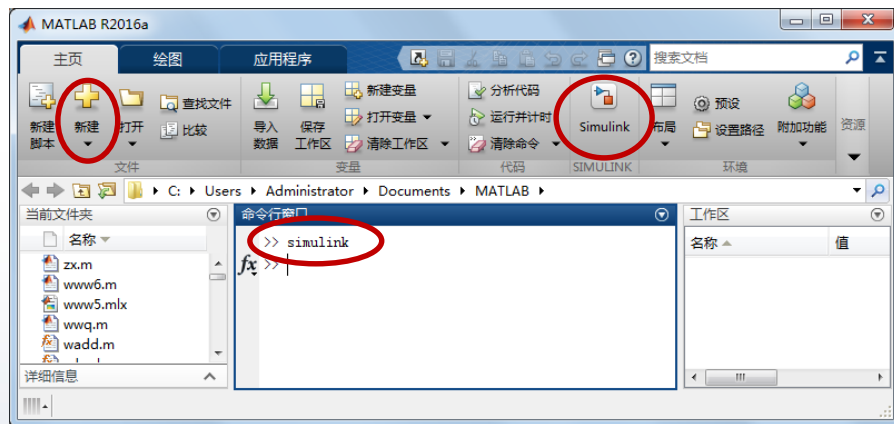


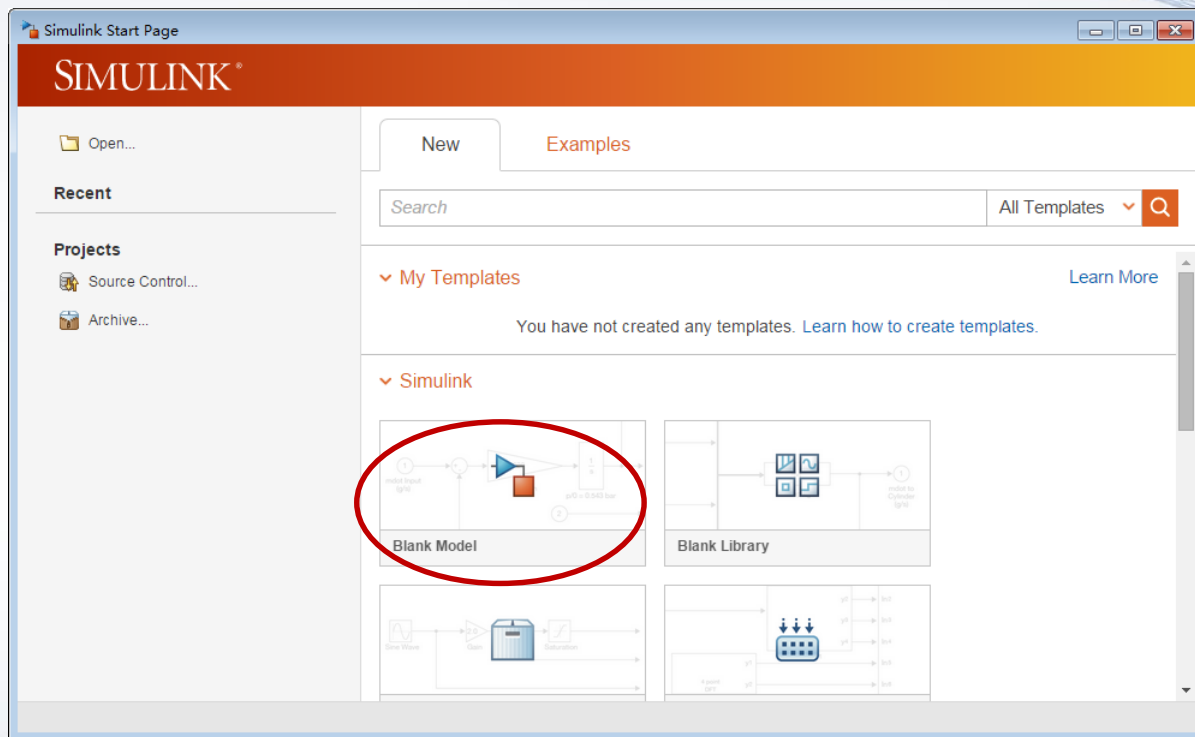
9.1 Simulink仿真基础

- Simulink的启动
- 系统仿真模型的创建
- 仿真参数的设置

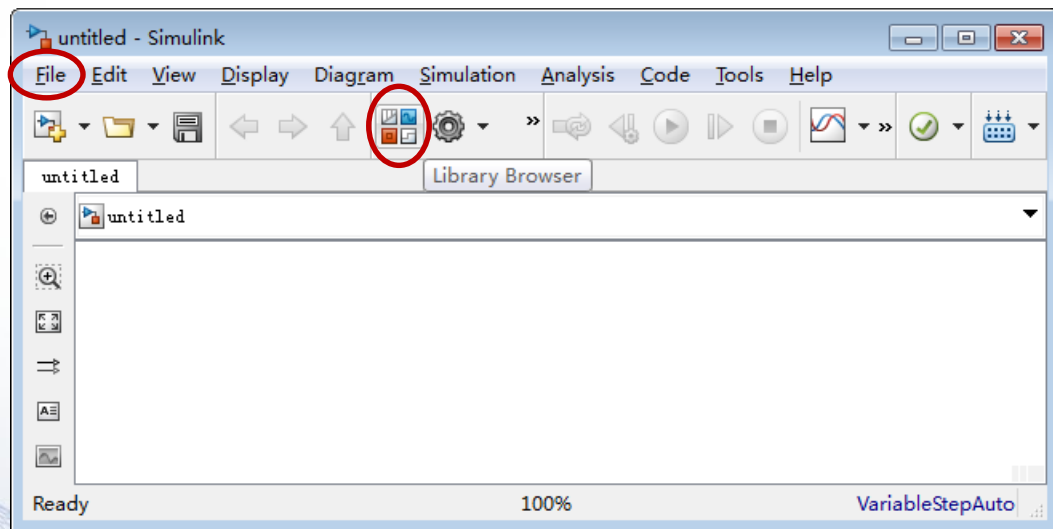
1. Simulink的启动

- ❑ 在MATLAB主窗口选择“主页”选项卡，再单击“文件”命令组中的“新建”命令按钮，然后从下拉菜单中选择“Simulink Model”命令。
- ❑ 在MATLAB主窗口选择“主页”选项卡，再单击“SIMULINK”命令组中的“Simulink”命令按钮。
- ❑ 在MATLAB的命令行窗口输入simulink命令。





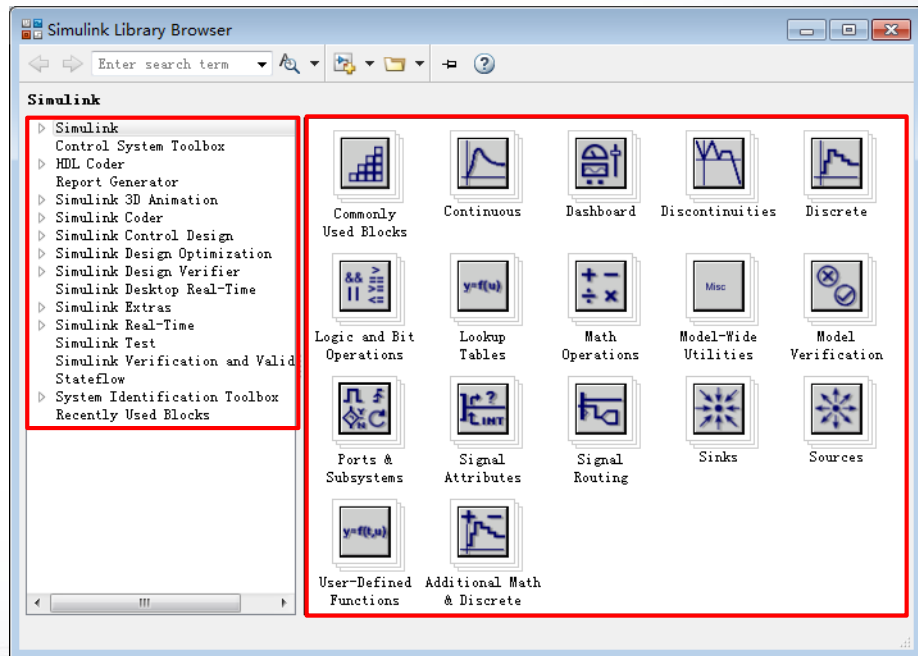
- 利用File→New命令，可以建立新的仿真模型；利用File→Open命令，可以打开已经建立好的模型文件。
- 单击“Library Browser”按钮，将打开Simulink模块库浏览器窗口，此时，可以通过鼠标将模块库中的模块拖动到模型编辑窗口，再将各个模块连接起来，就构成了仿真模型。



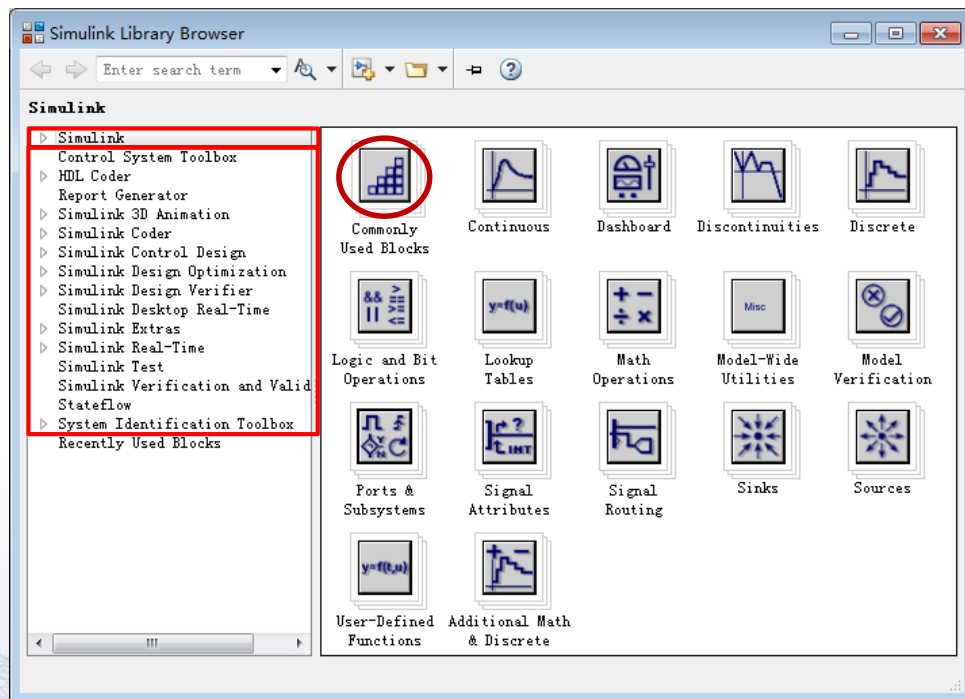
2. 系统仿真模型的创建

(1) Simulink Library Browser窗口

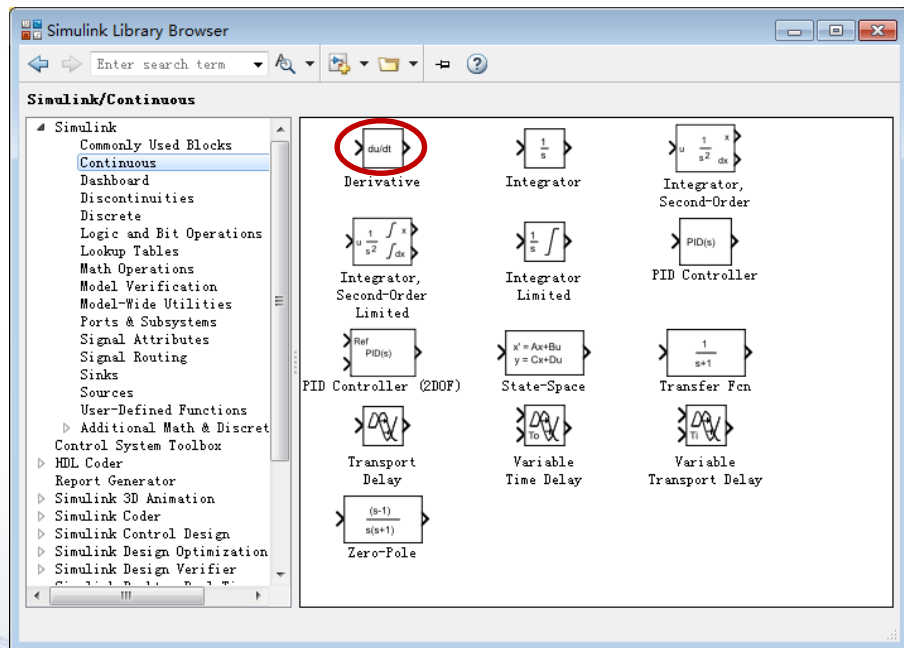
“Simulink模块库浏览器”窗口包含两个窗格，左侧的窗格以树状列表的形式列出了所有模块库。单击某个模块库，即在右侧窗格中列出该模块库的子模块库；再双击其中的子模块库图标，即列出该子模块库的所有模块。



Simulink的模块库大体分为两类，一类是基本模块库，即Simulink模块库，另一类是专业模块库，种类很多。



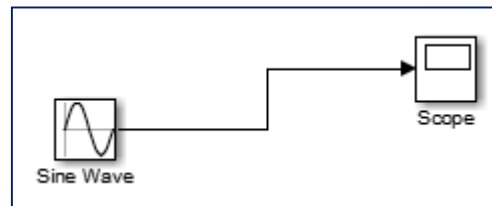
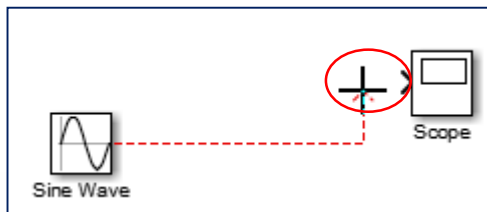
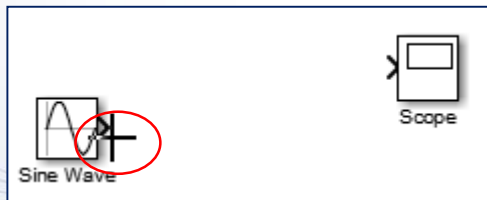
如果双击连续系统子模块库图标，模块库浏览器窗口左侧展开了Simulink基本模块库的全部子模块库，并且目前选中连续系统子模块库，右侧显示了连续系统子模块库的各个模块，可供连续系统建模使用。



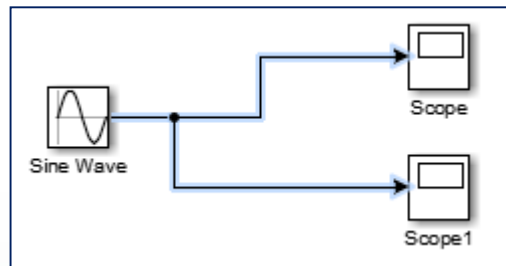
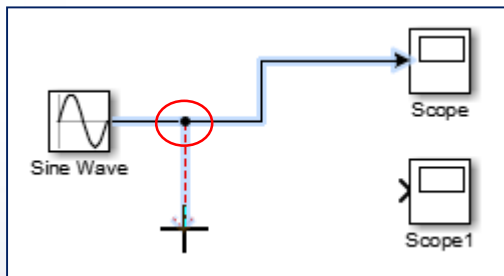
(2) Simulink模块的操作

- ❑ **模块的添加：**首先要在Simulink模块库浏览器窗口中找到该模块，然后用鼠标将这个模块拖曳到模型编辑窗口中即可。
- ❑ **模块的删除或复制：**需要先选定模块，再按删除键；或在模型编辑窗口选择Edit菜单项中的Cut、Copy、Paste等剪贴板操作命令。

两个模块的连接：先将鼠标指针移动到一个模块的输出端，当鼠标指针变成十字形光标时按住鼠标左键，移动鼠标指针到另一个模块的输入端，当连接线由虚线变成实线时，释放鼠标左键就完成了两个模块的连接。

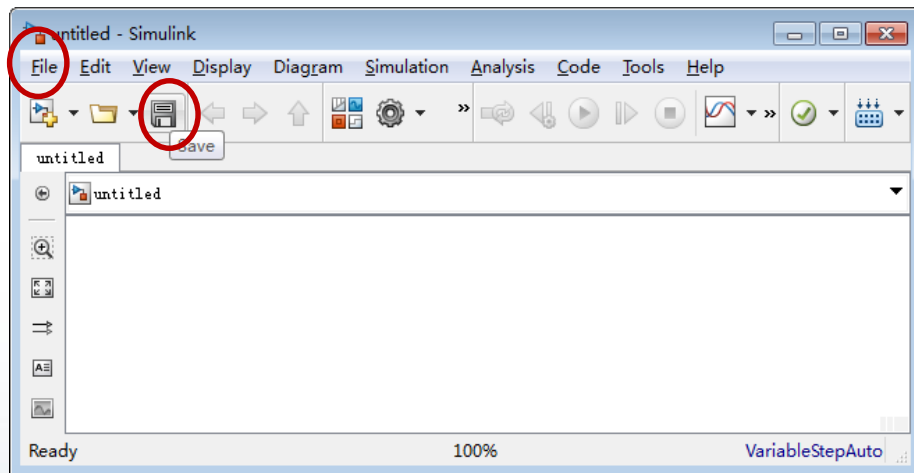


□ **连线的分支：** 在先连好一条线之后，把鼠标指针移到分支点的位置，先按下Ctrl键，然后按住鼠标拖曳到目标模块的输入端，释放鼠标和Ctrl键。



(3) 模型存盘

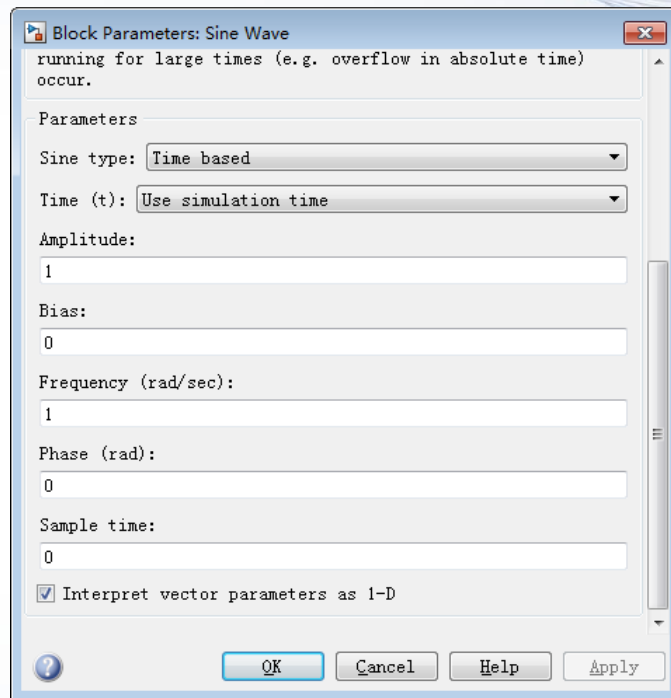
- ❑ 在Simulink模型编辑窗口选择File→Save命令或Save as命令。
- ❑ 单击模型编辑窗口工具栏中的Save命令按钮。



(4) 模块参数的设置

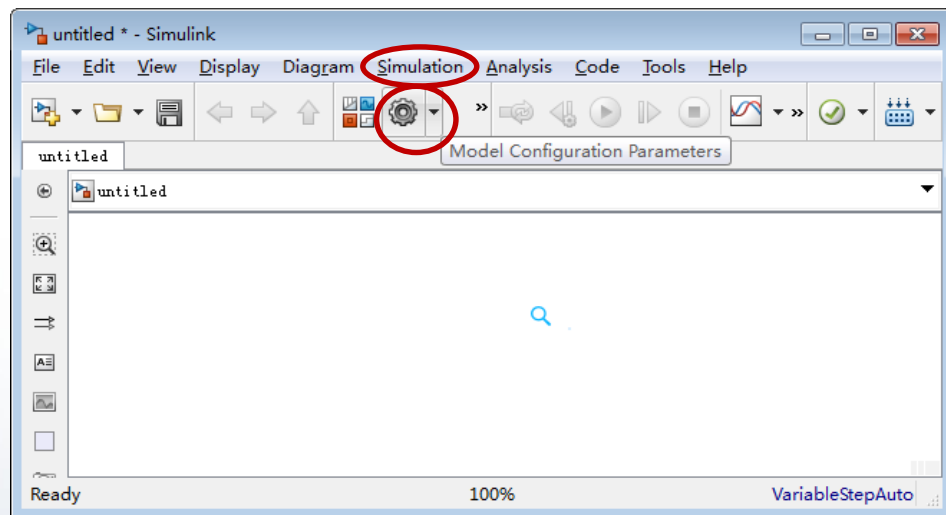
- ❑ 双击要设置的模块。
- ❑ 选择要设置的模块，再选择Diagram→Block Parameters命令。
- ❑ 右击要设置的模块，从快捷菜单中选择Block Parameters命令。

模块参数设置对话框分为两部分，上面一部分是模块功能说明，下面一部分用来进行模块参数设置。例如，正弦波模块参数对话框，用户可以设置它的幅值、偏移量、频率、相位、采样时间等参数。

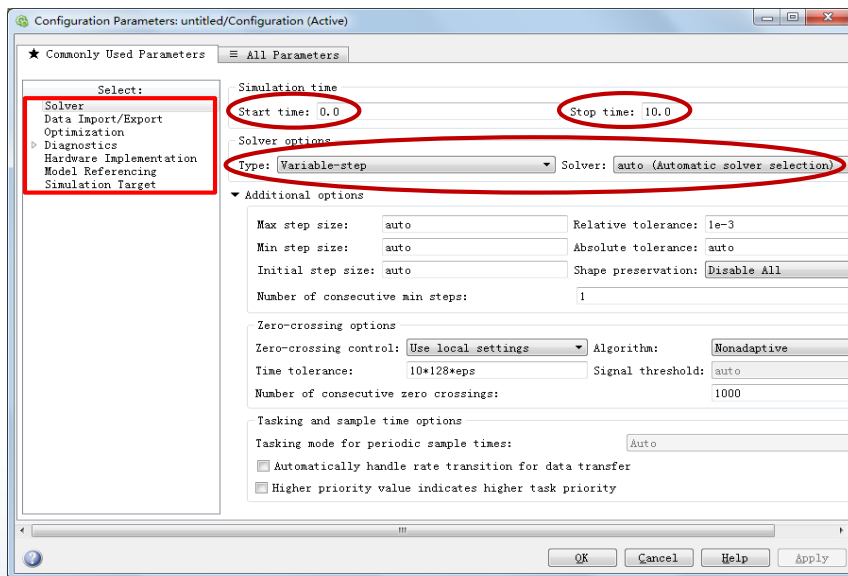


3. 仿真参数的设置

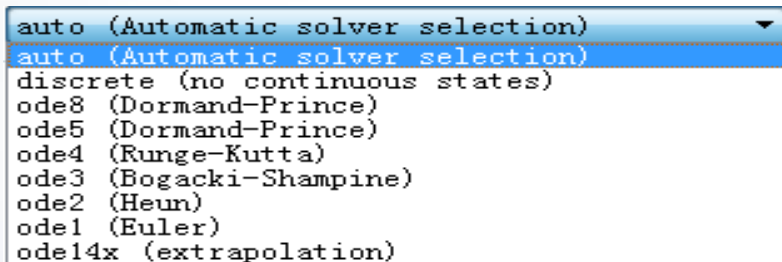
- ❑ 选择Simulation→Model Configuration Parameters命令。
- ❑ 二单击工具栏中的Model Configuration Parameters命令按钮。



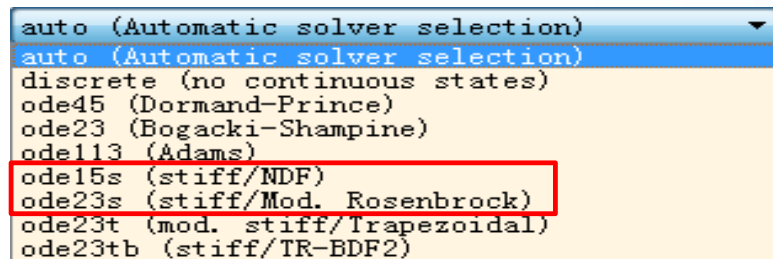
在仿真参数设置对话框中，仿真参数分为7类，Solver参数用于设置仿真起始和终止时间，选择微分方程求解算法并为其规定参数，以及选择某些输出选项。



仿真算法的选择首先设定算法类别：固定步长或变步长算法，再选择具体算法。

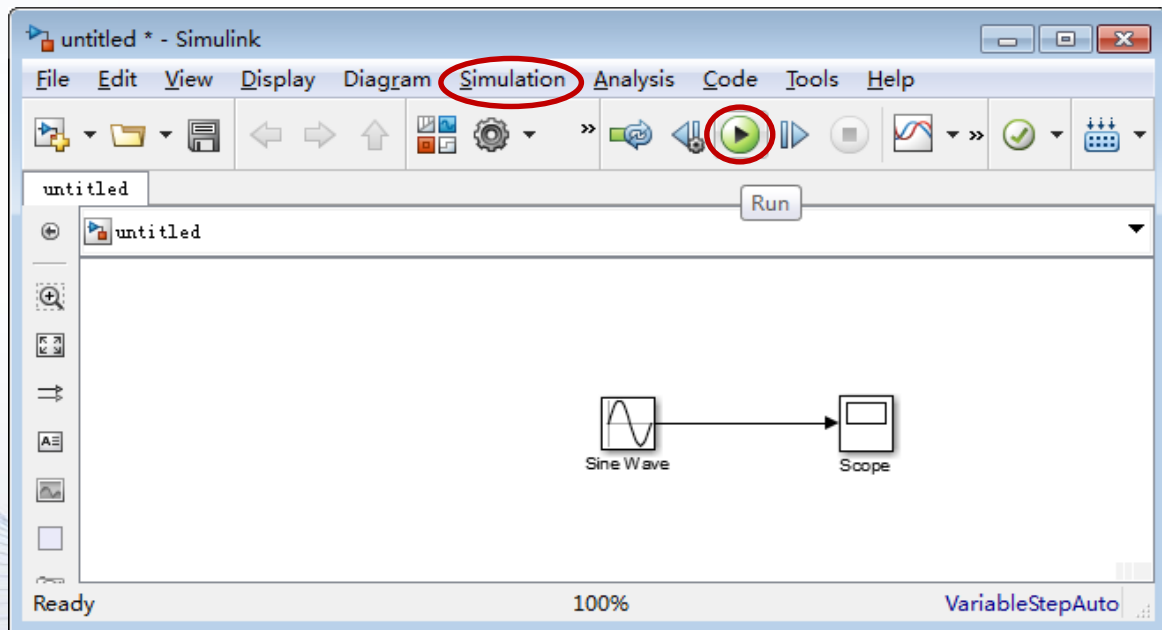


(a) 固定步长

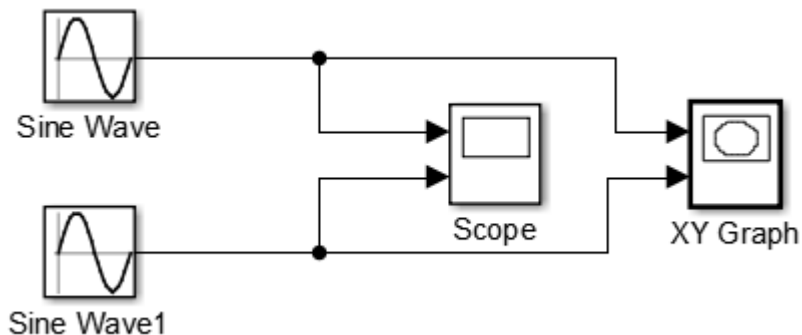


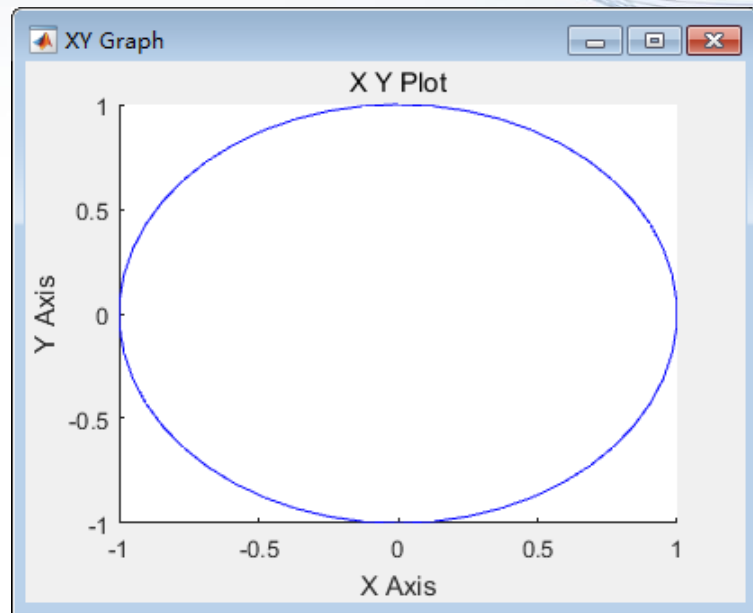
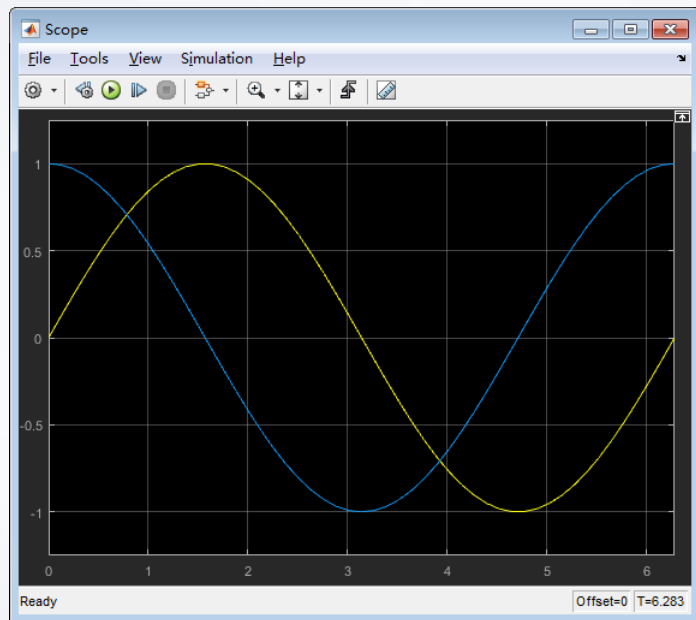
(b) 变步长

设置完仿真参数之后，单击模型编辑窗口工具栏中的Run按钮，或选择Simulation菜单项中的Run命令，便可启动对当前模型的仿真。



利用Simulink仿真，分别显示曲线 $y=\sin t$ 和 $y=\cos t$ ，同时显示 $\sin t$ 对 $\cos t$ 的变化曲线。





Simulink系统仿真的步骤:

- ❑ 建立系统仿真模型
- ❑ 设置仿真参数
- ❑ 启动仿真并分析仿真结果

9.2 子系统的创建与封装

- 子系统的创建
- 子系统的封装
- 子系统的条件执行

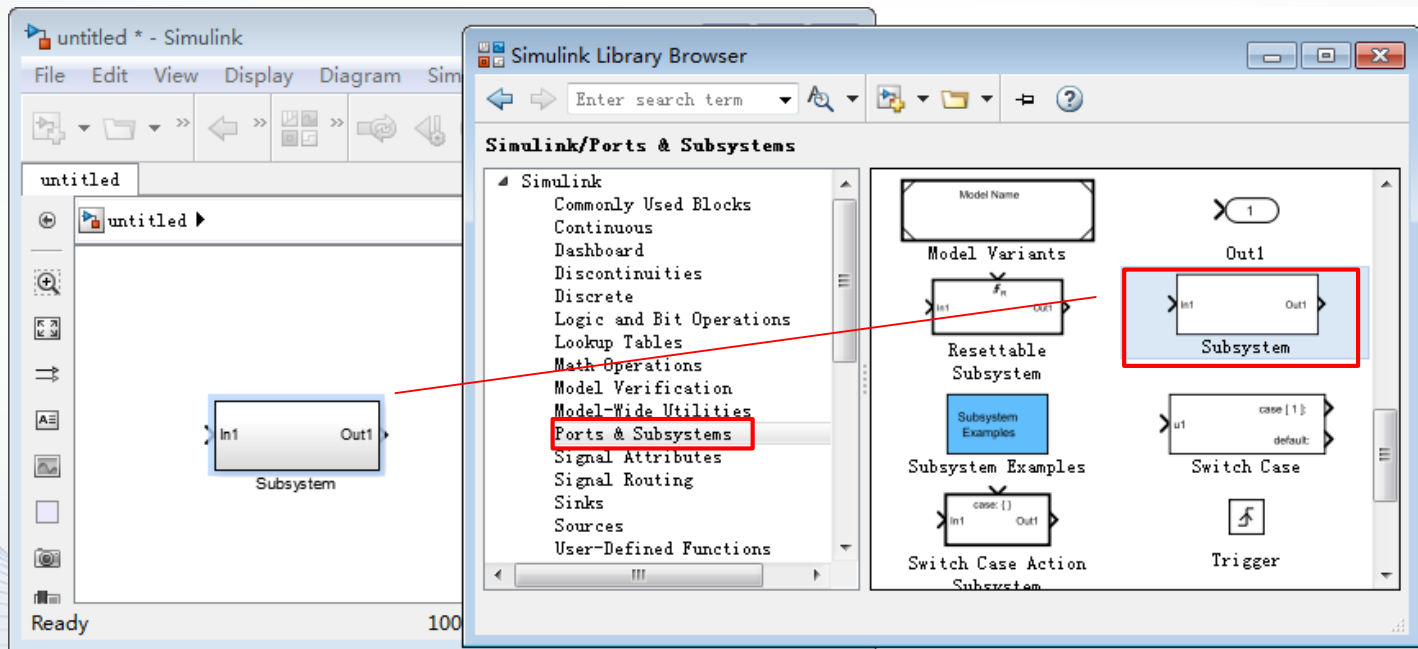
1. 子系统的创建

- 通过Subsystem模块建立子系统
- 将已有的模块转换为子系统

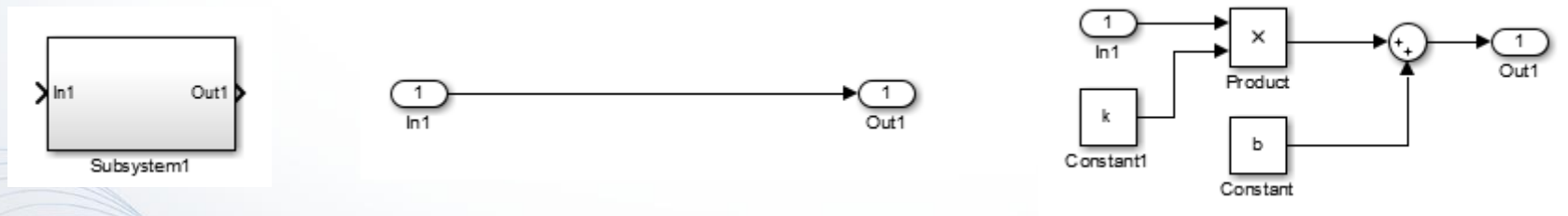
两者的区别是：前者先建立子系统，再为其添加功能模块；后者先选择模块，再建立子系统。

下面以创建 $y=kx+b$ 子系统为例，说明具体操作。

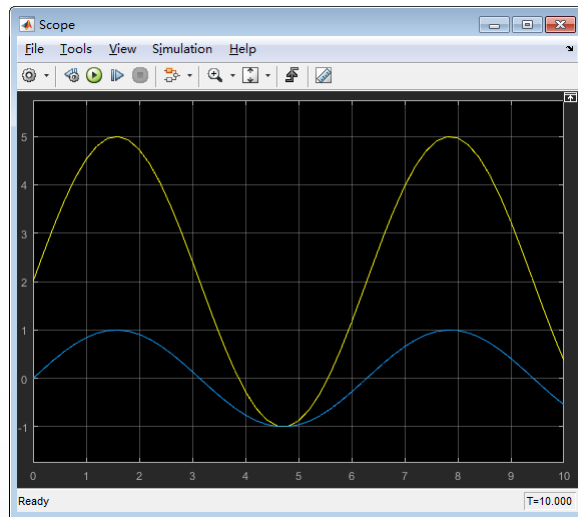
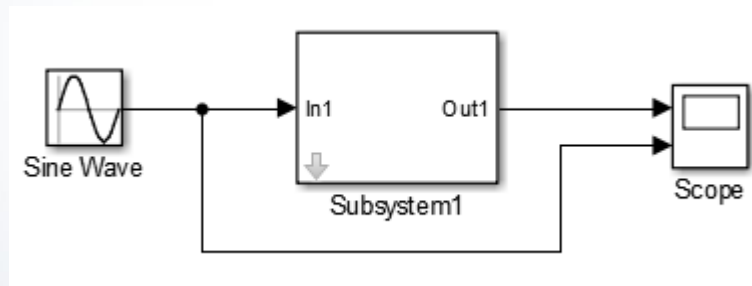
(1) 通过Subsystem模块建立子系统



双击子系统模块打开子系统编辑窗口，窗口中已经自动添加了相互连接的一个输入模块和输出模块，表示子系统的输入端口和输出端口。将要组合的模块插入到输入模块和输出模块中间，这里需要k和b两个常数模块，一个乘法模块和一个加法模块，将这些模块重新连接起来，一个子系统就建好了。

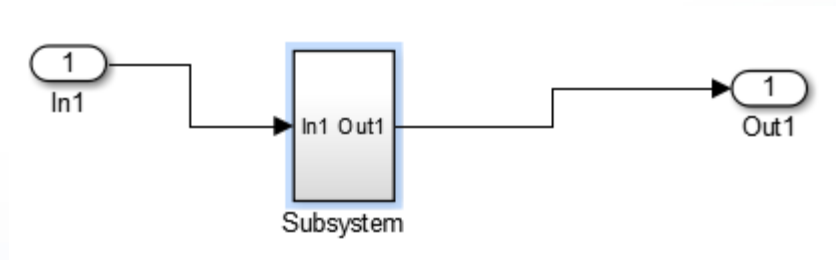
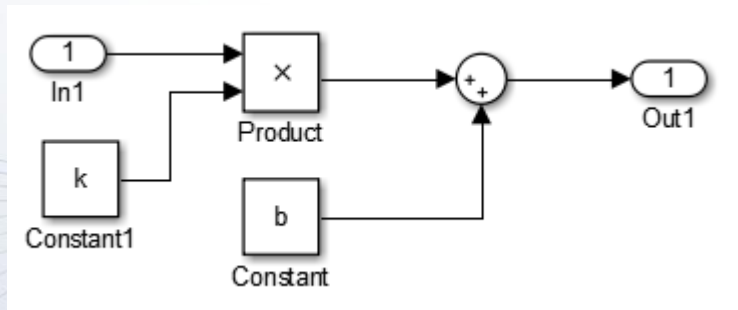


启动仿真后查看示波器的曲线，其中蓝色曲线是正弦曲线，黄色曲线是经过 $y=kx+b$ 变换之后的曲线，这里在MATLAB命令行窗口事先给k赋3，b赋2，即黄色曲线代表 $y=3\sin t+2$ 。



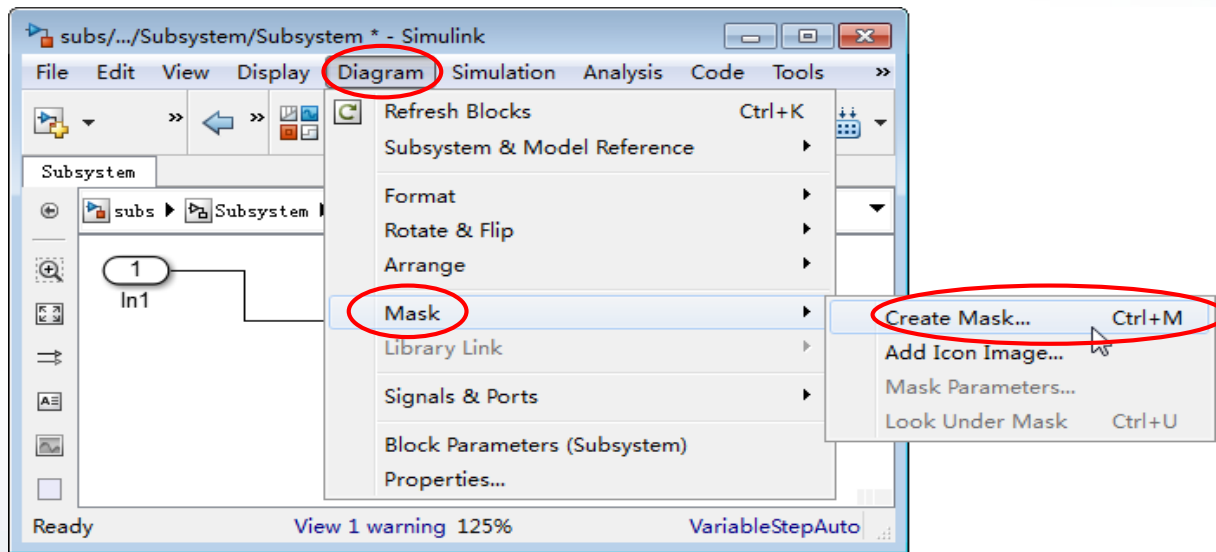
(2) 通过已有的模块建立子系统

建立的系统模型如图所示，选中系统模型中的所有模块，在模型编辑窗口选择Diagram菜单项中的相关命令（Diagram → Subsystem & Model Reference → Create Subsystem from Selection命令），或按Ctrl+G组合键建立子系统，所选模块将被一个子系统模块取代，如果想要查看子系统的内部结构又可以双击子系统模块。

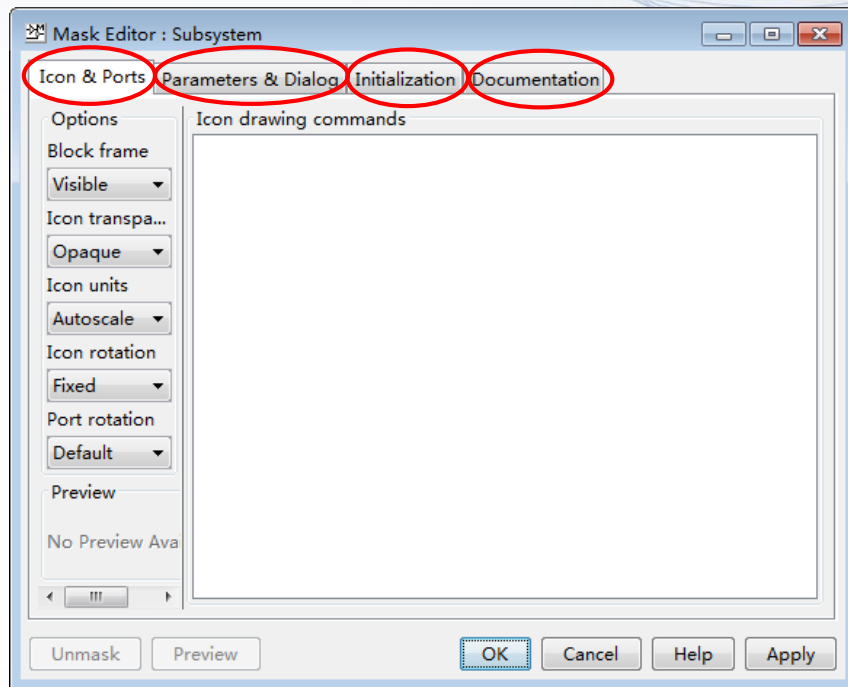


2. 子系统的封装

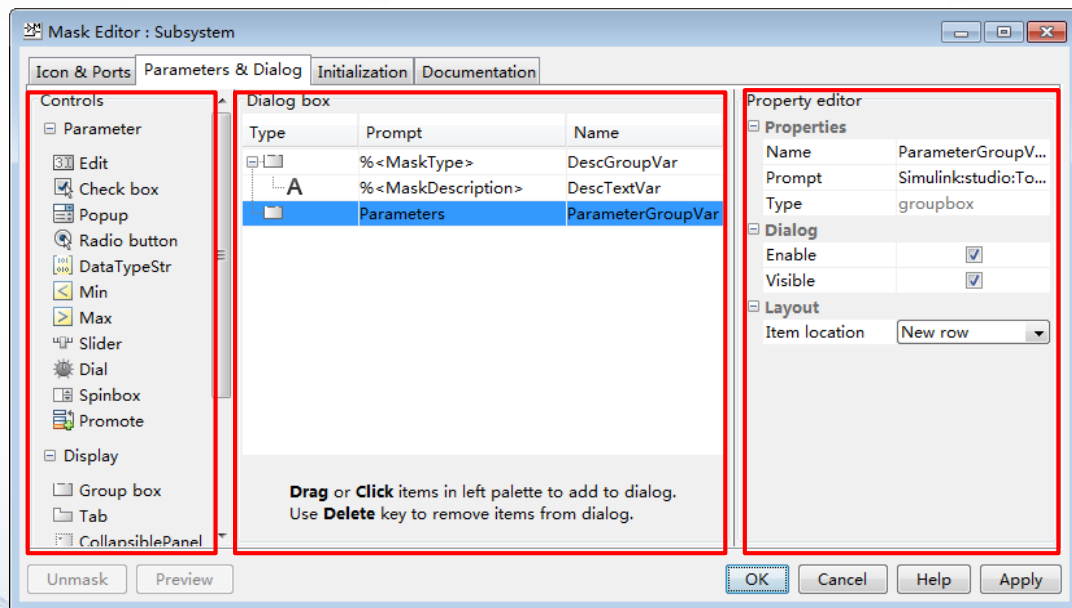
先选中所要封装的子系统，再在模型编辑窗口选择Diagram菜单项中的Create Mask命令，或按Ctrl+M组合键，这时将出现封装编辑器（Mask Editor）对话框。



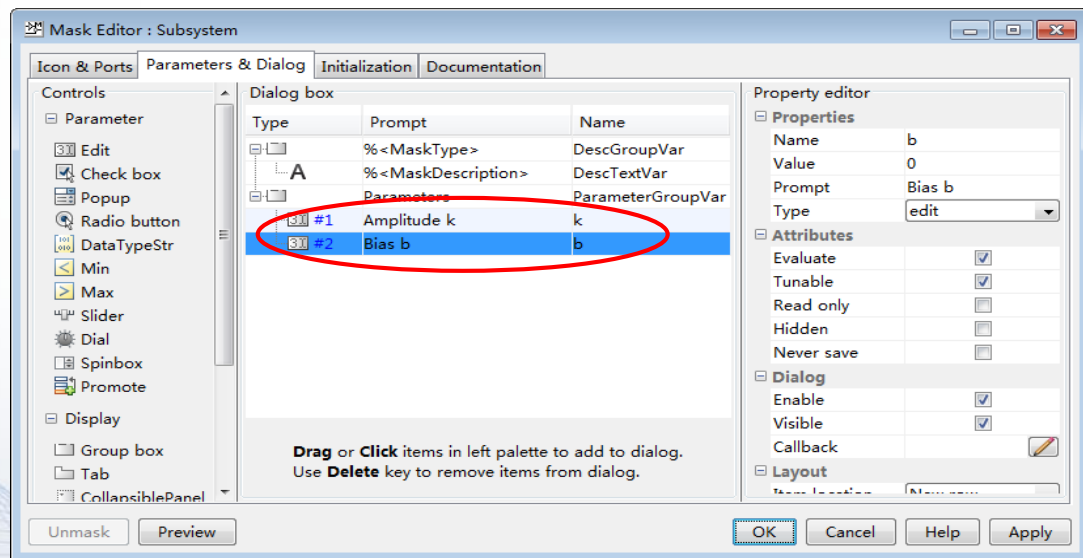
- ❑ 第一个选项卡用于设置被封装模块的图标。
- ❑ 第二个选项卡用来设置子系统参数设置对话框。
- ❑ 第三个选项卡用于设置初始化命令。
- ❑ 第四个选项卡用于定义封装模块的类型、描述和帮助文本。



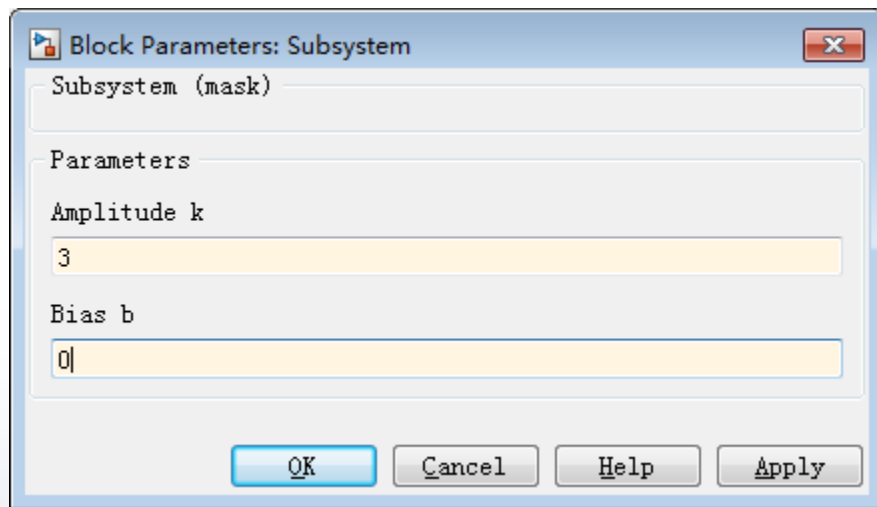
第二个选项卡由3部分组成：左侧为控件工具箱（Controls），中间区域显示对话框中的控件，右侧区域用于显示和修改控件的属性。



下面以 $y=kx+b$ 子系统为例，说明子系统参数设置对话框的设置方法。在第二个选项卡的左侧控件工具箱中，连续2次单击Edit按钮，为子系统的2个变量准备输入位置。在中间区域分别输入该控件的提示信息 and 控件名，最后单击OK按钮确认设置。



子系统参数设置对话框设置完成后，双击子系统图标将出现其参数对话框。例如，双击仿真模型中的 $y=kx+b$ 子系统图标，则弹出如图所示的参数对话框，允许用户输入参数 k 和 b 的值。



受控制信号控制的子系统称为条件执行子系统。在条件执行子系统中，输出信号取决于输入信号和控制信号。

条件执行子系统：

- 使能子系统
- 触发子系统
- 使能加触发子系统

3. 子系统的条件执行

(1) 使能子系统

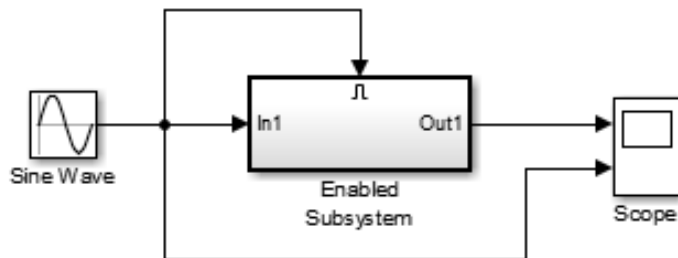
使能子系统表示控制信号由负变正时子系统开始执行，直到控制信号再次变为负时结束。控制信号可以是标量也可以是向量。如果控制信号是标量，则当标量的值大于0时子系统开始执行。如果控制信号是向量，则向量中任何一个元素大于0，子系统将执行。

使能子系统外观上有一个“使能”控制信号输入端口。“使能”是指当且仅当“使能”输入信号为正时，该模块才接收输入端的信号。可直接选择使能子系统模块来建立使能子系统，双击使能子系统模块，打开其内部结构窗口。

也可以展开已有子系统，添加端口与子系统（Ports & Subsystems）模块库中的使能（Enable）模块，将该子系统转换为使能子系统。

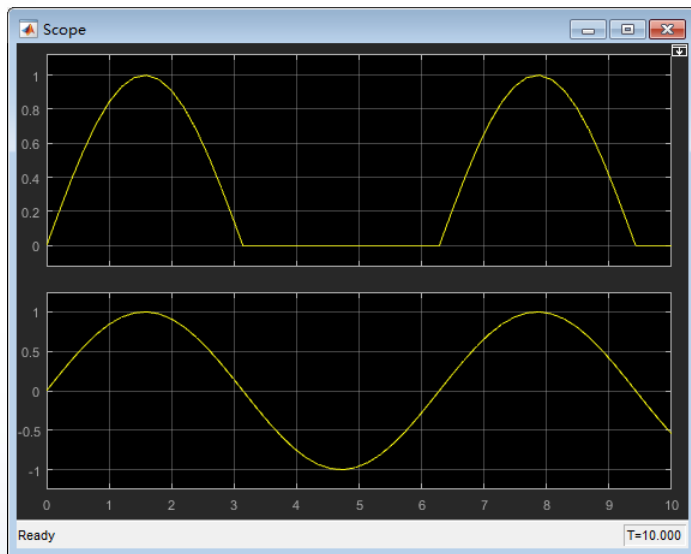


例1 利用使能子系统构成一个正弦半波整流器。



新建一个仿真模型，将正弦波（Sine Wave）、使能子系统（Enabled Subsystem）和示波器（Scope）三个模块拖至新打开的模型编辑窗口，连接各模块、设置参数并存盘，创建一个使能子系统。其中使能信号端接正弦模块。在示波器窗口设置输入端口数（Number of input ports）为2，并设置输出布局（Layout）。

在模型编辑窗口单击工具栏的Run命令按钮，就可看到半波整流波形和正弦波形。这里正弦信号当控制信号，当控制信号为负值时子系统停止执行，输出是零。

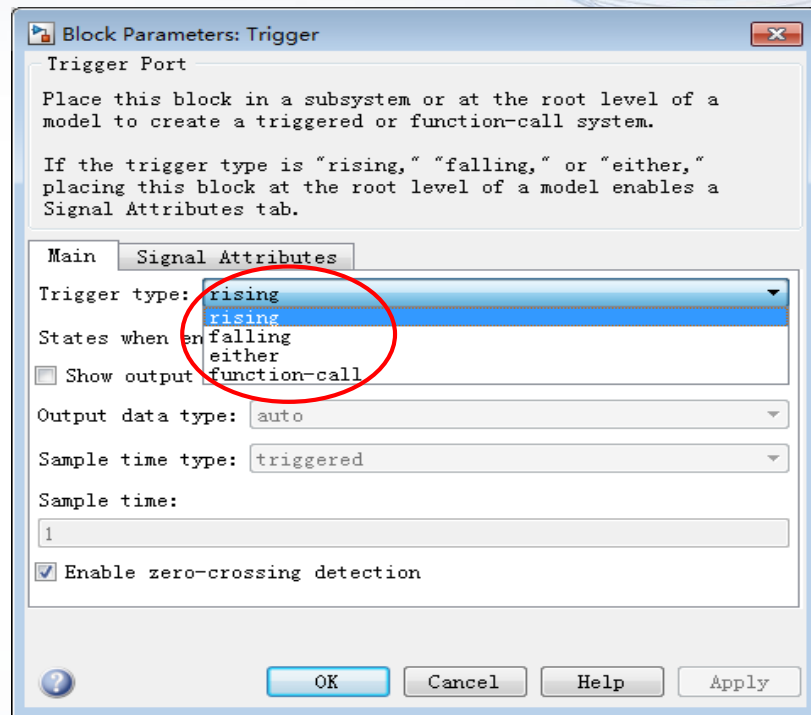


(2) 触发子系统

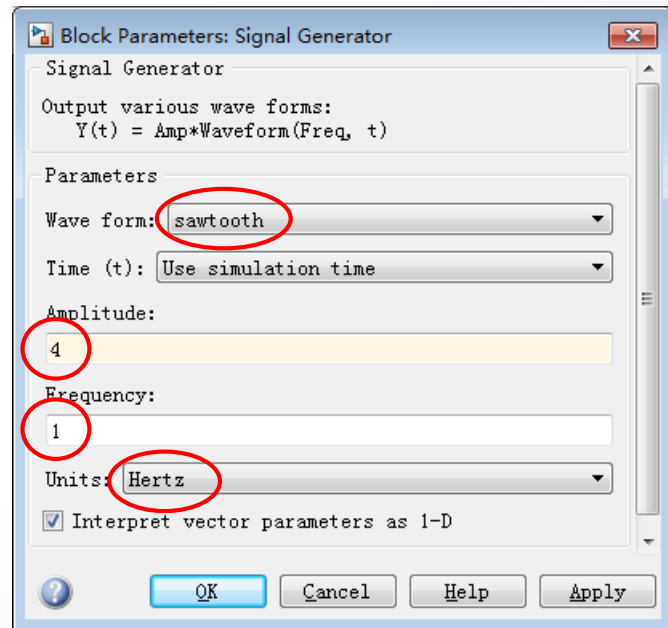
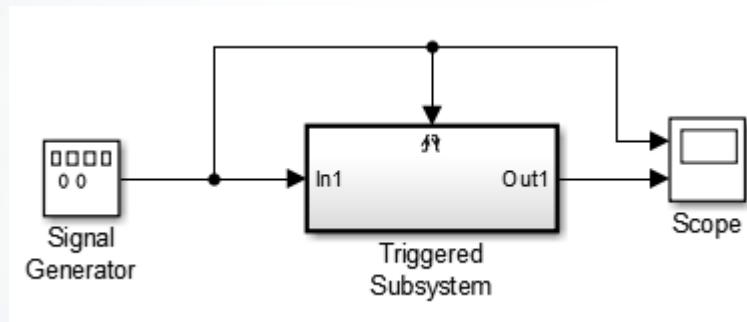
触发子系统是指当触发事件发生时开始执行子系统。

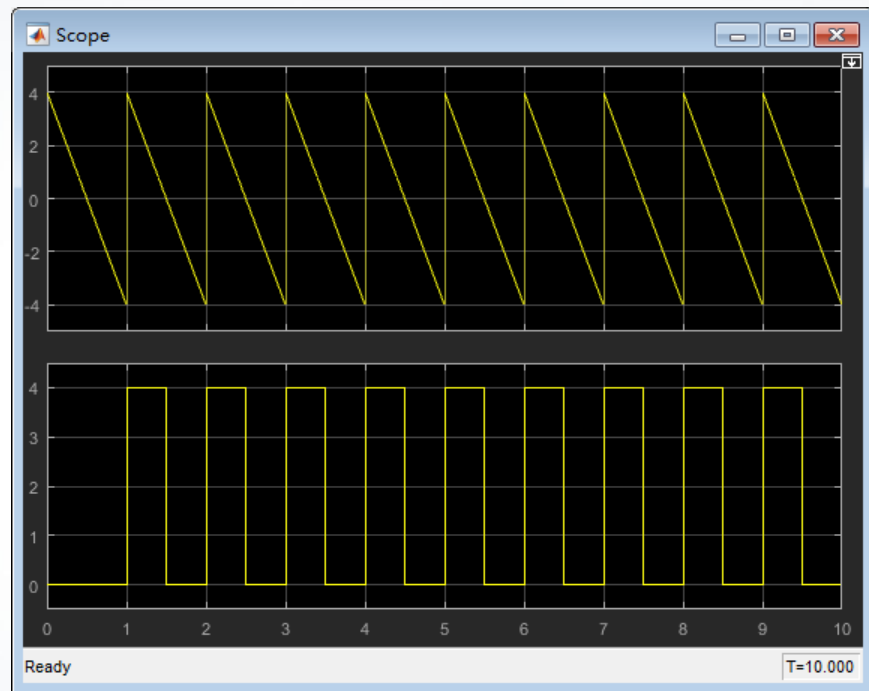
与使能子系统相类似，触发子系统的建立可直接选择Triggered Subsystem模块，或者展开已有子系统，添加Ports & Subsystems模块库中的Trigger模块，将该子系统转换为触发子系统。

- ❑ rising（上跳沿触发）：控制信号从负值或0上升到正值时子系统开始执行。
- ❑ falling（下跳沿触发）：控制信号从正值或0下降到负值时子系统开始执行。
- ❑ either（上跳沿或下跳沿触发）：当控制信号满足上跳沿或下跳沿触发条件时，子系统开始执行。
- ❑ function-call（函数调用触发）：这种触发方式必须与S函数配合使用。



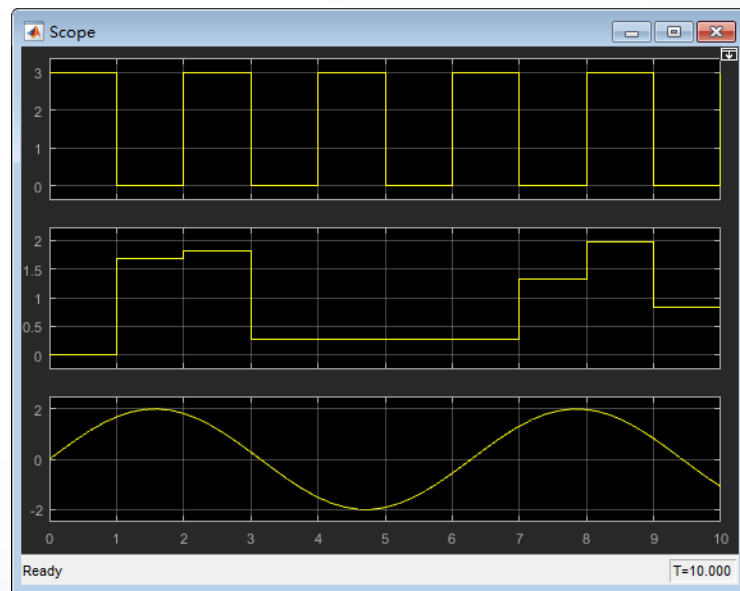
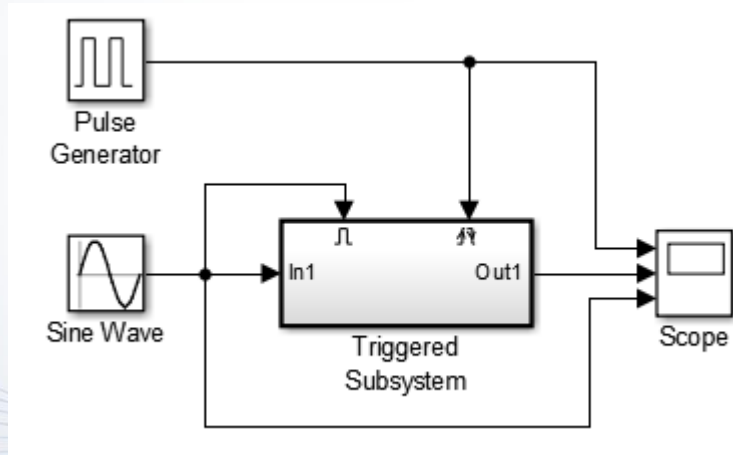
例2 利用触发子系统将一锯齿波转换成方波。





(3) 使能加触发子系统

所谓使能加触发子系统就是当使能控制信号和触发控制信号共同作用时执行子系统。



9.3 S函数的设计与应用

- 什么叫S函数
- 用M文件编写S函数的方法
- 在仿真模型中使用S函数的方法

1. 什么叫S函数

- ❑ S函数是系统函数 (System Function) 的简称, 是指采用一种程序设计语言描述的一个功能模块。
- ❑ 用户可以采用MATLAB语言, 也可以采用C、C++或FORTRAN等语言来编写S函数。
- ❑ S函数有自己特定的语法构成规则, 可以用来描述并实现连续系统、离散系统以及复合系统。
- ❑ S函数能够接收来自Simulink求解算法的相关信息, 并对求解算法发出的命令做出适当的响应, 这种交互作用类似于Simulink系统模块与求解算法的相互作用。

2. 用MATLAB语言编写S函数

在MATLAB命令行窗口输入命令，打开模板文件。

```
>> edit sfuntmpl.m
```

模板文件sfuntmpl.m包括：

- 1个主函数
- 6个子函数

(1) 主函数

主函数的引导语句为：

```
function [sys, x0, str, ts]=fname(t, x, u, flag)
```

- ❑ fname是S函数的函数名。
- ❑ 输入形参t、x、u、flag分别为仿真时间、状态向量、输入向量和子函数调用标志。
- ❑ 输出形参sys代表一种返回参数；x0是初始状态值；对于M文件S函数，str将被置成一个空阵；ts是一个两列矩阵。

(2) 子函数

S函数共有6个子函数， 这些子函数的前缀为mdl， 由flag的值来控制仿真各阶段调用S函数的哪一个子函数。

- ❑ flag取0： 调用初始化子函数mdlInitializeSizes。
- ❑ flag取1： 调用子函数mdlDerivatives实现连续状态的更新。
- ❑ flag取2： 调用子函数mdlUpdate实现离散状态的更新。
- ❑ Flag取3： 调用输出子函数mdlOutputs。



3. S函数的应用

采用S函数实现 $y=kx+b$ 。

(1) 定义S函数

①主函数

```
function [sys, x0, str, ts]=timekb(t, x, u, flag, k, b)
switch flag
    case 0
        [sys, x0, str, ts]=mdlInitializeSizes; %初始化
    case 3
        sys=mdlOutputs(t, x, u, k, b);          %计算输出量
    case {1, 2, 4, 9}
        sys=[];
    otherwise
        error(num2str(flag))                    %出错处理
end
```

②初始化子函数

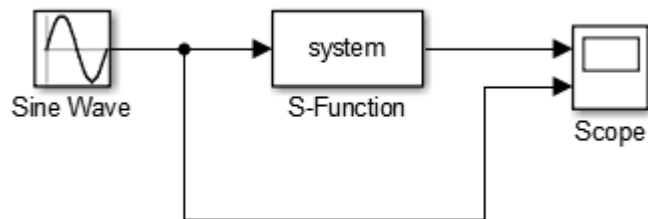
```
function [sys,x0,str,ts]=mdlInitializeSizes()  
sizes=simsizes;  
sizes.NumContStates=0;           %无连续状态  
sizes.NumDiscStates=0;          %无离散状态  
sizes.NumOutputs=1;             %有一个输出量  
sizes.NumInputs=1;              %有一个输入信号  
sizes.DirFeedthrough=1;         %输出量中含有输入量  
sizes.NumSampleTimes=1;         %单个采样周期  
sys=simsizes(sizes);  
%给其他返回参数赋值  
x0=[];                           %设置初始状态为零状态  
str=[];                           %将str变量设置为空字符串  
ts=[-1,0];                       %假定继承输入信号的采样周期
```

③输出子函数

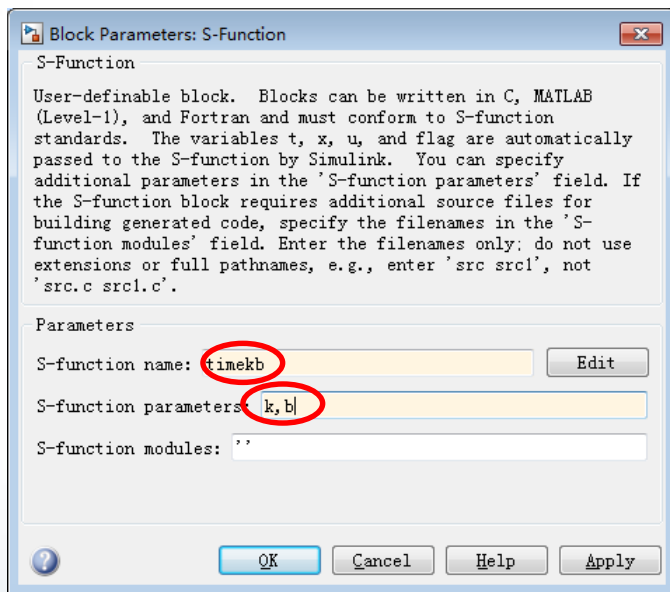
```
function sys=mdlOutputs(t,x,u,k,b)  
sys=k*u+b;
```

(2) 在Simulink模型中使用S函数

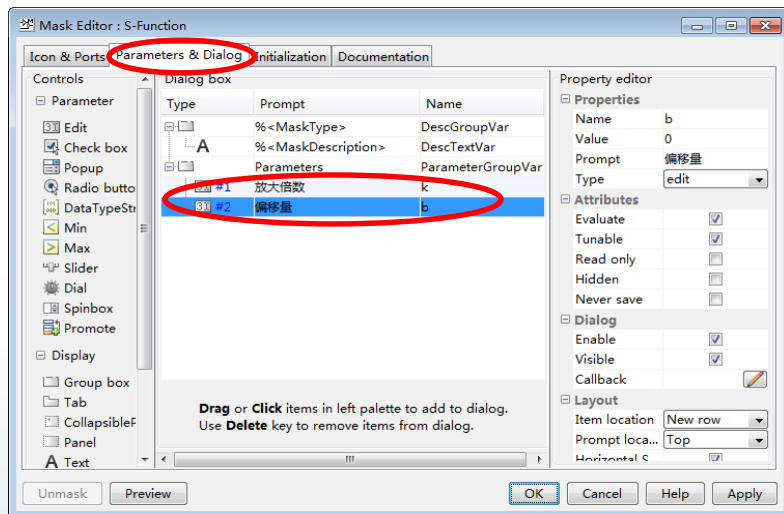
新建一个模型，向模型编辑窗口中添加S函数模块、Sine Wave模块和Scope模块。



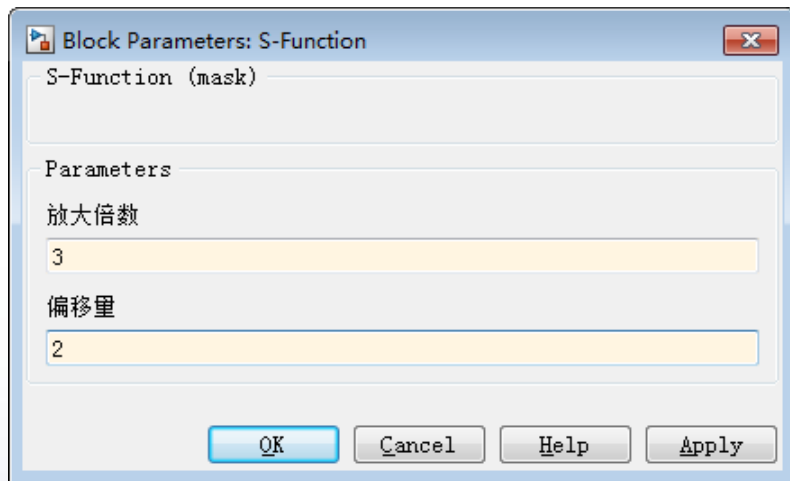
在模型编辑窗口中双击S函数模块，打开其参数对话框，填入S函数名 timesn，填入外部参数k和b。如果有多个外部参数，参数之间用逗号分隔。



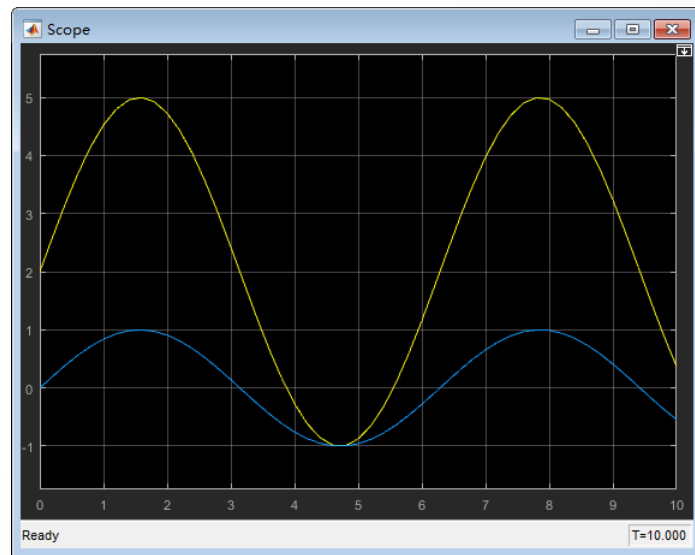
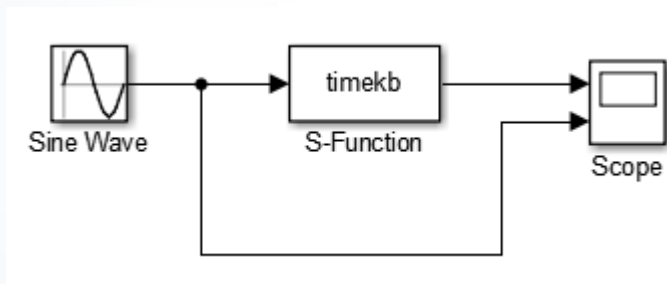
在模型编辑窗口选中S函数模块，选择Diagram→Mask→Create Mask命令，打开封装编辑器，选择Parameters & Dialog选项卡，在左侧控件工具箱中单击Edit工具，往中间的Dialog box区域的控件列表中添加编辑框控件#1，选中该控件后，在右侧的Property editor中，在Name栏填入k，Prompt栏填入“放大倍数”，再设置参数b，设置完成后单击OK按钮。



S函数模块被封装后，双击它，则得到模块参数对话框。当输入k的值为3，b的值为2时，得到的仿真结果。



第一根曲线是经过S函数变化后的曲线 $2+3\sin x$ ，第二根曲线是 $\sin x$ 。



- 利用S函数可以创建用户自己的Simulink模块，从而使得Simulink可以应用于更复杂系统的建模与仿真。
- Simulink提供了现成的S函数模板文件，用户可以通过直接修改模板文件来编写S函数。

9.4 Simulink仿真应用举例

- ❑ 蹦极跳系统的Simulink仿真模型
- ❑ 蹦极跳系统的安全性

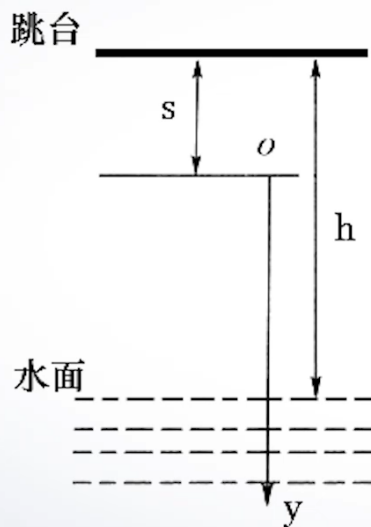
蹦极跳 (Bungee Jumping) 系统的建模与仿真

采用Simulink对蹦极跳系统进行仿真研究。要求如下：

- 建立蹦极跳系统的Simulink仿真模型。
- 分析蹦极跳系统的安全性，包括：
 - 当弹力绳弹性系数一定时，求蹦极者的安全体重。
 - 当蹦极者体重一定时，求弹力绳的最小弹性系数。



1. 系统分析



蹦极者从跳台自由下落，跳台距水面的高度为 h ，弹力绳的长度为 s 。蹦极者受到的力包括自身的重力、弹力绳的张力和空气的阻力。设 k 为弹力绳的弹性系数，同时我们定义人站在跳台时弹力绳的下端为坐标原点 O ， y 为蹦极者相对于坐标原点的距离， $b(y)$ 表示弹力绳的张力，其数学表达式为：

$$b(y) = \begin{cases} -ky, & y > 0 \\ 0, & y \leq 0 \end{cases}$$

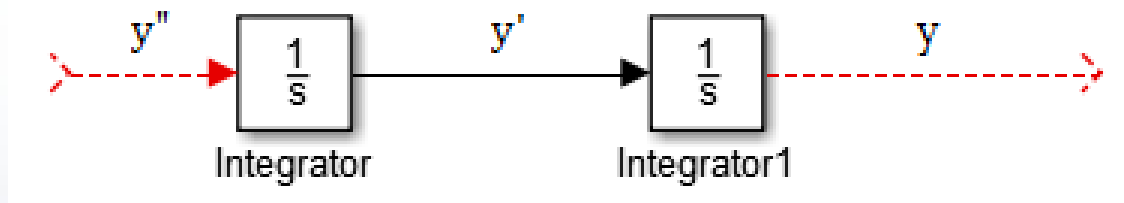
又设 m 为蹦极者的质量， g 为重力加速度， a_1 与 a_2 为空气阻力系数，则系统方程可表示为这样的方程：

$$mg + b(y) - a_1 y' - a_2 y' |y'| = m y''$$

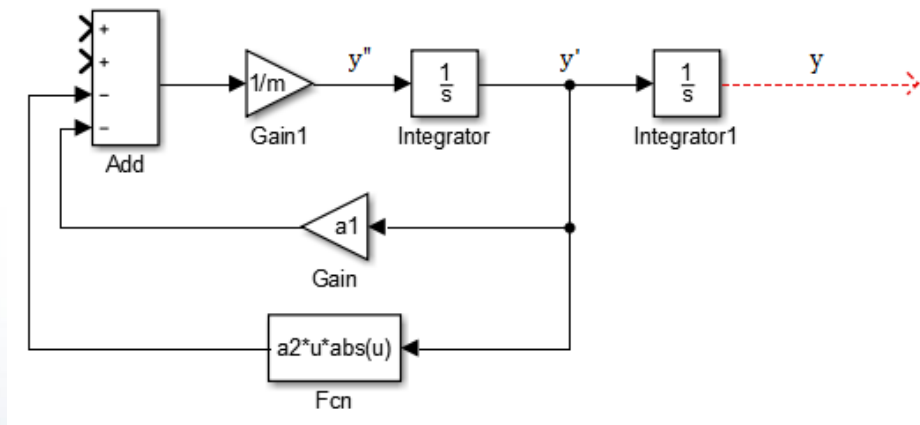
其中第3项和第4项表示空气的阻力。方程的初始条件为 $y(0) = -s$ ， $y'(0) = 0$ ，也就是初始速度为0。

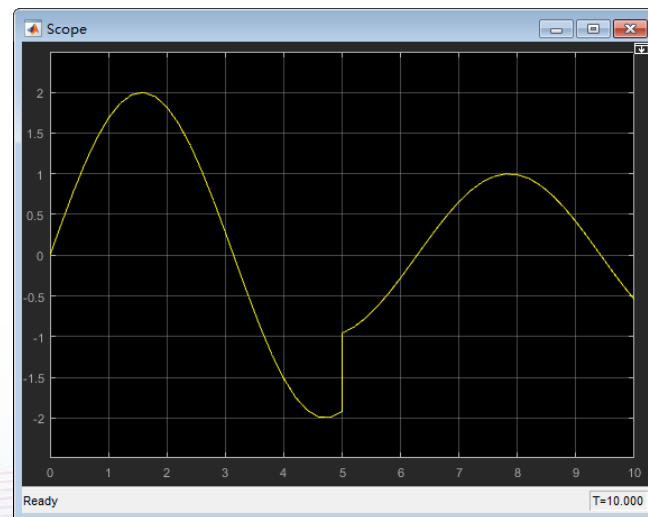
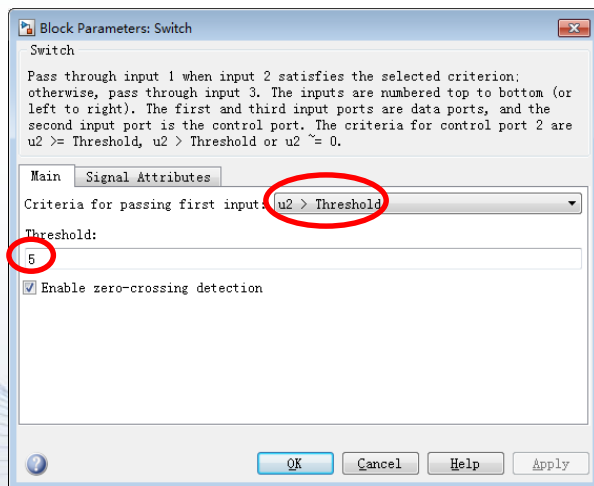
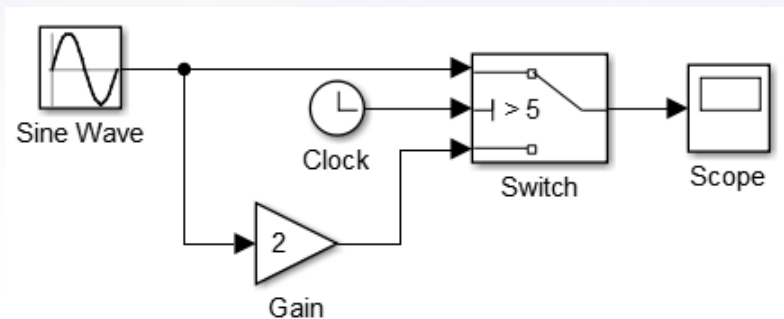
2. 仿真模型

启动Simulink，打开模型编辑窗口。因为该微分方程包含的导数最高为2阶，所以在仿真模型中需要两个积分器，可以从Simulink的连续系统模块库中，将积分模块添加到模型编辑窗口并实现连接， y 的二阶导数经过积分环节后得到 y 的一阶导数， y 的一阶导数经过积分环节后得到 y 。

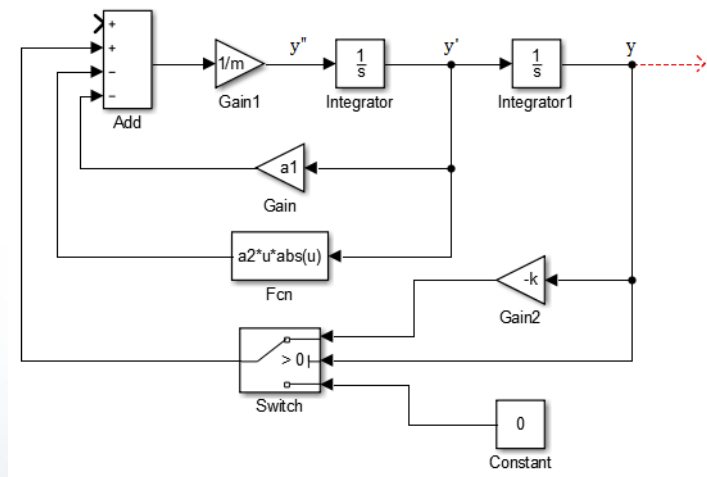


空气阻力有两项，都与 y 的一阶导数，即速度有关。使用数学运算模块库（Math Operations）中的增益模块表示空气阻力比例系数，使用用户自定义函数模块库（User-defined Functions）中函数（Fcn）模块表示空气阻力中的非线性部分，同时将增益模块和函数模块翻转180度，并确定它们的参数。另外添加数学运算模块库（Math Operations）中的加法模块，并设置好参数。还有个增益模块，其输出是 y 的二阶导数。

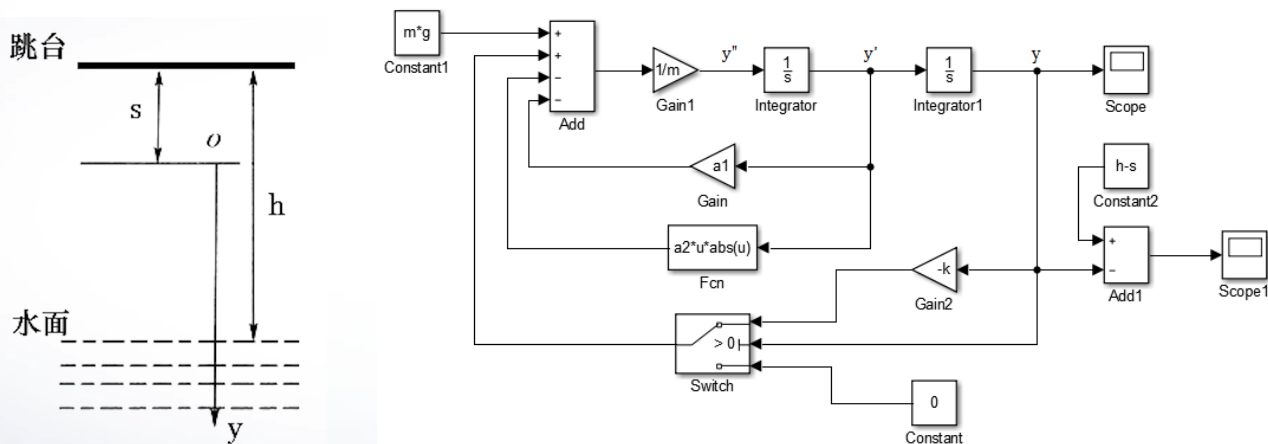




再回到蹦极跳系统，进行弹力绳张力的模块连接。用Switch模块来实现弹力绳的张力，同时需要增益模块和常数模块，设置好参数并进行模块连接后得到此时的仿真模型。



最后添加重力模块和输出模块。添加信号源模块中的常数模块，表示蹦极者的重力，添加输出模块库中的示波器模块，用于输出蹦极者弹出的距离，同时还输出蹦极者与水面的距离，即坐标原点距水面的距离，也就是 $h-s$ ，再减去蹦极者弹起的距离 y ，如果该值小于零，即表示人将在蹦极过程中接触水面，这是不安全的。添加常数模块、加法模块和示波器模块，设置模块参数并进行连接，这样系统仿真模型就建好了。



3. 仿真结果及分析

>> m=75;

>> g=9.8;

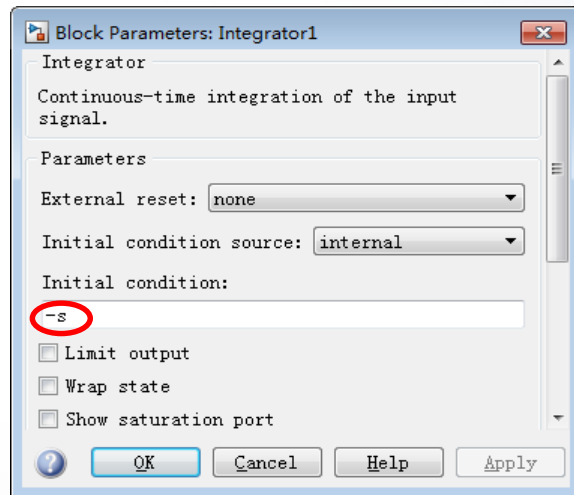
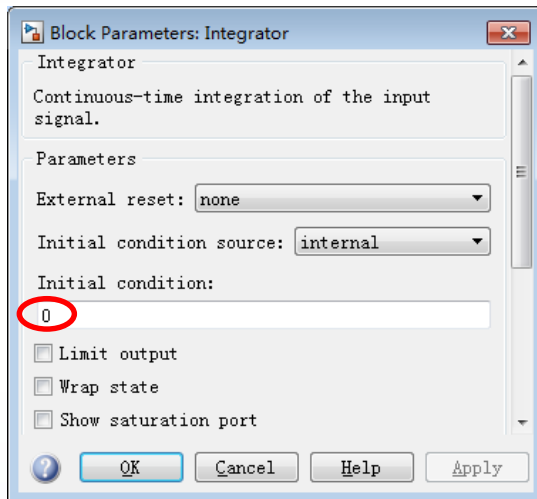
>> k=20;

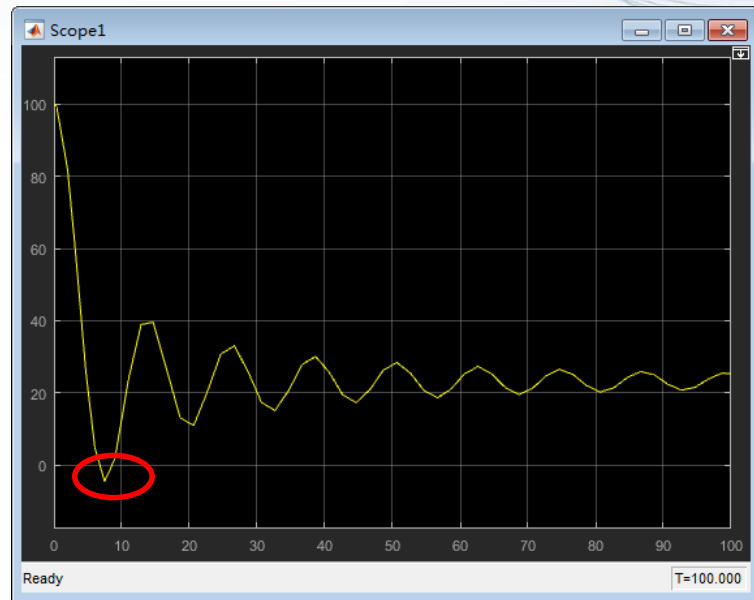
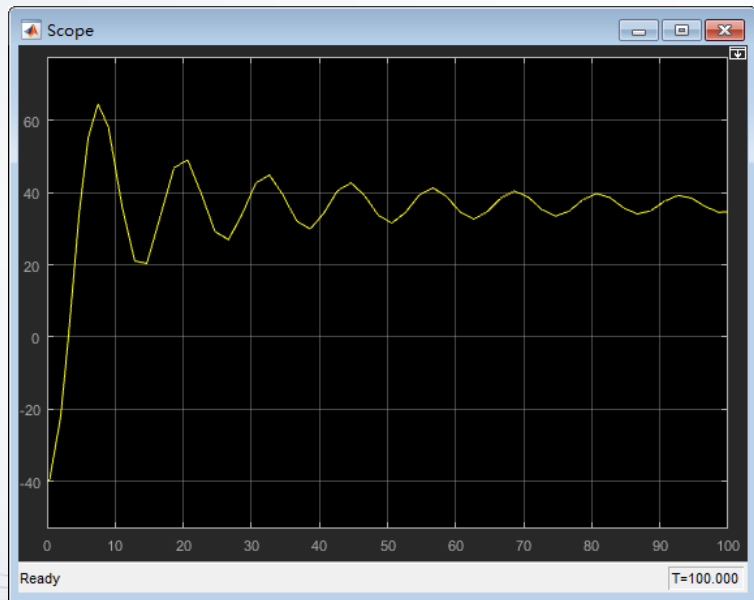
>> a1=1;

>> a2=1;

>> h=100;

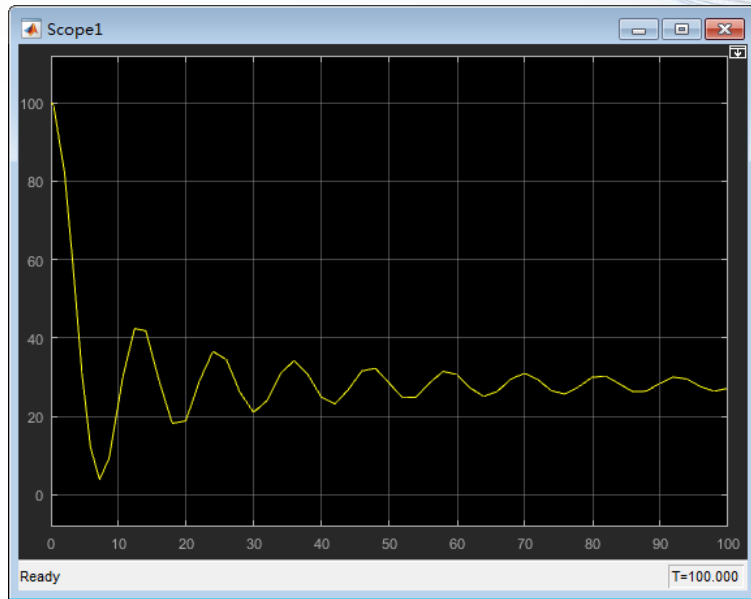
>> s=40;





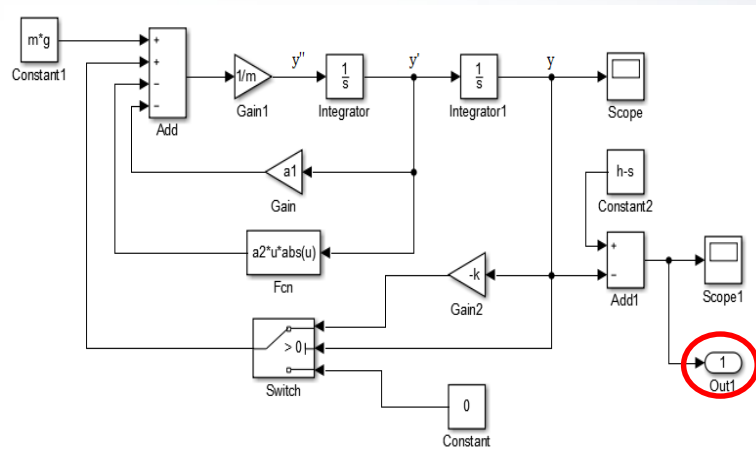
`>> m=65;`

现在将蹦极者的体重改为65kg，其他参数不变，查看此时的曲线，人距离水面的距离大于0，系统是安全的，因此在系统其他参数确定之后，对蹦极者的体重是有要求的。



(1) 求最大安全体重

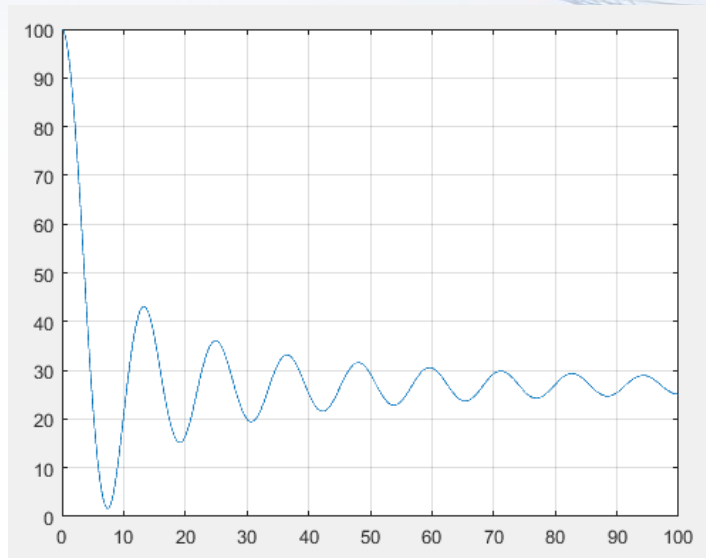
```
for m=100:-0.5:20
    [t,x,y_w]=sim('bengji',0:0.01:100);
    if min(y_w)>1.5
        break;
    end
end
disp(['最大安全体重是',num2str(m)])
dis=min(y_w);
disp(['最小的安全距离是',num2str(dis)])
plot(t,y_w)
grid on
```



程序运行后，得到结果如下：

最大安全体重是67.5

最小的安全距离是1.6803



弹力绳的弹性系数是20时，安全体重是67.5kg，安全距离是1.68m。

人到水面距离的变化曲线如图所示。



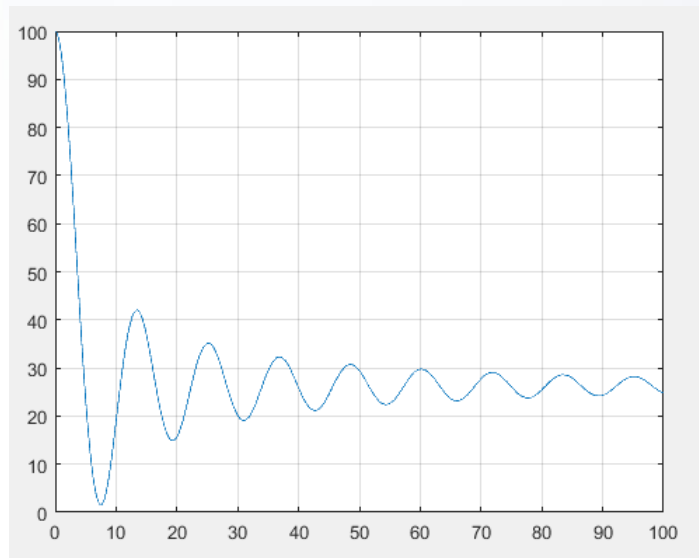
(2) 求最小弹性系数

```
m=65;  
for k=10:0.1:50  
    [t, x, y_w]=sim(' bengji', 0:0.01:100);  
    if min(y_w)>1.5  
        break;  
    end  
end  
disp([' 最小弹性系数k是', num2str(k)])  
dis=min(y_w);  
disp([' 最小的安全距离是', num2str(dis)])  
plot(t, y_w)  
grid on
```

程序运行后，得到结果如下：

最小弹性系数 k 是18.9

最小的安全距离是1.5521



最小安全弹性常数是18.9，蹦极者与地面之间的最小距离是1.55m，
最小安全弹性常数下系统的仿真结果如图所示。