Transactions of the Nigerian Association of Mathematical Physics
Volume 7, (March, 2018), pp179 – 184
© Trans. of NAMP

EXPONENTIALLY WEIGHTED BLOCK QR DECOMPOSITION-RECURSIVE LEAST SQUARES ALGORITHM

F.Z. Okwonu

DEPARTMENT OF MATHEMATICS, DELTA STATE UNIVERSITY, ABRAKA, NIGERIA.

Abstract

We considered preconditioned iterative algorithm by introducing non-orthogonal transform as an update recursion of the standard QR decomposition(Q is an orthonormal column vector and R is upper triangular matrix with positive diagonal entries) based recursive least squares algorithm by adapting an upper triangular block diagonal matrix. The Givens rotation is applied to derive the posterior and a priori error without the explicit computation of the tap weight coefficient vector. This computation requires the successive cosine term such that the performance function is minimized by choosing the tap weight coefficient and compute the upper triangular system via back-substitution. We extend the existing techniques by introducing an exponentially weighted factor. This technique has similar convergence property, stability as the conventional QRD-RLS approach. Although, the two techniques exhibit the tractability of signal information and are stable against white noise. The tractability of both techniques depends on the values assigned to the forgetting or control variable (α)***. As the value of the forgetting factor increases, the tractability and stability of signal information is observed. The learning curve indicates that the exponentially weighted block QRD-RLS algorithm is consistent and efficient.***

Keywords: QR decomposition recursive least squares, Givens rotations, adaptive filter.

1.0 Introduction

The standard recursive least squares algorithm recursively updates the weights using the matrix inversion lemma. A commonly used alternative solutions performs a set of orthogonal rotations on the incoming data thereby transforming the over specified rectangular data matrix into upper triangular form. The weight are then obtained by back substitution [1].This technique is called the QR decomposition (QRD) based recursive least squares (RLS). The standard recursive least squares algorithm has been proved to be numerically unstable when implemented in finite word length. This problem has generated a lot of research interest on how to reduce the computational complexity and improves its numerical properties. As such, the fast recursive least square algorithm utilizes the QR decomposition (Givens rotations or Householder transformation), this approach uses the data matrix [2].However, the orthogonal decomposition is a well known approach to eliminate numerical instability problem [3, 4].The QR decomposition possesses good numerical properties and it's possible implementation in systolic array which makes it interesting for real time application [5]. This study extend the QR-decomposition recursive least square by applying exponential weight factor. This procedure apply a non-orthogonal transform that recursively update the data matrix to generate the 2×2 block diagonal algorithm.

This paper is organized as follows. Section two briefly introduces the adaptive filter structure. The QR decomposition based recursive least squares algorithm using Givens rotations is considered in Section Three. The 2 by 2 block –diagonal algorithm is presented in Section Four Conclusions are drawn in Section Five.

2.0 Adaptive Filtering

Adaptive filtering can be considered as a process whereby the parameters used in signal processing changes according to some criterion. Normally these criteria are the estimated mean squared error or the correlation [1, 6]. Adaptive filters are time varying since their parameters are continually changing in other to meet a specify performance requirement as such adaptive filter can be interpreted as filter that performs approximation step on-line.

Corresponding Author: Okwonu F.Z., Email: fzokwonu_delsu@yahoo.com, Tel: +2347039151870

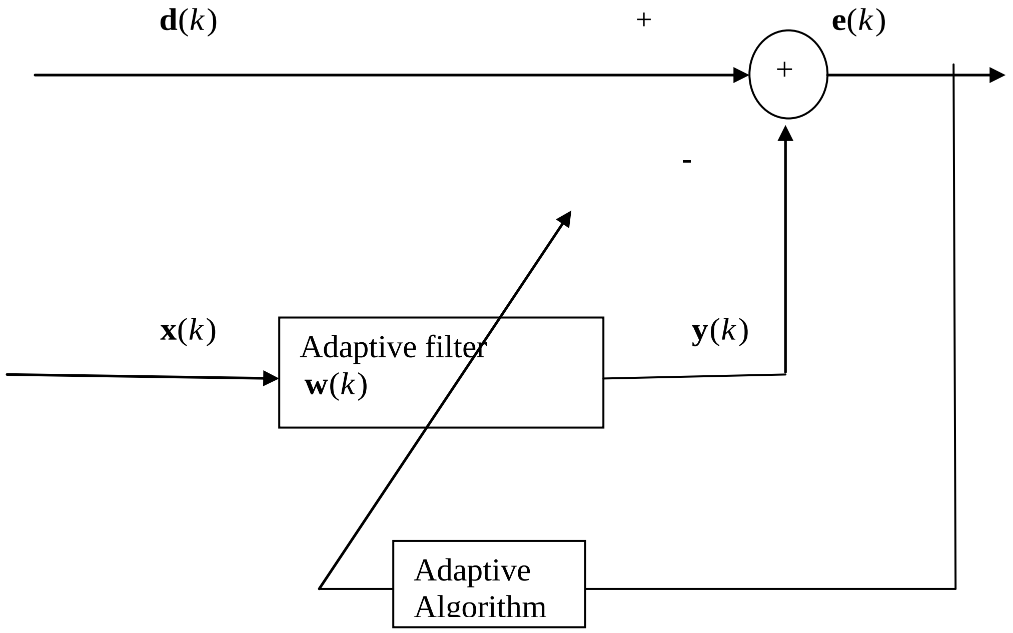

Fig .1.Structure of adaptive filter.

The structure above is the general configuration of the adaptive filter [6], where k is the length of iterations, $\mathbf{d}(k)$ is the reference signal or desired signal, $\mathbf{x}(k)$ is the input signal vector and

$$\mathbf{y}(k) = \mathbf{x}^T(k)\mathbf{w}(k) \qquad (2.1)$$

is the adaptive filter output. Where

$$\mathbf{e}(k) = \mathbf{d}(k) - \mathbf{x}^T(k)\mathbf{w}(k) \qquad (2.2)$$

is the error signal used to form the performance function that is required by the adaptation algorithm in other to determine the appropriate updating of the adaptive filter coefficient $w(k)$. The minimization of the performance function implies that there is a corresponding relationship between the adaptive filter output and the reference signal.

3.0 QRD- RLS USING GIVENS ROTATIONS

The triangularization approach can be applied to generate the QR-RLS algorithm[1,7]. The recursive least squares algorithm generate the recursive process the coefficient of the adaptive filter[6] minimizes the following performance function;

$$\varepsilon(k) = \sum_{i=0}^{k} \lambda^{k-i}\mathbf{e}^2(k) = \sum_{i=0}^{k} \lambda^{k-i}[\hat{\mathbf{d}}(k) - \mathbf{x}^T(k)\mathbf{w}(k)]^2 \qquad (3.1)$$

where $\mathbf{x}(k) = [x(k) \quad x(k-1) \quad ... \quad x(k-N)]^T$ is the input signal vector and $\mathbf{w}(k) = [w_0(k) \quad w_1(k) \quad ... \quad w_N(k)]^T$ is the filter coefficient at instant k. Where $\mathbf{e}(k)$ is the a posteriori error at instant k and λ is the forgetting factor. The above equation can be written as a function of increasing dimension matrices and vectors.

$$\mathbf{X}(k) = [\mathbf{x}(k) \quad \lambda^{1/2}\mathbf{x}(k-1) \quad ... \quad \lambda^{k/2}\mathbf{x}(0)]^T$$

$$y(k) = \mathbf{x}(k)\mathbf{w}(k) = [y(k) \quad \lambda^{1/2}(k-1) \quad ... \quad \lambda^{k/2}y(0)]^T$$

and $\mathbf{d}(k) = [\mathbf{d}(k) \quad \lambda^{1/2}\mathbf{d}(k-1) \quad ... \quad \lambda^{k/2}\mathbf{d}(0)]^T$ .Where $\hat{\mathbf{e}}(k)$ is the error vector containing the weighted past error values $\lambda^{k-i/2}\mathbf{e}(k)$.

$$\hat{\mathbf{e}}(k) = [\mathbf{e}(k)\lambda^{1/2}\mathbf{e}(k-1),...,\lambda^{1/2}\mathbf{e}(0)]^T = \hat{\mathbf{d}}(k) - \mathbf{x}(k)\mathbf{w}(k) \qquad (3.2)$$

Since each Givens rotation matrix is orthogonal, then it can be proved that $\mathbf{Q}(k)$ is also orthogonal [8] i.e., $\mathbf{Q}(k)\mathbf{Q}^T(k) = I_{k+1}$ .As such we have

$$\hat{\mathbf{Q}}(k)\mathbf{X}(k) = \hat{\mathbf{Q}}(k)\begin{bmatrix} 1 & 0 \\ 0 & \hat{\mathbf{Q}}(k-1)\end{bmatrix}\begin{bmatrix} I_2 & 0 \\ 0 & \hat{\mathbf{Q}}(k-2)\end{bmatrix}...\begin{bmatrix} I_{k-N} & 0 \\ 0 & \hat{\mathbf{Q}}(k-N)\end{bmatrix}\mathbf{X}(k) = \begin{bmatrix} 0 \\ \mathbf{U}(k)\end{bmatrix} \qquad (3.3)$$

where $\mathbf{Q}(k)$ is $(k+1)\times(k+1)$ matrix which is the overall triangularization matrix via elementary Givens rotation matrices. From (3.3) we note that

$$\mathbf{Q}(k)=\hat{\mathbf{Q}}(k)\begin{bmatrix}1 & 0\\ 0 & \mathbf{Q}(k-1)\end{bmatrix} \tag{3.4}$$

which is the recursive nature of $\mathbf{Q}(k)$ and $\hat{\mathbf{Q}}(k)$ is responsible for zeroing $\mathbf{x}^T(k)$ .By pre-multiplying (3.2) by $\mathbf{Q}(k)$, we obtain

$$\mathbf{Q}(k)\hat{\mathbf{e}}(k)=\hat{\mathbf{e}}_q(k)=\begin{bmatrix}\hat{\mathbf{e}}_{q1}(k)\\ \hat{\mathbf{e}}_{q2}(k)\end{bmatrix}=\begin{bmatrix}\hat{\mathbf{d}}_{q1}(k)\\ \hat{\mathbf{d}}_{q2}(k)\end{bmatrix}-\begin{bmatrix}0\\ \mathbf{U}(k)\end{bmatrix}\mathbf{w}(k) \tag{3.5}$$

where $\mathbf{U}(k)$ is the upper triangular matrix .The subscripts 1 and 2 indicate that the first $k-N$ and the last $N+1$ components of the vector. The performance function can be minimized by choosing $\mathbf{w}(k)$ such that $\hat{\mathbf{d}}_{q2}(k)-\mathbf{U}(k)\mathbf{w}(k)$ is zero. The tap weight coefficients are then calculated by using back substitution. From (3.3) we have

$$\mathbf{Q}(k)\mathbf{X}(k)=\hat{\mathbf{Q}}(k)\begin{bmatrix}x(k) & x(k-1)... & x(k-N)\\ 0 & 0 & 0\\ : & : & :\\ 0 & 0 & 0\\ & \lambda^{1/2}\mathbf{U}(k-1) & \end{bmatrix} \tag{3.6}$$

where $\lambda^{1/2}\mathbf{U}(k-1)$ is a triangular matrix. Consider the intermediate calculation of (3.6)

$$\hat{\mathbf{Q}}(k)\begin{bmatrix}\mathbf{x}^T(k)\\ 0\\ \lambda^{1/2}\mathbf{U}(k-1)\end{bmatrix}=\mathbf{Q}'_N(k).\mathbf{Q}'_{N-1}(k)...\mathbf{Q}'_0(k)\begin{bmatrix}\mathbf{x}'_i(k)\\ 0\\ \mathbf{U}'_i(k)\end{bmatrix} \tag{3.7}$$

where $\mathbf{x}'_i(k)=\begin{bmatrix}x'_i(k) & x'_i(k-1),..., & x'_i(k-N)0.. & 0\end{bmatrix}$ and $\mathbf{U}'_i(k)$ is an intermediate upper triangular matrix ,which implies that

$$\hat{\mathbf{Q}}(k)\begin{bmatrix}\mathbf{x}^T(k)\\ 0\\ \lambda^{1/2}\mathbf{U}(k-1)\end{bmatrix}=\begin{bmatrix}0\\ \mathbf{U}(k)\end{bmatrix} \tag{3.8}$$

By deleting the increasing section of I_{k-N-1} of $\hat{\mathbf{Q}}(k)$ thereby generating a matrix with reduced dimension $\mathbf{Q}_\theta(k)$ .By this we can rewrite (3.7) as

$$\mathbf{Q}_\theta(k)\begin{bmatrix}\mathbf{x}^T(k)\\ \lambda^{1/2}\mathbf{U}(k-1)\end{bmatrix}=\mathbf{Q}'_{\theta N}(k).\mathbf{Q}'_{\theta N-1}(k)...\mathbf{Q}'_{\theta 0}(k)\begin{bmatrix}x'_i(k)\\ \mathbf{U}'_i(k)\end{bmatrix}=\begin{bmatrix}0\\ \mathbf{U}(k)\end{bmatrix} \tag{3.9}$$

where $\mathbf{Q}_\theta(k)$ is derived from $\hat{\mathbf{Q}}(k)$ along with the corresponding rows and columns. From (3.5) we observe that $\hat{\mathbf{e}}_{q1}(k)=\hat{\mathbf{d}}_{q1}(k)$ and the product of $\mathbf{Q}(k)\mathbf{d}(k)$ can be written as

$$\begin{bmatrix}\hat{\mathbf{e}}_{q1}(k)\\ \hat{\mathbf{d}}_{q2}(k)\end{bmatrix}=\hat{\mathbf{Q}}(k)\begin{bmatrix}\hat{\mathbf{d}}(k)\\ \lambda^{1/2}\begin{bmatrix}\mathbf{e}_{q1}(k-1)\\ :\\ \mathbf{e}_{qk-N-1}(k-1)\\ \mathbf{d}_{q2}(k-1)\end{bmatrix}\end{bmatrix} \tag{3.10}$$

Where

$$\hat{\mathbf{Q}}(k)\begin{bmatrix}\hat{\mathbf{d}}(k)\\ \lambda^{1/2}\begin{bmatrix}\hat{\mathbf{e}}_{q1}(k-1)\\ \hat{\mathbf{d}}_{q2}(k-1)\end{bmatrix}\end{bmatrix}=\mathbf{Q}'_N(k).\mathbf{Q}'_{N-1}(k)...\mathbf{Q}'_0(k)\begin{bmatrix}\hat{\mathbf{d}}_{qi}(k)\\ \hat{\mathbf{e}}_{qi}(k)\\ \hat{\mathbf{d}}_{q2i}(k)\end{bmatrix} \tag{3.11}$$

where $\hat{\mathbf{d}}_{qi}(k)$, $\hat{\mathbf{e}}_{qi}(k)$ and $\hat{\mathbf{d}}_{q2i}(k)$ are intermediate quantities generated during the rotation

$\hat{\mathbf{d}}_{q1}(k)=\mathbf{e}_{q1}(k)$

$$\begin{bmatrix}\hat{\mathbf{e}}_{q1}(k)\\ \hat{\mathbf{d}}_{q2}(k)\end{bmatrix}=\hat{\mathbf{Q}}(k)\begin{bmatrix}\hat{\mathbf{d}}_{q1}(k)\\ \lambda^{1/2}\hat{\mathbf{e}}_{q1}(k-1)\\ \lambda^{1/2}\hat{\mathbf{d}}_{q2}(k-1)\end{bmatrix}=\begin{bmatrix}\mathbf{e}_{q1}(k)\\ \lambda^{1/2}\hat{\mathbf{e}}_{q1}(k-1)\\ \hat{\mathbf{d}}_{q2}(k)\end{bmatrix} \tag{3.12}$$

That is

$$\begin{bmatrix}\hat{\mathbf{e}}_{q1}(k)\\ \hat{\mathbf{d}}_{q2}(k)\end{bmatrix}=\hat{\mathbf{Q}}(k)\begin{bmatrix}1 & 0\\ 0 & Q(k-1)\end{bmatrix}\begin{bmatrix}\hat{\mathbf{d}}(k)\\ \lambda^{1/2}\hat{\mathbf{d}}(k-1)\end{bmatrix}=\hat{\mathbf{Q}}(k)\begin{bmatrix}\hat{\mathbf{d}}(k)\\ \lambda^{1/2}\hat{\mathbf{e}}(k-1)\\ \lambda^{1/2}\hat{\mathbf{d}}(k-1)\end{bmatrix}=\begin{bmatrix}\mathbf{e}_{q1}(k)\\ \lambda^{1/2}\hat{\mathbf{e}}_{q1}(k-1)\\ \hat{\mathbf{d}}_{q2}(k)\end{bmatrix}$$

Applying (3.10), (3.11) and (3.12) we can easily understand if we observed in (3.7) that $\hat{\mathbf{Q}}(k)$ only alter the first and the last $N+1$ components of the vector. From (3.12) it is possible to remove the increasing section of $\hat{\mathbf{Q}}(k)$ [2] to obtain

$$\mathbf{Q}_\theta(k)\begin{bmatrix}\hat{\mathbf{d}}_{q1}(k)\\ \lambda^{1/2}\hat{\mathbf{d}}_{q2}(k-1)\end{bmatrix}=\begin{bmatrix}\mathbf{e}_{q1}(k)\\ \hat{\mathbf{d}}_{q2}(k)\end{bmatrix} \qquad (3.13)$$

Where $\mathbf{e}_{q1}(k)$ is the first element of the rotated error signal and $\hat{\mathbf{d}}_{q2}(k)$ is a vector with the last $N+1$ element of the rotated desired signal vector. Expression (3.9) gives the detail structure of $\mathbf{Q}_\theta(k)$ as a product of Givens rotation matrices.

The transformation of the data matrix into upper triangular matrix results in the input data $\hat{\mathbf{x}}_i{}'(k)=\left[x_i'(k) \quad x_i'(k-1).... \quad x_i'(k-N-i) \quad 0... \quad 0\right]$ and $\mathbf{U}_i'(k)$ are assumed intermediate upper triangular matrix and note that $\hat{\mathbf{x}}_0'(k)=\hat{\mathbf{x}}^T(k)$, $\mathbf{U}_0'(k)=\lambda^{1/2}\mathbf{U}(k-1)$ and $\mathbf{U}_{N+1}'(k)=\mathbf{U}(k)$. We conclude that the matrix $\mathbf{U}(k)$ is triangularised as a upper triangular matrix with the corresponding structure of $\mathbf{Q}_{\theta i}'(k)$ [1, 4,9]

$$\mathbf{Q}_{\theta i}'(k)=\begin{bmatrix}\cos\theta_i(k) & 0^T & -\sin\theta_i(k) & 0^T\\ 0 & I_i & 0 & 0\\ \sin\theta_i(k) & 0^T & \cos\theta(k)_i & 0^T\\ 0 & 0 & 0 & I_{N-i}\end{bmatrix}$$

The Givens rotation elements are calculated as

$$\left.\begin{aligned}\cos\theta_i(k)&=\frac{[\mathbf{U}_i'(k)]_{i+1,N+1-i}}{c_i}\\ \sin\theta_i(k)&=\frac{x'(k-N-i)}{c_i}\end{aligned}\right\} \qquad (3.14)$$

where $c_i=\sqrt{\left[\mathbf{U}_i'(k)_i\right]^2_{i+1,N+1-i}+x_i'^2(k-N-i)}$ and $[.]_{i,j}$ is the (i,j) element of the matrix.

Matrix $\mathbf{U}(k)$ and $\hat{\mathbf{d}}_{q2}(k)$ are updated recursively. Vector $\mathbf{w}(k)$ is calculated using the back- substitution algorithm to solve $\mathbf{U}(k)\mathbf{w}(k)=\hat{\mathbf{d}}_{q2}(k)$. From the definition of $\mathbf{Q}(k)$ we obtain the following relation:

$$\varepsilon(k)=\lambda\mathbf{e}_{q1}(k)\psi(k)$$

which shows that the a posteriori error can be computed without explicit computation of the coefficient vector. The only information required is the Givens rotation cosines and the exponential weighting factor. The a priori error can be computed as $\mathbf{e}^*(k)=\dfrac{\mathbf{e}_{q1}(k)}{\psi(k)}$. The exponentially weighted QRD-RLS algorithm is summarized as follows:

```
For each $k$
{ Compute $\mathbf{Q}_\theta(k)$ and update $\mathbf{U}(k)$:
  $\begin{bmatrix}0^T\\ \mathbf{U}(k)\end{bmatrix}=\mathbf{Q}_\theta(k)\begin{bmatrix}\mathbf{x}^T(k)\\ \lambda^{1/2}\mathbf{U}(k-1)\end{bmatrix}$
  Compute $\psi(k)$:
  $\psi(k)=\prod_{i=0}^{N}\cos\theta_i(k)$
  Compute $\mathbf{e}_{q1}(k)$ and update $\mathbf{d}_{q2}(k)$:
  $\begin{bmatrix}\mathbf{e}_{q1}(k)\\ \mathbf{d}_{q2}(k)\end{bmatrix}=\mathbf{Q}_\theta(k)\begin{bmatrix}\hat{\mathbf{d}}(k)\\ \lambda^{1/2}\mathbf{d}_{q2}(k-1)\end{bmatrix}$
  Compute $\varepsilon(k)$:
  $\mathbf{e}(k)=\mathbf{d}(k)-\mathbf{x}^T(k)\mathbf{w}(k)$
  $\varepsilon(k)=\lambda\mathbf{e}_{q1}(k)\psi(k)$
}.
```

4.0 2×2 BLOCK –DIAGONAL ALGORITHM

The least squares parameter estimation problem is the minimization problem expressed as

$$\min_{\mathbf{w}(k)}[\varepsilon(k)] \qquad (4.1)$$

where

$$\varepsilon(k)=\mathbf{e}^T(k)\mathbf{Q}^T(k)\mathbf{Q}(k)\mathbf{e}(k)=\mathbf{e}_q^T(k)\mathbf{e}_q(k)=\left\|\mathbf{Q}^T(k)\mathbf{e}(k)\right\|^2$$

Since $\mathbf{Q}(k)$ is orthogonal matrix[8] and recall that $\hat{\mathbf{Q}}(k)$ is responsible for zeroing $\mathbf{x}^T(k)$. From (3.5) we obtain

$$\mathbf{Q}(k)\mathbf{e}(k)=\mathbf{e}_q(k)=\begin{bmatrix}\mathbf{d}_{q1}(k)\\ \mathbf{d}_{q2}(k)\end{bmatrix}-\begin{bmatrix}\mathbf{0}\\ \mathbf{U}(k)\end{bmatrix}\mathbf{w}(k),$$

where $\mathbf{U}(k)$ is the upper triangular matrix. The subscripts 1 and 2 indicate that the first $k-N$ and the last $N+1$ components of the vector. The performance function can be minimized by choosing $\mathbf{w}(k)$ such that $\mathbf{d}_{q2}(k)-\mathbf{U}(k)\mathbf{w}(k)$ is zero. The tap weight coefficient is obtained via back-substitution algorithm.

$$\hat{\mathbf{d}}_{q2}(k)=\mathbf{U}(k)\mathbf{w}(k) \qquad (4.2)$$

Based on the above, we introduce a regularizing transform that is non-orthogonal to update the QRD-RLS algorithm. The matrix $\mathbf{U}(k)$ is partitioned as follows

$$\mathbf{U}(k)=\begin{bmatrix}\mathbf{U}_1(k) & \mathbf{A}_{12}\\ 0 & \mathbf{U}_2(k)\end{bmatrix} \qquad (4.3)$$

Where $\mathbf{U}_1(k)$ and $\mathbf{U}_2(k)$ are 1×1 and 2×2 upper triangular matrices and $\mathbf{A}_{12}$ is 1×2 matrix. Compute the coefficient vector which is the solution to $\mathbf{d}_{q2}(k)-\mathbf{U}(k)\mathbf{w}(k)$. By (4.2) and using the non-orthogonal transform (4.2) can be written as

$$\mathbf{U}'(k)\mathbf{w}(k)=\mathbf{d}_{\mathbf{q}'\mathbf{2}}(k) \qquad (4.4)$$

where $\mathbf{U}'(k)$ and $\hat{\mathbf{d}}_{q'2}(k)$ are used as an update equation. From (3.9) and (3.13) we have

$$\mathbf{Q}_\theta(k)\begin{bmatrix}\mathbf{x}^T(k+1)\\ \lambda^{1/2}\mathbf{U}'(k)\end{bmatrix}=\begin{bmatrix}0\\ \mathbf{U}(k+1)\end{bmatrix} \text{ and } \begin{bmatrix}\mathbf{e}_{q1}(k+1)\\ \hat{\mathbf{d}}_{q2}(k+1)\end{bmatrix}=\mathbf{Q}_\theta(k)\begin{bmatrix}\hat{\mathbf{d}}_{q1}(k+1)\\ \lambda^{1/2}\hat{\mathbf{d}}_{q'2}(k)\end{bmatrix}.$$

The transformation from (4.2) to (4.3) reduces the matrix $\mathbf{U}(k)$ to its block –diagonal submatrix. The extra diagonal block $\mathbf{A}_{12}$ is re-assigned to (4.2), that is,

$$\mathbf{U}'(k)=\begin{bmatrix}\mathbf{U}_1(k) & 0\\ 0 & \mathbf{U}_2(k)\end{bmatrix} \text{ and } \hat{\mathbf{d}}_{q'2}(k)=\hat{\mathbf{d}}_{q2}(k)-\begin{bmatrix}0 & \mathbf{A}_{12}\\ 0 & 0\end{bmatrix}\mathbf{w}(k)=\hat{\mathbf{d}}_{q2}(k)-\mathbf{A}_{12}\mathbf{w}(k)$$

One can see that the submatrix $\mathbf{A}_{12}$ is a rank one matrix. We note the following relations

$$\mathbf{w}_2(k)=\mathbf{U}_2^{-1}(k)\hat{\mathbf{d}}_{q2}(k)$$

$$\hat{\mathbf{d}}_{q'1}(k)=\hat{\mathbf{d}}_{q1}(k)-\mathbf{A}_{12}\mathbf{w}_2(k)$$

$$\mathbf{w}_1(k)=\mathbf{U}_1^{-1}(k)\hat{\mathbf{d}}_{q'1}(k)$$

Where $\hat{\mathbf{d}}_{q'}(k)=\begin{bmatrix}\hat{\mathbf{d}}_{q'1}(k)\\ \hat{\mathbf{d}}_{q'2}(k)\end{bmatrix}$ and $\mathbf{w}(k)=\begin{bmatrix}\mathbf{w}_1(k)\\ \mathbf{w}_2(k)\end{bmatrix}$.

The 2×2 block–diagonal algorithm is summarized as follows:

```
1. Initialization:
$\mathbf{U}'(-1) = \mu I_N$
$\hat{\mathbf{d}}_{q'}(-1) = \mu \mathbf{w}(-1)$
        for $k = 0 : m$
2. $\mathbf{Q}_\theta(k)\begin{bmatrix} \mathbf{x}^T(k) \\ \lambda^{1/2}\mathbf{U}'(k-1) \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{U}(k) \end{bmatrix}$
$\mathbf{U}(k) = \begin{bmatrix} \mathbf{U}_1(k) & \mathbf{A}_{12} \\ 0 & \mathbf{U}_2(k) \end{bmatrix}$
$\mathbf{Q}_\theta(k)\begin{bmatrix} \hat{\mathbf{d}}_{q1}(k) \\ \lambda^{1/2}\hat{\mathbf{d}}_{q'2}(k-1) \end{bmatrix} = \begin{bmatrix} \mathbf{e}_{q1}(k) \\ \hat{\mathbf{d}}_{q2}(k) \end{bmatrix}$
3. $\mathbf{w}(k) = \mathbf{U}^{-1}(k)\hat{\mathbf{d}}_{q'}(k)$
4. $\mathbf{U}'(k) = \begin{bmatrix} \mathbf{U}_1(k) & 0 \\ 0 & \mathbf{U}_2(k) \end{bmatrix}$
$\hat{\mathbf{d}}_{q'2}(k) = \hat{\mathbf{d}}_{q2}(k) - \mathbf{A}_{12}\mathbf{w}(k)$
5. $\mathbf{e}(k) = \hat{\mathbf{d}}(k) - \mathbf{x}^T(k)\mathbf{w}(k)$
```

We observe that the transformation of step (4) from equation (4.2) and (4.3) improves the conditioning of the system; hence this algorithm is a preconditioned iterative algorithm.

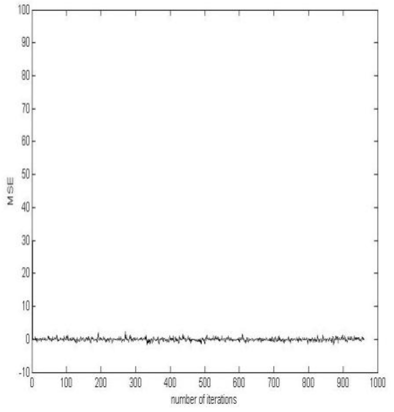

Fig .2.Learning curve of the proposed algorithm (exponentially weighted QR decomposition based recursive least squares algorithm).

5.0 Conclusion

In this paper, we used the triangularization approach to generate the exponentially weighted QR decomposition based recursive least squares algorithm and apply a non-orthogonal transform that recursively update the data matrix to generate the 2×2 block diagonal algorithm. The algorithms proposed so far are assumed numerically robust and stable and possess the fast convergence properties of the conventional recursive least squares algorithm.

References

[1] Diniz, P.S. R., Adaptive filtering: Springer +business, media LLC 2008.
[2] Chen, S. C., and Yang, X.X, (2004).Improved approximate QR-LS algorithms for adaptive filtering, IEEE, transformations on circuit and system 11, express brief, Vol.51,No1.29-39.
[3] Raymond, C.G.H.C., and O'Leary, D.P. Milestones in matrix computation: Selected works of Gene H.Golub, with commentaries, (ed). Oxford University Press, 2007.
[4] Li.M, He, S. and Li. X., (2009). Complex radial basis function networks trained by QR decomposition recursive least square algorithms applied in behavioural modeling of nonlinear power amplifiers.Wiley periodicals, 634-646.
[5] Bhouri, M, Bonnet, M., and Mboup, M., (2005). A new QR decomposition-based block adaptive algorithm, IEEE, 1497-1500.
[6] Vijaykuma, V. R.,Vanathi,P. T., and Kanagasapabathy, P. (2007).Modified adaptive filtering algorithm for noise cancellation in speech signal, IEEE,No.2(74),17-20.
[7] Harteneck, M., McWhirter, J. G., Proudler, I. K., andSewart, R.W., (1999). Algorithmically engineered fast multichannel adaptive filter based on QR-RLS.IEEE Pro.Vis.Image signal Processing, Vol.146, No.1., 7-13.
[8] Yu, D. L., Gommm, J. B., and Williams, D., (1997). A recursive orthogonal least squares algorithm for training RBF networks, Neural process lett.5,pp167-176,1997.
[9] Alexander, S. T., and Ghirnikar, A. L.,(1993). A method for recursive least squares filtering based upon an inverse QR decomposition. IEEE transaction on signal processing, 41: 20-30.
[10] Ahmad, N.A. and Okwonu, F. Z., (2010). Techniques and analysis of adaptive least squares problem. Global Journal of Pure and Applied Mathematics,6:53-62.