INTERNATIONAL JOURNAL OF LATEST TECHNOLOGY IN ENGINEERING, MANAGEMENT & APPLIED SCIENCE (IJLTEMAS)

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue I, January 2026

Noise Reduction and Signal Estimation For 5g Anntenna Using Least Mean Square (LMS) Algorithm and Kalman Filter

Bayem Donatus I, Alumona Theopilus

Department of Electronic and Computer Engineering, Faculty of Engineering, Nnamdi Azikiwe University Awka

DOI : https://doi.org/10.51583/IJLTEMAS.2026.150100019

Received: 05 January 2026; Accepted: 13 January 2026; Published: 24 January 2026

ABSTRACT

The Kalman filter code used in the active control system is described in detail in this thesis. In order to react to variations in the primary noise and 5G environment, traditional active noise management techniques typically use an adaptive filter, such as the filtered reference least mean square (FxLMS) algorithm. However, the weak convergence features of the FxLMS algorithm typically affect how well dynamic noise is reduced. This research suggests utilizing the Kalman filter in the active noise control (ANC) system to enhance the efficacy of noise reduction for dynamic noise. The Kalman filter is used effectively by the ANC application using a new dynamic ANC model. The numerical simulation shows that the proposed Kalman filter works better than the FxLMS approach in terms of convergence performance for handling dynamic noise. This suggests that the transition of the control filter has a higher degree of confidence than the observation function. When the effects of various mu and K values were examined, it was discovered that the LMS algorithms had a slow rate of convergence for mu = 0.01 and high starting error signals for both echo and noise cancellation that subsequently decreased. Although there was some improvement in the noise and echo-canceled signals, the residual noise and echo persisted. Error signals dropped more quickly and the convergence rate was better with mu = 0.05 than with mu = 0.01; this suggests more efficient cancellation. There was less lingering echo and noise in the resulting clearer signals, with echo and noise cancelled. Hybrid LMS algorithms exhibited rapid convergence at μ = 0.1, with error signals declining sharply, indicating effective cancellation. With little lingering interference, the noise-cancelled and echo-cancelled signals were noticeably clearer. Although the convergence was quite quick with mu = 0.5, there was a higher chance of instability, particularly for the LMS method. It is often advised to use a hybrid algorithm with a mu value of between 0.05 and 0.1. Achieving the ideal balance between convergence speed and stability requires proper mu tuning, which guarantees efficient cancellation without creating instability.

INTRODUCTION

The fifth-generation (5G) technology, which promises previously unheard-of speeds, capacity, and connection, is the result of wireless communication's growth. Managing and reducing noise, which can seriously impair signal quality and system performance, is one of the major issues facing 5G networks. For 5G antennas to guarantee high data speeds and dependable connectivity, noise reduction is consequently crucial (Araújo and Almeida, 2019). Massive machine-type communication, ultra-reliable low latency communication, and improved mobile broadband are the goals of 5G technology. To achieve these objectives, advanced signal processing methods are needed to manage the network's growing density and complexity. Noise is a major problem that can originate from a number of sources, such as ambient conditions, interference from other devices, and thermal noise. Any undesired signal that obstructs the intended communication signal is called noise. Noise can cause errors and lower service quality in wireless communication by distorting the sent signal (Uwaechia and Mahyuddin, 2019). To preserve the integrity of the data being communicated and to maximize the performance of communication systems, effective noise reduction techniques are crucial. With its promise of previously unheard-of data speeds, extremely low latency, and extensive connectivity, 5G technology represents a major turning point in the development of wireless communication. Applications ranging from augmented reality and Internet of Things (IoT) gadgets to driverless cars and smart cities will be supported by

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue I, January 2026

5G networks. Advanced antenna systems, such as large Multiple Input Multiple Output (MIMO) and beamforming technologies, which are essential to 5G infrastructure, are required to achieve these capabilities.

REVIEW OF RELATED LITERATURE

Srinivas et al. (2022) suggest a semi-blind estimator that uses one-bit Massive MIMO to mitigate the impact of a polluted pilot, which could significantly impair system performance. To enhance the CSI estimation, the semi-blind approach makes use of pilots and a few data symbols. Spectral efficiency and estimation accuracy are increased when data symbols are used in the channel estimation process. When compared to the pilot-based estimators, the suggested iteration-based semi blind one-bit massive MIMO algorithm performed better in terms of BER and MSE. Moreover, although utilizing a small number of pilot symbols, the spectral efficiency has been enhanced.

The Enhanced Channel Estimation for MIMO¬OFDM in 5G NR was examined by Taheri et al. in 2021. They believed that Two potential technologies to achieve high data rate transmission capabilities, excellent spectral efficiency, and greater robustness against multi-path fading are Orthogonal Frequency Division Multiplexing (OFDM) and Multiple Input Multiple Output (MIMO). An essential component of the MIMO OFDM system is the channel estimating approach, which is utilized to recover the originally sent signal by reducing the impact of the channel. Training-based, semi-blind, and blind channel estimation algorithms are the three primary categories into which the channel estimation techniques fall.

The study's primary goal was to choose and put into practice an appropriate channel estimation algorithm with a low degree of computational complexity that produces good performance. This study examined and carried out channel estimation in the MIMO¬OFDM system using block type pilot symbol layout. The Low Rank Approximation (LRA¬LMMSE) and Minimum Mean Square Error (MMSE) algorithms are the channel estimation techniques that have been examined. The performance of algorithms' Bit Error Rate (BER) and Block Error Rate (BLER) as well as their level of computational complexity are the primary subjects of this study.

MATLAB simulations based on BER and BLER vs Single to Noise Rate (SNR) are used to assess the efficacy of the developed techniques. The assessment takes into account each algorithm's level of computational complexity. According to the final results of several channel models, LRA¬LMMSE performs better than Linear Minimum Mean Square Error (LMMSE) in terms of complexity degree, whereas MMSE surpasses Least Square (LS) in terms of BER and BLER.
Multiple Input Multiple Output-Orthogonal Frequency Division Multiplexing (MIMO-OFDM) is a well-known contemporary wireless broadband technology because of its spectral efficiency, high data transmission rate, and resilience to multipath fading (Manasa & Venugopal, 2023). This method offers a wide range of coverage and reliable communication. Two significant issues for MIMO-OFDM systems are the accurate recovery of Channel State Information (CSI) and the synchronization between the transmitter and receiver. Channel state information is recovered using a variety of estimate techniques, including blind, pilot-aided, and semi-blind channel estimating. However, those systems operate poorly due to a number of shortcomings. Therefore, the basic introduction of the Channel Estimation (CE) process in the MIMO-OFDM system is described in this study. This survey's primary objective is to investigate the analysis and classification of simulation tools and channel estimation techniques in various contributions. It also highlights the performance study using various performance measures from various contributors.

Mathematical Formulation

Least Mean Square (LMS) Algorithm

An adaptive filter, the LMS method modifies its filter coefficients to reduce the mean square error (MSE) between the output signal and the intended signal. The following stages can be used to express the LMS algorithm, which is derived from Wiener filter theory:

Initialization: Initialize the filter weights w(0)w(0)w(0) to zero or small random values.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue I, January 2026

$$w(0) = \begin{bmatrix} w_1(0) \\ w_2(0) \\ \vdots \\ w_N(0) \end{bmatrix} \quad (3.1)$$

Filter Output: Compute the filter output y(n)y(n)y(n) by convolving the input signal x(n)x(n)x(n) with the filter weights w(n)w(n)w(n):

$$y(n) = w^T(n)x(n) = \sum_{i=0}^{N-1} w_i(n)x(n-i) \quad (3.2)$$

Error Signal: Determine the difference between the intended signal and the error signal, e(n)e(n)e(n).

d(n)d(n)d(n) and the filter output y(n)y(n)y(n):

$$e(n) = d(n) - y(n) \quad (3.3)$$

Weight Update: Update the filter weights using the LMS update rule:

$$w(n+1) = w(n) + \mu e(n)x(n) \quad (3.4)$$

where μ\muμ is the learning rate, or step size, that regulates the algorithm's stability and rate of convergence. Until the weights converge to their ideal values and the mean square error is minimized, the iterative process is continued.

Kalman Filter

One of the best recursive algorithms for estimating the state of dynamic systems is the Kalman filter. By reducing the mean square error, it offers the most accurate approximation of the state of the system. Prediction and updating are the two primary processes via which the Kalman filter functions.

State Space Model: Define the state space model of the system:

$$x_k = Ax_{k-1} + Bu_k + w_k \quad (3.5)$$

$$z_k = Hx_k + v_k \quad (3.6)$$

where uku_kuk is the control input, wkw_kwk and vkv_kvk are the process and measurement noise, respectively, and xkx_kxk is the state vector at time kkk. It is assumed that the noise is Gaussian with zero mean and covariance matrices QQQ and RRR.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue I, January 2026

Prediction Step: Estimate the error covariance Pk|k−1P_{k|k-1}Pk|k−1 and the next state x^k|k−1\hat{x}_{k|k-1}x^k|k−1:

$$\hat{x}_{k|k-1} = A\hat{x}_{k-1|k-1} + Bu_k \tag{3.7}$$

$$P_{k|k-1} = AP_{k-1|k-1}A^T + Q \tag{3.8}$$

Update Step: Compute the Kalman gain KkK_kKk:

$$K_k = P_{k|k-1}H^T(HP_{k|k-1}H^T + R)^{-1} \tag{3.9}$$

State Estimate and Error Covariance:

$$\hat{x}_{k|k-1} = \hat{x}_{k|k-1} + K_k(z_k - H\hat{x}_{k|k-1}) \tag{3.10}$$

$$P_{k|k-1} = (I - K_kH)P_{k|k-1} \tag{3.11}$$

To provide the best state estimation, the Kalman filter iteratively improves both the state estimate and the error covariance.

$$w(n+1) = w(n) + \mu e(n)x(n) \tag{3.12}$$

Kalman Prediction: Utilizing the Kalman filter equations, forecast the subsequent state and error covariance:

$$\hat{x}_{k|k-1} = A\hat{x}_{k-1|k-1} + Bu_k \tag{3.13}$$

$$P_{k|k-1} = AP_{k-1|k-1}A^T + Q \tag{3.14}$$

Update the state estimate and error covariance with the latest measurement:

$$K_k = P_{k|k-1}H^T(HP_{k|k-1}H^T + R)^{-1} \tag{3.15}$$

$$\hat{x}_{k|k-1} = \hat{x}_{k|k-1} + K_k(z_k - H\hat{x}_{k|k-1}) \tag{3.16}$$

$$P_{k|k-1} = (I - K_kH)P_{k|k-1} \tag{3.17}$$

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue I, January 2026

By iteratively updating the filter weights with the LMS algorithm and fine-tuning the state estimate with the Kalman filter, the hybrid technique improves signal estimation and noise reduction performance.

System Model Simulation

The project's implementation was carried out using a MATLAB simulation framework. The downlink physical shared channel layer is where the system is simulated. Throughout the implementation process, MATLAB's signal processing and communication toolboxes were utilized. The communication toolbox provides test and measurement tools, such as bit error rate, waveform generator, CRC encoding, and decoding operations, to examine and verify the channel system that has been put into place. The signal processing toolbox made it easier to build the channel estimators by doing a number of tasks, including calculating the covariance, measuring the SNR, and autocorrelation of the matrix. Two statistical techniques for figuring out the relationship between two random variables in two distinct matrices are covariance and autocorrelation.

The methodologies differ in that the autocorrelation measures the strength of the linear relationship between variables, whereas the covariance measures the linear relationship between two variables. As the number of antennas grows, the channel estimation's complexity rises linearly. Consequently, only 4x4—that is, four antenna ports on the transmitter side and four antenna ports on the reception end—is supported by the system. Implementing the system in the frequency domain is simpler than doing so in the time domain. Four transmitters (Tx), a channel, and four receivers (Rx) make up the majority of the system. Different outcomes of channel estimation were examined using a variety of channel model types.

Channel model cases are:

- To add noise that replicates the noise in an actual communication environment, the communication toolbox offers an AWGN channel model function.
- One reception antenna is correlated by the channel for each transmission phase in the MIMO simple channel model. The channel correlation coefficient, which is multiplied by the data being communicated

$$H_{Simple} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

during the transmission, is shown in the following dcequation. (3.18)

$$Y = H_{Simple}X + AWGN \tag{3.19}$$

MIMO sparse channel model, in which one or two receiver antennas are correlated by the channel for each transmission phase. The channel correlation coefficient is shown in the following equation:

$$H_{Sparse} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.20}$$

$$Y = H_{Sparse}X + AWGN \tag{3.21}$$

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue I, January 2026

In addition to their channel correlation, the basic MIMO and sparse MIMO channels also include AWGN.

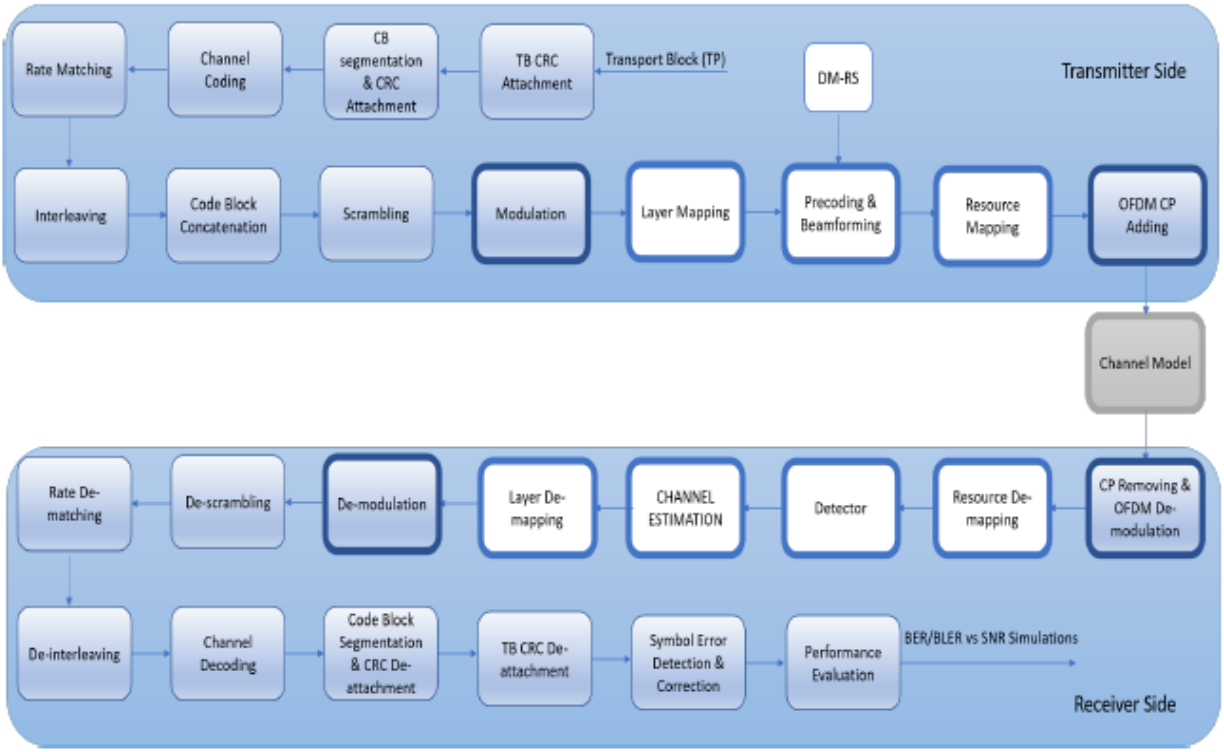

Figure 3.1: Block diagram of System model

In order to assess the hybrid LMS-Kalman filter's performance, we constructed a simulation environment using the following settings:

1. **System Model:**

- Carrier frequency: 28 GHz
- Bandwidth: 100 MHz
- Number of antennas: 64 (8x8 MIMO configuration)
- Channel model: Rayleigh fading

2. **Noise Model:**

- Additive White Gaussian Noise (AWGN)
- Signal-to-Noise Ratio (SNR): Varied from 0 to 30 dB

3. **Performance Metrics:**

- Signal-to-Interference-plus-Noise Ratio (SINR)
- Mean Square Error (MSE)
- Convergence speed

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue I, January 2026

CRC Implementation

The CRC block for error detection was implemented to start the channel coding process. The CRC Consultative Committee for International Telephony (CCIT) generator object is defined as gen by the crc.generator on the transmitter side. The 16-bit object gen defines a number of characteristics and generates the checksum. The transferred data bits are divided by a polynomial in many steps to determine the checksum. The 16-bit checksum (gen) is appended to the end of the sent data by the function generate. A 16-bit CRCCCIT detector object det is constructed on the receiver side by crc.detector. A number of properties are defined by the object det.

To recreate a checksum on the recipient's end, the polynomial division is repeated. By comparing the object det regenerated checksum with the object gen checksum attached to the transmitted data, the function detect finds the problem in the received data. The function gives the number of subframe (block) faults as well as the output data with the checksum removed. The outdata and indata bits are compared by the function isequal. Data was not impacted or altered during transmission if the outcome is equal, and vice versa. The rate of the number bit errors is returned by the TotalBLER function.

Channel Estimation Implementation

The transmitted pilot DMRS symbols (reference signal) provided by dmrsSym are utilized in the channel estimation process to estimate the channel coefficient specified by H, whereas dmrsDe specifies the DMRS symbols built at the receiver side. To recover the signal that was initially delivered, the receiver decodes the received signal using channel estimation. The LS and LMMSE algorithms are the channel estimation methods that are currently in use in the system; LS is used in four transmission phases. It is generally recognized that the pilot symbols channel estimations operate well. One disadvantage, though, is that delivering the pilot and data symbols simultaneously increases the transmission overhead.

The channel estimation algorithms MMSE and LRALMMSE are selected for implementation because they estimate the channel coefficient using training symbols.

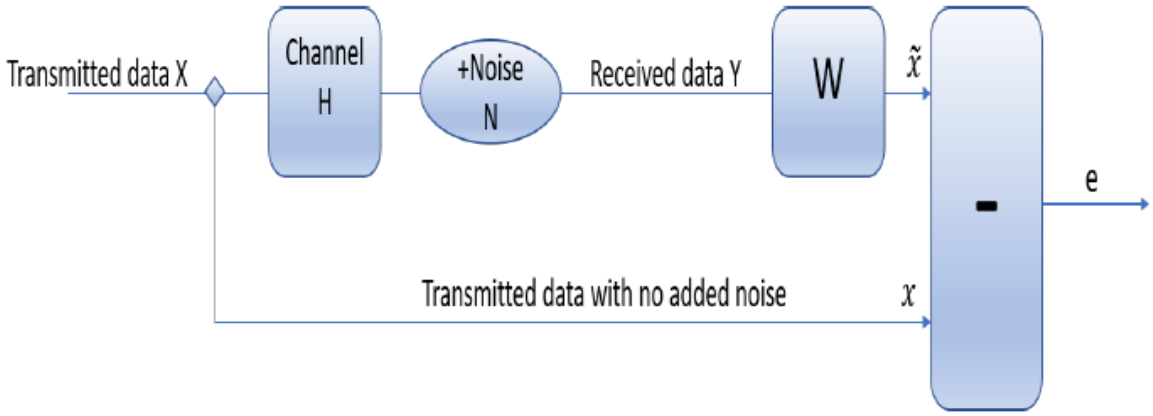

Figure 3.2: MMSE channel estimation

Algorithm Implementation

LMS Algorithm

In order to reduce the mean square error (MSE) between the intended and actual output, the LMS algorithm adaptively modifies the filter coefficients using a stochastic gradient-based methodology. The following is the fundamental LMS update equation:

$$w(n+1) = w(n) + \mu e(n) x(n)$$

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue I, January 2026

where:

- w(n)w(n)w(n) is the weight vector at iteration nnn.
- μ\muμ is the step size (learning rate).
- e(n)e(n)e(n) is the error signal, defined as e(n)=d(n)−y(n)e(n) = d(n) - y(n)e(n)=d(n)−y(n).
- d(n)d(n)d(n) is the desired signal.
- y(n)y(n)y(n) is the output signal of the adaptive filter, y(n)=wT(n)x(n)y(n) = w^T(n)x(n)y(n)=wT(n)x(n).
- x(n)x(n)x(n) is the input signal vector.

The step size μ\muμ, which is crucial for stability and quick convergence, determines how well the LMS algorithm converges. The following procedures are used to implement the LMS algorithm:

Initialization:

$$w(0) = \begin{bmatrix} w_1(0) \\ w_2(0) \\ \vdots \\ w_N(0) \end{bmatrix} \tag{3.22}$$

Initialize the filter weights to zero or small random values.

Filter Output:

$$y(n) = w^T(n)x(n) \tag{3.23}$$

Convolution of the input signal with the filter weights yields the filter output.

Error Signal:

$$e(n) = d(n) - y(n) \tag{3.24}$$

The difference between the intended signal and the filter output is the error signal.

$$w(n+1) = w(n) + \mu e(n)x(n) \tag{3.25}$$

Kalman Filter

The following procedures are used to implement the Kalman filter:

State Space Model: Describe the system's state space model:

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue I, January 2026

$$x_k = Ax_{k-1} + Bu_k + w_k \quad (3.26)$$

$$z_k = Hx_k + v_k \quad (3.27)$$

Prediction Step:

$$\hat{x}_{k|k-1} = A\hat{x}_{k-1|k-1} + Bu_k \quad (3.28)$$

$$P_{k|k-1} = AP_{k-1|k-1}A^T + Q \quad (3.29)$$

$$K_k = P_{k|k-1}H^T(HP_{k|k-1}H^T + R)^{-1} \quad (3.30)$$

$$\hat{x}_{k|k-1} = \hat{x}_{k|k-1} + K_k(z_k - H\hat{x}_{k|k-1}) \quad (3.31)$$

$$P_{k|k-1} = (I - K_kH)P_{k|k-1} \quad (3.32)$$

Hybrid LMS-Kalman Filter Implementation

The LMS and Kalman filter processes are integrated to create the hybrid LMS-Kalman filter. The procedure entails:

- Set the LMS filter weights and Kalman filter parameters to their initial values.
- Utilizing the LMS method, update the filter weights.
- Use the Kalman filter to forecast the next state and error covariance.
- Apply the most recent measurement to the state estimate and error covariance.

Performance Evaluation

The following measures are used to assess the hybrid LMS-Kalman filter's performance:

Signal-to-Interference-plus-Noise Ratio (SINR):

$$\text{SINR} = \frac{P_s}{P_i + P_n} \quad (3.33)$$

Where PsP_sPs, PiP_iPi, and PnP_nPn are the signal, interference, and noise powers, respectively.

Mean Square Error (MSE):

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue I, January 2026

$$\text{MSE} = \frac{1}{N}\sum_{n=0}^{N-1}(d(n) - y(n))^2 \quad (3.34)$$

where y(n)y(n)y(n) is the filter output, d(n)d(n)d(n) is the desired signal, and NNN is the number of samples.

Convergence Speed: How much iteration is necessary to get the algorithm to a stable state? Setting up the state function is the first step in using the Kalman filter methodology.

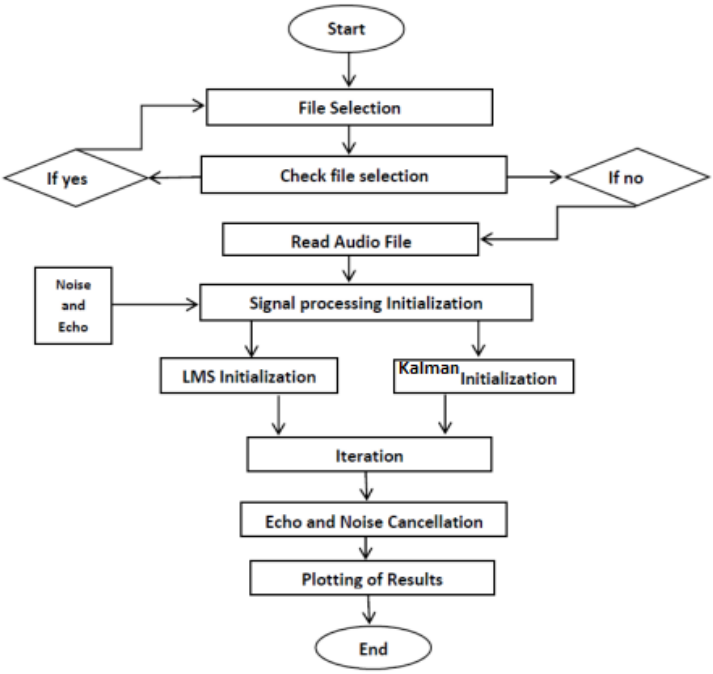

Figure 3.3: Simulation Flowchart

Result Analysis

The details of the observations for each set of mu values are described below:

Mu = 0.01 – 0.05

So, that more noise will not be generated

For K= 2:N

K= 3:N

K= 4:N

K= 5:N

Where N = number of time steps

In Kalman gain with updated Noise Variance;

Where Legends – Time State

- Estimated State

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue I, January 2026

In measurement of Noise Variance Estimation;

Where Legends – True Noise Variance

- Estimated Noise Variance

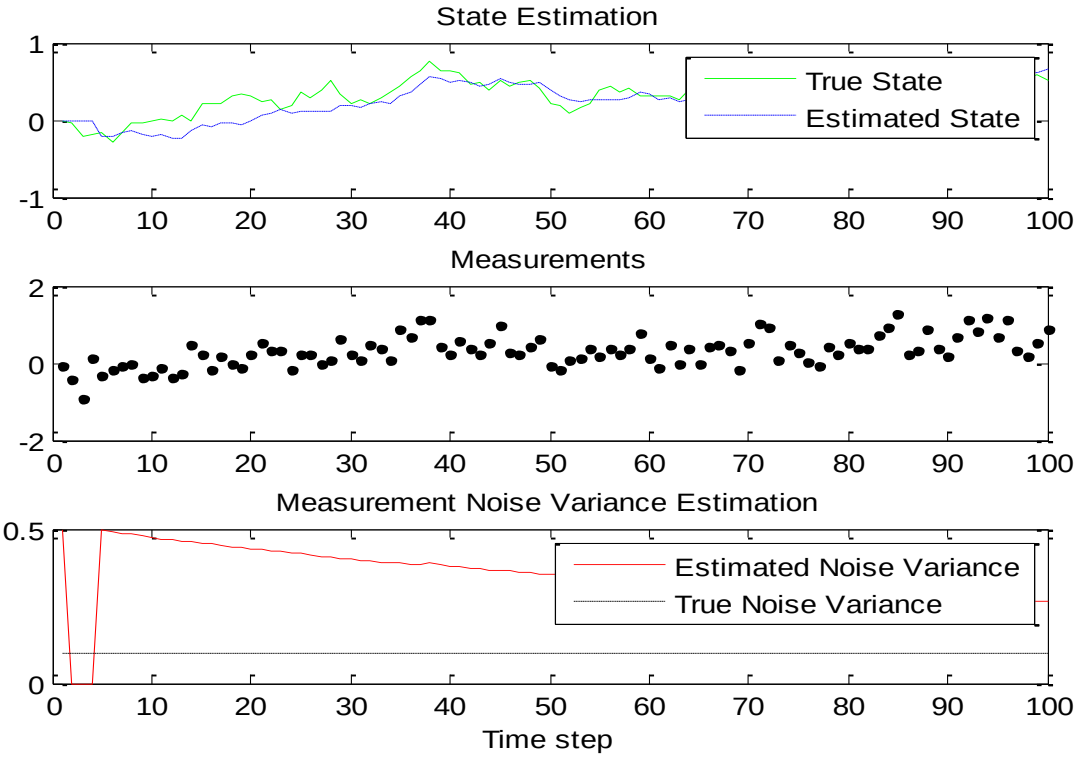

Figure 4.1 showing when the value of K is 2 with step size of noise variance at 0.01

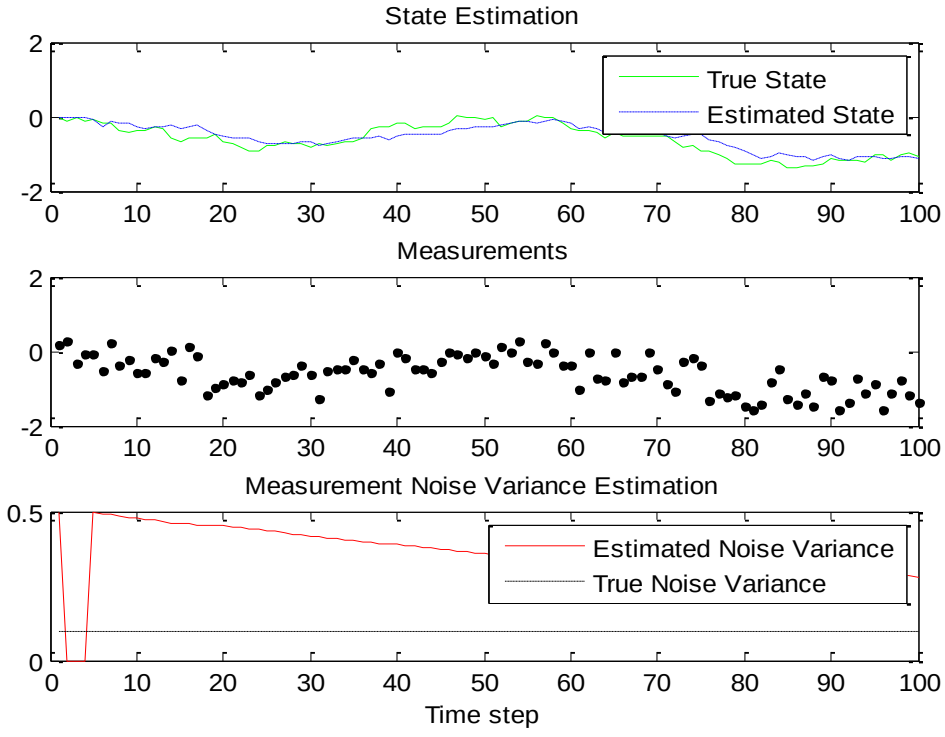

Figure 4.2 showing when the value of K is 3 with step size of noise variance at 0.01

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue I, January 2026

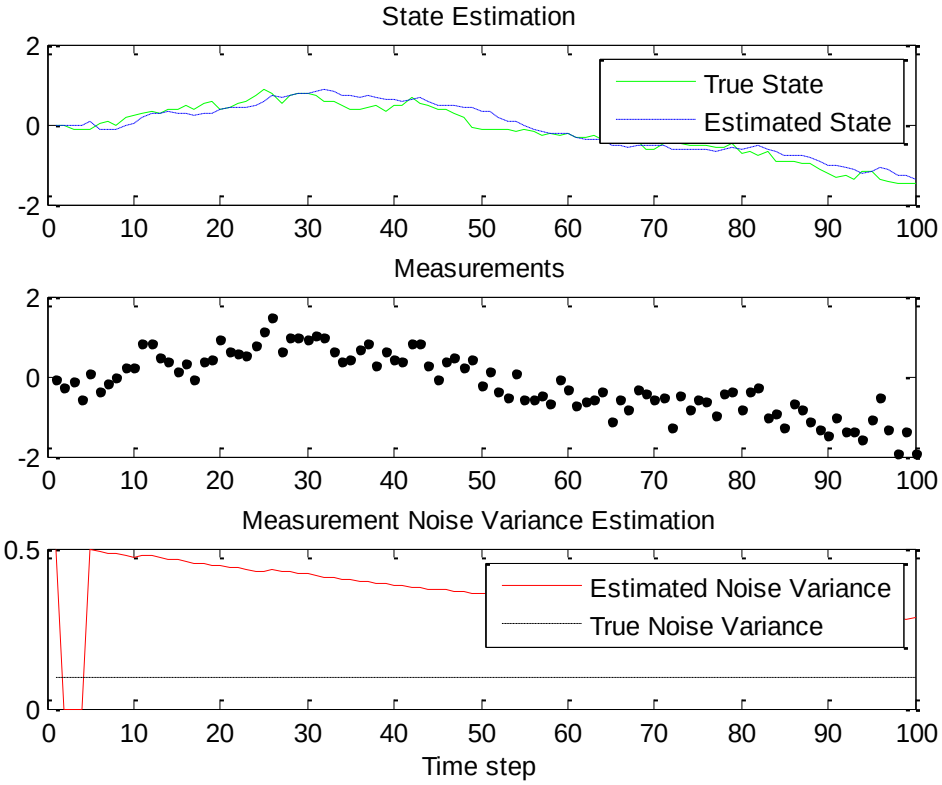

Figure 4.3 showing when the value of K is 4 with step size of noise variance at 0.01

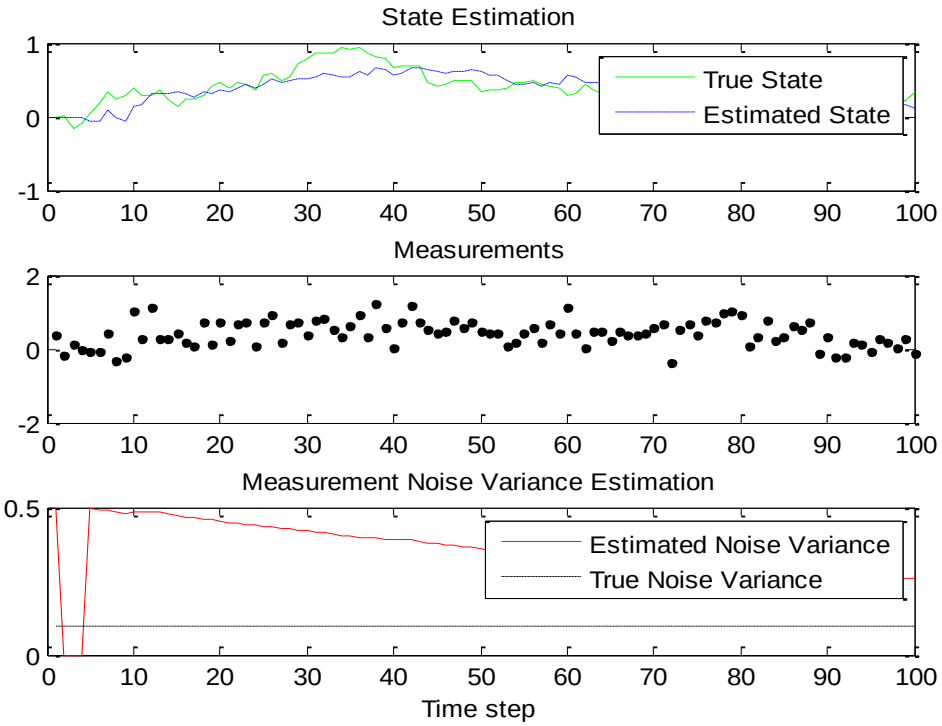

Figure 4.4 showing when the value of K is 5 with step size of noise variance at 0.01

From figure 4.1 to figure 4.4 -: Based on the innovation squared error, the LMS algorithm modifies the measurement noise variance estimate \(r_k\). This hybrid strategy enhances state estimation and noise variance tracking in non-stationary noise settings, as the Kalman filter updates the state estimate using this adaptive noise variance

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue I, January 2026

Kalman Filter Approach for Active Noise Control

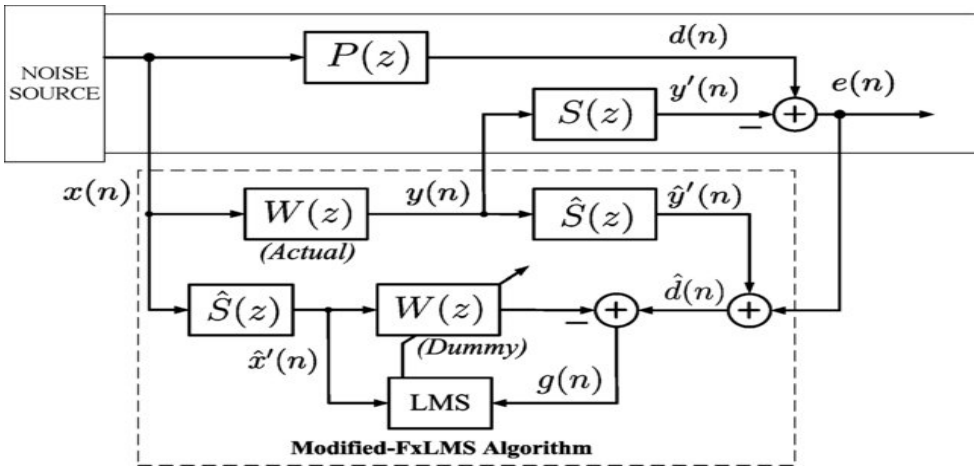

Figure 4.5: Block diagram of the modified ANC structure based on the FxLMS algorithm

Figure 4.5 shows the improved FxLMS algorithm, another well-known method for adaptive active noise reduction. By recovering the disturbance from the error signal using the secondary path estimate, the internal model technique makes it possible to control noise using the conventional least mean square (LMS) algorithm. This modified setup can also be used to apply the Kalman filter methodology, which involves replacing the LMS model with the Kalman filter to finish updating the control filter. Setting up the state function is the first step in using the Kalman filter methodology. Using Appendis 2, the next paragraphs will clarify the ANC's state function definition.

Code configuration of loading the primary and secondary path

The KF.mat file, which applies the Kalman filter method for a single-channel active noise control (ANC) application, is briefly introduced in this section. Also, a comparative examination of the FxLMS algorithm is carried out. While the FxLMS method uses the traditional feed-forward ANC structure, the Kalman filter technique uses the modified feed-forward active noise control (ANC) structure.

The primary path and secondary path are loaded in this section of the code from the Mat files, PriPath_3200.mat and SecPath_200_6000.mat. All these paths are synthesized from the band-pass filters, whose impulse responses are illustrated in Figure 4.

```
close all ; clear   ;

clc         ;
```

```
load ('PriPath_3200.mat');

load('SecPath_200_6000.mat') ;figure ;

subplot(2,1,1) plot( PriPath );

title('Primary Path'); grid on  ;

subplot(2,1,2); plot( SecPath );

title('Secondary Path'); xlabel('
Taps');

grid on       ;
```

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue I, January 2026

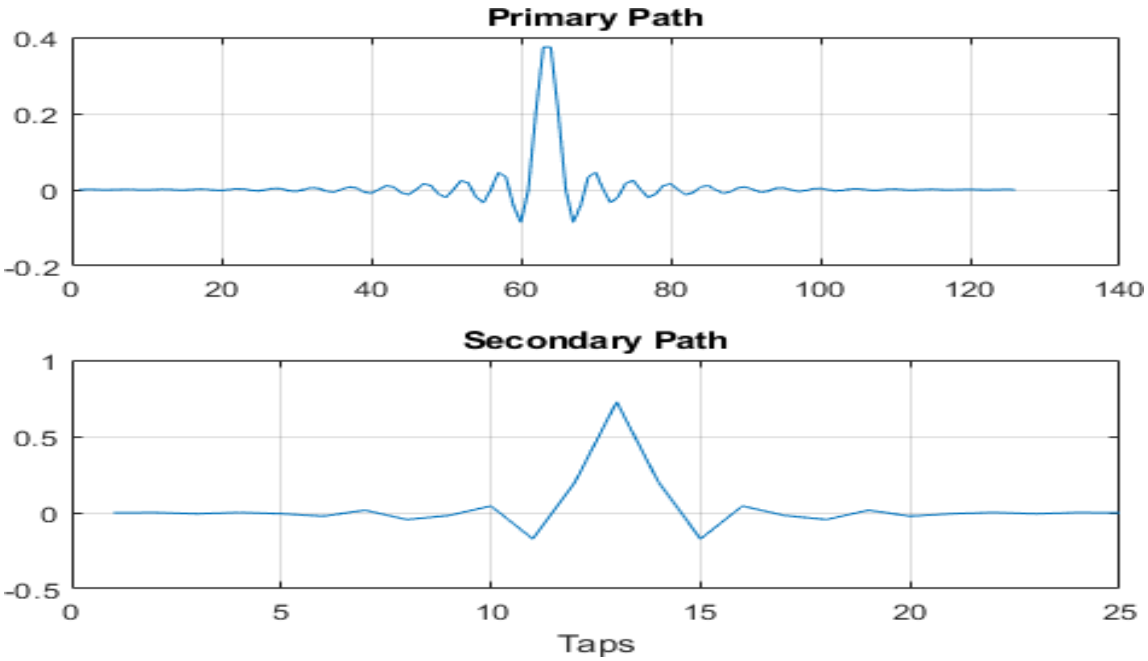

Figure 4:6a The impulse response of the primary path and the secondary path

Simulation system configuration

The active noise control (ANC) system's sampling rate is set to 16,000 Hz, and the simulation lasts for 0.25 seconds. As seen in Figure 5, the main noise in this ANC system is a chirp signal, whose frequency progressively fluctuates between 20 Hz and 1600 Hz in order to replicate the dynamic noise.

Parameter	Definition	Parameter	Definition
fs	Sampling rate	T	Simulation duration
y	Primary noise	N	Simulation taps

Table 1: Simulation input parameters value description and definition

```
fs = 16000          ; % sampling rate 16 kHz.
T  = 0.25           ; % Simulation duration (seconds).
t  = 0:1/fs:T       ;% Time variable.
N  = length(t) ;
fw = 500            ;
fe = 300            ;
y = chirp(t,20,T,1600);
figure    ;
plot(t,y);
title('Reference signal x(n)');
xlabel('Time (seconds)') ;
ylabel('Magnitude')    ;
axis([-inf inf -1.05 1.05]); grid on
;
```

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue I, January 2026

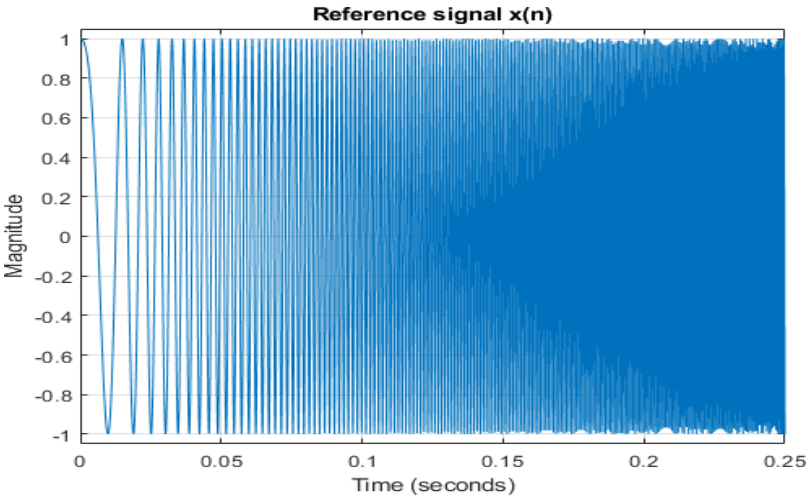

Figure 4: 6b The waveform of the reference signal that is a chirp signal ranging from

```
load ('PriPath_3200.mat');

load('SecPath_200_6000.mat') ;
figure ;

subplot(2,1,1)
plot(PriPath);

title('Primary Path');
grid on       ;
```

20 o 1600 Hz.

Creating the disturbance and filtered reference

The chirp signal is passed through the loaded primary and secondary routes to provide the disturbance and filtered reference utilized in the ANC system.

Parameter	**Definition**	**Parameter**	**Definition**
X	Reference signal vector	Y	Primary noise
D	Disturbance vector	PriPath	Primary path vector
Rf	Filtered reference vector	SecPath	Secondary path vector

Table 2: Simulation input parameters value description and definition

```
%X   =  0.4*sin(2*pi*fw*t)+0.3*sin(2*pi*fe*t);

X = y;

%plot(X(end-100:end))

D  = filter(PriPath,1,X); Rf =
filter(SecPath,1,X);

%plot(D(end-100:end))
```

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue I, January 2026

Dynamic noise cancellation by the single-channel FxLMS algorithm

Here, the chirp disturbance is minimized using the single-channel FxLMS method. In the FxLMS method, the control filter's length consists of 80 taps, with a step size of 0.0005. The error signal detected by the ANC system's error sensor is seen in Figure 4.6. This figure demonstrates that the dynamic noise during the 0.25 second cannot be completely attenuated by the FxLMS algorithm.

Parameter	Definition	Parameter	Definition
X	Reference signal vector	Y	Control signal
D	Disturbance vector	E	Error signal
L	Length of the control filter	muW	Step size

Table 3: Simulation input parameters value description and definition

```
L    = 80      ;
muW = 0.0005;

noiseController = dsp.FilteredXLMSFilter('Length',L,'StepSize',muW, ...
    'SecondaryPathCoefficients',SecPath);

[y,e] = noiseController(X,D);
figure;

plot(t,e) ;

title ('Fx LMS algorithm') ;
ylabel(' Error signal e( n)');
xlabel(' Time (seconds)') ;
grid on ;
```

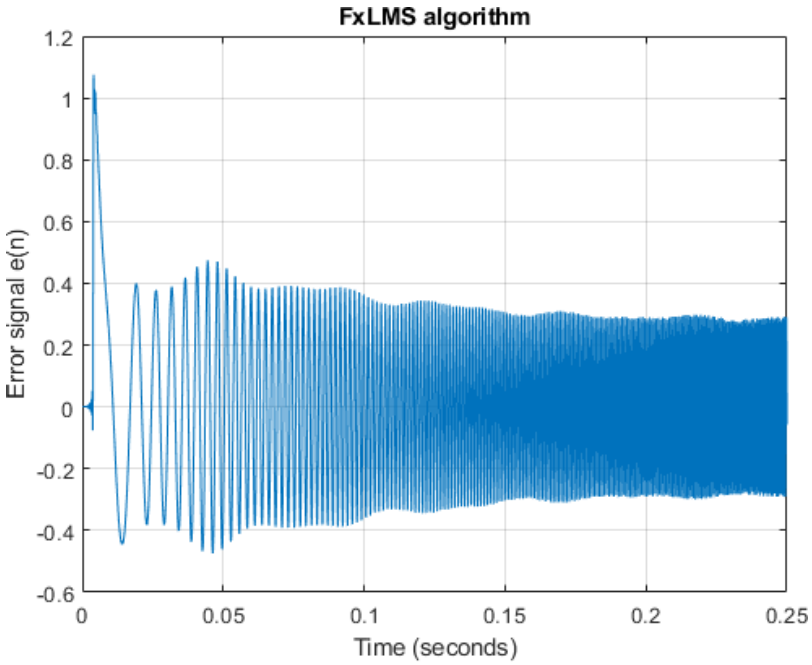

Figure 4.6c the error signal of the single-channel ANC system based on the FxLMS algorithm.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue I, January 2026

Dynamic noise cancellation by the Kalman filter approach

The Kalman filter is employed in the signal-channel ANC system to track the fluctuation of the chirp disturbance. The variance of the observed noise is initially set to 0.005, and the auto- correlation matrix of the state error is initially set to **I**, respectively. Figure 4. 7 The chirp disturbance was reduced in this section using the single-channel FxLMS technique. The step size is set to 0.0005, and the length control filter of the FxLMS algorithm has 80 taps. The error signal of the Kalman filter algorithm is displayed in Figure 4.6. Furthermore, Figure 4.8 shows how the coefficients w5(n) and w60(n) change over time. The result demonstrates how well the Kalman filter reduces the chirp disruptions. As shown in Figure 4.9, the convergence behavior of the Kalman filter approach is noticeably better than that of the FxLMS method. Appendix 1 illustrates the fluctuation caused by the ANC system's error sensor. This figure demonstrates that the dynamic noise during the 0.25 second cannot be completely attenuated by the FxLMS algorithm.

Parameter	**Definition**	**Parameter**	**Definition**
q	Variance of observe error	P	Cross-correlation matrix of state error
W	Control filter	Ek	Error signal
Xd	Input vector	Yt	Anti-noise
Rf	Reference signal vector		

Table 4: Simulation input parameters value description and definition

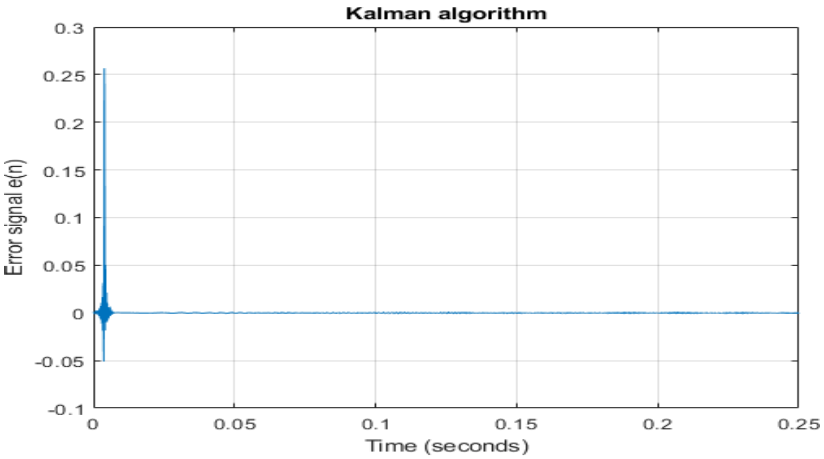

Figure .4 .7: The error signal of the single-channel ANC system based on the Kalman filter.

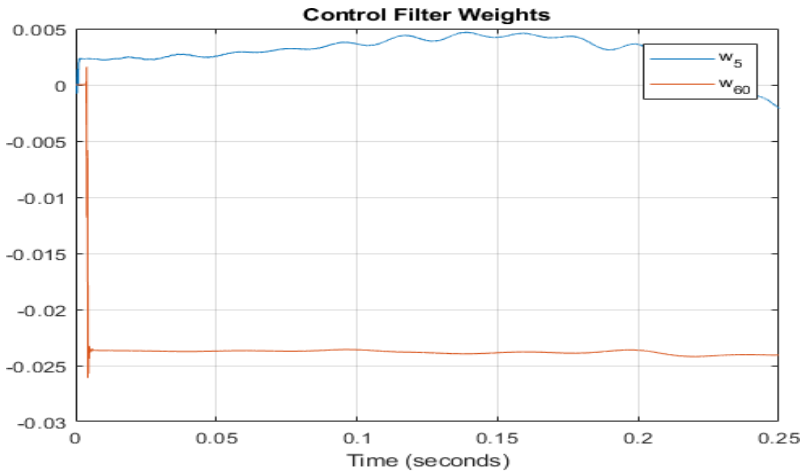

Figure 4. 8: The time history of the coefficients $w_5(n)$ and $w_{60}(n)$ in the control filter.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue I, January 2026

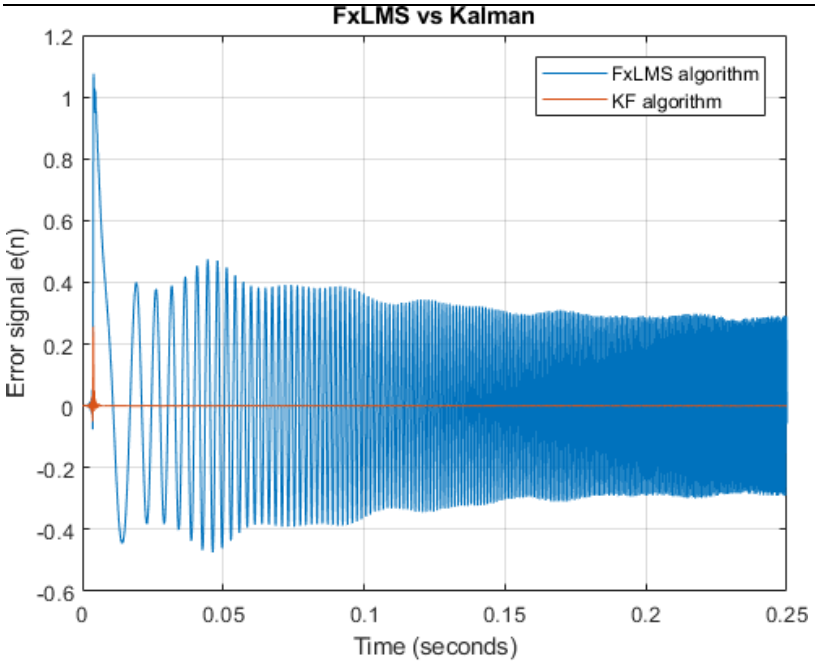

Figure4. 9: Comparison of the error signals in the FxLMS algorithm and the Kalman filter.

CONCLUSION

The conclusion was based on the outcomes of the simulation. This thesis offers a thorough examination of the active control system's Kalman filter code. The adaptive filter, such as the filtered reference least mean square (FxLMS) algorithm, is usually adjusted by traditional active noise management to accommodate variations in the primary noise and 5G environment. However, the noise reduction for dynamic noise is typically impacted by the FxLMS algorithm's slow convergence behavior. In order to enhance the noise reduction performance for dynamic noise, this work x-rayed employing the Kalman filter in the ANC system. The Kalman filter performs exceptionally well in the ANC application with a new dynamic ANC model. When it comes to handling dynamic noise, the numerical simulation showed that the suggested Kalman filter performs significantly better in terms of convergence than the FxLMS algorithm.

It should be noted that the approach still accounts for the observed error even though it assumes a variance of 0 for the state error. This suggests that the transition of the control filter has a higher degree of confidence than the observation function.

It should be noted that the approach still accounts for the observed error even though it assumes a variance of 0 for the state error. This suggests that the transition of the control filter has a higher degree of confidence than the observation function.

REFERENCES

1. Afroz, G. Lukin, S.K. Abramov, K.O. Egiazarian, and J.T. Astola, (2013) Blind evaluation of additive noise variance in textured images by nonlinear processing of block DCT coefficients, in Proceedings of the International Society for Optics and Photonics. Electronic Imaging, Image Processing: Algorithms and Systems II, vol. 5014, 2003, pp. 178–189. https://doi.org/10.1117/12.477717.
2. Akbarpour-Kasgari,N and Ardebilipour,J(2018). Nonparametric noise estimation method for raw images, Journal of the Optical Society of America A, 31 (2014). https://doi.org/10.1364/ JOSAA.31.000863.
3. Alam,T S. Thummaluru,R and Chaudhary,R(2019) “Integration of MIMO and cognitive radio for sub-6 GHz 5G applications,” IEEE Antennas and Wireless Propagation Letters, vol. 18, no. 10, pp. 2021–2025.
4. Analysis and Extension of the Ponomarenko et al. Method, Estimating a Noise Curve from a Single Image, Image Processing on Line, 3 (2013), pp. 173–197. https://doi.org/10.5201/ ipol.2013.45.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue I, January 2026

5. Anscombe,F(2015) The transformation of Poisson, binomial and negative-binomial data,
6. Araújo,T and Almeida,R(2017) Estimation of image noise variance, in Vision,
7. Barsanti,P and Gilmore k,(2016) "A wide-band monopolar wire-patch antenna for indoor base station applications," IEEE Antennas and Wireless Propagation Letters, vol. 4, no. 1, pp. 155–157.
8. Bayat-Makou,N Wu, K and Kishk,A (2019)"Single-Layer Substrate- Integrated Broadside Leaky Long-Slot Array Antennas with Embedded Reflectors for 5G Systems," IEEE Transactions on Antennas and Propagation, vol. 67, no. 12, pp. 7331–7339.