An Alternative Solution to the Dynamically Regularized RLS Algorithm

Feiran Yang*, Felix Albu†, and Jun Yang*
* Key Laboratory of Noise and Vibration Research, Institute of Acoustics,
Chinese Academy of Sciences, Beijing 100190, China
† Department of Electronics, Valahia University of Targoviste, Targoviste 130082, Romania
E-mail: {feiran@mail.ioa.ac.cn; felix.albu@gmail.com; jyang@mail.ioa.ac.cn}

***Abstract*—The recursive least-squares (RLS) algorithm should be explicitly regularized to achieve a satisfactory performance when the signal-to-noise ratio is low. However, a direct implementation of the involved matrix inversion results in a high complexity. In this paper, we present a recursive approach to the matrix inversion of the dynamically regularized RLS algorithm by exploiting the special structure of the correlation matrix. The proposed method has a similar complexity to the standard RLS algorithm. Moreover, the new method provides an exact solution for a fixed regularization parameter, and it has a good accuracy even for a slowly time-varying regularization parameter. Simulation results confirm the effectiveness of the new method.**

I. Introduction

Though the recursive least-squares (RLS) algorithm is computationally inefficient compared to the well-known least-mean-square (LMS) algorithm, it is still preferred in many applications due to the potential fast convergence [1]. There are two key parameters in the RLS algorithm, i.e., the forgetting factor and the regularization parameter. Recently, it has been well realized that the regularization parameter also plays an important role for the algorithm stability [2]–[6]. In the standard RLS algorithm, the regularization parameter exponentially decays as time progresses [1], which results in a poor stability when the signal-to-noise ratio (SNR) is low or the correlation matrix is poorly conditioned [2], [3].

To address this problem, an explicit regularization parameter is added to the diagonal elements of the correlation matrix prior to the inversion, which could reduce the eigenvalue spread and thus enhance the stability. Many criteria for choice of the regularization have been presented in [2]–[7]. However, the involved matrix inversion cannot be solved using the recursive method in the standard RLS algorithm. The direct implementation methods, e.g., Gaussian elimination method or Choleski decomposition, are too expensive, although they can provide an exact solution. The dichotomous co-ordinate descent (DCD) algorithm has been recommended to solve the norm equations [3], [8]–[9]. The DCD method does not require the multiplication operation, which is appreciated especially for the FPGA platform. The potential issue of the DCD method may be that the accuracy depends highly on the number of iterations and the input signal characteristics.

In this paper, we present an alternative solution to the matrix inversion in the dynamically regularized RLS algorithm, which is achieved by exploiting the structure property of the correlation matrix in the transversal structure. The partitioned matrix inversion lemma is employed to derive the recursive update equation for the matrix inversion [10]–[12]. Our approach provides an exact solution for the fixed regularization parameter, and the performance of the new method is very close to that of the exact matrix inversion for a time-varying regularization parameter. In addition, the proposed method has a similar complexity as the oracle RLS algorithm. The convergence performance of the new algorithm is verified by computer simulations in different experimental conditions.

This paper is organized as follows. We briefly review the standard RLS algorithm and the regularized version in Section II. Section III presents the proposed solution to the matrix inversion in the regularized RLS. We then carry out extensive computer simulations to verify the proposed method in Section IV. Finally, Section V concludes the paper.

II. Regularized RLS

In the context of system identification, we assume that the desired signal arises from the linear model

$$d(n) = \mathbf{w}^T\mathbf{x}(n) + v(n), \tag{1}$$

where $(\cdot)^T$ denotes the transpose operator, $\mathbf{w} = [w_0, w_1, ..., w_{L-1}]^T$ is the unknown system vector with length L, $\mathbf{x}(n) = [x(n), x(n-1), ..., x(n-L+1)]^T$ is the input signal vector, and $v(n)$ denotes the system noise.

The standard RLS algorithm is derived using the cost function [1]

$$J(n) = \sum_{i=0}^{n} \lambda^{n-i}\left[d(i) - \mathbf{x}^T(i)\hat{\mathbf{w}}(n)\right]^2 + \lambda^n \eta \|\hat{\mathbf{w}}(n)\|^2 \tag{2}$$

where $\|\cdot\|$ denotes the Euclidean norm, $\hat{\mathbf{w}} = [\hat{w}_0, \hat{w}_1, ..., \hat{w}_{L-1}]^T$ is the estimated weight vector, η is the initial regularization parameter, and λ is the forgetting

The work of F. Yang was supported by National Natural Science Foundation of China under Grants 61501449 and 11674348, Youth Innovation Promotion Association of Chinese Academy of Sciences under Grant 2018027, IACAS Young Elite Researcher Project QNYC201812, and the Strategic Priority Research Program of Chinese Academy of Sciences under Grant XDC02020400. The work of F. Albu was supported by a grant of the Romanian Ministry of Research and Innovation, CCCDI–UEFISCDI, project number PN-III-P3-3.1-PM-RO-CN-2018-0099/28BM/2018, within PNCDI III.

factor with $0 < \lambda \leq 1$. Minimizing $J(n)$ with respect to $\hat{\mathbf{w}}(n)$, we have

$$\hat{\mathbf{w}}(n) = (\mathbf{R}_x(n) + \lambda^n \eta \mathbf{I})^{-1}\mathbf{p}(n) \tag{3}$$

where $\mathbf{I}$ denotes the identity matrix, and

$$\begin{aligned}\mathbf{R}_x(n) &= \sum_{i=0}^{n} \lambda^{n-i}\mathbf{x}(i)\mathbf{x}^T(i) \\ &= \lambda \mathbf{R}_x(n-1) + \mathbf{x}(n)\mathbf{x}^T(n)\end{aligned} \tag{4}$$

is an estimate of the correlation matrix of the input signal, and

$$\mathbf{p}(n) = \sum_{i=0}^{n} \lambda^{n-i}\mathbf{x}(i)d(i). \tag{5}$$

is the estimate of the correlation between the input signal and the desired signal. The standard RLS algorithm can be updated as [1]

$$e(n) = d(n) - \mathbf{x}^T(n)\hat{\mathbf{w}}(n-1) \tag{6}$$

$$\hat{\mathbf{w}}(n) = \hat{\mathbf{w}}(n-1) + \mathbf{R}^{-1}(n)\mathbf{x}(n)e(n) \tag{7}$$

where

$$\mathbf{R}(n) = \mathbf{R}_x(n) + \lambda^n \eta \mathbf{I}, \tag{8}$$

$e(n)$ is the *a priori* estimation error, and $\mathbf{R}^{-1}(n)$ is calculated recursively using the matrix inversion lemma.

As shown, Eq. (8) corresponds to a time-varying regularization parameter $\lambda^n \eta$ at time index n, which fades quickly as time progresses. However, we expect that an independent and time-varying regularization parameter $\varepsilon(n)$ is used in many applications [2]–[7]. Thus, we need to invert the matrix $\mathbf{R}(n) = \mathbf{R}_x(n) + \varepsilon(n)\mathbf{I}$. In this situation, the matrix inversion lemma cannot be adopted to compute $\mathbf{R}^{-1}(n)$ recursively since $\varepsilon(n) \neq \lambda\varepsilon(n-1)$. A direct approach toward computation of the matrix inversion is via the Gaussian elimination method or Choleski decomposition, which usually requires $O(L^3)$ multiplications per iteration and hence is too expensive. The DCD approach can be used to solve the normal equations with a complexity of $O(L)$ [3], but the accuracy relies on the number of iterations and the input signal characteristics.

III. Proposed solution

We now present an alternative solution to the matrix inversion $\mathbf{R}^{-1}(n)$ by exploiting the time-shift property of the input vector. Defining the weighted input correlation

$$\rho_i(n) = \sum_{m=0}^{m=n} \lambda^{n-m} x(m)x(m-i), \tag{9}$$

the correlation matrix $\mathbf{R}(n)$ can be rewritten in the following two forms

$$\mathbf{R}(n) = \begin{bmatrix} r_0(n) & \boldsymbol{\alpha}^T(n) \\ \boldsymbol{\alpha}(n) & \tilde{\mathbf{R}}(n) \end{bmatrix} \tag{10}$$

$$\mathbf{R}(n) = \begin{bmatrix} \bar{\mathbf{R}}(n) & \boldsymbol{\beta}(n) \\ \boldsymbol{\beta}^T(n) & r_1(n) \end{bmatrix} \tag{11}$$

where

$$r_0(n) = \rho_0(n) + \varepsilon(n), \tag{12}$$

$$r_1(n) = \rho_0(n-L+1) + \varepsilon(n), \tag{13}$$

$$\boldsymbol{\alpha}(n) = [\rho_1(n), \rho_2(n), \cdots, \rho_{L-1}(n)]^T, \tag{14}$$

$$\boldsymbol{\beta}(n) = [\rho_{L-1}(n), \rho_{L-2}(n-1), \cdots, \rho_1(n-L+2)]^T \tag{15}$$

$\tilde{\mathbf{R}}(n)$ is the bottom-right $(L-1)\times(L-1)$ submatrix of $\mathbf{R}(n)$, and $\bar{\mathbf{R}}(n)$ is the top-left $(L-1)\times(L-1)$ submatrix of $\mathbf{R}(n)$.

The (i,j)-th entry of the matrix $\mathbf{R}_x(n)$ is $\rho_{|i-j|}(n - \min(i,j))$. For a fixed regularization parameter, we can then obtain an important relation

$$\tilde{\mathbf{R}}(n+1) = \bar{\mathbf{R}}(n). \tag{16}$$

In some applications, the regularization parameter $\varepsilon(n)$ varies slowly, and thus the approximation $\tilde{\mathbf{R}}(n+1) \approx \bar{\mathbf{R}}(n)$ holds. It should be mentioned that the matrix property in (16) only hold for the transversal structure.

Using (10) and the partitioned matrix inversion lemma, the inverse matrix $\mathbf{R}^{-1}(n)$ can be calculated as [13]

$$\mathbf{R}^{-1}(n) = \begin{bmatrix} 0 & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{R}}^{-1}(n) \end{bmatrix} + \frac{\mathbf{a}(n)\mathbf{a}^T(n)}{E_a(n)} \tag{17}$$

where

$$E_a(n) = r_0(n) - \boldsymbol{\alpha}^T(n)\tilde{\mathbf{R}}^{-1}(n)\boldsymbol{\alpha}(n), \tag{18}$$

$$\mathbf{a}(n) = \begin{bmatrix} 1 \\ -\tilde{\mathbf{R}}^{-1}(n)\boldsymbol{\alpha}(n) \end{bmatrix}, \tag{19}$$

and $\mathbf{0}$ is an all-zero vector. When $\tilde{\mathbf{R}}^{-1}(n)$ is available, we can compute the matrix inversion $\mathbf{R}^{-1}(n)$ according to (17). Using (11) and the partitioned matrix inversion lemma, the inverse matrix $\mathbf{R}^{-1}(n)$ can also be calculated as [13]

$$\mathbf{R}^{-1}(n) = \begin{bmatrix} \bar{\mathbf{R}}^{-1}(n) & \mathbf{0} \\ \mathbf{0} & 0 \end{bmatrix} + \frac{\mathbf{b}(n)\mathbf{b}^T(n)}{E_b(n)}, \tag{20}$$

where

$$E_b(n) = r_1(n) - \boldsymbol{\beta}^T(n)\bar{\mathbf{R}}^{-1}(n)\boldsymbol{\beta}(n), \tag{21}$$

$$\mathbf{b}(n) = \begin{bmatrix} -\bar{\mathbf{R}}^{-1}(n)\boldsymbol{\beta}(n) \\ 1 \end{bmatrix}. \tag{22}$$

Because $\mathbf{R}^{-1}(n)$ has been calculated as per (17), we could obtain $\bar{\mathbf{R}}^{-1}(n)$ using (20). But this can be achieved without the explicit computation of the vector $\mathbf{b}(n)$. Defining $\mathbf{c}(n)$ as the last column of $\mathbf{R}^{-1}(n)$ and $\theta(n)$ as the $(L-1, L-1)$-element of $\mathbf{R}^{-1}(n)$, we then obtain the relations

$$\mathbf{c}(n) = \frac{\mathbf{b}(n)}{E_b(n)} \tag{23}$$

and

$$\theta(n) = \frac{1}{E_b(n)}. \tag{24}$$

TABLE I
PROPOSED ALGORITHM

Step	Equation	Multiplications
	Initialization $\tilde{\mathbf{R}}^{-1}(0) = \varepsilon_0 \mathbf{I}$ with ε_0 a constant	
1	Compute $r_0(n)$ and $\boldsymbol{\alpha}(n)$ using (12) and (14)	$2L$
2	Compute $E_a(n)$ using (18)	$L^2 + L$
3	Update $\mathbf{a}(n)$ using (19)	0
4	Update $\mathbf{R}^{-1}(n)$ using (17)	$\frac{L^2+3L}{2}$
5	Update $\bar{\mathbf{R}}^{-1}(n))$ using (25)	$\frac{L^2+3L}{2}$
6	$\tilde{\mathbf{R}}^{-1}(n+1) = \bar{\mathbf{R}}^{-1}(n)$	0
7	Calculate the error signal $e(n)$ using (6)	L
8	Update the weight vector $\hat{\mathbf{w}}(n)$ using (7)	$L^2 + L$

We can then compute $\bar{\mathbf{R}}^{-1}(k)$ as

$$\begin{bmatrix} \bar{\mathbf{R}}^{-1}(n) & \mathbf{0} \\ \mathbf{0} & 0 \end{bmatrix} = \mathbf{R}^{-1}(n) - \frac{\mathbf{c}(n)\mathbf{c}^T(n)}{\theta(n)}. \quad (25)$$

Recalling the relation in (16), it has $\tilde{\mathbf{R}}^{-1}(n+1) = \bar{\mathbf{R}}^{-1}(n)$, which is then used for the calculation of (17). For a fixed regularization parameter, i.e., $\varepsilon(n)$ is constant, the proposed method could provide an exact solution to the matrix inversion. For a variable regularization parameter, the proposed approach can also track the exact solution very quickly and provides a satisfactory solution as shown in the simulations.

We summarize the proposed algorithm in Table 1. As seen, the proposed method requires $3L^2+8L$ multiplications per sample, which is only slightly higher than that of the standard RLS algorithm. However, the proposed method can adopt a variable regularization parameter, while the standard RLS cannot.

IV. SIMULATION RESULTS

Computer simulations are carried out to evaluate the performance of the proposed method. The standard RLS algorithm and the regularized RLS algorithm with an exact matrix inversion are involved for comparison. The system impulse response is taken from ITU-T G.168 Recommendation with length $L = 120$ [14]. Two types of signal are adopted as input. The first one is an AR(1) process that is obtained by filtering the white noise through the transfer function $H(z) = 1/(1 - 0.9z^{-1})$, and the second one is a speech signal. White noise is added to the desired signal to generate different SNRs. For all the algorithms, we use the forgetting factor $\lambda = 1 - 1/(2L) \approx 0.9958$. The normalized misalignment is adopted for the convergence evaluation, defined as $20\log_{10}(\|\mathbf{w} - \hat{\mathbf{w}}(n)\| \,/\, \|\mathbf{w}\|)$. The impulse response is multiplied by -1 in the middle of the iteration to model an abrupt system change.

In the first set of experiments, we adopt a fixed regularization parameter $\varepsilon(n) = 4\sigma_x^2$, where $\sigma_x^2 = E[x^2(n)]$ is the variance of the input signal. The standard RLS is initialized as $\mathbf{R}^{-1}(0) = \frac{1}{4\sigma_x^2}\mathbf{I}$. Fig. 1 and Fig. 2 present the misalignment curves of the three algorithms with SNR = 30 dB using AR(1) and speech as input, respectively. As seen, the standard RLS algorithm performs worst among the three algorithms. This is because that the equivalent regularization of the standard RLS decays as time and its effect becomes rather small. The learning curves of the proposed method and the regularized RLS algorithm with an exact matrix inversion become indistinguishable.

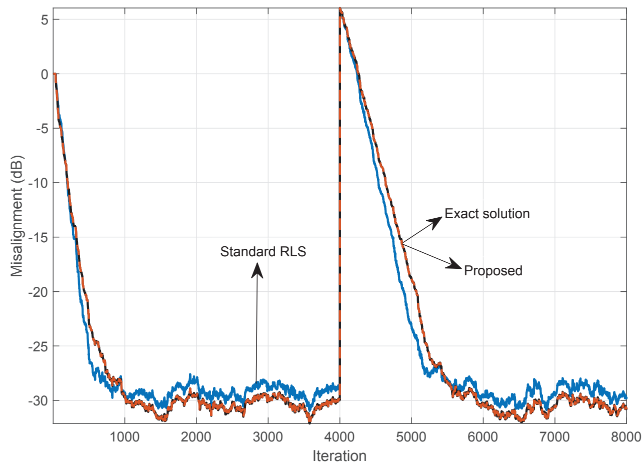

Fig. 1. Misalignment curves of the three algorithms with a fixed regularization. SNR = 30 dB, AR(1) as input.

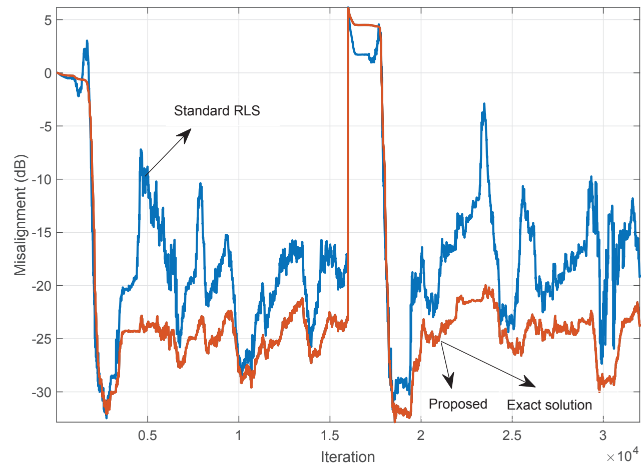

Fig. 2. Misalignment curves of the three algorithms with a fixed regularization. SNR = 30 dB, speech as input.

In the second set of experiments, we investigates the convergence performance of the proposed method with a time-varying regularization parameter. A low SNR = 10 dB is used. The time-varying regularization parameter for the regularized RLS algorithm is calculated using the method in [5], [7] as follows

$$\varepsilon(n) = \frac{L[1 + \sqrt{1 + \gamma(n)}]}{\gamma(n)} \sigma_x^2 \quad (26)$$

where $\gamma(n)$ is the SNR, which could be estimated by

$$\gamma(n) = \frac{\hat{\sigma}_{\hat{y}}^2(n)}{\left|\hat{\sigma}_d^2(n) - \hat{\sigma}_{\hat{y}}^2(n)\right|} \quad (27)$$

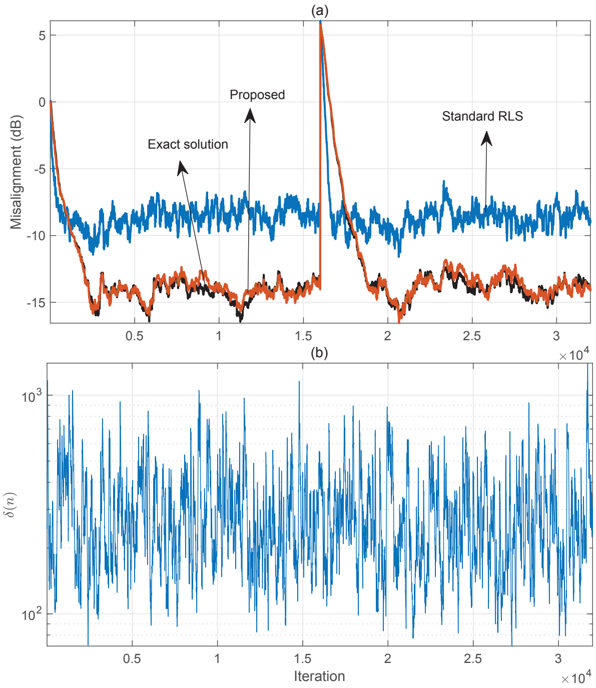

Fig. 3. Misalignment curves of the three algorithms with a time-varying regularization. SNR = 10 dB, AR(1) as input.

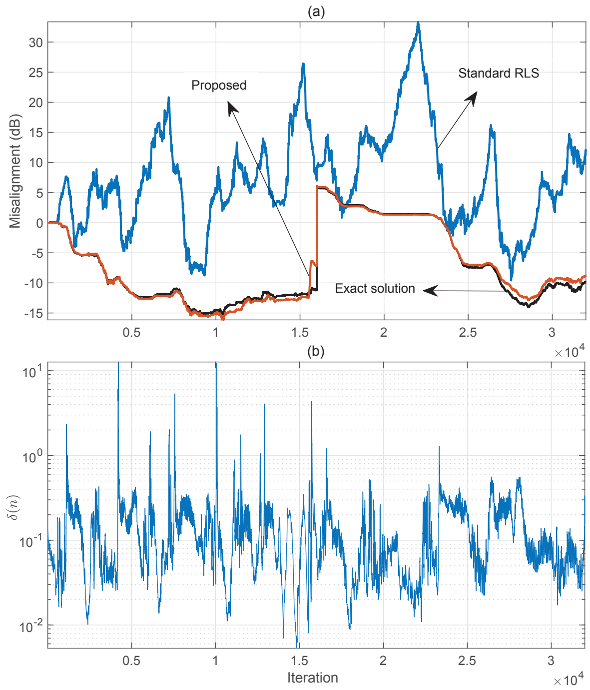

Fig. 4. Misalignment curves of the three algorithms with a time-varying regularization. SNR = 10 dB, speech as input.

The power estimates in (27) can be evaluated in a recursive manner as

$$\hat{\sigma}_d^2(n) = \kappa\hat{\sigma}_d^2(n-1) + (1-\kappa)d^2(n), \tag{28}$$

$$\hat{\sigma}_{\hat{y}}^2(n) = \kappa\hat{\sigma}_{\hat{y}}^2(n-1) + (1-\kappa)\hat{y}^2(n), \tag{29}$$

where κ is the smoothing factor, and $\hat{y}(n) = \mathbf{x}^T(n)\hat{\mathbf{w}}(n-1)$. We use $\kappa = 0.98$ in this paper. It should be mentioned that any other variable-regularization approach could be used.

We present the learning curve and the corresponding regularization parameter in Figs. 3(a) and 3(b) for AR(1) input, and that in Fig. 4 for speech input. The dynamic range of the regularization parameter for speech input is much larger than that for the AR(1) input. It is apparent that the convergence performance of the proposed method is very close to that of the exact regularized RLS algorithm for both AR(1) and speech inputs. This verifies that the proposed method can also perform well even for a variable regularization parameter.

V. Conclusion

This paper has presented a recursive method to solve the matrix inversion in the regularized RLS algorithm with a complexity of $O(3L^2)$. The complexity of the proposed method is between the exact solution with $O(L^3)$ and inexact DCD method with $O(L)$. However, the proposed approach provides an exact solution for a fixed regularization parameter and also gives a satisfactory result for a time-varying regularization parameter. Computer simulations confirmed the effectiveness of the proposed method.

References

[1] A.H. Sayed, *Fundamentals of Adaptive Filtering*, New York, NY, USA: Wiley-Inter-Science, 2003.
[2] S.L. Grant, "Dynamically regularized fast RLS with application to echo cancellation," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, May 1996, pp. 957–960
[3] J. Liu and Y. Zakharov, "Low complexity dynamically regularised RLS algorithm," *Electron. Lett.*, vol. 44, no. 14, pp. 886–885, July 2008.
[4] T.van Waterschoot, G. Rombouts, and M. Moonen, "Optimally regularized adaptive filtering algorithms for room acoustic signal enhancement," *Signal Process.*, vol. 88, no. 3, pp. 594–611, Mar. 2008.
[5] J. Benesty, C. Paleologu, and S. Ciochină: 'Regularization of the RLS algorithm," *IEICE Trans. Fundamentals*, vol. E94-A, pp. 1628–1629, Aug. 2011.
[6] M.C. Tsakiris, C.G. Lopes, and P.A. Naylor, "An alternative criterion for regularization in recursive least-squares problems," in *Proc. 7th International Symposium on Wireless Communication Systems*, Sept. 2010, pp. 360–363
[7] C. Paleologu, J. Benesty, C. Stanciu, C. Anghel, and M. Stenta, "Robust regularization of the recursive least-squares algorithm," in *Proc. 8th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, 2016, pp. 1–4
[8] Y.V. Zakharov and T.C. Tozer, "Multiplication-free iterative algorithm for LS problem," *Electron. Lett.*, vol. 40, no. 9, pp. 567–569, Apr. 2004.
[9] C. Elisei-Iliescu, C. Stanciu, C. Paleologu, J. Benesty, C. Anghel, and S. Ciochină, "Robust variable-regularized RLS algorithms," in *Proc. 2017 Hands-free Speech Communications and Microphone Arrays (HSCMA)*, Mar. 2017, pp. 171–175.
[10] G.O. Glentis, "An efficient implementation of the memory improved proportionate affine projection algorithm," *Signal Process.*, vol. 118, pp. 25–35, Jan. 2016.
[11] F. Yang and J. Yang, "A comparative survey of fast affine projection algorithms," *Digital Signal Process.*, vol. 83, pp. 297–322, Dec. 2018
[12] F. Yang and J. Yang, "Multiband-structured Kalman filter," *IET Signal Process.*, vol. 12, no. 6, pp. 722–728, Aug. 2018.
[13] C.D. Meyer, *Matrix Analysis and Applied Linear Algebra*, SIAM, Philadelphia, 2000.
[14] Digital Network Echo Cancellers, *ITU-T Rec. G.168*, 2002