**HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY**

SCHOOL OF ELECTRICAL & ELECTRONIC ENGINEERING



FINAL REPORT

SUBJECT: ADVANCED SIGNAL PROCESSING

TOPIC: ADAPTIVE FILTER FOR NOISE CANCELLATION APPLICATION

|  |  |
| --- | --- |
| Teacher: | TS. Đặng Quang Hiếu |
| Students: | Huỳnh Đức Minh – 20212456M  Trần Đức Dũng – 20212452M  Trần Thái Sơn – 20212460M |

Hà Nội, 02/2023

Contents

[I. Introduction 4](#_Toc127139346)

[II. Adaptive filter overview 5](#_Toc127139347)

[1. Weiner linear filter 7](#_Toc127139348)

[1.1. Linear Optimum Filtering: Statement of the Problem 7](#_Toc127139349)

[1.2. Principle of Orthogonality 9](#_Toc127139350)

[1.3. Minimum Mean-Square Error 11](#_Toc127139351)

[2. Weiner-Hopf equation 11](#_Toc127139352)

[2.1. Weiner-Hopf theory and equation 11](#_Toc127139353)

[2.2. Wiener–Hopf Equations for FIR Filters 12](#_Toc127139354)

[3. Adaptive filter 13](#_Toc127139355)

[3.1. FIR filter 13](#_Toc127139356)

[3.2. Lattice filter 15](#_Toc127139357)

[3.3. Adaptive filter algorithms 18](#_Toc127139358)

[3.4. Adaptive filter application 24](#_Toc127139359)

[III. Experiments 32](#_Toc127139360)

[1. Experiment 1: Testing a simple LMS transversal filter 32](#_Toc127139361)

[2. Experiment 2: Testing different parameters values for different adaptive filter algorithms 33](#_Toc127139362)

[3. Experiment 3: Filtering human speech signal 35](#_Toc127139363)

[4. Experiment 4:  Testing different noise scenarios 36](#_Toc127139364)

[4.1. Normal Gaussian noise, clean signal 37](#_Toc127139365)

[4.2. Delay Gaussian noise, delay sample higher than filter order 38](#_Toc127139366)

[4.3. Reference signal got part of the signal mixed in 39](#_Toc127139367)

[4.4. Uncorrelated noise in desired signal 40](#_Toc127139368)

[4.5. Uncorrelated noise as the reference input 40](#_Toc127139369)

[5. Experiment 5: Adaptive noise cancellation without reference signal 41](#_Toc127139370)

[6. Summary 43](#_Toc127139371)

[IV. Conclusion 43](#_Toc127139372)

[**Reference** 45](#_Toc127139373)

List of figures

[Figure 1 Noise cancellation intuition 4](#_Toc127139374)

[Figure 2 Adaptive Noise Canceller 5](#_Toc127139375)

[Figure 3 Block diagram representation of the statistical filtering problem 7](#_Toc127139376)

[Figure 4 Linear convolution: (a) impulse response; (b) filter input; (c) time-reversed and shifted version of filter input; (d) calculation of filter output at time n = 3 9](#_Toc127139377)

[Figure 5 Geometric interpretation of the relationship between the desired response, the estimate at the filter output, and the estimation error 11](#_Toc127139378)

[Figure 6 FIR filter for studying the Wiener filter 13](#_Toc127139379)

[Figure 7 Block diagram of a direct-form FIR adaptive filter 14](#_Toc127139380)

[Figure 8 Signal-flow graph for stage m of a lattice predictor 15](#_Toc127139381)

[Figure 9 Lattice equivalent model of prediction-error (all-zero) filter of order M 16](#_Toc127139382)

[Figure 10 All-pole, all-pass lattice filter of order M 17](#_Toc127139383)

[Figure 11 Adaptive filter algorithms 18](#_Toc127139384)

[Figure 12 Summary of the LMS algorithm (Adaptive filter theory – Simon Haykin) 20](#_Toc127139385)

[Figure 13 Summary of LMS lattice algorithm (Adaptive Filter Theory and application – Behrouz Farhang) 20](#_Toc127139386)

[Figure 14 Summary of the NLMS algorithm (Adaptive filter theory – Simon Haykin) 22](#_Toc127139387)

[Figure 15 Summary of the RLS algorithm (Adaptive filter theory – Simon Haykin) 24](#_Toc127139388)

[Figure 16 Structure of an generic adaptive filter system 24](#_Toc127139389)

[Figure 17 Adaptive filter noise cancellation application 25](#_Toc127139390)

[Figure 18 Adaptive noise cancellation with reference signal 26](#_Toc127139391)

[Figure 19 Adaptive noise cancellation without reference signal 27](#_Toc127139392)

[Figure 20 Adaptive filter evaluation parameters 28](#_Toc127139393)

[Figure 21 Original and input signals for the adaptive filter 32](#_Toc127139394)

[Figure 22 Output y(n) of the filter and error e(n) between the output y(n) and the desired signal 33](#_Toc127139395)

[Figure 23 Error of the output of the filter with the original signal and the filter learning curve 33](#_Toc127139396)

[Figure 24 Output, error and learning curve of all filter structures 34](#_Toc127139397)

[Figure 25 Original speech signal and input signals for the adaptive filter 35](#_Toc127139398)

[Figure 26 Output, error and learning curve off all filter structures (Experiment 3) 36](#_Toc127139399)

[Figure 27 Output, error and learning curve off all filter structures (Experiment 4.1) 38](#_Toc127139400)

[Figure 28 Output, error and learning curve off all filter structures(Experiment 4.2) 39](#_Toc127139401)

[Figure 29 Output, error and learning curve off all filter structures (Experiment 4.3) 39](#_Toc127139402)

[Figure 30 Output, error and learning curve off all filter structures (Experiment 4.4) 40](#_Toc127139403)

[Figure 31 Output, error and learning curve off all filter structures (Experiment 4.5) 41](#_Toc127139404)

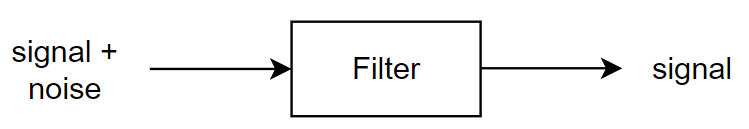
[Figure 32 Original speech signal and inputs signal of the filters 41](#_Toc127139405)

[Figure 33 Output and error of adaptive filters with delay of 200 and filter order of 50 42](#_Toc127139406)

[Figure 34 Output and error of adaptive filters with delay of 300 and filter order of 200 43](#_Toc127139407)

# Introduction

The usual method of estimating a signal corrupted by additive noise is to pass it through a filter that tends to suppress the noise while leaving the signal relatively unchanged i.e. *direct filtering*.



The design of such filters is the domain of optimal filtering, which originated with the pioneering work of Wiener and was extended and enhanced by Kalman, Bucy and others.

Filters used for direct filtering can be either *Fixed* or *Adaptive*.

1. *Fixed filters* - The design of fixed filters requires a priori knowledge of both the signal and the noise, i.e. if we know the signal and noise beforehand, we can design a filter that passes frequencies contained in the signal and rejects the frequency band occupied by the noise.
2. *Adaptive filters* - Adaptive filters, on the other hand, have the ability to adjust their impulse response to filter out the correlated signal in the input. They require little or no a priori knowledge of the signal and noise characteristics. (If the signal is narrowband and noise broadband, which is usually the case, or vice versa, no a priori information is needed, otherwise they require a signal (desired response) that is correlated in some sense to the signal to be estimated). Moreover, adaptive filters have the capability of adaptively tracking the signal under non-stationary conditions.

Noise Cancellation is a variation of optimal filtering that involves producing an estimate of the noise by filtering the reference input and then subtracting this noise estimate from the primary input containing both signal and noise.

A screenshot of a computer

Description automatically generated with medium confidence

Figure 1 Noise cancellation intuition

It makes use of an auxiliary or reference input which contains a correlated estimate of the noise to be cancelled. The reference can be obtained by placing one or more sensors in the noise field where the signal is absent or its strength is weak enough.

Subtracting noise from a received signal involves the risk of distorting the signal and if done improperly, it may lead to an increase in the noise level. This requires that the noise estimate should be an exact replica of *n.* If it were possible to know the relationship between *n* and *n*ˆ, or the characteristics of the channels transmitting noise from the noise source to the primary and reference inputs are known, it would be possible to make *n*ˆ a close estimate of *n* by designing a fixed filter. However, since the characteristics of the transmission paths are not known and are unpredictable, filtering and subtraction are controlled by an adaptive process. Hence an adaptive filter is used that is capable of adjusting its impulse response to minimize an error signal, which is dependent on the filter output. The adjustment of the filter weights, and hence the impulse response, is governed by an adaptive algorithm. With adaptive control, noise reduction can be accomplished with little risk of distorting the signal. In fact, Adaptive Noise Canceling makes possible attainment of noise rejection levels that are difficult or impossible to achieve by direct filtering.

The error signal to be used depends on the application. The criteria to be used may be the minimization of the mean square error, the temporal average of the least squares error etc. Different algorithms are used for each of the minimization criteria e.g. the *Least Mean Squares* (LMS) *algorithm*, the *Recursive Least Squares* (RLS) algorithm etc. To understand the concept of adaptive noise cancellation, we use the minimum mean-square error criterion. The steady-state performance of adaptive filters based on the *mmse* criterion closely approximates that of fixed Wiener filters. Hence, Wiener filter theory provides a convenient method of mathematically analyzing statistical noise canceling problems. From now on, throughout the discussion (unless otherwise stated), we study the adaptive filter performance after it has converged to the optimal solution in terms of unconstrained Wiener filters and use the LMS adaptive algorithm which is based on the Weiner approach.

# Adaptive filter overview

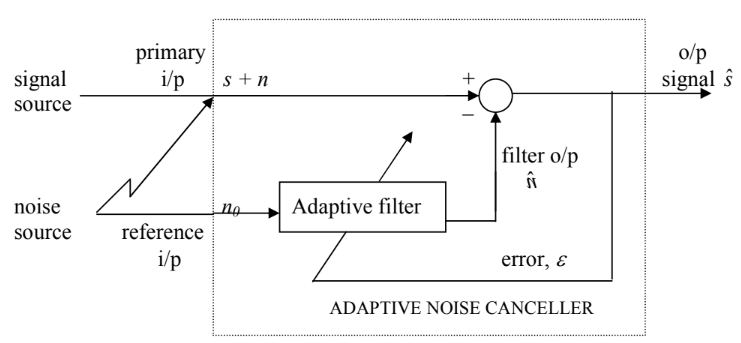


Figure 2 Adaptive Noise Canceller

As shown in the figure, an Adaptive Noise Canceller (ANC) has two inputs – primary and reference. The primary input receives a signal s from the signal source that is corrupted by the presence of noise n uncorrelated with the signal. The reference input receives a noise n0 uncorrelated with the signal but correlated in some way with the noise n. The noise no passes through a filter to produce an output that is a close estimate of primary input noise. This noise estimate is subtracted from the corrupted signal to produce an estimate of the signal at , the ANC system output. In noise canceling systems a practical objective is to produce a system output that is a best fit in the least squares sense to the signal s. This objective is accomplished by feeding the system output back to the adaptive filter and adjusting the filter through an LMS adaptive algorithm to minimize total system output power.

In other words the system output serves as the error signal for the adaptive process. The Weiner filter is a type of digital filter used for signal processing in communication systems and control systems. It was first developed by Norbert Wiener in the 1940s, and is named after him. The Weiner filter is a linear, time-invariant filter that optimizes a trade-off between mean square error and mean square control effort.

In communication systems, the Weiner filter is commonly used for channel equalization, where it helps to mitigate the effects of channel distortion on the transmitted signal. In this application, the filter helps to reduce inter-symbol interference and noise in the received signal, leading to improved signal quality and better error rates.

In control systems, the Weiner filter is used to estimate the state of a dynamic system. This is often done in the presence of measurement noise, which can cause significant inaccuracies in the estimated state. The Weiner filter is used to optimally balance the trade-off between the mean square error of the state estimate and the mean square control effort required to produce the estimate.

The Weiner filter can be implemented using two main approaches: the Wiener-Hopf equation and the Kalman filter. The Wiener-Hopf equation is a mathematical formula that can be used to directly calculate the filter coefficients for a given system. The Kalman filter is a recursive algorithm that can be used to estimate the state of a dynamic system over time. Both of these approaches have their own strengths and weaknesses, and the choice of which approach to use will depend on the specific requirements of the application.

One of the key advantages of the Weiner filter is its ability to handle both linear and non-linear systems. In the case of linear systems, the filter can be designed using straightforward mathematical techniques. In the case of non-linear systems, the filter can be designed using more advanced techniques, such as the extended Kalman filter or the unscented Kalman filter.

Another advantage of the Weiner filter is its versatility. The filter can be used in a wide range of applications, including signal processing, control systems, and estimation. It can also be used with both continuous-time signals and discrete-time signals, making it well-suited for use in digital signal processing applications.

In conclusion, the Weiner filter is a powerful tool for signal processing and control systems. Its ability to handle both linear and non-linear systems, its versatility, and its ability to optimally balance mean square error and mean square control effort make it a valuable tool for engineers and researchers working in these fields. With its long history and its continued relevance today, the Weiner filter is a fundamental building block of modern communication and control systems.

## Weiner linear filter

### Linear Optimum Filtering: Statement of the Problem

Diagram

Description automatically generated

Figure 3 Block diagram representation of the statistical filtering problem

The filter input consists of a time series u(0), u(1), u(2), . . . , and the filter is itself characterized by the impulse response represented by the sequence w0, w1, w2, . . . . At some discrete time n, the filter produces an output denoted by y(n). This output is used to provide an estimate of a desired response designated by d(n). The estimation error, denoted by e(n), is defined as the difference between the desired response d(n) and the filter output y(n). The requirement is to make the estimation error e(n) “as small as possible” in some statistical sense.

Two restrictions have been placed on the filter so far:

1. The filter is linear, which makes the mathematical analysis easy to handle.

2. The filter operates in discrete time, which makes it possible for the filter to be implemented using digital hardware or software.

The final details of the filter specification, however, depend on two other choices that have to be made:

1. Whether the impulse response of the filter has finite or infinite duration.

2. The type of statistical criterion used for the optimization.

The choice of a finite-duration impulse response (FIR) or an infinite-duration impulse response (IIR) for the filter is dictated by practical considerations. The choice of a statistical criterion for optimizing the filter design is influenced by mathematical tractability.

FIR filter is inherently stable, since its structure involves the use of forward paths only. In other words, the only mechanism for input–output interaction in the filter is via forward paths from the filter input to its output. Indeed, it is this form of signal transmission through the filter that limits its impulse response to a finite duration. On the other hand, an IIR filter involves both feedforward and feedback. The presence of feedback means that portions of the filter output and possibly other internal variables in the filter are fed back to the input. Consequently, unless the filter is properly designed, feedback can indeed make it unstable, with the result that the filter oscillates; this kind of operation is clearly unacceptable when the requirement is that stability is a “must.”

When the filter is required to be adaptive, bringing with it stability problems of its own, the inclusion of adaptivity combined with feedback that is inherently present in an IIR filter makes an already difficult problem that much more difficult to handle. It is for this reason that we find that in the majority of applications requiring the use of adaptivity, the use of an FIR filter is preferred over an IIR filter even though the latter is less demanding in computational requirements.

Turning next to the issue of what criterion to choose for statistical optimization, we find that there are indeed several criteria that suggest themselves. Specifically, may consider optimizing the filter design by minimizing a cost function, or index of performance, selected from the following short list of possibilities

1. Mean-square value of the estimation error.

2. Expectation of the absolute value of the estimation error.

3. Expectation of third or higher powers of the absolute value of the estimation error.

We develop the mathematical solution to this statistical optimization problem by following two entirely different approaches that are complementary. One approach leads to the development of an important theorem commonly known as the principle of orthogonality. The other approach highlights the error-performance surface that describes the second-order dependence of the cost function on the filter coefficients.

### Principle of Orthogonality

Consider again the statistical filtering problem described in Fig. 2.1. The filter input is denoted by the time series u(0), u(1), u(2), . . . , and the impulse response of the filter is denoted by w0, w1, w2, . . . , both of which are assumed to have complex values and infinite duration. The filter output at a discrete time n is defined by the linear convolution sum

The purpose of the filter in Fig. 2.1 is to produce an estimate of the desired response d(n). The estimation of d(n) is naturally accompanied by an error, which is defined by the difference

Chart

Description automatically generated

Figure 4 Linear convolution: (a) impulse response; (b) filter input; (c) time-reversed and shifted version of filter input; (d) calculation of filter output at time n = 3

The estimation error e(n) is the sample value of a random variable. To optimize the filter design, we choose to minimize the mean-square value of e(n). We thus define the cost function as the mean-square error

For complex input data, the filter coefficients are, in general, complex, too. Let the kth filter coefficient wk be denoted in terms of its real and imaginary parts as

Thus, for the situation at hand, applying the operator ∇ to the cost function J, we obtain a multidimensional complex gradient vector ∇J, the kth element of which is

The cost function J is a scalar that is independent of time n. Hence, substituting the first term of that equation into Eq. (2.6), we get

With

We have:

Let eo denote the special value of the estimation error that results when the filter operates in its optimum condition. We then find that the conditions specified in Eq. (2.7) are indeed equivalent to

Interpolate from the equation above, we can state that: The necessary and sufficient condition for the cost function J to attain its minimum value is for the corresponding value of the estimation error eo(n) to be orthogonal to each input sample that enters into the estimation of the desired response at time n

This statement constitutes the principle of orthogonality; it represents one of the most elegant theorems in the subject of linear optimum filtering.

Let denote the output produced by the filter optimized in the mean-square-error sense, with denoting the corresponding estimation error. Hence, using the principle of orthogonality, we get the desired result:

We may thus state the corollary to the principle of orthogonality as follows: When the filter operates in its optimum condition, the estimate of the desired response defined by the filter output yo(n) and the corresponding estimation error eo(n) are orthogonal to each other.

Equation above offers an interesting geometric interpretation of the conditions that exist at the output of the optimum filter. In this figure, the desired response, the filter output, and the corresponding estimation error are represented by vectors labeled d, yo, and eo, respectively; the subscript o in yo and eo refers to the optimum condition. We see that, for the optimum filter, the vector representing the estimation error is normal (i.e., perpendicular) to the vector representing the filter output.

Chart

Description automatically generated

Figure 5 Geometric interpretation of the relationship between the desired response, the estimate at the filter output, and the estimation error

### Minimum Mean-Square Error

When the linear discrete-time filter in Fig. 3 operates in its optimum condition, Eq. (2.2) takes on the special form

Let

denote the minimum mean-square error. Then, evaluating the mean-square values of both sides of equation above, we get:

The optimum filter operates perfectly, in the sense that there is complete agreement between the estimate at the filter output and the desired response and vice versal.

## Weiner-Hopf equation

### Weiner-Hopf theory and equation

The principle of orthogonality, described in Eq. (2.11), specifies the necessary and sufficient condition for the optimum operation of the filter. We may reformulate the necessary and sufficient condition for optimality by substituting Eq. (2.1) and (2.2) into Eq. (2.11). In particular, we write

where woi is the ith coefficient in the impulse response of the optimum filter. Expanding this equation and rearranging terms, we get

The two expectations in Eq. (2.24) may be interpreted as follows:

The expectation is equal to the autocorrelation function of the filter input for a lag of i - k. We may thus express this expectation as

The expectation 𝔼3u1n - k2d\*1n24 is equal to the cross-correlation between the filter input u1n - k2 and the desired response d(n) for a lag of -k. We may thus express this second expectation as

Accordingly, using the definitions of Eq. (2.25) and (2.26) in Eq. (2.24), we get an infinitely large system of equations as the necessary and sufficient condition for optimality of the filter:

These equations are called the Wiener–Hopf equations.

### Wiener–Hopf Equations for FIR Filters

The solution of the set of Wiener–Hopf equations is greatly simplified for the special case when an FIR filter, also known as a linear transversal filter, is used to obtain the estimation of the desired response d(n).

Diagram

Description automatically generated

Figure 6 FIR filter for studying the Wiener filter

FIR filter involves a combination of three basic operations: storage, multiplication, and addition, as described here:

1. The storage is represented by a cascade of M - 1 one-sample delays, with the block for each such unit labeled z-1. We refer to the various points at which the one-sample delays are accessed as tap points. The tap inputs are denoted by u(n). Thus, with u(n) viewed as the current value of the filter input, the remaining M - 1 tap inputs, u(n – 1), u(n-2), ..., u(n-M+1) , represent past values of the input.

2. The scalar inner products of tap inputs u(n), u(n – 1), u(n-2), ..., u(n-M+1) and tap weights w0, w1, ..., wM-1 are respectively formed by using a corresponding set of multipliers. In particular, the multiplication involved in forming the scalar inner product of u(n) and w0 is represented by a block labeled w\*0 , and so on for the other inner products.

3. The function of the adders is to sum the multiplier outputs to produce an overall output for the filter

## Adaptive filter

### FIR filter

Finite Impulse Response FIR filters as the name suggests, have an impulse response with finite length. A non-recursive filter has no feedback and its input-output relation is given in (2) by the linear constant coefficient difference equation.

A picture containing text

Description automatically generated

The output y(n) of a non-recursive filter is independent of the past output values, it is a function only of the input signal x(n) and the filter coefficient b(k), where k=0,1,…,q. The response of such a filter to an impulse consists of a finite sequence of q+1 samples, where q is the filter order. A direct-form FIR adaptive filter for estimating a desired signal d(n) from the related input signal x(n) is illustrated in the next figure

Diagram, schematic

Description automatically generated

Figure 7 Block diagram of a direct-form FIR adaptive filter

In designing the FIR adaptive filter, the goal is to find the coefficient vector wn at sample n that minimizes the mean-square error. In (3) the filter output y(n) of an FIR adaptive filter for estimating a desired signal d(n) from a related signal x(n) is calculated.

A picture containing text

Description automatically generated

It is assumed that both signals x(n) and d(n) are non-stationary signals and the goal is to find the coefficient vector at time n that minimizes the mean-square error.

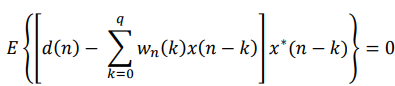


The error signal is calculated from the difference between the filter output signal y(n) and the desired signal d(n).



To find the filter coefficients that minimize the mean-square error it is necessary to set the derivative 𝜉(𝑛) equal to zero with respect to 𝑤𝑛 ∗ (k) for k = 0,1…, q and ∗ represents the complex conjugate, which leads to the result





FIR filters are commonly used for adaptive noise cancellation applications. The FIR filter in its non-recursive form is always stable. FIR filters can have a linear phase response and they can be set up in order introduce no phase distortion to the signal. The FIR filter can have any structure, like direct form, cascade form or lattice form, but the most common form is the direct form, also known as transversal structure

### Lattice filter

To implement the Gram–Schmidt algorithm for transforming an input vector u(n) consisting of correlated samples into an equivalent vector b(n) consisting of uncorrelated backward prediction errors, we may use the parallel connection of a direct path and an appropriate number of backward prediction-error filters. The vectors b(n) and u(n) are said to be “equivalent” in the sense that they contain the same amount of information. A much more efficient method of implementing the Gram–Schmidt orthogonalization algorithm, however, is to use an order-recursive structure in the form of a ladder, known as a *lattice predictor*. This system combines several forward and backward prediction-error filtering operations into a single structure. Specifically, a lattice predictor consists of a cascade connection of elementary units (stages), all of which have a structure similar to that of a lattice—hence the name. The number of stages in a lattice predictor equals the prediction order. Thus, for a prediction-error filter of order m, there are m stages in the lattice realization of the filter.

Diagram

Description automatically generated

Figure 8 Signal-flow graph for stage m of a lattice predictor

Diagram

Description automatically generated

Figure 9 Lattice equivalent model of prediction-error (all-zero) filter of order M

The lattice filter offers the following attractive features:

1. A lattice filter is a highly efficient structure for generating the sequence of forward prediction errors and the corresponding sequence of backward prediction errors simultaneously.

2. The various stages of a lattice predictor are “decoupled” from each other. This decoupling property shown that the backward prediction errors produced by the various stages of a lattice predictor are “orthogonal” to each other for wide-sense stationary input data.

3. The lattice filter is modular in structure; hence, if we are required to increase the order of the predictor, we simply add one or more stages (as desired) without affecting earlier computations

The multistage lattice predictor combines two all-zero prediction-error filters into a single structure. More specifically, invoking Properties 3 and 4 of prediction error filters, we may respectively make the following statements:

• The path from the common input to the forward prediction error is a minimum-phase all-zero filter.

• The alternative path from the common input to the backward prediction error  is a maximum-phase all-zero filter. The multistage lattice predictor may be rewired to operate as a combined all-pole, all-pass lattice filter. To do this rewiring, we first obtain:



where the forward prediction error is now treated as an input variable for the mth stage of the rewired lattice filter.



Two equation above, define the input–output relations of the mth stage in the rewired lattice filter. Thus, starting with the initial conditions corresponding to order m = 0 and progressively increasing the order of the filter, we obtain the rewired multistage lattice filter.

Diagram

Description automatically generated

Figure 10 All-pole, all-pass lattice filter of order M

To excite the multistage lattice filter of Fig 10, a sequence of samples taken from white noise is used as the input signal . The path from the input to the output in Fig. 10 is an all-pole filter of order M.

The backward prediction error constitutes the second output of the filter in Fig. 10. The path from the input to the output is an all-pass filter of order M. The poles and zeros of this filter are the inverse of each other with respect to the unit circle in the z-plane. Thus, the rewired multistage lattice filter of Fig. 10 combines an all-pole filter and an all-pass filter in a single structure

### Adaptive filter algorithms

Adaptive algorithms are used to adjust the coefficients of the digital filter, such that the error signal is minimized according to some criterion. There are different types of adaptive filtering methods. A distinction is made between least mean-square LMS and recursive least square RLS algorithm. An overview of the different types can be seen in Figure below:

Diagram

Description automatically generated

Figure 11 Adaptive filter algorithms

Least mean-squares LMS algorithms adapt the filter coefficients until the difference between the desired and the actual signal is minimized, that relate to producing the least mean-squares of the error signal. Apart from the Least mean-square LMS, the Normalized Least Mean-square NLMS, the Time Varying Least Mean-square TVLMS and the Variable Step-Size Least Mean-square VSSLMS belongs to this class of algorithms. The LMS algorithms are based on the stochastic gradient descent method in that the filter coefficients are only adapted based on the error at the current time.

Recursive adaptive filtering algorithms, like the Recursive Least Square RLS or the Fast Transversal Recursive Least Square FTRLS, recursively find the coefficients that minimize a weighted linear least squares cost function relating to the input signals. This approach is in contrast to the least mean-squares LMS that aim to reduce the mean-square error. The adaptation of the filter coefficients is based on all error data, but when using the forgetting factor, the older data can be de-emphasized compared to the newer data.

The most commonly used adaptive algorithms for noise cancellation are the Least Mean-square LMS, the Normalized Least Mean-square NLMS and the Recursive Least Square RLS algorithm. Therefore, the three different algorithms will be compared and investigated in greater detail.

#### Least Mean Square

The Least Mean Square LMS algorithm is one of the simplest and most widely used algorithms for adaptive filtering. The LMS algorithm is based on the stochastic gradient descent method to find a coefficient vector which minimizes a cost function. In contrast to the Wiener filter, the parameters of the LMS algorithm changes for each new sample. In the steepest descent adaptive filter, the weight-vector update equation is given by

The steepest descent algorithm shows a practical limitation in that the expectation is generally unknown. Therefore, it must be replaced with an estimate such as the sample mean

Incorporating this estimate into the steepest descent algorithm, the update for the  
weight vector becomes

In the following equation, the ensemble average is estimated using a one-point sample mean (L=l).

Finally, combining the equations leads to the update equation below, that is known  
as LMS algorithm:

where wn is the estimate of the weight value vector at time n, x(n) is the input signal  
vector, e(n) is the filter error vector and µ is the step-size, which determines the  
filter convergence rate and overall behavior. One of the difficulties in the design and implementation of the LMS adaptive filter is the selection of the step-size μ. This parameter must lie in a specific range, so that the LMS algorithm converges: 0 < 𝜇 < 2 /𝜆max, where 𝜆𝑚𝑎𝑥 is the largest eigenvalue of the autocorrelation matrix Rx.

A summary of the LMS transversal and LMS lattice filter algorithms are also provided for application in MATLAB later

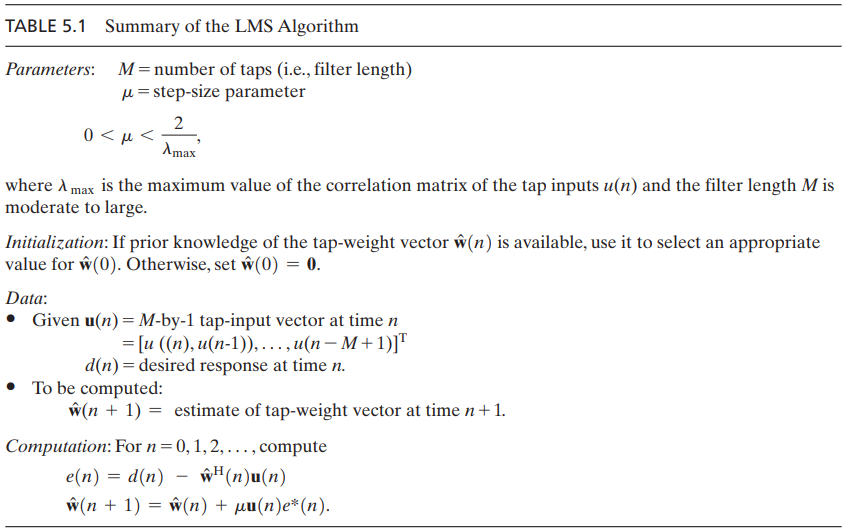


Figure 12 Summary of the LMS algorithm (Adaptive filter theory – Simon Haykin)

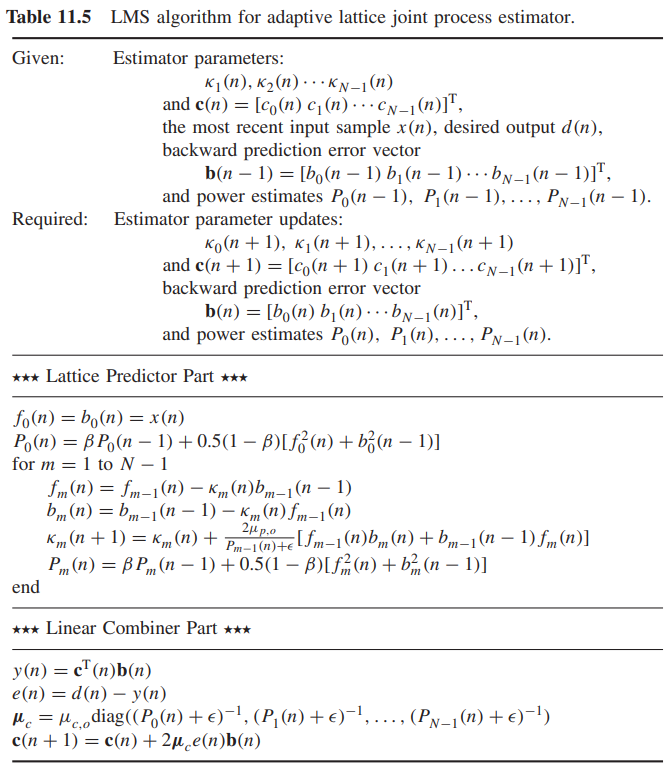


Figure 13 Summary of LMS lattice algorithm (Adaptive Filter Theory and application – Behrouz Farhang)

#### Normalize Least Mean Square

The design problem of the LMS algorithm lies in the step size μ. In order to solve  
this difficulty, the Normalized LMS algorithm was developed. The correction  
applied to the weight vector wn at sample n+1 is normalized with respect to the  
input vector x(n) at iteration n.

For wide-sense stationary processes, the LMS algorithm converges in the mean square of the autocorrelation matrix 0 < 𝜇 < 2 /𝑡𝑟(𝑅𝑥). The bound in the above equation can be calculated from

Therefore, the condition for mean-square convergence may be replaced with

where 𝐸{|x(𝑛)|2} is the power of input speech signal x(n). This power can be  
estimated using a time average such as

This leads to the following bound on the step-size for mean-square convergence

Then, the time-varying step-size is given by

where 𝛽 is a normalized step-size in the range of 0 < 𝛽 < 2. Replacing μ in the LMS weight vector update equation with μ(n) leads to the normalized LMS algorithm NLMS in

The Normalized LMS algorithm can be viewed as the LMS algorithm with a time varying step size para meter.

A summary of the NLMS transversal algorithm is also provided for application in MATLAB later

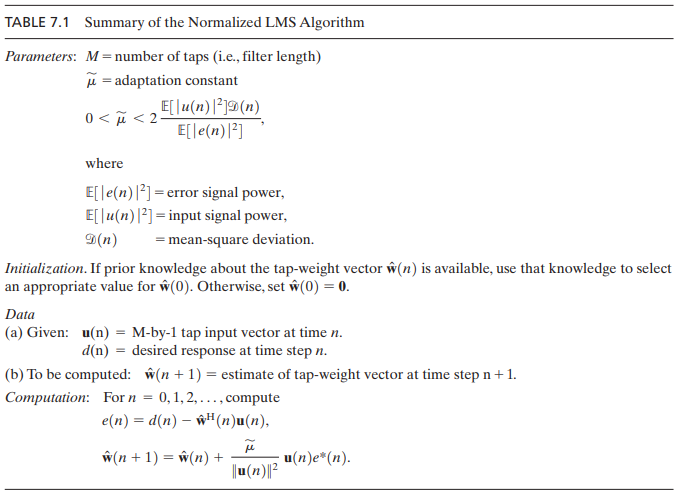


Figure 14 Summary of the NLMS algorithm (Adaptive filter theory – Simon Haykin)

#### Recursive Least Square

Unlike the LMS algorithm, which aims to reduce the mean-square error, the RLS  
algorithm aims to recursively find the filter coefficients that minimize the least  
squares cost function [14]. Compared to the mean-square error, the least squares  
error can be minimized directly from the data x(n) and d(n). The recursive least  
squares algorithm produces a set of filter coefficients wn(k) at sample n that  
minimize the weighted least squares error

The RLS algorithm involves the recursive updating of the vector wn and the inverse  
autocorrelation matrix P(n). In the evaluation of the gain vector g(n) and the inverse  
autocorrelation matrix P(n), it is necessary to compute the product

The gain vector g(n) can be calculated as follows

where 𝜆 is the so-called forgetting factor, which gives exponentially less weight to  
older error samples. Therefore, it is also defined as the exponential weighting factor  
and lies in the range of 0 < 𝜆 ≤ 1. Incorporating these definitions, the  
autocorrelation matrix P(n) results in

The so-called priori error α(n) is calculated, which is the difference between  
the desired signal d(n) and the estimate of d(n) that is formed by applying the  
pervious set of filter coefficients 𝑤𝑛-1 to the new data vector x(n)   
   
Finally, combining the equations leads to the exponentially weighted Recursive Least Squares RLS algorithm

The special case of 𝜆 = 1 is referred to as the growing window RLS algorithm,  
where all previous errors are considered of equal weight in the total error.  
Since the RLS algorithm involves the recursive updating of the vector wn and the  
inverse autocorrelation matrix P(n), the initial conditions for both of these terms are  
required

A summary of the RLS transversal algorithm is also provided for application in MATLAB later

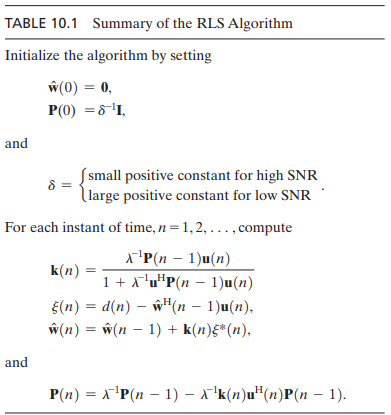


Figure 15 Summary of the RLS algorithm (Adaptive filter theory – Simon Haykin)

### Adaptive filter application

Adaptive filters are digital filters whose coefficients change with an objective to make the filter converge to an optimal state. The optimization criterion is a cost function, which is most commonly the mean square of the error signal between the output of the adaptive filter and the desired signal. As the filter adapts its coefficients, the mean square error (MSE) converges to its minimal value. At this state, the filter is adapted and the coefficients have converged to a solution. The filter output, *y(k)*, is then said to match very closely to the desired signal, *d(k)*. When you change the input data characteristics, sometimes called *filter environment*, the filter adapts to the new environment by generating a new set of coefficients for the new data.

Diagram

Description automatically generated

Figure 16 Structure of an generic adaptive filter system

#### Adaptive filter noise cancellation application

Adaptive noise cancellation is the approach used for estimating a desired signal d(n) from a noise-corrupted observation . The method uses a primary input containing the corrupted signal and a reference input containing noise correlated in some unknown way with the primary noise. The reference input v2(n) is adaptively filtered and subtracted from the primary input to obtain the signal estimate 𝑑̂ (n)

Diagram

Description automatically generated

Figure 17 Adaptive filter noise cancellation application

Subtracting noise from a received signal runs the risk of distorting the signal. If this is not done properly, it may increase the noise level. This requires that the noise estimate should be an exact replica of . Since the properties of the transmission paths are unknown and unpredictable, filtering and subtraction are controlled by an adaptive process. Therefore, an adaptive filter is used that is able to adjust its impulse response to minimize an error signal, that depends on the filter output.

Without any information about or it is not possible to separate the signal from the noise. When a reference signal is given, that is correlated with the primary noise , it can be used to estimate the desired signal d(n) from the noisy observation. The reference signal can be obtained by placing one or more sensors in the noise field. Unfortunately, in many applications no reference signal is available. Particularly in non-stationary processes, the required statistics of and are generally unknown. When there is no reference signal, a noise canceller without reference can be designed with an adaptive filter. A distinction is made between adaptive noise cancellation with and without reference signal.

* + Adaptive noise cancellation with reference signal

A desired signal is transmitted over a channel to a sensor that also receives a noise uncorrelated with the signal. The primary input to the canceller is a combination of both signal and noise Here, the reference signal is known, which is uncorrelated with the signal but correlated with the noise . In that case, a second sensor receives the reference signal and provides the input to the adaptive filter. Figure below shows the principle of adaptive noise cancellation with reference signal.

Diagram, schematic

Description automatically generated

Figure 18 Adaptive noise cancellation with reference signal

The reference signal is filtered to produce the noise estimate , that is close to the primary noise . The output of the adaptive filter is then subtracted from the primary input to produce the signal estimate that is the best fit in the least squares sense to the desired signal . This objective is accomplished by feeding the signal estimate back to the adaptive filter and adjusting the filter through an adaptive algorithm to minimize the total system output power E. In an adaptive noise cancelling system, the system output serves as the input error signal for the adaptive process e (n) = .

The system output is

Taking expectations of both sides and realizing that d(n) is uncorrelated with

The signal power will be unaffected as the filter is adjusted to minimize

When the filter is adjusted to minimize the output noise power , the output noise is also minimized. Since the signal in the output remains constant, therefore minimizing the total output power maximizes the output signal-to-noise ratio.

If the reference signal is uncorrelated with d(n), it follows that minimizing the mean-square error E{} is equivalent to minimizing E{[}. In other words, the output of the adaptive filter is the minimum mean-square estimate of , then it follows that is the minimum mean-square estimate of d(n).

Since [𝑑̂(𝑛) − 𝑑(𝑛)] = [𝑣 1 (𝑛) − 𝑣̂ 1 (𝑛)]

This is equivalent to causing the output 𝑑̂(𝑛) to be best least squares estimate of the signal d(n).

* + Adaptive noise cancellation without reference signal

In many non-stationary applications, it is difficult or impossible to obtain the reference signal. For example, when a broadband signal is corrupted by periodic interference, no reference signal is available. In that case, it is possible to derive a reference signal by delaying the noisy process . The delayed signal is used as the reference signal to eliminate the interference. The principle of the noise cancellation without reference signal is illustrated in figure below:

Diagram, schematic

Description automatically generated

Figure 19 Adaptive noise cancellation without reference signal

The fixed delay chosen must be of sufficient length to cause the broadband signal components in the reference input to become uncorrelated from those in the primary input. The interference components, because of their periodic nature, will remain correlated with each other. The mean-square error is formed by taking the difference between x(n) and the output of the adaptive filter . Taking the same ideal as “Adaptive noise cancellation with reference signal “, after some mathematic transformation, the system output is:

In addition, the input to the adaptive filter is

Since the interference is uncorrelated with d(n) as well as with , it follows that minimizing the mean-square error is equivalent to minimizing E{}, the mean-square error between d(n) and the output of the adaptive filter y(n).

Under the assumption that is a broadband process and the delay n 0 is greater than the decorrelation time, the delayed process is uncorrelated with the noise, but correlated with . This results in an adaptive filter that produces the minimum mean-square estimate of the broadband process and the input error signal corresponds to an estimate of the noise

#### Evaluation parameter

In order to be able to compare the discussed adaptive filtering algorithms against each other in terms of the efficiency of noise cancelling, some characteristics must be defined which can be evaluated for each algorithm. For the comparison of the chosen algorithms discussed in the previous subchapters, the following performance criteria are used: the rate of convergence, the performance of the mean-square error MSE or least squares error LSE and the signal-to-noise ratio SNR after filtering.

Diagram

Description automatically generated

Figure 20 Adaptive filter evaluation parameters

In many noise cancellation applications, a low rate of convergence and a minimum mean-square error are desired characteristics. For satisfactory performance in noise cancellation, a low rate of convergence allows the algorithm to adapt rapidly to a stationary environment of unknown statistics, but the convergence speed is not independent from other performance characteristics.

There will be a trade-off in other performance criteria for an improved convergence rate and there will be a reduced convergence performance for an increase in other performance. In some applications, the system stability will drop when the rate of convergence is decreased, causing the system more likely to diverge rather than converging to a proper solution. To ensure a stable system, the parameters that affect the rate of convergence must be within certain limits.

Each algorithm works on different methods for noise cancellation and reaches system stability in different ways. In order to find the best adaptive filtering algorithm for noise cancellation, a trade-off between the three performance criteria must be considered. The performance characteristics of the LMS, NLMS and RLS algorithms are studied by taking the criteria such as convergence speed and mean- square error into consideration along with the number of iterations.

* Rate of convergence

The rate of convergence is defined as the number of adaptation cycles required for the algorithm to converge from some initial condition to its steady-state or close enough to an optimum, like the optimum Wiener solution in the mean-square error sense. The rate of convergence can be found out by using a learning curve, which shows the averaged mean-square error MSE or least squares error LSE performances as a function of the number of iterations. Depending on each algorithm, the rate of convergence is influenced by different factors.

The convergence characteristics of the LMS adaptive algorithm depends on two factors: the step-size µ and the eigenvalue spread of the autocorrelation matrix χ(Rx). The step-size µ must lie in a specific range 0 < 𝜇 < 2 / 𝜆 𝑚𝑎𝑥 where 𝜆 𝑚𝑎𝑥 is the largest eigenvalue of the autocorrelation matrix Rx. A large value of the step-size µ will lead to a faster convergence but may be less stable around the minimum value. The convergence of the algorithm is inversely proportional to the eigenvalue spread of the correlation matrix.

In the NLMS algorithm the dependence of µ from the autocorrelation matrix is overcome through using a variable step-size parameter in which the variation is achieved due to the division, at each iteration, of the fixed step-size by the input power. The variable step-size is computed by

where 𝛽 is a normalized step-size with 0 < 𝛽 < 2. The use of the variable step- size eliminates much of the trade-off between residual error and convergence speed compared with the fixed step-size.

In comparison to the LMS adaptive algorithm, where the convergence behavior depends on the step-size µ, the convergence rate of the RLS adaptive filter is based on the inverse autocorrelation matrix P(n), which has the effect of whitening the tap inputs. Further it depends on the exponential weighting factor 𝜆. The exponential weighting factor 𝜆 must be greater than zero and less than or equal to one 0 < 𝜆 ≤ 1.

* Error performance

Adaptive filters attempt to optimize the performance by minimizing the error signal between the output of the adaptive filter and the desired signal according to some criterion. A large error value indicates that the adaptive filter cannot accurately track the desired signal. A minimal error value ensures that the adaptive filter is optimal. The different adaptive filtering algorithms are highly dependent on the optimization criterion.

The criterion of the LMS and NLMS algorithm is the minimum mean-square of the error signals. The MSE is defined as the ensemble average of the squared error sequence, denoted as 𝜉(𝑛) = 𝐸{}

The so-called miss-adjustment is another performance measure for algorithms that use the minimum MSE criterion. The miss adjustment ℳ is the ratio of the steady- state excess mean-square error to the minimum mean-square error, which can be mathematically described as

A trade-off between a low rate of convergence and a small mean-square error or miss adjustment is necessary, because when the step-size 𝜇 increases, the rate of convergence decreases, but the MSE increases.

In the RLS algorithm, two different errors must be considered, the a priori estimation error 𝜉(𝑛) is the error that would occur if the filter coefficients were not updated and the a posteriori error e(n), on the other hand, occurs after the weight vector is updated. The least-squares optimization criterion of the RLS algorithm depends in general on the cost function 𝜀(𝑛) based on e(n), not 𝜉(𝑛).

The error signal e(n) in the RLS algorithm is defined differently from that in the LMS algorithm. But it is possible to make a direct graphical comparison between the learning curves of the RLS with the other two algorithms by choosing 𝜉(𝑛) as the error of interest. For the comparison, the ensemble-average of the a priori estimation error 𝜉(𝑛) in the equation below has to be computed. 𝐸{|}. The learning curve will have the same general form as the LMS algorithm

* + Signal-to-noise ratio SNR

The signal-to-noise ratio SNR is another important performance criterion in adaptive noise cancellation and describes the relationship between the strength of the input signal and the noise signal. The SNR is defined in by the ratio of the signal power to the noise power and is often expressed in decibel.

In order to compare the different adaptive filtering algorithms in the efficiency of noise cancellation, the so-called improvement SNR level in is used, which is the difference between the input and output SNR: Therefore, the SNR is calculated before and after applying the adaptive filter. The signal-to-noise ratio SNR in decibels is computed by the ratio of the summed squared magnitude of the signal to that of the noise. The input SNR is the ratio between the power of input signal and power of noise at the input

where is the noise-corrupted signal and is the noise sequence. As there is no information about the noise signal, it is not possible to calculate exactly the input SNR, it can only be estimated from the sinusoid.

The output SNR has to be higher than the input SNR, which indicates the success of noise removal. A lower value of the output SNR compared with the input SNR means that the filtering process introduces more noise instead of reducing noise. The output SNR is the ratio between the power of the filtered signal and power of the noise at output.

where is the output signal of the adaptive filter and is the noise signal. A large value of the output SNR is desirable, which indicates that the adaptive filter can remove a large amount of noise and is able to produce an accurate estimate of the desired signal. The signal-to-noise ratio increases when the output noise power decreases. Minimizing the output power causes the filtered signal to be perfectly noise-free.

# Experiments

In this section, a few experiments were performed to test the performance of different adaptive filter structures and algorithms in different scenarios. All the experiments are performed in MATLAB R2020a application. All the code for each experiment will be provided in the appendix of this report and also in the Github repository <https://github.com/huynhducmink/adsp_final>

## Experiment 1: Testing a simple LMS transversal filter

A sinusoidal signal x = sin(0.05\*pi\*t) is generated with a length of 2000 samples to be the original, non corrupted signal. A White Gaussian noise signal with strength of 0dB is generated then added to the original signal to be the desired signal of the filter. The noise signal is also the reference signal for the filter.

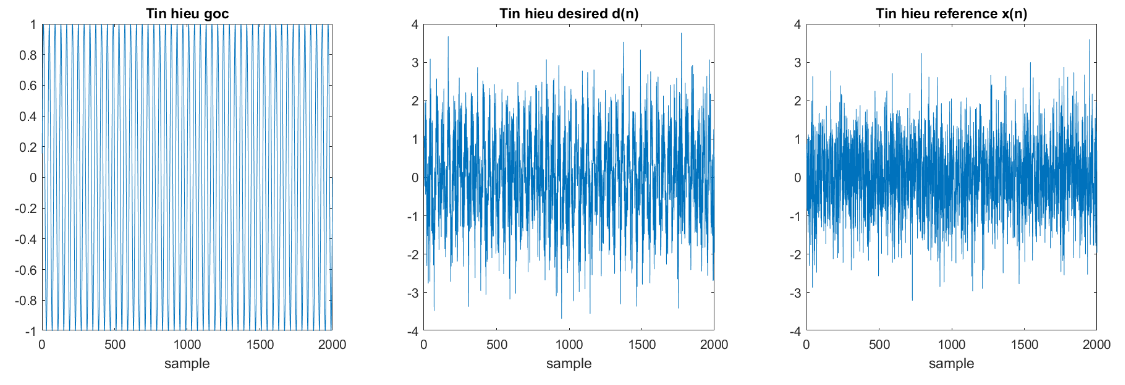


Figure 21 Original and input signals for the adaptive filter

Next, we construct a LMS transversal filter with a filter order M = 5 and the step-size parameter mu\_LMS = 0.003.

By observing the output of the adaptive filter y(n) and the error of the system e(n), we can see the filter output y(n) converge to the reference signal, resulting in the output of the system e(n) converging to the original sinusoidal signal. The system reaches steady state after ~1000 filter update iterations, with the mean square error value of -20.97dB for all 2000 signal samples, and -46.52dB for the final 1000 sample, reducing from -0.05dB of the original corrupted signal.

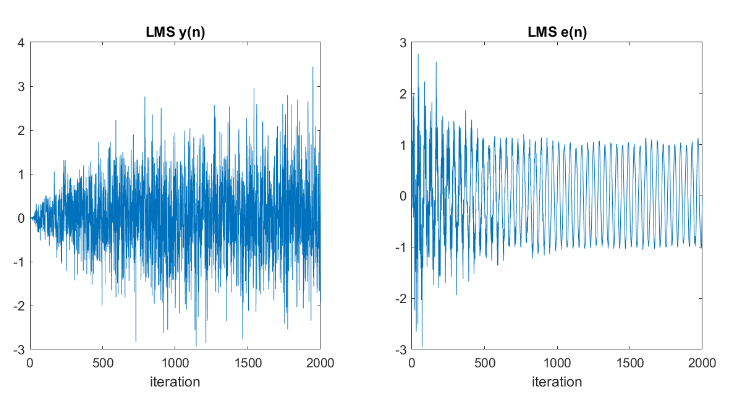


Figure 22 Output y(n) of the filter and error e(n) between the output y(n) and the desired signal

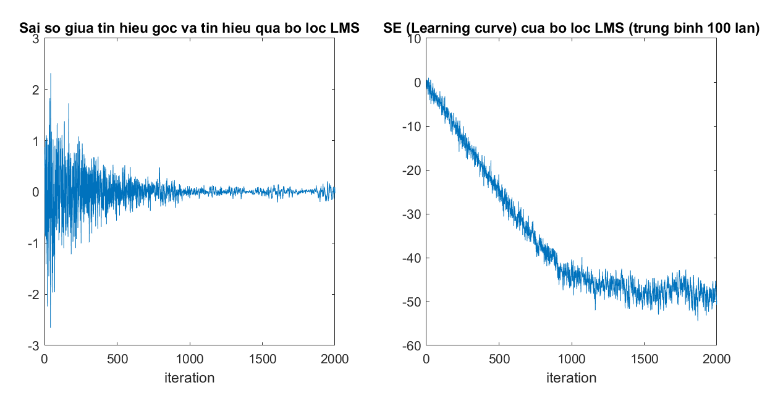


Figure 23 Error of the output of the filter with the original signal and the filter learning curve

As we can observe the convergence rate of this filter is low, we can also try to improve the convergence rate by increasing the mu parameter of the LMS algorithm. However, the trade off is that the error value of the final signal will be higher. Vice versa, we can also try to reduce the error of the filter by reducing the mu parameter, but the convergence rate will be lower.

The next experiment will be performed to test this theory.

## Experiment 2: Testing different parameters values for different adaptive filter algorithms

In this experiment, we tested 4 different adaptive filters with changes in filter parameters to observe the change in convergence rate, mean square error and calculation cost (calculation time).

The 4 filter structure in this experiments are:

* Transversal LMS adaptive filter
* Transversal NLMS adaptive filter
* Transversal RLS adaptive filter
* Lattice LMS adaptive filter

The original signal here is a sinusoidal signal x = sin(0.05\*pi\*t) with a length of 5000 samples. The noise source is a Gaussian White noise with noise power of 0dB. To simplify the experiment, all adaptive filter structures tested here have a filter order M = 10.

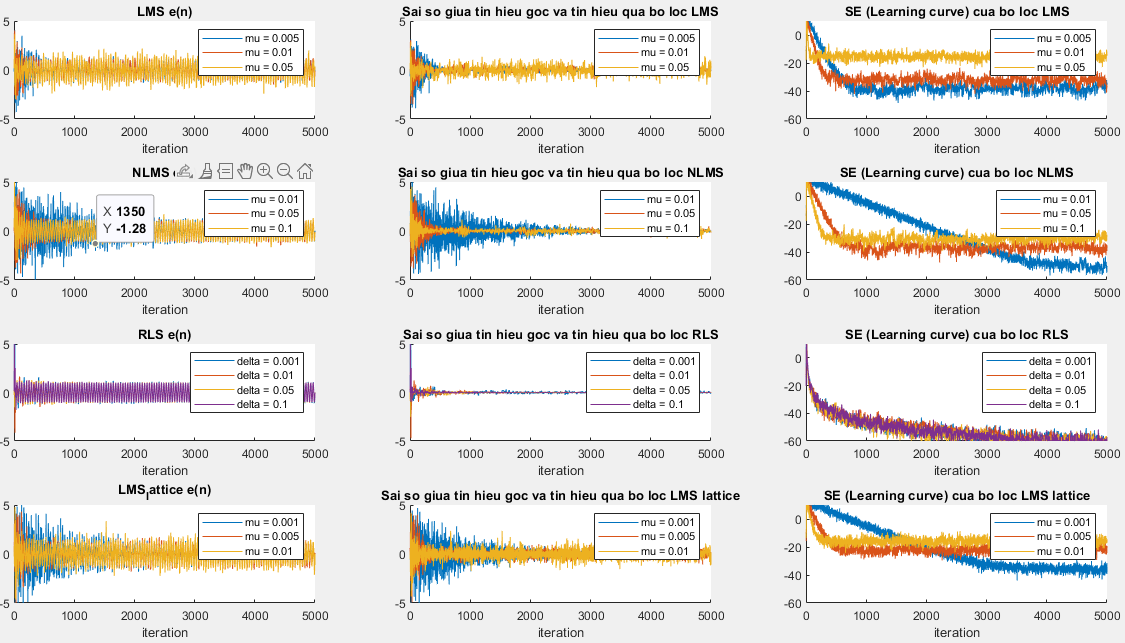


Figure 24 Output, error and learning curve of all filter structures

As expected, for most filter structure except for RLS transversal filter, the higher the step-size value, the faster the filter reach steady state but also the filter have higher error. Only in the RLS transversal filter, the filter reach steady state very fast and also the mean error does not change much with very different step-size value.

About the filters performance, in this situation, the RLS transversal filter perform the best with very fast convergence speed (reach steady state after only ~ 30 update iteration) and mean square error of ~60dB after reach steady state. The LMS transversal filter and the NLMS transversal filter have about the same error reduction performance and convergence speed (~600 samples to reach -40dB), but with different learning rate parameter. And finally, the LMS lattice filter have the lowest convergence rate and highest steady state error.

Through this experiment, we can conclude that the RLS adaptive filter is superior to other adaptive filter algorithm in both convergence speed and error performance. However, this test is for a simple sinusoidal signal. Performance comparison with more complex signal will be tested in the next experiment.

It is also advised to use LMS filter in case of having limit computational power, as the RLS algorithm is more complex and in real time testing it is approximately 3 times slower than LMS and NLMS algorithm.

## Experiment 3: Filtering human speech signal

In this experiment, a human speech signal with length of 178000 sample is used as the original signal. A more complex noise with is a combination of a White Gaussian noise plus a delay of the same noise by 10 and 25 sample. At signal rate of 16000, its similar to a 1.5ms delay. The desired signal for the filter is the combination of the original speech signal and this more complex noise, while the reference signal remains the original White Gaussian noise signal.

Because of the delay in noise component of the signal, the filter order of all filter types must be higher to keep all the noise components in the desired signal correlated with the reference signal. In this experiment, a filter order of 100 is used.

The following table show the parameters of the filters used in this experiment:

|  |  |
| --- | --- |
| Filter | Parameters |
| LMS transversal filter | mu  = 0.005 |
| NLMS transversal filter | mu = 0.05  theta = 0.01 |
| RLS transversal filter | delta = 0.1  lambda = 0.999 |
| LMS lattice filter | mu = 0.0001 |

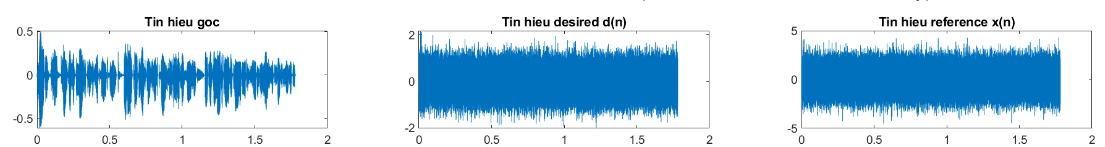


Figure 25 Original speech signal and input signals for the adaptive filter

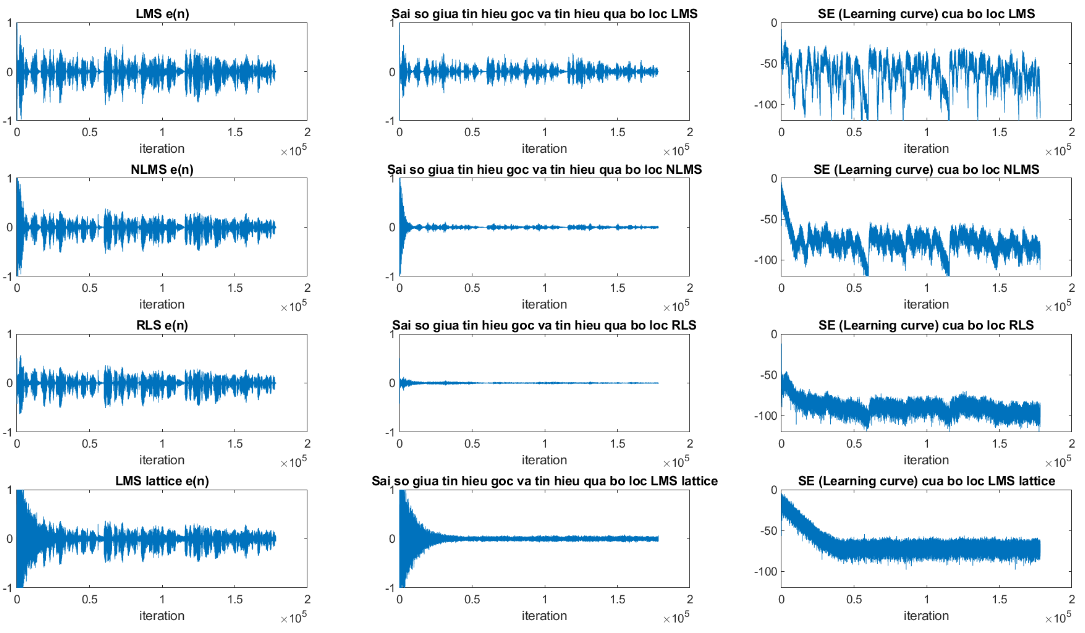


Figure 26 Output, error and learning curve off all filter structures (Experiment 3)

Analyzing the result, we can observe that the RLS transversal filter and the LMS lattice filter perform better in this scenario as both these filter are less affect by the change in amplitude of the original signal.

While the LMS transversal filter and NLMS transversal filter can remove noise of signal when the amplitude of the speech signal is low, they struggle when the signal amplitude changes to higher, lead to as high as -50db in MSE value where there is high amplitude original signal.

About the RLS filter and the LMS lattice filter, they perform better throughout the whole signal with amplitude change, with MSE value are ~-85dB and ~-70dB respectively.

## Experiment 4:  Testing different noise scenarios

We have tested noise cancelation of adaptive filters with simple noise model and clean desired signal and noise, which mean original signal are not mixed into reference signal, delay of the noise or signal is low, or there is no uncorrelated noise in the signal. In this experiment, we will these scenarios to see how it affect the noise cancelation performance of the adaptive filters.

In the first scenarios, a base line test will be setup for comparison purpose. The following scenarios will demonstrate different noise and filter situation mentioned above. The original sound signal, filter order and filter parameters will be the same through different scenarios.

The original signal here is a sinusoidal signal with length of 5000 sample. All the noise added to the signal in different scenarios bellow result in a MSE value of ~0dB between the noisy signal and the original signal.

The following table show the parameters of the filters used in this experiment:

|  |  |
| --- | --- |
| **Filter** | **Parameters** |
| All | filter order M = 20 |
| LMS transversal filter | mu = 0.005 |
| NLMS transversal filter | mu = 0.05  theta = 0.01 |
| RLS transversal filter | delta = 0.1  lambda = 0.999 |
| LMS lattice filter | mu = 0.0001 |

### Normal Gaussian noise, clean signal

In the first scenarios, a simple White Gaussian noise signal is generated and added to the original signal. The same noise signal is also used as the reference input for the filters. As expected, all filters can separate the noise from the noisy signal successfully, with the MSE value hover around -40dB to -45dB.

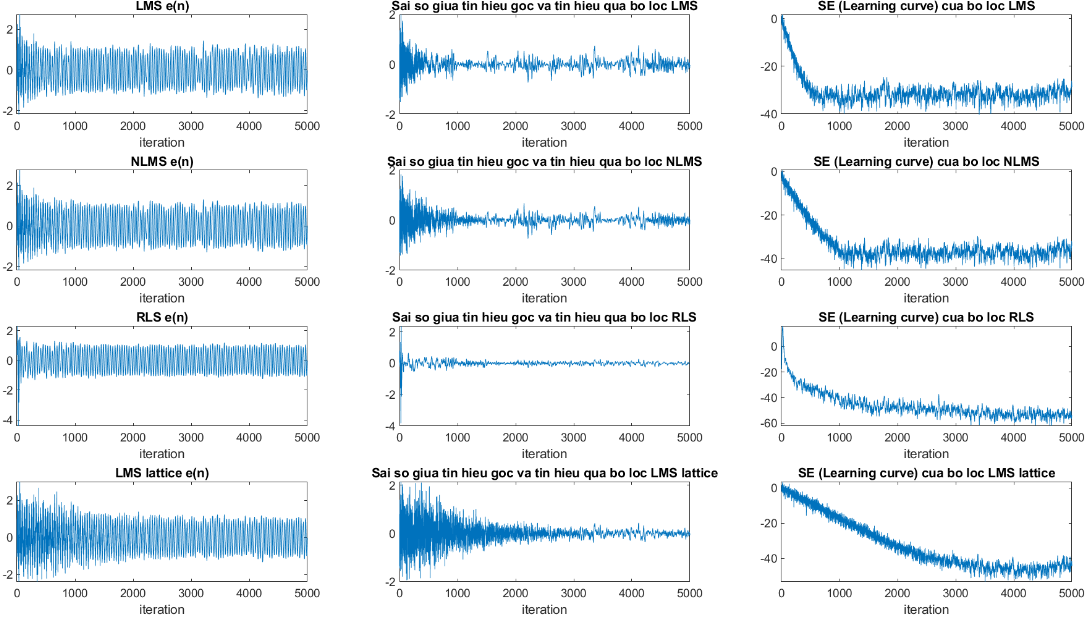


Figure 27 Output, error and learning curve off all filter structures (Experiment 4.1)

### Delay Gaussian noise, delay sample higher than filter order

In this scenarios, a part of the noise component in the noisy signal have a delay of 30 samples and amplitude of ¼ original noise source, which is higher than the 20 delay samples that the filter order structure provide. This lead to the part of the noise that got high delay became uncorrelated with the reference noise.

The result show that the filter performance drop significantly due to the filter cannot filtered out the delay noise component in the signal. All output of the filter now have MSE value of -20dB compare to the original signal.

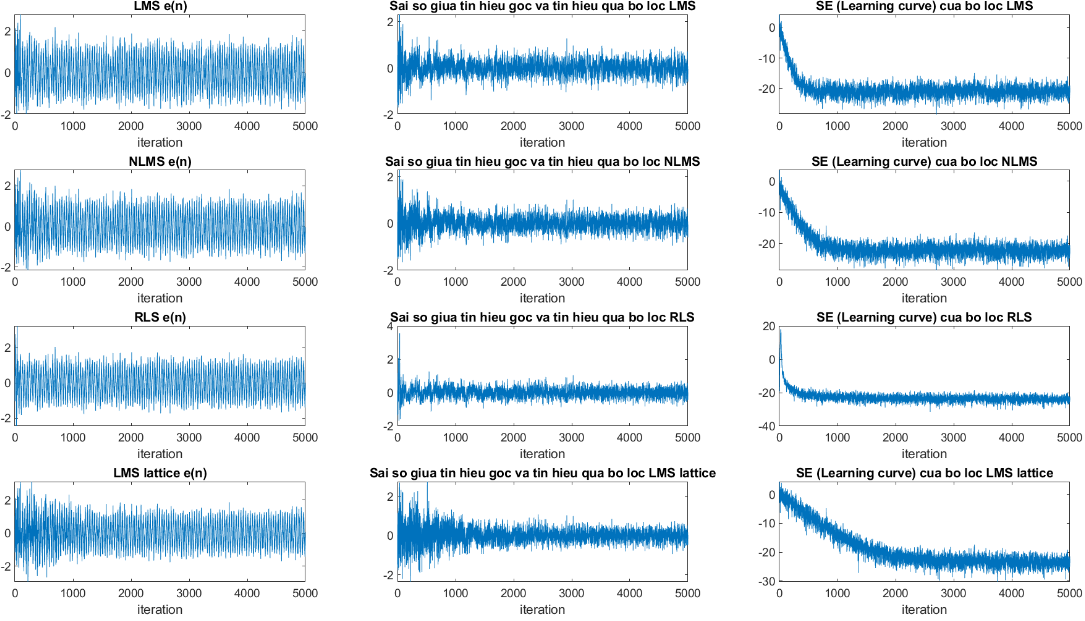


Figure 28 Output, error and learning curve off all filter structures(Experiment 4.2)

### Reference signal got part of the signal mixed in

In this scenarios, the reference signal of the filter, when it should be only noise, get part of the original signal mixed in. This also hurt the performance of the filters, which result in approximately -25dB MSE value for all filter types.

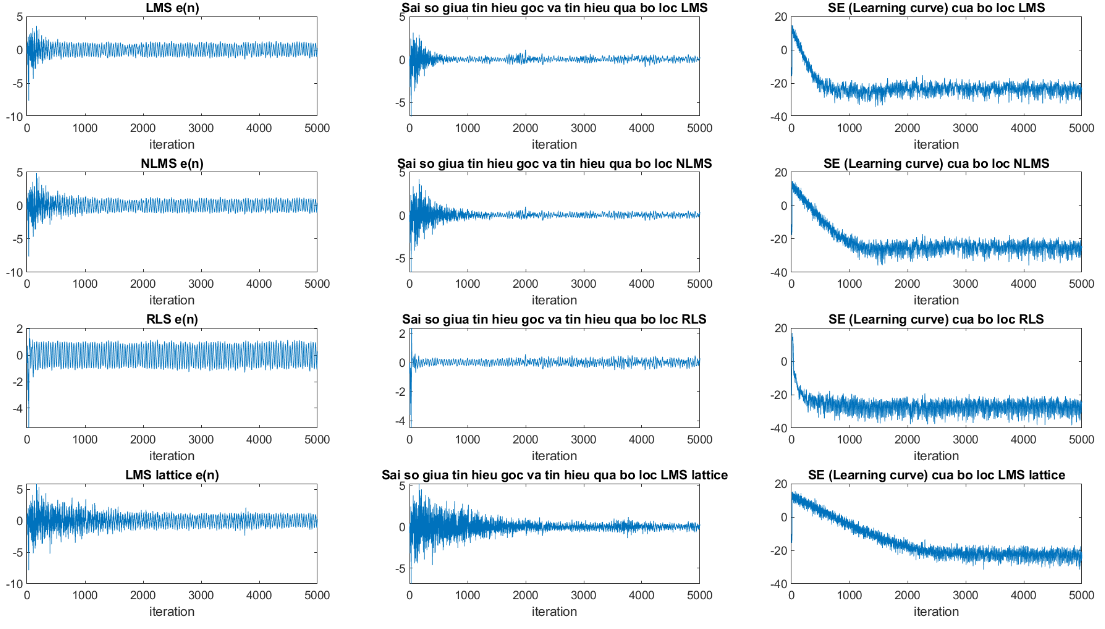


Figure 29 Output, error and learning curve off all filter structures (Experiment 4.3)

### Uncorrelated noise in desired signal

In this case, a part of the reference signal get a noise signal from difference source mixed in. This also result in the reduction in noise canceling performance at only -10dB MSE at all filter types.

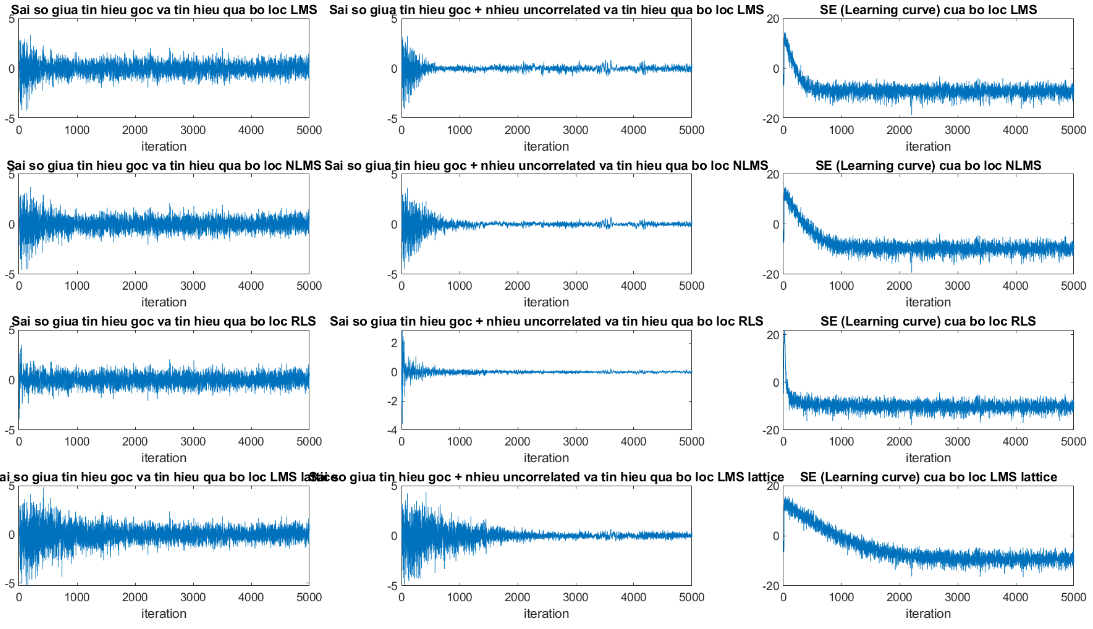


Figure 30 Output, error and learning curve off all filter structures (Experiment 4.4)

### Uncorrelated noise as the reference input

In the final scenarios, the reference noise signal is from another source and completely uncorrelated to the noise in the noisy signal. This result in the output signal worse than the original signal.

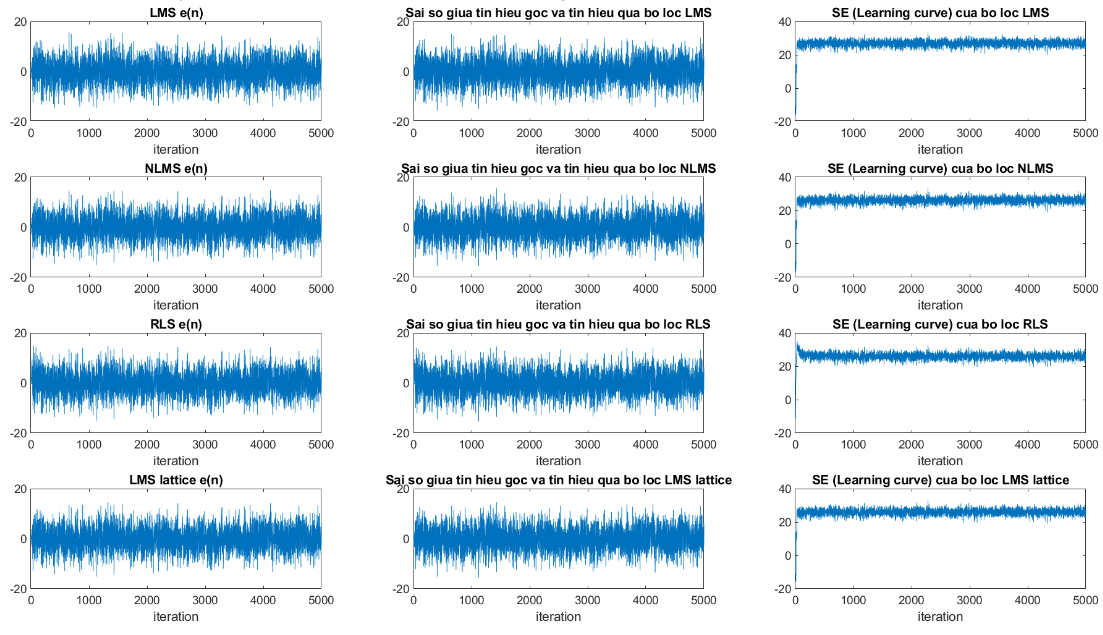


Figure 31 Output, error and learning curve off all filter structures (Experiment 4.5)

To sum up this experiment, all the filter perform as expected, following the theory in all different noise situations.

## Experiment 5: Adaptive noise cancellation without reference signal

In this final experiment, all the adaptive filter were putted in a real life scenarios, where there is no reference noise for a noisy signal. The signal is the real speech signal from experiment 3, and the noise here is a 50Hz sinusoidal wave with amplitude of 1, as following the theory, noise cancelation without reference signal only work with periodic noise.

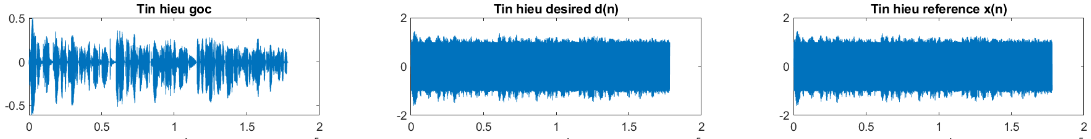


Figure 32 Original speech signal and inputs signal of the filters

In the first case, the reference signal is the desired signal delay by 200 sample. Filters are set up with filter order M = 50 and filter parameters the same as the previous experiment. We can observe that the error between the output e(n) of the filter system and the original signal is still high.

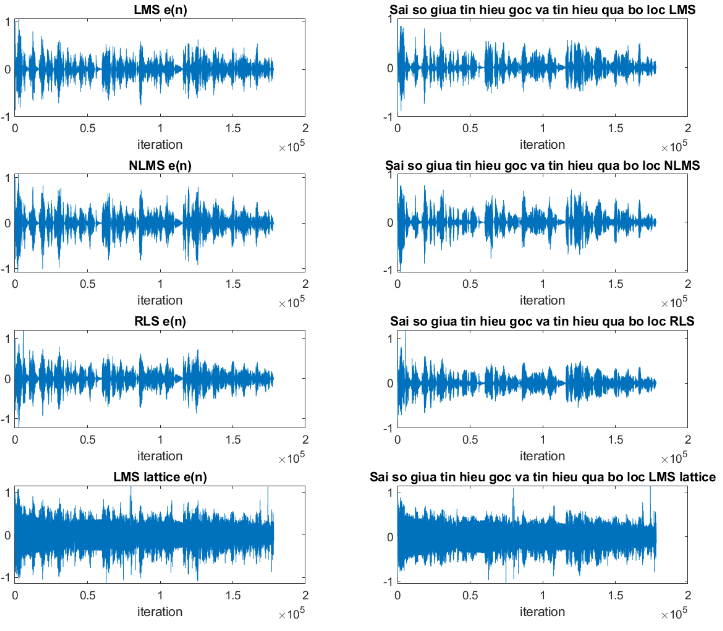


Figure 33 Output and error of adaptive filters with delay of 200 and filter order of 50

In the second case, we try a delay of 300 samples and a filter order of 200. The results are significant better when observing the error, with the RLS adaptive filter provide the best performance, then NLMS filter and the LMS lattice filter. Finally, the LMS transversal filter perform the worse.

It seem that increasing the order of the filter structure help the filter deal with more complex signal like human speech signal.

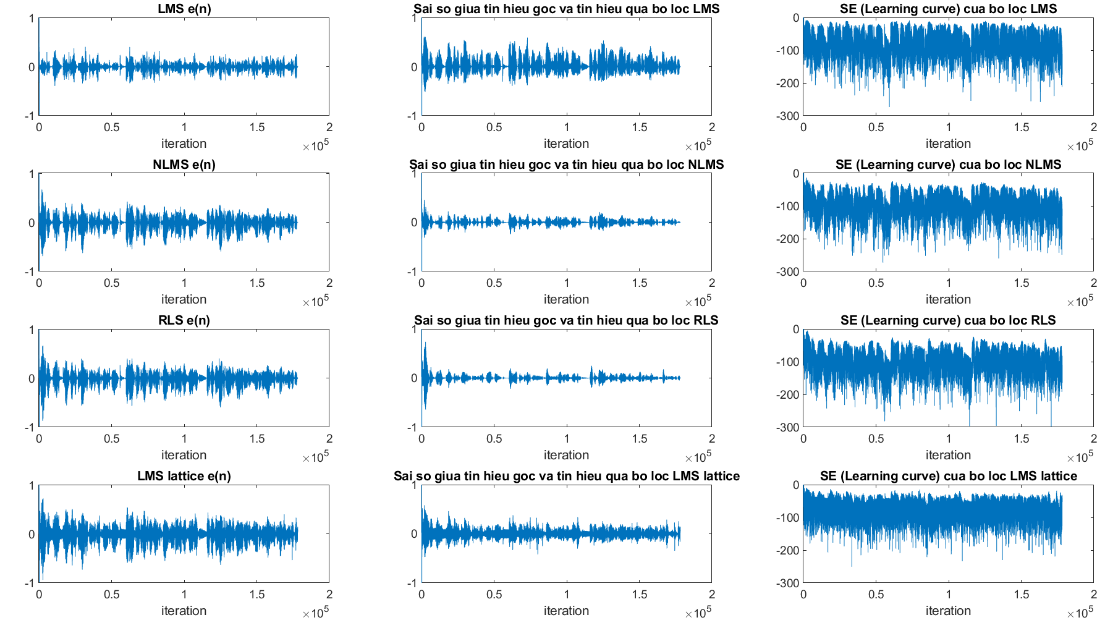


Figure 34 Output and error of adaptive filters with delay of 300 and filter order of 200

## Summary

Throughout the previous experiments that we have performed, we have tested different adaptive filter structure and algorithm, with different noise canceling scenarios. In the first experiment, we have tested the viability of the adaptive filter structure created in MATLAB environment. The second experiment help us observe the effect of changing filter parameters and choosing the optimal parameters for best noise canceling performance. In the third experiment, we apply the noise canceling application of the adaptive filters onto a real life, complex signal like a human speech to observe the performance. The fourth experiment help us to test the theory about signal and noise correlation and its effect on the noise canceling performance of the filters. And finally, the final experiment show the performance of different adaptive filter algorithm and structure in the real life application of noise canceling without the reference signal.

# Conclusion

In this project, we have learned about the theory and applications of adaptive filter in general, with different adaptive filter structure and adaptive filter algorithm. We have also successfully apply different adaptive filters structures on the MATLAB environment and test the performance of different structures and algorithm. This have help us to understand about digital signal processing in general, about adaptive filter and developing skill in MATLAB programming.

Code for this project is provided in the following Github repository:

https://github.com/huynhducmink/adsp\_final

# **Reference**

[1] Adaptive Filter Theory Fifth Edition – Simon Haykin

[2] Adaptive Filter Theory and Applications –Behrouz Farhang-Boroujeny

[3] Performance Analysis of Adaptive Noise Cancellation for Speech Signal – Pratibha et al.

[4] Adaptive Noise Cancellation – Aarti Singh

[5] Simulation and Performance Analysis of Adaptive Filter in Noise Cancellation – Raj et al.

[6] Overview of Adaptive Filters and Applications – mathworks.com