The Applications and Simulation of Adaptive Filter in Speech Enhancement

Soni Changlani, Dr. M.K. Gupta

Assistant Professor Lakshmi Narain College of Technology, Bhopal
Department of Electronics and Communication Engineering,
e-mail: sonibpl@rediffmail.com

Professor Maulana Azad National Institute of Technology, Bhopal
Department of Electronics and Communication Engineering,
e-mail: gupta4@yahoo.com

Abstract

In many applications of noise cancellation the changes in signal characteristics could be quite fast. This requires the utilization of adaptive algorithms, which converge rapidly. Least mean square (LMS) and Normalized LMS (NLMS) adaptive filters have been used in a wide range of signal processing applications because of its simplicity in computation and implementation. The Recursive Least Squares (RLS) algorithm has established itself as the ultimate” adaptive filtering algorithm in the sense that it is the adaptive filter exhibiting the best convergence behavior. Unfortunately, practical implementations of this algorithm are often associated with high computational complexity and/or poor numerical properties. In this paper we have performed and compared these classical adaptive filters for attenuating noise in speech signals. In each algorithm, the optimum orders of filter of adaptive algorithms have also been found through experiments.

The objective of noise cancellation is to produce the estimate of the noise signal and to subtract it from the noisy signal and hence to obtain noise free signal. We simulate the adaptive noise canceller with MATLAB.

Keywords- Noise canceller; adaptive algorithm; LMS; NLMS RLS;
1 Introduction

Liner filtering is required in a variety of applications. A filter will be optimal only if it is designed with some knowledge about input data. If this information is not known, then adaptive filters [1] are used. An adaptive filter is essentially a digital filter with self-adjusting characteristics. It adapts, automatically, to changes in its input signals. In adaptive filtering, the adjustable filter parameters are to be optimized. The criteria arrived at for optimization should consider the filter performance and realisability. Adaptive algorithms are used to adjust the coefficient of the digital filter. Common algorithms that have found widespread applications are the least mean square (LMS), the recursive least square (RLS), and the kalman filter [3] algorithms. RLS algorithm is one of the most popular adaptive algorithm. As compared to the least-mean-square (LMS) algorithm, the RLS offers a superior convergence rate, especially for highly correlated input signals. The price to pay for this is an increase in the computational complexity.

2 RLS Adaptive Algorithm

The RLS (recursive least squares) algorithm is another algorithm for determining the coefficients of an adaptive filter. In contrast to the LMS algorithm, the RLS algorithm uses information from all past input samples (and not only from the current tap-input samples) to estimate the (inverse of the) autocorrelation matrix of the input vector. The RLS algorithm is a recursive form of the Least Squares (LS) algorithm. It is recursive because the coefficients at time n are found by updating the coefficients at time n-1 using the new input data. Whereas the LS algorithm is a block-update algorithm where the coefficients are computed from scratch at each sample time. To decrease the influence of input samples from the far past, a weighting factor for the influence of each sample is used. This weighting factor is introduced in the cost function.

\[ \zeta(n) = \sum_{i=1}^{n} \lambda^{n-i} |e(i)|^2 \]  

where \( e(i) \) is the difference between the desired response and output produced by a filter. \( \lambda \) is special form of weighting factor, known as forgetting factor, the performance of RLS-type algorithms in terms of convergence rate, tracking, misadjustment, and stability depends on the forgetting factor. The classical RLS algorithm uses a constant forgetting factor and needs to compromise between the previous performance criteria. When the forgetting factor is very close to one, the algorithm achieves low misadjustment and good stability, but its tracking capabilities are reduced. A smaller value of the forgetting factor improves the tracking but increases the misadjustment, and it could affect the stability of the
algorithm. The use of the forgetting factor is intended to ensure that data in the distant past are “forgotten” in order to afford the possibility of following the statistical variations of the observable data when the filter operates in a nonstationary environment. \( \lambda \) is a positive constant close to, but less than, 1. The rate of convergence of the RLS also depends on \( \lambda \).

### 3 Adaptive Noise Cancellation

The basic idea of an adaptive noise cancellation algorithm is to pass the noisy signal through a filter that tends to suppress the noise while leaving the signal unchanged. This process is an adaptive process, which means it can not require a priori knowledge of signal or noise characteristics. The technique adaptively adjusts a set of filter coefficients so as to remove the noise from the noisy signal. To realize the adaptive noise cancellation, we use two inputs and an adaptive filter. One input is the signal corrupted by noise (Primary Input, which can be expressed as \( (s(n) + N_o(n)) \)), the other input contains noise related in some way to that in the main input but does not contain anything related to the signal (Noise Reference Input expressed as \( N_1(n) \)). The noise reference input passes through the adaptive filter and output \( y(n) \) is produced as close a replica as possible of \( N_o(n) \). The filter readjusts itself continuously to minimize the error between \( N_o(n) \) and \( y(n) \) during this process. Then the output \( y(n) \) is subtracted from the primary input to produce the system output \( e = S + N_o - y \), which is the denoised signal. Assume that \( S, N_o, N_1 \) and \( y \) are statistically stationary and have zero means. Suppose that \( S \) is uncorrelated with \( N_o \) and \( N_1 \), but \( N_1 \) is correlated with \( N_o \). We can get the following equation of expectations:

\[
E[e^2] = E[s^2] + E[(N_o - y)^2]
\] (2)

When the filter is adjusted so that \( E[e^2] \) is minimized, \( E[(N_o - y)^2] \) is also minimized. So the system output can serve as the error signal for the adaptive filter.

![Fig. 1: Adaptive noise canceller.](attachment:image.png)
4 Simulation And Results:

This section describes parts of the sample MATLAB program and results. The Program can be divided into four parts: First employ the randn(N,1) function within MATLAB to generate \( x(k) \) which will have unity power and zero mean, where \( N \) is number of system point. The information bearing signal is a sine wave of 0.055 cycles/sample the noise picked up by the secondary microphone is the input for the RLS adaptive filter. The noise that corrupts the sine wave is a low pass filtered version of (correlated to) this noise. The sum of the filtered noise and the information bearing signal is the desired signal for the adaptive filter. Set and initialize RLS adaptive filter parameters and values Filter order \( N=32 \) Exponential weighting factor \( \lambda=1 \);

![The information bearing signal](image)

Fig 2: The information bearing signal
Fig 3: Noise picked up by the secondary microphone

Desired input to the Adaptive Filter = Signal + Filtered Noise

Fig 4: Input to the Adaptive filter


5 Conclusion

Simulation results for filter length N=32, and forgetting factor=1 shows that if Speech enhancement is done by using RLS algorithm and Filter order should be taken high, in this the RLS algorithm converges quickly, but its complexity grows with the square of the number of weights, roughly speaking. This algorithm can also be unstable when the number of weights is large.
References