

第3章

PSIM基本操作与分析方法

3.1 元件查找与放置

PSIM 仿真软件集成了电力电子、控制、传感器、测量仪表、信号源等元件模型,选择一个电路元件模型有三种方式。

(1) 第一种方式是选择“View→Library Browser”菜单项或单击工具栏快捷图标,打开元件库浏览器进行元件模型查找,如图 3-1 所示。

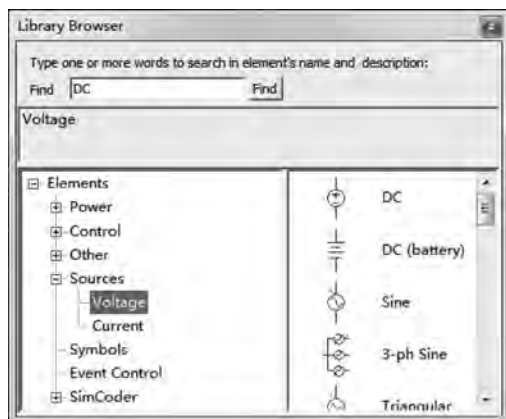


图 3-1 PSIM 元件库浏览器

在 PSIM 元件库浏览器中,可通过左侧的元件库树形目录定位到某一个元件库,此时右边区域将显示出可供选择的全部元件。若知道某个元件的具体名字或部分名字,也可以在查找框中输入,单击 Find 按钮可快速查找所需的元件。

在找到所需的具体元件后,可双击拾起元件,此时鼠标变成所选的具体元件,然后移动鼠标到电路原理图合适的位置,单击即可放置选择的元件;也可在元件浏览器中某一个具体元件上按住鼠标左键不放,拖动该元件到电路原理图中合适的位置,然后释放鼠标左键即可放置一个元件。

放置完一个元件后,鼠标处于元件拾起状态,可继续在合适位置放置该元件。若要放置其他元件,可返回元件库浏览器中,对需要放置的元件双击选中,或直接鼠标左键拖动该元件进行放置。

放置完一个元件后,鼠标处于元件拾起状态。如要释放该元件,可按键盘的“Esc”键释放元件,或者选择“Edit→Escape”菜单项释放元件,或者单击工具栏的选择图标释放元件。

(2) 第二种方式是通过菜单栏的“Elements”菜单,找到元件所在的元件库,然后找到所需的具体元件名称,在名称上单击即可拾起该元件,如图 3-2 所示。

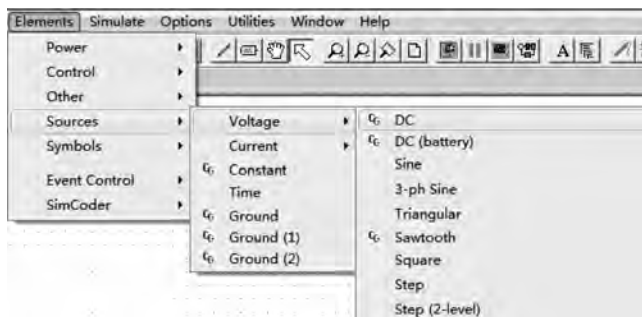


图 3-2 Elements 菜单下的元件

图 3-2 展示了“Elements→Sources→Voltage”元件库下面的元件,如要放置某一个具体元件,定位到具体元件后,在该元件名称上单击拾起该元件,然后在电路原理图绘制区域合适位置放置元件即可。放置完元件后,鼠标仍处于元件拾起状态。

(3) 第三种方式是通过 PSIM 窗口底部的元件工具栏进行元件选择、放置。元件模型工具栏如图 3-3 所示。该元件模型工具栏包含了一些使用频率较高的常用元件模型。



图 3-3 元件模型工具栏

若要通过元件模型工具栏放置元件,单击所需元件即可拾起该元件,然后移动鼠标到电路原理图绘制区域合适位置放置该元件即可。放置完元件后,鼠标仍处于元件拾起状态。

另外,在元件处于拾起状态时,可以右击对该元件的方向进行旋转调整,待调整到合适方向后即可放置元件。若需要调整已经放置的元件模型方向,可以单击该元件模型,以选中该元件,然后单击工具栏图标进行 90°旋转、水平旋转和垂直旋转,该旋转方式可适用于选中的整个区域内的元件方向调整。

3.2 创建仿真电路

3.2.1 新建电路原理图

双击“PSIM 9.1”快捷图标或者选择“开始菜单→PSIM 9.1.1→PSIM”菜单项启动

PSIM 仿真软件,如图 1-3 所示。随后选择“File→New”菜单项,或者单击工具栏的新建图标“”,创建一张无限大的电路设计图纸,如图 3-4 所示。

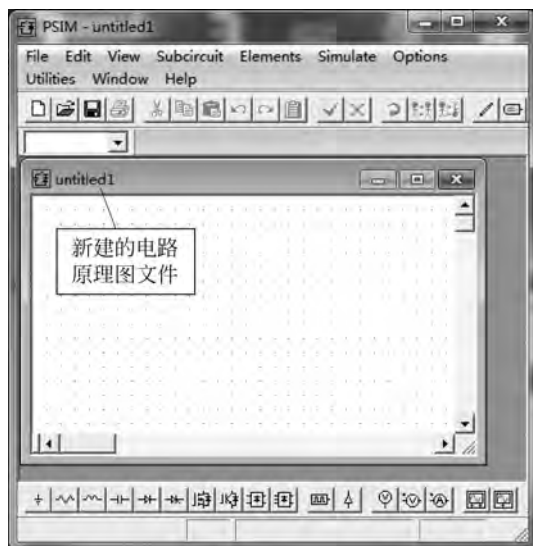


图 3-4 PSIM 新建电路设计文件

新建的电路图设计文件并未保存,可选择“File→Save”菜单项,或者单击工具栏的保存图标“”,或者按下快捷键“Ctrl+S”进行文件保存,在弹出保存对话框中选择存储目录及文件名进行保存。注意:电路图设计文件的后缀名是 psimsch。若要保存为旧版本的设计文件,可以选择“File→Save as Older Versions”菜单项,并在保持类型中选择相应版本进行保存。

3.2.2 绘制仿真电路模型

1. 放置电路元件

在新建电路设计图文件后,根据需要仿真的电路拓扑硬件,放置相应的仿真元件模型。元件的查找与放置可参看 3.1 节的具体操作方法。本小节以单相二极管整流仿真为例来绘制仿真电路模型。单相二极管整流电路硬件电路拓扑如图 3-5(a)所示。单相二极管整流硬件拓扑由正弦交流电源 AC、二极管 D 及负载 R 三个电路元件构成,因此需要从 PSIM 元件库中找到这三个元件,并放置到电路图设计文件上,如图 3-5(b)所示。

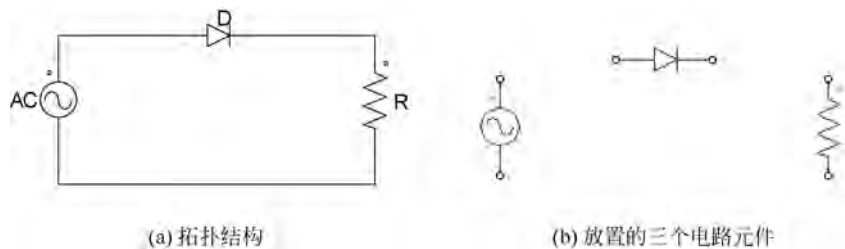


图 3-5 单相二极管整流拓扑

在放置元件过程中,可以利用删除、复制、粘贴、撤销等功能对元件进行处理,功能执行可以利用键盘上的快捷键,也可以利用工具栏上的功能图标“”。在执行相应功能

前需要选中元件,选中元件的方法是在元件上单击以选中一个元件,或者按住鼠标左键不放拖出一个区域,以选中该区域内的所有元件。在元件放置过程中,右击可以调整元件的放置方向,以适应电路图的绘制。

另外,如果在绘制电路原理图的过程中,需要放置一个图中已存在的元件,可以选中该元件,使用复制、粘贴功能进行相同元件放置。

2. 绘制元件连接线

在合适位置放置好三个元件后,需要将三个元件按照硬件电路拓扑连接的结构连接三个元件。连线方法是单击工具栏上类似于笔的画线图标“”,单击该图标后鼠标变成一支“笔”,然后移动“笔”到元件端点的小圆圈“”(如二极管的阴极 K 端)上,按住鼠标左键不放,并移动“笔”到负载电阻一个端点的小圆圈“”上,释放鼠标左键,此时一条电线将二极管的阴极 K 与电阻的一端连接起来了。在移动笔的过程中,可以看到一条电线跟着笔在移动。按照同样的方法将其他端点连接起来,连接好的电路图如图 3-6(a)所示。在绘制电线过程中,可以灵活使用编辑功能对元件及电线进行相应的操作。

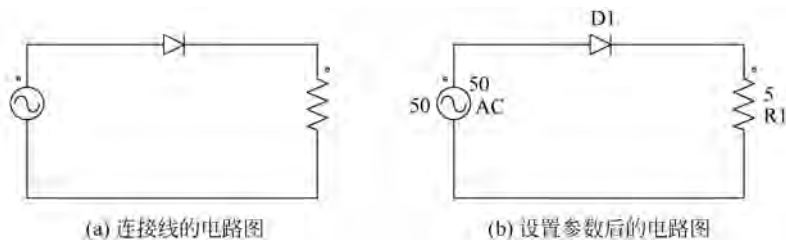


图 3-6 单相二极管整流拓扑

在绘制电线的过程中,可以单击已经绘制的电线,或者按住鼠标左键不放拖出一个区域选中该区域内的电线,随后可按键盘上的“Delete”按键或者单击工具栏的删除图标“”,对选中电线进行删除。

“笔”可以在任何想绘制电线的位置开始绘制电线,也可以在任何想结束绘制电线的位置停止绘制电线。未与元件端点或者其他线连接的线,在末端会出现一个小圆圈“”端点,以方便后续接着继续绘制。

3. 设置元件属性参数

元件连接好后,需要设置各元件的属性参数。以交流电源 AC 为例,先双击交流电源元件,弹出属性对话框,如图 3-7 所示。



图 3-7 元件属性参数设置对话框

图 3-7 左边对话框中显示的是元件默认参数,可以根据实际需要修改各参数值。如需要在电路图中显示设置的参数,可勾选参数设置文本框旁边的“Display”复选框,相应的参数就会显示在元件模型旁边。按照此方法将三个元件的属性参数依次设置为:

- 正弦交流电源名称设置为 AC,电源频率设置为 50Hz,最大峰值幅值设置为 50V,如图 3-7 右边视图所示;
- 电阻名称设置为 R1,阻值设置为 5Ω ;
- 二极管元件名称设置为 D1,其他参数默认不变。

设置完属性参数后的拓扑电路图如图 3-6(b)所示。元件旁边显示的参数可用鼠标拖放到合适的位置。另外,在设置元件参数时,若对某一个参数的定义不清楚,可在属性对话框上单击“Help”按钮,打开该元件的帮助文档,文档对元件的各参数及使用进行了详细描述。

4. 设置测量探针及仪表

为了查看电路运行的状态及性能,需要测量相关节点的电压、电流等参数,以便在电路运行结束后进行性能分析。假设需要查看正弦交流电源 AC 的输出电压、负载电流、负载电压,可利用 2.4.3 节中的相关探针及仪表进行测量。本例利用测量节点到地的电压探头“

图 3-8 放置测量探头及设置探头属性参数

放置完探头后,为区分各个测量探头,需要设置探头属性参数。探头属性参数设置方法类似于元件属性参数设置方法。正弦交流电源 AC 的输出电压测量探头命名为 V_{in} 、负载电流测量仪表命名为 I_o 、负载电压探头命名为 V_o 和 V_{o1} 。设置完属性及参考地的拓扑电路如图 3-8(b)所示。负载电压采用两种方式进行测量,其测量结果是一致的,因为 V_{o1} 的一个节点是 V_o 的测量节点, V_{o1} 的另一个节点是 Ground 参考地节点,所以测量结果是一致的。

通过上述四个步骤,即可完成仿真电路模型原理图的绘制。不论是简单仿真电路模型绘制,还是复杂仿真电路模型绘制,其绘制方法完全一样,必须经历上述四个步骤。四个绘制步骤的顺序可以交叉,并没有先后关系,在绘制电路原理图模型时灵活处理即可。

3.3 电路模型仿真

瞬态时域仿真分析是计算仿真模型的时域响应特性,并描述为时间的函数。每个计算数据的时间点称为一个时间步长。瞬态时域仿真分析是通过仿真控制选项设置瞬态时域分析的参数,如仿真时间步长、仿真总时间等。

对将要仿真的电路模型,必须设置时域仿真控制,需要在模型电路原理图的任意位置放置仿真控制元件(元件外形类似时钟),放置后即可设置仿真控制参数。本节将详细讲解瞬态时域仿真的操作细节。

3.3.1 仿真控制设置

在创建完仿真电路模型后,可以选择“Simulate→Simulation control”菜单项放置仿真控制元件。在原理图的合适位置放置“Simulation control”控制元件后,双击该元件,弹出仿真控制属性参数设置对话框,如图 3-9 所示。在该界面中,需根据仿真需要,对仿真控制参数进行设置,各仿真参数定义如表 3-1 所示。在属性界面的 SimCoder 页面,可添加 SimCoder 注释文本,该注释文本将插入自动生成的 C 程序代码中。



图 3-9 仿真控制属性设置

表 3-1 仿真控制参数定义

参 数	描 述
Time Step	仿真时间步长,单位为 s(sec.)
Total Time	总仿真时间,单位为 s(sec.)
Free Run	自由运行。如果勾选“自由运行”复选框,仿真将以自由运行模式运行,直到手动停止为止。在仿真过程中,单通道或双通道示波器(在“Elements→Other→Probes”菜单项下)可用于查看电压波形,电流示波器可用于查看分支电流(右击元件顶部,选择“电流示波器”)
Print Time	将仿真结果保存到输出文件的时间(默认值=0)。在此之前不保存输出
Print Step	打印步长(默认=1)。如果打印步骤设置为 1,则每个数据点都将保存到输出文件中。如果是 10,则每 10 个数据点中保存一个数据点,这有助于减小输出文件的大小

续表

参 数	描 述
Load Flag	加载功能标志(默认值= 0)。如果标志为 1,则从文件(扩展名为. ssf)中优先加载仿真参数值作为初始条件
Save Flag	保存功能标志(默认值= 0)。如果标志为 1,则当前仿真结束时自动将仿真设置参数保存到扩展名为. ssf 的文件中
Hardware Target	硬件目标板选择。用于指定 SimCoder 自动代码生成的硬件目标板。硬件目标板类型可以是以下之一： • None: 无硬件目标 • TI F28335 : TI F28335 硬件,选择此项后可选择生成代码的版本 • PE-Pro/F28335: Myway 公司的 PE-Pro/F28335 • General_Hardware: 通用硬件 • PE_Expert3: Myway 公司的 PE_Expert3 硬件,对于 PE-Expert3 硬件,需通过下拉框将 PE-View 版本设置为 PE-View8 或 PE-View9

在设置仿真步长时,应注意以下两点:

(1) 在 PSIM 中,仿真时间步长在整个仿真过程中都是固定的。为了确保准确的仿真结果,必须正确选择时间步长。限制时间步长的因素包括开关周期、脉冲/波形的宽度以及瞬变的间隔。建议时间步长至少比上述最小步长小一个量级。

另外,PSIM 采用了一种插值技术,该技术将计算出准确的开关时刻。利用这种技术,由于开关时刻和离散仿真点的未对准引起的误差将被大大减小。通过该技术可以以较大的时间步长进行仿真,其仿真结果仍非常准确。

(2) 仿真允许的最大时间步长在 PSIM 中自动计算。自动计算结果将与用户设置的时间步长进行比较,然后在仿真中使用两者中较小的一个。

3.3.2 仿真运行与控制

在设置完仿真控制参数后,可选择“Simulate→Run Simulation”菜单项,或单击工具栏的仿真运行“”图标,启动 PSIM 仿真器进行电路模型仿真。

在仿真运行时,可单击“Simulate”菜单下的“Cancel Simulation/取消仿真”“Pause Simulation/暂停仿真”“Restart Simulation/重启仿真”对仿真进行控制。也可以单击工具栏的仿真运行控制工具图标“”实现仿真运行控制。

3.4 仿真结果查看与分析

3.4.1 运行 Simview

默认情况下,在仿真结束后会自动打开 Simview 波形显示和后处理程序。如果 Simview 没有自动打开,可以选择 PSIM 菜单的“Simulate→Run SIMVIEW”菜单项,或单击工具栏上的“”工具图标,运行波形显示程序 Simview。

如果需要 PSIM 仿真结束后自动运行 Simview 波形显示分析程序,可在 PSIM 的

“Options”菜单中,勾选“Auto-run SIMVIEW”项即可在仿真结束后自动运行 Simview,如图 3-10 所示。



图 3-10 自动运行 Simview 设置

对 3.2 节创建的仿真电路模型,单击图标“”启动 PSIM 仿真。仿真结束后自动打开的 Simview 界面如图 3-11 所示。在弹出的属性窗口左侧列出了在电路模型中设置的测量探头名称。测量探头名称实际就是保存测量值的变量名称,该变量保存了在整个仿真中采集的数据,后续可以在 Simview 中对该数据进行查看与分析。

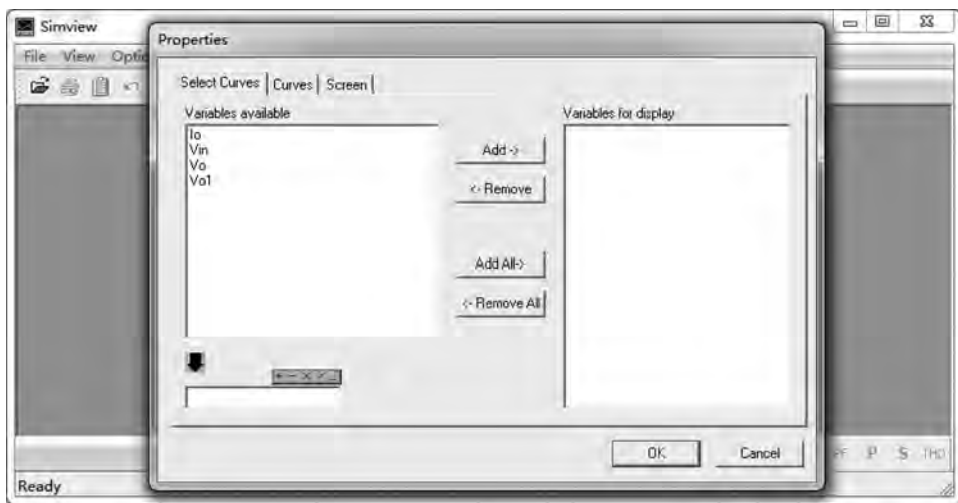


图 3-11 Simview 运行界面

3.4.2 仿真结果查看

1. 添加显示变量

在图 3-11 的属性窗口中,可以将左侧列表中变量添加到右侧显示变量列表中,添加完成后,单击“OK”按钮进行结果数据显示。显示变量添加方法有三种方式:

- ◇ 第一种: 双击左侧可用变量列表中需要显示的变量,则该变量立即添加到右侧显示变量列表中。
- ◇ 第二种: 单击左侧可用变量列表中需要显示的变量,然后单击“Add->”按钮即可添加到右侧显示变量列表中。该方法可以单击选中多个变量,然后单击“Add->”按钮完成多个变量的添加。
- ◇ 第三种: 如果想添加所有变量,直接单击“Add All->”按钮添加即可。

另外,Simview 支持对变量进行数学运算,以表达式的形式进行显示。构建数学运算表达式的方法有两种方式:

- ◇ 第一种: 在图 3-11 左下角的空白框中手动输入“左侧列表中可用变量”来构建相应的数学表达式。表达式支持的数学运算函数可单击表达式输入框右上角的图标“”进行添加或查看。
- ◇ 第二种: 在“左侧列表可用变量”中选中一个变量,然后单击表达式输入框左上角

“”图标添加到编辑框中,再单击“”图标添加相应的数学函数,以此方式构建出需要查看的变量表达式。

构建完数学表达式后,单击“Add->”按钮添加到右侧显示变量列表中。添加完需要显示变量的属性对话框如图 3-12 所示。在图 3-12 所示显示列表框中的变量,如果不想观察或者添加了错误的变量,可以在右侧列表中双击该变量进行删除;或者在右侧列表中单击选中需要删除的变量(可以一次性选中多个需要删除的变量),然后单击“<-Remove”按钮进行删除。如果想删除所有变量,直接单击“<-Remove All”按钮即可。

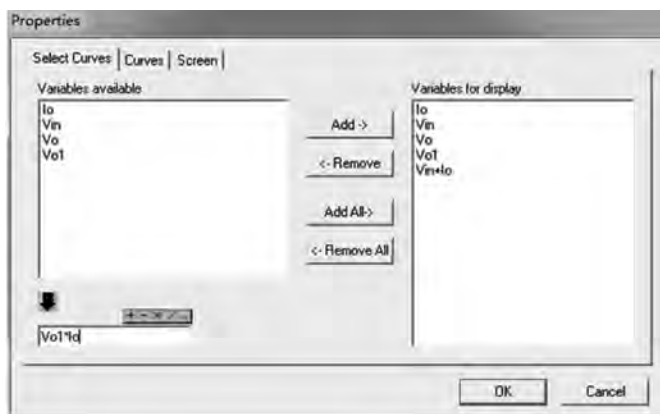


图 3-12 添加需要观察的变量

在图 3-12 窗口的“Curves”页面,可设置各显示变量波形的颜色、线型、符号标志等;在“Screen”页面可以设置 Simview 显示窗口的前景颜色、背景颜色、栅格颜色、显示字体等属性。

2. 波形显示

在图 3-12 中单击“OK”按钮进入波形显示界面,所有需要显示的变量均在同一个坐标系窗口中显示,如图 3-13 所示。

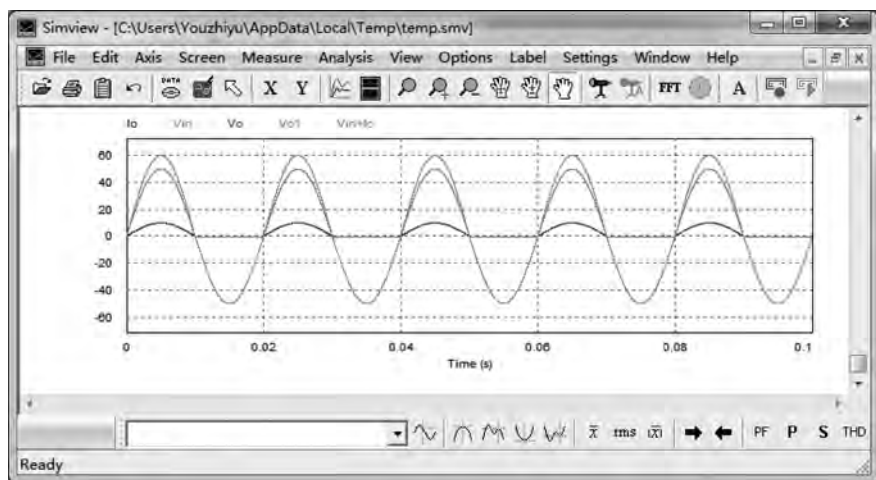


图 3-13 变量波形显示窗口

3. 增加/删除显示变量

如果要增加/删除显示坐标窗口中的变量,可双击波形显示区域,或者在图形区域右击并在右键菜单中选择“Add/Delete Curves...”项,或者单击工具栏“”图标,随后弹出变量设置属性对话框,可按照上述“添加显示变量”的方法进行增加或删除需要显示的变量,同时也可以修改显示波形的颜色等属性参数。

4. 分窗口显示波形

多个变量在同一个窗口进行显示,有时不便于观察分析。可以选择 Simview 的“Screen→Add Screen”菜单项或者单击工具栏的“”图标增加一个显示坐标系窗口,在弹出的属性对话框中添加需要观察的变量,随后单击“OK”按钮回到波形显示窗口,如图 3-14 所示(可以根据显示需要,添加多个显示窗口)。

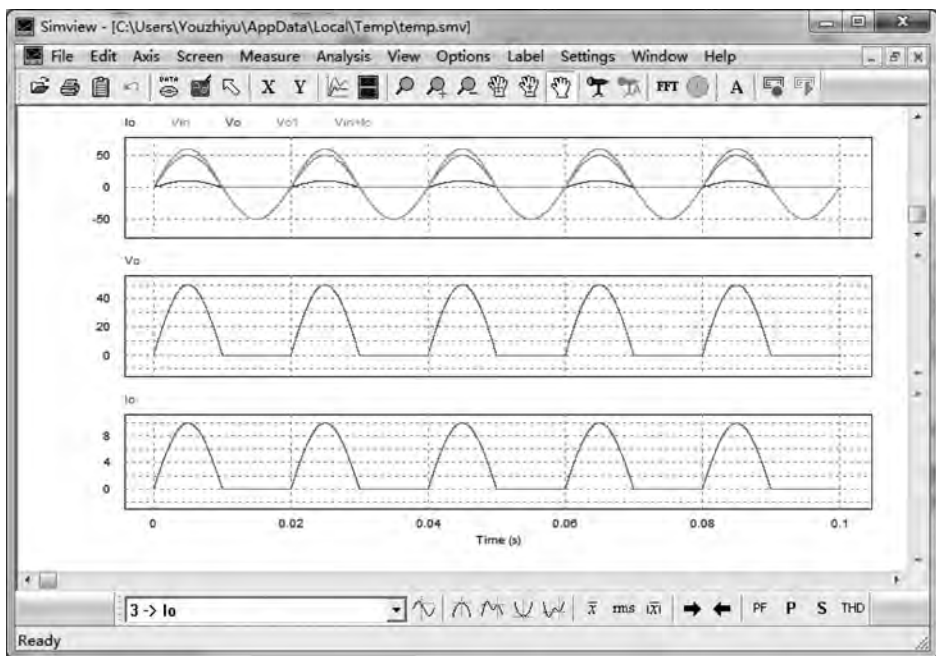


图 3-14 多波形坐标显示窗口

在显示出所需变量波形后,便可根据仿真电路硬件拓扑工作原理,对波形进行详细分析,以确定是否存在问题。若存在问题可返回仿真电路原理图设计窗口,对电路模型进行调整、修改,修改后重新进行仿真分析,直至仿真结果符合设计要求为止。

5. 波形测量

Simview 提供了测量功能,可以对选定的变量波形进行测量。选择“Measure→Measure”菜单项,弹出测量值显示对话框窗口。在波形显示窗口单击将出现一条测量线,同时显示选中变量的波形值;在波形显示窗口右击将显示另一条测量线,并且将测量两条竖线之间的波形。

6. 波形分析

待测量变量可以通过测量工具栏进行选择,也可以在波形显示坐标系窗口左上角的变量显示栏进行选择,单击需要测量的变量名即可选中需要分析的具体变量。选择变量后,可

利用测量工具栏进行相关变量的测量与分析,分析工具体功能说明见表 1-23 的说明。

7. 显示波形调整

显示波形可以利用菜单“Axis”设置 X、Y 坐标的属性及显示范围,也可以通过“View”菜单中的“Zoom”“Re-draw”等工具进行波形显示的调整。

8. 添加注释及导出

在分析完波形后,可以在波形显示区域,利用菜单“Label”添加文字、箭头、直线等波形注释符号。可选择“Edit→Copy to Clipboard”菜单项将调整好的波形整体复制到剪贴板供使用。分析完的显示波形,可以选择“File→Save As”菜单项进行数据保存,便于后续继续分析。

3.5 子电路创建

在进行仿真模型搭建时,为简化模型视图,可以将某一功能单元封装成 PSIM 子电路元件形式。某一功能单元以子电路元件的形式在模型电路中进行显示,其视图简单明了、功能清晰。如图 3-15(a)为 0/1 电平脉冲转换成-3/6 电平脉冲的子电路元件,图 3-15(b)为其具体功能实现电路模型。

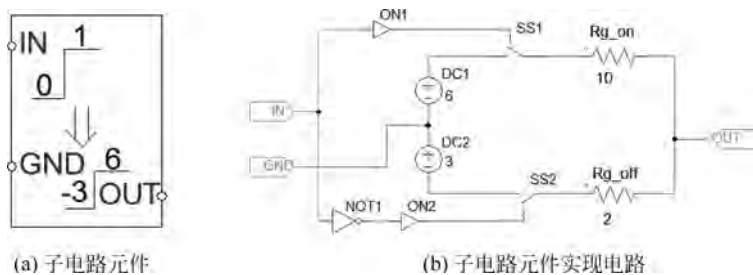


图 3-15 0/1 电平转-3/6 电平子电路元件

3.5.1 子电路元件创建

PSIM 仿真软件“Subcircuit”菜单下的各项子菜单是创建子电路元件的菜单项,其详细功能描述见表 1-4(a)及表 1-4(b)。本节以图 3-15 所示子电路元件为例,详细讲解创建过程。

1. 创建子电路原理图文件

启动 PSIM 软件,新建 PSIM 电路图文件。保存新建的 PSIM 电路图文件,并取名为“subPWM.psimsch”。

2. 构建子电路元件实现电路

在创建的 subPWM 电路原理图设计文件中,按照图 3-15(b)所示电路放置模型实现元件。

- ◇ 放置两个直流电源(选择“Elements→Sources→Voltage→DC”菜单项)DC1、DC2,分别设置为 6V 和 3V;

- ◇ 将 DC1 与 DC2 串联,并从串联点引出公共输入端,作为参考 GND 连接点,形成 6V 和 -3V 电源;
- ◇ 放置两个双向开关(选择“Elements→Power→Switches→Bi-directional Switch”菜单项) SS1、SS2;
- ◇ 放置两个电阻(选择“Elements→Power→RLC Branches→Resistor”菜单项) Rg_on、Rg_off,阻值分别设置为 10Ω 和 2Ω ;
- ◇ 放置两个通断控制器(选择“Elements→Other→Switch Controllers→On-off Controller”菜单项) ON1、ON2;
- ◇ 放置一个非门(选择“Elements→Control→Logic Elements→NOT”菜单项) NOT1。在所有元件放置完成后,按照图 3-16 所示将各元件连接起来。

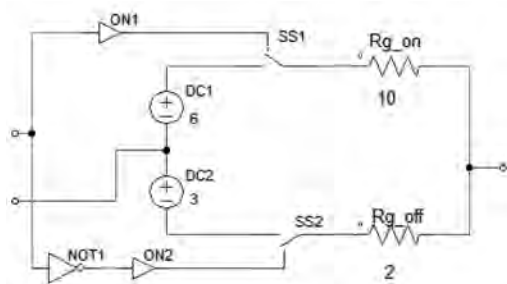


图 3-16 子电路功能实现电路

3. 放置输入/输出端口

图 3-16 已经将实现元件连接起来了,并引出了输入/输出连接点。

- ◇ 选择“Subcircuit→Place Input Signal Port”菜单项,鼠标指针变成一个输入端口,放置在 ON1 和 NOT1 连接的导线上,并弹出如图 3-17(a)所示的端口设置窗口。窗口包括参数设置及颜色设置两个页面。在参数设置页面,单击端口预设的位置,随后在端口名称 Port Name 框内输入端口名称“IN”,如图 3-17(b)所示。随后关闭该设置窗口完成 IN 端口名称设置。以同样的方法设置 DC1 与 DC2 串联点的公共参考点 GND 输入端口,并将名称设置为“GND”。

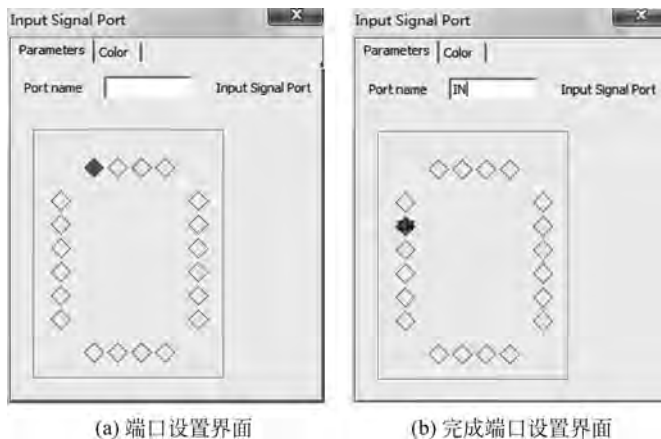


图 3-17 端口设置界面

◇ 选择“Subcircuit→Place Output Signal Port”菜单项,鼠标指针变成一个输出端口,放置在两个电阻相连的中间位置,在弹出的端口设置界面中完成输出端口设置。完成输入/输出端口设置后的模型电路图如图 3-15(b)所示。

选择“Subcircuit→Display Port”菜单项,可查看子电路所有端口设置情况,如图 3-18 所示。

4. 设置子元件尺寸

选择“Subcircuit→Set Size”菜单项,打开子元件尺寸设置窗口,如图 3-19 所示。在该窗口可以选择子元件的外形尺寸大小。

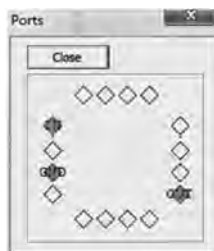


图 3-18 子电路端口设置显示

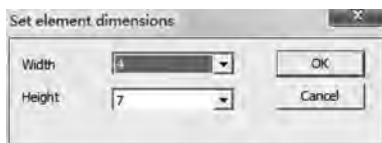


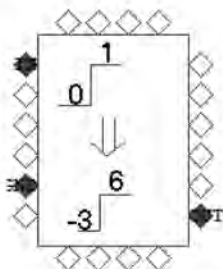
图 3-19 子元件尺寸设置窗口

5. 绘制子元件图形符号

选择“Subcircuit→Edit Image”菜单项,打开子电路图形符号编辑窗口,如图 3-20(a)所示。根据该子电路的具体功能绘制其形象图形符号,以表达子电路具体功能,如图 3-20(b)所示。当然也可以不绘制外形图形符号,不影响元件功能。



(a) 子电路图形符号编辑窗口



(b) 编辑完成的子电路元件图形

图 3-20 子电路图形符号绘制

绘制完成后关闭子电路图形符号绘制窗口,回到子电路模型窗口。通过菜单或者快捷键保存该子电路,随后关闭该电路图设计文件窗口,即完成子电路元件创建。

3.5.2 子电路元件调用

在 3.5.1 节创建了一个电平转换子电路,本小节对该元件调用并进行仿真测试。

◇ 启动 PSIM 仿真软件,新建一个 PSIM 仿真电路图设计文件。

◇ 选择“Subcircuit→Load Subcircuit”菜单项,弹出加载子电路文件窗口如图 3-21 所示。双击子电路文件“subPWM. psimsch”进行子电路元件加载。随后鼠标指针变成该子电路元件的外形图形符号,根据需要放置在适当位置即可。



图 3-21 子电路加载窗口

另外,如果在创建子电路元件时,将该子电路元件模型电路图设计文件保存到自定义元件目录“\PSIM9.1.1\User Defined\Custom library”下,将会在“Elements→Custom library”菜单项下出现创建的子电路元件,如图 3-22 所示。选择菜单下的元件“Elements→Custom library→subPWM”菜单项,鼠标立即拾起该元件,移动鼠标到适当位置,即可放下该元件。

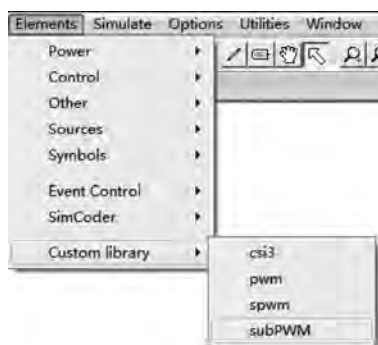
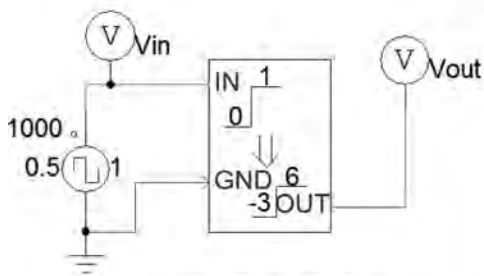


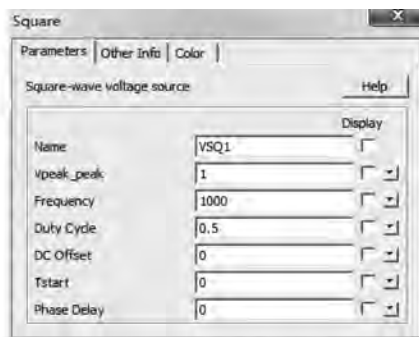
图 3-22 元件库中的自创子电路元件菜单

◇ 利用方波电压源(选择“Elements→Sources→Voltage→Square”菜单项或者单击工具栏的方波电源图标“”)构建如图 3-23(a)所示的

测试电路,并添加输入、输出电压探头对输入、输出波形进行测量,方波电压源参数设置如图 3-23(b)所示。



(a) 子电路测试电路模型



(b) 方波电源参数设置

图 3-23 测试电路模型及测试信号

◇ 设置“仿真控制”参数后进行仿真测试,测试结果波形如图 3-24 所示。

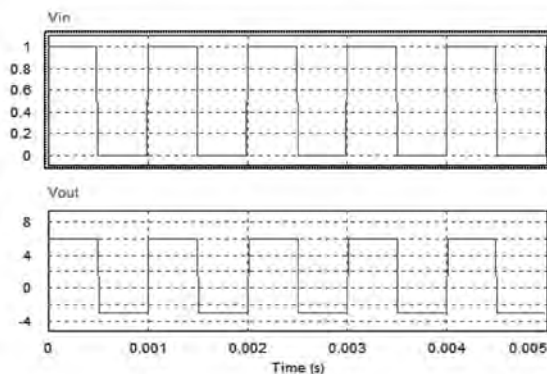


图 3-24 仿真结果波形

从图 3-24 仿真结果波形可以看出,设计的子电路元件实现了将 0/1 电平转换成 $-3/6$ 电平脉冲。从图 3-23(a)所示电路模型看,电路模型简单、清晰、直观。

3.5.3 主电路定义子电路元件参数

在 3.5.1 节创建的子电路元件,其元件参数是在子电路中设置的常量值,无法在调用主电路中设置或修改。另外,若在主电路中调用多个子电路元件,也无法将各个子电路元件的参数设置成不同的值。

为解决该问题,在创建子电路元件时,子电路具体实现电路元件的参数可设置成变量的形式,在主电路中就可以修改各变量的参数值,具体步骤如下:

◇ 在构建子电路实现电路时,将各元件的参数值设置为变量,如图 3-15(b)中 DC1 和 DC2 的电压值分别设置为 VH、VL,修改设置后的电路原理图如图 3-25 所示。

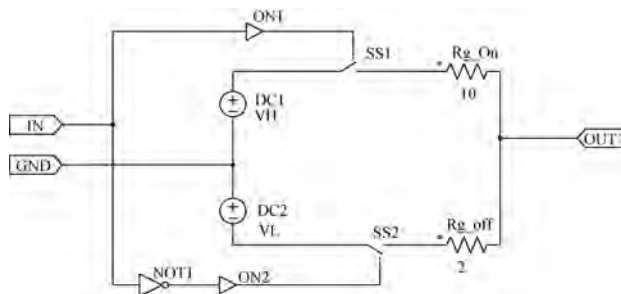


图 3-25 电源参数设置为变量

◇ 选择“Subcircuit→Edit Default Variable List”菜单项,在弹出的子电路默认变量列表中添加电压源 DC1、DC2 的默认电压参数值。添加方法是通过单击“Add”“Modify”“Remove”按钮进行添加、修改和删除,如图 3-26 所示。设置完成后单击“OK”按钮保存。

◇ 默认参数值设置完后,保存子电路模型电路图,随后关闭子电路模型电路图文件,并返回调用主电路,在主电路中调用子电路元件,搭建功能仿真电路模型。

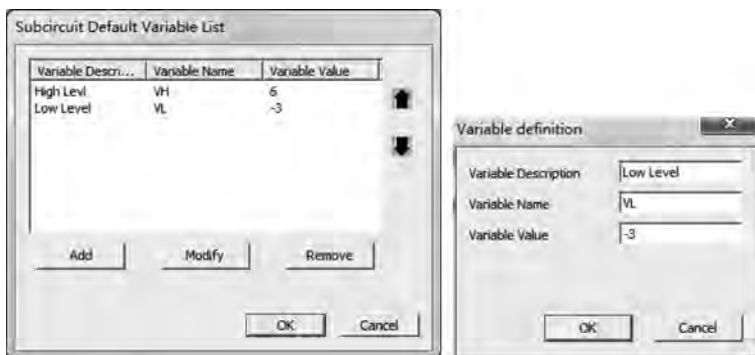


图 3-26 子电路变量默认值设置

- ◇ 在搭建好的功能仿真电路模型中单击选中子电路元件,随后选择“Subcircuit→EditSubcircuit”菜单项弹出设置窗口界面,切换到 Subcircuit Variables 页面,如图 3-27 所示,该界面显示了子电路元件可以设置的参数变量。
- ◇ 在图 3-27 中,单击需要修改的变量,再单击“Modify”按钮对该变量的值进行修改。或者双击需要修改的变量,在弹出变量修改窗口进行修改,如图 3-28 所示。

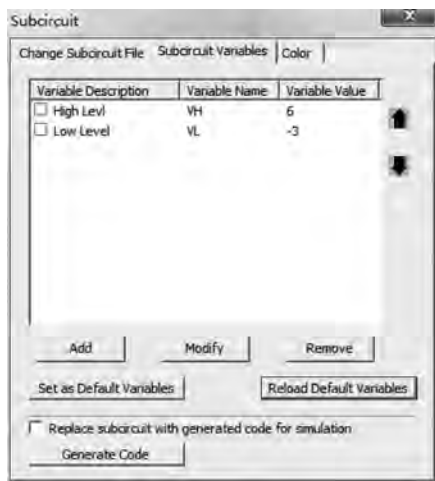


图 3-27 主电路中设置子电路变量窗口

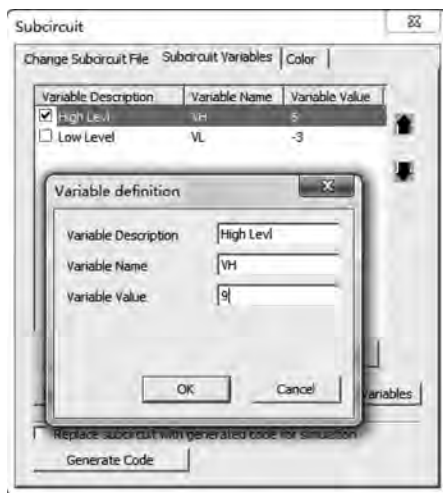


图 3-28 修改变量默认值

- ◇ 修改设置完成后,在 Subcircuit Variables 页面将变量前面的复选框选中,使其变量参数能在主电路调用的子电路元件旁显示。
- ◇ 设置完成后,直接关闭该对话框并返回到主电路设计图,在主电路图中可以看到子电路元件参数的具体设置,如图 3-29 所示。

在图 3-29 中,VSQ1、VSQ2 参数设置完全一样,但两个子电路元件模型参数设置不一样,子电路元件模型内部实现完全一样。对该电路模型进行仿真测试,测试波形如图 3-30 所示。

图 3-30 表明同样的子电路在主电路中被多次使用时,不同的参数值可赋值给同样的变量,达到在主电路中定义子电路元件参数的目的。



3.6.1 参数文件格式说明

参数文件是设计者在创建仿真电路模型时创建的文本文件。参数文件的格式有以下几种形式：

% 注释

```
(global) <变量名> = <值>           % 注释:定义"(global)"仅在 SimCoder 中使用
<变量名> <值>                       //添加评论
LIMIT <变量名> <下限> <上限>
% 注释
//注释
```

其中,

- ◇ <变量名>: 是在电路元件参数中设置的参数变量(例如: R1);
- ◇ <值>: 可以是数字(例如: $R1 = 15.1$)或数学表达式(例如: $R3 = R1 + R2/2$);
- ◇ =: 用来连接<变量名>和<值>(例如: $R1 = 15.1$), 等号“=”也可以用“空格”(例如: $R1\ 15.1$)代替;
- ◇ %或//: 字符%或//到本行末尾的文本被视为注释(例如: %R3 是负载电阻);
- ◇ (global): 仅在 SimCoder 中使用, 定义后面的变量为全局变量。

注意:

- ▶ 在数学表达式中, 变量与运算符之间不能有空格。例如“ $R3 = R1 + R2/2$ ”是正确的, 但是“ $R3 = R1 + R2/2$ ”(“R1”与“+”之间加入空格)将被视为“ $R3 = R1$ ”, 因为空格被视为数学表达式的末尾, “ $R3 = R1$ ”后面的字符将被忽略。
- ▶ “(global)”标识仅在 SimCoder 的自动代码生成中起作用, 表明是一个全局变量, 对于 PSIM 仿真将被忽略此定义。例如: 在 PSIM 仿真时将参数定义为“(global) $k_p = 1.2$ ”与定义为“ $k_p = 1.2$ ”效果相同。
- ▶ 参数文件的文件名可任意取, 其后缀为 txt, 定义一个参数文件的示例如图 3-31 所示。

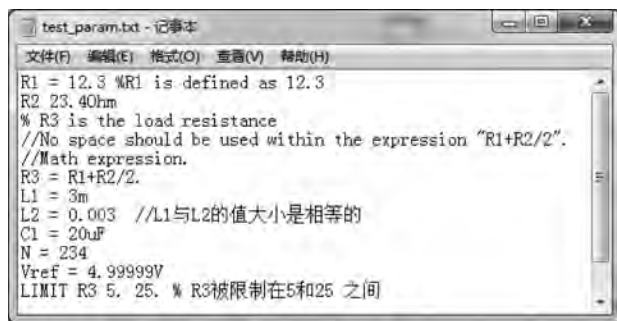


图 3-31 参数文件示例

3.6.2 参数文件调用示例

利用参数文件的方式定义仿真电路模型中各元件的参数非常方便, 也便于对参数的保存与修改。利用参数文件构建仿真电路模型中各元件参数的方法如下:

(1) 搭建仿真电路模型, 并将元件参数设置为变量的形式, 示例如图 3-32 所示。示例定义一个直流电源的电压值为 V_{in} , 三个电阻元件的阻值分别为 $R1$ 、 $R2$ 、 $R2 * R1$ 。放置三个电压探头和三个电流探头, 用于测量输入电压及流过电阻的电流, 以检验模型的工作状态。

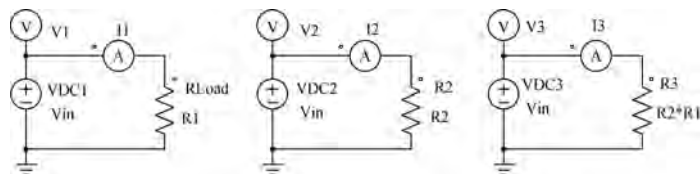


图 3-32 元件参数变量模型

(2) 在仿真电路模型保存的目录下新建文本文件“test_param.txt”，并在文本文件中输入元件参数变量的值并保存，如图 3-33 所示。

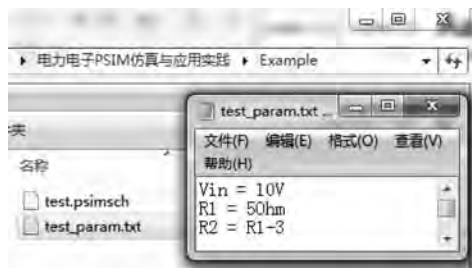


图 3-33 创建参数文本文件

(3) 在仿真电路模型中，选择“Elements→Other→Parameter File”菜单项添加参数文件元件模型。然后双击参数文件元件模型，在弹出的属性对话框中单击“File”按钮，在弹出菜单中选择“Open...”，打开刚才创建的“test_param.txt”参数文件，如图 3-34 所示。



图 3-34 导入参数文件

随后关闭该属性对话框即可。若需要在参数文件元件旁边显示文件的内容，可选中显示复选框。

(4) 设置仿真控制对电路模型进行仿真，仿真结果如图 3-35 所示。V1、V2、V3 测量的电压值均为 10V，I1 电流 2A，I2 电流 5A，I3 电流为 1A，符合设计的仿真电路模型。

(5) 若要修改参数变量值，可打开参数文件元件的属性对话框，在变量值定义处直接进行修改，修改完成后 PSIM 将自动保存并更新参数值。也可以手动单击“File”按钮，在弹出菜单中选择“Save...”进行保存。

另外，也可以不用手动创建参数文件，而直接在添加的参数文件元件属性对话框中输入各变量的具体定义。即上述第(2)步不执行，在第(3)步中双击打开参数文件元件属性框后，PSIM 默认创建一个未保存的“para-untitled1.txt”文本文件。直接在属性变量定义框内输

入参数变量的具体定义,然后单击“File”按钮,在弹出菜单中选择“Save As...”,保存文件名为“test_param2.txt”文件,如图 3-36 所示。

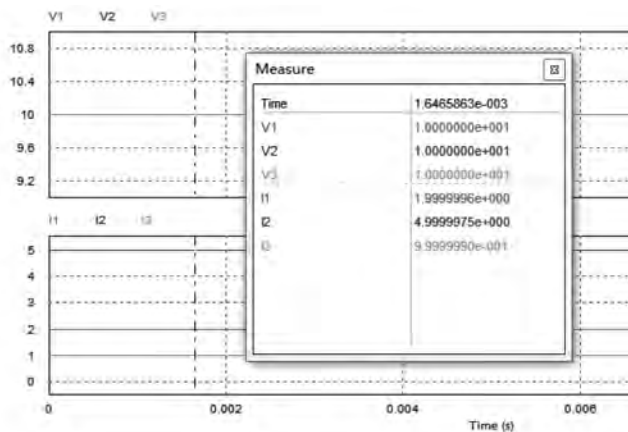


图 3-35 仿真测试结果波形

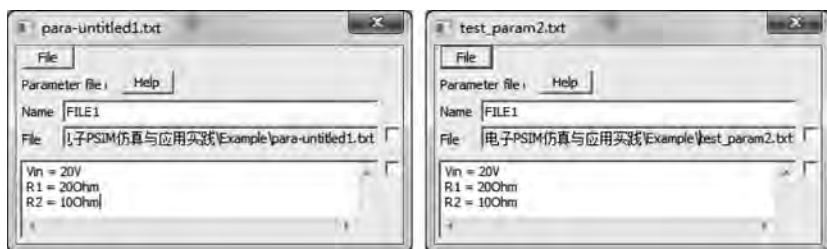


图 3-36 PSIM 参数文件元件自动创建参数文件

若参数文件名及保存路径直接采用 PSIM 默认创建的文件名及存储路径,在输入完变量参数值设置后,直接关闭属性对话框,PSIM 将自动保存该参数文本文件。

3.7 内嵌 C 程序块应用

3.7.1 简化 C 程序块

在搭建电路仿真模型时,若某一功能不便使用 PSIM 元件库中的电路元件模型构建,可以使用简化 C 程序块元件,编写 C 程序代码实现。简化 C 程序块元件在 PSIM 元件库的“Elements→Other→Function Blocks→Simplified C Block”菜单项下。Simplified C Block 允许用户直接输入 C 代码而无须编译代码,C 解释器引擎将在运行时解释并执行 C 代码。这使得搭建模型时编写自定义 C 代码变得非常容易,并且可以定义和修改模块的功能。

与 PSIM 元件库中“Elements→Other→Function Blocks→C Block”菜单项的 C Block 相比,Simplified C Block 更易于使用,并可用于自动代码生成。

1. 简化 C 程序块构建

在 PSIM 电路原理图中添加 Simplified C Block 元件,双击该元件,打开简化 C 程序块

的属性对话框,如图 3-37 所示。



图 3-37 简化 C 程序块属性对话框

在打开的属性对话框中,在 Name、Input、Output 输入框中定义模块的名称、输入端口数、输出端口数。在 C 程序代码输入框内编写具体功能的 C 程序代码。

自定义的输入变量分别为 $x_1, x_2, \dots, x_n$ (n 为定义的输入端口数量),自定义的输出变量分别为 $y_1, y_2, \dots, y_n$ (n 为定义的输出端口数量)。输入变量 x_n 是从 PSIM 传入的输入参量,输出变量 y_n 是简化 C 程序块传到 PSIM 的输出参量。在编辑 C 代码时,除了使用自定义的输入、输出变量外,还可以在代码中使用 t (PSIM 经过的时间 t)、 $delt$ (时间步,从 PSIM 传递)两个变量。所有输入和输出变量都是 double 数据类型,编写的简化 C 程序块代码会在每个时间步被 PSIM 调用。完成 C 程序代码输入后,可单击属性对话框的“Check Code”按钮,检查代码是否正确。

简化 C 程序块定义的输入/输出节点排列顺序是从上到下,输入端口在左侧,输出端口在右侧。例如,对于具有 2 个输入和 3 个输出的块,节点分别为 x_1, x_2, y_1, y_2, y_3 。单击“Edit Image”按钮,可对元件外形图形进行编辑,如图 3-38(a)所示。利用图形编辑工具,可在外形图形上放置一些图形符号及字符,如图 3-38(b)所示。设置完成后关闭图形编辑返回属性对话框,再返回电路原理图中, Simplified C Block 元件外形如图 3-38(c)所示。

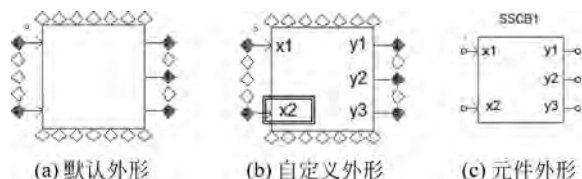


图 3-38 简化 C 程序块外形图形编辑

Simplified C Block 元件外形中带点的输入引脚是第一个输入 x_1 端口。注意:未使用的输入节点必须接地。同时与 DLL 块不同,简化 C 程序块的代码不可调试和逐步执行。

2. 简化 C 程序块应用

在构建完如图 3-37 所示的简化 C 程序块功能后,即可利用 C 程序块元件搭建仿真电路

模型,并进行仿真测试,如图 3-39 所示。SSCB1 为图 3-37 所创建的简化 C 程序块。

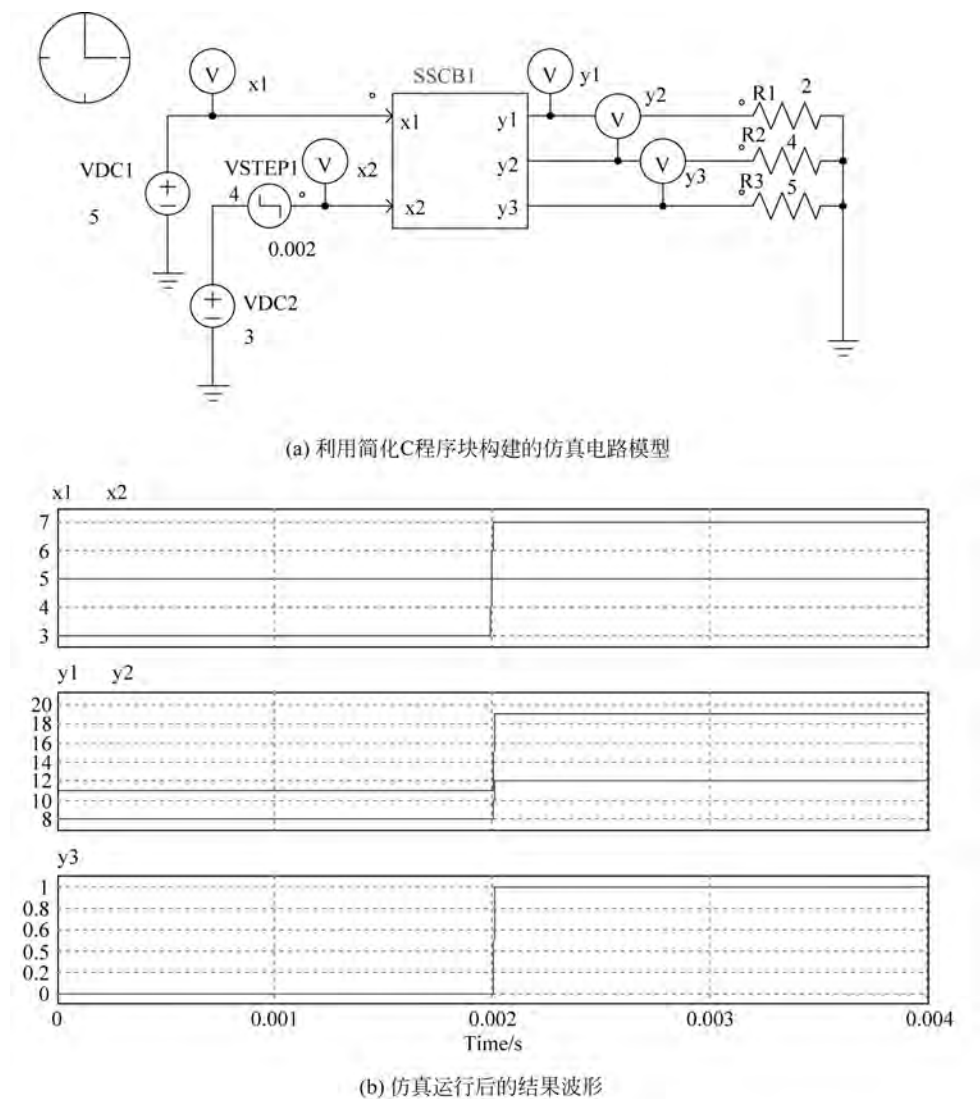


图 3-39 简化 C 程序块构建仿真模型

每个简化 C 程序块都是一个独立单元,一个简化 C 程序块中的全域变量对其他 C 程序块来说是不可见的。将 C 程序块中的值传递给另一个 C 程序块或其他任何电路的唯一方法是通过 C 程序块的输入/输出端口。当 C 程序块的输出直接连接到另一个 C 程序块的输入时,PSIM 仿真引擎首先运行第一个 C 程序块,然后运行第二个 C 程序块。

3.7.2 通用 C 程序块

通用 C 程序块(C Block)的功能类似 3.7.1 节的简化 C 程序块。C Block 在 PSIM 元件库的“Elements→Other→Function Blocks→C Block”菜单项下,C Block 允许用户直接输入 C 代码而无须编译代码,C 解释器引擎将在运行时解释并执行 C 代码。这使得搭建模型时编写自定义 C 代码实现自定义功能变得非常容易,并且可以定义和修改模块的功能。与

Simplified C Block 相比, C Block 功能更强,但不可用于 SimCoder 自动代码生成。

1. C 程序块构建

在 PSIM 电路原理图中添加 C Block 元件,双击该元件,打开 C 程序块的属性对话框,如图 3-40 所示。

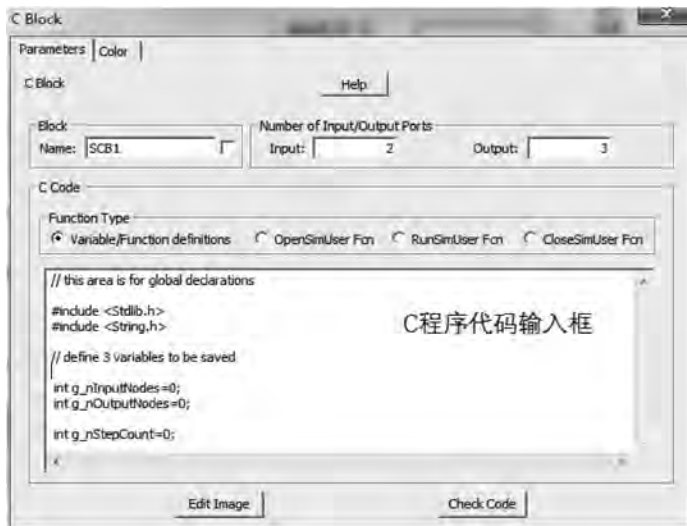


图 3-40 C Block 元件属性框

在打开的属性对话框中,在 Name、Input、Output 输入框中定义模块的名称、输入端口数、输出端口数。在 C 程序代码区域可选择 C 代码的功能类型,并在相应的 C 程序输入框内编写具体实现 C 代码程序。

C Block 元件的输入值存储在“in[n]”数组中,输出值存储在“out[n]”数组中。数组的大小 n 由 Input、Output 端口数确定。例如,对于具有 2 个输入和 3 个输出的 C Block 元件,输入节点分别为 in[0]和 in[1];输出节点分别为 out[0]、out[1]和 out[2]。类似 Simplified C Block 元件外形设置,可以单击“Edit Image”按钮对元件外形图形进行编辑。

C 程序块定义的输入/输出节点排列顺序是从上到下的,输入端口在左侧,输出端口在右侧,同时 C Block 元件外形中带点的输入引脚是第一个输入 in[0]端口。注意:未使用的输入节点必须接地,与 DLL 块不同,C 程序块的代码不可调试和逐步执行。

在定义完输入/输出端口数量,设置好元件外形图形后,即可编写 C 程序块的功能程序代码。完整的 C 代码包括“变量/函数定义”“打开仿真用户功能”“运行仿真用户功能”和“关闭仿真用户功能”四个功能类型。简单来说,C Block 程序代码包含四部分,变量或函数的声明部分、仿真开始前执行的初始化代码部分、仿真运行的具体功能实现代码部分及仿真结束后执行的代码部分。

- 选择“功能类型”下的“变量/函数定义”,在程序编辑框内定义任何需要的头文件和变量。
- 选择“功能类型”下的“OpenSimUser”,在程序编辑框输入初始化代码。在每一次仿真开始时,仅调用“OpenSimUser”代码一次,进行初始化设置。
- 选择“功能类型”下的“RunSimUser”,在程序编辑框输入具体功能的 C 程序执行代

码。PSIM 仿真时,在每个时间步都会调用此功能函数,实现具体的功能。

- 选择“功能类型”下的“CloseSimUser”,在程序编辑框输入终止执行代码。在每一次仿真结束时,将调用此函数一次。

2. C 程序块应用

为说明 C Block 元件的应用,此处以 RMS 的计算为例,设计的 RMS 功能 C Block 元件如图 3-41 所示。

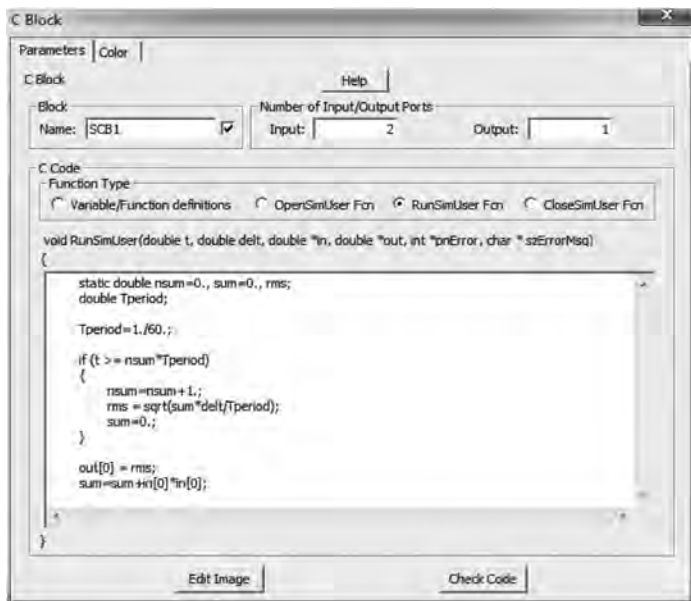
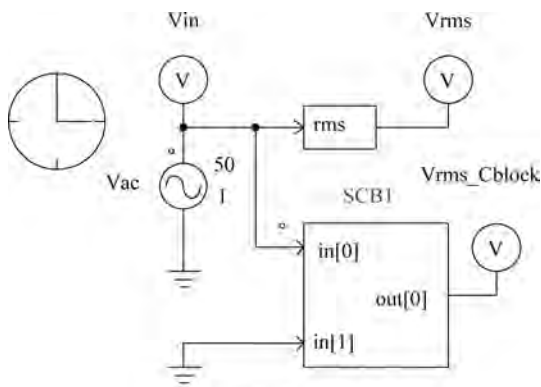


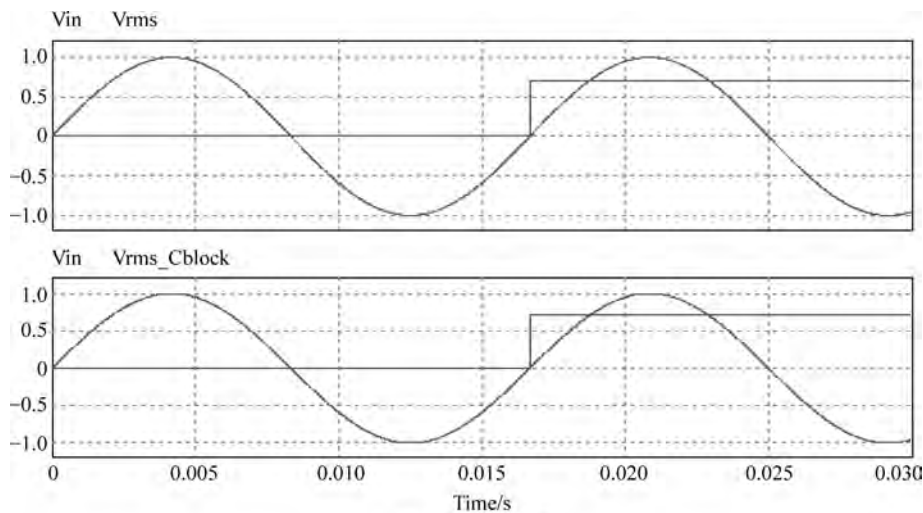
图 3-41 RMS 计算的 C Block

元件名称为 SCB1、输入端口 2 个,输出端口 1 个,RunSimUserFcn 的代码如图 3-41 所示。其他三个功能类型代码未添加,即在启动仿真、结束仿真时不执行任何功能。利用 PSIM 自带的 RMS 元件及自定义 RMS C Block 元件构建测试仿真电路模型如图 3-42(a)所示,仿真结果波形如图 3-42(b)所示,仿真结果完全一致。



(a) RMS 仿真电路模型

图 3-42 RMS 仿真



(b) RMS仿真波形

图 3-42 (续)

3.8 外部动态链接库 DLL 应用

为了增强 PSIM 对复杂电力电子系统的仿真能力,允许用户用 C/C++ 编写功能程序代码,并使用 Microsoft Visual C/C++ 将其编译成 Windows 动态链接库 DLL,然后将其链接到 PSIM 执行。PSIM 在每个仿真时间步调用该 DLL 例程,执行用户自定义的功能算法。但是,当 DLL 元件块的输入端连接到一些离散分立元件(如:零阶保持,单位延迟,离散积分器和微分器, z 域传递功能块和数字滤波器)的输出端时,PSIM 将 DLL 元件块作为离散元件,仅在离散采样时间点调用 DLL 元件块。

3.8.1 外部 DLL 元件概述

PSIM 元件库中的外部 DLL 元件提供了调用外部动态链接库 DLL 的接口,这些外部 DLL 元件可用于电源功率电路中,也可以用于控制电路中。PSIM 提供了简单 DLL 块和通用 DLL 块两种类型的 DLL 元件。简单 DLL 元件块具有固定数量的输入和输出,并且 DLL 文件名是唯一需要定义的参数;通用 DLL 元件块允许用户定义任意数量的输入/输出和附加参数,且用户还可以自定义通用 DLL 块的元件图形。

1. 简单 DLL 元件

PSIM 提供具有固定输入/输出端口数量的简单 DLL 元件模型,该简单 DLL 元件块易于编程和使用。PSIM 提供具有 1、3、6、12、20 和 25 个输入/输出端口的六种简单 DLL 元件,位于“Elements→Other→Function Blocks”菜单项下,外形如图 3-43 所示。

简单 DLL 元件节点分配是输入节点在左侧,输出节点在右侧,排序是从上到下。左侧带小圆点的输入引脚是第 1 个输入($in[0]$)端,其对应的右侧引脚为第 1 个输出引脚($out[0]$),如具有 3 个输入/输出端口的 DLL 元件,输入/输出引脚排列如图 3-44 所示。

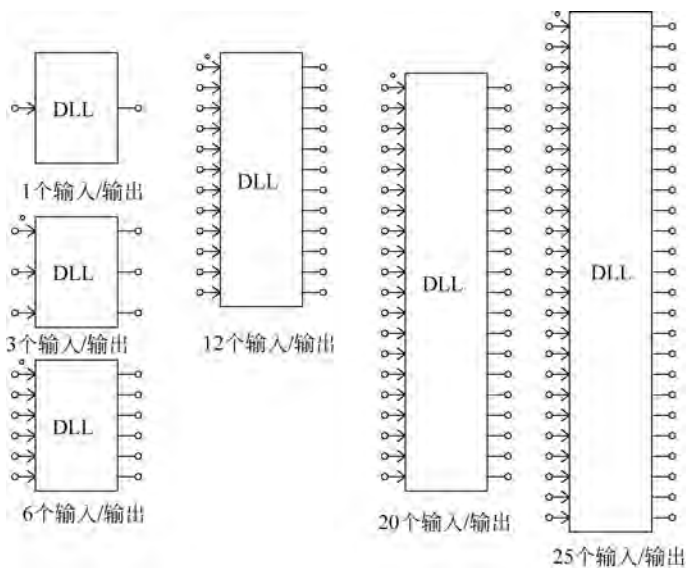


图 3-43 简单 DLL 元件外形

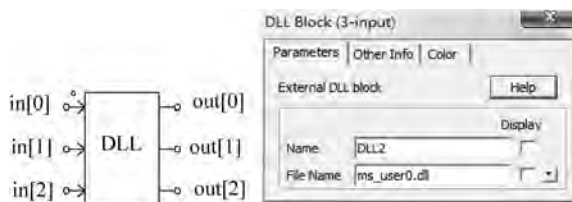


图 3-44 DLL 输入/输出端口排序及属性设置

作为具有固定数量输入、输出端口的简单 DLL 元件模块,其属性参数只有一个,为所调用的 DLL 文件名,如图 3-44 右侧属性窗口所示。Name 是 DLL 元件模块的名字,File Name 框输入的是 DLL 元件将要调用的动态链接库的文件名,如“ms-user0.dll”。注意:该参数框仅输入文件名,而不需要指定其存放路径。PSIM 会按照规定的路径进行文件查找。

DLL 元件左侧的输入端口(输入、输出是相对于 DLL 元件来说的,即将 DLL 当作一个真实元件,其具有输入、输出端口)接收与之相连的其他 PSIM 元件给出的参量值,然后对输入的参数值进行特定功能的运算处理,运算处理结束后获得当前运算结果,最后通过右侧的输出端口将 DLL 元件运算处理后的结果输出给与之相连接的其他 PSIM 元件。注意,未使用的输入节点必须接地。

被调用 DLL 文件名称可以是任意的,在创建 DLL 动态链接库时可指定文件名,也可在创建完 DLL 后手动修改为其他文件名。DLL 元件模块调用的 DLL 文件可以按照优先顺序在两个搜索路径进行文件查找。一是存放在 PSIM 目录下,二是存放在调用 DLL 文件的电路原理图保存路径下。

注意: 若在原理图中包含多个 DLL 元件块且使用同一个 DLL 文件时,如果在 DLL 代码中声明并使用了全局或静态变量,则这些全局或静态变量将相同,并且将在所有 DLL 块之间共享。如果这不是用户想要的,则 DLL 计算可能不正确,在这种情况下应避免在代码中使用全局或静态变量。

2. 通用 DLL 元件

与具有固定数量输入和输出端口的简单 DLL 块不同,通用 DLL 块提供了更大的灵活性。通用 DLL 元件模块的输入/输出端口数不固定,默认情况下由用户在其属性窗口自定义输入和输出的数量。通用 DLL 元件输入和输出的数量、节点名称以及参数的数量和参数名称也可以在 DLL 程序中定义。通用 DLL 元件模块位于“Elements→Other→Function Blocks”菜单项下,元件外形及属性窗口如图 3-45 所示。

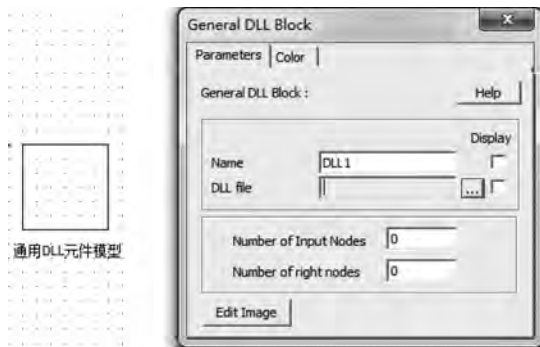


图 3-45 通用 DLL 外形及属性对话框

通用 DLL 元件默认不带任何输入、输出端口,使用者可以在属性窗口的端口数量输入框中指定输入/输出端口的数量。同时单击 DLL file 文本框旁的“...”浏览按钮,查找需要调用的 DLL 文件。单击该按钮,弹出打开文件对话框,找到并打开将要调用的 DLL 文件,即可设置通用 DLL 元件调用的动态链接库 DLL 文件。

通用 DLL 元件模型根据调用的 DLL 动态链接库文件不同,其打开 DLL 文件后的属性窗口将不同。如果在 DLL 程序代码中实现了界面属性设置函数,则打开后的界面将根据 DLL 文件的具体实现进行显示,用户也可以手动修改。在 PSIM 安装目录的“examples\custom dll”子文件夹中提供了一些简单 DLL 块和常规 DLL 块使用的 DLL 示例。读者可以打开不同的 DLL,观察属性窗口的变化。如图 3-46 所示,左图是调用 PSIM 提供的“pfc_vi_dll\pfc_vi_dll.dll”示例 DLL 的属性界面,右图是调用 PSIM 提供的“general_dll_block1\TestBlock.dll”示例 DLL 的属性界面。

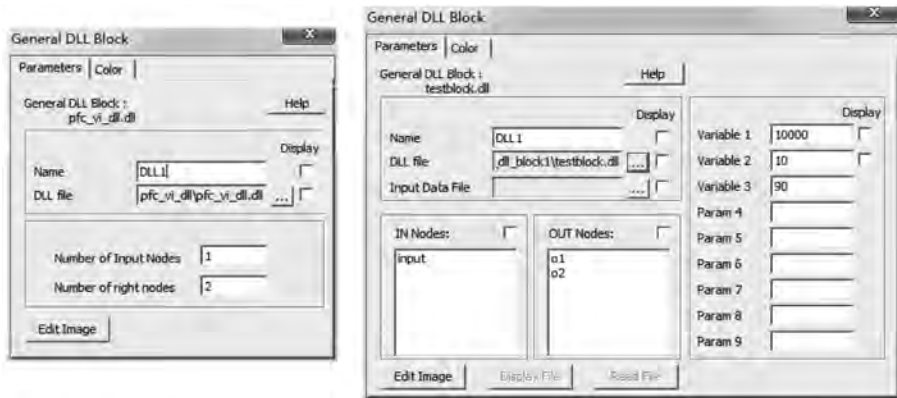


图 3-46 通用 DLL 调用不同 DLL 的属性窗口

简单 DLL 有关使用规定同样适用于通用 DLL,差别仅在参数数量可以变化,并且用户可以根据需要添加其他参数。通用 DLL 元件默认参数有 DLL File(DLL 文件名)、Number of Input Nodes(输入节点数)和 Number of Output Nodes(输出节点数),如图 3-46 左图所示。一旦选择了一个确定的 DLL 文件,会根据 DLL 文件中的定义,可能会显示 Input Data File(输入数据文件)、IN Nodes(输入节点名称)、OUT Nodes(输出节点名称)、输入 DLL 例程的 Parameter 1~ Parameter n 等可选参数,如图 3-46 右图所示。

通用 DLL 元件模块可以定义任意数量的输入和输出,定义从 PSIM 传递到 DLL 例程的任意数量的参数,并为 DLL 例程定义输入数据文件。例如,输入/输出数量和参数名称的信息可以存储在输入数据文件中,并在运行时由 DLL 例程读取。注意,输入数据文件是可选的,如果不需要输入数据文件,则可以将其留为空白。

另外,通用 DLL 可以自定义 DLL 块的外观图形,在属性对话框窗口底部的三个按钮执行以下功能:

- 编辑图像: 允许用户绘制和自定义 DLL 块的图像。
- 显示文件: 在文本编辑器中显示输入数据文件的内容。
- 读取文件: 重新读取输入数据文件。

3.8.2 外部 DLL 导出功能函数

PSIM 可调用的 DLL 有四个导出功能函数,其中 PSIM 仿真引擎使用其中的三个,另一个由用户界面使用(打开 DLL 文件时,属性窗口界面的改变由此函数实现)。在用 C/C++ 编程实现 DLL 时,需要实现这四个函数(其中一个必须实现,另外三个可以不实现)。三个 PSIM 引擎调用的函数是 RUNSIMUSER、OPENSIMUSER 和 CLOSESIMUSER,用户界面输出函数是 REQUESTUSERDATA。

1. RUNSIMUSER 函数

RUNSIMUSER 函数是 DLL 例程中唯一的强制性实现函数,其他三个函数是可选的,可以不实现。PSIM 在每个仿真时间步都会调用此函数。DLL 例程从 PSIM 接收参数值作为 DLL 的输入,随后执行计算,计算结束后将结果发送回 PSIM。输入/输出节点分配是输入节点在左侧,输出节点在右侧,其顺序是从上到下。函数的原型定义如下:

```
void RUNSIMUSER(  
    double t,  
    double delt,  
    double * in,  
    double * out,  
    void ** ptrUserData,  
    int * pnError,  
    char * szErrorMsg)
```

其中,

- double t: 仿真时间 t,单位 s,输入参量,只读。
- double del: 仿真步长,单位 s,输入参量,只读。
- double * in: 输入值数组,如果 DLL 模型有三个输入,则在程序代码中用 in[0]、in[1]和 in[2]可以访问这些输入值,输入参量,只读。

- double * out: 输出值数组, 执行计算后, 应将输出值写入此数组中。如果 DLL 模型具有四个输出, 则在程序代码中可用 out [0]、out [1]、out [2] 和 out [3] 可将值输出, 输出参量, 只写。
- void ** ptrUserData: 用户定义数据的指针, 更多信息参考函数 OPENSIMUSER, 输入/输出参量, 可读写。
- int * pnError: 成功与否指示, 成功时返回 0, 出错时返回 1。输出参量, 只写。例如, * pnError = 0; //success * pnError = 1; //error。
- char * szErrorMsg: 错误信息字符串, 如果有错误, 请将错误消息复制到此字符串。输出参量, 只写。例如, strcpy(szErrorMsg, “输入 2 不能超过 50V”)。

DLL 模块要实现的特定功能需要在 RUNSIMUSER 函数中实现, 在函数计算处理时, 可以利用 t、delt、输入数组 in[] 和 ptrUserData 等输入数据参与计算, 计算结果通过输出数组 out[] 输出, 计算过程中产生的错误信息可以通过 szErrorMsg 输出。

2. OPENSIMUSER 函数

OPENSIMUSER 是可选的, 在 PSIM 仿真开始时仅调用一次。PSIM 调用此函数从 DLL 例程接收信息, 并允许 DLL 例程分配内存供其自己使用。函数原型定义如下:

```
void OPENSIMUSER(
    const char * szId,
    const char * szNetlist,
    void * * ptrUserData,
    int * pnError,
    LPSTR szErrorMsg,
    void * pPsimParams)
```

其中,

- const char * szId: DLL 模块的 ID 字符串, 输入参量, 只读。
- const char * szNetlist: DLL 块的网表字符串, 输入参量, 只读。在 PSIM 中通过菜单“Simulate→Generate Netlist file”进行网表文件查看, 网表字符串是一系列由空格分隔的参数。常规 DLL 块的网表格式参看 PSIM 的帮助文档。
- void * * ptrUserData: 指向用户定义数据的指针, 可读写。必须在函数 OPENSIMUSER 中分配内存, 并在 CLOSESIMUSER 中释放内存。每次调用时, 它将传递给 RUNSIMUSER。它允许 DLL 在仿真过程中管理自己的数据。

注意: 此指针与函数 REQUESTUSERDATA 中用户定义的指针不同, 无法将指针从 REQUESTUSERDATA 函数传递到仿真功能函数中。它们仅通过 Netlist 线路进行通信。

- int * pnError: 成功与否指示, 成功时返回 0, 出错时返回 1。输出参量, 只写。
- char * szErrorMsg: 错误信息字符串, 输出参量, 只写。
- void * pPsimParams: EXT_FUNC_PSIM_INFO 结构体指针, 输入参量, 只读。通过该指针传入相关信息。

```
struct EXT_FUNC_PSIM_INFO
{
    char m_szPsimDir[260];           //PSIM 文件夹名称
    char m_szSchDir[260];           //文件夹下存放的 PSIM 电路图文件名(*.sch)
```

```
char m_szSchFileName[260];      //PSIM 电路图文件完整路径及文件名
};
```

在 PSIM 仿真启动时,调用此函数,传入信息到 DLL 例程或从 DLL 例程获取信息。

3. CLOSESIMUSER 函数

CLOSESIMUSER 函数是可选的,在 PSIM 仿真结束时仅调用一次,其主要目的是允许 DLL 释放已分配的任何内存或资源。函数原型定义如下:

```
void CLOSESIMUSER(
    const char * szId,
    void ** ptrUserData)
```

其中,

- const char * szId: DLL 模块的 ID 字符串,输入参量,只读。
- void ** ptrUserData: 指向用户定义数据的指针,可读写。更多信息参考函数 OPENSIMUSER。

4. REQUESTUSERDATA 函数

REQUESTUSERDATA 函数是可选的,它是与 PSIM 的处理用户界面接口。在创建通用 DLL 元件模块或在其属性框中修改其属性时,PSIM 会调用它。函数原型定义如下:

```
void REQUESTUSERDATA(
    int nRequestReason,      //描述了调用此函数时用户的操作
    int nRequestCode,        //描述了从 DLL 请求的信息
    int nRequestParam,       //此值取决于参数 nRequestCode
    void * * ptrUserData,    //指向用户定义数据的指针,它包含在每个函数调用中,
                             //允许用户管理自己的数据,内存由用户分配和释放

    int * pnParam1,
    int * pnParam2,
    char * szParam1,
    char * szParam2)
```

int * pnParam1, int * pnParam2, char * szParam1, char * szParam2: 这些参数取决于 nRequestReason, nRequestCode 和 nRequestParam 的值。

3.8.3 外部 DLL 创建

一些复杂的控制算法,使用 C/C++ 编程非常容易实现。为了能在 PSIM 中调用该控制算法,必须将控制算法程序代码封装成动态链接库。本节以 Visual Studio 2010 为开发环境,介绍 PSIM 可调用 DLL 动态链接库的创建方法。有关 Visual Studio 2010 开发环境安装不属于本书介绍的内容,读者可查阅有关文献进行了解。

1. 创建 DLL 工程

► 启动 Visual Studio 2010 开发环境,如图 3-47 所示。单击“New Project...”创建新工程,并弹出如图 3-48 所示的项目设置对话框。



图 3-47 Visual Studio 2010 启动界面



图 3-48 项目设置对话框

- 在图 3-48 项目设置对话框中选择“MFC DLL”项,在 Name 文本框输入项目名称,在 Location 文本框选择项目保存的路径。本项目保存路径设置为“D:\PSIMProject\”,项目名称设置为 TESTPSIMDLL。设置完成后单击“OK”按钮弹出 MFC DLL Wizard 欢迎信息,单击“NEXT”按钮进入下一步。
- 在弹出的应用设置对话框中选择“Regular DLL using shared MFC DLL”项,如图 3-49 所示,创建常规动态链接库 DLL,随后单击“Finish”按钮,完成项目创建。

2. 编写 DLL 程序代码

- 在完成新建项目创建后,Visual Studio 2010 进入项目,出现如图 3-50 所示的项目解决方案资源管理器。

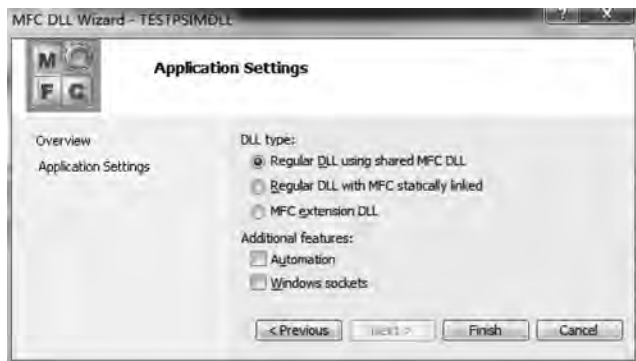


图 3-49 DLL 类型选择

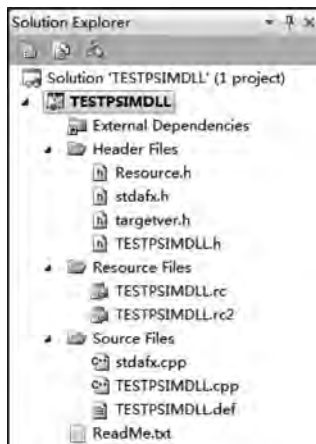


图 3-50 项目解决方案资源管理器

- 在项目解决方案资源管理器的 Header Files 文件夹下双击“TESTPSIMDLL.h”头文件,在编辑窗口中打开该文件。删除该文件中由 Visual Studio 2010 创建的所有程序代码(注意是全部删除)。
- 在“TESTPSIMDLL.h”头文件中输入 DLL 导出函数的声明代码,具体代码如下:

```
#ifndef __TESTPSIMDLL_H__
#define __TESTPSIMDLL_H__
#ifdef TESTPSIM_DLL_CPP
    #define TESTPSIM_DLL_API __declspec(dllexport)
#else
    #define TESTPSIM_DLL_API __declspec(dllimport)
#endif
#ifdef __cplusplus
extern "C"{
#endif
TESTPSIM_DLL_API void RUNSIMUSER(double t, double delt, double *in,
    double *out, void * * ptrUserData, int *pnError, char * szErrorMsg);
#ifdef __cplusplus
}
#endif
#endif
```

输入完代码的头文件如图 3-51 所示。此段代码主要是声明 PSIM 仿真引擎调用的 RUNSIMUSER 函数,其他三个函数未实现,读者可自行研究,本书不再赘述。代码中包含了一些条件编译选项,告诉编译器 RUNSIMUSER 函数声明是一个标准 C 写成的库文件,同时使用 TESTPSIM_DLL_API 将函数声明为导出函数。

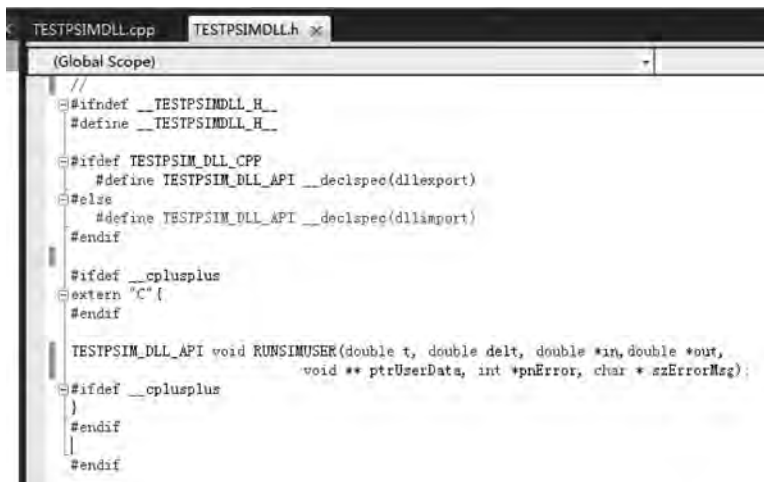


图 3-51 TESTPSIMDLL.h 头文件代码

- 在项目解决方案资源管理器的 Source Files 文件夹下,双击“TESTPSIMDLL.cpp”实现文件,在编辑窗口中打开该文件。删除该文件中由 Visual Studio 2010 创建的所有程序代码(注意是全部删除)。
- 在“TESTPSIMDLL.cpp”实现文件中输入 RUNSIMUSER 函数的具体实现代码,如下:

```

#include "stdafx.h"
#ifdef TESTPSIM_DLL_CPP
    #define TESTPSIM_DLL_CPP
#endif
#include "TESTPSIMDLL.h"
extern "C" void RUNSIMUSER(double t, double delt, double * in, double * out,
    void ** ptrUserData, int * pnError, char * szErrorMsg)
{
    out[0] = in[0] + 10;
    return ;
}

```

RUNSIMUSER 函数的具体功能实现代码根据具体功能编写,本示例是将输入值加上 10,再将结果输出,仅用于演示 DLL 的创建。

3. 编译生成 DLL 动态链接库

在完成代码编程后保存项目,然后编译生成 DLL 动态链接库。项目编译完成后生成的 DLL 文件如图 3-52 所示。其中 TESTPSIMDLL.dll 是 PSIM 可以调用的 DLL 文件。

前面介绍了 DLL 创建及生成的过程,代码编写主要是根据 PSIM 引擎将要调用的 RUNSIMUSER 函数的输入、输出参数进行具体功能实现。PSIM 导出的其余三个函数类似 RUNSIMUSER 函数进行函数声明和实现,读者可以自行研究。



图 3-52 生成的 DLL 文件

3.8.4 简单 DLL 元件应用示例

PSIM 提供的具有固定输入/输出端口数量的简单 DLL 元件,在调用 DLL 动态链接库时仅需要设置 DLL 的文件名即可。本节以 3.8.3 节创建的 DLL 为例,介绍简单 DLL 的调用过程。

1. 1 个输入/输出简单 DLL 调用

启动 PSIM 仿真软件,添加 PSIM 电路模型,并在原理图中添加具有 1 个输入/输出的简单 DLL 元件(DLL Block (1-input)),创建的电路模型及仿真波形如图 3-53 所示。模型中直流电源设置为 4V,并连接到 DLL 的输入端口,DLL 的输出端口通过一个电阻连接到地。

从仿真结果可知,TESTPSIMDLL.dll 实现的功能完全正确,说明创建的 TESTPSIMDLL.dll 通过简单 DLL 元件模型被 PSIM 正确调用,并正确执行。

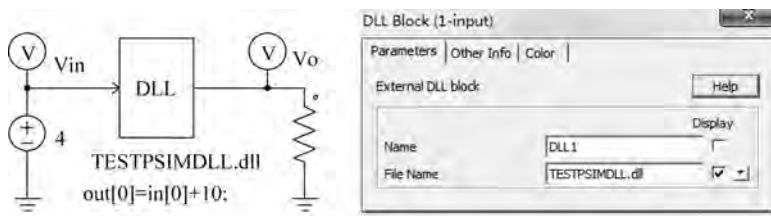
2. 3 个输入/输出简单 DLL 调用

将 3.8.3 节中 RUNSIMUSER 函数的实习代码改成如下代码:

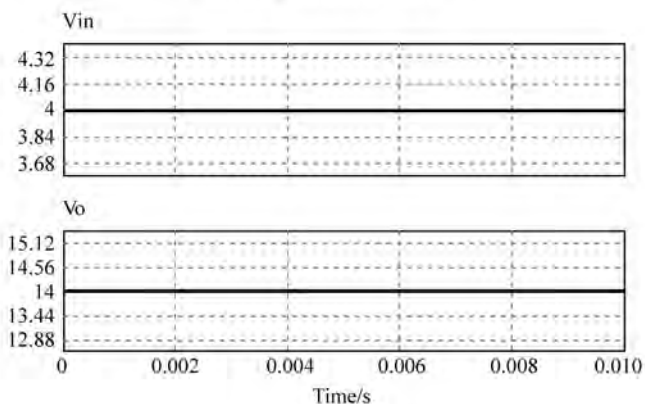
```

out[0] = in[0] + in[1] + in[2];    //将 3 个输入相加,并从第 1 个输出端口输出
out[1] = (in[0] + in[1]) * in[2]; //将第 1 个和第 2 个输入相加,再乘以第 3 个输入,
                                   //并从第 2 个输出端口输出
out[2] = in[0] * (in[1] + in[2]); //将第 2 个和第 3 个输入相加,再乘以第 1 个输入,
                                   //并从第 3 个输出端口输出

```



(a) 仿真电路模型及属性设置



(b) 仿真波形

图 3-53 简单 DLL 调用 TESTPSIMDLL.dll 模型及仿真

修改完代码后,在项目属性中,设置编译输出 DLL 名字为“PSIMDLL33”,如图 3-54 所示,随后编译生成 DLL 动态链接库。

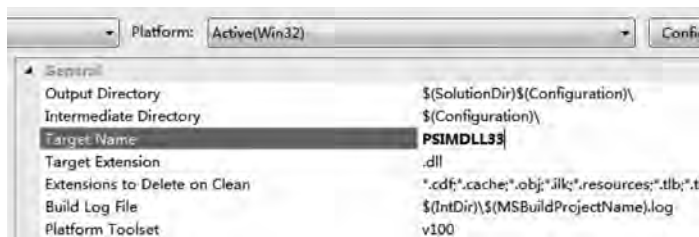


图 3-54 修改生成 DLL 的名称

新建一个 PSIM 仿真电路模型,并在原理图中添加具有 3 个输入/输出的简单 DLL 元件(DLL Block (3-input)),创建的电路模型及仿真波形如图 3-55 所示。

从仿真结果可知,PSIMDLL33.dll 实现的功能仿真完全正确,说明创建的 PSIMDLL33.dll 通过简单 DLL 元件模型被 PSIM 正确调用,并正确执行。

3. DLL 动态链接库实现 RMS 测量

将 3.7.2 节用 C 程序块实现的 RMS 功能用外部 DLL 实现。将 3.8.3 节中 RUNSIMUSER 函数实现代码改成如下代码,注意在头文件中加入 include “math.h”数学库函数头文件。

```
static double nsum = 0.0, sum = 0.0, rms = 0;
double Tperiod, freq;
freq = in[1];
```

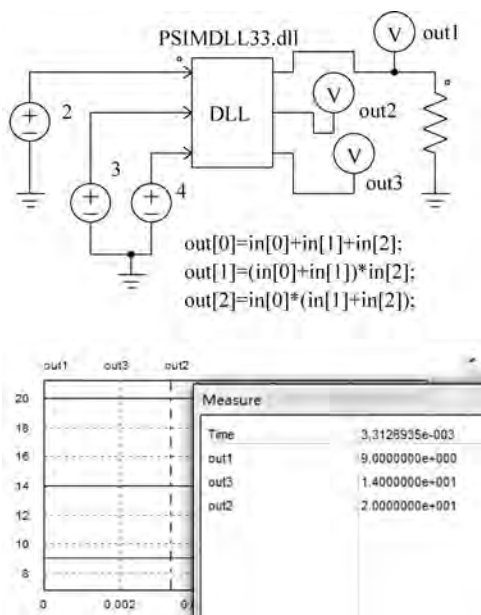


图 3-55 简单 DLL 调用 TESTPSIMDLL33.dll 模型及仿真

```
Tperiod=1.0/freq;
if (t >= nsum * Tperiod)
{
    nsum = nsum + 1.0;
    rms = sqrt(sum * delt/Tperiod);
    sum = 0.0;
}
out[0] = rms;
sum = sum + in[0] * in[0];
```

代码中第 1 个输入端口 in[0]是信号输入端口,第 2 个输入端口是信号的频率值。求出的 RMS 从第 1 个输出端口输出。编辑完 RMS 实现代码后,将输出 DLL 文件取名为“testdll_rms_freq.dll”,编译生成 testdll_rms_freq.dll 库文件。随后在 PSIM 中建立 RMS 仿真电路模型,模型及仿真结果如图 3-56 所示。

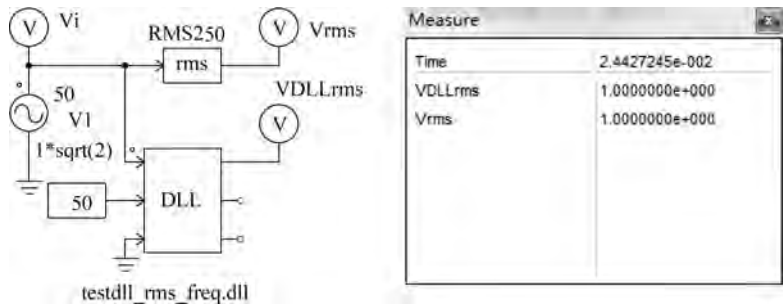


图 3-56 RMS 实现 testdll_rms_freq.dll 仿真测试

从图 3-56 仿真测量结果可知,编写的 DLL 实现了 RMS 测量,与 3.7.2 节采用 C 程序块实现的效果完全一样。testdll_rms_freq.dll 的实现代码要求 2 个输入、1 个输出,简单

DLL 元件中没有 2 输入 1 输出的 DLL 元件模块,但可以用输入/输出端口大于要求端口数量的简单 DLL 元件模型进行调用,将多余的输入端口接地,多余的输出端口悬空即可。

3.8.5 通用 DLL 元件应用示例

通用 DLL 比简单 DLL 具有更多的灵活性,本节利用通用 DLL 代替简单 DLL,对 3.8.4 节的 DLL 进行调用,讲解通用 DLL 的具体使用。

1. 具有 1 个输入/输出的 DLL 调用

将 3.8.4 节中创建的 TESTPSIMDLL.dll 用通用 DLL 元件进行调用,建立的仿真电路及仿真波形如图 3-57 所示。

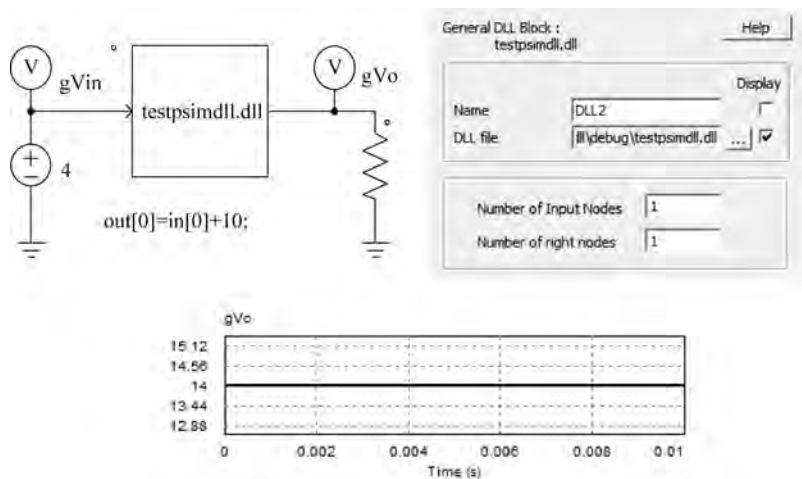


图 3-57 通用 DLL 调用 TESTPSIMDLL.dll 仿真

从图 3-57 仿真结果可知,与图 3-53 仿真结果一致,说明通用 DLL 与简单 DLL 实现的功能一致。

2. DLL 动态链接库实现 RMS 测量

3.8.4 节生成的 testdll_rms_freq.dll 库文件具有 2 个输入、1 个输出,采用通用 DLL 元件模型建立仿真电路模型并仿真,如图 3-58 所示。

从图 3-58 仿真测量结果可知,与图 3-56 仿真结果一致,说明通用 DLL 与简单 DLL 调用功能完全一致。

3. 全局变量或静态变量测试

在 3.8.1 节简单 DLL 元件应用中提到,若在原理图中包含多个 DLL 元件块且使用同一个 DLL 文件时,如果在 DLL 代码中声明并使用了全局变量或静态变量,则多个 DLL 元件块所调用的 DLL 文件将使用相同的全局变量或静态变量,并且将在所有 DLL 块之间共享。在 3.8.4 节生成的 testdll_rms_freq.dll 中包含有静态变量“static double nsum=0.0, sum=0.0, rms=0;”的定义,若在同一个 PSIM 原理图文件放置两个 DLL 元件块(简单或者通用),并同时调用 testdll_rms_freq.dll,模型及仿真结果如图 3-59 所示。

从图 3-59 的仿真测量结果可知,Vrms 和 gVrms 是使用 PSIM 的标准元件库中 RMS 元件测量的结果,完全正确;VDLLrms 是使用简单 DLL 元件调用 testdll_rms_freq.dll 的

结果,gVDLLrms 是使用通用 DLL 元件调用 testdll_rms_freq.dll 的结果,其结果不正确,因为它们共用了静态变量。

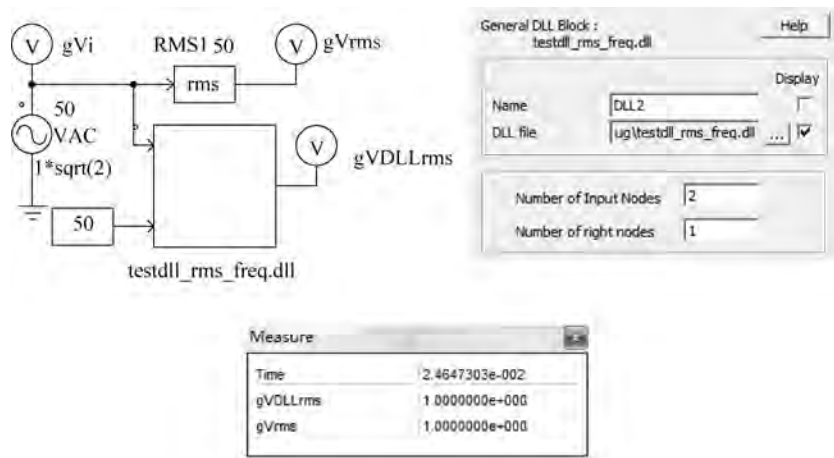


图 3-58 通用 DLL 调用 testdll_rms_freq.dll 仿真

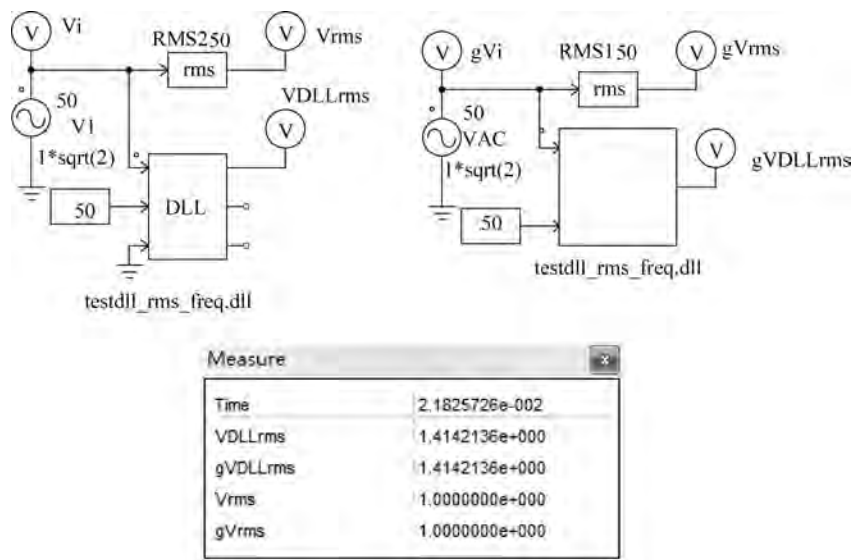


图 3-59 同一个带全局变量或静态变量的 DLL 被多个 DLL 元件同时调用

3.9 PSIM 与 MATLAB 协同仿真

MATLAB 是 MathWorks 公司开发的大型科学计算与数学软件,用于算法开发、数据可视化、数据分析以及数值计算等。Simulink 是 MATLAB 中的一种可视化仿真工具,是一种基于 MATLAB 的框图仿真环境,可实现动态系统建模、仿真和分析,被广泛应用于线性系统、非线性系统、连续系统、离散系统、连续离散混合系统、数字控制及数字信号处理的建

模和仿真。

MATLAB/Simulink 包含大量的计算算法、数学运算函数和专用工具箱,在控制系统设计与仿真方面具有不可替代的地位。PSIM 在电源功率电路建模与仿真方面具有独特的一面,但在控制系统设计与仿真方面不及 MATLAB/Simulink。为充分利用 PSIM 在电源功率仿真和 MATLAB/Simulink 在控制仿真的各自优势,PSIM 软件开发了 SimCoupler 模块,实现 PSIM 软件与 MATLAB/Simulink 的协同仿真。通过 SimCoupler 模块,系统仿真的一部分可以在 PSIM 中实现和模拟,而其余部分可在 MATLAB/Simulink 中实现,以充分发挥各自的优点,充分互补,实现更为复杂的电力电子系统仿真。PSIM 仿真软件的 SimCoupler 模块支持 MATLAB/Simulink Release 13 或者更高版本。

3.9.1 SimCoupler 接口模块概述

SimCoupler 模块是 PSIM 软件的附加模块,提供了 PSIM 和 MATLAB/Simulink 之间的协同仿真接口。SimCoupler 接口由 PSIM 中的链接节点和 MATLAB/Simulink 中的 SimCoupler 模型模块两部分组成,节点及模型如图 3-60 所示。

PSIM 中的链接节点位于 PSIM 元件库“Elements→Control→SimCoupler Module”菜单项下,用于为 PSIM 中构建的仿真模型提供输入、输出接口。

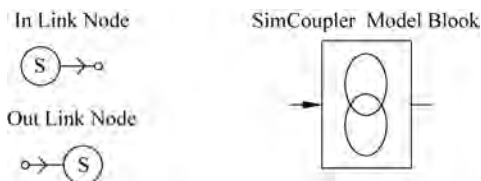


图 3-60 SimCoupler 链接节点和 Simulink 模型

- “In Link Node”元件模型是输入链接节点,用于从 MATLAB/Simulink 接收一个值,并将该值传给 PSIM 仿真模型。
- “Out Link Node”元件模型是输出链接节点,用于从 PSIM 中输出一个值给 MATLAB/Simulink, MATLAB/Simulink 仿真模型可利用接收到的值进行下一步的分析、计算及仿真。
- “In Link Node”和“Out Link Node”都是控制元件,只能在控制电路中使用。

SimCoupler Model Block 是用于 MATLAB/Simulink 中建模的模块,位于 Simulink Librarybrowser 的 S-function SimCoupler 工具箱中,SimCoupler 模型模块通过输入/输出端口连接到 Simulink 模型中。要在 MATLAB/Simulink 中找到 SimCoupler 模型模块,需要先将其加入 Simulink Library 中才能使用。加入 MATLAB/Simulink 库的方法是:

- 先启动 PSIM 软件,选择“Utilities→SimCoupler Setup”菜单项。随后弹出 SimCoupler 设置对话框,单击“Next”按钮进行下一步设置,如图 3-61 所示。示例使用的 MATLAB 版本为 2010b,在单击“Next”按钮设置时,就将 SimCoupler 添加到系统中所安装的 MATLAB 软件中,安装完成后给出安装成功提示信息,单击“确定”按钮即可。
- 将 SimCoupler 添加到 MATLAB/Simulink 库的操作,只需执行一次即可,除非需要关联其他 PSIM 版本的 SimCoupler 模型模块。
- 如果在同一台计算机上安装有多个 PSIM 软件版本,当你从一个 PSIM 版本切换到另一个 PSIM 版本进行仿真时,必须重新运行相应版本的“Utilities→SimCoupler Setup”进行

- 设置(即需要将相应版本的 SimCoupler 添加到 MATLAB/Simulink 库中)。
- 选择“Utilities→SimCoupler Setup”菜单项,实际是调用 SetSimPath.exe 程序进行 SimCoupler 设置。SetSimPath.exe 程序位于 PSIM 安装目录下。

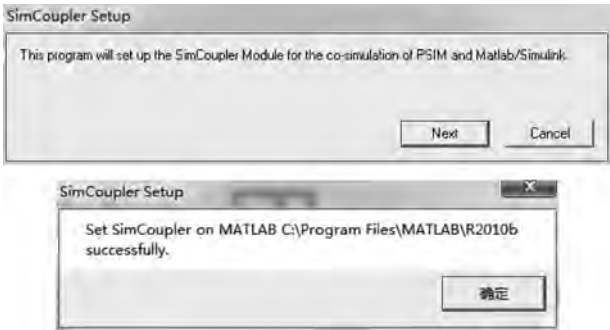
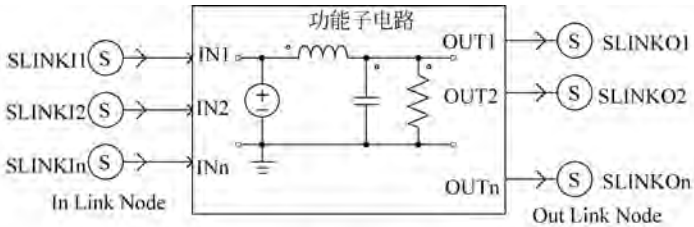
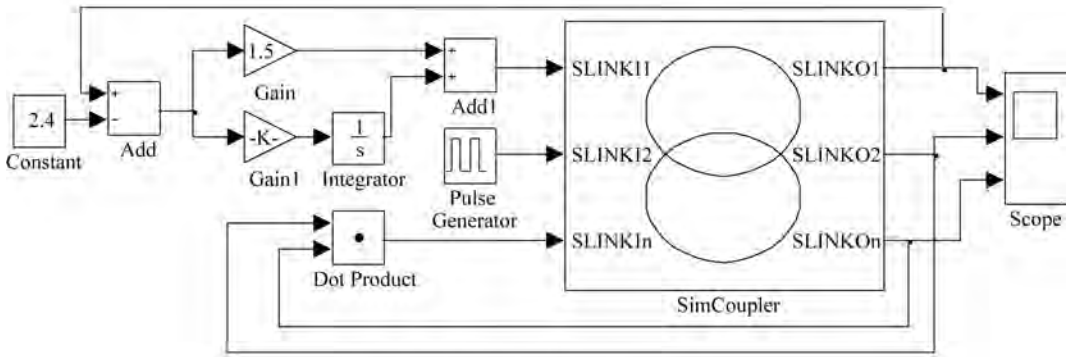


图 3-61 添加 SimCoupler 到 Simulink 库

利用 SimCoupler 进行协同建模时,两部分模型分别在各自的仿真平台构建,各自模型构建的框图如图 3-62 所示。



(a) PSIM中建立的模型示意框图



(b) Simulink中建立的模型示意框图

图 3-62 各模型构建框图

图 3-62(a)是在 PSIM 中建立的仿真电路模型,其中功能子电路是具体实现的功能电路,SLINKI1、SLINKI2、SLINKIn 是功能子电路的输入节点,SLINKO1、SLINKO2、SLINKOn 是功能子电路的输出节点。该功能电路通过 In Link Node 输入链接节点“SLINKI1、SLINKI2、SLINKIn”从 MATLAB/Simulink 获得输入值,然后功能子电路进行运算,将运算后需要输

出给 MATLAB/Simulink 的值从 Out Link Node 输出节点“SLINKO1、SLINKO2、SLINKOn”输出。

图 3-62(b)是在 MATLAB/Simulink 中建立的仿真模型,其中 SimCoupler 模型块是载入图 3-62(a)所建的 PSIM 模型后自动生成的模型块,模型块的输入端口对应 PSIM 模型的输入链接节点,模型块的输出端口对应 PSIM 模型的输出链接节点。换句话说, MATLAB/Simulink 中的 SimCoupler 模型块就是 PSIM 建立的模型,与 PSIM 模型等同。 MATLAB/Simulink 通过 SimCoupler 模型块将 PSIM 仿真电路模型包含进 MATLAB/Simulink 的仿真模型中。

在 MATLAB/Simulink 搭建完仿真模型后,就可以在 MATLAB/Simulink 中设置仿真参数,并运行仿真。在 MATLAB/Simulink 仿真结束后, MATLAB/Simulink 模型中要求观察的参量在 MATLAB/Simulink 中进行显示;同时 PSIM 的 Simview 将自动启动,在 PSIM 模型中设置了观察探头的参量,将在 Simview 中显示。

3.9.2 SimCoupler 协同仿真步骤

SimCoupler 模块的使用既简单又直接,利用 PSIM 与 MATLAB/Simulink 进行电力电子系统协同仿真,一般功率电路模型在 PSIM 中构建,系统控制电路模型在 MATLAB/Simulink 中构建。以下是 PSIM 与 MATLAB/Simulink 进行协同仿真的具体步骤。

1. 将 SimCoupler 模块添加到 Simulink 库中

在 PSIM 软件中选择“Utilities→SimCoupler Setup”菜单项,调用 SetSimPath. exe 程序将 SimCoupler 模块添加到 Simulink 库中,并设置 SimCoupler 模块以对 PSIM 和 MATLAB/Simulink 进行协同仿真。执行后, SimCoupler 块在 Simulink 库浏览器中的“S-function SimCoupler”工具箱中。需要注意:

- ◇ 此步是必需的,否则 MATLAB/Simulink 将不能与 PSIM 协同仿真。
- ◇ 此步只需要运行一次,一旦运行过此步后,在后续进行其他模型的协同仿真时,此步可以直接跳过,不操作。
- ◇ 在 PSIM 文件夹或 MATLAB 文件夹改变时,必须重新运行此步,才能再次进行协同仿真。

2. 在 PSIM 中构建所需部分仿真电路模型

启动 PSIM 仿真软件,首先将需要在 PSIM 中实现的部分电路进行建模;其次,在电路模型构建完成后,添加输入、输出链接节点,并对各链接节点命名;最后,保存该电路模型到项目文件夹中。

如果有一个以上的 In Link Node 或 Out Link Node 节点,可以设置这些节点在 MATLAB/Simulink 的 SimCoupler 模型块中出现的顺序。选择 PSIM 的“Simulate→Arrange SLINK Nodes”选项,弹出如图 3-63 所示的对话框。

图 3-63 对话框中 SLINK In 栏列出了所有的输入链接节点, SLINK Out 栏列出了所有的输出链接节点。可以选中某一个节点,单击左侧或者右侧的上/下箭头可调整该节点在列表中的位置。链接节点在 SLINK In 栏和 SLINK Out 栏中从上到下的排列顺序,与 MATLAB/Simulink 的 SimCoupler 模型块中节点出现的顺序一致。输入节点将在 SimCoupler 模型块的左侧,从顶部到底部按照 SLINK In 栏中的顺序排列;输出节点将在

SimCoupler 模型的右侧,从顶部到底部按照 SLINK Out 栏中的顺序排列。



图 3-63 输入/输出链接节点排列顺序调整对话框

3. 在 MATLAB/Simulink 中构建剩余部分仿真电路模型

启动 MATLAB,进入 Simulink。打开已有的 Simulink 模型文件或者新建一个 Simulink 模型文件,构建剩余部分仿真模型(去除已经在 PSIM 中构建了的部分)。

4. 添加 SimCoupler 模型块到电路原理图中

Simulink 库浏览器中的“S-function SimCoupler”工具箱中,将 SimCoupler 模型拖放到 Simulink 模型文件中。

5. 加载 SimCoupler 模型块的 PSIM 仿真电路模型

在 Simulink 原理图中双击 SimCoupler 模型块,在弹出的对话框中单击“Browse...”按钮,找到并选择第 2 步创建的 PSIM 原理图文件,然后单击“Apply”按钮,SimCoupler 模型块的输入和输出端口数将自动匹配成 PSIM 中设置的链接节点数。

6. 将 SimCoupler 模型块连接到 Simulink 电路模型中

加载 PSIM 仿真模型后,SimCoupler 模型块出现了输入、输出端口,将 SimCoupler 模型块的输入、输出端口与 Simulink 模型电路连接起来,形成完整的仿真模型。

7. 设置 Simulink 的仿真参数

转到 Simulink 仿真参数设置窗口,配置仿真参数。需要设置仿真时间、求解器类型及仿真步长。若求解器类型选择 Fixed-step(固定步长),则将固定步长设置为与 PSIM 的时间步长相同或接近的值;若求解器类型选择 Variable-step(可变步长),得到的仿真结果将不正确。为了获得正确的结果,必须在 SimCoupler 模型块的输入端放置零阶保持器,且该零阶保持器的采样时间必须与 PSIM 时间步长相同或接近。

8. 在 Simulink 中开始仿真

完成上述步骤后,启动 Simulink 仿真。仿真结束后,可以查看需要观察的参量数据,同时 PSIM 的 Simview 程序也会自动打开,显示相关参量曲线。

另外需要注意,在 Simulink 的反馈系统模型中使用 SimCoupler 模型块时,SimCoupler 模型块可能是代数循环的一部分。MATLAB/Simulink 的某些版本无法解决包含代数循环的系统,而其他一些版本则可以解决包含代数循环的系统,但性能降低。为了打破代数循环的限制,可在 SimCoupler 模型块的每个输出端口处放置一个存储模块。该存储模块引入了一个积分时间步长的延迟,以打破代数循环问题。

3.9.3 平均电流控制 Buck 变换器协同仿真

本节将以单环平均电流反馈控制 Buck 变换器的协同仿真为例,说明 PSIM 与 MATLAB/Simulink 的协同仿真。本示例的完整仿真模型如图 3-64 所示。模型功率电路

部分电源设置为 50V,电感为 1mH,电容为 $47\mu\text{F}$,负载电阻为 5Ω 。设置电感电流采集传感为反馈控制提供控制参量。控制电路部分,采用 PI 补偿调节控制器,将当前电感电流与设置的参考电流 I_{ref} 比较,比较后的误差量进行 PI($k_p=1.5$, $k_i=0.0001$)运算,运算结果再进行限幅处理,随后与锯齿波(频率为 20kHz、幅值为 1、占空比为 1)进行比较,产生一定占空比的 PWM 波控制功率开关管工作,实现 Buck 变换控制。

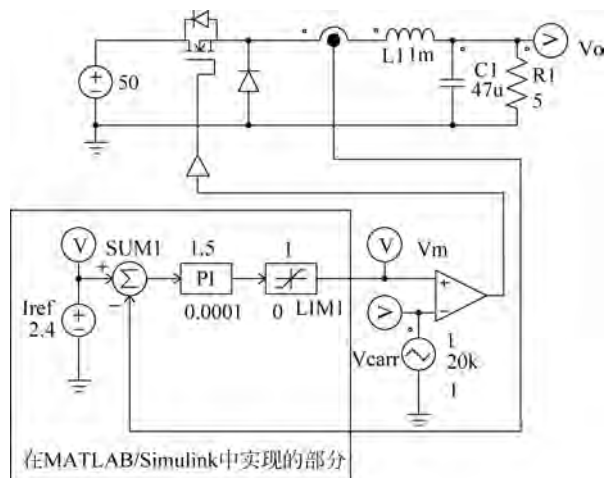


图 3-64 平均电流反馈控制电路模型

在图 3-64 的模型中,将实线框内的部分放到 MATLAB/Simulink 中实现,其余部分(功率电路及部分控制电路)在 PSIM 中实现。本实例使用的 MATLAB 版本是 R2010b。

(1) 将当前版本 PSIM 的 SimCoupler 模块添加到 Simulink 库中。

本示例使用的是 PSIM 9.1.1 版本,启动 PSIM 软件,并选择“Utilities→SimCoupler Setup”选项,将 PSIM9.1.1 与 MATLAB/Simulink 关联以进行协同仿真。注意此操作只需要执行一次,直到下次希望将其他版本的 PSIM 与 MATLAB/Simulink 进行关联以协同仿真。

(2) 在 PSIM 中新建原理图文件,构建图 3-64 中的功率电路及部分控制电路模型,并添加输入、输出链接节点,设置仿真控制参数,电路模型如图 3-65 所示。

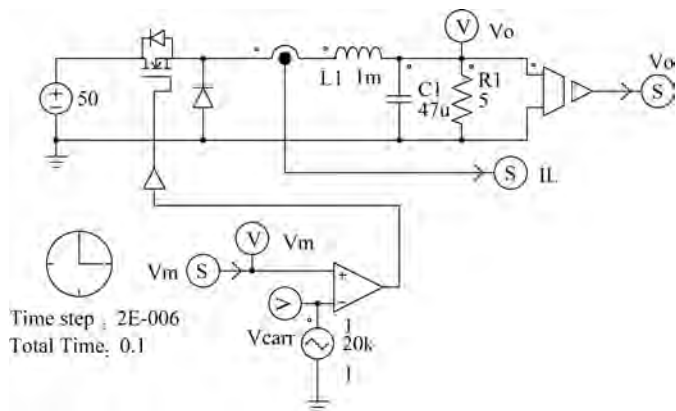


图 3-65 PSIM 中仿真电路模型

图 3-65 中的电路模型,将电流传感器(增益为 1)采集输出的电流连接到输出链接节点,并命名为 IL,将其值输出给 MATLAB/Simulink; 利用电压传感器(增益为 1)测量变换器的输出电压,并将采集输出连接到输出链接节点,并命名为 V_o ,将其值输出给 MATLAB/Simulink; 比较器的同相输入端是输入的调制信号,连接到输入链接节点,节点命名为 V_m ,将从 MATLAB/Simulink 中获得控制所需的调制信号。在 PSIM 模型中添加了 V_m 、 V_o 、 V_{carr} 的测量探头,在仿真结束后可以查看相关仿真运行数据。根据需要可以选择“Simulate→Arrange SLINK Nodes”菜单项调整链接节点的排列顺序,此处采用默认排列顺序。

模型仿真运行控制参数的仿真时间步设置为 $2E-006(2\mu s)$,总仿真时间为 $0.1s$,其他参数采用默认参数。设置完成后,将所建模型保存到项目文件夹,本示例保存路径及文件名为“I:\PSIM\test\SimCouplertest.psimsch”。

(3) 在 MATLAB/Simulink 中构建实线框中的控制电路模型,如图 3-66 所示。模型准备从 PSIM 中获得电感电流 IL,并与常量 2.4 比较,将比较的误差经 PI 运算后获得的调整控制信号输出给 PSIM。模型中的参数设置与图 3-64 完全一致。

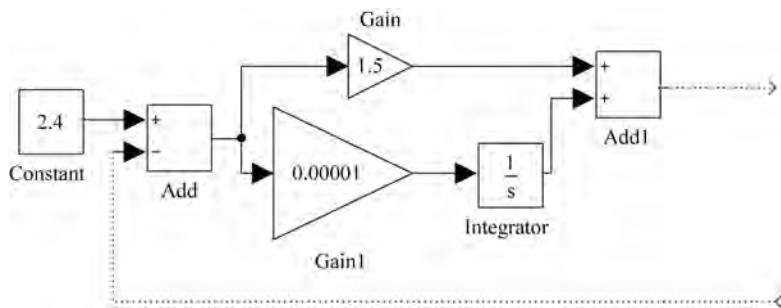


图 3-66 MATLAB/Simulink 中搭建的部分模型

(4) 在 Simulink 库浏览器中的“S-function SimCoupler”工具箱中,将 SimCoupler 模型拖放到 Simulink 模型文件中,如图 3-67 所示。

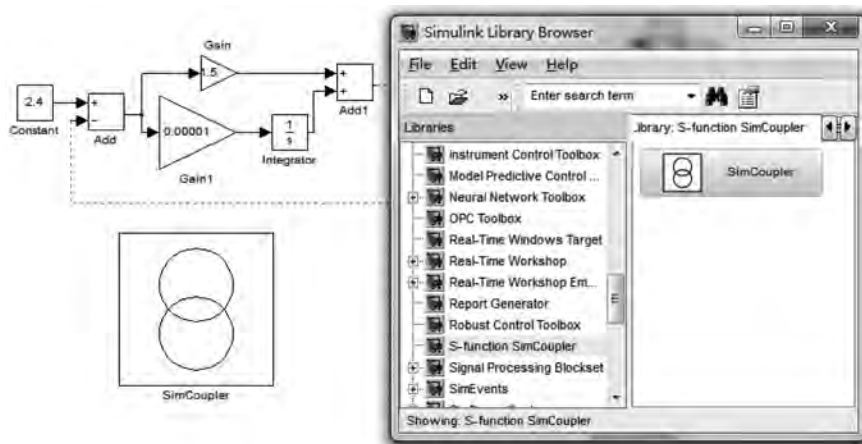


图 3-67 SimCoupler 模型拖放

(5) 双击 SimCoupler 模型块加载 PSIM 仿真电路模型,在弹出的对话框中单击“Browse...”按钮,找到并选择“I:\PSIM\test\SimCouplertest.psimsch”原理图文件,然后单击“Apply”按钮,随后关闭对话框,如图 3-68 所示。SimCoupler 模型块的输入和输出端口数将自动匹配成 PSIM 中设置的链接节点数及相应节点名称。如果之后在 PSIM 中的原理图节点数目发生变化,需要选择“Edit→Update Diagram”选项来更新 SimCoupler 模块。

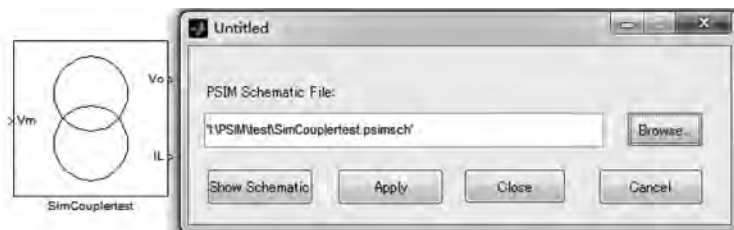


图 3-68 添加 SimCoupler 的 PSIM 电路模型

(6) 将 SimCoupler 模型块连接到 Simulink 电路模型中,形成完整的仿真电路模型,如图 3-69 所示。添加 Scope 观察器,并将输出电压、电感电流接入,仿真过程中可进行查看。

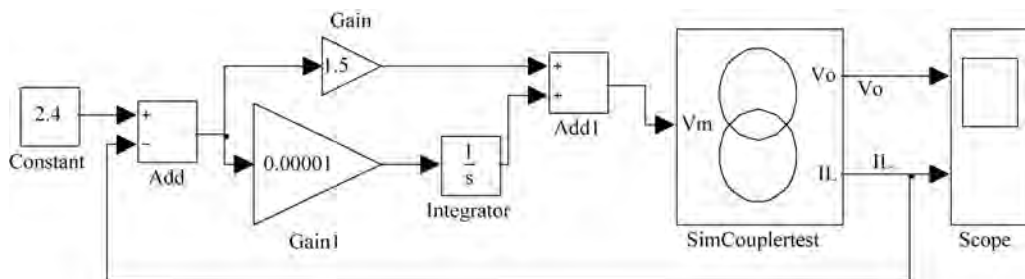


图 3-69 SimCoupler 构建的 Simulink 电路模型

(7) 设置 Simulink 的仿真参数,将仿真结束时间设置为 0.1s、求解器类型设置为 Fixed-step(固定步长)、求解器设置为 ode5,仿真步长设为 2×10^{-6} ,与 PSIM 的时间步长相同。参数设置界面如图 3-70 所示。

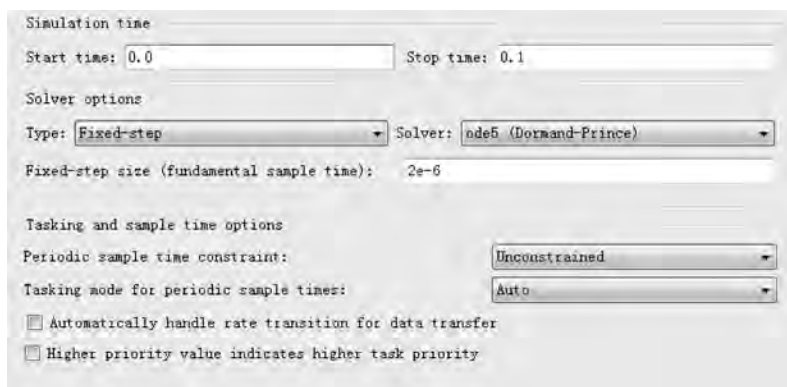


图 3-70 Simulink 仿真参数设置

(8) 完成上述步骤后启动 Simulink 仿真。仿真结束后,在 MATLAB/Simulink 查看的参量数据波形如图 3-71(a)所示,同时,自动启动 PSIM 的 Simview 波形如图 3-71(b)所示。从两者波形可知,仿真波形一致。

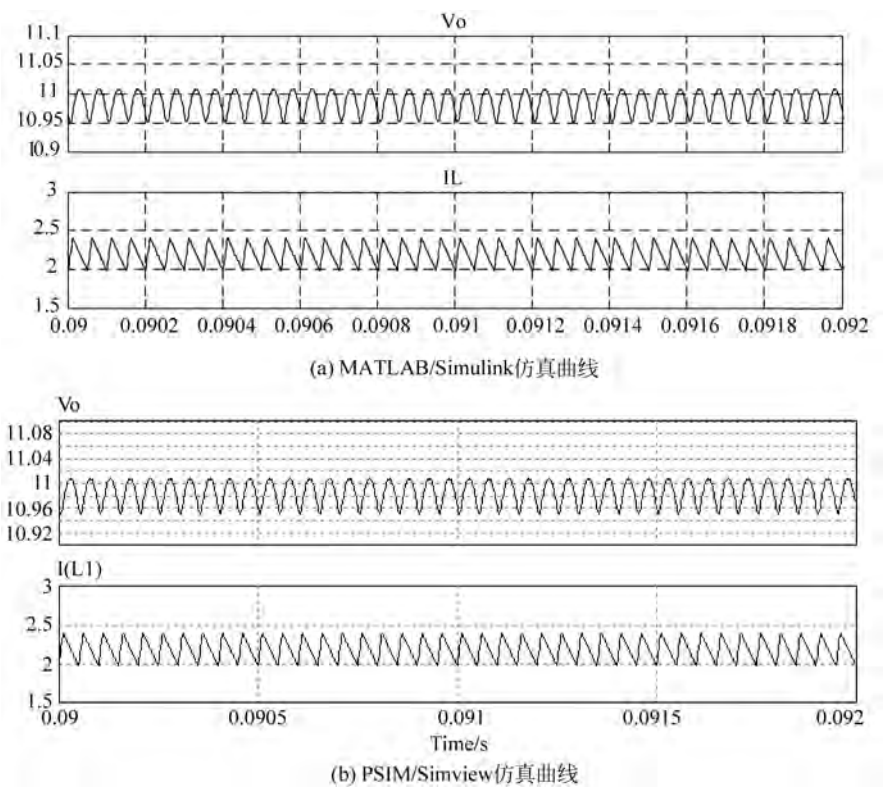


图 3-71 仿真波形

若将求解器类型选择 Variable-step(可变步长),得到的仿真结果将不正确,如图 3-72 所示。

为了获得正确的结果,必须在 SimCoupler 模型块的输入端放置零阶保持器,且该零阶保持器的采样时间必须与 PSIM 时间步长相同或接近,加入零阶保持器的采样时间设置为 $2e-6$,新的 MATLAB/Simulink 模型如图 3-73 所示,仿真结果与图 3-71 完全一致。

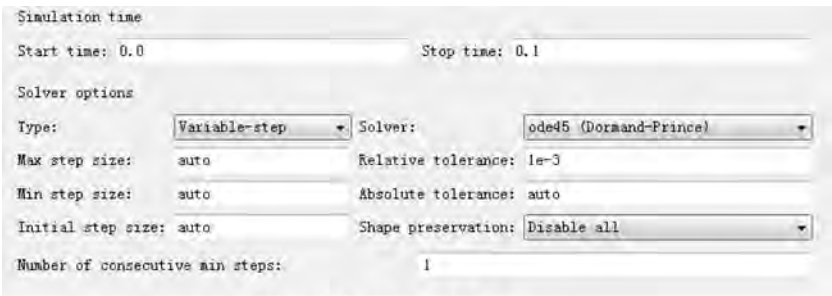


图 3-72 求解器类型选择 Variable-step 的仿真

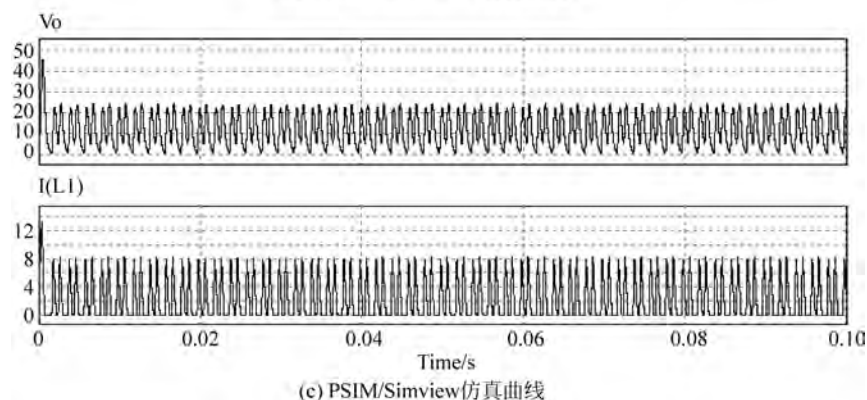
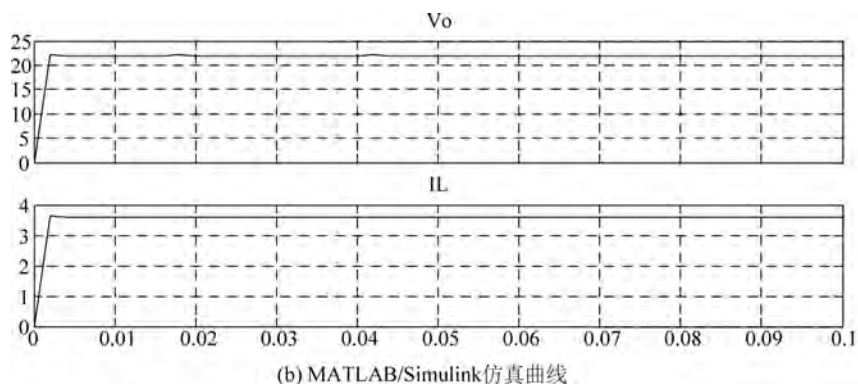


图 3-72 (续)

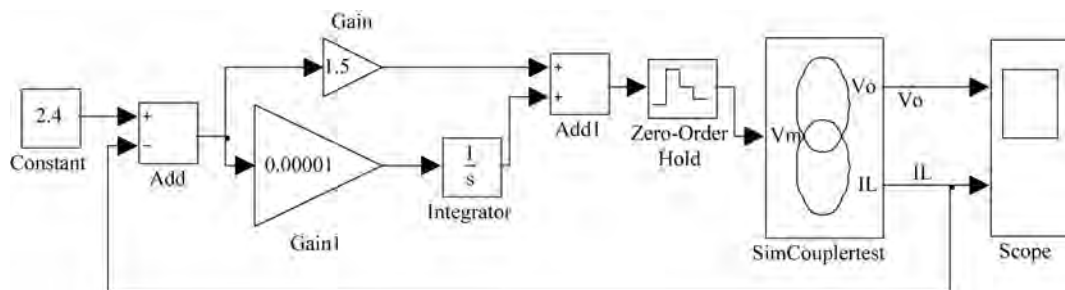


图 3-73 添加零阶保持器的仿真模型

3.10 交流频域仿真分析

通过交流频域(简称 AC)仿真分析可以获得一个电路或控制环路的频率响应特性。在 PSIM 中进行 AC 分析的一个显著特点是电路可以保持原有的开关模式,而不需要转换成平均模型,尽管通过平均模型执行 AC 分析可以节省更多的时间。

3.10.1 交流频域分析元件模型

PSIM 元件库中带有 AC 分析的元件模型,其位于“Elements→Other”和“Elements→

Other→Probes”菜单项下,各元件外形及说明如表 3-2 所示。

表 3-2 AC 分析元件模型说明

元 件 外 形	元 件 名 称	说 明
	AC Sweep Probe	交流扫描探头,用于对某个输出节点的 AC 分析
	AC Sweep Probe (loop)	环路交流扫描探头(测量环路 AC 响应)
	AC Sweep	交流扫描设置(频率响应分析参数设置)

AC 仿真分析控制由 AC Sweep 元件进行设置。在进行 AC 仿真分析前需要在电路原理图模型中的任意位置放置 AC 仿真分析控制元件(AC Sweep),并设置 AC Sweep 的仿真分析控制参数。在放置的 AC Sweep 元件上,双击弹出该元件的属性对话框,如图 3-74 所示,具体参数定义如表 3-3 所示。

表 3-3 AC Sweep 参数定义

属 性 参 数	说 明
Start Frequency	交流扫描的起始频率(Hz)
End Frequency	交流扫描的末点频率(Hz)
No. of Points	数据点数
Flag for Points	用于定义如何生成数据点的标志。标志=0: 数据点按 Log10 比例线性分布; 标志=1: 数据点以线性比例线性分布
Source Name	激励源名称
Start Amplitude	起始频率处激励源的振幅(幅值)
End Amplitude	末点频率处激励源的振幅(幅值)
Freq. for extra Points	附加数据点的频率。如果频域特性在某个频率范围内快速变化,则可以在该区域中添加额外的点以获得更好的数据分辨率

交流分析的原理是将一个小的交流激励信号(激励源)作为扰动注入系统中,并在输出点提取相同频率的信号。为了获得准确的交流分析结果,交流激励源的幅值必须设置适当。一方面,激励源的幅值必须足够小,以使扰动保持在线性区域内;另一方面,激励源的幅值又必须足够大,以使输出信号不受数值误差的影响。

通常,一个物理系统在低频段衰减较小,而在高频段衰减很大。因此,设置激励源时,最好是在低频段(起始频率处)设置一个相对较小的幅值,而在高频段(结束频率处)设置一个相对较大的幅值。

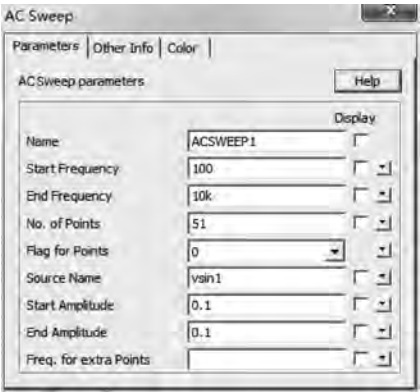


图 3-74 AC Sweep 参数设置属性对话框

另外,有时交流分析完成后会出现一个警告信息:“Warning: The program did not reach the steady state after 60 cycles. See File ‘message.txt’ for more details(警告: 60 个周期后程序未达到稳态。有关警告信息的具体内容,可查阅‘message.txt’文件以获取详细信息说明)”。该警告产生的原因是当软件在执行了 60 个周期后,交流扫描输出仍然未检测到稳态。为了解决这个问题,可以增加电路中的阻尼(包括寄生电阻)、调整激励源幅值,或者减小仿真时间步长。文件“message.txt”将提供有关发生这种情况的频率和相对误差的信息(相对误差将指示数据点距离稳定状态有多远)。

3.10.2 交流频域分析设置方法

在 PSIM 仿真中进行 AC 仿真分析非常简单,仅需要四个设置即可进行 AC 仿真分析。具体如下:

- 确定一个正弦电压源作为 AC 扫描分析的激励源(需要在仿真模型中加入一个正弦交流信号源)。
- 将交流扫描探头(AC Sweep Probe)放置在需要分析(检测点)的位置。如果要测量闭环控制系统的环路响应,需要使用节点到节点的环路交流扫描探头(AC Sweep Probe(loop))。
- 将 AC Sweep 扫描控制元件模块放置在电路原理图模型中的任意位置,并定义 AC 扫描参数。
- 运行仿真,进行 AC 仿真分析。

3.10.3 交流频域分析示例

1. Buck 变换的开环响应分析

在 PSIM 中建立一个开环 Buck 变换器模型,如图 3-75 所示。在调制信号(U_r)上注入一个激励源 SINE,在变换器的输出端进行测量,检测变换器输出电压 V_o 与调制信号 U_r 之间的开环频率响应特性。

2. 闭环电路传递函数响应分析

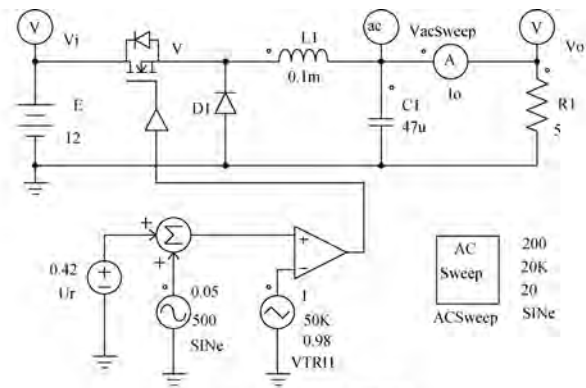
通过交流分析可以获得闭环系统环路响应特性。在 PSIM 建立一个单环平均电流反馈控制模式的 Buck 变换器,如图 3-76 所示。在电流反馈环路中注入激励信号 SINE(反馈电流和激励 SINE 相加作为反馈信号),采用环路交流扫描探头获得环路传递函数。根据环路传递函数,可以确定控制环路的带宽和相位裕度。

需注意,交流扫描探头的连接需要保证在激励源注入后,探头两端覆盖反馈环路。

3. 开关电源传递函数响应分析

由补偿调节器控制的开关电源也可进行 AC 分析。在 PSIM 中建立一个单环电压反馈控制的 Buck 变换器,补偿调节器采用 Type 3 调节器。激励源在运算放大器输出之前插入反馈路径中,如图 3-77 所示。

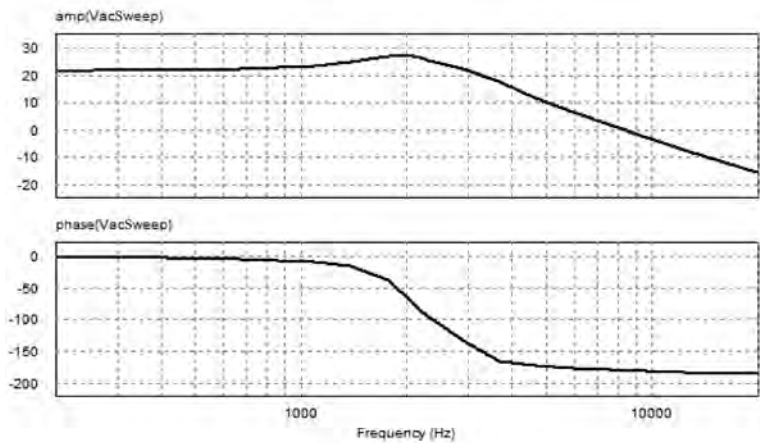
从图 3-77(c)的频率响应曲线可以看出,系统的相位裕量约为 50° ,增益裕量接近 20dB,系统是稳定系统。因此可以通过 AC Sweep 频率分析来判断闭环系统的稳定性,以辅助 DC/DC 变换器控制环路的设计。图 3-77(b)模型中,AC Sweep 也可以前移到运算放大器的输出之后,仿真曲线与图 3-77(c)的频率响应曲线一样。



(a) AC仿真分析模型

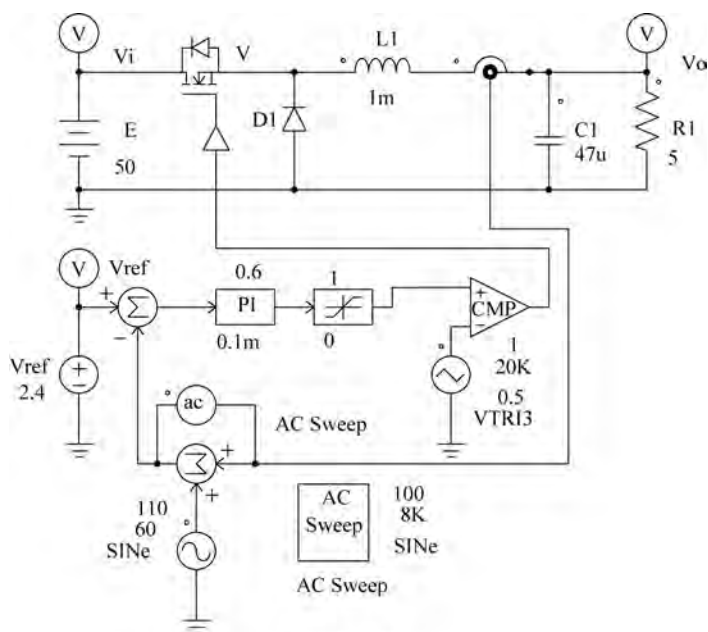


(b) AC Sweep参数设置

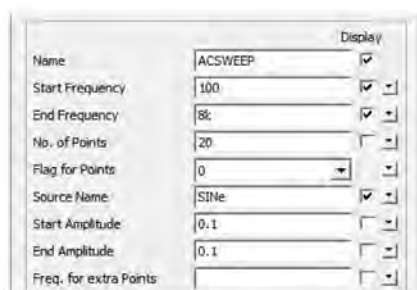


(c) 频率响应曲线

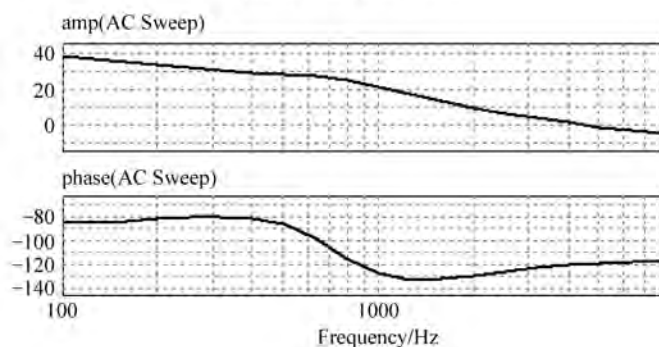
图 3-75 Buck 变换开环 AC 仿真分析



(a) 电路模型

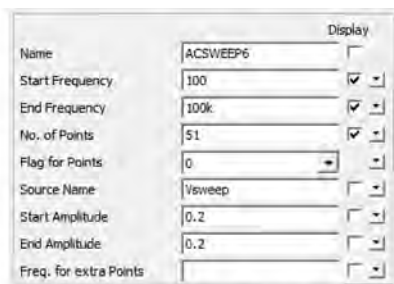


(b) AC Sweep参数设置

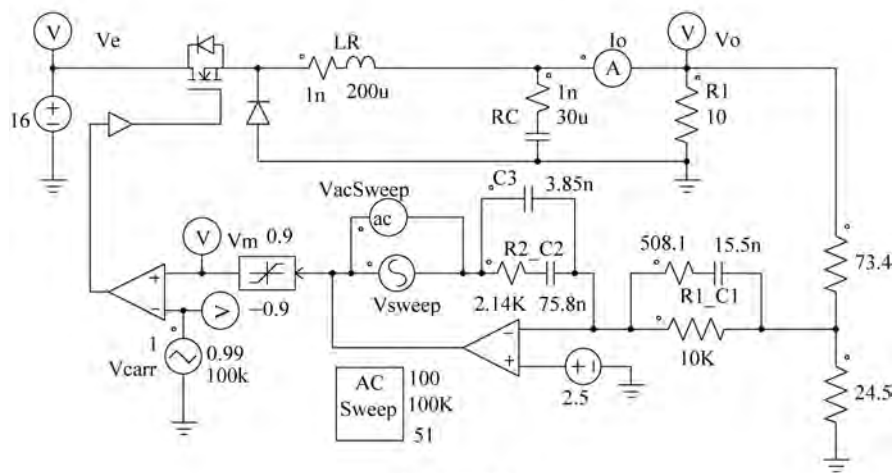


(c) 频率响应曲线

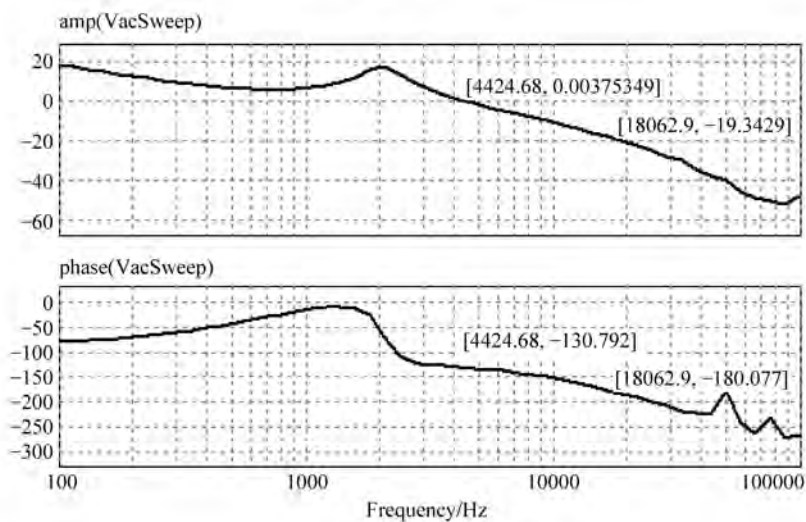
图 3-76 单环平均电流反馈控制 Buck 变换器 AC 仿真分析



(a) AC Sweep参数设置



(b) 电路模型



(c) 频率响应曲线

图 3-77 Type 3 型单环电压反馈控制 Buck 变换器 AC 仿真分析

3.11 参数扫描分析

PSIM 元件库带有参数扫描元件模型,可以通过该模型对某些元件进行参数扫描,评估参数对系统的影响。参数扫描模块位于“Elements→Other”菜单项下,元件外形如图 3-78 所示,元件属性参数如表 3-4 所示。参数扫描元件模块可扫描的参数一般包括:

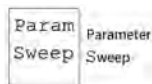


图 3-78 参数扫描元件模型

- RLC 分支的电阻、电感和电容(R、L、C);
- 比例控制器的增益(P,Proportional);
- 积分器的时间常数(I,Integrator);
- 比例积分控制器的增益和时间常数(PI, proportional-integral);
- 二阶低通和高通滤波器的增益、截止频率和阻尼比(2nd-order Low-pass Filter/2nd-order High-pass Filter);
- 二阶带通和带阻滤波器的增益、中心频率以及通带和阻带(2nd-order Band-pass Filter/2nd-order Band-stop Filter)。

表 3-4 Param Sweep 属性参数说明

参 数	说 明	参 数	说 明
Start Value	参数的起始值	Increment Step	参数步长增量值
End Value	参数的终止值	Parameter to be Swept	启用参数扫描功能

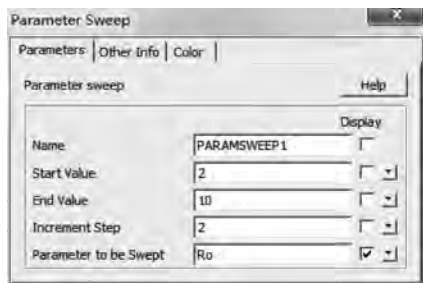


图 3-79 参数扫描属性参数设置

例如,某个电阻器“R1”的电阻设置为“Ro”。要将电阻从 2Ω 扫描到 10Ω (增量值为 2Ω)。则相应参数设置为: 参数起始值设置为 2, 参数终止值设置为 10, 参数步长增量值设置为 2, 扫描参数设置为 Ro, 并勾选“Display”复选框, 具体设置如图 3-79 所示。注意: 要扫描的参数值应该是元件的参数值, 而不是元件的名称。例如示例中, 将要扫描的参数应定义为“Ro”, 而不是元件名称“R1”, Ro 才是元件的参数值。

参数扫描分析将输出两条曲线: 一条是输出量随时间变化的曲线, 另一条是最后一个仿真点的输出与扫描参数之间的变化曲线。

例如, 一个具有两个输出变量 V1 和 V2 的电路, 对电阻 R1 的阻值 Ro 进行扫描, 总仿真时间设置为 0.1s。仿真结束后, 在 Simview 中会显示两个曲线图, 一个曲线图是 V1 和 V2 随时间变换的曲线, 另一个是 V1 和 V2 随 Ro 变换的曲线, V1 和 V2 值是 0.1s 时最后一个仿真点的值。

3.12 本章小结

本章首先对 PSIM 的元件查找与放置、仿真电路原理图设计、仿真控制、仿真结果查看与分析等基本操作进行详细的讲解与分析；随后，对 PSIM 子电路创建、元件参数文件使用、C 程序块使用、外部动态链接库 DLL 的设计与调用、PSIM 与 MATLAB 协同仿真等高级功能的使用与操作进行详细的讲解；最后，对 PSIM 电力电子仿真的频域仿真分析及参数扫描分析进行讲解。通过对本章的学习，读者能掌握 PSIM 构建仿真电路模型的基本操作方法与分析方法。