Appl. Math. and Comp. Sci., 1998, Vol.8, No.2, 321–334

A FAST CONVERGING ALGORITHM FOR LANDAU'S OUTPUT ERROR METHOD

MORITZ HARTENECK*, ROBERT W. STEWART*

In this paper an approach to on-line ARMA parameter estimation based on a pseudo-linear regression and a QR matrix decomposition is developed. The algorithm has proven to be stable and has fast convergence properties if the unknown ARMA model satisfies the strictly positive real condition. The derivation of the algorithm is straightforward and the computational complexity is $O(N^2)$, however, fast versions of $O(N)$ computational complexity are readily available.

1. Introduction

Identification of the parameters which form a model of an unknown plant or transfer path is a very important issue in many control and signal processing applications such as controller design or in acoustic noise cancelation applications. In situations where the parameters of the plant or transfer path are non-stationary or not-known *a priori* then the use of on-line adaptive identification algorithms is a common approach. As a moving average model (MA) requires many adjustable parameters to model the plant satisfactorily usually auto regressive moving average (ARMA) models are preferred. However, the estimation algorithm becomes very complex and due to the non-quadratic nature of the performance surface the algorithm can converge to local minima.

Therefore many adaptive filtering algorithms for ARMA models have been developed such as full gradient or simplified gradient algorithms (Shynk, 1989). However, these algorithms exhibit a slow convergence if they are initialized in regions with a small gradient (i.e. close to local maxima, minima or saddle points) and can even converge to a local minimum yielding a non-optimal solution. To increase convergence speed of these algorithms, variants which incorporate the inverse of the autocorrelation matrix have been proposed but the on-line calculation of this matrix bears a high computational burden in addition to the computational complexity introduced by the weight update schemes. Furthermore, the necessary dynamic range of the algorithm is considerably increased.

One way to obtain simpler parameter estimation algorithms is to apply a pseudo-linear regression in the derivation of the algorithm as done in Landau's output error method (Landau, 1976) and in the signal processing community by Feintuch (1976). This method is shown to converge for unknown systems which satisfy the strictly

* Signal Processing Division, Department of Electrical & Electronic Engineering, University of Strathclyde, Glasgow, G1 1XW, Scotland, UK, e-mail: {moritz,bob}@spd.eee.strath.ac.uk.

positive real condition (Ljung and Söderström, 1983). To overcome this strictly positive real constraint adaptive filtering algorithms based on hyperstability have been proposed (Larimore *et al.*, 1980; Johnson *et al.*, 1981) but these assume some prior knowledge of the unknown system.

In this paper a parameter estimation algorithm based on a pseudolinear regression and an orthogonal matrix decomposition is presented which operates on data values directly and therefore reduces the necessary dynamic range for computation. The algorithm shows rapid convergence even in noisy conditions and is able to track system changes effectively.

In Section 2, the algorithm is derived and some comments on convergence and computational complexity are made. In Section 3, a hierarchical signal-flow-graph representation is presented which is the basis for fast versions of the algorithm which can be derived using algorithmic engineering techniques as shown in (Harteneck *et al.*, 1996a; McWhirter, 1992; Proudler and McWhirter, 1994). In Section 4, simulation results are presented which show the performance of the algorithm in different set-ups. Finally, in Section 6, some concluding remarks are made.

2. Derivation

The output $\hat{y}(k)$ of the structure shown in Fig. 1 at time k can be written as

$$\hat{y}(k) = \sum_{i=0}^{M-1} \hat{a}_i(k)u(k-i) + \sum_{i=1}^{L-1} \hat{b}_i(k)\hat{y}(k-i) \tag{1}$$

where $u(k)$ is the driving input, $\hat{y}(k)$ is the output sequence of the estimation algorithm and $\{\hat{a}\}$ and $\{\hat{b}\}$ are the adaptive feedforward and feedback parameters, respectively. Note that this is a representation of Landau's output-error model (Landau, 1976; Ljung and Söderström, 1983). The error signal $e(k)$ of the estimation algorithm can then be written as

$$e(k) = d(k) - \hat{y}(k) \tag{2}$$

where $d(k)$ is the desired signal which is composed as

$$d(k) = y(k) + n(k) \tag{3}$$

where $y(k)$ is the signal to be estimated and $n(k)$ is additive observation noise (c.f. Fig. 4). It is assumed that $x(k)$ and $n(k)$ are statistically independent and as $y(k)$ is calculated by a similar model as (1), $y(k)$ and $n(k)$ are also statistically independent.

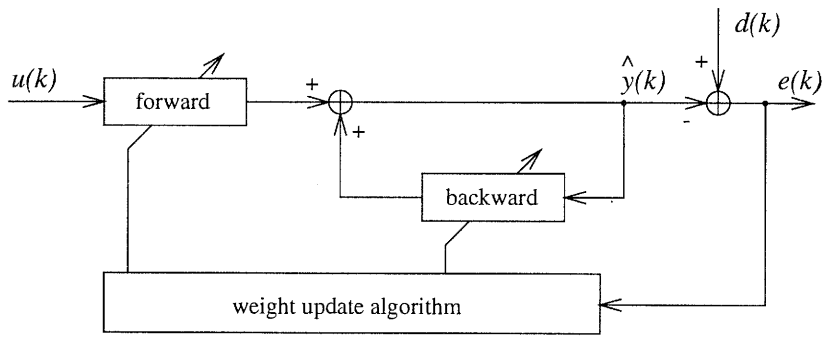


Fig. 1. Underlying structure.

In a least-squares formulation, the aim of the parameter estimation algorithm is to minimize the power of the error signal $e(k)$ assuming the parameters $\{\hat{a}\}$ and $\{\hat{b}\}$ to be fixed for all time prior to k. In the following equations this condition is denoted by subscript $\hat{\Theta}$. The performance criterion we wish to minimize is therefore

$$\xi(k) = \sum_{i=0}^{k} \lambda^i e^2(k-i)|_{\hat{\Theta}} \tag{4}$$

where λ is a forgetting factor slightly smaller than 1 which enables the parameter estimation algorithm to adapt to changing environments. To present this performance criterion in a more structured way it is useful to define some vectors and write it in a matrix-vector notation. The necessary definitions are

$$\hat{\boldsymbol{a}}(k) = [\hat{a}_0(k)\ \hat{a}_1(k)\ \ldots\ \hat{a}_{M-1}(k)]^T \tag{5a}$$

$$\hat{\boldsymbol{b}}(k) = [\hat{b}_1(k)\ \hat{b}_2(k)\ \ldots\ \hat{b}_{L-1}(k)]^T \tag{5b}$$

$$\boldsymbol{u}(k) = [u(k)\ u(k-1)\ \ldots\ u(k-M+1)]^T \tag{5c}$$

$$\hat{\boldsymbol{y}}(k)|_{\hat{\Theta}} = [\hat{y}(k-1)|_{\hat{\Theta}}\ \hat{y}(k-2)|_{\hat{\Theta}}\ \ldots\ \hat{y}(k-L+1)|_{\hat{\Theta}}]^T \tag{5d}$$

$$\hat{\boldsymbol{y}}(k) = [\hat{y}(k-1)\ \hat{y}(k-2)\ldots\ \hat{y}(k-L+1)]^T \tag{5e}$$

$$\boldsymbol{d}(k) = [d(1)\ d(2)\ \ldots\ d(k)]^T \tag{5f}$$

$$\boldsymbol{e}(k) = [e(1)\ e(2)\ \ldots\ e(k)]^T \tag{5g}$$

$$\boldsymbol{\Lambda}(k) = \text{diagonal}(\lambda^{k-1}, \lambda^{k-2}, \ldots, \lambda, 1) \tag{5h}$$

where 'T' denotes matrix transpose. In particular, note the difference between (5d) and (5e). The elements of (5d) are calculated assuming the adaptive weights were fixed for all time prior to k, whereas (5e) uses the time varying weights, i.e. (5e) is composed of the output signal $\hat{y}(k)$ of the parameter estimation algorithm. Now (4)

can be written as

$$\xi(k) = \left\| \boldsymbol{\Lambda}^{\frac{1}{2}}(k) \left[\underbrace{\begin{bmatrix} d(1) \\ d(2) \\ \vdots \\ d(k) \end{bmatrix}}_{\boldsymbol{d}(k)} - \underbrace{\begin{bmatrix} \boldsymbol{u}^T(1) & \hat{\boldsymbol{y}}^T(1)|_{\hat{\Theta}} \\ \boldsymbol{u}^T(2) & \hat{\boldsymbol{y}}^T(2)|_{\hat{\Theta}} \\ \vdots & \vdots \\ \boldsymbol{u}^T(k) & \hat{\boldsymbol{y}}^T(k)|_{\hat{\Theta}} \end{bmatrix}}_{\boldsymbol{A}(k)|_{\hat{\Theta}}} \underbrace{\begin{bmatrix} \hat{\boldsymbol{a}}(k) \\ \hat{\boldsymbol{b}}(k) \end{bmatrix}}_{\hat{\Theta}(k)} \right] \right\|^2$$

$$= \left\| \boldsymbol{\Lambda}^{\frac{1}{2}}(k)\boldsymbol{d}(k) - \boldsymbol{\Lambda}^{\frac{1}{2}}(k)\boldsymbol{A}(k)|_{\hat{\Theta}}\hat{\Theta}(k) \right\|^2 \tag{6}$$

where $\|\cdot\|$ denotes Euclidean norm. The estimation algorithm has to minimize the norm of (6) by choosing the adaptive parameters accordingly. This is, however, a nonlinear minimization problem because of the dependency of the data matrix $\boldsymbol{A}(k)|_{\hat{\Theta}}$ on the adaptive parameters $\hat{\Theta}$.

One possible approximation is to substitute $\hat{y}(k)|_{\hat{\Theta}}$ by $\hat{y}(k)$ (the output sequence of the parameter estimation algorithm) and so break the parameter dependency of $\boldsymbol{A}(k)|_{\hat{\Theta}}$ on $\hat{\Theta}$. Equation (6) can then be approximated as

$$\xi(k) \approx \left\| \boldsymbol{\Lambda}^{\frac{1}{2}}(k) \left[\underbrace{\begin{bmatrix} d(1) \\ d(2) \\ \vdots \\ d(k) \end{bmatrix}}_{\boldsymbol{d}(k)} - \underbrace{\begin{bmatrix} \boldsymbol{u}^T(1) & \hat{\boldsymbol{y}}^T(1) \\ \boldsymbol{u}^T(2) & \hat{\boldsymbol{y}}^T(2) \\ \vdots & \vdots \\ \boldsymbol{u}^T(k) & \hat{\boldsymbol{y}}^T(k) \end{bmatrix}}_{\boldsymbol{A}(k)} \underbrace{\begin{bmatrix} \hat{\boldsymbol{a}}(k) \\ \hat{\boldsymbol{b}}(k) \end{bmatrix}}_{\hat{\Theta}(k)} \right] \right\|^2$$

$$\approx \left\| \boldsymbol{\Lambda}^{\frac{1}{2}}(k)\boldsymbol{d}(k) - \boldsymbol{\Lambda}^{\frac{1}{2}}(k)\boldsymbol{A}(k)\hat{\Theta}(k) \right\|^2 \tag{7}$$

This approximation is known in the literature as pseudo-linear regression (PLR) and its convergence criteria are reviewed in (Ljung, 1987; Shynk, 1989) and studied in (Ljung and Söderström, 1983). In (Ljung and Söderström, 1983) a convergence proof for the PLR is given for the sufficient-order case if the transfer function $1/B(z)$ of the unknown system is strictly positive real (SPR), i.e.

$$\mathrm{Re}\left[\frac{1}{B(e^{\jmath\omega})} - \frac{1}{2}\right] > 0, \quad \forall -\pi < \omega < \pi$$

where $B(z) = 1 + \sum_{i=1}^{L-1} b_i z^{-i}$. For the undermodeled case some useful results can be found in (Regalia, 1994).

To achieve the minimization of (7), this equation is premultiplied by a $k \times k$ unitary matrix $\boldsymbol{Q}(k)$ to produce a structured matrix equation which is easy to solve. The unitary matrix $\boldsymbol{Q}(k)$ is chosen in such a way that

$$\boldsymbol{Q}(k)\boldsymbol{\Lambda}^{\frac{1}{2}}(k)\boldsymbol{A}(k) = \begin{bmatrix} \boldsymbol{R}(k) \\ \boldsymbol{0} \end{bmatrix} \tag{8}$$

where $\boldsymbol{R}(k)$ is an $(M+L-1)\times(M+L-1)$ upper-triangular matrix and $\mathbf{0}$ is a $(k-M-L+1)\times(M+L-1)$ zero matrix. This matrix decomposition is referred to in the literature as QR decomposition (Haykin, 1991; Golub and Van Loan, 1989). Premultiplying (7) with $\boldsymbol{Q}(k)$ and using (8) gives

$$\begin{aligned}\xi(k) &\approx \|\boldsymbol{Q}(k)\boldsymbol{\Lambda}^{\frac{1}{2}}(k)\boldsymbol{d}(k) - \boldsymbol{Q}(k)\boldsymbol{\Lambda}^{\frac{1}{2}}(k)\boldsymbol{A}(k)\hat{\boldsymbol{\Theta}}(k)\|^2 \\ &\approx \left\| \begin{bmatrix} \boldsymbol{p}(k) - \boldsymbol{R}(k)\hat{\boldsymbol{\Theta}}(k) \\ \boldsymbol{v}(k) \end{bmatrix} \right\|^2 \end{aligned} \tag{9}$$

where the vector $\boldsymbol{Q}(k)\boldsymbol{\Lambda}^{\frac{1}{2}}(k)\boldsymbol{d}(k)$ is partitioned in a suitable way into the vectors $\boldsymbol{p}(k)$ and $\boldsymbol{v}(k)$. As $\boldsymbol{R}(k)$ is an upper-triangular matrix, the parameters which minimize this performance criterion can be extracted via backsubstitution from

$$\boldsymbol{R}(k)\hat{\boldsymbol{\Theta}}(k) = \boldsymbol{p}(k) \tag{10}$$

In (Haykin, 1991) a time-recursive version of the above algorithm is derived which can be summarized as

$$\begin{bmatrix} \boldsymbol{R}(k) \\ \boldsymbol{O}^T \end{bmatrix} = \boldsymbol{J}(k) \begin{bmatrix} \lambda^{\frac{1}{2}}\boldsymbol{R}(k-1) \\ \boldsymbol{u}^T(k) \quad \hat{\boldsymbol{y}}^T(k) \end{bmatrix} \tag{11a}$$

$$\begin{bmatrix} \boldsymbol{p}(k) \\ \delta(k) \end{bmatrix} = \boldsymbol{J}(k) \begin{bmatrix} \lambda^{\frac{1}{2}}\boldsymbol{p}(k-1) \\ d(k) \end{bmatrix} \tag{11b}$$

where $\delta(k)$ is the last element of $\boldsymbol{Q}(k)\boldsymbol{\Lambda}^{\frac{1}{2}}(k)\boldsymbol{d}(k)$ and $\boldsymbol{J}(k)$ is an $(M+L-1)\times(M+L-1)$ orthogonal rotation matrix which annihilates $[\boldsymbol{u}^T(k)\ \hat{\boldsymbol{y}}^T(k)]$. One way to construct $\boldsymbol{J}(k)$ is to use a sequence of $M+L-1$ Given's rotations which each zero one element of $[\boldsymbol{u}^T(k)\ \hat{\boldsymbol{y}}^T(k)]$. Note that all the matrices of the algorithm are of fixed size and so it is possible to implement the algorithm on-line.

If standard Given's rotations are used for the calculation of the orthogonal matrix $\boldsymbol{J}(k)$, then $2N^2+7N$ multiply-accumulates (MACs) and N inverse-square-roots are needed for the computation of the rotation (11) and $\frac{1}{2}N^2+\frac{1}{2}N$ MACs and N divides are needed for the backsubstitution (10) where N is the number of adaptive parameters, i.e. $N=M+L-1$.

In applications where the parameter vector $\hat{\boldsymbol{\Theta}}(k)$ is of no interest but the error signal $e(k)$, or the parameter vector is only needed after adaptation, the potentially unstable backsubstitution step can be circumvented by applying the technique of direct residual extraction (McWhirter, 1983) which calculates the *a-posteriori* error residual directly from

$$e(k) = \delta(k)\gamma(k) \tag{12}$$

where $\delta(k)$ is defined by (11b) and $\gamma(k)$ is the likelihood factor which is defined as

$$\gamma(k) = \boldsymbol{\pi}^T\boldsymbol{J}(k)\boldsymbol{\pi} = \boldsymbol{\pi}^T\boldsymbol{Q}(k)\boldsymbol{\pi} \tag{13}$$

Table 1. Parameter estimation algorithm for Landau's output-error method.

1	$\boldsymbol{R} = \mathbf{0}$; $\boldsymbol{p} = \mathbf{0}$;
2	FOR $k = 0$ TO SIMTIME DO
3	$\boldsymbol{R} = \lambda^{\frac{1}{2}} \cdot \boldsymbol{R}$; $\boldsymbol{p} = \lambda^{\frac{1}{2}} \cdot \boldsymbol{p}$;
4	FOR $i = 0$ TO $L-1$ DO
5	$r_{L+M-1,i} = u_{k-i}$;
6	FOR $i = 0$ TO $M-2$ DO
7	$r_{L+M-1,L+i} = \hat{y}_{k-i-1}$;
8	$p_{L+M-1} = d_k$;
9	$\delta_k = 1$;
10	FOR $i = 0$ TO $L+M-2$ DO
11	IF $r_{i,i} == 0$ THEN
12	$c = 0$; $s = 1$;
13	ELSE
14	$d = (r_{i,i}^2 + r_{L+M-1,i}^2)^{-\frac{1}{2}}$;
15	$c = r_{i,i} \cdot d$; $s = r_{L+M-1,i} \cdot d$;
16	FOR $j = i$ TO $L+M-2$ DO
17	$r_{i,j} = c \cdot r_{i,j} + s \cdot r_{L+M-1,j}$;
18	$r_{L+M-1,j} = c \cdot r_{L+M-1,j} - s \cdot r_{i,j}$;
19	$p_i = c \cdot p_i + s \cdot p_{M+L-1}$;
20	$p_{L+M-1} = c \cdot p_{L+M-1} - s \cdot p_i$;
21	$\delta_k = \delta_k \cdot c$;
22	$e_k = p_{L+M-1} \cdot \delta_k$;
23	$\hat{y}_k = d_k - e_k$;

where $\pi^T = [0 \ \ldots \ 0 \ 1]$ is a pinning vector of appropriate dimension. In Tab. 1 the proposed algorithm is shown in pseudo-code applying the direct residual extraction method. If the parameters are needed after adaptation, then the backsubstitution step (10) has to be performed once when, the matrix $\boldsymbol{R}(k)$ is well behaved and so no numerical problems occur.

By using this technique, the computational complexity reduces to $2N^2 + 8N$ MACs and N square-roots for the rotation and the calculation of the likelihood value.

3. Signal-Flow-Graph Implementation

In Fig. 2, a signal-flow-graph representation of the proposed algorithm is shown for 3 feedforward and 2 feedback parameters for the case of direct residual extraction. The

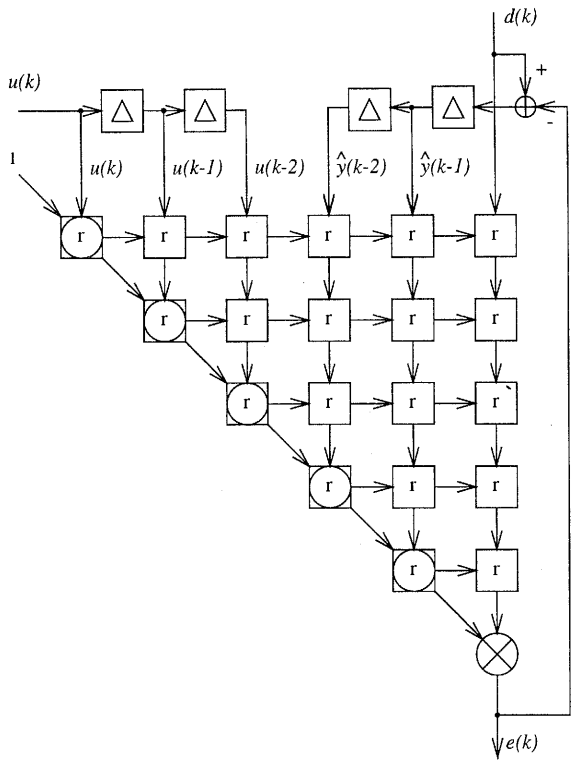


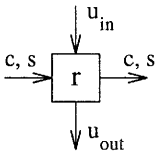
Fig. 2. Signal-flow-graph implementation of the proposed algorithm for 3 feedforward and 2 feedback parameters.

shown array consists of two types of cells, internal and boundary, where the boundary cells calculate the rotation parameters of the rotations which zero the elements of $[\boldsymbol{u}^T(k)\ \hat{\boldsymbol{y}}^T(k)]$ (c.f. (11)) and the internal cells perform the actual rotations. The implementation of these two types of cells depends strongly on the environment where the algorithm is used (e.g. floating point, fixed point, dynamic range) and which type of rotation is therefore preferred. Some specific implementations can be found in (Haykin, 1991; 1996; McWhirter *et al.*, 1995). In Fig. 3 one possible set of cells implementing standard Given's rotations is shown.

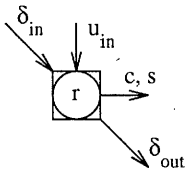
4. Simulations

Representative simulations were carried out using a system identification set-up (Fig. 4) as used in (Åström and Eykhoff, 1971). There, the driving input $u(k)$ is a pseudo-random binary sequence (PRBS) and the unknown system is realized by

$$y(k) = u(k-1) + 0.5u(k-2) + 1.5y(k-1) + 0.7y(k-1)$$



$$u_{\text{out}} = c \cdot u_{\text{in}} - s \cdot \lambda^{\frac{1}{2}} \cdot r;$$
$$r = s \cdot u_{\text{in}} + c \cdot \lambda^{\frac{1}{2}} \cdot r;$$



if $u_{\text{in}} = 0$ then
$c = 1;\ s = 0;\ r = \lambda^{\frac{1}{2}} \cdot r;$
$\delta_{\text{out}} = c \cdot \delta_{\text{in}};$
else
$c = \lambda^{\frac{1}{2}} \cdot r \cdot (\lambda \cdot r^2 + u_{\text{in}}^2)^{-\frac{1}{2}};$
$s = u_{\text{in}} \cdot (\lambda \cdot r^2 + u_{\text{in}}^2)^{-\frac{1}{2}};$
$r = (\lambda \cdot r^2 + u_{\text{in}}^2)^{-\frac{1}{2}};$
$\delta_{\text{out}} = c \cdot \delta_{\text{in}};$
endif

Fig. 3. Realization of the internal and boundary cells implementing a standard Given's rotation.

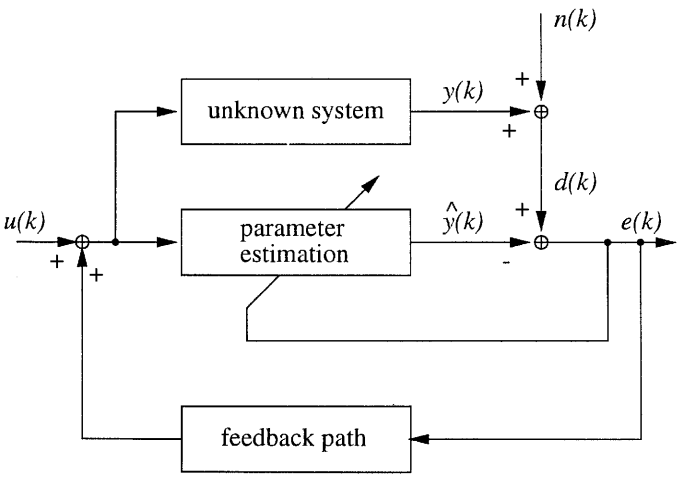


Fig. 4. System identification set-up with feedback path.

which has a complex conjugate pole pair at $0.75 \pm j0.3708$ and a single zero at -0.5. The observation noise $n(k)$ is realized in the simulations as a pseudo-random sequence (PRS) with a Gaussian distribution which is independent of the input sequence $u(k)$. The feedback path is meant to simulate the effects of online simulation of a closed loop controller.

The first set of simulations, which is shown in Fig. 5, is parameter identification for the noise free case, i.e. $n(k) = 0$, and no feedback path present. In Fig. 5, the averaged squared error (over 20 simulations) against time are shown for (a) an IIR least mean squares (IIR-LMS) algorithm (Landau, 1976; Feintuch, 1976) with 3 feedforward and 2 feedback parameters and (b) for the proposed IIR-QR algorithm with 3 feedforward and 2 feedback parameters. The step size for the gradient based algorithm was $\mu = 0.001$ for all parameters and the forgetting factor was chosen to be $\lambda = 0.999$ which results in the excess mean squared error (which is, in this case, higher than the noise floor). The proposed algorithm converges almost instantaneously to the real parameters, which are found, in this case, as well by the IIR-LMS algorithm. Simulations for higher order unknown systems as shown in (Harteneck *et al.*, 1996b; Harteneck and Stewart, 1996) show that the IIR-LMS can converge to a local minimum while the proposed algorithm finds the real parameters. In (Harteneck *et al.*, 1996b; Harteneck and Stewart, 1996) the algorithm is compared with an IIR recursive prediction error (IIR-RPE) (Haykin, 1996) which is outperformed but in the presented (low order) simulations the IIR-LMS and the IIR-RPE behave very similar.

The second set of simulations was system identification of the same unknown system but with an observation noise present, i.e. $n(k) \neq 0$. The level of observation noise was chosen such that a signal-to-noise ratio in the desired signal $d(k)$ of $-20\,\mathrm{dB}$ was achieved. In Fig. 6 the averaged squared error (over 20 simulations) is shown for the (a) IIR-LMS algorithm and the (b) IIR-QR algorithm with the same parameters as in the first set of simulations. It can be seen that the IIR-QR algorithm adapts again very quickly to a low mean squared error which this time is the noise floor. To demonstrate that the proposed algorithm converges to the real parameters, the learning curves of the adaptive parameters are shown in Fig. 7 which reveal that the true parameters are found. The IIR-LMS converges to the same solution but with a much slower rate of convergence.

The third set of simulations was system identification with the same unknown system, observation noise and a feedback path present. The feedback path was modeled as a delayed and attenuated version of the unknown system. The simulation results obtained with the proposed algorithm showed no degradation compared to the second set of simulations (c.f. Figs. 6 and 7). The algorithm converged quickly to a low minimum mean squared error reaching the true parameters. Trials with the IIR-LMS algorithms resulted in instability of the algorithm which is caused by the feedback nature of the set-up and therefore an ill-conditioned autocorrelation matrix of the input signal.

Finally, Table 2 and 3 show the results of the identification of a real world acoustic transfer path to compare the number of parameters necessary to model the system. The 16 bit/8 kHz data was captured inside two acoustic enclosures which are slightly

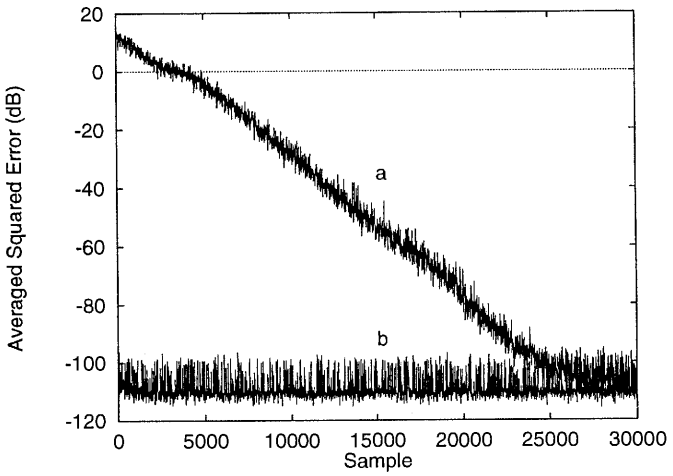


Fig. 5. Ensemble squared error signals (over 20 simulations) of system identification with no observation noise, (a) IIR-LMS algorithm, (b) IIR-QR algorithm.

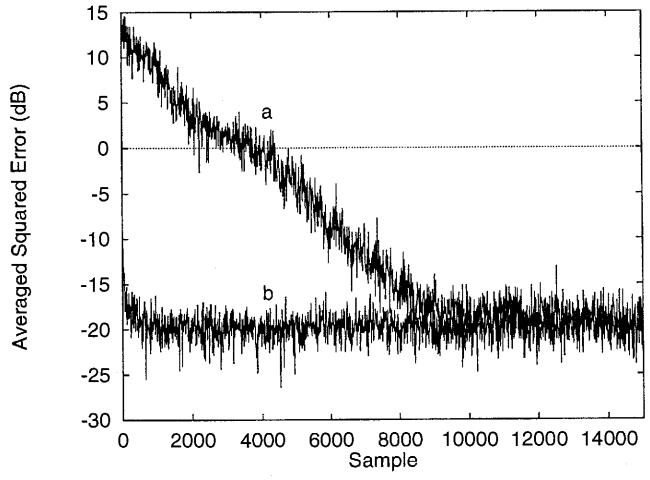


Fig. 6. Ensemble squared error signals (over 20 simulations) of system identification with 15 dB observation noise, (a) IIR-LMS algorithm, (b) IIR-QR algorithm.

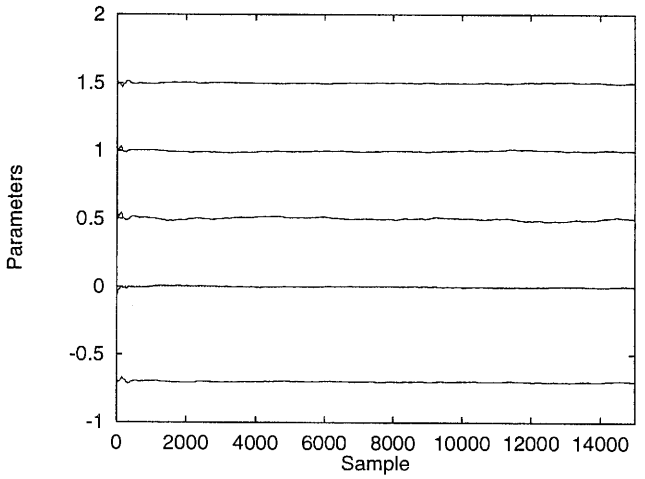


Fig. 7. Adaptive parameters with SNR of -20 dB in the desired signal $d(k)$.

Table 2. Real world data: slightly reverberant environment.

FIR-LMS	ERLE	IIR-QR	ERLE
50 taps	1.6 dB	$L = M = 9$	1.5 dB
100 taps	2.9 dB	$L = M = 12$	2.7 dB
150 taps	4.2 dB	$L = M = 21$	4.3 dB

Table 3. Real world data: less reverberant environment.

FIR-LMS	ERLE	IIR-QR	ERLE
50 taps	7.1 dB	$L = M = 19$	7.1 dB
100 taps	10.2 dB	$L = M = 30$	10.8 dB
150 taps	11.2 dB	$L = M = 35$	11.5 dB

reverberant (length of impulse response about 0.06 seconds, Table 2) and less reverberant (length of impulse response about 0.02 seconds, Table 3). The criterion chosen for this comparison was the power of the desired signal $d(k)$ divided by the power of the residual error signal $e(k)$ after adaptation, which is called in echo cancelation systems echo-return-loss-enhancement (ERLE). The results show that to achieve the same amount of ERLE, the IIR-QR needs considerably less adaptive parameters than the FIR-LMS algorithm.

Table 4. Complexities of different algorithms ($N = L + M - 1$).

Algorithm	MAC's	Square Roots	Divides
IIR-RPE Algorithm with Simplified Gradient (Shynk, 1989)	$5N^2 + 3N$	0	1
IIR-LMS Algorithm (Feintuch, 1976)	$2N$	0	0
IIR-QR with Direct Residual Extraction	$2N^2 + 5N$	N	N
Square root free IIR-QR with Direct Residual Extraction	$2N^2 + 8N$	0	$2N$

5. Computational Complexity

In Table 4, the computational complexities of the proposed algorithm, a square-root free version and two other adaptive IIR algorithms are shown where N is the order of the adaptive filtering problem ($N = L+M-1$). The IIR-QR algorithm has a lower computational complexity than the simplified gradient RPE. However, it needs more divisions and square-roots to obtain the results but this is compensated by the lower number of required MAC's. In the square-root free version, the number of required divisions is increased and therefore the overall complexity is actually not reduced. (On a state of the art signal processor, an inverse square-root needs 11 clock cycles whereas a square-root needs 12 clock cycles and a division 8 clock cycles (Motorola, 1989; Texas Instruments, 1991).) Compared with the IIR-LMS algorithm proposed by Feintuch, the IIR-QR algorithm is a factor of N more complex but it clearly converges faster than the IIR-LMS algorithm (Shynk, 1989).

To reduce the computational complexity from $O(N^2)$ to $O(N)$ fast versions are available for the direct residual extraction method, as presented in this paper, and a version with parallel weight extraction as shown in (Haykin, 1991). These derivations are presented in (Harteneck *et al.*, 1996a) based on a derivation using algorithmic engineering techniques applied to the representation shown in Fig. 2.

6. Conclusion

In this paper we propose a straightforward and efficient parameter estimation algorithm for ARMA model estimation. The algorithm is realized by applying a pseudo-linear regression (PLR) technique and a QR matrix decomposition to the non-linear optimization problem and requires $O(N^2)$ computations. The PLR technique is well studied and convergence proofs are known for strictly positive real systems in the sufficient order case. A signal flow graph representation of the proposed algorithm is shown which can be used to develop fast $O(N)$ and parallel realizations of the

algorithm. To date floating point simulations with synthetic data and real acoustic data have demonstrated that the proposed algorithm is a powerful and stable algorithm and superior to gradient search based algorithms in terms of adaptation speed, achievable minimum mean squared error and computational requirements. Another advantage is that the algorithm remains stable if a feedback path is present from the output to the input of the unknown system, a constellation which is often encountered when on-line modeling of a plant in a feedback controller set-up.

Acknowledgments

We would like to thank J.G. McWhirter and I.K. Proudler for their useful comments and input.

References

Åström K.J. and Eykhoff P. (1971): *System identification–A survey.* — Automatica, Vol.7, pp.123–162.

Feintuch P.L. (1976): *An adaptive recursive LMS filter.* — Proc. IEEE, Vol.64, No.11, pp.1622–1624.

Golub G.H. and Van Loan C.F. (1989): *Matrix Computations.* — John Hopkins University Press.

Harteneck M. and Stewart R.W. (1996): *Acoustic Echo Cancellation Using a Pseudo-Linear Regression and QR-Decomposition.* — Proc. ISCAS, Atlanta, Vol.II, pp.233–236.

Harteneck M., Stewart R.W., McWhirter J.M. and Proudler I.K. (1996a): *An algorithmically engineered fast multichannel adaptive filter based on the QR-RLS.* — Proc. 4-th Int. Conf. Mathematics in Signal Processing, Warwick, U.K.

Harteneck M., Stewart R.W., McWhirter J.G. and Proudler I.K. (1996b): *An effective approach to adaptive IIR filtering.* — Proc. ICASSP, Atlanta, Vol.3, pp.1634–1637.

Haykin S. (1991): *Adaptive Filter Theory, 2nd Ed.* — Englewood Cliffs, New Jersey: Prentice Hall.

Haykin S. (1996): *Adaptive Filter Theory, 3rd Ed.* — Englewood Cliffs, New Jersey: Prentice Hall.

Johnson C.R., Larimore M.G., Treichler J.R. and Anderson B.D.O. (1981): *SHARF convergence properties.* — IEEE Trans. Acoust. Speech Signal Process., Vol.29, No.3, pp.659–670.

Landau I.D. (1976): *Unbiased recursive identification using model reference techniques.* — IEEE Trans. Automat. Contr., Vol.21, pp.194–202.

Larimore M.T., Treichler J.R. and Johnson C.R. (1980): *SHARF: An algorithm for adapting IIR digital filters.* — IEEE Trans. Acoust. Speech Signal Process., Vol.28, No.4, pp.428–440.

Ljung L. (1987): *System Identification: Theory for the User.* — Englewood Cliffs, New Jersey: Prentice Hall.

Ljung L. and Söderström T. (1983): *Theory and Practice of Recursive Identification.* — Cambridge, Massachusetts: MIT Press.

McWhirter J.G. (1983): *Recursive least-squares minimization using a systolic array.* — Proc. SPIE, Real-Time Signal Process. VI, Vol.431, pp.105–112.

McWhirter J.G. (1992): *Algorithmic engineering in adaptive signal processing.* — IEE Proc.-F, Vol.139, No.3, pp.226–232.

McWhirter J.G., Walke R.L. and Kadlec J. (1995): *Normalised givens rotations for recursive least squares processing.* — Proc. IEEE Workshop *VLSI Signal Processing*, Japan.

Motorola, Inc. (1989): *DSP96002, IEEE Floating-Point Dual-Port Processor User's Manual.*

Proudler I.K. and McWhirter J.G. (1994): *Algorithmic engineering in adaptive signal processing: Worked examples.* — IEE Proc. Vis. Image Signal Process., Vol.141, No.1, pp.19–26.

Regalia P.A. (1994): *Adaptive IIR Filtering in Signal Processing and Control.* — New York: Marcel Dekker.

Shynk J.J. (1989): *Adaptive IIR filtering.* — IEEE ASSP Mag., Vol.6, No.4, pp.4–21.

Texas Instruments, Inc. (1991): *TMS320C4x — User's Guide.*