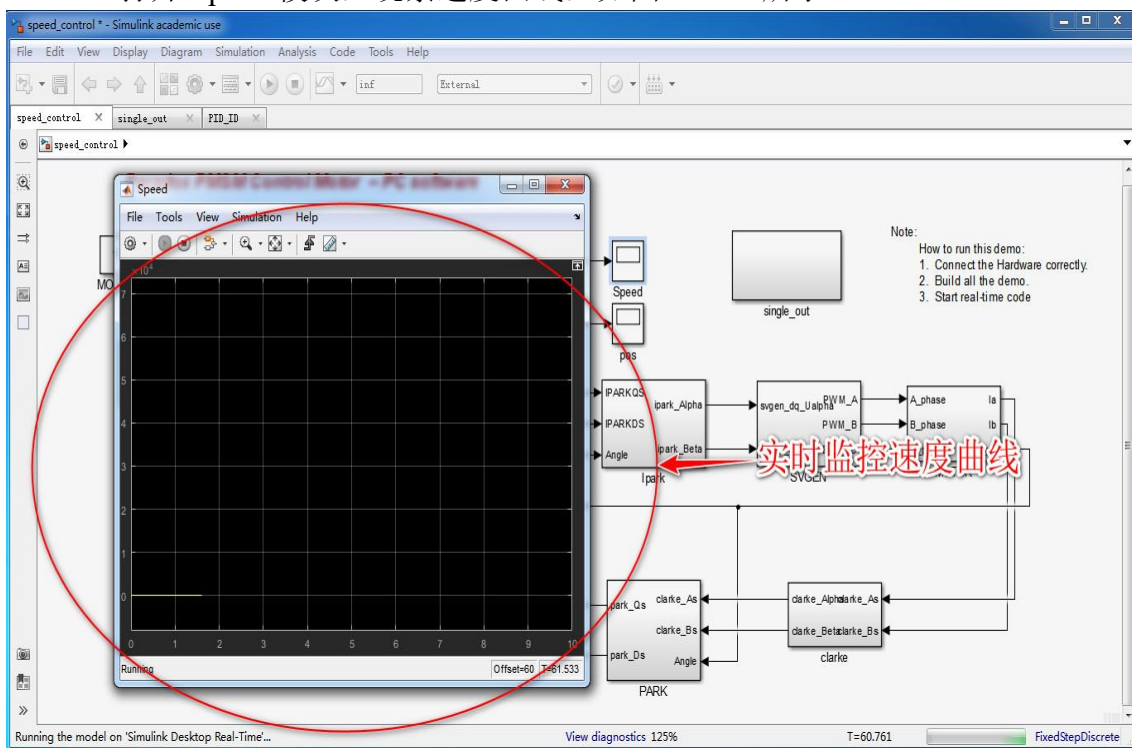


图 3-2-16 力矩模式速度实时监控

通过打开 speed 或 pos 模块观察速度位置变化情况，并保存几张速度或位置变化曲线。其中通过 view-style-调整图像背景颜色。

注意事项：在按下停止按钮前，若没有将 pid_ref 滑块置零，电机依然会继续运转，请一定要将 pid_ref 置零，再点击停止按钮。

4. 实验分析

4.1 直流无刷电机六步换向

1、理解和写出无刷直流电机换相的逻辑（逆时针/顺时针旋转时霍尔器件的状态，以及相应三相通电的方式）。

2、通过对 CCS 程序的工程导入、程序编译烧录、信号传输通道设置、电机工作模式选择，了解直流无刷电机的整个控制操作流程，实现对电机的运动控制，特别是运动速度大小的控制。该实验属于验证性实验，因此为体现实验效果的完成程度，需要记录保存几张不同 pid_ref 参考电流下的速度变化图像。pid_ref 可以动态持续的变动，相应的也能查看到对应速度的变化，因此每个实验组的同学得到图像应当不同，以此体现实验工作差异和实验完成程度。

3、实验涉及到部分源代码的设计，根据分析理解六步换向原理和现已给出的源代码，编写程序关键词 MOTOR_DIR==1 处的代码（项目可能已经给出），实现电机的相对反方向转动。

4.2 交流伺服电机 Clark 变换和 Park 变换

1. 根据 Clark 变换和 Park 变换的原理,写出永磁同步电机的 Park 和 Clark 变换公式。

2. 通过对 CCS 程序的工程导入、程序编译烧录、信号传输通道设置、电机工作模式选择了解伺服电机的整个控制操作流程，实现对电机的运动控制，特别是运动速度大小控制。该实验为验证性实验，通过实验操作了解交流伺服电机是如何实现位置的精确控制，以及速度的改变调整，能够对交流伺服电机的控制原理和技术平台有一定了解。为体现实验顺利的完成程度，通过速度 speed 或者位置 pos 观察框观察电机速度/位置随着 PID_ref 滑动模块的变化曲线，并在停止后保存图像，每组实验得到的曲线图应当不同，以此体现实验工作差异和实验完成程度。

附录 1 系统硬件介绍

1.1 系统构成

该实验装置由交流伺服电机、直流无刷电机、DSP 驱动板及驱动器等主要部件组成，结构如图 1-1、1-2 所示。

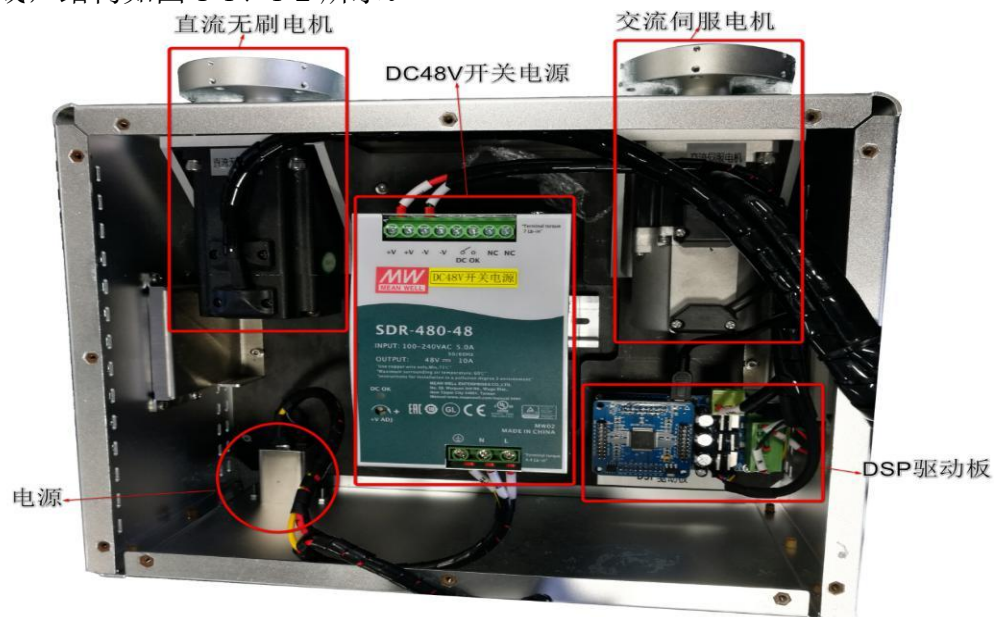


图 1-1 电机调试平台内部结构示意图



图 1-2 电机调试平台外部结构

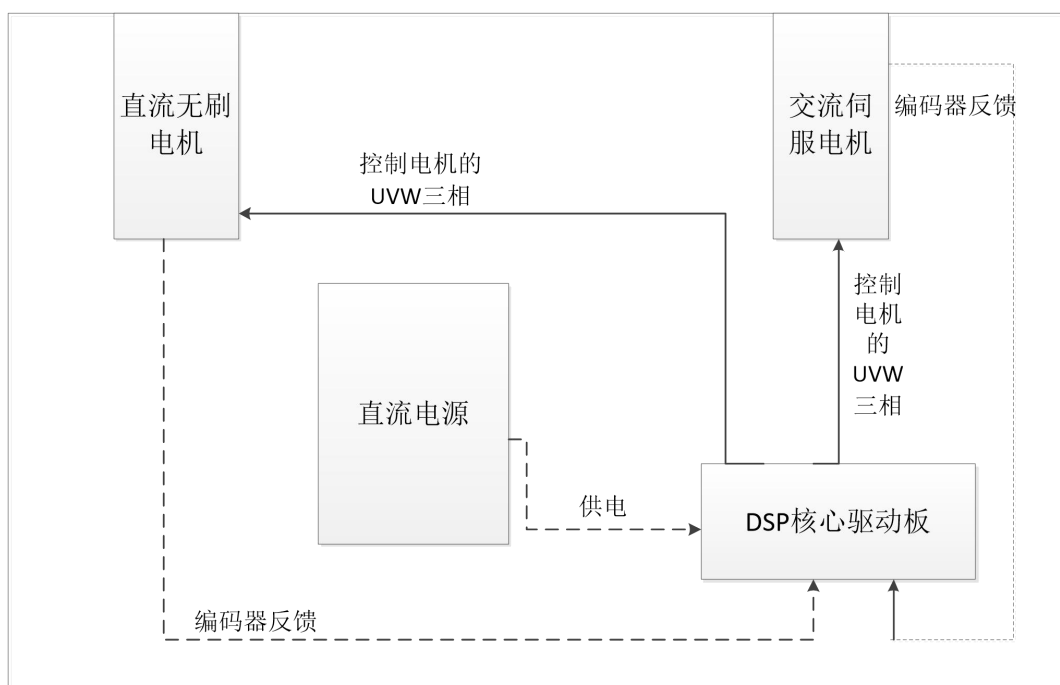


图 1-3 硬件连接原理框图

1.2 DSP 核心板

1.2.1 接线定义

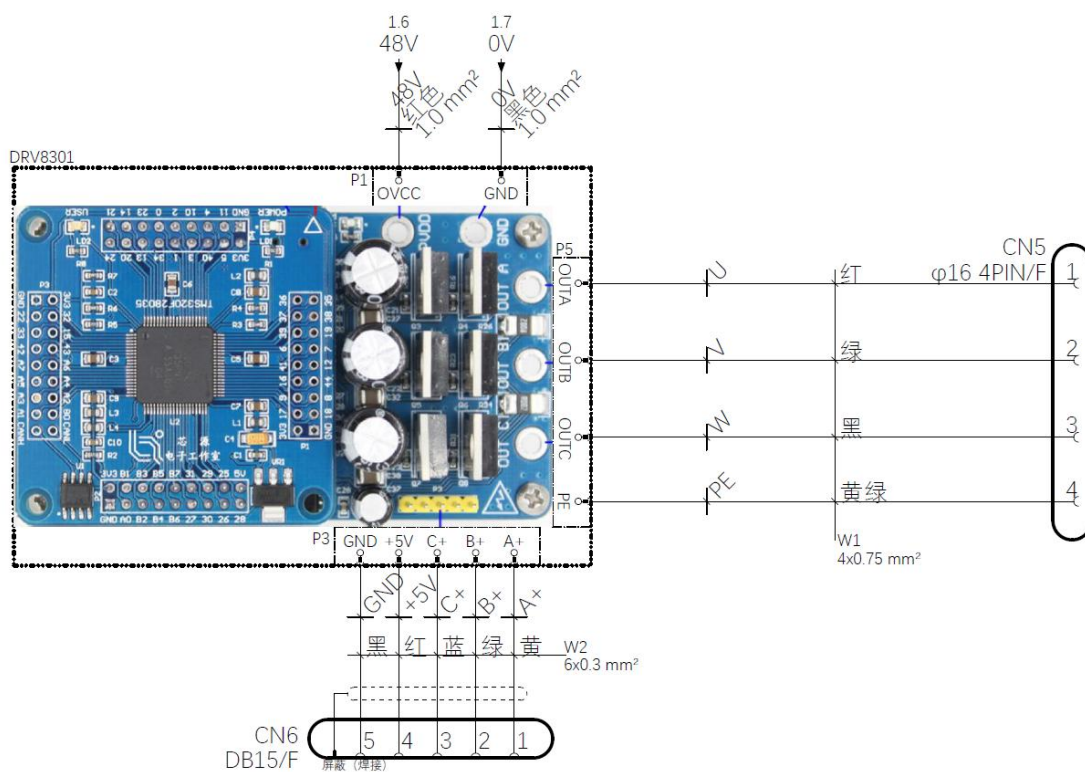


图 1-4 DSP 驱动板接线定义

1.2.2 参数

表 1-1 DSP 核心板参数

28035 核心板	
特点	32 位中央处理单元(CPU)
	60MHz 主频
	45 个复用通用输入输出(GPIO)引脚
	三个 32 位 CPU 定时器
	串行端口外设(一路 SCI/两路 SPI/一路 I2C/一路 LIN/一路 eCAN)
	14 路增强型脉宽调制器(ePWM)，可选高分辨率 PWM(HRPWM)
	一路增强型捕捉(eCAP)
	两路高分辨率输入捕获(HRCAP)
	一组增强型正交编码器脉冲(eQEP)
	16 路模数转换器 (ADC) 12 位 AD
备注	可编程控制律加速器(CLA)，32 位浮点算术加速器独立于主 CPU 之外的代码执行

表 1-2 驱动单元参数

8301 驱动板	
特点	工作电源电压范围为 8V-60V
	2.3A 沉电流和 1.7A 源门驱动电流能力
	具有可调节增益和抵消补偿的集成双重并联电流放大器。可选放大 10、20 倍、40 倍以及 80 倍
	3 或 6 路 PWM 输入的独立控制
	支持 100%的占空比的引导门驱动器
	可编程空载时间以避免外部 FET 直通
	外部 MOSFETs 的可编程过流保护
	能减少电磁干扰(EMI)的转换速率控制
备注	驱动器是由 28035 核心板和 8301 驱动板组合而成，驱动板能接一组霍尔或一组编码器，可以驱动 500W 以下的无刷直流电机和永磁同步电机。驱动板供电电压 12-60V,目前最大电流 5A，可以搪锡改变最大电流到 20A。

1.2.3 功能特点

DSP 芯片，也称数字信号处理器，是特别适用于数字信号处理运算的一种微处理器，其主要应用是实时快速地实现各种数字信号处理算法。根据数字信号处理的要求，DSP 芯片一般具有如下主要特点：

1. 在一个指令周期内可完成一次乘法和一次加法；
2. 程序和数据空间分开，可以同时访问指令和数据；
3. 片内具有快速 RAM，通常可通过独立的数据总线在两块中同时访问；

4. 具有低开销或无开销循环及跳转的硬件支持;
 5. 快速的中断处理和硬件 I/O 支持;
 6. 具有在单周期内操作的多个硬件地址产生器;
 7. 可以并行执行多个操作;
 8. 支持流水线操作, 使取指、译码和执行等操作可以重叠执行。当然, 与通用微处理器相比, DSP 芯片的其他通用功能相对较弱。
- 本系统采用的为 DSP-F28035 核心板, 如图 1-5 所示。



图 1-5 DSP 核心驱动板示意图

该系列微控制器为 C28x™ 内核和控制律加速器(CLA)供电, 此内核和 CLA 与低引脚数量器件中的高集成控制外设向耦合。该系列的代码与以往基于 C28x 的代码相兼容, 并且提供了很高的模拟集成度。一个内部电压稳压器允许单一电源轨运行。对 HRPWM 模块实施了改进, 以提供双边缘控制(调频)。增设了具有内部 10 位基准的模拟比较器, 并可直接对其进行路由以控制 PWM 输出。ADC 可在 0V 至 3.3V 固定全标度范围内进行转换操作, 并支持公制比例 VREFHI /VREFLO 基准。ADC 接口专门针对低开销/低延迟进行了优化。

1.3 直流无刷电机

1.3.1 接线定义

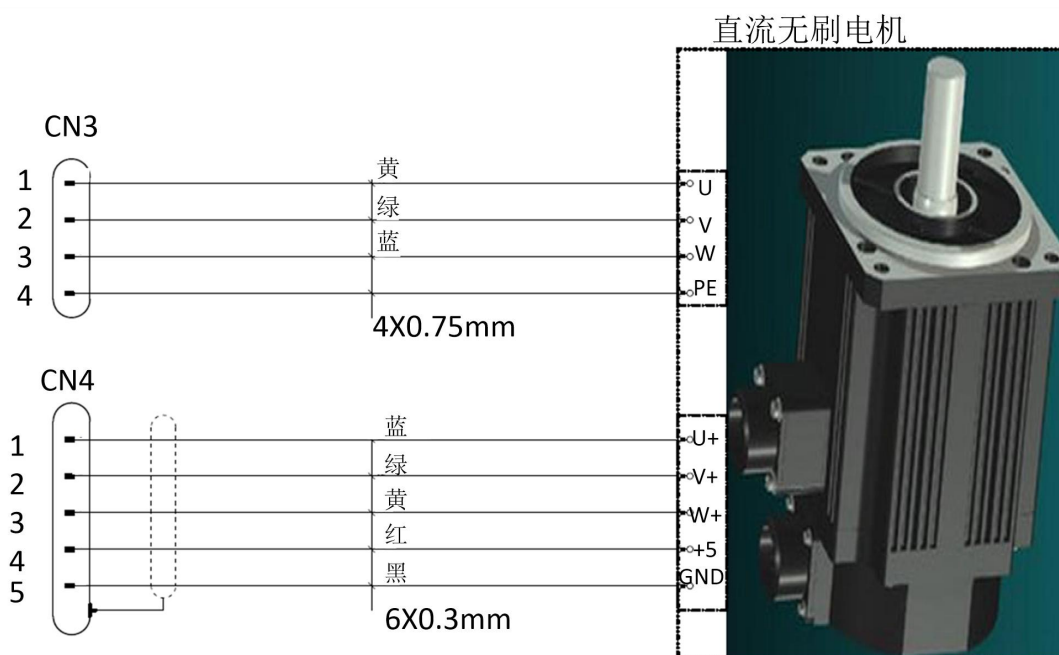


图 1-6 直流无刷电机接线定义

1.3.2 参数

表 1-3 直流无刷电机参数

电机型号	DT60BL100-230
额定功率(W)	200
额定扭矩(N·m)	0.65
瞬时最大扭矩(N·m)	1.75
额定/最大转速(r/min)	3000/4000
电机转动惯量(kg·cm ²)	0.48
每秒最大功率(kw/s)	21.5
电阻(Ω)	0.22
电感 (mh)	0.29
额定电流(A)	10.5
瞬时最大电流(A)	31
极对数 (对)	4
反电动势常数 (V/Krpm)	4.2

1.3.3 结构特点

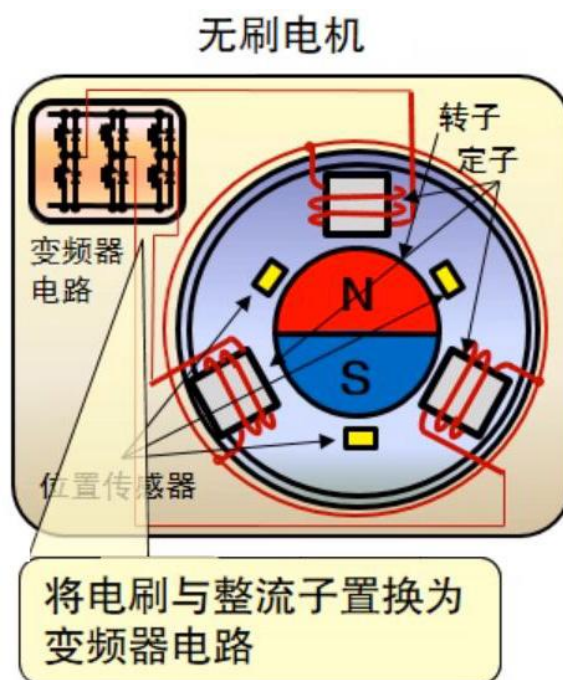


图 1-7 直流无刷电机模型

直流无刷电机由电动机主体和驱动器组成，是一种典型的机电一体化产品。电动机的定子绕组多做成三相对称星形接法，同三相异步电动机十分相似。电动机的转子上粘有已充磁的永磁体，为了检测电动机转子的极性，在电动机内装有位置传感器。驱动器由功率电子器件和集成电路等构成，其功能是：接受电动机的启动、停止、制动信号，以控制电动机的启动、停止和制动；接受位置传感器信号和正反转信号，用来控制逆变桥各功率管的通断，产生连续转矩；接受速度指令和速度反馈信号，用来控制和调整转速，提供保护和显示等等。

直流无刷电机的特点：

1. 可替代直流电机调速、变频器+变频电机调速、异步电机+减速机调速；
2. 具有传统直流电机的优点，同时又取消了碳刷、滑环结构；
3. 可以低速大功率运行，可以省去减速机直接驱动大的负载；
4. 体积小、重量轻、出力大；
5. 转矩特性优异，中、低速转矩性能好，启动转矩大，启动电流小；
6. 无级调速，调速范围广，过载能力强；

1.4 直流无刷电机

1.4.1 接线定义

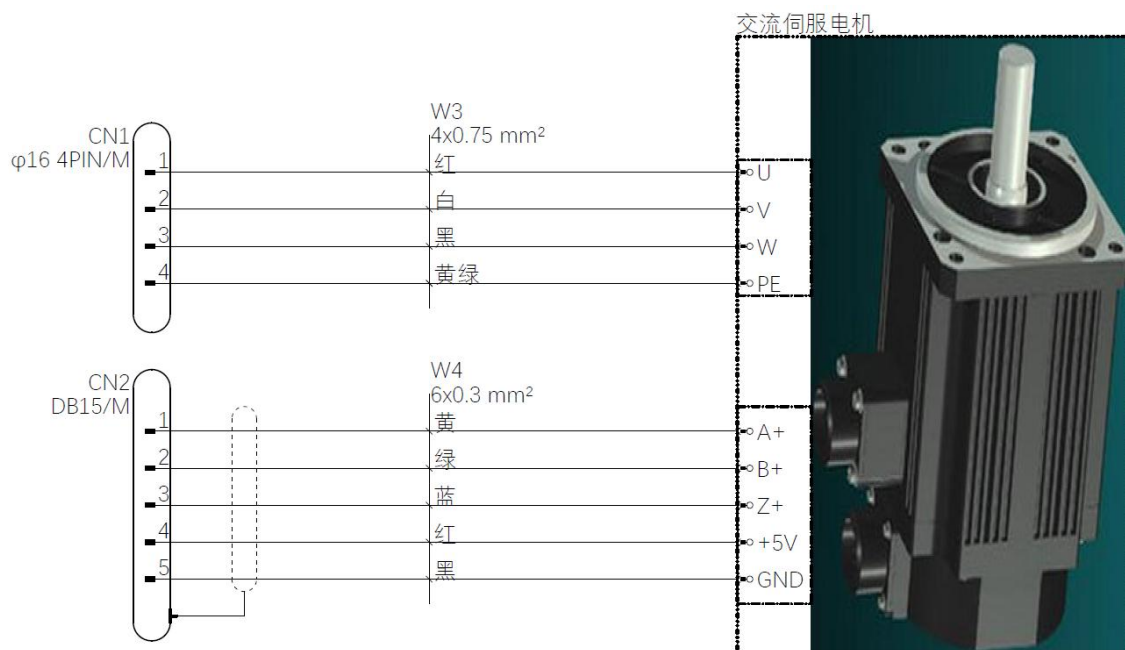


图 1-8 电机接线定义

1.4.2 参数

表 1-4 交流伺服电机参数

电机型号	TS4607 N1680 E200
额定功率(W)	200
额定扭矩(N.m)	0.64
瞬时最大扭矩(N.m)	1.91
额定/最大转速(r/min)	3000/5000
Encoder 电机转子惯量(kg.m ²)	0.19×10 ⁻⁴
Resolver 电机转子惯量(kg.m ²)	0.2×10 ⁻⁴
每秒最大功率(kw/s)	21.5
机械常数(ms)	0.9
额定电流(A)	1.7
瞬时最大电流(A)	5
极对数(对)	4
绝缘等级	F 级
使用温度(°C)	0—40
保存温度(°C)	-10—85
耐震性	5G 以下
使用湿度	85%RH 以下(不结露)
IP 等级	标准 IP42(可支持 IP65)

1.4.3 结构特点

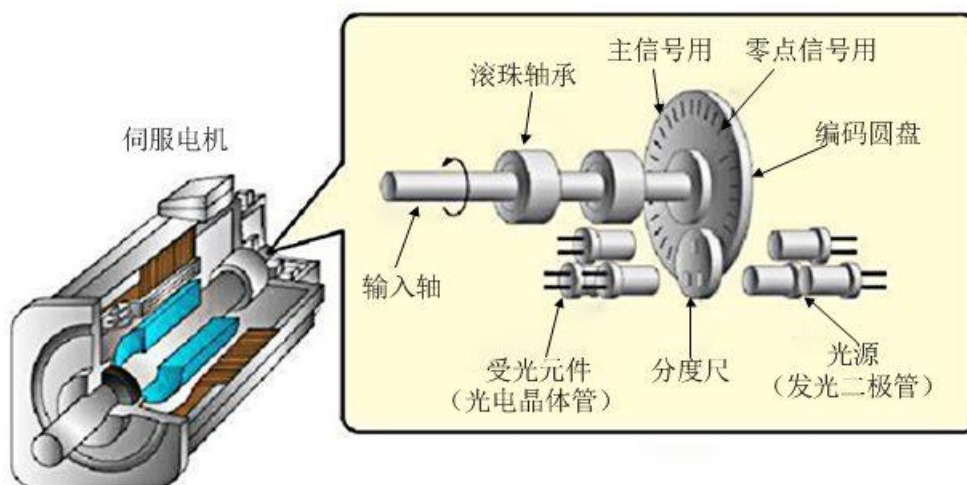


图 1-9 交流伺服电机基本结构图

如上图 1-9 所示，交流伺服电机主要由定子和转子构成。定子铁心通常用硅钢片叠压而成。定子铁心表面的槽内嵌有两相绕组，其中一相绕组是励磁绕组，另一相绕组是控制绕组，两相绕组在空间位置上互差 90° 电角度。工作时励磁绕组与交流励磁电源相连，绕组上电，电机运动。

输入轴上装有玻璃制编码器圆盘，圆盘上印刷有能够遮住光的黑色条纹。圆盘两侧有一对光源与受光元件，中间有一个分度尺元件。圆盘转动时，遇到玻璃透明的地方光就会通过，遇到黑色条纹光就会被遮住。受光元件将光的有无转变为电信号之后就成为脉冲（反馈脉冲）。“圆盘上条纹的密度=伺服电机的分辨率”亦即“每转脉冲数”，根据条纹可以掌握圆盘的转动量。

附录 A CCS 工程导入

1. 打开电脑桌面 DSP 编译软件 CCS，出现位置空间选择，直接电机确定。选择 Project，点击 Import CCS Projects，如下图 A-1 所示；

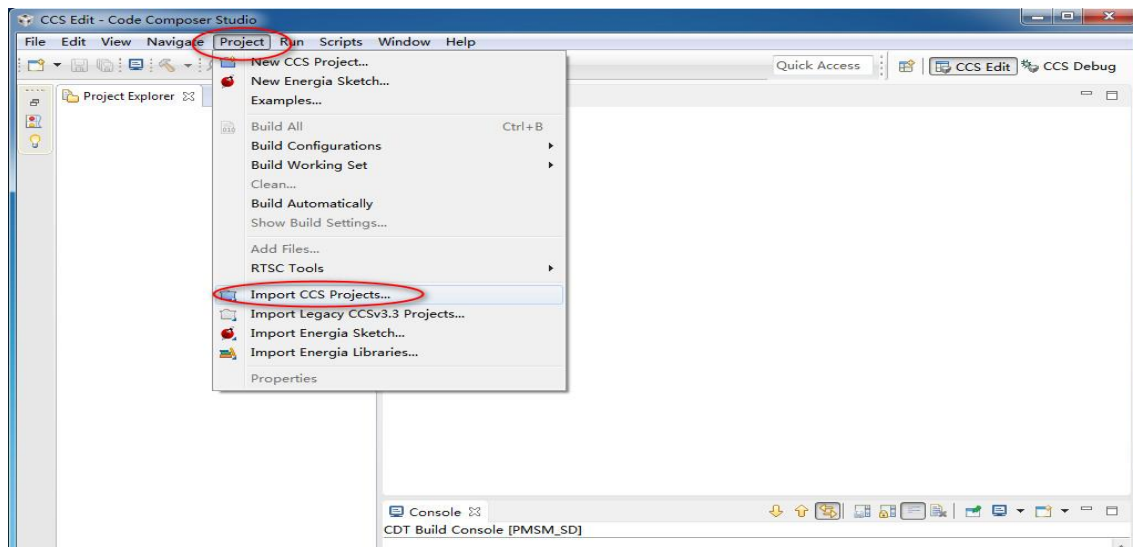


图 A-1 打开添加选项

14. 点击 Browse, 在目录 **C:\ProgramFiles\MATLABR2015b\workspace\pwm** 下中先导入第一个小实验工程 **BLDC**，第二个小实验导入 **PMSM**。选择 pwm(提供直流无刷电机实验 BLDC，提供交流伺服电机实验 PMSM) 文件夹中的 **BLDC_SD/PMSM_SD**，然后点击确定，如图 A-2 所示。**注意该案例是导入第二个小实验的过程，第一个小实验与此只是名字不同，其他过程都相同。**

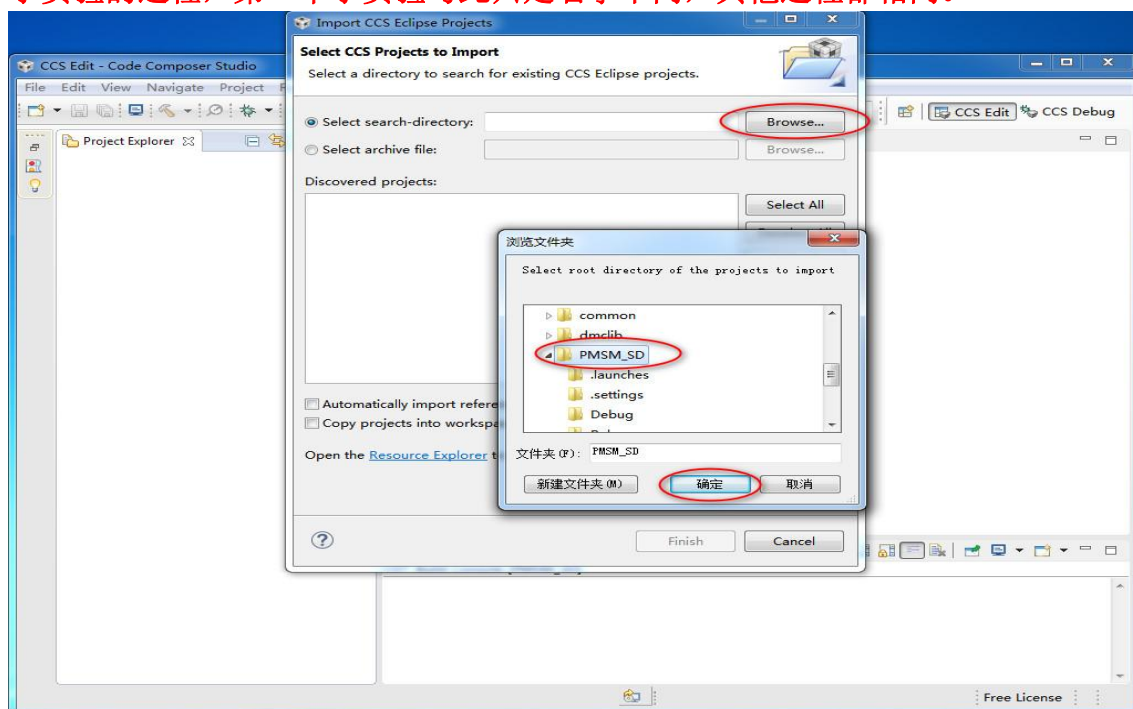


图 A-2 添加工程示意图

2. 点击 Finish，完成工程导入，如图 A-3 所示；

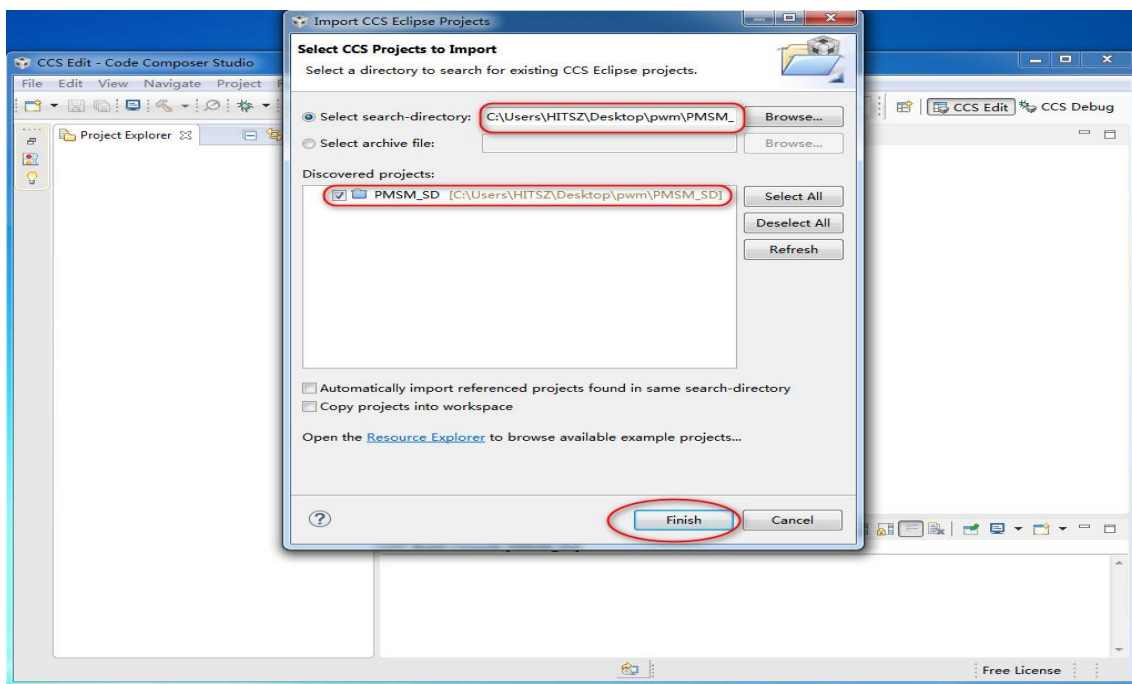


图 A-3 添加工程完成示意图

3. 工程导入成功，如图 A-4 所示；

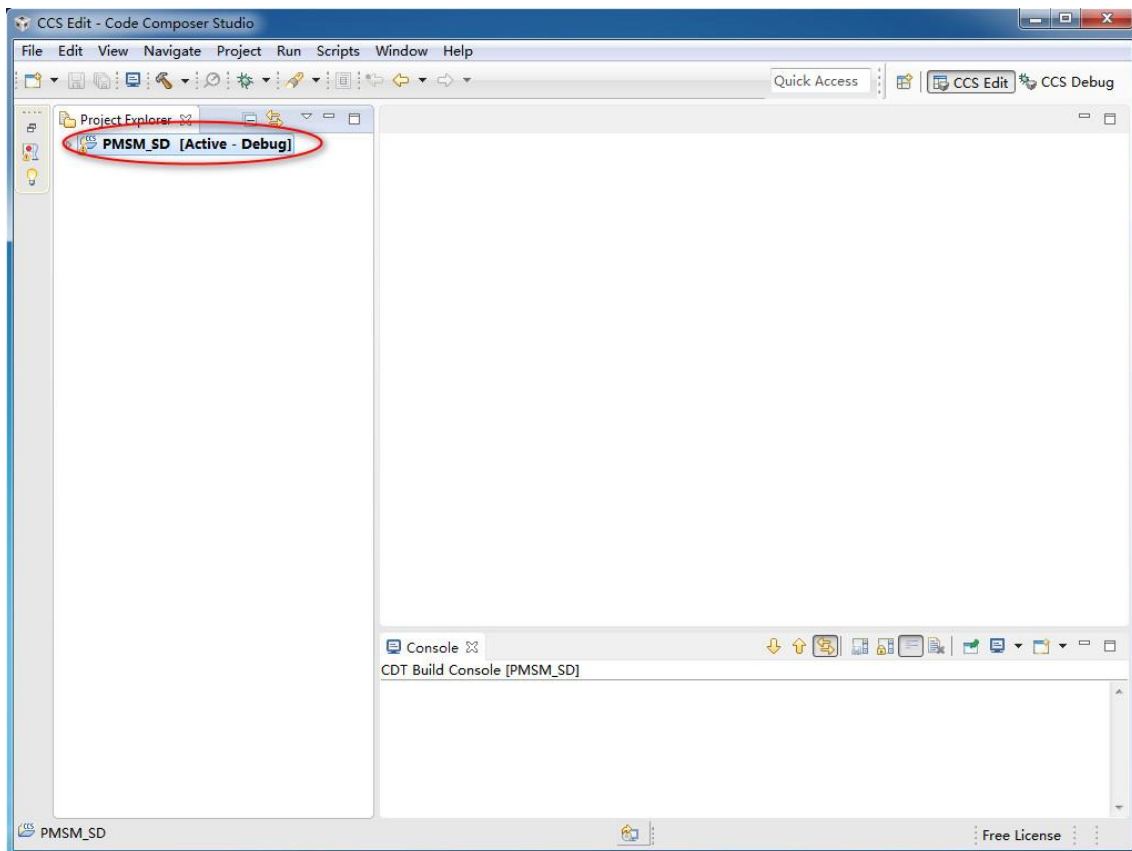


图 A-4 工程导入成功界面

4. 选择 Release 模式，如图 A-5 所示；

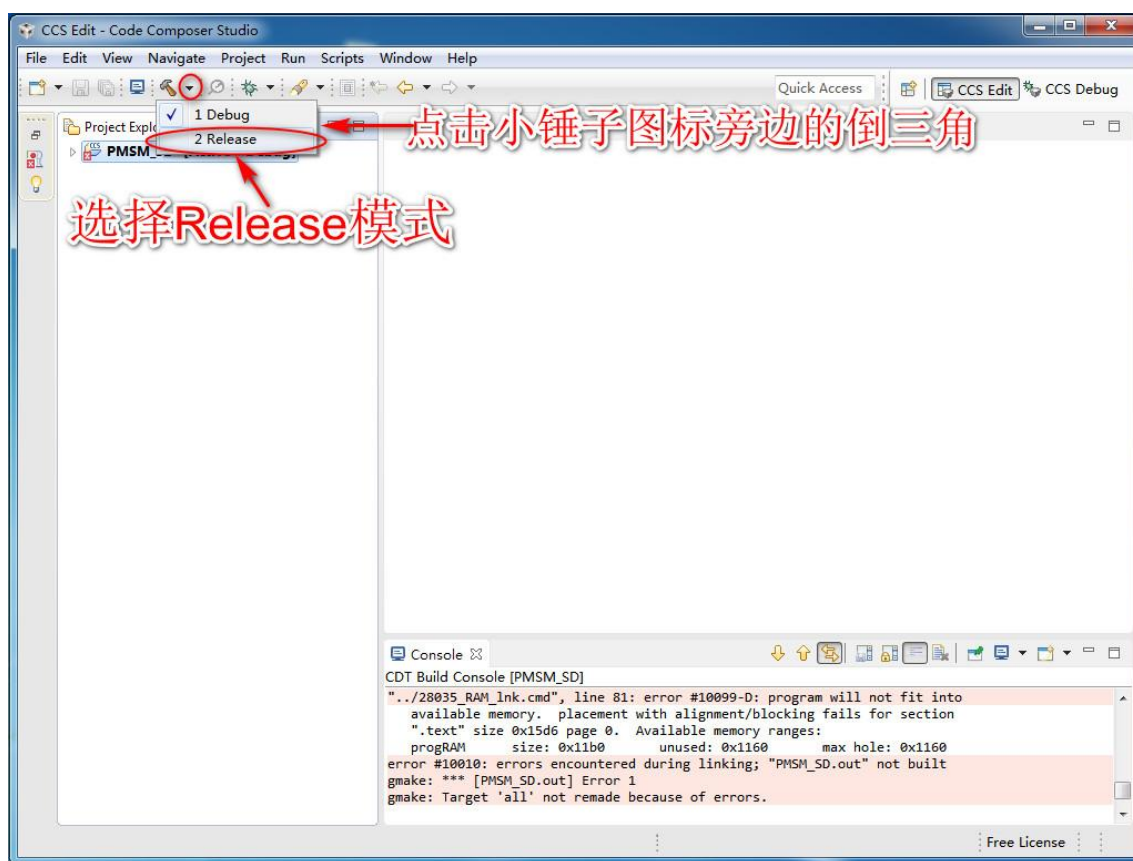


图 A-5 选择 Release 模式

5. 进入 Release 模式后会报错，可以忽略报错继续操作，右键单击导入的工程文件，选择 Properties 选项，如 A-6 所示；

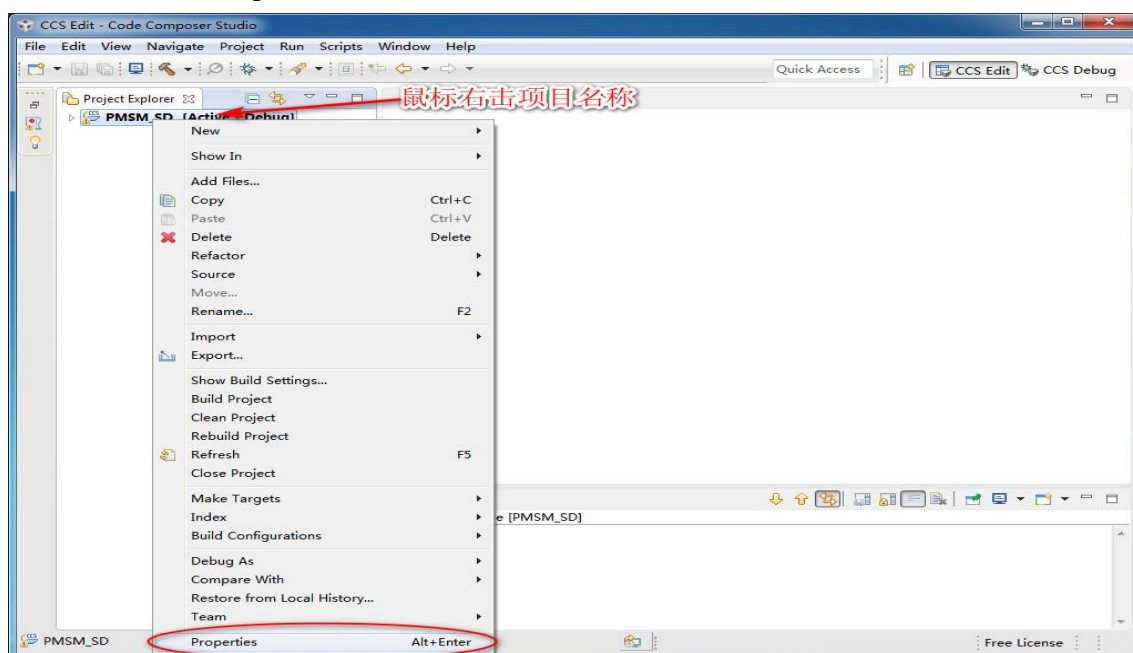


图 A-6 选择 Properties 选项

6. 选择 Include Options 选项，如图 A-7 所示；

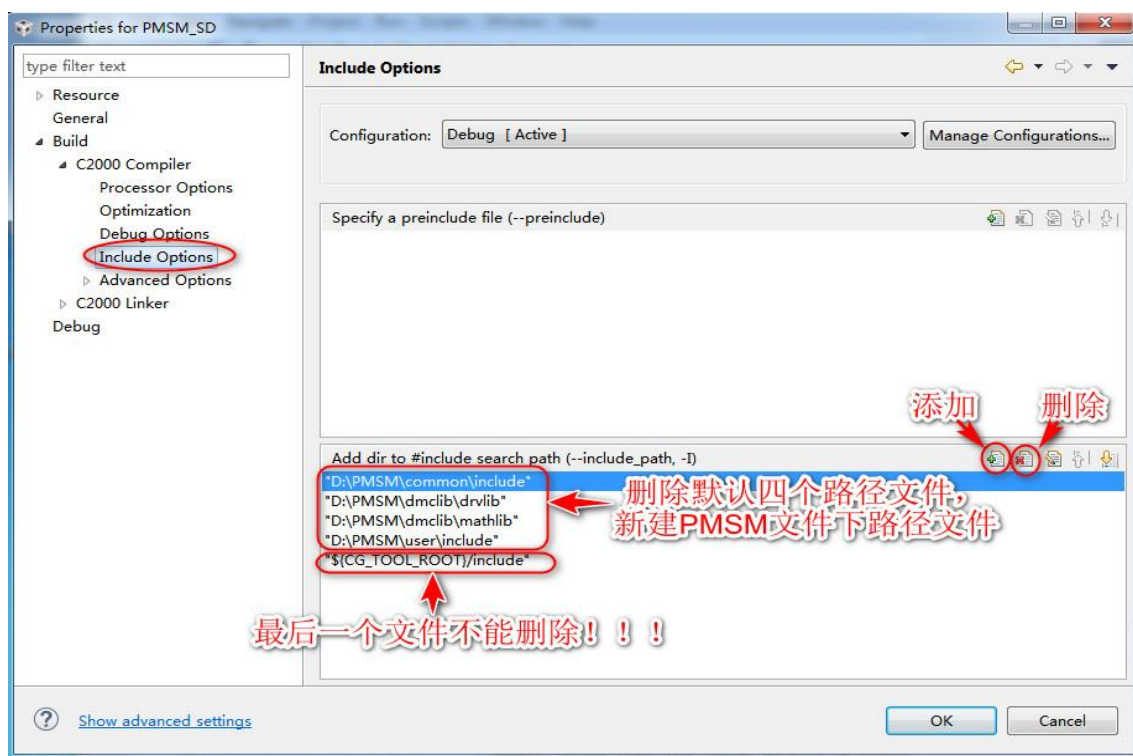


图 A-7 配置系统文件

8. 删除前四个默认路径（注意最后一个文件不能删除），如图 A-8 所示；

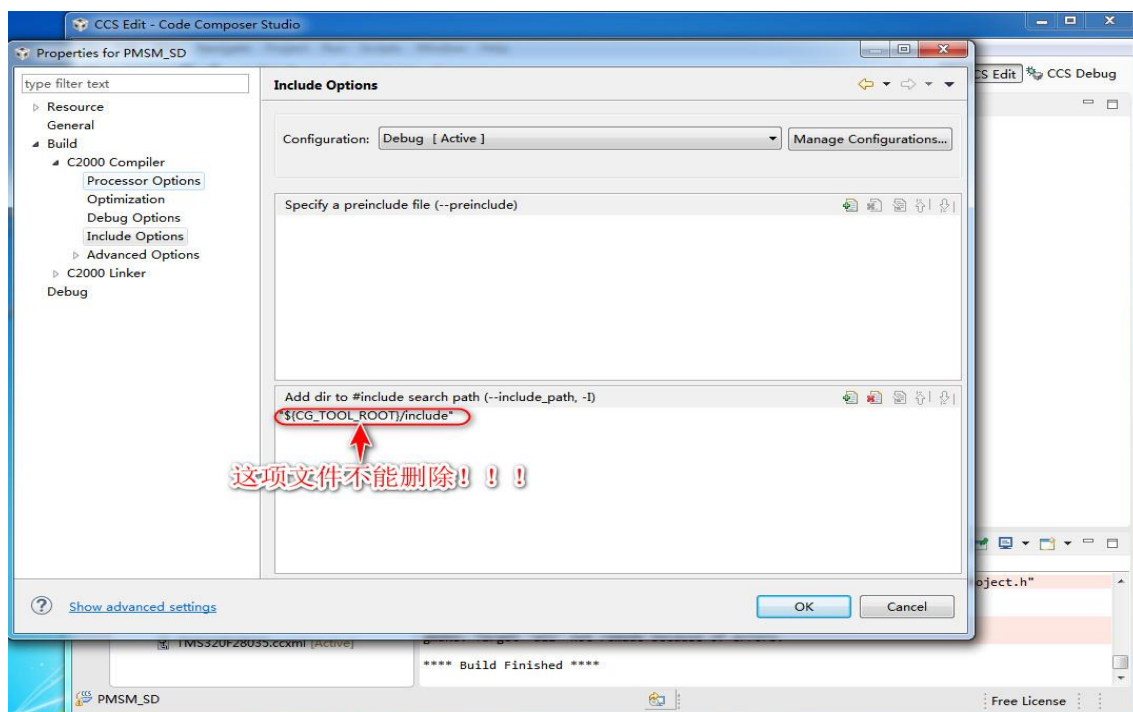


图 A-8 删除 4 个默认路径

9. 添加新路，添加目录 **C:\ProgramFiles\MATLAB\R2015b\workspace\pwm\BLDC** (第一个实验是 BLDC, 第二个小实验是 PMSM) 文件夹下面的四个路径: \user\include、\common\include、\dmclib\drvlib、\dmclib\mathlib, 如图 A-9 所

示；注意是在 **pwm** 文件下的 **BLDC** 或 **PMSM** 中的子文件中进一步添加。

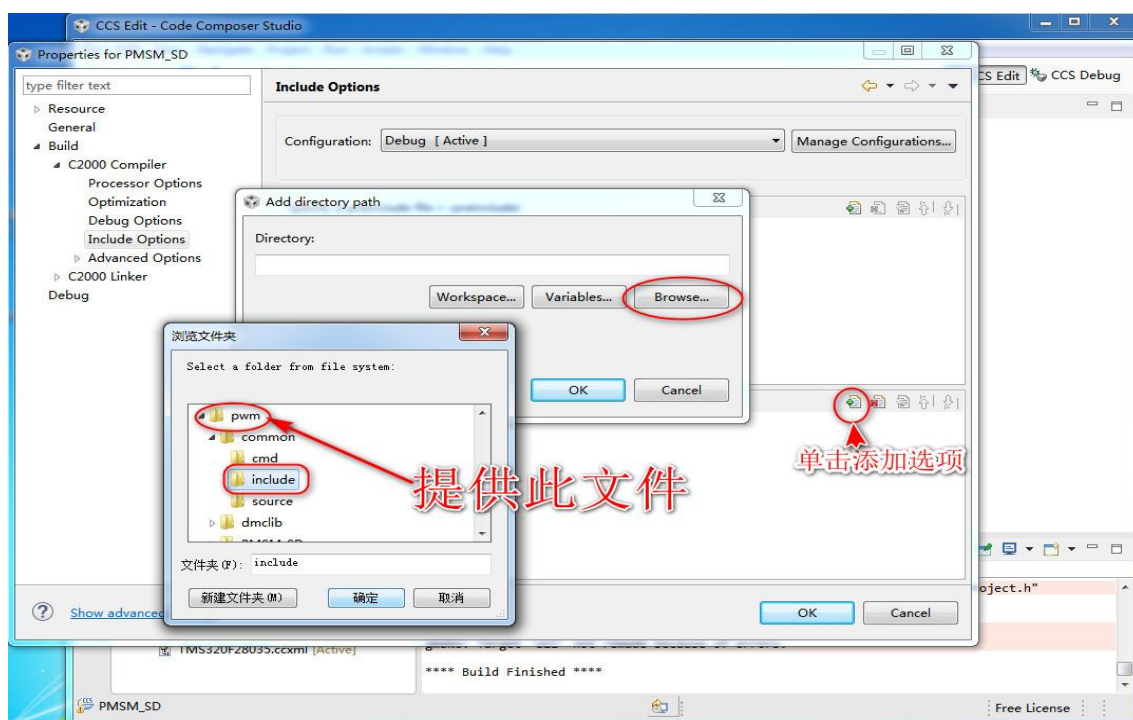


图 A-9 添加 PMSM 文件

10. 添加完成后，单击 OK（注：必须是全英文路径），如图 A-10 所示；

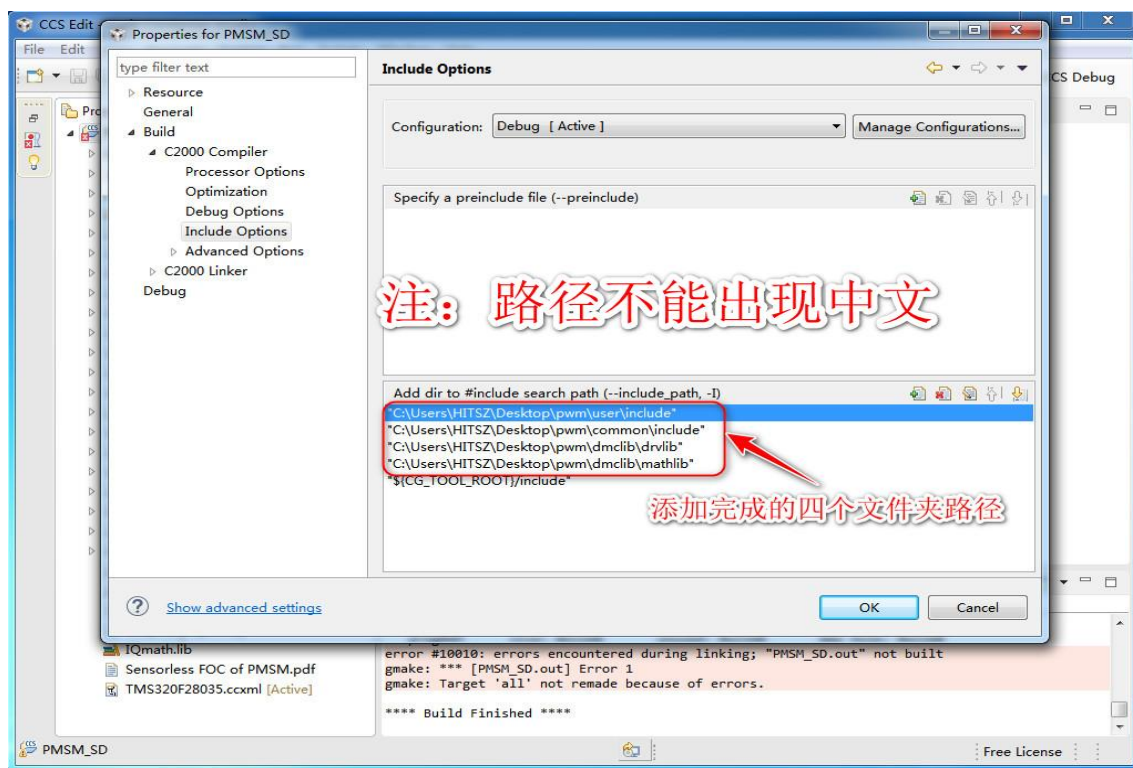



图 A-10 添加完成

11. 点击编译工具图标旁边的下三角图标 ，选择 Release 模式，等待编译

完成，如图 A-11 所示；

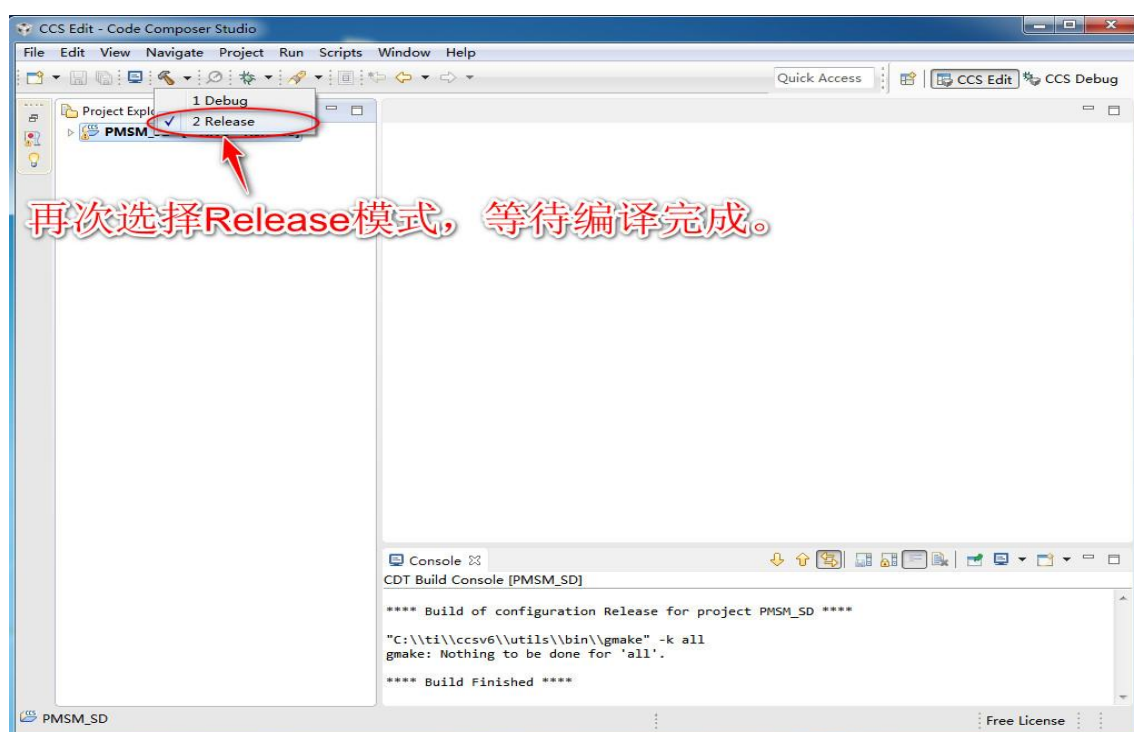


图 A-11 重新编译

12. 编译完成后，单击仿真图标，进入仿真模式，如图 A-12 所示；

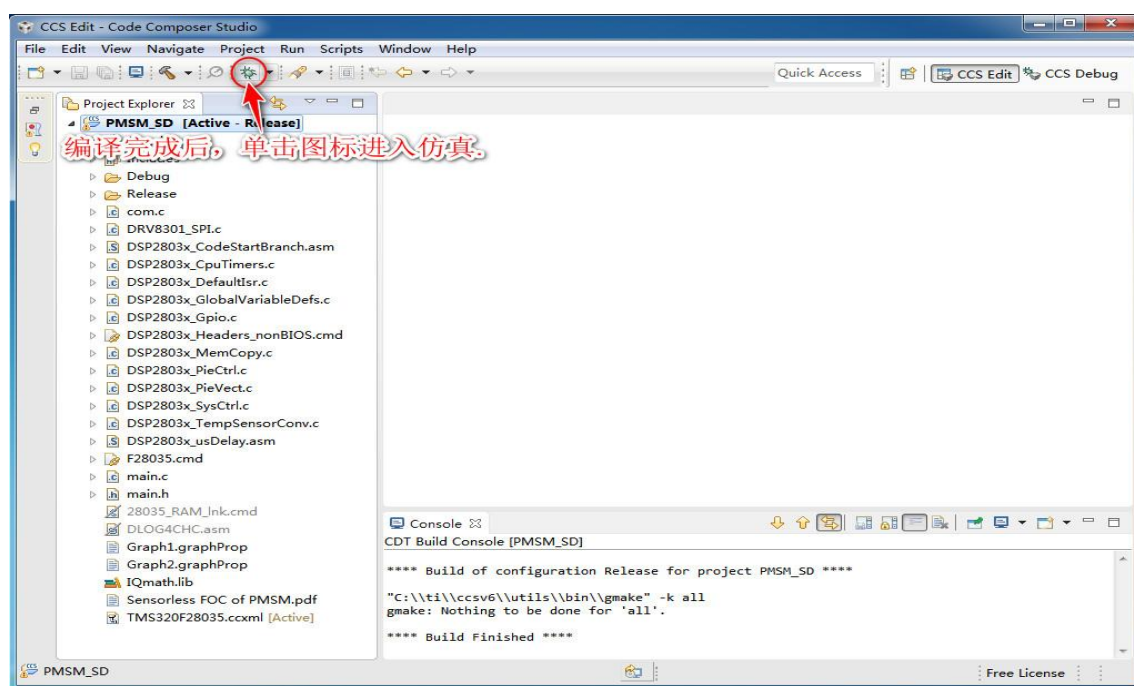


图 A-12 进入仿真模式

13. 编译完成后，自动进入 Debug 测试界面，其三个符号分别表示运行-停止-退出功能，如图 A-13 所示。此时点击运行按钮，让程序持续处于运转状态，才能进行接下来的操作。回到第一个实验的实验步骤 3，或者是第二个实验的步骤 3，根据实际实验顺序操作。注意不要拔掉 USB 线，如果程序如下警告，重新按照指导操作一次。

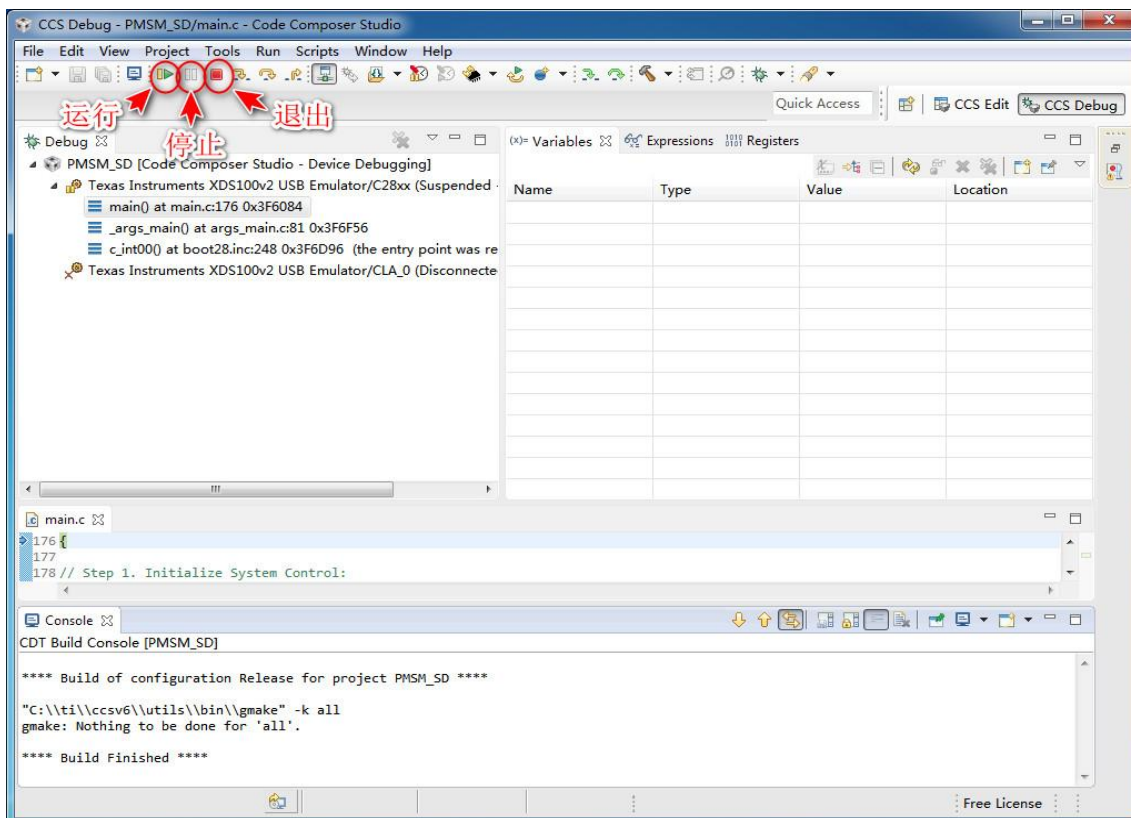


图 A-13 Debug 仿真界面

附录 B 上位机实时内核设置

1. 点击 Simulation，在下拉菜单中选择 Model Configuration Parameters（Simulation 参数设置），如图 B-1 所示。

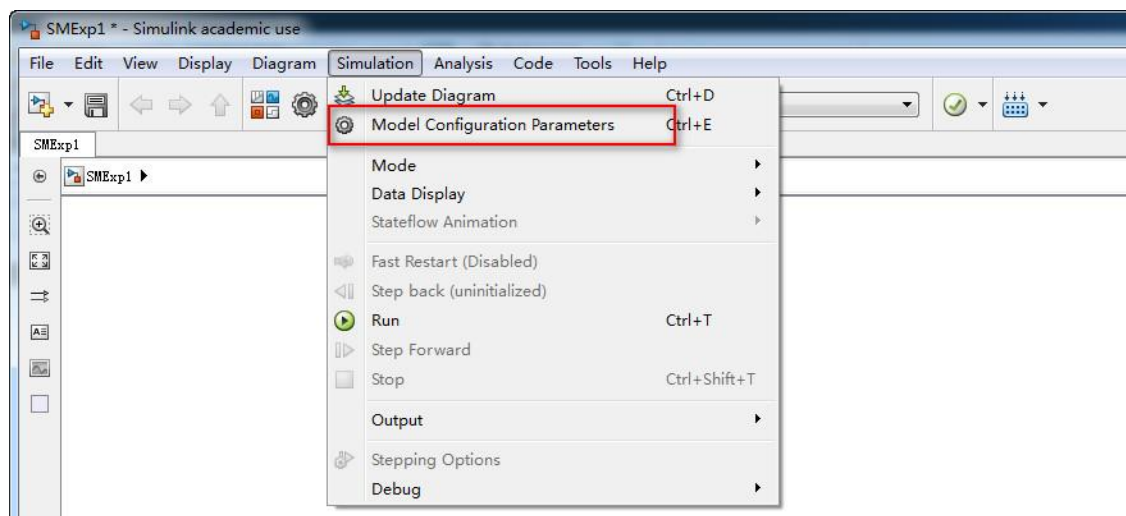


图 B-1

2. 点击左侧菜单 Solver，设置仿真开始时间 Start time 为 0，结束时间 Stop time 为 inf；Solver options 菜单下，Type 选 Fixed-step，求解器 Solver 选 ode1（Euler）算法，设置 Fixed-step size 为 0.01，如图 B-2 所示。

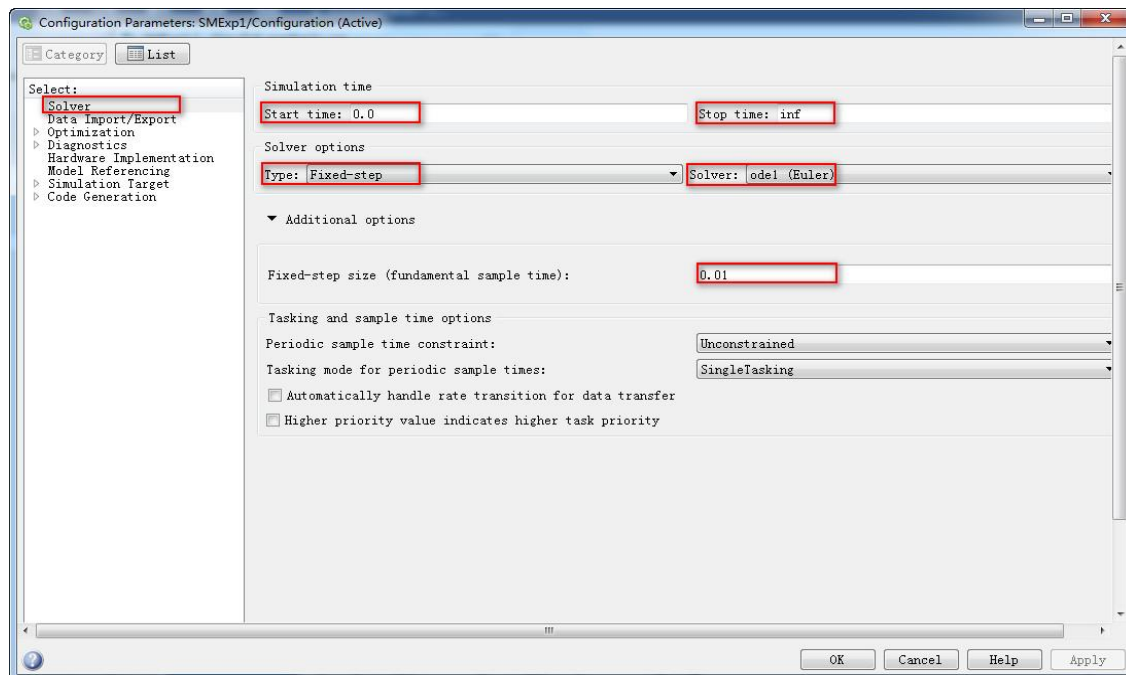


图 B-2

3. 点击左侧 Code Generation, 点击 Browse, 选择 sldrt.tlc (配置实时内核), 点击 OK 完成设置, 如图 B-3 所示。

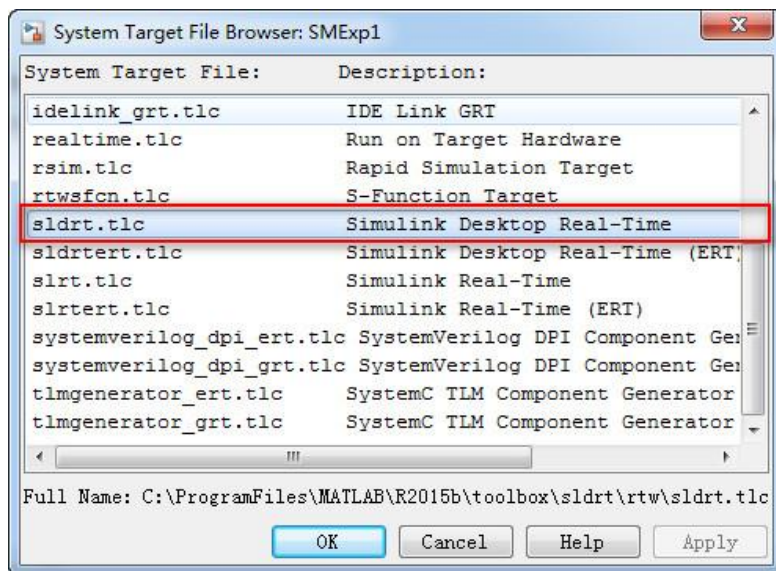
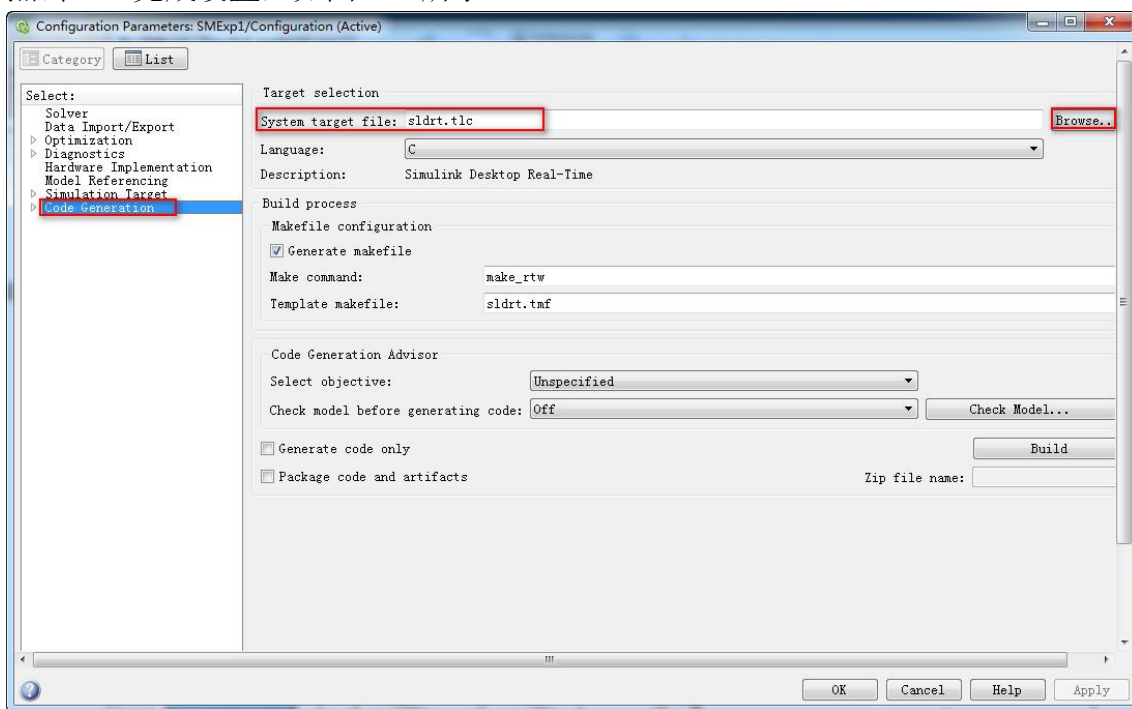


图 B-3

4. 在 Simulink 界面，设置运行模式为 External，如图 B-4 所示。

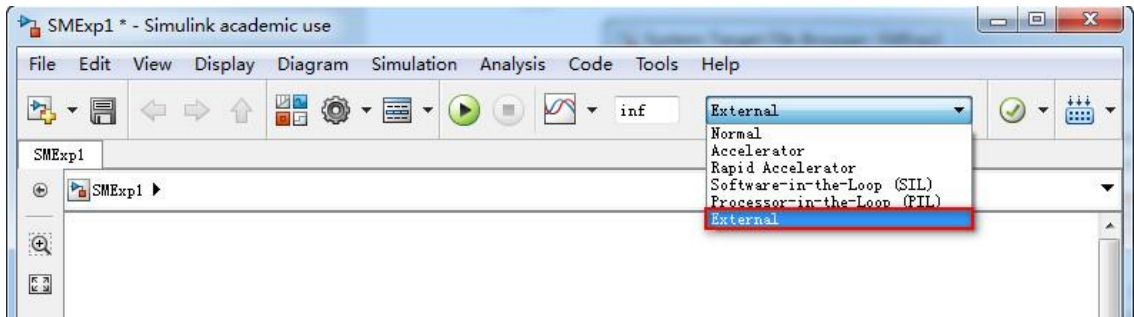


图 B-4