

3 控制系统建模仿真

3.1 永磁同步电机参数

要模拟永磁同步电动机的直接转矩控制使用 Matlab/ Simulink 软件。仿真如图 2.5 所示。在 Matlab/Simulink 软件中进行的仿真如图 2.6 所示。电动机永磁体的参数在表 3.1 中^[20]。在仿真中，永磁同步电动机使用凸极。在此仿真中，使用可变步长 ode23tb 算法。ode23tb 是 Matlab/Simulink 中模型为连续状态的求解器。

表 3.1 永磁同步电动机数据模拟

转子转速的参考 (W_r^*)	600 rpm
直流侧电压 (V_{dc})	311 V
定子电阻 (R)	1.2 Ω
极对数 (Pn)	4
定子电感 (L_s)	8.5 mH
磁链 (ψ_f)	0.175 Wb
转动惯量 (J)	0.008 Kg.m ²

根据表 3.1 中的参数，图 3.1 中的仿真中应用了永磁参数。

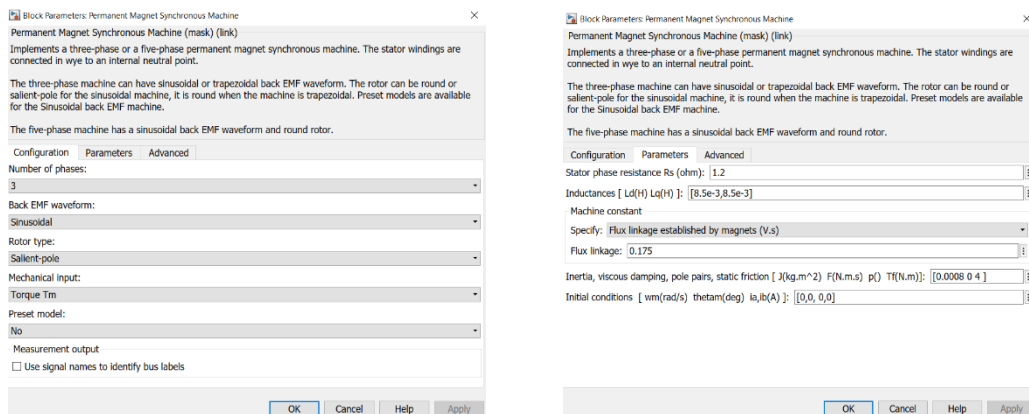


图 3.1 Matlab/Simulink 永磁同步电机参数

从图 3.2 中可以看出，三相 V-I 模块测量功能用于测量电路中的瞬时三相电压和电流。与三相元件串联时，可以选择要使用三个相接地或相间峰值电压和电流。在用于接地的三相 V-I 测量中，数学模型可以描述为

$$V_{abc}(pu) = \frac{V_{phase\ to\ ground}\ (V)}{V_{base}\ (V)} \quad (3.1)$$

其中

$$V_{base}(V) = \frac{V_{nom}\ (V_{rms})}{\sqrt{3}} \cdot \sqrt{2} \quad (3.2)$$

此仿真中的逆变器使用绝缘栅双极型晶体管（IGBT）。工业上首选使用的逆变器类型为 MOSFET 和 IGBT。IGBT 在电机驱动和中低功率范围的其他应用中具有广泛的接受度，其他逆变器的其他类型和特性请参见表 3.2^[5]。

表 3.2 设备电源和交换功能

设备	动力能力	切换速度
BJT	中	中
GTO	高	低
IGBT	中	中
MOSFET	低	高
THYRISTOR	高	低

3.2 直接转矩控制参数

为了使用 Clark 变换将三相（abc）的电压和电流变为两相（αβ），Clark 变换的数学模型使用等式（2.18）Clark 变换的方程式变为图 3.3 中的仿真。将电压和电流 αβ 相结合是根据等式（2.12）和（2.13），可以在图 3.4 的通量定子内部的磁通定子中看到。

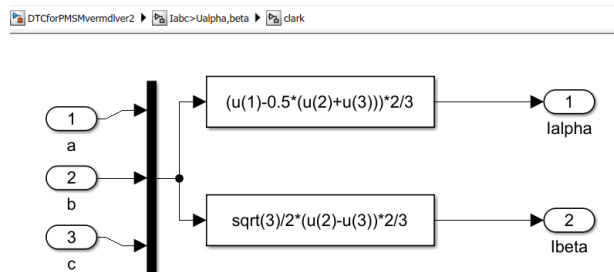


图 3.3 Clark 变换

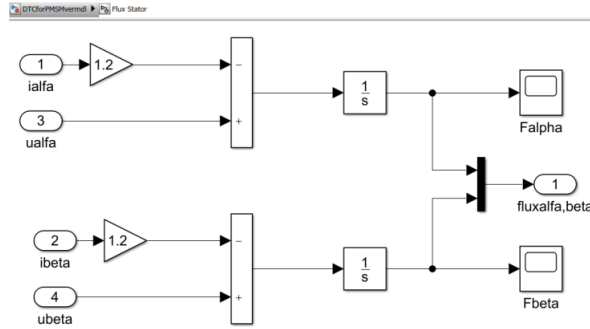


图 3.4 定子磁链计算的数学模型

在图 3.2 中，Sector 块是 S-Function。Sector 是计算 Sector 的通量位置，并使用 Clark 逆函数计算。m-文件 S-Function 的程序可以在下面看到

```
function [sys,x0,str,ts] = sector(t,x,u,flag) % 以下概述了 S-Function 的一般
                                          % 结构，S-Function 名字为 Sector
switch flag,
case 0, % 该声明 Switch 功能
[sys,x0,str,ts]=mdlInitializeSizes;
% Initialization %
case 3,
sys=mdlOutputs(t,x,u); %计算输出
case {1,2,4,9},
sys=[]; %未使用的 Flags
% Unexpected flags %
otherwise
error(['Unhandled flag = ',num2str(flag)]); %错误处理
end % sfuntmpl 结束
```

```

%=====
% mdlInitializeSizes
%=====
function [sys,x0,str,ts]=mdlInitializeSizes
sizes = simsizes; % 调用 simsizes 作为 sizes 结构，
                  % 将其填充并将其转换为 sizes 数
                  % 组

%用初始化信息加载 Sizes 结构
sizes.NumContStates = 0; %连续状态变量的个数
sizes.NumDiscStates = 0; %离散状态数
sizes.NumOutputs = 1; %输出数量 S-function
sizes.NumInputs = 2; %输入数量 S-function
sizes.DirFeedthrough = 1; % Flag 对直接馈通
sizes.NumSampleTimes = 1; %采样时间个数
sys = simsizes(sizes); %使用 sizes 信息加载 sys 向量
x0 = []; %初始状态值为空变量，因为没有
         %连续状态
str = []; %没有州订购
ts = [-1 0]; %两列矩阵，包含块的采样时
            %间和偏移量，该值表示 S-
            %function 块以与块相同的速率运
            %行

% mdlInitializeSizes 结束
%=====
% mdlOutputs
%=====
function sys=mdlOutputs(t,x,u)
if(u(1)==0) % 比如输入是 0，参考电压在第一扇区

```

```
N=1;
else
%此计算是使用 Clark 逆变换
    a1=u(1);
    b1=u(1)*(-0.5)+(sqrt(3)/2)*u(2);
    c1=u(1)*(-0.5)-(sqrt(3)/2)*u(2);
% If 的陈述是反 clark 的结果
    if a1>0
        a=0;
    else
        a=1;
    end
    if b1>0
        b=0;
    else
        b=1;
    end
    if c1>0
        c=0;
    else
        c=1;
    end
    N=4*a+2*b+c;           %扇区计算
end
    sys=N;                 %通用返回 N
%mdlOutputs 结束
```

在图 3.2 中，转矩输出的磁滞控制器参数设定点为[0.1 -0.1]，通量为 [0.002 -0.002]，输出转矩和通量的设定点为[1 0]。其中值 1 是均值开，值 0 是均值关。仿真中设定点的实现如图 3.5

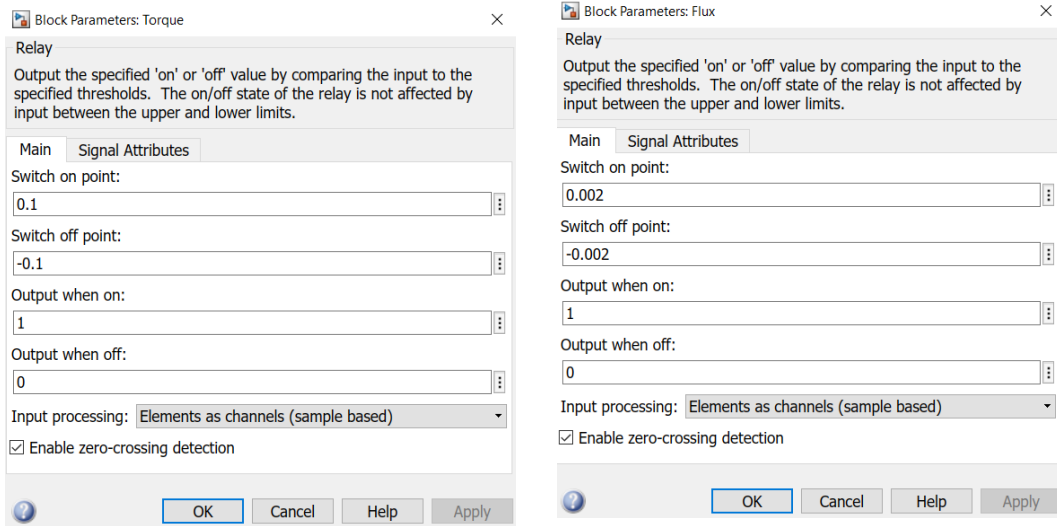


图 3.5 转矩和磁通参数

在此仿真中，使用 PID 控制器的三个参数。 Simulink 中有一个可以使用的即时块 PID，PID 控制器的数学模型是根据方程式的各个参数。在仿真中，比例参数为 0.1，积分为 5，导数为 0.1。滤波器系数用于实现微分作用，其值设置为 0.1。PID 控制器参数的输出饱和度为[5 -5]。如果参数之一的值的设定值太高或太低，都会使系统控制错误或响应性降低。仿真中 PID 控制器的参数见图 3.6 和图 3.7

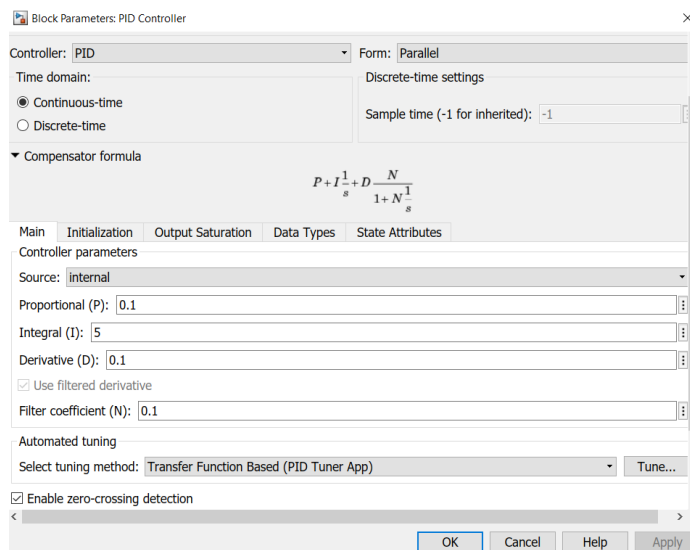


图 3.6 PID 控制参数

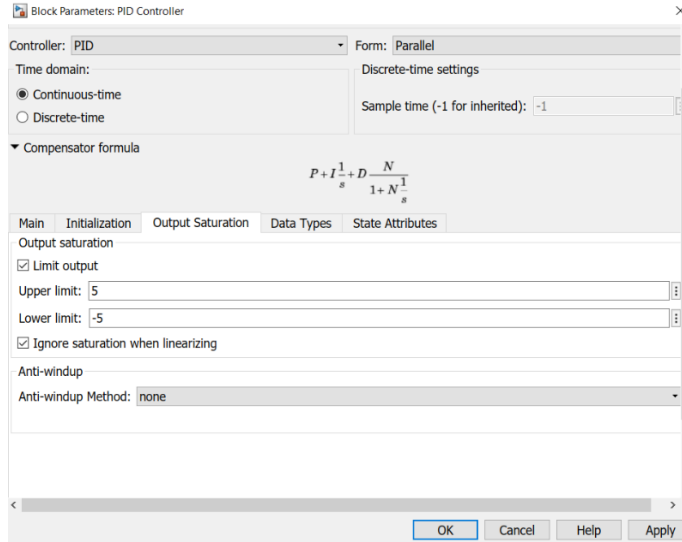


图 3.7 输出饱和 PID 控制器参数

在图 3.8 的 PWM 块内部，输出用于向逆变器提供脉冲的 pwn 的输出。在 1-D 找表中，表数据来自六个扇区并设置为表数据。1-D 找表为信号值不超过 1。布尔函数使信号输入在输出中声明 true 为 1，在 false 中为 0。Not 块使输出值从 1 变为 0，因此 not 的输出信号类似于 boolean 输出信号的反相信号。

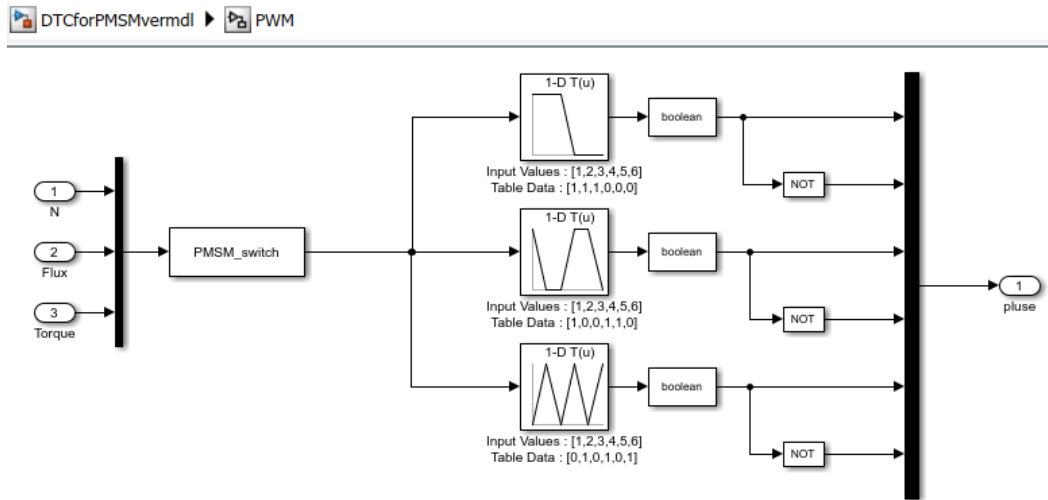


图 3.8 PWM 模块

PMSM_switch 用于 S-function，其中下面的 m-file S-function 的程序

```
function [sys,x0,str,ts] = PMSM_switch(t,x,u,flag) %下面概述了 S 函数的一般结构，并以 PMSM_switch 作为
```

```

switch flag,
    % 该声明 Switch 功能

    %%%%%%%%%%%
    % Initialization %
    %%%%%%%%%%%
    case 0,
        [sys,x0,str,ts]=mdlInitializeSizes;
    %%%%%%%%%%%
    % Outputs %
    %%%%%%%%%%%
    case 3,
        sys=mdlOutputs(t,x,u); %计算输出
    case {1,2,4,9},
        sys=[]; %未使用的 Flags
    %%%%%%%%%%%
    % Unexpected flags %
    %%%%%%%%%%%

    otherwise
        error(['Unhandled flag = ',num2str(flag)]); %错误处理
    end % sfuntmpl 结束

%=====
% mdlInitializeSizes
%=====

function [sys,x0,str,ts]=mdlInitializeSizes
    sizes = simsizes; % 调用 size 结构的
                        % simsize, 将其填充
                        % 并将其转换为 size
                        % 数组。

```

```

% 加载带有初始化信息的大小结构
sizes.NumContStates= 0; %连续状态变量的个数
%连续状态变量的个数

sizes.NumDiscStates = 0; %离散状态数
%离散状态数

sizes.NumOutputs = 1; %输出数量
%输出数量

S-function
sizes.NumInputs = 3; %输入数量
%输入数量

S-function
sizes.DirFeedthrough = 1; % Flag 对直接馈通
% Flag 对直接馈通

sizes.NumSampleTimes = 1; %采样时间个数
%采样时间个数

sys = simsizes(sizes); % 使用 sizes 信息加载 sys 向量
% 使用 sizes 信息加载 sys 向量

x0 = []; %初始状态值为空变量，因为没有连续状态
%初始状态值为空变量，因为没有连续状态

str = []; %没有州订购
%没有州订购

ts = [-1 0]; %两列矩阵，包含块的采样时间和偏移量，该值表示 S-function 块以与块相同的速率运行
%两列矩阵，包含块的采样时间和偏移量，该值表示 S-function 块以与块相同的速率运行

% mdlInitializeSizes 结束
%=====
% mdlOutputs
%=====
function sys=mdlOutputs(t,x,u)
V_Table=[2 4 6 1 3 5;4 1 5 2 6 3;3 6 2 5 1 4 ;5 3 1 6 4 2 ]; % 根据等式表 2.3
N=u(3) % Sector 输入
% Sector 输入

```

```
x=2*u(1)+u(2)+1;  
sys=V_Table(x,N);  
% mdlOutputs 结束
```