

基于 CORDIC 的滑窗最小二乘递推算法

石斌斌, 钱林杰, 程 翥, 皇甫堪

(国防科学技术大学电子科学与工程学院, 湖南 长沙 410073)

摘要: 针对最小二乘算法涉及对数据自相关矩阵求逆, 若直接求逆则计算量比较大, 且求逆过程对误差比较敏感, 有限字长效应明显的问题, 提出一种基于 Givens 旋转的滑窗 RLS 实现方法, 并且给出了 CORDIC 实现的方法, 提高了基于 QR 分解的最小二乘算法实现的计算效率和数据吞吐率, 使其适应于需要极高采样率的应用, 并且该算法能够以滑窗形式的样本作为输入, 比传统的 Givens 算法能够适应更多的应用环境。仿真结果验证了该方法的有效性。

关键词: 滑窗; QR 分解; Givens 旋转; 坐标旋转数字计算方法

中图分类号: TN 911.7

文献标志码: A

DOI: 10.3969/j.issn.1001-506X.2010.11.10

Sliding window RLS method based on CORDIC

SHI Bin-bin, QIAN Lin-jie, CHENG Zhu, HUANGFU Kan

(School of Electronic Science and Engineering, National Univ. of Defense Technology, Changsha 410073, China)

Abstract: A sliding window RLS method based on givens rotation is proposed, which can be used to improve calculative efficiency of the least square adaptive algorithm based on QR decomposition and increase data throughput. This method is suitable for high sampling rate application, and the input data with sliding window form can suit more application environment than traditional Givens algorithm. Simulation results are given to demonstrate the effectiveness of the new method.

Keywords: sliding window; QR decomposition; Givens rotation; coordinate rotation digital computer (CORDIC)

0 引言

最小二乘估计算法拥有较快的收敛速度, 在自适应信号处理中有广泛的应用。但是最小二乘算法涉及对数据自相关矩阵求逆, 若采用协方差阵直接求逆方法 (direct matrix inversion, DMI), 则计算量比较大, 并且求逆的过程对误差比较敏感, 有限字长效应明显。为解决这一问题, 工程上常用的方法是避开对数据自相关矩阵的求逆, 而对数据矩阵进行 QR 分解, 改善系统的条件数, 提高数值稳定性^[1]。尽管所有最小二乘估计算法都是解同样一组方程, 但只有在计算中采用无限精度时, 不同算法才是等价的。当在算法实现时采用有限精度运算时, 舍入误差的影响将导致估计误差对不同算法的影响不同。Givens 旋转可以利用 CORDIC 方法实现^[2-5], 计算精度可以得到很大提高, 因此具有较高的数值稳定性^[6-7]。并且如果 Givens 算法以 CORDIC 的方式实现^[8-11], 则具有模块化的结构, 这个特点使得该算法适于用 VLSI 实现, 则有很强的实用价值。文献[1]介绍了一种基于 Givens 的最小二乘时间递推算法。

但在某些实际应用中, 如雷达的杂波抑制、声纳阵列、卫星导航信号抗干扰, 采用遗忘因子的样本选取方式, 不能适应局部平稳的杂波环境或时间变化较快的干扰环境, 滑窗形式的样本选取方式能较好的体现最小二乘算法收敛速度快的优势, 因而更适合这些场景。为了解决这一问题, 本文提出了一种基于 Givens 的滑窗 RLS (sliding window RLS method based on Givens, SWG-RLS) 方法。

1 信号模型

假设有 M 个天线同时接收信号, 数据矩阵的列为某一个天线接收到的信号数据。则 n 时刻的数据矩阵为

$$\mathbf{A}_M(n) = \begin{bmatrix} x_1(n-k+1) & \cdots & x_M(n-k+1) \\ \vdots & \ddots & \vdots \\ x_1(n) & \cdots & x_M(n) \end{bmatrix} = \begin{bmatrix} \mathbf{x}(n-k+1) \\ \mathbf{x}(n-k+2) \\ \vdots \\ \mathbf{x}(n) \end{bmatrix} \quad (1)$$

式中, $x_i(n)$ 表示第 i 个天线在 n 时刻接收到的信号; $\mathbf{x}(n)$ 表示 n 时刻 M 个天线接收到的信号矢量; $\mathbf{x}_i(n)$ 表示第 i 个天线从 $n-k+1$ 时刻到 n 时刻接收到的共 k 点的信号矢量。

n 时刻的数据矩阵 QR 分解后的 \mathbf{R} 阵可以表示为

$$\mathbf{R}(n) = \begin{bmatrix} r_{11}(n) & r_{12}(n) & \cdots & r_{1M}(n) \\ 0 & r_{22}(n) & \cdots & r_{2M}(n) \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & r_{MM}(n) \end{bmatrix} \quad (2)$$

2 基于 Givens 的 QR 分解时间递推形式

当 $n-1$ 时刻数据矩阵的 QR 分解已经求得

$$\mathbf{Q}(n-1) \mathbf{A}_M(n-1) = \begin{bmatrix} \mathbf{R}(n-1) \\ \mathbf{0} \end{bmatrix} \quad (3)$$

加入一个新的数据行时

$$\tilde{\mathbf{A}}_M(n) = \begin{bmatrix} x_1(n-k) & \cdots & x_M(n-k) \\ x_1(n-k+1) & \cdots & x_M(n-k+1) \\ \vdots & \ddots & \vdots \\ x_1(n) & \cdots & x_M(n) \end{bmatrix} = \begin{bmatrix} \mathbf{A}_M(n-1) \\ \mathbf{x}(n) \end{bmatrix} \quad (4)$$

构造一个矩阵 $\mathbf{Q}'(n-1)$, 使得

$$\mathbf{Q}'(n-1) = \begin{bmatrix} \mathbf{Q}(n-1) & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix} \quad (5)$$

则可以得到

$$\mathbf{Q}'(n-1) \tilde{\mathbf{A}}_M(n) = \begin{bmatrix} \mathbf{Q}(n-1) \mathbf{A}_M(n-1) \\ \mathbf{x}(n) \end{bmatrix} = \begin{bmatrix} \mathbf{R}(n-1) \\ \mathbf{0} \\ \mathbf{x}(n) \end{bmatrix} \quad (6)$$

现在, 假设构造一个正交矩阵 $\mathbf{Q}'_n(n)$, 对式(6)右端的矩阵进行 QR 分解。可以利用正交矩阵 $\mathbf{Q}'_n(n) \mathbf{Q}'(n-1)$ 来得到 $\tilde{\mathbf{A}}_M(n)$ 的 QR 分解。另一方面, 构造 $\mathbf{Q}'_n(n)$ 等价于寻找一个正交矩阵, 对 $\begin{bmatrix} \mathbf{R}(n-1) \\ \mathbf{x}(n) \end{bmatrix}$ 做 QR 分解。因为, 式(6)中右端的矩阵中间的零值对构造 $\mathbf{Q}'_n(n)$ 没有影响。这表明当一个新的数据矢量加入后, 可以递推实现 $\tilde{\mathbf{A}}_M(n)$ 的 QR 分解。其中的 \mathbf{R} 矩阵为

$$\tilde{\mathbf{R}}(n) = \begin{bmatrix} \tilde{r}_{11}(n) & \tilde{r}_{12}(n) & \cdots & \tilde{r}_{1M}(n) \\ 0 & \tilde{r}_{22}(n) & \cdots & \tilde{r}_{2M}(n) \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \tilde{r}_{M,M}(n) \end{bmatrix} \quad (7)$$

显然, $r_{ii}(n)$ 会随时间增加而越来越大, 因此往往这种时间递推算法需要加入遗忘因子, 用来防止数据溢出。加入遗忘因子的方法非常简单, 只需将表 1 的计算流程中的 $\tilde{r}_{ii}(n) =$

$$[r_{ii}^2(n-1) + |x_i^{(i)}(n)|^2]^{\frac{1}{2}}, c_i = \frac{r_{ii}(n-1)}{\tilde{r}_{ii}(n)} \text{ 和 } x_j^{(i+1)}(n) = c_i x_j^{(i)}(n) - s_i^* r_{ij}(n-1) \text{ 改为 } \tilde{r}_{ii}(n) = [\tau w r_{ii}^2(n-1) + |x_i^{(i)}(n)|^2]^{\frac{1}{2}}, c_i = \frac{\sqrt{\tau w} r_{ii}(n-1)}{\tilde{r}_{ii}(n)} \text{ 和 } x_j^{(i+1)}(n) = c_i x_j^{(i)}(n) - s_i^* \sqrt{\tau w} r_{ij}(n-1) \text{ 即可, 其中 } 0 < \tau w < 1。$$

表 1 基于 Givens 的 QR 分解时间递推算法^[1]

(n=0)的初始化
$r_{ii}(0) = 0, i = 1, 2, \dots, M$
$r_{ij}(0) = 0, i = 1, 2, \dots, M, j = i+1, \dots, M$
对于每个 n 时刻的初始化
$x_i^{(1)}(n) = x_i(n), i = 1, 2, \dots, M$
do $i = 1, 2, \dots, M$
$\tilde{r}_{ii}(n) = [r_{ii}^2(n-1) + x_i^{(i)}(n) ^2]^{\frac{1}{2}}$
$c_i = \frac{r_{ii}(n-1)}{\tilde{r}_{ii}(n)}$
$s_i = \frac{x^{(i)*}(n)_i}{\tilde{r}_{ii}(n)}$
do $j = i+1, i+2, \dots, M$
$\tilde{r}_{ij}(n) = c_i r_{ij}(n-1) + s_i x_j^{(i)}(n), r_{ij}(n) = \tilde{r}_{ij}(n)$
$x_j^{(i+1)}(n) = c_i x_j^{(i)}(n) - s_i^* r_{ij}(n-1)$
end of loop j
end of loop i

采用 QR 分解的好处在于: 分解过程中计算出来的 $x_M^{(M)}(n)$ 可以直接视为最小二乘误差。 $x_M^{(M)}(n)$ 为先验误差和后验误差的几何平均, 在文献[1]中被定义为角度归一化最小二乘误差, 在大多数情况下先验误差, 后验误差和角度归一化最小二乘误差是等价的^[1]。对于一些只需计算最小二乘误差的应用, 该算法非常适用。

3 基于 Givens 的 QR 分解滑窗递推形式

若要得到 n 时刻滑窗长度为 k 的数据矩阵的 QR 分解, 只需要去掉 $n-k$ 时刻的数据矢量的影响即可。若要消除 n 时刻的数据矢量的影响, 只需要进行 Givens 旋转的逆过程即可。而对于一个 $n-k$ 时刻的输入数据矢量, 只须证明输入顺序的变换对 \mathbf{R} 矩阵不产生影响, 就可以将 $n-k$ 时刻的数据当作 n 时刻输入的数据, 在 n 时刻正推过程以后, 采用 Givens 的逆过程消除其对 \mathbf{R} 阵的影响。

对于数据矩阵 $\tilde{\mathbf{A}}_M(n)$, 将其一行调换位置

$$\tilde{\mathbf{B}}_M(n) = \mathbf{T} \tilde{\mathbf{A}}_M(n) = \begin{bmatrix} x_1(n-k+1) & \cdots & x_M(n-k+1) \\ \vdots & \ddots & \vdots \\ x_1(n) & \cdots & x_M(n) \\ x_1(n-k) & \cdots & x_M(n-k) \end{bmatrix} \quad (8)$$

式中

$$T = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \ddots & \vdots \\ \vdots & \vdots & 0 & \ddots & 0 \\ 0 & 0 & \cdots & \ddots & 1 \\ 1 & 0 & \cdots & \cdots & 0 \end{bmatrix} \quad (9)$$

若 $\tilde{Q}(n)\tilde{A}_M(n) = \begin{bmatrix} \tilde{R}(n) \\ \mathbf{0} \end{bmatrix}$, 其中 $\tilde{Q}(n)$ 为正交矩阵,

$\tilde{R}(n)$ 为上三角方阵, 则 $B_M = TA_M = T\tilde{Q}^H(n) \begin{bmatrix} \tilde{R}(n) \\ 0 \end{bmatrix}$, 显然,

$T\tilde{Q}^H(n)$ 仍然是正交矩阵, 由 QR 分解的唯一性可知, 矩阵行向量交换顺序后进行 QR 分解, 分解的结果对 $\tilde{R}(n)$ 没有影响。

对 $\tilde{B}_M(n)$ 的 QR 分解, 可以看作是对 $\begin{bmatrix} A_M(n) \\ \mathbf{x}(n-k) \end{bmatrix}$ 或

$\begin{bmatrix} R(n) \\ \mathbf{x}(n-k) \end{bmatrix}$ 的 QR 分解, 其分解后的 R 阵为 $\tilde{R}(n)$, 而从

$\begin{bmatrix} R(n) \\ \mathbf{x}(n-k) \end{bmatrix}$ 进行 QR 分解, 可以套用前面提到的 Givens 算法

实现。可见, 要得到 $A_M(n)$ 的 QR 分解, 只需将 $\tilde{B}_M(n)$ 的 QR 分解中数据向量 $\mathbf{x}(n-k)$ 的影响消除即可。要消除 $\mathbf{x}(n-k)$ 在 $\tilde{B}_M(n)$ 的 QR 分解中的影响, 只需做 Givens 算法的逆过程, 按表 2 操作即可。

表 2 SWG-RLS 的一步消除

对每个 n 时刻的初始化 ($n > k$)

$$x_i^{(1)}(n) = x_i(n-k), i=1, 2, \dots, M$$

do $i=1, 2, \dots, M$

$$r_{ii}(n) = [\tilde{r}_{ii}^2 2(n) - |x_i^{(i)}(n)|^2]^{\frac{1}{2}}$$

$$\tilde{c}_i = \frac{\tilde{r}_{ii}(n)}{r_{ii}(n)}$$

$$\tilde{s}_i = \frac{x_i^{(i)*}(n)}{r_{ii}(n)}$$

do $j=i+1, i+2, \dots, M$

$$r_{ij}(n) = \tilde{c}_i \tilde{r}_{ij}(n) - \tilde{s}_i x_j^{(i)}(n)$$

$$x_j^{(i+1)}(n) = \frac{x_j^{(i)}(n)}{\tilde{c}_i} - \frac{\tilde{s}_i^* r_{ij}(n)}{\tilde{c}_i} = \tilde{c}_i x_j^{(i)}(n) - \tilde{s}_i^* r_{ij}(n)$$

end of loop j

end of loop i

4 SWG-RLS 的 CORDIC 实现

实数输入 (x, y) 通过旋转得到 $(\sqrt{x^2+y^2}, 0)$ 的形式, 可以由式(10)表示其计算过程。

$$(x_{i+1}, y_{i+1}) = \cos \phi_i (x_i + \rho_i 2^{-i} y_i, y_i - \rho_i 2^{-i} x_i) \quad (10)$$

式中, $\rho_i = \pm 1, \phi_i$ 是使 $\tan \phi_i = 2^{-i}$ 的特殊角度。

同样, 实数输入 (x, y) 通过旋转得到 $(\sqrt{x^2-y^2}, 0)$ 的形式, 则可以采用双曲 CORDIC 计算。

$$(x_{i+1}, y_{i+1}) = \cosh \phi_i (x_i - \rho_i 2^{-i} y_i, y_i - \rho_i 2^{-i} x_i) \quad (11)$$

无论普通的 CORDIC 还是双曲 CORDIC 都可以采用图 1 的流水线结构用加法器实现。Givens 旋转可采用普通的 CORDIC 实现, 而 Givens 逆旋转可以用双曲 CORDIC 实现。

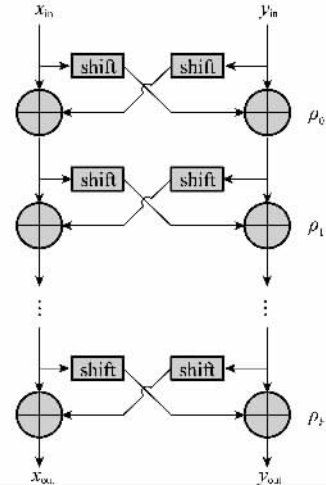


图 1 流水线 CORDIC 结构

复数输入的 Givens 旋转可以通过 3 个实数的 Givens 旋转模块的组合实现^[1], SWG-RLS 方法可以采用图 2 所示的方法实现, 其中方框表示复数 GIVENS 旋转模块, 八角形表示的是双曲 CORDIC 实现的复数 Givens 逆旋转模块。

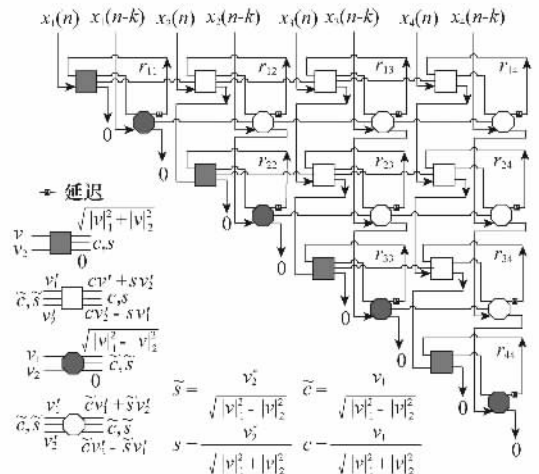


图 2 SWG-RLS 方法的信号流图

5 仿真实验

仿真实验以卫星导航抗干扰系统为背景。在卫星导航信号抗干扰中, 最小二乘权值并不是感兴趣的量, 卫星导航信号为扩频信号, 由于卫星到地球表面的距离遥远, 接收到

的卫星导航信号功率一般远低于接收机热噪声,只有在解扩之后才能被检测,故在进行全球导航卫星系统(global navigation satellite system, GNSS)抗干扰自适应处理时,能够被消除的只有高于热噪声的干扰信号,而卫星的扩频信号将在干扰抑制以后保留在误差信号之中,因此,在这种应用中,感兴趣的量是误差信号(或剩余信号)。

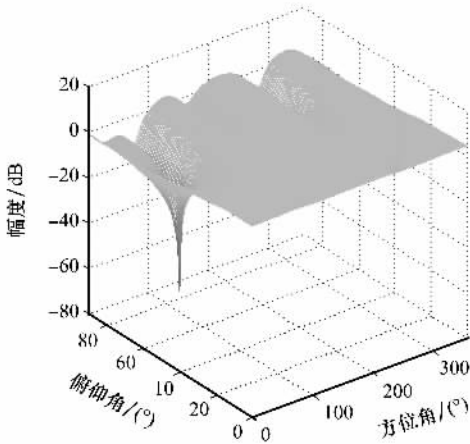
根据 ICD-GPS-200 标准,在单位增益 RHCP(右旋圆极化)天线上 GPS 用户的最低接收功率为 -159.6 dBW,热噪声密度为 -205 dBW/Hz, C/A 码接收机处理带宽为一般为 2 MHz,带内热噪声功率约为 -139 dBW,信噪比约为 -20 dB。仿真实验中就取 $SNR = -20$ dB。GPS 的人为干扰可以轻易达到干噪比 $INR = 30$ dB,总的信干噪比 $SINR = -50$ dB。

仿真的天线阵列均采用 3×3 的 9 阵元方阵,相邻阵元间的距离为 GPS 卫星 L1 波段的半波长 $\lambda/2$ 。

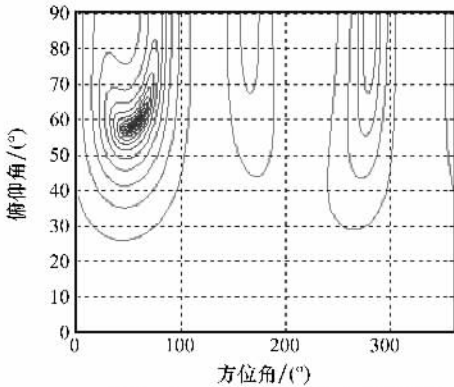
5.1 SWG-RLS 方法抗干扰处理后的方向图

这一部分的实验假设视场内存在一个宽带干扰,信干噪比 $SINR = -50$ dB,干扰入射的方位角和俯仰角均为 60° 。

从图 3 可以看出,SWG-RLS 方法能够在正确的干扰来向上形成自适应零陷,从零陷形成角度上,可以看出 SWG-RLS 方法的有效性。



(a) SWG-RLS方法形成的方向图



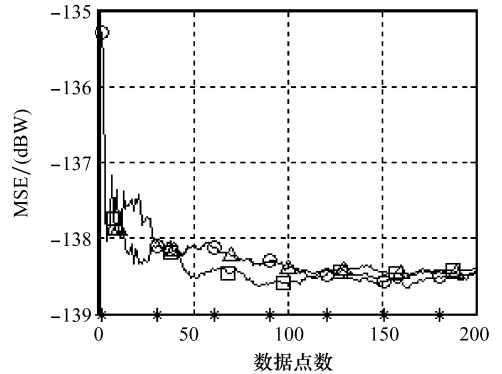
(b) SWG-RLS方法形成的方向图俯视图

图 3 SWG-RLS 方法形成的自适应方向图

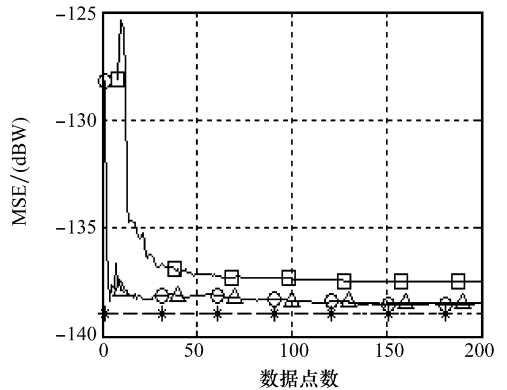
5.2 收敛性能比较

这一部分的实验假设视场内存在一个连续宽带干扰,信干噪比 $SINR = -50$ dB,干扰入射的方位角和俯仰角均为 60° 。

分别采用不同的计算精度条件下,QR-RLS、SWG-RLS 和 DMI 方法的收敛性能比较,收敛后的 MSE 越大,性能越差,在极高精度条件下,QR-RLS、SWG-RLS 和 DMI 方法具有相似的收敛性能,而当计算精度下降时,图 4(b) 所示在 32 位定点精度条件下 DMI 方法明显性能下降,QR-RLS 方法和 SWG-RLS 方法基本没有性能上的下降。



(a) 在极高精度下不同方法的收敛性能



(b) 在32位定点精度条件下不同方法的收敛性能

△: QR-RLS; ○: SWG-RLS; □: DMI;
*: 噪声背景输出功率为 -139 dBW

图 4 不同方法收敛性能比较

5.3 不同干扰环境条件下 QR-RLS 和 SWG-RLS 性能比较

这一部分实验假设视场内存在某颗 GPS 卫星,方位角为 30° ,俯仰角为 20° 。仿真中取信噪比 $SNR = -20$ dB。接收数据采样率为 2.046 M,即一个 C/A 码片采两个点。

当只存在一个连续宽带干扰,信干噪比 $SINR = -50$ dB,干扰入射的方位角和俯仰角均为 60° 时,预检测积分采用非相关累积,积分长度为 5 个 C/A 码周期。两种方法抗干扰处理后得到的相关峰如图 5 所示。

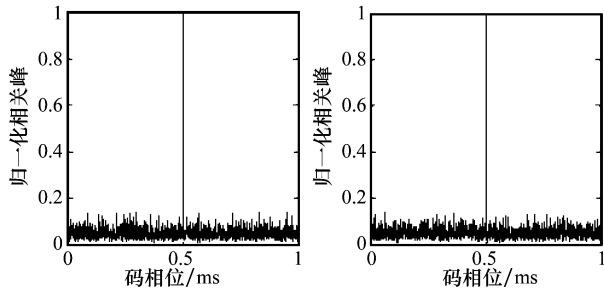


图 5 一个连续宽带干扰条件下两种方法抗干扰处理后的预检测积分结果

在地平面附近设置不同周期的脉冲形式的宽带干扰,信干噪比 $\text{SINR} = -50 \text{ dB}$, 干扰个数为 6, 方位角依次为 $10^\circ, 70^\circ, 130^\circ, 190^\circ, 250^\circ, 310^\circ$; 俯仰角均为 2° 。两种方法抗干扰处理后得到的相关峰如图 6 所示。

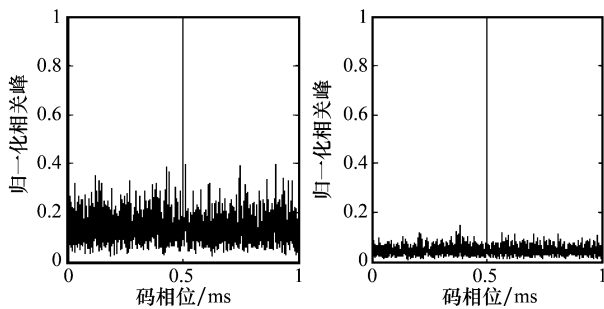


图 6 复杂干扰条件下两种方法抗干扰处理后的预检测积分结果

从处理结果看,在连续形式的干扰条件下,采用滑窗式处理方法或采用传统的遗忘因子处理方法,具有相似的处理效果,但是在干扰形式复杂的条件下,采用滑窗处理方法的性能明显优于非滑窗形式的处理方法,得到的相关峰更容易被检测。

6 结论

SWG-RLS 算法可以应用于需要滑窗式样本输入的信号处理系统,由于该方法是基于 Givens 的 QR 分解方法,不仅避免了最小二乘算法中的求逆问题,且具有较好的数值特性和模块化结构,使得算法可以很方便的映射到 systolic 阵进行并行流水处理,能适应很高的数据吞吐率。在现代战争环境中,GNSS 往往受到各种有意无意的干扰,各种雷达发射的往往是高功率的脉冲信号,电子战飞机往往采用各种复杂的干扰形式,在 GNSS 抗干扰的应用中,SWG-RLS 方法比传统的 QR-RLS 方法更能适应复杂电磁环境。

参考文献:

- [1] Proakis G J. *Algorithms for statistical signal processing*[M]. New Jersey: American Prentice-Hall, 2002.
- [2] Gao L J, Parhi K K. Hierarchical pipelining and folding of QRD-RLS adaptive filters and its application to digital beam forming[J]. *IEEE Trans. on Circuits and Systems-II: Analog and Digital Signal Processing*, 2000,47(12):1503 - 1519.
- [3] Wu C S, Wu A Y. Modified vector rotational CORDIC (MVR-CORDIC) algorithm and architecture[J]. *IEEE Trans. on Circuits and Systems II: Analog and Digital Signal Processing*, 2001,48(1):548 - 561.
- [4] Wu C S, Wu A Y, Lin C H. A high-performance/low-latency vector rotational CORDIC architecture based on extended elementary angle set and trellis-based searching schemes[J]. *IEEE Trans. on Circuits and Systems-II: Analog and Digital Signal Processing*, 2003,50(9):589 - 601.
- [5] Park S Y, Cho N I. Fixed-point error analysis of CORDIC processor based on the variance propagation formula[J]. *IEEE Trans. on Circuits and Systems I: Fundamental Theory Regular Papers*, 2004,51(3):573 - 584.
- [6] Gomes J, Barroso V A N. Array-based QR-RLS multichannel lattice filtering[J]. *IEEE Trans. on Signal Processing*, 2008, 56(8):3510 - 3522.
- [7] Meher P K, Valls J, Juang T B. 50 years of CORDIC: algorithms, architectures, and applications[J]. *IEEE Trans. on Circuits and Systems I: Regular Papers*, 2009, 56(9): 1893 - 1907.
- [8] Maltsev A, Pestretsov V, Maslennikov R. Triangular systolic array with reduced latency for QR-decomposition of complex matrices[C]//*Proc. of IEEE International Symposium on Circuits and Systems*, 2006:244 - 256.
- [9] Boppana D, Dhanoa K, Kempa J. FPGA based embedded processing architecture for the QRD-RLS algorithm[C]//*Proc. of 12th Annual IEEE Symposium on Field-Programmable Custom Computing Machines*, 2004:330 - 331.
- [10] Jun M, Parhi K K, Deprettere E F. Annihilation-reordering look-ahead pipelined CORDIC-based RLS adaptive filters and their application to adaptive beamforming[J]. *IEEE Trans. on Signal Processing*, 2000,48(8):2414 - 2431.
- [11] Hamill R, Mccanny J V, Walke R L. Online CORDIC algorithm and VLSI architecture for implementing QR-array processors[J]. *IEEE Trans. on Signal Processing*, 2000,48(2): 592 - 598.