Novel Methods For Volterra Filter Representation, Identification and Realization

Ender M. EKŞİOĞLU, M.Sc.

Advisor: Prof. Ahmet H. KAYRAN

Istanbul Technical University ⋆ Institute Of Science And Technology

Electronics and Communications Engineering Department

February 2005

Main Headings

Main Headings

- Introduction

Main Headings

- Introduction
- Volterra Filters

Main Headings

- Introduction
- Volterra Filters
- A Novel Volterra filter Representation

Main Headings

- Introduction
- Volterra Filters
- A Novel Volterra filter Representation
- Identification Of The Volterra Kernel Vectors Using Deterministic Input Signals

Main Headings

- Introduction
- Volterra Filters
- A Novel Volterra filter Representation
- Identification Of The Volterra Kernel Vectors Using Deterministic Input Signals
- Simulations for the System Identification Setting

Main Headings

- Introduction
- Volterra Filters
- A Novel Volterra filter Representation
- Identification Of The Volterra Kernel Vectors Using Deterministic Input Signals
- Simulations for the System Identification Setting
- Some Numerical Properties of the Identification Algorithm

Main Headings

- Introduction
- Volterra Filters
- A Novel Volterra filter Representation
- Identification Of The Volterra Kernel Vectors Using Deterministic Input Signals
- Simulations for the System Identification Setting
- Some Numerical Properties of the Identification Algorithm
- Application to the Identification of Nonlinear Communication channels

Main Headings

- Introduction
- Volterra Filters
- A Novel Volterra filter Representation
- Identification Of The Volterra Kernel Vectors Using Deterministic Input Signals
- Simulations for the System Identification Setting
- Some Numerical Properties of the Identification Algorithm
- Application to the Identification of Nonlinear Communication channels
- A Novel Volterra Filter Realization Method and its Application to Nonlinear Adaptive Filtering

Main Headings

- Introduction
- Volterra Filters
- A Novel Volterra filter Representation
- Identification Of The Volterra Kernel Vectors Using Deterministic Input Signals
- Simulations for the System Identification Setting
- Some Numerical Properties of the Identification Algorithm
- Application to the Identification of Nonlinear Communication channels
- A Novel Volterra Filter Realization Method and its Application to Nonlinear Adaptive Filtering
- Concluding Remarks

Introduction

In this dissertation we deal with

- discrete-time,
- finite-order,
- time-invariant

Volterra filters.

Introduction

- Firstly, we develop a new representation for the finite-order Volterra filters. This representation introduces a novel partitioning of the Volterra kernels.

Introduction

- Firstly, we develop a new representation for the finite-order Volterra filters. This representation introduces a novel partitioning of the Volterra kernels.
- Next, we formulate a novel exact identification method for Volterra filters.

Introduction

- Firstly, we develop a new representation for the finite-order Volterra filters. This representation introduces a novel partitioning of the Volterra kernels.
- Next, we formulate a novel exact identification method for Volterra filters.
- This identification method is based on the novel representation we develop and uses deterministic sequences consisting of impulses with distinct levels.

Introduction

- We know that the unit impulse response is insufficient to fully characterize a nonlinear system unlike linear time-invariant systems.

Introduction

- We know that the unit impulse response is insufficient to fully characterize a nonlinear system unlike linear time-invariant systems.
- The identification method might be considered as a successful extension of the impulse response of the linear, time-invariant systems to the realm of nonlinear systems.

Introduction

- We know that the unit impulse response is insufficient to fully characterize a nonlinear system unlike linear time-invariant systems.
- The identification method might be considered as a successful extension of the impulse response of the linear, time-invariant systems to the realm of nonlinear systems.
- The developed method indeed includes identification using the unit impulse response as a subcase when the system under consideration is a linear system.

Introduction

To our best knowledge, this method is the first full-scale generalization of the impulse response to the finite order Volterra type nonlinear systems.

Introduction

- Our identification method is exact.

Introduction

- Our identification method is exact.
- Our method calculates each Volterra kernel individually.

Introduction

- Our identification method is exact.
- Our method calculates each Volterra kernel individually.
- Our method calculates directly the Volterra kernels, instead of calculating first some intermediary representation.

Introduction

- Our identification method is exact.
- Our method calculates each Volterra kernel individually.
- Our method calculates directly the Volterra kernels, instead of calculating first some intermediary representation.
- Our method does not introduce and identify any kernels which are redundant for the regular Volterra filter.

Introduction

- Our identification method is exact.
- Our method calculates each Volterra kernel individually.
- Our method calculates directly the Volterra kernels, instead of calculating first some intermediary representation.
- Our method does not introduce and identify any kernels which are redundant for the regular Volterra filter.
- Our method is parsimonious in the number of kernels identified and in the length of the input sequence utilized to identify them.

Introduction

- We show that the input sequence we develop for identification is persistently exciting for the Volterra filters under consideration.

Introduction

- We show that the input sequence we develop for identification is persistently exciting for the Volterra filters under consideration.
- We further prove the equivalence of our identification algorithm to the least squares solution formulation.

Introduction

- We apply the novel identification method to the identification of the Volterra kernels of nonlinear communication channels modelled as third-order Volterra filters.

Introduction

- We apply the novel identification method to the identification of the Volterra kernels of nonlinear communication channels modelled as third-order Volterra filters.
- We demonstrate with several simulations that the identification algorithm can produce better parameter estimates than some most recent algorithms in the literature.

Introduction

- A secondary contribution of this dissertation is in the area of orthogonal realizations for Volterra filters.

Introduction

- A secondary contribution of this dissertation is in the area of orthogonal realizations for Volterra filters.
- We present a novel fully orthogonal structure for the realization of Volterra filters. This structure is based on a recently proposed 2D orthogonal lattice model.

Volterra Filters

Volterra Filters

- The use of linear system models has been well established with successful applications.

Volterra Filters

- The use of linear system models has been well established with successful applications.
- However, there are still a large number of problems where one has to resort to nonlinear system models.

Volterra Filters

- The use of linear system models has been well established with successful applications.
- However, there are still a large number of problems where one has to resort to nonlinear system models.
- Linear systems are fully described by their impulse response.

Volterra Filters

- The use of linear system models has been well established with successful applications.
- However, there are still a large number of problems where one has to resort to nonlinear system models.
- Linear systems are fully described by their impulse response.
- There is no such unified framework for the representation of nonlinear systems. There are various categories for modelling nonlinear systems.

Volterra Filters

- The use of linear system models has been well established with successful applications.
- However, there are still a large number of problems where one has to resort to nonlinear system models.
- Linear systems are fully described by their impulse response.
- There is no such unified framework for the representation of nonlinear systems. There are various categories for modelling nonlinear systems.
- In this dissertation we will be dealing with nonlinear polynomial system models based on the Volterra series representation.

Volterra Filters

- Volterra filters based on the Volterra series have been an attractive nonlinear system class due to some desired properties.

Volterra Filters

- Volterra filters based on the Volterra series have been an attractive nonlinear system class due to some desired properties.
  - Volterra filters bear similarities to the well-developed linear system theory.

Volterra Filters

- Volterra filters based on the Volterra series have been an attractive nonlinear system class due to some desired properties.
  - Volterra filters bear similarities to the well-developed linear system theory.
  - Volterra filters can approximate a large class of nonlinear systems with a finite number of coefficients.

Volterra Filters

- Volterra filters based on the Volterra series have been an attractive nonlinear system class due to some desired properties.
  - Volterra filters bear similarities to the well-developed linear system theory.
  - Volterra filters can approximate a large class of nonlinear systems with a finite number of coefficients.
  - Many real world processes lend themselves to get modelled naturally by polynomial systems.

Volterra Filters - Overview

Volterra Filters - Overview

- In this section we will provide an overview of the Volterra series representation for nonlinear systems.

Volterra Filters - Overview

- In this section we will provide an overview of the Volterra series representation for nonlinear systems.
- The Taylor series expansion with memory is known as the Volterra series. The naming is due to Vito Volterra, the Italian mathematician who introduced this polynomial series.

Volterra Filters - Overview

- For a general continuous-time nonlinear system the input-output relationship is represented by the following infinite continuous-time Volterra series integral.

Volterra Filters - Overview

- For a general continuous-time nonlinear system the input-output relationship is represented by the following infinite continuous-time Volterra series integral.

$$
\begin{aligned}
y(t) &= b_0 + \int_{-\infty}^{\infty} b_1(\tau_1)x(t-\tau_1)d\tau_1 \\
&+ \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} b_2(\tau_1,\tau_2)x(t-\tau_1)x(t-\tau_2)d\tau_1 d\tau_2 + \dots \\
&+ \int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} b_M(\tau_1,\tau_2,\dots,\tau_M)x(t-\tau_1)x(t-\tau_2)\cdots x(t-\tau_M)d\tau_1 d\tau_2\cdots d\tau_M \\
&+ \dots
\end{aligned}
\tag{1}
$$

Volterra Filters - Overview

- The equivalent discrete-time Volterra series sum is given as follows.

Volterra Filters - Overview

- The equivalent discrete-time Volterra series sum is given as follows.

$$
\begin{aligned}
y(n) = & \bar{b}_0 + \sum_{i_1=-\infty}^{\infty} \bar{b}_1(i_1)x(n-i_1) \\
& + \sum_{i_1=-\infty}^{\infty} \sum_{i_2=-\infty}^{\infty} \bar{b}_2(i_1, i_2)x(n-i_1)x(n-i_2) + \dots \\
& + \sum_{i_1=-\infty}^{\infty} \cdots \sum_{i_M=-\infty}^{\infty} \bar{b}_M(i_1, i_2, \dots, i_M)x(n-i_1)x(n-i_2)\cdots x(n-i_M) \\
& + \dots
\end{aligned} \tag{2}
$$

Truncated Volterra Filters

Truncated Volterra Filters

- The truncated or doubly finite Volterra series is obtained by confining the infinite summations to finite values.

Truncated Volterra Filters

- The truncated or doubly finite Volterra series is obtained by confining the infinite summations to finite values.
- The truncated Volterra series is suitable for the modelling of a wide variety of nonlinearities encountered in real-life systems.

Truncated Volterra Filters

- The truncated Volterra series is given as

Truncated Volterra Filters

- The truncated Volterra series is given as

$$
\begin{aligned}
y(n) &= \sum_{i_1=0}^{N} b_1(i_1)x(n-i_1) \\
&+ \sum_{i_1=0}^{N}\sum_{i_2=i_1}^{N} b_2(i_1,i_2)x(n-i_1)x(n-i_2) + \ldots \\
&+ \sum_{i_1=0}^{N}\sum_{i_2=i_1}^{N}\cdots\sum_{i_M=i_{M-1}}^{N} b_M(i_1,i_2,\ldots,i_M)x(n-i_1)x(n-i_2)\cdots x(n-i_M)
\end{aligned} \tag{3}
$$

Truncated Volterra Filters

- This representation is called as the triangular Volterra representation in the literature.

Truncated Volterra Filters

- This representation is called as the triangular Volterra representation in the literature.
- In this thesis we will be calling this representation as the Volterra filter.

Truncated Volterra Filters

- This representation is called as the triangular Volterra representation in the literature.
- In this thesis we will be calling this representation as the Volterra filter.
- The corresponding kernels will be called simply as the Volterra kernels to avoid confusion.

Truncated Volterra Filters

- This representation is called as the triangular Volterra representation in the literature.
- In this thesis we will be calling this representation as the Volterra filter.
- The corresponding kernels will be called simply as the Volterra kernels to avoid confusion.
- We will use the triangular representation as the starting point for our studies to develop novel Volterra representations.

Truncated Volterra Filters

- The output $y(n)$ in (3) can be rewritten as below.

$$y(n) = \mathcal{N}[x(n)] = \sum_{k=0}^{M} y_k(n) \tag{4}$$

in which

$$y_k(n) = \mathcal{B}_k[x(n)] \tag{5}$$

Truncated Volterra Filters

- The output $y(n)$ in (3) can be rewritten as below.

$$y(n) = \mathcal{N}[x(n)] = \sum_{k=0}^{M} y_k(n) \tag{4}$$

  in which

$$y_k(n) = \mathcal{B}_k[x(n)] \tag{5}$$

- $\mathcal{N}[\,\cdot\,]$ represents the nonlinear system under consideration.

Truncated Volterra Filters

- The output $y(n)$ in (3) can be rewritten as below.

$$y(n) = \mathcal{N}[x(n)] = \sum_{k=0}^{M} y_k(n) \tag{4}$$

in which

$$y_k(n) = \mathcal{B}_k[x(n)] \tag{5}$$

- $\mathcal{N}[\,\cdot\,]$ represents the nonlinear system under consideration.
- $\mathcal{B}_k[\,\cdot\,]$ represents a k^{th}-order Volterra subsystem with an input-output relationship constituting k summations.

Truncated Volterra Filters

- The input-output relation for each of these subsystems is as indicated below.

Truncated Volterra Filters

- The input-output relation for each of these subsystems is as indicated below.

$$
\begin{aligned}
y_k(n) &= \mathcal{B}_k[x(n)] \\
&= \sum_{i_1=0}^{N} \sum_{i_2=i_1}^{N} \cdots \sum_{i_k=i_{k-1}}^{N} b_k(i_1, i_2, \ldots, i_k) x(n-i_1) x(n-i_2) \cdots x(n-i_k)
\end{aligned}
\tag{6}
$$

Volterra Filter - Figure

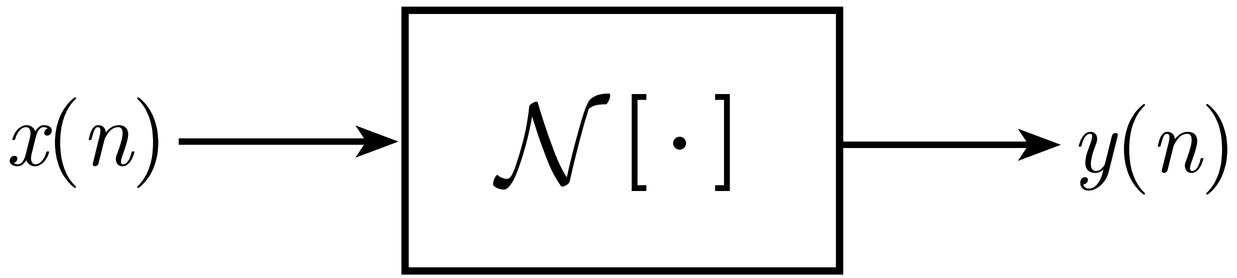


Figure 1: The input-output relationship of the nonlinear Volterra filter $\mathcal{N}[\,\cdot\,]$.

Volterra Filter - Sum of Subsystems

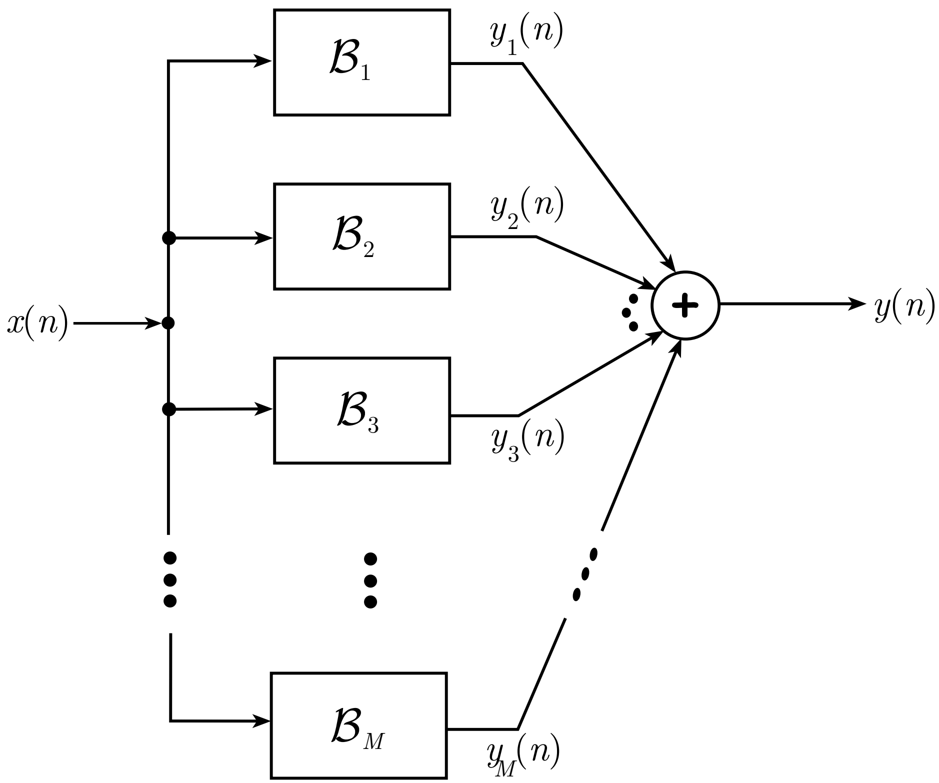


Figure 2: The nonlinear Volterra filter $\mathcal{N}$ as a sum of nonlinear subsystems $\mathcal{B}_k$.

Novel Representation

Novel Representation

- In this section, the M^{th}-order discrete, causal, time-invariant Volterra system is reformulated, and a new representation for the Volterra system is given.

Novel Representation

- In this section, the M^{th}-order discrete, causal, time-invariant Volterra system is reformulated, and a new representation for the Volterra system is given.
- This novel representation will enable us to devise an exact closed form algorithm, for identifying the Volterra kernels using deterministic multilevel sequences, in the coming chapters.

Novel Representation

- The M^{th}-order nonlinear system, $\mathcal{N}[\,\cdot\,]$, under consideration can be modelled by the triangular representation.

Novel Representation

- The M^{th}-order nonlinear system, $\mathcal{N}[\,\cdot\,]$, under consideration can be modelled by the triangular representation.

$$y(n) = \mathcal{N}[x(n)] = \sum_{k=1}^{M} y_k(n) = \sum_{k=1}^{M} \mathcal{B}_k[x(n)]$$

$$= \sum_{k=1}^{M} \underbrace{\sum_{i_1=0}^{N} \sum_{i_2=i_1}^{N} \cdots \sum_{i_k=i_{k-1}}^{N} b_k\left(i_1, i_2, \ldots, i_k\right) x(n-i_1)x(n-i_2)\cdots x(n-i_k)}_{\mathcal{B}_k[x(n)]} \qquad (7)$$

Novel Representation

- We introduce a new representation for the Volterra system by rearranging the Volterra kernels.

Novel Representation

- We introduce a new representation for the Volterra system by rearranging the Volterra kernels.
- We propose that the output $y(n)$ can be considered as the sum of the outputs of M different multivariate cross-term nonlinear subsystems, $\mathcal{H}^{(\ell)}$, $\ell = 1, \ldots, M$.

Novel Representation

- We introduce a new representation for the Volterra system by rearranging the Volterra kernels.
- We propose that the output $y(n)$ can be considered as the sum of the outputs of M different multivariate cross-term nonlinear subsystems, $\mathcal{H}^{(\ell)}$, $\ell = 1, \ldots, M$.

$$y(n) = \mathcal{N}[x(n)] = \sum_{\ell=1}^{M} y^{(\ell)}(n) \tag{8a}$$

Novel Representation

- We introduce a new representation for the Volterra system by rearranging the Volterra kernels.
- We propose that the output $y(n)$ can be considered as the sum of the outputs of M different multivariate cross-term nonlinear subsystems, $\mathcal{H}^{(\ell)}$, $\ell = 1, \ldots, M$.

$$y(n) = \mathcal{N}[x(n)] = \sum_{\ell=1}^{M} y^{(\ell)}(n) \tag{8a}$$

$$y^{(\ell)}(n) = \mathcal{H}^{(\ell)}[x(n)] \tag{8b}$$

Novel Representation

- The input-output relation for each of the subsystems $\mathcal{H}^{(\ell)}$, $\ell = 1, 2, \ldots, M$ is as indicated below.

Novel Representation

- The input-output relation for each of the subsystems $\mathcal{H}^{(\ell)}$, $\ell = 1, 2, \ldots, M$ is as indicated below.

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

$$\mathcal{H}^{(\ell)}[x(n)] = \sum_{q_1=1}^{Q_1} \cdots \sum_{q_{\ell-1}=1}^{Q_{\ell-1}} \sum_{i=0}^{N-\bar{q}_{\ell-1}} \mathbf{h}^{(\ell)^T}(q_1, \ldots, q_{\ell-1}; i)\, \mathbf{x}^{(\ell)}(q_1, \ldots, q_{\ell-1}; n-i)$$
$$\text{for } 2 \leqslant \ell \leqslant M \tag{9}$$

Novel Representation - Figure

Novel Representation - Figure

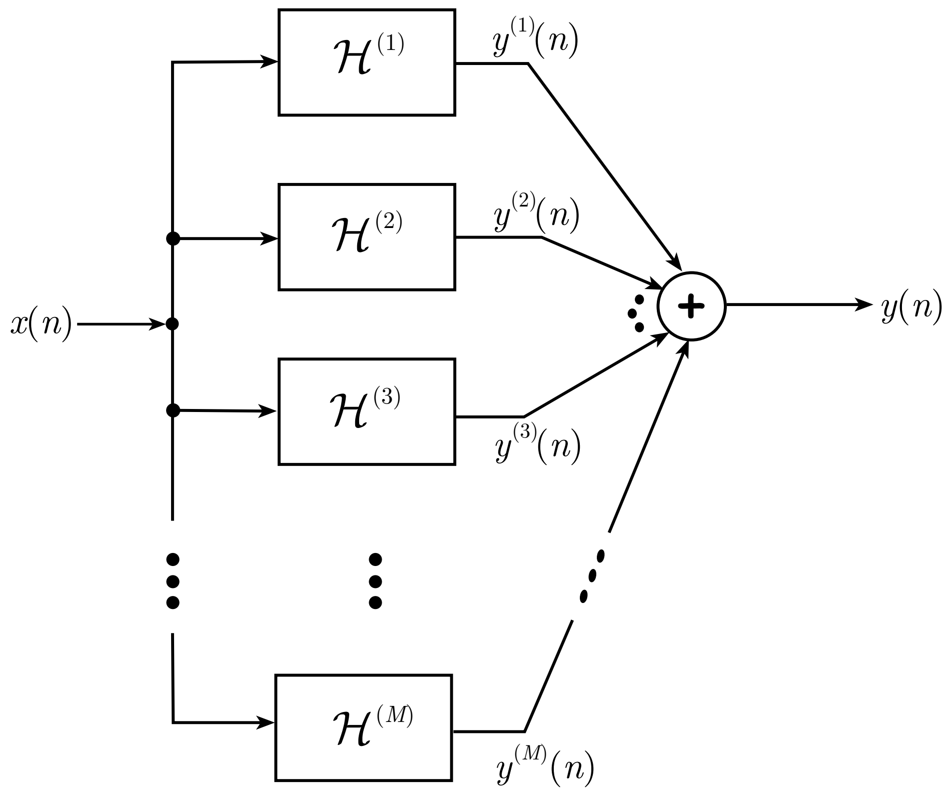


Figure 3: The novel decomposition for the nonlinear Volterra filter $\mathcal{N}$ as a sum of cross-term subsystems $\mathcal{H}^{(\ell)}$, $\ell = 1, \ldots, M$.

Novel Representation

- The symbol $\mathcal{H}^{(\ell)}[\cdot]$ 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.

Novel Representation

- The symbol $\mathcal{H}^{(\ell)}[\cdot]$ 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 given in terms of the triangular kernels and the input signal $x(n)$, respectively.

Novel Representation

$$\mathbf{h}^{(1)}(i) = \begin{bmatrix} h_1^{(1)}(i) \\ h_2^{(1)}(i) \\ \vdots \\ h_M^{(1)}(i) \end{bmatrix} = \begin{bmatrix} b_1(i) \\ b_2(i,i) \\ \vdots \\ b_M(i,\ldots,i) \end{bmatrix} \tag{10}$$

$$\mathbf{x}^{(1)}(n) = \begin{bmatrix} x(n) \\ x^2(n) \\ \vdots \\ x^M(n) \end{bmatrix} \tag{11}$$

Novel Representation

- The ℓ-D input vector in (9) can be expressed in the following form:

Novel Representation

- The ℓ-D input vector in (9) can be expressed in the following form:

$$\mathbf{x}^{(\ell)}(q_1,\ldots,q_{\ell-1};n) = \begin{bmatrix} x_\ell^{(\ell)}(q_1,\ldots,q_{\ell-1};n) \\ \mathbf{x}_{\ell+1}^{(\ell)}(q_1,\ldots,q_{\ell-1};n) \\ \vdots \\ \mathbf{x}_M^{(\ell)}(q_1,\ldots,q_{\ell-1};n) \end{bmatrix} \tag{12}$$

Novel Representation

- The ℓ-D input vector in (9) can be expressed in the following form:

$$\mathbf{x}^{(\ell)}(q_1, \ldots, q_{\ell-1}; n) = \begin{bmatrix} x_\ell^{(\ell)}(q_1, \ldots, q_{\ell-1}; n) \\ \mathbf{x}_{\ell+1}^{(\ell)}(q_1, \ldots, q_{\ell-1}; n) \\ \vdots \\ \mathbf{x}_M^{(\ell)}(q_1, \ldots, q_{\ell-1}; n) \end{bmatrix} \tag{12}$$

in which

Novel Representation

- The ℓ-D input vector in (9) can be expressed in the following form:

$$\mathbf{x}^{(\ell)}\left(q_1,\ldots,q_{\ell-1};n\right) = \begin{bmatrix} x_\ell^{(\ell)}(q_1,\ldots,q_{\ell-1};n) \\ \mathbf{x}_{\ell+1}^{(\ell)}(q_1,\ldots,q_{\ell-1};n) \\ \vdots \\ \mathbf{x}_M^{(\ell)}(q_1,\ldots,q_{\ell-1};n) \end{bmatrix} \tag{12}$$

in which

$$x_\ell^{(\ell)}(q_1,\ldots,q_{\ell-1};n) = x(n)\,x(n-q_1)\,\cdots\,x(n-q_1-\cdots-q_{\ell-1}) \tag{13}$$

Novel Representation

$$\mathbf{x}_k^{(\ell)}(q_1, \ldots, q_{\ell-1}; n) \equiv \left[x_k^{(p_1, \cdots, p_\ell)}(q_1, \ldots, q_{\ell-1}; n) \right]_{\sigma(p_1, \ldots, p_\ell)} \qquad (14)$$

Novel Representation

$$\mathbf{x}_k^{(\ell)}(q_1, \ldots, q_{\ell-1}; n) \equiv \left[x_k^{(p_1, \cdots, p_\ell)}(q_1, \ldots, q_{\ell-1}; n) \right]_{\sigma(p_1, \ldots, p_\ell)} \tag{14}$$

- The subinput vector $\mathbf{x}_k^{(\ell)}(q_1, \ldots, q_{\ell-1}; n)$ for $k = \ell, \ell + 1, \ldots, M$ in (12) consists of all possible input products of degree k.

Novel Representation

$$\mathbf{x}_k^{(\ell)}(q_1,\ldots,q_{\ell-1};n) \equiv \left[x_k^{(p_1,\cdots,p_\ell)}(q_1,\ldots,q_{\ell-1};n) \right]_{\sigma(p_1,\ldots,p_\ell)} \tag{14}$$

- The subinput vector $\mathbf{x}_k^{(\ell)}(q_1,\ldots,q_{\ell-1};n)$ for $k=\ell,\ell+1,\ldots,M$ in (12) consists of all possible input products of degree k.

$$x_k^{(p_1,\cdots,p_\ell)}(q_1,\ldots,q_{\ell-1};n) = x^{p_1}(n)x^{p_2}(n-q_1)\cdots x^{p_\ell}(n-q_1-\cdots-q_{\ell-1}) \tag{15}$$

Novel Representation

Example 2.1: We consider $\ell = 3$ and $M = 5$. $\mathbf{x}^{(3)}(q_1, q_2; n)$ will be given as

$$\mathbf{x}^{(3)}(q_1, q_2; n) = \begin{bmatrix} x_3^{(3)}(q_1, q_2; n) \\ \mathbf{x}_4^{(3)}(q_1, q_2; n) \\ \mathbf{x}_5^{(3)}(q_1, q_2; n) \end{bmatrix} \tag{16}$$

Here,

$$x_3^{(3)}(q_1, q_2; n) = x(n)\, x(n - q_1)\, x(n - q_1 - q_2) \tag{17}$$

Novel Representation

For $\mathbf{x}_4^{(3)}(q_1, q_2; n)$, the $\binom{3}{2} = 3$ possible combinations can be written as $\sigma(p_1, p_2, p_3) = \{(2, 1, 1)\ (1, 2, 1),\ (1, 1, 2)\}$.

Hence, $\mathbf{x}_4^{(3)}(q_1, q_2; n)$ is written as

$$\mathbf{x}_4^{(3)}(q_1, q_2; n) = \begin{bmatrix} x^2(n)\, x(n - q_1)\, x(n - q_1 - q_2) \\ x(n)\, x^2(n - q_1)\, x(n - q_1 - q_2) \\ x(n)\, x(n - q_1)\, x^2(n - q_1 - q_2) \end{bmatrix} \tag{18}$$

Novel Representation

For $\mathbf{x}_5^{(3)}(q_1, q_2; n)$, all $\binom{4}{2} = 6$ possible combinations can be written as,

Novel Representation

For $\mathbf{x}_5^{(3)}(q_1, q_2; n)$, all $\binom{4}{2} = 6$ possible combinations can be written as,

$$\sigma(p_1, p_2, p_3) = \Big\{(3,1,1)\ (2,2,1),\ (2,1,2),\ (1,3,1),\ (1,2,2),\ (1,1,3)\Big\}$$

Novel Representation

For $\mathbf{x}_5^{(3)}(q_1, q_2; n)$, all $\binom{4}{2} = 6$ possible combinations can be written as,

$$\sigma(p_1, p_2, p_3) = \left\{(3,1,1)\ (2,2,1),\ (2,1,2),\ (1,3,1),\ (1,2,2),\ (1,1,3)\right\}$$

$$\mathbf{x}_5^{(3)}(q_1, q_2; n) = \begin{bmatrix} x^3(n)\, x(n-q_1)\, x(n-q_1-q_2) \\ x^2(n)\, x^2(n-q_1)\, x(n-q_1-q_2) \\ x^2(n)\, x(n-q_1)\, x^2(n-q_1-q_2) \\ x(n)\, x^3(n-q_1)\, x(n-q_1-q_2) \\ x(n)\, x^2(n-q_1)\, x^2(n-q_1-q_2) \\ x(n)\, x(n-q_1)\, x^3(n-q_1-q_2) \end{bmatrix} \tag{19}$$

Novel Representation

- The ℓ-D kernel vectors in (9) can be written in terms of subkernels as

$$\mathbf{h}^{(\ell)}(q_1,\ldots,q_{\ell-1};i) = \begin{bmatrix} h_\ell^{(\ell)}(q_1,\ldots,q_{\ell-1};i) \\ \mathbf{h}_{\ell+1}^{(\ell)}(q_1,\ldots,q_{\ell-1};i) \\ \vdots \\ \mathbf{h}_M^{(\ell)}(q_1,\ldots,q_{\ell-1};i) \end{bmatrix} \tag{20}$$

in which

$$h_\ell^{(\ell)}(q_1,\ldots,q_{\ell-1};i) = h_\ell^{(1,1,\cdots,1)}(q_1,\ldots,q_{\ell-1};i) \tag{21}$$

Novel Representation

$$\mathbf{h}_k^{(\ell)}(q_1, \ldots, q_{\ell-1}; i) \equiv \left[h_k^{(p_1, p_2, \cdots p_\ell)}(q_1, \ldots, q_{\ell-1}; i) \right]_{\sigma(p_1, p_2, \ldots, p_\ell)} \quad (22)$$

- Here, the subkernel vector $\mathbf{h}_k^{(\ell)}(q_1, \ldots, q_{\ell-1}; i)$ corresponds to the subinput vector $\mathbf{x}_k^{(\ell)}(q_1, \ldots, q_{\ell-1}; n-i)$ defined in (14). $\mathbf{h}_k^{(\ell)}(q_1, \ldots, q_{\ell-1}; i)$ consists of all the Volterra kernels of degree k with ℓ cross-terms.

Novel Representation

Example 2.2: Continuing the previous example, let us consider $\ell = 3$ and $M = 5$.

$$\mathbf{h}^{(3)}(q_1, q_2; i) = \begin{bmatrix} h_3^{(3)}(q_1, q_2; i) \\ \mathbf{h}_4^{(3)}(q_1, q_2; i) \\ \mathbf{h}_5^{(3)}(q_1, q_2; i) \end{bmatrix} \tag{23}$$

Here,

$$h_3^{(3)}(q_1, q_2; i) = h_3^{(1,1,1)}(q_1, q_2; i) \tag{24}$$

Novel Representation

$\mathbf{h}_4^{(3)}(q_1, q_2; i)$ for $\ell = 3$ can be written as,

$$\mathbf{h}_4^{(3)}(q_1, q_2; i) = \begin{bmatrix} h_4^{(2,1,1)}(q_1, q_2; i) \\ h_4^{(1,2,1)}(q_1, q_2; i) \\ h_4^{(1,1,2)}(q_1, q_2; i) \end{bmatrix} \tag{25}$$

Novel Representation

$\mathbf{h}_5^{(3)}(q_1, q_2; i)$ is written as,

$$\mathbf{h}_5^{(3)}(q_1, q_2; i) = \begin{bmatrix} h_5^{(3,1,1)}(q_1, q_2; i) \\ h_5^{(2,2,1)}(q_1, q_2; i) \\ h_5^{(2,1,2)}(q_1, q_2; i) \\ h_5^{(1,3,1)}(q_1, q_2; i) \\ h_5^{(1,2,2)}(q_1, q_2; i) \\ h_5^{(1,1,3)}(q_1, q_2; i) \end{bmatrix} \tag{26}$$

Novel Representation

- There exists an equivalent triangular Volterra kernel $b_k(i_1, i_2, \ldots, i_k)$ as given in (7) for each component of the subkernel vector $\mathbf{h}_k^{(\ell)}(q_1, \ldots, q_{\ell-1}; i)$,

Novel Representation

- There exists an equivalent triangular Volterra kernel $b_k(i_1, i_2, \ldots, i_k)$ as given in (7) for each component of the subkernel vector $\mathbf{h}_k^{(\ell)}(q_1, \ldots, q_{\ell-1}; i)$,
- The relationship between the triangular Volterra kernels and cross-term Volterra kernels is as given below.

Novel Representation

- There exists an equivalent triangular Volterra kernel $b_k(i_1, i_2, \ldots, i_k)$ as given in (7) for each component of the subkernel vector $\mathbf{h}_k^{(\ell)}(q_1, \ldots, q_{\ell-1}; i)$,
- The relationship between the triangular Volterra kernels and cross-term Volterra kernels is as given below.

$$h_k^{(p_1,p_2,\ldots,p_\ell)}(q_1, \ldots, q_{\ell-1}; i) = \\ b_k(\underbrace{i, \ldots, i}_{p_1}, \underbrace{i+\bar{q}_1, \ldots, i+\bar{q}_1}_{p_2}, \ldots, \underbrace{i+\bar{q}_{\ell-1}, \ldots, i+\bar{q}_{\ell-1}}_{p_\ell}) \quad (27)$$

Novel Representation

Example 2.3: We continue with Example 2.2. We want to find the representations for the kernel vectors, but this time in terms of the Volterra kernels. The kernel vectors for $\ell = 3$ and $M = 5$ can now be written as

$$h_3^{(3)}(q_1, q_2; i) = b_3(i, i+q_1, i+q_2) \tag{28}$$

$$\mathbf{h}_4^{(3)}(q_1, q_2; i) = \begin{bmatrix} h_4^{(2,1,1)}(q_1, q_2; i) \\ h_4^{(1,2,1)}(q_1, q_2; i) \\ h_4^{(1,1,2)}(q_1, q_2; i) \end{bmatrix} = \begin{bmatrix} b_4(i, i, i+q_1, i+q_2) \\ b_4(i, i+q_1, i+q_1, i+q_2) \\ b_4(i, i+q_1, i+q_2, i+q_2) \end{bmatrix} \tag{29}$$

Novel Representation

$$\mathbf{h}_5^{(3)}(q_1, q_2; i) = \begin{bmatrix} h_5^{(3,1,1)}(q_1, q_2; i) \\ h_5^{(2,2,1)}(q_1, q_2; i) \\ h_5^{(2,1,2)}(q_1, q_2; i) \\ h_5^{(1,3,1)}(q_1, q_2; i) \\ h_5^{(1,2,2)}(q_1, q_2; i) \\ h_5^{(1,1,3)}(q_1, q_2; i) \end{bmatrix} = \begin{bmatrix} b_5(i, i, i, i+q_1, i+q_2) \\ b_5(i, i, i+q_1, i+q_1, i+q_2) \\ b_5(i, i, i+q_1, i+q_2, i+q_2) \\ b_5(i, i+q_1, i+q_1, i+q_1, i+q_2) \\ b_5(i, i+q_1, i+q_1, i+q_2, i+q_2) \\ b_5(i, i+q_1, i+q_2, i+q_2, i+q_2) \end{bmatrix} \tag{30}$$

Novel Representation

Example 2.4: Let us give an example for the novel representation for a system with $M = 3$ and $N = 2$. The usual (triangular) Volterra representation is given in the following form.

$$y(n) = \sum_{i_1=0}^{2} b_1(i_1)x(n-i_1) + \sum_{i_1=0}^{2}\sum_{i_2=i_1}^{2} b_2(i_1,i_2)x(n-i_1)x(n-i_2)$$
$$+ \sum_{i_1=0}^{2}\sum_{i_2=i_1}^{2}\sum_{i_3=i_2}^{2} b_3(i_1,i_2,i_3)x(n-i_1)x(n-i_2)x(n-i_3) \tag{31}$$

Novel Representation

$$
\begin{aligned}
y(n) &= b_1(0)x(n) + b_1(1)x(n-1) + b_1(2)x(n-2) + b_2(0,0)x^2(n) \\
&+ b_2(1,1)x^2(n-1) + b_2(2,2)x^2(n-2) + b_2(0,1)x(n)x(n-1) \\
&+ b_2(1,2)x(n-1)x(n-2) + b_2(0,2)x(n)x(n-2) + b_3(0,0,0)x^3(n) \\
&+ b_3(1,1,1)x^3(n-1) + b_3(2,2,2)x^3(n-2) + b_3(0,0,1)x^2(n)x(n-1) \\
&+ b_3(1,1,2)x^2(n-1)x(n-2) + b_3(0,0,2)x^2(n)x(n-2) \\
&+ b_3(0,1,1)x(n)x^2(n-1) + b_3(1,2,2)x(n-1)x^2(n-2) \\
&+ b_3(0,2,2)x(n)x^2(n-2) + b_3(0,1,2)x(n)x(n-1)x(n-2)
\end{aligned}
\tag{32}
$$

Novel Representation

For this Volterra filter, using (8), the novel representation that we introduced will be given as follows.

$$
\begin{aligned}
y(n) =& \sum_{i=0}^{2} \mathbf{h}^{(1)^T}(i)\, \mathbf{x}^{(1)}(n-i) + \sum_{q_1=1}^{2} \sum_{i=0}^{2-q_1} \mathbf{h}^{(2)^T}(q_1; i)\mathbf{x}^{(2)}(q_1; n-i) \\
&+ \sum_{q_1=1}^{1} \sum_{q_2=1}^{2-q_1} \sum_{i=0}^{2-\bar{q}_2} \mathbf{h}^{(3)^T}(q_1, q_2; i)\, \mathbf{x}^{(3)}(q_1, q_2; n-i) \\
=& \sum_{i=0}^{2} \mathbf{h}^{(1)^T}(i)\, \mathbf{x}^{(1)}(n-i) \\
&+ \mathbf{h}^{(2)^T}(1;0)\mathbf{x}^{(2)}(1;n) + \mathbf{h}^{(2)^T}(1;1)\mathbf{x}^{(2)}(1;n-1) + \mathbf{h}^{(2)^T}(2;0)\mathbf{x}^{(2)}(2;n) \\
&+ \mathbf{h}^{(3)^T}(1,1;0)\, \mathbf{x}^{(3)}(1,1;n)
\end{aligned}
$$

Novel Representation

Here,

$$\mathbf{h}^{(1)}(i) = \begin{bmatrix} h_1^{(1)}(i) \\ h_2^{(2)}(i) \\ h_3^{(3)}(i) \end{bmatrix} = \begin{bmatrix} b_1(i) \\ b_2(i,i) \\ b_3(i,i,i) \end{bmatrix}, \text{for } i = 0, 1, 2$$

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

Novel Representation

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

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

Novel Representation

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

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

Novel Representation

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

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

Novel Representation

$$\mathbf{h}^{(3)}(1,1;0) = \left[\ h_3^{(1,1,1)}(1,1;0)\ \right] = \left[\ b_3(0,1,2)\ \right]$$

$$\mathbf{x}^{(3)}(1,1;n) = \left[\ x_3^{(3)}(1,1;n)\ \right] = \left[\ x(n)x(n-1)x(n-2)\ \right]$$

Novel Representation

- The novel cross-product kernel representation presented in this section does not increase the number of kernels in the Volterra filter

Novel Representation

- The novel cross-product kernel representation presented in this section does not increase the number of kernels in the Volterra filter
- We have grouped the Volterra kernels in an novel manner introducing the concept of delay-wise dimensionality and cross-term subsystem, rather than using the multiplicational order of the Volterra kernels to group the kernels.

Novel Representation

- The novel cross-product kernel representation presented in this section does not increase the number of kernels in the Volterra filter
- We have grouped the Volterra kernels in an novel manner introducing the concept of delay-wise dimensionality and cross-term subsystem, rather than using the multiplicational order of the Volterra kernels to group the kernels.
- This novel grouping enables us to devise an exact closed form algorithm for identifying the Volterra kernels using deterministic multilevel sequences.

Novel Identification Algorithm

Novel Identification Algorithm

Novel Identification Algorithm - 1D

Novel Identification Algorithm - 1D

- In the novel representation we decomposed the output of the overall nonlinear system in terms of the outputs of newly defined cross-term subsystems $\mathcal{H}^{(\ell)}[\,\cdot\,]$, $\ell = 1, 2, \ldots, M$.

Novel Identification Algorithm - 1D

- In the novel representation we decomposed the output of the overall nonlinear system in terms of the outputs of newly defined cross-term subsystems $\mathcal{H}^{(\ell)}[\,\cdot\,]$, $\ell = 1, 2, \ldots, M$.
- We called the subsystem $\mathcal{H}^{(\ell)}[\,\cdot\,]$ as the ℓ-D subsystem.

Novel Identification Algorithm - 1D

- In the novel representation we decomposed the output of the overall nonlinear system in terms of the outputs of newly defined cross-term subsystems $\mathcal{H}^{(\ell)}[\,\cdot\,]$, $\ell = 1, 2, \ldots, M$.
- We called the subsystem $\mathcal{H}^{(\ell)}[\,\cdot\,]$ as the ℓ-D subsystem.
- 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 $\mathcal{H}^{(1)}[\,\cdot\,]$.

Novel Identification Algorithm - 1D

- In the novel representation we decomposed the output of the overall nonlinear system in terms of the outputs of newly defined cross-term subsystems $\mathcal{H}^{(\ell)}[\,\cdot\,]$, $\ell = 1, 2, \ldots, M$.
- We called the subsystem $\mathcal{H}^{(\ell)}[\,\cdot\,]$ as the ℓ-D subsystem.
- 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 $\mathcal{H}^{(1)}[\,\cdot\,]$.
- Using the novel cross-term representation,the higher dimensional outputs are zero for these multilevel single impulses.

Novel Identification Algorithm - 1D

$$\begin{aligned}\mathcal{N}[a_{m_1}\delta(n)] &= \mathcal{H}^{(1)}[a_{m_1}\delta(n)] \\ \mathcal{H}^{(\ell)}[a_{m_1}\delta(n)] &= 0,\ \text{for } \ell = 2,\dots,M\end{aligned} \tag{33}$$

This relation in (33) is shown pictorially in Fig. 4.

Novel Identification Algorithm - 1D

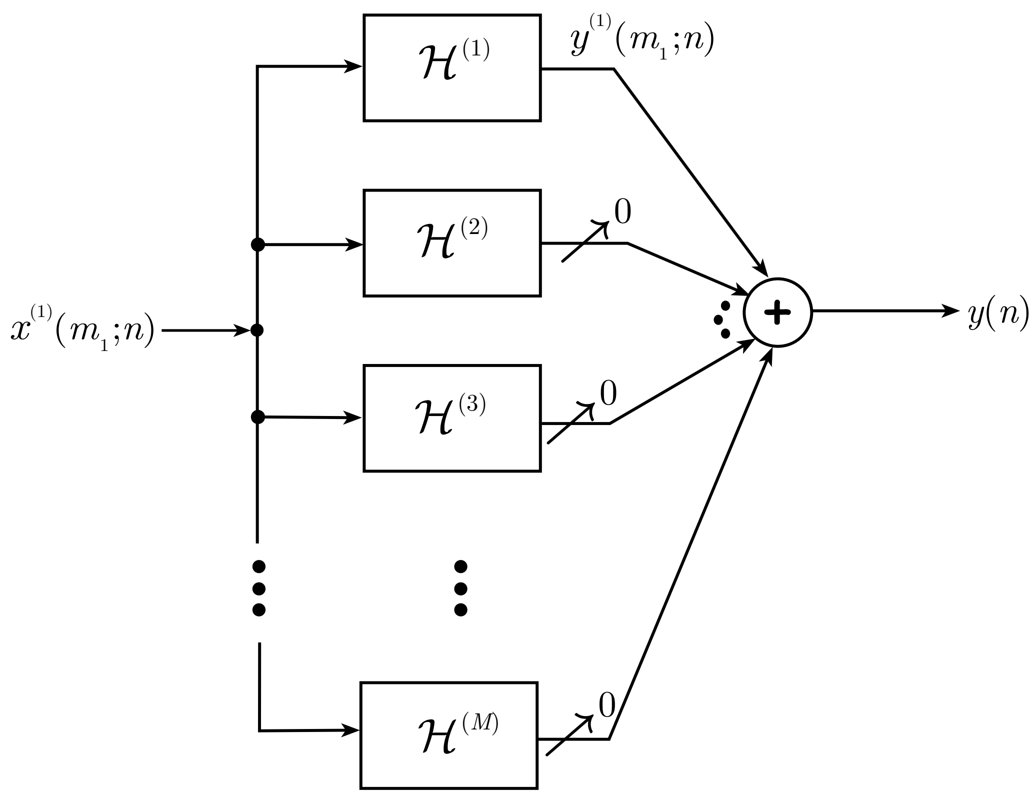


Figure 4: Pictorial description for (33). The output of the nonlinear system $\mathcal{N}$ for $x(n) = a_{m_1}\delta(n)$ is equal to the output of the subsystem $\mathcal{H}^{(1)}$.

Novel Identification Algorithm - 1D

- Hence, the output of the M^{th}-order nonlinear system, when the input is $x(n) = x^{(1)}(m_1; n)$, is given by

$$\begin{aligned} y(n) =& \mathcal{N}\left[x^{(1)}(m_1; n)\right] = y(m_1; n) \\ =& \mathcal{H}^{(1)}\left[x^{(1)}(m_1; n)\right] = y^{(1)}(m_1; n) \\ =& \sum_{i=0}^{N} \mathbf{h}^{(1)^T}(i)\mathbf{u}^{(1)}(m_1; n-i) \end{aligned} \tag{34}$$

Novel Identification Algorithm - 1D

$$\mathbf{u}^{(1)}(m_1;n) \equiv \begin{bmatrix} x^{(1)}(m_1;n) \\ {x^{(1)}}^2(m_1;n) \\ \vdots \\ {x^{(1)}}^M(m_1;n) \end{bmatrix} = \begin{bmatrix} a_{m_1} \\ a_{m_1}^2 \\ \vdots \\ a_{m_1}^M \end{bmatrix} \delta(n) \qquad (35)$$

Novel Identification Algorithm - 1D

- Here, $m_1 = 1, 2, \ldots, \binom{M}{1}$ denotes the ensemble index of the input sequence. Now we can write all $\binom{M}{1}$ outputs in the ensemble matrix form as follows:

$$\mathbf{y}_e^{(1)}(n) = \mathcal{N}\left[\mathbf{x}_e^{(1)}(n)\right] = \mathcal{H}^{(1)}\left[\mathbf{x}_e^{(1)}(n)\right] = \sum_{i=0}^{N} \mathbf{U}_e^{(1)}(n-i)\,\mathbf{h}^{(1)}(i) \quad (36)$$

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.

Novel Identification Algorithm - 1D

$$\mathbf{x}_e^{(1)}(n) \equiv \begin{bmatrix} x^{(1)}(1;n) \\ x^{(1)}(2;n) \\ \vdots \\ x^{(1)}(M;n) \end{bmatrix} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_M \end{bmatrix} \delta(n) \tag{37}$$

$$\mathbf{y}_e^{(1)}(n) \equiv \begin{bmatrix} \mathcal{N}\left[x^{(1)}(1;n)\right] \\ \mathcal{N}\left[x^{(1)}(2;n)\right] \\ \vdots \\ \mathcal{N}\left[x^{(1)}(M;n)\right] \end{bmatrix} = \begin{bmatrix} y^{(1)}(1;n) \\ y^{(1)}(2;n) \\ \vdots \\ y^{(1)}(M;n) \end{bmatrix} \tag{38}$$

Novel Identification Algorithm - 1D

$$\mathbf{U}_e^{(1)}(n) \equiv \begin{bmatrix} \mathbf{u}^{(1)^T}(1;n) \\ \mathbf{u}^{(1)^T}(2;n) \\ \vdots \\ \mathbf{u}^{(1)^T}(M;n) \end{bmatrix} \tag{39}$$

Novel Identification Algorithm - 1D

- $\mathbf{U}_e^{(1)}(n-i)$ in (36) 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

$$\mathbf{U}_e^{(1)} = \begin{bmatrix} a_1 & a_1^2 & \cdots & a_1^M \\ a_2 & a_2^2 & \cdots & a_2^M \\ \vdots & \vdots & \ddots & \vdots \\ a_M & a_M^2 & \cdots & a_M^M \end{bmatrix} \tag{40}$$

Novel Identification Algorithm -1D

- Hence, we get

$$\mathbf{y}_e^{(1)}(n) = \sum_{i=0}^{N} \mathbf{U}_e^{(1)}\, \mathbf{h}^{(1)}(i)\, \delta(n-i) = \mathbf{U}_e^{(1)}\, \mathbf{h}^{(1)}(n) \qquad (41)$$

Novel Identification Algorithm -1D

- Hence, we get

$$\mathbf{y}_e^{(1)}(n) = \sum_{i=0}^{N} \mathbf{U}_e^{(1)}\, \mathbf{h}^{(1)}(i)\, \delta(n-i) = \mathbf{U}_e^{(1)}\, \mathbf{h}^{(1)}(n) \tag{41}$$

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

$$\mathbf{h}^{(1)}(n) = \left[\mathbf{U}_e^{(1)}\right]^{-1} \mathbf{y}_e^{(1)}(n), \quad \text{for } \; n = 0, 1, \ldots, N \tag{42}$$

Novel Identification Algorithm -1D

- Hence, we get

$$\mathbf{y}_e^{(1)}(n) = \sum_{i=0}^{N} \mathbf{U}_e^{(1)}\, \mathbf{h}^{(1)}(i)\, \delta(n-i) = \mathbf{U}_e^{(1)}\, \mathbf{h}^{(1)}(n) \qquad (41)$$

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

$$\mathbf{h}^{(1)}(n) = \left[\mathbf{U}_e^{(1)}\right]^{-1} \mathbf{y}_e^{(1)}(n), \quad \text{for } \ n = 0, 1, \ldots, N \qquad (42)$$

- Fig. 5 depicts the identification method for 1-D kernels as outlined in this section and finalized in (42).

Novel Identification Algorithm - 1D

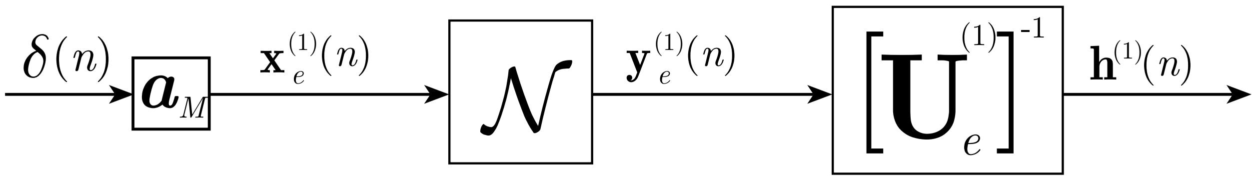


Figure 5: Method used for identification of 1-D Volterra kernels, $\mathbf{h}^{(1)}(n)$.

Novel Identification Algorithm - 1D

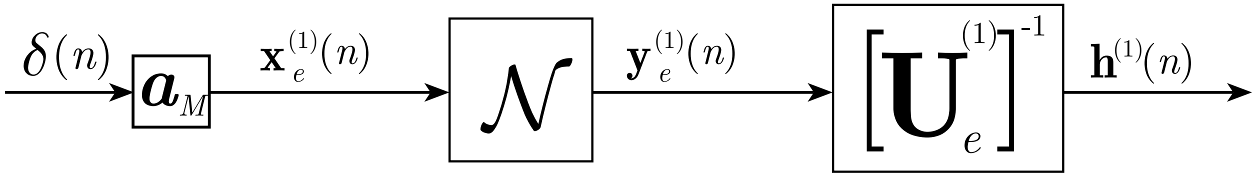


Figure 5: Method used for identification of 1-D Volterra kernels, $\mathbf{h}^{(1)}(n)$.

- Note that the linear FIR filter identification via the impulse response is covered by this method as the special case $M = 1$.

Novel Identification Algorithm - 1D

- In order to guarantee the Vandermonde like input matrix $\mathbf{U}_e^{(1)}$ in (40) to be nonsingular, the levels of the multilevel ensemble inputs must be chosen to be distinct and nonzero,

Novel Identification Algorithm - 1D

- In order to guarantee the Vandermonde like input matrix $\mathbf{U}_e^{(1)}$ in (40) to be nonsingular, the levels of the multilevel ensemble inputs must be chosen to be distinct and nonzero, i.e., $a_i \neq 0$ and $a_i \neq a_j$, $\forall\ i \neq j$.

Novel Identification Algorithm - 2D

Novel Identification Algorithm - 2D

- Now we are interested in computing the 2-D kernel vectors, $\mathbf{h}^{(2)}(q_1; i)$ for $q_1 = 1, \ldots, N$ and $i = 0, 1, \ldots, N - q_1$.

Novel Identification Algorithm - 2D

- Now we are interested in computing the 2-D kernel vectors, $\mathbf{h}^{(2)}(q_1; i)$ for $q_1 = 1, \ldots, N$ and $i = 0, 1, \ldots, N - q_1$.
- We use 2-D ensemble inputs which consist of two impulses with distinct amplitudes.

Novel Identification Algorithm - 2D

- Now we are interested in computing the 2-D kernel vectors, $\mathbf{h}^{(2)}(q_1; i)$ for $q_1 = 1, \ldots, N$ and $i = 0, 1, \ldots, N - q_1$.
- We use 2-D ensemble inputs which consist of two impulses with distinct amplitudes.
- The following 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 $\mathbf{h}^{(1)}(n - q_1)$.

Novel Identification Algorithm - 2D

$$
\begin{aligned}
x^{(2)}\big((m_1, m_2), q_1; n\big) &= x^{(1)}(m_1; n) + x^{(1)}(m_2; n - q_1) \\
&= a_{m_1}\delta(n) + a_{m_2}\delta(n - q_1) \qquad (43)
\end{aligned}
$$

$$\text{for } m_1 = 1, \ldots, M-1;\; m_2 = m_1 + 1, \ldots, M$$

Novel Identification Algorithm - 2D

$$
\begin{aligned}
x^{(2)}\big((m_1, m_2), q_1; n\big) &= x^{(1)}(m_1; n) + x^{(1)}(m_2; n - q_1) \\
&= a_{m_1}\delta(n) + a_{m_2}\delta(n - q_1) \qquad (43) \\
\text{for } m_1 = 1, \ldots, M-1;\ m_2 &= m_1 + 1, \ldots, M
\end{aligned}
$$

- It is possible to show that the 2-D input signal in (43) does not excite the Volterra kernels having more than two cross-terms, i.e.,

$$
\mathcal{H}^{(\ell)}\left[x^{(2)}\big((m_1; m_2), q_1; n\big)\right] = 0 \text{ for } \ell \geqslant 3. \qquad (44)
$$

Novel Identification Algorithm - 2D

$$
\begin{aligned}
x^{(2)}\big((m_1, m_2), q_1; n\big) &= x^{(1)}(m_1; n) + x^{(1)}(m_2; n - q_1) \\
&= a_{m_1}\delta(n) + a_{m_2}\delta(n - q_1) \qquad (43)
\end{aligned}
$$

$$
\text{for } m_1 = 1, \ldots, M-1;\ m_2 = m_1 + 1, \ldots, M
$$

- It is possible to show that the 2-D input signal in (43) does not excite the Volterra kernels having more than two cross-terms, i.e.,

$$
\mathcal{H}^{(\ell)}\Big[x^{(2)}\big((m_1; m_2), q_1; n\big)\Big] = 0 \text{ for } \ell \geqslant 3. \qquad (44)
$$

- (44) is depicted in Fig. 6.

Novel Identification Algorithm - 2D

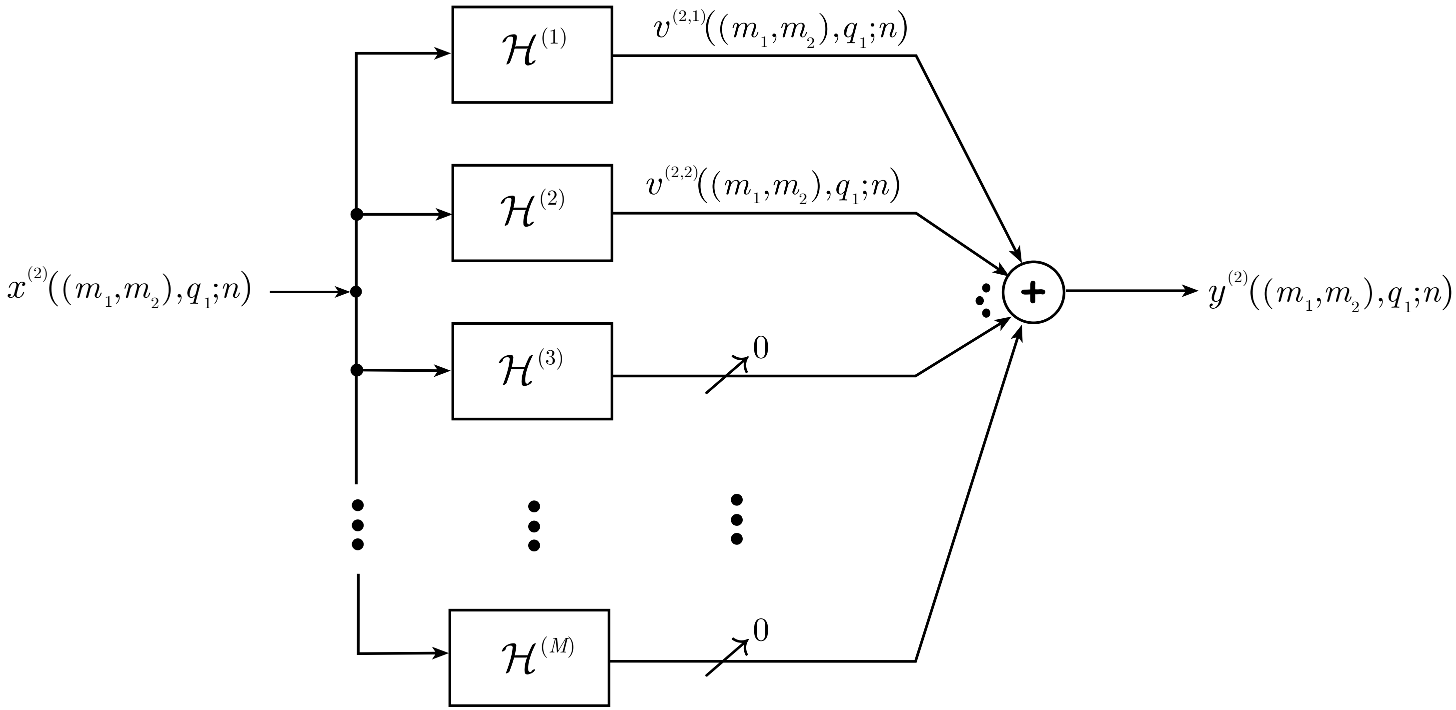


Figure 6: Pictorial description for (44). The output of the nonlinear system $\mathcal{N}$, for $x(n) = a_{m_1}\delta(n) + a_{m_2}\delta(n-q_1)$, is equal to the sum of the outputs of subsystems $\mathcal{H}^{(1)}$ and $\mathcal{H}^{(2)}$.

Novel Identification Algorithm - 2D

- Hence, the output for the input in (43) can be written as

$$
\begin{aligned}
\mathcal{N}\left[x^{(2)}((m_1, m_2), q_1; n)\right] &= y^{(2)}((m_1, m_2), q_1; n) \\
&= \mathcal{H}^{(1)}\left[x^{(2)}((m_1, m_2), q_1; n)\right] + \mathcal{H}^{(2)}\left[x^{(2)}((m_1, m_2), q_1; n)\right]
\end{aligned}
\tag{45}
$$

Novel Identification Algorithm - 2D

- Hence, the output for the input in (43) can be written as

$$\mathcal{N}\left[x^{(2)}((m_1, m_2), q_1; n)\right] = y^{(2)}((m_1, m_2), q_1; n)$$
$$= \mathcal{H}^{(1)}\left[x^{(2)}((m_1, m_2), q_1; n)\right] + \mathcal{H}^{(2)}\left[x^{(2)}((m_1, m_2), q_1; n)\right] \tag{45}$$

- where

$$\mathcal{H}^{(2)}\left[x^{(2)}((m_1, m_2), q_1; n)\right] = v^{(2,2)}((m_1, m_2), q_1; n) \tag{46}$$

Novel Identification Algorithm - 2D

$$\begin{aligned}\mathcal{H}^{(1)}\left[x^{(2)}\big((m_1,m_2),q_1;n\big)\right] &= v^{(2,1)}\big((m_1,m_2),q_1;n\big) \\ &= v^{(1,1)}(m_1;n) + v^{(1,1)}(m_2;n-q_1)\end{aligned} \tag{47}$$

with

$$v^{(1,1)}(m_i;n) = \mathcal{H}^{(1)}\left[x^{(1)}(m_i;n)\right].$$

Novel Identification Algorithm - 2D

Here,

$$v^{(i,j)}\Big((m_1,\dots,m_i),q_1,\dots,q_{i-1};n\Big)=\mathcal{H}^{(j)}\Big[x^{(i)}\Big((m_1,\dots,m_i),q_1,\dots,q_{i-1};n\Big)\Big] \tag{48}$$

The output of the 2-D subsystem can be obtained from (45), (46) and (47) as

$$\begin{aligned}v^{(2,2)}\Big((m_1,m_2),q_1;n\Big)=\mathcal{N}\Big[&x^{(2)}\Big((m_1,m_2),q_1;n\Big)\Big]\\ &-\Big[v^{(1,1)}(m_1;n)+v^{(1,1)}(m_2;n-q_1)\Big]\end{aligned} \tag{49}$$

Novel Identification Algorithm - 2D

- The output of the 2-D subsystem is obtained by subtracting the appropriate previously computed 1-D outputs from the overall nonlinear system output.

Novel Identification Algorithm - 2D

- The output of the 2-D subsystem is obtained by subtracting the appropriate previously computed 1-D outputs from the overall nonlinear system output.
- We observe the outputs for $\binom{M}{2}$ distinct 2-D ensemble inputs which are given in the matrix form as follows:

$$\mathbf{x}_e^{(2)}(q_1; n) = \mathbf{T}_{2,1}^{(M)} \mathbf{x}_e^{(1)}(n) + \mathbf{T}_{2,2}^{(M)} \mathbf{x}_e^{(1)}(n - q_1) \tag{50}$$

where

$$\mathbf{x}_e^{(1)}(n) = \begin{bmatrix} a_1 & a_2 & \cdots & a_M \end{bmatrix}^T \delta(n)$$

Novel Identification Algorithm - 2D

- The constant $\mathbf{T}_{2,1}^{(M)}$ and $\mathbf{T}_{2,2}^{(M)}$ matrices with $\binom{M}{2}$ rows and $\binom{M}{1}$ columns are used to determine the necessary $\binom{M}{2}$ combinations of $\binom{M}{1} = M$ ensemble inputs when taken two at a time.

Novel Identification Algorithm - 2D

- The constant $\mathbf{T}_{2,1}^{(M)}$ and $\mathbf{T}_{2,2}^{(M)}$ matrices with $\binom{M}{2}$ rows and $\binom{M}{1}$ columns are used to determine the necessary $\binom{M}{2}$ combinations of $\binom{M}{1} = M$ ensemble inputs when taken two at a time.
- These matrices are called as the ensemble input formation matrices.

Novel Identification Algorithm - 2D

- Similar to the single-input single-output case in (45), using the 2-D ensemble input vector in (50), 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.

Novel Identification Algorithm - 2D

- Similar to the single-input single-output case in (45), using the 2-D ensemble input vector in (50), 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.
- The notation $\mathbf{v}_e^{(i,j)}(q_1, \ldots, q_{i-1}; n)$ will denote the output of the j-D subsystem for an i-D input ensemble.

Novel Identification Algorithm - 2D

- Similar to the single-input single-output case in (45), using the 2-D ensemble input vector in (50), 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.
- The notation $\mathbf{v}_e^{(i,j)}(q_1, \ldots, q_{i-1}; n)$ will denote the output of the j-D subsystem for an i-D input ensemble.

$$\mathbf{v}_e^{(i,j)}(q_1, \ldots, q_{i-1}; n) = \mathcal{H}^{(j)}\left[\mathbf{x}_e^{(i)}(q_1, \ldots, q_{i-1}; n)\right] \tag{51}$$

Novel Identification Algorithm - 2D

The response of the 1-D system to the 2-D input ensemble can be decomposed in terms of the 1-D responses as

$$\begin{aligned}\mathbf{v}_e^{(2,1)}(q_1;n) &= \mathcal{H}^{(1)}\left[\mathbf{T}_{2,1}^{(M)}\mathbf{x}_e^{(1)}(n)\right] + \mathcal{H}^{(1)}\left[\mathbf{T}_{2,2}^{(M)}\mathbf{x}_e^{(1)}(n-q_1)\right] \\ &= \mathbf{T}_{2,1}^{(M)}\mathbf{v}_e^{(1,1)}(n) + \mathbf{T}_{2,2}^{(M)}\mathbf{v}_e^{(1,1)}(n-q_1)\end{aligned} \tag{52}$$

Novel Identification Algorithm - 2D

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

$$\begin{aligned} \mathbf{v}_e^{(2,2)}(q_1;n) &= \mathbf{y}_e^{(2)}(q_1;n) - \mathbf{v}_e^{(2,1)}(q_1;n) \\ &= \mathbf{y}_e^{(2)}(q_1;n) - \left[\mathbf{T}_{2,1}^{(M)} \mathbf{v}_e^{(1,1)}(n) + \mathbf{T}_{2,2}^{(M)} \mathbf{v}_e^{(1,1)}(n-q_1) \right] \end{aligned} \tag{53}$$

Novel Identification Algorithm - 2D

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

$$\begin{aligned} \mathbf{v}_e^{(2,2)}(q_1;n) &= \sum_{i=0}^{N-q_1} \mathbf{U}_e^{(2)}(q_1;n-i)\mathbf{h}^{(2)}(q_1;i) \\ &= \mathbf{U}_e^{(2)}\mathbf{h}^{(2)}(q_1;n-q_1) \end{aligned} \tag{54}$$

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 $\binom{M}{2} \times \binom{M}{2}$, and it is written in terms of the amplitude levels, $a_1,\ a_2,\ldots,a_M$.

Novel Identification Algorithm - 2D

As an example, for $M = 3$ and $M = 4$, the $\mathbf{U}_e^{(2)}$ matrices will be given respectively as follows.

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

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

Novel Identification Algorithm - 2D

The 2-D Volterra kernel vectors are obtained as

$$\mathbf{h}^{(2)}(q_1; n - q_1) = \left[\mathbf{U}_e^{(2)}\right]^{-1} \mathbf{v}_e^{(2,2)}(q_1; n) \tag{55}$$

for $q_1 = 1, \ldots, N$ and $n = q_1, q_1 + 1, \ldots, N$, provided the inverse exists.
Fig. 7 depicts the identification method for 2-D kernels as outlined in this section and in (55).

Novel Identification Algorithm - 2D

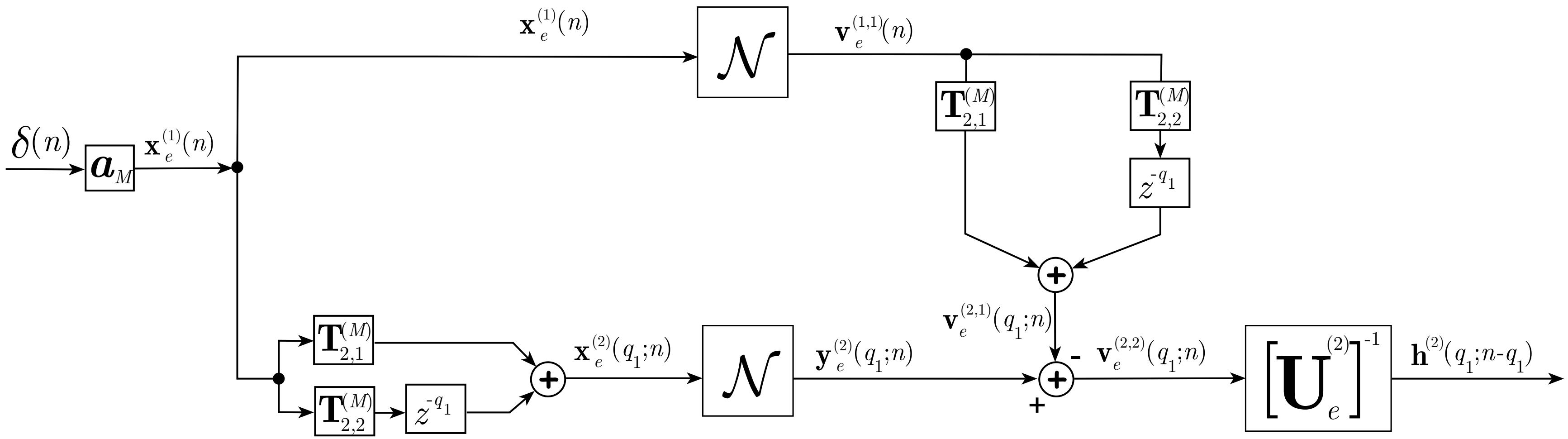


Figure 7: Method used for identification of 2-D Volterra kernels, $\mathbf{h}^{(2)}(q_1;n)$.

Novel Identification Algorithm - 3D

Novel Identification Algorithm - 3D

- The 3-D kernel vectors $\mathbf{h}^{(3)}(q_1, q_2; i)$ are determined by using the 3-D ensemble responses along with the responses of the 1-D and 2-D subsystems.

Novel Identification Algorithm - 3D

- The 3-D kernel vectors $\mathbf{h}^{(3)}(q_1, q_2; i)$ 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.

Novel Identification Algorithm - 3D

- The 3-D kernel vectors $\mathbf{h}^{(3)}(q_1, q_2; i)$ 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.

$$\mathbf{x}_e^{(3)}(q_1, q_2; n) = \mathbf{T}_{3,1}^{(M)}\mathbf{x}_e^{(1)}(n) + \mathbf{T}_{3,2}^{(M)}\mathbf{x}_e^{(1)}(n-q_2) + \mathbf{T}_{3,3}^{(M)}\mathbf{x}_e^{(1)}(n-q_1-q_2) \tag{56}$$

Novel Identification Algorithm - 3D

The $\mathbf{T}_{3,1}^{(M)}$, $\mathbf{T}_{3,2}^{(M)}$ and $\mathbf{T}_{3,3}^{(M)}$ input formation matrices with $\binom{M}{3}$ rows and $\binom{M}{1}$ columns are used to determine the necessary $\binom{M}{3}$ combinations of the multilevel impulse functions, when taking triplets at a time.

Novel Identification Algorithm - 3D

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

Novel Identification Algorithm - 3D

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

$$\mathbf{y}_e^{(3)}(q_1, q_2; n) = \mathcal{N}\left[\mathbf{x}_e^{(3)}(q_1, q_2; n)\right] \tag{57}$$

Novel Identification Algorithm - 3D

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

$$\begin{aligned} \mathbf{y}_e^{(3)}(q_1, q_2; n) =& \mathcal{N}\left[\mathbf{x}_e^{(3)}(q_1, q_2; n)\right] \\ =& \sum_{i=1}^{3} \mathcal{H}^{(i)}\left[\mathbf{x}_e^{(3)}(q_1, q_2; n)\right] \end{aligned} \tag{57}$$

Novel Identification Algorithm - 3D

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

$$\begin{aligned} \mathbf{y}_e^{(3)}(q_1, q_2; n) &= \mathcal{N}\left[\mathbf{x}_e^{(3)}(q_1, q_2; n)\right] \\ &= \sum_{i=1}^{3} \mathcal{H}^{(i)}\left[\mathbf{x}_e^{(3)}(q_1, q_2; n)\right] \\ &= \sum_{i=1}^{3} \mathbf{v}_e^{(3,i)}(q_1, q_2; n) \end{aligned} \tag{57}$$

Novel Identification Algorithm - 3D

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

$$\begin{aligned}
\mathbf{y}_e^{(3)}(q_1, q_2; n) =& \mathcal{N}\left[\mathbf{x}_e^{(3)}(q_1, q_2; n)\right] \\
=& \sum_{i=1}^{3} \mathcal{H}^{(i)}\left[\mathbf{x}_e^{(3)}(q_1, q_2; n)\right] \\
=& \sum_{i=1}^{3} \mathbf{v}_e^{(3,i)}(q_1, q_2; n)
\end{aligned} \tag{57}$$

The input-output relationship in (57) is depicted pictorially in Fig. 8.

Novel Identification Algorithm - 3D

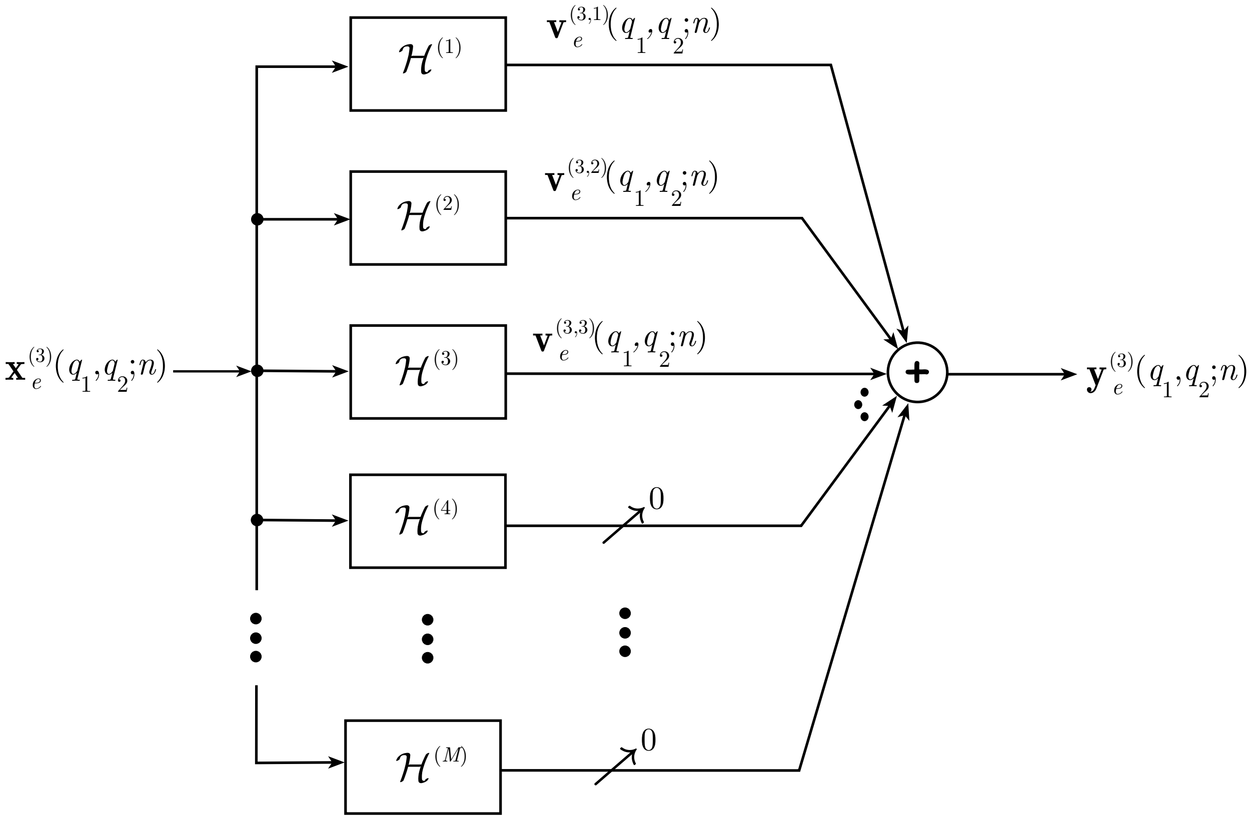


Figure 8: Pictorial description for (57). The output of the nonlinear system $\mathcal{N}$ for the 3-D input ensemble $\mathbf{x}_e^{(3)}(q_1, q_2; n)$, is equal to the sum of the outputs of subsystems $\mathcal{H}^{(1)}$, $\mathcal{H}^{(2)}$ and $\mathcal{H}^{(3)}$.

Novel Identification Algorithm - 3D

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

Novel Identification Algorithm - 3D

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

$$\mathbf{v}_e^{(3,1)}(q_1, q_2; n) = \sum_{j=1}^{\binom{3}{1}} \mathbf{S}_{31,j}^{(M)} \mathbf{v}_e^{(1,1)}(n - n_j^{(3,1)}) \quad (58)$$

Novel Identification Algorithm - 3D

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

$$\mathbf{v}_e^{(3,1)}(q_1, q_2; n) = \sum_{j=1}^{\binom{3}{1}} \mathbf{S}_{31,j}^{(M)} \mathbf{v}_e^{(1,1)}(n - n_j^{(3,1)}) \tag{58}$$

- The matrices $\mathbf{S}_{31,j}^{(M)}$ for $j = 1, 2, 3$ are used to pick up the appropriate 1-D ensemble output values.

Novel Identification Algorithm - 3D

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

$$\mathbf{v}_e^{(3,1)}(q_1, q_2; n) = \sum_{j=1}^{\binom{3}{1}} \mathbf{S}_{31,j}^{(M)} \mathbf{v}_e^{(1,1)}(n - n_j^{(3,1)}) \tag{58}$$

- The matrices $\mathbf{S}_{31,j}^{(M)}$ for $j = 1, 2, 3$ are used to pick up the appropriate 1-D ensemble output values.
- We call these matrices as the output pick-up matrices.

Novel Identification Algorithm - 3D

It is also possible to determine the responses of the 2-D subsystem for the 3-D ensemble inputs,

$$\mathbf{v}_e^{(3,2)}(q_1, q_2; n) = \sum_{j=1}^{\binom{3}{2}} \mathbf{S}_{32,j}^{(M)} \mathbf{v}_e^{(2,2)}(q_j^{(3,2)}; n - n_j^{(3,2)}) \tag{59}$$

where the output pick up matrices $\mathbf{S}_{32,j}^{(M)}$ for $j = 1, 2, 3$, which have $\binom{M}{3}$ rows and $\binom{M}{2}$ columns, are used to determine the appropriate 2-D ensemble output values for $\mathbf{x}_e^{(3)}(q_1, q_2; n)$.

Novel Identification Algorithm - 3D

The output of the 3-D subsystem $\mathbf{v}_e^{(3,3)}(q_1, q_2; n)$ can be written as

$$\mathbf{v}_e^{(3,3)}(q_1, q_2; n) = \mathbf{U}_e^{(3)}\mathbf{h}^{(3)}(q_1, q_2; n - q_1 - q_2) \tag{60}$$

The matrix $\mathbf{U}_e^{(3)}$ has the dimensions $\binom{M}{3} \times \binom{M}{3}$.

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

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

Novel Identification Algorithm - 3D

From (57)-(60), we get the desired calculation formula for the 3-D kernel vectors.

$$\mathbf{h}^{(3)}(q_1, q_2; n - q_1 - q_2) = \left[\mathbf{U}_e^{(3)}\right]^{-1} \mathbf{v}_e^{(3,3)}(q_1, q_2; n) \qquad (61)$$

where,

$$\mathbf{v}_e^{(3,3)}(q_1, q_2; n) = \mathbf{y}_e^{(3)}(q_1, q_2; n) - \left(\sum_{j=1}^{3} \mathbf{S}_{31,j}^{(M)} \mathbf{v}_e^{(1,1)}(n - n_j^{(3,1)}) + \sum_{j=1}^{3} \mathbf{S}_{32,j}^{(M)} \mathbf{v}_e^{(2,2)}(q_j^{(3,2)}; n - n_j^{(3,2)})\right) \qquad (62)$$

Novel Identification Algorithm - 3D

Fig. 9 depicts the identification method for 3-D kernels as described by (61) and (62).

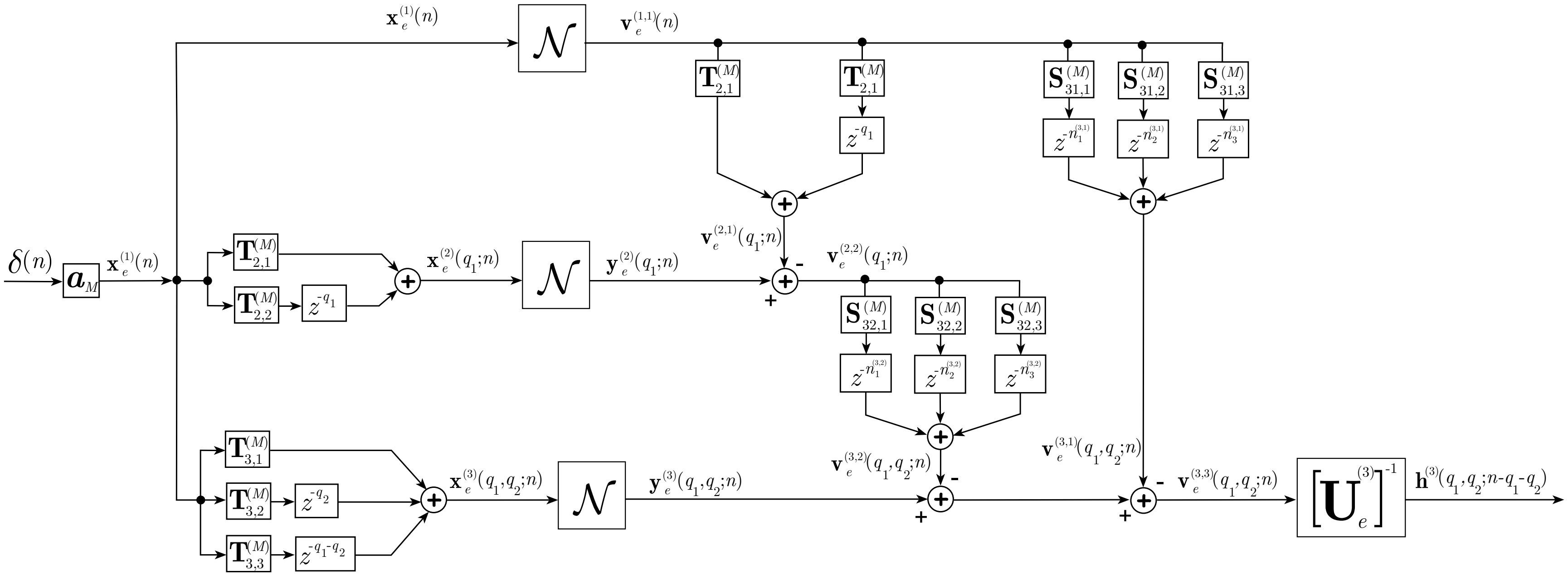


Figure 9: Method used for identification of 3-D Volterra kernels, $\mathbf{h}^{(3)}(q_1, q_2; n)$.

Novel Identification Algorithm - $\ell-$D

Novel Identification Algorithm - ℓ—D

- 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, $\mathbf{v}_e^{(k,k)}(q_1, \ldots, q_{k-1}; n)$.

Novel Identification Algorithm - ℓ—D

- 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, $\mathbf{v}_e^{(k,k)}(q_1, \ldots, q_{k-1}; n)$.
- Similar to 1-, 2-, and 3-D ensemble input vectors, the ℓ-D input ensemble vector can be written using the input formation matrices and the 1-D input ensemble.

Novel Identification Algorithm - $\ell-$D

- 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, $\mathbf{v}_e^{(k,k)}(q_1, \ldots, q_{k-1}; n)$.
- Similar to 1-, 2-, and 3-D ensemble input vectors, the ℓ-D input ensemble vector can be written using the input formation matrices and the 1-D input ensemble.

$$\mathbf{x}_e^{(\ell)}(q_1, \ldots, q_{\ell-1}; n) = \sum_{i=1}^{\ell} \mathbf{T}_{\ell,i}^{(M)} \mathbf{x}_e^{(1)}(n - n_i^{(\ell)}) \tag{63}$$

Novel Identification Algorithm - ℓ—D

- The response of the nonlinear system to the ensemble input in (63) can be written in terms of the outputs of the subsystems,

(64)

Novel Identification Algorithm - $\ell-$D

- The response of the nonlinear system to the ensemble input in (63) can be written in terms of the outputs of the subsystems,

$$\mathbf{y}_e^{(\ell)}(q_1, \ldots, q_{\ell-1}; n) = \mathcal{N}\left[\mathbf{x}_e^{(\ell)}(q_1, \ldots, q_{\ell-1}; n)\right] \tag{64}$$

Novel Identification Algorithm - ℓ—D

- The response of the nonlinear system to the ensemble input in (63) can be written in terms of the outputs of the subsystems,

$$\begin{aligned}
\mathbf{y}_e^{(\ell)}(q_1, \ldots, q_{\ell-1}; n) =& \mathcal{N}\left[\mathbf{x}_e^{(\ell)}(q_1, \ldots, q_{\ell-1}; n)\right] \\
=& \sum_{k=1}^{\ell} \mathcal{H}^{(k)}\left[\mathbf{x}_e^{(\ell)}(q_1, \ldots, q_{\ell-1}; n)\right]
\end{aligned} \tag{64}$$

Novel Identification Algorithm - ℓ—D

- The response of the nonlinear system to the ensemble input in (63) can be written in terms of the outputs of the subsystems,

$$\begin{aligned}
\mathbf{y}_e^{(\ell)}(q_1,\ldots,q_{\ell-1};n) &= \mathcal{N}\left[\mathbf{x}_e^{(\ell)}(q_1,\ldots,q_{\ell-1};n)\right] \\
&= \sum_{k=1}^{\ell}\mathcal{H}^{(k)}\left[\mathbf{x}_e^{(\ell)}(q_1,\ldots,q_{\ell-1};n)\right] \\
&= \sum_{k=1}^{\ell}\mathbf{v}_e^{(\ell,k)}(q_1,\ldots,q_{\ell-1};n)
\end{aligned} \tag{64}$$

Novel Identification Algorithm - ℓ—D

- The response of the nonlinear system to the ensemble input in (63) can be written in terms of the outputs of the subsystems,

$$\begin{aligned}
\mathbf{y}_e^{(\ell)}(q_1, \ldots, q_{\ell-1}; n) =& \mathcal{N}\left[\mathbf{x}_e^{(\ell)}(q_1, \ldots, q_{\ell-1}; n)\right] \\
=& \sum_{k=1}^{\ell} \mathcal{H}^{(k)}\left[\mathbf{x}_e^{(\ell)}(q_1, \ldots, q_{\ell-1}; n)\right] \\
=& \sum_{k=1}^{\ell} \mathbf{v}_e^{(\ell,k)}(q_1, \ldots, q_{\ell-1}; n)
\end{aligned} \tag{64}$$

- Fig. 10 draws a picture of (64), by showing that for an ℓ-D input ensemble, the outputs of all subsystems $\mathcal{H}^{(k)}$, for $k > \ell$, are equal to zero.

Novel Identification Algorithm - ℓ—D

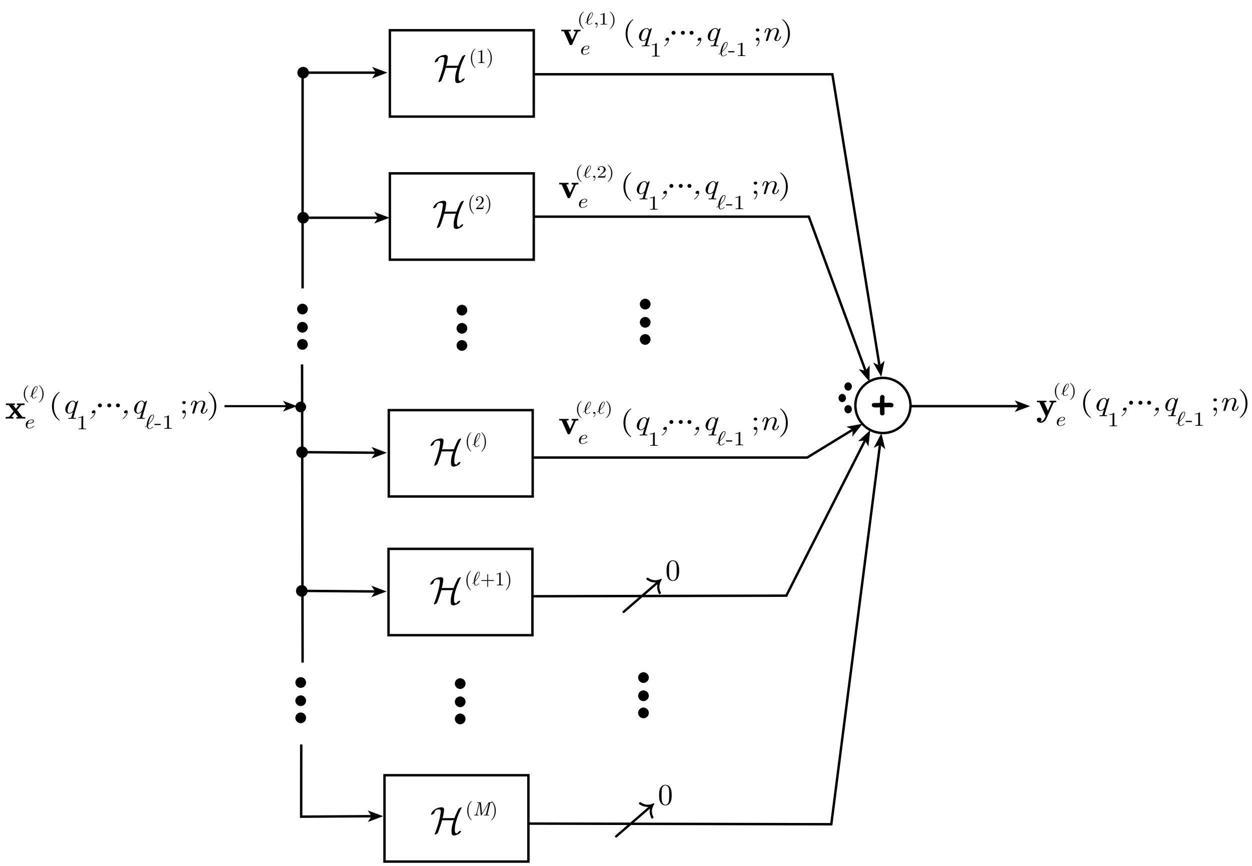


Figure 10: Pictorial description for (64). The output of the nonlinear system $\mathcal{N}$ for the ℓ-D input ensemble $\mathbf{x}_e^{(\ell)}(q_1,\ldots,q_{\ell-1};n)$, is equal to the sum of the outputs of subsystems $\mathcal{H}^{(1)}$ through $\mathcal{H}^{(\ell)}$.

Novel Identification Algorithm - ℓ—D

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

$$
\begin{aligned}
\mathbf{v}_e^{(\ell,1)}(q_1, \ldots, q_{\ell-1}; n) &= \sum_{j=1}^{\binom{\ell}{1}} \mathbf{S}_{\ell 1,j}^{(M)} \mathbf{v}_e^{(1,1)}(n - n_j^{(\ell,1)}) \\
&\vdots \\
\mathbf{v}^{(\ell,k)}(q_1, \ldots, q_{\ell-1}; n) &= \sum_{j=1}^{\binom{\ell}{k}} \mathbf{S}_{\ell 1,j}^{(M)} \mathbf{v}_e^{(k,k)}(q_{j,1}^{(\ell,k)}, \ldots, q_{j,k-1}^{(\ell,k)}; n - n_j^{(\ell,k)})
\end{aligned}
\tag{65}
$$

for $k = 2, 3, \ldots, \ell - 1$.

Novel Identification Algorithm - ℓ—D

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

$$\mathbf{v}_e^{(\ell,\ell)}(q_1,\ldots,q_{\ell-1};n) = \mathcal{H}^{(\ell)}\left[\mathbf{x}_e^{(\ell)}(q_1,\ldots,q_{\ell-1};n)\right]$$
$$= \sum_{i=0}^{N-\bar{q}_{\ell-1}} \mathbf{U}_e^{(\ell)}(q_1,\ldots,q_{\ell-1};n-i)\mathbf{h}^{(\ell)}(q_1,\ldots,q_{\ell-1};i) \quad (66)$$

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 $\binom{M}{\ell}\times\binom{M}{\ell}$ and can be written in terms of the amplitudes $a_1,\ a_2,\ldots,a_M$.

Novel Identification Algorithm - ℓ—D

As an example, for $M = 5$ and $\ell = 4$, the matrix $\mathbf{U}_e^{(4)}$ will be given as,

$$\mathbf{U}_e^{(4)} = \begin{bmatrix} a_1a_2a_3a_4 & a_1a_2a_3a_4^2 & a_1a_2a_3^2a_4 & a_1a_2^2a_3a_4 & a_1^2a_2a_3a_4 \\ a_1a_2a_3a_5 & a_1a_2a_3a_5^2 & a_1a_2a_3^2a_5 & a_1a_2^2a_3a_5 & a_1^2a_2a_3a_5 \\ a_1a_2a_4a_5 & a_1a_2a_4a_5^2 & a_1a_2a_4^2a_5 & a_1a_2^2a_4a_5 & a_1^2a_2a_4a_5 \\ a_1a_3a_4a_5 & a_1a_3a_4a_5^2 & a_1a_3a_4^2a_5 & a_1a_3^2a_4a_5 & a_1^2a_3a_4a_5 \\ a_2a_3a_4a_5 & a_2a_3a_4a_5^2 & a_2a_3a_4^2a_5 & a_2a_3^2a_4a_5 & a_2^2a_3a_4a_5 \end{bmatrix}$$

Novel Identification Algorithm - ℓ—D

The ℓ-D Volterra kernel vectors can be written in the following form.

$$\mathbf{h}^{(\ell)}(q_1, \ldots, q_{\ell-1}; n - \bar{q}_{\ell-1}) = \left[\mathbf{U}_e^{(\ell)}\right]^{-1} \mathbf{v}_e^{(\ell,\ell)}(q_1, \ldots, q_{\ell-1}; n) \quad \text{(67a)}$$

Here,

$$\mathbf{v}_e^{(\ell,\ell)}(q_1, \ldots, q_{\ell-1}; n) = \mathbf{y}_e^{(\ell)}(q_1, \ldots, q_{\ell-1}; n) - \sum_{j=1}^{\binom{\ell}{1}} \mathbf{S}_{\ell 1,j}^{(M)} \mathbf{v}_e^{(1,1)}(n - n_j^{(\ell,1)})$$

$$- \sum_{k=2}^{\ell-1} \sum_{j=1}^{\binom{\ell}{k}} \mathbf{S}_{\ell k,j}^{(M)} \mathbf{v}_e^{(k,k)}(q_{j,1}^{(\ell,k)}, \ldots, q_{j,k-1}^{(\ell,k)}; n - n_j^{(\ell,k)}) \quad \text{(67b)}$$

Novel Identification Algorithm - ℓ—D

(67) shows that our algorithm can form the estimate for any Volterra kernel independent from other kernels.
Let us define the following output pick-up operators for $k = 1, 2, \ldots, \ell - 1$:

$$\mathbb{S}_{\ell,k}^{(M)}\left[\mathbf{v}_e^{(k,k)}(q_1, \ldots, q_{k-1}; n)\right] = \sum_{j=1}^{\binom{\ell}{k}} \mathbf{S}_{\ell k,j}^{(M)} \mathbf{v}_e^{(k,k)}(q_{j,1}^{(\ell,k)}, \ldots, q_{j,k-1}^{(\ell,k)}; n - n_j^{(\ell,k)}) \tag{68}$$

Novel Identification Algorithm - $\ell-$D

With this definition, (67b) can be rewritten in a more compact form.

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

Fig. 11 depicts the identification of the Volterra kernels of orders one through M using the proposed algorithm. In this figure the output pick-up operators $\mathbb{S}_{\ell,k}^{(M)}[\,\cdot\,]$ are utilized to simplify the picture.

Novel Identification Algorithm - ℓ—D

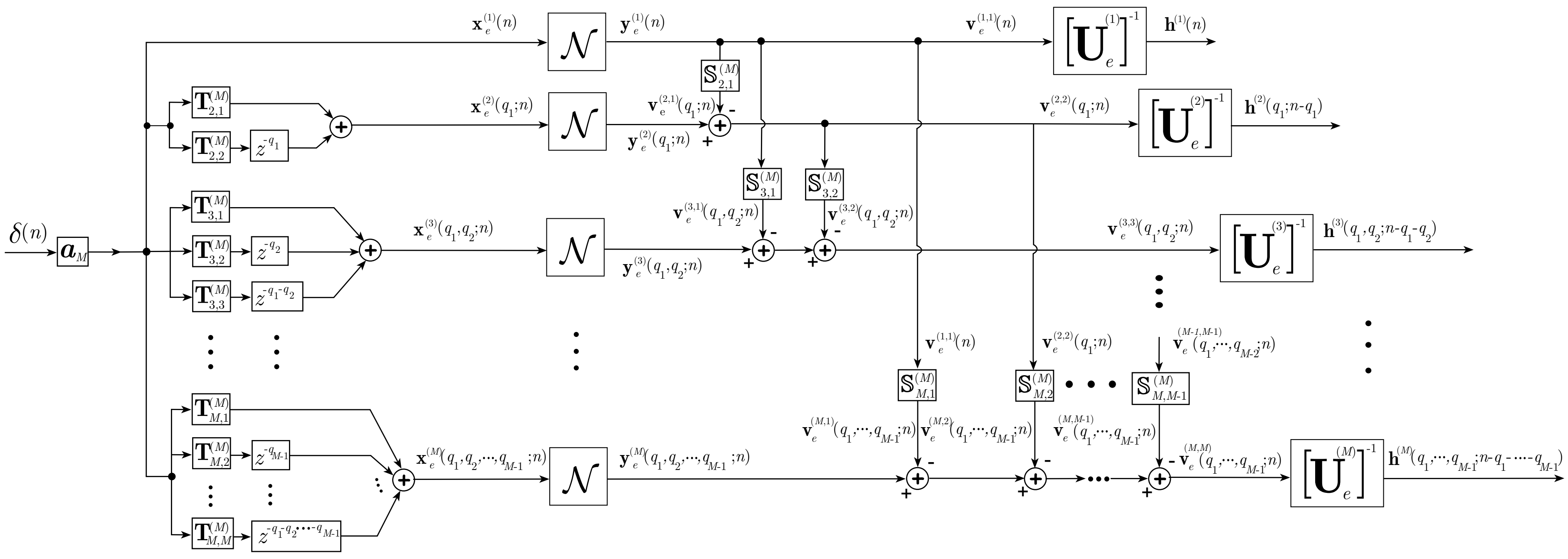


Figure 11: Proposed Volterra kernel identification method using multilevel deterministic sequences as inputs.

Deterministic Sequence Example

Deterministic Sequence Example

- We consider as an example the identification of a Volterra filter with $M = 3$ and $N = 2$.

Deterministic Sequence Example

- We consider as an example the identification of a Volterra filter with $M = 3$ and $N = 2$.
- The overall deterministic input sequence which should be applied to identify the kernels of this system is shown in Fig. 12.

Deterministic Sequence Example

- We consider as an example the identification of a Volterra filter with $M = 3$ and $N = 2$.
- The overall deterministic input sequence which should be applied to identify the kernels of this system is shown in Fig. 12.
- This figure depicts all the input ensembles utilized for the identification of the 1-D, 2-D and 3-D nonlinear subsystems.

Deterministic Sequence Example

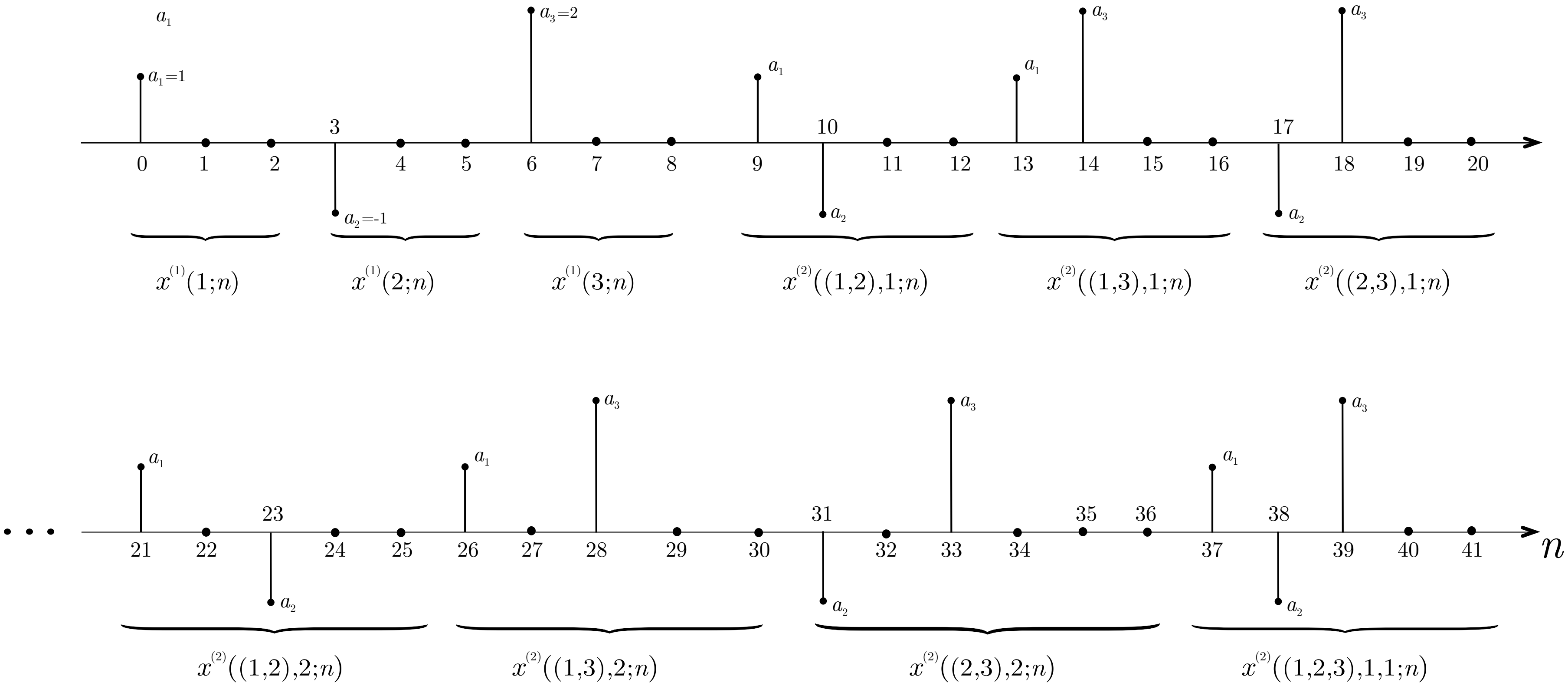


Figure 12: Deterministic multilevel input sequence

Identification Simulations

We present two numerical simulations to illustrate the performance of our novel identification procedure, where the setup for the examples are taken from Zhou and Giannakis (1997) and Nowak and Van Veen (1994b).

Simulation 1: We simulate a second order Volterra filter with memory length $N = 2$.

$$y(n) = \sum_{i_1=0}^{2} b_1(i_1)x(n-i_1) + \sum_{i_1=0}^{2}\sum_{i_2=i_1}^{2} b_2(i_1,i_2)x(n-i_1)x(n-i_2) \quad (70)$$

The average input power is unity for both PRMS and our multilevel sequence. Independent GWN of power 0.1 is added to the system output to represent observation noise.

Identification Simulations

In Table 104 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. The squared kernel error averaged over independent trials is defined as

$$error = \sum_{i_1=0}^{N-1} \left[b_1(i_1) - \hat{b}_1(i_1; n) \right]^2 + \sum_{i_1=0}^{N-1} \sum_{i_2=i_1}^{N-1} \left[b_2(i_1, i_2) - \hat{b}_2(i_1, i_2; n) \right]^2$$

From the results in Table 104, it is clear that our algorithm uses less operations and gives better results than the PRMS method (Nowak and Van Veen, 1994b).

Identification Simulations

Averaged Squared Error of Estimates for Simulation 1

PRMS of Nowak and Van Veen (1994b)			proposed deterministic input sequence		
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}

Identification Simulations

Simulation 2: We simulate a fourth-order Volterra filter with memory length $N = 3$ after the example 2 in Zhou and Giannakis (1997).

$$y(n) = x(n)^4 + 4.8x(n)^3x(n-1) + 4.8x(n)^2x(n-1)^2$$
$$- 6x(n)^2x(n-1)x(n-2) + 14.4x(n)x(n-1)x(n-2)x(n-3)$$

Average input power is set to 4 and additive independent AGWN 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.

Identification Simulations

Table 107 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 Zhou and Giannakis (1997).

There are five nonzero Volterra kernels, and the values are taken from (Zhou and Giannakis, 1997). 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.

Identification Simulations

Results for Simulation 4.2

(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	4.8000	4.8000	-6.0000	14.4000
mean of $\hat{b}_4\,(i_1, i_2, i_3, i_4)$ for PSK	0.9944	4.7927	4.8006	-6.0084	14.3892
std of $\hat{b}_4\,(i_1, i_2, i_3, i_4)$ for PSK	0.2013	0.1940	0.1745	0.1766	0.1004
mean of $\hat{b}_4\,(i_1, i_2, i_3, i_4)$ for our alg.	1.0002	4.7999	4.8000	-6.0003	14.4001
std of $\hat{b}_4\,(i_1, i_2, i_3, i_4)$ for our alg.	0.0024	0.0042	0.0021	0.0038	0.0069

Persistence of Excitation

Persistence of Excitation

- We will prove that the multilevel input signals as persistently excite a Volterra filter.

Persistence of Excitation

- We will prove that the multilevel input signals as persistently excite a Volterra filter.
- We start by rewriting the input-output relation of the ℓ-D cross-term subsystem, $\mathcal{H}^{(\ell)}$.

Persistence of Excitation

- We will prove that the multilevel input signals as persistently excite a Volterra filter.
- We start by rewriting the input-output relation of the ℓ-D cross-term subsystem, $\mathcal{H}^{(\ell)}$.

$$y^{(\ell)}(n) = \mathcal{H}^{(\ell)}\left[x(n)\right] = \sum_{p=1}^{\binom{N}{\ell-1}} \sum_{i=0}^{N-\mathrm{sum}(\boldsymbol{q}_p)} \mathbf{h}^{(\ell)^T}(\boldsymbol{q}_p; i)\, \mathbf{x}^{(\ell)}(\boldsymbol{q}_p; n-i), \tag{71}$$

Persistence of Excitation

- We will prove that the multilevel input signals as persistently excite a Volterra filter.
- We start by rewriting the input-output relation of the ℓ-D cross-term subsystem, $\mathcal{H}^{(\ell)}$.

$$y^{(\ell)}(n) = \mathcal{H}^{(\ell)}\left[x(n)\right] = \sum_{p=1}^{\binom{N}{\ell-1}} \sum_{i=0}^{N-\mathrm{sum}(\boldsymbol{q}_p)} \mathbf{h}^{(\ell)^T}(\boldsymbol{q}_p; i)\, \mathbf{x}^{(\ell)}(\boldsymbol{q}_p; n-i), \tag{71}$$

- We can reformulate the output in (71)

Persistence of Excitation

- We will prove that the multilevel input signals as persistently excite a Volterra filter.
- We start by rewriting the input-output relation of the ℓ-D cross-term subsystem, $\mathcal{H}^{(\ell)}$.

$$y^{(\ell)}(n) = \mathcal{H}^{(\ell)}\left[x(n)\right] = \sum_{p=1}^{\binom{N}{\ell-1}} \sum_{i=0}^{N-\mathrm{sum}(\boldsymbol{q}_p)} \mathbf{h}^{(\ell)^T}(\boldsymbol{q}_p; i)\, \mathbf{x}^{(\ell)}(\boldsymbol{q}_p; n-i), \tag{71}$$

- We can reformulate the output in (71)

$$y^{(\ell)}(n) = \mathcal{H}^{(\ell)}\left[x(n)\right] = \sum_{p=1}^{\binom{N}{\ell-1}} \mathbf{h}^{(\ell)^T}(\boldsymbol{q}_p)\mathbf{x}_n^{(\ell)}(\boldsymbol{q}_p) \tag{72}$$

Persistence of Excitation

$\mathbf{h}^{(\ell)}(\boldsymbol{q}_p)$ 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 $\mathbf{x}^{(\ell)}(\boldsymbol{q}_p; n-i)$.

$$\mathbf{h}^{(\ell)}(\boldsymbol{q}_p) = \begin{bmatrix} \mathbf{h}^{(\ell)}(\boldsymbol{q}_p; 0) \\ \mathbf{h}^{(\ell)}(\boldsymbol{q}_p; 1) \\ \vdots \\ \mathbf{h}^{(\ell)}\left(\boldsymbol{q}_p; N - \mathrm{sum}(\boldsymbol{q}_p)\right) \end{bmatrix} \qquad \mathbf{x}_n^{(\ell)}(\boldsymbol{q}_p) = \begin{bmatrix} \mathbf{x}^{(\ell)}(\boldsymbol{q}_p; n) \\ \mathbf{x}^{(\ell)}(\boldsymbol{q}_p; n-1) \\ \vdots \\ \mathbf{x}^{(\ell)}\Big(\boldsymbol{q}_p; n - \big(N - \mathrm{sum}(\boldsymbol{q}_p)\big)\Big) \end{bmatrix} \tag{73}$$

Persistence of Excitation

We can rewrite the linear combination in (72) as a single vector product,

$$y^{(\ell)}(n) = \mathcal{H}^{(\ell)}\left[x(n)\right] = \mathbf{h}^{(\ell)^T}\mathbf{x}^{(\ell)}(n) \tag{74}$$

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.

$$\mathbf{h}^{(\ell)} = \begin{bmatrix} \mathbf{h}^{(\ell)}(\boldsymbol{q}_1) \\ \mathbf{h}^{(\ell)}(\boldsymbol{q}_2) \\ \vdots \\ \mathbf{h}^{(\ell)}\left(\boldsymbol{q}_{\binom{N}{\ell-1}}\right) \end{bmatrix} \qquad \mathbf{x}^{(\ell)}(n) = \begin{bmatrix} \mathbf{x}_n^{(\ell)}(\boldsymbol{q}_1) \\ \mathbf{x}_n^{(\ell)}(\boldsymbol{q}_2) \\ \vdots \\ \mathbf{x}_n^{(\ell)}\left(\boldsymbol{q}_{\binom{N}{\ell-1}}\right) \end{bmatrix} \tag{75}$$

Persistence of Excitation

The length of the vector $\mathbf{h}^{(\ell)}$ is $\binom{M}{\ell}\binom{N+1}{\ell}$.
Suppose we begin observing the output of the ℓ-D subsystem at some time n and collect data over an observation period $\tau > 0$. The output for times n through $n+\tau$ can be written as a single vector.

$$\mathbf{y}^{(\ell)}(n) = \left[y^{(\ell)}(n) \quad y^{(\ell)}(n+1) \quad \cdots \quad y^{(\ell)}(n+\tau)\right]^H$$

The output vector is related to the input by

$$\mathbf{y}^{(\ell)}(n) = \mathbf{X}^{(\ell)}(n)\mathbf{h}^{(\ell)^*} \tag{76}$$

Persistence of Excitation

$\mathbf{X}^{(\ell)}(n)$ is the data matrix as defined below.

$$\mathbf{X}^{(\ell)}(n) = \begin{bmatrix} \mathbf{x}^{(\ell)}(n)^H \\ \mathbf{x}^{(\ell)}(n+1)^H \\ \vdots \\ \mathbf{x}^{(\ell)}(n+\tau)^H \end{bmatrix} \tag{77}$$

$(\,\cdot\,)^H$ denotes the Hermitian transpose, and $(\,\cdot\,)^*$ denotes complex conjugation.
(76) defines the input-output relation of the ℓ-D subsystem $\mathcal{H}^{(\ell)}$ as a single matrix product.

Persistence of Excitation

- This equation is a pseudo-linear regression equation, since although it is linear with respect to the kernels, the expanded input matrix $\mathbf{X}^{(\ell)}(n)$ consists of nonlinear products of $x(n)$.

Persistence of Excitation

- This equation is a pseudo-linear regression equation, since although it is linear with respect to the kernels, the expanded input matrix $\mathbf{X}^{(\ell)}(n)$ consists of nonlinear products of $x(n)$.
- For this pseudo-linear regression problem we can formulate the least-squares solution. The optimal least-squares solution for the kernel vector in (76) can be written as below.

Persistence of Excitation

- This equation is a pseudo-linear regression equation, since although it is linear with respect to the kernels, the expanded input matrix $\mathbf{X}^{(\ell)}(n)$ consists of nonlinear products of $x(n)$.
- For this pseudo-linear regression problem we can formulate the least-squares solution. The optimal least-squares solution for the kernel vector in (76) can be written as below.

$$\hat{\mathbf{h}}^{(\ell)^*} = \hat{\boldsymbol{\mathcal{R}}}^{(\ell)^{-1}} \hat{\mathbf{d}} \qquad (78)$$

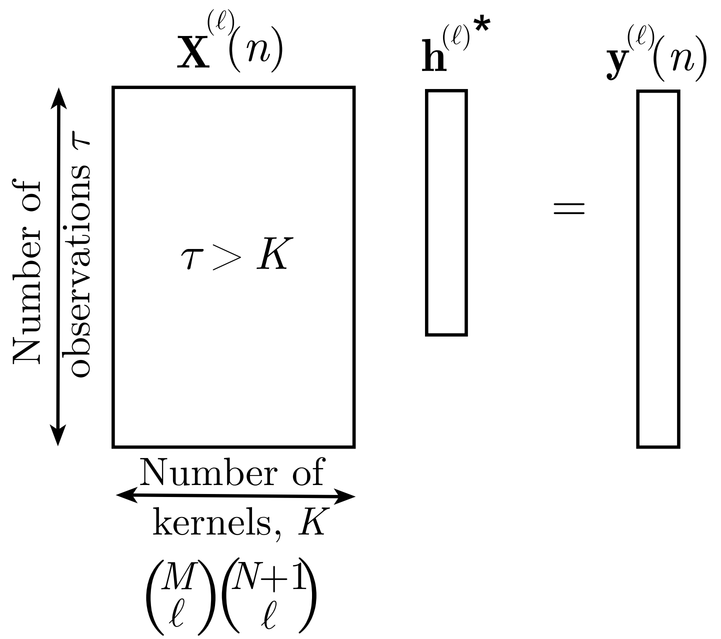
Persistence of Excitation

- This equation is a pseudo-linear regression equation, since although it is linear with respect to the kernels, the expanded input matrix $\mathbf{X}^{(\ell)}(n)$ consists of nonlinear products of $x(n)$.
- For this pseudo-linear regression problem we can formulate the least-squares solution. The optimal least-squares solution for the kernel vector in (76) can be written as below.

$$\hat{\mathbf{h}}^{(\ell)^*} = \hat{\boldsymbol{\mathcal{R}}}^{(\ell)^{-1}} \hat{\mathbf{d}} \qquad (78)$$

- (76) and (78) are portrayed in Fig. 13.

Persistence of Excitation



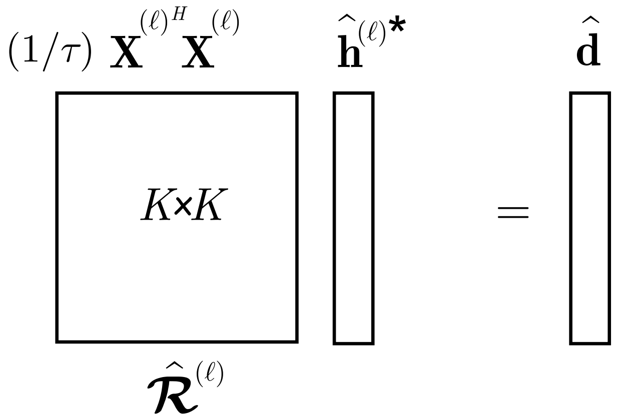


Figure 13: Graphical description for (76) and (78).

Persistence of Excitation

In (78), $\hat{\mathcal{R}}^{(\ell)}$ is the time-averaged autocorrelation matrix for the input, and $\hat{\mathbf{d}}$ is the time-averaged cross-correlation vector between the input and the output. They are defined as given below.

$$\hat{\mathcal{R}}^{(\ell)} = \frac{1}{\tau} \mathbf{X}^{(\ell)^H}(n)\, \mathbf{X}^{(\ell)}(n) = \frac{1}{\tau} \sum_{m=n}^{n+\tau} \mathbf{x}^{(\ell)}(m)\, \mathbf{x}^{(\ell)^H}(m) \tag{79}$$

$$\hat{\mathbf{d}} = \frac{1}{\tau} \mathbf{X}^{(\ell)^H}(n)\, \mathbf{y}^{(\ell)}(n) = \frac{1}{\tau} \sum_{m=n}^{n+\tau} \mathbf{x}^{(\ell)}(m)\, y^{*}(m) \tag{80}$$

Persistence of Excitation

- The least squares solution as formulated in (78) and (**??**) has a unique solution if the autocorrelation matrix $\hat{\mathcal{R}}^{(\ell)}$ is invertible. This brings us to the definition of persistence of excitation.

Persistence of Excitation

- The least squares solution as formulated in (78) and (**??**) has a unique solution if the autocorrelation matrix $\hat{\mathcal{R}}^{(\ell)}$ is invertible. This brings us to the definition of persistence of excitation.
- Basically signals, for which the time-average autocorrelation matrix is nonsingular for all times, are said to persistently excite a system.

Persistence of Excitation

- The least squares solution as formulated in (78) and (**??**) has a unique solution if the autocorrelation matrix $\hat{\mathcal{R}}^{(\ell)}$ is invertible. This brings us to the definition of persistence of excitation.
- Basically signals, for which the time-average autocorrelation matrix is nonsingular for all times, are said to persistently excite a system.
- We formulate the persistence of excitation condition for the ℓ-D cross-term subsystem $\mathcal{H}^{(\ell)}$.

Persistence of Excitation

- **Definition: Persistence of Excitation for $\mathcal{H}^{(\ell)}$**
  Let $\tau > 0$ be a fixed observation period of choice. Let $\lambda_{\max}$ and $\lambda_{\min}$ denote the largest and smallest eigenvalues of the time-average correlation matrix $\hat{\boldsymbol{\mathcal{R}}}^{(\ell)}$, as defined in (79). If there exist positive constants $\rho_{1,}\rho_2 > 0$ such that for every time instant k

$$\rho_1 \leqslant \lambda_{\min} \leqslant \lambda_{\max} \leqslant \rho_2 \tag{81}$$

  then the input signal $x(n)$ is said to be persistently exciting (PE) for the ℓ-D cross-term subsystem $\mathcal{H}^{(\ell)}$.

Persistence of Excitation

- It can be shown that the time-average correlation matrix, $\hat{\boldsymbol{\mathcal{R}}}^{(\ell)}$ for our deterministic ℓ-D ensemble input signals $\mathbf{x}_e^{(\ell)}$ becomes a block diagonal matrix.

Persistence of Excitation

- It can be shown that the time-average correlation matrix, $\hat{\boldsymbol{\mathcal{R}}}^{(\ell)}$ for our deterministic ℓ-D ensemble input signals $\mathbf{x}_e^{(\ell)}$ becomes a block diagonal matrix.

$$\hat{\boldsymbol{\mathcal{R}}}^{(\ell)} = \frac{1}{\tau}\begin{bmatrix} \left[\hat{\mathbf{R}}^{(\ell)}\right] & \left[\mathbf{0}_{\binom{M}{\ell}\times\binom{M}{\ell}}\right] & \cdots & \left[\mathbf{0}_{\binom{M}{\ell}\times\binom{M}{\ell}}\right] \\ \left[\mathbf{0}_{\binom{M}{\ell}\times\binom{M}{\ell}}\right] & \left[\hat{\mathbf{R}}^{(\ell)}\right] & \ddots & \vdots \\ \vdots & \ddots & \ddots & \left[\mathbf{0}_{\binom{M}{\ell}\times\binom{M}{\ell}}\right] \\ \left[\mathbf{0}_{\binom{M}{\ell}\times\binom{M}{\ell}}\right] & \cdots & \left[\mathbf{0}_{\binom{M}{\ell}\times\binom{M}{\ell}}\right] & \left[\hat{\mathbf{R}}^{(\ell)}\right] \end{bmatrix} \tag{82}$$

Persistence of Excitation

- $\hat{\mathbf{R}}^{(\ell)}$ in (82) has a size of $\binom{M}{\ell} \times \binom{M}{\ell}$.

Persistence of Excitation

- $\hat{\mathbf{R}}^{(\ell)}$ in (82) has a size of $\binom{M}{\ell} \times \binom{M}{\ell}$.
- $\hat{\mathbf{R}}^{(\ell)}$ can be written as

$$\hat{\mathbf{R}}^{(\ell)} = \left[\mathbf{U}_e^{(\ell)H} \mathbf{U}_e^{(\ell)} \right]^T \tag{83}$$

where $\mathbf{U}_e^{(\ell)}$ is the ℓ-D ensemble input matrix.

Persistence of Excitation

- $\hat{\mathbf{R}}^{(\ell)}$ in (82) has a size of $\binom{M}{\ell} \times \binom{M}{\ell}$.
- $\hat{\mathbf{R}}^{(\ell)}$ can be written as

$$\hat{\mathbf{R}}^{(\ell)} = \left[\mathbf{U}_e^{(\ell)^H} \mathbf{U}_e^{(\ell)} \right]^T \tag{83}$$

  where $\mathbf{U}_e^{(\ell)}$ is the ℓ-D ensemble input matrix.
- Hence, for our deterministic input ensemble the eigenvalues of the sample correlation matrix $\hat{\boldsymbol{\mathcal{R}}}^{(\ell)} = \frac{1}{\tau} \mathbf{X}^{(\ell)^H}(n) \mathbf{X}^{(\ell)}(n)$ are equal to the eigenvalues of the matrix $\hat{\mathbf{R}}^{(\ell)} = \left[\mathbf{U}_e^{(\ell)^H} \mathbf{U}_e^{(\ell)} \right]^T$.

Persistence of Excitation

- A matrix is said to be positive definite if all of its eigenvalues are positive. Under this definition, positive definiteness of $\hat{\mathbf{R}}^{(\ell)}$ is necessary and sufficient for the persistence of excitation condition given in (81).

Persistence of Excitation

- A matrix is said to be positive definite if all of its eigenvalues are positive. Under this definition, positive definiteness of $\hat{\mathbf{R}}^{(\ell)}$ is necessary and sufficient for the persistence of excitation condition given in (81).
- The Hermitian matrix $\hat{\mathbf{R}}^{(\ell)} = \left[{\mathbf{U}_e^{(\ell)}}^H \mathbf{U}_e^{(\ell)} \right]^T$ will be a positive definite matrix if and only if the square matrix $\mathbf{U}_e^{(\ell)}$ is nonsingular.

Persistence of Excitation

- A matrix is said to be positive definite if all of its eigenvalues are positive. Under this definition, positive definiteness of $\hat{\mathbf{R}}^{(\ell)}$ is necessary and sufficient for the persistence of excitation condition given in (81).
- The Hermitian matrix $\hat{\mathbf{R}}^{(\ell)} = \left[{\mathbf{U}_e^{(\ell)}}^H \mathbf{U}_e^{(\ell)} \right]^T$ will be a positive definite matrix if and only if the square matrix $\mathbf{U}_e^{(\ell)}$ is nonsingular.
- Therefore, the ℓ-D input sequence is PE for the subsystem $\mathcal{H}^{(\ell)}$ if and only if the input ensemble matrix $\mathbf{U}_e^{(\ell)}$ is nonsingular.

Persistence of Excitation

- 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.

Persistence of Excitation

- 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.

- The input ensemble is assured to be PE when we choose distinct and nonzero amplitude levels $a_1, a_2, \ldots, a_M$. $\square$

Least Squares Solution

- In the case of a general nonspecific input sequence, the least-squares solution, if it exists, requires the calculation of the inverse of an autocorrelation matrix $\hat{\boldsymbol{\mathcal{R}}}^{(\ell)}$, of size $\binom{M}{\ell}\binom{N+1}{\ell} \times \binom{M}{\ell}\binom{N+1}{\ell}$.

Least Squares Solution

- In the case of a general nonspecific input sequence, the least-squares solution, if it exists, requires the calculation of the inverse of an autocorrelation matrix $\hat{\mathcal{R}}^{(\ell)}$, of size $\binom{M}{\ell}\binom{N+1}{\ell} \times \binom{M}{\ell}\binom{N+1}{\ell}$.
- However, for our specific ℓ-D input sequences, the autocorrelation matrix greatly simplifies and attains the very sparse form of a block diagonal matrix.

Least Squares Solution

- In the case of a general nonspecific input sequence, the least-squares solution, if it exists, requires the calculation of the inverse of an autocorrelation matrix $\hat{\boldsymbol{\mathcal{R}}}^{(\ell)}$, of size $\binom{M}{\ell}\binom{N+1}{\ell} \times \binom{M}{\ell}\binom{N+1}{\ell}$.
- However, for our specific ℓ-D input sequences, the autocorrelation matrix greatly simplifies and attains the very sparse form of a block diagonal matrix.
- It is only necessary to calculate the inverse of the matrix $\hat{\mathbf{R}}^{(\ell)}$ of size $\binom{M}{\ell} \times \binom{M}{\ell}$.

Least Squares Solution

- In the case of a general nonspecific input sequence, the least-squares solution, if it exists, requires the calculation of the inverse of an autocorrelation matrix $\hat{\boldsymbol{\mathcal{R}}}^{(\ell)}$, of size $\binom{M}{\ell}\binom{N+1}{\ell} \times \binom{M}{\ell}\binom{N+1}{\ell}$.
- However, for our specific ℓ-D input sequences, the autocorrelation matrix greatly simplifies and attains the very sparse form of a block diagonal matrix.
- It is only necessary to calculate the inverse of the matrix $\hat{\mathbf{R}}^{(\ell)}$ of size $\binom{M}{\ell} \times \binom{M}{\ell}$.
- Hence, our identification algorithm provides a very special input sequence for which the least squares solution always exists and is much easier to calculate than the case of general inputs.

Communication Channel Identification

Communication Channel Identification

- Nonlinear channel identification is important in mitigating the effects of nonlinear distortions and for the equalization of the nonlinear communication channels

Communication Channel Identification

- 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 efforts for compensation of nonlinearities and channel equalization are highly dependent on the accuracy of the nonlinear channel estimate.

Communication Channel Identification

- We have applied the novel identification method to the identification of communication channels with nonlinearities.

Communication Channel Identification

- We have applied the novel identification method to the identification of communication channels with nonlinearities.
- We have modelled the channel as a third-order discrete Volterra filter and the Volterra kernels are measured using deterministic input sequences and the corresponding channel outputs.

Communication Channel Identification

- We have applied the novel identification method to the identification of communication channels with nonlinearities.
- We have modelled the channel as a third-order discrete Volterra filter and the Volterra kernels are measured using deterministic input sequences and the corresponding channel outputs.
- We present two numerical examples to illustrate the performance of our novel identification procedure.

Communication Channel Identification

Simulation 5.1: We simulate a linear-quadratic-cubic Volterra channel with memory length $N = 2$.

$$
\begin{aligned}
y(n) = {} & x(n) + 0.5x(n-1) - 0.8x(n-2) + x(n)^2 + \\
& 0.6x(n)x(n-1) - 0.3x(n-1)^2 + x(n)^3 + 1.2x(n)^2x(n-1) \\
& + 0.8x(n)x(n-1)^2 - 0.5x(n-1)^3 + x(n)x(n-1)x(n-2)
\end{aligned}
$$

We use QPSK modulated signals as the input, where the deterministic input levels are chosen from the set $4e^{j(2\pi k/4+\pi/4)}, k = 0, 1, 2, 3$.

Communication Channel Identification

Additive independent GWN observation noise with unit variance is present.
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 (Zhou and Giannakis, 1997).

Communication Channel Identification

Table 1: Results for Simulation 5.1

(i_1)	(0)	(1)	(2)
true $b_1\ (i_1)$	1.0000	0.5000	-0.8000
mean of $\hat{b}_1\ (i_1)$ for Zhou and Giannakis (1997)	0.9955	0.4886	-0.8150
mean of $\hat{b}_1\ (i_1)$ for our method	1.0045	0.5108	-0.8164
std of $\hat{b}_1\ (i_1)$ for Zhou and Giannakis (1997)	0.5195	0.3758	0.3326
std of $\hat{b}_1\ (i_1)$ for our method	0.1219	0.1266	0.1237

(i_1, i_2)	$(0, 0)$	$(0, 1)$	$(1, 1)$
true $b_2\ (i_1, i_2)$	1.0000	0.6000	-0.3000
mean of $\hat{b}_2\ (i_1, i_2)$ for Zhou and Giannakis (1997)	1.0035	0.6026	-0.2958
mean of $\hat{b}_2\ (i_1, i_2)$ for our method	1.0009	0.5984	-0.3035
std of $\hat{b}_2\ (i_1, i_2)$ for Zhou and Giannakis (1997)	0.1148	0.1335	0.0788
std of $\hat{b}_2\ (i_1, i_2)$ for our method	0.0014	0.0764	0.0050

Communication Channel Identification

(i_1, i_2, i_3)	$(0,0,0)$	$(0,0,1)$	$(0,1,1)$	$(1,1,1)$	$(0,1,2)$
true $b_3\,(i_1, i_2, i_3)$	1.0000	1.2000	0.8000	-0.5000	0.6000
mean of $\hat{b}_3\,(i_1, i_2, i_3)$ for Zhou and Giannakis (1997)	0.99995	1.2005	0.7993	-0.5009	0.5997
mean of $\hat{b}_3\,(i_1, i_2, i_3)$ for our method	1.0002	1.2006	0.8019	-0.4997	0.6036
std of $\hat{b}_3\,(i_1, i_2, i_3)$ for Zhou and Giannakis (1997)	0.0164	0.0235	0.0215	0.0281	0.0204
std of $\hat{b}_3\,(i_1, i_2, i_3)$ for our method	0.0077	0.0196	0.0185	0.0082	0.0285

Bandpass Communication Channel

The bandpass Volterra series is employed in the baseband representation of narrow-band communication channels.
The bandpass Volterra filter including nonlinearities up to third order is given as:

$$
\begin{aligned}
y(n) = & \sum_{i_1=0}^{N} b_1(i_1)x(n-i_1) + \\
& \sum_{i_1=0}^{N} \sum_{i_2=0}^{N} \sum_{i_3=i_2}^{N} b_3(i_1, i_2, i_3)x^*(n-i_1)x(n-i_2)x(n-i_3)
\end{aligned}
\tag{84}
$$

Bandpass Communication Channel

We can easily modify the identification method we developed for the regular Volterra filter to the bandpass Volterra channel case.
Simulation 2: We simulate a linear-cubic "bandpass" Volterra filter, where the input-output relationship for the bandpass Volterra filter is given in (84).

$$
\begin{aligned}
y(n) = x(n) &+ (0.5+0.5j)x(n-1) - 0.6x(n-2) + x(n)^* x(n-1)^2 \\
&+ (0.4+0.4j)x(n-1)^* x(n)^2 - 0.4x(n-1)^* x(n-2)^2 \\
&+ 0.6x(n-2)^* x(n-1)^2 + (0.6+0.7j)x(n-2)^* x(n)^2 \\
&+ 0.5x(n)^* x(n-2)^2 + (0.3+0.4j)x(n)^* x(n-1)x(n-2)
\end{aligned}
$$

Bandpass Communication Channel

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.
We also realized the method for bandpass Volterra kernel identification as given in Cheng and Powers (2001) for the simulation setup given above.

Bandpass Communication Channel

Table 2 shows the true values for the non-redundant bandpass Volterra kernels and the mean and the standard deviations of the estimates from our algorithm and the method detailed in (Cheng and Powers, 2001).
The results for our algorithm are better than those for the method of Cheng and Powers (2001) even though our method employed an input sequence of shorter length.

Bandpass Communication Channel

Table 2: Results for Simulation 5.2

(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 Cheng and Powers (2001)	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 Cheng and Powers (2001)	0.1131	0.1086	0.1050
std of $\hat{b}_1\ (i_1)$ for our method	0.0183	0.0193	0.0184

Bandpass Communication Channel

(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 Cheng and Powers (2001)	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 Cheng and Powers (2001)	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

Bandpass Communication Channel

(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 Cheng and Powers (2001)	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 Cheng and Powers (2001)	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

Lattice Realization for Volterra

Lattice Realization for Volterra

- We present a novel method for realizing nonlinear Volterra filters using the reduced-order 2-D orthogonal lattice filter structure.

Lattice Realization for Volterra

- We present a novel method for realizing nonlinear Volterra filters using the reduced-order 2-D orthogonal lattice filter structure.
- This method provides an orthogonal structure for arbitrary input signals and is capable of handling arbitrary lengths of memory for the system model.

Lattice Realization for Volterra

- We present a novel method for realizing nonlinear Volterra filters using the reduced-order 2-D orthogonal lattice filter structure.
- This method provides an orthogonal structure for arbitrary input signals and is capable of handling arbitrary lengths of memory for the system model.
- A recursive least squares adaptive-second order Volterra filter based on this structure is included to verify the performance.

Lattice Realization for Volterra

Consider the nonlinear system with the input-output relation based on the truncated second-order Volterra series expansion.

$$d(n) = \sum_{i_1=0}^{N-1} b_1\left(i_1;n\right)\ x(n-i_1) + \sum_{i_1=0}^{N-1}\sum_{i_2=i_1}^{N-1} b_2\left(i_1,i_2;n\right)\ x(n-i_1)x(n-i_2) \tag{85}$$

It is possible to describe the input-output relationship given in (85) as a pseudo-linear regression in the form of a vector product.

$$d(n) = \mathbf{X}_2^T(n)\mathbf{B}_2(n) \tag{86}$$

Lattice Realization for Volterra

$\mathbf{X}_2(n)$ will be of the form as described in the following equation.

$$\mathbf{X}_2(n) = \begin{bmatrix} x(n) \\ x(n-1) \\ \vdots \\ x(n-N+1) \\ x(n)^2 \\ x(n)x(n-1) \\ \vdots \\ x(n-N+1)^2 \end{bmatrix} \tag{87}$$

Lattice Realization for Volterra

$\mathbf{B}_2(n)$ is a vector which contains all the Volterra kernels as required in (85). $\mathbf{B}_2(n)$ will be given as,

$$\mathbf{B}_2(n) = \begin{bmatrix} b_1(0;n) \\ b_1(1;n) \\ \vdots \\ b_1(N-1;n) \\ b_2(0,0;n) \\ b_2(0,1;n) \\ \vdots \\ b_2(N-1,N-1;n) \end{bmatrix} \tag{88}$$

Lattice Realization for Volterra

- The direct form realization as indicated by (85) and (86) can suffer from ill conditioning, especially in nonlinear adaptive filtering applications.

Lattice Realization for Volterra

- The direct form realization as indicated by (85) and (86) can suffer from ill conditioning, especially in nonlinear adaptive filtering applications.
- In the literature, attempts have been made to find numerically robust alternative realization methods for the truncated Volterra filter (Lee and Mathews, 1993; Mathews, 1991; Ozden et al., 1996a; Syed and Mathews, 1994).

Lattice Realization for Volterra

- The direct form realization as indicated by (85) and (86) can suffer from ill conditioning, especially in nonlinear adaptive filtering applications.
- In the literature, attempts have been made to find numerically robust alternative realization methods for the truncated Volterra filter (Lee and Mathews, 1993; Mathews, 1991; Ozden et al., 1996a; Syed and Mathews, 1994).
- Both methods in Syed and Mathews (1994) and Ozden et al. (1996a) are based on the multichannel lattice structure as developed in Ling and Proakis (1984).

Lattice Realization for Volterra

- These methods convert the input signal vector as given in (87) into a multichannel signal and apply orthogonalization onto the multichannel signal while calculating the nonlinear output.

Lattice Realization for Volterra

- These methods convert the input signal vector as given in (87) into a multichannel signal and apply orthogonalization onto the multichannel signal while calculating the nonlinear output.
- We restructure the expanded input signal vector $\mathbf{X}_2(n)$ into a 2D array rather than using a multichannel setup.

Lattice Realization for Volterra

- These methods convert the input signal vector as given in (87) into a multichannel signal and apply orthogonalization onto the multichannel signal while calculating the nonlinear output.
- We restructure the expanded input signal vector $\mathbf{X}_2(n)$ into a 2D array rather than using a multichannel setup.
- It is possible to realize the Volterra system as a joint-process estimator with a lattice-ladder structure instead of the direct form realization as in (85).

Lattice Realization for Volterra

- We reshape the vector $\mathbf{X}_2(n)$ into a 2D array using the proposed ordering as in Fig. 14.

Lattice Realization for Volterra

- We reshape the vector $\mathbf{X}_2(n)$ into a 2D array using the proposed ordering as in Fig. 14.
- Fig. 15 shows the indexing arrangement we chose for the input array.

Lattice Realization for Volterra

- We reshape the vector $\mathbf{X}_2(n)$ into a 2D array using the proposed ordering as in Fig. 14.
- Fig. 15 shows the indexing arrangement we chose for the input array.
- Fig. 16 depicts the 2-D orthogonal lattice structure-based nonlinear joint-process estimator.

Lattice Realization for Volterra

- We reshape the vector $\mathbf{X}_2(n)$ into a 2D array using the proposed ordering as in Fig. 14.
- Fig. 15 shows the indexing arrangement we chose for the input array.
- Fig. 16 depicts the 2-D orthogonal lattice structure-based nonlinear joint-process estimator.
- The depicted full-complexity nonlinear joint-process estimator is complete with the lattice predictor part and the ladder section.

Lattice Realization for Volterra

$x(n)x(n\text{-}N{+}1)$					
$x(n)x(n\text{-}N{+}2)$	$x(n\text{-}1)x(n\text{-}N{+}1)$				
$\vdots$	$\vdots$	$\ddots$			
$x(n)x(n\text{-}1)$	$x(n\text{-}1)x(n\text{-}2)$	$x(n\text{-}2)x(n\text{-}3)$	$\cdots$	$x(n\text{-}N{+}2)$ $x(n\text{-}N{+}1)$	
$x(n)^2$	$x(n\text{-}1)^2$	$x(n\text{-}2)^2$	$\cdots$	$x(n\text{-}N{+}2)^2$	$x(n\text{-}N{+}1)^2$
$x(n)$	$x(n\text{-}1)$	$x(n\text{-}2)$	$\cdots$	$x(n\text{-}N{+}2)$	$x(n\text{-}N{+}1)$

Figure 14: Ordering scheme for the 2-D input array.

Lattice Realization for Volterra

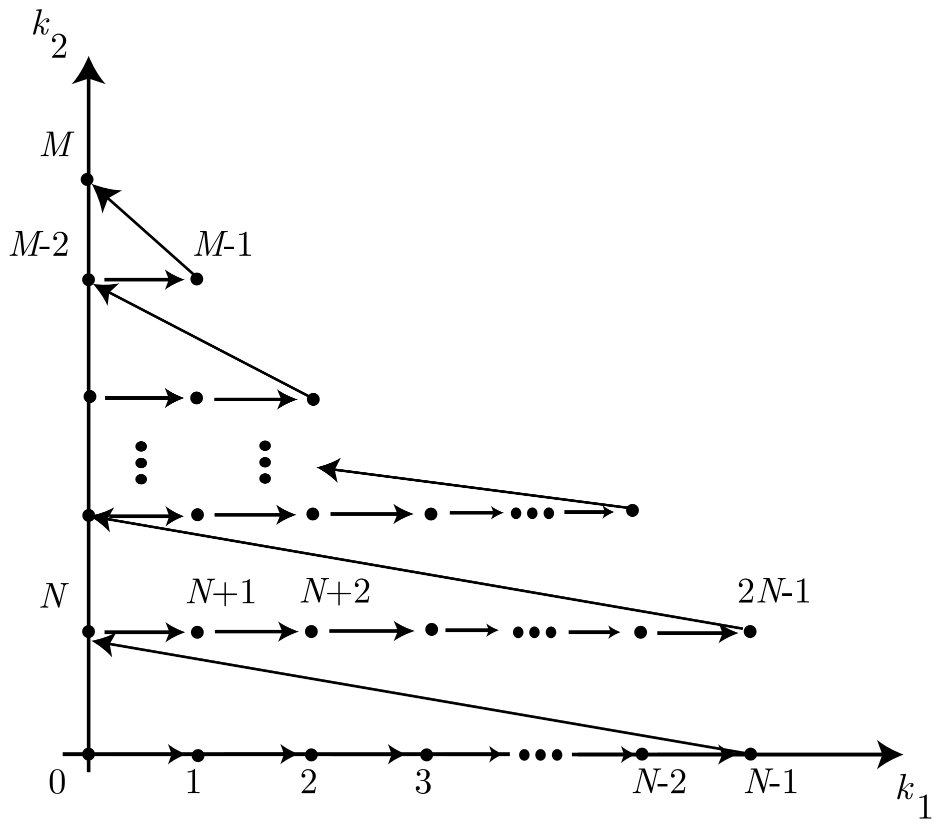


Figure 15: Indexing scheme for the 2-D input array.

Lattice Realization for Volterra

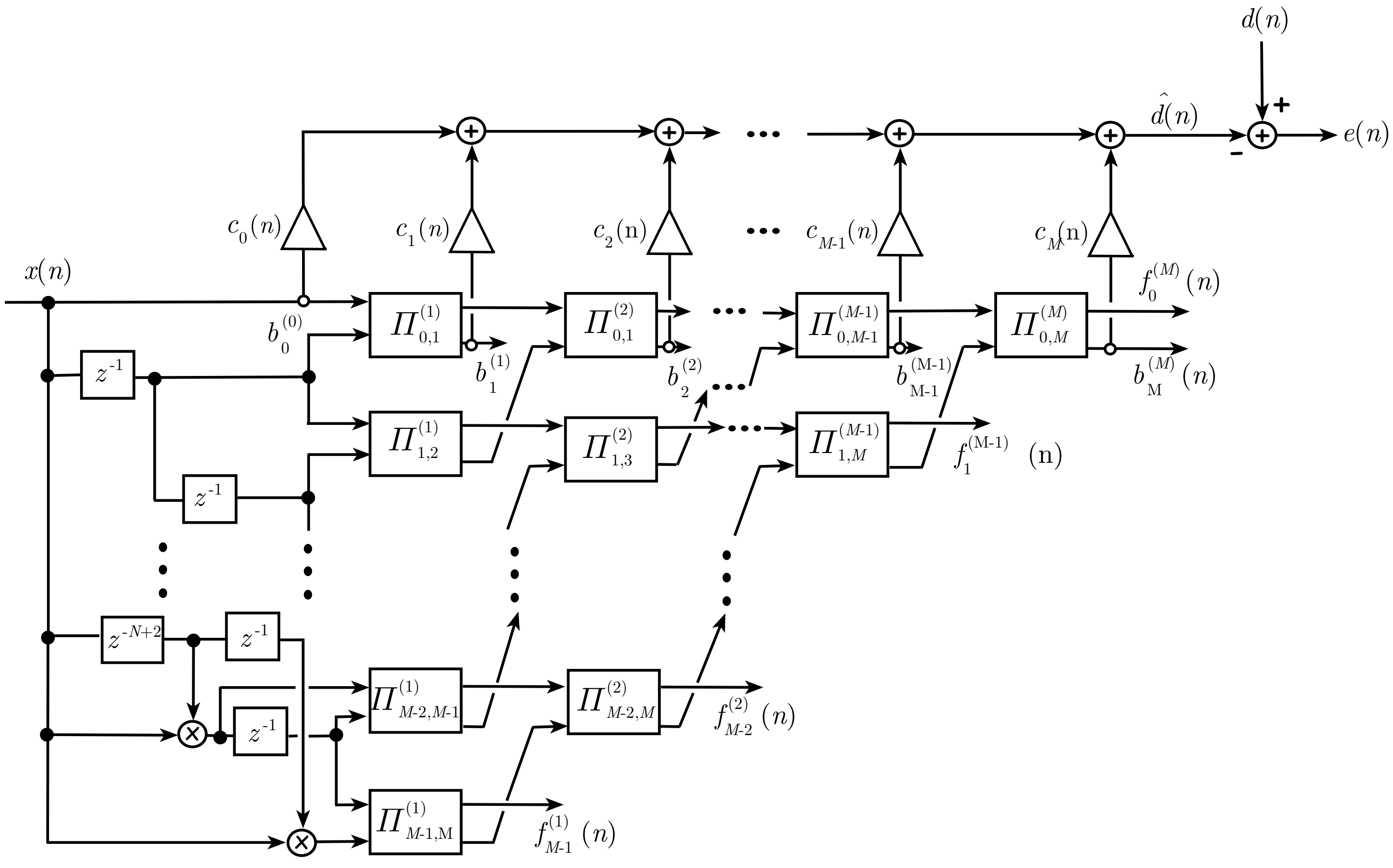


Figure 16: Full complexity nonlinear joint-process estimator.

Lattice Realization for Volterra

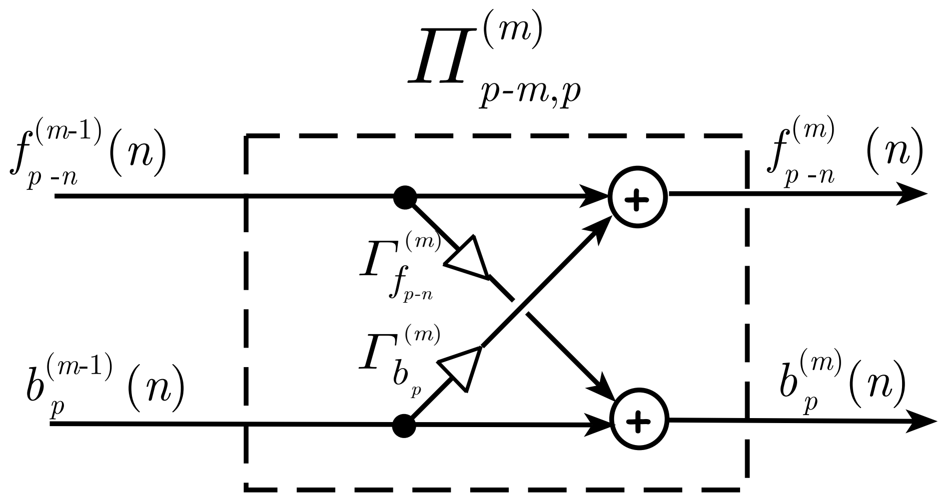


Figure 17: Internal structure of the basic lattice module utilized in the nonlinear joint process estimator.

Lattice Realization for Volterra

- The backward prediction errors $b_0^{(0)}(n), b_1^{(1)}(n), \ldots, b_M^{(M)}(n)$ generated using the 2D lattice filter are orthogonal to each other (Kayran, 1996b).

Lattice Realization for Volterra

- The backward prediction errors $b_0^{(0)}(n), b_1^{(1)}(n), \ldots, b_M^{(M)}(n)$ generated using the 2D lattice filter are orthogonal to each other (Kayran, 1996b).
- This result provides the main advantage of our structure over the multichannel lattice structure in (Syed and Mathews, 1994). For the structure in Syed and Mathews (1994), although the backward prediction errors in different channels are orthogonal to each other, the elements within each channel are not orthogonalized.

Lattice Realization for Volterra

- The backward prediction errors $b_0^{(0)}(n), b_1^{(1)}(n), \ldots, b_M^{(M)}(n)$ generated using the 2D lattice filter are orthogonal to each other (Kayran, 1996b).
- This result provides the main advantage of our structure over the multichannel lattice structure in (Syed and Mathews, 1994). For the structure in Syed and Mathews (1994), although the backward prediction errors in different channels are orthogonal to each other, the elements within each channel are not orthogonalized.
- However, in our structure the backward prediction errors are fully orthogonalized to each other. This will result in faster and less input dependent adaptation.

Lattice Realization for Volterra

Simulation 1: The setting for the simulations is shown in Fig. 18. In the simulation the adaptive filter was run with the same memory length N as that of the second-order Volterra filter to be identified.
The Volterra system we identify has $N = 4$, hence there are 4 linear and 10 quadratic coefficients.
The desired response signal $d(n)$ was obtained by adding white Gaussian noise uncorrelated with the input signal to the output.
The variance of the observation noise was chosen to obtain an SNR of 20 dB.

Lattice Realization for Volterra

- We present the learning curves in Fig. 19, for our lattice structure, the multichannel lattice structure in Syed and Mathews (1994) and the direct form transversal realization (Mathews, 1991), all with RLS adaptation in Fig. 19. The error curves are mean squared for 500 cycles and $\lambda = 0.9975$.

Lattice Realization for Volterra

- We present the learning curves in Fig. 19, for our lattice structure, the multichannel lattice structure in Syed and Mathews (1994) and the direct form transversal realization (Mathews, 1991), all with RLS adaptation in Fig. 19. The error curves are mean squared for 500 cycles and $\lambda = 0.9975$.
- The novel lattice-based structure maintains the excellent numerical behavior of the lattice models. Our structure exhibits better performance than both the transversal realization and the multichannel lattice structure.

Lattice Realization for Volterra

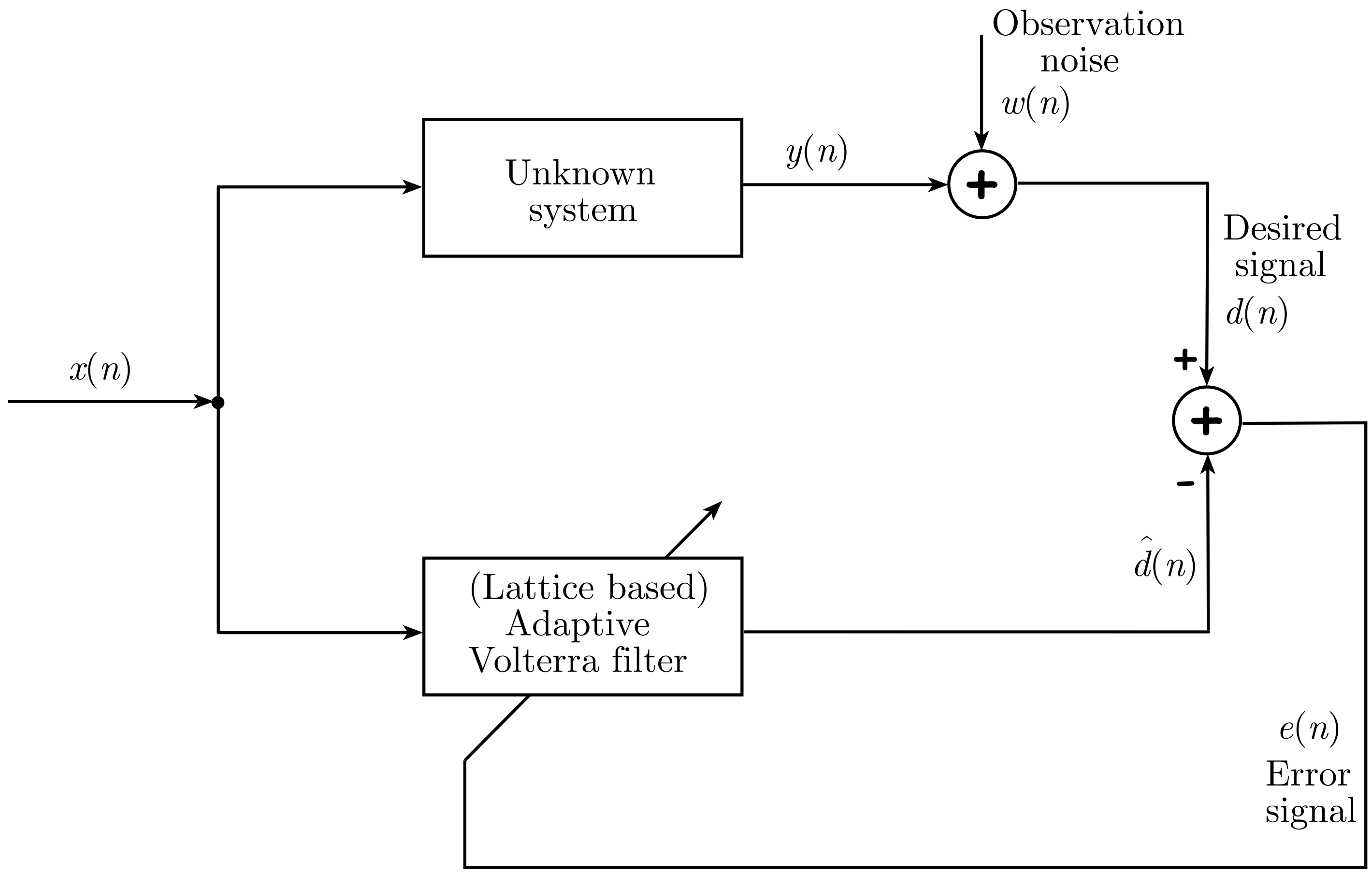


Figure 18: The general setup for the adaptive second-order Volterra filter identification simulations.

Lattice Realization for Volterra

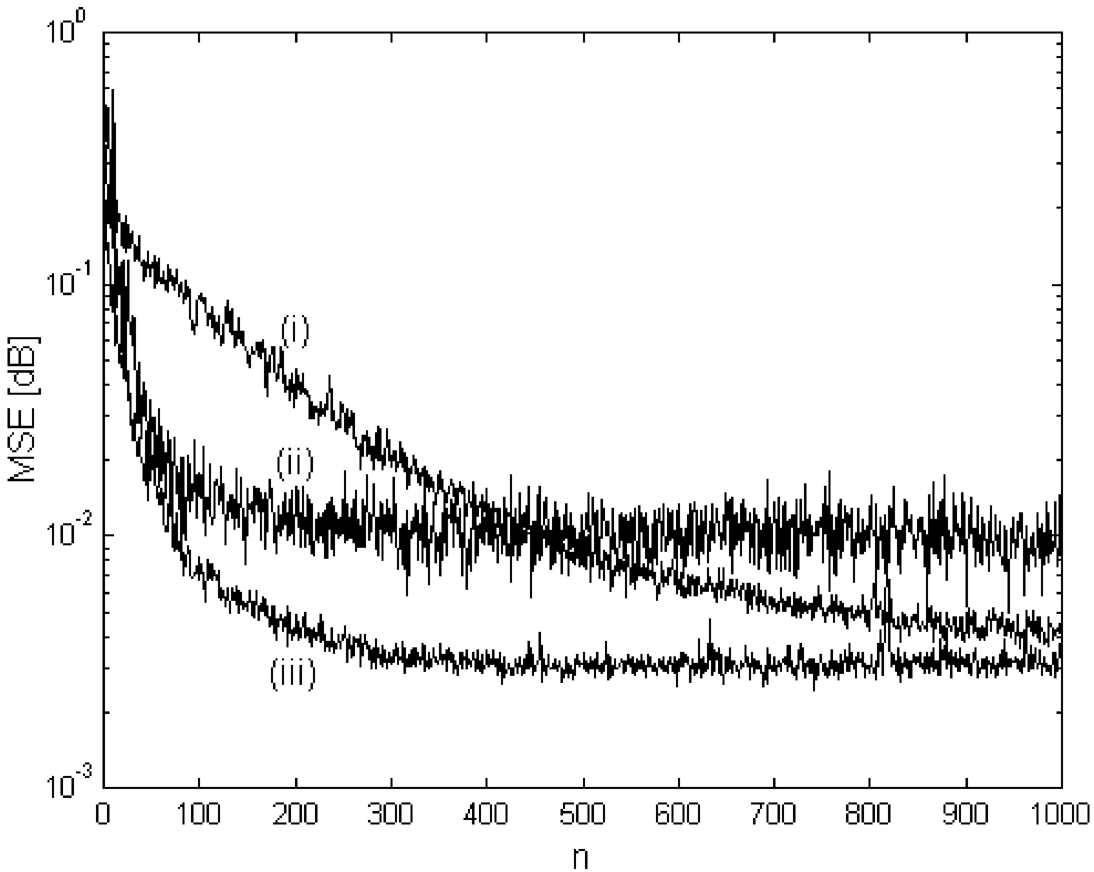


Figure 19: Learning curves for different models; (i) multichannel lattice structure, (ii) transversal direct-form realization, (iii) model based on 2-D lattice structure.

Concluding Remarks

Concluding Remarks

- This dissertation considered the design of a novel representation for the discrete-time, time-invariant, finite-order Volterra filters.

Concluding Remarks

- This dissertation considered the design of a novel representation for the discrete-time, time-invariant, finite-order Volterra filters.
- We also developed a novel extension of the unit impulse response to the case of the identification of nonlinear Volterra filters.

Concluding Remarks

- This dissertation considered the design of a novel representation for the discrete-time, time-invariant, finite-order Volterra filters.
- We also developed a novel extension of the unit impulse response to the case of the identification of nonlinear Volterra filters.
- We applied the developed identification algorithm successfully to the nonlinear communication channels and baseband Volterra communication channels with communication signals as inputs.

Concluding Remarks

- Future work on identification might be by fusing the novel Volterra system representation with the use of random sequences as inputs, rather than deterministic sequences.

Concluding Remarks

- Future work on identification might be by fusing the novel Volterra system representation with the use of random sequences as inputs, rather than deterministic sequences.
- The novel representation might be also applied in the efficient implementation of Volterra filters and in transform domain structures.

Concluding Remarks

- Future work on identification might be by fusing the novel Volterra system representation with the use of random sequences as inputs, rather than deterministic sequences.
- The novel representation might be also applied in the efficient implementation of 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.

Concluding Remarks

- The work on the novel orthogonal quadratic Volterra filter realization based on the 2D lattice structure can be adapted to the realization of Volterra filter with higher order nonlinearities.

Concluding Remarks

- The work on the novel orthogonal quadratic Volterra filter realization based on the 2D lattice structure can be adapted to the realization of Volterra filter with higher order nonlinearities.
- The orthogonal structure can be also utilized in the realization of polynomial systems with feedback such as bilinear systems.

Teşekkürler

References

Agazzi, O., Messerschmitt, D. G. and Hodges, D. A., 1982. Nonlinear echo cancelation of data signals, *IEEE Trans. Communications*, **COM30**(11), 2421–2433.

Baik, H. K. and Mathews, V. J., 1993. Adaptive lattice bilinear filters, *IEEE Trans. Signal Processing*, **41**(6), 2033–2046.

Benedetto, S. and Biglieri, E., 1983. Nonlinear equalization of digital satellite channels, *IEEE Journal on Selected Areas in Communications*, **SAC-1**(1), 57–62.

Benedetto, S., Biglieri, E. and Castellani, V., 1987. *Digital Transmission Theory*, Prentice-Hall, New Jersey.

Benedetto, S., Biglieri, E. and Daffara, R., 1979. Modeling and performance evaluation of nonlinear satellite links–a Volterra series approach, *IEEE Transactions on Aerospace and Electronic Systems*, **AES15**(4), 494–507.

Biglieri, E., Barberis, S. and Catena, M., 1988. Analysis and compensation of nonlinearities in digital transmission systems, *IEEE Journal on Selected Areas in Communications*, **6**(1), 42–51.

Billings, S. A., 1980. Identification of nonlinear systems–a survey, *IEE Proceedings*, **127**(8), 272–285.

Boyd, S. and Chua, L., 1985. Fading memory and the problem of approximating nonlinear operators with Volterra series, *IEEE Trans. on Circuits and Systems*, **CAS32**(11), 1150–1161.

Boyd, S., Tang, Y. S. and Chua, L. O., 1983. Measuring Volterra kernels (I), *IEEE Trans. on Circuits and Systems*, **CAS-30**(8), 571–577.

Chen, S., Billings, S. A. and Luo, W., 1989. Orthogonal least squares methods and their application to nonlinear system identification, *Int. J. Control*, **50**, 1873–1896.

Cheng, C., 1998. Nonlinear communication system identification and compensation, Ph.D. thesis, University of Texas, Austin, USA.

Cheng, C. and Powers, E. J., 2001. Optimal Volterra kernel estimation algorithms for a nonlinear communication system for PSK and QAM inputs, *IEEE Trans. Signal Processing*, **49**(12), 147–163.

Ching, H. and Powers, E., 1993. Nonlinear channel equalization in digital satellite system, *Proceedings of GLOBECOM'93*, volume 3, Houston, USA, pp. 1639–1643.

Chua, L. and Liao, Y., 1983. Measuring Volterra kernels (II), *International Journal of Circuit Theory and Applications*, **17**, 151–190.

Falconer, D. D., 1978. Adaptive equalization of channel nonlinearities in QAM data transmission systems, *Bell System Technology Journal*, **57**(7), 2589–2611.

Frank, W. A., 1995. An efficient approximation to the quadratic Volterra filter and its application to realtime loudspeaker linearization, *Signal Processing*, **45**, 97–113.

Giannakis, G. B. and Serpedin, E., 2001. A bibliography on nonlinear system identification, *Signal Processing*, **81**(3), 533–580.

Hermann, R., 1990. Volterra modeling of digital magnetic saturation recording channels, *IEEE Trans. Magnetics*, **26**(5), 2125–2127.

Kajikawa, Y., 2000. The adaptive Volterra filter: Its present and future, *Electronics and Communications in Japan*, **83**(12), 51–61.

Kayran, A., 1996a. 2D ARMA lattice modelling using two-channel AR lattice, *Electronics Letters*, **32**(16), 1434–1435.

Kayran, A., 1996b. Two-dimensional orthogonal lattice structures for autoregressive modeling of random fields, *IEEE Trans. Signal Processing*, **44**(4), 963–978.

Kayran, A. and Ekşioğlu, E., 2000. 2D FIR Wiener filter realization using orthogonal lattice structure, *Electronics Letters*, **36**(12), 1078–1079.

Koh, T. and Powers, E. J., 1985. Second-order Volterra filtering and its application to nonlinear system identification, *IEEE Trans. on Acoust. Speech and Signal Processing*, **ASSP33**(16), 1445–1455.

Korenberg, M. J. and Paarman, L. D., 1991. Orthogonal approaches to time series analysis and system identification, *IEEE Signal Proc. Magazine*, **8**(3), 29–43.

Koukoulas, P. and Kalouptsidis, N., 1995. Nonlinear system identification using Gaussian inputs, *IEEE Trans. Signal Processing*, **43**(8), 1831–1841.

Lazzarin, G., Pupolin, S. and Sarti, A., 1994. Nonlinearity compensation in digital radio systems, *IEEE Trans. Communications*, **42**, 988–998.

Lee, J. and Mathews, V. J., 1993. A fast recursive least squares adaptive second-order Volterra filter and its performance analysis, *IEEE Trans. Signal Processing*, **41**(3), 10871102.

Lee, Y. W. and Schetzen, M., 1965. Measurement of the Wiener kernels of a nonlinear system by cross-correlation, *Int. J. Control*, **2**, 237–254.

Ling, F. and Proakis, J., 1984. A generalized multichannel least-squares lattice algorithm based on sequential processing stages, *IEEE Trans. on Acoust. Speech and Signal Processing*, **ASSP32**(2), 381–389.

Manolakis, D., Ingle, V. and Kogon, S., 2000. *Statistical and Adaptive Signal Processing*, McGraw-Hill, Boston.

Marmarelis, P. Z. and Marmarelis, V. Z., 1978. *Analysis of Physiological Systems*, Plenum Press, New York.

Marmarelis, V. Z. and Naka, K., 1972. White noise analysis of a neuron chain: an application of the Wiener theory, *Science*, **175**(5), 1276–1278.

Mathews, V., 1991. Adaptive polynomial filters, *IEEE Signal Processing Mag.*, **8**, 10–26.

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

Nowak, R. and Van Veen, B., 1994a. Efficient methods for identification of Volterra filter models, *Signal Processing*, **38**(3), 417–428.

Nowak, R. and Van Veen, B., 1994b. Random and pseudorandom inputs for Volterra filter identification, *IEEE Trans. Signal Processing*, **42**(8), 2124–2135.

Ozden, M., Panayırcı, E. and Kayran, A., 1997. Identification of nonlinear magnetic channels with lattice orthogonalisation, *Electronics Letters*, **33**(5), 376–377.

Ozden, M. T., Kayran, A. H. and Panayırcı, E., 1996a. Adaptive Volterra filtering with complete lattice orthogonalization, *IEEE Trans. Signal Processing*, **44**(4), 2092–2098.

Ozden, M. T., Kayran, A. H. and Panayırcı, E., 1996b. Satellite channel identification with lattice orthogonalisation, *Electronics Letters*, **32**(4), 302–304.

Ozden, M. T., Kayran, A. H. and Panayırcı, E., 1998. Adaptive Volterra channel equalization with lattice ortogonalisation, *IEE Proceedings–Communications*, **145**(2), 109–115.

Palm, G. and Poggio, T., 1985. The Volterra representation and the Wiener expansion: Validity and pitfalls, *SIAM J. Appl. Math.*, **33**, 195–216.

Parker, S. and Perry, F., 1981. A discrete ARMA model for nonlinear system identification, *IEEE Trans. Circuits and Systems*, **CAS28**, 224–233.

Petrochilos, N. and Comon, P., 2000. Nonlinear channel identification and performance analysis, *IEEE International Conference on Acoustics*, volume 1, Istanbul, Turkey, pp. 209–212.

Ramponi, G., 1990. Bi–impulse response design of isotropic quadratic filters, *Proc. IEEE*, **78**(4), 665–677.

Raz, G. and Van Veen, B., 1998. Baseband Volterra filters for implementing carrier based nonlinearities, *IEEE Trans. Signal Processing*, **46**(1), 103–114.

Raz, G. M., 1998. Nonlinear system representation, identification and implementation, Ph.D. thesis, University of Wisconsin, Madison, USA.

Reed, M. and Hawksford, M., 1996. Identification of discrete Volterra series using maximum length sequences, *Proc. IEE, Circuits, Devices and Systems*, **143**(5), 241–248.

Rugh, W. J., 1981. *Nonlinear System Theory. The Volterra-Wiener Approach*, Johns Hopkins University Press, Baltimore.

Saleh, A. A. M., 1981. Frequency-independent and frequency-dependent nonlinear models of TWT amplifiers, *IEEE Trans. Communications*, **29**(11), 1715–1720.

Sandberg, I. W., 1983. The mathematical foundations of associated expansions for mildly nonlinear systems, *IEEE Trans. Circuits Systems*, **30**, 441–445.

Sandberg, I. W., 1992. Uniform approximation with doubly finite Volterra series, *IEEE Trans. Signal Processing*, **40**, 1438–1442.

Schetzen, M., 1965. Measurement of the kernels of a non-linear system of finite order, *Int. J. Control*, **1**(3), 251–263.

Schetzen, M., 1974. A theory of non-linear system identification, *Int. J. Control*, **20**(4), 577–592.

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

Shi, Y. and Hecox, K. E., 1991. Nonlinear system identification by m-pulse sequences: Application to brainstem auditory evoked responses, *IEEE Trans. Biomedical Engineering*, **438**(9), 834–845.

Strang, G., 1998. *Introduction to Linear Algebra*, Wellesley Cambridge Press.

Syed, M. and Mathews, V., 1993. QR--decomposition based algorithms for adaptive Volterra filtering, *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, **40**(6), 372–382.

Syed, M. and Mathews, V., 1994. Lattice algorithms for recursive least squares adaptive second-order Volterra filtering, *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, **41**(3), 202–214.

Tick, L. J., 1961. The estimation of transfer functions of quadratic systems, *Technometrics*, **3**(4), 563–567.

Tseng, C. and Powers, E. J., 1993. Nonlinear channel equalization in digital satellite systems, *Proc. GLOBECOM*, volume 3, Houston, TX, USA, pp. 1639–1643.

Tseng, C.-H., 1993. Advanced nonlinear system identification techniques and their applications to engineering problems, Ph.D. thesis, University of Texas, Austin, USA.

Volterra, V., 1887. Sopra le funzioni che dipendono de altre funzioni, *Rend. Regia Accademia dei Lincei*, 97–105.

Volterra, V., 1959. *Theory of Functions and of Integral and Integro-Differential Equations*, Dover Publications, New York.

Wiener, N., 1958. *Nonlinear Problems in Random Theory*, M.I.T. Press.

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