

# 信号与系统软件说明书

高金焘 唐灵 NNLXD 张乐 张羽桐 周子航

(上海交通大学信号与系统课程大作业)

## 目录

<b>1 引言</b>	<b>2</b>
1.1 项目选择背景	2
1.2 参与项目成员	2
<b>2 软件设计说明</b>	<b>2</b>
2.1 软件底层设计	2
2.1.1 基本信号类的实现	2
2.1.2 连续信号的定义与典型连续信号	3
2.1.3 离散信号的定义与典型离散信号	3
2.2 面向信号的操作	4
2.2.1 卷积运算	4
2.2.2 傅里叶级数的求取	4
2.2.3 根据傅里叶系数求 N 阶部分和	5
2.2.4 傅里叶变换的实现	5
2.3 CT AM modulator 的实现	6
2.4 DT processing of CT signals 的实现	6
2.5 软件界面设计	7
2.5.1 环境配置	7
2.5.2 简要步骤	7

2.5.3	各功能界面部分 . . . . .	7
2.5.4	设计优点 . . . . .	9
2.6	软件前后端 . . . . .	9
2.6.1	预备步骤 . . . . .	9
2.6.2	设计思路 . . . . .	9
2.6.3	代码实现 . . . . .	10
2.6.4	调试打包 . . . . .	10

# 1 引言

## 1.1 项目选择背景

信号与系统这门课程主要涉及对信号的处理、频域时域变换等问题。由于信号表达式比较抽象，信号的各种运算处理比较复杂。故本软件对信号以及其典型处理结果进行函数和图像展示，结果兼具直观性和严谨性。

## 1.2 参与项目成员

高金焄 唐灵 NNLXD 张乐 张羽桐 周子航

# 2 软件设计说明

## 2.1 软件底层设计

### 2.1.1 基本信号类的实现

首先对“信号”做出最基本的定义，一个信号有以下若干项属性：

1. 信号的名字：由用户自己进行定义。
2. 是否为数组信号：即该信号能否用显式的解析式表达，若能，则应提供信号的解析式；若不能，则需要用数对组的形式表达该信号。
3. 信号类型：计算机难以处理连续值，对于连续值，我们使用取样间隔较短的离散形式进行模拟；而离散信号则将其取样间隔固定为 1。

4. 是否为连续时间的冲激信号：由于连续时间单位冲激信号定义的特殊性，难以找到适宜的显式解析表达，故作为一种特殊情况进行表示。

5. 信号范围：考虑到计算机难以处理无限值，对于所有的信号，均规定了上下限。

6. 实部与虚部、模与相角：复数信号的四个关键参数，在信号基类中定义了获取之并进行分别作图的方法。

### 2.1.2 连续信号的定义与典型连续信号

对于连续信号 (ct\_signal) 类，继承信号 (signal) 类，将其信号类型锁定为“CT”。几类典型连续信号的实现：1. 单位冲激信号 (ct\_unit\_impulse)：如前所述，单位冲激信号无显式表达。在该类中基于单位冲激将所有冲激信号进行表示，有两个参数：冲激大小 k 和冲激零点 t0。t 与 x 的范围根据 k 与 t0 两个参数决定。对于具体函数值进行近似计算。

2. 冲激串 (ct\_impulse\_train)：冲激串默认以 0 作为冲击零点，有两个参量 c 和 k，表示自零点开始往两侧每隔 c 处有一个高度为 k 的冲激串。

3. 单位阶跃信号 (ct\_unit\_step)：单位阶跃信号在  $t \geq 0$  时取 1，否则取 0，本信号没有参数，仅表示 u(t) 信号。

4. (复) 指数信号 (ct\_exp)：复指数信号有四个参量：幅值 k，直流分量 b，复指数的实部 r 和虚部 i，构成了一个形如  $x(t) = k * e^{(r+i*j)*t} + b$  的复指数函数。

5. 矩形信号 (ct\_rect)：定义的矩形窗仅含一个参数 T1，表示矩形窗宽度的一半，定义得出的信号是  $x(t) = u(t+T1) - u(t-T1)$ 。

6. 采样信号 (ct\_sample)：定义采样信号  $x(t) = Sa(t) = \text{sinc}(t/\pi)$  为基本的采样信号。

7. 周期方波信号 (ct\_rect\_period)：在矩形信号的基础上添加表示周期的参数 T 即得周期方波信号，表示每隔周期 T 产生一个宽度为 T1 的矩形窗。

### 2.1.3 离散信号的定义与典型离散信号

离散信号类同样继承信号类，将信号类型锁定为“DT”。几种典型离散信号的实现：1. 单位冲激信号 (dt\_unit\_impulse)：

2. 单位阶跃信号 (dt\_unit\_step)：与连续时间采用相同的定义方式，定义基本的  $x[n] = u[n]$ ，即对于  $n \geq 0$  时令 x 为 1，否则为 0。

3. (复) 指数信号 (dt\_exp)：由四个参量——幅值 k，直流分量 b，复指数的实部 r 和虚部 i——确定唯一的复指数信号  $x[n] = k * e^{(r+i*j)n} + b$ 。

4. 采样信号 (dt\_sample): 定义基本采样信号  $x[n] = Sa[n] = \text{sinc}(n/\pi)$ 。
5. 矩形信号 (dt\_rect): 定义基本矩形信号  $x[n] = u[n+N1] - u[n-N1]$  表示基本矩形信号
6. 周期方波 (dt\_rect\_period): 由两个参量 N, N1 决定, 表示每隔 N 个点产生一个宽为 2N1 的矩形脉冲。

## 2.2 面向信号的操作

### 2.2.1 卷积运算

1. 连续时间的卷积运算:

对于卷积积分式

$$y(t) = \int_{-\infty}^{\infty} x_1(\tau) \cdot x_2(t - \tau) d\tau$$

将其中的无穷在软件中用相对较大值 100 替代。

普通信号的卷积计算, 通过面积积分估算方法进行计算, 针对某一时间得到值。

冲激函数的卷积运算: 由于冲激函数没有显式表达, 需要单独进行计算, 利用冲激函数的性质可知冲激函数与其他信号的卷积即为原信号的时移和比例变换。

2. 离散时间的卷积运算:

对于卷积和:

$$y[n] = \sum_{k=-\infty}^{\infty} x_1[k] \cdot x_2[n - k]$$

同样将无穷用 100 替代。之后将卷积和公式用累加函数进行实现。离散信号卷积函数, 任意输入两个离散信号可通过累加得到卷积结果。

除连续时间冲激函数与其他信号的卷积输出另一个信号的比例变换和时移之外, 连续时间与离散时间输出的都是数组形式的信号。

### 2.2.2 傅里叶级数的求取

1. 连续信号的傅里叶级数:

根据连续信号的傅里叶级数分析公式:

$$a_k = \frac{1}{T} \int_T x(t) \cdot e^{jk\omega_0 t} dt = \frac{1}{T} \int_T x(t) \cdot e^{jk(2\pi/T)t} dt$$

用离散的累加代替连续域中的积分，可以求得信号的第  $k$  傅里叶级数系数  $a_k$ 。

## 2. 离散信号的傅里叶级数：

根据离散信号傅里叶系数的分析公式

$$a_k = \frac{1}{N} \sum_{n=\langle N \rangle} x[n] \cdot e^{-jk\omega_0 n} = \frac{1}{N} \sum_{n=\langle N \rangle} x[n] \cdot e^{-jk(2\pi/N)n}$$

设计累加函数，根据定义可以求得第  $k$  个傅里叶级数  $a_k$ 。

最终在文件中用函数实现统一包装。

### 2.2.3 根据傅里叶系数求 $N$ 阶部分和

已知某周期信号的  $N$  阶傅里叶系数  $a_k$ ，由傅里叶系数综合公式

$$x(t) = \sum_{k=-\infty}^{\infty} a_k \cdot e^{jk\omega_0 t} = \sum_{k=-\infty}^{\infty} a_k \cdot e^{jk(2\pi/T)t}$$

根据综合公式，在给定的参数（周期  $T$  和求和阶数  $N$ ）的条件下，对已知的  $a_k$  数列进行循环求和，对于每一个  $k$  求出  $x_k(t)$ ，再将所有  $k$  值下的  $x_k(t)$  叠加得到求和结果  $x(t)$ 。

### 2.2.4 傅里叶变换的实现

#### 1. 连续信号的傅里叶变换

根据连续信号傅里叶变换式

$$X(j\omega) = \int_{-\infty}^{\infty} x(t) \cdot e^{-j\omega t} dt$$

首先先用函数实现对于特定的  $\omega$  值求得对应的  $X(j\omega)$  值，再生成一个  $\omega$  序列并对序列每一项分别调用该函数求得整个  $X(j\omega)$ ，求得的  $X(j\omega)$  是一个数组信号。

#### 2. 离散信号的傅里叶变换

根据离散信号傅里叶变换式

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n] \cdot e^{-j\omega n}$$

与连续时间的算法类似，同样先对特定的  $\omega$  求得  $X(e^{j\omega})$ ，再对  $\omega$  序列的每一项调用该函数求得整个数组形式的信号。

除了一般性的分析公式求法之外，本软件还实现了对于几种离散和连续典型信号的傅里叶变换的直接求取。

## 2.3 CT AM modulator 的实现

CT AM modulator 实现原理为：利用傅里叶变换的乘积性质，

$$r(t) = s(t)p(t) \longleftrightarrow R(j\omega) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} S(j\theta)P(j(\omega - \theta))d\theta$$

在时域上将基带信号  $s(t)$  和载波信号  $p(t) = \cos(\omega_0 t)$  乘积，获得调制信号

$$r(t) = s(t) \cos(\omega_0 t)$$

于是，在频域上，调制后的信号的傅里叶变换为

$$R(j\omega) = \frac{1}{2} [S(j(\omega - \omega_0)) + S(j(\omega + \omega_0))]$$

，即完成基带信号在频域上的两个等距相反平移的线性叠加。

1. 调制后的时域波形表示首先实现一个对特定时间  $t$  输入返回调制信号  $r(t)$  的函数，之后，在波形显示函数里调用该函数，输入一组有限长的时间数组得到一组  $r(t)$  的值并利用其绘出波形。
2. 调制后的频谱波形表示利用上述调制后信号频域的最终表达式，和时域同理，实现对于一个特定的频率  $\omega$  返回调制信号的傅里叶变换的函数，再调用此函数，利用一组数据映射实现波形绘制。

## 2.4 DT processing of CT signals 的实现

对于一个 DT processing 过程，由于涉及到在频域与 impulse train 卷积（与无数个频域平移后的原信号频谱叠加），故以初始信号为带限（bandlimited）信号为总前提才有意义。

### 1. 采样功能的实现

对于采样信号  $r(t)$ ，使其在时域上与一个 impulse train  $x(t) = \sum_{k=-\infty}^{\infty} \delta(t - kT)$  乘积。由于冲激串的采样特性， $r(t)$  在时域上被等间距采样。同时，由于乘积特性， $r(t)$  在时域上与另一个 impulse train  $P(j\omega) = \frac{2\pi}{T} \sum_{k=-\infty}^{+\infty} \delta(\omega - \frac{2\pi k}{T})$  卷积，在采样率大于奈奎斯特率的情况下，可以得到频域的周期波形（原信号的平移）。由此便得出采样过程下两域的波形图。

在代码实现上，对于时域采样，首先实现冲激串的绘制，再在每一个采样点形成一个相应幅值的冲激信号；对于频域处理，则是将其变换成周期信号（乘以相应的幅值）。

### 2. 滤波功能的实现

时域采样后的频谱图是一个周期信号，首先我们将其离散化（即将时域采样后结果进行 scaling，使冲激串间间隔为 1），利用 scaling 性质，频谱周期拉伸或压缩为  $2\pi$ 。接着，在低通滤波器下，仅有和原始信号频谱幅值成  $A/T$  倍（ $A$  为滤波器幅值）的频谱被保留下来，在傅里叶逆变换下，获得时域波形，和初始信号波形类似，滤波完成。

在代码实现上，对于频域，乘以相应的滤波器增益  $A$  后，保留原始信号频谱，对于时域，将处理后的频谱进行傅里叶逆变换。

### 3. 线性插值功能的实现

对于一个接受采样并离散化的信号，同样可以使用线性插值的方式恢复。首先是通过 scaling，恢复原始信号采样后的时域波形。其次，利用线性插值 (linear interpolation) 方法，在频域乘积一阶保持器  $H_1(j\omega) = \frac{1}{T} \left[ \frac{\sin(\omega T/2)}{\omega/2} \right]^2$ ，即时域卷积三角波，获得一阶保持的近似原始信号。

## 2.5 软件界面设计

### 2.5.1 环境配置

- 应用软件：Qt Designer 软件
- 环境配置：python3 下安装 PyQt5 库，安装 Qt 工具，配置 Qt 工具系统环境变量。

### 2.5.2 简要步骤

1. 打开应用界面，创建窗口，设置窗口参数；
2. 选择合适的控件，拖拽到合适位置，调整大小，添加使用提示信息；
3. 为使界面随拖拽同比例缩放，使用布局，添加布局设置，调整界面；
4. 设置接口。

### 2.5.3 各功能界面部分

#### (I) 菜单栏

一个界面，包含 6 种功能选择键、公式定义规范提示界面选择键、退出键。

#### (II) 功能实现界面

##### (1) 任意信号图像展示

界面 1：任意信号图像展示，选择离散信号、连续信号和散点画图；

界面 2：连续和连续信号参数设置，输入信号名称、t 和 x 范围、信号函数，展示和退出按键；

界面 3：散点信号参数设置，输入信号名称，t 和 x 散点；

界面 4：图像波形展示界面。

##### (2) 典型信号波形展示

界面 1：典型信号波形展示，选择连续信号和离散信号；

界面 2：典型连续信号选择，选择 6 种连续信号；

界面 3：典型离散信号选择，选择 6 种离散信号；

界面 4：图像波形展示；

界面 5-10：典型连续信号根据选择弹出，进行参数设置和确认展示；

界面 11-16：典型离散信号根据选择弹出，进行参数设置和确认展示。

### (3) 典型信号卷积（和）

界面 1：典型信号卷积（和），选择连续信号和离散信号；

界面 2：典型连续信号选择，下拉框两个，每个从 6 种连续信号选择两种；

界面 3：典型离散信号选择，下拉框两个，每个从 6 种离散信号选择两种；

界面 4：图像波形展示；

界面 5-25：连续信号卷积根据选择弹出，进行参数设置和确认；

界面 26-46：离散信号卷积根据选择弹出，进行参数设置和确认。

### (4) 典型信号傅里叶变换

界面 1：典型信号傅里叶变换，5 个按键选择 5 种信号之一进行变换操作；

界面 2-6：根据选择的信号设置参数，点击变换键输出傅里叶变换公式，点击展示键，进行图像波形展示；

界面 7：图像波形展示。

### (5) 周期信号傅里叶级数

界面 1：周期信号傅里叶级数，选择离散信号、连续信号、常用信号；

界面 2：连续周期信号傅里叶级数，输入周期连续信号函数，点击变换，输出框会给出傅里叶级数函数结果，点击展示，会显示时域图像和频谱图像；

界面 3：离散周期信号傅里叶级数，输入周期离散信号函数，点击变换，输出框会给出傅里叶级数函数结果，点击展示，会显示时域图像和频谱图像；

界面 4：常用信号选择，5 个按键选择 5 种信号；

界面 5：图像波形展示，并且如果需要展示  $N$  阶傅里叶级数和（从  $-N$  到  $N$ ）就给出参数  $N$ ，点击展示键要求绘出图像；

界面 6：展示  $N$  阶傅里叶级数和图像；

界面 7-11：5 种被选择信号参数设置；

### (6) 连续信号离散处理

界面 1：CT AM 调幅器，输入基带信号，设置载波信号参数，展示键展示调制信号图像；

界面 2：展示调制信号时域和频域图像，采样键进行下一步骤；



界面 3: 采样, 设置采样频率, 展示键展示采样信号;

界面 4: 展示采样信号时域和频域图像, 滤波键进行下一步骤;

界面 5: 滤波, 设置截止频率和增益, 展示键展示滤波信号;

界面 6: 展示滤波信号时域和频域图像, 线性插值键进行下一步骤;

界面 7: 线性插值, 设置周期, 展示键展示插值后信号;

界面 8: 展示插值信号, 退出键结束应用。

### **(III) 报错界面**

一个界面, 提示错误。

### **(IV) 公式定义规范格式**

一个界面, 显示常用函数、常量类型规范输入以及示例。

## **2.5.4 设计优点**

- (1) 软件功能以及功能需要的各项下级操作清晰明确;
- (2) 功能具体实现界面不重不漏, 保证用户个性化设置和自定义内容完整全面;
- (3) 界面设计在具体功能实现代码衔接部分简单可行;
- (4) 界面设计美观, 布局简洁合理, 提示信息清楚易懂;
- (5) 软件各功能使用操作简单方便。

## **2.6 软件前后端**

### **2.6.1 预备步骤**

首先编写函数用于将设计好的界面文件 (UI 文件) 转为 python 文件, 得到基于 pyqt5 语法的界面类。

### **2.6.2 设计思路**

1. 使用 python 语言编写, 主要基于 pyqt5 库实现界面的连接跳转, 输入读取, 以及按键与实现函数的对接;
2. 整体上使用 sigtem-main 文件作为顶层文件, 用于展示主界面菜单并对接其他六个界面。使用 sigtem-f1x..... 到 sigtem-f6x, 分别实现六个功能界面。
3. 使用时, 用户根据需求选择任意功能, 按照界面输入提示进行操作即可

### 2.6.3 代码实现

1. 界面内功能实现时使用了基本的 pyqt5 指令，展示如下：
2. 功能的实现主要基于以实现的包内的各种信号类及函数，包括构造信号并对信号进行傅里叶变换等操作。
3. 值得一提的是展示图像时的代码实现我们使用 graphicsview 进行展示，为了解决对接时的冲突问题，我们采用将图片暂存在本地，再重新读取然后展示的方法，并在展示完成之后将图片删除。

### 2.6.4 调试打包

1. 代码编写完成后进行仔细的调试，确保结果的正确性，并对界面进行适当调整 and 美化；
  2. 最终的打包我们使用 pyinstaller，打开 cmd 界面并跳转到目标文件夹，输入 `pyinstaller sigtem-main.py` 指令，进行一轮包装。随后输入 `pyinstaller -F -w sigtemmain.py`
    - (1) 其中-F 将所有内容全部打包成一个 exe 可执行文件，而不会有其它的一些奇奇怪怪的小依赖文件。
    - (2)-w 使运行生成的 exe 文件时，不会弹出命令行窗口，而是直接弹出我们做的 GUI。
- 打包完成最终得到没有任何依赖的可执行文件。