

编号: 20172013211445

审定成绩: _____

重庆邮电大学 毕业设计（论文）

中文题目	自适应投影仿射滤波算法及应用研究
英文题目	Research on the Affine Projection-type Adaptive Filtering Algorithms and Applications
学院名称	通信与信息工程学院
学生姓名	刘亚林
专业	通信工程
班级	0111307
学号	2013211445
指导教师	周 翊 教授
答辩组 负责人	叶志红 讲师

2017 年 05 月

重庆邮电大学教务处制

摘要

用电子器件组合成的滤波器对电磁波信号进行功能性的处理已成为当今通信和其他科学前沿技术的基础，为使滤波器能处理多样化的信号以及不确定模型的系统，人们提出了能随一种能迭代算法实时更新系数的自适应滤波器。这个迭代算法即是一个自适应滤波器的自适应算法。已经提出并得以应用的自适应滤波算法有：简单且易于实现的最小均方算法(LMS, Least Mean Square)、其改进算法有归一化的最小均方算法(NLMS, Normalized Least Mean Square)以及通过利用多个输入向量来更新权值的投影仿射算法(APA, Affine Projection Algorithm)。

本文重点研究基于投影仿射的自适应算法，已有的对 APA 算法的改进算法很多，比如变正则因子的 APA 算法、结合凸组合技术的 APA 算法以及结合子带自适应滤波的 APA 算法等等，均可提升 APA 算法在收敛速度和稳态误差上的性能。针对 APA 在不能解决非高斯的冲激性噪声的问题，已提出的基于 l_1 范数的投影仿射符号算法 APSA(Affine Projection Sign Algorithm)可以解决。另外，针对 APA 对大相关性的输入信号处理性能衰减的问题，已提出的结合 SAF 算法的 APA 算法 IMSAF 可以很好的解决它。本文结合投影仿射符号算法和子带自适应滤波器的结构，探索出了一种可解决高相关性输入信号时滤波问题，并可以对抗冲激性噪声干扰的基于 APSA 和 SAF 的自适应滤波算法，即子带投影仿射符号算法(SAPSA, Sub-band Affine Projection Sign Algorithm)。从算法对声学回声消除的仿真结果上，也可以对比已有的 APSA、NSAF 和 IAPSAF(Combined Step Size Improved Affine Projection Sub-band Adaptive Filtering)算法，可以看出我们的算法在作针对性处理大相关性输入信号和冲激性噪声去噪上面具有更优的性能。

关键词： 自适应滤波，投影仿射自适应滤波算法，子带自适应滤波算法，投影仿射符号算法，MATLAB

Abstract

The filter which is usually composed by the electronic devices and is used to do some functional signal process, has been the basic technology of today's telecommunications and other scientific frontier technology. To make these filters able to process different and variable signal and make them have stronger adaptive capability to the variable system, some researchers began to propose the adaptive filter which could update its coefficients by an iterative algorithm in real-time. And the iterative algorithm is the adaptive algorithm for one adaptive filter. There have been some common adaptive algorithm in application, such as LMS(Least Mean Square) algorithm which is easy to implement, its improved algorithm NLMS(Normalized Least Mean Square) and APA(Affine Projection Algorithm) which is the basic researching algorithm in this paper.

The key part of this paper is researching some algorithms which not only is based on APA but also can arrive more targeted performances. There are a lot of pre-existing improved APA algorithm which were more advanced than APA from its tracking performances, such as the APA algorithm based on variable regulation factor or variable step. The pre-existing APSA(Affine Projection Sign Algorithm), which is based on the can L1-norm optimization criteria, can process the non-Gaussian impulse noise. Moreover, the pre-existing APSAF(Affine Projection Sub-band Adaptive Filtering) algorithm, which combined the APA and the SAF(Sub-band Adaptive Filtering), can process the correlated input signal. This paper combined the APSA and the Sub-band adaptive filter, and proposed a new adaptive algorithm---S-APSA(Sub-band Affine Projection Sign Algorithm), which could process the input-signal with large correlation value and the non-Gaussian impulse noise. The simulations in the acoustic echo cancellation context confirmed that our algorithm's thought was right. Moreover, compared with the simulations of the pre-existing algorithm APSA and IAPSAF(Combined Step Size Improved Affine Projection Sub-band Adaptive Filtering), the proposed algorithm performed better from processing the correlated input signal and the non-Gaussian impulse noise.

Keywords: the adaptive filter, Affine Projection Adaptive Algorithm, Sub-band adaptive filter, Affine Projection Sign Algorithm, MATLAB

目录

摘要	I
目录	III
第 1 章 绪论	1
1.1 自适应算法概述	1
1.1.1 滤波	1
1.1.2 自适应滤波	2
1.1.3 自适应滤波算法	3
1.2 国内外研究动态	4
1.2.1 从 LMS 发展到 APA	5
1.2.2 NSAF 算法的提出及其改进算法	5
1.2.3 NSAF 结合 APA 算法的提出	6
1.2.4 APSA 算法的提出	6
1.2.5 本文的研究方向	6
1.3 章节安排	7
第 2 章 常用自适应算法分析	8
2.1 LMS 算法和 NLMS 算法	8
2.2 APA 算法	10
2.3 NSAF 算法	11
2.4 APSA	12
2.5 本章小结	13
第 3 章 子带投影仿射符号算法 SAPSA	15
3.1 算法推导	15
3.2 算法的收敛性分析	17
3.3 算法仿真	18
第 4 章 自适应滤波算法及 APA 类算法的应用研究	24

4.1 自适应滤波器的四种应用	24
4.1.1 系统辨识.....	24
4.1.2 逆模型.....	25
4.1.3 信号预测.....	25
4.1.4 干扰消除.....	26
4.2 自适应投影仿射类算法的应用	27
4.2.1 回声消除.....	27
4.2.2 电子回声消除.....	28
4.2.3 数字预失真系统.....	29
4.3 本章小结.....	30
第 5 章 总结与改进.....	31
5.1 主要工作总结和创新点.....	31
5.1 改进方面和后续研究展望.....	31
参考文献.....	33
致谢.....	37
附录 A 算法仿真 MATLAB 代码.....	38
附录 B 英文翻译	49
附录 C 英文原文	65

第 1 章 绪论

本绪论将对本课题的研究背景及意义、国内外研究动态、本篇论文的研究内容和章节安排四个方面进行一个简单的介绍。在研究背景及意义这一个上面，本文其用 1.1 小节的自适应滤波概述来进行说明。由于本文所要研究的自适应滤波投影仿射算法及应用是属于自适应滤波算法下的投影仿射类研究，因此我认为有必要详尽写清楚滤波、自适应滤波以及自适应滤波算法它们之间的关系，才能帮助更好的阐明本论文的研究意义，同时也作为本次课题研究的一部分。

而本课题研究的是自适应投影仿射算法及应用研究，可以将这个课题分解成三个研究部分：一是自适应算法的学习研究；二是投影仿射与自适应算法的改进研究；三是投影仿射类算法的应用研究。为更系统地了解自适应算法的用途和来源，本章将从滤波到自适应滤波算法之间的发展和联系来介绍本课题的研究背景。本章的 1.1 节以及 1.2 节将会完成第一个研究部分——对自适应算法进行学习研究。我将从滤波到自适应滤波算法的知识基础进行概括性的介绍说明。

1.1 自适应算法概述

自适应滤波是用来适应不确定性的输入信号的滤波，自适应算法是用来得到自适应滤波中滤波器权值收敛更新的算法。

1.1.1 滤波

顾名思义，波即电磁波信号，滤即过滤，所以滤波可以简单地理解为对电磁波信号的过滤，可见滤波是一个对电磁波信号进行过滤的过程。可怎么实现这个过程呢？便是通过一堆可以用来达到我们期望的滤波效果的电子器件，如电感、电容、二极管等等来实现。由于当前用来进行数据通信的大多是电磁波，所以我们可以将电磁波信号简称为信号。因此滤波这个过程可以这样来定义：滤波是通过一些特意组合的电子器件过滤已知信号得到期望信号的过程。

将实际应用的滤波过程进行一定的假设和简化，滤波器在时域上的数学模型可以用其对应的单位冲激响应来表示。有限长单位冲激响应滤波器即 FIR(finite

impulsive response)滤波器是一种常见结构的滤波器，又称为非递归型滤波器，由于其只用到了前向路径而使其具有固定的稳定性，而与其对应的 IIR(infinite impulsive response)滤波器中具有反馈路径，除非经过合理的设计，不然滤波器反馈会使的输出结果不稳定。

然而一般的 FIR 滤波器作为一个线性组合电路，其参数往往是固定的，这在实际应用中处理不确定性信号的效果并不尽如人意，因为它只能解决某一类信号的滤波问题，而遇到其他滤波问题时，往往起不到期望的滤波作用。因此一类可以改变参数的滤波器被提出了，由于其设计原理一般是根据其输入信号的变化来调整滤波系数的，所以被称为自适应滤波，以下研究算法均是基于 FIR 滤波器结构的滤波器权值洗漱更新来开展的。

1.1.2 自适应滤波

自适应滤波就是对输入信号的不确定性有更强的自适应处理能力的滤波器，因此可以定义：自适应滤波就是通过某种算法自适应地调整滤波器参数以适应输入信号与期望信号之间不确定的系统或信号以达到最优性能的滤波过程。

毕竟实际的应用过程中得到的信号都是充满各种不确定性的，有经过无线传输被复杂环境引入加性的高斯噪声或者冲激性的不稳定的噪声，比如移动通信系统中复杂多变的传输环境；也有期望信号本身就是一种冲激性的不平稳信号，比如人声语音。这些不确定的系统或信息导致了输入信号和期望信号之间不可能通过一个简单的线性滤波或者固定模型就能实现。

而自适应滤波技术采用了实时迭代的算法，更新滤波器系数，实时模拟出最适合从输入信号得到期望信号的滤波器模型。根据这个观点，自适应滤波器的应用体现在 4 个方面：系统辨识（对不确定性的系统进行自适应的模拟和辨识）、逆模型（得到未知系统传递函数的倒数）、预测未知信号或噪声估计（对随机信号或者噪声的当前值提供某种意义上的一个最好预测）和干扰消除（消除接收信号的未知干扰，常见的应用是回波消除）。

如图 1.1，自适应滤波系统一般包括三个模块：滤波结构，性能判据和自适应算法。

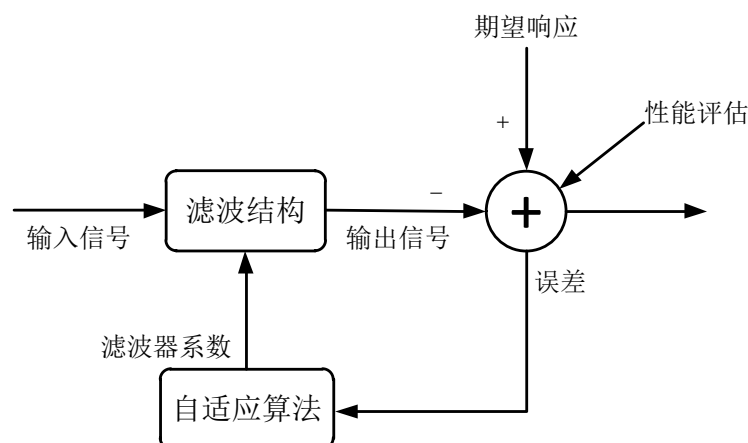


图 1.1 自适应滤波系统架构

现在分别对自适应滤波系统进行介绍。滤波结构就是实现滤波的电子器件的组合，这类结构由滤波器设计者决定，用数学模型表示就是时域或者变换域的系统函数，这个函数的一些参数比如滤波器系数是可以根据自适应算法进行更新的，一般用 FIR 滤波器的结构来进行研究（以下算法对滤波器权值矢量的更新均是基于 FIR 滤波器的结构）。性能判据是用来判断一个自适应滤波系统质量的标准，比如用自适应滤波器的输出信号与期望相应做差值得误差，使误差值最小时作为自适应滤波系统质量的评判，也可以用其差值的平方或者一段时间内 L 个差值平方的期望最小作为自适应滤波系统质量的评判等等。自适应算法是根据性能判据设计迭代规则如迭代次数、滤波器系数更新公式等的算法，以达到最优的自适应性能。

1.1.3 自适应滤波算法

从上一节中随自适应滤波器结构的叙述中，可以了解到研究自适应算法的意义在于，要采取怎样的规则或者函数进行迭代，以什么样的性能判据算作收敛，从而得到最优的自适应滤波器系数的更新。因而判断一个自适应算法的性能指标^[1]有：

表 1.1 自适应滤波算法的性能指标介绍

性能指标	指标含义
自适应算法的收敛速度	自适应算法收敛到稳态所需要的迭代次数
收敛后的稳态误差性能	收敛达到稳态后的输出与期望信号的偏差
跟踪性能	包括算法的收敛速度与稳态误差性能
非稳定环境下的跟踪性能	处理非稳定系统的跟踪能力
抗冲激噪声鲁棒性	处理冲击噪声影响下的稳定性
计算复杂度	一次有效稳定收敛下需要进行的加法和乘法运算的次数

已有的易于实现的算法有 LMS 算法、RLS 算法、NLMS 算法以及它们相关的改进算法。另外，随着近年，人工智能和神经网络技术的兴起，基于神经网络、QR 分解、统一模型等的自适应滤波也陆续被提出。而 APA 算法作为 NLMS 的推广，仍然可以带来更高的自适应算法的收敛精度，这与之前的 LMS 等算法比较起来，在自适应滤波的各大应用中，比如系统辨识、干扰消除、预测模型及逆模型均具有更广泛的实用性。更实际一点结合当今社会的通信计算机网络的应用，用优化后的 APA 自适应滤波算法可以给网络中的各个网元设备带来精度更高更快质量更好的信号处理功能，结合机器学习、自组织网络和软件定义网络等最新技术，未来的网络可以实现高效精准的信息采集与处理，其中必不可少的信号接收中的自适应滤波算法是非常值得研究的。

1.2 国内外研究动态

自适应信号处理的研究工作始于 20 世纪中叶，为了消除混在有用信号中的噪声和干扰，在 1957 年左右美国通用电气公司的 P. Howells 和 P. Applebaum 与其同事研究出了简单的自适应滤波器。而后续对更为复杂结构的自适应滤波器的研究，则是由美国斯坦福大学的 B. Widrow 和 M. Hoff 在 1959 年开始的^[2]。Wiener 滤波器和最小均方算法 LMS 便是他们的研究成果之一，其中 Wiener 滤波器的原理是在已知所要处理的数据统计方面的先验知识的情况下，通过求解 Wiener-Hopf 方程，对平稳随机信号进行最优预测和滤波，使输出滤波器的均方估计误差最小。但是

在实际中的未知系统的应用中，因为无法得到信号特征的先验知识，因而无法应用 Wiener 滤波器。

1.2.1 从 LMS 发展到 APA

而 LMS 算法是在 Wiener 滤波的基础上，借鉴最速下降法的思想发展起来的。它的原理在随机输入 Wiener 滤波器递归计算中使用确定性梯度，再基于最小均方误差准则和最陡下降法，对权值进行迭代更新。LMS 算法因为方法简单，计算量小，易于实现且对信号的统计特性具有稳健性等，在信号处理领域得到广泛应用。但 LMS 算法收敛过程慢，步长与收敛速度失调，还需要设定一个合适的步长来保证其稳定性^[3]。

为解决 LMS 算法的步长问题，文献[3]中还介绍了一种归一化的最小均方滤波算法(NLMS)，NLMS 算法在工程实践中获得了广泛的应用。但是不能解决当输入信号严重相关时，算法收敛速度慢的问题。为了克服 NLMS 算法的这个缺点，文献[4]中提出的仿射投影算法(APA)拥有可以提高收敛速度和减少稳态误差的性能。不同于 NLMS 算法，投影仿射算法不仅利用当前的输入矢量，而且重复利用过去的 $P-1$ 个输入矢量（ P 为投影阶数）。当 P 为 1 时，APA 算法就是 NLMS 算法，因此 APA 可视为矢量化了的 NLMS 算法。

APA 算法的跟踪性能主要受步长影响，即步长越大，算法收敛速度越快，但稳态误差越高。反之亦然，即步长越小，稳态误差越低，但收敛速度越慢。因此，可以用变步长方法提高算法的性能，即在算法的开始阶段设定较大的步长来保证算法高速收敛，在算法接近稳态时设定较小的步长来保证算法有小的稳态误差。基于这个思想发展起来的算法称为变步长 APA(Variable Step Size APA, VSS-APA)^[5]。另外还有基于变正则因子的 APA 算法^[6]，可以极大地提高算法的鲁棒性，还有可以提高跟踪性能的结合凸组合技术的 APA 算法^[7]。

1.2.2 NSAF 算法的提出及其改进算法

在针对高相关性输入信号的自适应滤波处理上，文献[8]中 Lee 和 Gan 提出了在输入信号和期望信号经过分析滤波器分解后再进行迭代收敛的子带自适应滤波

算法(SAF), 这些被分解后的信号近似为独立的白信号, 这可以在维持稳态失调的同时提高算法的收敛速度。与 LMS 类似, Lee 和 Gan 又在[9]文提出的归一化的 SAF(NSAF)算法, 它一经提出就受到了很大的关注, 因为跟 NLMS 相比, 它在相关的输入信号下可实现更快的收敛速度而其计算负载并不增加。有关 NSAF 的改进算法也陆续提出, 比如基于 NSAF 的变步长算法, 可抗冲激噪声的 NSAF 算法 [10]等等。

1.2.3 NSAF 结合 APA 算法的提出

基于 SAF 和 APA, 文献[12]提出了 IMSAF, 它结合了 APA 和 SAF 的优点, 既可以平衡收敛速度和稳态误差, 同时还可以解决高相关性输入信号时的滤波问题。周海泉等人在[15]文中基于修改的最小化标准^[13], 提出了一种改进的仿射投影子带滤波器 (I-APSAF), 通过在每个迭代中重用过去抽头系数矢量, 同时牺牲了一些计算复杂度, 该算法可以实现在背景噪声具有较高的信噪比(信噪比)的低稳态误差。

1.2.4 APSA 算法的提出

基于 L_1 范数优化准则的符号类散发由于其简单并且抗非高斯冲激性噪声鲁棒性能好, 被研究起与 APA 算法结合形成了 APSA 算法^[16]。这种 APSA 算法虽然对抗非高斯的冲激性噪声有极好的性能, 但是在收敛速度和稳态误差的协调性和处理相关性的输入信号上面却无能为力。

1.2.5 本文的研究方向

对传统投影仿射类算法不能解决高相关性输入信号以及不能解决非高斯的冲激性噪声的两个主要问题来说, 本文结合已提出的能对高相关性输入信号进行有效滤波处理的 NSAF 的滤波结构以及能对非高斯的冲激性噪声进行有效处理的 APSA 算法。因而, 本文的主要研究方向是, 结合 SAF 和 APSA 提出 SAPSA 算法, 以及在有能力的情况下探索收敛速度和稳态误差更优的改进算法。

1.3 章节安排

根据最先的分析可明确，本文要研究三个循序渐进需要进行三个研究：第一个研究部分已经在第一章的 1 和 2 小节完成了；而对自适应算法的改进研究即是需要我们根据已有的自适应滤波算法，探索一个更优的自适应滤波算法的，这将在本文的第 2 和第 3 章完成；最后的投影仿射类的应用研究是需要对大量参考文献进行参考查阅，最后进行总结得到的，这将在本文的第 4 章完成。

所以本文的章节安排如下：第 1 章为绪论，主要介绍自适应滤波概述以及国内外研究动态，作为本课题研究背景及意义的说明；第 2 章将会逐个对 LMS、APA、SAF 和 APSA 进行介绍和算法推导；第 3 章将会提出一个结合了 SAF 和 APSA 的新型算法，包括其推导方法和 MATLAB 仿真测试；第 4 章将对自适应投影仿射类的应用研究情况做一个概述总结；第 5 章将会对本次毕业设计的研究课题进行一个总结，包括本文提出算法的改进思想和后续工作展望。

第 2 章 常用自适应算法分析

一个好的自适应滤波算法应该是要收敛速度快、稳态误差低、跟踪性能好、抗噪声鲁棒性能高以及计算复杂度上要满足可实现。本章将介绍一些基本的算法，并分别从这些性能上进行针对性的分析。

2.1 LMS 算法和 NLMS 算法

从文献[3]中可知，传统的 LMS 算法具体推导过程如下：设输入噪声 $n(k)$ 服从零均值方差 $\delta(k)^2$ 的高斯分布，LMS 算法的代价函数可表示为：

$$J(k) = [e(k)]^2, \text{ 其中 } e(k) = d(k) - \mathbf{x}^T(k)\mathbf{w}(k), \quad \mathbf{x}(k) = (x(k), x(k-1), \dots, x(k-L+1))^T \quad (2.1)$$

其中 k 为迭代次数， $\mathbf{x}(k)$ 是输入矢量， $\mathbf{w}(k)$ 是自适应滤波器系数矢量，设 $d(k)$ 是第 k 次迭代时的输出期望， $\mathbf{x}^T(k)\mathbf{w}(k)$ 是自适应滤波器的输出值， $e(k)$ 是输出期望与自适应滤波器输出的差值即误差值， μ 是步长参数（用来控制自适应滤波器的更新速度）。

利用最陡下降方法，自适应滤波器的更新公式可以表示为：

$$\mathbf{w}(k) = \mathbf{w}(k-1) - \mu \frac{\partial J(k)}{\partial \mathbf{w}(k)} \quad (2.2)$$

对 2.1 式进行求导可得：

$$\frac{\partial J(k)}{\partial \mathbf{w}(k)} = -2d(k) \cdot \mathbf{x}^T(k) + 2\mathbf{w}(k-1) \cdot [\mathbf{x}^T(k)]^2 \quad (2.3)$$

将 2.3 式代入 2.2 式化简后可得滤波器系数更新公式为：

$$\mathbf{w}(k) = \mathbf{w}(k-1) + 2\mu \cdot e(k) \cdot \mathbf{x}^T(k) \quad (2.4)$$

则可知，最小均方算法 LMS 的电路实现框图如下：

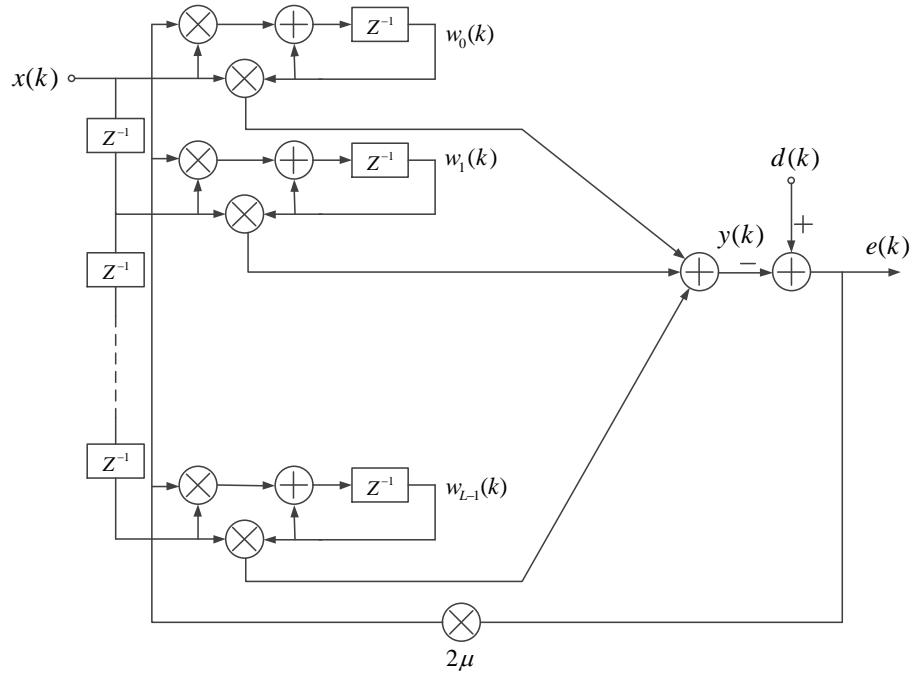


图 2.1 LMS 算法电路实现

2.4 式权值更新收敛的条件 $0 < \mu < 1/\lambda_{\max}$ ，其中 λ_{\max} 是输入信号相关矩阵的最大特征值。由以上推导公式结合图 2.1 的 LMS 算法电路实现，我们可以 LMS 算法每次迭代仅需 $2L+1$ 次乘法运算与 $L+2$ 次加法运算，其中 L 为滤波器长度，因而 LMS 算法简单容易实现。但同时当输入信号的功率谱密度集中或输入信号高度相关时，LMS 算法收敛速度缓慢。另外，LMS 的固定步长导致了其收敛速度与稳态误差不调，即在需要较低稳态误差时需要牺牲一定得收敛速度，此时需要设定一个较大的合适的步长，相反情况下则要重新设置一个较小的合适的步长。这给 LMS 算法的工程实践带来不便。

因此，基于步长的改进，文献[4]提出了归一化 LMS 算法（NLMS 算法）。考虑滤波器输入信号特性，并设定一个归一化的步长参数。NLMS 算法的推导可归结为一个受线性约束的最小化问题：

$$\min_{\mathbf{w}(k)} \|\mathbf{w}(k) - \mathbf{w}(k-1)\|_2^2 \quad \text{s.t.} \quad e_p(k) = 0 \quad (2.5)$$

其中 $e_p(k) = d(k) - \mathbf{x}^T(k)\mathbf{w}(k-1)$ ， $\|\cdot\|_2^2$ 表示某矢量 l_2 范数的平方， $e_p(k)$ 表示后验错误。则上式可理解为：对最优矢量 \mathbf{w}_0 的第 k 次评估值 $\mathbf{w}(k)$ 需满足该值使后验错误 $e_p(k)$ 为 0，且距离第 $k-1$ 次的矢量评估值得欧式距离最小。利用拉格朗日

法， $\mathbf{w}(k)$ 的更新式为：

$$\mathbf{w}(k) = \mathbf{w}(k-1) + \mu \frac{e(k)\mathbf{x}(k)}{\|\mathbf{x}(k)\|_2^2 + \delta} \quad (2.6)$$

其中， δ 表示正则因子，是为了避免 $\|\mathbf{x}_i\|_2^2$ 过小或者为 0 而引入的。NLMS 算法的收敛条件为， $0 < \mu < 2$ ，且可以证明当 $\mu=1$ 时，算法收敛速度最快。比较 NLMS 与 LMS 可以看出，NLMS 可以视为一个变步长的 LMS 算法，其中步长参数 $\mu_{NLMS} = \mu_{LMS} / (\|\mathbf{x}_i\|_2^2 + \delta)$ 。NLMS 算法在工程实践中获得了广泛的应用。但是不能解决当输入信号严重相关时，算法收敛速度慢的问题。此外，NLMS 算法也未考虑非高斯冲激噪声干扰情况。因此，当干扰为冲激噪声干扰时，NLMS 算法的性能严重衰减。

2.2 APA 算法

与 NLMS 不同的是，投影仿射算法^[6]不仅利用当前的输入矢量，而且重复利用过去的 $P-1$ 个输入矢量（ P 为投影阶数）。APA 可视为 NLMS 算法的推广。APA 算法也可由一个受线性约束的最小化表达式推导：

$$\min_{\mathbf{w}(k)} \|\mathbf{w}(k) - \mathbf{w}(k-1)\|_2^2 \quad \text{s.t.} \quad \mathbf{e}_p(k) = \mathbf{d}(k) - \mathbf{X}^T(k)\mathbf{w}(k-1) = 0 \quad (2.7)$$

与 NLMS 类似，只是输入信号的形式由当前输入矢量变成，由当前以及过去 $P-1$ 个输入信号矢量构成的输入信号矩阵 $\mathbf{X}(k) = [\mathbf{x}(k), \mathbf{x}(k-1), \dots, \mathbf{x}(k-P+1)]^T$ ，随之期望信号也变为了期望信号矢量 $\mathbf{d}(k) = (d(k), d(k-1), \dots, d(k-P+1))^T$ ，后验误差值也变成了后验误差矢量 $\mathbf{e}_p(k)$ ，那么上述线性约束模型可以这样描述：自适应滤波器系数的第 k 次评估值 $\mathbf{e}_p(k)$ （矢量）需要满足使后验误差矢量为 0 的约束条件，利用拉格朗日乘法可以得到权值矢量的更新公式为：

$$\mathbf{w}(k) = \mathbf{w}(k-1) + \mu \mathbf{X}(k)(\mathbf{X}^T(k)\mathbf{X}(k) + \delta \mathbf{I})^{-1} \mathbf{e}_p(k) \quad (2.8)$$

APA 的收敛条件是 $0 < \mu < 2$ ，但是 $0 < \mu \leq 1$ 时的算法收敛速度更快，并且在 $\mu=1$ 时，算法收敛速度最快。与 NLMS 中为了避免 $\|\mathbf{x}(k)\|_2^2$ 过小或者为 0 而添加正则因子 δ 的原因一样， $\delta \mathbf{I}$ 是为避免 $\mathbf{X}^T(k)\mathbf{X}(k)$ 是不满矩阵而引入的。由此可知，当投影阶数 $P=1$ 时，APA 即为 NLMS 算法。

与 LMS 类似, 标准 APA 算法的步长 μ 或正则因子 δ 控制算法收敛速度与稳态误差性能的平衡, 固定步长与正则因子只能满足一部分信号的收敛性能与稳态误差性。因此 APA 算法的改进优化的一个方向就是变步长。即在算法的开始阶段设定较大的步长来保证算法高速收敛, 在算法接近稳态时设定较小的步长来保证算法有小的稳态误差。基于这个思想发展起来的算法称为 VSS-APA (Variable Step Size APA)。另外, APA 在抗冲激噪声以及解决高相关性输入信号上也存在一些问题, 因此 APA 的改进优化方案可以基于这些不足来进行。

2.3 NSAF 算法

子带自适应滤波器实现的原理^[8], 即输入信号和期望信号经过一些分析滤波器因而被去相关化, 这使得大相关性的输入信号被分成了各自独立的子带信号, 再对各个子带信号应用进行有效地自适应算法更新。由 Lee 和 Gan 提出的规范化 SAF(NSAF)算法越来越受欢迎, 因为它比 NLMS 在相关的输入信号下可实现更快的收敛速度而其计算负载并不增加。

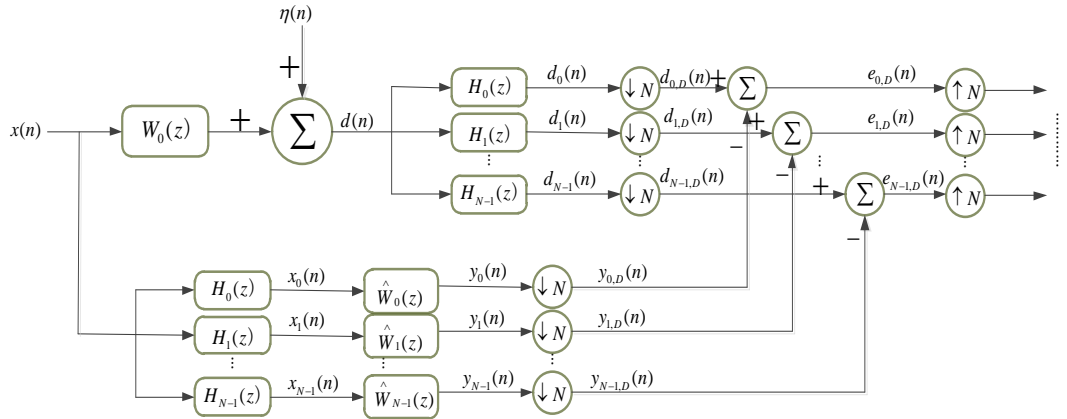


图 2.2 SAF 滤波器结构

子带自适应滤波器在系统辨识应用下的结构 (SAF)如图 2.2, 可知期望信号 $d(n)$ 可描述为以下线性模型:

$$d(n) = \mathbf{x}^T(n) \mathbf{w}_0 + \eta(n) \quad (2.9)$$

其中, \mathbf{w}_0 表示未知系统的 L 维权值矢量, $\eta(n)$ 表示输入噪声。

$\mathbf{x}^T(n) = [x(n), x(n-1), \dots, x(n-L+1)]^T$ 表示 L 维输入矢量。期望信号 $d(n)$ 可被分为 N

个分析滤波器 $H_i(Z), i=0,1,...,N-1$ 分解为 N 个子带信号 $d_i(n)$ ，对于输入信号的也可分解为 N 个子带信号。子带输入信号 $x_i(n)$ 可以通过该自适应滤波器得到子带输出信号 $y_i(n)$ 。子带信号 $d_i(n)$ 和 $y_i(n)$ 被抽样后分别得到 $y_{i,D}(k)$ 和 $d_{i,D}(k)$ 。被分解的子带误差信号可以由下列公式计算：

$$e_{i,D}(n) = d_{i,D}(n) - \mathbf{x}_i^T(n) \mathbf{w}(n) \quad (2.10)$$

这里 $\mathbf{x}^T(n) = [x(n), x(n-1), ..., x(n-L+1)]^T$ ， $\mathbf{w}(n) = [w_0(n), w_1(n), ..., w_{L-1}(n)]^T$ 表示自适应滤波器的权值系数矢量。传统 NSAF 算法通过以下成本函数最小化达到优化。

$$J(n) = \frac{1}{2} \sum_{i=0}^{N-1} \left(\frac{e_{i,D}(n)}{\|\mathbf{x}_i(n)\|} \right)^2 \quad (2.11)$$

这里， $\|\cdot\|$ 表示矢量的 l_2 范数模。应用梯度下降法，我们可以得到 NSAF 中的权值更新公式如下：

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \mu \frac{\partial J(k)}{\partial \mathbf{w}(k)} = \mathbf{w}(k) + \mu \sum_{i=0}^{N-1} \frac{e_{i,D}(k) \mathbf{x}_i(k)}{\|\mathbf{x}_i(k)\|^2 + \delta} \quad (2.12)$$

其中 μ 是步长， δ 是为了避免分母为 0 而引入，一般设置为一个接近为 0 的正数。

2.4 APSA

基于 L1 范数优化准则的自适应算法由于其简单并且抗非高斯冲激性噪声鲁棒性好，被研究起与 APA 结合形成了 APSA 算法，[16]文提出的 APSA 算法通过 L1 模优化准则来对权矢量进行更新，这保证了 APSA 算法对抗非高斯的冲激性噪声有极好的性能。但是对于高相关性输入信号和收敛速度稳态误差的协调上却不及其他算法。APSA 算法的原理 APA 类似，只是在进行优化时，用的是后验误差矢量的 L1 模范数准则。APSA 的算法可写成以下线性优化模型：

$$\begin{aligned} \min_{\mathbf{w}(k+1)} \quad & \|\mathbf{y}(k) - \mathbf{X}^T(k) \mathbf{w}(k+1)\|_1 \\ \text{subject to} \quad & \|\mathbf{w}(k+1) - \mathbf{w}(k)\|_2^2 \leq \delta^2 \end{aligned} \quad (2.13)$$

其中 δ^2 是保证更新的权矢量不要变化得过多的参数。也可以将它看作是控制

算法收敛快慢的一个参数。用拉格朗日乘法，定义的一个不受限代价函数如下：

$$J(\mathbf{w}(k+1)) = \|\mathbf{y}(k) - \mathbf{X}^T(k)\mathbf{w}(k+1)\| + \beta[\|\mathbf{w}(k+1) - \mathbf{w}(k)\|_2^2 - \delta^2] \quad (2.14)$$

β 是拉格朗日乘数。该代价函数对 $\hat{\mathbf{w}}(k+1)$ 求导后化简可得：

$$\frac{\partial J(\mathbf{w}(k+1))}{\partial \mathbf{w}(k+1)} = -\mathbf{X}(k) \text{sgn}[\mathbf{y}(k) - \mathbf{X}^T(k)\mathbf{w}(k+1)] + 2\beta[\mathbf{w}(k+1) - \mathbf{w}(k)] \quad (2.15)$$

其中，

$$\text{sgn}[\mathbf{y}(k) - \mathbf{X}^T(k)\mathbf{w}(k+1)] = \text{sgn}[\mathbf{e}_p(k)] = [\text{sgn}[\varepsilon(k)], \text{sgn}[\varepsilon(k-1)], \dots, \text{sgn}[\varepsilon(k-M+1)]]^T \quad (2.16)$$

令代价函数求导后为 0，可得权值更新公式：

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \frac{1}{2\beta} \mathbf{X}(k) \text{sgn}[\mathbf{e}_p(k)] \quad (2.17)$$

又结合限制公式可得， $\frac{1}{2\beta} = \frac{\delta}{\sqrt{\text{sgn}[\mathbf{e}_p^T(k)]\mathbf{X}^T(k)\mathbf{X}(k)\text{sgn}[\mathbf{e}_p(k)]}}$

代入权值更新公式得：

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \frac{\delta \mathbf{X}(k) \text{sgn}[\mathbf{e}_p(k)]}{\sqrt{\text{sgn}[\mathbf{e}_p^T(k)]\mathbf{X}^T(k)\mathbf{X}(k)\text{sgn}[\mathbf{e}_p(k)]}} \quad (2.18)$$

因为后验误差不可得，所以将其近似为先验误差 $\mathbf{e}(k)$ 。 δ 是控制收敛程度的一个系数，跟之前的步长因子 μ 是一样的功能，所以可以用 μ 来代替表示 δ 。设 $\mathbf{X}_s(k) = \mathbf{X}(k) \text{sgn}[\mathbf{e}_p(k)]$ 。则权值更新公式为：

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu \frac{\mathbf{X}_s(k)}{\sqrt{\mathbf{X}_s^T(k)\mathbf{X}_s(k) + \zeta}} \quad (2.19)$$

ζ 是添加用来避免分母为 0 的。 $0 < \mu \ll 1$ 来保证算法的稳定性。

2.5 本章小结

本章介绍的这几个算法虽说都是很基础的算法，但是在解决实际问题上却各有特长。现通过下表对这四个算法进行简单的对比。

表 2.1 四种算法的性能对比

算法	NLMS	APA	NSAF	APSA
权值更新公式	$\mathbf{w}(k) = \mathbf{w}(k-1) + \mu \frac{e(k)\mathbf{x}(k)}{\ \mathbf{x}(k)\ _2^2 + \delta}$	$\mathbf{w}(k) = \mathbf{w}(k-1) + \mu \mathbf{X}(k) \times (\mathbf{X}(k)^T \mathbf{X}(k) + \delta \mathbf{I})^{-1} \mathbf{e}_p(k)$	$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu \sum_{i=0}^{N-1} \frac{e_{i,D}(k)\mathbf{x}_i(k)}{\ \mathbf{x}_i(k)\ ^2 + \delta}$	$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu \frac{\mathbf{X}_s(k)}{\sqrt{\mathbf{X}_s^T(k)\mathbf{X}_s(k) + \zeta}}$
原理	l_2 范数优化准则 最陡下降法	利用过去 $P-1$ 个输入矢量 相当于矢量化的 NLMS	输入信号去相关化 最陡下降法	l_1 范数优化准则 APA
实现	简单	比 NLMS 难	与 NLMS 差不多	比 NLMS 稍难 比 APA 简单
稳态性	是变步长(归一化)的 LMS 算法	比较 NLMS, 具有较低的稳态误差	比较 NLMS 具有较低的稳态误差	与 APA 差不多
处理大相关性信号	算法性能差	算法性能差	可以有效处理	算法性能差
处理冲激噪声	算法性能差	算法性能差	算法性能差	可以有效处理

可以看出, LMS 的算法最简单, APA 基于 LMS 发展起来, 只是用到了更多的输入信号矢量, 当然算法复杂度也比 LMS 要高, 但是在稳态误差和收敛性能上却是大有所提高。另外由于 SAF 子带自适应滤波器用到了若干个分析滤波器将输入信号进行了去相关化, 因此可以处理大相关性输入信号时的自适应滤波问题。符号类自适应滤波算法一向对非高斯性的冲激噪声具有优良的处理能力, APSA 作为结合了符号类算法的投影仿射算法, 自然也是结合了这一优点。

第 3 章 子带投影仿射符号算法 SAPSA

在 APSA 的基础上, 为同时解决高相关性输入信号和非高斯性的冲激噪声的问题, 本文提出采用子带自适应滤波器的结构, 如图 2.2 的 SAF 滤波器结构, 但是在权值矢量系数更新时用 APSA 的迭代原理, 从而得出一种新的算法是 SAPSA。

3.1 算法推导

与 APSA 算法类似, SAPSA 采用基于 l_1 范数优化误差准则, 线性优化模型如下:

$$\begin{aligned} \min_{\mathbf{w}(k+1)} \quad & \|\mathbf{d}_D(k) - \mathbf{X}^T(k)\mathbf{w}(k+1)\|_1 \\ \text{subject to} \quad & \|\mathbf{w}(k+1) - \mathbf{w}(k)\|_2^2 \leq \delta^2 \end{aligned} \quad (3.1)$$

其中, δ^2 是保证更新的权矢量不要变化得过多的参数, 也可以将它看作是控制算法收敛快慢的一个参数。本次推导将以图 2.2 示的子带自适应滤波器为模型, 并设子带数为 N 。另设投影数为 P , 期望信号也就变成了一个 $N \times P$ 的矩阵, 输入信号变成了一个三重方阵 $N \times P \times L$, 其中 L 为滤波器抽头数, 滤波器抽头权值矢量还是一个 L 维的矢量。从 2.3 小节可得, 子带自适应滤波器各矢量之间有如下关系式:

$$\begin{aligned} \mathbf{X}(k) &= [\mathbf{X}_0(k), \mathbf{X}_1(k), \dots, \mathbf{X}_{N-1}(k)] \\ \mathbf{d}_D(k) &= [\mathbf{d}_{0,D}^T(k), \mathbf{d}_{1,D}^T(k), \dots, \mathbf{d}_{N-1,D}^T(k)]^T \\ \mathbf{e}_D(k) &= [\mathbf{e}_{0,D}^T(k), \mathbf{e}_{1,D}^T(k), \dots, \mathbf{e}_{N-1,D}^T(k)]^T \end{aligned}$$

其中:

$$\begin{aligned} \mathbf{X}_i(k) &= [\mathbf{x}_i(k), \mathbf{x}_i(k-1), \dots, \mathbf{x}_i(k-P+1)] \\ \mathbf{d}_{i,D}(k) &= [d_{i,D}(k), d_{i,D}(k-1), \dots, d_{i,D}(k-P+1)]^T \\ \mathbf{e}_{i,D}(k) &= [e_{i,D}(k), e_{i,D}(k-1), \dots, e_{i,D}(k-P+1)]^T \end{aligned}$$

其中还用 \mathbf{w}_0 表示由 L 维权矢量为模型表示的未知系统。基于公式(3.2)的线性模型用拉格朗日乘法可以定义面向第 i 个子带信号的不受限代价函数如下:

$$J_{i,D}(\mathbf{w}(k+1)) = \|\mathbf{d}_{i,D}(k) - \mathbf{X}_i^T(k)\mathbf{w}(k+1)\| + \beta[\|\mathbf{w}(k+1) - \mathbf{w}(k)\|_2^2 - \delta^2] \quad (3.3)$$

其中， β 是拉格朗日乘数。与 APSA 算法的推导类似，代价函数对 $\mathbf{w}(k+1)$ 求导后化简可得：

$$\frac{\partial J_{i,D}(\mathbf{w}(k+1))}{\partial \mathbf{w}(k+1)} = -\mathbf{X}_i(k) \text{sgn}[\mathbf{y}(k) - \mathbf{X}_i^T(k)\mathbf{w}(k+1)] + 2\beta[\mathbf{w}(k+1) - \mathbf{w}(k)] \quad (3.4)$$

由于有 N 个子带信号都需要进行这样算法处理，才能保证对高相关性信号的有效处理。推导进行到这里我们可以知道，如果要满足 N 维代价函数的导数矢量为 0 矢量的话，不能得到唯一的权值更新公式。即如果用本代价函数求导来推导出权值更新矢量的话，明显是得不到一个确切的更新公式的。

因为投影仿射符号算法针对的只是一个输入信号矢量，而经过了子带自适应滤波器的输入信号矢量已经被分解成 N 个子带信号矢量了。这样的话，我们可以借鉴 NSAF 的原理，即将代价函数进行多个信号的归一化处理，因而本文对式 3.2 进行如下修正：

$$J(\mathbf{w}(k+1)) = \sum_{i=0}^{N-1} \frac{\|\mathbf{d}_D(k) - \mathbf{X}_i^T(k)\mathbf{w}(k+1)\|}{\|\mathbf{x}_i(k)\|_2 + \delta_1} + \beta[\|\mathbf{w}(k+1) - \mathbf{w}(k)\|_2^2 - \delta_2^2] \quad (3.5)$$

其中，我们引入的 δ_1 是为了避免分母为 0 的情况，为了以示区别将前式的 δ 变为 δ_2 。可以看到 3.4 定义的代价函数已经包括了所有的子带信号，因而相当于合并了 N 个子带信号的代价函数。新的代价函数对权值矢量求导可得：

$$\frac{\partial J(\mathbf{w}(k+1))}{\partial \mathbf{w}(k+1)} = \sum_{i=0}^{N-1} \frac{-\mathbf{X}_i^T(k) \text{sgn}[\mathbf{d}_{i,D}(k) - \mathbf{X}_i^T(k)\mathbf{w}(k+1)]}{\|\mathbf{x}_i(k)\|_2 + \delta_1} + 2\beta[\mathbf{w}(k+1) - \mathbf{w}(k)] \quad (3.6)$$

令其为 0 可得：

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \frac{1}{2\beta} \sum_{i=0}^{N-1} \frac{\mathbf{X}_i^T(k) \text{sgn}[\mathbf{d}_{i,D}(k) - \mathbf{X}_i^T(k)\mathbf{w}(k+1)]}{\|\mathbf{x}_i(k)\|_2 + \delta_1} \quad (3.7)$$

将上式代入约束公式得：

$$\frac{1}{2\beta} = \delta_2 \cdot \frac{\|\mathbf{x}_i(k)\|_2 + \delta_1}{\left\| \sum_{i=0}^{N-1} \mathbf{X}_i^T(k) \text{sgn}[\mathbf{d}_{i,D}(k) - \mathbf{X}_i^T(k)\mathbf{w}(k+1)] \right\|_2} \quad (3.8)$$

代入权值更新公式得：

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \delta_2 \sum_{i=0}^{N-1} \frac{\mathbf{X}_i^T(k) \text{sgn}[\mathbf{d}_{i,D}(k) - \mathbf{X}_i^T(k) \mathbf{w}(k+1)]}{\left\| \sum_{i=0}^{N-1} \mathbf{X}_i^T(k) \text{sgn}[\mathbf{d}_{i,D}(k) - \mathbf{X}_i^T(k) \mathbf{w}(k+1)] \right\|_2} \quad (3.9)$$

因为后验误差不可得，所以将其近似为先验误差。

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \delta_2 \sum_{i=0}^{N-1} \frac{\mathbf{X}_i^T(k) \text{sgn}[\mathbf{d}_{i,D}(k) - \mathbf{X}_i^T(k) \mathbf{w}(k)]}{\left\| \sum_{i=0}^{N-1} \mathbf{X}_i^T(k) \text{sgn}[\mathbf{d}_{i,D}(k) - \mathbf{X}_i^T(k) \mathbf{w}(k)] \right\|_2} \quad (3.10)$$

在推导的过程中 δ_1 已经被抵消，因而最终更新公式中可以用来控制收敛程度的只有 δ_2 ，我们不妨将其设为 μ ，并设后验误差的符号矢量为

$$\Lambda_i(k) = \text{sgn}[\mathbf{d}_{i,D}(k) - \mathbf{X}_i^T(k) \mathbf{w}(k)] \quad (3.11)$$

可见，上式是对应第 i 个子带信号的 P 维误差矢量的符号向量，误差矢量为正时取值为 1，为负时取值为-1。进一步地，为了简化公式，我们还可设

$$\Psi(k) = \sum_{i=0}^{N-1} \mathbf{X}_i^T(k) \Lambda_i(k) \quad (3.12)$$

则权值更新公式可简化为：

$$\begin{aligned} \mathbf{w}(k+1) &= \mathbf{w}(k) + \mu \sum_{i=0}^{N-1} \frac{\mathbf{X}_i^T(k) \cdot \Lambda_i}{\left\| \sum_{i=0}^{N-1} \mathbf{X}_i^T(k) \cdot \Lambda_i \right\|_2 + \delta} \\ &= \mathbf{w}(k) + \mu \frac{\Psi(k)}{\|\Psi(k)\|_2 + \delta} \end{aligned} \quad (3.13)$$

同样为了避免分母为 0，我们可引入一个近似为 0 的正数，还是用 δ 来表示，与前面部分算法中的正则因子的作用类似。

3.2 算法的收敛性分析

本次算法用到的 NSAF 算法是基于系统辨识提出的，可定义权值误差矢量为 $\tilde{\mathbf{w}}(k) = \mathbf{w}_0 - \mathbf{w}(k)$ ，则根据本次算法的权值更新公式 3.9 和 3.12 可得：

$$\tilde{\mathbf{w}}(k+1) = \tilde{\mathbf{w}}(k) - \mu \frac{\sum_{i=0}^{N-1} \mathbf{X}_i^T(k) \text{sgn}[\mathbf{d}_{i,D}(k) - \mathbf{X}_i^T(k) \mathbf{w}(k)]}{\left\| \sum_{i=0}^{N-1} \mathbf{X}_i^T(k) \text{sgn}[\mathbf{d}_{i,D}(k) - \mathbf{X}_i^T(k) \mathbf{w}(k)] \right\|_2 + \delta} \quad (3.14)$$

在子带自适应滤波器里面，关于被降噪的期望信号的关系式是：

$\mathbf{d}_D(k) = \mathbf{X}^T(k) \mathbf{w}_0 + \boldsymbol{\eta}_D(k)$ 。其中

$$\boldsymbol{\eta}_D(k) = [\boldsymbol{\eta}_{0,D}^T(k), \boldsymbol{\eta}_{1,D}^T(k), \dots, \boldsymbol{\eta}_{N-1,D}^T(k)]^T$$

$$\boldsymbol{\eta}_{i,D}(k) = [\eta_{i,D}(k), \eta_{i,D}(k-1), \dots, \eta_{i,D}(k-P+1)]^T$$

其中， $\eta_{i,D}(k)$ 是第 i 阶子带噪声。那么由 $\mathbf{d}_D(k) = [\mathbf{d}_{0,D}^T(k), \mathbf{d}_{1,D}^T(k), \dots, \mathbf{d}_{N-1,D}^T(k)]^T$ 可得：

$$\mathbf{d}_{i,D}(k) = \mathbf{X}_i^T(k) \mathbf{w}_0 + \boldsymbol{\eta}_{i,D}(k) \quad (3.15)$$

代入式 3.13 中可得：

$$\tilde{\mathbf{w}}(k+1) = \tilde{\mathbf{w}}(k) - \mu \sum_{i=0}^{N-1} \frac{\mathbf{X}_i^T(k) \text{sgn}[\mathbf{X}_i^T(k) \tilde{\mathbf{w}}(k) + \boldsymbol{\eta}_{i,D}(k)]}{\left\| \sum_{i=0}^{N-1} \mathbf{X}_i^T(k) \text{sgn}[\mathbf{X}_i^T(k) \tilde{\mathbf{w}}(k) + \boldsymbol{\eta}_{i,D}(k)] \right\|_2 + \delta} \quad (3.16)$$

为了简化公式，又可设 $\tilde{\Psi}(k) = \sum_{i=0}^{N-1} \mathbf{X}_i^T(k) \text{sgn}[\mathbf{X}_i^T(k) \tilde{\mathbf{w}}(k) + \boldsymbol{\eta}_{i,D}(k)]$ ，则式 3.15 可简化为

$$\tilde{\mathbf{w}}(k+1) = \tilde{\mathbf{w}}(k) - \frac{\mu}{\left\| \tilde{\Psi}(k) \right\|_2 + \delta} \tilde{\Psi}(k) \quad (3.17)$$

借鉴[17]文中的均方权值偏差定义为(MSD, the mean-square deviation)：

$\text{MSD}(k) = E[\tilde{\mathbf{w}}^T(k) \tilde{\mathbf{w}}(k)]$ ，则有：

$$\begin{aligned} \text{MSD}(k+1) = \text{MSD}(k) &+ \frac{\mu}{\left\| \tilde{\Psi}(k) \right\|_2 + \delta} \left\{ E[\tilde{\Psi}(k)^T \tilde{\mathbf{w}}(k) + \tilde{\mathbf{w}}^T(k) \tilde{\Psi}(k)] \right\} \\ &+ \left[\frac{\mu}{\left\| \tilde{\Psi}(k) \right\|_2 + \delta} \right]^2 E[\tilde{\Psi}(k)^T \tilde{\Psi}(k)] \end{aligned} \quad (3.18)$$

设子带输入信号近似为白信号，则子带输入信号的能量波动在一次次中的迭代下会逐渐变小。而且包含了噪声的符号函数 $\text{sgn}[\mathbf{X}_i^T(k) \tilde{\mathbf{w}}(k) + \boldsymbol{\eta}_{i,D}(k)]$ ，模值也是不超过 1 的，因此很明显 $E[\tilde{\Psi}(k)^T \tilde{\mathbf{w}}(k) + \tilde{\mathbf{w}}^T(k) \tilde{\Psi}(k)]$ 以及 $E[\tilde{\Psi}(k)^T \tilde{\Psi}(k)]$ 随着迭代次数增加而减少，并最终趋于稳定。

3.3 算法仿真

现在需要测试本算法在处理大相关性输入信号以及非高斯性的冲激噪声方面的自适应性能。参考[17]文的仿真模型，将从在 AEC 应用上进行仿真测试。回声

路径将采用[17]文中的 512 个系数模型，自适应滤波器的长度也将设置为 512。我们采用正弦调制滤波器，子带数目是 $4^{[16]}$ ，投影阶数是 $N=4$ 。输入信号模型^[17]将选用零均值高斯白噪声通过极点为 0.8 的一阶 AR 系统产生有一个有色信号。为了比较算法的跟踪性能，回声路径的冲激响应将在迭代中乘以-1。另外，输出信号的回声路径将会分别用由不同信噪比的高斯白噪声以及非高斯的冲激噪声组成^[17]。高斯白噪声的模型比较容易生成，非高斯白噪声的冲激噪声模型本文采用伯努利-高斯混合模型来产生。

为测试算法的稳态误差，可用归一化权误差矢量期望 (normalized weight error vector): $10\log_{10}\left(E[\|\mathbf{w}(k)-\mathbf{w}_0\|_2^2]/\|\mathbf{w}_0\|_2^2\right)$ ，它随迭代次数的变化曲线反映了一个算法的收敛速度和稳态误差，所以以下将这个指标简称为归一化稳态误差。仿真结果将取 50 个仿真结果的平均值。另外，为了方便对比出算法在高相关性输入信号和冲激噪声处理上的性能，本文在仿真上同时在相同的条件下对 APSA、NSAF、IAPSA^[17]和 SAPSA 算法进行仿真性能测试。以下分别从不同条件下给出算法性能（归一化稳态误差）的仿真结果图，具体仿真源码见附录 A。

3.3.1 不同步长值的 SAPSA 算法性能

下图是 SAPSA 算法在 AR 输入信号条件下、系统噪声含有-10dB 信噪比的冲激噪声以及 5dB 的高斯噪声时 S-APSA 算法对声学回波消除应用的归一化稳态误差仿真结果。

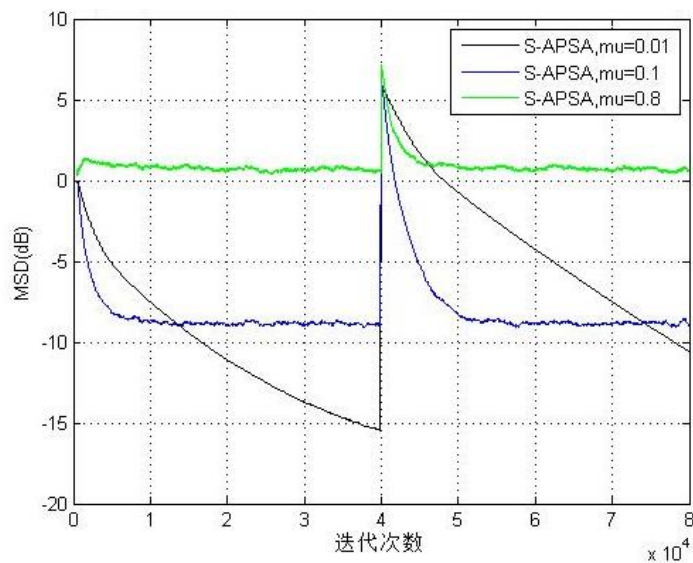
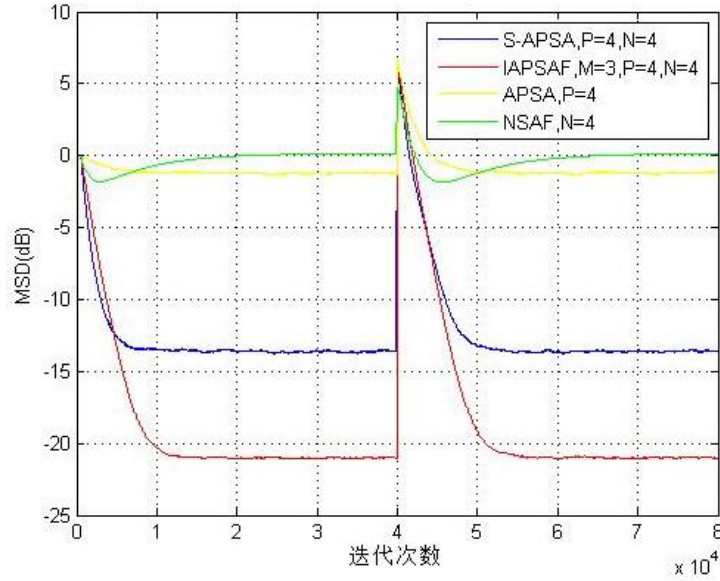


图 3.1 步长分别为 0.01, 0.1, 0.8 条件下的 SAPSA 算法性能图

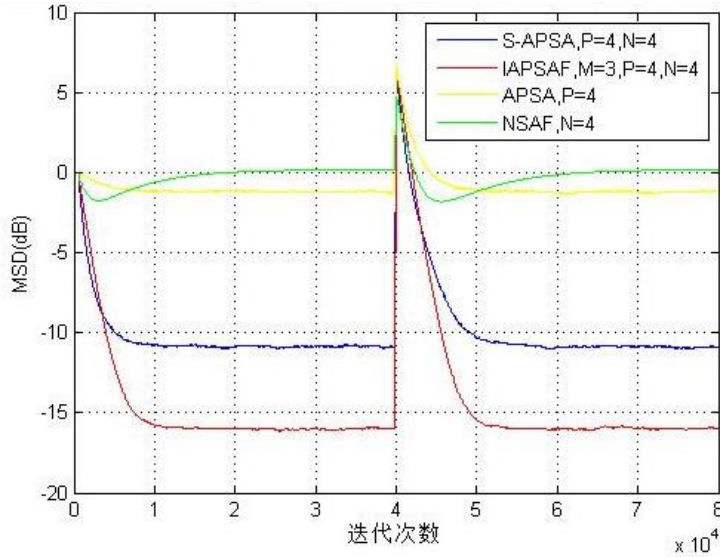
从上图可看出在三种不同步长下的仿真对比下，本次实验条件中 SAPSA 算法在步长为 0.1 附近的收敛速度和稳态误差性能较优。故后续仿真皆用 0.1 的补偿值。

3.3.2 AR 信号+高斯白噪声

用 MATLAB 软件对所提算法进行 AR 输入信号、系统噪声仅为高斯白噪声条件下的性能测验，其归一化稳态误差随迭代次数变化曲线如下图所示：



(a)10dB 信噪比的高斯噪声



(a)5dB 信噪比的高斯噪声

图 3.2 $\mu=0.1$, $AR=[1 \ -0.8]$, $\delta=0.001$, 条件下的归一化稳态误差

由图 3.2 可以看出，5dB 信噪比条件下的 IAPSAF 和 SAPSA 算法的稳态性能

低于 10dB 信噪比条件下的稳态性能,而 NSAF 和 APSA 算法的稳态性能却没有太大的变化,这说明 NSAF 和 APSA 两种算法对于含系统噪声的回声消除效果均不是特别理想,而 SAPSA 和 IAPSAF 两种算法均可以对含系统噪声进行有效地回声消除,并很容易理解较大信噪比下的处理效果肯定优于较小信噪比的处理效果。

但不论信噪比是 5dB 还是 10dB,从图上归一化稳态误差的收敛情况和收敛值得高低都可以看出,本文所提的算法都是远远优于基本函数 APSA 和 NSAF 算法的。当然了,在低稳态误差的效果上看,本文所提的算法不及[17]所提出的 IMSAF 的改进算法 IAPSAF,毕竟在优化准则上,IAPSAF 最小化了更新的抽头权值和过去 M 维抽头权值矢量的差值平方和,而不仅仅用当前抽头权值和前一个抽头权值的差值做优化函数,这无疑增大了算法的复杂度,但同时也使得算法的稳态性能更优,这点可以用于进一步地后期研究工作的优化方面。

3.3.3 AR 信号+非高斯的冲激性噪声

用 MATLAB 软件对所提算法进行 AR 输入信号、系统噪声为高斯白噪声+冲激性噪声条件下的性能测验,其归一化稳态误差随迭代次数变化曲线如下图所示:

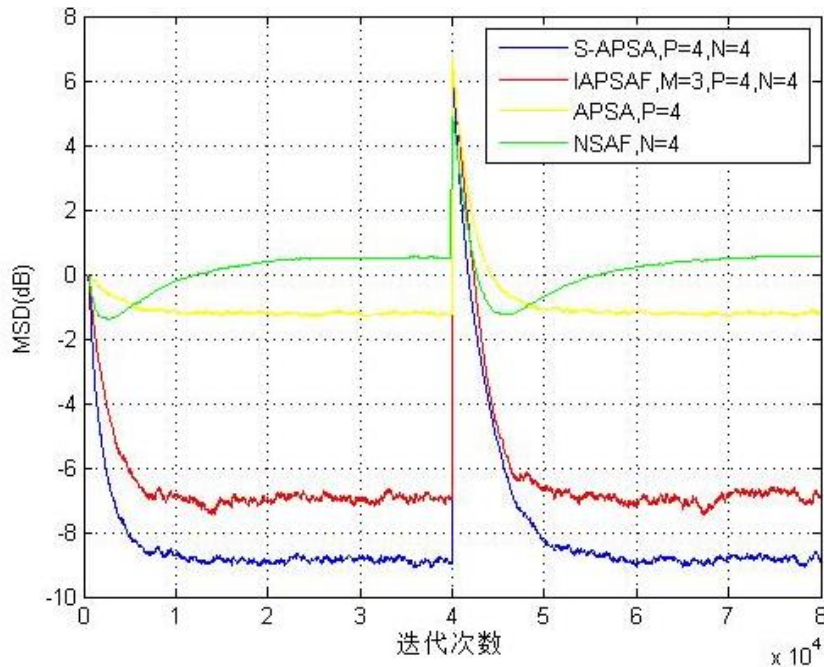


图 3.3 $\mu=0.1$, $AR=[1 \ -0.8]$, $\delta=0.001$, -10dB 信噪比的冲激噪声和 5dB 信噪比的高斯噪声条件下的归一化稳态误差

从上图可以看出,在处理系统噪声含有冲激噪声条件下的声学回波消除应用

时，SAPSA 的性能最优（收敛最快以及稳态误差最小），其次是 IAPSAF 算法，而 APSA 算法的处理效果并不是特别好，随着迭代次数的增加，NSAF 算法甚至出现了稳态误差回升至超过 0 的现象。

3.3.4 仿真数据总结

以上一系列仿真实验的相关数据记录如下两图所示：

表 3.1 不同条件下 S-APSA 算法仿真的数据记录

50 次仿真用时/s	1 次仿真用时/s	冲激噪声/dB	高斯噪声/dB	μ	δ	稳态误差	收敛速度
410.633521	≈ 8.2	-10	5	0.8	0.001	大	快
545.621611	≈ 10.9	-10	5	0.8	0.02	较大	快
595.562063	≈ 11.9	-10	5	0.1	0.02	较小	慢
728.173792	≈ 14.6	0	5	0.1	0.001	小	慢

可以看出，本算法在步长和正则因子对于收敛速度和稳态误差上仍存在比较严重的失调现象，即在较小步长时稳态误差较小但是收敛速度也变慢了，在较大补偿时收敛速度变快了但是同时稳态误差又受到了一定的影响。故而针对本文所提的针对 AR 信号和冲激噪声处理的 S-APSA 算法还需要后续工作对步长、收敛速度和稳态误差失调上进行改进。

表 3.2 IAPSAF/S-APSA/APSA/NSAF 算法仿真的数据记录

50 次仿真用时/s	1 次用时/s	AR 信号	冲激噪声/dB	高斯噪声/dB	μ	δ	实验次数	算法
681.234514	≈ 13.6	[1 -0.8]	-10	5	0.8	M=3	50	IAPSAF
585.6231	≈ 11.7	[1 -0.8]	-10	5	0.1	0.001	50	S-APSA
351.090029	≈ 7.02	[1 -0.8]	-10	5	0.1	0.001	50	APSA
135.5691	≈ 2.7	[1 -0.8]	-10	5	0.1	0.001	50	NSAF

该算法本就针对 AR 输入信号和冲激噪声而提出的，从 3.3.3 的仿真结果图上也可以看出该算法确有这个效果。表 3.2 记录了四种算法在各自较为理想的条件下 80000 次迭代以及 50 次重复实验下总共需要的时间为以及平均每次实验用时，由于都是相同条件的声学回波消除的，其平均实验用时即可从某种意义上代表其算法复杂度。由此可见，所算法与 NSAF 和 APSA 相比之下算法复杂度增大了很多，但没有 IAPSAF 的算法复杂度高，但是其同时针对 AR 输入信号和非高斯冲激噪声

条件下的应用时效果尤其显著。

第 4 章 自适应滤波算法及 APA 类算法的应用研究

自适应滤波对于信号处理具有很强的自学习和自跟踪能力，在对信号噪声的强检测增强、噪声干扰抵消、波形编码的预测、通信系统的自适应均衡、图像自适应压缩编码、系统识别等等方面应用很多。另外从 1.1 节中的自适应滤波叙述中可知，自适应滤波器的应用主要可分为 4 类：系统辨识、逆模型、预测未知信号或噪声估计和干扰消除^[22]，而投影仿射类自适应滤波算法作为自适应滤波器下多种算法的一种，如果应用在信号滤波处理上，大致的应用情况也应该是这四种。由于研究的时间问题，不能一一将这四种应用用 APA 算法进行仿真模拟实现，因而本章将只是对相关应用的理论知识进行以下概括。本章将先简单对这四种自适应滤波器的应用进行介绍说明，再分别结合投影仿射类算法进行对应的研究说明，最后总结投影仿射类的应用情况。

4.1 自适应滤波器的四种应用

4.1.1 系统辨识

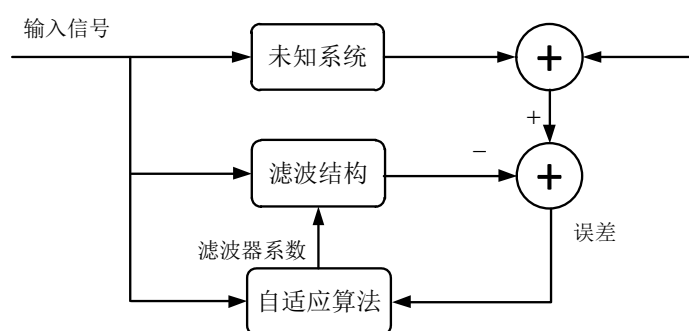


图 4.1 系统辨识下的自适应滤波框图

从文献[21]和文献[22]可以参考得到系统辨识是最常用的自适应滤波应用。如图 4.1，这时自适应滤波算法的参考信号（即期望信号）就是未知系统的输出，当它与自适应滤波器输出信号的误差最小时，此时自适应滤波器就与未知系统具有相近的特性。所以在系统辨识应用中，自适应滤波器可提供一个能够近似拟合未知系统的线性模型。自适应系统辨识在自动控制、建模等方面有着广泛的应用。

4.1.2 逆模型

如图 4.2 所示，自适应滤波器的作用是提供一个逆模型，该模型可在某种意义上最好拟合未知噪声装置。实现原理是，最小化输入信号通过未知系统和自适应滤波器的输出值与延时后的输入信号之间的误差值，经自适应算法多次迭代后实现稳态后，自适应滤波器便成了一个对未知系统最近的逆模型，在线性系统的情况下，该逆模型的系统函数近似等于未知系统的系统函数的倒数。

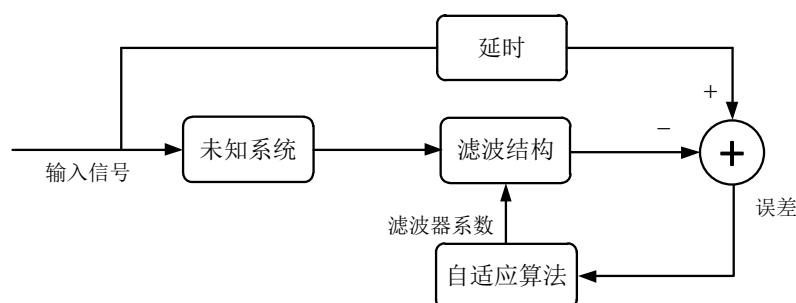


图 4.2 逆模型应用下的自适应滤波框图

常用于通信系统的逆模型的自适应滤波原理是自适应信道均衡技术，整个自适应滤波逆模型系统构成信号接收端的最前端处理部分，由于通信技术中的输入信号经常受到了来自传输过程中的各类信号干扰，在接收端得到的信号往往存在着各类幅度和相位上的信号失真，而自适应均衡器能够消除这种失真，所以它经常用于无线电通信中对付码间干扰的一类抗干扰技术。

4.1.3 信号预测

自适应滤波在预测的应用框图如图 4.3，其可以达到预测的原理是通过最小化输入信号和其延时信号通过自适应滤波器的输出值的误差值，多次收敛后误差值达到稳定是，自适应滤波器即可以近似实现对其输入端进行信号预测的作用。

换一种说话就是，将信号的当前值用作自适应滤波器的期望响应。信号的过去值（延时后的当前值）加到滤波器的输入端。自适应滤波器的输出或估计误差均可作为系统的输出。根据具体应用情况来说的话，只应用自适应输出值作为系

统输出的话，系统就实现了信号预测的作用；而只应用自适应滤波器的估计误差作为系统输出的话，系统就实现了预测噪声的作用。

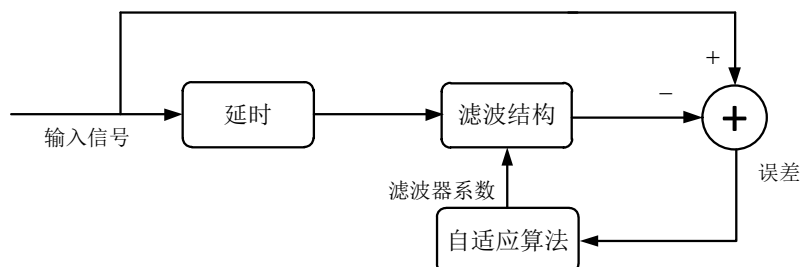


图 4.3 预测应用下的自适应滤波框图

另外，自适应预测的一个主要的具体应用是语音信号的波形编码。为了要减少传输必须的比特数，预测误差信号将被量化而且送给接收器。这种波形编码被称为自适应微分脉冲代码调制，可以提供 32kb/s 的语音压缩率。自适应预测器也能用来发现并且加强调制在宽带噪声中的窄带信号。

4.1.4 干扰消除

干扰消除是自适应信号处理技术中最常见的应用，自适应滤波算法用于干扰消除下的自适应滤波框图如图 4.4。可以看到，滤波器结构的输入端为噪声信号，期望信号输入端为带有噪声信号的输入信号，这样的话，自适应滤波器以带噪声信号减去经过自适应滤波的预测噪声值，从而达到消除噪声或干扰的目的。

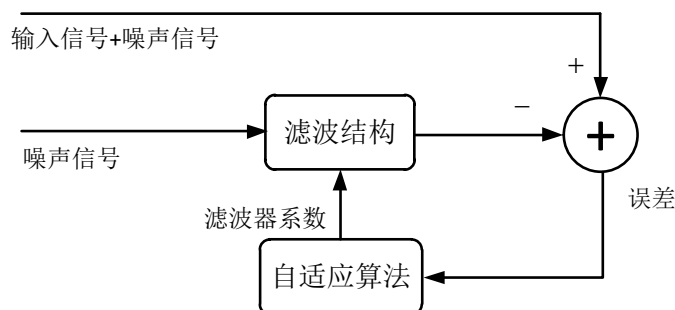


图 4.4 干扰消除应用下的自适应滤波框图

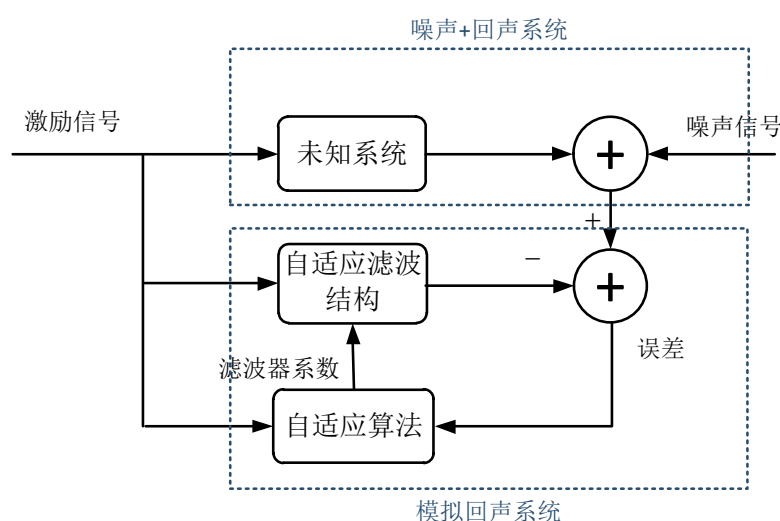
自适应去噪常见的应用包括各种形式的干扰消除，比如心电图仪、语音信号噪声、无线电干扰以及声学回声消除等等。声学回波消除器的输入信号为语音信号，语音信号属于非平稳信号，动态范围比较大，可达 40dB 以上。其回波路径也常随声音传播的环境而变化。这些不确定因素都自适应滤波算法提出了很高的要求。

4.2 自适应投影仿射类算法的应用

自适应仿射投影类算法以其算法复杂度居于 LMS 和 RLS 之间，且经过改进后的 APA 算法可以实现收敛速度和稳态误差、可抗冲激噪声以及可处理大相关性输入信号等等的性能提升。所以对于不同方面改进的 APA 算法在各个方面的处理能力的不同，其可实现方向以及应用方向也有很大的侧重点。以下分别就本人所查阅的应用内容做一个分类概述。

4.2.1 回声消除

从文献[25]中可知，一个传输系统的输出和输入端会存在通过反馈路径产生的反馈信号，由此产生具有回声效应。而在自适应滤波中在回声消除应用中，基本原理是用自适应滤波器模拟回声路径，通过自适应滤波算法的调整，使其冲激响应与回声路径相逼近，从而得到回声预测信号，再将预测信号从近端接收到的语音信号中减去，即可实现回声消除。回声消除应用下的自适应滤波框图如图 4.5，比较图 4.1，可以发现两者框图结构几乎一样，则说明自适应滤波器在回声消除下的应用是属于系统辨识这个大的应用条件的。

图 4.5 回波声消除应用下的自适应滤波框图^[26]

由于回声信号具有相关性比较大的特性，因而用结合子带自适应滤波器 SAF 的 APA 算法比如[17]文提到的 IAPSAF(Improved Affine Projection Sub-band Adaptive Filter)算法。有的文章也把回声效应的消除叫做回波消除，[22]文将其分为电学回波消除和声学回波消除。可以这样简单的理解，消除通过电路传输系统中的反馈回波叫做电学回波消除，比如消除长途电话和调制解调器中的回波效应；那么声学回波消除就是消除语音信号通过物体反射而产生的回波效应，比如在电话或者视频会议系统应用中，远端会议者传输的语音信号通过扬声器播放后经过柜子、桌子、墙壁等物体的反射，又被麦克风录入并反馈回远端系统，形成远端的声学回波。因此[22]文提出了针对立体声回波消除的两路投影仿射算法，该算法强调输入信号之间的相关性的变化使得加强权系数向量收敛到真实的回波路径，因此该算法对立体声回声消除的应用尤其有效。

4.2.2 电子回声消除

从上个小结的[21]文介绍中可以看出，电子回波消除是属于电子传输系统中的回声消除一类的。[26]文对用仿射投影类算法来电子回声消除进行了研究。该文指出回声产生的原因有模拟电路 2-4 线转换过程阻抗不匹配导致电气泄露、数字信号在编码压缩传输解压过程中引发了时延等等，因此为避免电子回声影响通信质量，需要再通信设备里集成回声消除器以抑制回声能量。

从上节对回声消除的描述中可知，自适应滤波在回声消除上的应用原理属于系统辨识大类。同样的，自适应滤波在电子回声消除的应用上也同属于系统辨识大类。另外[26]文也提到针对回声路径的稀疏性，Duttweiler 等学者已经提出过结合了等比例思想的 NLMS 算法。另外，由于 APA 在算法的收敛速度稳态误差等上面比 NLMS 性能更优，故结合了等比例思想的 APA 算法被提出了，即 PAPA (Proportionate Affine Projection Algorithm)。PAPA 算法通过步长控制矩阵按自适应滤波器各系数幅值大小等比例地为其制定相应的收敛步长。文献 [26]中对 PAPA 的仿真结果也表明了，PAPA 算法针对电子回波上的应用的话性能将比 NLMS 和 PNLMS 好。

4.2.3 数字预失真系统

将自适应滤波算法应用到了现代移动通信技术中的数字预失真系统中，用来补偿其中非线性功率放大器的误差，从而达到“数字预失真”的目的。其中，现代移动通信系统采用了大量的基于非线性功率放大器和非恒定包络的多载波数字调制技术，这些放大器必须要求有较高的线性度，以防止信号经放大输出后产生互调失真分量，导致其输出频谱扩展最后形成对邻近信道的干扰。而数字预失真技术就是为了提高这些非线性功率放大器的一种新型技术，其原理之一就是利用自适应算法来跟踪补偿功率放大器由于温度、适度、电源漂移、机械振动、线路老化等等造成各部分的误差。由此可见，该文中为实现数字预失真系统的自适应算法的应用对应于 4.1.3 节的信号预测的应用大类，因为其对误差信号补偿之前的估计预测是属于信号预测这个类别的。

但是具体的来说，[27]文中直接研究到自适应算法的地方是数字预失真技术中的查找表(Look up table, LUT)法。然后自适应算法就可以通过系统中功率放大器的一些输入输出数据不断地进行更新查找表里的参数，从而达到拟合功放非线性曲线的目的，从这个应用层面上来看，自适应算法在查找表里的应用就像是系统辨识的应用大类一样，只不过拟合的是功率放大器的非线性参数。

4.3 本章小结

本章完成了此次研究课题的第三个研究部分----对投影仿射类算法的应用研究。4.1 小节主要介绍了对自适应滤波的四种应用大类：系统辨识、逆模型、信号预测和干扰消除。在 4.2 小节则结合了相关对投影仿射类的应用研究文献来进行总结介绍的。该小节介绍了三种直接用 APA 算法或其改进算法的应用，他们是回声消除、电子回声消除以及数字预失真系统。当然了还有更多的场景都可以应用到 APA 类算法，这三种应用仅仅作为参考例子，可以让我们更加实在地了解到 APA 算法在实际生产生活中的应用。

第 5 章 总结与改进

本章节将从本次研究课题的全部工作量、研究成果以及对后续学者在类似研究课题中的工作展望对本次毕业设计做一个简单的总结。

5.1 主要工作总结和创新点

根据第一章的介绍, 本文将此次课题分为三个研究部分, 并分别已在第 1、2、3、4 章依次完成。其中第一个研究部分“自适应算法的学习研究”我主要是用过对相关自适应滤波理论书籍的学习总结以及结合相关基本自适应滤波算法比如 LMS、NLMS、APA 等的参考文献进行研究和对比学习的, 最终顺利地完成了对本部分的研究, 并且给第二个研究部分打下了深刻的理论基础。

关于第二个研究部分“投影仿射与自适应算法的改进研究”的完成, 即本文第 2 章和第 3 章的内容, 其中第 3 章是本次研究最主要的成果部分也是创新点。本文主要参考了文献[5-18]从而对相关的自适应滤波算法做了大量的算法推导和了解, 对它们各自的优缺点以及算法原理也有了更加明确的对比, 最后从 APSA 和 IAPSAF 两个算法中得到启发并推导得出了本文的 SAPSA 算法, 这个新的算法的提出解决之前学者们对自适应算法不能同时觉得大相关性输入信号和冲激性噪声处理的问题, 最后通过 MATLAB 仿真结果对比 APSA 和 IAPSAF 算法, 可以验证本文提出的算法确实具有能针对性解决大相关性和冲激噪声的问题。

最后一个研究部分“投影仿射类算法的应用研究”, 这个研究点比较大, 按个人的理解的话, 这个研究是需要对大量的投影仿射类算法进行不同应用上的仿真实验的, 但是由于时间关系我只能先通过理论对它进行系统性的概括总结, 参考资料也是结合了相关自适应书籍和参考文献来的。

5.1 改进方面和后续研究展望

本次研究的重点在于提出了 SAPSA 算法, 结合 l_1 优化准则和子带自适应滤波来进行算法推导的, 并且对同时解决大相关性输入信号以及非高斯性冲激噪声具有很好的处理效果。但是与 APA、LMS 这些基本算法一样, 它仍然存在着步长和

收敛速度/稳态误差之间失调的问题，即跟踪性能不是特别理想。因而对本算法对于跟踪性能的改进点，应着重在于对其步长、正则因子的变化以及结合凸组合函数来入手；另外对于处理稀疏系统的性能来进行提升的话，可以结合比例仿射投影的方法来进行改进。

另外，虽然研究的过程是单调且冗长的，但这个经历已经让我受益无穷并且使我开始尝试到科学研究的喜悦和成就感。我对本文的后续研究工作很是看好，因为已经有大量的可以参考的改进文献，并且在本文同时可以处理大相关性输入信号以及非高斯冲激性噪声的算法条件下，对提升算法的跟踪性能或者处理稀疏系统等等不同情况下的自适应滤波问题的改进，都可以通过持之以恒的研究逐一解决。

参考文献

- [1] Dimitris G. Manolakis, Vinay K. Ingle, Stephen M. Kogon. Statistical and Signal Processing[M]. Norwood, MA: Ariech House, 2005: 499-620.
- [2] B. Widrow. Thinking about thinking: the discovery of the LMS algorithm[J]. IEEE, Signal Processing Magazine, 2005, 22(1): 100-106.
- [3] S. Haykin. Adaptive Filter Theory. 4th Edition[M]. Upper Saddle River, NJ: Prentice Hall, 2002(4): 96-125.
- [4] Insun Song, PooGyeon Park, Robert W. Newcomb. A normalized least mean squares algorithm with a step-size scaler against impulsive measurement noise[J]. IEEE, Transactions on Circuits and Systems II: Express Briefs, 2013, 60(7): 442-445.
- [5] Jong Hyun Choi, Hyeonwoo Cho, Jae Jin Jeong et al. Combination of step sizes for affine projection algorithm with variable mixing parameter[J]. Electronics Letters, 2013, 49(18): 1149-1150.
- [6] 师黎明, 林云. 基于无噪后验错误矢量信号能量的变正则因子仿射投影算法[J]. 电子学报, 2015, 43(1): 7-12.
- [7] Leonel Arevalo, Jose A. Apolinario, Marcello L. R. de Campos et al. Convex combination of three affine projections adaptive filters[C] // The Tenth International Symposium on Wireless Communication Systems (ISWCS). Ilmenau: VDE, 2013: 1-5.
- [8] Kong-Aik Lee, WoonSeng Gan, Sen-Maw Kuo. Sub-band Adaptive Filtering: Theory and Implementation[M]. Wiley, UK: Chichester, 2009
- [9] Kong-Aik Lee, WoonSeng Gan. Inherent decorrelating and least perturbation properties of the normalized Sub-band adaptive filter[J]. IEEE, Transactions on Signal Processing, 2006, 54 (11): 4475-4480.
- [10] Mohammad Shams Esfand Abadi, Mohammad Saeed Shafiee. A new variable step-size normalized subband adaptive filter algorithm employing dynamic selection of subband filters[C] // The 21st Iranian Conference on Electrical Engineering (ICEE), IEEE Conference Publications, 2013: 1-5

- [11] JinWoo Yoo, JaeWook Shin, Poo Gyeon. Park Variable step-size sign algorithm against impulsive noises[J]. IET Signal Processing, 2015(6): 506-510
- [12] F. Yang, M. Wu, J. Yang. An improved multiband-structured Sub-band adaptive filter algorithm[J]. IEEE Signal Processing Letters, 2012,19 (10): 647-650.
- [13] H. Cho, C.W. Lee, S.W. Kim. Deviation of a new normalized least mean squares algorithm with modified minimization criterion[J]. IEEE Signal Processing, 2009,89(4) : 692-695.
- [14] J. Ni. Improved normalized Sub-band adaptive filter[J]. Electronics Letters, 2012,48 (6): 320-321.
- [15] K. A. Lee, W. S. Gan. Improving convergence of the NLMS algorithm using constrained Sub-band updates[J]. IEEE Signal Process. 2004,11(9): 736-739.
- [16] Tiange Shao, Yahong Rosa Zheng, Jacob Benesty, et al. An Affine Projection Sign Algorithm Robust Against Impulsive Interferences[J]. IEEE Signal Processing Letters. 2010, 17(4): 327-330.
- [17] Haiquan Zhao, Zongsheng Zheng, Zhongyu Wang. Improved affine projection Sub-band adaptive filter for high background environments[J]. IEEE Signal Processing, 2017(137): 356-362.
- [18] Z. Zheng , H. Zhao. Affine projection M-estimate Sub-band adaptive filters for robust adaptive filtering in impulsive noise[J]. IEEE Signal Process, 2016(120): 64–70.
- [19] 郭莹, 白艳梅. 稳健的仿射投影符号自适应滤波算法[J]. 仪器仪表学报. 2017. 38(1): 23-32.
- [20] 郭华, 自适应滤波算法及应用研究[D]. 西北师范大学. 2007.
- [21] 林耀荣, 自适应滤波理论及其在回波消除中的应用研究[D]. 华南理工大学. 1999.
- [22] KIM Young-Sear, SONG Ji-hyun, KIM Sang-Kyun et al. Variable step-size affine projection algorithm based on global speech absence probability for adaptive feedback cancellation[J]. J. Cent, South Univ, (2014) 21: 646–650.
- [23] Markus Rupp, Walter Kellermann, Abdelhak Zoubir. Advances in adaptive filtering theory and applications to acoustic and speech signal processing[J]. IEEE Signal Processing 2016: 63.

- [24] Behrouz Farhang. Adaptive Filters Theory and Applications. Second edition[M]. Boroujeny University of Utah, USA: A John Wiley & Sons, Ltd. 2013: 139-348
- [25] 王炳扬, 杨志成. 多种自适应算法在回声消除中的应用[J]. 舰船电子工程. 2015(255): 68-71.
- [26] 文昊翔, 陈隆道, 蔡忠法. 改进仿射投影算法及其在回声消除中的应用[J]. 电子学报. 2012: 1229-1234.
- [27] 唐可然, 张冬生, 熊于菽. 仿射投影自适应算法在数字预失真系统中的应用[J]. 数据采集与处理. 2014(29): 795-800.

致谢

首先，本人对帮助指导此次研究工作的周翊老师和李季碧老师表示感谢，正是这两位老师对我平时的工作进度提出了指导性以及方向性的意见；再者，我要感谢我的学长师黎明、舒晓峰、我的学姐魏丹丹以及来自西南交通大学的学术交流学长王忠玉，他们对我的算法仿真给予了丰富的经验交流和关键性的建议；最后，我要感谢中国知网和 IEEE 两大期刊查阅网站以及本校提供的阅读账号，让我能够免费阅读到大量支持性的参考文献。

附录 A 算法仿真 MATLAB 代码

AR 信号+冲激噪声条件下的仿真代码

```

% S-APSA、IAPSAF、APSA 和 NSAF 四个算法的对比仿真
clc; % 擦去一页命令窗口，光标回屏幕左上角
clear all; % 从工作空间清除所有变量和函数
close all; % 关闭所有窗口
tic % 计时器计时开始
%运行参数////////////////////////////////(1)////////////////////////////////
run = 50; % 独立运行次数
L = 512; % 滤波器阶数
N = 80000; % 采样点数/迭代次数
P = 4;%仿射投影的阶数
sub_N=4; % 子带数目
load filter_bank_4.mat % hk: 分析滤波器, gk: 综合滤波器, 子带数 4
% .....系统初始化.....
load 8k512d.mat; % 稀疏系统
unknown_w1 = air512d;
unknown_w2 = -unknown_w1; % 未知系统权向量
AR = [1 -0.8]; % 1 阶自回归参数，输入信号为有色信号
mu1 = 0.1; % 固定步长
M = 3;
delta=0.001;%固定正则因子
MSD1 = zeros(1,N); % 均方权值偏差
MSD1_1 = zeros(1,N); % 均方权值偏差
MSD1_2 = zeros(1,N); % 均方权值偏差
MSD1_3 = zeros(1,N); % 均方权值偏差
%算法////////////////////////////////(2)////////////////////////////////
for g = 1:run

```

```

% .....算法初始化.....

w1 = zeros(L,N/sub_N); % 权值初始化
w1_1 = zeros(L,N/sub_N); % 权值初始化
w1_2 = zeros(L,N); % 权值初始化
w1_3 = zeros(L,N/sub_N); % 权值初始化

% .....输入信号.....

u = randn(1,N); % 高斯信号
U = filter(1,AR,u); % 有色信号

% .....期望信号.....

d1 = filter(unknown_w1,1,U(1:N/2)); % 未知系统输出信号
d2 = filter(unknown_w2,1,U(N/2+1:N));
d = [d1 d2]; % 1*80000
impulsive = binornd(1,0.01,1,N)*sqrt(10*var(d)).*randn(1,N);
% 产生-10db 冲击噪声，通过伯努利-高斯过程

D = awgn(d,10); % 加入 30db 高斯白噪声
D = D + impulsive; % 加入冲击噪声 1*80000

% SAPSA 算法////////////////////////////////////(3)////////////////////////////////////

% .....子带分解并赋值.....

sub_U = zeros(sub_N,N); % 输入信号 4*80000
sub_D = zeros(sub_N,N/sub_N); % 子带期望信号 4*20000
for j1 = 1:sub_N
    sub_U(j1,:) = filter(hk(j1,:),1,U);
    sub_d = filter(hk(j1,:),1,D);
    sub_D(j1,:) = sub_d(1:sub_N:N);
end

% .....更新过程.....

sub_x = zeros(L,sub_N*P); % 定义 512*16 的子带输入信号
sub_e1 = zeros(sub_N*P,N/sub_N); % 定义 16*20000 的子带误差信号
sign_e1 = zeros(sub_N*P,N/sub_N); % 定义 16*20000 的子带误差信号的符号矢

```

量

```

sub_Ue1 = zeros(L,N/sub_N); % 定义中间变量 512*20000
sum_Ue1 = zeros(L,N/sub_N); % 定义中间变量 512*20000
sub_DD = zeros(sub_N*P,1); % 定义 16*1 的子带期望信号
for k = L/sub_N+P:N/sub_N-1 % k 从 512/4+4 到 20000-1
    for i1 = 1:sub_N % 第 i1 个子带
        for i2 = 1:P % 第 i2 阶投影
            sub_x(:,(i1-1)*P+i2) =
                sub_U(i1,(k-i2+1)*sub_N-3:-1:(k-i2+1)*sub_N-L-2)';
            sub_DD((i1-1)*P+i2) = sub_D(i1,k-i2+1);
            sub_e1((i1-1)*P+i2,k) = sub_DD((i1-1)*P+i2) -
sub_x(:,(i1-1)*P+i2)*w1(:,k-1);
            m = sub_e1((i1-1)*P+i2,k);
            sign_e1((i1-1)*P+i2,k) = sign(m);
            sub_Ue1(:,k) = sub_Ue1(:,k) +
sub_x(:,(i1-1)*P+i2)*sign_e1((i1-1)*P+i2,k);
        end
        sum_Ue1(:,k) = sum_Ue1(:,k) + sub_Ue1(:,k);
    end
    w1(:,k) = w1(:,k-1) + mu1*sum_Ue1(:,k)/(norm(sum_Ue1(:,k))+delta);
    if k*sub_N <= N/2
        MSD1(k*sub_N) = MSD1(k*sub_N) + (norm(w1(:,k) -
unknown_w1)/norm(unknown_w1))^2;
    else
        MSD1(k*sub_N) = MSD1(k*sub_N) + (norm(w1(:,k) -
unknown_w2)/norm(unknown_w2))^2;
    end
    MSD1(k*sub_N+(1:sub_N-1)) = MSD1(k*sub_N); % 插值
end

%IAPSAF 算法////////////////////////////////////(4)////////////////////////////////
sub_x_1 = zeros(L,sub_N*P); % 子带输入信号

```

```

sub_e1_1 = zeros(sub_N*P,N/sub_N); %子带误差信号
sub_DD_1 = zeros(sub_N*P,1);
for k_1 = L/sub_N+P+M:N/sub_N-1
    for i1_1 = 1:sub_N
        for i2_1 = 1:P
            sub_x_1(:,(i1_1-1)*P+i2_1) =
sub_U(i1_1,(k_1-i2_1+1)*sub_N-3:-1:(k_1-i2_1+1)*sub_N-L-2)';
            sub_DD_1((i1_1-1)*P+i2_1) = sub_D(i1_1,k_1-i2_1+1);
        end
    end
    sub_e1_1(:,k_1) = sub_DD_1 - sub_x_1'*1/M*sum(w1_1(:,k_1-M:k_1-1),2);
    w1_1(:,k_1) = 1/M*sum(w1_1(:,k_1-M:k_1-1),2) +
mul*sub_x_1/(sub_x_1'*sub_x_1 + 0.01*eye(sub_N*P))*sub_e1_1(:,k_1);
    if k_1*sub_N <= N/2
        MSD1_1(k_1*sub_N) = MSD1_1(k_1*sub_N) + (norm(w1_1(:,k_1) -
unknown_w1)/norm(unknown_w1))^2;
    else
        MSD1_1(k_1*sub_N) = MSD1_1(k_1*sub_N) + (norm(w1_1(:,k_1) -
unknown_w2)/norm(unknown_w2))^2;
    end
    MSD1_1(k_1*sub_N+(1:sub_N-1)) = MSD1_1(k_1*sub_N); % 插值
end
%APSA 算法////////////////////////////////////(5)////////////////////////////////
x = zeros(L,P); %定义 512*4 的输入信号
e1 = zeros(P,N); %定义 4*80000 的误差信号
sign_e1_1 = zeros(P,N); %定义 4*80000 的误差信号的符号矢量
Ue1 = zeros(L,N); % 定义中间变量 512*20000
for k_2 = L+P:N-1 %k 从 512+4 到 80000-1
    for i2 = 1:P %第 i2 阶投影
        x(:,i2) = U((k_2-i2):-1:(k_2-i2+1-L))';
    end
end

```

%第 i2 阶投影输入信号的值，通过 1*80000 输入信号取第 i2 阶时间延迟信号并且以 512 列反序输出

```
e1(i2,k_2) = d(k_2-i2) - x(:,i2)*w1_2(:,k_2-1);
```

%更新 4*80000 误差矢量的第 k 列值，通过第 k 次迭代时的 4*1 期望信号减去 512*4 子带输入信号与 512*80000 的积可得

```
sign_e1_1(i2,k_2) = sign(e1(i2,k_2));
```

```
Ue1(:,k_2) = x(:,i2)*sign_e1_1(i2,k_2);
```

```
end
```

```
w1_2(:,k_2) = w1_2(:,k_2-1) + mu1*Ue1(:,k_2)/(sqrt(Ue1(:,k_2)*Ue1(:,k_2))+delta); % 滤波器权系数更新
```

```
if k_2 <= N/2
```

```
MSD1_2(k_2) = MSD1_2(k_2) + (norm(w1_2(:,k_2) - unknown_w1)/norm(unknown_w1))^2;
```

```
else
```

```
MSD1_2(k_2) = MSD1_2(k_2) + (norm(w1_2(:,k_2) - unknown_w2)/norm(unknown_w2))^2;
```

```
end
```

```
end
```

%NSAF 算法////////////////////////////////////(6)////////////////////////////////////

%.....更新过程.....

```
sub_x_3 = zeros(L,sub_N); %定义 512*16 的子带输入信号
```

```
sub_e1_3 = zeros(sub_N,N/sub_N); %定义 4*20000 的子带误差信号
```

```
sub_Ue1_3 = zeros(L,N/sub_N); % 定义中间变量 512*20000
```

```
sum_Ue1_3 = zeros(L,N/sub_N); % 定义中间变量 512*20000
```

```
sub_DD_3 = zeros(sub_N,1);%定义 4*1 的子带期望信号
```

```
for k_3 = L/sub_N:N/sub_N-1 %k 从 512/4 到 20000-1
```

```
for i1_3 = 1:sub_N %第 i1 个子带
```

```
sub_x_3(:,i1_3) = sub_U(i1_3,k_3*sub_N:-1:k_3*sub_N-L+1)';
```

%第 i1 个子带输入信号的值，通过 4*80000 子带输入信号取第 i1 行信号并且以 512 列反序输出

```
sub_DD_3(i1_3) = sub_D(i1_3,k_3);
```

%第 i1 个子带期望信号的值，通过 4*20000 子带期望信号取第 i1 行信号列的值

```
sub_e1_3(i1_3,k_3) = sub_DD_3(i1_3) -
sub_x_3(:,i1_3)*w1_3(:,k_3-1);
```

%更新 4*20000 误差矢量的第 k 列值，通过第 k 次迭代时的 4*1 期望信号减去 512*4 子带输入信号与 512*20000 的积可得

```
sub_Ue1_3(:,k_3) = sub_Ue1_3(:,k_3) + (sub_x_3(:,i1_3).*
sub_e1_3(i1_3,k_3))/((sub_x_3(:,i1_3)*sub_x_3(:,i1_3))+delta);
```

```
end
```

```
sum_Ue1_3(:,k_3) = sum_Ue1_3(:,k_3) + sub_Ue1_3(:,k_3);
```

```
w1_3(:,k_3) = w1_3(:,k_3-1) + mu1*sum_Ue1_3(:,k_3); % 滤波器权系数更新
```

```
if k_3*sub_N <= N/2
```

```
MSD1_3(k_3*sub_N) = MSD1_3(k_3*sub_N) + (norm(w1_3(:,k_3) -
unknown_w1)/norm(unknown_w1))^2;
```

```
else
```

```
MSD1_3(k_3*sub_N) = MSD1_3(k_3*sub_N) + (norm(w1_3(:,k_3) -
unknown_w2)/norm(unknown_w2))^2;
```

```
end
```

```
MSD1_3(k_3*sub_N+(1:sub_N-1)) = MSD1_3(k_3*sub_N); % 插值
```

```
end
```

```
end
```

```
%.....结果输出 MSD.....
```

```
MSD1 = MSD1/run;
```

```
MSD1_1 = MSD1_1/run;
```

```
MSD1_2 = MSD1_2/run;
```

```
MSD1_3 = MSD1_3/run;
```

```
plot(1:N,10*log10(MSD1),'b',1:N,10*log10(MSD1_1),'r',1:N,10*log10(MSD1_2),'y',1:
N,10*log10(MSD1_3),'g');
```

```
xlabel('迭代次数');
```

```
ylabel('MSD(dB)');
```



```

legend('S-APSA,P=4,N=4','IAPSAF,M=3,P=4,N=4','APSA,P=4','NSAF,N=4');
title('mu=0.1,AR=[1 -0.8],delta=0.001,-10dB 冲激噪声+5dB 高斯噪声条件下的归一
化稳态误差(db)');
grid on;
hold on;
toc

```

不同步长参数条件下的 S-APSA 算法的仿真代码

```

clc; % 擦去一页命令窗口，光标回屏幕左上角
clear all; % 从工作空间清除所有变量和函数
close all; % 关闭所有窗口
tic % 计时器计时开始
run = 50; % 独立运行次数
L = 512; % 滤波器阶数
N = 80000; % 采样点数/迭代次数
P = 4; % 仿射投影的阶数
sub_N = 4; % 子带数目
load filter_bank_4.mat % hk: 分析滤波器, gk: 综合滤波器, 子带数 4
load 8k512d.mat; % 稀疏系统
unknown_w1 = air512d;
unknown_w2 = -unknown_w1; % 未知系统权向量
AR = [1 -0.8]; % 1 阶自回归参数，输入信号为有色信号
mu1 = 0.01; % 固定步长
mu2 = 0.1; % 固定步长
mu3 = 0.8; % 固定步长
delta=0.001; % 固定正则因子
MSD1 = zeros(1,N); % 均方权值偏差
MSD2 = zeros(1,N); % 均方权值偏差
MSD3 = zeros(1,N); % 均方权值偏差
for g = 1:run

```

```

w1 = zeros(L,N/sub_N); % 权值初始化
w2 = zeros(L,N/sub_N); % 权值初始化
w3 = zeros(L,N/sub_N); % 权值初始化
u = randn(1,N); % 高斯信号
U = filter(1,AR,u); % 有色信号
d1 = filter(unknown_w1,1,U(1:N/2)); % 未知系统输出信号
d2 = filter(unknown_w2,1,U(N/2+1:N));
d = [d1 d2]; % 1*80000
impulsive = binornd(1,0.01,1,N)*sqrt(10*var(d)).*randn(1,N);
D = awgn(d,5); % 加入 30db 高斯白噪声
D = D + impulsive; % 加入冲击噪声 1*80000
sub_U = zeros(sub_N,N); % 输入信号 4*80000
sub_D = zeros(sub_N,N/sub_N); % 子带期望信号 4*20000
for j1 = 1:sub_N
    sub_U(j1,:) = filter(hk(j1,:),1,U);
    sub_d = filter(hk(j1,:),1,D);
    sub_D(j1,:) = sub_d(1:sub_N:N);
end
sub_x = zeros(L,sub_N*P); % 定义 512*16 的子带输入信号
sub_DD = zeros(sub_N*P,1); % 定义 16*1 的子带期望信号
sub_e1 = zeros(sub_N*P,N/sub_N); % 定义 16*20000 的子带误差信号
sub_e2 = zeros(sub_N*P,N/sub_N); % 定义 16*20000 的子带误差信号
sub_e3 = zeros(sub_N*P,N/sub_N); % 定义 16*20000 的子带误差信号
sign_e1 = zeros(sub_N*P,N/sub_N);
sign_e2 = zeros(sub_N*P,N/sub_N);
sign_e3 = zeros(sub_N*P,N/sub_N);
sub_Ue1 = zeros(L,N/sub_N); % 定义中间变量 512*20000
sub_Ue2 = zeros(L,N/sub_N); % 定义中间变量 512*20000
sub_Ue3 = zeros(L,N/sub_N); % 定义中间变量 512*20000
sum_Ue1 = zeros(L,N/sub_N); % 定义中间变量 512*20000

```

```

sum_Ue2 = zeros(L,N/sub_N); % 定义中间变量 512*20000
sum_Ue3 = zeros(L,N/sub_N); % 定义中间变量 512*20000
for k = L/sub_N+P:N/sub_N-1 %k 从 512/4+4 到 20000-1
    for i1 = 1:sub_N % 第 i1 个子带
        for i2 = 1:P % 第 i2 阶投影
            sub_x(:,(i1-1)*P+i2) =
sub_U(i1,(k-i2+1)*sub_N-3:-1:(k-i2+1)*sub_N-L-2)';
            sub_DD((i1-1)*P+i2) = sub_D(i1,k-i2+1);
sub_e1((i1-1)*P+i2,k) = sub_DD((i1-1)*P+i2) - sub_x(:,(i1-1)*P+i2)*w1(:,k-1) ;
            sub_e2((i1-1)*P+i2,k) = sub_DD((i1-1)*P+i2) -
sub_x(:,(i1-1)*P+i2)*w2(:,k-1) ;
            sub_e3((i1-1)*P+i2,k) = sub_DD((i1-1)*P+i2) -
sub_x(:,(i1-1)*P+i2)*w3(:,k-1) ;
            m1 = sub_e1((i1-1)*P+i2,k);
            m2 = sub_e2((i1-1)*P+i2,k);
            m3 = sub_e3((i1-1)*P+i2,k);
            sign_e1((i1-1)*P+i2,k) = sign(m1);
            sign_e2((i1-1)*P+i2,k) = sign(m2);
            sign_e3((i1-1)*P+i2,k) = sign(m3);
            sub_Ue1(:,k) = sub_Ue1(:,k) +
sub_x(:,(i1-1)*P+i2)*sign_e1((i1-1)*P+i2,k);
            sub_Ue2(:,k) = sub_Ue2(:,k) +
sub_x(:,(i1-1)*P+i2)*sign_e2((i1-1)*P+i2,k);
            sub_Ue3(:,k) = sub_Ue3(:,k) +
sub_x(:,(i1-1)*P+i2)*sign_e3((i1-1)*P+i2,k);
        end
        sum_Ue1(:,k) = sum_Ue1(:,k) + sub_Ue1(:,k);
        sum_Ue2(:,k) = sum_Ue2(:,k) + sub_Ue2(:,k);
        sum_Ue3(:,k) = sum_Ue3(:,k) + sub_Ue3(:,k);
    end
end
%—————APSA-SAF—————
    
```

```

w1(:,k) = w1(:,k-1) + mu1*sum_Ue1(:,k)/(norm(sum_Ue1(:,k))+delta); % 滤波器权系数更新

w2(:,k) = w2(:,k-1) + mu2*sum_Ue2(:,k)/(norm(sum_Ue2(:,k))+delta); % 滤波器权系数更新

w3(:,k) = w3(:,k-1) + mu3*sum_Ue3(:,k)/(norm(sum_Ue3(:,k))+delta); % 滤波器权系数更新

if k*sub_N <= N/2
    MSD1(k*sub_N) = MSD1(k*sub_N) + (norm(w1(:,k) - unknown_w1)/norm(unknown_w1))^2;
    MSD2(k*sub_N) = MSD2(k*sub_N) + (norm(w2(:,k) - unknown_w1)/norm(unknown_w1))^2;
    MSD3(k*sub_N) = MSD3(k*sub_N) + (norm(w3(:,k) - unknown_w1)/norm(unknown_w1))^2;
else
    MSD1(k*sub_N) = MSD1(k*sub_N) + (norm(w1(:,k) - unknown_w2)/norm(unknown_w2))^2;
    MSD2(k*sub_N) = MSD2(k*sub_N) + (norm(w2(:,k) - unknown_w2)/norm(unknown_w2))^2;
    MSD3(k*sub_N) = MSD3(k*sub_N) + (norm(w3(:,k) - unknown_w2)/norm(unknown_w2))^2;
end

MSD1(k*sub_N+(1:sub_N-1)) = MSD1(k*sub_N); % 插值
MSD2(k*sub_N+(1:sub_N-1)) = MSD2(k*sub_N); % 插值
MSD3(k*sub_N+(1:sub_N-1)) = MSD3(k*sub_N); % 插值

end
end

% .....结果输出 MSD.....

MSD1 = MSD1/run;
MSD2 = MSD2/run;
MSD3 = MSD3/run;
plot(1:N,10*log10(MSD1),'k',1:N,10*log10(MSD2),'b',1:N,10*log10(MSD3),'g');

```

```
xlabel('迭代次数');  
ylabel('MSD(dB)');  
legend('S-APSA,mu=0.01','S-APSA,mu=0.1','S-APSA,mu=0.8');  
title('S-APSA 在 5 种步长参数条件下的归一化稳态误差(db)');  
grid on;  
toc
```

附录 B 英文翻译

一种可以解决高背景噪声的改进投影仿射子带自适应滤波

Haiquan Zhao, Zongsheng Zheng, Zhongyu Wang, Badong Chen. Signal Processing 137(2017) 356-362

摘要

对于处理高噪声的系统，提出了一种新型的子带自适应滤波器——即改进型的仿射投影子带自适应滤波器 (I-APSAF: improved affine projection Sub-band adaptive filter)。该算法通过最小化过去权值向量与当前权值向量的差值的二范数模方来更新权值矢量，可以实现比改进的多波结构子带自适应滤波算法更快的收敛速度。

另外，一种组合步长的 IAPSAF (CSS-IAPSAF) 也发展出来用于提升 IAPSAF 的性能。用声学回波消除(AEC)的模拟仿真结果表明 I-APSAF 可以实现比已有算法（规范化的子带自适应滤波器(NSAF)、IMSAF 和改进的归一化子带自适应滤波器(INSAF)）更低的稳态误差。

改进版的 CSS-IAPSAF 也证实了在收敛速度和稳态误差上的优良性能。

【关键词】 投影仿射 子带自适应滤波器 (SAE) 声学回波消除

1. 介绍

在多变的自适应滤波算法中，最流行的是归一化最小均方(NLMS)算法[1]，通常用于系统辨识、降噪、回波消除、等等，因为它具有计算简单和鲁棒性好的优点。然而它在处理有色输入信号时的算法性能会急剧衰减。

为克服 NLMS 算法的这个缺点，[2]中提出的仿射投影算法(APA)以及相关变体拥有可以提高收敛速度和减少稳态误差的性能。不同于 NLMS 算法,APA 通过利用多个输入向量来更新权值。随着 AP 算法投影阶数的增加，收敛速度会变好，同时算法复杂度也会增加。

另外，归一化的子带自适应滤波器 **NSAF** 被提出用来克服自适应算法处理有色信号时性能衰减的问题。**NSAF** 将输入信号分解成很多近似于白信号的子带信号。它也可以实现与 **APA** 算法相同的收敛速度，同时它的算法复杂度接近于 **NLMS**。

最近一些 **NSAF** 的变体也被提出用来增加它的性能，基于 **AP** 算法，提出了改进的多波带结构的子带自适应滤波器，它像 **APA** 一样，基于多个输入矢量来推导权值矢量的更新。可知，改进型的 **IMPSAF** 的性能优于 **NSAF** 和 **APA**。[21]中提出的一些减少 **IMSAF** 算法的方法，都基于误差矢量计算和矩阵变换操作上。虽然 **IMSAF** 展示了快速的收敛速度，它仍然不能处理背景噪声特别大时性能不稳的问题。

为了实现在高背景噪声下较好的鲁棒性，提出的改进版 **NSAF** 基于修改的最小化标准（在迭代中重复使用过去的权值矢量）。改良 **I-APSAF** 算法的性能的话，可以从高信噪比环境下(AEC)仿真声学回波消除入手。**I-APSAF** 的仿真结果在第五节中给出，这说明该算法优于现有的一些算法，如 **NSAF**，**APSAF** 和 **APA**。注意 [29]中提出的 **INSAF** 与 [30]中提出的 **NSAF** 具有相同的噪声鲁棒性。**INSAF** 不仅有接近 **NSAF** 的收敛速度，还具有在搞背景噪声条件下比 **NSAF** 更低的稳态误差。得利于 [29][30]**INSAF** 的优良性能，本文作者提出了一种改进型的投影仿射自适应滤波算法，**IAPSAF**。更进一步地，也提出了一种组合步长的 **IAPSAF**（**ISS-IAPSAF**），具有更高的性能。与已存在的算法相比较，本文的贡献在于以下几点：

（1） 对高背景噪声环境，提出了 **IAPSAF**，原理是最小化更新权值矢量与过去的一些权值矢量的差的二范数模方和，从而推导出更新权值矢量的公式。因为使用了过去的权值矢量，**IAPSAF** 实现了比 **IMSAF** 更低的稳态误差。

（2） 根据 **IAPSAF** 分析的稳态误差结果。理论上，它对不同情况的仿真结果表示其性能更佳。

（3） 由于固定的步长反映了一种收敛速度和稳态误差的平衡，**CSS-IAPSAF** 被进一步提出，可以根据其变化步长来提升 **IAPSAF** 的性能。仿真结果表示，

CSS-IAPSAF 可以实现较高的收敛速度以及较低的稳态误差。

2. IMSAF

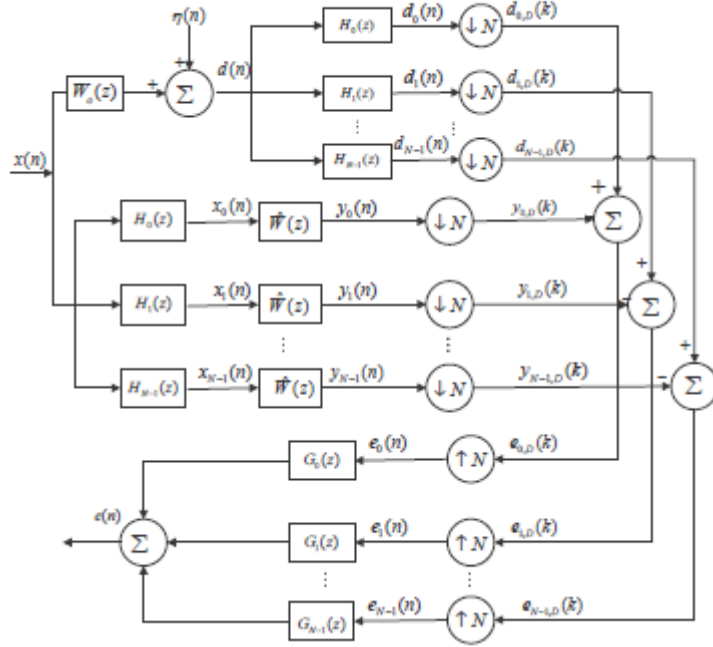


图 3 SAF 算法框图

如图 1，给出了子带自适应滤波器的结构 (SAF)。期望信号 $d(n)$ 是通过使用下面的线性模型获得的。

$$d(n) = \mathbf{x}^T(n) \mathbf{w}_0 + \eta(n) \quad (1)$$

其中， \mathbf{w}_0 表示由 L 维权矢量为模型表示的未知系统， $\eta(n)$ 表示测量噪声（误差）。 $\mathbf{x}^T(n) = [x(n), x(n-1), \dots, x(n-L+1)]^T$ 表示有 L 维输入矢量，上标 T 表示矩阵转置。期望信号 $d(n)$ 可被分为 N 个子带信号 $d_i(n)$ ，它们之间的关系可以通过分析滤波器 $H_i(Z), i = 0, 1, \dots, N-1$ 来得到，对于输入信号的衍生变量也是的一样变换原则。子带输入信号 $x_i(n)$ 可以通过该自适应滤波器得到子带输出信号 $y_i(n)$ 。子带信号 $d_i(n)$ 和 $y_i(n)$ 被降噪后分别得到 $y_{i,D}(k)$ 和 $d_{i,D}(k)$ 。这里，我们采用 n 和 k 来分别表示原始信号序列和被分解的信号序列。

另外，被降噪了的子带误差信号可以由下列公式计算：

$$e_{i,D}(k) = d_{i,D}(k) - \mathbf{x}_i^T(k) \mathbf{w}(k) \quad (2)$$

这里 $\mathbf{x}^T(n) = [x(n), x(n-1), \dots, x(n-L+1)]^T$, $\mathbf{w}(k) = [w_0(k), w_1(k), \dots, w_{L-1}(k)]^T$ 表示自适应滤波器 $\hat{\mathbf{W}}(z)$ 。

与上节表示类似，我们可分别定义输入信号矩阵，期望信号矢量，误差信号矢量如下表示：

$$\mathbf{X}(k) = [\mathbf{X}_0(k), \mathbf{X}_1(k), \dots, \mathbf{X}_{N-1}(k)] \quad (3)$$

$$\mathbf{d}_D(k) = [\mathbf{d}_{0,D}^T(k), \mathbf{d}_{1,D}^T(k), \dots, \mathbf{d}_{N-1,D}^T(k)]^T \quad (4)$$

$$\mathbf{e}_D(k) = [\mathbf{e}_{0,D}^T(k), \mathbf{e}_{1,D}^T(k), \dots, \mathbf{e}_{N-1,D}^T(k)]^T \quad (5)$$

其中：

$$\mathbf{X}_i(k) = [\mathbf{x}_i(k), \mathbf{x}_i(k-1), \dots, \mathbf{x}_i(k-P+1)] \quad (6)$$

$$\mathbf{d}_{i,D}(k) = [d_{i,D}(k), d_{i,D}(k-1), \dots, d_{i,D}(k-P+1)]^T \quad (7)$$

$$\mathbf{e}_{i,D}(k) = [e_{i,D}(k), e_{i,D}(k-1), \dots, e_{i,D}(k-P+1)]^T \quad (8)$$

这里，P 代表投影阶数。基于干扰最小原则，IMPAF 的实现模型如下^[19]：

$$\begin{aligned} \min_{\mathbf{w}(k+1)} & \|\mathbf{w}(k+1) - \mathbf{w}(k)\|^2 \\ \text{subject to} & \quad \mathbf{d}_D(k) - \mathbf{X}^T(k) \mathbf{w}(k+1) = 0 \end{aligned} \quad (9)$$

其中， $\|\cdot\|$ 表示矢量的欧式几何距离。它的权值更新公式如下：

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu \frac{\mathbf{e}_D(k) \mathbf{X}(k)}{\mathbf{X}^T(k) \mathbf{X}(k)} \quad (10)$$

其中 μ 是步长。

3. I-APSAF

3-1 引言

提议中的 I-APSAF 通过最小化更新的抽头权值和过去的 M 维抽头权值矢量的差值平方和：

$$\sum_{m=0}^{M-1} \|\mathbf{w}(k+1) - \mathbf{w}(k-m)\|^2 \quad (11)$$

$$\text{Subject to } \mathbf{d}_D(k) - \mathbf{X}^T(k)\mathbf{w}(k+1) = 0$$

整合上面的公式可以得到拉格朗日函数：

$$J(k) = \sum_{m=0}^{M-1} \|\mathbf{w}(k+1) - \mathbf{w}(k-m)\|^2 + \lambda [\mathbf{d}_D(k) - \mathbf{X}^T(k)\mathbf{w}(k+1)]^T \quad (12)$$

其中， $\lambda = [\lambda_0, \lambda_1, \dots, \lambda_{NP-1}]^T$ ， $\lambda_i, i = 0, 1, \dots, NP-1$ 是拉格朗日系数。

该拉格朗日函数对抽头权值矢量的求导结果如下：

$$\frac{\partial J(k)}{\partial \mathbf{w}(k+1)} = 2 \sum_{m=0}^{M-1} [\mathbf{w}(k+1) - \mathbf{w}(k-m)] - \lambda \mathbf{X}(k) = 0 \quad (13)$$

设该导数为 0，可得：

$$\mathbf{w}(k+1) = \frac{1}{M} \sum_{m=0}^{M-1} \mathbf{w}(k-m) + \lambda \frac{1}{2M} \mathbf{X}(k) \quad (14)$$

将以上自适应滤波权值矢量更新公式代入约束条件方程里可得：

$$\lambda \frac{1}{2M} = [\mathbf{X}^T(k)\mathbf{X}(k)]^{-1} \times \left[\mathbf{d}_D(k) - \mathbf{X}^T(k) \frac{1}{M} \sum_{m=0}^{M-1} \mathbf{w}(k-m) \right] \quad (15)$$

把以上公式又代回自适应滤波权值矢量的更新公式可得：

$$\mathbf{w}(k+1) = \frac{1}{M} \sum_{m=0}^{M-1} \mathbf{w}(k-m) + \mu \mathbf{X}(k) [\mathbf{X}^T(k)\mathbf{X}(k)]^{-1} \bar{\mathbf{e}}_D(k) \quad (16)$$

其中

$$\bar{\mathbf{e}}_D(k) = \mathbf{d}_D(k) - \mathbf{X}^T(k) \frac{1}{M} \sum_{m=0}^{M-1} \mathbf{w}(k-m) \quad (17)$$

μ 是正值常数，一般被称为步长。

然而，上述方程在数值计算上是很复杂的，因此，我们引入了一个正值约束量 δ 。现在，目的算法的更新公式就变成了：

$$\mathbf{w}(k+1) = \frac{1}{M} \sum_{m=0}^{M-1} \mathbf{w}(k-m) + \mu \mathbf{X}(k) [\mathbf{X}^T(k)\mathbf{X}(k) + \delta \mathbf{I}_{NP}]^{-1} \bar{\mathbf{e}}_D(k) \quad (18)$$

这里的 \mathbf{I}_{NP} 是一个 $NP \times NP$ 大小的特征矩阵。

3-2I-APSAF 的稳态误差分析

定义抽头误差矢量为 $\tilde{\mathbf{w}}(k) = \mathbf{w}_0 - \mathbf{w}(k)$, 则 (16) 式可以写成:

$$\begin{aligned} \tilde{\mathbf{w}}(k+1) = & \frac{1}{M} \sum_{m=0}^{M-1} \tilde{\mathbf{w}}(k-m) - \mu \mathbf{X}(k) [\mathbf{X}^T(k) \mathbf{X}(k)]^{-1} \bar{\mathbf{e}}_D(k) \\ & \times [\mathbf{X}^T(k) \frac{1}{M} \sum_{m=0}^{M-1} \mathbf{w}(k-m) + \boldsymbol{\eta}_D(k)] \end{aligned} \quad (19)$$

这里我们用到了被降噪的期望信号的关系式: $\mathbf{d}_D(k) = \mathbf{X}^T(k) \mathbf{w}_0 + \boldsymbol{\eta}_D(k)$ 。其中 $\boldsymbol{\eta}_D(k) = [\boldsymbol{\eta}_{0,D}^T(k), \boldsymbol{\eta}_{1,D}^T(k), \dots, \boldsymbol{\eta}_{N-1,D}^T(k)]^T$, $\boldsymbol{\eta}_{i,D}(k) = [\eta_{i,D}(k), \eta_{i,D}(k-1), \dots, \eta_{i,D}(k-P+1)]^T$, $\eta_{i,D}(k)$ 是第 i 阶子带噪声。

用矩阵形式重写公式 (19), 可得:

$$\tilde{\mathbf{w}}(k+1) = \tilde{\mathbf{W}}(k) \mathbf{b} - \mu \mathbf{X}(k) [\mathbf{X}^T(k) \mathbf{X}(k)]^{-1} [\mathbf{X}^T(k) \tilde{\mathbf{W}}(k) \mathbf{b} + \boldsymbol{\eta}_D(k)] \quad (20)$$

其中, $\tilde{\mathbf{W}}(k) = [\tilde{\mathbf{w}}(k), \tilde{\mathbf{w}}(k-1), \dots, \tilde{\mathbf{w}}(k-P+1)]$, $\mathbf{b} = \frac{1}{M} [1, 1, \dots, 1]^T$, 整理公式 20,

我们可以得到:

$$\tilde{\mathbf{w}}(k+1) = (\mathbf{I}_L - \mu \mathbf{X}(k) [\mathbf{X}^T(k) \mathbf{X}(k)]^{-1} \mathbf{X}^T(k) \tilde{\mathbf{W}}(k) \mathbf{b} - \mu [\mathbf{X}^T(k) \mathbf{X}(k)]^{-1} \boldsymbol{\eta}_D(k)] \quad (21)$$

假设噪声 $\eta(n)$ 是一个零均值的高斯过程, 并且信号 $\eta_{i,D}(k)$, $\mathbf{x}_i(k)$, $\tilde{\mathbf{w}}(k)$ 是统计相关的^[33-36]。均方误差被定义为 $\text{MSD}(k) = E[\tilde{\mathbf{w}}^T(k) \tilde{\mathbf{w}}(k)]$ 。[23] 文谈到的关于欧式模方和数学期望, 我们有:

$$\text{MSD}(k+1) = E[\mathbf{b}^T \tilde{\mathbf{W}}^T(k) \Lambda^T(k) \Lambda(k) \tilde{\mathbf{W}}(k) \mathbf{b}] + \mu^2 E[\boldsymbol{\eta}_D^T(k) [\mathbf{X}^T(k) \mathbf{X}(k)]^{-1} \boldsymbol{\eta}_D(k)] \quad (22)$$

其中 $\Lambda(k) = \mathbf{I}_L - \mu \mathbf{X}(k) [\mathbf{X}^T(k) \mathbf{X}(k)]^{-1} \mathbf{X}^T(k)$ 。

设子带输入信号 $\mathbf{x}_i(k)$ 近似为白信号, 子带输入信号的能量 $\|\mathbf{x}_i(k)\|^2$ 波动在一次中的迭代下会逐渐变小^[33-36]。很自然地, 我们可以得到以下近似:

$$\begin{aligned} E[\Lambda^T(k) \Lambda(k)] = & E[\mathbf{I}_L - 2\mu \mathbf{X}(k) [\mathbf{X}^T(k) \mathbf{X}(k)]^{-1} \mathbf{X}^T(k) + \mu^2 \mathbf{X}(k) [\mathbf{X}^T(k) \mathbf{X}(k)]^{-1} \mathbf{X}^T(k)] \\ \approx & \left(1 - \frac{2\mu NP}{L} + \frac{\mu^2 NP}{L} \right) \mathbf{I}_L \end{aligned} \quad (23)$$

$$E[\boldsymbol{\eta}_D^T(k)[\mathbf{X}^T(k)\mathbf{X}(k)]^{-1}\boldsymbol{\eta}_D(k)] \approx \sum_{i=0}^{N-1} \frac{P^2 \sigma_{\eta_{i,D}}^2}{L \sigma_{x_i}^2} \quad (24)$$

其中，我们用到了关系式 $E[\mathbf{x}_i^T(k)\mathbf{x}_i(k)] \approx L \sigma_{x_i}^2$ ， $E[\mathbf{x}_i(k)\mathbf{x}_i^T(k)] \approx \sigma_{x_i}^2 \mathbf{I}_L$ ， \mathbf{I}_L 是 $L \times L$ 的特征矩阵^[33-36]。(23) 和 (24) 代入 (22) 可得：

$$\text{MSD}(k+1) = \left(1 - \frac{2\mu NP}{L} + \frac{\mu^2 NP}{L}\right) \mathbf{b}^T E[\tilde{\mathbf{W}}^T(k)\tilde{\mathbf{W}}(k)] \mathbf{b} + \mu^2 \sum_{i=0}^{N-1} \frac{P^2 \sigma_{\eta_{i,D}}^2}{L \sigma_{x_i}^2} \quad (25)$$

如[34]文，我们用相同的方法推导期望 $E[\tilde{\mathbf{W}}^T(k)\tilde{\mathbf{W}}(k)]$ ，得到：

$$E[\tilde{\mathbf{W}}^T(\infty)\tilde{\mathbf{W}}(\infty)] = \begin{bmatrix} 1 & \frac{A}{1-(M-1)A} & \cdots & \frac{A}{1-(M-1)A} \\ \frac{A}{1-(M-1)A} & 1 & \cdots & \frac{A}{1-(M-1)A} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{A}{1-(M-1)A} & \frac{A}{1-(M-1)A} & \cdots & 1 \end{bmatrix} \quad (26)$$

其中， $A = \frac{1}{M}(1 - \frac{\mu NP}{L})$ 。在稳态时， $\text{MSD}(k+1) = \text{MSD}(k) = \text{MSD}(\infty)$ 。那么

(25) 可被写成：

$$\text{MSD}(\infty) = \frac{1}{M} B \left(\frac{A}{1-(M-1)A} \right) \text{MSD}(\infty) + \mu^2 \sum_{i=0}^{N-1} \frac{P^2 \sigma_{\eta_{i,D}}^2}{L \sigma_{x_i}^2} \quad (27)$$

其中 $B = 1 - \frac{2\mu NP}{L} + \frac{\mu^2 NP}{L}$ 。

将上式化简整理后，稳态 MSD 可以由以下式子获得：

$$\text{MSD}(\infty) = \frac{\mu^2 M(1-MA+A)}{M(1-MA+A)-B} \sum_{i=0}^{N-1} \frac{P^2 \sigma_{\eta_{i,D}}^2}{L \sigma_{x_i}^2} \quad (28)$$

3-2 总结讨论

与 IMSAF 的最小化标准相比较，IMSAF 最小化了当前抽头权值矢量 $\mathbf{w}(k+1)$ 和过去的抽头权值矢量 $\mathbf{w}(k)$ 的差值，I-APSAF 的最小化标准是最小化当前抽头权值矢量 $\mathbf{w}(k+1)$ 和过去 M 个抽头权值矢量 $\mathbf{w}(k)$ ， $\mathbf{w}(k-1)$ ， \cdots ， $\mathbf{w}(k-M+1)$ 的差值，然后，对于 IAPSAF 来说， $\mathbf{w}(k+1)$ 的更新依赖于过去 M 个抽头权值矢量，这已经在[32]文

中进行过集合诠释了。所以，所提议的 IAPSAF 拥有在测验噪声相当高情况下避免抽头权值矢量较大的波动的性能。

在已研究的稳态 MSD 下，探索 M 的效果。首先，我们重写 (28) 式可得：

$$\begin{aligned} \text{MSD}(\infty) &= \frac{\mu^2 M(1-MA+A)}{M(1-MA+A)-B} \sum_{i=0}^{N-1} \frac{P^2 \sigma_{\eta_{i,D}}^2}{L \sigma_{x_i}^2} \\ &= \frac{\mu^2 (M-1+L/(\mu NP))}{M-1-\mu} \sum_{i=0}^{N-1} \frac{P^2 \sigma_{\eta_{i,D}}^2}{L \sigma_{x_i}^2} \end{aligned} \quad (29)$$

我们可以定义一个关于 M 的函数：

$$F = \frac{\mu^2 (M-1+L/(\mu NP))}{M-1-\mu} \quad (30)$$

将 (30) 对 M 进行求导可得：

$$\frac{\partial F}{\partial M} = \frac{\mu^2 (M-1+L/(\mu NP))}{(M-1-\mu)^2} \quad (31)$$

使步长因子满足 $0 < \mu < 2$ 【19-33】， $L \gg NP$ ，函数 F 的导数比 0 小。这样，函数 F 是 M 的单调递减函数。因此稳态 MSD 对着 M 值得增加而减小，在仿真中我们也得到相同的结果。

4. CSS-IAPSAF

固定的步长大小反映一种快速收敛率和低稳态误差的平衡。为了解决这个问题，两个滤波器用不同步长值的凸组合技术在 [37]–[41] 中被提出。凸组合技术的原理是在大步长的滤波器下获得快收敛率的同时在小步长的滤波器下获得低稳态误差。很明显，用凸组合技术的话会增加算法在计算和实现上的复杂度。我们现在提出一种只用一个滤波器的变步长方案。所提方案的权值更新公式如下：

$$\mathbf{w}(k+1) = \lambda(k) \mathbf{w}_1(k+1) + (1+\lambda(k)) \mathbf{w}_2(k+1) \quad (32)$$

$$\mathbf{w}_1(k+1) = \frac{1}{M} \sum_{m=0}^{M-1} \mathbf{w}(k-m) + \mu_1 \mathbf{X}(k) \times [\mathbf{X}^T(k) \mathbf{X}(k) + \delta \mathbf{I}_{NP}]^{-1} \tilde{\mathbf{e}}_D(k) \quad (33)$$

$$\mathbf{w}_2(k+1) = \frac{1}{M} \sum_{m=0}^{M-1} \mathbf{w}(k-m) + \mu_2 \mathbf{X}(k) \times [\mathbf{X}^T(k) \mathbf{X}(k) + \delta \mathbf{I}_{NP}]^{-1} \tilde{\mathbf{e}}_D(k) \quad (34)$$

其中 $\lambda(k)$ 是一个混合参数, 包括了两个 IAPSAF, (33) 式表示的是大步长, (34) 式表示的是小步长。组合 (32)–(36) 可以得到:

$$\mathbf{w}(k+1) = \frac{1}{M} \sum_{m=0}^{M-1} \mathbf{w}(k-m) + \mu_c(k) \mathbf{X}(k) \times [\mathbf{X}^T(k) \mathbf{X}(k) + \delta \mathbf{I}_{NP}]^{-1} \tilde{\mathbf{e}}_D(k) \quad (35)$$

其中 $\mu_c(k) = [\lambda(k)\mu_1 + (1-\lambda(k))\mu_2]$ 是一个从 μ_2 到 μ_1 的组合步长。与凸组合技术类似^{【37-41】}, 由 $\beta(k)$ 构成的一个用来估算 $\lambda(k)$ 的 c 形函数可以这样定义:

$$\lambda(k) = \frac{1}{1 + e^{\beta(k)}} \quad (36)$$

为了最小化平方误差, $\beta(k)$ 的更新用梯度下降法。

$$\begin{aligned} \beta(k) &= \beta(k-1) - \frac{\mu_\beta}{2} \frac{\partial \sum_{i=0}^{N-1} e_{i,D}^2(k)}{\partial \beta(k-1)} \\ &= \beta(k-1) + \mu_\beta \sum_{i=0}^{N-1} e_{i,D}(k) \mathbf{X}_i^T(k) (\mu_1 - \mu_2) \lambda(k-1) \\ &\quad \times (1 - \lambda(k-1)) \mathbf{X}(k-1) [\mathbf{X}^T(k-1) \mathbf{X}(k-1) + \delta \mathbf{I}_{NP}]^{-1} \tilde{\mathbf{e}}_D(k) \end{aligned} \quad (37)$$

其中 μ_β 是为适应 $\beta(k)$ 的学习速率。

【注1】 如[46]文中讨论, (36) 式中的混合参数从 0 到 1 的变化是非常缓慢的, 这降低了算法的性能。所以, [42]提出了一种改进的 c 形函数:

$$\lambda(k) = \frac{1}{1 + e^{\beta(k)}} - \left(\frac{C}{2} - \frac{1}{2} \right) \quad (38)$$

$$\beta(k) = \begin{cases} \beta_c, & \beta(k) \geq \beta_c \\ \beta(k), & -\beta_c \leq \beta(k) < \beta_c \\ -\beta_c, & \beta(k) < -\beta_c \end{cases} \quad (39)$$

其中 $\beta_c = \ln(C+1)/(C-1)$, C 是一个大于 1 的正常数。注意当 $C=1$ 时, (38) 式变成了 (36) 式。

【注2】 如 (37) 式, $\beta(k)$ 的更新在 $\lambda(k)=0$ 或者 1 时就会停止。为了克服这个缺点, (37) 式可以改写成这样:

$$\begin{aligned}
\beta(k) = & \beta(k-1) + \mu_\beta \sum_{i=0}^{N-1} e_{i,D}(k) \mathbf{X}_i^T(k) (\mu_1 - \mu_2) \\
& \times [\lambda(k-1)(1-\lambda(k-1)) + \xi] \mathbf{X}(k-1) \\
& \times [\mathbf{X}^T(k-1) \mathbf{X}(k-1) + \delta \mathbf{I}_{NP}]^{-1} \tilde{\mathbf{e}}_D(k)
\end{aligned} \tag{40}$$

其中， ξ 是一个接近为 0 的很小的正常数。

5. 计算复杂度

如表 1，整数 H 表示典型滤波器的长度[3]，展示了所提算法与 NSAF[7]，IMSAF[19]和 INSAF[29]相比较下的计算复杂度。因为 IMSAF 算法中用到了多个输入矢量，IMSAF 的算法复杂度大于 NSAF 的算法复杂度。比较 IMSAF 与 NSAF，他们都需要在计算过去系数矢量的平均值上进行 L 次操作。CSS-IAPSAF 需要 L 次算术操作来计算组合步长。

表格 1 各个算法复杂度的比较

算法	乘法数	加法数	存储数
NSAF	$3L+HN+2H+2$	$3L+HN+2H-N-2$	$LN+L+3HN+4N+2$
INSAF	$4L+HN+2H+2$	$ML+2L+HN+2H-N-2$	$LN+ML+L+3HN+4N+2$
IMSAF	$2NP^2L+2PL+HN+2H+P$	$2NP^2L-NP^2+PL+HN+2H-N-2$	$NPL+L+N^2P^2+3HN+2NP$
IAPSAF	$2NP^2L+2PL+HN+2H+P+L$	$2NP^2L-NP^2+PL+ML+HN+2H-L-N-2$	$NPL+ML+L+N^2P^2+3HN+2NP$
CSS-IAPSAF	$2NP^2L+2PL+HN+2H+P+2L+9$	$2NP^2L-NP^2+PL+ML+HN+2H-N+7$	$NPL+ML+2L+N^2P^2+3HN+2NP+6$

6. 仿真结果

本文所提的 I-APSAF 和 CSS-IAPSAF 将在 AEC（声学回声消除）应用上进行仿真测试。回声路径将采用[26]文中的 512 个系数模型。自适应滤波器的长度也被设置为 512。为了比较算法的跟踪性能，回声路径的冲激响应将在迭代中乘以-1。输入信号为自回归（AR）信号或者语音信号[26]。输出信号的回声路径将会由不同信噪比的高斯白噪声组成[32]。我们采用正弦调制滤波器，子带数目是 4[17][18]，投影阶数是 $N=4$ [33]。归一化误差期望被定义为： $10\log_{10}\left(E[\|\mathbf{w}(k)-\mathbf{w}_0\|_2^2]/\|\mathbf{w}_0\|_2^2\right)$ ，它被用来测量所提议算法的性能。仿真结果将取 50 个仿真结果的平均值。

6.1 AR 输入信号

图 2、图 3 展示了算法 I-APSAF 在参数 $M=1,2,3$, $\text{SNR}=5, 20\text{dB}$ 下的仿真曲线，

其中，输入信号为一个以 0.8 为极点的自回归信号 AR(1)。从比较结果上看，在 I-APSAF 算法下我们可以得到一个较低的稳态误差，并且这个误差还低于相同 M 参数下 IMSAF 算法的稳态误差。另外，随着 M 值的增加，I-APSAF 性能更好，其稳态误差更低。注意 IAPSAF 的收敛速率随着 M 值得增加而降低，在高 SNR 下这个趋势更加明显。

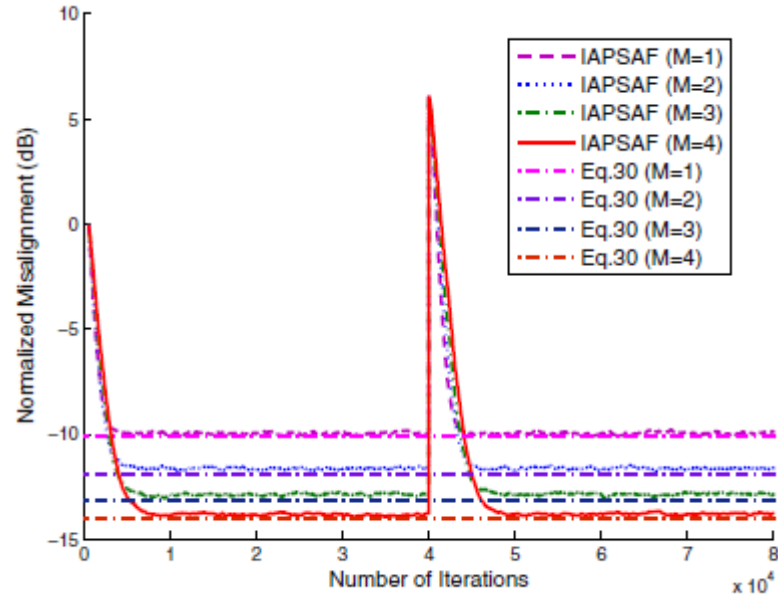


图 4 IAPSAF 算法在 $M = 1, 2, 3$, SNR= 5 dB, $\mu=0.2$, $\delta=0.01$ 下的归一化稳态误差

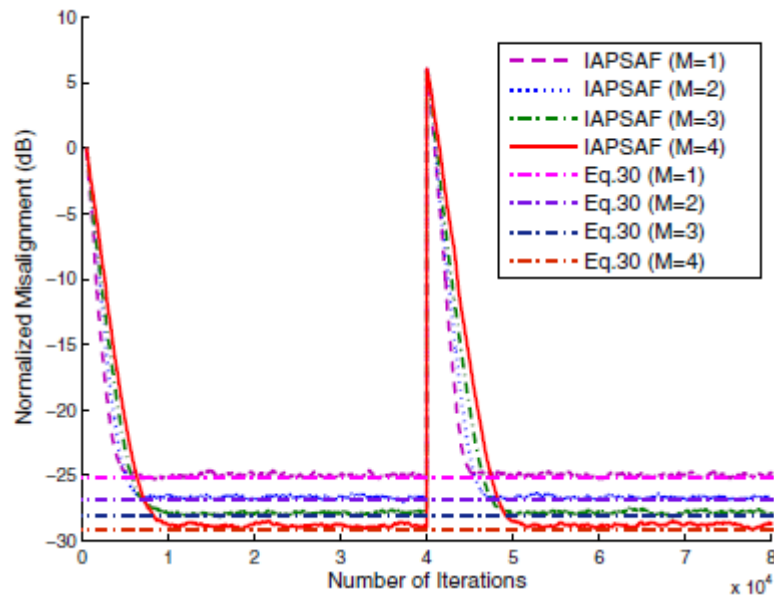


图 5 IAPSAF 算法在 $M = 1, 2, 3$, SNR= 20 dB, $\mu=0.2$, $\delta=0.001$ 下的归一化稳态误差

另外,如图 2 和图 3 所示,对于不同 SNR 值的仿真结果也与理论结果很匹配。尽管在理论和仿真结果上由轻微的不同,我们用了一些假设和近似来简化了模型,所以这些轻微的差异也是很合理的。

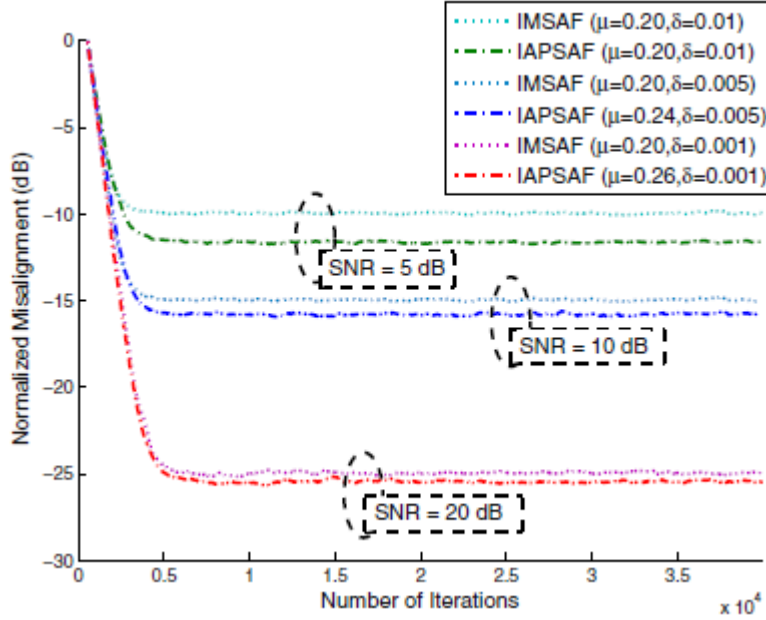


图 6 IMSAF/IAPSAF 算法在 $M = 2$, $\text{AR}(1)$ 输入信号下的归一化稳态误差

图 4 测试了 IAPSAF 在不同 SNR 下的性能。还可以看出,与 IMSAF 相比,在相同的初试收敛速率下 IAPSAF 可以提供更低的稳态误差,这种优势在更低的 SNR 下尤其明显。注意与 IMSAF 相比, IAPSAF 改进的优势仅在 SNR 低于 20dB 时,即 IAPSAF 在信噪比低于 20dB 时,对于高背景噪声环境的信号处理很有优势。

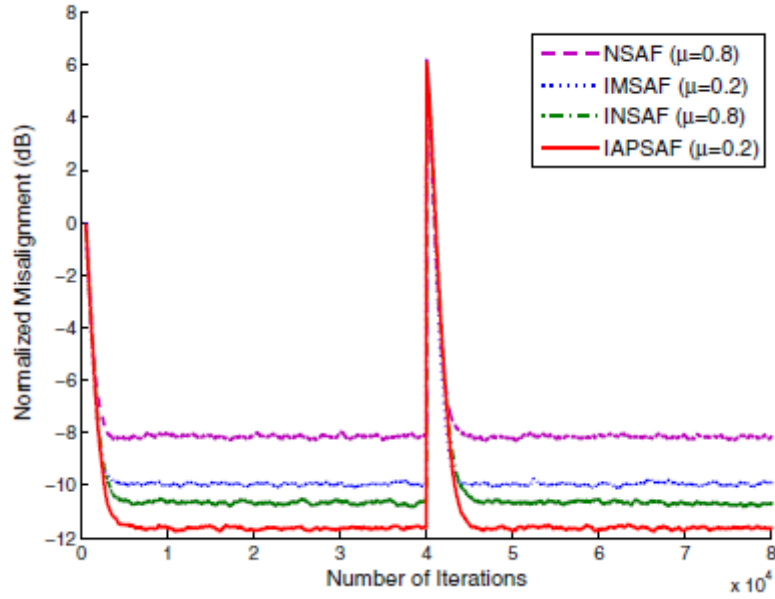


图 7 NSAF/IMSAF/INSAF/IAPSAF 算法在 $M = 2$, $\text{SNR} = 5 \text{ dB}$, $\delta = 0.01$, AR(1) 输入信号下的归一化稳态误差

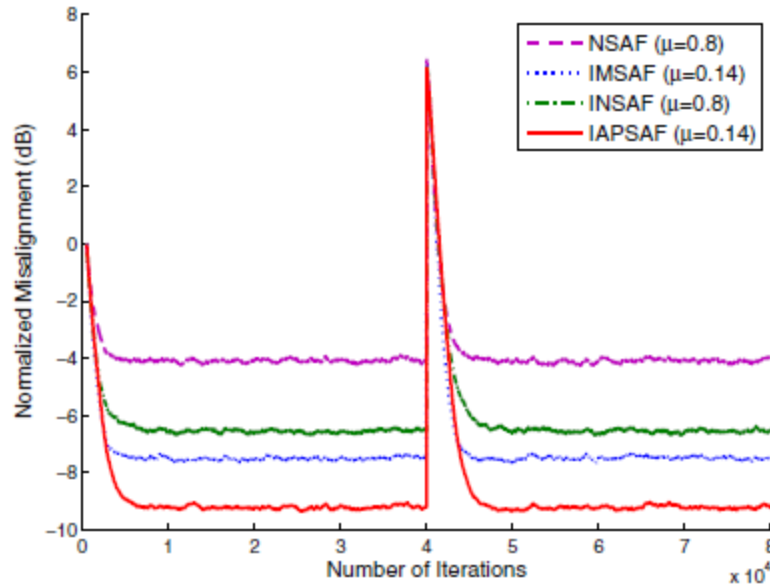


图 8 NSAF/IMSAF/INSAF/IAPSAF 算法在 $M = 2$, $\text{SNR} = 5 \text{ dB}$, $\delta = 0.01$, AR(2) 输入信号下的归一化稳态误差

算法 NSAF[17]、IMSAF[19]、INSAF[29]以及 IAPSAF 的性能比较如图 5、图 6 所示，其中，输入信号分别为极点为 0.8 的自回归信号 AR(1)和 AR(2)，其系数参数为(1, -1.6, 0.81)[33]。如图所示，NSAF 的稳态误差最高，其他算法自然相对来说稳态误差性能有所提升。比较 IMSAF, INSAF 和 IAPSAF，当他们在相同的初

试收敛速率下，IAPSAF 具有最低的稳态误差。

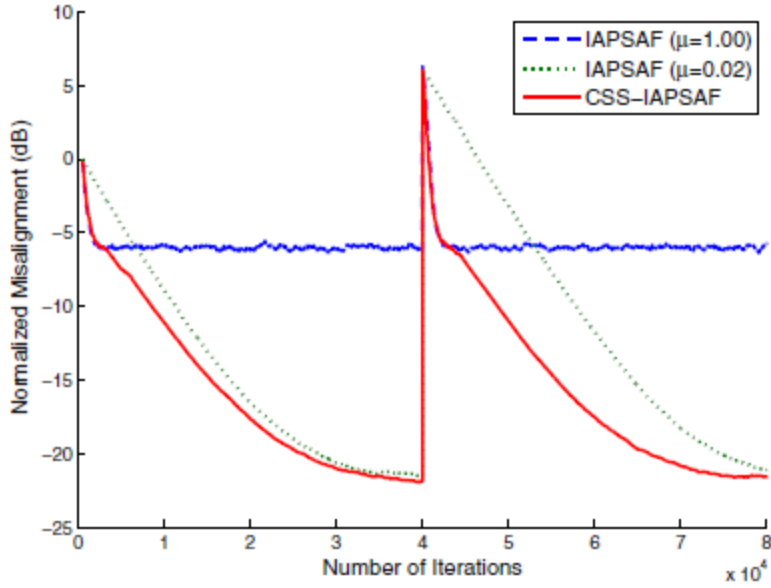


图 9 IAPSAF 和 CSS-IAPSAF 算法在，AR(1)输入信号下的归一化稳态误差
其中 $M = 2$, $\text{SNR} = 5 \text{ dB}$, $\delta = 0.01$, $C = 2$, $\mu_\beta = 50$, $\xi = 10 \times 10^{-10}$

另外，图 7 是我们测试 CSS-APSAF 和 IAPSAF 之间的性能情况。其中 AR(1) 是极点为 0.8 的自回归信号。如图所示，有更大的步长的 IAPSAF 拥有更快的收敛速率，但同时也意味着比较高的稳态误差。相反，更小步长的 IAPSAF 有较低的稳态误差，但是收敛速率比较慢。比较组合步长的 IAPSAF 算法，CSS-IAPSAF 既有较快的收敛速率，又有较低的稳态误差。

6.1 语音输入信号

我们还用语音输入信号对所提算法进行了 AEC 环境下的性能测试。

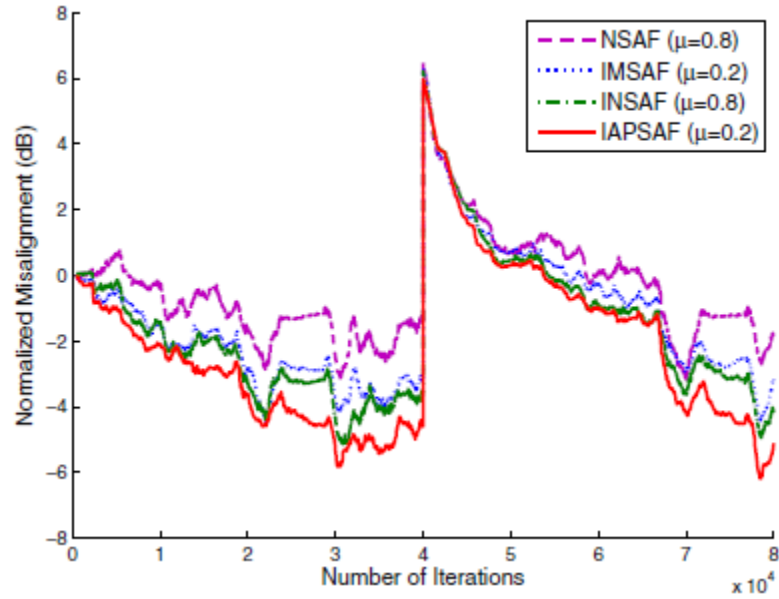


图 10 NSAF/IMSAF/INSAF/IAPSAF 算法在 $M = 2$, $\text{SNR} = 5 \text{ dB}$, $\delta = 0.01$, 语音输入信号下的归一化稳态误差

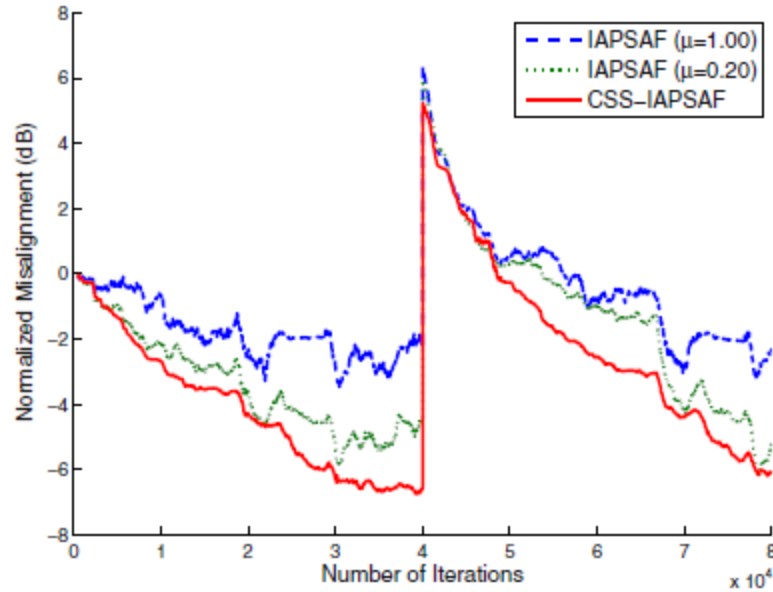


图 11 IAPSAF 和 CSS-IAPSAF 算法在，语音输入信号下的归一化稳态误差
其中 $M = 2$, $\text{SNR} = 5 \text{ dB}$, $\delta = 0.01$, $C = 2$, $\mu_\beta = 50$, $\xi = 10 \times 10^{-10}$

图 8 和图 9 中，我们比较了 NSAF, IMSAF, INSAF, IAPSAF 和 CSS-IAPSAF 的性能。图 8 和图 9 仿真中用到的是相同的语音输入信号，图中可以看出所提的 IAPSAF 算法比传统的自适应滤波算法有更好的性能，另外 CSS-IAPSAF 也在收敛速度和稳态误差的平衡上具有很大的提升。

7. 总结

本文提出了一种改进型的投影仿射自适应滤波算法 **I-APSAF**，它能够保证在高背景噪声环境下的低稳态误差。所提算法的自适应更新公式是由最小化当前权值矢量和过去 M 个权值矢量的差值平方的期望值导出的。另外，为了进一步提升 **IAPSAF** 的性能，本文又提出了一种组合步长的 **IAPSAF**。在存在高背景噪声的声学回声消除应用下的仿真情况也证实了所提算法比已知的其他算法更好的性能。

鸣谢

本研究成果由中国国家自然科学基金会（补助编码：61571374, 61271340 和 61433011）联合赞助。本研究成果感谢联合编者和评阅人员们宝贵的评论和建议，他们给了我们的稿子极大地提升。