

利用实数 LMS 自适应滤波器 计算 RDFT 与 DFT *

吴一全 朱兆达 叶蓁如

(南京航空航天大学电子工程系 南京 210016)

1994 年 4 月 10 日收到

1996 年 1 月 10 日定稿

摘要 实离散 Fourier 变换 (RDFT) 是 1985 年 Ersoy 提出的一种实变换, 在许多信号处理应用中其性能优于离散 Fourier 变换 (DFT)。本文建立了实数 LMS 自适应算法与 RDFT 之间的联系, 提出了利用实数 LMS 自适应滤波器计算 RDFT 与 DFT 的一种方法。该算法与 Widrow 算法不同, 把 RDFT 的实数变换核作为自适应滤波器的输入矢量, 用实数 LMS 自适应算法进行计算。整个过程仅涉及实数运算, 所需存贮单元的数目只有 Widrow 算法的一半。当计算实序列 DFT 时, 实乘次数只有 Widrow 算法的三分之一, 实加次数不到 Widrow 算法的五分之一; 而计算复序列 DFT 时, 实乘次数也只有 Widrow 算法的三分之二, 实加次数还不到 Widrow 算法的五分之三。此算法更适于并行处理与 VLSI 实现, 它为计算 RDFT 与 DFT 提供了一条新的神经网络途径。

PACS 数 43.60

Computation of the RDFT and DFT through the real LMS adaptive filter

WU Yiquan ZHU Zhaoda YIE Zhenru

(Nanjing University of Aeronautics and Astronautics Nanjing 210016)

Received Apr. 10, 1994

Revised Jan. 10, 1996

Abstract The real discrete Fourier transform (RDFT) is a real transform introduced by Ersoy in 1985. The RDFT has been found superior to the discrete Fourier transform (DFT) in signal processing applications. A relation between the real LMS adaptive algorithm and the RDFT is established. A new algorithm is proposed to compute the RDFT and DFT via the real LMS adaptive filter. Instead of Widrow's approach, the real transform kernel of the RDFT serves as the input vector of the real LMS adaptive filter. All the operations involved are real. As compared with Widrow's algorithm, the proposed algorithm reduces the storage by a factor of 2. For the real-value DFT, it requires one-third as many real multiplications and slightly less than one-fifth as many additions. For the complex DFT, it requires

* 国家教委高校博士点专项科研基金、国家航空科学基金资助

two-third as many multiplications and less than three-fifth as many additions. The proposed algorithm is applicable to parallel processing and to VLSI implementation. It provides a neural net approach to the RDFT and DFT.

引言

1987 年 Widrow 等人提出了利用复数最小均方 (LMS) 自适应算法^[1,2] 计算离散 Fourier 变换 (DFT) 的一种新方法^[3]. 该方法为 DFT 的并行计算与 VLSI 实现提供了与 FFT 不同的另一种手段, 然而缺点是无论对实信号还是复信号, 整个过程都必须采用复数运算. 这是由于 Widrow 把 DFT 的复数变换核 $\{(1/\sqrt{N}) \exp[-i2\pi kn]; k = 0, 1, \dots, N-1\}$ 作为自适应滤波器的输入矢量, 因而只能用复数 LMS 算法进行计算.

1985 年 Ersoy 曾提出一种实数变换核的离散正交变换, 称之为实离散 Fourier 变换 (RDFT), 它对应于三角函数形式的 Fourier 级数, 如同 DFT 对应于复指数形式的 Fourier 级数^[4]. 在许多信号处理应用中, RDFT 的性能优于 DFT^[5-9], 且 RDFT 仅涉及实数运算, 其硬件与软件实现比 DFT 优越, 利用 RDFT 也可得到实序列与复序列的 DFT.

本文建立了实数 LMS 自适应算法与 RDFT 之间的联系, 提出了一种利用实数 LMS 算法计算 RDFT 与 DFT 的方法. 与 Widrow 算法不同, 本文提出的算法把 RDFT 的实数变换核 $\{\sqrt{2/N} V(k) \cos[(2\pi kn/N) + \theta(k)]; k = 0, 1, \dots, N-1\}$ 作为自适应滤波器的输入矢量, 并用实数 LMS 算法进行计算. 由于整个运算过程不涉及任何复数运算, 因此, 所需存贮单元的数目只有 Widrow 算法的一半, 而运算量大大减少, 更适于并行计算与 VLSI 实现.

1 RDFT 的定义及其与 DFT 的关系

RDFT 是一种实离散正交变换, 它将给定的长度为 N 的实序列 $\{x(n); n = 0, 1, \dots, N-1\}$ 变换成另一个长度为 N 的实序列 $\{X(k); k = 0, 1, \dots, N-1\}$, 其归一化正交的正反变换分别定义为^[4]:

$$\text{正变换(RDFT): } X(k) = \text{RDFT}[x(n)] = \sqrt{\frac{2}{N}} V(k) \sum_{n=0}^{N-1} x(n) \cos \left[\frac{2\pi kn}{N} + \theta(k) \right] \quad (1)$$

这里,

$$\theta(k) = \begin{cases} 0, & k = 0, 1, \dots, N_1 \\ \frac{\pi}{2}, & k = N_1 + 1, \dots, N-1 \end{cases} \quad N_1 = \begin{cases} \frac{N}{2}, & N \text{ 为偶数} \\ \frac{N-1}{2}, & N \text{ 为奇数} \end{cases}$$

$$V(k) = \begin{cases} \frac{1}{\sqrt{2}}, & k = 0, \frac{N}{2} \\ 1, & \text{其它} \end{cases}$$

当 N 为偶数时, (1) 式也可写成:

$$\begin{cases} X(k) = X_1(k) = \sqrt{\frac{2}{N}} V(k) \sum_{n=0}^{N-1} x(n) \cos \frac{2\pi kn}{N}, & k = 0, 1, \dots, \frac{N}{2} \\ X(N-k) = X_0(k) = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} x(n) \sin \frac{2\pi kn}{N}, & k = 1, \dots, \frac{N}{2} - 1 \end{cases}$$

$$\text{反变换(IRDFT): } x(n) = \text{IRDFT}[X(k)] = \sqrt{\frac{2}{N}} \sum_{k=0}^{N-1} X(k)V(k) \cos \left[\frac{2\pi kn}{N} + \theta(k) \right]$$

而实序列 DFT 则是一种酉变换, 它将给定的长度为 N 的实序列 $x(n)$ 变换为频域复序列 $\{Y(k); k = 0, 1, \dots, N-1\}$, 其归一化正交形式定义为:

$$Y(k) = \text{DFT}[x(n)] = \sqrt{\frac{1}{N}} \sum_{n=0}^{N-1} x(n)e^{-i\frac{2\pi kn}{N}} \quad (2)$$

由 (1)、(2) 式可见, 实序列的 DFT 与 RDFT 之间有如下关系:

$$\begin{cases} Y(0) = X(0) \\ Y\left(\frac{N}{2}\right) = X\left(\frac{N}{2}\right) \\ Y(k) = \frac{1}{\sqrt{2}}[X(k) - iX(N-k)] \\ Y(N-k) = \frac{1}{\sqrt{2}}[X(k) + iX(N-k)] \quad k = 1, 2, \dots, \frac{N}{2} - 1 \quad (N \text{ 为偶数}) \end{cases} \quad (3)$$

或

$$\begin{cases} Y(0) = X(0) \\ Y(k) = \frac{1}{\sqrt{2}}[X(k) - iX(N-k)] \\ Y(N-k) = \frac{1}{\sqrt{2}}[X(k) + iX(N-k)] \quad k = 1, 2, \dots, \frac{N-1}{2} \quad (N \text{ 为奇数}) \end{cases} \quad (4)$$

即实序列 $x(n)$ 的 DFT $Y(k)$ (复数) 可直接由 $x(n)$ 的 RDFT $X(k)$ (实数) 得到。对于复数序列 $x(n) = \text{Re}[x(n)] + i\text{Im}[x(n)]$, 其 DFT 等价于计算两个 RDFT, 最后再把所得结果组合起来。

若

$$\begin{aligned} Y(k) &= \text{DFT}[x(n)] \\ X_R(k) &= \text{RDFT}\{\text{Re}[x(n)]\}, \quad X_I(k) = \text{RDFT}\{\text{Im}[x(n)]\} \end{aligned}$$

则

$$\begin{cases} Y(0) = X_R(0) + iX_I(0) \\ Y\left(\frac{N}{2}\right) = X_R\left(\frac{N}{2}\right) + iX_I\left(\frac{N}{2}\right) \\ Y(k) = \frac{1}{\sqrt{2}}\{X_R(k) + X_I(N-k) + i[X_I(k) - X_R(N-k)]\} \quad k = 1, 2, \dots, \frac{N}{2} - 1 \\ Y(N-k) = \frac{1}{\sqrt{2}}[X_R(k) - X_I(N-k) + i[X_I(k) + X_R(N-k)]] \quad (N \text{ 为偶数}) \end{cases} \quad (5)$$

即复序列的 DFT 也可经过 $4(N/2 - 1)$ 次实加的附加运算由两个 RDFT 求得, 而不涉及任何复数运算。

2 利用实数 LMS 自适应滤波器计算 RDFT

我们构造的利用实数 LMS 自适应算法计算 RDFT 的自适应滤波器的框图如图 1 所示。

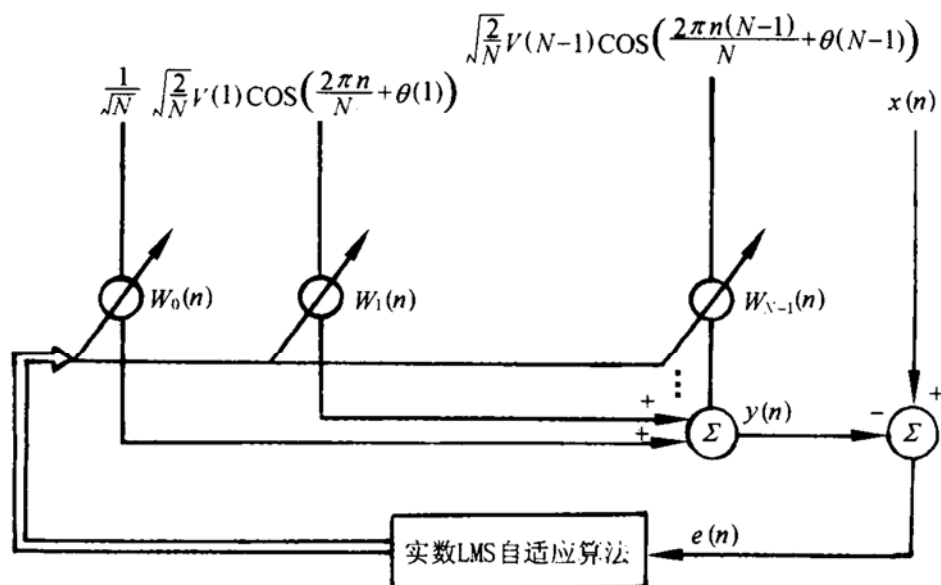


图 1 利用实数 LMS 自适应算法计算 RDFT 的自适应滤波器

RDFT 的变换核 $\{\sqrt{2/N}V(k)\cos[(2\pi kn/N) + \theta(k)]; k = 0, 1, \dots, N-1\}$ 作为自适应过程的输入矢量, 而欲进行 RDFT 的实信号 $x(n)$ 作为“期望响应”。这里, 权系数 $W_0(n), \dots, W_{N-1}(n)$, 加权和 $y(n)$ 和误差 $e(n)$ 为实数。

若令

$$\mathbf{R}(n) \triangleq [r_0(n), r_1(n), \dots, r_{N-1}(n)]^T \quad (6)$$

其中

$$r_k(n) = \sqrt{\frac{2}{N}}V(k)\cos\left(\frac{2\pi kn}{N} + \theta(k)\right), \quad k = 0, 1, \dots, N-1$$

则

$$\mathbf{W}(n) \triangleq [W_0(n), W_1(n), \dots, W_{N-1}(n)]^T \quad (7)$$

$$y(n) = \mathbf{R}^T(n)\mathbf{W}(n) \quad (8)$$

$$e(n) = x(n) - \mathbf{R}^T(n)\mathbf{W}(n) \quad (9)$$

自适应滤波的目的是根据一定的自适应算法, 调节权矢量, 使得滤波器的输出误差为最小。对于实数 LMS 自适应算法, 选择性能函数为 $e^2(n) = [x(n) - \mathbf{R}^T(n)\mathbf{W}(n)]^2$, 权矢量更新的方向是取 $e^2(n)$ 增加的负向, 可得

$$\mathbf{W}(n+1) = \mathbf{W}(n) + 2\mu e(n)\mathbf{R}(n) \quad (0 < \mu < 1) \quad (10)$$

这里, $\mathbf{W}(n)$ 为当前权矢量, $\mathbf{W}(n+1)$ 为下一时刻权矢量, μ 为收敛因子。将 (9) 式代入 (10) 式得

$$\mathbf{W}(n+1) = \mathbf{W}(n) + 2\mu x(n)\mathbf{R}(n) - 2\mu \mathbf{R}(n)\mathbf{R}^T(n)\mathbf{W}(n) \quad (11)$$

若设置初始权矢量为零矢量, 即 $\mathbf{W}(0) = [0, \dots, 0]^T$, 则可通过迭代得到随时间更新的权矢量:

$$\mathbf{W}(1) = 2\mu x(0)\mathbf{R}(0) \quad (12)$$

$$\mathbf{W}(2) = 2\mu[x(0)\mathbf{R}(0) + x(1)\mathbf{R}(1)] - 4\mu^2\mathbf{R}(1)[\mathbf{R}^T(1)\mathbf{R}(0)]x(0) \quad (13)$$

利用 RDFT 的归一化正交性关系:

$$\mathbf{R}^T(p)\mathbf{R}(q) = \sum_{k=0}^{N-1} \left\{ \sqrt{\frac{2}{N}} V(k) \cos \left[\frac{2\pi kp}{N} + \theta(k) \right] \right\} \left\{ \sqrt{\frac{2}{N}} V(k) \cos \left[\frac{2\pi kq}{N} + \theta(k) \right] \right\} = \begin{cases} 1, & p = q \\ 0, & p \neq q \end{cases} \quad p, q = 0, 1, \dots, N-1 \quad (14)$$

得 $\mathbf{R}^T(1)\mathbf{R}(0) = 0$, 因此 (13) 式可简化为:

$$\mathbf{W}(2) = 2\mu[x(0)\mathbf{R}(0) + x(1)\mathbf{R}(1)]$$

同理可得

$$\mathbf{W}(n) = 2\mu \sum_{m=0}^{n-1} x(m)\mathbf{R}(m) \quad n = 1, \dots, N \quad (15)$$

当 $n = N$ 时, 由 (15) 式得:

$$\mathbf{W}(N) = 2\mu \sum_{m=0}^{N-1} x(m)\mathbf{R}(m) = 2\mu \begin{bmatrix} \frac{1}{\sqrt{N}} \sum_{m=0}^{N-1} x(m) \cos \left[\frac{2\pi m \cdot 0}{N} + \theta(0) \right] \\ \sqrt{\frac{2}{N}} \sum_{m=0}^{N-1} x(m) \cos \left[\frac{2\pi m \cdot 1}{N} + \theta(1) \right] \\ \vdots \\ \sqrt{\frac{2}{N}} \sum_{m=0}^{N-1} x(m) \cos \left[\frac{2\pi m \cdot (N-1)}{N} + \theta(N-1) \right] \end{bmatrix} \quad (16)$$

除了比例因子 2μ 外, 权矢量 $\mathbf{W}(N)$ 的 N 个元素构成了 $x(n)$ 的最初 N 个样值 $\{x(n); n = 0, 1, \dots, N-1\}$ 的 RDFT。值得注意的是, (15) 式是基于矢量 $\mathbf{R}(n)$ 在 $1 \leq n \leq N$ 范围内各时刻的正交性导出的, n 超出这一范围, (15) 式不再成立。因此, 我们需要再按 (11) 式计算 $\mathbf{W}(N+1)$:

$$\begin{aligned} \mathbf{W}(N+1) &= \mathbf{W}(N) + 2\mu x(N)\mathbf{R}(N) - 2\mu\mathbf{R}(N)\mathbf{R}^T(N)\mathbf{W}(N) = \\ &2\mu \sum_{m=0}^N x(m)\mathbf{R}(m) - 4\mu^2\mathbf{R}(N)[\mathbf{R}^T(N) \sum_{m=0}^{N-1} x(m)\mathbf{R}(m)] \end{aligned}$$

注意到 $\mathbf{R}(n)$ 的周期性 $\mathbf{R}(N) = \mathbf{R}(0)$, 并利用 (14) 式的归一化正交关系, 得

$$\mathbf{W}(N+1) = 2\mu \sum_{m=1}^N x(m)\mathbf{R}(m) + 2\mu(1-2\mu)x(0)\mathbf{R}(0)$$

同理可得

$$\mathbf{W}(N+2) = 2\mu \sum_{m=2}^{N+1} x(m)\mathbf{R}(m) + 2\mu(1-2\mu)[x(0)\mathbf{R}(0) + x(1)\mathbf{R}(1)]$$

上述结果可以推广为

$$\mathbf{W}(n) = 2\mu \sum_{m=n-N}^{n-1} x(m)\mathbf{R}(m) + 2\mu(1-2\mu) \sum_{m=0}^{n-N-1} x(m)\mathbf{R}(m) \quad n = N+1, \dots, 2N$$

按照同样的方法可以导出在初始条件 $\mathbf{W}(0) = \mathbf{o}$ 情况下, 对所有 $n \geq 1$ 成立的计算权矢量 $\mathbf{W}(n)$ 的一般公式:

$$\begin{aligned} \mathbf{W}(n) = & 2\mu \sum_{m=n-N}^{n-1} x(m)\mathbf{R}(m) + 2\mu(1-2\mu) \sum_{m=n-2N}^{n-N-1} x(m)\mathbf{R}(m) + \\ & 2\mu(1-2\mu)^2 \sum_{m=n-3N}^{n-2N-1} x(m)\mathbf{R}(m) + 2\mu(1-2\mu)^3 \sum_{m=n-4N}^{n-3N-1} x(m)\mathbf{R}(m) + \cdots \end{aligned} \quad (17)$$

上式中, 如果求和上限变成负数, 则该求和项不存在, 当下限为负数时, m 的范围必须从 $m=0$ 开始。现选择 $\mu = 1/2$, n 为 N 的任意整数倍, 即 $n = lN (l = 1, 2, \cdots)$ 则 (17) 式变成:

$$\begin{aligned} \mathbf{W}(lN) = & \sum_{m=lN-N}^{lN-1} x(m)\mathbf{R}(m) = \\ & \begin{bmatrix} \sqrt{\frac{2}{N}}V(0) \sum_{m=lN-N}^{lN-1} x(m) \cos \left[\frac{2\pi m \cdot 0}{N} + \theta(0) \right] \\ \sqrt{\frac{2}{N}}V(1) \sum_{m=lN-N}^{lN-1} x(m) \cos \left[\frac{2\pi m \cdot 1}{N} + \theta(1) \right] \\ \vdots \\ \sqrt{\frac{2}{N}}V(N-1) \sum_{m=lN-N}^{lN-1} x(m) \cos \left[\frac{2\pi m \cdot (N-1)}{N} + \theta(N-1) \right] \end{bmatrix} = \\ & \sum_{m=0}^{N-1} x(m+lN-N)\mathbf{R}(m) \end{aligned} \quad (18)$$

可见, $\mathbf{W}(lN)$ 恰好是在时刻 $n = lN$ 输入 $x(n)$ 的前 N 个样值 $\{x(n); n = lN - N, \cdots, lN - 1\}$ 的 RDFT。至此证明了适当选择收敛因子 μ , 可利用实数 LMS 自适应滤波器计算 RDFT。根据 RDFT 与 DFT 的关系 ((3)、(5) 式), 可得到实序列或复序列的 DFT。

3 与 Widrow 算法的比较

本文提出的利用实数 LMS 自适应算法计算 RDFT 与 DFT 的方法, 与 Widrow 提出的利用复数 LMS 自适应算法计算 DFT 的方法相比有明显的优点:

(1) Widrow 算法用 DFT 的复数变换核 $\{(1/\sqrt{N}) \exp[-i(2\pi kn/N)]; k = 0, 1, \cdots, N-1\}$ 作为自适应滤波器的输入矢量, 因此只能用复数 LMS 自适应算法, 权 $W_0(n), \cdots, W_{N-1}(n)$ 、加权和 $y(n)$ 与误差 $e(n)$ 均为复数。而本文算法用 RDFT 的实数变换核 $\{\sqrt{2/N}V(k) \cos[(2\pi kn/N) + \theta(k)]; k = 0, 1, \cdots, N-1\}$ 作为输入矢量, 采用实数 LMS 自适应算法, 相应的权、加权和及误差均为实数。因此, 本文算法所需存贮单元的数目只有 Widrow 算法的一半。

(2) Widrow 算法全部按复数运算, 而本文算法的整个运算过程仅涉及实数运算。因此, 本文算法的运算量大大减少。用 Widrow 算法计算每点 DFT 需 $2N-1$ 次复乘与 $2N-2$ 次复加 (计算实序列 DFT 与计算复序列 DFT 所需运算量相同, 均按复数运算)。由于每次复乘需采用 3 次实乘与 3 次实加实现, 每次复加需采用两次实加实现, 因此, Widrow 算法需 $3(2N-1)$ 次实乘与 $3(2N-1) + 2(2N-2) = 10N-7$ 次实加。而用本文算法计算实序列的每点 RDFT

与 DFT 只需 $2N - 1$ 次实乘与 $2N - 2$ 次实加。由此可见, 当用本文算法计算实序列 RDFT 与 DFT 时, 所需的实乘次数只有 Widrow 算法的三分之一, 实加次数还不到 Widrow 算法的五分之一。当需用本文算法计算复序列 DFT 时, 可先后计算两个实序列的 RDFT, 再把所得结果按 (5) 式组合起来即可。所需的实乘次数 ($= 2(2N - 1)$) 也只有 Widrow 算法的三分之二, 实加次数 ($= 2(2N - 2) + 4(N/2 - 1) = 6N - 8$) 不到 Widrow 算法的五分之三。事实上, 当并行计算这两个 RDFT 时, 用本文算法计算复序列 DFT 和计算实序列 DFT 所需的运行时间相近。为了进一步验证, 我们编制了 Widrow 算法与本文提出算法的一段小程序, 并对 $N = 2^4 \sim 2^{10}$ 的多个实序列在计算机上进行了测试, 结果表明本文算法所需的运行时间均不到 Widrow 算法的三分之一。

(3) 本文算法比 Widrow 算法更适于并行处理与 VLSI 实现。由于实数 LMS 自适应算法可直接用 Hopfield 神经网络实现^[10], 因此, 本文算法也为计算 RDFT 与 DFT 提供了一条新的神经网络计算途径。

4 结论

本文提出了利用实数 LMS 自适应算法计算 RDFT 与 DFT 的一种方法。该算法仅涉及实数运算, 所需存贮单元的数目只有 Widrow 算法的一半。当计算实序列 DFT 时, 本文算法的实乘次数只有 Widrow 算法的三分之一, 实加次数还不到 Widrow 算法的五分之一; 而当计算复序列 DFT 时, 实乘次数也只有 Widrow 算法的三分之二, 实加次数不到 Widrow 算法的五分之三。本文提出的算法更适合于并行处理与 VLSI 实现, 它为用 Hopfield 神经网络实现 RDFT 与 DFT 提供了一条新的途径。

参 考 文 献

- 1 Widrow B, Mccool J M, Ball M. The complex LMS algorithm. *Proc. IEEE*, 1975, **63**: 719—720
- 2 Widrow B, Stearns S D. Adaptive signal processing, Englewood Cliffs. NJ: Prentice Hall, 1985
- 3 Widrow B, Baudrenghien P, Vetterli M, Titchener P F. Fundamental relations between the LMS algorithm and the DFT. *IEEE Trans.*, 1987, **CAS-34**: 814—820
- 4 Ersoy O K. Real discrete Fourier transform. *IEEE Trans.*, 1985, **ASSP-33**: 880—882
- 5 Ersoy O K, Chen C H. Transform-coding of images with reduced complexity. *Computer Vision, Graphics and Image Processing*, 1988, **42**: 19—31
- 6 Ersoy O K, Zeng M. Nonlinear matched filtering. *J. Optical Society America*, 1989, **6**: 634—648
- 7 Ersoy O K, Chen C H. Learning of fast transform and spectral domain neural computing. *IEEE Trans.*, 1989, **CAS-36**: 704—711
- 8 Ersoy O K, Hu N C. Fast algorithms for the real discrete Fourier transforms. *Proc. IEEE ICASSP*, 1988: 1902—1905
- 9 Hu N C, Ersoy O K. Fast computation of real discrete Fourier transform for any number of data points. *IEEE Trans.*, 1991, **CAS-38**: 1280—1292
- 10 刘茜君, 陈钧量. 用 Hopfield 网络实现 LMS 自适应滤波器. 第四届全国信号处理学术会议论文集. 1992: 361—364