
NOISE CANCELLATION WITH LMS VARIANTS

*Fahri VATANSEVER** 

Received: 18.09.2020; revised: 25.01.2021; accepted: 02.02.2021

Abstract: The noise is one of the important factors that affect negatively the operation of circuits and systems. There are different methods and techniques for cancelling/reducing/de-noising these unwanted signals that mix with the original signal. In this study, an application/simulator that performs noise cancellation operations with least mean-squares algorithms was designed. The application, which has powerful visual support and a user-friendly interactive interface, can make single or comparative noise cancellation operations and performance analyzes with different algorithms on noisy signals. The effects of the algorithm parameters can be observed, information about the related algorithms can be obtained via the developed application that can also be used for educational purpose.

Keywords: Noise cancellation, LMS, adaptive filter, simulator.

LMS Varyantlarıyla Gürültü Arındırma

Öz: Devre ve sistemlerin işleyişlerini olumsuz yönde etkileyen önemli faktörlerden birisi de gürültülerdir. Orijinal işarete karışan bu istenmeyen işaretlerin arındırılması/azaltılması/temizlenmesi için farklı yöntem ve teknikler mevcuttur. Gerçekleştirilen çalışmada, en küçük karesel ortalama algoritmalarıyla gürültü arındırma işlemlerini gerçekleştiren bir uygulama/simülör tasarlanmıştır. Güçlü görsel desteğe ve kullanıcı dostu etkileşimli arayüze sahip, eğitim amaçlı da kullanılabilen uygulama ile gürültülü işaretler üzerinde, farklı algoritmalarla tekli veya karşılaştırmaları gürültü arındırma işlemleri ve performans analizleri yapılabilmekte, algoritma parametrelerinin etkileri gözlemlenebilmekte ve ilgili algoritmalar hakkında bilgiler edinilebilmektedir.

Anahtar Kelimeler: Gürültü arındırma, LMS, adaptif filtre, simülör.

1. INTRODUCTION

The main purpose of the noise cancellation/reduction process, which is used in wide areas such as communication, biomedical, control, signal/image processing, is to reduce or completely eliminate the undesired signal (noise) in the desired signal. The fundamental process used in de-noising in digital signal processing is filtering. For this purpose, finite impulse response (FIR) and infinite impulse response (IIR) digital filters are used. However, problems are encountered in de-noising process if filters which have fixed coefficients are used, and the performance rate decreases. Adaptive filters have been used frequently in recent years to eliminate these problems.

The adaptive filters are filters with self-adjusting characteristics. These filters, which can be designed as analog, digital or hybrid (mixed), have advantages and disadvantages compared to each other. For example, while analog filters consume low power and respond quickly, almost all

*Bursa Uludağ University, Faculty of Engineering, Electrical-Electronics Eng. Dept. 16059 Bursa/Turkey

Correspondence author: Fahri Vatansever (fahriv@uludag.edu.tr)

of them are implemented as digital (FIR or IIR) due to the complexity of the optimization algorithms.

As a result of the rapid development in the computer world, computer-aided technologies (CAx) concepts (such as computer-aided engineering (CAE), computer-aided design (CAD), computer-aided manufacturing (CAM), computer-aided software engineering (CASE), computer-aided instruction (CAI), computer-aided learning (CAL) etc.) have currently gained significant place. Simulators, applications, web pages, etc. are developed in many areas using computer software (Maddala, 2010; Kaçar and Çankaya, 2010; Kaçar and Çankaya, 2012; Yalcin and Vatansever, 2016; Hatun and Vatansever, 2016; Vatansever and Yalcin, 2017; Kaçar et al., 2017; Vatansever and Hatun, 2019).

In this study, an application/simulator that performs noise cancellation operations with different variants of the least mean-squares (LMS) algorithm is designed. The application, which is designed using MATLAB App Designer (MathWorks, 2018), can be also used for educational purposes, can perform single or comparative noise cancellation operations, observe the effects of algorithm parameters, and examine the information about algorithms.

This paper is organized as follows: In Section 2, adaptive filters and their application areas are summarized. In Section 3 , LMS algorithm and its variants are briefly explained. In Section 4, noise cancellation process with LMS is presented. In Section 5, designed application/simulator is explained and sample applications are given. Finally, Section 6 contains conclusions.

2. ADAPTIVE FILTER

The commonly used adaptive filter structure is given in Fig. 1. The error signal here is used to create the objective/performance function that the related adaptive algorithm will handle. Minimizing this function ensures filter output matches the desired signal. The mean square of the error signal is the popularly used cost function. Table 1 shows the basic application classes of adaptive filtering (Widrow and Stearns, 1985; Morales, 2011; Haykin, 2014; Diniz, 2020).

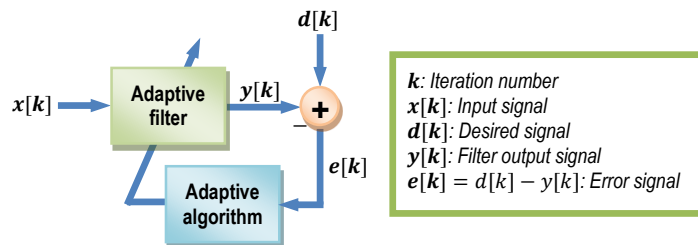
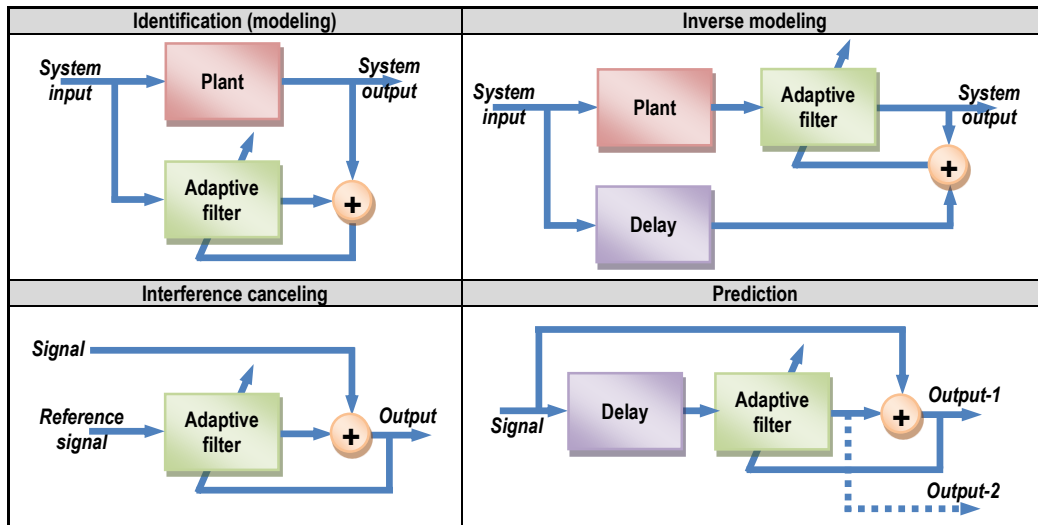


Figure 1:
The commonly used adaptive filter structure

As can be seen from Table 1, adaptive filters have wide usage areas: audio and video signal processing, noise and echo cancellation, channel estimation, channel equalization, modeling, inverse modeling, system identification, linear prediction, signal enhancement, active noise control, adaptive notch filter, jammer suppression, speech coding, adaptive line enhancement (Morales, 2011; Farhang-Boroujeny, 2013). Many studies have been carried out with different methods and techniques in the field of noise cancellation (Widrow et al., 1975; Thakor and Zhu, 1991; Gorriz et al., 2009; Sankar et al., 2010; Romero et al., 2012; Guda et al., 2014; Bismor et al., 2016; Dixit and Nagaria, 2017; Lee et al., 2017; Qureshi et al., 2017; Maurya, 2018; Taha et al., 2018; Goel and Chandra, 2019; Maurya et al., 2019).

Table 1. The basic application classes of adaptive filtering



3. LEAST MEAN-SQUARES ALGORITHM

LMS, one of the most used algorithms in adaptive filter applications, was developed in 1960 by Prof. Bernard Widrow and his Ph.D. student Ted Hoff. The main properties of this algorithm, of which many variants have been developed, can be listed as simplicity, ease of implementation, low computational complexity/cost, robustness etc. (Farhang-Boroujeny, 2013; Poularikas, 2015; Diniz, 2020).

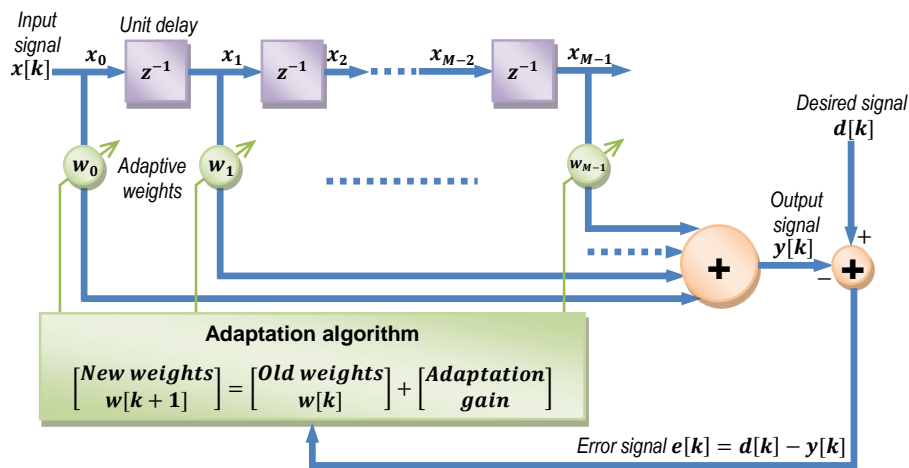


Figure 2:
Adaptive linear combiner (ALC) in transversal form

The adaptive linear combiner (adaptive FIR filter, adaptive transversal filter, ADALINE) as shown in Fig. 2.

$$x = \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{M-1} \end{bmatrix} = \begin{bmatrix} x[k] \\ x[k-1] \\ \vdots \\ x[k-M+1] \end{bmatrix}, \quad w = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_{M-1} \end{bmatrix} \quad (1)$$

The output of the filter is written by Eq. (2), with x filter input and w tap-weight (coefficients) vector.

$$y[k] = \sum_{i=0}^{M-1} w_i[k]x[k-i] = w^T x = x^T w \quad (2)$$

The error signal, its square and the mean square (squared) error (expectation of the error power or energy) are defined by Eq. (3-5).

$$e[k] = d[k] - y[k] = d[k] - w^T x = d[k] - x^T w \quad (3)$$

$$e^2[k] = (d[k] - x^T w)^2 = d^2[k] - 2d[k]x^T w + \frac{(x^T w)^2}{w^T x x^T w} \quad (4)$$

$$J = E\{e^2[k]\} = E\{d^2[k]\} - 2 \underbrace{E\{d[k]x^T\}}_{r_{dx}^T} w + w^T \underbrace{E\{xx^T\}}_{R_x} w \quad (5)$$

Since the correlation matrix R_x is a symmetric and positive definite matrix then Eq. (5) is a quadratic function of w . This function has only one minimum solution known as the optimum Wiener solution (Wiener weight vector). This solution is obtained by equating the derivative of the Eq. (5) to zero. Thus, the gradient of the mean square error function in Eq. (6) is obtained and its solution is given in Eq. (7).

$$\nabla_{E\{e^2[k]\}} = \frac{\partial E\{e^2[k]\}}{\partial w} = \frac{\partial (E\{d^2[k]\} - 2r_{dx}^T w + w^T R_x w)}{\partial w} = -2r_{dx}^T + 2R_x w = 0 \quad (6)$$

$$w_0 = R_x^{-1} r_{dx} \quad (7)$$

Generally, Eq. (7) (Wiener-Hopf equations) solution is not applicable and alternative solutions should be used as in Table 2 (Widrow and Stearns, 1985; Zaknich, 2005; Sayed, 2008; Farhang-Boroujeny, 2013; Haykin, 2014; Poularikas, 2015).

Table 2. The some iterative algorithms

Method/algorithm	Iteration equation	Solution iteration
Newton's method in gradient search	$w(k+1) = w(k) - \frac{\partial J(w(k))/\partial w(k)}{\partial^2 J(w(k))/\partial w^2(k)}$	$w(k+1) = w(k) + \mu R_x^{-1} \{r_{dx} - R_x w(k)\}$
		$w(k+1) = w(k) + \mu(k) \{\epsilon(k)I + R_x\}^{-1} \{r_{dx} - R_x w(k)\}$
Steepest-descent algorithm	$w(k+1) = w(k) - \mu \nabla J(w(k))$	$w(k+1) = w(k) + \mu \{r_{dx} - R_x w(k)\}$
		$w(k+1) = w(k) + \mu(k) \{r_{dx} - R_x w(k)\}$

By using steepest-descent algorithm, where $\nabla = \left[\frac{\partial}{\partial w_0} \quad \dots \quad \frac{\partial}{\partial w_{M-1}} \right]^T$ is the gradient operator, the following LMS recursion obtained.

$$w[k + 1] = w[k] - \underbrace{\mu \nabla J(w[k])}_{\nabla_{E\{e^2[k]\}}} \tag{8}$$

$$\nabla_{E\{e^2[k]\}} = \frac{\partial e^2[k]}{\partial w} = 2e[k] \frac{\partial e[k]}{\partial w} = 2e[k] \frac{\partial (d[k] - w^T x)}{\partial w} = -2e[k]x \tag{9}$$

$$w[k + 1] = w[k] + 2\mu e[k]x[k] \tag{10}$$

Replacing the iteration index k by the time index n in Eq. (10) is rewritten as Eq. (11).

$$w(n + 1) = w(n) + \underbrace{2\mu}_{\mu} e(n)x(n) \tag{11}$$

In Table 3, some variants of LMS algorithm, which also used in this study are given (Sayed, 2008; Farhang-Boroujeny, 2013; Poularikas, 2015; Diniz, 2020).

Table 3. The some variants of LMS algorithm

Algorithm	Definition	Parameters
Least Mean-Square (LMS) (LMS with constant step size)	$w(n + 1) = w(n) + \frac{2\mu}{\mu} e(n)x(n)$	$w(n) = [w_0 \quad \dots \quad w_{M-1}]^T$: Filter taps (coefficients) at time n $x(n) = [x(n) \quad \dots \quad x(n - M + 1)]^T$: Input data/signal to the filter $y(n) = w^T(n)x(n)$: Filter output $e(n) = d(n) - y(n)$: Error $d(n)$: Desired data/signal μ : Step-size factor/parameter M : Order of the filter
LMS with time-varying step size	$w(n + 1) = w(n) + \mu(n)e(n)x(n)$	
Normalized LMS (NLMS)	$w(n + 1) = w(n) + \frac{\mu}{\ x(n)\ ^2} e(n)x(n)$	$\ x(n)\ ^2 = x(n)^T x(n)$
Modified version of the normalized LMS (ϵ - NLMS)	$w(n + 1) = w(n) + \frac{\mu}{\epsilon + \ x(n)\ ^2} e(n)x(n)$	ϵ : Small positive number (constant)
Power normalized LMS (PNLMS)	$w(n + 1) = w(n) + \frac{\mu}{\epsilon + p(n + 1)} e(n)x(n)$	$p(n + 1) = ap(n) + (1 - a) x(n) ^2$ $p(0) = 0$ $0 \leq a < 1$
Self-correcting LMS filter (SCNLMSF)	The arrangement the standard LMS filter in a series form	
Sign-error LMS	$w(n + 1) = w(n) + \mu \{sign[e(n)]\}x(n)$	$sign(f) = \begin{cases} 1 & , f > 0 \\ 0 & , f = 0 \\ -1 & , f < 0 \end{cases}$
Sign-error NLMS	$w(n + 1) = w(n) + \mu \frac{\{sign[e(n)]\}x(n)}{\epsilon + \ x(n)\ ^2}$	
Dual-sign algorithm	$w(n + 1) = w(n) + \mu \{ds[e(n)]\}x(n)$	$ds(f) = \begin{cases} \epsilon sign(f) & , f > \alpha \\ sign(f) & , f \leq \alpha \end{cases}$

Sign-regressor LMS (Sign-data algorithm)	$w(n + 1) = w(n) + \mu e(n)\{sign[x(n)]\}$	
Self-correcting sign-regressor LMS	The arrangement the standard sign-regressor LMS filter in a series form	
Normalized sign-regressor LMS	$w(n + 1) = w(n) + \mu \frac{e(n)\{sign[x(n)]\}}{\varepsilon + \ x(n)\ ^2}$	
Sign-sign LMS	$w(n + 1) = w(n) + \mu\{sign[e(n)]\}\{sign[x(n)]\}$	
Normalized sign-sign LMS	$w(n + 1) = w(n) + \mu \frac{\{sign[e(n)]\}\{sign[x(n)]\}}{\varepsilon + \ x(n)\ ^2}$	
Power-of-two error algorithm	$w(n + 1) = w(n) + \mu\{pw[e(n)]\}x(n)$	$pw(f) = \begin{cases} sign(f) & , f \geq 1 \\ 2^{\lfloor \log_2 f \rfloor} sign(f) & , 2^{1-f_d} \leq f < 1 \\ \tau sign(f) & , f < 2^{1-f_d} \end{cases}$
Variable step-size LMS (VSLMS)	$w_i(n + 1) = w_i(n) + \mu_i(n)e(n)x(n - i)$ $\mathbf{w}(n + 1) = \mathbf{w}(n) + \boldsymbol{\mu}(n)e(n)\mathbf{x}(n)$	$i = 0, 1, \dots, M - 1$ $\boldsymbol{\mu}(n) = \begin{bmatrix} \mu_0(n) & & 0 \\ & \ddots & \\ 0 & & \mu_{M-1}(n) \end{bmatrix}$
Leaky LMS	$w(n + 1) = (1 - \mu\gamma)w(n) + \mu e(n)x(n)$	γ : Leakage coefficient ($0 < \gamma \ll 1$)
Modified LMS	$w(n + 1) = w(n) + \mu K x(n)e^{2^{K-1}}(n)$	K : Positive integer
Momentum LMS	$w(n + 1) = w(n) + (1 - g)\{w(n) - w(n - 1)\} + \mu e(n)x(n)$	$0 < g < 1$ $1 < a$
Error normalized step-size LMS (ENSS)	$w(n + 1) = w(n) + \frac{\mu}{1 + \mu \ e_L(n)\ ^2} e(n)x(n)$	$\ e_L(n)\ ^2 = \sum_{i=0}^{L-1} e(n - i) ^2$
Robust variable step-size LMS (RVSS)	$w(n + 1) = w(n) + \frac{\mu \ e_L(n)\ ^2}{\alpha \ e(n)\ ^2 + (1 - \alpha)\ x(n)\ ^2} e(n)x(n)$	
Error-data normalized step-size LMS (EDNSS)	$w(n + 1) = w(n) + \frac{\mu}{\alpha \ e_L(n)\ ^2 + (1 - \alpha)\ x(n)\ ^2} e(n)x(n)$	
Linearly constrained LMS	$w(n + 1) = w'(n) + \frac{a - c^T w'(n)}{c^T c} c$	$w'(n) = w(n) + \mu e(n)x(n)$ a : Constant , c : Constant vector
Least mean fourth (LMF)	$w(n + 1) = w(n) + \mu x(n)e(n)(e(n))^2$	
Least mean mixed norm (LMMN)	$w(n + 1) = w(n) + \mu x(n)e(n)\{\delta + (1 - \delta)(e(n))^2\}$	$0 \leq \delta \leq 1$
Affine projection LMS (APLMS)	$w(n + 1) = w(n) + \frac{\mu}{x(n)^T x(n) + \psi I} x(n)e(n)$	I : Identity matrix ψ : Small positive constant

4. NOISE CANCELLATION PROCESS WITH LMS

Adaptive noise cancellation (ANC) is used in wide area to reduce noise from a corrupted signal. The aim of an ANC is to cancel the noise from a signal adaptively to improve the SNR. Fig. 3 shows the block diagram of adaptive noise cancellation. The system output in Fig. 3 is written by Eq. (12) (Widrow et al., 1975).

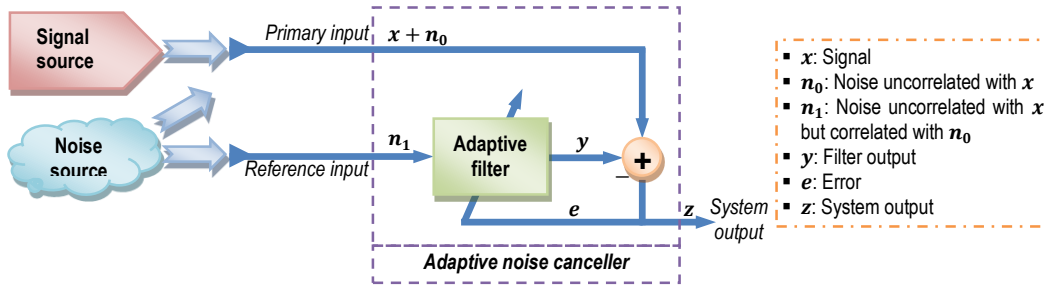


Figure 3:
The block diagram of adaptive noise cancellation

$$z = x + n_0 - y \quad (12)$$

If both sides of Eq. (12) are squared, Eq. (13) is obtained.

$$z^2 = x^2 + 2x(n_0 - y) + (n_0 - y)^2 \quad (13)$$

$$E\{z^2\} = E\{x^2\} + 2E\{x(n_0 - y)\} + E\{(n_0 - y)^2\} = E\{x^2\} + E\{(n_0 - y)^2\} \quad (14)$$

The minimum output power of system is

$$\min[E\{z^2\}] = E\{x^2\} + \min[E\{(n_0 - y)^2\}] \quad (15)$$

In order to provide $E\{z^2\}$ to be minimum, $E\{(n_0 - y)^2\}$ must also be minimum in Eq. (15), hence

$$z - x = n_0 - y \quad (16)$$

5. DESIGNED APPLICATION

In this study, an application/simulator was designed using MATLAB App Designer (MathWorks, 2018). With this application, noise cancellation operations can be performed with different variants of LMS algorithms. In the application, which includes information about algorithms, the effects of parameter changes can also be observed. In addition, single or comparative analyzes can be made on internal (sample or user-defined) and external (uploaded by the user) signals, and the results are displayed both numerically and graphically. Thus, the performance effects of both of the methods and the parameters related to the methods can be observed. In addition, it supports better understanding of both noise cancellation operations and LMS algorithms with its use in the field of education.

The main screen of the designed application is shown in Fig. 4 and its menus are given in Table 4. The user/student can select the sample signal for analysis from the "Signal" drop-down list (combobox), generate different types of signals (generating sinusoidal, exponential, constant signals by entering their parameters) (Fig. 5a) or load signals from the file. Likewise, users can select the sample noise from the "Noise" drop-down list, generate noise themselves (generating noise by entering the scaling and shifting parameters in equation $a\{random(0,1) - b\}$) (Fig. 5b) or load noise from the file. After each process, the related signal/noise graphics are simultaneously seen/plotted on the main screen. Finally, noise cancellation is performed by selecting LMS or one of its variants from the "Algorithm" drop-down list. If it is desired, comparative analysis can also be made with the "Comparative" option here. In addition, by selecting "Error analysis" or

"Parameter analysis" from the menu that opens with the mouse right-click on the "de-noised signal" graph, error graphics/values and the effects of parameter changes can be viewed both numerically and graphically on the related screens. The flowchart of single analysis process in designed application is given in Fig. 6.

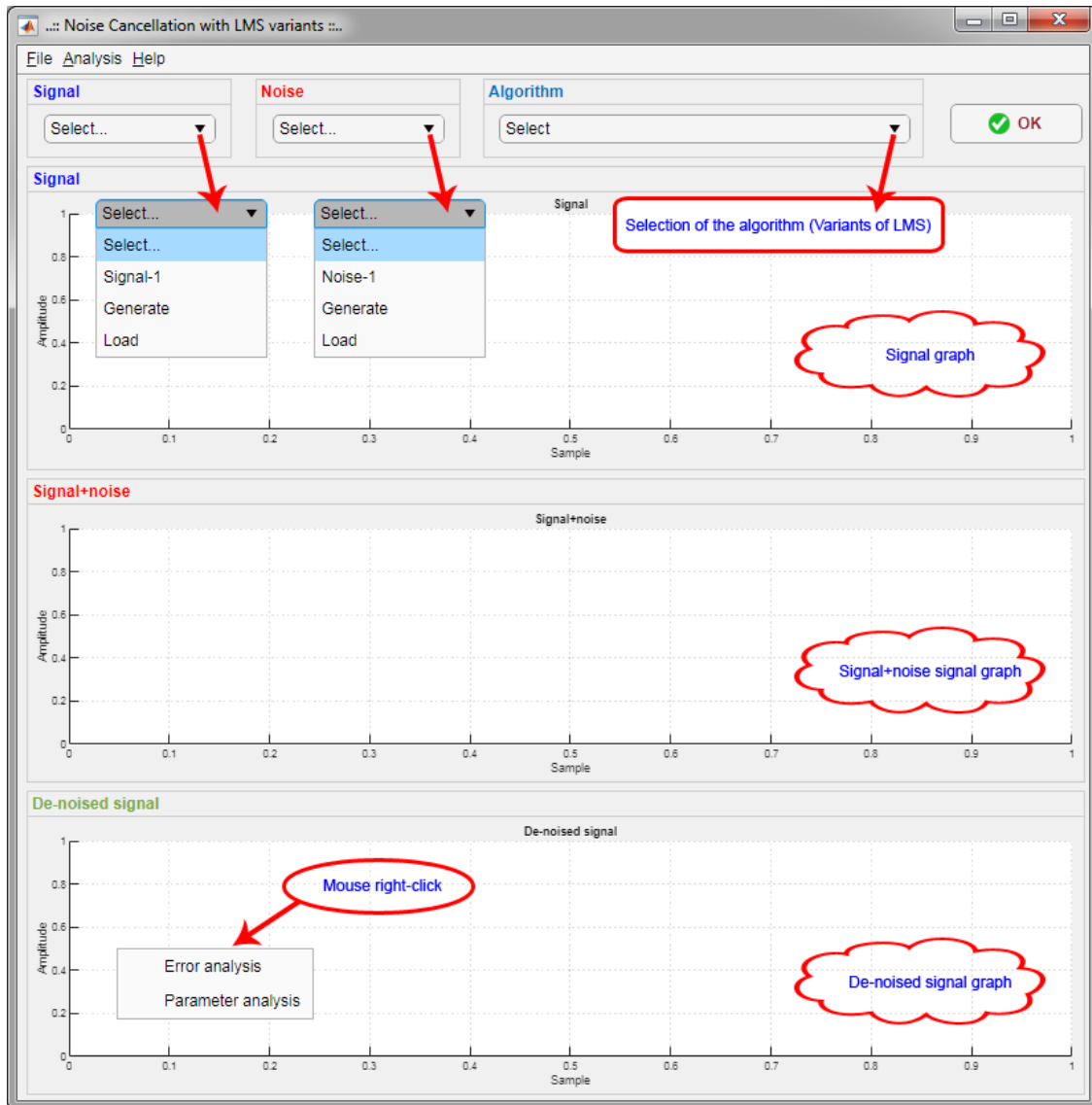
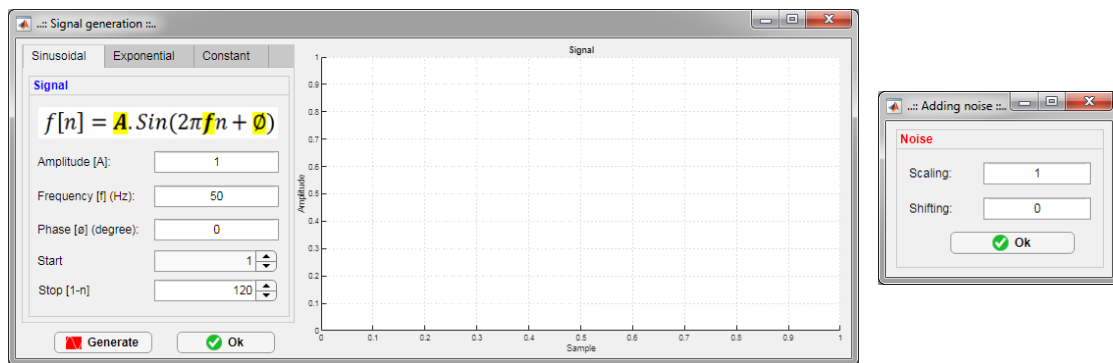


Figure 4:
The main screen of designed application

Table 4. The menu items of designed application

File	Analysis	Help
File Analysis Help	Analysis Help	Help
New Ctrl+N	Single	Algorithms
Open Ctrl+O	Comparative Ctrl+K	About
Save Ctrl+S	Analysis Ctrl+T	
Print Ctrl+P	Error analysis Ctrl+E	
Exit	Parameter analysis Ctrl+P	



(a)

(b)

Figure 5:
The signal and noise generation screens

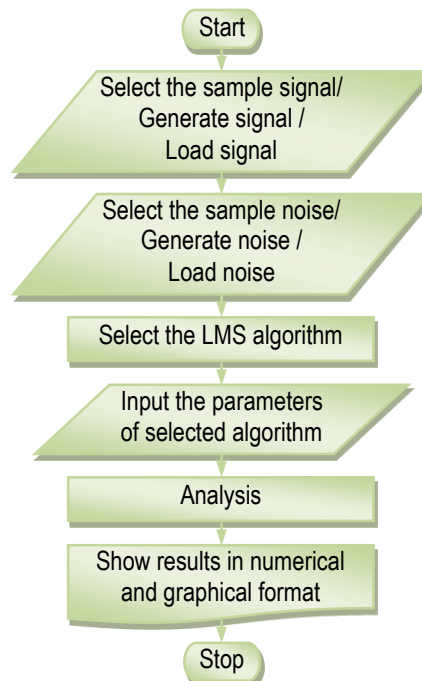
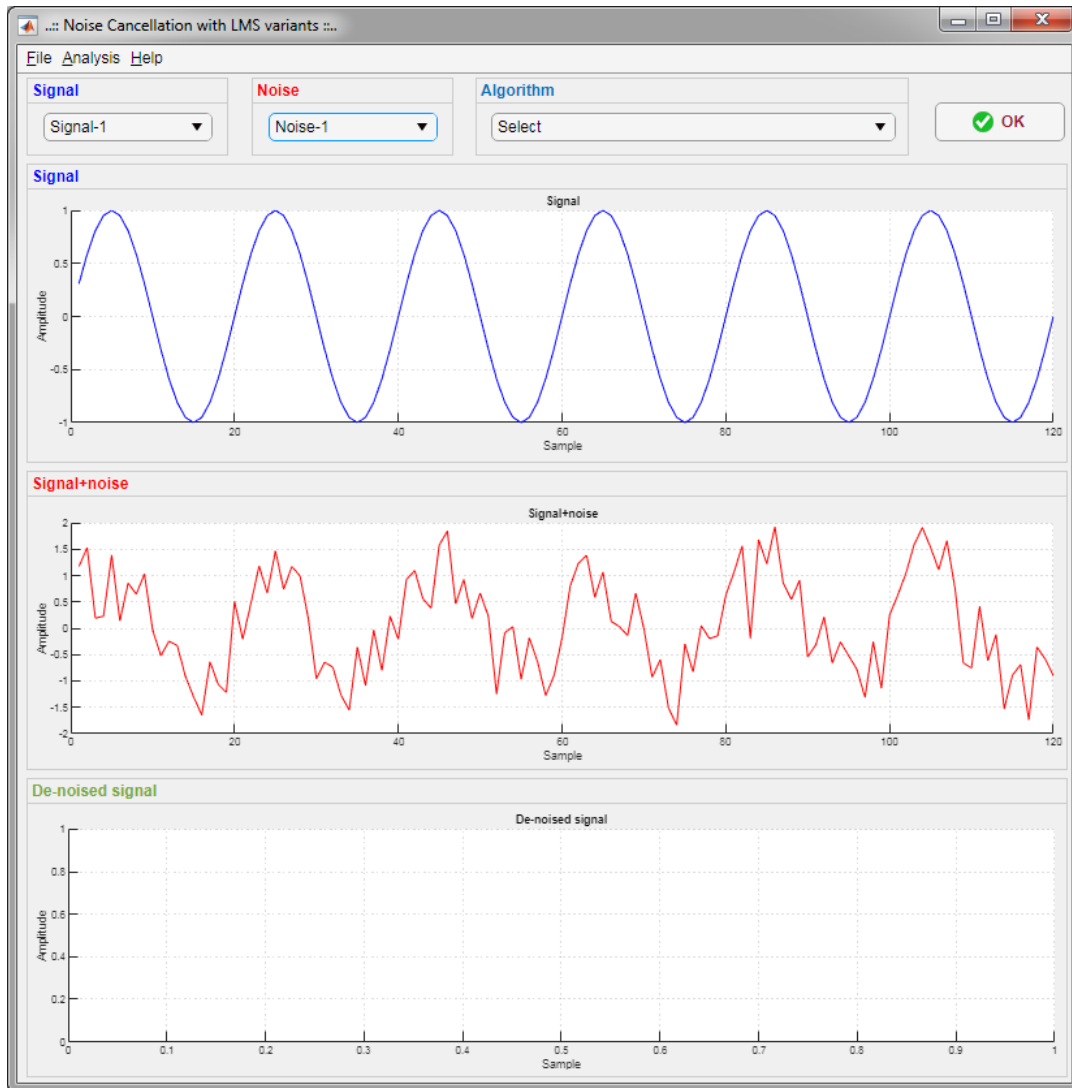


Figure 6:
The flowchart of single analysis process in designed application

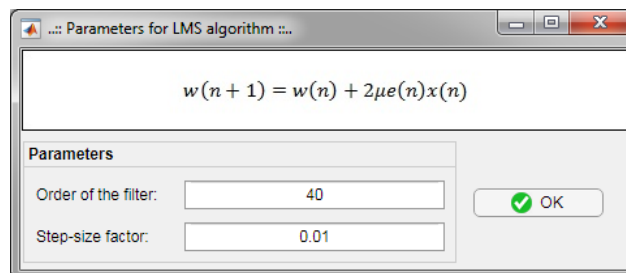
For example; when *Signal-1* (x signal) and *Noise-1* (n_0 noise) (Poularikas, 2015) items are selected, the screenshot in Fig. 7a is obtained.

$$x(n) = 1.0\text{Sin}\left(\frac{\pi}{10}n\right) \quad (17)$$

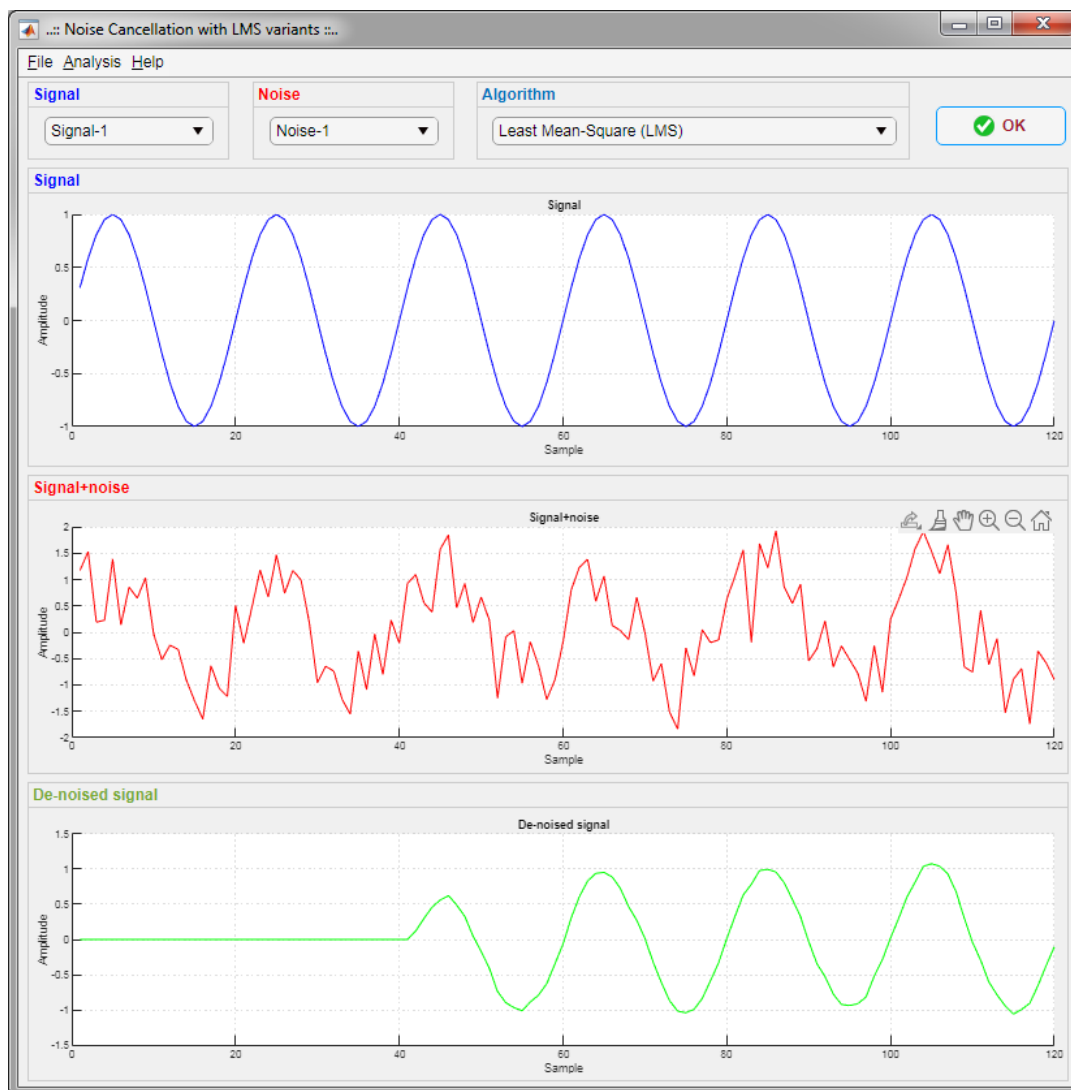
$$n_0(n) = 2.0\left(\text{random}(0,1) - \frac{1}{2}\right) \quad (18)$$



(a)



(b)

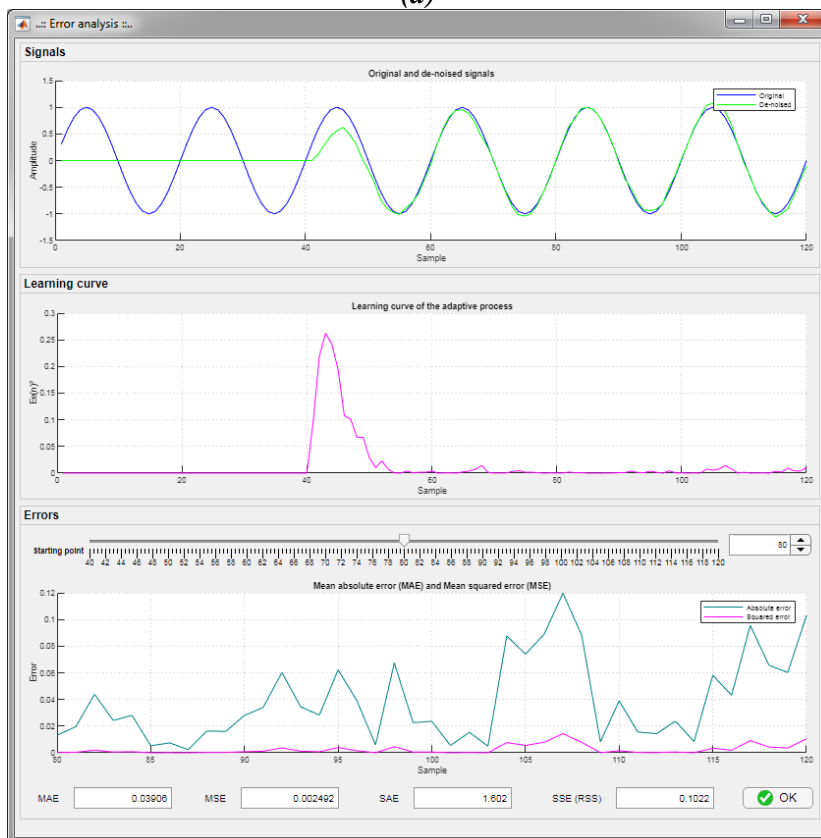


(c)

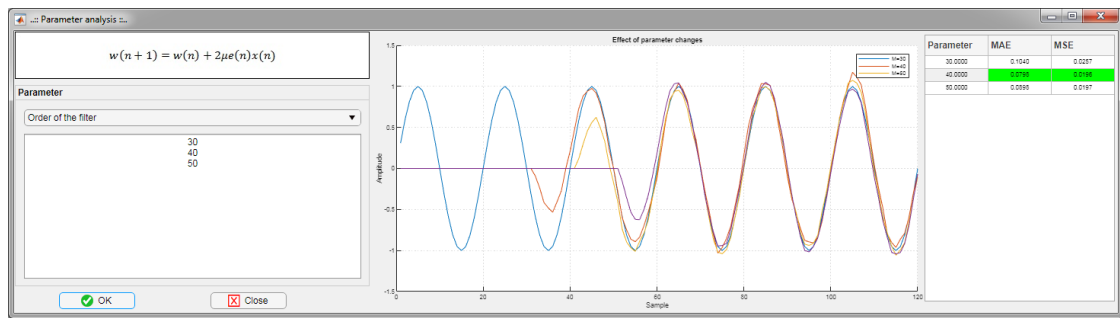
When "Least Mean-Square (LMS)" is selected from the "Algorithm" drop-down list, the screen for setting the related algorithm parameters appears (Fig. 7b). After the related parameters are entered, the result screen in Fig. 7c is obtained. When the mouse is right-clicked on the "De-noise signal" graph and "Error analysis" is selected, the screen in Fig. 7d appears. This screen displays "learning curve", "absolute error" and "squared error" graphics and mean absolute error (MAE), mean squared error (MSE), sum absolute error (SAE) and sum squared error (SSE) performance values. In addition, by changing the starting point of the samples with related slider or spinner, the change of this error values can be monitored (Fig. 7e). On the other hand, if "Parameter analysis" is selected, comparative performance analysis can be performed by entering changeable parameter values of the related algorithm (Fig. 7f-g). In addition to the comparative performance graphics, the minimum values in the relevant error table are automatically colored on this screen. For "Comparative analysis", algorithms are selected on the algorithm selection screen, their parameters are set and the comparison screen in Fig. 7h appears. On this screen, the MSE values of each algorithm are listed in the table, and de-noised signals and error graphs are also plotted comparatively. Also, re-analysis can be made by changing the parameters (Fig. 7h).



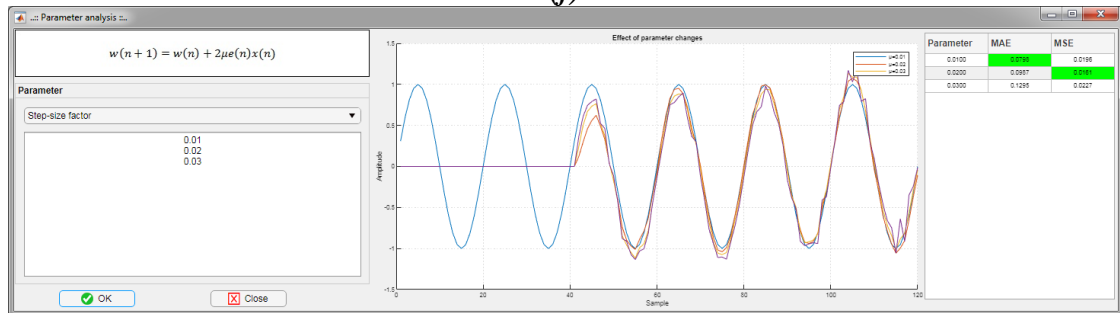
(d)



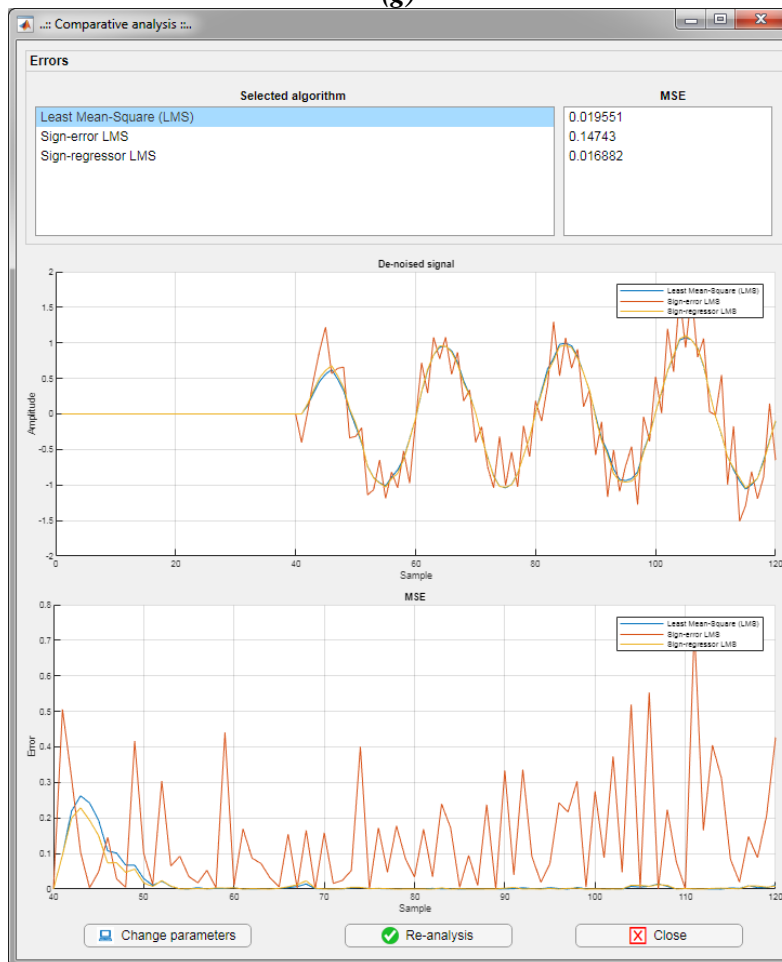
(e)



(f)



(g)



(h)

Figure 7:
Screenshots of the sample applications

Summarization of comparison this study with similar studies in noise cancellation field are given below:

- Generally, several methods or LMS variants are examined and performance comparisons are made in the studies. In the application developed in this study, there are many LMS variants, and single or comparative analysis can be performed.
- In the studies, methods/methods are usually encoded in MATLAB environment (or ready-made functions are used) or SIMULINK models are prepared. In this study, an interactive interface is designed for the methods.
- In some studies, hardware applications were carried out using FPGA or DSP kits. The hardware implementation is not included in the study in this article.
- The general comparison of the simulator developed in the reference (Maddala, 2010) and the simulator designed in this study is given in Table 4.

Table 4. The general comparison of designed application with other simulator

Properties	(Maddala, 2010)	This study
Used software	MATLAB GUI	MATLAB App Designer
The number of methods	Few	Many
Menus (open, save, print etc.)		√
Method description		√
Sample signals	√	√
Generate signal		√
Load signal		√
Generate noise		√
Load noise		√
Other applications with LMS	√	
Parameters settings of methods	√	√
Graphic of signal	√	√
Graphic of learning curve		√
Graphic of error	√	√
Error values		√
Error analysis		√
Parameter analysis		√
Comparative analysis		√

The interactive application developed has features that can be used both in practice and education. For practical applications, the most appropriate algorithm and parameter values can be determined by analyzing the relevant noisy signals (by creating or loading) in the simulator. It can be used effectively in the field of education thanks to its capabilities such as including lectures, showing parameter effects in detail, and making comparisons.

6. CONCLUSIONS

In this study, an application/simulator was carried out for noise cancellation, in one of the most important application areas of adaptive filters. With this application, which includes many variants of the LMS algorithm; the sample, user-defined or loading noisy signals can be de-noised easily and quickly with the selected algorithm or algorithms. It is obvious that LMS algorithms and noise cancellation processes can be better understood by using the developed interactive application especially in the field of engineering education. The ability of the simulator to give opportunity to students/users to examine lectures, change algorithm parameters, and observe single or comparative results both numerically and graphically will contribute greatly to learning and comprehension. As a future study, it is aimed to add more LMS variants to the application/simulator and move it to the web environment.

CONFLICT OF INTEREST

Author approves that to the best of his knowledge, there is not any conflict of interest or common interest with an institution/organization or a person that may affect the review process of the paper.

AUTHOR CONTRIBUTION

Fahri Vatansever has full responsibility for all parts of the paper.

REFERENCES

1. Bismor, D., Czyz, K., Ogonowski, Z. (2016) Review and comparison of variable step-size LMS algorithms, *The International Journal of Acoustics and Vibration*, 21, 24-39. doi: <https://doi.org/10.20855/ijav.2016.21.1392>.
2. Diniz, P.S.R. (2020) *Adaptive Filtering: Algorithms and Practical Implementation*, 5th ed., Springer, Switzerland.
3. Dixit, S., Nagaria, D. (2017) LMS adaptive filters for noise cancellation: A review, *International Journal of Electrical and Computer Engineering*, 7(5), 2520-2529. doi: <https://doi.org/10.11591/ijece.v7i5.pp2520-2529>
4. Farhang-Boroujeny, B. (2013) *Adaptive Filters: Theory and Applications*, 2nd ed., Wiley.
5. Goel, P., Chandra, M. (2019) FPGA implementation of adaptive filtering algorithms for noise cancellation - A Technical survey, In: Nath, V., Mandal, J. (eds) Proceedings of the Third International Conference on Microelectronics, Computing and Communication Systems. *Lecture Notes in Electrical Engineering*, 556, Springer, Singapore. https://doi.org/10.1007/978-981-13-7091-5_42.
6. Gorriz, J.M, Ramirez, J., Cruces-Alvarez, S., Puntonet, C.G., Lang, E.W., Erdogmus, D. (2009) A Novel LMS algorithm applied to adaptive noise cancellation, *IEEE Signal Processing Letters*, 16(1), 34-37. doi: <https://doi.org/10.1109/LSP.2008.2008584>.
7. Guda, M., Gasser, S., El Mahallawy, M.S. (2014) MATLAB simulation comparison for different adaptive noise cancelation algorithms, *The International Conference on Digital Information, Networking, and Wireless Communications (DINWC)*, Czech Republic, 68-73.
8. Hatun, M., Vatansever, F. (2016) Differential equation solver simulator for Runge-Kutta methods, *Uludağ University Journal of The Faculty of Engineering*, 21(1), 147-164. doi: <https://doi.org/10.17482/uujfe.70981>.
9. Haykin, S. (2014) *Adaptive Filter Theory*, 5th ed., Pearson, USA.
10. Kaçar, S., Çankaya, İ. (2010) Volterra serileri metodu ile doğrusal olmayan sistemlerin frekans boyutunda analizi için .NET tabanlı arayüz tasarımı, *Dokuz Eylül Üniversitesi Mühendislik Fakültesi Fen ve Mühendislik Dergisi*, 12(3), 87-102.
11. Kaçar, S., Çankaya, İ. (2012) MATLAB ve ASP.NET tabanlı web arayüzü kullanılarak doğrusal olmayan sistemlerin analizi, *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi*, 27(4), 795-806.
12. Kaçar, S., Boz, A.F., Arıcıoğlu, B., Tekin, H. (2017) PID denetleyici uygulamaları için yeni bir online deney seti tasarımı, *Sakarya Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, 21(1), 34-46. doi: <https://doi.org/10.16984/saufenbilder.284220>.

13. Lee, J-H., Ooi, L-E., Ko, Y-H., Teoh, C-Y. (2017) Simulation for noise cancellation using LMS adaptive filter, *IOP Conf. Series: Materials Science and Engineering*, 211, 012003, doi: <https://doi.org/10.1088/1757-899X/211/1/012003>.
14. Maddala, S. (2010) A simulator for depicting and comparing adaptive algorithms in signal processing, *2010 International Conference on Signal and Image Processing*, Chennai, 515-520. doi: <https://doi.org/10.1109/ICSIP.2010.5697530>.
15. Maurya, A.K. (2018) Cascade–cascade least mean square (LMS) adaptive noise cancellation, *Circuits Syst Signal Process*, 37, 3785–3826. doi: <https://doi.org/10.1007/s00034-017-0731-2>.
16. Maurya, A.K., Agrawal, P., Dixit, S. (2019) Modified model and algorithm of LMS adaptive filter for noise cancellation, *Circuits Syst Signal Process*, 38, 2351-2368. doi: <https://doi.org/10.1007/s00034-018-0952-z>.
17. Morales, L.G. (Ed.) (2011) *Adaptive Filtering Applications*, InTech, Croatia.
18. Poularikas, A.D. (2015) *Adaptive Filtering*, CRC Press.
19. Romero, I., Geng, D., Berset, T. (2012) Adaptive filtering in ECG denoising: A comparative study, *2012 Computing in Cardiology*, Krakow, 45-48.
20. Qureshi, R., Rizvi, S.A.R., Musavi, S.H.A., Khan, S., Khurshid, K. (2017) Performance analysis of adaptive algorithms for removal of low frequency noise from ECG signal, *2017 International Conference on Innovations in Electrical Engineering and Computational Technologies (ICIEECT)*, Karachi, 1-5. doi: <https://doi.org/10.1109/ICIEECT.2017.7916551>.
21. Sankar, A.B., Kumar, D., Seethalakshmi, K. (2010) Performance study of various adaptive filter algorithms for noise cancellation in respiratory signals, *Signal Processing: An International Journal (SPIJ)*, 4(5), 267-278.
22. Sayed, A.H. (2008) *Adaptive Filters*, Wiley, USA.
23. Taha, T.M.F., Adeel, A., Hussain, A. (2018) A Survey on techniques for enhancing speech, *International Journal of Computer Applications*, 179(17), 1-14. doi: <https://doi.org/10.5120/ijca2018916290>.
24. Thakor, N.V., Zhu, Y-S. (1991) Applications of adaptive filtering to ECG analysis: noise cancellation and arrhythmia detection, *IEEE Transactions on Biomedical Engineering*, 38(8), 785-794. doi: <https://doi.org/10.1109/10.83591>.
25. The MathWorks Inc. (2018) MATLAB , App Designer.
26. Vatansever, F., Yalcin, N.A. (2017) e-Signals & Systems: A web-based educational tool for signals and systems, *Computer Applications in Engineering Education*, 25(4), 625-641. doi: <https://doi.org/10.1002/cae.21826>.
27. Vatansever, F., Hatun, M. (2019) The system stability software tool based on Routh-Hurwitz criterion, *Uludağ University Journal of The Faculty of Engineering*, 24(2), 229-238. doi: <https://doi.org/10.17482/uumfd.545361>.
28. Yalcin, N.A., Vatansever, F. (2016) A Web-based virtual power electronics laboratory, *Comput. Appl. Eng. Educ.*, 24(1), 71-78. doi: <https://doi.org/10.1002/cae.21673>.
29. Widrow, B., Glover, J.R., McCool, J.M., Kaunitz, J., Williams, C.S., Hearn, R.H., Zeidler, J.R., Donc, E., Goodlin, R.C. (1975) Adaptive noise cancelling: Principles and applications, *Proceedings of the IEEE*, 63(12), 1692-1716. doi: <https://doi.org/10.1109/PROC.1975.10036>.

- 30.** Widrow, B., Stearns, S.D. (1985) *Adaptive Signal Processing*, Prentice-Hall, USA.
- 31.** Zaknich, A. (2005) *Principles of Adaptive Filters and Self-learning Systems*, Springer, Germany.

