Title	Approximate QR-based algorithms for recursive nonlinear least squares estiamtion
Author(s)	Chan, SC; Zhou, Y; Lau, WY
Citation	Proceedings - Ieee International Symposium On Circuits And Systems, 2005, p. 4333-4336
Issued Date	2005
URL	http://hdl.handle.net/10722/45784
Rights	Creative Commons: Attribution 3.0 Hong Kong License

Approximate QR-Based Algorithms for Recursive Nonlinear Least Squares Estiamtion

S. C. Chan, Y. Zhou and W. Y. Lau
Department of Electrical and Electronic Engineering
The University of Hong Kong, Pokfulam Road, Hong Kong
E-mail: {scchan, yizhou, wylau@eee.hku.hk}

***Abstract*—This paper proposes new approximate QR-based algorithms for recursive nonlinear least squares (NLS) estimation. Two QR decomposition-based recursive algorithms are introduced based on the classical Gauss-Newton (GN) and Levenberg-Marquardt (LM) algorithms in nonlinear unconstrained optimization or least squares problems. Instead of using the matrix inversion formula, recursive QR decomposition is employed, which is known to be numerically more stable in finite wordlength implementation. A family of *p*-A-QR-LS algorithms is then proposed to solve the LS problem resulting from the linearization of the NLS problem. It achieves different complexity-performance tradeoffs by retaining different number of diagonal plus off-diagonals (denoted by an integer *p*) of the triangular factor of the augmented data matrix. Simulation results on identifying a nonlinear perceptron are provided to illustrate the principle of the new algorithms.**

I. INTRODUCTION

Recursive least squares parameter estimation is frequently encountered in digital signal processing, communications and control applications. The problem of linear least squares (LS) has been extensively studied and many efficient algorithms such as the least mean squares (LMS) and the recursive least squares (RLS) algorithms have been proposed. Interested readers are referred to the references in [1,2,6] for more details. On the other hand, the training of neural networks [9,11,12] nonlinear system identification [10], and many other real-time applications involving nonlinear models usually require the recursive parameter estimation of the system parameters using the least squares criterion. A commonly used method, which is based on the Levenberg Marquardt (LM) method [8] and the matrix inversion formula-based RLS implementation, is the Recursive Maximum Likelihood algorithm in [10]. The recursive LM, recursive Gauss-Newton (GN) and recursive steepest descent (SD) have also been proposed for neural network training in [9,11,12], respectively. In this paper, we develop a QR decomposition-based framework for implementing the recursive GN and LM algorithms. It is well known that QR decomposition-based method for solving the linear LS problem is numerically more stable than using the matrix inversion formula under finite wordlength implementation. Also, efficient hardware implementation in form of systolic arrays and cordic processors are readily available. Therefore, it is highly desirable to develop a QR-based framework for solving the recursive nonlinear LS (NLS) problem. Starting from the conventional GN or LM algorithms for solving the nonlinear LS problem [8], we shall systematically show that the recursive GN and LM algorithms can be formulated as a time-recursive LS problem and solved using the recursive QR method in adaptive filtering. Further, a new family of suboptimal algorithms called the *p*-A-QR-LS algorithm, which spans both the LMS and QR-RLS algorithms, is proposed to achieve different complexity-performance tradeoffs in solving the linear LS problem resulting from linearizing the recursive NLS problem. Simulation results for identifying the parameters of a perceptron are used to demonstrate the usefulness of the proposed algorithms. The paper is organized as follows: Sections II and III are devoted to the derivation of the QR-based recursive GN and LM algorithms. The *p*-A-QR-NLS algorithms are introduced in Section IV. Simulation results are given in Section V and conclusions are drawn in Section VI.

II. THE QR-BASED RECURSIVE GAUSS-NEWTON (RGN) ALGORITHM

Consider the estimation of a parameter vector $\boldsymbol{\theta}^\circ$ from a time series: $d(j) = g_{\theta^\circ}(j) + \eta(j)$, j=0,…,n, with $g_{\theta^\circ}(j)$ being a known function of $\boldsymbol{\theta}^\circ$ and $\eta(j)$ being a zero mean, additive white Gaussian noise sequence. In least squares estimation, the time averaged squares error $e_{\boldsymbol{\theta}}(j) = d(j) - g_{\theta}(j)$ is minimized:

$$\xi(n) = \sum_{j=0}^{n} \lambda^{n-j} \left| e_{\theta}(j) \right|^2 \tag{1}$$

where λ is a constant forgetting factor with a value between 0 and 1. This nonlinear least squares problem can be solved by a number of methods such as the steepest descent, Gauss-Newton and Levenberg-Marquardt algorithms. Using the conventional Gauss-Newton algorithm, $\boldsymbol{\theta}(n)$, the locally optimal estimate can be iteratively computed as follows:

$$\boldsymbol{\theta}_{l+1}(n) = \boldsymbol{\theta}_l(n) - (\boldsymbol{J}^T(\boldsymbol{\theta}_l(n))\boldsymbol{J}(\boldsymbol{\theta}_l(n)))^{-1}\boldsymbol{J}^T(\boldsymbol{\theta}_l(n))\boldsymbol{R}(\boldsymbol{\theta}_l(n)) \tag{2}$$

where $[\boldsymbol{J}(\boldsymbol{\theta})]_{jk} = \partial g_{\boldsymbol{\theta}}(j)/\partial\theta_k$, $\boldsymbol{R}(\boldsymbol{\theta}) = [0,0,\cdots,e_{\boldsymbol{\theta}}(n)]^T$, θ_k is the k-th element of $\boldsymbol{\theta}$, l is the iteration number. In real-time application, such iteration is usually computationally prohibitive to implement and it is desirable to embed this iteration into the time recursion. More precisely, the estimate at the n-th time instant is:

$$\boldsymbol{\theta}(n+1) = \boldsymbol{\theta}(n) - (\boldsymbol{J}^T(\boldsymbol{\theta}(n))\boldsymbol{J}(\boldsymbol{\theta}(n)))^{-1}\boldsymbol{J}^T(\boldsymbol{\theta}(n))\boldsymbol{R}(\boldsymbol{\theta}(n)). \tag{3}$$

In other words, only one Gauss-Newton iteration is performed at each time instant and hopefully when more observations are available as n increases, the parameter vector $\boldsymbol{\theta}^\circ$ can be estimated. The term $\Delta\boldsymbol{\theta}(n) = (\boldsymbol{J}^T(\boldsymbol{\theta}(n))\boldsymbol{J}(\boldsymbol{\theta}(n)))^{-1}\boldsymbol{J}^T(\boldsymbol{\theta}(n))\boldsymbol{R}(\boldsymbol{\theta}(n))$ can be viewed

This work is supported by the Hong Kong research grant council (RGC)

as the solution of the following linear LS problem resulting from the linearization of the nonlinear objective function at $\boldsymbol{\theta}(n)$:

$$\Delta\boldsymbol{\theta}(n) = \arg\min_{\boldsymbol{x}} \| \boldsymbol{J}(\boldsymbol{\theta}(n))\boldsymbol{x} - \boldsymbol{R}(\boldsymbol{\theta}(n)) \|_2^2 . \quad (4)$$

Since the linearization is done around the current estimate $\boldsymbol{\theta}(n)$, the whole Jacobian matrix $\boldsymbol{J}(\boldsymbol{\theta}(n))$ and the residual have to be evaluated again and their sizes grow linearly with n. Because of this reason, a further simplification is necessary. Suppose that the approximated Jacobian matrix at the (n-1)-th time instant is $\boldsymbol{J}(n-1)$. Then $\boldsymbol{J}(n)$ can be approximated by

$$\boldsymbol{J}(n) = \begin{bmatrix} \lambda \boldsymbol{J}(n-1) \\ \boldsymbol{\varphi}^T(n) \end{bmatrix} \quad (5)$$

where $\boldsymbol{\varphi}(n)$ is a column vector with its k-th element given by $[\boldsymbol{\varphi}(n)]_k = \partial g_{\boldsymbol{\theta}}(n)/\partial\theta_k \mid_{\boldsymbol{\theta}(n)}$. In principle, we are assuming that the change in $\boldsymbol{\theta}(j)$ is small enough that the previous values can be reused in the calculation. As a result, only the partial derivatives of $\partial g_{\boldsymbol{\theta}}(n)/\partial\theta_k$ at the current estimate need to be calculated. Similarly, the residual vector $\boldsymbol{R}(\boldsymbol{\theta}(n))$ can be recursively approximated as

$$\boldsymbol{R}(n) = \begin{bmatrix} \boldsymbol{0} \\ e_{\boldsymbol{\theta}(n-1)}(n) \end{bmatrix} . \quad (6)$$

Accordingly, $\Delta\boldsymbol{\theta}(n)$ can be approximated by solving the following simplified LS problem:

$$\Delta\hat{\boldsymbol{\theta}}(n) = \arg\min_{\boldsymbol{x}} \| \boldsymbol{J}(n)\cdot\boldsymbol{x} - \boldsymbol{R}(n) \|_2^2 . \quad (7)$$

This is recognized as the conventional recursive LS estimation problem, which can be computed using the QR decomposition method in Table 1. To limit the effect of additive noise, a stepsize parameter $0 < \mu(n) \le 1$ can be introduced in the parameter updates:

$$\boldsymbol{\theta}(n+1) = \boldsymbol{\theta}(n) - \mu(n)\cdot\Delta\hat{\boldsymbol{\theta}}(n) . \quad (8)$$

TABLE I. QR-RLS ALGORITHM

1. Given the augmented data matrix

$\boldsymbol{D}(n-1) = \boldsymbol{W}(n-1)[\boldsymbol{J}(n-1) \quad \boldsymbol{R}(n-1)]$, where

$\boldsymbol{W}(n) = diag(\sqrt{\lambda}^n, \sqrt{\lambda}^{n-1}, \cdots, \sqrt{\lambda}, 1)$, and its QRD at time ($n$-1):

$$\boldsymbol{D}^*(n-1) = \boldsymbol{Q}(n-1)\boldsymbol{J}(n-1) = \begin{bmatrix} \Re(n-1) & \hat{\boldsymbol{R}}(n-1) \\ 0 & \boldsymbol{c_{n-1}} \end{bmatrix}$$

where $\boldsymbol{Q}(n-1)$ and $\Re(n-1)$ are unitary and upper triangular matrices, respectively.

2. (**QRD**) Form the new augmented data matrix

$$\boldsymbol{D}(n) = \boldsymbol{W}(n)[\boldsymbol{J}(n) \quad \boldsymbol{R}(n)] = \begin{bmatrix} \sqrt{\lambda}\boldsymbol{J}(n-1) & \boldsymbol{0} \\ \psi^T(n) \end{bmatrix}$$

where $\psi(n) = \begin{bmatrix} \boldsymbol{\varphi}^T(n) & e_{\boldsymbol{\theta}(n-1)}(n) \end{bmatrix}^T$. Get the new QRD by Givens rotations or Householder reflections as

$$\boldsymbol{Q}^{(N)}(n)\cdots\boldsymbol{Q}^{(1)}(n)\boldsymbol{Q}'(n)\boldsymbol{J}(n) = \begin{bmatrix} \Re(n) & \hat{\boldsymbol{R}}(n) \\ 0 & \boldsymbol{c_n} \\ 0^T & c(n) \end{bmatrix}$$

3. (**Back-solving**) Solve the triangular system $\Re(n)(\Delta\hat{\boldsymbol{\theta}}(n)) = \hat{\boldsymbol{R}}(n)$ for the estimate $\Delta\hat{\boldsymbol{\theta}}(n)$ at time n by back-substitution:

$$\Delta\hat{\theta}_N(n) = [r_{N,N+1}(n)]/r_{N,N}(n)$$

$$\Delta\hat{\theta}_i(n) = [r_{i,N+1}(n) - \sum_{j=i+1}^{N} r_{i,j}(n)\Delta\hat{\theta}_j(n)]/r_{i,i}(n), \quad i = N-1,\cdots,1$$

where $r_{i,j}$ and $r_{i,N+1}$ are the corresponding elements in $\Re(n)$ and $\hat{\boldsymbol{R}}(n)$. $\Delta\hat{\theta}_i(n)$ is the i-*th* element of $\Delta\hat{\boldsymbol{\theta}}(n)$.

III. THE QR-BASED RECURSIVE LEVENBERG-MARQUARDT (RLM) ALGORITHM

If the LM algorithm is used to solve (1), then (2) is modified to

$$\boldsymbol{\theta}_{l+1}(n) = \boldsymbol{\theta}_l(n) - (\boldsymbol{J}^T(\boldsymbol{\theta}_l(n))\boldsymbol{J}(\boldsymbol{\theta}_l(n)) + \mu\boldsymbol{I})^{-1}\boldsymbol{J}^T(\boldsymbol{\theta}_l(n))\boldsymbol{R}(\boldsymbol{\theta}_l(n)) , \quad (9)$$

where the term $\mu\boldsymbol{I}$ is included to prevent the matrix from being singular and μ is the regularization parameter. Following a similar simplification for the recursive GN algorithm, one gets the following equation for the recursive LM algorithm:

$$\boldsymbol{\theta}(n+1) = \boldsymbol{\theta}(n) - \Delta\boldsymbol{\theta}(n) , \quad (10)$$

where $\Delta\boldsymbol{\theta}(n) = (\boldsymbol{J}^T(\boldsymbol{\theta}(n))\boldsymbol{J}(\boldsymbol{\theta}(n)) + \mu\boldsymbol{I})^{-1}\cdot\boldsymbol{J}^T(\boldsymbol{\theta}(n))\boldsymbol{R}(\boldsymbol{\theta}(n))$. It can be shown easily that $\Delta\boldsymbol{\theta}(n)$ is the solution of the following LS problem

$$\Delta\boldsymbol{\theta}(n) = \arg\min_{\boldsymbol{x}} \| \boldsymbol{A}(\boldsymbol{\theta}(n))\cdot\boldsymbol{x} - \boldsymbol{b}(\boldsymbol{\theta}(n)) \|_2^2 , \quad (11)$$

where

$$\boldsymbol{A}(\boldsymbol{\theta}(n)) = \begin{bmatrix} \boldsymbol{J}(\boldsymbol{\theta}(n)) \\ \sqrt{\mu}\boldsymbol{I} \end{bmatrix} \text{ and } \boldsymbol{b}(\boldsymbol{\theta}(n)) = \begin{bmatrix} \boldsymbol{R}(\boldsymbol{\theta}(n)) \\ \boldsymbol{0} \end{bmatrix} . \quad (12)$$

By reusing the partial derivatives as in the RGN algorithm, one gets the following approximation for

$$\Delta\hat{\boldsymbol{\theta}}(n) = \arg\min_{\boldsymbol{x}} \| \boldsymbol{A}(n)\cdot\boldsymbol{x} - \boldsymbol{b}(n) \|_2^2 ,. \quad (13)$$

where

$$\boldsymbol{A}(n) = \begin{bmatrix} \boldsymbol{J}(n) \\ \sqrt{\mu}\boldsymbol{I} \end{bmatrix} \text{ and } \boldsymbol{b}(n)) = \begin{bmatrix} \boldsymbol{R}(n) \\ \boldsymbol{0} \end{bmatrix} . \quad (14)$$

Unlike (7), this LS problem cannot be solved by the recursive QRD in Table 1 due to the regularization term $\sqrt{\mu}\boldsymbol{I}$ and the use of the forgetting factor. A possible method to overcome this difficulty is to append successively the row vectors $\sqrt{\mu N}\boldsymbol{E}_m$ of the identity matrix, instead of the whole matrix, to the previous QR decomposition, where $\boldsymbol{E}_m = [0,\ldots,\underset{\uparrow}{1},0,\ldots,0]$ (1 in the m-th position), N is the regressor order and $m = (n \bmod N) + 1$. More precisely, at each time instant, the algorithm in Table 1 is executed once for the vector $\begin{bmatrix} \boldsymbol{\varphi}^T(n) & e_{\boldsymbol{\theta}(n-1)}(n) \end{bmatrix}^T$ and again for the vector $\begin{bmatrix} \sqrt{\mu N}\boldsymbol{E}_m & 0 \end{bmatrix}^T$. The arithmetic complexity is thus increased by a factor of two. Like the QR-RGN algorithm, a stepsize parameter $0 < \mu(n) \le 1$ can be introduced to reduce parameter variation due to additive noises.

IV. THE *P*-A-QR RECURSIVE NLS ALGORITHM

We now consider the further simplification of the QRD algorithm in Table 1 in order to achieve other complexity-performance tradeoffs. In particular, we shall solve the LS problem using an approximate QR-LS algorithm called *p*-A-QR-LS algorithm. It is a generalization of the approximate QR-LS (A-QR-LS) algorithms in [4,7], where the upper triangular matrix is approximated as a diagonal matrix in order to simplify the QRD and the back substitution to yield an arithmetic complexity of $O(N)$. More precisely, the quantities inside the square bracket in step 3 of Table 1 are computed from $\Delta\hat{\theta}_i(n-1)$ as

$$s_N(n-1) = r_{N,N+1}(n-1) ,$$

$$s_i(n-1) = [r_{i,N+1}(n-1) - \sum_{j=i+1}^{N} r_{i,j}(n-1)\Delta\hat{\theta}_j(n-1)]$$

$$i = N-1, N-2, \cdots 1 , \quad (15)$$

or $r_{i,i}(n-1)\Delta\hat{\theta}_i(n-1) = s_i(n-1),\ i = 1,...,N.$ (16)

Given the values of $s_i(n-1)$ and $r_{i,i}(n-1)$, (4) together with $\boldsymbol{\varphi}(n)$ and $e_{\theta(n-1)}(n)$, the new data at time n can be viewed as a system of linear equations in the variable $\Delta\hat{\theta}(n)$:

$$w \cdot r_{i,i}(n-1)\Delta\hat{\theta}_i(n) = w \cdot s_i(n-1),\ i = 1,...,N ,$$

$$\boldsymbol{\varphi}^T(n)\Delta\hat{\boldsymbol{\theta}}(n) = e_{\theta(n-1)}(n) , \quad (17)$$

where w is the square root of the forgetting factor. (17) can be modified and rewritten in matrix form as:

$$\boldsymbol{\Phi}(n)\Delta\hat{\boldsymbol{\theta}}(n) = \boldsymbol{b}(n) , \quad (18)$$

where $\boldsymbol{\Phi}(n) = \begin{bmatrix} w\boldsymbol{D}(n-1) \\ \boldsymbol{\varphi}^T(n) \end{bmatrix}$, $\boldsymbol{b}(n) = \begin{bmatrix} w[\boldsymbol{D}(n-1)\Delta\hat{\boldsymbol{\theta}}(n-1) - \boldsymbol{r}_{N+1}] \\ e_{\theta(n-1)}(n) \end{bmatrix}$,

$\boldsymbol{D}(n-1) = diag\{r_{1,1}(n-1), \cdots, r_{N,N}(n-1)\}$. $\boldsymbol{r}_{N+1} = [r_{1,N+1}, r_{2,N+1}, \cdots, r_{N,N+1}]^T$

Therefore, (18) can be solved by computing the QRD of $\boldsymbol{\Phi}(n)$, which actually works with the following appended matrix:

$$\widetilde{\boldsymbol{D}}(n) = \begin{bmatrix} wr_{1,1}(n-1) & & & w[s_1(n-1) - r_{1,N+1}] \\ & wr_{2,2}(n-1) & \boldsymbol{O} & w[s_2(n-1) - r_{2,N+1}] \\ \boldsymbol{O} & & \ddots & \vdots \\ & & wr_{N,N}(n-1) & w[s_N(n-1) - r_{N,N+1}] \\ [\varphi(n)]_1 & \cdots & [\varphi(n)]_N & e_{\theta(n-1)}(n) \end{bmatrix} \quad (19)$$

With this special structural approximation of the triangulated augmented data matrix (c.f. step 2 in Table 1), the A-QR-LS algorithm is able to combine the updating and the back solving processes together using the Householder transformation, yielding a very efficient algorithm. Liu *et al.* also proposed a related QR-LMS algorithm [5]. In [7], Chan and Yang proposed an improved transform domain approximate QR-LS (TA-QR-LS) algorithm based on the Givens rotations. Moreover, unitary transformation such as DCT is employed to improve the convergence speed when the input is colored.

The proposed p-QR-LS algorithm retains the main diagonal and p-1 nearby off-diagonals of the triangular factor, hence the name p-A-QR-LS algorithm. We shall show later that the p-A-QR-LS algorithm with a given positive integer p has a complexity of order $O(Np)$ and a performance, which generally improves as p increases. Therefore, it provides a practical tradeoff between performance and complexity when p is varied from 1 to N. The matrix $\boldsymbol{D}(n-1)$ in (18) is now modified to $\boldsymbol{D}_p(n-1)$ as:

$$\boldsymbol{D}_p(n-1) = \begin{bmatrix} r_{1,1}(n-1) & r_{1,2}(n-1 & \cdots & r_{1,p}(n-1) & 0 & \cdots & 0 & 0 \\ & r_{2,2}(n-1) & \cdots & r_{2,p}(n-1) & r_{2,p+1}(n-1) & 0 & \boldsymbol{0} & 0 \\ & & \ddots & & & \ddots & 0 & \vdots \\ & \boldsymbol{0} & & \ddots & & & \ddots & \\ & & & & \ddots & \vdots & & r_{N-p+1,N}(n-1) \\ & & & & & r_{N-2,N-2}(n-1) & r_{N-2,N-1}(n-1) & \vdots \\ 0 & 0 & \cdots & & & 0 & r_{N-1,N-1}(n-1) & r_{N-1,N}(n-1) \\ 0 & 0 & \cdots & & & & 0 & r_{N,N}(n-1) \end{bmatrix} \quad (20)$$

Motivated by the A-QR-LS algorithm, (18) is now modified to

$$\boldsymbol{\Phi}(n) = \begin{bmatrix} w\boldsymbol{D}_p(n-1) \\ \boldsymbol{\varphi}^T(n) \end{bmatrix} \text{ and}$$

$$\boldsymbol{b}(n) = \begin{bmatrix} w[\boldsymbol{D}_p(n-1)\Delta\hat{\boldsymbol{\theta}}(n-1) - \boldsymbol{r}_{N+1}] \\ e_{\theta(n-1)}(n) \end{bmatrix} = \begin{bmatrix} w[\boldsymbol{s}(n-1) - \boldsymbol{r}_{N+1}] \\ e_{\theta(n-1)}(n) \end{bmatrix} . \quad (21)$$

The QRD is then applied to solve for (21) by eliminating all the elements of $\boldsymbol{\varphi}^T(n)$ sequentially using Givens rotations. The procedure is very similar to that used in the TA-QR-LS algorithm [7]. However, the back substitution is considerably different from that in [7]. The p-TA-QR-LS algorithm so obtained is summarized in Table 2. From (8) and Table 2, it can be seen that when p=1, the proposed algorithm reduces to the TA-QR-LS algorithm with a complexity of $O(N)$, and when p=N, it reduces to the QR-RLS algorithm with a complexity of $O(N^2)$.

TABLE II. SQUARE-ROOT FREE GIVENS ROTATION-BASED P-TA-QR-LS ALGORITHM

1. *Initialization*
$\boldsymbol{\theta} = [0,0,\cdots,0]^T$
$\boldsymbol{\delta} = [1,1,\cdots,1]^T$, $\alpha_0 = 1$
2. *Recursive Operations*
$\boldsymbol{b} = \boldsymbol{D}_p\boldsymbol{\theta} - \boldsymbol{b}^*_{N+1}, \boldsymbol{\delta}' = w\boldsymbol{\delta}$
Add New Equation and DCT Transformation
$\boldsymbol{b}' = DCT(\boldsymbol{x}_d)$
Upper triangular Algorithm
For $i = 1,2,\cdots,N$ *Loop*
$\alpha' = \alpha_{i-1}\tilde{b}^{(i)}_{N+1,i}$, $\delta_i = \delta'_i + \alpha'\tilde{b}^{(i)}_{N+1,i}$,
$\sigma_i = \delta'_i / \delta_i$, $\rho_i = \alpha' / \delta_i$,
$\alpha_i = \alpha_{i-1}\sigma_i$, $\tau = b_{i,N+1}$,
$b^*_{i,N+1} = \sigma_i b_{i,N+1} + \rho_i b^{(i)}_{N+1,N+1}$,
$b^{(i+1)}_{N+1,N+1} = b^{(i)}_{N+1,N+1} - \tau b_{N+1,i}$
} 8N Multiplications, 3N Additions
If $i > N-p$ *or* $j < i+p$
$\tilde{b}^{(i+1)}_{N+1,j} = \tilde{b}^{(i)}_{N+1,j} - \tilde{b}_{i,j}\tilde{b}^{(i)}_{N+1,i}$,
$\tilde{b}^*_{i,j} = \sigma_i\tilde{b}_{i,j} + \rho_i\tilde{b}^{(i)}_{N+1,j}$
} (3/2)*(2N-p)(p-1) Multiplications, (2N-p)(p-1) Additions
End
End of Loop
Back Substitution
$\gamma_{1\sim N-1} = 0, \tilde{\theta}(N) = b^*_{N,N+1}$
For $i = N-1, N-2, \cdots, N-p+1$ *Loop*
For $j = N, N-1, \cdots, i+1$ *Loop*
$\gamma_i = \gamma_i + \tilde{b}^*_{i,j}\tilde{\theta}(j)$
End of Loop
$\tilde{\theta}(i) = b^*_{i,N+1} - \gamma_i$
} $p(p-1)/2$ Multiplications, $p(p-1)/2$ Additions
End of Loop
$\hat{\gamma}_{N-p+1} = 0$
For $i = N-p, N-p-1, \cdots, 2, 1$ *Loop*
$\hat{\gamma}_i = \hat{\gamma}_{i+1} + b_{N+1,p+1}\tilde{\theta}(i+p)$
$\hat{b}^*_{i,N+1} = b^*_{i,N+1} - \rho_i\hat{\gamma}_i$
For $j = i+p, \cdots, i+1$ *Loop*
$\gamma_i = \gamma_i + \tilde{b}^*_{i,j}\tilde{\theta}(j)$
End of Loop
$\tilde{\theta}(i) = \hat{b}^*_{i,N+1} - \gamma_i$
} (N-p)(p+1) Multiplications, (N-p)(p+1) Additions
End of Loop

V. SIMULATION RESULTS

We now evaluate the performance of the proposed algorithm by computer simulation of the following nonlinear system identification problem:

$$y(n) = g(\boldsymbol{X}^T(n)\cdot\boldsymbol{\theta}^\circ) \quad (22)$$

where $y(n)$ and $\boldsymbol{X}(n) = [x(n), x(n-1), \cdots, x(n-N+1)]^T$ are, respectively, the output and input of the system. $g(.)$ is a nonlinear function and

$$g(u) = [1 + \exp(-cu)]^{-1} - 0.5 \quad (23)$$

(22) is a simple perceptron and it reduces to the conventional linear transversal filter when $g(.)$ is the identity function: $g(u) = u$. The

derivative of $g(u)$, $g'(u)$, is given by $g'(u) = c \cdot \exp(-cu)[1+\exp(-(cu))]^{-2}$. Using the chain rule, one gets $\boldsymbol{\varphi}(n) = \nabla_\theta g_\theta(n)|_{\theta(n)} = g'(\boldsymbol{X}^T(n)\boldsymbol{\theta}(n)) \cdot \boldsymbol{X}$. The order of the regressor is set to 9 and the coefficients are randomly generated. The power of the additive white Gaussian noise $\eta(n)$ is set to be -40dB. The Mean Square Difference (MSD), defined as the sum of the square of the coefficient estimation errors, is used to evaluate the convergence behavior of the algorithms. All the results are averaged over 100 independent runs. QR-based and *p*-A-QR-based recursive Gauss-Newton (QR-RGN, *p*-A-QR-RGN) and Levenberg-Marquardt (QR-RLM, *p*-A-QR-RLM) algorithms have been tested. The forgetting factor *w* for all the tested algorithms are equally set to 0.99. The regularizing parameter δ in the *p*-A-QR-RLM algorithm is fixed at 0.00001. Three experiments were conducted. **Exp.1**: input is a white Gaussian noise (WGN) sequence with zero mean and unit variance, no transformation is employed. The results shown in Fig. 1 indicate that the two *p*-A-QR-based algorithms with different *p* approach a similar steady state error, whereas when *p* increases, the algorithms with larger *p* have faster initial convergence, which complies well with our expectation ; **Exp.2**: input is a WGN-driven AR process whose parameters are fixed at [1 –1.5 1 -0.25] with normalized unit power, no transformation employed; From Fig. 2, a similar conclusion can be drawn except that the converging speed of those algorithms with small *p* is hindered by the colored input. **Exp. 3**: input is the same as in **Exp.2**, and discrete cosine transform (DCT) of the input signal is employed. It can be observed that when the DCT is included, the results depicted in Fig. 3 show that the converging speed of the resulting *p*-TA-QR-RGN and *p*-TA-QR-RLM algorithms with small *p* are significantly improved. All these observations verify the principle and flexibility of the proposed algorithms and reveal their potential to be applied to different nonlinear applications.

VI. CONCLUSIONS

A family of approximate QR-based algorithms for recursive nonlinear least squares (NLS) estimation is presented. They are based on two new QR decomposition-based recursive algorithms derived from the classical Gauss-Newton (GN) and Levenberg Marquardt (LM) algorithms in nonlinear unconstrained optimization or least squares problems. A *p*-QR-LS algorithm is also proposed to solve the LS problem resulting from the linearization of the NLS problem. It achieves different complexity-performance tradeoffs by retaining different number of diagonal plus off-diagonals of the triangular factor of the augmented data matrix. Simulation results on identifying a nonlinear perceptron are provided to illustrate the usefulness of the new algorithms.

REFERENCES

[1] S. Haykin, *Adaptive Filter Theory*. 4th edition, Prentice Hall, 2001.

[2] G. Glentis, K. Berberidis, and S. Theodoridis, "Efficient least squares adaptive algorithms for FIR transversal filtering," *IEEE Signal Proc. Magazine*, pp. 13-41, July 1999.

[3] P. A. Regalia and M. G. Bellanger, "On the quality between fast QR methods and lattice methods in least squares adaptive filtering," *IEEE Trans. Signal Processing*, vol. 39, pp. 879-891, April 1991.

[4] Z. S. Liu. "QR methods of $O(N)$ complexity in adaptive parameter estimation," *IEEE Trans. Signal Processing*, vol. 43, pp. 720-729, 1995.

[5] Z. S. Liu and J. Li, "A QR-based least mean squares algorithm for adaptive parameter estimation," *IEEE Trans. Circuits Syst*., vol. 45, pp. 321-329, March 1998.

[6] J. G. Proakis, C. M. Rader, F. Ling, C. L. Nikias, M. Moonen, and I. K. Proudler. *Algorithms for Statistical signal Processing*. Prentice Hall Press, New Jersey, 2002.

[7] S. C. Chan and X. X. Yang, "Improved Approximate QR-LS Algorithms for Adaptive Filtering", *IEEE Trans. on Circuits and Systems II: Express Briefs*, vol. 51, no. 1, pp. 29-39, Jan. 2004.

[8] J. E. Dennis and R. B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice-Hall, Englewood Cliffs, NJ, 1983.

[9] S. C. Douglas and T. H. Meng. "Linearized least-squares training of multilayer feedforward neural networks," in *Int. Joint Conf. of Neural Networks*, pp. 307–312, 1991.

[10] L. Ljung and T. Soderstrom. *Theory and Practice of Recursive Identification*. M.I.T. Press, Cambridge, MA, 1983.

[11] L. S.H. Ngia, J. Sjoberg and M. Viberg, "Adaptive Neural Nets Filter Using a Recursive Levenberg-Marquardt Search Direction," in *Proc. Asilomar Conf. Signals, Systems, Computers,* pp. 697-701, Nov. 1998.

[12] O. Nerrand, P. Roussel-Ragot, L. Personnaz, G. Drefys, and S. Marcos, "Neural networks and nonlinear adaptive filtering: Unifying concepts and new algorithms," *Neural Computation*, 5(2), pp. 165–199, March 1993.

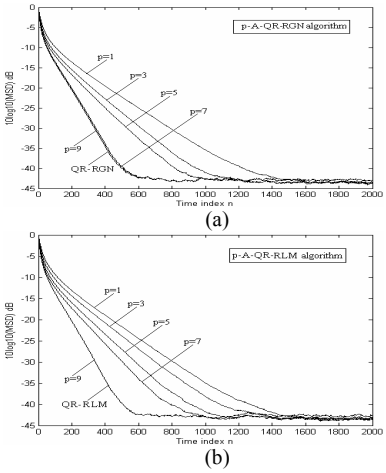

Fig. 1 MSD vs. time *n* (white input, no transformation employed) (a) *p*-A-QR-RGN algorithm,(b) *p*-A-QR-RLM algorithm.

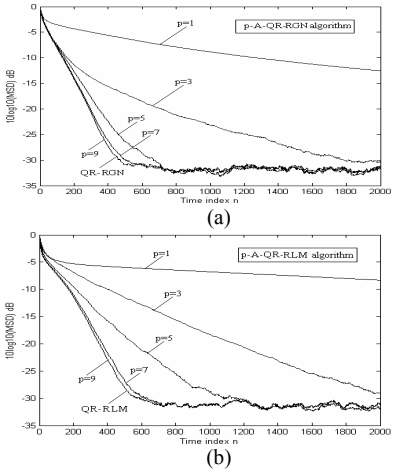

Fig. 2 MSD vs. time *n* (colored input, no transformation employed) (a) *p*-A-QR-RGN algorithm,(b) *p*-A-QR-RLM algorithm.

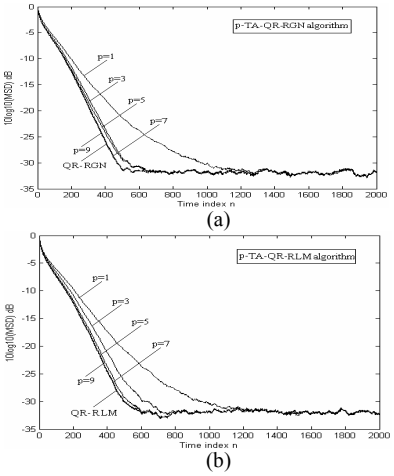

Fig. 3 MSD vs. time *n* (colored input, DCT employed), (a) *p*-TA-QR-RGN algorithm, (b) *p*-TA-QR-RLM algorithm.