New Approaches Without Postprocessing to FIR System Identification Using Selected Order Cumulants

Wei Li and Wan-Chi Siu, *Senior Member, IEEE*

***Abstract*—In this paper, we address the problem of identifying the parameters of the nonminimum-phase FIR system from the cumulants of noisy output samples. The system is driven by an unobservable, zero-mean, independent and identically distributed (i.i.d) non-Gaussian signal. The measurement noise may be white Gaussian, colored MA, ARMA Gaussian processes, or even real noises.**

For this problem, two novel methods are proposed. The methods are designed by using higher order cumulants with the following advantages. i) Flexibility: Method 1 employs two arbitrary adjacent order cumulants of output, whereas Method 2 uses three cumulants of output: two cumulants with arbitrary orders and the other one with an order equal to the summation of the two orders minus one. Because of this flexibility, we can select cumulants with appropriate orders to accommodate different applications. ii) Linearity: Both the formulations in Method 1 and Method 2 are linear with respect to the unknowns, unlike the existing cumulant-based algorithms. The post-processing is thus avoided.

Extensive experiments with ARMA Gaussian and three real noises show that the new algorithms, especially Algorithm 1, perform the FIR system identification with higher efficiency and better accuracy as compared with the related algorithms in the literature.

***Index Terms*—FIR system identification, higher order cumulants, parameter estimation.**

I. Introduction

FINITE IMPULSE response (FIR) system identification based on higher order cumulants of system output has received great attention in recent years. Tools that deal with problems related to either nonlinearities, non-Gaussianity, or nonminimum-phase (NMP) systems are available, and they are of great value in applications, such as radar, sonar, array processing, blind equalization, time-delay estimation, data communication, image and speech processing, and seismology.

Consider an unknown FIR system (i.e., the MA process) given by

$$x(k) = \sum_{i=0}^{q} h(i)u(k-i) \tag{1}$$

where

- $\{x(k)\}$ system output;
- $\{u(k)\}$ non-Gaussian, nth-order white independent and identically distributed (i.i.d) signal with zero-mean;
- $\{h(i)\}$ impulse response sequence with $h(0) = 1$;
- q order of the model.

The received signal in the real world is not $x(k)$ in general but is contaminated by some noises. Let us denote this received signal as $y(k)$

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

where $n(k)$ is a measurement noise, which may be any of the white Gaussian, colored MA Gaussian, or colored ARMA Gaussian noises and is independent of $u(k)$. The identification problem requires the estimation of parameters $\{h(i), i = 1, \ldots, q\}$ from the real output $y(k)$ and its higher order cumulants.

Based on the model in (1), a number of algorithms have been proposed that have been summarized in the survey paper written by Mendel [1]. According to this paper, the algorithms mentioned belong to three broad classes:

1) closed-form solutions;
2) linear algebra solutions;
3) optimization solutions.

Recently, the linear algebra solutions have received great attention because they have simpler computations and are free of the problems of local extreme values that often occur in the optimization solutions. Although the closed-form solutions have similar features, they usually do not smooth out the noises caused from observation and computation. Therefore, while these solutions are interesting from the theoretical point of view, they are not recommended for practical applications [2].

The key in linear algebra solutions is to establish a linear relationship between parameters $\{(h(i)\}$ (sometimes including $\{h^2(i)\}$) and higher order statistics of $y(k)$. Applying a certain criterion (for example, the least-square principle), we can get the estimates of $\{h(i)\}$. The pioneering work of using higher order statistics of the output for identifying FIR system was done by Giannakis in [3]. No doubt, this method provides a new way for FIR system identification, but it cannot smooth out the effect of additive noise arising from the computation of higher order cumulants. To overcome the deficiency, Giannakis and Mendel first proposed the linear algebra solution by using the correlation functions and the third-order cumulants in [4], which is now known as the GM approach. This method, however, does not

Manuscript received January 14, 1999. This work was supported by the Centre for Multimedia Signal Processing, Department of Electronic and Information Engineering, the Hong Kong Polytechnic University. The associate editor coordinating the review of this paper and approving it for publication was Dr. Shubha Kadambe.

The authors are with the Centre for Multimedia Signal Processing, Department of Electronic and Information Engineering, the Hong Kong Polytechnic University, Kowloon, Hong Kong (e-mail: enliwei@en.polyu.edu.hk; enwcsiu@hkpucc.polyu.edu.hk).

Publisher Item Identifier S 1053-587X(00)02353-9.

work well when measurement noise, even if it is white Gaussian noise, is present. Since the correlation functions are not blind to Gaussian noise, some equations affected by the measurement noise have to be removed, resulting in a set of underdetermined equations. In addition, numerical ill conditioning may exist in the GM approach when ε ($\varepsilon = \sigma_u^2/\gamma_{3u}$, where σ_u and γ_{3u} denotes the second- and the third-order statistics, respectively.) is "small." To avoid the above problems, some modified versions [5]–[7] of the GM approach and some new algorithms in [2], [6], and [8] have been suggested. For example, in [5], a new method called the T-approach was proposed by Tugnait, which also makes use of the correlation functions and the third-order cumulants. The main difference between the T and the GM-approaches is that different lags in third-order cumulants are taken. Two modified GM-based algorithms developed by Tugnait called the GMT1 and the GMT2 have been suggested in [6], and [7]. The aims are to make the GM approach able to deal with noisy observations. The common point is that both algorithms remove the equations corrupted by the measurement noise from the corresponding GM equations. The difference is that some equations derived from bispectral are used as auxiliary equations in the GMT1 approach, whereas some equations derived from T approach are used as auxiliary equations in the GMT2 approach. Extensions from the third-order to the fourth-order cumulants have also been suggested by Tugnait in [6] and [7], in order to accommodate the signal with symmetrical distribution. The R-GM approach as shown in [5] succeeded in averting the numerical ill conditioning due to small ε in the GM approach with $\varepsilon' = 1/\varepsilon$. Note that all algorithms in [3]–[7] make use of the correlation and the third-order (or fourth-order) cumulants, which are known as the R-C algorithms. Although some of them considered the measurement noise, they are designed only for i.i.d. non-Gaussian noise or white Gaussian noise.

For cases with relatively complicated measurement noise such as the ARMA Gaussian noise, using cumulants with orders greater than two may be a good choice for FIR system identification. A few algorithms in [2] and [9], which make use of the third- and the fourth-order cumulants, have achieved FIR system identification under the ARMA Gaussian noise environment. However, almost all the above-mentioned algorithms dealt with cases on resolving nonlinear equations due to the unknown parameters $\{h(k)\}$ and $\{h^2(k)\}$ for $k = 1, 2, \ldots q$. A commonly used treatment is to find the least-square solutions by supposing that $\{h(k)\}$ and $\{h^2(k)\}$ are independent at the beginning, although this assumption is not reasonable and may generate a larger estimation error, and then to obtain final estimate for each unknown $\{h(k)\}$ by considering the estimates of $\{h(k)\}$ and $\{h^2(k)\}$ together, which is generally called postprocessing. Postprocessing is, to some extent, just a compensating means for the fault resulting from misunderstanding $\{h(k)\}$ and $\{h^2(k)\}$ being independents.

To avoid the additional step, i.e., the postprocessing, and, meanwhile, to accommodate a complicated noisy environment, we will make use of output cumulants with arbitrarily selected orders and propose two new *linear* formulations, namely, Method 1 and Method 2. By selecting different orders of cumulants, the formulations in the two methods lead to a series of algorithms. To demonstrate how to derive the algorithms from the formulations, we conduct two particular algorithms, namely, Algorithm 1 and Algorithm 2, by using two sets of fixed orders. Our methods have the following features:

1) Flexibility: Method 1 employs two cumulants of $y(k)$ with arbitrary adjacent orders, whereas Method 2 uses three cumulants of $y(k)$: two cumulants with arbitrary orders and one with an order equal to the summation of the two orders minus one. For clarity, we select a set of orders in each method as an implementation of the corresponding method. They form Algorithm 1, for using the third- and the fourth-order cumulants, and Algorithm 2, for using the second-, third-, and fourth-order cumulants.
2) Suitability: As we have mentioned, by selecting different orders, Methods 1 and 2 can lead to a number of algorithms. These algorithms can be applied in different situations such as the systems with minimum phase (MP) or nonminimum phase (NMP) and the systems with white Gaussian noise or colored (MA or ARMA) Gaussian noise. For example, Algorithm 1 can serve for NMP system even under ARMA Gaussian noise, and Algorithm 2 is available for NMP system under white Gaussian noise only. This, however, does not imply that Method 2, from which Algorithm 2 is derived by selecting the second-, third-, and fourth-order cumulants, is largely confined. We can actually select other orders in Method 2 instead of the second- , third-, and fourth-order shown in Algorithm 2 to accommodate certain complicated situations.
3) Linearity: Both the formulations in Method 1 and Method 2 are linear with respect to the unknowns. The postprocessing step is thus avoided.

II. Two New Methods for FIR System Identification

Let us recall (1) and (2)

$$x(k) = \sum_{i=0}^{q} h(i)u(k-i)$$
$$y(k) = x(k) + n(k).$$

Before we elaborate on our approach, let us make the following assumptions.

1) The driving signal $u(k)$ is unobservable i.i.d. non-Gaussian process with zero-mean.
2) The system is of nonminimum phase, where $h(0) = 1$, $h(q) \neq 0$, and $h(i) = 0$ for $i < 0$ and $i > q$. The system order q is known or may be measured by *a priori* knowledge [14]–[16].
3) The measurement noise $n(k)$ is white Gaussian noise or Gaussian ARMA process with unknown statistics and is independent of $u(k)$; hence, it is also independent of the output $x(k)$.

A. New Formulation

In order to use higher order cumulants of the output for identifying the system described by (1) and (2), or particularly estimating those coefficients in (1), we need to establish a relationship between the output higher order cumulants and the coefficients.

Consider (1) and the assumption of i.i.d on $u(k)$. The mth-order and $(m+n)$th-order cumulants of $x(k)$ may be expressed in terms of the FIR coefficients [12] as

$$\begin{aligned} c_m^x(i_1, i_2, \ldots, i_{m-1}) &= E\{x(k)x(k+i_1)\ldots x(k+i_{m-1})\} \\ &= \gamma_m^u \sum_{k=0}^{q} h(k)h(k+i_1)\ldots h(k+i_{m-1}) \end{aligned} \tag{3}$$

and

$$\begin{aligned} c_{m+n}^x(i_1, i_2, \ldots, i_{m+n-1}) \\ = \gamma_{m+n}^u \sum_{k=0}^{q} h(k)h(k+i_1)\ldots h(k+i_{m+n-1}) \end{aligned} \tag{4}$$

where m and n are two positive integers, and γ_m^u denotes the mth-order cumulant of $u(k)$.

Let $i_{m-1} = q$ in (3), which gives

$$\begin{aligned} c_m^x(i_1, i_2, \ldots, i_{m-2}, q) &= \gamma_m^u \sum_{k=0}^{q} h(k)h(k+i_1)\ldots h(k+i_{m-2})h(k+q) \\ &= \gamma_m^u h(0)h(i_1)\ldots h(i_{m-2})h(q). \end{aligned} \tag{5}$$

Similarly, for $i_{m+n-1} = q$, (4) becomes

$$\begin{aligned} c_{m+n}^x(i_1, i_2, \ldots, i_{m+n-2}, q) &= \gamma_{m+n}^u \sum_{k=0}^{q} h(k)h(k+i_1)\ldots h(k+i_{m+n-2})h(k+q) \\ &= \gamma_{m+n}^u h(0)h(i_1)\ldots h(i_{m-2})h(i_{m-1})\ldots h(i_{m+n-2}) \\ &\quad \times h(q) \end{aligned} \tag{6}$$

where we have used $h(i) = 0$ for $i > q$.

Combining the last two equations and setting $\varepsilon = (\gamma_m^u / \gamma_{m+n}^u)(\gamma_k^u \neq 0)$, we obtain the following ratio:

$$\begin{aligned} \frac{c_{m+n}^x(i_1, i_2, \ldots, i_{m-2}, i_{m-1}, \ldots, i_{m+n-2}, q)}{c_m^x(i_1, i_2, \ldots, i_{m-2}, q)} \\ = \frac{1}{\varepsilon} h(i_{m-1})h(i_m)\ldots h(i_{m+n-2}). \end{aligned} \tag{7}$$

Obviously, the relationship between the unknown coefficients and the output cumulants are now expressed by this equation. Because the cumulants of $x(k)$ at a fixed lag q on the left-hand side of the equation can be computed by the sample average, all the unknowns involved on the right-hand side of the equation can be solved by some linear or nonlinear methods that depend on whether nonlinear terms related to $h(i)$ exist. By selecting appropriate values for m and n, that is, by using different order cumulants, we are able to attain different expressions derived from (7). Therefore, this equation provides a basic relationship to identify the FIR system by using arbitrary order cumulants. Based on this formulation, we will present two methods.

B. Method 1

Let us consider a simple case by selecting $n = 1$ so that only one $h(i_{m-1})$ is preserved on the right-hand side of (7) and two cumulants with adjacent orders are left on the left-hand side of the equation. Equation (7) thus becomes

$$\begin{aligned} \frac{c_{m+1}^x(i_1, i_2, \ldots, i_{m-2}, i_{m-1}, q)}{c_m^x(i_1, i_2, \ldots, i_{m-2}, q)} = \frac{1}{\varepsilon} h(i_{m-1}) \\ 0 \le i_k \le q \quad \text{and} \quad k = 1, 2, \ldots, m-1. \end{aligned} \tag{8}$$

Equation (8) is in a form similar to the solution in [4]. However, they are, in fact, inherently different. Giannakis and Mendel [4] provide an algebraic solution by using one equation that cannot smooth out the effect of the additive noise generated from estimating the cumulants. However, from (8), we may obtain a set of equations by choosing different values for $i_1, \ldots, i_{m-2}, i_{m-1}$. To estimate all the coefficients $h(0), \ldots, h(q)$ and to make the cumulants meaningful, we may naturally let i_{m-1} be $0, 1, \ldots, q$ in the sequence and take the range $0 \le i_k \le q$ (for $k = 1, 2, \ldots, m-2$) for every i_{m-1}. For a fixed i_{m-1}, and $0 \le i_k \le q$ (for $k = 1, 2, \ldots, m-2$), (8) yields a system with only a single unknown and $(q+1)^{m-2}$ linear equations. The linear least-square solutions for unknown ε and $\{h(k)\ k = 1, \ldots, q\}$ could thus be obtained one after another.

For clarity, we take $m = 3$ as an example. For $m = 3$, (8) becomes

$$\frac{c_4^x(i_1, i_2, q)}{c_3^x(i_1, q)} = \frac{1}{\varepsilon} h(i_2) \quad 0 \le i_1, \quad i_2 \le q. \tag{9}$$

Accordingly, fixing i_2 and taking i_1 from 0 to q, we have $q+1$ equations with only one unknown for each i_2. Applying the least-square principle, ε and $\{h(i_2)$ for $i_2 = 1, \ldots, q\}$ are obtained. For this particular situation, let us call it Algorithm 1. We summarize Algorithm 1 as follows.

1) Initially, let $i_2 = 0$ and take $i_1 = 0, 1, \ldots, q$; this leads to $(q+1)$ equations. In matrix form, we have

$$A_0 \varepsilon = c_0$$

where

$$\begin{aligned} A_0 &= [c_4^x(0,0,q) \quad c_4^x(1,0,q) \quad \cdots \quad c_4^x(q,0,q)]^T \\ c_0 &= [c_3^x(0,q) \quad c_3^x(1,q) \quad \cdots \quad c_3^x(q,q)]^T \end{aligned}$$

and we have used the assumption that $h(0) = 1$.

The least-square estimation $\hat{\varepsilon}$ of ε can easily be obtained by $\hat{\varepsilon} = (A_0^T A_0)^{-1} A_0^T c_0$.

2) Let $i_2 = 1$, and take $i_1 = 0, 1, \ldots, q$. The unknown $h(1)$ in matrix form can be written as

$$A_1 h(1) = c_1$$

where

$$\begin{aligned} A_1 &= [c_3^x(0,q) \quad c_3^x(1,q) \quad \cdots \quad c_3^x(q,q)]^T \\ c_1 &= [\hat{\varepsilon} c_4^x(0,1,q) \quad \hat{\varepsilon} c_4^x(1,1,q) \quad \cdots \quad \hat{\varepsilon} c_4^x(q,1,q)]^T. \end{aligned}$$

We can then estimate $\hat{h}(1)$ of $h(1)$ by applying the least-square principle.

3) Similarly, repeating step 2) for $i_2 = 2, 3, \ldots, q$, respectively, we can solve other unknowns $h(2), h(3), \ldots, h(q)$ with the least-square solutions.

It is easy to see that Algorithm 1 makes use of the fixed $(q+1)$ slices of the third-order cumulants at one lag q as well as the $(q+1)$ slices of the fourth-order cumulants at lags q and i_2 in each step. These slices form the vectors A_k and c_k ($k = 0, 1, \ldots, q$). As long as a nonzero element exists in A_k, we can obtain an unique lest-square solution for every unknown parameter. In fact, from the first assumption and the well-known Brillinger–Rosenblatt formula in [12], these cumulants should be nonzero. Generally, the cumulants are needed to be calculated by output sample average *a prior*. In theory, if the length N of samples is very large, i.e., $N \to \infty$, the sampled cumulants converge with probability one to the true cumulants [6]. In practice, only some samples of the output are known, and we have to make use of these finite samples to estimate its cumulants to replace the "true" cumulants. This certainly leads to some errors (or noises) on computing the cumulants.

Algorithm 1 is derived without considering the measurement noise. In the presence of the measurement noise, the observations are considered to be noisy. As we know, nth-order cumulants of Gaussian processes vanish for $n \geq 3$. If the measurement noise is white Gaussian or even colored (MA or ARMA) Gaussian noises, Algorithm 1 is still available by directly replacing $c_3^x(i_1, q)$ and $c_4^x(i_1, i_2, q)$ with $c_3^y(i_1, q)$ and $c_4^y(i_1, i_2, q)$, respectively, without any changes.

Algorithm 1, by taking $m = 3$, is an example of Method 1. Of course, we can assign other values for m. A number of algorithms are therefore generated. However, a special solution occurs when $m = 2$, which is a closed-form solution that is very similar to the $c(q, k)$ approach in [4] and is obtained by following the above steps. Obviously, this is not a solution and, hence, cannot smooth out the additive noise generated by calculating the cumulants. We suggest that the value of m should be larger than 2 for actual implementation. Similarly, Method 1, which makes use of two adjacent-order cumulants, is the simplest case of (7) by taking $n = 1$. If we take other values for n in (7), the orders of the two cumulants will not be adjacent, and nonlinearity will appear on the right side of the corresponding equation. For the sake of simplicity, we suggest that $n = 1$.

The novelty of Method 1 lies in that only one unknown is estimated via the linear least-square principle in every step. It, on one hand, avoids the postprocessing procedure due to nonlinearity. On the other hand, it is not only able to smooth out the noises due to calculating the cumulants but is also able to avoid the error propagation due to the inaccuracy of the other estimations. However, the estimates of $\{h(k),\ \text{for } k = 1, 2, \ldots, q\}$ are dependent on the estimate of ε. It implies that if the estimation of ε is incorrect, the estimates of $h(1), \ldots h(q)$ are certainly not accurate. This problem is also present in almost all related algorithms.

C. Method 2

We have derived Algorithm 1 by evaluating $n = 1$ and $m = 3$ in (7). Of course, if we take different values for n and m, a large number of algorithms can also result. In addition, the common point of these algorithms is that only the cumulants with two adjacent-order (the case of $n = 1$) or two arbitrary order (other values for n) are used to perform the estimation of the FIR system. Additionally, only one unknown is estimated in every step for this algorithm. For comparison purposes, we attempt to design another algorithm by using more cumulants to obtain the estimates of all the unknowns simultaneously. To do this, the $(n+1)$th-order cumulant of $x(k)$ is suggested as

$$c_{n+1}^x(j_1, j_2, \ldots, j_n) = \gamma_{n+1}^u \sum_{l=0}^{q} h(l)h(l+j_1)\ldots h(l+j_n). \tag{10}$$

Our idea is to try to express $h(l+j_1)h(l+j_2)\ldots h(l+j_n)$ by certain quantities that may be known or may be computed to leave $h(l)$ alone. Let us set $l + j_1 = i_{m-1}, l + j_2 = i_m, \ldots, l + j_n = i_{m+n-2}$, and substitute them into (7); we then have

$$\varepsilon \frac{c_{m+n}^x(i_1, i_2, \ldots, i_{m-2}, l+j_1, \ldots, l+j_n, q)}{c_m^x(i_1, i_2, \ldots, i_{m-2}, q)} = h(l+j_1)h(l+j_2)\ldots h(l+j_n). \tag{11}$$

Combining (10) and (11), we obtain

$$\begin{aligned} c_{n+1}^x(j_1, j_2, \ldots, j_n) &= \varepsilon\gamma_{n+1}^u \sum_{l=0}^{q} h(l) \\ &\times \frac{c_{m+n}^x(i_1, i_2, \ldots, i_{m-2}, l+j_1, \ldots l+j_n, q)}{c_m^x(i_1, i_2, \ldots, i_{m-2}, q).} \end{aligned} \tag{12}$$

Another form of (12) is

$$\begin{aligned} \sum_{l=0}^{q} & c_{m+n}^x(i_1, i_2, \ldots, i_{m-2}, l+j_1, \ldots, l+j_n, q)h(l) \\ &= e c_{n+1}^x(j_1, j_2, \ldots, j_n) c_m^x(i_1, i_2, \ldots, i_{m-2}, q) \end{aligned} \tag{13}$$

where

$$e = \frac{1}{\varepsilon\gamma_{n+1}^u} = \frac{\gamma_{m+n}^u}{\gamma_m^u \gamma_{n+1}^u}.$$

With $h(0) = 1$, we rewrite (13) as

$$\begin{aligned} \sum_{l=1}^{q} & c_{m+n}^x(i_1, i_2, \ldots, i_{m-2}, l+j_1, \ldots, l+j_n, q)h(l) \\ &- e c_{n+1}^x(j_1, j_2, \ldots, j_n) c_m^x(i_1, i_2, \ldots, i_{m-2}, q) \\ &= -c_{m+n}^x(i_1, \ldots, i_{m-2}, j_1, \ldots, j_n, q). \end{aligned} \tag{14}$$

Equation (14) has, when it is viewed as a linear equation, $(q+1)$ unknowns involving $\{h(k)\ k = 1, 2, \ldots, q\}$ and e. By selecting appropriate values for m and n and taking all meaningful values for i_{k_1} and j_{k_2} ($k_1 = 1, \ldots, m-2$ and $k_2 = 1, \ldots, n$), a set of overdetermined equations is formed. The least-square solution to the unknowns can then be obtained. Similar to Algorithm 1, let us clarify our approach by giving a simple example with $m = 3$ and $n = 1$ for Algorithm 2.

Evaluating (14) at $m = 3$ and $n = 1$, we have

$$\begin{aligned} \sum_{l=1}^{q} & c_4^x(i_1, l+j_1, q)h(l) - e c_2^x(j_1) c_3^x(i_1, q) \\ &= -c_4^x(i_1, j_1, q). \end{aligned} \tag{15}$$

$$A_k = \begin{bmatrix} c_4^x(0,k+1,q) & c_4^x(0,k+2,q) & \cdots & c_4^x(0,k+q-1,q) & c_4^x(0,k+q,q) & -c_2^x(k)c_3^x(0,q) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ c_4^x(q,k+1,q) & c_4^x(q,k+2,q) & \cdots & c_4^x(q,k+q-1,q) & c_4^x(q,k+q,q) & -c_2^x(k)c_3^x(q,q) \end{bmatrix}$$

$$\mathbf{c}_k = [-c_4^x(0,k,q) \quad -c_4^x(1,k,q) \quad \cdots \quad -c_4^x(q,k,q)]^T$$

Obviously, three cumulants with different orders appear in this equation. Let us set $0 \le i_1 \le q$ and $-q \le j_1 \le q$ in (15). We have $q+1$ unknowns, i.e., $h(1), \ldots, h(q)$, e, and $(2q+1)(q+1)$ equations. We rewrite (15) in matrix form as (for $0 \le i_1 \le q$ and $-q \le j_1 \le q$)

$$A\mathbf{b} = \mathbf{c} \tag{16}$$

where

$$\begin{aligned} \mathbf{b} &= [h(1) \quad \cdots \quad h(q) \quad e]^T \\ &\quad \text{a } (q+1)\text{-column vector} \\ A &= \begin{bmatrix} A_{-q}^T & A_{-q+1}^T & \cdots & A_q^T \end{bmatrix}^T \\ &\quad \text{a } (2q+1)(q+1) \times (q+1) \text{ matrix} \\ \mathbf{c} &= \begin{bmatrix} \mathbf{c}_{-q}^T & \mathbf{c}_{-q+1}^T & \cdots & \mathbf{c}_q^T \end{bmatrix}^T \\ &\quad \text{a } (2q+1)\text{-column vector.} \end{aligned}$$

The symbol "T" denotes the transpose of the matrix or the vector. A_k is a $(q+1), \times (q+1)$ matrix, and $\mathbf{c}_k$ is a $(q+1)$ column vector for $j_1 = k$ $(k = -q, -q+1, \ldots, q)$ and $0 \le i_1 \le q$, shown in the expression at the top of the page, where $c_4^x(0,i,q) = 0$ for $i < 0$ and $i > q$.

A and $\mathbf{c}$ are expressed as in the equations at the bottom of the page. We can find that matrixes A and $\mathbf{c}$ utilize all meaningful autocorrelation functions of $x(k)$ and all third-order and fourth-order cumulants of $x(k)$ at a fixed lag q. As we have pointed out in Algorithm 1, as long as we have enough output samples, we can get very accurate estimation of these statistics. From (16), it is easy to see that A, $\mathbf{c}$, and $\mathbf{b}$ constitute a set of overdetermined *linear* equations. Naturally, we adopt the linear least-square principle to solve it. Because matrix A has full rank $(q+1)$, the above overdetermined equations have a unique and accurate least-squares solution as in

$$\hat{b} = (A^T A)^{-1} A^T c. \tag{17}$$

$$\mathbf{c} = [0 \quad \cdots \quad 0 \quad -c_4^x(0,0,q) \quad \cdots \quad -c_4^x(q,0,q) \quad -c_4^x(0,1,q) \cdots \quad -c_4^x(q,1,q) \quad \cdots \quad -c_4^x(0,q,q) \quad \cdots \quad -c_4^x(q,q,q)]^T$$

$$A = \begin{bmatrix} 0 & 0 & \cdots & 0 & c_4^x(0,0,q) & -c_2^x(-q)c_3^x(0,q) \\ \vdots & \vdots & \cdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & c_4^x(q,0,q) & -c_2^x(-q)c_3^x(q,q) \\ 0 & 0 & \cdots & c_4^x(0,0,q) & c_4^x(0,1,q) & -c_2^x(-q+1)c_3^x(0,q) \\ \vdots & \vdots & \cdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & c_4^x(q,0,q) & c_4^x(q,1,q) & -c_2^x(-q+1)c_3^x(q,q) \\ \vdots & \vdots & \cdots & \vdots & \vdots & \vdots \\ c_4^x(0,0,q) & c_4^x(0,1,q) & \cdots & c_4^x(0,q-2,q) & c_4^x(0,q-1,q) & -c_2^x(-1)c_3^x(0,q) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ c_4^x(q,0,q) & c_4^x(q,1,q) & \cdots & c_4^x(q,q-2,q) & c_4^x(q,q-1,q) & -c_2^x(-1)c_3^x(q,q) \\ c_4^x(0,1,q) & c_4^x(0,2,q) & \cdots & c_4^x(0,q-1,q) & c_4^x(0,q,q) & -c_2^x(0)c_3^x(0,q) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ c_4^x(q,1,q) & c_4^x(q,2,q) & \cdots & c_4^x(q,q-1,q) & c_4^x(q,q,q) & -c_2^x(0)c_3^x(q,q) \\ c_4^x(0,2,q) & c_4^x(0,3,q) & \cdots & c_4^x(0,q,q) & 0 & -c_2^x(1)c_3^x(0,q) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ c_4^x(q,2,q) & c_4^x(q,3,q) & \cdots & c_4^x(q,q,q) & 0 & -c_2^x(1)c_3^x(q,q) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & -c_2^x(q)c_3^x(0,q) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & -c_2^x(q)c_3^x(q,q) \end{bmatrix}$$

Note that all cumulants in A and $\mathbf{c}$ are expressed in noise-free output $x(k)$. In the presence of the measurement noise, matrix A and vector $\mathbf{c}$ have to be written in terms of the cumulants of $y(k)$. If the noise is white Gaussian, it is true that $c_2^y(i_1) = c_2^x(i_1)$ for $i_1 \neq 0$ and $c_3^y(i_1, q) = c_3^x(i_1, q)$ for any i_1 as well as $c_4^y(i_1, i_2, q) = c_4^x(i_1, i_2, q)$ for any i_1, i_2, where only $c_2^y(0)$ is corrupted by the noise. Removing the rows that include $c_2^y(0)$ in A and $\mathbf{c}$, Algorithm 2 is readily available to estimate the unknowns.

From (17), the $(q+1)$ unknowns are simultaneously obtained, and we refer it to as Algorithm 2. In addition, there also exist other least-square solutions by using subsystems $(A_k\mathbf{c}_k)$. We describe one of these possible solutions as follows.

Step 1) Let $k = q$ for the subsystem $(A_q\mathbf{c}_q)$ consisting of $q+1$ equations with only one unknown e. The least-squares solution is then easily obtained.

Step 2) Let $k = q-1$ (or $-q$). Since e has been estimated in step 1, the subsystem $(A_{q-1}\mathbf{c}_{q-1})$ or $(A_{-q}\mathbf{c}_{-q})$ consisting of $q+1$ equations involves one unknown $h(1)$ [or $h(q)$] only. We can then obtain the least-squares solution of $h(1)$ [or $h(q)$].

Step 3) Similarly, let $k = q-2, \ldots, 1, -1$ (Note that for the noisy case, $k = 0$ is not included) or $k = -q+1, -q+2, \ldots, -1$, respectively. Utilizing the estimated parameters and repeating step 2, the least-squares solutions of the other unknowns can also be obtained recursively.

It should be emphasized that Algorithm 2 is just an example of Method 2 by taking $m = 3$ and $n = 1$, and it is only available for the white Gaussian noise due to using the second-order statistics. To accommodate a harsh environment, we can select other values for m and n and thus obtain a number of algorithms. For example, setting $n = 2$ and $m = 3$ in (14), the third- and fifth-order cumulants will be employed. The produced algorithm can be applied to the environment with the ARMA Gaussian noise.

We have proposed two methods. One is based on (8), requiring two different order cumulants c_{m+n}^x and c_m^x, whereas the other one is based on (14), requiring three different order cumulants c_{m+n}^x, c_{n+1}^x and c_m^x. Because both methods deal with solving linear equations, the postprocessing is avoided. For the sake of clarification, we have evaluated two particular values for m and n and, hence, generated two algorithms. From the point of statistics, it seems that if we use as many output cumulants as possible, the estimation results should be better. According to this understanding, Algorithm 2 seems to perform the estimation over Algorithm 1. To compare the performance of the proposed algorithms as well as some existing algorithms, we will give some experimental results in the next section.

III. Simulation Results

Two new methods for identifying FIR system have been theoretically proposed. To verify the availability of the methods, we have carried out a number of experiments with third-order and fourth-order FIR systems under the ARMA Gaussian noise and three real noises. We are especially interested in comparing the performance of the new algorithms with that of the existing R-C approaches such as the GM approach [4], the T approach [6], the R-GM approach [5], the GMT1 method [6], the GMT2 method [7], and the methods based on higher order cumulants alone [2]. All the second- , third-, and fourth-order cumulants that might be used in these algorithms are computed first. Each algorithm employs parts of these cumulants, according to the requirements in the corresponding algorithm. Note that some of the cumulants with the same order in these algorithms might be the same, but some of them are different.

TABLE I
Estimated Parameters of Example 1 under the ARMA Gaussian Noise (30 Monte Carlo Runs, $N = 2048$ for Each Run)

Approach		SNR=0dB			SNR=10dB			SNR=20dB		
	Real	h(1)	h(2)	h(3)	h(1)	h(2)	h(3)	h(1)	h(2)	h(3)
	values	0.900	0.385	-0.771	0.900	0.385	-0.771	0.900	0.385	-0.771
Algorithm 2	Mean	0.295	0.368	-0.090	0.828	0.418	-0.659	0.764	0.393	-0.703
	Std. Dev.	0.020	0.003	0.001	0.022	0.016	0.011	0.020	0.010	0.010
Algorithm 1	Mean	0.866	0.394	-0.688	0.910	0.433	-0.773	0.897	0.408	-0.772
	Std. Dev.	0.011	0.017	0.043	0.009	0.022	0.013	0.014	0.009	0.016
GM [4]	Mean				0.753	0.155	-0.523	0.839	0.144	-0.604
	Std. Dev.	*	*	*	0.027	0.004	0.005	0.027	0.001	0.007
R-GM [5]	Mean	0.066	0.749	-0.193	0.763	0.436	-0.668	0.894	0.386	-0.774
	Std. Dev.	0.020	0.024	0.004	0.034	0.004	0.005	0.020	0.004	0.006
Tugnait [6]	Mean	0.426	0.391	-0.114	0.881	0.386	-0.678	0.889	0.416	-0.786
	Std. Dev.	0.047	0.003	0.001	0.017	0.008	0.010	0.012	0.012	0.012
GMT1 [6]	Mean				0.707	0.114	-0.434	0.814	0.162	-0.593
	Std. Dev.	*	*	*	0.042	0.012	0.017	0.041	0.004	0.018
GMT2 [7]	Mean	0.290	0.370	-0.097	0.862	0.356	-0.701	0.888	0.410	-0.782
	Std. Dev.	0.024	0.004	0.001	0.017	0.010	0.009	0.013	0.012	0.011
Method 1[2]	Mean	1.037	0.637	-0.755	0.955	0.432	-0.781	0.951	0.444	-0.802
	Std. Dev.	0.122	0.168	0.309	0.026	0.023	0.014	0.027	0.015	0.014
Method 2[2]	Mean	1.001	0.825	-0.803	0.982	0.746	-0.918	0.980	0.756	-0.927
	Std. Dev.	0.011	0.030	0.227	0.003	0.007	0.002	0.003	0.005	0.002

Note: the symbol "*" means the values are too large to accept.

To illustrate this comparison, we give three examples. For each test case, the input $u(k)$ is a normalized, i.i.d. exponential random sequence with zero mean. Others like $x(k), n(k)$, and $y(k)$ have the same meanings given in (1) and (2). To measure the strength of the noise, we define the signal-to-noise ratio as $\text{SNR} = 10\log_{10}(||u(k)||^2/||n(k)||^2)$ (dB). To measure the accuracy of parameter estimation with respect to the real values, we define the mean squared error (MSE) for each run as

$$\text{MSE} = \left(\sum_{i=1}^{q} (h(i) - \hat{h}(i))^2 \Bigg/ \sum_{i=1}^{q} h(i)^2 \right)^{1/2} \tag{18}$$

where $\{\hat{h}(i)\}$ are the estimated parameters in each run, and $\{h(i)\}$ are the real parameters in the model. Tables I–III show the original results of the means and the average standard deviations (Std. Dev.) of the estimated parameters, with sample length $N = 2048$, over 30 Monte Carlo runs for each approach. For clarity of expression and vision, we calculate the corresponding MSE's of these estimation results and draw them in Figs. 1–3, respectively.

Example 1: The third-order NMP MA model taken from [2]–[4] is

$$\begin{aligned} x(k) = u(k) + 0.9u(k-1) + 0.385u(k-2) \\ - 0.771u(k-3) \end{aligned} \tag{19}$$

with zeroes at $-0.75 \pm j0.85$ and 0.6. With reference to (1), in this case, we have $h(0) = 1, h(1) = 0.9, h(2) =$

TABLE II
ESTIMATED PARAMETERS OF EXAMPLE 2 UNDER THE ARMA GAUSSIAN NOISE (30 MONTE CARLO RUNS, $N = 2048$ FOR EACH RUN)

Approach	Real values	SNR=0dB h(1) -0.800	h(2) 1.520	h(3) -0.640	h(4) 0.990	SNR=10dB h(1) -0.800	h(2) 1.520	h(3) -0.640	h(4) 0.990	SNR=20dB h(1) -0.800	h(2) 1.520	h(3) -0.640	h(4) 0.990
Algorithm 2	Mean	0.286	1.355	-0.065	0.839	-0.550	1.224	-0.514	0.813	-0.691	1.411	-0.681	0.845
	Std. Dev.	0.126	0.508	0.257	0.215	0.144	0.469	0.154	0.140	0.114	0.459	0.122	0.060
Algorithm 1	Mean	-0.788	1.421	-0.616	0.839	-0.696	1.424	-0.455	0.813	-0.729	1.421	-0.531	0.845
	Std. Dev.	0.141	0.284	0.167	0.215	0.116	0.235	0.159	0.140	0.057	0.137	0.077	0.060
GM [4]	Mean	-0.154	1.787	0.032	0.753	-0.693	2.818	-0.438	1.121	-0.657	2.818	-0.391	0.959
	Std. Dev.	0.226	1.625	0.195	0.301	0.046	2.162	0.348	0.402	0.241	1.778	0.106	0.090
R-GM [5]	Mean	-0.502	1.341	-0.456	0.843	-0.912	1.723	-0.798	1.051	-0.857	1.694	-0.669	0.960
	Std. Dev.	0.084	0.105	0.114	0.088	0.075	0.351	0.203	0.101	0.077	0.261	0.065	0.020
Tugnait [6]	Mean	0.031	1.368	-0.049	0.830	-0.636	1.480	-0.459	0.945	-0.686	1.566	-0.542	0.960
	Std. Dev.	0.204	0.273	0.035	0.127	0.015	0.106	0.025	0.022	0.056	0.118	0.024	0.014
GMT1 [6]	Mean	0.192	0.834	0.160	0.999	-0.625	2.454	-0.279	0.958	-0.642	2.545	-0.425	0.928
	Std. Dev.	1.511	74.76	2.331	2.047	0.029	1.036	0.036	0.077	0.115	1.229	0.022	0.048
GMT2 [7]	Mean	-0.030	1.353	-0.025	0.892	-0.671	1.511	-0.445	0.971	-0.715	1.569	-0.538	0.977
	Std. Dev.	0.121	0.262	0.043	0.084	0.011	0.131	0.031	0.017	0.049	0.129	0.026	0.009
Method 1[2]	Mean	-1.090	1.749	-0.932	1.042	-0.847	1.490	-0.662	0.949	-0.785	1.619	-0.863	1.197
	Std. Dev.	1.249	0.836	1.039	0.233	0.048	0.097	0.091	0.050	0.040	0.288	0.491	0.296
Method 2[2]	Mean	-0.982	1.180	-0.869	0.957	-0.938	1.137	-0.851	0.977	-0.915	1.162	-0.908	1.044
	Std. Dev.	0.040	0.028	0.116	0.112	0.006	0.006	0.017	0.006	0.007	0.014	0.038	0.018

TABLE III
ESTIMATED PARAMETERS OF EXAMPLE 1UNDER REAL NOISES AT SNR OF 0 dB (30 MONTE CARLO RUNS, $N = 2048$ FOR EACH RUN)

Approach	Real values	'Drum' h(1) 0.900	h(2) 0.385	h(3) -0.771	'Car' h(1) 0.900	h(2) 0.385	h(3) -0.771	'Engine' h(1) 0.900	h(2) 0.385	h(3) -0.771
Algorithm 2	Mean	0.254	0.352	-0.072	0.481	0.365	-0.278	0.474	0.471	-0.404
	Std. Dev.	0.023	0.002	0.001	0.040	0.007	0.005	0.036	0.016	0.009
Algorithm 1	Mean	0.869	0.377	-0.777	0.871	0.395	-0.711	0.895	0.439	-0.714
	Std. Dev.	0.013	0.014	0.019	0.018	0.032	0.027	0.015	0.034	0.039
GM [4]	Mean	*	*	*	*	*	*	0.691	0.362	-0.605
	Std. Dev.							0.020	0.032	0.008
R-GM [5]	Mean	*	*	*	*	*	*	0.757	0.502	-0.702
	Std. Dev.							0.074	0.019	0.006
Tugnait [6]	Mean	0.471	0.394	-0.071	0.367	0.351	-0.074	0.601	0.471	-0.464
	Std. Dev.	0.048	0.005	0.001	0.050	0.002	0.001	0.033	0.024	0.022
GMT1 [6]	Mean	*	*	*	*	*	*	0.574	-0.121	-0.312
	Std. Dev.							0.105	0.143	0.143
GMT2 [7]	Mean	*	*	*	0.744	0.589	-0.311	0.902	0.600	-0.662
	Std. Dev.				6.122	1.444	1.143	0.009	0.010	0.007
Method 1[2]	Mean	0.836	0.539	-0.951	0.869	0.496	-0.845	0.955	0.463	-0.782
	Std. Dev.	0.050	0.213	0.128	0.024	0.137	0.062	0.073	0.035	0.039
Method 2[2]	Mean	0.934	0.771	-0.969	0.951	0.760	-0.937	0.977	0.762	-0.913
	Std. Dev.	0.007	0.031	0.014	0.003	0.023	0.008	0.007	0.009	0.008

Note: the symbol "*" means the values are too large to accept.

0.385, and $h(3) = -0.771$. Let us compare the algorithms under the ARMA Gaussian noise. The ARMA Gaussian noise $n(k)$ is generated by a white Gaussian signal $w(k)$ passsing an ARMA$(3, 1)$ process given in [4]

$$n(k) - 2.2n(k-1) + 1.77n(k-2) - 0.52n(k-3) = w(k) + 1.25w(k-1) \tag{20}$$

with poles at 0.8 and $-0.7 \pm j0.4$ and with a zero -1.25.

We added the ARMA Gaussian noise to the output samples ($N = 2048$). By using the above algorithms, the simulation results for the ARMA Gaussian noise at SNR 0, 10, and 20 dB are given in Table I. By calculating the MSE, we can find that at 0, 10, and 20 dB, the percentages of MSE's of Algorithm 1 are 7.23%, 3.94% and 1.86%, respectively. For the methods [2] based on higher order cumulants alone, the best results are 23.05%, 5.86%, and 6.74% at 0, 10, and 20 dB, respectively, obtained by Method 1 in [2]. That implies an improvement of 69% for a comparison of these two figures at 0 dB. For the R-C algorithms, the best results are 65.02% obtained by the T approach at 0 dB, 6.8% obtained by the GMT2 approach at 10 dB, and 0.54% obtained by the R-GM approach at 20 dB. An improvement of 88% is achieved for a comparison of Algorithm 1 and the T-approach with SNR of 0 dB. For convenience, we draw the MSE's in Fig. 1. We only give results on our algorithms, the R-GM, the T, and the GMT2 approaches as well as the methods in [2]. The results of the other algorithms are not shown in Fig. 1 because their MSE's are so large that the precision of the comparison could be adversely affected. Fig. 1 shows that Algorithm 1 is superior to all others at lower SNR, whereas the methods in [2] take second place as they are designed for situation with the ARMA Gaussian noise.

Fig. 4 illustrates the MSE against SNR for 2048 output samples with the ARMA Gaussian noise. We give the results for our algorithms, the T, and the GMT2 approaches as well as the methods in [2]. Obviously, the curves in this figure are located

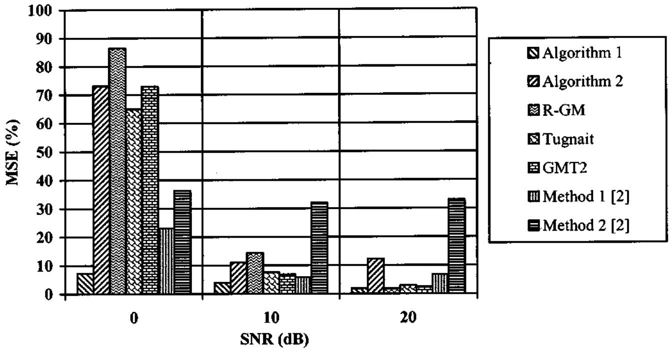

Fig. 1. MSE's of parameter estimates of Example 1 (Gaussian ARMA noise, 30 runs, $N = 2048$ in each run).

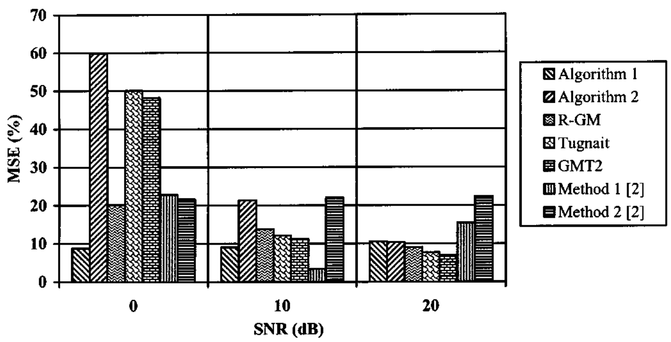

Fig. 2. MSE's of parameter estimates of Example 2 (Gaussian ARMA noise, 30 runs, $N = 2048$ in each run).

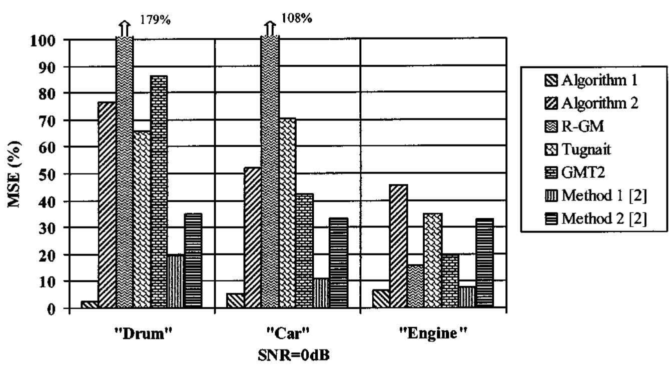

Fig. 3. MSE's of parameter estimates of Example 3 (three real noises, 30 runs, $N = 2048$ in each run).

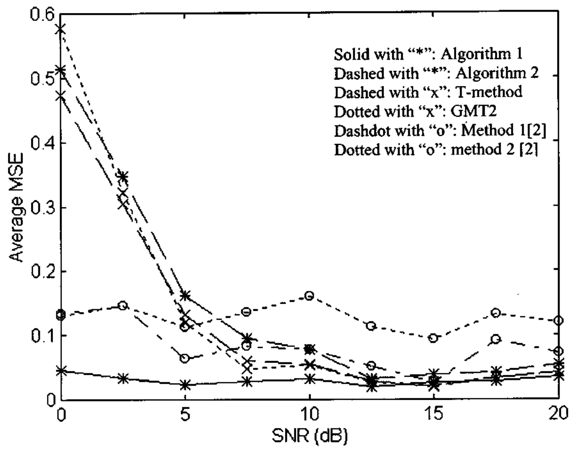

Fig. 4 Average MSE against SNR in the ARMA Gaussian Noise.

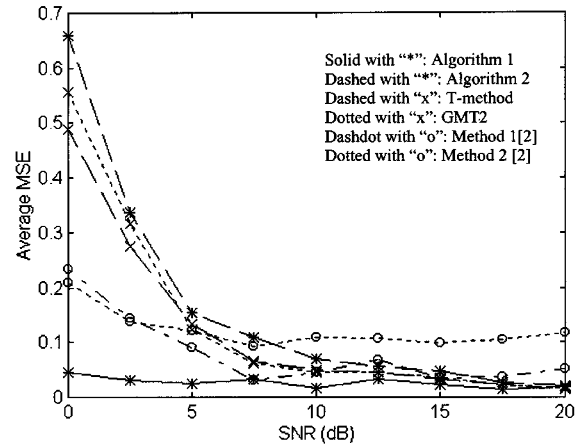

Fig. 5. Average MSE against SNR under "Drum" noise.

in three different levels. The lowest curve that corresponds to Algorithm 1 clearly shows that the MSE varies little against SNR, whereas the curves at the middle level correspond to the methods [2]. The top curves corresponding to Algorithm 2, the T, and GMT2 approaches have significant changes when the SNR varies from low to high values.

Example 2: Let us simulate the following fourth-order NMP MA model used in [9] under the above ARMA Gaussian noise

$$x(k) = u(k) - 0.8u(k-1) + 1.52u(k-2) - 0.64u(k-3) + 0.99u(k-4) \tag{21}$$

with zeros at $0.6 \pm j0.8602$ and $-0.2 \pm j0.9274$.

Table II shows the results. For clarity, we compute the corresponding MSE's and draw them in Fig. 2. It is readily observed that the overall accuracy of the parameter estimation decreases. Even so, Algorithm 1 still provides the best performance at lower SNR. For example, at 0 dB, the MSE of Algorithm 1 is 7.23%, whereas the best one among other algorithms is 20.16%. An improvement of 56% is achieved.

Example 3: Let us simulate the above third-order NMP MA model again by adding some real noises in order to test the availability of our algorithms to real environments. All real noises are recorded in the real world. They are the sounds of drum, car, and engine, respectively. The experiments in this example are carried out with SNR of 0 dB.

Let us see Table III and Fig. 3. It is easy to see that only Algorithm 1 works well under any of these real noises. The R-C algorithms such as Algorithm 2, the T, the R-GM, and the GMT2 approaches, however, have larger values of MSE's for the "drum" and "car" and have acceptable values of MSE's for the "engine," whereas the methods [2], especially Method 1 in [2], provide better results as compared with the R-C algorithms. A possible explanation is that the "drum" and "car" may be close to the colored Gaussian noise, resulting in degraded performance for the R-C algorithms. This example implies that Algorithm 1 is very robust to the real noises, which can provide potential in practical applications.

Figs. 5–7 illustrate the MSE against SNR for 2048 output samples with the three types of real noises. The curves in these figures are still in three different levels like that in Fig. 4. These

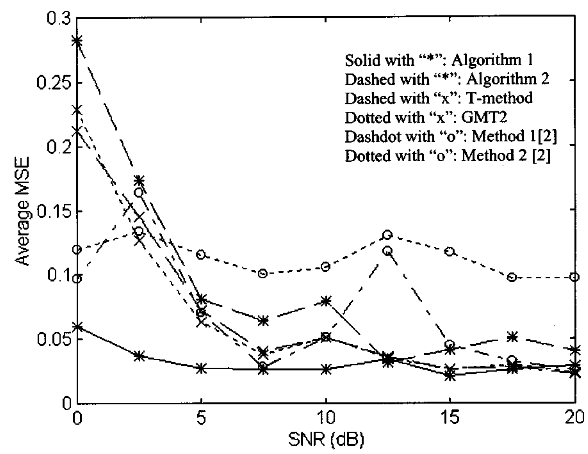

Fig. 6. Average MSE against SNR under "Car" noise.

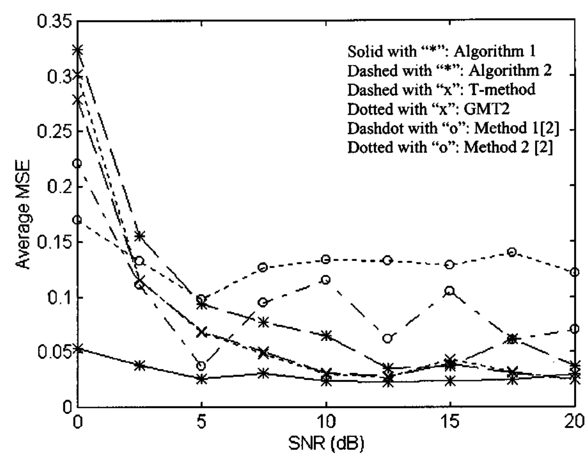

Fig. 7. Average MSE against SNR under "Engine" noise.

figures consistently show that Algorithm 1 obtains the best estimation results at lower SNR and is very insensitive to SNR as the corresponding curve has little fluctuations. Although the methods in [2] take second place in performance, the curve corresponding to Method 1 in [2] has wide fluctuations of the MSE when the SNR varies from low to high values, and Method 2 in [2] has large overall bias, although it is robust to SNR. The curves corresponding to the R-C algorithms have a sharp drop at an SNR of 5 dB or so.

We have tested our algorithms with the third- and fourth-order FIR systems under the ARMA Gaussian noise. Experimental results show that our algorithms are able to perform the parameter estimation under colored noise environment. Especially as compared with other existing algorithms with postprocessing, Algorithm 1 without postprocessing is very insensitive to the SNR. It provides comparable results at higher SNR and superior results at lower SNR (0–5 dB). Particularly at 0 dB, it improves 88% and 56% over the best results obtained from all existing algorithms in Examples 1 and 2. However, other algorithms produce better estimates only when the SNR is higher. This is not surprising. On one hand, a real linear least-squares problem is addressed in Algorithms 1 and 2, which avoids the postprocessing step and thus reduces the estimation errors. However, the postprocessing step is irreducible in other existing algorithms. On the other hand, Algorithm 1 is applicable for the ARMA Gaussian noise because of the use of the third- and fourth-order cumulants, and the estimated parameters have no interaction, resulting in no propagation of the estimation errors. Although the methods in [2] are also available for the ARMA Gaussian noise, their performances are not as good as Algorithm 1 in the estimation.

From the simulation results, Algorithm 2 does not achieve a better performance over the existing algorithms. The main reason is that it cannot theoretically suppress the ARMA Gaussian noise due to the use of second-order cumulants. To accommodate this kind of noisy environment, we can actually select the cumulants with order greater than two in Method 2. For example, setting $m = 3$ and $n = 2$ in Method 2, an Algorithm that makes use of two third-order and one fifth-order cumulants will be derived. Without doubt, it can be applied to the environment with the ARMA Gaussian noise. The computational burden, however, is largely increased.

In addition, we have also tested all the algorithms with three real noises. As a result, Algorithm 1 outperforms other algorithms.

IV. Conclusion

We have presented two new higher order cumulants-based methods for NMP FIR system identification. These methods give the foundation to form a class of algorithms by selecting different orders. Algorithms 1 and 2 are two simple realizations of the two methods. Algorithm 1 employs the third- and fourth-order output cumulants, and Algorithm 2 adopts the second- , third-, and fourth-order cumulants.

A distinct feature of the proposed algorithms lies in leaving out the postprocessing step that happens in almost all existing algorithms. Because the formulations in our algorithms are linear, the linear least-squares solutions can be directly achieved. Additionally, the flexibility of the orders selection of the methods is another advantage. We can choose appropriate orders to accommodate different environments. The higher the order is, the larger the computational burden. A compromise between the choice of order and the computational complexity is generally considered. In this paper, Algorithm 1 can serve cases with the ARMA Gaussian noise whereas Algorithm 2 is theoretically available for white Gaussian noise. Although we can develop other algorithms by selecting the cumulants with order greater than two in Method 2, the computational burden will be increased significantly.

Simulation results have shown that Algorithm 1 outperforms other published cumulant-based linear approaches and is very robust to the SNR under the ARMA Gaussian and real noises. It provides comparable results at higher SNR and superior results at lower SNR (0–5 dB). The reasons follow.

i) The algorithm avoids the postprocessing step.
ii) It only employs several slices of third- and fourth-order cumulants.
iii) It estimates one parameter in every step.

References

[1] J. M. Mendel, "Tutorial on higher-order statistics (spectra) in signal processing and system theory: Theoretical results and some applications," *Proc. IEEE*, vol. 79, pp. 278–305, Mar. 1991.

[2] X.-D. Zhang and Y.-S. Zhang, "FIR system identification using higher order statistics alone," *IEEE Trans. Signal Processing*, vol. 42, pp. 2854–2858, Oct. 1994.

[3] G. B. Giannakis, "Cumulants: A powerful tool in signal processing," *Proc. IEEE*, vol. 75, pp. 1333–1334, Sept. 1987.

[4] G. B. Giannakis and J. M. Mendel, "Identification of nonminimum phase systems using higher-order statistics," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 360–377, Mar. 1989.

[5] B. Friedlander and B. Porat, "Asympotically optimal estimation of MA and ARMA parameters of non-Gaussian processes from higher-order moments," *IEEE Trans. Automat. Contr.*, vol. 35, pp. 27–35, Jan. 1990.

[6] J. K. Tugnait, "Approaches of FIR system identification with noisy data using higher order statistics," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 38, pp. 1307–1317, July 1990.

[7] ——, "New results on FIR system identification using higher order statistics," *IEEE Trans. Signal Processing*, vol. 39, pp. 2216–2221, Oct. 1991.

[8] M. C. Carrión, D. P. Ruiz, A. Gallego, and J. A. Morente, "FIR system identification using third- and fourth-order cumulants," *Electron. Lett.*, vol. 31, pp. 612–614, Apr. 1995.

[9] Y. Xiao, M. Shadaydeh, and Y. Tadokoro, "Overdetermined $C(q, k)$ formula using third and fourth order cumulants," *Electron. Lett.*, vol. 32, pp. 601–603, Mar. 1996.

[10] J. Vidal and J. A. R. Fonollosa, "Impulse response recovery of linear systems through weighted cumulant slices," *IEEE Trans. Signal Processing*, vol. 44, pp. 2626–2632, Oct. 1996.

[11] J. A. R. Fonollosa and J. Vidal, "System identification using a linear combination of cumulant slices," *IEEE Trans. Signal Processing*, vol. 41, pp. 2405–2411, July 1996.

[12] D. R. Brillinger and M. Rosenblatt, "Computation and interpretation of kth-order spectra," in *Specrtral Analysis of Time Seriers*, B. Harrias, Ed. New York: Wiley, 1967, pp. 189–232.

[13] S. A. Alshebeili, M. T. Ozgen, A. E. Cetin, and A. N. Venetsanopoulos, "Cumulant-based parametric multichannel FIR system identification methods," in *Proc. IEEE Signal Process. Workshop Higher Order Stat.*, June 1993, pp. 200–204.

[14] G. B. Giannakis and J. M. Mendel, "Cumulant-based order determination of non-Gaussian ARMA models," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 38, pp. 1411–1422, Aug. 1993.

[15] X.-D. Zhang and Y.-S. Zhang, "Singular value decomposition-based MA order determination of non-Gauusian ARMA models," *IEEE Trans. Signal Processing*, vol. 41, pp. 2657–2664, Aug. 1993.

[16] T. W. S. Chow and H.-Z. Tan, "Semiblind identification of nonminimum-phase ARMA models via order recursion with higher order cumulants," *IEEE Trans. Ind. Electron.*, vol. 45, pp. 663–671, Aug. 1998.

[17] C. L. Nikias and A. P. Petropulu, *Higher-Order Spectra Analysis: A Nonlinear Signal Processing Framework*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

Wei Li received B.Eng. and M.Eng. degrees from Northeastern University, Shenyang, China, in 1985 and 1988, respectively. She is currently pursuing the Ph.D. degree at the Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong.

She was a Lecturer in the Department of Automation Control, Northeastern University, from 1988 to 1995. Since 1995, she has been with the Hong Kong Polytechnic University. Her current research interests include statistical signal processing, source separation, and system identification.

Wan-Chi Siu (M'77–SM'90) received the Associateship from The Hong Kong Polytechnic University (formerly called the Hong Kong Polytechnic), the M.Phil. degree from The Chinese University of Hong Kong, and the Ph.D. degree from Imperial College of Science, Technology, and Medicine, London, U.K., in 1975, 1977 and 1984, respectively.

He was with The Chinese University of Hong Kong between 1975 and 1980. He then joined The Hong Kong Polytechnic University as a Lecturer in 1980 and was Chair Professor and Associate Dean of Engineering Faculty from 1992 to 1994. He has been Chair Professor and Head of Department of Electronic and Information Engineering of the same university since 1994. He has published over 180 research papers. His research interests include digital signal processing, fast computational algorithms, transforms, video coding, computational aspects of image processing and pattern recognition, and neural networks. He is a Member of the Editorial Board of the *Journal of VLSI Signal Processing Systems for Signal, Image and Video Technology* and an Overseas Member of the Editorial Board of the *IEE Review*.

Prof. Siu was a Guest Editor of the Special Issue on ISCAS'97 of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II, which was published in May 1998. He was also an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II from 1995 to 1997. He was the General Chairman of the International Symposium on Neural Networks, Image, and Speech Processing (ISSIPNN'94) and a Co-Chair of the Technical Program Committee of the IEEE International Symposium on Circuits and Systems (ISCAS'97), which were held in Hong Kong in April 1994 and June 1997, respectively. He is the General Chair of the 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP) to be held in Hong Kong. From 1991 to 1995, he was a Member of the Physical Sciences and Engineering Panel of the Research Grants Council (RGC), Hong Kong Government, and in 1994, he chaired the first Engineering and Information Technology Panel to assess the research quality of 19 Cost Centers (departments) from all universities in Hong Kong. He is an Honorary Professor of the South China University of Technology, Guangzhou, China. He is a Chartered Engineer, a Fellow of the IEE and the HKIE, and has also been listed in *Marquis Who's Who in the World, Marquis Who's Who in Science and Engineering* and other citation biographies.