

上篇——声像信号时域处理

本篇中案例设计的主旨是通过对声音、图形、图像信号的处理，使读者对数字信号的时域处理方法、步骤和效果建立直观概念，理解和掌握数字信号的产生和时域（空域）常用处理，尤其是进一步深化对于线性卷积概念和实现的理解。

本篇中案例的情境任务设计涵盖了数字信号常用的时域处理，包括信号叠加、调制、滤波、相关等，对应数字信号处理理论中的时域离散序列的加法、乘法、延时、翻转、样点抽取（降速率采样）、线性卷积等基本运算。此外，案例中专门涉及了时域离散系统的差分方程描述及该描述下的信号处理，为后续滤波器的描述与应用做好铺垫。

在 MATLAB 使用方面，本篇中案例的内容设计涵盖了 MATLAB 的常用基本操作、矩阵乘法和加法、函数调用、子程序的编写、运算时间或效率的评估、结果的图形显示和声音呈现，以及数据的保存等内容，旨在夯实 MATLAB 的使用基础。

建议至少完成声音图像 DIY、信号周期化及时域采样和滤波可以卷出来三个案例。其中涉及的矩阵操作和子函数编写对后续内容有非常重要的作用。

案例一 —— 牛刀初试

【案例设置目的】

通过在 MATLAB 环境下产生序列及对序列的基本操作，掌握序列的矩阵表示、图形表示及声音呈现方法，在加深对序列基本运算理解的同时，学会和熟悉在 MATLAB 平台上进行矩阵创建、矩阵运算、画图、图形标注、视听声效的方法，以便为后续的仿真实验奠定坚实基础。

【相关基础理论】

设有一频率为 f 、初相为 0、幅度为 A 的正弦信号 $x(t)$ ，数学上可以表示为

$$x(t) = A \sin(2\pi ft) = A \sin(\Omega t) \quad (1.1)$$

若以采样速率 F_s 或采样间隔 $T=1/F_s$ 对 $x(t)$ 进行采样，可以得到时域离散序列 $x(nT_s)$ 或 $x(n)$ ，即

$$x(t)\Big|_{t=nT} = x(nT) = A \sin(2\pi fnT) = A \sin(\Omega Tn) = A \sin(\omega n) = x(n) \quad (1.2)$$

这里 $\omega = \Omega \cdot T$ 。

该正弦序列 $x(n)$ 也可以看成是复指数序列 $e^{j\omega n}$ 的虚部构成的序列。

设信号序列为 $x(n)$ ，则信号序列 $x(n)$ 的翻转序列可表示为

$$y(n) = x(-n)$$

【情境任务及步骤】

本案例共设置了三个情境，分别是音频试听与处理、基本信号（序列）生成和程序阅读，每个情境下有各自的情境任务。

一、MATLAB 之初体验——音频文件的视听与处理

1. 声音文件读取

读取一个 wav 声音文件（这里指定系统启动的声音文件，对于安装于 C 盘的 Windows XP 系统而言，Windows XP 启动.wav 的存储路径为 C:\WINDOWS\Media\），并把数据存放在矩阵 y 中，采样速率存放在 F_s 中，采样深度或位数存放在 $nbits$ 中，并查看 F_s 和 $nbits$ 的值及 y 的维数。

MATLAB 提供的函数为 `wavread` 和 `uigetfile`，可以通过阅读相关的 Help 文件，学习函数的调用语法。

2. 耳听声音

要求在命令窗口下分别执行 `sound(y, F_s)`，`sound(y, F_s/2)`，`sound(y, 2F_s)`，以试听三种声效，体验正常速度录音正常速度播放、快录慢放、慢录快放的效果。

MATLAB 提供了声音播放函数 `sound` 和 `wavplay`，可以通过阅读相关的 Help 文件，学习函数的调用语法。

3. 眼看声音

在两个图形窗口中画出声音波形图，观察波形特点。创建图形窗口 Figure 1，并在其中画出 $y \sim t$ 的图；创建图形窗口 Figure 2，调用 subplot 函数自上而下分成两个子窗口，第一个子窗口中画出 $y(:,1) \sim t$ ，第二个画出 $y(:,2) \sim t$ ，要得出如图 1.1 所示的效果。横轴的单位为秒，起点为 0（提示：横轴上的 t 由采样速率和样点数决定，样点数可以用 MATLAB 提供的 size 函数确定）。

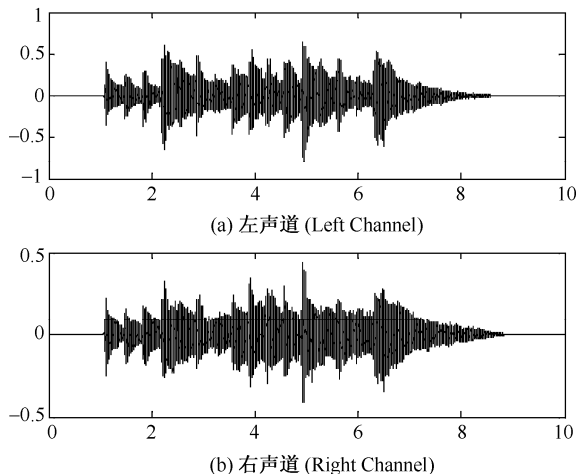


图 1.1 Windows XP 启动.wav 左右声道波形图

MATLAB 提供的画连续波形的函数为 plot，创建新图形窗口的函数为 figure，可以通过阅读相关的 Help 文件，学习函数的调用语法。

4. 特效制作

将一个声道的信号进行翻转（这里指定左声道，即 $y(:,1)$ ），画图表示翻转前后的结果（横轴为 t ），收听翻转后声音效果，体会倒序播放的特效。

MATLAB 支持函数为 fliplr 和 flipud，可以通过阅读相关的 Help 文件，学习函数的调用语法。

5. 眼睛和耳朵里的噪声

产生均值为 0 功率为 -30dBW 的白色高斯噪声序列 nx ，序列的长度与 $y(:,1)$ 相同。创建图形窗口 Figure 3，在其中画出 $nx \sim t$ ，以观看噪声波形，用 sound 函数收听噪声音效。

MATLAB 提供函数为 wgn 和 size，可以通过阅读相关的 Help 文件，学习函数的调用语法。

6. 声音加料

用产生的噪声 nx 对 $y(:,1)$ 中的数据进行污染，污染后的声音用 noisyy 表示，即 $\text{noisyy} = y(:,1) + nx$ 。在新的窗口中观看波形，收听噪声污染前后的音效。

7. 成果保存

用 MATLAB 提供的函数 save 将 y 和 F_s 保存到文件 mydata.mat 中（一定要记清当前的工作路径）。之后，在命令窗口中依次执行 clear all 和 clc 命令；接着分别键入 y 和 F_s （此时忽略一切提示）；再依次执行 load mydata.mat，whos 和 sound(y, F_s)，注意观察每条指令的执行结果。

可以通过阅读相关的 Help 文件，学习相关函数的调用语法。

二、基本序列生成与显示

本部分内容要求全部在命令窗口中完成。继续下面的内容之前，用 `clear all` 释放所有变量，用 `close all` 关闭所有图形窗口，用 `clc` 清理命令窗口。

1. 信号生成之符号函数

首先逐行执行如下代码，并注意观察最后的图形结果。

```
syms y Omega t;
Omega=0.5*pi;
y=sin(Omega*t);
figure(1)
subplot(211)
ezplot(t,y);
axis([0 6 -1 1]);
```

有兴趣的话，可以通过阅读相关的 Help 文件，学习相关函数的含义及调用语法。

2. 正弦序列的多样式生成

在式 (1.2) 中，取 $A=1$, $f=0.25\text{Hz}$, $F_s=10\text{Hz}$, $T=1/F_s$, $t=0:T:6$ ，由模拟角频率与数字角频率之间的关系可知

$$\omega = \Omega \times T = 2\pi \times 0.25/10 = 0.05\pi$$

(1) 按照采样过程生成正弦序列：

$$xnT_s = \sin(2\pi ft) \quad (1.3)$$

(2) 直接用数字角频率 ω 生成正弦序列：

$$xn_1 = \sin(\omega n), \quad n = t/T \quad (1.4)$$

(3) 对复指数序列求虚部生成正弦序列：

$$xn_2 = \text{imag}(\exp(j\omega n)) \quad (1.5)$$

(4) 图示序列。

在 MATLAB 中生成式 (1.3) ~ 式 (1.5) 的序列，之后逐条执行如下代码，以便在 Figure 1 中的第二个图形子窗口中显示上述不同方式得到的正弦序列，重点关注 MATLAB 函数各画图函数、标注函数的使用方法和效果。

```
subplot(212)
stem(t,xnTs);
stem(n,xn1,'filled');
hold on
stem(n,xn2,'r','filled');
plot(n,xn1,'k-.')
grid
```

```
axis([n(1) n(end) -1.2 1.2])
axis([-n(end) n(end) -2 2])
title('Discrete time sequence');
ylabel('Amplitude');
xlabel('\itn')
```

三、程序阅读

在命令窗口中逐条执行如下代码，之后借助 MATLAB 的 Help 逐个解释每个函数的功能和用法。

```
I = imread('eight.tif');
imshow(I)
J = imnoise(I, 'salt & pepper', 0.02);
figure, imshow(J)
```

【思考题】

- (1) 在 MATLAB 平台上，余弦序列如何生成？
- (2) 在 MATLAB 平台上，复指数序列如何生成？
- (3) 描述科学问题的数学公式在使用 MATLAB 实现时，在符号表达、运算符表达、变量索引上有哪些区别？

【总结报告要求】

(1) 情境任务总结报告中相关基础理论部分可以不写，书写情境任务时可适当进行归纳和总结，但至少列出【情境任务及步骤】相关内容的各级标题。

(2) 程序清单除在报告中出现外，还必须以 m 文件形式单独提交。程序清单要求至少按程序块进行注释。本案例要求提交“MATLAB 之初体验——音频文件的视听与处理”的程序清单。为尽可能地减少重复性工作，建议在开始本情境任务前先通过 MATLAB Help 学习函数 diary 的使用和利用命令历史生成脚本文件的方法。

(3) “MATLAB 之初体验——音频文件的视听与处理”的执行效果图要标注图题和横纵坐标，并将效果图附于相关内容之后。

(4) 总结本次情境任务所使用的与画图相关的函数及其使用语法、与文件打开相关的函数及其使用语法、与声音播放相关的函数及其使用语法、与数据保存和装载相关的函数及其使用语法。另外，翻译函数 dlmwrite 的含义及使用语法，并附于报告中。

(5) 简要回答【思考题】中的问题。

(6) 报告中还可包括完成本案例的个人心得，对该案例设置的建议等。

【参考程序】

```
%%%%%%%%%%
%%% Codes for mission 1-7, circumstance 1
clear all
close all
clc
%%%%%%%%%% Choose a Wav-file from a Dialog box
[filename, pathname] = uigetfile('*.wav', 'Pick a Wav-file');
```

```

if isequal(filename,0)
    disp('User selected Cancel')
else
    disp(['User selected', fullfile(pathname, filename)])
end
%%%% Synthesize the full string with path pathname and file name filename
s=[pathname filename];
%%%% Open the sound file directed by s
[y,Fs,nbits]=wavread(s);
[ry,cy]=size(y);

pause
sound(y,Fs)
pause
sound(y,Fs/2)
pause
sound(y,2*Fs)

pause
figure
subplot(211)
plot((0:ry-1)/Fs,y(:,1),'linewidth',2);
title('Left Channel','fontsize',14);
xlabel('Time(s)','fontsize',14);
subplot(212)
plot((0:ry-1)/Fs,y(:,2),'linewidth',2);
title('Right Channel','fontsize',14);
xlabel('Time(s)','fontsize',14);

pause
reversedy1=flipud(y(:,1));
pause
sound(reversedy1,Fs)

P=-30;
nx=wgn(size(y(:,1)),P);
pause
sound(nx,Fs);
pause

figure
plot((0:ry-1)/Fs,nx,'linewidth',2);

noisy=y(:,1)+nx;
pause
sound(noisy,Fs);
pause

```

```
figure
plot((0:ry-1)/Fs,noisy,'linewidth',2);

save mydata.mat y Fs

%%%%%%%%%%%%%%
%%%Codes for missions 1-2, circumference 2
clear all
close all
clc
syms y Omega t;
Omega=0.5*pi;
y=sin(Omega*t);
figure(1)
subplot(211)
ezplot(t,y);
axis([0 6 -1 1]);
subplot(212)
stem(t,xnTs);
f=0.25;
Fs=10;
T=1/Fs;
t=0:T:6;
xnTs=sin(2*pi*f*t);
n=t/T;
xn1=sin(0.05*pi*n);
xn2=imag(exp(j*0.05*pi*n));
subplot(212)
stem(t,xnTs);
stem(n,xn1,'filled');
hold on
stem(n,xn2,'r','filled');
plot(n,xn1,'k-.')
axis([n(1) n(end) -1.2 1.2])
axis([-n(end) n(end) -2 2])
axis([n(1) n(end) -1.2 1.2])
title('Discrete time sequenc');
ylabel('Amplitude');
xlabel('\itn')
```

案例二 —— 声音图像 DIY

【案例设置目的】

通过在 MATLAB 环境下实现单声道音频变立体声、图像处理，掌握序列的乘法、尺度变换（降速率采样）的实现，在加深对序列基本运算理解的同时，进一步熟悉 MATLAB 软件，以便为后续的仿真实验奠定坚实基础。

【相关基础理论】

1. 立体声构造

立体声能给听众以方位感和深度感，大大提高了听觉效果和声音品质。为了实现立体声效果，录制时通常需要多个摆放在不同位置的麦克风，播放时也需要两只或两只以上的扬声器。当只利用一个麦克风进行声音录制时，只能得到单声道音频，即便使用两只或两只以上的扬声器播放，也只能得到“平面化”的声音，而不会形成空间声像。利用人的空间听觉特性，如耳间声级差（Interaural Level Difference, ILD）、耳间时间差（Interaural Time Difference）和耳间相关性对空间声像定位，可以按照预期的听觉效果通过对多个声道信号进行相应处理，在听众听觉上形成空间感。对多声道信号进行的处理包括：扬声器排列法、分频法、移相法、延迟法等。

本案例拟通过将单声道音频转换为双声道音频，并通过周期性地、交替地对两个声道上的声音进行衰减和增强的方式，在感官上造成声源远离听者的一侧而运动到另一侧的感觉，从而形成声音方位感变化的立体声效果。情境任务要求实现听者能感觉到声源在围绕自己进行圆周转动的效果。

设有两通道信号序列 $x(n)=[x_l(n), x_r(n)]$ ，时变增益系数 $G(n)=[G_l(n), G_r(n)]$ ，信号序列 $x(n)$ 放大后的结果为

$$y(n) = G(n) \cdot x(n) \quad (2.1)$$

2. 行列抽取的图像压缩

为减少图像、视频对存储空间的消耗量，可以对它们进行降空间分辨率、降色彩分辨率和去统计冗余等处理，从而达到图像压缩的目的。前两种方法有时会造成失真，即有损压缩。结合数字信号处理的内容，这里讨论的方法是降低空间分辨率的压缩方法。

设有信号序列 $x(n)$ ，以 Scale 为比例因子对 $x(n)$ 进行尺度变换，可表示为

$$y(n) = x(\text{Scale} \times n) \quad (2.2)$$

当 $\text{Scale} > 1$ 时， $y(n)$ 是从序列 $x(n)$ 中每 Scale 个样点中取 1 个样点得到的，相当于采样频率降为 F_s/Scale ，实现了对信号 $x(n)$ 的降速率采样；当 $\text{Scale} < 1$ 时，实现对信号 $x(n)$ 升速率采样，相当于采样频率升为 $\text{Scale} \times F_s$ 。假设 Scale 可写成有理分式的形式，即 $\text{Scale} = M/D$ ， M 和 D 为不小于 1 的自然数，如 $\text{Scale} = 1.25$ 可写成 $5/4$ ，则不论是升速率采样还是降速率采样，都称为变速率采样问题，或称为再采样问题，都可以表示为采样速率先升 M 倍再降 D 倍的情形。

图 2.1 所示为变速率采样流程图。图中 $x_{\text{old}}(n)$ 表示以原采样率 F_s 采样的结果， $x_{\text{new}}(n)$ 表示以新采样率采样的结果； $\downarrow D$ 表示每 D 个样点保留 1 个样点； $\uparrow M$ 表示序列 $x_{\text{old}}(n)$ 相邻样点间

插入 $M-1$ 个零； LPF_D 表示为保证以每秒采 F_s/D 个样点时仍满足 Nyquist 采样率定律而对 $x_{\text{old}}(n)$ 进行抗混叠滤波时所用的低通滤波器； LPF_M 表示为滤除镜像频谱所用的低通滤波器。

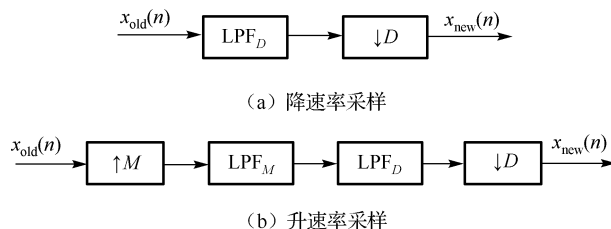


图 2.1 变速率采样流程图

严格讲，变速率采样需要按图 2.1 所示的过程处理，以避免二次采样失真或解决 Scale 为非整数的情况。假设获得信号序列 $x(n)$ 时所用采样速率 F_s 足够高，而降速率的比例因子 D 又不大，因此 F_s/D 仍然大于信号最高频率的 2 倍，则图 2.1 (a) 中降速率过程中可省去滤波的过程。下面的情境任务均是基于此假设而进行的。

一维信号的降速率采样，在 MATLAB 环境下可直接调用 decimate 函数；升速率采样调用函数 interp。这两个函数的使用语法详细说明和实现算法要写在情境任务总结报告中。

【情境任务及步骤】

本案例设置了立体声构造、粗暴的图像压缩和图像 PS 三个情境，各情境有不同的情境任务。

一、立体声构造

1. 单声道音频信号生成

参考案例一——“牛刀初试”中正弦序列产生方法生成一单频的正弦信号，并保存在列矩阵 \mathbf{x}_l 中。

$$x(t) = A \sin(2\pi ft) \tag{2.3}$$

在式 (2.3) 中，幅度 A 初设为 1，频率 f 设置为 800Hz，采样频率 F_s 设定为 f 的 10 倍，持续时间在 5s 以上。

2. 收听声效

听生成的信号 \mathbf{x}_l 的效果，并根据听的效果调整信号的频率和幅度，以舒适为原则（涉及个人听力灵敏度，建议先将音量旋钮调小，频率在 50~3000Hz 范围内选择）。

MATLAB 声音播放函数包括 sound 和 wavplay，可借助 Help 文件学习函数的调用语法。

3. 单声道音频变立体声音频

1) 单声变双声

构造一个 2 列的矩阵 \mathbf{x} ，其每一列内容均与 \mathbf{x}_l 相同。

要求使用 MATLAB 中提供的 repmat 函数，可借助 Help 文件学习函数的调用语法。

2) 构造空间差

(1) 生成左声道增益系数 G_l 和右声道增益系数 G_r ，使其在一个圆周内的幅度变化有类似如图 2.2 (a) 和 (b) 所示的效果。建议左右声道增益变化频率设置为 0.1Hz。

(2) 分别用 G_l 和 G_r 对 \mathbf{x} 的两个列向量进行放大，放大后结果记作 \mathbf{y} 。

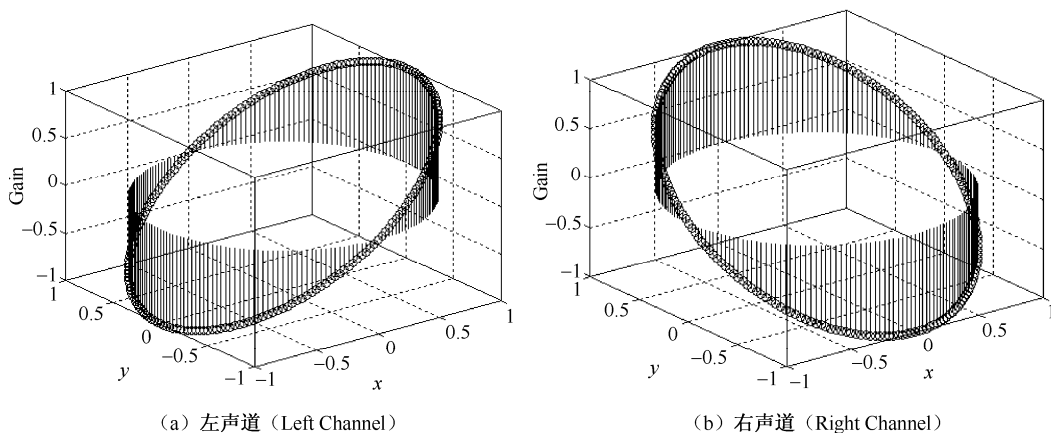


图 2.2 左右声道在一个圆周内增益效果图

4. 视听声音

听声音 y 的效果, 进一步调整增益参数 G_L 和 G_R (正余弦的频率) 和幅度至声音的方位感清晰, 并观察左右声道波形。

二、粗暴的图像压缩

图像压缩有多种方式, JPEG 压缩最为常见, 该方法不改变图像的空间分辨率和色彩分辨率, 一般压缩比也不是很高, 能达到 10 倍的数量级左右。要想得到更大压缩比的压缩效果, 降低分辨率是其中一种途径, 下面将用实验说明基于行列抽取降低图像分辨率的方法。

1. 图像读取

读取一幅 tif 格式的灰度图像, 图像数据存入矩阵 I_1 中并进行显示。MATLAB 提供的 tif 格式图像多存于 (MATLAB $\times\times$) \toolbox\images\imddemos 中 (这里指定 moon.tif 文件)。

MATLAB 支持函数 imread 和 imshow, 可以通过 Help 文件学习相关函数的调用语法。为了便于文件的选取, 可以调用 uigetfile 函数。

2. 行列抽取不断降低图像分辨率

1) 图像高度压缩

对 I_1 进行行数据抽取, 每 M ($M=8$) 行抽取一行的结果记为 $I1RM$, 在新窗口中显示结果。

2) 图像宽度压缩

对 I_1 进行列数据抽取, 每 N ($N=8$) 列抽取一列的结果记为 $I1CN$, 在新窗口中显示结果。

3) 幅面整体压缩

对 $I1RM$ 进行列数据抽取, 每 N 列取一列的结果记为 $I1MN$, 在新窗口中显示结果。

改变 M 和 N 的取值重新对图像进行压缩, 并显示结果。

3. 借用 MATLAB 函数实现幅面同时压缩

调用 MATLAB 函数对 I_1 代表的图像按如下比例进行放缩, 并分别在两个窗口中显示。

(1) scale 取 0.5, 结果记为 $I1S05$ 。

(2) scale 取 2.5, 结果记为 $I1S25$ 。

MATLAB 支持函数 `imresize`，可以通过 Help 文件学习相关函数的调用语法。

本情境任务开始前，建议首先在命令窗口中执行如下两条语句（句末不带分号），体会抽取含义。

```
x=1:10  
x(1:2:end)
```

三、图像 PS

1. 图像读取

读取一幅 `tif` 格式的灰度图像，数据存入 I_2 中，计算 I_2 的维数，并在新开辟的图形窗口中进行图像显示。

2. 马赛克

将 I_2 中的一个 64×64 块的灰度设置为固定值，如 80，修改后结果记为 I_{22} ，并在新开辟的图形窗口中显示图形 I_{22} 。

3. 图像拼接

调用函数 `fliplr` 生成 I_2 的镜像翻转图像 I_{2lr} ，并在新开辟的图形窗口中显示 $I_3=[I_2 \ I_{2lr}]$ 。

4. 图像反色

(1) 调用函数 `size` 计算 I_2 的维数。

(2) 确定矩阵 I_2 中的最大值，记为 `maxI2`。

(3) 借助函数 `ones` 生成维数与 I_2 相同、每个元素值均取 `maxI2` 的矩阵 x ，并在新开辟的图形窗口中显示 x 。

(4) 计算 $I_n I_2 = x - I_2$ ，并在新开辟的图形窗口中显示 $I_n I_2$ 。

提示：差值计算需要统一数据类型。可以通过 `whos` 命令查看已使用过的所有变量的维数、数据类型等参数。MATLAB 提供的数据转换函数包括 `double`、`uint8` 等。

【思考题】

(1) 增益 G_I 、 G_r 和序列 x 的采样率 F_s 有何关系？

(2) 在 MATLAB 环境下实现两个序列的乘法、加法等基本运算时，对序列有什么要求？

(3) 与图像有关的分辨率是如何定义的？

【总结报告要求】

(1) 情境任务总结报告中简要描述每个情境任务的原理，书写情境任务时可适当进行归纳和总结，但至少要列出【情境任务及步骤】相关内容的各级标题。

(2) 情境任务的程序清单除在报告中出现外，还必须以独立的 `m` 文件形式单独提交。程序清单要求至少按程序块进行注释。

(3) 执行效果图都至少要标注图题，并附于相关内容之后。

(4) 总结完成本案例所使用的新的函数及其使用语法。

(5) 简要回答【思考题】中的问题。

(6) 报告中还可包括完成本案例的个人心得，对该案例设置的建议等。

【参考程序】

```

clear all
close all
clc
%%%%%%%%%%
%%%Codes for missions 1-4, circumstance 1
f=800;
Fs=10*f;
T=1/Fs;
t=0:T:5;
x=sin(2*pi*f*t);
sound(x,Fs)
%%%%%%%%%%
Fs=22050;
Ts=1/Fs;
f=500;
f0=.1;
t=0:Ts:20;
x=sin(2*pi*f*t);
xx=[x' x'];
Ax1=sin(2*pi*f0*t);
Ax2=cos(2*pi*f0*t);
xx2=xx;
xx2(:,1)=xx2(:,1).*Ax1';
xx2(:,2)=xx2(:,2).*Ax2';
sound(xx2,Fs)
figure
subplot(211)
plot(t,xx2(:,1));
subplot(212)
plot(t,xx2(:,2));

%%%%%%%%%%
%%% Codes for missions 1-3, circumstance 2
close all
clear all
clc
I=imread('cameraman.tif');
[rI,cI]=size(I)
figure
imshow(I);
disp('Input Scale Factor M,please')
% M=4;
M=input('M=')
pause
cMI=I(:,1:M:end);
pause
rMI=I(1:M:end,:);

```

```
pause
crMI=cMI(1:M:end,:);
pause
figure
imshow(cMI)
pause
figure
imshow(rMI)
pause
figure
imshow(crMI);

%%%%%%%%%%%%%%
%%% Codes for missions 1-4, circumstance 3
clear all
close all
clc
% [fn pn]=uigetfile('* .tif');
% I2=imread([pn fn]);
I2=imread('F:\Program Files\MATLAB\R2013a\toolbox\images\imdemos\moon.tif');

figure
imshow(I2)

[rI2,cI2]=size(I2)
I22=I2;
I22(50:50+64-1,60:60+64-1)=80*ones(64,64);
figure
imshow(I22)

I2lr=flipI2r(I2);
I3=[I2 I2lr];
figure
imshow(I3)

maxI2=max(max(I2));
x=maxI2*uint8(ones(size(I2)));
InI2=x-I2;
figure
imshow(InI2)
```

案例三 —— 信号周期化及时域采样

【案例设置目的】

通过从有限长序列产生周期序列的过程，进一步理解有限长序列和周期序列的关系，并掌握利用平移相加法生成周期序列的方法；理解信号时域采样的原理；掌握脉冲串的产生方法。

【相关基础理论】

1. 周期信号产生

有限长序列和周期序列之间存在密切关系。有限长序列可以看成是由周期序列截取得到的，而周期序列可以看成是由有限长序列拓展而来的。

设 $x(n)$ 为一有限长序列，其中 $0 \leq n \leq M-1$ ，则周期为 N 的周期序列 $\tilde{x}(n)$ 可以这样得到：

$$\tilde{x}(n) = \sum_{r=-\infty}^{\infty} x(n+rN) \quad (3.1)$$

若 $N \geq M$ ，有限长序列 $x(n)$ 可以从周期序列 $\tilde{x}(n)$ 这样截取出来：

$$x(n) = \tilde{x}(n) \cdot R_N(n) = \tilde{x}(n) \cdot R_M(n) \quad (3.2)$$

若 $N < M$ ，则有限长序列 $x(n)$ 就不能再从周期序列 $\tilde{x}(n)$ 恢复。

当 $N \geq M$ 时，还有另一种从有限长序列得到周期序列的方法：

$$\tilde{x}(n) = x((n))_N \quad (3.3)$$

式 (3.3) 中 $((\cdot))_N$ 表示对 N 求模或余数。

式 (3.3) 所定义的生成周期序列的方法与式 (3.1) 定义的方法等价。

2. 模拟信号的时域采样

模拟信号采样是数字信号处理的一个重要内容。采样中使用的开关函数在理想情况下是冲击函数串，而实际中都是有一定宽度的脉冲串。

模拟信号 $x_a(t)$ 用矩形脉冲串采样的过程可以表示为

$$\hat{x}_a(t) = x_a(t)p_T(t) \quad (3.4)$$

其中：

$$p_T(t) = \sum_{n=-\infty}^{\infty} \text{rect}(t-nT) \quad (3.5)$$

$$\text{rect}(t) = 1, \quad |t| \leq \tau/2 \quad (3.6)$$

【情境任务及步骤】

本案例设置了周期信号生成和信号采样过程模拟共两个情境，情境一设置的目的是体验周期信号不同生成方法的区别，情境二是建立时域信号采样的概念。

一、周期信号生成

已知有限长序列 $x(n) = \{1, 1, 1, 1, 1\}$ 。要求使用三种方法由有限长序列扩展得到周期序列， N 分别取 3, 5, 8，要求产生变量 n 在 $-20 \sim 30$ 范围内的值。

1. 移位叠加出周期

根据式 (3.1) 生成周期序列, 尝试使用小于、等于和大于序列 $x(n)$ 的长度 M 的 N 进行周期扩展 (建议先手动推导序列移位后非零区间变量取值, 再编程实现), 结果分别记为 xp11、xp12 和 xp13。

在同一窗口分三个子图显示结果, 要求作图时周期序列显示 $-10 \leq n \leq 20$ 的范围, 并用红色垂直虚线表示主值区间的界线。

MATLAB 提供的相关函数包括 stem、line 和 subplot, 可以通过 Help 文件学习相关函数的调用语法。

2. 求模得周期

根据式 (3.3) 生成 N 在三种取值下的周期序列, 结果分别记为 xp21、xp22 和 xp23。注意必要时对序列进行补零。

创建一个新的图形窗口, 并在同一窗口分三个子图显示结果, 要求作图时周期序列显示 $-10 \leq n \leq 20$ 的范围, 并用红色垂直虚线表示主值区间的界线。

3. 调用函数 repmat 实现

调用函数 repmat 生成周期为 5 和 8 的序列, 结果分别记为 xp32 和 xp33。

创建一个新的图形窗口, 并将该窗口自上而下分成三个子窗口, 下面两个子窗口分别显示 xp32 和 xp33, 要求作图时周期序列显示 $-10 \leq n \leq 20$ 的范围, 并用红色垂直虚线表示主值区间的界线。

二、信号采样过程模拟

1. 生成待采样的模拟信号

生成 $x_a(t) = \sin(2\pi \times 2 \times 10^3 t)$, 时间长度为 2 个周期。

采样率自定, 建议取不低于信号频率 10 倍以上, 以保证显示波形更为平滑。

2. 生成采样脉冲

调用 rectpuls 函数生成矩形脉冲串 $P_T(t)$, 周期为 $100\mu\text{s}$, 矩形脉冲宽度为 $10\mu\text{s}$ 。整个脉冲串的时间长度满足对 $x_a(t)$ 采样即可。

相关函数的调用语法可以通过 Help 文件学习。

3. 采样

按照式 (3.4) 计算 $x_a(t)$ 与 $P_T(t)$ 的乘积, 并将结果记为 $x_a(nT)$ 。

创建一个新的图形窗口, 并在三个子图中分别显示 $x_a(t) \sim t$ 、 $P_T(t) \sim t$ 、 $x_a(nT)$, 并添加图题。

下角标的加注可以通过在 Help 文件中查找 “Text Properties” 来学习。

【思考题】

(1) 综合比较三种方法在 N 大于、等于和小于 M 三种关系下生成的周期序列, 试评价三种生成方法的优劣。

(2) 试总结 rectpuls 的使用方法。

【总结报告要求】

(1) 情境任务总结报告中简要描述每个情境任务的原理，书写情境任务时可适当进行归纳和总结，但至少要列出【情境任务及步骤】相关内容的各级标题。

(2) 情境任务的程序清单除在报告中出现外，还必须以独立的 m 文件形式单独提交。程序清单要求至少按程序块进行注释。

(3) 执行效果图都至少要标注图题，并附于相关内容之后。

(4) 总结完成本案例所使用的新的函数及其使用语法。

(5) 简要回答【思考题】中的问题。

(6) 报告中还可包括完成本案例的个人心得，对该案例设置的建议等。

【参考程序】

```

%%%%%%%%%%
%%%Codes for mission 1-3,circumstance 1
clear all
close all
clc

xn=ones(1,5);
N1=3;
N2=5;
N3=8;
n0=0:4;
n=-20:30;
lxn=length(xn);
ln=length(n);

figure
stem(n0,xn, '.')
axis([n(1) n(end) -1 3])

tempn=-24:30;
tempxp1=zeros(size(tempn));
tempxp1=[tempxp1 zeros(1,5-1)]
for nind=-24:N1:30
    tempxp1(nind+24+1:nind+24+lxn)=tempxp1(nind+24+1:nind+24+lxn)+xn;
end
xp11=tempxp1(5:5+ln-1);
figure
stem(n,xp11, '.')
hold on
line([0-.2 0-.2],[0 2.5], 'color', 'r');
line([2+.2 2+.2],[0 2.5], 'color', 'r')
grid
axis([n(1) n(end) -1 3])

```



```

tempn=-25:30;
tempxp2=zeros(size(tempn));
tempxp2=[tempxp2 zeros(1,5-1)]
for nind=-25:N2:30
    tempxp2(nind+25+1:nind+25+lxn)=tempxp2(nind+25+1:nind+25+lxn)+xn;
end
xp12=tempxp2(6:6+ln-1);
figure
stem(n,xp12,')
hold on
line([0-.2 0-.2],[0 2.5], 'color','r');
line([4+.2 4+.2],[0 2.5], 'color','r')
grid
axis([n(1) n(end) -1 3])

tempn=-32:30;
tempxp3=zeros(size(tempn));
tempxp3=[tempxp3 zeros(1,5-1)]
for nind=-32:N3:30
    tempxp3(nind+32+1:nind+32+lxn)=tempxp3(nind+32+1:nind+32+lxn)+xn;
end
xp13=tempxp3(13:13+ln-1);
figure
stem(n,xp13,')
hold on
line([0-.2 0-.2],[0 2.5], 'color','r');
line([7+.2 7+.2],[0 2.5], 'color','r')

grid
axis([n(1) n(end) -1 3])

xn=[xn zeros(1,N3-lxn)];
xp21=xn(mod(n,N1)+1);
xp22=xn(mod(n,N2)+1);
xp23=xn(mod(n,N3)+1);
figure
subplot(311)
stem(n,xp21,');
hold on
line([0-.2 0-.2],[0 2.5], 'color','r');
line([2+.2 2+.2],[0 2.5], 'color','r')
grid
axis([n(1) n(end) -1 3])
subplot(312)
stem(n,xp22,');
hold on
line([0-.2 0-.2],[0 2.5], 'color','r');

```

```

line([4+.2 4+.2],[0 2.5],'color','r')
grid
axis([n(1) n(end) -1 3])
subplot(313)
stem(n, xp23, '.');
hold on
line([0-.2 0-.2],[0 2.5],'color','r');
line([7+.2 7+.2],[0 2.5],'color','r')
grid
axis([n(1) n(end) -1 3])

xn=ones(1,5);
tempxp32= repmat(xn,1,11);
xp32=tempxp32(1:51);
xn=[xn zeros(1,N3-lxn)];
tempxp33= repmat(xn,1,7);
xp33=tempxp33(5:5+ln-1);
figure
subplot(311)
subplot(312)
stem(n, xp32, '.');
hold on
line([0-.2 0-.2],[0 6],'color','r');
line([4+.2 4+.2],[0 6],'color','r')
grid
axis([n(1) n(end) -1 3])
subplot(313)
stem(n, xp33, '.');
hold on
line([0-.2 0-.2],[0 6],'color','r');
line([7+.2 7+.2],[0 6],'color','r')
grid
axis([n(1) n(end) -1 3])

%%%%%%%%
%%% Codes for missions 1-3, circumstance 2
Ts=125*1e-4;
duty=1/4;
Pw=Ts*duty;

f=1*1e+1;

ts=Ts/100;
fs=1/ts;

% t=-.002:ts:.002;
t=-1/f*10:ts:1/f*10;

```