



# The Generalized Complex Kernel Affine Projection Algorithms

Xiaojian Wang<sup>1</sup> · Ming Zhang<sup>1</sup> · Jianxing Li<sup>1</sup> · Wenchao Chen<sup>1</sup> · Anxue Zhang<sup>1</sup>

Received: 18 May 2020 / Revised: 19 July 2021 / Accepted: 21 July 2021 /

Published online: 10 August 2021

© The Author(s) 2021

## Abstract

The complex kernel adaptive filter (CKAF) has been widely applied to the complex-valued nonlinear problem in signal processing and machine learning. However, most of the CKAF applications involve the complex kernel least mean square (CKLMS) algorithms, which work in a pure complex or complexified reproducing kernel Hilbert space (RKHS). In this paper, we propose the generalized complex kernel affine projection (GCKAP) algorithms in the widely linear complex-valued RKHS (WL-RKHS). The proposed algorithms have two main notable features. One is that they provide a complete solution for both circular and non-circular complex nonlinear problems and show many performance improvements over the CKAP algorithms. The other is that the GCKAP algorithms inherit the simplicity of the CKLMS algorithm while reducing its gradient noise and boosting its convergence. The second-order statistical characteristics of WL-RKHS have also been developed. An augmented Gram matrix consists of a standard Gram matrix and a pseudo-Gram matrix. This decomposition provides more underlying information when the real and imaginary parts of the signal are correlated and learning is independent. In addition, some online sparsification criteria are compared comprehensively in the GCKAP algorithms, including the nov-

---

✉ Ming Zhang  
itjerry@163.com

Xiaojian Wang  
wangxj235@163.com

Jianxing Li  
Jianxingli.china@xjtu.edu.cn

Wenchao Chen  
15156981@qq.com

Anxue Zhang  
anxuezhang@mail.xjtu.edu.cn

<sup>1</sup> School of Information and Communications Engineering, Xi'an Jiaotong University, Xi'an, Shaanxi 710049, China

elty criterion, the coherence criterion, and the angle criterion. Finally, two nonlinear channel equalization experiments with non-circular complex inputs are presented to illustrate the performance improvements of the proposed algorithms.

**Keywords** Kernel adaptive filter · Affine projection · Complex kernel methods · Widely linear approach

## 1 Introduction

Kernelized processing algorithms provide an attractive framework for dealing with many nonlinear problems in signal processing and machine learning. The fundamental mathematical notion of these technologies is to transform low-dimensional data into high-dimensional reproducing kernel Hilbert spaces (RKHSs) [28]. Nevertheless, the batch form of these algorithms usually requires a large amount of memory and computational complexity [27]. Kernel adaptive filters (KAFs) for online kernel processing have been studied extensively [9–12, 14, 15, 17–20, 26, 31, 33–35, 37, 38], including the kernel least mean square (KLMS) [17], kernel affine projection (KAP) [18, 26], kernel conjugate gradient (KCG) [38] and kernel recursive least squares (KRLS) [14].

As far as we know, most of the KAFs construct a real RKHS by real kernels, which are designed to process real data. However, complex-valued signals are also very common in practical nonlinear applications. For example, in communication system, when a QPSK signal passes through a nonlinear channel, the in-phase and quadrature components of the signal can be expressed compactly by a complex-value signal. In nonlinear electromagnetic calculation problems, the amplitude and phase of an electromagnetic wave can also be equivalent to a complex-value form. Therefore, several complex KAFs have recently been proposed [3–8, 22, 24]. The complex KLMS (CKLMS) was first proposed in [5, 6], which uses Wirtinger's calculus to generalize a complex RKHS. The authors proposed two CKLMS algorithms. One is CKLMS1, which uses real kernels to model the complex RKHS, called complexification of the real RKHSs. The other is CKLMS2, which uses the pure complex kernel. However, the convergence rate of CKLMS decreases from circular (proper) to highly non-circular (improper) complex inputs [25]. The widely linear approach is a generalized way to process the complex signal [30]. In [7], by employing this approach, the authors proposed an augmented CKLMS (ACKLMS) algorithm, in which the streams of input and desired signal pairs have the augmented vector form, which stacks the standard complex representation on top of its complex conjugate. Nevertheless, the ACKLMS algorithm exploits the same kernel for both real and imaginary parts. It is proved to be the same as CKLMS1. In [2], the widely linear reproducing kernel Hilbert space (WL-RKHS) theory was developed, in which a new pseudo-kernel was introduced to complement the standard kernel for a complete representation of the complex Hilbert space. In [3], the authors borrowed from the results on the WL-RKHS for nonlinear regression and proposed the generalized CKLMS (GCKLMS). That algorithm can provide a better representation of complex-valued signal than the pure complex and complexified methods and perform the independent learning of the real and the imag-

inary parts. Meanwhile, it is concluded that the previous version of the CKLMS can be expressed as particular cases of the GCKLMS.

Other complex KAFs have also been discussed. The linearly complex affine projection (CAP) algorithm, which uses the widely linear approach, was proposed in [13,36]. In [22], the authors introduced the CAP algorithm into the feature space and proposed the complex kernel affine projection (CKAP) algorithm. However, it is limited in a pure complex RKHS or a complexified RKHS (which uses real kernels to model the complex RKHS), and only circular complex signals are considered. When dealing with a non-circular complex signal, it is well known that widely linear minimum mean-squared error (MSE) estimation has many performance advantages over traditional linear MSE estimation. Meanwhile, the widely linear representation in the RKHS has been proved to be more powerful and convenient than the pure complex and the complexified representations [21,30].

Therefore, by employing the WL-RKHS theory in [2], we propose the generalized CKAP (GCKAP) algorithms. This is the first and the main contribution of this paper. The GCKAP algorithms have two main notable features. One is that they provide a complete solution for both circular and non-circular complex nonlinear problems, and show many performance improvements over the CKAP algorithms. The other is that the GCKAP algorithms inherit the simplicity of the CKLMS algorithm while reducing its gradient noise and boosting its convergence.

The affine projection algorithms use more recent augmented inputs in the current iteration to achieve the affine projection of weights onto the affine subspace [1,23]. During this process, the inverse of the Gram matrix (or kernel matrix) is needed for calculating the expansion coefficients. The forms of the augmented input and desired signal pairs determine the Gram matrix in GCKAP also has the augmented form. This leads to the second contribution of this paper: the development of second-order statistical characteristics of WL-RKHS. An augmented Gram matrix is introduced which consists of a standard Gram matrix and a pseudo-Gram matrix. This decomposition provides more underlying information when the real and imaginary parts of the signal are correlated and learning is independent. If the pseudo-Gram matrix vanishes, the input signal is proper; otherwise, it is improper. Improper can arise due to imbalance between the real and imaginary parts of a complex vector.

In addition, the inverse operation of the Gram matrix in the traditional KAP algorithm can no longer be solved by the previous iteration method [23] at the GCKAP iteration. To overcome this problem, we propose a decomposition method to reduce its complexity cost. This is the third contribution. Moreover, both the basic and the regularized GCKAP algorithms are proposed, named GCKAP-1 and GCKAP-2, respectively. Furthermore, it is found that the previous CKAP algorithms can be expressed as particular cases of the GCKAP algorithms. In addition, some online sparsification criteria are compared comprehensively in the GCKAP-2 algorithm, including the novelty criterion [19], the coherence criterion [26], and the angle criterion [38].

The rest of this paper is organized as follows. Section 2 discusses the theory of the WL-RKHS. Section 3 describes the details of the proposed GCKAP algorithms. Two simulation experiments are presented in Sect. 4. Finally, Sect. 5 summarizes the conclusions of this paper.

*Notations* We use bold lower-case letters to denote vectors and bold upper-case letters to denote matrices. The superscripts  $(\cdot)^T$ ,  $(\cdot)^H$  and  $(\cdot)^*$  denote the transpose, Hermitian transpose, and complex conjugate, respectively. The inner product is denoted by  $\langle \cdot, \cdot \rangle$ ; in the Hilbert space, it is denoted by  $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ . The expectation is denoted by  $\mathbb{E}(\cdot)$ .

## 2 Widely Linear RKHS

Let  $\mathcal{X}$  be a nonempty set of  $\mathbb{F}^M$ , where  $\mathbb{F}$  can be the field of real numbers  $\mathbb{R}$  or complex numbers  $\mathbb{C}$ , and  $\mathcal{H}$  be a Hilbert space of functions  $f : \mathcal{X} \rightarrow \mathbb{F}$ .  $\mathcal{H}$  is called the reducing kernel Hilbert space endowed with the inner product  $\langle \cdot, \cdot \rangle_{\mathcal{H}}$  and the norm  $\| \cdot \|_{\mathcal{H}}$  if there exists a kernel function  $\kappa : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{F}$  that satisfies the following properties.

1. For every  $\mathbf{x}$ ,  $\kappa(\mathbf{x}, \mathbf{x}')$  as a function of  $\mathbf{x}'$  belongs to  $\mathcal{H}$ .
2. The kernel function satisfies the reproducing property

$$f(\mathbf{x}) = \langle f(\cdot), \kappa(\mathbf{x}, \cdot) \rangle_{\mathcal{H}} \quad \mathbf{x} \in \mathcal{X}. \tag{1}$$

In particular,  $\langle \kappa(\mathbf{x}, \cdot), \kappa(\mathbf{x}', \cdot) \rangle_{\mathcal{H}} = \kappa(\mathbf{x}, \mathbf{x}')$ .

In the RKHS, a complex function is in a closed linear combination of the kernels at the training points

$$f(\mathbf{x}') = \sum_{i=1}^K \alpha_i \kappa(\mathbf{x}', \mathbf{x}(i)) = \kappa(\mathbf{x}', \mathbf{X}) \boldsymbol{\alpha}, \tag{2}$$

where  $\alpha_i \in \mathbb{F}$  is the linear combination coefficient of a kernel,  $\boldsymbol{\alpha} = [\alpha_1, \dots, \alpha_K]^T$ ,  $K$  is the number of training points, and  $\kappa(\mathbf{x}', \mathbf{X}) = [\kappa(\mathbf{x}', \mathbf{x}_1), \dots, \kappa(\mathbf{x}', \mathbf{x}_K)]$  is a row vector. The kernel trick states that we can construct a  $q$ -dimensional (possibly infinite) mapping  $\boldsymbol{\varphi}(\cdot)$  into the RKHS  $\mathcal{H}$  such that  $\kappa(\mathbf{x}', \mathbf{x}) = \langle \boldsymbol{\varphi}(\mathbf{x}'), \boldsymbol{\varphi}(\mathbf{x}) \rangle_{\mathcal{H}}$ . If  $\boldsymbol{\varphi}(\cdot)$  is the complex mapping  $\boldsymbol{\varphi}(\cdot) = \boldsymbol{\varphi}_r(\cdot) + j\boldsymbol{\varphi}_j(\cdot)$ , then the complex function  $f(\mathbf{x}')$  can be denoted by

$$f(\mathbf{x}') = \left[ \boldsymbol{\varphi}_r^T \boldsymbol{\Phi}_r + \boldsymbol{\varphi}_j^T \boldsymbol{\Phi}_j - j(\boldsymbol{\varphi}_j^T \boldsymbol{\Phi}_r - \boldsymbol{\varphi}_r^T \boldsymbol{\Phi}_j) \right] \boldsymbol{\alpha}, \tag{3}$$

where the coefficient vector is  $\boldsymbol{\alpha} = \boldsymbol{\alpha}_r + j\boldsymbol{\alpha}_j$  and the real and imaginary mappings are  $\boldsymbol{\varphi}_r = \boldsymbol{\varphi}_r(\mathbf{x}')$  and  $\boldsymbol{\varphi}_j = \boldsymbol{\varphi}_j(\mathbf{x}')$ , respectively.  $\boldsymbol{\Phi}_r = [\boldsymbol{\varphi}_r(\mathbf{x}_1), \dots, \boldsymbol{\varphi}_r(\mathbf{x}_K)]$ , and  $\boldsymbol{\Phi}_j = [\boldsymbol{\varphi}_j(\mathbf{x}_1), \dots, \boldsymbol{\varphi}_j(\mathbf{x}_K)]$ . The complex function  $f(\mathbf{x}') = f_r(\mathbf{x}') + j f_j(\mathbf{x}')$  can be represented by a real vector-valued function  $\mathbf{f}_{\mathbb{R}}(\mathbf{x}') = [f_r(\mathbf{x}'), f_j(\mathbf{x}')]^T \in \mathbb{R}^2$ , which is composed of its real and imaginary parts. The real vector-valued function in the RKHS can be represented by the product of the real transformation matrix and the real

coefficient vector:

$$\begin{aligned} \mathbf{f}_{\mathbb{R}}(\mathbf{x}') &= \begin{bmatrix} \boldsymbol{\varphi}_{r'}^T \boldsymbol{\Phi}_r + \boldsymbol{\varphi}_{j'}^T \boldsymbol{\Phi}_j & -\boldsymbol{\varphi}_{r'}^T \boldsymbol{\Phi}_j + \boldsymbol{\varphi}_{j'}^T \boldsymbol{\Phi}_r \\ \boldsymbol{\varphi}_{r'}^T \boldsymbol{\Phi}_j - \boldsymbol{\varphi}_{j'}^T \boldsymbol{\Phi}_r & \boldsymbol{\varphi}_{r'}^T \boldsymbol{\Phi}_r + \boldsymbol{\varphi}_{j'}^T \boldsymbol{\Phi}_j \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha}_r \\ \boldsymbol{\alpha}_j \end{bmatrix} \\ &\triangleq \begin{bmatrix} \boldsymbol{\kappa}_{rr} & \boldsymbol{\kappa}_{rj} \\ \boldsymbol{\kappa}_{jr} & \boldsymbol{\kappa}_{jj} \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha}_r \\ \boldsymbol{\alpha}_j \end{bmatrix} \triangleq \mathbf{K} \boldsymbol{\alpha}_{\mathbb{R}}, \end{aligned} \tag{4}$$

where  $\mathbf{K}$  denotes the real transformation matrix and  $\boldsymbol{\alpha}_{\mathbb{R}}$  denotes the real vector-valued expansion coefficient. This is a real vector linear system. It is observed that the complex RKHS of the real vector-valued functions is limited as  $\boldsymbol{\kappa}_{rr} = \boldsymbol{\kappa}_{jj}$ ,  $\boldsymbol{\kappa}_{rj} = -\boldsymbol{\kappa}_{jr}$ . However, in general, this limitation does not always hold. By employing the widely linear approach, the WL-RKHS theory [2] breaks this limitation. In the WL-RKHS, a complex vector-valued function  $f(\mathbf{x}')$  can be represented by an augmented form obtained by stacking the complex function  $f(\mathbf{x}')$  on top of its complex conjugate  $f(\mathbf{x}')^*$  with the expression  $\mathbf{f}(\mathbf{x}') = [f(\mathbf{x}'), f(\mathbf{x}')^*]^T \in \mathbb{C}^2$ . Therefore, a widely linear system  $\mathbf{f}(\mathbf{x}') = \mathbf{K}_A \underline{\boldsymbol{\alpha}}$  is modeled, where  $\underline{\boldsymbol{\alpha}} = [\boldsymbol{\alpha}^T, \boldsymbol{\alpha}^H]^T$  is the augmented expansion coefficient and  $\mathbf{K}_A \in \mathbb{C}^{2 \times 2K}$  is the augmented transformation matrix defined as

$$\mathbf{K}_A = \begin{bmatrix} \boldsymbol{\kappa} & \tilde{\boldsymbol{\kappa}} \\ \tilde{\boldsymbol{\kappa}}^* & \boldsymbol{\kappa}^* \end{bmatrix}, \tag{5}$$

where

$$\boldsymbol{\kappa} = \frac{1}{2} [\boldsymbol{\kappa}_{rr} + \boldsymbol{\kappa}_{jj} + j(\boldsymbol{\kappa}_{jr} - \boldsymbol{\kappa}_{rj})], \tag{6}$$

$$\tilde{\boldsymbol{\kappa}} = \frac{1}{2} [\boldsymbol{\kappa}_{rr} - \boldsymbol{\kappa}_{jj} + j(\boldsymbol{\kappa}_{jr} + \boldsymbol{\kappa}_{rj})]. \tag{7}$$

Here,  $\boldsymbol{\kappa}_{rr}$ ,  $\boldsymbol{\kappa}_{jj}$ ,  $\boldsymbol{\kappa}_{jr}$ , and  $\boldsymbol{\kappa}_{rj}$  are different row vectors that are defined in the same way as  $\boldsymbol{\kappa}(\mathbf{x}', \mathbf{X})$ . The complex function in (2) is represented by the first row of the widely linear system as

$$f(\mathbf{x}') = \boldsymbol{\kappa}(\mathbf{x}', \mathbf{X}) \boldsymbol{\alpha} + \tilde{\boldsymbol{\kappa}}(\mathbf{x}', \mathbf{X}) \boldsymbol{\alpha}^*, \tag{8}$$

where  $\boldsymbol{\kappa}(\mathbf{x}', \mathbf{X})$  is the standard kernel vector and  $\tilde{\boldsymbol{\kappa}}(\mathbf{x}', \mathbf{X}) = [\tilde{\boldsymbol{\kappa}}(\mathbf{x}', \mathbf{x}_1), \dots, \tilde{\boldsymbol{\kappa}}(\mathbf{x}', \mathbf{x}_K)]$  is defined as the pseudo-kernel vector. The augmented representation in the WL-RKHS obviously has some built-in redundancy, but it is very powerful and convenient, as we have to deal with non-circular complex signals.

Next, the second-order statistical properties of the widely linear system are discussed. Suppose the complex input  $\boldsymbol{\Phi} = \boldsymbol{\Phi}_r + j\boldsymbol{\Phi}_j$  in the RKHS has a zero mean. The real correlation matrix of  $\boldsymbol{\Phi}_{\mathbb{R}} = [\boldsymbol{\Phi}_r^T, \boldsymbol{\Phi}_j^T]^T$  is

$$\mathbf{R}_{\mathbb{R}} = \begin{bmatrix} \mathbf{R}_{rr} & \mathbf{R}_{rj} \\ \mathbf{R}_{rj}^T & \mathbf{R}_{jj} \end{bmatrix}, \tag{9}$$

where  $\mathbf{R}_{rr} = \mathbb{E}[\boldsymbol{\Phi}_r \boldsymbol{\Phi}_r^T]$ ,  $\mathbf{R}_{rj} = \mathbb{E}[\boldsymbol{\Phi}_r \boldsymbol{\Phi}_j^T] = \mathbf{R}_{jr}^T$ , and  $\mathbf{R}_{jj} = \mathbb{E}[\boldsymbol{\Phi}_j \boldsymbol{\Phi}_j^T]$ . Define the real-to-complex transformation matrix  $\mathbf{T}$  as

$$\mathbf{T} = \begin{bmatrix} \mathbf{I} & j\mathbf{I} \\ \mathbf{I} & -j\mathbf{I} \end{bmatrix} \in \mathbb{C}^{2q \times 2q}, \tag{10}$$

which is unitary up to a factor of 2, that is,  $\mathbf{T}\mathbf{T}^H = \mathbf{T}^H\mathbf{T} = 2\mathbf{I}$ . The augmented input  $\underline{\boldsymbol{\Phi}} = [\boldsymbol{\Phi}^T, \boldsymbol{\Phi}^H]^T$  can be expressed by the transformation of  $\boldsymbol{\Phi} = \mathbf{T}\boldsymbol{\Phi}_{\mathbb{R}}$ . Then, the augmented correlation matrix can be derived as

$$\underline{\mathbf{R}} = \mathbf{T}\mathbf{R}_{\mathbb{R}}\mathbf{T}^H = \begin{bmatrix} \mathbf{R} & \tilde{\mathbf{R}} \\ \tilde{\mathbf{R}}^* & \mathbf{R}^* \end{bmatrix}, \tag{11}$$

where  $\mathbf{R}$  is the standard correlation matrix

$$\mathbf{R} = \mathbf{R}_{rr} + \mathbf{R}_{jj} + j(\mathbf{R}_{jr} - \mathbf{R}_{rj}) = \mathbf{R}^H \tag{12}$$

and  $\tilde{\mathbf{R}}$  is the pseudo-correlation matrix

$$\tilde{\mathbf{R}} = \mathbf{R}_{rr} - \mathbf{R}_{jj} + j(\mathbf{R}_{jr} + \mathbf{R}_{rj}) = \tilde{\mathbf{R}}^T. \tag{13}$$

It is worth noting that both  $\mathbf{R}$  and  $\tilde{\mathbf{R}}$  are required for a complete second-order characterization of the WL-RKHS. The degree of impropriety can be measured by the complex correlation coefficient  $\rho$  between  $\boldsymbol{\Phi}$  and  $\boldsymbol{\Phi}^*$ . Many correlation analysis techniques transform  $\boldsymbol{\Phi}$  and  $\boldsymbol{\Phi}^*$  into internal representations  $\boldsymbol{\xi} = \mathbf{A}\boldsymbol{\Phi}$  and  $\boldsymbol{\psi} = \mathbf{B}\boldsymbol{\Phi}^*$ . The full-rank matrices  $\mathbf{A}$  and  $\mathbf{B}$  are chosen to maximize all partial sums over the absolute values of the correlations  $\eta_i = \mathbb{E}[\xi_i \psi_i]$ ,

$$\max_{\mathbf{A}, \mathbf{B}} = \sum_{i=1}^r |\eta_i|, \quad r = 1 \cdots p_r, \tag{14}$$

subject to the following constraints that determine three popular correlation analysis techniques: canonical correlation analysis (CCA), multivariate linear regression (MLR), and partial least squares (PLS) [29]. Some expressions of correlation coefficient  $\underline{\rho}$  are

$$\begin{aligned} \underline{\rho}_1 &= 1 - \prod_{i=1}^r (1 - \eta_i^2) = 1 - \frac{\det \underline{\mathbf{R}}}{\det^2 \mathbf{R}}, \\ \underline{\rho}_2 &= \prod_{i=1}^r \eta_i^2 = \frac{\det(\tilde{\mathbf{R}}\mathbf{R}^{-*}\tilde{\mathbf{R}})}{\det \mathbf{R}}, \\ \underline{\rho}_3 &= \frac{1}{n} \sum_{i=1}^r \eta_i^2 = \frac{1}{n} \text{tr}(\mathbf{R}^{-1}\tilde{\mathbf{R}}\mathbf{R}^{-*}\tilde{\mathbf{R}}), \end{aligned} \tag{15}$$

where  $\mathbf{R}^{-*}$  denotes  $(\mathbf{R}^{-1})^*$ . These correlation coefficients all satisfy  $0 \leq \rho \leq 1$ .

However, the dimensionality of  $\varphi(\cdot)$  is generally high, this leads to the operation of correlation matrix unacceptable in practical kernel algorithms. Fortunately, using Gram matrix is an alternative way to solve this problem. Similar to the derivation of the correlation matrix, we firstly define the real Gram matrix of  $\Phi'_{\mathbb{R}} = [\Phi_r, \Phi_j]$  as

$$\mathbf{G}_{\mathbb{R}} = \Phi'^T_{\mathbb{R}} \Phi'_{\mathbb{R}} = \begin{bmatrix} \mathbf{G}_{rr} & \mathbf{G}_{rj} \\ \mathbf{G}_{rj}^T & \mathbf{G}_{jj} \end{bmatrix}, \tag{16}$$

where  $\mathbf{G}_{rr} = \mathbb{E}[\Phi_r^T \Phi_r]$ ,  $\mathbf{G}_{rj} = \mathbb{E}[\Phi_r^T \Phi_j] = \mathbf{G}_{jr}^T$ , and  $\mathbf{G}_{jj} = \mathbb{E}[\Phi_j^T \Phi_j]$ . The elements of these Gram matrix can be denoted as kernel functions  $[\mathbf{G}_{rr}]_{m,n} = \kappa_{rr}(\mathbf{x}_m, \mathbf{x}_n)$ ,  $[\mathbf{G}_{rj}]_{m,n} = \kappa_{rj}(\mathbf{x}_m, \mathbf{x}_n)$ ,  $[\mathbf{G}_{jj}]_{m,n} = \kappa_{jj}(\mathbf{x}_m, \mathbf{x}_n)$ . The augmented form of  $\Phi'_{\mathbb{R}}$  is  $\underline{\Phi}' = \mathbf{T}\Phi'_{\mathbb{R}}$ . Then, the augmented Gram matrix can be derived as.

$$\underline{\mathbf{G}} = \begin{bmatrix} \mathbf{G} & \tilde{\mathbf{G}} \\ \tilde{\mathbf{G}}^* & \mathbf{G} \end{bmatrix}, \tag{17}$$

where the standard Gram matrix  $\mathbf{G}$  and the new pseudo-Gram matrix  $\tilde{\mathbf{G}}$  are defined as

$$\mathbf{G} = \mathbf{G}_{rr} + \mathbf{G}_{jj} + j(\mathbf{G}_{jr} - \mathbf{G}_{rj}), \tag{18}$$

$$\tilde{\mathbf{G}} = \mathbf{G}_{rr} - \mathbf{G}_{jj} + j(\mathbf{G}_{jr} + \mathbf{G}_{rj}). \tag{19}$$

As a complement to the standard Gram matrix, the pseudo-Gram matrix provides the variance between the real and imaginary parts of the non-circular complex signal.

### 3 Algorithm Design

In this section, we discuss the generalized complex kernel affine projection algorithms and the online sparsification methods.

#### 3.1 Generalized Complex Kernel Affine Projection Algorithms

To gain insight, we employ the affine subspace method [23] to derive the AP algorithms. In linear adaptive filters, let  $(\mathbf{x}_1, d_1), (\mathbf{x}_2, d_2), \dots, (\mathbf{x}_K, d_K)$  be a stream of input and desired signal pairs. At the  $k$ th instant, we can define a hyperplane  $\Pi_k = \{\mathbf{w} \in \mathbb{C}^M \mid \langle \mathbf{x}_k, \mathbf{w} \rangle = d_k\}$ , which is orthogonal to  $\mathbf{x}_k$  and passes through the point  $\frac{d_k}{\|\mathbf{x}_k\|^2} \mathbf{x}_k$ . Figure 1 shows a geometric description of the AP algorithms, which uses more recent inputs  $\mathbf{x}_k, \mathbf{x}_{k-1}, \mathbf{x}_{k-P}$  and carries out the affine projection of  $\mathbf{w}_k$  onto the affine subspace  $\Pi^{(k)} = \Pi_k \cap \Pi_{k-1} \cap \dots \cap \Pi_{k-(P-1)}$  to generalize the update

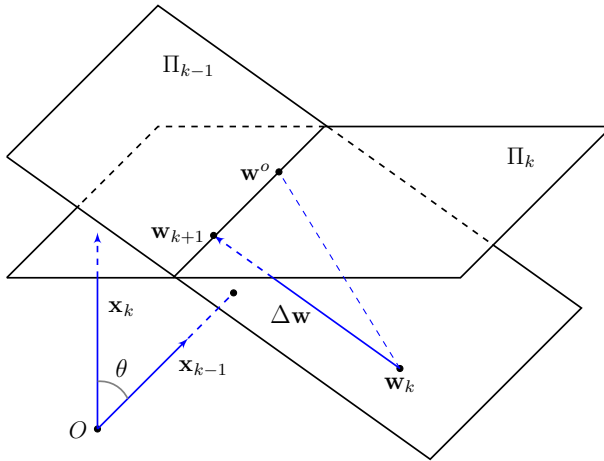


Fig. 1 The geometric description of affine projection algorithms

process of weights as

$$\begin{aligned} \mathbf{w}_{k+1} &= \mathbf{w}_k + \mu \Delta \mathbf{w}_k \\ \Delta \mathbf{w}_k &= \text{Proj}_{\Pi^{(k)}} \mathbf{w}_k - \mathbf{w}_k = \mathbf{X}_k^\dagger (\mathbf{d}_k - \mathbf{X}_k \mathbf{w}_k), \end{aligned} \tag{20}$$

where  $\mathbf{d}_k = [d_k, d_{k-1}, \dots, d_{k-P+1}]^T$ ,  $\mathbf{X}_k = [\mathbf{x}_k, \mathbf{x}_{k-1}, \dots, \mathbf{x}_{k-P+1}]^T$ ,  $\mu$  denotes step update factor, and  $\text{Proj}_{\Pi^{(k)}} \mathbf{w}_k$  denotes the projection of  $\mathbf{w}_k$  onto the affine subspace  $\Pi^{(k)}$ . The Moore–Penrose pseudo-inverse [16] of  $\mathbf{X}_k$  is  $\mathbf{X}_k^\dagger = \mathbf{X}_k^H (\mathbf{X}_k \mathbf{X}_k^H)^{-1}$ . Therefore, the basic AP algorithm, named AP-1, can be derived as

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \mu \mathbf{X}_k^H (\mathbf{X}_k \mathbf{X}_k^H)^{-1} \mathbf{e}_k, \tag{21}$$

where  $\mathbf{e}_k$  is the error defined by

$$\begin{aligned} \mathbf{e}_k &\triangleq \mathbf{d}_k - \mathbf{X}_k \mathbf{w}_k \\ &= \begin{bmatrix} d_k - \langle \mathbf{x}_k, \mathbf{w}_k \rangle \\ d_{k-1} - \langle \mathbf{x}_{k-1}, \mathbf{w}_k \rangle \\ \vdots \\ d_{k-P+1} - \langle \mathbf{x}_{k-P+1}, \mathbf{w}_k \rangle \end{bmatrix}. \end{aligned} \tag{22}$$

Adding the regularized item  $\delta \mathbf{I}_P$  to  $\mathbf{X}_k \mathbf{X}_k^H$  to stabilize the numerical inversion process, we obtain the regularized AP algorithm, named AP-2, which can be derived as

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \mu \mathbf{X}_k^H (\mathbf{X}_k \mathbf{X}_k^H + \delta \mathbf{I}_P)^{-1} \mathbf{e}_k. \tag{23}$$



AP-2 reduces to AP-1 when  $\delta = 0$ . We can map all the inputs into the complex RKHS using the complex feature mapping  $\varphi$ . The update equation of AP-2 in the complex RKHS is

$$\omega_{k+1} = \omega_k + \mu \Phi_k^H (\Phi_k \Phi_k^H + \delta \mathbf{I}_P)^{-1} \mathbf{e}_k, \tag{24}$$

where  $\Phi_k = [\varphi(\mathbf{x}_k), \varphi(\mathbf{x}_{k-1}), \dots, \varphi(\mathbf{x}_{k-P+1})]^T$  and

$$\mathbf{e}_k = \mathbf{d}_k - \Phi_k \omega_k = \begin{bmatrix} d_k - f_k(\mathbf{x}_k) \\ d_{k-1} - f_k(\mathbf{x}_{k-1}) \\ \vdots \\ d_{k-P+1} - f_k(\mathbf{x}_{k-P+1}) \end{bmatrix}. \tag{25}$$

The output  $f_k(\mathbf{x}')$  in (25) follows the inner product of  $\varphi(\mathbf{x}')$  and  $\omega_k$  in RKHS

$$f_k(\mathbf{x}') = \langle \varphi(\mathbf{x}'), \omega_k \rangle_{\mathcal{H}}, \tag{26}$$

where the weight  $\omega_k$  can be expressed as a linear combination of mapped inputs  $\varphi(\mathbf{x}(i))$  in the RKHS

$$\omega_k = \sum_{j=-P+1}^{k-1} a_{k,j} \varphi(\mathbf{x}_j). \tag{27}$$

Define a Gram matrix  $\mathbf{G}_k \triangleq \Phi_k \Phi_k^H \in \mathbb{C}^{P \times P}$ , where  $[\mathbf{G}_k]_{m-k+P, n-k+P} = \kappa(\mathbf{x}_m, \mathbf{x}_n)$ ,  $k - P + 1 \leq m, n \leq k$ , and a new vector

$$\mathbf{b}_k \triangleq \mu (\mathbf{G}_k + \delta \mathbf{I}_P)^{-1} \mathbf{e}_k. \tag{28}$$

Then, the update of the weight in (24) can be rewritten as

$$\begin{aligned} \omega_{k+1} &= \sum_{j=-P+1}^{k-1} a_{k,j} \varphi(\mathbf{x}_j) + \Phi_k^H \mathbf{b}_k \\ &= \sum_{j=-P+1}^{k-1} a_{k,j} \varphi(\mathbf{x}_j) + \sum_{j=k-P+1}^k b_{k,j} \varphi(\mathbf{x}_j) \\ &= \sum_{j=-P+1}^k a_{k+1,j} \varphi(\mathbf{x}_j), \end{aligned} \tag{29}$$

where

$$a_{k+1,j} = \begin{cases} a_{k,j} & -P + 1 \leq j \leq k - P \\ a_{k,j} + b_{k,j-k} & k - P + 1 \leq j \leq k - 1 \\ b_{k,k} & j = k \end{cases}. \tag{30}$$

The output can be rewritten as a linear combination of kernels

$$f_k(\mathbf{x}') = \langle \boldsymbol{\varphi}(\mathbf{x}'), \boldsymbol{\omega}_k \rangle_{\mathcal{H}} = \sum_{j=-P+1}^{k-1} a_{k,j} \kappa(\mathbf{x}', \mathbf{x}_j). \tag{31}$$

This describes the direct form of the regularized CKAP algorithm, named CKAP-2, in which  $\kappa(\mathbf{x}', \mathbf{x}_j)$  is a complex kernel function. When  $\delta = 0$ , CKAP-2 reduces to the basic CKAP algorithm, named CKAP-1. Next, we use the widely linear approach in (8) to obtain the output of the GCKAP algorithms:

$$\begin{aligned} f_k(\mathbf{x}') &= \kappa(\mathbf{x}', \mathcal{D}_{k-1}) \mathbf{a}_k + \tilde{\kappa}(\mathbf{x}', \mathcal{D}_{k-1}) \mathbf{a}_k^* \\ &= \sum_{j=-P+1}^{k-1} \left[ a_{k,j} \kappa(\mathbf{x}', \mathbf{x}_j) + a_{k,j}^* \tilde{\kappa}(\mathbf{x}', \mathbf{x}_j) \right], \end{aligned} \tag{32}$$

where  $\mathcal{D}_{k-1} = \{\mathbf{x}_j\}_{j=-P+1}^{k-1}$  is the training dictionary set when all the inputs are added. The augmented coefficient vector  $\underline{\mathbf{b}}_k = [\mathbf{b}_k^T, \mathbf{b}_k^H]^T$  can be calculated by

$$\underline{\mathbf{b}}_k = \mu(\underline{\mathbf{G}} + \delta \mathbf{I}_{2P})^{-1} \underline{\boldsymbol{\varepsilon}}_k, \tag{33}$$

where  $\underline{\boldsymbol{\varepsilon}}_k = [\boldsymbol{\varepsilon}_k^T, \boldsymbol{\varepsilon}_k^H]^T$  is the augmented error vector. Let  $\underline{\boldsymbol{\Phi}}_k = [\boldsymbol{\Phi}_k^T, \boldsymbol{\Phi}_k^H]^T$  be the augmented inputs. The augmented Gram matrix can be denoted as  $\underline{\mathbf{G}} = \underline{\boldsymbol{\Phi}}_k \underline{\boldsymbol{\Phi}}_k^H \in \mathbb{C}^{2P \times 2P}$ . The inverse operation of  $\underline{\mathbf{G}} + \delta \mathbf{I}_{2P}$  can no longer be solved by the iteration method in [23]. However, the directly inverse operation of the augmented Gram matrix requires the complexity of  $\mathcal{O}(8P^3)$ . We propose a decomposition method to reduce the complexity. We rewrite the block form of  $\underline{\mathbf{G}}$  as

$$\underline{\mathbf{G}} = \begin{bmatrix} \mathbf{G} & \tilde{\mathbf{G}} \\ \tilde{\mathbf{G}}^* & \mathbf{G} \end{bmatrix}, \tag{34}$$

where the standard Gram matrix  $\mathbf{G}$  and the pseudo-Gram matrix  $\tilde{\mathbf{G}}$  are

$$\mathbf{G} = \mathbf{G}_{rr} + \mathbf{G}_{jj} + j(\mathbf{G}_{jr} - \mathbf{G}_{rj}), \tag{35}$$

$$\tilde{\mathbf{G}} = \mathbf{G}_{rr} - \mathbf{G}_{jj} + j(\mathbf{G}_{jr} + \mathbf{G}_{rj}). \tag{36}$$

The real Gram matrices are defined as

$$\begin{aligned} [\mathbf{G}_{rr}]_{m-k+P, n-k+P} &= \kappa_{rr}(\mathbf{x}_m, \mathbf{x}_n), \\ [\mathbf{G}_{rj}]_{m-k+P, n-k+P} &= \kappa_{rj}(\mathbf{x}_m, \mathbf{x}_n), \quad \mathbf{G}_{jr} = \mathbf{G}_{rj}^T, \\ [\mathbf{G}_{jj}]_{m-k+P, n-k+P} &= \kappa_{jj}(\mathbf{x}_m, \mathbf{x}_n), \quad k - P + 1 \leq m, n \leq k. \end{aligned} \tag{37}$$

The inverse of  $\underline{\mathbf{G}}$  can be factored as

$$\underline{\mathbf{G}}^{-1} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{W}^H & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{V}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{G}^{-*} \end{bmatrix} \begin{bmatrix} \mathbf{I} - \mathbf{W} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}, \tag{38}$$

where  $\mathbf{W} = \tilde{\mathbf{G}}\mathbf{G}^{-*}$ ,  $\mathbf{V} = \mathbf{G} - \mathbf{W}\tilde{\mathbf{G}}^*$  denotes the Schur complement [16] of  $\mathbf{G}^*$  within  $\underline{\mathbf{G}}$ , and  $\mathbf{G}^{-*}$  denotes  $(\mathbf{G}^{-1})^*$ . Using the factor of (38), the augmented vector  $\underline{\mathbf{b}}_k$  can be derived as

$$\begin{aligned} \underline{\mathbf{b}}_k &= \mu (\underline{\mathbf{G}} + \delta \mathbf{I}_{2P})^{-1} \underline{\boldsymbol{\varepsilon}}_k \\ &= \mu \begin{bmatrix} \mathbf{G} + \delta \mathbf{I}_P & \tilde{\mathbf{G}} \\ \tilde{\mathbf{G}}^* & \mathbf{G} + \delta \mathbf{I}_P \end{bmatrix}^{-1} \underline{\boldsymbol{\varepsilon}}_k \\ &= \mu \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{W}_d^H & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{V}_d^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{G}_d^{-*} \end{bmatrix} \begin{bmatrix} \mathbf{I} - \mathbf{W}_d \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{\varepsilon}_k \\ \boldsymbol{\varepsilon}_k^* \end{bmatrix} \\ &= \mu \begin{bmatrix} \mathbf{V}_d^{-1} & \mathbf{0} \\ -\mathbf{W}_d^H \mathbf{V}_d^{-1} & \mathbf{G}_d^{-*} \end{bmatrix} \begin{bmatrix} \mathbf{I} - \mathbf{W}_d \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{\varepsilon}_k \\ \boldsymbol{\varepsilon}_k^* \end{bmatrix} \\ &= \mu \begin{bmatrix} \mathbf{V}_d^{-1} & -\mathbf{V}_d^{-1} \mathbf{W}_d \\ -\mathbf{W}_d^H \mathbf{V}_d^{-1} & \mathbf{W}_d^H \mathbf{V}_d^{-1} \mathbf{W}_d + \mathbf{G}_d^{-*} \end{bmatrix} \begin{bmatrix} \boldsymbol{\varepsilon}_k \\ \boldsymbol{\varepsilon}_k^* \end{bmatrix} \\ &= \begin{bmatrix} \mu \mathbf{V}_d^{-1} (\boldsymbol{\varepsilon}_k - \mathbf{W}_d \boldsymbol{\varepsilon}_k^*) \\ -\mu \mathbf{W}_d^H \mathbf{V}_d^{-1} (\boldsymbol{\varepsilon}_k - \mathbf{W}_d \boldsymbol{\varepsilon}_k^*) + \mu \mathbf{G}_d^{-*} \boldsymbol{\varepsilon}_k^* \end{bmatrix}, \end{aligned} \tag{39}$$

where  $\mathbf{G}_d = \mathbf{G} + \delta \mathbf{I}_P$  is the regularized Gram matrix,  $\mathbf{W}_d = \tilde{\mathbf{G}}\mathbf{G}_d^{-*}$ , and  $\mathbf{V}_d = \mathbf{G}_d - \mathbf{W}_d \tilde{\mathbf{G}}^*$ . Therefore, by the upper part of the representation of  $\underline{\mathbf{b}}_k$ , the complex coefficient  $\mathbf{b}_k$  can be further simplified as

$$\mathbf{b}_k = \mu \mathbf{V}_d^{-1} (\boldsymbol{\varepsilon}_k - \mathbf{W}_d \boldsymbol{\varepsilon}_k^*). \tag{40}$$

The vector form iteration of  $\mathbf{a}_k$  can be derived from (30):

$$\mathbf{a}_{k+1} = [\mathbf{a}_k^T, 0]^T + [0, \mathbf{b}_k^T]^T. \tag{41}$$

The final GCKAP-2 algorithm is summarized in Algorithm 1. If the current input satisfies some sparsification criteria,  $(\mathbf{x}_i, d_i)$  is added to the dictionary. Note that GCKAP-2 reduces to GCKAP-1 when the regularized factor  $\delta = 0$ . In addition, the CKAP algorithms can be denoted as particular cases of the GCKAP algorithms when the pseudo-kernel vanishes. The inverse operation of the standard regularized Gram matrix  $\mathbf{G}_d^{-1}$  can be substituted by the iteration form [32], which requires the complexity  $\mathcal{O}(P^2)$ . Through the decomposition method, the complexity of calculating the coefficient vector  $\mathbf{b}_k$  is reduced to  $\mathcal{O}(3P^3)$ . In general, this complexity is acceptable for the AP algorithms because of the limited projection order. The computational costs of complex KAF algorithms are summarized in Table 1, where  $N$  denotes the dictionary size at current iteration and  $P$  denotes the projection order of AP algorithms. As presented in this table, compared with CKAP algorithm, GCKAP algorithm does

**Algorithm 1:** GCKAP-2 algorithm

**Input:**  $(\mathbf{x}_i, d_i) \in \mathbb{C}^M \times \mathbb{C} \ (i = 1, 2, \dots)$   
 $\mathcal{D}_P = [\mathbf{x}_1, \dots, \mathbf{x}_P], \ \mathbf{d}_P = [d_1, \dots, d_P]^T$   
 $[\mathbf{G}_{rr}]_{mn} = \kappa_{rr}(\mathbf{x}_m, \mathbf{x}_n), \ [\mathbf{G}_{rj}]_{mn} = \kappa_{rj}(\mathbf{x}_m, \mathbf{x}_n)$   
 $\mathbf{G}_{jr} = \mathbf{G}_{rj}^T, \ [\mathbf{G}_{jj}]_{mn} = \kappa_{jj}(\mathbf{x}_m, \mathbf{x}_n), \ 1 \leq m, n \leq P$   
 $\mathbf{a}_{P+1} = \mu(\mathbf{G} + \delta \mathbf{I}_P)^{-1} \mathbf{d}_P, \ k = P + 1$   
**if**  $(\mathbf{x}_i, d_i)$  is available **then**  
     $\mathcal{D}_k = [\mathcal{D}_{k-1}, \mathbf{x}_i], \ \mathbf{d}_k = [\mathbf{d}_{k-1}, d_i]$   
    **for**  $p = k - P + 1, \dots, k$  **do**  
         $\mathbf{x}' = \mathcal{D}_k(:, p)$   
         $\kappa(\mathbf{x}', \mathcal{D}_{k-1}) = \frac{1}{2}[\kappa_{rr}(\mathbf{x}', \mathcal{D}_{k-1}) + \kappa_{jj}(\mathbf{x}', \mathcal{D}_{k-1})$   
             $+ j(\kappa_{jr}(\mathbf{x}', \mathcal{D}_{k-1}) - \kappa_{rj}(\mathbf{x}', \mathcal{D}_{k-1}))]$   
         $\tilde{\kappa}(\mathbf{x}', \mathcal{D}_{k-1}) = \frac{1}{2}[\kappa_{rr}(\mathbf{x}', \mathcal{D}_{k-1}) - \kappa_{jj}(\mathbf{x}', \mathcal{D}_{k-1})$   
             $+ j(\kappa_{jr}(\mathbf{x}', \mathcal{D}_{k-1}) + \kappa_{rj}(\mathbf{x}', \mathcal{D}_{k-1}))]$   
         $f_k(\mathbf{x}') = \kappa(\mathbf{x}', \mathcal{D}_{k-1})\mathbf{a}_k + \tilde{\kappa}(\mathbf{x}', \mathcal{D}_{k-1})\mathbf{a}_k^*$   
         $\varepsilon_{k,p} = \mathbf{d}_k(p) - f_k(\mathbf{x}')$   
    **end**  
     $k - P + 1 \leq m, n \leq k$   
     $[\mathbf{G}_{rr}]_{m-k+P, n-k+P} = \kappa_{rr}(\mathbf{x}_m, \mathbf{x}_n)$   
     $[\mathbf{G}_{rj}]_{m-k+P, n-k+P} = \kappa_{rj}(\mathbf{x}_m, \mathbf{x}_n), \ \mathbf{G}_{jr} = \mathbf{G}_{rj}^T$   
     $[\mathbf{G}_{jj}]_{m-k+P, n-k+P} = \kappa_{jj}(\mathbf{x}_m, \mathbf{x}_n)$   
     $\mathbf{G}_d = \mathbf{G}_{rr} + \mathbf{G}_{jj} + j(\mathbf{G}_{jr} - \mathbf{G}_{rj}) + \delta \mathbf{I}_P$   
     $\tilde{\mathbf{G}} = \mathbf{G}_{rr} - \mathbf{G}_{jj} + j(\mathbf{G}_{jr} + \mathbf{G}_{rj})$   
     $\mathbf{W}_d = \tilde{\mathbf{G}}\mathbf{G}_d^{-*}, \ \mathbf{V}_d = \mathbf{G}_d - \mathbf{W}_d\tilde{\mathbf{G}}^*$   
     $\mathbf{b}_k = \mu\mathbf{V}_d^{-1}(\varepsilon_k - \mathbf{W}_d\mathbf{e}_k^*)$   
     $\mathbf{a}_{k+1} = [\mathbf{a}_k^T, 0]^T + [\mathbf{0}_{1 \times (k-P)}, \mathbf{b}_k^T]^T$   
     $k = k + 1$   
**end**

**Table 1** Computational cost of complex KAF algorithms

Algorithm	Computational complexity
KLMS	$N$
CKLMS	$4N$
GCKLMS	$4N$
KAP	$PN + 2P^2$
CKAP	$4PN + 16P^2$
GCKAP	$4PN + 3P^3$

not add much extra computational cost for small projection order. Widely linear KAF algorithms (GCKLMS and GCKAP) have almost the same computational cost to the pure complex KAF algorithms.

### 3.2 Online Sparsification Methods

Several online sparsification methods have been proposed to overcome the infinite growth of the dictionary set while keeping the remaining data sufficiently well. The novelty criterion (NC) calculates the distance from a new data point to the current dictionary. The approximate linear dependency (ALD) criterion measures how well the data can be approximated in the RKHS as a linear combination of the dictionary set. However, the ALD criterion is not suitable for KLMS and KAP due to the quadratic complexity. The coherence sparsification (CS) criterion checks the similarity by the kernel function between the new data and the dictionary set. Recently, the angle sparsification (AS) criterion was proposed in [38], which defined the geometric structure in the RKHS by inner production. We add this criterion to the GCKAP algorithms.

The basic idea of the angle criterion is to define the angles among functions in the feature space as the sparsification criterion. The cosine of the angle between  $\varphi(\mathbf{x})$  and  $\varphi(\mathbf{y})$  is defined by

$$v(\mathbf{x}, \mathbf{y}) = \frac{\langle \varphi(\mathbf{x}), \varphi(\mathbf{y}) \rangle_{\mathcal{H}}}{\|\varphi(\mathbf{x})\|_{\mathcal{H}} \|\varphi(\mathbf{y})\|_{\mathcal{H}}} = \frac{\kappa(\mathbf{y}, \mathbf{x})}{\sqrt{\kappa(\mathbf{x}, \mathbf{x})\kappa(\mathbf{y}, \mathbf{y})}}. \quad (42)$$

Suppose the current dictionary is  $\mathcal{D} = \{(\varphi(\tilde{\mathbf{x}}_k), \tilde{d}_k)\}_{k=1}^N$  and a new sample  $(\mathbf{x}_i, d_i)$  is coming. The procedure of the angle criterion can be described as follows. First, the parameter

$$v_i = \max_{1 \leq k \leq N} |v(\mathbf{x}_i, \tilde{\mathbf{x}}_k)| \in [0, 1] \quad (43)$$

is calculated. Second, if  $v_i$  is smaller than a predefined threshold  $v_0$ ,  $(\varphi(\mathbf{x}_i), d_i)$  is added to  $\mathcal{D}$ ; otherwise, it is discarded. The parameter  $v_0$  controls the level of similarity among the elements in  $\mathcal{D}$  and is called the similarity parameter.

## 4 Simulation Experiments

In this section, we compare the performance of the proposed GCKAP algorithms with that of the CKLMS2 [6], GCKLMS [3], CKAP-1 and CKAP-2 [22] algorithms in nonlinear channel equalization, as shown in Fig. 2. The nonlinear channel consists of a linear filter followed by a memoryless strong or soft nonlinearity. At the end of the channel, the signal is corrupted by additive noise. The equalizer performs an inverse filtering of the received signal  $r_k$  to recover the input signal  $s_k$  with as small error as possible. Two experiments are considered: (a) a strong nonlinear channel equalization with non-circular Gaussian distributed signal input and (b) a soft nonlinear channel equalization with QPSK input. In all experiments, the complex Gaussian kernel functions [6] are used in CKLMS2, CKAP-1 and CKAP-2:

$$\kappa_{\mathbb{C}} = \exp\left(-(\mathbf{x} - \mathbf{x}'^*)^T (\mathbf{x} - \mathbf{x}'^*) / \gamma^2\right), \quad (44)$$

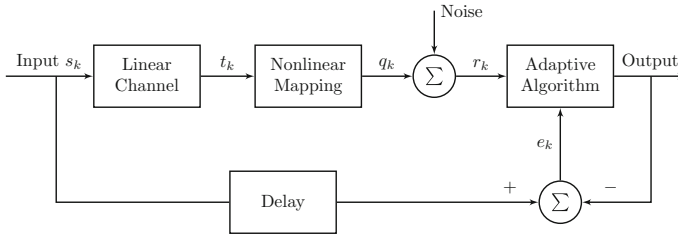


Fig. 2 The nonlinear channel equalization

where  $\gamma$  is the kernel parameter. The real Gaussian kernel functions [3] with complex inputs are used in GCKLMS, GCKAP-1 and GCKAP-2:

$$\kappa_{\mathbb{R}} = \exp\left(-(\mathbf{x} - \mathbf{x}')^H (\mathbf{x} - \mathbf{x}') / \gamma^2\right). \tag{45}$$

### 4.1 Strong Nonlinear Channel Equalization

In the first experiment, we reproduce the nonlinear channel equalization task in [7]. The channel consists of a linear filter

$$\begin{aligned} t_k &= (-0.9 + 0.8j)s_k + (0.6 - 0.7j)s_{k-1} \\ &\quad + (-0.4 + 0.3j)s_{k-2} + (0.3 - 0.2j)s_{k-3} \\ &\quad + (-0.1 - 0.2j)s_{k-4} \end{aligned}$$

and a memoryless nonlinearity

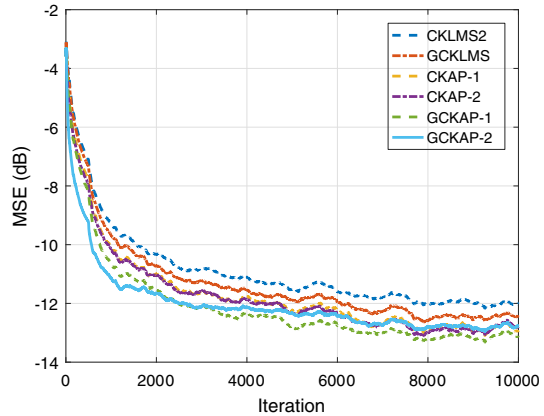
$$q_k = t_k + (0.2 + 0.25j)t_k^2 + (0.08 + 0.09j)t_k^3.$$

At the end of the channel, the signal is corrupted by white circular Gaussian noise and then observed as  $r_k$ . The signal-to-noise ratio (SNR) is set to 15 dB. The non-circular Gaussian distributed signal as the input is fed into the channel:

$$s_k = 0.7(\sqrt{1 - \rho^2} \cdot x_k + j\rho \cdot y_k),$$

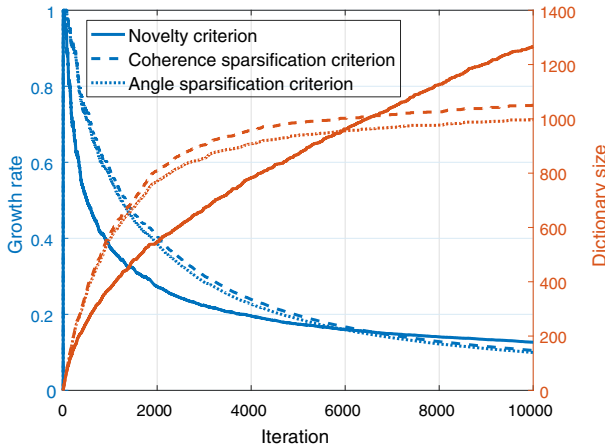
where  $x_k$  and  $y_k$  are independent Gaussian random variables, with  $\rho = 0.1$  for non-circular input signals. The vector  $\mathbf{x} = [r_{k+D}, r_{k+D-1}, \dots, r_{k+D-L+1}]^T$  is the training samples of the equalizer, where  $L$  and  $D$  are the length and delay of the equalizer, respectively. Here, we set  $L = 5$ ,  $D = 2$ . The equalizer is conducted on 100 independent trails of 10000 samples of the input signal. The purpose of the equalizer is to estimate the original input signal. We set the step update factor  $\mu = 1/6$  for all algorithms. In the AP algorithms, we set the order of projection  $P = 4$ . The kernel parameters are set as  $\gamma = 4$  for a complex kernel and as  $\gamma_r = 5$ ,  $\gamma_j = 3$  for a real kernel and pseudo-kernel, respectively.

**Fig. 3** Learning curves of CKLMS and CKAP for strong nonlinear channel equalization with the novelty criterion



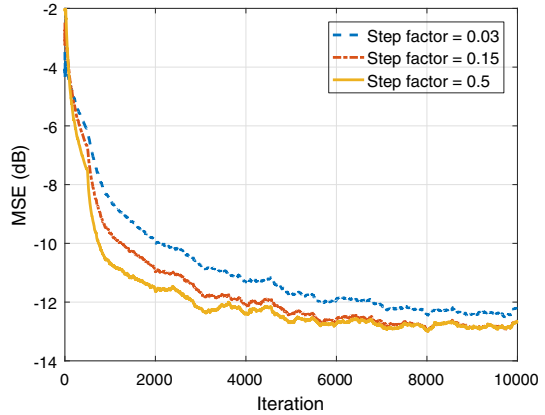
The average MSEs are shown in Fig. 3. The novelty criterion is used for the sparsification of all algorithms with  $\delta_1 = 0.15$ ,  $\delta_2 = 0.2$ . It can be seen that the convergence rates of the CKAP and GCKAP algorithms are faster than that of the CKLMS algorithms. In addition, the GCKAP algorithms provide the smallest steady-state MSE. To compare the online sparsification criteria, the dictionary size and growth rate of GCKAP-2 are shown in Fig. 4. The similarity parameter is chosen as  $\nu_0 = 0.9$  for the coherence and the angle criteria. It can be seen that the growth rate drops dramatically from around 1 to 0.1. To achieve almost the same steady-state MSE, more data are selected for the dictionary with the novelty criterion. Only round 1030 inputs out of 10,000 (10.3%) are eventually selected into the dictionary with the coherence criterion and angle criterion. We know that the coherence and the angle criteria perform in the RKHS, while the novelty criterion works in the original space. Therefore, the criterion in the RKHS can represent the space more accurately than that in the original space.

The simulations of the effect of step size factor and projection order of the GCKAP-2 are shown in Figs. 5 and 6, respectively. The step size factor is set to 0.03, 0.15, 0.5 from small to large. The simulation results show that the larger the step size factor, the faster the convergence rate of the algorithm, and vice versa. The projection order of the algorithm is set to 1, 2, and 5, respectively. The GCKAP algorithm reduces to GCKLMS when the projection order is 1. The simulation results show that the larger the projection order, the faster the convergence rate of the algorithm. It can be seen that the effect of step size factor and projection order of the GCKAP-2 algorithm is consistent with the traditional AP algorithm. However, in the RKHS, the kernel algorithms need to use the growing dictionary set to construct the nonlinear feature space. It will further lead to an increase in the misadjustment in steady state. This stack effect reduces the difference between the misadjustment behaviors under the various parameter settings.



**Fig. 4** Dictionary size and growth rate of GCKAP-2 with the novelty criterion, coherence criterion, and angle criterion

**Fig. 5** The effect of step size factor of the GCKAP-2 algorithm



### 4.2 Soft Nonlinear Channel Equalization with the QPSK Input

In this experiment, we consider a soft nonlinear channel equalization task with the QPSK input. The channel consists of a linear filter

$$t_k = s_k + (0.5 - 0.5j)s_{k-1} + (0.1 + 0.1j)s_{k-2}$$

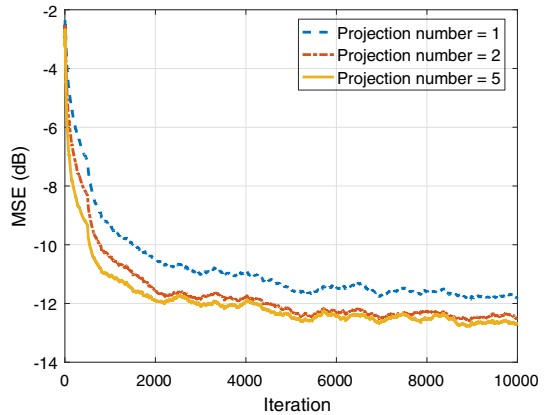
and a memoryless nonlinearity

$$q_k = t_k + (0.1 + 0.15j)t_k^2.$$

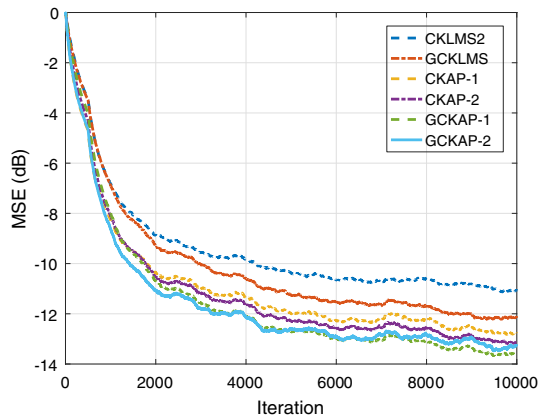
The QPSK input signal is  $s_k = x_k + jy_k$ , where  $x_k$  and  $y_k$  are independent binary  $\{-1, +1\}$  data streams. The channel parameters are set the same as strong nonlinear channel equalization. The step update factor and the order of projection also remain



**Fig. 6** The effect of projection number of the GCKAP-2 algorithm



**Fig. 7** Learning curves of CKLMS and CKAP for soft nonlinear channel equalization with the QPSK input

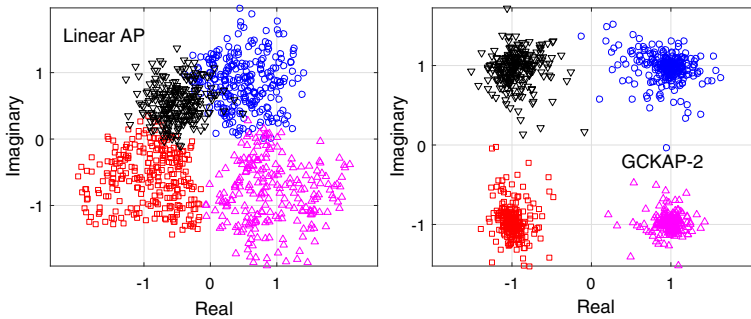


the same. We set the kernel parameter  $\gamma = 2.7$  for the complex kernel and  $\gamma_r = 2.3$ ,  $\gamma_j = 2.1$  for the real kernel and pseudo-kernel, respectively.

The average MSE learning curves are plotted in Fig. 7. It can be shown that the proposed GCKAP-1 and GCKAP-2 algorithms outperform the other algorithms. This reflects the main advantage of GCKAP algorithms: the kernel and pseudo-kernel provide more representation to the QPSK improper signal. The estimated symbols from the training data are shown in Fig. 8. As can be seen, the linear AP algorithm produces a poor estimation of the symbols; however, the GCKAP-2 algorithm can model and invert this nonlinear behavior.

### 5 Conclusions

The generalized complex affine projection algorithms in the WL-RKHS were developed in this paper. The proposed GCKAP algorithms retain the simplicity while outperforming the performance of the CKLMS algorithms. In addition, they work in the WL-RKHS, providing the complete solution for both circular and non-circular



**Fig. 8** Estimated symbols from the training data by the linear AP (left) and GCKAP-2 (right)

complex nonlinear problems. After discussing the statistical properties of the WL-RKHS, the augmented Gram matrix, which includes the standard Gram matrix and the pseudo-Gram matrix, was proposed to calculate the expansion coefficients in the GCKAP algorithms. As a complement to the standard Gram matrix, the pseudo-Gram matrix provides the variance between the real and imaginary parts of a non-circular signal. In addition, with the proposed decomposition method, the complexity of the inverse operation of the augmented Gram matrix is reduced. Finally, our simulation results show that the sparsification criterion in the RKHS can represent the space more accurately than that in the original space.

**Acknowledgements** This work was supported in part by the China Postdoctoral Science Foundation under Grant 2018M640990 and in part by the Fundamental Research Funds for the Central Universities under Grant xjh012019041.

**Data availability** The data that support the findings of this study are available from the corresponding author on request.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. M.Z.A. Bhotto, A. Antoniou, Affine-projection-like adaptive-filtering algorithms using gradient-based step size. *IEEE Trans. Circuits Syst.* **61**(7), 2048–2056 (2014)
2. R. Boloix-Tortosa, J.J. Murillo-Fuentes, I. Santos, F. Pérez-Cruz, Widely linear complex-valued kernel methods for regression. *IEEE Trans. Signal Process.* **65**(19), 5240–5248 (2017)
3. R. Boloix-Tortosa, J.J. Murillo-Fuentes, S.A. Tsiftaris, The generalized complex kernel least-mean-square algorithm. *IEEE Trans. Signal Process.* **67**(20), 5213–5222 (2019)
4. P. Bouboulis, K. Slavakis, S. Theodoridis, Adaptive learning in complex reproducing kernel Hilbert spaces employing Wirtinger's subgradients. *IEEE Trans. Neural Netw. Learn. Syst.* **23**(3), 425–438 (2012)

5. P. Bouboulis, S. Theodoridis, The complex Gaussian kernel LMS algorithm, in *Inter. Conf. on Artifi. Neural Netw.* (Springer, Berlin, 2010), pp. 11–20
6. P. Bouboulis, S. Theodoridis, Extension of Wirtinger’s calculus to reproducing kernel Hilbert spaces and the complex kernel LMS. *IEEE Trans. Signal Process.* **59**(3), 964–978 (2010)
7. P. Bouboulis, S. Theodoridis, M. Mavroforakis, The augmented complex kernel LMS. *IEEE Trans. Signal Process.* **60**(9), 4962–4967 (2011)
8. E. Castro, V. Gomezverdejo, M. Martinezramon, K.A. Kiehl, V.D. Calhoun, A multiple kernel learning approach to perform classification of groups from complex-valued fMRI data analysis: application to schizophrenia. *Neuro Image* **87**, 1–17 (2014)
9. B. Chen, J. Liang, N. Zheng, J.C. Principe, Kernel least mean square with adaptive kernel size. *Neurocomputing* **191**, 95–106 (2016)
10. B. Chen, S. Zhao, P. Zhu, J.C. Principe, Mean square convergence analysis for kernel least mean square algorithm. *Signal Process.* **92**(11), 2624–2632 (2012)
11. B. Chen, S. Zhao, P. Zhu, J.C. Principe, Quantized kernel least mean square algorithm. *IEEE Trans. Neural Netw.* **23**(1), 22–32 (2012)
12. L. Dang, B. Chen, S. Wang, Y. Gu, J.C. Principe, Kernel Kalman filtering with conditional embedding and maximum correntropy criterion. *IEEE Trans. Circuits Syst. I Regul. Pap.* **66**(99), 4265–4277 (2019)
13. A.A. De Lima, R.C. De Lamare, Adaptive detection using widely linear processing with data reusing for DS-CDMA systems, in *2006 IEEE Inter. Telecom. Sympos.* (2006), pp. 187–192
14. Y. Engel, S. Mannor, R. Meir, The kernel recursive least-squares algorithm. *IEEE Trans. Signal Process.* **52**(8), 2275–2285 (2004)
15. J.M. Gilcacho, M. Signoretto, T. Van Waterschoot, M. Moonen, S.H. Jensen, Nonlinear acoustic echo cancellation based on a sliding-window leaky kernel affine projection algorithm. *IEEE Trans. Audio Speech Lang. Process.* **21**(9), 1867–1878 (2013)
16. G.H. Golub, C.F.V. Loan, *Matrix Computations*, 4th edn. (Johns Hopkins University Press, Baltimore, 2013)
17. W. Liu, P.P. Pokharel, J.C. Principe, The kernel least-mean-square algorithm. *IEEE Trans. Signal Process.* **56**(2), 543–554 (2008)
18. W. Liu, J.C. Principe, Kernel affine projection algorithms. *EURASIP J. Adv. Signal Process.* **2008**, 1–12 (2008)
19. W. Liu, J.C. Principe, S. Haykin, *Kernel Adaptive Filtering: A Comprehensive Introduction*, vol. 57 (John Wiley & Sons, Hoboken, 2011)
20. Y. Liu, C. Sun, S. Jiang, A reduced Gaussian kernel least-mean-square algorithm for nonlinear adaptive signal processing. *Circuits Syst. Signal Process.* **38**(1), 371–394 (2019)
21. C.A. Micchelli, M. Pontil, On learning vector-valued functions. *Neural Comput.* **17**(1), 177–204 (2005)
22. T. Ogunfunmi, T. Paul, *On the complex kernel-based adaptive filter*, in 2011 IEEE Inter. Sympos. Circuits Syst. (2011), pp. 1263–1266
23. K. Ozeki, *Theory of Affine Projection Algorithms for Adaptive Filtering* (Springer, Tokyo, 2016)
24. A. Papaioannou, S. Zafeiriou, Principal component analysis with complex kernel: The widely linear model. *IEEE Trans. Neural Netw. Learn. Syst.* **25**(9), 1719–1726 (2014)
25. T. Paul, T. Ogunfunmi, *Analysis of the convergence behavior of the complex Gaussian kernel LMS algorithm*, in 2011 IEEE Inter. Sympos. Circuits Syst. (2012), pp. 2761–2764
26. C. Richard, J.C.M. Bermudez, P. Honeine, Online prediction of time series data with kernels. *IEEE Trans. Signal Process.* **57**(3), 1058–1067 (2008)
27. J.L. Rojo-Álvarez, M. Martínez-Ramón, J. Muñoz-Marí, G. Camps-Valls, *Digital Signal Processing with Kernel Methods* (Wiley-IEEE, Hoboken, 2018)
28. B. Schölkopf, A.J. Smola, F. Bach et al., *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond* (MIT press, Massachusetts, 2002)
29. P.J. Schreier, A unifying discussion of correlation analysis for complex random vectors. *IEEE Trans. Signal Process.* **56**(4), 1327–1336 (2008)
30. P.J. Schreier, L.L. Scharf, *Statistical Signal Processing of Complex-Valued Data: The Theory of Improper and Noncircular Signals* (Cambridge University Press, New York, 2010)
31. M.A. Takizawa, M. Yukawa, Adaptive nonlinear estimation based on parallel projection along affine subspaces in reproducing kernel Hilbert space. *IEEE Trans. Signal Process.* **63**(16), 4257–4269 (2015)
32. L.N. Trefethen, D. Bau, *Numerical Linear Algebra* (SIAM, Philadelphia, 1997)
33. S. Van Vaerenbergh, M. Lazarogredilla, I. Santamaria, Kernel recursive least-squares tracker for time-varying regression. *IEEE Trans. Neural Netw.* **23**(8), 1313–1326 (2012)

34. S. Wang, W. Wang, L. Dang, Y. Jiang, Kernel least mean square based on the Nyström method. *Circuits Syst. Signal Process.* **38**(7), 3133–3151 (2019)
35. Z. Wu, J. Shi, X. Zhang, W. Ma, B. Chen, Kernel recursive maximum correntropy. *Signal Process.* **117**, 11–16 (2015)
36. Y. Xia, C.C. Took, D.P. Mandic, An augmented affine projection algorithm for the filtering of noncircular complex signals. *Signal Process.* **90**(6), 1788–1799 (2010)
37. M. Yukawa, Multikernel adaptive filtering. *IEEE Trans. Signal Process.* **60**(9), 4672–4682 (2012)
38. M. Zhang, X. Wang, X. Chen, A. Zhang, The kernel conjugate gradient algorithms. *IEEE Trans. Signal Process.* **66**(16), 4377–4387 (2018)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.