NONLINEAR ACOUSTIC ECHO CANCELLATION BASED ON A PARALLEL-CASCADE KERNEL AFFINE PROJECTION ALGORITHM

Jose M. Gil-Cacho, Toon van Waterschoot, Marc Moonen * *Søren Holdt Jensen* †

K.U.Leuven, Dept. ESAT-SISTA,
3001 Leuven, Belgium.

U. Aalborg, Dept. Elec. Syst.,
DK-9220 Aalborg, Denmark

ABSTRACT

In acoustic echo cancellation (AEC) applications, oftentimes an acoustic path from a loudspeaker to a microphone is estimated by means of a linear adaptive filter. However, loudspeakers introduce nonlinear distortions which may strongly degrade the adaptive filter performance, thus nonlinear filters have to be considered. This paper proposes two adaptive algorithms namely the parallel and cascade sliding-window kernel based affine projection algorithm (PSW-KAPA and CSW-KAPA) to solve the problem of nonlinear AEC (NLAEC) while keeping the computational complexity low. They are based on a leaky KAPA which employs the theory and algorithms of kernel methods. The basic concept is to perform adaptive filtering in a linear space that is nonlinearly related to the original input space. A kernel specifically designed for acoustic applications is proposed, which consists in a weighted sum of the linear and the Gaussian kernels. The motivation is basically to separate the problem into linear and nonlinear subproblems. The weights in the kernel also impose different forgetting mechanisms in the sliding window which in turn translates to a more flexible regularization. Simulation results show that PSW-KAPA and CSW-KAPA consistently outperform the linear NLMS, and generalize well both in high and low linear to nonlinear ratio (LNLR).

Index Terms— Kernel adaptive filters, Nonlinear Acoustic Echo Cancellation.

1. INTRODUCTION

Acoustic echo cancellation (AEC) is of great importance in many practical systems for instance for mobile communications, hands-free telephony inside a car or in teleconferencing where the existence of echoes degrades speech intelligibility and listening comfort. Standard approaches to AEC rely on the assumption that the echo path to be identified can be modeled by a linear filter. However, loudspeakers (and also amplifiers, DACs, coders..) introduce nonlinear distortions and must be considered as nonlinear systems; therefore nonlinear adaptive filters should be used instead. Several nonlinear models meant to overcome the limitations of linear filters have been implemented with more or less success [1][2]. The main problem of these implementations usually resides in the fact that many more parameters are needed than in the linear case. Truncated Volterra filters are a common solution although only very low nonlinear degrees are considered due to complexity constraints.

Kernel adaptive algorithms [3][4] and on-line learning algorithms [5] have been subject of great attention due to their good performance in nonlinear signal processing applications. Kernel methods are developed based on the theory of reproducing kernel Hilbert spaces (RKHS) [6] to implement a nonlinear transformation of the input data into a high-dimensional feature space via a reproducing kernel. If the adaptive filtering operations can be expressed by inner products of input samples, then it is possible to apply the so called kernel trick. The power of this idea is that while the solution, which is a nonlinear function of the input data, is implicitly obtained in the feature space, it is calculated by applying linear methods on the transformed data.

Kernel affine projection algorithms (KAPA) [3] has been successfully applied to nonlinear equalization, nonlinear system identification and nonlinear noise cancellation as well as prediction of nonlinear time series. Its application in nonlinear acoustic echo cancellation (NLAEC) is, however, lacking so far. In the former examples the time span (i.e., input dimension or filter length) is typically very small, e.g. a few taps. Conversely, in NLAEC applications the input dimension is very long which makes the direct application of KAPA impractical. The aim of the paper is therefore two-fold: first to apply KAPA to the NLAEC problem and second to develop algorithms that are efficient in NLAEC applications. To this end a leaky KAPA [3], which is the basis to obtain a sliding-window KAPA (SW-KAPA), is derived. Moreover, a kernel specifically designed for acoustic applications is proposed, which consists in a weighted sum of the linear and the Gaussian kernels. The motivation is basically to separate the problem into linear and nonlinear subproblems. The weights in the kernel also impose different forgetting mechanisms in the sliding window which in turn translates to a more flexible regularization. Using the proposed kernel, two structures are proposed to reduce the computational burden of SW-KAPA namely parallel and cascade SW-KAPA (PSW-KAPA and CSW-KAPA). Simulation results show that PSW-KAPA and CSW-KAPA consistently outperform the linear NLMS, and generalize well both in high and low linear to nonlinear ratio (LNLR).

The paper is organized as follows: in Section 2 the necessary theory of kernel methods to derive the SW-KAPA algorithms and the proposed kernel is presented. A detailed description of the proposed algorithms and structures is given in Section 3. In Section 4 these are applied to tackle the NLAEC problem and some results are presented. Finally, Section 5 summarizes the main conclusions.

2. NONLINEAR ACOUSTIC ECHO CANCELLATION

2.1. Affine Projection Algorithm (APA)

The affine projection algorithm (APA) [7] is a good compromise between NLMS and RLS. It is adopted in AEC applications due to its

*This research work was carried out at the ESAT Laboratory of Katholieke Universiteit Leuven, in the frame of K.U.Leuven Research Council CoE EF/05/006 'Optimization in Engineering' (OPTEC) and PFV/10/002 (OPTEC), Concerted Research Action GOA-MaNet, the Belgian Programme on Interuniversity Attraction Poles initiated by the Belgian Federal Science Policy Office IUAP P6/04 'Dynamical systems, control and optimization' (DYSCO) 2007-2011, Research Project FWO nr. G.0600.08 'Signal processing and network design for wireless acoustic sensor networks', EC-FP6 project 'Core Signal Processing Training Program' (SIGNAL). The scientific responsibility is assumed by its authors

†Aalborg University, Dept. Electrical Systems,

improved convergence performance and tracking capabilities compared to LMS, while being less complex than RLS. It belongs to the class of stochastic gradient algorithms which replace the covariance matrix and the cross-covariance vector of the optimal Wiener solution at each iteration by a local approximation. While the LMS algorithm simply uses instantaneous values, APA employs better approximations by using the P most recent inputs and observations.

2.2. Kernel Affine Projection Algorithm (KAPA)

A kernel [6] is a continuous, symmetric, positive-definite function $\mathbf{k} : \mathbb{U} \times \mathbb{U} \rightarrow \mathbb{R}$. $\mathbb{U}$ is the input domain, a compact subset of $\mathbb{R}^L$. Mercer's theorem [6] states that any kernel $\mathbf{k}(\mathbf{u}(i), \mathbf{u}(j))$ can be expanded as $\mathbf{k}(\mathbf{u}(i), \mathbf{u}(j)) = \sum_{k=1}^{\infty} \zeta_k \phi_k(\mathbf{u}(i)) \phi_k(\mathbf{u}(j))$ where ζ_k and ϕ_k are the eigenvalues and the eigenfunctions, respectively, where the eigenvalues are nonnegative. Therefore, a mapping $\mathbf{f}(\cdot)$ can be constructed as $\mathbf{f}(\mathbf{u}(i)) = [\sqrt{\zeta_1}\phi_1(\mathbf{u}(i)), \sqrt{\zeta_2}\phi_2(\mathbf{u}(i))...]$ such that

$$\mathbf{k}(\mathbf{u}(i), \mathbf{u}(j)) = \mathbf{f}(\mathbf{u}(i))^T \mathbf{f}(\mathbf{u}(j)) \tag{1}$$

The Mercer theorem is employed to transform the input signal vector $\mathbf{u}(i)$ into $\mathbf{f}(\mathbf{u}(i))$ in a high-dimensional feature space $\mathbb{F}$. It naturally allows to formulate the least-squares (LS) cost in the feature space as,

$$J(l) = \arg\min_{\mathbf{w}} \sum_{i=1}^{l} \left| d(i) - \mathbf{w}^T \mathbf{f}(i) \right|^2 \tag{2}$$

In the sequel a simplified notation $\mathbf{f}(i) = \mathbf{f}(\mathbf{u}(i))$ and $\mathbf{k}(\mathbf{u}(i), \mathbf{u}(j)) = \mathbf{k}(i, j)$ is adopted for compactness. The use of a high-dimensional space provides kernel methods with a very high degree of flexibility in solving minimization problems [4]. However this appealing characteristic may cause the solution to perfectly fit any given input-output data set while it will not generalize well to new incoming data. This problem so called overfitting, is specially so if the Gaussian kernel is used and no precautions are taken. In order to prevent this overfitting, the solution should be regularized, which is commonly achieved adding a constraint on the L_2 norm of the solution [3],[4],[5]. By introducing the regularization λ, the complexity of the solution will be limited, and as a result, it will generalize better to new data points. The regularized LS problem on the data $\{d(1), d(2), ...\}$ and $\{\mathbf{f}(1), \mathbf{f}(2), ...\}$ can be formulated in the feature space as

$$J'(l) = \arg\min_{\mathbf{w}} \sum_{i=1}^{l} \left| d(i) - \mathbf{w}^T \mathbf{f}(i) \right|^2 + \lambda \left\| \mathbf{w} \right\|^2 \tag{3}$$

where λ is the regularization parameter. The APA is then formulated in the feature space to solve for $\mathbf{w}$ thus resulting in the so called Leaky KAPA [3],

$$\mathbf{e}(i) = \mathbf{d}(i) - \mathbf{y}(i) = \mathbf{d}(i) - \mathbf{\Phi}(i)^T \mathbf{w}(i-1) \tag{4}$$

$$\mathbf{w}(i) = (1 - \lambda\mu)\mathbf{w}(i-1) + \mu \mathbf{\Phi}(i) \mathbf{G}(i) \mathbf{e}(i) \tag{5}$$

$$\mathbf{G}(i) = \left[\mathbf{\Phi}(i)^T \mathbf{\Phi}(i) + \lambda \mathbf{I} \right]^{-1} \tag{6}$$

where $\mathbf{e}(i) = [e(i), e(i-1), \ldots, e(i-P+1)]$, $\mathbf{\Phi}(i) = [\mathbf{f}(i), \mathbf{f}(i-1), \ldots, \mathbf{f}(i-P+1)]$ and $\mathbf{I}$ is the identity matrix. As discussed before, $\mathbf{w}^T \mathbf{f}(i)$ is a much more powerful model than the usual $\mathbf{h}^T \mathbf{u}(i)$ because of the transformation from $\mathbf{u}(i)$ to $\mathbf{f}(i)$. Finding $\mathbf{w}$ through APA may prove to be an effective way of nonlinear filtering. The solution $\mathbf{w}$ can also be represented in the basis defined by the transformed data vectors $\mathbf{f}(i)$ [4] as

$$\mathbf{w}(i-1) = \sum_{j=1}^{i-1} \mathbf{a}_j(i-1) \mathbf{f}(j), \qquad \forall i > 0, \tag{7}$$

that is, the weight vector at time $i-1$ is a linear combination of all previous transformed input vectors with a vector of expansion coefficients $\mathbf{a}$ defined below. It is here where the "kernel trick" is exploited: Given $\mathbf{w}(i-1)$ from (7) and the transformed input matrix $\mathbf{\Phi}(i)$, the output vector at time i (see 4) is given as

$$\mathbf{y}(i) = \mathbf{\Phi}(i)^T \mathbf{w}(i-1) = \mathbf{\Phi}(i) \sum_{j=1}^{i-1} \mathbf{a}_j(i-1) \mathbf{f}(j)$$

$$= \sum_{j=1}^{i-1} \mathbf{a}_j(i-1) \left[\mathbf{\Phi}(i)^T \mathbf{f}(j) \right] = \sum_{j=1}^{i-1} \mathbf{a}_j(i-1) \mathbf{k}(i-P+1:i, j)$$

In practice there is no access to the weight vector $\mathbf{w}$ since it is in the (possibly) infinite dimensional feature space $\mathbb{F}$ and it would be then practically impossible to update for $\mathbf{w}$ directly [3][5]. Besides, $\mathbf{f}$ is only implicitly known (i.e., it is the kernel's eigenfunctions), so by (7) the updating of the weight vector reduces to the updating on the expansion coefficients $\mathbf{a}_p(i)$ as

$$\mathbf{a}_p(i) = \begin{cases} \mu e_{i+1-p}(i) \mathbf{G}(i), & \text{if} \quad p = i \\ (1 - \lambda\mu)\mathbf{a}_p(i-1) + \mu e_{i+1-p}(i) \mathbf{G}(p), & \\ & \text{if} \quad i - P + 1 \leq p \leq i - 1 \\ (1 - \lambda\mu)\mathbf{a}_p(i-1), & \text{if} \quad 1 \leq p < i - P + 1. \end{cases}$$

where $e_{i+1-p}(i) = \left(d(p) - \sum_{j=1}^{i-1} \mathbf{a}_j(i-1) \mathbf{k}(p, j) \right)$ is the prediction error normalized by the $P \times P$ matrix $\mathbf{G}(i)$. Details for the complete derivation of $\mathbf{a}_p(i)$ can be found in [3]. So far nothing has been said about pruning the memory buffers to make the problem size fixed; actually in (7) the memory buffers grows linearly as new data arrives up to time i. In the NORMA algorithm [5], which is equivalent to a kernel version of leaky LMS, this is solved by truncating the kernel expansion coefficients: since at each instant i, the expansion coefficients are scaled by $(1 - \lambda\mu)$, which is less than 1, the oldest terms can be dropped without incurring significant error. This truncation scheme fundamentally converts NORMA into a sliding-window kernel LMS (SW-KLMS) algorithm. Following the spirit of NORMA this paper uses leaky KAPA [3] to obtain a sliding-window kernel APA (SW-KAPA) to solve the problem of NLAEC. The implementation of the leaky KAPA here differs from that of [3] in that in NLAEC applications the error signal is indeed computed explicitly as this is the signal sent back to the far-end.

2.3. Weighted Sum of Kernels

The choice of the kernel is vital in the development of different algorithms and the rationale behind the choice may be multiple. For instance, one of the most commonly used kernels is the Gaussian kernel since its performance is superior than that of other kernels, for instance, the polynomial kernel. However in [8] polynomial kernels are the preferred choice since the obtained solutions can be directly transformed into their corresponding Wiener or Volterra representation. In NLAEC applications the system impulse response is usually of very high order, e.g., hundreds of taps in mobile communication systems and even thousands of taps in room acoustics applications. The size of these problems makes the direct application of most of the kernels e.g. Gaussian kernels, impractical for real-time applications.

In this paper a kernel which consists in a weighted sum of the linear and the Gaussian kernels is proposed:

$$\mathbf{k}_{\text{wsk}}(i, j) = \alpha \mathbf{k}_L(i, j) + \beta \mathbf{k}_G(i, j) \tag{8}$$

$$\mathbf{k}_{\text{wsk}}(i, j) = \alpha \mathbf{u}^T(i) \mathbf{u}(j) + \beta \exp(-\kappa \left\| \mathbf{u}(i) - \mathbf{u}(j) \right\|^2) \tag{9}$$

where $\alpha < 1$, $\beta = (1 - \alpha)$ and κ is the Gaussian kernel parameter controlling its bandwidth. The proposed kernel, $\mathbf{k}_{\text{wsk}}$, can be

directly applied using the kernel methods theory and algorithms previously presented. The main benefits of this kernel are: **First**, the computational burden can be significantly decreased by choosing a different input dimension in each kernel. The idea is then to choose the complete input dimension in the cheap linear kernel (i.e., the dimension modeling the complete acoustic impulse response) and a smaller dimension in the Gaussian kernel to model the nonlinear mapping between the variables. This is possible since the estimation complexity of the nonlinear mapping is linear in the input dimension and independent of the degree of the nonlinearity [8] as opposed to, for instance, truncated Volterra filters [2] [1]. **Second**, it elegantly fits into the leaky KAPA since the parameters α and β give yet another degree of flexibility in the regularization of the solution norm. Taking (3), (7) and (8) the regularization of the solution norm at time l may be written as

$$\lambda \left\|\mathbf{w}\right\|^2 = \lambda \mathbf{w}^T \mathbf{w} = \lambda \sum_{i=1}^{l} \sum_{j=1}^{l} \mathbf{a}_j \mathbf{a}_i \mathbf{f}^T(i)\mathbf{f}(j) \tag{10}$$

$$= \lambda \sum_{i=1}^{l} \sum_{j=1}^{l} \mathbf{a}_j \mathbf{a}_i \mathbf{k}_{\text{wsk}}(i,j) \tag{11}$$

$$= \lambda \sum_{i=1}^{l} \sum_{j=1}^{l} \mathbf{a}_j \mathbf{a}_i \left(\alpha \mathbf{k}_L(i,j) + \beta \mathbf{k}_G(i,j)\right) \tag{12}$$

this shows how the regularization can be favored in one kernel more than the other by varying the weights.

3. PARALLEL-CASCADE SW-KAPA

This section presents two configurations of the SW-KAPA using the proposed kernel $\mathbf{k}_{\text{wsk}}$ for NLAEC namely parallel and cascade SW-KAPA (PSW-KAPA and CSW-KAPA). These algorithms (see Algorithm 1 for details) share some common steps: the computation of the kernel and error signals, the expansion coefficients update and the storage and truncation of the memory buffers (i.e., expansion coefficients vector and input signal matrix). The parallel configuration is actually the direct application of $\mathbf{k}_{\text{wsk}}$ into the leaky KAPA to obtain the PSW-KAPA. The main characteristic of this algorithm is that the input dimension is different in both kernels. While the linear kernel assumes sufficient order, the Gaussian kernel may assume a much smaller dimension. The cascade configuration, on the other hand, consists of two steps: in the first step a standard linear NLMS is performed independently and the output of the filter is stored for the second step; in the second step the stored NLMS output is used as input to the SW-KAPA. The idea behind this configuration is that, as SW-KAPA will work with linearly transformed input data $y(i) = \hat{\mathbf{h}}^T(i)\mathbf{u}(i)$, the ideas of kernel methods can still be used here; in fact if sufficient order is used in the NLMS stage, very little input dimension has to be used in the SW-KAPA stage to model the nonlinear mapping. The performance of PSW-KAPA and CSW-KAPA for NLAEC is demonstrated in the next section.

In algorithm 1 the following variables are adopted: P is the APA projection order, F is the length of the memory buffers,μ is the step-size, λ is the regularization parameter, $\mathbf{u}(i) = [u(i), u(i-1), ..., u(i-L+1)]$ is the input (far-end) signal, $d(i)$ is the desired (microphone) signal, $\hat{\mathbf{h}}$ is the NLMS weight vector of size $(L \times 1)$, $\mathbf{y}_{ker}(i) = [y_{ker}(i-P+1), ..., y_{ker}(i)]$ is a $P \times 1$ output vector, $\mathbf{a}$ is the $F \times 1$ expansion coefficients, $\hat{\mathbf{x}}(i) = [\mathbf{x}(i-P+1), ...\mathbf{x}(i)]$ is a $P \times L$ input vector, $\mathbf{x}(i) = [x(i-L+1), ...x(i)]$ is a $L \times 1$ input vector, the input memory buffer $\mathbf{X}$ is a $L \times F$ matrix, $\mathbf{d}(i) = [d(i-P+1), ..., d(i)]$ is the desired signal vector of size $P \times 1$, $e_{AEC}(i)$ is the NLAEC error (residual) signal.

Algorithm 1: Sliding-Window Parallel-Cascade Kernel APA

while $\{\mathbf{u}(i), d(i)\}$ available **do**
  if $cascade$ **then**
    perform NLMS and assign the filter output
    $\mathbf{x}(i) \leftarrow \mathbf{y}(i) = \hat{\mathbf{h}}^T(i)\mathbf{u}(i)$;
  else
    $\mathbf{x}(i) \leftarrow \mathbf{u}(i)$;
  end if
  $\mathbf{e}_{ker}(i) = \mathbf{d}(i) - \mathbf{y}_{ker}(i) = \mathbf{d}(i) - \mathbf{a}^T \mathbf{k}_{\text{wsk}}(\hat{\mathbf{x}}, \mathbf{X})$;
  $e_{AEC}(i) = e_{ker}(1)$;
  $\mathbf{\Psi} = \begin{bmatrix} \hat{\mathbf{a}} \\ 0 \end{bmatrix} + \mu \mathbf{e}_{ker} \mathbf{G}(\hat{\mathbf{x}})$;
  $\hat{\mathbf{a}} = \mathbf{\Psi}(1 : P-1)$;
  $\mathbf{a} = [(1-\lambda\mu)\mathbf{a}; \quad \Psi(P)]$ % sliding window;
  $\mathbf{X} = [\mathbf{X} \quad \mathbf{x}(i)]$ % input memory buffer;
  if length$(\mathbf{a}) > F$ **then**
    $\mathbf{a}(1) \leftarrow \emptyset$ % Delete first element
    $\mathbf{X}(:,1) \leftarrow \emptyset$% Delete first vector
  end if
end while

4. RESULTS

The performance measure is the *Echo Return Loss Enhancement* (ERLE) which is given as,

$$ERLE(i) = 10 \log_{10} \frac{\sum_{j=1}^{q} d^2[(i-1)q+j]}{\sum_{k=1}^{q} e^2[(i-1)q+j]} \tag{13}$$

which can be seen as the achieved attenuation averaged over time frames of length q. Simulations were performed using speech signals (female speech at sampling frequency 8 kHz), i.i.d. background noise $N(i)$ with SNR 25 dB and the following Hammerstein-like nonlinearity

$$y_{Lin}(i) = \mathbf{h}_0^T \mathbf{u}(i), \quad y_{NL}(i) = \mathbf{h}_0^T \left(\sigma_{NL}[\mathbf{u}^2(i) + \mathbf{u}^3(i) + \mathbf{u}^5(i)]\right)$$
$$d(i) = y_{Lin}(i) + y_{NL}(i) + N(i)$$

where y_{Lin} is the linear echo, y_{NL} is the nonlinear echo, $\mathbf{h}_0$ is an 80 taps measured acoustic impulse response from a mobile phone, $u(i)$ is the far-end signal, $d(i)$ is the microphone signal, σ_{NL} controls the linear to nonlinear echo ratio (LNLR) ratio. The degree of the nonlinearity is chosen so high to demonstrate the validity of the algorithms in modeling high-degree nonlinearities without having to explicitly know the order, in contrast with Volterra filters where the order has to be explicitly set in advance. Even if the memory of the Volterra kernels is chosen small the number of parameters will explode in a fifth order model. The parameters in every simulations are: $\alpha = 0.85$, $\beta = 0.15$, $\mu = 0.5$, $P = 3$, $\kappa = 1$, $F = 1000$, $\lambda = 0.01$, the input dimension of the Gaussian kernel in both the PSW-KAPA and CSW-KAPA is $N_G = 5$, the NLMS filter lengths is $L = 80$, the input dimension of the linear kernel in CSW-KAPA is $N_L = 10$ whereas in PSW-KAPA $N_L = 80$.

Figure 1 shows the result of PSW-KAPA, CSW-KAPA, NLMS-only and Gaussian-only-SW-KAPA (GSW-KAPA) with input dimension $N_G = 80$. The LNLR is set to 24, 12 and 6 dB in Figures 1(a) to 1(c) respectively. It is clear that GSW-KAPA outperforms the rest but at the cost of high computational complexity. GSW-KAPA absolutely outperforms linear NLMS, which performs very poorly, at low LNLR. In between them, both in terms of complexity and performance, PSW-KAPA and CSW-KAPA appear as very attractive alternatives. Their performance is consistently much better than linear NLMS in the lowest LNLR at the cost of some increase of computational complexity. Although, CSW-KAPA performs worse than NLMS in high LNLR, a very interesting (and appealing) characteristic of all the presented SW-KAPA-based algorithms is that

their performance is almost the same regardless of the LNLR. Notice that this characteristic is not usually present in Volterra filters [1]. This fact also proves the efficiency of the regularization in keeping the modeling capabilities almost constant. The involved complexity, in terms of multiplications-additions, only looking at the kernel evaluations are: the Gaussian Kernel is O($[N_G \times F]$), the linear kernel is O($N_L \times F$) and the NLMS is O(L), so this makes linear NLMS = 80, PSW-KAPA = $(80+5) \times 1000 = 85000$, CSW-KAPA $= 80 + (5 + 10) \times 1000 = 15080$. However computing the exponential in the Gaussian kernel evaluation is very expensive if a high input dimension is used. On the other hand, PSW-KAPA and CSW-KAPA have a reasonable complexity while providing a significant improvement with respect to the linear NLMS.

5. CONCLUSIONS

This paper proposes two adaptive algorithms namely PSW-KAPA and CSW-KAPA to solve the problem of NLAEC while keeping the computational complexity low. They are based on leaky KAPA that employs the theory and algorithms of kernel methods. By applying the concept of regularization and deriving a gradient descent method a leaky KAPA is obtained which is the basis to obtain a sliding-window KAPA. A kernel specifically designed for acoustic applications is proposed, which consists in a weighted sum of linear kernel and Gaussian kernels. The motivation is basically to separate the problem in linear and nonlinear subproblems. This strategy reduces the computational complexity as compared with GSW-KAPA and improves performance as compared with linear NLMS. The separated weighting in the proposed kernel also imposes different forgetting mechanisms in the sliding-window approach which in turn translates to a more flexible regularization. Simulation results showed that GSW-KAPA, PSW-KAPA and CSW-KAPA consistently outperform the linear NLMS, and generalize well both in high and low NLNR. However the computational complexity of the GSW-KAPA when using a high input dimension may be prohibitive compared to the much cheaper PSW-KAPA and CSW-KAPA.

6. REFERENCES

[1] L. A. Azpicueta-Ruiz, M. Zeller, J. Arenas-García, and W. Kellermann, "Novel schemes for nonlinear acoustic echo cancellation based on filter combinations," 19–24 April 2009.

[2] F. Küch, *Adaptive Polynomial Filters and their application to Nonlinear Acoustic Echo Cancellation*, Ph.D. thesis, Friedrich–Alexander–Universität Erlanger–Nürnberg, 2005.

[3] W. Liu, J. C. Príncipe, and S. Haykin, *Kernel Adaptive Filtering: A Comprehensive Introduction*, John Wiley, 2010.

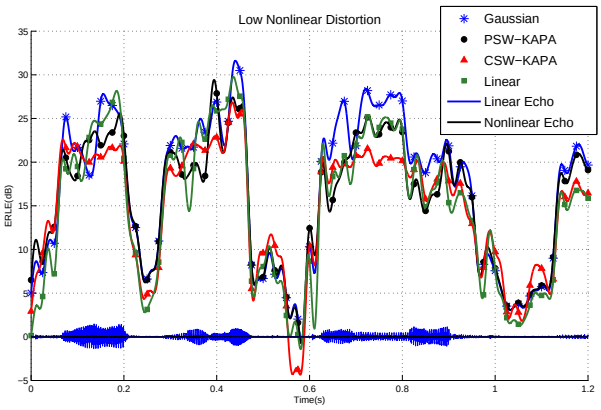
[4] S. Van Vaerenbergh, *Kernel Methods for Nonlinear Identification, Equalization and Separation of Signals*, Ph.D. thesis, Universidad de Cantabria, 2010.

[5] J. Kivinen, A. J. Smola, and R. C. Williamson, "Online learning with kernels," *IEEE Transaction on Signal Processing*, vol. 52, no. 8, pp. 165–176, August 2004.

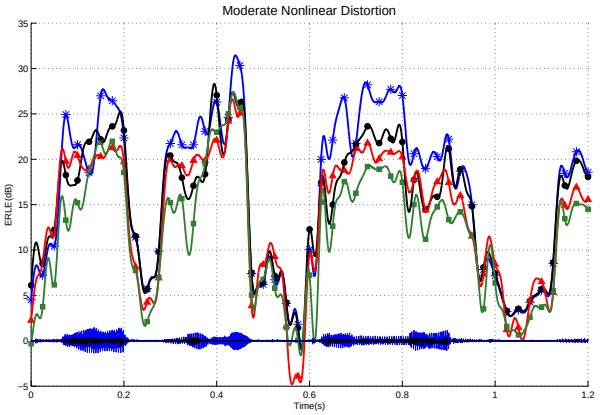
[6] N. Aronszajn, "Theory of reproducing kernels," *Trans. Amer. Math. Soc.*, vol. 68, pp. 337–404, January 1950.

[7] K. Ozeki and T. Umeda, "An adaptive filtering algorithm using an orthogonal projection to an affine subspace and its properties," *Electronics and Communication in Japan*, vol. 67, no. 5, pp. 19–27, August 1984.

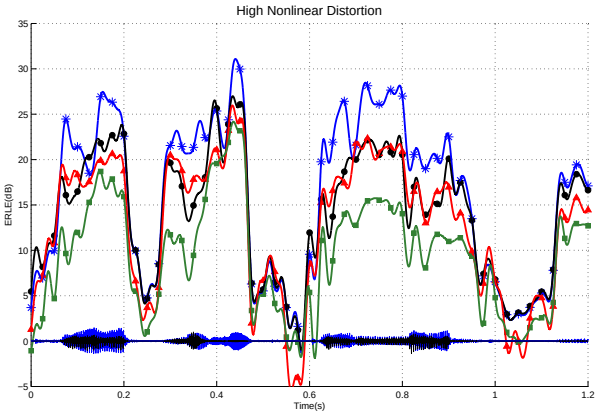
[8] M. O. Franz and B. Schölkopf, "A unifying view of Wiener and Volterra theory and polynomial kernel regression," *Neural Computation*, vol. 18, no. 12, pp. 3097–3118, 2006.



(a) 24 dB LNLR



(b) 12 dB LNLR



(c) 6 dB LNLR

Fig. 1. ERLE at different LNLR comparing the four methods: Gaussian kernel only (GSW-KAPA), Linear NLMS only, Parallel and Cascade configuration using the weighted sum of kernels approach (PSW-KAPA and CSW-KAPA). The stars are points of GSW-KAPA, squares are points of NLMS, triangles are points of CSW-KAPA and circles are points of PSW-KAPA.