

第 5 章 测试信号处理及 LabVIEW 实现

数据采集系统完成数据采集任务之后，另一个很重要的任务就是对采集到的信号数据进行处理，以便得出数据的各种特征，如有效值、频谱信息、过渡过程等。广义上讲，信号处理包括信号生成和信号分析两大类。LabVIEW 提供了进行测试信号分析处理所需要的各种类型的数据分析和处理工具。

通过本章，用户能掌握 LabVIEW 中信号处理 VIs 的使用，能通过 LabVIEW 对测试信号进行基本的分析和处理。

5.1 信号处理概述

本节介绍信号处理的一些基础知识，同时介绍在 LabVIEW 中信号处理 VIs 的类型和组织形式。

5.1.1 信号处理的任务

由数据采集系统的任务可知，系统除了采集数据外，还要根据实际需要，对采集到的信号进行各种形式的处理。信号处理的类型有多种，一般根据以下方式进行分类。

按处理的方式划分，数据处理可分为实时处理（即在采集数据的同时，对数据进行处理）和事后处理。一般来说，实时处理由于处理时间受到限制，因而只能对有限的数据做一些简单的、基本的处理，以提供用于实时监控的数据；而事后处理由于是非实时处理，处理时间不受限制，因而可以做各种复杂的处理。

按处理的性质划分，数据处理可分为预处理和二次处理两种。预处理通常是剔除数据奇异项、去除数据趋势项、数据的数字滤波、数据的转换等。二次处理包括各种数学运算，如微分、积分和傅里叶变换等。

数据处理的任务主要有以下三个方面：

（1）对采集到的电信号作物理量解释

在数据采集系统中，被采集的物理量经传感器转换成电量，又经过信号放大、采样、量化和编码等环节后，被系统中的计算机所采集，但是采集到的数据仅仅是以电压的形式表现。它虽然含有被采集物理量变化规律的信息，但由于没有明确的物理意义，因而不便于处理和使用，必须把它还原成原来对应的物理量。

（2）消除数据中的干扰信号

在数据的采集、传送和转换过程中，由于系统内部和外部各种干扰、噪声的影响，或多或少会在采集的数据中混入干扰信号。因而必须采用各种方法（如剔除奇异项、滤波等）最大限度地消除混入数据中的干扰，以保证数据采集系统的精度。

(3) 分析计算数据的内在特征

通过对采集到的数据进行变换加工,或在有关联的数据之间进行某些相互的运算,从而得到能表达该数据内在特征的二次数据,所以有时也称这种处理为二次处理。

例如,采集到一个振动过程的振动波形(随时间变化的数据,即时域数据),由于频谱更能说明振动波形对机械结构所产生的影响,因此,可用傅里叶变换得出振动波形的频谱。

5.1.2 信号处理的方法

信号处理与分析的方法很多,在设计测试系统时,需要根据系统的功能要求和所处理信号的实际情况选择合适的分析与处理方法方法。常用的信号分析与处理方法有:

(1) 时域分析

通过时域分析可以得到信号在时域的各种特征量,如测量波形的幅度信息(峰值、均值、有效值)和时间信息(周期、频率等),主要的时域分析方法有相关性分析、卷积处理以及对信号的其他一些处理。

(2) 频域分析

测量时直接采集到的信号是时域波形,由于时域分析的局限性,所以往往把问题转换到频域来处理。通过频域分析可以得到信号在频域的各种特征量及信号的频率组成信息。最主要的频域分析方法就是快速傅立叶变换及其反变换。

(3) 加窗处理

LabVIEW 中提供了多种窗处理函数,如汉宁窗(Hanning)、海明窗(Hamming)、布拉克曼窗(Blackman)及凯赛窗(Kaiser)等。

(4) 滤波处理

使用滤波器对信号进行滤波,可以得到想要的频率分量。滤波器分为模拟滤波器和数字滤波器,可以用软件实现的只有数字滤波器。

(5) 信号仿真

当没有信号输入时,可以用数字函数通过计算得到数据,用来模拟实际信号的离散值,称为仿真信号。仿真信号可用于检验后续的信号分析是否正确,这种方法称之为信号仿真。

在测试系统设计和软件开发过程中,数学分析与信号处理是两个不可缺少的重要内容。系统采集到的测量信号与数据必须经过一定的数学分析和处理,才能给出人们需要的信号和结果,数学分析与信号处理已经形成了标准的算法程序。

5.1.3 LabVIEW 中的信号处理实现

与其他编程语言或平台相比,LabVIEW 尤其适合信号处理的学习和相关设计,这是因为它具有以下其他平台无可比拟的优势。

- 具有良好的图形显示功能,提供种类齐全的各种波形图和波形图表控件,能够以多样化的方式直观显示各种信号波形;
- 图形化的编程方式,学习门槛较低,易于掌握,省去了许多烦琐的编程细节,能够使用户专注于数字信号处理算法的研究和设计;
- 拥有数量众多、功能齐全的各种信号分析与处理 VI,供用户随意调用,从而组合

实现出所需要的信号处理功能，对于各种应用需求，无论是学习还是工程设计，都能够胜任；

- 具有良好的扩展性，无论是通过附加工具包扩展，还是通过与其他平台（如 MATLAB/Simulink）的接口扩展，都能很方便地进一步扩展其信号处理功能。

LabVIEW 将数据采集和测试分析中常用的数学与信号分析算法程序集成在一起，提供了先进的数学与信号分析环境，所有数学分析节点都集中函数选板“数学运算”下，而信号处理节点都集中在函数选板“信号处理”下。

使用 LabVIEW 编写和设计数字信号处理程序，主要靠其提供的种类丰富的各种信号处理 VI 实现，包括信号生成、调制、滤波、频谱分析、变换等各种算法都有相关可调用的 VI 库，这些 VI 库的位置位于函数选板“信号处理”下，如图 5-1 所示。

其中第一行包括三个与波形信号相关的 VI 库，分别是波形生成、波形调理和波形测量，这些 VI 库中所包含的 VI 的数据接受类型为波形类型，含有采样间隔和起始时间信息，使用起来较为方便，本章中主要介绍这类 VI 的使用方法。

其他 VI 库中所包含的 VI 的数据接受类型为数值型数组，这些 VI 库涵盖了信号生成、信号运算、窗、滤波器、谱分析、变换等数字信号处理的各领域。

“逐点”VI 库中提供了能够对信号数据进行逐点计算的 VI，信号数据可以不用依照数组形式一次输入，而是可以按照标量形式逐个输入，计算结果逐次不断修正，这非常适用于实时计算和在线分析。

表 5-1 对【信号处理】子选板的各个下级子选板的功能进行简要地描述。

表 5-1 信号处理子选板功能简述

子选板	图标	功能描述
波形生成		利用波形生成 VIs 可以生成各种不同类型的信号波形，如正弦波形、噪声波形等
波形调理		使用波形调理 VIs 实现数字滤波、加窗等功能
波形测量		波形测量 VIs 用于信号的时域和频域处理，如 DC、RMS、Harmonic Distortion 等
信号生成		利用信号生成 VIs 可以产生指定模式的信号，信号以一维数组的形式返回
信号运算		利用信号运算节点对信号进行各种形式的运算，如卷积、自相关、互相关等
窗		窗 VIs 执行平滑窗和数据窗等功能

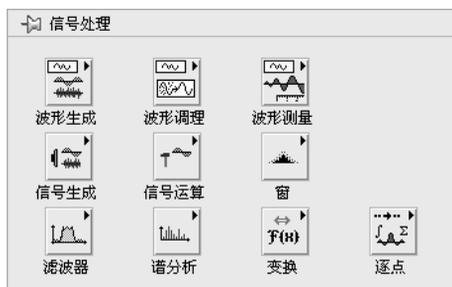


图 5-1 信号处理相关 VI 库

续表

子选板	图标	功能描述
滤波器		滤波器 VIs 包括 IIR 滤波器 VI、FIR 滤波器 VI 及非线性滤波器 VI 等
谱分析		谱分析 VIs 可以对信号进行各种频谱分析, 如自功率谱、互功率谱等
变换		利用变换 VIs 可以对信号进行各种变换处理, 如 FFT、反 FFT、离散余弦变换等
逐点		使用逐点 VIs 可以便捷有效地对信号进行逐点分析处理。仅 LabVIEW 的完整开发版本和专业开发版本包含逐点 VIs

5.2 波形和信号生成

典型信号的生成是测试信号处理中首先遇到的问题, 准确快捷地产生符合所需参数的信号波形, 是准确进行后续分析和处理的基础。

5.2.1 波形和信号生成相关的 VI

测试信号是测试系统工作的基础, 测试信号产生的途径有:

(1) 对被测的模拟信号, 使用数据采集卡或其他硬件电路, 进行采样和 A/D 变换, 送入计算机。

(2) 从文件读入以前存储的波形数据, 或由其他仪器采集的波形数据。

(3) 在 LabVIEW 中的波形产生函数得到的仿真信号波形数据。

LabVIEW 提供了产生各种基本函数的信号发生器, 可以产生正弦波、锯齿波、方波、三角波和白噪声等各种常用的周期和非周期信号。

LabVIEW 的信号发生器函数大体分为 4 类:

- Express VI 信号发生器;
- 波形发生器 VI;
- 普通信号发生器 VI;
- 公式波形发生器 VI。

这 4 类 VI 的基本功能是相同的, 只是在参数设置和使用的复杂度上略有区别。在 LabVIEW 中测试信号已经是离散化的时域波形数据, 信号表示的数据类型有数组、波形数据和动态数据 3 种。在 LabVIEW 中产生一个仿真信号, 相当于通过软件实现了一个信号发生器的功能。

在 LabVIEW 中提供了两类 VI 库, 波形生成和信号生成, 分别用于生成数据类型为波形和数值型数组的信号, 这两类 VI 库所包含的 VI 分别如图 5-2 和图 5-3 所示。

这些波形和信号生成 VI 所涉及的信号类型相当广泛, 不仅包括基本的正弦波、三角波、锯齿波、方波等波形信号, 也包括各种高斯信号、脉冲信号和噪声信号, 通过设定适当的参数, 几乎可以生成一切所需要的波形信号。在此以波形生成 VI 库中的几个常用 VI

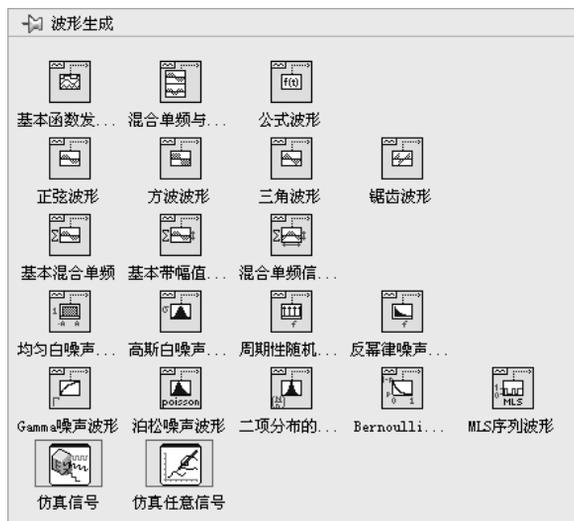


图 5-2 波形生成 VI 库

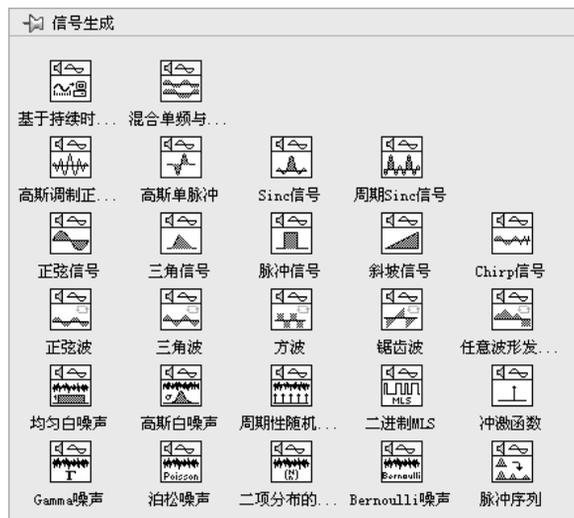


图 5-3 信号生成 VI 库

举例作介绍，其他波形生成 VI 和信号生成 VI 用法类似，可在实际需要时结合帮助文档中的详细说明进行设置和调用。

5.2.2 波形信号生成举例

下面通过几个实例，讲述如何使用 LabVIEW 中信号生成相关的 VI 来产生信号。

例 5-1 基本信号生成实例

信号处理中最常用到的几类基本信号：正弦波、三角波、方波和锯齿波，在波形生成 VI 库中都有各自对应的 VI 可以生成，也可以由一个通用的基本函数发生器 VI (Basic Function Generator.vi) 生成。

该 VI 的功能类似于一个简单的信号发生器，该 VI 的连线板如图 5-4 所示。

该 VI 主要输入输出参数的含义为：

- 偏移量：输入参数，所生成信号的偏移量，也就是直流分量；
- 充值信号：输入参数，为真时将信号的初始相位设为相位参数指定的角度值，并将时间信息重置为零，为假时则信号初始相位和时间信息都使用上一次调用后的末值，该参数默认为假；
- 信号类型：输入参数，所生成信号的类型，可指定为 Sine Wave、Triangle Wave、Square Wave、Sawtooth Wave 等参数，分别与正弦波、三角波、方波、锯齿波相对应；
- 频率：输入参数，所生成信号的频率，以 Hz 为单位；
- 幅值：输入参数，所生成信号的幅度，即信号峰值；
- 相位：输入参数，在 reset signal 参数为真时指定信号的初始相位；
- 采样信息：输入参数，为簇类型，元素包括采样率和样本点数；
- 方波占空比 (%)：输入参数，指定方波的占空比；
- 信号输出：输出参数，为所生成的波形信号数据，可以直接连接到波形图显示；
- 相位输出：输出参数，为所生成信号的末相位。

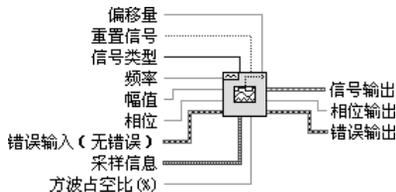


图 5-4 基本函数发生器的连线板

使用 LabVIEW 实现基本信号发生器的具体步骤如下。

① 新建“基本信号的生成.vi”，为了连续生成和观察信号波形，这里使用基于 while 循环结构的框架实现本程序，所以在框图中添加一个 while 循环结构。

② 在循环结构内放入“Basic Function Generator.vi”，并为它的各个输入参数创建相应的输入控件，将它的输出参数“信号输出”连接到一个波形图进行显示。

最终程序前面板和框图如图 5-5 所示。运行程序后通过前面板上的各个控件调整输入参数，就可以观察到所生成波形的动态变化。

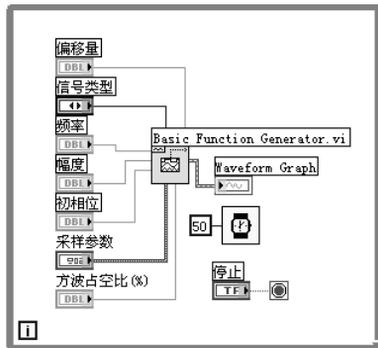
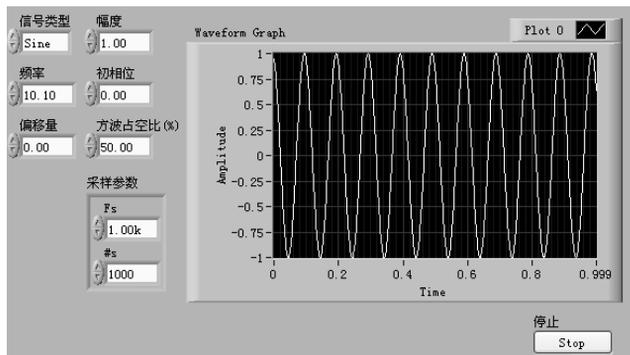


图 5-5 “基本信号的生成.vi” 的前面板和框图

注意，为了满足采样定理的需要，设置的信号频率不能过大，更不能超过采样率的一半。

例 5-2 公式设定信号的生成实例

如果需要根据已知的公式生成一定规律的波形信号，可以使用公式波形 VI (Formula Waveform.vi) 生成，该 VI 的连线板如图 5-6 所示。

该 VI 的各个输入和输出参数的含义与上例中的“Basic Function Generator.vi”十分相似，不同之处仅在于多了一个新的参数“公式”，通过这个参数可使用字符串型的“公式”来表达所需要的波形信号函数式。

公式可以为任意形式的合法公式表达式，而且预定义了一些变量，可以在表达式中直接使用，这些变量的含义如表 5-2 所示。

如表达式 $\sin(w*t)*\sin(2*\pi(1)*t)+1$ 表达的公式相当于 $\sin(2\pi f \cdot t)\sin(2\pi \cdot t)+1$ 。

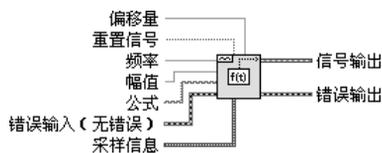


图 5-6 公式波形 VI 的连线板

表 5-2 公式表达式中的预定义变量

变 量	含 义
f	输入的频率参数
a	输入的幅度参数
w	$2*\pi*f$
n	已生成的采样数
t	已用时间
f_s	输入的采样频率

新建一个名为“公式设定信号的生成.vi”的 VI，添加一个 while 循环结构，以便连续地生成和观察信号波形。在 while 循环中放入“Formula Waveform.vi”，并为它的各个输入参数添加相应的控件，将它的参数“信号输出”连接到一个波形图进行显示。

程序前面板和框图如图 5-7 所示。

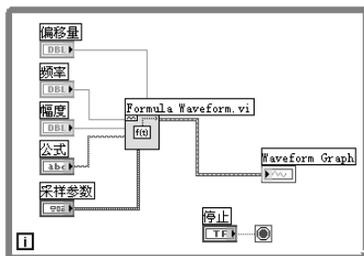
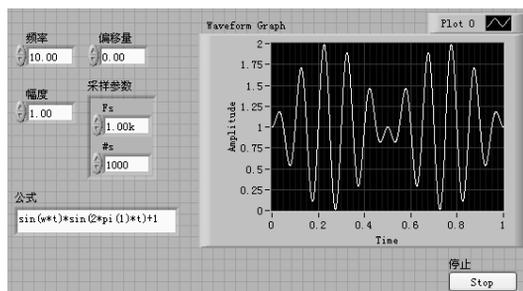


图 5-7 “公式设定信号的生成”的前面板和框图

运行程序后通过前面板上的各控件调整输入参数，以及写入正确的公式，就可以观察到所生成波形的动态变化。

除了基本函数发生器和公式波形这两类最常用到的信号生成 VI 外，还有生成各种其他类型信号，如高斯信号、白噪声信号等的各种 VI 可供调用，在此不一一举例。

5.2.3 仿真信号的生成

值得说明的是，信号生成 VI 库里还提供了两个信号生成 Express VI 可快速生成所需的

信号，它们分别是“仿真信号”(Simulate Signal.vi)和“仿真任意信号”(Simulate arbitrary Signal.vi)，两者的区别在于，Simulate Signal.vi 用来发生规则的周期信号或者叠加有噪声的周期信号；而 Simulate arbitrary Signal.vi 则是通过手动输入数据或者载入数据文件的方法产生任意信号的。

1. Simulate Signal.vi

Simulate Signal.vi 的适用场合与基本信号的生成类似，可根据指定参数生成正弦波、三角波、方波、锯齿波、直流信号等几种基本类型的信号。在框图上放入该 VI 后，弹出的配置对话框如图 5-8 所示。

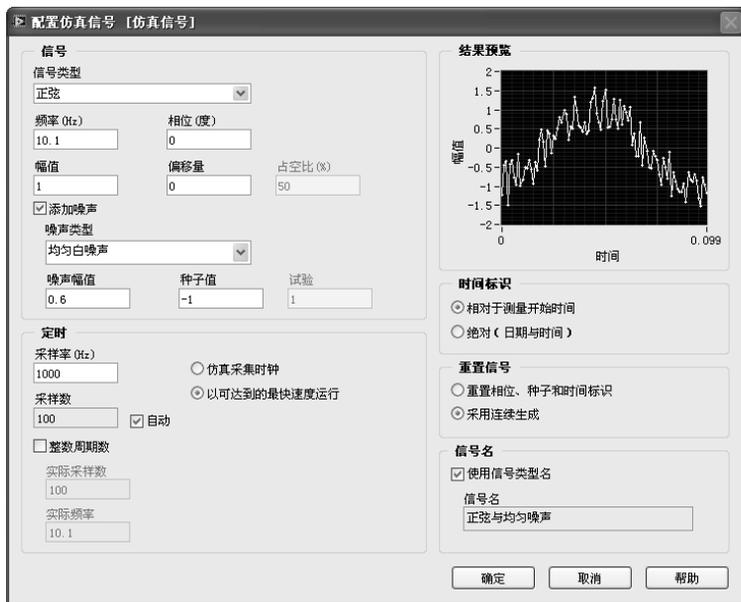


图 5-8 Simulate Signal.vi 的配置对话框

从对话框中不仅可以配置基本的信号类型、频率、初始相位、幅度、偏移量、采样率等参数，还可选择是否叠加上某种类型的噪声(噪声类型同样也提供了多种常见类型可选)，以及是否微移采样率以保证整周期采样(即保证一个周期里的采样点数为整数个，这在抑制傅里叶分析和抑制谱泄漏时需要考虑)。

该对话框中所提供的配置功能已经相当全面，在保证快捷性的同时又不失灵活性，是快速搭建程序原型时信号产生的首选方法之一。

2. Simulate arbitrary Signal.vi

Simulate arbitrary Signal.vi 用于产生一个任意的波形信号。它有两种产生任意信号的方式：一种是在定义信号对话框中手动编写信号，所编写的信号文件可以输出为 LVM 文件；另一种是通过导入已有的 LVM 格式的数据文件，产生信号。通常，如果希望调用在其他软件中生成的信号，或者调用其他方法采集的信号，在 LabVIEW 中进行仿真时，经常采用第二种方法。

在框图上放入该 VI 后，弹出的配置对话框如图 5-9 所示。

单击该对话框中左上方的“定义信号...”按钮后，可以弹出另外一个专用对话框，用来逐点编辑信号点的精确数据，编辑完毕返回主对话框后就可即时预览到波形曲线。另外

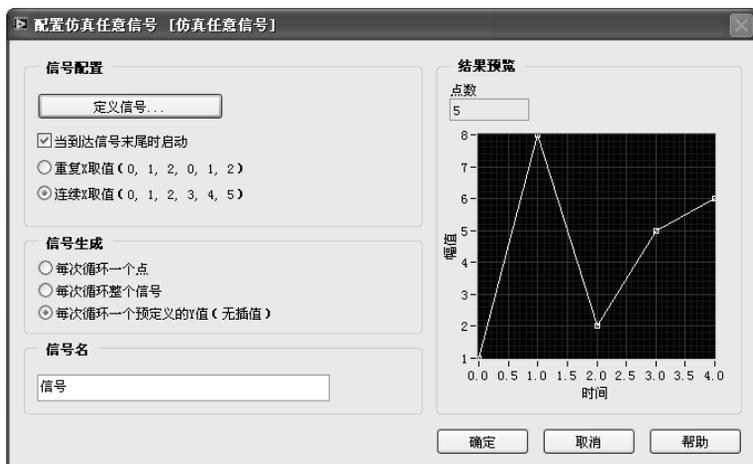


图 5-9 Simulate arbitrary Signal.vi 的配置对话框

也对信号 X 轴数据的重复方式，以及信号循环生成方法等提供了可供设置的选项。这种可精确设置信号点数据的 Express VI 在信号数据量不是很大，或需要生成某些特殊类型的信号时非常实用。

例 5-3 利用 Simulate Signal.vi 设置一个占空比可调的方波信号发生器实例

首先对 Simulate Signal.vi 通过对话框参数设置，设定输出信号类型为方波，并设置好相应的采样参数，点击确定后，Simulate Signal.vi 将变成带有方波信号发生器相应端口的 VI。将该 VI 的占空比端口和噪声信号幅度端口与旋钮控件相连，则实现了该设计。

在该方波信号发生器运行时，旋动占空比旋钮控件，可以明显地看出方波信号相应的变宽或者变窄，而旋动噪声信号旋钮控件时，可以发现噪声幅度的变化。

程序前面板和框图如图 5-10 所示。

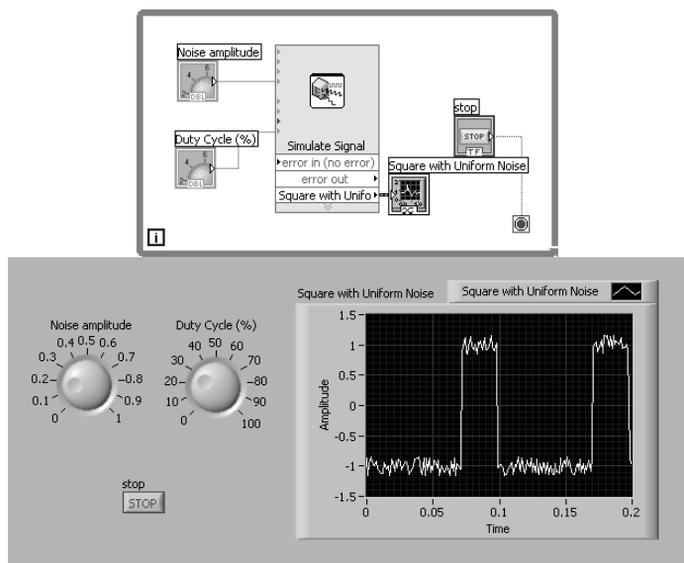


图 5-10 占空比噪声幅度可调的附带标准白噪声方波发生器的程序前面板和框图

例 5-4 利用 Simulate arbitrary Signal.vi 编写的三角波信号实例

程序前面板和框图如图 5-11 所示。

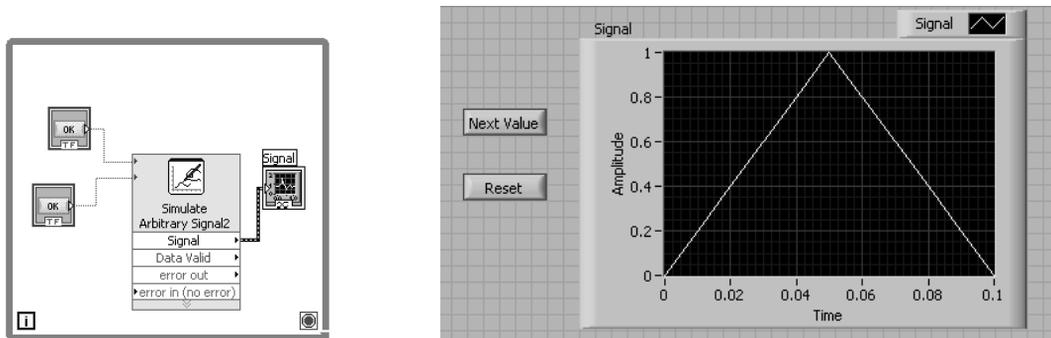


图 5-11 利用 Simulate arbitrary Signal.vi 生成三角波信号的程序前面板和框图

Simulate arbitrary Signal.vi 可以像普通信号发生器一样发生三角波信号，但是，值得注意的是，该三角波的频率和幅度都是在图 5-11 的信号定义对话框中确定了的，不能更改。

Simulate arbitrary Signal.vi 编写功能使得产生许多非常规信号成为可能，大大丰富了 LabVIEW 的信号发生功能。

直接编写信号固然直观，但是，很多时候，进行数据仿真往往需要把一个通过其他传感器或者设备获得的信号导入 LabVIEW 进行处理。这时，就可以通过 Load Signal 按钮载入已有信号的 LVM 文件，以便导入该信号。

需要注意的是，LVM 文件有严格的格式，只有符合这种格式的文件才能被 LabVIEW 识别并载入，下面就是一个 LVM 文件的例子。

```

LabVIEW Measurement
Writer_Version 0.92
Reader_Version 1
Separator Tab
Multi_Headings Yes
X_Columns No
Time_Pref Relative
Operator Administrator
Date 2013/03/12
Time 21:31:34.264999
***End_of_Header***

Channels 2
Samples 11 11
Date 2013/03/12 2013/03/12
Time 21:31:34.264999 21:31:34.264999
X_Dimension Time Time
X0 0.0000000000000000E+0 0.0000000000000000E+0
Delta_X 1.000000 1.000000
***End_of_Header***
X_Value X Array Y Array Comment

```