实验2

学号	姓名	分工
21307035	邓栩瀛	完成分析题
21307056	钟欣余	完成实验二中A代码的编写
21307037	金思琪	完成实验二中B代码的编写
21307062	郑越	验证程序实例中的相关程序

1、问题描述

(1) 验证程序实例中的相关程序；
(2) 编写程序，绘制下列信号的卷积波形。
A. 已知 $x1(t)=tu(t)$，$x2(t)=e^{-t}*u(t)$，求$x1(t)*x2(t)$；（要求：抽样频率$fs=1000$；时间$t=-1.1-2.1$）
B. 已知 $x[n]=[3,2,1,-2,1,0,4,0,3;n=0:8]$, $h[n]=[1,-2,3,-4,3,2,1;n=0:6]$;求$x[n]*h[n]$。
(3) 完成分析题：
A. 连续时间与离散时间信号的卷积定义分别是什么？卷积的作用是什么？
B. `conv` 函数只输出了卷积的结果，没有输出对于的时间向量，如何使得时间向量和卷积的结果对应起来?

2、问题分析

实验原理

在Python语言中，Scipy库中的scipy.signal可用于计算信号的卷积。
本实验Python库导入：

```
import numpy as np # 科学计算
import matplotlib.pyplot as plt # 画图
import scipy.signal as sg # 导入 scipy 的 signal 库 命名为 sg<br/>
```

编程计算卷积

A. 连续时间信号的卷积：求信号 $x1(t)=u(t)$与 $x2(t)=e^{-3t}u(t)$的卷积积分。注意：$x2(t)$是个持续时间无限长的连续时间信号，而计算机数值计算只能计算有限时长的离散信号，因此需要对 $x2(t)$进行抽样离散化，并且要使得$x2(t)$衰减到足够小。样例程序可见 `experiment_2_1.py`。

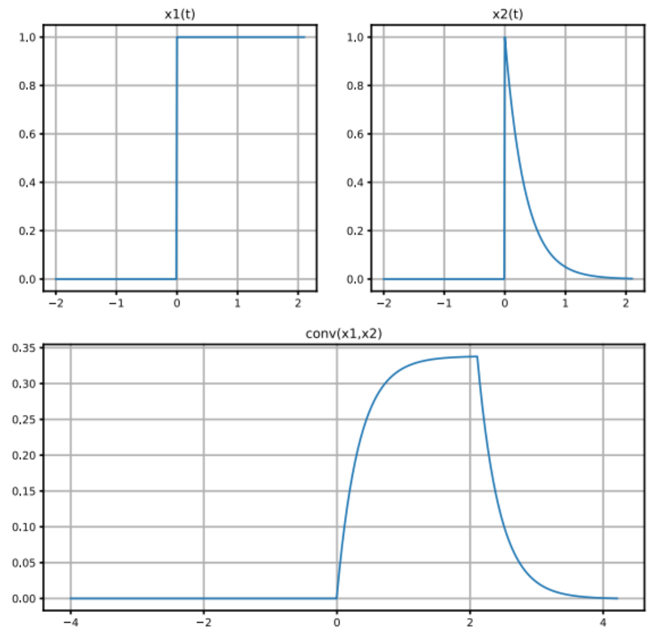


图 1 连续时间信号卷积图示

B. 离散时间信号的卷积：计算 $x[k] = [1, 2, 1, 1, 0, -3; k = 0, 1, 2, 3, 4, 5]$与$h[k] = [1, -1, 1; k = 0, 1, 2]$的卷积和。样例程序见`experiment_2_2.py`。

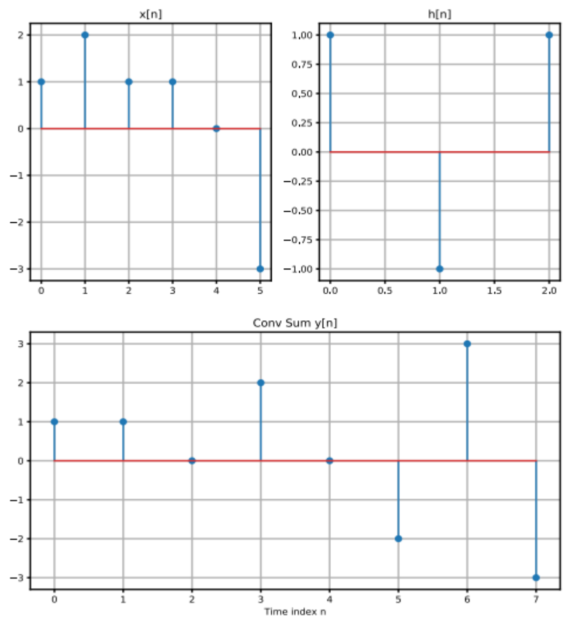


图 2 离散时间信号卷积图示

3、实验代码

(1) 验证程序实例中的相关程序

A. 连续时间信号的卷积：求信号 $x1(t) = u(t)$与$x2(t) = e^{-3t}u(t)$的卷积积分。

```
# 导入 需要的 library 库
import numpy as np # 科学计算
import matplotlib.pyplot as plt # 画图
import scipy.signal as sg # 导入 scipy 的 signal 库 命名为 sg

fs = 100 # 采样频率，注意和时间轴右端点结合使用，用于控制右端点的范围
t1 = np.array([t/fs for t in range(-200,211)])
 # t in [-2.0, 2.1]
 # 时间序列，注意右端点控制在2.1范围内，即2.1=211/fs
x1 = np.array([1 if t>=0 else 0 for t in t1]) # 定义x1(t)阶跃信号
t2 = np.array([t/fs for t in range(-200,211)])
 # t in [-2.0, 2.1]
 # 时间序列，注意右端点控制在2.1范围内，即2.1=211/fs
x2 = np.array([np.exp(-3*t) if t>=0 else 0 for t in t1]) # 定义x2(t)信号
y1 = sg.convolve(x1,x2)/fs # 卷积
n = len(y1) # 卷积结果采样点数量
tt = np.linspace(-400,421,n)/fs
```

```
# 定义新序列时间范围，卷积结果时间轴
# 卷积左端点=x1左端点+x2左端点，卷积右端点=x1右端点+x2右端点-1，
fig, axs = plt.subplots(2, 2, figsize=(10, 10)) #通过figsize调整图大小
plt.subplots_adjust(wspace = 0.2, hspace = 0.2)
# 通过wspace和hspace调整子图间距
plt.subplot(221) # 绘制x1(t)信号的子图
plt.plot(t1,x1) # 绘制x1(t)信号
plt.grid() # 显示网格
_ = plt.title('x1(t)') # x1(t)信号title
plt.subplot(222) # 绘制x2(t)信号的子图
plt.plot(t2,x2) # 绘制x2(t)信号
plt.grid() # 显示网格
_ = plt.title('x2(t)') # x2(t)信号title
plt.subplot(212) # 绘制卷积信号的子图
plt.plot(tt,y1) # 绘制卷积信号
plt.grid() # 显示网格
_ = plt.title('conv(x1,x2)') # 卷积信号title
plt.show() # 显示图像
```

B. 离散时间信号的卷积：计算 $x[k]=[1,2,1,1,0,-3;k=0,1,2,3,4,5]$与 $h[k]=[1,-1,1;k=0,1,2]$的卷积和。

```
# 导入 需要的 library 库
import numpy as np # 科学计算
import matplotlib.pyplot as plt # 画图
import scipy.signal as sg # 导入 scipy 的 signal 库 命名为 sg

n1 = np.linspace(0,5,6) # 时间序列[0 1 2 3 4 5]
x1 = [1,2,1,1,0,-3] # 信号x[n]
fig, axs = plt.subplots(2, 2, figsize=(10, 10)) # 通过figsize调整图大小
plt.subplot(221) # 绘制x[n]信号的子图
plt.stem(n1,x1,'-',use_line_collection=True) # 绘制x[n]信号
plt.grid(True) # 显示网格
_ = plt.title('x[n]') # 信号x[n] title

n2 = np.linspace(0,2,3) # 时间序列[0 1 2]
x2 = [1,-1,1] # 信号h[n]
plt.subplot(222) # 绘制h[n]信号的子图
plt.stem(n2,x2,'-',use_line_collection=True) # 绘制h[n]信号
plt.grid(True) # 显示网格
_ = plt.xticks(np.arange(0, 3, step=1.0)) # 设置横坐标间隔
_ = plt.title('h[n]') # 信号h[n] title

plt.subplot(212) # 绘制卷积信号的子图
y = sg.convolve(x1, x2,'full') # 使用 scipy.signal 的卷积函数 convolve
n3 = np.linspace(0,7,8)
plt.stem(n3,y,'-',use_line_collection=True) # 绘制卷积信号
plt.grid(True) # 显示网格
_ = plt.title('Conv Sum y[n]') # 卷积和信号y[n] title

plt.xlabel('Time index n') # 时间轴
plt.subplots_adjust(top=1, wspace=0.2, hspace=0.2) # 调整视图
plt.show() # 显示图像
```

(2) 编写程序，绘制下列信号的卷积波形。

A. 已知 $x1(t) = tu(t)$，$x2(t) = e^{-t}u(t)$，求$x1(t) * x2(t)$；（要求：抽样频率 $fs = 1000$；时间 $t = -1.1 - 2.1$）

```python
# 导入 需要的 library 库
import numpy as np # 科学计算
import matplotlib.pyplot as plt # 画图
import scipy.signal as sg # 导入 scipy的 signal 库 命名为sg

n1 = np.linspace(0,8,9) # 时间序列[0 1 2 3 4 5 6 7 8]
x1 = [3,2,1,-2,1,0,4,0,3] # 信号x[n]

n2 = np.linspace(0,6,7) # 时间序列[0 1 2 3 4 5 6]
x2 = [1,-2,3,-4,3,2,1] # 信号h[n]

plt.subplot(212) # 绘制卷积信号的子图
y = sg.convolve(x1, x2,'full') # 使用 scipy.signal 的卷积函数 convolve
n3 = np.linspace(0,14,15)
plt.stem(n3,y,'-',use_line_collection=True) # 绘制卷积信号
plt.grid(True) # 显示网格
_ = plt.xticks(np.arange(0, 15, step=1.0))
_ = plt.title('Conv Sum y[n]') # 卷积和信号y[n] title

plt.xlabel('Time index n') # 时间轴
plt.subplots_adjust(top=1, wspace=0.2, hspace=0.2) # 调整视图
plt.show() # 显示图像
```

B. 已知 $x[n] = [3,2,1,-2,1,0,4,0,3; n = 0:8]; h[n] = [1,-2,3,-4,3,2,1; n = 0:6]$;求 $x[n] * h[n]$。

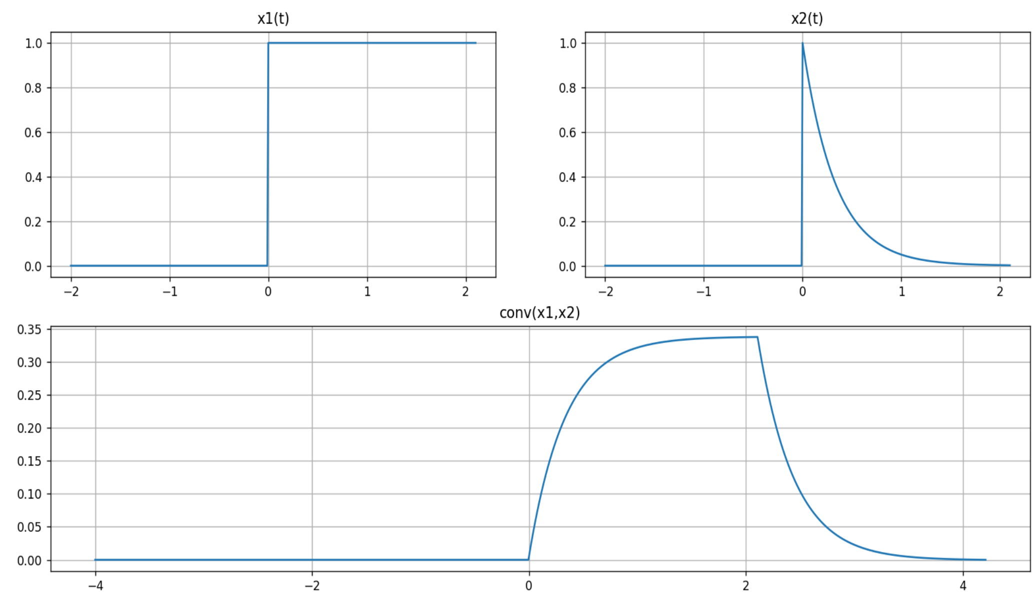
```python
# 导入 需要的 library 库
import numpy as np # 科学计算
import matplotlib.pyplot as plt # 画图
import scipy.signal as sg # 导入 scipy 的 signal 库 命名为 sg

fs = 1000 # 采样频率，注意和时间轴右端点结合使用，用于控制右端点的范围
t1 = np.array([t/fs for t in range(-1100,2101)]) # t in [-1.1, 2.1]
 # 时间序列，注意右端点控制在2.1范围内，即2.1=211/fs
x1 = np.array([t if t>=0 else 0 for t in t1]) # 定义x1(t)阶跃信号
t2 = np.array([t/fs for t in range(-1100,2101)]) # t in [-1.1, 2.1]
 # 时间序列，注意右端点控制在2.1范围内，即2.1=211/fs
x2 = np.array([np.exp(-t) if t>=0 else 0 for t in t2]) # 定义x2(t)信号
y1 = sg.convolve(x1,x2)/fs # 卷积
n = len(y1) # 卷积结果采样点数量
tt = np.linspace(-2200,4201,n)/fs
 # 定义新序列时间范围，卷积结果时间轴，
 # 卷积左端点=x1左端点+x2左端点，卷积右端点=x1右端点+x2右端点-1,
plt.subplot(212) # 绘制卷积信号的子图
plt.plot(tt,y1) # 绘制卷积信号
plt.grid() # 显示网格
_ = plt.title('conv(x1,x2)') # 卷积信号title
plt.show() # 显示图像
```

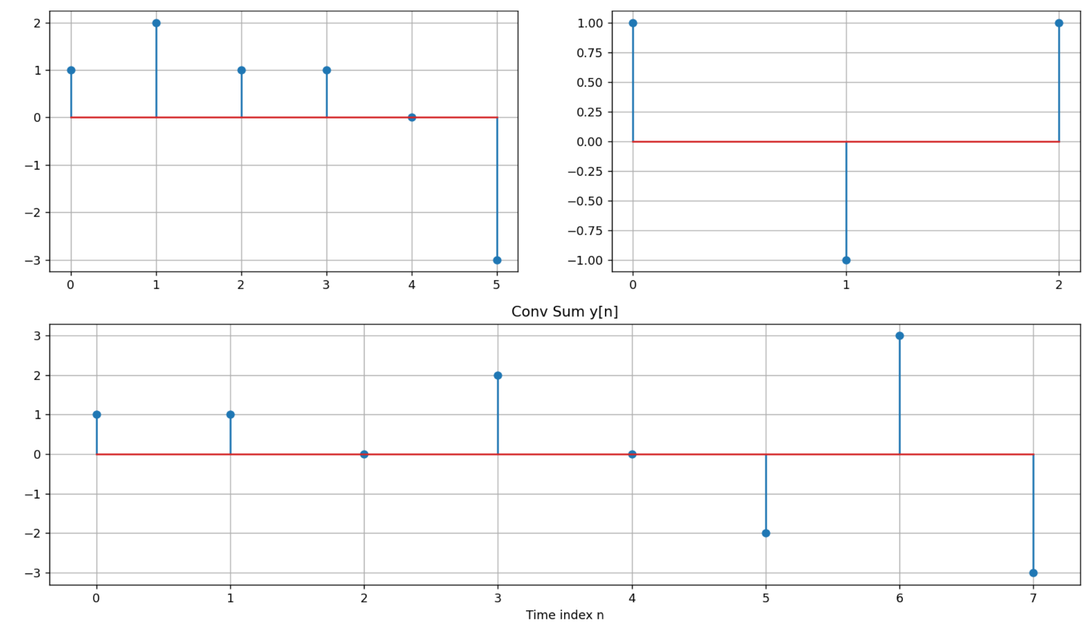
4、实验结果

(1) 验证程序实例中的相关程序

A. 连续时间信号的卷积：求信号 $x1(t) = u(t)$与 $x2(t) = e^{-3t}u(t)$的卷积积分。

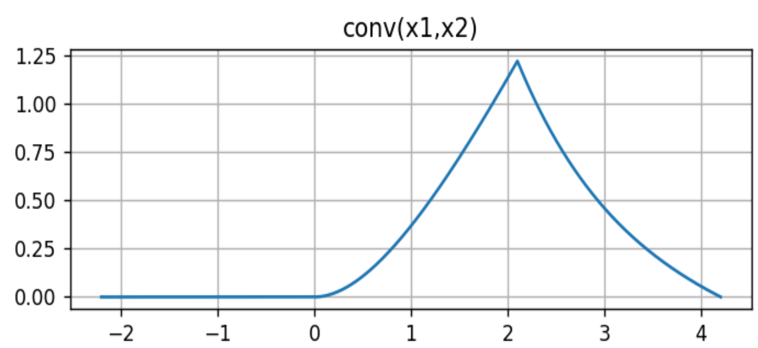


B. 离散时间信号的卷积： 计算 $x[k] = [1, 2, 1, 1, 0, -3; k = 0, 1, 2, 3, 4, 5]$ 与 $h[k] = [1, -1, 1; k = 0, 1, 2]$的卷积和。

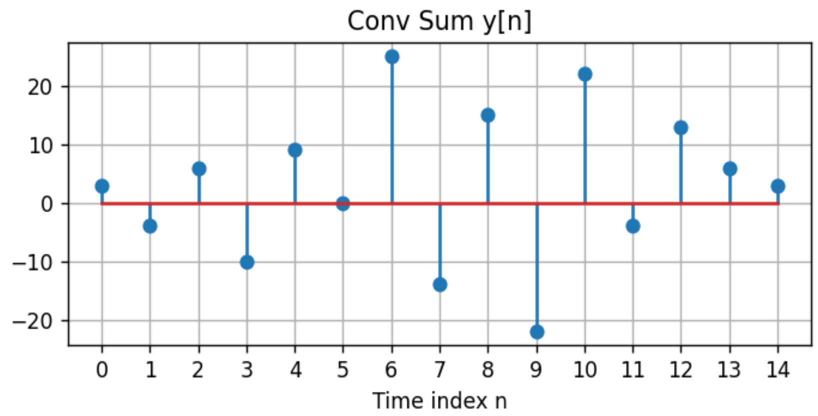


(2) 编写程序，绘制下列信号的卷积波形。

A. 已知 $x1(t) = tu(t)$，$x2(t) = (e^{-t}) * u(t)$，求$x1(t) * x2(t)$；（要求：抽样频率 $fs = 1000$；时间 $t = -1.1 - 2.1$）



B. 已知 $x[n] = [3, 2, 1, -2, 1, 0, 4, 0, 3; n = 0 : 8]; h[n] = [1, -2, 3, -4, 3, 2, 1; n = 0 : 6]$;求 $x[n] * h[n]$。



(3)分析题

A.连续时间与离散时间信号的卷积定义分别是什么？卷积的作用是什么？

连续时间信号的卷积定义：

给定两个连续时间信号$f(t)$和$g(t)$，其卷积$y(t)$定义为：

$$\begin{gathered} y(t)=\int_{-\infty}^{\infty} f(\tau) g(t-\tau) d \tau \\ y(t): \text{卷积的结果} \\ f(t)\text{和}g(t): \text{卷积的两个输入信号} \\ t: \text{时间变量} \\ \tau: \text{积分变量} \end{gathered} \tag{1}$$

离散时间信号的卷积定义：

给定两个离散时间信号$f[n]$和$g[n]$，其卷积$y[n]$定义为：

$$\begin{gathered} y[n]=\sum_{m=-\infty}^{\infty} f[m] g[n-m] \\ y[n]: \text{卷积结果} \\ f[n]\text{和}g[n]: \text{卷积的两个输入信号} \\ n: \text{离散时间变量} \\ m: \text{求和变量} \end{gathered} \tag{2}$$

卷积的作用：

卷积的作用是将一个函数或者信号与另一个函数或者信号进行相乘并求和，以得到它们之间的关系。

例如：卷积可以在不同领域中用来提取数据的特征

- 在图像处理中，卷积可以用来提取图像中的特征，例如边缘、纹理等。卷积操作将图像与一个卷积核进行卷积运算，可以使得图像中某些部分的像素值受到更强的影响，从而突出特定的特征。
- 在语音识别和自然语言处理中，卷积可以用来提取语音信号和文本的特征，例如声谱图、语音识别中的MFCC特征、文本中的n-gram特征等。卷积操作将语音信号或文本与一组卷积核进行卷积运算，可以提取出不同尺度的特征，从而捕捉数据中的局部特征。

B.conv 函数只输出了卷积的结果，没有输出对于的时间向量，如何使得时间向量和卷积的结果对应起来？

1. 手动生成时间向量：可以手动生成一个时间向量，使其与卷积结果具有相同的长度和采样间隔。

   例如：

```
convolution = np.convolve(x1, x2) / fs
conv_time = np.arange(start, end,1 / fs)[:len(convolution)]
plt.plot(conv_time,convolution)
```

使用 `np.arange` 函数生成时间向量 `conv_time`，其中：

- `start` 表示时间向量的起点
- `end` 表示时间向量的终点
- `1/fs` 表示时间向量的采样间隔，即每个采样点之间的时间间隔；
- `[:len(convolution)]` 表示截取时间向量的前面一部分，使其长度与卷积结果相同。

  最终生成的 `conv_time` 可以与 `convolution` 对应用于绘制卷积结果的图像。

2. 使用 `convolve` 函数：`scipy.signal` 库中的 `convolve` 函数可以同时输出卷积结果和对应的时间向量

   例如，在`experiment2_2.py`中

```
y = sg.convolve(x1, x2, 'full')  # 使用 scipy.signal 的卷积函数 convolve
n3 = np.linspace(0, 7, 8)
plt.stem(n3, y, '-', use_line_collection=True)  # 绘制卷积信号
```

其中，`'full'` 表示输出完整的卷积结果

5、结论

- 卷积是一种基本的信号处理操作，可以用于信号滤波、信号重构、信号分析等方面。
- 卷积的计算方法有多种，可以通过积分计算连续信号的卷积，也可以通过离散求和计算离散信号的卷积。
- 卷积操作具有交换律、结合律和分配律等基本性质，这些性质可以方便地用于卷积的计算和推导。
- 在计算卷积时，需要注意采样频率和时间范围等参数的选择，以保证结果的准确性和可靠性。
- 通过编写程序和绘制图形，可以更加直观地理解卷积的概念和性质，这有助于加深对信号处理的理解和掌握。

6、收获与感想

邓栩瀛：

通过这次实验，我更加深入地了解卷积的概念和性质，以及如何计算卷积。通过编写程序和绘制图形，更加直观地理解卷积，并且加深对信号处理的理解。同时，在处理信号时需要注意采样频率和时间范围等参数的选择，以保证结果的准确性和可靠性。

钟欣余：

完成信号处理的代码实现的过程中，我不仅学到了在计算卷积时，需要注意采样频率和时间范围等参数的选择，以保证结果的准确性和可靠性，还知道了在进行卷积的计算和推导过程中，可以利用卷积操作交换律、结合律和分配律等基本性质，这些性质可以使计算更加方便，对信号处理的理解有所加深，收获满满。

金思琪：

这次实验，在实验一的基础上，结合运用卷积的知识，我通过Python绘制了离散信号的卷积波形。我们可以通过离散求和的方式，计算离散信号的卷积波形。通过程序绘制波形，我们可以更加直观地理解卷积的概念和性质，从而加深对信号处理的理解和掌握。

郑越：

通过本次实验，对卷积有更进一步的了解，连续信号的卷积通过积分计算，离散信号的卷积通过离散求和计算，并且通过绘制卷积波形可以更直观地了解卷积的性质，包括交换律、结合律和分配律。