

几种自适应 IIR 陷波器的复数算法及其应用*

梁红† 李志舜

(西北工业大学航海学院 西安 710072)

摘要 在自适应 IIR 陷波器实数算法的基础上, 本文提出了三种基于梯度的自适应 IIR 陷波器的复数算法, 并进行了仿真比较。结果表明, 这三种复数算法均可检测复信号, 并估计信号频率, 其中“改进的”简化格型 IIR 陷波器复数算法收敛速度快、低信噪比下稳定性能好。湖上试验表明该算法不仅实时性好, 而且在低信噪比下的检测效果令人满意。

关键词 自适应陷波器, 复数算法, 主动声纳系统, 湖上试验

Complex adaptive IIR notch filter algorithms and their applications

Liang Hong Li Zhi-Shun

(College of Marine Northwestern Polytechnical University, Xi'an 710072)

Abstract The problem of detecting the complex narrowband signal using complex adaptive IIR notch filter is investigated. In this paper, three complex coefficient adaptive IIR notch filters using gradient-based algorithms are proposed. These three methods are compared among themselves in terms of detecting performance and estimation accuracy. Simulation results show that these proposed algorithms can all detect and estimate the frequency of sinusoid, but an “improved” simplified lattice complex algorithm outperforms the other two in the convergence speed, tracking speed and steady-state mean square error at low SNR. Lake trial results confirm the real-time detecting ability of our proposed “improved” algorithm.

Key words Adaptive notch filter, Complex algorithms, Active sonar system, Lake trial

2006-08-28 收稿; 2007-10-08 定稿

* “十五”国防重点预研项目资助。

作者简介: 梁红(1969-), 女, 江苏南通人, 副教授, 博士, 研究方向: 信号检测、参量估计、自适应信号处理。

李志舜(1938-2007), 男, 教授, 博士生导师。

† 通讯联系人 E-mail: lianghong@nwpu.edu.cn

1 引言

自适应 IIR 陷波器可以去除宽带信号中的窄带或单频干扰, 另一方面, 它还可增强噪声中的窄带或单频信号。近年来, 实数算法的自适应 IIR 陷波器在生物工程、通信和控制等领域获得了广泛的应用。但是在雷达、通信等系统中, 实际信号可能包含同相和正交分量, 其样本为复数, 在这种情况下, 就必须使用复数算法的滤波器来去除干扰或增强信号。复数算法实现线谱增强和干扰抵消研究较多的是采用 FIR 滤波器实现, 但是 FIR 滤波器的传递函数为多项式, 为了得到尖锐的滤波器截止带宽, 需要很长的滤波器阶数, 实时性较差。而 IIR 陷波器就显示出很大的优点, IIR 陷波器的幅频响应在单位圆上的一个特殊值点(即陷波频率 ω_0)处为零, 在单位圆其它点上的幅频响应值近似为常数, 用一个二阶滤波器就可很好地近似。

自适应 IIR 陷波器有直接型和格型两种, 格型结构可以独立调整滤波器的频率和带宽, 算法具有快速收敛的优良数值特性。本文主要讨论自适应最小均方误差准则的格型滤波器, 并在此基础上提出相应的复数算法, 用于对复信号进行处理。

2 格型 IIR 滤波器结构

图 1 所示为二阶格型 IIR 陷波器结构图, 其中输入序列为

$$x(k) = Ae^{j(2\pi f k + \theta)} + n(k), k = 1, 2, 3 \dots; \quad (1)$$

f 为归一化信号频率; $n(k)$ 是复高斯随机噪声, $n(k) = n_r(k) + jn_i(k)$, $n_r(k)$ 和 $n_i(k)$ 分别是零均值, 且方差为 σ_n^2 的实高斯随机过程。系统的传输函数为:

$$H(z) = \frac{1 + \sin \theta_2}{2} \frac{1 + 2 \sin \theta_1 z^{-1} + z^{-2}}{1 + \sin \theta_1 (1 + \sin \theta_2) z^{-1} + \sin \theta_2 z^{-2}} \quad (2)$$

式中 θ_1 (同图 1) 为陷波器的频率参数, θ_2 为与陷波器带宽有关的参数。如果陷波器的频率为 ω_0 , 3dB 衰减带宽为 B (ω_0 和 B 均为用采样频率进行归一化后的值), 则:

$$\theta_1 = \omega_0 - \pi/2 \quad \omega_0 \in [0, \pi] \quad (3)$$

$$\sin \theta_2 = \tan \left(\frac{\pi}{4} - \frac{B}{2} \right) \quad \theta_2 \in (0, \pi/2) \quad (4)$$

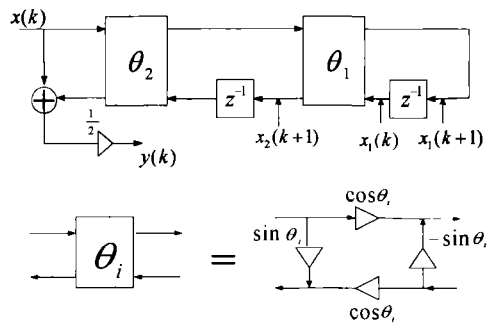


图 1 二阶格型陷波器结构

从 (3) 式可以看出, 陷波器的陷波频率 ω_0 只与 θ_1 有关, 与 θ_2 无关, 陷波器的带宽 B 只与 θ_2 有关, 也就是说在格型 IIR 陷波器中, 陷波器的陷波频率 ω_0 和陷波带宽 B 可以分别调整, 其中 θ_2 越靠近 $\pi/2$, 陷波器的带宽就越窄, 也就是说截止频率越尖锐。

3 复数算法

3.1 三种复数算法的提出

文献 [1] 中介绍了一种基于梯度的实数陷波器自适应算法, 见表 1, 其中 $\varphi(n)$ 是输出 $y(n)$ 对参数 θ_1 的差分, 称为梯度信号 $\varphi(n) = \partial y / \partial \theta_1$, y 为陷波器输出。 y_1 为需要检测的信号, μ 是自适应步长, 可以控制收敛速度和精度, μ_0 为初始步长, 且 $\mu_0 > 0$ 。

在此基础上本论文提出相应的自适应复数算法, 称为格型复数算法。因为 θ_1 的选择应使滤波器输出信号的能量最小, 即使代价函数 $y(k)y^*(k)$ 最小, 这里 “*” 代表复共轭。这样 $y(k)y^*(k)$ 对 θ_1 的导数为:

表 1 基于梯度的自适应 IIR 陷波器的实数算法

$$\begin{aligned} \begin{bmatrix} g_1 \\ \omega_1 \end{bmatrix} &= \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 \\ \sin \theta_2 & \cos \theta_2 \end{bmatrix} \begin{bmatrix} x(n) \\ x_2(n) \end{bmatrix} \\ y(n) &= \frac{1}{2} [x(n) + \omega_1] \quad y_1(n) = \frac{1}{2} [x(n) - \omega_1] \\ \begin{bmatrix} g_2 \\ \omega_2 \end{bmatrix} &= \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 \\ \sin \theta_2 & \cos \theta_2 \end{bmatrix} \begin{bmatrix} x_1(n) \\ \xi_2(n) \end{bmatrix} \\ \theta_1(n+1) &= \theta_1(n) - \mu(n)y(n)\varphi(n) \\ \begin{bmatrix} x_1(n+1) \\ x_2(n+1) \end{bmatrix} &= \begin{bmatrix} \cos \theta_1(n+1) & -\sin \theta_1(n+1) \\ \sin \theta_1(n+1) & \cos \theta_1(n+1) \end{bmatrix} \\ \begin{bmatrix} g_1 \\ x_1(n) \end{bmatrix} \begin{bmatrix} \xi_1(n+1) \\ \xi_2(n+1) \end{bmatrix} &= \\ \begin{bmatrix} \cos \theta_1(n+1) - \sin \theta_1(n+1) \\ \sin \theta_1(n+1) \cos \theta_1(n+1) \end{bmatrix} \begin{bmatrix} g_1 \\ \xi_1(n) \end{bmatrix} \\ \mu(n) &= \frac{\mu_0}{\sum_{k=0}^n \lambda^{n-k} [\varphi(k)]^2} \end{aligned}$$

$$\frac{\partial y(n)y^*(n)}{\partial \theta_1(n)} = 2Re[y(n)\varphi^*(n)] \quad (5)$$

$$\varphi(n) = \frac{\partial y(n)}{\partial \theta_1(n)} \quad (6)$$

格型复数算法滤波器中 θ_1 的迭代公式变为：

$$\theta_1(n+1) = \theta_1(n) - \mu Re[y(n)\varphi^*(n)] \quad (7)$$

$$\mu(n) = \frac{\mu_0}{\sum_{k=0}^n \lambda^{n-k} |\varphi(k)|^2} \quad 0 \leq \lambda < 1 \quad (8)$$

文献 [1] 对表 1 中的实数算法进行了简化，得到了简化的梯度实数算法，见表 2。

表 2 基于简化梯度的自适应 IIR 陷波器的实数算法

$$\begin{aligned} \begin{bmatrix} g_1 \\ g_2 \end{bmatrix} &= \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 \\ \sin \theta_2 & \cos \theta_2 \end{bmatrix} \begin{bmatrix} x(n) \\ x_2(n) \end{bmatrix} \\ \mu(n) &= \frac{\mu_0}{\sum_{k=0}^n \lambda^{n-k} [x_1(k)]^2} \quad 0 \leq \lambda < 1 \quad (9) \\ y(n) &= [x(n) + g_2]/2 \\ \theta_1(n+1) &= \theta_1(n) - \mu(n)[y(n)x_1(n)] \\ \begin{bmatrix} x_1(n+1) \\ x_2(n+1) \end{bmatrix} &= \begin{bmatrix} \cos \theta_1(n+1) - \sin \theta_1(n+1) \\ \sin \theta_1(n+1) \cos \theta_1(n+1) \end{bmatrix} \begin{bmatrix} g_1 \\ x_1(n) \end{bmatrix} \end{aligned}$$

据表 2 的实数算法，我们又可以得到另一种简化的格型复数算法：

$$\theta_1(n+1) = \theta_1(n) - \mu Re[y(n)x_1^*(n)] \quad (10)$$

相应地

$$\mu(n) = \frac{\mu_0}{\sum_{k=0}^n \lambda^{n-k} |x_1(k)|^2} \quad 0 \leq \lambda < 1 \quad (11)$$

该算法称为简化格型复数算法，其中 $x_1(k)$ 被称为简化的梯度信号，它是输入信号 $u(k)$ 经过下面的传输函数后产生的：

$$H'_B(z) = \frac{\cos \theta_1 \cos \theta_2 z^{-1}}{1 + \sin \theta_1 (1 + \sin \theta_2) z^{-1} + \sin \theta_2 z^{-2}} \quad (12)$$

文献 [2] 在简化的梯度实数算法上对自适应步长 μ 提出了一种改进算法，即在 (9) 式的分母中加入一控制收敛稳定性的因子 c 。对应地，我们的复数算法称为改进的简化格型复数算法，算法中自适应步长 μ 即为：

$$\mu(n) = \frac{\mu_0}{c + \sum_{k=0}^n \lambda^{n-k} |x_1(k)|^2} \quad 0 \leq \lambda < 1 \quad (13)$$

式 (13) 中 $\mu(n)$ 为自适应时变步长， μ_0 为初始步长，且 $\mu_0 > 0$ 。 c 为一常数，是控制收敛稳定性的因子， $1 < c \leq \sum_{k=0}^n \lambda^{n-k} |x_1(k)|^2$ ，这样在高信噪比下 $\mu(n)$ 受 c 的影响可以忽略，而在低信噪比时避免了 $\sum_{k=0}^n \lambda^{n-k} |x_1(k)|^2 \ll 1$ 而导致步长太大不满足收敛条件的情形，从而保证了在低信噪比时算法的稳健性，提高了算法的跟踪性能。

3.2 计算机仿真和外场试验结果

在实际工程应用中，由于某些声纳主动系统的工作频率高，如果不进行降采处理，就不能保证算法的实时性。对于发射单频或窄带信号情况，回波仍为窄带，其信息包含在其包络中，故可用正交采样或延时采样降低采样频率，然后进行信号检测。仿真中窄带信号的中心频率为 30kHz，采样频率为 200kHz，观测时间

300ms, 信号持续时间 100ms, 高斯白噪声; 解调频率 30.05 kHz, 包络的采样频率为 25kHz。采用正交解调处理, 得到的复信号的实部和虚部分别为:

$$x_r(i) = [x(8i) \cos(16\pi f_0 i T_s)]_{LP} \quad i = 1, 2, 3 \dots N$$

$$x_i(i) = [x(8i) \sin(16\pi f_0 i T_s)]_{LP} \quad i = 1, 2, 3 \dots N$$

式中 $[*]_{LP}$ 为低通滤波算子, $x(i)$ 为高采样获得的样本, T_s 为高采样间隔, f_0 为解调频

率, N 为降采样后的样本总数。图 2 的 (c)、(d)、(e)、(f) 是在输入信噪比为 -20dB 时, 三种复数算法对信号检测和频率估计的 100 次 Monte-Carlo 实验平均结果。可以看出, 在低信噪比情况下, 改进的简化格型复数算法不仅检测效果好(相当于匹配滤波), 而且频率估计准确, 平均结果稳定; 而其它两种复数算法在低信噪比时检测效果较差, 因为不能保证每次频率估计结果都正确, 从而频率的 100 次平均估计结果也不准确(见图 2(f))。

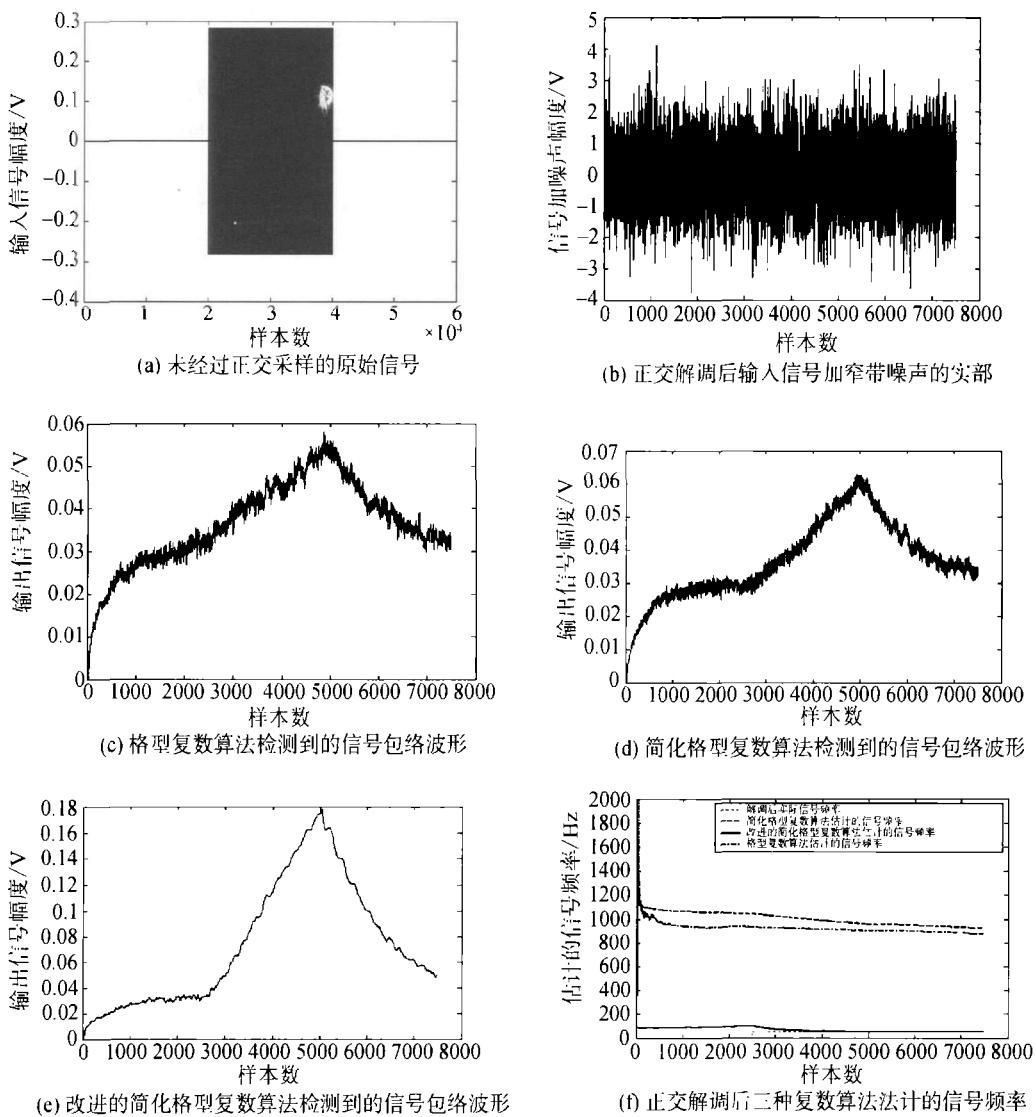


图 2 三种复数算法的检测和频率估计结果

由于计算机仿真结果表明改进的简化格型复数算法在低信噪比下检测和估计性能好, 我们编制了该算法的应用程序在高速数字信号处理机上运行, 进行了大量的实验室实时实验研究, 并经过了水池实验和外场试验, 取得了满意的结果。图 3 是在湖上试验时示波器上观测到的单频信号回波经正交解调、降采、波束形成后检测的包络波形(检测结果有溢出)。可见回波信号大约在 10.4s 左右出现。检测连续有效, 效果令人满意。

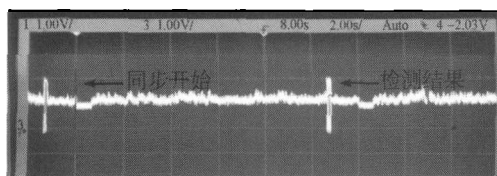


图 3 改进的简化格型复数算法的湖上试验检测结果

4 结论

本文提出了三种基于梯度的二阶自适应 IIR 陷波器的复数算法, 这些算法用来对复信号进行检测和频率估计, 仿真实验表明陷波器的复数算法可以很好地从背景噪声中检测到信

号, 并可对单频信号进行频率估计。三种复数算法在高信噪比条件下性能接近, 格型复数运算量稍大, 但频率估计效果最好, 简化格型复数算法运算量最小, 但“改进的”简化格型复数算法在低信噪比时检测效果比其它两种算法好, 频率估计准确, 而且性能稳定。湖上试验结果进一步证明了该复数算法的优良检测性能。

参 考 文 献

- [1] P.A.Regalia. Adaptive IIR Filtering in Signal Processing and Control. New York: Marcel Dekker, inc, 1995.
- [2] 梁红, 侯颜平. 稳健的变步长自适应 IIR 陷波器及其在动目标检测中的应用. 声学技术, 2005, 4: 197-201.
- [3] S.Nishimura, Hai-Yun Jiang, T.Hinamoto. Convergence analysis of complex adaptive IIR notch filters with colored noisy signal. IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, 1999: 325-328.
- [4] Soo-Chang Pei, Chien-Cheng Tseng. Complex adaptive IIR notch filter algorithm and its applications. IEEE Transactions on Circuits and Systems: Analog and Digital Signal Processing, 1994, 41(2): 158-163.
- [5] S.Nishimura, Hai-Yun Jiang. Gradient-based complex adaptive IIR notch filters for frequency Estimation. Proceeding of IEEE Asia Pacific Conference on Circuits and Systems /96, 1996: 235-239.