

文章编号: 1001-2486(2003)03-0036-05

采用 RLS 算法的功率倒置阵列的性能*

桑怀胜, 李峥嵘, 王飞雪, 雍少为, 郁文贤

(国防科技大学 ATR 国家重点实验室, 湖南长沙 410073)

摘要: 为解决功率倒置阵列采用 LMS 算法时收敛速度和稳态失调误差不能兼顾的缺点, 提出了采用 RLS 算法实现功率倒置阵列的方法。仿真结果表明, 功率倒置阵列采用 RLS 算法和 LMS 算法都能达到较好的抗干扰效果。和 LMS 算法相比, RLS 算法同时具有收敛速度快和稳态失调误差小的优点, 代价是计算量增大。

关键词: 功率倒置阵列; LMS 算法; RLS 算法

中图分类号: TN820.1 **文献标识码:** A

The Performance of Power Inversion Array Using RLS Algorithm

SANG Huai-sheng, LI Zheng-rong, WANG Fei-xue, YONG Shao-wei, YU Wen-xian

(ATR Lab, National Univ. of Defense Technology, Changsha 410073, China)

Abstract: The high convergent speed and small steady-state misadjustment error cannot be improved simultaneously in the power inversion array when using LMS algorithm. To solve this problem, a power inversion array using RLS algorithm is proposed. The simulation result shows that, both the RLS algorithm and LMS algorithm in the power inversion array have good anti-jamming performance. Comparing to LMS algorithm, the RLS algorithm can simultaneously get high convergent speed and small steady-state misadjustment error, but it bears a burden of large computation.

Key words: power inversion array; LMS algorithm; RLS algorithm

自适应阵列是一种有效的抗干扰措施。功率倒置(Power Inversion, PI)阵列^[1,2]不需要预先知道有用信号的特性和入射方向等先验信息, 实现简单, 在雷达、扩频通信等强干扰、弱信号的环境中得到了广泛应用。卫星导航定位系统具有重要的军事应用价值, 但其军事应用面临着强对抗环境, 卫星导航定位接收机(如 GPS 接收机)需要在弱信号、强干扰的信号接收环境下工作, 因此卫星导航定位接收机可以采用 PI 阵列提高抗干扰能力。

Gecan 研究了采用 LMS 算法的 GPS 接收机 PI 阵列的性能^[2]。LMS 算法的性能与步长相关, 算法的收敛速度和稳态失调误差对步长的要求互相矛盾, 难以兼顾。为解决上述问题, 本文提出采用 RLS 算法^[3]实现 PI 阵列, 通过仿真对采用 RLS 算法的 PI 阵列的性能进行了研究, 并与采用 LMS 算法的 PI 阵列的性能进行了对比。

1 采用 RLS 算法的 PI 阵列

1.1 PI 阵列

“功率倒置”指自适应阵列翻转两个接收信号的功率比的能力^[1]。Zahm 研究了 1 个弱信号、1 个强干扰条件下自适应阵列抑制强干扰、接收弱信号的信号接收技术^[4], 他的研究体现的正是 PI 的思想; Compton 阐述了 PI 的概念, 并对 PI 阵列的性能进行了研究^[1]。PI 阵列适合于卫星导航定位接收机的信号接收环境, 目前已有一些采用 PI 阵列的 GPS 接收机抗干扰的相关文献^[2,5,6]。

设天线阵元个数为 L , PI 阵列选择加权向量 $\mathbf{w} = [w_1, w_2, \dots, w_L]^T$, 使阵列输出信号的功率最

* 收稿日期: 2002-11-05

基金项目: 国家部委基金项目资助(D99-21)

作者简介: 桑怀胜(1970-), 男, 博士生。

小^[2,5]。为了防止得到无意义解 $w_1 = w_2 = \dots = w_L = 0$, 引入约束条件 $w_1 = C$, C 为任意不为 0 的常数, 为方便起见, 通常取 $C = 1$, PI 阵列的示意图见图 1。用 $\mathbf{x} = [x_1, x_2, \dots, x_L]^T$ 表示阵列输入信号向量, PI 阵列的最优权值为^[1,4]:

$$\mathbf{w}_{opt} = \gamma \mathbf{R}_{xx}^{-1} \mathbf{a}_0 \quad (1)$$

其中: γ 为任意不为零的常数; $\mathbf{a}_0 = [1, 0, \dots, 0]^T$ 是约束向量, 它使第 1 个阵元上的加权值固定为某个常数; $\mathbf{R}_{xx} = E[\mathbf{x} \mathbf{x}^H]$ 为阵列输入信号的自相关矩阵。

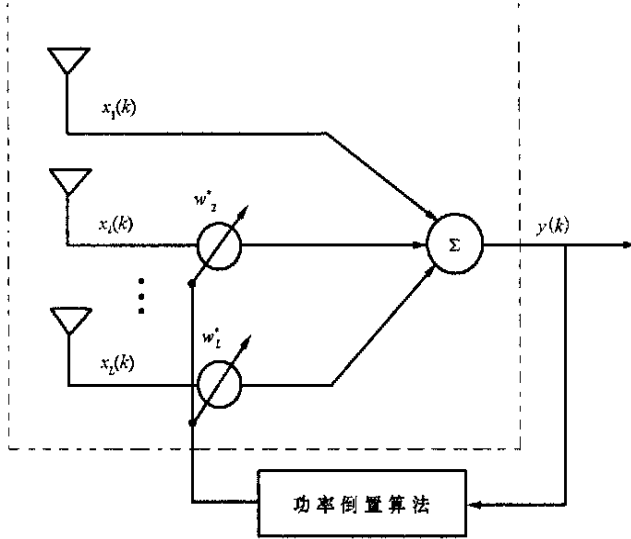


图 1 L 阵元 PI 阵列

Fig. 1 The L elements PI array

上述的 PI 阵列可以从另外一个角度理解: 阵元 1 上的信号作为参考信号, PI 阵列选择加权向量 $\mathbf{w}_a = [w_2, \dots, w_L]^T$, 使阵元 2~L 的输出加权和与参考信号之间的误差信号的均方值最小。记 $\mathbf{x}_a = [x_2, x_3, \dots, x_L]^T$, 阵列的输出信号为:

$$y = x_1 - \mathbf{w}_a^H \mathbf{x}_a \quad (2)$$

上标 H 表示共轭转置。相应于阵元 2~L 的最优权值为^[2,5]:

$$\mathbf{w}_{opt} = \mathbf{R}_{aa}^{-1} \mathbf{r}_{a1} \quad (3)$$

其中, $\mathbf{R}_{aa} = E[\mathbf{x}_a \mathbf{x}_a^H]$ 为阵元 2~L 上输入信号的自相关矩阵, $\mathbf{r}_{a1} = E[\mathbf{x}_a x_1^*]$ 为参考阵元上的信号与阵元 2~L 上信号的互相关向量。可以证明, (3) 式和(1) 式是等价的。

1.2 采用 RLS 算法的 PI 阵列

(3) 式的最优权值可以采用 LMS 算法趋近^[2]。众所周知, LMS 算法的性能与步长相关: 步长大, 算法收敛速度快, 但稳态失调误差大; 步长小, 稳态失调误差小, 但算法收敛速度慢; 两者不能兼顾。为解决上述矛盾, 本文提出采用 RLS 算法^[3]实现 PI 阵列。LMS 算法和 RLS 算法是两种应用非常广泛的自适应算法, 很多有关自适应滤波的文献都对这两种算法有详细的讨论, 本文重点研究 RLS 算法在功率倒置阵列中的应用, 对 LMS 算法和 RLS 算法本身不作详细讨论, 有兴趣的读者可参阅相关文献。功率倒置阵列中的 RLS 算法描述如下:

$$\mathbf{k}(k) = \frac{\lambda^{-1} \mathbf{P}(k-1) \mathbf{x}_a(k)}{1 + \lambda^{-1} \mathbf{x}_a^H(k) \mathbf{P}(k-1) \mathbf{x}_a(k)} \quad (4)$$

$$y(k) = x_1(k) - \mathbf{w}_a^H(k-1) \mathbf{x}_a(k) \quad (5)$$

$$\mathbf{w}_a(k) = \mathbf{w}_a(k-1) + \mathbf{k}(k) y^*(k) \quad (6)$$

$$\mathbf{P}(k) = \lambda^{-1} \mathbf{P}(k-1) - \lambda^{-1} \mathbf{k}(k) \mathbf{x}_a^H(k) \mathbf{P}(k-1) \quad (7)$$

其中 λ 称为遗忘因子, 它是小于但又接近于 1 的正常数, 平稳输入时 λ 可取为 1; $P(k)$ 定义为:

$$P(k) = \left[\sum_{i=1}^k \lambda^{k-i} \mathbf{x}_a(i) \mathbf{x}_a^H(i) \right]^{-1} \quad (8)$$

RLS 算法采用循环求逆方法计算逆矩阵, 与矩阵直接求逆相比, 大大减小了运算量; $k(k)$ 是为了书写方便引入的中间变量。 $w_a(k)$ 和 $P(k)$ 的初始值可设置为:

$$w_a(0) = \mathbf{0} \quad (9)$$

$$P(0) = \delta^{-1} \mathbf{I} \quad (10)$$

其中 $\mathbf{0}$ 表示零向量, 即把 w_a 的初始值设为零; \mathbf{I} 是 $(L-1) \times (L-1)$ 单位矩阵; δ 是一个小的正常数, 它只影响算法初始值的设置, 迭代次数很大时, 算法性能对 δ 值的选择不敏感。

RLS 算法比 LMS 算法收敛速度快, 其代价是计算复杂性增加。下面通过计算机仿真研究采用 RLS 算法的 PI 阵列的性能, 并与采用 LMS 算法的 PI 阵列进行对比。

2 仿真实验

天线阵采用一个阵元位于圆心、其余阵元均匀排列于圆周的 4 阵元圆天线阵, 圆周半径为射频载波的半个波长。信号仿真环境参照强干扰环境下的卫星导航定位接收机的信号接收环境进行设置: 信号入射角用 (θ, φ) 表示, 其中 θ 为俯仰角, φ 为方位角; 以对一颗卫星信号的接收为例, 有用信号为扩谱信号, 入射角为 $(70^\circ, 10^\circ)$, 信噪比为 -20dB ; 干扰信号取为宽带高斯干扰, 在信号通带内具有噪声特性, 入射角为 $(45^\circ, 120^\circ)$, 干噪比为 40dB ; 可计算得到输入信干噪比为 -60dB ; 噪声用高斯噪声进行模拟; 以干扰信号作为参考对输入信号进行归一化后送入天线阵进行处理, 这个归一化过程可视为对接收机 AGC 模块的模拟。仿真采用基带信号进行模拟, 仿真软件为 Matlab 6.1。

算法的性能主要考察以下几个方面:

(1) 收敛性能: 通过算法的收敛曲线进行评估;

(2) 稳态性能: 包括算法收敛后的稳态失调误差和抗干扰性能。注意到 PI 阵列的输出功率对应于通常的 LMS 算法和 RLS 算法的均方误差, 算法收敛后天线阵输出信号的功率用 P_a 表示, 最优维纳解对应的 PI 阵列输出信号的功率记为 P_{\min} , 失调误差定义为:

$$\varepsilon = (P_a - P_{\min}) / P_{\min} \quad (11)$$

失调误差通常用百分比来表示。抗干扰性能用算法收敛后 PI 阵列的输出信干噪比评估, 输出信干噪比采用下式计算:

$$SINR_{\text{out}} = \frac{\mathbf{w}^H \mathbf{R}_{ss} \mathbf{w}}{\mathbf{w}^H \mathbf{R}_{jj} \mathbf{w} + \sigma_n^2 \mathbf{w}^H \mathbf{w}} \quad (12)$$

其中 \mathbf{w} 为算法收敛后的阵列加权向量, \mathbf{R}_{ss} 为输入信号中有用信号的自相关矩阵, \mathbf{R}_{jj} 为输入信号中的干扰的自相关矩阵, σ_n^2 为输入信号中噪声的方差。

(3) 算法的运算量: 比较算法一次迭代所需的加法、乘法和除法次数。

2.1 收敛性能

LMS 算法和 RLS 算法的收敛特性曲线如图 2 所示, 图中的曲线是对 100 次运算进行平均的结果。可以看出, LMS 算法的收敛性能与步长相关, 步长为 0.005 时, LMS 算法需要大约 400 次左右迭代才能收敛, 其稳态下的输出功率和 RLS 算法相近; 步长为 0.1 时, LMS 算法只需 10 余次迭代就可收敛, 但收敛后不稳定, 且稳态下的输出功率比 RLS 算法及步长为 0.005 的 LMS 算法大。RLS 算法只需 10 余次迭代就可收敛, 和步长为 0.1 的 LMS 算法收敛速度相当, 收敛后的输出信号功率较小, 与步长为 0.005 的 LMS 算法的稳态输出功率相当。

2.2 稳态方向图

图 3 分别是 LMS 算法(步长为 0.005)和 RLS 算法收敛后 PI 阵列的阵列方向图, 为便于比较, 用最大增益进行了归一化。从图 3 可以看出, 两种算法收敛后都能在 $(45^\circ, 120^\circ)$ 的干扰方向上形成比较深的

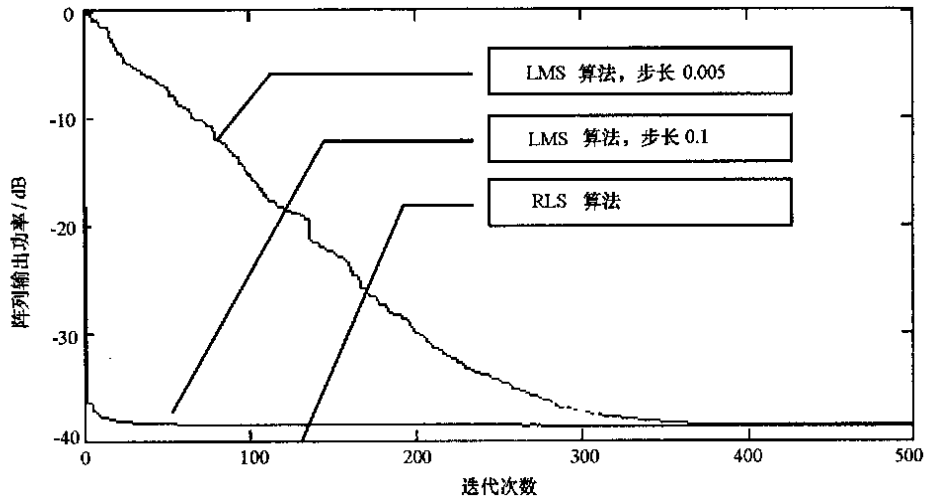


图 2 采用 LMS 算法和 RLS 算法的 PI 阵列输出功率曲线

Fig. 2 The curve of PI array output power using LMS algorithms and RLS algorithm

“零陷”, 达到了抑制干扰的目的。相比而言, RLS 算法形成的方向图“零陷”深度更深些, 通过多次仿真观察发现, LMS 算法(步长为 0.005)收敛后形成的阵列方向图“零陷”深度一般在 -60dB 左右, RLS 算法收敛后形成的阵列方向图“零陷”深度一般在 -70dB 左右。由于有用信号很弱, 阵列方向图在 $(70^\circ, 10^\circ)$ 的有用信号方向上没有形成“零陷”, 不会抑制有用信号; 当有用信号和干扰方向一致或接近时, 有用信号也会被抑制。

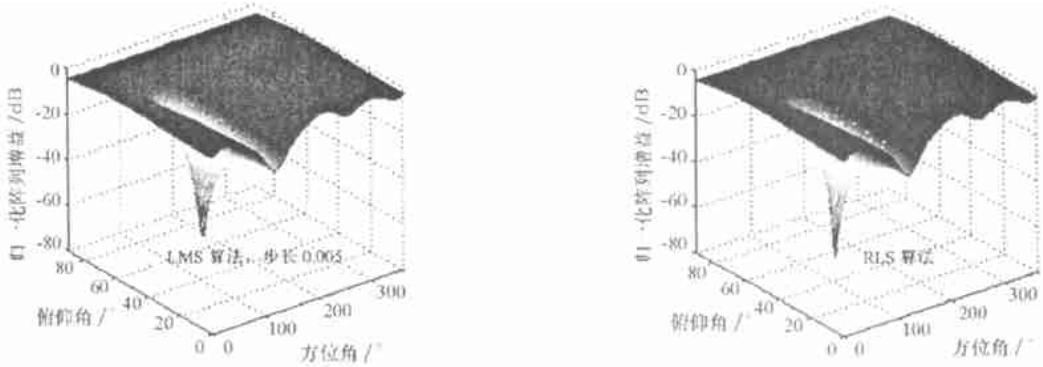


图 3 采用 LMS 算法和 RLS 算法的 PI 阵列稳态方向图

Fig. 3 The directional patterns after convergence of PI array using LMS algorithms and RLS algorithm

2.3 稳态输出功率和输出信干噪比

表 1 列出了算法收敛后的输出信号功率、失调误差和输出信干噪比, 表中的数值是对收敛后的 5000 个稳态权向量下的输出能指标值进行平均的结果, 表中还列出了(1)式中 $\gamma = 1$ 时的最优权值对应的输出性能指标值以便对比。从表 1 可以看出, RLS 算法收敛后的输出性能与小步长(0.005) LMS 算法收敛后的输出性能接近, 且与最优权值对应的输出性能相当; 而大步长(0.1) LMS 算法收敛后的性能较差。

表 1 采用 LMS 算法和 RLS 算法的 PI 阵列的稳态输出功率、输出信干噪比和失调误差

Tab. 1 The output power, output SINR and misadjustment error after convergence of PI array using LMS algorithms and RLS algorithm

	LMS 算法(步长 0.005)	LMS 算法(步长 0.1)	RLS 算法	最优权值
输出功率(dB)	- 38.62	- 37.02	- 38.67	- 38.72
失调误差(%)	2.33	47.91	1.16	0
输出信干噪比(dB)	- 19.82	- 21.44	- 19.89	- 19.97

由表 1 我们还可以发现一种现象,最优权值对应的输出信噪比反而比小步长的 LMS 算法和 RLS 算法收敛后的输出信干噪比低,这是因为(1)式的最优权值是以阵列输出功率最小作为准则,它并不使阵列的输出信干噪比最大。

2.4 运算量的比较

表 2 列出了迭代一次 LMS 算法和 RLS 算法所需的实数加法、乘法和除法的次数, L 为阵元个数。可见 RLS 算法的运算量比 LMS 算法大得多,当阵元个数较多时,RLS 算法运算量将迅速增大。

表 2 1 次迭代 LMS 算法和 RLS 算法的运算量比较

Tab. 2 The computation of LMS algorithm and RLS algorithm in one iteration

	实数加法次数	实数乘法次数	实数除法次数
LMS 算法	$4(L-1)$	$8(L-1)+2$	0
RLS 算法	$4(L-1)^3+10(L-1)^2+8(L-1)$	$4(L-1)^3+14(L-1)^2+16(L-1)$	$2(L-1)$

3 小结

本文研究了采用了 RLS 算法的 PI 阵列的性能,并把它与采用 LMS 算法的 PI 阵列的性能进行了比较。结果表明,采用 RLS 算法和 LMS 算法的 PI 阵列都可以达到较好的抗干扰效果。和 LMS 算法相比,RLS 算法较好地解决了 LMS 算法收敛速度和稳态失调误差不能兼顾的矛盾,算法初始阶段收敛速度快,收敛后的稳态失调误差小,但是算法所需的运算量大大增加。如果不需要过多考虑硬件实现代价,且硬件运算速度能够满足 RLS 算法对实时运算的要求,采用 RLS 算法实现 PI 阵列能够获得更好的性能。

参考文献:

- [1] Compton R T. The Power-inversion Adaptive Array: Concept and Performance [J]. IEEE Trans. on AES, 1979, 15(6): 803- 814.
- [2] Gecan A, Zoltowski M. Power Minimization Techniques for GPS Null Steering Antenna [C]. ION GPS- 95 Proceedings, 1995: 861- 868.
- [3] Haykin S. Adaptive Filter Theory (Third Edition) [M]. 北京: 电子工业出版社, 1998.
- [4] Zahn C L. Application of Adaptive Arrays to Suppress Strong Jammers in the Presence of Weak Signals [J]. IEEE Trans. on AES, 1973, 9(2): 260- 271.
- [5] 贾洪峰, 康锡章. GPS 接收机天线自适应抗干扰系统的设计[J]. 通信学报, 2001, 22(8): 54- 59.
- [6] 康锡章, 雷志东. 基于自适应二维阵列的 GPS 接收机抗干扰的研究[J]. 电子对抗, 1998(1): 8- 16.