System Identification Based on QR-Decomposition

J.Mehena*

Electronics & Telecommunication Engineering
Dhaneswar Rath Institute of Engineering and Management Studies,
Tangi, Cuttack, Orissa – 754022, India
E-mail: j_mehena@rediffmail.com
*Corresponding author

L.Swain

Dhaneswar Rath Institute of Engineering and Management Studies,
Tangi, Cuttack, Orissa – 754022, India
E-mail: lopamudra.swain@driems.ac.in

G.Patnaik

Dhaneswar Rath Institute of Engineering and Management Studies,
Tangi, Cuttack, Orissa – 754022, India
E-mail: gayatripattnaik00@gmail.com

Abstract: This paper presents a unified derivation of rotation-based recursive least squares (RLS) algorithms based on QR decomposition. They solve the adaptive least squares problems of the linear combiner without a desired signal, the single and multichannel linear prediction and transversal filtering. All algorithms derived in this paper are based on Givens rotations. They offer superior numerical properties as shown by computer simulations. They are computationally efficient and highly concurrent. Parallel implementation of the resulting methods in the form of systolic array and its application to system identification is discussed in this paper.

Keywords: QR-decomposition, RLS, Systolic array, System identification.

References to this paper should be as follow : Mehena, J., Swain, L., Patnaik., G.(2013**)** "System Identification Based on QR-Decomposition", *Int. J. Intelligent Computing and Applied Sciences*, Vol. 1, Issue. 1, pp 31-40

1 Introduction

ADAPTIVE least squares (LS) problems arise in many disciplines of science and engineering. Typical applications are system identification, spectrum estimation, speech processing, echo cancellation, channel equalization, line enhancing, and beam forming [1]. Two different classes of methods have been developed for solving these problems, the least mean square (LMS) algorithm based on the gradient descent technique and the recursive least squares (RLS) algorithm using the exponentially weighted LS criterion[7]. The LMS algorithm is an approximation method. It is computationally simple but has relatively slow rate of convergence and is sensitive to the eigen value spread of the input correlation matrix. The RLS algorithm provides an exact solution of the LS problem at each time step. It offers better convergence behavior and parameter tracking capability at the expense of a higher computational

complexity. In adaptive filtering, however, the difference in computational complexity between the LMS and the RLS algorithms is reduced significantly due to the development of the so called fast algorithms. The RLS approach is considered here.

Solving LS problems using Givens rotations [1], [2] has been known for decades. A well known approach is to compute the QR decomposition of the input data matrix and then the LS weight vector by back substitution. In adaptive applications, the QR decomposition is updated by the Givens rotations. This result in a rotation based QR-RLS algorithm. Gentleman and Kung [3] have proposed a systolic array consisting of a triangular and a linear section to implement the algorithm. McWhirter [4] has simplified the Structure by eliminating the linear array section for applications in which only the LS estimation error is required. This RLS algorithm does not impose any restriction on the input data structure. As a consequence of this generality, the computational complexity is proportional to m^2 where m is the order of the problem.

In linear prediction and adaptive filtering applications [8], time series input data are involved. By exploiting the shifting property, a variety of fast RLS algorithms of complexity m have been derived. The most known algorithm is the Kalman algorithm. Recently there has been an increasing interest in rotation based fast RLS algorithms due to the numerical robustness. All rotation-based RLS algorithms [9] use, however, different notations and derivation techniques. Hence it is quite difficult to understand their relationships and to survey the developments in this growing area. The primary aim is to present a unified and simplified derivation of certain rotation based algorithms .We use a single matrix identity to derive the QR-RLS, and the Extended QR-RLS algorithm for solving different LS problems. Because of consistent notations and a unified approach, the relationship between distinct algorithms becomes obvious.

The QR-RLS algorithm is computationally inefficient because the data throughput of the linear section is lower than that of the triangular section. A preferable approach is to modify the QR-RLS algorithm so as to avoid the need for the cumbersome method of back substitution. This should be possible in light of what we know about the extended Square-root information filter, which computes the state estimation directly from the postarray without invoking back substitution. The modified form of QR-RLS algorithm derived in this way is called the *Extended QR-RLS algorithm*. The issues of parallel implementation by the use of Systolic array and its application to System Identification are discussed in this paper.

The paper is organized as follows. Section 2 is concerned with the elementary linear LS problems, the linear combiner and also derivation of RLS algorithm based on QR-decomposition is illustrate in this section. Section 3 is devoted to systolic array implementation of this algorithm. In section 4 System Identification is discussed. In section 5 discusses the simulation result of the plant to be identified. Section 6 concludes the paper.

2. The Linear Combiner

2.1 The QR-RLS Algorithm

The QR-RLS algorithm, or more precisely, the QR decomposition-based RLS algorithm, derives its name from the fact that the computation of the least-squares weight vector in a finite-duration impulse response filter implementation of the adaptive filtering algorithm is accomplished by working directly with the incoming data matrix via the QR decomposition rather than working with the (time-averaged) correlation matrix of the input data as in the standard RLS algorithm [3], [4],[7]. Accordingly, the QR-RLS algorithm is numerically more stable than the standard RLS algorithm.

Assuming the use of prewindowing on the input data, the data matrix is defined by

$\mathbf{A}^{H}(n) = [\ \mathbf{u}(1),\ \mathbf{u}(2), \ldots.., \mathbf{u}(M), \ldots\ldots., \mathbf{u}(n)]$

$$= \begin{pmatrix} u(1) & u(2) & \bullet\bullet\bullet & u(M) & \bullet\bullet\bullet & u(n) \\ 0 & u(1) & \bullet\bullet\bullet & u(M-1) & \bullet\bullet\bullet & u(n-1) \\ \bullet & \bullet & & \bullet & & \bullet \\ \bullet & \bullet & & \bullet & & \bullet \\ \bullet & \bullet & & \bullet & & \bullet \\ 0 & 0 & \bullet\bullet\bullet & u(1) & \bullet\bullet\bullet & u(n-M+1) \end{pmatrix}$$

Where M is the number of FIR filter coefficients. Correspondingly, the correlation matrix of the input data is defined by

$$\boldsymbol{\Phi}(n) = \sum_{i=1}^{n} \lambda \tag{1}$$

$$= \mathbf{A}^{H}(n)\boldsymbol{\Lambda}(n)\mathbf{A}(n)$$

The matrix $\boldsymbol{\Lambda}(n)$ is called the exponential weighting matrix, defined by

$\boldsymbol{\Lambda}(n) = \text{diag}[\ \lambda^{n-1}, \lambda^{n-2}, \ldots\ldots 1\ ]$ (2)

Where λ is the exponential weighting factor**.**

The matrix $\mathbf{P}(n)$ used in deriving the RLS algorithm is defined as the inverse of the correlation matrix.

$$\mathbf{P}(n) = \boldsymbol{\Phi}^{-1}(n) \tag{3}$$

According to the normal equations the least squares estimate of the tap weight vector $\hat{\mathbf{w}}(n)$ is defined by

$$\boldsymbol{\Phi}(n)\hat{\mathbf{w}}(n)=\mathbf{z}(n) \tag{4}$$

where $\mathbf{z}(n)$ is the cross-correlation vector between the desired response d(n) and input data vector $\mathbf{u}(n)$. Let $\boldsymbol{\Phi}(n)$ be expressed in its factored form:

$$\boldsymbol{\Phi}(n)=\boldsymbol{\Phi}^{1/2}(n)\boldsymbol{\Phi}^{H/2}(n) \tag{5}$$

Premultiplying both sides of (4) by the square root $\boldsymbol{\Phi}^{-1/2}(n)$. We can introduce a new vector variable defined by

$$\mathbf{p}(n) = \boldsymbol{\Phi}^{H/2}(n)\hat{\mathbf{w}}(n) = \boldsymbol{\Phi}^{-1/2}(n)\mathbf{z}(n) \tag{6}$$

We can translate the square-root information filtering algorithm into the corresponding prearray-to-postarray transformation for the QR-RLS algorithm as follows (after cancelation of common terms):

$$\begin{pmatrix} \lambda^{1/2}\boldsymbol{\Phi}^{1/2}(n-1) & \mathbf{u}(n) \\ \lambda^{1/2}\mathbf{p}^{H}(n-1) & d(n) \\ \mathbf{0}^{T} & 1 \end{pmatrix} \boldsymbol{\Theta}(n) = \begin{pmatrix} \boldsymbol{\Phi}^{1/2}(n) & \mathbf{0} \\ \mathbf{p}^{H}(n) & \xi(n)\gamma^{1/2}(n) \\ \mathbf{u}^{H}(n)\,\boldsymbol{\Phi}^{-H/2}(n) & \gamma^{1/2}(n) \end{pmatrix} \tag{7}$$

Where $\boldsymbol{\Theta}(n)$ is a unitary rotation that operates on the elements of the input data vector $\mathbf{u}(n)$ in the prearray, annihilating them one by one so as to produce a block zero entry in the top block row of the postarray . Naturally, the lower triangular structure of the square root of the correlation matrix, namely, $\boldsymbol{\Phi}^{1/2}$, is preserved in its exact form before and after the transformation. This is indeed the very essence of the QR decomposition for RLS estimation, hence the name "QR-RLS algorithm".

Having computed the updated block values $\mathbf{\Phi}^{1/2}(n)$ and $\mathbf{p}^{H}(n)$, we can then solve for the least-squares weight vector $\hat{\mathbf{w}}(n)$ by using (6).

$$\hat{\mathbf{w}}^{H}(n=\mathbf{p}^{H}(n)\mathbf{\Phi}^{-1/2}(n) \quad (8)$$

To initialize the algorithm we set

$$\mathbf{\Phi}^{1/2}(0) = \sqrt{\delta}\mathbf{I}$$

With δ being a nonnegative constant and p(0) = **0**, Summary of the algorithm is given below

Summary of the QR-RLS Algorithm

Inputs:

Input signal vector = { $\mathbf{u}(1)$, $\mathbf{u}(2)$, ….., $\mathbf{u}(n)$ }

Desired response = { d(1),d(2),……..,d(n) }

Known parameter:

Exponential weighting factor = λ

Initial conditions:

$$\mathbf{\Phi}^{1/2}(0) = \sqrt{\delta}\mathbf{I} \quad (\delta \geq 0)$$

$$\mathbf{p}(0) = \mathbf{0}$$

For n = 1, 2, ……, compute

$$\begin{pmatrix} \lambda^{1/2}\mathbf{\Phi}^{1/2}(n-1) & \mathbf{u}(n) \\ \lambda^{1/2}\mathbf{p}^{H}(n-1) & d(n) \\ \mathbf{0}^{T} & 1 \end{pmatrix} \mathbf{\Theta}(n) = \begin{pmatrix} \mathbf{\Phi}^{1/2}(n) & \mathbf{0} \\ \mathbf{p}^{H}(n) & \xi(n)\gamma^{1/2}(n) \\ \mathbf{u}^{H}(n)\,\mathbf{\Phi}^{-H/2}(n) & \gamma^{1/2}(n) \end{pmatrix} \quad (9)$$

$$\hat{\mathbf{w}}^{H}(n) = \mathbf{p}^{H}(n)\mathbf{\Phi}^{-1/2}(n)$$

2.2 Extended QR-RLS Algorithm

The QR-RLS algorithm is suitable for adaptive filtering applications such as adaptive beam forming and acoustic echo cancelation, where the primary function is to compute the a posteriori estimation error without explicit knowledge of the least-squares weight vector. However, in other adaptive filtering applications such as system identification and spectrum analysis, Knowledge of the weight vector on a continuing basis is a necessary requirement. Gentleman and Kung have proposed a systolic array consisting of a triangular and a linear section to implement the algorithm. The use of such a procedure is computationally inefficient because the data throughput of the linear section is lower than that of the triangular section. A preferable approach is to modify the QR-RLS algorithm so as to avoid the need for the cumbersome method of back substitution. This should be possible in light of what we know about the extended square-root information filter, which computes the state estimate directly from the post array without invoking back substitution. The modified form of QR-RLS algorithm derived in this way is called the *extended QR-RLS algorithm* [6].

Consider the transformation of the extended square-root information filter. We can formulate the prearray-to-postarray transformation for the extended QR-RLS algorithm by using the one to one correspondences that exit between the Kalman variables and the RLS variables, and so write the following [5]:

$$\begin{pmatrix} \lambda^{1/2}\boldsymbol{\Phi}^{1/2}(n-1) & \mathbf{u}(n) \\ \lambda^{1/2}\,\mathbf{p}^{H}(n-1) & d(n) \\ \mathbf{0}^{T} & 1 \\ \lambda^{-1/2}\boldsymbol{\Phi}^{-H/2}(n-1) & \mathbf{0} \end{pmatrix} \boldsymbol{\Theta}(n) = \begin{pmatrix} \boldsymbol{\Phi}^{1/2}(n) & \mathbf{0} \\ \mathbf{p}^{H}(n) & \xi(n)\gamma^{1/2}(n) \\ \mathbf{u}^{H}(n)\,\boldsymbol{\Phi}^{-H/2}(n) & \gamma^{1/2}(n) \\ \boldsymbol{\Phi}^{-H/2}(n) & -\mathbf{k}(n)\gamma^{-1/2}(n) \end{pmatrix}$$

The updated value of the state estimate can be computed as follows:

$$\hat{\mathbf{w}}(n)= \hat{\mathbf{w}}(n-1) + (\mathbf{k}(n)\gamma^{-1/2}(n))\,(\xi(n)\gamma^{1/2}(n))^{*} \qquad (10)$$

Where the quantities $\mathbf{k}(n)\gamma^{-1/2}(n)$ and $\xi(n)\gamma^{1/2}(n)$ are read directly from the post array in equation (9).Summary of the Extended QR-RLS algorithm is shown below. Initialization of the algorithm as same as QR-RLS algorithm.

Summary of the extended QR-RLS algorithm

Inputs:

Input signal vector = { $\mathbf{u}(1), \mathbf{u}(2), \ldots, \mathbf{u}(n)$ }

Desired response = { $d(1), d(2), \ldots, d(n)$ }

Known parameter:

Exponential weighting factor = λ

Initial conditions:

$$\boldsymbol{\Phi}^{1/2}(0) = \sqrt{\delta}\mathbf{I} \quad (\delta \geq 0)$$

$$\mathbf{p}(0) = \mathbf{0}$$

For n = 1, 2, ……, compute

$$\begin{pmatrix} \lambda^{1/2}\boldsymbol{\Phi}^{1/2}(n-1) & \mathbf{u}(n) \\ \lambda^{1/2}\,\mathbf{p}^{H}(n-1) & d(n) \\ \mathbf{0}^{T} & 1 \\ \lambda^{-1/2}\boldsymbol{\Phi}^{-H/2}(n-1) & \mathbf{0} \end{pmatrix} \boldsymbol{\Theta}(n) = \begin{pmatrix} \boldsymbol{\Phi}^{1/2}(n) & \mathbf{0} \\ \mathbf{p}^{H}(n) & \xi(n)\gamma^{1/2}(n) \\ \mathbf{u}^{H}(n)\,\boldsymbol{\Phi}^{-H/2}(n) & \gamma^{1/2}(n) \\ \boldsymbol{\Phi}^{-H/2}(n) & -\mathbf{k}(n)\gamma^{-1/2}(n) \end{pmatrix}$$

$$\hat{\mathbf{w}}(n)= \hat{\mathbf{w}}(n-1) + (\mathbf{k}(n)\gamma^{-1/2}(n))\,(\xi(n)\gamma^{1/2}(n))^{*}$$

3. Systolic Array Implementation

Fig.-1 shows a systolic array structure for implementing a simplified form of the extended QR-RLS algorithm [5], [6] for the case when the weight vector $\hat{\mathbf{w}}(n)$ has three elements (i.e.,M = 3). This structure consists of two triangular sections appended to each other: The top triangular section, shown unshaded and the bottom triangular section, shown lightly shaded in figure 1. The entire systolic array is controlled by a single clock. Each section of the array consists of two types of processing cells: internal cells and boundary cells. The specific arithmetic functions of these cells are shown in the figure 2. Each cell receives its input data from the directions indicated for one clock cycle, performs the specific arithmetic functions, and then, on the next clock cycle, deliver the resulting output values to neighboring cells as indicated. A distinctive feature of systolic arrays is that each processing cell is always kept active as data flow across the array.

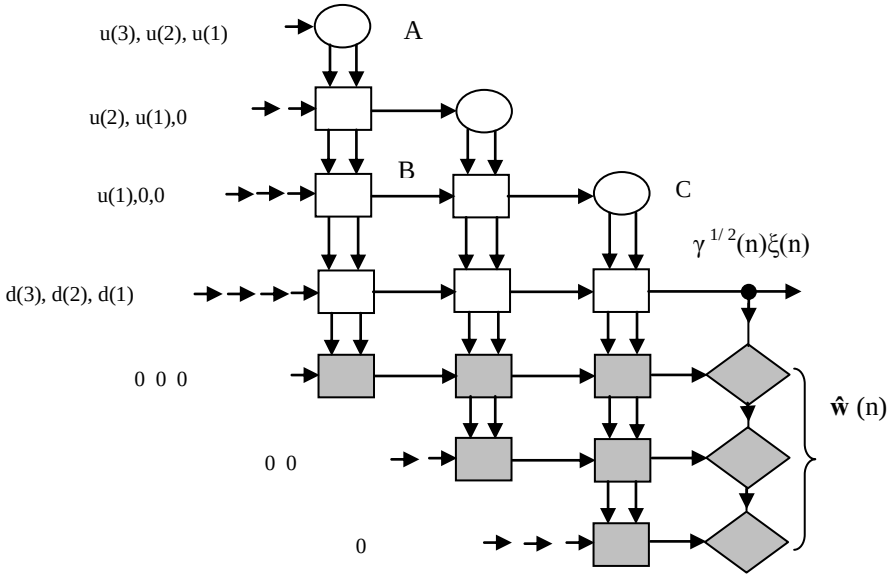

Figure 1. Systolic array implementation of the extended QR-RLS algorithm

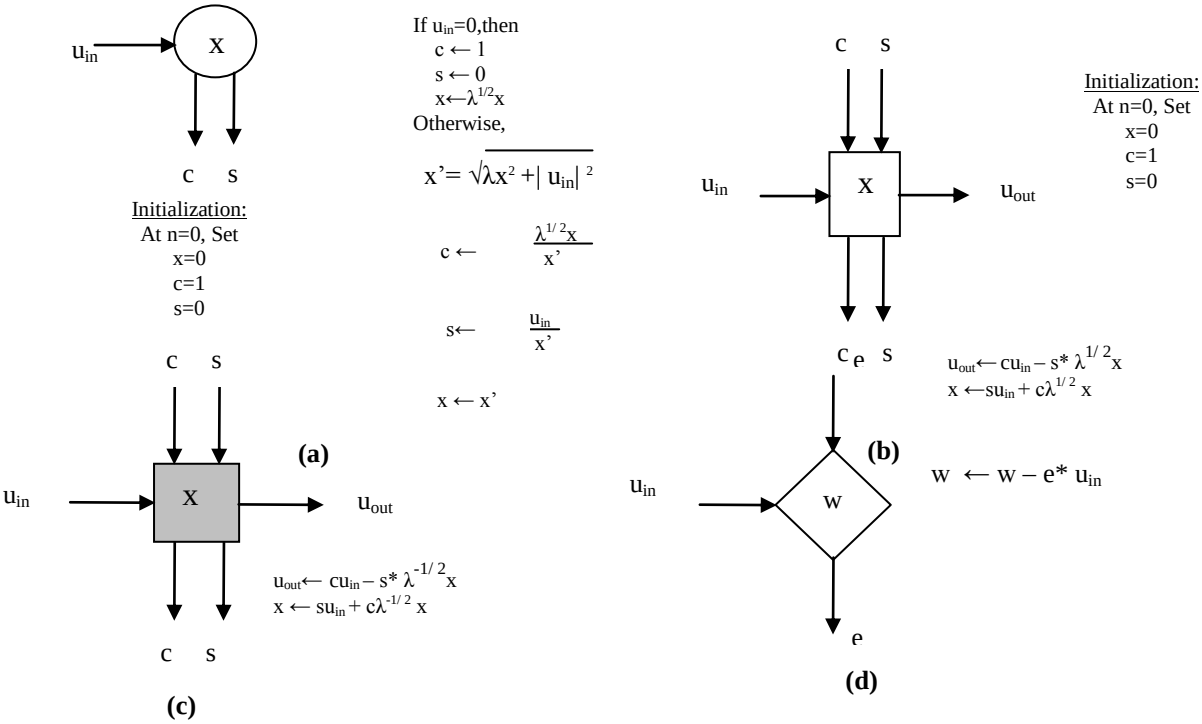

Figure 2. **Cells for the lower triangular section**
(a) boundary cell (b) internal cell.
Cells for the upper triangular section
(c) internal cell (d) boundary cell

The systolic array structure can be implemented here consists of two triangular section appended to each other:

I. The *top triangular section*, shown unshaded in figure 2. consists of its own boundary and internal cells. The computations performed by the boundary and internal cells of this section are described in fig. 2. Consider first the operation of the triangular systolic array section labeled ABC in fig. 1. Basically the internal cells perform only additions and multiplications. The boundary cells, on the other hand, are considerably more complex, in that they compute square roots and reciprocals, as described here. Each cell of the triangular systolic array section (depending on its location) stores a particular element of the lower triangular matrix $\mathbf{\Phi}^{1/2}(n)$ and are updated every clock cycle. The function of each column of processing cells in the triangular systolic array section is to rotate one column of the stored triangular matrix with a vector of data received from the left in such a way that the leading element of the received data vector is annihilated. The reduced data vector is then passed to the right on to the next column of cells. The boundary cells in each column of the section computes the pertinent rotation parameters and then passes them downward on the next clock cycle. The internal cells subsequently apply the same rotation to all other elements in the received data vector. Since a delay of one clock cycle per cell is incurred in passing the rotation parameters downward along a column, it is necessary that the input data vector enter the triangular systolic array in a *skewed order* as shown in fig. 1. for the case of M = 3. This arrangement of the input data ensures that as each column vector $\mathbf{u}(n)$ of the data matrix $\mathbf{A}^{H}(n)$ propagates through the array , it interacts with the previously stored data matrix $\mathbf{\Phi}^{1/2}(n-1)$ and thereby undergoes the sequence of Givens rotations denoted by $\mathbf{\Theta}(n)$, as required. Accordingly, all the elements of the column vector $\mathbf{u}(n)$ are annihilated, one by one, and an updated lower triangular matrix $\mathbf{\Phi}^{1/2}(n)$ is produced and stored in the process.

The systolic array operates in a highly pipelined manner, whereby, as input data vectors enter the array from the left, we find that in effect each such vector defines a processing wave front that moves across the array. It should therefore be appreciated that, on any particular clock cycle, elements of the pertinent lower triangular matrix $\mathbf{\Phi}^{1/2}(n)$ only exist along the corresponding wave-front. At the same time that the orthogonal triangularization process is being performed by the triangular systolic array section labeled ABC in fig. 1, the row vector $\mathbf{p}^{H}(n)$ is computed by the appended bottom row of internal cells. Overall the function of the top triangular section is to compute the quantity $\xi(n)\gamma^{1/2}(n)$.

II. The *bottom triangular section*, shown lightly shaded in fig. 2 consists of its own boundary and internal cells. The computations performed by the cells are described as shown in Fig. 2. This second triangular section rotates stored values of $\lambda^{-1/2}\mathbf{\Phi}^{-H/2}(n-1)$ and an externally applied vectors of zeros, yielding the updated $\mathbf{\Phi}^{-H/2}(n)$ and the desired quantity $\mathbf{k}(n)\gamma^{-1/2}(n)$.The correction needed to update the weight vector is obtained simply by complex conjugating the top triangular section's output $\xi(n)\gamma^{1/2}(n)$ and then multiplying it by the bottom triangular section's output $\mathbf{k}(n)\gamma^{-1/2}(n)$, in accordance with equation (10). These computations are performed in the diamond-shaped boundary cells of the bottom triangular section.

4. System Identification

System identification is the experimental approach to the modeling of a process or a plant. It involves the following steps: experimental planning, the selection of a model structure. Parameter estimation and model validation. The procedure of system identification, as pursued in practice, is iterative in nature in that we may have to go back and fourth between these steps until a satisfactory model is built. Here we discuss briefly the idea of extended QR-RLS algorithm for estimating the parameters of an unknown plant modeled as a transversal filter. Suppose we have an unknown dynamic plant that is linear and time varying [8],[9]. The plant is characterized by a real valued set of discrete-time measurements that describe the variation of the plant output in response to a known stationary input. The requirement is to develop an *on-line transversal filter model* for this plant as illustrated in fig..The model consists of a finite number of unit-delay elements and a corresponding set of adjustable parameters (tap weight).

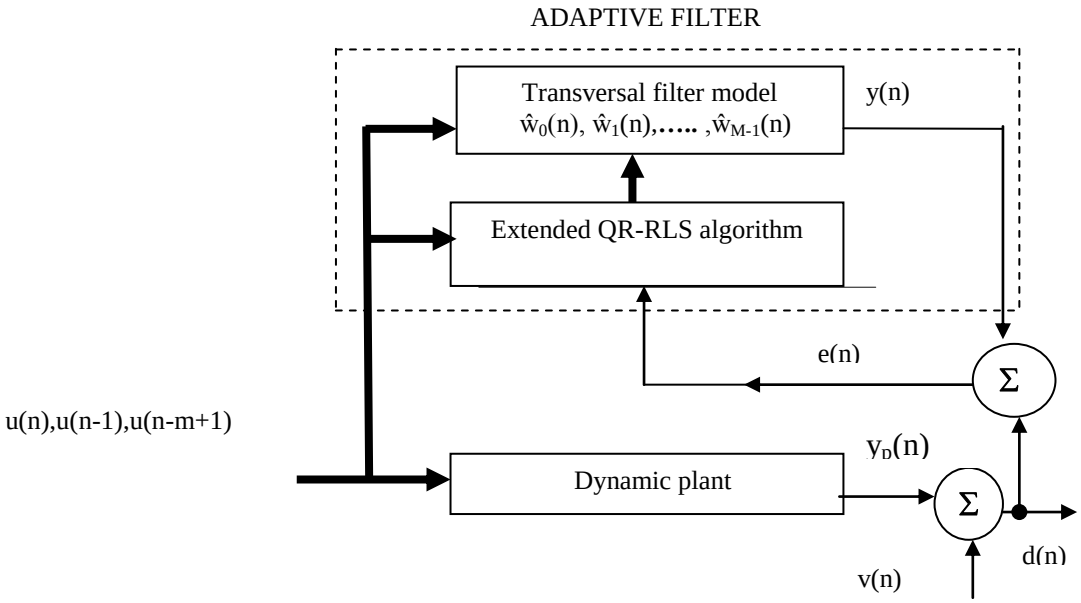

Figure 3 System identification

The input signal at time n be denoted by the set of samples such as u(n), u(n-1),……..,u(n-M+1), where M is the number of adjustable parameters in the model. This input signal is applied simultaneously to the plant and the model. Let their respective outputs be denoted by $y_p(n)$ and y(n). The plant output $y_p(n)$ and with white gaussian noise v(n) serves the purpose of a desired response d(n) .The extended QR-RLS algorithm employed to adjust the model parameters. The model output is given by

$$y(n) = \sum_{k=0}^{M-1} \hat{w}_k(n)u(n-k) \tag{11}$$

Where $w_0(n)$, $w_1(n)$, ……, and $w_{M-1}(n)$ are the estimated model parameters. The model output y (n) is compared with the desired output d(n). The difference between them, d(n)-y(n), defines the modeling (estimation) error.

$$e(n) = d(n) - y(n) \tag{12}$$

Where e(n) is the estimation error. When the plant is time varying, the plant output is nonstationary, and so is the desired response presented to the extended QR-RLS algorithm. In such a situation, the extended QR-RLS algorithm has the task of not only keeping the modeling error small but also continually tracking the time variations in the dynamics of the plant.

4. Simulation Results

The plant to be identified is governed by the difference equation $y_p(n) = 0.747y_p(n-1) – 0.272y_p(n-2) + 0.187u(n) + 0.149u(n-1) + 0.187u(n-2)$,Where the input u(n) to be generated using the first-order autoregressive model is given by [1], [7]: $u(n) = v(n) + 0.9u(n-1)$ where v(n) is drawn from a white-noise process of zero mean and variance = 0.0731.The simulation result performed for $\lambda = 0.99$ and $\delta = 0.01$ are shown in Figure 4 and Figure 5.

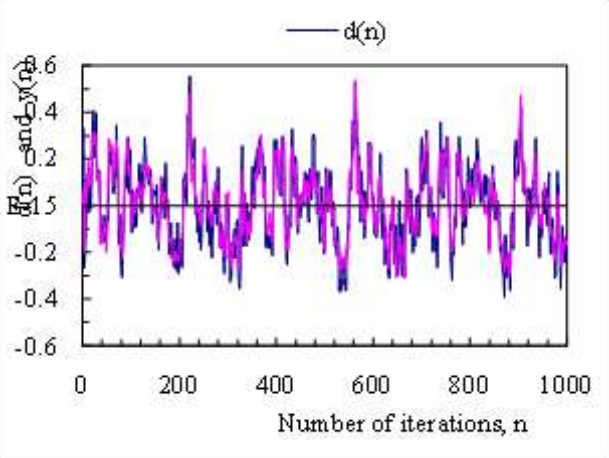

Figure 4. Response of Plant at $\lambda = 0.99$ and $\delta = 0.01$.

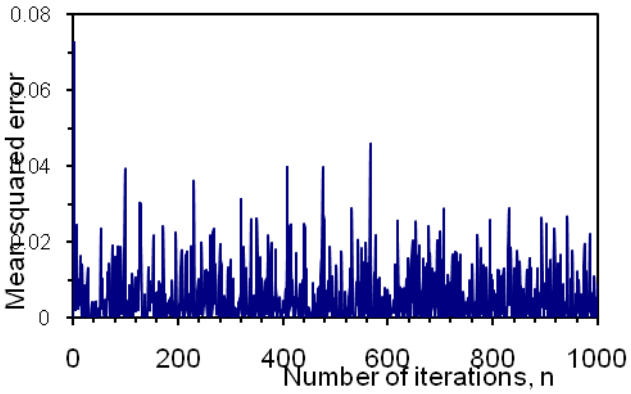

Figure 5. Mean squared error at $\lambda = 0.99$ and $\delta = 0.01$.

6. Conclusion and future scope

In this paper, we presented a unified derivation of rotation based RLS algorithm based on QR-decomposition. We use a single matrix identity to derive the Extended QR-RLS algorithm for solving different LS problems. The systolic array implementations of the QR-decomposition involved in the design of the variants of the RLS algorithm described in this paper are all based on the Givens rotation. This form of rotation provides one method for constructing the unitary rotation $\mathbf{\Theta}(n)$. The parallel implementations of this algorithm permit the computation of the least–squares weight vector in an efficient manner. Accordingly, this algorithm is well suited for application such as system identification and is discussed here, where knowledge of the weight vector is a necessary requirement. We compared rotation based extended QR-RLS algorithm with non-rotation based counterparts. The results indicate that there are no significant differences in the numerical performance for high-precision computations. If however, a low-precision arithmetic is used in order to reduce the implementation costs, the nonrotation-based algorithms may produce unexpected results or break down. The reason is that some of the internal variables are pushed out of their infinite precision domains due to rounding errors. This can result in operations like division by zero. The algorithm derived in this paper avoids these problems.

Finally, we mention that the algorithm derived here is highly concurrent and has fast rate of convergence, as shown by simulation. We described Systolic implementation and parallel array

architecture. Moreover, we presented system identification that can be easily integrated into the systolic array architecture for computing LS errors. As a future scope, the systolic implementation can be extended to the House- holder transformation, which will improve the numerical stability of the algorithm.

References

[1] S. Haykin, *Adaptive Filter Theory*. Prentice Hall, Inc.,1996.

[2] S. Haykin, *Modern Filters*. Maxwell Macmillan International Editions, 1990.

[3] W.M. Gentleman and H.T. Kung, "Matrix triangularization by systolic arrays,"*Proc. SPIE* , vol. 298, Real Time Signal Processing IV, pp.298-303, 2006.

[4] J. G. McWhirter, "Recursive least-squares minimization using a systolic array," *Proc. SPIE*, Real-Time Signal Processing VI, vol. 431, pp. 105-112, 1983.

[5] Sayed, A. H and T. Kailath, "A state space approach to adaptive RLS filtering," IEEE Trans. Signal Processing, pp 18-60, July 1994.

[6] Bin Yang and Johann F. Bohme, "Rotation-Based RLS Algorithms: Unified Derivations, Numerical Properties, and Parallel Implementation," IEEE Trans. Signal Processing, vol. 40, no.5, pp. 1151-1167, May 2006.

[7] S. Thomas Alexander and Avinash L. Ghirnikar, " A method for Recursive Least Squares Filtering Based an Inverse QR Decomposition," IEEE Trans. Signal Processing, vol. 41, no. 1, pp. 20-30, January 2008.

[8] Asriel U.Levin and Kumpati S. Narendra, " Control of Nonlinear Dynamical Systems Using Neural Networks Observability, Identification, and Control," IEEE Trans. Neural Network, vol. 7, no. 1, pp. 30-42, January 1996.

[9] Kumpathi S. Narendra and Kannan Parthasarathy, " Identification and Control of Dynamical Systems Using Neural Networks," vol. 1, no. 1, pp. 4-27, March 2009.

Biographical Notes : Jibanananda Mehena is presently working as Associate Professor and Head in the department of Electronics and Telecommunication Engg., DRIEMS, Tangi, Cuttack. He has 15 years of teaching experience. He received his M. Tech in Electronics Engg. From Visvesvaraya National Institute of Technology (VNIT) , Nagpur and continuing his Ph.D in the area of Digital Image Processing. He has published many research papers in national and international journals. His areas of interest include Digital Image Processing, VLSI Design, Digital Signal Processing, Microprocessor & Microcontroller. He has authored five books to his credit.

Lopamudra Swain is presently working as Assistant Professor in the department of Electronics and Telecommunication Engg., DRIEMS, Tangi, Cuttack. She has 7 years of teaching experience. She received her M. Tech in Telecommunication system Engg. from SOA University Bhubaneswar. She has published one research paper in the international journals. Her areas of interest includes Digital signal processing, Wireless Communication and Optical fiber communication.

Gayatri Pattnaik is presently working as Assistant Professor in the department of Electronics and Telecommunication Engg., DRIEMS, Tangi, Cuttack. She has 6 years of teaching experience. She received her M. Tech in Electronics and Telecommunication Engg. from BPUT,Odisha. She has published one research paper in the international journals. Her areas of interest includes communication technique, Image and signal processing.