

测控技术实验

实验报告



姓 名:	刘侃
学 院:	机械工程学院
专 业:	机械工程
学 号:	3220103259
分 组:	组 10

浙江大学机械工程实验教学中心

2024 年 9 月

实验名称：Labview 基础操作实验

一、实验目的

1. 熟悉 LabVIEW 软件界面，了解 LabVIEW 的编程环境。
2. 掌握 LabVIEW 的基本操作方法，并编制简单的程序。

二、实验原理

1. 虚拟仪器技术

虚拟仪器是基于计算机的软硬件测试平台，允许用户灵活定义专用仪器。LabVIEW 是由 NI 公司开发的一种图形化编程语言，主要用于虚拟仪器的开发。LabVIEW 的优势在于编程方式简洁直观，通过流程图实现数据处理。

2. LabVIEW 的应用结构

LabVIEW 程序由前面板、框图和图标/连接器三部分组成：

- 前面板：用户界面，包含输入控制和输出显示。
- 框图：程序的图形化源代码，控制前面板上对象的行为。
- 图标/连接器：用于其他 VI 调用当前 VI 时的数据输入/输出端口。

3. 程序调试技术

LabVIEW 提供多种调试工具，如高亮执行、断点设置和探针，用于逐步检查程序执行和数据流动，帮助用户定位程序中的语法和逻辑错误。

三、实验步骤

1. 启动 LabVIEW，创建新的 VI 项目。
2. 在前面板中加入液体容积显示器和温度计，设置各自的量程范围。
3. 使用程序框图设计流程，加入随机数发生器、数值常数和乘法器，模拟数据采集。
4. 连接对象并确保连线正确，保存 VI 文件。
5. 运行 VI，观察数据在前面板中的显示。

（“一、实验目的、二、实验原理、三、实验步骤”合计篇幅限定 2 页以内）

四、实验结果

文件见附件“温度和容积.vi”

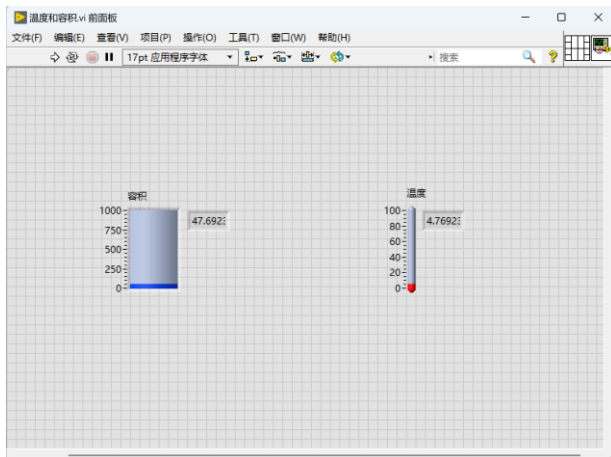


图 1 前面板

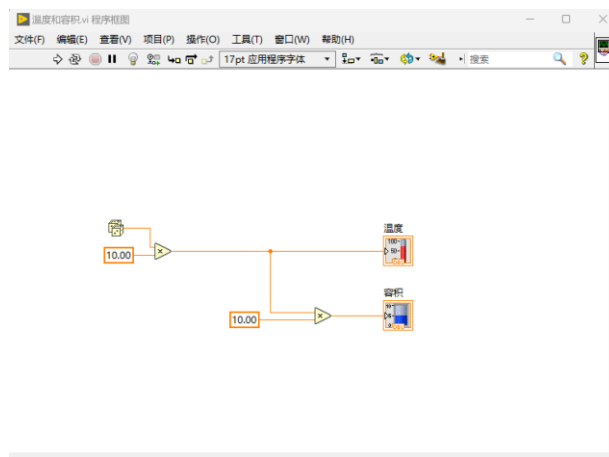


图 2 程序框图

五、思考题

1. 理解基于数据流编程的基本编程思路

数据流编程是一种高效的并行计算编程模型，旨在充分利用多核处理器的潜力。与传统编程语言相比，数据流编程的特点是基于数据的流动而非程序的顺序执行。它通过将计算任务分配给不同的处理器核心，使得每个核心都能够独立处理属于它的数据，从而实现高并行性。这种方式使得计算与数据传输可以相互分离，从而提高了整体的执行效率。

在数据流编程的范式下，程序被表示为一个有向图，节点表示计算单元，边表示数据传输路径。节点只在其所有输入端口都接收到有效数据时才会被激活执行，执行完成后将结果传输到相应的输出端口，这样下一个节点可以继续执行。因此，程序的执行顺序并不是由代码的位置决定的，而是由数据流动的方向决定。通过这种方式，数据流编程能够充分利用多核处理器的并行性，同时避免传统编程中对代码执行顺序的依赖问题。

2. 了解 VI 程序的构成及其各部分的功能

VI（虚拟仪器）程序由三个主要部分组成，分别是前面板、程序框图以及图标/连接器。这三部分共同构成了 LabVIEW 中的程序设计框架。

● 前面板

前面板是用户与 VI 交互的图形化界面。在前面板上，用户可以添加各种控件用于数据输入，比如滑块、按钮、数值输入框等。同时，也可以在前面板上放置显示对象，用于显示计算结果，如图表、指示灯等。通过前面板，用户可以直观地观察到输入和输出的变化。

● 程序框图

程序框图是 VI 的核心部分，它包含了 VI 的图形化源代码。在框图中，用户使用图形符号来编写程序，连线工具用于连接不同的函数和控件，代表数据的流动。框图的作用是通过这些图形化的编程元素控制前面板上的输入输出行为，最终实现程序的逻辑功能。

● 图标/连接器

每个 VI 都可以作为子 VI 被其他 VI 调用，图标/连接器就是 VI 对外接口的表现形式。图标是 VI 在其他程序中的代表，而连接器则提供输入和输出端口，允许其他 VI 将数据传递给这个 VI，或者从 VI 中获取数据。通过这种方式，用户可以将多个 VI 组合在一起构建复杂的程序系统。

3. 了解 LabVIEW 基础控件、函数，了解循环结构（for、while）

● 基础控件

LabVIEW 提供了多种基础控件用于数据的输入和输出，列举如下。

数值型控件：数值型控件用于表示不同类型的数值数据。用户可以通过右键点击数值型控件来改变数据的表示法，例如整数、浮点数等。

布尔型控件：布尔型控件用于处理逻辑数据，即真或假状态。布尔型控件的关键在于其机械动作的设置，不同的机械动作决定了控件如何响应用户的输入。

字符串控件：字符串控件用于处理文本数据。通过 LabVIEW 中的函数，可以将多个字符串进行拼接，或者根据格式要求生成指定的字符串输出，常用于报告生成等场景。

枚举型控件：枚举型控件将有限的字符串与特定的数值相绑定，通常用于选择操作。通过枚举型控件，用户可以实现类似下拉菜单的效果，并根据选中的字符串执行不同的程序分支。

● 函数

LabVIEW 的函数库涵盖了丰富的运算和数据处理功能，主要包括基本运算、数组处理、位运算、逻辑操作、关系运算等。用户可以通过函数库直接调用这些函数来实现对数据的各种操作，从而简化编程工作。

● 循环结构

While 循环：While 循环是一种常见的循环结构，用于在特定条件满足之前不断执行循环体中的代码。While 循环的显著特点是它没有预设的循环次数，循环的终止条件完全依赖于用户定义的条件。每次循环结束后，程序会检查该条件是否已经满足，若未满足则继续执行循环。While 循环广泛应用于需要实时监测某些条件的场景。

For 循环：For 循环是一种固定迭代次数的循环结构。与 While 循环不同，For 循环在开始时就预先设定了迭代次数，因此它可以确保程序只运行指定的次数。如果用户希望在达到特定条件时提前结束循环，也可以将 For 循环配置为根据条件中断执行，从而灵活处理数据流中的异常情况。

实验名称：LabVIEW 信号发生与频谱分析实验

一、实验目的

1. 熟悉 LabVIEW 中的波形生成函数，了解常见波形函数的输入输出参数及其接线方法。
2. 掌握 LabVIEW 中数字信号的 FFT（快速傅里叶变换）方法，搭建 FFT 变换电路。
3. 掌握 LabVIEW 中数字信号波形图的显示方法。

二、实验原理

本实验采用的是数字处理式频谱分析原理，经过采样，使连续时间信号变为离散时间信号，然后利用 LabVIEW 的强大的数字信号处理的功能，对采样得到的数据进行滤波、加窗、FFT 运算处理，就可得到信号的幅度谱、相位谱以及功率谱。

FFT 的输出都是双边的，它同时显示了正负频率的信息。通过只使用一半 FFT 输出采样点转换成单边 FFT。FFT 的采样点之间的频率间隔是 f_s/N ，这里 f_s 是采样频率。FFT 和能量频谱可以用于测量静止或者动态信号的频率信息。FFT 提供了信号在整个采样期间的平均频率信息。在采样过程中，为了满足采样定理，对不同的频率信号，选用合适的采样速率，从而防止频率混叠。实际中，我们只能对有限长的信号进行分析与处理，而进行傅立叶变换的数据理论上应为无限长的离散数据序列，所以必须对无限长离散序列截断，只取采样时间内有限数据。这样就导致频谱泄漏的存在。所以利用加窗的方法来减少频谱泄漏。

本实验中利用枚举作为信号发生器模块，可产生任意标准周期信号，包括正弦波、三角波、方波、锯齿波。其中产生的周期信号的输入参数如频率、幅值、相位、占空比、噪声幅值、偏移量等均可调节。还有一个频谱分析模块，测试信号经滤波、加窗处理后，进行时域分析、频域分析以及谐波分析。可以进行各种参数设置，包括采样设置、滤波器类型选择及其参数设置、窗函数类型选择等。

三、实验步骤

- 1、设计一个虚拟信号发生器，能够产生正弦波、方波、三角波、锯齿波、白噪声等信号，且波形参数可调整。（20 分）
- 2、设计一个频谱分析仪，对上述虚拟信号发生器生成的信号进行频谱分析，并绘制出分析信号的幅频特性，指出主要频率分量的频率和幅值。（20 分）
- 3、查看噪声信号对上述信号幅频、相频的影响。（10 分）
- 4、搭建周期信号频率测量程序：产生一个正弦信号，要求其频率为 50Hz；改变采样频率和采样样本数，观察由 FFT 计算得到的频谱图，研究非整周期采样对 FFT 的影响。（给了参考 VI，请根据需求进行更改。）（30 分）
信号 1: $f=50\text{Hz}$, $F_s=1280$, $\#s=128$
信号 2: $f=50\text{Hz}$, $F_s=1000$, $\#s=128$
①给出被测信号（采用矩形窗）的频谱图，观察频谱特性的变化，分析频率分量产生泄露的原理。
②给出几种非周期采样情况下，提高被测周期信号频率准确率的方法，并通过改变实验参数进行验证。
- 5、选做：数据记录仪的设计（一个数据记录仪应用程序包括数据采集、数据分析、数据存储、数据调用），编制程序，完成其中数据记录和数据调用回放的功能。（10 分）

（“一、实验目的、二、实验原理、三、实验步骤” 合计篇幅限定 2 页以内）

四、实验结果

1. 设计一个虚拟信号发生器，能够产生正弦波、方波、三角波、锯齿波、白噪声等信号，且波形参数可调整。
2. 设计一个频谱分析仪，对上述虚拟信号发生器生成的信号进行频谱分析，并绘制出分析信号的幅频特性，指出主要频率分量的频率和幅值。

此处将上述两思考题中的虚拟信号发生器和频谱分析仪集成，文件见附件“信号发生与频谱分析.VI”

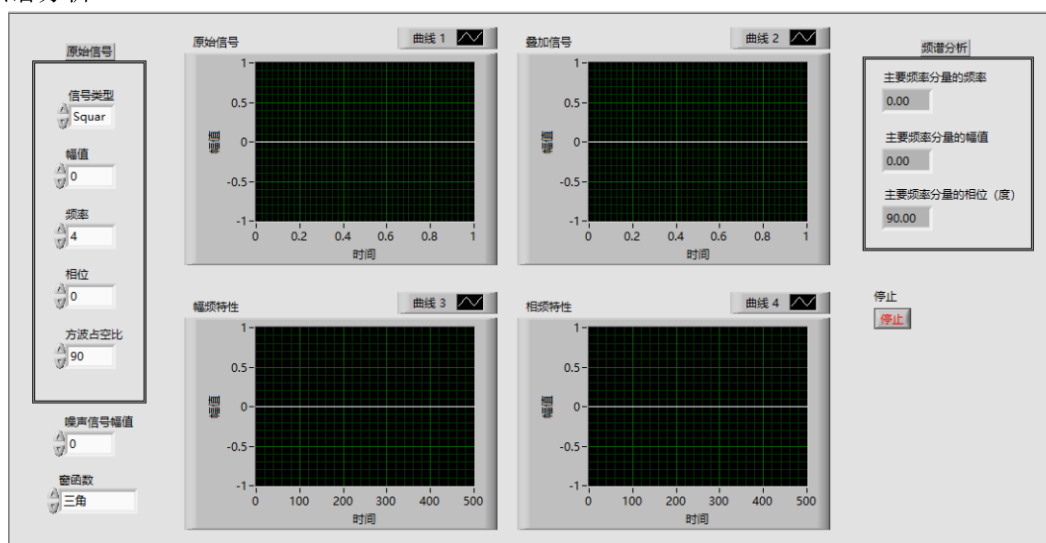


图 3 前面板

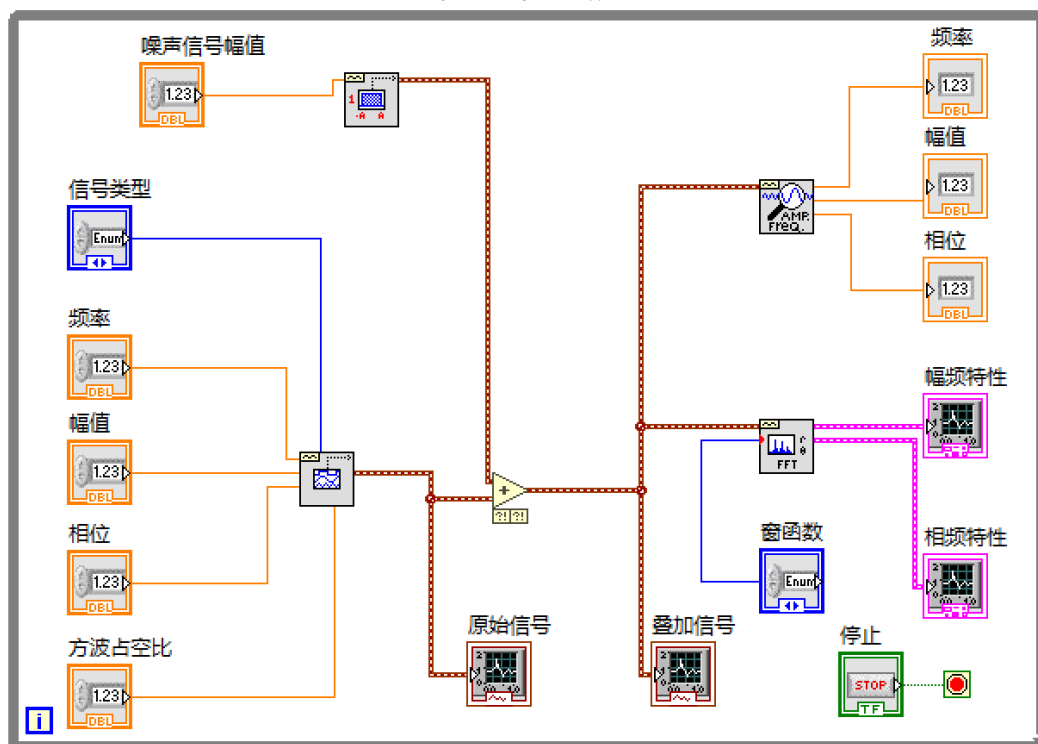


图 4 程序框图

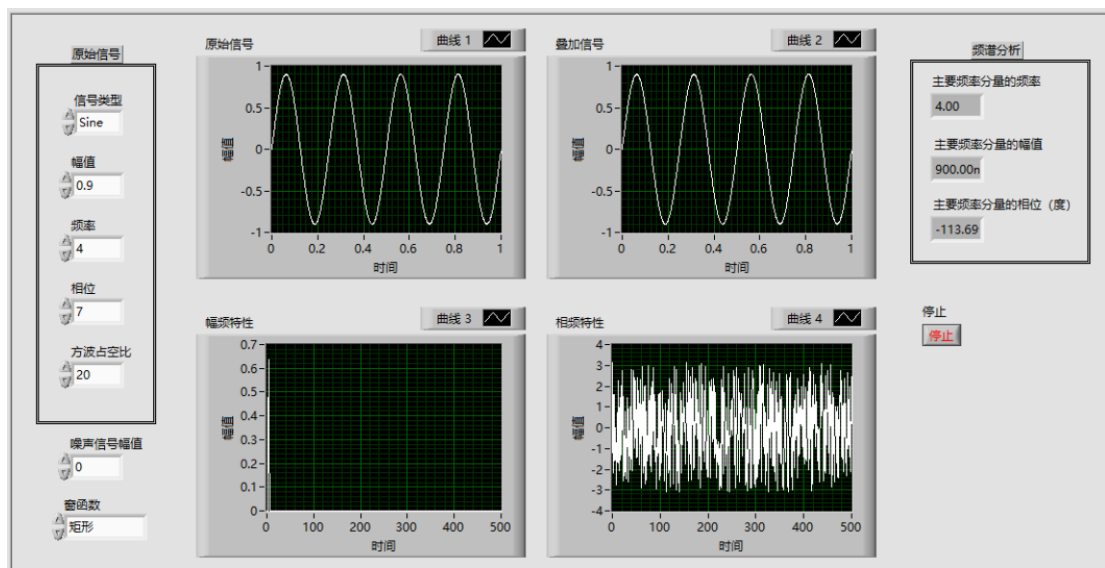


图 5 正弦波

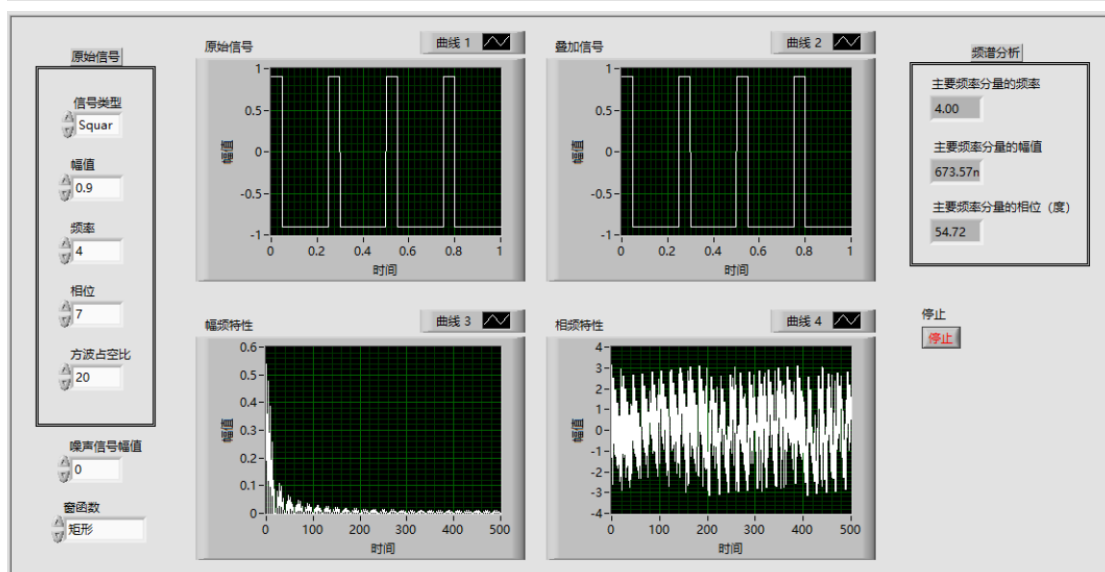
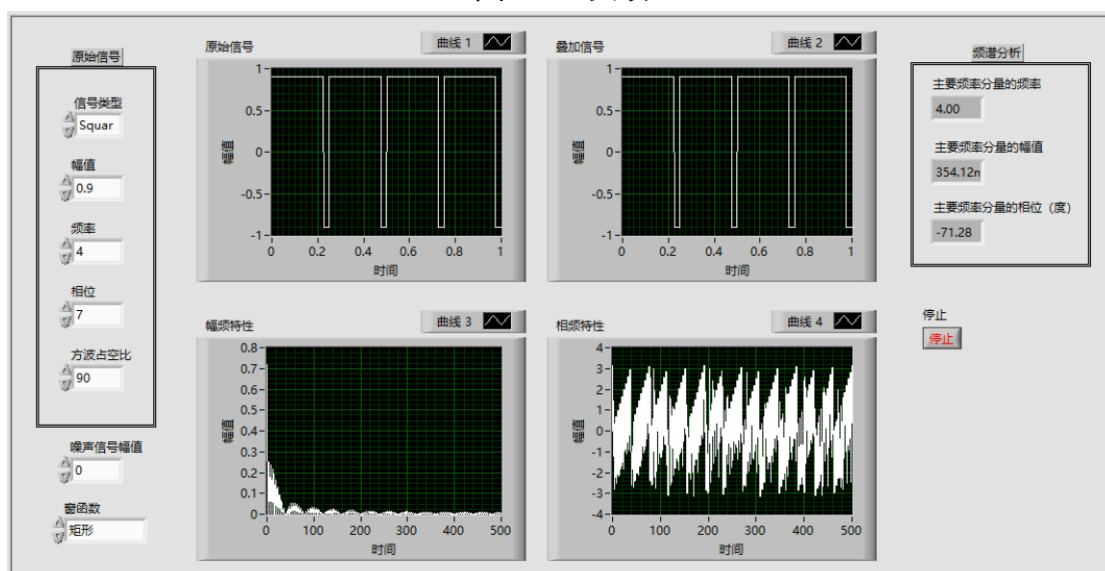


图 6 方波（可调占空比）

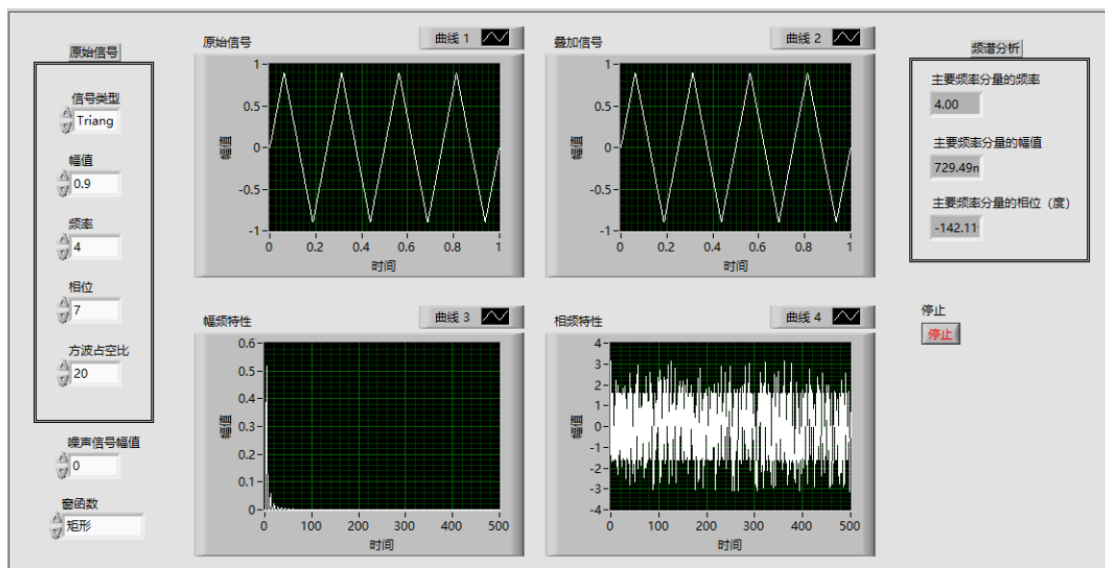


图 7 三角波

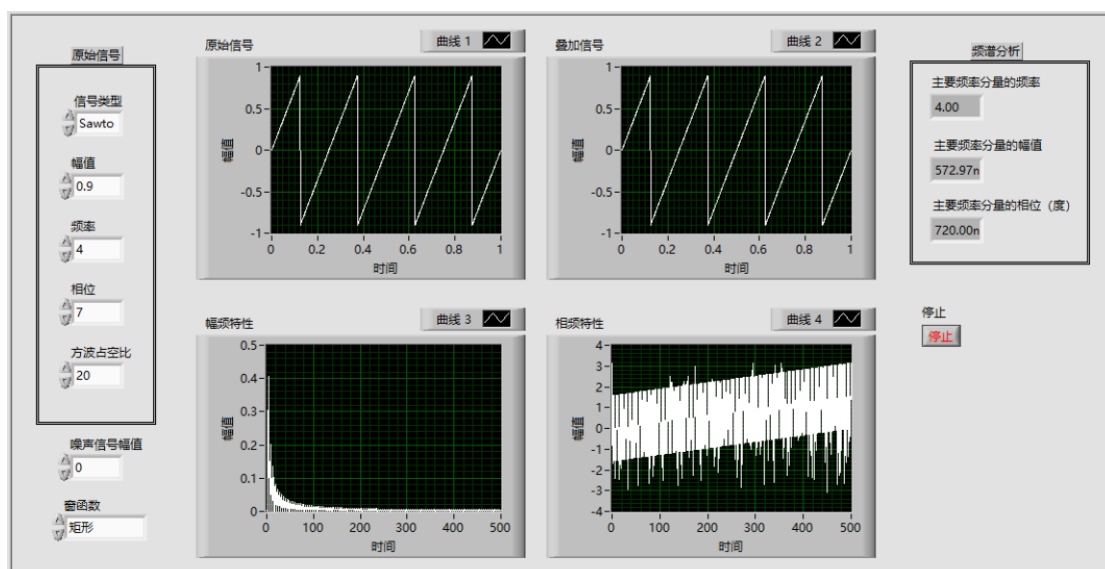


图 8 锯齿波

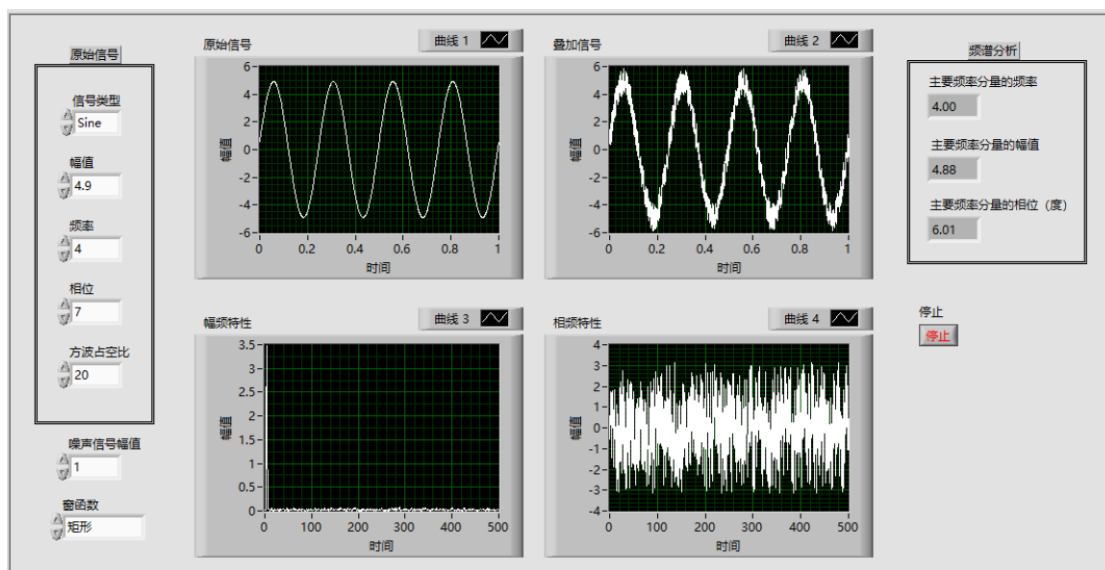


图 9 加入白噪声之后的信号

可以在左侧栏更改噪声信号的幅值和原始信号的类型、幅值、频率、相位、方波占空比，在中间栏查看信号的幅频、相频特性图像，在右侧栏查看实时的主要频率分量的频率、幅值和相位。

3. 查看噪声信号对上述信号幅频、相频的影响。

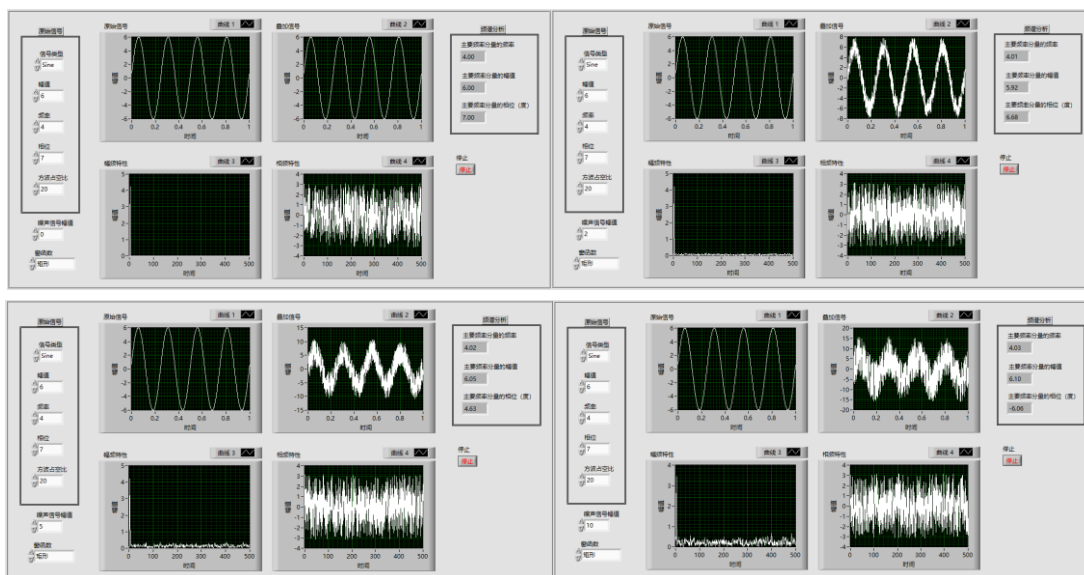


图 10 幅值为 6 的正弦波信号分别加入幅值为 0,2,5,10 的噪声后的变化

从中可以看出，当噪声信号较小时，噪声信号对对信号的幅频、相频影响较小，从频谱上看，只是把有用信号的频谱幅值有所提高。然而如果噪声信号远大于原始信号幅值，频谱上会更多的体现出白噪声的特征，频谱也会被搅乱。（此处的白噪声为均匀白噪声）

4. 搭建周期信号频率测量程序：产生一个正弦信号，要求其频率为 50Hz；改变采样频率和采样样本数，观察由 FFT 计算得到的频谱图，研究非整周期采样对 FFT 的影响。（给了参考 VI，请根据需求进行更改。）

信号 1: $f=50\text{Hz}$, $F_s=1280$, $\#s=128$

信号 2: $f=50\text{Hz}$, $F_s=1000$, $\#s=128$

① 给出被测信号（采用矩形窗）的频谱图，观察频谱特性的变化，分析频率分量产生泄露的原理。

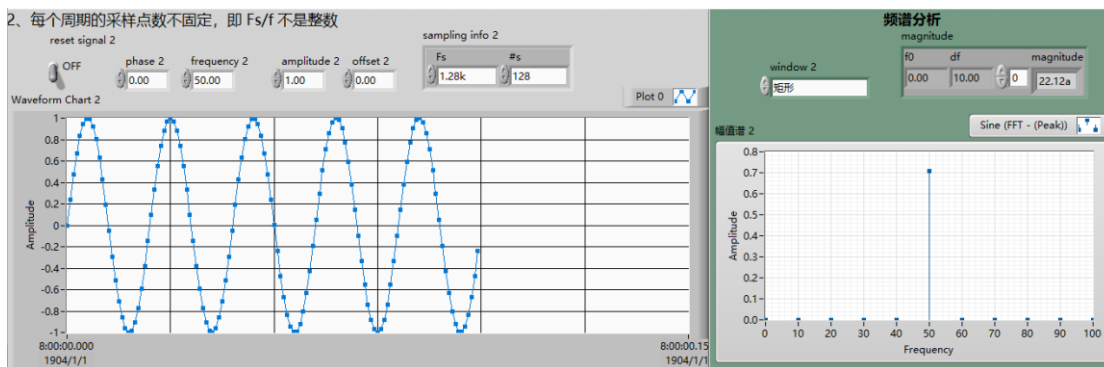


图 11 信号 1 的时域图和频谱图

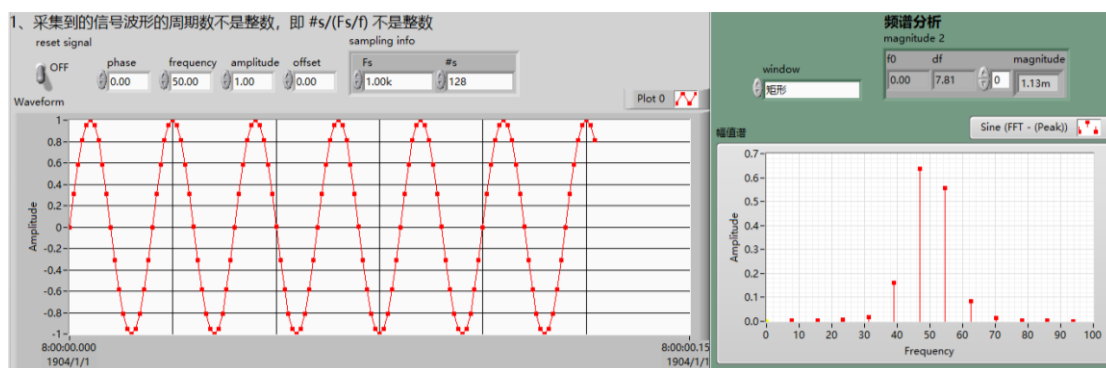


图 12 信号 2 的时域图和频谱图

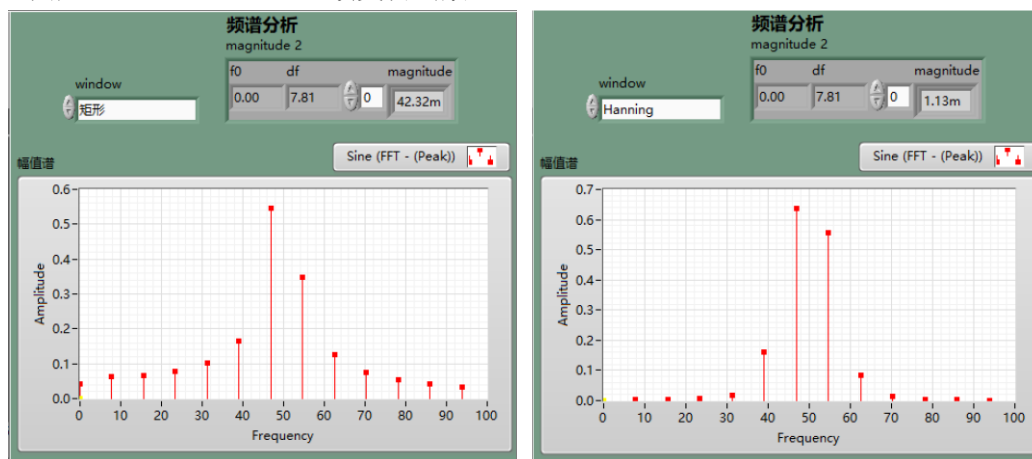
在对信号做 FFT 分析时，如果采样频率固定不变，由于被采样信号自身频率的微小变化以及干扰因素的影响，就会使数据窗记录的不是整数个周期。从时域来说，这种情况在信号的周期延拓时就会导致其边界点不连续，使信号附加了高频分量；从频域来说，由于 FFT 算法只是对有限长度的信号进行变换，有限长度信号在时域相当于无限长信号和矩形窗的乘积，也就是将这个无限长信号截短，对应频域的傅里叶变换是实际信号傅里叶变换与矩形窗傅里叶变换的卷积。

当信号被截短后的频谱不同于它以前的频谱。例如，对于频率为 f_s 的正弦序列，它的频谱应该只是在 f_s 处有离散谱。但是，在对它的频谱做了截短后，信号的频谱不只是在 f_s 处有离散谱，而是在以 f_s 为中心的频带范围内都有谱线出现，这些谱线可以理解为是从 f_s 频率上泄露出去的，这便是频谱泄露发生的原因。

- ② 给出几种非周期采样情况下，提高被测周期信号频率准确率的方法，并通过改变实验参数进行验证。

1、选择合适的窗函数

固定 $F_s=1000$ ， $\#s=128$ ，改变窗函数。



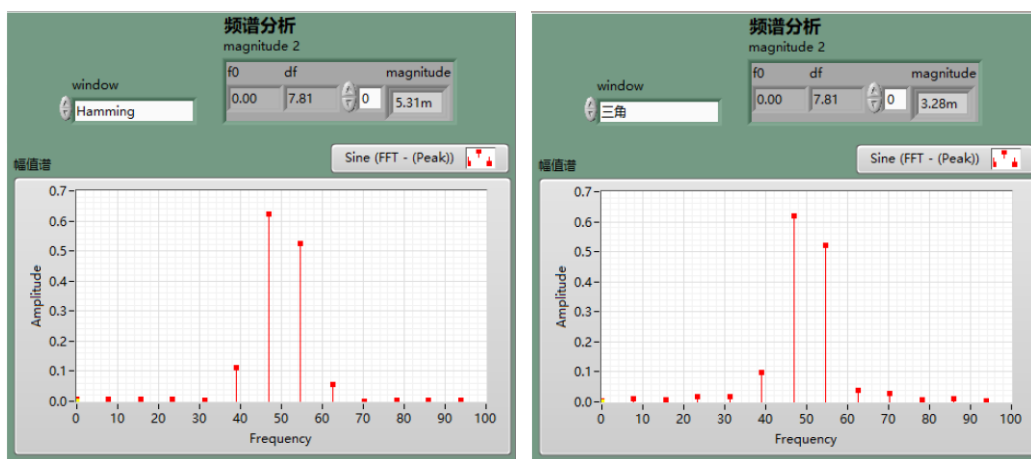


图 13 固定采样率和采样样本数不同窗函数下的频谱

其他条件不变，可以发现矩形窗分辨率高，泄露大，汉宁窗、汉明窗、三角窗分辨率低，泄露小。

2、增加采样样本数

选用矩形窗，固定采样率 $F_s=1000$ ，改变采样样本数。

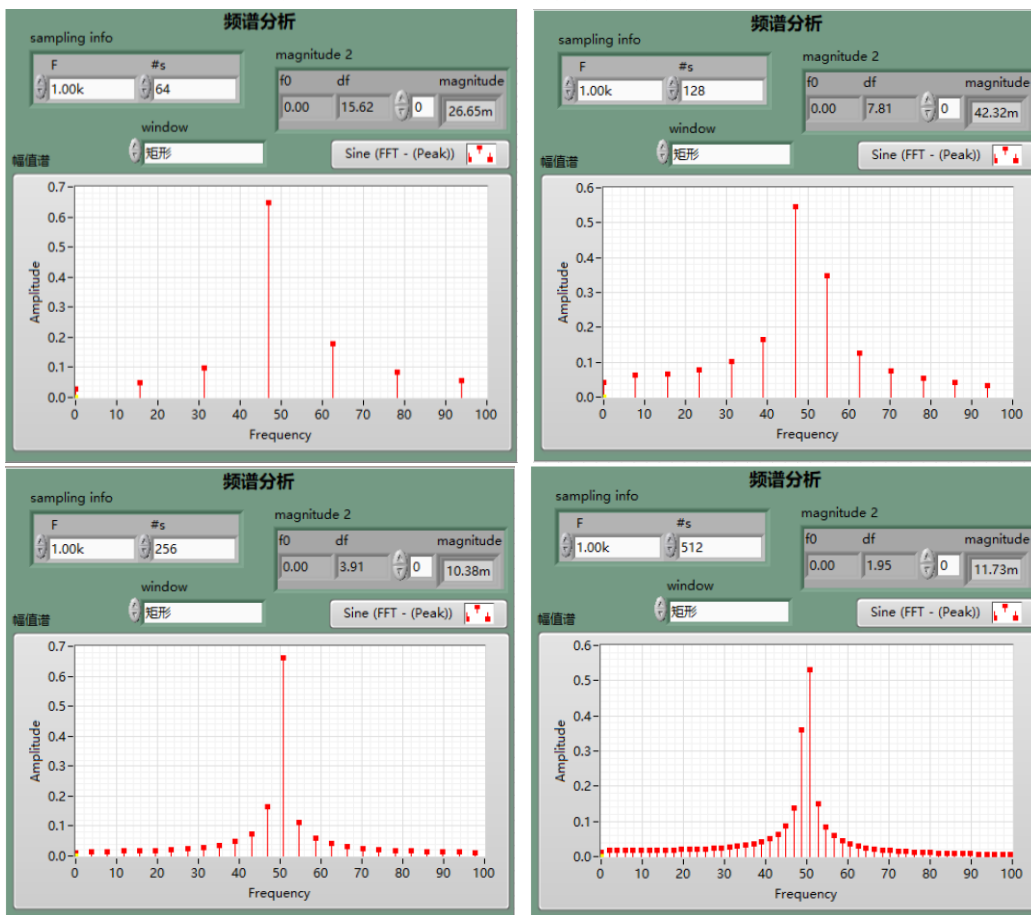


图 14 固定采样率和窗函数不同采样样本数下的频谱

其他条件不变，可以发现采样样本数越大，分辨率越高，泄露越小。因此适当增大样本数可提高分辨率、减少谱泄漏，提高频域分析的准确性。

3、降低采样率

选用矩形窗，固定采样样本数 $N_s=128$ ，改变采样率。

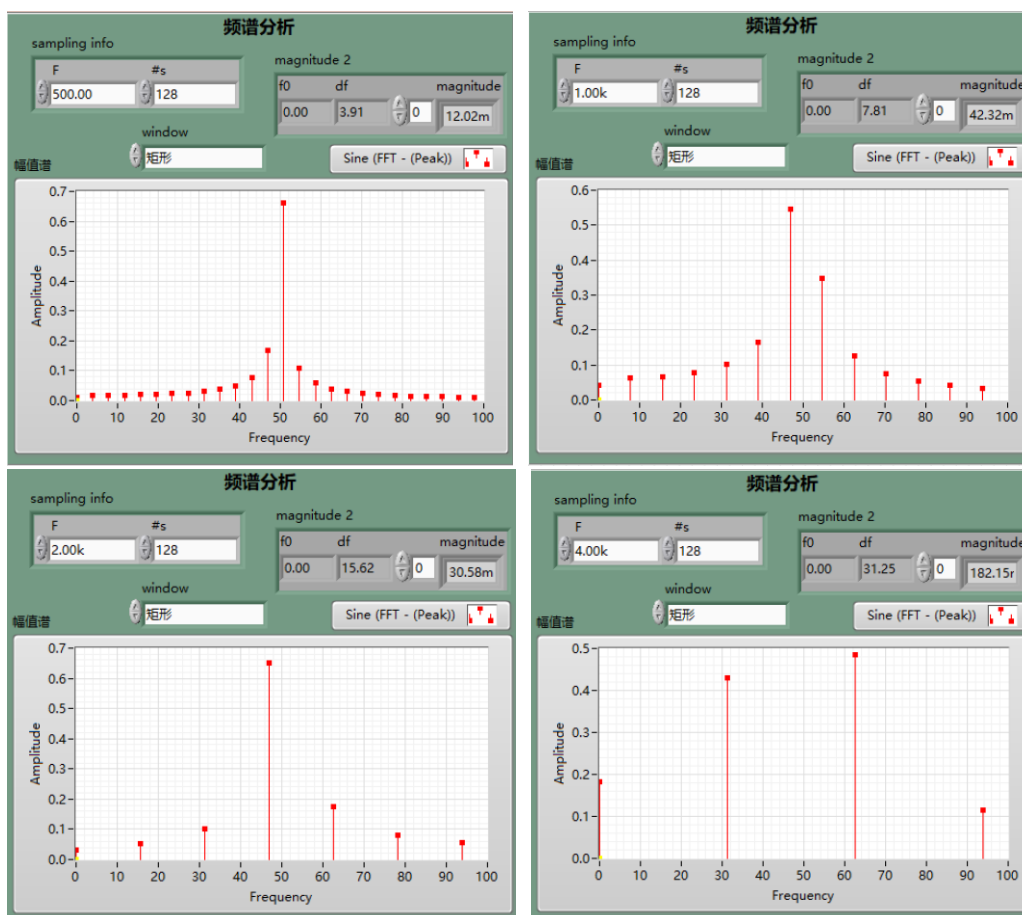


图 15 固定采样样本数和窗函数不同采样率下的频谱

其他条件不变，可以发现采样率越低，分辨率越高，泄露越小。因此适当降低采样率可提高分辨率、减少谱泄露，提高频域分析的准确性。