ELSEVIER

Signal Processing 60 (1997) 339–347

SIGNAL PROCESSING

Identification of nonminimum phase FIR systems via fourth-order cumulants and genetic algorithm

Maha Shadaydeh[a], Yegui Xiao[b,*], Yoshiaki Tadokoro[b]

[a] Graduate School of Engineering, Tohoku University, Aramaki, Aoba, Aoba-ku, Sendai-shi, 981-77, Japan
[b] Faculty of Engineering, Toyohashi University of Technology, Toyohashi-shi, 441, Japan

Received 27 November 1995; revised 7 February 1997

Abstract

In this paper, the problem of estimating the parameters of an FIR system from only the fourth-order cumulants of the noisy system output is considered. The FIR system is driven by a symmetric, independent, and identically distributed (i.i.d) non-Gaussian sequence. We propose a new formula called Weighted Overdetermined $C(q,k)$ (WOC(q,k)) by extending the conventional $C(q,k)$ formula. The optimal selection of the weights in WOC(q,k) is performed by using the Genetic Algorithm (GA) optimization method which minimizes a nonlinear error function based on the fourth-order cumulants alone. Simulations are provided to reveal the effectiveness and the superiority of this novel technique over the $C(q,k)$ and other existing techniques. © 1997 Elsevier Science B.V.

Zusammenfassung

In diesem Artikel wird das Problem der Parameterschätzung eines FIR-Systems nur aus Kumulanten 4. Ordnung eines verrauschten Systemausganges betrachtet. Das FIR-System wird von einer symmetrisch, unabhängig identisch verteilten, nicht-gaußschen Sequenz angesteuert. Wir schlagen eine neue Methode, Weighted Overdetermined $C(q,k)$ (WOC(q,k)), als Erweiterung des konventionellen $C(q,k)$-Verfahrens vor. Die optimale Wahl der Gewichte in WOC(q,k) wird von einem genetischen Algorithmus (GA) übernommen, welcher einen nichtlinearen Fehlerterm lediglich anhand von Kumulanten 4. Ordnung minimiert. Simulationen zeigen die Effektivität und bessere Qualität dieser neuen Technik gegenüber $C(q,k)$ und anderen Techniken. © 1997 Elsevier Science B.V.

Résumé

Nous abordons dans cet article le problème de l'estimation des paramètres d'un système FIR à partir seulement des cumulants d'ordre quatre de la sortie bruitée du système. Le système FIR est excité par une séquence non-gaussienne à distribution invariante, indépendante (i.i.d.) et symétrique. Nous proposons une formule nouvelle appelée $C(q,k)$ surdéterminée pondérée (WOC(q,k)) étendant la formule $C(q,k)$ conventionnelle. La sélection optimale des coefficients de pondération dans WOC(q,k) se fait à l'aide d'une méthode d'optimisation par algorithme génétique (GA) qui minimise une fonction d'erreur non-linéaire basée sur les cumulants d'ordre quatre seuls. Des simulations sont fournies pour mettre en évidence l'efficience et la supériorité de cette technique nouvelle vis-à-vis du $C(q,k)$ et d'autres techniques existantes. © 1997 Elsevier Science B.V.

Keywords: FIR system identification; Fourth-order cumulants; Weighted least-square; Genetic algorithm

* Corresponding author. Tel.: +81-532-44-6756; fax: +81-532-44-6757; e-mail: xiao@signal.tutics.tut.ac.jp.

PII S0165-1684(97)00084-4

1. Introduction

In the past few years higher-order cumulants have motivated considerable research work in system identification due to their ability to estimate nonminimum phase systems and their immunity from Gaussian noise. FIR system identification using higher-order cumulants has received considerable attention, and a lot of techniques have been developed [1–3, 5–14]. The well-known $C(q,k)$ formula [2] was the first to show that an FIR system with non-Gaussian input can be identified by its third or fourth-order output cumulants alone. This formula carries over to any mth-order cumulants ($m \geqslant 5$). However, it does not smooth out the effect of the measurement noise, and hence produces large estimation variance. Many techniques, such as the GM method [3], the T-method [10], and so on [5], have been developed to overcome its statistical deficiency. These techniques use both the higher-order cumulants and the autocorrelation. They show improved performance compared to the $C(q,k)$ formula [2] when the system is contaminated by a white Gaussian noise or by a colored noise generated by an MA process with known system order. However, if the additive measurement noise is an ARMA Gaussian process, their performance will be severely degraded due to the use of the autocorrelation which is not blind to the additive colored noise.

Recently, several methods using third or fourth or both third- and fourth-order cumulants alone have been proposed [1, 6–9, 11–12, 14] to handle the colored measurement noise. They work much better than those methods using both higher-order cumulants and autocorrelation for colored measurement noise scenarios. However, in many digital communication applications (see [10] and references therein), the input signals are symmetrically distributed, and hence their third-order cumulants are all zero. For such cases, only fourth-order cumulants can be used. So far, to the best of our knowledge, only two methods which make use of the fourth-order cumulants alone have been proposed. The first one is the method by Zhang et al. [14] which consists of a system of linear and overdetermined equations derived using the $C(q,k)$ formula and based on third- or fourth-order cumulants. This method was found to work better in general than the $C(q,k)$ and the previous methods using both higher-order cumulants and autocorrelation. However, its performance is system dependent, and it may give considerably large estimation variance due to the direct use of the $C(q,k)$ formula in its derivation. The second one is a 'modified' $C(q,k)$ with overdeterminacy derived by Mo and Shafai [6], and the authors of [8, 9, 11] independently. It was referred to as Overdetermined $C(q,k)$ ($OC(q,k)$) in [8, 9, 11]. It is very simple in the form, but works much better than the $C(q,k)$ formula. In $OC(q,k)$, all the used cumulant slices are treated equally. However, selective use of them may lead to further improved estimation if we weigh them in an appropriate way. It is this observation that has motivated the present novel technique.

This paper focuses on the FIR system identification in an additive Gaussian ARMA noise using the fourth-order cumulants alone. First, a Weighted Overdetermined $C(q,k)$ ($WOC(q,k)$) formula is derived. This formula is based on the weighted least-squares (LS) solution of a system of overdetermined linear equations derived by extending the $C(q,k)$. The weights can provide a room to select the cumulant slices which are less noisy, or can contribute more information about the FIR system. Next, a Genetic Algorithm (GA) [4] is used to find the optimal weights via the minimization of a nonlinear error function of the fourth-order output cumulants. In this way, we can keep the linearity of the estimator $WOC(q,k)$ while the information buried in other cumulant slices that are not included in $WOC(q,k)$ formula is utilized. The introduction of weights to the $OC(q,k)$ and the use of GA to search the optimal weights form the major novelty of this paper. Extensive simulations have revealed that this technique leads to a considerable improvement in estimation performance compared with many other previous techniques developed in the literature.

The organization of this paper is as follows. Section 2 describes the derivation of the $WOC(q,k)$. In Section 3 the use of the GA optimization method for the weights is considered. Simulation results are shown in Section 4 to demonstrate the better performance of the proposed technique. Finally, conclusions are drawn in Section 5.

2. The Weighted Overdetermined $C(q,k)$ formula

Consider the following FIR system:

$$x(n)=\sum_{i=0}^{q} b(i)w(n-i), \tag{1}$$

$$y(n)=x(n)+v(n), \tag{2}$$

where $b(i)$ $(i=0,\ldots,q)$ is the coefficient of the FIR system, and $y(n)$ is the noisy output. It is assumed here that

(A1) The driving noise sequence $w(n)$ is a zero-mean, i.i.d non-Gaussian process that is not observed, and its fourth-order cumulant γ_{4w} satisfies $0<|\gamma_{4w}|<\infty$.

(A2) The system is nonminimum phase with $b(0)=1$, and $b(q)\neq 0$, where q denotes the system order which is assumed to be known or correctly estimated by the existing techniques such as [13].

(A3) The additive noise $v(n)$ is a zero-mean Gaussian ARMA process with unknown power spectrum, and is independent of the input $w(n)$.

The fourth-order cumulant of the output signal $y(n)$ is calculated by

$$c_{4y}(\tau_1,\tau_2,\tau_3) = \gamma_{4w}\sum_{i=0}^{q} b(i)b(i+\tau_1)b(i+\tau_2)b(i+\tau_3). \tag{3}$$

Setting $\tau_1=q$, $\tau_2=i$ $(0\leqslant i\leqslant q)$ and $\tau_3=k$ in (3), and using the fact that $b(i)=0$ for $i>q$, we find that

$$c_{4y}(q,i,k)=\gamma_{4w}\, b(q)b(k)b(i). \tag{4}$$

Next setting $\tau_1=q$, $\tau_2=i$ $(0\leqslant i\leqslant q)$ and $\tau_3=0$ in (3) and using the assumption that $b(0)=1$, we find that

$$c_{4y}(q,i,0)=\gamma_{4w}b(q)b(i). \tag{5}$$

Substituting (5) in (4), we can derive a system of $(q+1)$ linear and overdetermined equations for each of the unknown parameters $b(k)$, $k=1,\ldots,q$, given in a matrix form as follows:

$$G_kAb(k)=G_kB_k, \quad k=1,\ldots,q, \tag{6}$$

where

$$A=[c_{4y}(q,0,0)\ldots c_{4y}(q,i,0)\ldots c_{4y}(q,q,0)]^{\mathrm{T}},$$

$$B_k=[c_{4y}(q,0,k)\ldots c_{4y}(q,i,k)\ldots c_{4y}(q,q,k)]^{\mathrm{T}}$$

and G_k is a diagonal weight matrix

$$G_k=\mathrm{diag}(g_{k0},\ldots g_{ki},\ldots g_{kq})$$

with $0<g_{ki}\leqslant 1$, $(i=0,\ldots,q)$. Letting $G_kA=C_k$, $G_kB_k=D_k$ and solving (6) for $b(k)$ yield a least-squares solution as follows:

$$b(k)=(C_k^{\mathrm{T}}C_k)^{-1}C_k^{\mathrm{T}}D_k, \quad k=1,\ldots,q, \tag{7}$$

which is called the Weighted Overdetermined $C(q,k)$ (WOC(q,k)) formula [11]. If G_k is a unit matrix, we call (7) the Overdetermined $C(q,k)$ (OC(q,k)) [8]. Note that $C_k^{\mathrm{T}}C_k$ and $C_k^{\mathrm{T}}D_k$ are scalars, and hence matrix inversion is not needed here.

By giving a weight to every equation in the proposed method a better exploitation of the information contained in the cumulant slices involved can be expected. So far, we have not found an analytical way to handle the weight matrix selection, and here suggest to use the GA [4] to search the optimal weights. It is known that GA has the ability to efficiently search large spaces about which little is known, and spaces involving noise. The use of the GA is described in the following section.

3. The selection of the weights using GA

GA is a stochastic search and optimization algorithm based on the mechanics of evolution and natural genetics. In nature, competition among individuals for scanty resources results in the fittest individuals dominating over weaker ones. GA simulates the survival of the fittest among individuals over consecutive generation for solving a problem. Each generation consists of a population of *individuals* represented by the *chromosomes*, a set of character strings. Each individual represents a point in a search space and a possible solution. The individuals in the population are then made to go through a process of evolution. Each individual is evaluated, selected and recombined with other individuals on the basis of its overall fitness with respect to the given application domain. Therefore,

$$\begin{array}{lcl} 01010|011 & \xrightarrow{Crossover} & 01010110 \\ 11000|110 & & 11000011 \end{array}$$

(a)

$$01010011 \xrightarrow{Mutation} 01010001$$

(b)

Fig. 1. (a) Crossover operation. (b) Mutation operation: the preultimate bit to the right is mutated from a 1 to a 0.

high-performing individuals may be chosen for replication several times. This eventually leads to a population that has improved fitness with respect to the given goal. New individuals (offspring) of the next generation are formed using two main genetic operators, *crossover* and *mutation*. Crossover operates with crossover probability P_{cross} by randomly selecting a point in the two selected parents gene structures and exchanging the remaining segments of the parents to create new offspring. This operation is depicted in Fig. 1(a). Mutation operates with mutation probability P_{mutation} by randomly changing one or more components of a selected individuals as shown in Fig. 1(b). It acts as a population perturbation operator and is a mean of inserting new information into the population. The evolution process, i.e, reproduction of new generation continues until the GA reaches a termination criteria such as predefined maximum number of generations, fitness value, etc.

The main issues in applying GA to any optimization problem are an appropriate representation of the individual (chromosome), and an adequate evaluation function (fitness). For the weight optimization problem these two issues are explained as follows.

3.1. Representation issue

The first step in applying GA to any optimization problem is to map the search space into a representation suitable for genetic search. In our problem we present all the weights used in (6), (g_{ki}: $0 \leqslant i \leqslant q$, $1 \leqslant k \leqslant q$) in one chromosome as a binary string of length $L = l_w(q+1)q$, where l_w denotes the number of bits used to code a weight g_{ki} to a binary substring $w_{k,i}$. This representation is depicted in Fig. 2. Using this representation, it is easy to apply the GA operations, i.e, crossover and mutation, as shown in Fig. 1, to the weight optimization problem.

$$\overbrace{\underbrace{1011}_{w_{q,q}} \cdots \underbrace{1101}_{w_{k,i}} \cdots \underbrace{1001}_{w_{1,1}}\,\underbrace{1000}_{w_{1,0}}}^{\text{chromosome}}$$

Fig. 2. The chromosome representation.

3.2. Evaluation function

The second important issue for successful use of GA is the appropriate selection of the evaluation function which provides the GA with a feedback about the fitness of every individual in the population. Here, the evaluation function is defined as follows:

$$\begin{aligned} &\text{fitness} = \frac{E_{\max} - E}{E_{\max}} \quad && \text{if } E < E_{\max}, \\ &\text{fitness} = 0 && \text{if } E \geqslant E_{\max}, \end{aligned} \tag{8}$$

where E is an error function defined as

$$E = \sum_{l=0}^{q} \sum_{m=l}^{q} \sum_{n=m}^{q} \left\{ c_{4y}(l,m,n) - \hat{\gamma}_{4w} \sum_{i=0}^{q} \hat{b}(i)\hat{b}(i+l)\hat{b}(i+m)\hat{b}(i+n) \right\}^2 \tag{9}$$

and $E_{\max}$ is the error function E evaluated by the system parameters obtained by the OC(q,k). The estimated value of $\hat{\gamma}_{4w}$ is calculated by

$$\hat{\gamma}_{4w} = \frac{c_{4y}(0,0,0)}{\sum_{i=0}^{q} [\hat{b}(i)]^4}, \tag{10}$$

where $\hat{b}(1), \hat{b}(2), \ldots, \hat{b}(q)$ are system parameters obtained from (7), with the weights obtained by decoding each l_w bits of the binary string of the chromosome ($w_{k,i}$) as depicted in Fig. 2 into its corresponding real value g_{ki}. The decoding procedure is

$$g_{ki} = \frac{1 + \sum_{n=1}^{l_w} 2^{n-1} w_{k,i}(n)}{2^{l_w}}, \tag{11}$$

where $w_{k,i}(n)$ is the nth bit of the binary substring $w_{k,i}$. Eq. (11) maps the binary range $[0, 2^{l_w}]$ to the normalized real range $[2^{-l_w}, 1]$.

3.3. Estimation algorithm

The proposed method consists of the following two essential steps:

- *Step 1*. Perform parameter estimation using (7), with G_k equal to unit matrix for $1 \leqslant k \leqslant q$ (the OC(q,k) formula). Then, calculate the error $E_{\max}$ using (9).
- *Step 2*. Use the GA optimization method to search for an optimal weight matrix G_k in the sense of minimizing the error function E in (9). This solution is referred to as GA-WOC(q,k) [9] in this paper.

The GA terminates when no more improvement in the best fitness value can be obtained, i.e., the best fitness value remains the same during a specified number of consecutive generations. Then, the weights with the best fitness value in the last generation are taken as the final solution. If the GA fails to give a better solution than the OC(q,k), the unit weight matrix is considered as the final solution. We used the above termination strategies in our simulations. Furthermore, the number of generations needed in the optimization process is generally proportional to the system order q.

It could be possible to use other optimization methods to optimize the FIR system parameters using initial values obtained by the OC(q,k) like the GR-OC(q,k) (see next section), but they may involve high degree of nonlinearity, which results in a great increase in the computational complexity. In the case of using GA for the weight matrix optimization, we can take advantage of the linearity of the WOC(q,k). Furthermore, by using a specific range $[2^{-l_w},1]$ for the weights we can decrease to a great extent the search space, and hence can increase the convergence rate.

4. Simulations

In this section we illustrate the performance of the proposed algorithm through many examples. The simulation process was carried out for several algorithms, the conventional C(q,k) [2], the simple OC(q,k), the GR-OC(q,k) (a gradient-based algorithm using error function (9) and initial values obtained by the OC(q,k)), the GA-WOC(q,k), and the algorithm proposed by Zhang et al. [14]. In GR-OC(q,k), the $\hat{\gamma}_{4w}$ estimated by (10) is fixed at its initial value, and not considered as a function of the coefficients. And, it was found in our extensive simulations that this way results in better coefficient estimates than treating it as a highly nonlinear function of the coefficients.

In all these examples the system is nonminimum phase. The input $w(n)$ is a uniformly distributed i.i.d non-Gaussian process with $\sigma_w^2 = 1$. The additive noise $v(n)$ is a Gaussian ARMA(3,1) process defined as

$$v(n) + 2.2v(n-1) + 1.77v(n-2) - 0.52v(n-3) \\ = e(n) - 1.25e(n-1), \tag{12}$$

where $e(n)$ is a zero-mean white Gaussian noise with variance $\sigma_e^2 = 1$. We define the signal-to-noise ratio as SNR (dB) $= 10\log(P_x/P_v)$, with P_x and P_v the powers of the system output and the observation noise, respectively.

For the weight optimization we used the Simple Genetic Algorithm (SGA) software described in [4]. In our simulation we used $l_w = 4$ (the number of bits used to represent a weight). Generally, l_w must be chosen large enough to give a good variation in the weights of the $(q+1)$ set of equations derived for each of the filter coefficients. However, it must also be chosen as small as possible to decrease the search space. Generally, the larger the system order is, the more bits have to be used to get enough 'resolution' for the weights. We found that if l_w is too small, then the GA-WOC(q,k) presents almost no performance improvement, and if it is too long, the performance improvement saturates and the search process will last much longer. For the systems considered in the simulations, we confirmed that $l_w = 4$ is a reasonable selection.

The selection procedure for the next generation in these examples is based on the stochastic tournament selection, which operates by randomly picking a number of individuals equal to the tournament-size which is less than the total population size. Then among the chosen individuals the one with the highest fitness is chosen for the next generation production. This process continues until the population size is reached. In our simulations we used tournament size $= 4$. Parameters for crossover and mutation operations in GA, $P_{\mathrm{cross}} = 0.7$ and $P_{\mathrm{mutation}} = 0.001$, have been used in all the simulations, and it is also confirmed that these operations operate properly during the search process.

The numerical results are shown in Tables 1–4. Fig. 3 shows a convergence process of the best fitness

Table 1
Results for $B(z) = 1 - 2.33z^{-1} + 0.667z^{-2}$ (25 runs)

Data length			1024				5196			
SNR			0 dB		10 dB		0 dB		10 dB	
Algorithm	True value		mean	σ	mean	σ	mean	σ	mean	σ
$C(q,k)$	b(1)	−2.33	0.526	1.188	0.156	2.364	−0.318	2.132	−3.761	10.87
Ref. [2]	b(2)	0.667	1.300	1.785	−0.208	2.193	1.094	2.099	1.527	4.479
$OC(q,k)$	b(1)	−2.33	−0.357	0.466	−1.240	0.675	−1.085	0.619	−1.804	0.767
Refs. [6, 8]	b(2)	0.667	0.742	0.835	0.644	0.491	0.348	0.697	0.749	0.376
GR-$OC(q,k)$	b(1)	−2.33	−0.859	0.669	−1.648	0.697	−1.486	0.677	−1.861	0.772
	b(2)	0.667	0.733	1.000	0.632	0.536	0.445	0.838	0.729	0.390
Zhang et al.	b(1)	−2.33	−1.088	1.233	−0.763	1.315	−0.938	0.639	−2.283	2.408
Ref. [14]	b(2)	0.667	0.317	1.999	0.327	1.183	0.429	1.031	3.360	3.511
GA-$WOC(q,k)$	b(1)	−2.33	−1.097	0.917	−1.799	0.558	−1.459	0.995	−2.227	0.357
	b(2)	0.667	0.828	1.034	0.721	0.486	0.409	0.786	0.689	0.203

Table 2
Results for $B(z) = 1 + 0.9z^{-1} + 0.385z^{-2} - 0.771z^{-3}$ (25 runs)

Data length			1024				5196			
SNR			0 dB		10 dB		0 dB		10 dB	
Algorithm	True value		mean	σ	mean	σ	mean	σ	mean	σ
$C(q,k)$	b(1)	0.9	0.7089	1.127	0.872	1.930	2.566	7.443	0.972	0.223
Ref. [2]	b(2)	0.385	1.562	6.632	0.102	2.515	1.558	6.706	0.441	0.195
	b(3)	−0.771	−0.630	4.562	−0.778	1.600	−1.944	4.965	−0.815	0.188
$OC(q,k)$	b(1)	0.9	0.862	0.271	0.873	0.151	0.887	0.106	0.903	0.081
Refs. [6, 8]	b(2)	0.385	0.386	0.247	0.354	0.202	0.385	0.137	0.380	0.100
	b(3)	−0.771	−0.675	0.336	−0.751	0.272	−0.734	0.193	−0.793	0.101
GR-$OC(q,k)$	b(1)	0.9	0.867	0.258	0.876	0.142	0.894	0.097	0.899	0.075
	b(2)	0.385	0.389	0.243	0.353	0.195	0.384	0.132	0.380	0.096
	b(3)	−0.771	−0.665	0.319	−0.743	0.241	−0.737	0.175	−0.788	0.093
Zhang et al.	b(1)	0.9	−0.616	5.572	0.808	0.793	0.527	1.148	0.894	0.075
Ref. [14]	b(2)	0.385	−0.502	5.341	0.146	0.735	0.129	1.257	0.227	0.404
	b(3)	−0.771	−1.213	6.991	−0.702	0.733	−0.246	1.436	−0.780	0.076
GA-$WOC(q,k)$	b(1)	0.9	0.880	0.168	0.892	0.117	0.888	0.079	0.894	0.053
	b(2)	0.385	0.427	0.406	0.366	0.149	0.391	0.114	0.396	0.079
	b(3)	−0.771	−0.747	0.448	−0.767	0.248	−0.805	0.154	−0.771	0.080

Table 3
Results for $B(z) = 1 - 0.8z^{-1} + 1.52z^{-2} - 0.64z^{-3} + 0.99z^{-4}$ (data length = 5196, 25 runs)

SNR			0 dB		10 dB	
Algorithm	True value		mean	σ	mean	σ
$C(q,k)$ Ref. [2]	b(1)	−0.8	−0.363	1.946	−0.635	0.863
	b(2)	1.52	1.345	2.157	1.238	1.240
	b(3)	−0.64	−0.217	0.708	−0.505	0.803
	b(4)	0.99	1.895	6.678	0.676	1.290
$OC(q,k)$ Refs. [6, 8]	b(1)	−0.8	−0.688	0.582	−0.769	0.230
	b(2)	1.52	1.262	0.518	1.396	0.357
	b(3)	−0.64	−0.569	0.470	−0.540	0.405
	b(4)	0.99	0.700	0.452	0.840	0.410
GR-$OC(q,k)$	b(1)	−0.8	−0.737	0.525	−0.786	0.234
	b(2)	1.52	1.322	0.491	1.425	0.352
	b(3)	−0.64	−0.613	0.460	−0.553	0.406
	b(4)	0.99	0.740	0.436	0.840	0.409
Zhang et al. Ref. [14]	b(1)	−0.8	0.180	1.363	−0.528	0.686
	b(2)	1.52	0.116	1.378	1.260	0.554
	b(3)	−0.64	−0.097	1.396	−0.583	0.725
	b(4)	0.99	0.076	1.437	0.967	0.547
GA-$WOC(q,k)$	b(1)	−0.8	−0.931	0.441	−0.785	0.192
	b(2)	1.52	1.295	0.434	1.400	0.230
	b(3)	−0.64	−0.708	0.476	−0.630	0.281
	b(4)	0.99	0.827	0.363	0.962	0.270

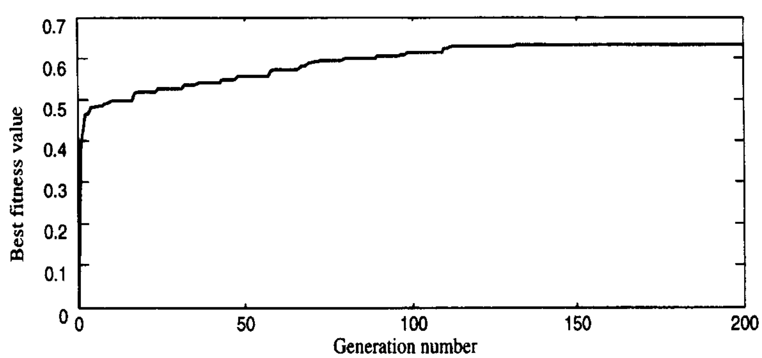

Fig. 3 Convergence process of the best fitness value of each generation for the system $B(z) = 1 - 2.33z^{-1} + 0.75z^{-2} + 0.5z^{-3} + 0.3z^{-4} - 1.44z^{-5}$ (data length = 5196, SNR = 10 dB, $l_w = 4$, population size = 20, $P_{cross} = 0.7$, $P_{mutation} = 0.001$).

value (8) of each generation for a fifth-order FIR system. From these simulation results we can draw the following conclusions:

(1) The $OC(q,k)$ formula yields a considerable improvement in the performance compared to the conventional $C(q,k)$, since it uses 2-D slices of cumulants, and also works much better than the method by Zhang et al. [14] while it has an elegant form and small computational load. Moreover, it can provide a very good initial values for the optimization process of the error function (9).

Table 4
Results for $B(z) = 1 - 2.33z^{-1} + 0.75z^{-2} + 0.5z^{-3} + 0.3z^{-4} - 1.44z^{-5}$ (data length = 5196, 25 runs)

SNR			0 dB		10 dB	
Algorithm	True value		mean	σ	mean	σ
$C(q,k)$	b(1)	−2.33	0.031	2.034	0.335	139.6
Ref. [2]	b(2)	0.75	0.430	1.607	−1.470	7.998
	b(3)	0.5	0.182	1.786	1.006	2.204
	b(4)	0.3	0.239	1.051	0.217	3.086
	b(5)	−1.4	0.064	2.644	0.270	10.030
$OC(q,k)$	b(1)	−2.33	−1.067	0.406	−1.745	0.713
Refs. [6,8]	b(2)	0.75	0.403	0.307	0.513	0.530
	b(3)	0.5	0.223	0.225	0.347	0.340
	b(4)	0.3	0.191	0.188	0.338	0.256
	b(5)	−1.4	−0.821	0.479	−1.133	0.432
GR-$OC(q,k)$	b(1)	−2.33	−1.673	0.479	−1.621	0.542
	b(2)	0.75	0.487	0.307	0.350	0.245
	b(3)	0.5	0.346	0.178	0.207	0.219
	b(4)	0.3	0.258	0.156	0.448	0.273
	b(5)	−1.4	−1.056	0.370	−1.050	0.384
Zhang et al.	b(1)	−2.33	0.064	1.737	−2.126	2.95
Ref. [14]	b(2)	0.75	−0.330	1.749	0.663	2.131
	b(3)	0.5	0.269	1.698	0.133	0.941
	b(4)	0.3	0.528	1.438	0.649	0.636
	b(5)	−1.4	−0.553	1.426	−0.691	0.903
GA-$WOC(q,k)$	b(1)	−2.33	−1.695	0.469	−2.002	0.550
	b(2)	0.75	0.451	0.308	0.596	0.344
	b(3)	0.5	0.339	0.172	0.394	0.194
	b(4)	0.3	0.272	0.159	0.354	0.114
	b(5)	−1.4	−1.00	0.384	−1.230	0.346

(2) The GA-$WOC(q,k)$, in all these examples, enabled a better exploitation of the information of the cumulant slices involved, and outperformed $OC(q,k)$, GR-$OC(q,k)$ and the method by Zhang et al. [14] in terms of both the mean value and the standard deviation.

It should be noted that in our simulations for the method by Zhang et al. [14], we estimated the system parameters using both Algorithms 1 and 2 presented in it, and took the best estimates of these two algorithms according to Remark 2 in [14].

5. Conclusions

We have presented a fourth-order cumulant-based algorithm for an FIR system identification. This method is based on the weighted LS solution of a system of linear equations obtained by extending the conventional $C(q,k)$. We suggested the use of the GA for the optimal weight selection. Simulations were carried out for several examples to show the marked estimation performance of the proposed technique.

Acknowledgements

The authors would like to express their cordial thanks to the anonymous reviewers whose comments have helped improve the paper.

References

[1] M.C. Carrion, D.P. Ruiz, A. Gallego, J.A. Morente, FIR system Identification using the third- and fourth-order cumulants, Electron. Lett. 31 (8) (1995) 612–614.

[2] G. Giannakis, Cumulant: A powerful tool in signal processing, Proc. IEEE, 75 (9) (1987) 1333–1334.
[3] G. Giannakis, J.M. Mendel, Identification of nonminimum phase system using higher order statistics, IEEE Trans. Acoust. Speech Signal Process. 37 (1989) 360–377.
[4] D.E. Goldberg, Genetic Algorithm in Search, Optimization, and Machine Learning, The University of Albama.
[5] J.M. Mendel, Tutorial on higher-order statistics (spectra) in signal processing and system theory: Theoretical results and some applications, Proc. IEEE 79 (3) (1991) 278–305.
[6] S. Mo, B. Shafai, Blind equalization using higher order cumulants and neural network, IEEE Trans. Signal Process. 42 (11) (1994) 3209–3217.
[7] Y.J. Na, K.S. Kim, I. Song, T. Kim, Identification of nonminimum phase FIR systems using the third and fourth-order cumulants, IEEE Trans. Signal Process. 43 (8) (1995) 2018–2022.
[8] M. Shadaydeh, Y. Xiao, Y. Tadokoro, Moving average system identification using overdetermined $C(q,k)$ formula, Toukai-section joint convention record of the six institutes of electrical and related engineers, Japan, October 1994, p. 264.
[9] M. Shadaydeh, Y. Xiao, Y. Tadokoro, Moving average system identification using fourth-order cumulants, IEICE Technical Report, DSP95-111, Japan, October 1995, pp. 55–62.
[10] J.K. Tugnait, New results on FIR system identification using higher order statistics, IEEE Trans. Acoust. Speech Signal Process. 39 (10) (1991) 2216–2223.
[11] Y. Xiao, M. Shadaydeh, Y. Tadokoro, Moving average system identification using fourth-order cumulants, Presented at IEICE Spring Conf. Japan, March 1995, p. 168.
[12] Y. Xiao, M. Shadaydeh, Y. Tadokoro, Overdetermined $C(q,k)$ formula using the third- and fourth-order cumulants, Electron. Lett. 32 (6) (1996) 601–603.
[13] X.-D. Zhang, Y.-S. Zhang, Singular value decomposition-based MA order determination of non-Gaussian ARMA models, IEEE Trans. Signal Process. 41 (1993) 2657–2664.
[14] X.-D. Zhang, Y.-S. Zhang, FIR system identification using higher order statistics alone, IEEE Trans. Signal Process. 42 (10) (1994) 2854–2858.