Nonlinear System Identification Using Deterministic Multilevel Sequences

by

Ahmet H. Kayran and Ender M. Ekşioğlu

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

Abstract: A new exact method of measuring the Volterra kernels of finite order discrete nonlinear systems is presented. The kernels are rearranged in terms of multivariate cross-products in the vector form. The one-, two-, ..., and ℓ-dimensional kernel vectors are determined using a deterministic multilevel sequence with ℓ distinct levels at the input of the system. It is shown that the defined multilevel sequence with ℓ distinct levels is persistently exciting for a truncated Volterra filter with nonlinearities of polynomial degree ℓ. Examples demonstrating the rearrangement of the Volterra kernels and the novel method for estimation of the kernels are presented. Simulation results are given to illustrate the effectiveness of the proposed method.

Key Words: Nonlinear system identification, Volterra series, Volterra kernels, persistence of excitation.

I. INTRODUCTION

Nonlinear system identification is important due to the shortcoming of linear models when applied to inherently nonlinear problems which are abundant in real life applications [1]-[9]. Typical examples include trying to relate two signals whose significant spectral components do not overlap, satellite communication channels [3]-[5] where amplifiers are operated near saturation or magnetic recording channels [6]. The truncated (or "doubly finite") Volterra series representation constitutes an appealing nonlinear system model, since the output is linearly dependent on the kernel parameters, hence making the identification process mathematically tractable [1], [7]. The most common class of Volterra system identification methods include cross-correlation methods based on random inputs [8]-[15]. In this class some effort has been directed towards Gaussian inputs [2], [13] because their higher order moments are tractable. However the results are non-trivial and prone to error due to their sequential nature where error in one kernel estimate propagates to the next. In similar vein use of phase shfit keying (PSK) inputs which also have many vanishing moments has been considered [15]. With these sequences, the Volterra kernels are made orthogonal to each other and they are estimated separately. However, still the closed form results depend on higher order cumulants and hence necessitate relatively long input sequences. Harmonic probing methods using multitone signals to determine frequency domain Volterra kernels have also been developed [16], [17]. Another attractive approach [18] is the use of pseudorandom multilevel sequences (PRMS). Encouraged by the use of pseudorandom binary sequences (PRBS) for linear system identification, PRMS which can be chosen to be persistently exciting for any finite order Volterra system were considered for nonlinear system identification. However, as stated in [18] PRMS include input sequences which are redundant for the identification of a regular Volterra system. The condition on the order of the PRMS to guarantee persistence of excitation is necessary for the extended Volterra filter, which is introduced using Kronecker products [18]. However, this condition is sufficient but not necessary for the regular Volterra filter and this introduces redundancy. The PRMS includes input sequences which are unnecessary when the system under consideration is a regular Volterra filter.

In this work, we focus on deterministic excitation sequences for the identification of nonlinear systems modelled using the truncated Volterra series representation. A similar approach was used in [19] for the continuous-time case and it was shown that a pth-order kernel can be completely determined using p distinct impulses at the input of the system. This method was applied to the discrete-time two-dimensional (2-D) quadratic filters in [20]. However, for higher order systems because of the high complexity of this approach no practical applications were devised [1, pp.38-39].

This paper proposes a novel partitioning of the Volterra kernels, resulting in simple closed form solutions when deterministic multilevel input sequences are used. Kernels are estimated separately hence error propagation is prevented. A scheme where kernels are calculated using past outputs rather than using lower order kernel estimates provides robustness to noise and simplicity of implementation. Special matrices, which can be calculated recursively are utilized to form the input ensembles and to pick up the correct past outputs which should be subtracted. These matrices supply us with a closed form formula, and the high complexity problem which impeded the use of the algorithm in [19] for higher order systems is overcome. Our algorithm utilizes only the necessary input sequences and eliminates redundancy. The proposed identification algorithm is simulated for two Volterra systems of different orders and memory lengths, and its performance is compared with existing algorithms [15], [18].

The paper is organized is as follows: Section II describes the novel partitioning of the Volterra kernels, which is called as the multivariate kernel vector representation and proves the equivalence between this setup and the regular Volterra parameterizations. Section III describes the method to identify the Volterra kernel vectors using multilevel deterministic input signals. The algorithm is first described for 1-D kernel vectors, then for 2-D and 3-D kernel vectors and the necessary matrices are defined in a step-by-step fashion. Then, the algorithm is generalized to the identification of the ℓ-D kernel vectors. Section IV discusses the persistence of excitation of the multilevel input sequence. The condition number of the matrices whose inverses appear in the algorithm and the problem of ill-conditioning are discussed. Section V contains numerical studies demonstrating the algorithm and comparing its performance to PRMS and PSK excitations. Lastly, comments and concluding remarks are provided in Section VI.

II. MULTIVARIATE KERNEL VECTOR REPRESENTATION

In this section, the M^{th}-order discrete Volterra system is reformulated in terms of the multivariate cross-term complexity. The M^{th}-order system $\mathcal{N}[\cdot]$ under consideration can be modelled by the following triangular representation:

$$\begin{aligned} y(n) &= \mathcal{N}[x(n)] \\ &= \sum_{k=1}^{M} \sum_{i_1=0}^{N} \sum_{i_2=i_1}^{N} \cdots \sum_{i_k=i_{k-1}}^{N} b_k\left(i_1, i_2, \ldots, i_k\right) x(n-i_1) x(n-i_2) \cdots x(n-i_k) \end{aligned} \tag{1}$$

where N is the memory length of the nonlinear system and $b_k\left(i_1, i_2, \ldots, i_k\right)$ is the triangular Volterra kernel of degree k [1]. In this section, a new representation for the Volterra system is developed utilizing a rearrangement of the Volterra kernels. The nonlinear system output $y(n)$ is decomposed as the sum of the outputs of the M different multivariate cross-term nonlinear subsystems, $\mathcal{H}^{(\ell)}$; that is

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

in which

$$y^{(\ell)}(n) = \mathcal{H}^{(\ell)}[x(n)] \tag{2b}$$

$$\mathcal{H}^{(\ell)}[x(n)] = \begin{cases} \sum\limits_{i=0}^{N} \mathbf{h}^{(1)^T}(i)\, \mathbf{x}^{(1)}(n-i) & \text{for} \quad \ell = 1 \\ \sum\limits_{q_1=1}^{Q_1} \cdots \sum\limits_{q_{\ell-1}=1}^{Q_{\ell-1}} \sum\limits_{i=0}^{N-\bar{q}_{\ell-1}} \mathbf{h}^{(\ell)^T}(q_1, \ldots, q_{\ell-1}; i)\, \mathbf{x}^{(\ell)}(q_1, \ldots, q_{\ell-1}; n-i) & \text{for} \quad 2 \leqslant \ell \leqslant M \end{cases} \tag{2c}$$

where $Q_j = N + 1 + j - \ell - \bar{q}_{j-1}$ for $j = 1, \ldots, \ell - 1$ with $\bar{q}_m = q_1 + \cdots + q_m$ and $\bar{q}_0 = 0$. In this representation, the symbol $\mathcal{H}^{(\ell)}[\cdot]$, which represents ℓ summations, is called as an ℓ-D cross-term Volterra subsystem and $\mathbf{h}^{(\ell)}(q_1, \ldots, q_{\ell-1}; i)$ is called as an ℓ-D kernel vector. The 1-D kernel vectors $\mathbf{h}^{(1)}(i)$ and the corresponding input vectors $\mathbf{x}^{(1)}(n)$ can be formulated in terms of the triangular kernels and the input signal $x(n)$, respectively.

$$\mathbf{h}^{(1)}(i) = [b_1(i)\; b_2(i,i)\; \cdots\; b_M(i, \ldots, i)]^T \tag{3a}$$

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

The ℓ-D input vector in (2c) can be expressed in the following form:

$$\mathbf{x}^{(\ell)}\left(q_1,\ldots,q_{\ell-1};n\right)=\begin{bmatrix} x_\ell^{(\ell)}(q_1,\ldots,q_{\ell-1};n) \\ \mathbf{x}_{\ell+1}^{(\ell)}(q_1,\ldots,q_{\ell-1};n) \\ \vdots \\ \mathbf{x}_M^{(\ell)}(q_1,\ldots,q_{\ell-1};n) \end{bmatrix} \tag{4a}$$

in which

$$\mathbf{x}_k^{(\ell)}(q_1,\ldots,q_{\ell-1};n)\equiv\left[x_k^{(p_1,\cdots,p_\ell)}(q_1,\ldots,q_{\ell-1};n)\right]_{\sigma(p_1,\ldots,p_\ell)}. \tag{4b}$$

The subinput vectors $\mathbf{x}_k^{(\ell)}(q_1,\ldots,q_{\ell-1};n)$'s for $k=\ell,\ell+1,\ldots,M$ in (4a) consist of all possible inputs of degree k,

$$x_k^{(p_1,\cdots,p_\ell)}(q_1,\ldots,q_{\ell-1};n)=x^{p_1}(n)\,x^{p_2}(n-q_1)\,\cdots\,x^{p_\ell}(n-q_1-\cdots-q_{\ell-1}). \tag{5}$$

$\sigma(p_1,p_2,\ldots,p_\ell)$ defines $\binom{k-1}{\ell-1}=\binom{k-1}{k-\ell}=C_{k-1,\ell-1}$ combinations for $p_i=1,2,\ldots,k-\ell$ and $p_1+p_2+\cdots+p_\ell=k$ and $k=\ell,\ell+1,\ldots,M$. Hence, we get the $C_{k-1,\ell-1}$ terms of the subinput vector in (4b). For example, for $\ell=3$ and $k=5$, all possible combinations ($C_{4,2}=6$) can be written as $\sigma(p_1,p_2,p_3)=(1,1,3),\ (1,2,2),\ (1,3,1),\ (2,2,1),\ (2,1,2),\ (3,1,1)$. Therefore, the ℓ-D input vector in (4a) has a total of $C_{M,\ell}=\sum_{k=\ell}^{M}C_{k-1,\ell-1}$ terms. Similarly, the corresponding ℓ-D kernel vector in (2c) can be rewritten in terms of subkernels as

$$\mathbf{h}^{(\ell)}(q_1,\ldots,q_{\ell-1};i)=\begin{bmatrix} h_\ell^{(\ell)}(q_1,\ldots,q_{\ell-1};i) \\ \mathbf{h}_{\ell+1}^{(\ell)}(q_1,\ldots,q_{\ell-1};i) \\ \vdots \\ \mathbf{h}_M^{(\ell)}(q_1,\ldots,q_{\ell-1};i) \end{bmatrix} \tag{6a}$$

in which

$$\mathbf{h}_k^{(\ell)}(q_1,\ldots,q_{\ell-1};i)\equiv\left[h_k^{(p_1,p_2,\ldots p_\ell)}(q_1,\ldots,q_{\ell-1};i)\right]_{\sigma(p_1,p_2,\ldots,p_\ell)} \tag{6b}$$

where the subkernel vector $\mathbf{h}_k^{(\ell)}(q_1,\ldots,q_{\ell-1};i)$ consists of all possible kernels of degree k with ℓ cross-terms corresponding to the subinput vector $\mathbf{x}_k^{(\ell)}(q_1,\ldots,q_{\ell-1};n-i)$. However, there exists an equivalent triangular kernel $b_k(i_1,i_2,\ldots,i_k)$ as given in (1) for each component of the subkernel vector $\mathbf{h}_k^{(\ell)}(q_1,\ldots,q_{\ell-1};i)$ in (6a);

$$h_k^{(p_1,p_2,\ldots,p_\ell)}(q_1,\ldots,q_{\ell-1};i)=b_k(i_1,i_2,\ldots,i_k)$$

where

$$i_j = \begin{cases} i & \text{for} \quad 1 \leqslant j \leqslant p_1 \\ i + q_1 & \text{for} \quad p_1 + 1 \leqslant j \leqslant p_1 + p_2 \\ \quad \vdots & \qquad\qquad \vdots \\ i + q_1 + \cdots + q_{\ell-1} & \text{for} \quad p_1 + \cdots + p_{\ell-1} + 1 \leqslant j \leqslant p_1 + \cdots + p_\ell \end{cases}$$

Example 2.1: Let us give an example for the novel representation for a system with $M = 3$ and $N = 1$. The triangular representation is given by:

$$y(n) = b_1(0)x(n) + b_1 x(n-1) + b_2(0,0)x^2(n) + b_2(0,1)x(n)x(n-1) + b_2(1,1)x^2(n-1) \\ + b_3(0,0,0)x^3(n) + b_3(0,0,1)x^2(n)x(n-1) + b_3(0,1,1)x(n)x^2(n-1) + b_3(1,1,1)x^3(n-1) \tag{7}$$

Since $N = 1$, we do not have any three cross-term nonlinear subsystem. Then,

$$y(n) = \sum_{i=0}^{1} \mathbf{h}^{(1)^T}(i)\, \mathbf{x}^{(1)}(n-i) + \mathbf{h}^{(2)^T}(1;0)\, \mathbf{x}^{(2)}(1;n)$$

where

$$\mathbf{h}^{(1)}(i) = \begin{bmatrix} h_1^{(1)}(i) \\ h_2^{(1)}(i) \\ h_3^{(1)}(i) \end{bmatrix} = \begin{bmatrix} b_1(i) \\ b_2(i,i) \\ b_3(i,i,i) \end{bmatrix}, \text{for } i = 0, 1$$

$$\mathbf{x}^{(1)}(n-i) = \begin{bmatrix} x_1^{(1)}(n-i) \\ x_2^{(1)}(n-i) \\ x_3^{(1)}(n-i) \end{bmatrix} = \begin{bmatrix} x(n-i) \\ x^2(n-i) \\ x^3(n-i) \end{bmatrix}, \text{for } i = 0, 1$$

$$\mathbf{h}^{(2)}(1;0) = \begin{bmatrix} h_2^{(2)}(1;0) \\ \mathbf{h}_3^{(2)}(1;0) \end{bmatrix} = \begin{bmatrix} h_2^{(2)}(1;0) \\ h_3^{(2,1)}(1;0) \\ h_3^{(1,2)}(1;0) \end{bmatrix} = \begin{bmatrix} b_2(0,1) \\ b_3(0,0,1) \\ b_3(0,1,1) \end{bmatrix}$$

$$\mathbf{x}^{(2)}(1;n) = \begin{bmatrix} x_2^{(2)}(1;n) \\ \mathbf{x}_3^{(2)}(1;n) \end{bmatrix} = \begin{bmatrix} x_2^{(2)}(1;n) \\ x_3^{(2,1)}(1;n) \\ x_3^{(1,2)}(1;n) \end{bmatrix} = \begin{bmatrix} x(n)x(n-1) \\ x^2(n)x(n-1) \\ x(n)x^2(n-1) \end{bmatrix}$$

Remark 2.1: The cross-product kernel representation presented in this section does not increase the number of kernels in the triangular form, i.e., C_{tri} in (1). However, many identification procedures presented in the recent years require high number of unnecessary kernel computations, [18], [21]. The number of kernels in the cross-term representation is $C_{cross} = \sum_{\ell=1}^{M} \binom{M}{\ell}\binom{N+1}{\ell}$. It is not difficult to show that these numbers are the same, i.e., $C_{tri} = C_{cross} = \binom{N+M+1}{M} - 1$.

Remark 2.2: Another way to think about (2c) is as a parallel combination of 1-D LTI filters. If we define new filters $\mathbf{h}^{(\ell)}_{(q_1,\ldots,q_{\ell-1})}(n) = \mathbf{h}^{(\ell)}(q_1,\ldots,q_{\ell-1};n)$ and new signals $\mathbf{x}^{(\ell)}_{(q_1,\ldots,q_{\ell-1})}(n) = \mathbf{x}^{(\ell)}(q_1,\ldots,q_{\ell-1};n)$, then (2c) can be rewritten as

$$y^{(\ell)}(n) = \sum_{q_1=1}^{Q_1} \cdots \sum_{q_{\ell-1}=1}^{Q_{\ell-1}} \mathbf{h}^{(\ell)^T}_{(q_1,\ldots,q_{\ell-1})}(n) * \mathbf{x}^{(\ell)}_{(q_1,\ldots,q_{\ell-1})}(n)$$

where $*$ is the convolution operation. A similar break-up of the Volterra system was proposed in [22]. It is called as the *Diagonal Coordinate Representation* (DCS) for the discrete-time finite memory Volterra filters. There, the output of the n^{th}-order triangular Volterra kernel is expressed as a sum of 1-D convolutions, where the linear filter coefficients are obtained from the diagonals of the sampled hypercube defined by the triangular Volterra kernels. This representation parallels the one used for the identification of continuous-time Volterra filters given in [19]. Our representation is similar to the DCS, in that the linear filter coefficients in DCS are given in a coordinate system analogous to our delay variables, $(q_1,\ldots,q_\ell)$. However, we group the linear filters differently, introducing the concept of delay-wise dimensionality and cross-term subsystem rather than the multiplicational dimensionality of the triangular Volterra kernels. This novel grouping enables us to devise an exact closed form algorithm for identifying the Volterra kernels using deterministic multilevel sequences.

III. IDENTIFICATION OF THE KERNEL VECTORS USING MULTILEVEL INPUT SIGNALS

In this section, we derive an efficient algorithm to identify the kernel vectors $\mathbf{h}^{(\ell)}(q_1,\ldots,q_{\ell-1};i)$ defined in (6) by using multilevel input sequences with ℓ distinct impulses. Starting to identify the kernel vectors with one cross-term ($\ell = 1$), which will be called as the 1-D kernel vectors, all other kernel vectors with increasing cross-term complexity can be determined by using only the present and the appropriate previous outputs of the lower dimensional subsystems given in (2). In order to explain this identification technique, we first shall discuss the procedure for determining Volterra kernels of one-, two-, and three-dimensional subsystems. The generalization of the procedure to the ℓ-D subsystems will then be given.

A. 1-D Kernel Vectors and the Ensemble Input Matrix

The input-output relation of the 1-D subsystem $\mathcal{H}^{(1)}[\cdot]$ can be written as in (2c)

$$y^{(1)}(n) = \mathcal{H}^{(1)}[x(n)] = \sum_{i=0}^{N} \mathbf{h}^{(1)^T}(i)\, \mathbf{x}^{(1)}(n-i) \tag{8}$$

It is possible to show that multilevel single impulses, $x^{(1)}(m_1;n) = a_{m_1}\delta(n)$, for $m_1 = 1, 2, \ldots, \binom{M}{1}$ can be used to obtain the 1-D kernel vectors in (3a). Using the cross-term representation in (2), it is trivial to prove that the higher dimensional outputs are zero for these multilevel single impulses, i.e., $y^{(\ell)}(n) = 0$ for $\ell > 1$. Hence, the output of the M^{th}-order nonlinear system when the input is $x^{(1)}(m_1;n)$ is given by

$$\mathcal{N}\big[x^{(1)}(m_1;n)\big] = \sum_{i=0}^{N} \mathbf{h}^{(1)^T}(i)\mathbf{u}^{(1)}(m_1;n-i) \tag{9a}$$

in which

$$\begin{aligned} \mathbf{u}^{(1)}(m_1;n) &= \Big[x^{(1)}(m_1;n)\; x^{(1)^2}(m_1;n)\; \cdots\; x^{(1)^M}(m_1;n)\Big]^T \\ &= \big[a_{m_1}\; a_{m_1}^2\; \cdots\; a_{m_1}^M\big]^T \delta(n) \end{aligned} \tag{9b}$$

where m_1 denotes the ensemble index of the input sequence. Now we can write all $\binom{M}{1}$ ensemble outputs in the matrix form as follows:

$$\mathbf{y}_e^{(1)}(n) = \mathcal{N}\big[\mathbf{x}_e^{(1)}(n)\big] = \mathcal{H}^{(1)}\big[\mathbf{x}_e^{(1)}(n)\big] = \sum_{i=0}^{N} \mathbf{U}_e^{(1)}(n-i)\, \mathbf{h}^{(1)}(i) \tag{10}$$

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

$$\mathbf{x}_e^{(1)}(n) = [a_1\; a_2\; \cdots\; a_M]^T \delta(n) \tag{11}$$

$$\mathbf{y}_e^{(1)}(n) = \Big[\mathcal{N}\big[x^{(1)}(1;n)\big]\; \mathcal{N}\big[x^{(1)}(2;n)\big]\; \cdots\; \mathcal{N}\big[x^{(1)}(M;n)\big]\Big]^T \tag{12}$$

and

$$\mathbf{U}_e^{(1)}(n) = \begin{bmatrix} \mathbf{u}^{(1)^T}(1;n) \\ \mathbf{u}^{(1)^T}(2;n) \\ \vdots \\ \mathbf{u}^{(1)^T}(M;n) \end{bmatrix} \tag{13}$$

where $\mathbf{u}^{(1)}(m_1;n)$ is defined in (9b). Hence, $\mathbf{U}_e^{(1)}(n-i)$ in (10) can be replaced with $\mathbf{U}_e^{(1)}(n-i) = \mathbf{U}_e^{(1)}\delta(n-i)$ and the matrix $\mathbf{U}_e^{(1)}$ is written as

$$\mathbf{U}_e^{(1)} = \begin{bmatrix} a_1 & a_1^2 & \cdots & a_1^M \\ a_2 & a_2^2 & \cdots & a_2^M \\ \vdots & \vdots & \ddots & \vdots \\ a_M & a_M^2 & \cdots & a_M^M \end{bmatrix} \tag{14}$$

Hence, we get

$$\mathbf{y}_e^{(1)}(n) = \sum_{i=0}^{N} \mathbf{U}_e^{(1)}\,\mathbf{h}^{(1)}(i)\,\delta(n-i) = \mathbf{U}_e^{(1)}\,\mathbf{h}^{(1)}(n) \tag{15}$$

and therefore, provided the inverse of the $M \times M$ matrix $\mathbf{U}_e^{(1)}$ exists, the 1-D kernel vectors can be obtained as

$$\mathbf{h}^{(1)}(n) = \left[\mathbf{U}_e^{(1)}\right]^{-1}\mathbf{y}_e^{(1)}(n) \qquad \text{for} \quad n = 0, 1, \ldots, N \tag{16}$$

It is also possible to write the solution for all 1-D kernel vectors together in the matrix form as follows:

$$\mathbf{H}^{(1)} = \left[\mathbf{U}_e^{(1)}\right]^{-1}\mathbf{Y}_e^{(1)} \tag{17}$$

where $\mathbf{H}^{(1)} = \left[\mathbf{h}^{(1)}(0)\ \mathbf{h}^{(1)}(1)\ \cdots\ \mathbf{h}^{(1)}(N)\right]$ and $\mathbf{Y}_e^{(1)} = \left[\mathbf{y}_e^{(1)}(0)\ \mathbf{y}_e^{(1)}(1)\ \cdots\ \mathbf{y}_e^{(1)}(N)\right]$. This result shows that all $M(N+1)$ 1-D kernels with one cross-term can be determined by using only the inverse of an $M \times M$ ensemble matrix times the ensemble output matrix as shown in (17). Note that the linear FIR filter identification via the impulse response is covered by this method as the special case $M = 1$.

Example 3.1: Consider as an example the nonlinear system $M = 3$ and $N = 1$, described by

$$\begin{aligned} y(n) =& x(n) - x(n-1) + 2x^2(n) - 4x(n)x(n-1) - 2x^2(n-1) \\ &+ 3x^3(n) + 4x^2(n)x(n-1) + 5x(n)x^2(n-1) - 3x^3(n-1) \end{aligned} \tag{18}$$

For this section, our aim is to calculate the 1-D kernel vectors, $\mathbf{h}^{(1)}(0) = [1\ 2\ 3]^T$ and $\mathbf{h}^{(1)}(1) = [-1 - 2 - 3]^T$, from the 1-D input sequences and the corresponding outputs. We choose $a_1 = 1$, $a_2 = -1$, $a_3 = 2$, and the 1-D input ensemble to the system becomes

$$\mathbf{x}_e^{(1)}(n) = \begin{bmatrix} 1 \\ -1 \\ 2 \end{bmatrix} \delta(n) \tag{19}$$

The 1-D output ensemble defined in (12) becomes

$$\mathbf{y}_e^{(1)}(n) = \begin{bmatrix} 6 \\ -2 \\ 34 \end{bmatrix} \delta(n) + \begin{bmatrix} -6 \\ 2 \\ -34 \end{bmatrix} \delta(n-1) \tag{20}$$

Using (14) and (19), the matrix $\mathbf{U}_e^{(1)}$ is given as,

$$\mathbf{U}_e^{(1)} = \begin{bmatrix} 1 & 1 & 1 \\ -1 & 1 & -1 \\ 2 & 4 & 8 \end{bmatrix} \tag{21}$$

Using (16), (20) and (21) we can calculate the 1-D kernel vectors as,

$$\mathbf{h}^{(1)}(0) = \left[\mathbf{U}_e^{(1)}\right]^{-1} \begin{bmatrix} 6 \\ -2 \\ 34 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \text{ and } \mathbf{h}^{(1)}(1) = \left[\mathbf{U}_e^{(1)}\right]^{-1} \begin{bmatrix} -6 \\ 2 \\ -34 \end{bmatrix} = \begin{bmatrix} -1 \\ -2 \\ -3 \end{bmatrix} \tag{22}$$

Remark 3.1: In order to guarantee the Vandermonde like input matrix $\mathbf{U}_e^{(1)}$ in (14) to be nonsingular, the coefficients of the multilevel ensemble inputs must be chosen to be distinct and nonzero, i.e., $a_i \neq 0$ and $a_i \neq a_j$, $\forall\ i \neq j$. In the following sections we will deal with the inverses of input matrices of similar form for the higher order cross-terms. The invertibility of the input matrix $\mathbf{U}_e^{(1)}$ is sufficient to assure the invertibility of the higher order input matrices. However, in addition to the nonsingularity constraint, we have to take into consideration the condition number of these input matrices $\mathbf{U}_e^{(\ell)}$, because these condition numbers will determine the effect of the observation noise on the accuracy of the kernel estimates. Hence, the values of the parameter vector, $\mathbf{a} \hat{=} [a_1\ a_2\ \cdots\ a_M]^T$ are determined to optimize a collective criterion for the condition numbers of $\mathbf{U}_e^{(\ell)}$ for $\ell = 1, 2, \ldots, M$ which will be discussed in the following sections.

B. 2-D Kernel Vectors and Ensemble Input Formation Matrices

At this stage, we are interested in computing the 2-D kernel vectors namely $\mathbf{h}^{(2)}(q_1; i)$ for $q_1 = 1, \ldots, N$ and $i = 0, 1, \ldots, N - q_1$ using the 2-D ensemble inputs which are related to the 1-D ensemble ones. Indeed, the following 2-D sequence consisting of two impulses with distinct amplitudes will only excite the 2-D kernel vector $\mathbf{h}^{(2)}(q_1; n - q_1)$ and the 1-D kernel vectors $\mathbf{h}^{(1)}(n)$ and $\mathbf{h}^{(1)}(n - q_1)$.

$$x^{(2)}((m_1, m_2), q_1; n) = x^{(1)}(m_1; n) + x^{(1)}(m_2; n - q_1)$$
$$\text{for } m_1 = 1, \ldots, M-1;\ m_2 = m_1 + 1, \ldots, M \quad (23)$$

where $x^{(1)}(m_i; n) = a_{m_i}\delta(n),\ i = 1, 2$ and the amplitudes are assigned by using the parameters of the 1-D input ensemble sequences, $a_{m_1}, a_{m_2} \in \{a_1, \ldots, a_M\}$. It is possible to show that the 2-D input signal in (23) does not excite the Volterra kernels having more than two cross-terms, i.e., $\mathcal{H}^{(\ell)}\big[x^{(2)}((m_1; m_2), q_1; n)\big] = 0$ for $\ell \geqslant 3$. Hence, the output for the input in (23) can be written as

$$\mathcal{N}\big[x^{(2)}\big((m_1, m_2), q_1; n\big)\big] = \mathcal{H}^{(1)}\big[x^{(2)}\big((m_1, m_2), q_1; n\big)\big] + \mathcal{H}^{(2)}\big[x^{(2)}\big((m_1, m_2), q_1; n\big)\big] \quad (24)$$

Here,

$$\begin{aligned}\mathcal{H}^{(2)}\big[x^{(2)}\big((m_1, m_2), q_1; n\big)\big] &= v^{(2,2)}\big((m_1, m_2), q_1; n\big)\\ \mathcal{H}^{(1)}\big[x^{(2)}\big((m_1, m_2), q_1; n\big)\big] &= v^{(2,1)}\big((m_1, m_2), q_1; n\big)\\ &= v^{(1,1)}(m_1; n) + v^{(1,1)}(m_2; n - q_1)\end{aligned}$$

with $v^{(1,1)}(m_i; n) = \mathcal{H}^{(1)}\big[x^{(1)}(m_i; n)\big]$. The notation $v^{(i,j)}(\cdots)$ will denote the output of the j-D subsystem to the i-D input signal, i.e., $v^{(i,j)}(\cdots) = \mathcal{H}^{(j)}\big[x^{(i)}(\cdots)\big]$. The output of the 2-D subsystem can be obtained from (24).

$$v^{(2,2)}\big((m_1, m_2), q_1; n\big) = \mathcal{N}\big[x^{(2)}\big((m_1, m_2), q_1; n\big)\big] - \big(v^{(1,1)}(m_1; n) + v^{(1,1)}(m_2; n - q_1)\big) \quad (25)$$

(25) can be verified by direct substitution in (2). It is also interesting to note that the output of the 2-D subsystem is obtained by subtracting the appropriate previously computed 1-D outputs from the overall nonlinear system output. Since the 2-D kernel vectors have $C_{M,2}$ components, we must use $C_{M,2}$ distinct ensemble inputs as given in (23). The

main idea of the proposed method is based upon the observation of the outputs for these particular 2-D ensemble inputs which are given in the matrix form as follows:

$$\mathbf{x}_e^{(2)}(q_1;n) = \mathbf{T}_{2,1}^{(M)}\mathbf{x}_e^{(1)}(n) + \mathbf{T}_{2,2}^{(M)}\mathbf{x}_e^{(1)}(n-q_1) \tag{26}$$

where $\mathbf{x}_e^{(1)}(n) = [a_1 \;\; a_2 \;\; \cdots \;\; a_M]^T\delta(n)$ and the constant $\mathbf{T}_{2,1}^{(M)}$ and $\mathbf{T}_{2,2}^{(M)}$ matrices with $C_{M,2}$ rows and $C_{M,1}$ columns are used to determine the necessary $C_{M,2}$ combinations of $C_{M,1} = M$ ensemble inputs when taken two at a time. A recursive algorithm is given in Table 1 for constructing these $\mathbf{T}_{\ell,k}^{(M)}$matrices for $\ell = 1, 2, \ldots, M$ and $k = 1, 2, \ldots, \ell$. Each row of these ensemble input formation matrices has only one nonzero unity element. Similar to the single-input single-output case in (24), using the 2-D ensemble input vector in (26), we can determine the corresponding output vectors of 1-D and 2-D subsystems, $\mathbf{v}_e^{(2,1)}(q_1;n) = \mathcal{H}^{(1)}\left[\mathbf{x}_e^{(2)}(q_1;n)\right]$ and $\mathbf{v}_e^{(2,2)}(q_1;n) = \mathcal{H}^{(2)}\left[\mathbf{x}_e^{(2)}(q_1;n)\right]$, respectively. Using (24) and (26), 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

$$\begin{aligned}\mathbf{v}_e^{(2,1)}(q_1;n) &= \mathcal{H}^{(1)}\left[\mathbf{T}_{2,1}^{(M)}\mathbf{x}_e^{(1)}(n)\right] + \mathcal{H}^{(1)}\left[\mathbf{T}_{2,2}^{(M)}\mathbf{x}_e^{(1)}(n-q_1)\right] \\ &= \mathbf{T}_{2,1}^{(M)}\mathbf{v}_e^{(1,1)}(n) + \mathbf{T}_{2,2}^{(M)}\mathbf{v}_e^{(1,1)}(n-q_1)\end{aligned} \tag{27}$$

The response of the 2-D subsystem can be obtained by subtracting the response of the 1-D subsystem from the nonlinear system output vector $\mathbf{y}_e^{(2)}(q_1;n) = \mathcal{N}[\mathbf{x}_e^{(2)}(q_1;n)]$ as follows:

$$\begin{aligned}\mathbf{v}_e^{(2,2)}(q_1;n) &= \mathbf{y}_e^{(2)}(q_1;n) - \mathbf{v}_e^{(2,1)}(q_1;n) \\ &= \mathbf{y}_e^{(2)}(q_1;n) - \left[\mathbf{T}_{2,1}^{(M)}\mathbf{v}_e^{(1,1)}(n) + \mathbf{T}_{2,2}^{(M)}\mathbf{v}_e^{(1,1)}(n-q_1)\right]\end{aligned} \tag{28}$$

It is possible to write the 2-D subsystem equation for the ensemble input case,

$$\mathbf{v}_e^{(2,2)}(q_1;n) = \sum_{i=0}^{N-q_1} \mathbf{U}_e^{(2)}(q_1;n-i)\mathbf{h}^{(2)}(q_1;i) \tag{29}$$

Similar to the 1-D case, $\mathbf{U}_e^{(2)}(q_1;n-i)$ is replaced with $\mathbf{U}_e^{(2)}\delta(n-q_1-i)$. $\mathbf{U}_e^{(2)}$ has the dimension $C_{M,2} \times C_{M,2}$ and it is written in terms of $a_1,\ a_2, \ldots, a_M$ gain factors,

$$\mathbf{U}_e^{(2)} = \begin{bmatrix} \mathbf{u}^{(2)^T}(1,1) \\ \mathbf{u}^{(2)^T}(1,2) \\ \vdots \\ \mathbf{u}^{(2)^T}(M-1,M) \end{bmatrix} \tag{30a}$$

where

$$\mathbf{u}^{(2)^T}(i,j) = \left[u_2^{(2)}(i,j) \vdots \mathbf{u}_3^{(2)^T}(i,j) \vdots \cdots \vdots \mathbf{u}_M^{(2)^T}(i,j) \right] \tag{30b}$$

with

$$\begin{aligned}
u_2^{(2)}(i,j) &= [a_i a_j]_{1\times 1} \\
\mathbf{u}_3^{(2)^T}(i,j) &= \left[a_i a_j^2 \quad a_i^2 a_j\right]_{1\times 2} \\
\mathbf{u}_4^{(2)^T}(i,j) &= \left[a_i a_j^3 \quad a_i^2 a_j^2 \quad a_i^3 a_j\right]_{1\times 3} \\
&\;\vdots \\
\mathbf{u}_M^{(2)^T}(i,j) &= \left[a_i a_j^{M-1} \quad a_i^2 a_j^{M-2} \quad \cdots \quad a_i^{M-1} a_j\right]_{1\times M}
\end{aligned} \tag{30c}$$

As an example, for $M = 4$ and $\ell = 2$, the matrix $\mathbf{U}_e^{(2)}$ will be given as

$$\mathbf{U}_e^{(2)} = \begin{bmatrix}
a_1a_2 & a_1a_2^2 & a_1^2a_2 & a_1a_2^3 & a_1^2a_2^2 & a_1^3a_2 \\
a_1a_3 & a_1a_3^2 & a_1^2a_3 & a_1a_3^3 & a_1^2a_3^2 & a_1^3a_3 \\
a_1a_4 & a_1a_4^2 & a_1^3a_4 & a_1a_4^3 & a_1^2a_4^2 & a_1^3a_4 \\
a_2a_3 & a_2a_3^2 & a_2^2a_3 & a_2a_3^3 & a_2^2a_3^2 & a_2^3a_3 \\
a_2a_4 & a_2a_4^2 & a_2^2a_4 & a_2a_4^3 & a_2^2a_4^2 & a_2^3a_4 \\
a_3a_4 & a_3a_4^2 & a_3^2a_4 & a_3a_4^3 & a_3^2a_4^2 & a_3^3a_4
\end{bmatrix}$$

The 2-D Volterra kernel vectors are obtained by using (28) and (29),

$$\mathbf{h}^{(2)}(q_1; n-q_1) = \left[\mathbf{U}_e^{(2)}\right]^{-1} \mathbf{v}_e^{(2,2)}(q_1; n) \tag{31}$$

for $q_1 = 1, \ldots, N$ and $n = q_1, q_1+1, \ldots, N$, provided the inverse exists.

Example 3.2: Consider the same nonlinear system given in (18). For this section,the problem is to determine the 2-D kernel vector $\mathbf{h}^{(2)}(1;0) = [-4\ 4\ 5]^T$. Using the 1-D input ensemble vector $\mathbf{x}_e^{(1)}(n)$ in (19) and constructing the constant $\mathbf{T}_{2,1}^{(2)}$ and $\mathbf{T}_{2,2}^{(2)}$ ensemble input formation matrices from Table 1, the 2-D input ensemble in (26) is found as

$$\begin{aligned}
\mathbf{x}_e^{(2)}(1;n) &= \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \\ 2 \end{bmatrix} \delta(n) + \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \\ 2 \end{bmatrix} \delta(n-1) \\
&= \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix} \delta(n) + \begin{bmatrix} -1 \\ 2 \\ 2 \end{bmatrix} \delta(n-1)
\end{aligned} \tag{32}$$

For this particular 2-D input ensemble in (32), the output of the nonlinear system in (18) becomes

$$\mathbf{y}_e^{(2)}(1;n) = \begin{bmatrix} 6 \\ 6 \\ -2 \end{bmatrix} \delta(n) + \begin{bmatrix} -5 \\ 46 \\ 38 \end{bmatrix} \delta(n-1) + \begin{bmatrix} 2 \\ -34 \\ -34 \end{bmatrix} \delta(n-2) \tag{33}$$

We can determine the response of the 1-D subsystem to the 2-D input ensemble in (32) from the 1-D output ensemble given in (20) as follows:

$$\begin{aligned} \mathbf{v}_e^{(2,1)}(1;n) &= \mathbf{T}_{2,1}^{(2)}\mathbf{y}_e^{(1)}(n) + \mathbf{T}_{2,2}^{(2)}\mathbf{y}_e^{(1)}(n-1) \\ &= \begin{bmatrix} 6 \\ 6 \\ -2 \end{bmatrix} \delta(n) + \begin{bmatrix} -8 \\ 28 \\ 36 \end{bmatrix} \delta(n-1) + \begin{bmatrix} 2 \\ -34 \\ -34 \end{bmatrix} \delta(n-2) \end{aligned} \tag{34}$$

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

$$\begin{aligned} \mathbf{v}_e^{(2,2)}(1;n) &= \mathbf{y}_e^{(2)}(1;n) - \mathbf{v}_e^{(2,1)}(1;n) \\ &= \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \delta(n) + \begin{bmatrix} 3 \\ 18 \\ 2 \end{bmatrix} \delta(n-1) + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \delta(n-2) \end{aligned} \tag{35}$$

The 2-D Volterra kernel $\mathbf{h}^{(2)}(1;0)$ is determined by substituting (35) into (31)

$$\begin{aligned} \mathbf{h}^{(2)}(1;0) &= \left[\mathbf{U}_e^{(2)}\right]^{-1}\mathbf{v}_e^{(2,2)}(1;1) \\ &= \begin{bmatrix} -1 & 1 & -1 \\ 2 & 4 & 2 \\ -2 & -4 & 2 \end{bmatrix}^{-1} \begin{bmatrix} 3 \\ 18 \\ 2 \end{bmatrix} = \begin{bmatrix} -4 \\ 4 \\ 5 \end{bmatrix} \end{aligned} \tag{36}$$

The constant matrix $\mathbf{U}_e^{(2)}$ is calculated using $\mathbf{x}_e^{(1)}(n)$ and (30).

C. 3-D Kernel Vectors and Output Pick-up Matrices

The 3-D kernel vectors $\mathbf{h}^{(3)}(q_1, q_2; i)$ for $q_1 = 1, \ldots, N-1$, $q_2 = 1, \ldots, N - q_1$ and $i = 0, 1, \ldots, N - q_1 - q_2$ are determined by using the 3-D ensemble responses along with the responses of the 1-D and 2-D subsystems. The following 3-D ensemble input with three distinct impulses excites only 1-D, 2-D and 3-D subsystems.

$$\mathbf{x}_e^{(3)}(q_1, q_2; n) = \mathbf{T}_{3,1}^{(M)}\mathbf{x}_e^{(1)}(n) + \mathbf{T}_{3,2}^{(M)}\mathbf{x}_e^{(1)}(n - q_2) + \mathbf{T}_{3,3}^{(M)}\mathbf{x}_e^{(1)}(n - q_1 - q_2) \tag{37}$$

The output of the nonlinear system can be written as the sum of the outputs of the exited subsystems,

$$\mathbf{y}_e^{(3)}(q_1, q_2; n) = \mathcal{N}\left[\mathbf{x}_e^{(3)}(q_1, q_2; n)\right] = \sum_{i=1}^{3} \mathbf{v}_e^{(3,i)}(q_1, q_2; n) \tag{38}$$

where the output of the 1-D subsystem for the 3-D ensemble input can be written as the sum of the 1-D ensemble outputs

$$\mathbf{v}_e^{(3,1)}(q_1, q_2; n) = \sum_{j=1}^{C_{3,1}} \mathbf{S}_{31,j}^{(M)}\mathbf{v}_e^{(1,1)}(n - n_j^{(3,1)}) \tag{39}$$

where the matrices $\mathbf{S}_{31,j}^{(M)}$ for $j = 1, 2, 3$ which have $C_{M,3}$ rows and $C_{M,1}$ columns, are used to pick up the appropriate 1-D ensemble output values. Each row of these matrices has only one nonzero unity element. The superscript M denotes the order of the nonlinearity of the nonlinear system $\mathcal{N}$. The subscripts ℓ, k and j in $\mathbf{S}_{\ell k,j}^{(M)}$ indicate the dimension of the ensemble input, the dimension of the subsystem, and the index number of the excited k-D outputs, respectively. A recursive algorithm is given in Table 2 for constructing the $\mathbf{S}_{\ell k,j}^{(M)}$ matrices for $\ell = 1, \ldots, M$, $\quad k = 1, \ldots, \ell - 1$ and $j = 1, \ldots, C_{\ell,k}$. The necessary delays $n_j^{(3,1)}$ in (39) for $j = 1, 2, 3$ can be obtained by using the formation matrix $\mathbf{T}_{1,1}^{(3)} = \mathbf{I}_{C_{3,1}} = \mathbf{I}_3$.

$$\mathbf{n}^{(3,1)} = \mathbf{T}_{1,1}^{(3)}\mathbf{q}^{(3)} \tag{40}$$

where $\mathbf{n}^{(3,1)} = \left[n_1^{(3,1)}\ n_2^{(3,1)}\ n_3^{(3,1)}\right]^T$ and $\mathbf{q}^{(3)} = [0\ \ q_2\ \ q_1 + q_2]^T$. It is also possible to determine the responses of the 2-D subsystem for the 3-D ensemble inputs,

$$\mathbf{v}_e^{(3,2)}(q_1, q_2; n) = \sum_{j=1}^{C_{3,2}} \mathbf{S}_{32,j}^{(M)}\mathbf{v}_e^{(2,2)}(q_j^{(3,2)}; n - n_j^{(3,2)}) \tag{41}$$

where the output pick up matrices $\mathbf{S}_{32,j}^{(M)}$ for $j = 1, 2, 3$ which have $C_{M,3}$ rows and $C_{M,2}$ columns are used to determine the appropriate 2-D ensemble output values for $\mathbf{x}_e^{(3)}(q_1, q_2; n)$. The necessary 2-D parameters $q_j^{(3,2)}$ and $n_j^{(3,2)}$ can be obtained by using the input formation matrices $\mathbf{T}_{2,1}^{(3)}$ and $\mathbf{T}_{2,2}^{(3)}$ as follows

$$\mathbf{q}^{(3,2)} = \left[\mathbf{T}_{2,2}^{(3)} - \mathbf{T}_{2,1}^{(3)}\right]\mathbf{q}^{(3)} \quad \text{and} \quad \mathbf{n}^{(3,2)} = \mathbf{T}_{2,1}^{(3)}\mathbf{q}^{(3)} \tag{42}$$

where $\mathbf{q}^{(3,2)} = \left[q_1^{(3,2)}\ q_2^{(3,2)}\ q_3^{(3,2)}\right]^T$ and $\mathbf{n}^{(3,2)} = \left[n_1^{(3,2)}\ n_2^{(3,2)}\ n_3^{(3,2)}\right]^T$. In a similar fashion to the equations for the 1-D and 2-D subsystems ((15), (29)), the output of the 3-D subsystem $\mathbf{v}_e^{(3,3)}(q_1, q_2; n)$ can be written as

$$\mathbf{v}_e^{(3,3)}(q_1, q_2; n) = \mathbf{U}_e^{(3)}\mathbf{h}^{(3)}(q_1, q_2; n - q_1 - q_2) \tag{43}$$

for $q_1 = 1, \ldots, N-1$, $q_2 = 1, \ldots, N - q_1$ and $n = q_1 + q_2, q_1 + q_2 + 1, \ldots, N$, where the matrix $\mathbf{U}_e^{(3)}$ has the dimensions $C_{M,3} \times C_{M,3}$ and can be written in terms of the gain factors in a way analogous to (13) and (30)

$$\mathbf{U}_e^{(3)} = \begin{bmatrix} \mathbf{u}^{(3)^T}(1,2,3) \\ \mathbf{u}^{(3)^T}(1,2,4) \\ \vdots \\ \mathbf{u}^{(3)^T}(M-2,\ M-1,\ M) \end{bmatrix} \tag{44a}$$

where

$$\mathbf{u}^{(3)^T}(i,j,k) = \left[u_3^{(3)}(i,j,k) \vdots \mathbf{u}_4^{(3)^T}(i,j,k) \vdots \cdots \vdots \mathbf{u}_M^{(3)^T}(i,j,k)\right] \tag{44b}$$

with

$$\begin{aligned} u_3^{(3)}(i,j,k) &= [a_i a_j a_k]_{1\times\binom{2}{2}} \\ \mathbf{u}_4^{(3)^T}(i,j,k) &= \left[a_i a_j a_k^2 \quad a_i a_j^2 a_k \quad a_i^2 a_j a_k\right]_{1\times\binom{3}{2}} \\ \mathbf{u}_5^{(3)^T}(i,j,k) &= \left[a_i a_j a_k^3 \quad a_i a_j^2 a_k^2 \quad \cdots \quad a_i^3 a_j a_k\right]_{1\times\binom{4}{2}} \\ &\vdots \\ \mathbf{u}_M^{(3)^T}(i,j,k) &= \left[a_i a_j a_k^{M-2} \quad a_i a_j^2 a_k^{M-3} \quad \cdots \quad a_i^{M-2} a_j a_k\right]_{1\times\binom{M-1}{2}} \end{aligned} \tag{44c}$$

As an example, for $M = 4$ and $\ell = 3$, the matrix $\mathbf{U}_e^{(3)}$ will be given as

$$\mathbf{U}_e^{(3)} = \begin{bmatrix} a_1a_2a_3 & a_1a_2a_3^2 & a_1a_2^2a_3 & a_1^2a_2a_3 \\ a_1a_2a_4 & a_1a_2a_4^2 & a_1a_2^2a_4 & a_1^2a_2a_4 \\ a_1a_3a_4 & a_1a_3a_4^2 & a_1a_3^2a_4 & a_1^2a_3a_4 \\ a_2a_3a_4 & a_2a_3a_4^2 & a_2a_3^2a_4 & a_2^2a_3a_4 \end{bmatrix}$$

From (38)-(43), we get

$$\mathbf{h}^{(3)}(q_1, q_2; n - q_1 - q_2) = \left[\mathbf{U}_e^{(3)}\right]^{-1}$$
$$\left[\mathbf{y}_e^{(3)}(q_1, q_2; n) - \left(\sum_{j=1}^{3} \mathbf{S}_{31,j}^{(M)} \mathbf{v}_e^{(1,1)}(n - n_j^{(3,1)}) + \sum_{j=1}^{3} \mathbf{S}_{32,j}^{(M)} \mathbf{v}_e^{(2,2)}(q_j^{(3,2)}; n - n_j^{(3,2)})\right)\right] \quad (45)$$

D. ℓ-D Kernel Vectors

We assume that all k-D kernel vectors $\mathbf{h}^{(k)}(q_1, \ldots, q_{k-1}; i)$ for $k = 1, 2, \ldots, \ell - 1$ are determined or that all the k-D output vectors $\mathbf{v}_e^{(k,k)}(q_1, \ldots, q_{k-1}; n)$ for $k = 1, 2, \ldots, \ell - 1$ are available. Now, we try to identify the ℓ-D kernel vectors by using the response of the nonlinear system for the ℓ-D ensemble input vector and all the previously computed subsystem outputs, $\mathbf{v}_e^{(k,k)}(q_1, \ldots, q_{k-1}; n)$ for $k = 1, 2, \ldots, \ell - 1$. Similar to 1-, 2-, and 3-D ensemble input vectors in (11), (26) and (37), the ℓ-D input ensemble vector can be written as

$$\mathbf{x}_e^{(\ell)}(q_1, \ldots, q_{\ell-1}; n) = \sum_{i=1}^{\ell} \mathbf{T}_{\ell,i}^{(M)} \mathbf{x}_e^{(1)}(n - n_i^{(\ell)}) \quad (46)$$

where $n_1^{(\ell)} = 0$ and $n_i^{(\ell)} = \sum_{j=1}^{i-1} q_{\ell-j}$ for $i = 2, \ldots, \ell$. The input formation matrices, $\mathbf{T}_{\ell,i}^{(M)}$'s can be constructed from the Table 1. The response of the nonlinear system to the ensemble input in (46) can be written in terms of the outputs of the subsystems,

$$\mathbf{y}_e^{(\ell)}(q_1, \ldots, q_{\ell-1}; n) = \mathcal{N}\left[\mathbf{x}_e^{(\ell)}(q_1, \ldots, q_{\ell-1}; n)\right] = \sum_{k=1}^{\ell} \mathbf{v}_e^{(\ell,k)}(q_1, \ldots, q_{\ell-1}; n) \quad (47)$$

where $\mathbf{v}_e^{(\ell,k)}(q_1,\ldots,q_{\ell-1};n)$ for $k=1,\ldots,\ell-1$ can be obtained from the previous subsystem outputs

$$\begin{aligned}
\mathbf{v}_e^{(\ell,1)}(q_1,\ldots,q_{\ell-1};n) &= \sum_{j=1}^{\binom{\ell}{1}} \mathbf{S}_{\ell 1,j}^{(M)} \mathbf{v}_e^{(1,1)}(n-n_j^{(\ell,1)}) \\
&\vdots \\
\mathbf{v}^{(\ell,k)}(q_1,\ldots,q_{\ell-1};n) &= \sum_{j=1}^{\binom{\ell}{k}} \mathbf{S}_{\ell 1,j}^{(M)} \mathbf{v}_e^{(k,k)}(q_{j,1}^{(\ell,k)},\ldots,q_{j,k-1}^{(\ell,k)};n-n_j^{(\ell,k)})
\end{aligned} \tag{48}$$

for $k=2,3,\ldots,\ell-1$, where the parameters $q_{j,i}^{(\ell,k)}$ and $n_j^{(\ell,k)}$ for $i=1,\ldots,k-1$ and $j=1,2,\ldots,\binom{\ell}{k}$ can be written in the vector form as follows

$$\mathbf{q}_j^{(\ell,k)} = \left[q_{j,1}^{(\ell,k)} \ \cdots \ q_{j,k-1}^{(\ell,k)}\right]^T \text{ and } \quad \mathbf{n}^{(\ell,k)} = \left[n_1^{(\ell,k)} \ \cdots \ n_{\binom{\ell}{k}}^{(\ell,k)}\right]^T \tag{49}$$

Now we can determine the necessary parameter vectors $\mathbf{q}_j^{(\ell,k)}$ and $\mathbf{n}^{(\ell,k)}$ using the ℓ-D input delay vector $\mathbf{q}^{(\ell)} = [0 \ q_1 \ \cdots \ q_{\ell-1}]^T$ and the input formation matrices given in Table 1.

$$\begin{aligned}
\mathbf{q}_j^{(\ell,k)} &= \left[\mathbf{T}_{k,j+1}^{(\ell)} - \mathbf{T}_{k,j}^{(\ell)}\right]\mathbf{q}^{(\ell)} \qquad \text{for } j=1,\ldots,\binom{\ell}{k} \\
\mathbf{n}^{(\ell,k)} &= \mathbf{T}_{k,1}^{(\ell)}\mathbf{q}^{(\ell)}
\end{aligned} \tag{50}$$

The output of the ℓ-D subsystem can be written as

$$\begin{aligned}
\mathbf{v}_e^{(\ell,\ell)}(q_1,\ldots,q_{\ell-1};n) &= \mathcal{H}^{(\ell)}\left[\mathbf{x}_e^{(\ell)}(q_1,\ldots,q_{\ell-1};n)\right] \\
&= \sum_{i=0}^{N-\bar{q}_{\ell-1}} \mathbf{U}_e^{(\ell)}(q_1,\ldots,q_{\ell-1};n-i)\mathbf{h}^{(\ell)}(q_1,\ldots,q_{\ell-1};i)
\end{aligned} \tag{51}$$

The input matrix $\mathbf{U}_e^{(\ell)}(q_1,\ldots,q_{\ell-1};n-i)$ is replaced with $\mathbf{U}_e^{(\ell)}\delta(n-\bar{q}_{\ell-1}-i)$. The matrix $\mathbf{U}_e^{(\ell)}$ has the dimension $C_{M,\ell} \times C_{M,\ell}$ and can be written in terms of the gain factors $a_1, \ a_2,\ldots,a_M$ as,

$$\mathbf{U}_e^{(\ell)} = \begin{bmatrix} \mathbf{u}^{(\ell)^T}(1,2,\ldots,\ell-1,\ell) \\ \mathbf{u}^{(\ell)^T}(1,2,\ldots,\ell-1,\ell+1) \\ \vdots \\ \mathbf{u}^{(\ell)^T}(M-\ell+1, \ M-\ell+2, \ \ldots,M) \end{bmatrix} \tag{52a}$$

where

$$\mathbf{u}^{(\ell)^T}(i_1,\ldots,i_\ell) = \left[u_\ell^{(\ell)}(i_1,\ldots,i_\ell) \vdots \mathbf{u}_{\ell+1}^{(\ell)^T}(i_1,\ldots,i_\ell) \vdots \cdots \vdots \mathbf{u}_M^{(\ell)^T}(i_1,\ldots,i_\ell)\right] \tag{52b}$$

The subvector $\mathbf{u}_k^{(\ell)}(i_1,\ldots,i_\ell)$ includes all the possible input gain factor products corresponding to the Volterra kernels of degree k with ℓ cross-terms,

$$\mathbf{u}_k^{(\ell)}(i_1,i_2,\ldots,i_\ell) = \left[a_{i_1}^{p_1} a_{i_2}^{p_2} \cdots a_{i_\ell}^{p_\ell}\right]_{\sigma(p_1,p_2,\ldots,p_\ell)} \tag{53}$$

for all $\binom{k-1}{\ell-1}$ combinations $\sigma(p_1,p_2,\ldots,p_\ell)$, where $p_1+p_2+\ldots+p_\ell = k$ and $p_i \geqslant 1$, $i=1,\ldots,\ell$ and $k=\ell,\ell+1,\ldots,M$. Thus

$$\begin{aligned}
u_\ell^{(\ell)}(i_1,i_2,\ldots,i_\ell) &= [a_{i_1}a_{i_2}\cdots a_{i_\ell}]_{1\times\binom{\ell-1}{\ell-1}} \\
\mathbf{u}_{\ell+1}^{(\ell)T}(i_1,i_2,\ldots,i_\ell) &= \left[\left(a_{i_1}\cdots a_{i_{\ell-1}}a_{i_\ell}^2\right) \quad \ldots \quad \left(a_{i_1}a_{i_2}^2a_{i_3}\cdots a_{i_\ell}\right) \quad \left(a_{i_1}^2a_{i_2}\cdots a_{i_\ell}\right)\right]_{1\times\binom{\ell}{\ell-1}} \\
\mathbf{u}_{\ell+2}^{(\ell)T}(i_1,i_2,\ldots,i_\ell) &= \left[\left(a_{i_1}\cdots a_{i_{\ell-1}}a_{i_\ell}^3\right) \quad \ldots \quad \left(a_{i_1}^2a_{i_2}^2a_{i_3}\cdots a_{i_\ell}\right) \quad \left(a_{i_1}^3a_{i_2}\cdots a_{i_\ell}\right)\right]_{1\times\binom{\ell+1}{\ell-1}} \\
&\;\;\vdots \\
\mathbf{u}_{M}^{(\ell)T}(i_1,i_2,\ldots,i_\ell) &= \left[\left(a_{i_1}\cdots a_{i_{\ell-1}}a_{i_\ell}^{M-\ell+1}\right) \cdots \left(a_{i_1}^{M-\ell}a_{i_2}^2a_{i_3}\cdots a_{i_\ell}\right) \left(a_{i_1}^{M-\ell+1}a_{i_2}a_{i_{i_3}}\cdots a_{i_\ell}\right)\right]_{1\times\binom{M-1}{\ell-1}}
\end{aligned} \tag{54}$$

The nonsingularity of the input matrix $\mathbf{U}_e^{(1)}$, i.e. $a_i \neq 0$ and $a_i \neq a_j$, $\forall\, i \neq j$, is a sufficient condition for the nonsingularity of the higher order input matrices $\mathbf{U}_e^{(\ell)}$ $1 < \ell \leqslant M$, as defined above.

The ℓ-D Volterra kernel vectors can be written for $n = \bar{q}_{\ell-1},\ \bar{q}_{\ell-1}+1,\ldots,N$ as

$$\mathbf{h}^{(\ell)}(q_1,\ldots,q_{\ell-1}; n-\bar{q}_{\ell-1}) = \left[\mathbf{U}_e^{(\ell)}\right]^{-1}\mathbf{v}_e^{(\ell,\ell)}(q_1,\ldots,q_{\ell-1};n) \tag{55a}$$

where

$$\begin{aligned}
\mathbf{v}_e^{(\ell,\ell)}(q_1,\ldots,q_{\ell-1};n) = \mathbf{y}_e^{(\ell)}(q_1,\ldots,q_{\ell-1};n) &- \sum_{j=1}^{\binom{\ell}{1}} \mathbf{S}_{\ell 1,j}^{(M)}\mathbf{v}_e^{(1,1)}(n-n_j^{(\ell,1)}) \\
&- \sum_{k=2}^{\ell-1}\sum_{j=1}^{\binom{\ell}{k}} \mathbf{S}_{\ell k,j}^{(M)}\mathbf{v}_e^{(k,k)}(q_{j,i}^{(\ell,k)},\ldots,q_{j,k-1}^{(\ell,k)}; n-n_j^{(\ell,k)})
\end{aligned} \tag{55b}$$

(55) shows that our algorithm can form the estimate for any Volterra kernel independent from other kernels. (55a) only includes the inverse of a matrix ($\mathbf{U}_e^{(\ell)}$) which can be calculated beforehand, and output values ($\mathbf{v}_e^{(\ell,\ell)}$) which should be obtained for certain multilevel input sequences. In this equation there is no reference to estimates of other kernel values. Hence, any error in previous kernel estimates cannot leak into current kernel estimate calculations. This equation proves the no-error propagation property of our algorithm.

We define the following output pick-up operators for $k = 1, 2, \ldots, \ell - 1$:

$$\mathbb{S}_{\ell,k}^{(M)}\left[\mathbf{v}_e^{(k,k)}(q_1, \ldots, q_{k-1}; n)\right] = \sum_{j=1}^{\binom{\ell}{k}} \mathbf{S}_{\ell k,j}^{(M)} \mathbf{v}_e^{(k,k)}(q_{j,1}^{(\ell,k)}, \ldots, q_{j,k-1}^{(\ell,k)}; n - n_j^{(\ell,k)}) \tag{56}$$

With this definition, (55b) can be rewritten in a more compact form.

$$\mathbf{v}_e^{(\ell,\ell)}(q_1, \ldots, q_{\ell-1}; n) = \mathbf{y}_e^{(\ell)}(q_1, \ldots, q_{\ell-1}; n) - \sum_{k=1}^{\ell-1} \mathbb{S}_{\ell,k}^{(M)}\left[\mathbf{v}_e^{(k,k)}(q_1, \ldots, q_{k-1}; n)\right] \tag{57}$$

Fig. 1 depicts the identification of the Volterra kernels of orders one through M using the proposed algorithm. Let us discuss the practical implementation of the algorithm. The system to be modelled is assumed to have a known order of nonlinearity M. The memory length N might be assumed to be known, or it might get estimated during the identification procedure on the fly. The steps of the measurement process might be listed as follows:

(i) The impulse levels are chosen, possibly using numerical optimization.

(ii) Multilevel sequences composed of a single impulse are applied to the system and the observed output vectors are stored ($\mathbf{y}_e^{(1)}(n)$ in Fig. 1).

(iii) For $i = 2, \ldots, M$

Multilevel sequences composed of i distinct impulses are applied to the system and the observed output vectors are stored ($\mathbf{y}_e^{(i)}(q_1, \ldots, q_{i-1}; n)$ in Fig. 1). Appropriate previously stored output vectors are chosen by $\mathbb{S}_{i,j}^{M}$ operators, and these are subtracted from $\mathbf{y}_e^{(i)}(q_1, \ldots, q_{i-1}; n)$. Hence, $\mathbf{v}_e^{(i,i)}(q_1, \ldots, q_{i-1}; n)$ are formed.

(iv) For $i = 1, \ldots, M$

$\mathbf{v}_e^{(i,i)}(q_1, \ldots, q_{i-1}; n)$ are multiplied by the inverse of $\mathbf{U}_e^{(i)}$, and the Volterra kernel estimates are formed.

Remark 3.2: We find the length of the deterministic input sequence that should be applied to identify the kernels $\ell = 1, \ldots, M$ of an M^{th}-order system, assuming the input ensembles are applied serially to a single system box. We put an all zero guarding interval of length N at the end of each of the input ensembles $\mathbf{x}_e^{(\ell)}(q_1, \ldots, q_{\ell-1}; n)$, to flush out that ensemble and to prepare the system for the next ensemble. The total length of the overall input sequence is calculated as

$$L = \sum_{\ell=1}^{M} (\bar{q}_{\ell-1} + N + 1) \binom{M}{\ell}\binom{N}{\ell-1} \leqslant (2N+1) \sum_{\ell=1}^{M} \binom{M}{\ell}\binom{N}{\ell-1} = (2N+1) \binom{N+M}{M-1} \qquad (58)$$

The length of the deterministic input sequence is upper-bounded by $(2N+1)\binom{N+M}{M-1}$.

Example 3.3: We consider the identification of a Volterra filter with $M = 3$ and $N = 1$ as discussed in Examples 3.1 and 3.2. The overall deterministic input sequence which should be applied to identify the kernels of this system is shown in Fig. 2. This figure depicts all the input ensembles utilized in the Examples 3.1 and 3.2 for the identification of the nonlinear system. The overall input sequence is assumed to be applied serially to a single system, and there is a guarding interval of length $N = 1$ between all individual input ensembles. The distinct input amplitude levels are chosen as $a_1 = 1$, $a_2 = -1$ and $a_3 = 2$ as in the examples. The length of the overall sequence is seen from the figure as $L = 14$. This value is indeed less than the upper-bound $(2N+1)\binom{N+M}{M-1} = (3)\binom{4}{2} = 18$ from (58).

IV. PERSISTENCE OF EXCITATION AND NUMERICAL ISSUES FOR THE MULTILEVEL INPUT SIGNALS

In this section, we prove that the multilevel input signals given in Section III persistently excite a Volterra filter. We also show that the identification algorithm of Section III gives the optimal least squares estimate of the Volterra kernels. Ideally, the input signal excites all modes of the filter equally. However, it is known that polynomial systems suffer from severe ill-conditioning if the order is high. Here, the condition numbers of the input matrices are quantified to assess the uniformity of excitation to the system.

A. Persistence of Excitation for the Multilevel Deterministic Inputs

The definition of the persistence of excitation condition for Volterra filters with deterministic inputs is as follows [18]: Let $\tau > 0$ be a fixed observation period and $\lambda_{\max}$ and $\lambda_{\min}$ denote the largest and smallest eigenvalues of the sample correlation matrix. If there exists $\rho_1,\rho_2 > 0$ such that for every time instant k, $\rho_1 \leqslant \lambda_{\min} \leqslant \lambda_{\max} \leqslant \rho_2$, then the input signal is said to be persistently exciting (PE).

The ℓ-D ensemble input signals do not excite the Volterra kernels having more than ℓ

cross-terms, i.e., $\mathbf{v}_e^{(\ell,k)} = \mathcal{H}^{(k)}\left[\mathbf{x}_e^{(\ell)}\right] = \mathbf{0}$, $\forall\, k > \ell$, and $\mathcal{N}\left[\mathbf{x}_e^{(\ell)}\right] = \sum_{k=1}^{\ell} \mathbf{v}_e^{(\ell,k)}$. By subtracting the appropriate previously stored lower dimensional outputs, we can obtain the response of the ℓ-D subsystem alone. With this subtraction we single out the input-output ensembles for the ℓ-D cross-term subsystem $\mathcal{H}^{(\ell)}$. If we can prove that the input ensembles are PE for $\mathcal{H}^{(\ell)}$, then the input ensembles will also be PE for the composite nonlinear structure $\mathcal{N}$. The ℓ-D multivariate cross-term subsystem can be further decomposed according to the delay variables. (2c) can be rewritten as

$$y^{(\ell)}(n) = \mathcal{H}^{(\ell)}\left[x(n)\right] = \sum_{p=1}^{\binom{N}{\ell-1}} \sum_{i=0}^{N-\mathrm{sum}(\boldsymbol{q}_p)} \mathbf{h}^{(\ell)^T}(\boldsymbol{q}_p; i)\, \mathbf{x}^{(\ell)}(\boldsymbol{q}_p; n-i),$$

where the sum over p is a sum over all possible delay structures, and $\boldsymbol{q}_p = [q_1, \ldots, q_{\ell-1}]_p$ is an $(\ell-1)$-D delay vector with $\mathtt{sum}(\boldsymbol{q}_p) \leqslant N$. We can reformulate the output at time n as $y^{(\ell)}(n) = \sum_{p=1}^{\binom{N}{\ell-1}} \mathbf{h}^{(\ell)^T}(\boldsymbol{q}_p)\mathbf{x}_n^{(\ell)}(\boldsymbol{q}_p)$, where $\mathbf{h}^{(\ell)}(\boldsymbol{q}_p)$ and $\mathbf{x}_n^{(\ell)}(\boldsymbol{q}_p)$ are column vectors. $\mathbf{h}^{(\ell)}(\boldsymbol{q}_p)$ is a concatenation of the kernel vectors $\mathbf{h}^{(\ell)}(\boldsymbol{q}_p; i)$, whereas $\mathbf{x}_n^{(\ell)}(\boldsymbol{q}_p)$ is a concatenation of the expanded input vectors $\mathbf{x}^{(\ell)}(\boldsymbol{q}_p; n-i)$. We can rewrite this linear combination as a single vector product

$$y^{(\ell)}(n) = \mathbf{h}^{(\ell)^T}\mathbf{x}^{(\ell)}(n) \tag{59}$$

by rearranging the vectors $\mathbf{h}^{(\ell)}(\boldsymbol{q}_p)$ and $\mathbf{x}_n^{(\ell)}(\boldsymbol{q}_p)$. $\mathbf{h}^{(\ell)}$ and $\mathbf{x}^{(\ell)}(n)$ are column vectors generated by concatenating respectively the vectors $\mathbf{h}^{(\ell)}(\boldsymbol{q}_p)$ and $\mathbf{x}_n^{(\ell)}(\boldsymbol{q}_p)$ for all $\binom{N}{\ell-1}$ possible delay structures $\boldsymbol{q}_p$ together. Let us assume that we apply our total multilevel ensemble input signal periodically. Suppose we begin observing the output of the ℓ-D system at some time n and collect data over an observation period $\tau > 0$. The output vector $\mathbf{y}^{(\ell)}(n) = \left[y^{(\ell)}(n)\ \ldots\ y^{(\ell)}(n+\tau)\right]^T$ is related to the input by

$$\mathbf{y}^{(\ell)}(n) = \mathbf{X}^{(\ell)}(n)\mathbf{h}^{(\ell)} \tag{60}$$

where the data matrix $\mathbf{X}^{(\ell)}(n) = \left[\mathbf{x}^{(\ell)}(n)\ \cdots\ \mathbf{x}^{(\ell)}(n+\tau)\right]^T$. We want to form the sample correlation matrix

$$\boldsymbol{\mathcal{S}}^{(\ell)} = \frac{1}{\tau}\mathbf{X}^{(\ell)^T}(n)\,\mathbf{X}^{(\ell)}(n) = \frac{1}{\tau}\sum_{m=n}^{n+\tau} \mathbf{x}^{(\ell)}(m)\,\mathbf{x}^{(\ell)^T}(m) \tag{61}$$

We take τ as the overall period of the input ensembles together, i.e., as the length of the overall input sequence which we assumed is applied periodically. Then, at each time m for which the current ensemble input has a delay structure other than $\boldsymbol{q}_p$, $\mathbf{x}_m^{(\ell)}(\boldsymbol{q}_p) = \mathbf{0}$. If the current input has a delay structure given by $\boldsymbol{q}_p$, then all the subvectors of $\mathbf{x}_m^{(\ell)}(\boldsymbol{q}_p)$, i.e., $\mathbf{x}^{(\ell)}(\boldsymbol{q}_p; m-i)$, $i = 0 \ldots N - \mathtt{sum}(\boldsymbol{q}_p)$ will be zero vectors except only one subvector for a certain shift $\bar{i}$, that is, $\mathbf{x}_m^{(\ell)}(\boldsymbol{q}_p) = \begin{bmatrix}\mathbf{0}^T & \cdots & \mathbf{0}^T & \mathbf{x}^{(\ell)}(\boldsymbol{q}_p; m-\bar{i})^T & \mathbf{0}^T & \cdots & \mathbf{0}^T\end{bmatrix}^T$. Hence, the sample correlation matrix $\boldsymbol{\mathcal{S}}^{(\ell)}$ will become a block diagonal matrix. For each ensemble input sequence with a delay structure given by $\boldsymbol{q}_p$, $\mathbf{x}_m^{(\ell)}(\boldsymbol{q}_p)$ will have the nonzero subvector at the position $i = 0$ at some time m, and at every next point in time the nonzero subvector will travel down until it reaches $i = N - \mathtt{sum}(\boldsymbol{q}_p)$. The total ensemble input sequence includes all possible ℓ-level combinations for all possible delay structures. Hence, when we form the block diagonal sample correlation matrix, the matrices on the diagonal will be equal, i.e., for the multilevel deterministic ensemble input sequence

$$\boldsymbol{\mathcal{S}}^{(\ell)} = \frac{1}{\tau}\begin{bmatrix} \left[\boldsymbol{S}_{(\ell)}\right] & [\mathbf{0}] & \cdots & [\mathbf{0}] \\ [\mathbf{0}] & \left[\boldsymbol{S}_{(\ell)}\right] & \cdots & \vdots \\ \vdots & \vdots & \ddots & [\mathbf{0}] \\ [\mathbf{0}] & \cdots & [\mathbf{0}] & \left[\boldsymbol{S}_{(\ell)}\right] \end{bmatrix}. \tag{62}$$

where $\boldsymbol{S}_{(\ell)}$ is $\binom{M}{\ell} \times \binom{M}{\ell}$. It can be shown that $\boldsymbol{S}_{(\ell)} = \sum_{j=1}^{\binom{M}{\ell}} \mathbf{u}^{(\ell)}(\boldsymbol{i}_j)\mathbf{u}^{(\ell)^T}(\boldsymbol{i}_j)$, where the vector $\mathbf{u}^{(\ell)}(\boldsymbol{i}_j)$ is as defined in (52b). $\boldsymbol{i}_j = [i_1 \ i_2 \ \cdots \ i_\ell]_j$ denotes which impulse amplitudes to pick and the index $j = 1 \ldots \binom{M}{\ell}$ is such that all possible impulse amplitude ensembles are covered. $\boldsymbol{S}_{(\ell)}$ can be rewritten as

$$\boldsymbol{S}_{(\ell)} = \mathbf{U}_e^{(\ell)^T}\mathbf{U}_e^{(\ell)} \tag{63}$$

where $\mathbf{U}_e^{(\ell)}$ is the ℓ-D ensemble input matrix as given in (52a). Hence, for our deterministic input ensemble the eigenvalues of the sample correlation matrix $\boldsymbol{\mathcal{S}}^{(\ell)} = \frac{1}{\tau}\mathbf{X}^{(\ell)^T}(n)\mathbf{X}^{(\ell)}(n)$ are equal to the eigenvalues of the matrix $\boldsymbol{S}_{(\ell)} = \mathbf{U}_e^{(\ell)^T}\mathbf{U}_e^{(\ell)}$. Positive definiteness of $\boldsymbol{S}_{(\ell)}$ is necessary and sufficient for the persistence of excitation condition, in which case $\rho_1 = \lambda_{\boldsymbol{S},\min}$ and $\rho_2 = \lambda_{\boldsymbol{S},\max}$ where $\lambda_{\boldsymbol{S},\min}$ and $\lambda_{\boldsymbol{S},\max}$ are the smallest and largest eigenvalues of $\boldsymbol{S}_{(\ell)}$, respectively. It is a known fact that $\boldsymbol{S}_{(\ell)} = \mathbf{U}_e^{(\ell)^T}\mathbf{U}_e^{(\ell)}$ will be a symmetric and positive definite matrix if and only if the square matrix $\mathbf{U}_e^{(\ell)}$ is nonsingular. Therefore, the multilevel input sequence is PE for the subsystem $\mathcal{H}^{(\ell)}$ if and only if the input ensemble matrix $\mathbf{U}_e^{(\ell)}$ is nonsingular. Hence, the multilevel input sequence will be PE for the overall nonlinear system $\mathcal{N}$ if and only if all the input ensemble matrices $\mathbf{U}_e^{(\ell)}$, $1 \leq \ell \leq M$ are nonsingular, for which the nonsingularity of $\mathbf{U}_e^{(1)}$ is a sufficient condition. Therefore, the input ensemble is assured to be PE when we choose distinct and nonzero amplitude levels.

Remark 4.1: A PRMS sequence is PE of order M if and only if the number of distinct levels of the sequence, i.e., the order of the sequence $q \geqslant M+1$ [18]. Our deterministic ℓ-D sequence is PE of degree M, if and only if the total number of distinct levels the sequence can assume is $M+1$ (counting '0' as a level also).

Remark 4.2: Let $\texttt{blkdiag}\left(\mathbf{A}, p\right)$ denote the block diagonal matrix which results from $p-$times block diagonal concatenation of the matrix $\mathbf{A}$. For example, (62) can be rewritten as $\boldsymbol{\mathcal{S}}^{(\ell)} = \texttt{blkdiag}\left(\boldsymbol{S}_{(\ell)}, \binom{N+1}{\ell}\right)$. Using this definition and (63)

$$\boldsymbol{\mathcal{S}}^{(\ell)} = \texttt{blkdiag}\left(\mathbf{U}_e^{(\ell)^T}, \binom{N+1}{\ell}\right) \texttt{blkdiag}\left(\mathbf{U}_e^{(\ell)}, \binom{N+1}{\ell}\right) \tag{64}$$

From (60)-(64), the least squares estimate of the kernel vector $\mathbf{h}^{(\ell)}$ is given by

$$\begin{aligned} \hat{\mathbf{h}}^{(\ell)} &= \left(\mathbf{X}^{(\ell)^T}(n)\mathbf{X}^{(\ell)^T}(n)\right)^{-1}\mathbf{X}^{(\ell)^T}(n)\mathbf{y}^{(\ell)}(n) \\ &= \texttt{blkdiag}\left(\left[\mathbf{U}_e^{(\ell)}\right]^{-1}, \binom{N+1}{\ell}\right) \texttt{blkdiag}\left(\left[\mathbf{U}_e^{(\ell)^T}\right]^{-1}, \binom{N+1}{\ell}\right) \mathbf{X}^{(\ell)^T}(n)\mathbf{y}^{(\ell)}(n) \end{aligned} \tag{65}$$

It is not difficult to show that $\texttt{blkdiag}\left(\left[\mathbf{U}_e^{(\ell)^T}\right]^{-1}, \binom{N+1}{\ell}\right)\mathbf{X}^{(\ell)^T}(n)$ in (65) is simply a permutation matrix, which changes the order of the output terms in $\mathbf{y}^{(\ell)}(n)$ and regroups them. This reordered output vector $\mathbf{v}^{(\ell)}(n) = \texttt{blkdiag}\left(\left[\mathbf{U}_e^{(\ell)^T}\right]^{-1}, \binom{N+1}{\ell}\right)\mathbf{X}^{(\ell)^T}(n)\mathbf{y}^{(\ell)}(n)$ is a concatenation of the output ensemble vectors $\mathbf{v}_e^{(\ell,\ell)}(q_1, \ldots, q_{\ell-1}; n)$ which were defined in (55b). The least squares solution in (65) becomes

$$\hat{\mathbf{h}}^{(\ell)} = \texttt{blkdiag}\left(\left[\mathbf{U}_e^{(\ell)}\right]^{-1}, \binom{N+1}{\ell}\right)\mathbf{v}^{(\ell)}(n)$$

This result is equivalent to the ℓ-D kernel vector calculation equation in (55a). Hence, equation (55a) is indeed the least squares solution for $\mathbf{h}^{(\ell)}(q_1, \ldots, q_{\ell-1}; n)$.

B. Uniform Excitation of Volterra System

Practically the estimation error is determined by the noise power at the output of the system and the size of the eigenvalues of the sample correlation matrix. However, here we examine rather the ability of the deterministic sequences to uniformly excite all the modes of the Volterra filter. The condition to uniformly excite all modes of the ℓ-D substructure $\mathcal{H}^{(\ell)}$ requires to find an input level ensemble such that the condition number of the matrix $\mathbf{U}_e^{(\ell)}$ is minimized. Since we want uniformity of excitation for all the substructures of the orders $\ell = 1, \ldots, M$, the condition numbers for all the input matrices $\mathbf{U}_e^{(\ell)}$ should be minimized simultaneously. A collective criterion should be chosen to define what the optimally conditioned input matrices would be. The criterion we used to optimize the input level values was to minimize the maximum condition number among all the condition numbers. Other criteria, for example the mean-square of the condition numbers, could as well be used. The maximum condition numbers and the optimal level values are given in Table 3 for nonlinearity orders $M = 1, \ldots, 6$. Here the optimization was unconstrained; however, for the simulations in next section the input levels should also fulfill certain power constrains. Hence, in the case of the simulations the optimal levels which are subject to certain average power constrains are found by constrained optimization.

All the results in this paper are based on the real input case. However, it is possible to use impulses with complex values as the multilevel input sequences of our algorithm. Changes in the values of the impulses included in the multilevel sequence will simply result in different $\mathbf{U}_\mathbf{e}^{(\ell)}$ matrices. The data correlation matrices will still be of the same sparse block diagonal form. Using complex values for the impulses might result in input ensemble matrices with better condition numbers. Hence, we can have better error performance for the algorithm without compromising any power constraints.

V. SIMULATIONS

Two numerical simulations to demonstrate the performance of the novel identification procedure are presented. The simulation setups are taken from [15] and [18]. To compare the performance with the existing algorithms, we have to use input lengths equal to the setups given in these. However, our required input ensemble length is much shorter than the required lengths for the PRMS and PSK inputs. In order to retain comparable input lengths, we resend our input ensembles until the total input length is close to those given in [15] and [18] and we take the mean of the estimates coming from individual input cycles. The input impulse levels are chosen optimally as to satisfy the average power constrains of their respective examples, and to minimize the condition number criterion as defined in Section IV. The optimal level values are given in Table 4.

Example 5.1: We simulate a second order Volterra filter with memory length $N = 2$. The average input power is unity for both PRMS and our multilevel sequence. Gaussian white noise (GWN) of power 0.1 is added to the system output to represent observation noise. This setup is equivalent to the simulation in [18] with the difference that in the example there an extended Volterra filter had to be used, which is a drawback of using the Kronecker product representation. In Table 5 the averaged squared error between the estimated and true kernels and the number of floating point operations required are given for four different input sequence lengths. For PRMS the results are taken from Table IV in [18], and the average is over 200 independent trials. For our algorithm the average is also taken over 200 independent trials.

From the results in Table 5, it is clear that our algorithm uses less operations and gives better results than the PRMS method [18]. To get comparable data lengths, we simply resent our input of length 15 and averaged over the estimates. This is reflected in the decrease of the mean square error for our algorithm. The length of the PRMS sequences are calculated as q^{N+1}, where $q > M$ is the order of the PRMS sequence. As we go down the Table 5, the order of the PRMS sequence used is increased as $q = 3, 4, 5, 7$, and the improvement in the condition number of the PRMS sample correlation matrix gets reflected in the error performance of the PRMS. Only after $q = 7$, PRMS has better error performance than our algorithm. However, it should not be overlooked that this example has a limited memory length $N = 2$, and for larger N using higher order PRMS would make the input length q^{N+1} prohibitively large.

The reason why our algorithm performs better than the PRMS method in Table 5 is that our algorithm has an input diversity better than the PRMS input. The PRMS has to spend some of the allowed input power on input sequences, which are redundant for the identification of the regular Volterra filter, whereas our algorithm only includes the necessary input signals.

There are examples for the use of the PRMS in Volterra system identification also in [23]. However, these examples illustrate the errors associated with the use of finite data in the absence of observation noise. In the case of PRMS, to get an exact least squares solution in the absence of observation noise, a full period of the PRMS, i.e., a sequence of length $(M+1)^{N+1}$ has to be used. For example, for a third-order filter ($M=3$) with memory length 11 ($N=11$), to get the exact solution with PRMS a data record of length 4^{12} is required. Use of shorter data records removes the PE property, and estimate errors occur as studied in [23]. However, our algorithm is much more effective in the input diversity than PRMS, because our multilevel deterministic input completely eliminates those input combinations which are required for the identification of the extended Volterra filter but not for the regular Volterra system, which are included in the PRMS. For example, using (58) we can show that for the Volterra filter with ($M=3$) and ($N=11$) to get the exact kernel estimates our algorithm needs a data record of length less than $(2N+1)\binom{N+M}{M-1} = 2093$. This is a radical improvement over the required length 4^{12} for PRMS.

Example 5.2: We simulate a fourth-order Volterra filter with memory length $N=3$ after the example 2 in [15]. Average input power is set to 4, and additive GWN observation noise with variance 0.5 is present. The data length for the PSK input is 4096. Our multilevel sequence is of length 211 and we send it through 19 times; thus our total input length is 4009. The optimal level values used in our multilevel sequence are given in Table 4. Table 6 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 [15]. There are five nonzero Volterra kernels, and the values are taken from [15]. The results for PSK and the results for our algorithm are averaged over 200 independent trials. The results for our algorithm are better than those for PSK inputs and our estimates are very accurate despite the high order of nonlinearity and the presence of noise.

The PSK input method uses higher-order moments. Example 1 in [15] presents the results for the identification a third order Volterra filter with $N=4$, where the mean and deviations of the estimates are tabulated in the absence of observation noise for an input length of 4096. Our algorithm is an exact algorithm. Hence, in the absence of observation noise, our input sequence identifies the *exact* kernel values of this filter for an input length as short as 155.

VI. CONCLUDING REMARKS

We have developed a novel algorithm for input-output nonlinear system identification. The optimal estimation of Volterra kernels requires the solution of a large system of simultaneous linear equations. Our algorithm avoids the direct computation of the least squares estimate. By using a certain deterministic multilevel sequence and a novel partitioning of the Volterra kernels, the data correlation matrix reduces to a block diagonal matrix with relatively smaller dimensional matrices on the diagonal. These matrices depend only on the order of nonlinearity M and the input levels, hence they and their inverses can be calculated and stored beforehand. Our algorithm is in closed-form, exact, non-iterative and a computationally efficient implementation is also presented. It is shown that the input sequence is PE and the problem of ill-conditioning is also discussed. It is demonstrated with simulations that the algorithm can produce better parameter estimates than some existing algorithms [15], [18].

There are some other merits of this input sequence, which it shares with the PRMS. In some identification problems the normal operating input of the nonlinear system may be restricted to a fixed amplitude range. In this case the deterministic input sequence can utilize the full dynamic amplitude range of the normal operating interval of the system input. Another benefit of our algorithm is that it gives the Volterra kernel estimates directly. Some methods in the literature first calculate coefficients which have no directly interpretable results (i.e. the Wiener coefficients in [24]), and then obtain the Volterra kernels from these. An additional advantage is that the deterministic input sequence is suited to low rank approximations. The eigenvectors and the eigenvalues of the sample correlation matrix of the deterministic input sequence can be computed a priori as in the case of the PRMS. The algorithm presented in this paper will facilitate better Volterra kernel estimates for short input sequences. This will help in devising better Volterra-based compensators, because the quality of such compensators depend on the accuracy of the Volterra kernel estimates for the nonlinear system. One disadvantage our algorithm might have is that the excitation sequence cannot be controlled according to the experimental setting. For example, the PSK input method [15] is natural to get used in a channel identification setting where the communication signal is PSK modulated. The probing signal simply becomes the communication signal. However, our algorithm requires the use of synthetic impulse sequences, which might result in additional overhead.

Finally, we comment on the possible future research directions using the results in this paper. One application might be the use of the novel kernel partitioning in the efficient implementation of the Volterra filters and in transform domain structures. Another direction might be the use of the identification algorithm in the implementation of nonlinear compensators and nonlinear system inverses and equalization.

References

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

[2] M. Schetzen, *The Volterra and Wiener Theories of Nonlinear Systems*, New York: Wiley, 1980. Reprint edition with additional published by Robert E. Krieger Co., Malabar, Fla., 1989.

[3] S. Benedetto and E. Biglieri, "Nonlinear equalization of digital satellite channels", *IEEE J. Sel. Areas in Comm.*, vol. SAC1, no. 1, pp. 57-62, Jan. 1983.

[4] H. T. Ching and E. J. Powers, "Nonlinear channel equalization in digital satellite system", *Proceedings of GLOBECOM'93*, vol. 3, Houston, TX, 1993, pp. 1639-1643.

[5] 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.

[6] M. T. Özden, E. Panayırcı and A. H. Kayran, "Identification of nonlinear magnetic channels with lattice orthogonalisation", *Electronics Letters*, vol.33, no.5, pp.376–377, Feb. 1997.

[7] V. J. Mathews, "Adaptive polynomial filters", *IEEE Signal Processing Mag.*, vol. 8, pp. 10-26, July 1991.

[8] M. Schetzen, "A theory of non-linear system identification", *Int. J. Control*, vol. 20, no. 4, pp. 577-592, 1974.

[9] 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.

[10] Y. S. Cho and E. J. Powers, "Quadratic system identification using higher order spectra of i.i.d. signals", *IEEE Trans. Signal Processing*, vol. 42, no. 5, pp. 1268-1271, May 1994.

[11] S. W. Nam and E. J. Powers, "Application of higher order spectral analysis to cubically nonlinear system identification", *IEEE Trans. Signal Processing*, vol. 42, no. 7, pp. 1746-1765, July 1994.

[12] G. A. Glentis, P. Koukoulas and N. Kalouptsidis "Efficient algorithms for Volterra system identification", *IEEE Trans. Signal Processing*, vol. 47, no. 11, pp. 3042–3057, Nov. 1999.

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

[14] P. Koukoulas and N. Kalouptsidis, "Second-order Volterra system identification", *IEEE Trans. Signal Processing*, vol. 48, no. 12, pp.3574–3577, Dec. 2000.

[15] 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.

[16] S. Boyd, Y. S. Tang and L. O. Chua, "Measuring Volterra Kernels (I)", *IEEE Trans. on Circuits and Systems*, vol. CAS-30, no. 8, pp. 571–577, Aug. 1983.

[17] L. O. Chua and Y. Liao, "Measuring Volterra Kernels (II)", *International Journal of Circuit Theory and Applications*, vol. 17, pp. 151–190, 1983.

[18] R. D. Nowak and B. D. Van Veen, "Random and pseudorandom inputs for Volterra filter identification", *IEEE Trans. Signal Processing*, vol. 42, no. 8, pp. 2124–2135, Aug. 1994.

[19] M. Schetzen, "Measurement of the kernels of a non-linear system of finite order", *Int. J. Contr.*, vol. 1, no. 3, pp. 251–263, Mar. 1965.

[20] G. Ramponi, "Bi–impulse response design of isotropic quadratic filters", *Proc. IEEE*, vol. 78, no. 4, pp. 665–677, Apr. 1990.

[21] S. R. Parker and F. A. Perry, "A discrete ARMA model for nonlinear system identification", *IEEE Trans. Circuits and Systems*, vol. CAS28, pp. 224-233., Mar. 1981.

[22] G. M. Raz and B. D. Van Veen, "Baseband Volterra filters for implementing carrier based nonlinearities", *IEEE Trans. Signal Processing*, vol. 46, no. 1, pp. 103-114, Jan. 1998.

[23] R. D. Nowak and B. Van Veen, "Efficient methods for identification of Volterra filter models", *Signal Processing*, vol. 38, no. 3, pp. 417-428, Aug. 1994.

[24] M. J. Reed and M. O. J. Hawksford, "Identification of discrete Volterra series using maximum length sequences", *Proc. IEE, Circuits, Devices and Systems*, vol. 143, no. 5, pp. 241-248, Oct. 1996.

Table 1

Ensemble Input Formation Matrices

Define $\mathbf{T}_{1,1}^{(p)} = \mathbf{I}_p \,, \quad \mathbf{i}_k = [1 \; \cdots \; 1]_{1\times k}^T \,,$ $C_{p,\ell} = \binom{p}{\ell}, \quad \mathbf{\Pi}_{k,i} = [0 \; \cdots \; 0 \; 1 \; 0 \; \cdots \; 0]_{1\times k}$ where only ith-entry is unity.
for $\ell = 2, 3, \ldots, M$ $\mathbf{T}_{\ell,i}^{(\ell)} = \mathbf{\Pi}_{C_{\ell,1},i} \,; \quad i = 1, 2, \ldots, \ell$
for $p = \ell, \ell+1, \ldots, M-1$ $$\mathbf{T}_{\ell,1}^{(p+1)} = \begin{bmatrix} \mathbf{i}_{C_{p,\ell-1}} & \mathbf{0}_{C_{p,\ell-1}\times p} \\ \mathbf{0}_{C_{p,\ell}\times 1} & \mathbf{T}_{\ell,1}^{(p)} \end{bmatrix}_{C_{p+1,\ell}\times(p+1)}$$
for $i = 2, 3, \ldots, \ell$ $$\mathbf{T}_{\ell,i}^{(p+1)} = \begin{bmatrix} \mathbf{0}_{C_{p,\ell-1}\times 1} & \mathbf{T}_{\ell-1,i-1}^{(p)} \\ \mathbf{0}_{C_{p,\ell}\times 1} & \mathbf{T}_{\ell,i}^{(p)} \end{bmatrix}_{C_{p+1,\ell}\times(p+1)}$$

Table 2

Ensemble Output Pick Up Matrices

Initial Matrices: for $p = 2, 3, \ldots, M$, $\ell = 1, 2, \ldots, p$ and $k = 1, 2, \ldots, \ell$ $\mathbf{S}^{(p)}_{\ell 1,i} = \mathbf{T}^{(p)}_{\ell,i}$, $i = 1, 2, \ldots, C_{\ell,1}$ $\mathbf{S}^{(p)}_{pk,j} = \mathbf{\Pi}_{C_{p,k},j}$, $j = 1, 2, \ldots, C_{p,k}$ $\mathbf{S}^{(p)}_{\ell\ell,1} = \mathbf{I}_{C_{p,\ell}}$
for $p = 4, 5, \ldots, M$, $\ell = 3, 4, \ldots, p-1$ and $k = 2, 3, \ldots, \ell - 1$
for $i = 1, 2, \ldots, C_{\ell-1,k-1}$ $\mathbf{S}^{(p)}_{\ell k,i} = \begin{bmatrix} \mathbf{S}^{(p-1)}_{(\ell-1)(k-1),i} & \mathbf{0}_{C_{p-1,\ell-1}\times C_{p-1,k}} \\ \mathbf{0}_{C_{p-1,\ell}\times C_{p-1,k-1}} & \mathbf{S}^{(p-1)}_{\ell k,i} \end{bmatrix}_{C_{p,\ell}\times C_{p,k}}$
for $j = 1, 2, \ldots, C_{\ell-1,k}$ $\mathbf{S}^{(p)}_{\ell k,i} = \begin{bmatrix} \mathbf{0}_{C_{p-1,\ell-1}\times C_{p-1,k}} & \mathbf{S}^{(p-1)}_{(\ell-1)k,j} \\ \mathbf{0}_{C_{p-1,\ell}\times C_{p-1,k-1}} & \mathbf{S}^{(p-1)}_{\ell k,C_{\ell-1,k-1}+j} \end{bmatrix}_{C_{p,\ell}\times C_{p,k}}$

Table 3

Optimal Deterministic Input Levels

M	$\texttt{max}\left(\mathrm{cond}_2\ \mathbf{U}_e^{(\ell)}\right)$	optimal level values
1	1.00	1.00
2	1.00	-1.00, 1.00
3	7.54	0.59, -0.92, 1.11
4	14.15	-0.90, 0.51, 1.00, -0.7291
5	40.31	-0.92, 0.86, -0.38, 1.05, -1.12
6	48.74	-1.13, 1.03, -0.84, 0.76, -0.50, 0.93,

Table 4

OPTIMAL INPUT LEVELS FOR SIMULATIONS

Example	power constraint	optimal level values	$\texttt{max}\left(\mathrm{cond}_2\ \mathbf{U}_e^{(\ell)}\right)$
I, $M = 2, N = 2$	$P = 1.00$	-1.58, 1.58	1.58
II, $M = 4, N = 3$	$P = 4.00$	-3.84, -2.53 2.53, 3.84	161.05

Table 5

AVERAGED SQUARED ERROR OF ESTIMATES FOR EXAMPLE I

PRMS			multilevel deterministic input		
length	error	flops	length	error	flops
27	7.80×10^{-1}	1.22×10^{3}	15	2.26×10^{-1}	0.12×10^{3}
64	9.93×10^{-2}	1.64×10^{3}	60	5.42×10^{-2}	0.29×10^{3}
125	2.89×10^{-2}	2.24×10^{3}	120	2.84×10^{-2}	0.53×10^{3}
343	6.18×10^{-3}	4.12×10^{3}	330	9.70×10^{-3}	1.34×10^{3}

Table 6

Results for Example II

(i_1, i_2, i_3, i_4)	$(0,0,0,0)$	$(0,0,0,1)$	$(0,0,1,1)$	$(0,0,1,2)$	$(0,1,2,3)$
true $b_4\,(i_1, i_2, i_3, i_4)$	1.0000	1.2000	0.8000	-0.5000	0.6000
mean of $\hat{b}_4\,(i_1, i_2, i_3, i_4)$ for PSK	1.0066	1.1969	0.7982	-0.5021	0.5997
std of $\hat{b}_4\,(i_1, i_2, i_3, i_4)$ for PSK	0.1890	0.0454	0.0263	0.0147	0.0039
mean of $\hat{b}_4\,(i_1, i_2, i_3, i_4)$ for new alg.	0.9998	1.1997	0.8001	-0.4997	0.6000
std of $\hat{b}_4\,(i_1, i_2, i_3, i_4)$ for new alg.	0.0023	0.0042	0.0019	0.0038	0.0069

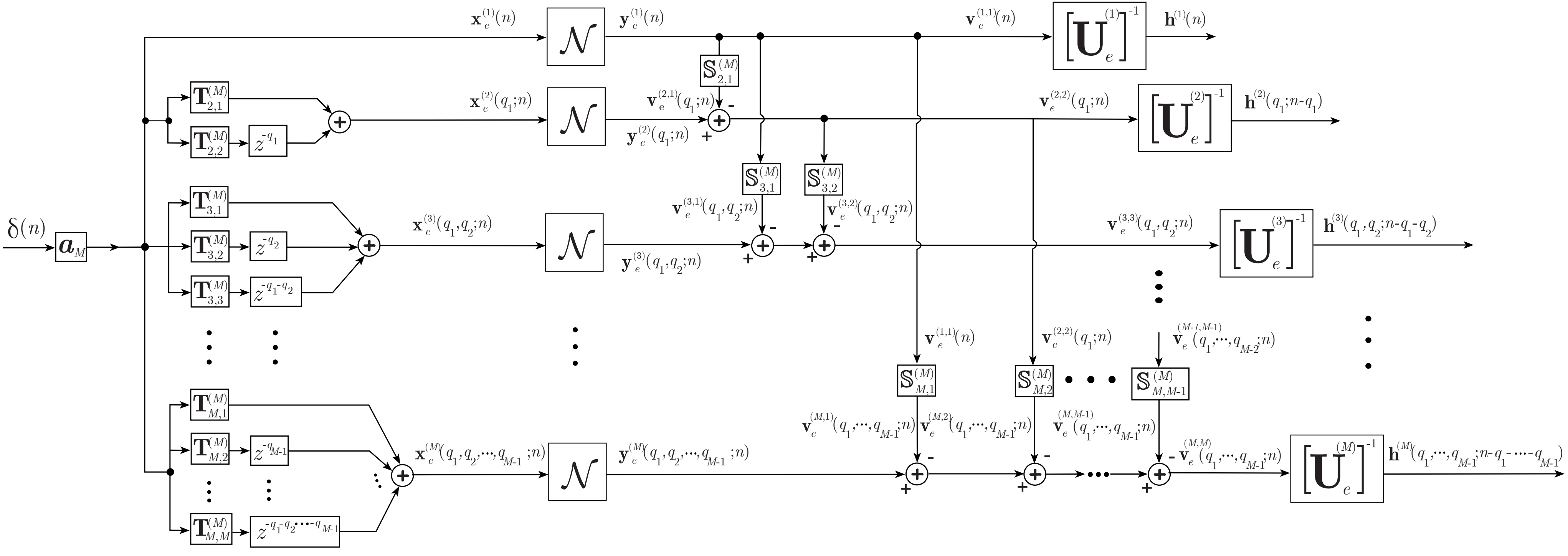

Fig. 1. Proposed Volterra kernel identification method using multilevel deterministic sequences as inputs.

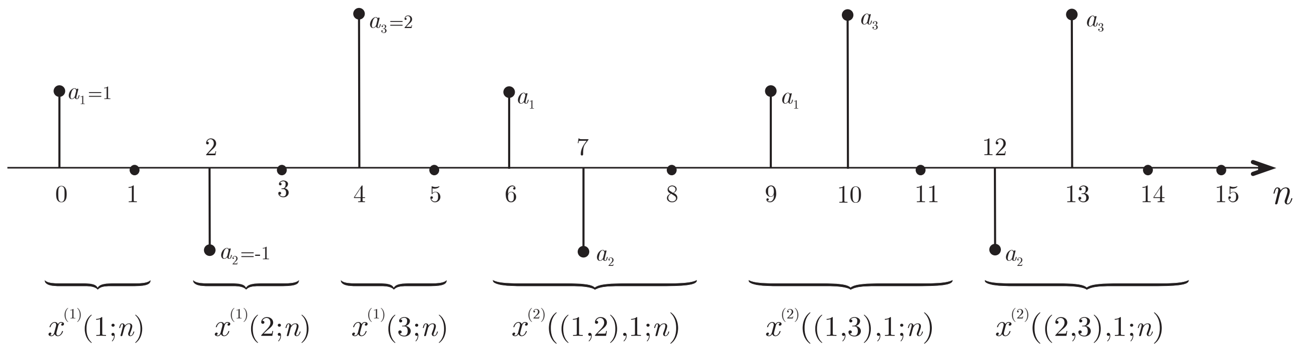

Fig. 2. Deterministic multilevel input sequence for Example 3.3.