iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** I **Month of publication:** January 2026

DOI: https://doi.org/10.22214/ijraset.2026.76784

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue I Jan 2026- Available at www.ijraset.com

Comparison between LC-LMS and LC-RLS Algorithms for Improving Adaptive Beamforming

Gaddam Aishwarya
Post Graduate Student, MTech (Systems and Signal Processing), JNTUH University College of Science & Technology, Hyderabad

Abstract: Adaptive filtering is a crucial technology that enables interference suppression and enhances signal quality. This research presents a sophisticated adaptive beamforming method that minimises the Mean Squared Error (MSE) and significantly increases convergence rates by employing the Recursive Least Squares (RLS) algorithm. Utilising the RLS algorithm, which is renowned for its quick convergence and resilience to changing signal conditions, the antenna array's weights are dynamically modified. The suggested approach outperforms more established algorithms like Least Mean Squares (LMS) in terms of quicker convergence and lower MSE, as demonstrated by comprehensive simulations and real-world applications. The findings demonstrate the significant efficacy of the RLS-based adaptive beamforming technology in real-time applications, providing improved signal clarity and reliability in challenging and noisy environments. The potential of RLS to develop adaptive beamforming technologies and open the door to more dependable and effective systems for communication is highlighted in this work.
Keywords: Adaptive Filtering, Beamforming, LMS Algorithm, RLS Algorithm, MSE.

I. INTRODUCTION

An adaptive filter is a type of digital system capable of adjusting its coefficients dynamically to minimise an error function—commonly defined as the mean square difference between the desired and actual output [1] [2]. Unlike fixed filters, adaptive filters continually refine their parameters to suit variations in the input environment. The difference between fixed filters and adaptive filters is that adaptive filters can learn and adapt to changing signal or noise characteristics in real time [2]. This makes adaptive filters extremely useful in contexts where the signal statistics are unknown in advance and vary over time.

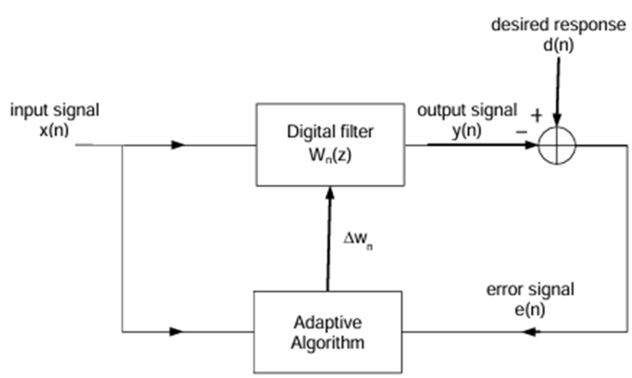


Fig.1 Block Diagram of Adaptive Filter

The adaptive filtering operation begins by calculating the filter's output, y(n) [1] [2], using the current input signal, x(n). This output is then compared to the desired output, d(n), which generates an error signal, e(n). This error signal e(n) is crucial. It is fed back into the adaptive algorithm, which uses it to update the filter's coefficients (or weights). The entire purpose of this update is to systematically reduce the error according to a predefined optimisation goal (like minimising the MSE) [2]. An adaptive filter technique requires a coefficient update equation to compute new filter parameters for each sample. The equation for updating the coefficient is structured as follows:

$$w_{n+1} = w_n + \Delta w_n$$

In Fig. 1, it is integral to the adaptive method, whereby Δw_n signifies the change applied to the filter's current coefficients [2]. This correction step results in a new, updated set of coefficients that will be used at the very next time sample, n+1.

The error signal e(n) is essential for an adaptive filter to function [2]. Without e(n), the filter cannot determine the necessary adjustments for its coefficients, making it incapable of adapting to changes in the operating environment. However, in certain real-world scenarios, obtaining this crucial error signal can be difficult.

Adaptive beamforming refers to a signal processing strategy employed in array systems, such as antenna or hydrophone arrays, that automatically adjusts the directional response to enhance desired signals while suppressing interference [1]. The fundamental objective is to optimise the receipt or transmission of a desired signal while concurrently reducing interference and noise by establishing "nulls" in their direction. In contrast to traditional (or fixed) beamforming [1], which employs predetermined static weights, adaptive beamforming dynamically computes and modifies the complex weights assigned to each sensor element according to the attributes of the received data. The system functions in a closed-loop manner to enhance a performance parameter, usually the Signal-to-Interference-plus-Noise Ratio (SINR). [4] [5] [6]

II. SYSTEM MODEL

A. LMS Algorithm

The LMS algorithm [2] is a very popular adaptive filtering technique because it is both simple and effective. It operates using the stochastic gradient descent method. Its core purpose is to continuously adjust the filter's parameters to minimise the MSE, which is the average squared difference between the desired signal and the actual output of the filter

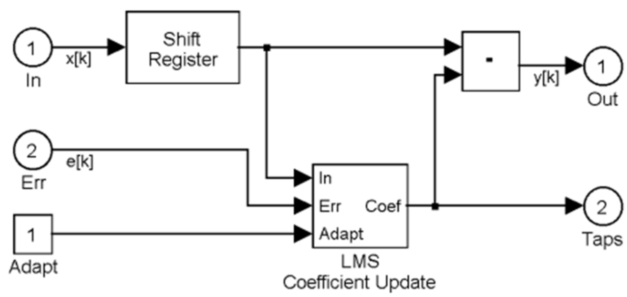


Fig.2 LMS Adaptive Filter Algorithm

Unlike the Wiener filter, which requires prior knowledge of signal statistics, LMS operates by adjusting its filter coefficients iteratively with each new input sample [1] [2]. The weight update rule in LMS is derived by estimating the gradient of the error function using the instantaneous squared error. The weight-vector update equation for the steepest descent adaptive filter is provided by

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

At the nth time, the estimation of the weight is represented as w_n, input signal vector is represented as x(n), the filter error vector is represented as e(n), and μ is the step-size

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

Where λ_{max} is the largest eigenvalue of the autocorrelation matrix Rx.

B. NLMS Algorithm

One major challenge in LMS is selecting an appropriate step size. A large step size may lead to instability, while a small one may slow down convergence. The Normalised LMS (NLMS) algorithm [2] [3] addresses this issue by adjusting the step size dynamically based on the energy of the input signal. The mean square of the autocorrelation matrix is where the LMS method converges for wide-sense stationary processes.[2]—[6]

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

C. RLS Algorithm

The RLS method iteratively determines the filter coefficients that minimise the least squares cost function, in contrast to the LMS algorithm, which seeks to minimise the mean-square error.

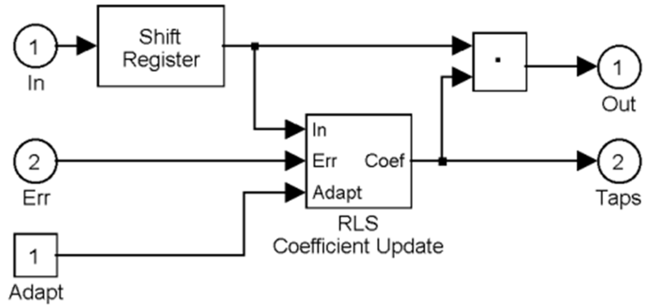


Fig.3 RLS Adaptive Filtering Algorithm

The data of x(n) and d(n) may be used to directly reduce the least squares error in comparison to the mean-square error [2]. The group of filter coefficients w_n(k) at a sample of n that minimises the least squares error, which is weighted using the recursive least squares method.

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

The distinguish between the intended signal d(n) and the estimate of d(n) created by applying the previous set of filter coefficients w_n−1 to the new data vector x(n) is known as the "a priori error," or α(n) [2] [3], which is computed in the equation below.

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

When the equations are finally combined, the exponentially weighted Recursive Least Squares (RLS) procedure is produced.

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

The increasing window RLS technique [3], which considers all prior errors to be equally weighted in the total error, is the exceptional case of λ = 1. Initial conditions are needed for both the vector w_n and the inverse autocorrelation matrix P(n), since the RLS method entails their recurrent update.[5] [4]

D. Linearly Constrained Minimum Variance (LCMV):

The Linearly Constrained Minimum Variance (LCMV) beamformer is a key adaptive filtering technique that generalises the classic Minimum Variance Distortionless Response (MVDR) beamformer by incorporating multiple linear constraints [1] [2]. This method is designed to effectively suppress interference while maintaining specific signal responses in designated directions.

In the context of array signal processing, consider an array consisting of M sensors or antennas receiving incoming signals. The received signal can be modelled as:

$$x(t) = As(t) + n(t)$$

where $\mathrm{x}(t)$is the $M \times 1$received signal vector, A represents the steering matrix with columns corresponding to the array responses of different sources, $\mathrm{s}(t)$is the source signal vector, and $\mathrm{n}(t)$denotes additive noise and interference. [1]

The beamformer output is given by:

$$y(t) = w^H x(t)$$

where w is a complex weight vector used to linearly combine the array elements. The objective of LCMV is to determine wthat minimises the output power $E[| y(t) |^2]$, subject to specific linear constraints. [1] [2]

The optimisation problem is expressed as:

$$min_w w^H R w \quad \text{subject to} \quad C^H w = f$$

where $\mathbf{R} = E[\mathbf{x}(t)\mathbf{x}^H(t)]$is the covariance matrix of the received signal, **C** is the $M \times L$ constraint matrix containing the steering vectors for desired signal directions, and **f** is an $L \times 1$ vector that defines the desired gain pattern (e.g., unit gain in the target direction and nulls in interference directions).

Applying the Lagrange multipliers method, the optimal weight vector is derived as:

$$w_{opt} = R^{-1}C(C^H R^{-1}C)^{-1}f$$

Eq represents the formulation that ensures minimum output power while satisfying the specified linear constraints.

The LCMV beamformer offers flexibility in beam pattern design through the selection of suitable constraints. It can, for example, enforce a unit gain in the desired direction while placing nulls in the directions of known interferers, thus providing strong interference mitigation and robustness.

III. IMPLEMENTATION

Adaptive LCMV Beamforming with LMS and RLS Techniques:

The Linearly Constrained Minimum Variance (LCMV) beamformer is designed to reduce the output power of an array signal while maintaining multiple linear constraints that protect the desired signal components. In real-world scenarios, where noise and signal conditions fluctuate over time, adaptive techniques such as the LMS and RLS are employed to adjust the beamforming weights continually. The LCMV optimisation problem is typically formulated as: [2] [3] [4]

$$min_w w^H Rw \quad \text{subject to} \quad C^H w = f$$

where: In Eq, represents the following:

R: Covariance matrix of the received signal,

C: Matrix of constraints (columns represent steering vectors),

f: Vector representing desired gains.

To implement this adaptively, the LCMV constraints are incorporated into either the LMS or RLS frameworks.

A. LCMV-LMS (LC-LMS) Method

The LCMV-LMS algorithm merges the constraint enforcement of LCMV with the simplicity of LMS. A common technique involves modifying the standard LMS update by projecting it onto the set of weights that satisfy the LCMV constraints.

Let w(n)be the weight vector at time n, and the beamformer output is given by:

$$y(n) = w^H(n)x(n)$$

Eq symbolises the output of the beamformer, and the typical unconstrained LMS update is:

$$w(n+1) = w(n) - \mu y^*(n)x(n)$$

To satisfy the LCMV constraint C^H w(n+1) =f, a projection approach is used:

1. Apply the unconstrained LMS update:

$$\mathrm{w}'(n+1) = \mathrm{w}(n) - \mu\, y^*(n)\mathrm{x}(n)$$

2. Project the updated vector onto the constraint set:

$$\mathrm{w}(n+1) = \mathrm{w}'(n+1) + \mathrm{C}(\mathrm{C}^H\mathrm{C})^{-1}(\mathrm{f} - \mathrm{C}^H\mathrm{w}'(n+1))$$

This ensures that each update adheres to the LCMV constraints while remaining adaptive to environmental changes.

B. LCMV-RLS (LC-RLS) Method

The LCMV-RLS approach integrates the fast convergence properties of RLS with the LCMV constraint framework. RLS updates the weights by minimising a weighted least squares error:

$$J(n) = \sum_{i=1}^{n} \lambda^{n-i} \mid y(i) \mid^2 = \sum_{i=1}^{n} \lambda^{n-i} \mid \mathbf{w}^H(n)\mathbf{x}(i) \mid^2$$

Subject to the constraint:

$$\mathrm{C}^H\mathrm{w}(n) = \mathrm{f}.$$

To implement LCMV-RLS, the inverse covariance matrix estimate P(n)is updated, and the optimal weights are determined using Lagrange multipliers:

$$\mathrm{P}(n) = \lambda^{-1}[\mathrm{P}(n-1) - \frac{\mathrm{P}(n-1)\mathrm{x}(n)\mathrm{x}^H(n)\mathrm{P}(n-1)}{\lambda + \mathrm{x}^H(n)\mathrm{P}(n-1)\mathrm{x}(n)}]$$

The corresponding weight update equation becomes:

$$\mathrm{w}(n) = \mathrm{P}(n)\mathrm{C}(\mathrm{C}^H\mathrm{P}(n)\mathrm{C})^{-1}\mathrm{f}$$

This solution mirrors the batch LCMV formulation but replaces the fixed inverse covariance matrix with its adaptive estimate P(n).

IV. SIMULATION AND RESULTS

The study involves implementing adaptive Least Squares-Constrained Minimum Variance (LCMV) algorithms in MATLAB. The primary objective is to enhance the convergence rate, reduce the error rate, and analyse the beam patterns of these algorithms. This work aims to improve industrial measurement systems within communication networks. To enable better comparison among various adaptive filtering techniques, the process begins by using a simulated environment to generate clear findings. All adaptive filters are tested using identical primary input data, which consists of the delayed input, x(n-n0), and the target signal, d(n).

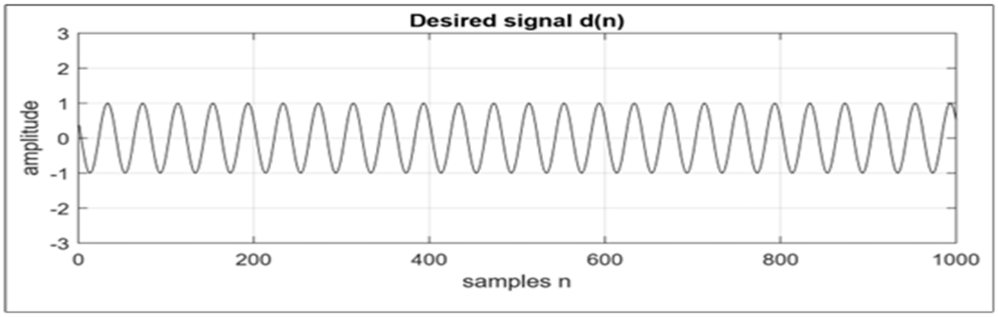


Fig.4 Desired signal d(n) is to be estimated

A. *Noise-corrupted Signal for x(n)*

The noise-corrupted signal for the input signal is given below:

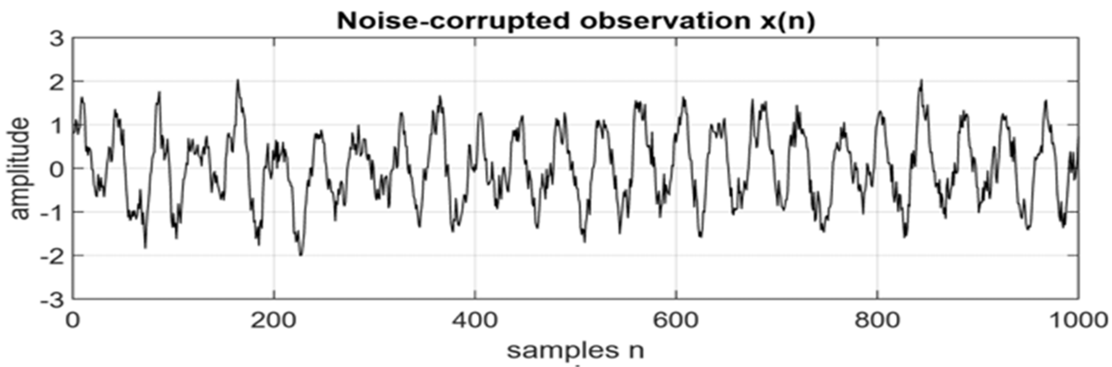


Fig.5 Noise-corrupted observation signal

Along with the desired signal, the system will have the voice corrupted signal along, that corrupted signal is given below:

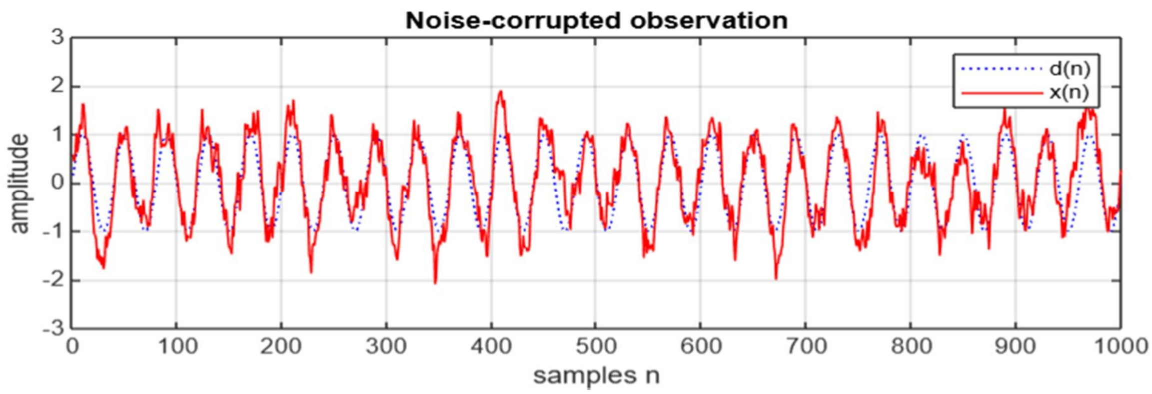


Fig.6 Noise corrupted signal with d(n) & x(n)

The reference signal is generated by introducing a delay of several samples to the input signal, x(n). This effect is visually demonstrated in the subsequent plot.

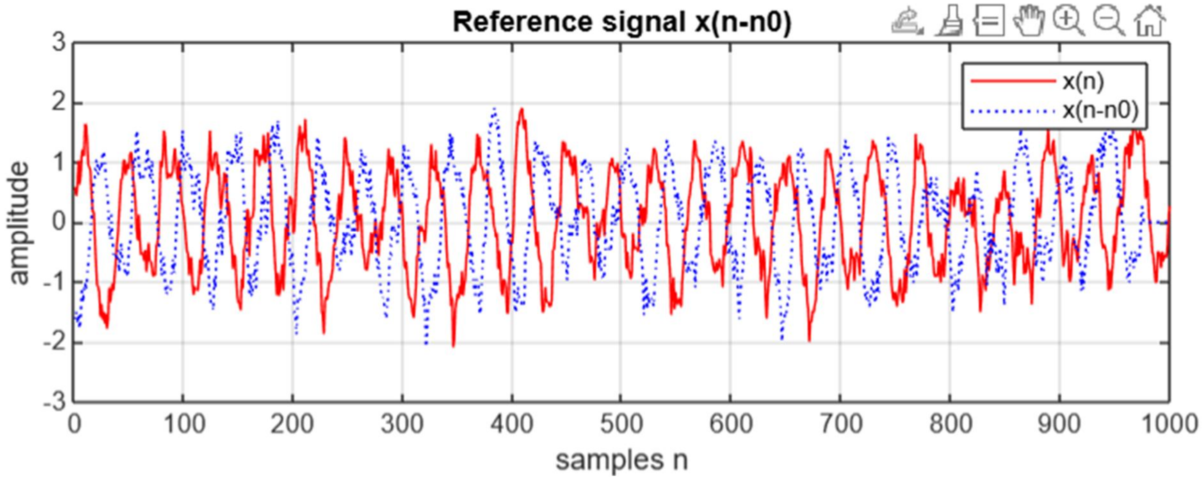


Fig.7 Reference signal

The test series provides additional performance parameters for each method. The most important finding is the specific convergence rate at which all tested algorithms perform identically and perform well in noise cancellation were chosen for comparison

B. Estimate signal for the LMS Algorithm

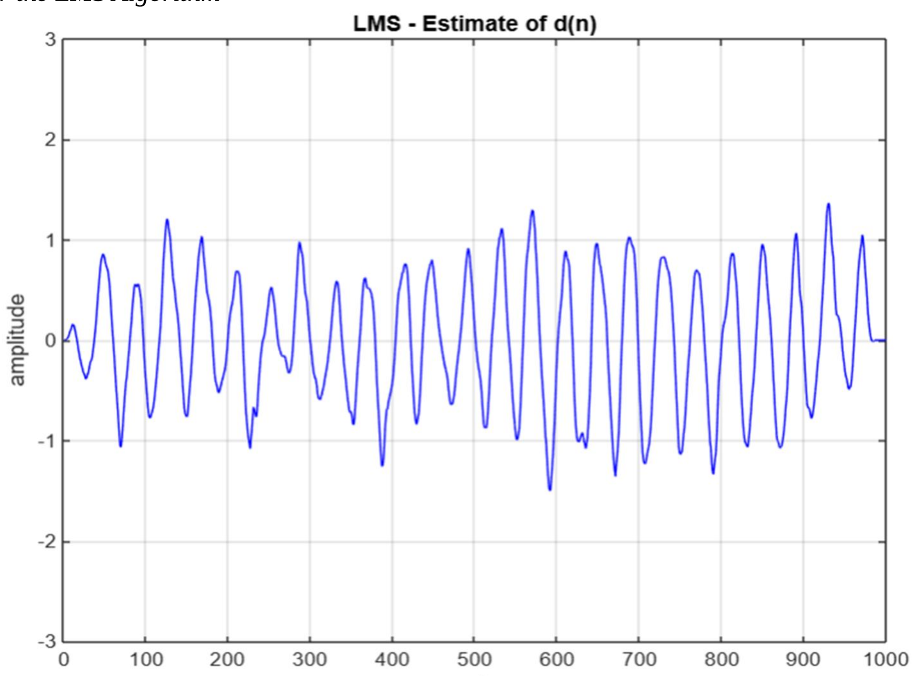


Fig.8 LMS – Estimate of desired signal

Based on the data, the Least Mean Squares (LMS) algorithm shows poor performance in its initial attempts at noise cancellation. The initial estimated signal, d(n), has an amplitude that is too low relative to the desired signal, making it impossible to accurately reproduce the sine wave shape. The adaptive filter only begins providing a more accurate estimate of d(n) after approximately 100 samples.

The plotted results for the LMS algorithm show three key signals: the noisy process x(n), the intended sinusoidal signal, and the estimated desired signal d(n). These outputs were produced by a 12th-order adaptive noise canceller operating with a normalised step-size parameter

C. Estimate signal for the NLMS Algorithm

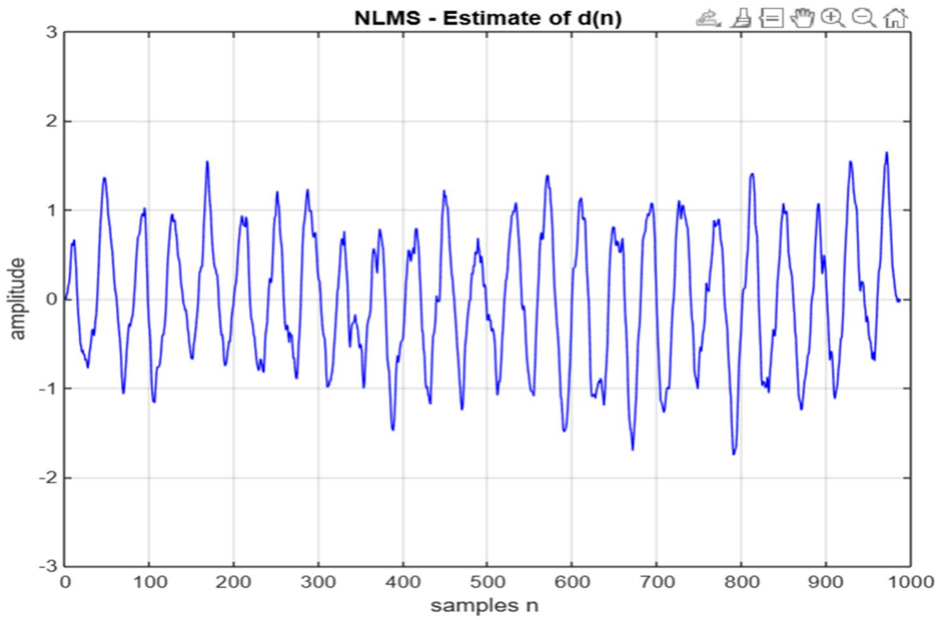


Fig.9 NLMS – Estimate of desired signal

Initially, the signal estimate d(n) has a much higher amplitude than the original signal, and it is impossible to replicate the form of a sinusoidal. After around 50 samples. The adaptive filter gets the estimate of d(n) accurately. When it comes to noise cancellation, the RLS algorithm performs well.

D. Estimate signal for the RLS Algorithm

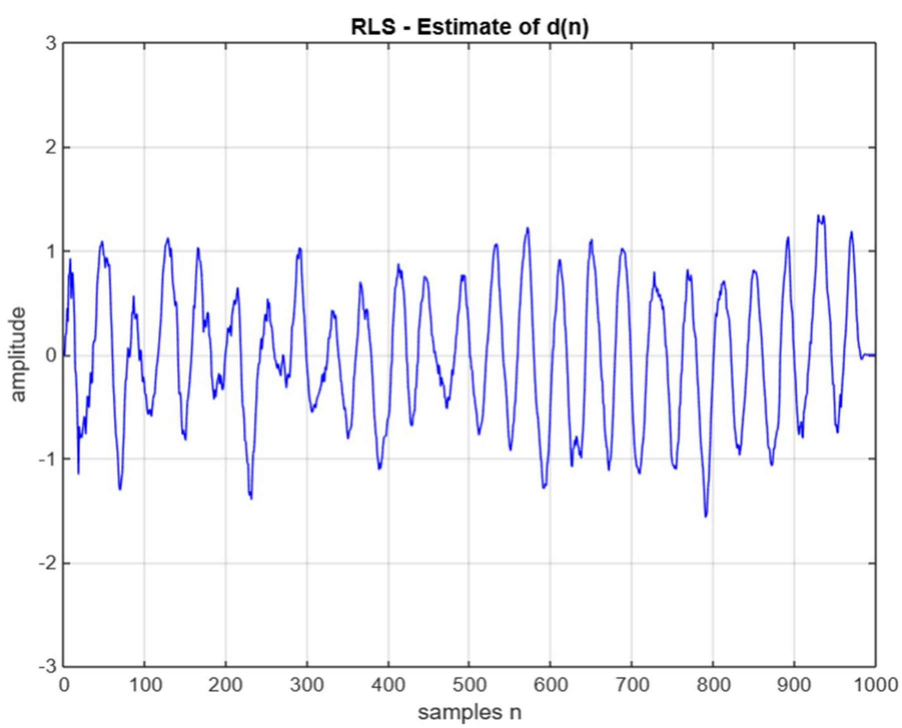


Fig.10 RLS – Estimate of desired signal

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue I Jan 2026- Available at www.ijraset.com

Combining all the outputs of the algorithms is shown below:

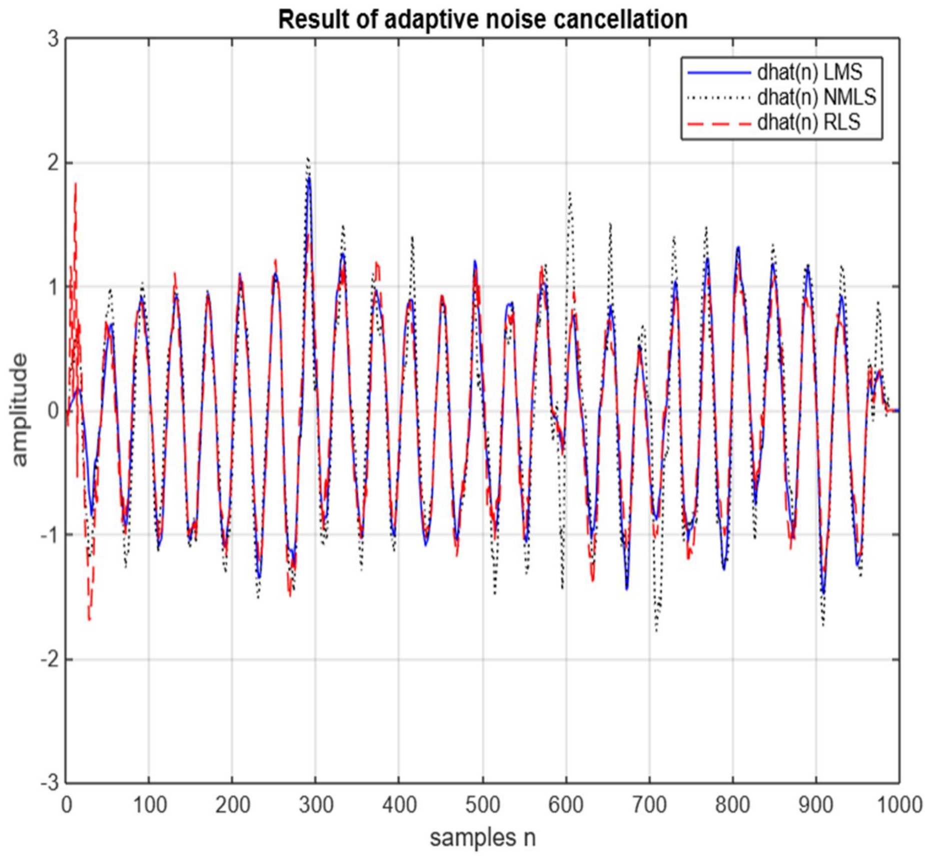


Fig.11 Outputs of all algorithms

E. Comparison of LMS and RLS Algorithms Based on Parameters, Convergence Rate, and Error Performance

The parameters, convergence rates, and error performance of both algorithms are compared in the table below. Across all measured values, RLS exhibits much quicker convergence and smaller error than LMS. LMS produces an increase in accuracy and requires more repeats, particularly when step sizes are smaller. While LMS provides simplicity for low-complexity systems, RLS is often more efficient.

TABLE: Comparison Table of Algorithms

Algorithms	Parameters	Rate of Convergence	Error Performance
LMS λ – constant	μ = 0.01	25 Iterations	0.285
	μ = 0.002	100 Iterations	0.307
	μ = 0.02	50 Iterations	0.295
RLS μ - constant	λ = 0.9	25 Iterations	0.069
	λ = 1	30 Iterations	0.078
	λ = 0.8	10 Iterations	0.048

F. Convergence Rate for the LC- LMS and LC-RLS Algorithms

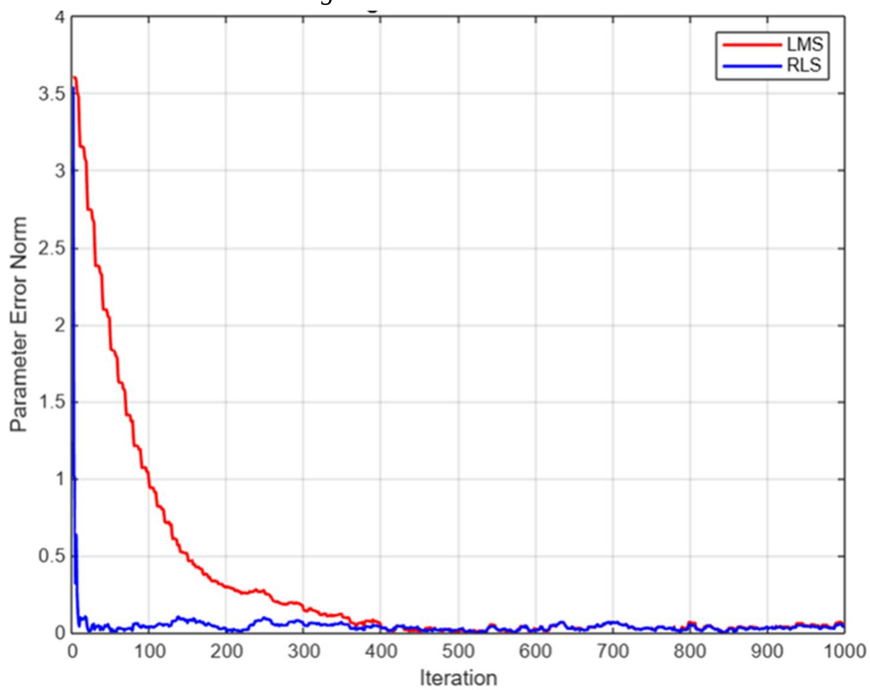


Fig.12 Graph of Convergence rate (LMS & RLS)

The graph plots the parameter error norm versus the number of iterations to show how both algorithms converge. It is clear from the figure that the RLS method, which is represented by the blue curve, converges far more quickly than the red LMS algorithm. Within the first 20 repetitions, RLS attains a low error norm, demonstrating its quick adaptability and suitability for dynamic situations. The slower convergence of the LMS method is shown by the fact that it takes more than 300 iterations to achieve a similar error level. Nevertheless, upon convergence, both algorithms ultimately settle and sustain low error norms. But although LMS displays higher fluctuation in the steady-state area, RLS performs better in the steady-state with less volatility.

Overall, the graph demonstrates that LMS is still a good choice for systems with limited computing resources because of its simplicity and reduced complexity, while RLS provides greater convergence speed and accuracy, making it perfect for real-time and rapidly changing signal environments.

G. Mean Squared Error for LMS and RLS Algorithms

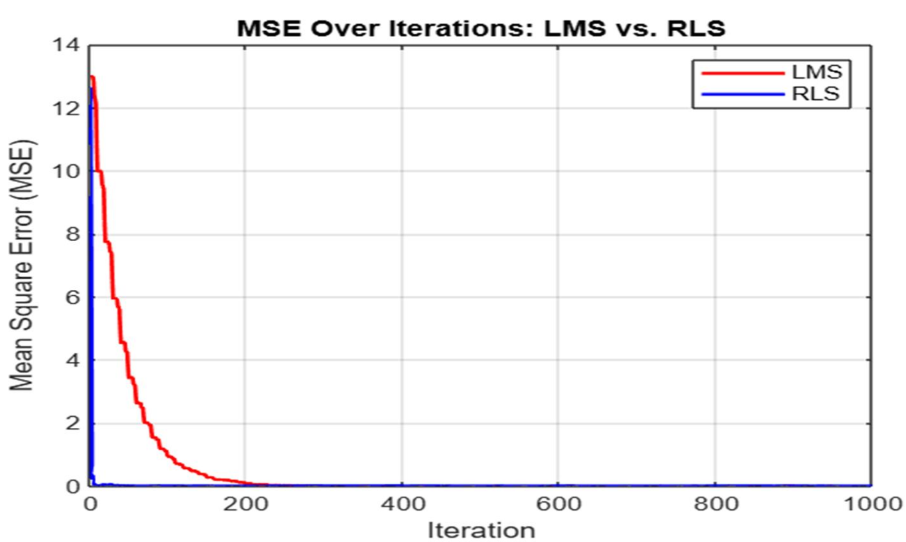


Fig.13 Graph of MSE (LMS & RLS)

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue I Jan 2026- Available at www.ijraset.com

The Mean Squared Error (MSE) for both algorithms is compared over iterations in the graph. The RLS approach (blue curve), as shown, reduces the MSE far more quickly and reaches a low error level in the first few repetitions. This speedy convergence demonstrates how the RLS method can rapidly adjust and reduce error, which makes it a good fit for contexts that change over time or are dynamic. The LMS method (red curve), on the other hand, shows a slower reduction in MSE, requiring around 200 iterations to reach a comparable steady-state error level. The LMS algorithm finally reaches low MSE values that are equivalent to RLS, although with a longer adaptation period, despite its slower convergence.
This graph unequivocally demonstrates the RLS algorithm's higher convergence speed and learning efficiency while simultaneously demonstrating that LMS, despite its slower speed, may still be useful in scenarios where computational simplicity is more important than quick convergence.

H. Beam Pattern Comparison of Adaptive Algorithms: LMS, NLMS, and RLS

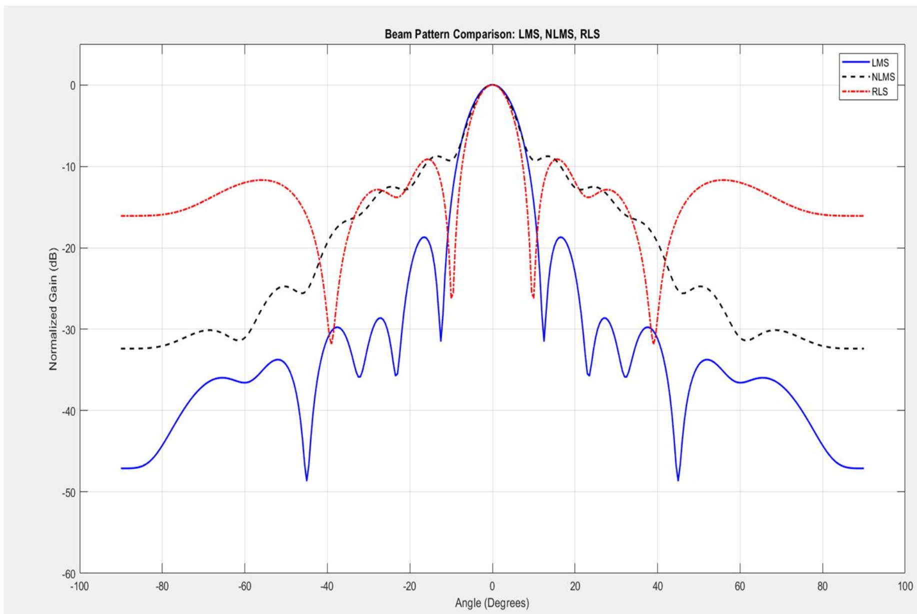


Fig.14 Beam pattern

1) LMS Algorithm Performance (μ-step size constant):
- μ = 0.01 shows a moderate convergence rate (25 iterations) with an error performance of 0.285.
- μ = 0.002 requires the highest number of iterations (100), indicating the slowest convergence, and yields the highest error (0.307).
- μ = 0.02 converges faster than μ = 0.002 but has a slightly better error performance (0.295).

2) RLS Algorithm Performance (λ - forgetting factor constant):
- λ = 0.8 gives the fastest convergence (10 iterations) and the lowest error (0.048).
- λ = 0.9 also achieves fast convergence (25 iterations) with low error (0.069).
- λ = 1 results in slightly slower convergence (30 iterations) and moderate error (0.078).

3) RLS consistently outperforms LMS in both convergence rate and error performance.
4) Higher values of λ in RLS lead to slightly slower convergence but still maintain low error.
5) Lower step sizes (μ) in LMS result in slower convergence and higher error.
6) RLS is more efficient for real-time or dynamic systems, while LMS is simpler and better suited for low-complexity applications.

V. CONCLUSION

The main agenda of this project goal is to enhance adaptive filtering, which yields the accurate approximation of the intended signal from the interference signals and exhibits high-performance outcomes. Implementation, analysis, and comparison of the Least Mean-Square (LMS), Normalised Least Mean-Square (NLMS), and Recursive Least Square (RLS) algorithms are conducted. These algorithms are studied using three performance criteria: the Beam pattern, the error performance, and the rate of convergence.

After comparing the algorithms, it can be concluded that the RLS algorithm achieves the fastest convergence rate and a lower mean square error. While the RLS algorithm provides better results than the LMS algorithm, it is preferred in situations requiring high precision and rapid convergence, while the LMS algorithm is better suited for applications.

However, it must be kept in mind that the input data, including the reference signal and the noise sequence, have an effective response on the performance of the system. As an output, the optimal adaptive filtering method depends on the application. Finding the adaptive filtering algorithm that suits better for the particular design and application demands better may be accomplished by testing many input datasets using the provided test environment, regardless of whether they are simulation or actual measurement data from the industry. The RLS adaptive filtering technique will be implemented in an integrated data processing unit in future development.

REFERENCES

[1] C. Li and S. Wang, "An improved Adaptive Beamforming Algorithm for 5G Interference-coexistence communication," 2019 IEEE/CIC International Conference on Communications Workshops in China (ICCC Workshops), Changchun, China, 2019, pp. 257261, DOI: 10.1109/ICCChinaW.2019.8849961.

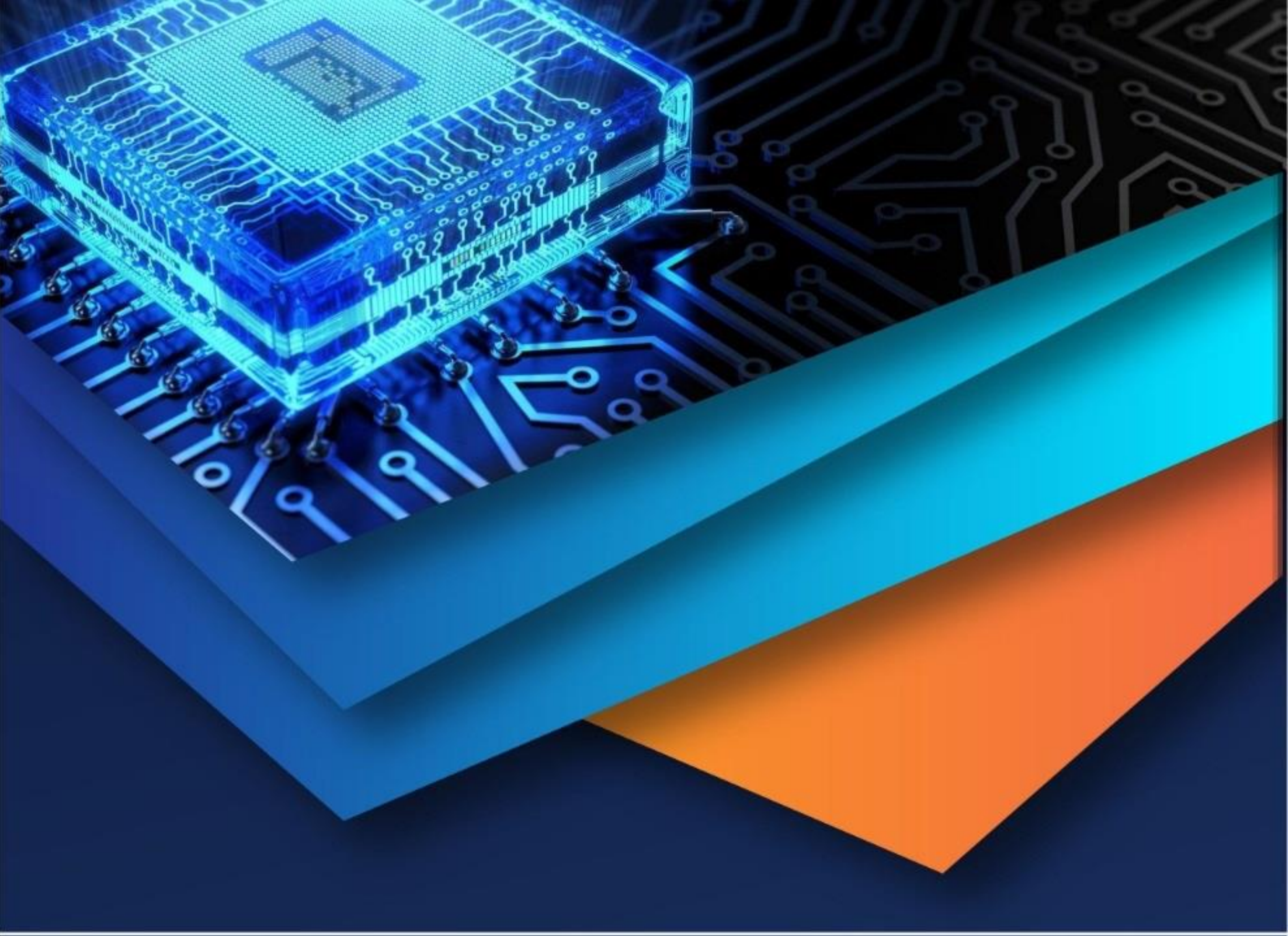
[2] "Adaptive Filter Theory," Fourth Edition, by Simon Haykin, Communications Research Laboratory, McMaster University, Hamilton, Ontario, Canada

[3] S. A. Ghauri and M. F. Sohail, "System identification using LMS, NLMS and RLS," 2013 IEEE Student Conference on Research and Development, Putrajaya, Malaysia, 2013, pp. 65-69, doi: 10.1109/SCOReD.2013.7002542.

[4] K. Elangovan, "Comparative study on the channel estimation for OFDM system using LMS, NLMS and RLS algorithms," International Conference on Pattern Recognition, Informatics and Medical Engineering (PRIME-2012), Salem, India, 2012, pp. 359-363, doi: 10.1109/ICPRIME.2012.6208372.

[5] Ç. Özçetin and S. Kent, "Noise Cancellation with Adaptive Filters on the Public Radio Spectrum," 2023 10th International Conference on Electrical and Electronics Engineering (ICEEE), Istanbul, Turkiye, 2023, pp. 273-278, doi: 10.1109/ICEEE59925.2023.00056.

[6] A. Parvathy and G. Narayanan, "Comparative Study of Adaptive Algorithms Using Matlab and Verilog," 2018 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), Chennai, India, 2018, pp. 1-6, doi: 10.1109/WiSPNET.2018.8538654.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY