

# On the Performance Analysis of a Class of Transform-domain NLMS Algorithms with Gaussian Inputs and Mixture Gaussian Additive Noise Environment

S. C. Chan · Y. Zhou

Received: 6 October 2009 / Revised: 3 May 2010 / Accepted: 3 May 2010 / Published online: 29 June 2010  
© The Author(s) 2010. This article is published with open access at Springerlink.com

**Abstract** This paper studies the convergence performance of the transform domain normalized least mean square (TDNLMS) algorithm with general nonlinearity and the transform domain normalized least mean M-estimate (TDNLMM) algorithm in Gaussian inputs and additive Gaussian and impulsive noise environment. The TDNLMM algorithm, which is derived from robust M-estimation, has the advantage of improved performance over the conventional TDNLMS algorithm in combating impulsive noises. Using Price's theorem and its extension, the above algorithms can be treated in a single framework respectively for Gaussian and impulsive noise environments. Further, by introducing new special integral functions, related expectations can be evaluated so as to obtain decoupled difference equations which describe the mean and mean square behaviors of the TDNLMS and TDNLMM algorithms. These analytical results reveal the advantages of the TDNLMM algorithm in impulsive noise environment, and are in good agreement with computer simulation results.

**Keywords** Adaptive filters · Transform domain normalized least mean square (TDNLMS) · TD normalized least mean M-estimate (TDNLMM) · M-estimation · Convergence performance analysis · Impulsive noise

## 1 Introduction

Adaptive filters are widely used for filtering problems in which the statistics of the underlying signals are either unknown *a priori*, or in some cases, slowly-varying. Many adaptive filtering algorithms have been proposed and they are usually variants of the well known least mean square (LMS) [1] and the recursive least squares (RLS) [2] algorithms. An important variant of the LMS algorithm is the normalized least mean square (NLMS) algorithm [3], where the step size is normalized with respect to the energy of the input vector. Due to the numerical stability and computational simplicity of the LMS and the NLMS algorithms, they have been widely used in various applications [4, 5].

An important class of NLMS is the transform domain NLMS (TDNLMS) algorithms [6–11] where unitary transformations such as the discrete Fourier transform (DFT), the discrete cosine transform (DCT), and the wavelet transform (WT) are employed to pre-whiten the input signal. Prewhitening and element-wise normalization usually help to reduce the eigenvalue spread of the input autocorrelation matrix and hence significantly improve the convergence speed. Driven by the practical advantages of the TDNLMS family, there is also considerable interest in the performance analysis of these algorithms [8, 9]. Results concerning the performance behaviors of the TDNLMS algorithm were studied in [6–11].

---

Part of this work was presented at IEEE APCCAS 2008 [31].

S. C. Chan · Y. Zhou  
Department of Electrical and Electronic Engineering,  
The University of Hong Kong,  
Pokfulam Road, Hong Kong  
e-mail: scchan@eee.hku.hk

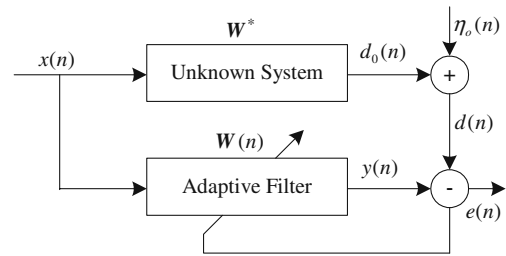
Y. Zhou (✉)  
Institute of Acoustics, Chinese Academy of Sciences,  
21#, North 4th Ring Road West,  
Beijing, China  
e-mail: yizhou@mail.ioa.ac.cn  
e-mail: yizhou@eee.hku.hk

In this paper, we study a more general TDNLMS algorithm, the TDNLMS algorithm with general error nonlinearity. The convergence performance of this algorithm in Gaussian inputs and additive Gaussian and impulsive noise environments are studied. The main novelty lies in handling the normalization, evaluating the expectations specific to this algorithm and dealing with the error nonlinearity. We study with particular emphasis on two special cases of this algorithm: the conventional TDNLMS algorithm with no nonlinearity, and the transform domain normalized least mean M-estimate (TDNLMM) algorithm [12], which is based on robust M-estimation [13, 14] and adaptive threshold selection (ATS) [12, 15]. These techniques have been successfully employed in the LMM [12], the recursive least M-estimate (RLM) [15] and the normalized LMM (NLMM) [16] algorithms for robust filtering in impulsive noise environment. The motivation of studying this algorithm is that the performance of the TDNLMS algorithm, which is based on LS estimation as in the LMS algorithm, will deteriorate considerably when the desired or the input signal is corrupted by impulsive noise. The mean and mean square convergence analysis for the TDNLMS algorithm with general error nonlinearity is treated in a single framework using the Price's theorem [17] for Gaussian case and its extension [18] for contaminated Gaussian (CG) case. The finally obtained decoupled difference equations clearly interpret the convergence performance of all the studied algorithms. The validity of the analytical results is verified through extensive simulations and they are in good agreement with each other. The rest of this paper is organized as follows: In section 2, the TDNLMS and TDNLMM algorithms are reviewed and the TDNLMS algorithm with general error nonlinearity is formulated. Their convergence performance analysis is given in section 3. Computer simulations are conducted in section 4. Finally, conclusions are drawn in section 5.

## 2 TDNLMS Algorithm with General Error Nonlinearity and TDNLMM Algorithm

### 2.1 The TDNLMS Algorithm

Consider the adaptive system identification problem in Fig. 1 where an input signal  $x(n)$  is applied simultaneously to an adaptive transversal filter of order  $L$  with weight vector  $\mathbf{W}(n) = [w_1(n), w_2(n), \dots, w_L(n)]^T$  and an unknown system to be identified with an impulse response  $\mathbf{W}^* = [w_1, w_2, \dots, w_L]^T$ .  $\mathbf{X}(n) = [x(n), x(n-1), \dots, x(n-L+1)]^T$  is the input vector and the superscript  $T$  denotes the transpose of a vector or a matrix.  $e(n)$  is the estimation error and  $d(n)$  is the desired signal of the adaptive filter, which may be corrupted by an additive noise  $\eta_o(n)$ . Hence



**Figure 1** Adaptive system identification.

$$d(n) = \mathbf{X}^T(n)\mathbf{W}^* + \eta_o(n) \quad (1)$$

The update equations for the TDNLMS algorithm can be written as:

$$e(n) = d(n) - \mathbf{W}^T(n)\mathbf{X}_C(n), \quad (2)$$

$$\mathbf{W}(n+1) = \mathbf{W}(n) + \mu\Lambda_C^{-1}\mathbf{X}_C(n)e(n), \quad (3)$$

where  $\mu$  is a constant step size parameter controlling the convergence rate and steady state error of the algorithm.  $\mathbf{X}_C(n) = \mathbf{C}\mathbf{X}(n) = [X_{C,1}(n), X_{C,2}(n), \dots, X_{C,L}(n)]^T$  is the transformed signal vector.  $\mathbf{C}$  is an  $L \times L$  transform matrix such as (DFT) or (DCT).  $\Lambda_C^{-1} = \text{diag}[\varepsilon_1^{-1}(n), \varepsilon_2^{-1}(n), \dots, \varepsilon_L^{-1}(n)]$  which is an element-wise normalization matrix with  $\varepsilon_i(n)$  being the estimated power of the  $i$ -th signal component after transformation. Common methods for choosing  $\varepsilon_i(n)$  include  $\varepsilon_i + X_{C,i}^2(n)$  and  $\varepsilon_i(n) = (1 - \alpha_\varepsilon)\varepsilon_i(n-1) + \alpha_\varepsilon X_{C,i}^2(n)$ , where  $\alpha_\varepsilon$  is a positive forgetting factor smaller than one.  $\varepsilon_i$  is a small positive value used to avoid division by zero or it can be chosen as certain prior power estimate of the corresponding component. In the analysis to be presented in section 3, a form  $\varepsilon_i + \alpha_\varepsilon X_{C,i}^2(n)$  similar to the above two choices will be chosen. In the simulation section, we shall introduce a method to approximately analyze the effect of this choice.

### 2.2 The TDNLMM Algorithm and TDNLMS Algorithm with General Error Nonlinearity

Many techniques have been proposed to combat the adverse effect of impulsive noise on adaptive filters. They include the median-filtering algorithms [19, 20], the nonlinear clipping approaches [21, 22], and approaches based on robust statistics [12, 15, 16]. The LMM [12] and the RLM [15] algorithms are two effective algorithms derived from robust M-estimation and their improved robustness in impulsive noise and performance comparison with other relevant algorithms were thoroughly discussed in [12] and [15].

In the TDNLMM algorithm [12], an M-estimate distortion measure  $J_\rho = E[\rho(e(n))]$  is minimized, where  $\rho(e)$ , as illustrated in Fig. 2 (a), is chosen as the modified Huber (MH) function:

$$\rho(e) = \begin{cases} e^2/2, & 0 \leq |e| < \xi \\ \xi^2/2, & \xi \leq |e|. \end{cases} \quad (4)$$

$\xi$  is a threshold parameter used to suppress the effect of outlier when the estimation error  $e$  is very large. Other M-estimate function such as the Hampel’s three-part redescending function [14] can also be used. Notice that when  $\rho(e) = e^2/2$  it reduces to the conventional mean square error (MSE) criterion. Like the LMS algorithm,  $J_\rho$  is minimized by updating  $\mathbf{W}(n)$  in the negative direction of the instantaneous gradient vector  $\widehat{\nabla}_{\mathbf{W}\rho}$ . Therefore, the gradient vector,  $\nabla_{\mathbf{W}}(J_\rho)$ , is approximated by  $\widehat{\nabla}_{\mathbf{W}\rho} = -\partial\rho(e(n))/\partial\mathbf{W} = -\psi(e(n))\mathbf{X}(n)$ , where  $\psi(e) = \partial\rho(e)/\partial e$  is the score function, which is depicted in Fig. 2 (b). The following LMM algorithm can be obtained:

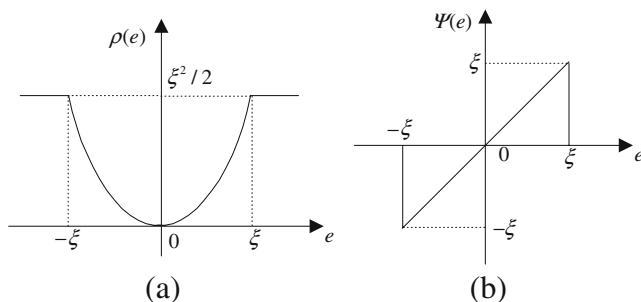
$$\begin{aligned} \mathbf{W}(n+1) &= \mathbf{W}(n) - \mu\widehat{\nabla}_{\mathbf{W}\rho} \\ &= \mathbf{W}(n) + \mu\psi(e(n))\mathbf{X}(n). \end{aligned} \quad (5)$$

It can be seen that when  $e(n)$  is smaller than  $\xi$ ,  $\psi(e(n))$  is equal to  $e(n)$  and (5) becomes identical to the LMS algorithm. When  $|e(n)| > \xi$ ,  $\psi(e(n))$  will become zero. Thus the LMM algorithm can effectively reduce the adverse effect of large estimation error on updating the filter coefficients. In the adaptive threshold selection (ATS) method used in [12, 15],  $e(n)$  is assumed to be Gaussian distributed except being corrupted occasionally by additive impulsive noise and the following robust variance estimate is proposed

$$\widehat{\sigma}_e^2(n) = \lambda_\sigma \widehat{\sigma}_e^2(n-1) + c_1(1 - \lambda_\sigma)\text{med}(A_e(n)), \quad (6)$$

where  $\lambda_\sigma$  is a forgetting factor close to but smaller than one,  $c_1=2.13$  is the finite sample correction factor and  $N_w$  is the length of the data set.  $\text{med}(\cdot)$  is the median operator and  $A_e(n) = [e^2(n), e^2(n-1), \dots, e^2(n-N_w+1)]$ . Using (6), the following adaptive threshold  $\xi$  can be obtained:

$$\xi = k_\xi \widehat{\sigma}_e(n). \quad (7)$$



**Figure 2** a The MH function  $\rho(e)$ ; b  $\psi(e)$ , the MH score function of  $\rho(e)$ .

$k_\xi$  is a constant used to control the suppression of impulsive interference. A reasonable value of  $k_\xi$  is 2.576 and the window length  $N_w$  is usually chosen between 5 and 9 [12, 15].

If the step sizes for updating the coefficients are normalized according to the power of the corresponding transform signal components as in the TDNLMS algorithm, the following TDNLMM algorithm can be obtained from (5) [12]:

$$e(n) = d(n) - \mathbf{W}^T(n)\mathbf{X}_C(n), \quad (8)$$

$$\mathbf{W}(n+1) = \mathbf{W}(n) + \mu\Lambda_C^{-1}\psi(e(n))\mathbf{X}_C(n). \quad (9)$$

The convergence performance of the LMS algorithm with other nonlinearity than MH function can be found in literature. The LMS algorithm with error function nonlinearity was studied in [23]. A related algorithm is the dual-sign LMS [24] algorithm. The former concluded that the nonlinearity will slow down the convergence rate, while the latter is mainly introduced to reduce the implementation complexity. The robustness of this class of algorithms to impulsive outliers was later studied by Koike in [22, 25, 26], and in [21] using the clipping nonlinearity. On the contrary, in [12, 15] the threshold parameter  $\xi$  in the MH function is continuously updated as in (7), which greatly improves the convergence speed and steady state error.

### 3 Mean and Mean Square Convergence Analysis

In this section, the convergence performance analysis of the TDNLMS algorithm with general nonlinearity and particularly the TDNLMS and TDNLMM algorithms will be studied. The main contributions of the analysis include: i) the use of the Price’s theorem [17] to handle the nonlinearity for Gaussian noise case and its extension [18] for the CG noise case, and ii) introduction of new special functions and the evaluation of related expectations in order to obtain decoupled difference equations describing the mean and mean square behaviors of the algorithms. To simplify the analysis, we make the following assumptions:

*Assumption 1* The input signal  $x(n)$  is an ergodic process which is Gaussian distributed with zero mean and autocorrelation matrix  $\mathbf{R}_{XX} = E[\mathbf{X}(n)\mathbf{X}^T(n)]$ .

*Assumption 2* The additive noise  $\eta_o(n)$  is assumed to be a Gaussian noise ( $\eta_o(n) = \eta_g(n)$ ) for the analysis in sec-

tion 3.1 below. For the analysis in section 3.2 below,  $\eta_o(n)$  is modeled as a CG noise [27] which is a frequently used model for analyzing impulsive noise. More precisely, it is given by:

$$\eta_o(n) = \eta_g(n) + \eta_{im}(n) = \eta_g(n) + b(n)\eta_w(n), \tag{10}$$

where  $\eta_g(n)$  and  $\eta_w(n)$  are both independent and identically distributed (i.i.d.) zero mean Gaussian sequences with respective variance  $\sigma_g^2$  and  $\sigma_w^2$ .  $b(n)$  is an i.i.d. Bernoulli random sequence whose value at any time instant is either zero or one, with occurrence probabilities  $P_r(b(n) = 1) = p_r$  and  $P_r(b(n) = 0) = 1 - p_r$ . The variances of the random processes  $\eta_{im}(n)$  and  $\eta_o(n)$  are then given by  $\sigma_{im}^2 = p_r\sigma_w^2$  and  $\sigma_{\eta_o}^2 = \sigma_g^2 + \sigma_{im}^2 = \sigma_g^2 + p_r\sigma_w^2$ . The ratio  $r_{im} = \sigma_{im}^2/\sigma_g^2 = p_r\sigma_w^2/\sigma_g^2$  is a measure of the impulsive characteristic of the CG noise. Accordingly, the probability distribution function (PDF) of this CG distribution is given by

$$f_{\eta_o}(\eta) = \frac{1 - p_r}{\sqrt{2\pi\sigma_g^2}} \exp\left(-\frac{\eta^2}{2\sigma_g^2}\right) + \frac{p_r}{\sqrt{2\pi\sigma_{\Sigma}^2}} \exp\left(-\frac{\eta^2}{2\sigma_{\Sigma}^2}\right). \tag{11}$$

*Assumption 3*  $\mathbf{W}(n)$ ,  $x(n)$  and  $\eta_o(n)$  are statistically independent (the independent assumption [1]). Although this assumption is not completely valid in general applications, it is a good approximation for large value of  $L$  and is commonly used to simplify the convergence analysis of adaptive filtering algorithms. Moreover, we denote  $\mathbf{W}^* = \mathbf{R}_{X_C X_C}^{-1} \mathbf{P}_{dX_C}$ , where  $\mathbf{P}_{dX_C} = E[d(n)\mathbf{X}_C(n)]$  is the ensemble-averaged cross-correlation vector between  $\mathbf{X}_C(n)$  and  $d(n)$ .  $\mathbf{W}^*$  is related to the optimal Wiener solution  $\mathbf{W}_{OPT} = \mathbf{R}_{XX}^{-1} \mathbf{P}_{dX}$  by  $\mathbf{W}_{OPT} = \mathbf{C}\mathbf{W}^*$ .

### 3.1 Mean and Mean Square Convergence Behaviors in Gaussian noise

#### 3.1.1 Mean Behavior

From (9), the weight-error vector  $\mathbf{v}(n) = \mathbf{W}^* - \mathbf{W}(n)$  for the TDNLMS algorithm with general nonlinearity can be written as

$$\mathbf{v}(n + 1) = \mathbf{v}(n) - \mu \Lambda_C^{-1} \psi(e(n)) \mathbf{X}_C(n), \tag{12}$$

where  $\mathbf{W}^*$  is the transformed optimal weight vector defined above and  $\psi(e(n))$  is a general nonlinearity. When it is equal to  $e(n)$ , (12) reduces to the conventional TDNLMS algorithm. Taking expectation over  $\{\mathbf{v}, \mathbf{X}_C, \eta_g\}$  on both sides of (12), one gets

$$E[\mathbf{v}(n + 1)] = E[\mathbf{v}(n)] - \mu \mathbf{H}, \tag{13}$$

where  $E[\cdot]$  denotes the expectation over  $\{\mathbf{v}(n), \mathbf{X}_C(n), \eta_g(n)\}$  (also written as  $E_{\{\mathbf{v}, \mathbf{X}_C, \eta_g\}}[\cdot]$  for clarity), and  $\mathbf{H} = E_{\{\mathbf{v}, \mathbf{X}_C, \eta_g\}}[\Lambda_C^{-1} \psi(e(n)) \mathbf{X}_C(n)]$ . By dropping the time index of  $\mathbf{X}_C(n)$ ,  $e(n)$ , and  $\eta_g(n)$ , one gets

$$\mathbf{H} = E_{\{\mathbf{v}, \mathbf{X}_C, \eta_g\}}[\Lambda_C^{-1} \psi(e) \mathbf{X}_C] = E_{\{\mathbf{v}\}}[\mathbf{H}_1] \tag{14}$$

where  $\mathbf{H}_1 = E_{\{\mathbf{X}_C, \eta_g\}}[\Lambda_C^{-1} \psi(e) \mathbf{X}_C | \mathbf{v}]$  and the second equation is obtained from the independence assumption of  $\eta_g(n)$ ,  $\mathbf{W}(n)$  and  $x(n)$  in Assumption 3.

The  $i$ -th component of  $\mathbf{H}_1$  is evaluated in Appendix A to be

$$H_{1,i} \approx \overline{\psi'}(\sigma_e^2(n)) \alpha_i \mathbf{e}_i^T \mathbf{R}_{X_C X_C} \mathbf{v}(n), \tag{15}$$

where  $\sigma_e^2(n) = E[\mathbf{v}^T(n) \mathbf{R}_{X_C X_C} \mathbf{v}(n)] + \sigma_g^2$ ,  $\overline{\psi'}(\sigma_e^2) = \int_{-\infty}^{\infty} \frac{\psi'(e)}{\sqrt{2\pi\sigma_e}} \exp\left(-\frac{e^2}{2\sigma_e^2}\right) de$ ,  $\alpha_i = \int_0^{\infty} \exp(-\beta \varepsilon_i) \left(g_i(\tilde{\beta})\right)^{-3/2} d\beta$ ,  $g_i(\tilde{\beta}) = \left(1 + 2\tilde{\beta} R_{X_C X_C - i, i}\right)$ ,  $R_{X_C X_C - i, i}$  is the  $(i, i)$ -th element of  $\mathbf{R}_{X_C X_C}$ ,  $\mathbf{e}_i$  is a column vector with the  $i$ -th element equal to one and zero elsewhere. For a given  $\psi(e)$ ,  $\overline{\psi'}(\sigma_e^2)$  can be evaluated analytically or numerically. Substituting (14), (15) into (13), the following mean weight-error vector update equation is obtained:

$$E[\mathbf{v}(n + 1)] = (\mathbf{I} - \mu A_{\psi}(\sigma_e^2(n)) \mathbf{D}_{\alpha} \mathbf{R}_{X_C X_C}) E[\mathbf{v}(n)], \tag{16}$$

where  $\mathbf{D}_{\alpha} = \text{diag}(\alpha_1, \dots, \alpha_L)$  is a diagonal matrix. For notation convenience, we write  $\overline{\psi'}(\sigma_e^2(n))$  as  $A_{\psi}(\sigma_e^2(n))$  and use  $\sigma_e^2(n)$  and  $\sigma_e^2$  interchangeably. Also we replace the approximate sign in (16) by the equality sign. Let  $\mathbf{V}(n) = \mathbf{D}_{\alpha}^{-1/2} \mathbf{v}(n)$ , (16) can be simplified to

$$E[\mathbf{V}(n + 1)] = (\mathbf{I} - \mu A_{\psi}(\sigma_e^2(n)) \mathbf{R}_{X_D X_D}) E[\mathbf{V}(n)], \tag{17}$$

where  $\mathbf{R}_{X_D X_D} = \mathbf{D}_{\alpha}^{1/2} \mathbf{R}_{X_C X_C} \mathbf{D}_{\alpha}^{1/2}$  is the correlation matrix of a scaled input vector  $\mathbf{X}_D = \mathbf{D}_{\alpha}^{1/2} \mathbf{X}_C$ . Since it is symmetric, it can be written as the following eigenvalue decomposition (EVD):  $\mathbf{R}_{X_D X_D} = \mathbf{U}_{X_D} \Lambda_{X_D} \mathbf{U}_{X_D}^T$ , where  $\mathbf{U}_{X_D}$  is certain orthogonal matrix and  $\Lambda_{X_D} = \text{diag}(\lambda'_1, \lambda'_2, \dots, \lambda'_L)$  contains the corresponding eigenvalues. Pre-multiplying both sides of (17) with  $\mathbf{U}_{X_D}^T$  gives

$$E[\mathbf{V}_D(n + 1)] = (\mathbf{I} - A_{\psi}(\sigma_e^2(n)) \Lambda_{X_D}) E[\mathbf{V}_D(n)], \tag{18}$$

where  $E[\mathbf{V}_D(n)] = \mathbf{U}_{X_D}^T E[\mathbf{V}(n)]$ . This is equivalent to the following  $L$  scalar first order finite difference equations:

$$E[\mathbf{V}_D(n + 1)]_i = \left(1 - A_{\psi}(\sigma_e^2(n)) \lambda'_i\right) E[\mathbf{V}_D(n)]_i, \tag{19}$$

where  $E[\mathbf{V}_D(n)]_i$  is the  $i$ -th element of the vector  $E[\mathbf{V}_D(n)]$  for  $i = 1, 2, \dots, L$ .

#### Remarks

(R-A1): The TDNLMS algorithm

For conventional TDNLMS algorithm,  $\psi(e) = e$  and  $\overline{\psi'}(\sigma_e^2) = A_\psi(\sigma_e^2) = 1$ . The algorithm will converge if

$$|1 - \mu\lambda'_i| < 1, \quad \text{for all } i, \tag{20}$$

where  $\lambda'_i$  is the  $i$ -th eigenvalue of  $\mathbf{R}_{X_D X_D}$ . The corresponding maximum step size for convergence should satisfy

$$\mu_{\max} < 2/\lambda'_{\max}, \quad \text{for all } i, \tag{21}$$

where  $\lambda'_{\max}$  is the maximum eigenvalue of  $\mathbf{R}_{X_D X_D}$ . Let us examine the eigenvalues of  $\mathbf{R}_{X_D X_D}$ . We note that  $\mathbf{R}_{X_D X_D} = \mathbf{D}_\alpha^{1/2}(\mathbf{C}\mathbf{R}_{XX}\mathbf{C}^T)\mathbf{D}_\alpha^{1/2}$ . It can be shown that  $\alpha_i = \frac{1}{2\alpha_\varepsilon R_{X_C X_C-i,i}} \exp\left(\frac{1}{2}\varepsilon_i \alpha_\varepsilon^{-1} R_{X_C X_C-i,i}^{-1}\right) \cdot E_{3/2}\left(\frac{1}{2}\varepsilon_i \alpha_\varepsilon^{-1} R_{X_C X_C-i,i}^{-1}\right)$ , where  $E_n(x) = \int_1^\infty \frac{\exp(-\beta x)}{\beta^n} d\beta$ . The  $i$ -th diagonal element of  $\mathbf{R}_{X_D X_D}$  is

$$R_{X_D X_D-i,i} = \alpha_i R_{X_C X_C-i,i} = \frac{1}{2\alpha_\varepsilon} \exp\left(\frac{1}{2}\varepsilon_i \alpha_\varepsilon^{-1} R_{X_C X_C-i,i}^{-1}\right) \times E_{3/2}\left(\frac{1}{2}\varepsilon_i \alpha_\varepsilon^{-1} R_{X_C X_C-i,i}^{-1}\right) = \hat{\lambda}_i. \tag{22}$$

It can be seen that  $R_{X_D X_D-i,i}$  has the same order as  $R_{X_C X_C-i,i}$ . Therefore, the order of the elements in  $\mathbf{R}_{X_C X_C}$  after scaling, i.e.  $\mathbf{D}_\alpha \mathbf{R}_{X_C X_C} \mathbf{D}_\alpha$ , is preserved. If  $\varepsilon_i$  is simply chosen as  $R_{X_C X_C-i,i}^{-1}$  with  $\alpha_\varepsilon = 0$ , i.e. perfect power estimation, then  $\alpha_i = R_{X_C X_C-i,i}^{-1}$ , and hence  $\hat{\lambda}_i = 1$  for all  $i$ .

If  $\mathbf{C}$  diagonalizes  $\mathbf{R}_{XX}$ , then  $\mathbf{R}_{X_D X_D}$  becomes the identity matrix. The eigenvalue spread is equal to one and it will significantly speed up the convergence of the algorithm, especially for situations with large eigenvalue spread.

Usually  $\mathbf{C}$  only approximately diagonalizes  $\mathbf{R}_{XX}$  and the detailed analysis becomes rather difficult. Here we try to study the eigenvalue and obtain bounds for their values using the Gershgorin circle theorem (GCT). For orthogonal transformation, the eigenvalues of  $\mathbf{R}_{XX}$  and  $\mathbf{R}_{X_C X_C} = \mathbf{C}\mathbf{R}_{XX}\mathbf{C}^T$  are the same. From the GCT, we have

$$|\lambda_i - R_{X_C X_C-i,i}| \leq \sum_{\substack{1 \leq j \leq L \\ j \neq i}} |R_{X_C X_C-i,j}| \\ = \sum_{\substack{1 \leq j \leq L \\ j \neq i}} \left| (R_{X_C X_C-i,i} R_{X_C X_C-j,j})^{1/2} \rho_{X_C X_C-i,j} \right|,$$

where  $\rho_{X_C X_C-i,j}$  is the normalized correlation coefficients. Similarly, the eigenvalues of  $\mathbf{R}_{X_D X_D}$  satisfy

$$|\lambda'_i - \alpha_i R_{X_C X_C-i,i}| \leq \sum_{1 \leq j \neq i \leq L} \left| (\alpha_i \alpha_j)^{1/2} R_{X_C X_C-i,j} \right|.$$

Since  $\alpha_i = (R_{X_C X_C-i,i})^{-1} \hat{\lambda}_i$ , we have  $|\lambda'_i - \hat{\lambda}_i| \leq \sum_{1 \leq j \neq i \leq L} \left| \left( \frac{\hat{\lambda}_i \hat{\lambda}_j}{R_{X_C X_C-i,i} R_{X_C X_C-j,j}} \right)^{-1/2} R_{X_C X_C-i,j} \right|$ . If  $\mathbf{R}_{X_C X_C}$  is diagonal-dominant, then the off-diagonal elements  $\rho_{X_C X_C-i,j}$ ,  $i \neq j$  will be small and

all the eigenvalues of  $\mathbf{R}_{X_D X_D}$  will be close to one with a tight bound.  $\hat{\lambda}_i$  can therefore be viewed as the estimated eigenvalues of  $\mathbf{R}_{X_D X_D}$ . The corresponding estimated eigenvalue spread for diagonal-dominant  $\mathbf{R}_{X_C X_C}$  is

$$\frac{\hat{\lambda}_{i \in \max}}{\hat{\lambda}_{i \in \min}} = \frac{\exp\left(\frac{1}{2}\varepsilon_i \alpha_\varepsilon^{-1} R_{X_C X_C-i,i \in \max}^{-1}\right) E_{3/2}\left(\frac{1}{2}\varepsilon_i \alpha_\varepsilon^{-1} R_{X_C X_C-i,i \in \max}^{-1}\right)}{\exp\left(\frac{1}{2}\varepsilon_i \alpha_\varepsilon^{-1} R_{X_C X_C-i,i \in \min}^{-1}\right) E_{3/2}\left(\frac{1}{2}\varepsilon_i \alpha_\varepsilon^{-1} R_{X_C X_C-i,i \in \min}^{-1}\right)}, \tag{23}$$

which is close to one for a relatively wide range of  $R_{X_C X_C-i,i}$  and  $R_{X_C X_C-j,j}$ . This explains the speed-up in convergence rate of the TDNLMS algorithm even if sub-optimal transformations are used. It was also shown that [9], pp.219] the performance of the TDNLMS algorithm can never be worse than its conventional LMS counterpart and the degree of improvement achieved depends on the distribution of the signal powers at transformed outputs.

(R-A2): TDNLMS algorithm with general nonlinearity and the TDNLMM algorithm

For general nonlinearity other than  $\psi(e) = e$ , (18) or (19) becomes a set of nonlinear difference equations. A general solution is rather difficult to obtain because the term  $A_\psi(\sigma_e^2)$  is dependent on MSE.

For  $\mathbf{C} = \mathbf{D}_\alpha = \mathbf{I}$ , we obtain the LMS algorithm with general nonlinearity. (19) agrees with the result for the LMS algorithm with dual-sign nonlinearity [23]. (18) also agrees with the result in [22] for the LMS algorithm with error function nonlinearity. The case for LMS and NLMM algorithms with general nonlinearity was studied in [30]. For most M-estimate functions,  $\psi(e) = q(e)e$ , where  $q(e)$  is equal to 1 when  $|e|$  is less than a certain threshold  $\xi$  and will gradually decrease to reduce its sensitivity to impulses with large amplitude. Hence,  $0 \leq \psi'(e) \leq 1$  and  $\psi'(e) \approx 1$  when  $|e| < \xi$ . For MH nonlinearity, it can be shown that  $A_{MH}(\sigma_e^2) = A_{MH}(\sigma_e^2) = \frac{2}{\sqrt{2\pi}} \int_0^{\xi/\sigma_e} \exp\left(-\frac{u^2}{2}\right) du - \frac{2\xi}{\sqrt{2\pi}\sigma_e} \exp\left(-\frac{\xi^2}{2\sigma_e^2}\right)$  with  $\lim_{\sigma_e^2 \rightarrow 0} A_{MH}(\sigma_e^2) \rightarrow 1$  and  $\lim_{\sigma_e^2 \rightarrow \infty} A_{MH}(\sigma_e^2) \rightarrow 0$ . For sufficiently small step size  $\mu$ , the algorithm will converge and  $\sigma_e^2$  will decrease. If  $A_{MH}(\sigma_e^2)$  is not made adaptive, an inappropriately chosen  $\xi$  may suppress the signal component, instead of the outliers. This will cause  $A_{MH}(\sigma_e^2)$  to increase gradually and lead to slow adaptation. For the TDNLMS algorithm,  $\xi$  is chosen as a multiple of the estimated  $\sigma_e$  as shown in (7). This helps to maintain a fairly stationary  $A_{MH}(\sigma_e^2)$  so as to avoid significant signal suppression since  $A_{MH}(\sigma_e^2) \approx \text{erf}\left(\frac{\xi}{\sqrt{2}\sigma_e}\right) = A_C$  (if  $\hat{\sigma}_e^2 \approx \sigma_e^2$ ) is approximately constant and slightly less than one. The degradation in convergence over its TDNLMS counterpart is therefore minimal. Though the maximum possible step size is in general difficult to obtain, a sufficient condition

for the algorithm to converge is  $|1 - \mu A_\psi(\sigma_e^2)\lambda'_i| < 1$ , for all  $i$ . If  $\bar{\psi}'(\sigma_e^2)$  is bounded above by a constant  $A_{\psi\_max}$ , then a conservative maximum step size is

$$\mu_{max} < 2 / (A_{\psi\_max}\lambda'_{max}), \tag{24}$$

which yields good estimates in practical algorithms.  $A_\psi(\sigma_e^2)$  for some commonly used error nonlinearities are summarized in Table 1.

### 3.1.2 Mean Square Behavior

Post-multiplying (12) by its transpose and taking expectation over  $\{v, X_C, \eta_g\}$  gives

$$\Xi(n+1) = \Xi(n) - M_1 - M_2 + M_3, \tag{25}$$

where  $\Xi(n) = E[v(n)v^T(n)]$ ,

$$\begin{aligned} M_1 &= \mu E_{\{v\}} \left[ E_{\{X, \eta_g\}} [A_C^{-1} \psi(e) X_C |v\rangle v^T] \right] \\ &= \mu E_{\{v\}} [H v^T] \approx \mu A_\psi(\sigma_e^2) D_\alpha R_{X_C X_C} \Xi(n), \end{aligned} \tag{26}$$

$$M_2 = M_1^T = \mu E_{\{v\}} [v H^T] \approx \mu A_\psi(\sigma_e^2) \Xi(n) R_{X_C X_C} D_\alpha, \tag{27}$$

and

$$\begin{aligned} M_3 &= E_{\{v, X_C, \eta_g\}} \left[ (\mu \psi(e))^2 A_C^{-1} X_C X_C^T A_C^{-1} \right] \\ &= \mu^2 E_{\{v\}} [s_3], \end{aligned} \tag{28}$$

where  $s_3 = E_{\{X, \eta_g\}} [\psi^2(e) A_C^{-1} X_C X_C^T A_C^{-1} |v]$ . Note, the final expressions in (26) and (27) are obtained by using our previous result in (15). The  $(i, j)$ -th element of  $s_3$  is evaluated in Appendix B to be:

$$\begin{aligned} s_{3,ij} &= C_\psi(\sigma_e^2) \left[ s_{ij}^{(0)} (r_{X_C X_C - j})^T v v^T r_{X_C X_C - j} + s_{ij}^{(1)} (r_{X_C X_C - j})^T v v^T \cdot r_{X_C X_C - j} + s_{ij}^{(2)} (r_{X_C X_C - i})^T v v^T r_{X_C X_C - i} + s_{ij}^{(3)} (r_{X_C X_C - i})^T v v^T r_{X_C X_C - j} \right] \\ &\quad + B_\psi(\sigma_e^2) \sum_{m=0}^{\infty} \alpha_{ij}^{(m)} (4^m) \binom{-\frac{3}{2} + m - 1}{m} R_{X_C X_C - ij}^{(2m+1)}. \end{aligned} \tag{29}$$

where

$$\begin{aligned} s_{ij}^{(0)} &= 2 \sum_{m=0}^{\infty} \binom{-\frac{5}{2} + m - 1}{m} (-4 R_{X_C X_C - j, i})^m \alpha_i^{(m, (3+2m)/2)} \alpha_j^{(m, (3+2m)/2)}, \\ s_{ij}^{(1)} &= \frac{1}{2} \sum_{m=0}^{\infty} \binom{-\frac{5}{2} + m - 1}{m} (-4 R_{X_C X_C - j, j})^{m+1} \alpha_i^{(m, (3+2m)/2)} \alpha_j^{(m+1, (5+2m)/2)}, \\ s_{ij}^{(2)} &= \frac{1}{2} \sum_{m=0}^{\infty} \binom{-\frac{5}{2} + m - 1}{m} (-4 R_{X_C X_C - j, j})^{m+1} \alpha_i^{(m+1, (5+2m)/2)} \alpha_j^{(m, (3+2m)/2)}, \\ s_{ij}^{(3)} &= \frac{1}{2} \sum_{m=0}^{\infty} \binom{-\frac{5}{2} + m - 1}{m} (-4 R_{X_C X_C - j, j})^{m+2} \alpha_i^{(m+1, (5+2m)/2)} \alpha_j^{(m+1, (5+2m)/2)}. \\ \alpha_{ij}^{(k)} &= \int_0^\infty \int_0^\infty (\tilde{\beta}_1 \tilde{\beta}_2)^k (g_i(\tilde{\beta}_1) g_j(\tilde{\beta}_2))^{-(2k+3)/2} \cdot \exp(-(\beta_1 \varepsilon_i + \beta_2 \varepsilon_j)) d\beta_2 d\beta_1 = \alpha_i^{(k)} \alpha_j^{(k)}, \end{aligned}$$

**Table 1** List of  $A_\psi(\sigma_e^2)$ ,  $B_\psi(\sigma_e^2)$  and  $C_\psi(\sigma_e^2)$  for three related algorithms.

Nonlinearity	$\psi(\varepsilon)$	$A_\psi(\sigma_e^2)$	$B_\psi(\sigma_e^2)$	$C_\psi(\sigma_e^2)$
Modified Huber	$\psi_{MH}(\varepsilon) = \begin{cases} \varepsilon, &  \varepsilon  \leq \xi \\ 0, & \text{otherwise} \end{cases}$	$\frac{2}{\sqrt{2\pi\sigma_e}} \left[ \int_0^\xi \exp\left(-\frac{e^2}{2\sigma_e^2}\right) de - \xi \exp\left(-\frac{\xi^2}{2\sigma_e^2}\right) \right]$	$\frac{2\sigma_e}{\sqrt{2\pi}} \left[ \int_0^\xi \exp\left(-\frac{e^2}{2\sigma_e^2}\right) de - \xi \exp\left(-\frac{\xi^2}{2\sigma_e^2}\right) \right]$	$\frac{2}{\sqrt{2\pi\sigma_e}} \left[ \int_0^\xi \exp\left(-\frac{e^2}{2\sigma_e^2}\right) de - \xi \exp\left(-\frac{\xi^2}{2\sigma_e^2}\right) - \left(\frac{\xi^3}{\sqrt{2\pi\sigma_e^3}}\right) \exp\left(-\frac{\xi^2}{2\sigma_e^2}\right) \right]$
Error function	$\psi_{EF}(\varepsilon) = \int_0^\varepsilon e^{-u^2/2\sigma_e^2} du$	$\sigma_y / \sqrt{\sigma_y^2 + \sigma_e^2}$	$\sigma_y^2 \sin^{-1}\left(\frac{1}{1 + (\sigma_y^2/\sigma_e^2)}\right)$	$\left[ (1 + 2\sigma_e^2/\sigma_y^2) (1 + \sigma_e^2/\sigma_y^2) \right]^{-1}$
Dual sign	$\psi_{DS}(\varepsilon) = \begin{cases} \text{sgn}(\varepsilon), &  \varepsilon  \leq \tau \\ L \text{sgn}(\varepsilon), &  \varepsilon  > \tau \end{cases}$	$\sqrt{\frac{2}{\pi}} \frac{1}{\sigma_e} \left\{ 1 + (L-1)e^{-[\tau^2/(2\sigma_e^2)]} \right\}$	$L^2 - (L-1) \sqrt{\frac{2}{\pi}} \int_0^{\tau/\sigma_e} \exp(-u^2/2) du$	$\frac{1}{2} e^{-(\tau^2/2\sigma_e^2)} (L^2 - 1) (\tau/\sigma_e^2)$

where  $\alpha_i^{(k)} = \int_0^\infty \int_0^\infty \tilde{\beta}^k (g_i(\tilde{\beta}))^{-(2k+3)/2} \exp(-\beta \varepsilon_i) d\beta$ ,  $\alpha_i^{(m,n)} = \int_0^\infty \tilde{\beta}^m \exp(-\beta \varepsilon_i) / (1 + 2\beta R_{X_C X_C-i,i})^n d\beta$ ,  $B_\psi(\sigma_e^2) = \int_{-\infty}^\infty \frac{\psi^2(e)}{\sqrt{2\pi}\sigma_e} \exp(-\frac{e^2}{2\sigma_e^2}) de$ ,  $C_\psi(\sigma_e^2) = \frac{d}{d\sigma_e^2} E[\psi^2(e)]$ , and  $(r_{X_C X_C-i})^T$  is the  $i$ -th row of  $R_{X_C X_C}$ . For a given nonlinearity  $\psi(e)$ , the

above two integrals can be computed analytically or numerically.

Substituting (26–29) into (25) gives

$$\begin{aligned} \Xi(n+1) &= \Xi(n) - \mu A_\psi(\sigma_e^2) \mathbf{D}_\alpha R_{X_C X_C} \Xi(n) - \mu A_\psi(\sigma_e^2) \Xi(n) R_{X_C X_C} \mathbf{D}_\alpha + \mu^2 C_\psi(\sigma_e^2) \\ &\times \left[ \mathbf{S}^{(0)} \circ (R_{X_C X_C} \Xi(n) R_{X_C X_C}) + \mathbf{S}^{(1)} \mathbf{D}_\sigma + \mathbf{D}_\sigma \mathbf{S}^{(2)} + \mathbf{S}^{(3)} \circ (R_{X_C X_C} \Xi(n) R_{X_C X_C}) \right] + \mu^2 B_\psi(\sigma_e^2) \Gamma_\alpha, \end{aligned} \tag{30}$$

where  $\mathbf{D}_\sigma$  is a diagonal matrix with its  $i$ -th element  $[\mathbf{D}_\sigma]_{i,i} = (r_{X_C X_C-i})^T \Xi(n) r_{X_C X_C-i}$ ,  $[\mathbf{S}^{(k)}]_{i,j} = s_{ij}^{(k)}$  and  $[\Gamma_\alpha]_{ij} = \sum_{m=0}^\infty \alpha_{ij}^{(m)} (4^m) \binom{-\frac{3}{2} + m - 1}{m} R_{X_C X_C-i,j}^{(2m+1)}$ .

Let  $\Phi(n) = \mathbf{D}_\alpha^{-1/2} \Xi(n) \mathbf{D}_\alpha^{-1/2}$ , (30) can be further simplified to

$$\begin{aligned} \Phi(n+1) &= \Phi(n) - \mu A_\psi(\sigma_e^2) \mathbf{D}_\alpha^{1/2} R_{X_C X_C} \mathbf{D}_\alpha^{1/2} \Phi(n) - \mu A_\psi(\sigma_e^2) \Phi(n) \mathbf{D}_\alpha^{1/2} R_{X_C X_C} \mathbf{D}_\alpha^{1/2} + \mu^2 C_\psi(\sigma_e^2) \\ &\times \left\{ \left[ (\mathbf{S}^{(0)} + \mathbf{S}^{(3)}) \circ (\mathbf{D}_\alpha^{-1/2} R_{X_C X_C} \mathbf{D}_\alpha^{1/2} \Phi(n) \mathbf{D}_\alpha^{1/2} R_{X_C X_C} \mathbf{D}_\alpha^{-1/2}) \right] + \mathbf{D}_\alpha^{-1/2} \mathbf{S}^{(1)} \mathbf{D}_\alpha^{-1/2} \mathbf{D}_\sigma + \mathbf{D}_\sigma \mathbf{D}_\alpha^{-1/2} \mathbf{S}^{(2)} \mathbf{D}_\alpha^{-1/2} \right\} \\ &+ \mu^2 B_\psi(\sigma_e^2) \mathbf{D}_R^{-1/2} \Gamma_\alpha \mathbf{D}_R^{-1/2}, \end{aligned} \tag{31}$$

where  $[\mathbf{D}_\sigma]_{i,i} = (\mathbf{D}_\alpha^{1/2} r_{X_C X_C-i})^T \Phi(n) (\mathbf{D}_\alpha^{1/2} r_{X_C X_C-i})$ .

Since  $\mathbf{D}_\alpha^{1/2} R_{X_C X_C} \mathbf{D}_\alpha^{1/2}$  is symmetric, it can be diagonalized as  $\mathbf{U}_{X_D} \Lambda_{X_D} \mathbf{U}_{X_D}^T$ . Again, let  $\Psi(n) = \mathbf{U}_{X_D}^T \Phi(n) \mathbf{U}_{X_D}$ , (31) yields

$$\begin{aligned} \Psi(n+1) &= \Psi(n) - \mu A_\psi(\sigma_e^2) \Lambda_{X_D} \Psi(n) - \mu A_\psi(\sigma_e^2) \Psi(n) \Lambda_{X_D} + \mu^2 C_\psi(\sigma_e^2) \\ &\left\{ \mathbf{U}_{X_D}^T \left[ (\mathbf{S}^{(0)} + \mathbf{S}^{(3)}) \circ (\mathbf{D}_\alpha^{-1} \mathbf{U}_{X_D} \Lambda_{X_D} \Psi(n) \Lambda_{X_D} \mathbf{U}_{X_D}^T \mathbf{D}_\alpha^{-1}) \right] \mathbf{U}_{X_D} + \mathbf{U}_{X_D}^T \mathbf{D}_\alpha^{-1/2} \mathbf{S}^{(1)} \mathbf{D}_\alpha^{-1/2} \mathbf{D}_\sigma \mathbf{U}_{X_D} + \mathbf{U}_{X_D}^T \mathbf{D}_\sigma \mathbf{D}_\alpha^{-1/2} \mathbf{S}^{(2)} \mathbf{D}_\alpha^{-1/2} \mathbf{U}_{X_D} \right\} \\ &+ \mu^2 B_\psi(\sigma_e^2) \mathbf{U}_{X_D}^T \mathbf{D}_\alpha^{-1/2} \Gamma_\alpha \mathbf{D}_\alpha^{-1/2} \mathbf{U}_{X_D}. \end{aligned} \tag{32}$$

Since  $[\mathbf{D}_\sigma]_{i,i} = (\mathbf{D}_\alpha^{1/2} r_{X_C X_C-i})^T \mathbf{U}_{X_D} \Psi(n) \mathbf{U}_{X_D}^T (\mathbf{D}_\alpha^{1/2} r_{X_C X_C-i})$  is a scalar, after taking the  $\text{vec}(\cdot)$  operation we have the following:

where the  $[(i-1)L+i]$ -th row of  $\Delta$  is equal to  $\Delta_i$  and zero elsewhere. Let  $\Theta(n) = \text{vec}(\Psi(n))$ . (32) can be rewritten as

$$\begin{aligned} [\mathbf{D}_\sigma]_{i,i} &= \left( \mathbf{D}_\alpha^{1/2} r_{X_C X_C-i} \right)^T \otimes \left( \mathbf{D}_\alpha^{1/2} r_{X_C X_C-i} \right)^T (\mathbf{U}_{X_D} \otimes \mathbf{U}_{X_D}) \text{vec}(\Psi(n)) \\ &= (\Delta_i) \cdot \text{vec}(\Psi(n)). \end{aligned} \tag{34}$$

Hence,

$$\begin{aligned} \text{vec}(\mathbf{D}_\sigma) &= \left[ (\mathbf{D}_\sigma)_{1,1}, 0, \dots, 0, 0, (\mathbf{D}_\sigma)_{2,2}, 0, \dots, 0, \dots, 0, \dots, 0, (\mathbf{D}_\sigma)_{L,L} \right]^T \\ &= \Delta \cdot \text{vec}(\Psi(n)), \end{aligned} \tag{33}$$

where  $\Gamma_1(n) = \mathbf{I} - \mu A_\psi(\sigma_e^2) (\mathbf{I} \otimes \Lambda_{X_D}) - \mu A_\psi(\sigma_e^2) (\Lambda_{X_D} \otimes \mathbf{I}) + \mu^2 C_\psi(\sigma_e^2) \{ (\mathbf{U}_{X_D}^T \otimes \mathbf{U}_{X_D}^T) (\mathbf{S}^{(0)+(3)}) [(\mathbf{D}_\alpha^{-1} \mathbf{U}_{X_D}) \otimes (\mathbf{D}_\alpha^{-1} \mathbf{U}_{X_D})] \cdot (\Lambda_{X_D} \otimes \Lambda_{X_D}) + (\mathbf{U}_{X_D}^T \otimes \mathbf{U}_{X_D}^T) [\mathbf{I} \otimes (\mathbf{D}_\alpha^{-1/2} \mathbf{S}_D^{(1)} \mathbf{D}_\alpha^{-1/2}) + (\mathbf{D}_\alpha^{-1/2} \mathbf{S}_D^{(2)T} \mathbf{D}_\alpha^{-1/2}) \otimes \mathbf{I}] \Delta \}$ ,  $\Gamma_2(n) = \mu^2 B_\psi(\sigma_e^2) \text{vec}(\mathbf{U}_{X_D}^T \mathbf{D}_\alpha^{-1/2} \Gamma_\alpha \mathbf{D}_\alpha^{-1/2} \mathbf{U}_{X_D})$ , and  $\mathbf{S}^{(0)+(3)} = \text{diag}(\mathbf{S}^{(0)} + \mathbf{S}^{(3)})$ .

The algorithm will converge if  $\|I - \Gamma_1(n)\|_2 < 1$ . Using triangular inequality, we have

$$\begin{aligned} \|I - \Gamma_1(n)\|_2 &= \|\mu A_\psi(\sigma_e^2)(I \otimes \Lambda_{X_D}) + \mu A_\psi(\sigma_e^2)(\Lambda_{X_D} \otimes I) - \mu^2 C_\psi(\sigma_e^2) \{ (U_{X_D}^T \otimes U_{X_D}^T) S^{(0)+(3)} [(D_\alpha^{-1} U_{X_D}) \otimes (D_\alpha^{-1} U_{X_D})] \\ &\quad \cdot (\Lambda_{X_D} \otimes \Lambda_{X_D}) + (U_{X_D}^T \otimes U_{X_D}^T) [I \otimes (D_\alpha^{-1} S_D^{(1)} D_\alpha^{-1}) + (D_\alpha^{-1} S_D^{(2)T} D_\alpha^{-1}) \otimes I] \Delta \} \|_2 \\ &\leq 2\mu A_\psi(\sigma_e^2) \lambda'_{\max} + \mu^2 C_\psi(\sigma_e^2) (\lambda'_{\max})^2 \alpha' < 1, \end{aligned}$$

where  $\alpha' = s_{\max}^{(0)+(3)} + (\lambda'_{\max})^{-2} (s_{\max}^{(2)} + s_{\max}^{(1)}) \Delta_{\max}$ .

Therefore the algorithm converges if

$$\begin{aligned} &\left| 1 - 2\mu A_\psi(\sigma_e^2) \lambda'_{\max} + \mu^2 C_\psi(\sigma_e^2) (\lambda'_{\max})^2 \alpha' \right| < 1 \\ &\Leftrightarrow \left| \left( 1 - \mu (r'_0)^{-1} \right) \left( 1 - \mu (r'_1)^{-1} \right) \right| < 1 \end{aligned}$$

where  $r'_{0,1} = \lambda'_{\max} (A_\psi(\sigma_e^2) \pm \sqrt{A_\psi^2(\sigma_e^2) - C_\psi(\sigma_e^2) \alpha'})$ . Hence, the maximum possible step size for mean square convergence is

$$\mu_{\max} (r'_1)^{-1} = \left[ \lambda'_{\max} (A_\psi(\sigma_e^2) + \sqrt{A_\psi^2(\sigma_e^2) - C_\psi(\sigma_e^2) \alpha'}) \right]^{-1}.$$

If the algorithm converges, we have from (34)

$$\Theta(\infty) = (I - \Gamma_1(\infty))^{-1} \Gamma_2(\infty).$$

The excess mean square error (EMSE) at time instant  $n$  is  $\text{EMSE}(n) = \text{Tr}(\Xi(n) \mathbf{R}_{X_C X_C}) = \text{Tr}(\Psi(n) \Lambda_{X_D})$ . Hence

$$\text{EMSE}(\infty) = \text{Tr}(\text{vec}^{-1}(\Theta(\infty)) \Lambda_{X_D}), \tag{35}$$

where  $\text{vec}^{-1}(\cdot)$  is the inverse  $\text{vec}(\cdot)$  operator. (35) is rather difficult to further simplify in general. We shall analyze the cases with small step size and uncorrelated transform output below.

*Small step sizes* If  $\mu$  is small enough, then we can drop the terms involving  $\Psi(n)$  and  $\mu^2$ , and (32) becomes

$$\begin{aligned} \Psi(n+1) &\approx \Psi(n) - \mu A_\psi(\sigma_e^2) \Lambda_{X_D} \Psi(n) \\ &\quad - \mu A_\psi(\sigma_e^2) \Psi(n) \Lambda_{X_D} \\ &\quad + \mu^2 B_\psi(\sigma_e^2) \hat{\Gamma}_{\text{UD-}\alpha}, \end{aligned} \tag{36}$$

where  $\hat{\Gamma}_{\text{UD-}\alpha} = U_{X_D}^T D_\alpha^{-1/2} \Gamma_\alpha D_\alpha^{-1/2} U_{X_D}$ . Let  $\mathbf{D}_{\text{diag}}(\mathbf{K})$  be an operator which retains only the diagonal values of a square

matrix  $\mathbf{K}$  and setting the others to zero. When the algorithm converges, we have

$$\mathbf{D}_{\text{diag}}(\Psi(n)) = \mu \frac{B_\psi(\sigma_e^2)}{2A_\psi(\sigma_e^2)} \mathbf{D}_{\text{diag}}(\Lambda_{X_D}^{-1} \hat{\Gamma}_{\text{UD-}\alpha}), \tag{37}$$

Hence, (35) reduces to

$$\begin{aligned} \text{EMSE}(\infty) &= \frac{\mu B_\psi(\sigma_e^2(\infty))}{2A_\psi(\sigma_e^2(\infty))} \text{Tr}(\hat{\Gamma}_{\text{UD-}\alpha}) \\ &= \frac{\mu B_\psi(\sigma_e^2(\infty))}{2A_\psi(\sigma_e^2(\infty))} \text{Tr}(\Gamma_\alpha \mathbf{D}_\alpha^{-1}) \end{aligned} \tag{38}$$

*Uncorrelated Case* If  $\mathbf{R}_{X_C X_C}$  is diagonal, then it can be shown that  $I_{1,i,i} = 2(v_i \lambda_i)^2 \tilde{\alpha}_i^{(-5/2)}$ ,  $I_{2,i,i} = \lambda_i \tilde{\alpha}_i^{(-3/2)}$ , and zero otherwise, where  $\tilde{\alpha}_i^{(k)} = \int_0^\infty \int_0^\infty \exp(-(\beta_1 + \beta_2)\varepsilon_i) g_i(\beta_1 + \beta_2)^{-k} d\beta_2 d\beta_1$  and  $\lambda_i = R_{X_C X_C -i,i}$ . Hence, (30) reduces to

$$\begin{aligned} \Xi(n+1) &= \Xi(n) - \mu A_\psi(\sigma_e^2) \mathbf{D}_\alpha \mathbf{R}_{X_C X_C} \Xi(n) \\ &\quad - \mu A_\psi(\sigma_e^2) \Xi(n) \mathbf{R}_{X_C X_C} \mathbf{D}_\alpha \\ &\quad + 2\mu^2 C_\psi(\sigma_e^2) \mathbf{R}_{X_C X_C} \mathbf{D}_\alpha^{-s/2} \text{diag}(\Xi(n)) \mathbf{R}_{X_C X_C} \\ &\quad + \mu^2 B_\psi(\sigma_e^2) \mathbf{R}_{X_C X_C} \mathbf{D}_\alpha^{-3/2} \end{aligned}$$

which is equivalent to the following set of scalar equations:

$$\begin{aligned} \Xi_{i,i}(n+1) &= \Xi_{i,i}(n) - 2\mu A_\psi(\sigma_e^2) \alpha_i \lambda_i \Xi_{i,i}(n) \\ &\quad + 2\mu^2 C_\psi(\sigma_e^2) \lambda_i^2 \tilde{\alpha}_i^{(-5/2)} \Xi_{i,i}(n) \\ &\quad + \mu^2 B_\psi(\sigma_e^2) \lambda_i \tilde{\alpha}_i^{(-3/2)}. \end{aligned} \tag{39}$$

Assuming the difference equation converges, the corresponding steady state value of  $\Xi_{i,i}(\infty)$  can be obtained from (39) as

$$\Xi_{i,i}(\infty) = \frac{\mu B_\psi(\sigma_e^2(\infty)) \tilde{\alpha}_i^{(-3/2)}}{2(\alpha_i A_\psi(\sigma_e^2(\infty)) - \mu C_\psi(\sigma_e^2(\infty)) \lambda_i \tilde{\alpha}_i^{(-5/2)})}. \tag{40}$$



The EMSE is then given by

$$\text{EMSE}(\infty) = \text{Tr}(\mathbf{R}_{X_c X_c} \Xi(\infty)) = \sum_{i=1}^L \lambda_i \Xi_{i,i}(\infty). \quad (41)$$

Remarks

(R-A3): TDNLMS algorithm

In this case,  $A_\psi(\sigma_e^2) = C_\psi(\sigma_e^2) = 1$ ,  $B_\psi(\sigma_e^2) = \sigma_e^2$ . Since  $\sigma_e^2(n) = \text{EMSE}_{\text{TDNLMS}}(n) + \sigma_g^2$ , the EMSE from the small step size result in (38) is

$$\text{EMSE}_{\text{TDNLMS}}(\infty) = \frac{\frac{1}{2} \mu \sigma_g^2 \phi_{\text{TDNLMS}}}{1 - \frac{1}{2} \mu \phi_{\text{TDNLMS}}}, \quad (42)$$

where  $\phi_{\text{TDNLMS}} = \text{Tr}(\mathbf{I} \alpha \mathbf{D}_\alpha^{-1})$ . Particularly, for the LMS algorithm,  $\mathbf{C} = \mathbf{I}$ ,  $\mathbf{D}_\alpha = \mathbf{I}$ , and  $\mathbf{I} \alpha = \mathbf{R}_{XX}$ . (42) will reduce to

$$\text{EMSE}_{\text{LMS}}(\infty) = \frac{\frac{1}{2} \mu \sigma_g^2 \phi_{\text{LMS}}}{1 - \frac{1}{2} \mu \phi_{\text{LMS}}}, \quad \phi_{\text{LMS}} = \text{Tr}(\mathbf{R}_{XX}), \quad (43)$$

which agrees with the conventional result for the LMS algorithm.

For the uncorrelated case,

$$\text{EMSE}_{\text{TDNLMS}_U}(\infty) = \frac{\frac{1}{2} \mu \sigma_g^2 \phi_{\text{TDNLMS}_U}}{1 - \frac{1}{2} \mu \phi_{\text{TDNLMS}_U}}, \quad (44)$$

where  $\phi_{\text{TDNLMS}_U} = \sum_{i=1}^L \frac{\tilde{\alpha}_i^{(-3/2)} \lambda_i}{\tilde{\alpha}_i^{(-5/2)}}$ . For perfect power estimation,  $\varepsilon_i = \sigma_{X_c}^2$  and  $\alpha_\varepsilon = 0$ ,  $\alpha_i = \varepsilon_i^{-1} \tilde{\alpha}_i^{(-3/2)} = \tilde{\alpha}_i^{(-5/2)} = \varepsilon_i^{-2}$  and  $\mu \phi_{\text{TDNLMS}_U} = \sum_{i=1}^L \frac{(\mu/\varepsilon_i) \lambda_i}{1 - (\mu/\varepsilon_i) \lambda_i} = \frac{\mu L}{1 - \mu}$ , which reduces to the classical result of the LMS algorithm with an exact power normalized step size ( $\mu/\varepsilon_i$ ). For stability, EMSE( $\infty$ ) should be a finite quality and it gives the following two conditions on  $\mu$  for stability:

$$0 < \mu < \alpha_i / \left( \lambda_i \tilde{\alpha}_i^{(-5/2)} \right) \text{ and } \sum_{i=1}^L \frac{\mu \alpha_i \tilde{\lambda}_i^{(-4/2)}}{\alpha_i - \mu \lambda_i \tilde{\alpha}_i^{(-6/2)}} \leq 2.$$

Following the approach in [30], one gets the approximate stepsize bound as

$$\mu_B = \frac{2}{\sum_{i=1}^L \lambda_i (2d_i + c_i)}, \quad (45)$$

$$\text{where } c_i = \frac{1}{\alpha_i} \tilde{\alpha}_i^{(-3/2)} \text{ and } d_i = \frac{1}{\alpha_i} \mu \lambda_i \tilde{\alpha}_i^{(-5/2)}.$$

(R-A4): The TDNLMS algorithm with general nonlinearity and the TDNLMM algorithm

For the TDNLMS algorithms with general nonlinearity, (38) or (42) is a nonlinear equation in EMSE( $\infty$ ) since  $\sigma_e^2(\infty) = \text{EMSE}(\infty) + \sigma_g^2$ , general solution is difficult to obtain. In contrast, for the TDNLMM algorithm using MH

nonlinearity and ATS,  $A_{\text{MH}}(\sigma_e^2) \approx \text{erf}\left(\frac{k_\varepsilon}{\sqrt{2}}\right) - \frac{2k_\varepsilon}{\sqrt{2\pi}} \exp\left(-\frac{k_\varepsilon^2}{2}\right) = A_c$ ,  $B_{\text{MH}}(\sigma_e^2) \approx \left(\text{erf}\left(\frac{k_\varepsilon}{\sqrt{2}}\right) - \frac{2k_\varepsilon}{\sqrt{2\pi}} \exp\left(-\frac{k_\varepsilon^2}{2}\right)\right) \sigma_e^2 = A_c \sigma_e^2$ ,  $C_{\text{MH}}(\sigma_e^2) \approx A_c - \left(\frac{k_\varepsilon^3}{\sqrt{2\pi}}\right) \exp\left(-\frac{k_\varepsilon^2}{2}\right) \cdot \text{EMSE}_{\text{TDNLMM}}(\infty) \approx \frac{1}{2} \mu \sigma_e^2 \text{Tr}(\mathbf{I} \alpha \mathbf{D}_\alpha^{-1})$ . Solving for EMSE<sub>TDNLMM</sub>( $\infty$ ) gives

$$\text{EMSE}_{\text{TDNLMM}}(\infty) \approx \frac{\frac{1}{2} \mu \sigma_g^2 \phi_{\text{TDNLMM}}}{1 - \frac{1}{2} \mu \phi_{\text{TDNLMM}}}, \quad (46)$$

where  $\phi_{\text{TDNLMM}} = \phi_{\text{TDNLMS}}$ .

For the LMM algorithm with MH nonlinearity,  $\mathbf{C} = \mathbf{I}$ ,  $\mathbf{D}_R = \mathbf{I}$ , and  $\mathbf{I} \alpha = \mathbf{R}_{XX}$ . (41) will reduce to

$$\text{EMSE}_{\text{LMM}}(\infty) \approx \frac{\frac{1}{2} \mu \sigma_g^2 \phi_{\text{LMM}}}{1 - \frac{1}{2} \mu \phi_{\text{LMM}}}, \quad \phi_{\text{LMM}} = \phi_{\text{LMS}}, \quad (47)$$

which agrees with the result in [16] and is close to their LMS counterpart.  $B_\psi(\sigma_e^2)$  and  $C_\psi(\sigma_e^2)$  for some related algorithms are summarized in Table 1.

### 3.2 Convergence Behaviors in CG Noise

We now study the mean and mean square behaviors of the TDNLMS algorithm with general nonlinearity and particularly the TDNLMS and TDNLMM algorithms in CG noise environment. For most M-estimate functions which suppress outliers with large amplitude, the convergence rate will only be slightly impaired after employing ATS. We shall employ an extension of the Price’s theorem to Gaussian mixtures [18]. This extension was employed in the analysis of the LMS and NLMS algorithms with MH nonlinearity and CG noise in [16]. Similar techniques were also employed in analyzing the RLM and other related algorithms [15] for the MH nonlinearity. We shall show in the following that with the use of M-estimate function and ATS, the impulsive noise can be effectively suppressed and the EMSE is similar to the case where only Gaussian noise is present. On the other hand, the EMSE of the LMS-based algorithms will be substantially affected by the impulsive CG noise.

#### 3.2.1 Mean Behavior

Since  $\eta_o$  is now a CG noise as defined in (11), it is a Gaussian mixture consisting of two components  $\eta_{o,1}$  and  $\eta_{o,2}$ , each with zero mean and variance  $\sigma_1^2 = \sigma_g^2$  and  $\sigma_2^2 = \sigma_\Sigma^2$ , respectively. The occurrence probability of the impulsive noise is  $p_r$ . Accordingly,

$$\begin{aligned} E_{\{v, X, \eta_o\}}[f(\mathbf{X}(n), e(n))] &= (1 - p_r) E_{\{v, X, \eta_{o,1}\}}[f(\mathbf{X}(n), e(n))] \\ &\quad + p_r E_{\{v, X, \eta_{o,2}\}}[f(\mathbf{X}(n), e(n))], \end{aligned} \quad (48)$$

where  $f(\mathbf{X}(n), e(n))$  is an arbitrary quantity whose statistical average is to be evaluated. Since  $\mathbf{X}(n)$ ,  $\eta_{o-1}$ , and  $\eta_{o-2}$  are Gaussian distributed, each of the expectation on the right hand side can be evaluated using the Price's theorem. Consequently, the results in section A can be carried forward to the CG noise case by firstly changing the noise power respectively to  $\sigma_g^2$  and  $\sigma_\Sigma^2$ , and then combining the two results using (48).

Recall the relation of the mean weight-error vector in (13):

$$E[\mathbf{v}(n+1)] = E[\mathbf{v}(n)] - \mu \mathbf{H}', \tag{49}$$

where  $\mathbf{H}' = E_{\{\mathbf{v}, \mathbf{X}_C, \eta_o\}}[\Lambda_C^{-1} \psi(e(n)) \mathbf{X}_C(n)] = (1 - p_r) \mathbf{H}'_1 + p_r \mathbf{H}'_2$ ,  $\mathbf{H}'_1$  and  $\mathbf{H}'_2$  are respectively the expectation of the term inside the brackets above with respect to  $\{\mathbf{v}, \mathbf{X}_C, \eta_{o-1}\}$ , and  $\{\mathbf{v}, \mathbf{X}_C, \eta_{o-2}\}$ . From (16) and (17), we have  $\mathbf{H}'_i \approx \bar{\psi}'(\sigma_{e_i}^2) \mathbf{D}_\alpha \mathbf{R}_{X_C X_C} \mathbf{v}(n)$ ,  $i = 1, 2$ , where  $\sigma_{e_1}^2(n) = \sigma_{e_g}^2(n) = E[\mathbf{v}^T(n) \mathbf{R}_{X_C X_C} \mathbf{v}(n)] + \sigma_g^2$ ,  $\sigma_{e_2}^2(n) = \sigma_{e_\Sigma}^2(n) = E[\mathbf{v}^T(n) \mathbf{R}_{X_C X_C} \mathbf{v}(n)] + \sigma_\Sigma^2$ . Hence

$$\mathbf{H}' \approx \tilde{A}_\psi(n) \mathbf{D}_\alpha \mathbf{R}_{X_C X_C} \mathbf{v}(n), \tag{50}$$

where  $\tilde{A}_\psi(n) = (1 - p_r) \bar{\psi}'(\sigma_{e_g}^2(n)) + p_r \bar{\psi}'(\sigma_{e_\Sigma}^2(n))$ . Substituting (50) into (49) and using the transformation  $\mathbf{V}_D(n) = \mathbf{U}_{X_D}^T \mathbf{D}_\alpha^{-1/2} \mathbf{v}(n)$ , one gets

$$E[\mathbf{V}_D(n+1)] = (\mathbf{I} - \mu \tilde{A}_\psi(n) \Lambda_{X_D}) E[\mathbf{V}_D(n)]. \tag{51}$$

For simplicity, we have replaced the approximate symbol by the equality symbol. This yields the same form as (18), except for  $A_\psi(n)$ . Similar argument regarding the mean convergence in section 3.1 also applies to (51). A sufficient condition for the algorithm to converge is  $|1 - \mu \tilde{A}_\psi(n) \lambda'_i| < 1$ , for all  $i$ . If  $\bar{\psi}'(\sigma_e^2)$  is upper bounded and so is  $\tilde{A}_\psi(n)$ , say by

$\tilde{A}_{\psi\_max}$ , then following the argument in part 3.1, the following conservative maximum step size is obtained:

$$\mu_{max} < 2 / (\tilde{A}_{\psi\_max} \lambda'_{max}).$$

Remarks:

(R-B1): TDNLMS algorithm

In this case,  $\tilde{A}_\psi(n) = 1$ . Compared with the Gaussian case, the convergence rate remains unchanged. All the conclusions in (R-A1) apply.

(R-B2): TDNLMS algorithm with general nonlinearity and TDNLMM algorithm:

For general nonlinearity without ATS, both  $\sigma_{e_g}^2$  and  $\sigma_{e_\Sigma}^2$  can be very large due to the large value of  $\sigma_{e_\Sigma}^2$  and the slow decay of the EMSE  $E[\mathbf{v}^T(n) \mathbf{R}_{X_C X_C} \mathbf{v}(n)]$ , as the gain  $\tilde{A}_\psi(n) = (1 - p_r) \bar{\psi}'(\sigma_{e_g}^2(n)) + p_r \bar{\psi}'(\sigma_{e_\Sigma}^2(n))$  can be very small initially. This leads to nonlinear adaptation and slow convergence. Near convergence,  $E[\mathbf{v}^T(n) \mathbf{R}_{X_C X_C} \mathbf{v}(n)]$  and  $\tilde{A}_\psi(n)$  will become stable. The convergence is exponential and the convergence rate of the  $i$ -th mode is approximately  $1 - \mu \tilde{A}_\psi(\infty) \lambda'_i$ , where  $\tilde{A}_\psi(\infty)$  is the steady state value of  $\tilde{A}_\psi(n)$ . Normally, the second term  $p_r \bar{\psi}'(\sigma_{e_\Sigma}^2(n))$  will be much smaller than the first one due to the clipping property of the nonlinearity and the large variance of the impulsive noise  $\sigma_\Sigma^2$ . For the TDNLMM algorithm with ATS, the degradation in convergence rate is not so serious since if  $\sigma_{e_g}^2 \ll \sigma_{e_\Sigma}^2$ ,  $\tilde{A}_{MH} \approx (1 - p_r) A_c$  is a constant close to one if  $p_r$  is not too large.

### 3.2.2 Mean Square Behavior

Using a similar approach, it can be shown that

$$\begin{aligned} \Psi(n+1) = & \Psi(n) - \mu \tilde{A}_\psi(n) \Lambda_{X_D} \Psi(n) - \mu \tilde{A}_\psi(n) \Psi(n) \Lambda_{X_D} \\ & + \mu^2 \tilde{C}_\psi(n) \left\{ \mathbf{U}_{X_D}^T \left[ (\mathbf{S}^{(0)} + \mathbf{S}^{(2)}) \circ (\mathbf{D}_\alpha^{-1} \mathbf{U}_{X_D} \Lambda_{X_D} \Psi(n) \Lambda_{X_D} \mathbf{U}_{X_D}^T \mathbf{D}_\alpha^{-1}) \right] \mathbf{U}_{X_D} + \mathbf{U}_{X_D}^T \mathbf{D}_\sigma \mathbf{D}_\alpha^{-1/2} \mathbf{S}^{(1)} \mathbf{D}_\alpha^{-1/2} \mathbf{U}_{X_D} \right. \\ & \left. + \mathbf{U}_{X_D}^T \mathbf{D}_\alpha^{-1/2} \mathbf{S}^{(3)} \mathbf{D}_\alpha^{-1/2} \mathbf{D}_\sigma \mathbf{U}_{X_D}^T \right. \\ & \left. + \mu^2 \tilde{B}_\psi(n) \mathbf{U}_{X_D}^T \mathbf{D}_\alpha^{-1/2} \mathbf{F}_\alpha \mathbf{D}_\alpha^{-1/2} \mathbf{U}_{X_D} \right\} \end{aligned} \tag{52}$$

where  $\tilde{C}_\psi(n) = (1 - p_r) C_\psi(\sigma_{e_g}^2(n)) + p_r C_\psi(\sigma_{e_\Sigma}^2(n))$  and  $\tilde{B}_\psi(n) = (1 - p_r) B_\psi(\sigma_{e_g}^2(n)) + p_r B_\psi(\sigma_{e_\Sigma}^2(n))$ .

Due to page limitation, we only summarize the result for the small step size case as:

$$\text{EMSE}(\infty) \approx \mu \frac{\tilde{B}_\psi(\infty)}{2 \tilde{A}_\psi(\infty)} \text{Tr}(\mathbf{F}_\alpha \mathbf{D}_\alpha^{-1}). \tag{53}$$

(R-B3): TDNLMS algorithm

In these cases,  $\tilde{A}_\psi(n) = \tilde{C}_\psi(n) = 1$ , and  $\tilde{B}_\psi(n) = (1 - p_r)(\sigma_{\text{excess}}^2(n) + \sigma_g^2) + p_r(\sigma_{\text{excess}}^2(n) + \sigma_\Sigma^2) = \sigma_{\text{excess}}^2(n) + \sigma_{\eta_o}^2$ , where  $\sigma_{\eta_o}^2 = (1 - p_r)\sigma_g^2 + p_r\sigma_\Sigma^2$ ,  $\sigma_{\text{excess}}^2(n) = E[\mathbf{v}^T(n) \mathbf{R}_{X_C X_C} \mathbf{v}(n)]$  is the EMSE. Hence

$$\begin{aligned} \text{EMSE}_{\text{TDNLMS}}(\infty) = & \sigma_{\text{excess}}^2(\infty) \\ \approx & \frac{1}{2} \mu \left( \sigma_{\text{excess}}^2(\infty) + \sigma_{\eta_o}^2 \text{Tr}(\mathbf{F}_\alpha \mathbf{D}_\alpha^{-1}) \right), \end{aligned}$$

which gives

$$\text{EMSE}_{\text{TDNLMS}}(\infty) \approx \frac{\frac{1}{2}\mu\sigma_{\eta_0}^2\phi_{\text{TDNLMS}}}{1 - \frac{1}{2}\mu\phi_{\text{TDNLMS}}}, \quad (54)$$

It can be seen that the EMSE will be considerably increased over the Gaussian case by  $p_r\mu\sigma_w^2\phi_{\text{TDNLMS}}/(1 - \mu\phi_{\text{TDNLMS}})$ , which increases with the probability of occurrence of the impulses and the difference in power between the impulsive and Gaussian components.

For the TDNLMM algorithm with MH nonlinearity and ATS,  $\tilde{A}_{\text{MH}}(\infty) \approx (1 - p_r)A_c$ ,  $\tilde{B}_{\text{MH}}(\infty) \approx (1 - p_r)\sigma_{e_g}^2 \text{erf}\left(\frac{k_g}{\sqrt{2}}\right) = (1 - p_r)\sigma_{e_g}^2 A_c$ ,  $\tilde{C}_{\text{MH}}(\infty) \approx \tilde{A}_{\text{MH}}(\infty) - (1 - p_r)\left(\frac{k_g^2}{\sqrt{2\pi}}\right) \exp\left(-\frac{k_g^2}{2}\right)$ .

$$\text{EMSE}_{\text{TDNLMM}}(\infty) \approx \frac{\frac{1}{2}\mu\sigma_g^2\phi_{\text{TDNLMS}}}{1 - \frac{1}{2}\mu\phi_{\text{TDNLMS}}}, \quad (55)$$

which is identical to the case with Gaussian noise only. This illustrates the robustness of the TDNLMM algorithm to impulsive noise.

For the LMM algorithm with the MH nonlinearity,  $D_R = I$ , and  $\Gamma_\alpha = R_{XX}$ . (55) will reduce to

$$\text{EMSE}_{\text{LMM}}(\infty) \approx \frac{\mu\sigma_g^2\text{Tr}(R_{XX})}{2 - \mu\text{Tr}(R_{XX})}, \quad (56)$$

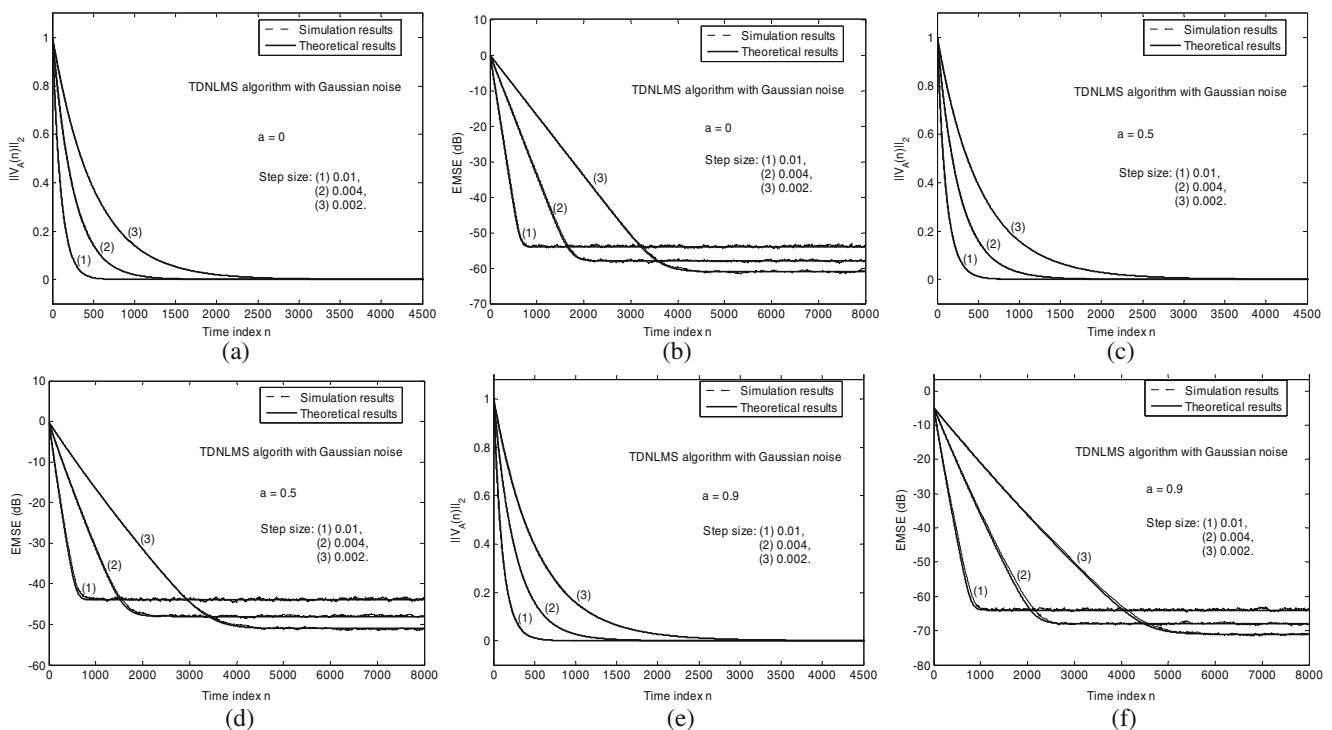
which is also similar to its conventional LMS counterpart when the additive noise is Gaussian. This illustrates the robustness of the M-estimation based algorithms to impulsive noise.

### 4 Simulation Results

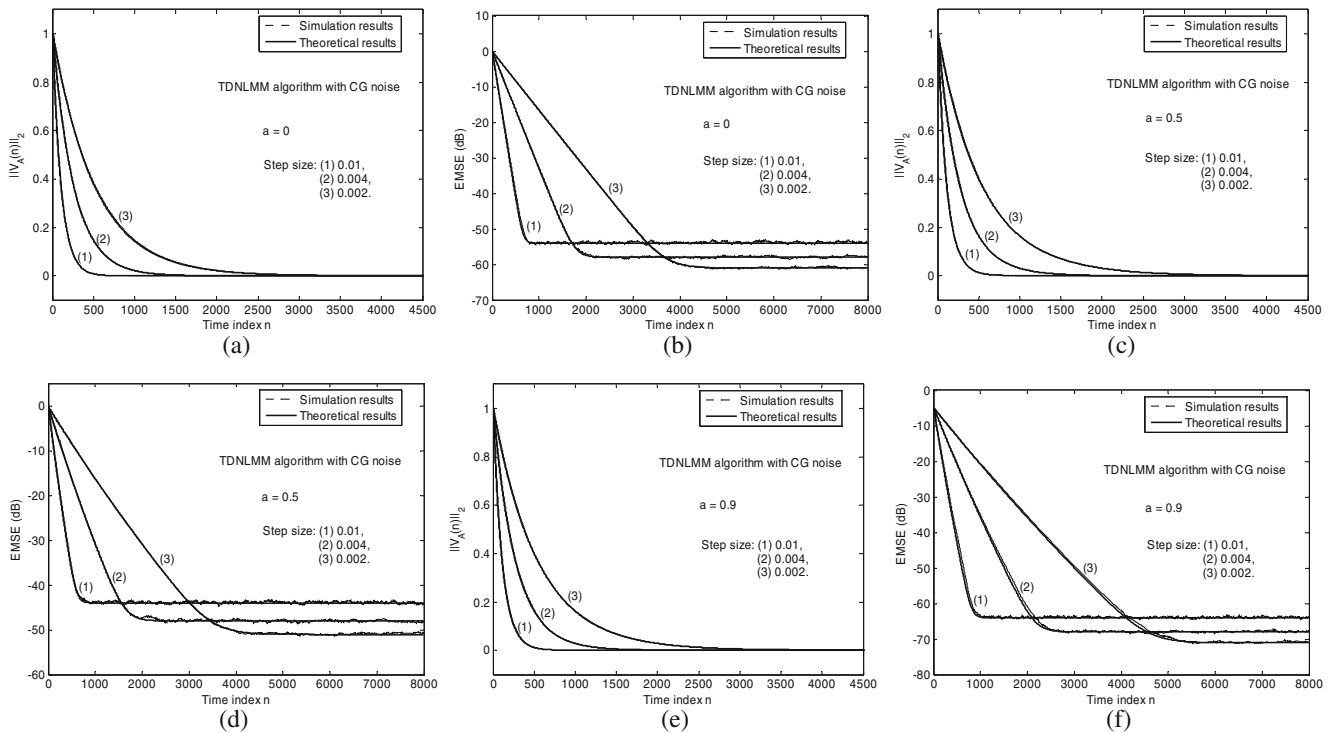
In this section, computer simulations on the system identification problem shown in Fig. 1 are conducted to evaluate the analytical results for the TDNLMS and TDNLMM algorithms obtained in section 3. The unknown system  $W^*$  is a FIR filter with  $L=8$ . Its coefficients are randomly generated and normalized to unit energy. The input signal  $x(n)$  is generated as a first-order AR process

$$x(n) = ax(n - 1) + v(n), \quad (57)$$

where  $v(n)$  is a white Gaussian noise sequence with zero mean and variance  $\sigma_v^2$ .  $0 < a < 1$  is the correlation coefficient and in our experiment it is set to be 0, 0.5 and 0.9. DCT is employed due to its wide usage and efficiency in practice. The simulation results are averaged over  $K=200$  independent runs. Only impulses in the desired signal are considered. The locations of impulses are not fixed for each independent run and their amplitudes are varying. For the CG impulsive noise, we test  $p_r=0.005, 0.01$  and  $0.02$ ;  $r_{im}=50, 100$  and



**Figure 3** The mean and mean square convergence performance of the TDNLMS algorithm with Gaussian noise: **a, b:**  $\alpha=0$ ,  $\sigma_g^2 = 10^{-4}$ , **c, d:**  $\alpha=0.5$ ,  $\sigma_g^2 = 10^{-3}$ , **e, f:**  $\alpha=0.9$ ,  $\sigma_g^2 = 10^{-5}$ ; Three step sizes are used: (1)  $\mu=0.01$ , (2)  $\mu=0.004$ , (3)  $\mu=0.002$ .



**Figure 4** The mean and mean square convergence performance of the TDNLMM algorithm with CG noise: **a**, **b**:  $a=0$ ,  $\sigma_g^2 = 10^{-4}$ ,  $r_{im}=200$ ,  $p_r=0.02$ . **c**, **d**:  $a=0.5$ ,  $\sigma_g^2 = 10^{-3}$ ,  $r_{im}=100$ ,  $p_r=0.01$ . **(e)**, **(f)**:  $a=0.9$ ,

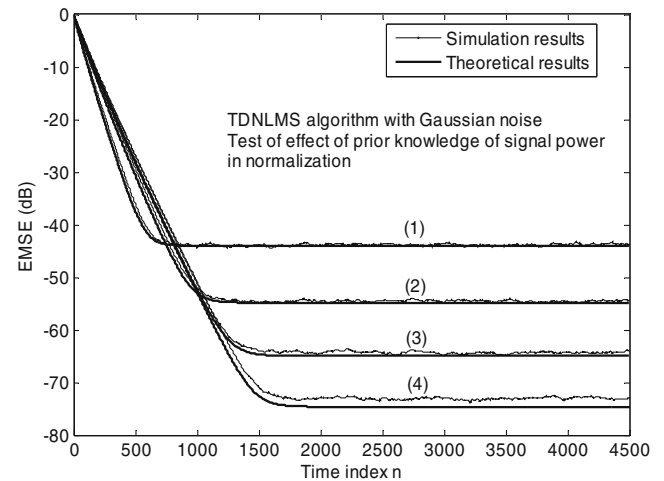
$\sigma_g^2 = 10^{-5}$ ,  $r_{im}=100$ ,  $p_r=0.005$ ; Three step sizes are used: (1)  $\mu=0.01$ , (2)  $\mu=0.004$ , (3)  $\mu=0.002$ .

200.  $\lambda_\sigma = 0.95$ ,  $N_w=9$ ,  $k_\xi = 2.576$ . For mean convergence, the norm of the mean square weight-error vector

$$\|v_A(n)\|_2 = \sqrt{\sum_{i=1}^L \left[ \frac{1}{K} \sum_{j=1}^K v_i^{(j)}(n) \right]^2}, i = 1, \dots, L, j = 1, \dots, K,$$

is used as the performance measure.  $EMSE(n) = \text{Tr}(\Xi(n)\mathbf{R}_{XX}) = \text{Tr}(\Phi(n)\Lambda)$  is adopted as the mean square performance measure. The integrals  $\alpha_i$  defined in (A-9b),  $\alpha_i^{(k)}$  in (B-9) and  $\alpha_i^{(m,n)}$  in (B-10) are evaluated numerically [28]. Figures 3 and 4 respectively depict the mean and mean square performance of the TDNLMS algorithm in Gaussian noise and the TDNLMM algorithm with CG noise. The theoretical results are computed respectively from (19), (30) and (51), (52). Different values of  $a$ ,  $\mu$ ,  $\sigma_g^2$ ,  $r_{im}$  and  $p_r$  are used as specified in respective figure caption. All these figures show a satisfactory agreement between the theoretical and simulation results. Since the results for the TDNLMM algorithm in Gaussian noise is similar to those in CG noise, they are omitted to save space. For the TDNLMS algorithm in CG noise, the mean weight vector can be considerably affected by the impulsive noise and the independent assumption in assumption 3 becomes less accurate. Since this case is of little interest, the simulation result is also omitted.

To study the effect of the recursive power estimation of the signal components in the normalization part of the TDNLMS algorithm,  $\varepsilon_i(n) = (1 - \alpha_\varepsilon)\sigma_{X_{C-i}}^2 + \alpha_\varepsilon X_{C,i}^2(n)$  is used, which allows us to approximately model the effect of prior knowledge of the signal power on the algorithms. This is valid when the recursive estimation of the signal power converges. The value of  $\sigma_{X_{C-i}}^2$  can be obtained from



**Figure 5** Test of the effect of a priori knowledge of signal power in normalization part of the TDNLMS algorithm.  $\mu=0.01$ . (1)  $\alpha_\varepsilon = 0.01$ ,  $\sigma_g^2 = 10^{-3}$ , (2)  $\alpha_\varepsilon = 0.25$ ,  $\sigma_g^2 = 10^{-4}$ , (3)  $\alpha_\varepsilon = 0.5$ ,  $\sigma_g^2 = 10^{-5}$ , (4)  $\alpha_\varepsilon = 0.75$ ,  $\sigma_g^2 = 10^{-6}$ .

calculation or offline estimation. In our experiment, it is derived from (57) plus DCT operation and known parameters. Figure 5 illustrates that with the increase of  $\alpha_\varepsilon$ , the estimation accuracy slightly deteriorates. This verifies the efficiency of power normalization in TDNLMS algorithm.

### 5 Conclusions

The convergence performance of the TDNLMS algorithm and its TDNLMM generalizations with Gaussian inputs and additive Gaussian and contaminated Gaussian noises is presented. Difference equations describing the mean and mean square convergence behaviors for these algorithms are derived. The analytical results reveal the advantages of the TDNLMM algorithms in impulsive noise environment, and they are shown to be in good agreement with computer simulation results.

**Open Access** This article is distributed under the terms of the Creative Commons Attribution Noncommercial License which per-

mits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

### Appendix A

In this Appendix,  $\mathbf{H}_1 = E_{\{X_C, \eta_g\}} [A_C^{-1} \psi(e(n)X_C(n)) | \mathbf{v}]$  is evaluated. For notational convenience, we shall drop the subscript  $C$  in  $X_C$ . An approach similar to [15, 29] is employed to evaluate this expectation. As  $\eta_g(n)$  and  $x(n)$  are assumed to be statistically independent, and  $\mathbf{X}$  are jointly Gaussian with covariance matrix  $\mathbf{R}_{X_C X_C}$ , the  $i$ -th element of the vector  $\mathbf{H}_1$  is

$$H_{1,i} = C_R \int \int_{L+1 \text{ fold}} \frac{\psi(e)X_i}{\varepsilon_i + \alpha_\varepsilon X_i^2} \exp\left(-\frac{1}{2} \mathbf{X}^T \mathbf{R}_{X_C X_C}^{-1} \mathbf{X}\right) f_{\eta_g}(\eta_g) d\eta_g d\mathbf{X}, \tag{A-1}$$

where  $C_R = (2\pi)^{-L/2} |\mathbf{R}_{X_C X_C}|^{-1/2}$  and  $f_{\eta_g}(\eta_g)$  is the PDF of the Gaussian noise  $\eta_g$ .  $|\cdot|$  denotes the determinant of a matrix. Similar to [17], let us consider the integral

$$F_i(\beta) = C_R \int \int_{L+1 \text{ fold}} \frac{\psi(e)X_i \exp(-\beta(\varepsilon_i + \alpha_\varepsilon X_i^2))}{\varepsilon_i + \alpha_\varepsilon X_i^2} \cdot \exp\left(-\frac{1}{2} \mathbf{X}^T \mathbf{R}_{X_C X_C}^{-1} \mathbf{X}\right) f_{\eta_g}(\eta_g) d\eta_g d\mathbf{X} \tag{A-2}$$

It can be seen that  $H_{1,i} = F_i(0)$ . Differentiating (A-2) with respect to  $\beta$ , one gets

$$\frac{dF_i(\beta)}{d\beta} = -\exp(-\beta\varepsilon_i) C_R \int \int_{L+1 \text{ fold}} \psi(e)X_i \exp\left(-\frac{1}{2} \mathbf{X}^T \mathbf{B}_i^{-1} \mathbf{X}\right) f_{\eta_g}(\eta_g) d\eta_g d\mathbf{X}, \tag{A-3}$$

where  $\mathbf{B}_i = (2\tilde{\beta}\mathbf{e}_i\mathbf{e}_i^T + \mathbf{R}_{X_C X_C}^{-1})^{-1}$ ,  $\tilde{\beta} = \alpha_\varepsilon\beta$  and  $\mathbf{e}_i$  is a column vector with its  $i$ -th element equal to one and zero elsewhere. Using the matrix inversion lemma, we get

$$\mathbf{B}_i = (2\tilde{\beta}\mathbf{e}_i\mathbf{e}_i^T + \mathbf{R}_{X_C X_C}^{-1})^{-1} = \mathbf{R}_{X_C X_C} \mathbf{G}_i, \tag{A-4}$$

where  $\mathbf{G}_i = (\mathbf{I} - 2\tilde{\beta}(\mathbf{g}_i(\tilde{\beta}))^{-1} \mathbf{E}_i)$ ,  $\mathbf{g}_i(\tilde{\beta}) = 1 + 2\tilde{\beta}R_{X_C X_C-i,i}$ ,  $R_{X_C X_C-i,j}$  is the  $(i, j)$ -th element of  $\mathbf{R}_{X_C X_C}$  and  $\mathbf{E}_i = \mathbf{e}_i \mathbf{r}_{X_C X_C-i}^T$ .  $\mathbf{r}_{X_C X_C-i}^T$  is the  $i$ -th row of  $\mathbf{R}_{X_C X_C}$ . The determinant of  $\mathbf{B}_i$  is  $|\mathbf{B}_i| = |\mathbf{R}_{X_C X_C}| |\mathbf{G}_i| = |\mathbf{R}_{X_C X_C}| (\mathbf{g}_i(\tilde{\beta}))^{-1}$ . (A-3) can be rewritten as follows

$$\begin{aligned} \frac{dF_i(\beta)}{d\beta} &= -\gamma_i(\beta) C_{B_i} \int \int_{L+1 \text{ fold}} \psi(e)X_i \cdot \exp\left(-\frac{1}{2} \mathbf{X}^T \mathbf{B}_i^{-1} \mathbf{X}\right) f_{\eta_g}(\eta_g) d\eta_g d\mathbf{X} \\ &= -\gamma_i(\beta) E_{\{X, \eta_g\}} [\psi(e)X_i | \mathbf{v}]_{E[\mathbf{X}\mathbf{X}^T] = \mathbf{B}_i} = -\gamma_i(\beta) L_{2,i}, \end{aligned} \tag{A-5}$$

where  $C_{B_i} = (2\pi)^{-L/2} |\mathbf{B}_i|^{-1/2}$ ,  $\gamma_i(\beta) = \exp(-\beta\varepsilon_i) (\mathbf{g}_i(\tilde{\beta}))^{-1/2}$ , and  $L_{2,i} = E_{\{X, \eta_g\}} [\psi(e)X_i | \mathbf{v}]_{E[\mathbf{X}\mathbf{X}^T] = \mathbf{B}_i}$  is the expectation of  $\psi(e)X_i$  conditioned on  $\mathbf{v}$  when  $X_i, X_j \in \mathbf{X}$  are jointly Gaussian with covariance matrix  $\mathbf{B}_i$ . Since  $\mathbf{X}$  and  $\mathbf{e}$  are assumed to be jointly Gaussian in Assumption 3, the Price's theorem [18] for  $\mathbf{X}$  and  $\mathbf{e}$  can be invoked to obtain the following,

$$L_{2,i} = \overline{\psi'}(\sigma_e^2) \mathbf{b}_i^T \mathbf{v}(n). \tag{A-6}$$

$\mathbf{b}_i$  is the  $i$ -th column of  $\mathbf{B}_i$ . Inserting (A-6) into (A-5) and integrating with respect to  $\beta$  yields

$$\begin{aligned} F_i(\beta) &= -\left(\int^\beta \gamma_i(\beta) \overline{\psi'}(\sigma_e^2) \mathbf{b}_i^T d\beta\right) \cdot \mathbf{v}(n), \\ &\approx \overline{\psi'}(\sigma_e^2) \mathbf{I}_i^T(\beta) \mathbf{v}(n), \end{aligned} \tag{A-7}$$

where  $\overline{\psi'}(\sigma_e^2) = \int_{-\infty}^\infty \frac{\psi'(e)}{\sqrt{2\pi\sigma_e}} \exp\left(-\frac{e^2}{2\sigma_e^2}\right) de$ ,  $\mathbf{I}_i^T(\beta) = -\int^\beta \gamma_i(\beta) \mathbf{b}_i^T d\beta$ , and the constant of integration is equal to

zero because of the boundary condition  $F_i(\infty)=0$ . Here, we have assumed that  $\sigma_e^2(\mathbf{v})$  depends weakly on  $\beta$  and can be taken outside of the integral. This is a good approximation if the variation of  $\overline{\psi}^T(\sigma_e^2)$  is limited, such as in the TDNLMM algorithm with adaptive threshold selection or at the steady state of the algorithm. To evaluate  $\mathbf{I}_i^T(\beta)$ , we note from (A-4) that

$$[\mathbf{B}_i]_{i,j} = R_{X_C X_C -j,j} - 2\tilde{\beta} \left( g_i(\tilde{\beta}) \right)^{-1} [R_{X_C X_C} \mathbf{E}_i]_{i,j} \\ = \left( g_i(\tilde{\beta}) \right)^{-1} R_{X_C X_C -j,j}. \tag{A-8}$$

Hence,

$$[\mathbf{I}_i^T(0)]_j = \int_0^\infty \gamma_i(\beta) [\mathbf{B}_i]_{i,j} d\beta = \alpha_i R_{X_C X_C -j,j}, \tag{A-9a}$$

and

$$\mathbf{I}_i^T(0) = \alpha_i \mathbf{r}_{X_C X_C -i}^T, \tag{A-9b}$$

where  $\alpha_i = \int_0^\infty \exp(-\beta \varepsilon_i) \left( g_i(\tilde{\beta}) \right)^{-3/2} d\beta$ . Combining, we have the desired result

$$\mathbf{H}_{1,i} = \mathbf{F}_i(0) \approx \overline{\psi}^T(\sigma_e^2) \alpha_i \mathbf{e}_i^T R_{X_C X_C} \mathbf{v}(n). \tag{A-10}$$

### Appendix B

In this appendix,  $\mathbf{s}_3 = E_{\{X_C, \eta_g\}} [\psi^2(e) \Lambda_C^{-1} \mathbf{X}_C \mathbf{X}_C^T \Lambda_C^{-1} | \mathbf{v}]$  is evaluated. Similar to deriving  $\mathbf{H}_i$  in Appendix A, the  $(i, j)$ -th element of  $\mathbf{s}_3$  is given by

$$\mathbf{s}_{3,i,j} = C_R \iint_{L+1 \text{ fold}} \frac{\psi^2(e) X_i X_j}{(\varepsilon_i + \alpha_\varepsilon X_i^2)(\varepsilon_j + \alpha_\varepsilon X_j^2)} \cdot \exp\left(-\frac{1}{2} \mathbf{X}^T \mathbf{R}_{X_C X_C}^{-1} \mathbf{X}\right) f_{\eta_g}(\eta_g) d\eta_g d\mathbf{X}, \tag{B-1}$$

Let us define

$$\overline{\mathbf{F}}_{i,j}(\beta_1, \beta_2) = C_R \iint_{L+1 \text{ fold}} \frac{\psi^2(e) X_i X_j \exp(-\beta_1(\varepsilon_i + \alpha_\varepsilon X_i^2) - \beta_2(\varepsilon_j + \alpha_\varepsilon X_j^2))}{(\varepsilon_i + \alpha_\varepsilon X_i^2)(\varepsilon_j + \alpha_\varepsilon X_j^2)} \cdot \exp\left(-\frac{1}{2} \mathbf{X}^T \mathbf{R}_{X_C X_C}^{-1} \mathbf{X}\right) f_{\eta_g}(\eta_g) d\eta_g d\mathbf{X}. \tag{B-2}$$

It can be seen that  $\mathbf{s}_{3,i,j} = \overline{\mathbf{F}}_{i,j}(0, 0)$ . To evaluate  $\overline{\mathbf{F}}_{i,j}(\beta_1, \beta_2)$ , let's differentiate (B-2) twice with respect to  $\beta_1$  and  $\beta_2$ :

$$\frac{\partial^2 \overline{\mathbf{F}}_{i,j}(\beta_1, \beta_2)}{\partial \beta_1 \partial \beta_2} = C_R \exp(-(\beta_1 \varepsilon_i + \beta_2 \varepsilon_j)) \cdot \iint_{L+1 \text{ fold}} \psi^2(e) X_i X_j \exp\left(-\frac{1}{2} \mathbf{X}^T \mathbf{B}_{ij}^{-1} \mathbf{X}\right) f_{\eta_g}(\eta_g) d\eta_g d\mathbf{X} \\ = \gamma_{ij}(\beta_1, \beta_2) \mathbf{L}_{3,i,j}, \tag{B-3}$$

where  $\mathbf{L}_{3,i,j} = E_{\{X, \eta_g\}} [\psi^2(e) X_i X_j | \mathbf{v}] |_{E[\mathbf{X} \mathbf{X}^T] = \mathbf{B}_{ij}}$ ,  $\mathbf{B}_{ij} = \left( 2\tilde{\beta}_1 \mathbf{e}_i \mathbf{e}_i^T + 2\tilde{\beta}_2 \mathbf{e}_j \mathbf{e}_j^T + \mathbf{R}_{X_C X_C}^{-1} \right)^{-1}$ ,  $\gamma_{ij}(\beta_1, \beta_2) = \exp(-(\beta_1 \varepsilon_i + \beta_2 \varepsilon_j)) |\mathbf{B}_{ij}|^{1/2} |\mathbf{R}_{X_C X_C}|^{-1/2}$ , and  $C_{B_{ij}} = (2\pi)^{-L/2} |\mathbf{B}_{ij}|^{-1/2}$ .

Using the matrix inversion formula, it can be shown that [30] the determinant of  $\mathbf{B}_{ij}$  and its  $(i, k)$ -th and  $(k, j)$ -th elements are respectively given by

$$|\mathbf{B}_{ij}| = \left( u_{ij}(\tilde{\beta}_1, \tilde{\beta}_2) \right)^{-1} |\mathbf{R}_{X_C X_C}|, \tag{B-4a}$$

$$[\mathbf{B}_{ij}]_{i,k} = \left( u_{ij}(\tilde{\beta}_1, \tilde{\beta}_2) \right)^{-1} \phi_{i,j,i,k}(\tilde{\beta}_2), \tag{B-4b}$$

$$[\mathbf{B}_{ij}]_{k,j} = \left( u_{ij}(\tilde{\beta}_1, \tilde{\beta}_2) \right)^{-1} \phi_{i,j,k,j}(\tilde{\beta}_1), \tag{B-4c}$$

where  $u_{ij}(\tilde{\beta}_1, \tilde{\beta}_2) = g_i(\tilde{\beta}_1) g_j(\tilde{\beta}_2) - 4\tilde{\beta}_1 \tilde{\beta}_2 R_{X_C X_C -j,i}^2$ ,  $\phi_{i,j,i,k}(\tilde{\beta}_2) = [R_{X_C X_C -j,k} + 2\tilde{\beta}_2 (R_{X_C X_C -j,j} R_{X_C X_C -i,k} - R_{X_C X_C -i,j} R_{X_C X_C -j,k})]$ ,  $\phi_{i,j,k,j}(\tilde{\beta}_1) = [R_{X_C X_C -k,j} + 2\tilde{\beta}_1 (R_{X_C X_C -k,j} R_{X_C X_C -i,i} - R_{X_C X_C -j,i} R_{X_C X_C -k,i})]$ , and

$$[\mathbf{B}_i]_{i,k} = R_{X_C X_C -i,k} [1 + 2(\tilde{\beta}_1 + \tilde{\beta}_2) R_{X_C X_C -j,i}]^{-1} \tag{B-4d}$$

Using (B-4),  $\gamma_{ij}(\beta_1, \beta_2)$  is determined as follows

$$\gamma_{ij}(\beta_1, \beta_2) = \left( u_{ij}(\tilde{\beta}_1, \tilde{\beta}_2) \right)^{-1/2} \exp(-(\beta_1 \varepsilon_i + \beta_2 \varepsilon_j)). \tag{B-5}$$

Using the Price's theorem,  $\mathbf{L}_{3,i,j}$  is evaluated to be [30]

$$\mathbf{L}_{3,i,j} \approx 2C_\psi(\sigma_e^2) \mathbf{b}_i^T \mathbf{v} \mathbf{v}^T \mathbf{b}_j + B_\psi(\sigma_e^2) b_{i,j}. \tag{B-6}$$

where  $b_{i,j}$  is the  $(i, j)$ -th element of  $\mathbf{B}_i$  and  $B_\psi(\sigma_e^2) = E[\psi^2(e)] = \frac{1}{\sqrt{2\pi\sigma_e}} \int_{-\infty}^\infty \psi^2(e) \exp(-\frac{e^2}{2\sigma_e^2}) de$ ,  $C_\psi(\sigma_e^2) = \frac{d}{d\sigma_e^2} E[\psi^2(e)]$ . From (B-3) and (B-6), we have

$$\frac{\partial^2 \overline{\mathbf{F}}_{i,j}(\beta_1, \beta_2)}{\partial \beta_1 \partial \beta_2} = 2C_\psi(\sigma_e^2) \gamma_{ij}(\beta_1, \beta_2) \mathbf{b}_i^T \mathbf{v} \mathbf{v}^T \mathbf{b}_j \\ + B_\psi(\sigma_e^2) \gamma_{ij}(\beta_1, \beta_2) b_{i,j}. \tag{B-7}$$

Integrating (B-7) with respect to  $\beta_1$  and  $\beta_2$  yields

$$\mathbf{s}_{3,i,j} \approx C_\psi(\sigma_e^2) I_{1,i,j} + B_\psi(\sigma_e^2) I_{2,i,j}, \tag{B-8}$$

where  $I_{1,ij} = \int_0^\infty \int_0^\infty 2\gamma_{ij}(\beta_1, \beta_2) \mathbf{b}_i^T \mathbf{v} \mathbf{v}^T \mathbf{b}_j d\beta_2 d\beta_1$  and  $I_{2,ij} = \int_0^\infty \int_0^\infty \gamma_{ij}(\beta_1, \beta_2) b_{ij} d\beta_2 d\beta_1$ .

To simplify the analysis, we shall assume that  $\sigma_e^2$  depends weakly on  $\beta$  and is taken outside the integral (mean value theorem). Like  $A_\psi(\sigma_e^2)$ , this is a good

approximation if the variations of  $B_\psi(\sigma_e^2)$  and  $C_\psi(\sigma_e^2)$  are limited. The integrals are evaluated below.

**Evaluation of  $I_{2,ij}$ .**

From (B-5) and (B-8), we have

$$I_{2,ij} = R_{X_C X_C - ij} \int_0^\infty \int_0^\infty \left( u_{ij}(\tilde{\beta}_1, \tilde{\beta}_2) \right)^{-\frac{3}{4}} \exp(-(\beta_1 \varepsilon_1 + \beta_2 \varepsilon_2)) d\beta_2 d\beta_1 = \sum_{m=0}^\infty \alpha_{ij}^{(m)} (4^m) \binom{-\frac{3}{2} + m - 1}{m} R_{X_C X_C - ij}^{(2m+1)}. \tag{B-9}$$

where  $\alpha_{ij}^{(k)} = \int_0^\infty \int_0^\infty (\tilde{\beta}_1 \tilde{\beta}_2)^k (g_i(\tilde{\beta}_1) g_j(\tilde{\beta}_2))^{-(2k+3)/2} \exp(-(\beta_1 \varepsilon_1 + \beta_2 \varepsilon_2)) d\beta_2 d\beta_1 = \alpha_i^{(k)} \alpha_j^{(k)}$ , and  $\alpha_i^{(k)} = \int_0^\infty \int_0^\infty (\tilde{\beta})^k (g_i(\tilde{\beta}))^{-(2k+3)/2} \exp(-\beta \varepsilon_i) d\beta$ .

**Evaluation of  $I_{1,ij}$ .**

Similarly from (B-5) and (B-8), we get

$$\begin{aligned} I_{1,ij} &= 2 \sum_{k=1}^L \sum_{l=1}^L v_l v_k \int_0^\infty \int_0^\infty \left( u_{ij}(\tilde{\beta}_1, \tilde{\beta}_2) \right)^{-5/2} \exp(-(\beta_1 \varepsilon_i + \beta_2 \varepsilon_j)) \cdot \phi_{ij,l,j}(\tilde{\beta}_1) \phi_{ij,i,k}(\tilde{\beta}_2) d\beta_1 d\beta_2 \\ &= 2 \sum_{k=1}^L \sum_{l=1}^L v_l v_k \int_0^\infty \int_0^\infty \left( g_i(\tilde{\beta}_1) g_j(\tilde{\beta}_2) \right)^{-5/2} \exp(-(\beta_1 \varepsilon_i + \beta_2 \varepsilon_j)) \\ &\quad \cdot \left[ \sum_{m=0}^\infty \binom{-\frac{5}{2} + m - 1}{m} \left( -4\tilde{\beta}_1 \tilde{\beta}_2 R_{X_C X_C - j,i}^2 \right)^m \left( g_i(\tilde{\beta}_1) g_j(\tilde{\beta}_2) \right)^{-m} \right] \\ &\quad \cdot \phi_{ij,l,j}(\tilde{\beta}_1) \phi_{ij,i,k}(\tilde{\beta}_2) d\beta_1 d\beta_2 \\ &= 2 \sum_{k=1}^L \sum_{l=1}^L v_l v_k \sum_{m=0}^\infty \binom{-\frac{5}{2} + m - 1}{m} \left( -4R_{X_C X_C - j,i} \right)^m \cdot \left( R_{X_C X_C - l,j} R_{X_C X_C - i,k} \alpha_i^{(m,(3+2m)/2)} \alpha_j^{(m,(3+2m)/2)} \right. \\ &\quad - 2R_{X_C X_C - i,j} R_{X_C X_C - j,k} R_{X_C X_C - l,j} \cdot \alpha_i^{(m,(3+2m)/2)} \alpha_j^{(m+1,(5+2m)/2)} \\ &\quad \left. - 2R_{X_C X_C - i,j} R_{X_C X_C - l,i} R_{X_C X_C - i,k} \alpha_i^{(m+1,(5+2m)/2)} \alpha_j^{(m,(3+2m)/2)} + 4R_{X_C X_C - i,j}^2 R_{X_C X_C - l,i} R_{X_C X_C - j,k} \alpha_i^{(m+1,(5+2m)/2)} \alpha_j^{(m+1,(5+2m)/2)} \right), \end{aligned} \tag{B-10}$$

where we have assumed that the matrix  $R_{X_C X_C}$  is diagonal-dominant so that we can employ the binomial expansion and

$$\alpha_i^{(m,n)} = \int_0^\infty \frac{\tilde{\beta}^m \exp(-\beta \varepsilon_i)}{\left( 1 + 2\tilde{\beta} R_{X_C X_C - i,i} \right)^n} d\beta.$$

In matrix form, we have

$$I_{1,ij} = s_{ij}^{(0)} (\mathbf{r}_{X_C X_C - i})^T \mathbf{v} \mathbf{v}^T \mathbf{r}_{X_C X_C - j} + s_{ij}^{(1)} (\mathbf{r}_{X_C X_C - j})^T \mathbf{v} \mathbf{v}^T \mathbf{r}_{X_C X_C - i} + s_{ij}^{(2)} (\mathbf{r}_{X_C X_C - i})^T \mathbf{v} \mathbf{v}^T \mathbf{r}_{X_C X_C - i} + s_{ij}^{(3)} (\mathbf{r}_{X_C X_C - i})^T \mathbf{v} \mathbf{v}^T \mathbf{r}_{X_C X_C - j},$$

where

$$\begin{aligned} s_{ij}^{(0)} &= 2 \sum_{m=0}^\infty \binom{-\frac{5}{2} + m - 1}{m} \left( -4R_{X_C X_C - j,i} \right)^m \alpha_i^{(m,(3+2m)/2)} \alpha_j^{(m,(3+2m)/2)}, \\ s_{ij}^{(1)} &= \frac{1}{2} \sum_{m=0}^\infty \binom{-\frac{5}{2} + m - 1}{m} \left( -4R_{X_C X_C - j,i} \right)^{m+1} \alpha_i^{(m,(3+2m)/2)} \alpha_j^{(m+1,(5+2m)/2)}, \\ s_{ij}^{(2)} &= \frac{1}{2} \sum_{m=0}^\infty \binom{-\frac{5}{2} + m - 1}{m} \left( -4R_{X_C X_C - j,i} \right)^{m+1} \alpha_i^{(m+1,(5+2m)/2)} \alpha_j^{(m,(3+2m)/2)}, \\ s_{ij}^{(3)} &= \frac{1}{2} \sum_{m=0}^\infty \binom{-\frac{5}{2} + m - 1}{m} \left( -4R_{X_C X_C - j,i} \right)^{m+2} \alpha_i^{(m+1,(5+2m)/2)} \alpha_j^{(m+1,(5+2m)/2)}. \end{aligned}$$

## References

1. Widrow, B., McCool, J., Larimore, M. G., & Johnson, C. R., Jr. (1976). Stationary and nonstationary learning characteristics of the LMS adaptive filter. *Proceedings of IEEE*, 64, 1151–1162.
2. Plackett, R. L. (1972). The discovery of the method of least-squares. *Biometrika*, 59(2), 239–251.
3. Nagumo, J. I., & Noda, A. (1967). A learning method for system identification. *IEEE Transactions on Automatic Control*, AC-12, 282–287.
4. Sayed, A. H. (2003). *Fundamentals of adaptive filtering*. NY: Wiley.
5. Haykin, S. (2001). *Adaptive filter theory*, 4th edn. Prentice Hall Press.
6. Narayan, S., Peterson, A. M., & Narasimha, M. J. (1983). Transform domain LMS algorithm. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-31, 609–615.
7. Lee, J. C., & Un, C. K. (1986). Performance of transform-domain LMS adaptive digital filters. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 34(3), 499–510.
8. Boroujeny, B. F., & Gazor, S. (1992). Selection of orthonormal transforms for improving the performance of the transform domain normalized LMS algorithm. *IEE Proceedings. Radar and Signal Processing*, 139(5), 327–335.
9. Boroujeny, B. F. (1998). *Adaptive filters: theory and applications*. John Wiley & Sons.
10. Pei, S. C., & Tseng, C. C. (1996). Transform domain adaptive linear phase filter. *IEEE Transactions on Signal Processing*, 44(12), 3142–3146.
11. Marshall, D. F., Jenkins, W. K., & Murphy, J. J. (1989). The use of orthogonal transforms for improving performance of adaptive filters. *IEEE Transactions on Circuits and Systems*, 36(4), 474–484.
12. Zou, Y., Chan, S. C., & Ng, T. S. (2000). Least mean M-estimate algorithms for robust adaptive filtering in impulsive noise. *IEEE Transactions on Circuits and Systems II*, 47, 1564–1569.
13. Huber, P. J. (1981). *Robust statistics*. New York: John Wiley.
14. Hampel, F. R., Ronchetti, E. M., Rousseeuw, P. J., & Stahel, W. A. (2005). *Robust statistics: the approach based on influence functions*. New York: Wiley.
15. Chan, S. C., & Zou, Y. (2004). A recursive least M-estimate algorithm for robust adaptive filtering in impulsive noise: fast algorithm and convergence performance analysis. *IEEE Transactions on Signal Processing*, 52(4), 975–991.
16. Chan, S. C., & Zhou, Y. (2007). On the convergence analysis of the normalized LMS and the normalized least mean M-estimate algorithms. In: *Proc. IEEE Int. Symp. Signal Processing and Information Technology* (pp. 1059–1065).
17. Price, R. (1958). A useful theorem for nonlinear devices having Gaussian inputs. *IEEE Transactions on Information Theory*, 4(2), 69–72.
18. Price, R. (1964). Comment on: 'A useful theorem for nonlinear devices having Gaussian inputs'. *IEEE Transactions on Information Theory*, IT-10, 171.
19. Haweel, T. I., & Clarkson, P. M. (1992). A class of order statistic LMS algorithms. *IEEE Transactions on Signal Processing*, 40(1), 44–53.
20. Settineri, R., Najim, M., & Ottaviani, D. (1996). Order statistic fast Kalman filter. *Proceedings of IEEE International Symposium on Circuits and Systems*, 2, 116–119.
21. Weng, J. F., & Leung, S. H. (1997). Adaptive nonlinear RLS algorithm for robust filtering in impulsive noise. *Proceedings of IEEE International Symposium on Circuits and Systems*, 4, 2337–2340.
22. Koike, S. (1997). Adaptive threshold nonlinear algorithm for adaptive filters with robustness against impulsive noise. *IEEE Transactions on Signal Processing*, 45(9), 2391–2395.
23. Bershad, N. J. (1988). On error-saturation nonlinearities in LMS adaptation. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-36(4), 440–452.
24. Mathews, V. J. (1991). Performance analysis of adaptive filters equipped with the dual sign algorithm. *IEEE Transactions on Signal Processing*, 39, 85–91.
25. Koike, S. (2006). Performance analysis of the normalized LMS algorithm for complex-domain adaptive filters in the presence of impulse noise at filter input. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Science*, E89-A(9), 2422–2428.
26. Koike, S. (2006). Convergence analysis of adaptive filters using normalized sign-sign algorithm. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Science*, E88-A(11), 3218–3224.
27. Tukey, J. W. (1960). *A survey of sampling from contaminated distributions in contributions to probability and statistics: I*. In: Olkin (Ed.) Stanford University Press.
28. Recktenwald, G. (2000). *Numerical methods with MATLAB: implementations and applications*. Englewood Cliffs: Prentice-Hall.
29. Bershad, N. J. (1986). Analysis of the normalized LMS algorithm with Gaussian inputs. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-34, 793–806.
30. Zhou, Y. (2006). Improved analysis and design of efficient adaptive transversal filtering algorithms with particular emphasis on noise, input and channel modeling. Ph. D. Dissertation, The Univ. Hong Kong, Hong Kong.
31. Chan, S. C., & Zhou, Y. (Dec. 2008). On the convergence analysis of the transform domain normalized LMS and related M-estimate algorithms. *Proc. IEEE APCCAS 2008*, 205–208.





**S. C. Chan (S'87–M'92)** received the B.Sc. (Eng.) and Ph.D. degrees from the University of Hong Kong in 1986 and 1992, respectively. He joined City Polytechnic of Hong Kong in 1990 as an Assistant Lecturer and later as a University Lecturer. Since 1994, he has been with the Department of Electrical and Electronic Engineering, the University of Hong Kong, and is now a Professor. He held visiting positions in Microsoft Corporation, Redmond, USA, Microsoft Research Asia, University of Texas at Arlington, and Nanyang Technological University. Dr. Chan's main research area is signal processing and applications. His previous research covers fast transform algorithms, filter design and realization, multirate signal processing, adaptive and communications signal processing, data compression and image-based rendering. Dr. Chan has served in a number of professional and administrative committees. He is currently a member of the Digital Signal Processing Technical Committee of the IEEE Circuits and Systems Society, Associate Editors for IEEE Trans. Circuits and Systems I and Journal of Signal Processing systems, and was the Chairman of the IEEE HK Chapter of Signal Processing. He

was the special session chairman of ICASSP2003 and is an organizing committee member of the IEEE 2010 ICIP.



**Y. Zhou (S'04–M'06)** received the B.Sc., M.Sc. degrees in 1997 and 1999 respectively, both in control engineering from Harbin Institute of Technology, Harbin, China, and Ph.D. degree in electrical engineering in 2006, from the University of Hong Kong, Hong Kong, China. From July 2006 to May 2009, he was with the Department of Electrical and Electronic Engineering, the University of Hong Kong as a postdoctoral research fellow. His research interests are adaptive signal processing, statistical signal processing, system identification and estimation theory. Dr. Zhou joined the Key Laboratory of Noise and Vibration Research, Institute of Acoustics, Chinese Academy of Sciences in August 2009, where he is now an associate professor and researching statistical signal processing theory and techniques applied to acoustics.