MULTI-KERNEL EQUALIZATION FOR NON-LINEAR CHANNELS

Minh Nguyen-Viet
Posts and Telecommunications Institute of Technology, Hanoi City, Vietnam

Abstract: Nonlinear channel equalization using kernel equalizers is a method that has attracted lots of attention today due to its ability to solve nonlinear equalization problems effectively. Kernel equalizers based on Recursive Least Squared, K-RLS, are successful methods with high convergent rate and overcome the local optimization problem of RBF neural equalizers. In recent years, some simple K-LMS algorithms are used in nonlinear equalizers to further enhance the flexibility with the adaptive capability of equalizers and reduce the computational complexity. This paper proposes a new approach to combine the convex of two single-kernel adaptive equalizers with different convergent rates and different efficiencies in order to get the best kernel equalizer. This is the Gaussian multi-kernel equalizer.

Keywords: Adaptive equalization, kernel equalizer, multi-kernel filter, nonlinear channel.

I. INTRODUCTION

Single-kernel adaptive filters are used widely today to identify and track the non-linear systems [1,2,3]. The developments of kernel adaptive filters enable us to solve non-linear estimation problems using linear structures. In this paper, we use kernel adaptive filters for equalizations of non-linear wireless channels such as satellite channels.

Wireless channels with their time-variant parameters cause non-linear and linear distortions to the transmitted signal. Channel equalizers are used to minimize these distortions. Commonly channel equalizers can be considered as reverse filters which have characteristics must repeat the structure and the conversion rule of the channel. To execute that task, the equalizer in the receiver must has ability to perform the channel estimation using MLSE algorithms [3] with the complexity increases following the exponential function of the impulse response dimension. So far the most popular used equalizers are equalizers using neural networks such as MLP (Multi-Layer Perceptron), FBNN (Feed Back Neural Network), RBF (Radial Basis Function), RNN (Recursive Neural Network), SOM (Self Organization Mapping), the wavelet neural networks [3].

The mentioned equalizers have different complexities but they have a common advantage that is the capability of well solving the non-linear equalization problems. However, there are still some issues that should be noticed [3]:

+ The neural networks are only able to find the local optimization, cannot solve the overall optimization problem due to the partial derivative characteristic.
+ If the system transmits the M-QAM signals, the linear and non-linear distortions at the receiver will be a non-stop process. Therefore the equalizer must has two parts which are the time-variant linear part and the non-linear part results in a complex system.
+ The low convergent rate due to the complexity if the network structure and the training phase takes time.

Correspondence: Minh Nguyen-Viet,
email: minhnv@ptit.edu.vn
Communication: received: Mar. 3, 2016,
revised: May 6 2016, accepted: May 30, 2016.

To solve the above problems, recently the single kernel adaptive filters based on common algorithms such as K-RLS (Kernel Recursive Least Squared) [1],the sliding-window K-RLS [4,6], the extended K-RLS [5], the standard kernel LMS [7,8] are proposed. In recent years, there are some simple K-LMS algorithms [9,10,11,12].

To further enhance the flexibility with the adaptive capability of equalizers and reduce the computational complexity, in this paper we propose the multi-kernel equalizer based on some researches about the multi-kernel [13,14,16,17,18,19]. The solution here is to combine the convex of two single-kernel adaptive filters with different convergent rates and different efficiencies in order to get the best equalizer. In our proposal, two simple K-LMS equalizers are used.

The following content will be organized as follows: Section 2 is about multi-kernel LMS adaptive algorithm; Section 3 is about multi-kernel equalization; simulation results will be shown in Section 4 and Section 5 is conclusion.

II. Multi-kernel LMS Adaptive Algorithm

As mentioned above, two K-LMS filters are combined to build a novel equalizer, so first of all we present multi-kernel LMS adaptive algorithm. This content is refered to [3].

Consider a time-variant mapping:

$$\Psi^t : X \to H_L$$
$$x \to \psi^t(x) = \begin{bmatrix} c_1^t \psi_1(x) \\ c_2^t \psi_2(x) \\ \vdots \\ c_L^t \psi_L(x) \end{bmatrix} \quad (1)$$

Here we have t is the time index and $\{c_\ell^t\}_{\ell=1,2,\ldots}$ is a time-variant parameters row, approximate the output d_t:

$$y_t = \langle \mathrm{W}, \psi^t(x_t) \rangle \quad (2)$$

The parameter $\{c_\ell^t\}_{\ell=1,2,\ldots}$ servers the instant distribution of each kernel in multi-kernel algorithm at t, therefore how they are updated decides the adaptive characteristic of the algorithm. The parameter matrix W (with L elements) separates information from specific patterns to repeat the non-linear characteristic of the signal.

Use statistical gradient to update W:

$$\mathrm{W}_t = \mathrm{W}_{t-1} + \mu e_t \psi^t(x_t) = \mu \sum_{j=1}^{t} e_j \psi^j(x_j) \quad (3)$$

Here μ is learning rate. We can estimate the output:

$$\begin{aligned} y_t &= \left\langle \mu \sum_{j=1}^{t-1} e_j \psi^j(x_j), \psi^t(x_t) \right\rangle \\ &= \mu \sum_{j=1}^{t-1} e_j \langle \psi^j(x_j), \psi^t(x_t) \rangle \end{aligned} \quad (4)$$

Use scalar multiplication feature for K-RLS vector values, the value <*> of the right side of (4) is:

$$\begin{aligned} \langle \psi^j(x_j), \psi^t(x_t) \rangle &= \sum_{\ell=1}^{L} \langle c_\ell^j \psi_\ell(x_j), c_\ell^t \psi_\ell(x_t) \rangle_{\mathrm{H}_\ell} \\ &= \sum_{\ell=1}^{L} c_\ell^j c_\ell^t k_\ell(x_j, x_t) \end{aligned} \quad (5)$$

Put (5) into (4) we have output estimation:

$$\hat{d}_i = \mu \sum_{j=1}^{i-1} e_j \sum_{\ell=1}^{L} c_\ell^i c_\ell^j k_\ell(x_i, x_j) \quad (6)$$

To simplify (6), let $\omega_{i,j,l} = e_j c_\ell^i c_\ell^j$ we have:

$$y_t = \mu \sum_{j=1}^{i-1} \sum_{\ell=1}^{L} \omega_{t,j,\ell} k_\ell(x_t, x_j) \quad (7)$$

The effect of using a multi-kernel combination in the MK-LMS algorithm is the adaptive design Ψ^t is performed by updating $\omega_{t,j,\ell}$ therefore the parameters $\{c_\ell^t\}_{\ell=1\div L}$ don't have to be updated directly. The result of combining LMS update for W in (2) indicates that the relationship of estimation $\hat{d}_t$ is a linear combination of multi-kernel. Therefore (7) can be considered as a common multi-kernel rule and will be used in multi-kenel equalizers.

III. Multi-kernel Equalization

Base on multi-kernel LMS adaptive angorithm discribed in section 2, here we build a novel multi-kernel adaptive equalizer for nonlinear channel. In this paper, we limit the research in case the equalizer has two single-kernel. The block diagram of the equalizer is shown in Figure 1.

From (7), the output estimation in two kernel case is:

$$y_t = \mu_1 \sum_{\ell=1}^{L_1} \omega_{1,j,\ell} k_1(x_1, x_\ell) + \mu_2 \sum_{\ell=1}^{L_2} \omega_{2,j,\ell} k_2(x_2, x_\ell) \quad (8)$$

Here μ is the learning rate of the algorithm.

$k_1(.,.);k_2(.,.)$ is the kernel functions of equalizers 1 and 2.

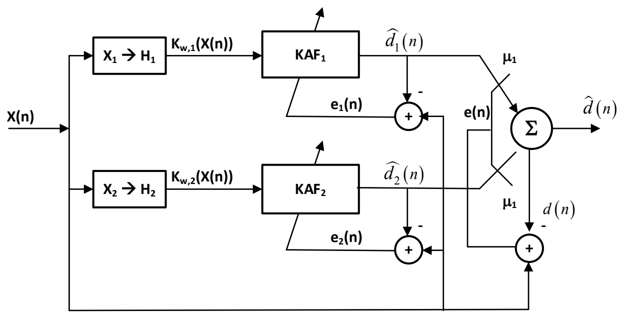

Figure 1. Multi-kernel equalization

In two kernel equalizer, $\omega_{t,i,\ell}$ is calculated due to the standard LMS [2]:

$$\omega_{t,j,\ell} = \omega_{t-1,j,\ell} + \mu e_t \frac{k_\ell(x_t, x_j)}{\varepsilon + k_\ell^2(x_t, x_j)} \quad (9)$$

$e_t = d_t - y_t \quad \in \quad \Re^m$ is the error estimation.

The multi-kernel algorithm:

Multi-Kernel Least Mean Square algorithm – MK-LMS

Initialization:

Dictionary: $D = \{x_0\}$

Kernel set: $K = \{k_1, k_2, \ldots, k_L\}$

Initial weight: $\omega_{k,x_1} = \hat{\mu} d_1$ (for each kernel)

(To simplify, let $\omega_{k,x}$ is the corresponding weigh with kernel k and support vector x)

for *training pair* (x_t, d_t) **do**:

Pattern variance: $e_D \leftarrow \min_{x_j \in D} \|x_t - x_j\|$

Predict: $y_t \leftarrow \mu \sum_{x_j \in D} \sum_{k \in K} \omega_{k,x_j} k(x_t, x_j)$

Error: $e_t \leftarrow d_t - y_t$

New characteristic

if $\|e_t\| \ge \delta_e \wedge e_D \ge \delta_d$ **then**

Add new pattern: $D \leftarrow D \cup (x_t)$

for all $k \in K$ do

Starting new weigh: $\omega_{k,x_t} \leftarrow \hat{\mu} d_t$

end for

else

for all $k \in K, x_j \in D$ do

Update: $\omega_{k,x_j} \leftarrow \omega_{k,x_j} + \hat{\mu} e_t \dfrac{k(x_t, x_j)}{\varepsilon + k(x_t, x_j)^2}$

end for

end if

Perform and discard

for all $x_j \in D$ **do**

Instant perform: $p_t(x_j) \leftarrow K_G(x_j, x_t)$

Perform: $P_t(x_j) \leftarrow (1-\rho) P_{t-1}(x_j) + \rho p_t(x_j)$

end for

if Doing discard **then**

Discard pattern: $D \leftarrow \{x_j \in D : P_t(x) \ge \delta_p\}$

end if

end for

IV. SIMULATION RESULTS

In this section, we consider the combination between two K-LMS algorithms and the Gaussian kernel with different bandwiths. The equalizer uses the MK-LMS algorithms discribed in section 3, here called ComKAF. A non-linear system used in the simulation is described as follow:

$$\begin{aligned} d(n) &= \left[0{,}8 - 0{,}5\exp\left(-d(n-1)^2\right)\right] d(n-1) \\ &- \left[0{,}3 + 0{,}9\exp\left(-d(n-1)^2\right)\right] d(n-2) + 0{,}1\sin\left(d(n-1)\pi\right) \end{aligned} \quad (10)$$

Here $d(n)$: system output,

$u(n)=\left[d(n-1),d(n-2)\right]^T$: system input. The initial condition is $d(0)=d(1)=0{,}1$. The output $d(n)$ is affected by AWGN $z(n)$ with standart deviation $\sigma=0{,}1$.

The comparison is performed between ComKAF which is a combination of two K-LMS algorithm models, two independent K-LMS algorithms, the MK-LMS algorithm in [15,18] and the MxKLMS in [20]. A consistent property is used to build the equalization dictionary. A consistent threshold is set to achieve the same length for all equalizers. Parameters set for each algorithm is shown in Table I. The parameter μ and a_0 is set to 80 and 4 respectively. The learning rate to update port function of the MxKLMS algorithm is 0.1. The experimental results are averaged for 200 Monte Carlo runs.

Table I. The parameters set for the equalizers

Algorithm	Kernel band-width ξ	Step size η	Correlation Threshold μ
KLMS1	0,25	0,05	0,5
KLMS2	1	0,05	0,9576
MKLMS	[0,25;1]	0,03	[0,5;0,9576]
MxKLMS	[0,25;1]	0,15	[0,5;0,9576]
ComKAF	[0,25;1]	[0,05;0,05]	[0,5;0,9576]

Figure 2(a) shows that the proposed algorithm has better performance than two independent KLMS: It has the high convergent rate as the fastest KLMS algorithm and it achieves lowest stable state EMSE. This is due to the adaptive port function enables switching between two independent single kernel algorithms, as illustrated in Figure 1. Figure 2(b) shows that if equally compare, the consistent thresholds are set in order to achieve the same dictionary length for all algorithms. According the compare method, Figure 2(a) shows that three multi-kernel methods achieve nearly similar performance.

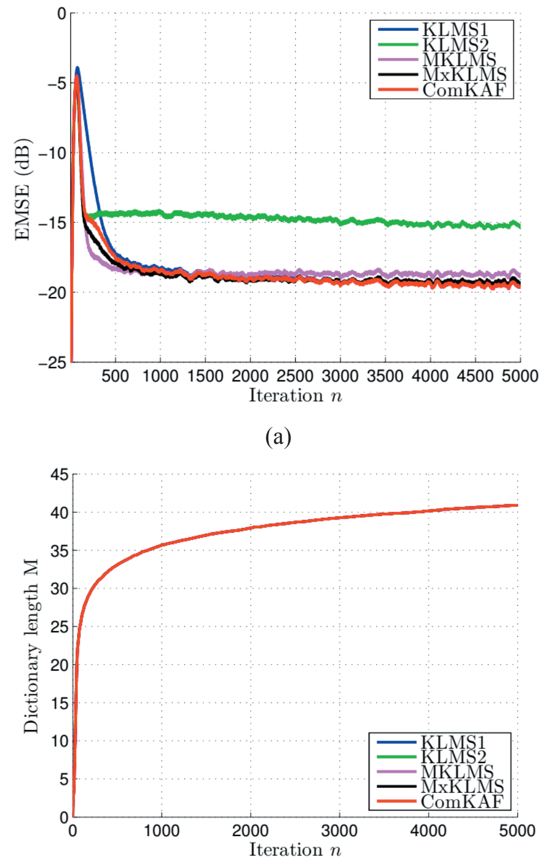

Figure 2. The result of performance analysis.
(a) The average learning curve EMSE; (b) Development of average combined dictionary length;

Comparing MxKLMS and ComKAF for function weight, figure 3 shows that the port function of MxKLMS does not converge to the same value as proposed.

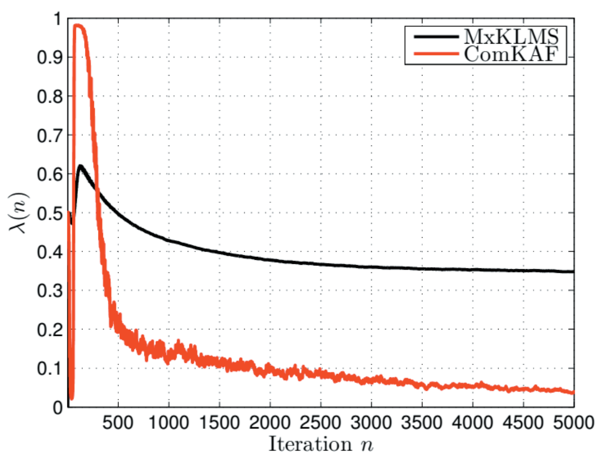

Figure 3. The average curves for functional weight

V. CONCLUSION

In this paper, we propose a flexible approach that combines two single adaptive kernel equalizers using the K-LMS algorithm. The simulation results show the ability of the equalizer in achieving the best equal performance compared to each independent single equalizer. Obviously using multi-kernel in building adaptive equalizers for non-linear channels has many advantages. Futher work will be about analyzing the convergence characteristic and consider the combination of more than two algorithms, possibly with K-RLS.

References

[1]. Y. Engel, S. Mannor, and R. Meir, "Kernel recursive least squares," IEEE Transactions on Signal Processing, vol. 52, no. 8, pp. 2275–2285, 2004.

[2. W. Liu, P. P. Pokharel, and J. C. Pr´ıncipe, "The kernel least mean-square algorithm," IEEE Transactions on Signal Processing, vol. 56, no. 2, pp. 543–554, 2008.

[3]. W. Liu, J. C. Pr´ıncipe, and S. Haykin, Kernel Adaptive Filtering: A Comprehensive Introduction, Jonh Wiley & Sons, New-York, 2010.

[4]. S. Van Vaerenbergh, J. V´ıa, and I. Santamar´ıa, "A sliding window kernel RLS algorithm and its application to nonlinear channel identification," in Proc. IEEE ICASSP, Toulouse, France, May 2006, pp. 789–792.

[5]. W. Liu, I. M. Park, Y. Wang, and J. C. Prıncipe, "Extended kernel recursive least squares algorithm," IEEE Transactions on Signal Processing, vol. 57, no. 10, pp. 3801–3814, 2009.

[6]. S. Slavakis and S. Theodoridis, "Sliding window generalized kernel affine projection algorithm using projection mappings," EURASIP Journal on Advances in Signal Processing, vol. 2008:735351, Apr. 2008.

[7]. B. Chen, S. Zhao, P. Zhu, and J. C. Pr´ıncipe, "Quantized kernel least mean square algorithm," IEEE Transactions on Neural Networks and Learning Systems, vol. 23, no. 1, pp. 22–32, 2012.

[8]. W. D. Parreira, J.-C. M. Bermudez, C. Richard, and J.-Y. Tourneret, "Stochastic behavior analysis of the Gaussian kernel-least-mean-square algorithm," IEEE Transactions on Signal Processing, vol. 60, no. 5, pp. 2208–2222, 2012.

[9]. C.Richard and J.-C.M.Bermudez, "Closed-form conditions for convergence of the Gaussian kernel-least-mean-square algorithm," in Proc. Asilomar, Pacific Grove, CA, USA, 2012.

[10]. W. Gao, J. Chen, C. Richard, J. Huang, and R. Flamary, "Kernel LMS algorithm with forward-backward splitting for dictionary learning," in Proc. IEEE ICASSP, Vancouver, Canada, 2013, pp. 5735–5739.

[11]. W. Gao, J. Chen, C. Richard, and J. Huang, "Online dictionary learning for kernel LMS," IEEE Transactions on Signal Processing, vol. 62, no. 11, pp. 2765–2777, 2014.

[12]. J. Chen, W. Gao, C. Richard, and J.-C. M. Bermudez, "Convergence analysis of kernel LMS algorithm with pre-tuned dictionary," in Proc. IEEE ICASSP, Florence, Italia, 2014.

[13]. M. Yukawa, "Nonlinear adaptive filtering techniques with multiple kernels," in Proc. EUSIPCO, Barcelona, Spain, 2011, pp. 136–140.

[14]. M. Yukawa, "Multikernel adaptive filtering," IEEE Transactions on Signal Processing, vol. 60, no. 9, pp. 4672–4682, 2012.

[15]. M. Yukawa and R. Ishii, "Online model selection and learning by multikernel adaptive filtering," in Proc. EUSIPCO, Marrakech, Morocco, Sept. 2013, pp. 1–5.

[16]. F. A. Tobar and D. P. Mandic, "Multikernel least squares estimation," in Proceedings of Sensor Signal Processing for Defense, London, UK, 2012.

[17]. F.A. Tobar, S.-Y Kung, and D.P. Mandic, "Multikernel least mean square algorithm," IEEE Transactions on Neural Networks and Learning Systems, vol.25, no.2, pp.265–277, 2014.

[18]. T. Ishida and T. Tanaka, "Multikernel adaptive filters with multiple dictionaries and regularization," in Proc. APSIPA, Kaohsiung, Taiwan, Oct.-Nov. 2013.

[19]. R. Pokharel, S. Seth, and J. Pr´ıncipe, "Mixture kernel least mean square," in Proc. IEEE IJCNN, 2013.

[20]. J. Arenas-Garc´ıa, A. R. Figueiras-Vidal, and A. H. Sayed, "Mean-square performance of a convex combination of two adaptive filters," IEEE Transactions on Signal Processing, vol. 54, no. 3, pp. 1078–1090, 2006

Minh Nguyen-Viet received the BS degree and MS degree of electronics engineering from Posts and Telecommunications Institute of Technology, PTIT, in 1999 and 2010 respectively. His research interests include mobile and satellite communication systems, transmission over nonlinear channels. Now he is PhD student of telecommunications engineering, PTIT, Vietnam.