

Square Root Extended Kernel Recursive Least Squares Algorithm for Nonlinear Channel Equalization

Bilal Shoaib, I.M. Qureshi, Ihsan ul Haq and Shahid Mehmood
Department of Electronics Engineering IIU, H-10, Islamabad Pakistan

Abstract: This study presents a square root version of extended kernel recursive least square algorithm. Basically main idea is to overcome the divergence phenomena arise in the computation of weights of the extended kernel recursive least squares algorithm. Numerically stable gives orthogonal transformations are used to obtain the next iteration of the algorithm. The usefulness of the proposed algorithm is illustrated by discussing its application on the nonlinear multipath fading channel equalization based on Rayleigh distribution. Experiments are performed on slow fading Rayleigh channel with scattered signals.

Keywords: Channel equalization, cholesky decomposition, Ex-kRLS, square root filtering

INTRODUCTION

Kernel methods became very popular in the past decade because of their firm mathematical foundation and broadly used glorious applications. Powerful nonlinear extensions to linear algorithm in RKHS have been made in the recent past which includes support vector machines, kernel principal component analysis, kernel least mean square, affine projection algorithms, kernel Adaline, normalized, leaky and Quantized KLMS (Haykin, 1998, 1996; Frieb and Harrison, 1999). Successful applications for the kernel based algorithm have been reported continuously in the various fields of signal processing for instance time series prediction, protein and DNA analysis, equalization of nonlinear channels, pattern recognition, regression and many more. Recently recursive least squares (Engel *et al.*, 2004, 2008) in RKHS have been developed and extension to this algorithm have made through L2-norm regularization (Vaerenbergh *et al.*, 2006) to improve generalization referred as Sliding Window Kernel Recursive Least Squares (SWKRLS) algorithm. Least but most important extended recursive least squares algorithm in RKHS has been developed recently by Liu *et al.* (2009). This development will be extremely useful due to the close relationship between Ex-RLS and Kalman filter. After development of Ex-KRLS in RKHS, a new research line has been formed which leads to the formulation of nonlinear kalman filter. Different nonlinear extensions towards kalman filter have been developed so far in the literature, which includes extended, unscented and cubature kalman filter. These filters use either linearization or sampling method at each iteration for state estimation. Where, in Ex-KRLS the weights of the kernel regressor is

obtained by using kalman filter with state transition process.

Many of the fast recursive algorithms suffer numerical stability issues, which typically or in some sense inborn from the parent algorithms. For example, the recursive least squares algorithm provides solution of the normal equation by obtaining inverse of correlation matrix as a difference of two correlation matrices. This difference may lose its positive definite characteristic (Sayed, 2003). Same is the case with the kernel RLS algorithm. After computation of the inverse of the correlation matrix in the feature space, numerical difficulties may arises, which results in unstable behavior and finally approaches towards divergence (Vapnic, 1995). In kernel methods the issue of negative definiteness is almost vanished, because they use Gaussian and tangent hyperbolic function to transform the incoming data. But the problem arises when the state error correlation matrix $S(i)$ is computed; there is a possibility that the resultant matrix will violate the symmetry and nonnegative property. The hypothesized general nonlinear state space model given by Liu *et al.* (2010) formulate the base for the derivation of algorithm.

$$\begin{aligned} \mathbf{x}_{i+1} &= A\mathbf{x}_i + \mathbf{n}_i \\ y_i &= \varphi(\mathbf{u}_i)^T \mathbf{x}_i + v_i \end{aligned}$$

which is similar to the model Ex-RLS (Engel *et al.*, 2004) except the input $\mathbf{u}(i)$ after transformation in the feature space becomes $\varphi(\mathbf{u}_i) \in \mathcal{F}$. $\mathbf{x}(i)$ Also now lies in the infinite dimensional feature space. Where the state transition operator has the form $A = \alpha I$. The detail of the algorithm is.

Extended Kernel Recursive Least Squares (Ex-KRLS) algorithm: Initialize the algorithm by setting

$$\begin{aligned}
 S(0) &= \lambda^{-1}I \\
 \varrho(1) &= |\alpha|^2\lambda^{-1} + \beta \\
 K(1) &= \frac{|\alpha|^2\lambda^{-2}}{\beta + \lambda^{-1}K(u(1), u(1))} \\
 &\text{Iterate for } i = 2, 3, 4, \dots \dots \dots \text{do} \\
 r(i) &= \beta^i + k(u_i, u_i) - h(i)^T z(i) \\
 e(i) &= y(i) - h(i)^T w(i-1) \\
 w(i) &= \begin{bmatrix} \alpha w(i-1) - \alpha z(i)e(i)r^{-1}(i) \\ \alpha e(i)r^{-1}(i) \end{bmatrix}
 \end{aligned}$$

$$\begin{aligned}
 K(i) &= |\alpha|^2 \\
 &\begin{bmatrix} K(i-1) + z(i)z(i)^T r(i)^{-1} & -\varrho(i-1)z(i)r(i)^{-1} \\ -\varrho(i-1)z(i)r(i)^{-1} & \varrho^2(i-1)r(i)^{-1} \end{bmatrix}
 \end{aligned}$$

- Proposed methodology:** Our goal here in this research study is to present computationally convenient square root form of the ex-KRLS. In riccati recursive equation the state error correlation matrix may result in non-negative definite. To resolve this issue we apply cholesky factorization which greatly helps us in removing the nonnegative definite character of the state error correlation matrix. Stable gives orthogonal transformations are then used to obtain the next iterations of the algorithm, which greatly helps us in achieving numerical stability and also to overcome the divergence.

Square root Ex-KRLS: we introduce here square root extended kernel recursive least squares algorithm. Riccati equation obtained from the formulation of Ex-KRLS is written as:

$$S(i) = |\alpha|^2 \left[S(i-1) - \frac{S(i-1)\varphi(u_i)\varphi(u_i)^T S(i-1)}{r(i)} \right] + \beta^i qI \quad (1)$$

Start with cholesky factorization of riccati equation the state error correlation matrix $S(i)$ propagated in its square root form, according to which:

$$S(i) = S(i)^{1/2} S(i)^{T/2} \quad (2)$$

where, the term reserved for the lower triangular matrix is $S(i)^{1/2}$ and the transposed term is $S(i)^{T/2}$ is upper triangular matrix. The matrix product written in the above equation never became negative definite, because the product and its transpose is always nonnegative definite (Pokharel *et al.*, 2009). And generally square root form of the correlation matrix is generally better than the original one in terms of numerical errors.

L-by-L matrix: correlation matrix of the predicted state error $S(i-1)$

L-by-1 vector: Transformed input vector $\varphi(u_i)$

Scaler term: variance of the innovation or state error process $\beta^i + \varphi^T(u_i)S(i-1)\varphi(u_i)$

By keeping in view the dimensionalities of the three terms we may order as:

$$H(n) = \begin{bmatrix} \beta^i + \varphi^T(u_i)S(i-1)\varphi(u_i) & \lambda^{-1/2}\varphi(u_i)^T S(i-1) \\ \lambda^{-1/2}S(i-1)\varphi(u_i) & \lambda^{-1}S(i-1) \end{bmatrix} \quad (3)$$

And can be explained in the factored form as:

$$\begin{aligned}
 H(n) &= \begin{bmatrix} \beta^{\frac{i}{2}} & \varphi(u_i)^T S^{\frac{1}{2}}(i-1) \\ 0 & \lambda^{-\frac{1}{2}} S^{\frac{1}{2}}(i-1) \end{bmatrix} \\
 &\begin{bmatrix} \beta^{i/2} & 0^T \\ S^{T/2}(i-1)\varphi(u_i) & \lambda^{-1/2} S^{T/2}(i-1) \end{bmatrix} \quad (4)
 \end{aligned}$$

The matrix product stated on the right hand side of the equation 4 may be viewed as the matrix product X and its transpose X^T . Now the stage is possibly set for invoking the matrix lemma stated earlier, according to which

$$\begin{bmatrix} \beta^{i/2} & \varphi(u_i)^T S^{1/2}(i-1) \\ 0 & \lambda^{-1/2} S^{1/2}(i-1) \end{bmatrix} \Theta = \begin{bmatrix} y_{11} & 0^T \\ y_{21} & Y_{22} \end{bmatrix} \quad (5)$$

where, Θ an orthogonal matrix, operates on X . In such a way that resulting matrix Y bearing a lower triangular structure having all its elements above the main diagonal are zero. Invoking orthogonality of transformation matrix Θ . We may write the above equation as:

$$\begin{aligned}
 \theta_i(t) &= \begin{bmatrix} I_{i-1} & \vdots & \vdots \\ \cdots & \cos\theta_i(t) & \cdots & \sin\theta_i(t) \\ \cdots & \vdots & I_{i-1} & \vdots \\ \cdots & -\sin\theta_i(t) & \cdots & \cos\theta_i(t) \end{bmatrix} \\
 &(i = 1, 2, 3, 4, \dots \dots \dots, L)
 \end{aligned}$$

$(L+1) \times (L+1)$ Identity matrix I_{L+1} the elements at the position $(i, i), (i, L+1), (L+1, i)$ and $(L+1, L+1)$. The rotation angle θ_i is carefully chosen to annihilate the elements at the position $(L+1, i)$ of the matrix to which gives rotation (Golub *et al.*, 1996) applied. By following this procedure $\varphi(i)^T S^{1/2}(i-1)$ will be eliminated term by term:

$$\begin{bmatrix} \beta^{\frac{i}{2}} & \varphi(u_i)^T S^{\frac{1}{2}}(i-1) \\ 0 & \lambda^{-\frac{1}{2}} S^{\frac{1}{2}}(i-1) \end{bmatrix} \\
 \text{X}$$

$$\begin{aligned} & \begin{bmatrix} \beta^i & 0^T \\ P^T(i-1)\varphi(u_i) & \lambda^{-\frac{1}{2}}S^T(i-1) \end{bmatrix} \\ & = \frac{\begin{bmatrix} y_{11} & 0^T \\ y_{21} & Y_{22} \end{bmatrix}}{Y} \frac{\begin{bmatrix} X^T \\ 0 & Y_{22}^T \end{bmatrix}}{Y^T} \end{aligned}$$

We proceed by comparing terms and expanding products on both sides of the equalities to get the identities:

$$Y_{11} = r(i)^{1/2}, y_{21} = g(i)r(i)^{\frac{1}{2}}, \text{ and } Y_{22} = S(i)^{1/2} \quad (6)$$

The equations defined in 6, now distinguishes between two arrays of numbers, which deserves close resemblance.

Prearray: the array of numbers on the left side of Eq. (2) is operated on by the orthogonal transformation, that is designed to annihilate the vector term $\lambda^{-\frac{1}{2}}\varphi(i)^T S(i-1)$ element by element.

Postarray: The second array of numbers, on the right hand side of Eq. (2) will be a lower triangular matrix that results from the annihilation performed by the orthogonal rotation in the prearray. The scalar term in the prearray induces the generation of two useful terms:

Square root of the mean square error covariance term $r(i)^{1/2}$ and the vector product $g(i)r(i)^{1/2}$ which makes possible in computing the gain vector.

Detail summary of Square Root Extended Kernel Recursive Least Squares (sqrEx-kRLS) Algorithm. Initialize the algorithm by setting:

$$S(0) = \lambda^{1/2}I$$

- **Given parameters:**
Transition matrix $A = \alpha I$
Transformed input vector $\varphi(u_i)$
Input correlation matrix $R(i)^{1/2}$
- **Parameters to be updated:**
Predicted estimate of the weight vector: $w(i-1)$
Predicted square-root state-error correlation matrix $S(i-1)^{1/2}$
- **Transformation from the prearray to postarray.**

$$\begin{aligned} & \begin{bmatrix} 1 & \varphi(u_i)^T S^{\frac{1}{2}}(i-1) \\ 0 & \lambda^{-\frac{1}{2}} S^{\frac{1}{2}}(i-1) \end{bmatrix} \Theta \\ & = \begin{bmatrix} r(i)^{1/2} & 0^T \\ g(i)r(i)^{1/2} & S(i)^{1/2} \end{bmatrix} \end{aligned}$$

• **Updated parameters:**

$$\begin{aligned} e(i) &= d(i) - h(i)^T c(i-1) \\ r(i) &= \beta^i + \varphi^T(u_i) S(i-1) \varphi(u_i) \\ S(i) &= S(i)^{1/2} [S(i)^{1/2}]^T \\ w(i) &= \alpha w(i-1) + g(i) e(i) \end{aligned}$$

As the computation of the matrix $S(i-1)$ will play an important role in the development of the innovation $r(i)$. By reproducing the lemmas given in Liu *et al.* (2010), greatly helps us solving these values,

Lemma 1: The matrices $S(i)$ of the reccatti recursive equation are of the following form.

$$S(k) = \varrho(k)I - M(k)Q(k)M(k)^T$$

Where the term $\varrho(k), Q(k), M(k)$ a scalar, $j \times j$ real valued and input matrix respectively.

Proof: Start with the equation:

$$\begin{aligned} S(0) &= \lambda^{-1}I \\ r(1) &= \beta + \lambda k(u(1), u(1)) \\ S(1) &= \left[\frac{|\alpha|^2}{\lambda} + \beta \right] I - \varphi(1) \left[\frac{|\alpha|^2 \lambda^{-2}}{\beta + \lambda^{-1} k(u(1), u(1))} \right] \varphi(1)^T \end{aligned}$$

If the claim is valid for $k = 1$, then by induction, assuming that it is true for $k = i - 1$, Substituting the above value of $S(1)$ in the recursive equation of the algorithm, we have:

$$S(i) = (|\alpha|^2 \varrho(i-1) + \beta^i) I - |\alpha|^2 r^{-1}(i) M(i) \begin{bmatrix} K(i-1)r(i) + z(i)z(i)^T & -\varrho(i-1)z(i) \\ -\varrho(i-1)z(i)^T & \varrho^2(i-1) \end{bmatrix} M(i)^T$$

The terms in the above equation are defined as:

$$\varrho(i) = |\alpha|^2 \varrho(i-1) + \beta^i \quad (7)$$

$$k(i) = |\alpha|^{2r-1}(i) [k(i-1)r(i)z(i)^T] - \varrho(i-1)z(i) - \varrho(i-1)z(i)^T \varrho^2(i-1) \quad (8)$$

$$S(i) = \varrho(i)I - M(i)Q(i)M(i)^T \quad (9)$$

By applying cholesky decomposition we can split the matrix $S(i)$ in upper and lower triangular matrix $S(i)^{1/2}$ and $S(i)^{T/2}$ respectively. The term $r(i)$ will be computed as:

$$\begin{aligned} r(i) &= \beta^i + \varphi^T(u_i) S(i-1) \varphi(u_i) \\ r(i) &= \beta^i + \varphi^T(u_i) [\varrho(i-1)I - M(i-1) \\ & K(i-1)M(i-1)^T] \varphi(u_i) r(i) = \beta^i + \\ & \varrho(i-1)k(u(i), u(i)) - m(i)^T z(i) \end{aligned} \quad (10)$$

The weight vector which lies in a high dimension feature space can be expressed as discussed (Liu *et al.*, 2008) in the next lemma:

Lemma 2: Start with initial weights $w(0) = 0$,

$$\begin{aligned} w(1) &= w(0) + g(1)e(1) \\ w(1) &= 0 + \frac{\alpha\lambda^{-1}\varphi(1)d(1)}{r(1)} \end{aligned} \quad (11)$$

Then by induction, assuming that it is true for $i - 1$. We pursue as:

$$\begin{aligned} w(i) &= w(i - 1) + g(i)e(i) \\ w(i) &= \alpha M(i - 1)c(i - 1) \\ &+ \alpha S(i - 1)\varphi(i)e(i)r^{-1}(i) \end{aligned}$$

By putting the value of $S(i - 1)$ from lemma 1 we have:

$$\begin{aligned} w(i) &= \alpha M(i - 1)c(i - 1) + \alpha[\varrho(i - 1)I \\ &- M(i - 1)K(i - 1)M(i - 1)^T]\varphi(i)e(i)r^{-1}(i) \\ w(i) &= [M(i - 1) \quad \varphi(i)] \\ &[\alpha c(i - 1) - \alpha z(i)e(i)r^{-1}(i)] \\ &[\alpha \varrho(i - 1)e(i)r^{-1}(i)] \end{aligned} \quad (12)$$

Weight vector described in Eq. (12) simplifies the computation of the weights of square root kRLS. The proposed algorithm is tested for the equalization of nonlinear channel discussed briefly in the simulation section.

SIMULATIONS AND EXPERIMENTAL RESULTS

We apply square root extended kernel recursive least square filter to the problem of tracking a nonlinear Rayleigh multipath channel and compare

its performance with the Ex-KRLS filter. We assumed traditional nonlinear Rayleigh fading channel, 10 paths have been chosen for this experiment with 15 assumed numbers of scatters. 500 symbols are assigned to run the experiment with maximum Doppler's frequency of 50 Hz. Sampling rate of 6μ seconds is chosen for all scattered paths. Channel matrix of specified dimensions are created and passed through the noise (AWGN) with variance 0.001. The kernel type taken here in this experiment is Gaussian with $\delta^2 = .05$. 500 symbols are used for the experiment and all the coefficients of the channel matrix are obtained assumed to be real. Mean square error has been plotted to show the effectiveness of the proposed square root Ex-RLS. (Fig. 1)

The experiment is tested on two algorithms. The first one is Ex-KRLS filter and the second algorithm is the proposed square root Ex-KRLS filter. The experimental result clearly shows that there is an improvement of approximately 3dB in terms of mean square error, will be observed while using square root Ex-RLS filter and it outperforms its Ex-KRLS and enjoys better performance in terms of mean square error. Scattered multipath channel input signal is demonstrated in the next result.

Details of the parameters of the filter used in the experiment:

Extended kernel recursive least squares algorithm:

- Kernel type: Gaussian
- Kernel parameter: 0.05
- Transition matrix: $A = \alpha I, \alpha = 0.9999$
- Forgetting factor: $\beta = 0.999$
- Regularization parameter: $\lambda = 0.01$

Square root extended kernel recursive least squares algorithm:

- Kernel type: Gaussian

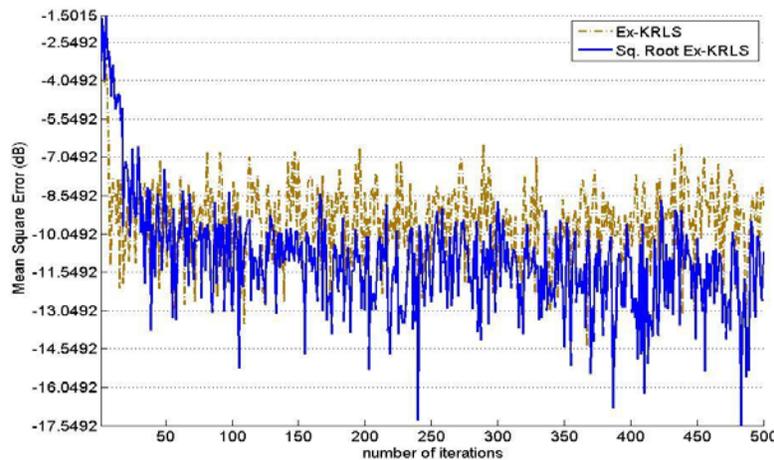


Fig. 1: Mean square error analysis

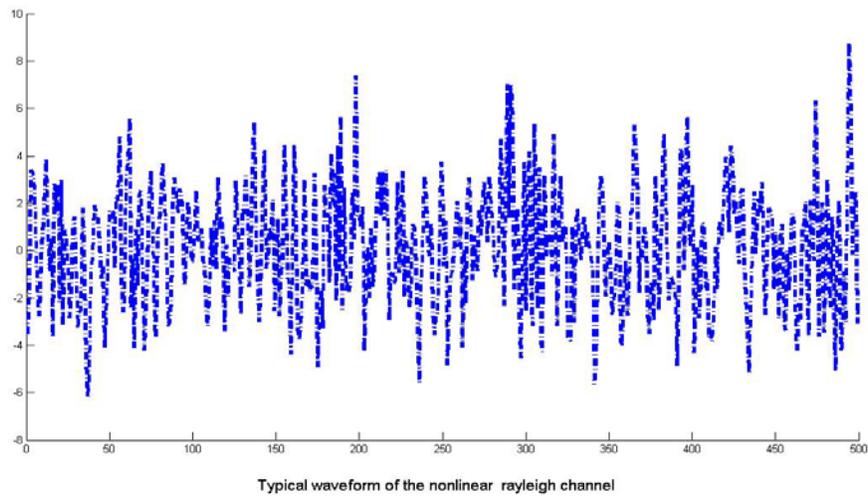


Fig. 2: Waveform of the Rayleigh fading channel

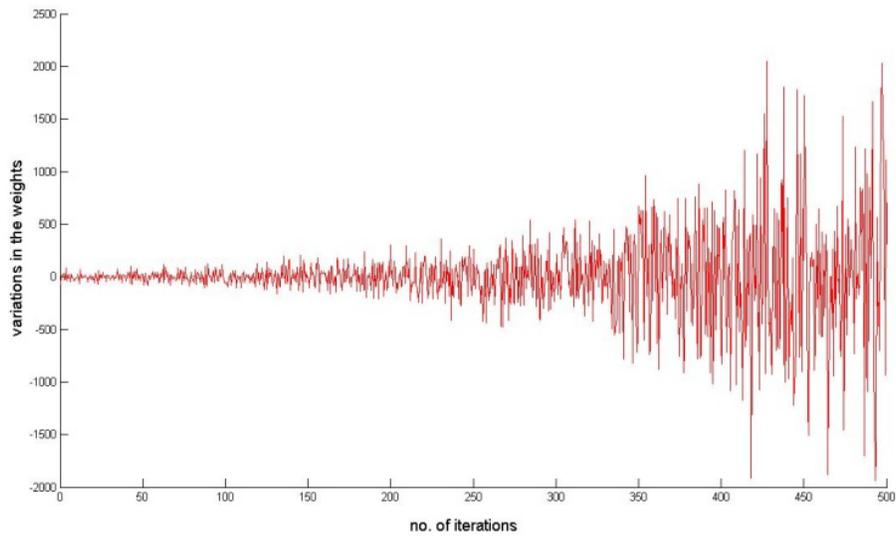


Fig. 3: Weights of the Ex-kRLS

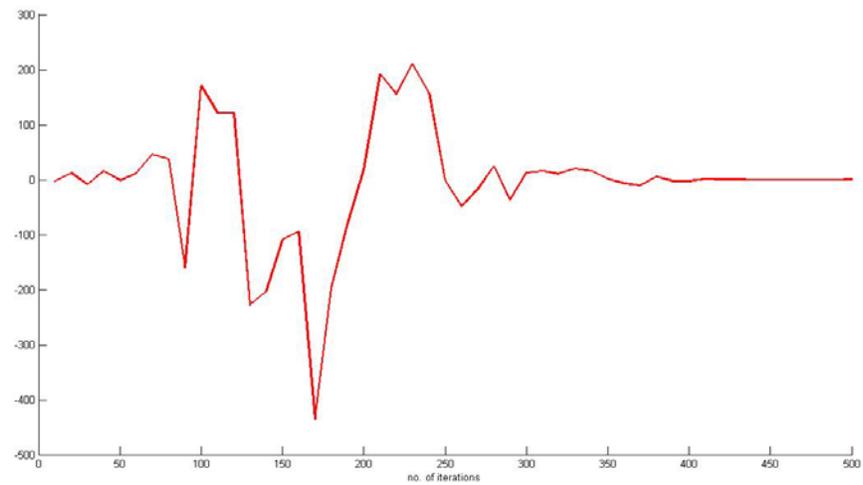


Fig. 4: Weights of the square root Ex-kRLS

Kernel parameter: 0.05
Transition matrix: $A = \alpha I$, $\alpha = 0.9997$
Forgetting factor: $\beta = 0.9$
Regularization parameter: $\lambda = 0.001$

The performed experiment is tested on two algorithms. The first one is Ex-RLS filter and the second algorithm is the proposed square root Ex-KRLS filter. The experimental result clearly shows that there is an improvement of approximately 3dB in terms of mean square error, will be observed while using square root Ex-RLS filter and it outperforms its Ex-KRLS and enjoys better performance in terms of mean square error. Scattered multipath channel input signal is demonstrated in the next result.

Figure 2 demonstrates the response of a slow fading nonlinear Rayleigh channel with number of multipath is chosen as $M = 10$. Only the real coefficients are used in this experiment. With Doppler's frequency of 50 Hz and the sampling rate of 6μ seconds. In Rayleigh fading model several objects disperse the radio signal before it reaches at the receiver. In our model we assume 15 no. of scatterers.

Figure 3 and 4 demonstrates the weights of both the algorithms, Ex-kRLS and sq.root Ex-kRLS. Which clearly support our claim? Figure 4 shows the weights of the proposed algorithm. The weights of the square root algorithm achieve stable behavior in last 200 iterations.

CONCLUSION AND RECOMMENDATIONS

In this study, a new square root extended kernel recursive least square algorithm is proposed based on cholesky decomposition and givens rotation. In order to overcome the problem of divergence in the weights of Ex-kRLS, the concept of square root filtering is applied in this study. The proposed algorithm is tested on nonlinear Rayleigh fading channel equalization problem illustrates that proposed square root Ex-kRLS algorithm achieves better tracking performance in comparison with the Ex-kRLS. The learning curves of the weights of both the algorithm are observed, which clearly demonstrates that proposed algorithm achieve early after the few no. of iterations to a stable state while the other didn't achieve its stable state throughout

the iterations. Kernel algorithm gives better tracking performance in comparison with their linear counterparts, but this superiority is availed at the cost of greater computation complexity. Square root kernel RLS should be tested in the future for nonlinear problems.

REFERENCES

- Engel, Y., S. Mannor and R. Meir, 2004. The kernel recursive least-squares algorithm. *IEEE T. Signal Process.*, 52(8): 2275-2285.
- Engel, Y., S. Mannor and R. Meir, 2008. The kernel recursive least-squares algorithm. *IEEE T. Signal Process.*, 52(8): 543-554.
- Frieb, T. and R. Harrison, 1999. A kernel-based adaline for function approximation. *Intell. Data Anal.*, 3(4): 307-313.
- Golub, G.H. and C.F. Loan, 1996. *Matrix Computations*. 3rd Edn., The John Hopkins University Press, Baltimore, MD, USA.
- Haykin, S., 1996. *Adaptive Filter Theory*. Prentice Hall PTR, Upper Saddle River, NJ.
- Haykin, S., 1998. *Neural Networks: A Comprehensive Foundation*. 2nd Edn., Englewood Cliffs, NJ.
- Liu, W., P. Pokharel and J.C. Principe, 2008. The kernel least-mean-square algorithm. *IEEE T. Signal Process.*, 56(2): 543-554.
- Liu, W., Y. Wang and J.C. Principe, 2009. Extended kernel recursive least squares algorithm. *IEEE T. Signal Process.*, 57(10): 2215-2224.
- Liu, W., J.C. Principe and S. Haykin, 2010. *Kernel Adaptive Filtering*. Wiley, New York.
- Pokharel, P.P., W. Liu and J.C. Principe, 2009. Kernel least mean square algorithm with constrained growth. *ELSEVIER, Signal Process.*, 89: 257-265.
- Sayed, A.H., 2003. *Fundamentals of Adaptive Filtering*. Wiley, New York.
- Vaerenbergh, S., V.J. Via and I. Santamara, 2006. A sliding-window kernel RLS algorithm and its application to nonlinear channel identification. *Proceeding of International Conference Acoustics, Speech, Signal Processing*, pp: 789.
- Vapnic, V., 1995. *The Nature of Statistical Learning Theory*. Springer, New York.