NEAREST KRONECKER PRODUCT DECOMPOSITION BASED NORMALIZED LEAST MEAN SQUARE ALGORITHM

Sankha Subhra Bhattacharjee and Nithin V. George, Member, IEEE

Department of Electrical Engineering, Indian Institute of Technology Gandhinagar, India

ABSTRACT

Recently, nearest Kronecker product (NKP) decomposition based Wiener filter and Recursive Least Squares (RLS) have been proposed and was found to be a good candidate for system identification and echo cancellation and was shown to offer better tracking performance along with lower computational complexity, especially for identification of low-rank systems. In this paper, we derive the Least Mean Square (LMS) versions of adaptive algorithms which take advantage of NKP decomposition, namely NKP-LMS and NKP Normalized LMS (NKP-NLMS) algorithms. We compare the convergence and tracking performance along with computational complexity between standard NLMS, standard RLS, NKP based RLS (RLS-NKP), the standard Affine Projection Algorithm (APA) and NKP-NLMS algorithm, to evaluate the efficacy of NKP-NLMS algorithm in the context of system identification. Simulation results show that NKP-NLMS can be a good candidate for system identification, especially for sparse/low rank systems.

Index Terms— System identification, nearest Kronecker product, Adaptive filter, Least mean square, Low rank approximation.

1. INTRODUCTION

System identification problems have been at the heart of development of adaptive filtering algorithms. Some common applications of system identification include network and acoustic echo cancellation [1–4], active noise control [5, 6], feedback cancellation [7–9], etc. to name a few. The Least Mean Square (LMS) and Normalized LMS (NLMS) are two of the most widely used adaptive algorithms because of their simplicity and ease of implementation [6, 10]. However, it is known that coloured inputs can degrade the performance of LMS and NLMS algorithms [6, 10]. The Affine Projection Algorithm (APA) which is a generalized form of the NLMS algorithm, can improve convergence when the input signal is correlated, at the cost of increased computational complexity [10]. APA and its variants have been widely used in different applications [11–13] and many fast implementations and improved versions of APA have been reported [14–16]. Nonetheless, dealing with long length impulse responses is still a critical issue in problems like echo cancellation [17, 18].

Kronecker product decomposition have been used in a wide range of applications in [19–21] mostly involving bilinear/trilinear forms. However, these methods are applicable only on certain (quasi-periodic) forms of impulse responses which are very different from real world echo paths [3, 17, 18, 22]. To overcome this limitation, [3] recently proposed a NKP decomposition based iterative Wiener filter which uses low rank approximation of impulse responses and was explored in the context of system identification. In [17, 18, 22], the authors extended this idea to derive NKP decomposition based adaptive algorithms, namely NKP based Recursive Least Squares (RLS-NKP) and Kalman filter and their advantages in the context of echo cancellation were explored. This idea has also been used to design beamformers [23–25]. In NKP decomposition, the problem of modelling a long impulse response is broken down into two subproblems involving the identification of two smaller length impulse responses which are then combined using Kronecker product to get the estimated long impulse response, thus reducing computational complexity. In this paper, we extend this idea and propose a NKP decomposition based adaptive algorithm which minimizes the mean squared error, namely NKP-LMS algorithm. To further improve the convergence rate and make it independent of input signal power, we propose a normalized version, namely NKP Normalized LMS (NKP-NLMS) algorithm. In [17, 18], RLS-NKP algorithm was compared with other versions of RLS algorithm but was not compared to APA, which has been widely used in various system identification scenarios. In this paper, extensive simulation studies are carried out to compare the performance and computational complexity between NKP-NLMS, RLS, RLS-NKP, APA with different projection orders and NLMS algorithms.

The rest of the paper is organized as follows, Section 2 gives background on LMS algorithm and nearest Kronecker product decomposition. In Section 3, the proposed NKP-LMS and NKP-NLMS algorithms are derived and their computational complexity is compared with NLMS, traditional APA, RLS and the recently proposed RLS-NKP algorithms. In Section 4, simulation results are discussed and conclusions are provided in Section 5.

2. LINEAR SYSTEM IDENTIFICATION AND KRONECKER PRODUCT DECOMPOSITION

Let's consider the following signal model $y(n) = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}(n) + v(n)$, where $\boldsymbol{w} = [w_0, w_1, \ldots, w_{L-1}]^{\mathrm{T}}$ is a Finite Impulse Response (FIR) filter of length L, $y(n)$ is the zero mean reference signal with n being the discrete time index, $\boldsymbol{x}(n) = [x(n), x(n-1), \ldots, x(n-L+1)]^{\mathrm{T}}$ is a tap vector of the input signal $x(n)$, of length L and $v(n)$ is a zero mean additive white Gaussian noise. The output of the FIR filter used to model $\boldsymbol{w}$ is given by $\widehat{y}(n) = \widehat{\boldsymbol{w}}^{\mathrm{T}}(n)\boldsymbol{x}(n)$, where $\widehat{\boldsymbol{w}}(n) = [\widehat{w}_0(n), \widehat{w}_1(n), \ldots, \widehat{w}_{L-1}(n)]^{\mathrm{T}}$ are the coefficients of an adaptive FIR filter of length L. The error signal $e(n)$ can then be written as

$$e(n) = y(n) - \widehat{y}(n) = y(n) - \widehat{\boldsymbol{w}}^{\mathrm{T}}(n)\boldsymbol{x}(n) \quad (1)$$

From adaptive filter theory and considering instantaneous approximation, the optimal filter can be obtained by minimizing the cost function, here the Mean Square Error (MSE) [10]

$$J(\widehat{\boldsymbol{w}}) = E[e^2(n)/2] \approx e^2(n)/2 \quad (2)$$

This work is supported by the Department of Science and Technology, Government of India under the Core Grant Scheme (CRG/2018/002919) and TEOCO Chair of Indian Institute of Technology Gandhinagar. e-mail: sankha.bhattacharjee, nithin@iitgn.ac.in

Minimizing (2) w.r.t. $\widehat{\boldsymbol{w}}$, we get the instantaneous gradient $\mathbf{GR}(n) = -e(n)\boldsymbol{x}(n)$ and hence the well known Least Mean Square (LMS) algorithm [10, 26]

$$\widehat{\boldsymbol{w}}(n+1) = \widehat{\boldsymbol{w}}(n) - \mu\mathbf{GR}(n) = \widehat{\boldsymbol{w}}(n) + \mu e(n)\boldsymbol{x}(n) \quad (3)$$

Next, we briefly introduce the nearest Kronecker product formulation [27] for the FIR filter coefficients. Suppose $\boldsymbol{w}$ is a real valued impulse response, of length $L = L_1 \times L_2$ with $L_1 \geq L_2$, then we can decompose $\boldsymbol{w}$ as [3, 17]

$$\boldsymbol{w} = [\boldsymbol{g}_1^{\mathrm{T}}\ \boldsymbol{g}_2^{\mathrm{T}}\ \dots\ \boldsymbol{g}_{L_2}^{\mathrm{T}}]^{\mathrm{T}} \quad (4)$$

where $\boldsymbol{g}_i$'s for $i = 1, 2, \dots, L_2$, are assumed to be strongly linearly dependent impulse responses of length L_1 each. Then $\boldsymbol{w}$ can be approximated by $\mathbf{w}_2 \otimes \mathbf{w}_1 = vec(\mathbf{w}_1^{\mathrm{T}}\mathbf{w}_2)$, where $\mathbf{w}_1$ and $\mathbf{w}_2$ are impulse responses of length L_1 and L_2, $\otimes$ denotes Kronecker product and $vec(.)$ is the vectorization operation [28]. The misalignment in approximating $\boldsymbol{w}$ is given by $\mathcal{M}(\mathbf{w}_1, \mathbf{w}_2) = \|\boldsymbol{w} - \mathbf{w}_2 \otimes \mathbf{w}_1\|_2 / \|\boldsymbol{w}\|_2$ [3], where $\|.\|_2$ denotes ℓ_2 norm. $\boldsymbol{w}$ can also be reorganized and written as

$$\boldsymbol{W} = [\boldsymbol{g}_1\ \boldsymbol{g}_2\ \dots\ \boldsymbol{g}_{L_2}] \quad (5)$$

and hence the misalignment can be written as $\mathcal{M}(\mathbf{w}_1, \mathbf{w}_2) = \left\|\boldsymbol{W} - \mathbf{w}_1\mathbf{w}_2^{\mathrm{T}}\right\|_F / \|\boldsymbol{W}\|_F$, where $\|.\|_F$ denotes Frobenius norm. Hence, from [27, 29], minimizing $\mathcal{M}(\mathbf{w}_1, \mathbf{w}_2)$ is equal to finding the closest rank-1 matrix to $\boldsymbol{W}$, which can be found using singular value decomposition (SVD) as $\boldsymbol{W} = \mathbf{Q}_1\boldsymbol{\Sigma}\mathbf{Q}_2^{\mathrm{T}} = \sum_{i=1}^{L_2}\sigma_i\mathbf{q}_{1,i}\mathbf{q}_{2,i}^{\mathrm{T}}$, where $\boldsymbol{\Sigma}$ is an $L_1 \times L_2$ rectangular diagonal matrix containing singular values of $\boldsymbol{W}$ in decreasing order $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_{L_2} \geq 0$ and $\mathbf{Q}_1$ of size $L_1 \times L_1$ and $\mathbf{Q}_2$ of size $L_2 \times L_2$, contain the left and right singular vectors of $\boldsymbol{W}$ as their columns respectively. Then, the optimal $\mathbf{w}_1$ and $\mathbf{w}_2$ that minimizes $\mathcal{M}(\mathbf{w}_1, \mathbf{w}_2)$ can be obtained as $\overline{\mathbf{w}}_1 = \sqrt{\sigma_1}\mathbf{q}_{1,1}$ and $\overline{\mathbf{w}}_2 = \sqrt{\sigma_1}\mathbf{q}_{2,1}$, where $\mathbf{q}_{1,1}$ and $\mathbf{q}_{2,1}$ are the first column of $\mathbf{Q}_1$ and $\mathbf{Q}_2$ respectively, with the optimal approximation of $\boldsymbol{w}$ as $\overline{\boldsymbol{w}} = \overline{\mathbf{w}}_2 \otimes \overline{\mathbf{w}}_1$. Generally, $\boldsymbol{g}_i$'s for $i = 1, 2, \dots, L_2$ may not be linearly dependent. In that case, $\boldsymbol{w}$ can be approximated as

$$\boldsymbol{w} \approx \sum_{d=1}^{D}\mathbf{w}_{2,d}\otimes\mathbf{w}_{1,d} = vec\left(\sum_{d=1}^{D}\mathbf{w}_{1,d}\mathbf{w}_{2,d}^{\mathrm{T}}\right) = vec(\mathbf{W}_1\mathbf{W}_2^{\mathrm{T}}) \quad (6)$$

where $D \leq L_2$ and $\mathbf{w}_{1,d}$ and $\mathbf{w}_{2,d}$ for $d = 1, 2, \dots, D$, are impulse responses of length L_1 and L_2 respectively and

$$\mathbf{W}_1 = [\mathbf{w}_{1,1}\ \mathbf{w}_{1,2}\ \dots\ \mathbf{w}_{1,D}]_{(L_1\times D)} \quad (7)$$

$$\mathbf{W}_2 = [\mathbf{w}_{2,1}\ \mathbf{w}_{2,2}\ \dots\ \mathbf{w}_{2,D}]_{(L_2\times D)} \quad (8)$$

This leads to minimizing $\mathcal{M}(\mathbf{W}_1, \mathbf{W}_2) = \left\|\boldsymbol{W} - \mathbf{W}_1\mathbf{W}_2^{\mathrm{T}}\right\|_F / \|\boldsymbol{W}\|_F$, which leads to following optimal solution

$$\overline{\mathbf{W}}_1 = [\overline{\mathbf{w}}_{1,1}\ \overline{\mathbf{w}}_{1,2}\ \dots\ \overline{\mathbf{w}}_{1,D}]_{(L_1\times D)} \quad (9)$$

$$\overline{\mathbf{W}}_2 = [\overline{\mathbf{w}}_{2,1}\ \overline{\mathbf{w}}_{2,2}\ \dots\ \overline{\mathbf{w}}_{2,D}]_{(L_2\times D)} \quad (10)$$

where $\overline{\mathbf{w}}_{1,d} = \sqrt{\sigma_d}\mathbf{q}_{1,d}$ and $\overline{\mathbf{w}}_{2,d} = \sqrt{\sigma_d}\mathbf{q}_{2,d}$ for $d = 1, 2, \dots, D$ and hence the following approximation of $\boldsymbol{w}$

$$\overline{\boldsymbol{w}} = \sum_{d=1}^{D}\overline{\mathbf{w}}_{2,d}\otimes\overline{\mathbf{w}}_{1,d} = \sum_{d=1}^{D}\sigma_d\mathbf{q}_{2,d}\otimes\mathbf{q}_{1,d} \quad (11)$$

For $D = L_2$, we get back $\boldsymbol{w}$ exactly

$$\boldsymbol{w} = \sum_{d=1}^{L_2}\sigma_d\mathbf{q}_{2,d}\otimes\mathbf{q}_{1,d} \quad (12)$$

NOTATION: Vectors and matrices are represented using bold lowercase and bold uppercase letters respectively. All vectors used are considered as column vectors. $[.]_{a\times b}$ denotes size of matrix/vector $[.]$. $[.]^{\mathrm{T}}$ denotes transpose operation.

3. PROPOSED METHOD

Similar to the decomposition in (11), we can also decompose $\widehat{\boldsymbol{w}}$ as

$$\widehat{\boldsymbol{w}}(n) = \sum_{d=1}^{D}\widehat{\mathbf{w}}_{2,d}(n)\otimes\widehat{\mathbf{w}}_{1,d}(n) \quad (13)$$

where $\widehat{\mathbf{w}}_{1,d}$ and $\widehat{\mathbf{w}}_{2,d}$ are impulse responses of lengths L_1 and L_2 respectively. Using the following relationship [3, 27]

$$\begin{aligned}\widehat{\mathbf{w}}_{2,d}(n)\otimes\widehat{\mathbf{w}}_{1,d}(n) &= \Big(\widehat{\mathbf{w}}_{2,d}(n)\otimes\mathbf{I}_{L_1}\Big)\widehat{\mathbf{w}}_{1,d}(n)\\ &= \Big(\mathbf{I}_{L_2}\otimes\widehat{\mathbf{w}}_{1,d}(n)\Big)\widehat{\mathbf{w}}_{2,d}(n)\end{aligned} \quad (14)$$

where $\mathbf{I}_{L_1}$ and $\mathbf{I}_{L_2}$ are identity matrices of size $L_1 \times L_1$ and $L_2 \times L_2$, we can write

$$\widehat{\boldsymbol{w}}(n) = \sum_{d=1}^{D}\widehat{\mathbf{W}}_{2,d}(n)\widehat{\mathbf{w}}_{1,d}(n) = \sum_{d=1}^{D}\widehat{\mathbf{W}}_{1,d}(n)\widehat{\mathbf{w}}_{2,d}(n) \quad (15)$$

with$[\widehat{\mathbf{W}}_{2,d}(n)]_{L_1L_2\times L_1} = \widehat{\mathbf{w}}_{2,d}(n)\otimes\mathbf{I}_{L_1}$, and $[\widehat{\mathbf{W}}_{1,d}(n)]_{L_1L_2\times L_2} = \mathbf{I}_{L_2}\otimes\widehat{\mathbf{w}}_{1,d}(n)$. Using (13), (14) and (15) in (1), we can write the error signal in any of the following two forms

$$\begin{aligned}e(n) = e_1(n) &= y(n) - \sum_{d=1}^{D}\widehat{\mathbf{w}}_{1,d}^{\mathrm{T}}(n)\widehat{\mathbf{W}}_{2,d}^{\mathrm{T}}(n)\boldsymbol{x}(n)\\ &= y(n) - \sum_{d=1}^{D}\widehat{\mathbf{w}}_{1,d}^{\mathrm{T}}(n)\widehat{\mathbf{x}}_{2,d}(n) = y(n) - \widehat{\mathbf{w}}_1^{\mathrm{T}}(n)\mathbf{x}_2(n)\end{aligned} \quad (16)$$

$$\begin{aligned}e(n) = e_2(n) &= y(n) - \sum_{d=1}^{D}\widehat{\mathbf{w}}_{2,d}^{\mathrm{T}}(n)\widehat{\mathbf{W}}_{1,d}^{\mathrm{T}}(n)\boldsymbol{x}(n)\\ &= y(n) - \sum_{d=1}^{D}\widehat{\mathbf{w}}_{2,d}^{\mathrm{T}}(n)\widehat{\mathbf{x}}_{1,d}(n) = y(n) - \widehat{\mathbf{w}}_2^{\mathrm{T}}(n)\mathbf{x}_1(n)\end{aligned} \quad (17)$$

where

$$\widehat{\mathbf{x}}_{2,d}(n) = \widehat{\mathbf{W}}_{2,d}^{\mathrm{T}}(n)\boldsymbol{x}(n) \quad (18)$$

$$\widehat{\mathbf{w}}_1(n) = [\widehat{\mathbf{w}}_{1,1}^{\mathrm{T}}(n)\ \widehat{\mathbf{w}}_{1,2}^{\mathrm{T}}(n)\ \dots\ \widehat{\mathbf{w}}_{1,D}^{\mathrm{T}}(n)]^{\mathrm{T}} \quad (19)$$

$$\mathbf{x}_2(n) = [\widehat{\mathbf{x}}_{2,1}^{\mathrm{T}}(n)\ \widehat{\mathbf{x}}_{2,2}^{\mathrm{T}}(n)\ \dots\ \widehat{\mathbf{x}}_{2,D}^{\mathrm{T}}(n)]^{\mathrm{T}} \quad (20)$$

$$\widehat{\mathbf{x}}_{1,d}(n) = \widehat{\mathbf{W}}_{1,d}^{\mathrm{T}}(n)\boldsymbol{x}(n) \quad (21)$$

$$\widehat{\mathbf{w}}_2(n) = [\widehat{\mathbf{w}}_{2,1}^{\mathrm{T}}(n)\ \widehat{\mathbf{w}}_{2,2}^{\mathrm{T}}(n)\ \dots\ \widehat{\mathbf{w}}_{2,D}^{\mathrm{T}}(n)]^{\mathrm{T}} \quad (22)$$

$$\mathbf{x}_1(n) = [\widehat{\mathbf{x}}_{1,1}^{\mathrm{T}}(n)\ \widehat{\mathbf{x}}_{1,2}^{\mathrm{T}}(n)\ \dots\ \widehat{\mathbf{x}}_{1,D}^{\mathrm{T}}(n)]^{\mathrm{T}} \quad (23)$$

Following the above discussion, we can write that minimizing (2) is identical to minimizing the following cost functions [3, 17, 22]

$$J_{\widehat{\mathbf{w}}_2}[\widehat{\mathbf{w}}_1(n)] = E[e_1^2(n)/2] \approx e_1^2(n)/2 \quad (24)$$

$$J_{\widehat{\mathbf{w}}_1}[\widehat{\mathbf{w}}_2(n)] = E[e_2^2(n)/2] \approx e_2^2(n)/2 \quad (25)$$

One can observe that $J_{\widehat{\mathbf{w}}_2}[\widehat{\mathbf{w}}_1(n)]$ is dependent on values of $\widehat{\mathbf{w}}_2$ at previous time instants, i.e., $\widehat{\mathbf{w}}_2(n-1), \widehat{\mathbf{w}}_2(n-2), \dots, \widehat{\mathbf{w}}_2(0)$ through $\mathbf{x}_2(n)$ and similarly $J_{\widehat{\mathbf{w}}_1}[\widehat{\mathbf{w}}_2(n)]$ is dependent on $\widehat{\mathbf{w}}_1(n-1), \widehat{\mathbf{w}}_1(n-2), \dots, \widehat{\mathbf{w}}_1(0)$ via $\mathbf{x}_1(n)$. Following the assumptions considered in [3, 17], we consider values of $\widehat{\mathbf{w}}_2$ at previous time instants to be fixed while minimizing (24) and previous values of $\widehat{\mathbf{w}}_1$ to be fixed while minimizing (25) [3, 17]. Differentiating (24) w.r.t. $\widehat{\mathbf{w}}_1(n)$ and (25) w.r.t. $\widehat{\mathbf{w}}_2(n)$, we obtain the respective instantaneous gradients

$\mathbf{GR}_{\widehat{\mathbf{w}}_1}(n) = -e_1(n)\mathbf{x}_2(n)$ and $\mathbf{GR}_{\widehat{\mathbf{w}}_2}(n) = -e_2(n)\mathbf{x}_1(n)$ respectively. We can therefore derive the LMS update rules for $\widehat{\mathbf{w}}_1$ and $\widehat{\mathbf{w}}_2$ respectively [10]

$$\widehat{\mathbf{w}}_1(n+1) = \widehat{\mathbf{w}}_1(n) - \mu\mathbf{GR}_{\widehat{\mathbf{w}}_1}(n) = \widehat{\mathbf{w}}_1(n) + \mu e_1(n)\mathbf{x}_2(n) \quad (26)$$

$$\widehat{\mathbf{w}}_2(n+1) = \widehat{\mathbf{w}}_2(n) - \mu\mathbf{GR}_{\widehat{\mathbf{w}}_2}(n) = \widehat{\mathbf{w}}_2(n) + \mu e_2(n)\mathbf{x}_1(n) \quad (27)$$

where μ is the learning rate. Henceforth, we will call (26) and (27) as the Nearest Kronecker Product LMS (NKP-LMS) algorithm. One can observe that similar to the standard LMS algorithm, the amount of change in $\widehat{\mathbf{w}}_1(n)$ and $\widehat{\mathbf{w}}_2(n)$ will be proportional to the norm of $\mathbf{x}_2(n)$ and $\mathbf{x}_1(n)$ [10], respectively. Therefore, when norm of $\mathbf{x}_2(n)$ or $\mathbf{x}_1(n)$ is large, $\widehat{\mathbf{w}}_1(n)$ and $\widehat{\mathbf{w}}_2(n)$ respectively, will change substantially and vice versa. This can have detrimental effect on performance, in applications which involves large variations in the input signal, like speech [26]. With this motivation and following the steps in Section 4.4 of [10], we obtain the normalized versions of weight updates in (26) and (27) as

$$\widehat{\mathbf{w}}_1(n+1) = \widehat{\mathbf{w}}_1(n) + \frac{\mu}{\mathbf{x}_2^{\mathrm{T}}(n)\mathbf{x}_2(n) + \eta} e_1(n)\mathbf{x}_2(n) \quad (28)$$

$$\widehat{\mathbf{w}}_2(n+1) = \widehat{\mathbf{w}}_2(n) + \frac{\mu}{\mathbf{x}_1^{\mathrm{T}}(n)\mathbf{x}_1(n) + \eta} e_2(n)\mathbf{x}_1(n) \quad (29)$$

where η is a small positive constant, to avoid division by zero. Henceforth, (28) and (29) will be called Nearest Kronecker Product Normalized LMS (NKP-NLMS) algorithm.

We compare the computational complexity of RLS, RLS-NKP [17, 18], APA [10, 26], proposed NKP-NLMS, and NLMS [10] algorithms in terms of no. of multiplications and additions. From Table. 1, we can observe that the proposed NKP-NLMS algorithms have much lower computational complexity compared to RLS, RLS-NKP and APA algorithms, especially because it does not require any matrix inversion operation. We have used the same notation D to denote the rank of decomposition considered for NKP-NLMS and RLS-NKP algorithms as well as the projection order of APA, to make the computational complexities easily comparable. Even though the complexity of NKP-NLMS increases linearly with D and is higher than that of NLMS algorithm, the advantages of using the NKP-NLMS algorithm with respect to traditional APA is quite appealing and will be apparent from the simulation studies. It should be noted that L_1 should be as close as possible to L_2 [3] for NKP based algorithms to have least possible computational complexity.

Table 1. Total computational complexity

Algorithm	Multiplications	Additions
RLS [17]	$2L^2 + 2L$	$2L^2 + L + 1$
RLS-NKP [17, 18]	$(D+2)L+2(DL_1)^2$ $+2(DL_2)^2 + 2DL_1$ $+3DL_2$	$(D+1)L - (L_1 + L_2)$ $+2(DL_1)^2 + 2(DL_2)^2$ $+DL_2 + 3$
APA [10, 26]	$(D^2 + 2D)L$ $+D^3 + D$	$(D^2 + 2D)L$ $+D^3 + D^2$
NKP-NLMS	$2DL + 2DL_1$ $+3DL_2 + 2$	$2DL + DL_1$ $+2DL_2$
NLMS [10]	$3L + 1$	$3L$

4. SIMULATION RESULTS

We compare the proposed NKP-NLMS algorithm with RLS, RLS-NKP [17, 18], APA with projection order 2, 3 and 5 and NLMS

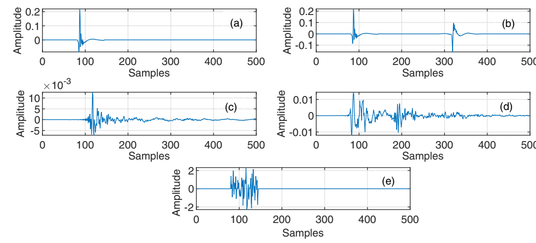

Fig. 1. Impulse responses with $L = 500$ and sparseness measure [3] (a) 0.8957, (b) 0.8080, (c) 0.6139, (d) 0.5142 and (e) 0.7479

algorithms for the problem of linear system identification. The input signal taken is an AR(1) process obtained by filtering a white Gaussian noise $\sim \mathcal{N}(0, 1)$ via the first order system $1/(1 - 0.9z^{-1})$. The signal to noise ratio (SNR) i.e., $E[y^2(n)]/E[v^2(n)]$ is considered to be 20 dB [3]. The performance measure considered is Normalized Misalignment as defined in [18] and the results reported are averaged over 50 independent runs. We have considered 5 impulse responses of length $L = 500$ each [3] shown in Fig. 1 as the optimal weights $\boldsymbol{w}$ of the system to be modelled. The impulse responses were taken from www.comm.pub.ro/imp-resp [3]. We consider $L_1 = 25$ and $L_2 = 20$ for NKP-NLMS and RLS-NKP in the simulations.

Fig. 1(a), the first impulse response from G168 Recommendation [30] and Fig. 1(b) resulting from concatenation of the first and fifth impulse response from G168 Recommendation [30], represent two typical network echo paths [22, 31] with high sparsity. Fig. 1(c) and 1(d) are two acoustic echo paths and Fig. 1(e) is a burst impulse response with 64 taps taken from a Gaussian distribution with rest all zeros. As shown in [3], for Fig. 1(a) and 1(b), the singular values of $\boldsymbol{W}$ in (5) quickly decays to zero, hence, they can be approximated very well using small values of $D << L_2$ in (11) and hence fit the notion of low rank systems. The singular values of Fig. 1(c) and 1(d) decays to zero more gradually, and therefore require higher value of D to be approximated well and hence may not be considered as low rank systems. For Fig. 1(e), even though it is less sparse compared to Fig. 1(a) and 1(b) [3], the fast decay of singular values to zero suggest that it can be approximated with small value of $D << L_2$ and hence, also qualifies as a low rank system. Due to lack of space, plots of singular values from SVD of $\boldsymbol{W}$ for the paths are not presented here. However, the reader is requested to refer to [3] where the singular values obtained with $L_1 = 25$ and $L_2 = 20$ for the paths considered here are discussed. The simulation parameters used are $\lambda = 1 - 1/(KL)$ for RLS and $L = 500$, $\lambda_1 = 1 - 1/(KDL_1)$ and $\lambda_2 = 1 - 1/(KDL_2)$ for RLS-NKP, with $K = 10$ [17, 18].

For APA in the simulation studies, D represents projection order. Considering the decompositions of the paths in Fig. 1 [3], we consider $L_1 = 25$, $L_2 = 20$ and the following values of D for NKP-NLMS and RLS-NKP algorithms, $D = 3$, $D = 5$, $D = 8$, $D = 8$ and $D = 3$ for the paths Fig. 1(a), Fig. 1(b), Fig. 1(c), Fig. 1(d) and Fig. 1(e) respectively, which can give exact (Fig. 1(a) & Fig. 1(e)) or good (Fig. 1(b)) or close enough (Fig. 1(c) & Fig. 1(d)) approximation of the respective paths. For path in Fig. 1(a): From Fig. 2(i), we can see that initial convergence speed of NKP-NLMS algorithm is faster than NLMS and APA with $D = 2$, similar to that of APA with $D = 3$, but slightly less than APA with $D = 5$ and quite less than RLS-NKP. Also, NKP-NLMS algorithm achieves lower steady state misalignment compared to NLMS, APA with $D = 2, 3$ and 5 and RLS-NKP algorithm. NKP-NLMS algorithm achieves this performance with lower computational complexity compared to

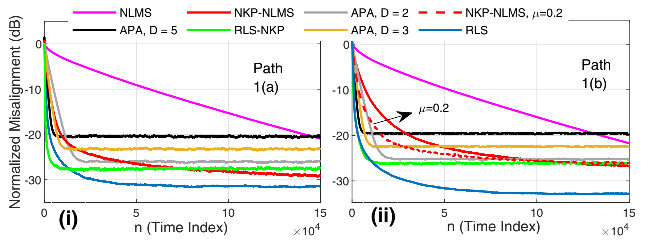

Fig. 2. Misalignment for path in Fig. 1(a) (left plot) and Fig. 1(b) (right plot), with $\mu = 0.1$ unless otherwise stated.

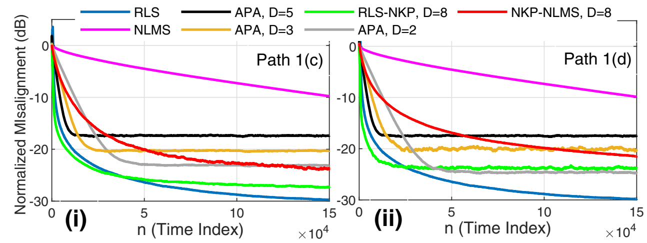

Fig. 3. Misalignment for path in Fig. 1(c) (left plot) and Fig. 1(d) (right plot), with $\mu = 0.04$.

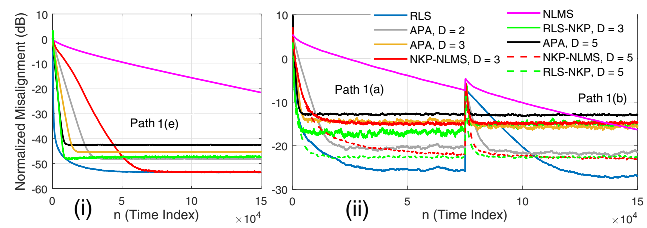

Fig. 4. Misalignment for path in Fig. 1(e) (left plot) with $\mu = 0.04$ and Misalignment for change in path from Fig. 1(a) to Fig. 1(b) (right plot) with $\mu = 0.2$.

APA with $D = 2, 3$ and 5 and much less than RLS-NKP algorithm (Table 1). For path in Fig. 1(b): From Fig. 2(ii), we can observe that NKP-NLMS algorithm has faster initial convergence compared to NLMS algorithm but slower than that of APA with $D = 2, 3$ and 5 and RLS-NKP algorithm. However, NKP-NLMS algorithm achieves lower steady state misalignment than APA with $D = 2, 3$ and 5 and slightly less than that of RLS-NKP algorithm. However, on increasing the learning rate for NKP-NLMS algorithm to $\mu = 0.2$, its convergence speed can be improved with a slight deterioration in steady state misalignment. NKP-NLMS algorithm achieves this performance with slightly higher computational complexity compared to APA with $D = 2$, but less than APA with $D = 3$ and 5 and much less than RLS-NKP algorithm (Table 1). We can also observe from Fig. 2(i)&(ii) that RLS-NKP algorithm proposed in [17, 18] is able to achieve faster convergence and lower steady state misalignment compared to APA but with a higher computational cost.

For paths in Fig. 1(c) and 1(d): From Table 1, NKP-NLMS with $D = 8$ has lower computational complexity than APA with $D = 5$ but slightly higher than APA with $D = 2, 3$. From Fig. 3(i), we can see that initial convergence speed of NKP-NLMS algorithm is faster than NLMS and APA with $D = 2$, similar to that of APA with $D = 3$, but slightly less than APA with $D = 5$ and quite less than RLS-NKP. Also, NKP-NLMS algorithm achieves lower steady state misalignment compared to NLMS, APA with $D = 2, 3$ and 5 but higher than RLS-NKP algorithm. From Fig. 3(ii), we can see that initial convergence speed of NKP-NLMS algorithm is faster than NLMS and APA with $D = 2$, close enough to APA with $D = 3$ but quite less than APA with $D = 5$ and RLS-NKP. However, NKP-NLMS algorithm achieves lower steady state misalignment compared to NLMS, APA with $D = 2, 3$ but higher than APA with $D = 5$ and RLS-NKP algorithm. For path in Fig. 1(e): Similar to the path in Fig. 1(a), we consider $D = 3$ for NKP-NLMS and RLS-NKP algorithms as its decomposition contains only 3 non-zero singular values [3]. From Fig. 4(i), we can see that convergence speed of NKP-NLMS algorithm is faster than NLMS but much less compared to APA with $D = 2, 3$ and 5 and RLS-NKP algorithm. However, it achieves lower steady state misalignment compared to NLMS, APA with $D = 2, 3, 5$ and RLS-NKP algorithm and similar steady state misalignment as that achieved by RLS. The computational complexity of NKP-NLMS in this case is same as that for the case in Fig. 2(i). For all the misalignment comparisons mentioned above, RLS performs best both in terms of convergence rate and steady state misalignment compared to all the other algorithms but has much higher computational complexity.

To compare the tracking performance of the algorithms, in Fig. 4(ii), we compare the algorithms for the case where the path changes halfway between the iterations from Fig. 1(a) to Fig. 1(b). We can observe that NKP-NLMS with $D = 3$ achieves slightly faster reconvergence compared to APA with $D = 2$ and similar reconvergence rate as APA with $D = 3, 5$. However, with $D = 3$ NKP-NLMS settles at higher steady state misalignment compared to APA with $D = 3$. However, with $D = 5$, NKP-NLMS steady state misalignment is better than APA with $D = 2, 3, 5$ similar to NKP-RLS before path change and fast reconvergence and better steady state misalignment compared to APA with $D = 2, 3, 5$. The reconvergence and steady state misalignment achieved by NKP-NLMS is very close to that achieved by RLS-NKP and both achieve faster reconvergence rate compared to RLS which is in accordance with the results reported in [17]. However, RLS achieves best steady state misalignment compared to all the other algorithms. APA with $D = 3, 5$ achieves similar reconvergence rate as RLS-NKP but the steady state misalignment is much worse. APA with $D = 2$ achieves steady state misalignment very close to that of RLS-NKP at the cost of slower reconvergence.

5. CONCLUSION

In this paper, we propose the NKP based LMS (NKP-LMS) and NKP based Normalized LMS (NKP-NLMS) algorithms and compare the performance and computational complexity of NKP-NLMS algorithm to that of RLS-NKP, traditional RLS, APA with different projection orders and NLMS algorithms. Experimental results suggest that the proposed NKP-NLMS algorithm is a viable option for identification of low rank systems and can offer better convergence characteristics for low rank systems compared to standard APA at a lower computational complexity. However as the rank of the system increases, compared to standard APA, the benefits decrease along with increase in computational complexity. However, it may still provide better tracking performance. Also, RLS-NKP can provide better convergence characteristics compared to standard APA for low rank systems. For higher rank systems, RLS-NKP can provide faster convergence rate compared to standard APA, however, the same may not be concluded for steady state misalignment and requires more detailed study.

6. REFERENCES

[1] Constantin Paleologu, Jacob Benesty, and Silviu Ciochina, "Sparse adaptive filters for echo cancellation," *Synthesis Lectures on Speech and Audio Processing*, vol. 6, no. 1, pp. 1–124, 2010.

[2] Jacob Benesty, Constantin Paleologu, Tomas Gänsler, and Silviu Ciochină, *A perspective on stereophonic acoustic echo cancellation*, vol. 4, Springer Science & Business Media, 2011.

[3] Constantin Paleologu, Jacob Benesty, and Silviu Ciochina, "Linear system identification based on a Kronecker product decomposition," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 26, no. 10, pp. 1793–1808, 2018.

[4] Jacob Benesty, Tomas Gänsler, Dennis R Morgan, M Mohan Sondhi, Steven L Gay, et al., *Advances in network and acoustic echo cancellation*, Springer, 2001.

[5] Nithin V George and Ganapati Panda, "Advances in active noise control: A survey, with emphasis on recent nonlinear techniques," *Signal processing*, vol. 93, no. 2, pp. 363–377, 2013.

[6] Kong-Aik Lee, Woon-Seng Gan, and Sen M Kuo, *Subband adaptive filtering: theory and implementation*, John Wiley & Sons, 2009.

[7] Somanath Pradhan, Vinal Patel, Dipen Somani, and Nithin V George, "An improved proportionate delayless multiband-structured subband adaptive feedback canceller for digital hearing aids," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 8, pp. 1633–1643, 2017.

[8] Linh Thi Thuc Tran, Sven Erik Nordholm, Henning Schepker, Hai Huyen Dam, and Simon Doclo, "Two-microphone hearing aids using prediction error method for adaptive feedback control," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 5, pp. 909–923, 2018.

[9] Somanath Pradhan, Nithin V George, Felix Albu, and Sven Nordholm, "Two microphone acoustic feedback cancellation in digital hearing aids: A step size controlled frequency domain approach," *Applied Acoustics*, vol. 132, pp. 142–151, 2018.

[10] Paulo SR Diniz, *Adaptive filtering*, Springer, 1997.

[11] Tadeu N Ferreira, Wallace A Martins, Markus V S Lima, and Paulo SR Diniz, "Convex combination of constraint vectors for set-membership affine projection algorithms," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 4858–4862.

[12] Mahmoud Hadef, Samir Bendoukha, Stephan Weiss, and Markus Rupp, "Affine projection algorithm for blind multiuser equalisation of downlink DS-CDMA system," in *Conference Record of the Thirty-Ninth Asilomar Conference on Signals, Systems and Computers, 2005.* IEEE, 2005, pp. 802–806.

[13] Markus Rupp, "Pseudo affine projection algorithms revisited: robustness and stability analysis," *IEEE Transactions on Signal Processing*, vol. 59, no. 5, pp. 2017–2023, 2011.

[14] Alberto Gonzalez, Miguel Ferrer, Felix Albu, and Maria De Diego, "Affine projection algorithms: evolution to smart and fast algorithms and applications," in *2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*. IEEE, 2012, pp. 1965–1969.

[15] Kazuhiko Ozeki, *Theory of affine projection algorithms for adaptive filtering*, Springer, 2016.

[16] Martin Bouchard, "Multichannel affine and fast affine projection algorithms for active noise control and acoustic equalization systems," *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 1, pp. 54–60, 2003.

[17] Camelia Elisei-Iliescu, Constantin Paleologu, Jacob Benesty, Cristian Stanciu, Cristian Anghel, and Silviu Ciochina, "Recursive least-squares algorithms for the identification of low-rank systems," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 27, no. 5, pp. 903–918, 2019.

[18] Camelia Elisei-Iliescu, Constantin Paleologu, Jacob Benesty, and Silviu Ciochina, "A recursive least-squares algorithm based on the nearest Kronecker product decomposition," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 4843–4847.

[19] Markus Rupp and Stefan Schwarz, "A tensor LMS algorithm," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 3347–3351.

[20] Alex P da Silva, Pierre Comon, and André LF de Almeida, "A finite algorithm to compute rank-1 tensor approximations," *IEEE Signal Processing Letters*, vol. 23, no. 7, pp. 959–963, 2016.

[21] Jacob Benesty, Constantin Paleologu, and Silviu Ciochină, "On the identification of bilinear forms with the Wiener filter," *IEEE Signal Processing Letters*, vol. 24, no. 5, pp. 653–657, 2017.

[22] Laura-Maria Dogariu, Constantin Paleologu, Jacob Benesty, and Silviu Ciochină, "An efficient Kalman filter for the identification of low-rank systems," *Signal Processing*, vol. 166, pp. 107239, 2020.

[23] Israel Cohen, Jacob Benesty, and Jingdong Chen, "Differential Kronecker product beamforming," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 27, no. 5, pp. 892–902, 2019.

[24] Wenxing Yang, Gongping Huang, Jacob Benesty, Israel Cohen, and Jingdong Chen, "On the design of flexible Kronecker product beamformers with linear microphone arrays," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 441–445.

[25] Xuehan Wang, Jacob Benesty, Gongping Huang, Jingdong Chen, and Israel Cohen, "Design of Kronecker product beamformers with cuboid microphone arrays," .

[26] Ali H Sayed, *Adaptive filters*, John Wiley & Sons, 2011.

[27] Charles F Van Loan, "The ubiquitous Kronecker product," *Journal of computational and applied mathematics*, vol. 123, no. 1-2, pp. 85–100, 2000.

[28] DA Harville, "Matrix algebra from a statistician's perspective. 1997," *Inc., Springer-Verlag New York*.

[29] Charles F Van Loan and Gene H Golub, *Matrix computations*, Johns Hopkins University Press, 1983.

[30] *Digital Network Echo Cancellers,* ITU-T Recommendations G.168, 2002, ," .

[31] Jacob Benesty and Steven L Gay, "An improved PNLMS algorithm," in *2002 IEEE International conference on Acoustics, Speech, and Signal Processing*. IEEE, 2002, vol. 2, pp. II–1881.