Kernel Recursive Least Squares-Type Neuron for Nonlinear Equalization

Mohammed Naseri Tehrani*, Majid Shakhsi**, Hossein khoshbin***

*Ferdowsi university, Mn_tehrani1120@yahoo.com
** Ferdowsi university, Majidshakhsi@gmail.com
*** Ferdowsi university, khoshbin@um.ac.ir

Abstract —*The nonlinear channel distotions and the nonminum phase channel characteristics modelling, are a significant part in channel equalization problems . on the other hand, the nonlinear system requiring equalization is often noninvertible, resulting in a drastic loss of information. So far, Hammerstein and wiener models, Artificial Neural Networks (ANN), radial basis function (RBF) have been widely used as nonlinear methods in different applications, such as equalization. The kernel methods are well known for their great modelling capacity of nonlinear systems in addition to their modest complexity. A new kernel recursive least square-type neuron (NKRLS) equalizer is proposed which improves aforementioned nonlinear methods problems such as, classical training algorithm drawbacks to parameter definition, slow convergence, local minima, non-convex optimization, loss of universal approximation . NKRLS does that thanks to its nonparametric and universal approximation properties. NKRLS cosnsists of Kenel recursive least square followed by a simple neuron. In the first part of paper the new proposed KRLS-type neuron algorithm is introduced. The second part of paper corroborates our results with simulation results.*

Index Terms — Reproducing Kernel Hilbert spaces , Kernel recursive least squares, Neural network, Equlization.

I. Introduction

Kernel methods have been recently considered as indispensible component of nonlinear supervised and unsupervised learning algorithm versus their linear counterparts [1]. They are applied in wide range of areas from pattern analysis [2] , [3], identification or equalization in communication systems [4], [5] to time series analysis and probability density estimation [6]_[8]. Thanks to kernel function, Kernel methods exploit an efficient nonlinear mapping from finite dimensional *data space* to a very high dimentional Hilbert space called feature space [1]-[3]. Thus the all benefit of working in higher dimension is obtained. Kernel recursive least square is considered as the most successful method [9], enabling actual and desired output pairs well matching and fast rate of convergence in kernel adaptive filtering. KRLS has been Recently a very hot research area of regression tasks that results in works such as sliding window kernel recursive least square (SW_KRLS) [10], extended kernel recursive least square (EX_KRLS) [11]. Both exponentialy weighted kernel recursive least square (EW_KRLS) and random walk kernel recursive least square are special cases of EX_KRLS.

During the data transmition over communication channel, the data are affected by linear and nonlinear distortions. Linear ditortions include inter-symbol interference (ISI), co-channel interference (CCI) in the presence of white Gaussian noise (AWGN). Nonlinear distortions are caused due to the subsystems such as amplifiers, modulator, demodulator, etc. In order to suppress all these channel distortions for the best recovery of symbols, different adaptive equalization techniques were proposed. Nonlinear equalizers are superior to linear ones in applications where the channel distortion is too severe for a linear equalizer to handle. In particular a linear equalizer does not perform well on channels with deep spectral nulls in their amplitude characteristics or with nonlinear distortion. In an attempt to compensate for the distortion, the linear equalizer places too much gain in the vicinity of the spectral nulls, thereby enhancing the noise present in these frequencies. Linear equalizers view equalization as inverse problem while nonlinear equalizers view equalization as a pattern classification problem where equalizer classifies the input signal vector into discrete classes based on transmitted data. Given minimum and non-minimum phase channels, problem starts when equalizing non-minimum phase channels [12]. For this channel, a simple linear decision boundary cannot classify the symbols easily. It needs a nonlinear decision boundary or even a hyper-plane in high-dimensional channel space. Such a decision boundary cannot be achieved using a linear filter. Conventional nonlinear methods in nonlinear equalization include Hammerstein and wiener models, Artificial Neural Networks (ANN), radial basis function (RBF) networks, Volterra models, Recurrent Networks. For instance, the first method suffers from loss of universal approximation , second and third methods suffers from non-convex optimization and local minima [13]-[14].

978-1-4673-5634-3/13/$31.00 ©2013 IEEE

TABLE I [1]
NEURAL NETWORKS VERSUS KERNEL FILTERS

Properties	ANNs	Kernel filters
Universal Approximators	YES	YES
Convex Optimization	NO	YES
Model Topology grows with data	NO	YES
Require Explicit Regularization	NO	YES/NO(KLMS)
Online Learning	YES	YES
Computational Complexity	LOW	MEDIUM

In order to show advantages and disadvantages of kernel filters versus the Neural Networks, a comprehensive comparison is done in TABLE I that show the motivation of working with kernels. Given the aforementioned nonlinear methods drawbacks , for instance as in TABLE I, Multilayer Neural networks as parametric model with the major problems of non-convex optimization and local minima has given rise to the proposed nonparametric kernel model of KRLS-type neuron.

The KRLS-type neuron (NKRLS) exhibits properties such as *kernel abilities*, *RLS features*, and *traditional neuron capabalities, universal approximation* property, *convex optimization* property, and *facilitating kernel trick* property with regularzation. NKRLS is a neuron with structure proposed by [15] which is trained in KRLS, in other words NKRLS is a nonparametric classifier that can discriminate every nonlinear pattern without predefined parameters. The proposed method faces problems such as the computational and data storage complexity increment with each training . In order to cope with these problems, sparsification is exploited. But in this paper, it is focused on the non-sparse version of all the algorithms.

The organization of the paper is as follwos. Kernel method is reviwed in section II , the proposed NKRLS algorithm is introduced in section III . Then simulation results show the proposed algorithm superiority in section IV. finally the conclusions are presented in section V.

II. *A KERNEL METHOD*

A kernel [16] is described as continuous, symmetric, positive-definite function $k(.,.): \mathrm{R}^m \times \mathrm{R}^m \rightarrow \mathrm{R}$ with R^m as input domain. There are different types of kernels such as Gaussian (1), polynomial(2):

$$k\ (\mathrm{u},\mathrm{u}') = \exp\left(-a \left\|\mathrm{u}-\mathrm{u}'\right\|^2\right) \quad (1)$$

$$k\ (\mathrm{u},\mathrm{u}') = \left(\mathrm{u}^T \mathrm{u}'+1\right)^P \quad (2)$$

According to mercer theorem [16]-[17] kernel expansion is proportional to eigenvalues ξ_i and eigenfunctions ϕ_i as follows:

$$k\ (\mathrm{u},\mathrm{u}') = \sum_{i=1}^{\infty} \xi_i \phi_i\ (\mathrm{u}) \phi_i\ (\mathrm{u}') \quad (3)$$

1. Jose C.principe, Online kernel learning, Computational NeuroEngineering Laboratory (CNEL)University of Florida (www.cnel.ufl.edu).

A mapping φ is defined as

$$\varphi : R^m \rightarrow F$$

$$\varphi(u) = \left[\sqrt{\xi_1}\phi_1(u), \sqrt{\xi_2}\phi_2(u), ...\right] \quad (4)$$

Dimension of F is proportional to number of strictly positive eigenvalues, and can be finite or infinite. The main idea of kernel method is equivalence of feature mapping inner products in feature space to kernel evaluation, which escapes from challenging task of feature mapping calculation in high dimentional feature space. This equivalence is defined as kernel trick.

$$\varphi^T(u)\varphi(u') = k(u,u') \quad (5)$$

After this data mapping to high dimentional space, linear methods are employed in taking advantage of it.

III. *KERNEL RECURSIVE LEAST SQUARES-TYPE NEURON*

Multilayer Neural network as parameteric model, suffers from major losses, such as non-convex optimization , local minima, and parameter initialization sensitivity . The neural network is susceptible to overtraining and that the number of layers, the number of neurons per layer, and when to stop adapting must be determined in an ad hoc fashion **.** In the case of neural networks, the feature space is generated via multiple layers of nonlinear functions (i.e., neurons) acting on either filter states (first layer) or outputs of the previous layer. But the proposed structure represents the fature space via kernel Hilbert space and just one neuron**.** In order to suppress these problems, nonparametric kernel recursive least squares–type neuron algorithm is proposed. There is no need to initialize parameter as in neural networks, therefore the KRLS-type neuron adapts itself with data structure. The NKRLS takes advantages of kernel abilities, RLS features, and traditional neuron capabalities. The NKRLS is a neuron which is trained in KRLS, in other words NKRLS is a nonparametric classifier that can discriminate every nonlinear pattern without predefined parameters . These advantages comes at the expense of a higher computational complexity. To derive kernel recursive least squares-type neuron, the structure of kernel recursive least squares algorithm and RLS training algorithm for neural networks are exploited (Fig. 1.). the notations are established in this section.

III.A *Notations*

$\varphi_{n\times 1}(\mathbf{i})$ is input vector .

$\mathbf{W}_{n\times 1}(\mathbf{i}) = [w_1(i), ..., w_n(i)]^T$ is weight vector.

And define

$$\Phi(i) = [\varphi(1), ..., \varphi(i)] \quad (6)$$

$$\mathbf{b}(i) = [b(1), ..., b(i)], \quad (7)$$

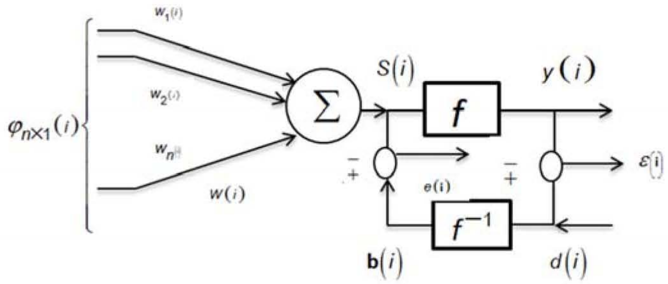

Fig. 1. kernel recursive least squares-type neuron structure

$$\mathbf{F}(i) = \mathbf{diag}\left[f'^2(s(1)),...,f'^2(s(i))\right] \quad (8)$$

$$f(b(j)) = d(j) \Leftrightarrow f^{-1}(d(j)) = b(j) \quad (9)$$

$$s(j) = \varphi^T(j)w(i) = \varphi(j)w^T(i) \quad (10)$$

Rewrite (10)

$$s(i) = h^T(i)\mathbf{a}(i-1)$$
$$= \left[k(\mathbf{u}(1),\mathbf{u}(i)),...,k(\mathbf{u}(i-1),\mathbf{u}(i))\right] \times \left[a_1(i-1),...,a_j(i-1)\right]^{\mathbf{T}}$$

$$s(i) = \sum_{j=1}^{i-1} a_j(i-1)k(\mathbf{u}(j),\mathbf{u}(i)) \quad (11)$$

And the final output is $y(j) = f(s(j))$ (12)

III.B *problem formulation*

the regularized cost function for NKRLS is expressed as follows :

$$Q(i) = \sum_{j=1}^{i} \gamma^{i-j} \left| d(j) - f\left(w^T(i)\varphi(j)\right)\right|^2 + \gamma^i \lambda \|w\|^2 \quad (13)$$

Where it is supposed that γ =1 (forgetting factor). As in

KRLS the cost function is derived partially with respect to

weight vector $\frac{\partial Q(i)}{\partial w(i)} = 0$

$$\frac{\partial Q(i)}{\partial w(i)} = 2\sum_{j=1}^{i} \frac{\partial f(s(j))}{\partial s(j)} \varphi(j)\left(f(b(j)) - f(s(j))\right) + 2\lambda w(i) \quad (14)$$

Where the output error is

$$\varepsilon(j) = d(j) - y(j) = f(b(j)) - f(s(j)) \quad (15)$$

And the activation function derivative with respect to $s(j)$

$$f'(s(j)) = \frac{\partial f(s(j))}{\partial s(j)} \quad (16)$$

Suppose the approximation [15]

$$f(b(j)) \approx f(s(j)) + f'(s(j))(b(j) - s(j)) \quad (17)$$

Rewrite eq (14) it turns out

$$r(i) = R(i)w(i) \quad (18)$$

Where $r(i)$ and $R(i)$ is defined

$$r(i) = \sum_{j=1}^{i} f'^2(s(j))b(j)\varphi(j) \quad (19)$$

$$R(i) = \sum_{j=1}^{i} f'^2(s(j))\varphi(j)\varphi^T(j) + \lambda \mathrm{I} \quad (20)$$

Then rewrite eq (19),(20)

$$r(i) = \Phi(i)\mathbf{F}(i)\mathbf{b}(i) \quad (21)$$

$$R(i) = \left[\Phi(i)\mathbf{F}(i)\Phi^{\mathbf{T}}(i) + \lambda \mathrm{I}\right] \quad (22)$$

Subsequently, the weight is expressed as

$$w(i) = R(i)^{-1} r(i)$$
$$= \left[\Phi(i)\mathbf{F}(i)\Phi^{\mathbf{T}}(i) + \lambda \mathrm{I}\right]^{-1} \Phi(i)\mathbf{F}(i)\mathbf{b}(i) \quad (23)$$

from inversion lemma [18]

$$A = \left[\Phi(i)\mathbf{F}(i)\Phi^{\mathbf{T}}(i) + \lambda \mathrm{I}\right]^{-1} \Phi(i)\mathbf{F}(i) \quad (24)$$

$$A = \Phi(i)\left[\lambda \mathbf{F}^{-1}(i) + \Phi^{\mathbf{T}}(i)\Phi(i)\right]^{-1} \quad (25)$$

Rewrite eq (23)

$$w(i) = \Phi(i)\left[\lambda \mathbf{F}^{-1}(i) + \Phi^{\mathbf{T}}(i)\Phi(i)\right]^{-1} \mathbf{b}(i)$$
$$= \Phi(i)a(i) \quad (26)$$

the kernel coefficient is expressed as follows

$$a(i) = \left[\lambda \mathbf{F}^{-1}(i) + \Phi^{\mathbf{T}}(i)\Phi(i)\right]^{-1} \mathbf{b}(i) \quad (27)$$

The inversed part of eq(27) is expressed in a recursive form,i.e.

$$Q(i) = \left[\lambda \mathbf{F}^{-1}(i) + \Phi^{\mathbf{T}}(i)\Phi(i)\right]^{-1} \quad (28)$$

$$Q^{-1}(i) = \begin{pmatrix} Q^{-1}(i-1) & h(i) \\ h^{T}(i) & \lambda\left(f'^{2}(s(i))\right)^{-1} + \varphi^{\mathbf{T}}(i)\varphi(i) \end{pmatrix} \quad (29)$$

$$\theta(i) = \left(f'^{2}(s(i))\right)^{-1} \quad (30)$$

From matrix inversion lemma [18]

$$Q(i) = r^{-1}(i)\begin{pmatrix} Q(i-1)r(i) + z(i)z^{T}(i) & -z(i) \\ -z^{T}(i) & 1 \end{pmatrix} \quad (31)$$

$$\begin{cases} r(i) = \lambda\theta(i) + \varphi^{\mathbf{T}}(i)\varphi(i) - z^{\mathbf{T}}(i)h(i) \\ z(i) = Q(i-1)h(i) \end{cases} \quad (32)$$

Rewrite eq (27)

$$\mathbf{a}(i) = Q(i)\mathbf{b}(i)$$

$$= \begin{pmatrix} Q(i-1) + z(i)z^{\mathbf{T}}(i)r^{-1}(i) & -z(i)r^{-1}(i) \\ -z^{\mathbf{T}}(i)r^{-1}(i) & r^{-1}(i) \end{pmatrix}\begin{pmatrix} \mathbf{b}(i-1) \\ b(i) \end{pmatrix}$$

$$\mathbf{a}(i) = \begin{pmatrix} \mathbf{a}(i-1) - z(i)r^{-1}(i)e(i) \\ r^{-1}(i)e(i) \end{pmatrix} \quad (33)$$

$$e(i) = b(i) - z^{\mathbf{T}}(i)\mathbf{b}(i-1) \quad (34)$$

KRLS-type neuron algorithm is summarized in algorithm 2.

Algorithm 2: KRLS-type neuron (NKRLS) ALGORITHM

Initialization

$f(.)$ define the activation function

$$f(b(j)) = d(j) \Leftrightarrow f^{-1}(d(j)) = b(j)$$

$$Q(1) = \left(\lambda F^{-1}(1) + k(\mathbf{u}(1), \mathbf{u}(1))\right), \ a(1) = Q(1)\mathbf{b}(1)$$

Computation

While $\{u(i), d(i)\}$ (i>1) available **do**

$$h(i) = \left[k(\mathbf{u}(1), \mathbf{u}(i)), \ldots, k(\mathbf{u}(i-1), \mathbf{u}(i))\right]^{T}$$

$$z(i) = Q(i-1)h(i)$$

$$r(i) = \lambda\theta(i) + \varphi^{\mathbf{T}}(i)\varphi(i) - z^{\mathbf{T}}(i)h(i)$$

$$Q(i) = r^{-1}(i)\begin{pmatrix} Q(i-1)r(i) + z(i)z^{T}(i) & -z(i) \\ -z^{T}(i) & 1 \end{pmatrix}$$

$$e(i) = b(i) - z^{\mathbf{T}}(i)\mathbf{b}(i-1)$$

$$\mathbf{a}(i) = \begin{pmatrix} \mathbf{a}(i-1) - z(i)r^{-1}(i)e(i) \\ r^{-1}(i)e(i) \end{pmatrix}$$

End while

V . NUMERICAL EXAMPLES

Equiprobable sequence of BPSK signal $(s_n)_n$ is transmitted through linear time invariant (LTI) [20] channel which results in the signal $(w_n)_n$. Given the strategy of $\sum_{i=0}^{i=2} h_{l\,i}^{2} = 1$ [5,(28)] in the transfer function $H_l(z) := \sum_{i=0}^{i=2} h_{l\,i} z^{-i}$, $z \in C, l = 1, 2$, Two LTI channel $H_l(z) := \sin(\theta_l)/\sqrt{2} + \cos(\theta_l)z^{-1} + \left(\sin(\theta_l)/\sqrt{2}\right)z^{-2}$, $\forall z \in C, l = 1, 2,$ is considered where $\theta_1 = 29.5^{\circ}$ and $\theta_2 = -35^{\circ}$. The signal $(w_n)_n$ passes through nonlinearity of Fig. 2. Represented as $p_n := w_n + 0.2w_n^2 - 0.1w_n^3, \forall n$ [5,(29)]. Independent identically distributed (i.i.d) Gaussian noise $(n_n)_n$, with zero mean and $\text{SNR} = 10\,\text{dB}$ with respect to $(p_n)_n$, is added to yeild the received signal $(x_n)_n$.

The data space is considered to be the Euclidean $\mathbb{R}^4$, thus the data are formed as $\mathbf{x}_n := (x_n, x_{n-1}, x_{n-2}, x_{n-3})^T \in \mathbb{R}^4, \forall n \in \mathbb{Z} \geq 0,$ where the superscript T stands for transposition. The traint arget (label) at time instant n, is the transmitted training symbol $s_{n-\tau}, \forall n \in \mathbb{Z} \geq 0,$ where $\tau := 1$ in our case [5]. In other words, The data space dimension $m := 4$ and the parameter τ are the equalizer order and delay, respectively. The Gaussian (RBF) kernel with kernel variance of 0.5 is considered for all the cases which offers fast convergence speed with low Bit error Rate levels. Regularization factor $\lambda = 0.01$ and nonlinear tangant hyperbolic activation function is selected with parameter c=0.4. All the NKRLS parameters are chosen in a way that maximize the performance . No sparsification is exploited in all the methods.

We compared the proposed methodology with the soft

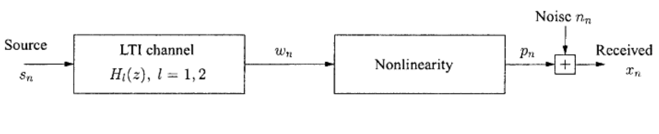

Fig. 2. Model of the nonlinear channel for which adaptive equalization is needed

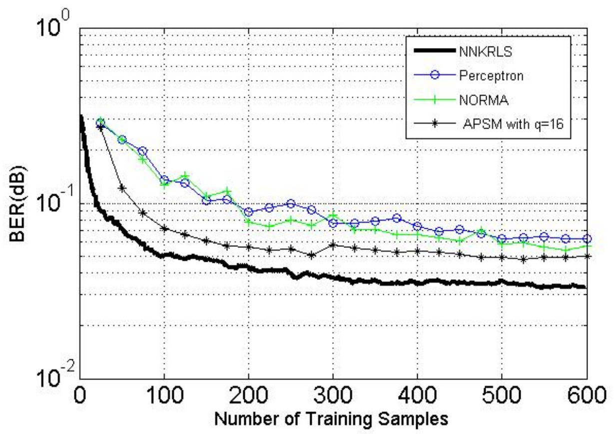

Fig. 3. Equalization performance for the LTI systems H_1 with the RBF kernel variance $\sigma^2 := 0.5$.

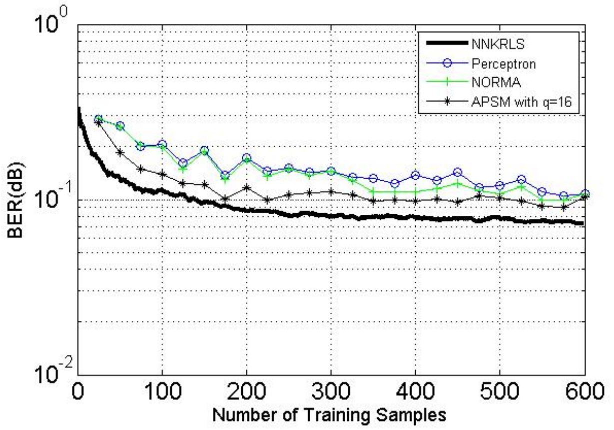

Fig. 4. Equalization performance for the LTI systems H_2 with the RBF kernel variance $\sigma^2 := 0.5$.

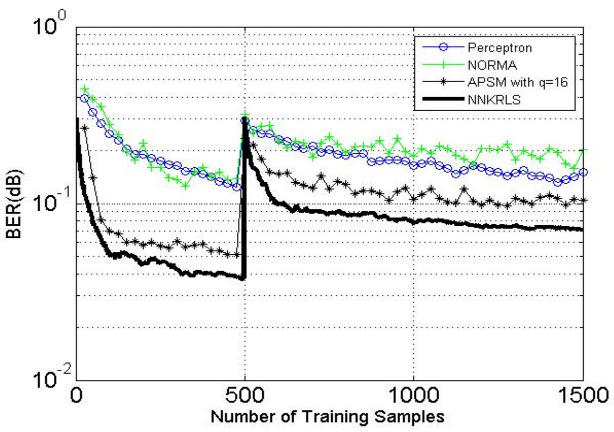

Fig. 5 . Tracking performance with the channel switch at time n=500 , from H_1 to H_2 , for the LTI systems in Fig. 2. with the RBF kernel variance $\sigma^2 := 0.5$.

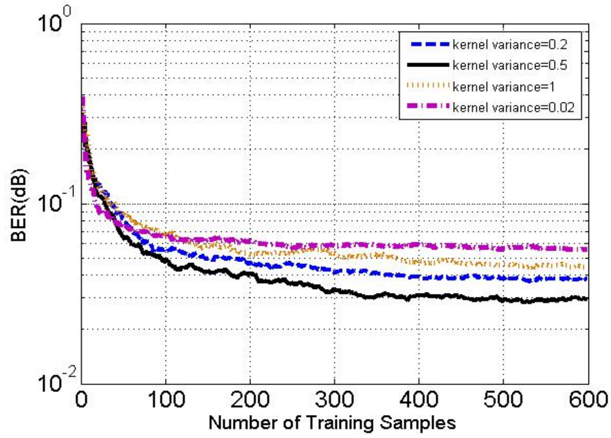

Fig. 6. kernel size effect on nonlinear channel equalization ber (dB) for the LTI systems H_1 with the RBF kernel varianc $\sigma^2 := 0.5$.

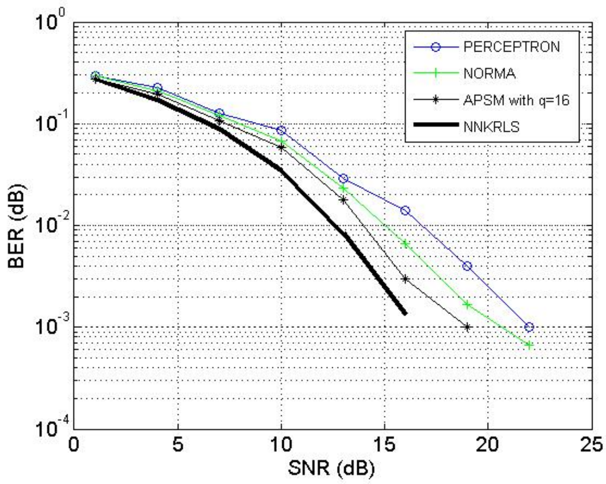

Fig. 7. Equalization performance for the LTI systems H_1 with the RBF kernel variance $\sigma^2 := 0.5$ versus SNR variation.

TABLEII

KERNEL SIZE EFFECT ON NONLINEAR CHANNEL EQUALIZATION BER (dB) OF H1

Algorithm *NNKRLS*	***KERNEL VARIANCE***	*0.02*	*0.2*	*0.5*	*1*
BER (dB)		0.055	0.038	0.029	0.045

margin version of the stochastic gradient descent method NORMA [20,sec. III-A] and the classical kernel perceptron algorithm [2],[3], which is special case of NORMA method [20,sec. VI-A]. But the main comparison is with the powerfull Adaptive Projection Subgraidient Method (APSM) [21] . ALL the methods are validated on a number of 100 test data and each curve in the figures is the result of 100 uniformly averaged experiments. By definition, the kernel perceptron method does not provide with any regularization [10]. Although NORMA offers regularization, this option is not considered here. The parameters of last three methods were chosen to maximize their performance as in [21, Fig. 6., Fig.

7., Fig. 8.] such as appropriate soft margin, concurrent processing (q=16 $q \in \mathbb{Z} > 0$) , extrapolation parameter ($\mu_n := 1, \ \forall n \in \mathbb{Z} > 0$) . we set a large value of radius for the sparsification projection ball in APSM which represents nonsparse case.

The Proposed equalizer performance versus the other three equalizers are observed in Fig. 3 and Fig. 4 (first and second N at NNKRLS stands for Nonlinear and Neuron respectively). It is evident that the proposed method has a considerably faster convergence speed with lower bit error rate than the others . In other words, the proposed method requires less pilot overhead in order to reach lower bit error rate levels. But APSM outperform the stochastic descent graidient NORMA and kernel perceptron in terms of convergence speed. Moreover, concurrent processing version of APSM with q=16 results also into a lowest bit error rate. It has been proved in [21] that kernel variance in the neihbourhood of 0.2, exhibited slow convergence and higher bit error rate levels for APSM, NORMA and kernel perceptron. It must be mentioned that H_2 has larger eigenspread than H_1. Thus the performance of all the equalization algorithms in Fig. 4 deteriorate with the data received from H_2.

The tracking performance of the adaptive equalizer after initial convergence is demonstrated for a sudden channel change , from H_1 to H_2 in Fig. 5.

The channel changes at the time instant n=500. All the parameters remain the same as before. The superiority of the ttracking performance of the proposed algorithm is absorved from the figure. Note that the performance levels before the channel change is approximately settling to the same as in Fig.3 . Also after the channel change, the performance is settling to the same as in Fig.4 .

Kernel size effect on the proposed nonlinear channel equalization bit error rate is examined in TABLEII. It is observed from Fig. 6. that too small and too large kernel variance parameter, results into slower convergence and higher bit error rate levels. Finally the bit error rate performance with snr variations is demonstrated in Fig. 7. it is observed that the proposed nonlinear equalizer BER, converges considerably faster with the lower level bit error rate at 16 dB than the other equalizers. APSM converges at 19 dB in comparison with perceptron and NORMA which converges at 22 dB. Thus, APSM, NORMA and kernel perceptron requires approximately 3-6 dB more snr, in order to match the same performance as the proposed NKRLS method.

VI. *Conclusion*

A new kernel recursive least square-type neuron (NKRLS) equalizer is proposed which improves nonlinear methods problems such as, classical training algorithm drawbacks to parameter definition, slow convergence, local minima, non-convex optimization, loss of universal approximation.. KRLS-type neuron equalizer in RKHS, is equipped with universal approximation property, convex optimization property, and facilitating kernel trick property with reregularzation. Given the nonlinear channel distotions and the nonminum phase channel characteristics, simuation results demonstrated that the proposed algorithm exhibits faster convergence with lower bit error rate levels , in comparison to the recently developed nonlinear equalizers APSM , NORMA and classical kernel perceptron techniques. Also, it was illustrated that The tracking performance of the proposed algorithm , undergone by an abrupt sudden channel change, outperforms the other nonlinear equalizers. Next, the kernel variance significance on rate of convergence and levels of bit error rate, demonstrated that too small and too large kernel variance results into slower convergence and higher bit error rate levels. Finally, APSM, NORMA and kernel perceptron has a degradation of approximately 3-6 dB snr, in comparison to the proposed NKRLS method performance.

References

[1] B. Schölkopf and A. J. Smola, *Learning With Kernels*. Cambridge, MA: MIT Press, 2002.

[2] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, 3rd ed. Amsterdam, The Netherlands: Academic Press, 2006.

[3] J. Shawe-Taylor and N. Cristianini, Kernel Methods for Pattern Analysis.Cambridge, U.K.: Cambridge University Press, 2004.

[4] F. Pérez-Cruz and O. Bousquet, "Kernel methods and their potential use in signal processing," *IEEE Signal Process. Mag.*, vol. 21, no. 3,pp. 57–65, May 2004.

[5] S. Chen, B. Mulgrew, and P. M. Grant, "A clustering technique for digital communications channel equalization using radial basis function networks," *IEEE Trans. Neural Netw.*, vol. 4, no. 4, pp. 570–579, Jul. 1993.

[6] E. Parzen, "Probability density functionals and reproducing kernel Hilbert spaces," in *Proc. Symp. Time Series Anal.*, 1963, pp. 155–169.

[7] G. Wahba, "Multivariate function and operator estimation based onsmoothing splines and reproducing kernels," in *Proc. Nonlinear Model.Forecasting*, 1992, vol. XII, pp. 95–112.

[8] V. Vapnik, *Statistical Learning Theory*. New York: Wiley, 1998.

[9] Y. Engel, S. Mannor, and R. Meir, "The kernel recursive least-squares algorithm," *IEEE Trans. Signal Process.*, vol. 52, no. 8, pp. 2275–2285, Aug. 2004.

[10] S. V. Vaerenbergh, J. Via, and I. Santamaría, "A sliding-window kernel RLS algorithm and its application to nonlinear channel identification," in Proc. *Int. Conf. Accoustics, Speech, Signal Processing* ,pp. 789–792 May 2006.

[11] W. Liu , I. Park , Y.Wang , and J. C. Príncipe. "Extended kernel recursive least squares algorithm" . *IEEE Transactions on Signal Processing* , 57 (10) ,October 2009 .

[12] G.J. Gibson, S. Siu, C. Cowan, The application of nonlinear structures to the reconstruction of binary signals, *IEEE Trans. Signal Process.*vol 39, pp.1877–1884, 1991.

[13] S. Siu, G.J. Gibson, C.F.N. Cowan, Decision feedback equalization using neural network structures and performance comparison with standard architecture, *IEE Proc.* Vol.137, pp. 221–225, 1990.

[14] S. Chen, G.J. Gibson, C.F.N. Cowan, P.M. Grant, Adaptive equalization of finite nonlinear channels using multilayer perceptrons, *Signal Process.* Vol. 20 ,pp.107–119, 1990

[15] J. Bilski, L. Rutkowski, A fast training algorithm for neural networks, *IEEE Trans. Circuits Syst.* II vol.45,no 6,pp.749–753, 1998

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

[17] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd ed.Englewood Cliffs, NJ: Prentice-Hall, 1998.

[18] S. Haykin, *Adaptive Filter Theory*, Prentice Hall, Englewood Cliffs, NJ, 1991.

[19] N. K. Bose, Digital Filters: Theory and Applications. Malabar, FL: Krieger, 1993.

[20] J. Kivinen, A. J. Smola, and R. C. Williamson, "Online learning withkernels," *IEEE Trans. Signal Process.*, vol. 52, no. 8, pp. 2165 2176, Aug. 2004.

[21] K. Slavakis, S. Theodoridis, and I. Yamada, "Online kernel-based classification using adaptive projection algorithms," *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 2781–2796, Jul. 2008.