

## 第七章 随机信号的参数建模法

### (Modern modeling method for signal)

为随机信号建立参数模型是研究随机信号的一种基本方法，其含义是认为随机信号  $x(n)$  是由白噪  $w(n)$  激励某一确定系统的响应（如图 7.5）。只要白噪的参数确定了，研究随机信号就可以转化成研究产生随机信号的系统。

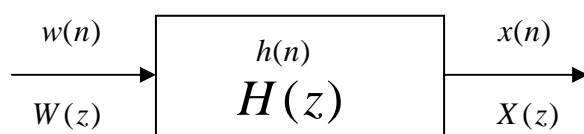


图 7.5 随机信号的参数模型

经典信号建模法（classical modeling method for signal）前面已经指出，医学信号处理的目的是提取包含于随机信号中的确定性成分，以便在一定的准确性（最小二乘意义）上进行预测。这就是建立各种各样的确定性数学模型，包括代数、微分、积分、差分方程模型。这是经典的信号建模方法。

信号的现代建模方法（Modern modeling method for signal）是建立在具有最大的不确定性基础上的预测。提出了众多的数学模型（mathematical models）。根据 Wold 的证明：任何平稳的 ARMA（自回归移动平均）模型或 MA 模型均可用无限阶或阶数足够的 AR 模型去近似。因此本节着重介绍 AR 模型的基本原理和方法。

对平稳随机信号，三种常用的线性模型分别是 AR 模型（自回归模型 Auto-regression model），MA 模型（滑动平均模型 Moving average model）和 ARMA 模型（自回归滑动平均模型 Auto-regression-Moving average model）。

## 第一节 三种参数模型

### 7.1.1 MA 模型

随机信号  $x(n)$  由当前的激励  $w(n)$  和若干次过去的激励  $w(n-k)$  线性组合产生：

$$x(n) = \sum_{k=0}^q b_k w(n-k) \quad (7-22)$$

该模型的系统函数是：
$$H(z) = \frac{X(z)}{W(z)} = \sum_{k=0}^q b_k z^{-k} \quad (7-23)$$

$q$  表示系统阶数，系统函数只有零点，没有极点，所以该系统一定是稳定的系统，也称为全零点模型，用 MA (  $q$  ) 来表示。

### 7.1.2 AR 模型

随机信号  $x(n)$  由本身的若干次过去值  $x(n-k)$  和当前的激励值  $w(n)$  线性组合产生：

$$x(n) = w(n) - \sum_{k=1}^p a_k x(n-k) \quad (7-24)$$

该模型的系统函数是：
$$H(z) = \frac{1}{1 + \sum_{k=1}^p a_k z^{-k}} \quad (7-24)$$

$p$  是系统阶数，系统函数中只有极点，无零点，也称为全极点模型，系统由于极点的原因，要考虑到系统的稳定性，因而要注意极点的分布位置，用 AR (  $p$  ) 来表示。

### 7.1.3 ARMA 模型

ARMA 是 AR 与 MA 模型的结合：

$$x(n) = \sum_{k=0}^q b_k w(n-k) - \sum_{k=1}^p a_k x(n-k) \quad (7-25)$$

该模型的系统函数是：
$$H(z) = \frac{\sum_{k=0}^q b_k z^{-k}}{1 + \sum_{k=1}^p a_k z^{-k}} \quad (7-26)$$

它既有零点又有极点，所以也称极零点模型，要考虑极零点的分布位置，保证系统的稳定，用 ARMA (  $p, q$  ) 表示。

在随机信号时域分析中，提出了许多数学模型用来由已知在最大不确定原则下预测将来值，其优点是只需要很少的已知值。但是它不能用在信号是确定性的场合，在确定信号的情况下，信号是由确定的数学方程预测的。这点要特别注意。例如，如果我们用心电的 R 波升枝作已知数据进行随机预测，则预测值即为与 R 波上升枝有关的数据，决不可能预测出整个 P-QRS-T 复合波来。

## 第二节 AR 模型参数的估计

随机信号的建模法最近在生物医学信号处理中应用相当普遍，在自发脑电、诱发脑电、肌电、心电、胃电等方面都有人尝试应用模型法进行研究。应用较多的是 AR 模型，因为建立这种的模型计算工作比较容易。作为数学逼近，三种模型都可以互相转换。实际中选用哪一种模型就要考虑到节约和计算量，选定模型后，剩下的任务就是用适当的算法估计模型参数（ $a_k$ 、 $b_k$ 、 $p$ 、 $q$ ），以便使用模型对随机信号进行预测。

下面我们就以 AR 模型为例进行参数的估计。

### 1. AR 模型参数和自相关函数的关系

根据图 7.5 和式(7-24):  $x(n) = w(n) - \sum_{k=1}^p a_k x(n-k)$ ，对该式两边同时乘以  $x(n-m)$ ，

然后求均值：

$$E[x(n)x(n-m)] = E[w(n)x(n-m) - \sum_{k=1}^p a_k x(n-k)x(n-m)] \quad (7-27)$$

因为自相关函数：

$$R_{xx}(m) = E[x(n)x(n+m)] = E[x(n-m)x(n)] = E[x(n)x(n-m)] = R_{xx}(-m)$$

所以自相关函数呈现偶对称，(7-27)式化为：

$$R_{xx}(m) = R_{xw}(m) - \sum_{k=1}^p a_k R_{xx}(m-k) \quad (7-28)$$

系统的单位脉冲响应  $h(n)$  是因果的，所以输出的平稳随机信号和输入的白噪声之间的互相关函数有下列推导：

$$R_{xw}(m) = E[x(n)w(n+m)] \quad (7-29)$$

$$x(n) = w(n) * h(n) = \sum_{k=0}^{\infty} h(k)w(n-k) \quad (7-30)$$

$$\begin{aligned} R_{xw}(m) &= E\left[\sum_{k=0}^{\infty} h(k)w(n-k)w(n+m)\right] = \sum_{k=0}^{\infty} h(k)E[w(n-k)w(n+m)] \\ &= \sum_{k=0}^{\infty} h(k)R_{ww}(m+k) = \sum_{k=0}^{\infty} h(k)\sigma_w^2\delta(m+k) = \sigma_w^2 h(-m) \end{aligned} \quad (7-31)$$

所以

$$R_{xw}(m) = \begin{cases} 0 & m > 0 \\ \sigma_w^2 h(-m) & m \leq 0 \end{cases} \quad (7-32)$$

带入式(7-28)得到:

$$R_{xx}(m) = \begin{cases} -\sum_{k=1}^p a_k R_{xx}(m-k) & m > 0 \\ -\sum_{k=1}^p a_k R_{xx}(m-k) + h(0)\sigma_w^2 & m = 0 \\ R_{xx}(-m) & m < 0 \end{cases} \quad (7-33)$$

由于  $H(z) = \frac{1}{1 + \sum_{k=1}^p a_k z^{-k}}$ , 转换到时域得到:  $h(n) + \sum_{k=1}^p a_k h(n-k) = \delta(n)$ , 因而  $h(0) = 1$

显然, AR 模型输出信号的自相关函数具有递推的性质, 即:

$$R_{xx}(m) = -\sum_{k=1}^p a_k R_{xx}(m-k) \quad m > 0 \quad (7-34)$$

上式就是著名的 Yule-Walker (Y-W) 方程, 将上式变换:

$$0 = R_{xx}(m) + \sum_{k=1}^p a_k R_{xx}(m-k) \quad m > 0 \quad (7-35)$$

从 (7-33) 求得输入白噪声方差为:

$$\sigma_w^2 = R_{xx}(0) + \sum_{k=1}^p a_k R_{xx}(-k) \quad m = 0 \quad (7-36)$$

将(7-35)和(7-36)结合, 把该式的下标简化并写成矩阵的形式, 可以写成单一的正规矩阵方程:

$$\begin{bmatrix} R(0) & R(-1) & \cdots & R(-p) \\ R(1) & R(0) & \cdots & R(-p+1) \\ \vdots & \vdots & \cdots & \vdots \\ R(p) & R(p-1) & \cdots & R(0) \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} \sigma_w^2 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (7-37)$$

式 (7-37) 方程组的系数都是自相关矩阵  $[R]_{p+1}$ , 由于自相关函数是偶对称函数:

$R_{xx}(m) = R_{xx}(-m)$ , 因而自相关矩阵是对称矩阵, 与主对角线平行的斜对角线上的元素都是相同的, 是  $(p+1) \times (p+1)$  的维托毕兹 (Toeplitz) 矩阵, 所以存在高效算法, 其中应用

广泛的有 Levinson-Durbin (L-D) 算法。Yule-Walker (Y-W) 方程表明, 只要已知输出平稳随机信号的自相关函数, 就能求出 AR 模型中的参数  $\{a_k\}$ , 并且需要的观测数据较少。

**【例 7-2】** 已知自回归信号模型 AR (3) 为:

$$x(n) = \frac{14}{24}x(n-1) + \frac{9}{24}x(n-2) - \frac{1}{24}x(n-3) + w(n)$$

式中  $w(n)$  是具有方差  $\sigma_w^2=1$  的平稳白噪声，求

- 自相关序列  $R_{xx}(m)$ ， $m=0, 1, 2, 3, 4, 5$ 。
- 用 a 求出的自相关序列来估计 AR (3) 的参数  $\{\hat{a}_k\}$ ，以及输入白噪声的方差  $\hat{\sigma}_w^2$  大小。
- 利用给出的 AR 模型，用计算机仿真给出 32 点观测值  $x(n) = [0.4282 \quad 1.1454$   
 $1.5597 \quad 1.8994 \quad 1.6854 \quad 2.3075 \quad 2.4679 \quad 1.9790 \quad 1.6063 \quad 1.2804$   
 $-0.2083 \quad 0.0577 \quad 0.0206 \quad 0.3572 \quad 1.6572 \quad 0.7488 \quad 1.6666 \quad 1.9830$   
 $2.6914 \quad 1.2521 \quad 1.8691 \quad 1.6855 \quad 0.6242 \quad 0.1763 \quad 1.3490 \quad 0.6955$   
 $1.2941 \quad 1.0475 \quad 0.4319 \quad 0.0312 \quad 0.5802 \quad -0.6177]$ ，用观测值的自相关序列直接来估计 AR(3)的参数  $\{\hat{a}_k\}$  以及输入白噪声的  $\hat{\sigma}_w^2$ 。

解：a. 已知的是模型参数  $\{a_k\}$ ， $a_1 = -14/24$ ， $a_2 = -9/24$ ， $a_3 = 1/24$ ，来自相关

序列  $R_{xx}(m)$ 。利用式 (7-37) 
$$\begin{bmatrix} R(0) & R(-1) & \cdots & R(-p) \\ R(1) & R(0) & \cdots & R(-p+1) \\ \vdots & \vdots & \cdots & \vdots \\ R(p) & R(p-1) & \cdots & R(0) \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} \sigma_w^2 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$
，把  $\{a_k\}$  代

入，利用自相关函数的偶对称，得到一个  $4 \times 4$  的的矩阵：

$$\begin{bmatrix} R(0) & R(1) & R(2) & R(3) \\ R(1) & R(0) & R(1) & R(2) \\ R(2) & R(1) & R(0) & R(1) \\ R(3) & R(2) & R(1) & R(0) \end{bmatrix} \begin{bmatrix} 1 \\ -14/24 \\ -9/24 \\ 1/24 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

解线性方程组得：

$$R(0) = 4.9377 \quad R(1) = 4.3287 \quad R(2) = 4.1964 \quad R(3) = 3.8654$$

利用式 (7-34)  $R_{xx}(m) = -\sum_{k=1}^p a_k R_{xx}(m-k) \quad m > 0$ ，可以求出  $R(4)$ ， $R(5) \cdots$

$$R_{xx}(4) = -\sum_{k=1}^3 a_k R_{xx}(4-k) = 3.6481, \quad R_{xx}(5) = -\sum_{k=1}^3 a_k R_{xx}(5-k) = 3.4027$$

当然还可以求出无穷多的自相关序列值。

b. 已知自相关序列值，来估计 3 阶 AR 模型的参数  $\{\hat{a}_k\}$  以及  $\sigma_w^2$

利用式 (7-37) 得到矩阵：

$$\begin{bmatrix} R(0) & R(1) & R(2) & R(3) \\ R(1) & R(0) & R(1) & R(2) \\ R(2) & R(1) & R(0) & R(1) \\ R(3) & R(2) & R(1) & R(0) \end{bmatrix} \begin{bmatrix} 1 \\ \hat{a}_1 \\ \hat{a}_2 \\ \hat{a}_3 \end{bmatrix} = \begin{bmatrix} \hat{\sigma}_w^2 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (7-38)$$

解线性方程组得到：

$$\hat{a}_1 = -14/24, \quad \hat{a}_2 = -9/24, \quad \hat{a}_3 = 1/24, \quad \hat{\sigma}_w^2 = 1$$

可以发现对 AR 模型参数是无失真的估计，因为已知 AR 模型，我们可以得到完全的输出观测值，因而求得的自相关函数没有失真，当然也就可以不失真的估计。

c. 利用给出的 32 点观测值，先求自相关序列（按照上节的样本自相关定义

$$R_{xx}(m) = \frac{1}{n} \sum_{i=1}^n x_i x_{i+m} \text{ 计算），由于偶对称，只给出 } m=0, 1, 2 \cdots 31 \text{ 的 } R_{xx}(m) = [1.9271$$

$$\begin{matrix} 1.6618 & 1.5381 & 1.3545 & 1.1349 & 0.9060 & 0.8673 & 0.7520 & 0.7637 \\ 0.8058 & 0.8497 & 0.8761 & 0.9608 & 0.8859 & 0.7868 & 0.7445 & 0.6830 \\ 0.5808 & 0.5622 & 0.5134 & 0.4301 & 0.3998 & 0.3050 & 0.2550 & 0.1997 \\ 0.1282 & 0.0637 & 0.0329 & -0.0015 & -0.0089 & -0.0143 & -0.0083 \end{matrix}$$

把头 4 个相关序列值代入矩阵（7-38）求得估计值：

$$\hat{a}_1 = -0.6984 \quad \hat{a}_2 = -0.2748 \quad \hat{a}_3 = 0.0915, \quad \hat{\sigma}_w^2 = 0.4678$$

与真实 AR 模型参数误差为： $e_1 = 0.1151$ ， $e_2 = 0.1002$ ， $e_3 = 0.0498$ ，原因在于我们只有一

部分的观测数据，使得自相关序列值与理想的完全不同。输入信号的方差误差比较大： $e_\sigma =$

0.5322，造成的原因比较多，计算机仿真的白噪声由于只有 32 点长，32 点序列的方差不可

能刚好等于 1。给出一段观测值求 AR 模型参数这样直接解方程组，当阶数越高时直接解方

程组计算就越复杂，因而要用特殊的算法使得计算量减小且精确度高。

## 2. Y-W 方程的解法——L-D 算法

在求解【例 7-2】时要得到更精确的估计值，就要建立更高阶的 AR 模型，直接用观测值的自相关序列来求解 Y-W 方程计算量太大。因此把 AR 模型和预测系统联系起来，换个方法来估计 AR 参数。

从 AR 模型的时域表达式： $x(n) = w(n) - \sum_{k=1}^p a_k x(n-k)$ ，我们知道模型的当前输出值

与它过去的输出值有关。预测是推断一个给定序列的未来值，即利用信号前后的相关性来估计未来的信号值。

若序列的模型已知而用过去观测的数据来推求现在和将来的数据称为前向预测器，表示为：

$$\hat{x}(n) = -\sum_{k=1}^m a_m(k)x(n-k) \quad (7-39)$$

式中  $\{a(k)\}$ ,  $k=1, 2, \dots, m$ , 代表  $m$  阶预测器的预测系数, 负号是为了与技术文献保持一致。显然预测出来的结果与真实的结果存在预测误差或前向预测误差, 设误差为  $e(n)$ :

$$e(n) = x(n) - \hat{x}(n) = x(n) + \sum_{k=1}^m a_m(k)x(n-k) \quad (7-40)$$

把  $e(n)$  看成是系统的输出,  $x(n)$  看成是系统的输入, 得到系统函数:

$$\frac{E(z)}{X(z)} = 1 + \sum_{k=1}^m a_m(k)z^{-k} \quad (7-41)$$

假如  $m=p$ , 且预测系数和 AR 模型参数相同, 比较式(7-41)和式(7-24), 把预测误差系统框图和 AR 模型框图给出, 如图 7.6 所示, 即有  $w(n) = e(n)$ , 即前向预测误差系统中的输入为  $x(n)$ , 输出为预测误差  $e(n)$  等于白噪声。也就是说前向预测误差系统对观测信号起了白化的作用。由于 AR 模型和前向预测误差系统有着密切的关系, 两者的系统函数互为倒数, 所以求 AR 模型参数就可以通过求预测误差系统的预测系数来实现。

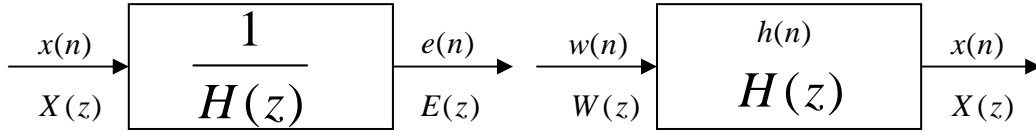


图 7.6 预测误差系统和 AR 模型

对(7-40)式求预测误差均方值:

$$\begin{aligned} E[e^2(n)] &= E[(x(n) + \sum_{k=1}^m a_m(k)x(n-k))^2] \\ &= R_{xx}(0) + 2[\sum_{k=1}^m a_m(k)R_{xx}(k)] + \sum_{k=1}^m \sum_{l=1}^m a_m(l)a_m(k)R_{xx}(l-k) \end{aligned} \quad (7-42)$$

要使得均方误差最小, 将上式右边对预测系数求偏导并且等于零, 得到  $m$  个等式:

$$R_{xx}(l) = -\sum_{k=1}^m a_m(k)R_{xx}(l-k) \quad l=1, 2, \dots, m \quad (7-43)$$

将(7-43)代入(7-42)求得最小均方误差:

$$E_m[e^2(n)] = R_{xx}(0) + \sum_{k=1}^m a_m(k)R_{xx}(k) \quad (7-44)$$

或

$$E_p[e^2(n)] = R_{xx}(0) + \sum_{k=1}^p a_k R_{xx}(k) \quad (7-45)$$

$$a_k = a_m(k) \quad m = p \quad (7-46)$$

也就是  $p$  阶预测器的预测系数等于  $p$  阶 AR 模型的参数，由于  $e(n) = w(n)$ ，所以最小均方预测误差等于白噪声方差，即  $E_p[e^2(n)] = \sigma_w^2$ 。

有了上面的知识后，我们回来看怎样估计 AR 模型参数，也即要估计参数  $\{a_k, p, \sigma_w^2\}$ ，这里介绍应用广泛的 L-D 算法。L-D 算法的基本思想就是根据 Y-W 方程式或式(7-43)、(7-44)、(7-45)，自相关序列具有递推的性质，L-D 递推算法是模型阶数逐渐加大的一种算法，先计算阶次  $m=1$  时的预测系数  $\{a_m(k)\} = a_1(1)$  和  $\sigma_{w1}^2$ ，然后计算  $m=2$  时的  $\{a_m(k)\} = a_2(1)$ ， $a_2(2)$  以及  $\sigma_{w2}^2$ ，一直计算到  $m=p$  阶时的  $a_p(1)$ ， $a_p(2)$ ， $\dots$ ， $a_p(p)$  以及  $\sigma_{wp}^2$ 。这种递推算法的特点是，每一阶次参数的计算是从低一阶次的模型参数推算出来的，既可减少工作量又便于寻找最佳的阶数值，满足精度时就停止递推。

例如按照式(7-43)、(7-45)，取  $m=1$ ，代入，简化下标，则有：

$$\begin{cases} R(1) = -a_1(1)R(0) & (1) \\ E_1 = R(0) + a_1(1)R(1) & (2) \end{cases} \implies \sigma_{w1}^2 = E_1 = R(0)[1 - a_1^2(1)] \quad (7-47)$$

$m=2$ :

$$\begin{cases} R(1) = -a_2(1)R(0) - a_2(2)R(1) \\ R(2) = -a_2(1)R(1) - a_2(2)R(0) \end{cases}$$

把 (7-47) 的 (1) 代入上式得到：

$$\begin{cases} a_2(1) = \frac{-R(1) - a_2(2)R(1)}{R(0)} = a_1(1) + a_2(2)a_1(1) & (1) \\ a_2(2) = -\frac{R(2) + a_2(1)R(1)}{R(0)} = -\frac{R(2) + [a_1(1) + a_2(2)a_1(1)]R(1)}{R(0)} \end{cases}$$

$$\implies a_2(2) = -\frac{R(2) + a_1(1)R(1)}{R(0) + a_1(1)R(1)} = -\frac{R(2) + a_1(1)R(1)}{E_1} \quad (2) \quad (7-48)$$

根据(7-45)，估计的方差为：

$$\begin{aligned} E_2 &= R(0) + a_2(1)R(1) + a_2(2)R(2) \\ &= R(0) + a_2(1)R(1) + a_2(2)[-a_2(1)R(1) - a_2(2)R(0)] \end{aligned}$$



把(7-48)的 (1) 代入:  $= R(0) + a_2(1)[1 - a_2(2)]R(1) - a_2^2(2)R(0)$

$$= R(0) + a_1(1)[1 + a_2(2)][1 - a_2(2)]R(1) - a_2^2(2)R(0)$$

$$= [1 - a_2^2(2)][R(0) + a_1(1)R(1)] = [1 - a_2^2(2)]E_1 \quad (7-49)$$

这样递推下去可以得到预测系数和均方误差估计的通式:

$$\left\{ \begin{array}{l} a_m(k) = a_{m-1}(k) + a_m(m)a_{m-1}(m-k) \quad (1) \\ a_m(m) = -\frac{R(m) + \sum_{k=1}^{m-1} a_{m-1}(k)R(m-k)}{E_{m-1}} \quad (2) \\ E_m = \sigma_{wm}^2 = [1 - a_m^2(m)]E_{m-1} = R(0) \prod_{k=1}^m [1 - a_k^2(k)] \quad (3) \end{array} \right. \quad (7-50)$$

其中  $a_m(m)$  称为反射系数, 从上式知道整个迭代过程需要已知自相关函数, 给定初始值

$E_0 = R(0)$ ,  $a_0(0) = 1$ , 以及 AR 模型的阶数  $p$ , 就可以按照图 7.7 所示流程图进行估计。

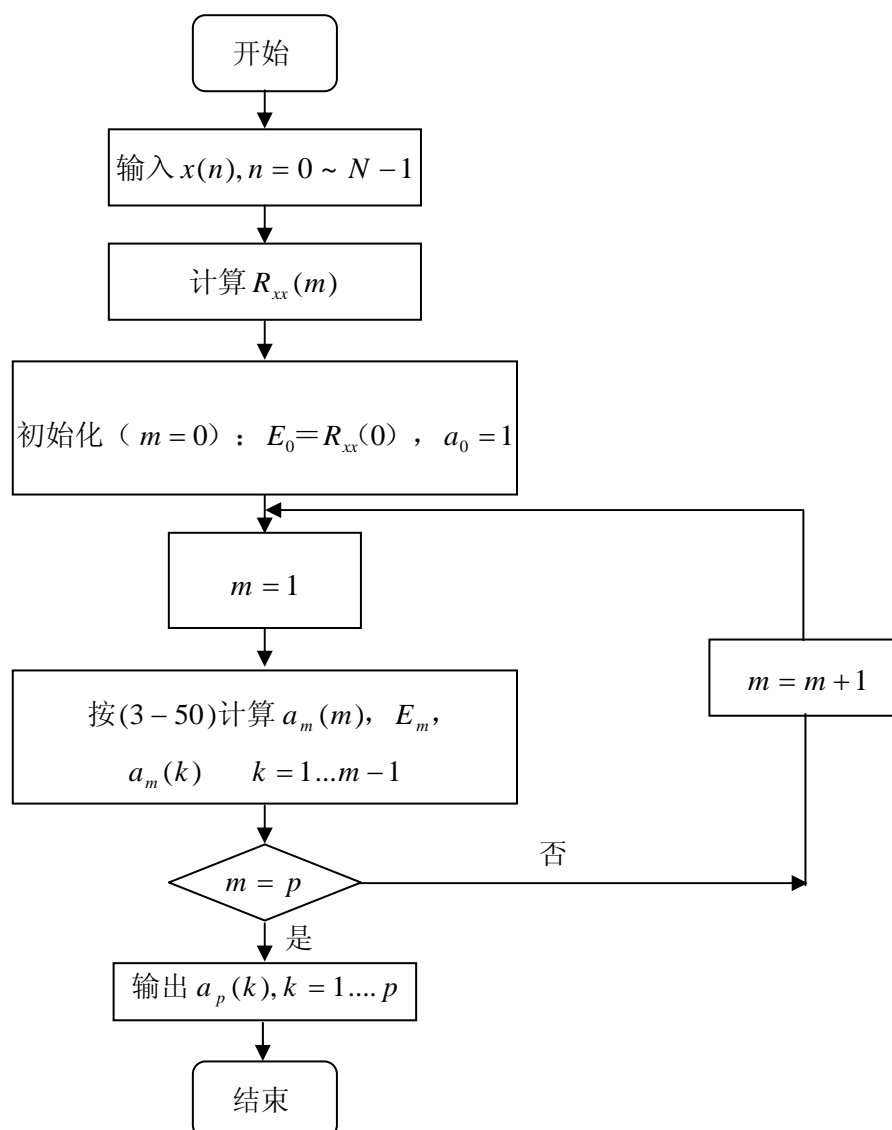


图 7.7 L-D 算法流程图

L-D 算法的优点就是计算速度快，求得的 AR 模型必定稳定，且均方预测误差随着阶次的增加而减小（见(7-50)的(3)式）。L-D 算法的缺点，由于在求自相关序列时，是假设除了观测值之外的数据都为零，必然会引入较大误差。

【例 7-3】已知自回归信号模型 AR（3）为：

$$x(n) = \frac{14}{24}x(n-1) + \frac{9}{24}x(n-2) - \frac{1}{24}x(n-3) + w(n)$$

式中  $w(n)$  是具有方差  $\sigma_w^2=1$  的平稳白噪声，利用给出的 AR 模型，用计算机仿真给出 32 点

观测值  $x(n) = [0.4282 \quad 1.1454 \quad 1.5597 \quad 1.8994 \quad 1.6854 \quad 2.3075 \quad 2.4679$   
 $1.9790 \quad 1.6063 \quad 1.2804 \quad -0.2083 \quad 0.0577 \quad 0.0206 \quad 0.3572 \quad 1.6572$   
 $0.7488 \quad 1.6666 \quad 1.9830 \quad 2.6914 \quad 1.2521 \quad 1.8691 \quad 1.6855 \quad 0.6242$   
 $0.1763 \quad 1.3490 \quad 0.6955 \quad 1.2941 \quad 1.0475 \quad 0.4319 \quad 0.0312 \quad 0.5802$

$-0.6177]$ ，用 L-D 算法来估计 AR(3)的参数  $\{\hat{a}_k\}$  以及输入白噪声的方差  $\hat{\sigma}_w^2$ 。

解：步骤 1

利用给出的 32 点观测值，先求自相关序列（如何计算自相关函数将在第六章详细介绍），

由于偶对称，只给出  $m=0, 1, 2\cdots 31$  的  $R_{xx}(m) = [1.9271 \quad 1.6618 \quad 1.5381 \quad 1.3545$   
 $1.1349 \quad 0.9060 \quad 0.8673 \quad 0.7520 \quad 0.7637 \quad 0.8058 \quad 0.8497 \quad 0.8761$   
 $0.9608 \quad 0.8859 \quad 0.7868 \quad 0.7445 \quad 0.6830 \quad 0.5808 \quad 0.5622 \quad 0.5134$   
 $0.4301 \quad 0.3998 \quad 0.3050 \quad 0.2550 \quad 0.1997 \quad 0.1282 \quad 0.0637 \quad 0.0329$   
 $-0.0015 \quad -0.0089 \quad -0.0143 \quad -0.0083]$

步骤 2

初始化：  $E_0 = R_{xx}(0) = 1.9271$ ，  $a_0 = 1$

步骤 3

根据式（7-50）计算，

$$m=1: \begin{cases} a_1(1) = -\frac{R(1)}{E_0} = -\frac{1.6618}{1.9271} = -0.8623 \\ E_1 = R(0)[1 - a_1^2(1)] = 0.4942 \end{cases}$$

$$m=2: \begin{cases} a_2(2) = -\frac{R(2) + a_1(1)R(1)}{E_1} = -0.2127 \\ a_2(1) = a_1(1)[1 + a_2(2)] = -0.6789 \\ E_2 = E_1[1 - a_2^2(2)] = 0.4718 \end{cases}$$

$$m=3: \begin{cases} a_3(3) = -\frac{R(3) + a_2(1)R(2) + a_2(2)R(1)}{E_2} = 0.0914 \\ a_3(1) = a_2(1) + a_3(3)a_2(2) = -0.6983 \\ a_3(2) = a_2(2) + a_3(3)a_2(1) = -0.2748 \\ E_3 = E_2[1 - a_3^2(3)] = 0.4679 \end{cases}$$

因而当  $p = 3$  时，估计到的 AR 模型参数为： $\hat{a}_1 = -0.6983$ ， $\hat{a}_2 = -0.2748$ ， $\hat{a}_3 = 0.0914$ ，

估计的输入信号的方差为： $\hat{\sigma}_w^2 = E_3 = 0.4679$ 。和例题 7-2 的 c 结果一致，误差分析也一样。当要计算的阶数比较高时，可以利用递推程序来实现。

Matlab 有专门的函数实现 L-D 算法的 AR 模型参数估计： $[a \ E] = \text{aryule}(x, p)$ ，输入  $x$  表示观测信号，输入  $p$  表示要求的阶数，输出  $a$  表示估计的模型参数， $e$  表示噪声信号的方差估计。例如本题用该函数计算结果为：

```
[a E]=aryule(x,3)
a=1.0000    -0.6984    -0.2748     0.0915
E=0.4678
```

这里  $a$  的第一个值等于 1，指的是  $a_0$ ，依次是  $\hat{a}_1$ ， $\hat{a}_2$ ， $\hat{a}_3$ 。

假如我们用更高阶的 AR 模型来估计：

```
[a E]=aryule(x,12)
a =1.0000    -0.6703    -0.3254    -0.0793     0.1407     0.3676    -0.2451     0.0483
   -0.0912    -0.0522     0.0515     0.0186    -0.0955
E=0.3783——阶数越高均方误差越来越小。
```

给定观测序列后我们很容易用 L-D 算法来进行 AR 模型参数的估计，但是如何确定阶数  $p$ ？以及 L-D 算法存在的缺点如何克服？下面进行简单的介绍。

### 3. AR 模型参数估计的各种算法的比较和阶数的选择

为了克服 L-D 算法导致的误差，1968 年 Burg 提出了 Burg 算法，其基本思想是对观测的数据进行前向和后向预测，然后让两者的均方误差之和为最小作为估计的准则估计处反射系数，进而通过 L-D 算法的递推公式求出 AR 模型参数。Burg 算法的优点是，求得的 AR 模型是稳定的，较高的计算效率，但递推还是用的 L-D 算法，因此仍然存在明显的缺点。

Matlab 中有专门的函数实现 Burg 算法的 AR 模型参数估计： $[a \ E] = \text{arburg}(x, p)$ 。例如上面的例 7-3 用 Burg 算法计算结果为：

[a E]=arburg(x,3)

a =1.0000    -0.6982    -0.2626    0.0739——结果与 L-D 算法结果略有不同， $\hat{a}_3$  估计得更精确些。

E =0.4567

高阶模型也是一样计算：

[a E]=arburg(x,12)

a =1.0000    -0.6495    -0.3066    -0.0934    0.0987    0.4076    -0.1786    -0.0126  
-0.0805    -0.0899    0.0382    0.1628    -0.2501

E =0.3237

1980 年 Marple 在前人的基础上提出一种高效算法，Marple 算法或者称不受约束的最小二乘法 (LS)，该算法的思想是，让每一个预测系数的确定直接与前向、后向预测的总的平方误差最小，这样预测系数就不能由低一阶的系数递推确定了，所以不能用 L-D 算法求解，实践表明该算法比 L-D、Burg 算法优越。由于该算法是从整体上选择所有的模型参数达到总的均方误差最小，与自适应算法类似，不足是该算法不能保证 AR 模型的稳定性。

AR 模型的阶数选择不同得到的模型不同，效果相差较大，因而如何选择阶数很重要。因此国内外学者在这方面都做了许多研究工作，其中基于均方误差最小的最终预测误差 (FPE: final predidyion error) 准则是确定 AR 模型阶次比较有效的准则。

最终预测误差 (FPE: final predidyion error) 准则定义：给定观测长度为 N，从某个过程的一次观测数据中估计到了预测系数，然后用该预测系数构成的系统处理另一次观察数据，则有预测均方误差，该误差在某个阶数  $p$  时为最小，其表达式为：

$$FPE(p) = \hat{\sigma}_{wp}^2 \left( \frac{N + P + 1}{N - P - 1} \right) \quad (7-52)$$

上式中估计的方差随着阶数的增加而减小，而括号内的值随着  $p$  的增加而增加，因而能找到一最佳的  $p_{opt}$ ，使得 FPE 最小。

【例 7-4】以例 7-3 的数据为例，用 FPE(p)最小的方法寻找 L-D 算法的最佳阶数  $p_{opt}$ 。

解：第一步：利用 filter 函数得到两段 32 点长的仿真序列  $x_1(n)$  和  $x_2(n)$ ；假设仿真出的数据分别为：  
 $x_1(n) = [-0.98976 \quad 0.76223 \quad 0.36297 \quad 2.0177 \quad 2.4194 \quad 1.4687 \quad 0.388$   
 $0.60336 \quad 0.10567 \quad -0.57189 \quad 0.17865 \quad 1.3738 \quad 0.34575 \quad -0.13732 \quad -0.25403 \quad 0.44894$   
 $-0.68185 \quad -1.4201 \quad -1.2227 \quad -1.2827 \quad -0.66225 \quad -1.4119 \quad -1.1682 \quad -1.618 \quad -1.4024$   
 $0.15899 \quad -0.97222 \quad -1.7964 \quad -0.94975 \quad -2.0907 \quad -1.465 \quad -2.2266]$ ， $x_2(n) = [0.5354$   
 $0.8652 \quad 0.50178 \quad -1.4595 \quad -0.56668 \quad 0.69417 \quad 1.2716 \quad -0.55468 \quad 0.045716 \quad -0.91598$

-1.5186 -2.4656 -1.6808 -2.2711 -1.7966 -2.1975 -2.3259 -1.735 -1.0645 0.93749  
-1.1373 -1.2901 -0.1803 -0.93138 -1.9384 -1.157 0.19024 0.46577 2.3486 2.0413  
3.9166 2.6125]。

第二步：用其中一个序列  $x_1(n)$  估计出不同阶数时的预测系数  $a_p(k)$ ,  $p = 1, 2, \dots, 30$ ;  
利用 `aryule()` 函数计算  $p = 1, 2, \dots, 30$  的预测系数, 数据较多省去。

第三步：用计算出来的预测系数构成的系统处理另外一个序列  $x_2(n)$ , 得到不同阶数时的预测均方误差  $\hat{\sigma}_{wp}^2 = [1.1382 \quad 1.1433 \quad 1.1445 \quad 1.1453 \quad 1.1311 \quad 1.1336 \quad 1.1403 \quad 1.1357$   
 $1.1444 \quad 1.1569 \quad 1.1674 \quad 1.168 \quad 1.1688 \quad 1.1197 \quad 1.041 \quad 1.045 \quad 1.033 \quad 1.0192$   
 $1.0199 \quad 1.0178 \quad 1.0505 \quad 1.0674 \quad 1.0515 \quad 1.062 \quad 1.0547 \quad 1.0521 \quad 1.0599 \quad 1.066 \quad 1.0632$   
 $1.0414]$ ,  $p = 1, 2, \dots, 30$ ;

第四步：求  $FPE(p) = \hat{\sigma}_{wp}^2 \left( \frac{N+p+1}{N-p-1} \right)$ , 计算得到  $FPE(p) = [1.29 \quad 1.3799 \quad 1.4715$   
 $1.5695 \quad 1.6532 \quad 1.7685 \quad 1.9005 \quad 2.0244 \quad 2.1847 \quad 2.3688 \quad 2.5682 \quad 2.7664 \quad 2.9868$   
 $3.0957 \quad 3.1229 \quad 3.4137 \quad 3.6892 \quad 3.9984 \quad 4.4194 \quad 4.9039 \quad 5.6727 \quad 6.5232 \quad 7.3607$   
 $8.6475 \quad 10.196 \quad 12.415 \quad 15.899 \quad 21.675 \quad 32.959 \quad 65.608]$ , 找到最小值, 最佳阶数  $p_{opt} = 1$ 。

除了用式(3-52)进行阶数的选择, 还可以采用试验的方法, 如果某一个  $p$  值满足预先规定的某些要求, 则认为该  $p$  值是最优的。在短数据情况下, 根据经验, AR 模型的阶次选在  $N/3 \sim N/2$  的范围内较好。