Journal of Information Hiding and Multimedia Signal Processing ©2023 ISSN 2073-4212
Ubiquitous International Volume **14**, Number **1**, March **2023**

Underwater Active Noise Cancellation Combining Kalman Filter with FxLMS

Md. Mostafizur Rahman[1,*], Sumonto Sarker[1,a], Md. Mehedi Islam[1,b]

[1] Department of Electronics and Communication Engineering (ECE),
Hajee Mohammad Danesh Science and Technology University,
Dinajpur-5200, Bangladesh

[*] Corresponding Author: mostafiz.rahman.r@gmail.com
[a] sumonto@hstu.ac.bd
[b] mehedi@hstu.ac.bd

Received December 2022; revised February 2023

Abstract. *Underwater environments are more challenging than that of terrestrial. The performance of a controller or the augmented system depends on the real measured data, so noise on data readings can be fatal. To effectively and adaptively control noise frequency, the Active Noise Cancellation (ANC) was developed. The primary focus of this paper is to design a system by modifying the parameters of FxLMS to reduce noise. The required equations is analyzed and discussed briefly. Moreover, the system is simulated in MATLAB, and the filtered result is analyzed. Based on the simulation results, this proposed model can filter out noisy signals, particularly when there is a significant variation in the data and no knowledge of the noise frequency that might affect sensor readings.*
Keywords: FxLMS, Kalman Filter, Signal Denoising, Signal Enhancement

1. **Introduction.** It is very challenging to eliminate noise from a signal in an underwater environment without losing some of its information. Acoustic active noise cancellation (ANC) got a lot of attention for separating noise from a noisy signal. In ANC, a 180-degree phase signal (anti-noise) is generated and used to interfere destructively with the unnecessary noise. Bernard Widrow et al. pioneered the core concept [1][2].

Various methods have been proposed in order to improve the performance of the ANC. RLC, LMS, and their variants (NLMS, VLMS, and so on) are popular because they have fewer complications [1]. The RLS algorithm is known for its fast rate of convergence. However, the algorithm's ability to track estimation can be compromised due to its reliance on the model, input data, and correlation matrix. As the computation progresses, the correlation matrix may lose its property and become unstable, resulting in explosive divergence. This instability can cause the algorithm to fail in accurately tracking the estimation [3]. As a result, the LMS and its successor algorithms are the most widely used algorithms. The Filtered-X LMS (FxLMS) algorithm is a simple variant of the LMS algorithm, which was developed independently in the context of adaptive control systems, and originally introduced as a modification in applications where an intervening system exists in the error path [4] [5].

The LMS algorithm, which is based on the steepest descent method, cannot produce an accurate anti-noise signal as it does not consider secondary path effects. In contrast, the FxLMS algorithm is computationally simple and accounts for secondary path effects, making it a viable option for ANC applications [6]. However, several other ANC algorithms have been proposed to improve convergence properties, such as ANC systems in the frequency domain, RLS-based algorithms like FxRLS and FxFTF, Lattice ANC systems, and IIR filter-based LMS algorithms like FuRLMS and filtered-v algorithms. Despite their advantages, these methods may suffer from inherent stability problems, computational requirements, and numerical instability problems [7]. For these reasons, FxLMS remains a promising option for ANC applications.

The step size is the most inherent feature of the Least Mean Squares (LMS) algorithm that FxLMS inherited, and it requires careful adjustment. The Small step size, required for small excess mean square error, results in slow convergence. Large step size, needed for fast adaptation, may result in loss of stability. For controlling the step size and making it variable rather than fixed, we used the Kalman filter. Kalman filter was proposed by R. E. Kalman in 1960 [8] is popular for having easy computation, memory requirements and good capability on overcoming noises. There are various types of Kalman Filter, such as standard Kalman Filter, Extended Kalman Filter, Unscented Kalman Filter etc [9]. The paper used standard Kalman filter since it contains enough part of equation for noise reducing.

The paper is organized in the following way: Section II presented all the necessary equations of FxLMS and linear Kalman filter, and also devoted to the developing of a modified FxLMS with Kalman filter to reduce active noises. Section III is devoted to the simulation and discussions of the obtained results. The conclusion section closes the paper.

2. Materials and Methods.

The aim of this paper is to demonstrate a simulation, where we proposed a noise reduction model by modifying existing FxLMS algorithm, employing Kalman filter. The step size of FxLMS needs to be carefully adjusted. We replaced step size μ with kalman gain in each iteration. There are two step sizes in FxLMS, one in LMS part and another in the secondary noise path. So, in that case, the whole simulation based on three assumptions: using μ during LMS and Kalman-Gain at secondary noise path, using Kalman-Gain during LMS and μ at secondary noise path, and using Kalman-Gain during both in LMS and in secondary noise path. Finally, we compared data with novel FxLMS algorithm based on Signal to noise ratio (SNR). Our experimental data shows that after using kalman, it filtered noisy signal more efficiently than before. All the simulations are carried away by MATLAB R2015b on a Windows 10 PC (x64) with an Intel i3-7100U CPU and an Nvidia GeForce 920MX GPU card.

2.1. FxLMS Algorithm.

From data preparation to final evaluation, total methodology is divided into several subsections. Widrow, Shur and Shaffer proposed the integration of a secondary path model in the reference signal path (from speaker to error microphone) [11]. Figure 1 shows the block diagram of FXLMS algorithm on how the noise reduction algorithm works and the definition of each symbol is shown in Table 1.

ANC has commonly used in two different configurations of the FxLMS algorithm, one is a feedback ANC approach proposed by Olson and May in 1953 [12], in this model, a microphone is used as an error sensor, and also as a reference sensor. The second is a feed-forward ANC approach, which uses two sensors, an error sensor, and a reference sensor. This setup is used for narrow-band noise control using a non-acoustic reference sensor [13].

Figure 1 shows the general how the feedforward approach uses a different microphone to measure the signal at the output. It is significant to know the software elements that are part of the ANC controller, these are the LMS adaptive algorithm that updates the

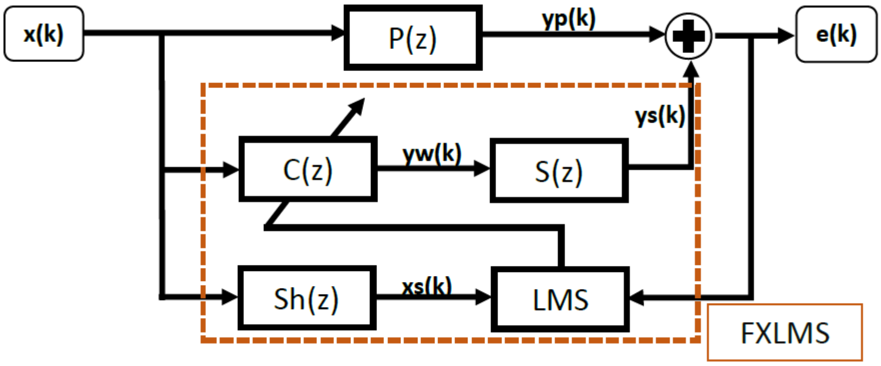

FIGURE 1. Block diagram of FxLMS algorithm.

TABLE 1. Symbols and Definitions for Figure 1

Symbols	**Definitions**
$x(k)$	Noise signal
$xs(k)$	Noise signal combined with assumed Sh(z) based on S(z)
$P(z)$	Primary path transfer function
$yp(k)$	Primary noise signal at the error microphone
$e(k)$	Modified error signal
$S(z)$	Secondary path transfer function
$C(z)$, $Sh(z)$	Controller for the FxLMS algorithm
$yw(k)$	Generated noise based on C(z) controller
$ys(k)$	Output of adaptive filter

coefficients of the W(z) adaptive filter, which in this case is represented as an FIR filter. The C(z) filter represented the secondary path estimation or the transfer function between the secondary source (control source) and the error microphone. To derive the FxLMS algorithm, a similar method of the LMS algorithm is utilized but with the steepest descent, the following update equation can lead to this minimization:

$$W_{New} = W_{Old} + \nabla \mathrm{J(n)} \tag{1}$$

Where W is the controller weight error,
µ is an adaption step size (scalar),
J(n) is the power of error signal.
The derivation of ∇J(n),

$$J\,(n) = E\{e^2(n)\}$$

Where E{.} denotes statistical expectation operator and E{.} is a theoretical function. To avoid this operator, J(n) is approximated by

$$J\,(n) \approx e^2(n)$$

Then, estimate ∇J(n) as follows,

$$\nabla J\,(n) = \nabla e^2(n)$$
$$\nabla J\,(n) = 2e(n)\nabla e(n) \tag{2}$$

Now to estimate ∇e(n), the derivation is as follows based on the block diagram,

$$e(n) = d(n) + s(n) * y(n)$$

Where s(n) is the secondary path impulse response.

$$\nabla e(n) = s(n) * \nabla y(n) \tag{3}$$

Now to estimate $\nabla y(n)$, the derivation is as follows based on the block diagram,

$$y(n) = W^T x(n)$$

Where W is the controller weight vector and x is the reference signal tap vector (of the same length as the controller length)

Now $\nabla y(n)$ can be expressed by,

$$\nabla y(n) = \frac{\partial y(n)}{\partial W}$$

$$\nabla y(n) = x(n) \tag{4}$$

Substitute (4) into (3)

$$\nabla e(n) = s(n) * x(n) \tag{5}$$

Substitute (5) into (2)

$$\nabla J(n) = 2e(n) .s(n) * x(n) \tag{6}$$

Substitute (6) into (1)

$$W_{New} = W_{Old} + 2e(n) .s(n) * x(n) \tag{7}$$

The reference signal is filtered by ŝ(n) before passing through the standard LMS algorithm. Therefore, resulting the compensation for secondary path. ŝ(n) should be estimated through off-line or online secondary path techniques. If ŝ(n) denotes an estimate of s(n), then

$$W_{New} = W_{Old} + 2e(n) .\hat{s}(n) * x(n)$$

OR

$$W_{New} = W_{Old} + 2e(n) .x_f(n)$$

The stability of the FxLMS algorithm is highly dependent on the x_f(n) power where it directly proportional to the step-size μ. So, Step-size is indirectly proportional to the steady state performance.

FxLMS is simple, fast, and surprisingly robust. Despite its straightforwardness, FxLMS acquired the most central feature of the Least Mean Squares (LMS) algorithm is the step size, and it undoubtedly requires precise adjustment. To properly control step size, we utilized the Kalman filter.

2.2. **Kalman Filter.** The paper will use a standard Kalman filter since it contains enough parts of the equation for noise cutting. Kalman Filter has two parts, the predicted part, and the update part. The standard Kalman Filter equation is shown in (8)-(12).

Predict:

$$\bar{x}_{t|t-1} = F_t \bar{x}_{t-1|t-1} + B_t u_t \tag{8}$$

$$P_{t|t-1} = F_t P_{t-1|t-1} F_t^T + Q_t \tag{9}$$

Update:

$$\bar{x}_{t|t} = \bar{x}_{t|t-1} + K_t(y_t - H_t\bar{x}_{t|t-1}) \tag{10}$$

$$K_t = P_{t|t-1}H_t^T(H_tP_{t|t-1}H_t^T + R_t)^{-1} \tag{11}$$

$$P_{t|t} = (1 - K_tH_t)\ P_{t|t-1} \tag{12}$$

where $\bar{x}$ is estimated state, F is state transition matrix,u is control variables, B is control matrix, P is state variance matrix, Q is process variance matrix, y is measurement variables, H is measurement matrix, K is Kalman gain, R is measurement matrix, $t|t$ is current time period, $t-1|t-1$ is previous time period, and $t|t-1$ is intermediate steps.

To implement Kalman Filter algorithm, so that it can be used to reduce noise of sensor-readings, some adjustments for the conditions are needed. Those adjustments are as follows [14].

2.2.1. *Predicting the state.* On this stage, adjustments are done in (11) by giving the score $\mathrm{F}_t = 1$ because there is no state transition. Thus, reducing the system's input component B_t because the used system does not have any input u_t. The adjusted equation is,

$$x_{t|t-1} = x_{t-1|t-1} \tag{13}$$

2.2.2. *Predicting the error.* Since $\mathrm{F}_t = 1$, then (9) becomes,

$$P_{t|t-1} = P_{t-1|t-1} + Q_t \tag{14}$$

2.2.3. *Updating the state value.* From (10), $\mathrm{H}_t = 1$ since the sensor data that will be filtered is only consisted of one sensor reading. Hence, the equation can be written as,

$$\bar{x}_{t|t} = \bar{x}_{t|t-1} + K_t(y_t - \bar{x}_{t|t-1}) \tag{15}$$

2.2.4. *Calculating the gain of Kalman.* Since H_t=1, then (11) can be written as,

$$K_t = P_{t|t-1}(P_{t|t-1} + R)^{-1} \tag{16}$$

2.2.5. *Updating the error value.* Since H_t=1, then (12) can be written as,

$$P_{t|t} = (1 - K_t)\ P_{t|t-1} \tag{17}$$

The Kalman Filter equation can be modified to reduce sensor reading noise once the necessary adjustments have been performed. The weights given to the data and the current-state estimate are represented by the Kalman-gain (at eq. 16), which can be "adjusted" to get a specific performance. We replaced the step-size μ in the FxLMS with this Kalman gain so that the step size is flexible according to the signal elements rather than being fixed.

As we replaced step size, μ with Kalman-gain in the original FxLMS algorithm, we needed to declare some necessary variables to calculate Kalman-gain out of the noisy signal. During calculating Kalman-gain, we had to specify the values for Q (process noise covariance) and R (measurement noise covariance).

The value of Q and R are chosen according to the system operations. Covariance Q and R states may not be in general observable but the measurements should be related to the states [16].

Q, the process noise covariance, contributes to the overall uncertainty. When Q is large, the Kalman Filter more closely tracks large changes in the data than when Q is small. The measurement noise covariance R determines how much information is used from the measurement. When R is large, the Kalman Filter considers the measurements to be

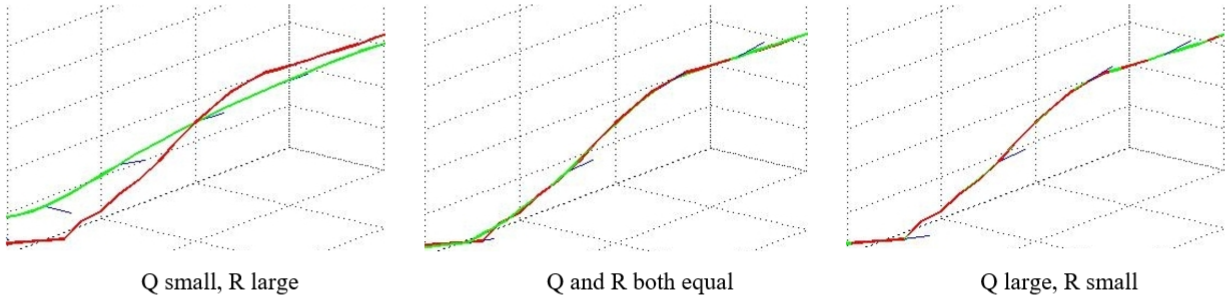

Figure 2. Relations between Q and R.

inaccurate. The three images below visualize the positional data. The red lines represent the measurement data, the green lines are the estimated states. [17]

We need to balance between Q and R according to our needs. The vast majority of noise estimation methods were designed with the assumption of uncorrelated state and measurement noise in mind [18]. For example, if kalman used in tracking cars on a road, then the constant velocity model should be reasonably good, and the entries of Q should be small. Else if it is used tracking people's faces, they are not likely to move with a constant velocity, so the Q need to cranked up [19].

In [14], author used kalman filter to denoising signals. During their operations, they discovered that the greater the difference between R and Q, the greater the mean error values. Furthermore, the same R and Q values result in similar value of mean error, whatever the values of R and Q. According to their analysis, the best parameters that provide results with their original data characteristics have mean error values ranging from 40 to 55 in the table below.

They experimentally showed that in case of signal denoising, the kalman filter yields best results if the ratio between R and Q are in 100:1. Therefore, in our operation, we kept R, Q ratio 100:1 too.

Table 2. Ratio between R and Q and their yielding mean error

No. of Analysis	**Kalman Filter Parameter Value**		**R and Q Ratio**	**Mean Error**
	R	**Q**		
1	1	1	1	26.0677
2	1	0.1	10	44.7392
3	1	0.01	100	53.4466
4	10	0.1	100	53.4541
5	100	0.1	1000	56.9959

The flowchart of our modified FxLMS showed at above figure. When we replace step size μ with kalman gain at secondary path operation, we get the best output by far.

The reference signal $x(n)$ travels from the source to the sensor through the fluid medium P(z), where the sensor measures the resulting noise as p(n). To reduce this noise, a controller W(z) is used to generate another noise signal y(n), with the aim of destructively interfering with the noise signal x(n). It means that the controller has to be a model of the propagation medium P(z). The least mean square algorithm is employed to adjust the controller weight/coefficients. However, there is a fluid medium S(z) between the actuator and sensor, known as the secondary propagation path, which must also be taken

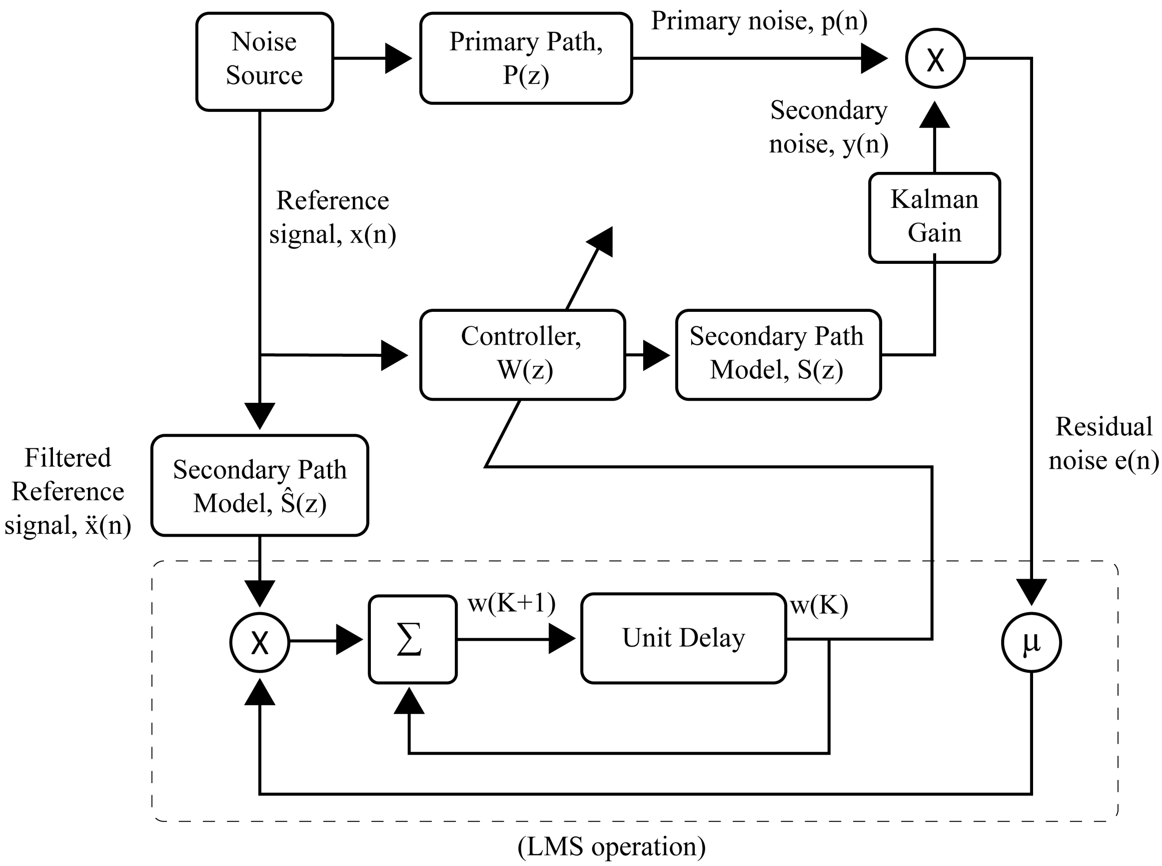

Figure 3. Modified FxLMS algorithm.

into account. Therefore, the adjustment process must be compensated for, taking into consideration the estimate of $\hat{S}(z)$.

3. **Results and Discussions.** In this research, we generate a noisy signal, which will then be filtered using two algorithms: the conventional FxLMS algorithm and a modified version of the FxLMS algorithm developed in this study. The effectiveness of the modified algorithm will be evaluated and compared to the performance of the conventional algorithm. The step size, μ is used in two major operations in the FxLMS algorithm; one in the LMS calculating process which is run in the secondary path, and another one in the whole secondary noise path of FxLMS. So, in that case, we run whole simulation based on three assumptions: a) Use μ during LMS and Kalman-Gain at secondary noise path, b) Use Kalman-Gain during LMS and μ at secondary noise path, c) Use Kalman-Gain during both LMS and secondary noise path.

In section (a), where we used μ during LMS and Kalman-Gain at secondary noise path, yields better SNR value. In section (b), where we used Kalman-Gain during LMS and μ at secondary noise path, we almost got same SNR in novel FxLMS and in our modified FxLMS algorithm. In section (c), we used Kalman-Gain during both LMS and secondary noise path. Here we got slightly better output than that of in section (b), but in some samples, we got less SNR than novel FxLMS. So, the operation here is considered unstable.

So, we concluded, if we don't change μ during LMS operation and use Kalman-gain as a replacement of μ during secondary noise path of FxLMS, we will get more stable and better output. In figure 4, existing FxLMS and modified FxLMS plots are shown. In

TABLE 3. SNR analysis between three assumption: a) Use μ during LMS and Kalman-Gain at secondary noise path, b) Use Kalman-Gain during LMS and μ at secondary noise path, c) Use Kalman-Gain during both LMS and secondary noise path.

Use μ during LMS and Kalman-Gain at secondary noise path			**Use Kalman-Gain during LMS and μ at secondary noise path**			**Use Kalman-Gain during both LMS and secondary noise path**		
no.	**SNR of FxLMS**	**SNR of modified FxLMS**	**no.**	**SNR of FxLMS**	**SNR of modified FxLMS**	**no.**	**SNR of FxLMS**	**SNR of modified FxLMS**
1	12.41158	13.40830	**1**	11.48765	11.48765	**1**	12.39523	13.34103
2	11.50421	12.46322	**2**	11.06984	11.06986	**2**	11.62135	12.43485
3	12.10779	12.72214	**3**	10.97759	10.97759	**3**	12.02066	12.42624
4	11.52281	12.35170	**4**	11.56018	11.56018	**4**	11.15641	11.55346
5	12.16835	12.79602	**5**	11.73240	11.73242	**5**	10.33399	9.478464
6	12.23658	13.26459	**6**	11.81109	11.81109	**6**	10.11580	11.03627
7	11.09591	11.38486	**7**	11.73250	11.73251	**7**	11.80953	12.60883
8	11.64333	11.96308	**8**	11.64378	11.64377	**8**	12.36314	13.29479
9	11.64407	12.45536	**9**	11.31957	11.31932	**9**	11.59755	12.55345
10	11.10289	11.52138	**10**	11.52379	11.52379	**10**	11.36943	12.10035

this particular sample we have shown, having SNR of FxLMS is 11.706468 and modified FxLMS is 12.226385.

As having higher SNR means more information, at figure 4(b) we got best output. In 4(a), filtered signal lost more information than that of 4(b). If we analysis noise residue plots at figure 5, modified FxLMS got relatively less noise after each iteration, and at the end of discrete time T, noise figures are way smaller than before that indicates, our modified FxLMS capable of reducing noises much efficiently.

4. **Acknowledgment.** The authors gratefully acknowledge the Institute of Research and Training (IRT), Hajee Mohammad Danesh Science and Technology University for their support.

5. **Conclusion.** To conclude, this paper demonstrated an underwater acoustic active noise cancellation (ANC) system using Filtered-x LMS and kalman filter. Based on the simulation and test results, proposed model of modified Filtered-x LMS is able to reduce noise in received signals. Performance of modified FxLMS is best when we keep using step size μ during LMS and replace it with kalman gain at secondary noise path calculation. The future recommendations that can be taken into consideration is by using variations of kalman filter, multiple frequency tone testing, and also implementation towards the industrial system.

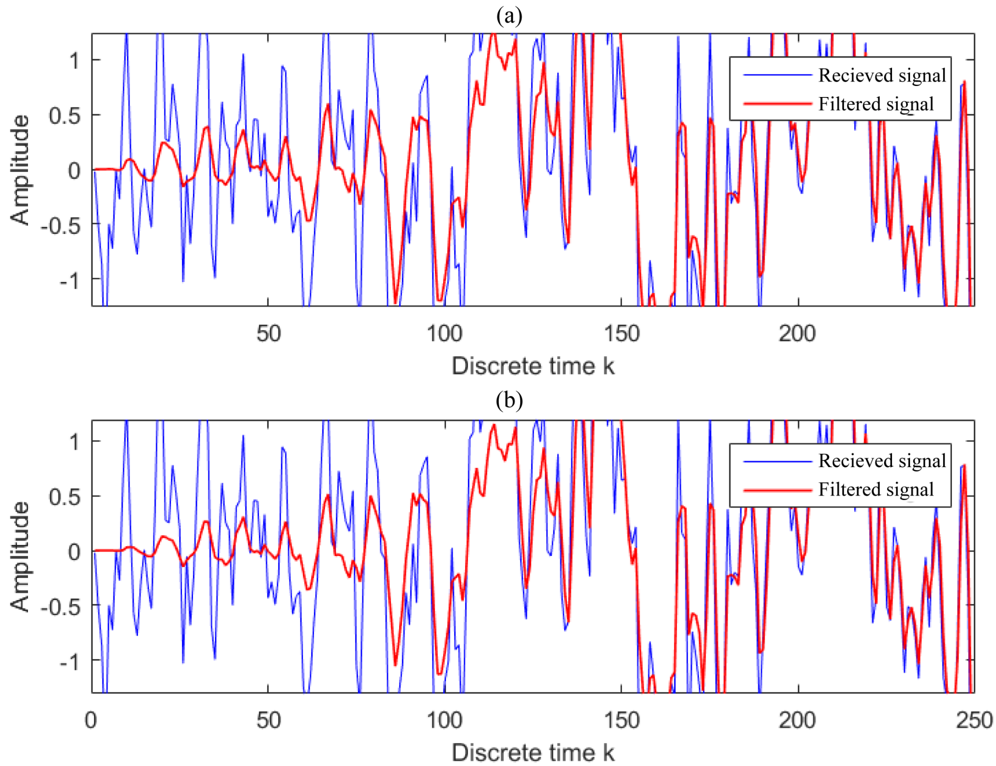

Figure 4. Plot analysis between received noisy signal and filtered signal of (a) FxLMS (SNR = 11.706468), (b) modified FxLMS (SNR = 12.226385).

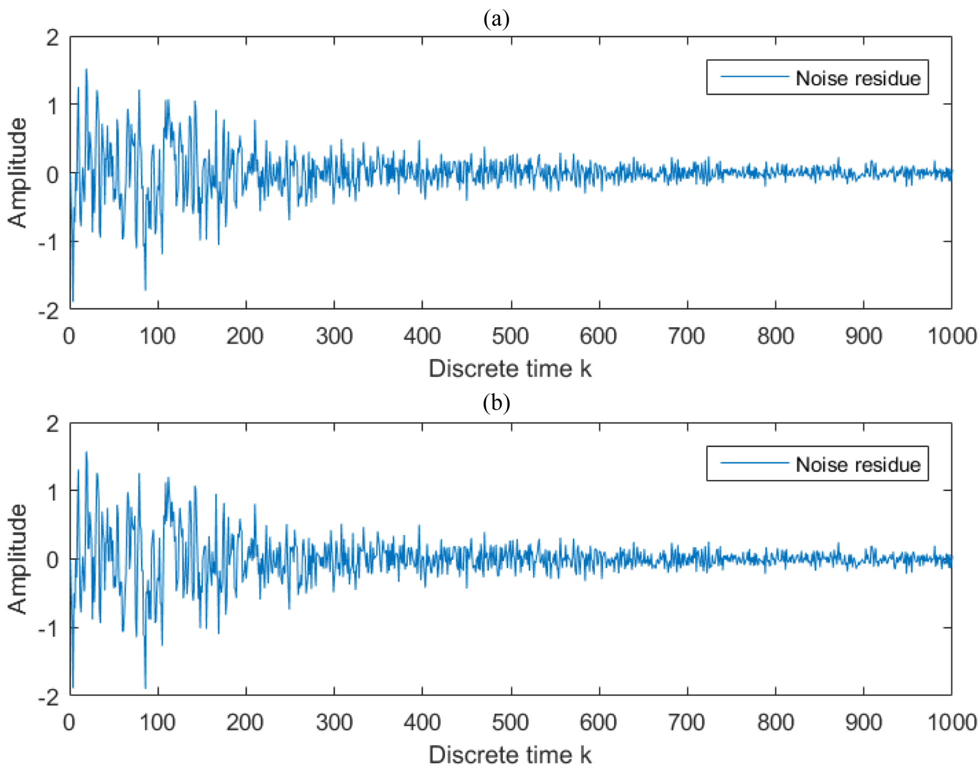

Figure 5. Plot analysis of noise residue after each filtering iteration of (a) FxLMS (SNR = 11.706468), (b) modified FxLMS (SNR = 12.226385).

REFERENCES

[1] Dewasthale, Mugdha M. and Kharadkar, R.D., "Acoustic Noise Cancellation Using Adaptive Filters: A Survey," in 2014 International Conference on Electronic Systems, Signal Processing and Computing Technologies, 2014. DOI: 10.1109/ICESC.2014.11

[2] M. Rajesh., J. Jeevamalar. and J. Jancirani, "Active noise reduction of automotive HVAC system using filtered LMS [Part-1 sound measurement]," in IEEE-International Conference On Advances in Engineering, Science and Management (ICAESM -2012), 2012, Available: https://ieeexplore.ieee.org/document/6216261.

[3] Madravam, Venu and Jagadeesha, S N and Yerriswamy, T, "An Analysis of LMS, NLMS and RLS Filters with Application to DOA Tracking," in 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), 2020, DOI: 10.1109/ICIMIA48430.2020.9074887.

[4] Haykin, Simon S, "Adaptive filter theory", Pearson Education India, 2008, Available: https://books.google.com.bd/books/about/Adaptive_Filter_Theory.html?id=MdDi_PF7gMsC&redir_esc=y.

[5] Mayyas, K. and Aboulnasr, T., "Comparative Study of the Filtered-X Lms and Lms Algorithms with Undermodelling Conditions," International Journal of Modelling and Simulation, vol. 22, pp. 159-166, 2002, DOI: 10.1080/02286203.2002.11442237.

[6] Hirave, Bageshree Pathak & Padma. P., "FxLMS Algorithm for Feed Forward Active Noise Cancellation," in First International Conference on Advances in Computer, Electronics and Electrical Engineering - CEEE, 2012, DOI: 10.15224/978-981-07-1847-3-203.

[7] Gupta, Sakshi Gaur and V. K., "A Review on Filtered-X LMS Algorithm," International Journal of Signal Processing Systems, vol. 4, no. 2, pp. 172-176, 2016, DOI: 10.12720/ijsps.4.2.172-176.

[8] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," Journal of Fluids Engineering, Transactions of the ASME, vol. 82, no. 1, pp. 35-45, 1960, DOI: 10.1115/1.3662552.

[9] Kumari, N., Kulkarni, R., Ahmed, M.R., Kumar, N., "Use of Kalman Filter and Its Variants in State Estimation: A Review," in Artificial Intelligence for a Sustainable Industry 4.0, Springer International Publishing, 2021, pp. 213-230, DOI: 10.1007/978-3-030-77070-9_13.

[10] K. Paliwal, L Alsteris, "Usefulness of phase in speech processing," in IPSJ Spoken Language Processing Workshop, Japan, 2003, Available: https://www.semanticscholar.org/paper/Usefulness-of-phase-in-speech-processing-Paliwal/56385f10a46eef3b2195ba7dfe8389b50211a30a.

[11] M. Bernard Widrow & Bilello, "Adaptive Inverse Control. - CONTROL ENG PRACTICE. 5. 1 - 6.," Control Engineering Practice, vol. 5, pp. 1-6, 1993, DOI: 10.1109/ISIC.1993.397732.

[12] May, Harry F. Olson and Everett G., "Electronic Sound Absorber," The Journal of the Acoustical Society of America, vol. 25, no. 6, pp. 1130-1136, 1953, 10.1121/1.1907249.

[13] Morgan, S. M. Kuo and D. R. "Active noise control: a tutorial review," Proceedings of the IEEE, vol. 87, no. 6, pp. 943-973, 1999, DOI: 10.1109/5.763310.

[14] Alfian Ma'arif, Iswanto, Aninditya Anggari Nuryono, Rio Ikhsan Alfian, "Kalman Filter for Noise Reducer on Sensor Readings," Signal and Image Processing Letters, vol. 1, no. 2, pp. 50-60, 2019, DOI: 10.31763/simple.v1i2.2.

[15] Wojcicki, Kamil and Milacic, Mitar and Stark, Anthony and Lyons, James and Paliwal, Kuldip, "Exploiting Conjugate Symmetry of the Short-Time Fourier Spectrum for Speech Enhancement," IEEE Signal Processing Letters, vol. 15, pp. 461-464, 2008, DOI: 10.1109/LSP.2008.923579.

[16] M. R. Ananthasayanam, "Tuning of the Kalman Filter Using Constant Gains," in Introduction and Implementations of the Kalman Filter, IntechOpen, 2018, DOI: 10.5772/intechopen.81795.

[17] D. K. Buckl, "Sensor Fusion using the Kalman Filter," TUM: The Entrepreneurial University, 2005, Available: https://campar.in.tum.de/Chair/KalmanFilter.

[18] Dunik, Jindrich and Straka, Ondřej and Kost, Oliver and Havlík, Jindřich, "Noise covariance matrices in state-space models: A survey and comparison of estimation methods—Part I," International Journal of Adaptive Control and Signal Processing, vol. 31, 2017, DOI: 10.1002/acs.2783.

[19] Cyberdyne, "Kalman filter in computer vision: the choice of Q and R noise covariances," 2014, Available: https://stackoverflow.com/questions/21245167/kalman-filter-in-computer-vision-the-choice-of-q-and-r-noise-covariances.