

文章编号:1002-8684(2006)02-0055-03

# 基于简化型 RLS 算法的延迟改进型 噪声抵消系统模型与实现

· 论文 ·

殷文铮, 杜 旭

(华中科技大学 电子与信息工程系, 湖北 武汉 430074)

**【摘要】**传统 RLS 算法由于复杂的矩阵运算而无法满足实时性要求, 在传统 RLS 算法的基础上, 提出基于简化型 RLS 算法的延迟改进型噪声抵消系统模型, 并将此模型应用于 TMS320C5509 DSP 的语音处理平台上实时实现。实验证明, 该模型滤噪性能较普通的 LMS 自适应滤波器有显著提高, 且可满足实时性要求。

**【关键词】**简化型 RLS 算法; LMS; TMS320C5509; 自适应噪声抵消

**【中图分类号】**TN912

**【文献标识码】**A

## An Improved Delay Model and Realization of Noise Cancellation System Based on Simplified RLS Algorithm

YIN Wen-zheng, DU Xu

(Department of Electronics &amp; Information Engineering,

Huazhong University of Science &amp; Technology, Wuhan 430074, China)

**【Abstract】**The normalized RLS algorithm has the drawback of complicated matrix computations and cannot satisfy the real-time request. This paper presents an improved delay model of noise cancellation system based on simplified RLS algorithm. By realizing the model on the platform of speech processing system which uses TMS320C5509 DSP chip, it is proved that the performance of this model is obviously superior to that of the LMS algorithm, and can satisfy the real-time request.

**【Key words】**simplified RLS algorithm; LMS; TMS320C5509; adaptive noise cancellation

## 1 引言

目前, 自适应信号处理技术已经广泛应用于自适应均衡、自适应回波抵消、自适应噪声抵消等领域。在自适应信号处理算法中, 作为一种最小二乘法的快速算法, 递归最小二乘(RLS)算法由于收敛速度快且性能优于 LMS 算法而得到了广泛的应用。然而, 此算法迭代计算量较大, 因此应用于实时自适应噪声抵消实现时受到一定限制。笔者着重介绍简化型 RLS 算法及其在延迟改进型噪声抵消模型中的实现。基本型自适应滤波器以带噪信号作为输入, 以与带噪信号中噪声分量相关的噪声信号作为参考输入, 因此需要单独设置噪声采集通道。但在许多噪声抵消系统的实际应用中, 由于条件限制往往无法采用双传声器输入方案, 从而难以获取与噪声相关的参考噪声输入。笔者采用延迟改进型噪声抵消模型<sup>[1]</sup>, 自适应滤波算法采用改进型 RLS 算法, 如图 1 所示。

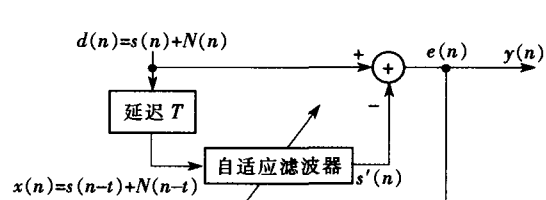


图 1 延迟改进型自适应滤波器

其中,  $s(n)$  为纯净的语音信号,  $N(n)$  为噪声信号,  $d(n)$  则为带噪语音信号,  $y(n)$  为噪声信号的估计值, 而  $s'(n)$  为  $s(n)$  的估计值。此模型的原理是将带噪信号延迟一个周期作为参考输入, 由于信号相关性较强,  $s(n)$  与  $s(n-t)$  的相关性就较大, 在噪声相关性较弱(如白噪声)时, 即  $N(n)$  与  $N(n-t)$  相关性较小时, 输出信号  $s'(n)$  就是降噪后  $s(n)$  的估计值。对于语音信号而言, 虽然其具有时变特性, 但在一个短时间范围内(一般认为在 10~30 ms 的短时间内), 其特性基本保持不变, 因而可在这样的短段时间内将其看作一个准稳态过程<sup>[1]</sup>。

将语音信号的特性应用于该模型中,即周期取值为10~30 ms,  $t$  的取值由延时时间  $T$  (一般取一个周期的时间)与采样频率获取。

## 2 算法原理

RLS 算法<sup>[2]</sup>以最小误差平方和为优化目标,这里的误差即图 1 中自适应滤波器的期望响应  $d(n)$  与滤波器输出  $y(n)$  的差值。RLS 算法确定 FIR 滤波器权系数所依据的标准是

$$\varepsilon(n) = \sum_{k=0}^n e^2(k) = \sum_{k=0}^n |d(k) - y(k)| = \min \quad (1)$$

对于非平稳输入信号,为能更好地进行跟踪,引入遗忘因子  $\lambda$  修正式(1)得到

$$\varepsilon(n) = \sum_{k=0}^n \lambda^{n-k} e^2(k) \quad (2)$$

按照最小化准则决定最佳权系数矢量  $w$ : 将  $\varepsilon(n)$  对  $w$  求导并令其等于 0, 得到

$$\frac{\partial \varepsilon(n)}{\partial w} = -2 \sum_{k=0}^n \lambda^{n-k} e(k) x(k) = 0 \quad (3)$$

整理得到

$$\left[ \sum_{k=0}^n \lambda^{n-k} x(k) x^T(k) \right] w = \sum_{k=0}^n \lambda^{n-k} d(k) x(k) \quad (4)$$

将算法写成按  $n$  进行迭代计算的形式为

$$k(n) = \frac{\lambda^{-1} P(n-1) x(n)}{1 + \lambda^{-1} x^T(n) P(n-1) x(n)} \quad (5)$$

$$y(n) = w(n-1) x(n) \quad (6)$$

$$e(n) = d(n) - y(n) \quad (7)$$

$$w(n) = w(n-1) + k(n) e(n) \quad (8)$$

$$P(n) = \lambda^{-1} P(n-1) - \lambda^{-1} k(n) x^T(n) P(n-1) \quad (9)$$

$w(n)$  根据  $(n-1)$  时刻及以前数据得到的最佳权系数  $w(n-1)$  来预测, 增益  $k(n)$  用于  $w(n)$  的迭代计算。在 RLS 算法中, 各式迭代计算的乘除运算 MADPR (Multiplications and Divisions per Recursion)<sup>[3]</sup> 的次数统计如下:  $k(n)$  为  $(N^2+2N)$  次;  $y(n)$  为  $N$  次;  $e(n)$  为 0 次;  $w(n)$  为  $N$  次;  $P(n)$  为  $2N^2$  次; RLS 算法的总计算量为  $(3N^2+4N)$  次。其中  $N$  为滤波器阶数。可见, 在 RLS 算法中计算量最大的部分是对  $k(n)$  和  $P(n)$  的计算, 其所包含的矩阵乘除运算的计算量为  $N^2$  级, 计算量随着  $N$  的变大呈平方倍增长, 这在实际应用中会引起较大的延时。此外, 遗忘因子  $\lambda$  和初值  $w(0)$ ,  $P(0)$  也将影响算法性能和实时性

## 3 算法简化与实时实现

据以上分析, 将 RLS 算法应用于基于 TMS320C5509 DSP 芯片的延迟改进型噪声抵消系统时, 为保证算法的实时性, 需在保证算法性能的基础上对 RLS 算法进行相应的优化。

TMS320C5509 DSP 芯片是定点 DSP 芯片, 并且其指令集中不包括算术除法运算。所有的除法运算必须通过减法运算来实现, 这不仅在实现上非常复杂, 而且计算量非常大。为保证算法的实时性, 将 RLS 算法中对  $k(n)$  的计算进行一定的简化。设  $\lambda$  远小于  $P(n-1) \cdot x(n)$ , 即  $\lambda^{-1} x^2(n) P(n-1)$  远大于 1, 那么  $k(n)$  的分母部分可以近似为  $\lambda^{-1} x^2(n) P(n-1)$ , 对  $k(n)$  的计算可以简化为  $k(n) = 1/x(n)$ , 从而使  $k(n)$  计算的 MADPR 从  $(N^2+2N)$  降低到  $N$ , 总 MADPR 从  $(3N^2+4N)$  降低到  $(2N^2+3N)$ 。在对  $P(n)$  的计算中, 限于 DSP 芯片的计算精度,  $k(n)$  不会精确等于  $x(n)$  的倒数, 因此计算出的  $P(n)$  接近于 0 但不等于零。经过以上简化, 算法的实现更简洁, 处理速度更快。

影响算法处理时间和算法性能的最重要参数是阶数  $n$ 。显然当阶数较大时滤波效果较好, 但这会大大增加算法的处理时间。可以由简化型 RLS 算法的总 MADPR 表达式算出,  $N$  每增加 1, MADPR 增加  $(4N+5)$ , 因此  $N$  取值过大往往不能满足实时性要求。实际应用中需根据系统允许的时延, 对效果和实时性进行权衡之后, 相应地选取阶数范围。

RLS 算法中, 影响算法收敛性的参数  $\lambda$  通常的取值范围通常在 0.95~1 之间。 $\lambda$  的值越小, 收敛速度越快。对于平稳信号,  $\lambda$  的最佳值为 1。由于语音信号是短时 (10~30 ms 内) 准稳态过程, 在短时时段内其物理特性和频谱特性可以近似的看作是稳定不变的。因此, 短时时段内可以认为语音信号是平稳随机过程, 从而可采用线性时不变模型, 即  $\lambda$  可取值为 1<sup>[4]</sup>。

## 4 算法实验结果与性能比较

将上述自适应噪声抵消模型应用于 TMS320C5509 DSP 芯片语音处理系统中, 试验证明以上假设有利于算法的实时实现。

采用简化型 RLS 算法的延迟改进型噪声抵消系统对语音信号进行处理, 其输入带噪信号波形见图 2, 输出波形见图 3; 采用标准 RLS 算法, 其输出波形如图 4 所示 (图 2~图 4 表示了 1~12 000 个采样点内的信

号波形图)。由实际测量得到,当阶数取 10 时,未经简化的 RLS 算法所需的 DSP 时钟周期为 198 764, 耗时为 1.38 ms, 简化型 RLS 算法所需时钟周期为 154 732, 耗时为 1.07 ms。与标准 RLS 算法相比,简化型 RLS 算法耗时减少了 0.31 ms。两者效果相比,简化型 RLS 对信噪比的提高比标准 RLS 算法要高。造成此结果的原因源于标准 RLS 算法计算过程复杂,且 TMS320C5509 是定点 DSP 芯片,由定点运算引入的误差反而更大。

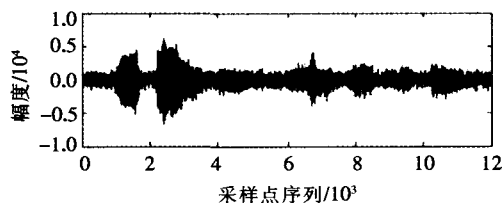


图 2 输入带噪信号波形图

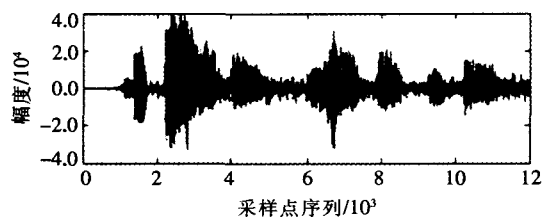


图 3 简化型 RLS 算法的延迟改进型噪声抵消系统输出波形图

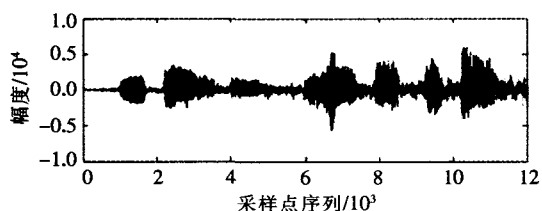


图 4 标准 RLS 算法的延迟改进型噪声抵消系统输出波形图

不同阶数的简化型 RLS 算法性能比较见表 1, 其中输入信号与图 2 所示的输入信号相同。当阶数为 32 阶时,算法耗时为 1 022 060 个时钟周期,即延时为 7.1

表 1 不同阶数简化型 RLS 算法性能比较

阶数 NH	算法 MADPR	算法耗时 (时钟周期)	输出波形图
10	230	154 732	
16	560	317 548	

ms, 由于延时过大,在多输入复杂应用的场合,无法实时实现,因而不将对 32 阶的输出性能做比对。阶数为 16 时,算法耗时是阶数为 10 时的 2 倍,而效果与 10 阶时相近。

将该模型与采用延时 LMS 算法(采用 C55x DSP 库提供的 Delay LMS 算法<sup>[4]</sup>)的延迟改进型噪声抵消系统性能比较如表 2 所示,其中输入信号与图 2 所示的输入信号相同。延时 LMS 算法耗时为 3 403 个时钟周期,简化型 RLS 算法耗时为 154 732 个时钟周期。比较两种算法耗时,简化型 RLS 算法仍远大于延时 LMS 算法。单从输出波形效果上看,延时 LMS 算法对信噪比的提高显然不及简化型 RLS 算法。

表 2 简化型 RLS 算法与 LMS 算法性能比较

算法	阶数	算法耗时 (时钟周期)	输出波形图
简化型 RLS	10	154 732	
LMS	10	3 403	

表 1 及表 2 中的输出波形图为输出信号图 2 在第 1~12 000 个采样点内的波形图。其中横坐标为采样点序列号,纵坐标为信号幅度。

## 5 总结

笔者所述的采用简化型 RLS 算法的延迟改进型噪声抵消系统已在 TMS320C5509 DSP 平台的语音处理系统中得到应用,其性能和实时性(算法时延在 1 ms 左右)均较好。可见,兼顾算法精确度和实时性的基础上对 RLS 算法进行一定的优化,则 RLS 算法完全可适用于实时性要求苛刻的环境。

### 参考文献

- [1] 赵力. 语音信号处理[M]. 北京:机械工业出版社,2003.
- [2] 姚天任,等. 现代数字信号处理[M]. 武汉:华中理工大学出版社,1999.
- [3] 何振亚. 自适应信号处理[M]. 北京:科学出版社,2002.
- [4] LONG G Z, LING F Y, PROAKIS J G. The LMS algorithm with delayed coefficient adaptation[J]. IEEE Trans on Signal Processing, 1989,37(9):1 397-1 405.

[收稿日期] 2005-10-18