Int. j. inf. tecnol. (August 2022) 14(5):2245–2253
https://doi.org/10.1007/s41870-022-00985-y

ORIGINAL RESEARCH

Analysis of convexly combined recursive inverse algorithms

Hasan Abu Hilal[1] · **Mohammad Salman**[2] · **Alaa Hameed**[3]

Received: 1 March 2022 / Accepted: 30 April 2022 / Published online: 1 June 2022

Abstract The RI method, which has only lately been developed, has showed outstanding performance in a wide range of application fields. Convexly linked RI method In this study, we demonstrate the algorithm's tracking evaluation as well as the findings obtained. It is also possible to create a formula for the steady state based on the mean square error (MSE). In this case, it is proposed that a 2-dimensional (2D) form of the technique be used. Comparisons have been made between theoretical and experimental results, and it has been discovered that they are in agreement. Also included in this study are comparisons of this method with that of NLMS in both additive white Gaussian noise and additive correlated Gaussian noise settings, both of which are used in a noise canceling situation. According to the results of the simulations, the proposed approach outperforms the NLMS algorithm is based on MSE as well as the frequency at which it converges. The 2D form of the approach, on the other hand, exhibited excellent achievement in the images de-noising.

✉ Hasan Abu Hilal
hasan.abuhilal@hct.ac.ae; Hasanabuhilal@hotmail.com

Mohammad Salman
mohammad.salman@aum.edu.kw

Alaa Hameed
aalihameed@aydin.edu.tr

[1] Higher Colleges of Engineering and Technology, HCT, Dubai, UAE

[2] College of Engineering and Technology, American University of the Middle East, Kuwait, Kuwait

[3] Department of Computer Engineering, Istanbul Sabahattin Zaim University, Istanbul, Turkey

Keywords Convex combination · Noise cancelation · NLMS Algorithm · Tracking analysis

1 Introduction

Filtering raw data may be defined as performing standard processing operations on series of data in order to remove the unnecessary information. By utilizing well-defined features, it is possible to create a digital filter with set coefficients. A filter that updates the coefficients with time is necessary in some scenarios, such as when the specifications are not accessible or are time variables. This type of filter, known as an adaptive filter, is used in these instances. The field of adaptive filtering has attracted attention of a large group of scholars during the last few decades. This interest can be attributed to the numerous application areas for adaptive filters [1–7]. Least mean square (LMS) algorithm is a well-known adaptive filtering method that has been used successfully to solve system identification difficulties for a long period of time. However, the effectiveness of an adaptive filter is often governed by a number of parameters that provide a trade-off between the rates at which the adaptive filter converges and the mean-square error of the adaptive filter in steady state (SS-MSE). Given that the LMS method is a gradient descent-based technique, the filter coefficients are updated using a constant step-size parameter. When it comes to the performance of the algorithm, this step-size parameter is crucial to consider. A big step-size results in a rapid convergence but a high mean square error (MSE), whereas a small step-size value results in sluggish convergence but a low MSE.

By employing a variable step-size [8–12], this trade-off may be adjusted in favor of both an improvement in the

convergence rate and a decrease in misadjustment for the optimal performance.

The field of adaptive filtering has produced a number of variable step-size LMS-type algorithms in the attempt to solve this trade-off. A novel recursive inverse (RI) adaptive algorithm [13] has been presented by Ahmad et al. When compared to other adaptive algorithms with lower or equivalent computational complexity [14], this approach has demonstrated excellent performance. Several studies [15, 16] have demonstrated that it performs well in impulsive non Gaussian noise and non-stationary situations. On the other hand, the trade-off seen between convergence rate and the MSE, is not completely resolved. Several convexly combined adaptive filters have been presented [17–24] in order to solve this trade-off. The underlying concept behind these ideas is to extract the quick convergence and low MSE qualities of two adaptive filters and integrate their performances in a single algorithm. However, the majority of these approaches still result in an MSE that is relatively high.

The main contribution of this paper is to offer a novel convexly combined RI adaptive filtering algorithm that, when compared to the previous solutions, has extremely high performance through means of convergence rate and SS-MSE. Also, is to provide a detailed theoretical tracking analysis of the proposed method. In addition, the derived theoretical results will be compared to the experimental ones to show how much they are in match. Furthermore, A 2-dimensional (2D) version of the method, will be presented in this paper and its performance in image denoising will be examined.

The structure of this paper is as follows. The proposed convexly combined RI algorithm is provided in Sect. 2. In Sect. 3, the tracking analysis of the algorithm is presented, and a SS-MSE expression is derived from the results of the study. In Sect. 4, a 2D convexly combined RI method is presented. Section 5 presents the findings of the simulations, and Sect. 6 presents the conclusions reached as a result of the simulation and the future work.

2 The proposed algorithm

Figure 1 depicts the mixture of two adaptive filters under noise reduction setting, which has been presented in [20].

Starting with the update equation of the RI algorithm [13],

$$\mathbf{w}_i(n) = [\mathbf{I} - \mu_i(n)\mathbf{R}_i(n)]\mathbf{w}_i(n-1) + \mu_i(n)\mathbf{P}_i(n) \tag{1}$$

where n denotes the index of time $(n = 1, 2, \ldots)$, $\mathbf{w}_i(n)$ is the ith filter weight array with length N at time n. N (i = 1, 2), $\mathbf{I}$ is an $N \times N$ identity matrix, $\mu_i(n)$ denotes the ith variable step-size, $\mathbf{R}_i(n)$ stands for the ith autocorrelation

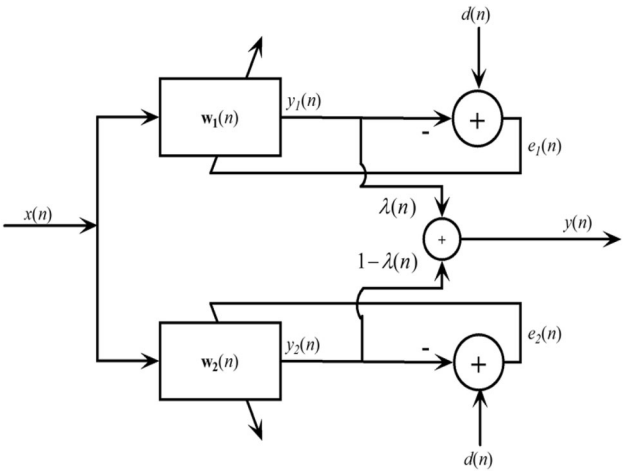

Fig. 1 Convex combination of two adaptive filters for a noise cancelation setting

matrix elements that has the tap-input vector and $\mathbf{P}_i(n)$ holds the value of the ith cross-correlation vector between the tap-input vector and desired response, $\mathbf{x}(n)$ is the distance between the tap-input vector $\mathbf{x}(n)$ and the desired response vector $d(n)$ of an adaptive filter. According to the following, the correlations are evaluated recursively:

$$\mathbf{R}_i(n) = \beta\mathbf{R}_i(n-1) + \mathbf{x}(n)\mathbf{x}^T(n) \tag{2}$$

$$\mathbf{P}_i(n) = \beta\mathbf{P}_i(n-1) + d(n)\mathbf{x}(n) \tag{3}$$

The forgetting factor, β, has a value that is very close to one.

$$\mu_i(n) = \frac{\mu_0}{1-\beta^n} \tag{4}$$

where $\mu_0 < \mu_{\max}$, $\mu_{\max} < \frac{2}{\lambda_{\max}(\mathbf{R}_i(n))}$ and $\lambda_{\max}(\mathbf{R}_i(n))$ is the highest eigenvalue of $\mathbf{R}_i(n)$.

The error of each separate filter is denoted by the notation

$$e_i(n) = d(n) - \mathbf{w}_i^T(n-1)x(n) \tag{5}$$

and the vector of the desired response can be written as:

$$d(n) = \mathbf{w}_0^T x(n) + v(n) \tag{6}$$

where w_0 represents the optimum weighted values vector and $v(n)$ represents the measurement noise. The outputs of both of the adaptive filters can be merged as per [23], by using the following expression,

$$y(n) = \lambda(n)y_1(n) + [1-\lambda(n)]y_2(n), \tag{7}$$

where $y_i(n) = \mathbf{w}_i^T(\mathrm{n}-1)\mathbf{x}(n)$ and convex mixture constraint, $\lambda(n)$ is specified by

$$\lambda(n) = \frac{E[(d(n) - y_2(n))(y_1(n) - y_2(n))]}{E[(y_1(n) - y_2(n))^2]}. \tag{8}$$

where the error signal generated by the previously mentioned combination is:

$$e(n) = d(n) - y(n) = d(n) - \lambda(n)y_1(n) - (1 - \lambda(n))y_2(n) \tag{9}$$

3 Tracking analysis

This section describes the tracking details of the proposed method, as well as how the SS-MSE criteria can be determined from the results of the algorithm. Let's begin with the random walk model, which is simple and straightforward.

$$\mathbf{w}_0(n) = \mathbf{w}_0(n-1) + \mathbf{q}(n) \tag{10}$$

where $\mathbf{q}(n)$ is a zero mean independent and identically distributed (i.i.d.) random process with covariance matrix $\mathbf{Q} = E\{\mathbf{q}(n)\mathbf{q}(n)\}$. The following is the definition of the weight error signal of the ith filter:

$$\underset{\mathbf{w}_i}{\sim}(n) = \mathbf{w}_0(n) - \mathbf{w}_i(n) \tag{11}$$

The apriori error and the a posteriori error are described as,

$$e_{i,a}(n) = \mathbf{x}^T(n)[\mathbf{w}_0(n) - \mathbf{w}_i(n-1)] \tag{12}$$

$$e_{i,p}(n) = \mathbf{x}^T(n)[\mathbf{w}_0(n) - \mathbf{w}_i(n)] \tag{13}$$

As a result of subtracting the total output of filters in (7) from the required response, as well as by applying (12) and (13), we can determine the overall output error (13),

$$\begin{aligned} e_a(n) &= d(n) - \lambda(n)y_1(n) - (1 - \lambda(n))y_2(n) \\ &= d(n) - \lambda(n)y_1(n) - y_2(n) - \lambda(n)y_2(n) \\ &= e_{2,a}(n) - \lambda(n)[-d(n) + y_1(n) + d(n) - y_2(n)] \\ &= e_{2,a}(n) - \lambda(n)[-e_{1,a}(n) + e_{2,a}(n)] \\ &= e_{2,a}(n) + \lambda(n)e_{1,a}(n) - \lambda(n)e_{2,a}(n) \\ &= (1 - \lambda(n))e_{2,a}(n) + \lambda(n)e_{1,a}(n) \end{aligned} \tag{14}$$

Evaluate $E\left[e_a^2(n)\right]$ using (14),

$$\begin{aligned} E[e_a^2(n)] &= E[((1 - \lambda(n))e_{2,a}(n) \\ &\quad + \lambda(n)e_{1,a}(n))((1 - \lambda(n))e_{2,a}(n) \\ &\quad + \lambda(n)e_{1,a}(n))] \\ &= (1 - \lambda(n))^2 E\left[e_{2,a}^2(n)\right] \\ &\quad + 2\lambda(n)(1 - \lambda(n))E\left[e_{1,a}(n)e_{2,a}(n)\right] \\ &\quad + \lambda^2(n)E\left[e_{2,a}^2(n)\right] \end{aligned} \tag{15}$$

In order to evaluate $E[e_a^2(n)]$, first we must evaluate the cross terms in (15),

$$\begin{aligned} E\left[e_{1,a}(n)e_{2,a}(n)\right] &= E[(\mathbf{w}_0(n) - \mathbf{w}_1(n-1))^T\mathbf{x}(n)\mathbf{x}^T(n)(\mathbf{w}_0(n) \\ &\quad - \mathbf{w}_2(n-1))] \end{aligned} \tag{16}$$

Subtracting both sides of (1) from $\mathbf{w}_0(n)$ and manipulating, then using (12) and (13) we get

$$\underset{\mathbf{w}_i}{\sim}(n) = \underset{\mathbf{w}_i}{\sim}(n-1) - \mu_i(n)\mathbf{x}(n)e_i(n) + \mu_i(n)\beta_i\xi_i(n-1), \tag{17}$$

where $\xi_i(n-1) = \mathbf{R}(n)\mathbf{w}_i(n-1) - \mathbf{p}(n)$. Multiplying both sides of (17) by $\mathbf{x}^T(n)$ guides to,

$$\begin{aligned} e_{i,p}(n) &= e_{i,a}(n) - \mu_i(n)\mathbf{x}^T(n)\mathbf{x}(n)e_i(n) \\ &\quad + \mu_i(n)\beta_i\mathbf{x}^T(n)\xi_i(n-1) \end{aligned} \tag{18}$$

Substituting (18) in (17) yields

$$\begin{aligned} \underset{\mathbf{w}_i}{\sim}(n) &= \underset{\mathbf{w}_i}{\sim}(n-1) - \mu_i(n)\beta_i\xi_i(n-1) \\ &\quad + \frac{\mu_i(n)\mathbf{x}(n)}{\mu_i(n)\mathbf{x}^T(n)\mathbf{x}(n)}\left[e_{i,a}(n) - e_{i,p}(n)\right. \\ &\quad \left. + \mu_i(n)\beta_i\mathbf{x}^T(n)\xi_i(n-1)\right] \end{aligned} \tag{19}$$

Note that

$$tr\{\mathbf{x}(n)\mathbf{x}^T(n)\xi_i(n-1)\} = tr\{\xi_i(n-1)\mathbf{x}^T(n)\mathbf{x}(n)\}$$

[2] and, hence,

$$\mathbf{x}(n)\mathbf{x}^T(n)\xi_i(n-1) \approx \xi_i(n-1)\mathbf{x}^T(n)\mathbf{x}(n)$$

. Reorganizing (19) delivers

$$\underset{\mathbf{w}_i}{\sim}(n) - \frac{\mathbf{x}(n)}{\mathbf{x}^T(n)\mathbf{x}(n)}e_{i,a}(n) = \underset{\mathbf{w}_i}{\sim}(n-1) - \frac{\mathbf{x}(n)}{\mathbf{x}^T(n)\mathbf{x}(n)}e_{i,p}(n) \tag{20}$$

The result of multiplying both sides of the first filter in (20) by their counterparts in the second filter is as follows:

$$\underset{\mathbf{w}_1}{\overset{\sim}{}}^{T}(n)\underset{\mathbf{w}_2}{\overset{\sim}{}}(n)+\frac{e_{1,a}(n)e_{2,a}(n)}{\mathbf{x}^T(n)\mathbf{x}(n)}=\underset{\mathbf{w}_1}{\overset{\sim}{}}^{T}(n-1)\underset{\mathbf{w}_2}{\overset{\sim}{}}(n-1)+\frac{e_{1,p}(n)e_{2,p}(n)}{\mathbf{x}^T(n)\mathbf{x}(n)} \tag{21}$$

The random walk model is used to construct the given formula for the average of the inner product of the weighted error vectors of the two separate filters at the time instant n, which is written below as,

$$\begin{aligned}&E[(\mathbf{w}_0(n)-\mathbf{w}_1(n-1))^T(\mathbf{w}_0(n)-\mathbf{w}_2(n-1))]\\&=E\big[\mathbf{w}_0(n-1)+\mathbf{q}(n)-\mathbf{w}_1(n-1))^T(\mathbf{w}_0(n-1)\\&\quad+\mathbf{q}(n)-\mathbf{w}_2(n-1))]\\&=E\left[\underset{\mathbf{w}_1}{\overset{\sim}{}}(n-1)+\mathbf{q}(n))^T\left(\underset{\mathbf{w}_2}{\overset{\sim}{}}(n-1)+\mathbf{q}(n)\right)\right]\\&=E\left[\underset{\mathbf{w}_1}{\overset{\sim}{}}^{T}(n-1)\underset{\mathbf{w}_2}{\overset{\sim}{}}(n-1)\right]+E\left[\underset{\mathbf{w}_1}{\overset{\sim}{}}^{T}(n-1)\mathbf{q}(n)\right]\\&\quad+E\left[\mathbf{q}^T(n)\underset{\mathbf{w}_2}{\overset{\sim}{}}(n-1)\right]+E[\mathbf{q}(n)\mathbf{q}(n)]+Tr\{\mathbf{Q}\}\end{aligned} \tag{22}$$

Substituting (22) into (21) and shortening provides,

$$\begin{aligned}&E[\underset{\mathbf{w}_1}{\overset{\sim}{}}^{T}(n)\underset{\mathbf{w}_2}{\overset{\sim}{}}(n)]+E\left[\frac{e_{1,a}(n)e_{2,a}(n)}{x^T(n)x(n)}\right]\\&=E\left[\underset{\mathbf{w}_1}{\overset{\sim}{}}^{T}(n-1)\underset{\mathbf{w}_2}{\overset{\sim}{}}(n-1)\right]+E\left[\frac{e_{1,p}(n)e_{2,p}(n)}{\mathbf{x}^T(n)\mathbf{x}(n)}\right]\\&\quad+Tr\{\mathbf{Q}\}\end{aligned} \tag{23}$$

in the SS,

$$E[\underset{\mathbf{w}_1}{\overset{\sim}{}}^{T}(n)\underset{\mathbf{w}_2}{\overset{\sim}{}}(n)\approx E[\underset{\mathbf{w}_1}{\overset{\sim}{}}^{T}(n-1)\underset{\mathbf{w}_2}{\overset{\sim}{}}(n-1)] \tag{24}$$

and hence,

$$E\left[\frac{e_{1,a}(n)e_{2,a}(n)}{x^T(n)x(n)}\right]=E\left[\frac{e_{1,p}(n)e_{2,p}(n)}{\mathbf{x}^T(n)\mathbf{x}(n)}\right]+Tr\{\mathbf{Q}\} \tag{25}$$

Now, if we substitute (18) in (25) and rearrange we find,

$$\begin{aligned}&E\big[e_{1,a}(n)\mu_2(n)e_2(n)\big]+E\big[e_{2,a}(n)\mu_1(n)e_1(n)\big]\\&=E\big[\mu_1(n)\mu_2(n)x^T(n)x(n)e_1(n)e_2(n)\big]+Tr\{Q\}\end{aligned} \tag{26}$$

Now substitute $e_i(n)=e_{i,a}(n)+v(n)$ in (26), putting in mind that $x^T(n)x(n)\approx E\{x^T(n)x(n)\}=\sigma_x^2$ and $E[e_{i,a}(n)v(n)]=0$ and simplifying,

$$\begin{aligned}&\mu_2(n)E\big[e_{1,a}(n)(e_{2,a}(n)+v(n))\big]\\&\quad+\mu_1(n)E\big[e_{2,a}(n)(e_{1,a}(n)+v(n))\big]\\&=\mu_1(n)\mu_2(n)E\big[x^T(n)x(n)(e_{1,a}(n)+v(n))(e_{2,a}(n)+v(n))\big]\\&\quad+Tr\{Q\}\end{aligned} \tag{27}$$

substituting (27) in (16) and taking $\lim_{n\to\infty}$ gives,

$$\begin{aligned}&\lim_{n\to\infty}E\big[e_{1,a}(n)e_{2,a}(n)\big]\\&\quad=z(n)\big[\mu_1(n)\mu_2(n)\sigma_v^2E\big[x^T(n)x(n)\big]+Tr\{Q\}\big]\end{aligned} \tag{28}$$

where

$$z(n)=\frac{1}{\mu_1(n)+\mu_2(n)-\mu_1(n)\mu_2(n)E[x^T(n)x(n)]}$$

For a single filter case, we have

$$\begin{aligned}&\lim_{n\to\infty}E\big[e_{i,a}(n)e_{i,a}(n)\big]\\&=\frac{1}{2\mu_i(n)-\mu_i^2(n)E[x^T(n)x(n)]}[\mu_i^2(n)\sigma_v^2E\big[x^T(n)x(n)\big]+Tr\{\mathbf{Q}\}]\end{aligned} \tag{29}$$

Substituting (28) and (29) in (15) gives

$$\begin{aligned}&\lim_{n\to\infty}E\big[e_a^2(n)\big]\\&=\frac{2\lambda(\infty)(1-\lambda(\infty))}{\mu_1(n)+\mu_2(n)-\mu_1(n)\mu_2(n)E[x^T(n)x(n)]}\\&\quad\times\big[\mu_1(n)\mu_2(n)\sigma_v^2E\big[x^T(n)x(n)\big]+Tr\{Q\}\big]\\&\quad+\frac{\lambda^2(\infty)}{2\mu_1(n)-\mu_1^2(n)E[x^T(n)x(n)]}\big[\mu_1^2(n)\sigma_v^2E\big[x^T(n)x(n)\big]\\&\quad+Tr\{Q\}]+\frac{(1-\lambda(\infty))^2}{2\mu_2(n)-\mu_2^2(n)E[x^T(n)x(n)]}\big[\mu_2^2(n)\sigma_v^2E\big[x^T(n)x(n)\big]\\&\quad+Tr\{Q\}]\end{aligned} \tag{30}$$

4 Extending to 2D convex recursive inverse algorithms

If we combine two RI algorithms such as the one shown below, you can expand the update equation in (1) into a convex 2D form by following the same concept in [25],

$$\mathbf{w}_{i,n}(k_1,k_2)=\big[\mathbf{I}-\mu_i(n)\mathbf{R}_{i,n}\big]\mathbf{w}_{i,n-1}(k_1,k_2)+\mu_i(n)\mathbf{P}_{i,n} \tag{31}$$

where $\mathbf{w}_{i,n}(k_1,k_2)$ is the stretched weight vector of 2D form with size $N\times N$ and at time n, $i=1,2$, $k_1=0,1,\ldots,N-1$ and $k_2=0,1,\ldots,N-1$. The autocorrelation matrix $\mathbf{R}_{i,n}$ and cross-correlation vector $\mathbf{P}_{i,n}$ are specified as

$$\mathbf{R}_{i,n}=\beta\mathbf{R}_{i,n-1}+\mathbf{x}(m_1,m_2)\mathbf{x}^T(m_1,m_2) \tag{32}$$

$$-\mathbf{P}_{i,n}=\beta\mathbf{P}_{i,n-1}+d(m_1,m_2)\mathbf{x}(m_1,m_2) \tag{33}$$

where $\mathbf{x}(m_1,m_2)$ is the filter contribution input and $d(m_1,m_2)$ is the looked-for output. The column-ordered vectors planned the filter input $\mathbf{x}(m_1,m_2)$ and tap-weight vector $\mathbf{w}_{i,n}(k_1,k_2)$ in 2-D form for instance, and

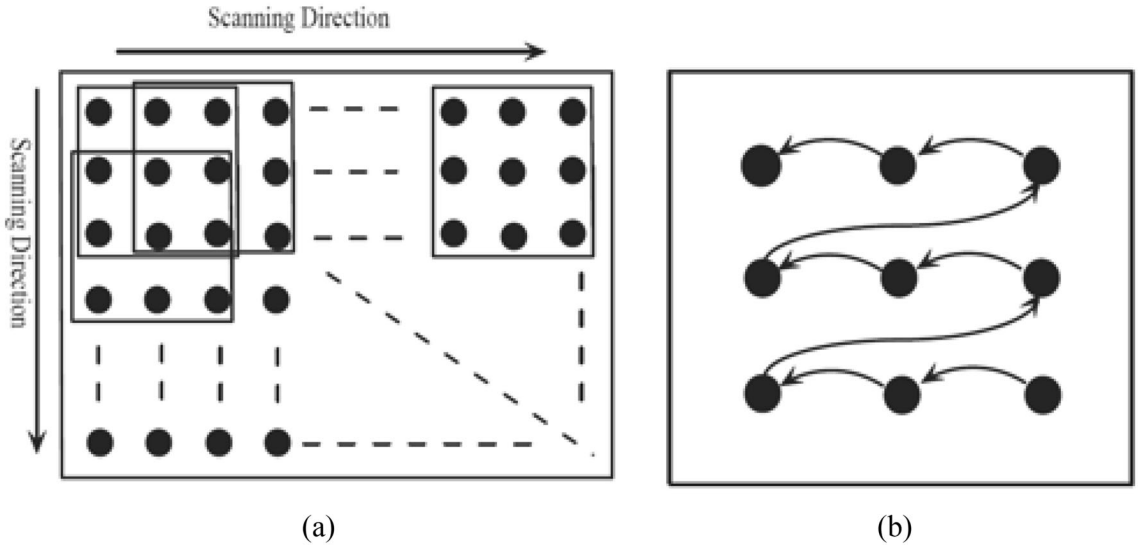

Fig. 2 Rectangular configuration of data-reusing in 2D plane

$$\mathbf{x}(m_1, m_2) = \begin{bmatrix} x(m_1, m_2) \\ \vdots \\ x(m_1, m_2 - N + 1) \\ \vdots \\ x(m_1 - N + 1, m_2) \\ \vdots \\ x(m_1 - N + 1, m_2 - N + 1) \end{bmatrix} \tag{34}$$

$$\mathbf{w}_{i,n}(k_1, k_2) = \begin{bmatrix} \mathbf{w}_{i,n}(0, 0) \\ \vdots \\ \mathbf{w}_{i,n}(0, N - 1) \\ \vdots \\ \mathbf{w}_{i,n}(N - 1, 0) \\ \vdots \\ \mathbf{w}_{i,n}(N - 1, N - 1) \end{bmatrix} \tag{35}$$

Figure 2 depicts one possible method of data reusing [25]. As illustrated in Fig. 2a, we can examine a mask, consisting of 3×3 pixels, which travels horizontally to the right by a single column at a time till the end of each row until the end of the mask. Then, starting with the following row below, the process is repeated until the conclusion of the last row and column of the image is reached. The data can be reshaped at the end of each operation of the mask, as illustrated in Fig. 2b, beginning with the last pixel in the lower right corner and moving left and up. The output of the ith filter is determined by the following 2D convolution:

$$y_{i,n}(m_1, m_2) = \sum_{k_1=0}^{N-1} \sum_{k_2=0}^{N-1} w_{i,n}(k_1, k_2) x(m_1 - k_1, m_2 - k_2) \tag{36}$$

The output of the 2D convexly combined algorithms can be computed as [23],

$$y_n(m_1, m_2) = \lambda(m_1, m_2) y_{1,n}(m_1, m_2) + [1 - \lambda(m_1, m_2)] y_{2,n}(m_1, m_2) \tag{37}$$

5 Experimental results

Here, the effectiveness of the presented algorithm is compared to the theoretical SS-MSE given in (30), as well as to the performance of the convexly combined normalized LMS (NLMS) method in additive white Gaussian noise (AWGN) as well as additive correlated Gaussian noise (ACGN) contexts for the noise cancelation setting illustrated in Fig. 1. The input signal is considered being a white Gaussian process with a mean of zero and a variance of one, as described above. Simulations are carried out with a filter size of $N = 16$ taps and 300 independent runs. It is possible to replace the expectation operation in (8) with the following:

$$P_x(n) = (1 - \gamma) P_x(n - 1) + \gamma x^2(n) \tag{38}$$

where $x(n)$ is the signal to be averaged and $\gamma = 0.01$.

5.1 Additive white Gaussian noise

The SS-MSE performance of the convex RI algorithm is compared to that of the theoretical and observational SS-

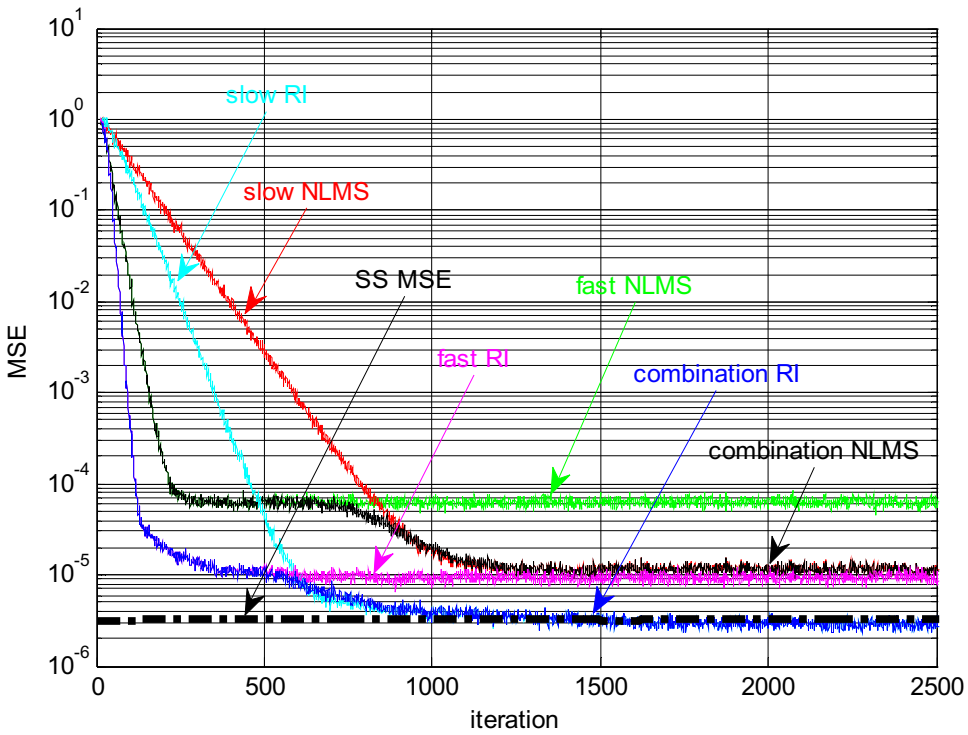

Fig. 3 MSE combination curves of NLMS (MSE = −52.4 dB) and RI algorithms (MSE = −58 dB) in AWGN

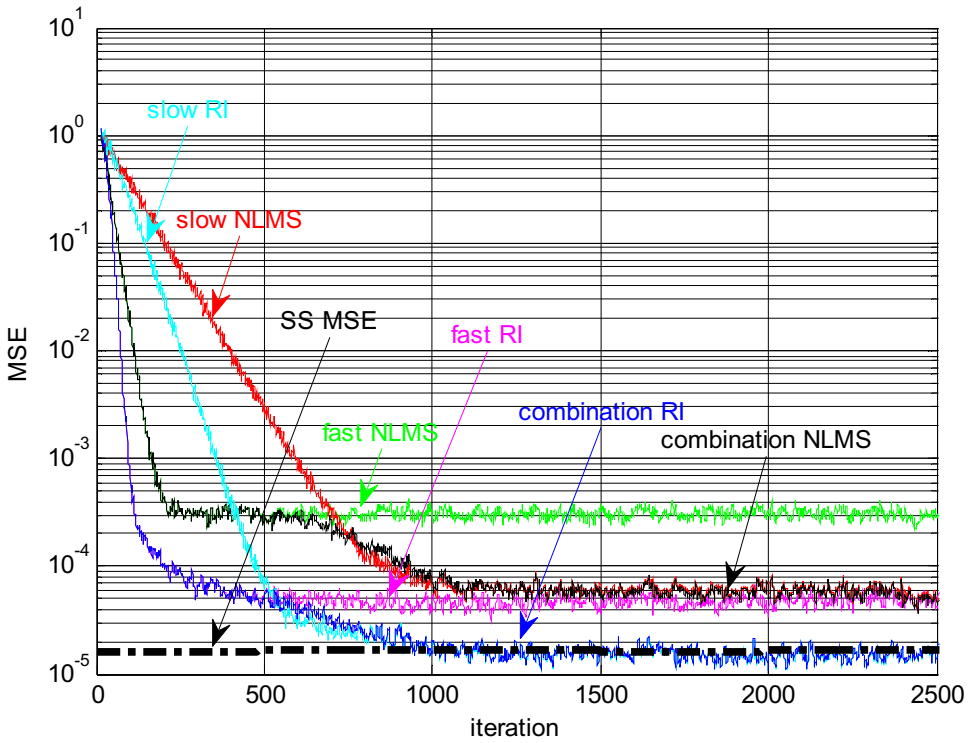

Fig. 4 MSE combination curves of NLMS (MSE = −44.5 dB) and RI algorithms (MSE = −51 dB) in ACGN

MSE performance in the first experiment, which is carried out in an AWGN setting. The convex RI method's performance is also evaluated against the convex NLMS method in terms of convergence rate and the SS-MSE. The mean of the AWGN process is considered to be zero and variance $\sigma_v^2 = 2 \times 10^{-4}$.

The following parameters were used in the simulation: For the convex RI: $\beta_1 = 0.994$, $\mu_1 = 0.001$, $\beta_2 = 0.998$ and $\mu_2 = 0.00003$. For the convex NLMS, the values are:

$\mu_1 = 0.45$ and $\mu_2 = 0.1$. The performances of the convex RI and NLMS techniques are depicted in Fig. 3. It can be seen from the figure that the theoretical and experimental SS-MSEs of the convex RI method are in agreement with each other (−58 dB). On the other hand, when subjected to identical circumstances, the proposed method converges to the SS-MSE roughly 350 iterations earlier than the convex NLMS technique, with an MSE that is 5.6 dB lower.

5.2 Additive correlated Gaussian noise

In order to demonstrate the influence of noise correlation on the capabilities (both theoretically and experimentally) of the convex RI approach, it is assumed that the input signal has been corrupted by an ACGN. Using an AR(1) process ($\eta(n) = 0.9\eta(n) + v(n)$), where $v(n)$ is an AWGN process with zero mean and variance $\sigma_v^2 = 2 \times 10^{-4}$, the ACGN is formed. Simulations of the algorithms are performed using the same parameters as those used in the experiment described in Sect. 5.1.

Figure 4 illustrates that the theoretical and experimental SS-MSE of the convex RI are once again in agreement with each other. Furthermore, the convex RI technique converges to a SS-MSE with 6.5 dB less than that of the convex NLMS approach and with a convergence rate that is nearly 350 iterations faster than the convex NLMS algorithm.

5.3 MRI image with additive white Gaussian noise

In this experiment, the capability of the proposed technique in denoising images id examined on an MRI image of a breast cancer patient. The input image is considered to be normalized and of size 204 × 204 pixels with 150 Gy levels, and it has been distorted by an AWGN with zero mean and variance ($\sigma^2 = 0.3$). The parameters used for the proposed method: $\beta = 0.998$ and $\mu_0 = 0.0005$. For the slow algorithm: $\beta = 0.992$ and $\mu_0 = 0.000001$.

The original image is depicted in Fig. 5a. Figure 5b depicts an image that has been corrupted by an AWGN.

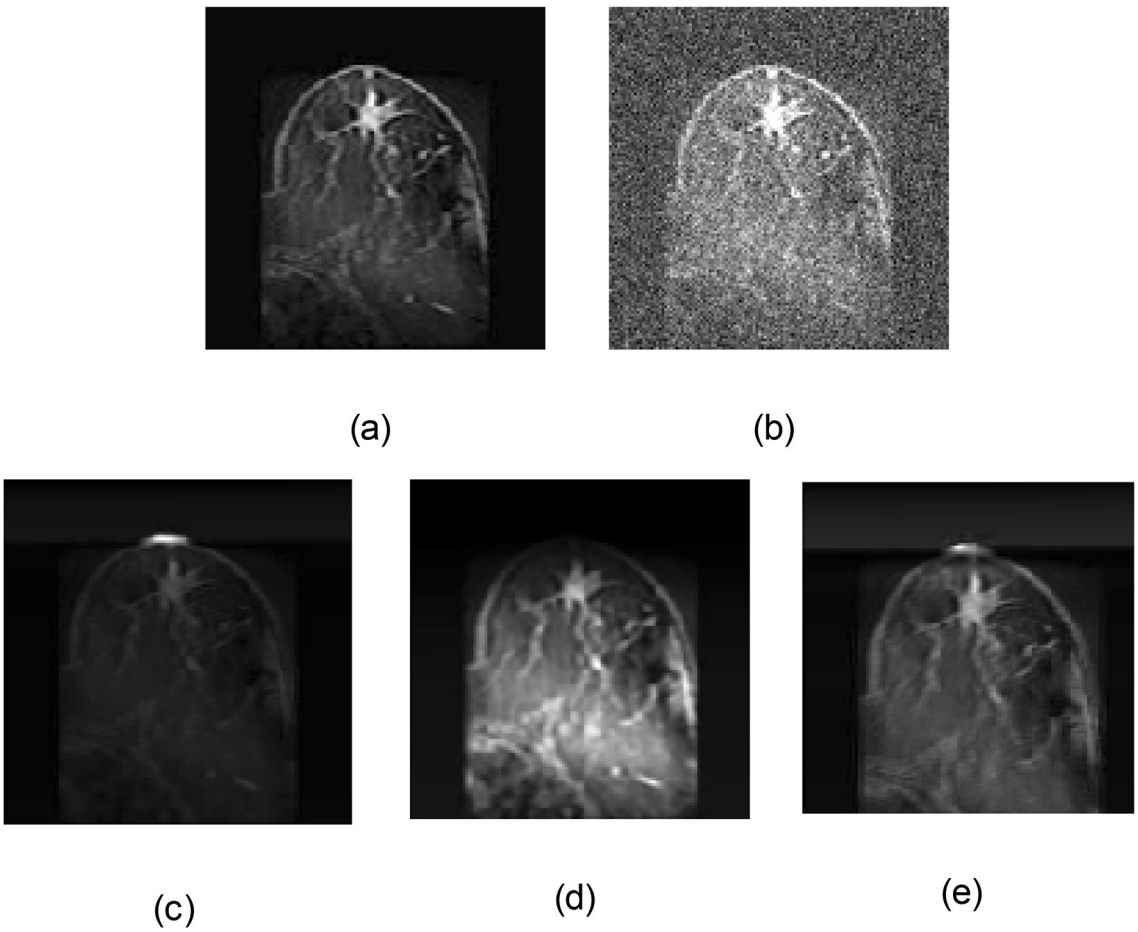

Fig. 5 **a** Original image of breast cancer, **b** Original image corrupted by an AWGN, **c** Recovered image using the fast RI algorithm (PSNR = 30 dB), **d** Recovered image using the slow RI (PSNR = 33.5 dB), and **e** Recovered image using the convex combination of RI algorithms (PSNR = 35.3 dB)

The image restored by the fast RI technique is depicted in Fig. 5c. It is easy to observe that the method converges quickly (the top part of the image is relatively clear). Additionally, the image's relative darkness can be used to determine the image's relatively high MSE. Figure 5d, depicts the image that was recovered by the slow RI technique. The slow convergence and lower MSE of the slow RI method are indicated by the dark area at the top of the image and the relatively better reconstructed image, respectively, which are both visible in the dark region at the top of the image and the considerably cleaner restored image. A combination of the higher performance of Fig. 5c, d is depicted in Fig. 5e. Because of this, we can observe the clear top of the image (which indicates fast convergence) and a clearer overall image (which indicates low MSE) in Fig. 5e.

6 Conclusion

In conclusion, this paper presented the convexly combined RI algorithm and its analysis, the SS-MSE criterion is deduced from the data. Based on the simulation findings, the experimental data of the convexly combined RI algorithm are consistent with the theoretical results. On the other hand, the algorithm outperforms the convexly combined NLMS algorithm in both AWGN and ACGN environments for the stated noise cancelation setting when compared to the convexly combined NLMS algorithm. In addition, a 2D variant of the technique is proposed based on the consideration of a mask that moves horizontally to the right by one column at a time till the end of each row, a 2D version has been created to illustrate the concept of a mask. Then the process is repeated with the following row below it until the end of the last row and column of the image is reached, at which point the operation is completed. The proposed algorithm has demonstrated excellent performance in image de-noising applications. As a future work, the convergence analysis of the 2D version may be investigated. Also, investigating the performances of the proposed methods in different applications and noise environments will be a very interesting research topic.

Declarations

Conflict of interest No financial interest. This research is not funded. This work is neither sent nor published elsewhere.

References

1. Elsayed WM, El-Bakry HM, El-Sayed SM (2020) Fault autonomous model handling through integrated adaptive-filters for eliminating deployment faults in wireless sensor networks. Wireless Sensor Syst IET 10(5):236–241
2. Shukl P, Singh B (2021) Combined IIR and FIR filter for improved power quality of PV interfaced utility grid. IEEE Trans Ind Appl 57(1):774–783
3. Jiang B, Lu W (2021) Adaptive multiple subtraction based on an accelerating iterative curvelet thresholding method. IEEE Trans Image Process 30:806–821
4. Liu S, Chen J, Xun Y, Zhao X, Chang CH (2020) A new polarization image demosaicking algorithm by exploiting inter-channel correlations with guided filtering. IEEE Trans Image Process 29:7076–7089
5. Zhang B, Wang M, Shen X (2021) Image haze removal algorithm based on nonsubsampled contourlet transform. IEEE Access 9:21708–21720
6. Galetto FJ, Deng G, Al-Nasrawi M, Waheed W (2021) Edge-aware filter based on adaptive patch variance weighted average. IEEE Access 9:118291–118306
7. Gwadabe TR, Salman MS (2017) A new leaky-LMS algorithm with analysis. Int Arab J Inf Technol 14(17):324–331
8. Capizzi G, Sciuto GL (2019) A novel 2-D FIR filter design methodology based on a Gaussian-based approximation. IEEE Signal Process Lett 26(2):362–366
9. Salman MS (2014) Sparse leaky-LMS algorithm for system identification and its convergence analysis. Int J Adapt Control Signal Process 28(10):1065–1072
10. Ma W, Qu H, Gui G, Xu L, Zhao J, Chen B (2015) Maximum correntropy criterion based sparse adaptive filtering algorithms for robust channel estimation under non-Gaussian environments. J Franklin Inst 352(7):2708–2727
11. Muneyasu M, Hinamoto T, Yagi H (1998) A realization of 2-D adaptive filters using affine projection algorithm. J Franklin Inst 335(7):1185–1193
12. Turan C, Salman MS (2014) A sparse function controlled variable step-size LMS algorithm for system identification. In Proceedings of the Signal Processing and Communications Applications Conference, pp. 329–332
13. Ahmad MS, Kukrer O, Hocanin A (2011) Recursive inverse adaptive filtering algorithm. Digital Signal Process 21(4):491–496
14. Ahmad MS, Kukrer O, Hocanin A (2010) Recursive inverse adaptive filter with second order estimation of autocorrelation matrix. In IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), pp. 482–484
15. Ahmad MS, Kukrer O, Hocanin A (2012) Robust recursive inverse adaptive algorithm in impulsive noise. Circuits Syst Signal Process 31(2):703–710
16. Ahmad MS, Kukrer O, Hocanin A (2013) Recursive inverse adaptive algorithm: a second-order version, a fast implementation technique, and further results. SIViP 9(3):665–673
17. Bernardi G, van Waterschoot T, Wouters J, Moonen M (2017) Adaptive feedback cancellation using a partitioned-block frequency-domain Kalman filter approach with PEM-based signal prewhitening. IEEE/ACM Trans Audio Speech Lang Process 25(9):1784–1798
18. Song X, Han X, Wang F, Xin F (2019) Recursive updating algorithm for robust adaptive beamforming in a uniform circular array. Int J Wireless Inf Networks 26:331–343

19. Suresh S, Lal S (2017) Two-dimensional CS adaptive FIR Wiener filtering algorithm for the denoising of satellite images. IEEE J Sel Top Appl Earth Observ Remote Sens 10(12):5245–5257
20. Jiang H, Liu L, Jonker PP, Elliott DG, Lombardi F, Han J (2019) A high-performance and energy-efficient FIR adaptive filter using approximate distributed arithmetic circuits. IEEE Trans Circuits Syst I Regul Pap 66(1):313–326
21. Elsayed WM, El-Bakry HM, El-Sayed SM (2019) Data reduction using integrated adaptive filters for energy-efficient in the clusters of wireless sensor networks. IEEE Embed Syst Lett 11(4):119–122
22. Huang G, Geng C, Li F, Liu J, Li X (2018) Control bandwidth promotion of adaptive fiber-optics collimator and its application in coherent beam combination. IEEE Photonics J 10(6):1–13
23. Guo X, Jiang J, Chen J, Du S, Tan L (2019) Convex combination recursive even mirror Fourier nonlinear filter for nonlinear active noise control. In 22nd International Conference on Electrical Machines and Systems (ICEMS), pp. 1–6
24. Huang L, Fan S, Xu W, Yang TC (2016) Improving robustness of passive source localization via convex optimization based mode filtering. OCEANS 2016 - Shanghai, pp. 1–5
25. Ahmad MS, Kukrer O, Hocanin A (2011) A 2-D recursive inverse adaptive algorithm. SIViP 7(2):221–226