

第 1 章 离散时间信号的时域分析

1.1 引言

数字信号处理是处理一种称为输入信号的离散时间信号,以产生另一种称为输出信号的离散时间信号,这种输出信号具有更多人们所需要的特性。在某些应用中,需要使用特定的数字信号处理算法来提取原信号中的一些重要特性。通过观察系统对给定输入信号的输出信号,还可以研究离散时间系统的特性。因此,首先学会用 MATLAB 在时域中产生一些基本的离散时间信号,并对这些信号进行一些基本的运算,是很重要的,这也正是第 1 章练习的主要目的。第二个目的是学会使用基本的 MATLAB 命令,并将它们应用到简单的数字信号处理问题中。

1.2 入门

本书配套文件包括全书所有 MATLAB 程序,提供了在 PC 和 Macintosh PowerPC 下都适用的未完成的报告,还特别提供了前 10 个练习中 PC 和 Macintosh 版本的 MATLAB M 文件和 Word 版本的报告,这些文件按照章节存放在不同的子目录下。完成每一个实验练习例子后,读者可以在指定的位置完成有关该练习的一些问题。

PC 下的练习安装向导^①

按照下面的步骤,可将包含程序和报告的文件夹复制到装有 Windows 95/98 系统的计算机硬盘中:

1. 插入软盘。
2. 双击桌面上的**我的电脑**图标,打开**我的电脑**窗口。
3. 双击相关图标,打开软驱窗口。
4. 双击相关图标,打开将安装该文件夹的目的硬盘驱动器窗口。
5. 在软驱窗口中,将标记为 **PC** 的文件夹拖到目的硬盘驱动器的目录中。

Macintosh PowerPC 下的练习安装指导

按照下面的步骤,可将包含程序和报告的文件夹复制到装有 Mac OS 或更高版本的 PowerPC 的硬盘中:

1. 插入软盘。
2. 双击桌面上显示的图标,打开硬盘驱动器窗口。
3. 双击相关图标,打开软驱的窗口。
4. 在软驱窗口中,将标记为 **MAC** 的文件夹拖到目的硬盘驱动器的目录中。

通过 FTP 和互联网下载

下载这些文件的 FTP 地址是 **iplserv. ece. ucsb. edu**。针对 PC, Macintosh PowerPC 及 UNIX

^① 为便于读者使用,相关文件都改为通过网络获取,但部分译文仍按原著文字翻译。——编者注

工作站的文件目录如下:

pub/mitra/Labs/pc

pub/mitra/Labs/mac

pub/mitra/Labs/unix(只含有 M 文件)

按照以下步骤,可从 FTP 网址将文件下载到 PC 或 UNIX 电脑中:

1. 在电脑中运行 FTP 程序。
2. 输入 **ftp iplserv. ece. ucsb. edu**。
3. 在 **Login**:提示符处,输入 **anonymous**。
4. 在 **Password** 提示符处,输入你的电子邮件地址。
5. 输入 **cd pub/mitra/Labs/pc** 找到适用于 PC 的文件。输入 **cd pub/mitra/Labs/unix** 找到适用于 UNIX 的文件。这些目录中包含有我们要找的 M 文件和报告文件。
6. 输入 **cd** 目录名称,进入存储 M 文件和报告文件的目录。
7. 输入 **asc**,下载 ASCII 文件(例如 M 文件)。输入 **bin**,下载二进制文件,例如 Word 报告文件(对 PC)。输入 **get** 文件名下载所需要的文件类型。最后那条命令会将需要下载的文件存放在本地系统的当前目录中。
8. 使用 **get** 命令,下载其他的文件,或者输入 **cd**,回到前一级目录中。
9. 下载完所有的文件后,输入 **bye**。

将文件下载到 Macintosh 电脑中的 FTP 程序有多种。下面给出来自 Dartmouth College 的 **Fetch** 程序的下载步骤。

1. 运行该 FTP 程序。
2. 在 **New Connection** 对话框中输入下列信息。若看不到该对话框,可从 **File** 菜单中选择 **New Connection** 来打开它。

Host: **iplserv. ece. ucsb. edu**

User ID: **anonymous**

Password: 你的电子邮件地址

Directory/**pub/mitra/Labs/mac**

3. 选定 **OK**。这时会出现一个显示 FTP 站点的目录内容的窗口。双击其他期望的文件夹可进入相应的目录。
4. 双击所要的文件可下载文件。这时会跳出一个对话框。在电脑上选择存储该文件的位置并选定 **Save**。在 **Fetch** 程序的最新版本中,选择目录名并选定 **Get** 后,就可下载整个目录。

通过互联网下载文件的步骤如下:

1. 打开可利用的网络浏览器。
2. 在 URL 窗口中输入 **ftp://iplserv. ece. ucsb. edu**。
3. 双击期望的目录,以进入目录(PC 和 Macintosh 形式的目录如上所示)。
4. 双击所需要的文件进行下载。这时会跳出一个询问将文件存储至何处的对话框。

1.3 基础知识回顾

R1.1 离散时间信号用一个称为样本的数字序列来表示。典型的离散时间信号或序列 $\{x[n]\}$ 的样本值用 $x[n]$ 表示,其中自变量 n 的取值范围是从 $-\infty$ 到 ∞ 之间的整数。为了表示方便,序列 $\{x[n]\}$ 通常不用大括号表示。

R1.2 离散时间信号可以是一个有限长序列,也可以是一个无限长序列。有限长(也称为有限时宽或有限范围)序列仅定义在有限的时间间隔中:

$$N_1 \leq n \leq N_2 \quad (1.1)$$

其中 $-\infty < N_1$ 而 $N_2 < \infty$, $N_2 \geq N_1$ 。有限长序列的长度或时宽为

$$N = N_2 - N_1 + 1 \quad (1.2)$$

R1.3 满足

$$\tilde{x}[n] = \tilde{x}[n + kN] \quad \text{对所有 } n \quad (1.3)$$

的序列称之为周期为 N 的周期序列,其中 N 取任意正整数, k 取任意整数。

R1.4 序列 $x[n]$ 的能量定义为

$$\mathcal{E} = \sum_{n=-\infty}^{\infty} |x[n]|^2 \quad (1.4)$$

有限间隔 $-K \leq n \leq K$ 上的序列能量定义为

$$\mathcal{E}_K = \sum_{n=-K}^K |x[n]|^2 \quad (1.5)$$

R1.5 非周期序列 $x[n]$ 的平均功率定义为

$$\mathcal{P}_{av} = \lim_{K \rightarrow \infty} \frac{1}{2K+1} \mathcal{E}_K = \lim_{K \rightarrow \infty} \frac{1}{2K+1} \sum_{n=-K}^K |x[n]|^2 \quad (1.6)$$

周期为 N 的周期序列 $\tilde{x}[n]$ 的平均功率为

$$\mathcal{P}_{av} = \frac{1}{N} \sum_{n=0}^{N-1} |\tilde{x}[n]|^2 \quad (1.7)$$

R1.6 单位样本序列(通常称为离散时间冲激或单位冲激)用 $\delta[n]$ 表示,其定义为

$$\delta[n] = \begin{cases} 1, & n = 0 \\ 0, & n \neq 0 \end{cases} \quad (1.8)$$

单位阶跃序列用 $\mu[n]$ 表示,其定义为

$$\mu[n] = \begin{cases} 1, & n \geq 0 \\ 0, & n < 0 \end{cases} \quad (1.9)$$

R1.7 指数序列由

$$x[n] = A\alpha^n \quad (1.10)$$

给定。其中 A 和 α 可以是任意实数或任意复数,表示为

$$\alpha = e^{(\sigma_o + j\omega_o)}, \quad A = |A|e^{j\phi}$$

式(1.10)可改写为

$$x[n] = |A|e^{\sigma_o n + j(\omega_o n + \phi)} = |A|e^{\sigma_o n} \cos(\omega_o n + \phi) + j|A|e^{\sigma_o n} \sin(\omega_o n + \phi) \quad (1.11)$$

R1.8 带有常数振幅的实正弦序列形如

$$x[n] = A \cos(\omega_o n + \phi) \quad (1.12)$$

其中 A , ω_o 和 ϕ 是实数。在式(1.11)和式(1.12)中,参数 A , ω_o 和 ϕ 分别称为正弦序列 $x[n]$ 的振幅、角频率和初始相位。 $f_o = \omega_o / 2\pi$ 称为频率。

R1.9 若 $\omega_o N$ 是 2π 的整数倍,当 $\sigma_o = 0$ 时,式(1.11)的复指数序列和式(1.12)的正弦序列是周期序列,即

$$\omega_o N = 2\pi r \quad (1.13)$$

其中 N 为任意正整数,而 r 为任意整数。满足这种情况的最小 N 称为序列的周期。

R1.10 长度 N 的两个序列 $x[n]$ 和 $h[n]$ 的乘积,产生长度也为 N 的序列 $y[n]$

$$y[n] = x[n] \cdot h[n] \quad (1.14)$$

长度为 N 的两个序列 $x[n]$ 和 $h[n]$ 相加,产生长度也为 N 的序列 $y[n]$

$$y[n] = x[n] + h[n] \quad (1.15)$$

用标量 A 与长度为 N 的序列 $x[n]$ 相乘,得到长度为 N 的序列 $y[n]$

$$y[n] = A \cdot x[n] \quad (1.16)$$

无限长序列 $x[n]$ 通过时间反转,可得到无限长序列 $y[n]$

$$y[n] = x[-n] \quad (1.17)$$

无限长序列 $x[n]$ 经过 M 延时,可得到无限长序列 $y[n]$

$$y[n] = x[n - M] \quad (1.18)$$

若 M 是一个负整数,式(1.18)运算得到序列 $x[n]$ 的超前。

长度为 N 的序列 $x[n]$,可被长度为 M 的另一个序列 $g[n]$ 增补,得到长度为 $N + M$ 的更长序列 $y[n]$

$$\{y[n]\} = \{\{x[n]\}, \{g[n]\}\} \quad (1.19)$$

1.4 用到的 MATLAB 命令

这个练习中将用到的 MATLAB 命令如下:

运算符和特殊符号

: . + - * / ; %

基本矩阵和矩阵控制

i ones pi rand randn zeros

基本函数

cos exp imag real

数据分析

sum

二维图形

axis grid legend plot stairs
stem title xlabel ylabel

通用图形函数

clf subplot

信号处理工具箱

sawtooth square

关于这些命令的辅助说明,请参见 *MATLAB Reference Guide* [Mat94] 和 *Signal Processing Toolbox User's Guide* [Mat96],也可在命令窗口中键入 `help` 命令名称。在附录 B 中可找到此处使用的 MATLAB 函数的简要解释。

1.5 序列的产生

本节的目的是使读者熟悉 MATLAB 中产生信号和绘制信号的基本命令。MATLAB 已被开发成能对数据向量或矩阵进行运算的工具。因此,序列以向量的形式存储,并且所有的信号被限定为因果的和有限长的。采用何种步骤执行本书中的程序,要根据 MATLAB 所运行的平台来决定。

基于 Windows PC 的 MATLAB

程序可通过在命令窗口中键入不带 `.m` 的程序名并按回车来执行。也可从命令窗口中的 **File** 菜单上,选择 **Open**,以选定所要的 M 文件,这将在 Editor/Debugger 窗口中打开这个 M 文件,使用 **Tools** 菜单下的 **Run** 命令,可运行该 M 文件。

基于 Macintosh 的 MATLAB

程序可通过在命令窗口中键入不带 `.m` 的程序名并按回车来执行。也可在屏幕上使用 **Open M-File** 命令,将该文件复制到 Editor 窗口,然后在屏幕上选择 **Save and Execute** 命令。

例 1.1 单位样本和单位阶跃序列

此例产生的两个基本离散时间序列分别是式(1.8)和式(1.9)给出的单位样本序列和单位阶跃序列。使用如下 MATLAB 命令,可产生长度为 N 的单位样本序列 $u[n]$

$$u = [1 \text{ zeros}(1,N-1)];$$

延时 M 个样本且长度为 N 的单位样本序列 $ud[n]$, 可用如下的 MATLAB 命令产生, 其中 $M < N$:

```
ud = [zero(1,M) 1 zeros(1,N-M-1)];
```

同样, 使用下面的 MATLAB 命令可产生长度为 N 的单位阶跃序列 $s[n]$

```
s = [ones(1,N)];
```

产生延时单位阶跃序列的方法, 类似于产生延时单位样本序列的方法。

程序 P1.1 产生并绘制了一个单位样本序列。

```
% 程序 P1.1
% 一个单位样本序列的产生
clf;
% 产生从 -10 到 20 的一个向量
n = -10:20;
% 产生单位样本序列
u = [zeros(1,10) 1 zeros(1,20)];
% 绘制单位样本序列
stem(n,u);
xlabel('时间序号 n');ylabel('振幅');
title('单位样本序列');
axis([-10 20 0 1.2]);
```

习题

- Q1.1** 运行程序 P1.1, 以产生单位样本序列 $u[n]$ 并显示它。
- Q1.2** 命令 `clf`, `axis`, `title`, `xlabel` 和 `ylabel` 的作用是什么?
- Q1.3** 修改程序 P1.1, 以产生带有延时 11 个样本的延迟单位样本序列 $ud[n]$ 。运行修改的程序并显示产生的序列。
- Q1.4** 修改程序 P1.1, 以产生单位步长序列 $s[n]$ 。运行修改后的程序并显示产生的序列。
- Q1.5** 修改程序 P1.1, 以产生带有超前 7 个样本的延时单位阶跃序列 $sd[n]$ 。运行修改后的程序并显示产生的序列。

例 1.2 指数信号

另一种基本的离散时间序列是指数序列。此序列可使用 MATLAB 运算符 `^` 和 `exp` 产生。下面给出的程序 P1.2, 可用来生成一个复数值的指数序列。

```
% 程序 P1.2
% 生成一个复指数序列
clf;
c = -(1/12) + (pi/6) * i;
K = 2;
n = 0:40;
x = K * exp(c * n);
subplot(2,1,1);
stem(n,real(x));
xlabel('时间序号 n');ylabel('振幅');
title('实部');
subplot(2,1,2);
```

```
stem(n,imag(x));
xlabel('时间序号 n');ylabel('振幅');
title('虚部');
```

下面给出的程序 P1.3,可用于生成一个实数值的指数序列。

```
% 程序 P1.3
% 生成一个实指数序列
clf;
n = 0:35; a = 1.2; K = 0.2;
x = K*a.^n;
stem(n,x);
xlabel('时间序号 n');ylabel('振幅');
```

习题

- Q1.6** 运行程序 P1.2,以产生复数值的指数序列。
- Q1.7** 哪个参数控制该序列的增长或衰减率? 哪个参数控制该序列的振幅?
- Q1.8** 若参数 c 更改为 $(1/\sqrt{2}) + (pi/6) * i$,将会发生什么情况?
- Q1.9** 运算符 `real` 和 `imag` 的作用是什么?
- Q1.10** 命令 `subplot` 的作用是什么?
- Q1.11** 运行程序 P1.3,以产生实数值的指数序列。
- Q1.12** 哪个参数控制该序列的增长或衰减率? 哪个参数控制该序列的振幅?
- Q1.13** 算术运算符 `^` 和 `.` 之间的区别是什么?
- Q1.14** 若参数 a 小于 1,会发生什么情况? 将参数 a 更改为 0.9,将参数 k 更改为 20,再次运行程序 P1.3。
- Q1.15** 该序列的长度是多少? 怎样才能改变它?
- Q1.16** 使用 MATLAB 命令 `sum(s.*s)` 可计算用向量 s 表示的实数序列 $s[n]$ 的能量。试求在习题 Q1.11 和习题 Q1.14 中产生的实数值指数序列 $x[n]$ 的能量。

例 1.3 正弦序列

本例产生另一种非常有用的实正弦序列,该序列的形式如式(1.12)所示。这样的正弦序列在 MATLAB 中可使用三角运算符 `cos` 和 `sin` 产生。

程序 P1.4 是产生一个正弦信号的简单示例。

```
% 程序 P1.4
% 产生一个正弦序列
n = 0:40;
f = 0.1;
phase = 0;
A = 1.5;
arg = 2 * pi * f * n - phase;
x = A * cos(arg);
clf; % 清除旧图形
stem(n,x); % 绘制产生的序列
axis([0 40 -2 2]);
grid;
```

```
title('正弦序列');
xlabel('时间序号 n');
ylabel('振幅');
axis;
```

习题

- Q1.17** 运行程序 P1.4,以产生正弦序列并显示它。
- Q1.18** 该序列的频率是多少? 怎样可以改变它? 哪个参数控制该序列的相位? 哪个参数控制该序列的振幅? 该序列的周期是多少?
- Q1.19** 该序列的长度是多少? 怎样可以改变它?
- Q1.20** 计算所产生的正弦序列的平均幂。
- Q1.21** axis 和 grid 命令的作用是什么?
- Q1.22** 修改程序 P1.4,以产生一个频率为 0.9 的正弦序列并显示它。把此序列和习题 Q1.17 中产生的序列相比较。修改程序 P1.4,以产生一个频率为 1.1 的正弦序列并显示它。把此序列与习题 Q1.17 中产生的序列相比较,评价你的结果。
- Q1.23** 修改上述程序,以产生长度为 50、频率为 0.08、振幅为 2.5、相移为 90 度的一个正弦序列并显示它。该序列的周期是多少?
- Q1.24** 在程序 P1.4 中用 plot 命令代替 stem 命令,运行新程序。新图形与习题 Q1.17 中产生的图形有什么区别?
- Q1.25** 在程序 P1.4 中用 stairs 命令代替 stem 命令,运行新程序。新图形与习题 Q1.17 和习题 Q1.24 中产生的图形有什么区别?

例 1.4 随机信号

在区间(0,1)中均匀分布的长度为 N 的随机信号,可通过如下的 MATLAB 命令产生:

$$x = \text{rand}(1,N);$$

同理,使用下面的 MATLAB 命令,可产生长度为 N 且具有零均值和单位方差的正态分布的随机信号 $x[n]$

$$x = \text{randn}(1,N);$$

习题

- Q1.26** 编写一个 MATLAB 程序,以产生并显示一个长度为 100 的随机信号,该信号在区间 $[-2,2]$ 中均匀分布。
- Q1.27** 编写一个 MATLAB 程序,以产生并显示一个长度为 75 的高斯随机信号,该信号正态分布且均值为 0,方差为 3。
- Q1.28** 编写一个 MATLAB 程序,以产生并显示五个长度为 31 的随机正弦信号。

$$\{X[n]\} = \{A \cdot \cos(\omega_0 n + \phi)\} \quad (1.20)$$

其中振幅 A 和相位 ϕ 是统计独立的随机变量,振幅在区间 $0 \leq A \leq 4$ 内均匀分布,相位在区间 $0 \leq \phi \leq 2\pi$ 内均匀分布。

1.6 序列的简单运算

正如前面提到的,数字信号处理的目的,是从一个或多个给定的离散时间信号中,产生一个具有我们所需性质的信号。处理算法由诸如加法、标量乘法、时间反转、延时和乘积运算等基本运算的组合所组成(见 R1.10)。在此,我们通过三个非常简单的例子来说明这些运算的应用。

例 1.5 信号平滑

数字信号处理应用的一个常见例子是从被加性噪声污染的信号中移除噪声。假定信号 $s[n]$ 被噪声 $d[n]$ 所污染,得到了一个含有噪声的信号 $x[n] = s[n] + d[n]$ 。我们的目的是对 $x[n]$ 进行运算,产生一个合理逼近 $s[n]$ 的信号 $y[n]$ 。因此,对时刻 n 的样本附近的一些样本求平均,产生输出信号是一种简单有效的方法。例如,采用三点滑动平均算法的表达式如下:

$$y[n] = \frac{1}{3}(x[n-1] + x[n] + x[n+1]) \quad (1.21)$$

程序 P1.5 可用于实现上面的算法。

```
% 程序 P1.5
% 通过平均的信号平滑
clf;
R = 51;
d = 0.8 * (rand(R,1) - 0.5); % 产生随机噪声
m = 0:R-1;
s = 2 * m * (0.9.^m); % 产生未污染的信号
x = s + d'; % 产生被噪音污染的信号
subplot(2,1,1);
plot(m,d','r-',m,s,'g-',m,x,'b-');
xlabel('时间序号 n');ylabel('振幅');
legend('d[n] ','s[n] ','x[n] ');
x1 = [0 0 x];x2 = [0 x 0];x3 = [x 0 0];
y = (x1 + x2 + x3)/3;
subplot(2,1,2);
plot(m,y(2:R+1),'r-',m,s,'g-');
legend('y[n] ','s[n] ');
xlabel('时间序号 n');ylabel('振幅');
```

习题

- Q1.29** 运行程序 P1.5,以产生所有相关的信号。
- Q1.30** 未污染的信号 $s[n]$ 是什么样的形式? 加性噪声 $d[n]$ 是什么样的形式?
- Q1.31** 使用语句 $x = s + d$ 能产生被噪声污染的信号吗? 若不能,为什么?
- Q1.32** 信号 x_1, x_2 和 x_3 与信号 x 之间的关系是什么?
- Q1.33** `legend` 命令的作用是什么?

例 1.6 复杂信号的产生

更复杂的信号可通过在简单信号上执行基本的运算来产生。例如,振幅调制信号可用低

频调制信号 $x_L = \cos(\omega_L n)$ 来调制高频正弦信号 $x_H[n] = \cos(\omega_H n)$ 。得到的信号 $y[n]$ 为

$$y[n] = A(1 + m \cdot x_L[n])x_H[n] = A(1 + m \cdot \cos(\omega_L n)) \cos(\omega_H n)$$

其中 m 称为调制指数,用来确保 $(1 + m \cdot x_L[n])$ 在所有可能的 n 的情况下 m 都是正数。程序 P1.6 可用来产生一个振幅调制信号。

```
% 程序 P1.6
% 振幅调制信号的产生
n = 0:100;
m = 0.4; fH = 0.1; fL = 0.01;
xH = sin(2 * pi * fH * n);
xL = sin(2 * pi * fL * n);
y = (1 + m * xL) .* xH;
stem(n,y); grid;
xlabel('时间序号 n'); ylabel('振幅');
```

习题

Q1.34 在载波信号 $x_H[n]$ 和调制信号 $x_L[n]$ 采用不同频率、不同调制指数 m 的情况下,运行程序 P1.6,以产生振幅调制信号 $y[n]$ 。

Q1.35 算术运算符 $*$ 和 $.*$ 之间的区别是什么?

由于正弦信号的频率是其相位对于时间的导数,为了产生频率随时间线性增加的扫频正弦信号,正弦信号的自变量必须是时间的二次函数。假定自变量的形式是 $an^2 + bn$ (即角频率为 $2an + b$)。在给出的条件(最小角频率和最大角频率)下,求出 a 和 b 的值。程序 P1.7 是产生这种信号的范例程序。

```
% 程序 P1.7
% 产生一个扫频正弦序列
n = 0:100;
a = pi/2/100;
b = 0;
arg = a * n . * n + b * n;
x = cos(arg);
clf;
stem(n,x);
axis([0,100, -1.5,1.5]);
title('扫频正弦信号');
xlabel('时间序号 n');
ylabel('振幅');
grid; axis;
```

习题

Q1.36 运行程序 P1.7,以产生扫频正弦序列 $x[n]$ 。

Q1.37 该信号的最小频率和最大频率是多少?

Q1.38 如何修改上述程序才能产生一个最小频率为 0.1、最大频率为 0.3 的扫频正弦信号?

1.7 工作区信息

命令 `who` 和 `whos` 可用于获得存储于工作区的所有变量的信息。

习题

Q1.39 在命令窗口中键入 `who`。此时在命令窗口中会显示什么信息？

Q1.40 在命令窗口中键入 `whos`。此时在命令窗口中会显示什么信息？

1.8 其他类型的信号(可选)

例 1.7 方波和锯齿波信号

MATLAB 函数 `square` 和 `sawtooth` 可分别用于产生图 1.1 和图 1.2 中显示的序列。

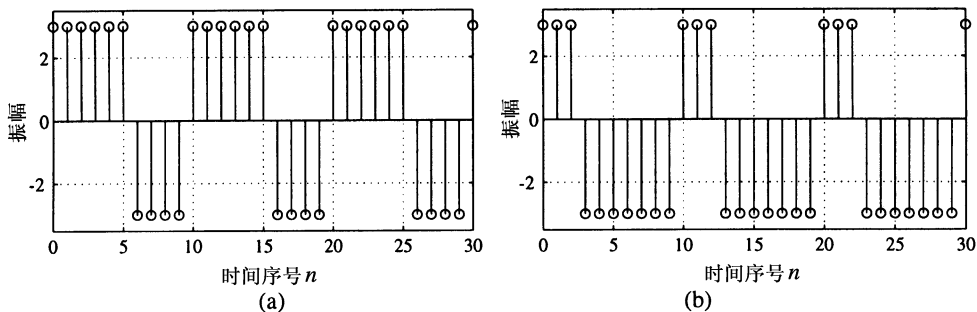


图 1.1 方波序列

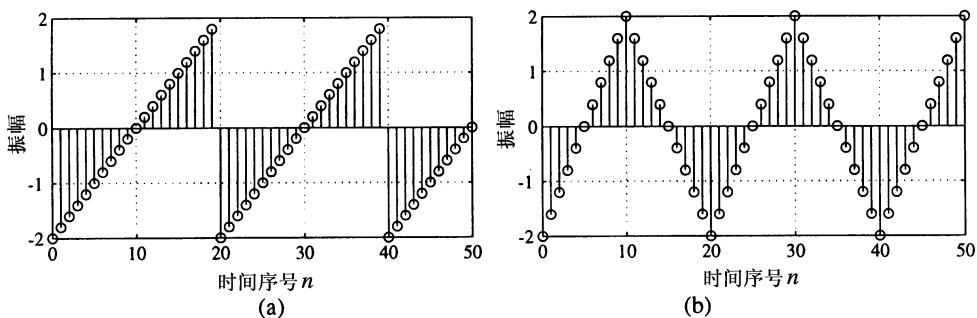


图 1.2 锯齿波序列

习题

Q1.41 编写 MATLAB 程序,以产生图 1.1 和图 1.2 中所示的方波和锯齿波序列,并将序列绘制出来。

1.9 基础知识阅读

- [1] E. Cunningham. *Digital Filtering: An Introduction*. Houghton-Mifflin, Boston MA, 1992. Secs. 1.2–1.3.
- [2] D. J. DeFatta, J. G. Lucas, and W. S. Hodgkiss. *Digital Signal Processing: A System Design Approach*. Wiley, New York NY, 1988. Secs. 2.1.2–2.1.4.
- [3] L. B. Jackson. *Digital Filters and Signal Processing*. Kluwer, Boston MA, third edition, 1996. Secs. 2.2–2.3.
- [4] R. Kuc. *Introduction to Digital Signal Processing*. McGraw-Hill, New York NY, 1988. Secs. 2-2, 2-4.
- [5] L. C. Ludeman. *Fundamentals of Digital Signal Processing*. Harper & Row, New York NY, 1986. Sec. 1.2.
- [6] S. K. Mitra. *Digital Signal Processing: A Computer-Based Approach*. McGraw-Hill, New York NY, 1998. Secs. 2.4–2.5.
- [7] A. V. Oppenheim and R. W. Schaffer. *Discrete-Time Signal Processing*. Prentice-Hall, Englewood Cliffs NJ, 1989. Sec. 2.1.
- [8] B. Porat. *A Course in Digital Signal Processing*. Wiley, New York NY, 1996. Sec. 2.7–2.8.
- [9] J. G. Proakis and D. G. Manolakis. *Digital Signal Processing: Principles, Algorithms, and Applications*. Prentice-Hall, Englewood Cliffs NJ, 1996. Secs. 2.2–2.4.
- [10] R. A. Roberts and C. T. Mullis. *Digital Signal Processing*, Addison-Wesley, Reading MA, 1987. Sec. 2.2.