

**Electrical and Computer Engineering Department**

**ECSE 512- Digital Signal Processing**

**Fall 2021 Term Project**

**Analysis of Adaptive FIR Filter Performance in Narrow Band Interference Suppression**

**Prepared By:**

|  |  |  |
| --- | --- | --- |
| Names: | Ahmad Sawwas | Jake Peterson |
| ID Number: | 260831244 | 260851431 |

|  |
| --- |
| **Abstract:** |
| This study presents an assessment of the performance of adaptive FIR filters in suppressing narrow band interferences corrupting signals with wideband desired contents. The effects of multiple parameters governing the overall system performance were considered. Adaptations were governed utilizing LMS and RLS algorithms where a comparison between both algorithms’ capabilities is presented. Simulations of both adaptation algorithms were conducted on MATLAB considering stationary and dynamic environments with different SNR levels. Results showed that, under appropriate tuning of the adaptive algorithm parameters and a suitable system delay, the FIR filter provides satisfactory results under all tested operating environments. |

**Table of Contents**

[I - Introduction 3](#_Toc89939614)

[II - Background Theory 4](#_Toc89939615)

[a. Finite Impulse Response (FIR) Filters 4](#_Toc89939616)

[b. Development of Linear Adaptive Filters for Narrow Band Interference Suppression 5](#_Toc89939617)

[c. Linear Adaptive Algorithms 6](#_Toc89939618)

[III – Operating Environments, Testing Approach and Methodology 8](#_Toc89939619)

[a. Operating Environment 8](#_Toc89939620)

[b. Testing Approach and Methodology 9](#_Toc89939621)

[IV – Simulation Results 10](#_Toc89939622)

[a. Adaptive FIR Filter Performance in Stationary Environment 10](#_Toc89939623)

[b. Adaptive FIR Filter Performance in Dynamic Environment 15](#_Toc89939624)

[V – Conclusions 15](#_Toc89939625)

[Appendix 16](#_Toc89939626)

[References 18](#_Toc89939627)

## I - Introduction

Filtering describes the functionality of a wide category of signal processing systems. It is the ability to extract (*filter*) desired information at a timestamp from an aggregate of preceding and current noisy data. Adaptive filters extend this concept further, describing a family of filters that allow the system to adjust for noise which traditional filter design, for various reasons, cannot compensate for. These filters can describe either linear or nonlinear functions on our data aggregate input. In this report, we limit our discussion to the first category.

The linear adaptive filter design problem seeks to minimize the effect of noise on the signal of interest. To do so, one needs to formulate a statistical model that describes the desired signal, the noise, and a quantization of the noise content in the output signal. Thus, one can view the filtering problem as a design optimization problem aiming to minimize an error signal, such as the difference between the filter’s actual output and the desired output.

In stationary environments, where the desired signal and the noise are not time varying, the Wiener filter, given reliable a priori statistics describing those quantities, describes an optimal solution to the filtering problem. However, such an approach is inadequate when the environment is non-stationary, since designing a filter requires finding and tuning specific parameters according to the statistical description of the surrounding environment. If such environment is time varying, then the optimally designed filter at the current time may not be optimal at later times.

Adaptive filters are self-designing filters: they rely on self-adaptive algorithms to update the filter parameters autonomously. This behavior addresses our above problem by providing an ability to track the state of the changing environment. Designed appropriately, adaptive filters are not only efficient for non-stationary environments but display an ability to converge to the Wiener filter in stationary environments without requiring a priori statistical description [1]. With such broad advantages, adaptive filtering sees significant use in diverse fields, including communications systems, navigation, and biomedicine, as well as other mechanical and electrical engineering specialities.

Several research articles discussed and tested the use of adaptive filtering in electrical power systems. Adaptive filtering techniques, such as adaptive notch filters [2] and adaptive traversal filters [3] improve active power filter (APF) performance by treating the estimated harmonic content (detected by the adaptive filter) as a reference signal. This reference is fed to an APF which generates equal but opposite currents at the point of connection, thus compensating for the harmonic contents caused by nonlinear loads.

Adaptive filtration possesses significant utility for various medical applications. For example, authors in [4] provided a comparison between different adaptive filter techniques, aiming to remove impulsive noise in MRI and ultrasound imaging. Adaptive filtration additionally sees frequent use in extracting medical signals. For instance, authors in [5] [6] [7] investigated the effectiveness of adaptive filtering in removing power line noises in Electromyography (EMG) and Electrocardiogram (ECG) signals. Another study provided a comparison between different adaptive filtering algorithms aiming to extract the ECG signal of a fetus [8].

Unmanned Aerial Vehicles (UAV) are another interesting opportunity for using adaptive filtering. Authors in [9] explored the use of recursive least square (RLS) and least mean square (LMS) based adaptive filters in extracting respiratory motion signatures captured by UAVs to compensate for interference from motion artefacts.

Communication systems constitute perhaps the most important and well-researched sector for adaptive filtration. Within the literature, a multitude of research articles aim to improve system performance across a diverse range of applications. For instance, authors in [10] proposed the implementation of an evolutionary variable step size LMS algorithm to remove noise from high-speed communication systems. References [11] [12] [13] [14] [15] tackled acoustic echo cancellation in communication channels, each investigating a unique adaptive algorithm. For more applications utilizing adaptive filters, readers are referred to [16].

This report aims to design, test, and compare the narrow band interference suppression performance of an LMS and RLS based adaptive finite impulse response (FIR) filter under both stationary and non-stationary environments.

We organize the rest of this report as follows: Section 2 provides a brief information about FIR filters, LMS algorithm and RLS algorithm. Section 3 involves modeling of the entire system quantities and describing operating environments. We present our results in section 4 and our conclusions in section 5.

## II - Background Theory

### Finite Impulse Response (FIR) Filters

FIR filters, also known as tapped-delay filters or transversal filters, consist of 3 basic elements as shown in Figure 1. These elements are a unit-delay element (represented by a ), a multiplier, and an adder. The number of delay elements defines the order of the FIR filter. Having such representation, the output of the FIR filter can be represented in terms of the input signal , and filter coefficients as shown in eq (1):

|  |  |
| --- | --- |
|  | (1) |

A picture containing text, wall

Description automatically generated

Figure 1: FIR Filter

### Development of Linear Adaptive Filters for Narrow Band Interference Suppression

The goal in FIR filter design is to determine an order and a set of tapped weights that achieve the desired performance. In static (stationary) environments (assuming a fixed order ) we are interested in finding a set that eliminates certain frequency bandwidths. In such conditions, the optimal tapped weights are computable and fixed as time elapses. However, in dynamic (non-stationary) environments, those weights must update constantly as the frequency of the desired or undesired signals might be time variant.

Providing such filter coefficients *a priori* is a deficient approach for several important reasons. When we consider dynamic environments (the most frequent use case for adaptive filters), continuously updating a set of filter coefficients is unfeasibly intensive. In addition, gathering enough information describing the operating environment to design filter coefficients can be exceedingly difficult even in static environments. If this data does not precisely describe the operating environment, the resulting filter will not achieve optimal performance.

As mentioned earlier, using adaptive filtering can help mitigate such limitations. Figure 2 shows a general scheme of utilizing adaptive filters for narrow band interference suppression. The main objective is to extract a wideband desired signal from the detected input signal , which consists of and some undesirable narrow band interference . This interference, being narrow banded, exhibits a strong degree of autocorrelation and thus can be estimated by the filter. In comparison, the wide band desired signal decorrelates rapidly over time. Thus, by introducing an adequate processing delay, we can guarantee that the signal fed to the adaptive filter is strongly correlated with the interference signal and completely uncorrelated with the desired signal. Hence, the filter will place a notch on the narrow interference band, leaving only the desired wideband signal as output.

Diagram

Description automatically generated

Figure 2: Narrow band suppression through adaptive filtering

### Linear Adaptive Algorithms

There are a variety of recursive algorithms that are used in the linear adaptive filtering problem. Each algorithm offers a desirable feature of its own. Thus, the main bulk of the challenge is twofold: understanding the capabilities and limitations of those adaptive algorithms and selecting the most appropriate one for the application considered. Generally, we can identify 2 disparate adaptive approaches that are used to update the tap weights of the adaptive filter. These are: (1) Method of Stochastic Gradient Descent and (2) Method of Least Squares.

*Method of Stochastic Gradient Descent (Least Mean Square “LMS” Algorithm):*The mean square error as a function of tapped weights, is an -dimensional paraboloid. Reaching the minimum point (optimal design) is done through updating the tapped weights, moving in the negative direction of the gradient vector. The algorithm is summarized by the set of equations illustrated in eq (2) – eq (6).

|  |  |
| --- | --- |
|  | (2) |
|  | (3) |
|  | (4) |
|  | (5) |
|  | (6) |

Here, represents the step size in the negative direction of the gradient of the minimization function (mean square error function) and is sample delay. For a more detailed derivation of the LMS algorithm, we refer readers to [1].

*Method of Least Squares (Recursive Least Square “RLS” Algorithm):*In RLS, we start the computation with a set of initial conditions and incrementally update them using information from new data samples. Unlike LMS, in RLS we seek to minimize an objective function illustrating the sum of squared error. With some modifications, the objective function to be minimized is represented in eq (7).

|  |  |
| --- | --- |
|  | (7) |

Here, represents length of input data, is the sample point at each instant of time, is a forgetting factor close to unity but less than 1, and is the regularization parameter. We also note that the tap weights of the FIR filter remain fixed during the observation interval .

The objective function shown in eq (7) consists of 2 parts: the first one represents the sum of weighted error squares whereas the second one represents the regularizing term. For thorough derivations for the equations shown below readers are referred to [1]. Under given reformulations, eq (7) can be rewritten as shown in eq (8).

|  |  |
| --- | --- |
|  | (8) |

Here, is the time average correlation matrix of the tap input vector . Similarly, we can formulate the time average cross-correlation vector between and .

|  |  |
| --- | --- |
|  | (9) |

According to the method of least squares, the optimum tapped weights minimizing eq (7) is defined by the normal equations represented in matrix format in eq (10).

|  |  |
| --- | --- |
|  | (10) |

However, finding the inverse of , especially for large filter orders, can be quite the computational burden. Therefore, finding the inverse matrix is done through computing the Riccati equation of the RLS algorithm. Hence, the RLS algorithm can be summarized by the equations illustrated in eq (11) – eq (13).

|  |  |
| --- | --- |
|  | (11) |
|  | (12) |
|  | (13) |

Where, **k[n]** is the gain vector and is .

## III – Operating Environments, Testing Approach and Methodology

### Operating Environment

To investigate implementing adaptive filtering for narrow band interference suppression, two different environments are considered. The first environment is static; both the desired and interference signal frequency spectra are fixed over time. The second environment is dynamic, meaning that part of the signal has a time varying frequency spectrum. (We divide this environment into two cases and explain further below.) The main objective of the adaptive filtering system is to estimate the interference signal and eliminate it from the output such that it faithfully recreates the desired signal. Both environments are investigated at low and medium signal-to-noise ratios (SNR) to gauge the benefit of both adaptive filtration techniques at various levels of noise. We describe both environments in further detail below:

The *Stationary Environment* was modelled through setting the desired signal to be a zero-mean white gaussian noise (WGN) and the narrow band interference to be a deterministic fixed-amplitude-single-frequency sinusoidal wave representing a single frequency.

The *Non-Stationary Environment****,*** in this study, was modelled considering two different scenarios. In the first scenario, we consider a speech signal superimposed with an undesired frequency as previously described. Second, we reconsider a WGN desired signal against an undesired frequency that changes over time. This compares favorably with most real applications, where interference frequencies corrupting the desired signal are not fixed with time. Table 1 illustrates a summary of the environments’ operating conditions considered in this study.

Table 1: Operating environments description

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Operating Conditions** | **SNR level** | **Interference Signal** | | **Desired Signal** |
| Stationary Scenario #1 | Low: SNR -11dB | f = 200Hz | Amp = 1v | WGN with power = -14dB |
| Stationary Scenario #2 | Medium: SNR 3dB | f = 200Hz | Amp = 2v | WGN with power = 6dB |
| Dynamic Scenario #1 | Low: SNR -10dB | f = 300Hz | Amp = 1v | Speech Signal |
| Dynamic Scenario #2 | Medium: SNR 3dB |  | Amp = 2v | WGN with power = 6dB |

### Testing Approach and Methodology

We investigate the performance of the adaptive FIR filter utilizing the 2 different adaptation algorithms described in section II, namely LMS and RLS. Defining the system the way shown in section II, we can generally divide the important parameters into 3 main categories. A list of these parameters with their corresponding category is provided in table 2.

This study was organized such that it fulfills the following objectives:

1. Investigate the effect of filter order on system performance in both LMS and RLS.
2. Study the effect of step size () on system performance
3. Explore the influence of varying regularization parameter () on system performance.
4. Compare both adaptation algorithms in terms of speed of convergence and estimation error.
5. Test the effects of SNR level on system performance
6. Analyze system performance under static and dynamic environments and assess the role of forgetting factor () in both environments.

Table 2: Parameters governing system performance

|  |  |  |  |
| --- | --- | --- | --- |
| **Filter Parameters** | **Adaptive Algorithm Parameters** | | **System Parameter** |
| **LMS** | **RLS** |
| Filter Order (M) | Step Size () | Regularization () | System Delay () |
| Tapped weights () | Forgetting Factor () |

## IV – Simulation Results

### Adaptive FIR Filter Performance in Stationary Environment

As mentioned in section II, the sample delay must be set to ensure that the desired component of the delayed input is uncorrelated with the present version. Given that noise signals are stochastic and stochastic processes captured at different instances are uncorrelated, setting to one suffices.

When utilizing LMS algorithm, only 2 variables control the system’s performance (filter order and step size ). However, when utilizing RLS algorithm there are 3 variables to be considered (filter order , regularization parameter , and forgetting factor ). Therefore, understanding the effect of each quantity on the system’s performance was pursued through fixing one parameter and varying the other. Following the above-mentioned approach, and for the 1st stationary environment mentioned in table 1, figures 3 and 4 were constructed.

Analyzing figure 3, the following observations can be concluded:

1. Increasing improves system performance by allowing the filter output to converge to the original interference signal (colored red) at a faster pace. This is subject to diminishing returns above a certain order threshold. For example, comparing the filter’s output signals for = 26 and = 34, almost no benefit was attained from increasing the filter’s order.
2. The step size similarly affects system performance to a significant degree by controlling the speed of convergence of the algorithm. Referencing figure 3, we see that small step sizes cause slow performance, and large step sizes can cause vigorous oscillations in filter output (blue colored signal) around the original interference signal. Such oscillations, although they illustrate faster performance, also represent a degradation in performance in terms of estimation error. Further increasing the step size eventually causes the algorithm to diverge and system to lose stability. Therefore, careful tuning of the step size is essential.

It is also important to mention that more advanced developed LMS algorithms implement variable step size approach to get faster tracking approach and smaller estimation errors. However, such algorithms are outside the scope of this report.

Graphical user interface

Description automatically generated with low confidence

Figure 3: Filter’s output utilizing LMS algorithm at different filter order (M) with (Top) and different step size () with order M = 26 (bottom)

Figure 4 reveals the following observations:

1. In terms of the speed of convergence, all simulated filter orders showed similar converging rates. Thus, we can conclude that the rate of convergence with RLS is not significantly sensitive to filter order . In particular, even a filter of low order (M = 2) is sufficient to provide a close estimation to the original interference signal.
2. The regularization parameter strongly affects the speed of convergence. As depicted from figure 4 (bottom), small values of elicit large oscillations (blue colored response) before decaying into smaller oscillations around the reference signal (colored red). Moreover, large values of slow the convergence rate. Extending such observations to cover the estimation error performance, low and high values of regularization parameters yield convergence to higher estimation errors.

Note that, when varying the forgetting factor , the performance of the system did not change. We thus conclude that system performance in static environments is insensitive to variations in .

Chart, line chart, histogram

Description automatically generated

Figure 4: Filter’s output utilizing RLS algorithm at different filter order (M) with (Top) and different regularization parameter () with order M = 26 and = 1 (bottom)

Results represented above are strong evidence of the ability of both LMS and RLS (when tuned properly) to accurately estimate the interference signal. However, for a better understanding of each algorithm’s strengths and weaknesses, we include a comparison below.

Considering the first stationary environment mentioned in table 1, the performance of an 18th order FIR filter was simulated using both adaptation algorithms having the best environment-tuned adaptation parameters ( = 0.01 and = = 1). The estimation error of each algorithm was calculated and plotted as shown in figure 5. After a thorough analysis of the outcome, we observe the following:

1. RLS provides a faster convergence than the LMS algorithm.
2. LMS is capable of converging to a closer estimate than that reached via RLS.

Therefore, we can say that RLS is preferred for applications when performance is mainly assessed on speed and not on its final state estimation, and LMS is preferred where the inverse is true.

Chart, line chart

Description automatically generated

Figure 5: Estimation errors utilizing LMS (in green) and RLS (in blue) algorithms

Graphical user interface, application

Description automatically generated

Figure 6: LMS and RLS based adaptive FIR filter performance in medium level SNR environment

Using filters is not always reserved for environments where the desired signal is significantly corrupted by external interferences. Thus, it is also important to investigate the performance of adaptive FIR filters with signals characterized by low energy noise content. To fulfill this objective, both LMS and RLS were utilized to test system performance considering the 2nd static environment described in table 1 (having SNR 3dB). Similarly, tuning for adaptation parameters was conducted before analyzing the performance of a 26th order filter. Figure 6 presents the performance attained in terms of the output signal (top) and the mean square estimation error (bottom) utilizing both adaptation algorithms. Analyzing Figure 6, we observe the following:

1. Both algorithms revealed acceptable performance in estimating the interference signal. Furthermore, asserting our previous conclusions, in medium SNR environments, RLS provides faster tracking and convergence whereas LMS provides closer final estimates.
2. When tuning LMS algorithm’s parameter, certain considerations are recommended. In particular, a moderate step size is preferable for low SNR and a smaller step size is preferable for medium SNR. In alternate terms: the stronger the interference content is in a signal, the easier it is for an LMS filter to capture and estimate with relatively moderate adaptation step size. The weaker such content is, the smaller the required step size gets.
3. When tuning the RLS parameters, the best value of was found to be 1000 where in the previous static environment such value was 1. Interpreting this observation requires an understanding of how parameter impacts the RLS algorithm. When trying to implement RLS as provided in section 2, iterations start by assigning an initial estimate of the inverse of the time average auto correlation matrix . Such value is initially estimated to be equal to inverse of regularization parameter multiplied by the identity matrix (). Therefore, assigning high regularization parameter elicits an initial small normed matrix, allowing the algorithm to effectively track the weak interference signal. On the other hand, in low level SNR medium, when the interference signal more strongly deteriorates the total input, increasing the initial norm of the inverse of the time average correlation matrix (selecting lower regularization parameter ) provides better performance.

Therefore, even in higher SNR environments, adaptive filtering can be a strong method for improving the quality of a captured signal through reducing its noisy contents. Furthermore, the SNR provides initial insight for tuning parameters in both LMS and RLS.

### Adaptive FIR Filter Performance in Dynamic Environment

*Case 1: Desired Signal is a Speech*:For this case, the interference signal consists of a single frequency tone at 300 Hz, and the desired signal consists of 10 seconds of speech taken from freely available tracks on the Open Speech Repository. Dynamic environments constructed from these tracks were configured to have an SNR of –10 dB, and ten tracks were used to study the environment. An important quality about this environment is that speech does not decorrelate immediately like gaussian noise does, so the autocorrelation of the speech samples was calculated and found to be 8 samples on average. This value is used for .

*Case 2: Desired Signal is WGN*: The interference signal (in this scenario) was modelled to update its frequency every 0.5 seconds. It starts initially with a frequency of 50Hz and increases in a stepwise manner every 0.5 over the course of five seconds.

The ability of the proposed system to estimate the desired signal, utilizing both adaptive algorithms was illustrated in Figures 7 and 8. In both figures, we observe that the forgetting factor () affects the performance, but this effect is more drastic for time-variant noise environments. For instance, when was set to unity, the system in Figure 7 sees a marked decrease in estimation error whereas Figure 8 exhibits an increase as the frequency of the interference signal increases with time. Modifying that value slightly eliminates such phenomenon, however, it leads to different estimation error thresholds; here, setting = 0.998 yielded the best results.

The forgetting parameter in the RLS algorithm plays the role of assigning weights on samples. In environments with time-variant noise, adaptation must be strongly oriented toward recent observations and such orientation must decay as these observations becomes older and older. Thus, setting to be different from unity provided better performance as this gears the algorithm to eliminate the current noise content rather than giving equal importance to signals captured previously, which hold outdated noise content. Recall that in static environments (or environments where the noise is static) this parameter’s role is reduced because the interference is time invariant. Thus, accumulating even very old observations increases the convergence rate.

## V – Conclusions

In this report, a thorough assessment and analysis of the performance of adaptive FIR filters in stationary and dynamic environments was presented. Adaptations were conducted utilizing LMS and RLS algorithms. Under all tested operating environments, the designed system showed significant performance in estimating the narrow band interference signal and extracting the desired one when properly tuned.

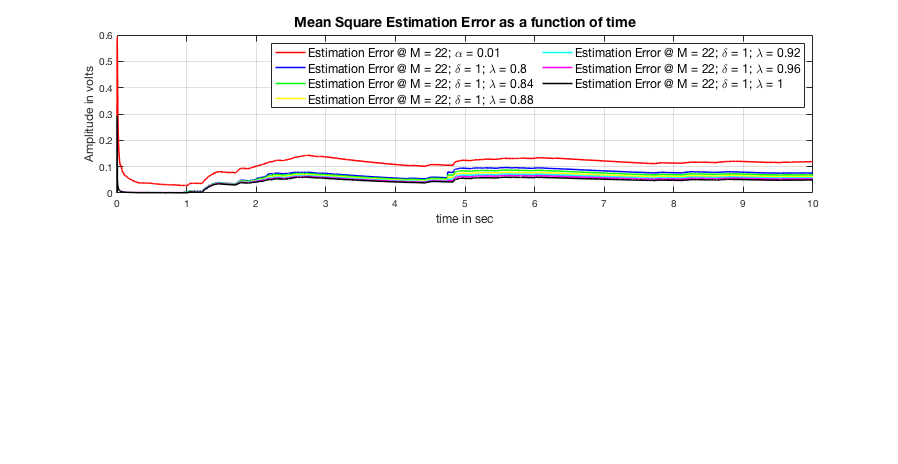


Figure 7: LMS and RLS based adaptive filter performances under speech dynamic environment

Calendar

Description automatically generated

Figure 8: LMS and RLS based adaptive filter performances under time-varying noise

Simulations revealed that increasing filter order M improves performance, however to a certain extent. Oversizing M, above a certain value, will only increase system’s memory requirements without significant improvement in the performance. In terms of the algorithms’ general performance features, LMS showed better convergence ability than RLS at the expense of slower convergence rate. Adaptation algorithms’ parameters () significantly influence the overall performance. Selecting an appropriate value of these parameters is strongly dependent on the operating environment conditions (stationary or dynamic, SNR level, etc…).

## Appendix

Please consider the following to successfully run the m-files attached to this report. There are 3 main folders attached. Each one of these folders contains scripts that belong to certain operating environment.

In the folder named “Stationary Environment Scripts, there are 6 MATLAB scripts. 2 of them are used to create the operating environments (“StatEnv\_LSNR.m” and “StatEnv\_MSNR.m”). The scripts “LMSfilter.m” and “RLSfilter.m” represents LMS and RLS algorithms implementation respectively. The script named “EnsembleMean.m” is a subfunction called in the main script to compute ensemble mean error. Finally, the script “LMSRLS.m” is a script that provides the performance of both algorithms on same plots (for comparison). To successfully run the scripts with adaptive algorithms, you must first run the script defining the operating environment (“StatEnv\_LSNR.m” or “StatEnv\_MSNR.m”), save the workspace as “StatEnvLSNR.mat” or “StatEnvMSNR.mat” and then update line 4 in “LMSfilter.m” and/or “RLSfilter.m” or lines 5 and 44 in “LMSRLS.m” to load that workspace. Finally, run any of the MATLAB files “LMSfilter.m”, “RLSfilter.m” or “LMSRLS.m” and observe the outputs. Note that comments are provided within the MATLAB scripts so that users can freely adjust algorithms parameters and simulate the system’s performance.

In the folder named “Dynamic Environment 1 Scripts” there are 6 MATLAB scripts and a folder of sound clips labelled “Samples”. Similarly, to successfully run those scripts, the user must first run the script “DynmEnvLSNR.m” defining the operating environment, save that workspace as “DynmEnvLSNR.mat”, and then run any of the adaptation algorithms provided. Likewise, in the folder named “Dynamic Environment 2” there are 5 MATLAB scripts. To Run these, the user should run “DynEnvMSNR”, save the MATLAB workspace as “DynEnvMSNR.mat”, and then use the desired filter script.

Make sure to clear the MATLAB workspace between switching environments to avoid errors in filtration scripts.

# References

|  |  |
| --- | --- |
| [1] | S. Haykin, Adaptive Filter Theory, Ontario: Pearson Education, 2014. |
| [2] | D. Yazdani, A. Bakhshai, G. Joos and M. Mojiri, "An Adaptive Notch Filtering Approach for Harmonic and Reactive Current Extraction in Active Power Filters," in *34th Annual Conference of IEEE Industrial Electronics*, Orlando, 2008. |
| [3] | A. Mohamed, A. Berzoy and O. A. Mohammed, "Adaptive Transversal Digital Filter for Reference Current Detection in Shunt Active Power Filter," in *IEEE Power & Energy Society General Meeting*, Denver, 2015. |
| [4] | A. M. Vispute, B. U. Rindhe and D. N. Shekokar, "Efficient Adaptive Mean Filtration Technique in Denoising of Medical Imaging," in *International Conference on Global Trends in Signal Processing, Information Computing and Communication*, Jalgaon, 2016. |
| [5] | M. Malboubi, F. Razzazi, M. Aliyari Sh. and A. Davari, "Power line noise elimination from EMG signals using adaptive Laguerre filter with fuzzy step size," in *17th Iranian Conference of Biomedical Engineering (ICBME)*, Isfahan, 2010. |
| [6] | S. O. Gilani, Y. Ilyas and M. Jamil, "Power line noise removal from ECG signal using notch, band stop and adaptive filters," in *International Conference on Electronics, Information, and Communication (ICEIC)*, Honolulu, 2018. |
| [7] | A. C. Mugdha, F. S. Rawnaque and M. U. Ahmed, "A study of recursive least squares (RLS) adaptive filter algorithm in noise removal from ECG signals," in *International Conference on Informatics, Electronics & Vision (ICIEV)*, Fukuoka, 2015. |
| [8] | R. R. Vaidya and N. Chaitra, "Comparison of Adaptive filters in extraction of Fetal ECG," in *International Conference on Smart Electronics and Communication (ICOSEC)*, Trichy, 2020. |
| [9] | S. M. M. Islam, L. C. Lubecke, C. Grado and V. M. Lubecke, "An Adaptive Filter Technique for Platform Motion Compensation in Unmanned Aerial Vehicle Based Remote Life Sensing Radar," in *50th European Microwave Conference (EuMC)*, Utrecht, 2020. |
| [10] | H. B. M. Ajjaiah, D. B. Ganure, P. M. Hadalgi, P. V. Hunagund, A. S. Nayak and B. R. Prakash, "Adaptive filters using EVSSLMS algorithms based on the communication system," in *International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES)*, Paralakhemundi, 2016. |
| [11] | U. Mahbub and S. A. Fattah, "Gradient based adaptive filter algorithm for single channel acoustic echo cancellation in noise," in *7th International Conference on Electrical and Computer Engineering*, Dhaka, 2012. |
| [12] | L. Yunwei, "Study on Adaptive Filtering Algorithm in the Echo Cancellation in Mine," in *IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC)*, Dalian, 2021. |
| [13] | B. Yang and J. Feng, "Design of adaptive filtering algorithm and its application in chaotic echo cancellation," in *24th Chinese Control and Decision Conference (CCDC)*, Taiyuan, 2012. |
| [14] | M. Hamidia and A. Amrouche, "Effect of a signal decorrelation on adaptive filtering algorithms for acoustic echo cancellation," in *5th International Conference on Electrical Engineering - Boumerdes (ICEE-B)*, Boumerdes, 2017. |
| [15] | D. N. Priyanka and H. M. Chandrashekar, "Implementation and comparison of echo cancellation algorithms using adaptive filtering techniques in TMS320C6748 DSK," in *2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, Bangalore, 2017. |
| [16] | J. G. Avalos, J. C. Sanchez and J. Velazquez, "Applications of Adaptive Filtering," in *Adaptive Filtering Applications*, Dr Lino Garcia (Ed.), ISBN: 978-953-307-306-4, InTech, 2011, pp. 1- 20. |