Modeling and Simulation 建模与仿真, 2023, 12(1), 366-379
Published Online January 2023 in Hans. https://www.hanspub.org/journal/mos
https://doi.org/10.12677/mos.2023.121035

Hans汉斯

基于自适应滤波算法的车载ANC仿真

代　闯，方　鸣

上海理工大学，上海

收稿日期：2022年12月2日；录用日期：2023年1月23日；发布日期：2023年1月30日

摘　要

本文利用自适应滤波器的原理对车载主动降噪系统ANC中的语音降噪功能进行一次模拟仿真。仿真的语音输入通过安卓车机系统和芯科科技的DSP芯片Si47925进行输入，以尽可能还原实车语音场景。本次设计的车载ANC系统以自适应滤波器为核心，以最小均方算法LMS为主进行自适应控制，对自适应滤波常用的四种算法：最小均方算法LMS、基于最小四阶矩阵的组合算法LMS/F、归一化最小均方算法NLMS和变步长最小均方算法VSS-LMS进行迭代推导，并基于Matlab软件，实现了这四种算法的代码实现，并对带噪语音进行数字滤波仿真。经过仿真发现，四种算法均能对带噪音频产生滤波降噪作用，其中LMS的权系数迭代式最简单和方便的，定步长LMS/F算法和LMS算法下的系统稳定性和降噪性相差不多，VSS-LMS算法和NLMS算法均可提高收敛速度，其中NLMS算法能实现较高的信噪比，并且稳态精度更加高。

关键词

Matlab，自适应滤波器，LMS算法，LMS/F算法，NLMS算法，VSS-LMS算法

Automotive ANC Simulation Based on Adaptive Filter Algorithm

Chuang Dai, Ming Fang

University of Shanghai for Science and Technology, Shanghai

Received: Dec. 2nd, 2022; accepted: Jan. 23rd, 2023; published: Jan. 30th, 2023

Abstract

In this paper, we use the principle of adaptive filter to perform a simulation of the voice noise reduction function in the ANC of the automotive noise reduction system. The simulated speech input is performed through the Android car system and the DSP chip Si47925 from CoreTech to restore

文章引用: 代闯, 方鸣. 基于自适应滤波算法的车载 ANC 仿真[J]. 建模与仿真, 2023, 12(1): 366-379.
DOI: 10.12677/mos.2023.121035

the real car speech scenario as much as possible. This designed automotive ANC system is based on the adaptive filter as the center and the least mean square algorithm LMS as the main adaptive control. Four algorithms commonly used for adaptive filtering: least mean square algorithm LMS, combined algorithm LMS/F based on the minimum fourth order matrix, normalized least mean square algorithm NLMS and variable step size least mean square algorithm VSS-LMS, will be derived, realize the code implementation of these four algorithms based on Matlab software, and digital filtering simulation is performed for noisy speech. According to the result of simulation, it is found that all four algorithms can realize noise reduction effects on noisy band frequencies, among which the LMS has the simplest and most convenient iterative weight coefficients, the stability and noise reduction of the system under the fixed-step LMS/F algorithm and the LMS algorithm are similar, the VSS-LMS algorithm and the NLMS algorithm can both improve the convergence speed, among which the NLMS algorithm can achieve a higher snr and the steady-state accuracy is much higher.

Keywords

Matlab, Adaptive Filter, LMS, LMS/F, NLMS, VSS-LMS

1. 引言

目前针对汽车上的噪声控制方法有主动噪声控制(主动降噪，ANC)，和被动噪声控制(被动降噪，PNC)。被动降噪控制的方法通常采用隔声材料、优化车身结构等方式[1]，能够对大于 500 Hz 的中高频噪声进行有效滤除，在电路上常布置 RC 滤波电路或者 LC 滤波电路等方法来实现[2]，这样的控制方法需要采购有较高吸声的材料，较多的元器件来实现，有时会增加整车的整备质量以及整车成本。主动噪声控制则是基于声学中的声波相消干涉原理，通过扬声器发出与初级噪声幅值相等，但相位相反的次级声音来抵消噪声[3]，这才实现了降噪的目的[4]。相比于被动控制，主动噪声控制更适合用来控制 200 Hz 以下的低频噪音，降低车内噪音，消除通话回声[5]，提高音效质量。尤其是数字信号处理器(Digital Signal Process, DSP)的发展，让主动降噪系统的运用越发灵活。本文对采用自适应滤波器，对滤波器算法诸如最小均方算法(LMS)，以及基于 LMS 算法改进的归一化最小均方算法(NLMS)、基于最小四阶矩阵的组合算法的定步长算法(LMS/F)和变步长最小均方算法(VSS-LMS)进行了理论梳理，设计自适应滤波器的参数，并对设备上导出的音频信号在 Matlab 进行代码的实现，以对整个降噪的过程进行仿真并对仿真的结果进行分析。

2. 自适应滤波器及其算法

在主动降噪系统中有两种常见的线性滤波器，分别是有限脉冲响应滤波器(Finite Impulse Response, FIR)和无限响应滤波器(Infinite Impulse Response, IIR)。

2.1. 自适应滤波器种类

2.1.1. 有限脉冲响应滤波器(FIR)

构成自适应数字滤波器的基本部件是自适应线性组合器，FIR 滤波器与 IIR 滤波器的不同在于，该滤

波器的脉冲响应在经过 N 个采样周期后，逐渐衰减为零。设滤波器系数为 N，在时间序列为 n 时，线性组合器的 N 个输入为 $x(n),\cdots,x(n-N+1)$，其输出信号 $y(n)$ 是这些输入加权后的线性组合，即为：

$$y(n)=W^{\mathrm{T}}(n)X(n)=\sum_{i=0}^{N-1}w_i(n)x(n-i) \tag{1}$$

$$X(n)=\left[x(n),x(n-1),\cdots,x(n-N+1)\right]^{\mathrm{T}} \tag{2}$$

$$W(n)=\left[w_0(n),w_1(n),\cdots,w_{N-1}(n)\right]^{\mathrm{T}} \tag{3}$$

上式中，$X(n)$为输入矢量；$W(n)$为权系数矢量。对系统做 z 变换，得到系统的传递函数为：

$$H(z)=\frac{Y(z)}{X(z)}=\sum_{i=0}^{N-1}w_i(n)z^{-i} \tag{4}$$

由于 FIR 滤波器的$(N-1)$个极点全都位于 z 平面的原点位置，因此系统是稳定的，加之 FIR 滤波是线性系统，因此 FIR 滤波器是主动降噪系统中优先考虑的滤波器，其横向结构如图 1 所示。

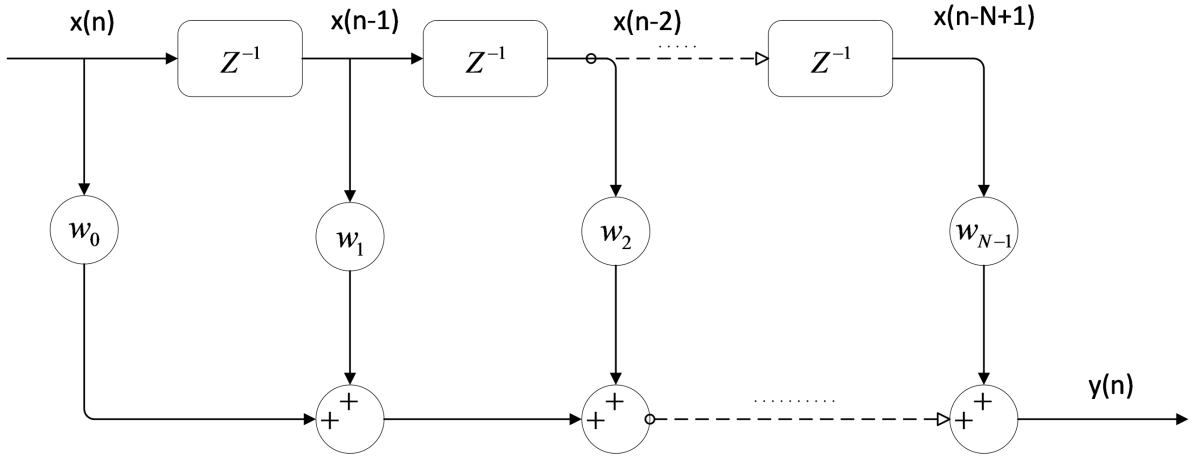

Figure 1. The transversal structure of FIR filter
图 1. FIR 滤波器横向结构

2.1.1. 无限脉冲响应滤波器(IIR)

FIR 滤波器是让信号逐渐衰减至零，但 IIR 滤波器存在反馈环节，导致系统的输出与当前时刻输入的信号有关系，这种层层迭代的递归特性导致脉冲响应的序列是无限长的。在时间序列为 n 时，滤波器的输出信号为：

$$y(n)=\sum_{i=0}^{N-1}a_i(n)y(n-i)+\sum_{i=0}^{M-1}b_i(n)x(n-i) \tag{5}$$

上式中，M 为反馈滤波器阶数；$a_i(n)$为 n 时刻下输入权系数；$b_i(n)$为 n 时刻下递归权系数。因此该系统的传递函数为：

$$H(z)=\frac{Y(z)}{X(z)}=\frac{\sum_{i=0}^{M-1}b_i(n)z^{-i}}{1-\sum_{i=0}^{N}a_i(n)z^{-i}} \tag{6}$$

由上式可知，IIR 滤波器至少存在一个极点不在 z 平面的原点处，会导致在递归的某个时刻极点会偏移出单位圆，即所谓此时滤波器处于不稳定状态。在稳定性方面，IIR 滤波器不如 FIR 滤波器。常见的 IIR 滤波器结构如图 2 所示。

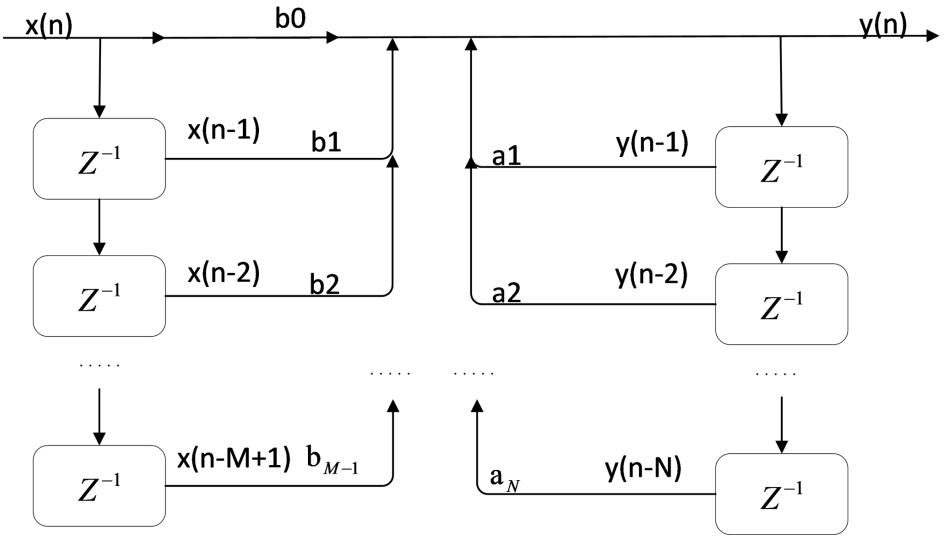

Figure 2. The structure of IIR filter
图 2. IIR 滤波器结构

2.2. 自适应算法

自适应滤波算法对抵消含噪声语音的降噪效果较好，因为使用这种方法比其他方法多用了一个参考噪声作为辅助输入，从而获得了比较全面的关于噪声的信息，因而能得到更好的降噪效果。特别是在辅助输入噪声与语音中的噪声完全相关的情况下，自适应噪声抵消法能完全排除噪声的随机性，彻底抵消语音中的噪声成分，无论在信噪比 SNR 方面还是在语音辨识度方面都能获得较大的提高。其工作原理实质上以均方误差 $E\left[e^2(n)\right]$ 或方差 $e^2(n)$ 为最小准则，对噪声 $d(n)$ 进行最优语音增强的目的。

2.2.1. 最小均方算法(Least Mean Square, LMS)

最小均方算法是最常见的自适应算法，其权系数的更新准则是保证其期望信号与实际输出信号之差的平方期望值最小，即最小均方误差最小[6]。传统的 N 阶固定步长最小均方自适应算法通常采取如下步骤：

1) 定义滤波输出：$y(n)=W^{\mathrm{T}}(n)X(n)$；

2) 定义误差信号：$e(n)=d(n)-y(n)$；

3) 用瞬时梯度得到标准时域的更新公式：$W(n+1)=W(n)+2\mu X(n)e(n)$。

上式中，n 为时间序列；$d(n)$ 为期望输出；N 为滤波器系数；μ 为步长因子，通常是个常数，为保证算法收敛，给出 μ 的取值范围为：$0<\mu<1/\lambda$，λ 是输入信号自相关矩阵的最大特征值。

2.2.2. 归一化的最小均方算法(Normalized Least Mean Square, NLMS)

在自适应滤波算法中归一化最小均方误差法是较为常用的一种算法。虽然 LMS 算法的运算比较简单并且收敛较快，但是如果输入的音频信号能量较高时，会导致迭代步长的增大，另一方面，若输入的音频信号能量较弱，又会使得迭代步长的减小，这样一来会导致最终的收敛状态不稳定，为了避免对算法收敛造成的影响，在 LMS 算法的基础上进行归一化处理[7]，即每当滤波器权系数发生变化时，可以通过算法对步长进行调整，对信号能量归一化，使得补偿迭代不会受到信号能量的影响，保证算法的鲁棒性和较快的收敛速度[8]。

NLMS 算法中，通过归一化的方式对迭代步长加以控制，滤波器权系数更新如下式：

$$W(n+1)=W(n)+\frac{\alpha}{\|x(n)\|^2}X(n)e(n) \tag{7}$$

上式中，α 为步长控制参数，在仿真阶段取值为 0.6；$\|x(n)\|^2$ 为输入信号矢量的欧式范数；算法的收敛条件为 $0<\mu<2/tr[R]$，R 为输入信号的自相关矩阵，求法为 $R=E\{X(n)X^{\mathrm{T}}(n)\}$。

2.2.3. 基于最小四阶矩阵的组合算法(LMS/F)

LMS/F 算法是基于基本 LMS 算法和最小四阶矩阵算法 LMF 结合而来的。LMF 算法又是是在 LMS 算法上提出的，旨在不改变计算复杂度的前提下，改善 LMS 在低信号强度下收敛不稳定的问题[9]。LMF 的权重系数更新公式为：

$$W(n+1)=W(n)+\mu X(n)e^3(n) \tag{8}$$

LMF 算法在权系数距离最优解较远时可以比 LMS 算法收敛更快并且稳态性能更好，不过在权系数距离最优解较近时其收敛速度和稳态性能又不及 LMS 算法，这导致计算的复杂度大大增加了。为了同时克服 LMS 和 LMF 算法的缺点提出了一种 LMS/F 的算法，兼顾了较高的稳态精度和较简易的计算度。LMS/F 的权系数更新公式为：

$$W(n+1)=W(n)+\frac{\mu e^3(n)}{e^2(n)+\varepsilon}X(n) \tag{9}$$

上式中，ε 为阈值参数，为一正值，通过选取合适的值来平衡稳态精度和收敛速度。此文中取 $\varepsilon=0.001$。

2.2.4. 变步长最小均方算法(Variable Step-Size LMS-VSS-LMS)

VSS-LMS 算法的基本思想是：在初始收敛阶段或者系统参数发生改变时，权系数与理想的未知的权系数相距较远时，为了保证有较快的收敛速度及对时变系统的跟踪速度，会选取较大的步长[10]。在算法接近收敛时，会选取较小的步长，以减少稳态误差。此算法的一大特点就在于迭代步长可以根据下式进行调整：

$$\mu(n+1)=\alpha\mu(n)+\beta e^2(n) \tag{10}$$

上式中，α,β 均为常数，并且有 $0<\alpha<1$，$\beta>0$。典型的取值有 $\alpha=0.97$，$\beta=0.00048$，且有边界条件：

$$\mu(n+1)=\begin{cases}\mu_{\max} & \mu(n+1)>\mu_{\max}\\ \mu_{\min} & \mu(n+1)<\mu_{\min}\\ \mu(n+1) & \text{other}\end{cases} \tag{11}$$

初始步长通常取 $\mu_0=\mu_{\max}$，且步长 $\mu(n)$ 是正值，此值的大小收到两个控制常数 α,β 的影响，直观上看，最小值通常比较接近正常 LMS 算法的步长值，耳较大的预测误差会导致步长增大，保证收敛速度的迅速。保证 $\mu_{\max}$ 最大有界的一个充分条件为：

$$\mu_{\max}\le\frac{2}{3tr(R)}$$

3. 基于 Matlab 的自适应滤波仿真

3.1. 仿真条件

本文借助 Matlab 和某车机安卓系统进行自适应滤波降噪的仿真，仿真流程的结构如图 3 所示。

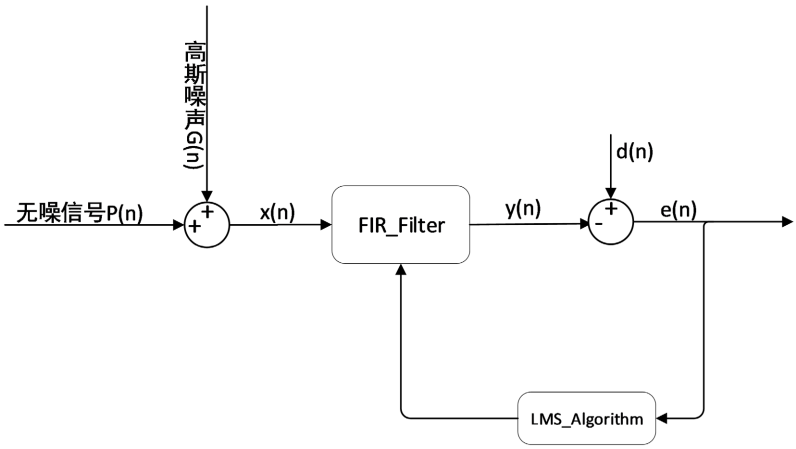

Figure 3. Diagram of the simulation structure
图 3. 仿真流程结构图

本次仿真的输入利用安卓车机系统，通过对着麦克风说话来模拟真实司机的实车操作，通过安卓的adb 命令将麦克风输入的语音信号以.wav 的形式提取出来。

基于某产品车机的实验设备如图 4 所示，所需的设施需要：

1) 芯科科技的 DSP 芯片 Si47925 所在车机开发板，用于处理音频；

2) 开发板线束电源；

3) 2 路车载数字麦克风和 2 路音响，并且集成在线束上；

4) 计算机用于使用 adb 命令将音源输出到 PC 路径；

5) 一根安卓数据线用于连接开发板和电脑 PC 端。

利用 adb 命令进行录音并保存到 PC 本地的命令如下所示：

```
tinycap /sdcard/test.wav –D 0 -d 33 -c 2 -r 48000
adb pull /sdcard/test.wav E:\
```

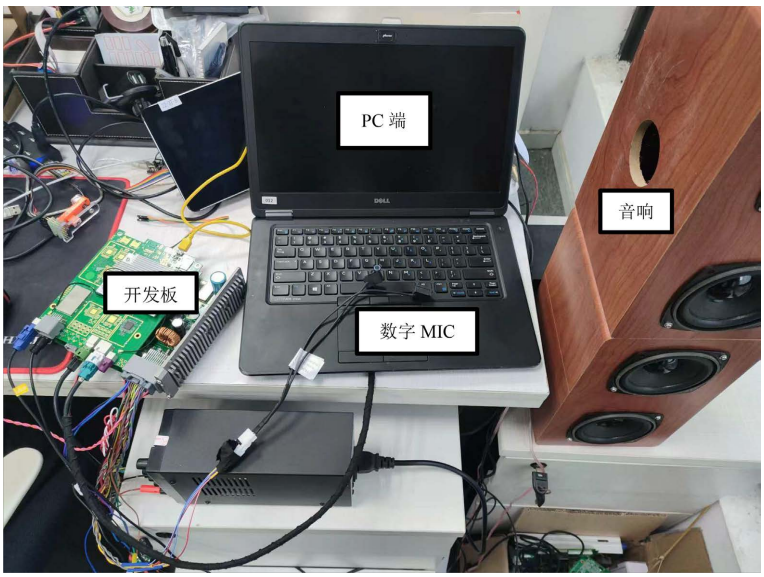

Figure 4. Diagram of the simulation equipment
图 4. 仿真设备图

其中，/sdcard 是安卓 linux 系统下的地址；-D 0 代表设备号；-d33 表示 DSP 芯片的第 33 号端口是用来输出音频的；-c 2 表示频道选择 2 通道；-r 表示采样频率。操作窗口如图 5 所示。

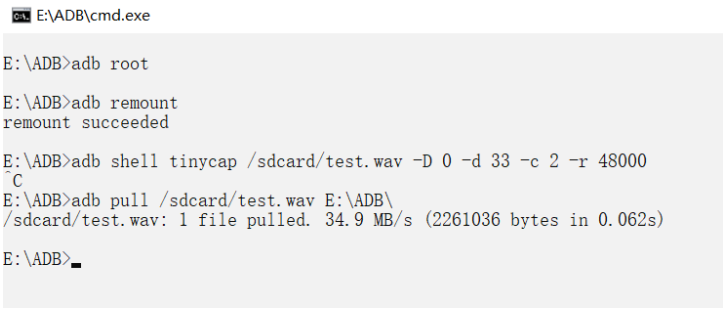

Figure 5. Diagram of exporting audio from Android
图 5. 从安卓系统导出音频

3.2. 程序设计

由于本车机并未安装在实车上，并且在较为安静的封闭环境进行语音输入，因此在 Matlab 中需要额外提取麦克输入的长度并添加高斯噪声来模拟真实场景。基于 Matlab 的语音输入和噪声添加的代码[12]如下：

```
[y_sign, fs] = audioread('E:\ADB\test.wav');          %本地语音输入
x_sign=y_sign(:,1);                                   %提取输入语音的列向量
speh_in=x_sign';                                      %转置
input=( speh_in (:,(50001:60000)));                   %任取 10000 万个数据
noise = awgn(input,20);                               %输入高斯噪声
sigwn=input+noise;                                    %合成带噪声的语音信号
```

随后进行滤波器的初始化，在程序中通过创造零函数来实现W(n)，X(n)的初始化，代码如下：

```
X=zeros(1,N);                                         %初始化输入矩阵
W_LMS=zeros(1,N);                                     %初始化权系数
E_nm=zeros(1,10000);                                  %定义误差且与 input 维数一致
```

在初始化滤波器之后，再结合之前对信号的采样，计算误差信号得到梯度，回到 LMS 算法中更新权系数方程，并循环更新直至结束，得到输出序列和误差序列。代码实现如下：

```
for k=1:n                                             %定义时间序列开始循环
X(1,2:end) = X(1,1:end-1);                            %输入从初始后开始迭代
X(1,1) = noise(k);                                    %输入的初始值更新
y= W_LMS * W_LMS ';                                   %更新输出
E_nm(k)=sigwn(k)-y;                                   %得到误差值
W_LMS = W_LMS + 2*mu*E_nm(k) .*X                      %更新权系数
end
```

在上述代码中，W_LMS 代表了基本的最小均方算法的权系数迭代公式，由于改进的 LMS 算法的本质还是在于由原先的固定步长变为可变步长，因此在代码中最直观的改变是权系数的改变。考虑到对变步长的取值，此处定义步长的最大值 $\mu_{\max}=0.0003$，步长最小值 $\mu_{\min}=0.0001$，而输入的自相关矩阵可用命令下方命令来实现：

```
R=xcorr(X);                                          %取自相关矩阵 R
```

由此，根据第二节的公式，针对不同的改进 LMS 算法可对权系数更新代码进行跟新，代码如下所示：

```
W_LMS = W_LMS + 2*mu*E_nm(k) .*X                     %LMS 算法
W_NLMS = W_NLMS + alfa*((E_nm(k)*X)/(X*X'))          %NLMS 算法
W_LMSF = W_LMF +mu*((E_nm(k)^3)/((E_nm(k)^2)+0.001)).*X   %LMS/F 算法
W_VL = W_VL + 2*mu*E_nm(k)*X                         %VSS-LMS 算法
```

4. 仿真结果及分析

将上节的代码进行整理，于 Matlab 中进行代码的完成，运行得到仿真结果。其中，初步取定步长 $\mu = 0.004$；取 NLMS 算法的控制参数 $\alpha = 0.6$，带入权系数更新公式，进行仿真迭代。得到的初始音频信号、噪声信号和带噪信号如图 6 所示。后续利用四种算法得到滤波后的音频的幅频响应图如下图 7~图 10 所示。

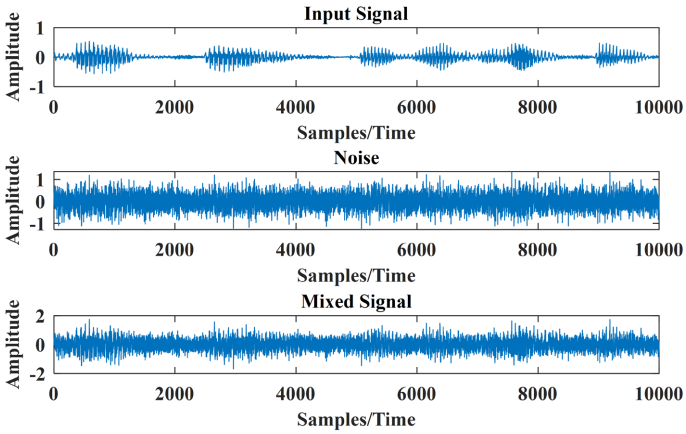

Figure 6. Diagram of the input signal、noise signal and noisy speech signal
图 6. 输入语音信号图、噪声图、带噪语音图

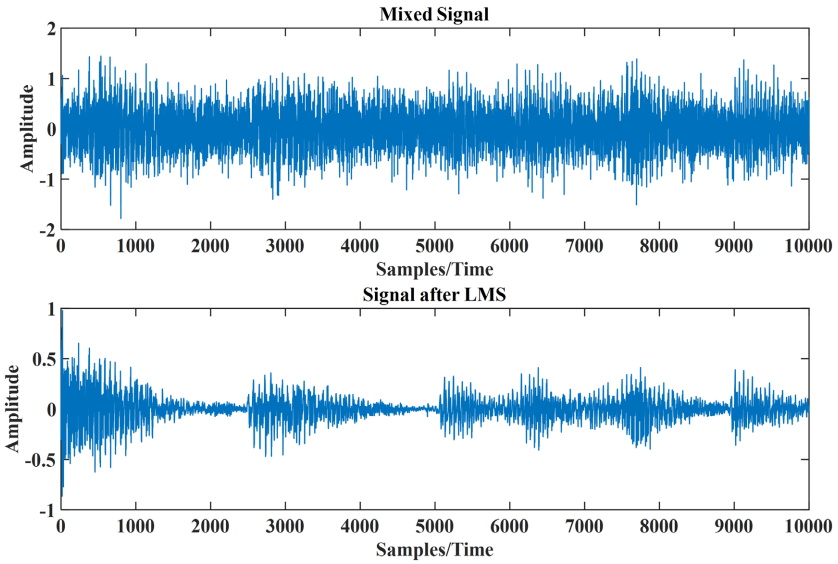

Figure 7. Diagram of the speech signal after LMS algorithm
图 7. 经过 LMS 算法后的语音信号

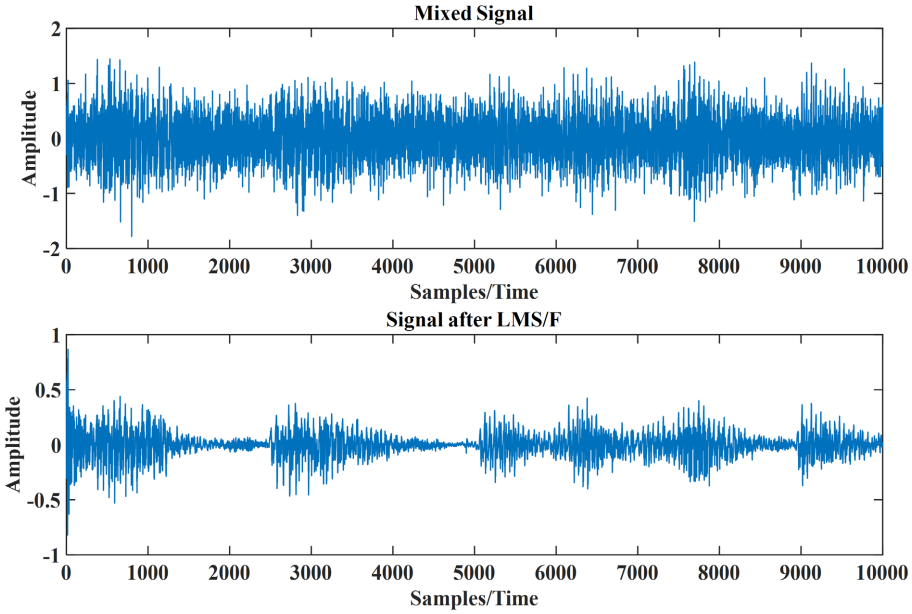

Figure 8. Diagram of the speech signal after LMS/F algorithm
图 8. 经过 LMS/F 算法后的语音信号

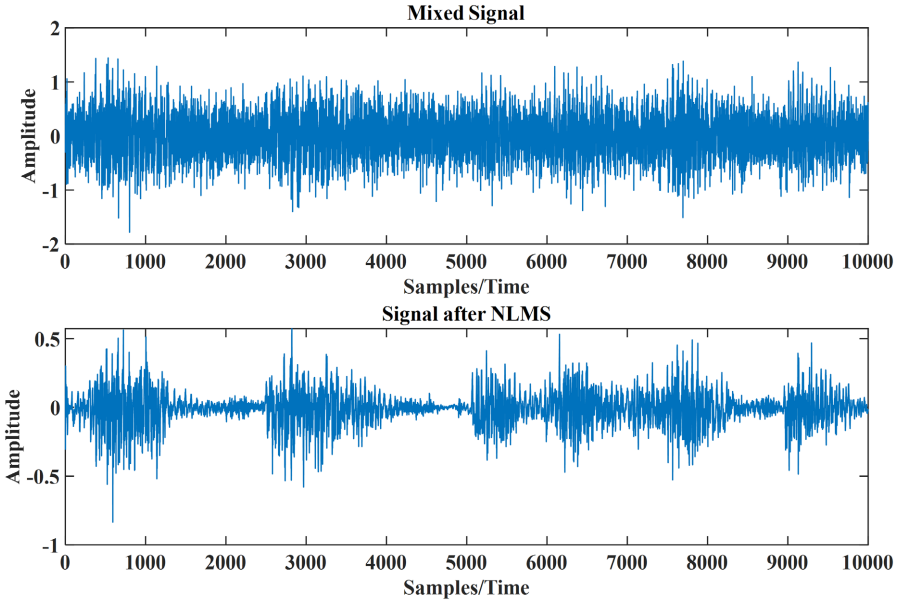

Figure 9. Diagram of the speech signal after NLMS algorithm
图 9. 经过 NLMS 算法后的语音信号

4.1. 总体算法性能分析

将各个权系数更新公式中的参数带入代码并运行后，在 20 dB 的噪声信噪比情况下，取迭代次数为 1000 次，得到图 11 所示的结果。其中黑色曲线是 NLMS 算法，红色曲线是 VSS-LMS 算法，黄色曲线是 LMS/F 算法，蓝色曲线是 LMS 算法。在结果中，四种算法都使得系统达到了稳态。其中，就迭代速度而言，NLMS 算法的迭代速度是最快的，但在迭代初期会有较高的突变。VSS-LMS 算法的速度其次，基于组合定步长的 LMS/F 算法是最慢的，但与传统的 LMS 算法的迭代速度相仿，并无较大区别，可见

局部放大图 12。四种算法在 500 次左右均逐步达到稳定，由于稳态误差值无法通过图直观描述，可见后文的表 1~表 3。

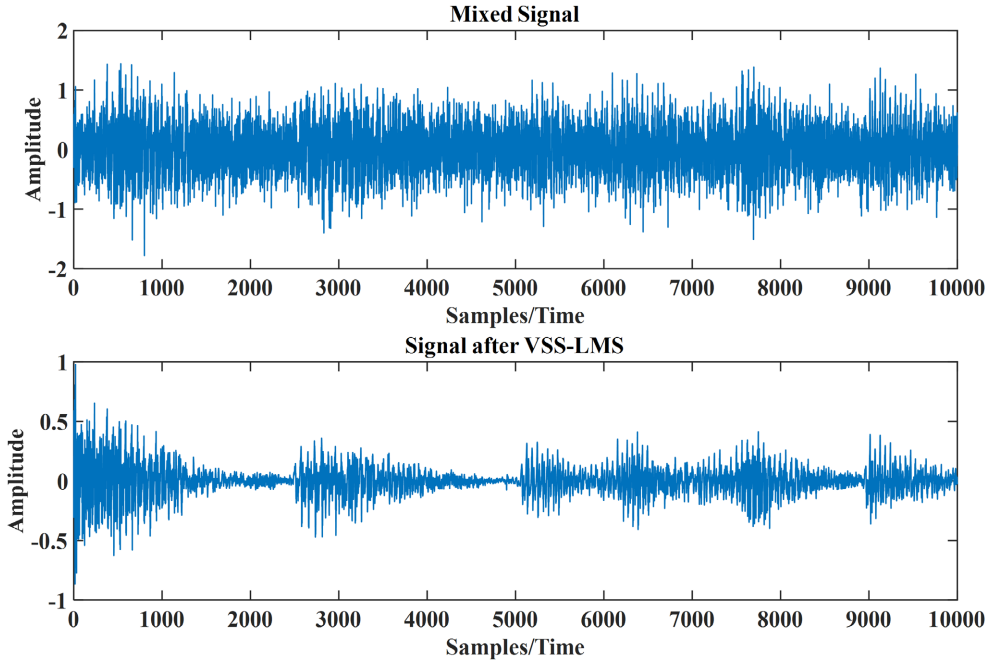

Figure 10. Diagram of the speech signal after VSS-LMS algorithm
图 10. 经过 VSS-LMS 算法后的语音信号

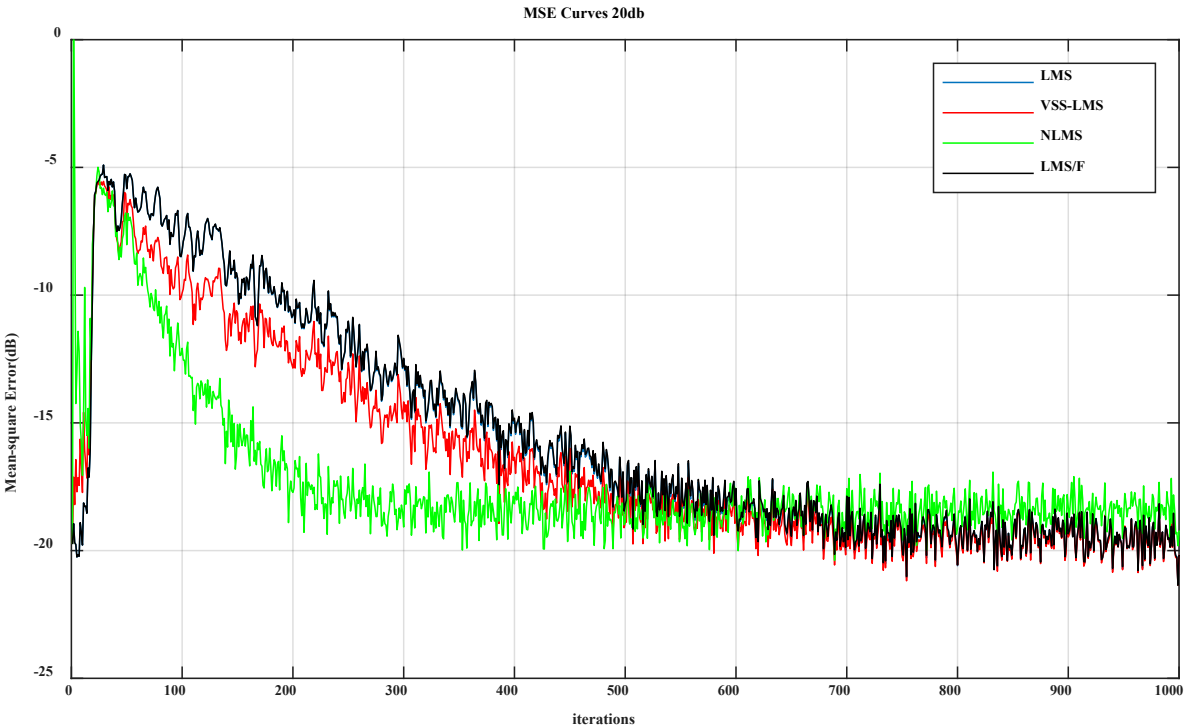

Figure 11. Performance curves of four algorithms with 20 dB noise
图 11. 20 dB 噪声下的四种算法的性能曲线

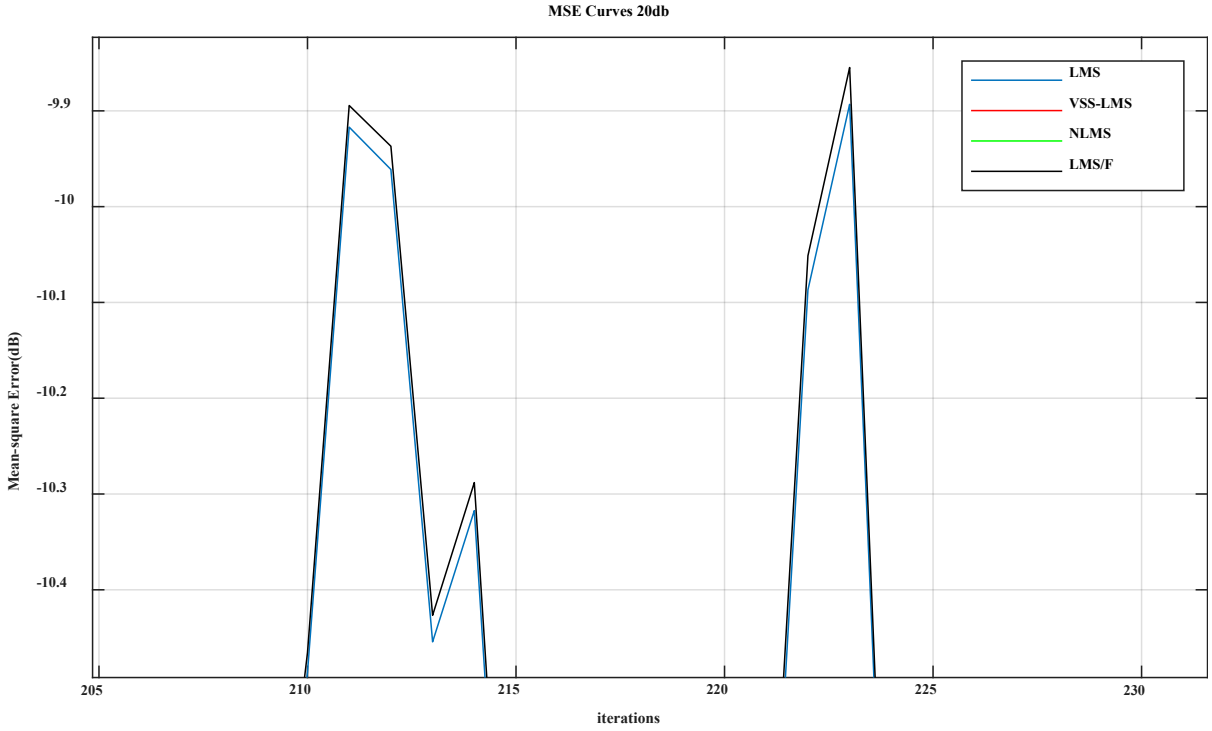

Figure 12. Localzoom between LMS and LMS/F algorithms
图 12. LMS 和 LMS/F 算法的局部放大比较

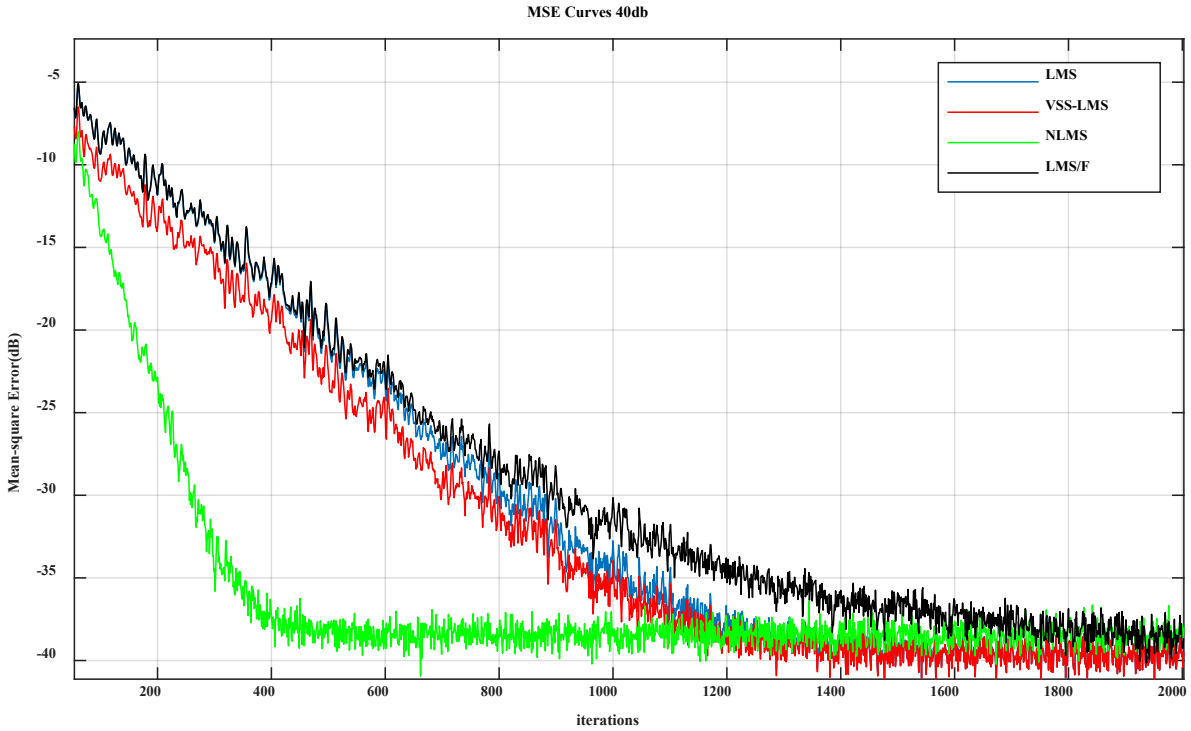

Figure 13. Performance curves of four algorithms with 40 dB noise
图 13. 40 dB 噪声下的四种算法的性能曲线

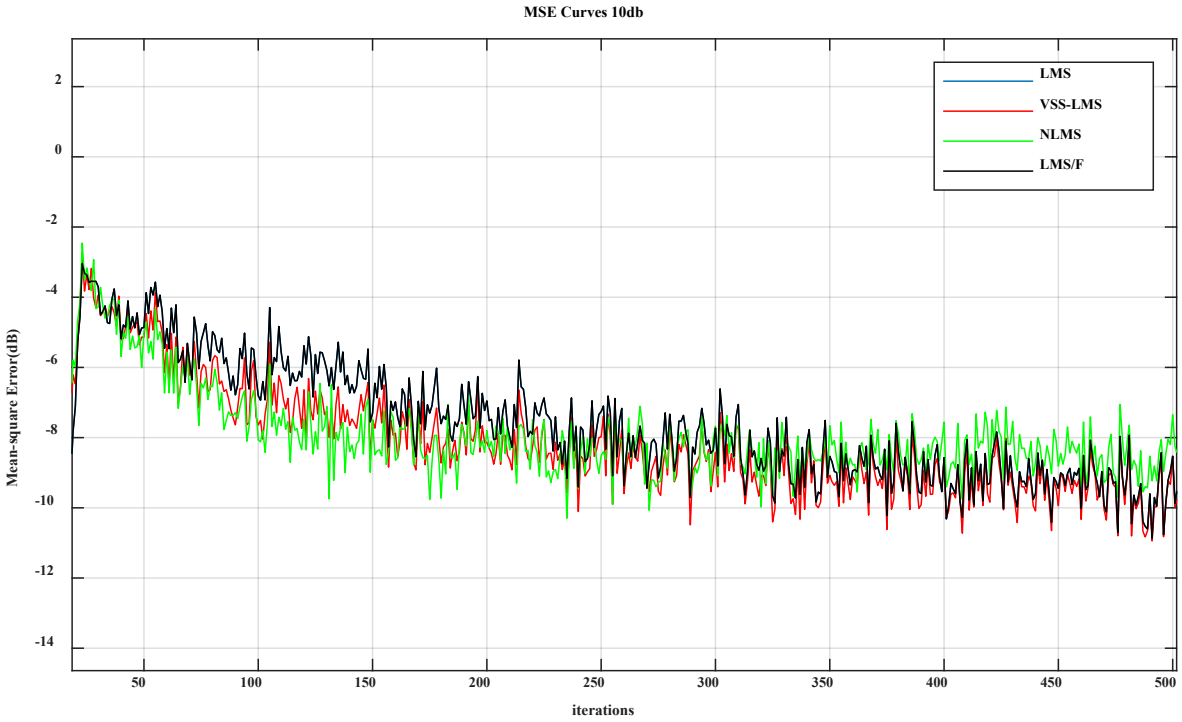

Figure 14. Performance curves of four algorithms with 10 dB noise
图 14. 10 dB 噪声下的四种算法的性能曲线

4.2. 不同噪音信噪比下各算法性能分析

分别对信噪比为 40 db、20 db 和 10 db 下四种算法的表现情况，20 db 的情况可见图 11，剩余两种情况的仿真结果如下图 13 和图 14 所示。

由图 13 可见，在高信噪比噪声的情况下，定步长 NLMS 算法的收敛速度相互对而言更加迅速，在 500 次即达到收敛，但初期的突变现象并无改善，LMS/F 算法收敛速度最慢，并且相比于 20 db 的情况下，其收敛速度进一步放慢并且在 1800 次左右达到稳态。而 VSS-LMS 算法比 LMS 算法收敛稍快，但两者均在 1200 次左右收敛。

由图 14 可见，在低信噪比噪声的情况下，四种算法的收敛速度几乎相仿，均在 250 次作用达到收敛，并且在相同的步长值下，LMS 算法与 LMS/F 算法并无差别。

4.3. 降噪效果分析

在评价语音降噪效果的指标中，常常使用的是降噪后语音的信噪比 SNR 和均方误差 MSE 两种性能指标[6]。其中，SNR 和 MSE 的计算公式可按照下式计算：

$$\mathrm{SNR}=10\lg\left(\frac{\sum_i^N x^2(n)}{\sum_i^N\left(x(n)-y(n)\right)^2}\right) \tag{12}$$

$$\mathrm{MSE}=\frac{1}{N}\sum_{k=1}^N\left(x(n)-y(n)\right)^2 \tag{13}$$

改变高斯噪音的信噪比，将基于公式(10)和(11)计算得到的数据记录于表 1~表 3 中来分析四种算法在不同噪音强度下的降噪表现，表中的值均为多次仿真计算结果的平均值。

Table 1. Comparison of noise reduction effect between 4 algorithms with 10 dB noise
表 1. 在加入 10dB 的噪声下四种算法间的降噪效果对比

算法	SNR (dB)	MSE
LMS	21.8753	0.1082
LMS/F	22.4687	0.1089
NLMS	23.6919	0.1443
VSSLMS	22.4638	0.1088

Table 2. Comparison of noise reduction effect between 4 algorithms with 20 dB noise
表 2. 在加入 20dB 的噪声下四种算法间的降噪效果对比

算法	SNR (dB)	MSE
LMS	11.9143	0.0127
LMS/F	13.0929	0.0115
NLMS	14.1238	0.0145
VSSLMS	13.0197	0.0115

Table 3. Comparison of noise reduction effect between 4 algorithms with 40 dB noise
表 3. 在加入 40dB 的噪声下四种算法间的降噪效果对比

算法	SNR (dB)	MSE
LMS	6.3704	0.0798
LMS/F	6.3946	0.0803
NLMS	8.2235	0.0349
VSSLMS	6.9531	0.0588

通过上表可知，NLMS 算法无论在较高信噪比和较低信噪比噪声环境下，都能保持较高的稳态精度和信号的保真度，是一种比较理想的算法，VSS-LMS 算法在高信噪比噪声环境中要比传统的 LMS 算法精度高，而定步长 LMS/F 算法跟 LMS 算法在本次仿真中几乎相差不多，但是无论是在强噪声环境还是弱噪声环境下，LMS/F 算法都要比 LMS 算法有微强的降噪效果。

5. 总结与展望

本文将 LMS 算法、LMS/F 算法、NLMS 算法和 VSS-LMS 算法进行了代码的实现，软件上结合数字滤波器 FIR 原理和自适应算法原理，在硬件上利用 DSP 芯片 Si47925 和安卓系统进行了对实车语音的模拟，将语音加以噪声后利用算法进行滤波降噪，并且由仿真结果可以知道，四种算法对降噪都有一定的优化。四种算法中，LMS 算法的计算过程和代码实现最为简单，VSS-LMS 算法最为复杂，但是收敛速度有所提升，定步长 LMS/F 算法在本次仿真中与 LMS 效果相差不多，NLMS 在本次仿真中体现出了快速收敛，高精度，高保真的特点。

目前的自适应算法中，NLMS 算法、VSS-LMS 算法都有很多的优化的空间并且已被大力开发，都形成了较为成熟的改进算法，本文只是对这些基础算法进行了降噪仿真，来模拟车机系统环境下的降噪算法，不过基于基础算法的改进算法在本文中没有体现，可以在后续的工作中继续完成，运用更加先进的算法并实现车机系统下的麦克风语音降噪。

参考文献

[1] 吴开明. 基于频域自适应算法的车内主动噪声控制方法研究[D]: [硕士学位论文]. 长春: 吉林大学, 2021. https://doi.org/10.27162/d.cnki.gjlin.2021.005690

[2] 竺增宝. 车载多媒体音频系统的设计与实现[D]: [硕士学位论文]. 上海: 上海交通大学, 2015.

[3] 吴世杰. 车载多媒体语音降噪识别系统设计[J]. 机电工程技术, 2018, 47(9): 47-50.

[4] 孟凡姿, 张瑜, 王铮, 关显卓, 张超, 张东上. 基于 ANC 技术的车载主动降噪系统设计[J]. 工业控制计算机, 2021, 34(4): 148-149.

[5] 张帆. 一种基于双麦克架构的机载噪声抑制方法[J]. 信息技术与信息化, 2018(5): 64-67.

[6] 刘庆强, 郑长敏, 何红凯, 吴力. 一种基于小波阈值的变步长 LMS 语音降噪算法[J]. 吉林大学学报(理学版), 2022, 60(4): 943-949. https://doi.org/10.13413/j.cnki.jdxblxb.2021227

[7] 韩宇宸, 余志勇, 曹俊杰, 韩佳, 刘荣. 箕舌线变步长 LMS/F 算法数字域自干扰消除研究[J/OL]. 微波学报: 1-5. http://kns.cnki.net/kcms/detail/32.1493.TN.20220812.1302.004.html, 2022-12-02.

[8] 张健. 基于 NLMS 算法回波消除的研究与实现[D]: [硕士学位论文]. 哈尔滨: 哈尔滨工业大学, 2015.

[9] 孙道宗, 占旭锐, 刘思菁, 薛秀云, 谢家兴, 李震, 宋淑然. 基于 NLMS 算法的风送式喷雾机窄带有源降噪仿真研究[J]. 华南农业大学学报, 2021, 42(6): 71-78.

[10] 张继荣, 张天. 一种改进的变步长 LMS 自适应滤波算法[J]. 西安邮电大学学报, 2021, 26(1): 7-12. https://doi.org/10.13682/j.issn.2095-6533.2021.01.002

[11] 程颖菲, 于伟健, 王建明. 基于 DSP 的回声消除系统设计与实现[J]. 电子产品世界, 2018, 25(12): 59-62+51.

[12] 申俊杰. MATLAB 语音信号处理[J]. 数字通信世界, 2018(6): 87.