第 3 章

CHAPTER 3

随机变量的产生

任何建模仿真系统都必须包含比较完善的、能够产生多种分布类型的随机变量的生成模块，这些模块是建模仿真系统中不可缺少的基本组成部分。实际上在建模仿真时，随机数序列是通过某些算法产生的，而系统建模仿真结果的准确性，往往依赖于所产生的随机数序列是否能够准确地再现实际系统随机过程的相关特性。

由于[0,1]均匀分布随机数是其他随机变量产生的基础，本章首先研究均匀分布随机数的产生方法，然后讨论其他类型分布随机变量的具体转换技术；在研究高斯白噪声序列和二进制伪随机序列产生的基础上，分析多进制和相关随机序列的产生方法；最后对于所产生的输入波形，即随机序列质量进行测试评估。

3.1 要求与特点

在建模仿真过程中需要重复地处理大量的随机因素。无论是随机事件的发生时刻，还是持续时间，以及多径信号的传播时延等，都是遵循不同概率分布的随机变量，每次仿真运行都要从这些概率分布中进行随机抽样，以便获得该次仿真运行的实际参数。当进入系统的随机事件数量很多，每个随机事件流经的环节也较多时，建模仿真过程中就需要成千上万次地进行随机抽样，从而使原系统在运行中的随机因素和相互关系得以复现，并得到理想的仿真结果。因此，随机变量的生成模块是建模仿真系统中不可缺少的组成部分。当用户在建模仿真程序中赋予某一随机变量以确定参数的分布类型时，仿真系统能够自动调用和生成相应的随机变量，以保证系统的随机特征在仿真运行中复现。因此，在建模仿真过程中，需要能够自动生成随机变量序列，用来模拟调度随机事件的发生、运行和终止，或者用来模拟建模仿真系统中具有随机性特征的数值。

在通信系统中，信道在传输信号的同时常伴随有噪声的加入，由此看来，产生满足某种统计特性的随机信号和噪声，以及信道的不确定传输特征(例如信道的多径特性)等，都是开展通信系统建模仿真的关键。但是严格地说，通信系统建模仿真中采用的随机数不是在概率论意义下的真正的随机数，而只能称之为伪随机数。由于[0,1]均匀分布随机数在建模仿真系统中的重要性，生成这种类型随机数的模块有一个专门的称谓，即随机数发生器(Random-Number Generator,RNG)。建模仿真系统的其他各种分布类型的随机变量，都是在随机数发生器的基础上进行扩展的，而这类随机数发生器通常都具备以下特点。

1. 随机性(randomness)

这是随机数发生器最重要的性质,它是指所产生的伪随机数序列应当具有独立性、均匀性,并且具有与真实随机数相同的数字特征,如期望、方差等。

2. 长周期(long period)

因为随机数发生器都是基于准确的数学公式描述而设计的,所以产生的随机数序列最终会不可避免地回到序列的起点,重复以前出现过的序列,于是具备固有的周期(period)。所以,好的随机数发生器应当能够生成较长周期的随机数序列,使得在仿真系统运行期间尽量不出现随机序列重复的现象。这一点对仿真系统的可靠性、有效性非常重要,因为所仿真的实际系统出现重复过程的概率非常小,甚至是不可能的。

3. 可再现性(reproducibility)

当调试一个仿真系统时,要求随机数发生器能准确地多次再现同样的随机数序列,这样做的目的是调试、校正仿真系统的个别参数,或者是分析在随机因素可控的情况下,改变其他输入量时,系统输出的变化。当然,有时也要求随机数发生器产生不同以往的随机数序列,以开展系统建模仿真的深入研究。因此,在建模仿真时,根据分析人员的需要,随机数发生器既要能再次生成同样的随机序列,又要能生成不同于以往的随机序列。

4. 计算效率要高(computational efficiency)

因为仿真系统在短时间内需要大量的随机数,所以要求随机数发生器生成每个随机数所花费时间应尽量少。而且随机数发生器不应占用过多的计算机内存,因为特别是可视化仿真系统运行时,有限的内存是非常宝贵的。

大多数常用的发生器运行速度很快,占用内存少,并能很方便地产生随机数序列,但是,并不是所有的随机数发生器都能满足独立性和随机性准则,这时就需要进行随机数发生器的性能测试和检验。

计算机是具有完全确定性的机器,在实现随机性方面,表现并不尽如人意。因此,当仿真系统中需要一个或一组真正的随机数时,必须通过某种方式或者算法近似地生成这些随机数。实际上,这些数值并非真正的“随机”,因为它们都是根据某种固定不变的方式生成的;然而,它们又能表现出某种程度的“随机性”,进而满足仿真系统对随机变量统计特性的要求,所以在实际建模仿真应用中,通常假定这些数值是随机的,并且称它们为伪随机数(pseudorandom number)。

3.2 随机数的产生

在对某随机过程进行建模仿真时,通常假设该随机过程是各态历经的。在第 2 章已经讲过,各态历经性可以理解为平稳过程的各个样本都同样地经历了随机过程的各种可能状态;各态历经性还意味着,随机过程的统计特性仅由一维和二维特性即可确定,并且这些特性是时不变的。上述假设不仅简化了随机过程的模型,而且使得产生随机数的算法易于构建,同时可以准确地获得这些随机数的一维和二维统计特性,以及均值、方差、相关函数和功率谱密度等随机过程的主要数字特征。

在建模仿真应用中,所有随机过程需要由随机变量序列来表示。因此,根据二维分布函数(概率密度函数)和相关函数(功率谱密度),就能够确定获得产生随机数的具体方法。各

类形式随机数产生的基础，首先是产生具有独立均匀分布的随机数序列，然后让该序列经过无记忆非线性变换，就能够得到一个具有任意维分布（通常不超过二维）的独立序列。同样，通过利用有记忆功能的线性或非线性变换，也能将一个独立序列变换成具有任意相关性和功率谱密度的序列。因此，在本节中首先介绍均匀分布随机数的产生。

3.2.1　均匀分布随机数的产生

为了提高运算速度，提升计算效率，均匀分布随机数的产生通常采用迭代算法，其中比较有代表性的方法被称为同余算法或幂剩余算法，所使用的迭代公式如下：

$$X(k)=[aX(k-1)+c]\bmod M \tag{3-1}$$

其中，a 是在 1 与 M 间根据某种规则选择的整数，M 是一个很大的素数，或者是一个素数的整数幂。利用初始种子值 $X(0)$启动随机数的产生迭代过程，这里取 $0<X(0)<M$。

式(3-1)将产生均匀分布于$[0,M-1]$区间的整数，将 $X(k)$除以 M，即

$$U(k)=\mathrm{Float}\left[\frac{X(k)}{M}\right] \tag{3-2}$$

式中，$U(k)$就是近似在$[0,1]$区间内均匀分布的随机数序列。

式(3-1)的输出结果属于$[0,M-1]$的一个整数集，最多有 M 个状态，因此，输出序列的最大周期是 M。随机数生成器输出结果的统计特性和周期取决于式(3-1)中 a、c、M 的选择，有时甚至会受到初始种子值 $X(0)$的影响。但是一个好的随机数生成器是通过精心选择 a、c、M 这些参数来设计的，其统计特性通常不受初始种子值 $X(0)$的影响，$X(0)$仅仅指定了周期性输出序列的初始值。

评价随机数生成器的输出结果，需要关注它的两个重要属性：

(1) 代数（时间的）特性，例如输出序列的结构和周期。

(2) 统计特性，例如输出序列的分布情况。

若 a、c、M 这些参数选择恰当，就可以得到代数特性和统计特性均优良的输出结果，同时还可以提高计算效率，并且兼容各种形式的运算结构。

如果从输出序列的结构性能方面考虑，则要求随机数生成器的输出序列是互不相关的，并且应当具有最大的重复周期。输出序列的重复周期在仿真中起着至关重要的作用，仿真时输出序列的周期必须比仿真长度长出许多，否则部分仿真输出将会出现重复。这时，仿真精度并不取决于仿真长度，而是取决于输出序列的周期。

前面已经说明式(3-1)输出序列的最大周期为 M，只有精心选择 a、c、M 这些参数才能保证具有最大周期为 M 的输出序列。除了要求输出序列应当具有最大周期外，还要求输出序列尽可能地互不相关。可以证明，在仿真时基于相关采样值的估计与基于不相关采样值的估计相比，将会出现更大的方差。式(3-1)产生的序列的特性和结构已被广泛研究，但只有几个随机数发生器被认为具有较好的特性和结构。

为了在字长为 32 位的计算机上产生随机数，假设其最高位表示符号，剩下的 31 位表示数值。在这种情况下，最通用的均匀分布随机数产生器的迭代公式如下：

$$X(k)=[16807\cdot X(k-1)]\bmod(2^{31}-1) \tag{3-3}$$

应用式(3-3)产生的序列，其周期为 $2^{31}-2$。如果利用上述相同结构产生 32 位无符号均匀分布随机数，其随机数产生器的迭代公式如下：

$$X(k)=[69069\cdot X(k-1)+1]\bmod(2^{32}) \tag{3-4}$$

应用式(3-4)产生的序列,其周期为 2^{32}。将式(3-3)或式(3-4)产生的随机数,利用式(3-2)进行处理,就能够产生在[0,1]区间上均匀分布的随机数序列。

如果要产生更长周期的随机数序列,可以利用同余算法的线性组合来构建随机数生成器。具体表达式如下:

$$X(k)=[a_1X(k-1)+a_2X(k-2)+\cdots+a_mX(k-m)]\bmod p \tag{3-5}$$

式中,$(a_1,a_2,\cdots,a_m)$是 GF(p)上的本原多项式的系数,即

$$f(x)=x^m-a_1x^{m-1}-\cdots-a_m \tag{3-6}$$

式(3-5)确定的随机数生成器产生的序列周期为 p^m-1。比较式(3-1)和式(3-5)可以看到,每输出一个采样值,后者都要进行更多的运算,但同时将产生周期更长的序列。

目前计算机上运行的多种应用程序,例如 Matlab 或 C 语言中均拥有的均匀随机序列产生器,是对计算机的运算量和字长的优化算法。尽管这些随机数产生器是可靠的,但在很多情况下,它们并不能产生具有最大周期和良好统计特性的序列。

在开发性能优越,结构合理的均匀随机数产生器方面,人们做了大量工作,提出了多种实用的算法。这些算法计算效率高,产生的序列周期长,并且具有良好的统计特性。比较有代表性的算法包括 Wichman-Hill 算法和 Marsaglia-Zaman 算法等。

1. Wichman-Hill 算法

通过观察式(3-5)可以发现,利用两个短周期序列进行合理的线性组合,构建的随机数生成器能够产生长周期序列。例如,将两个周期分别为 N_1、N_2 的波形相加,所得到波形的周期为

$$N=\mathrm{lcm}(N_1,N_2) \tag{3-7}$$

式中,N 是 N_1 和 N_2 的最小公倍数。

如果 N_1 和 N_2 互质,则

$$N=N_1\times N_2 \tag{3-8}$$

因此,通过合并几个随机数生成器的输出,就可以产生一个长周期的序列。基于上述原则的 Wichman-Hill 算法,按以下步骤合并了三个随机数生成器的输出:

$$X(k)=[171\cdot X(k-1)]\bmod 30269 \tag{3-9a}$$

$$Y(k)=[171\cdot Y(k-1)]\bmod 30307 \tag{3-9b}$$

$$Z(k)=[170\cdot Z(k-1)]\bmod 30323 \tag{3-9c}$$

然后计算

$$U(k)=\frac{\dfrac{X(k)}{30269}+\dfrac{Y(k)}{30307}+\dfrac{Z(k)}{30323}}{3} \tag{3-10}$$

算法中前三步是整形运算,最后一步是浮点运算,此算法得到的序列周期是

$$p=30268\times 30306\times 30322\approx 2.78\times 10^{13}$$

可以证明,如果处理上述算法的计算机变量字长超过 24 位,算法就能正常运行,并且输出一个具有长周期和良好统计特性的随机序列。就目前计算机而言,几乎所有计算机处理字长均超过 24 位,因此,Wichman-Hill 算法得到了广泛应用。当然与式(3-1)相比,Wichman-Hill 算法显得更为复杂,但是它实现便捷,在使用较小的硬件代价(24 位字长)的

条件下，就能够产生具有较长周期的随机序列。

2. Marsaglia-Zaman 算法

Marsaglia-Zaman 算法是由学者 Marsaglia 和 Zaman 共同提出的。该算法是一个线性迭代算法，它有两种相似的版本：带借位的减法和带进位的加法。下面就给出带借位的减法形式的线性迭代算法：

$$Z(k)=X(k-r)-X(k-s)-C(k-1) \tag{3-11}$$

式中

$$X(k)=\begin{cases}Z(k), & Z(k)\geqslant 0\\ Z(k)+b, & Z(k)<0\end{cases}$$

$$C(k)=\begin{cases}0, & Z(k)\geqslant 0\\ 1, & Z(k)<0\end{cases}$$

在该算法中，常数 b、r 和 s 均是正整数，借位位初值为 0，即 $C(0)=0$。Marsaglia 和 Zaman 指出，为了保证输出的序列周期最大，即周期为 $M-1$，则常数 b、r 和 s 必须满足下面的规则：

$$M=b^r-b^s+1 \tag{3-12}$$

式中，M 为素数，b 是模 M 的本原根。

对于字长为 32 位的计算机，$b=2^{32}-5$，$r=43$，$s=22$，则产生的序列周期为

$$M-1=b^r-b^s\approx 1.65\times 10^{414} \tag{3-13}$$

为了得到在[0,1]区间上均匀分布的随机数序列，需要将 $X(k)$ 换成 $U(k)$，即

$$U(k)=\mathrm{Float}\left[\frac{X(k)}{b}\right] \tag{3-14}$$

3.2.2　任意概率密度函数随机数的生成

在通信系统建模仿真过程中，通常需要产生某种分布形式的随机数序列。虽然产生各类分布形式随机数序列的方法有许多种，但是在实际仿真过程中，以下两方面需要重点考虑。

(1) 准确性需求。产生的随机数序列应准确地具备所要求的分布形式。

(2) 快速性需求。在建模仿真中，一次运行往往需要产生几万甚至几十万个随机数，这样，产生随机数序列的速度将极大地影响建模仿真执行的效率。

本节将介绍 4 种生成随机数的方法，它们的概率密度函数可以是任意形式，这些方法包括解析变换法、经验变换法、离散随机变量的变换法和产生随机数的接收/拒收法。

1. 解析变换法

对于概率密度函数是任意形式的随机数序列，变换法是最常使用且最为直观的随机数产生方法，它以概率积分变换定理为基础，通过对均匀分布随机变量 U 的变换，可以得到具有任意概率密度函数的随机变量 Z。

设需要产生的随机变量 Z 的分布函数为 $F(Z)$，为了得到该随机变量的抽样值，先在[0,1]区间上产生均匀分布的独立随机变量 U，求其反分布函数 $F^{-1}(U)$ 所得到的值，即为所需要的随机变量，即

$$Z=F^{-1}(U) \tag{3-15}$$

由于这种方法对分布函数进行反变换，因此有时也称为解析反变换法。反变换法的原理可用图 3-1 加以说明。

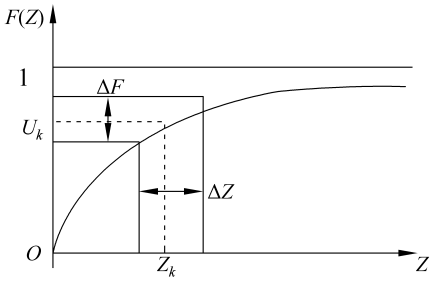

图 3-1　产生随机数的变换方法

从图 3-1 可以看出，随机变量概率分布函数 $F(Z)$ 的取值范围为[0,1]，以在[0,1]区间上均匀分布的独立随机变量作为 $F(Z)$ 的取值规律，则落在 ΔZ 内的样本个数的概率就是 ΔF，从而随机变量 Z 在区间 ΔZ 内出现的概率密度函数的平均值为 $\Delta F/\Delta Z$。当 ΔZ 趋于 0 时，其概率密度函数就等于 $\mathrm{d}F/\mathrm{d}Z$，即符合原来给定的分布函数。

这样看来，由任意概率密度函数生成随机数的方法可以用以下两步实现：

(1) 产生在[0,1]区间上均匀分布的独立随机变量 U。

(2) 根据 Z 的概率密度函数确定分布函数 $F(Z)$，输出 $Z=F^{-1}(U)$。

如果随机变量 Z 的分布函数的逆函数可以用解析式进行表述，在上述变换中步骤(2)，即 $Z=F^{-1}(U)$ 便能够准确地表示，否则，$F(\cdot)$ 和 $F^{-1}(\cdot)$ 必须用适当的数值方法进行计算。例如，当已知概率密度函数 $f(Z)$ 时，经过数值积分可以得到 $F(Z)$。将结果存储在一个表中，通过搜索和内插 $F(Z)$ 表中的值，就可以得到逆变换 $F^{-1}(U)$。

例 3-1　利用解析变换法建立一个产生指数型分布随机变量的算法。

解：指数型分布随机变量的概率密度函数为

$$f(Z)=\begin{cases}\lambda \mathrm{e}^{-\lambda Z}, & Z>0\\ 0, & Z\leqslant 0\end{cases}$$

由 $f(Z)$ 可得到 Z 的分布函数为

$$F(Z)=1-\mathrm{e}^{-\lambda Z},\quad Z>0$$

根据实现任意概率密度函数生成随机数的方法，令

$$U=F(Z)=1-\mathrm{e}^{-\lambda Z},\quad Z>0$$

或者

$$Z=F^{-1}(U)=\left(-\frac{1}{\lambda}\right)\ln(1-U)$$

这里，U 是在[0,1]区间上均匀分布的独立随机变量。容易得出结论，$1-U$ 也是在[0,1]区间上均匀分布的独立随机变量。因此，忽略减法运算就得到

$$Z=\left(-\frac{1}{\lambda}\right)\ln U \tag{3-16}$$

式(3-16)就是产生指数型分布随机变量的算法，其中，U 是在[0,1]区间上均匀分布的独立随机变量。

产生指数型分布随机变量的算法如下：

(1) 产生在[0,1]区间上均匀分布的独立随机变量 U。

(2) 计算 $Z=\left(-\frac{1}{\lambda}\right)\ln U$。

由上面例子可以看出，利用解析变换法产生随机变量时，首先必须用随机数发生器产生在[0,1]区间上均匀分布的独立随机变量$U(k)$，以此为基础得到的随机变量X才能保证分布的准确性。可见，$U(k)$的均匀性和独立性，是高质量随机数发生器的基础。

2. 经验变换法

当分布函数解析表达式不存在时，就不能利用用式(3-15)进行反变换，实现随机数的产生。此时，只能利用经验搜索算法来替代解析变换法。具体做法是，首先对连续的随机变量Z分布函数进行量化处理，也就是将Z的取值范围均匀地划分为N份，如图3-2所示，p_1，p_2，…，p_N表示N个单元的概率，则可以采取以下方式产生随机变量Z的取样。

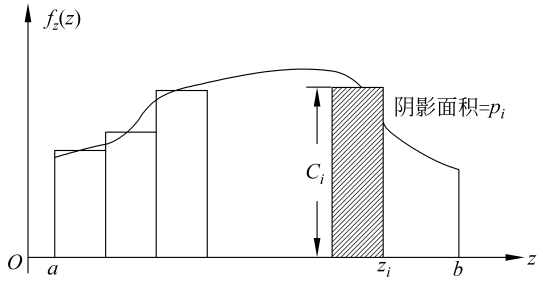

图3-2　任意分布形式产生随机数

(1) 产生在[0,1]区间上均匀分布的独立随机变量U。

(2) 令$F_i=\sum_{j=1}^{i}p_j$，$i=0,1,\cdots,N$，且$F_0=0$。

(3) 找到满足下式最小的i值：$F_{i-1}<U\leqslant F_i$，其中，$i=1,2,\cdots,N$。

(4) 输出$Z=Z_{i-1}+(U-F_{i-1})/C_i$，返回第(1)步。

上述算法中的最后一步是为了获得区间$[Z_{i-1},Z_i]$上的插值。

3. 离散随机变量的变换法

当Z是离散随机变量时，其反变换法的形式略有不同，原因在于离散随机变量的分布函数也是离散的，因而不能直接利用反函数来获得随机变量的抽样值，下面讨论这类随机变量的反变换法。

设离散随机变量Z分别以概率$p(z_1)$，$p(z_2)$，…，$p(z_N)$取值z_1，z_2，…，z_N，其中$0<p(z_i)<1$，且$\sum_{i=1}^{N}p(z_i)=1$，其分布函数如图3-3所示。

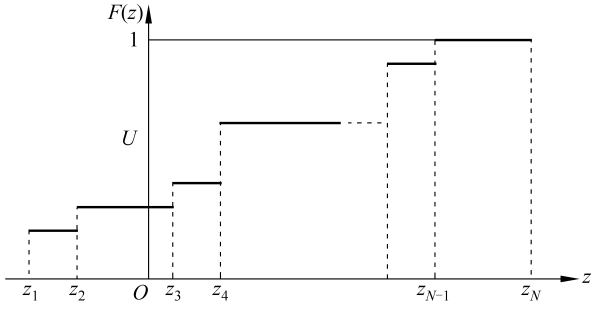

图3-3　离散随机变量分布函数

为使用反变换法获得离散随机变量，先将[0,1]区间上按 $p(z_1)$，$p(z_2)$，…，$p(z_N)$的值分成 N 个子区间，然后产生在[0,1]区间上均匀分布的独立随机变量 U。根据 U 的值落在何区间，相应区间对应的随机变量就是所需要的随机变量 z_i。如果 N 取有限值，则得到有限型离散随机变量产生方法，将该方法写成相应的算法。

有限型离散随机变量 Z 的产生算法如下：

(1) 设定 $k=1$。

(2) 产生在[0,1]区间上均匀分布的独立随机变量 U。

(3) $U \leqslant F_k$，输出 $Z=z_k$，并返回第(1)步，否则跳到第(4)步。

(4) 令 $k=k+1$，返回第(3)步。

在算法中 $p(z_i)$和 $F(z_i)$都是预先计算出来的，并且存放在一个表中，具体产生过程见下面的例 3-2。

例 3-2 设离散随机变量 Z 的分布概率 $p(z_i)$以及累积分布函数如下表，用离散随机变量变换法产生随机变量 Z。

z_k	0	1	2	3	4	5
$p(z_k)$	0	0.1	0.51	0.19	0.15	0.05
$F(z_k)$	0	0.1	0.61	0.80	0.95	1.00

根据给出的算法，首先由随机数发生器产生[0,1]区间上均匀分布的独立随机变量 U。假设 $U=0.72$，按算法步骤(3)，判断是否 $U<F(z_1)$；若条件不满足，再判断是否 $U<F(z_2)$；若仍不满足，再判断是否 $U<F(z_3)$，满足 $F(z_2)<U<F(z_3)$，从而得到 $Z=z_3=3$。然后产生下一个离散随机变量 Z。

在上述算法介绍中，N 取的是有限值，以此为基础，构成了有限型离散随机变量产生方法。当然，N 有无穷多种取值情况时，就必须按无限型离散随机变量产生方法产生随机变量。

无限型离散随机变量的产生算法如下：

(1) 设定 $k=1$，令 $C=p_1$，$B=C$。

(2) 产生在[0,1]区间上均匀分布的独立随机变量 U。

(3) 如果 $U \leqslant B$(即 $U \leqslant F_k$)，输出 $Z=z_k$，并返回第(1)步，否则跳到第(4)步。

(4) 令 $k=k+1$。

(5) 令 $C=A_{k+1}C$，$B=B+C$，其中 $A_{k+1}=\dfrac{p_{k+1}}{p_k}$，返回第(3)步。

当离散随机变量 Z 有无穷多种取值情况时，$p(z_i)$和 $F(z_i)$是无法存放在一个表中的，只能在搜索程序中通过计算产生。例如，对于泊松分布，$\dfrac{A_{k+1}}{A_k}=\dfrac{\lambda}{k+1}$，因此第(5)步很容易实现，而且整个程序的计算效率很高。

4. 舍选法

上面介绍的三种方法有一个共同的特点，即直接面向分布函数，因而可以统称为直接法。它们均以反变换法为基础。但是，当反变换法难于使用时(例如随机变量的分布函数不存在封闭形式)，舍选法就成为产生随机数的主要方法之一。下面介绍这种方法的基本

思想。

设随机变量 X 的概密度函数为 $f(X)$，$f(X)$的最大值为 C。如果独立地产生两个[0,1]区间上均匀分布的随机变量 U_1 和 U_2，则 CU_1 是在[0,C]区间上均匀分布的随机变量；若以随机变量 U_2 求 $f(U_2)$的值，显然，满足 $CU_1 \leqslant f(U_2)$的概率可以表示为

$$P\{CU_1 \leqslant f(U_2)\} = \int_0^1 \mathrm{d}X \int_0^{f(X)} \mathrm{d}Y / C = \frac{1}{C} \tag{3-17}$$

舍选法的做法是：若式(3-17)成立，则选取 U_2 为所需要的随机变量 X，即 $X = U_2$，否则舍弃 U_2。

舍选法的算法实现如下：

(1) 确定 $f(X)$的最大值为 C。

(2) 分别产生在[0,1]区间上均匀分布的独立随机变量 U_1 和 U_2。

(3) 令 $V = CU_1$，这时 V 变成在[0,C]区间上均匀分布的独立随机变量。

(4) 如果 $CU_1 \leqslant f(U_2)$，则输出 $X = U_2$；否则，拒收 U_2 并返回第(2)步。

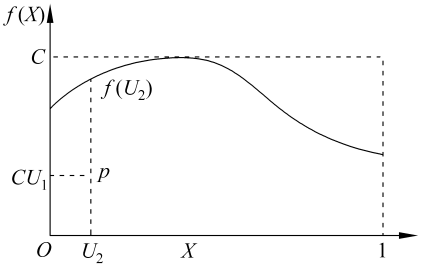

图 3-4　舍选法产生随机变量

下面，结合图 3-4 来解释舍选法的数学机理。

从图形上看，在 $1 \times C$ 这块矩形面积上任意投下一点 p，p 的纵坐标为 CU_1，横坐标为 U_2。若该点位于 $f(X)$曲线下面，则认为抽样成功。成功的概率为 $f(X)$曲线下的面积除以总面积 C，$f(X)$下的面积的值等于分布函数的值。由于假设随机变量 X 的取值范围为[0,1]，因此该面积的值为 1，那么成功的概率就是 $1/C$，成功抽样的点符合随机变量 X 分布。

3.2.3　高斯随机变量的产生

在通信建模仿真中，高斯随机变量是一类重要的随机变量，其概率密度函数在本书第 2 章中已经给出。标准高斯分布的概率密度函数(即均值为 0，方差为 1)可以表示为

$$f(X) = \frac{1}{\sqrt{2\pi}} \exp\left[-\frac{X^2}{2}\right] \tag{3-18}$$

通常将标准高斯分布表示为 $N(0,1)$。下面就介绍两种 $N(0,1)$随机变量产生方法。

1. 12 求和方法

在客观实际中有这样一类随机变量，它们是由大量相互独立的随机因素综合影响所形成的，而且其中每一个因素在总的影响中所起的作用都是微小的。这种随机变量往往近似地服从正态分布，这种现象就是中心极限定理的客观背景。因此，产生 $N(0,1)$随机变量的最简单方法是利用中心极限定理，这样就可以构建出生成公式

$$X = \sum_{k=1}^{12} U(k) - 6.0, \quad k = 1,2,\cdots,12 \tag{3-19}$$

式中，$U(k)$是在[0,1]上均匀分布，同时相互独立的随机变量，其均值为 0.5，方差为 1/12。

利用中心极限定理可以证明，式(3-19)中的随机变量 X 与均值为 0，方差为 1 的高斯随机变量很接近。需要注意的是，尽管 $N(0,1)$随机变量的取值范围是$(-\infty,\infty)$，而

式(3-19)确定的随机变量 X 的取值范围是(−6.0,6.0),但是考虑到 $N(0,1)$随机变量的产生速度与“准确性”的相互矛盾,将 k 的取值范围确定为 12 是一个恰当的选择。

2. Box-Muller 算法

尽管正态分布随机变量的分布函数没有直接的封闭形式,但将其转换到极坐标系后,可以得到其封闭形式,这时就可以采用反变换法产生正态分布随机变量,下面对这一方法加以说明。

设 X 和 Y 是两个相互独立的 $N(0,1)$随机变量,则其联合概率密度函数可以表示为

$$f(X,Y)=\frac{1}{2\pi}\exp\left(-\frac{X^2+Y^2}{2}\right) \tag{3-20}$$

将其转换成极坐标形式:

$$\begin{cases}X=\rho\cos\varphi\\ Y=\rho\sin\varphi\end{cases} \tag{3-21}$$

则

$$f(\rho,\varphi)=f(X,Y)\cdot|\boldsymbol{J}| \tag{3-22}$$

式中,$|\boldsymbol{J}|$为雅可比行列式,即

$$|\boldsymbol{J}|=\begin{vmatrix}\dfrac{\partial X}{\partial\rho} & \dfrac{\partial X}{\partial\varphi}\\ \dfrac{\partial Y}{\partial\rho} & \dfrac{\partial Y}{\partial\varphi}\end{vmatrix}=\begin{vmatrix}\cos\varphi & -\rho\sin\varphi\\ \sin\varphi & \rho\cos\varphi\end{vmatrix}=\rho \tag{3-23}$$

从而可得

$$f(\rho,\varphi)=\frac{\rho}{2\pi}\exp\left(-\frac{\rho^2}{2}\right) \tag{3-24}$$

在上式中,$\rho\geqslant 0$,φ 在$(0,2\pi)$内取值,这样,利用概率论中的边际分布知识,可求得 $f(\rho)$:

$$f(\rho)=\int_{-\infty}^{\infty}f(\rho,\varphi)\mathrm{d}\varphi=\int_0^{2\pi}\frac{\rho}{2\pi}\exp\left[-\frac{\rho^2}{2}\right]\mathrm{d}\varphi=\rho\exp\left[-\frac{\rho^2}{2}\right] \tag{3-25}$$

式(3-25)表示 ρ 服从瑞利分布,而

$$f(\varphi)=\int_{-\infty}^{\infty}f(\rho,\varphi)\mathrm{d}\rho=\int_0^{\infty}\frac{\rho}{2\pi}\exp\left[-\frac{\rho^2}{2}\right]\mathrm{d}\rho=\frac{1}{2\pi} \tag{3-26}$$

可见,φ 服从均匀分布,且有

$$f(\rho,\varphi)=f(\rho)\cdot f(\varphi) \tag{3-27}$$

由式(3-27)可见,ρ 和 φ 是统计独立的,它们相应的分布函数分别为

$$\begin{cases}F(\rho)=\int_0^{\rho}\rho'\exp\left(-\dfrac{\rho^2}{2}\right)\mathrm{d}\rho=1-\exp\left(-\dfrac{\rho^2}{2}\right)\\ F(\varphi)=\int_0^{\varphi}\dfrac{1}{2\pi}\mathrm{d}\varphi'=\dfrac{\varphi}{2\pi}\end{cases} \tag{3-28}$$

由式(3-28)可见,对随机变量 ρ 和 φ 来说,它们的分布函数均具有封闭形式。因而可采用反变换法,即独立地产生两个[0,1]区间上均匀分布的随机变量 U_1 和 U_2,分别对 $F(\rho)$及 $F(\varphi)$进行反变换,可得

$$\begin{cases}\rho=\sqrt{-2\ln(1-U_2)}\\ \varphi=2\pi U_1\end{cases} \tag{3-29}$$

由于 $1-U_2$ 也是[0,1]区间上均匀分布的随机变量,因此式(3-29)可以写为

$$\begin{cases}\rho=\sqrt{-2\ln U_2}\\ \varphi=2\pi U_1\end{cases} \tag{3-30}$$

根据 X 和 Y 与 ρ 和 φ 之间的变换关系,可得

$$\begin{cases}X=[-2\ln(U_2)]^{1/2}\cos(2\pi U_1)\\ Y=[-2\ln(U_2)]^{1/2}\sin(2\pi U_1)\end{cases} \tag{3-31}$$

采用上述反变换法,每次能够产生 X 和 Y 两个相互独立的 $N(0,1)$ 随机变量,数值范围在$(-\infty,\infty)$上分布。这种方法直观,易于理解,但是由于要进行三角函数及对数函数运算,因此计算速度比式(3-19)产生 $N(0,1)$ 随机变量慢。需要强调的是,为了保证输出随机变量的动态范围,式(3-31)必须在双精度的运算条件下执行。

3.3　独立随机序列的产生

前面已经讨论了在给定概率密度函数时,产生随机数序列的方法,下面给出独立随机变量序列的产生方法。

3.3.1　高斯白噪声序列

第 2 章已经讨论了理想的白噪声有关性质,其功率谱密度通常被定义为

$$P_n(f)=\frac{n_0}{2},\quad -\infty<f<\infty \tag{3-32}$$

式中,n_0 是常数,单位是 W/Hz。

当白噪声幅度的分布满足正态分布时,称这种噪声为高斯白噪声。如果噪声的功率谱密度如式(3-32) 所示,相应的自相关函数为

$$R(\tau)=\frac{1}{2\pi}\int_{-\infty}^{\infty}\frac{n_0}{2}e^{j\omega\tau}d\omega=\frac{n_0}{2}\delta(\tau) \tag{3-33}$$

显然,高斯白噪声序列是不相关序列,由于又满足高斯分布,这个序列也是统计独立的。然而,在实际系统中,带宽总是有限的。假设带宽为 B,根据采样定理,建模仿真的采样频率 f_s 必须大于 $2B$,因此,建模仿真时采用的是带宽有限的高斯白噪声,这时的噪声在仿真带宽$(-f_s/2,f_s/2)$内具有恒定的功率谱密度,即

$$P_{nc}(f)=\begin{cases}n_0/2, & |f|\leqslant f_s/2\\ 0, & \text{其他}\end{cases} \tag{3-34}$$

与式(3-34)对应,其相关函数为

$$R(\tau)=\int_{-f_s/2}^{f_s/2}\frac{n_0}{2}e^{j2\pi f\tau}df=\frac{f_s n_0 Sa(\omega_0\tau)}{2} \tag{3-35}$$

式中,$\omega_0=\pi f_s=\dfrac{\pi}{T_s}$。

需要注意,对于建模仿真系统而言,无论输入功率谱密度是 P_n 还是 P_{nc},系统的响应都是一样的。这是因为,对于式(3-35),相关函数在 $\tau=kT_s(k=1,2,3,\cdots)$时,其自相关函数为零,也就是说,如果在这些时刻进行采样,其采样值在时域上是不相关的。对于正态分布

的随机过程(即高斯白噪声)而言,在这些时刻进行采样,其采样值之间是统计独立的。因此,带限高斯白噪声的采样值,可以用高斯变量的一个独立序列来仿真。

具体来讲,均值为零,方差为

$$\sigma_{nc}^2 = \frac{n_0 f_s}{2} \tag{3-36}$$

的一个序列,可以用 $N(0,1)$随机数发生器的输出乘以 σ_{nc}来产生。

3.3.2 二进制伪随机序列

二元随机序列$\{a_k\}$是由统计独立的 0 和 1 码元组成的序列。如果每个 0 和 1 发生的概率均为 0.5(等概率),它就可以通过在[0,1]区间均匀分布的随机变量 $U(k)$来产生,即

$$a_k = \begin{cases} 1, & U(k) > 0.5 \\ 0, & U(k) \leqslant 0.5 \end{cases} \tag{3-37}$$

与上述二元随机序列对应,可以预先确定、重复实现的序列称为确定序列。而在一定条件下,具有某些随机特性,貌似随机序列的确定序列则称为伪随机序列,或伪噪声(PN)码。

产生伪随机序列的方法很多,例如,带有反馈的移位寄存器是其中最易实现的电路。根据结构的不同,上述电路又可进一步分为线性反馈移位寄存器和非线性反馈移位寄存器两类。由线性反馈移位寄存器产生的周期最长的二进制数字序列,被称为最大长度线性反馈移位寄存器序列,简称为 m 序列。由于它的理论成熟、实现简便,在实际中被广泛应用。但是,由于 m 序列的种类有限,不能完全满足某些具体需要,因此,带有非线性反馈的移位寄存器产生的一些伪随机序列也有一定的应用价值。关于这方面的内容,请参阅相关文献。

本小节将结合通信建模仿真系统的具体需求,重点讨论 m 序列的产生原理,以及它的有关性质。

1. m 序列的产生

m 序列是由带线性反馈的移位寄存器产生的周期最长的一种序列。为了掌握其工作原理,首先给出一个关于 m 序列的例子,如图 3-5 所示。

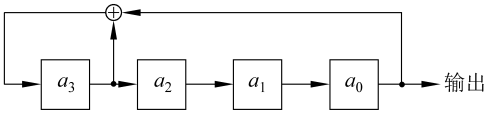

图 3-5 m 序列产生器

在图 3-5 中给出了一个 4 级反馈移位寄存器。若其初始状态为$(a_3,a_2,a_1,a_0)=(1,0,0,0)$,在移位一次时,由 a_3 和 a_0 的输出模 2 相加产生新的输入 $a_4=1\oplus 0=1$,则新的状态变为$(a_4,a_3,a_2,a_1)=(1,1,0,0)$。这样移位 15 次后又回到初始状态(1,0,0,0),具体输出情况如表 3-1 所示。

不难看出,若初始状态为全"0",即(0,0,0,0),则移位后得到的仍为全"0"状态。这就意味着在这种反馈移位寄存器中应避免出现全"0"状态,不然移位寄存器的状态将不会改变。因为 4 级移位寄存器共有 $2^4=16$ 种可能的不同状态,除全"0"状态外,只剩 15 种状态可用,所以由任何 4 级反馈移位寄存器产生的序列的周期最长为 15。

表 3-1　各次移位后移位寄存器的状态

序号	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	…
a_3	1	1	1	1	0	1	0	1	1	0	0	1	0	0	0	1	…
a_2	0	1	1	1	1	0	1	0	1	1	0	0	1	0	0	0	…
a_1	0	0	1	1	1	1	0	1	0	1	1	0	0	1	0	0	…
a_0	0	0	0	1	1	1	1	0	1	0	1	1	0	0	1	0	…

通常，仿真时希望用尽可能少的级数产生尽可能长的序列。由上例可见，一个 n 级反馈移位寄存器可能产生的最长周期等于(2^n-1)，因此，反馈电路如何连接才能使移位寄存器产生的序列最长，这就是本小节后面将要讨论的主题。

为了研究产生 m 序列的规律性，图 3-6 中给出了一个线性反馈移位寄存器组成的通用结构。图中每一级移位寄存器的状态用 a_i 表示，$a_i=0$ 或 1，其中 i 为整数；反馈线的连接状态用 c_i 表示，$c_i=1$ 表示此线接通(参加反馈)，$c_i=0$ 表示此线断开。可以看到，反馈线的连接状态不同，就可能改变此移位寄存器输出序列的周期 p。为了进一步研究它们之间的关系，需要用数学表达式进行描述。

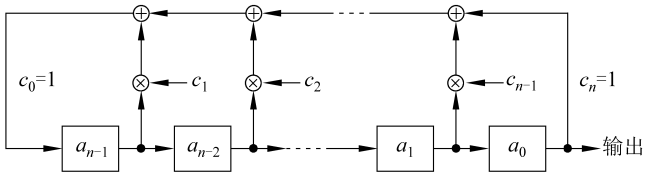

图 3-6　线性反馈移位寄存器

设 n 级移位寄存器的初始状态为$(a_{-1}\quad a_{-2}\quad \cdots\quad a_{-n})$，经过一次移位后，状态变为$(a_0\quad a_{-1}\quad \cdots\quad a_{-n+1})$；经过 n 次移位后，状态为$(a_{n-1}\quad a_{n-2}\quad \cdots\quad a_0)$。图 3-6 展示的就是线性反馈移位寄存器相应工作原理。按照图中线路连接关系，移位寄存器左端输入 a_n 可以写为

$$a_n=c_1a_{n-1}+c_2a_{n-2}+\cdots+c_{n-1}a_1+c_na_0=\sum_{i=1}^{n}c_ia_{n-i}(\text{模 }2) \tag{3-38}$$

将其推广，对于任意一状态 a_k，则有

$$a_k=\sum_{i=1}^{n}c_ia_{k-i} \tag{3-39}$$

式(3-39)中求和仍按模 2 运算。由于本章中类似方程都是按模 2 运算，故公式中不再每次注明(模 2)了。式(3-39)被称为递推方程，它给出移位输入 a_k 与移位前各级状态的关系。

前面曾经指出，c_i 的取值决定了移位寄存器的反馈连接和序列的结构，故 c_i 也是一个很重要的参量。它可以用下列方程表示：

$$f(x)=c_0+c_1x+c_2x^2+\cdots+c_nx^n=\sum_{i=0}^{n}c_ix^i \tag{3-40}$$

式(3-40)称为特征方程(或特征多项式)。式中系数 c_i 取 1 或 0；x^i 仅指明 c_i 所在的位置，其本身的取值并无实际意义。例如，若特征方程为

$$f(x)=1+x+x^4 \tag{3-41}$$

则它仅表示 x^0、x^1 和 x^4 的系数为 1($c_0=c_1=c_4=1$),其余的 c_i 为零($c_2=c_3=0$)。按这一特征方程构成的反馈移位寄存器就是图 3-6 所示的。

同样,也可以将反馈移位寄存器的输出序列$\{a_k\}$用代数方程表示,即

$$G(x)=a_0+a_1x+a_2x^2+\cdots=\sum_{k=1}^{\infty}a_kx^k \tag{3-42}$$

式(3-42)被称为母函数。

当然,还可以将式(3-39)中的系数与一个 n 次多项式关联起来,构成多项式

$$p(x)=c_0x^n+c_1x^{n-1}c_2x^{n-2}+\cdots+c_{n-1}x+c_n \tag{3-43}$$

除了上述描述线性反馈移位寄存器组成和输出的方法以外,还有许多描述方式,但是递推方程、特征方程和母函数就是关于产生或者描述 m 序列的三个基本关系式,同时也是分析移位寄存器产生 m 序列的有力工具。

可以证明,一个 n 级线性反馈移位寄存器的相继状态是具有周期性的,其周期最大值为 2^n-1。当然,要产生这个最大周期($p=2^n-1$) 序列的充要条件,就是线性反馈移位寄存器的特征多项式 $f(x)$为本原多项式。同时还可以发现,若 $f(x)$和 $p(x)$互为逆多项式,则 $p(x)$也为本原多项式。假设 $f(x)$多项式是一个 n 次本原多项式,则 $f(x)$多项式所需要满足的条件如下:

(1) $f(x)$为既约多项式,即不能分解因子的多项式。

(2) $f(x)$能够被(x^p+1)整除,其中 $p=2^n-1$。

(3) 但 $f(x)$不能被(x^q+1)整除,其中 $q<p$。

例 3-3 要求用一个 4 级反馈移位寄存器产生 m 序列,试求其特征多项式。

解:由于给定 $n=4$,故此移位寄存器产生的 m 序列的长度为 $p=2^n-1=15$。由于其特征多项式 $f(x)$应能够被$(x^p+1)=(x^{15}+1)$整除,或者说应是$(x^{15}+1)$的一个因子,故将$(x^{15}+1)$分解因式,从其因子中找 $f(x)$

$$(x^{15}+1)=(x^4+x+1)(x^4+x^3+1)(x^4+x^3+x^2+x+1)(x^2+x+1)(x+1) \tag{3-44}$$

$f(x)$不仅应为$(x^{15}+1)$的一个因子,而且还应该是一个 4 次本原多项式。式(3-44) 表明,$(x^{15}+1)$可以分解为 5 个既约因子,其中 3 个是 4 次多项式。可以证明,这 3 个 4 次多项式中,前两个是本原多项式,第 3 个不是。因为

$$(x^4+x^3+x^2+x+1)(x+1)=(x^5+1) \tag{3-45}$$

从式(3-45)可以看出,$(x^4+x^3+x^2+x+1)$不仅可以被$(x^{15}+1)$整除,而且还可以被(x^5+1)整除,故它不是本原多项式。因此,找到了两个 4 次本原多项式:(x^4+x+1)和(x^4+x^3+1)。由其中任何一个都可产生 m 序列。其中,用(x^4+x+1)作为特征多项式构成的 4 级反馈移位寄存器,如图 3-6 所示。

由上述论述可见,只要找到了本原多项式,就能由它构成 m 序列产生器。但是寻找本原多项式并不简单。经过前人大量的计算,已将常用本原多项式列成表备查,如,在表 3-2 中列出了一部分。在制作 m 序列产生器时,本原多项式的项数确定移位寄存器反馈线(及模 2 加法电路)的数目。为了使 m 序列产生器的组成尽量简单,希望使用项数最少的那些本原多项式。由表 3-2 可见,本原多项式最少有三项,这时只需用一个模 2 加法器。对于某

些 n 值，由于不存在三项的本原多项式，只好列入较长的本原多项式。

表 3-2　常用本原多项式

n	本原多项式		n	本原多项式	
	代数式	八进制表示法		代数式	八进制表示法
2	x^2+x+1	7	14	$x^{14}+x^{10}+x^6+x+1$	42103
3	x^3+x+1	13	15	$x^{15}+x+1$	100003
4	x^4+x+1	23	16	$x^{16}+x^{12}+x^3+x+1$	210013
5	x^5+x^2+1	45	17	$x^{17}+x^3+1$	400011
6	x^6+x+1	103	18	$x^{18}+x^7+1$	1000201
7	x^7+x^3+1	211	19	$x^{19}+x^5+x^2+x+1$	2000047
8	$x^8+x^4+x^3+x^2+1$	435	20	$x^{20}+x^3+1$	4000011
9	x^9+x^4+1	1021	21	$x^{21}+x^2+1$	10000005
10	$x^{10}+x^3+1$	2011	22	$x^{22}+x+1$	20000003
11	$x^{11}+x^2+1$	4005	23	$x^{23}+x^5+1$	40000041
12	$x^{12}+x^6+x^4+x+1$	10123	24	$x^{24}+x^7+x^2+x+1$	100000207
13	$x^{13}+x^4+x^3+x+1$	20033	25	$x^{25}+x^3+1$	200000011

因为本原多项式的逆多项式也是本原多项式，例如，式(3-41)中的(x^4+x+1)与(x^4+x^3+1)互为逆多项式，即 10011 与 11001 互为逆码，所以在表 3-2 中每一本原多项式可以构成两种 m 序列产生器。同时在表 3-2 中将本原多项式用八进制数字表示，简化了原多项式的表示方法。例如，对于 $n=4$ 时，表中给出“23”，它表示：

八进制数据	2	3
二进制数据	010	011
对应系数 c_i	$c_5c_4c_3$	$c_2c_1c_0$

根据上面的对应关系，由图 3-2 可见，在 $n=4$ 的线性反馈移位寄存器中，反馈连接 $c_4=c_1=c_0=1$，$c_5=c_3=c_2=0$。

2. m 序列的性质

(1) 均衡特性。

在 m 序列的一个周期中，“1”和“0”的数目基本相等。准确地说，“1”的个数比“0”的个数多一个，这种特性被称为 m 序列的均衡特性。以图 3-5 构成的 m 序列产生器为例，产生的 m 序列的周期 $p=15$，“1”的个数为 8，“0”的个数为 7。

(2) 游程分布。

在序列当中，取值相同的，相继的(连在一起的)元素合称为 1 个“游程”。这个游程中元素的个数被称为游程长度。下面分析由图 3-5 产生的 m 序列的游程情况。首先重新写出它的 m 序列输出，如下

$$\cdots\overbrace{100011110101100}^{p=15\text{个}}10\cdots$$

在上述 m 序列的 1 个周期(15 个元素)中,共有 8 个游程,其中长度为 4 的游程有 1 个,即“1111”;长度为 3 的游程有 1 个,即“000”;长度为 2 的游程有 2 个,即“11”与“00”;长度为 1 的游程有 4 个,即两个“1”与两个“0”。

一般说来,在 m 序列中,长度为 1 的游程占游程总数的 1/2;长度为 2 的游程占游程总数的 1/4;长度为 3 的占 1/8;……,以此规律向后递推。可以证明,长度为 k 的游程数目占游程总数的 2^{-k},其中 $1\leqslant k\leqslant(n-1)$;而且当 $1\leqslant k\leqslant(n-2)$时,在长度为 k 的游程中,连“1”的游程和连“0”的游程各占一半。

(3) 移位相加特性。

一个周期为 p 的 m 序列 M_p,与其任意次移位后的序列 M_r 模 2 相加,所得序列 M_s 必是 M_p 某次移位后的序列,即仍是周期为 p 的 m 序列。现在仍以图 3-5 构成的 m 序列产生器为例来说明。

假设 $M_p=100011110101100$,左移两位后得到 M_r,即 $M_r=001000111101011$,将 M_p 与 M_r 进行模 2 相加,得到 $M_s=M_p\oplus M_r$,即

$$M_p=100011110101100$$
$$M_r=001000111101011$$
$$M_s=101011001000111$$

从上面的计算结果可以看到,所得到的 M_s 相当于将 M_p 循环左移 8 位后得到的序列。

(4) 自相关函数。

连续的周期函数 $s(t)$的自相关函数可以表示为

$$R(\tau)=\frac{1}{T}\int_{-T/2}^{T/2}s(t)s(t+\tau)\mathrm{d}t \tag{3-46}$$

式中,T 是 $s(t)$的周期。

对于用“0”和“1”表示的二进制数序列,根据编码理论可以证明,其自相关函数的计算公式为

$$R(j)=\frac{A-D}{A+D}=\frac{A-D}{p} \tag{3-47}$$

式中,A 表示该序列与其 j 次移位序列在一个周期中对应元素相同的数目;D 表示该序列与其 j 次移位序列在一个周期中对应元素不同的数目;p 表示该序列的周期。

假设 M_p 移 j 位后得到 M_r,则 M_p 序列元素 x_i 与 M_r 序列元素 x_{i+j} 相对应,这时式(3-47)就可以写为

$$R(j)=\frac{[x_i\oplus x_{i+j}=0]\text{的数目}-[x_i\oplus x_{i+j}=1]\text{的数目}}{p} \tag{3-48}$$

式中,$x_i=0$ 或 1。

现在就可以利用式(3-48)来计算 m 序列的自相关函数。式(3-48)分子当中的模 2 运算相当于对 M_r 与 M_p 进行模 2 运算,由 m 序列的迟延相加特性可知,其产生的新序列还是 m 序列,所以上式分子就等于 m 序列一个周期中“0”的数目与“1”的数目之差;另外,由 m 序列的均衡性可知,在 m 序列的一个周期中“0”的数目比“1”的数目少一个,所以上式分子等于(−1)。这样式(3-48)就可以写成

$$R(j)=\frac{-1}{p},\quad j=1,2,\cdots,p-1$$

式中，p 表示 m 序列的周期。

当 $j=0$ 时，显然 $R(0)=1$。所以，m 序列的自相关函数可以表示为

$$R(j)=\begin{cases}1, & j=0\\ \dfrac{-1}{p}, & j=1,2,\cdots,p-1\end{cases} \tag{3-49}$$

不难看出，由于 m 序列具有周期性，故其自相关函数也具有周期性，其周期为 p，即

$$R(j)=R(j+kp),\quad k=1,2,3,\cdots \tag{3-50}$$

而且，$R(j)$还是偶函数，即有

$$R(j)=R(-j),\quad j=1,2,\cdots,p-1 \tag{3-51}$$

由式(3-49)m 序列自相关函数的计算结果可见，$R(j)$只可能有 2 种取值(1 和$-1/p$)，通常把这类相关函数只有两种取值的序列称为双值自相关序列。

虽然上面数字序列的相关函数 $R(j)$只在离散点上取值，但也可以按式(3-46) 计算 m 序列连续波形的自相关函数 $R(\tau)$。计算结果表明，$R(\tau)$曲线是由 $R(j)$各点连成的折线，如图 3-7 所示。

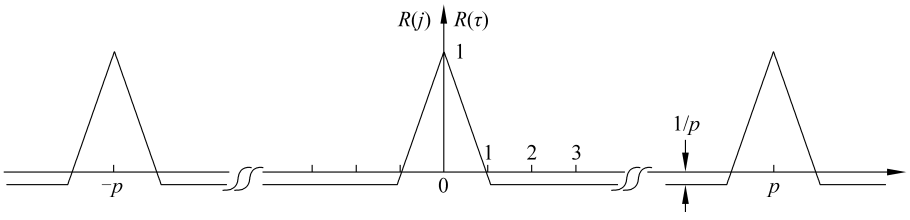

图 3-7　m 序列的自相关函数

其相应的 $R(\tau)$曲线的数学表达式为

$$R(\tau)=\begin{cases}1-\dfrac{p+1}{T}|\tau-iT|, 0\leqslant|\tau-iT|\leqslant\dfrac{T}{p}, i=0,1,2,\cdots\\ \dfrac{-1}{p}\end{cases} \tag{3-52}$$

(5) 功率谱密度。

在第 2 章已经讨论过，信号的自相关函数与功率谱密度构成一对傅里叶变换。因此，当得到 m 序列的自相关函数 $R(\tau)$以后，经过傅里叶变换，就可以求出相应的功率谱密度 $P(\omega)$，其计算结果为

$$\begin{aligned}P(\omega)&=\int_{-\infty}^{\infty}R(\tau)\mathrm{e}^{-\mathrm{j}\omega\tau}\mathrm{d}\tau\\&=\frac{p+1}{p^2}\cdot\left[Sa\left(\frac{\omega T}{2p}\right)\right]^2\cdot\sum_{\substack{n=-\infty\\ n\neq0}}^{\infty}\delta\left(\omega-\frac{2\pi n}{T}\right)+\frac{1}{p^2}\delta(\omega)\end{aligned} \tag{3-53}$$

按式(3-53)计算并画出相应的曲线，如图 3-8 所示，其中 $\omega_0=\dfrac{2\pi}{T}$。从图上可以看到，在 $T\to\infty$时，功率谱密度 $P(\omega)$的特性趋于白噪声的功率谱特性。

(6) 伪噪声特性。

对高斯白噪声进行取样，若取样值为正，则记为“+”；若取样值为负，则记为“−”。将每次取样所得极性构成一个随机序列，它具有如下基本性质：

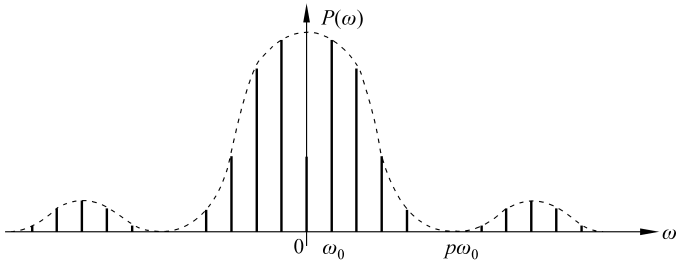

图 3-8　m 序列的功率谱密度

① 序列中“+”和“−”的出现概率几乎相等。

② 序列中长度为 1 的游程约占 1/2；长度为 2 的游程约占 1/4；长度为 3 的占 1/8；……，以此规律向后递推。一般来说，长度为 k 的游程数目占游程总数的 2^{-k}，而且在长度为 k 的游程中，连“1”的游程和连“0”的游程各占一半。

③ 由于白噪声的功率谱密度为常数，功率谱的逆傅里叶变换，即自相关函数为冲激函数 $\delta(\tau)$。当 $\tau \neq 0$ 时，$\delta(\tau)=0$；仅当 $\tau=0$ 时，$\delta(\tau)$是个面积为 1 的脉冲。

因为 m 序列的均衡性、游程分布、自相关特性和功率谱与上述随机序列的基本性质很相似，所以通常认为 m 序列属于伪噪声序列或伪随机序列。但是，具有或基本具有上述性质的序列不仅只有 m 序列一种，m 序列只是其中最常见的一种。

(7) m 序列具有 n 比特所有可能的组合。

线性移位寄存器的各个寄存器状态构成一个组合，每一种组合在一个周期内只出现一次，但 n 个 0 的组合除外，因此，最长的 0 序列为 $n-1$ 位(比特)。例如，利用图 3-5 产生的 m 序列，各个寄存器状态输出如表 3-1 所示，在一个周期中出现了 15 种状态输出组合，最长的 0 序列是 3 比特。

为了仿真数字通信系统中滤波器引入的码间串扰(ISI)，m 序列的第(7)条性质特别有用。在数字通信系统中，ISI 是使系统性能变坏的主要原因。为了模拟仿真 ISI 的影响，需要用 1 个二元序列驱动这个系统，这个序列应当具有所有可能的 n 比特组合，其中 n 是系统存储长度。

虽然 1 个二元随机序列能用作输入，但不能保证 1 个任意长度的二元随机序列绝对包含某 1 个特定比特图样。然而，用 n 级反馈移位寄存器产生 m 序列，能够产生 2^n-1 种所有 n 比特的组合图样，当然这里并不包含 n 比特全 0 组合图样。通常，数字通信系统的记忆长度小于 10 比特，因此，要模拟仿真二元系统的 ISI，采用小于 1024 比特的 1 个 PN 序列就足够了。

在其他应用场合，如果要产生二元随机序列，则需要 1 个尽可能长的 PN 序列。同时需要注意，PN 序列可以比二元随机序列更迅速地产生。当然，为了启动 PN 序列的产生，需要向移位寄存器中设置初始值，这个初始值可以选择 1～(2^n-1)间的任何二进制数。

3.3.3　M 进制伪随机序列

很多数字通信系统为了提高通信效率而采用多进制(M 进制)波形来传输信息，例如 MASK、MFSK、MPSK 以及 QAM 等系统。当然，如果要仿真这些系统的性能，就需要利用 M 进制 PN 序列。它也可以利用反馈移位寄存器来产生，是产生二进制 PN 序列方法的推

广。与二进制反馈移位寄存器的描述类似，M 进制反馈移位寄存器也可以用特征多项式来表示，具体写为

$$f(x)=c_0+c_1x+c_2x^2+\cdots+c_nx^n=\sum_{i=0}^{n}c_ix^i \tag{3-54}$$

其相应逆多项式为

$$p(x)=c_0x^n+c_1x^{n-1}c_2x^{n-2}+\cdots+c_{n-1}x+c_n \tag{3-55}$$

在上面两式中，系数 c_i 可在共有 q 个元素的有限区域内取值(q 进制)，这个有限域定义为 GF(q)。一个在 GF(q)域的 n 阶多项式，如果能被 x^p+1 整除(这里 $p=q^n-1$)，就称该多项式为本原多项式。与二进制情况类似，可以得到结论，以 GF(q)域上的本原多项式构建的 M 进制反馈移位寄存器，能够产生一个最大长度为 q^n-1 的 q 进制序列，其中 n 是该移位寄存器的阶数。

与式(3-54)或式(3-55)相对应，q 进制输出序列的递推描述关系式类似于式(3-39)，于是可以写为

$$a_k=\sum_{i=1}^{n}c_ia_{k-i} \tag{3-56}$$

式中，a_k 是时间 k 的输出。

$\{a_k\}$序列可以用线性移位寄存器产生，与式(3-39)相比，其差别在于，式(3-56)中的计算不再是模 2 运算，而是模 q 运算。

在 M 进制通信系统中，通常 $M=2^n$，即 $q=M=2^n$，这时 GF(q)中的元素可以利用系数在 GF(2)中所有次数小于 n 的多项式标识，同时在 GF(2^n)域上，减和加相同，相乘被定义为

$$C(x)=[A(x)\cdot B(x)]\bmod g(x) \tag{3-57}$$

式中，$g(x)$为 GF(2)上某一 n 次既约多项式。

例 3-4　计算多项式 $A(x)=x$ 和 $B(x)=x^2+x$ 在 GF(2^3)域上相乘的结果 $C(x)$，其中，GF(2)上某一 3 次既约多项式为 $g(x)=x^3+x+1$。

解：$C(x)=[A(x)\cdot B(x)]\bmod g(x)=[x\cdot(x^2+x)]\bmod(x^3+x+1)$

$$=x^2+x+1$$

式(3-54)描述了 M 进制反馈移位寄存器的特征多项式，式(3-56) 描述了 M 进制输出序列的递推关系，它们都可以对 GF(2^n)域上产生的 PN 序列进行表示。除此之外，还有第 3 种表示方法，这种方法是基于这样一个事实，即在每个有限域里都存在一个元素 α，称为本原值；在这个有限域中，每个非零元素可以用 α 的某次幂来表示。具体情况如表 3-3 所示，在表中用既约多项式 $g(x)=x^3+x+1$ 来产生 GF(2^3)域的元素，表的最左边一栏表示 GF(2^3)域中的缩写符号。

表 3-3　GF(2^3) 域中元素的表示方法

缩写符号	多项式根的幂	多项式表示	二元数组表示
A	α	x	(0,1,0)
B	α^2	x^2	(1,0,0)
C	α^3	$x+1$	(0,1,1)

续表

缩写符号	多项式根的幂	多项式表示	二元数组表示
D	α^4	x^2+x	(1,1,0)
E	α^5	x^2+x+1	(1,1,1)
F	α^6	x^2+1	(1,0,1)
1	α^7	1	(0,0,1)
0	0	0	(0,0,0)

为了生成一个最长的八进制 PN 序列，需要一个系数属于 GF(8)的本原多项式，这个多项式将描述式(3-56)中的系数与一个 n 次多项式的关系。如果采用上面定义的 GF(8)的表示法，则这个多项式是

$$p_1(x)=x^3+x+\alpha \sim (101\mathrm{A}) \tag{3-58}$$

由式(3-58)表示的多项式，可以得到对应的递推关系式

$$a_k=a_{k-1}+\alpha a_{k-3} \tag{3-59}$$

根据式(3-58)就可以构建出这个八进制反馈移位寄存器电路，如图 3-5(a)所示；同时，根据该电路还可以产生八进制输出样本序列。如果寄存器的始值为 010，则输出序列为 0101A10F。因为 $f(x)$是 3 次多项式，所以它适合于有 3 个符号记忆的信道。

如果要仿真两正交分量八进制 PN 序列，需要两个八进制 PN 序列产生器，则此时需要第二个本原多项式。这样的三次多项式是

$$p_2(x)=x^3+x+\alpha^2 \sim (101\mathrm{B}) \tag{3-60}$$

对应的递推关系式为

$$a_k=a_{k-1}+\alpha a_{k-3} \tag{3-61}$$

根据式(3-61)构建的八进制反馈移位寄存器电路，如图 3-9(b)所示。

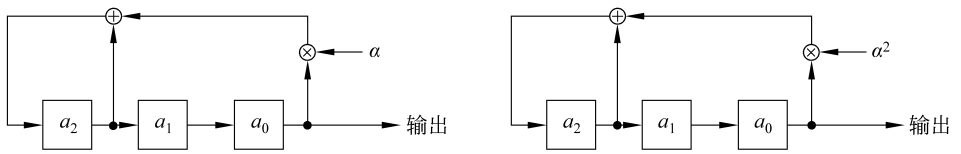

图 3-9　八进制 PN 序列产生器

如果要仿真 16-QAM 系统，可以采用上述类似的方法，在 GF(4)中选择系数，确定本原多项式，构建两个八进制反馈移位寄存器电路，来表示 I 和 Q 信道。感兴趣的读者可以参阅相关文献。

3.4　相关随机序列的产生

在许多实际应用中，经常需要产生具有某种特定自相关函数，或者具有某种特定功率谱密度形式的输出序列，这类处理过程被统称为相关随机序列的产生。相关随机序列在通信建模仿真系统中有许多应用，例如，当对随机时变的移动通信信道建模仿真时，就需要产生