Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2016) 24: 4322 – 4338

doi:10.3906/elk-1410-76

A new step-size searching algorithm based on fuzzy logic and neural networks for LMS adaptive beamforming systems

Walter OROZCO-TUPACYUPANQUI, Mariko NAKANO-MIYATAKE, Hector PEREZ-MEANA*
Postgraduate Section, Mechanical Electrical Engineering, Instituto Politecnico Nacional, Mexico

Received: 13.10.2014 • **Accepted/Published Online:** 04.08.2015 • **Final Version:** 20.06.2016

Abstract: In this paper, a novel algorithm based on fuzzy logic and neural networks is proposed to find an approximation of the optimal step size μ for least-mean-squares (LMS) adaptive beamforming systems. A new error ensemble learning (EEL) curve is generated based on the final prediction value of the ensemble-average learning curve of the LMS adaptive algorithm. This information is classified and fed into a back propagation neural network, which automatically generates membership functions for a fuzzy inference system. An estimate of the optimal step size is obtained using a group of linguistic rules and the corresponding defuzzification method. Computer simulations show that a useful approximation of the optimal step size is obtained under different signal-to-noise plus interference ratios. The results are also compared with data obtained from a statistical analysis performed on the EEL curve. As a result of this application, a better mean-square-error is observed during the training process of the adaptive array beamforming system, and a higher directivity is achieved in the radiation beam patterns.

Key words: Adaptive filtering, adaptive beamforming, neural-fuzzy systems, least-mean-square algorithm, membership functions

1. Introduction

Currently, the development of communication systems based on adaptive algorithms is widely used to minimize the effects of noise and interferences. One of the most commonly used systems is the smart antenna array [1–3], whose filtering capability depends on the structure and the adaptive algorithm used in its design. For arrays that incorporate the standard least-mean-squares (LMS) algorithm [4–6], the updating weights, speed of convergence, and steady-state stability generally depend on the selected fixed step size μ. Therefore, the main problem focuses on determining the appropriate step size μ to obtain the best qualities of the adaptive algorithm, which is the core of the smart antenna array. For this reason, the step size μ must be carefully selected to maintain a balance between a low steady-state error and a rapid convergence rate in response to the changing environment. Moreover, a good step size can minimize the mean-square-error (MSE) during the learning process of the adaptive algorithm.

This paper proposes a novel neural-fuzzy algorithm to seek the optimal step size for LMS adaptive arrays. The proposed search algorithm consists of three processing units: the learning block, the neural-fuzzy unit, and the searching stage. In the first part of the algorithm, the error ensemble learning (EEL) curve is generated. In addition, curve-fitting methods and divergence analysis are included to process the EEL curve and obtain suitable information to remove irregularities due to contamination by additive white Gaussian noise (AWGN).

*Correspondence: hmperezm@ipn.mx

In the next stage, the previous information is clustered using the fuzzy c-mean and the max-membership methods [7]. The data points are assigned membership values according to the cluster to which they belong. A back propagation artificial neural network uses the data points and the corresponding membership values to generate the input and output membership functions of the fuzzy inference system (FIS). Finally, in the last part of the algorithm, an estimate of the optimal step size is obtained using a group of IF-THEN rules and the centroid defuzzification method. To evaluate the proposed search algorithm, a single LMS adaptive array and a least-recurve-mean-square (LRMS) hybrid cascade array beamforming [8] system are used.

The organization of this paper is as follows. In section 2, the convergence analysis of the LMS algorithm is reviewed alongside the problem of finding its optimal step size. In section 3, the methods used for each part of the proposed algorithm are detailed. The results of the search experiments performed to evaluate the estimate of the optimal step size are reported and discussed in section 4. Finally, the conclusions are presented in section 5.

2. Convergence of the LMS algorithm

To understand the dilemma of finding the step size, let us examine the basic concepts and equations regarding the convergence of an LMS adaptive filter as depicted in Figure 1.

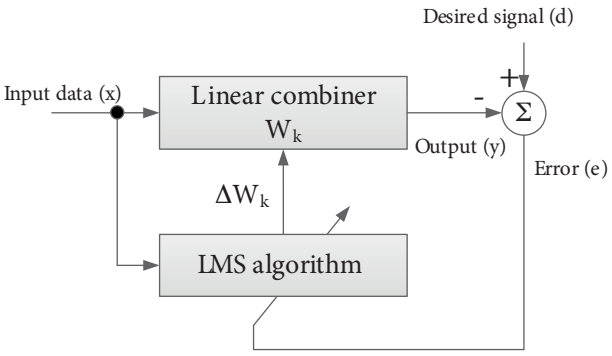

Figure 1. Adaptive LMS filter.

For a transversal linear combiner of N weights, the overall adaptive filter response may be found by considering the sum of the samples from the signal source [9]; therefore,

$$y_k = \mathbf{X}_k^T \mathbf{W}_k, \tag{1}$$

where $\mathbf{X}_k$ and $\mathbf{W}_k$ are the sample input and the adaptive-filter coefficient vectors, respectively. For this case, both vectors are columns. The subscript k is the time index or the sample number, and the operator T represents transposition. The adaptation process of the weights is performed by the following LMS equations [10,11]:

$$e_k = d_k - \mathbf{X}_k^T \mathbf{W}_k, \tag{2}$$

$$\mathbf{W}_{k+1} = \mathbf{W}_k + \mu e_k \mathbf{X}_k. \tag{3}$$

Eq. (2) calculates the instantaneous output error between the desired signal d_k and the output of the filter y_k, and Eq. (3) updates the weights in each iteration of the algorithm. The parameter μ, known as the step size, regulates the speed and stability of adaptation. If we assume this procedure to be a stationary process in

which successive input vectors are independent over time, and $\mathbf{X}_k$ and $\mathbf{W}_k$ are mutually independent, then the expected value of the weight vector converges to the Wiener solution $\mathbf{W}_{op} = \mathbf{R}^{-1}\mathbf{P}$ [11] after an adequate number of iterations. By substituting Eq. (2) into Eq. (3), after some straightforward manipulations, the expected value of Eq. (3) can be expressed as

$$E[\mathbf{W}_{k+1}] = E[\mathbf{W}_k] + \mu\left(E[d_k\mathbf{X}_k] - E[\mathbf{X}_k\mathbf{X}_k^T]E[\mathbf{W}_k]\right). \tag{4}$$

In Eq. (4), $E[d_k\,\boldsymbol{X}_k]$ represents the cross-correlation vector $\mathbf{P}$ between the desired and input signals, while $\mathrm{E}[\mathbf{X}_k\,\mathbf{X}_k^T]$ is the input autocorrelation matrix $\mathbf{R}$ [12,13]. Thus, after some substitutions and manipulations, Eq. (4) can be written as [12,14]

$$E[\mathbf{W}_{k+1}] = E[\mathbf{W}_k] + \mu\left(\mathbf{P} - \mathbf{R}E[\mathbf{W}_k]\right). \tag{5}$$

Subtracting $\mathbf{W}_{op}$ from both sides of Eq. (5), and defining $\mathbf{M}_k = \mathbf{W}_k - \mathbf{W}_{op}$, it follows that

$E[\mathbf{M}_{k+1}] = (\mathbf{I} - \mu\mathbf{R})\,E[\mathbf{M}_k]$. (6) Next, using the fact that the autocorrelation matrix $\mathbf{R}$ can be expressed as $\mathbf{R} = \mathbf{Q}^H\Lambda\mathbf{Q}$ [13], where the operator H is the Hermitian transposition for matrices and $\mathbf{Q}$ is an orthonormal transformation whose columns are the eigenvectors of $\mathbf{R}$, Eq. (6) becomes

$$E[\mathbf{M}_{k+1}] = \mathbf{Q}^H\left(\mathbf{I} - \mu\Lambda\right)\mathbf{Q}E[\mathbf{M}_k]. \tag{6}$$

Next, upon multiplying Eq. (6) on the left by $\mathbf{Q}$ and defining $\mathbf{V}_k = \mathbf{Q}E\left[\mathbf{M}_k\right]$, it follows that

$$\mathbf{V}_{k+1} = \left(\mathbf{I} - \mu\Lambda\right)\mathbf{V}_k. \tag{7}$$

Finally, iterating Eq. (7) starting from an arbitrary initial solution $\mathbf{V}_0$, it follows that

$$\begin{bmatrix} V_{k,0} \\ V_{k,1} \\ \vdots \\ V_{k,N-1} \end{bmatrix} = \begin{bmatrix} (1-\mu\lambda_0)^k & & & \\ & (1-\mu\lambda_1)^k & & \\ & & \ddots & \\ & & & (1-\mu\lambda_{N-1})^k \end{bmatrix} \begin{bmatrix} V_{0,0} \\ V_{0,1} \\ \vdots \\ V_{0,N-1} \end{bmatrix}. \tag{8}$$

Thus, the expected value of the weight vector will converge to the Wiener solution if

$$\lim_{k\to\infty}\left(\mathbf{I} - \mu\Lambda\right)^k = \mathbf{0} \Rightarrow |1 - \mu\lambda_n| < 1 \;\; for \;\; k = 0, 1, \ldots N-1. \tag{9}$$

That is, the convergence of the LMS adaptive algorithm is guaranteed if and only if [12]

$$0 < \mu < \frac{1}{\lambda_{\max}}. \tag{10}$$

In general, the trace of the input autocorrelation matrix $\mathbf{R}$, the sum of the elements on its main diagonal, is greater than its largest maximum eigenvalue; therefore, an alternative expression of Eq. (10) can be used to formulate the convergence condition, as follows [12]:

$$0 < \mu < \frac{1}{tr[\mathbf{R}]} = \frac{1}{\mathbf{X}^T\mathbf{X}}. \tag{11}$$

Eq. (11) provides the range of μ in which the adaptive algorithm converges; however, the optimal step size is still unidentified.

3. The proposed search algorithm

The flow diagram of the proposed search algorithm is shown in Figure 2. To describe the structure of the new algorithm, three subblocks have been defined. First is the learning block, in which the behavior of the least-mean-square (LMS) adaptive algorithm is characterized by generating a new error ensemble learning (EEL) curve. Second is the fuzzy-neural block in which the membership functions are automatically generated for the fuzzy inference system (FIS) using fuzzy clustering and neural network techniques. Finally, in the last block, the searching block, an approximation of the optimal step size μ is obtained for the adaptive algorithm as a result of the combination of different fuzzy rules.

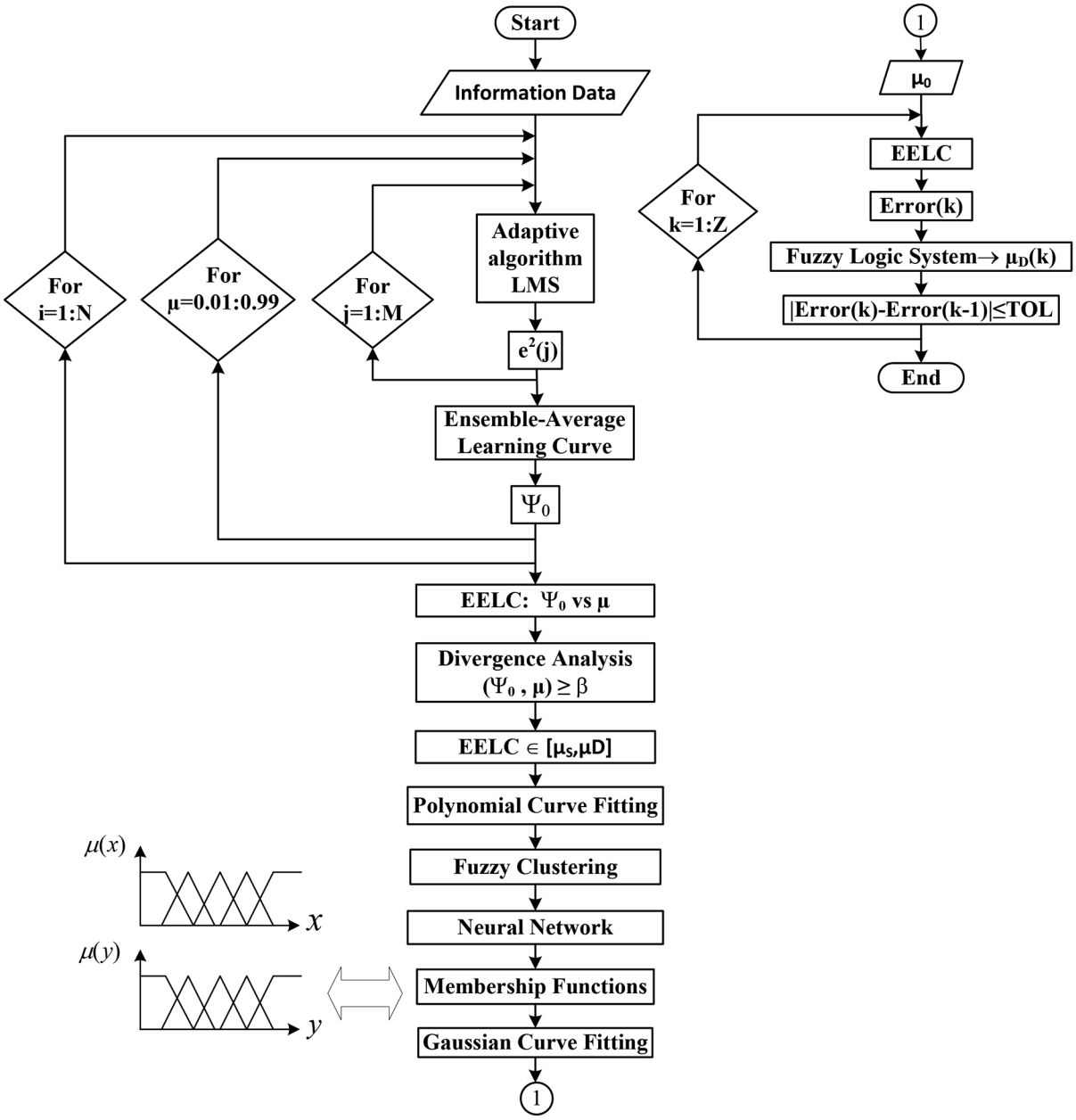

Figure 2. Flow diagram of the proposed search algorithm.

3.1. The learning block

The learning block estimates the error behavior of the LMS algorithm. First, M independent experiments are performed to train the adaptive filter under the same input statistical conditions but for different step sizes. For each k training process, an instantaneous learning curve of N_T samples is calculated by [10]

$$\left(e_i^{(k)}\right)^2 = \left(d_i - x_i w_i\right)^2, \;\; 0 \leq i \leq N_T \tag{12}$$

Because it is essential to evaluate the performance of the adaptive algorithm, an ensemble-average learning (EAL) curve is computed after all k experiments have concluded. A common approach is to average the independent instantaneous learning curves as follows [10]:

$$\xi = \frac{1}{M}\sum_{k=1}^{M}\left[e_i^{(k)}\right]^2, \;\; 0 \leq i \leq N \tag{13}$$

By examining the EAL curve of the LMS algorithm, it is possible to extract important information about the adaptation process, such as the error variance that remains in steady state and whether the adaptive algorithm reaches such a state. For example, by averaging over 200 training processes of such instantaneous learning curves, a typical EAL curve is obtained and shown in Figure 3. Once the previous calculations have been performed, the simple exponential smoothing (SES) method [11] is used to estimate the final magnitude of this curve. This method is appropriate to predict data without a clear tendency towards a specific value. Although the data in Figure 3 exhibit a certain trend to a specific MSE, a better final estimation is required to build the EEL curve. The problem is observed due to the presence of a slight sinusoidal behavior in the curve. After the ensemble-average learning curve is calculated, whether or not it reaches the steady state of convergence, its horizontal axis is divided into N intervals with M samples containing the amplitude of the curve. Figure 3 shows this procedure for three segments.

An average initial value Ψ_0 is calculated by Eq. (14) for each segment $[S_i, S_e]$ using the corresponding sequences of data Ψ_i. The estimate for each sequence of information is given by Eq. (16):

$$\Psi_0^{(n)} = \frac{1}{M}\sum_{m=1}^{M}\xi(m), \;\; 1 \leq n \leq N, \tag{14}$$

$$\Psi_i(m) = \xi(m) \in [S_i, S_e], \tag{15}$$

$$\Psi^{(n)}(k) = (1-\alpha)\Psi_0 + \alpha\Psi_i(k-1), \;\; 1 \leq k \leq M. \tag{16}$$

Finally, the estimated value of the complete ensemble-average-learning (EAL) curve is calculated by averaging the individual estimates as follows:

$$\Psi_0 = \frac{1}{N}\sum_{n=1}^{N}\Psi^{(n)} \;\; for\, \mu \in [\mu_s, \mu_e]. \tag{17}$$

Because this procedure is performed for each step size belonging to an interval under analysis $[\mu_s, \mu_e]$, the statistical behavior of the variability of the final estimated error for each ensemble-learning curve is obtained.

As a result of this computation, a new curve defined as the error ensemble learning curve (EEL) is generated. This curve is shown in Figure 4 for a specific μ-interval $[\mu_s,\ \mu_f]$.

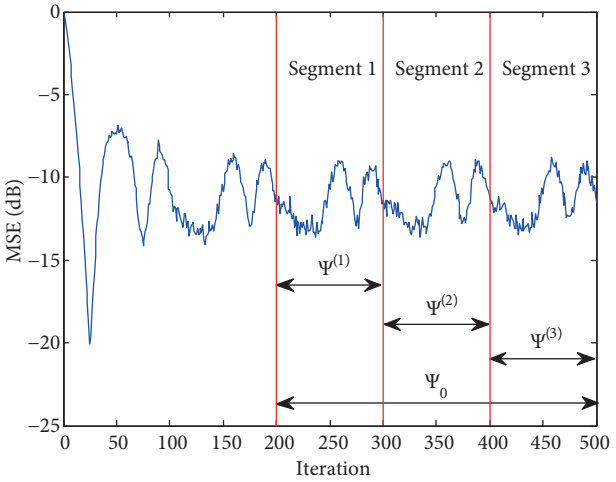

Figure 3. EAL curve divided into three sections to estimate the final value.

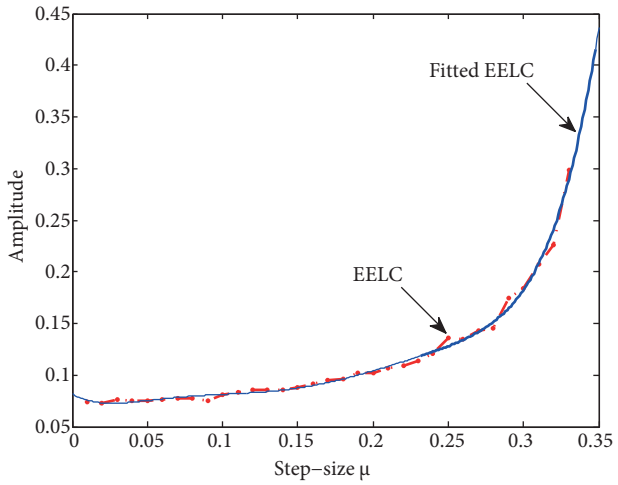

Figure 4. TEEL curve and fitted TEEL curve.

According to the graph, it is possible that the increment of the parameter μ also implies an increment of the error magnitude. As a consequence, there is the possibility that for some range of step sizes, the LMS algorithm diverges. Therefore, to avoid obtaining unnecessary data due to divergence, a simple divergence analysis is incorporated into the algorithm. When the error magnitude given by Eq. (17) exceeds a certain threshold β for each step size, this value is eliminated from the curve, and only a significant part of the data are used for analysis. In fact, Figure 4 shows the truncated error ensemble learning (TEEL) curve after the divergence analysis was performed for a $\mu \in [0.01,1]$. A representation of this procedure is given by the following equation:

$$IF\ |\Psi_{EEL}| \in [\mu_s, \mu_e] > \beta \Rightarrow |\Psi_{TEEL}| \in [\mu_s, \mu_f]\, where\, [\mu_s, \mu_f] \subseteq [\mu_s, \mu_e]. \tag{18}$$

Although the TEEL curve is a worthy source of information to generate membership functions, it is not a smooth curve due to contamination from different sources of noise. Therefore, a polynomial curve fitting process is used to construct a curve that has the best fit to the data points. One of the most common fitting techniques is polynomial regression [15], in which the least-squares procedure can be readily extended to fit the data to a higher-order polynomial given by

$$f(x) = \sum_{i=0}^{N} P_i x^i, \tag{19}$$

where N is the order of the polynomial and P_i denotes its coefficients. In Figure 4, both the original TEEL and the fitted TEEL curves are shown. It is important to indicate that similar curves can be obtained under the same SINR, which occurs due to the random characteristics of the noise interference.

3.2. The fuzzy-neural block

After the data have been fitted according to the desired polynomial, this information is loaded into the fuzzy-neural block, in which the membership functions will be generated. First, the information is classified into two

sets of data, one for the input (amplitude of the EEL curve) and another for the output (truncated μ-interval) of the fuzzy inference system (FIS). For each set of information, M different classification groups are generated using fuzzy c-means clustering and the max-membership method. The M groups are defined according to the number of membership functions required for the input and output of the FIS, respectively. To establish the location of each data sample for a specific group in the space, each cluster requires a center with m coordinates to describe its location. Therefore, the cluster center is given by

$$v_{ij} = \frac{\sum_{k=1}^{n} \alpha_{ik}^{\delta} \cdot x_{ki}}{\sum_{k=1}^{n} \alpha_{ik}^{\delta}}, \; j = 1, 2, \ldots J, \tag{20}$$

where α_{ik} is the membership of the kth data point in the ith cluster, and the parameter $\delta \in [1, \infty)$ is defined as the weighting parameter. The subscript j represents the total number of coordinates for each center cluster. The assignment of each data sample to its corresponding cluster is given by minimizing an object function J_m defined as

$$J_m = \sum_{k=1}^{n} \sum_{i=1}^{c} (\alpha_{ik})^{\delta} \left[\sum_{j=1}^{m} (x_{kj} - vi_j) \right]^2, \tag{21}$$

$$J_m = \sum_{k=1}^{n} \sum_{i=1}^{c} (\alpha_{ik})^{\delta} (d_{ik})^2, \tag{22}$$

where d_{ik} in Eq. (22) is the Euclidean distance between the ith group center and the kth data set. Finally, a soft partition matrix $\mathbf{U}_s$ is defined to assign membership values to each point of information. The elements of the classification matrix are updated using a recursive process that involves a set of rules described in [7] along with the following equations:

$$J_m^{(opt)}(\mathbf{U}_s^{(opt)}, \mathbf{v}^{(opt)}) \rightarrow \min J(\mathbf{U}_s, \mathbf{v}), \tag{23}$$

$$U_s^{(r+1)} = \begin{bmatrix} \alpha_{11}^{(r+1)} & \alpha_{12}^{(r+1)} & \cdots & \alpha_{1N}^{(r+1)} \\ \alpha_{21}^{(r+1)} & \alpha_{22}^{(r+1)} & \cdots & \alpha_{2N}^{(r+1)} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{c1}^{(r+1)} & \alpha_{c2}^{(r+1)} & \cdots & \alpha_{cN}^{(r+1)} \end{bmatrix}, \tag{24}$$

$$\alpha_{ik}^{(r+1)} = \left[\sum_{j=1}^{c} \left(\frac{d_{ik}^{(r)}}{d_{jk}^{(r)}} \right)^{\frac{2}{m-1}} \right]^{-1} \quad r \geq 0, \tag{25}$$

where r is the number of iterations, $\mathbf{U}_s^{(opt)}$ is the optimal soft partition matrix, and $\mathbf{v}^{(opt)}$ is the corresponding optimal cluster center vector. Once the partitions are obtained for both the amplitude of the TEEL curve and the truncated μ-interval, each point of information is assigned membership values between 0 and 1 for the classes into which they were placed. Hence, a single point can have partial membership in more than one group.

For example, the following soft partition matrix $\mathbf{U}_s$ might be the outcome for three clusters and the five first samples of the TEEL amplitude curve:

$$\mathbf{U}_s = \begin{bmatrix} 0.0198 & 0.0002 & 0.9945 & 0 & 0.0076 \\ 0.8101 & 0.9984 & 0.0014 & 0.9998 & 0.0212 \\ 0.1701 & 0.0014 & 0.0041 & 0.0002 & 0.9712 \end{bmatrix}$$

It is important to note that the sum of each column in $\mathbf{U}_s$ is one. Next, the final fuzzy partition for each point of information is achieved by hardening the fuzzy classification matrix $\mathbf{U}_s$. One of the most popular approaches is the max-membership method [7], in which the largest element of the partition matrix is assigned a value of one and the remaining elements in each column are assigned a value of zero. The hardening equation used for this purpose is given by

$$\alpha_{ik} = \max_{j \in c} \{\alpha_{ik}\} \rightarrow \alpha_{ik} = 1; \;\; \alpha_{jk} = 0, \forall j \neq i \; for \, i = 2, \ldots, c \, and \, k = 1, 2 \ldots, n \tag{26}$$

For the soft partition matrix $\mathbf{U}_s$ given as an example, the final hard matrix U_H is

$$\mathbf{U}_H = \begin{bmatrix} 0 & 0 & 1 & 0 & 0. \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The final hard matrix $\mathbf{U}_H$ is used to train a back propagation neural network to generate the membership functions. In Figure 5, a schematic representation of the cluster idea with the fuzzy c-means and max-membership method is shown.

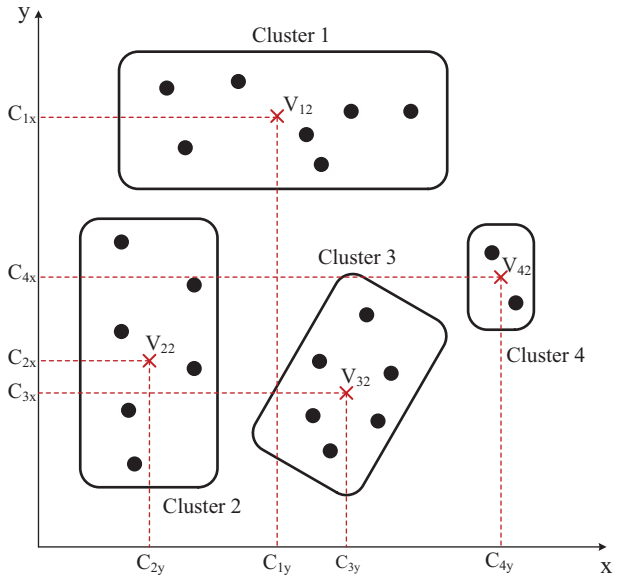

Figure 5. Fuzzy hard partition.

Because there is a straight relationship between the data point and the membership values, this characteristic is learned by the neural network during the training process. In this manner, the neural network classifies data points into one of the earlier defined clusters. Once the neural network is ready, a larger sample

of the amplitude of the TEEL curve and the truncated μ-interval are entered into it to generate their respective membership functions. An example of the membership functions is shown in Figure 6.

To obtain a better shape for the membership functions, a Gaussian fitting curve is used to obtain the final result. The equation for the Gaussian model is given by [16]

$$g(x) = \sum_{i=1}^{N} \alpha_i e^{\left[-\left(\frac{x-\beta_i}{\sigma_i}\right)^2\right]}, \tag{27}$$

where α_i is the amplitude, usually one for membership functions; β_i is the position of the center of the peak and σ_i is related to the peak width; and N is the total number of functions used to fit the data points. Figure 7 shows the fitted membership curves. For the case of the Gaussian fitting method, there are small differences between the original curve and its fitted version, but the main characteristics such as the center peak and the width are maintained.

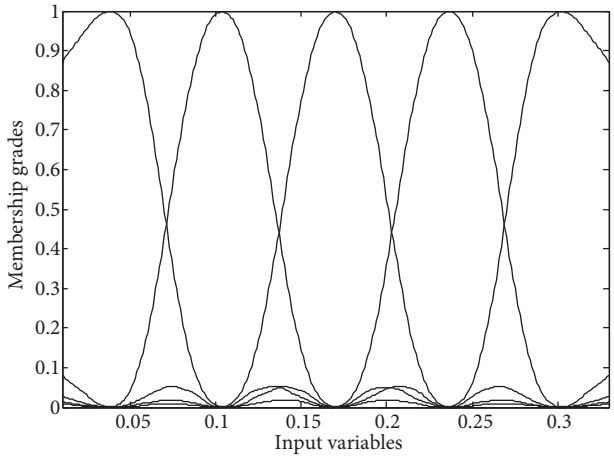

Figure 6. Typical membership functions obtained by the neural networks.

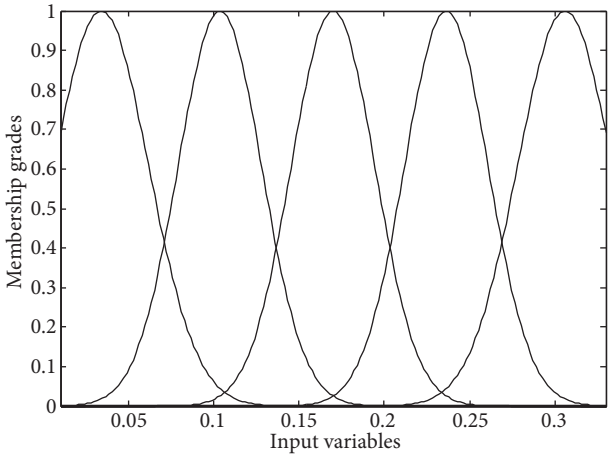

Figure 7. Fitted Gaussian membership functions.

3.3. The search block

For the search block, a fuzzy inference system (FIS) is designed according to the Mamdani rules [17] whose equations are given by a collection of M linguistic IF-THEN single-input, single-output propositions:

$$\text{IF } x \, is \, A_i, \text{ THEN } y \text{ is } B_i, \text{ for } i = 1, 2, \ldots, M, \tag{28}$$

where A_i and B_i are fuzzy numbers. This method of representing human knowledge is known as the deductive form, in which a conclusion (consequent) is derived from a given known fact (antecedent). For the proposed algorithm, an initial random step size μ_0 begins the search process for the optimal μ. The errors calculated based on the initial condition and the subsequent step sizes are the inputs of the fuzzy inference system; therefore,

$$\mu_D^{(i+1)} = FIS(\mu_D^{(i)} \rightarrow \varepsilon^{(i)}) \tag{29}$$

The fuzzy variable μ_D is defuzzified by the center of gravity method described by the following equation [7]:

$$\mu_D = \frac{\sum_{k=1}^{N} \mu_k \varphi(\mu_k)}{\sum_{k=1}^{N} \varphi(u_k)}, \tag{30}$$

where μ_k is the selected value on the abscissa of the output membership function, $\varphi(\mu_k)$ is the value of the membership function at μ_k, and N is the number of samples of the membership function. After the defuzzified step size μ_D is calculated and the corresponding error is obtained from the TEEL curve, the search block iterates until the absolute error between successive TEEL errors is less than a given tolerance.

4. Experiments and results

The proposed search algorithm is tested on the LMS beamforming and the LRMS-CHIS beamforming system. The tracking capacity of finding an optimal step size for both algorithms is evaluated by simulations. For this purpose, the following signals and parameters are used: a) a linear array of 4 isotropic antennas spaced a quarter wavelength apart; b) a cosine information signal of unitary amplitude and a frequency of 1 Hz arriving at an angle of $\pi/3$; c) a sine interference signal of amplitude of 0.2 and a frequency of 2 Hz arriving at an angle of $\pi/4$; and d) three different signal to interference plus noise ratios (SINR) of 2 dB, –2 dB, and –3 dB. The noise signal at each antenna is white with a normal distribution.

4.1. LMS beamforming

Figure 8 illustrates a 4-LMS beamforming system. A fitted TEEL curve was calculated for each SINR condition. The indicator known as the adjusted R-square (ARS), the degrees of freedom adjusted R-square, was used to determine the best fit. In this method, the R-square is defined as "the square of the correlation between the response values and the predicted response values" [16]. If the value of ARS is closer to one, a better fit is obtained. After performing several fitting tests, it was established that the best fitting curve corresponds to a polynomial of degree six (ARS between 0.98 and 0.99). Figure 9 shows typical TEEL curves for the LMS beamforming system.

According to Figure 9, the TEEL amplitudes increase with electromagnetic noise contamination (increment of the noise variance, σ_η). Furthermore, the maximum step size of convergence changes and tends to decrease for higher SINRs. Fifty tests of convergence were performed to obtain an averaging maximum convergence step size μ. Table 1 summarizes the results.

Table 1. Maximum step size before the LMS algorithm diverges.

SINR	2 dB	–2 dB	–3 dB
μ	0.328	0.200	0.169

In Figure 10, the input and output membership functions for the fuzzy inference system, the adaptive beamforming system, are shown. As expected, the shapes of the membership functions were fitted to Gaussian equations.

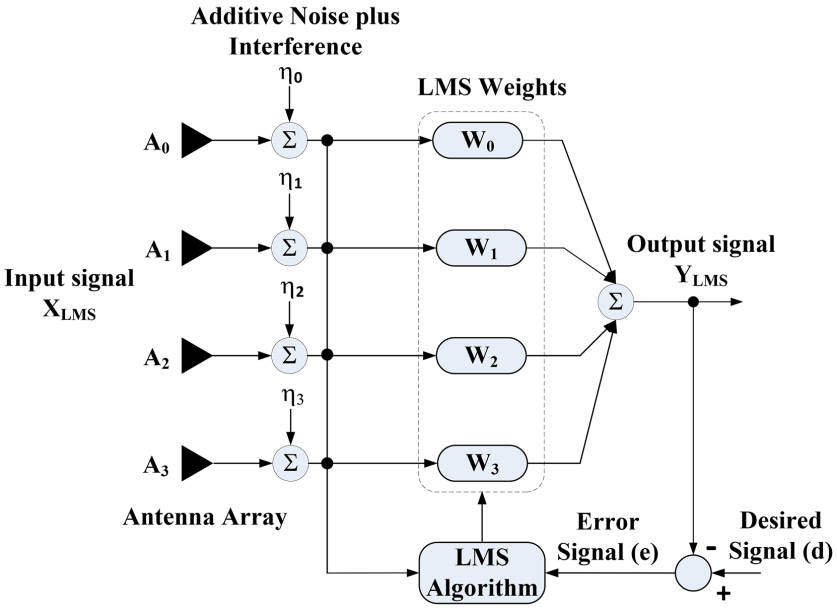

Figure 8. LMS beamforming system.

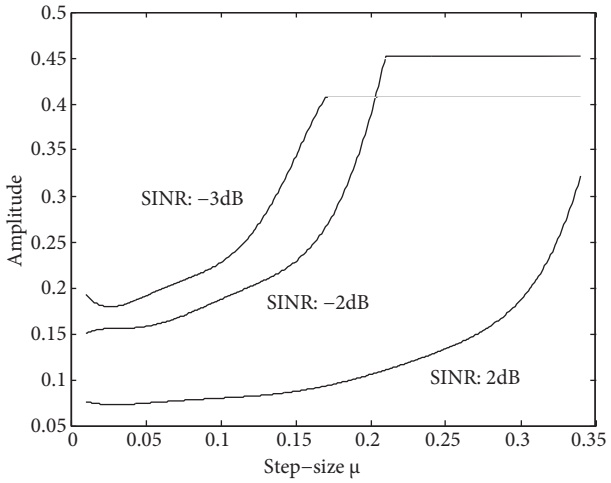

Figure 9. Typical TEEL curves for the LMS beamforming system. SINRs: 2 dB, –2 dB, and –3 dB.

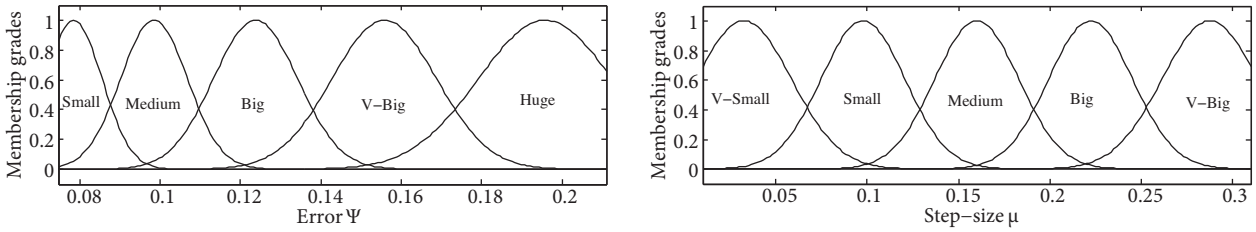

Figure 10. Gaussian membership functions for the input and output fuzzy inference system for the LMS beamforming system. SINR = 2 dB.

The best fitting results were obtained after using a double fitting process. First, both the TEEL amplitude and the truncated μ-interval were fitted using Gaussian models of 3 functions. Although the resulting curve

was good, some cosine waves were observed in the bottom of the fitted membership function curves. Therefore, a first-order Gaussian model was subsequently applied to the first result. In this case, the information obtained from the first fitting process was used to maintain certain characteristics of the original data. With the lower fitting model, the waves were eliminated, and the main characteristics such as the center of the peak and the peak width were maintained. Five functions were selected for the EELC amplitude and the step size μ according to the following fuzzy IF-THEN rules [18]:

Rule 1: IF Ψ is Huge THEN μ is Very Large

Rule 2: IF Ψ is Very big THEN μ is Large

Rule 3: IF Ψ is Big THEN μ is Medium.

Rule 4: IF Ψ is Medium THEN μ is Small.

Rule 5: IF Ψ is Small THEN μ is Very small.

The tracking performance of the proposed algorithm is illustrated in Figures 11 and 12 for a SINR condition of 2 dB. Similar results were observed for the other SINR conditions.

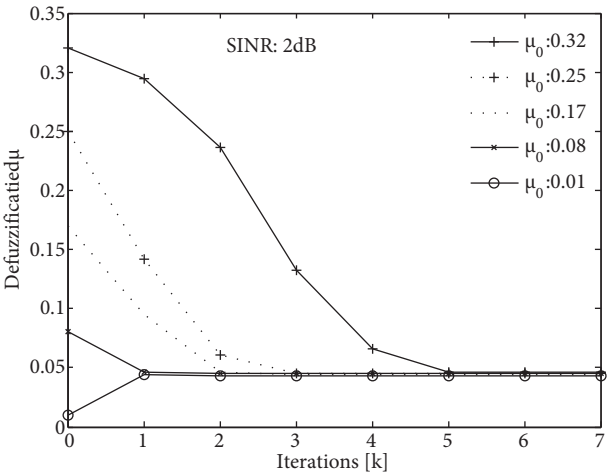

Figure 11. Behavior of the μ-LMS beamforming search algorithm (defuzzified μ vs. iteration). SINR: 2 dB.

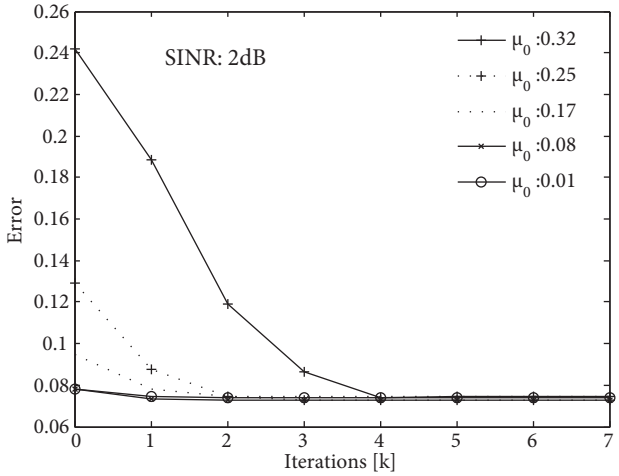

Figure 12. Behavior of the error Ψ_0 due to the step sizes obtained by the μ-LMS beamforming search algorithm in each iteration. SINR: 2 dB.

In Figure 11, the number of iterations required to reach steady state is shown, whereas Figure 12 shows the behavior of the error according to the variations of the corresponding defuzzificated step size μ. Table 2 summarizes the results of the first three tests of a total of twenty. The first column indicates the SINR. The second column contains the initial step sizes used to start the searching algorithm. For each initial condition μ_0, there are two rows. The first row displays the results obtained by the proposed algorithm, while the second row shows the results of a statistical analysis performed on the TEEL curve. Similar results were observed in the remaining tests. Although the experiments were performed under the same SINR, different results were obtained from the statistical analysis due to the random characteristic of the AWGN. In contrast, the proposed searching algorithm shows a more stable behavior; that is, the final defuzzified step size is approximately the same at the end of the iterations. According to Table 2, in most cases, the number of iterations required to stop the algorithm is between 3 and 8. In the case of 2 dB, the percentage of difference between the minimum statistical error and the algorithmic error is a maximum of 4%, even though there is a higher difference between their corresponding step sizes. For –2 dB and –3 dB, the maximum percentages are 6% and 5%, respectively.

This result is considered suitable if we consider that the decimal difference between two errors is located in the thousandth place.

Table 2. Step size and error Ψ obtained by the proposed LMS beamforming search algorithm.

SINR	Initial μ_0	First test			Second test			Third test		
		μ_D	Ψ	Ite	μ_D	Ψ	Ite	μ_D	Ψ	Ite
2 dB	0.32	0.046	0.0741	5	0.045	0.0725	4	0.047	0.0748	7
		0.029	0.0729	–	0.030	0.0701	–	0.064	0.0740	–
	0.25	0.046	0.0752	4	0.045	0.0728	4	0.045	0.0730	4
		0.010	0.0734	–	0.031	0.0716	–	0.370	0.0727	–
	0.17	0.045	0.0750	3	0.041	0.0762	4	0.044	0.0748	3
		0.032	0.0746	–	0.010	0.0747	–	0.052	0.0747	–
–2 dB	0.19	0.032	0.1559	5	0.029	0.1543	8	0.031	0.1593	5
		0.010	0.1504	–	0.180	0.1508	–	0.100	0.1513	–
	0.13	0.032	0.1549	4	0.031	0.1544	4	0.030	0.1523	6
		0.028	0.1546	–	0.027	0.1543	–	0.021	0.1513	–
	0.08	0.032	0.1547	8	0.030	0.1550	4	0.031	0.1599	4
		0.020	0.1507	–	0.017	0.1516	–	0.010	0.1513	–
–3 dB	0.15	0.028	0.1793	6	0.026	0.1787	7	0.027	0.1837	6
		0.027	0.1793	–	0.019	0.1701	–	0.010	0.1755	–
	0.11	0.027	0.1826	4	0.027	0.1839	6	0.027	0.1797	6
		0.010	0.1756	–	0.010	0.1746	–	0.140	0.1739	–
	0.08	0.027	0.1758	5	0.028	0.1809	6	0.026	0.1841	6
		0.022	0.1750	–	0.010	0.1776	–	0.010	0.1756	–

4.2. LRMS-CHIS beamforming

Figure 13 shows the basic scheme of a new adaptive hybrid filter for beamforming systems. According to the diagram, the "least recursive mean square-cascade hybrid independent structure" (LRMS-CHIS) consists of two independent LMS and RLS adaptive algorithms in cascade form. As a second application, the proposed search algorithm has been incorporated into the LMS pre-filter to evaluate its performance.

During the learning and searching process, the switch (a) is located in position 1, while the other switches (b and c) are open. In this operation mode, the error signal and the signals from the antenna array feed the LMS μ-searching algorithm. At the same time, the RLS and LMS adaptive filters do not process any data. After the search algorithm finds the optimal step size μ, the switch (a) changes to position 2, and the switches (b) and (c) are closed. The optimal μ_D is loaded into the single LMS algorithm, the switch (c) returns to its open position, and the single LMS filter continues to work with this step size until the SINR conditions or input signals change. Finally, the complete LRMS-CHIS beamforming system works in this mode. Table 3 summarizes the results of 3 tests performed for the hybrid scheme. Similar results were obtained for the other 7 tests. The first row shows the SINR ratio. The first column contains the number of tests, as in Table 2. There are 2 rows for each test in Table 3. The first row displays the results obtained by the proposed algorithm, whereas the second row shows the results of a statistical analysis.

For the condition of 2 dB, the percentage of difference between the statistical error and the algorithmic result has an approximate maximum of 2%. For –2 dB and –3 dB, the maximum percentage is approximately 3% and 4%, respectively. On average, the number of iterations required for this application was between 4 and 8. Figure 14 shows the behavior of the error versus the variations in the step size μ for a SINR of –2 dB as an example.

In the previous figure, the value of 0.032 is indicated on the plot to illustrate that the solution can be considered a good approximation. Figure 15 shows the MSE for the hybrid cascade systems. According to the results, a good learning curve is obtained for the step size of 0.032.

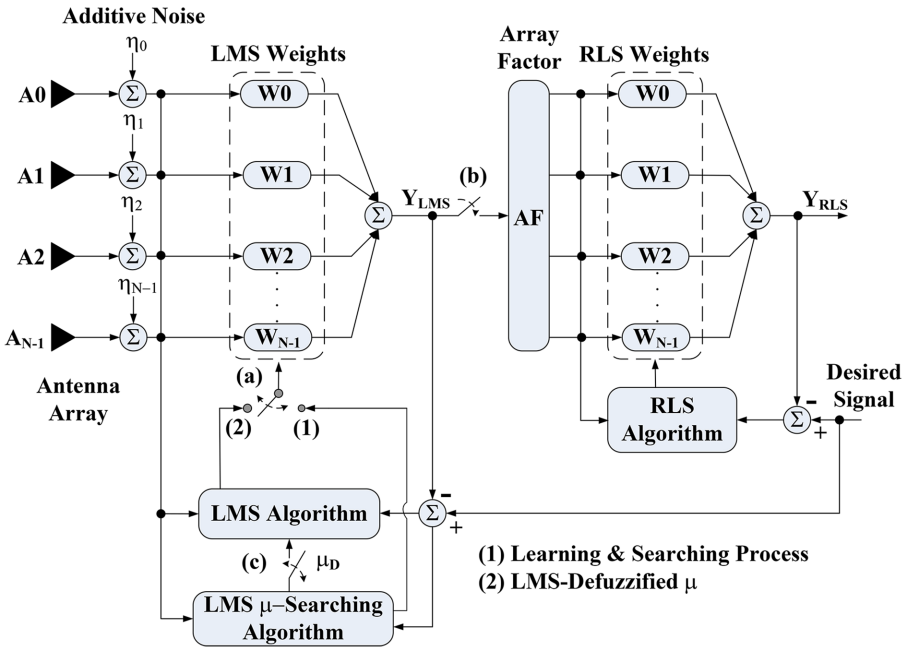

Figure 13. LRMS-CHIS beamforming system.

Figure 16 shows the recovered signals after applying the optimal step size to the LMS pre-filter stage of the hybrid scheme. The real component of the recovered signal is similar to the desired one. In contrast, the imaginary component presents a residual random behavior (the transmitted imaginary component is zero).

Table 3. Step size and error Ψ obtained by the proposed μ-search LMS algorithm for the LRMS-CHIS beamforming system.

SINR	2 dB			–2 dB			–3 dB		
Test no.	μ_D	Ψ	% Dif	μ_D	Ψ	% Dif	μ_D	Ψ	% Dif
1	0.045	0.0752	1.46%	0.032	0.1565	1.47%	0.028	0.1821	2.31%
	0.047	0.0741		0.010	0.1542		0.010	0.1779	
2	0.040	0.0748	0.67%	0.032	0.1568	2.10%	0.026	0.1835	3.05%
	0.045	0.0743		0.029	0.1535		0.015	0.1779	
3	0.046	0.0758	1.45%	0.032	0.1552	0.39%	0.028	0.1800	1.06%
	0.030	0.0747		0.020	0.1546		0.010	0.1781	

Figure 17 shows the recovered signal at the output of the adaptive beamformer, the RLS-filter. The real component is similar to the one obtained from the previous pre-filter stage; however, the imaginary component has been virtually eliminated. This behavior is a main characteristic of this filter. Two radiation beam patterns for the 4-LRMS-CHIS beamforming systems are shown in Figure 18 for a SINR of –2 dB. Similar beam patterns were obtained from the other ratios. An improved beam pattern is generated by using the step size recommended by the searching algorithm. In addition, better directivity and a lower half-power beam width of the main lobe are achieved.

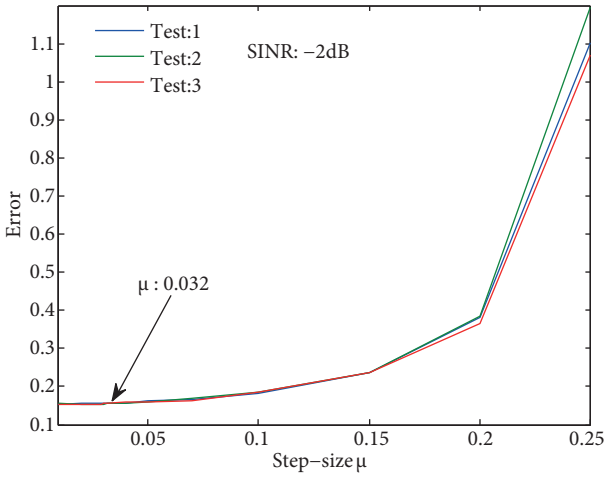

Figure 14. TEEL curves for the LRMS-CHIS beamforming system, $\mu \in [0.01, 0.25]$, SINR: –2 dB.

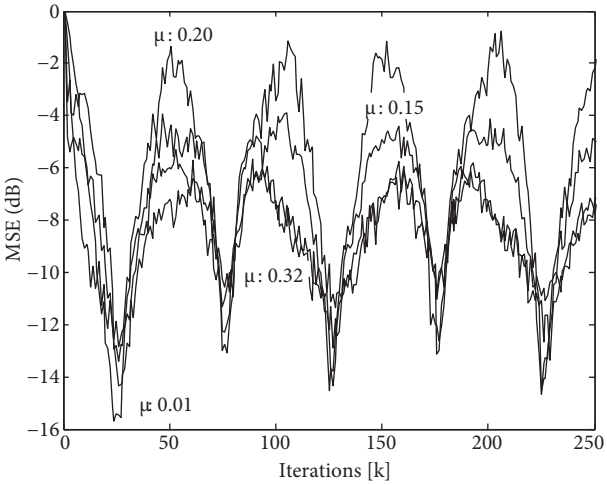

Figure 15. Mean square errors for the LRMS-CHIS beamforming system. SINR = –2 dB.

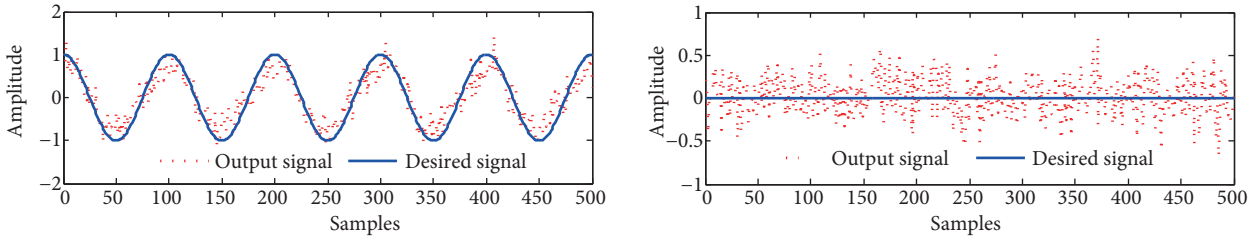

Figure 16. Recovered signal at the output of the LMS filter. $\mu = 0.032$. SINR = –2 dB.

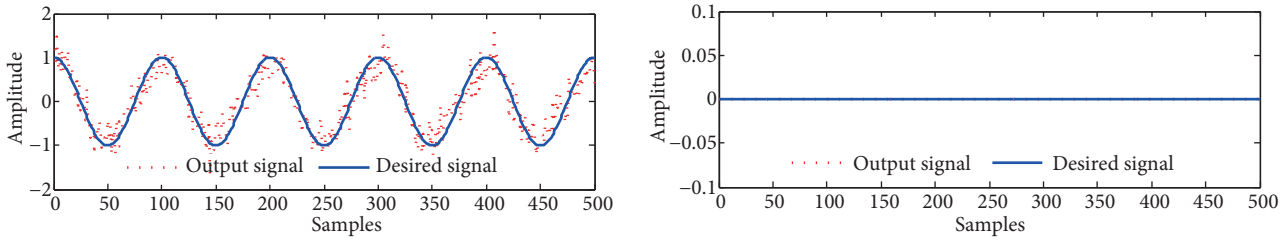

Figure 17. Recovered signal at the output of the LRMS-CHIS beamforming system. $\mu = 0.032$. SINR = –2 dB.

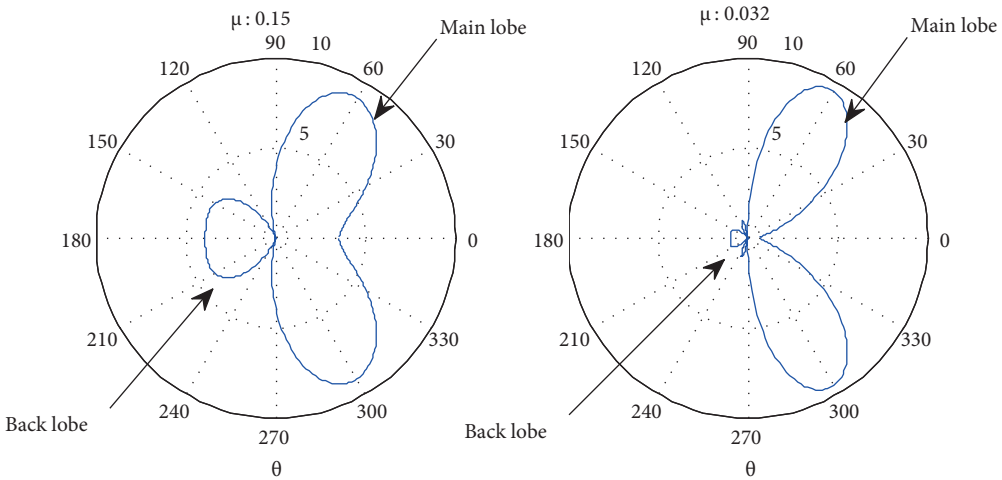

Figure 18. Beam patterns for $\mu = 0.15$ and the suggested $\mu = 0.032$. SINR = –2 dB.

5. Conclusion

In this work, a novel search algorithm to find the optimal step size for LMS adaptive filters was presented. The simulation results show that the obtained step size is quite close to the statistically optimal μ. The main advantage of the proposed algorithm is that it is not necessary to know the input signals to find the optimal step size. The automatic generation of membership functions using fuzzy classification and neural networks provides an adaptive fuzzy inference system whose membership functions change according to the SINR. In this sense, a divergence analysis was incorporated to automatically limit the amount of data used to generate the TEEL curves, their corresponding step-size ranges, and the fuzzy levels in the membership functions; therefore, the fuzzy levels are also automatically adjusted according to the fuzzy rules and the provided step-size range. As a result, the inference quality is improved, and the FIS design time is reduced. In addition, the steady-state value of the cost function is minimized. Finally, as part of the beamforming system, the proposed algorithm improves the directivity of the adaptive linear array beam patterns.

Acknowledgments

The authors thank the National Science and Technology Council of Mexico and the National Polytechnic Institute of Mexico for the support provided during the realization of this research.

References

[1] Liu W, Weiss S. Wideband Beamforming Concepts and Techniques. 1st ed. Chichester, UK: Wiley, 2010.

[2] Mozingo RA, Haupt RL, Miller TW. Introduction to Adaptive Arrays. 2nd ed. Raleigh, NC, USA: SciTech Publishing, 2011.

[3] Haykin S. Radar Array Processing for Angle of Arrival Estimation. In: Haykin S, editor. Array Signal Processing. Englewood Cliffs, NJ, USA: Prentice-Hall, 1985. pp. 194-292.

[4] Diniz P. Adaptive Filtering Algorithms and Practical Implementation. 4th ed. New York, NY, USA: Springer, 2013.

[5] Perez-Meana HM. Transversal Adaptive Filters. In: Kravchenko VF, editor. Adaptive Digital Signal Processing of Multidimensional Signals with Applications. Moscow, Russia: FizMatLit, 2009. pp. 238-260.

[6] Hänsler E, Schmidt G. Acoustic Echo and Noise Control: A Practical Approach. 1st ed. Hoboken, NJ, USA: Wiley, 2004.

[7] Ross JT, Fuzzy Logic with Engineering Applications. 3rd ed. Chichester, UK: Wiley, 2010.

[8] Orozco-Tupacyupanqui W, Nakano-Miyatake M, Pérez-Meana H. Cascade Hybrid LRMS Adaptive Configurations for Linear Array Beamforming. In: WSEAS 13th International Conference on Applied Computer Science; 23–25 April 2013; Morioka City, Japan. Morioka City, Japan: WSEAS Press. pp. 66-71.

[9] Widrow B, Mantey PE, Griffiths LJ, Goode BB. Adaptive antenna systems. IEEE Proc 1967; 55: 2143-2159.

[10] Sayed AH. Adaptive Filters. 1st ed. Hoboken, NJ, USA: Wiley, 2008.

[11] Ostertagová E, Ostertag O. The Simple Exponential Smoothing Model. In: The 4th International Conference Modelling of the Mechanical and Mechatronic Systems; 20–22 September 2011; Herl'any, Slovak Republic. Košice, Slovak Republic: TUK Press. pp. 380-384.

[12] Widrow B, Stearns SD. Adaptive Signal Processing. 1st ed. Upper Saddle River, NJ, USA: Prentice-Hall, 1985.

[13] Papoulis A, Unnikrishna-Pillai S. Probability Random Variables and Stochastic Processes. 4th ed. New York, NY, USA: McGraw-Hill, 2002.

[14] Haykin S. Adaptive Filter Theory. 3rd ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 1986.

[15] Chapra SC, Canale RP. Numerical Methods for Engineers. 6th ed. New York, NY, USA: McGraw-Hill, 2010.

[16] The MathWorks. Curve Fitting Toolbox^{TM} User's Guide 2014a. Version 3.4.1. Natick, MA, USA: The MathWorks, 2014.

[17] Mamdani EH. Applications of Fuzzy Logic to Approximate Reasoning using Linguistic Synthesis. IEEE T Comput 1977; 26: 1182-1191.

[18] Orozco-Tupacyupanqui W, Pérez-Meana H, Nakano-Miyatake M. A New Fuzzy Logic Application to Search the Optimal Step Size for NLMS Beamforming. In: 37th International Conference on Telecommunications and Signal Processing; 1–3 July 2014; Berlin, Germany. Brno, Czech Republic: BUT Press. pp. 568-573.