Volterra Kernel Estimation for Nonlinear Communication Channels Using Deterministic Sequences

Ender M. Ekşioğlu and Ahmet H. Kayran

Department of Electrical and Electronics Engineering,
Istanbul Technical University,
Istanbul, 34469, Turkey.
E-mail: {ender, kayran}@ehb.itu.edu.tr

Abstract: We present a new exact method for the identification of communication channels with nonlinearities. The channel is modelled as a third-order discrete Volterra filter and the Volterra kernels are measured using deterministic input sequences and the corresponding channel outputs. The solutions are in closed form, exact and kernels are estimated in a non-recursive manner, thus eliminating error propagation. Complex inputs and complex kernels are allowed, which permits the use of PSK or QAM modulated signals for the identification of the bandpass Volterra channel. Simulation examples and comparison with other methods in the literature are provided to verify the method.

Key words: Nonlinear channel identification, Volterra systems, nonlinear systems

I. INTRODUCTION

Nonlinear channel identification is important in mitigating the effects of nonlinear distortions and for the equalization of the nonlinear communication channels. The performance of the compensation efforts for the channel nonlinearities are highly dependent on the accuracy of the nonlinear channel estimate. Hence, nonlinear system and channel identification has been a subject of significance [1], [2]. The Volterra series representation has been widely utilized to describe the input-output relationship of nonlinear systems. The truncated (or "doubly finite") Volterra filter representation can provide an approximation for a large class of nonlinear systems, and it is appropriate for modelling the nonlinearities encountered in communication systems. Hence, the characterization of nonlinear communication channels as finite order discrete Volterra filters has been studied in the literature.

The determination of the Volterra kernels of the nonlinear communication system model has been largely based on the use of input-output relations for random inputs. In the case of zero-mean Gaussian input, the closed form estimates for the Volterra kernels can be formed using the cross cumulants between the input and the output [3]. However, the Gaussian assumption is far fetched for most communication applications. Moreover, the estimates are found in a sequential manner, which causes error propagation between kernel estimates. In [4] PSK (phase shift keying) modulated input signals, which have vanishing higher order moments, have been studied as probing signals for nonlinear channels. A similar method has been given in [5] for more general input sequences. Another Volterra kernel estimation algorithm based on cyclostationarity of communication signals is presented in [6] where the kernels for a bandpass nonlinear system are identified under i.i.d. M-QAM (quadrature amplitude modulation) and M-PSK inputs. All of the methods presented depend on higher-order moments and cross correlations between input and output sequences.

In this work, we focus on deterministic excitation sequences for the identification of nonlinear channels modelled using the third-order truncated Volterra series representation. The algorithm developed in this paper is based on the novel Volterra system identification method presented in [7]. However, in [7] only real input data was considered. Here, we will allow for complex inputs and complex Volterra kernels. Our algorithm provides exact, closed form solutions for the Volterra kernels. The results do not depend on statistics of random sequences; rather the algorithm utilizes specially designed deterministic sequences. Therefore, the algorithm shows better performance than the above random-input based methods even for input sequences of shorter length. The kernels are estimated separately, thus eliminating error propagation amongst the estimates. Complex inputs are allowed, hence PSK or QAM modulated communication signals can be utilized as the deterministic input sequence. In the context of a narrowband communication channel, the nonlinear system gets modeled by a “bandpass” Volterra series [6], [8]-[10]. We will modify our algorithm for the identification of bandpass Volterra channels and present the corresponding simulation results.

II. NOVEL VOLTERRA FILTER REPRESENTATION

The third-order nonlinear channel model can be given as:

$$\begin{aligned} y(n) = \mathcal{N}[x(n)] = &\sum_{i_1=0}^{N} b_1\left(i_1\right) x(n-i_1) + \sum_{i_1=0}^{N}\sum_{i_2=i_1}^{N} b_2\left(i_1, i_2\right) x(n-i_1)x(n-i_2) + \\ &\sum_{i_1=0}^{N}\sum_{i_2=i_1}^{N}\sum_{i_3=i_2}^{N} b_3\left(i_1, i_2, i_3\right) x(n-i_1)x(n-i_2)x(n-i_3) \end{aligned} \tag{1}$$

where N is the memory length of the nonlinear system and $b_k\left(i_1, i_2, \ldots, i_k\right)$ is the Volterra kernel of degree k [1]. We introduce a new representation for the Volterra system by rearranging the Volterra kernels. We reformulate the third-order discrete Volterra system in terms of the multivariate cross-term complexity. The output $y(n)$ can be considered as the sum of the outputs of 3 different multivariate cross-term nonlinear subsystems, $\mathcal{H}^{(\ell)}$; that is

$$y(n) = \sum_{\ell=1}^{3} y^{(\ell)}(n) = \sum_{\ell=1}^{3} \mathcal{H}^{(\ell)}[x(n)] \tag{2a}$$

where

$$\mathcal{H}^{(1)}[x(n)] = \sum_{i=0}^{N} \mathbf{h}^{(1)^T}(i)\ \mathbf{x}^{(1)}(n-i) \tag{2b}$$

$$\mathcal{H}^{(2)}[x(n)] = \sum_{q_1=1}^{N}\sum_{i=0}^{N-q_1} \mathbf{h}^{(2)^T}(q_1; i)\ \mathbf{x}^{(2)}(q_1; n-i) \tag{2c}$$

$$\mathcal{H}^{(3)}[x(n)] = \sum_{q_1=1}^{N-1}\sum_{q_2=1}^{N-q_1}\sum_{i=0}^{N-q_1-q_2} \mathbf{h}^{(3)^T}(q_1, q_2; i)\ \mathbf{x}^{(3)}(q_1, q_2; n-i) \tag{2d}$$

$\mathcal{H}^{(\ell)}[\cdot]$ is called as an ℓ-D cross-term Volterra operator, $\mathbf{h}^{(\ell)}$ is called as an ℓ-D kernel vector and $\mathbf{x}^{(\ell)}$ is called as an ℓ-D input vector. Here ℓ-D denotes the cross-term complexity of the input vector corresponding to the kernel vector, rather than the literal dimension of the vectors. It is possible to give the ℓ-D kernel vectors and the corresponding input vectors in terms of the Volterra kernels and the input signal $x(n)$, respectively:

$$\mathbf{h}^{(1)}(i) = \left[b_1(i)\ b_2(i,i)\ b_3(i,i,i)\right]^T \tag{3a}$$

$$\mathbf{x}^{(1)}(n) = \left[x(n)\ x^2(n)\ x^3(n)\right]^T \tag{3b}$$

$$\mathbf{h}^{(2)}\left(q_1; i\right) = \begin{bmatrix} b_2\left(i, i+q_1\right) \\ b_3(i, i, i+q_1) \\ b_3(i, i+q_1, i+q_1) \end{bmatrix} \tag{4a}$$

$$\mathbf{x}^{(2)}\left(q_1; n\right) = \begin{bmatrix} x(n-i)x\big(n-(i+q_1)\big) \\ x\big(n-i\big)^2 x\big(n-(i+q_1)\big) \\ x\big(n-i\big)x\big(n-(i+q_1)\big)^2 \end{bmatrix} \tag{4b}$$

$$\mathbf{h}^{(3)}\left(q_1, q_2; i\right) = \left[\ b_3\big(i, i+q_2, i+q_1+q_2\big)\ \right] \tag{5a}$$

$$\mathbf{x}^{(3)}\left(q_1, q_2; n\right) = \left[\ x(n-i)x\big(n-(i+q_2)\big)x\big(n-(i+q_1+q_2)\big)\ \right] \tag{5b}$$

where the limits for q_1, q_2 and i are as given in (2). We can see that the ℓ-D kernels correspond to an input vector, which can be factored into powers of ℓ distinct input terms with ℓ-distinct delays. Hence by cross-term complexity we consider the number of distinct input powers utilized in forming the terms of the input vectors and group the kernels accordingly. In the regular Volterra representation (1) on the other hand, the Volterra kernels are grouped together using the degree of the nonlinearity of the corresponding input terms. The introduced novel representation enables us to devise an exact closed form algorithm for identifying the Volterra kernels of a third-order Volterra system utilizing deterministic input sequences.

III. IDENTIFICATION OF THE VOLTERRA KERNELS USING DETERMINISTIC INPUT SIGNALS

In this section, we derive an efficient algorithm to identify the kernel vectors $\mathbf{h}^{(\ell)}$, $\ell = 1, 2, 3$ defined in (2) by using deterministic input sequences with 3 distinct levels (other than zero). All kernel vectors can be determined by using only the output of the system.

A. Identification of the 1-D Kernel Vectors

We will prove that an ensemble of three input sequences composed of single impulses with distinct values, $x^{(1)}(j; n) = a_j \delta(n)$, for $j = 1, 2, 3$ is adequate to obtain the 1-D kernel vectors in (3a). Looking at the cross-term representation in (2), we can see that for these single impulses the outputs of the 2-D and 3-D subsytems are zero, i.e., $\mathcal{H}^{(2)}[a_j \delta(n)] = 0$

and $\mathcal{H}^{(3)}[a_j\delta(n)] = 0$. Hence, the output of the overall nonlinear system when the input is $x^{(1)}(j;n)$ is given by

$$\mathcal{N}\big[x^{(1)}(j;n)\big] = \mathcal{H}^{(1)}\big[x^{(1)}(j;n)\big] = v^{(1,1)}(j;n) = \sum_{i=0}^{N} \mathbf{h}^{(1)^T}(i)\mathbf{u}^{(1)}(j;n-i) \tag{6a}$$

where

$$\mathbf{u}^{(1)}(j;n) = \big[a_j \; a_j^2 \; a_j^3\big]^T \delta(n) \tag{6b}$$

We can write the three output sequences together in the matrix form as follows:

$$\mathbf{y}_e^{(1)}(n) = \mathcal{H}^{(1)}[\mathbf{x}_e^{(1)}] = \sum_{i=0}^{N} \mathbf{U}_e^{(1)}(n-i)\,\mathbf{h}^{(1)}(i) \tag{7}$$

where $\mathbf{y}_e^{(1)}(n)$, $\mathbf{x}_e^{(1)}(n)$ and $\mathbf{U}_e^{(1)}(n)$ denote the ensemble output vector, ensemble input vector and the input matrix, respectively:

$$\mathbf{y}_e^{(1)}(n) = \mathbf{v}_e^{(1,1)}(n) = \begin{bmatrix} v^{(1,1)}(1;n) \\ v^{(1,1)}(2;n) \\ v^{(1,1)}(3;n) \end{bmatrix} ; \quad \mathbf{x}_e^{(1)}(n) = \begin{bmatrix} a_1\delta(n) \\ a_2\delta(n) \\ a_3\delta(n) \end{bmatrix} ; \quad \mathbf{U}_e^{(1)}(n) = \begin{bmatrix} \mathbf{u}^{(1)^T}(1;n) \\ \mathbf{u}^{(1)^T}(2;n) \\ \mathbf{u}^{(1)^T}(3;n) \end{bmatrix} \tag{8}$$

Here $\mathbf{u}^{(1)}(j;n)$ is as defined in (6b). We can replace $\mathbf{U}_e^{(1)}(n-i)$ in (7) with $\mathbf{U}_e^{(1)}\delta(n-i)$, where the matrix $\mathbf{U}_e^{(1)}$ is given as

$$\mathbf{U}_e^{(1)} = \begin{bmatrix} a_1 & a_1^2 & a_1^3 \\ a_2 & a_2^2 & a_2^3 \\ a_3 & a_3^2 & a_3^3 \end{bmatrix} \tag{9}$$

We can rewrite (7) as

$$\mathbf{y}_e^{(1)}(n) = \sum_{i=0}^{N} \mathbf{U}_e^{(1)}\,\mathbf{h}^{(1)}(i)\,\delta(n-i) = \mathbf{U}_e^{(1)}\,\mathbf{h}^{(1)}(n) \tag{10}$$

It follows that if the inverse of the matrix $\mathbf{U}_e^{(1)}$ exists, we can determine the 1-D kernel vectors as

$$\mathbf{h}^{(1)}(n) = \big[\mathbf{U}_e^{(1)}\big]^{-1}\mathbf{y}_e^{(1)}(n) \qquad \text{for} \quad n = 0, 1, \ldots, N \tag{11}$$

The input matrix $\mathbf{U}_e^{(1)}$ in (9) is invertible if the levels of the single impulse input sequences are distinct and nonzero, i.e., $a_i \neq 0$ and $a_i \neq a_j$, $\forall\, i \neq j$.

B. Identification of the 2-D Kernel Vectors

We form the following ensemble of input sequences which consist of two impulses with distinct amplitudes. The impulses are separated by q_1.

$$\mathbf{x}_e^{(2)}(q_1;n) = \begin{bmatrix} x^{(2)}\big((1,2),q_1;n\big) \\ x^{(2)}\big((1,3),q_1;n\big) \\ x^{(2)}\big((2,3),q_1;n\big) \end{bmatrix} = \begin{bmatrix} a_1\delta(n) + a_2\delta(n-q_1) \\ a_1\delta(n) + a_3\delta(n-q_1) \\ a_2\delta(n) + a_3\delta(n-q_1) \end{bmatrix} \tag{12}$$

These sequences will only excite the 2-D subsystem $\mathcal{H}^{(2)}$ and the 1-D subsystem $\mathcal{H}^{(1)}$. The output ensemble of the overall nonlinear system can be written as a sum of the outputs for $\mathcal{H}^{(2)}$ and $\mathcal{H}^{(1)}$.

$$\mathbf{y}_e^{(2)}(q_1;n) = \mathcal{H}^{(1)}\big[\mathbf{x}_e^{(2)}\big(q_1;n\big)\big] + \mathcal{H}^{(2)}\big[\mathbf{x}_e^{(2)}\big(q_1;n\big)\big] = \mathbf{v}_e^{(2,1)}\big(q_1;n\big) + \mathbf{v}_e^{(2,2)}\big(q_1;n\big) \tag{13}$$

Using (2b) and (12), it is not difficult to show that the response of the 1-D system to the 2-D input ensemble can be decomposed in terms of the 1-D responses as

$$\begin{aligned} \mathbf{v}_e^{(2,1)}(q_1;n) &= \mathcal{H}^{(1)} \begin{bmatrix} a_1\delta(n) \\ a_1\delta(n) \\ a_2\delta(n) \end{bmatrix} + \mathcal{H}^{(1)} \begin{bmatrix} a_2\delta(n-q_1) \\ a_3\delta(n-q_1) \\ a_3\delta(n-q_1) \end{bmatrix} \\ &= \begin{bmatrix} v^{(1,1)}(1;n) \\ v^{(1,1)}(1;n) \\ v^{(1,1)}(2;n) \end{bmatrix} + \begin{bmatrix} v^{(1,1)}(2;n-q_1) \\ v^{(1,1)}(3;n-q_1) \\ v^{(1,1)}(3;n-q_1) \end{bmatrix} \end{aligned} \tag{14}$$

Note that all the terms in (14) are output sequences which were found in section III.A in (6). We can obtain the response of the 2-D subsystem alone by subtracting the response of the 1-D subsystem from the nonlinear system output vector.

$$\mathbf{v}_e^{(2,2)}(q_1;n) = \begin{bmatrix} v^{(2,2)}\big((1,2),q_1;n\big) \\ v^{(2,2)}\big((1,3),q_1;n\big) \\ v^{(2,2)}\big((2,3),q_1;n\big) \end{bmatrix} = \mathbf{y}_e^{(2)}(q_1;n) - \mathbf{v}_e^{(2,1)}(q_1;n) \tag{15}$$

On the other hand using (2c) we can write the 2-D subsystem output for our ensemble input as,

$$\mathbf{v}_e^{(2,2)}(q_1;n) = \mathbf{U}_e^{(2)}\mathbf{h}^{(2)}(q_1;n-q_1) \tag{16}$$

$\mathbf{U}_e^{(2)}$ is a matrix which is given as

$$\mathbf{U}_e^{(2)} = \begin{bmatrix} a_1a_2 & a_1a_2^2 & a_1^2a_2 \\ a_1a_3 & a_1a_3^2 & a_1^2a_3 \\ a_2a_3 & a_2a_3^2 & a_2^2a_3 \end{bmatrix} \tag{17}$$

We can identify the 2-D Volterra kernel vectors by using (15) and (16),

$$\mathbf{h}^{(2)}(q_1; n-q_1) = \left[\mathbf{U}_e^{(2)}\right]^{-1} \mathbf{v}_e^{(2,2)}(q_1; n) \tag{18}$$

for $q_1 = 1, \ldots, N$ and $n = q_1, q_1+1, \ldots, N$.

C. Identification of the 3-D Kernel Vectors

Now we apply an input sequence which includes three distinct impulses to the nonlinear system.

$$x^{(3)}(q_1, q_2; n) = a_1\delta(n) + a_2\delta(n-q_2) + a_3\delta(n-q_1-q_2) \tag{19}$$

The output of the overall nonlinear system can be written as the sum of the outputs of the individual subsystems, $\mathcal{H}^{(1)}$, $\mathcal{H}^{(2)}$, and $\mathcal{H}^{(3)}$,

$$y^{(3)}(q_1, q_2; n) = \sum_{i=1}^{3} v^{(3,i)}(q_1, q_2; n) \tag{20}$$

The output of the 1-D subsystem for the three impulse input $x^{(3)}(q_1, q_2; n)$ can be written as

$$v^{(3,1)}(q_1, q_2; n) = v^{(1,1)}(1; n) + v^{(1,1)}(2; n-q_2) + v^{(1,1)}(3; n-q_1-q_2) \tag{21}$$

The output of the 2-D subsystem for the three impulse input $x^{(3)}(q_1, q_2; n)$ can be written as

$$v^{(3,2)}(q_1, q_2; n) = v^{(2,2)}\big((1,2), q_2; n\big) + v^{(2,2)}\big((1,3), q_1+q_2; n\big) + v^{(2,2)}\big((2,3), q_1; n-q_2\big) \tag{22}$$

Note that all the terms for the above subsytem output sequences are previously observed and calculated in the identification of the 1-D and 2-D kernel vectors. The output for the 3-D subsystem can be left alone by subtracting the responses of the 1-D and 2-D subsytems from the overall system output.

$$v^{(3,3)}(q_1, q_2; n) = y^{(3)}(q_1, q_2; n) - v^{(3,1)}(q_1, q_2; n) - v^{(3,3)}(q_1, q_2; n) \tag{23}$$

On the other hand, in a similar fashion to the equations for the 1-D and 2-D subsystems $\big((10), (16)\big)$, the output of the 3-D subsystem $\mathbf{v}_e^{(3,3)}(q_1, q_2; n)$ can be written as,

$$v^{(3,3)}(q_1, q_2; n) = \mathbf{U}_e^{(3)} \mathbf{h}^{(3)}(q_1, q_2; n-q_1-q_2) \tag{24}$$

where $\mathbf{U}_e^{(3)} = a_1 a_2 a_3$.

Hence, we get

$$\mathbf{h}^{(3)}(q_1, q_2; n-q_1-q_2) = \left[\mathbf{U}_e^{(3)}\right]^{-1} v^{(3,3)}(q_1, q_2; n) \tag{25}$$

for $q_1 = 1, \ldots, N-1$, $q_2 = 1, \ldots, N-q_1$ and $n = q_1+q_2, q_1+q_2+1, \ldots, N$.

Fig. 1 depicts the identification of the Volterra kernels using the proposed algorithm. The matrices **T** and **S** shown in this figure are utilized to form the input ensembles and to choose the past output ensembles which get subtracted as in (15) and (23). The **T** matrices are given as,

$$\mathbf{T}_{2,1} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}; \quad \mathbf{T}_{2,2} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix};$$

$$\mathbf{T}_{3,1} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}; \quad \mathbf{T}_{3,2} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}; \quad \mathbf{T}_{3,3} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$$

The **S** matrices are given as

$$\mathbf{S}_{2,1} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}; \quad \mathbf{S}_{3,1} = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}; \quad \mathbf{S}_{3,2} = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$$

General closed form recursive algorithms to calculate the matrices **T** and **S** from scratch can be found in [7]. Using the approach in [7], the identification algorithm developed here can be extended to the identification of systems with nonlinearities higher than third order.

Example: We consider the third order nonlinear system with $N = 2$ given by

$$y(n) = x(n) + 2x(n-1) - x^2(n) - 4x(n)x(n-1) - 2x^2(n-1) + 3x^3(n) +$$
$$4x^2(n)x(n-1) + 5x(n)x^2(n-1) - 3x^3(n-1) - 5x(n)x(n-2) + 6x(n)x(n-1)x(n-2) \tag{26}$$

We want to find the Volterra kernels using the method outlined in this section. First we are interested in calculating the 1-D kernel vectors $\mathbf{h}^{(1)}(0) = [1 \ \ -1 \ \ 3]^T$ and $\mathbf{h}^{(1)}(1) = [2 \ \ -2 \ \ -3]^T$ from the input and the output. We choose the impulse levels a_1, a_2, a_3 from the QPSK signal set $e^{j(2\pi k/4)}, k = 0, 1, 2, 3$. Hence, we define the 1-D input ensemble to the system as

$$\mathbf{x}_e^{(1)}(n) = \begin{bmatrix} 1 \\ -1 \\ j \end{bmatrix} \delta(n) \tag{27}$$

The 1-D output ensemble from (8) becomes

$$\mathbf{y}_e^{(1)}(n) = \begin{bmatrix} 3 \\ -5 \\ 1-2j \end{bmatrix} \delta(n) + \begin{bmatrix} -3 \\ -1 \\ 2+5j \end{bmatrix} \delta(n-1) \tag{28}$$

From (9) and (27), the matrix $\mathbf{U}_e^{(1)}$ is written as,

$$\mathbf{U}_e^{(1)} = \begin{bmatrix} 1 & 1 & 1 \\ -1 & 1 & -1 \\ j & -1 & -j \end{bmatrix} \tag{29}$$

Applying (28) and (29) to (11) yields,

$$\mathbf{h}^{(1)}(0) = \left[\mathbf{U}_e^{(1)}\right]^{-1} \begin{bmatrix} 3 \\ -5 \\ 1-2j \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \\ 3 \end{bmatrix}; \text{ and } \mathbf{h}^{(1)}(1) = \left[\mathbf{U}_e^{(1)}\right]^{-1} \begin{bmatrix} -3 \\ -1 \\ 2+5j \end{bmatrix} = \begin{bmatrix} 2 \\ -2 \\ -3 \end{bmatrix} \tag{30}$$

Next, we want to determine the 2-D kernel vectors $\mathbf{h}^{(2)}(1;0) = [-4\ 4\ 5]^T$ and $\mathbf{h}^{(2)}(2;0) = [-5\ 0\ 0]^T$ of the nonlinear system given in (26). Using the 1-D input ensemble vector $\mathbf{x}_e^{(1)}(n)$ in (27), the 2-D input ensembles in (12) can be written as

$$\mathbf{x}_e^{(2)}(1;n) = \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix} \delta(n) + \begin{bmatrix} -1 \\ j \\ j \end{bmatrix} \delta(n-1) \tag{31}$$

$$\mathbf{x}_e^{(2)}(2;n) = \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix} \delta(n) + \begin{bmatrix} -1 \\ j \\ j \end{bmatrix} \delta(n-2) \tag{32}$$

For the 2-D input ensemble in (31), the output of the nonlinear system in (26) is calculated as

$$\mathbf{y}_e^{(2)}(1;n) = \begin{bmatrix} 3 \\ 3 \\ -5 \end{bmatrix} \delta(n) + \begin{bmatrix} -5 \\ -6-j \\ 4+7j \end{bmatrix} \delta(n-1) + \begin{bmatrix} -1 \\ 2+5j \\ 2+5j \end{bmatrix} \delta(n-2) \tag{33}$$

It is possible to determine the response of the 1-D subsystem to the 2-D input ensemble in (31). To accomplish this we use the 1-D output ensemble given in (28) and (14).

$$\mathbf{v}_e^{(2,1)}(1;n) = \begin{bmatrix} 3 \\ 3 \\ -5 \end{bmatrix} \delta(n) + \begin{bmatrix} -8 \\ -2-2j \\ -2j \end{bmatrix} \delta(n-1) + \begin{bmatrix} -1 \\ 2+5j \\ 2+5j \end{bmatrix} \delta(n-2) \tag{34}$$

Using this result, the response of the 2-D subsystem can be obtained by subtracting $\mathbf{v}_e^{(2,1)}(1;n)$ from the nonlinear system output $\mathbf{y}_e^{(2)}(1;n)$ in (33).

$$\begin{aligned}\mathbf{v}_e^{(2,2)}(1;n) &= \mathbf{y}_e^{(2)}(1;n) - \mathbf{v}_e^{(2,1)}(1;n) \\ &= \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \delta(n) + \begin{bmatrix} 3 \\ -4+j \\ 4+9j \end{bmatrix} \delta(n-1) + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \delta(n-2)\end{aligned} \tag{35}$$

The desired 2-D Volterra kernel $\mathbf{h}^{(2)}(1;0)$ can be calculated by substituting (35) into (18)

$$\begin{aligned}\mathbf{h}^{(2)}(1;0) &= \left[\mathbf{U}_e^{(2)}\right]^{-1} \mathbf{v}_e^{(2,2)}(1;1) \\ &= \begin{bmatrix} -1 & 1 & -1 \\ j & -1 & j \\ -j & 1 & j \end{bmatrix}^{-1} \begin{bmatrix} 3 \\ -4+j \\ 4+9j \end{bmatrix} = \begin{bmatrix} -4 \\ 4 \\ 5 \end{bmatrix}\end{aligned} \tag{36}$$

where the constant matrix $\mathbf{U}_e^{(2)}$ is obtained from $\mathbf{x}_e^{(1)}(n)$ and (17).

For the 2-D input ensemble in (32), the output of the nonlinear system is calculated as

$$\mathbf{y}_e^{(2)}(2;n) = \begin{bmatrix} 3 \\ 3 \\ -5 \end{bmatrix} \delta(n) + \begin{bmatrix} -3 \\ -3 \\ -1 \end{bmatrix} \delta(n-1) + \begin{bmatrix} 0 \\ 1-7j \\ 1+3j \end{bmatrix} \delta(n-2) + \begin{bmatrix} -1 \\ 2+5j \\ 2+5j \end{bmatrix} \delta(n-3) \tag{37}$$

It is possible to determine the response of the 1-D subsystem to the 2-D input ensemble in (32). To accomplish this we use the 1-D output ensemble given in (28) and (14).

$$\mathbf{v}_e^{(2,1)}(2;n) = \begin{bmatrix} 3 \\ 3 \\ -5 \end{bmatrix} \delta(n) + \begin{bmatrix} -3 \\ -3 \\ -1 \end{bmatrix} \delta(n-1) + \begin{bmatrix} -5 \\ 1-2j \\ 1-2j \end{bmatrix} \delta(n-2) + \begin{bmatrix} -1 \\ 2+5j \\ 2+5j \end{bmatrix} \delta(n-3) \tag{38}$$

Using this result, the response of the 2-D subsystem can be obtained by subtracting $\mathbf{v}_e^{(2,1)}(2;n)$ from the nonlinear system output $\mathbf{y}_e^{(2)}(2;n)$ in (37).

$$\begin{aligned}\mathbf{v}_e^{(2,2)}(2;n) &= \mathbf{y}_e^{(2)}(2;n) - \mathbf{v}_e^{(2,1)}(2;n) \\ &= \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \delta(n) + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \delta(n-1) + \begin{bmatrix} 5 \\ -5j \\ 5j \end{bmatrix} \delta(n-2) + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \delta(n-3)\end{aligned} \tag{39}$$

The desired 2-D Volterra kernel $\mathbf{h}^{(2)}(2;0)$ can be calculated by substituting (39) into (18)

$$\mathbf{h}^{(2)}(2;0) = \left[\mathbf{U}_e^{(2)}\right]^{-1} \mathbf{v}_e^{(2,2)}(2;2)$$

$$= \begin{bmatrix} -1 & 1 & -1 \\ j & -1 & j \\ -j & 1 & j \end{bmatrix}^{-1} \begin{bmatrix} 5 \\ -5j \\ 5j \end{bmatrix} = \begin{bmatrix} -5 \\ 0 \\ 0 \end{bmatrix} \tag{40}$$

Finally, we want to determine the 3-D kernel $\mathbf{h}^{(3)}(1,1;0) = 6$ of the nonlinear system. The 3-D input can be written as

$$x^{(3)}(1,1;n) = 1 \cdot \delta(n) + (-1) \cdot \delta(n-1) + j \cdot \delta(n-2) \tag{41}$$

For the 3-D input ensemble in (41), the output of the nonlinear system in (26) is calculated as

$$y^{(3)}(1,1;n) = 3\delta(n) - 5\delta(n-1) + (4-4j)\delta(n-2) + (2+5j)\delta(n-3) \tag{42}$$

It is possible to determine the response of the 1-D subsystem to the 3-D input ensemble in (41). The output of the 1-D subsystem is found by utilizing (21) and the 1-D output ensemble given in (28).

$$v^{(3,1)}(1,1;n) = 3\delta(n) - 8\delta(n-1) - 2j\delta(n-2) + (2+5j)\delta(n-3) \tag{43}$$

It is also possible to determine the response of the 2-D subsystem to the 3-D input in (41). We find the response of the 2-D subsystem by using (22) and the 1-D output ensemble given in (28).

$$v^{(3,2)}(1,1;n) = 3\delta(n) - 8\delta(n-1) + -2j\delta(n-2) + (2+5j)\delta(n-3) \tag{44}$$

Using these results, the response of the 3-D subsystem alone can be obtained by subtracting $v^{(3,1)}(1,1;n)$ and $v^{(3,2)}(1,1;n)$ from the nonlinear system output $y^{(3)}(1;n)$ in (42).

$$\begin{aligned} v^{(3,3)}(1,1;n) &= y^{(3)}(1,1;n) - v^{(3,1)}(1;n) - v^{(3,2)}(1;n) \\ &= -6j\delta(n-2) \end{aligned} \tag{45}$$

The desired 3-D Volterra kernel $\mathbf{h}^{(3)}(1,1;0)$ can be calculated by substituting (45) into (25):

$$\mathbf{h}^{(3)}(1,1;0) = \left[\mathbf{U}_e^{(3)}\right]^{-1} v^{(3,3)}(1,1;2) = (-j)^{-1}(-6j) = 6$$

Here, the constant matrix $\mathbf{U}_e^{(3)}$ is calculated as $\mathbf{U}_e^{(3)} = a_1 a_2 a_3$.

D. Length of the Required Probing Signal

We will give an upper-bound for the length of the overall input sequence, which should be applied to identify the Volterra kernels of the nonlinear channel. We consider the third-order Volterra filter with memory length N as the channel model. We assume the 1-D, 2-D and 3-D input ensembles constituting the input signal are applied serially to a single nonlinear system box. We put a guarding interval of length N with all zero levels at the end of each of the individual input ensembles, $x^{(1)}(j;n)$, $x^{(2)}\big((i,j),q_1;n\big)$, and $x^{(3)}(q_1,q_2;n)$. This guarding interval ensures to flush out that input ensemble from the memory of the nonlinear system and prepares the nonlinear system for the next ensemble by clearing the memory. Under these assumptions, we calculate the total length of the input sequence by summing the length of all the necessary 1-D, 2-D and 3-D input signals and adding N to each of them. Hence, the total input length is given by

$$L = 3(N+1) + 3\sum_{q_{1,2}=1}^{N}(q_{1,2}+N+1) + \sum_{q_{1,3}=1}^{N-1}\sum_{q_{2,3}=1}^{N-q_{1,3}} q_{1,3}+q_{2,3}+N+1 \tag{46}$$

Here, $q_{1,2}$ takes values between 1 and N as suggested by (2c). Similarly, the sum $q_{1,3}+q_{2,3}$ takes values between 2 and N, as suggested by (2d). Hence, both of these terms are upper-bounded by N, and we can write from (46)

$$\begin{aligned} &L \leqslant 3(N+1) + 3N(2N+1) + \frac{(N^2-N)}{2}(2N+1) \\ \Rightarrow \quad &L \leqslant (2N+1)\binom{N+3}{2} - 3N \end{aligned} \tag{47}$$

Therefore, the required length of our total input sequence is upper-bounded by $(2N+1)\binom{N+3}{2} - 3N$.

IV. IDENTIFICATION OF BANDPASS VOLTERRA CHANNELS

The bandpass Voltarra series is employed in the baseband representation of narrow-band communication systems. For bandpass communication signals, where the carrier frequency is much larger than the modulated channel bandwidth, the complex envelope of the nonlinear channel output signal gets described by a bandpass Volterra series rather than the regular Volterra filter representation as in (1). The even-order terms in the regular representation diappear, since they generate spectral components which fall outside the channel bandwidth and hence can be filtered by bandpass filter [10]. The bandpass Volterra filter including nonlinearities up to third order is given as:

$$y(n) = \sum_{i_1=0}^{N} b_1(i_1)x(n-i_1) + \sum_{i_1=0}^{N}\sum_{i_2=0}^{N}\sum_{i_3=0}^{N} b_3(i_1,i_2,i_3)x^*(n-i_1)x(n-i_2)x(n-i_3) \tag{48}$$

Here, $()^*$ denotes complex conjugation. N is the memory length of the bandpass nonlinear system. $b_1\left(i_1\right)$ and $b_3\left(i_1,i_2,i_3\right)$ are the complex-valued linear and cubic bandpass Volterra kernels, respectively [6].

We can easily modify the identification method we developed for the regular Volterra filter to the bandpass Volterra channel case. We will first reformulate the input-output relationship for the bandpass Volterra filter. The new representation will be similar to (2). The output $y(n)$ for the bandpass Volterra filter in (49) can be considered as the sum of the outputs of three different nonlinear subsystems, $\mathcal{H}^{(\ell)}$; that is

$$y(n) = \sum_{\ell=1}^{3} y^{(\ell)}(n) = \sum_{\ell=1}^{3} \mathcal{H}^{(\ell)}[x(n)] \tag{49a}$$

where

$$\mathcal{H}^{(1)}[x(n)] = \sum_{i=0}^{N} \mathbf{h}^{(1)^T}(i)\ \mathbf{x}^{(1)}(n-i) \tag{49b}$$

$$\mathcal{H}^{(2)}[x(n)] = \sum_{q_1=1}^{N}\sum_{i=0}^{N-q_1} \mathbf{h}^{(2)^T}(q_1;i)\ \mathbf{x}^{(2)}(q_1;n-i) \tag{49c}$$

$$\mathcal{H}^{(3)}[x(n)] = \sum_{q_1=1}^{N-1}\sum_{q_2=1}^{N-q_1}\sum_{i=0}^{N-q_1-q_2} \mathbf{h}^{(3)^T}(q_1,q_2;i)\ \mathbf{x}^{(3)}(q_1,q_2;n-i) \tag{49d}$$

The kernel vectors and the input vectors utilized in this representation can be given in terms of the bandpass Volterra system kernels (49) and the input signal $x(n)$, respectively.

$$\mathbf{h}^{(1)}(i) = \left[b_1(i) \quad b_3(i,i,i)\right]^T \tag{50}$$

$$\mathbf{x}^{(1)}(n) = \left[x(n) \quad |x(n)|^{\,2} x(n)\right]^T \tag{51}$$

$$\mathbf{h}^{(2)}\left(q_1;i\right) = \begin{bmatrix} b_3(i,i,i+q_1) \\ b_3(i+q_1,i+q_1,i) \\ b_3(i,i+q_1,i+q_1) \\ b_3(i+q_1,i,i) \end{bmatrix} \tag{52}$$

$$\mathbf{x}^{(2)}\left(q_1;n\right) = \begin{bmatrix} |x(n-i)|^2 x\big(n-(i+q_1)\big) \\ \big|x\big(n-(i+q_1)\big)\big|^2 x(n-i) \\ x(n-i)^* x^2\big(n-(i+q_1)\big) \\ x\big(n-(i+q_1)\big)^* x^2(n-i) \end{bmatrix} \tag{53}$$

$$\mathbf{h}^{(3)}\left(q_1,q_2;i\right)=\left[\begin{array}{c} b_3(i,i+q_2,i+q_1+q_2) \end{array}\right] \tag{54}$$

$$\mathbf{x}^{(3)}\left(q_1,q_2;n\right)=\left[\begin{array}{c} x(n-i)^{*}x\big(n-(i+q_2)\big)x\big(n-(i+q_1+q_2)\big) \end{array}\right] \tag{55}$$

Here, the limits for q_1, q_2 and i are as given in (49).

This representation enables us to form an algorithm for identifying the bandpass Volterra kernels using deterministic sequences. The algorithm as we detailed in Section III can be used again for identification, however this time for the bandpass Volterra system kernels. The sole difference will be in the matrices $\mathbf{U}_e^{(1)}$, $\mathbf{U}_e^{(2)}$ and $\mathbf{U}_e^{(3)}$. For the identification of the bandpass Volterra channel, these matrices will be given as

$$\mathbf{U}_e^{(1)}=\left[\begin{array}{cc} a_1 & |a_1|^2a_1 \\ a_2 & |a_2|^2a_2 \end{array}\right] \tag{56}$$

$$\mathbf{U}_e^{(2)}=\left[\begin{array}{cccc} |a_1|^2a_2 & |a_2|^2a_1 & a_1^*a_2^2 & a_2^*a_1^2 \\ |a_1|^2a_3 & |a_3|^2a_1 & a_1^*a_3^2 & a_3^*a_1^2 \\ |a_1|^2a_4 & |a_4|^2a_1 & a_1^*a_4^2 & a_4^*a_1^2 \\ |a_2|^2a_3 & |a_3|^2a_2 & a_2^*a_3^2 & a_3^*a_2^2 \end{array}\right] \tag{57}$$

$$\mathbf{U}_e^{(3)}=\left[\begin{array}{ccc} a_1^*a_2a_3 & a_2^*a_1a_3 & a_3^*a_1a_2 \\ a_1^*a_2a_4 & a_2^*a_1a_4 & a_4^*a_1a_2 \\ a_2^*a_3a_4 & a_3^*a_2a_4 & a_4^*a_2a_3 \end{array}\right] \tag{58}$$

Other than these changes in the utilized matrices, the algorithm as detailed in Section III works also for the identification of the bandpass Volterra channel.

V. SIMULATIONS

We present two numerical examples to illustrate the performance of our novel identification procedure.

Example 1: We simulate a linear-quadratic-cubic Volterra filter with memory length $N = 2$. We use QPSK modulated signals as the input, where the deterministic input levels are chosen from the set $4e^{j(2\pi k/4+\pi/4)}, k = 0, 1, 2, 3$ Additive independent GWN observation noise with unit variance is present. Our required deterministic sequence is of length 41. In order to compare the performance of our algorithm, we also simulate the method given in [4] for this setup. The PSK data length used for the method given in [4] is 4096. Table 1 shows the true values for the non-redundant kernels and the mean and the standard deviations of the estimates from our algorithm and the PSK input method of [4]. Note that the non-redundant kernels given in Table 1 are the triangular kernels. In the simulations in [4], symmetric kernel values are used [1, pp.34]. However, we utilized triangular kernel representation in our simulations to concord with our notation in (1). There are 11 nonzero Volterra kernels. The results for both methods are calculated over 400 independent trials. The results for our algorithm are better than those for the method of [4] even though our method uses an input sequence of length almost 100 times smaller (41 vs. 4096). Our estimates are very accurate despite the presence of noise and the short length of input utilized.

The method in [4] uses higher-order moments. In Example 1 of [4], a third order filter with $N = 4$ is simulated and the mean and deviations of the estimates are examined in the absence of noise for an input length of 4096. Since our algorithm is an exact algorithm, for this filter our input sequence gives the *exact* kernel values for an input length as short as 155 in the absence of noise.

Example 2: We simulate a linear-cubic "bandpass" Volterra filter, where the input-output relationship for the bandpass Volterra filter is given in (48). The channel model we simulate has a memory length of $N = 2$. We use QPSK modulated signals as the input, where we choose the input levels for our deterministic sequence from the set $2e^{j(2\pi k/4+\pi/4)}, k = 0, 1, 2, 3$. Additive independent GWN observation noise with variance 0.5 is present. The length of the required deterministic input sequence for our method is 32. We resend this input sequence 125 times through the nonlinear channel and calculate kernel estimates for each turn. Then, we take the mean over these kernel estimates and form our final estimate. Hence, for this example, the total length of the utilized input sequence is 32x125=4000.

We also realized the method for bandpass Volterra kernel identification as given in [6] for the simulation setup given above. The PSK data length used for the method of [6] is 4096. Table 2 shows the true values for the non-redundant kernels and the mean and the standard deviations of the estimates from our algorithm and the method detailed in [6]. The non-redundant kernels are the kernels given as $b_3(i, j, k)$, $b_3(i, k, k)$ and $b_1(i)$ [6]. There are a total of 10 Volterra kernels. The results for both methods are calculated over 400 independent trials. The results for our algorithm are better than those for the method of [6] even though our method employed an input sequence of shorter length.

VI. CONCLUDING REMARKS

We presented a novel method for input-output identification of the Volterra kernels of a nonlinear channel modelled as a third-order Volterra filter. Our method utilizes carefully designed deterministic sequences as the probing signal and avoids the shortcomings of the use of random signals and correlation methods. The algorithm works also for complex inputs, allowing the use of complex baseband communication signals. We give simulation examples demonstrating the performance of the algorithm compared to methods available in the literature. These methods can be easily extended to the identification of nonlinear channels with higher-order nonlinearities following the results given in [7].

REFERENCES

[1] V. J. Mathews and G. L Sicuranza, *Polynomial Signal Processing*, John Wiley&Sons, 2000.

[2] M. T. Özden, A. H. Kayran and E. Panayırcı, "Adaptive Volterra channel equalization with lattice ortogonalisation", *IEE Proceedings - Communications*, vol. 145, no. 2, pp. 109–115, Apr. 1998.

[3] P. Koukoulas and N. Kalouptsidis, "Nonlinear system identification using Gaussian inputs", *IEEE Trans. Signal Processing*, vol. 43, no. 8, pp. 1831-1841, Aug. 1995.

[4] G. T. Zhou and G. B. Giannakis, "Nonlinear channel identification and performance analysis with PSK inputs", *Proc. 1st IEEE Sig. Proc. Workshop on Wireless Comm.*, Paris, France, Apr. 1997, pp. 337–340.

[5] N. Petrochilos and P. Comon, "Nonlinear channel identification and performance analysis, ", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Istanbul, Turkey, June 2000, vol. 1, pp. 209–212.

[6] C. Cheng and E.J. Powers, "Optimal Volterra kernel estimation algorithms for a nonlinear communication system for PSK and QAM inputs", *IEEE Trans. Signal Processing*, vol. 49, no. 12, pp.147–163, Jan. 2001.

[7] E. M. Ekşioğlu and A. H. Kayran, "Nonlinear System Identification Using Deterministic Multilevel Sequences", *Proc. 14th International Conference on Digital Signal Processing*, Santorini, Greece, July. 2002, vol. 2, pp. 947–950

[8] C.-H. Tseng and E.J. Powers, "Identification of cubic systems using higher order moments of I.I.D. signals", *IEEE Trans. Signal Processing*, vol. 43, no. 7, pp.1733–1735, July 1995.

[9] E. Biglieri, S. Barberis, and M. Catena, "Analysis and compensation of nonlinearities in digital transmission systems", *IEEE Journal on Selected Areas in Communications*, vol. 6, no. 1, pp. 42-51, January 1988.

[10] S. Benedetto and E. Biglieri, "Nonlinear equalization of digital satellite channels", *IEEE Journal on Selected Areas in Communications*, vol. sac-1. no. 1, pp. 42-51, January 1983.

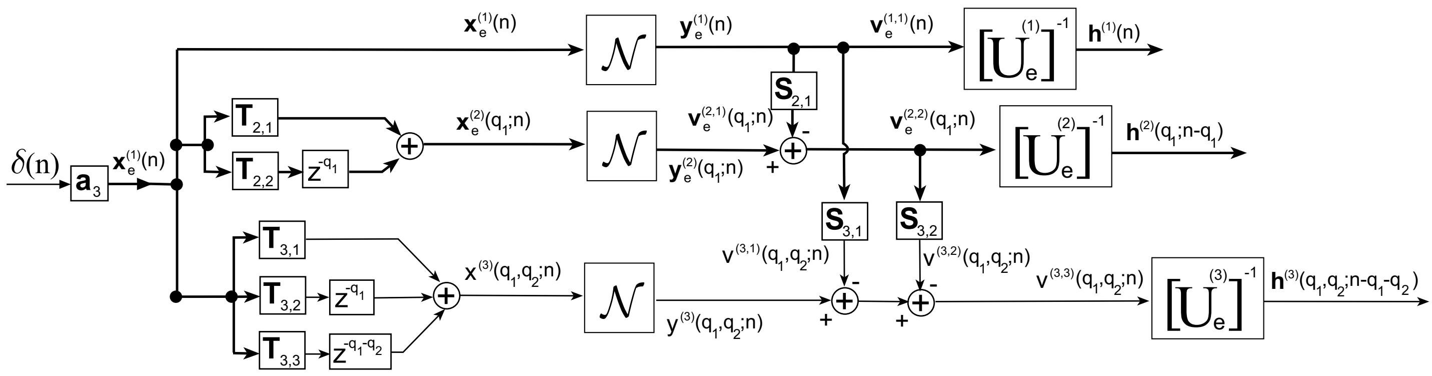

Fig. 1. Proposed third-order nonlinear channel identification method using deterministic sequences as inputs.

Table 1. Results for Example 1

(i_1)	(0)	(1)	(2)
true $b_1\left(i_1\right)$	1.0000	0.5000	-0.8000
mean of $\hat{b}_1\left(i_1\right)$ for [4]	0.9955	0.4886	-0.8150
mean of $\hat{b}_1\left(i_1\right)$ for our method	1.0045	0.5108	-0.8164
std of $\hat{b}_1\left(i_1\right)$ for [4]	0.5195	0.3758	0.3326
std of $\hat{b}_1\left(i_1\right)$ for our method	0.1219	0.1266	0.1237

(i_1, i_2)	$(0,0)$	$(0,1)$	$(1,1)$
true $b_2\left(i_1, i_2\right)$	1.0000	0.6000	-0.3000
mean of $\hat{b}_2\left(i_1, i_2\right)$ for [4]	1.0035	0.6026	-0.2958
mean of $\hat{b}_2\left(i_1, i_2\right)$ for our method	1.0009	0.5984	-0.3035
std of $\hat{b}_2\left(i_1, i_2\right)$ for [4]	0.1148	0.1335	0.0788
std of $\hat{b}_2\left(i_1, i_2\right)$ for our method	0.0014	0.0764	0.0050

(i_1, i_2, i_3)	$(0,0,0)$	$(0,0,1)$	$(0,1,1)$	$(1,1,1)$	$(0,1,2)$
true $b_3\left(i_1, i_2, i_3\right)$	1.0000	1.2000	0.8000	-0.5000	0.6000
mean of $\hat{b}_3\left(i_1, i_2, i_3\right)$ for [4]	0.99995	1.2005	0.7993	-0.5009	0.5997
mean of $\hat{b}_3\left(i_1, i_2, i_3\right)$ for our method	1.0002	1.2006	0.8019	-0.4997	0.6036
std of $\hat{b}_3\left(i_1, i_2, i_3\right)$ for [4]	0.0164	0.0235	0.0215	0.0281	0.0204
std of $\hat{b}_3\left(i_1, i_2, i_3\right)$ for our method	0.0077	0.0196	0.0185	0.0082	0.0285

Table 2. Results for Example 2

(i_1)	(0)	(1)	(2)
true $b_1\,(i_1)$	1.0000	0.5000+0.5000j	-0.6000
mean of $\hat{b}_1\,(i_1)$ for [6]	0.9995+0.0065j	0.5091+0.4971j	0.5967+0.0076j
mean of $\hat{b}_1\,(i_1)$ for our method	0.9999 + 0.0005j	0.5003 + 0.4994j	-0.5999+0.0006j
std of $\hat{b}_1\,(i_1)$ for [6]	0.1131	0.1086	0.1050
std of $\hat{b}_1\,(i_1)$ for our method	0.0183	0.0193	0.0184

(i_1, i_2, i_3)	$(0,1,1)$	$(1,0,0)$	$(1,2,2)$	$(2,1,1)$
true $b_3\,(i_1, i_2, i_3)$	1.00	0.40+0.40j	-0.40	0.60
mean of $\hat{b}_3\,(i_1, i_2, i_3)$ for [6]	0.9995+0.0024j	0.4017+0.3998j	-0.3993+0.0016j	0.5987-0.0004j
mean of $\hat{b}_3\,(i_1, i_2, i_3)$ for our method	1.0000-0.0007j	0.3995+0.3999j	-0.4000+0.0004j	0.6002+0.0001j
std of $\hat{b}_3\,(i_1, i_2, i_3)$ for [6]	0.0270	0.0320	0.0297	0.0259
std of $\hat{b}_3\,(i_1, i_2, i_3)$ for our method	0.0166	0.0144	0.0160	0.0139

(i_1, i_2, i_3)	$(2,0,0)$	$(0,2,2)$	$(0,1,2)$
true $b_3\,(i_1, i_2, i_3)$	0.60+0.70j	0.50	0.30+0.40j
mean of $\hat{b}_3\,(i_1, i_2, i_3)$ for [6]	0.5995 + 0.6995j	0.4996 - 0.0010j	0.3002 + 0.4003j
mean of $\hat{b}_3\,(i_1, i_2, i_3)$ for our method	0.6005 + 0.7002j	0.5000 + 0.0007j	0.2993 + 0.3993j
std of $\hat{b}_3\,(i_1, i_2, i_3)$ for [6]	0.0243	0.0263	0.0260
std of $\hat{b}_3\,(i_1, i_2, i_3)$ for our method	0.0138	0.0160	0.0217