UNIVERSITY
OF GÄVLE

FACULTY OF ENGINEERING AND SUSTAINABLE DEVELOPMENT
Department of Electronics, Mathematics and Natural Sciences

Implementation of adaptive filtering algorithms for noise cancellation

Tanja Lampl

2020

Student thesis, Advanced level (Master degree), 15 HE
Electronics
Master Programme in Electronics/Automation (online)

Supervisor: José Chilo (PhD)
Examiner: Niclas Björsell (PhD)

Abstract

This paper deals with the implementation and performance evaluation of adaptive filtering algorithms for noise cancellation without reference signal. Noise cancellation is a technique of estimating a desired signal from a noise-corrupted observation. If the signal and noise characteristics are unknown or change continuously over time, the need of adaptive filter arises. In contrast to the conventional digital filter design techniques, adaptive filters do not have constant filter parameters, they have the capability to continuously adjust their coefficients to their operating environment. To design an adaptive filter, that produces an optimum estimate of the desired signal from the noisy environment, different adaptive filtering algorithms are implemented and compared to each other. The Least Mean Square LMS, the Normalized Least Mean Square NLMS and the Recursive Least Square RLS algorithm are investigated. Three performance criteria are used in the study of these algorithms: the rate of convergence, the error performance and the signal-to-noise ratio SNR. The implementation results show that the adaptive noise cancellation application benefits more from the use of the NLMS algorithm instead of the LMS or RLS algorithm.

Table of contents

1 Introduction 1
1.1 Problem 1
1.2 Motivation 3
1.3 Goal 4
1.4 Agenda 2030 6
1.5 Structure 7
2 Theory 8
2.1 Adaptive Filter 8
2.1.1 FIR Adaptive filters 10
2.1.2 IIR Adaptive filters 12
2.2 Adaptive filtering algorithms 14
2.2.1 Least Mean Square - LMS 16
2.2.2 Normalized LMS - NLMS 17
2.2.3 Recursive Least Square - RLS 18
2.3 Adaptive noise cancellation 20
2.3.1 Adaptive noise cancellation with reference signal 21
2.3.2 Adaptive noise cancellation without reference signal 23
2.4 Comparison criteria 25
2.4.1 Rate of convergence 27
2.4.2 Error performance 28
2.4.3 Signal-to-noise ratio SNR 29
3 Process and results 30
3.1 Input parameters 30
3.2 Practical implementation 33
3.2.1 The LMS solution 34
3.2.2 The NLMS solution 38
3.2.3 The RLS solution 42
4 Discussion 46
5 Conclusions 51
References 52
List of Figures 55
List of Tables 56
List of Abbreviations 57
Appendix A – Matlab Code A1

1 Introduction

The first chapter of this thesis provides an overview about the topic. The problem and the goal of the work are described. The topic of the thesis arises from the need to improve an industrial measurement application.

1.1 Problem

In real-time digital signal processing applications, there are many situations in which useful signals are corrupted by unwanted generated signals that are classified as noise. Noise can occur randomly or as white noise with an even frequency distribution or as frequency-dependent noise. The term noise includes not only thermal or flicker noise, but all disturbances, either in stimuli, environment or components of sensors and circuits [1]. Noisy data may arise from a variety of internal and external sources.

For example, the data may have been derived using noisy sensors or may represent a useful signal component that has been corrupted by transmission over a communication channel [2]. The thesis addresses the problem of an industrial measurement application used for fault detection, in which a desired signal d(n) has to be estimated from a noise-corrupted observation $x(n) = d(n) + v_1(n)$. The information bearing signal d(n) that is recorded by a sensor is corrupted by additive broadband noise $v_1(n)$, which is illustrated in Figure 1. Thermal noise is generated by the amplifiers in the sensor.

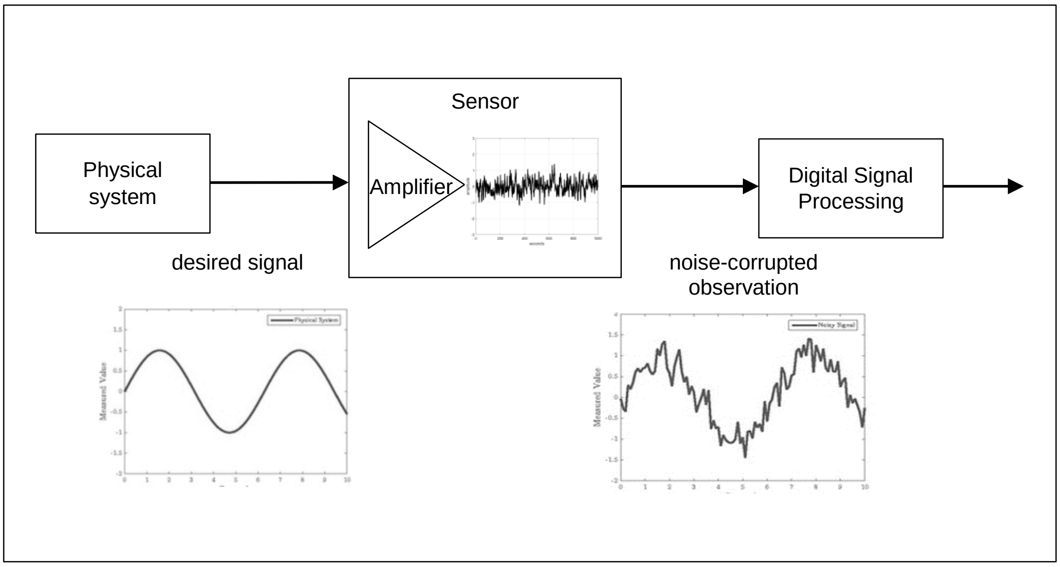

Figure 1: Noise-corrupted observation in the measurement application

The measurement system can be seen as black box as there is no information about the noise available, because there is no direct access to the noise source nor a way of isolating the noise from the useful signal. For the enhancement of the measurement system, it is necessary to extract the desired signal from the noise-corrupted observation by means of digital signal processing DSP to be then analyzed and processed.

A common method to remove the noise from a corrupted signal is applying a linear filtering method. When the signal of interest and the noise are in separate frequency bands, a classical digital filter such as low-pass filter, high-pass filter or band-pass filter can be used to extract the desired signal. The filter passes frequencies contained in the signal and rejects the frequency band occupied by the noise. The use of these filters only makes sense in very simple and idealized environments as they are rarely optimal to get the best estimate of the signal [3].

Another method is based on optimization theory and includes the Wiener filter and the Kalman filter. The approach of the Wiener filter is to minimize the mean-square error that is defined as the difference between the desired signal and the actual filter output. The design of a Wiener filter requires a priori information about the statistics of the data to be processed and cannot be used for non-stationary processes [4]. A major drawback of linear filtering, including Wiener and Kalman filters, is that the signal and noise characteristics must be known to determine the filter coefficients.

In the measurement application, the signal of interest become contaminated by noise occupying the same band of frequency. It results in a spectral overlap between the desired narrowband signal and the broadband noise. Furthermore, if the signal statistics are not known beforehand or they change over time, the coefficients of the filter cannot be specified in advance and makes the use of linear filters inadequate. The change in signal and noise characteristics in real-time requires the utilization of filter algorithms that converge rapidly to the unknown environment [5].

1.2 Motivation

Adaptive noise cancellation is the approach used for estimating a desired signal d(n) from a noise-corrupted observation $x(n) = d(n) + v_1(n)$. Usually 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 $v_2(n)$ can be filtered and subtracted from the primary input to obtain the signal estimate $\hat{d}(n)$. As the measurement system is a black box, no reference signal that is correlated with the noise is available. The principle of noise cancellation can be seen in the next figure.

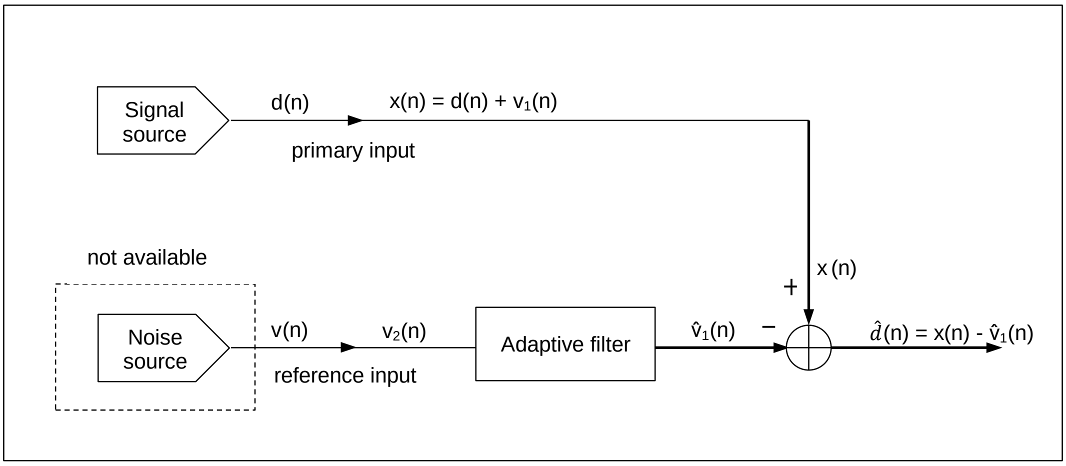

Figure 2: Principle of noise cancellation (Source: modified from [9])

To estimate the desired signal, adaptive filters can be implemented. As the name suggests, adaptive filters are filters with the ability of adaptation to an unknown environment. These filters can be used in applications where the input signal is unknown or not necessarily stationary. In contrast to the conventional digital filter design techniques, adaptive filters do not have constant filter coefficients [6]. Adaptive filters have the capability to adjust their impulse response to filter out the correlated signal in the input. They require modest or no a priori knowledge of the signal and noise characteristics [5].

An adaptive filter is composed of two parts, the digital filter and the adaptive algorithm. There are different types of adaptive filtering algorithms. A distinction is made between least mean-square algorithms and recursive least square algorithms. Recursive least square is an adaptive filter algorithm that recursively finds the coefficients that minimize a weighted linear least squares cost function relating to the input signals. This approach is different from the least mean-square algorithm that aim to reduce the mean-square error [7].

1.3 Goal

For the enhancement of the measurement system, adaptive noise cancellation without reference signal is realized. This paper is going to compare the performance of different adaptive filtering algorithms to get the best estimate of the desired signal from the noise-corrupted observation. The desired signal is received by the sensor, that is connected to a microcontroller. To extract the desired signal from the noisy process, test series are performed, where the desired signal is a sinusoid with unit amplitude that is observed in the presence of additive noise.

Figure 3 shows the measurement enhancement system, which represents a digitized, noise-corrupted signal that contains a desired narrowband signal and noise that occupies all the frequency range. The electrical signals generated by the sensor are digitized by the analog-to-digital converter ADC. After the analog-to-digital conversion, the digitized noisy signal x(n), where n is the sample number, is enhanced by means of digital signal processing DSP using adaptive noise cancellation.

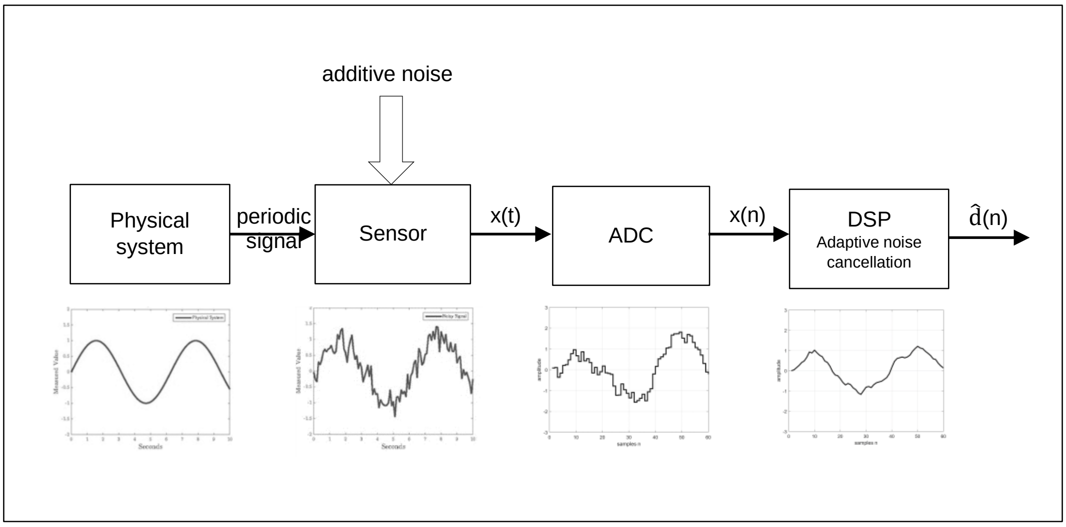

Figure 3: Measurement enhancement system

The goal is to design an optimum adaptive filter that produces the best estimate of the desired signal from the noisy environment and converges as fast as possible to its steady state. Optimal filters are optimum because they are designed based on optimization theory to minimize the mean-square error or least squares error between a processed signal and a desired signal, or equivalently provides the best estimation of a desired signal from a measured noisy signal [8].

In order to find the optimum filter, different adaptive filter concepts will be implemented and compared to each other in terms of the efficiency of noise cancellation. The Least mean-square LMS, the Normalized Least mean-square NLMS and the Recursive Least square RLS algorithm will be investigated. Three performance criteria are used in the study of these algorithms: the rate of convergence, the error performance and the signal-to-noise ratio SNR.

1.4 Agenda 2030

The Agenda 2030 for Sustainable Development is a roadmap for shaping global economic progress in harmony with social justice and within Earth's ecological limits. The Sustainable Development Goal 9 is based on three interconnected pillars: infrastructure, industry and innovation. These pillars all share the goal of achieving socially inclusive and environmentally sustainable economic development [22].

This thesis also takes the importance and necessity of achieving the Sustainable Development Goal 9 set out in the Agenda 2030 into account. Although, the work has no direct impact on building resilient infrastructure or promoting inclusive and sustainable industrialization, the focus of the work is on scientific research and innovation, in particular on researching innovative ways of solving technical challenges, that also contribute to sustainable development.

1.5 Structure

This work is focused on the implementation of different adaptive filtering algorithms for noise cancellation without reference signal to improve the industrial measurement system. In order to achieve the understanding of the proposed solution, the thesis includes a theoretical and practical part. The theoretical part of the thesis gives an introduction to the adaptive filtering theory and the practical part contains the practical implementation of an adaptive noise canceller without reference signal using the LMS, NLMS and RLS algorithm.

In Chapter 2, an introduction to adaptive filters is given and the differences between the different adaptive filtering algorithms are described mathematically. Furthermore, the approach of noise cancellation with and without reference signal is discussed and the comparison criteria are described in detail. The adaptive filtering theory described by Monson H. Hayes in the book "Statistical Digital Signal Processing and Modeling" [9] is used as main reference source.

Chapter 3 deals with the implementation of the different adaptive filtering algorithms in Matlab and the presentation of the performance results. Matlab will be used to view and filter the signals. Using Matlab will help to optimize the filter coefficients and it is useful for determining whether the filter is working properly. In this chapter the simulation results of each adaptive filtering algorithm are presented.

In Chapter 4, the experimental results of the adaptive noise cancellation using the LMS, NLMS and RLS algorithms are discussed and compared against each other. Three performance criteria are used in the study of these algorithms: the rate of convergence, the error performance and the signal-to-noise ratio SNR.

Chapter 5, the last chapter, includes a summary of the present work and the performance results.

2 Theory

In this chapter the theory behind the adaptive filtering process is described in greater detail. Mathematical connections are described in order to be able to compare the different adaptive filtering algorithms for noise cancellation. The differences between noise cancellation with and without reference signal are outlined and the comparison criteria are described.

2.1 Adaptive Filter

An adaptive filter consists of two distinct parts: a digital filter with adjustable coefficients $W_n(z)$ and an adaptive algorithm which is used to adjust or modify the coefficients of the filter [10]. The adaptive filter can be a Finite Impulse Response FIR filter or an Infinite Impulse Response IIR filter. The block diagram of an adaptive filter can be seen in the following figure.

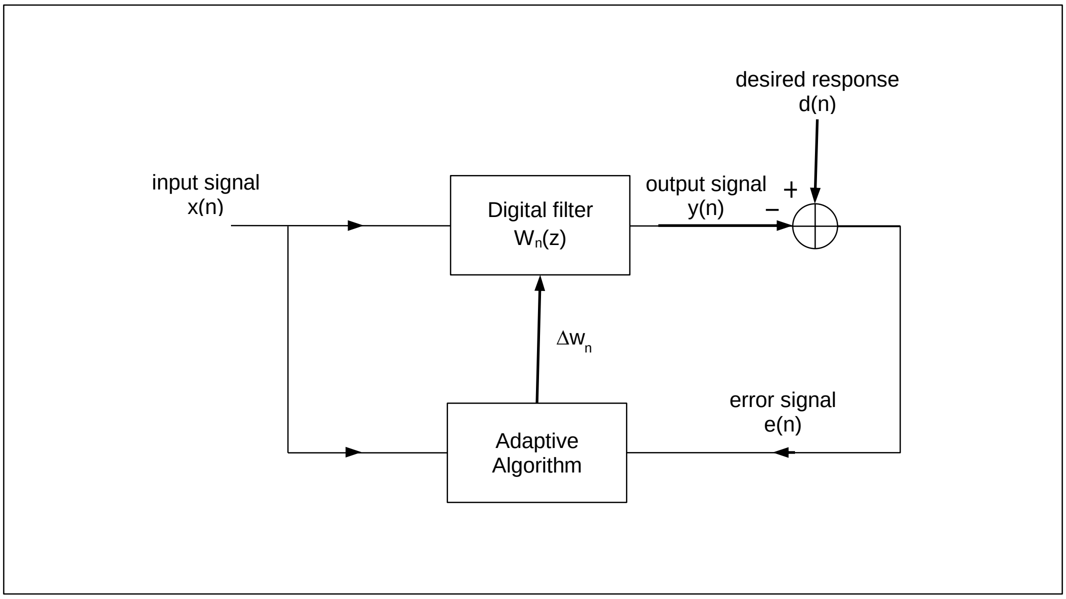

Figure 4: Block diagram of an adaptive filter (Source: modified from [9])

The filtering process involves the computation of the filter output y(n) in response to the filter input signal x(n). The filter output is compared with the desired response d(n), generating an error estimation e(n), as shown in Figure 4. The feedback error signal e(n) is then used by the adaptive algorithm to modify the adjustable coefficients of the filter w_n, generally called weight in order to minimize the error according to some optimization criterion.

An adaptive filter algorithm needs a so-called coefficient update equation to calculate new filter parameters for each sample. The coefficient update equation of the form

$$w_{n+1} = \mathrm{w}_n + \Delta w_n \tag{1}$$

is part of the adaptive algorithm, where Δw_n is a correction at sample n that is applied to the filter coefficients w_n to form a new set of coefficients w_{n+1} at sample n+1 [9].

The development of an adaptive filter involves defining how the correction Δw_n is to be formed. The primary goal is that the sequence of corrections should decrease the least squares or mean-square error. The adaptive filter should have the following properties according to Monson H. Hayes [9]:

- *"In a stationary environment, the adaptive filter should produce a sequence of corrections Δw_n, in such a way that w_n converges to the Wiener solution.*
- *It should not be necessary to know the signal statistics in order to compute Δw_n. The estimation of the statistics should be "built into" the adaptive filter.*
- *For nonstationary signals, the filter should be able to adapt to the changing statistics and "track" the solution as it evolves in time."*

The most important thing in the implementation of an adaptive filter is the requirement of the error signal e(n). Without the error signal, the filter could not adapt to the environment, since the error sequence determines how the filter coefficients should be adjusted. In some applications it is difficult to obtain the error sequence.

2.1.1 FIR Adaptive filters

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.

$$y(n) = \sum_{k=0}^{q} b_n(k)x(n-k) \tag{2}$$

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.

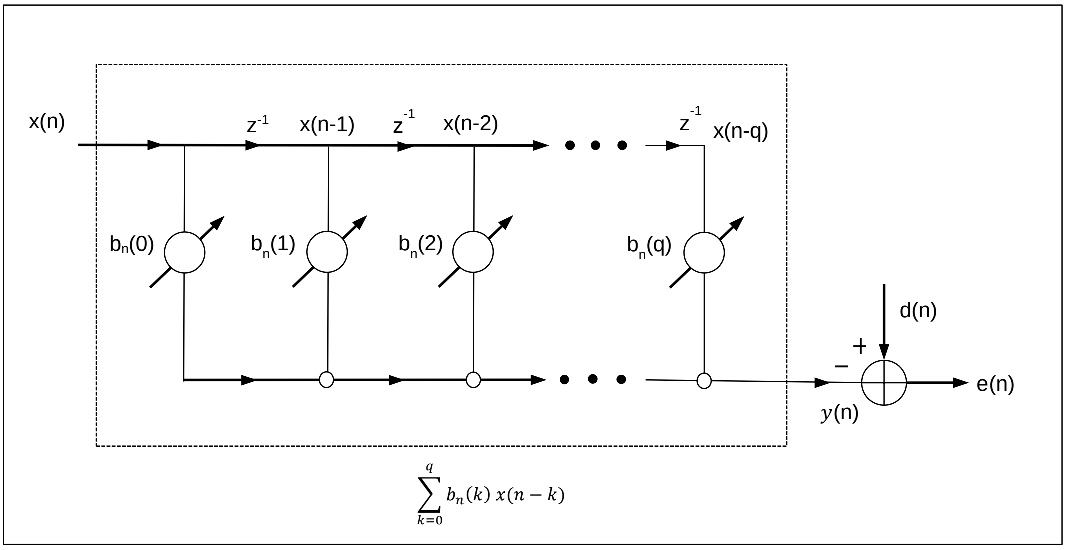

Figure 5: Block diagram of a direct-form FIR adaptive filter (Source: modified from [9])

In designing the FIR adaptive filter, the goal is to find the coefficient vector w_n 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 [9].

$$y(n) = \sum_{k=0}^{q} w_n(k)x(n-k) = w_n^T\, x(n) \tag{3}$$

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.

$$\xi(n) = E\{|e(n)|^2\} \tag{4}$$

The error signal in (5) is calculated from the difference between the filter output signal y(n) and the desired signal d(n).

$$e(n) = d(n) - y(n) = d(n) - w_n^T x(n) \tag{5}$$

To find the filter coefficients that minimize the mean-square error it is necessary to set the derivative $\xi(n)$ equal to zero with respect to w_n^*(k) for k = 0,1..., q and * represents the complex conjugate, which leads to the result

$$E\{e(n)\, x^*(n-k)\} = 0 \tag{6}$$

Substituting equation (5) in equation (6), it becomes [9]

$$E\left\{\left[d(n) - \sum_{k=0}^{q} w_n(k)x(n-k)\right]x^*(n-k)\right\} = 0 \tag{7}$$

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.

2.1.2 IIR Adaptive filters

The impulse response of an IIR filter has an infinite number of coefficients. An IIR filter has feedback from output to input and in its output is a function of the previous output samples and the present and past input samples [11]. In (8) the linear constant coefficient difference equation of the IIR filter can be seen.

$$y(n) = \sum_{k=1}^{p} a_n(k)y(n-k) + \sum_{k=0}^{q} b_n(k)x(n-k) \tag{8}$$

where $a_n(k)$ and $b_n(k)$ are the coefficients of the adaptive filter at sample n. The output sample y(n) depends on past output samples y(n-k), as well as recent and past input samples x(n-k), that is known as the IIR filter's feedback. Shown in the following figure is the block diagram of an IIR adaptive filter.

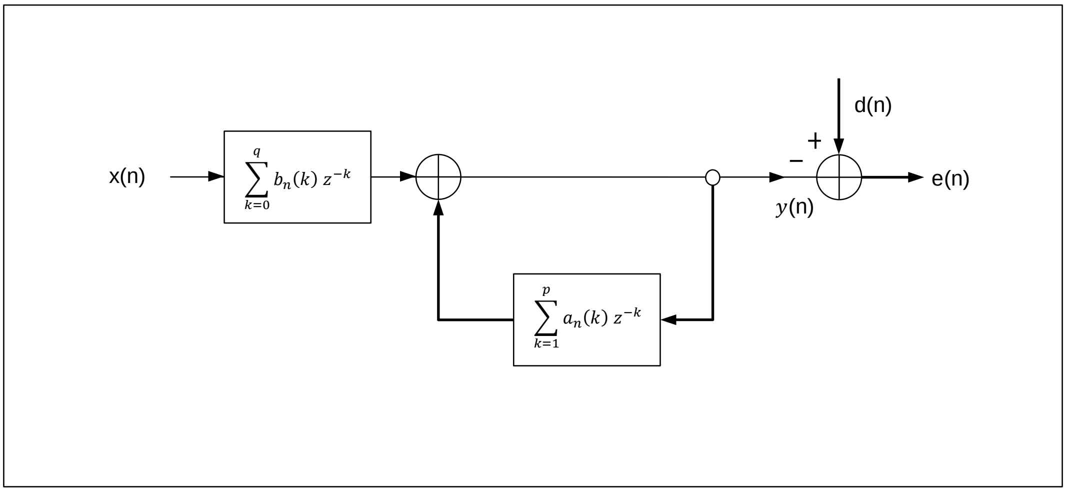

Figure 6: Block diagram of an IIR adaptive filter

In order to minimize the mean-square error $\xi(n) = E\{|e(n)|^2\}$, where e(n) is the difference between the desired process d(n) and the output of the adaptive filter y(n), it is necessary to define some vectors [9]. The vector of the filter coefficients Θ can be expressed as

$$\Theta = \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} a(1), a(2), \dots, a(p) \\ b(0), b(1), \dots, b(q) \end{bmatrix}^T \tag{9}$$

The data vector z(n) in (10) denotes the aggregate data vector and contains the past output samples as well as recent and past input samples.

$$z(n) = \begin{bmatrix} y(n-1) \\ x(n) \end{bmatrix} = \begin{bmatrix} y(n-1,), y(n-2,), \dots, y(n-p) \\ x(n-p)\,, x(n), \dots, x(n-q) \end{bmatrix}^T \tag{10}$$

The output of the filter in (11) can be expressed in terms of the filter coefficients Θ and the data vector z(n)

$$y(n) = a_n^T y(n-1) + b_n^T x(n) = \Theta^T z(n) \tag{11}$$

With the feedback coefficients $a_n(k)$ the mean-square error is no longer quadratic, $\xi(n)$ may have multiple local minima and maxima [9],[12]. The gradient vector must be set to zero

$$E\{e(n)\nabla e^*(n)\} = 0 \tag{12}$$

Since e(n) = d(n) – y(n) then

$$E\{e(n)\nabla y^*(n)\} = 0 \tag{13}$$

Differentiating $\nabla y^*(n)$ with respect to $a^*(k)$ and $b^*(k)$ results in

$$\begin{aligned} \frac{\delta y^*(n)}{\delta a^*(k)} &= y^*(n-k) + \sum_{k=1}^{p} a^*(k) * \frac{\partial y^*(n-k)}{\partial a^*(k)} \qquad ; k = 1,2,\dots,p \\ \frac{\delta y^*(n)}{\delta b^*(k)} &= y^*(n-k) + \sum_{k=0}^{q} b^*(k) * \frac{\partial y^*(n-k)}{\partial b^*(k)} \qquad ; k = 0,1,\dots,q \end{aligned} \tag{14}$$

Finally, combining the equations leads to

$$\begin{aligned} E\left\{e(n)\left[y(n-k) + \sum_{k=1}^{p} a(k) * \frac{\partial y(n-k)}{\partial a(k)}\right]^*\right\} &= 0 \qquad ; k = 1,2,\dots,p \\ E\left\{e(n)\left[x(n-k) + \sum_{k=0}^{q} b(k) * \frac{\partial y(n-k)}{\partial b(k)}\right]^*\right\} &= 0 \qquad ; k = 0,1,\dots,q \end{aligned} \tag{15}$$

Since the equations are nonlinear, they are difficult to solve for the optimum filter coefficients. Due to the nonlinearity, the solution may not be unique, therefore the solution will rather correspond to a local rather than a global minimum. The strength of the IIR filters comes from the feedback procedure, but the disadvantage of it is that the IIR filter becomes unstable or poor in performance if it is not well designed. A common form of the recursive IIR filter is the lattice structure.

2.2 Adaptive filtering 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 algorithms. An overview of the different types can be seen in Figure 7.

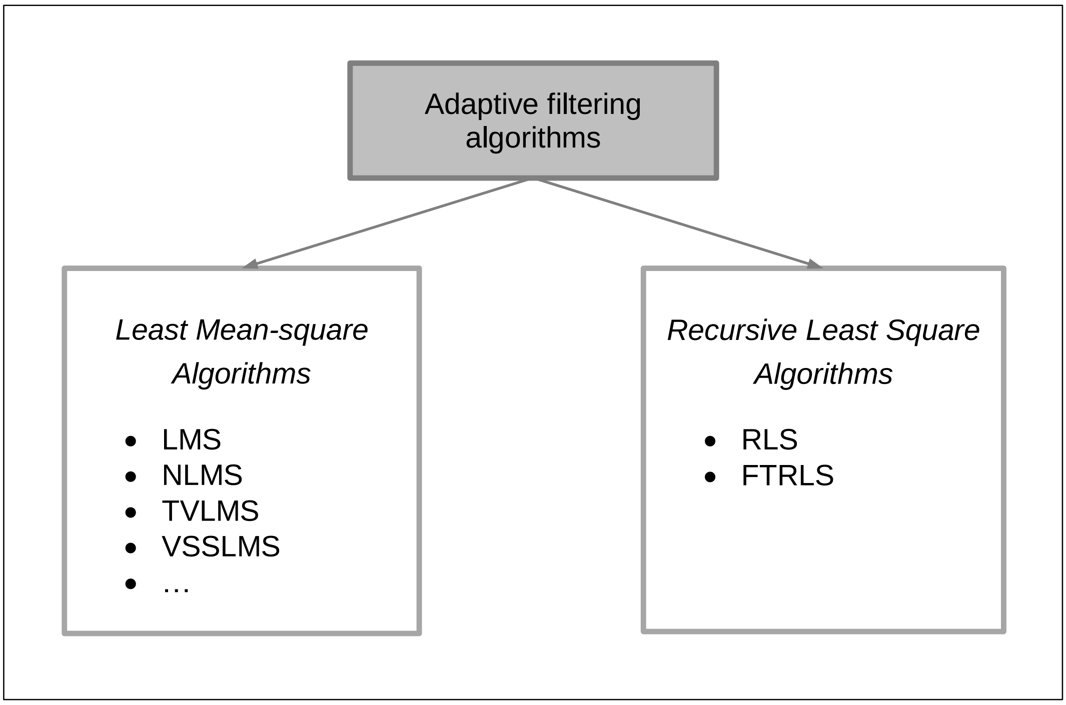

Figure 7: Types of adaptive filtering algorithms

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 [13]. 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 [7]. 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 [13].

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.

2.2.1 Least Mean Square - LMS

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 [9]. 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

$$w_{n+1} = w_n + \mu E\{e(n)x^*(n)\} \tag{16}$$

The steepest descent algorithm shows a practical limitation in that the expectation $E\{\mathrm{e(n)x^*(n)}\}$ is generally unknown. Therefore, it must be replaced with an estimate such as the sample mean

$$\hat{E}\{e(n)x^*(n)\} = \frac{1}{L}\sum_{l=0}^{L-1} e(n-l)\,x^*(n-l) \tag{17}$$

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

$$w_{n+1} = w_n + \frac{\mu}{L}\sum_{l=0}^{L-1} e(n-l)\,x^*(n-l) \tag{18}$$

In (19) the ensemble average is estimated using a one-point sample mean (L=l).

$$\hat{E}\{e(n)x^*(n)\} = e(n)x^*(n) \tag{19}$$

Finally, combining the equations leads to the update equation in (20) that is known as LMS algorithm

$$w_{n+1} = w_n + \mu e(n)x^*(n) \tag{20}$$

where w_n 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:

$$0 < \mu < \frac{2}{\lambda_{max}} \tag{21}$$

where λ_{max} is the largest eigenvalue of the autocorrelation matrix R_x [9].

2.2.2 Normalized LMS - NLMS

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 w_n 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

$$0 < \mu < \frac{2}{tr(R_x)} \tag{22}$$

The bound in the above equation can be calculated from

$$tr(R_x) = (p+1)E\{|x(n)|^2\} \tag{23}$$

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

$$0 < \mu < \frac{2}{(p+1)E\{|x(n)|^2\}} \tag{24}$$

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

$$\hat{E}\{|x(n)|^2\} = \frac{1}{p+1}\sum_{k=0}^{p}|x(n-k)|^2 \tag{25}$$

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

$$0 < \mu < \frac{2}{x^H(n)\,x(n)} \tag{26}$$

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

$$\mu(n) = \frac{\beta}{x^H(n)\,x(n)} = \frac{\beta}{|\,x(n)|^2} \tag{27}$$

where β is a normalized step-size in the range of $0 < \beta < 2$. Replacing μ in the LMS weight vector update equation with μ(n) leads to the normalized LMS algorithm NLMS in (28).

$$w_{n+1} = w_n + \beta\frac{x^*(n)}{|\,x(n)|^2}e(n) \tag{28}$$

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

2.2.3 Recursive Least Square - RLS

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 $w_n(k)$ at sample n that minimize the weighted least squares error

$$\varepsilon(n) = \sum_{i=0}^{n} \lambda^{n-1} |e(i)|^2 \tag{29}$$

The RLS algorithm involves the recursive updating of the vector w_n 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

$$z(n) = P(n-1)x^*(n) \tag{30}$$

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

$$g(n) = \frac{1}{\lambda + x^T x(n) z(n)} z(n) \tag{31}$$

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 $0 < \lambda \leq 1$. Incorporating these definitions, the autocorrelation matrix P(n) results in

$$P(n) = \frac{1}{\lambda}[P(n-1) - g(n)z^H(n)] \tag{32}$$

In (33) 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 w_{n-1} to the new data vector x(n).

$$\alpha(n) = d(n) - w_{n-1}^T\, x(n) \tag{33}$$

Finally, combining the equations leads to the exponentially weighted Recursive Least Squares RLS algorithm

$$w_n = w_{n-1} + \alpha(n) g(n) \tag{34}$$

The special case of $\lambda = 1$ is referred to as the growing window RLS algorithm, where all previous errors are considered of equal weight in the total error [13]. Since the RLS algorithm involves the recursive updating of the vector w_n and the inverse autocorrelation matrix P(n), the initial conditions for both of these terms are required [9].

2.3 Adaptive noise cancellation

Adaptive noise cancellation is the approach used for estimating a desired signal d(n) from a noise-corrupted observation x(n) = d(n) + v_1(n). 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 v_2(n) is adaptively filtered and subtracted from the primary input to obtain the signal estimate $\hat{d}$(n) [15].

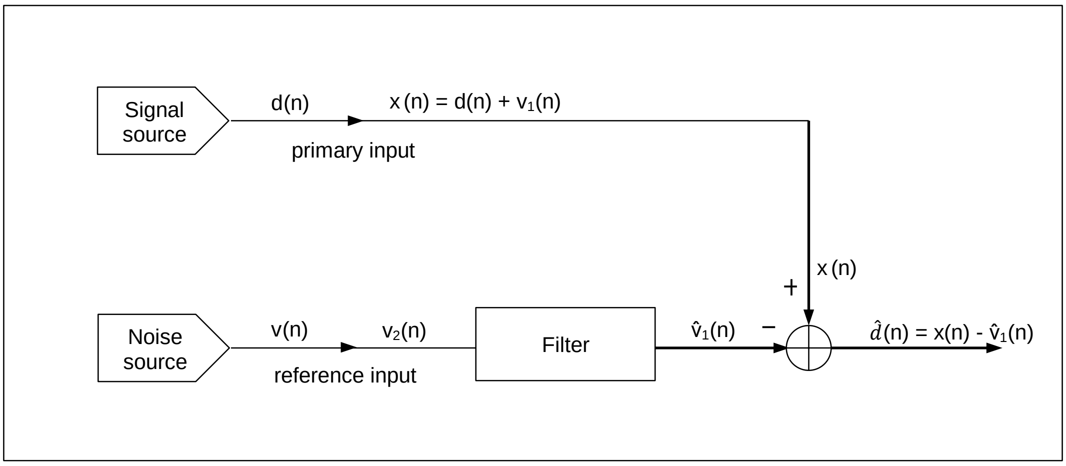

Figure 8: Principle of noise cancellation (Source: Figure 2)

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 $\hat{v}_1$(n) should be an exact replica of v_1(n). 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 [16].

Without any information about d(n) or v_1(n) it is not possible to separate the signal from the noise. When a reference signal v_2(n) is given, that is correlated with the primary noise v_1(n), it can be used to estimate the desired signal d(n) from the noisy observation [9]. 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 v_1(n) and v_2(n) 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.

2.3.1 Adaptive noise cancellation with reference signal

A desired signal d(n) is transmitted over a channel to a sensor that also receives a noise $v_1(n)$ uncorrelated with the signal. The primary input to the canceller is a combination of both signal and noise $x(n) = d(n) + v_1(n)$. Here, the reference signal $v_2(n)$ is known, which is uncorrelated with the signal but correlated with the noise $v_1(n)$. In that case, a second sensor receives the reference signal and provides the input to the adaptive filter. Figure 9 shows the principle of adaptive noise cancellation with reference signal.

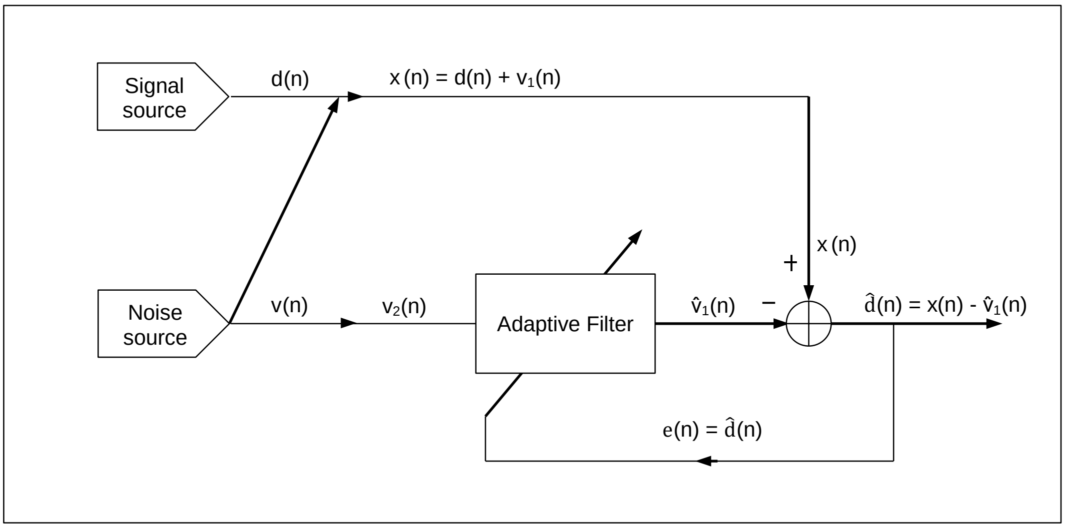

Figure 9: Noise cancellation with reference signal (Source: modified from [9])

The reference signal $v_2(n)$ is filtered to produce the noise estimate $\hat{v}_1(n)$, that is close to the primary noise $v_1(n)$. The output of the adaptive filter is then subtracted from the primary input to produce the signal estimate $\hat{d}(n)$ that is the best fit in the least squares sense to the desired signal d(n). This objective is accomplished by feeding the signal estimate $\hat{d}(n)$ 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 $e(n) = \hat{d}(n)$ [9].

The system output $\hat{d}(n)$ is

$$\hat{d}(n) = d(n) + v_1(n) - \hat{v}_1(n) \tag{35}$$

Squaring, we obtain

$$\hat{d}(n)^2 = d(n)^2 + [v_1(n) - \hat{v}_1(n)]^2 + 2 * d(n) * [v_1(n) - \hat{v}_1(n)] \tag{36}$$

Taking expectations of both sides and realizing that d(n) is uncorrelated with $\hat{v}_1(n)$.

$$E\{\hat{d}(n)^2\} = E\{d(n)^2\} + E\{[v_1(n) - \hat{v}_1(n)]^2\} + 2 * E\{d(n) * [v_1(n) - \hat{v}_1(n)]\}$$

$$= E\{d(n)^2\} + E\{[v_1(n) - \hat{v}_1(n)]^2\} \tag{37}$$

The signal power $E\{\mathrm{d(n)}^2\}$ will be unaffected as the filter is adjusted to minimize $E\{\hat{\mathrm{d}}(\mathrm{n})^2\}$

$$\min E\{\hat{d}(n)^2\} = E\{d(n)^2\} + \min E\{[v_1(n) - \hat{v}_1(n)]^2\} \tag{38}$$

When the filter is adjusted to minimize the output noise power $\mathrm{E}\{\hat{\mathrm{d}}(\mathrm{n})^2\}$, the output noise $\mathrm{E}\{[\mathrm{v}_1(\mathrm{n}) - \hat{\mathrm{v}}_1(\mathrm{n})]^2\}$ 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 $v_2(n)$ is uncorrelated with d(n), it follows that minimizing the mean-square error $\mathrm{E}\{|\mathrm{e(n)}|^2\}$ is equivalent to minimizing $\mathrm{E}\{|\mathrm{v}_1(\mathrm{n}) - \hat{\mathrm{v}}_1(\mathrm{n})|^2\}$. In other words, the output of the adaptive filter is the minimum mean-square estimate of $v_1(n)$, then it follows that e(n) is the minimum mean-square estimate of d(n).

Since

$$[\hat{d}(n) - d(n)] = [v_1(n) - \hat{v}_1(n)] \tag{39}$$

This is equivalent to causing the output $\hat{\mathrm{d}}(\mathrm{n})$ to be a best least squares estimate of the signal d(n) [9].

2.3.2 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 $x(n) = d(n) + v_1(n)$. The delayed signal $x(n\text{-}n_0)$ is used as the reference signal to eliminate the interference. The principle of the noise cancellation without reference signal is illustrated in Figure 10.

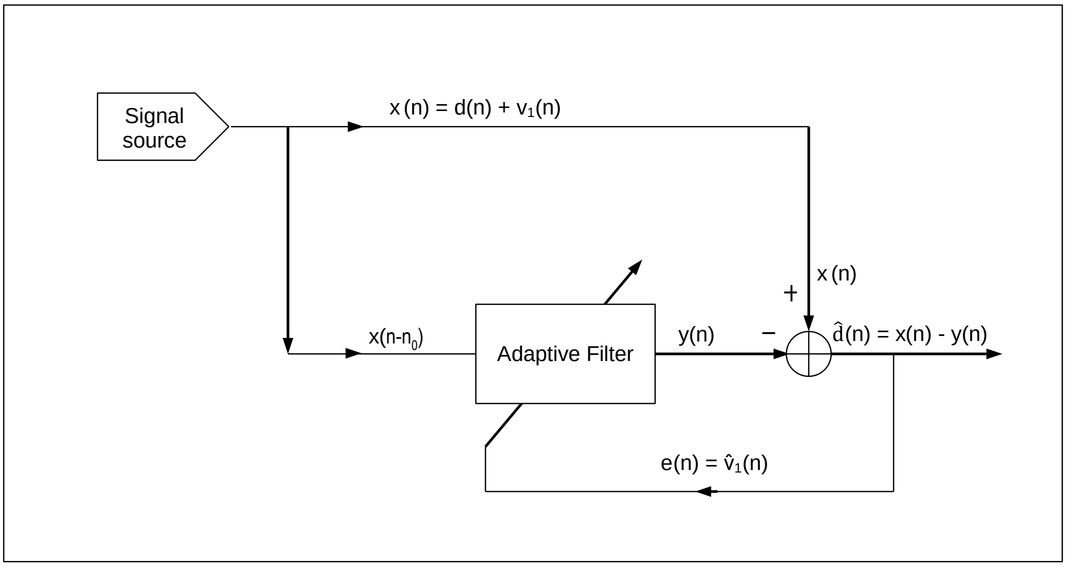

Figure 10: Noise cancellation without reference signal (Source: modified from [9])

The fixed delay n_0 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 y(n).

The system output e(n) is

$$e(n) = d(n) + v_1(n) - y(n) \tag{40}$$

Squaring, we obtain

$$e(n)^2 = v_1(n)^2 + [d(n) - y(n)]^2 + 2v_1(n) * [d(n) - y(n)] \tag{41}$$

Taking expectations of both sides

$$E\{e(n)^2\} = E\{v_1(n)^2\} + E\{[d(n) - y(n)]^2\} + 2E\{v_1(n) * [d(n) - y(n)]\} \tag{42}$$

In addition, the input to the adaptive filter is $x(n-n_0)$

$$E\{v_1(n) * y(n)\} = E\{[v_1(n) * d(n-k)]\} + E\{[v_1(n) * v_1(n-k)]\} \qquad (43)$$

Since the interference $v_1(n)$ is uncorrelated with d(n) as well as with $v_1(n-k)$, it follows that minimizing the mean-square error $E\{|e(n)|^2\}$ is equivalent to minimizing $E\{|d(n) - y(n)|^2\}$, the mean-square error between d(n) and the output of the adaptive filter y(n).

$$\min E\{e(n)^2\} = \min E\{v_1(n)^2\} + \min E\{[d(n) - y(n)]^2\} \qquad (44)$$

Under the assumption that d(n) is a broadband process and the delay n_0 is greater than the decorrelation time, the delayed process $x(n-n_0)$ is uncorrelated with the noise $v_1(n)$, but correlated with d(n). This results in an adaptive filter that produces the minimum mean-square estimate of the broadband process d(n) and the input error signal e(n) corresponds to an estimate of the noise $v_1(n)$ [9].

2.4 Comparison criteria

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.

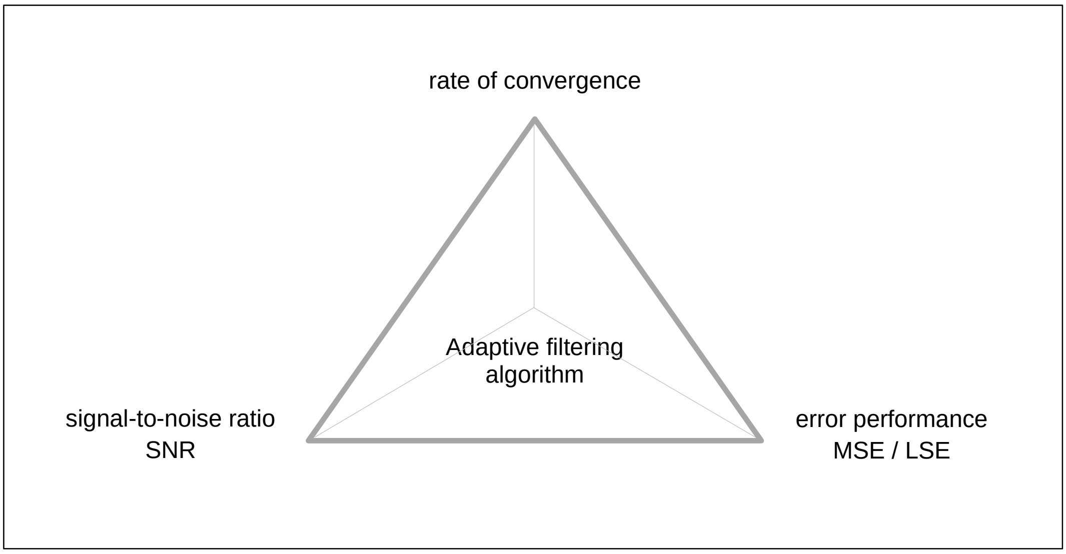

Figure 11: Comparison criteria for performance evaluation

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 [17].

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 [18]. 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.

2.4.1 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 [4]. 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.

- Convergence of the LMS adaptive algorithm

  The convergence characteristics of the LMS adaptive algorithm depends on two factors: the step-size μ and the eigenvalue spread of the autocorrelation matrix $\chi(R_x)$. The step-size μ must lie in a specific range

  $$0 < \mu < \frac{2}{\lambda_{max}} \tag{45}$$

  where λ_{max} is the largest eigenvalue of the autocorrelation matrix R_x. 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 [19].

- Convergence of the NLMS adaptive algorithm

  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

  $$\mu(n) = \frac{\beta}{x^H(n)\, x(n)} = \frac{\beta}{|\, x(n)|^2} \tag{46}$$

  where β is a normalized step-size with $0 < \beta < 2$. 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.

- Convergence of the RLS adaptive algorithm

  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 $0 < \lambda \le 1$.

2.4.2 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.

- Minimum mean-square error MSE

  The criterion of the LMS and NLMS algorithm is the minimum mean-square of the error signal. The MSE is defined as the ensemble average of the squared error sequence, denoted as

$$\xi(n) = E\{|e(n)|^2\} \tag{47}$$

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

$$\mathcal{M} = \frac{\xi_{ex}(\infty)}{\xi_{min}} \tag{48}$$

  A trade-off between a low rate of convergence and a small mean-square error or misadjustment is necessary, because when the step-size μ increases, the rate of convergence decreases, but the MSE increases [8].

- Minimum least squares error LSE

  In the RLS algorithm two different errors must be considered, the a priori estimation error $\xi(n)$ 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 $\varepsilon(n)$ based on e(n), not $\xi(n)$.

$$\varepsilon(n) = \sum_{i=0}^{n} \lambda^{n-1} |e(i)|^2 \tag{49}$$

  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 $\xi(n)$ as the error of interest. For the comparison, the ensemble-average of the a priori estimation error $\xi(n)$ in (50) has to be computed [20].

$$E\{|\xi(n)|^2\} \tag{50}$$

  The learning curve will have the same general form as the LMS algorithm.

2.4.3 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 (51) by the ratio of the signal power to the noise power and is often expressed in decibel.

$$SNR_{dB} = 10 \log_{10} \frac{S}{N} \tag{51}$$

In order to compare the different adaptive filtering algorithms in the efficiency of noise cancellation, the so-called improvement SNR level in (52) is used, which is the difference between the input and output SNR [21].

$$SNR_{imp} = SNR_{out} - SNR_{in} \tag{52}$$

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

$$SNR_{in} = 10 \log_{10} \frac{\sum_n x(n)^2}{\sum_n \mathrm{v}_1(\mathrm{n})^2} \tag{53}$$

where x(n) is the noise-corrupted signal and v_1 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.

$$SNR_{out} = 10 \log_{10} \frac{\sum_n y(n)^2}{\sum_n e(n)^2} \tag{54}$$

where y(n) is the output signal of the adaptive filter and e(n) 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 [21].

3 Process and results

Based on the theory of adaptive filtering examined in the previous chapter, the practical implementation of adaptive noise cancellation is carried out in Matlab. To improve the industrial measurement system, an adaptive noise canceller without reference signal is realized and the different adaptive filtering algorithms are implemented and compared against each other in terms of the efficiency of noise cancellation.

3.1 Input parameters

For a better comparison of the different adaptive filtering algorithms, simulation data are used to clearly present the results. Each adaptive filter gets the same primary input data containing the desired signal d(n) and the delayed process $x(n-n_0)$. In Chapter 2.1 the method of adaptive noise cancellation was examined in which a desired signal d(n) is to be estimated from a noise-corrupted observation.

$$x(n) = d(n) + v_1(n) \tag{55}$$

To extract the useful information from the noisy process, test series are performed, where the desired signal is a digitized sinusoid with unit amplitude, which can be seen in the following plot. In the simulation of the adaptive noise canceller the desired signal d(n) to be estimated is supposed to be a sinusoid

$$d(n) = sin(n\omega_0 + \phi) \tag{56}$$

with $\omega_0 = 0.05\pi$.

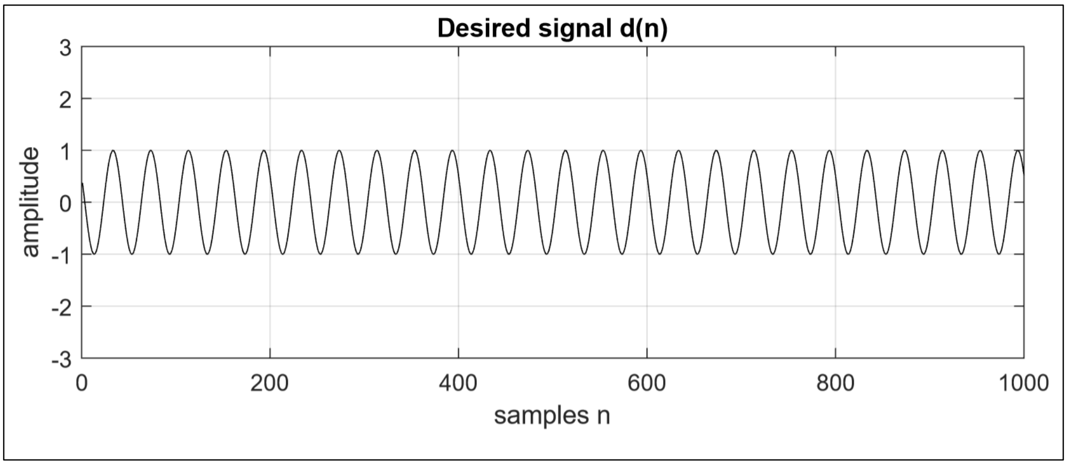

Figure 12: Desired signal d(n) that is to be estimated

The broadband noise signal comes from thermal noise generated by the amplifiers in the sensor. If the noise signal is uncorrelated there will be no information about future or past values and the desired signal cannot be estimated. Therefore, the noise sequence $v_1(n)$ is modelled as an autoregressive AR(1) process that is generated by the first-order difference equation

$$v_1(n) = 0.8v_1(n-1) + g(n) \tag{57}$$

where g(n) is a zero-mean, white Gaussian noise with a variance $\sigma^2 = 0.25$, that is uncorrelated with d(n). The autoregressive process is generated by filtering the white noise sequence in Matlab.

When filtering the white noise sequence, successive samples become correlated. The correlated noise signal can now be predicted by using the adaptive noise cancellation method. The noise-corrupted observation including the equations (55), (56) and (57) is generated in Matlab with the following sequence of commands:

```
%% Noise-corrupted observation: x(n) = d(n) + v1(n)
d = sin([1:N]*0.05*pi);     % desired signal
g = randn(1,N)*0.25;        % white Gaussian noise with a variance of
                            0.25
v1= filter(1,[1 -0.8],g);   % filtered white noise
x = d + v1;                 % noisy process
```

Shown in the next figure is a plot of 1000 samples of the desired sinusoid and the noisy-process $x(n) = d(n) + v_1(n)$.

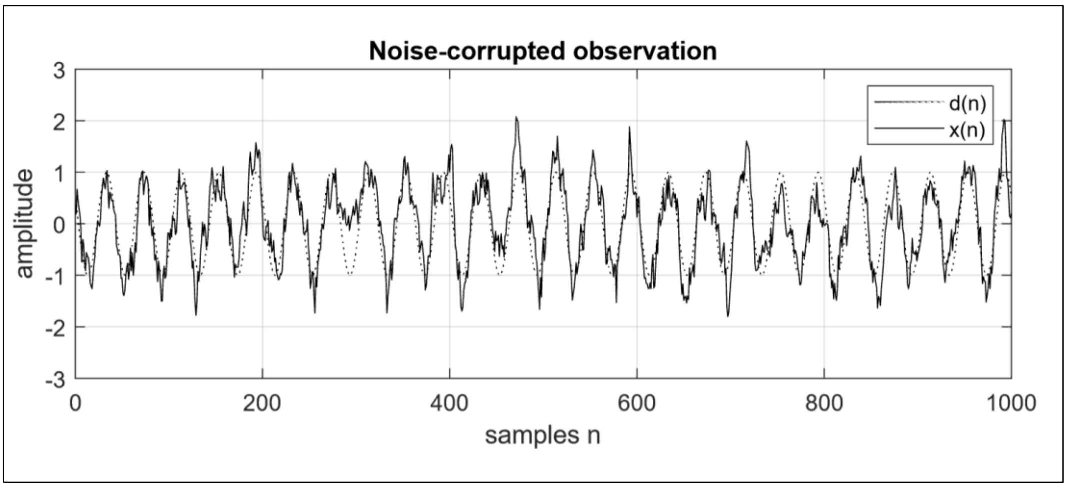

Figure 13: Noise-corrupted observation

The measurement system is a black box, in which no reference signal correlated with the noise is available. In that case, it is possible to derive a reference signal by delaying the noisy process x(n) = d(n) + v_1(n). The delayed process x(n-n_0) is correlated with d(n) and can be used as the reference signal to estimate the desired signal.

$$x(n - n_0) = x(n) \quad , n > n_0 \tag{58}$$

The reference signal in (58) is calculated in Matlab as follows, where n_0 equals 25 samples:

```
%% Reference signal
n0 = 25;                    % delay of 25 samples
len = N - n0;               % reduced vector length
x_del = zeros(N,1);         % create array of all zeros
% generate delayed signal
for i = 1:len
    x_del(i) = x(i+n0);
end
```

The reference signal is obtained by delaying x(n) by n_0 = 25 samples, which can be seen in the next plot.

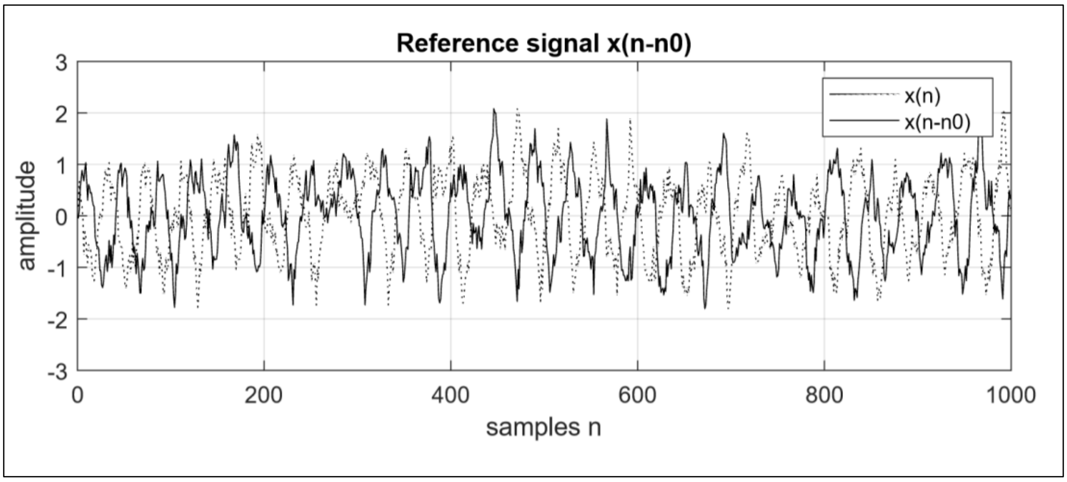

Figure 14: Reference signal x(n-n0)

The simulation of the different adaptive filtering algorithms is carried out using a 12th-order adaptive noise canceller. The other parameters for each algorithm result from the test series. For comparison, the values at which the different algorithms perform well in noise cancellation and show the same rate of convergence were selected.

3.2 Practical implementation

In this chapter the experimental results of each adaptive filtering algorithm implemented for adaptive noise cancellation without reference signal are presented. This is carried out through presenting images of diagrams representing the adaptive behavior of the different algorithms. Based on these results a discussion is carried out in Chapter 4. In order to be able to compare the discussed filtering methods against each other, some characteristics must be defined which can be evaluated for each algorithm.

For the comparison of the three different adaptive filtering algorithms the following performance characteristics were chosen:

• Rate of convergence

• Error performance

• Signal-to-noise ratio SNR

To study the rate of convergence and the error performance, ensemble-averaged learning curves are computed. The learning curves $J(n) = E\{|e(n)|^2\}$ are averaged over 100 independent runs. A low rate of convergence is desired, which means that the algorithm can quickly converge to its steady-state after a few iterations. The error performance can be evaluated from the steady-state error. The steady-state error can be estimated from the learning curves by averaging $\xi(n)$ over n after the algorithm has reached its steady state. The target is a low MSE or LSE value to achieve a proper performance.

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 output SNR has to be higher than the input SNR, which indicates the success of noise removal. The input SNR can only be estimated from the reference signal and therefore only the output SNR is computed. A large value of the output SNR is desirable, which indicates that the adaptive filter can remove a large amount of noise and produces an accurate estimate of the desired signal.

The filtering process involves the computation of the filter output y(n) in response to the reference signal $x(n-n_0)$. The filter output is compared with the desired response, generating an error estimation. That error is then used to modify the adjustable coefficients of the filter, generally called weight in order to minimize the error according to some optimization criterion. The Matlab code with all computations can be found in Appendix A.

3.2.1 The LMS solution

The adaptive noise cancellation using the LMS adaptive filtering algorithm was implemented in Matlab as described in the previous chapters. This algorithm modifies the filter coefficients in such a way that the error gets minimized in the mean-square sense. The LMS algorithm is summarized in the following table.

Inputs	x = delayed process x(n-n_0) to estimate d(n) d = desired signal d(n)
Outputs	y = filter output y(n) w= tap-weight vector update
Parameters	p = filter order μ = step-size parameter: $0 < \mu < \frac{2}{\lambda_{max}}$
Initialization	$w_0 = 0$
Computation	1. Filtering: $y(n) = w_n^T x(n)$ 2. Error Estimation: $e(n) = d(n) - y(n)$ 3. Tap-weight vector adaptation: $w_{n+1} = w_n + \mu e(n) x^*(n)$

Table 1: Summary of the LMS algorithm

The next figure depicts the results obtained by applying the LMS algorithm for adaptive noise cancellation without reference signal. The desired sinusoid, the noisy process x(n) and the estimate of the desired signal d(n) that is produced by the adaptive noise cancellation with a filter order of 12 and a step-size of $\mu = 0.002$ can be seen.

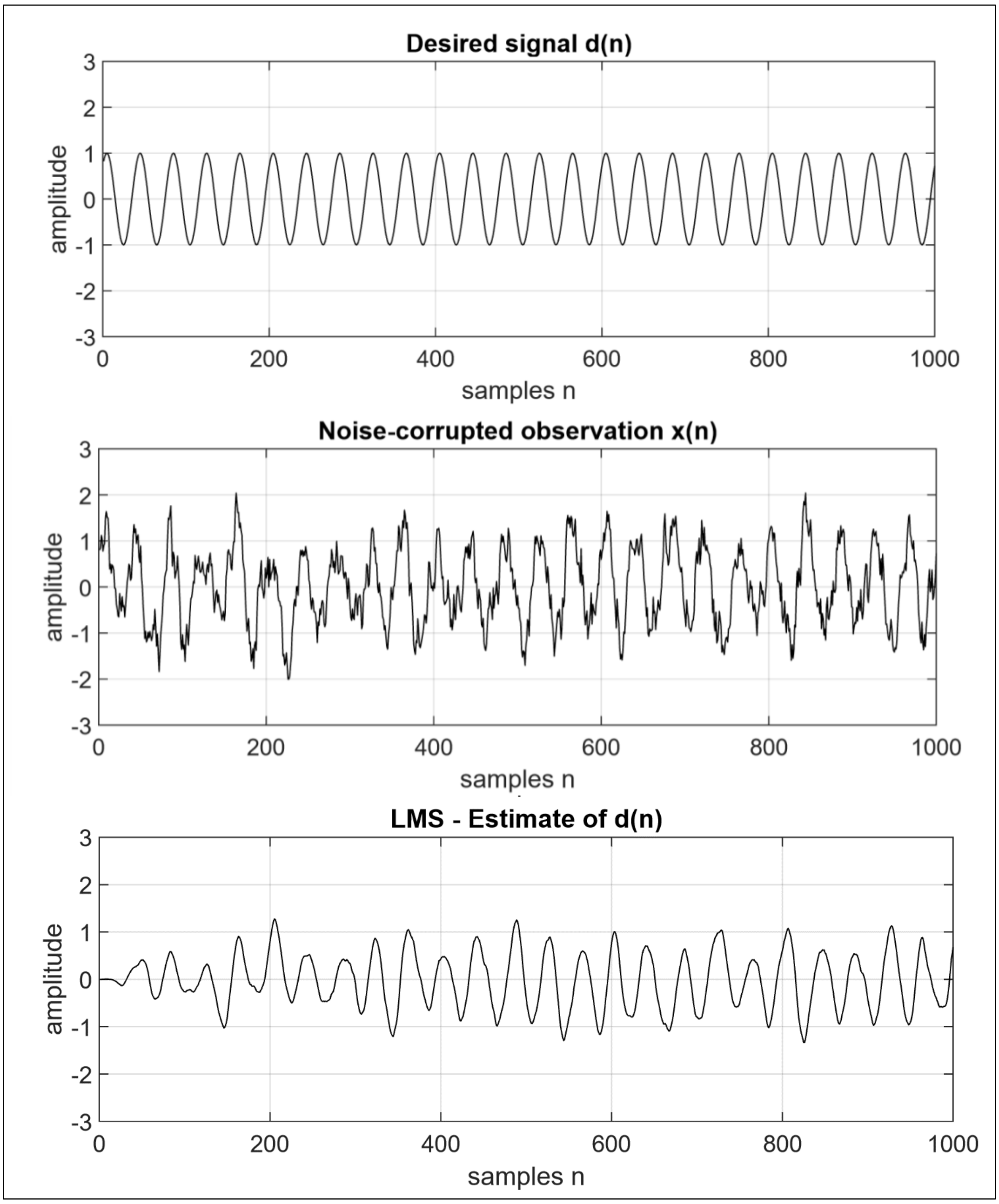

Figure 15: ANC without reference signal using LMS algorithm

Analyzing the figures, it can be carried out that the LMS algorithm shows not very good performance in noise cancellation. At the beginning, the amplitude of the signal estimate $\hat{d}(n)$ is very low compared with the desired signal and the shape of a sine wave cannot be reproduced. After about 100 samples, the adaptive filter can produce a more accurate estimate of d(n).

For the performance evaluation, the learning curve of the LMS algorithm was computed. The learning curve in the next figure shows the ensemble-averaged squared error versus the number of adaptation cycles. The ensemble averaging was performed over 100 independent runs of the experiment. The simulation of the LMS algorithm is carried out with different step-sizes. The values used for μ are 0.002 and 0.02.

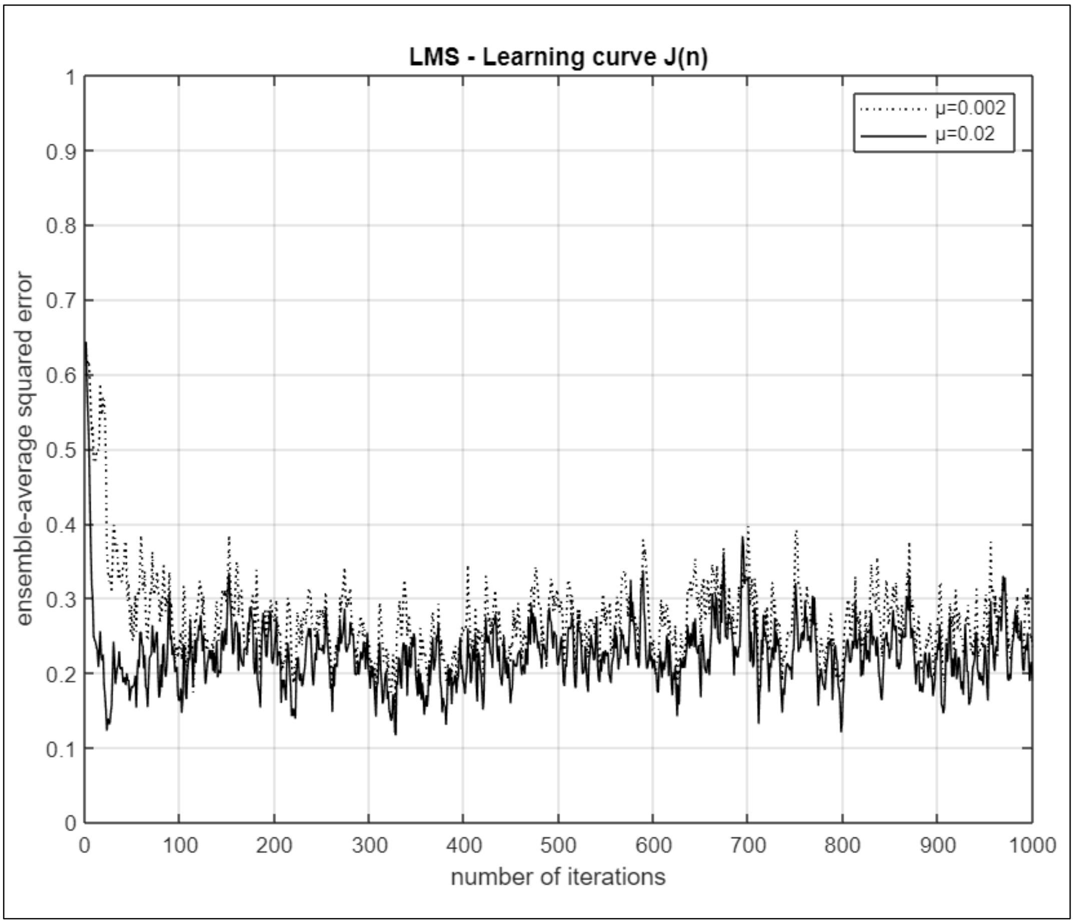

Figure 16: Learning curves of the LMS algorithm

- Rate of convergence

  The convergence behavior of the LMS algorithm can be determined from the learning curve and depends on the step-size parameter μ. The lower μ is, the higher is the rate of convergence, which means that the algorithm requires more adaptation cycles until it converges to its steady-state value. For a step-size of μ = 0.002, the algorithm requires more than 100 iterations to converge to its steady-state. A larger step-size of μ = 0.02 allows the algorithm to converge to steady-state conditions in approximately 50 adaptation cycles.

- Error performance

  The error performance is nearly the same for the different step-sizes. The cost function of the LMS algorithm aims to minimize the MSE.

  The estimated steady-state mean square error results in

$$\hat{\xi}(\infty)\begin{cases} 0.307 & ;\ for\ \mu\ =\ 0.002 \\ 0.295 & ;\ for\ \mu\ =\ 0.02 \end{cases}$$

  The steady-state value of the average squared error and hence the misadjustment decreases slightly with increasing step-size μ.

- Signal-to-noise ratio

  The signal-to-noise ratio in decibels was computed after applying the LMS algorithm. The output SNR is the ratio between the power of the filtered signal y(n) and the power of the noise e(n) at the output.

$$\mu\ =\ 0.002:\quad SNR_{out} = 10\log_{10}\frac{\sum_n y(n)^2}{\sum_n e(n)^2} = 6.73\ dB$$

$$\mu\ =\ 0.02:\quad SNR_{out} = 10\log_{10}\frac{\sum_n y(n)^2}{\sum_n e(n)^2} = 8.23\ dB$$

  The LMS algorithm with step-size μ = 0.002 produces an output SNR of 6.73 dB, which is lower compared with the output SNR when using a higher step-size value. A higher value of the output SNR indicates that a larger amount of noise could be removed from the noisy process.

3.2.2 The NLMS solution

The Normalized LMS algorithm can be viewed as the LMS algorithm with a normalized step-size parameter β. The input and output signals are the same as those described for the LMS algorithm. In the following table, a summary of the NLMS algorithm is presented.

Inputs	x = delayed process x(n-n_0) to estimate d(n) d = desired signal d(n)
Outputs	y = filter output y(n) w= tap-weight vector update
Parameters	p = filter order β = normalized step-size parameter: $0 < \beta < 2$
Initialization	$w_0 = 0$
Computation	1. Filtering: $$y(n) = w_n^T x(n)$$ 2. Error Estimation: $$e(n) = d(n) - y(n)$$ 3. Tap-weight vector adaptation: $$w_{n+1} = w_n + \beta \frac{x^*(n)}{\lvert x(n)\rvert^2} e(n)$$

Table 2: Summary of the NLMS algorithm

The results of the adaptive noise cancellation by using the NLMS algorithm are plotted. The desired sinusoid, the noisy process x(n) and the estimate of the desired signal d(n) that is produced by the 12^{th}- order adaptive noise canceller with the normalized step-size parameter of $\beta = 0.25$ are presented in the next figure.

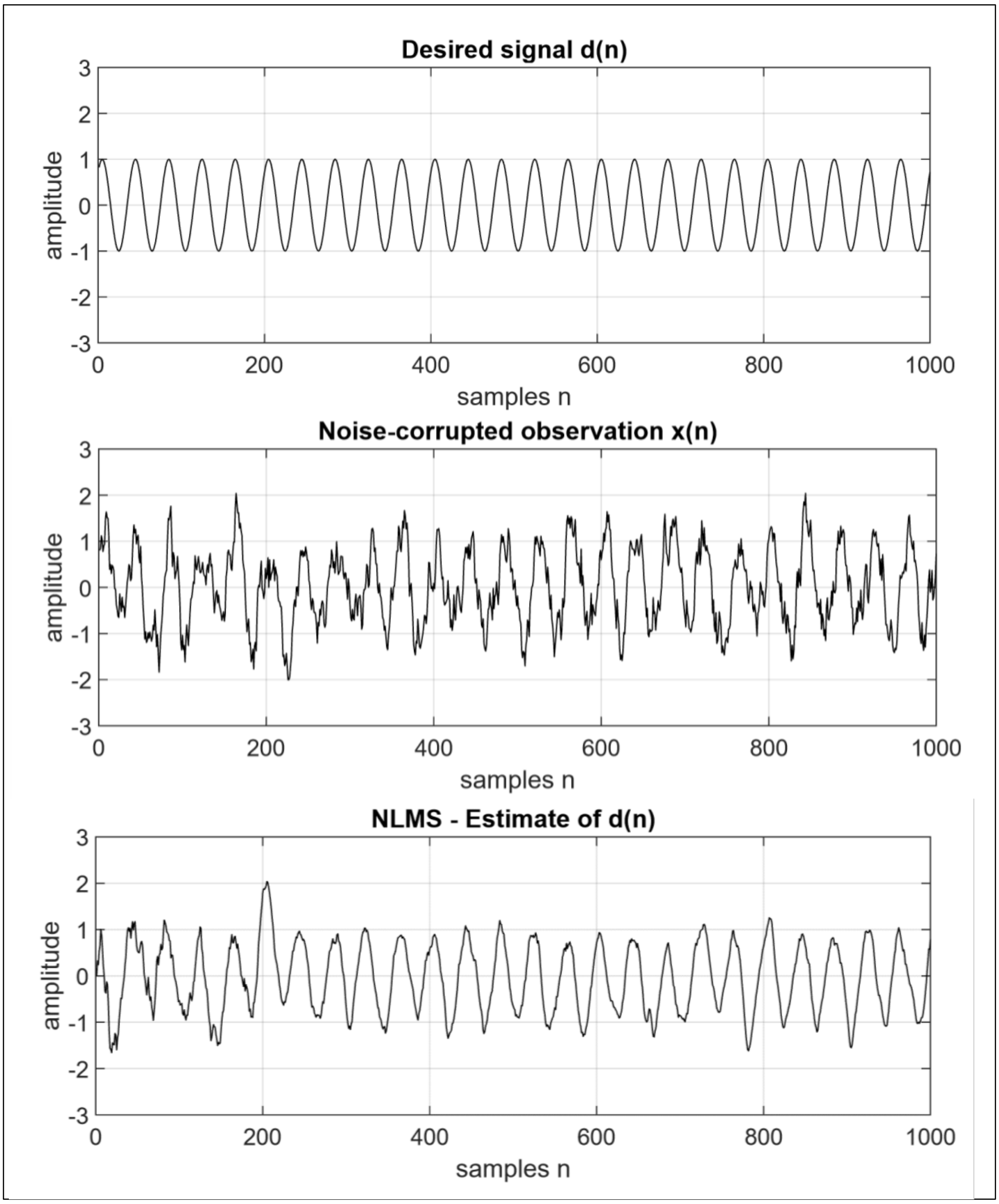

Figure 17: ANC without reference signal using NLMS algorithm

The plot shows, that the adaptive filter can produce a fairly accurate estimate of d(n) after about 50 samples. The NLMS adaptive filtering algorithm is very effective in adaptive noise cancellation.

For the performance evaluation, the learning curve of the LMS algorithm was computed. The learning curve in the next figure shows the ensemble-averaged squared error versus the number of adaptation cycles. The ensemble averaging was performed over 100 independent runs of the experiment. The values used in the implementation for β are 0.05 and 0.25.

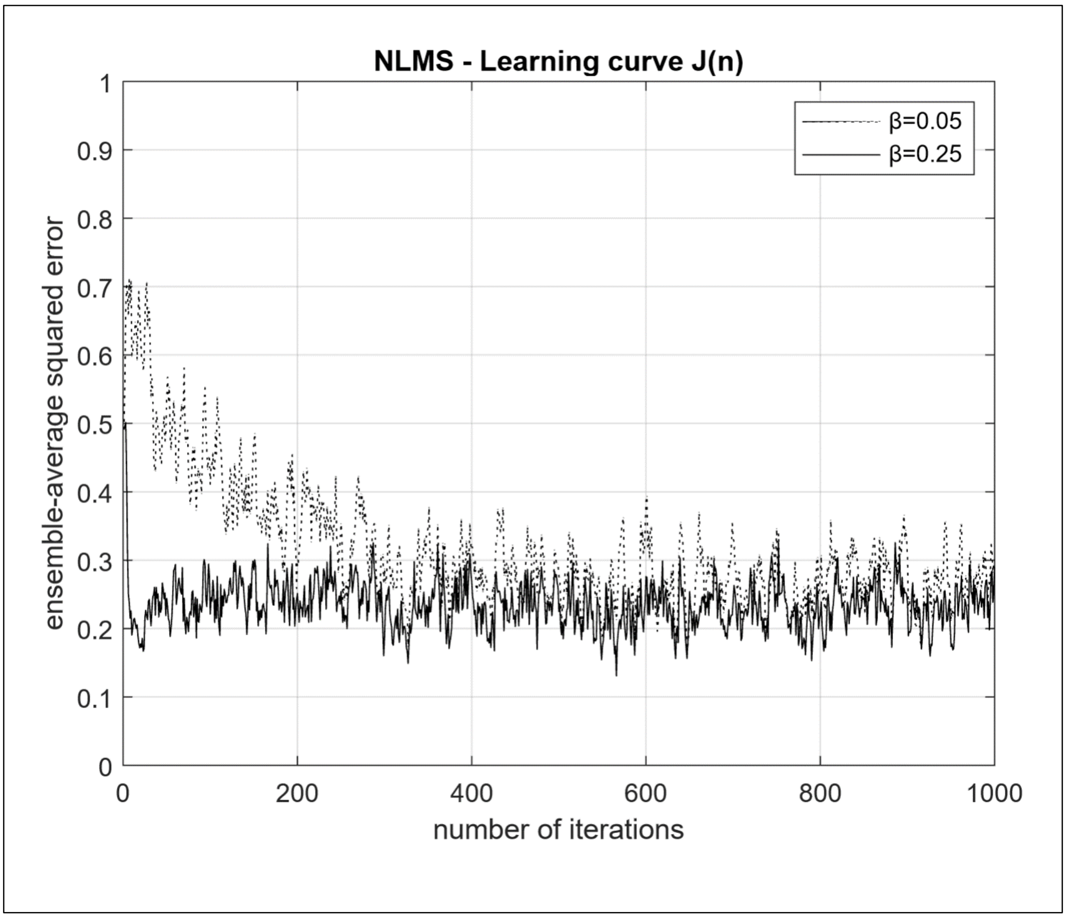

Figure 18: Learning curves of the NLMS algorithm

- Rate of convergence

  The convergence behavior of the NLMS algorithm can be determined from the learning curve and depends on the normalized step-size parameter β. The lower β is, the higher is the rate of convergence, which means that the algorithm requires more adaptation cycles until it converges to its steady-state value. For a step-size of $\beta = 0.25$, the algorithm requires only 50 iterations to converge to its steady-state. Using a step-size of $\beta = 0.05$, the algorithm needs approximately 300 adaptation cycles.

- Error performance

  The error performance is nearly the same for the different time-varying step-sizes. The cost function of the NLMS algorithm aims to minimize the MSE.

The estimated steady-state mean square error results in

$$\hat{\xi}(\infty)\begin{cases}0.233 & ;\ for\ \beta = 0.05\\ 0.215 & ;\ for\ \beta = 0.25\end{cases}$$

The steady-state value of the average squared error decreases slightly with increasing time-varying step-size.

- Signal-to-noise ratio

  The signal-to-noise ratio in decibels was computed after applying the NLMS algorithm. The output SNR is the ratio between the power of the filtered signal y(n) and the power of the noise e(n) at the output.

$$\beta = 0.05: \quad SNR_{out} = 10\log_{10}\frac{\sum_n y(n)^2}{\sum_n e(n)^2} = 6.97\ dB$$

$$\beta = 0.25: \quad SNR_{out} = 10\log_{10}\frac{\sum_n y(n)^2}{\sum_n e(n)^2} = 8.74\ dB$$

The NLMS algorithm with normalized step-size $\beta = 0.05$ produces an output SNR of 6.97 dB. Using the step-size $\beta = 0.25$ the output SNR results in a higher value of 8.74 dB.

3.2.3 The RLS solution

The RLS algorithm, summarized in the following table, produces the set of filter coefficients that minimize the weighted least squares error.

Inputs	W= tap-weight vector w_{n-1} x = delayed process x(n-n_0) to estimate d(n) d = desired signal d(n)
Outputs	y = filter output y(n) w= tap-weight vector update w(n)
Parameters	p = filter order λ = exponential weighting factor: $0 < \lambda \leq 1$ δ = value used to initialize P(0)
Initialization	$w_0 = 0$ $\mathrm{P}(0) = \delta^{-1}I$
Computation	1. Computing the gain vector: $$g(n) = \frac{1}{\lambda + x^T x(n) z(n)} z(n)$$ 2. Filtering: $$y(n) = w_n^T x(n)$$ 3. Error Estimation: $$e(n) = d(n) - y(n)$$ 4. Tap-weight vector adaptation: $$w_n = w_{n-1} + \alpha(n) g(n)$$ 5. Correlation update: $$\mathrm{P(n)} = \frac{1}{\lambda}[P(n-1) - g(n) z^H(n)]$$

Table 3: Summary of the RLS algorithm

Using a 12th- order adaptive noise canceller with coefficients that are updated using the RLS algorithm, the estimate of the desired signal d(n) that is produced with an exponential weighting factor of $\lambda = 1$ is shown in the next figure.

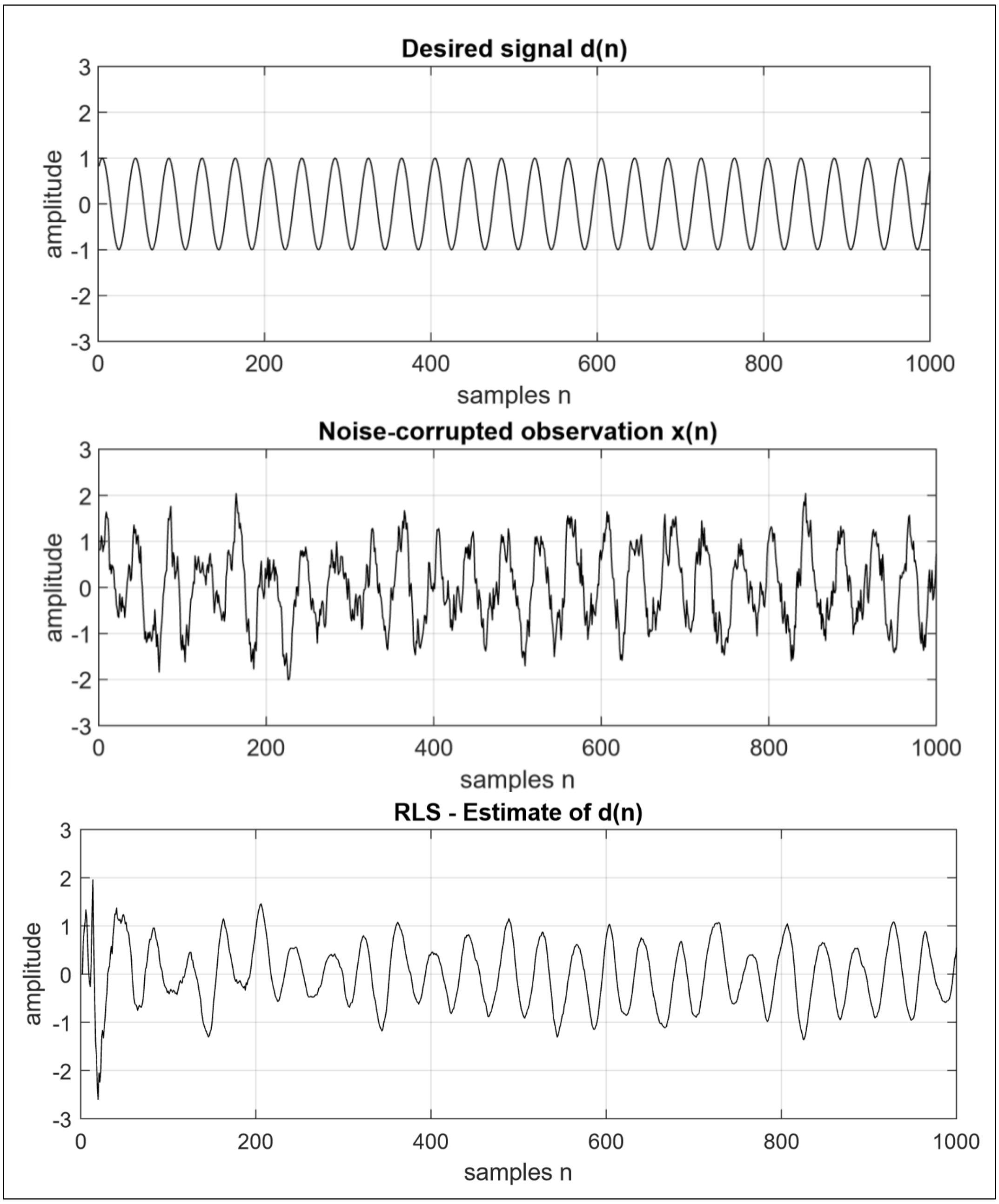

Figure 19: ANC without reference signal using RLS algorithm

At the beginning, the amplitude of the signal estimate $\hat{d}(n)$ is very high compared with the desired signal and the form of a sine wave cannot be reproduced. After about 50 samples, the adaptive filter can produce a fairly accurate estimate of d(n). The RLS algorithm shows good performance in noise cancellation.

Although, the error signal e(n) in the RLS algorithm is differently compared with the least mean-square algorithms, it is possible to compute the learning curves by choosing the a priori estimation error $\xi(n)$ as the error of interest. The learning curve shows the ensemble-averaged of the a priori estimation error $\xi(n)$ versus the number of adaptation cycles. The ensemble averaging was performed over 100 independent runs of the experiment. The implementation of the RLS algorithm is carried out with the exponential weighting factor $\lambda = 0.9$ and $\lambda = 1$.

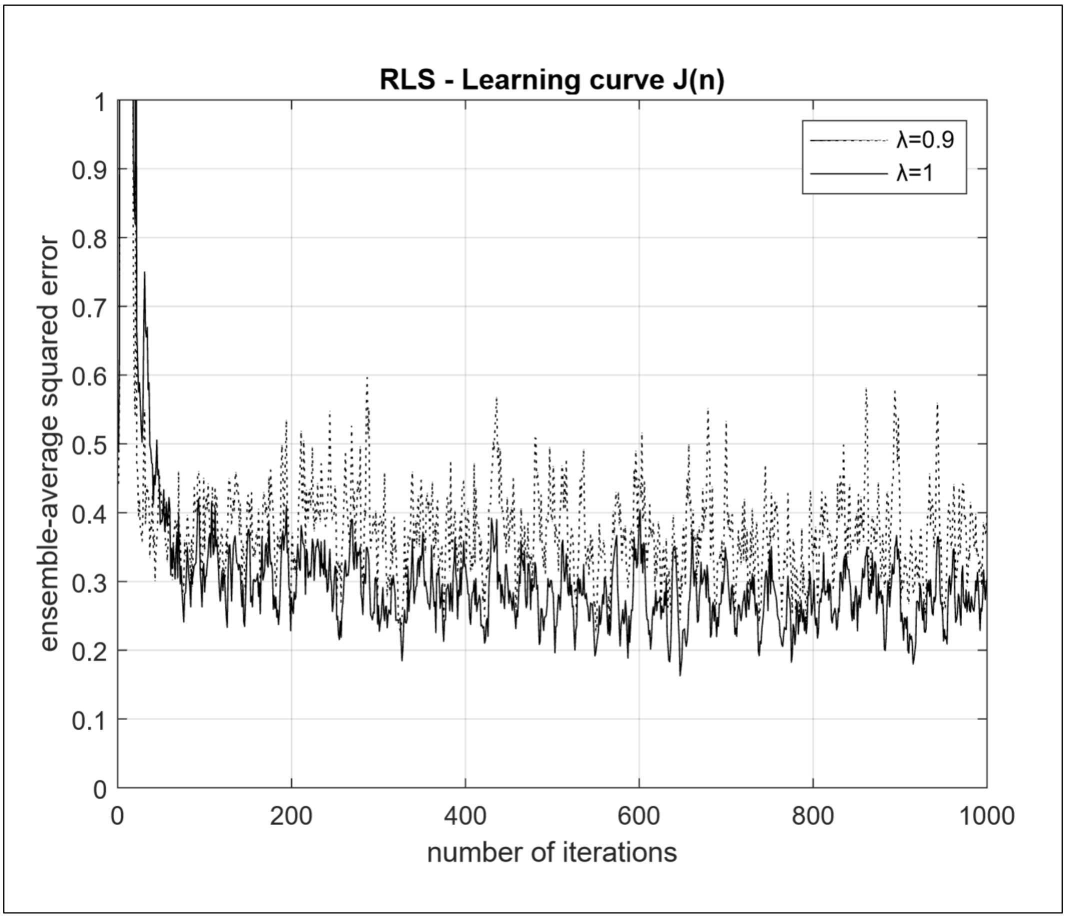

Figure 20: Learning curves of the RLS algorithm

- Rate of convergence

  Analyzing the learning curves, the rate of convergence of the RLS algorithm is relatively insensitive to variations in the inverse autocorrelation matrix P(n), which has the effect of whitening the tap inputs. For an exponential weighting factor $\lambda = 1$, the algorithm requires only 50 iterations to converge to its steady-state. A lower exponential weighting factor $\lambda = 0.9$ allows the algorithm to converge to steady-state conditions after approximately 25 adaptation cycles.

- Error performance

  The RLS algorithm minimizes a weighted linear least squares cost function relating to the input signals. The error performance is based on the a priori estimation error $\xi(n)$.

  The estimated steady-state square error results in

$$\hat{\xi}(\infty)\begin{cases} 0.376 & ;\ for\ \lambda\ =\ 0.9 \\ 0.327 & ;\ for\ \lambda\ =\ 1 \end{cases}$$

- Signal-to-noise ratio

  The signal-to-noise ratio in decibels was computed after applying the RLS algorithm. The output SNR is the ratio between the power of the filtered signal y(n) and the power of the noise e(n) at the output.

$$\lambda\ =\ 0.9:\quad SNR_{out} = 10\log_{10}\frac{\sum_n y(n)^2}{\sum_n e(n)^2} = -1.6\cdot 10^{-4}\ dB$$

$$\lambda\ =\ 1:\qquad SNR_{out} = 10\log_{10}\frac{\sum_n y(n)^2}{\sum_n e(n)^2} =\ 6.70\ dB$$

The RLS algorithm with an exponential weighting factor $\lambda\ =\ 1$ produces an output SNR of 6.70 dB. Using an exponential weighting factor $\lambda\ =\ 0.9$ the system becomes unstable, which can be seen in the negative SNR output value. The negative value also indicates that more noise than signal strength is present after the filtering process.

4 Discussion

Based on the experimental results in the previous chapter, the adaptive behavior of the different algorithms is discussed. Each algorithm works on different methods for adaptive noise cancellation and improves system performance in its own way. In order to find the optimum filter for noise cancellation, a trade-off between the three performance criteria must be considered. When examining these algorithms, three performance criteria are used: the rate of convergence, the error performance and the signal-to-noise ratio SNR. The following deductions can be made about the algorithms presented for adaptive noise cancellation without reference signal.

The statistical performance of the different adaptive algorithms, including the rate of convergence and the error performance, are studied using ensemble-averaged learning curves. The signal-to-noise ratio SNR in decibels was computed after filtering and describes the ratio between the power of the filtered signal y(n) and the power of the noise signal e(n). The results of the adaptive behavior of the LMS, NLMS and RLS algorithm from the previous chapter are summarized in the following table.

Algorithms	**Parameters**	**Rate of convergence**	**Error performance**	**SNR**
LMS	$\mu = 0.002$	100 iterations	0.307	6.73 dB
	$\mu = 0.02$	50 iterations	0.295	8.23 dB
NLMS	$\beta = 0.05$	300 iterations	0.233	6.97 dB
	$\beta = 0.25$	50 iterations	0.215	8.74 dB
RLS	$\lambda = 0.9$	25 iterations	0.376	$-1.6 \cdot 10^{-4}$ dB
	$\lambda = 1$	50 iterations	0.327	6.70 dB

Table 4: Performance comparison of the different adaptive filtering algorithms

- Rate of convergence

  The LMS algorithm with a step-size of 0.02, the NLMS algorithm with a normalized step-size of $\beta = 0.25$ and the RLS algorithm with an exponential weighting factor $\lambda = 1$ show the same rate of convergence. Each algorithm converges to its steady-state after approximately 50 adaptation cycles. Based on this information, it is possible to make a direct comparison between the algorithms with their specific parameters in the other performance criteria.

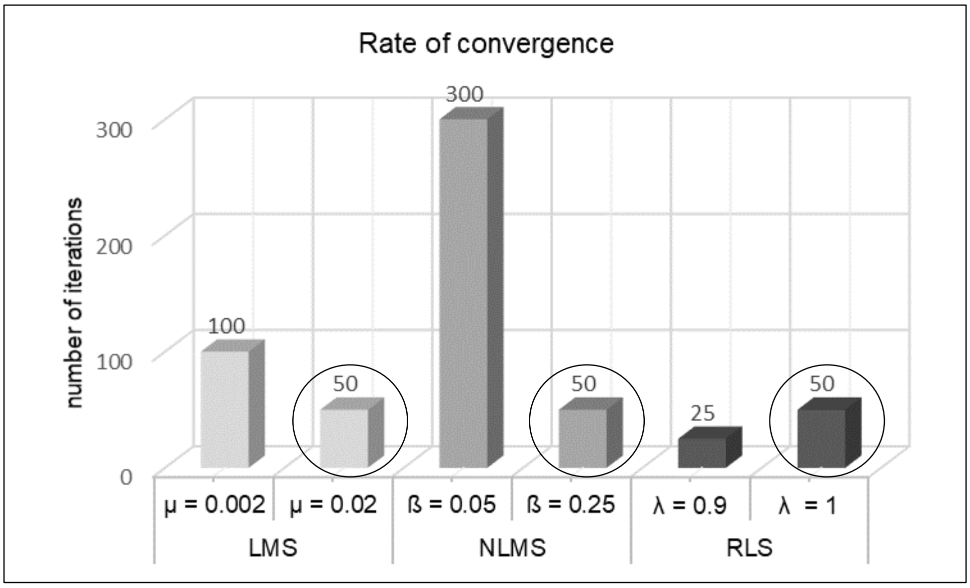

Table 5: Rate of convergence of LMS, NLMS and RLS

- Error performance

  The next analysis is about the steady-state squared error. The table depicts a comparison between the error of the three algorithms during the consecutive iterations of the algorithms. When stable, the RLS algorithm shows the highest error value with a steady-state priori estimation error value of 0.327 compared with the steady-state mean-square error of the LMS and NLMS algorithm. The LMS algorithm converges to a steady-state mean-square error of 0.295, which is a higher value compared with the NLMS. The NLMS algorithm shows the best error performance as it has the lowest steady-state squared error.

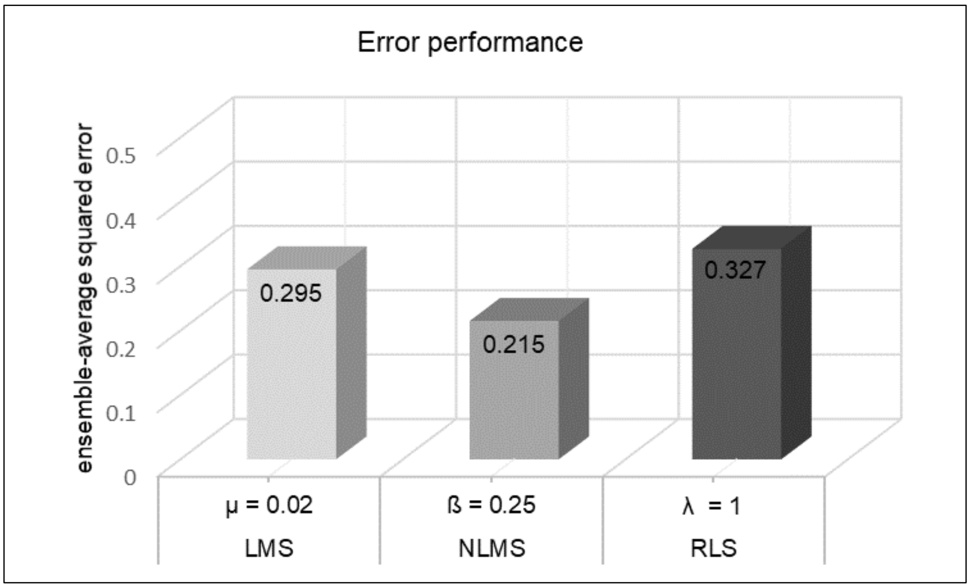

Table 6: Error performance of LMS, NLMS and RLS

- Signal-to-noise ratio SNR

  Here the comparison between the different adaptive filtering algorithms is described by the output SNR in decibels. The signal-to-noise ratio SNR after filtering equals 8.23 dB for the LMS algorithm with a step-size of $\mu = 0.02$, which results in a higher value compared with the RLS algorithm with an exponential weighting factor $\lambda = 1$. The NLMS with a normalized step-size of $\beta = 0.25$ shows the highest signal-to-noise ratio of 8.74 dB. A higher value of SNR indicates that the adaptive filter can remove a larger amount of noise.

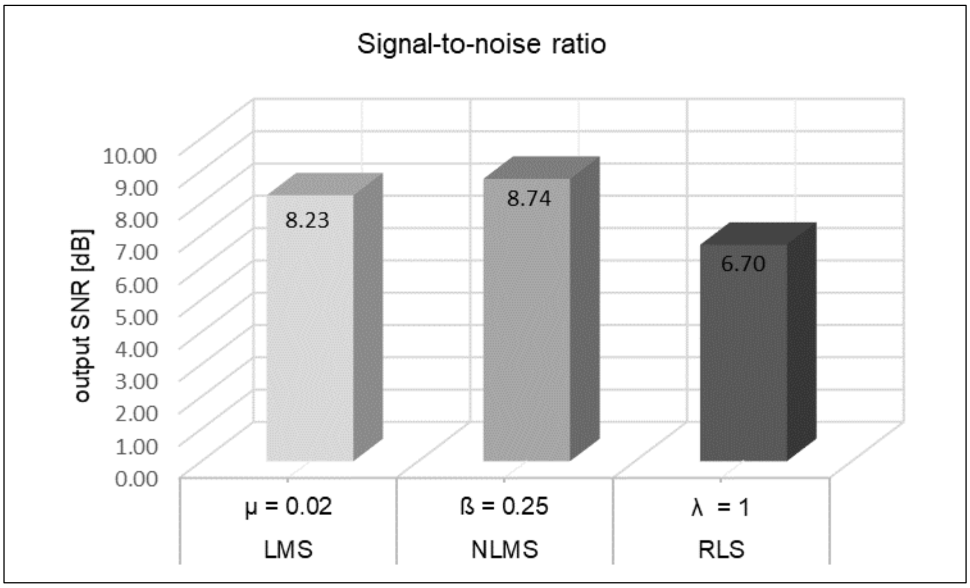

Table 7: Signal-to-noise ratio of LMS, NLMS and RLS

According to the performance results, the NLMS algorithm with a normalized step-size of $\beta = 0.25$ seems the best choice for the adaptive noise cancellation without reference signal. With the same rate of convergence as LMS and RLS, the NLMS shows a better error performance of the steady-state MSE and the signal-to-noise ratio after filtering is higher compared with the LMS and RLS algorithm, which improves the performance of the adaptive noise cancellation.

As a result, a better estimate of the desired signal can be generated using the NLMS algorithm, which can be compared with the LMS and RLS algorithm in the next figure.

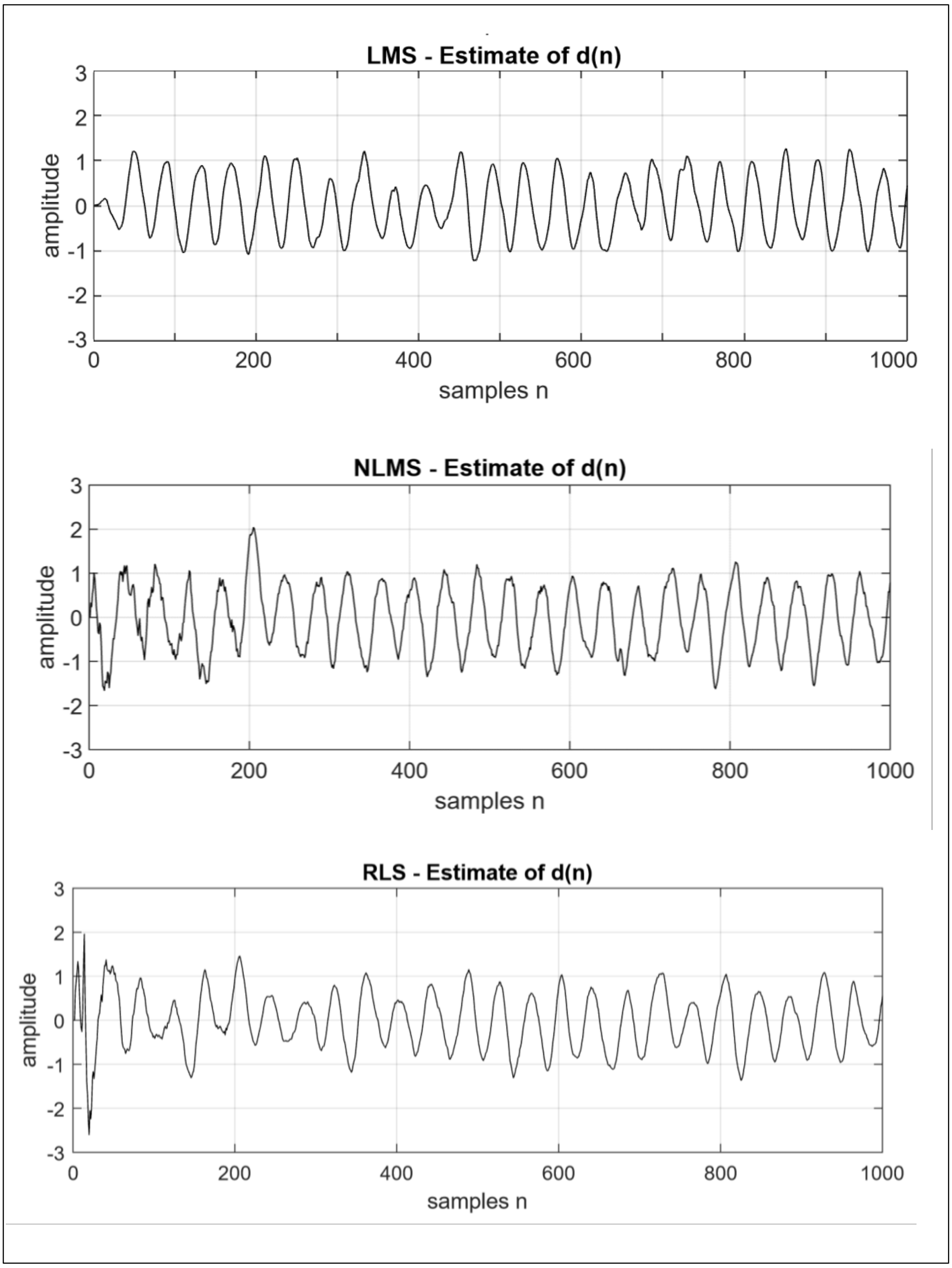

Figure 21: Results of the ANC without reference signal using LMS, NLMS and RLS

5 Conclusions

The main purpose of the thesis is to design an optimum adaptive filter, which shows good performance results in adaptive noise cancellation without reference signal and produces the best estimate of the desired signal from the noisy environment. The Least mean-square LMS, the Normalized Least mean-square NLMS and the Recursive Least square RLS algorithm are implemented, analyzed and compared against each other. Three performance criteria are used in the study of these algorithms: the rate of convergence, the error performance and the signal-to-noise ratio SNR.

Based on the implementation results, it concludes that the adaptive noise cancellation application without reference signal, benefits more from the use of the NLMS algorithm instead of the LMS or RLS algorithm. At the same rate of convergence, the NLMS algorithm shows a better error performance and a higher signal-to-noise ratio than the LMS and RLS algorithm. The NLMS algorithm with a normalized step-size of $\beta = 0.25$ seems the best choice for the adaptive noise cancellation without reference signal.

But it must be considered that the performance results strongly depend on the input data, including both the noise sequence and the reference signal. Therefore, the choice of the best adaptive filtering algorithm is application dependent. With the written test environment, it is possible to test several input datasets, regardless of real measurement data from industry or simulation data, to find the adaptive filtering algorithm, which best meets specific design and application needs. The future work involves the implementation of the NLMS adaptive filtering algorithm in an embedded data processing unit.

References

[1] J. Fraden, "Noise in Sensors and Circuits," in *Handbook of Modern Sensors*, 3rd ed. Springer International Publishing AG, 2015, pp. 204-212.

[2] S. Haykin, “Adaptive systems for signal process.,” in: S. Stergiopoulos, *Advanced Signal Processing: Theory and Implementation for Sonar, Radar, and Non-Invasive Medical Diagnostic Systems*, 2nd ed. CRC Press, 2017.

[3] M. H. Hayes, “Opimum filters,” in *Statistical Digital Signal Processing and Modeling*. John Wiley & Sons, 1996, ch.7, pp. 335–337.

[4] S. Haykin, “Adaptive Filter Theory,” 5th ed. McMaster University, Ontario Canada, Pearson, 2014, pp. 1-29.

[5] S. Dixit and D. Nagaria, “LMS Adaptive Filters for Noise Cancellation: A Review,” *International Journal of Electrical and Computer Engineering (IJECE),* vol. 7, pp. 2520-2529, October 2017.

[6] J. Chhikara and J. Singh, “Noise cancellation using adaptive algorithms,” *International Journal of Modern Engineering Research (IJMER),* vol.2, pp.792-795, June 2020.

[7] A. Tajammul, D. K. Gupta, V. K. Gupta, "Analysis of Noise Signal Cancellation using LMS, NLMS and RLS Algorithms of Adaptive Filter,” *Advanced Research in Electrical and Electronic Engineering,* vol. 2, pp. 50-53, December 2015.

[8] S. Douglas, et.Al., “Convergence Issues in the LMS Adaptive Filter,” CRC Press LLC, 2000.

[9] M. H. Hayes, "Adaptive filtering," in *Statistical digital signal processing and modeling*, John Wiley & Sons, 1996, pp. 493-570.

[10] E. Ifeachor and B. Jervis, “Adaptive Digital Filters,” in *Digital Signal Processing: A Practical Approach*, 2nd ed. Pearson Education, 2002, ch.10, p. 647.

[11] S. V. Vaseghi, “Digital Filters,” in *Multimedia Signal Processing: Theory and Applications in Speech, Music and Communications*, 1st ed. John Wiley & Sons, 2007, ch.4, pp.111-117.

[12] S. Stearns, “Error surfaces of recursive adaptive filters,” *IEEE Transactions on Circuits and Systems*, vol.28, pp. 763-766, June 1981.

[13] MathWorks, “Compare RLS and LMS Adaptive Filter Algorithms,” [Online]. Available: https://www.mathworks.com/help/dsp/ug/compare-rls-and-lms-adaptive-filter-algorithms.html [Accessed: 15 03 2020].

[14] G. Yadav and B. Krishna, “Study of different adaptive filter algorithms for noise cancellation in real-Time environment”. *International Journal of Computer Applications*, January 2014.

[15] B. Widrow et al., "Adaptive noise cancelling: Principles and applications," in *Proceedings of the IEEE*, vol. 63, no. 12, pp. 1692-1716, December 1975.

[16] A. Singh, “Adaptive Noise Cancellation,” 2001 [Online]. Available: http://www.cs.cmu.edu/~aarti/pubs/ANC.pdf [Accessed: 17 04 2020].

[17] S. Jimaa, “Convergence Evaluation of a Random Step-Size NLMS Adaptive Algorithm in System Identification and Channel Equalization,” 2011 [Online]. Available: https://www.intechopen.com/books/adaptive-filtering/convergence-evaluation-of-a-random-step-size-nlms-adaptive-algorithm-in-system-identification-and-ch [Accessed: 19 04 2020].

[18] T. Drumright, “Adaptive Filtering,” 1998 [Online]. Available: https://courseware.ee.calpoly.edu/~fdepiero/curr/dsp/adaptive/Adapt.html [Accessed: 17 04 2020].

[19] L. Surendra, S. Shameem and H. Khan, “Performance Comparison of LMS, SMI and RLS Adaptive Beamforming Algorithms for Smart Antennas,” *International Journal of Computer Science* and *Technology (IJCST),* vol. 3, June 2012.

[20] S. Haykin, “The recursive Least-Squares (RLS) algorithm,” in *Adaptive Filter Theory*, 5th ed. McMaster University, Ontario Canada, Pearson, 2014, ch.10, pp. 444–453.

[21] Q. Noman and H. Mohammed, (2017). “Performance of Adaptive Noise Cancellation with Normalized Last-Mean-Square Based on the Signal-to-Noise Ratio of Lung and Heart Sound Separation,” *Journal of Healthcare Engineering,* July 2018.

[22] L. Mead, "How Can Progress on Infrastructure, Industry and Innovation Contribute to Achieving the SDGs?," 2017 [Online]. Available: lhttps://sdg.iisd.org/commentary/policy-briefs/how-can-progress-on-infrastructure-industry-and-innovation-contribute-to-achieving-the-sdgs/ [Accessed: 11 06 2020].

List of Figures

Figure 1: Noise-corrupted observation in the measurement application 1
Figure 2: Principle of noise cancellation (Source: modified from [9]) 3
Figure 3: Measurement enhancement system 4
Figure 4: Block diagram of an adaptive filter (Source: modified from [9]) 8
Figure 5: Block diagram of a direct-form FIR adaptive filter (Source: modified from [9]) 10
Figure 6: Block diagram of an IIR adaptive filter 12
Figure 7: Types of adaptive filtering algorithms 14
Figure 8: Principle of noise cancellation (Source: Figure 2) 20
Figure 9: Noise cancellation with reference signal (Source: modified from [9]) 21
Figure 10: Noise cancellation without reference signal (Source: modified from [9]) 23
Figure 11: Comparison criteria for performance evaluation 25
Figure 12: Desired signal d(n) that is to be estimated 30
Figure 13: Noise-corrupted observation 31
Figure 14: Reference signal x(n-n0) 32
Figure 15: ANC without reference signal using LMS algorithm 35
Figure 16: Learning curves of the LMS algorithm 36
Figure 17: ANC without reference signal using NLMS algorithm 39
Figure 18: Learning curves of the NLMS algorithm 40
Figure 19: ANC without reference signal using RLS algorithm 43
Figure 20: Learning curves of the RLS algorithm 44
Figure 21: Results of the ANC without reference signal using LMS, NLMS and RLS 50

List of Tables

Table 1: Summary of the LMS algorithm 34
Table 2: Summary of the NLMS algorithm 38
Table 3: Summary of the RLS algorithm 42
Table 4: Performance comparison of the different adaptive filtering algorithms 46
Table 5: Rate of convergence of LMS, NLMS and RLS 47
Table 6: Error performance of LMS, NLMS and RLS 48
Table 7: Signal-to-noise ratio of LMS, NLMS and RLS 49

List of Abbreviations

ADC	Analog-to-digital converter
ANC	Adaptive Noise Cancellation
AR	autoregressive
DSP	Digital Signal Processing
FIR	Finite Impulse Response
IIR	Infinite Impulse Response
LMS	Least Mean Square
LSE	Least squares error
MSE	Mean-square error
NLMS	Normalized Least Mean Square
RLS	Recursive Least Square
SNR	Signal-to-noise ratio

Appendix A – Matlab Code

```
%% -----------------------------------------------------------------------
% Master Thesis -
% Implementation of adaptive filtering algorithms for noise
% cancellation
% Author: Tanja Lampl
% Task: Adaptive noise cancellation without reference signal
% Last change: 2020-04-22
%% -----------------------------------------------------------------------
clear all;
close all;

% The goal of the adaptive noise cancellation is to estimate the desired
% signal d(n) from a noise-corrupted observation x(n) = d(n) + v1(n)

%% Adaptive filter
N = 10000;     % number of samples
R = 100;       % ensemble average over 100 runs
nord = 12;     % filter order
% LMS coefficients
mu = 0.002;   % step size
% NLMS coefficients
beta = 0.25;  % normalized step size
% RLS coefficients
delta = 0.001;
lambda = 1; % exponential weighting factor

% create arrays of all zeros
MSE_LMS = zeros(N,1);
MSE_NLMS = zeros(N,1);
LSE_RLS = zeros(N,1);
dhat_LMS = zeros(N,1);
dhat_NLMS = zeros(N,1);
dhat_RLS = zeros(N,1);
err_LMS = zeros(N,1);
err_NLMS = zeros(N,1);
err_RLS = zeros(N,1);

%for r=1:R       % used for computation of learning curves

    %% Noise-corrupted observation: x(n) = d(n) + v1(n)
    d = sin([1:N]*0.05*pi);    % desired signal
    g = randn(1,N)*0.25;       % Gaussian white noise with a variance of 0.25
    v1= filter(1,[1 -0.8],g);  % filtered white noise
    x = d + v1;                % noisy process

    % plot of d(n) and x(n)
    figure(1)
    subplot(2,1,1)
    plot(d(1:1000),':k')
    hold on
```

```
plot(x(1:1000),'k')
legend("d(n)", "x(n)")
title("Noise-corrupted observation");
xlabel('samples n')
ylabel('amplitude')
axis([0 1000 -3 3])
grid on

%% Reference signal
% Here, the reference signal v2(n) is not known.
% In that case, it is possible to derive a reference signal by
% delaying the noisy process x(n) = d(n) + v1(n).
% The delayed signal x(n-n0) is used as the reference signal for
% the canceller.
n0 = 25;            % delay of 25 samples
len = N - n0;       % reduced vector length
x_del = zeros(N,1); % create array of all zeros

% generate delayed signal
for i = 1:len
    x_del(i) = x(i+n0);
end

% plot of x_del(n)
figure(2)
subplot(2,1,1)
plot(x(1:1000),':k')
hold on
plot(x_del(1:1000),'k')
legend("x(n)", "x(n-n0)")
title("Reference signal x(n-n0)");
xlabel('samples n')
ylabel('amplitude')
axis([0 1000 -3 3])
grid on

% create arrays of all zeros
W_LMS = zeros(nord,1);
W_NLMS = zeros(nord,1);
W_RLS = zeros(nord,1);
U = zeros(nord,1);
P = ((1/delta)*eye(nord,nord));

for i=1:N
    U = [x_del(i)
    U(1:(nord-1))];
    x_n = x(i);

    %% LMS Algorithm
    % Step 1: Filtering
    y_LMS = (W_LMS'*U);
    dhat_LMS(i) = (dhat_LMS(i)+y_LMS);
```

```
        % Step 2: Error Estimation
        E_LMS = (x_n-y_LMS);
        err_LMS(i) = err_LMS(i)+E_LMS;
        % Step 3: Tap-weight vector adaptation
        W_LMS = (W_LMS+(mu*E_LMS*U));

        %% NLMS Algorithm
        % Step 1: Filtering
        y_NLMS = (W_NLMS'*U);
        dhat_NLMS(i) = (dhat_NLMS(i)+y_NLMS);
        % Step 2: Error Estimation
        E_NLMS = (x_n-y_NLMS);
        err_NLMS(i) = err_NLMS(i)+E_NLMS;
        % Step 3: Tap-weight vector adaptation
        W_NLMS = (W_NLMS+((beta/((norm(U)^2)))*conj(E_NLMS)*U));

        %% RLS Algorithm
        % Step 1: Computing the gain vector
        g = (((1/lambda)*P*U)/(1+((1/lambda)*U'*P*U)));
        % Step 2: Filtering
        y_RLS = (W_RLS'*U);
        dhat_RLS(i) = (dhat_RLS(i)+y_RLS);
        % Step 3: Error Estimation
        E_RLS = (x_n-y_RLS);
        err_RLS(i) = err_RLS(i)+E_RLS;
        % Step 4: Tap-weight vector adaptation
        W_RLS = W_RLS+g*conj(E_RLS);
        %Step 5: Correlation Update
        P = (((1/(lambda))*P)-((1/lambda)*g*U'*P));
        %% Error performance
        MSE_LMS(i) = norm(MSE_LMS(i)+(abs(E_LMS)^2));
        MSE_NLMS(i) = norm(MSE_NLMS(i)+(abs(E_NLMS)^2));
        LSE_RLS(i) = norm(LSE_RLS(i)+(abs(E_RLS)^2));
    end
%end

%% Error performance
MSE_LMS = MSE_LMS/R;
MSE_NLMS = MSE_NLMS/R;
LSE_RLS = LSE_RLS/R;

% plot estimate of d(n)
figure(3)
plot(dhat_LMS(1:1000),'b');
title("LMS - Estimate of d(n)");
xlabel('samples n')
ylabel('amplitude')
axis([0 1000 -3 3])
grid on
figure(4)
plot(dhat_NLMS(1:1000),'b');
title("NLMS - Estimate of d(n)");
```

```
xlabel('samples n ')
ylabel('amplitude')
axis([0 1000 -3 3])
grid on
figure(5)
plot(dhat_RLS(1:1000),'b');
title("RLS - Estimate of d(n)");
xlabel('samples n')
ylabel('amplitude')
axis([0 1000 -3 3])
grid on

%% Result of adaptive noise cancellation
figure(6)
plot(dhat_LMS,':k');
hold on
plot(dhat_NLMS,'k');
hold on
plot(dhat_RLS,'--k');
legend("dhat(n) LMS", "dhat(n) NMLS","dhat(n) RLS");
hold off
title("Result of adaptive noise cancellation")
xlabel('samples n')
ylabel('amplitude')
axis([0 1000 -3 3]);
grid on;

%% Plot learning curves
figure(10)
plot(MSE_LMS(1:1000),':k')
ylabel('ensemble-average squared error')
xlabel('number of iterations')
title('LMS - Convergence rate ')
axis([0 1000 0 1])
grid on

figure(11)
plot(MSE_NLMS(1:1000),':k')
ylabel('ensemble-average squared error')
xlabel('number of iterations')
title('NLMS- Convergence rate ')
axis([0 1000 0 1])
grid on

figure(12)
plot(LSE_RLS(1:1000),':k')
ylabel('ensemble-average squared error')
xlabel('number of iterations')
title('RLS - Convergence rate ')
axis([0 1000 0 1])
grid on
```