

第八章 基于 Simulink 的动态系统仿真入门

Simulink 是基于 MATLAB 的图形化仿真设计环境，是 MATLAB 提供的进行动态系统建模、仿真和综合分析的集成软件包。它使用图形化的系统模块对动态系统进行描述，并在此基础上采用 MATLAB 的计算引擎对动态系统在时域内进行求解。MATLAB 计算引擎主要对系统微分方程和差分方程求解。Simulink 和 MATLAB 是高度集成在一起的，因此它们之间可以进行灵活的交互操作。


Simulink 可以处理的系统包括：线性、非线性系统；离散、连续及混合系统；单任务、多任务离散事件系统等。在 MATLAB7 版本中，可直接在 Simulink 环境中运作的工具箱很多，已覆盖航空/航天、通信、控制、信号处理、电力系统、机电系统等诸多领域，所涉内容专业性极强。

8.1 启用 Simulink 并建立系统模型

因为 Simulink 是基于 MATLAB 中的图形化仿真环境，因此启动 Simulink 之前必须首先运行 MATLAB，然后才能启用 Simulink 并建立图形化的系统模型。有两种方式启动 Simulink：

命令行方式启动：在 MATLAB 命令窗口键入 Simulink 即可；

快捷方式启动：鼠标点击 MATLAB 工具栏图标。

启动 Simulink 后，屏幕上会出现 Simulink 主窗口，鼠标点击 Simulink 主窗口工具栏上的或从主菜单 File 中选择 New\model，即可打开系统模型编辑器。见图 8.1。图 8.1 中由上到下依次是 MATLAB 主窗口、Simulink 主窗口（即 Simulink 库浏览器）和系统模型编辑器，图中箭头反映了操作的顺序。

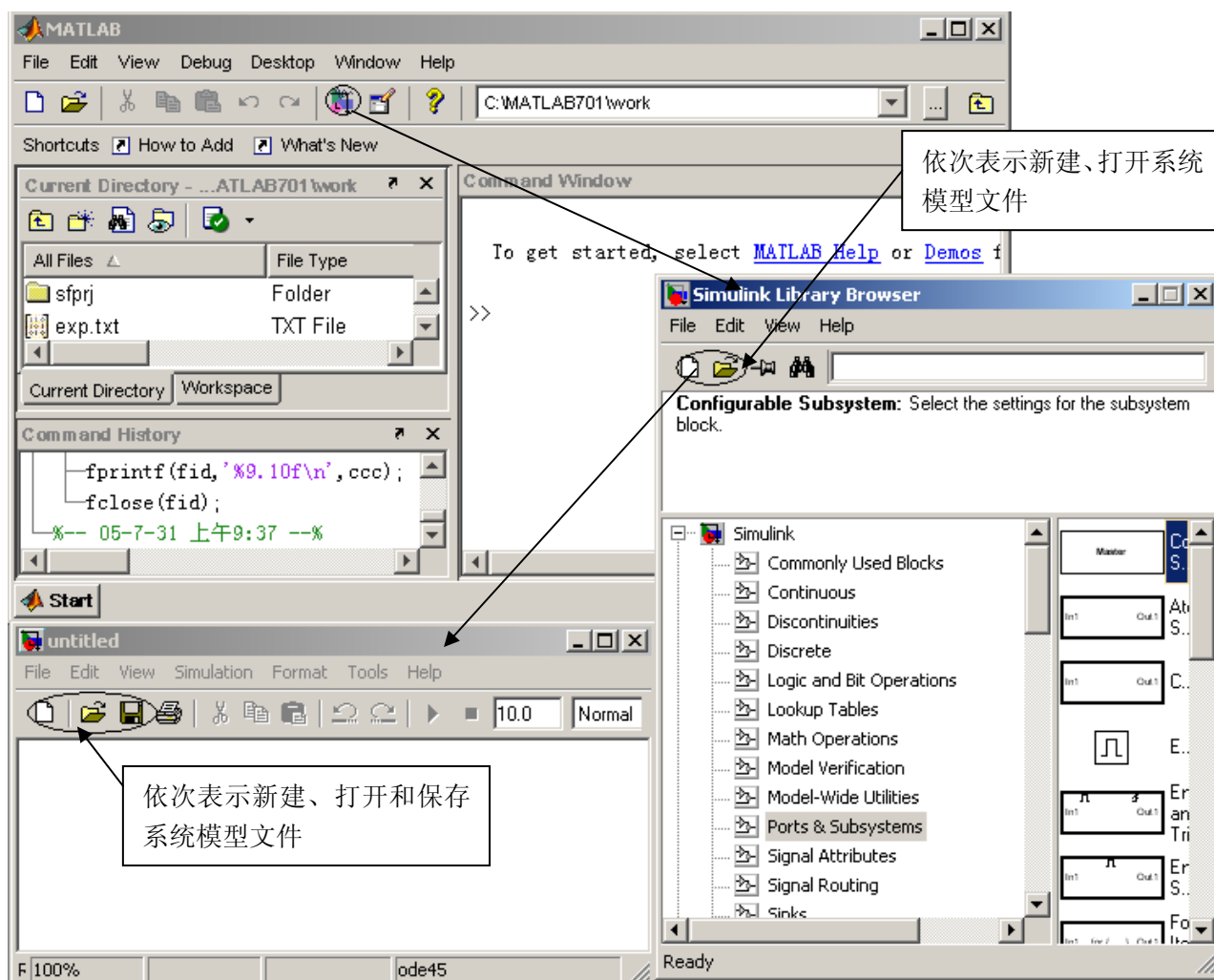


图 8.1 启动 Simulink 并建立系统模型的基本操作

当打开一个新的系统模型文件后，用户便可从 Simulink 模块库中选择合适的系统模块或自定义的模块来建立系统模型。因此，用户需先了解 Simulink 中内置的系统模块。

8.2 Simulink 模型库简介

为了方便用户快速构建所需的动态系统，Simulink 提供了大量的、以图形形式给出的内置系统模块，使用这些内置模块可以快速方便地设计出特定的动态系统。图 8.2 是 Simulink 的模型库浏览器。由该浏览器可以看出 Simulink 内置模型库包含公共模型库和专业模型库。

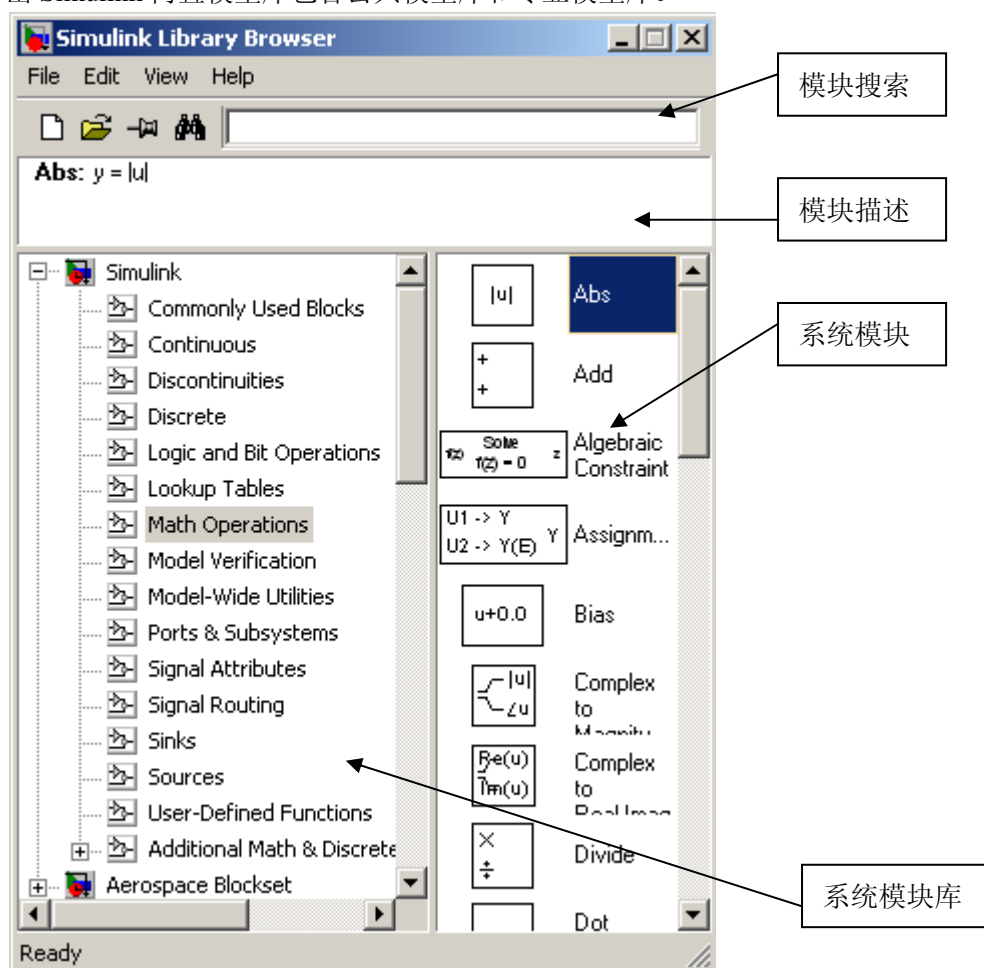


图 8.2 Simulink 模型库浏览器

8.2.1 Simulink 的公共模型库

Simulink 的公共模型库是 Simulink 中最通用的模型库，见图 8.3，它可以应用到不同专业，包括 16 个子模块库，下面对各子模型库及其功能做简要介绍。

一、Commonly Used Blocks(常用模型库)

新版本的 Simulink 中，为了方便用户使用，专门将常用的 22 种模块放在了常用模型库中。这些常用的模型包括输入模块、输出显示模块、连续（离散）系统积分模块、数学运算模块及信号路由模块等，其功能将在后面介绍的各种模型库中介绍。

二、Continuous（连续系统模型库）

连续系统模型库及其中各模块的功能见图 8.4。

三、Discontinuity（不连续环节）

不连续环节模型库及其中各模块的功能见图 8.5。

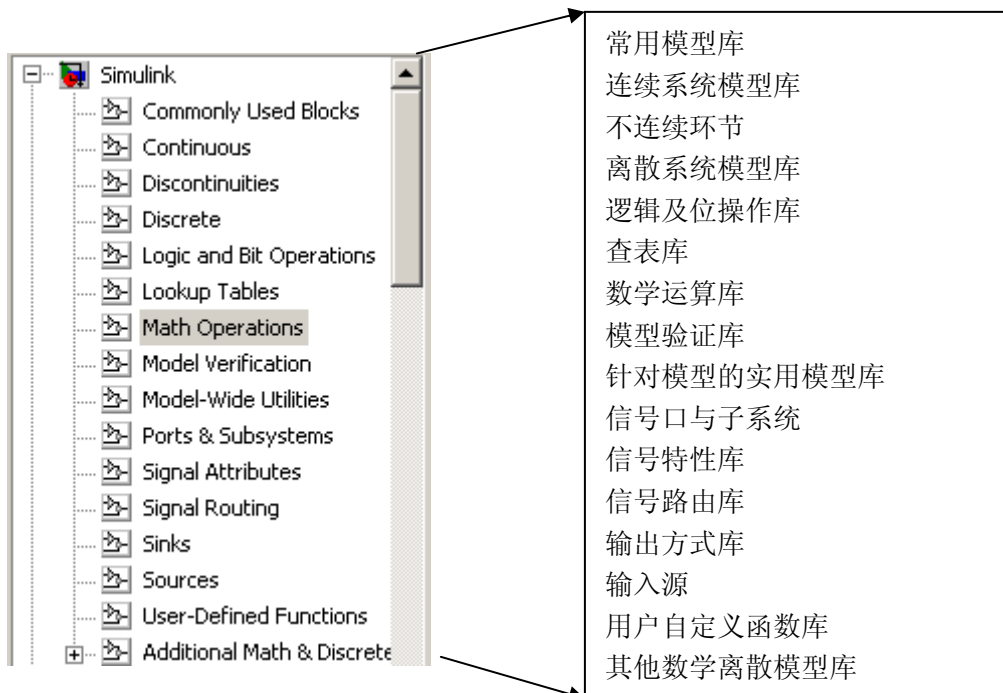


图 8.3 Simulink 的公共模型库

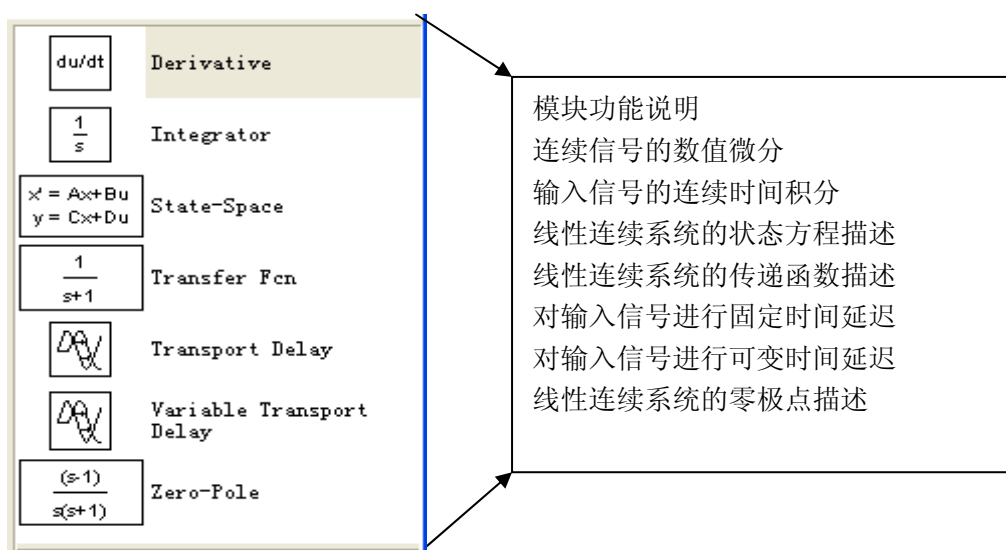


图 8.4 连续系统模型库及其功能

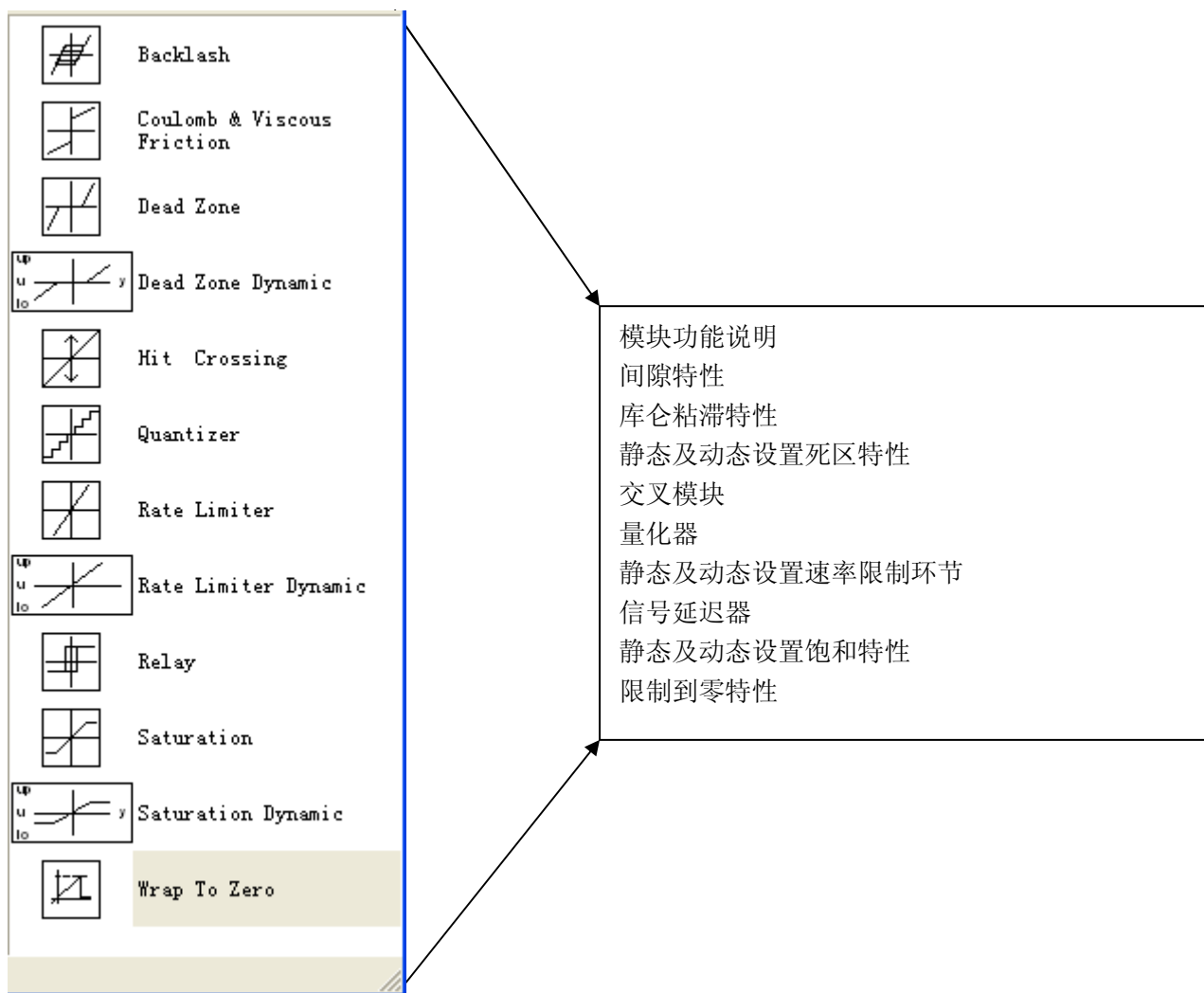


图 8.5 不连续环节模型库及其功能

四、Discrete（离散系统模型库）

离散系统模型库及其各模块的功能见图 8.6。

五、Logic and Bit Operations 逻辑及位操作库

逻辑及位操作库及其模块的功能见图 8.7。

六、Look_Up Tables(查表库)

查表库及其中各模块的功能见图 8.8。

七、Math Operation（数学运算库）

数学运算库及其中各模块的功能见图 8.9。

八、Model Verification（模型检验）

模型检验及其中各模块的功能见图 8.10。

九、Model-wide Utilities（针对模型的有用功能模块）

针对模型的有用功能模块及其中各模块的功能见图 8.11。

十、Ports and Subsystem（信号口与子系统）

信号口与子系统库及其中各模块的功能见图 8.12。

十一、Signal Attributes（信号特性库）

信号特性库及其中各模块的功能见图 8.13。

十二、Signal Routing（信号路由库）

信号路由库及其中各模块的功能见图 8.14。

十三、Sinks（输出方式库）

输出方式库及其中各模块的功能见图 8.15。

十四、Source（输入源）

输入源库及其中各模块的功能见图 8.16。

十五、User Defined Function（用户自定义函数库）

用户自定义函数库及其中各模块的功能见图 8.17。

十六、其他数学离散模型库

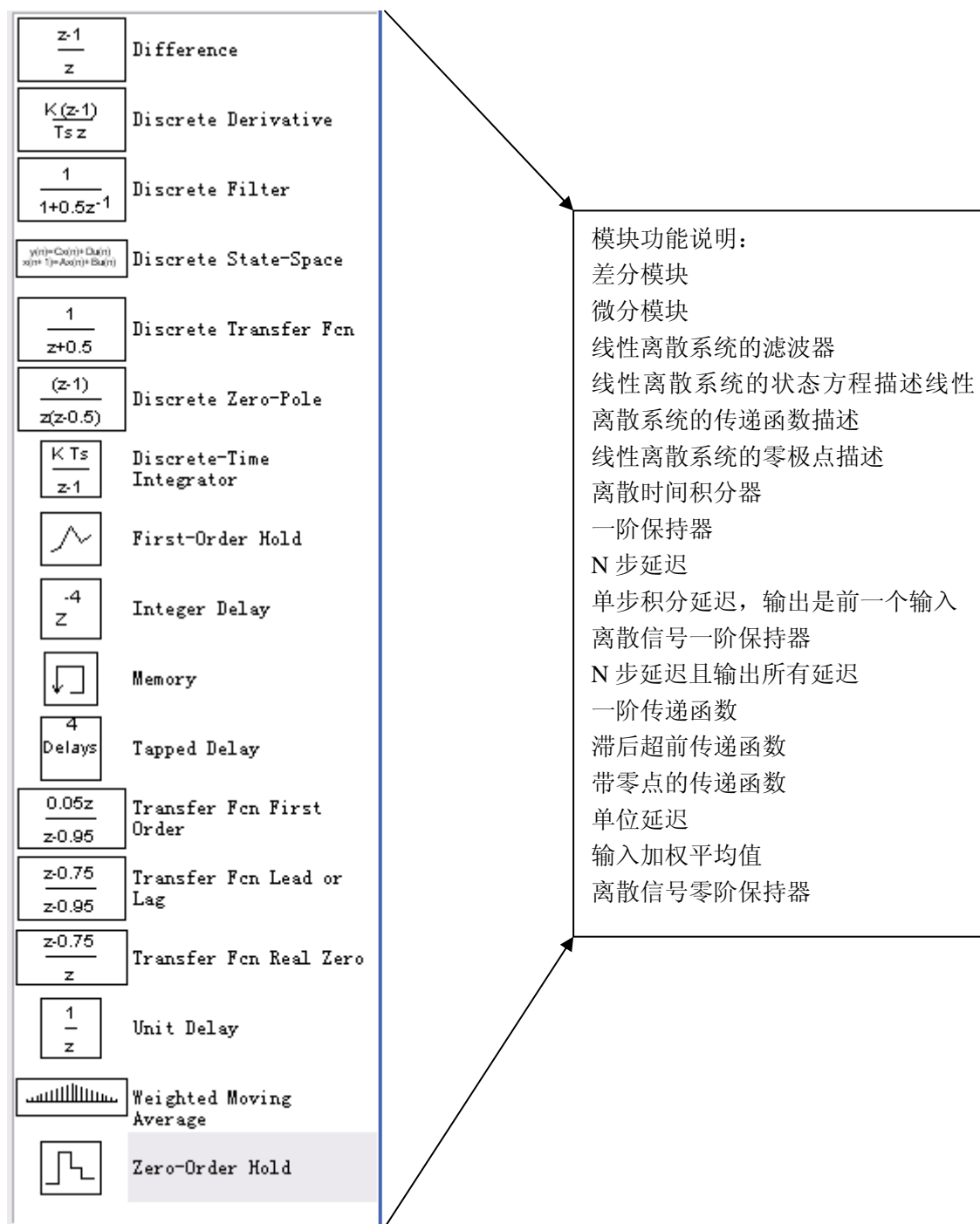


图 8.6 离散系统模型库及其中功能

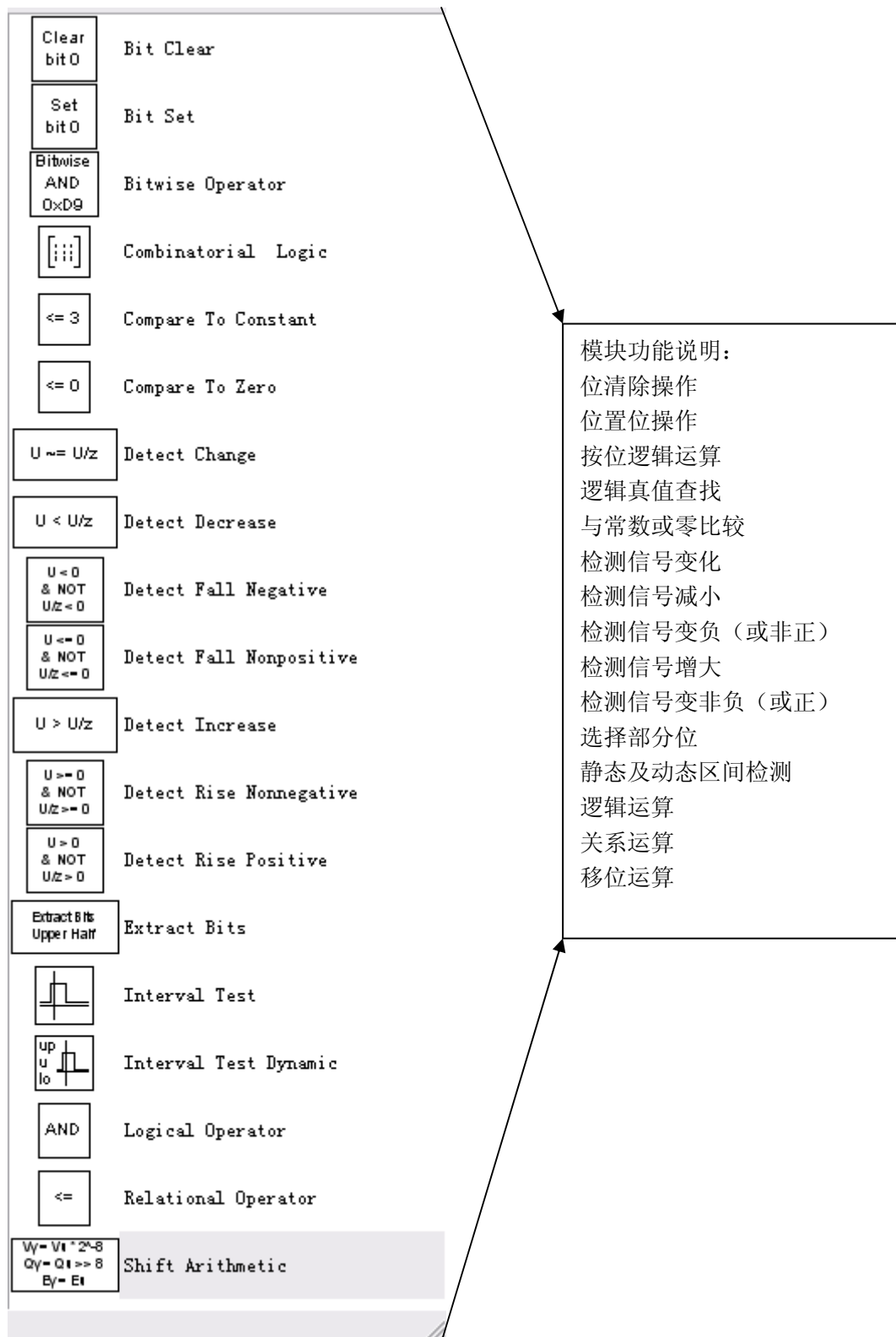


图 8.7 逻辑及位操作库及其功能

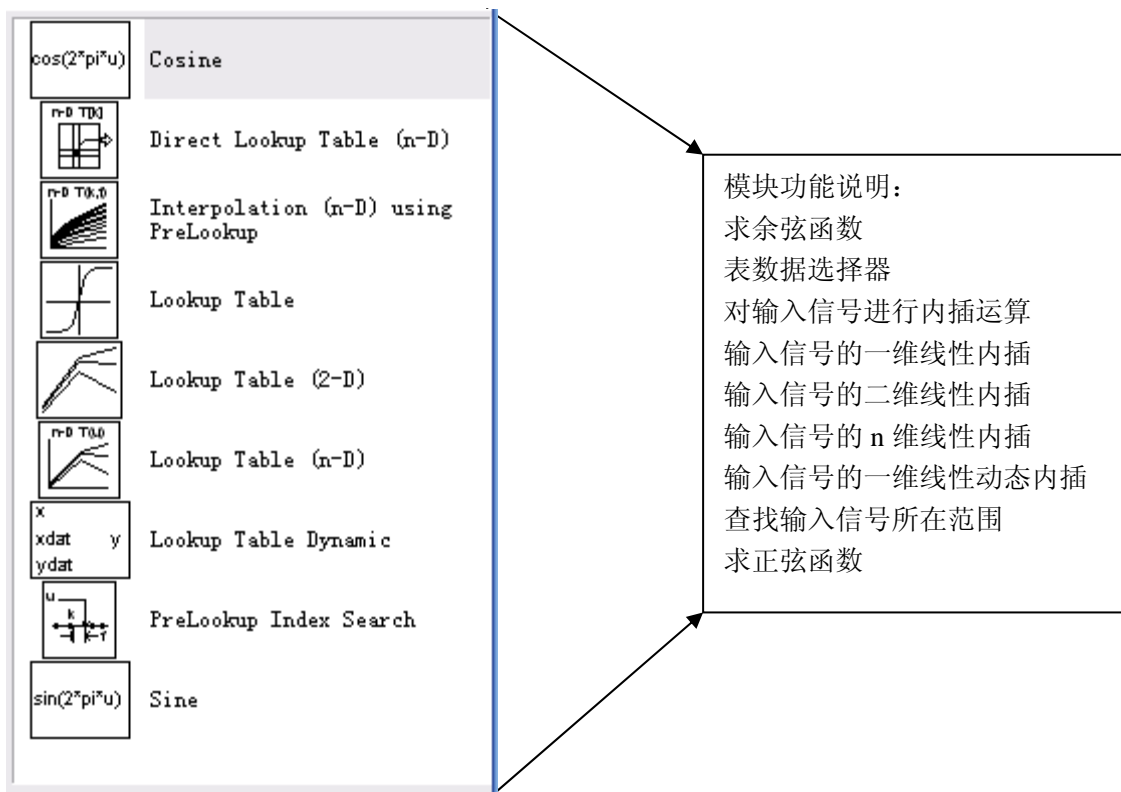
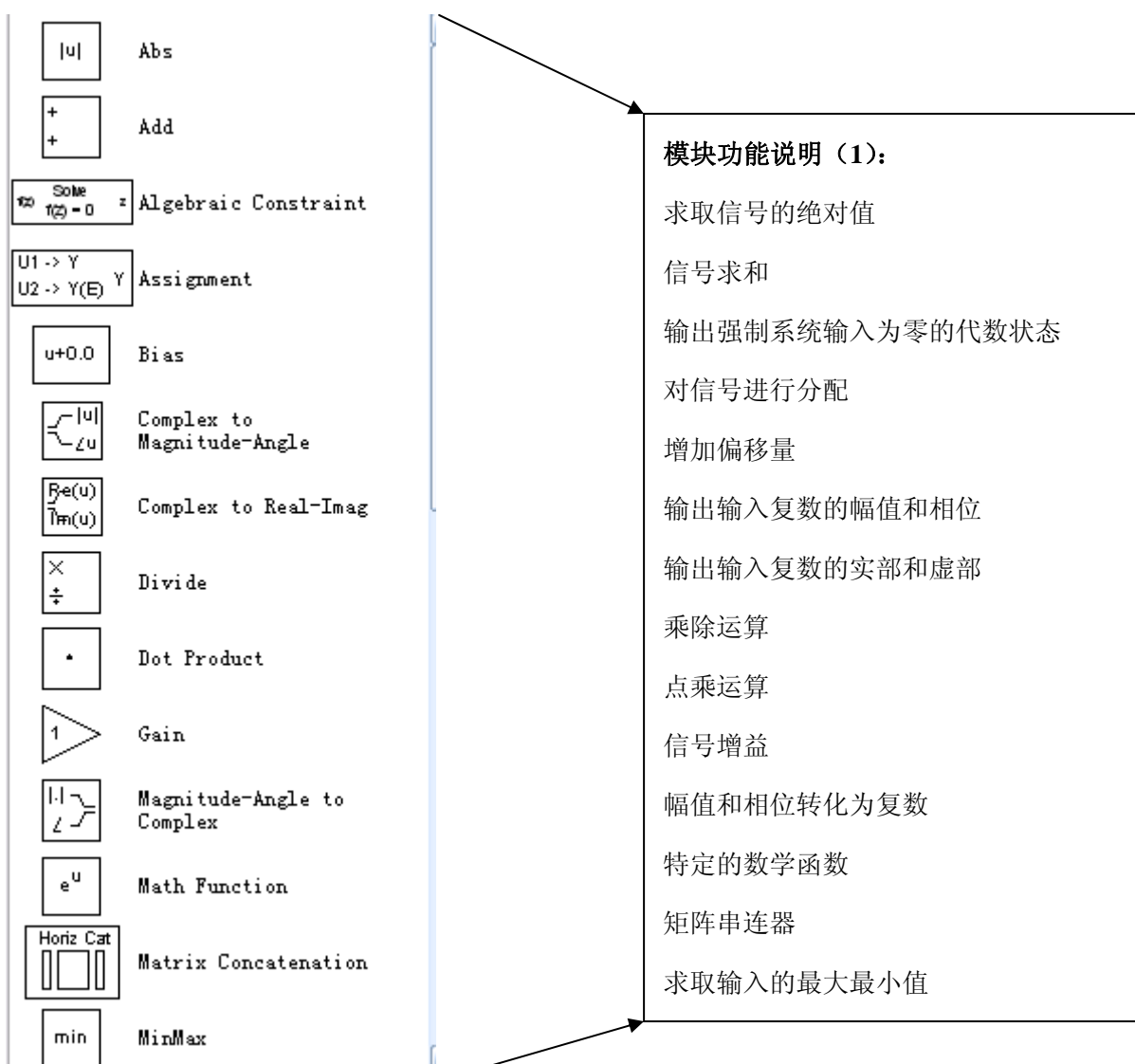


图 8.8 查表库及其功能



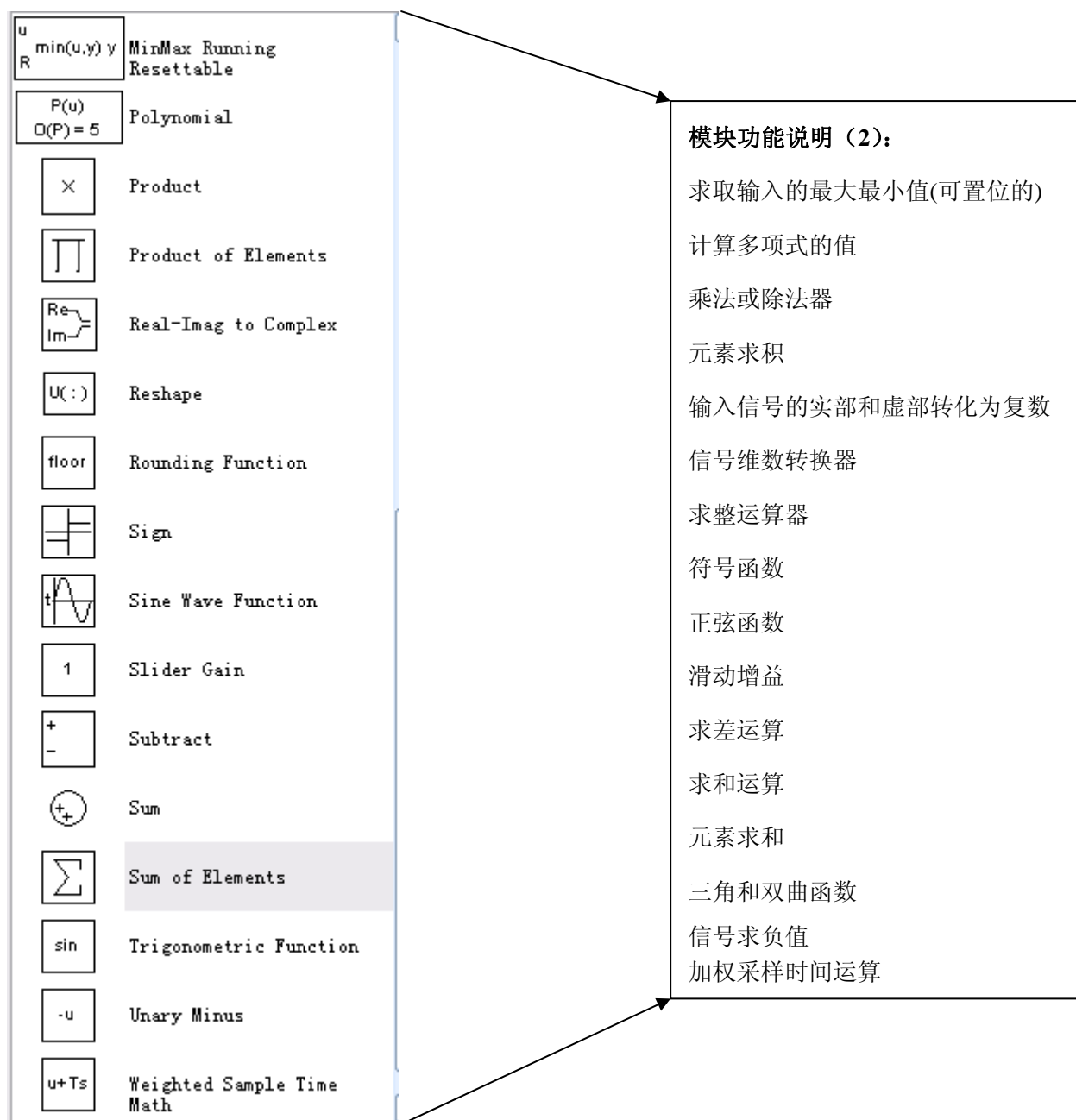


图 8.9 数学运算库及其功能

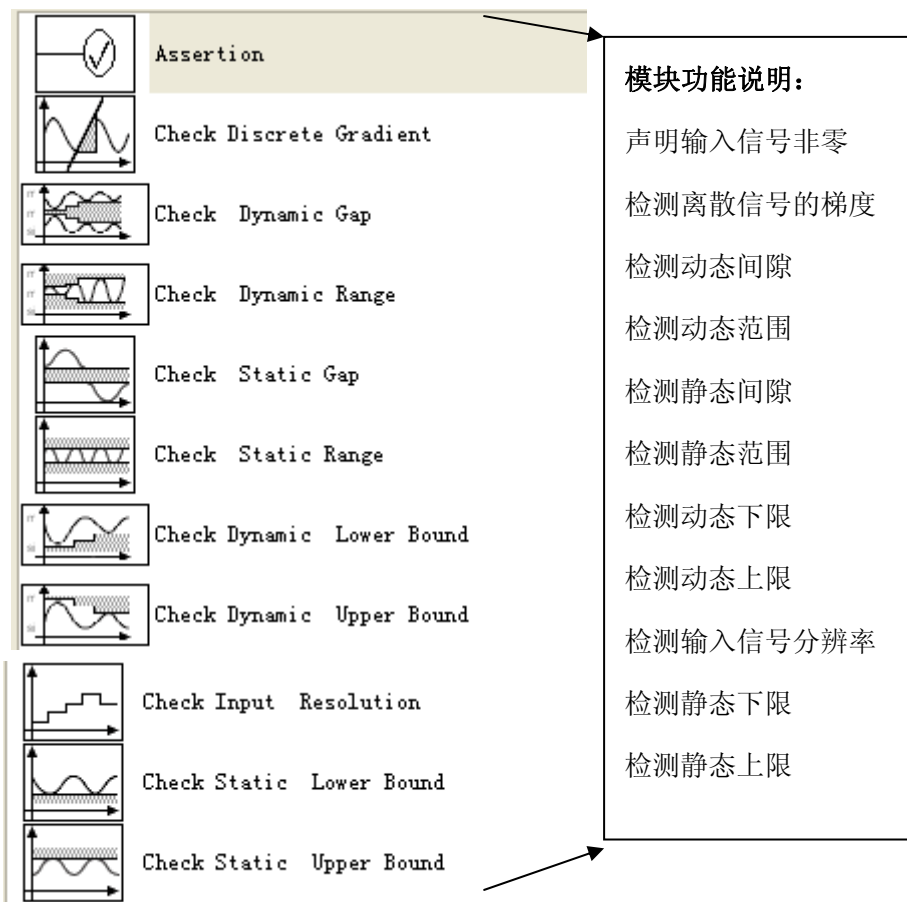


图 8.10 模型检验及其功能

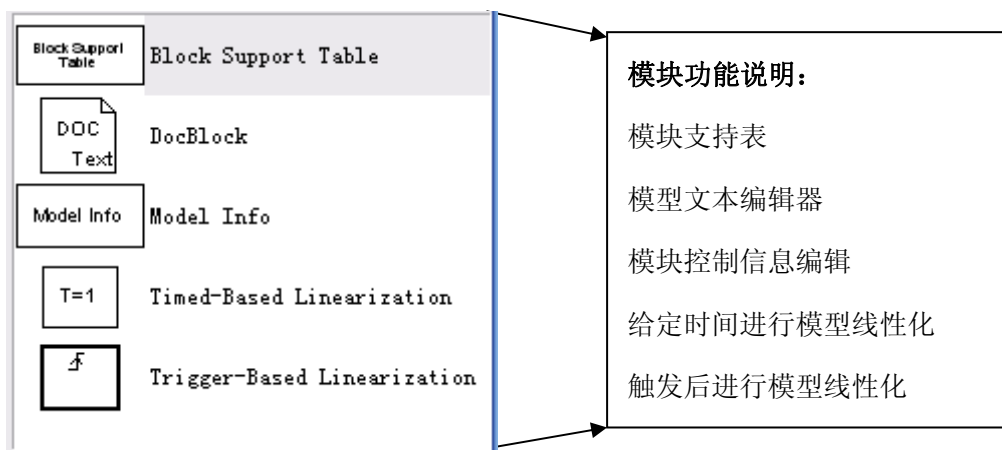


图 8.11 针对模型的有用模块及其功能

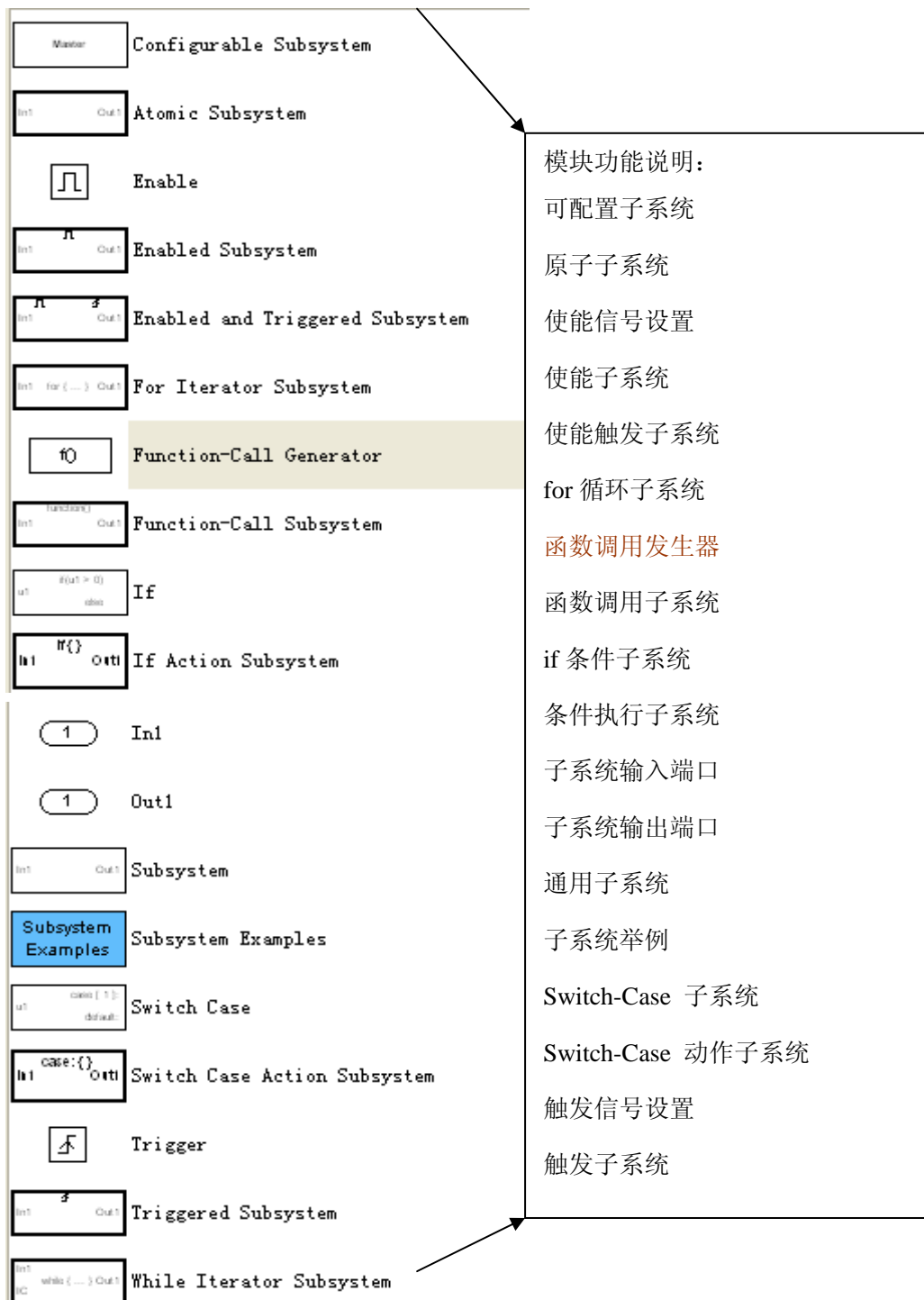


图 8.12 信号口与子系统库及其功能

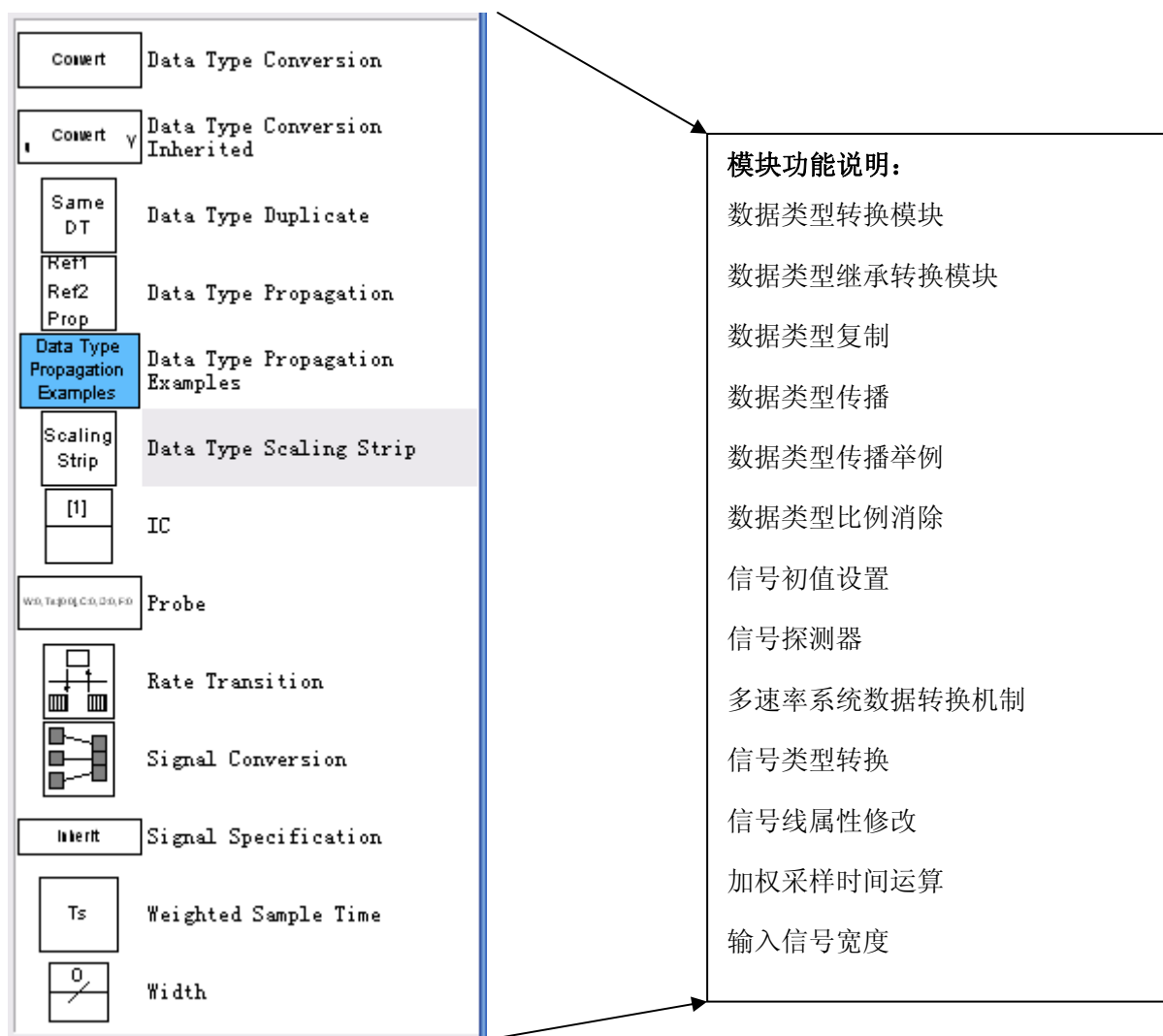


图 8.13 信号特性库及其功能

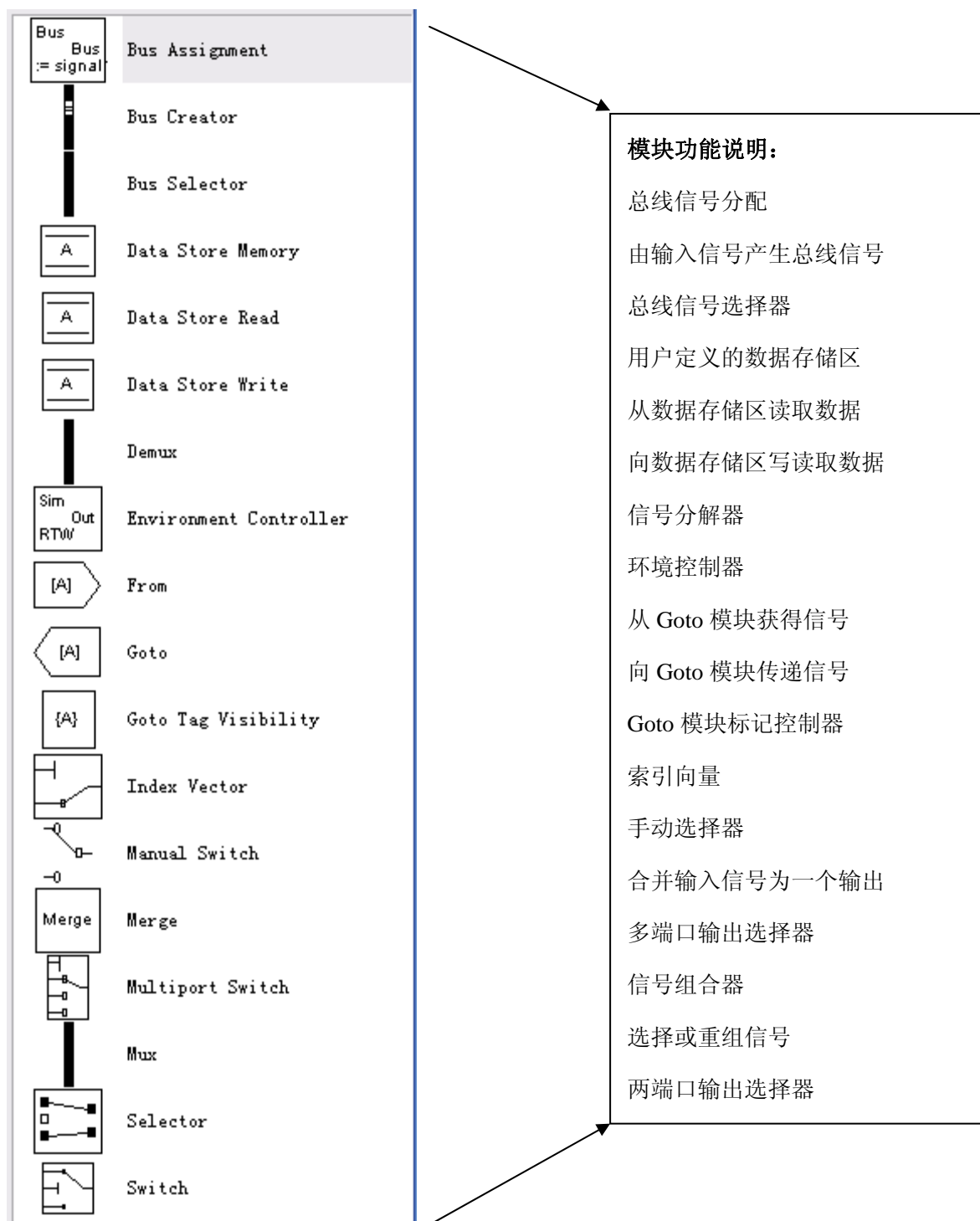


图 8.14 信号路由库及其功能

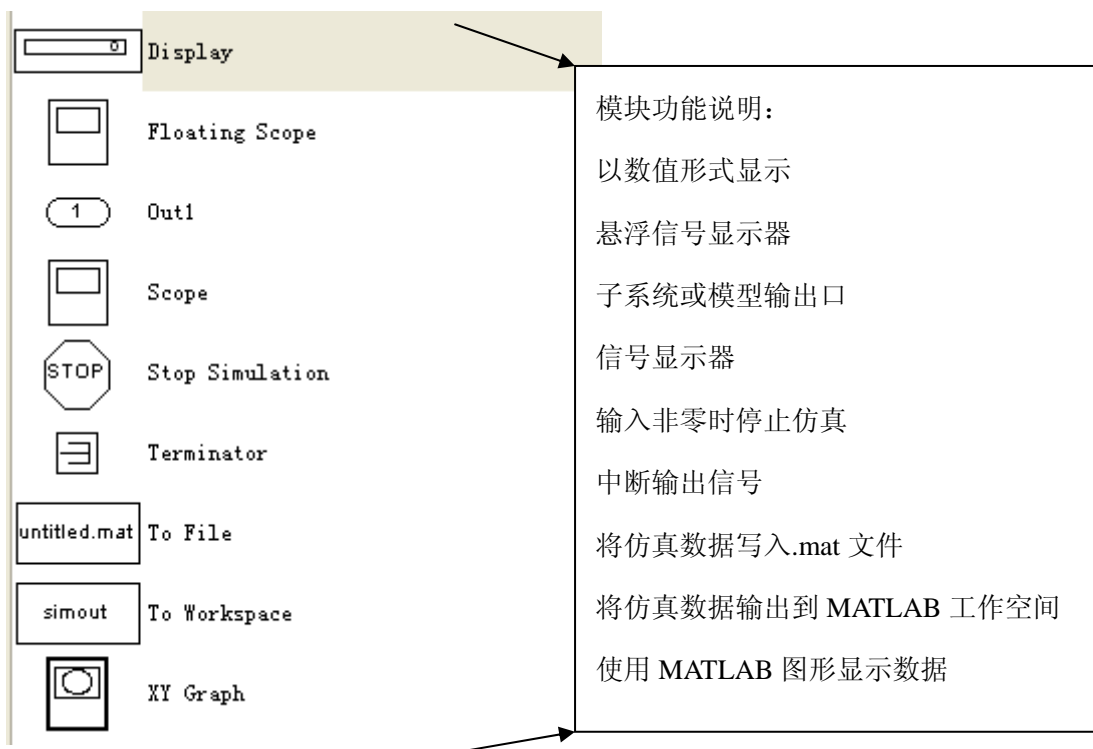


图 8.15 输出方式库及其功能

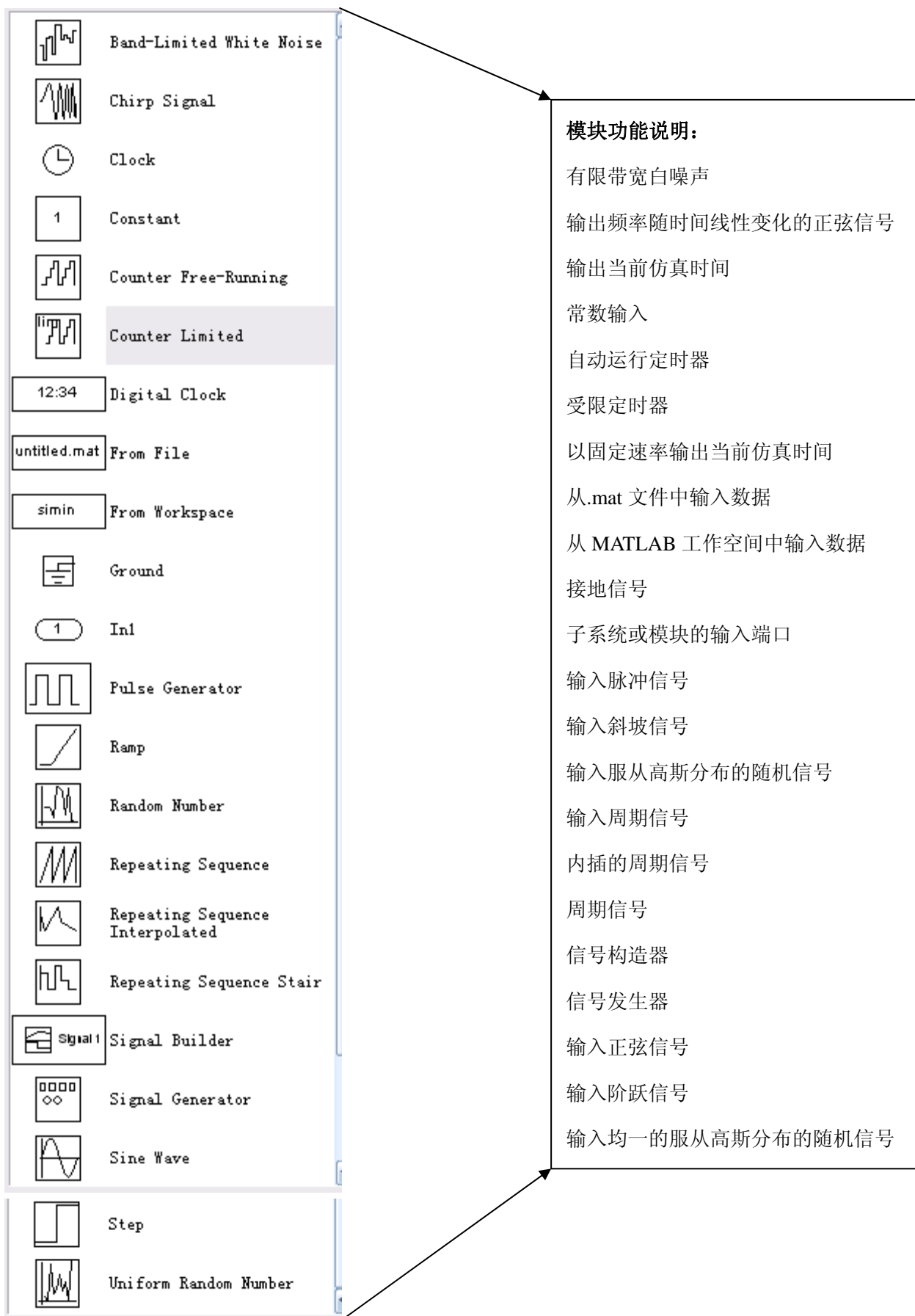


图 8.16 输入源库及其功能

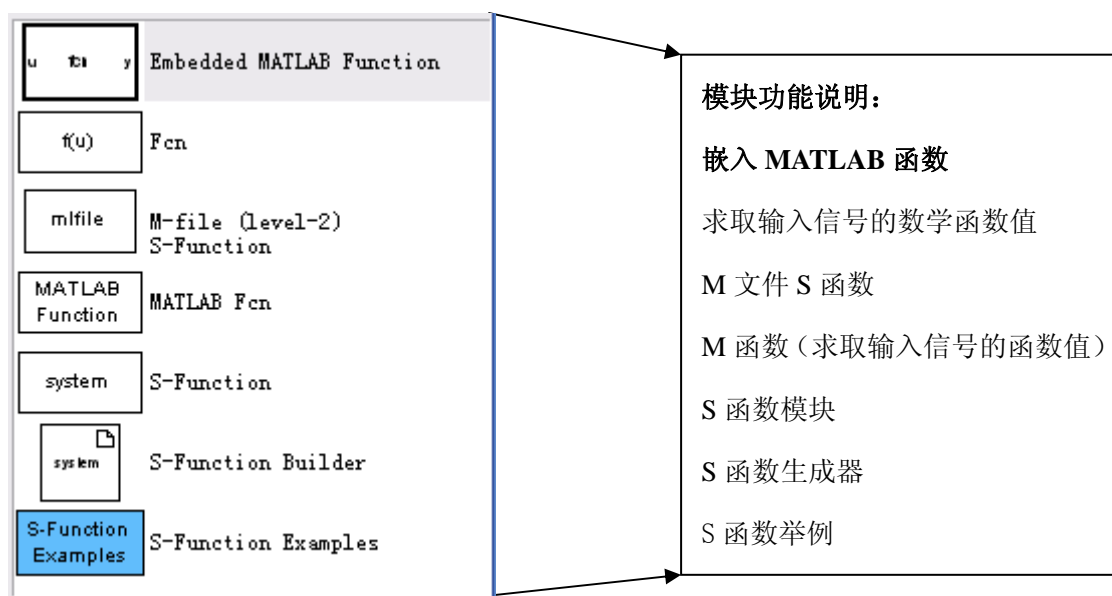


图 8.17 用户自定义函数库及其功能

8.2.2 Simulink 的专业模型库

前面对 Simulink 的公共模型库做了详细的介绍，除了公共模型库外，Simulink 中还集成了许多面向不同专业的专业模型库，不同领域的系统设计师可以使用这些系统模块快速构建自己的系统模型，然后在此基础上进行系统的仿真、分析，从而完成设计任务。下面仅介绍几种控制工程师可能用到的专业模型库的主要功能。

一、航空航天模型库 (Aerospace Blockset)

提供航空航天设计师常用的执行机构模块、空气动力模型、动画模块、环境仿真模块、三六自由度运动方程模块、风场、大气重力等环境模块、各种控制器模块、涡扇发动机模块、坐标和单位转换模块等。

二、控制系统模型库 (Control System Toolbox)

面向控制系统的设计和分析，主要提供线性时不变系统的模块。

三、数字信号处理模型库 (DSP Blockset)

面向数字信号处理系统的设计和分析，主要提供 DSP 输入/输出模块、信号预测与估计模块、滤波器模块、DSP 数学函数库、量化器模块、信号管理模块、信号操作模块、统计模块和信号变换模块等。

四、Simulink 附加模型库 (Simulink Extras)

补充 Simulink 公共模型库，提供附加离散系统模型库、附加连续系统模型库、附加输出模型库、触发器模型库、线性化模型库和转换模型库。

五、S 函数示例模型库 (S-function demos)

提供 C++、C、FORTRAN 以及 M 文件下的 S-函数的模型库的演示模块。

六、实时工作间、实时工作间嵌入式编码器及实时目标模型库 (Real-Time Workshop、Real-Time Workshop Embedded Coder 和 Real-Time Windows Target)

提供各种用来进行独立可执行代码或嵌入式代码生成，以实现高效实时仿真的模块。它们和 RTW、TLC 有着密切的关系。

七、状态流模型库 (Stateflow)

对使用状态图所表达的有限状态机模型进行建模仿真和代码生成，有限状态机用来描述基于事件的控制逻辑，也可以用于描述响应型系统。

八、通信模型库 (Communication Blockset)

专用于通信系统仿真的一组模块。

九、图形仪表模型库 (Gauges Blockset)

实际上是一组 ActiveX 控件。

十、神经网络模型库 (Neural Network Blockset)

用于神经网络的分析设计和实现的一组模块。

十一、模糊控制模型库 (Fuzzy Logic Toolbox)

用于模糊控制的分析设计和实现的一组模块。

十二、虚拟现实工具箱 (Virtual Reality Toolbox)

提供进行虚拟现实仿真分析的各种工具，包括输入、输出、信号扩展器等。

十三、xPC 模型库

提供一组用于 xPC 仿真的模块。xPC 是利用 PC 机，使用客户机服务器的模式进行实时仿真的一种经济仿真方案。它和 Simulink、RTW 相结合，可以在 PC 上进行单任务的实时仿真。

8.3 Simulink 模型的构建

8.3.1 对 Simulink 库浏览器的基本操作

8.2 节简单介绍了 Simulink 中的一些比较常用的系统模块。知道了这些模块的功能后，就可以使用这些系统模块构建用户自己的模型了。在构建用户模型的过程中，一定需要查找所需的模块，通过鼠标左键单击模型库的名称可以查看模型库中的模块，模型库中的模块会显示在 Simulink 库浏览器右边的一栏中。对 Simulink 库浏览器的基本操作主要包括：

使用鼠标左键单击模型库，则会在 Simulink 库浏览器右边的一栏中显示该库中的所有模块；

使用鼠标左键单击系统模块，则会在模块描述栏中显示该模块的功能描述；

使用鼠标右键单击系统模块，可以得到该模块的帮助信息，将模块拖进系统模型中，可以查看模块参数设置。

8.3.2 模块的基本操作

一、模块的选择

假设用户需要将正弦信号通过信号显示器显示。这时就需要两个模块：由系统输入模型库 Sources 中的 Sine Wave 模块产生正弦信号，用系统输出模型库 Sinks 中的 Scope 模块显示结果。

启动 Simulink 并新建一个系统模型文件。选择上面提到的两个模块将其拷贝（或拖）到新建的系统模型中，如图 8.18 所示。

二、模块的连接

选择好构建系统所需的模块后，需要按照系统的信号流程将各个模块正确连接。将光标指向起始块的输出端口，此时光标变成“+”。单击鼠标左键并拖动到目标模块的输入端口，在接近到一定程度时光标变成双“+”，此时松开鼠标键即可。完成后在连接点处出现一个箭头，表示系统中信号的流向，如图 8.19 所示。

三、模块的复制

如果需要几个相同的模块，可以使用三种方法进行复制，结果见图 8.20。

使用鼠标右键单击并拖动该模块；

选中所需模块后，使用 Edit 菜单上的 Copy 和 Paste；

使用热键 Ctrl+C 和 Ctrl+V。

四、模块的移动

选中需要移动的模块，按下鼠标左键将模块拖至合适的位置即可。

需要说明的是，模块移动时，与之相连的连线也随之移动；在不同模型窗之间移动模块时，需要同时按下 Shift 键。

五、模块的删除

选中需要删除的模块后，按下 Delete 键或采用剪切的方法可完成对模块的删除。

六、模块的旋转

在构建仿真模块时，有时系统的信号并非都是从左到右的，因而模块不能总处于其缺省的输入端在左，输出端在右的状态，在选中需要旋转的模块后，有两种方法可以旋转模块：

使用 **Ctrl+R** 热键；

使用 **Format** 菜单下的 **Flip Block** 可将模块旋转 180°，**Format** 菜单下的 **Rotate Block** 可将模块旋转 90°。

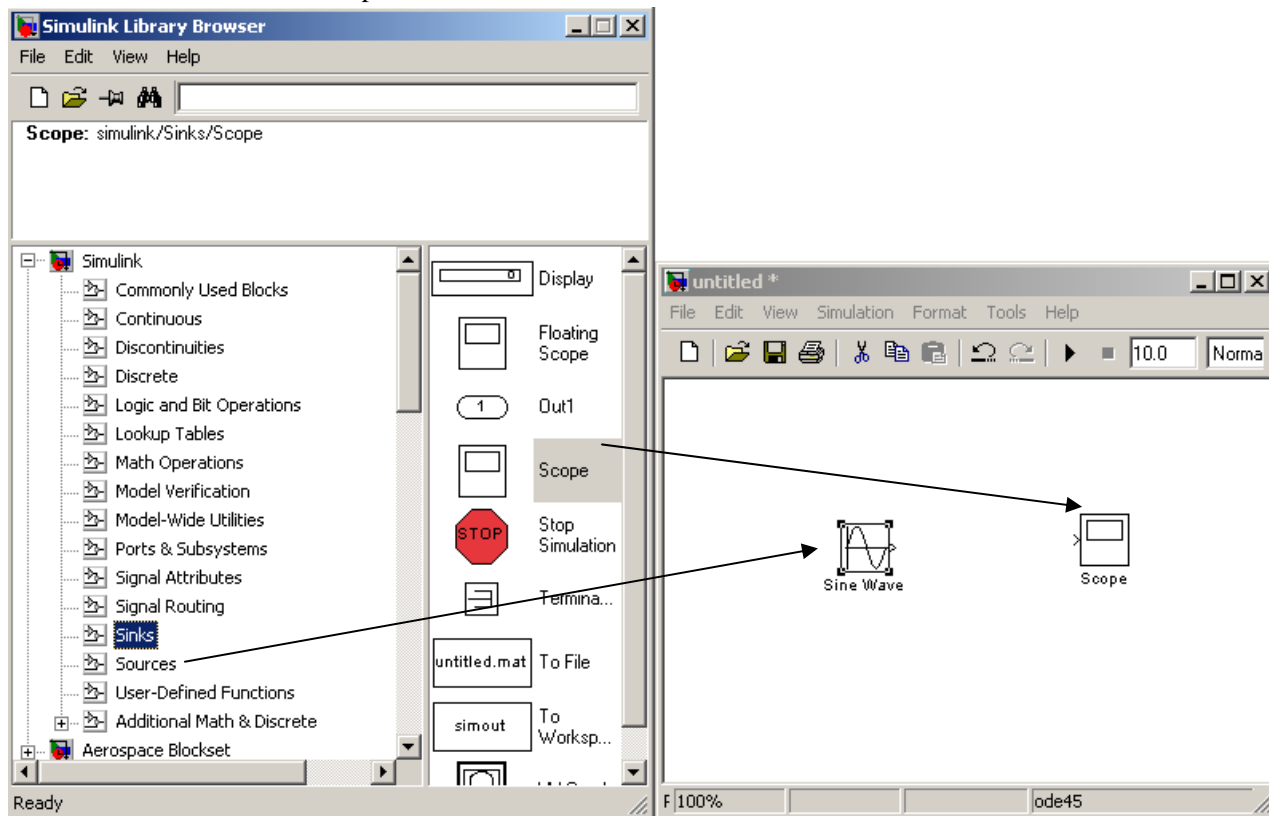


图 8.18 模块选择示例

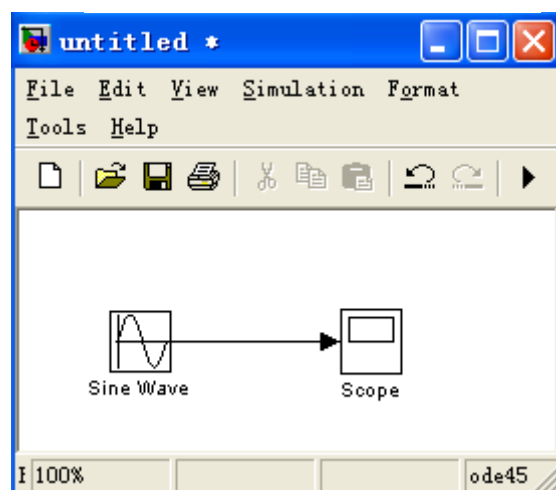


图 8.19 系统模块之间的连线

七、模块名的操作

- 1、修改模块名：点击模块名，会在原名字四周出现一个编辑框。此时可对模块名进行修改。修改完毕，将光标移出该编辑框，点击即结束修改。
- 2、模块名字体设置：使用菜单 **Format** 下的 **font**，打开字体设置对话框进行设置。

- 3、改变模块名的位置：选中模块后，使用菜单 **Format** 下的 **Flip Name**，可将模块名移至对侧或利用鼠标拖动模块名编辑框移至对侧。
- 4、隐藏模块名：选中模块后，选用 **Format** 菜单下的 **Hide Name** 即可，同时，菜单也变为 **Show Name**。

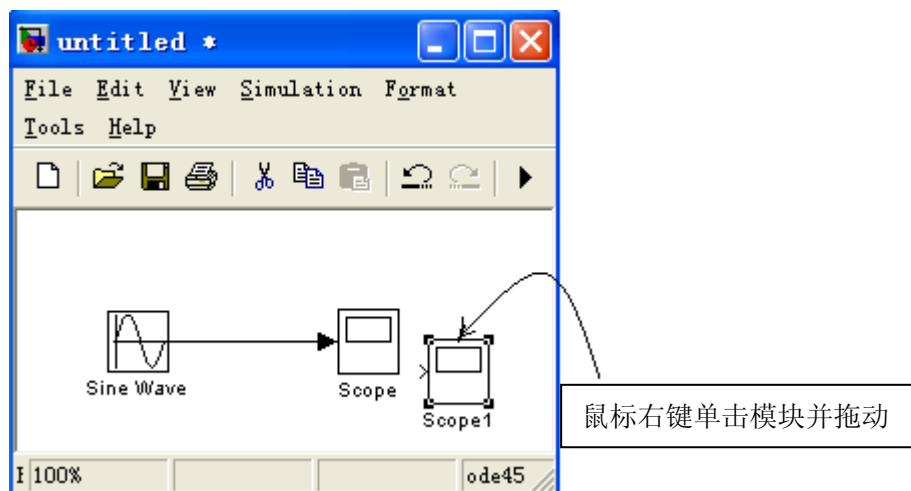


图 8.20 模块的复制

八、模块的阴影效果

使用 **Format** 菜单下的 **Show Drop Shadow** 可以给选定的模块加上阴影效果。

九、模块颜色的改变

使用鼠标右键点击模块，选择 **Foreground color** 或 **Background color** 菜单来设置颜色；或使用 **Format** 菜单下的相应命令设置颜色。如果模块的前景色发生变化，则所有由此模块引出的信号线的颜色也随之发生改变。

十、模块的插入

假设用户需要将输入信号放大后才显示，此时需要在图 8.19 所示的两个模块之间插入一个放大环节。用户只需将所需模块移到连接线上即可完成插入工作，见图 8.21。

8.3.3 信号的操作

Simulink 模型中的信号总是由模块之间的连线携带并传送的，因此模块间的连线称为信号线。

一、信号线的分支

有些系统模块的输出必须同时作为多个模块的输入，这时需要从此模块中引出若干连线，称作信号线分支。对信号线进行分支的操作有下列方式：

使用鼠标右键单击需要分支的信号连线拖动目标模块；

按下 **Ctrl** 键的同时，使用鼠标左键单击需要分支的信号连线拖动目标模块；

二、信号线的曲折

对信号连线路径的改变的方式有：

使用鼠标左键单击并拖动；

按下 **Shift** 键的同时，使用鼠标左键单击并拖动，可以改变连线路径。

三、信号线宽度的设置

信号线所携带的信息既可能是标量也可能是向量，并且不同信号线所携带的向量信号的长度可能互不相同。为了使信号传递一目了然，**Simulink** 不仅可以用粗宽线显示向量信号线，还可将向量的长度用数字标出。使用 **Format** 菜单中的 **Port/Signal Display—Wide nonscale Lines** 和 **Signal Dimensions** 完成这项功能。

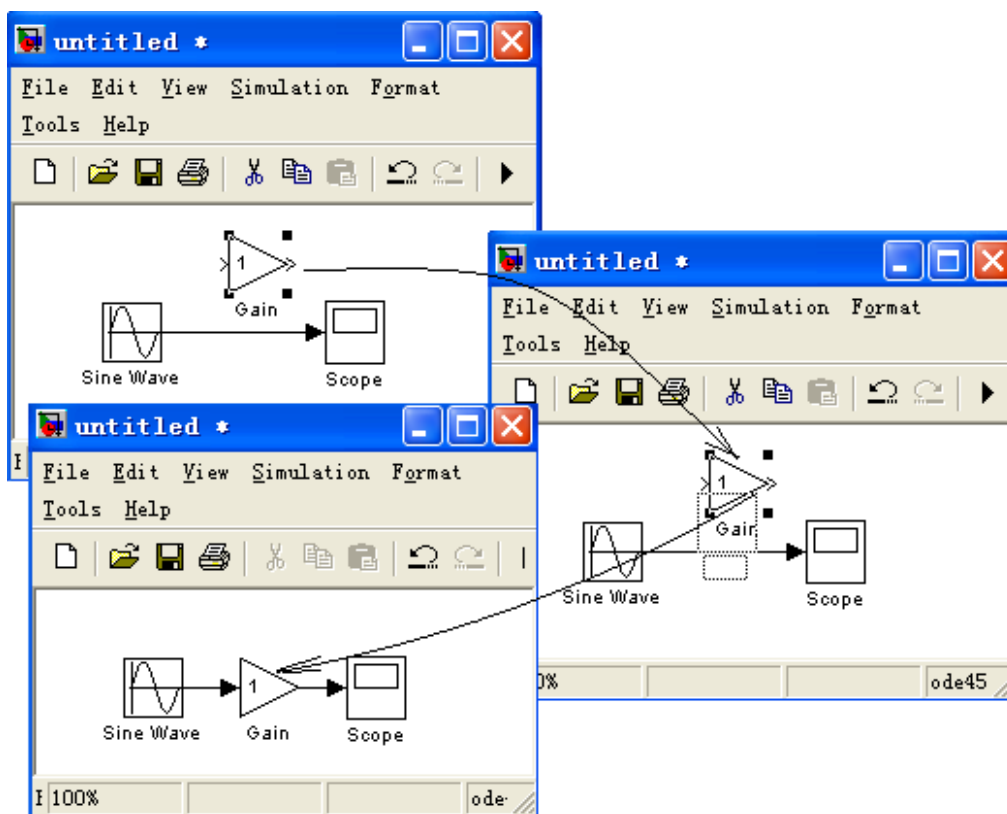


图 8.21 模块的插入

四、信号线的彩色显示

Simulink 所建的离散系统允许有多个采样频率。为了清楚显示不同采样频率的模块及信号线，选中 Format 菜单下的 Port/Signal Display—Sample Time Colors 可以用不同的颜色显示采样频率不同的模块和信号线。默认红色表示最高采样频率，黑色表示连续信号流经的模块和信号线。

五、信号的组合与分解

在利用 Simulink 进行系统仿真时，有时需要将某些模块的输出信号组合成向量信号，并将得到的向量信号作为另外模块所输入；有时又需要将向量信号分解成多个标量信号。能够完成信号组合与分解功能的模块是 Signal Routes 模型库中的 Mux 模块和 Demux 模块，使用 Mux 模块可以将多个标量信号组合成一个向量信号，使用 Demux 模块可以将向量信号分解成多个标量信号。如图 8.22 说明信号的组合与分解。

图 8.22 中使用示波器显示模块 Scope 显示信号，Scope 模块只有一个输入口，若输入信号是向量信号，则 Scope 模块会以不同的颜色显示每个信号。

六、信号标识

对信号进行说明，就必须在信号线上进行信号标识操作。信号标识操作包括：

- 1、添加标识：双击需要添加标识的信号线，弹出一个空白的文本编辑框，在其中输入文本即可。
- 2、修改标识：点击需要修改的标识，原标识四周出现一个编辑框，此时即可修改。
- 3、移动标识：点击标识出现编辑框后，光标指向编辑框，按下鼠标并拖至新的位置即可。
- 4、复制标识：与移动标识类似，只需同时按下 Ctrl 键或利用鼠标右键操作。
- 5、删除标识：点击标识出现编辑框后，双击标识选中整个标识，按下 Delete 键即可删除标识。
- 6、标识的传递：在 Simulink 模型库中，有一些象 Demux, Mux, Goto, From 等模块具有传递信号标识的功能。这种功能可使整个模型中的信号传递路径清晰，便于分析检查。信号标识的传递方法有两种。

第一种方法，选择信号线并用鼠标左键双击，在信号标识编辑框中键入<>,在此尖括号中键入信号标签即可传递信号标识。然后选择 Edit 菜单下的 Update Diagram 刷新模型。第二种方法，先选择信号线，然后

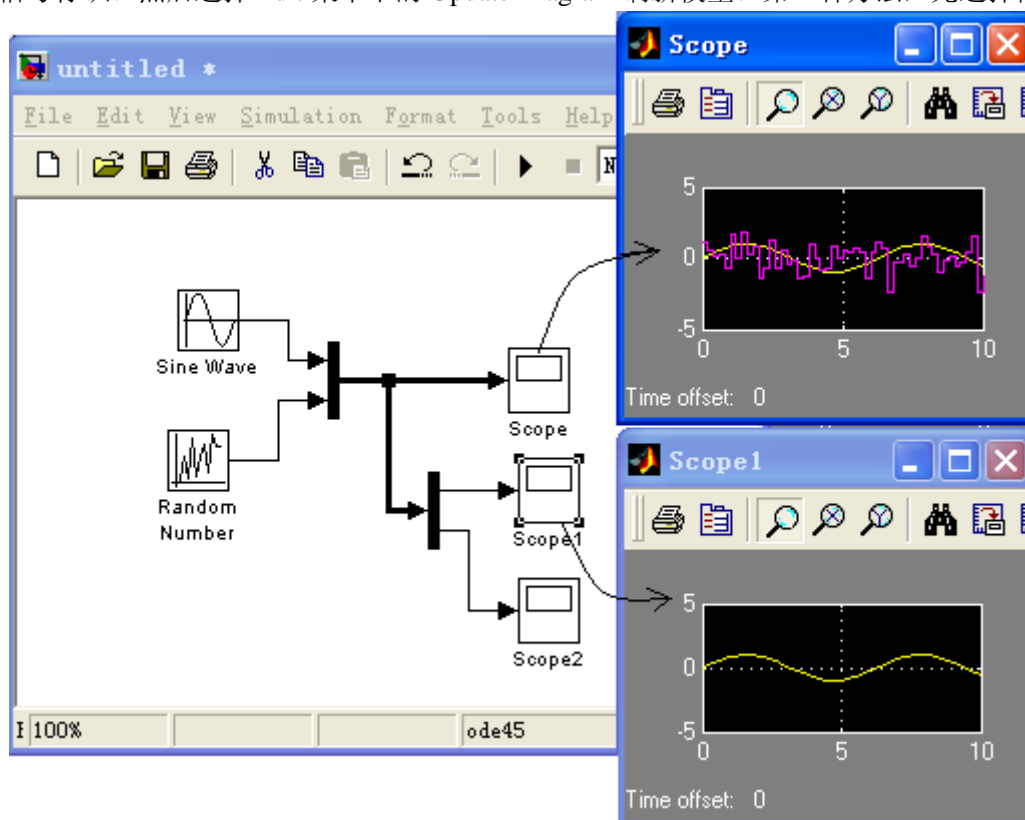


图 8.22 信号的组合与分解

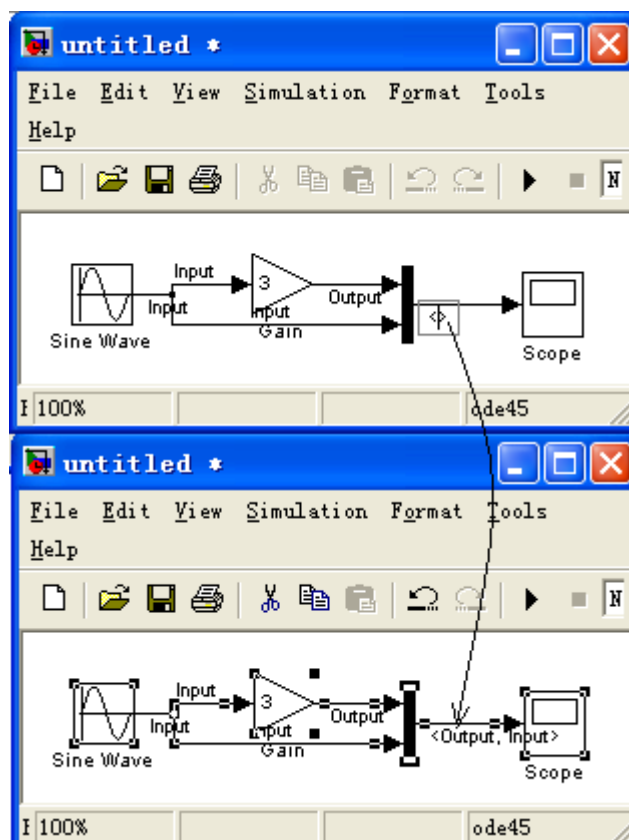


图 8.23 信号标识的传递

选择 Edit 菜单中的 Signal Properties; 或单击鼠标右键, 选择弹出式菜单中的 Signal Properties, 将 Show Propagated Signals 设置为 on 即可。图 8.23 说明信号标识的传递。

8.3.4 对模型的注释

在模型窗中书写注释可以帮助用户更方便、更好地理解模型。

模型注释的创建: 在将用作注释区的中心位置双击鼠标左键, 出现编辑框, 在框中输入所需的文字即可。

注释位置的移动: 在注释文字处单击鼠标左键出现编辑框, 将光标移至编辑框上按下鼠标左键并拖至希望位置即可。

注释文字的字体控制: 点击注释编辑框, 在选中菜单 Format 下的 Font, 弹出标准的字体对话框进行设置。

8.3.5 运行和仿真

前面讲述了如何利用 Simulink 模型库进行系统建模, 建模的目的是为了分析和设计系统的, 为了有效地分析系统, 就必须对所建的模型进行仿真计算。现在讨论运行仿真的问题。在系统仿真前, 用户必须对系统的模型参数和系统仿真的参数进行设置。

一、系统模块参数设置与系统仿真参数设置

当用户按照信号的输入输出关系连接好各个系统模块之后即完成了模块的构建工作。为了正确仿真和分析必须正确设置系统模块参数和仿真参数。

1、系统模块参数的设置

双击系统模块打开包含该模块的简单描述和模块参数选项等信息的系统模块对话框, 在该参数对话框中正确设置参数即可。模块参数设置过程见图 8.24。

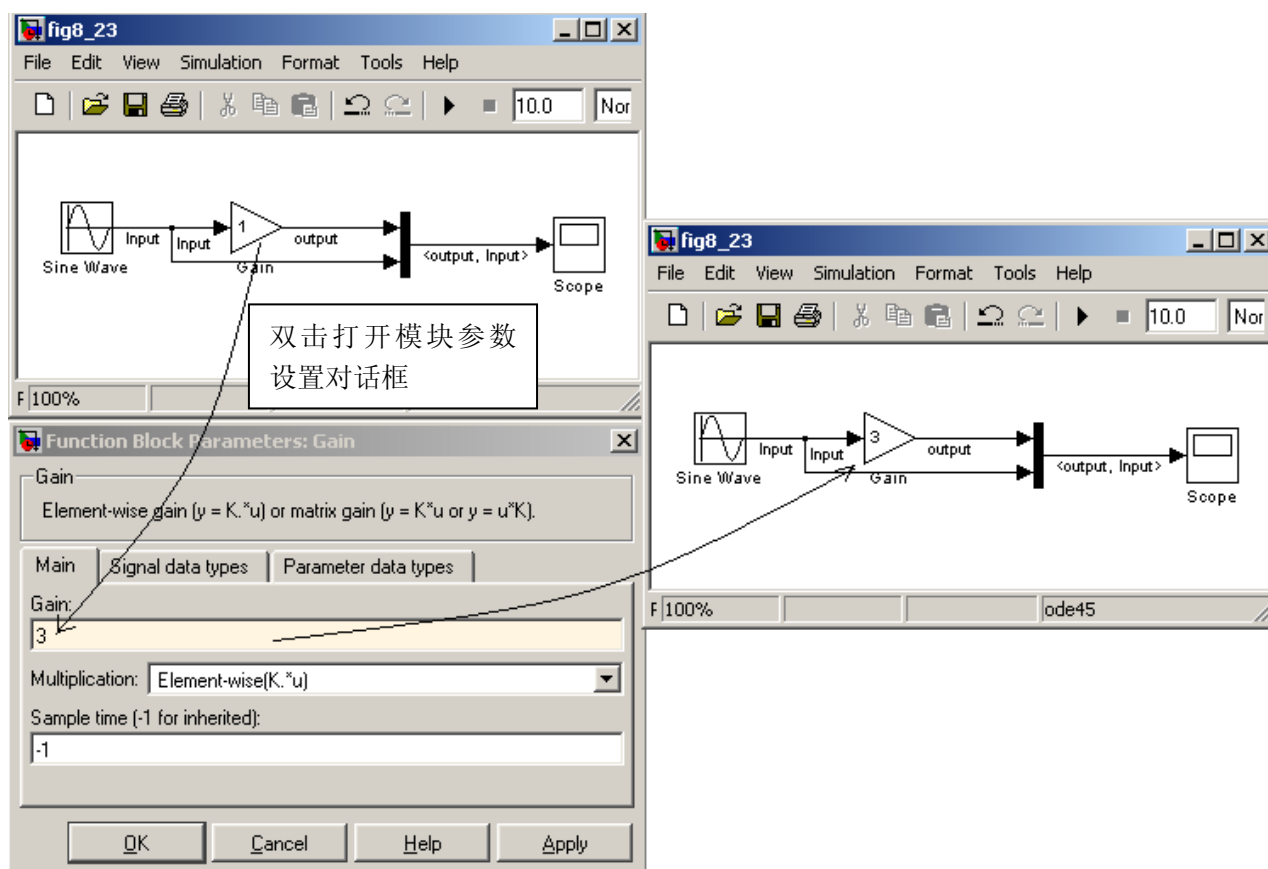


图 8.24 模块参数设置过程

2、系统仿真参数的设置

系统模块参数合理设置完毕之后就可设置系统仿真参数了，Simulink 的窗口中的 simulation / Configuration Parameters 选项就是用来设置仿真参数的。双击该选项后，弹出仿真参数设置对话框，见图 8.25,其中包括了 7 个选择页，它们是：solver(求解器),Data Import/Export(数据输入输出页)，Optimization（优化页），Diagnostics(诊断页)，Hardware Implementation（硬件实现页），Model Referencing（参考模型页），Real_time workshop(实时工作间页)。这里我们主要讲述 Solver 页，因为求解器的正确设置是得到合理、可信的仿真结果的前提条件，因而有必要详细了解求解器的设置内容和设置方法，以便进行正确地设置。关于 Data Import/Export(数据输入输出页)，Diagnostics(诊断页),Real_time workshop(实时工作间页)的设置与功能，我们在后面的章节中再做介绍。

通过 Simulink 的窗口中的 Simulation / Configuration Parameter 打开的对话框停留在 Solver 求解器设置页，见图 8.25。Solver 页包含了 simulation time（设置仿真时间）和 solver options（选择求解法）。这两个选项又各自包含若干小选项。

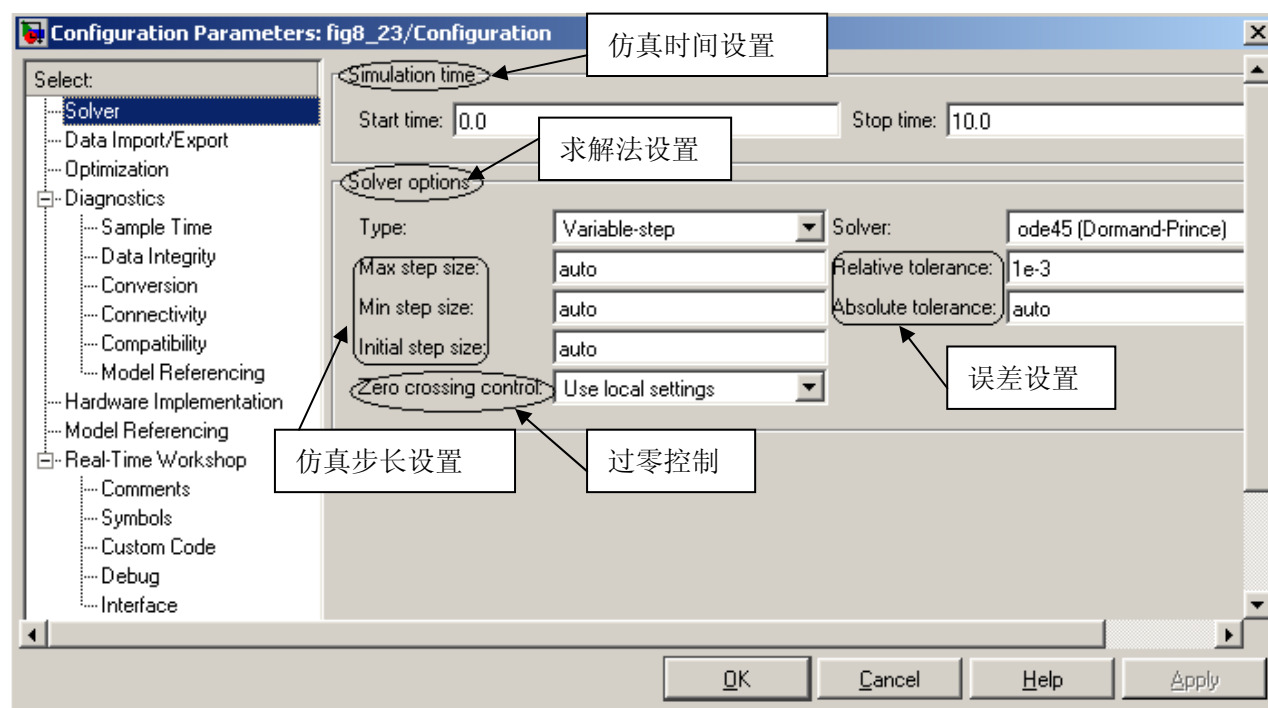


图 8.25 仿真参数设置对话框

Simulation time / start time 仿真起始时间(秒)
Stop time 仿真终止时间 (秒)

Solver options

Type /variable step 变步长

Fixed step 固定步长

Ode45,ode23 常微分方程的 runge-kutta 解法等选择

Max step size 最大步长

Min step size 最小步长

Initial step size 初始步长

Relative tolerance 相对误差

Absolute tolerance 绝对误差

Zero crossing control 仿真时过零控制

对应图 8.24 所示系统，系统仿真参数采用 Simulink 中默认的设置。

二、运行仿真

完成系统模块的构建、模块参数设置和仿真参数设置后，即可进行系统仿真了。单击模型文件工具栏上的 Start Simulation 图标 或通过 Simulation 菜单下的 Start 来启动仿真。仿真结束后，双击 Scope 模块可以显示仿真结果，见图 8.26。

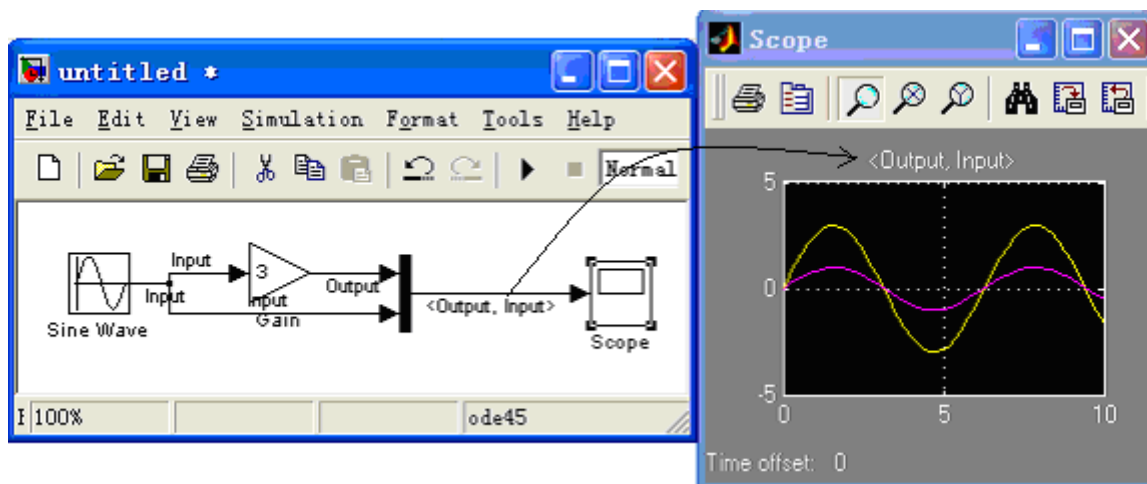


图 8.26 仿真结果显示

8.4 基于 Simulink 系统仿真技术应用举例

用 Simulink 建模并仿真应按照四个步骤进行，包括建模、设置模型参数、设置仿真参数和进行仿真分析四步。本节通过对连续系统、离散系统和混合系统的举例，使读者更牢固地掌握系统建模仿真的步骤、模型参数、仿真参数的设置方法以及仿真结果的分析 and 显示。

8.4.1 连续系统仿真分析

连续系统中既包含连续的线性系统也包含连续的非线性系统。首先举一个连续线性系统的例子，说明如何使用 Simulink 对连续线性系统进行仿真分析。

例 8.1 单位反馈的二阶系统见图 8.27。因为该系统的阻尼比较小，实际使用时需进行性能改善。采用比例加微分控制，可以在系统出现位置误差之前，提前对系统产生修正作用，最终达到改善系统性能的目的。加入比例加微分控制后的系统模型图及其仿真结果见图 8.28。

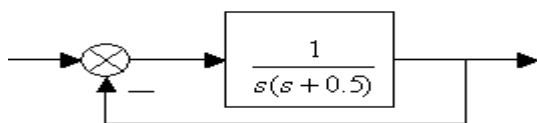


图 8.27 单位反馈二阶系统

解：

一、模型创建：

建立本系统模型需用的模块有：

- Source 模型库中的 Step 模块：输入阶跃信号；
- Math Operations 中的 Gain 和 Sum 模块；
- Continuous 中的 Derivatives 和 Zero-Pole 模块；
- Sinks 中的 Scope 模块。

二、模型参数及系统仿真参数设置

1、系统模块参数设置为：

Zero-Pole 模块设置：将零点 Zeros、极点 Poles 和增益分别设置为[]、 [0 -0.5]和 1；

Step 模块设置：使用系统的默认取值，即 1s 时阶跃的单位阶跃信号

其它模块设置可参见图 8.28。

2、系统仿真参数设置：

使用 Simulation parameters 仿真参数对话框中的 Solver 选项卡设置参数，

仿真时间范围设置为 0~20s；


使用变步长连续求解器（variable-step），仿真算法为 ode45；

最大仿真步长（Max step size）为 0.01；

绝对误差（Absolute tolerance）为 1e-6；

其余仿真参数使用默认值。

三、仿真

对模块参数和仿真参数进行了合理的设置之后，可以进行系统仿真，点击模型文件工具栏上的 Start Simulation 图标  或通过 Simulation 菜单下的 Start 来启动仿真。仿真结束后，双击 Scope 模块可以显示仿真结果，见图 8.28。

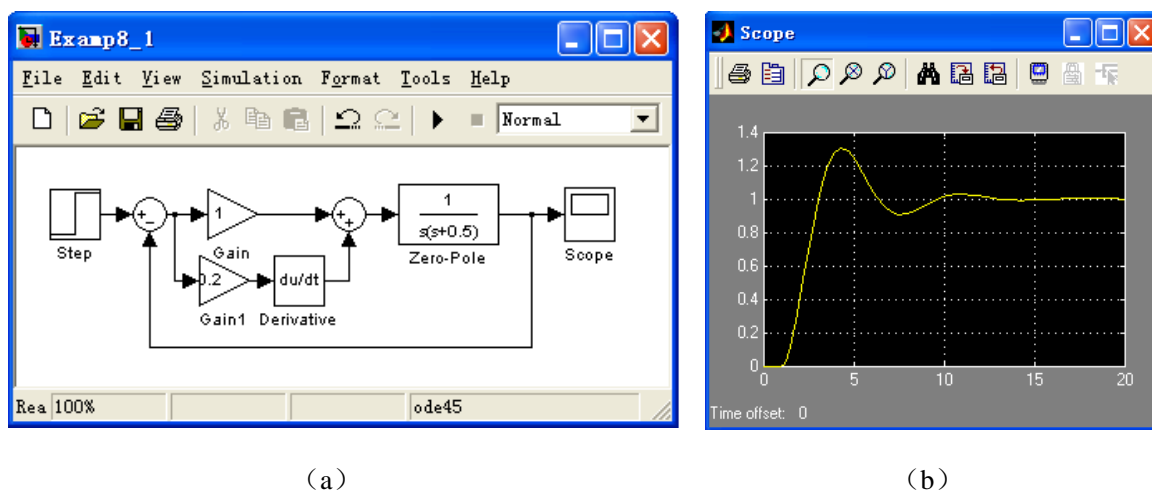


图 8.28 比例加微分控制系统模型及其仿真结果

例 8. 2 连续的非线性系统举例。利用 Simulink 计算 Van der pol 方程：
$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -m(x_1^2 - 1)x_2 - x_1 \end{cases}, \text{ 并}$$

用示波器 Scope 显示状态量 x_1 和 x_2 。

解：

一、系统模型创建

建立本系统模型需用的模块有：

Math Operations 中的 Gain、Slider Gain、Product 和 Sum

Source 中的 Constant 模块；

Continuous 中的 Integrator 模块；

Signal Routing 中的 Mux 模块：将两路输入进行向量化，混合成一路输出，直接连接到示波器上，可同时绘制两条曲线；

Sinks 中的 Scope 模块和 XY Graph 模块。由 Scope 模块显示系统各状态的时间响应曲线，XY Graph

显示系统的相平面曲线。

按照图 8.29 构建系统 Simulink 模型。

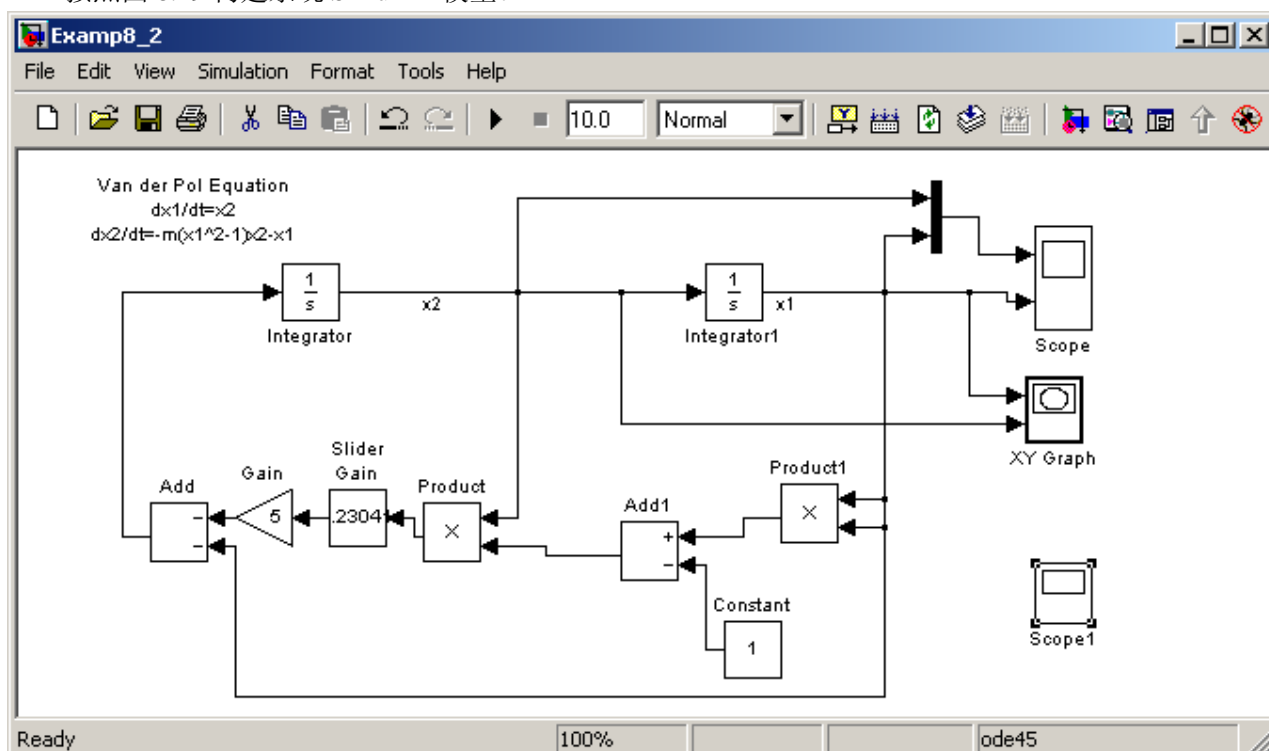



图 8.29 求解 Van der Pol 方程的 Simulink 模型

二、模型参数及系统仿真参数设置

1、系统模块参数设置为：


(1) Integrator 模块设置：将 Integrator 和 Integrator1 模块的初始值分别设置为 -2 和 1；

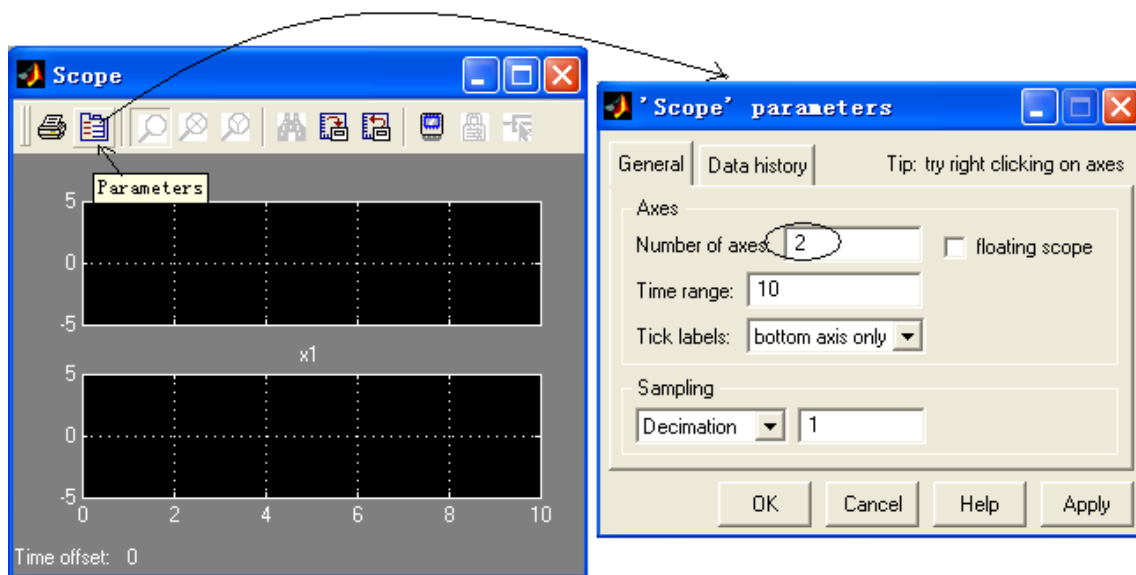
(2) Scope 模块设置：双击 Scope 模块，打开 Scope 显示窗口，见图 8.30(a)，点击 Parameter 图标  即可打开 Scope 参数设置窗口，将坐标轴数目设置为 2，即可产生两个示波器。

(3) XY Graph 模块设置：将 x 轴和 y 轴的最小、最大值均分别设置为 -5 和 5。

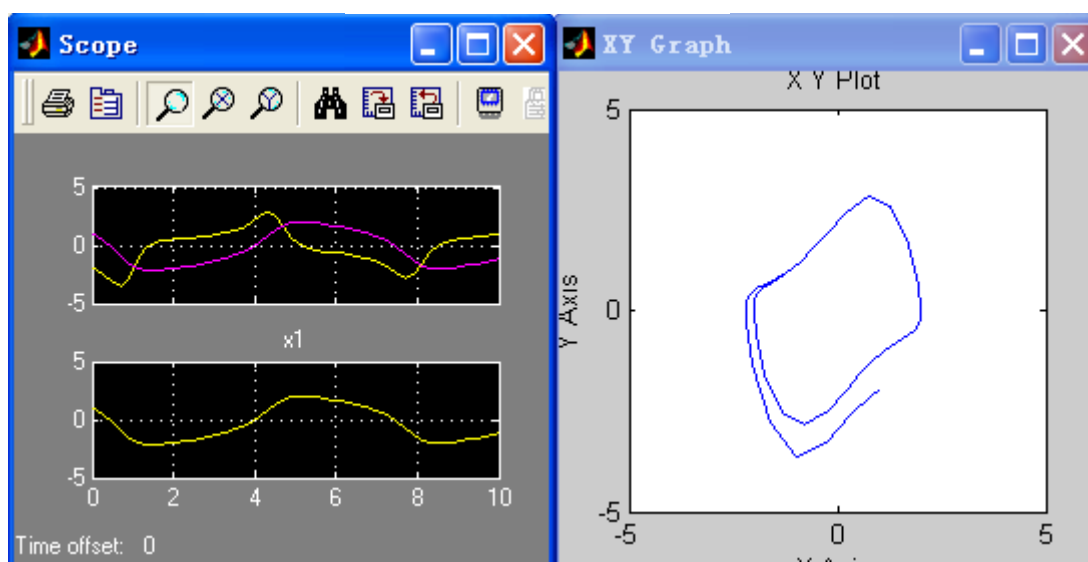
(4) 其它模块设置可参见图 8.29。

2、仿真参数的设置：均参用 Simulink 的默认值。

点击仿真开始图标 ，启动仿真程序，仿真结束后，系统的仿真结果可由 Scope 和 XY Graph 模块观察。见图 8.30(b)和(c)。



(a) Scope 模块设置



(b) 时间响应曲线

(c) 相平面曲线

图 8.30 Van der Pol 方程求解的 Scope 模块设置及仿真结果

8.4.2 离散系统仿真分析

许多离散系统中包含多种不同的采样速率。通常在离散控制系统中，控制器的更新速率一般要低于对象本身的工作频率，显示系统的更新速率更是比显示器的可读速度低得多。

例 8.3 假设某离散系统的状态方程为
$$\begin{cases} x_1(k+1) = x_1(k) + 0.1x_2(k) \\ x_2(k+1) = -0.05\sin(x_1(k)) + 0.094x_2(k) + u(k) \end{cases}$$
，其中 $u(k)$ 是

输入。该过程的采样周期是 0.1s。控制器采用采样周期为 0.25s 的比例控制器，显示系统的更新周期为 0.5s。

解：

一、系统模型创建，见图 8.31，需用的模块有：

Math Operations 中的 Gain 和 Sum；

Source 中的 Constant 模块;
Discrete 中的 Unite Delay 和 Zero-Order Hold 模块;
Sinks 中的 Scope 和 Display 模块;
User-Defined Function 中的 Fcn 模块。

二、系统模块参数及仿真参数设置

1、系统模块参数设置为:

(1) Unite Delay 和 Unite Delay1 模块设置: 双击模块, 打开模块参数窗口, 将此两个模块的采样时间均设置为 0.1;

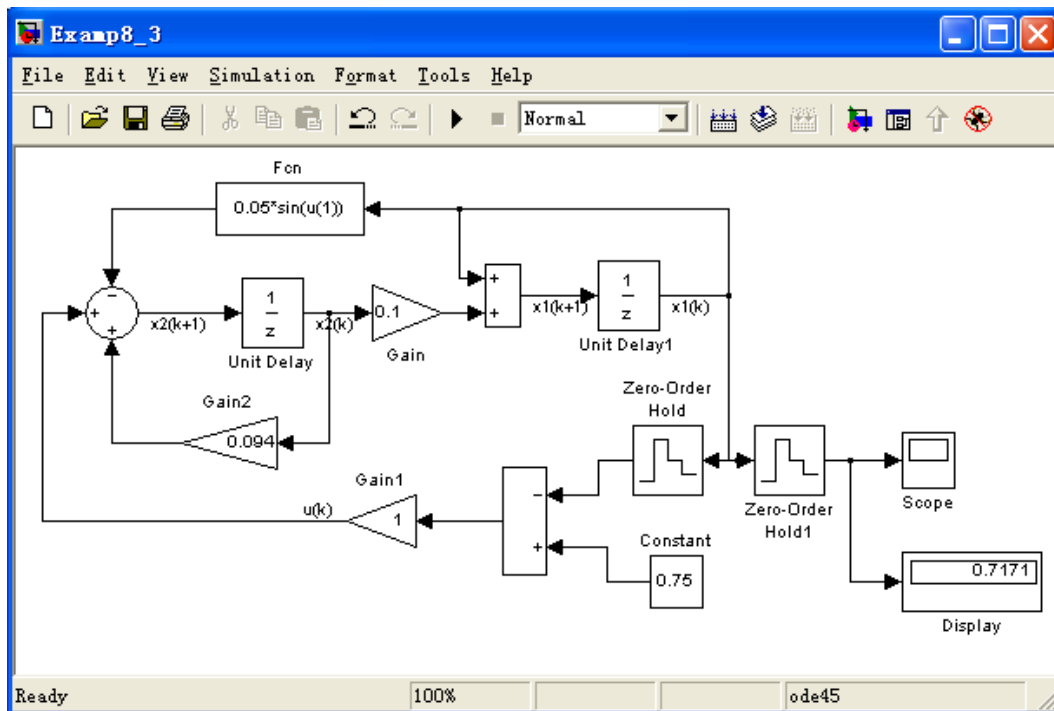
(2) Zero-Order Hold 和 Zero-Order Hold1 模块设置: 双击模块, 打开模块参数窗口, 将 Zero-Order Hold 和 Zero-Order Hold1 模块的采样时间分别设置为 0.25 和 0.5。

Fcn 模块设置: 双击 Fcn 模块, 将其表达式设置为 $0.05 * \sin(u(1))$;

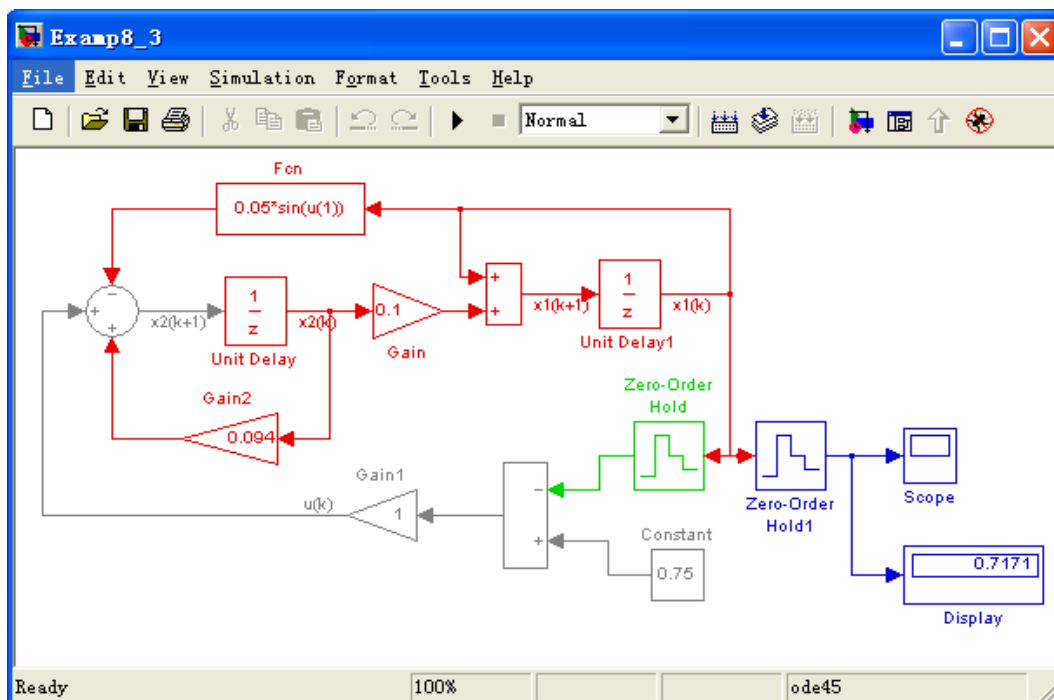
其它模块设置可参见图 8.31(a)。

2、仿真参数的设置: 求解器设置为离散求解器, 其余均参照 Simulink 的默认值, 见图 8.32。

在多采样周期的复杂系统中, 为了分清各部分信号的采样周期, 可以用不同的颜色标记不同采样周期的信号。具体的方法是 format 菜单下, 点击 Port/Signal Displays—Sample time colors 即可, 见图 8.31 (b)。



(a) 未加注采样周期的仿真模型



(b) 加注不同采样周期的仿真模型

图 8.31 例 8.3 离散系统仿真模型

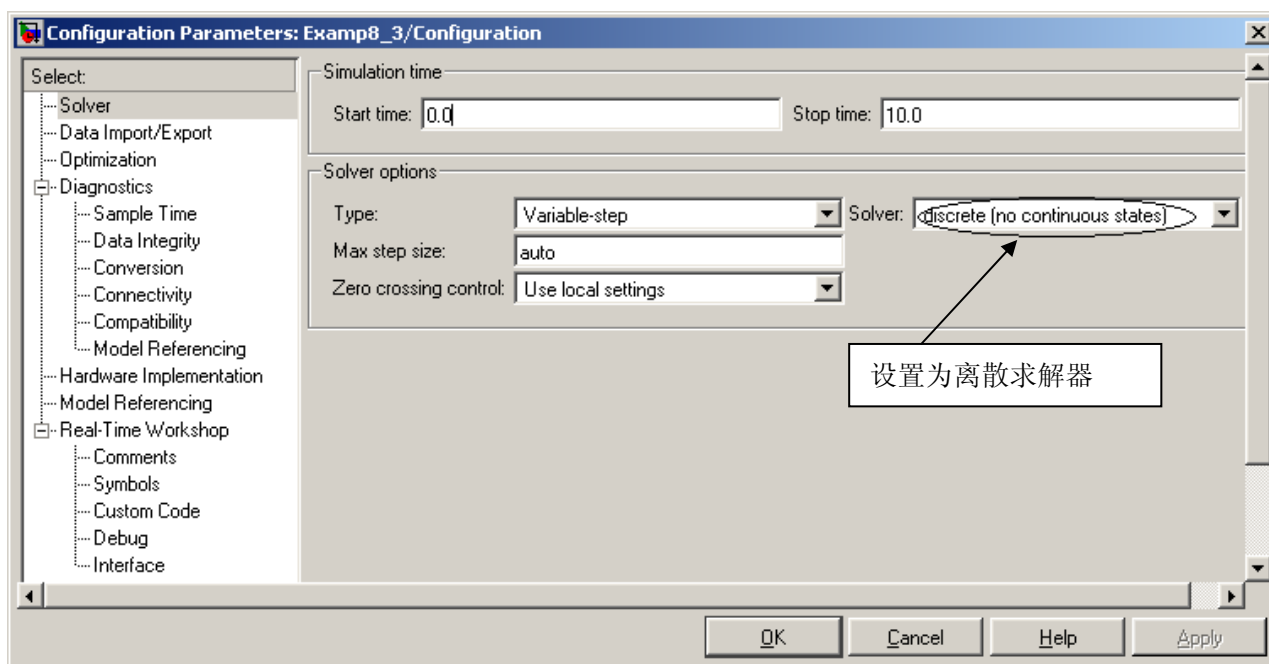



图 8.32 离散系统仿真参数设置

三、仿真

将系统模块参数与系统仿真参数设置之后，点击仿真开始图标 ，启动仿真程序，仿真过程中，Display 模块实时显示 $x_1(k)$ 的数值， $x_1(k)$ 的历史记录，可以由 Scope 模块观察。见图 8.33。

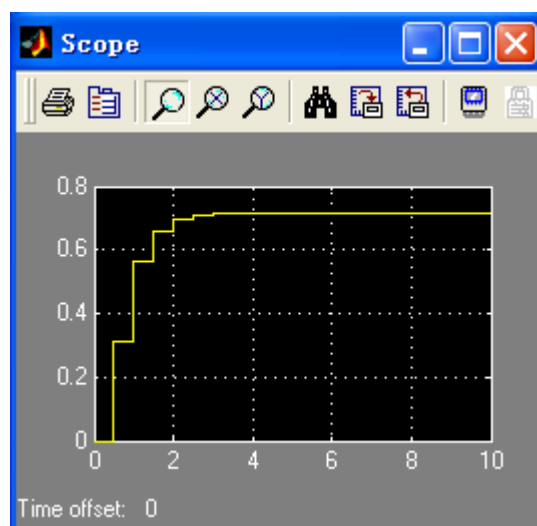


图 8.33 例 8.3 离散系统仿真结果

8.4.3 混合系统仿真分析

实际的系统常常是混合系统（即系统中有连续信号也有离散信号）。在对这类系统进行仿真时必须考虑连续信号和离散信号采样时间之间的匹配问题。Simulink 中的变步长连续求解器充分考虑了上述问题。所以在对混合系统进行仿真分析时，应该使用变步长连续求解器。

例 8.4 汽车行驶控制系统是应用很广的控制系统之一，控制的目的是对汽车速度进行合理的控制。它是一个典型的反馈控制系统，其工作原理如下：

使用汽车速度操纵机构的位置变化量设置汽车的指定速度，这是因为操纵机构的不同位置对应着不同的速度；测量汽车的当前速度，求取它与指定速度的差值；由差值信号产生控制信号驱动汽车产生相应的牵引力以改变并控制汽车速度直到达到指定速度。在对这个系统进行建模仿真前，需要先对此系统做简单的介绍。

汽车行驶控制系统包含三部分机构。

第一部分，速度操纵机构的位置变换器

位置变换器是汽车行驶控制系统的输入，其作用是将速度操纵机构的位置转换为相应的速度，速度操纵机构的位置和设定速度间的关系为：

$$v = 50x + 45 \quad x \in [0,1]$$

其中 x 为速度操纵机构的位置， v 是计算所得的设定速度。

第二部分，离散 PID 控制器

离散 PID 控制器是汽车行驶控制系统的核心部分。其作用在于根据汽车当前速度与设定速度的差值，产生相应的牵引力。其数学模型为：

$$\text{积分环节: } x(n) = x(n-1) + u(n)$$

$$\text{微分环节: } d(n) = u(n) - u(n-1)$$

$$\text{系统输出: } y(n) = Pu(n) + Ix(n) + Dd(n)$$

其中 $u(n)$ 是控制器输入，是汽车当前速度与设定速度的差值。 $y(n)$ 是控制器输出，即汽车的牵引力， $x(n)$

是控制器中的状态量。 P, I 和 D 分别是 PID 控制器的比例、积分和微分控制参数，在本例中取值分别为

$$P=1, I=0.01 \text{ 和 } D=0。$$

第三部分，汽车动力机构

汽车动力机构是行驶控制系统的执行机构。其功能是在牵引力的作用下改变汽车速度，使其达到设定的速度。牵引力与速度之间的关系为

$$F = m\dot{v} + bv$$

其中 v 是汽车速度， F 是汽车的牵引力， $m=1000\text{kg}$ 是汽车质量， $b=20$ 是阻力因子。

解：

一、系统模型创建

按照前面给出的汽车行驶控制系统的数学模型，构建系统的 Simulink 仿真模型，见图 8.34 (a)。此仿真模型需要的系统模块有：

Math 模块库中的 Gain 和 Slider Gain 滑动增益模块：Slider Gain 滑动增益模块用来调节位置变换器的输入信号 x 的取值；

Discrete 模型库中的 Unit Delay 单位延迟模块：产生信号的一步延迟，以实现 PID 控制算法；

Continuous 模型库中的 Integrator 积分器模块；

Math Operation 模型库中的 Sum 模块；

二、系统模块参数及仿真参数设置

1、系统模块参数设置

Slider Gain 模块：最小值 Low 为 0，最大值 High 是 1，可取 0~1 之间的任意值；

Unit Delay 模块：初始状态为 0，采样时间为 0.02s；

Intergrator 模块：初始状态为 0；

其余模块的参数设置参见系统仿真模型图 8.34 (a) 或使用默认取值。

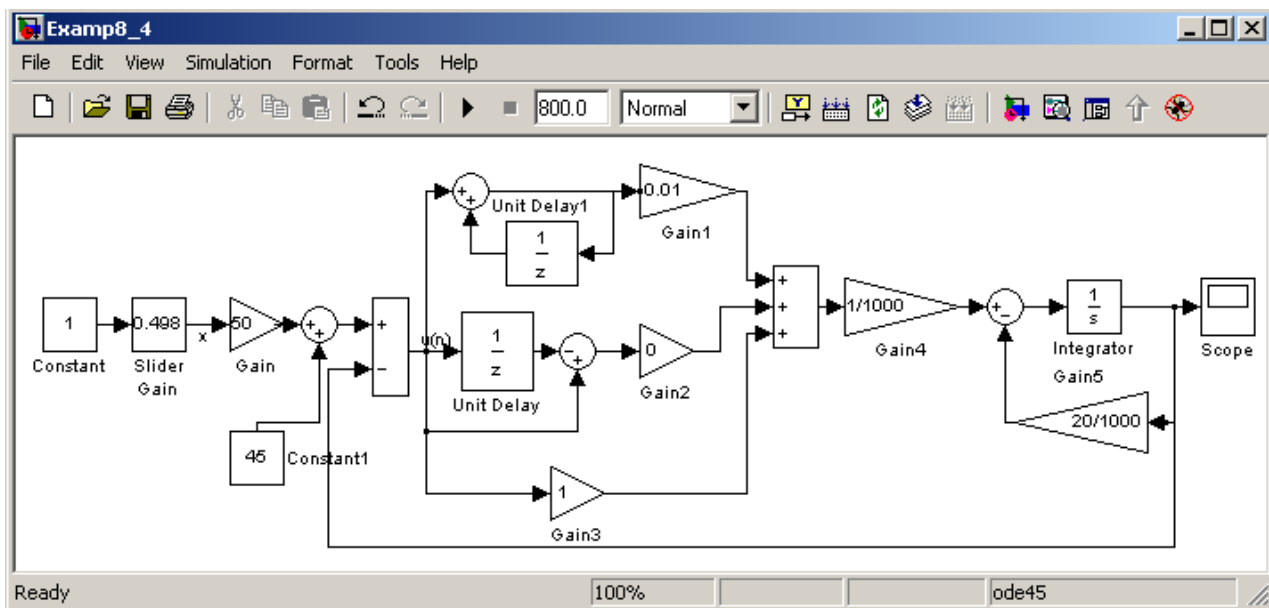
2、系统仿真参数设置

仿真时间范围：0~800；

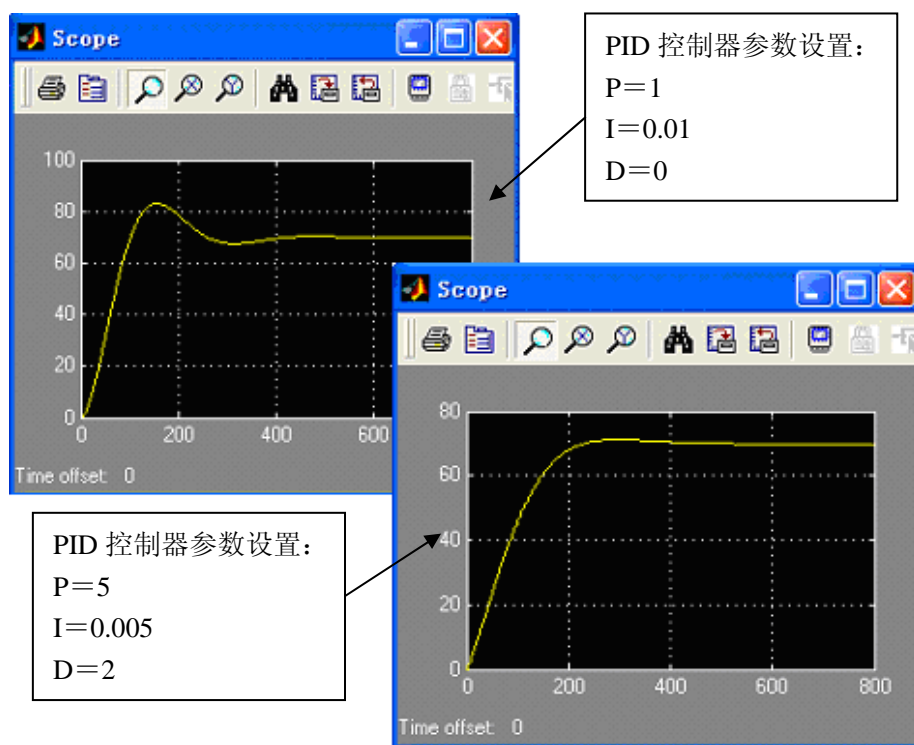
求解器：使用变步长连续求解器。

三、系统仿真与分析

将系统模块参数与系统仿真参数设置之后，对系统进行仿真，系统的仿真结果见图 8.34 (b)。



(a) 汽车行驶控制系统仿真模型



(b) 汽车行驶控制系统仿真结果

图 8.34 例 8.4 汽车行驶控制系统仿真模型及其仿真结果

8. 5 Simulink 的子系统技术

对于简单的系统，可以直接使用上面介绍的方法建立 **Simulink** 仿真模型进行动态仿真。然而，对于复杂的动态系统，直接对系统进行建模，不论是分析系统还是设计系统，都会给用户带来极大的不便。对于复杂系统，因为其动态系统中包含的功能模块较多，模块间的输入输出关系比较复杂，因而应该采用适当的策略建立系统的模型。**Simulink** 的子系统技术可以较好地解决复杂系统的建模、仿真问题。这

是因为模块中的子系统可以大大地增强系统模型的可读性。

子系统可以理解为一个单独的模块，它可以将一组相关的模块封装到它的内部，其实现的功能与其封装的模块组的功能相同。

8.5.1 子系统的生成

子系统生成的方法有两种：

一、自下而上的设计

此方法适用于对已有的系统模型建立子系统。方法是首先框选待封装的区域，即在模型编辑窗中单击鼠标左键并拖动，选中需放置到子系统模块与信号，然后选择 **Edit** 菜单下的 **Create Subsystem** 或单击鼠标右键，选中弹出菜单的 **Create Subsystem**，即可建立子系统。

例 8.5 采用自上而下设计子系统的方法构造例 8.4 汽车行驶控制系统的模型。要求将速度操纵机构的位置变换器、离散 PID 控制器、汽车动力机构分别用不同的子系统表示。

首先建立系统仿真模型，如图 8.34 (a)。将实现速度操纵机构的位置变换器功能的各个模块和信号选中，见图 8.35，单击鼠标右键，选中弹出菜单的 **Create Subsystem** 建立速度操纵机构位置变换器子系统，将子系统命名为 **Set Speed**。同法建立离散 PID 控制器和汽车动力机构子系统，分别命名为 **Discrete cruise controller** 和 **Car dynamics**，见图 8.36。

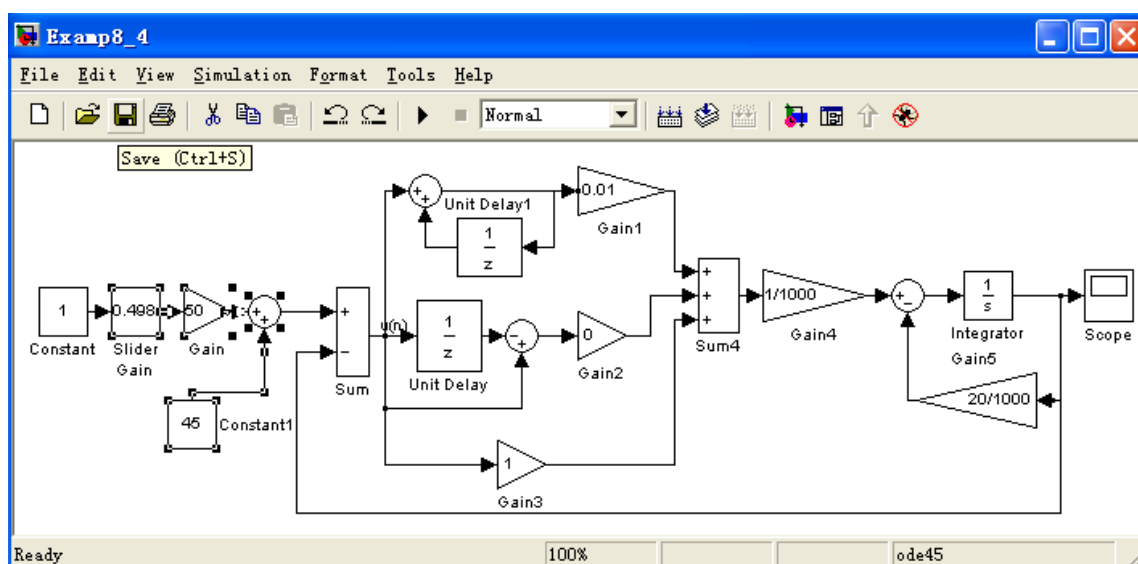


图 8.35 汽车行驶控制系统仿真模型

创建的子系统中，Simulink 自动添加了子系统的输入模块 **In1** 和输出模块 **Out1**，用户可以从图 8.36 显示的打开的子系统中看到。

二、自顶而下的设计

分析汽车行驶控制系统可知系统由 3 个功能模块组成：速度操纵机构的位置变换器、离散 PID 控制器、汽车动力机构。对于这种各部分功能较明确的系统，可以在建立系统模型时就考虑将各功能模块用不同的子系统实现。首先设计系统的总体模型，再进行细节设计。这种设计方法称作自顶而下的设计方法。

自顶而下的设计方法首先使用 **Ports & Subsystems** 模块库中的 **Subsystem** 建立子系统。这样建立的子系统内容是空的，然后双击此空子系统对其进行编辑。

需要说明的是，使用 **Ports & Subsystems** 模块库中的 **In1** 和 **Out1** 可以使子系统产生多个输入端口和多个输出端口，此两种模块只是用来对信号进行传递，完成子系统和主系统之间的通讯，不改变信号的任何属性。另外，信号标签可以越过它们进行传递。对于多输入多输出的子系统，因为需要多个 **In1** 和 **Out1**，最好使用合适的名称对它们进行命名。

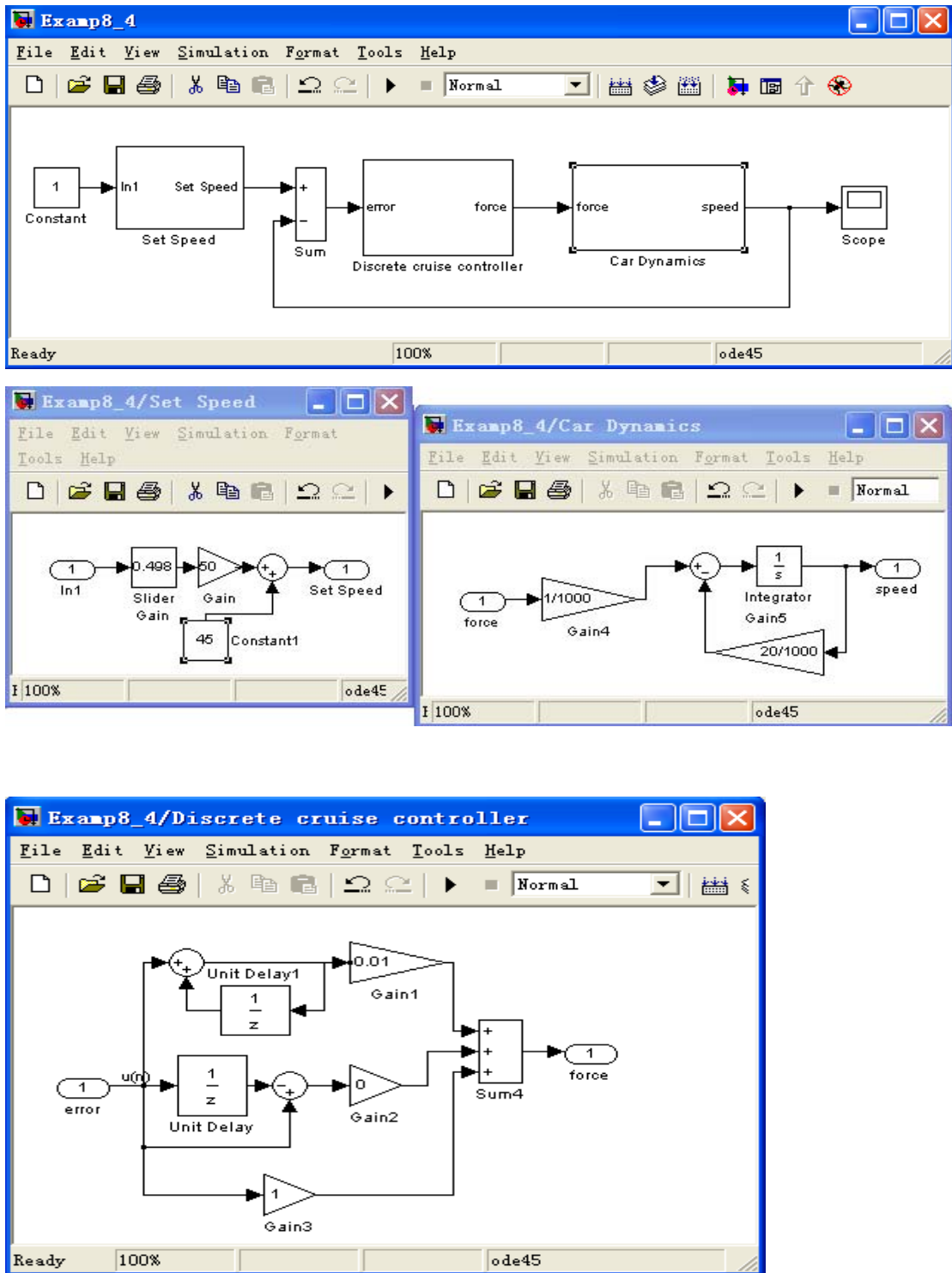


图 8.36 汽车行驶控制系统仿真模型及其各子系统

8.5.2 子系统的操作

在生成子系统之后，用户可以对子系统进行各种与系统模块相类似的操作，这时子系统相当于具有一定功能的系统模块。对子系统进行的操作可以是子系统的命名、子系统视图的修改、子系统的显示颜色等等，当然子系统也有其特有的操作，如子系统的显示、子系统的封装等等。

8.5.3 子系统的封装

使用子系统技术可以很好地改善系统模型的界面，使系统模型的可读性更好。前面介绍了如何生成子系统和子系统的操作方法。在对系统进行仿真分析时，首先需要对系统模块参数进行设置，对子系统也是这样，需要对子系统的所有模块进行正确的参数设置。在前面的介绍中，我们均是逐一地设置子系统中各模块的参数，这会给用户带来很多的不便。因为子系统一般均为具有一定功能的模块组的集合，在系统中相当于一个单独的模块，具有特定的输入输出关系。如果设计好的子系统能够像 Simulink 模块库中的模块一样进行参数设置，则会给用户带来很大的便利，这时用户只需对子系统参数选项中的参数进行设置，无需关心子系统的内部模块的组成。Simulink 的封装技术可以实现此功能，给用户提供这方面的便利。

一、封装子系统方法

将已经建立好的子系统进行封装的目的在于生成用户自定义、与子系统功能完全相同的模块。通过定义用户自己的图标。参数设置对话框以及帮助文档等，可以使封装后的子系统与 Simulink 模块库中的模块具有相同的操作。即对于封装子系统，用户可以（1）自定义子系统模块及其图标；（2）双击封装后的图标可以显示子系统的参数设置对话框；（3）自定义系统模块的帮助文档；（4）使封装后的子系统模块拥有自己的工作区。因此使用封装子系统技术具有以下优点：（1）向子系统模块中传递参数，屏蔽用户不需要看到的细节；（2）“隐藏”子系统中不需要过多展现的内容；（3）保护子系统模块中的内容，防止模块被随意更改。下面举例说明如何进行子系统封装。

例 8.6 对例 8.5 已经生成的汽车行驶控制系统的模型中的离散 PID 控制器进行封装。

封装子系统的基本过程是：首先生成需要封装的子系统，如例 8.5 中的 Discrete cruise controller 子系统；其次，将鼠标放在激活的子系统中，单击鼠标右键选择弹出菜单的 Mask Subsystem 或使用 Edit 菜单下的 Mask Subsystem 命令封装子系统，此时会弹出一个封装编辑器，见图 8.37。使用封装编辑器可以编辑封装后子系统模块的图标（Icon）、参数设置对话框（Parameters）、初始化设置对话框（Initialization）以及帮助文档（Documentation），从而使用户设计出比较友好的模块界面。

二、封装编辑器

当选择 Mask subsystem 命令封装子系统后，就会出现如图 8.37 所示的封装编辑器。使用此编辑器可以实现对封装子系统的各种编辑。封装编辑器中含有 4 个选项卡。

1、封装编辑器的图标编辑选项卡（Icon）

使用图标编辑选项卡用户可以自定义子系统模块的图标。虽然默认状态下，封装子系统不使用图标，但友好的子系统图标可以使子系统的功能一目了然。

在图标编辑选项卡中，用户可以自定义子系统模块的图标。只需在图标编辑选项卡中的子系统模块绘制命令栏（Drawing commands）中使用绘图命令举例（Examples of drawing commands）中列出的绘图命令即可绘制模块图标，并可设置不同的参数控制图标界面的显示。下面逐一介绍各对话框的使用。

（1）图标显示设置（Icon option）

图标边框设置（Frame）：设置图标边框为可见（Visible）或不可见（Invisible）；

图标透明性设置（Transparency）：设置图标为透明（Transparency）或不透明（Opaque）显示。图标设置为透明（Transparency）时，图标后面的内容如模块端口标签可以显示出来。见图 8.38，其中左侧为图标不透明设置结果，右侧为图标透明设置的结果。

图标旋转性设置（Rotation）：设置图标为固定（Fixed）或可旋转（Rotates）。见图 8.39，左侧为图标固定，图标“controller”不随模块的转动而转动；右侧为图标旋转，图标“controller”随模块的转动而转动。

图标单位设置（Units）：设置图标绘制命令所使用的坐标系单位，仅对 plot 和 text 命令有效。其选项分别为自动缩放（Autoscale）、像素（Pixels）以及归一化表示（Normalized）。其中，Autoscale 表示图标自动适合模块大小，与其成比例缩放；Pixels 表示图标绘制采用像素作为单位；Normalized 表示模块大小为单位长度，绘制命令中的坐标值不得超过单位值 1。

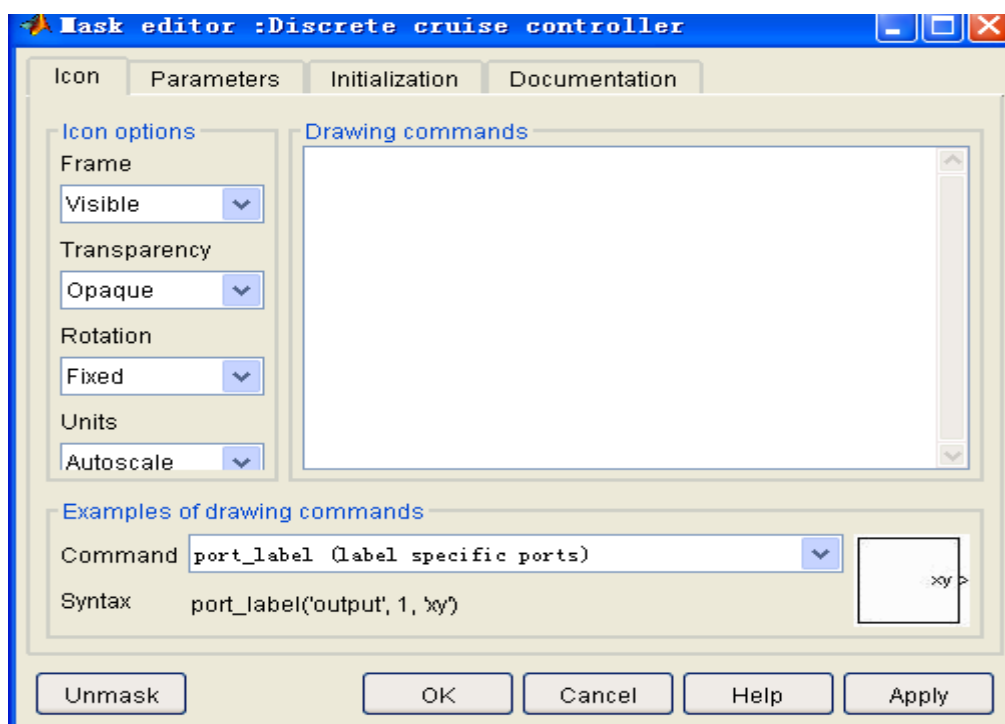
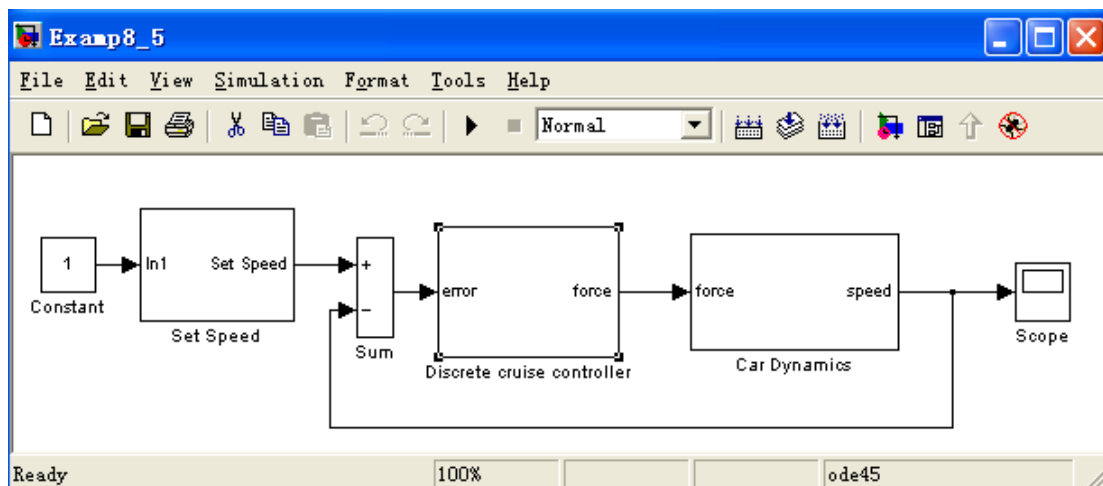


图 8.37 封装子系统及封装编辑器

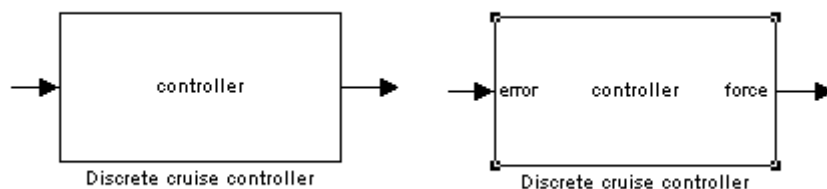


图 8.38 图标透明显示设置

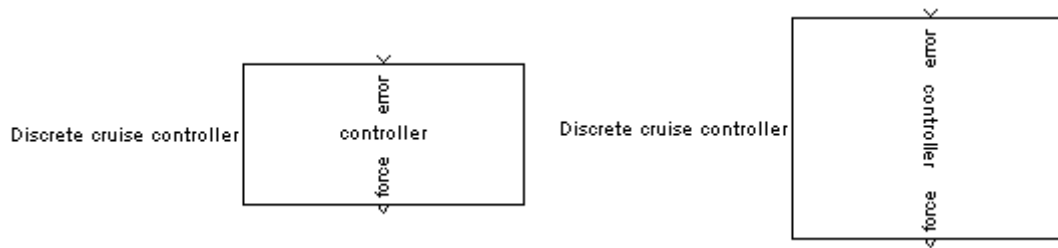


图 8.39 图标旋转显示设置

(2) 图标绘制命令栏 (Drawing commands)

封装后的子系统模块的图标均是在图标绘制命令栏中完成绘制的。使用不同的绘制命令可以生成不同的图标。可生成的图标可以是描述性文本、子系统数学模型图标、图像或图形等。如果在此栏中键入多个绘制命令，则图标的显示按照绘制命令的顺序显示。

①描述性文本图标

使用下列命令可以在模块图标上显示文本：

```
disp('text') %图标上显示 text 文本字样
disp(variablename) % variablename 为工作空间中的字符串变量名
text(x,y,'text') %在图标上特定位置显示 text 文本字样
text(x,y,stringvariablename) % stringvariablename 为已存在的字符串变量名
fprintf('text')
fprintf('format',variablename) % format 表示文本的格式
port_label(port_type, port_number, lable)
```

%此命令可以显示模块的端口名称，其中 `port_type` 为端口类型，取值为 'input' 或 'output'，`port_number` 为端口数目，`lable` 为端口文本。如果需要显示多行文本，可以使用 `\n` 换行。这时封装后的子系统图标为描述性文本。见图 8.40(a)。

②子系统数学模型图标

使用 `dpoly` 命令可以将封装的子系统模块的图标设置为系统的传递函数，使用 `droots` 命令可以设置为零极点传递函数，其命令格式为

```
dpoly(num, den)
dpoly(num, den, 'character')
droots(z,p,k)
```

其中 `num`, `den` 分别是传递函数的分子和分母多项式，`'character'` (取 `s` 或 `z`) 为系统的频率变量，`z,p,k` 分别是传递函数的零点、极点和系统增益。需要注意的是，参数 `num`, `den`, `z,p,k` 必须是工作空间中已经存在的变量，否则绘制命令的执行将出现错误。例 `num=[1 2];den=[1 3 4 1]`；将子系统图标设置为子系统传递函数的命令见图 8.40(b)。

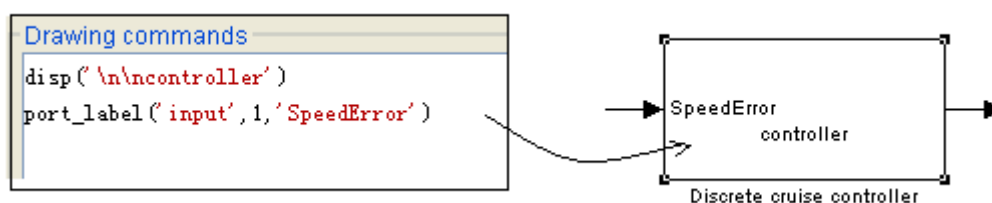
③图像或图形图标

使用 `plot` 或 `image` 命令可以将子系统模块的图标设置为图形或图像。其命令为

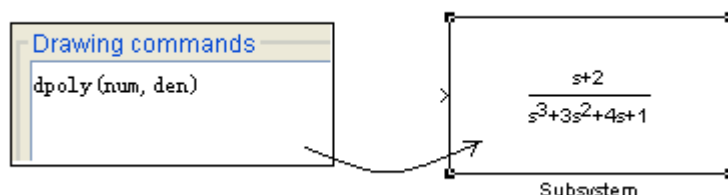
```
plot(x,y)
image(imread('photoname'))
```

例如：`x=0:0.02*pi:2*pi;y=sin(x)`；在图标命令栏键入 `plot(x,y)` 可将子系统图标设置为 `y-x` 曲线，见图 8.41 (a)。在图标命令栏键入 `image(imread('helicopter.bmp'))`，可将子系统图标设置为图像，见图 8.41 (b)。

需要注意的是，在图标中绘制图像时，图像可以是 BMP 或 JPG 格式，如不是，需要使用图形处理工具箱中的 `ind2rgb` 命令进行转换。

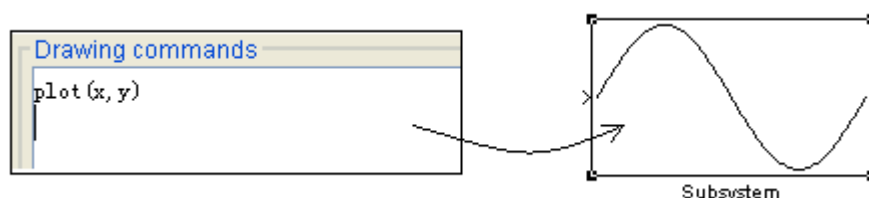


(a) 文本图标绘制举例

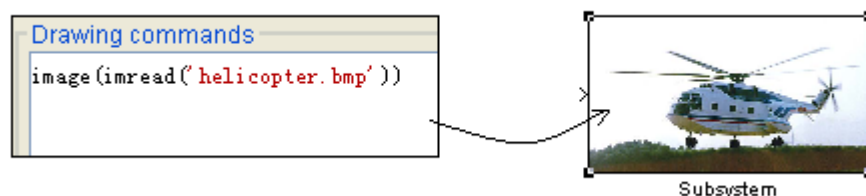


(b) 子系统传递函数图标绘制举例

图 8.40 文本及传递函数图标设置举例



(a) 图形图标设置举例



(b) 图像图标设置举例

图 8.41 图形、图像图标设置举例

2、封装编辑器的参数设置选项卡（Parameters）

子系统封装的目的之一是提供一个友好的参数设置界面。通常用户不需了解系统内部的细节，只需提供正确的模块参数即可完成对系统的设计与仿真。只有使用了子系统封装编辑器中提供的参数设置选项卡（Mask Editor 下的 Parameters 选项卡）进行子系统参数设置，才可以说是真正完成了子系统的封装，从而使用户设计出与 Simulink 模块库中的模块同样直观的参数设置界面。

不同于通常的子系统，封装的子系统具有独立的工作区。这是由于在没有对子系统进行封装之前，子系统内的模块可以直接使用 MATLAB 工作空间中的变量。通常的子系统可以看作是图形化的 MATLAB 脚本，即是说子系统只是将一些由模块实现的命令以图形化的方式组合起来。而封装的子系统的内部参数对系统模型中的其它系统不可见，而且只能使用参数设置选项卡输入。

在例 8.6 对汽车行驶控制系统的模型中的离散 PID 控制器进行封装，并进行了图标绘制，这里仍针对该例说明如何对封装后的子系统的参数设置选项卡进行设置，在本封装子系统模块的参数设置界面中，应该提供控制律所需的 P、I 和 D 参数，见图 8.42。

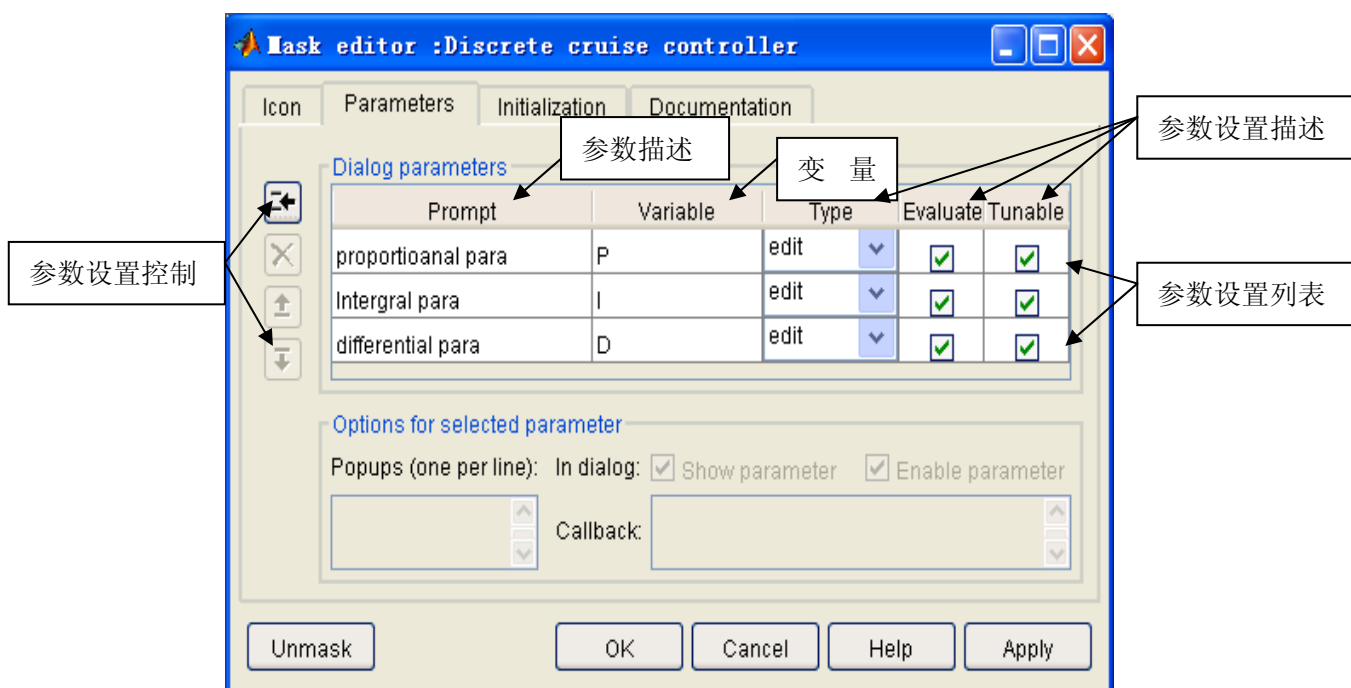


图 8.42 封装子系统模块参数设置选项卡

设置选项卡中包括以下几种设置内容：

(1) 参数设置控制

参数设置控制包括添加（Add）、删除（Delete）、上移（Move up）和下移（Move down），分别表示在即将生成的参数设置界面中添加、删除、上移与下移模块需要的输入参数。

(2) 参数描述

参数描述是对模块输入的参数作简要的说明，在子系统参数设置界面中用来区别不同的参数，因而其取值最好能够说明参数的意义或作用。

(3) 变量

用来指定键入的参数值将要传递给的封装子系统工作空间的相应变量，此处使用的变量必须与子系统中所使用的变量名相同。

(4) 参数设置描述

参数设置描述包括参数控制类型（Type）、是否为求值字符串（Evaluate）、是否可调整（Tunable）复选框。其中控制类型包括 Edit（需要用户在参数设置界面中键入参数值，适合多数情况）、Checkbox（复选框，表示逻辑值）和 Popup（在参数设置界面中弹出参数选项以便选择参数，弹出的参数选项值在 Popups 栏中输入）。

如图 8.42 所示对封装子系统进行参数设置后，双击封装的子系统即可打开子系统的参数设置界面，见图 8.43。用户只需在此界面中输入正确的参数即可进行系统的仿真设计分析。

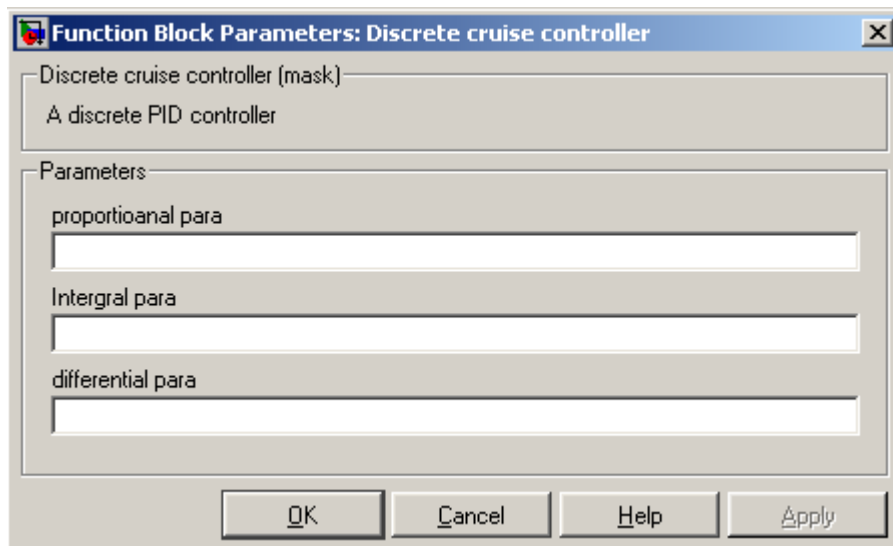


图 8.43 封装后子系统的参数设置界面

3、封装编辑器的初始化设置选项卡（Initialization）

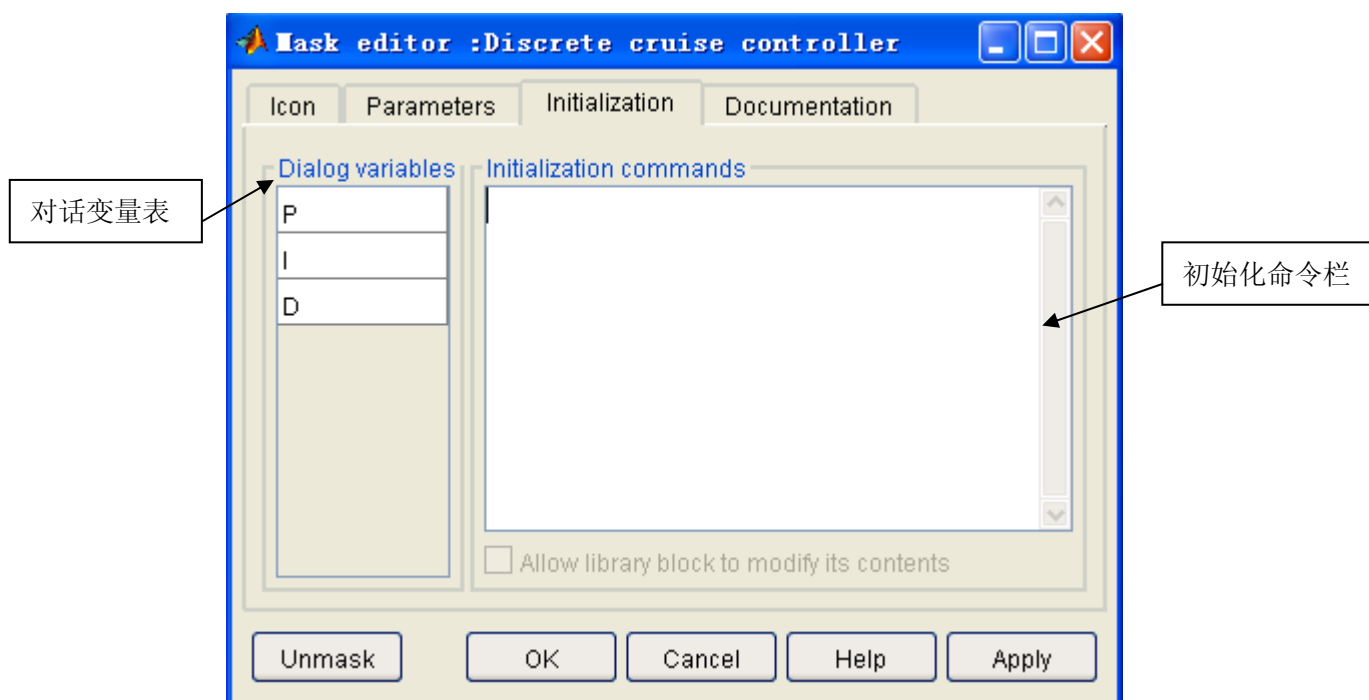


图 8.44 初始化设置选项卡

初始化设置选项卡中的对话变量表是用户设置了参数设置选项卡后自动生成的。初始化命令一般为 MATLAB 命令，在初始化命令栏中可以定义封装后子系统工作空间中的各种变量，这些变量可以被封装子系统模块图标绘制命令、其他初始化命令或子系统模块使用。在下列情况下，Simulink 开始执行初始化命令：

- 模型文件被载入；
- 框图被更新或模块被旋转；
- 绘制封装子系统模块图标时。

4、封装编辑器的文档编辑选项卡（Documentation）

Simulink 模型库中的模块均提供了模块的简单描述和详细的帮助文档，方便用户的使用和理解。对于用户封装的子系统模块，封装编辑器的文档编辑选项卡可以使用户建立被封装子系统的所有帮助文档。对于例 8.6 封装的离散控制器的文档编辑见图 8.45。

8.5.4 Simulink 高级子系统技术

前面介绍的子系统的输出和输入具有一定的对应关系，对应一定的输入，子系统必定会产生输出。但在有些情况下，只有满足一定的条件时，子系统才被执行；即子系统的执行依赖于其它信号，这个称为控制信号的信号从子系统单独的端口即控制端口输入。这样的子系统称为条件执行子系统。在条件执行子系统中，子系统的输出不仅依赖于子系统本身的输入信号，而且受子系统控制信号的控制。

根据控制信号对条件执行子系统所控制方式的不同，可以将条件执行子系统划分为如下几种类型：

使能子系统：控制信号的值为正时，子系统开始执行；

触发子系统：当控制信号的符号发生变化时（即信号发生过零时），子系统开始执行。触发子系统的触发执行有三种形式：控制信号上升沿触发形式、控制信号下降沿触发形式和控制信号双边沿（上升沿或下降沿）触发形式。

函数调用子系统：当用户自定义的 S 函数中发出函数调用命令时，子系统开始执行。

下面通过示例说明各种条件执行子系统。

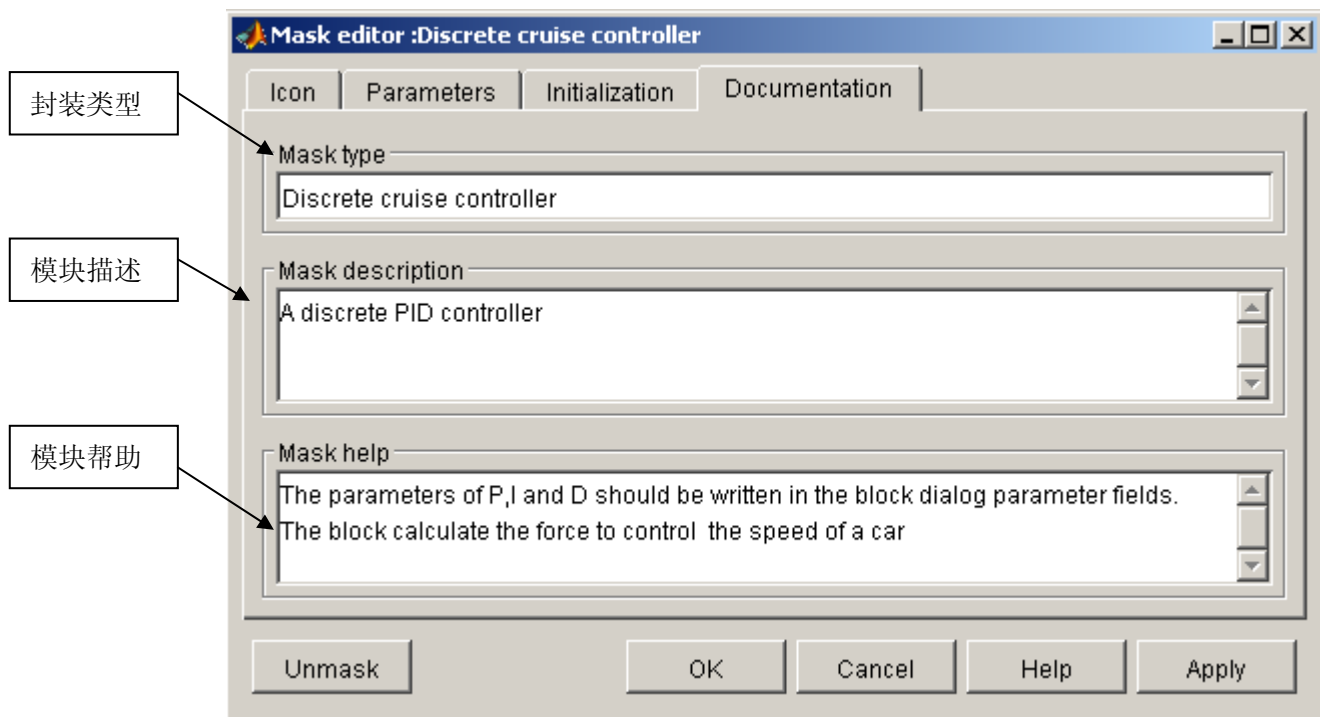


图 8.45 封装编辑器的文档编辑选项卡

一、使能子系统

用户可以使用 Simulink 的 Ports & Subsystems 模块库中的 Enable Subsystem（使能子系统）模块、Triggered Subsystem（触发子系统）模块及 Enable and Triggered Subsystem(使能触发子系统)模块构建条件执行子系统。使能子系统是指当子系统的使能信号为正时，子系统才开始执行。

例 8.7 图 8.46 所示的系统模型中，存在着两个由方波信号驱动的使能子系统（A 和 B）。当控制信号（即系统模型中的方波信号）为正时开始执行子系统 A，控制信号为负时开始执行子系统 B。

系统模型中各模块的参数设置：

Sine wave 模块：采用默认值；

Signal generator 模块：设置为周期为 4 的方波信号，其余采用默认值；

使能子系统 A 的使能状态设置为 reset，子系统 B 的使能状态设置为 held。

scope 模块：设置为 3 个坐标轴

系统仿真参数设置：

仿真时间设置为 0~20s；

其余采取默认设置。

运行系统仿真，其仿真结果见图 8.47。

需要说明的是，在使能信号的使能状态设置中，选择状态重置 reset 表示在使能子系统开始执行时，系统中的状态被重新设置为初始参数值；而状态保持 held 表示在使能子系统开始执行时，系统中的状态保持不变。从图 8.47 中可以看出，使能子系统 A 采用状态重置，在子系统开始执行时，其输出被重置；而使能子系统 B 采用状态保持，在子系统开始执行时，其输出被保持不变。

二、触发子系统

触发子系统是指在控制信号的符号发生改变时（即控制信号出现过零事件时），子系统才开始执行。根据控制信号符号改变方式的不同将触发子系统分为三类：上升沿触发子系统、下降沿触发子系统和双边沿（上升沿或下降沿）触发子系统。

例 8.8 图 8.48 所示的系统模型中，有三个使用不同触发方式的触发子系统。

系统模型中各模块的参数设置：

Ramp 模块：斜率 **slope** 设置为 1，其余采用默认值；

Signal generator 模块：设置为周期为 1 的方波信号，其余采用默认值；

触发子系统的触发方式分别设置为 **Rising**、**Falling** 和 **Either**。

scope 模块：设置为 4 个坐标轴

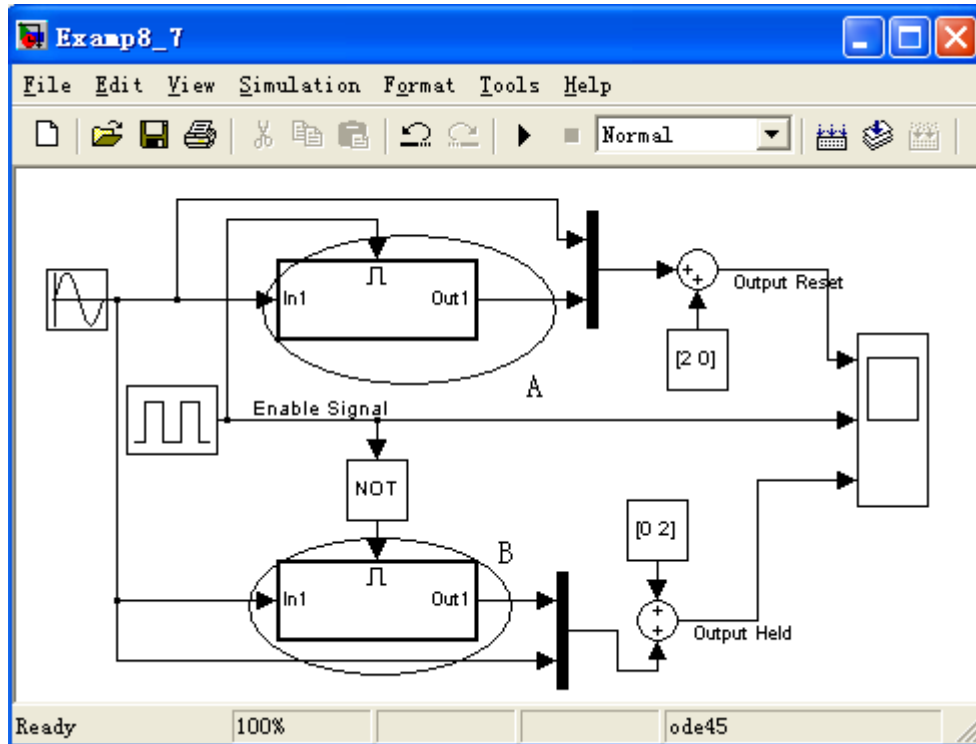
系统仿真参数设置：

仿真时间设置为 0~10s；

其余采取默认设置。

运行系统仿真，其仿真结果见图 8.49。其中，第一个示波器输出系统正弦信号和触发控制方波信号。第二至第四示波器分别输出上升沿、下降沿和双边沿触发子系统的输出。

需要说明的是，触发子系统具有零阶保持特性，即系统在触发信号控制下开始执行的时刻，系统由输入产生相应的输出，当触发信号离开过零时，系统的输出保持在原来的输出值，不发生变化。从仿真结果图 8.49 可以明显地看出触发子系统的零阶保持特性。此外，由于触发子系统的执行依赖于触发控制信号，因此对于触发子系统而言，不能指定常值的采样周期，只有带有继承采样时间的模块才能够在触发子系统中应用。



Block Parameters: Enable

Enable Port

Place this block in a subsystem to create an enabled subsystem.

Parameters

States when enabling: **reset**

☐ Show output port

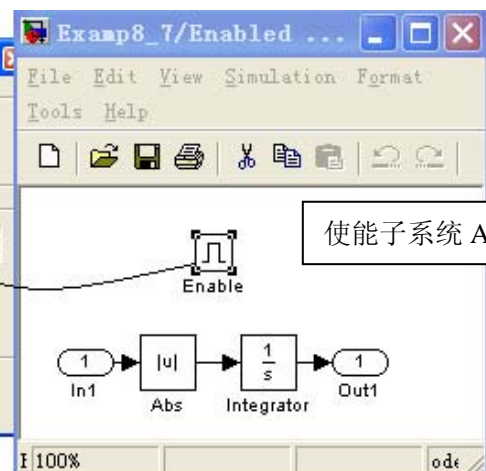
☒ Enable zero crossing detection

OK

Cancel

Help

Apply



使能子系统 A 结构

Block Parameters: Enable

Enable Port

Place this block in a subsystem to create an enabled subsystem.

Parameters

States when enabling: **held**

☒ Show output port

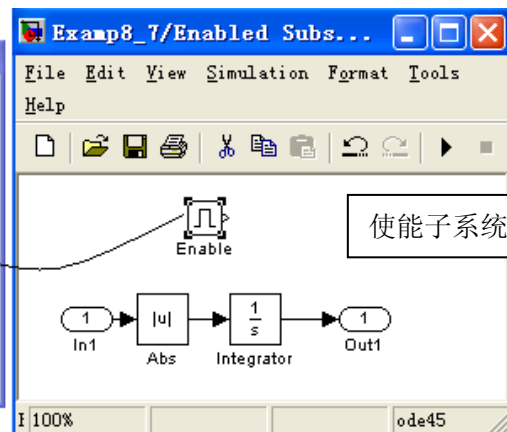
☒ Enable zero crossing detection

OK

Cancel

Help

Apply



使能子系统 B 结构

图 8.46 使能子系统模型及其使能参数设置

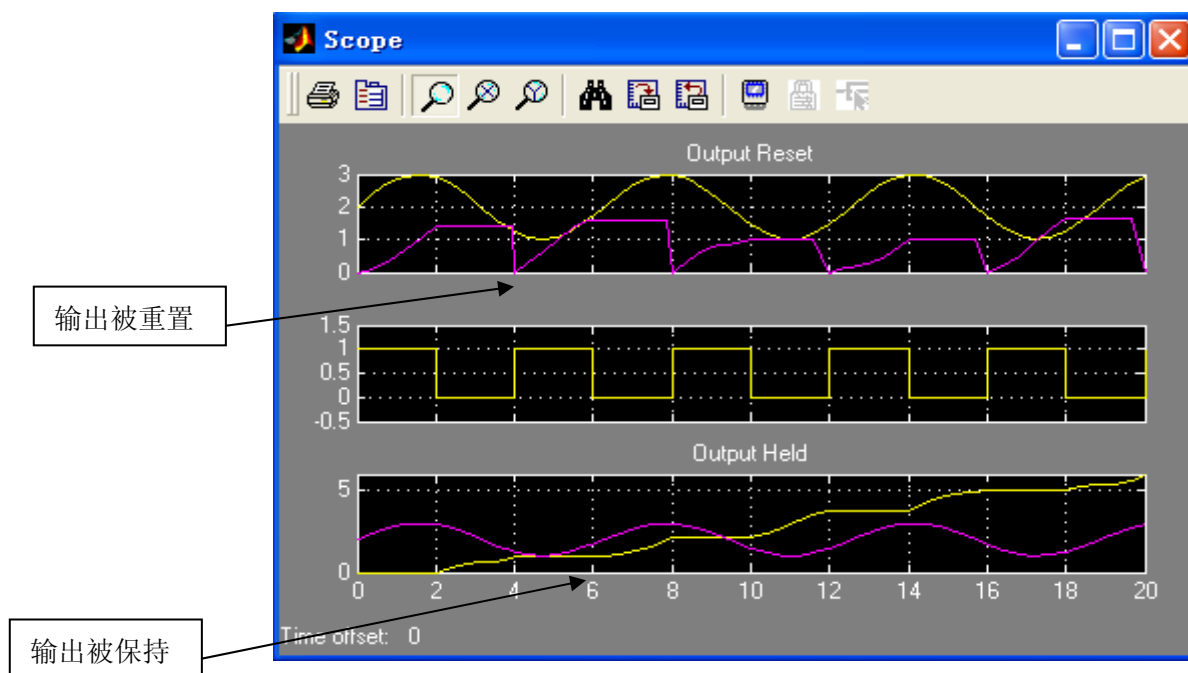


图 8.47 使能子系统仿真结果

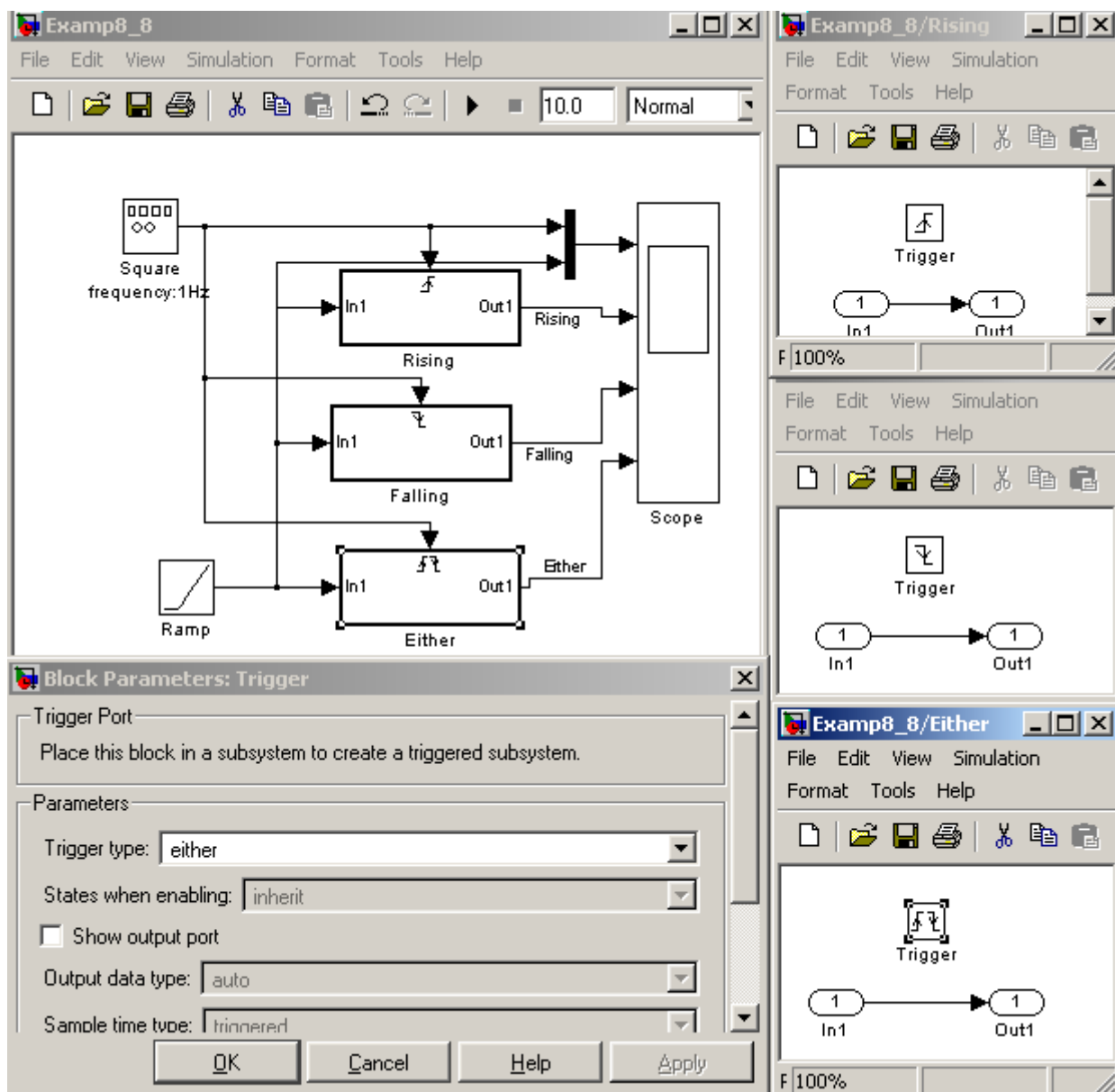


图 8.48 触发子系统模型及其参数设置

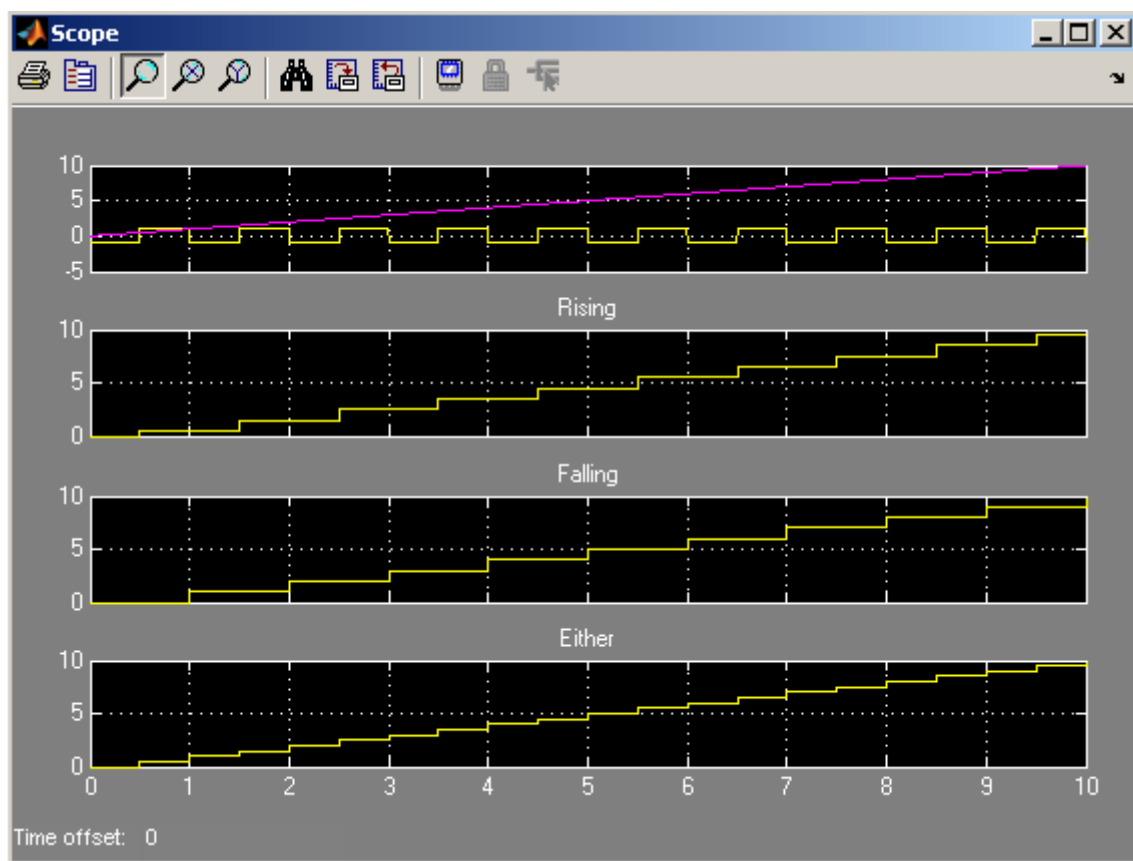


图 8.49 触发子系统模型仿真结果

三、触发使能子系统

某些条件执行子系统的控制信号可能不止一个，很多情况下，条件执行子系统受到触发控制信号和使能控制信号的共同控制，只有当触发条件和使能条件同时满足时，子系统才开始执行，这样的条件执行子系统就是触发使能子系统。其工作原理如图 8.50。系统等待一个触发事件的发生（即等待触发信号的产生），当触发事件发生后，Simulink 检测使能信号，若使能控制信号为正，则子系统执行一次，否则不执行子系统。

需要指出的是，条件执行子系统不允许同时使用多个 Trigger 信号或 Enable 信号，如果必须使用多个控制信号，用户可以先用逻辑操作符，将相关的控制信号相逻辑组合，以产生单一的触发控制信号或使能控制信号。

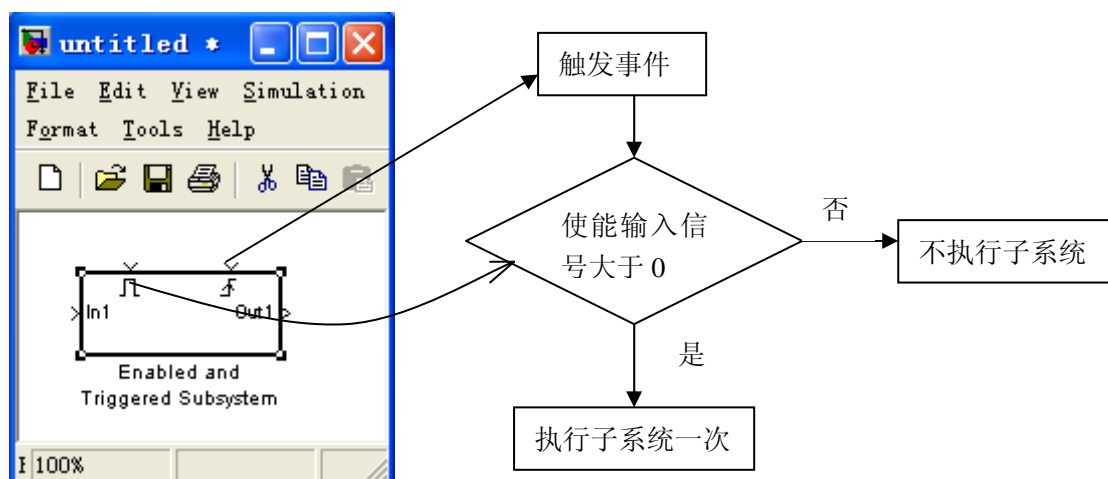


图 8.50 触发使能子系统工作原理

四、原子子系统

本节介绍了通用子系统、使能子系统和触发子系统的概念和使用方法。虽然子系统都可以将系统中相关的模块组合封装成一个单独的模块，方便用户对复杂系统进行分析，但除了这些共同点外，不同的子系统还有其各自的特性。

Simulink 在进行系统仿真时，通用子系统和使能子系统各个模块的执行不受子系统的限制，即系统的执行与通用子系统和使能子系统的存在与否无关。使用这两种子系统均可以使 **Simulink** 系统模型更层次化，增强系统模型的可读性。在执行过程中，这两种子系统模块与上一级的系统模块统一排序，模块的执行顺序与子系统本身无关，在一个仿真时间步长中，系统的执行可以多次进出同一个子系统。两种子系统相当于一种虚设的模块组容器，其中各模块与系统中其它模块的信号输入输出不受任何影响。因此，通用子系统和使能子系统被称为“虚子系统”。然而，触发子系统的工作原理与通用子系统和使能子系统的不同。在触发子系统中，当触发事件发生时，触发子系统的所有模块一同被执行。只有当触发子系统的所有模块全部执行完毕后，**Simulink** 才会转到系统模型的上一层执行其它模块。这种作为一个整体参与仿真，功能相当于一个单独的系统模块，且其中的模块在子系统中被排序执行的子系统称为“原子子系统”。

触发子系统是一种原子子系统。除此以外，有些情况下，特别是对多速率复杂系统作仿真分析时，需要将一个普通的子系统作为一个整体参与仿真计算。这是因为在多速率系统中，尤其是在生成系统可执行代码时，任何时序关系的差错都会导致整个系统仿真的失败，而且难以进行诊断分析。

在 **Simulink** 中建立原子子系统的方法有两种：

第一，使用 **Ports & Subsystems** 模型库中的 **Atomic Subsystem** 子系统模块建立一个空原子子系统，然后编辑该原子子系统；

其次，将已经建立的子系统强行转换成原子子系统。方法是先选择需要转换的子系统，选择 **Edit** 菜单下的 **Subsystem Parameters**，框选 **Treat as Atomic Unit** 即可或单击鼠标右键，在弹出菜单中选择相同选项。

五、其它子系统介绍

Simulink 的 **Ports & Subsystems** 模型库中除了前面介绍的通用子系统、使能子系统、触发子系统和原子子系统外，还提供了很多其它的条件执行子系统，见图 8.51。

1、可配置子系统（**Configurable Subsystem**）：只能在用户自定义库中使用，代表用户自定义库中的任意模块。

2、函数调用子系统（**Function-Call Subsystem**）

函数调用子系统属于触发子系统，它使用 **S** 函数的逻辑状态作为控制信号。在使用函数调用子系统时，子系统的函数触发端口必须使用 **Ports & Subsystems** 模型库中 **Function-Call Generator** 作为输入。在触发子系统中触发信号的参数设置中选择 **Function-Call** 可以将由普通信号触发的触发子系统转换为函数调用子系统。

3、For 循环子系统（**For Iterator Subsystem**）

For 循环子系统可以在一个仿真时间步长中循环执行子系统，用户可以指定在一个仿真时间步长中执行循环的次数。

4、While 循环子系统（**While Iterator Subsystem**）

与 **For** 循环子系统类似，**While** 循环子系统同样可以在一个仿真时间步长内循环执行子系统，但是其执行必须满足一定的条件。**While** 循环子系统有两种类型：当型与直到型，这与其它高级语言中的 **While** 循环类似。

5、选择执行子系统（**Switch Case Action Subsystem**）

与 **C** 语言和 **M** 语言中的 **Switch Case** 语句的功能类似，在某些情况下，系统对于输入的不同取值分别执行不同的功能，选择执行子系统可以完成这种功能。需要指出的是，选择执行子系统必须同时使用 **Switch Case** 模块。

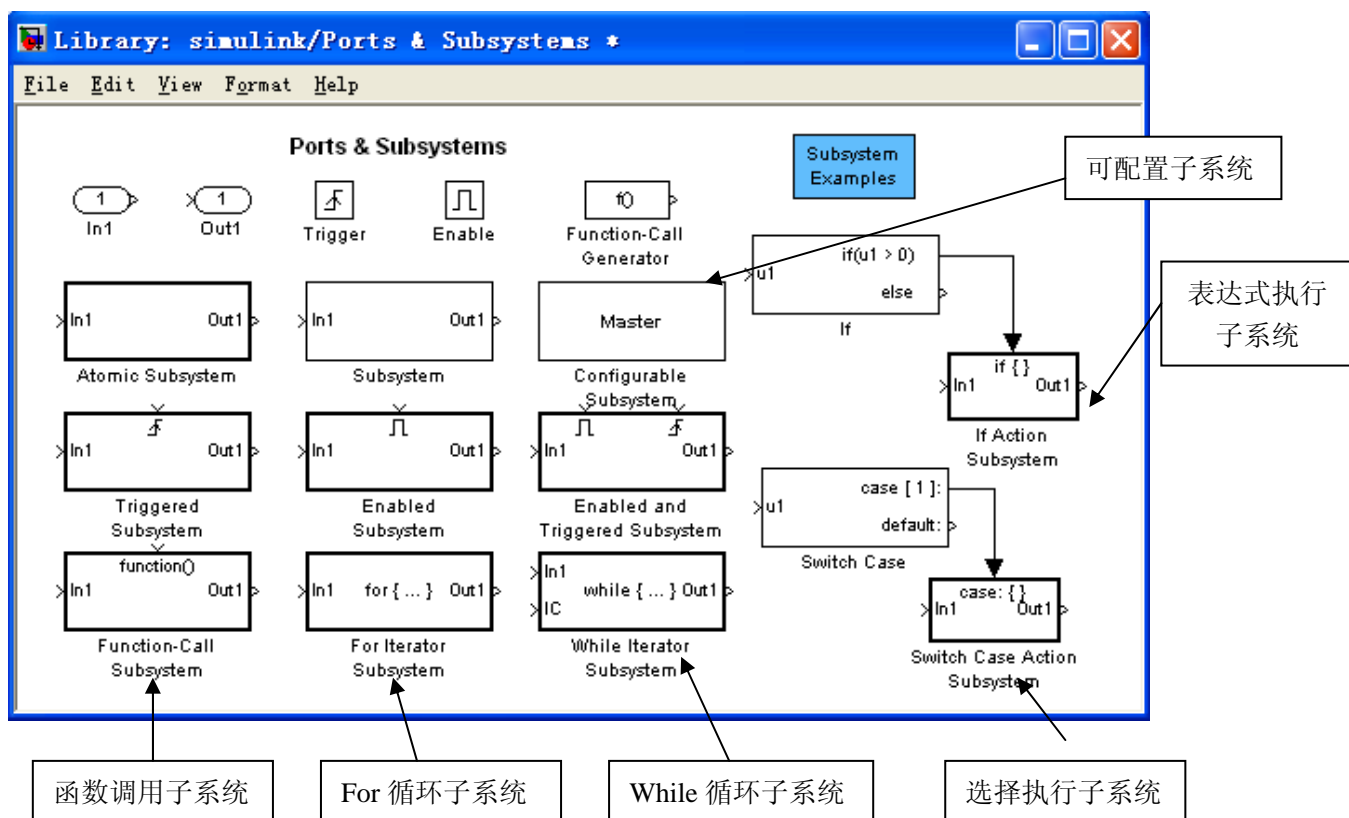


图 8.51 Ports & Subsystems 模型库中的所有模块

6、表达式执行子系统（If Action Subsystem）

与 C 语言和 M 语言中的 If-Else 语句的功能类似，表达式执行子系统的执行依赖于逻辑表达式的取值。表达式执行子系统必须同时使用 If 模块。


这里对 Ports & Subsystems 模型库中一些子系统模块的功能做了简单的说明，Ports & Subsystems 模型库中同时给出了这些子系统模块的使用算例，感兴趣的读者可以打开算例进行参考。

8.6 Simulink 的调试技术

功能强大、界面友好的调试功能是系统设计开发平台所必备的条件之一。Simulink 作为高性能的系统设计、仿真和分析平台，给用户提供了强大的模型调试工具。通过 Simulink 的调试工具，用户可以对动态系统模型进行调试，一种方法接着一种方法地运行仿真，以检查那种方法的仿真结果，发现其中可能存在的问题，并进行修改，从而快速完成系统设计、仿真和分析工作。

不同领域的不同系统模型的复杂程度相差悬殊，对系统模型进行调试的复杂程度也不相同。Simulink 所提供的图形调试器可以满足多数应用领域模型的调试。

8.6.1 Simulink 图形调试器启动

使用 Simulink 的 Tools 菜单下的 Simulink debugger 命令或 Simulink 模型工具栏中的调试器按钮  可以启动调试器。图 8.52 是 Simulink 图形调试器窗口。

8.6.2 调试器的操作与功能

启动调试器，设置合适的调试断点之后，便可对系统模型中指定的模块和信号进行调试。Simulink 调试器能完成怎样的功能以及经过怎样的设置可以完成这样的功能？这些问题是用户在设置断点进行调试之前需要了解的。

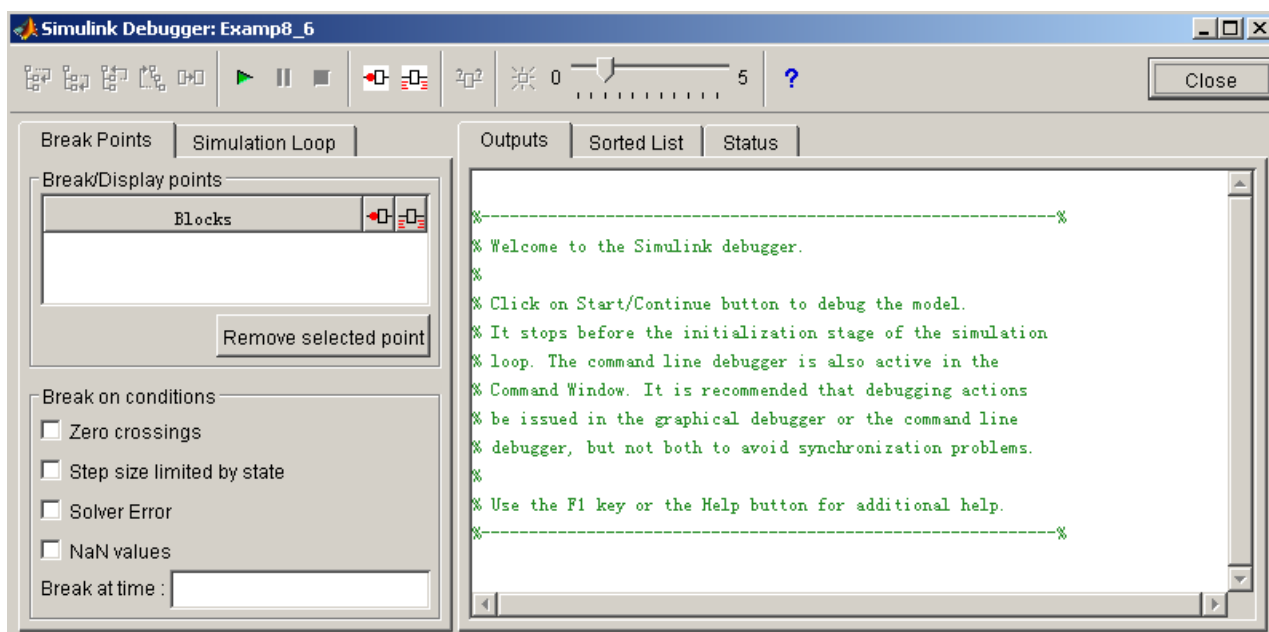


图 8.52 Simulink 图形调试器窗口

一、Simulink 调试器工具栏

Simulink 调试器工具栏命令及其功能见图 8.53。其工具栏包含如表 8.1 所示的命令键。



图 8.53 Simulink 调试器工具栏命令

表 8.1 Simulink 调试器工具栏命令键使用说明

命 令 键	用 途
	进入下一种方法，中断于当前和下一种方法之间
	跳过下一种方法
	跳出当前状态
	在下一时间步中执行第一种状态
	执行到下一模块
	开始或继续仿真
	暂停仿真
	中断仿真
	在指定模块之前设置断点

	执行时，显示指定模块的输入输出
	显示指定模块当前的输入输出
	选择动画方式开或关
	调试器在线帮助
Close	关闭调试器

二、断点显示及断点条件设置

Simulink 为用户提供了友好的调试界面。用户可以在断点显示框中了解到当前断点的信息，如断点位置、断点模块的输入输出等，如图 8.54（a）所示。

很多情况下，用户除了在调试器在指定模块前设置断点外，还需要在一定条件下设置系统断点。Simulink 调试器提供了五种断点条件设置，如图 8.54（b）所示。

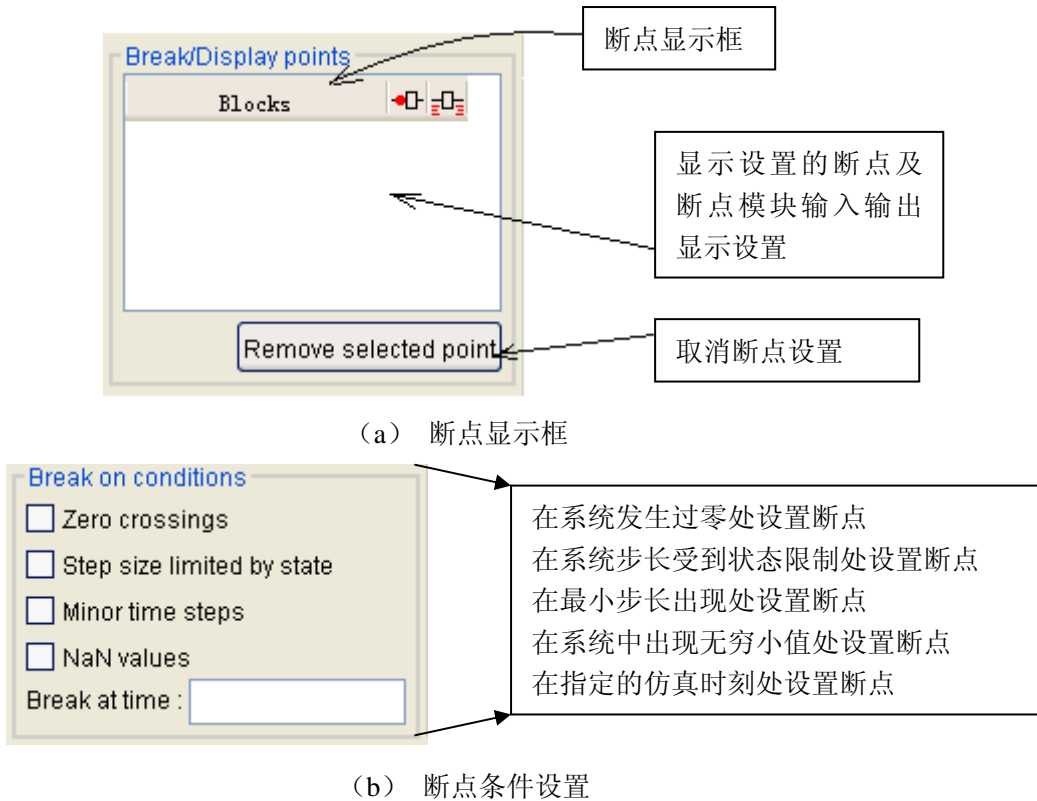


图 8.54 断点显示及断点条件设置

三、仿真回路窗

选择 Simulation Loop 选项卡，即可打开仿真回路窗，见图 8.55。

Simulink 调试器的仿真回路窗包含三列：方法列、断点列和方法标识列。

1、方法列

方法列中显示仿真执行到当前所调用的各种方法，是按含可扩展、可压缩的结点给出的一种方法树。方法树上的每个结点表示需调用其他方法的方法。当仿真结束时，调试器显示仿真终止时执行的方法及直接或调用此方法的所有方法的名称。

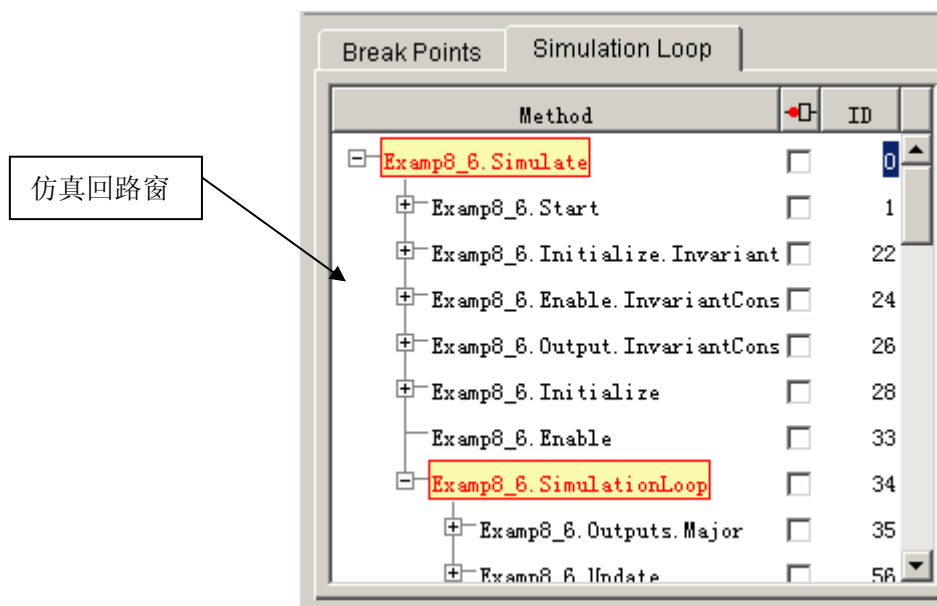


图 8.55 Simulink 调试器仿真回路窗

2、断点列

断点列由一系列选项框组成。选择选项框意味着在左边所示的方法中设置一个断点。

3、方法标识列

方法标识列列出方法列中所列各方法的标识符。有些 Simulink 命令使用方法表示来指代方法的。方法表示是该方法在仿真过程中第一次激活时，Simulink 给其指定的一个整数标号。

四、调试器输出窗口

在对指定系统模型进行调试时，调试结果都可在 Simulink 调试器的输出窗口显示。图 8.56 是 Simulink 调试器输出窗口，输出调试结果，包括调试时刻、调试的模块及模块的输入和输出。

五、调试器类型窗

选择调试器中的 Sorted list 选项卡，即可打开调试器类型窗。调试器类型窗显示模型根系统及其子系统模块类型及其执行顺序。

六、状态窗

选择调试器中的 Status 选项卡，即可打开调试器状态窗。调试器状态窗显示调试器中各种选项的值及其他一些状态信息。输出调试状态，包括当前仿真时间、缺省调试命令（执行至下一模块或执行至下一时刻）、调试断点设置及断点数等状态信息。

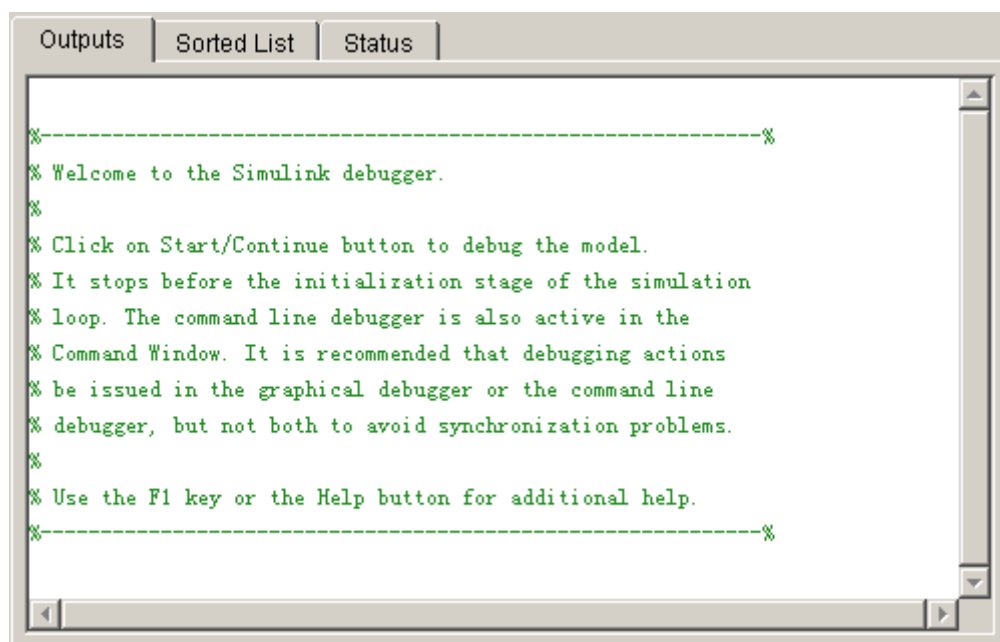


图 8.56 Simulink 调试器输出窗口

8.6.3 系统调试举例

下面仍以例 8.2 为例说明 Simulink 的调试技术，例 8.2 系统仿真模型见图 8.28。

一、设置系统调试断点

用户必须先启动 Simulink 的调试器，才可设置断点。在积分模块（Intergrator 模块）之前设置断点：选择 Intergrator 模块，单击 Simulink 调试器工具栏中的“在指定模块之前设置断点”图标即可。此时断点显示框中将显示断点设置情况，如图 5.57 所示。

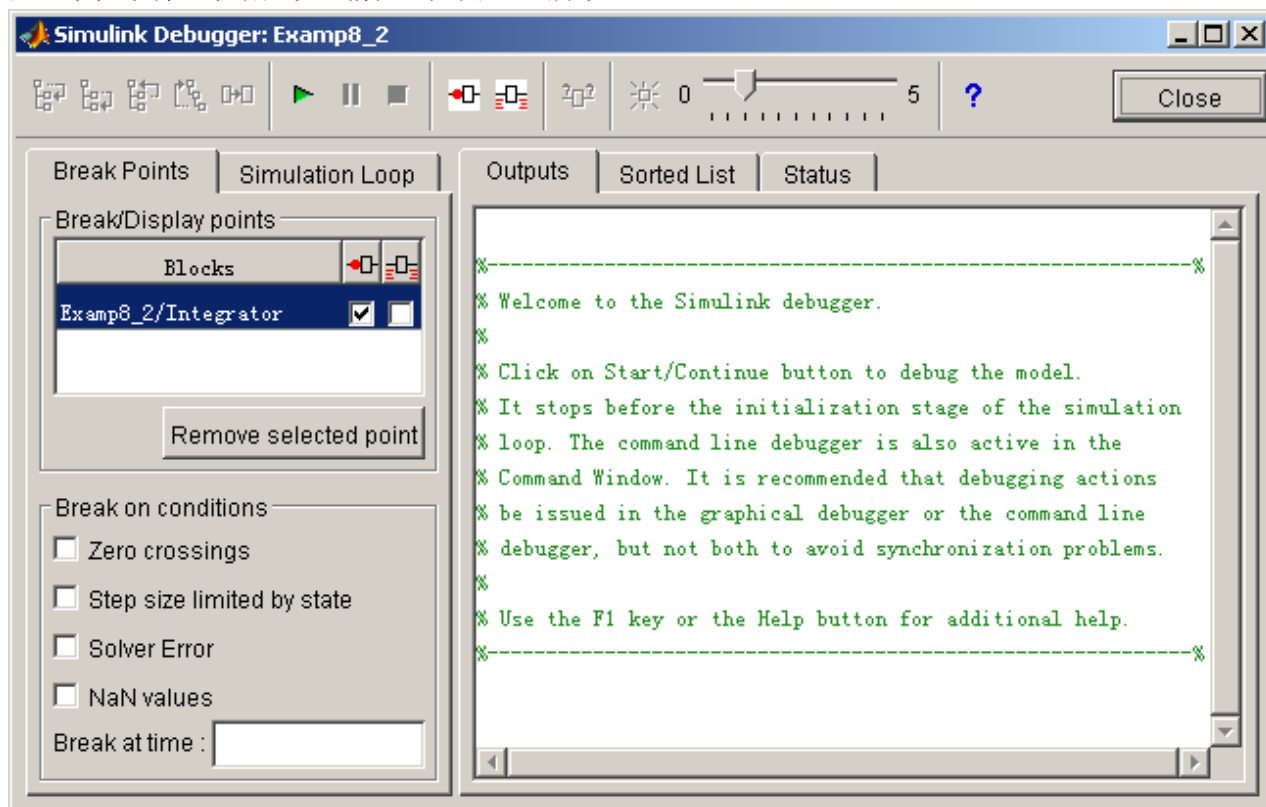




图 8.57 系统调试断点设置

如果需要，用户可以按照一定的断点设置条件设置系统调试断点。用户只需在相应的断点设置条件的复选框中选中即可。

二、调试

点击调试器工具栏中的“开始执行”按钮 ，系统模型即进入调试模式。Simulink 调试的方法有逐块调试法和逐法调试法。我们下面分别介绍。

1、逐块调试法

使用“执行下一模块”按钮 ，即可逐模块对系统进行调试。在调试过程中，用户可以在系统模型窗口中看到即将被执行的模块会用箭头指示。当此模块执行完毕，调试器的输出窗口将显示相应的系统仿真时刻、模块的输入与输出。图 8.58 显示了逐模块执行系统调试的步骤。


当系统进入调试模式，模块 Integrator 首先被执行（在系统模型窗口中用箭头指示），此时在调试器输出窗口中显示系统仿真时刻 $T_m=0$ ，即将被执行的模块是 Integrator 模块；单击调试器工具栏“执行下一模块”按钮后，调试器执行 Integrator 模块，此时调试器的输出窗口显示 Integrator 模块的输入输出，将执行模块为 Integrator1 模块；如此逐模块地执行、逐模块地进行系统调试。


如果系统中含有子系统，则系统调试至子系统时，调试器会自动打开相应的子系统。子系统中的 Inport

模块（In1 模块）、Output 模块（Out1 模块）和信号合并模块（Mux 模块）等“虚模块”均被调试器忽略。因为虚模块只是用来表示信号的某种操作，并不真正执行。

2、逐法调试法

Simulink 调试器允许用户从当前执行的方法处一个方法一个方法地运行仿真。使用 Simulink 调试左边的四个按钮，用户在每步调试时可以进入、跳过下一个方法或退出当前方法，还可以直接跳到仿真的开始。每步调试结束，调试器会显示仿真进行的方法以及进行到此法的结果。

点击调试器工具栏中的“开始执行”按钮 ，例 8.2 系统模型进入调试模式，点击一次 Simulink 调

试器工具栏按钮  或其他三个逐法调试按钮记做一步命令。在每步命令执行后，调试器在仿真回路窗中显示当前调用堆栈的方法，并用黄底色标记下一步要执行的方法，见图 8.59。

在系统调试过程中，Simulink 调试器按照一定的顺序对系统模块进行调试。使用调试器输出窗口的 Sorted List 窗口可以显示系统模块的执行顺序及其类型。此功能对于简单系统调试的作用不大，但对于大型复杂的多速率的动态系统来说是非常重要的。另外，如果系统中包含代数环，在 Execution Order 窗口中也会显示相关的系统模块。

此外，用户还可以在调试过程中随时对调试器所处的状态进行观察，此时需要使用调试器输出窗口中的 Status 窗口。图 8.60 是例 8.2 系统采用逐模块法调试中时模块执行顺序和调试时的状态。

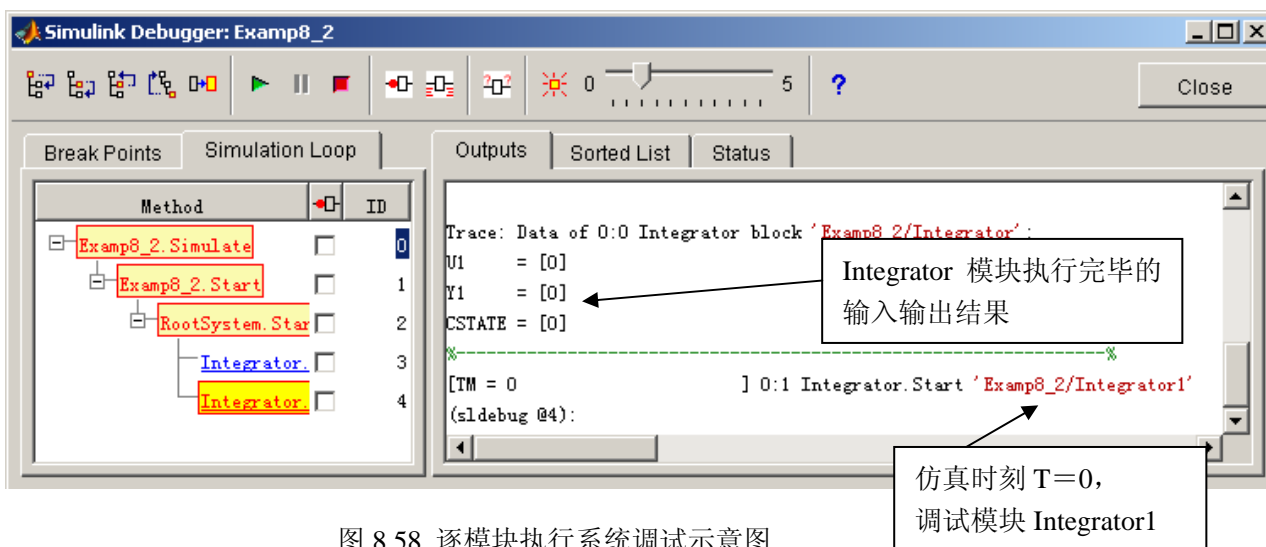
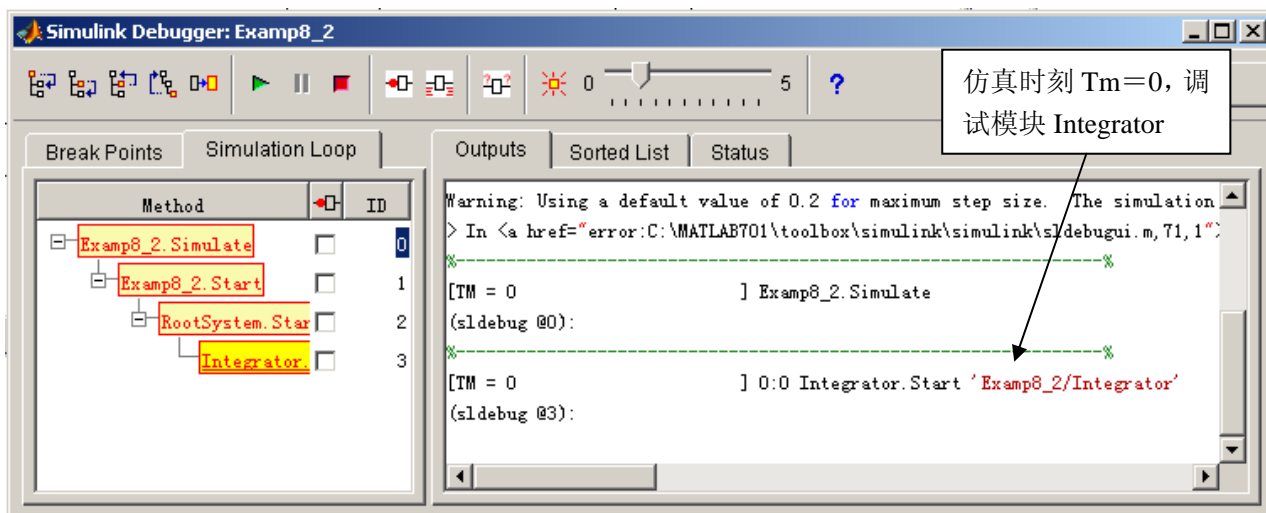
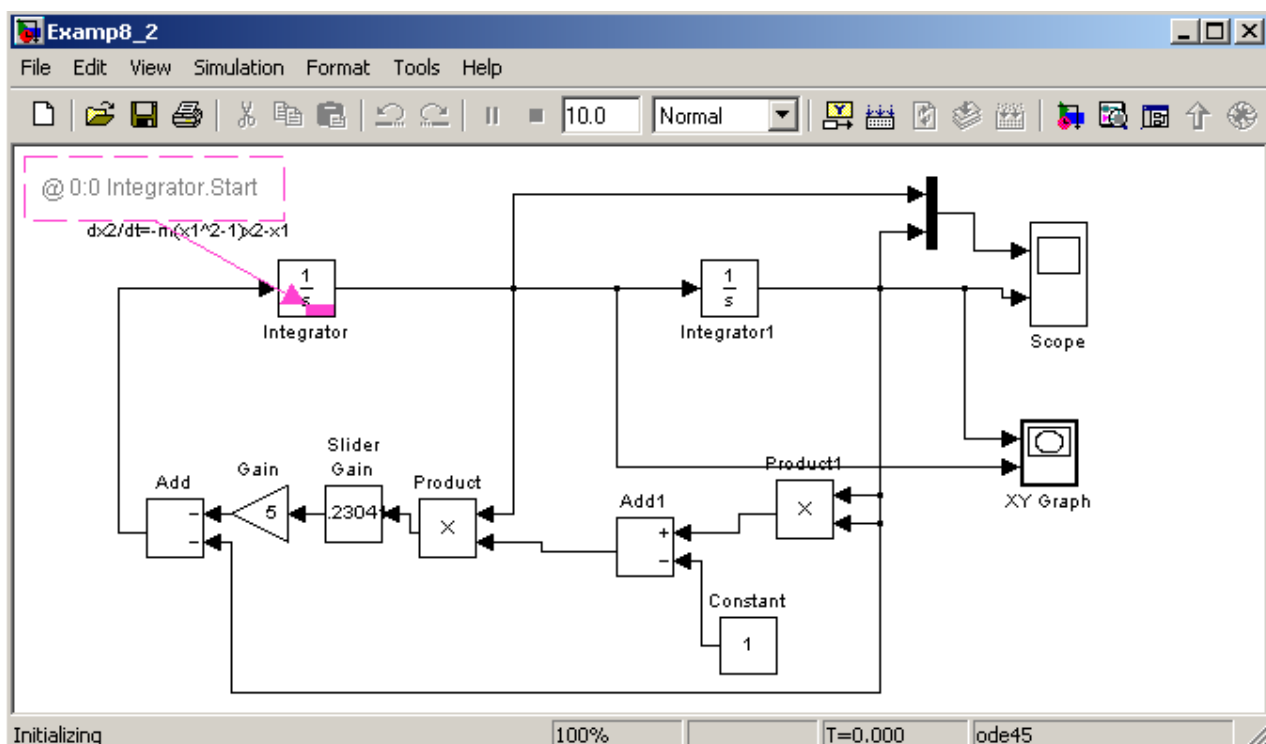


图 8.58 逐模块执行系统调试示意图

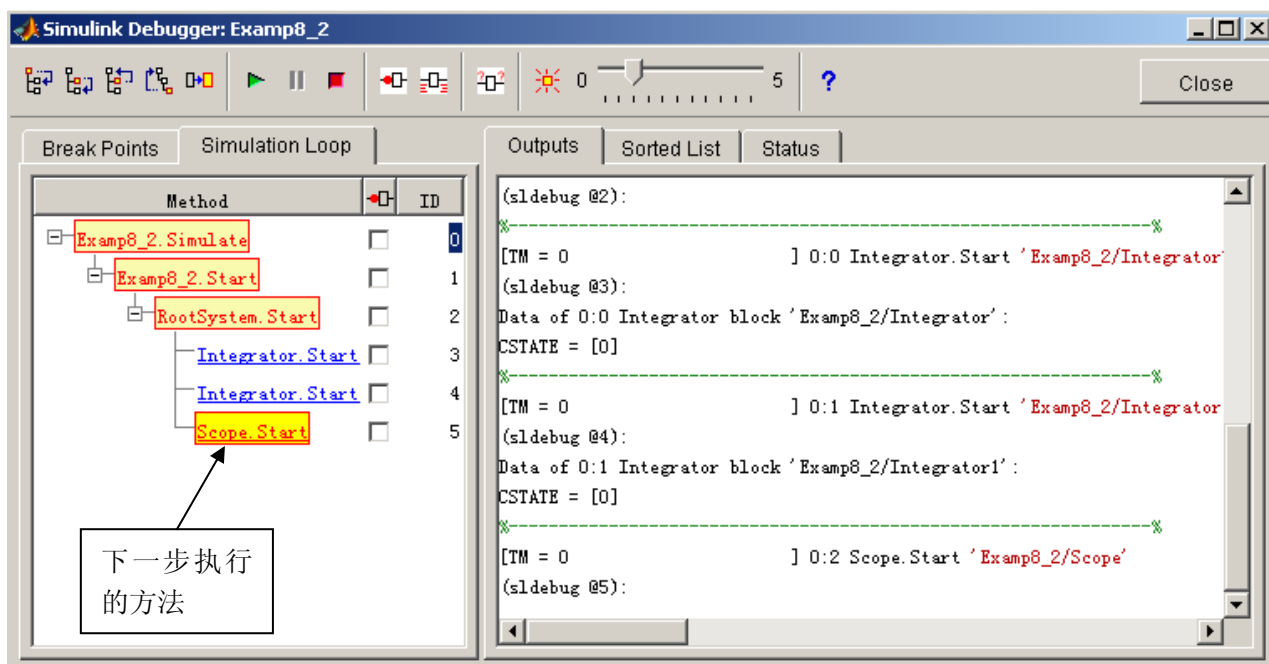


图 8.59 逐法调试法进行系统调试示意图

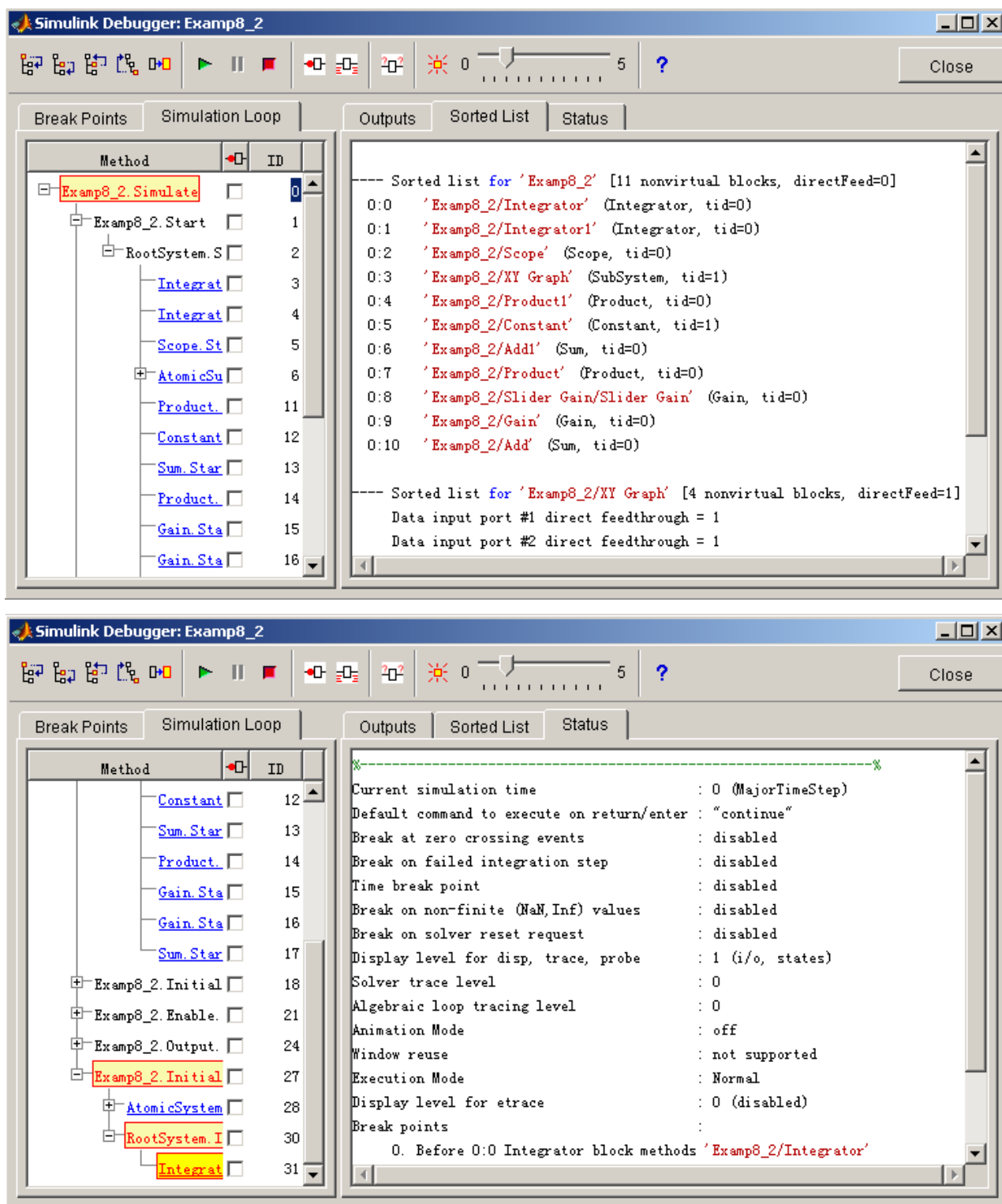


图 8.60 系统调试中模块的执行顺序和调试状态

习 题

8.1 图 8.61 示为弹簧—质量—阻尼器系统。系统的动态方程为

$$m \frac{d^2 x(t)}{dt^2} + f \frac{dx(t)}{dt} + kx(t) = F(t)$$

式中, $x(t)$ 为质量块的位移, 是质量块质量 $m = 5\text{kg}$, 阻尼器的阻尼系数 $f = 0.5$, 弹簧的弹性系数 $k = 5$; 并且质量块的初始位移和初始速度均为 0。求当受到阶跃外力或扰动外力作用下, 系统的响应 (即质量块的位移 $x(t)$), 并用示波器显示质量块位移随时间的变化。

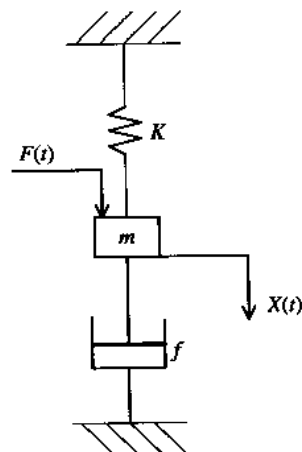


图 8.61 题 8.1 图

8.2 微分代数方程

$$\begin{cases} \dot{x}_1 = -0.2x_1 + x_2x_3 + 0.3x_1x_2 \\ \dot{x}_2 = 2x_1x_2 - 5x_2x_3 - 2x_2^2 \\ x_1 + x_2 + x_3 = 1 \end{cases}$$

已知初始条件为 $x_1(0) = 0.8$, $x_2(0) = 0.1$, $x_3(0) = 0.1$ 。构造 simulink 模型, 求 $x_1(t)$, $x_2(t)$ 和 $x_3(t)$ 。

8.3 典型 PID 控制系统方块图

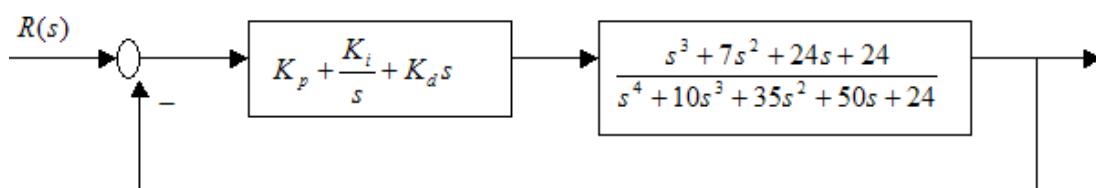


图 8.62 题 8.3 图

求 $K_p = 10, K_i = 3, K_d = 2$ 时, 系统单位阶跃信号作用下的响应, 并分析这些参数对系统性能的影响趋势。

8.4 控制系统中, 时常需要对信号计算其时域指标, 如 ITAE 指标, 定义为 $f(e) = \int_0^t \tau |e(\tau)| d\tau$; ISE 指标, 定义为 $f(e) = \int_0^t e^2(\tau) d\tau$, 其中, $e(\tau)$ 是系统的误差信号。建立 simulink 模型, 计算下列系统的 ITAE、ISE 指标。

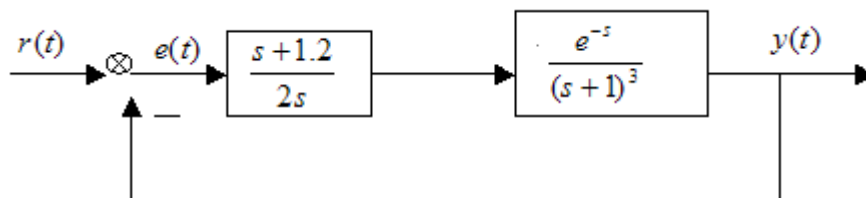


图 8.63 题 8.4 图

8.5 图 8.64 所示是简单的飞行控制系统, 试建立此动态系统的 Simulink 模型, 并进行简单的仿真分析。

其中， $G(s) = \frac{25}{s(s+0.8)}$ ，系统输入

为单位阶跃函数， $K_p=2$ ， $K_f=1$ 。

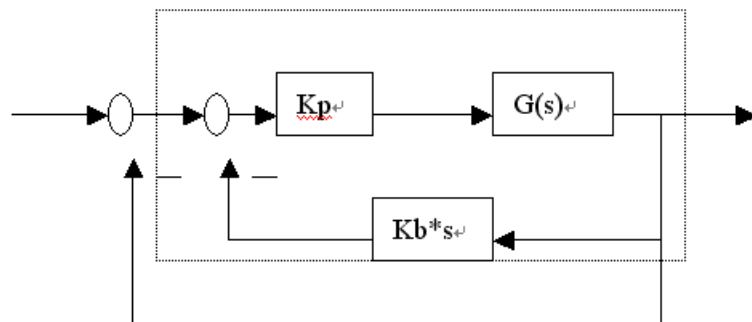


图 8.64 题 8.5 图

具体要求如下

- (1) 采用自顶向下的设计思路；
- (2) 对虚线框中的控制器采用子系统技术；
- (3) 用同一示波器显示输入和输出信号；
- (4) 输出数据到 MATLAB 工作空间，并绘制图形。

8.6 在 matlab 命令窗口下键入 f14，即可打开 f14 飞机控制系统的 simulink 仿真模块。阅读并理解此仿真的实现功能、求解器设置等信息。

8.7 将习题 8.3 中 PID 控制器作为子系统封装起来，并在封装模块上标注文字。