

第3章

Simulink模型的建立与仿真

Simulink 是 MATLAB 的仿真工具箱,它是面向框图的仿真软件。Simulink 能用绘制方框图代替程序,结构和流程清晰;利用 Simulink 可智能化地建立和运行仿真,仿真精细、贴近实际。Simulink 适应面广。可应用于线性、非线性系统,连续、离散及混合系统,以及单任务、多任务离散事件系统。采用 Simulink 模块库能够方便地进行模型的编辑和仿真构建。

学习目标:

- (1) 学习 Simulink 基本库原件;
- (2) 学习 Simulink 各模块的使用;
- (3) 学习 Simulink 各模块的参数配置;
- (4) 学习使用 Simulink 各模块搭建仿真框图。

3.1 Simulink 模块库简介

在 MATLAB 命令行窗口输入 `simulink`,打开 Simulink 工具箱,进行 Simulink 工具箱模块库的学习。Simulink 模块库很庞大,以下将主要介绍常规的 Simulink 应用模块,包括信号源模块组、连续模块组、离散模块组、查表模块组、用户自定义函数模块组、数学运算模块组、非线性模块组、输出池模块组、信号与系统模块组、子系统模块组、常用模块组、其他工具箱与模块集等。

3.2 信号源模块组

Simulink 模块库中提供了丰富的信号源模块组,下面逐一介绍。

3.2.1 Clock 模块

时钟模块以及时钟模块的属性如图 3-1 所示。

时钟模块如图 3-1 所示,在 Simulink 仿真中,时钟模块主要用于计时,效果很直观。

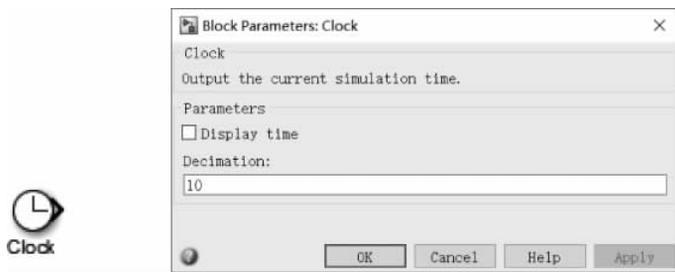


图 3-1 时钟模块

在时钟模块的属性窗口中：

(1) Display time: 如果该复选框被选中,则该时钟模块在仿真过程中,界面将显示时间,如果不显示,则可将其输入到工作区中。

(2) Decimation: 默认为 10,Decimation 的数值可以为任意整数,在仿真过程中,随着时钟不断地更新,其数值不断增加,例如对于 10s 的仿真,系统 Decimation 默认为 10,则表示系统将以 1s、2s、3s、…、10s 依次递增。

搭建时钟模块如图 3-2 所示。

运行仿真文件,输出结果如图 3-3 所示。

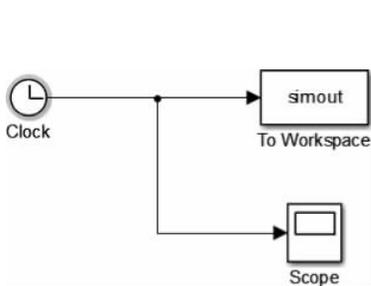


图 3-2 时钟使用

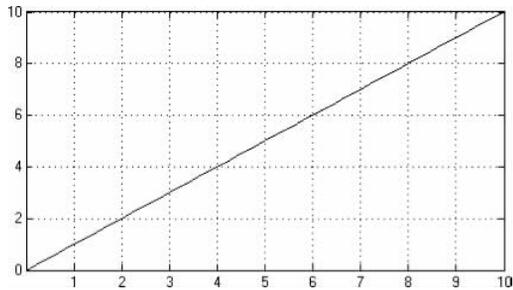


图 3-3 时钟模块示波器时钟变化图

3.2.2 Digital Clock 模块

数字时钟模块以及数字时钟模块的属性如图 3-4 所示。

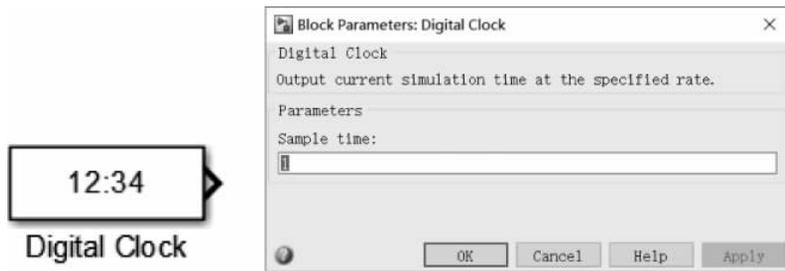


图 3-4 数字时钟模块

在 Simulink 仿真中,数字时钟模块主要用于离散系统的计时,该模块能够输出保持前一次的值不变。

对于其属性窗口: Sample time 表示采样时间,默认值为 1s。

搭建 Digital Clock 模块如图 3-5 所示。

运行仿真文件,输出结果如图 3-6 所示。

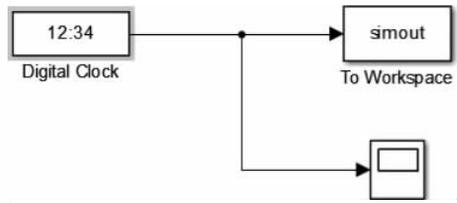


图 3-5 Digital Clock 模块使用

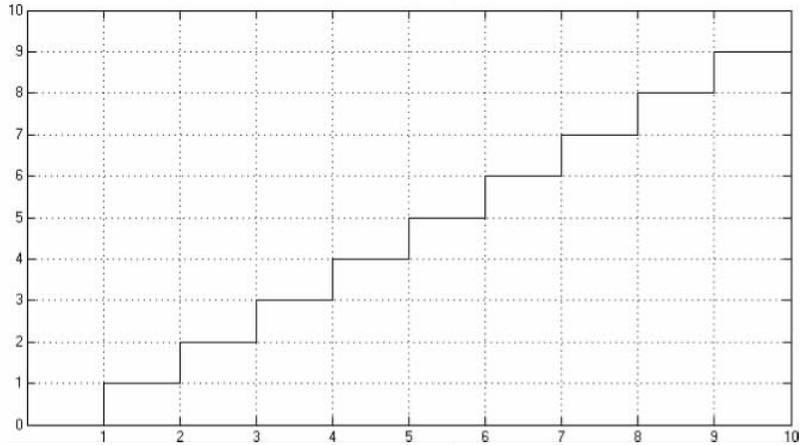


图 3-6 Digital Clock 模块示波器时钟变化图

3.2.3 Constant 模块

Constant 模块,表示常数输入,其模块属性如图 3-7 所示。

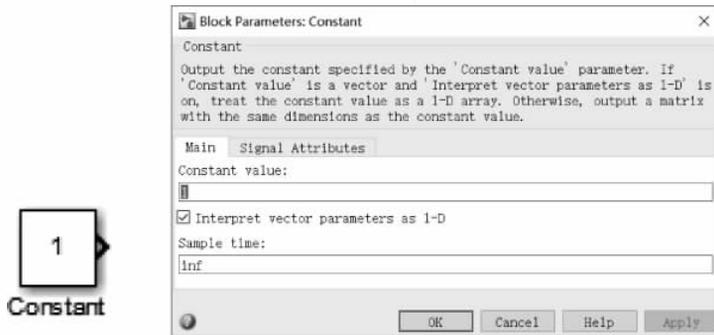


图 3-7 Constant 模块

在 Simulink 仿真中,常数模块主要用在输入的量值为定值的情况。

对于其属性窗口:

- (1) Constant value: 表示常数值,由用户指定。
- (2) Sample time: 表示采样时间,默认值为 inf,也可以设置为与系统的采样时间相一致。

搭建 Constant 模块如图 3-8 所示。

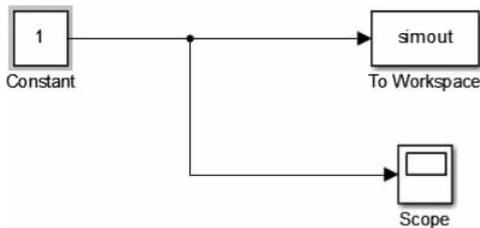


图 3-8 Constant 模块使用

运行仿真文件,输出结果如图 3-9 所示。

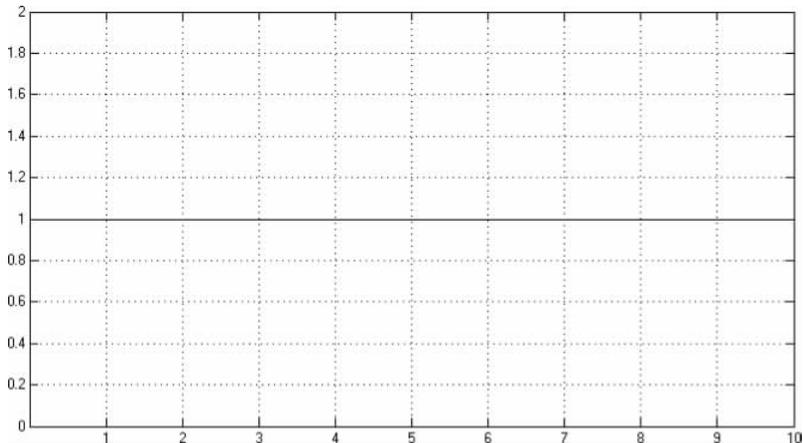


图 3-9 Constant 模块示波器时钟变化图

3.2.4 Band-Limited White Noise 模块

Band-Limited White Noise 模块产生服从正态分布的随机数,用于混合系统或者连续系统,用户可以采样该模块产生比系统最小时间常数更小的相关时间的随机序列来模拟白噪声的效果,通常噪声的相关时间 t 可计算如下:

$$t = \frac{2\pi}{100f_{\max}}$$

其中, f_{\max} (rad/s) 表示系统的带宽。

采用时间 t 作为换算因子,保证了一个连续系统对我们需要近似模拟的白噪声应具有的系统方差(系统噪声),Band-Limited White Noise 模块属性如图 3-10 所示。

对于其属性窗口:

- (1) Noise power: 表示白噪声 PSD 的幅度,默认值为 0.1。
- (2) Sample time: 表示采样时间,默认值为 0.1。
- (3) Seed: 表示随机数信号发生器的初始种子,默认值为 [23341]。

搭建 Band-Limited White Noise 模块如图 3-11 所示。

运行仿真文件,输出结果如图 3-12 所示。

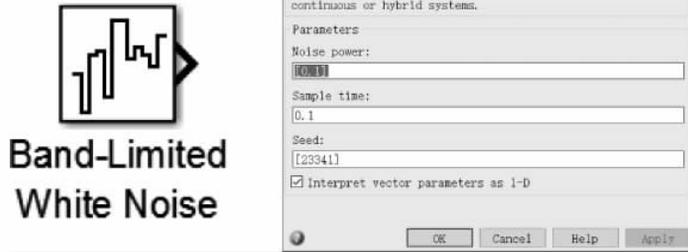


图 3-10 Band-Limited White Noise 模块

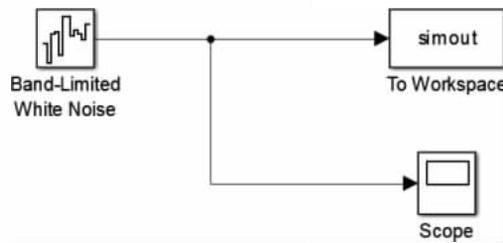


图 3-11 Band-Limited White Noise 模块使用

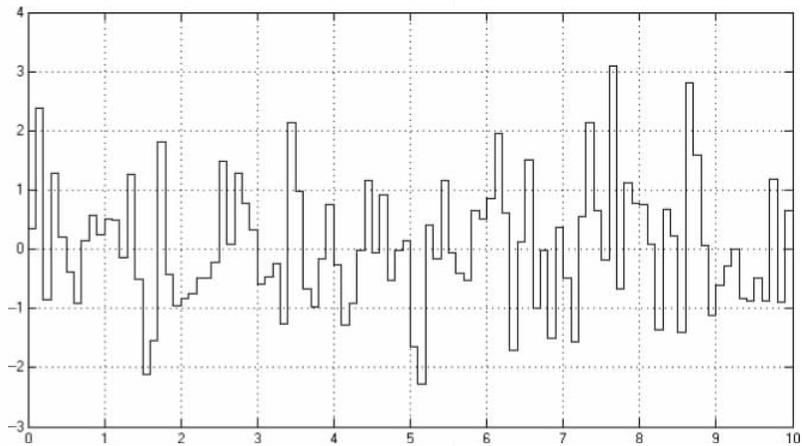


图 3-12 Band-Limited White Noise 模块示波器时钟变化图

3.2.5 Chirp Signal 模块

Chirp Signal 模块产生频率随时间线性增加的正弦信号,即调频信号,该模块可用于非线性系统的谱分析,且以矢量或标量输出。

Chirp Signal 模块的属性窗口如图 3-13 所示。

对于其属性窗口:

(1) Initial frequency(Hz): 表示信号的初始化频率,指定为标量或矢量,默认值为

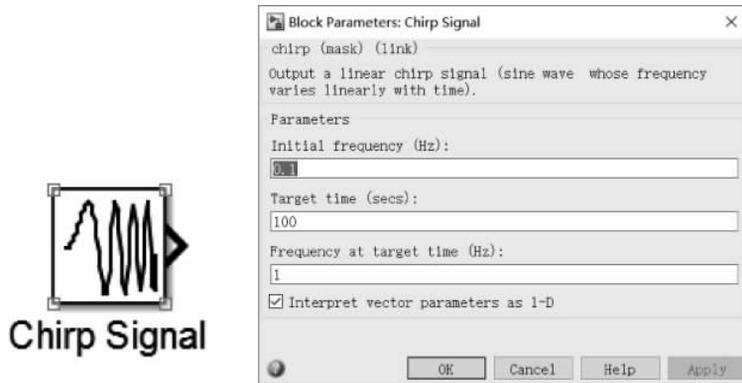


图 3-13 Chirp Signal 模块

0.1。

(2) Target time(secs): 表示频率变化的最大时间,默认值为 100。

(3) Frequency at target time(Hz): 表示对应目标时间的信号频率,输入为矢量或标量,默认值为 1。

搭建 Chirp Signal 模块如图 3-14 所示。

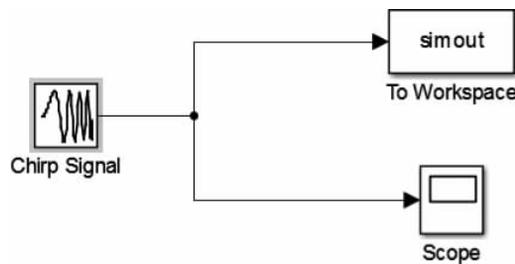


图 3-14 Chirp Signal 模块使用

运行仿真文件,输出结果如图 3-15 所示。

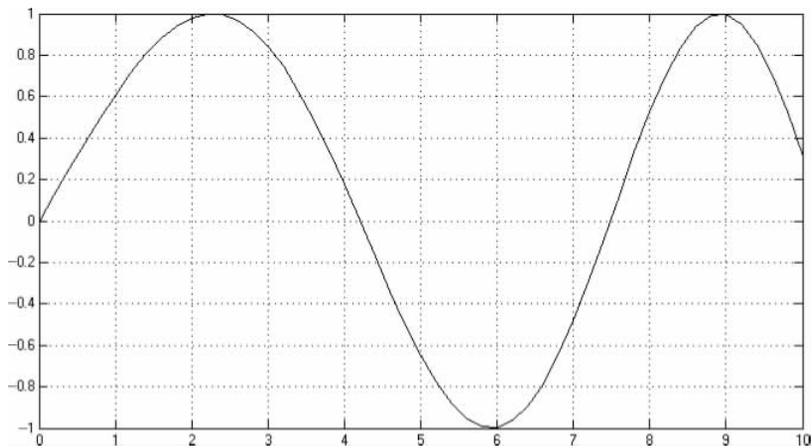


图 3-15 Chirp Signal 模块示波器时钟变化图

3.2.6 Sine Wave 模块

Sine Wave 正弦波模块,产生如下形式的正弦波:

$$f(t) = \text{Amp} \cdot \sin(\text{Freq} \cdot t + \text{Phase}) + \text{Bias}$$

其中,Amp 为正弦波振幅,Freq 为正弦波的频率,Phase 为初始相位,Bias 为正弦波上下移动的常量。

Sine Wave 正弦波的模块属性如图 3-16 所示。

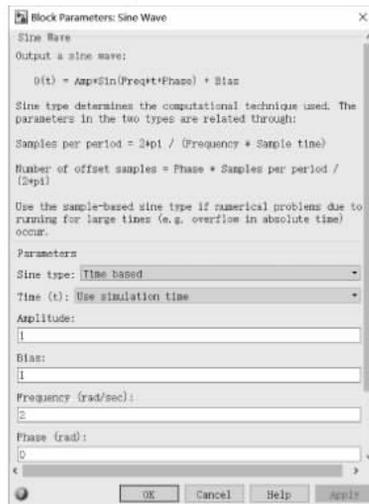
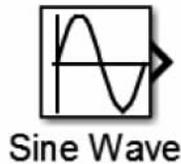


图 3-16 Sine Wave 正弦波模块

对于其属性窗口:

- (1) Amplitude: 表示正弦信号的振幅,指定为标量或矢量,默认值为 1。
- (2) Bias: 表示正弦信号离 0 均值线的偏移量,默认值为 0。
- (3) Frequency(rad/sec): 表示对应目标信号频率,输入为矢量或标量,默认值为 1。
- (4) Phase(rad): 表示信号的初始相位,默认值为 0。
- (5) Sample time: 表示系统采样时间。
- (6) Inter vector parameters as 1-D: 该复选框可以选中,也可以不选,选中表示信号按照一行的数据矢量进行输出;不勾选,则信号以列向量存储。

搭建 Sine Wave 模块如图 3-17 所示。

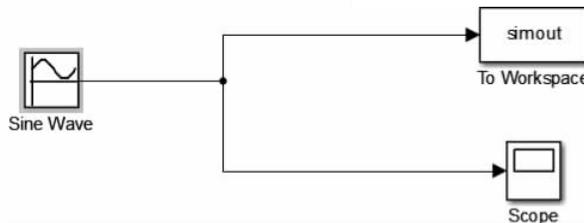


图 3-17 Sine Wave 模块使用

运行仿真文件,输出结果如图 3-18 所示。

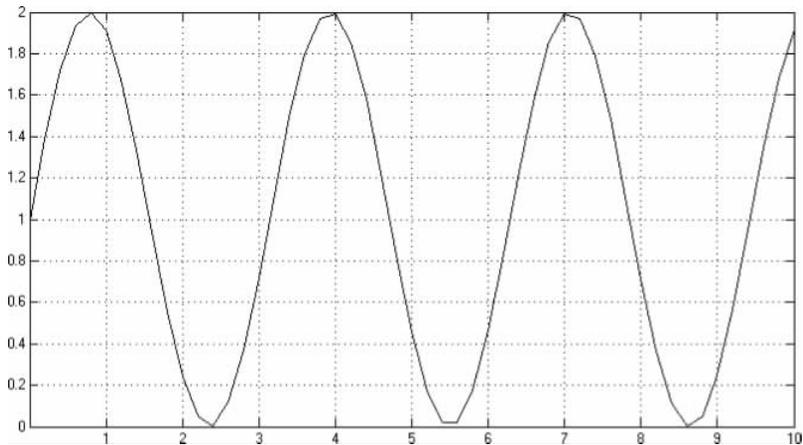


图 3-18 Sine Wave 模块示波器时钟变化图

3.2.7 Pulse Generator 模块

Pulse Generator 模块产生等间隔的脉冲波形,脉冲宽度就是脉冲持续高电平期间的数字采样周期数,脉冲周期等于脉冲持续高电平、低电平的数字采样周期之和,相位延迟则是起始脉冲所对应的数字采样周期数。

Pulse Generator 的模块属性如图 3-19 所示。

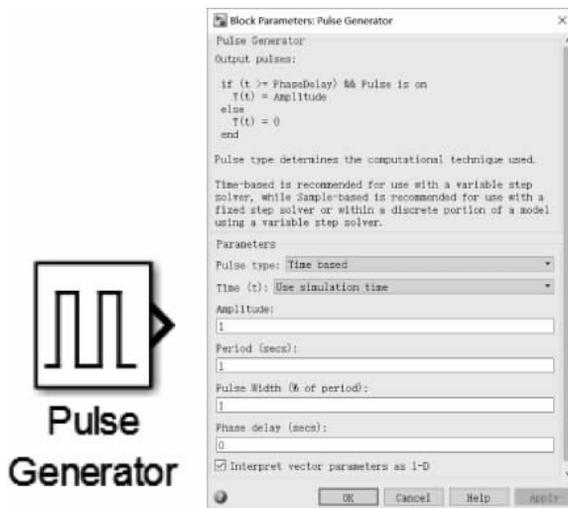


图 3-19 Pulse Generator 模块

对于其属性窗口:

- (1) Amplitude: 表示脉冲信号的振幅,指定为标量或矢量,默认值为 1。
- (2) Period(secs): 表示脉冲数字采样周期,默认值为 10。
- (3) Pulse width(% of period): 表示脉冲宽度,输入为矢量或标量,默认值为 5。

(4) Phase delay(secs): 表示信号的相位延迟, 默认值为 0。

(5) Inter vector parameters as 1-D: 该复选框可以选中, 也可以不选, 选中表示信号按照一行的数据矢量进行输出; 不勾选, 则信号以列向量存储。

搭建 Pulse Generator 模块如图 3-20 所示。

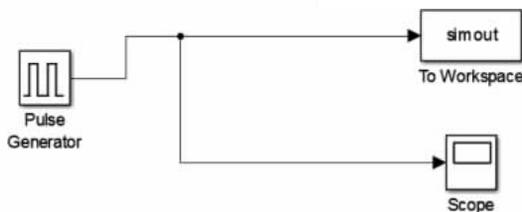


图 3-20 Pulse Generator 模块使用

运行仿真文件, 输出结果如图 3-21 所示。

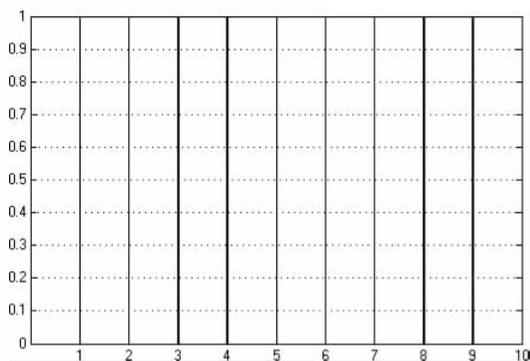


图 3-21 Pulse Generator 模块示波器时钟变化图

3.2.8 Random Number 模块

Random Number 模块产生服从正态分布的随机信号, 在每次仿真开始时, 种子都设置为指定的值, 默认的情况下, 产生方差为 1、均值为 0 的随机信号。如果想获得均匀分布的随机信号, 则可以使用 Uniform Random Number 模块; 如果仿真器对于比较平滑的信号能够积分, 那么对于随机波动的信号进行积分运算, 则需要采用 Band-Limited White Noise 信号。

Random Number 的模块属性如图 3-22 所示。

对于其属性窗口:

- (1) Mean: 表示随机信号的均值, 指定为标量或矢量, 默认值为 0。
- (2) Variance: 表示随机信号的方差, 默认值为 1。
- (3) Seed: 表示随机种子, 输入为矢量或标量, 默认值为 0。
- (4) Sample time: 表示信号的采样时间, 默认值为 0.1。

搭建 Random Number 模块如图 3-23 所示。运行仿真文件, 输出结果如图 3-24 所示。

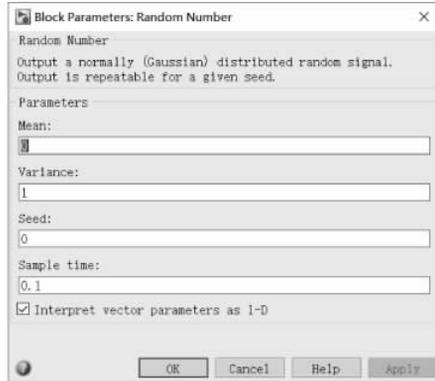


图 3-22 Random Number 模块

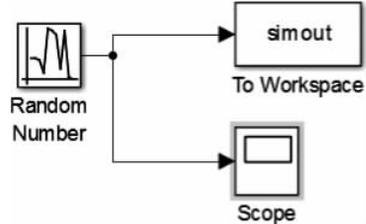


图 3-23 Random Number 模块使用

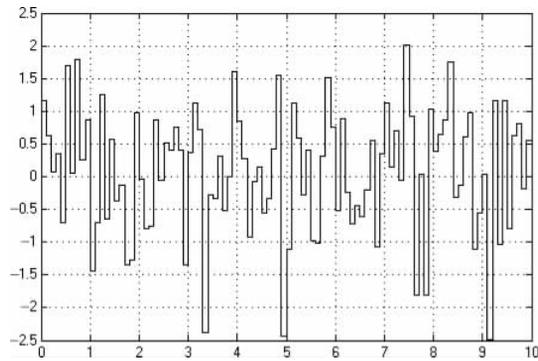


图 3-24 Random Number 模块示波器时钟变化图

3.2.9 Step 模块

Step 模块产生阶跃信号, Step 常用于控制系统仿真中, 用于测试系统的稳定性和收敛性。Step 模块在指定时间产生一个可定义上、下电平的阶跃信号, Step 产生一个矢量或标量进行输出。

Step 的模块属性如图 3-25 所示。

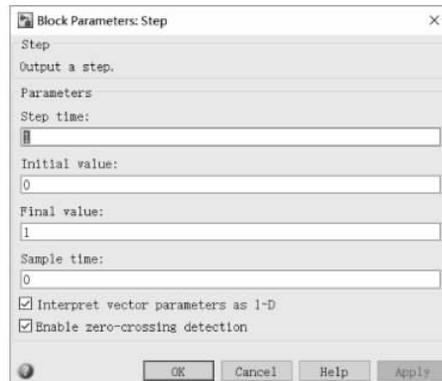
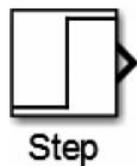


图 3-25 Step 模块

对于其属性窗口：

- (1) Step time: 表示初始阶跃的时间,指定为标量或矢量,系统默认值为 1。
- (2) Initial value: 表示仿真的初始时间,系统默认值为 0。
- (3) Final time: 表示仿真的结束时间,输入为矢量或标量,系统默认值为 1。
- (4) Sample time: 表示信号的采样时间,系统默认值为 0。

搭建 Step 模块如图 3-26 所示。运行仿真文件,输出结果如图 3-27 所示。

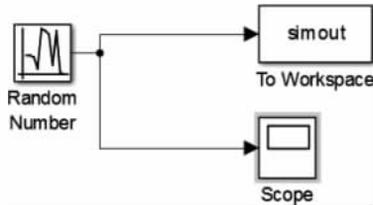


图 3-26 Step 模块使用

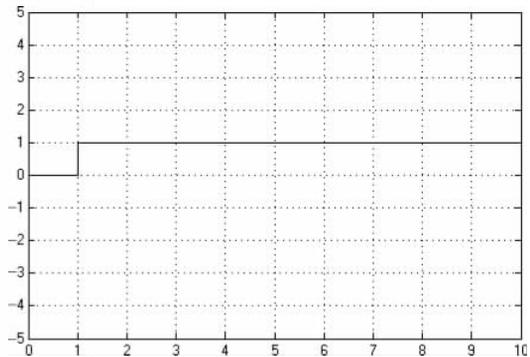


图 3-27 Step 模块示波器时钟变化图

3.2.10 Uniform Random Number 模块

Uniform Random Number 模块产生在整个指定时间周期内均匀分布的随机信号,信号的起始种子可由用户指定。将 Seed 种子指定为矢量,可以产生矢量随机数序列。

Uniform Random Number 的模块属性如图 3-28 所示。

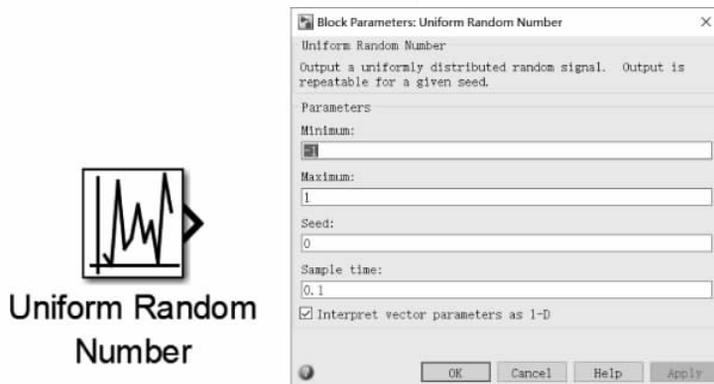


图 3-28 Uniform Random Number 模块

对于其属性窗口：

- (1) Minimum: 表示时间间隔的最小值,指定为标量或矢量,系统默认值为 -1。
- (2) Maximum: 表示时间间隔的最大值,指定为标量或矢量,系统默认值为 1。
- (3) Seed: 表示随机序列发生器的初始种子,输入为矢量或标量,系统默认值为 0。
- (4) Sample time: 表示信号的采样时间,系统默认值为 0.1。

搭建 Uniform Random Number 模块如图 3-29 所示。运行仿真文件,输出结果如图 3-30 所示。

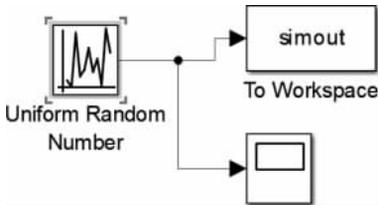


图 3-29 Uniform Random Number 模块使用

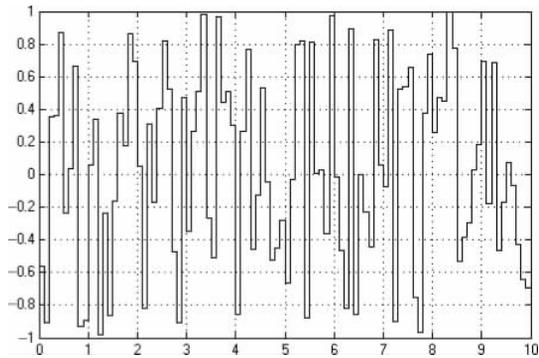


图 3-30 Uniform Random Number 模块示波器时钟变化图

3.3 连续模块组

连续模块仿真主要用于系统的积分分析。对于一个系统而言,传递函数的构建显得尤为重要。

3.3.1 Derivative 模块

Derivative 模块表示微分环节,为时间的一阶导数 $\frac{\Delta u}{\Delta t}$,其中, Δu 为输入的变化量, Δt 为前两次仿真时间点之差。

Derivative 模块的仿真精度取决于时间步长 Δt ,步长越小,结果越平滑,相应的结果越精确。如果输入为离散信号,当输入变化时,输入的连续导数是冲击信号,否则为 0。为得到离散型系统的离散导数,可采用

$$y(k) = \frac{1}{\Delta t} [u(k) - u(k-1)]$$

相应的 Z 变换为

$$\frac{Y(z)}{u(z)} = \frac{1 - z^{-1}}{\Delta t} = \frac{z - 1}{\Delta t \cdot z}$$

Derivative 的模块属性如图 3-31 所示。

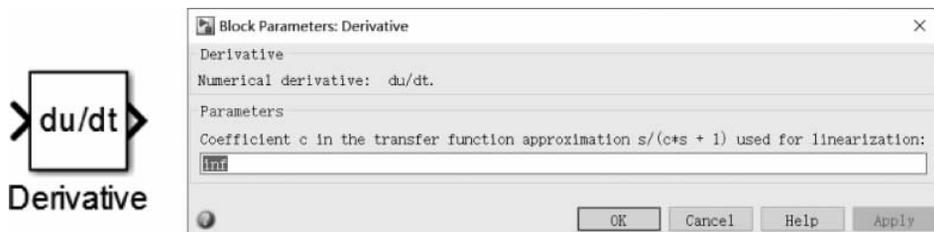


图 3-31 Derivative 模块

对于其属性窗口: Coefficient c in the transfer function approximation $s/(c * s + 1)$ used for linearization 表示步长的设置, 指定为标量或矢量, 默认值为 inf (无穷大)。

搭建 Derivative 模块如图 3-32 所示。运行仿真文件, 输出结果如图 3-33 所示。

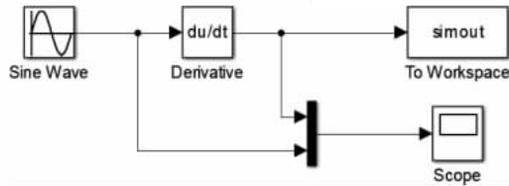


图 3-32 Derivative 模块使用

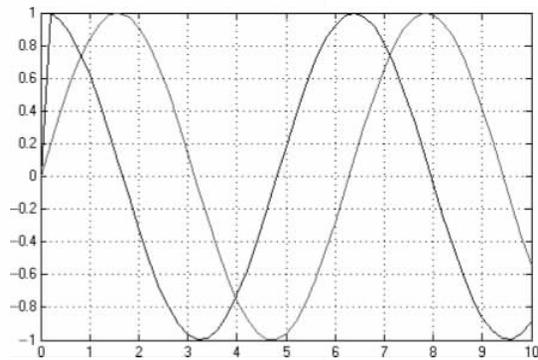


图 3-33 Derivative 模块示波器时钟变化图

3.3.2 Integrator 模块

Integrator 模块表示积分环节, 为时间的一阶导数 $\int u dt$, 其中, u 为输入的变化量, dt 为前两次仿真时间点之差。

Integrator 的模块属性如图 3-34 所示。

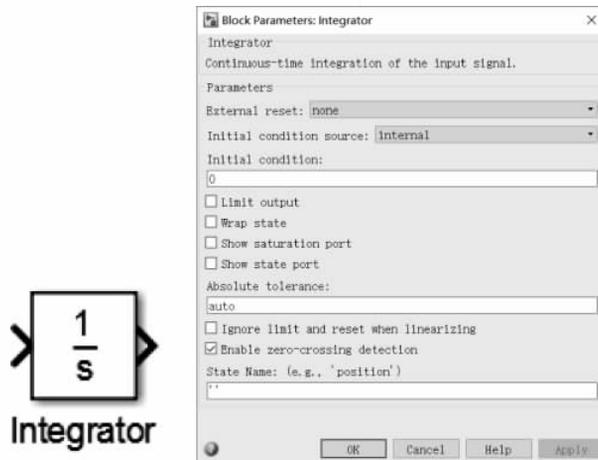


图 3-34 Integrator 模块

对于其属性窗口：

(1) External reset: 设置信号的触发事件(rising、falling、either、level、level hold 和 none), 默认设置为 none, 即保持系统原态。

(2) Initial condition source: 表示参数输入的状态, 分为外部输入 external 和内部输入 internal, 通常默认设置为 internal。

(3) Initial condition: 表示状态的初始条件, 用于设置 Initial condition source 的参数。

(4) Limit output: 若选中, 则可以设置积分的上界(Upper saturation limit)和下界(Lower saturation limit)。

(5) Upper saturation limit: 表示积分上界, 默认值为 inf。

(6) Lower saturation limit: 表示积分下界, 默认值为 inf。

(7) Show saturation port: 若选中, 则表示模块增加一个饱和输出端口。

(8) Show state port: 若选中, 则表示模块增加一个输出端口。

(9) Absolute tolerance: 表示模块状态的绝对容限, 默认值为 auto。

(10) Ignore limit and reset when linearizing: 若勾选此选项, 则表示当系统为线性化系统时, 前面的积分上下限制和触发事件无效, 默认为不勾选。

(11) Enable zero-crossing detection: 使系统通过零点检验, 默认勾选。

搭建 Integrator 模块如图 3-35 所示。运行仿真文件, 输出结果如图 3-36 所示。

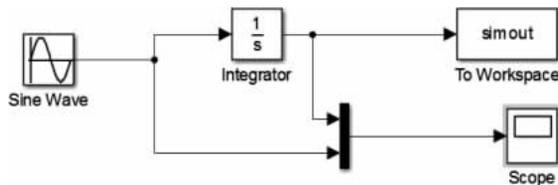


图 3-35 Integrator 模块使用

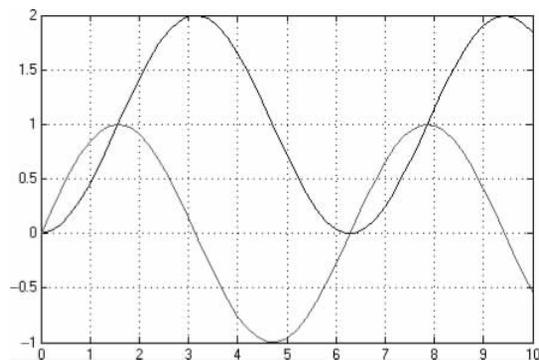


图 3-36 Integrator 模块示波器时钟变化图

3.3.3 Transfer Fcn 模块

Transfer Fcn 模块用于表征传递函数, 具体的传递函数的表达式如下:

$$H(s) = \frac{y(s)}{u(s)} = \frac{a_n s^n + a_{n-1} s^{n-1} + \dots + a_1 s + a_0}{b_m s^m + b_{m-1} s^{m-1} + \dots + b_1 s + b_0}$$

其中, $y(s)$ 为系统输出, $u(s)$ 为系统输入, 传递函数的计算则通过用户得到的系统模型而来, 对于一个收敛性系统而言, 分母中 s 的最高次幂大于分子中 s 的最高次幂。

Transfer Fcn 的模块属性如图 3-37 所示。

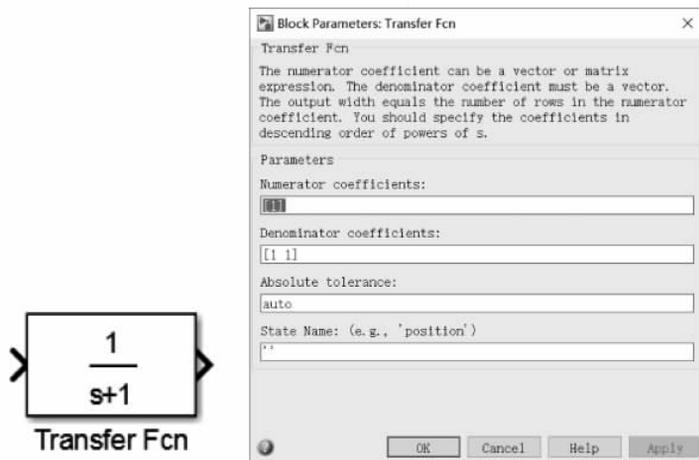


图 3-37 Transfer Fcn 模块

对于其属性窗口:

- (1) Numerator coefficients: 表示传递函数分子系数, 系统默认值为 [1]。
 - (2) Denominator coefficients: 表示传递函数分母系数, 系统默认值为 [1 1]。
 - (3) Absolute tolerance: 表示模块状态的绝对容限, 默认值为 auto。
 - (4) State Name (e. g. , 'position'): 表示状态空间的名字, 用户可以不加以定义。
- 搭建 Transfer Fcn 模块如图 3-38 所示。运行仿真文件, 输出结果如图 3-39 所示。

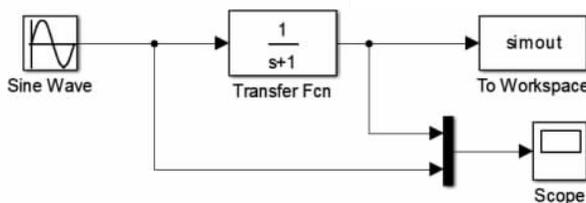


图 3-38 Transfer Fcn 模块使用

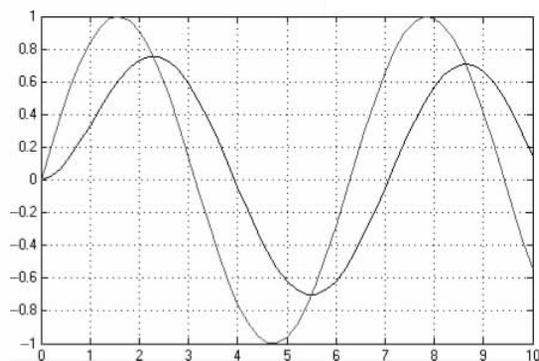


图 3-39 Transfer Fcn 模块示波器时钟变化图

3.3.4 Transport Delay 模块

Transport Delay 模块用于延时系统的输入,延时的时间可以由用户指定。在仿真过程中,模块将输入点和仿真时间存储在一个缓冲器内,该缓冲器的容量由 Initial buffer size 参数指定。若输入点数超出缓冲器的容量,模块将配置额外的存储区。

Transport Delay 模块不能对离散信号进行插值计算,模块返回区间 $t - t_{\text{delay}}$ (当前时间减去时间延迟) 对应的离散值。

Transport Delay 的模块属性如图 3-40 所示。

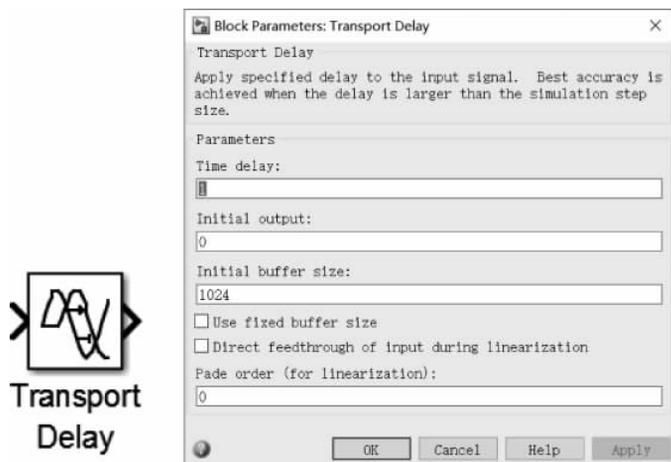


图 3-40 Transport Delay 模块

对于其属性窗口:

- (1) Time delay: 表示系统延时量,系统默认值为 1。
- (2) Initial output: 表示系统在开始仿真和 Time delay 之间产生的输出,系统默认值为 0。
- (3) Initial buffer size: 表示储存点数的初始存储区配置,系统默认值为 auto。
- (4) Use fixed buffer size: 储存点数的初始存储区配置为固定值,用户可以不加以定义。

搭建 Transport Delay 模块如图 3-41 所示。运行仿真文件,输出结果如图 3-42 所示。

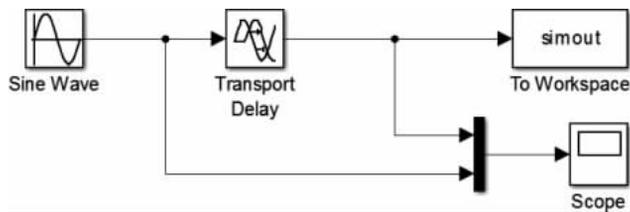


图 3-41 Transport Delay 模块使用

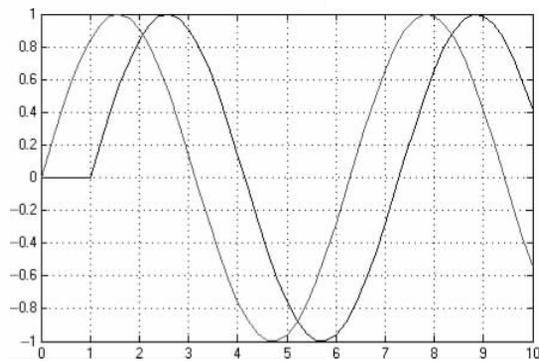


图 3-42 Transport Delay 模块示波器时钟变化图

3.3.5 Zero-Pole 模块

Zero-Pole 模块用于表征一个以 Laplace 算子 s 为变量的零点、极点和增益的系统，其传递函数可表示为：

$$H(s) = \frac{y(s)}{u(s)} = \frac{a_n s^n + a_{n-1} s^{n-1} + \cdots + a_1 s + a_0}{b_m s^m + b_{m-1} s^{m-1} + \cdots + b_1 s + b_0}$$

它的变形为以 s 为变量的零点、极点和增益的系统，如下：

$$H(s) = K \frac{Z(s)}{P(s)} = K \frac{(s - Z(1))(s - Z(2)) \cdots (s - Z(n))}{(s - P(1))(s - P(2)) \cdots (s - P(m))}$$

其中， Z 代表零点； P 为极点矢量； K 为增益。

Zero-Pole 模块的输入和输出宽度等于零点矩阵的行数。

Zero-Pole 的模块属性如图 3-43 所示。

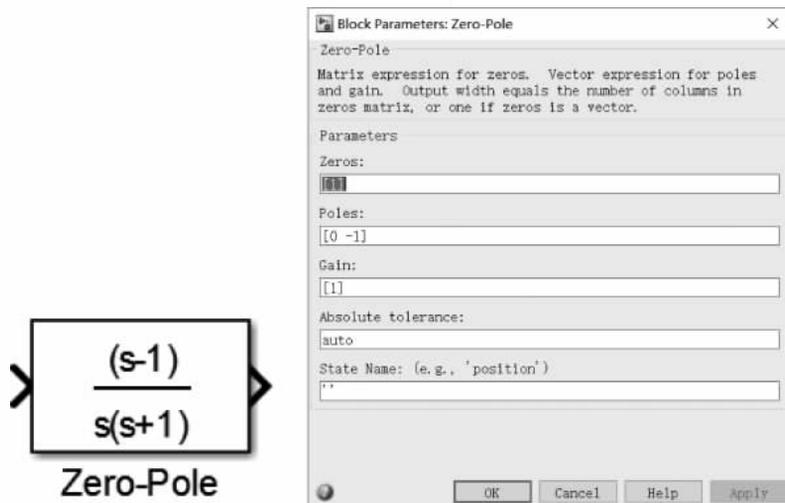


图 3-43 Zero-Pole 模块

对于其属性窗口：

(1) Zeros：表示系统传递函数零点向量，系统默认值为[1]。

- (2) Poles: 表示系统传递函数极点向量,系统默认值为 $[0 \ -1]$ 。
- (3) Gain: 表示系统传递函数增益向量,系统默认值为 $[1]$ 。
- (4) Absolute tolerance: 表示模块状态的绝对容限,系统默认值为 auto。
- (5) State Name(e. g. , 'position'): 表示状态空间的名字,用户可以不加以定义。
- 搭建 Zero-Pole 模块如图 3-44 所示。运行仿真文件,输出结果如图 3-45 所示。

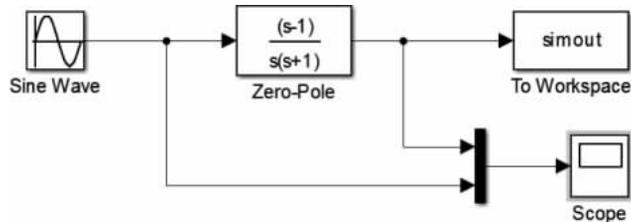


图 3-44 Zero-Pole 模块使用

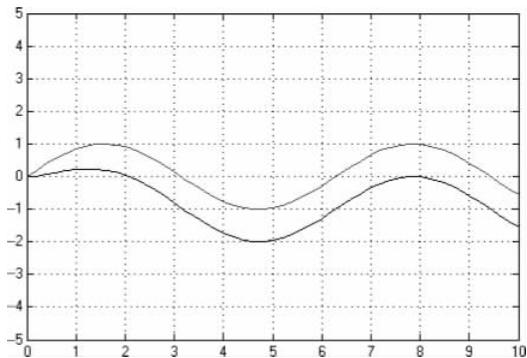


图 3-45 Zero-Pole 模块示波器时钟变化图

3.3.6 State-Space 模块

State-Space 模块用于表征一个控制系统的状态空间,具体的状态空间的表达式如下:

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \end{cases}$$

其中, x 为状态矢量; u 为输入矢量; y 为输出矢量。

State-Space 的模块属性如图 3-46 所示。

对于其属性窗口:

- (1) A: 表示系统状态空间矩阵系数,必须是一个 $n \times n$ 矩阵, n 为状态数,系统默认值为 1。
- (2) B: 表示系统状态空间矩阵系数,必须是一个 $n \times m$ 矩阵, m 为状态数,系统默认值为 1。
- (3) C: 表示系统状态空间矩阵系数,必须是一个 $r \times n$ 矩阵, r 为状态数,系统默认值

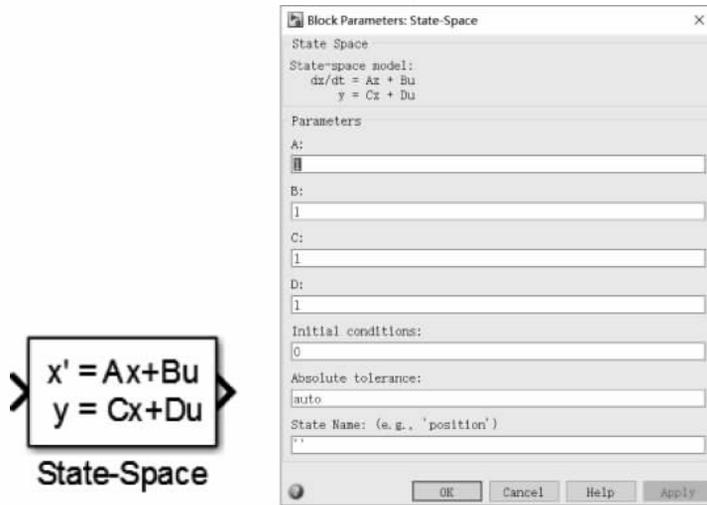


图 3-46 State-Space 模块

为 1。

- (4) D: 表示系统状态空间矩阵系数,必须是一个 $r \times m$ 矩阵,系统默认值为 1。
 - (5) Initial conditions: 表示初始状态矢量,系统默认值为 0。
 - (6) Absolute tolerance: 表示模块状态的绝对容限,系统默认值为 auto。
 - (7) State Name(e.g., 'position'): 表示状态空间的名字,用户可以不加以定义。
- 搭建 State-Space 模块如图 3-47 所示。运行仿真文件,输出结果如图 3-48 所示。

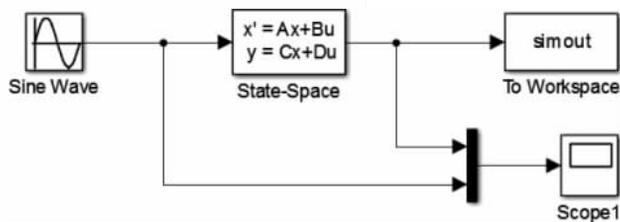


图 3-47 State-Space 模块使用

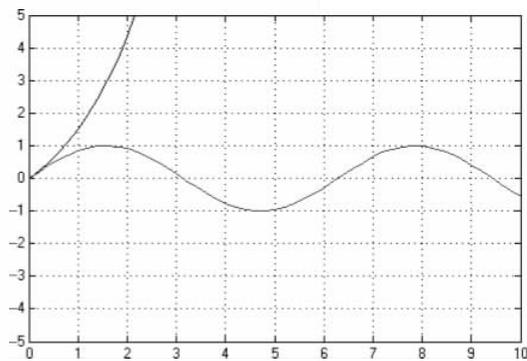


图 3-48 State-Space 模块示波器时钟变化图

3.4 离散模块组

现实系统中有很多系统都是离散系统,系统根据采样时间点进行数据采集分析,Simulink 中离散系统的表征主要是根据 Z 变换进行系统仿真建模。

3.4.1 Discrete Transfer Fcn 模块

对于 Discrete Transfer Fcn 模块,由通常的拉普拉斯变换后,得到相应的传递函数,再经过 Z 变换,得到离散系统传递函数,具体如下:

$$H(z) = \frac{\text{num}(z)}{\text{den}(z)} = \frac{a_n z^n + a_{n-1} z^{n-1} + \dots + a_0 z^0}{b_m z^m + b_{m-1} z^{m-1} + \dots + b_0 z^0}$$

其中,num(z)为离散系统传递函数的分子系数,den(z)为离散系统传递函数的分母系数。

Discrete Transfer Fcn 的模块属性如图 3-49 所示。

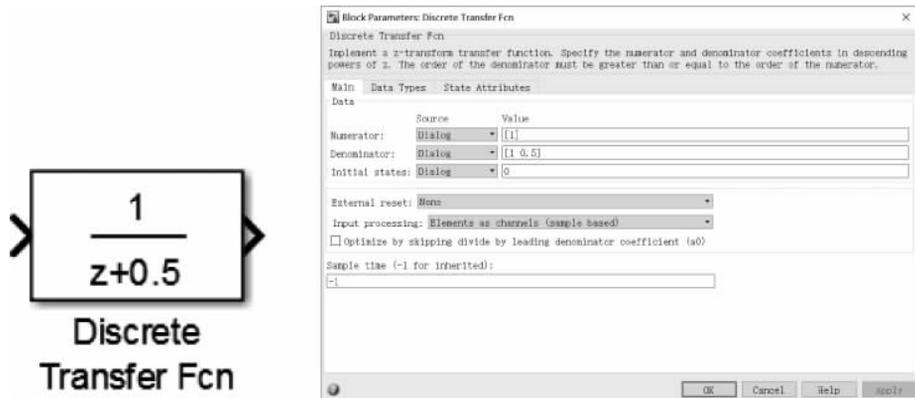


图 3-49 Discrete Transfer Fcn 模块

对于其属性窗口:

- (1) Numerator: 表示系统分子系数矢量,系统默认值为[1]。
- (2) Denominator: 表示系统分母系数矢量,系统默认值为[1 2]。
- (3) Sample time(-1 for inherited): 表示系统采样时间,系统默认值为[-1]。
- (4) Initial states: 表示系统初始状态矩阵,系统默认值为 0。

搭建 Discrete Transfer Fcn 模块,设置采样时间为 0.1s,如图 3-50 所示。运行仿真文件,输出结果如图 3-51 所示。

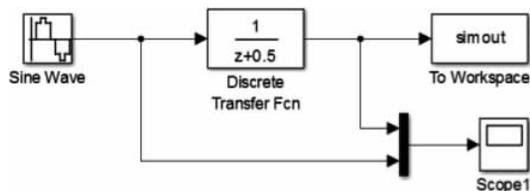


图 3-50 Discrete Transfer Fcn 模块使用

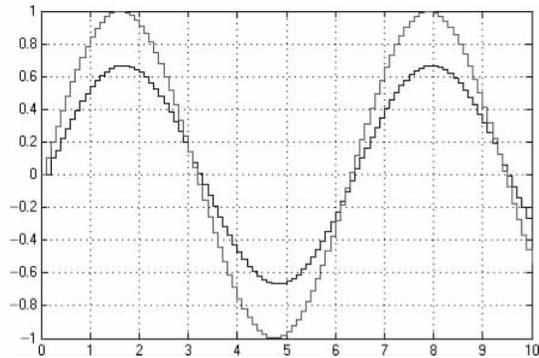


图 3-51 Discrete Transfer Fcn 模块示波器时钟变化图

3.4.2 Discrete Filter 模块

Discrete Filter 模块可实现无限冲激响应(IIR)和有限冲激响应(FIR)滤波器,用户可用 Numerator 和 Denominator 参数指定以 z^{-1} 的升幂为矢量的分子和分母多项式的系数。分母的阶数大于或等于分子的系数。

Discrete Filter 模块提供了自动控制中用 z 描述离散系统的方法。在信号处理中,Discrete Filter 模块提供了 z^{-1} (延迟算子)多项式以描述数字滤波器。

Discrete Filter 的模块属性如图 3-52 所示。

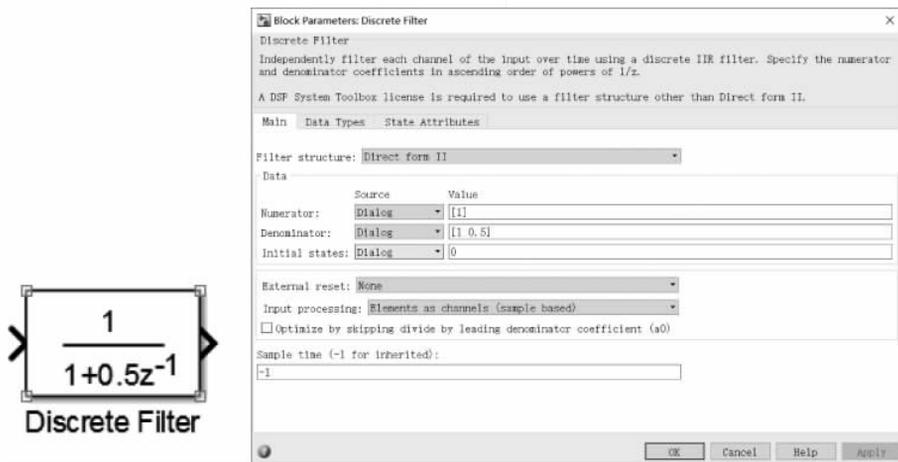


图 3-52 Discrete Filter 模块

对于其属性窗口:

- (1) Numerator: 表示系统分子系数矢量,系统默认值为[1]。
- (2) Denominator: 表示系统分母系数矢量,系统默认值为[1 2]。
- (3) Sample time: 表示系统采样时间,系统默认值为-1。
- (4) Initial states: 表示系统初始状态矩阵,系统默认值为0。

搭建 Discrete Filter 模块,设置采样时间为 0.1s,如图 3-53 所示。运行仿真文件,输

出结果如图 3-54 所示。

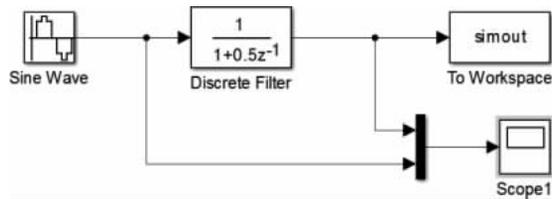


图 3-53 Discrete Filter 模块使用

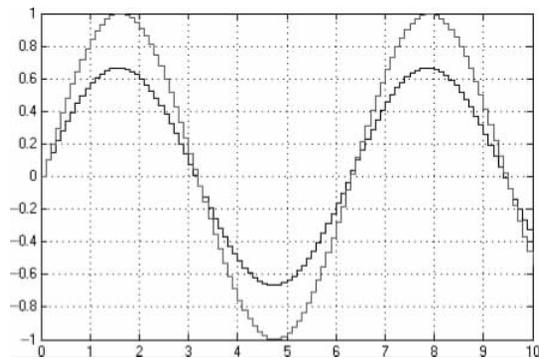


图 3-54 Discrete Filter 模块示波器时钟变化图

3.4.3 Unit Delay 模块

Unit Delay 模块将输入矢量延迟,并保持在同一个采样周期里。若模块的输入为矢量,则系统所有输出量均被延迟一个采样周期,本模块相当于一个 z^{-1} 的时间离散算子。

Unit Delay 的模块属性如图 3-55 所示。

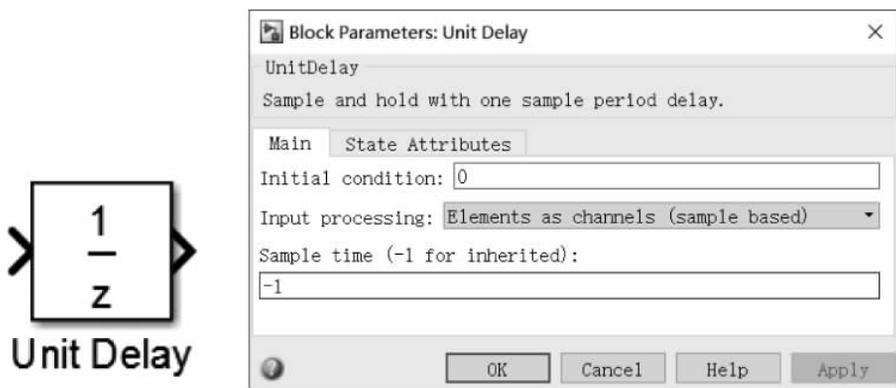


图 3-55 Unit Delay 模块

对于其属性窗口:

(1) Initial condition: 在模块未被定义时,模块的第一个仿真周期按照正常非延迟状态输出,系统默认值为 0。

(2) Input processing: 表示基于采样的元素通道。

(3) Sample time(-1 for inherited): 表示系统采样时间,系统默认值为-1。

搭建 Unit Delay 模块,设置采样时间为 0.1s,如图 3-56 所示。运行仿真文件,输出结果如图 3-57 所示。

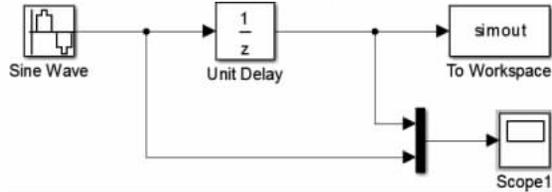


图 3-56 Unit Delay 模块使用

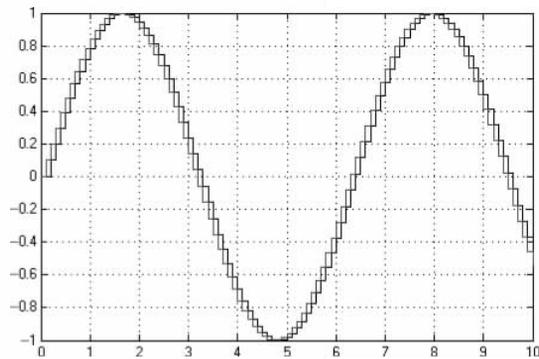


图 3-57 Unit Delay 模块示波器时钟变化图

3.4.4 Memory 模块

Memory 模块将前一个集成步的输入作为输出,相当于对前一个集成步内的输入进行采样-保持。

Memory 的模块属性如图 3-58 所示。

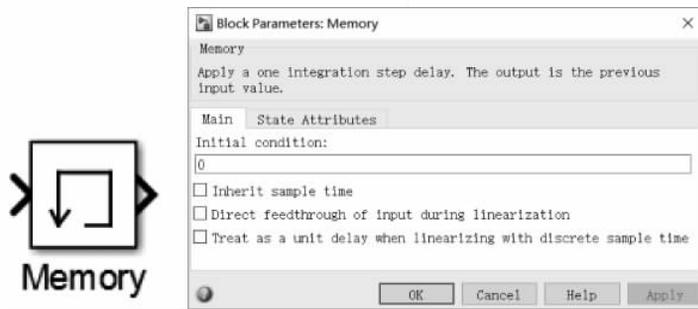


图 3-58 Memory 模块

对于其属性窗口:

(1) Initial condition: 表示系统初始集成步的输出,系统默认值为 0。

(2) Inherit sample time: 系统默认不被选中,若选中该复选框,表示使系统采样时间从驱动模块继承。

搭建 Memory 模块,设置采样时间为 0.1s,如图 3-59 所示。运行仿真文件,输出结果如图 3-60 所示。

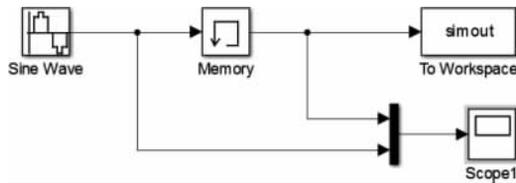


图 3-59 Memory 模块使用

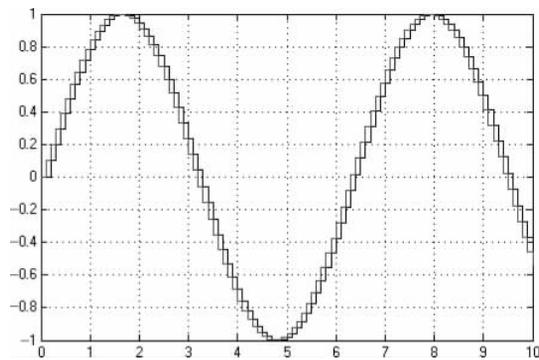


图 3-60 Memory 模块示波器时钟变化图

3.4.5 Discrete Zero-Pole 模块

对于 Discrete Zero-Pole 模块,由通常的拉普拉斯变换后,得到相应的传递函数,再经过 Z 变换,得到离散系统传递函数,具体如下:

$$H(z) = \frac{\text{num}(z)}{\text{den}(z)} = \frac{a_n z^n + a_{n-1} z^{n-1} + \dots + a_0 z^0}{b_m z^m + b_{m-1} z^{m-1} + \dots + b_0 z^0}$$

转化为离散零极点传递函数为

$$H(z) = K \frac{Z(z)}{P(z)} = K \frac{(z - Z_1)(z - Z_2) \dots (z - Z_n)}{(z - P_1)(z - P_2) \dots (z - P_m)}$$

其中,Z 表示零点矢量;P 表示极点矢量;K 表示系统增益。系统要求 $m \geq n$,若极点和零点是复数,它们必须是复共轭对。

Discrete Zero-Pole 的模块属性如图 3-61 所示。

对于其属性窗口:

- (1) Zeros: 表示系统零点矩阵,系统默认值为[1]。
- (2) Poles: 表示系统极点矩阵,系统默认值为[0 0.5]。
- (3) Gain: 表示系统增益,系统默认值为 1。
- (4) Sample time(-1 for inherited): 表示系统采样时间,系统默认值为 1。

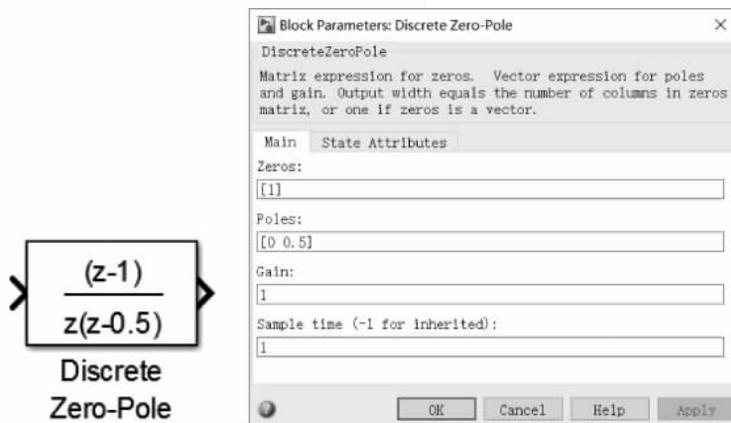


图 3-61 Discrete Zero-Pole 模块

搭建 Discrete Zero-Pole 模块,设置采样时间为 0.1s,如图 3-62 所示。运行仿真文件,输出结果如图 3-63 所示。

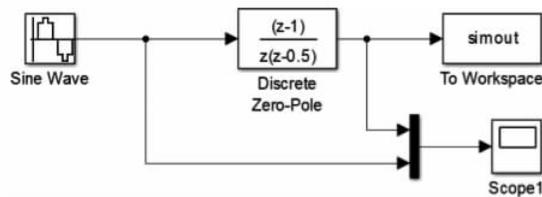


图 3-62 Discrete Zero-Pole 模块使用

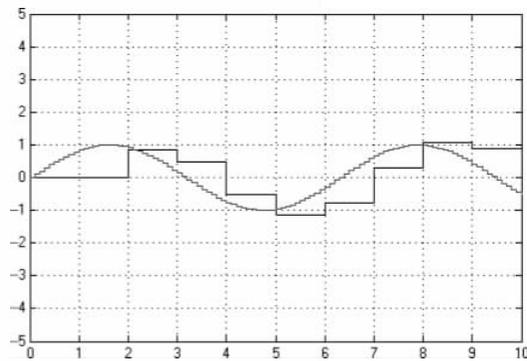


图 3-63 Discrete Zero-Pole 模块示波器时钟变化图

3.4.6 Discrete State-Space 模块

Discrete State-Space 模块可实现如下的离散系统:

$$\begin{cases} x(n+1) = Ax(n) + Bu(n) \\ y(n) = Cx(n) + Du(n) \end{cases}$$

其中, u 为输入; x 为状态; y 为输出。

Discrete State-Space 的模块属性如图 3-64 所示。

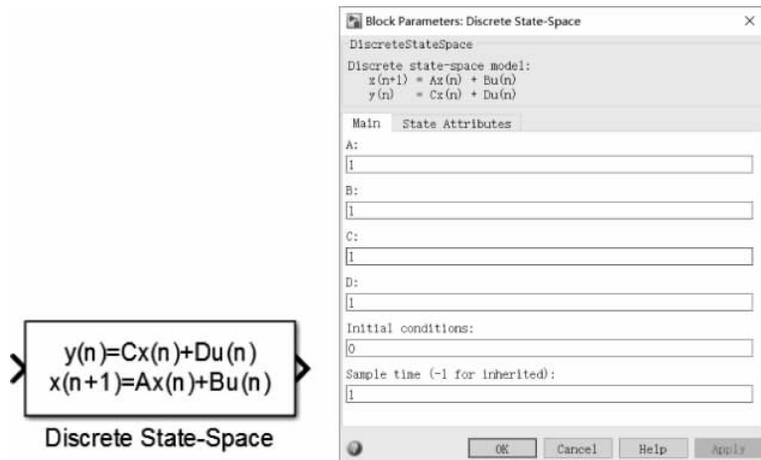


图 3-64 Discrete State-Space 模块

对于其属性窗口：

- (1) A: 表示系统状态空间矩阵系数,必须是一个 $n \times n$ 矩阵, n 为状态数,系统默认值为 1。
- (2) B: 表示系统状态空间矩阵系数,必须是一个 $n \times m$ 矩阵, m 为状态数,系统默认值为 1。
- (3) C: 表示系统状态空间矩阵系数,必须是一个 $r \times n$ 矩阵, r 为状态数,系统默认值为 1。
- (4) D: 表示系统状态空间矩阵系数,必须是一个 $r \times m$ 矩阵,系统默认值为 1。
- (5) Initial conditions: 表示初始状态矢量,系统默认值为 0。
- (6) Sample time(-1 for inherited): 表示系统采样时间,系统默认值为 1。

搭建 Discrete State-Space 模块,设置采样时间为 0.1s,如图 3-65 所示。运行仿真文件,输出结果如图 3-66 所示。

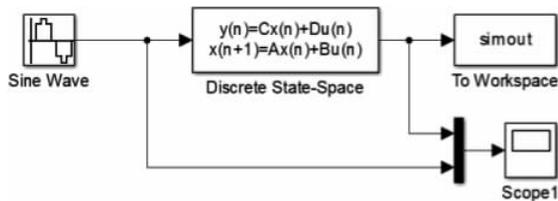


图 3-65 Discrete State-Space 使用

3.4.7 Zero-Order Hold 模块

Zero-Order Hold 模块实现一个以指定采样率的采样与保持函数操作,模块接收一个输入,并产生一个输出,输入和输出可以是标量或矢量。

Zero-Order Hold 的模块属性如图 3-67 所示。

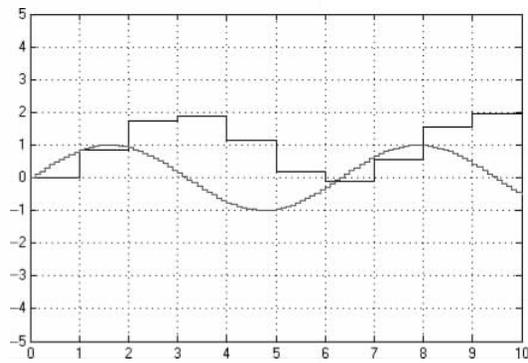


图 3-66 Discrete State-Space 模块示波器时钟变化图

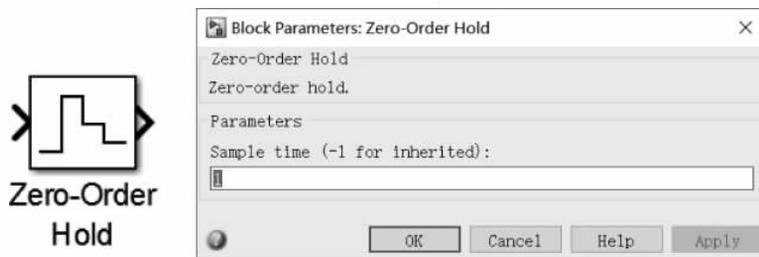


图 3-67 Zero-Order Hold 模块

对于其属性窗口: Sample time(-1 for inherited)表示系统采样时间,系统默认值为 1。
 搭建 Zero-Order Hold 模块,设置采样时间为 0.1s,如图 3-68 所示。运行仿真文件,
 输出结果如图 3-69 所示。

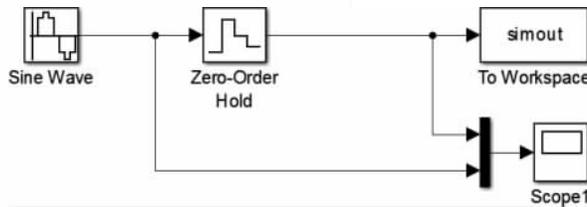


图 3-68 Zero-Order Hold 使用

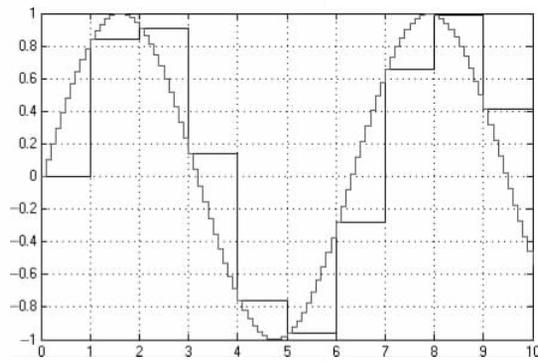


图 3-69 Zero-Order Hold 模块示波器时钟变化图

3.5 查表模块组

MATLAB Simulink 查表模块分为一维查找表模块(1-D Lookup Table)和二维查找表模块(2-D Lookup Table), 主要实现信号的插值功能。

(1) 一维查找表模块: 可实现对单路输入信号的查表和线性插值。

(2) 二维查找表模块: 根据给定的二维平面网格上的高度值, 把输入的两个变量经过查找表、插值, 计算出模块的输出值, 并返回该值。

3.5.1 1-D Lookup Table 模块

一维查找表模块的模块属性如图 3-70 所示。

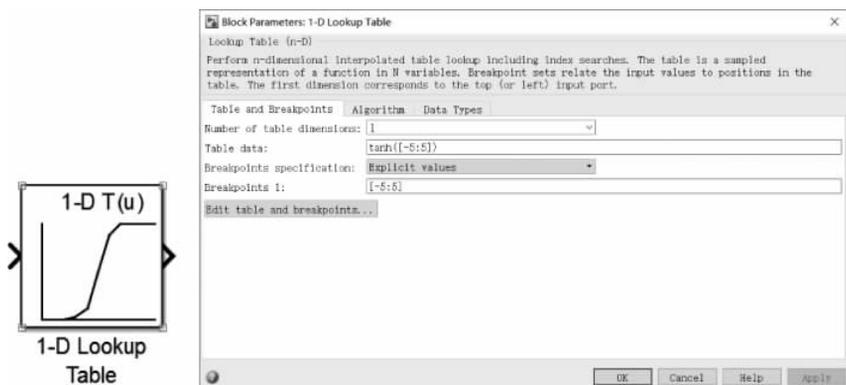


图 3-70 1-D Lookup Table 模块

对于其属性窗口:

(1) Number of table dimensions: 一维查找表模块默认为 1, 表示是一维的查表数据。

(2) Table data: 系统默认为 $\tanh([-5:5])$, 双曲正切函数, 取值范围为 -5 到 5 之间。

搭建 1-D Lookup Table 模块, 设置采样时间为 $0.1s$, 如图 3-71 所示。运行仿真文件, 输出结果如图 3-72 所示。

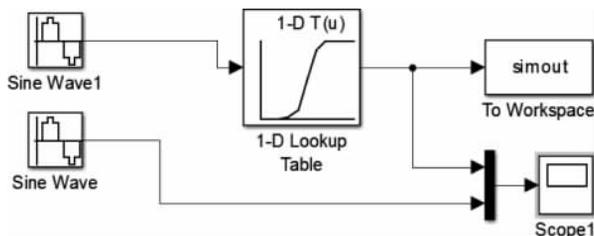


图 3-71 1-D Lookup Table 使用

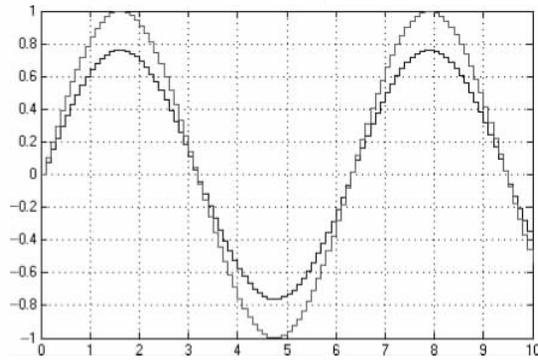


图 3-72 1-D Lookup Table 模块示波器时钟变化图

3.5.2 2-D Lookup Table 模块

2-D Lookup Table 模块的属性如图 3-73 所示。



图 3-73 2-D Lookup Table 模块

对于其属性窗口：

(1) Number of table dimensions: 一维查找表模块默认为 1, 表示一维的查表数据。

(2) Table data: 系统默认为 $\tanh([-5:5])$, 双曲正切函数, 取值范围为 -5 到 5 之间。

搭建 2-D Lookup Table 模块, 设置采样时间为 $0.1s$, 如图 3-74 所示。运行仿真文件, 输出结果如图 3-75 所示。

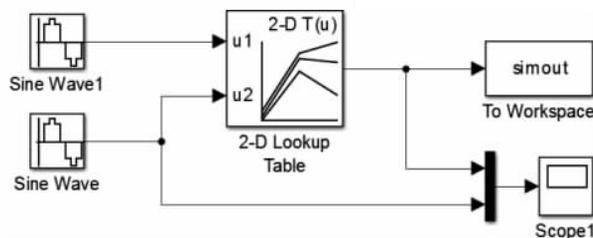


图 3-74 2-D Lookup Table 使用

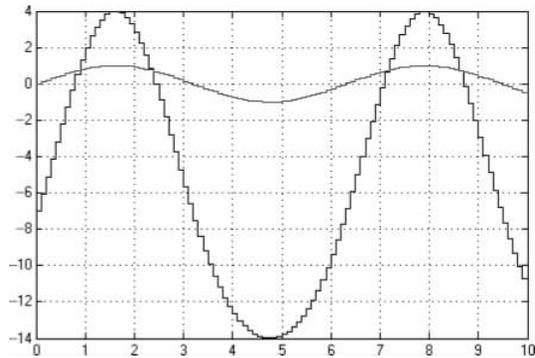


图 3-75 2-D Lookup Table 模块示波器时钟变化图

3.6 用户自定义函数模块组

MATLAB Simulink 提供了用户自定义函数模块,该模块可方便用户设计自己的仿真模型,实现模型的易移植性等特点。

3.6.1 Fcn 模块

Fcn 模块用于实现系统的数学表达式快捷计算, $u(i)$ 表示矢量的第 i 个元素。MATLAB 数学函数包括 abs 、 acos 、 asin 、 cos 、 log 和 tanh 等。模块的输入可以是一个标量或矢量,输出总为标量。

Fcn 的模块属性如图 3-76 所示。

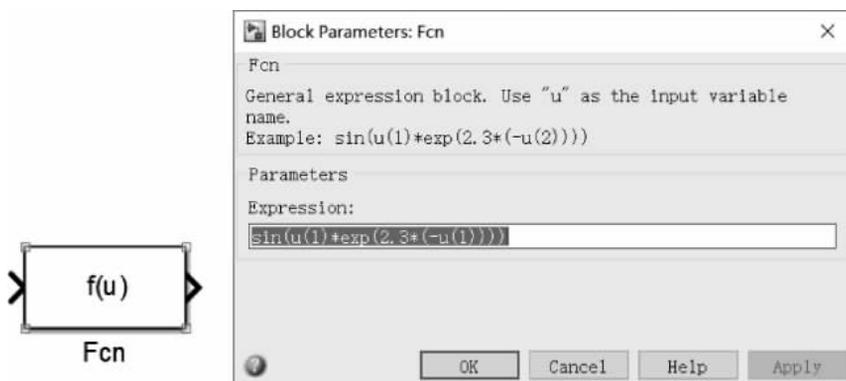


图 3-76 Fcn 模块

对于其属性窗口: Expression 表示系统默认方程式为 $\sin(u(1) * \exp(2.3 * (-u(2))))$,用于函数定义。

搭建 Fcn 模块,设置采样时间为 0.1s,如图 3-77 所示。运行仿真文件,输出结果如图 3-78 所示。

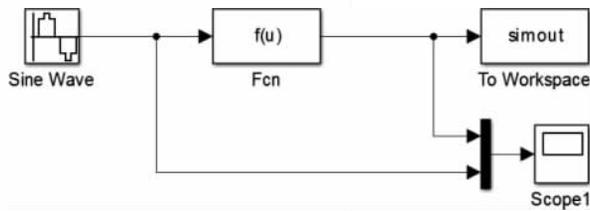


图 3-77 Fcn 使用

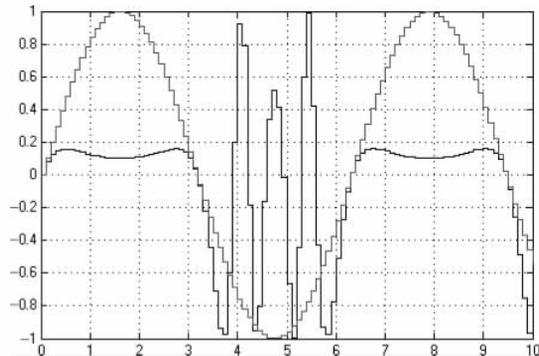


图 3-78 Fcn 模块示波器时钟变化图

3.6.2 MATLAB Fcn 模块

MATLAB Fcn 模块便于用户快速定义自己的函数,且能够完全适应 Fcn 模块。MATLAB Fcn 模块具有较强的程序移植功能,用户可以开发相应的算法,这也是一种嵌入式编程。

MATLAB Fcn 的模块属性如图 3-79 所示。

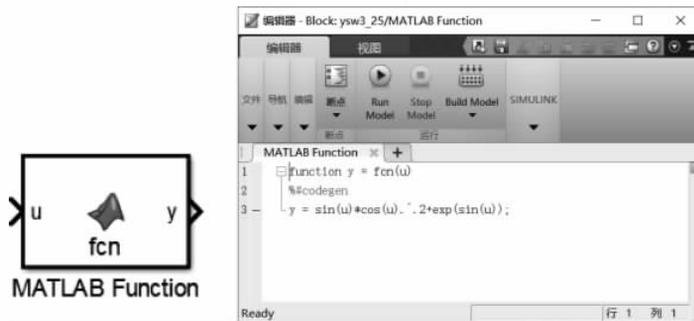


图 3-79 Fcn 模块

如图 3-79 所示 MATLAB Fcn 模块,双击该模块,将弹出其程序编写窗口,用户可以在此窗口下输入如下代码:

```
function y = fcn(u)
%生成代码
y = sin(u) * cos(u).^2 + exp(sin(u));
```

搭建 MATLAB Fcn 模块,设置采样时间为 0.1s,如图 3-80 所示。运行仿真文件,输出结果如图 3-81 所示。

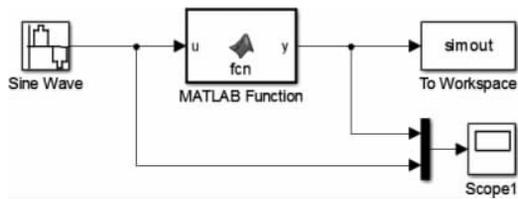


图 3-80 MATLAB Fcn 使用

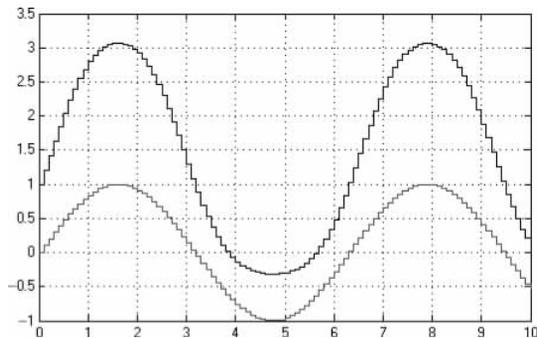


图 3-81 MATLAB Fcn 模块示波器时钟变化图

3.6.3 S-Function 模块

用户可以编写 M 文件供 S-Function 模块调用,需要遵循 S-Function 函数的格式,该模块允许附加参数直接赋给 S-Function。S-Function 有两个端口,一个输入端口,一个输出端口,输入端口的维数可以由用户函数指定,主要以行向量的形式进行输入和输出。

S-Function 的模块属性如图 3-82 所示。

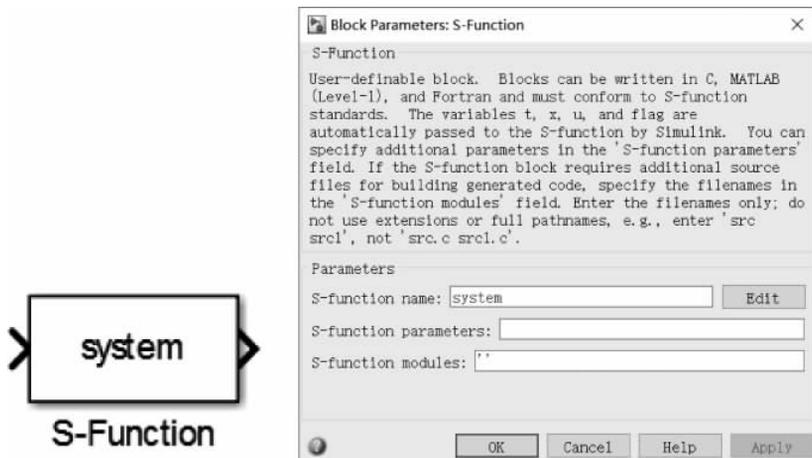


图 3-82 S-Function 模块

对于其属性窗口：

(1) S-Function name: 表示 S-Function 的函数文件名称,单击 Edit 按钮即可打开该函数文件。

(2) S-Function parameters: 表示 S-Function 模块的参数,一般默认为空。

(3) S-Function modules: 表示 S-Function 模块,默认为",一般无须编辑,采用系统默认设置。

采用 PID 控制器对正弦函数进行控制,S-Function 程序如下:

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

switch flag,
case 0,
    [sys,x0,str,ts] = mdlInitializeSizes;
case 1,
    sys = mdlDerivatives(t,x,u);
case 3,
    sys = mdlOutputs(t,x,u);
case {2,4,9}
    sys = [];
otherwise
    error(['Unhandled flag = ',num2str(flag)]);
end

function [sys,x0,str,ts] = mdlInitializeSizes
sizes = simsizes;
sizes.NumContStates = 0;
sizes.NumDiscStates = 0;
sizes.NumOutputs = 1;
sizes.NumInputs = 3;
sizes.DirFeedthrough = 1;
sizes.NumSampleTimes = 1;           % 至少需要一个采样点
sys = simsizes(sizes);
x0 = [];
str = [];
ts = [0 0];
function sys = mdlOutputs(t,x,u)
kp = 10;
ki = 2;
kd = 1;
ut = kp * u(1) + ki * u(2) + kd * u(3);
sys(1) = ut;
```

控制对象 S-Function 程序如下:

```
function [sys,x0,str,ts] = spacemodel(t,x,u,flag)
switch flag,
case 0,
```

```

[sys,x0,str,ts] = mdlInitializeSizes;
case 1,
    sys = mdlDerivatives(t,x,u);
case 3,
    sys = mdlOutputs(t,x,u);
case {2,4,9}
    sys = [];
otherwise
    error(['Unhandled flag = ',num2str(flag)]);
end
function [sys,x0,str,ts] = mdlInitializeSizes
sizes = simsizes;
sizes.NumContStates = 2;
sizes.NumDiscStates = 0;
sizes.NumOutputs = 1;
sizes.NumInputs = 1;
sizes.DirFeedthrough = 0;
sizes.NumSampleTimes = 1;           % 至少需要一个采样点
sys = simsizes(sizes);
x0 = [0;0];
str = [];
ts = [0 0];
function sys = mdlDerivatives(t,x,u) % 时变模型
ut = u(1);
J = 20 + 10 * sin(6 * pi * t);
K = 400 + 300 * sin(2 * pi * t);
sys(1) = x(2);
sys(2) = -J * x(2) + K * ut;
function sys = mdlOutputs(t,x,u)
sys(1) = x(1);

```

搭建 S-Function 模块, 设置采样时间为 0.1s, 如图 3-83 所示。

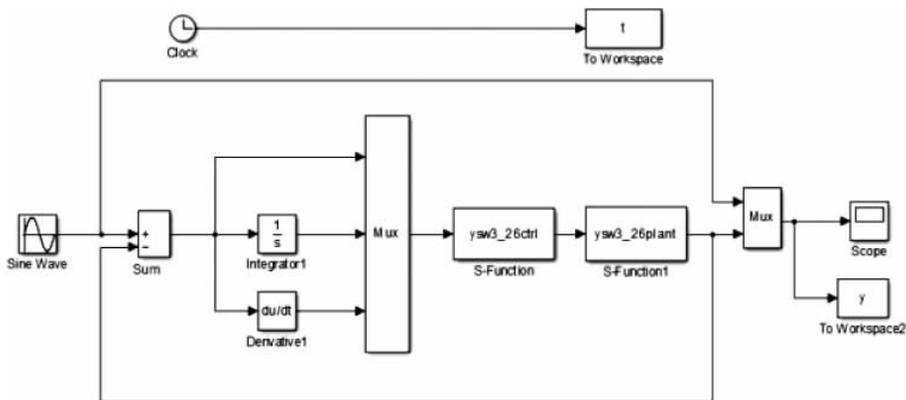


图 3-83 S-Function 使用

运行仿真文件,输出结果如图 3-84 所示。

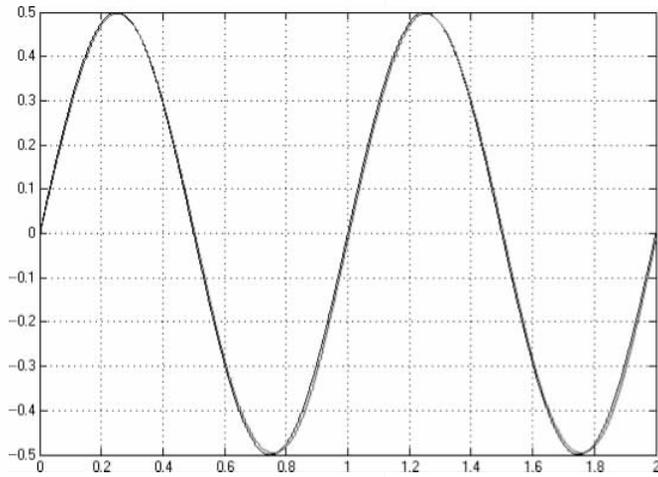


图 3-84 S-Function 模块示波器时钟变化图

3.7 数学运算模块组

数学运算模块主要针对基本运算符号进行模块化设计,用户可以很方便地进行输入信号的加、减、乘、除等基本运算,从而加速模型设计。

3.7.1 Abs 模块

Abs 模块用于绝对值操作,即对输入的矢量或者标量进行取绝对值运算。

Abs 的模块属性如图 3-85 所示。

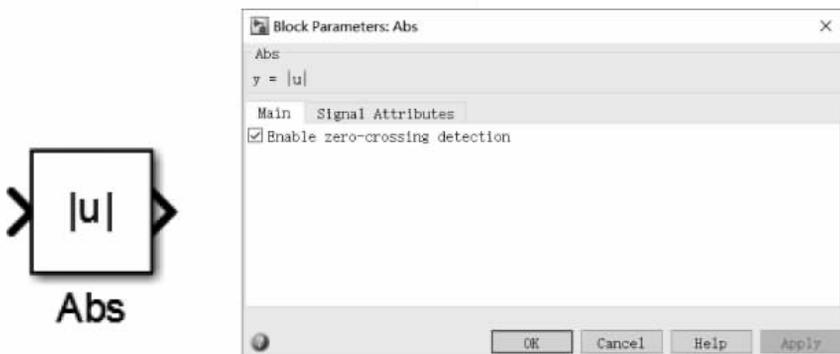


图 3-85 Abs 模块

对于其属性窗口: Enable zero-crossing detection 表示开启模块的过零检测。

搭建 Abs 模块,如图 3-86 所示。运行仿真文件,输出结果如图 3-87 所示。

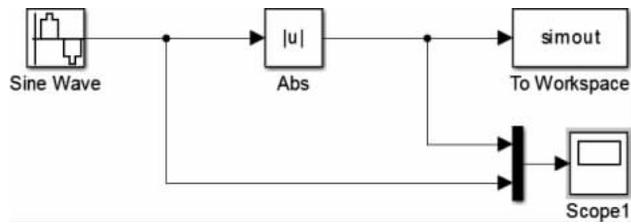


图 3-86 Abs 使用

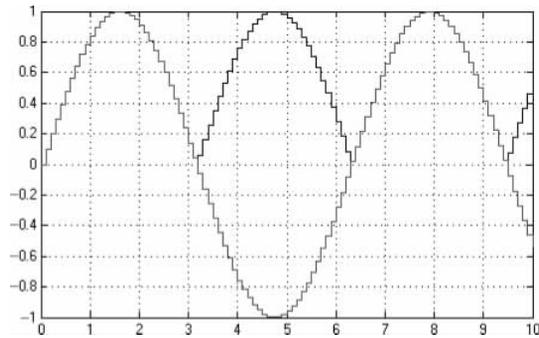


图 3-87 Abs 模块示波器时钟变化图

3.7.2 Add 模块

Add 模块用于加减运算,即对输入的矢量或标量进行加减操作。
Add 的模块属性如图 3-88 所示。

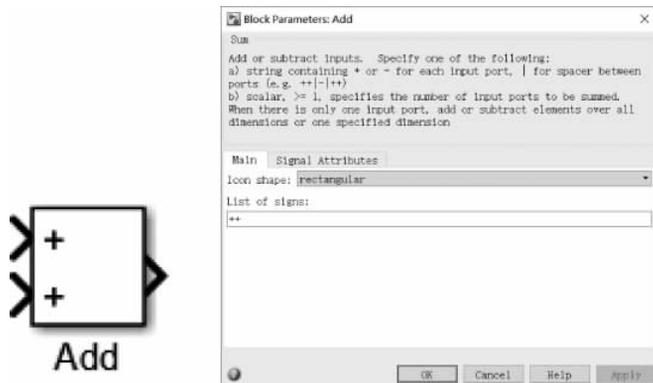


图 3-88 Add 模块

对于其属性窗口: List of signs 表示符号设置,可以设置为“+-”,表示第一个输入为正,第二个输入为负;也可以为“-+”,表示第一个输入为负,第二个输入为正;设置为“++”,表示第一个输入为正,第二个输入为正;设置为“--”,表示第一个输入为负,第二个输入为负;系统默认为“++”。

搭建 Add 模块,如图 3-89 所示。运行仿真文件,输出结果如图 3-90 所示。

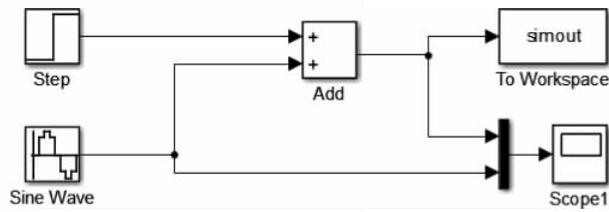


图 3-89 Add 使用

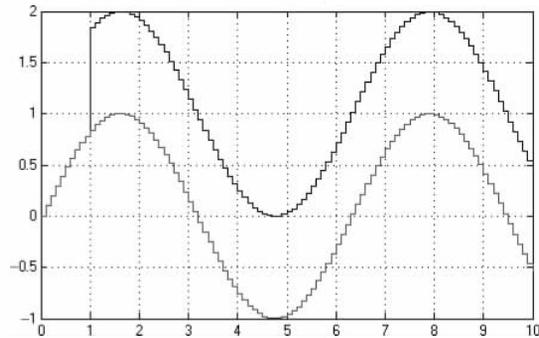


图 3-90 Add 模块示波器时钟变化图

3.7.3 Divide 模块

Divide 模块用于乘除运算,即对输入的矢量或标量进行乘除操作。
Divide 的模块属性如图 3-91 所示。

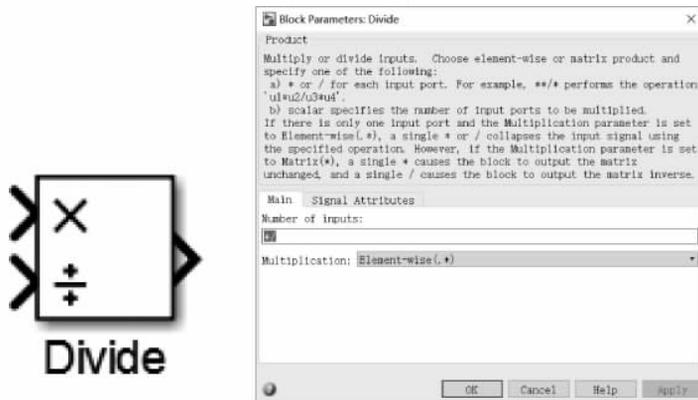


图 3-91 Divide 模块

对于其属性窗口:

(1) Number of inputs: 表示符号设置,可以设置为“*/”,表示第一个输入为分子,第二个输入为分母;也可以设置为“/*”,表示第一个输入为分母,第二个输入为分子;设置为“**”,表示第一个输入为分子,第二个输入为分子,两者直接相乘;设置为“//”,表示第一个输入为分母,第二个输入为分母,两者直接相乘;系统默认为“*/”。

(2) Multiplication: 包括两个选项 Element-wise (.*) 和 Matrix (*)。其中, Element-wise (.*) 表示元素点乘, Matrix (*) 表示矩阵相乘。

搭建 Divide 模块, 如图 3-92 所示。运行仿真文件, 输出结果如图 3-93 所示。

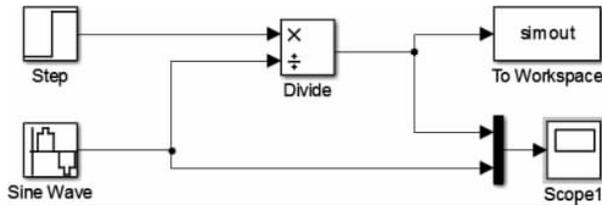


图 3-92 Divide 使用

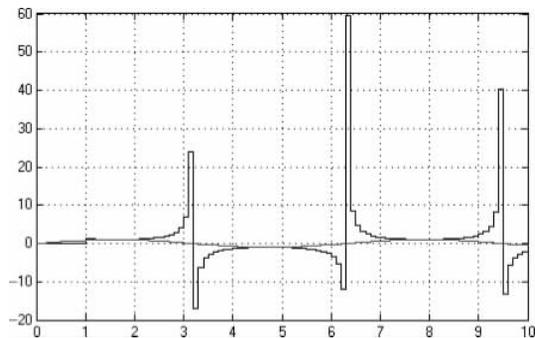


图 3-93 Divide 模块示波器时钟变化图

3.7.4 Dot Product 模块

Dot Product 模块用于点乘运算, 即对输入的矢量或标量进行点乘操作, 是 Simulink 提供的快捷模块。

Dot Product 的模块属性如图 3-94 所示。

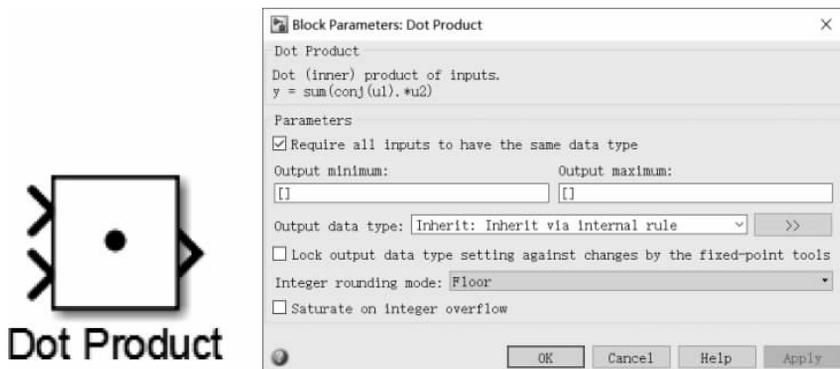


图 3-94 Dot Product 模块

对于其属性窗口:

(1) Output minimum: 指定模块输出的最小值, 默认是[]。

(2) Output maximum: 指定模块输出的最大值,默认是[]。

搭建 Dot Product 模块,如图 3-95 所示。运行仿真文件,输出结果如图 3-96 所示。

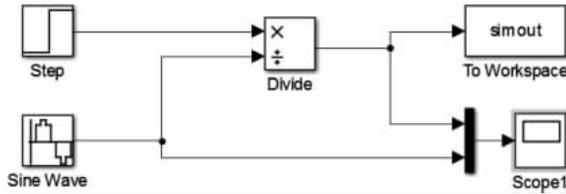


图 3-95 Dot Product 使用

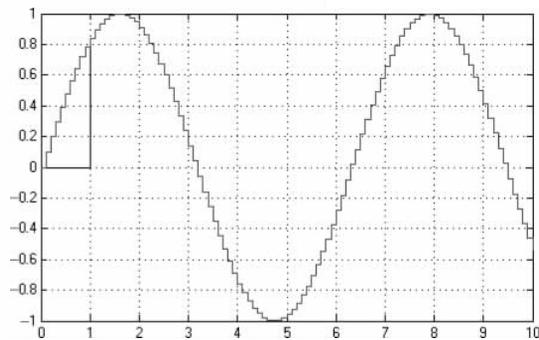


图 3-96 Dot Product 模块示波器时钟变化图

3.7.5 Gain 模块

Gain 模块,即增益模块,用于对输入的矢量或标量乘以放大增益倍数,是 Simulink 提供的快捷模块。增益模块的输入可以为矩阵也可以为向量。

Gain 的模块属性如图 3-97 所示。

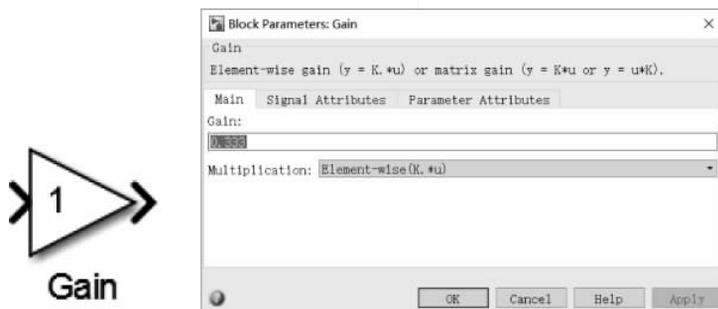


图 3-97 Gain 模块

对于其属性窗口:

(1) Gain: 表示输入的增益数值,可以为矩阵,也可以为数值,对输入的矢量或者标量进行点乘运算,实现放大或者缩小输入量的功能。

(2) Multiplication: 包括两个选项 Element-wise(.*)和 Matrix(*)。其中, Element-wise(.*)表示元素点乘,Matrix(*)表示矩阵相乘。

搭建 Gain 模块,如图 3-98 所示。运行仿真文件,输出结果如图 3-99 所示。

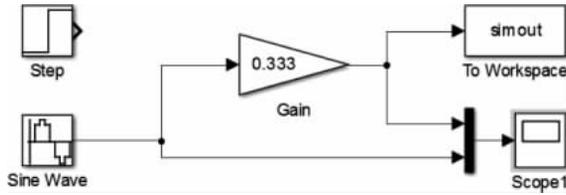


图 3-98 Gain 使用

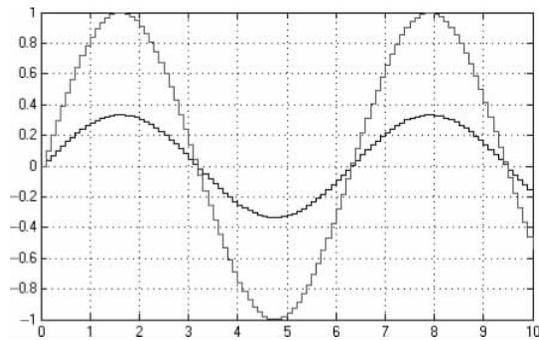


图 3-99 Gain 模块示波器时钟变化图

3.7.6 Complex to Magnitude-Angle 模块

Complex to Magnitude-Angle 模块接受双精度复信号,Complex to Magnitude-Angle 模块输出输入信号的幅值和相角,输入信号可以为矢量或者为标量。

Complex to Magnitude-Angle 的模块属性如图 3-100 所示。

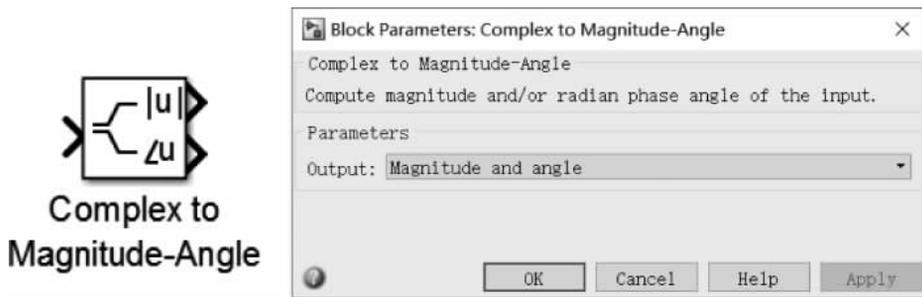


图 3-100 Complex to Magnitude-Angle 模块

对于其属性窗口: Output 输出分为 Magnitude、Angle 和 Magnitude and angle,分别用于输出输入信号的振幅、相角、振幅和相角。

搭建 Complex to Magnitude-Angle 模块,如图 3-101 所示。运行仿真文件,输出结果如图 3-102 所示。

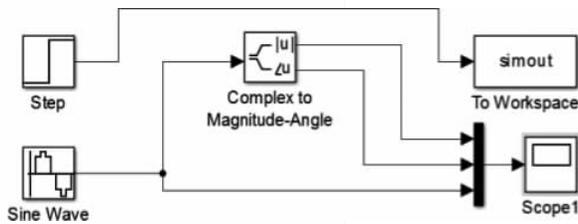


图 3-101 Complex to Magnitude-Angle 使用

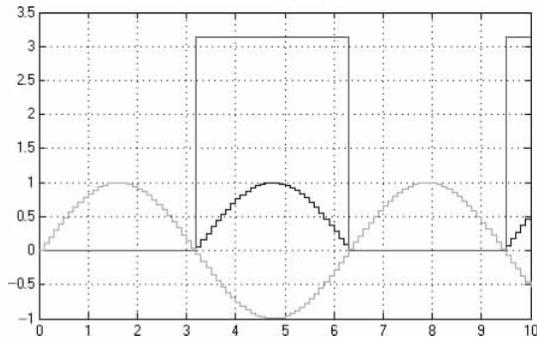


图 3-102 Complex to Magnitude-Angle 模块示波器时钟变化图

3.7.7 Magnitude-Angle to Complex 模块

Magnitude-Angle to Complex 模块的输出信号为双精度复信号。Magnitude-Angle to Complex 模块能将一个幅度和一个相角信号变换为复信号输出,输入信号可以为矢量或标量。如果输入信号是一个标量,则它映射到所有复输出信号的对应成分(幅度或相角)上。

Magnitude-Angle to Complex 的模块属性如图 3-103 所示。

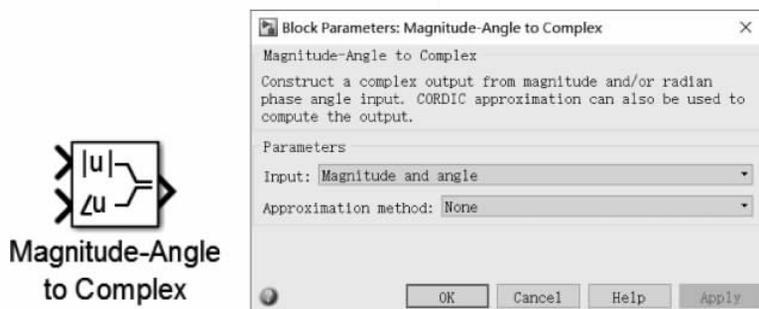


图 3-103 Magnitude-Angle to Complex 模块

对于其属性窗口: Input 输入分为 Magnitude、Angle 和 Magnitude and angle,分别为输入信号的振幅、相角、振幅和相角。

搭建 Magnitude-Angle to Complex 模块,如图 3-104 所示。运行仿真文件,输出结果如图 3-105 所示。

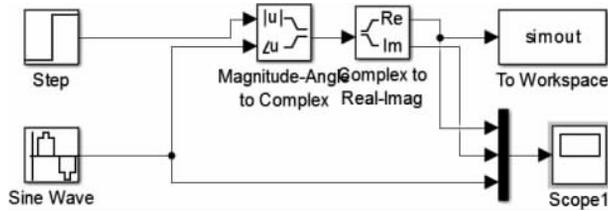


图 3-104 Magnitude-Angle to Complex 使用

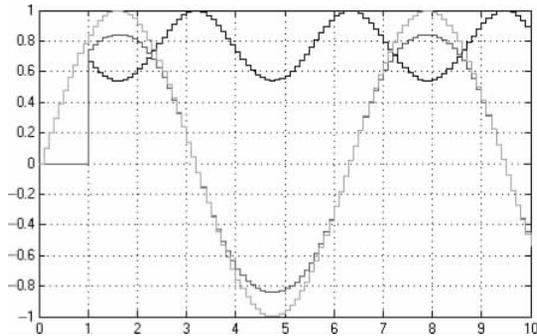


图 3-105 Magnitude-Angle to Complex 模块示波器时钟变化图

3.8 非线性模块组

非线性系统在实际中应用较多,由于理想的线性系统对于仿真控制存在很大的缺陷,因此,Simulink 提供了可供用户使用的非线性模块组。

3.8.1 Backlash 模块

Backlash 模块的主要功能是实现输入和输出变化同步。当输入量改变方向时,输入的初始变化对输出没有影响。

存在回差的系统有如下三种可能:

- (1) 分离模式——输入信号不控制输出,输出保持为常数;
- (2) 正向工作模式——输入以正斜率上升,而输出等于输入减去死区宽度的一半;
- (3) 负向工作模式——输入以负斜率上升,而输出等于输入加上死区宽度的一半。

Backlash 的模块属性如图 3-106 所示。如果初始输入落在死区以外,Initial output 参数值将决定模块是正向工作还是负向工作,并且决定在仿真开始时的输出是输入加上死区宽度的一半还是减去死区宽度的一半。

对于其属性窗口:

- (1) Deadband width: 表示死区宽度,系统默认为 1。
- (2) Initial output: 表示初始输出值,默认值为 0。
- (3) Initial processing: 设置为 Elements as channels (sample based),表示以数值元素进行输入输出。

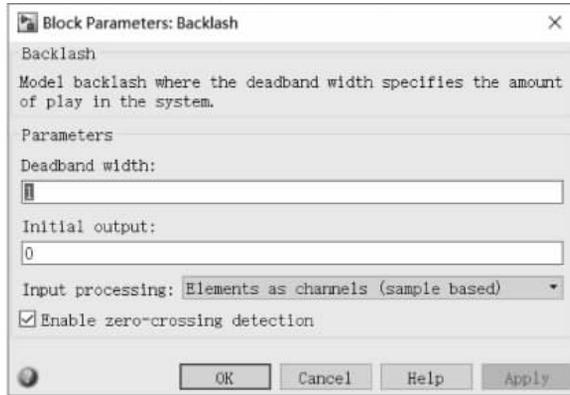
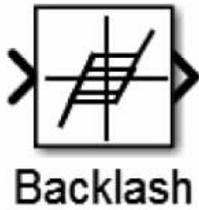


图 3-106 Backlash 模块

搭建 Backlash 模块,如图 3-107 所示。运行仿真文件,输出结果如图 3-108 所示。

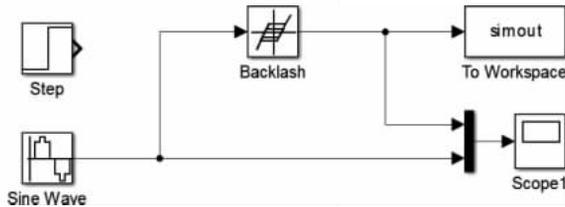


图 3-107 Backlash 使用

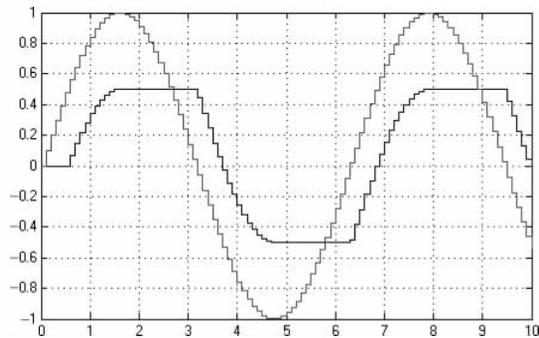


图 3-108 Backlash 模块示波器时钟变化图

3.8.2 Coulomb & Viscous Friction 模块

Coulomb & Viscous Friction 模块用于建立库仑力和粘滞力模型。该模块建立的是在零点不连续而其余点线性的增益模型。偏置对应库仑力;增益对应粘滞力。该模块由如下的函数表达式表示:

$$y = \text{sign}(u) \cdot (\text{Gain} \cdot |u| + \text{offset})$$

其中, y 是输出, u 是输入, Gain 和 offset 为模块参数。

Coulomb & Viscous Friction 的模块属性如图 3-109 所示。

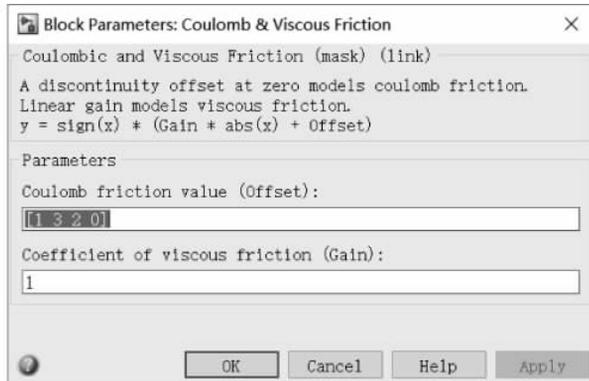


图 3-109 Coulomb & Viscous Friction 模块

对于其属性窗口：

(1) Coulomb friction value(Offset)：表示偏置,适应所有的输入,系统默认值为 [1 3 2 0]。

(2) Coefficient of viscous friction(Gain)：表示在非零输入点的信号增益,系统默认值为 1。

搭建 Coulomb & Viscous Friction 模块,如图 3-110 所示。运行仿真文件,输出结果如图 3-111 所示。

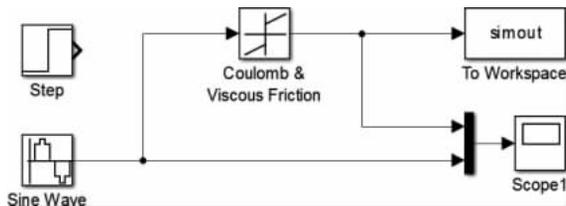


图 3-110 Coulomb & Viscous Friction 使用

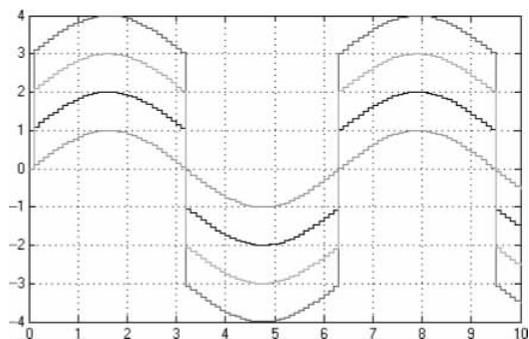


图 3-111 Coulomb & Viscous Friction 模型示波器时钟变化图

3.8.3 Dead Zone 模块

Dead Zone 模块产生指定范围(称为截止区)内的零输出。用 Start of dead zone 和

End of dead zone 参数指定截止区的上下限值。该模块的输入和输出的关系如下：

- (1) 如果输入落在截止区域内,则输出为 0;
- (2) 如果输入大于等于上限值,则输出为上限值;
- (3) 如果输入小于等于下限值,则输出为下限值。

Dead Zone 的模块属性如图 3-112 所示。

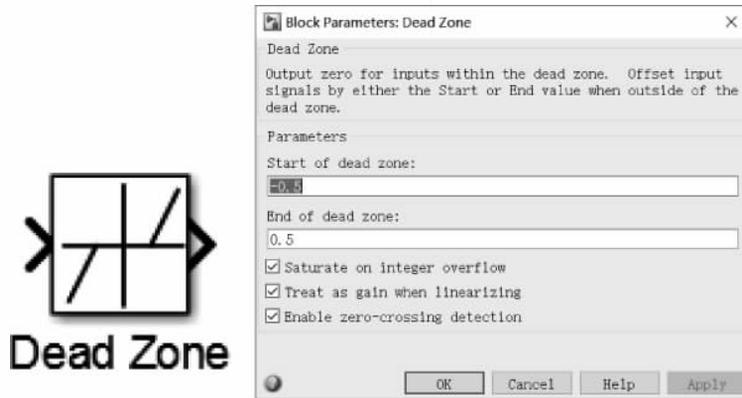


图 3-112 Dead Zone 模块

对于其属性窗口：

- (1) Start of dead zone: 表示下限值,系统默认为-0.5。
- (2) End of dead zone: 表示上限值,系统默认为 0.5。

搭建 Dead Zone 模块,如图 3-113 所示。运行仿真文件,输出结果如图 3-114 所示。

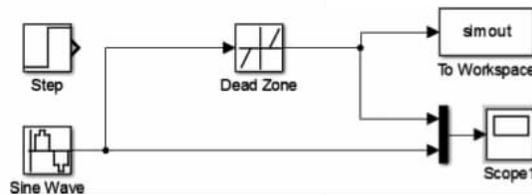


图 3-113 Dead Zone 使用

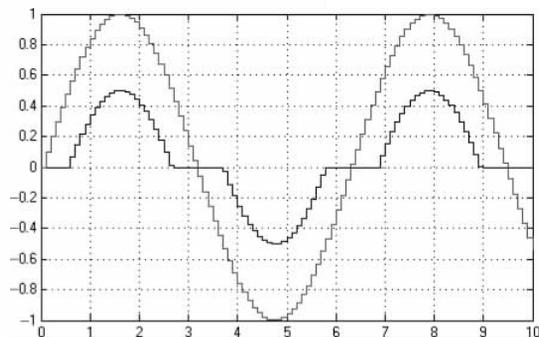


图 3-114 Dead Zone 模块示波器时钟变化图

3.8.4 Quantizer 模块

Quantizer 模块是量化输入的模块,用于将平滑的输入信号变为阶梯状输出。模块接收并输出双精度信号,输出计算采用四舍五入法,产生与零点对称的输出,具体如下:

$$y = q \cdot \text{round}(u/q)$$

其中, u 为一个整数; q 为 Quantization interval 参数,系统默认值为 0.5。

Quantizer 的模块属性如图 3-115 所示。

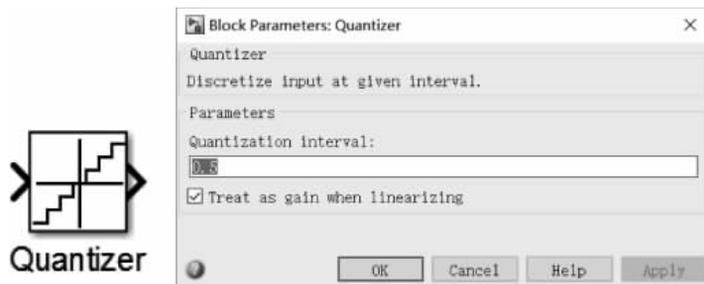


图 3-115 Quantizer 模块

对于其属性窗口: Quantization interval 表示量化输出的时间间隔。Quantizer 模块的输出允许值为 $n \times q$, 其中, n 为一个整数, q 为 Quantization interval 参数, 系统默认值为 0.5。

搭建 Quantizer 模块, 如图 3-116 所示。运行仿真文件, 输出结果如图 3-117 所示。

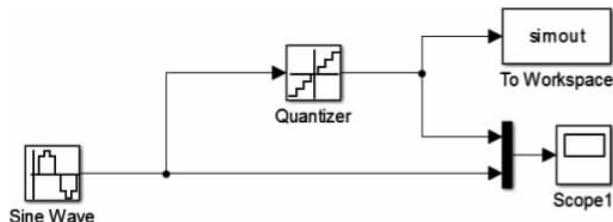


图 3-116 Quantizer 使用

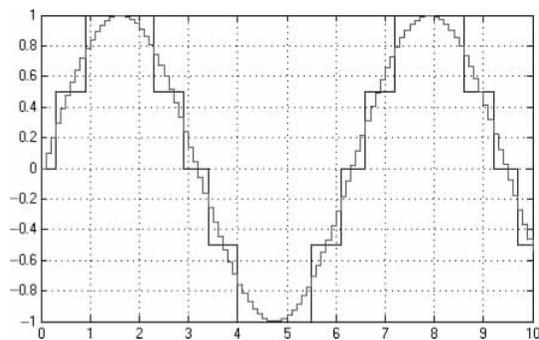


图 3-117 Quantizer 模块示波器时钟变化图

3.8.5 Rate Limiter 模块

Rate Limiter 模块限定通过该模块的信号的一阶导数,以使输出端的变化不超过指定界限,导数根据以下方程计算得到:

$$\text{rate} = \frac{u(i) - y(i-1)}{t(i) - t(i-1)}$$

其中, $u(i)$ 和 $t(i)$ 为当前模块的输入和时间, $y(i-1)$ 和 $t(i-1)$ 为前一时间的输出和时间,输出通过将 rate 与 Rising slew rate 和 Falling slew rate 参数比较得出:

(1) 如果 rate 大于 Rising slew rate 参数(R),输出计算为

$$y(i) = \Delta t \cdot R + y(i-1)$$

(2) 如果 rate 小于 Falling slew rate 参数(F),输出计算为

$$y(i) = \Delta t \cdot F + y(i-1)$$

(3) 如果 rate 大于 Falling slew rate 参数(F),且小于 Rising slew rate 参数(R),输出计算为

$$y(i) = u(i)$$

Rate Limiter 的模块属性如图 3-118 所示。

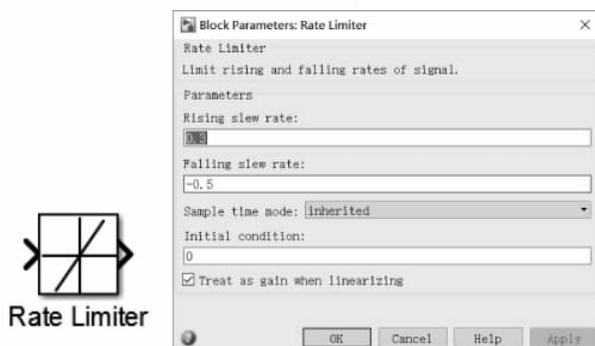


图 3-118 Rate Limiter 模块

对于其属性窗口:

- (1) Rising slew rate: 表示一个递增输入信号的导数极限,默认为 1。
- (2) Falling slew rate: 表示一个递减输入信号的导数极限,默认为 -1。
- (3) Initial condition: 表示系统初始化状态值,默认为 0。

搭建 Rate Limiter 模块,如图 3-119 所示。运行仿真文件,输出结果如图 3-120 所示。

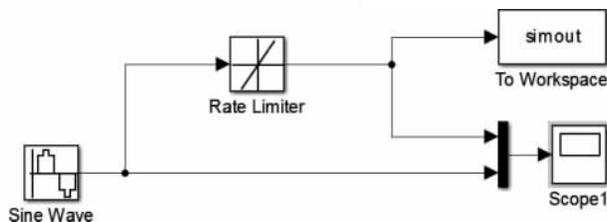


图 3-119 Rate Limiter 使用

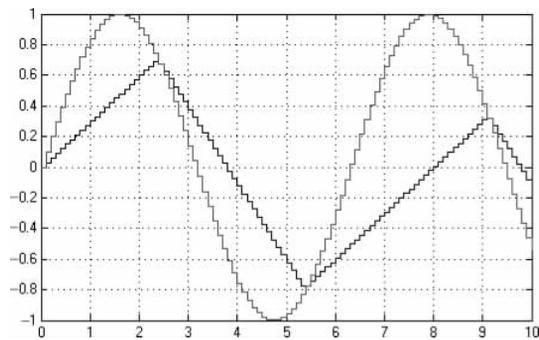


图 3-120 Rate Limiter 模块示波器时钟变化图

3.8.6 Saturation 模块

Saturation 模块用于对输入信号的上限、下限进行约束,如输入值大于等于上限;则取上限值,如输入值小于等于下限,则取下限值。

Saturation 的模块属性如图 3-121 所示。

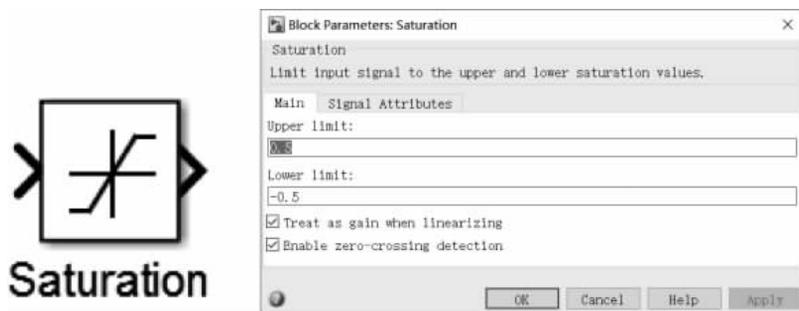


图 3-121 Saturation 模块

对于其属性窗口:

(1) Upper limit: 用于限定输入信号的上限,如输入值大于等于该值,则取该值,系统默认值为 0.5。

(2) Lower limit: 用于限定输入信号的下限,如输入值小于等于该值,则取该值,系统默认值为 -0.5。

搭建 Saturation 模块,如图 3-122 所示。运行仿真文件,输出结果如图 3-123 所示。

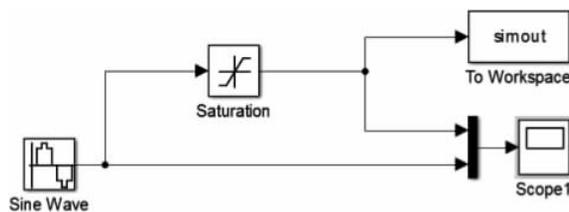


图 3-122 Saturation 使用

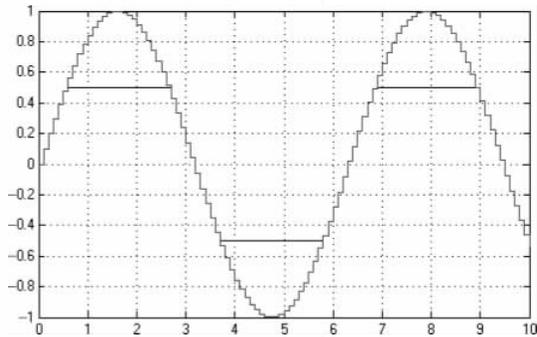


图 3-123 Saturation 模块示波器时钟变化图

3.9 信号与系统模块组

信号与系统模块主要对信号进行仿真运算,在信号系统中应用广泛,例如总线设置、数据存储、数据写和数据读操作等。Simulink 库涵盖范围广,因此适用于多学科的交叉运算。

3.9.1 Bus Selector 模块

Bus Selector 模块接受来自 mux 模块或者其他 Bus Selector 模块的信号, Bus Selector 模块只有一个输入端口,输出端口的数量取决于 Muxed output 复选框的状态。

Bus Selector 的模块属性如图 3-124 所示。



图 3-124 Bus Selector 模块

对于其属性窗口:

(1) Signals in the bus: 此列表框显示在输入母线上的信号。

(2) Selected signals: 此列表框显示输出信号,可以通过 Up、Down 和 Remove 按钮进行信号的上下移动和删除,如果在 Selected signals 列表选中的输出信号不是 Bus

Selector 模块的输入,则信号前将以“???”显示。

搭建 Bus Selector 模块,如图 3-125 所示。运行仿真文件,输出结果如图 3-126 所示。

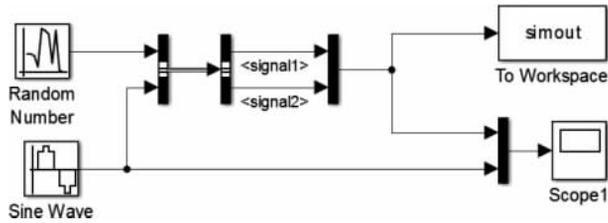


图 3-125 Bus Selector 使用

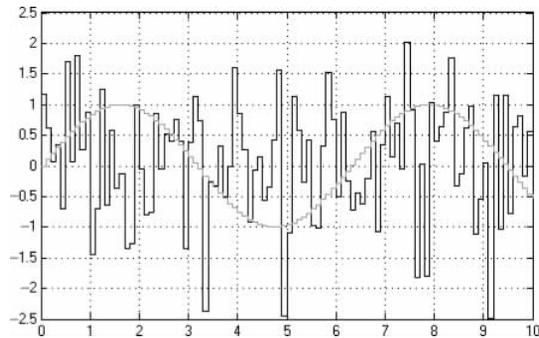


图 3-126 Bus Selector 模块示波器时钟变化图

3.9.2 Bus Creator 模块

Bus Creator 模块的输入信号可以是矢量或标量信号, Bus Creator 创建 Bus 输出信号,可供其他 Bus 模块调用。

Bus Creator 的模块属性如图 3-127 所示。

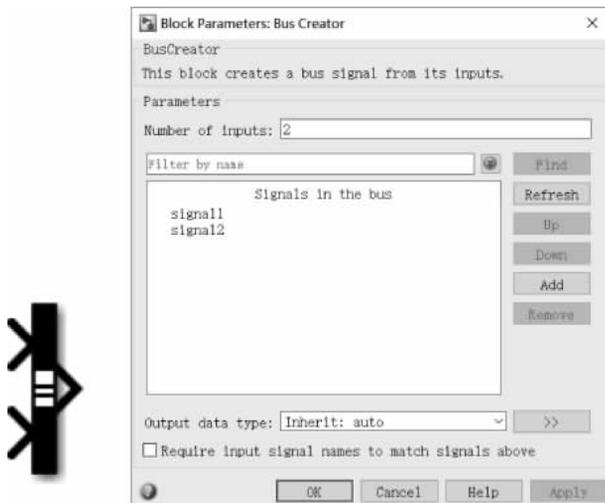


图 3-127 Bus Creator 模块

对于其属性窗口：

(1) Number of inputs: 表示输入信号的个数。

(2) Secreted signals: 此列表框显示输入信号,可以通过 Up、Down、Add 和 Remove 按钮进行信号的上下移动、增加信号和删除信号,如果在 Secreted signals 列表选中的输出信号不是 Bus Selector 模块的输入,则信号前将以“???”显示。

搭建 Bus Creator 模块,如图 3-128 所示。运行仿真文件,输出结果如图 3-129 所示。

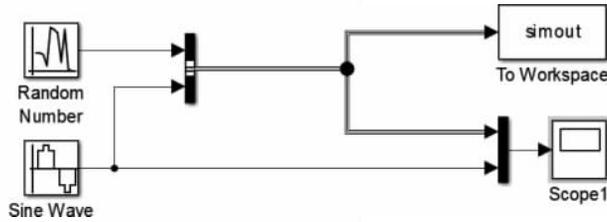


图 3-128 Bus Creator 使用

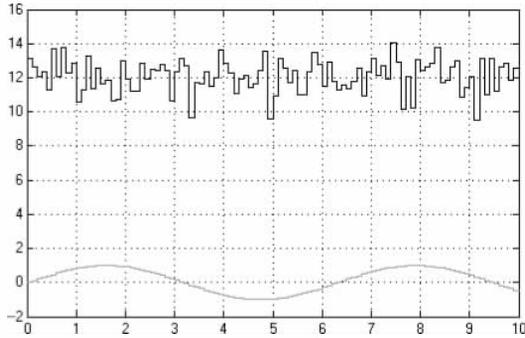


图 3-129 Bus Creator 示波器时钟变化图

3.9.3 Mux 模块

Mux 模块将多个输入行合成为一个矢量行输出。每一个输入行可携带一个标量或矢量信号,模块输出为一个矢量。

Mux 的模块属性如图 3-130 所示。

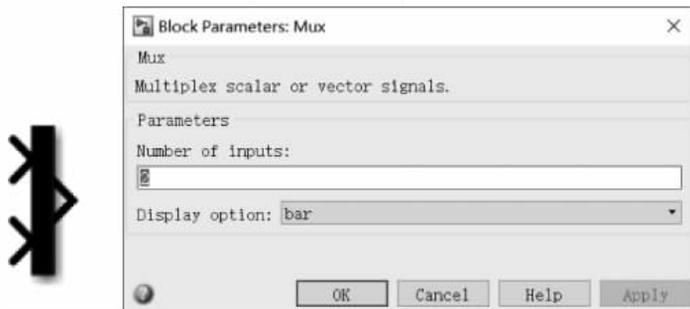


图 3-130 Mux 模块

对于其属性窗口：

(1) Number of inputs: 表示输入信号的个数或者宽度。行输出的宽度等于行输入宽度之和。

(2) Display option: 主要有三个选项 none、names 和 bar。none 表示 Mux 显示在模块图标的外观, names 表示在每一个端口显示信号名, bar 表示以实心前景色显示模块图标。搭建 Mux 模块, 如图 3-131 所示。运行仿真文件, 输出结果如图 3-132 所示。

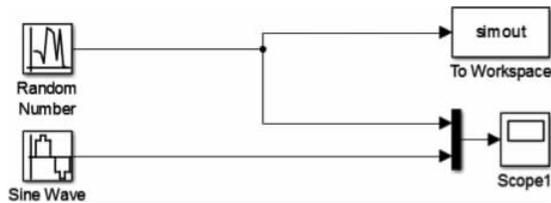


图 3-131 Mux 使用

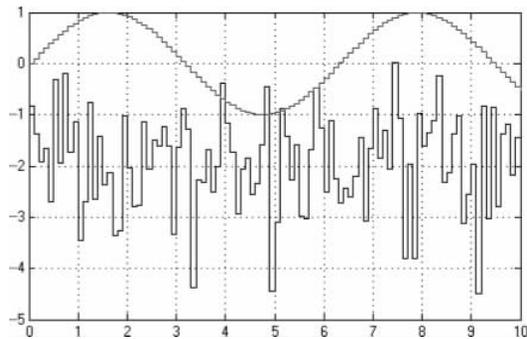


图 3-132 Mux 模块示波器时钟变化图

3.9.4 Demux 模块

Demux 模块将一个输入信号分成为多个行输出, 每一行可包含一个标量或矢量信号, Simulink 通过 Number of outputs 参数决定输出信号的行数或宽度。

Demux 的模块属性如图 3-133 所示。

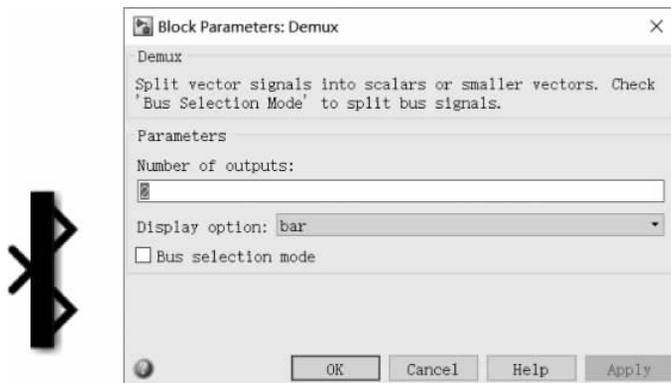


图 3-133 Demux 模块

对于其属性窗口：

(1) Number of outputs: 表示输出信号的个数或者宽度。行输出的总宽度之和等于行输入宽度。

(2) Display option: 主要有三个选项: none、names 和 bar。none 表示 Mux 显示在模块图标的外观, names 表示在每一个端口显示信号名, bar 表示以实心前景色显示模块图标。

搭建 Demux 模块, 如图 3-134 所示。运行仿真文件, 输出结果如图 3-135 所示。

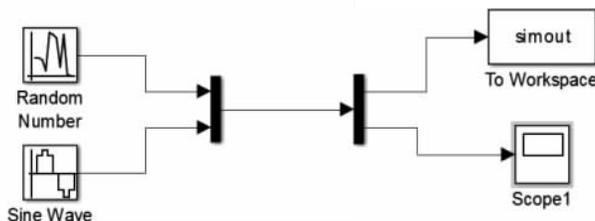


图 3-134 Demux 使用

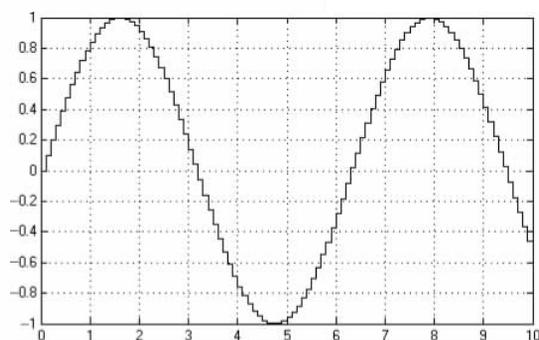


图 3-135 Demux 示波器时钟变化图

3.9.5 Data Store Memory 模块

Data Store Memory 模块用于定义共享数据存储区, 该存储区是与 Data Store Read 模块和 Data Store Write 模块共享的存储空间。

(1) 若 Data Store Memory 模块是在最高一级的系统中, 则处于模型中任何位置的 Data Store Read 模块和 Data Store Write 模块都可以访问该数据存储区。

(2) 若 Data Store Memory 模块处于子系统中, 并且 Data Store Read 和 Data Store Write 模块也位于该子系统或位于子系统的模型分层结构的下级子系统中, 则也能访问该数据存储区。

Data Store Memory 的模块属性如图 3-136 所示。

对于其属性窗口：

(1) Data Store name: 表示正在定义的数据存储区的名字, 系统默认值为字母 A。

(2) Initial value: 系统设定初始值为 0, 系统默认值为 0。

(3) Signal type: 通常仿真中需要指定, 分为实数 real、自动 auto 和复数 complex。

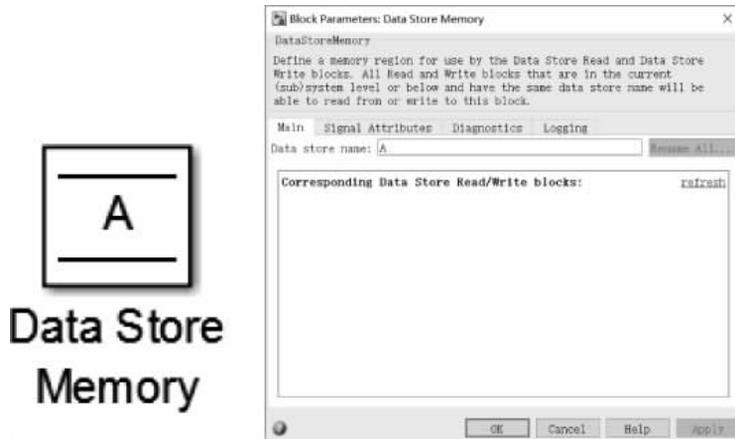


图 3-136 Data Store Memory 模块

(4) Data type: 通常仿真中需要指定, 分为 double、auto、uint8、single、uint16、uint32、boolean、fixdt(1,16) 和 fixdt(1,16,0) 等。

搭建 Data Store Memory 模块, 如图 3-137 所示。运行仿真文件, 输出结果如图 3-138 所示。

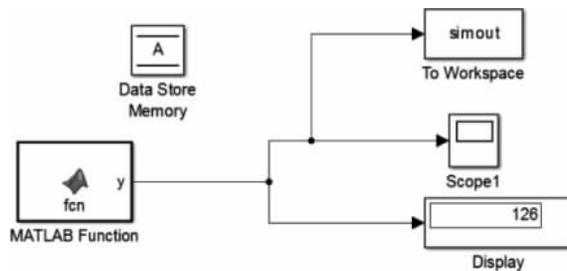


图 3-137 Data Store Memory 使用

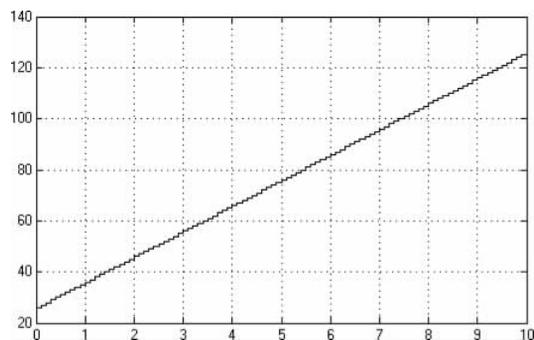


图 3-138 Data Store Memory 模块示波器时钟变化图

3.9.6 Data Store Read 模块

Data Store Read 模块从已经定义的一个共享数据存储区 Data Store Memory 模块

中读取数值,Data Store Read 模块和 Data Store Write 模块与 Data Store Memory 模块共享数据存储空间。

Data Store Read 的模块属性如图 3-139 所示。

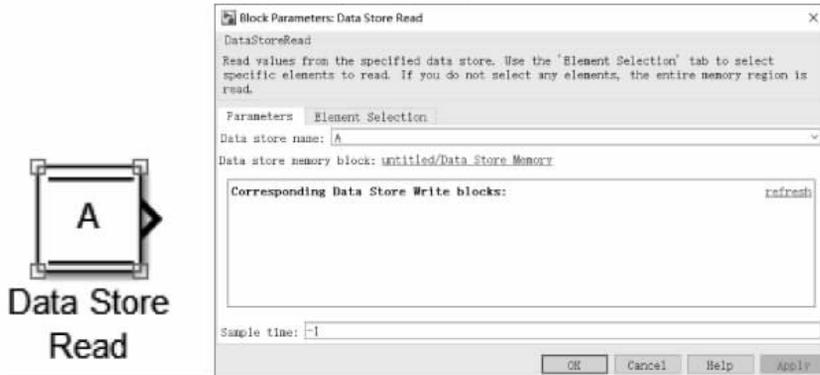


图 3-139 Data Store Read 模块

对于其属性窗口: Data store name 表示正在定义的数据存储区的名字,默认值为字母 A。

搭建 Data Store Read 模块,如图 3-140 所示。运行仿真文件,输出结果如图 3-141 所示。

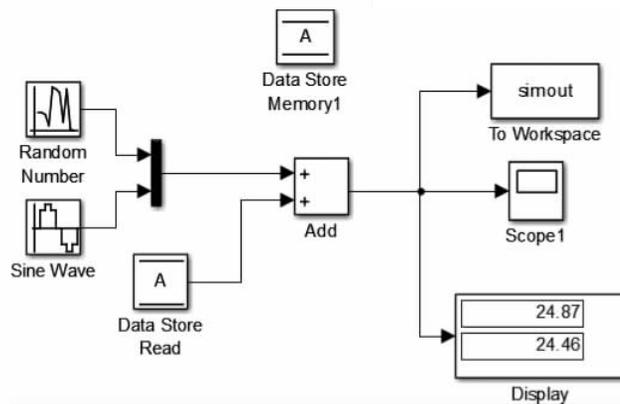


图 3-140 Data Store Read 使用

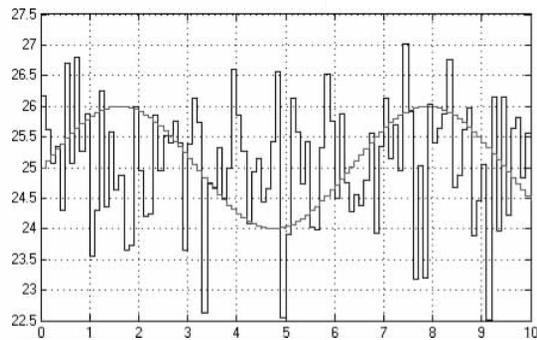


图 3-141 Data Store Read 模块示波器时钟变化图

3.9.7 Data Store Write 模块

Data Store Write 模块定义一个共享数据存储区 Data Store Memory 模块,将输入的数据源写入数值,并将该数值用 Data Store Read 读出和显示。Data Store Write 模块和 Data Store Read 模块与 Data Store Memory 模块共享数据存储空间。

Data Store Write 的模块属性如图 3-142 所示。

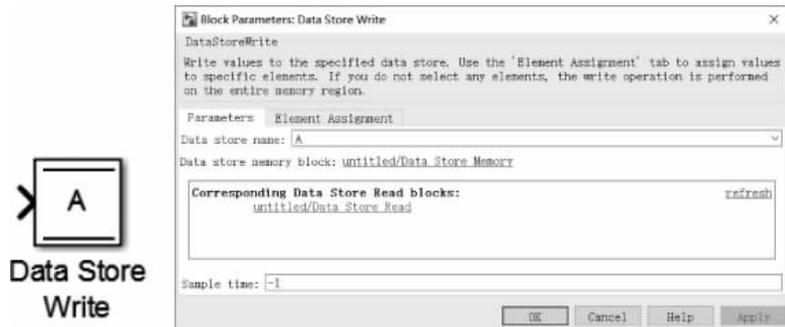


图 3-142 Data Store Write 模块

对于其属性窗口: Data Store name 表示正在定义的数据存储区的名字,默认值为字母 A。搭建 Data Store Write 模块,如图 3-143 所示。

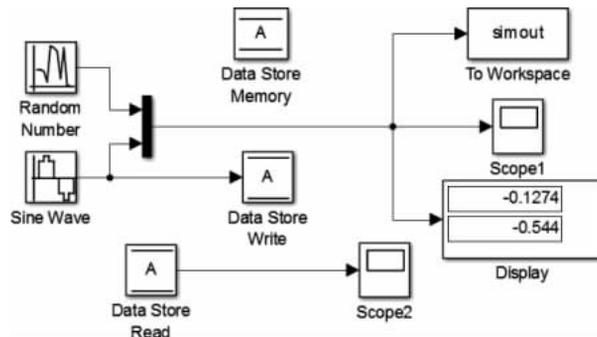


图 3-143 Data Store Write 使用

运行仿真文件,输出结果如图 3-144 所示。

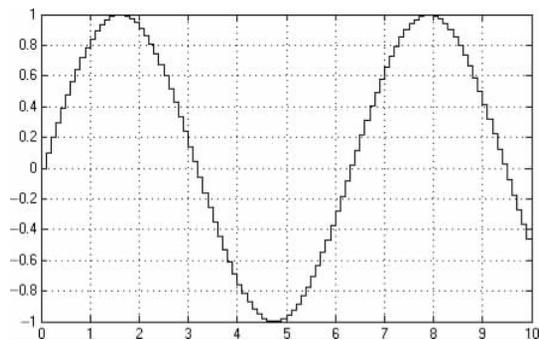


图 3-144 Data Store Write 模块示波器时钟变化图

3.9.8 Enable 模块

加上 Enable 模块的子系统就成为“使能(激活)子系统”,只有当进入 Enable 端口的输入大于 0 时,这种子系统才运行。

仿真运行时,Simulink 按照初始条件将包含在使能子系统内的模块初始化,当一个使能子系统被激活而再启动时,States when enabling 参数决定该子系统内模块的状态。

Enable 的模块属性如图 3-145 所示。

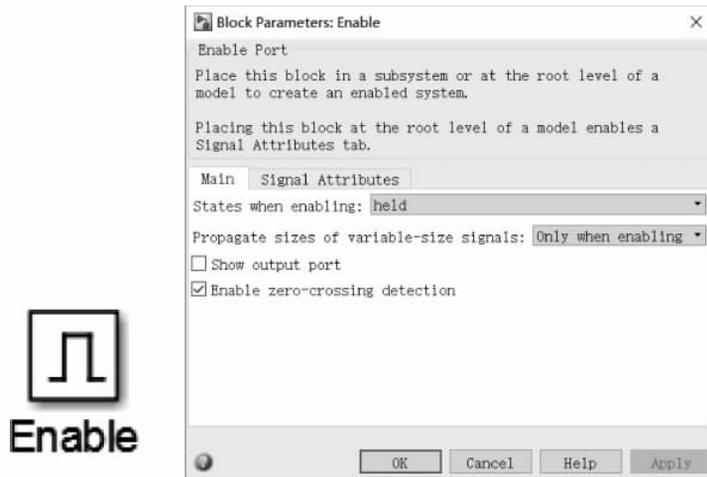


图 3-145 Enable 模块

对于其属性窗口:

(1) States when enabling: 指定当子系统再次被激活时,处理状态的方式。设置为 Reset,表示按照初始条件设置状态,若不知道初始条件,则设置为 0; 设为 held 表示保持原有状态。

(2) Show output port: 若选中该选项,Simulink 给 Enable 模块划分一个输出端口并输出使能信号。

搭建 Enable 模块,如图 3-146 所示。相应的子系统如图 3-147 所示。

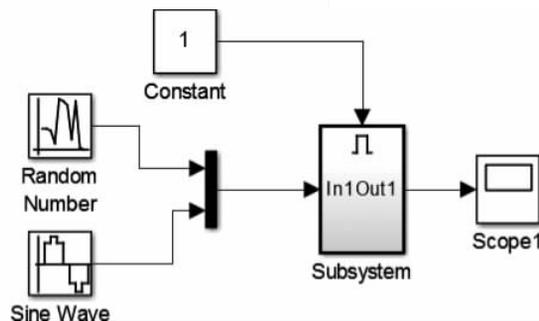


图 3-146 Enable 使用

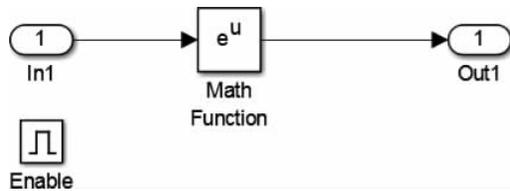


图 3-147 Enable 子系统

运行仿真文件,输出结果如图 3-148 所示。

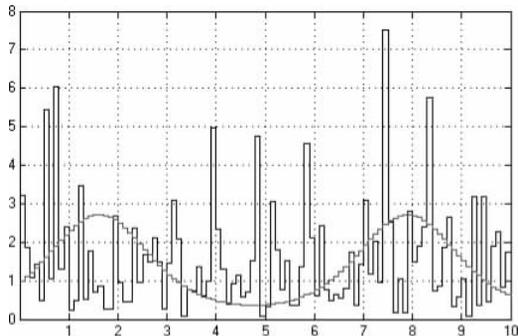


图 3-148 Enable 模块示波器时钟变化图

3.9.9 Ground 模块

Ground 模块可用于链接那些输入端口未与其他模块相连的模块,若用户运行一个带有这样的模块的模型,则 Simulink 就会发布警告。若使用该 Ground 模块,将这些模块“接地”,可避免警示出现,Ground 模块输出 0 值信号。

Ground 模块的输入类型和其他模块的数据类型相同。

Ground 的模块属性如图 3-149 所示。

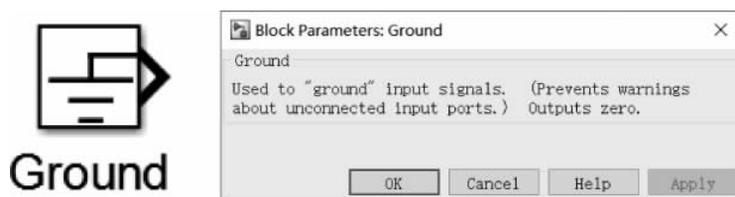


图 3-149 Ground 模块

搭建 Ground 模块,如图 3-150 所示。运行仿真文件,输出结果如图 3-151 所示。

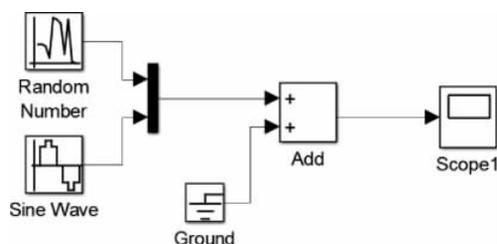


图 3-150 Ground 使用

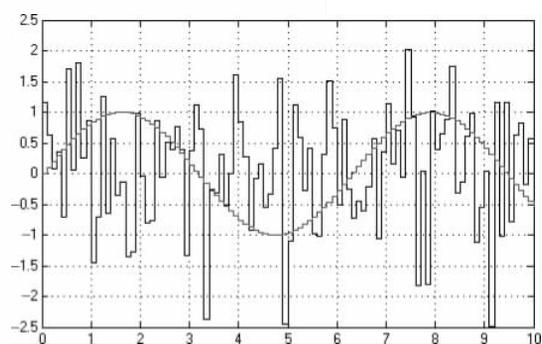


图 3-151 Ground 模块示波器时钟变化图

3.10 本章小结

本章主要介绍了 Simulink 各模块组的组件,包括信号源模块组、连续模块组、离散模块组、查表模块组、用户自定义函数模块组、数学运算模块组以及信号与系统模块组,并对每一个模块组的内部部件进行了 Simulink 模型构建和仿真,使得读者能够快速掌握该模块的使用。