Indian Institute of Information Technology, Sricity

COURSE PROJECT
DIGITAL SIGNAL PROCESSING

ADAPTIVE LINE ENHANCEMENT COMPARATIVE ANALYSIS AND AN APPLICATION

Team - 06
Dittakavi V Adithya - S20180020208
Bezawada Vishnu Vamsi - S20180020202
Manohar Sai Alapati - S20180020220

Submission Date: 12 December 2020

Abstract

One of the most important and complex techniques on elements of signal processing has been the noise suppression. Noise is a very basic, of whose frequency response cannot be known all the time. Considering the motivation of trying to suppress the noise component, a focus on filters and ADAPTIVENESS has been put to extract a narrow band signal masked with wide band noise. This work includes a comparative analysis on different FIR adaptive filter algorithms, with addition to an implementation on test audio signal. The results quite provided the difference in between LMS, RLS, NLMS and APAs.

Nomenclature and Definitions

FIR: Finite Impulse Response
dB: Decibels (Scale to represent the amplitude of frequency domain signals)
LMS: Least Mean Square
NLMS: Normalized Least Mean Square
RLS: Recursive Least Square
APA: Affine Projection Algorithm

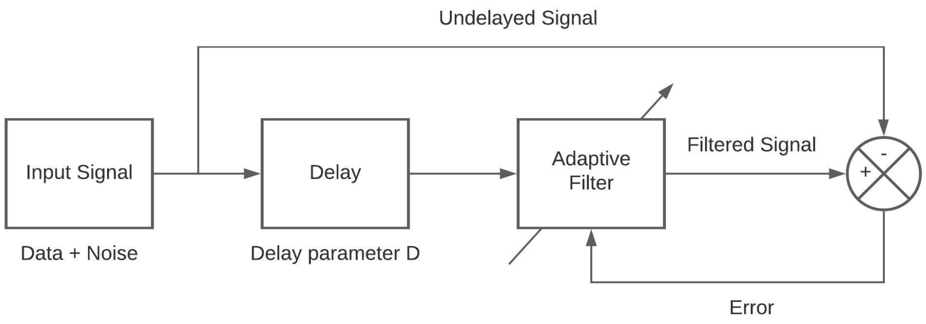
Introduction

Noise is generally present in all signals. It is generated automatically by the electronic systems in transmitters, surrounding EM waves, nearby stations or by weather conditions. It is also sometimes added to the signal in order to hide the actual information from capturing it. With this noise can be categorised into wanted and unwanted noise.

Removing unwanted noise from the signals is a very important aspect in signal processing technology. It is a major application for processing signals from one place to another. The general method of suppressing noise from corrupted signals is to pass it through a filter. If the prior information of both signal and noise is available, a fixed filter can be used. In most cases, the noise information is unknown. In this case, adaptive filters are used which have the ability to tune themselves automatically based on their own parameters.

Adaptive Line Enhancement

Adaptive Line Enhancer is the complete system including adaptive filters, focusing on the objective of cancelling/suppressing the wideband noise component which masks the data component. The data component is any narrowband signal which can be of speech data, system data or binary data. In the first part of enhancement, the enhancer tries to eliminate the noise by removing the correlativity or by observing the uncorrelativity between the original undelayed signal and delayed signal. The observation brings out a result, filtered signal, generated by an adaptive FIR filter, whose weights are updated by finding the error component. The delay, update iterations, and learning rate play an important role in tuning the filter to output its best.



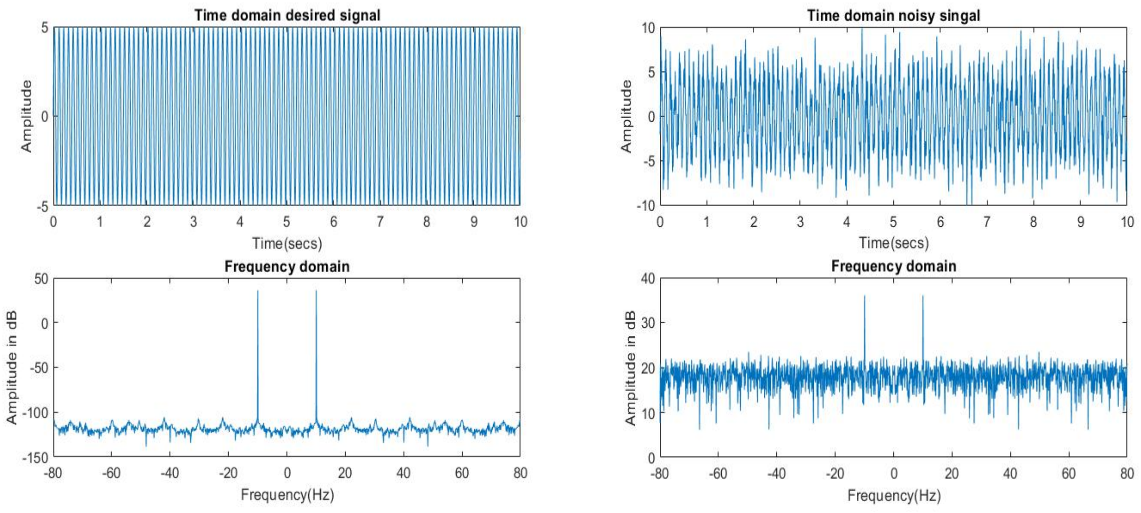
Block Diagram - Adaptive Line Enhancer

A. Data generation

A noise signal w(n) is generated by white Gaussian of a power 4 times to the input signal. The desired signal is the sinusoidal signal s(n).

$s(n) = A * sin(2 * pi * f * t)$

$w(n) = whiteGaussianNoise, SNR : -6dB$



B. Delay module

This is the one module which helps in removing the correlated noise samples. The delay D, when is added to original signal and then compared to undelayed signal, the uncorrelated samples can be eliminated which are actually noise components. The comparison is made by Adaptive Filter and Summer modules.

C. Adaptive Filter

The FIR filters has only zeroes and free of stability problems. Adaptive FIR filters are more stable compared to IIR filters.The stability of the filter depends critically on the algorithm for adjusting its coefficients.The combined signal is sent to an adaptive filter with delay. The error signal is calculated by the difference between filtered signal and undelayed signal.The error signal is sent to an adaptive filter for reconfiguring the weights. This process continues until a specified number of iterations.

LMS Algorithm

LMS is an algorithm which works by the objective of lowering the mean squared error between the true and the noisy signal. So, we just need to set some hyper parameters that suit the task, and the correct determination of the coefficients is taken care of by the algorithm. The minimizing objective of the LMS algorithm is given by,

$$\varepsilon = \sum_{n=0}^{M} e^2(n) = \sum_{n=0}^{M} [d(n) - \sum_{k=0}^{N-1} h(k)x(n-k)]^2$$

where d(n) is the true signal, h(k) are the filter coefficients which are to be determined, N is the order of the filter, M is the samples taken, x(n) is the noisy version of the true signal.

The algorithm tries to reduce the error e(n) as much as possible, and Gradient based optimizer is used as it is well suited for such ad-hoc functions. The gradients are calculated by differentiation with respect to the filter coefficients as given by the equation,

$$\frac{\partial \varepsilon}{\partial h(m)} = 0, 0 \leq m \leq N-1$$

From this we will have an update rule to update the filter coefficients in the right path which is given by,

$$h_n(k) = h_{n-1}(k) + \Delta * e(n) * x(n-k), 0 \leq k \leq N-1$$

where delta is the learning rate (hyper parameter) that controls the step size of this gradient based optimization.

A large value of delta leads to large step size adjustments and thus to rapid convergence, while a small value of results in slower convergence. However, if it is made too large the algorithm becomes unstable. Better results are often obtained when delta is small and trained for more iterations.

NLMS Algorithm

Since the LMS algorithm is direct, simple and powerful it is obvious that it has different versions. One of them is the Normalized Least Mean Squares Algorithm (NLMS).
The main drawback of the LMS algorithm is that it is sensitive to the scaling of its input. This makes it very hard (if not impossible) to choose a learning rate that guarantees stability of the algorithm. The Normalised least mean squares filter (NLMS) solves this problem by normalising with the power of the input.

$$h(n+1) = h(n) + \frac{\mu e^*(n)x(n)}{x^H(n)x(n)}$$

Visual results show that results generated by NLMS are more stable as compared to LMS.

RLS Algorithm

RLS is defined as Recursive Least Square Algorithm, which is very similar to LMS algorithm except for the involvement of forgetting factor.

Fast Affine Projection Algorithm

This being a test algorithm, actually projects the delayed signal into other dimensions. By this projection, model fails to understand the difference between correlated and uncorrelated elements, thus failing the system.

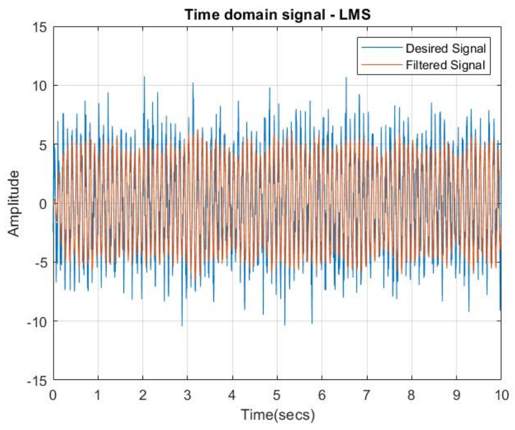
Weiner Filter

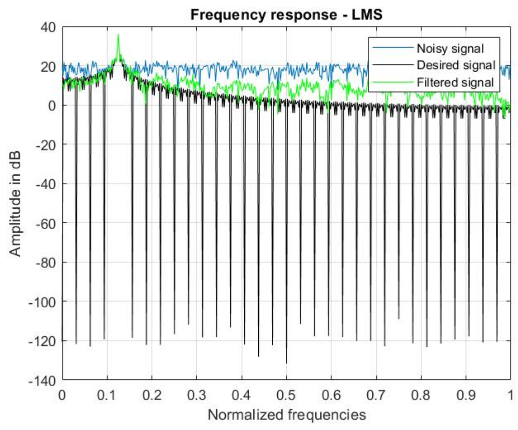
This is an other test algorithm, which uses circular correlation to understand terms. Even this model fails the system in suppressing noise.

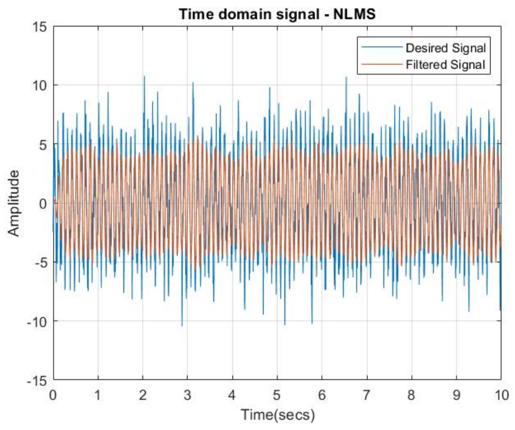
D. Observation and Comparision

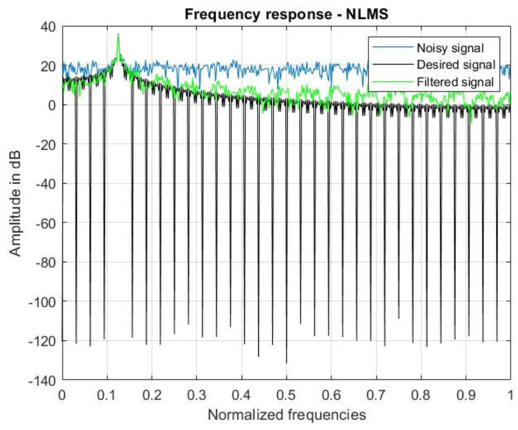
Algorithm	**Epoch**	**Time for analysis**	**Suppression**	**Comments**	**Selection**
LMS	100	0.37secs	✓	Basic Algorithm provided and focused on in project	✓
RLS	-	0.01secs	✓	Suppresses noise - More efficient than LMS	x
NLMS	100	0.35secs	✓	Suppresses Noise - More efficient than LMS and RLS	✓
Fast AP	-	0.02secs	-	Fails to suppress	x
Weiner Filter	-	0.001secs	-	Fails to suppress	x

E. Results









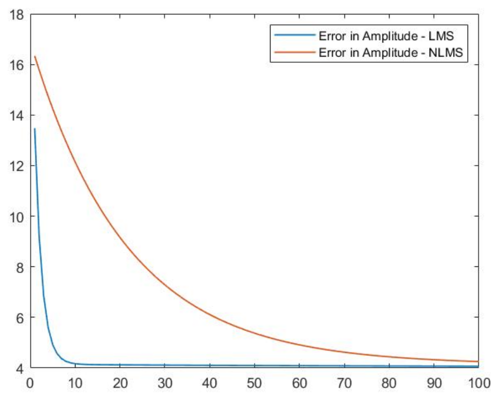


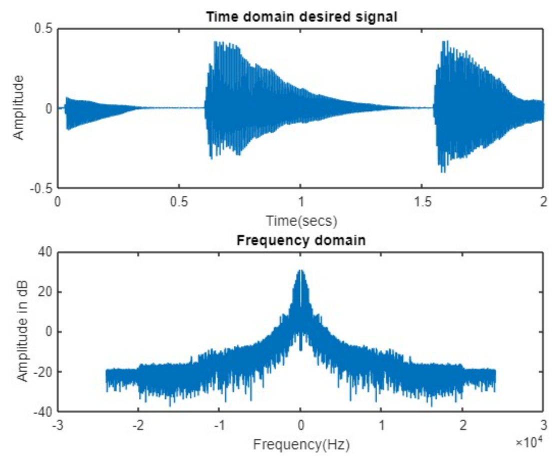
Fig. Squared Error for each epoch

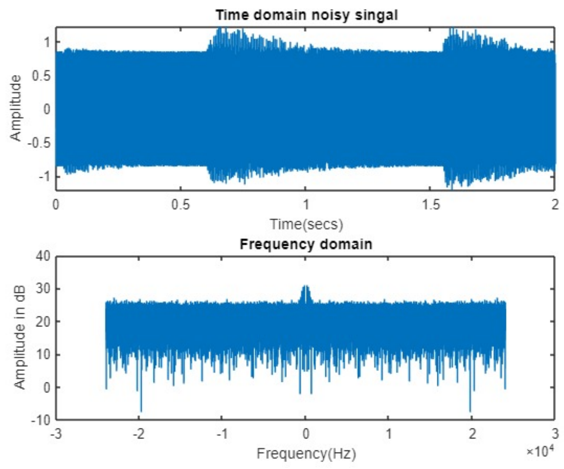
Into the Application

A. Data

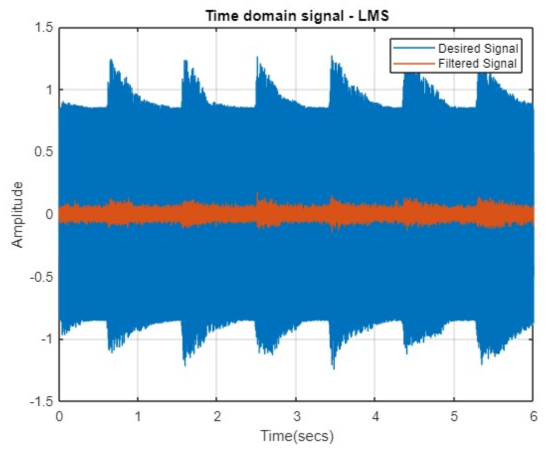
Small audio sample is extracted which plays "tuba" (49-349 Hz) as our true narrow band signal. The length of the sample is 14 seconds and the sampling rate is 48000. We add noise with amplitude half as that of the maximum amplitude of our true audio signal.

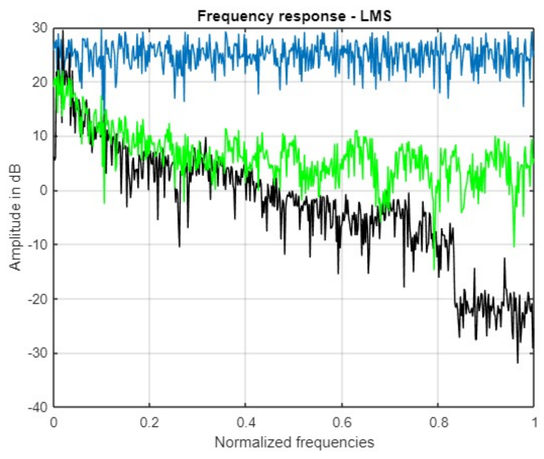
Noise signal here added is Non white Gaussian to check the efficiency of NLMS algorithm compared to LMS.

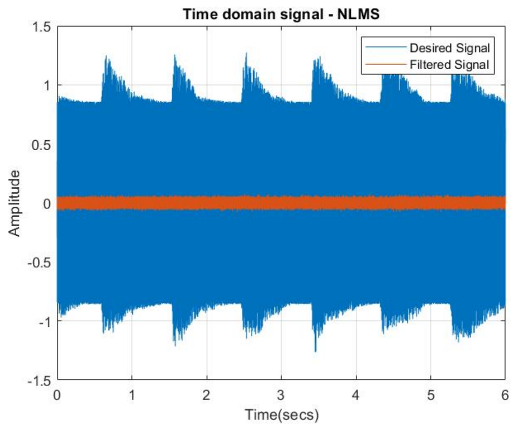


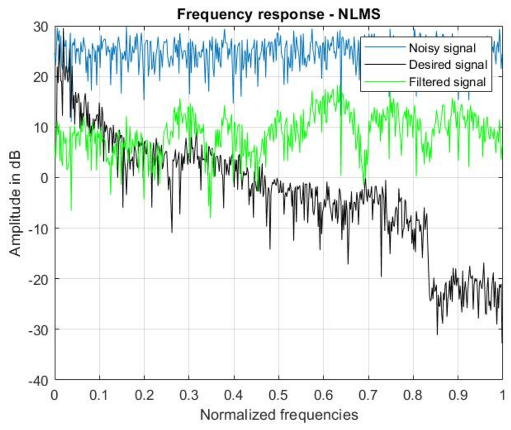


B. Results









Conclusion

We can see the power of adaptive filters in reduction of noise.There are many other applications of these adaptive filters like channel equalization, system modeling and system identification. Also, in almost all these applications, LMS is a primary algorithm. There are variants of LMS like N-LMS,etc which have their own pros and cons. The main critical thing in this Algorithm is that we have to tune the hyper parameters like learning rate, filter order carefully. Lower learning rates generally work well but take lots of time and compute to optimize. It is always best to try and tweak some parameters and see the effect. The time domain signal should be amplified at the next stage before completing the whole denoising.

Contribution

AdithyaDV - LMS algorithm, Affine Projection
Vamsi BV - RLS algorithm, Weiner Filter
Manohar Sai A - NLMS algorithm, Audio preprocessing

Repository

https://github.com/AdithyaDV/DSP-Team06-Project.git

References

1) Audio sample: https://www.youtube.com/watch?v=PzH4XAv9ZCQ)
2) https://en.wikipedia.org/wiki/Least_mean_squares_filter#:~:text=Least%20mean%20squares%20(LMS)%20algorithms,desired%20and%20the%20actual%20signal).
3) A. Chaturvedi, K. Raj and A. Kumar, "A comparative analysis of LMS and NLMS algorithms for adaptive filtration of compressed ECG signal," 2012 2nd International Conference on Power, Control and Embedded Systems, Allahabad, 2012, pp. 1-6, doi: 10.1109/ICPCES.2012.6508051.