

# 专 业 学 位 硕 士 学 位 论 文

## 有源噪声控制技术研究

Research on Active Noise Control Techniques

作 者 姓 名: 林坤鹏

工 程 领 域: 电子与通信工程

学 号: 31809070

指 导 教 师: 殷福亮 教授

陈 喆 教授

完 成 日 期: 2021 年 5 月

大连理工大学

Dalian University of Technology

## 摘 要

随着数字信号处理的发展和集成电路的进步，有源降噪产品应运而生，并得到消费者的青睐。近年来，有源降噪处理技术已取得一定进展，并在实际中得到应用。但在实际系统中，由于时延等因素的影响，现有的降噪技术对低频段噪声衰减效果明显，而对周边环境中的中高频干扰抑制能力不足，这严重影响了降噪系统的性能。

为了解决上述问题，本文应用非线性滤波和深度神经网络理论，对有源降噪技术进行了研究，主要工作如下：

(1) 基于语音非线性生成模型，提出了一种基于非线性卡尔曼噪声预测的有源噪声控制系统，以提升对干扰语音的衰减能力。该系统考虑了参考噪声的时延，并增加了噪声预测模块和延迟估计模块。其中：噪声预测模块通过扩展卡尔曼滤波方法，预测未来时刻系统噪声采样并依次补偿缓存；延迟估计模块通过计算多组缓存信号与噪声之间的相关性，得到系统的时延估计值，并将相应的噪声补偿信号通入系统参与反相控制噪声的计算。实验结果表明，该系统提升了对干扰语音的衰减能力，并且对常见环境噪声也具有良好的抑制效果。

(2) 在 FxLMS 算法中，次级路径的建模准确度会影响整体算法的降噪性能。为了提升次级路径的建模精度，提出了基于门控循环网络(GRU)的次级路径在线建模算法。该算法将 GRU 网络作为次级路径建模器，并且进行在线训练。实验结果表明，该算法具有建模误差小、抗突变性能强的优点，有效提升了系统的降噪性能。

**关键词：**有源噪声控制；FxLMS 算法；扩展卡尔曼滤波；GRU 网络

## Research on Active Noise Control Techniques

### Abstract

With the development of digital signal processing and the progress of integrated circuits, active noise cancellation products have emerged and are favored by consumers. In recent years, active noise cancellation processing techniques has made certain progress and has been applied in practice. However, in the actual system, due to the influence of time delay and other factors, the existing noise cancellation techniques has a significant effect on low-frequency noise attenuation, while the ability to suppress mid- and high-frequency interference in the surrounding environment is insufficient, which seriously affects the performance of the noise reduction system .

In order to solve the above problems, this thesis applies nonlinear filtering and deep neural network theory to research active noise cancellation techniques. The main work is as follows:

(1) Based on the nonlinear voice generation model, an active noise control system based on nonlinear Kalman noise prediction is proposed to improve the attenuation ability of interfering voice. The system considers the time delay of the reference noise, and adds a noise prediction module and a delay estimation module. Among them: the noise prediction module uses the extended Kalman filter method to predict the system noise samples in the future and compensate the buffer in turn; The delay estimation module obtains the system delay estimation value by calculating the correlation between multiple groups of buffered signals and noise, and passes the corresponding noise compensation signal into the system to participate in the calculation of the antiphase control noise. The experimental results show that the system improves the attenuation ability of interfering voice, and it also has a good suppression effect on common environmental noises.

(2) In the FxLMS algorithm, the modeling accuracy of the secondary path will affect the noise reduction performance of the overall algorithm. In order to improve the accuracy of secondary path modeling, an online secondary path modeling algorithm based on Gated Recurrent Network (GRU) is proposed. The algorithm uses the GRU network as a secondary path modeler and conducts online training. Experimental results show that the algorithm has the advantages of small modeling errors and strong anti-mutation performance, which effectively improves the noise cancellation performance of the system.

**Key Words:** Active Noise Control; FxLMS; Extended Kalman Filter; GRU network

## 目 录

摘 要.....	I
Abstract.....	II
1 绪论.....	1
1.1 研究背景及意义.....	1
1.2 有源噪声消除研究历史及现状.....	2
1.3 本文研究内容和章节安排.....	4
2 有源噪声控制理论基础.....	5
2.1 自适应控制算法.....	5
2.1.1 LMS 算法.....	5
2.1.2 LMS 算法性能分析.....	7
2.1.3 FxLMS 算法.....	9
2.1.4 次级路径辨识算法.....	10
2.2 时间序列预测方法概述.....	13
2.2.1 传统时序预测方法.....	13
2.2.2 卡尔曼滤波法.....	13
2.2.3 人工神经网络法.....	16
2.3 门控循环神经网络.....	16
2.3.1 循环神经网络.....	17
2.3.2 长短时记忆网络.....	17
2.3.3 门控循环神经网络.....	19
2.4 本章小结.....	20
3 基于噪声预测的有源噪声控制系统.....	21
3.1 有源噪声控制信号模型.....	21
3.2 基于噪声预测的噪声控制系统结构.....	23
3.3 噪声预测算法选择.....	26
3.3.1 各算法实施方案概述.....	26
3.3.2 预测算法比较与分析.....	30
3.4 实验结果及分析.....	32
3.4.1 纯净语音性能对比.....	32
3.4.2 带噪语音性能对比.....	33
3.5 本章小结.....	33

4 基于 GRU 网络的次级路径在线建模.....	35
4.1 次级路径在线建模算法分析.....	35
4.1.1 辅助噪声功率要求.....	36
4.1.2 最佳步长选取要求.....	38
4.1.3 建模算法的抗突变要求.....	39
4.1.4 独立性要求.....	39
4.2 基于 GRU 网络的次级路径在线建模.....	39
4.2.1 现有建模算法分析.....	39
4.2.2 基于 GRU 网络的在线建模算法结构.....	43
4.3 实验结果与分析.....	44
4.3.1 实验配置.....	44
4.3.2 各影响因素下实验对比.....	45
4.4 本章小结.....	49
结    论.....	50
参 考 文 献.....	51
攻读硕士学位期间发表学术论文情况.....	54
致    谢.....	55

# 1 绪论

## 1.1 研究背景及意义

随着科学技术的迅猛发展，人类社会进入到工业社会，在享受着科技发展所带来的成果同时，人们也逐渐发觉技术发展所带来一些副作用，噪声污染就是其中之一。由于世界各地人口增长和发展的结果，持续的城市化和人口密集化，以及对交通出行的需求日益增加，人们越来越趋向于使用机械自动化设备用于建筑、交通等生活设施，导致人们在生活中容易受到各种噪声的干扰<sup>[1]</sup>。生活中，人们的居住环境早已经被各种噪声所包围，尤其在网吧、餐馆、KTV 等场景下，噪声污染更是附近居民头疼的问题。在工业生产中，由生产加工和设备运行时产生的噪声平均超过 60dB，工人长期暴露在这种环境下，对人的生理和心理健康都会产生严重的危害，并且可能会导致某些设备的声疲劳，缩短其使用寿命，甚至会导致生产事故的发生。汽车、飞机、轮船等交通工具在运行时机械运转产生的噪声，以及鸣笛、刹车时的尖锐噪声，占据了城市噪声中的 30%。在军事中，武器装备的噪声不仅会影响作战人员的状态，不必要的噪声还可能会导致行踪暴露，从而影响战局。根据数据统计，世界上有 70%的人口（主要为城镇居民）不同程度的受到噪声污染的危害，有超过 30%的人在夜晚受到噪声的影响导致睡眠不足，并且每年至少有超过一百万人因为受到噪声干扰而严重影响身体健康，因此噪声污染对人类健康的影响仅次于空气污染，并且是永久性的伤害<sup>[2]</sup>。随着噪声越来越影响人类的正常生活，人们对降低噪声干扰的需求也在逐渐增加。

从方式上来说，控制噪声可以从噪声源、传播路径和噪声接收端三个方面进行<sup>[3]</sup>。控制噪声源，不让其产生噪声或者控制噪声的能量，是从根本上解决噪声污染的问题，但是在很多场景下，噪声的产生是由多重因素共同作用的结果，是不可控制的过程。并且在工业生产中，大型机械的运转和必要的人员沟通协作是无法避免的。在传播路径上对噪声进行衰减只能针对某些特定的场景，例如高速公路旁的设立的隔音板就是为了减轻高速公路上的噪声对周边居民的影响，但是这种在传播路径上进行阻断的场景单一，成本高昂，并且不能覆盖生活中的全部场景。在接收端进行噪声控制，随着近年智能终端的火热成为了有源噪声控制的热门方向。

传统的噪声消除依据声波的传播理论，使用吸收、隔离、阻尼、结构消声等方法，这种方法称为无源噪声消除(Passive Noise Cancellation, PNC)<sup>[4]</sup>理论。无源噪声控制理论主要是依靠不同的隔音材料或者吸声结构来隔绝、吸收噪声能量，在传播路径中被动地衰减到噪声以达到降噪的目的。被动降噪的优势主要在于对中、高频的噪声具有很强的

衰减能力,但是对于低频类噪声无法带来理想的降噪效果,并且对于小空间的噪声,例如空调工作、飞机发动机的噪声,衰减效果就大打折扣<sup>[5,6]</sup>。一些相应的设备体积庞大,价格昂贵,应用场合也受到了限制。

于是,针对一些无法采用无源噪声控制技术的场景,为了提升对中低频噪声的衰减能力,人们提出了一种新的解决方法,即有源噪声消除(Active Noise Cancellation, ANC)理论,该理论通过发出反相控制噪声与待衰减噪声相互抵消来实现降噪的目的。有源噪声控制技术在低频噪声控制上具有较为突出并且稳定的效果,并凭借其低成本以及易于维护的特点,在工业生产、交通出行、工作学习等诸多领域得到广泛应用。随着 ANC 性能的不断提高,稳定性的持续上升,以及成本的日益下降,有源噪声控制理论逐渐成为解决低频类噪声控制问题的首要方法。

## 1.2 有源噪声消除研究历史及现状

有源噪声控制理论源自于德国物理学家 Paul Leug 在 1936 年提出的一项专利<sup>[7]</sup>,并且列举了一部分可能的应用。这项专利利用了一项常见的声学现象:当两列频率相同,相位固定的声波叠加时,可能产生相加干涉或者相消干涉,因此就能增强或者衰减声波。Leug 设想如果能够利用扬声器发出和噪声频率相同,相位完全相反的声波,也就是发出噪声声波的镜像,就能在扬声器处实现噪声的消除。但是当时科学条件的限制无法满足两个必需的条件:(1)准确的测量噪声声波从采集的位置到扬声器位置所需要的时延;(2)电子器材需要具有良好的幅频和相频特性。

在 1953 年,Harry 发表了两篇论文<sup>[8,9]</sup>。论文中提出了一种电子吸声器的结构,其制造静区的原理和 Leug 所提出的专利相似,都体现了有源噪声控制的思想。虽然所提出的结构在一定频率范围内能够实现很好的效果,但是受限于电子器件的性能,导致吸声的频率范围收到极大的限制,所以很难在实际中使用。

有源噪声控制理论为了适应噪声的变化,对于系统时延的要求极高,但是受限于当时使用的模拟电子器件的精度较低,并没有办法实现这一理论。直到 1970 年代末期,随着电子技术的飞速发展,有源噪声控制技术在数字信号处理机(Digital Signal Processor, DSP)上易于实时处理,这就使更多的学者投身到有源噪声控制领域,并且逐渐由理论研究转向有源噪声控制的实际应用。

学术界的热情推动了有源噪声控制领域的不断扩展,但是也发现了一些影响着有源噪声控制系统的问题,比如处理的噪声会存在非平稳的特性,它的幅度、频率和相位会出现不规则的变化;另外,构成系统的电路和控制器的电子特性也会发生变化,播放反相声波的扬声器的参数也会发生改变,同时声场的环境同时也是时变的。这些因素都会

影响有源噪声控制的效果。因此人们迫切的需要一种方法，能够自动的追踪噪声的统计特性和控制器的电子特性。

1980 年代，Ross 提出了一种具有追踪特性的有源噪声控制系统，并且首次提出了“自适应”方式的降噪概念<sup>[10, 11]</sup>。该系统基于数字滤波器设计，滤波器的系数会根据噪声的输入以及环境的改变而自适应的修改自身参数，以达到追踪的目的。虽然 Ross 提出的“自适应”概念与目前应用的自适应技术有根本的区别，但是仍然把有源噪声控制理论向前推进了一步。1981 年，Burgess 首次将自适应滤波器引入到有源噪声控制系统中，并且提出了著名的滤波-x 最小均方(Filtered-x Least Mean Square, FxLMS)算法<sup>[12]</sup>，将有源噪声控制系统进一步发展为自适应有源噪声控制系统，提升了系统的降噪性能和稳定性。直到今天，FxLMS 算法及其改进算法仍然凭借着较低的计算量和较好的性能，广泛地应用在有源噪声控制领域。

自适应滤波器以及相应的更新算法是自适应有源噪声控制理论的核心。随着 Burgess 将自适应滤波理论引入到有源噪声控制中并且提出了 FxLMS 算法以后，人们开始注意到自适应算法对有源噪声控制的重要性。

1985 年，Widrow 和 Stearns 在 FxLMS 算法的基础之上，提出了一种次级通路在线建模的方式<sup>[13]</sup>。该方法的建模信号采用主控制器生成的反相控制信号，在系统运行的初始阶段对次级路径进行建模，将建模后的次级路径复制到系统中进行噪声控制。

1989 年，Eriksson 提出了附加噪声的次级通路在线建模算法<sup>[14]</sup>。相较于 Widrow 等直接用控制信号当作次级路径的建模信号，Eriksson 所提算法增加了一个随机噪声发生器，该发生器发出与目标噪声不相关的随机白噪声，并与主控制器产生的控制信号一同作为反相声波。该算法利用白噪声作为次级路径的建模信号，提升了次级路径的建模精准度，进一步提升了 FxLMS 算法的性能，并为 ANC 系统的次级路径在线建模提供了基本思路。其后几十年间，学者们提出了该方法的一些改进算法<sup>[15-19]</sup>，目的是为了改善次级路径的建模精准度，提升系统的降噪性能。

进入 21 世纪，有源噪声控制的研究方向更趋近于解决现实中的实际问题，越来越多的有源降噪产品面向大众消费者。2000 年，Bose 公司推出了第一款民用降噪耳机，掀起了 ANC 技术的民用浪潮，之后 Sony、Audio-Technica 等公司相继推出了相关产品。随着 ANC 技术在实际中的应用，学者们还发现许多问题亟待解决，比如元器件的布局会影响降噪性能，扬声器和麦克风的摆放位置以及个数也会影响系统的降噪性能<sup>[20]</sup>。

相比于国外的不断更新发展，有源噪声控制在国内的研究起步比较晚。但是随着电子技术的迭代更新以及国内科研人员的不懈努力，我国目前的有源噪声控制技术也已经取得重要进展。南京大学的沙家正是国内最早开始研究有源噪声控制的学者之一，在有



源噪声控制理论研究和次级声源改进问题上取得了显著的成果<sup>[21, 22]</sup>。1990 年代, 中科院声学所的马大猷教授在理论和实验两个方面证明了室内有源噪声控制的可行性<sup>[23]</sup>, 为推动我国有源噪声控制技术的发展起到了重要促进作用。西北工业大学的陈克安团队一直深耕于有源噪声控制技术, 涵盖了有源噪声控制理论、实验以及实际应用的系统实现。1993 年, 陈克安和马远良共同撰写了中国第一本有源噪声控制理论专著<sup>[24]</sup>, 为国内有源噪声控制领域的研究打下了理论基础, 并且被作为有源噪声控制理论的经典教材。此后, 国内有关有源噪声控制的研究开始进入快速发展的阶段, 南京大学声学所、哈尔滨工业大学、海军工程大学等学者都在有源噪声控制的不同应用场景取得了较大成果, 并且推动了国内有源噪声控制技术的发展。

国内大部分有源噪声控制研究都建立在数字信号处理芯片和平台之上, 但是实际的相关工程应用相较于国外还相对较少。在智能终端中进行有源降噪是 ANC 系统成功的应用之一, 并且 FxLMS 算法凭借出色的性能和稳定的结构, 成为了 ANC 系统中应用最广泛的自适应算法。

### 1.3 本文研究内容和章节安排

本文主要研究有源噪声控制算法, 并基于目前有源噪声控制系统中存在的时延问题和次级路径建模精度问题, 提出了相应的解决算法。首先应用非线性卡尔曼滤波理论, 提出了基于噪声预测模型的有源噪声控制系统。该系统在基本的 ANC 系统上, 增加了噪声预测模块和延迟估计模块, 即利用预测的方式预测未来时刻语音信号, 通过计算相关性估计出相对准确的时延, 并补偿到系统中, 实现该系统对人声衰减效果的明显提升。其次, 将次级路径建模滤波器用 GRU 网络替换, 提出了基于 GRU 网络的次级路径在线建模算法, 相对于 RNN 模型而言, 其结构简单, 更容易实现, 并且具备良好的映射能力。经过实验验证, 在多种实验条件下, 所提算法均表现出了较高的建模精度, 使得整个系统的降噪性能有较大提高。

本文章节安排如下:

第二章主要介绍了有源噪声控制的基础知识。首先介绍了有源噪声控制系统的典型结构, 然后介绍了有源噪声控制系统中核心的自适应控制算法, 最后介绍了后面章节中需要用到的基础知识。

第三章主要介绍了基于噪声预测有源噪声控制系统, 并对所提出的系统进行了实验验证。

第四章介绍本文提出的基于 GRU 网络的次级路径在线建模算法, 并利用实验验证了算法的良好性能。

## 2 有源噪声控制理论基础

本章着重介绍有源噪声控制理论的核心内容，首先介绍了自适应滤波器和相应的自适应算法，推导了简便有效的最小均方(Least Mean Square, LMS)算法，分析了该算法的关键性能，在此基础上进一步推导出 FxLMS 算法。接着，分析了有源噪声控制系统中次级路径的特性，并且介绍了次级路径建模的两种常用方式：离线建模和在线建模。最后，介绍了时间序列的预测算法和 GRU 网络原理结构，为之后的算法研究分析及改进提供理论基础。

### 2.1 自适应控制算法

有源噪声控制的基础就是自适应滤波器。一个基本的自适应滤波器包含两个部分：数字滤波器  $W(z)$  和自适应算法。其中  $W(z)$  实现相应的信号处理，结构上分为有限长(Finite Impulse Response, FIR)和无限长(Infinite Impulse Response, IIR)两种。FIR 结构稳定性强，且容易实现线性相位；IIR 结构难以控制极点的位置，容易引起不稳定的问题，所以在自适应滤波器中常选用 FIR 结构<sup>[28]</sup>。自适应算法是自适应滤波器的核心理论，其中最具有代表性的算法有最小均方(LMS)算法、递归最小二乘(Recursive Least Square, RLS)算法。LMS 算法因其结构简单、计算复杂度低和在平稳信号中具有良好的收敛性，成为应用最广泛的自适应算法。本节主要介绍基于 LMS 的自适应算法及其性能分析。

#### 2.1.1 LMS 算法

在自适应系统中，系统的输入信号  $x(n)$  的先验信息未知，无法利用维纳方法直接设计出最优的滤波器。利用最速下降法，可以用递推的方式逼近维纳解，这避免了求逆运算，但还是需要信号的相关性信息。

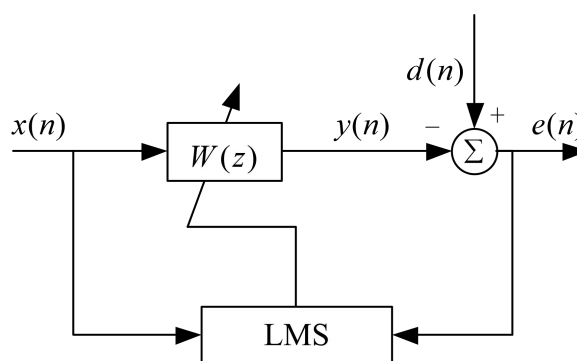


图 2.1 LMS 算法框图

Fig. 2.1 Block diagram of LMS algorithm

1960 年, Widrow 和 Hoff 在最速下降法的基础之上, 提出了 LMS 算法, 即利用瞬时误差代替均方误差的思想<sup>[29]</sup>, 改进了最速下降中求梯度的过程。图 2.1 为 LMS 算法的系统框图, 其中  $x(n)$  为系统输入信号,  $d(n)$  是系统所期望的信号,  $y(n)$  是自适应滤波器的输出信号,  $e(n)$  是期望信号和滤波器输出信号之间的误差。表示为:

$$e(n) = d(n) - y(n) \quad (2.1)$$

系统所期望的是找到最优的滤波器  $W(z)$ 。让滤波器输出信号  $y(n)$  尽可能地接近  $d(n)$ , 也就是使得误差信号  $e(n)$  尽可能的小。在最小均方准则下, 代价函数表示为:

$$\varepsilon = E\{e^2(n)\} = E\{[d(n) - y(n)]^2\} \quad (2.2)$$

式中  $y(n) = x(n) * w(n)$ 。将  $y(n)$  的表达式展开后, 代入到方程(2.2)中, 可得:

$$\begin{aligned} \varepsilon &= E\{d^2(n)\} - 2E\{d(n)y(n)\} + E\{y^2(n)\} \\ &= E\{d^2(n)\} - 2\sum_{k=0}^{\infty} w(k)E\{d(n)x(n-k)\} + \sum_{k=0}^{\infty} \sum_{m=0}^{\infty} w(k)w(m)E\{x(n-m)x(n-k)\} \end{aligned} \quad (2.3)$$

令  $r_d(0) = E\{d^2(n)\}$ ,  $r_{dx}(k) = E\{d(n)x(n-k)\}$ ,  $E\{x(n-m)x(n-k)\} = R_x(k-m)$  为信号  $x(n)$  的自相关。将最小均方准则的代价函数重写后:

$$\varepsilon = r_d(0) - 2\sum_{k=0}^{\infty} w(k)r_{dx}(k) + \sum_{k=0}^{\infty} \sum_{m=0}^{\infty} w(k)w(m)r_x(k-m) \quad (2.4)$$

令上式对  $w(k)$  求偏导并令其为得零, 即可得到在最下均方准则下的最优滤波器  $w_{opt}$  表达式:

$$\frac{\partial \varepsilon}{\partial w(k)} = -2r_{dx}(k) + 2\sum_{m=0}^{\infty} w(m)r_x(k-m) = 0, \quad k = 0, 1, \dots, \infty \quad (2.5)$$

$$\sum_{m=0}^{\infty} w_{opt}(m)r_x(k-m) = r_{dx}(k), \quad k = 0, 1, \dots, \infty \quad (2.6)$$

写成矩阵形式为:

$$w_{opt} = R_x^{-1}r_{dx} \quad (2.7)$$

其中  $w_{opt} = [w_{opt}(0), w_{opt}(1), \dots, w_{opt}(M-1)]^T$ ,  $r_{dx} = [r_{dx}(0), r_{dx}(1), \dots, r_{dx}(M-1)]^T$  为互相关矢量,  $R_x$  是一个  $M \times M$  大小的 Toeplitz 自相关矩阵。从最优解的表达式可以看见, 在(2.7)的维纳解中, 包含了矩阵的求逆运算, 这在实际使用时限制较多。

最速下降法可以用迭代的方式逼近最优维纳解。第  $n+1$  次迭代的滤波器系数更新方程如下:

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \frac{1}{2} \boldsymbol{\varphi}(n) \mu(n) \quad (2.8)$$

其中,  $\boldsymbol{\varphi}(n) = \frac{\partial \varepsilon}{\partial \mathbf{w}(n)} = 2 \mathbf{R}_x \mathbf{w}(n) - 2 \mathbf{r}_{dx}$  为迭代时候的梯度矢量,  $\mu(n)$  为  $n$  时刻的步长。这里引入步长因子, 可以有效地控制系统的迭代速率。当搜索到最优解时, 梯度向量为 0。最优维纳解为:

$$\mathbf{w}(n) = \mathbf{R}_x^{-1} \mathbf{r}_{dx} = \mathbf{w}_{opt} \quad (2.9)$$

最速下降法虽然避免了维纳解的矩阵求逆运算, 但是梯度向量  $\boldsymbol{\varphi}(n)$  的计算仍然需要信号的先验信息  $\mathbf{R}_x$  和  $\mathbf{r}_{dx}$  才能求得。当处理非平稳信号时, 需要在每一时刻计算信号的  $\mathbf{R}_x$  和  $\mathbf{r}_{dx}$ , 这在实际的使用中较难实现。

LMS 算法在最速下降的基础之上, 在求梯度  $\boldsymbol{\varphi}(n)$  时, 利用瞬时误差能量  $e^2(n)$  替换掉均方误差能量  $E\{e^2(n)\}$ , 即  $\varepsilon = e^2(n)$ 。将  $e(n)$  展开, 写成矩阵形式:

$$e(n) = d(n) - \sum_{k=0}^{M-1} w(k)x(n-k) = d(n) - \mathbf{w}^T(n) \mathbf{X}(n) \quad (2.10)$$

其中  $\mathbf{X}(n) = [x(n), x(n-1), \dots, x(n-M+1)]^T$  为系统的输入信号向量。将式(2.10)代入  $\boldsymbol{\varphi}(n)$  的求导公式中, 可得梯度估计值  $\hat{\boldsymbol{\varphi}}(n)$  为:

$$\hat{\boldsymbol{\varphi}}(n) = \frac{\partial \varepsilon}{\partial \mathbf{w}(n)} = -2e(n) \mathbf{X}(n) \quad (2.11)$$

再将估计出的误差代入到(2.8)中, 可得到 LMS 算法的更新迭代公式:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e(n) \mathbf{X}(n) \quad (2.12)$$

式(2.12)即 LMS 算法的迭代公式。显然, LMS 算法不需要矩阵的求逆运算, 还不需要信号的先验信息, 对于非平稳信号也具有较好的性能, 因此成为应用最广泛的自适应算法。

### 2.1.2 LMS 算法性能分析

LMS 算法的思想是在求梯度  $\boldsymbol{\varphi}(n)$  时, 利用瞬时能量代替平均能量, 利用梯度的估计值  $\hat{\boldsymbol{\varphi}}(n)$  代替真实值  $\boldsymbol{\varphi}(n)$ 。当信号是非平稳信号的时候, 信号的瞬时信息变化较大, 这时利用瞬时信息代替集总信息时必然会产生误差, 从而影响  $\mathbf{w}(n)$  收敛到  $\mathbf{w}_{opt}$  的过程, 所以 LMS 算法收敛稳定性分析就至关重要。

LMS 算法每次迭代都是在使  $\mathbf{w}(n)$  向着  $\mathbf{w}_{opt}$  逼近的过程，将每次迭代时所获得的滤波器  $\mathbf{w}(n)$  和最优解  $\mathbf{w}_{opt}$  之间的差值记为  $\mathbf{w}_e$ ，即： $\mathbf{w}_e(n) = \mathbf{w}(n) - \mathbf{w}_{opt}$ 。将(2.8)中的梯度  $\phi(n)$  展开，可得 LMS 算法更新公式：

$$\mathbf{w}(n+1) = [\mathbf{I} - \mu(n)\mathbf{R}_x]\mathbf{w}(n) + \mu(n)\mathbf{r}_{dx} \quad (2.13)$$

其中  $\mathbf{I}$  表示单位矩阵，将上式两侧同时减去  $\mathbf{w}_{opt}$ ：

$$\mathbf{w}(n+1) - \mathbf{w}_{opt} = [\mathbf{I} - \mu(n)\mathbf{R}_x]\mathbf{w}(n) + \mu(n)\mathbf{r}_{dx} - \mathbf{w}_{opt} \quad (2.14)$$

根据  $\mathbf{w}_e$  的定义，可得等式左侧为  $\mathbf{w}_e(n+1)$ ，并且已知  $\mathbf{R}_x\mathbf{w}_{opt} = \mathbf{r}_{dx}$  代入等式右侧化简后，可得：

$$\mathbf{w}_e(n+1) = [\mathbf{I} - \mu(n)\mathbf{R}_x]\mathbf{w}_e(n) \quad (2.15)$$

由于  $\mathbf{R}_x$  是对称正定的二次型矩阵，所以对  $\mathbf{R}_x$  做矩阵正交分解：

$$\mathbf{R}_x = \mathbf{Q}\mathbf{A}\mathbf{Q}^T \quad (2.16)$$

其中  $\mathbf{Q}$  是  $\mathbf{R}_x$  的正交矩阵， $\mathbf{Q} = \mathbf{Q}^T, \mathbf{Q}^T = \mathbf{Q}^{-1}$ ， $\mathbf{A} = \text{diag}[\lambda_0, \lambda_1, \dots, \lambda_{M-1}]$  是  $\mathbf{R}_x$  特征值组成的对角矩阵。式(2.15)表示为：

$$\mathbf{w}_e(n+1) = [\mathbf{I} - \mu(n)\mathbf{Q}\mathbf{A}\mathbf{Q}^T]\mathbf{w}_e(n) \quad (2.17)$$

两边同时乘  $\mathbf{Q}^T$  得：

$$\mathbf{Q}^T\mathbf{w}_e(n+1) = \mathbf{Q}^T\mathbf{w}_e(n) - \mu(n)\mathbf{Q}^T\mathbf{Q}\mathbf{A}\mathbf{Q}^T\mathbf{w}_e(n) \quad (2.18)$$

并且定义  $\mathbf{v}(n) = \mathbf{Q}^T\mathbf{w}_e(n)$ ，上式可化为：

$$\mathbf{v}(n+1) = \mathbf{v}(n) - \mu(n)\mathbf{A}\mathbf{v}(n) = [\mathbf{I} - \mu\mathbf{A}]\mathbf{v}(n) \quad (2.19)$$

在给定  $\mathbf{v}(n)$  的初始值  $\mathbf{v}(0)$  后，经过递推可以得到：

$$\mathbf{v}(n) = [\mathbf{I} - \mu\mathbf{A}]^n\mathbf{v}(0) \quad (2.20)$$

在  $\mathbf{w}(n)$  迭代时，根据  $\mathbf{v}(n)$  的定义其应该趋近于 0。对于  $\mathbf{v}(n)$  中的第  $k$  个元素  $v_k(n)$ ，同样也趋近于 0：

$$v_k(n) = [1 - \mu\lambda_k]^n v_k(0), \quad k = 0, 1, 2, \dots, M-1 \quad (2.21)$$

其中  $\lambda_k$  为第  $k$  个特征值。从式(2.21)可以得出， $v_k(n)$  趋近于 0 的必要条件是  $|1 - \mu\lambda_k| < 1$ ，从而可以得到最速下降法稳定时，步长因子  $\mu$  需满足如下条件<sup>[28]</sup>：

$$0 < \mu < \frac{2}{\lambda_{\max}} \quad (2.22)$$

其中  $\lambda_{\max}$  表示自相关矩阵的最大特征值。当步长因子满足上式条件时， $\mathbf{w}(n)$  会随着迭代次数的增加而趋近于  $\mathbf{w}_{opt}$ ，而  $v_k(n)$  也无限接近于 0。自相关矩阵最大特征值  $\lambda_{\max}$  满足：

$$\lambda_{\max} \leq \sum_{k=0}^{M-1} \lambda_k = \text{trace}\{\mathbf{R}_x\} = M \cdot r_x(0) \quad (2.23)$$

其中  $\text{trace}\{\mathbf{R}_x\}$  表示矩阵  $\mathbf{R}_x$  的迹,  $r_x(0)$  表示信号  $\mathbf{x}(n)$  的功率,  $M$  为矩阵阶数。从式(2.23)可见, 在不要求特征值的情况下, 仅凭借信号  $\mathbf{x}(n)$  即可确定迭代时的步长范围, 从而得到系统稳定时的收敛条件。

在  $\mathbf{w}(n)$  通过迭代收敛到最优解  $\mathbf{w}_{opt}$  的过程中, 每一次迭代都会因为梯度估计值的原因产生一定的误差。当  $\mathbf{w}(n)$  逼近到  $\mathbf{w}_{opt}$  附近时 LMS 算法收敛, 此时梯度真实值为应该为零, 但是估计值不是零。这就会导致  $\mathbf{w}(n+1)$  偏离  $\mathbf{w}_{opt}$ , 不能停留在最优解, 算法的代价函数均方误差  $\varepsilon$  也会偏离最小值  $\varepsilon_{\min}$ 。此时产生的均方误差被称为均方误差的超量。

从 LMS 算法的更新公式可见, 步长  $\mu$  在 LMS 算法的性能上起到了关键性作用。当步长选取的较大时, 算法拥有快速的收敛速度,  $\mathbf{w}(n)$  只需要很少的迭代次数就能到逼近到  $\mathbf{w}_{opt}$ , 但是同时较大的步长在算法收敛后也会导致较大的误差超量。当选取较小的步长时, 虽然在收敛后的误差超量较小, 偏离最优解  $\mathbf{w}_{opt}$  得程度较小, 但是会导致较慢的收敛速度, 尤其是在信号非平稳的情况下, 过慢的收敛速度会使算法无法跟上信号的变化速度, 导致算法始终无法收敛。

在选取步长的时候, 必须要在收敛速度和均方误差超量两个方面折中。对于平稳信号或者不强调收敛速度的应用中, 可以选取较小的步长, 以保证在稳定时具有较小的均方误差超量; 在强调收敛速度的应用中, 可以选取较大的步长, 以保证收敛速度符合要求, 在稳定后调整到小步长, 防止过大的均方误差超量。

### 2.1.3 FxLMS 算法

LMS 算法的模型过于简单理想化, 实际的有源噪声控制系统, 自适应控制器计算出反相信号后, 需要经过扬声器播放出来后, 经过空气路径后和噪声信号干涉相消, 两者叠加之后形成误差信号, 并不是 LMS 理想模型中直接利用信号相加减。在自适应有源噪声控制系统中, 需要额外考虑以下问题: (1)有源噪声控制系统处理的并不是简单的电信号, 还有声信号, 需要考虑电路中的声学器件的干扰; (2)反相的控制信号从扬声器播放出来后, 在次级路径中和噪声相互抵消后才能被误差麦克风拾取, 因此在分析时要考虑到次级路径的影响。

1981 年, Burgess 在结合管道有源噪声控制需求时, 提出了 FxLMS 算法<sup>[12]</sup>。单通道前馈式 FxLMS 算法框图如图 2.2 所示。

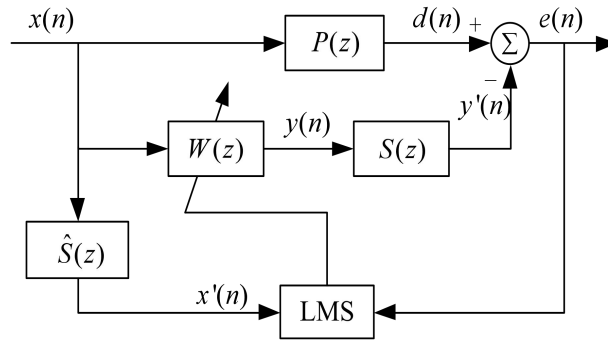


图 2.2 FxLMS 算法框图

Fig. 2.2 Block diagram of FxLMS algorithm

图 2.2 中在 LMS 算法的基础之上考虑了次级路径的影响， $P(z)$  表示噪声传播的初级路径， $S(z)$  表示次级声学路径， $y'(n)$  为控制信号经过声学次级路径后的信号表示。因为对反相控制信号考虑了次级路径的影响，自适应算法输入的参考信号同样也要补偿一个次级路径，称为滤波-x 信号，但是声学通路无法真实获得，所以用次级路径的估计值  $\hat{S}(z)$  来代替。在  $n$  时刻，误差麦克风附近的噪声信号表示为：

$$d(n) = \mathbf{x}(n) * \mathbf{p}(n) \quad (2.24)$$

此时滤波器输出的反相控制信号为：

$$y(n) = \mathbf{x}(n) * \mathbf{w}(n) \quad (2.25)$$

经过次级声学路径后的信号为：

$$y'(n) = y(n) * s(n) = \mathbf{x}(n) * \mathbf{w}(n) * s(n) \quad (2.26)$$

误差麦克风处的误差信号为：

$$e(n) = d(n) - y'(n) = d(n) - y(n) * s(n) \quad (2.27)$$

滤波-x 信号  $x'(n)$  为：

$$x'(n) = \mathbf{x}(n) * \hat{s}(n) \quad (2.28)$$

其中  $\hat{s}(n)$  为次级路径的估计值，将上述信号带入到 LMS 算法的更新公式可得：

$$\mathbf{w}(n+1) = \mathbf{w}(n) + 2\mu e(n) \mathbf{X}'(n) \quad (2.29)$$

上式(2.29)即为 FxLMS 算法的迭代公式，其中  $\mathbf{X}'(n) = [x'(n), x'(n-1), \dots, x'(n-M+1)]^T$  表示滤波-x 信号矢量。可见，FxLMS 算法核心还是基于 LMS 算法的思路，重点在于考虑了次级路径对控制信号的影响，并且对输入信号也补偿了次级路径传递函数的估计值。

#### 2.1.4 次级路径辨识算法

FxLMS 算法中考虑了次级路径对控制噪声的影响，并且对参与滤波器更新的参考信号补偿了次级路径的估计值  $\hat{S}(z)$ 。因此次级路径估计的准确性就成为了 FxLMS 算法的关键因素。根据建模值在有源噪声控制系统中是否改变，次级路径辨识算法分为离线建模(Offline)和在线建模(Online)两种<sup>[30]</sup>。

离线建模的方式主要可以分为有时延估计法和附加随机噪声法。时延估计法就是将次级路径化简成为一个简单的时延系统，不考虑对信号幅值的影响。使用时延估计法，可以直接利用人工测量出次级路径的时延估计值，操作简单，但是过于依赖时延测量的准确性。并且将次级路径简化成一个简单的时延系统在很多应用场景下并不合适。

附加随机噪声法就是利用 LMS 算法计算出次级路径的模型参数，在次级路径中播放随机噪声的同时通入建模滤波器，利用误差麦克风拾取到的信号作为期望信号。附加随机噪声法的框图如下：

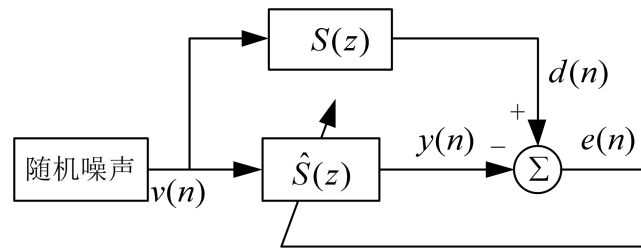


图 2.3 附加随机噪声法框图

Fig. 2.3 Block diagram of additional random noise method

为了保证在所有频带上都能得到准确的频率响应，图 2.3 中随机噪声信号用  $v(n)$  常用白噪声。误差信号表示为：

$$e(n) = d(n) - y(n) = v(n) * s(n) - v(n) * \hat{s}(n) \quad (2.30)$$

利用 LMS 算法得到：

$$\hat{s}(n+1) = \hat{s}(n) + \mu e(n)v(n) \quad (2.31)$$

当  $e(n) = 0$  时，系统收敛到稳定状态，此时有：

$$s(n) = \hat{s}(n) \quad (2.32)$$

此时，次级路径完成建模过程。完成离线建模后，估计出的次级路径  $\hat{S}(z)$  固定代入到 FxLMS 系统中，这样就只适用于一种次级路径的系统，如果次级路径发生变化，就会导致路径误差较大，甚至导致 ANC 系统发散，所以针对时变的次级路径情况，就需要使用在线建模的方式。



次级路径的在线建模就是在 FxLMS 算法收敛的过程中，同时附加随机噪声利用 LMS 算法估计次级路径。次级路径在线建模的框图如图 2.4 所示。

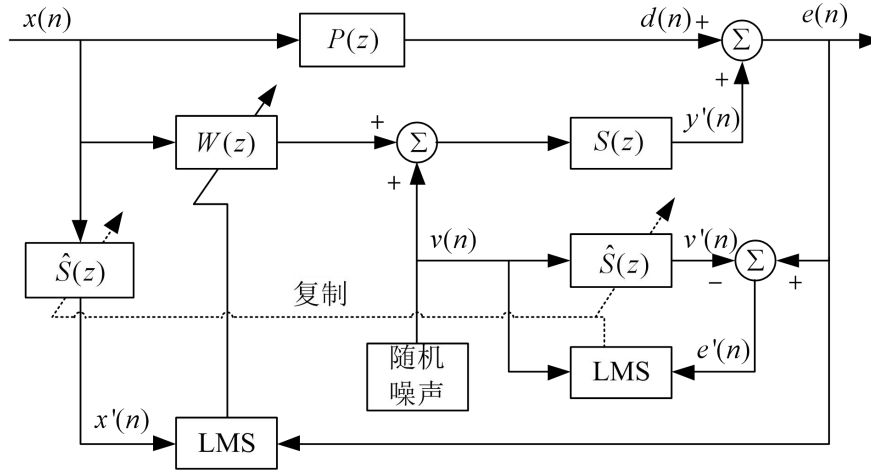


图 2.4 次级路径在线建模算法框图

Fig. 2.4 Block diagram of secondary path online modeling algorithm

相比于 FxLMS 算法框架，图 2.4 多了一个随机噪声发生器和一个自适应滤波器。反相控制噪声和随机噪声混合经过次级路径后表示为：

$$y'(n) = [y(n) + v(n)] * s(n) \quad (2.33)$$

其中  $v(n)$  表示和  $x(n)$  不相关的随机噪声，此时误差麦克风处拾取到的残留信号表示为：

$$e(n) = d(n) + y'(n) = d(n) + y(n) * s(n) + v(n) * s(n) \quad (2.34)$$

式(2.34)中的  $e(n)$  可以分为两部分， $d(n) + y(n) * s(n)$  表示初级噪声和控制噪声的残留误差， $v(n) * s(n)$  为辅助噪声经过次级路径的信号表达式。 $v(n)$  经过建模滤波器后表示为：

$$v'(n) = v(n) * \hat{s}(n) \quad (2.35)$$

由于随机噪声和参考输入  $x(n)$  一般不相关，所以  $d(n) - y(n) * s(n)$  和  $v(n) * s(n)$  也不相关。这时可得到  $e'(n)$  的表达式：

$$e'(n) = e(n) - v'(n) = d(n) + y(n) * s(n) + v(n) * s(n) - v(n) * \hat{s}(n) \quad (2.36)$$

代入到 LMS 算法中，就能够得到建模滤波器的更新公式：

$$\hat{s}(n+1) = \hat{s}(n) + \mu v(n) [d(n) + y(n) * s(n) + v(n) * s(n) - v(n) * \hat{s}(n)] \quad (2.37)$$

当系统收敛时，有  $\hat{s}(n) = s(n)$ 。此时，次级路径建模系统的误差信号表示为：

$$e'(n) = d(n) + y(n) * s(n) \quad (2.38)$$

只剩下初级噪声和控制噪声之间的误差，随着次级路径的收敛，主控制滤波器误差同样也收敛到最小值。

次级路径在线建模算法适用于次级路径时变的场景，建模精准度高，但是结构复杂，并需要实时的加入不相关的辅助噪声，这在有些使用环境下并不能满足这样的条件。

## 2.2 时间序列预测方法概述

时间序列预测就是利用过去一段时间内事件的特征来预测未来该事件的特征，本质上是利用历史数据的统计分析来预测未来事件，并且依赖于事件发生的先后顺序。本节介绍一些常见的时间序列预测方法。

### 2.2.1 传统时序预测方法

传统时序预测算法可以分为两种<sup>[31]</sup>，一种是简单的历史平均值算法如：滑动平均法、指数平均法。历史平均值算法是一种静态算法，算法逻辑简单，计算速度快，对变化平稳的事件预测效果较好。但是当事件发生突变的情况下，由于逻辑简单会导致预测精度下降，预测误差增大，这类算法通常应用在要求计算速度快，对精度要求不高的系统中。

另一种则是基于模型且考虑多种因素组合的参数方法，如自回归模型(AR)、滑动平均模型(MA)、自回归滑动平均模型(ARMA)。该类方法结构也比较简单，在历史数据连续且充足的情况下，预测精度较高，但同样在发生突变的时候无法准确的预测，鲁棒性较差。

### 2.2.2 卡尔曼滤波法

卡尔曼滤波(Kalman Filtering)也是时间序列中常用的预测算法，该方法利用线性系统的状态方程和系统的输入、输出数据，以最小均方误差为最佳估计准则，采用信号与噪声的状态空间模型，利用前一时刻的估计值和当前时刻的观测值来更新状态变量的估计值，求出当前时刻的估计值，之后根据建立的系统方程和观测方程对需要处理的信号做出满足最小均方误差的估计<sup>[32]</sup>。

在离散时间系统中，系统的状态方程和观测方程分别定义为：

$$\mathbf{x}_k = \mathbf{A}_{k-1}\mathbf{x}_{k-1} + \mathbf{G}_{k-1}\mathbf{w}_{k-1} \quad (2.39)$$

$$\mathbf{y}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k \quad (2.40)$$

其中  $\mathbf{x}_k$  表示系统的状态变量， $\mathbf{A}_k$ ， $\mathbf{G}_k$  为系统的状态转移矩阵， $\mathbf{y}_k$  为系统的观测信号矢量， $\mathbf{H}_k$  表示信号观测矩阵， $\mathbf{w}_k$  和  $\mathbf{v}_k$  分别表示系统中激励噪声和观测噪声，是相互独立且正态分布的白噪声，即满足：

$$p(\mathbf{w}) \sim N(0, \mathbf{Q}) \quad (2.41)$$

$$p(v) \sim N(0, R) \quad (2.42)$$

实际系统中, 系统激励噪声协方差矩阵  $\mathbf{Q}$  和观测噪声协方差矩阵  $\mathbf{R}$  可能随着每次的迭代计算而变化, 但在这假设他们为常数。

卡尔曼滤波工作时可以分为两个阶段, 第一阶段是状态变量的预测估计, 在这个阶段, 利用  $(k-1)$  时刻的预测估计值和控制输入进行预测。第二阶段则是信号状态变量的校正, 在此阶段, 基于观测值和估计值之间最小均方准则, 确定出滤波增益矩阵, 确定系统中预测值、观测值、增益矩阵和噪声之间的变换关系, 从而实现对第一阶段中预测值的校正。不需要整体的历史数据, 仅需要  $(k-1)$  时刻的系统估计值和观测值, 就能对  $k$  时刻的系统值进行更新。卡尔曼滤波在  $k$  时刻工作过程可以利用以下 5 个公式表示:

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{A}_{k-1} \hat{\mathbf{x}}_{k-1|k-1} \quad (2.43)$$

$$\mathbf{P}_{k|k-1} = \mathbf{A}_k \mathbf{P}_{k-1|k-1} \mathbf{A}_k^T + \mathbf{Q}_k \quad (2.44)$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \quad (2.45)$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k (\mathbf{y}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1}) \quad (2.46)$$

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1} \quad (2.47)$$

式(2.43)和(2.44)表示卡尔曼滤波的第一阶段, (2.45)~(2.47)为滤波的第二阶段。其中,  $\hat{\mathbf{x}}_{k|k-1}$  表示  $k$  时刻下系统的先验估计,  $\hat{\mathbf{x}}_{k|k}$  表示  $k$  时刻系统的后验估计,  $\mathbf{P}_{k|k-1}$  为先验估计的协方差矩阵,  $\mathbf{P}_{k|k}$  为后验估计的协方差矩阵,  $\mathbf{K}_k$  定义为  $k$  时刻的卡尔曼增益,  $\mathbf{Q}_k$  和  $\mathbf{R}_k$  分别是  $\mathbf{w}_k$  和  $\mathbf{v}_k$  的协方差矩阵。可以看到, 在式(2.46)中, 通过系统观测值  $\mathbf{y}_k$  来校正预测状态变量的先验估计, 得到后验估计值  $\hat{\mathbf{x}}_{k|k}$ 。当前时刻的后验估计随即进入下一轮计算, 下一轮的先验估计取值为上一时刻的后验估计, 整体是一个递推的过程, 直到获得最佳预测值。

当考虑以下非线性系统:

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}) + \mathbf{G}_{k-1} \mathbf{w}_{k-1} \quad (2.48)$$

$$\mathbf{y}_k = h(\mathbf{x}_k) + \mathbf{v}_k \quad (2.49)$$

其中  $f(*)$  和  $h(*)$  分别表示状态和观测的非线性方程, 其余变量和线性卡尔曼滤波定义一致。通常, 可以利用贝叶斯方法对非线性系统进行递推估计。

扩展卡尔曼滤波(Extended Kalman Filter, EKF)方法<sup>[33-35]</sup>是应用最广泛的一种贝叶斯近似滤波方法, 它在当前状态估计值上对系统进行线性处理, 以此将非线性系统转换成线性系统处理。扩展卡尔曼滤波方法利用一阶泰勒技术将非线性函数  $f(*)$  和  $h(*)$  展开,

只保留泰勒级数的一阶项，舍弃二阶项及更高次项；然后，利用卡尔曼滤波器的递推结构，更新系统状态均值和方差的估计值。根据泰勒公式，有：

$$f(\mathbf{x}_{k-1}) = f(\mathbf{x}_{k-1|k-1}) + \mathcal{D}_{\delta_X} f + \sum_{j=2}^{\infty} \frac{1}{j!} \mathcal{D}_{\delta_X}^j f \quad (2.50)$$

其中  $\delta_X = \mathbf{x}_{k-1} - \mathbf{x}_{k-1|k-1}$ ， $\mathcal{D}_{\delta_X} f$  是函数在  $\mathbf{x}_{k-1|k-1}$  处的全微分，即：

$$\mathcal{D}_{\delta_X} f = [(\delta_X^T \nabla) f(\mathbf{x})]^T \Big|_{\mathbf{x}=\mathbf{x}_{k-1|k-1}} = \sum_{p=1}^{n_x} \delta_{x_p} \frac{\partial f(\mathbf{x})}{\partial x_p} \Big|_{\mathbf{x}=\mathbf{x}_{k-1|k-1}} \triangleq \tilde{\mathbf{F}}_k \delta_X \quad (2.51)$$

$$\frac{1}{j!} \mathcal{D}_{\delta_X}^j f = \frac{1}{j!} \left[ \sum_{p=1}^{n_x} \delta_{x_p} \frac{\partial}{\partial x_p} \right]^j f(\mathbf{x}) \Big|_{\mathbf{x}=\mathbf{x}_{k-1|k-1}} \quad (2.52)$$

将上式带入系统状态方程中：

$$\mathbf{x}_k = f(\mathbf{x}_{k-1|k-1}) + \tilde{\mathbf{F}}_k \cdot (\mathbf{x}_{k-1} - \mathbf{x}_{k-1|k-1}) + \varepsilon_{k-1} \quad (2.53)$$

其中  $\varepsilon_{k-1} = \sum_{j=2}^{\infty} \frac{1}{j!} \mathcal{D}_{\delta_X}^j f + \mathbf{v}_{k-1}$ 。同理观测方程可写成：

$$\mathbf{z}_k = h(\mathbf{x}_{k|k-1}) + \tilde{\mathbf{H}}_k \cdot (\mathbf{x}_k - \mathbf{x}_{k|k-1}) + \eta_k \quad (2.54)$$

其中  $\tilde{\mathbf{H}}_k \triangleq \mathcal{D}h|_{\mathbf{x}=\mathbf{x}_{k|k-1}}$ ， $\eta_k = \sum_{j=2}^{\infty} \frac{1}{j!} \mathcal{D}_{\delta_X}^j h + \mathbf{n}_k$ 。

通过以上的泰勒级数展开，就可以将非线性系统的方程就转化为线性方程，从而可用卡尔曼滤波进行迭代求解。扩展卡尔曼滤波的过程如下：

#### (1) 预测步骤

$$\hat{\mathbf{x}}_{k|k-1} = \tilde{\mathbf{F}}_{k-1} \hat{\mathbf{x}}_{k-1|k-1} \quad (2.55)$$

$$\mathbf{P}_{k|k-1} = \tilde{\mathbf{F}}_k \mathbf{P}_{k-1|k-1} \tilde{\mathbf{F}}_k^T + \mathbf{Q}_{k-1} \quad (2.56)$$

#### (2) 更新步骤

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \tilde{\mathbf{H}}_k^T (\tilde{\mathbf{H}}_k \mathbf{P}_{k|k-1} \tilde{\mathbf{H}}_k^T + \mathbf{R}_k)^{-1} \quad (2.57)$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k (\mathbf{y}_k - \tilde{\mathbf{H}}_k \hat{\mathbf{x}}_{k|k-1}) \quad (2.58)$$

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \tilde{\mathbf{H}}_k) \mathbf{P}_{k|k-1} \quad (2.59)$$

在扩展卡尔曼滤波方法中，线性化会带来一些问题。首先，为了逼近系统中的非线性函数，对函数的泰勒级数展开做了一阶截断，由于忽略了高阶项，线性化过程会使系统产生较大的误差，甚至可能导致滤波器发散，数值稳定性稍差。其次，需要计算状态转移矩阵  $\tilde{\mathbf{F}}_k$  和观测矩阵  $\tilde{\mathbf{H}}_k$ ，这些雅克比矩阵的计算很复杂，计算量较大。此外，当系统存在模型失配、测量干扰、测量丢失和延迟等复杂突发情况，算法的鲁棒性稍差。

### 2.2.3 人工神经网络法

人工神经网络(Artificial Neural Network, ANN)<sup>[36]</sup>是一种模拟人类神经元学习过程的算法。它通过大量样本的学习训练之后,建立起一种类似人类神经元的网络模型,具有很强的非线性拟合能力,并且抗干扰能力强,可以使用任意精度使算法逼近最优值。神经网络的种类有很多,常用的有反向传播神经网络(Back Propagation, BP)、循环神经网络(Recurrent Neural Network, RNN)、卷积神经网络(Convolutional Neural Network, CNN)等<sup>[37]</sup>。BP神经网络作为基础的神经网络之一,因为其结构相对简单而广泛使用。在使用人工神经网络时,并不关心其内部实现的机制,仅通过样本数据学习训练,不断优化网络中的参数,从而完善网络模型。但是该算法也有缺点,过大的网络结构可能会导致收敛速度较慢,以及可能会逼近局部最小值。

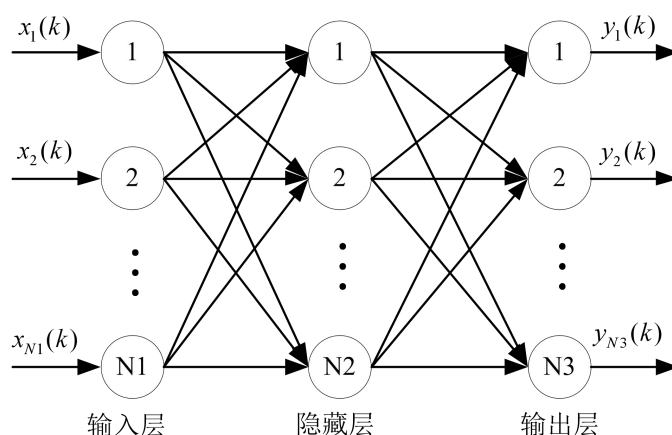


图 2.5 BP 神经网络结构图

Fig. 2.5 BP neural network structure

在图 2.5 中, BP 神经网络通常使用三层网络结构:输入层、隐藏层、输出层,隐藏层可以是一层网络也可以是多层网络结构。信号在网络中的传播是单向从输入到输出单向的,而误差则是从输出到输入反向传播的,因此 BP 神经网络在训练时也分为两个阶段:信号的前向运算和误差的反向传播。在前向运算阶段,运算的过程是单向的,此时第  $n$  层只接受  $n-1$  层的数据,  $n-1$  层并不接收第  $n$  层的反馈,并且同一层的神经元之间互不干扰。在误差反向传播阶段,逆向的逐层修正网络中神经元的阈值和权值,以此减小 BP 神经网络的整体误差,直到误差达到设定的标准或者到达指定的学习次数时结束。

### 2.3 门控循环神经网络

门控循环(Gate Recurrent Unit, GRU)神经网络是反馈神经网络(Recurrent Neural Networks, RNN)的一种, 基于长短时记忆(Long Short-Term Memory, LSTM)网络的简化和升级而得到。它保留了 LSTM 网络的优点, 并降低了网络的结构, 参数量相对较少。当训练数据量大时, GRU 网络能够节省更多的训练时间。本节将依次介绍 RNN、LSTM 和 GRU 的基础知识。

### 2.3.1 循环神经网络

循环神经网络结构如图 2.6 所示, 其中  $X_t$  表示  $t$  时刻的输入向量,  $H_t$  表示输出向量, A 表示神经网络。可以看到循环神经网络和传统神经网络之间的区别在于, 传统的神经网络在结构上是清晰的多层结构, 虽然每一层和每一层之间的节点相互连接, 但是同一层节点之间是没有连接的。但是在 RNN 的结构中, 除了每层节点之间的相互连接外, 同一隐藏层节点之间也存在连接, 即隐藏层节点既接受当前时刻的输入, 又接受上一时间点隐藏层输出结果。

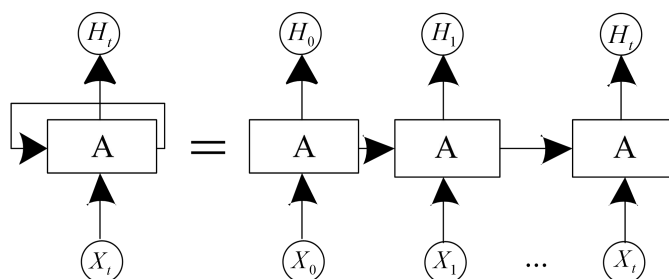


图 2.6 循环神经网络框图

Fig. 2.6 Block diagram of recurrent neural network

在当前输入计算结果输出后, RNN 网络隐藏层的神经单元会将输出的结果保存, 将这些历史信息作为下一时间点的输入, 也就是将记忆网络中所有的历史输出结果, 这些结果将会对后续序列的结果输出产生作用。这种结构特点使 RNN 能够更好的处理序列数据, 即当前时间数据和历史数据存在某种依赖关系时, 普通的神经网络无法处理这种时间上的相关性, 而 RNN 则可以很好的适应这种时间上具有相关性的场景。

虽然 RNN 在理论上可以很好的处理时间序列问题, 但在实际中, RNN 会出现无法学习“长期依赖关系”的问题<sup>[39]</sup>, 对于已经发生的长时间信息无法恰当的处理, 同时还出现了梯度爆炸。当使用 sigmoid 函数作为激活函数时, 该函数的导数小于 1, 若出现多个导数小于 1 的数连续相乘, 在一定程度上就容易出现梯度消失的现象。

### 2.3.2 长短时记忆网络

为解决 RNN 在长序列训练过程中存在的梯度消失和梯度爆炸问题，Hochreiter 和 Schmidhuber<sup>[40]</sup>提出了 LSTM 网络。LSTM 单元结构图如图 2.7 所示。

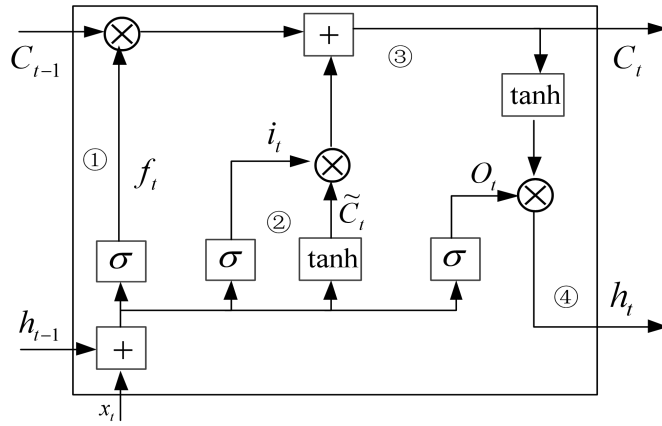


图 2.7 LSTM 单元结构图

Fig. 2.7 Structure of LSTM unit

不同于 RNN 的结构，LSTM 网络有四个隐藏单元，以特殊的方式进行交互。LSTM 网络有两种传递状态，分别为细胞状态(Cell State)和隐藏状态(Hidden State)。一般情况下细胞状态  $C$  在网络中改变较为缓慢，不会因为时间的变化而快速的做出应答。隐藏状态  $h$  对于时间变化十分敏感，不同时间段的隐藏状态会存在较大的差异。在此之上，LSTM 还引入“门”(Gate)的机制来控制细胞状态，经典 LSTM 单元中存在三个门：数入门、输出门和遗忘门，通过三个门对于原先的输入、输出和丢弃的信息进行相关的控制，对信息进行取舍，选择性的记忆或者忘记相关内容，并且在较长的序列进行处理时，性能比一般的 RNN 更加出色；对于时序问题的处理方式也很好的降低了梯度消失和梯度爆炸等不良现象出现的概率。

LSTM 网络单元的工作流程如下：

(1) 首先需要确定从单元中丢弃哪些信息，该功能通过遗忘门(Forget Gate)来实现，如图 2.7 中①所示，此时读取当前输入信号  $x_t$  和神经元前一时刻信息  $h_{t-1}$ ，将两者串联起来作为当前单元的输入矢量。单元输入通过一个 sigmoid 层后得到  $f_t$ ，是一个 0 到 1 范围之间的数值，用来描述每个部分有多少分量可以通过，其中 0 代表“完全丢弃”，1 代表“完全保留”。 $f_t$  的数值决定了上一时刻的细胞状态有多少信息需要丢弃。 $f_t$  的表达式为：

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (2.60)$$

其中  $\sigma(\cdot)$  表示 sigmoid 函数， $W_f$  为输入层与隐层间的权重矩阵， $b_f$  为偏置矢量。

(2)第二步需要确定单元中存放的新信息。如图 2.7 中②所示,这一步由两部分组成。首先,输入矢量经过一层  $\tanh$  函数,得到候选细胞状态  $\tilde{C}_t$ ; 然后,输入矢量通过一个由  $\text{sigmoid}$  函数构成的输入门(Input Gate)生成矢量  $i_t$ ,  $i_t$  决定了  $\tilde{C}_t$  中哪些数据加入会被加入到细胞状态  $C_{t-1}$  中。 $i_t$  和  $\tilde{C}_t$  的计算公式为:

$$i_t = \tanh(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2.61)$$

$$\tilde{C}_t = \sigma(W_c \cdot [h_{t-1}, x_t] + b_i) \quad (2.62)$$

(3) 这一步需要更新旧细胞的状态,即将  $C_{t-1}$  更新为  $C_t$ 。如图 2.7 中③所示,将上一时刻状态  $C_{t-1}$  与  $f_t$  相乘可以确定需要丢弃的信息,  $i_t$  与  $\tilde{C}_t$  相乘可以得到当前时刻的新信息,两部分之和为新的细胞状态:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (2.63)$$

(4) 最后一步需要确定当前时刻细胞输出  $h_t$ 。如图 2.7 中④所示,首先将输入矢量通过一个  $\text{sigmoid}$  层来确定哪些分量参与输出,然后将细胞状态  $C_t$  通过输出门(Output Gate)。输出门的  $\tanh$  函数将  $C_t$  映射到-1 到 1 之间,将其和  $\text{sigmoid}$  门的输出  $o_t$  相乘,得到细胞输出  $h_t$ , 计算过程如下:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (2.64)$$

$$h_t = o_t * \tanh(C_t) \quad (2.65)$$

### 2.3.3 门控循环神经网络

门控循环神经网络(Gated Recurrent Unit, GRU)<sup>[41, 42]</sup>结构如图 2.8 所示。该网络是 LSTM 网络的一种轻量级变体, GRU 单元在 LSTM 的基础上做了简化和升级,结构上只保留了两个门:将遗忘门和输入门合成一个单一的更新门,更新门控制前一时刻的信息被代入到当前时刻状态的程度,对过去的一定信息进行保留,同时也约束了输入层应该接收多少输入矢量的信息;另一个门为重置门,用于控制忽略前一时刻的状态信息的程度,重置门的值越小说明忽略的值越多。GRU 单元中没有输出门,也就是说单元的输出永远是完整的状态信息。另外,GRU 网络结构较 LSTM 网络更为简单,连接较少,网络的运算过程中参数量更少,计算量得到了优化,由此本身的速度也得到了进一步的提升。



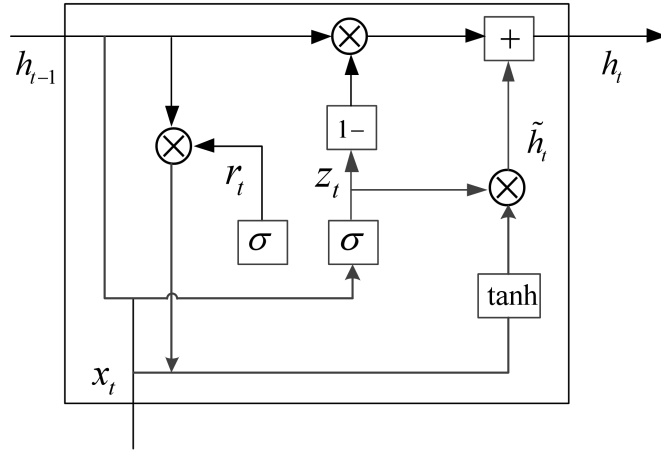


图 2.8 GRU 单元结构图

Fig. 2.8 Structure of GRU unit

GRU 单元的运算过程如下：

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t]) \quad (2.66)$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t]) \quad (2.67)$$

$$\tilde{h}_t = \tanh(W_h \cdot [r_t * h_{t-1}, x_t]) \quad (2.68)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t \quad (2.69)$$

## 2.4 本章小结

本章主要介绍了有源噪声控制的基础理论知识，包括 LMS 算法、FxLMS 算法、次级路径辨识算法；阐述了几种常见的时间序列预测算法以及多种循环神经网络的基本原理框架。其中 LMS 算法和 FxLMS 算法是有源噪声控制理论的基础；时间序列预测算法是第三章基于噪声预测的有源噪声控制系统的理论基础；次级路径辨识算法和 GRU 网络原理框架则是第四章的应用基础。

### 3 基于噪声预测的有源噪声控制系统

#### 3.1 有源噪声控制信号模型

本节主要对有源噪声控制系统的信号模型进行分析。在只考虑单个声源的情况下，前馈式有源噪声控制系统的框图简化如图 3.1 所示：

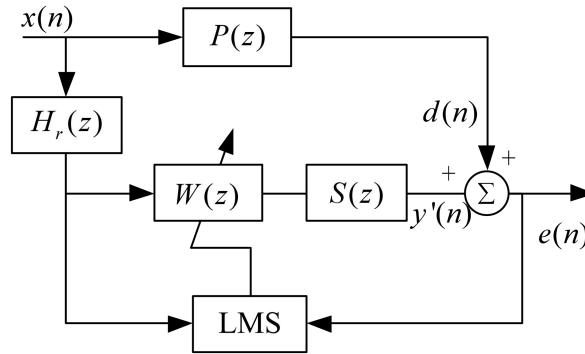


图 3.1 有源噪声控制系统简化框图

Fig. 3.1 Simplified block diagram of active noise cancellation system

噪声源发出噪声  $x(n)$  经过初级路径到达误差麦克风处，同时经过  $H_r(z)$  系统后被参考麦克风拾取，送入 ANC 系统进行处理，然后经由次级扬声器播放出来，经过次级路径传播后到达误差麦克风处。此时，系统的误差麦克风接收到两个信号，一个是噪声源经过初级路径传播直接到达误差麦克风  $d(n)$ ，另一个是次级扬声器播放出的反向控制噪声  $y'(n)$ ，它们分别表示为：

$$d(n) = x(n) * p(n) \quad (3.1)$$

$$y'(n) = s(n) * [w(n) * [x(n) * h_r(n)]] \quad (3.2)$$

其中  $h_r(n)$  表示噪声在被参考麦克风拾取之前经过的传播路径。此时，误差麦克风接收到的信号  $e(n)$  为：

$$e(n) = d(n) + y'(n) = x(n) * p(n) + s(n) * [w(n) * [x(n) * h_r(n)]] \quad (3.3)$$

$e(n)$  反馈给自适应滤波器，作为下一时刻滤波器更新的信息。理想情况下， $e(n) = 0$ 。此时的  $w(n)$  为：

$$w(n) = -s^{-1}(n) * p(n) * h_r^{-1}(n) \quad (3.4)$$

要想达到最优的降噪效果，必须正确估计出上式中三个系统函数。其中  $s^{-1}(n)$  表示次级路径的逆系统，可以通过在次级扬声器播放已知的信号，并测量误差麦克风采集到信号响应的方式准确估计。然而， $p(n)$  和  $h_r^{-1}(n)$  却无法获得，因为初级路径在系统工作的过程中是时变的。而对  $h_r(n)$  系统求逆是一个非因果操作，会导致最优滤波器  $w(n)$  估计不准确。

为了实现良好的降噪性能，超低延迟处理是 ANC 控制系统的一个必要条件。在 ANC 系统的理论分析中，都是直接利用  $x(n)$  作为参考信号，通过  $x(n) * p(n)$  得到的  $d(n)$  作为待衰减的噪声信号，并没有考虑实际应用的时候噪声经过实际传播路径后造成的时延，并且 ANC 中的滤波处理、数模转换以及在模拟器件中的传输也会带来额外的延迟，这也会成为影响系统性能的关键因素。

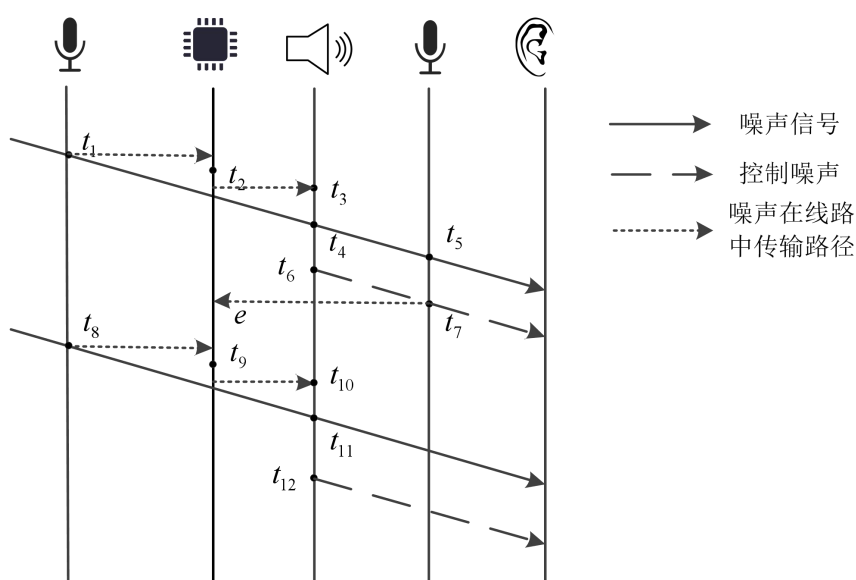


图 3.2 有源降噪系统的工作时间线

Fig. 3.2 Working timeline of active noise canceling systems

某有源降噪系统的工作时间线如图 3.2 所示。图 3.2 中每条竖线代表了降噪系统中的一个组件，噪声信号  $x(n)$  经过参考麦克风、数字信号处理器、次级扬声器和误差麦克风，最后到达抵消点处。噪声信号在  $t_1$  时刻到达系统上的参考麦克风，并且在  $t_2$  时刻到达 DSP。 $t_2 - t_1$  为模数转换器(A/D)所造成的延迟，DSP 计算出抗噪信号后，在  $t_3$  时刻发送到次级扬声器，扬声器在数模转换器(D/A)后将其播放出来。在  $t_4$  时刻，噪声信号到达次级扬声器的位置处，此时扬声器应该已经准备好将反相控制信号播放出来。但是因

为降噪系统各组件之间的距离十分接近,参考麦克风和次级扬声器之间的距离可能只有几厘米。当噪声以  $340\text{m/s}$  的速度传播时,噪声从参考麦克风到次级扬声器所需要的时间  $t_4 - t_1 \approx 30\mu\text{s}$ , 然而参考麦克风延迟、A/D 转换器、DSP 处理、D/A 转换器和扬声器延迟很容易就比  $t_4 - t_1$  的值高出很多。因此,在  $t_4$  时刻播放的反相控制信号在  $t_6$  时刻才能播放出来,在  $t_7$  时刻与噪声信号进行衰减,误差麦克风拾取到残留信号反馈给 DSP 用于下一时刻的更新。

由于  $t_6$  和  $t_4$  时刻之间延迟,导致反相抗噪信号与待衰减噪声之间发生相位偏移。对于低频的噪声信号,由于相位变化缓慢,即便是信号之间产生时延,但相位的移动很小,此时仍然能够实现大部分的噪声衰减。但是对于高频噪声信号,由于相位变化迅速,很小的时延都会造成严重的相位偏移,影响降噪的效果甚至将噪声放大。因此,目前的降噪系统设计主要是消除低频的噪声,而对于中高频的噪声,则是使用海绵等隔音材料包裹,利用被动降噪的方式衰减中高频噪声。在有源噪声控制系统中,自适应滤波器要学习、跟踪初级路径变化,克服对系统求逆所造成的误差。

### 3.2 基于噪声预测的噪声控制系统结构

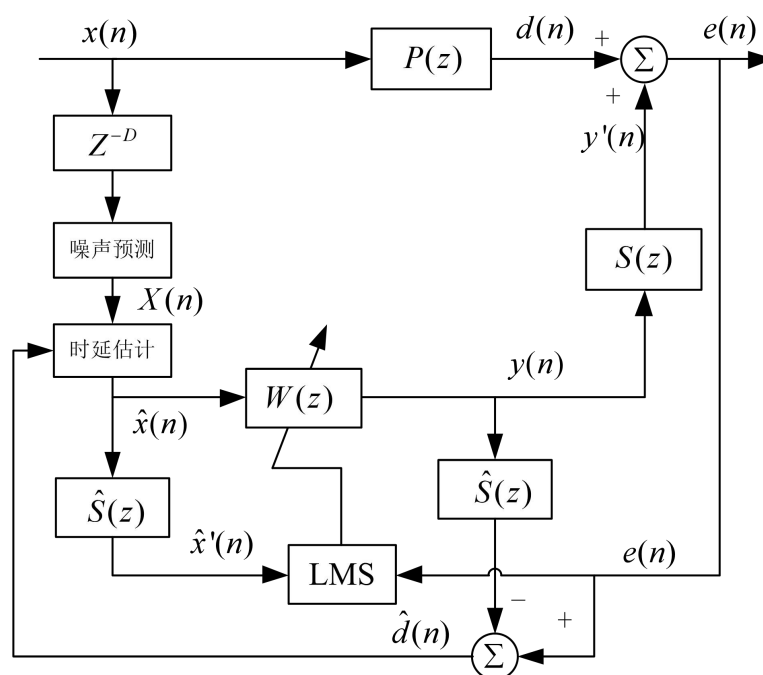


图 3.3 基于噪声预测 ANC 系统框图

Fig. 3.3 Block diagram of ANC system with noise prediction

为了解决上述问题，本文提出基于噪声预测模型的 ANC 系统，系统结构如图 3.3 所示。该系统利用预测的方式来估计未来时刻人声噪音干扰，然后估计出参考信号和待衰减信号之间的时延，以此达到提升对于人声部分降噪性能的目的。

将噪声经由参考麦克风系统以及 A/D 转换到数字域所造成的延迟设为  $D$ ，本系统在 FxLMS 算法的基础之上，增加了噪声预测模块和时延估计模块。噪声预测模块利用预测算法，预测出未来时刻的多个噪声估计值，并在时延范围内依次补偿预测值后缓存为  $\mathbf{X}$ 。

$$\mathbf{X}(n) = \begin{bmatrix} \mathbf{x}(n+1) \\ \mathbf{x}(n+2) \\ \vdots \\ \mathbf{x}(n+L) \end{bmatrix} = \begin{bmatrix} \hat{x}(n+1) & x(n) & \dots & x(n-M+1) \\ \hat{x}(n+2) & \hat{x}(n+1) & \dots & x(n-M+2) \\ \vdots & \vdots & \vdots & \vdots \\ \hat{x}(n+L) & \hat{x}(n+L-1) & \dots & x(n-M+L) \end{bmatrix} \quad (3.5)$$

其中  $M$  为自适应滤波器的长度， $L$  为估计时延的最大值。延迟估计模块的功能是在时延的范围内估计出相对准确的时延。自适应系统本质上就是一个去相关系统，参考噪声  $x(n)$  和待衰减噪声  $d(n)$  之间相关性越高，系统的误差也就越低。因此，可以根据  $d(n)$  和  $\mathbf{X}(n)$  中多个噪声向量之间的相关性，选取最大值即可估计出时延的大小。

人声的激励模型将语音信号分为清音和浊音两种，清音信号之间没有明显的相关性，在利用相关性为基础的预测算法时会影响信号的预测精度，导致预测值和真实值之间的误差较大，进而无法准确的估计出时延  $D$ 。因此，在时延估计模块，首先要对噪声信号进行清浊音检测(Unvoice and Voice Decision)。

清浊音检测目的是为了区分出语音信号中的清音和浊音，浊音信号参与时延估计，而清音由于预测效果不佳则跳过时延估计部分。清浊音检测主要是基于特征的检测方法，常见的区分特征有信号能量、信号短时过零率、信号短时自相关系数等。由于系统实时性的要求，本系统中采用了信号能量和短时过零率结合的方式判决清浊音。

首先根据信号的对数能量进行初次判决，对数能量表示为：

$$E_s = 10 \log(\varepsilon + \frac{1}{N} \sum_{n=1}^N x^2(n)) \quad (3.6)$$

其中  $N$  为噪声信号缓存向量  $\mathbf{x}(n)$  的长度，即为帧长， $\varepsilon$  为非零因子。根据对数能量，设定阈值  $Thr1$ ，当对数能量满足  $E_s \geq Thr1$  时，初步判定当前帧为浊音； $E_s < Thr1$  时则判决为清音。

利用对数能量的判决仅是初步判决，初步判决过后，利用信号的短时过零率作为特征对判决结果进行修正。短时过零率的计算方式为：

$$Zcr = \frac{1}{N-1} \sum_{k=1}^{N-1} |\text{sgn}[x(k)] - \text{sgn}[x(k-1)]| \quad (3.7)$$

其中  $\text{sgn}$  为符号函数，即：

$$\text{sgn}[x(k)] = \begin{cases} 1, & x(k) \geq 0 \\ -1, & x(k) < 0 \end{cases} \quad (3.8)$$

短时过零率表示的是一帧信号内时域上经过零点的几率，对于清音来说有较高的过零率，而浊音的过零率较低。设定短时过零率的阈值为  $Thr2$ ，利用初级判决以及当前帧的短时过零率对当前帧清浊音判断进行修正。如果当前帧在初次判决时为清音，而过去时刻被判断成浊音，并且当前帧满足  $Zcr < Thr2$  时，则修正当前帧的判决结果为浊音。对浊音也同样的处理。

经过上述处理，就可以判断出当前信号噪声缓存是清音还是浊音。若是浊音，则继续进行下一步的相关性计算，即计算噪声向量  $\mathbf{x}(n)$  和待衰减信号向量  $\mathbf{d}(n)$  之间的相关性，但是待衰减信号  $\mathbf{d}(n)$  在实际系统中无法获得，因此通过拾取到的误差信号和抗噪信号估计值计算  $\hat{\mathbf{d}}(n)$ ：

$$\hat{\mathbf{d}}(n) = \mathbf{e}(n) + \hat{\mathbf{y}}'(n) = \mathbf{e}(n) + \mathbf{y}(n) * \hat{\mathbf{s}}(n) \quad (3.9)$$

其中  $\mathbf{y}(n)$  和  $\mathbf{e}(n)$  可以通过滤波器输出和麦克风接收到， $\hat{\mathbf{s}}(n)$  为次级路径的建模值，可以通过离线建模的方式提前获取，通过上式即可得到待衰减信号的估计值  $\hat{\mathbf{d}}(n)$ 。 $\mathbf{x}(n)$  和  $\hat{\mathbf{d}}(n)$  的相关函数为：

$$r(k) = \sum_{n=1}^N \mathbf{x}(n) \hat{\mathbf{d}}(n+k) \quad (3.10)$$

根据相关函数  $r(k)$ ，可以确定时延  $D$ 。

实际系统中，参考麦克风系统函数和 A/D 转换器工作时都保持稳定，所造成的时延也是一个定值。于是，在系统工作开始的一段时间内估计出一个相对准确的时延  $D$ ，即可在后面工作时直接根据  $D$  进行补偿，不需要一直进行时延估计工作。这样，在系统开始工作时，当处理到第一个浊音段时，开始运行时延估计模块，记录所估计出的时延  $D$ 。噪声信号从被麦克风接收到进入到主控制滤波器之间所经过的处理流程如图 3.4 所示。

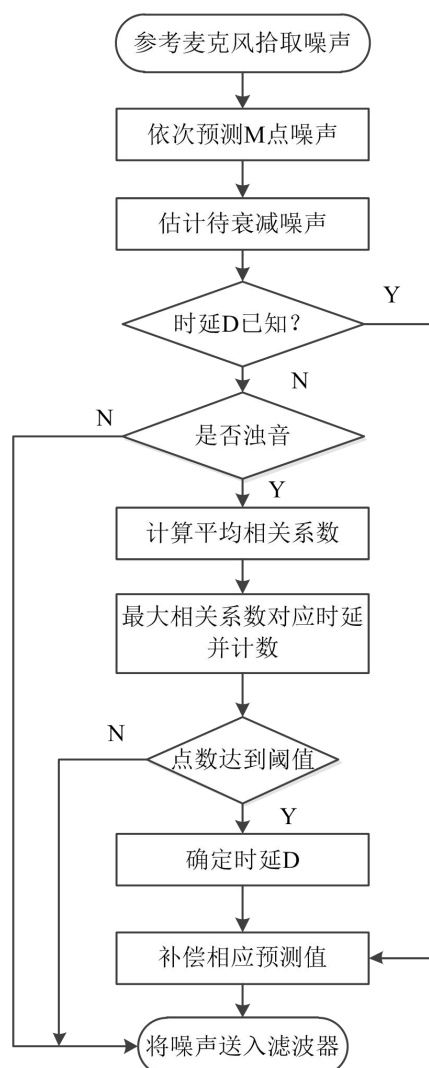


图 3.4 所提出 ANC 系统的工作流程

Fig. 3.4 Workflow of the proposed ANC system

### 3.3 噪声预测算法选择

在本章提出的 ANC 系统中，噪声预测的准确性是整个系统的关键因素，预测算法的性能决定了系统性能提升。在本文第二章中，介绍了集中常用的时间序列预测算法，包括传统的时序预测法、卡尔曼滤波法以及 BP 神经网络法。在本节将通过理论分析和实验测试，在预测性能和计算复杂度两方面考虑，以寻求最适合该系统的预测算法。

#### 3.3.1 各算法实施方案概述

##### (1) 线性预测法

人声的激励模型是将发声机制简化成一个 AR 模型。在该系统中，在噪声预测单元可以通过线性预测的方式，利用过去时刻样值之间的加权之和预测未来时刻的噪声样值  $\hat{x}$ 。

$$\hat{x}(n+D) = \sum_{i=0}^{p-1} w_i \hat{x}(n-i) \quad (3.11)$$

其中  $p$  表示预测阶数， $w_i$  表示线性预测滤波器系数， $D$  表示需要预测的时延。当  $D$  明确给出时，有：

$$\hat{x}(n) = x(n) \quad (3.12)$$

式(3.11)表示为：

$$\hat{x}(n+D) = \sum_{i=0}^{p-1} w_i x(n-i) \quad (3.13)$$

因此当前时刻的预测样值  $\hat{x}(n)$  为：

$$\hat{x}(n) = \sum_{i=0}^{p-1} w_i x(n-D-i) \quad (3.14)$$

用矩阵表示为：

$$\hat{x}(n) = \mathbf{w}^T \mathbf{x}(n-D) \quad (3.15)$$

其中  $\mathbf{w}$  为预测系数矢量， $\mathbf{x}(n-D)$  表示从过去  $D$  时刻开始的信号采样矢量，表示为

$$\mathbf{x}(n-D) = [x(n-D), x(n-D-1), \dots, x(n-D-M+1)]^T \quad (3.16)$$

可以看到，当  $D=1$  时，式(3.13)即为标准的语音线性预测公式，可以利用 Levinson-Durbin 算法进行快速求解，得到单步预测系数向量  $\mathbf{w}$ 。当  $D>1$  时，也可以利用求得的单步预测系数进行迭代  $D$  次来获得估计值  $\hat{x}(n)$ 。然而，由于预测滤波器是稳定的，无限递推单步预测的最后结果将会收敛到 0，所以当  $D$  的值过大时，不能利用此方法。

当系统的延迟  $D$  较小时，可以利用迭代  $D$  次的方式获取未来时刻的估计样值。首先通过单步预测的表达式  $\hat{x}(n+1) = \mathbf{w}^T \mathbf{x}(n)$  计算  $n+1$  时刻估计值，利用 Levinson-Durbin 快速求解出预测滤波器系数  $\mathbf{w}$ 。此时  $n+1$  时刻的噪声估计值表示为：

$$\hat{x}(n+1) = \sum_{i=0}^{M-1} w_i x(n-i) \quad (3.17)$$

展开后得

$$\hat{x}(n+1) = w_0 x(n) + w_1 x(n-1) + \dots + w_{L-1} x(n-M+1) \quad (3.18)$$



其中  $M$  为主控制滤波器的长度，得到第  $n+1$  时刻噪声估计值后，利用迭代  $D$  次的方式预测从  $n+2$  到  $n+D$  时刻的噪声，每次迭代时上一时刻的预测值也同样参与计算，表示为：

$$\hat{x}(n+2) = w_0 \hat{x}(n+1) + w_1 x(n) + \dots + w_{L-1} x(n-M+2) \quad (3.19)$$

$$\vdots$$

$$\hat{x}(n+D) = w_0 \hat{x}(n+D-1) + w_1 \hat{x}(n+D-2) + \dots + w_{L-1} x(n-M+D) \quad (3.20)$$

由此，可以在  $n$  时刻得到未来  $D$  个抽样点的噪声预测值  $\hat{x}(n+1) \dots \hat{x}(n+D)$ 。根据系统时延的估计最大值预测  $L$  个未来采样值后，依次和真实采样值构成输入噪声缓存向量  $\mathbf{x}$ ，即：

$$\mathbf{x}(n+1) = [\hat{x}(n+1), x(n), \dots, x(n-M+1)]^T \quad (3.21)$$

$$\vdots$$

$$\mathbf{x}(n+L) = [\hat{x}(n+L), \hat{x}(n+L-1), \dots, x(n-M+L)]^T \quad (3.22)$$

将噪声向量  $\mathbf{x}$  构造成矩阵  $\mathbf{X}(n)$  后，进入到下一时延估计模块。

$$\mathbf{X}(n) = \begin{bmatrix} \mathbf{x}(n+1)^T \\ \mathbf{x}(n+2)^T \\ \vdots \\ \mathbf{x}(n+L)^T \end{bmatrix} \quad (3.23)$$

## (2) 扩展卡尔曼滤波法

卡尔曼滤波在进行预测之前，首先要对语音信号建模，以确定系统的状态转移方程和观测方程。线性预测时基于语音生成模型，将人声信号利用线性系统进行建模。但是，实际的人声信号存在一定的非线性，对语音考虑使用以下非线性系统进行建模：

$$x(n) = \sum_{i=1}^p a_i x(n-i) + \sum_{i=1}^p b_i x^2(n-i) + Gw(n) \quad (3.24)$$

其中  $w(n)$  是均值为 0、方差  $\sigma_w^2$  的高斯白噪声。在考虑加性噪声的情况时，观测到带噪的语音信号为：

$$y(n) = x(n) + v(n) \quad (3.25)$$

其中  $v(n)$  是 0 均值、方差为  $\sigma_v^2$  的高斯白噪声。根据式(3.24)，可以得到单步预测时系统的状态方程为：

$$f(\mathbf{x}) = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{x}^2 \quad (3.26)$$

其中  $A=[a_1, \dots, a_p]$ ,  $B=[b_1, \dots, b_p]$ , 它们均为语音建模时根据最小均方误差准则得到的预测系数。将上式进行泰勒展开并保留一次项, 则有:

$$f(\mathbf{x}) = f(\mathbf{x}_0) + \nabla f(\mathbf{x})|_{\mathbf{x}=\mathbf{x}_0} (\mathbf{x} - \mathbf{x}_0) \quad (3.27)$$

其中  $\mathbf{x}_0 = \mathbf{x}_{k|k-1}$ ,  $\nabla$  为求梯度操作。此时, 已经将  $f(*)$  的非线性方程转换成上式的线性方程, 状态方程表示为:

$$\mathbf{x}_k = \tilde{\mathbf{F}}\mathbf{x}_{k-1} + \mathbf{G}\mathbf{w}_{k-1} \quad (3.28)$$

其中状态变量  $\mathbf{x}_{k-1} = [x(n), x(n-1), \dots, x(n-p+2)]^T$ ,  $\tilde{\mathbf{F}}$  为状态转移矩阵:

$$\tilde{\mathbf{F}} = \begin{bmatrix} \nabla f(\mathbf{x}_{k|k-1}) \\ 1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (3.29)$$

系统观测方程为线性方程:

$$\mathbf{y}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k \quad (3.30)$$

观测矩阵  $\mathbf{H}$  为  $1 \times p$  阶的矩阵:

$$\mathbf{H} = [1, 0, \dots, 0] \quad (3.31)$$

激励噪声和观测噪声为 0 均值的高斯白噪声, 方差分别为  $\sigma_w^2$  和  $\sigma_v^2$ 。当这些系统参数已知时, 就可以依据式(2.55)~(2.59)得到系统的状态估计。当需要多步预测时, 只需要在每一帧内修改状态变量, 通过迭代的方式即可获得相应的多步预测值。

### (3) 长时预测法

语音信号存在两种类型的相关性, 一种是线性预测理论中分析的样点之间的短时相关性, 另二种是相邻基音周期之间的长时相关性。语音中的浊音信号具有准周期性, 相邻周期内的样本之间具有很大的相关性, 可以利用这种关系, 可以对语音信号进行粗略的估计。区别于利用相邻样点之间相关性的短时预测法, 本文将基于基音周期的预测法称为“长时预测”。在预测时, 首先对每一帧的语音信号求出基音周期  $d$ , 单步预测噪声值即可方便的估计为:

$$\hat{x}(n+1) = x(n-d) \quad (3.32)$$

利用自相关法, 可以估计出语音信号的基音周期。首先对每一帧信号求出该帧信号的自相关函数:

$$R(k) = \sum_{m=0}^{N-1-k} x(m)x(m+k) \quad , 0 \leq k \leq K \quad (3.33)$$

其中  $K$  是最大的延迟点数,  $N$  为帧长。当求出该帧的自相关系数后, 去除掉前面的 10 个点的影响后, 最大的自相关系数所对应的延迟点数即为该帧信号的基音周期。

该方法利用了语音信号的长时相关性, 虽然计算上十分简单, 但是仍能保证一定程度的预测准确性。

#### (4) BP 神经网络法

利用 BP 神经网络预测时, 只需要将网络输入和相应的监督信号给出, 设定理想误差或者最大训练次数即可。神经网络会自动的向着误差最小的方向迭代, 直至达到训练目标或者最大训练次数。

在训练阶段, 需要给网络提供足够的数据进行学习, 网络输入设为:

$$\mathbf{x}(n) = [x(n-l+1), x(n-l+2), \dots, x(n-1), x(n)]^T \quad (3.34)$$

其中  $l$  表示搭建网络输入层的节点数, 以单步预测为例给定监督信号为:

$$d(n) = x(n+1) \quad (3.35)$$

即为  $n+1$  时刻的样值。当网络训练结束后, 整个网络即可以实现预测的功能。想要预测未来多个时刻的样值, 只需要在训练阶段将监督信号改为向量模式即可:

$$\mathbf{d}(n) = [x(n+1), x(n+2), \dots, x(n+D)]^T \quad (3.36)$$

### 3.3.2 预测算法比较与分析

上一小节阐述了几种备选的噪声预测方法, 在本小节将对备选的预测方式进行理论分析和实验测试, 以找到最适合的噪声预测方法。

线性预测和卡尔曼滤波方法在多步预测时, 通过迭代的方式利用估计出的预测值参与计算, 随着预测点数的增加, 必然会导致预测误差的增加; 而长时预测法和 BP 神经网络法在计算时并没有预测值参与计算, 但是如果预测点数过大时, 超过或者接近于信号的基音周期, 则会严重影响长时预测法的预测精度。BP 神经网络并不会因为预测点数而过多的影响精度, 可以根据需要任意的更改训练时的监督信号。为了更具体的比较各种算法的性能, 在单步预测的条件下本章测试各种算法的预测性能。实验数据为 NOIZEUS 中的纯净人声以及不同噪声信噪比情况下的语料, 计算预测值的预测信噪比, 并且利用感知语音质量评估(Perceptual Evaluation of Speech Quality, PESQ)来评价预测信号的语音质量。

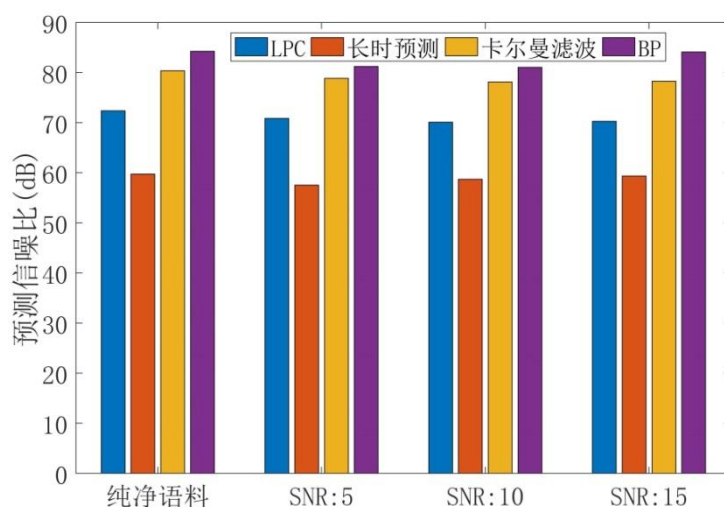


图 3.5 4 种方法在不同噪声信噪比下的预测信噪比

Fig. 3.5 Predictive signal-to-noise ratio of four methods under different signal-to-noise ratios

表 3.1 4 种方法在不同信噪比下预测语音的 PESQ 值比较

Tab. 3.1 PESQ value of speech predicted by four methods under different signal-to-noise ratios

预测方法	线性预测	长时预测	卡尔曼滤波	BP 网络
纯净语音	3.8096	2.8240	4.4090	4.4487
SNR=5	3.7470	2.9686	4.4470	4.4939
SNR=10	3.7263	3.0174	4.4263	4.4958
SNR=15	3.7091	3.0392	4.4091	4.4905

图 3.5 展示了在纯净语音和不同噪声信噪比的情况下，四种预测算法的预测信噪比，表 3.1 为不同情况下预测语音的 PESQ 比较。可以看出，无论是在纯净语音还是在带噪语音中，四种方法都保持稳定的性能，并没有因为噪声的加入而对预测精度造成严重影响。长时预测法因为算法思路简单，预测性能相较于其他三种最差，预测出的语音信号 PESQ 值也是四种方式中最低。卡尔曼滤波法和 BP 神经网络的预测性能接近。BP 神经网络因为优秀的建模能力，拥有四种方式中最高的预测精度和最高的语音质量，但是神经网络的缺点也很明显，计算量是所有方法中最高的，并且很容易出现过拟合的现象，泛化性不强，同时要保证训练数据的全面性也是一个重要考验。

所以综上分析，卡尔曼滤波方法预测精度相对较高，拥有较高的语音质量，算法计算量也可以保证在实时系统中运行，所以选取卡尔曼滤波方法作为该系统中噪声预测模块算法。

### 3.4 实验结果及分析

为验证本章所提出的基于噪声预测 ANC 控制系统的有效性, 本节将通过实验测试其实际性能, 并与不进行时延补偿的 ANC 系统对比。本系统主要是针对现有 ANC 系统由于硬件时延导致对人声信号降噪效果差所提出的改进算法, 所以实验主要针对人声信号进行对比, 实验数据包括 NOIZEUS 数据库中抽取的纯净人声语料以及相应的在不同噪声场景条件下的语料。

所有实验的基本参数相同, 其中系统和语音信号的采样率为 16kHz, 在噪声输入端设置了三个采样点的延迟, 即  $D=3$ 。主控制器为 80 阶的 FIR 滤波器, 次级路径建模滤波器为 12 阶 FIR 滤波器。初级路径  $P(z)$  和次级路径  $S(z)$  分别为 18 阶和 12 阶的 FIR 滤波器, 次级路径估计值  $\hat{S}(z)$  通过离线方式估计得到。

表 3.2 不同预测点数的噪声残留(dB)  
Tab. 3.2 Noise residue of different prediction points

文件名/RMS	不预测	预测 1 点	预测 2 点	预测 3 点
sp01	-43.05	-46.56	-47.33	-47.32
sp02	-39.08	-42.72	-42.74	-42.59
sp03	-39.00	-42.70	-43.04	-42.75
sp04	-37.69	-41.50	-41.82	-41.40
sp05	-39.71	-43.66	-44.00	-43.90
sp11	-38.40	-42.13	-42.67	-42.38
sp12	-38.90	-42.50	-42.48	-42.34
sp13	-40.58	-45.80	-45.88	-45.44
sp14	-40.30	-44.80	-45.38	-45.25
sp15	-39.87	-43.57	-43.34	-43.21

表 3.2 为纯净语音做噪声, 利用线性预测法预测时, 不同点数的系统残留噪声的 RMS 值。其中 sp01-05 为男性语料, sp11-15 为女性语料。可以看到经过噪声补偿后, 系统的降噪性能明显提升了 3-4dB, 但是残留 RMS 值并没有随着预测点数的增加而降低, 而是一个先降低后增加的过程。在理想情况下, 随着补偿点数越接近系统延迟, 系统降噪效果应该越好, 但是预测算法存在误差, 并且后面时刻的预测会用到先前时刻的预测值, 导致预测误差随着点数的增加而增大, 也就导致了表 3.2 中的现象。

#### 3.4.1 纯净语音性能对比

本小节测试当输入信号为纯净人声时, 对比所提出的系统和没有时延补偿的 ANC 系统的降噪性能。

表 3.3 纯净语料下噪声残留对比(dB)

Tab. 3.3 Comparison of residual noises under pure voice

文件名/RMS	无时延补偿	所提系统	增益
sp01	-43.05	-46.36	3.31
sp02	-39.08	-41.94	2.86
sp03	-39.00	-42.68	3.68
sp04	-37.69	-41.38	3.69
sp05	-39.71	-43.56	3.85
sp11	-38.40	-41.38	2.98
sp12	-38.90	-42.19	3.29
sp13	-40.58	-44.35	3.77
sp14	-40.30	-44.51	4.21
sp15	-39.87	-43.15	3.28

表 3.3 为纯净语料时残留噪声 RMS 值对比,通过计算对比,所提出系统相比于不处理时延的 ANC 系统,降噪效果平均提升了大约 3dB,证明其在纯净语音时的有效性。

### 3.4.2 带噪语音性能对比

表 3.4 3 种环境下系统的性能提升(dB)

Tab. 3.4 System performance improvement in three environments

文件名	Airport	Babble	Street
sp01	3.53	3.41	3.28
sp02	3.31	3.44	3.39
sp03	3.24	3.33	3.31
sp04	2.83	2.97	3.00
sp05	3.54	3.48	3.31
sp11	3.81	3.62	2.75
sp12	3.21	3.32	3.25
sp13	4.08	3.96	4.42
sp14	3.65	3.54	3.83
sp15	2.74	2.95	2.70

实际使用的时候,出现纯净人声噪声的情况极少,语音中都会存在一些噪声干扰。本小节测试当输入信号为不同环境下的带噪人声时,该系统是否仍然有效。此处比较了三个常见的使用场景,包括机场(Airport)、嘈杂人声(Babble)以及街道(Street),其中所有带噪语音的信噪比均为 10dB。从表 3.4 可以看出,尽管存在背景噪声,本系统仍然能保持与纯净语音情况下相似的性能提升,这说明了该系统在实际使用中的可行性。

## 3.5 本章小结

本章提出了一种基于噪声预测的 ANC 系统。该系统在传统 ANC 系统的基础上，考虑了参考麦克风和噪声传播的时延，增加了噪声预测和时延估计模块。为了补偿时延进而提升对人声的衰减能力，对比评估了多种时间序列的预测方法，最后选取扩展卡尔曼滤波作为噪声预测模块算法。通过实验测试对比，所提算法不仅对纯净语料有大约 3dB 的降噪收益，在不同带噪语音的情况下也能保证和纯净语料相似的性能提升。

## 4 基于 GRU 网络的次级路径在线建模

在有源噪声控制系统中，为了达到最理想的降噪效果，不能忽略次级路径对反相控制噪声的影响，FxLMS 算法在 LMS 的基础上补偿了次级路径的影响，所以获得准确的次级路径模型至关重要。次级路径的辨识方法在第二章中介绍有离线和在线两种方法。离线建模的方式操作简便，但是当次级路径发生变化时，系统中的离线建模值无法跟踪其系统函数，会对系统的性能造成影响。在线建模的方式虽然操作较为复杂，但是能够有效地应对时变的次级路径系统，不断地跟踪学习其系统函数。本章提出了一种基于 GRU 网络的次级路径在线建模算法。即首先对次级路径在线建模算法进行分析，分析影响次级路径建模准确性的因素；然后提出一种基于 GRU 网络的次级路径在线建模算法，并通过实验证明所提算法的优越性。

### 4.1 次级路径在线建模算法分析

基于附加随机噪声的次级路径在线建模算法已经在第二章中详细介绍，在这里只给出系统框图和滤波器更新公式。

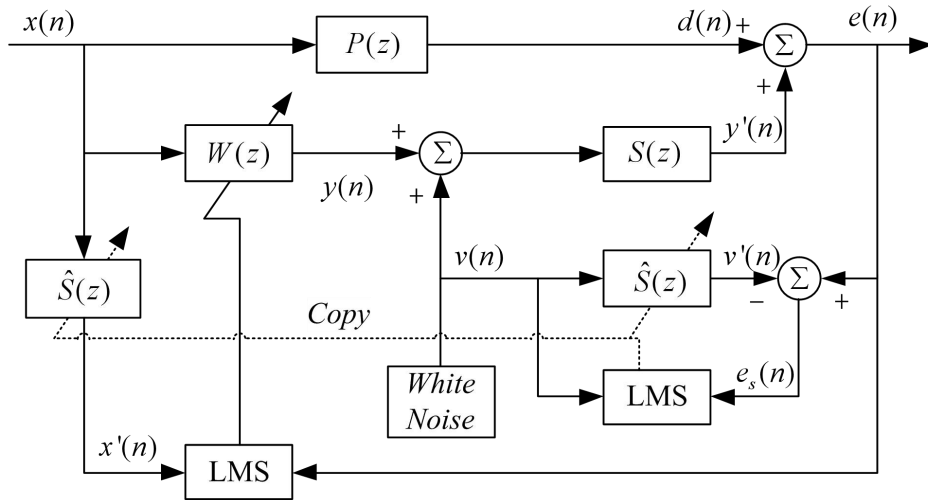


图 4.1 次级路径在线建模框图

Fig. 4.1 Block diagram of secondary path online modeling

次级滤波器建模滤波器更新公式为：

$$\hat{s}(n+1) = \hat{s}(n) + \mu v(n) [d(n) + y(n) * s(n) + v(n) * s(n) - v(n) * \hat{s}(n)] \quad (4.1)$$

主控制滤波器更新公式为：



$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu_w \mathbf{x}'(n)e(n) \quad (4.2)$$

附加随机噪声的次级路径建模算法主要性能包括自适应滤波器的收敛速度、主控制器和建模滤波器的建模精度、以及残留噪声功率。这些性能指标在系统中受很多因素影响，其中影响较大的四个因素为：辅助噪声功率、建模滤波器的最佳步长、抗突变稳定性以及主控制器和次级路径建模滤波器之间的独立性<sup>[44]</sup>。

#### 4.1.1 辅助噪声功率要求

当次级的辅助噪声持续的通入时，误差信号  $e(n)$  表示为：

$$e(n) = d(n) + \mathbf{y}(n) * \mathbf{s}(n) + v'(n) \quad (4.3)$$

其中， $v'(n) = v(n) * s(n)$  表示次级声源  $v(n)$  经过次级路径后的信号。将  $\mathbf{w}(n)$  的更新误差定义为  $\mathbf{f}(n)$

$$\mathbf{f}(n) = \mathbf{w}(n) - \mathbf{w}_{opt}(n) \quad (4.4)$$

其中  $\mathbf{w}_{opt}(n) = [w(0), w(1), \dots, w(n-N+1)]^T$  为理想最优滤波器系数矢量。将  $\mathbf{w}(n+1)$  减去  $\mathbf{w}(n)$  可得

$$\mathbf{f}(n+1) = \mathbf{f}(n) + \mu_w \mathbf{x}'(n) [d(n) + s(n) * \mathbf{y}(n)] + \mu_w \mathbf{x}'(n)v'(n) \quad (4.5)$$

因为  $v'(n)$  与  $x(n)$  一般不相关，上式中  $\mu_w \mathbf{x}'(n)v'(n)$  表示为主控制器  $\mathbf{w}(n)$  更新过程中的干扰项，这种干扰和次级辅助声源的功率成正比，并在  $\mathbf{w}(n)$  收敛的过程中，这种干扰会越来越突出，因此次级声源的功率是在线建模 ANC 系统重要性能指标。在次级路径建模的过程中， $v(n)$  是所需要的激励信号，但是对于主控制滤波器  $\mathbf{w}(n)$  却是一种干扰信号。在主控制器收敛的过程中，参考信号  $\mathbf{x}(n)$  对于主控制器  $\mathbf{w}(n)$  而言是期望的输入信号，而对  $\hat{\mathbf{s}}(n)$  来说却是干扰信号。因此， $v(n)$  和  $\mathbf{x}(n)$  在  $\mathbf{w}(n)$  和  $\hat{\mathbf{s}}(n)$  中分别充当相反的角色。

设  $v(n)$  和  $\mathbf{x}(n)$  是两个相互独立且均值为零的高斯白噪声，其方差分别为  $\sigma_v^2$  和  $\sigma_x^2$ 。为了体现信号功率对算法性能的影响，使用归一化最小均方(Normalized Least Mean Square, NLMS)算法更新次级路径滤波器  $\hat{\mathbf{s}}(n)$ ，即

$$\hat{\mathbf{s}}(n+1) = \hat{\mathbf{s}}(n) + \mu_s \frac{\mathbf{v}(n)e_s(n)}{N\sigma_v^2} \quad (4.6)$$

其中  $N$  为  $\hat{\mathbf{s}}(n)$  的长度， $e_s(n) = e(n) - v'(n)$  表示次级建模滤波器的误差信号。将  $e_s(n)$  展开后，可得：

$$e_s(n) = e(n) - v(n) * \hat{\mathbf{s}}(n) \quad (4.7)$$

定义  $\mathbf{s}_e(n) = \hat{\mathbf{s}}(n) - \mathbf{s}(n)$  为次级路径的建模误差，则式(4.7)可写成：

$$e_s(n) = d(n) + \mathbf{y}(n) * \mathbf{s}(n) - \mathbf{v}(n) * \mathbf{s}_e(n) \quad (4.8)$$

将  $\mathbf{p}(n)$ ,  $\mathbf{w}(n)$  和  $\mathbf{s}(n)$  对参考信号的影响表示成一个响应函数  $\mathbf{b}(n)$ :

$$\mathbf{b}(n) = \mathbf{p}(n) + \mathbf{w}(n) * \mathbf{s}(n) \quad (4.9)$$

令  $e_x(n) = \mathbf{x}(n) * \mathbf{b}(n)$ , 式(4.8)可表示成:

$$e_s(n) = \mathbf{x}(n) * \mathbf{b}(n) - \mathbf{v}(n) * \mathbf{s}_e(n) = e_x(n) - \mathbf{v}(n) * \mathbf{s}_e(n) \quad (4.10)$$

因此  $e_x(n)$  与  $\mathbf{v}(n)$  不相关。对于次级路径建模滤波器来说,  $\mathbf{v}(n)$  是期望的输入, 而  $e_x(n)$  则是干扰信号, 因此可以定义建模滤波器  $\mathbf{s}(n)$  的信噪比  $SNR_s(n)$ :

$$SNR_s(n) = \frac{\sigma_v^2}{\sigma_{e_x}^2} \quad (4.11)$$

定义  $\varepsilon_s(n) = -\mathbf{v}(n) * \hat{\mathbf{s}}_e(n)$ , 则  $e_s(n) = e_x(n) + \varepsilon_s(n)$ , 并且  $\varepsilon_s(n)$  和  $e_x(n)$  不相关。将  $\mathbf{s}_e(n)$  的定义代入到式(4.6)中, 得到次级路径建模均方误差更新公式:

$$\mathbf{s}_e(n+1) = \mathbf{s}_e(n) + \mu \frac{\mathbf{v}(n)e_s(n)}{N\sigma_v^2} \quad (4.12)$$

次级路径的建模均方误差为:

$$q_s(n) = E[\mathbf{s}_e^2(n)] \quad (4.13)$$

对式(4.13)两侧取均方值后, 有:

$$q_s(n+1) = q_s(n) - 2\beta E[\varepsilon_s(n)e_x(n)] + \beta^2 E[e_s(n)\mathbf{v}(n)^T \mathbf{v}(n)e_s(n)] \quad (4.14)$$

其中  $\beta = \frac{\mu_s}{N\sigma_v^2}$ , 将上式进一步展开<sup>[45]</sup>, 可得:

$$q_s(n+1) = q_s(n) - 2\frac{\mu_s}{N}q_s(n) + (N+2)\frac{\mu_s}{N}q_s(n) + \frac{\mu_s^2\sigma_{e_x}^2(n)}{N\sigma_v^2} \quad (4.15)$$

在  $\hat{\mathbf{s}}(n)$  收敛后或者步长  $\mu_s$  特别小时, 则有  $q_s(n+1) \approx q_s(n)$ 。此时的建模均方误差定义为  $q_s^{opt}(n)$ , 即

$$q_s^{opt}(n) \approx \frac{\mu_s\sigma_{e_x}^2(n)}{(2-(N+2)\sigma_v^2\mu_s)} = \frac{1}{\left[\frac{2}{\mu_s} - (N+2)\right]SNR_s(N)} \quad (4.16)$$

根据式(4.16)可以判断, 当次级路径建模滤波器收敛时, 次级路径的建模误差与辅助噪声的功率成反比, 即辅助噪声的功率越大, 次级路径建模误差也就越小, 也就拥有更高的建模精度, 但同时也给 ANC 系统引入了较大的不相关噪声; 反之,  $e_x(n)$  的功率越大, 建模误差也就越大, 建模精度也就越低。

对于主控制滤波器  $w(n)$  而言, 建模误差  $q_w^{opt}(n)$  和辅助噪声的功率成正比, 与输入的参考信号  $x(n)$  成反比。另外, 由于  $x(n)$  要经过  $\hat{s}(n)$  滤波, 因此  $q_w^{opt}(n)$  还与次级路径的建模精度有关。

#### 4.1.2 最佳步长选取要求

步长是自适应算法的一个重要参数。在保证 ANC 系统稳定的前提下, 步长越大系统收敛的速度也就越快, 跟踪学习能力也就越强。通常情况下, 系统的最佳步长都不是一个定值, 而是根据系统的收敛情况随时变化的。可以通过定义代价函数的方式, 使该代价函数最小, 即可得到该时刻下的最佳步长。对于次级路径建模滤波器来说, 次级滤波器的建模误差更新公式可写成:

$$s_e(n+1) = s_e(n) + \mu(n) \frac{v(n)e_s(n)}{N\sigma_v^2} \quad (4.17)$$

为了确保系统的收敛,  $s_e(n)$  的范数应该随着更新时刻的增加而减小<sup>[16]</sup>, 即

$$E[\|s_e(n+1)\|^2] < E[\|s_e(n)\|^2] \quad (4.18)$$

将  $q_s(n) = E[s_e^2(n)]$  和  $\varepsilon_{s(n)} = -v(n) * \hat{s}_e(n)$  代入(4.17)中后

$$q_s(n+1) - q_s(n) = \mu_s^2(n) E\left\{\frac{\sigma_s^2(n)}{N\sigma_v^2}\right\} - 2\mu_s(n) E\left\{\frac{e_s(n)\varepsilon_s(n)}{N\sigma_v^2}\right\} < 0 \quad (4.19)$$

得到  $n$  时刻步长的范围:

$$0 < \mu_s(n) < 2 \frac{E\{e_s(n)\varepsilon_s(n)\}}{E\{e_s^2(n)\}} \quad (4.20)$$

因为  $e_s(n) = e_x(n) + \varepsilon_s(n)$ , 并且  $e_x(n)$  和  $\varepsilon_s(n)$  相互独立不相关, 所以  $E\{e_x(n)\} = 0$ , 可得

$$E\{e_s(n)\varepsilon_s(n)\} = E\{\varepsilon_s^2(n)\} \quad (4.21)$$

$$E\{e_s^2(n)\} = E\{\varepsilon_s^2(n)\} + E\{e_x^2(n)\} \quad (4.22)$$

根据文献[45]可得:

$$E\{\varepsilon_s^2(n)\} = \sigma_v^2 q_s(n) \quad (4.23)$$

式(4.20)可以写为:

$$0 < \mu_s(n) < 2 \frac{q_s(n)}{q_s(n) + \frac{\sigma_{\varepsilon_x}^2(n)}{\sigma_v^2}} \quad (4.24)$$

选取中间值, 即为次级建模滤波器的最佳步长:

$$\mu_s^{opt}(n) = \frac{q_s(n)}{q_s(n) + \frac{\sigma_{e_x}^2(n)}{\sigma_v^2}} \quad (4.25)$$

从式(4.25)中可以看出, 当  $\sigma_{e_x}^2(n)$  增大时, 最佳步长随之减小。在没有干扰信号即  $e_x(n)=0$  时, 可得最佳步长为 1, 此时次级建模滤波器可以达到最快的收敛速度。然而实际使用中, 不可能没有来自参考信号的干扰, 所以最佳步长都会小于 1, 也就会导致系统建模时收敛速度的减慢。

#### 4.1.3 建模算法的抗突变要求

在分析辅助噪声对建模性能的影响时知道, 次级路径增加的辅助噪声会降低系统的降噪性能。对于时不变的次级路径模型, 在实际使用中, 可以通过在建模收敛时关闭次级声源的方式减少对系统的影响。但是对于时变的次级路径, 需要用自适应滤波器时刻跟踪真实路径的变化情况, 当次级路径模型发生突然剧烈变化时, 相应的建模算法应该具有一定的抗突变能力, 及时调整自身参数以保证整个系统的收敛。

#### 4.1.4 独立性要求

在次级路径建模的收敛过程中, 主控制器也在同时工作, 理想的算法应该保证主控制器和次级路径建模滤波器之间的相互独立, 即应该满足两个独立性条件<sup>[13]</sup>。首先, 次级路径的建模精度不应该取决于主控制器的收敛程度, 其次, 次级路径建模的过程也不应该影响主控制器  $w(n)$  的更新过程。满足这两个独立性条件就能够更好的对次级路径建模, 从而提升系统降噪性能。

上述四个因素在次级路径在线建模过程中起到了重要作用。但是在实际系统中, 即便满足上述四个因素的最佳条件, 对次级路径的建模仍然存在一定误差。这是因为许多声学器件和物理路径的传递函数并不是纯线性系统<sup>[46]</sup>, 都存在着一定的非线性, 而且当输出信号的幅值大于扬声器的上限幅值, 或者输出信号的频率小于扬声器的截止频率时, 次级路径也会表现出非线性特性。而利用线性自适应算法无法映射系统中的非线性分量, 所以使用线性 FIR 自适应滤波对次级路径建模时, 仍然存在无法消除的建模误差。

### 4.2 基于 GRU 网络的次级路径在线建模

#### 4.2.1 现有建模算法分析

上一节中分析了基于附加噪声的次级路径在线建模算法的几个关键因素, 这些因素对次级路径建模算法的性能影响很大。为了改善次级路径的建模算法, 降低次级路径建

模误差对系统的影响,许多学者在 Eriksson 算法之上提出了改进。Bao 提出用一个额外的滤波器  $h(n)$  来降低次级路径建模时的干扰<sup>[15]</sup>, 算法框图如下:

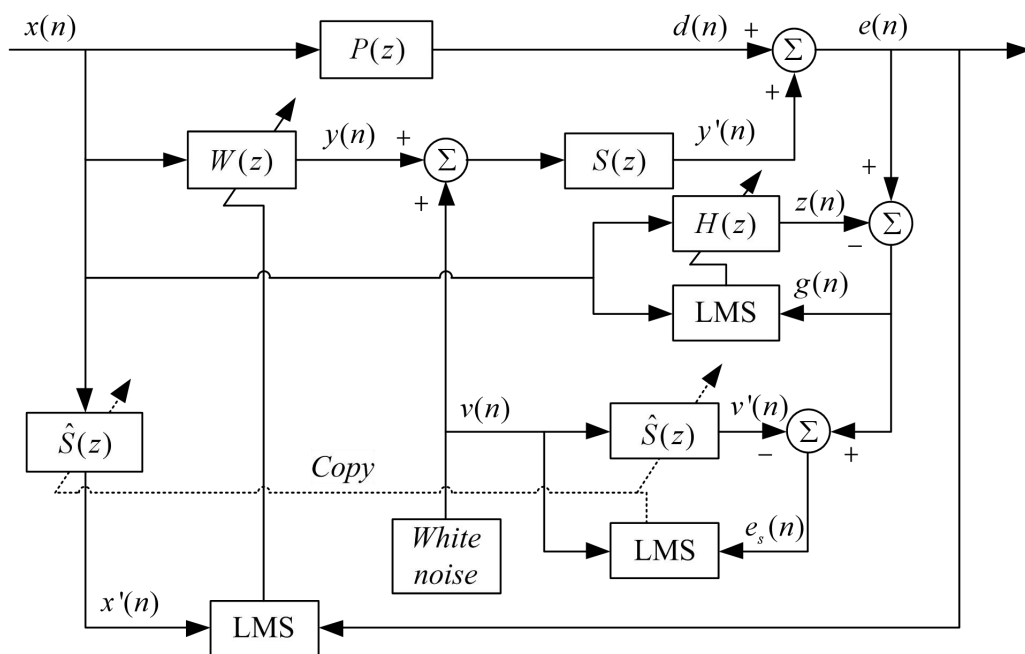


图 4.2 Bao 的算法框图

Fig. 4.2 Block diagram of Bao's algorithm

此时  $\mathbf{b}(n)$  表示为:

$$\mathbf{b}(n) = \mathbf{p}(n) + \mathbf{w}(n) * \mathbf{s}(n) - \mathbf{h}(n) \quad (4.26)$$

从图 4.2 可以看出, 对于  $h(n)$  而言  $x(n)$  是期望信号, 而  $e_v(n) = s(n) * v(n)$  则是干扰信号, 随着  $h(n)$  的收敛  $b(n)$  和  $e_x(n)$  会越来越小, 根据式(4.16)可知, 建模信噪比也会随之提高。但是该算法中, 次级路径和辅助噪声没有改变, 其功率  $\sigma_v^2$  是一个常数项, 所以  $\sigma_{e_v}^2$  也是一个常数, 该算法在次级路径建模过程仍然会对系统中  $w(n)$  造成影响, 无法满足独立性条件的第二条准则。

为了进一步降低  $v(n)$  对  $w(n)$  的干扰, Zhang 提出了一种交叉更新的自适应算法<sup>[17]</sup>。该算法在 Bao 的基础上进行了改进, 算法结构图如下:

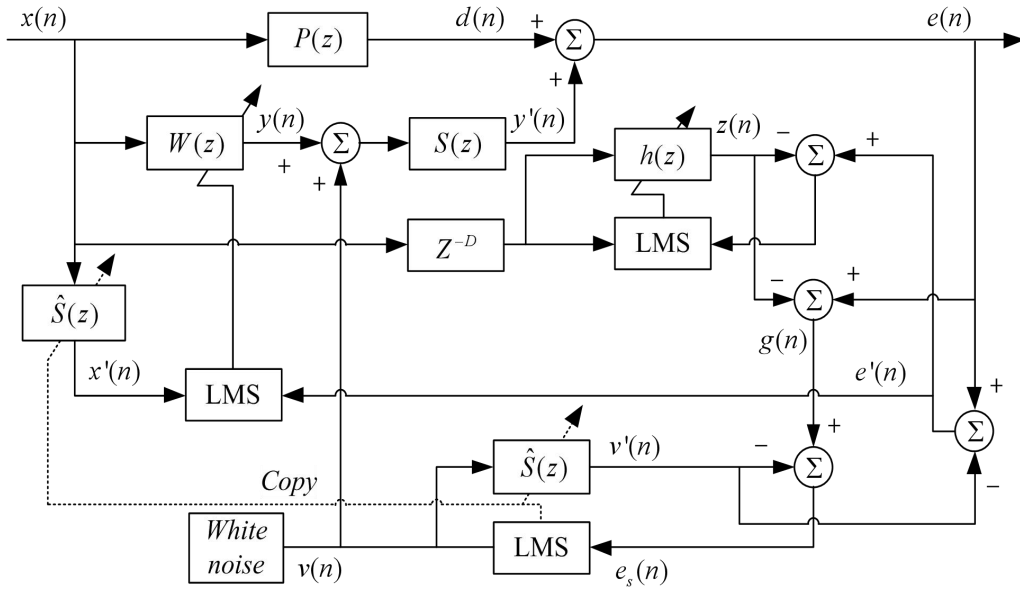


图 4.3 Zhang 的算法框图

Fig. 4.3 Block diagram of Zhang's algorithm

将用于更新  $w(n)$  的误差信号  $e(n)$  中去除  $v(n)$  的干扰得到  $e'(n)$ ，表示为：

$$e'(n) = e(n) - \hat{s}(n) * v(n) \quad (4.27)$$

当  $\hat{s}(n) \approx s(n)$  次级滤波器收敛后

$$e'(n) \approx d(n) + s(n) * y(n) = [p(n) + s(n) * w(n)] * x(n) \quad (4.28)$$

可以看到， $e'(n)$  中辅助噪声的分量  $v(n)$  被抑制，只含有参考噪声  $x(n)$  的分量，此时满足：

$$e_v(n) = v(n) * [s(n) - \hat{s}(n)] \quad (4.29)$$

随着  $\hat{s}(n)$  的收敛， $\sigma_{e_v}^2$  会越来越小，这也就意味着辅助噪声对  $w(n)$  的干扰也会逐渐地消失，不仅满足了独立性要求，而且建模误差  $q_w^{opt}(n)$  也会大大降低，从而获得更精确的初级路径模型。因为  $w(n)$  的准确度直接决定  $e_x(n)$  的大小，会进一步会影响  $\hat{s}(n)$  的收敛速度和建模精度，所以 Zhang 的算法不仅能够使  $w(n)$  达到更好的收敛效果，同时也能提高  $\hat{s}(n)$  的建模精度。

Bao 和 Zhang 的算法在结构中加入了第三滤波器  $h(n)$ ，虽然能够提升 ANC 系统的整体性能，但是由于三个滤波器之间相互依赖，在一个周期内同时进行自适应对于参数和环境的要求很高，很容易出现滤波器失调的现象。Akhtar 提出了一种修正 FxLMS 算法(Modified-FxLMS, MFxLMS)<sup>[18]</sup>，如图 4.4 所示，在结构上仅使用两个自适应滤波器，

并且提出了新的可变步长 LMS 算法(Variable Step Size-LMS, VSS-LMS)以更新次级路径建模滤波器,该算法的步长不同于 NLMS 算法根据参考信号的功率而变化,也不同于其他 VSS 算法在开始阶段选择大步长从而提升收敛速度,而在较小误差的时候选取小步长而准确定位。所提出的 VSS-LMS 算法在开始阶段选取小步长,在收敛的过程中随之增加。这是因为次级路径在建模过程中,作为干扰信号的  $d(n) - y'(n)$  会影响滤波器的收敛速度,甚至可能会使其发散,随着滤波器收敛,该干扰信号在理想情况下为 0,所以应该在较不稳定的初始阶段选取较小步长,随着干扰信号的降低而增加步长。该算法虽然避免使用了第三个滤波器,但是计算上的复杂性相对较高。

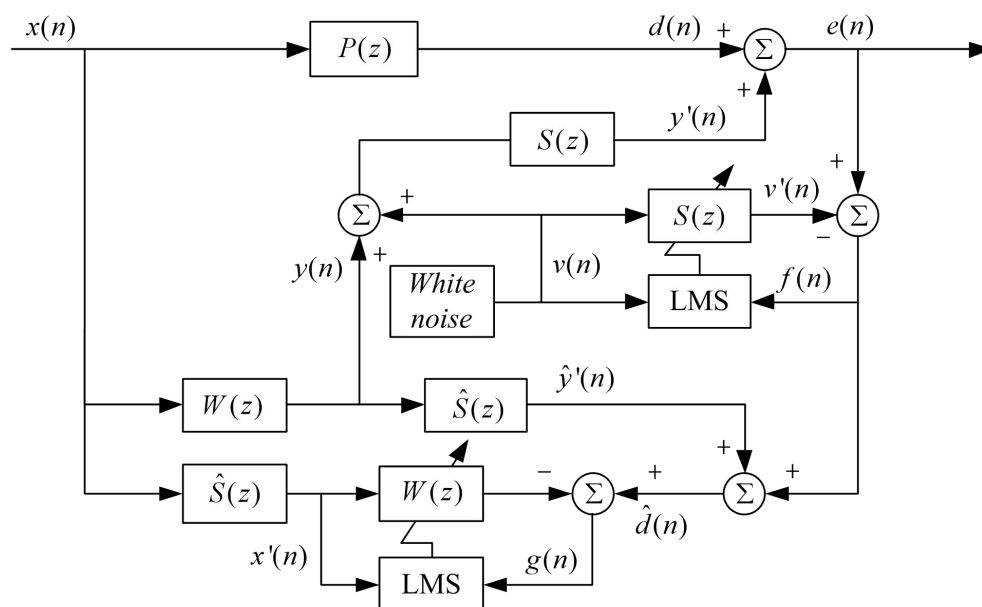


图 4.4 Akhtar 的算法框图

Fig. 4.4 Block diagram of Akhtar's algorithm

实际系统中的非线性特性也会导致基于线性滤波器的建模算法产生无法补偿的非线性误差<sup>[47]</sup>。常用的解决非线性 ANC 问题主要有两种方式,非线性自适应滤波器和非线性自适应算法。Volterra 滤波器是一种典型的非线性自适应滤波器,滤波器输出线性的依赖于滤波器自身参数,因此可以利用线性算法解决非线性问题,是一种线性参数滤波器(Linear-in-the-Parameters Filter, LPF)。这也就意味着当系统需要大量的参数时,Volterra 滤波器的计算量相对较大<sup>[48]</sup>,为了解决这个问题,通常使用二阶 Volterra(second-order Volterra, SOV)滤波器或者三阶 Volterra (Third-order Volterra, TOV)滤波器。

非线性自适应算法包括粒子群优化算法(Particle Swarm Optimization, PSO)、人工神经网络(ANN)算法、遗传算法等。PSO 算法最大的优势就是可以不需要次级路径的在线建模就可以实现降噪<sup>[49]</sup>，但是存在收敛速度缓慢的问题。人工神经网络算法在第二章中以 BP 神经网络为例进行了分析，ANN 算法可以任意精度逼近系统模型，具有良好的非线性建模能力，缺点就是计算量较传统算法大。

#### 4.2.2 基于 GRU 网络的在线建模算法结构

基于以上算法中的限制因素和不足，本节提出了基于 GRU 网络的次级路径建模算法。GRU 网络具有良好的非线性建模能力，对时间序列有较好的处理能力，解决了 RNN 网络固有的梯度消失问题，而计算量要比 RNN 少得多。

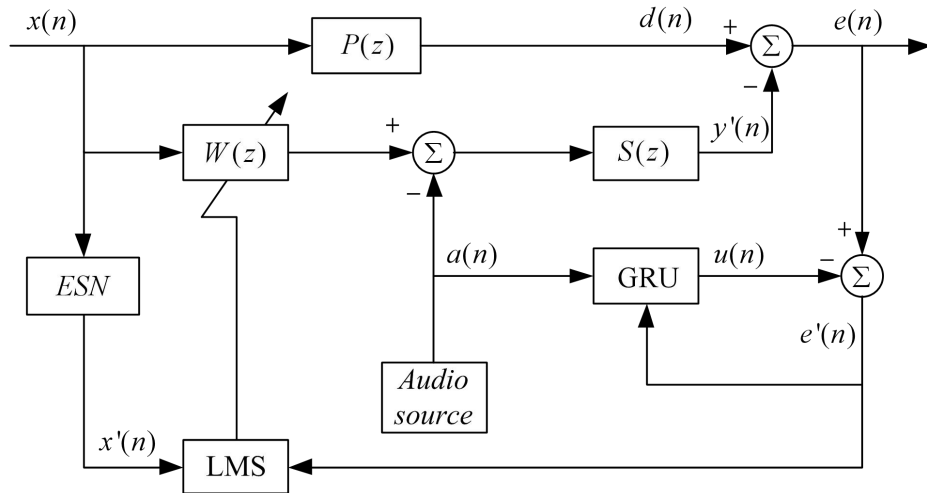


图 4.5 基于 GRU 网络的次级路径在线建模

Fig. 4.5 Online modeling of secondary path based on GRU network

图 4.5 为基于 GRU 网络的次级路径在线建模算法框图。该算法在 Eriksson 算法的基础上，将次级路径建模滤波器替换成 GRU 网络，利用 GRU 网络实时地对次级路径进行建模。网络的输入信号为次级辅助噪声源  $a(n)$ ，监督网络的样本信号为误差麦克风拾取的误差信号  $e(n)$ 。前文已经分析过，随着主控制滤波器的收敛， $e(n)$  中只包含辅助声源  $a(n)$  经过次级路径的信号，即：

$$e(n) \approx a(n) * s(n) \quad (4.30)$$

用  $e(n)$  作为网络的监督信号训练网络，当网络收敛时，网络输出  $a'(n)$  满足

$$a'(n) \approx e(n) \approx a(n) * s(n) \quad (4.31)$$

此时



$$e'(n) = e(n) - a'(n) \approx 0 \quad (4.32)$$

其中  $e'(n)$  作为主控制滤波器的误差信号参与主控制滤波器的下一次更新。

### 4.3 实验结果与分析

本节将对所提算法进行实验仿真，以测试算法的实际性能以及验证算法提出的有效性，并用 Eriksson<sup>[14]</sup>、Zhang<sup>[17]</sup>、Akhtar<sup>[18]</sup>所提出的在线建模算法作为对比实验。

#### 4.3.1 实验配置

环境初级路径和次级路径分别利用 80 阶和 12 阶的 FIR 滤波器仿真。所提算法中主控制器也同样为 80 阶自适应 FIR 滤波器，步长  $\mu_w = 0.01$ ，次级路径设置为 3 层 GRU 网络结构，对比实验的参数配置如表 4.1 所示：

表 4.1 实验参数配置

Tab. 4.1 Experimental parameter configuration

对比算法	参数	取值
Eriksson's	$W(z)$	80 阶
	$S(z)$	12 阶
	$\mu_w$	0.01
	$\mu_s$	0.005
Akhtar's	$W(z)$	80 阶
	$S(z)$	12 阶
	$\mu_{\min}$	0.2
	$\mu_{\max}$	0.8
	$\mu_w$	0.01
Zhang's	$W(z)$	80 阶
	$S(z)$	12 阶
	$H(z)$	80 阶
	$\mu_w$	0.01
	$\mu_s$	0.1
	$\mu_h$	0.1

实验测试文件为 Noisex-92 数据库中 11 个的环境噪声文件，采样率均为 16kHz，本节所有对比图均基于 11 个测试文件的平均值得出。

### 4.3.2 各影响因素下实验对比

#### (1) 辅助噪声功率不同时系统性能比较

次级辅助噪声的功率对次级路径的建模精度以及整个 ANC 系统的降噪性能至关重要，经分析较高功率辅助噪声有利于初级路径的建模，但是也会给整个系统引入无关噪声，影响整个系统的降噪性能，过低的辅助噪声则无法有效建模。理想的建模算法应该既能保证较低的噪声输入，又能保证准确的建模精度。

表 4.2 高功率辅助噪声性能比较(dB)

Tab. 4.2 Performance comparison of high-power auxiliary noise (dB)

文件名/RMS	Eriksson	GRU	增益
babble	-30.62	-39.42	8.80
buccaneer1	-29.95	-35.59	5.64
buccaneer2	-29.97	-36.15	6.18
destroyerengine	-29.22	-35.30	6.08
destroyerops	-26.91	-35.44	8.53
f16	-30.63	-39.12	8.49
Factory1	-29.08	-39.64	10.56
Factory2	-29.15	-38.25	9.10
hfchannel	-30.07	-36.77	6.70
pink	-28.91	-42.86	13.95
white	-31.64	-45.40	13.76

表 4.3 低功率辅助噪声性能比较(dB)

Tab. 4.3 Performance comparison of low-power auxiliary noise (dB)

文件名/RMS	Eriksson	GRU	增益
babble	-36.49	-39.89	3.40
buccaneer1	-32.39	-35.47	3.08
buccaneer2	-32.34	-35.85	3.51
destroyerengine	-32.20	-35.15	2.95
destroyerops	-32.19	-35.27	3.08
f16	-36.20	-38.88	2.68
factory1	-36.88	-39.72	2.84
factory2	-33.28	-38.03	4.75

表 4.3 续

Tab. 4.3 Cont

文件名/RMS	Eriksson	GRU	增益
hfchannel	-30.99	-36.79	5.80
pink	-39.67	-42.84	3.17
white	-42.29	-45.36	3.07

将表 4.2 和表 4.3 比较可以明显观察到, 在利用 Eriksson 算法在线建模时, 使用高功率辅助噪声, 系统残留噪声的 RMS 值明显高于低辅噪声的情况, 降噪能力受到了严重限制, 说明高功率的辅助噪声确实对系统整体的降噪效果产生了不利影响。而较低的辅助噪声对系统引入的额外噪声分量少, 对系统的影响较小, 降噪性能更优。所提出的建模算法在两种情况下均优于 Eriksson 算法的性能, 尤其是在高功率辅助噪声情况下, 其仍然保持了较高的噪声衰减。建模误差定义为:

$$\Delta s(n) = \frac{\sum_{i=0}^{N-1} [\hat{s}_i(n) - s_i(n)]^2}{\sum_{i=0}^{N-1} [s_i(n)]^2} \quad (4.33)$$

图 4.6 表示在两种情况下, 两种算法的次级路径建模误差对比。其中, (a) 为高辅助噪声情况, (b) 为低辅助噪声情况。从图 4.6 中可见, 在较高功率的辅助噪声时, 所有算法均获得了更低的建模误差, 证实了前文分析的正确性。GRU 算法在两种测试条件下均取得了最低的建模误差, 并且准确度更高。

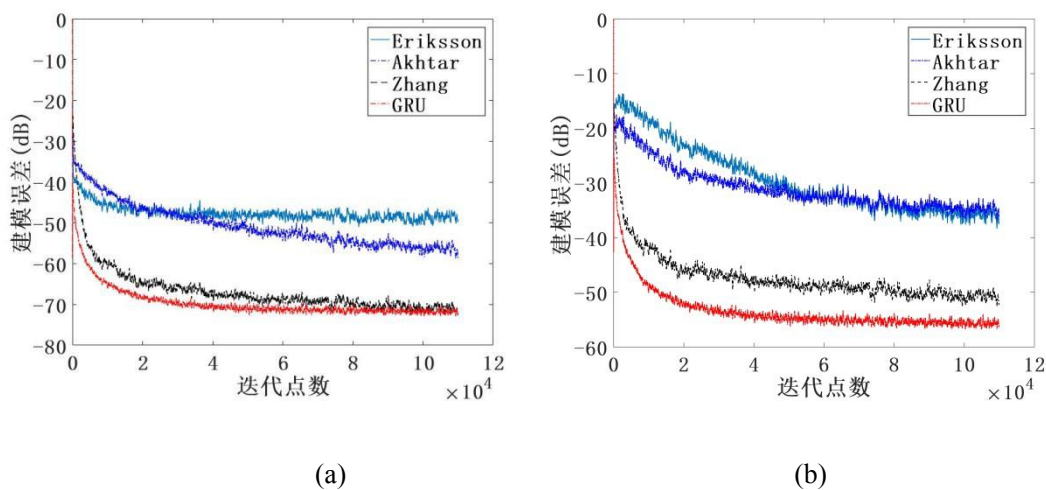


图 4.6 两种辅助噪声建模误差对比

Fig. 4.6 Comparison of two auxiliary noise modeling errors

上述噪声残留对比和建模误差对比的实验结果，均证实 GRU 建模算法对辅助噪声功率的不敏感性，无论辅助噪声功率的大小均能保持比传统算法优秀的降噪性能和建模精度。

### (2) 次级路径突变时系统性能比较

实验设置在 70000 点时次级路径突变。图 4.7 为当次级路径发生突变时，所提算法和其他算法的建模误差对比。从图 4.7 可以看到，Eriksson 和 Akhtar 的算法在发生突变后建模误差急速上升，并且一直维持着较大值，并没有再次收敛的趋势。而 Zhang 提出的算法由于引入第三滤波器  $h(n)$ ，建模滤波器  $\hat{s}(n)$  和主控制器  $w(n)$  之间相互独立，在发散后可以较快地收敛。GRU 建模算法在路径突变时虽然也出现尖峰，但是随着时间的推移，其建模误差也逐渐回落到之前的水平，仍然能再次收敛，并且保证最低的建模误差。因此，本文所提出的 GRU 算法抗突变干扰能力更强，在次级路径发生突变时，仍然能迅速收敛保证稳定。

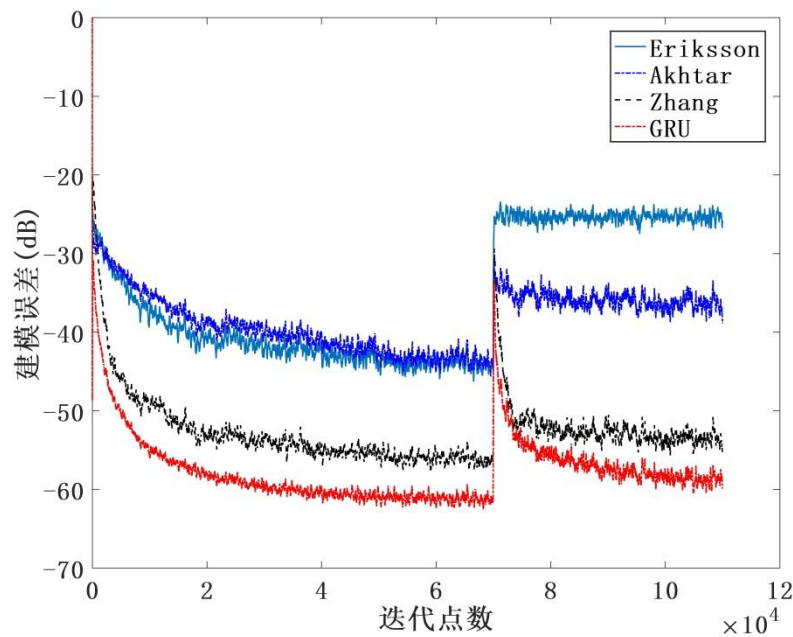


图 4.7 次级路径突变建模误差对比

Fig. 4.7 Comparison of modeling errors of secondary path mutations

### (3) 非线性次级路径性能比较

在实际的有源噪声控制系统中，次级路径经常包含一些非线性分量，从而体现出一定的非线性特性，因此在次级路径中添加非线性分量，以此测试所提出算法在非线性次级路径下的降噪性能。实验中非线性分量表示为：

$$y'(n) = -0.00005676 * [0.35 * y(n)]^2 - 0.0000219165 * [0.35 * y(n-1)]^2 + 0.000073412 * [0.35 * y(n-2)]^2$$

表 4.4 非线性次级路径系统残留噪声(dB)

Tab. 4.4 Non-linear secondary path system residual noise(dB)

文件/RMS	Eriksson	GRU	增益
babble	-35.65	-38.35	2.70
buccaneer1	-32.54	-34.74	2.20
buccaneer2	-32.82	-35.54	2.72
destroyerengine	-32.13	-34.35	2.22
destroyerops	-30.55	-34.62	4.07
f16	-35.26	-37.98	2.72
factory1	-35.44	-38.85	3.41
factory2	-33.46	-36.64	3.18
hfchannel	-32.40	-35.38	2.98
pink	-37.23	-41.69	4.46
white	-39.17	-43.88	4.71

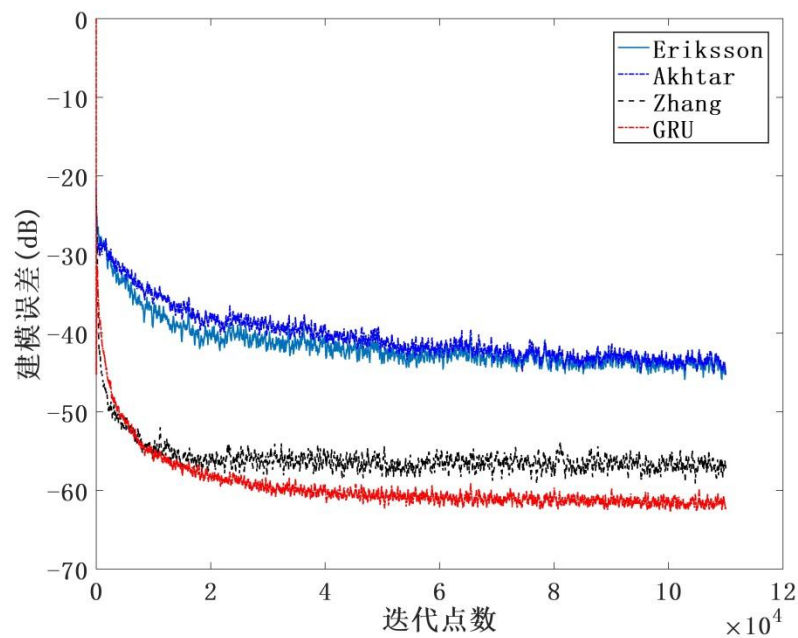


图 4.8 非线性次级路径建模误差

Fig. 4.8 Non-linear secondary path modeling error

表 4.4 为噪声文件经过非线性系统残留噪声的 RMS 值比较。因为自适应线性滤波器只能对线性分量建模，无法模拟非线性的部分，而 GRU 网络由于对非线性系统具备优秀的映射能力，所提的 GRU 算法在非线性系统情况中能获得更好的降噪性能。测试数据也证实了这一结论。图 4.8 为系统中存在非线性分量时次级路径的建模误差，在步长选取最优值的情况下，Zhang 所提出的算法比 Eriksson 和 Akhtar 的算法拥有更高的建模精度，但是无法映射系统中的非线性分量，而基于 GRU 的建模算法具有良好的非线性系统建模能力，且随着网络的收敛过程，其优越性愈发明显。

经过以上几组实验测试比较可以看出，所提算法在辅助噪声功率适应性、抗次级路径突变性以及非线性映射能力方面，均明显优于 Eriksson 等人提出的传统算法。Zhang 提出的算法在参数选取最优时，性能较接近所提的 GRU 网络建模算法，但是 Zhang 的算法无法映射系统中的非线性分量，而且需要在一个周期内调整三个滤波器，每个滤波器的步长就变得十分重要。在实际应用中，步长很难稳定在最佳状态，而且随着噪声的变化，最佳步长也是时变的，此时 Zhang 的算法在实际使用中很难达到最优的状态。而 GRU 网络建模算法在上述情况下均有良好的建模性能。

#### 4.4 本章小结

本章提出了一种基于 GRU 网络的次级路径在线建模算法。具体地，分析了传统滤波器建模方式中影响降噪性能的关键因素，并推导出对次级路径建模精度和 ANC 降噪性能的影响。为避免这些影响，本章提出了基于 GRU 网络的建模算法，通过多组实验验证了该算法在线性和非线性系统中均能保证较低的建模误差，从而有效地提升了有源噪声控制系统的降噪能力。

## 结 论

噪声污染已经是人类社会中最常见的污染源之一，长期暴露在高噪声环境下，会对人的身体和心理健康都造成不可逆的影响。减轻噪声干扰的方法分为主动降噪和被动降噪两种方式，被动方式是以阻断噪声的传播路径为主，而主动方式则是利用算法衰减人耳能听到的噪声。然而，在实际使用中，有源噪声控制系统仍存在多种问题，使得降噪效果不理想。

为了提升有源噪声控制系统的性能，本文对有源降噪技术进行分析和研究，主要工作如下：

(1)针对物理系统中无法解决的时延问题，本文提出了一种基于非线性卡尔曼预测的降噪系统。该方法采用扩展卡尔曼滤波的方法，预测未来时刻的噪声样点，再利用误差和计算出的控制信号合成 $d(n)$ 的估计值 $\hat{d}(n)$ ，计算预测信号和 $\hat{d}(n)$ 之间的相关性，以此确定系统的时延 $D$ 。在确定时延之后，为输入主控制器的信号补偿相应的预测信号，以此提高系统的降噪性能。仿真实验结果表明，在纯净语音下，该算法能够使系统获得大约 3dB 的性能提升；在带噪语音的情况下，也能保证同样的降噪提升量。

(2)对于传统次级路径在线建模算法，分析了影响次级路径建模精度的几个关键因素，并基于神经网络的思想，用 GRU 网络取代次级建模滤波器，实现了次级路径建模。该方法利用次级路径辅助噪声作为网络输入，将误差麦克风信号 $e(n)$ 作为网络的监督信号，进行在线训练。仿真实验表明，基于 GRU 网络的次级路径建模算法，拥有较高的建模精度和良好的鲁棒性。

因为时间水平有限，本文工作还存在许多问题需要进一步的研究。首先，扩展卡尔曼滤波通过一阶泰勒展开将非线性系统转换成线性处理，而线性化处理后的噪声不再满足独立、高斯分布的条件，容易导致滤波器发散。其次，GRU 网络虽然比 RNN 网络计算量少，但是对于实时系统使用还需要进一步降低网络模型的计算量。

## 参 考 文 献

- [1] 樵地英. 中国城市噪声污染的危害及控制技术的探讨[J]. 能源与节能, 2013(04):81-83.
- [2] Tor H. Oiamo, Gunn Marit Aasvang, Noise and Health [J]. International Encyclopedia of Human Geography (2 Ed), 2020, 409-413.
- [3] 陈克安. 有源噪声控制(第二版)[M]. 北京国防工业出版社, 2003.
- [4] 盛美萍, 王敏才, 孙进才. 噪声与振动控制技术基础[M]. 北京:科学出版社, 2001.
- [5] Eriksson L J, Allie M C, Bremigan C D, et al. Active noise control and specifications for fan noise problems[C]. Proceedings of Noise Conference, Purdue University, USA, 1988: 273-278.
- [6] Denenberg J N. Anti-noise-quieting the environment with active noise cancelation technology[J]. IEEE Potentials. 1992, 11(2): 36-40.
- [7] Lueg P.: Process of silencing sound oscillations, U.S. [P].2043416, June 9, 1936.
- [8] Olson H F , Everett G. Electronic sound absorber[J]. The Journal of the Acoustical Society of America, 1953, 25(6): 1130-1136.
- [9] Olson H F. Electronic control of noise, vibration, and reverberation[J]. The Journal of the Acoustical Society of America, 1956, 28.5: 966-972.
- [10] Ross C F. An algorithm for designing a broadband active sound control system[J]. Journal of sound and vibration, 1982, 80(3): 373-380.
- [11] Ross C F. An adaptive digital filter for broadband active sound control[J]. Journal of Sound and Vibration, 1982, 80(3): 381-388.
- [12] Burgess J C. Active adaptive sound control in a duct: A computer simulation[J]. The Journal of the Acoustical Society of America, 1981, 70(3): 715-726.
- [13] Widrow B, Stearns S, et al. Adaptive signal processing [M]. Prentice-Hall, 1985.
- [14] Eriksson L J, Allie M C. Use of random noise for on - line transducer modeling in an adaptive active attenuation system[J]. The Journal of the Acoustical Society of America, 1989, 85(2): 797-802.
- [15] Bao C, Sas P, Van Brussel H. Adaptive active control of noise in 3-D reverberant enclosures[J]. Journal of sound and vibration, 1993, 161 (3): 501-514.
- [16] Kuo S M, Vijayan D. A secondary path modeling technique for active noise control systems[J]. IEEE Transactions on Speech and Audio Processing, 1997, 5(4): 374-377.
- [17] Zhang M, Lan H, Ser W. Cross-updated active noise control system with online secondary path modeling[J]. IEEE Transactions on speech and audio processing, 2001, 9(5): 598-602.



- [18] Akhtar M T, Abe M, Kawamata M. A new structure for feedforward active noise control systems with improved online secondary path modeling[J]. IEEE Transactions on Speech & Audio Processing, 2005, 13(5):1082-1088.
- [19] Carini A, Malatini S. Optimal Variable Step-Size NLMS Algorithms With Auxiliary Noise Power Scheduling for Feedforward Active Noise Control[J]. IEEE Transactions on Audio Speech & Language Processing, 2008, 16(8):1383-1395.
- [20] 何东红. 有源噪声混合结构控制系统的研究及应用[D]. 哈尔滨工业大学, 2019.
- [21] 沙家正, 孙广荣, 曹水轩, 吴启学. 管道有源消声器[J]. 声学学报, 1981, (3): 137-141.
- [22] 沙家正, 孙广荣, 曹水轩, 吴启学. 有源消声器消声机理的研究[J]. 声学学报, 1983, (2): 93-99.
- [23] 马大猷. 室内有源噪声控制的潜力[J]. 声学学报, 1993, 18(3): 178 - 185.
- [24] 陈克安, 马远良. 自适应有源噪声控制——原理、算法及实现[M]. 西安:西北工业大学出版社
- [25] 韩宇菲. 有源噪声控制中自适应算法研究及在局部空间降噪的应用[D]. 哈尔滨:哈尔滨工业大学, 2019.
- [26] 邓帅博. 小空间有源降噪系统研究[D]. 南京航空航天大学, 2018.
- [27] 郭磊. 基于次级路径在线辨识的宽窄带室内主动噪声控制方法[D]. 北京邮电大学, 2018.
- [28] 胡广书. 数字信号处理: 理论、算法与实现[M]. 北京:清华大学出版社.
- [29] Widrow B, Hoff M E. Adaptive switching circuits[R]. Stanford Electronics Labs, USA, 1960.
- [30] 王进军, 陈克安. 有源噪声控制的次级通路建模研究. 中国声学学会青年学术会议论文集[C]. 中国声学学会, 2005:3.
- [31] 陈健冲. 基于改进卡尔曼滤波算法的短时公交客流预测模型研究[D]. 大连海事大学, 2020.
- [32] 宫晓燕, 汤淑明. 基于非参数回归的短时交通流量预测与事件检测综合算法[J]. 中国公路学报, 2003, (1): 83-87.
- [33] Kailath T. A view of three decades of linear filtering theory[J]. IEEE Transactions on Information Theory, 1974, 20(2): 146-181.
- [34] Daum F. Nonlinear filters: Beyond the Kalman filter[J]. IEEE Aerospace and Electronic Systems Magazine, 2005, 20(8): 57-69.
- [35] Leung H, Zhu Z. Performance evaluation of EKF-based chaotic synchronization[J]. IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, 2001, 48(9): 1118-1125.
- [36] 李航. 统计学习方法[M]. 北京:清华大学出版社, 2012.
- [37] Haykin S. Neural Networks and Learning Machines[M]. Upper Saddle River: Prentice Hall, USA, 2008.
- [38] Hagan M T. 神经网络设计[M]. 北京:机械工业出版社, 2007.
- [39] Bengio Y, Simard P, Frasconi P. Learning long-term dependencies with gradient descent is difficult[J]. IEEE Transactions on Neural Networks, 1994, 5(2):157-166.

- [40] Hochreiter S, Schmidhuber J. Long short-term memory[J]. *Neural Computation*, 1997, 9(8):1735-1780.
- [41] Cho K, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.
- [42] Chung J, et al. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.
- [43] Chang C Y, Siswanto A, Ho C Y, et al. Listening in a noisy environment: Integration of active noise control in audio products[J]. *IEEE Consumer Electronics Magazine*, 2016, 5(4): 34-43.
- [44] Zhang M, Lan H, Ser W. On comparison of online secondary path modeling methods with auxiliary noise[J]. *IEEE Transactions on Speech and Audio Processing*, 2005, 13(4): 618-628.
- [45] Farhang-Boroujeny B. *Adaptive Filters: Theory and Applications*[M]. John Wiley & Sons, 2013.
- [46] 罗磊. 基于前反馈混合结构的非线性主动噪声控制方法研究[D]. 哈尔滨工业大学, 2019.
- [47] Lu L, Yin K L, de Lamare R C, et al. A survey on active noise control in the past decade - Part II: Nonlinear systems[J]. *Signal Processing*, 2020: 107929.
- [48] Burt P M S, de Morais Goulart J H. Efficient computation of bilinear approximations and Volterra models of nonlinear systems[J]. *IEEE Transactions on Signal Processing*, 2017, 66(3): 804-816
- [49] Rout N K, Das D P, Panda G. Particle swarm optimization based active noise control algorithm without secondary path identification[J]. *IEEE Transactions on Instrumentation and Measurement*, 2011, 61(2): 554-563.
- [50] Jaeger H, Haas H. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication[J]. *Science*, 2004, 304(5667): 78-80.
- [51] Nyman J, Caluwaerts K, Waegeman T, et al. System modeling for active noise control with reservoir computing[C]. *The 9th IASTED International Conference on Signal Processing, Pattern Recognition, and Applications (SPPRA-2012)*. Crete, Greece, 2012: 162-167.
- [52] Akhtar M T, Abe M, Kawamata M. A new variable step size LMS algorithm-based method for improved online secondary path modeling in active noise control systems[J]. *IEEE Transactions on Audio, Speech, and Language Processing*, 2006, 14(2): 720-726.
- [53] Pillania A, Sharma M K, Kumar N. Noise reduction using feed-forward ANC system based on online secondary path modeling[C]. *IEEE International Conference on Computing, Communication & Automation, Greater Noida, India*, 2015: 1339-1343.

## 攻读硕士学位期间发表学术论文情况

- 1 林坤鹏, 陈喆, 殷福亮. 啸叫音频检测系统 V1.0. 中国, 计算机软件著作权, 登记号: 2020SR0170575. 2020-02-25.

## 致 谢

行文至此，三年研究生的学习生涯也接近了尾声，在学生生涯的最后期间，感谢大工陪我度过这三年的时光。

首先要感谢我的研究生导师：殷福亮教授。殷老师治学严谨，工作认真，为人亲和，在学习和生活都给与了我很大的帮助，是将我带入到语音处理领域的领路人，经常告诫我们做事情要全力以赴，少玩手机。在论文撰写期间，每当我遇到问题的时候，殷老师都会针对问题给与我相应的指导，帮我把握论文的写作方向。

其次要感谢陈喆教授。陈老师为人低调，做事情一丝不苟，无论是理论知识的储备还是实际解决问题的能力，陈老师一直都是我学习的目标。陈老师不断地帮助我们改善思维方式，引导我们如何正确的思考问题、解决问题。做项目遇到困难的时候，请教陈老师都会引导我去思考，培养我解决问题的能力。由衷的感谢陈老师对我的帮助。

在3年研究生学习生活中，有幸遇到结识了许多优秀的人。感谢师兄师姐们在我初入教研室给我的帮助，感谢何莹男师兄、何琪琪师姐在我遇到困难时伸以援手。感谢实验室同学王鹤、畅瑞江、李笑、秦萌萌，在科研遇到问题时，能够一起讨论解决，相互帮助。感谢邓翔宇师弟、梁羽贤师妹和闫钰师妹，在项目中分工明确，精诚合作。感谢我的室友郭轶群、张治民在我生活中的帮助。感谢呼德师兄，王国庆师弟在科研以外一起在篮球场上挥洒汗水。

同时，感谢我的女朋友贾卓颖，在异地期间对我的付出与陪伴，无论什么情况都在我身边，给与我支持和鼓励，包容我的一切。感谢我的父母和姐姐。给与我无私的爱与温暖，支持我继续求学，让我没有了后顾之忧。

最后，感谢评审本论文的专家和教授。感谢您在百忙之中为本论文提出宝贵的意见。