时域数字信号处理（4）

基本信号与基本算法

2024-9-24

1

信号的表示方法(章节2.1）

- 在本章的学习中我们以一维信号为学习对象
- 连续信号的表示 $f(t)$
- 对应的离散信号被表示成数字序列 x[n], 其中x[n] 是对$f(t)$的*取样*。
- 典型信号或序列的取样值，其中 n 表示范围落在 $-\infty \le n \le \infty$或者$0 \le n \le \infty$或者$-\infty \le n \le 0$的整数。
  - 注1：n表示第几个采样数据，单位是“个”，也就是没有单位，具体换算见采样率计算一节
  - 注2：$n < 0$指的是记录器开始之前的数据，而$n = \infty$表明“未知”

2024-9-24 2

离散时间信号的书写表示

- 离散时间信号也可以写成在大括号内的数字序列：

$$\{x[n]\}=\{\ldots,-0.2,\underset{\uparrow}{2.2},1.1,0.2,-3.7,2.9,\ldots\}$$

- 在上式中， $x[-1]=-0.2,\ x[0]=2.2,\ x[1]=1.1,$
- 离散信号序列通常罗列出了真实采集的信号，此时n的范围必须标注。
- 标注方法有：
  - 箭号被放在时间为 $n=0$ 的取样点下方，如上图。
  - 给出范围，如 $n=-3:5$,或者$-1<n<\infty$, 当x[n]是无限长序列时，只能用后面这一种或者箭头标注
  - 根据教材的不同， 有$x[n],\{x[n]\}$，极少时有$x\{n\}$，少量时候不加区分的写作$x(n)$ （连续信号）， X[k]（大写表示频域）

离散时间的长度（章节2.1.1）

- 离散时间分为两种，有限长信号和无限长信号
- 无限长信号 $n=-\infty:N$(正确写法 $n\le N$), $n\ge N$, $n=-\infty:+\infty$ ，分别叫左边序列/右边序列/双边序列
  - 无限长信号现实中什么样子？
  - 无限长信号现实中怎么处理？
- 有限长序列例如：$n=-3:5$
- 区别：在下一节课学习中，有一些处理只能针对于有限长序列！（上下采样/循环位移）

2024-9-24

离散时间信号的编程

- 在编程处理中，一个序列的起始时间不一定是0或者1，
  - 例如$x[n]$, $n =-3:5$
  - 各个编程语言中的数组都是从0或者1 indexed
  - 需要额外的方法，例如：定义n数列、定义起始坐标+换算的方法进行处理。

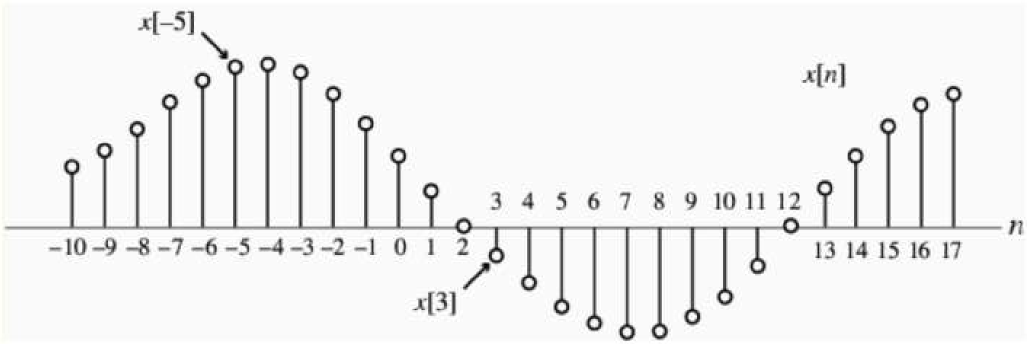
- 编程练习作业1：
- 定义一个数组表示信号，信号的起始时间不是0或者1。

2024-9-24

5

离散时间信号的图例

- 具实数取样值的离散时间信号图形表示如下：



- 注意：在图中每个值用孤立点或者点加直线表示，决不能每个点连在一起，那样是连续信号

2024-9-24

6

信号的表示（拓展）：消除对高维信号的恐惧

- 信号维度不受限制
  - 多维信号还可以分为单一信号源的多维信号和多源的多维信号。
- 本课程理论学习主要讨论1维信号，但在实践中，不可避免接触2维（图像），3维（彩色图像）以及更高维度(视频）信号
- 我们对不同维度进行简单展示

$$y = f(x_1, x_2, x_3, \ldots)$$

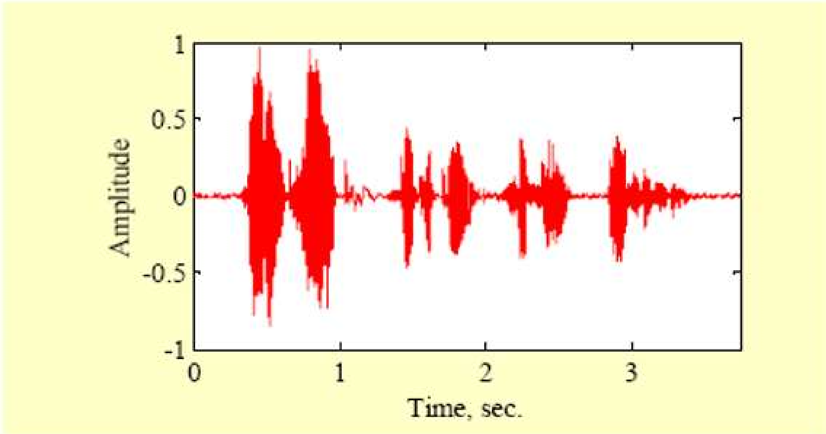
2024-9-24

7

信号的维度展示-1维信号

- 一维 (1-D D-Dimension) 信号是一个单一独立变量的函数。
- 语音信号就是一维信号的例子，其独立变量是时间。
- 在现实中可能会碰到双通道的采样声音，就会变成2维信号

$$A = f(t)$$



信号的维度-2维灰度图像

- 影像信号，如照片，就是二维信号的例子，其独立变量为两个空间变量，X和Y点坐标

$$I = f\left(x, y\right)$$

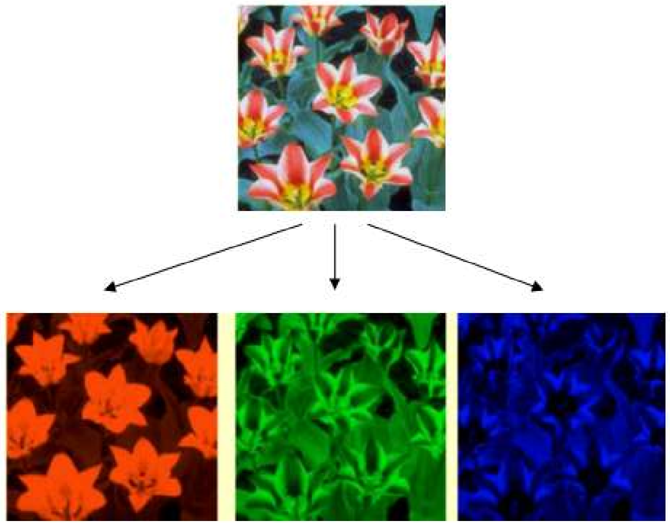
提示：对于现实中的信号，例如这张图像，不存在一个“规范”的函数，可以把图像写成函数表达的$f(x,y)$，真实图像只能是一个2维数据矩阵

但给出一个函数型的表达可以使我们在表示信号时容易很多。

雨课堂 Rain Classroom

信号维度-3维彩色图像

- 彩色图像信号包含了三个二维信号，分别表示三个主要颜色（红色、绿色及蓝色，RGB）。



$$(R, G, B) = f(x, y)$$

或者

$$v = f(x, y, c)$$

C是颜色通道channel的缩写字头，通道一词来自信号处理

2024-9-24

信号维度-4维信号

- 视频信号的根本原理是连续播放图像，每秒n帧（Frame Per Second FPS）
- 灰度视频信号的每一个帧是2维的图像信号，视频则是3维信号

$$V = f(x, y, t)$$

- 彩色视频信号的每一帧是一个3维彩色图像信号，彩色视频则是4维信号

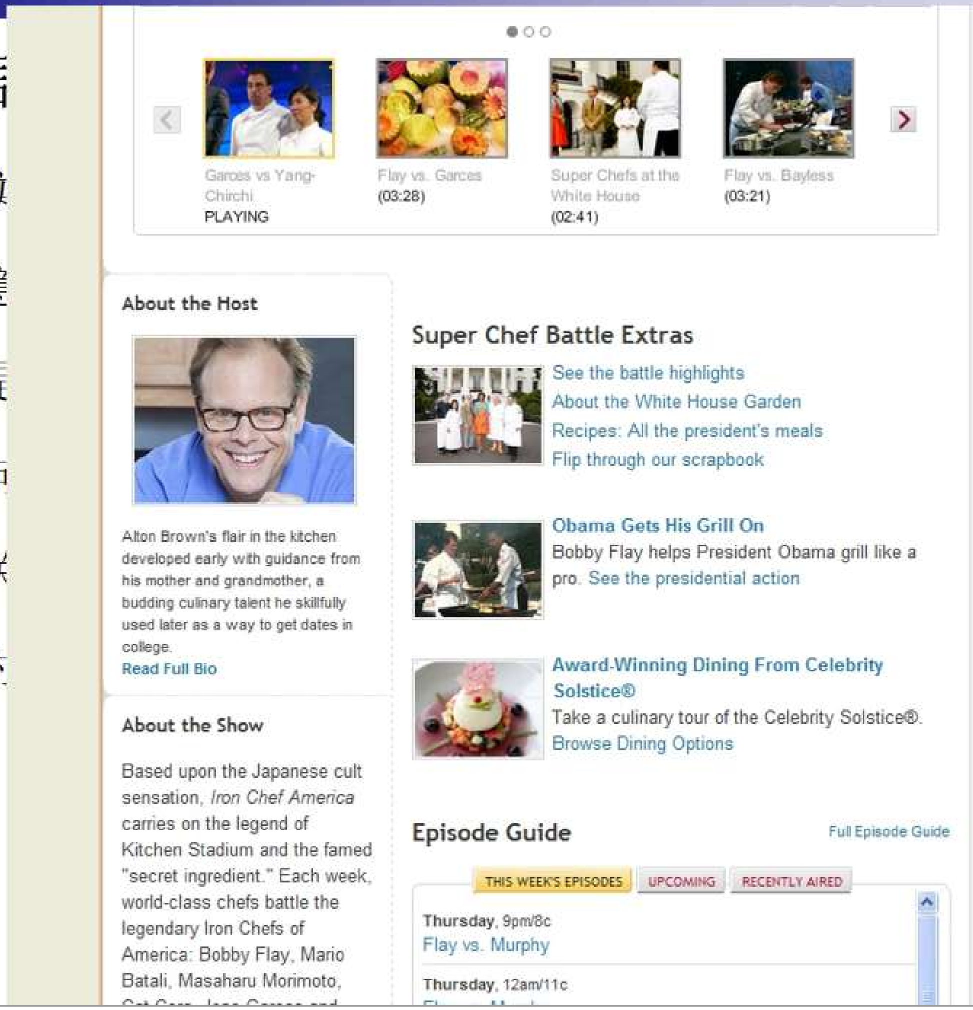
$$V = f(x, y, c, t)$$

信号维度的学习与理解

- 我们上课学习的算法都是处理1维信号的。
- 对于高维信号，很多算法会有升维的运算
  - 比较经典的有2维卷积、2维傅立叶变换等
  - 这种情况下，需要大家自行扩展学习，上课不做特别讲解，但本身难度不高
- 很多情况下，如果各个维度具有独立性，或者如果我们只希望观测某一个维度方向的性质，我们也可以把高维信号使用一维方法处理。
- 如下面的例子

4维信

- 真实
- 潜意
- 但是
- 如何
- 相关
- 此时



编辑词条

肖费者的某

理论表
他让老
弱的强度
有意识的
信息。
超过了5

因为它与
电影故

他们购买

中 "Absol

13

2024-9-24

几个基本的信号分类

- 除连续、离散的分类方法外，还有（但不是所有）信号分类方法有：
  - 实数 (real) 函数或复数 (complex) 函数：
    - 关于复数信号的一些说明
  - 信号如果是由单一个信号源产生（的一维变量），则称为纯量 (scalar) 信号。
  - 信号也可以由多个信号源产生且记为多维变量则称为向量 (vector) 信号或多频道 (multi-channel) 信号。
    - 注意区分多频道的多维信号与混合信号

基本处理方法（章节2.2）

- 基本处理包括
  - 多信号处理：加法、数乘、乘法
  - 单一信号处理：补零、时移、反折、拉伸
- 相对复杂处理：
  - 微分、积分、卷积*、相似计算

2024-9-24

基本处理-补零（1）

- **补零是基本的处理操作**
  - 对于不同长度或不在统一时间范围的序列间的运算，必须先将不一致的部分序列做适当的补零运算，使这些序列拥有相同的时间 n 范围。
  - 例1：$\mathbf{x[n] = \{-2\ \ 1\ \ -3\},\ 0 \le n \le 2}$ 与 $\mathbf{y[n] = \{3\ 4\ 6\ -9\ 0\}\ 0 \le n \le 4}$ 进行（加法）运算
    - 需要将 $x[n] = \{-2\ \ 1\ \ -3\}$ 后面补零为 $x[n] = \{-2\ \ 1\ \ -3\ 0\ 0\}$
  - 例2：$\mathbf{x[n] = \{-2\ \ 1\ \ -3\},\ n = -1{:}1}$ 与 $\mathbf{y[n] = \{3\ 4\ 6\}\ n=0{:}2}$ 进行（加法）运算
    - 需要将 $x[n] = \{-2\ \ 1\ \ -3\}$ 后面补零为 $x[n] = \{-2\ \ 1\ \ -3\ 0\}$，$n=-1{:}2$
    - 将 $y[n] = \{3\ 4\ 6\}$ 前面补零为 $y[n] = \{0\ \ 3\ \ 4\ \ 6\}$，$n=-1{:}2$

多选题 1分

两个序列进行运算，其中$x[n]=\{4\ \ 2\ \ 4\ 1\}$，$y[n]=\{3\ 1\}$，

则合理的补零方法有若

- A $y[n]=\{0, 0, 3, 1\}$
- B $y[n]=\{0, 3, 1, 0\}$
- C $y[n]=\{3, 1, 0, 0\}$
- D 还有好多，写不完
- E 以上都不对

基本处理-补零（2）编程

- **补零是一个书写简单，但是编程困难的问题**
  - 补零需要改变用于存储序列的数组长度，这在大部分编程中都是较为困难的。
  - 如何处理序列变化长度需要自行研究
  - 别忘了改变序列长度后，序列时间index也要做相应处理

- 编程练习作业2：
- 定义一个数组表示信号，对齐进行补零处理，并输出补零后的序列（同时输出时间指标）

2024-9-24

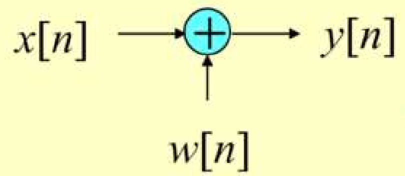
18

基本运算-加法1

- 加法运算：

$$f(t)=g(t)+k(t)$$

– Adder

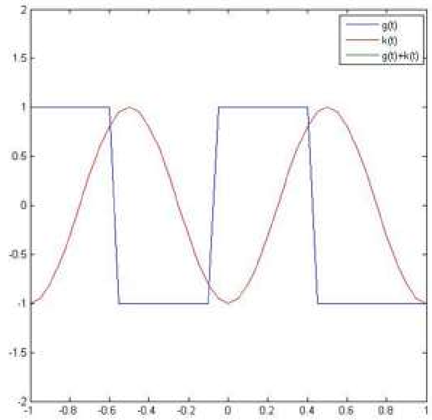


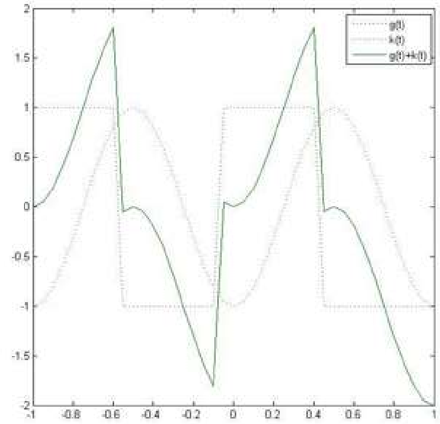
$$y[n]=x[n]+w[n]$$

2024-9-24

基本运算-加法3

- 加法运算：例子



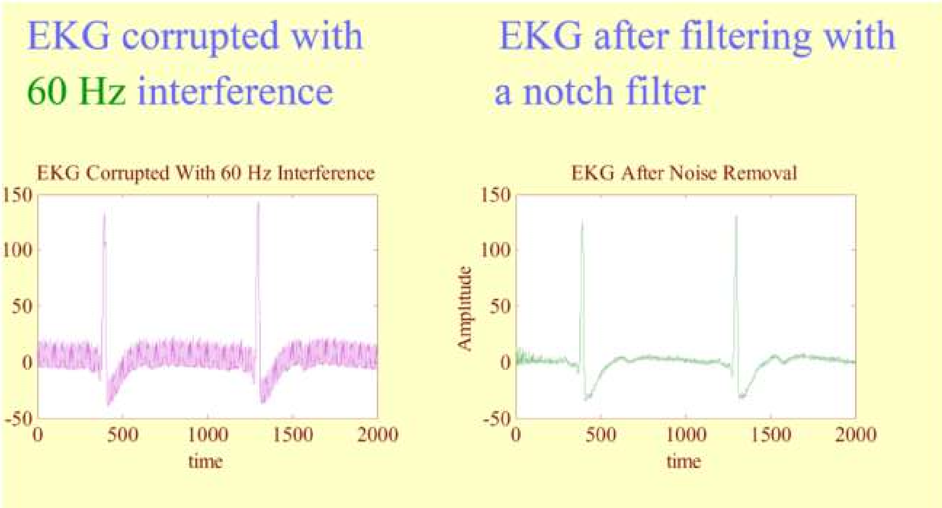


- 请问这个操作对于信号长度有要求不？

2024-9-24

基本运算-加法4

- 加法运算：应用例子，去噪



2024-9-24

基本运算-加法5

- 加法运算的一个简单应用：去噪--总集平均 (Ensemble Averaging)
  - 改进一个已被加成性随机噪声干扰的数据的质量。
  - 加成性随机噪声则可能是随机而且是无法重制的。
    - 随机噪声的一些假设：0均值，没有时间点上的关联性（白噪声）
  - 同时，数个测量信号中所包含未被干扰的数据向量 s 可能保持相同。
  - 假设 d_i表示干扰原资料向量 s 的第 i 次量测值的噪声序列：

$$x_i[n] = s[n] + d_i[n]$$

基本运算-加法6

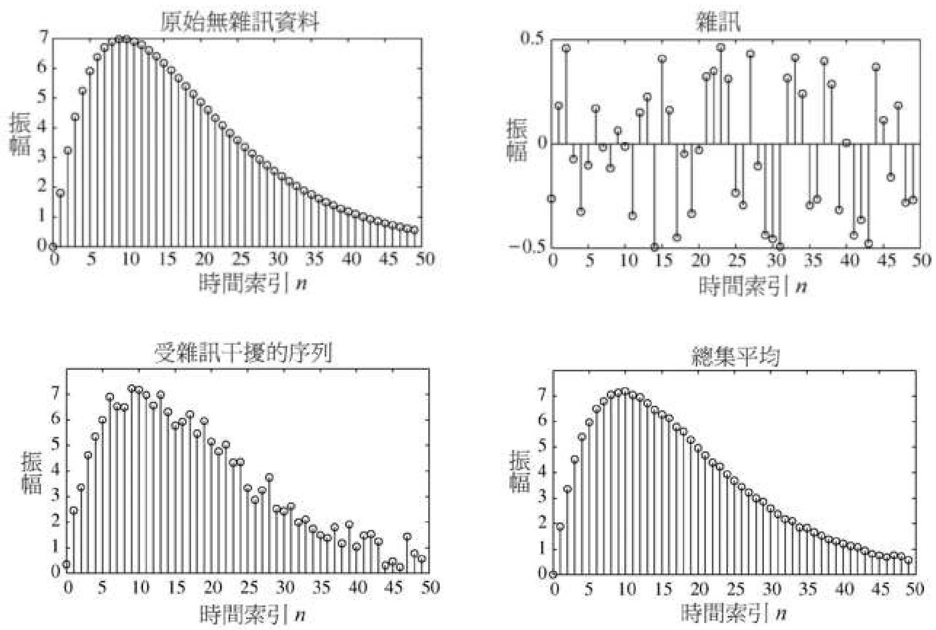
- 在 K 次量测后，计算一个平均数据向量，称为总集平均 (ensemble average)。

$$\mathbf{x}_{ave}=\frac{1}{K}\sum_{i=1}^{K}\mathbf{x}_i=\frac{1}{K}\sum_{i=1}^{K}(s+\mathbf{d}_i)=\mathbf{s}+\frac{1}{K}\sum_{i=1}^{K}\mathbf{d}_i$$

- 从噪声定义我们知道，$\sum_{i=1}^{k} d_i \approx 0$
- 对一个非常大的 K 值而言，X_{ave} 通常相当程度的近似于原资料向量 s。

基本运算-加法7

- 范例一



- 请问这个操作对于信号长度有要求不？

2024-9-24

24

基本运算-加法8

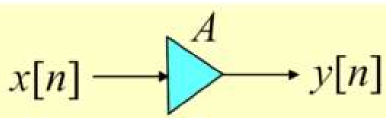
- 拓展：总集平均 要求K次重复测量，但绝大部分情况下信号是不可重复的
- 引入概念：滑动窗
- 连续观察一个信号，其在任意一个时间段内的信号，会有以下特征：
  - 噪声信号各时间点间的数值无关，且保持0均值。
  - 相邻信号具有一定的相似性
- 对于一个时间段（窗口）内的信号做加法和平均，然后延时间轴移动这个窗口
  - 这个概念在后续会进一步解释。
- 经验理解：加号两侧的信号往往是对等的（可加性），与后面的乘法不同。
- 请问：滑动窗对长度有要求不？

基本运算-乘法1

- 乘法运算：数乘

$$f(t) = k \cdot g(t)$$

– Multiplier



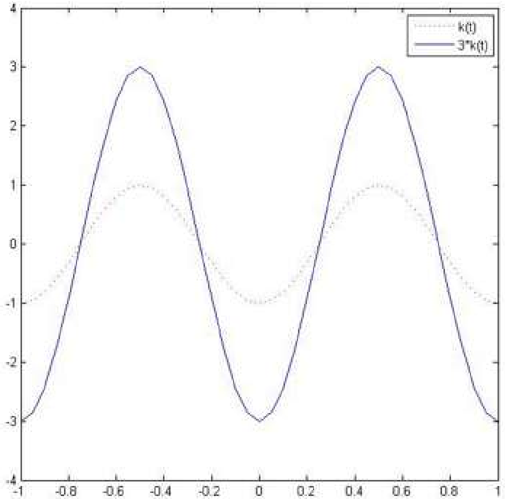
$y[n] = A \cdot x[n]$

2024-9-24

26

基本运算-乘法2

- 乘法运算：放大器



- 现实中的放大器往往也会将噪声等干扰一起放大，所以通常和其他处理配套使用。
- 放大器需要引入能量，在电路设计中通常代表需要提供新的能量源（功率放大器—功放）

2024-9-24

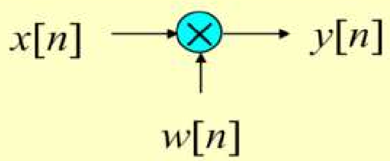
雨课堂 Rain Classroom

基本运算-乘法3

- 乘法运算：两个信号相乘

$$f(t) = k(t) \cdot g(t)$$

- 乘积 (调变，modulation) 运算：
  - Modulator



$$y[n] = x[n] \cdot w[n]$$

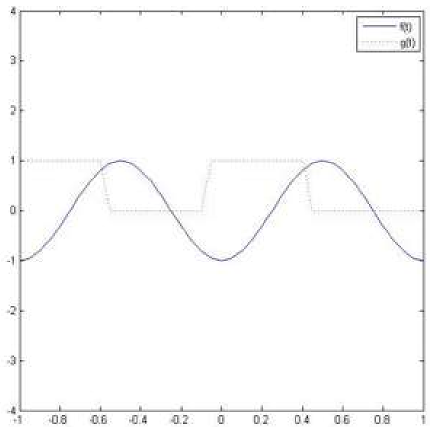
基本运算-乘法4

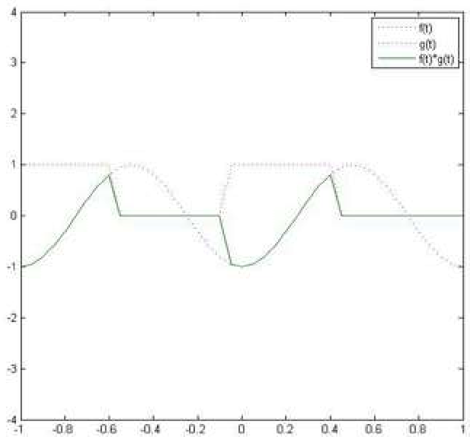
- 乘法在DSP中有着特殊的作用，通常表明的某种操作。
- 文字描述操作是无法量化计算的，例如
  - 屏蔽某频率的信号；放大某些局部信息；对当前时间点的信号，做延时间轴的弱化扩散处理
- 方法：将要实施的处理，转化成一个特定的信号，然后做乘法
  - 以及乘法后的**加法**和**滑动窗**处理，三个处理组成最核心的卷积运算。
- 结论：很多情况下，乘法的运算其中一方是特意设计的，表明某个效果的操作。
- 这个操作对长度有要求吗?

2024-9-24

基本运算-乘法5

- 乘法运算应用1：遮罩（masking）





遮罩最经典的作用就是之前谈到的分频通讯的接收信号环节。

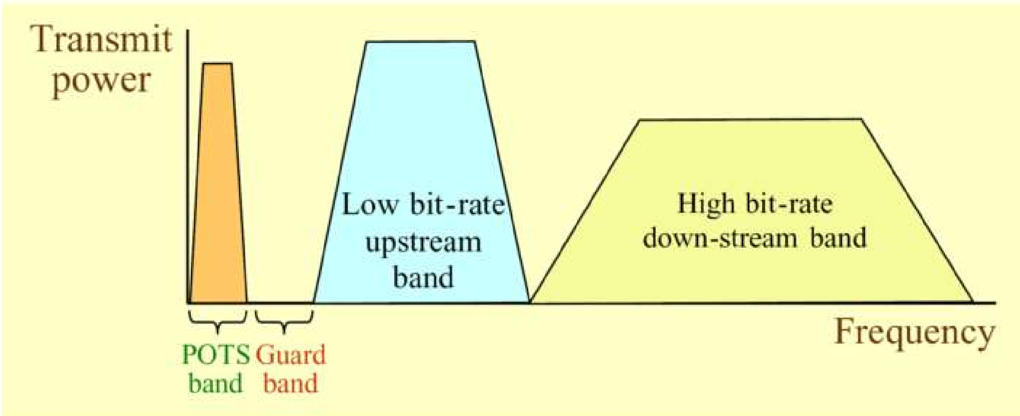
2024-9-24

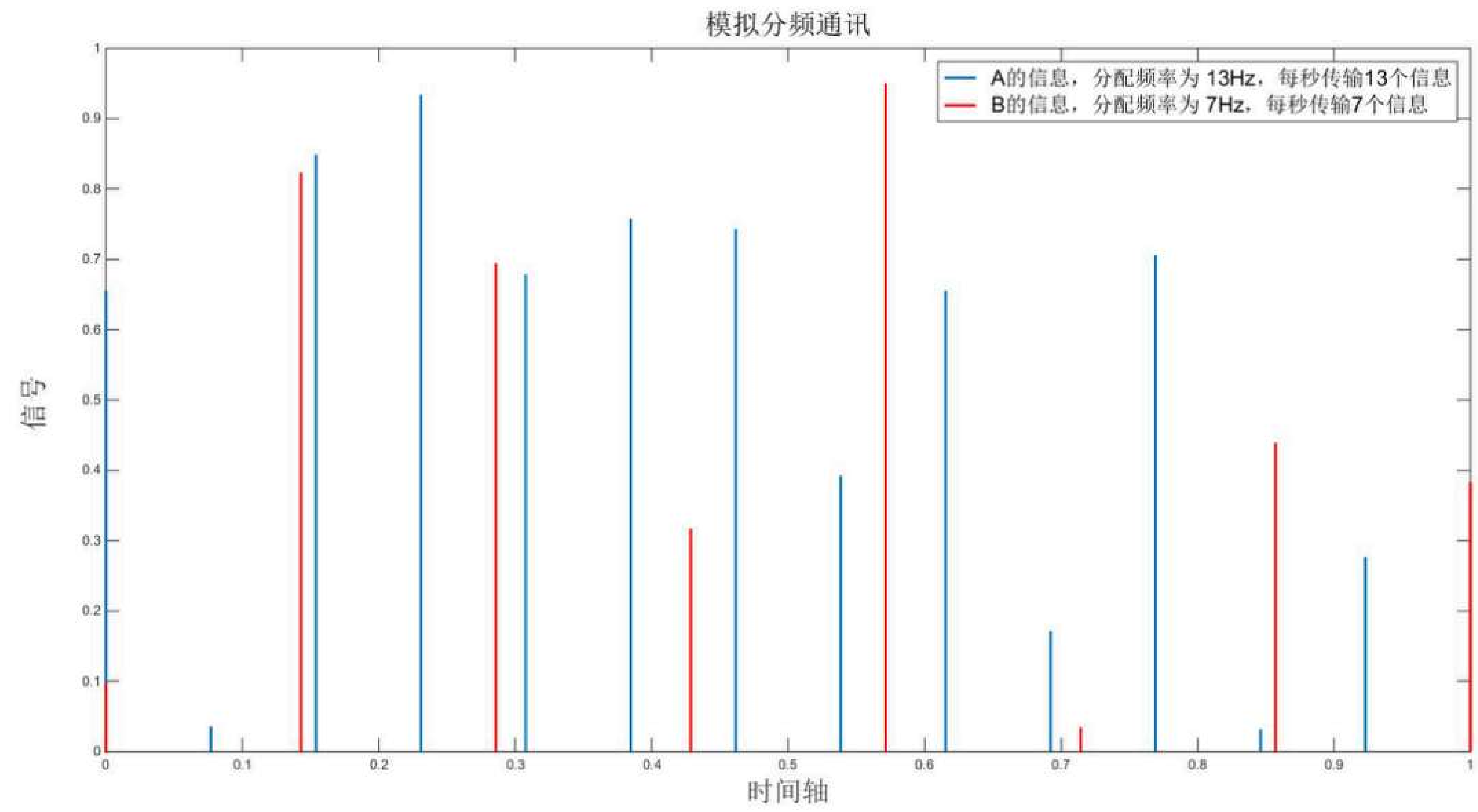
非对称数字用户回路 (ADSL)-1

- 属于区域传输系统，设计用于双绞线上同时提供三种服务：
  - 以最高 9 Mb/s 位率进行数据下载传输（数据朝向使用者）。
  - 以最高 1 Mb/s 位率进行数据上载传输（数据远离使用者）。
  - 原始电话服务 (POTS)。
- 以此为例，我们聊一聊分频通讯—加深对于频率的了解
  - 同一根双绞线却可以同时上传、下载和通话。
  - 天空中的无线电可以同时传播很多信号（2G-5G）
  - 现代通讯的基础

非对称数字用户回路 (ADSL)-2

- 一个分频的例子：分频多任务 (FDM) 架构的非对称数字用户回路，其频谱分配如下




模拟分频通讯
A的信息，分配频率为 13Hz，每秒传输13个信息
B的信息，分配频率为 7Hz，每秒传输7个信息
信号
时间轴
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

5G建设速度跟不上？美国要对中低频段下手了

人民邮电报
发布时间：19-07-14 10:54 | 人民邮电报

Verizon CEO：2020年将采用低频段在全国部署5G

暴走通信
发布时间：02-15 11:46

首页 资讯 产品 案例

美国发布5G高频段频谱新规划，中国该如何布局5G网络？

Eleanore 发布于 2年前 (2018-07-12) 分类：智能头条资讯 / 资讯 来源：通信世界

[导读] 近期，美国政府发布5G高频段（即毫米波频段）频谱的最新规划，至此，美国政府已经抢

固定时

越高，

SoC产

月上市

数量超

2(

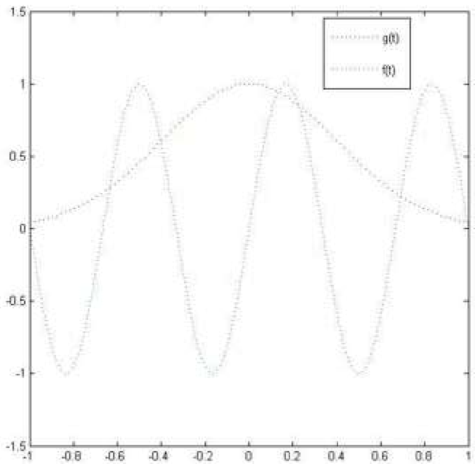
34

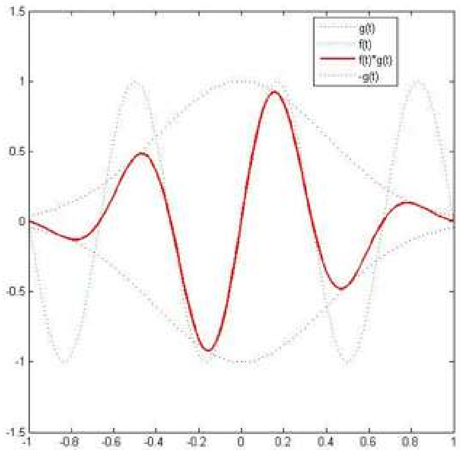
基本运算-乘法6

- 调制（ modulation ）：分频通讯中的发送环节
  - 在分频段通讯中，发送的信号本身不是周期的，所以也就没有频率的概念。
  - 分频发射需要将信号按照某些频率进行发送，所以要给该信号赋予频率的概念
- DSP中的调制是由调制解调器（modem）完成的。
  - 调制解调器，是调制器和解调器的缩写，一种计算机硬件 [1] ，它能把计算机的数字信号翻译成可沿普通电话线传送的模拟信号，而这些模拟信号又可被线路另一端的另一个调制解调器接收，并译成计算机可懂的语言。这一简单过程完成了两台计算机间的通信。
  - *张晓蕾.Modem的数据通信[J].科技资讯,2008(31):18. . 中国知网[引用日期2019-06-19]*
- 不同频率的脉冲信号，对应不同的频段。分频传输。

基本运算-乘法7

- 调制的展示
- 将高斯函数$g(t) = e^{-\frac{t^2}{2\sigma^2}}$ 用正弦函数$k(t) = \sin(3\pi \times t)$ 调制



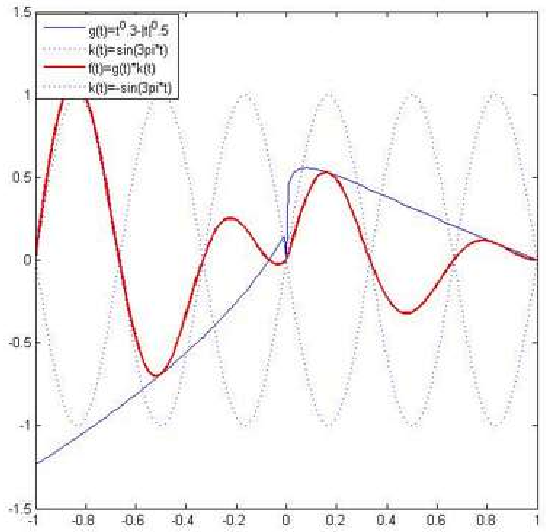


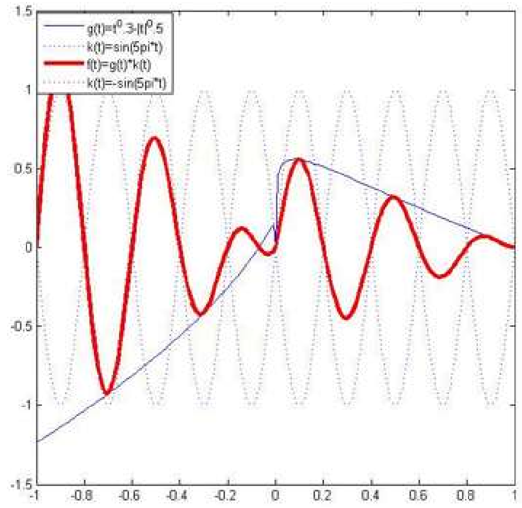
注：此处讨论的频率部分为震荡频率（详见模拟电路部分讨论）

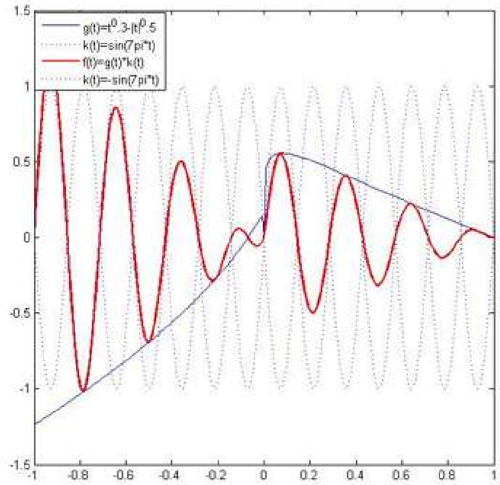
2024-9-24

基本运算-乘法8

- 用不同频率正弦信号调制一个任意模拟信号







2024-9-24

基本运算-加法与乘法对比

- 加法表现多代表了两个同等信号进行互相叠加（干扰）两个信号时平等的

$$x[n] = s[n] + d[n]$$

- 而乘法多表现在某个系统下，用一个信号作为一种操作来处理另一个信号。其中一个信号是手段而另一个信号时被处理的对象

$$f(t) = k(t) \cdot g(t)$$

基本运算-时移1

- 时间位移 (time-shifting) 运算：时移是一个单一信号的处理方法

$$g(t) = f(t - t_0)$$

- 离散格式$y(n) = x(n - N)$ 其中 N 是一个整数

若 N>0，称之为延迟 (delay) 运算：

– Unit delay

$$x[n] \longrightarrow \boxed{z^{-1}} \longrightarrow y[n] \qquad y[n] = x[n-1]$$

N<0，则称之为超前 (advance) 运算：

– Unit advance

$$x[n] \longrightarrow \boxed{z} \longrightarrow y[n] \qquad y[n] = x[n+1]$$

基本运算-时移2

- 已知x[n] = {1,2,1,3}, n =− 3: 0; y[n] = x[n − 2],求 y[n]
- **平移法**：将序列箭头标出，并做相应平移

$$x[n] = \{1 \quad 2\ 1 \quad \underset{\uparrow}{3}\}$$

- 减法，则箭头相应左移2位

$$y[n] = \{1 \quad \underset{\uparrow}{2}\ 1 \quad 3\}$$

– Unit delay $x[n] \longrightarrow \boxed{z^{-1}} \longrightarrow y[n]$ $\quad y[n]=x[n-1]$

2024-9-24

基本运算-时移3

x[n]		1	2	1	3
					↑
y[n]		1	2	1	3
			↑		

- 理解平移法：“*箭头移向了更早的时间点，所以是序列提前*” X 错误的！
- 正确的理解：箭头（0点是不动的）

x[n]		1	2	1	3		
					↑		
y[n]				1	2	1	3
					↑		

- 序列延后了

请问这个操作对于信号长度有要求不？

2024-9-24

41

基本运算-时移4

- 例题
- 已知$x[n] = \{1,2,1,3\},\ n =-3:0;\ y[n] = x[n-2]$,求 $y[n]$
- **罗列计算法：** 上述公式中公式左右两侧的n是同一个数值，所以有

$$y[-1] = x[-1-2] = x[-3] = 1$$
$$y[0] = x[0-2] = x[-2] = 2$$
$$y[1] = x[1-2] = x[-1] = 1$$
$$y[2] = x[2-2] = x[0] = 3$$
$$y[n] = \{1,2,1,3\},\ n =-1:2$$

- 原本信号3，发生在0时刻，现在发生在2时刻，发生的晚了，所以叫延迟

基本运算-时移5

- 平移法说明：平移法看似非常简单，但是当整体算式复杂了时候，及其容易出错，需要对整体序列变化有着清晰的处理，例如以下的例子

$$z[n] = x[n+3] - y[-2n-2]$$

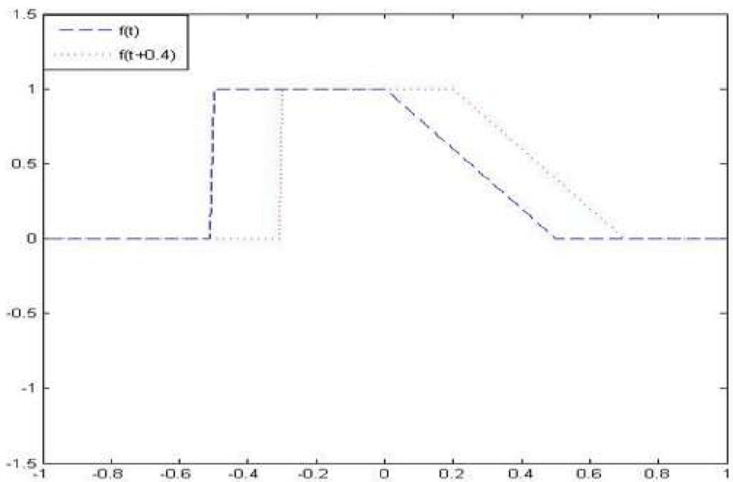
  - 在这个例子中，平移法分别要处理，平移，反转，拉伸三种操作，尤其是反转（$-2n-2$）、拉伸（$-2n-2$）、平移（$-2n-2$）的移动顺序，非常容易出错。

- 所以更多时候建议相对“笨”一些的方法，直接计算（保证没错）

2024-9-24

基本运算-时移6

- 时间位移 (time-shifting) 运算图例：



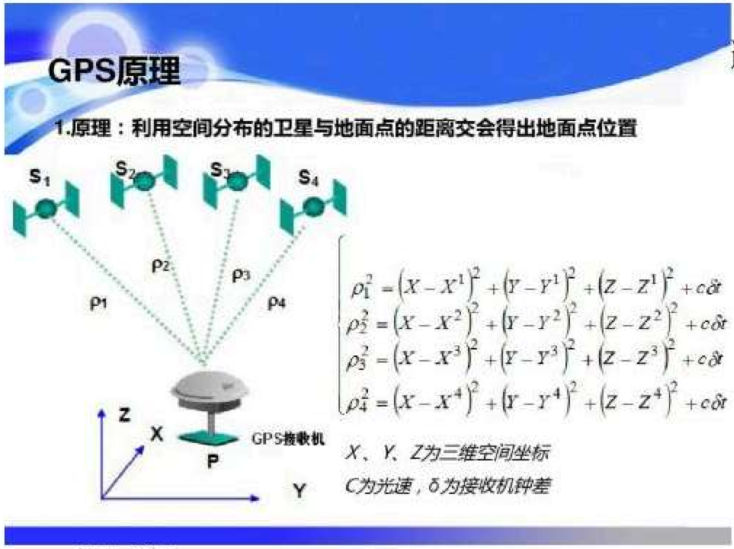
2024-9-24

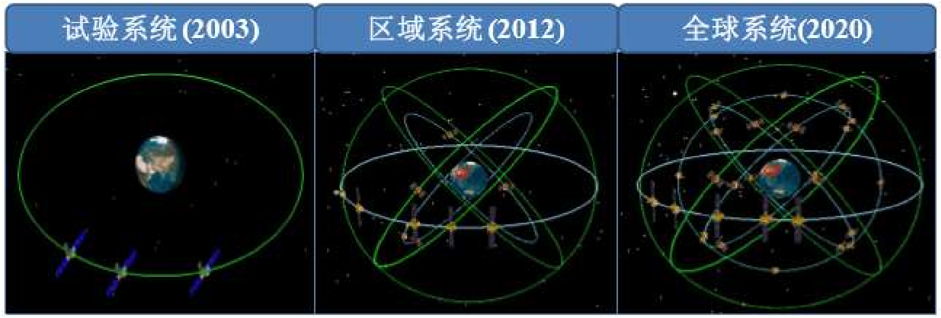
基本运算-时移7

- 应用:
  - 主动发射信号，并接受：激光定位，激

距离 = 延迟时间*速度/

进行发射:GPS





45

基本运

以

2024-9-24

基本运算-反折1

- 时间反向 (time-reversal) 或是折迭 (folding) 运算：

$$g(t) = f(-t)$$

- 离散格式　y[n] = x[-n]

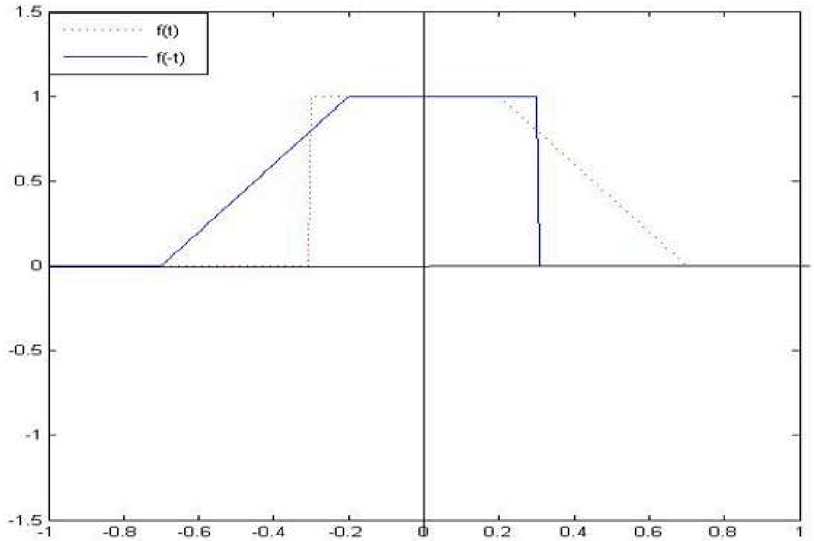
- 例子，$x[n] = \{1,2,4,3\}$, $n =- 1:2$, $x[n]$以$n = 0$做镜面对折，例如

x[n]			1	2	4	3
				↑		
x[-n]		3	4	2	1	
				↑		

2024-9-24

基本运算-折叠2

图例:请注意，对于x[-n]，需要以x[n=0] 做反转



基本运算-时域尺度变换（拉伸）1

- 拉伸 运算：$g(t) = f(k \cdot t)$
- 离散格式：$y(n) = x(kn)$
- 拉伸变化我们称为采样率（上采样和下采样）的变化，其中k通常采取以下定义：
  - 若$k > 1$，则k通常为正数，即 $y(n) = x(kn), k > 1, k \in Z$
  - 若$0 < k < 1$，则通常找到一个正整数L, 使得 $y(n) = x(kn) = x(\frac{n}{L}), L > 1, L \in Z$
  - 若$k < 0$,则先处理$-k$，然后在做反折
  - 通常对于复杂的变化，例如$y(n) = x\left(\frac{7}{9}n\right)$，我们定义为二次采样（采样率变化），不在拉伸部分讨论，把它单独作为一个系统。

2024-9-24

基本运算-时域尺度变换 上采样 2

- 若以整数因子L>1来进行上采样，计算公式见下图。
  - 除法是上采样，越除得到的数列越长。而“上”采样的上字与长短对应，上采样变长。
  - 对于x[0.5]之类的index不为整数的项，我们直接定义其为0
  - 为0的部分，通常通过内插算法（interpolation）后续将其补上
  - 在上述计算中，仍然左右不变的是n项，可以使用列举法进行计算。

$$x_u[n] = \begin{cases} x[n/L], & n = 0, \pm L, \pm 2L, \cdots \\ 0, & \text{otherwise} \end{cases}$$

$$x[n] \longrightarrow \boxed{\uparrow L} \longrightarrow x_u[n]$$

2024-9-24

基本运算-时域尺度变换（上采样）3 习题

- 按照公式，计算新序列$x_u[n]$

$$x_u[n] = \begin{cases} x[n/2], n = 0, \pm 2, \pm 4, \dots \\ 0, \qquad otherwise \end{cases}$$

$$x[n] = \{2,13,0,7,3,8,61\}$$

$$\uparrow$$

$$x_u[n = 0] = x[0] = 0$$

$$x_u[n = 1] = x[0.5] = 0$$

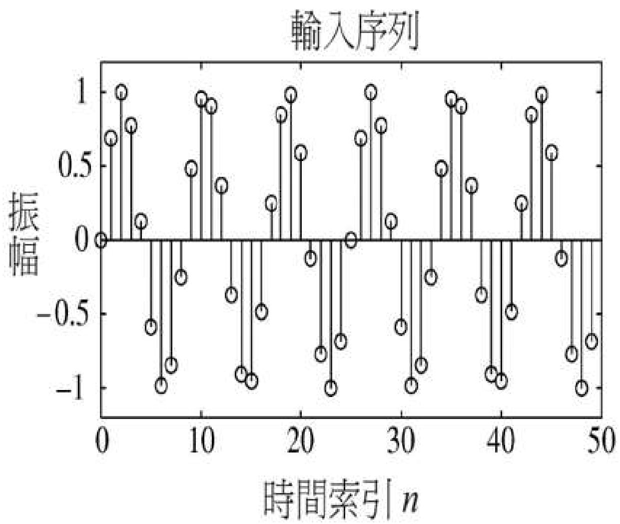
$$x_u[n = 2] = x[1] = 7$$

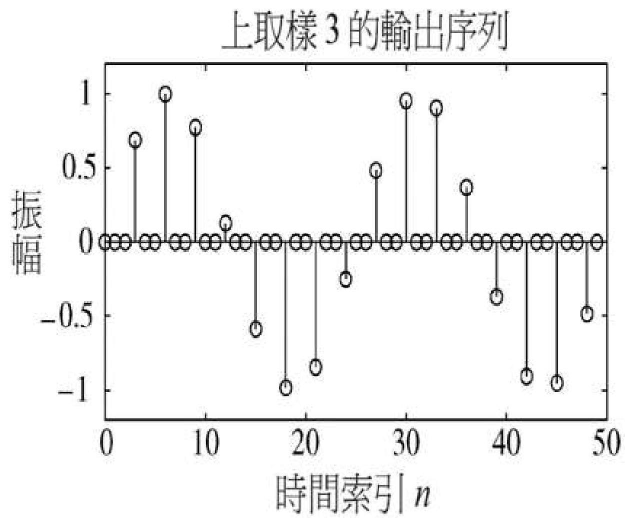
…

基本运算-时域尺度变换（上采样）4 图例

上采样产生的空置的0，
由内插算法补充完整

- 上采样运算的一个例子





2024-9-24

基本运算-时域尺度变换（下采样）5

- 若以整数因子M>1来进行下采样，计算公式见下图。
  - 乘法是下采样，越除得到的数列越短。“下”采样的序列变短。
  - 下采样不会产生新的项。
  - 在上述计算中，仍然左右不变的是n项，可以使用列举法进行计算

$$y[n] = x[nM]$$

$$x[n] \longrightarrow \boxed{\downarrow M} \longrightarrow y[n]$$

2024-9-24

53

基本运算-时域尺度变换（下采样）6 习题

- 按照公式，计算新序列$x_u[n]$

$$x_u[n] = x[2n]$$

$$x[n] = \{2,13,0,7,3,8,61\}$$

$$\uparrow$$

$$x_u[n=0] = x[0] = 0$$

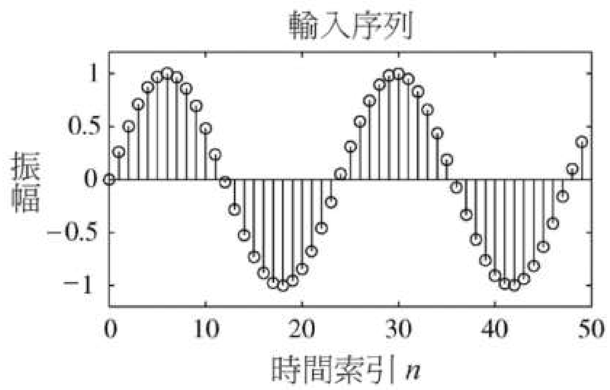
$$x_u[n=1] = x[2] = 3$$

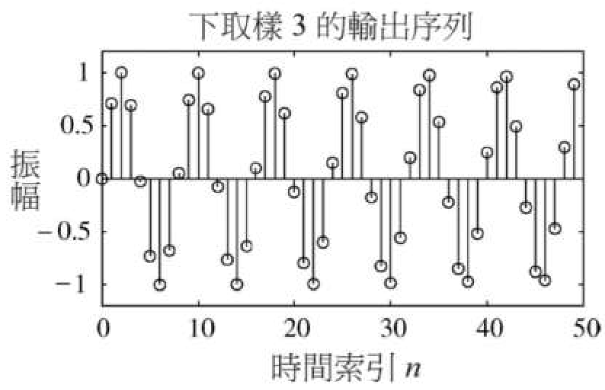
$$x_u[n=2] = x[4] = 8$$

$$\cdots$$

基本运算-时域尺度变换（下采样）7 图例

- 下取样运算的一个例子

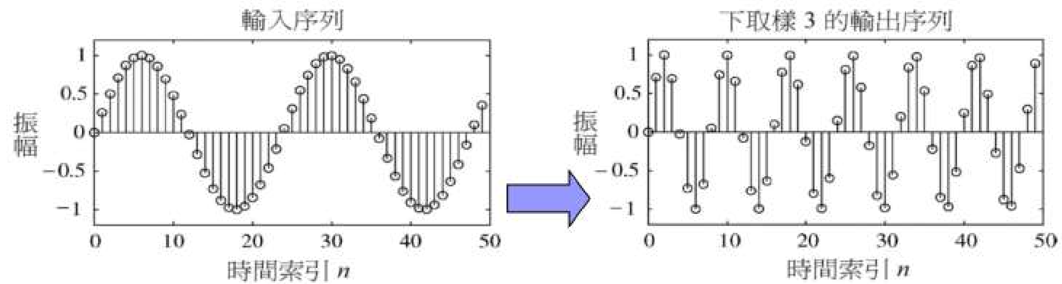




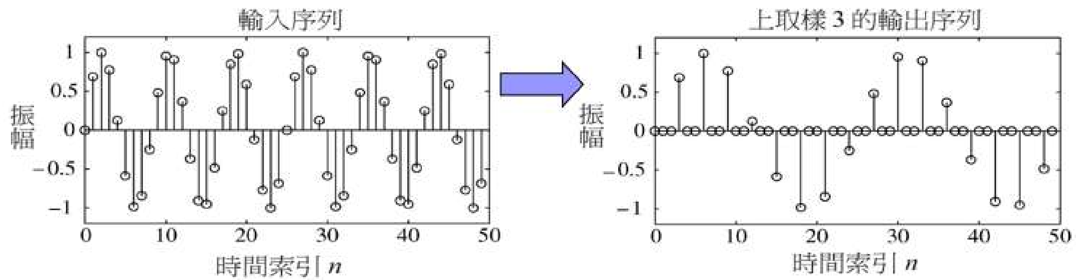
2024-9-24

基本运算-时域尺度变换8 图例

- 下取样



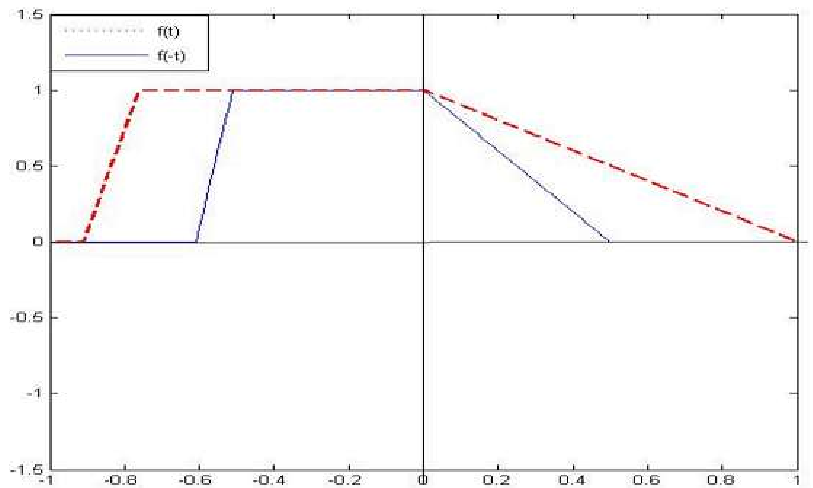
- 上采样



2024-9-24

基本运算-时域尺度变换9 双向拉伸

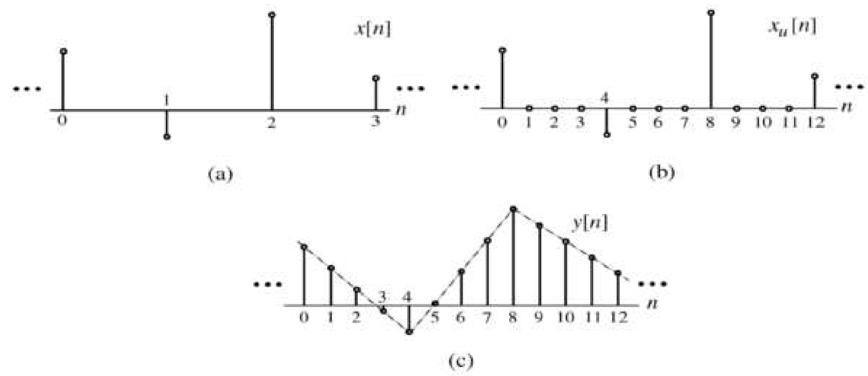
- 拉伸 运算：



2024-9-24

基本运算-内插一览

下面是一个4内插的图例



2024-9-24

基本运算-综合习题1

题1. 对于$x[n]=\{1,2,3,1,5,7,3,5,1\}$, $n=-4:4$, 分别用列举法和平移（箭头）法计算

a) $x[-n+2]$, $x[-n-1]$ 并通过相互验证，总结平移法的计算先后规律（平移与反转）*

b) $x[2n+2]$, $x[1/2n-1]$的结果，并互相验证，并总结平移法的计算先后规律（平移与反转）*

c) $x[-2n+1]$, $x[-1/2n-2]$的结果，并互相验证，并总结平移法的计算先后规律（平移与反转）*

基本运算-综合习题2

至此我们已经学习了所有的基本运算
下面我们看两道题，基本运算的复合应用

$$f(t)=\begin{cases}0 & t<0\\ t & 0<t\le 1\\ 1 & 1<t\le 2\\ 0 & t>2\end{cases}$$

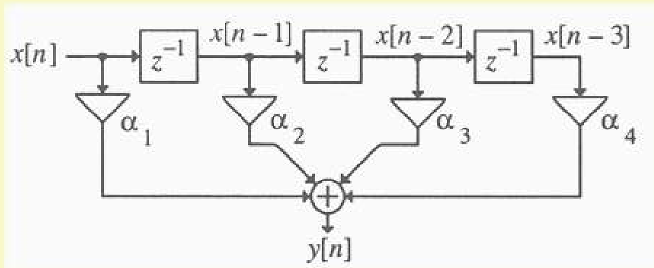
画出它的图形以及

$$3f(-2t-2)$$

的图形

基本运算-综合习题3

- 3.1 给出下面电路图的表达式



- 3.2 给出下面电路图的表达式

a) $z(n) = x(n+1) + x(n-2)$

b) $z(n) = x(2n+1)$

c) $z(n) = x(2n+2) + x(n-2)$ *

2024-9-24

61

时域数字信号处理（5）

相对复杂的基本运算

2024-9-24

62

基本运算-差分 1

- 微分运算：

$$f'(t) = \frac{\mathrm{d}f(t)}{\mathrm{d}t} = \lim_{\Delta t \to 0} \frac{f(t+\Delta t) - f(t)}{\Delta t}$$

- 离散格式: $x[n]$ 代替 $f(t)$，则最小间隔由 $\Delta t \to 0$ 而 $\Delta n = 1$，

$$f(t+\Delta t) \to x[n+\Delta n] = x[n+1]$$

$$f(t) \to x[n]$$

$$\Rightarrow x'[n] = x[n+1] - x[n]$$

- 上述公式成为离散序列的差分

基本运算-差分 2

- 离散差分: $x'[n] = x[n+1] - x[n]$
  - 注1：微分要求左右导数一致，才可导。
  - 差分，例如$x[n] = \{1,2,3,1,2\}$的差分，是否存在可以讨论左右差分？左右差分是否一致？
    - “左右”差分一致的条件就是局部为一条直线，例如2,3,4；所有基本都无法满足左右差分
  - 不一致是否还可以使用差分？
    - 可以，因为这种误差可以忽略。忽略的原因是由于，每个Δn所代表的“真实”时间极小，所以这个累计的误差就可以忽略（见积分—累加部分）
  - 那么应该使用左差分还是右差分？是否有差别
    - 左右差分在单纯计算差分时，影响不大，关键在于“-1”的延迟还是“+1”的超前，这个时间平移有一定影响，所以由电路设计决定。

基本运算-差分 3 二阶微分与差分

- 二阶微分

$$f''(t) = \frac{\mathrm{d}f'(t)}{\mathrm{d}t} = \lim_{\Delta t \to 0} \frac{f'(t+\Delta t) - f'(t)}{\Delta t}$$

- 二阶差分:$x''(n) = x'[n+1] - x'[n]$

$$= (x[n+2] - x[n+1]) - (x[n+1] - x[n])$$

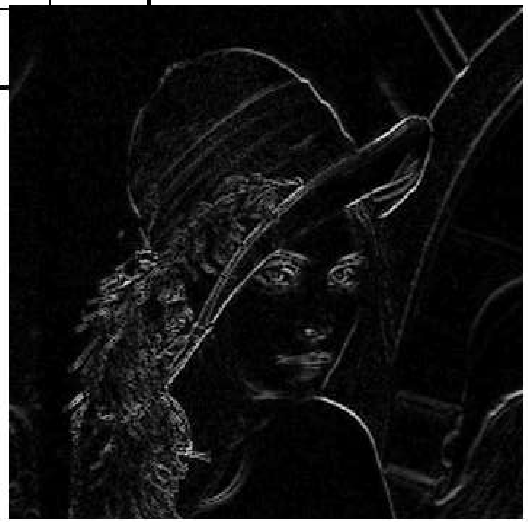
$$= x[n+2] + x[n] - 2x[n+1]$$

- 注1：做一个位移为1的时移变换（以求更简洁的格式）

$$x''[k-1] = x[k+1] + x[k-1] - 2x[k] \quad (k = n+1)$$

基本运算-微分 4 应用

162	162	160	162	
163	162	160	162	…
163	160	159	161	
		…		



2024-9-24

基本运算-微分 5 拓展

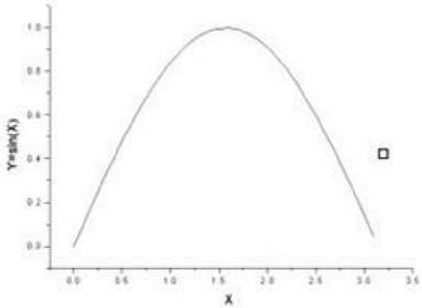
- 思考题：
- 推导离散序列的3、4阶差分，并尝试总结通项公式

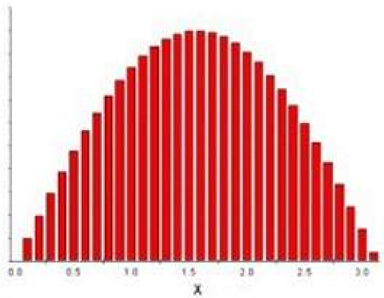
基本运算-累加

- 积分运算：

$$g(t)=\int_a^b f(t)dt=\lim_{\Delta t\to 0}\sum_{k=(a-t_0)/\Delta t}^{(b-t_0)/\Delta t} f(t_0+k\cdot\Delta t)\Delta t$$

- 上述表示为矩形法求积分，与黎曼积分本质相同，黎曼积分是求Δt的$\max(f(t))$，即区域内的最大值求和，当$\Delta t\to 0$时，两个积分一样





2024-9-24

基本运算-累加2

- 离散格式:以$x[n]$ 代替$f(t)$，则最小间隔由$\Delta t \rightarrow 0$变为$\Delta n = 1$,则最终积分变为累加式

$$g(n) = \sum_{a}^{b} x[n]$$

- 上式称为累加式:
  - 注1：累加式的上下限都是整数
  - 注2：累加得到的只是无单位的n上一段空间的求和，在现实中，由于采样率较大（即一秒钟采集的点数n很多）所以往往需要较大范围的积分才有现实意义
  - 注3：最常见的累加器是从$(-\infty, a]$，这个通常称为默认的“累加器”

卷积

- 卷积是整个数字信号处理中最重要的运算。
  - 我们所学的基础处理整合起来称为“系统”
  - 在本课程中我们只学习一种系统“线性时不变系统”
  - 一个线性时不变系统，不管多么复杂，其都可以变成输出信号=输入信息与特定（且固定）信号的卷积
- 即：本课程的任何系统，都可以用卷积来表示和研究。

2024-9-24

70

线性卷积1

- 卷积分为两种：线性卷积和周期（循环）卷积
- 线性卷积的连续性表达式为

$$F(t) = f(t) * g(t) = \int_a^b f(t-m) \cdot g(m) dm$$

- 其离散表示

$$y(n) = x(n) * h(n) = \sum_{k=-\infty}^{\infty} x[n-k] \cdot h[k]$$

  - 注1：上述式子中的$h(n)$为什么使用h在后文会有所解释
  - 注2：请注意上述式子中的所有时间序列符号，$y(n)$，$x(n)$，$h(n)$，累加中的$x[n-k]$，$h[k]$ 累加的上下限分别为$\infty, -\infty$

线性卷积 2 基本运算

- 运算 $y(n) = x(n) * h(n) = \sum_{k=-\infty}^{\infty} x[n-k] \cdot h[k]$

	线性卷积						h=[1,2,3]			x=[3,4,1]												
1	2	3		→			3	2	1			→		3	2	1			→	3	2	1
									3	4	1				3	4	1			3	4	1
							0	0	3	0	0			0	6	4	0			9	8	1
								x = 3							x=6+4=10					x=9+8+1=18		
				→				3	2	1						3	2	1				
							3	4	1			→		3	4	1						
							0	12	2	0				0	0	3	0	0				
								x=12+2=14							x=3							
										X=[3,10,18,14,3]												

2024-9-24

线性卷积3 基本运算

- 运算基本流程： $y(n) = x(n) * h(n) = \sum_{k=-\infty}^{\infty} x[n-k] \cdot h[k]$
  - 将$h(n)$反转，作为滑动窗
  - 对应公式中的 $h(k)$与$x(n-k)$
  - 窗口内处理$h(n)$，并进行对位相乘后求和
  - 对应公式中的$\sum_{k=-\infty}^{\infty}$ ⬚，滑动窗内求和
  - 将滑动窗做$x(n)$最左侧一直移动到最右侧
  - 对每一个点x(n) 进行运算，在公式中没有体现，

2024-9-24

线性卷积4 基本运算

- 运算基本流程： $y(n) = x(n) * h(n) = \sum_{k=-\infty}^{\infty} x[n-k] \cdot h[k]$
- 运算后的序列长度为?

3	2	1		
		3	4	1

		3	2	1
3	4	1		

- 平移了 5次， 通项公式为:
- 卷积后序列长度为 $L = N + M - 1$

2024-9-24

线性卷积5 基本运算

- 上述展示中没有给出的$x(n), h(n)$中分别n的范围，实际上是需要的。
- 假设两个序列都从0开始，则最后$y(n)$的 $n = 0: N + M - 1$
- 若$x(n), h(n)$不是从0开始，则

- 作业：研究$x(n), h(n)$不是由0开始，则$y(n)$的 n的范围
- 研究方法：查资料
- 呈交方式：写出推导

线性卷积6 卷积的说明

- 为什么么$h(n)$在运算时要反过来？
- 引用某网上例子[1]
  - 小泽在第k天被打了，自被打那天起，他每天的疼痛逐天减少，减少的方式为当天$h(0)$，事后第一天$h(1)$，第二天h(2),….
  - 小泽第m天又被打了，还按照上述疼痛衰减方法不变。
  - 假设小泽的疼痛度可以叠加。试问小泽在任意第n天的疼痛度是多少？
  - 这个问题需要反向追溯小泽之前挨过几次打。于是有
  - 第n　　天挨打对于当前第n天的累计为$x(n-0)\cdot h(0)$, $h(0)$为第n天传到当前第n天的影响
  - 第$n-1$天挨打对于当前第n天的累计为$x(n-1)\cdot h(1)$, $h(1)$为第$n-1$天传到第n天的影响
  - 第$n-2$天挨打对于当前第n天的累计为$x(n-2)\cdot h(2)$, $h(2)$为第$n-2$天传到第n天的影响
  - 。。。

2024-9-24　　76

线性卷积7 卷积的说明

- 为什么$h(n)$在运算时要反过来？
- 在计过程中，小泽在某一天的

疼痛值=$\sum$之前的挨打强度 × 往前追溯天数k的影响$h(k)$

3	2	1	
	3	4	1
0	6	4	0

x=6+4=10

- 注1：若那天没挨打，可以令那天信号为0
- 注2：这里的挨打，指代的是脉冲信号（冲击信号）
- 注3：上述的描述，就是卷积对于一个线性时不变系统的描述方法。其中
- 线性指的疼痛可以线性累加
- 时不变指的是系统不管计算那天的疼痛值，往前的追溯系数都是$h(k)$

2024-9-24

77

线性卷积8 深度学习中的卷积

- 在计算机视觉中，最常见的技术是卷积网络，例如CNN。
- 在CNN中，卷积核是不反转的[3]，这是由于深度学习中的卷积只是为了计算局部效果（见后面例子），他只使用了DSP中卷积的 对位乘法和滑动窗两个概念，而没有（也没有必要）使用追溯这个观念

- 相关文献：

[1] 浅谈(线性)卷积公式为什么要翻转:https://www.cnblogs.com/marsggbo/p/6622903.html

[2] 在定义卷积时为什么要对其中一个函数进行翻转？:https://www.zhihu.com/question/20500497

[3] 深入理解卷积(卷积核到底要不要翻卷):https://blog.csdn.net/weixin_37682263/article/details/87914913

线性卷积9 卷积的边缘效果

- 假设我们定义一个卷积核$h(n) = [-1\ 1]$
- 实际效果

1	-1	
153	152	100

x = 1

	1	-1
153	152	100

x = 52

		1	-1
153	152	100	

x = 100

- 这时最后一行出现较大误差，这时所谓的边缘效果
- 边缘效果在DSP中并不太被重视，这是由于
  - DSP通常认为信号是无限长的（实时系统）
  - DSP中通常认为极短时间内的差异是可以忽略的（采样频率的影响）

2024-9-24

雨课堂 Rain Classroom

卷积10

$$*y[m,n]=\begin{bmatrix}0&0&0\\0&1&0\\0&-1&0\end{bmatrix}$$

2024-9-24

卷积11

$$*y[m,n]=\begin{bmatrix}0 & -1 & 0\\ -1 & 4 & -1\\ 0 & -1 & 0\end{bmatrix}$$

VS.

2024-9-24

卷积12

$$*y[m,n]=\frac{1}{16}\begin{bmatrix}1&1&1\\1&8&1\\1&1&1\end{bmatrix}$$

$$*y[m,n]=\frac{1}{92}\begin{bmatrix}1&1&4&1&1\\1&4&8&4&1\\4&8&16&8&4\\1&4&8&4&1\\1&1&4&1&1\end{bmatrix}$$

2024-9-24

线性卷积8-2 深度学习中的卷积2

- 在计算机视觉中，最常见的技术是卷积网络，例如CNN。
- CNN中，3个 重要工作点：
- 目标：将一张图变成一组数
- 1， 卷积核自行学习，使用很多很多卷积核
- 2， 卷积核不反转
- 3， 不停进行上采样。

快速卷积

- 我们在手动计算卷积时，可以采取下列两个手动方法实现
- 左侧卷积和右侧快速卷积

$$\text{已知} x_1(n)=\left\{\underset{\substack{\uparrow\\ n=0}}{4},\ 3,\ 2,\ 1\right\},\ x_2(n)=\left\{\underset{\substack{\uparrow\\ n=0}}{3},\ 2,\ 1,\right\},$$

$$\text{求：} y(n)=x_1(n)*x_2(n)$$

- 使用右侧对位相乘求和法求卷积，步骤：
  - 两序列右对齐（不做反转）
  - 逐个样值对应相乘但不进位
  - 同列乘积值相加（注意n=0的点）

$$
\begin{array}{lllcccccc}
 & x_1(n) & : & & & \underset{\substack{\uparrow \\ n=0}}{4} & 3 & 2 & 1 \\
\times & x_2(n) & : & & & & \underset{\substack{\uparrow \\ n=0}}{3} & 2 & 1 \\
\hline
 & & & & & 4 & 3 & 2 & 1 \\
 & & & & 8 & 6 & 4 & 2 & \\
+ & & & 12 & 9 & 6 & 3 & & \\
\hline
 & y(n) & : & \underset{\substack{\uparrow \\ n=0}}{12} & 17 & 16 & 10 & 4 & 1
\end{array}
$$

$$
\therefore y(n)=\left\{\underset{\substack{\uparrow \\ n=0}}{12} \quad 17 \quad 16 \quad 10 \quad 4 \quad 1\right\}
$$

2024-9-24

左侧的快速卷积

$$
\begin{array}{rrrrrrrr}
 & x_1(n) & : & \underset{\substack{\uparrow \\ n=0}}{4} & 3 & 2 & 1 & \\
\times & x_2(n) & : & \underset{\substack{\uparrow \\ n=0}}{3} & 2 & 1 & & \\
\hline
 & & & 12 & 9 & 6 & 3 & \\
 & & & & 8 & 6 & 4 & 2 \\
+ & & & & & 4 & 3 & 2 & 1 \\
\hline
 & y(n) & : & \underset{\substack{\uparrow \\ n=0}}{12} & 17 & 16 & 10 & 4 & 1
\end{array}
$$

$$
\therefore y(n)=\left\{\underset{\substack{\uparrow \\ n=0}}{12} \quad 17 \quad 16 \quad 10 \quad 4 \quad 1\right\}
$$

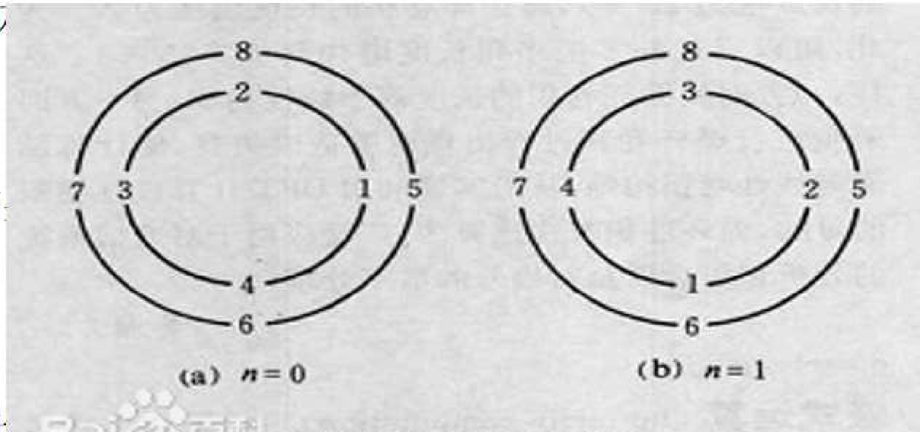
2024-9-24

快速卷积说明

- 所谓的快速卷积只是给人类手动算卷积准备的
  - 节省的时间是不需要处理边缘效果的那几次滑动
- 当窗口很大时，节省的时间微乎其微
  - 手动编写感受
- 所以计算机并不使用快速卷积
  - 计算机使用转频域-> 算乘法->转回时域的方法
- 这个需要循环卷积

循环卷积1

- 循环卷积的处理分为两部分：循
- 循环扩展：
  - 循环卷积表述方法为k-循环卷积：需
    后反复“拷贝”，构造循环序列
  - 表述为 $x(n) \cdot R_k\,(n)$
  - 做循环的意义在于引入信号的周期性
    快速卷积”
  - 请注意，循环卷积的k，是可能大于$x(n)$的长度N的
- 扩展后再做常规卷积，整体公式表示为



两个序列循环卷积的计算

$$y(n) = x(n) \circledast h(n) = \left(\sum_{k=-\infty}^{\infty} x[n-k] \cdot h[k]\right) \cdot R_k\,(n)$$

2024-9-24

88

百度文 × | 百度文 × | baike.b × | 循环卷 × | 循环卷 × | 循环卷 × | 循环卷 × | [循环卷 × | 循环卷 × | 概率论 × | 百度文 × | +

baike.baidu.com/item/循环卷积?fromModule=lemma_search-box

Gmail

Bai du 百科 | 循环卷积 | 进入词条 | 播报 | 编辑 | 讨论 | 收藏 | 赞 | wanatorun

定义

播报 编辑

计算两个长度均为N的序列$x_1(n)$和$x_2(n)$的循环卷积，一个简易的办法是先把$x_1(n)$的数据，设$x_1(n)=(1, 2, 3, 4)$，N=4，按逆时针方向均匀分布在一个圆周上。如图1中(a)。的内圆所示，而把$x_2(n)$，设$x_2(n)=\{5, 6, 7, 8\}$，按顺时针的方向均匀分布在另一个同心圆上，然后求两圆上相应序列的乘积，并把V项乘积叠加起来作为n=0时刻的卷积值y(0)，即

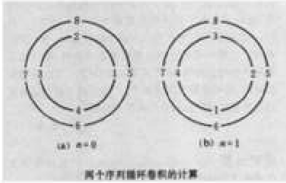


图1

y(0)=1×5+4×6+3×7+2×8=66

若求n=1时刻的y(l)值，可将外圆的：$x_2(n)$固定，把内圆上的序列$x_1(n)$顺时针旋转一个单位时间（或将$x_1(n)$固定，把外圆上的序列$x_2(n)$逆时针旋转），然后把对应项的乘积叠加起来，即为所求。如图1中的(b)图所示，即

y(1)=2×5+1×6+4×7+3×8=68

这样依次将内圆序列进行循环移位一周，便可以求得：y(2)=66，y(3)=60。所以循环卷积又称圆周卷积，卷积结果y(n)长度仍等于N，其定义式如图2。方括号内的运算代表两个周期序列的周期卷积。它与线性卷积不同之处是卷积过程只限在m=0到N-1的一个周期内。乘以$R_N(n)$表示卷积结果只取长度为N的主值序列。

$$y(n)=x_1(n)\circledast x_2(n)=\left[\sum_{m=0}^{N-1}\tilde{x}_1(m)\cdot\tilde{x}_2(n-m)\right]R_N(n)$$

图2

由于离散傅里叶变换(DFT)的实质是周期序列变换到频域的描述。可以证明：两个有限长序列在时域的循环卷积，其DFT等于在频域两个序列相应的DFT的乘积。 [1]

式中$X_1(k)$和$X_2(k)$分别是$x_1(n)$和$x_2(n)$的N点DFT。它表明DFT具有循环卷积性质（CCP），也是区别于其他变换的重要特性。正是这种性质，用计算机通过计算DFT达到计算循环卷积和线性卷积的目的，提高了运算效率。按长度为N的两个序列，其线性卷积的长度应为2N-1，而循环卷积的长度仍然为N。为此，可以通过补零把序列长度增加到L≥2N-1。这样，一方面使循环卷积的长度等于线性卷积，另一方面避免进行循环卷积过程出现混叠造成失真，使计算结果与线性卷积相等，从而实现利用DFT计算线性卷积的目的。循环卷积有快速算法，广泛应用于对通信系统的分析和综合以及对信号的数字处理

"科普中国"是为我国科普信息化建设塑造的全...

什么是权威编辑

资源提供

中国通信学会

中国通信学会是全国通信...

提供资源类型：内容

什么是资源合作

词条统计

浏览次数：110331次

编辑次数：8次历史版本

最近更新：阿妩云 （2021-08-03）

突出贡献榜

通信学会cic

循环卷积3

- 循环卷积与线性卷积的关系：
- 当$x(n)$与$h[k]$做线性卷积，其长度为$N+K-1$
- 做L点循环卷积，长度为 L
  - 当$L \geq N+K-1$时，两者输出序列一样
  - 当$L < N+K-1$时，两者输出不同
  - 请自行数学验证、计算验证、编程验证。
- 请根据算法知识，自行估算卷积的运算复杂度
- 为了降低卷积的算法复杂度，我们通常使用
  - 傅立叶变换->乘法->逆傅里叶变换的方法
- 来计算 有限长序列的卷积，所以需要循环卷积

卷积的性质

1. 交换律

$$x(n) * h(n) = h(n) * x(n)$$

2. 结合律

$$x(n) * h_1(n) * h_2(n) = x(n) * [h_1(n) * h_2(n)]$$

3. 分配律

$$x(n) * [h_1(n) + h_2(n)] = x(n) * h_1(n) + x(n) * h_2(n)$$

4. 不存在微分、积分性质。

请自行数学验证、计算验证、编程验证。

讯号的相关性-1

- 在有些应用中，有时必须将另外一个或是多个讯号与参考讯号进行比较，以决定讯号间的相似性。或是利用此讯号间的相似性来判读出其余更多的信息 。
- 自身与自身比较
  - 在雷达、声纳、激光和定位中的应用中，从目标物反射而被接收到的讯号属于延迟后的发射讯号，而藉由量测延迟时间可以判读目标物出现的位置---*自相关*。
  - 在这些应用中，我们是把两个同样（但是其中一个经过时移）的信号进行比较
- 自身与其他信号比较
  - 声控中的语音识别、雷达中的信号类型判定
- 在上述应用中，我们是把两个不同的信号进行比较

讯号的相关性-2 定义一

- 两个能量讯号 x[n] 及 y[n] 彼此间的相似度可利用如下定义的互相关序列 (cross-correlation sequence) 来量度。

$$r_{xy}[l] = \sum_{n=-\infty}^{\infty} x[n] \cdot y[n-l], l = 0, \pm 1, \pm 2, \ldots$$

- 相关性包含两个操作，平移（时移）和相乘求和；输出的序列请注意，参数是l
- 相关性计算和卷积相似，但是没有反转，请对比卷积公式

$$y[n] = \sum_{k=-\infty}^{\infty} x[n-k] \cdot h[k]$$

讯号的相关性-3

$$r_{xy}[l]=\sum_{n=-\infty}^{\infty} x[n]\cdot y[n-l]\,, l=0,\pm 1,\pm 2,\ldots$$

- 相关性中的平移（时移）的意义：
  - 参数l 表示的是将序列$y[n]$做相应的延迟（lag）后，对位相乘做加法
  - 按照滑动窗的解释，$r_{xy}[l]$中的l 表示的是，对于滑动窗每个滑动距离的结果上的记录
- 参数 l 称为延迟 (lag)，它代表讯号间的时间位移。
  - 若 l 是一个正数（例如1），则表示时间序列$y[n]\to y[n-1]$,按照前文的知识点（时移），相当于将整个序列零点不动情况下，整体“右移”，进行延迟；反之亦然。
- 公式中下标 xy 的顺序表示以 x[n] 作为参考讯号保持不动，而将序列 y[n] 相对 x[n] 产生位移。

讯号的相关性-4 相关系数

- 还存在另外一种相关性定义，称为相关系数（Correlation Coefficient）
- 相关系数处理的是有限长序列，需要将常规序列进行加上zero-mean（零均值）与normalization（正规化）

$$Corr = \frac{\sum_i \left[(A_i - \overline{A})(B_i - \overline{B})\right]}{\sqrt{\sum_i (A_i - \overline{A})^2 \sum_i (B_i - \overline{B})^2}}$$

$$= \frac{\text{Covariance}(A, B)}{\sigma_A \cdot \sigma_B}$$

讯号的相关性-5 正规化

- 对一个向量 $A=\{A_i\}$，正规化

$$K=\sqrt{\sum A_i^{2}},\ A_i'=\frac{A_i}{K}$$

- 有A_i'满足

$$\sum A_i'^2=\sum\frac{A_i^{2}}{K^2}=\frac{1}{K^2}\sum A_i^{2}=\frac{1}{\sqrt{\sum A_i^{2}}^{2}}\sum A_i^{2}=1$$

- A_i'平方和为 1，称A_i'为A的一种正规化结果
  - 正规化方法有很多，上面是最常见的一种

2024-9-24

讯号的相关性5：正规化（2）

- 将向量至于一个单位圆内（即一个单位下），以便进行后期的比较等运算
- 例子：当我们观察信号 A={3,4,5,5,5} 与信号B={-5,-8,-10,-9,-10}时，
- 由于放大器(K=-2)和噪声(Noise={1,0,0,1,0})的共同影响，不一定可以准确的观测到其关系，进行正规化以后
- A'={0.3， 0.4， 0.5，0.5， 0.5} 与
- B'={-0.26,-0.42,-0.52,-0.47,-0.52}的相似度就足够高了

讯号的相关性-6 零均值

- 对一个向量 $A = \{A_i\}$，零均值
- 则有

$$\overline{A} = \frac{1}{N}\sum A_i，\ A_i' = A_i - \overline{A}$$

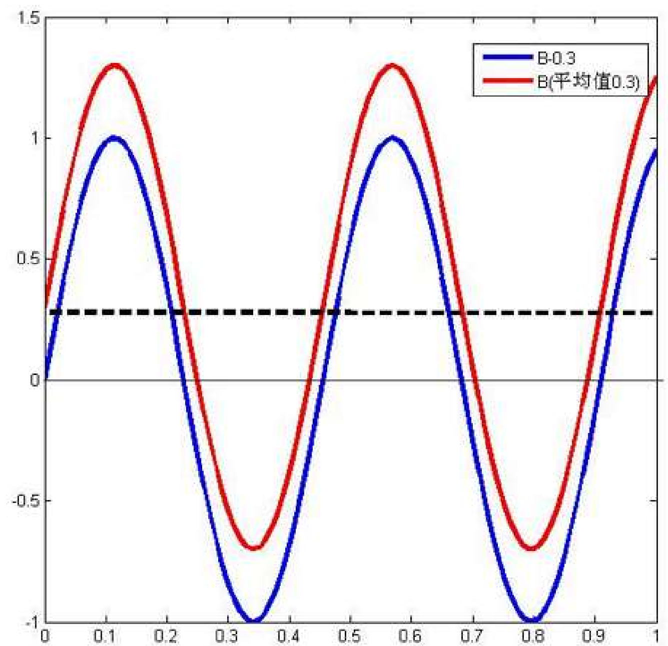
- A_i' 满足

$$\sum A_i' = \sum (A_i - \overline{A}) = \sum A_i - \sum \overline{A} = \sum A_i - N \cdot \frac{1}{N}\sum A_i = 0$$

- 称A_i' 为A的一种零均值

讯号的相关性4：零均值（2）

- 将向量至于一个按其内部元素和，“均匀”至于x轴上下两边



2024-9-24

讯号的相关性4：零均值（3）

- 例子，当我们观察信号 A={3,-4,2,5,-7} 与信号B={4,-2,5,7,-5}时，
- 由于叠加信号（D={2,2,2,2,2}和叠加噪声(Noise={-1,0,1,0,0})）的共同影响，不一定可以准确的观测到其关系，进行零均值化以后

- A'={3.2, -3.8,2.2,5.2,6.8} 与
- B'={2.2,-3.8,3.2,5.2,-6.8}的相似度就足够高了

讯号的相关性-5 相关系数(2)

- 再来看相关系数的定义，令A_i 零均值变化以后为A_i'

$$Corr = \sum_i \frac{(A_i' \cdot B_i')}{\sqrt{\sum_i (A_i')^2 \sum_i (B_i')^2}}$$

- 再令$\widehat{A'}$ 为正规化的A_i'

$$Corr = \sum_i \frac{(A_i')}{\sqrt{\sum_i (A_i')^2}} \cdot \frac{(B_i')}{\sqrt{\sum_i (B_i')^2}} = \sum_i \widehat{A_i'} \cdot \widehat{B_i'}$$

讯号的相关性-5 相关系数(3)

- 重要一点，这样得出的相似度，其绝对值 $0 \le |corr| \le 1$
- 其中两个向量的相似度与符号无关，由绝对值决定。
- 若$|corr| = 1$则两个向量相同（或反向），相似度最高
- 若$|corr| = 0$则两个信号完全无关（向量正交），相似度最低
  - 概率与统计中的传统定义

2024-9-24

讯号的相关性-6 相关性vs相关系数（1）

- 相关性一般只出现在DSP，而相关系数广泛用于统计、数据挖掘、数据分析、机器学习等
- 不做零均值及正规化的缺点：
  - 因为不做正规化，没有基准，很有可能两个向量的相似值处于 $(-\infty, +\infty)$，且对放大不免疫
  - 因为没有零均值，对任何叠加信号都不免疫。
- 反之，做过系列操作后，而相似度比较与信号 数值的范围无关

讯号的相关性-6 相关性vs相关系数（2）

- 传统算法脱胎于信号与系统，模拟电路，测试的是电流强度，该类信号大部分是能量可积的，积分和小于无穷，所以在部分研究时会先做能量正规化（等效于我们的正规化）
- 其次关于零均值，对于电路来说，负电流通常来说是不存在（被禁止的），所以缺乏物理意义。
- 做零均值和正规化，需要遍历整个序列，浪费太多的运算时间和运算资源
  - 无限长序列无法做零均值和正规化（只能在一个窗口内做）

讯号的相关性7 相关性深入研究

- 如果我们希望将 y[n] 当作参考讯号，并将 x[n] 相对于 y[n] 产生位移，则互相关序列可改写成下式

$$r_{yx}[\ell]=\sum_{n=-\infty}^{\infty}y[n]x[n-\ell]$$
$$=\sum_{m=-\infty}^{\infty}y[m+\ell]x[m]=r_{xy}[-\ell]$$

- 因此，将 $r_{xy}[l]$作时间反向可以得到$r_{yx}[-l]$
  - 请在此处再次比较相关性计算和卷积，卷积不存在交换位置结果反折的变换

2024-9-24

讯号的相关性-8 自相关

- 在互相关序列 $r_{xy}[l]$的公式中设定 y[n] = x[n]，则 x[n] 的自相关序列 (auto-correlation sequence) 可得到为:

$$r_{xx}[\ell] = \sum_{n=-\infty}^{\infty} x[n]x[n-\ell]$$

$$r_{xx}[0] = \sum_{n=-\infty}^{\infty} x^2[n] = \mathcal{E}_x$$

- 在零延迟时 (l = 0)自相关序列的取样值将有最大值。
  - 请尝试简单证明

相关序列范例(1)

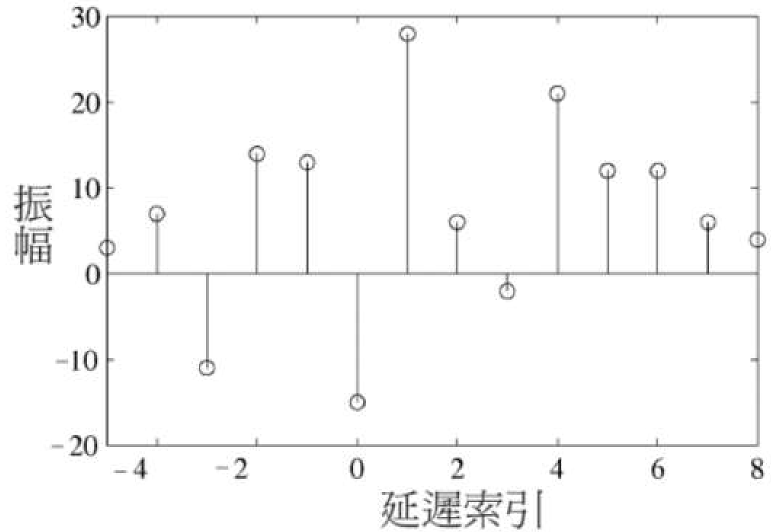
- 范例 — 考虑下列两个有限长度序列

$$x[n]=[1\ \ 3\ \ -2\ \ 1\ \ 2\ \ -1\ \ 4\ \ 4\ \ 2]$$
$$y[n]=[2\ \ -1\ \ 4\ \ 1\ \ -2\ \ 3]$$

- 1）我们对于超出的位置，补零计算
- 2）对于结果$r_{xy}[l]$是一个 l 的函数，对应不同的时移，有不同的相似度值

相关序列范例(2)

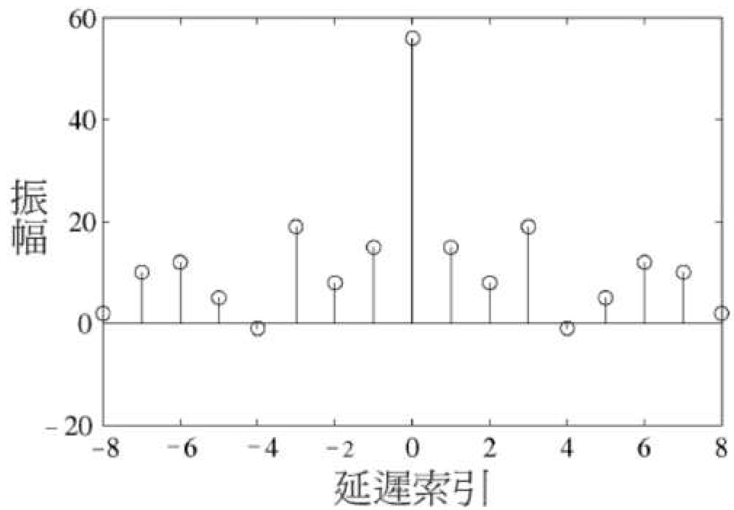
- 两个有限长度序列的互相关$r_{xy}[l]$ 的结果绘图如下



2024-9-24

相关序列范例(3)

- 自相关 $r_{xx}[l]$ 的结果绘图如下。
- 注意：在零延迟时， $r_{xx}[0]$ 有最大值



2024-9-24