166650

ISTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY

NOVEL METHODS FOR VOLTERRA FILTER REPRESENTATION, IDENTIFICATION AND REALIZATION

Ph.D. Thesis by

Ender Mete EKŞİOĞLU, M.Sc.

(504992415)

Date of submission : December 3, 2004

Date of defence examination : March 7, 2005

Supervisor (Chairman) : Prof. Dr. Ahmet Hamdi KAYRAN

Members of the Examining Committee : Prof. Dr. Erdal PANAYIRCI (I.U.)

Prof. Dr. Metin YÜCEL (Y.T.U.)

Prof. Dr. Serhat ŞEKER (I.T.U.)

Prof. Dr. Ümit AYGÖLÜ (I.T.U.)

MARCH 2005

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

VOLTERRA SÜZGECİ GÖSTERİLİMİ, TANILAMASI VE GERÇEKLEMESİ İÇİN YENİ YÖNTEMLER

DOKTORA TEZİ

Y. Müh. Ender Mete EKŞİOĞLU

(504992415)

Tezin Enstitüye Verildiği Tarih : 3 Aralık 2004
Tezin Savunulduğu Tarih : 7 Mart 2005

Tez Danışmanı : Prof. Dr. Ahmet Hamdi KAYRAN
Diğer Jüri Üyeleri : Prof. Dr. Erdal PANAYIRCI (I.Ü.)
Prof. Dr. Metin YÜCEL (Y.T.Ü.)
Prof. Dr. Serhat ŞEKER (İ.T.Ü.)
Prof. Dr. Ümit AYGÖLÜ (İ.T.Ü.)

MART 2005

PREFACE

I am most grateful to my supervisor Professor Ahmet H. Kayran for his committed guidance and continual support during the preparation of this thesis. This work would have been impossible without his deep insight and encouragement.

I express my gratitude to Professors Erdal Panayırcı and Ümit Aygölü for serving in the committee overseeing the progress of my thesis work.

Last but not least, I thank my family for their love and patience. This dissertation is dedicated to my parents Belkıs and Halil Ekşioğlu. Their sympathy has been my sustenance during the arduous writing process of this dissertation.

March 2005 Ender Mete Ekşioğlu

CONTENTS Page

LIST OF ABBREVIATIONS

AGWN	:	Additive Gaussian White Noise
LTI	:	Linear Time Invariant
PE	:	Persistently Exciting
PRBS	:	Pseudo Random Binary Sequence
PRMS	:	Pseudo Random Multilevel Sequence
PSK	:	Phase Shift Keying
QAM	:	Quadrature Amplitude Modulation
QPSK	:	Quadrature Phase Shift Keying
RLS	:	Recursive Least Squares
SNR	:	Signal to Noise Ratio

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

$b_k(\tau_1, \tau_2, \ldots, \tau_k)$	:	k^{th} order continuous-time Volterra kernel
$\bar{b}_k(i_1, i_2, \ldots, i_k)$	:	k^{th} order discrete-time redundant Volterra kernel
$\bar{b}_{m,sym}(i_1, i_2, \ldots, i_m)$	:	k^{th} order discrete-time symmetric Volterra kernel
$b_k(i_1, i_2, \ldots, i_k)$	:	k^{th} order discrete-time (triangular) Volterra kernel
$h_k^{(p_1,p_2,\ldots p_\ell)}(q_1, \ldots, q_{\ell-1}; i)$	:	cross-term Volterra kernel in the ℓ-D cross-term Volterra kernel vector
$\bar{b}_{m,reg}(k_1, k_2, \ldots, k_m)$	:	m^{th} order discrete-time regular Volterra kernel
$\mathbf{h}^{(\ell)}(q_1, \ldots, q_{\ell-1}; i)$	:	ℓ-D cross-term Volterra kernel vector
$\mathbf{x}^{(\ell)}(q_1, \ldots, q_{\ell-1}; n)$	:	ℓ-D input vector in the novel Volterra representation
$\mathbf{x}_e^{(\ell)}(q_1, \ldots, q_{\ell-1}; n)$	:	ℓ-D input ensemble vector utilized in the novel identification method
$\mathbf{y}_e^{(\ell)}(q_1, \ldots, q_{\ell-1}; n)$	:	ℓ-D output ensemble vector utilized in the novel identification method
$\mathbf{v}^{(\ell,m)}(q_1, \ldots, q_{\ell-1}; n)$	:	the output of the ℓ-D cross-term subsystem for the m-D input ensemble
$\binom{M}{n}$	:	number of combinations of M things taken n at a time
$C_{\text{redundant}}$	:	number of kernels in the redundant Volterra representation
$C_{\text{triangular}}$	:	number of kernels in the triangular Volterra representation
$\mathcal{N}[\cdot]$	:	nonlinear Volterra filter
$\mathcal{B}_k[\cdot]$	:	k^{th}-order subsystem in the triangular Volterra representation
$\mathcal{H}^{(\ell)}[\cdot]$	:	ℓ-D cross-term Volterra subsystem in the novel Volterra representation
$\mathcal{C}[\cdot]$	:	nonlinear communication channel based on the Volterra filter
$y_k(n)$	:	output signal for the subsystem $\mathcal{B}_k$
$y^{(\ell)}(n)$	:	output signal for the subsystem $\mathcal{H}^{(\ell)}$
C_ℓ	:	length of the ℓ-D input ensemble vector
L_ℓ	:	number of ℓ-D kernel vectors

C_{ℓ}^{delay}	:	number of possible delay combinations for ℓ-D subsystem
$\mathbf{U}_e^{(\ell)}$	:	ℓ-D ensemble input matrix
$\mathbf{S}_{\ell k,j}^{(M)}$	:	output pick-up matrix
$\mathbf{T}_{\ell,i}^{(M)}$	:	input formation matrix
$\mathbb{S}_{\ell,k}^{(M)}[\,\cdot\,]$	:	output pick-up operator
$\boldsymbol{\Pi}(N,K)$	:	$N \times K$ lattice block utilized in the reduced complexity orthogonal lattice structure for Volterra filters
$\Gamma_{f_{p-m}}^{(m)}(n)$	:	m^{th} stage forward prediction reflection coefficients
$\Gamma_{b_p}^{m}(n)$	:	m^{th} stage backward prediction reflection coefficients
λ	:	forgetting factor for RLS adaptation

NOVEL METHODS FOR VOLTERRA FILTER REPRESENTATION, IDENTIFICATION AND REALIZATION

SUMMARY

In this dissertation we deal with discrete-time, finite-order, time-invariant Volterra filters. Firstly, we develop a new representation for the finite-order Volterra filters. This representation introduces a novel partitioning of the Volterra kernels. We know that the unit impulse response is insufficient to fully characterize a nonlinear system unlike linear time-invariant systems. We formulate a novel exact identification method for Volterra filters, which uses deterministic sequences consisting of impulses with distinct levels. This identification method is based on the novel representation we develop. 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. To our best knowledge, this method is the first full-scale generalization of the impulse response representation to the finite order Volterra type nonlinear systems.

Our identification method is exact; hence, it calculates the exact Volterra kernels in the absence of noise for very short length input sequences. Our method calculates each Volterra kernel individually. The kernel estimates are not utilized in the calculation of further kernel estimates. This property hinders error propagation among estimates. Our method calculates directly the Volterra kernels, instead of calculating first some intermediary representation such as the Wiener kernels, which do not have any directly interpretable results. 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.

We also show that the input sequence we develop for identification is persistently exciting for the Volterra filters under consideration. We discuss also the problem of ill-conditioning and the numerical properties of the algorithm.

We apply the novel identification method to the identification of the Volterra kernels of nonlinear communication channels modelled as third-order Volterra filters. The algorithm works for complex inputs, allowing the use of complex baseband communication signals. Hence, the algorithm can be utilized in the identification of baseband Volterra channel models. We demonstrate with several simulations that the identification algorithm can produce better parameter estimates than some most recent algorithms in the literature. We develop simulations both for the nonlinear system identification setting, and the nonlinear and baseband communication channel identification setting.

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. An RLS adaptive nonlinear filter based on this fully orthogonal structure is presented. The performance of our structure in the adaptive system identification setting is superior to previous models from the literature.

VOLTERRA SÜZGECİ GÖSTERİLİMİ, TANILAMASI VE GERÇEKLEMESİ İÇİN YENİ YÖNTEMLER

ÖZET

Bu tezde ayrık-zaman, sınırlı dereceli, zamanla-değişmez Volterra süzgeçleri konu edilmektedir. İlk olarak sınırlı doğrusalsızlık derecesine sahip Volterra süzgeçleri için yeni bir gösterilim geliştirilmektedir. Bu gösterilimde Volterra çekirdekleri şimdiye kadar literatürde olmayan bir şekilde ayrıştırılmaktadır.

Bilindiği üzere birim darbe cevabı, doğrusal, zamanla-değişmez sistemlerdeki durumun tersine, doğrusal olmayan sistemleri tam olarak niteleyememektedir. Tezde Volterra süzgeçleri için yeni ve kesin bir tanılama yöntemi sunulmaktadır. Bu yeni yöntem, giriş işareti olarak farklı seviyelere sahip birim darbelerden oluşan gerekirci diziler kullanmaktadır ve geliştirilen yeni gösterilime dayanmaktadır. Yeni tanılama yöntemi doğrusal, zamanla-değişmez sistemlerdeki birim darbe cevabının doğrusal olmayan sistemlere başarılı bir uyarlaması olarak düşünülebilir. Gerçekten de bu yeni yöntem, tanılanacak olan sistem doğrusal bir sistem olduğunda, birim darbe cevabı kullanılarak tanılamayı bir alt durum olarak içermektedir. Bu yeni tanılama yöntemi, birim darbe gösteriliminin Volterra süzgeçlerine ilk başarılı genelleştirilmesidir.

Tezde sunulan tanılama yöntemi kesindir; böylece gözlem gürültüsü olmadığında Volterra çekirdeklerini hatasız kestirmektedir. Yöntemde tüm Volterra çekirdekleri ayrı ayrı kestirilmektedir. Çekirdek kestirimleri diğer çekirdeklerin bulunmasında kullanılmamaktadır. Böylece kestirimler arasında hata yayılımı engellenmektedir. Yöntem Wiener çekirdeği ve benzeri ara gösterilimleri değil, doğrudan Volterra çekirdeklerini kestirmektedir. Yeni Volterra gösteriliminde, düzgün Volterra süzgeci için gerekli olan Volterra çekirdekleri dışında kalan hiçbir katsayı gerekmemektedir. Tanılama yöntemi hem tanılanan çekirdeklerin sayısı hem de tanılama için kullanılan giriş dizilerinin uzunluğu açısından gereksizlikten arınmıştır. Tezde, tanılama için kullanılan giriş dizilerinin, Volterra süzgeçleri için direngen uyarıcı olduğu gösterilmektedir. Geliştirilen tanılama algoritmasının kötü-koşullanma sorunuyla karşılaşmaması için gerekli önlemler ve algoritmanın sayısal özellikleri de tartışılmaktadır.

Geliştirilen tanılama yöntemi doğrusal olmayan haberleşme kanallarının kestirimine uygulanmıştır. Algoritma karmaşık değerli giriş dizileri ve çekirdek değerleri için de sorunsuz çalışmaktadır. Böylece temelband Volterra kanallarının tanılanmasında kullanılabilmektedir. Bilgisayar benzetimleriyle tanılama yönteminin literatürde yakın zamanda sunulmuş olan bazı yöntemlerden daha iyi kestirim sonuçları verdiği gösterilmiştir. Benzetimler doğrusal olmayan sistem tanılama, doğrusal olmayan haberleşme kanalı tanılama ve doğrusal olmayan temelband haberleşme kanalı tanılanması uygulamaları için geliştirilmiştir.

Tezin ikincil katkısı, Volterra süzgeçleri için dik gerçeklemeler alanındadır. Volterra süzgeçlerinin gerçeklenmesi için tam olarak dikliğe sahip yeni bir yapı geliştirilmektedir. Bu yapı yakın zamanda literatüre giren 2-boyutlu kafes süzgeç modelini temel almaktadır. Geliştirilen yeni yapı RLS algoritmasıyla uyarlanan uyarlamalı, doğrusal olmayan süzgeçlerin gerçeklenmesinde kullanılmaktadır. Yeni gerçeklemenin uyarlamalı, doğrusal olmayan sistem tanılama uygulamasında başarımının, literatürde yer alan yapılardan daha yüksek olduğu bilgisayar benzetimleriyle gösterilmektedir.

1. INTRODUCTION

The use of linear system models has been well established with successful applications. However, there are still a large number of problems where the linear system paradigm fails and one has to resort to nonlinear system models. Nonlinear system research has been a very active area, a fact which is attested to by the sheer number and extent of the references listed in the recent bibliography by Giannakis and Serpedin (2001).

As known, linear systems are fully described by their impulse response. A linear system can be readily identified by finding the impulse response or equivalently the frequency response. However, since nonlinear systems encompass all systems which are not linear, there is no such unified framework for the representation of nonlinear systems. There are various categories for modelling nonlinear systems, such as homomorphic systems, neural networks and order-statistic filters. In this dissertation we will be dealing with nonlinear polynomial system models based on the Volterra series representation. Volterra filters based on the Volterra series have been an attractive nonlinear system class due to some desired properties. These advantages include the following:

1. Volterra filters bear similarities to the well-developed linear system theory.
2. Volterra filters can approximate a large class of nonlinear systems with a finite number of coefficients under relatively mild conditions.
3. Many real world processes lend themselves to get modelled naturally by polynomial systems. In many applications even polynomial systems with low-order nonlinearities provide significant performance improvements over their linear counterparts.

The main thesis of this dissertation is the development of a new representation and consequently a new identification method for the Volterra filters. We know that the unit impulse response is insufficient to fully characterize a nonlinear system unlike linear time-invariant systems. In this dissertation we develop an exact identification method for Volterra filters using deterministic sequences consisting of impulses with distinct levels. Hence, we are implementing an identification method for finite-order, time invariant Volterra filters akin to the impulse response identification method of the linear, time-invariant systems. The developed method indeed includes identification using the unit impulse response as a subcase when the system under consideration is a linear system. To our best knowledge, this method is the first full-scale generalization of the impulse response representation to the finite order Volterra type nonlinear systems. There have been previous attempts in this direction by Schetzen (1965) for continuous-time Volterra filters. However, paraphrasing from Mathews and Sicuranza (2000), no practical applications have been devised for identifying higher-order polynomial systems using the method in Schetzen (1965), because of the inherent high complexity. Our method on the other hand first introduces a novel representation for Volterra filters, and then develops the identification method based on this representation. Hence, we overcome the high complexity problem which impeded the use of the algorithm in Schetzen (1965) for higher order systems. Our method is exact; it calculates the exact Volterra kernels in the absence of noise for very short length input sequences. Our method calculates each Volterra kernel individually. The kernel estimates are not utilized in the calculation of further kernel estimates. This hinders error propagation among estimates. Our method calculates directly the Volterra kernels, instead of calculating first some intermediary representation such as the Wiener kernels which do not have any directly interpretable results. 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. It is not rigorously proved in this dissertation, but our work on the persistence of excitation strongly suggests that our deterministic input sequences have the shortest length possible to exactly identify the finite-order Volterra kernels.

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 the 2D orthogonal lattice structure developed in Kayran (1996b).

Most of the material in this dissertation is novel. First section of Chapter 2 gives a brief overview of polynomial systems based on Volterra series, as can be found in references on this subject (Mathews and Sicuranza, 2000; Rugh, 1981; Schetzen, 1989). The novel structure in Chapter 6 is based on the work in Kayran (1996b). Hence, there is some partial overlap between this chapter and the results in Kayran (1996b). Other than these mentioned brief overlaps, all of the material in the dissertation is completely novel work.

The dissertation is organized as follows: Chapter 2 describes the novel partitioning of the Volterra kernels and proves the equivalence between this setup and the regular Volterra parameterizations. Chapter 3 describes the method to identify the Volterra kernel vectors using multilevel deterministic input signals. The algorithm is first described for 1-D kernel vectors, then for 2-D and 3-D kernel vectors and the necessary matrices are defined in a step-by-step fashion. Then, the algorithm is generalized to the identification of the ℓ-D kernel vectors. Chapter 4 discusses the persistence of excitation of the multilevel input sequence. The condition number of the matrices whose inverses appear in the algorithm and the problem of ill-conditioning are discussed. This chapter also includes simulation examples comparing the performance of the developed identification algorithm with Volterra filter identification methods present in the literature. Chapter 5 presents an application of the novel identification method presented in Chapter 3. The method is applied in the nonlinear communication channel setting. The deterministic input sequence is constructed using signals from common modulation constellations, such as QPSK. The identification method is adapted to the case of the bandpass Volterra channels and the performance of the algorithm is compared to competing methods. Chapter 6 presents a digression from the main theme of this dissertation outlined in the first five chapters. Chapter 6 introduces a novel fully orthogonal structure for the realization of second-order Volterra filters. This structure is based on the 2D orthogonal lattice structure. The developed realization method is utilized in the adaptive system identification setting with RLS adaptation. Simulation results are included to verify the superiority of the developed realization to structures from the literature, in the adaptive setting. Chapter 7 concludes the dissertation.

2. NOVEL VOLTERRA FILTER REPRESENTATION

2.1 VOLTERRA SERIES EXPANSION

In this section we will provide an overview of the Volterra series representation for nonlinear systems. We will outline the time-domain Volterra series and give some of the fundamental results which are important for the understanding and application of the Volterra series in engineering problems.

2.1.1 Time-Domain Volterra Series Expansion

A system without memory can in general be described by a suitable series expansion. The Taylor series is a well known example for series expansions without memory. For continuous-time systems it is given as

$$y(t) = c_0 + \sum_{k=1}^{\infty} c_k x^k(t) \tag{2.1}$$

and for discrete-time systems as

$$y(n) = c_0 + \sum_{k=1}^{\infty} c_k x^k(n) \tag{2.2}$$

If the system under consideration is not memoryless, it can still be described by using an extension of the Taylor expansion. 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, 1887, 1959). For a general continuous-time nonlinear system the input-output relationship is represented by the following infinite continuous-time Volterra series integral. This integral is doubly infinite, infinite both in the order of the nonlinearity and the length of the memory of the system.

$$
\begin{aligned}
y(t) = b_0 &+ \int_{-\infty}^{\infty} b_1(\tau_1)x(t-\tau_1)d\tau_1 \\
&+ \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} b_2(\tau_1,\tau_2)x(t-\tau_1)x(t-\tau_2)d\tau_1 d\tau_2 + \dots \\
&+ \int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} b_M(\tau_1,\tau_2,\dots,\tau_M)x(t-\tau_1)x(t-\tau_2)\cdots x(t-\tau_M)d\tau_1 d\tau_2\cdots d\tau_M \\
&+ \dots
\end{aligned}
\tag{2.3}
$$

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

$$
\begin{aligned}
y(n) = \bar{b}_0 &+ \sum_{i_1=-\infty}^{\infty} \bar{b}_1(i_1)x(n-i_1) \\
&+ \sum_{i_1=-\infty}^{\infty}\sum_{i_2=-\infty}^{\infty} \bar{b}_2(i_1,i_2)x(n-i_1)x(n-i_2) + \dots \\
&+ \sum_{i_1=-\infty}^{\infty}\cdots\sum_{i_M=-\infty}^{\infty} \bar{b}_M(i_1,i_2,\dots,i_M)x(n-i_1)x(n-i_2)\cdots x(n-i_M) \\
&+ \dots
\end{aligned}
\tag{2.4}
$$

The validity and applicability of the infinite and finite Volterra series expansions in the description of general nonlinear systems has been subject to much scrutiny (Boyd and Chua, 1985; Palm and Poggio, 1985; Sandberg, 1983). The Weierstrass approximation theorem is of primary importance in justifying the use of polynomial expansions in the input-output description of systems. The statement of Weierstrass's theorem is as follows (Mathews and Sicuranza, 2000):

Theorem 2.1 [Weierstrass] *Let $f(\lambda)$ be a continuous, real-valued function on the closed interval $[\lambda_1, \lambda_2]$. For any $\epsilon > 0$, there exists a real polynomial $\mathcal{P}(\lambda)$ such that $|f(\lambda) - \mathcal{P}(\lambda)| < \epsilon$ for all $\lambda \in (\lambda_1, \lambda_2)$.*

The Weierstrass theorem proposes that every continuous function defined on an interval $[\lambda_1, \lambda_2]$ can be uniformly approximated as closely as desired by a polynomial function. A generalization of this theorem is the Stone-Weierstrass theorem. The Stone-Weierstrass theorem extends the Weierstrass theorem to the realm of compact signal spaces and functional mappings instead of real-valued functions defined on some closed interval $[\lambda_1, \lambda_2]$ (Mathews and Sicuranza, 2000).

Using the Stone-Weierstrass theorem researchers have shown that a large class of nonlinear systems can be uniformly approximated by the Volterra series expansion on a ball of bounded inputs (Sandberg, 1992). This result is valid not only for the infinite Volterra series as given in (2.3) and (2.4), but it is also valid for the truncated Volterra series. Hence, especially from an empirical viewpoint, the truncated Volterra series is suitable for the modelling of a wide variety of nonlinearities encountered in real-life systems. As a result, truncated Volterra series has found applications in an encompassing range of problems, where linear systems failed to provide adequate performance (Cheng and Powers, 2001; Ching and Powers, 1993; Frank, 1995; Hermann, 1990; Marmarelis and Naka, 1972; Mathews and Sicuranza, 2000; Özden et al., 1996b, 1997, 1998; Shi and Hecox, 1991).

2.1.2 Truncated Volterra Series Sum

The truncated or doubly finite Volterra series is obtained by confining the infinite summations in (2.3) and (2.4) to sufficiently large but finite values. In this thesis we will be concerned with discrete-time, causal, finite-memory, time-invariant nonlinear systems described by the discrete-time, truncated Volterra series expansion as given below.

$$\begin{aligned} y(n) = \bar{b}_0 &+ \sum_{i_1=0}^{N} \bar{b}_1(i_1)x(n-i_1) \\ &+ \sum_{i_1=0}^{N}\sum_{i_2=0}^{N} \bar{b}_2(i_1,i_2)x(n-i_1)x(n-i_2) + \dots \\ &+ \sum_{i_1=0}^{N}\sum_{i_2=0}^{N}\cdots\sum_{i_M=0}^{N} \bar{b}_M(i_1,i_2,\dots,i_M)x(n-i_1)x(n-i_2)\cdots x(n-i_M) \end{aligned} \tag{2.5}$$

The m^{th} order term with m summations as incorporated in the Volterra series expansion of (2.5) includes $(N+1)^m$ coefficients, $m = 1, 2, \dots, M$. The representation in (2.5) assumes that for each different permutation of the indices $\{i_1, i_2, \dots, i_m\}$ there is a distinct coefficient in the Volterra series. However, all the different permutations of the same set of indices multiply the same term $(x(n-i_1)x(n-i_2)\cdots x(n-i_m))$. Hence, the Volterra series can actually be uniquely represented by a fewer number of coefficients. Therefore the generic kernel $\bar{b}_m(i_1, i_2, \ldots, i_m)$ can be replaced by a so called symmetric kernel $\bar{b}_{m,sym}(i_1, i_2, \ldots, i_m)$ to be defined as follows.

$$\bar{b}_{m,sym}(i_1, i_2, \ldots, i_m) = \frac{1}{|\pi(i_1, i_2, \ldots, i_m)|} \sum_{\pi(\cdot)} \bar{b}_m(i_{\pi(1)}, i_{\pi(2)}, \ldots, i_{\pi(m)}) \tag{2.6}$$

Here the summation is over all the distinct permutations $\pi(\cdot)$ of the given set of indices $\{i_1, i_2, \ldots, i_m\}$. $|\pi(i_1, i_2, \ldots, i_m)|$ defines the total number of these possible permutations (Mathews and Sicuranza, 2000, 33-35).

The redundancy of the coefficients in the generic Volterra series can be utilized to derive another equivalent form which will be more suitable for our studies. We can introduce new coefficients defined as

$$b_m(i_1, i_2, \ldots, i_m) = \sum_{\pi(\cdot)} \bar{b}_m(i_{\pi(1)}, i_{\pi(2)}, \ldots, i_{\pi(m)}) \tag{2.7}$$

The sum in (2.7) is over all the distinct permutations $\pi(\cdot)$ of the given set of indices $\{i_1, i_2, \ldots, i_m\}$. After this definition, the truncated Volterra series is given as

$$\begin{aligned} y(n) = b_0 &+ \sum_{i_1=0}^{N} b_1(i_1)x(n-i_1) \\ &+ \sum_{i_1=0}^{N} \sum_{i_2=i_1}^{N} b_2(i_1, i_2)x(n-i_1)x(n-i_2) + \ldots \\ &+ \sum_{i_1=0}^{N} \sum_{i_2=i_1}^{N} \cdots \sum_{i_M=i_{M-1}}^{N} b_M(i_1, i_2, \ldots, i_M)x(n-i_1)x(n-i_2)\cdots x(n-i_M) \end{aligned} \tag{2.8}$$

This representation is called as the triangular Volterra representation and the corresponding kernels are called as the triangular Volterra kernels in the literature (Mathews and Sicuranza, 2000). In this thesis we will be calling this representation as the Volterra filter. The corresponding kernels will be called as the triangular Volterra kernels or simply as the Volterra kernels to avoid confusion. The number of kernels in the redundant Volterra representation in (2.5) is

$$C_{\text{redundant}} = \sum_{m=0}^{M} (N+1)^m = \frac{(N+1)^{M+1} - 1}{N} \tag{2.9}$$

The number of kernels in the triangular Volterra representation in (2.8) is on the other hand given as

$$C_{\text{triangular}} = \sum_{m=0}^{M} \binom{N+m}{m} = \binom{N+M+1}{M} \tag{2.10}$$

Table 2.1: Total number of kernels for the redundant and triangular representations of a Volterra filter with $N = 8$ as a function of nonlinearity order M.

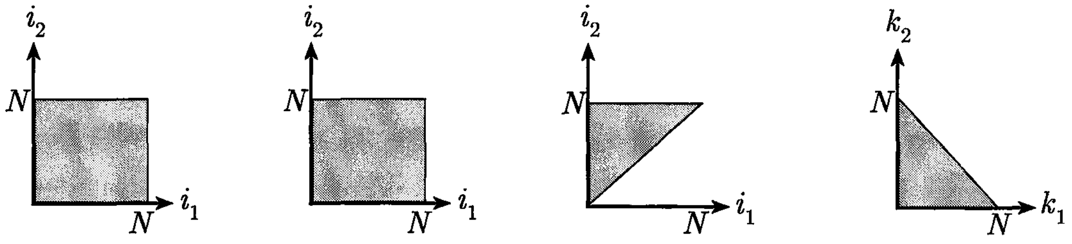
M	$C_{\text{redundant}}$	$C_{\text{triangular}}$
1	10	10
2	91	55
3	820	220
4	7381	715
5	66430	2002
6	597871	5005
7	5380840	11440
8	48427561	24310
9	435848050	48620
10	3922632451	92378

The saving in the number of kernels by using the symmetry property is measured by the ratio $C_{\text{redundant}}/C_{\text{triangular}}$. As $N \to \infty$, this ratio tends to $M!$ asymptotically. Hence, there is a significant reduction in the number of required kernels by using the triangular representation. Table 2.1 presents the distribution of the number of kernels for the redundant and triangular representations of a Volterra filter. The Voltarra filter under consideration has $N = 8$ and the number of kernels are shown as a function of the nonlinearity order $M = 1, 2, \ldots, 10$.

The kernel symmetry in the symmetric representation can be utilized in another way to develop the so called regular kernel representation $\bar{b}_{m,reg}(k_1, k_2, \ldots, k_m)$. The regular Volterra representation can be derived from the triangular representation via a variable exchange. When we substitute $k_m = i_m - i_{m-1}$ for $m = 2, 3, \ldots, M$ and $k_1 = i_1$ in (2.8), this equation becomes

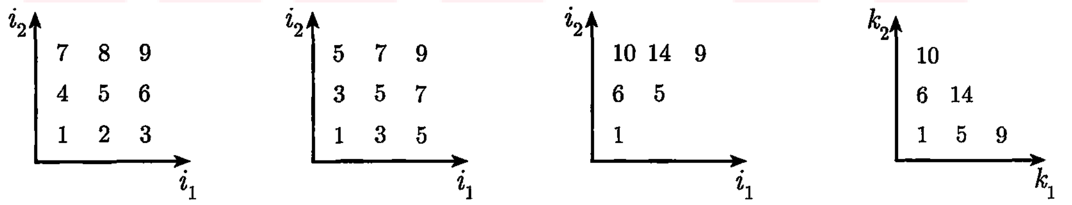
$$\begin{aligned} y(n) = \bar{b}_{0,reg} &+ \sum_{k_1=0}^{N} \bar{b}_{1,reg}(k_1) x(n-k_1) \\ &+ \sum_{k_1=0}^{N} \sum_{k_2=0}^{N-k_1} \bar{b}_{2,reg}(k_1, k_2) x(n-k_1) x(n-k_1-k_2) + \ldots \\ &+ \sum_{k_1=0}^{N} \sum_{k_2=0}^{N-k_1} \cdots \sum_{k_M=0}^{N-k_1-\cdots-k_{M-1}} \bar{b}_{M,reg}(k_1, k_2, \ldots, k_M) \\ &\qquad \times x(n-k_1) x(n-k_1-k_2) \cdots x(n-k_1-\cdots-k_M) \end{aligned} \tag{2.11}$$

Fig. 2.1 shows the different kernel supports for symmetric, triangular, and regular representations of the quadratic part of a Volterra filter. Fig. 2.2 provides an example for the relationship of the kernel values for the redundant, symmetric, triangular, and regular representations of the quadratic part of a Volterra filter. As we stated earlier, in this thesis we will consider the triangular representation as the standard representation for the Volterra filter. We will use the triangular representation as the starting point for our studies to develop novel Volterra representations.



Redundant	Symmetric	Triangular	Regular
$\bar{b}_2(i_1, i_2)$	$\bar{b}_{2,sym}(i_1, i_2)$	$b_2(i_1, i_2)$	$\bar{b}_{2,reg}(k_1, k_2)$
$0 \leq i_1 \leq N$	$0 \leq i_1 \leq N$	$0 \leq i_1 \leq N$	$0 \leq k_1 \leq N$
$0 \leq i_2 \leq N$	$0 \leq i_2 \leq N$	$i_1 \leq i_2 \leq N$	$0 \leq k_2 \leq N - k_1$

Figure 2.1: Different kernel support regions for the redundant, symmetric, triangular, and regular representations of the quadratic part of a Volterra filter.



Redundant	Symmetric	Triangular	Regular
$\bar{b}_2(i_1, i_2)$	$\bar{b}_{2,sym}(i_1, i_2)$	$b_2(i_1, i_2)$	$\bar{b}_{2,reg}(k_1, k_2)$
$0 \leq i_1 \leq 2$	$0 \leq i_1 \leq 2$	$0 \leq i_1 \leq 2$	$0 \leq k_1 \leq 2$
$0 \leq i_2 \leq 2$	$0 \leq i_2 \leq 2$	$i_1 \leq i_2 \leq 2$	$0 \leq k_2 \leq 2 - k_1$

Figure 2.2: An example showing kernel values of the redundant, symmetric, triangular, and regular representations for the quadratic part of a Volterra filter, $N = 2$.

The output $y(n)$ in (2.8) can be rewritten as below.

$$y(n) = \mathcal{N}[x(n)] = \sum_{k=0}^{M} y_k(n) \tag{2.12a}$$

in which

$$y_k(n) = \mathcal{B}_k[x(n)] \tag{2.12b}$$

$\mathcal{N}[\cdot]$ 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. The input-output relation for each of these subsystems is as indicated below.

$$\begin{aligned} y_k(n) &= \mathcal{B}_k[x(n)] \\ &= \sum_{i_1=0}^{N} \sum_{i_2=i_1}^{N} \cdots \sum_{i_k=i_{k-1}}^{N} b_k(i_1, i_2, \ldots, i_k) x(n-i_1) x(n-i_2) \cdots x(n-i_k) \end{aligned} \tag{2.12c}$$

Here, $b_k\,(i_1, i_2, \ldots, i_k)$ is the Volterra kernel of degree k. M is the order of the highest nonlinearity present in the system. The upper limit in the sums, N, is the memory length of the nonlinear system. In general form, this memory length can be chosen different for all the sums of different nonlinearity order. However, in the literature it has been customary to chose all the memory terms equivalent without loss of generality (Mathews and Sicuranza, 2000, p.26). The input-output relation as in (2.8) and (2.12), given by the truncated Volterra series sum is referred to in the literature as a Volterra filter, Volterra system or Volterra channel depending on the context and the kind of input signal used. The input-output relationship of the nonlinear Volterra filter $\mathcal{N}[\cdot]$ is shown pictorially in Fig. 2.3. The output of the nonlinear Volterra filter $\mathcal{N}$ can be decomposed in terms of the outputs of subsystems $\mathcal{B}_k$ (2.12). Fig. 2.4 depicts this decomposition.

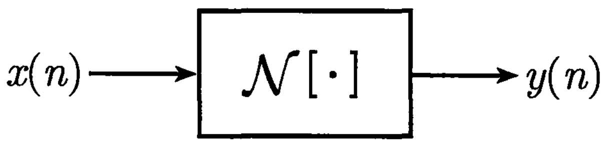


Figure 2.3: The input-output relationship of the nonlinear Volterra filter $\mathcal{N}[\,\cdot\,]$.

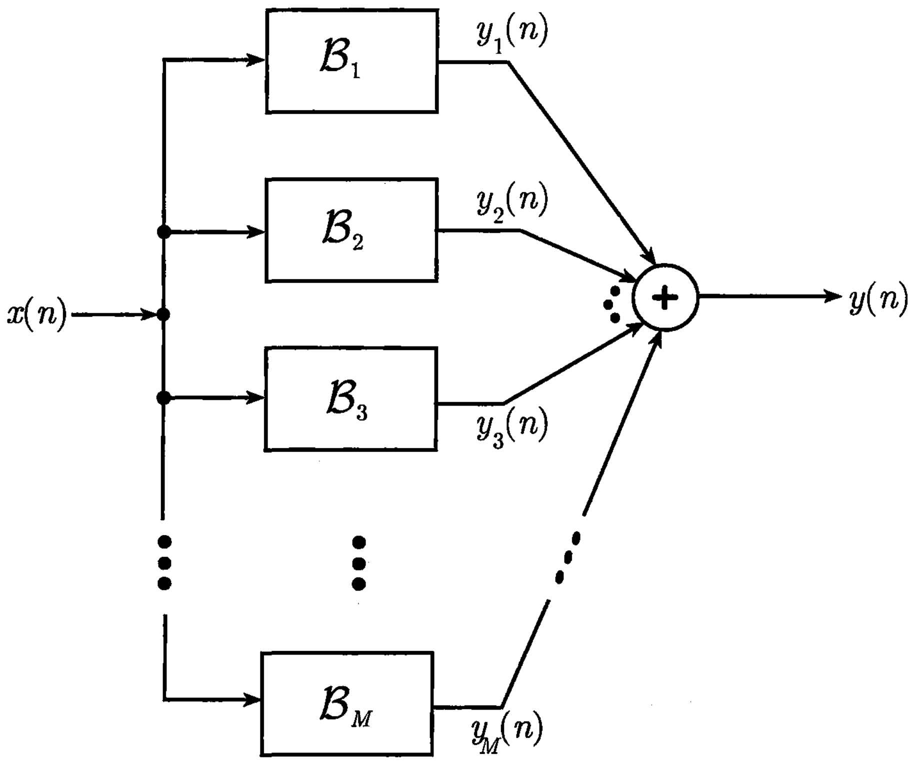


Figure 2.4: The nonlinear Volterra filter $\mathcal{N}$ as a sum of nonlinear subsystems $\mathcal{B}_k$.

2.2 NOVEL MULTIVARIATE KERNEL VECTOR REPRESENTATION FOR VOLTERRA FILTERS

In this section, the M^{th}-order discrete, causal, time-invariant Volterra system is reformulated in terms of the multivariate cross-term complexity of the Volterra kernels, 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.

2.2.1 Multivariate Kernel Vector Representation

The M^{th}-order nonlinear system, $\mathcal{N}[\,\cdot\,]$, under consideration can be modelled by the triangular representation as outlined in the previous section. The open form given in (2.8) can be rewritten in a more compact way as given below. From now on, the constant term b_0 in the Volterra series expansion is skipped, since it does not have any significance in the identification setting we are interested in.

$$\begin{aligned} y(n) = \mathcal{N}[x(n)] &= \sum_{k=1}^{M} y_k(n) = \sum_{k=1}^{M} \mathcal{B}_k[x(n)] \\ &= \sum_{k=1}^{M} \underbrace{\sum_{i_1=0}^{N} \sum_{i_2=i_1}^{N} \cdots \sum_{i_k=i_{k-1}}^{N} b_k\left(i_1, i_2, \ldots, i_k\right) x(n-i_1)x(n-i_2)\cdots x(n-i_k)}_{\mathcal{B}_k[x(n)]} \end{aligned} \tag{2.13}$$

Here N is the memory length of the nonlinear system and $b_k\left(i_1, i_2, \ldots, i_k\right)$ is the Volterra kernel of degree k. In this section 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, $\mathcal{H}^{(\ell)}$, $\ell = 1, \ldots, M$. Hence,

$$y(n) = \mathcal{N}[x(n)] = \sum_{\ell=1}^{M} y^{(\ell)}(n) \tag{2.14a}$$

in which

$$y^{(\ell)}(n) = \mathcal{H}^{(\ell)}[x(n)] \tag{2.14b}$$

Fig. 2.5 depicts the decomposition of the output of $\mathcal{N}$ in terms of the outputs of cross-term subsystems $\mathcal{H}^{(\ell)}$. The input-output relation for each of the subsystems $\mathcal{H}^{(\ell)}$, $\ell = 1, 2, \ldots, M$ is as indicated below.

$$\begin{aligned} \mathcal{H}^{(1)}[x(n)] &= \sum_{i=0}^{N} \mathbf{h}^{(1)^T}(i)\ \mathbf{x}^{(1)}(n-i) \\ \mathcal{H}^{(\ell)}[x(n)] &= \sum_{q_1=1}^{Q_1} \cdots \sum_{q_{\ell-1}=1}^{Q_{\ell-1}} \sum_{i=0}^{N-\bar{q}_{\ell-1}} \mathbf{h}^{(\ell)^T}(q_1, \ldots, q_{\ell-1}; i)\ \mathbf{x}^{(\ell)}(q_1, \ldots, q_{\ell-1}; n-i) \\ &\quad \text{for } 2 \leqslant \ell \leqslant M \end{aligned} \tag{2.15}$$

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

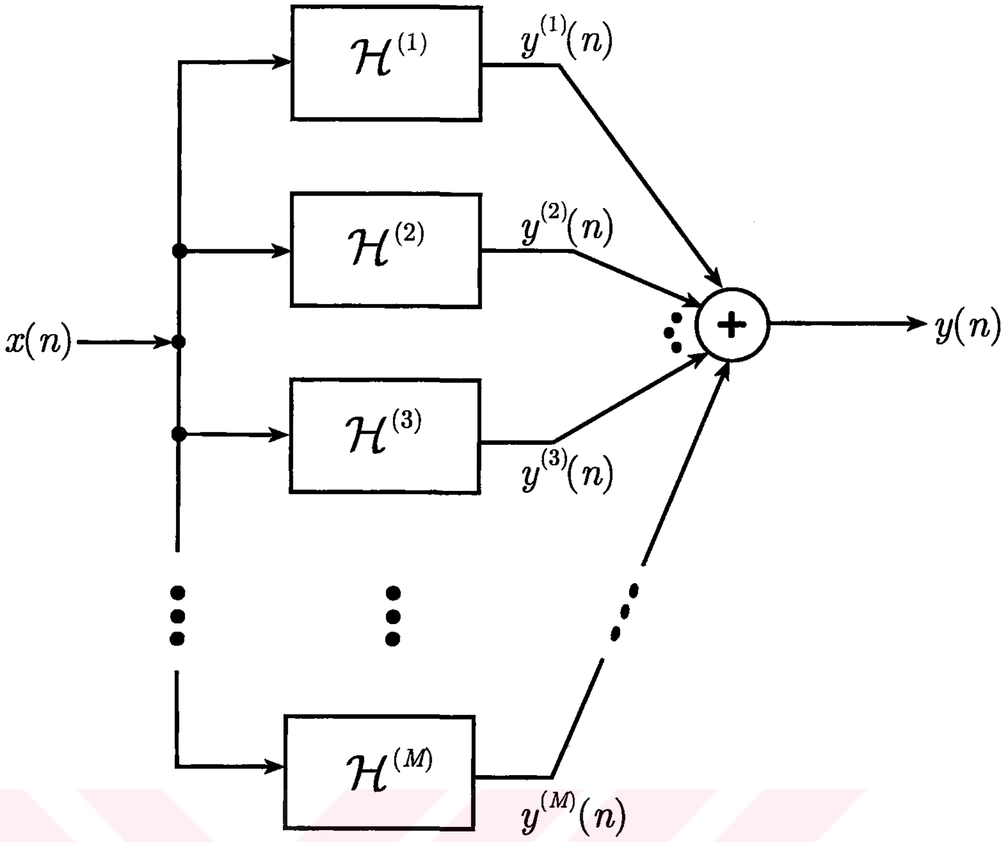


Figure 2.5: The novel decomposition for the nonlinear Volterra filter $\mathcal{N}$ as a sum of cross-term subsystems $\mathcal{H}^{(\ell)}$, $\ell = 1, \ldots, M$.

input signal $x(n)$, respectively.

$$\mathbf{h}^{(1)}(i) = \begin{bmatrix} h_1^{(1)}(i) \\ h_2^{(1)}(i) \\ \vdots \\ h_M^{(1)}(i) \end{bmatrix} = \begin{bmatrix} b_1(i) \\ b_2(i,i) \\ \vdots \\ b_M(i,\ldots,i) \end{bmatrix} \tag{2.16a}$$

$$\mathbf{x}^{(1)}(n) = \begin{bmatrix} x(n) \\ x^2(n) \\ \vdots \\ x^M(n) \end{bmatrix} \tag{2.16b}$$

The ℓ-D input vector in (2.15) can be expressed in the following form:

$$\mathbf{x}^{(\ell)}(q_1, \ldots, q_{\ell-1}; n) = \begin{bmatrix} x_\ell^{(\ell)}(q_1, \ldots, q_{\ell-1}; n) \\ \mathbf{x}_{\ell+1}^{(\ell)}(q_1, \ldots, q_{\ell-1}; n) \\ \vdots \\ \mathbf{x}_M^{(\ell)}(q_1, \ldots, q_{\ell-1}; n) \end{bmatrix} \tag{2.17a}$$

in which

$$x_\ell^{(\ell)}(q_1, \ldots, q_{\ell-1}; n) = x(n)\, x(n-q_1) \,\cdots\, x(n-q_1-\cdots-q_{\ell-1}) \tag{2.17b}$$

$$\mathbf{x}_k^{(\ell)}(q_1,\ldots,q_{\ell-1};n) \equiv \left[x_k^{(p_1,\cdots,p_\ell)}(q_1,\ldots,q_{\ell-1};n)\right]_{\sigma(p_1,\ldots,p_\ell)} \tag{2.17c}$$

The subinput vector $\mathbf{x}_k^{(\ell)}(q_1,\ldots,q_{\ell-1};n)$ for $k=\ell,\ell+1,\ldots,M$ in (2.17a) consists of all possible input products of degree k.

$$x_k^{(p_1,\cdots,p_\ell)}(q_1,\ldots,q_{\ell-1};n) = x^{p_1}(n)\,x^{p_2}(n-q_1)\,\cdots\,x^{p_\ell}(n-q_1-\cdots-q_{\ell-1}) \tag{2.18}$$

$\sigma(p_1,p_2,\ldots,p_\ell)$ defines all the possible combinations for p_i, $i=1,2,\ldots,\ell$. The condition imposed on p_i, $i=1,2,\ldots,\ell$ is, first that their sum is equal to k $\left(\sum_{i=1}^{\ell} p_i = k\right)$, and second that they are positive integers ($p_i>0$). Under these conditions there are a total of $\binom{k-1}{\ell-1} = \binom{k-1}{k-\ell}$ possible combinations. Hence, the length of the subinput vector $\mathbf{x}_k^{(\ell)}(q_1,\ldots,q_{\ell-1};n)$ in (2.17b) is

$$L_{k,\ell} = \binom{k-1}{\ell-1} \tag{2.19}$$

The overall ℓ-D input vector in (2.17a) has a total of

$$\sum_{k=\ell}^{M} L_{k,\ell} = \sum_{k=\ell}^{M}\binom{k-1}{\ell-1} = \binom{M}{\ell} \tag{2.20}$$

terms. Therefore, the length of the input vector $\mathbf{x}^{(\ell)}(q_1,\ldots,q_{\ell-1};n)$ in (2.17a) is

$$L_\ell = \binom{M}{\ell} \tag{2.21}$$

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

$$\mathbf{x}^{(3)}(q_1,q_2;n) = \begin{bmatrix} x_3^{(3)}(q_1,q_2;n) \\ \mathbf{x}_4^{(3)}(q_1,q_2;n) \\ \mathbf{x}_5^{(3)}(q_1,q_2;n) \end{bmatrix} \tag{2.22}$$

Here,

$$x_3^{(3)}(q_1,q_2;n) = x(n)\,x(n-q_1)\,x(n-q_1-q_2) \tag{2.23}$$

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

$$\mathbf{x}_4^{(3)}(q_1,q_2;n) = \begin{bmatrix} x^2(n)\,x(n-q_1)\,x(n-q_1-q_2) \\ x(n)\,x^2(n-q_1)\,x(n-q_1-q_2) \\ x(n)\,x(n-q_1)\,x^2(n-q_1-q_2) \end{bmatrix} \tag{2.24}$$

For $\mathbf{x}_5^{(3)}(q_1,q_2;n)$, all $\binom{4}{2} = 6$ possible combinations can be written as $\sigma(p_1,p_2,p_3) = \{(3,1,1)\ (2,2,1),\ (2,1,2),\ (1,3,1),\ (1,2,2),\ (1,1,3)\}$. Therefore,

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

$$\mathbf{x}_5^{(3)}(q_1, q_2; n) = \begin{bmatrix} x^3(n)\, x(n-q_1)\, x(n-q_1-q_2) \\ x^2(n)\, x^2(n-q_1)\, x(n-q_1-q_2) \\ x^2(n)\, x(n-q_1)\, x^2(n-q_1-q_2) \\ x(n)\, x^3(n-q_1)\, x(n-q_1-q_2) \\ x(n)\, x^2(n-q_1)\, x^2(n-q_1-q_2) \\ x(n)\, x(n-q_1)\, x^3(n-q_1-q_2) \end{bmatrix} \tag{2.25}$$

□

The ℓ-D kernel vectors in (2.15) can be written in terms of subkernels as

$$\mathbf{h}^{(\ell)}(q_1, \ldots, q_{\ell-1}; i) = \begin{bmatrix} h_\ell^{(\ell)}(q_1, \ldots, q_{\ell-1}; i) \\ \mathbf{h}_{\ell+1}^{(\ell)}(q_1, \ldots, q_{\ell-1}; i) \\ \vdots \\ \mathbf{h}_M^{(\ell)}(q_1, \ldots, q_{\ell-1}; i) \end{bmatrix} \tag{2.26a}$$

in which

$$h_\ell^{(\ell)}(q_1, \ldots, q_{\ell-1}; i) = h_\ell^{(1,1,\cdots,1)}(q_1, \ldots, q_{\ell-1}; i) \tag{2.26b}$$

and

$$\mathbf{h}_k^{(\ell)}(q_1, \ldots, q_{\ell-1}; i) \equiv \left[h_k^{(p_1,p_2,\ldots p_\ell)}(q_1, \ldots, q_{\ell-1}; i) \right]_{\sigma(p_1,p_2,\ldots,p_\ell)} \tag{2.26c}$$

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 (2.17c). $\mathbf{h}_k^{(\ell)}(q_1, \ldots, q_{\ell-1}; i)$ consists of all the Volterra kernels of degree k with ℓ cross-terms.

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

$$\mathbf{h}^{(3)}(q_1, q_2; i) = \begin{bmatrix} h_3^{(3)}(q_1, q_2; i) \\ \mathbf{h}_4^{(3)}(q_1, q_2; i) \\ \mathbf{h}_5^{(3)}(q_1, q_2; i) \end{bmatrix} \tag{2.27}$$

Here,

$$h_3^{(3)}(q_1, q_2; i) = h_3^{(1,1,1)}(q_1, q_2; i) \tag{2.28}$$

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

$$\mathbf{h}_4^{(3)}(q_1, q_2; i) = \begin{bmatrix} h_4^{(2,1,1)}(q_1, q_2; i) \\ h_4^{(1,2,1)}(q_1, q_2; i) \\ h_4^{(1,1,2)}(q_1, q_2; i) \end{bmatrix} \tag{2.29}$$

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

$$\mathbf{h}_5^{(3)}(q_1, q_2; i) = \begin{bmatrix} h_5^{(3,1,1)}(q_1, q_2; i) \\ h_5^{(2,2,1)}(q_1, q_2; i) \\ h_5^{(2,1,2)}(q_1, q_2; i) \\ h_5^{(1,3,1)}(q_1, q_2; i) \\ h_5^{(1,2,2)}(q_1, q_2; i) \\ h_5^{(1,1,3)}(q_1, q_2; i) \end{bmatrix} \tag{2.30}$$

□

2.2.2 Equivalence Between the Novel Representation and the Regular Representation

There exists an equivalent triangular Volterra kernel $b_k(i_1, i_2, \ldots, i_k)$ as given in (2.13) for each component of the subkernel vector $\mathbf{h}_k^{(\ell)}(q_1, \ldots, q_{\ell-1}; i)$ in (2.26), and the relationship between the triangular Volterra kernels and cross-term Volterra kernels is as given below.

$$h_k^{(p_1,p_2,\ldots,p_\ell)}(q_1, \ldots, q_{\ell-1}; i) = b_k(i_1, i_2, \ldots, i_k) \tag{2.31a}$$

where,

$$i_j = \begin{cases} i & \text{for} \quad 1 \leqslant j \leqslant p_1 \\ i + q_1 & \text{for} \quad p_1 + 1 \leqslant j \leqslant p_1 + p_2 \\ i + q_1 + q_2 & \text{for} \quad p_1 + p_2 + 1 \leqslant j \leqslant p_1 + p_2 + p_3 \\ \quad \vdots & \qquad\qquad \vdots \\ i + q_1 + \cdots + q_{\ell-1} & \text{for} \quad p_1 + \cdots + p_{\ell-1} + 1 \leqslant j \leqslant p_1 + \cdots + p_\ell \end{cases} \tag{2.31b}$$

(2.31) can be rewritten as,

$$h_k^{(p_1,p_2,\ldots,p_\ell)}(q_1, \ldots, q_{\ell-1}; i) = b_k(\underbrace{i, \ldots, i}_{p_1}, \underbrace{i + \bar{q}_1, \ldots, i + \bar{q}_1}_{p_2}, \ldots, \underbrace{i + \bar{q}_{\ell-1}, \ldots, i + \bar{q}_{\ell-1}}_{p_\ell}) \tag{2.32}$$

Example 2.3: We continue with Example 2.2. Using (2.31), we want to find the representations for the kernel vectors, but this time in terms of the Volterra kernels. Using (2.31), the kernel vectors for $\ell = 3$ and $M = 5$ can now be written as

$$h_3^{(3)}(q_1, q_2; i) = b_3(i, i + q_1, i + q_2) \tag{2.33}$$

$$\mathbf{h}_4^{(3)}(q_1,q_2;i) = \begin{bmatrix} h_4^{(2,1,1)}(q_1,q_2;i) \\ h_4^{(1,2,1)}(q_1,q_2;i) \\ h_4^{(1,1,2)}(q_1,q_2;i) \end{bmatrix} = \begin{bmatrix} b_4(i,i,i+q_1,i+q_2) \\ b_4(i,i+q_1,i+q_1,i+q_2) \\ b_4(i,i+q_1,i+q_2,i+q_2) \end{bmatrix} \tag{2.34}$$

$$\mathbf{h}_5^{(3)}(q_1,q_2;i) = \begin{bmatrix} h_5^{(3,1,1)}(q_1,q_2;i) \\ h_5^{(2,2,1)}(q_1,q_2;i) \\ h_5^{(2,1,2)}(q_1,q_2;i) \\ h_5^{(1,3,1)}(q_1,q_2;i) \\ h_5^{(1,2,2)}(q_1,q_2;i) \\ h_5^{(1,1,3)}(q_1,q_2;i) \end{bmatrix} = \begin{bmatrix} b_5(i,i,i,i+q_1,i+q_2) \\ b_5(i,i,i+q_1,i+q_1,i+q_2) \\ b_5(i,i,i+q_1,i+q_2,i+q_2) \\ b_5(i,i+q_1,i+q_1,i+q_1,i+q_2) \\ b_5(i,i+q_1,i+q_1,i+q_2,i+q_2) \\ b_5(i,i+q_1,i+q_2,i+q_2,i+q_2) \end{bmatrix} \tag{2.35}$$

□

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.

$$\begin{aligned} y(n) =& \sum_{i_1=0}^{2} b_1(i_1)x(n-i_1) + \sum_{i_1=0}^{2}\sum_{i_2=i_1}^{2} b_2(i_1,i_2)x(n-i_1)x(n-i_2) \\ &+ \sum_{i_1=0}^{2}\sum_{i_2=i_1}^{2}\sum_{i_3=i_2}^{2} b_3(i_1,i_2,i_3)x(n-i_1)x(n-i_2)x(n-i_3) \\ =& b_1(0)x(n) + b_1(1)x(n-1) + b_1(2)x(n-2) + b_2(0,0)x^2(n) \\ &+ b_2(1,1)x^2(n-1) + b_2(2,2)x^2(n-2) + b_2(0,1)x(n)x(n-1) \\ &+ b_2(1,2)x(n-1)x(n-2) + b_2(0,2)x(n)x(n-2) + b_3(0,0,0)x^3(n) \\ &+ b_3(1,1,1)x^3(n-1) + b_3(2,2,2)x^3(n-2) + b_3(0,0,1)x^2(n)x(n-1) \\ &+ b_3(1,1,2)x^2(n-1)x(n-2) + b_3(0,0,2)x^2(n)x(n-2) \\ &+ b_3(0,1,1)x(n)x^2(n-1) + b_3(1,2,2)x(n-1)x^2(n-2) \\ &+ b_3(0,2,2)x(n)x^2(n-2) + b_3(0,1,2)x(n)x(n-1)x(n-2) \end{aligned} \tag{2.36}$$

For this Volterra filter, using (2.14), the novel representation that we introduced will be given as follows.

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

Here,

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

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

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

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

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

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

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

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

$$
\mathbf{h}^{(3)}(1,1;0) = \begin{bmatrix} h_3^{(1,1,1)}(1,1;0) \end{bmatrix} = \begin{bmatrix} b_3(0,1,2) \end{bmatrix}
$$

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

□

Example 2.5: We consider a quadratic Volterra filter with $M = 2$ and $N = 3$. The usual (triangular) Volterra representation can be written in the following form.

$$\begin{aligned}
y(n) &= \sum_{i_1=0}^{3} b_1(i_1)x(n-i_1) + \sum_{i_1=0}^{3}\sum_{i_2=i_1}^{3} b_2(i_1,i_2)x(n-i_1)x(n-i_2) \\
&= b_1(0)x(n) + b_1(1)x(n-1) + b_1(2)x(n-2) + b_1(3)x(n-3) \\
&+ b_2(0,0)x^2(n) + b_2(1,1)x^2(n-1) + b_2(2,2)x^2(n-2) + b_2(3,3)x^2(n-3) \\
&+ b_2(0,1)x(n)x(n-1) + b_2(1,2)x(n-1)x(n-2) + b_2(2,3)x(n-2)x(n-3) \\
&+ b_2(0,2)x(n)x(n-2) + b_2(1,3)x(n-1)x(n-3) \\
&+ b_2(0,3)x(n)x(n-3)
\end{aligned} \tag{2.37}$$

For this Volterra filter, using (2.14) the novel representation becomes:

$$y(n) = \sum_{i=0}^{3} \mathbf{h}^{(1)^T}(i)\, \mathbf{x}^{(1)}(n-i) + \sum_{q_1=1}^{3}\sum_{i=0}^{3-q_1} \mathbf{h}^{(2)^T}(q_1;i)\mathbf{x}^{(2)}(q_1;n-i)$$

The cross-term kernels for this novel representation are determined in terms of the triangular Volterra kernels as follows.

$$\begin{array}{ll}
h_1^{(1)}(0) = b_1(0) & \\
h_1^{(1)}(1) = b_1(1) & h_2^{(1,1)}(1;0) = b_2(0,1) \\
h_1^{(1)}(2) = b_1(2) & h_2^{(1,1)}(1;1) = b_2(1,2) \\
h_1^{(1)}(3) = b_1(3) & h_2^{(1,1)}(1;2) = b_2(2,3) \\
h_2^{(2)}(0) = b_2(0,0) & h_2^{(1,1)}(2;0) = b_2(0,2) \\
h_2^{(2)}(1) = b_2(1,1) & h_2^{(1,1)}(2;1) = b_2(1,3) \\
h_2^{(2)}(2) = b_2(2,2) & h_2^{(1,1)}(3;0) = b_2(0,3) \\
h_2^{(2)}(3) = b_2(3,3) &
\end{array} \tag{2.38}$$

The relationship between the triangular Volterra kernels and cross-term Volterra kernels for this quadratic Volterra filter are depicted also in Fig. 2.6. This figure outlines the novel grouping of the Volterra kernels in terms of cross-term complexity rather than nonlinearity order. □

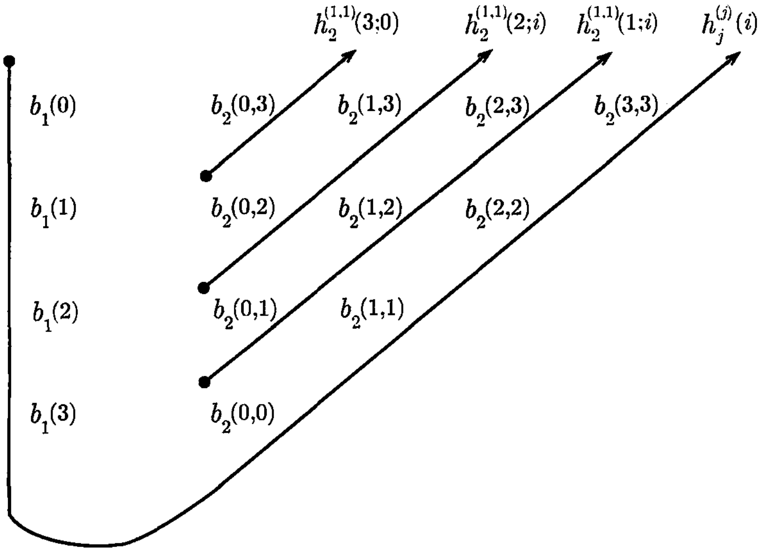


Figure 2.6: The relationship between the triangular Volterra kernels and cross-term Volterra kernels for a quadratic Volterra filter with $N = 3$ as given in Example 2.2.

2.2.3 Kernel Complexity for the Novel Volterra Filter Representation

The novel cross-product kernel representation presented in this section does not increase the number of kernels $C_{\text{triangular}}$ in the Volterra filter, where $C_{\text{triangular}}$ is the total number of kernels in (2.13). However, many identification procedures presented in recent years require high number of unnecessary kernel computations (Nowak and Van Veen, 1994b; Parker and Perry, 1981). Nowak and Van Veen (1994b) introduces tensor products for the Volterra filter representation of nonlinear systems. Tensor products generate product terms that are not required for the evaluation of the regular Volterra filter as given in (2.13). These redundant product terms have nonlinearity orders higher than M, which is the order of nonlinearity for the system under consideration. This representation with many redundant kernels was termed as an extended Volterra filter (Nowak and Van Veen, 1994b). Our representation completely avoids such redundancy.

To calculate the number of kernels in the novel cross-term representation we need the number of ℓ-D kernel vectors utilized for the realization of each ℓ-D subsystem $\mathcal{H}^{(\ell)}$. From the definition of the subsystems as given in (2.15), we can calculate that the total number of ℓ-D kernel vectors is

$$C_\ell = \binom{N+1}{\ell} \tag{2.39}$$

Using the length of the ℓ-D kernel vectors as calculated in (2.21) and the total number of ℓ-D kernel vectors as given in (2.39), the number of kernels in the novel cross-term representation can be evaluated as given below.

$$C_{\text{cross-term}} = \sum_{\ell=1}^{M} C_\ell L_\ell = \sum_{\ell=1}^{M} \binom{N+1}{\ell}\binom{M}{\ell} = \binom{N+M+1}{M} - 1 \tag{2.40}$$

Hence, the number of kernels in the novel cross-term representation is equivalent to the number of kernels in the triangular representation, i.e.,

$$C_{\text{cross-term}} = C_{\text{triangular}} = \binom{N+M+1}{M} - 1$$

For Example 2.1, it is seen that the number of kernel coefficients in both the cross-term and triangular representations are equal to $C_{\text{triangular}} = C_{\text{cross-term}} = \binom{N+M+1}{M} - 1 = \binom{2+3+1}{3} - 1 = 19$. On the other hand, in the case of Example 2.2, the number of kernel coefficients in both the cross-term and triangular representations are equal to $C_{\text{triangular}} = C_{\text{cross-term}} = \binom{N+M+1}{M} - 1 = \binom{3+2+1}{2} - 1 = 14$.

2.2.4 Novel Representation as a Sum of 1-D Convolutions

Another way to think about (2.15) is as a parallel combination of 1-D LTI filters. If we define new filters with impulse responses $\mathbf{h}^{(\ell)}_{(q_1,\ldots,q_{\ell-1})}(n) = \mathbf{h}^{(\ell)}(q_1,\ldots,q_{\ell-1};n)$ and new signals $\mathbf{x}^{(\ell)}_{(q_1,\ldots,q_{\ell-1})}(n) = \mathbf{x}^{(\ell)}(q_1,\ldots,q_{\ell-1};n)$, then (2.15) can be rewritten as,

$$\begin{aligned}
\mathcal{H}^{(1)}[x(n)] &= \mathbf{h}^{(1)^T}(n) * \mathbf{x}^{(1)}(n) \\
\mathcal{H}^{(\ell)}[x(n)] &= \sum_{q_1=1}^{Q_1} \cdots \sum_{q_{\ell-1}=1}^{Q_{\ell-1}} \mathbf{h}^{(\ell)^T}_{(q_1,\ldots,q_{\ell-1})}(n) * \mathbf{x}^{(\ell)}_{(q_1,\ldots,q_{\ell-1})}(n) \\
&\text{for } 2 \leqslant \ell \leqslant M
\end{aligned} \tag{2.41}$$

Here, "$*$" is the linear convolution operation, and $Q_j = N+1+j-\ell-\bar{q}_{j-1}$ for $j = 1,\ldots,\ell-1$, with $\bar{q}_m = q_1 + \cdots + q_m$ and $\bar{q}_0 = 0$. As we see in (2.41), the output of the subsystem $\mathcal{H}^{(\ell)}$, therefore the output of the overall Volterra filter can be expressed as a parallel combination of linear filters applied to nonlinear combinations of the input $x(n)$. This realization as a sum of linear filters is

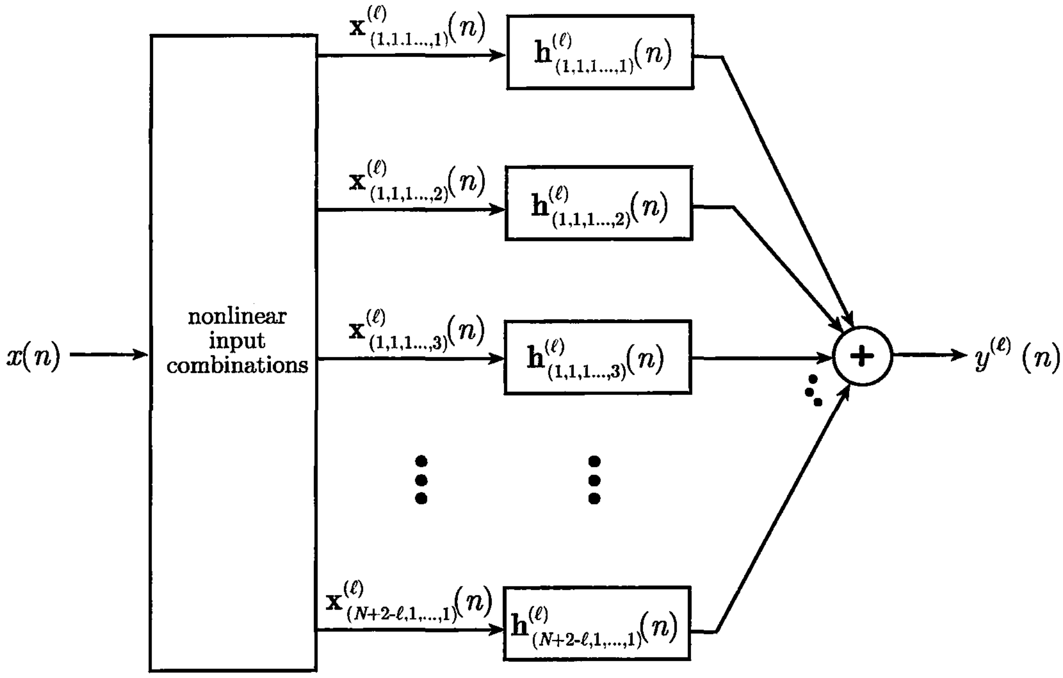


Figure 2.7: The subsystem $\mathcal{H}^{(\ell)}$ realized as a parallel combination of linear filters applied to nonlinear combinations of the input $x(n)$.

depicted in Fig. 2.7. From the definition of the ℓ-D subsystem $\mathcal{H}^{(\ell)}$ as given in (2.15), we can deduce that the total number of possible combinations for $\{q_1, q_2, \ldots, q_{\ell-1}\}$ is,

$$C_\ell^{\text{delay}} = \binom{N}{\ell - 1} \tag{2.42}$$

Hence, the number of linear filters in (2.41) utilized in the realization of the subsystem $\mathcal{H}^{(\ell)}$ is $\binom{N}{\ell-1}$.

A similar break-up of the Volterra system was proposed in Raz and Van Veen (1998) and Raz (1998). It is called as the *Diagonal Coordinate System* (DCS) for the discrete-time finite memory Volterra filters. In Raz and Van Veen (1998), the output of the k^{th}-order Volterra subsystem (as given in (2.12)) is expressed as a sum of 1-D convolutions, where the linear filter coefficients are obtained from the diagonals of the sampled hypercube defined by the k^{th}-order Volterra kernels. This representation in Raz and Van Veen (1998) parallels the one used for the identification of continuous-time Volterra filters given in Schetzen (1965). Our novel representation is similar to the DCS, in that the linear filter coefficients in DCS are given in a coordinate system analogous to our delay variables, $(q_1, \ldots, q_\ell)$. However, we group the linear filters differently, introducing the concept of delay-wise dimensionality and cross-term subsystem, rather than 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.

3. IDENTIFICATION OF THE KERNEL VECTORS USING MULTILEVEL INPUT SIGNALS

Nonlinear system identification is important due to the shortcoming of linear models when applied to inherently nonlinear problems which are abundant in real life applications (Cheng and Powers, 2001; Mathews and Sicuranza, 2000). Typical examples include trying to relate two signals whose significant spectral components do not overlap, satellite communication channels (Benedetto and Biglieri, 1983; Biglieri et al., 1988; Lazzarin et al., 1994; Özden et al., 1996b, 1998; Tseng and Powers, 1993) where amplifiers are operated near saturation, magnetic recording channels (Hermann, 1990; Özden et al., 1997) or biological systems (Marmarelis and Marmarelis, 1978; Marmarelis and Naka, 1972). The truncated (or "doubly finite") Volterra series representation constitutes an appealing nonlinear system model, since the output is linearly dependent on the kernel parameters, hence making the identification process mathematically tractable (Mathews and Sicuranza, 2000, p.29). Since Wiener introduced the use of the Volterra series for nonlinear modelling in engineering problems (Wiener, 1958), researchers have developed several methods for the estimation of the Volterra kernels. The most common class of Volterra system identification methods include cross-correlation methods based on random inputs (Billings, 1980; Rugh, 1981; Schetzen, 1989). In this class some effort has been directed towards Gaussian inputs because their higher order moments are tractable (Koukoulas and Kalouptsidis, 1995; Lee and Schetzen, 1965; Schetzen, 1974; Tick, 1961). However the results are non-trivial and prone to error due to their sequential nature where error in one kernel estimate propagates to the next. In similar vein use of PSK inputs which also have many vanishing moments has been considered (Zhou and Giannakis, 1997). With these sequences, the Volterra kernels are made orthogonal to each other and they are estimated separately. However, still the closed form results depend on higher order cumulants and hence necessitate relatively long input sequences.

Harmonic probing methods using multitone signals to determine frequency domain Volterra kernels have also been developed (Boyd et al., 1983; Chua and Liao, 1983). Another attractive approach is the use of pseudorandom multilevel sequences (PRMS) (Nowak and Van Veen, 1994b). Encouraged by the use of pseudorandom binary sequences (PRBS) for linear system identification, PRMS which can be chosen to be persistently exciting for any finite order Volterra system were considered for nonlinear system identification. However, as stated in Nowak and Van Veen (1994b) PRMS include input sequences which are redundant for the identification of a regular Volterra system. The condition on the order of the PRMS to guarantee persistence of excitation is necessary for the extended Volterra filter, which is introduced using Kronecker products (Nowak and Van Veen, 1994b). However, this condition is sufficient but not necessary for the regular Volterra filter and this introduces redundancy. The PRMS includes input sequences which are unnecessary when the system under consideration is a regular Volterra filter.

The use of deterministic signals in Volterra system identification has been first proposed in Schetzen (1965). It is viable to identify the kernels of a truncated Volterra filter using impulses at the input of the system. This approach was used in Schetzen (1965) for the continuous-time case and it was shown that a p^{th}-order kernel can be completely determined using p distinct impulses at the input of the system. This method was applied to the design of discrete-time two-dimensional (2-D) quadratic filters in Ramponi (1990). However, for systems with orders higher than two no practical algorithms or applications were devised, because of the high complexity of this approach (Mathews and Sicuranza, 2000, pp.38-39).

In this chapter, we focus on deterministic excitation sequences for the identification of nonlinear systems modelled using the truncated Volterra series representation. In the previous chapter we proposed a novel partitioning of the Volterra kernels. This representation will result in simple closed form solutions for the kernels when deterministic multilevel input sequences are used. Kernels are estimated separately hence error propagation is prevented. In this kernels are calculated using past outputs rather than using lower order kernel estimates. This provides robustness to noise and simplicity of implementation. Special matrices, which can be calculated recursively are utilized to form the input ensembles and to pick up the correct past outputs which should be subtracted. These matrices supply us with a closed form formula, and the high complexity problem which impeded the use of the algorithm in Schetzen (1965) for higher order systems is overcome. Our algorithm utilizes only the necessary input sequences and eliminates redundancy. The proposed identification algorithm is simulated for two Volterra systems of different orders and memory lengths, and its performance is compared with existing recent algorithms (Nowak and Van Veen, 1994a,b; Zhou and Giannakis, 1997).

In the coming sections, we derive an efficient algorithm to identify the kernel vectors $\mathbf{h}^{(\ell)}(q_1, \ldots, q_{\ell-1}; i)$ defined in (2.26) by using multilevel input sequences with ℓ distinct impulses. Starting to identify the kernel vectors with one cross-term ($\ell = 1$), which will be called as the 1-D kernel vectors, all other kernel vectors with increasing cross-term complexity can be determined by using only the present and the appropriate previous outputs of the lower dimensional subsystems given in (2.14). In order to explain this identification technique, we shall first discuss the procedure for determining Volterra kernels of one-, two-, and three-dimensional subsystems. The generalization of the procedure to the ℓ-D subsystems will then be given.

3.1 1-D KERNEL VECTORS AND THE ENSEMBLE INPUT MATRIX

3.1.1 Introduction

We restate (2.14) here to recall the novel representation we developed for the M^{th} order discrete Volterra filter.

$$y(n) = \mathcal{N}[x(n)] = \sum_{\ell=1}^{M} y^{(\ell)}(n) \tag{3.1a}$$

in which,

$$y^{(\ell)}(n) = \mathcal{H}^{(\ell)}[x(n)] \tag{3.1b}$$

$$
\begin{aligned}
\mathcal{H}^{(1)}[x(n)] &= \sum_{i=0}^{N} \mathbf{h}^{(1)^T}(i)\, \mathbf{x}^{(1)}(n-i) \\
\mathcal{H}^{(\ell)}[x(n)] &= \sum_{q_1=1}^{Q_1} \cdots \sum_{q_{\ell-1}=1}^{Q_{\ell-1}} \sum_{i=0}^{N-\bar{q}_{\ell-1}} \mathbf{h}^{(\ell)^T}(q_1,\ldots,q_{\ell-1};i)\, \mathbf{x}^{(\ell)}(q_1,\ldots,q_{\ell-1};n-i) \\
&\text{for } 2 \leqslant \ell \leqslant M \\
&Q_j = N+1+j-\ell-\bar{q}_{j-1} \text{ for } j=1,\ldots,\ell-1, \\
&\text{with } \bar{q}_m = q_1+\cdots+q_m \text{ and } \bar{q}_0 = 0
\end{aligned}
\tag{3.1c}
$$

Here, 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. The input-output relationship of the 1-D subsystem $\mathcal{H}^{(1)}[\,\cdot\,]$ can be written as in (2.15)

$$
y^{(1)}(n) = \mathcal{H}^{(1)}[x(n)] = \sum_{i=0}^{N} \mathbf{h}^{(1)^T}(i)\, \mathbf{x}^{(1)}(n-i) \tag{3.2}
$$

The 1-D kernel vector $\mathbf{h}^{(1)}(i)$ and 1-D the input vector $\mathbf{x}^{(1)}(n)$ were defined as follows.

$$
\mathbf{h}^{(1)}(i) = \begin{bmatrix} h_1^{(1)}(i) \\ h_2^{(2)}(i) \\ \vdots \\ h_M^{(M)}(i) \end{bmatrix} = \begin{bmatrix} b_1(i) \\ b_2(i,i) \\ \vdots \\ b_M(i,\ldots,i) \end{bmatrix} \tag{3.3a}
$$

$$
\mathbf{x}^{(1)}(n) = \begin{bmatrix} x(n) \\ x^2(n) \\ \vdots \\ x^M(n) \end{bmatrix} \tag{3.3b}
$$

As we remember from (2.17), the ℓ-D input vectors ($\ell = 2,\ldots,M$) in (3.1c) are defined in the following form:

$$
\mathbf{x}^{(\ell)}(q_1,\ldots,q_{\ell-1};n) = \begin{bmatrix} x_\ell^{(\ell)}(q_1,\ldots,q_{\ell-1};n) \\ \mathbf{x}_{\ell+1}^{(\ell)}(q_1,\ldots,q_{\ell-1};n) \\ \vdots \\ \mathbf{x}_M^{(\ell)}(q_1,\ldots,q_{\ell-1};n) \end{bmatrix} \tag{3.4a}
$$

in which

$$
x_\ell^{(\ell)}(q_1,\ldots,q_{\ell-1};n) = x(n)\, x(n-q_1) \cdots x(n-q_1-\cdots-q_{\ell-1}) \tag{3.4b}
$$

$$
\mathbf{x}_k^{(\ell)}(q_1,\ldots,q_{\ell-1};n) \equiv \Big[x_k^{(p_1,\cdots,p_\ell)}(q_1,\ldots,q_{\ell-1};n) \Big]_{\sigma(p_1,\ldots,p_\ell)} \tag{3.4c}
$$

The subinput vectors, $\mathbf{x}_k^{(\ell)}(q_1, \ldots, q_{\ell-1}; n)$ for $k = \ell, \ell+1, \ldots, M$, in (3.4c), consist of all possible input products of degree $\sum_{i=1}^{\ell} p_i = k$ with ℓ cross-terms.

$$x_k^{(p_1,\cdots,p_\ell)}(q_1, \ldots, q_{\ell-1}; n) = x^{p_1}(n)\, x^{p_2}(n-q_1)\, \cdots\, x^{p_\ell}(n - q_1 - \cdots - q_{\ell-1}) \quad (3.5)$$

Each cross-term in the above product has a different delay value varying from q_1 to $q_1 + q_2 + \ldots + q_{\ell-1}$.

Similarly, the corresponding ℓ-D kernel vectors in (3.1c) can be rewritten in terms of subkernels as,

$$\mathbf{h}^{(\ell)}(q_1, \ldots, q_{\ell-1}; i) = \begin{bmatrix} h_\ell^{(\ell)}(q_1, \ldots, q_{\ell-1}; i) \\ \mathbf{h}_{\ell+1}^{(\ell)}(q_1, \ldots, q_{\ell-1}; i) \\ \vdots \\ \mathbf{h}_M^{(\ell)}(q_1, \ldots, q_{\ell-1}; i) \end{bmatrix} \quad (3.6a)$$

in which,

$$\begin{aligned} \mathbf{h}_k^{(\ell)}(q_1, \ldots, q_{\ell-1}; i) &\equiv \left[h_k^{(p_1,p_2,\ldots p_\ell)}(q_1, \ldots, q_{\ell-1}; i) \right]_{\sigma(p_1,p_2,\ldots,p_\ell)} \\ &= \left[b_k(\underbrace{i, \ldots, i}_{p_1}, \underbrace{i+\bar{q}_1, \ldots, i+\bar{q}_1}_{p_2}, \ldots, \underbrace{i+\bar{q}_{\ell-1}, \ldots, i+\bar{q}_{\ell-1}}_{p_\ell}) \right]_{\sigma(p_1,p_2,\ldots,p_\ell)} \end{aligned} \quad (3.6b)$$

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 (3.4c).

3.1.2 Identification of 1-D Kernel Vectors

It is possible to show that multilevel single impulses, $x^{(1)}(m_1; n) = a_{m_1}\delta(n)$, for $m_1 = 1, 2, \ldots, \binom{M}{1}$ can be used to obtain the 1-D kernel vectors in (3.3a). Using the cross-term representation in (3.1), it is trivial to prove that the higher dimensional outputs are zero for these multilevel single impulses, i.e.,

$$\begin{aligned} \mathcal{N}[a_{m_1}\delta(n)] &= \mathcal{H}^{(1)}[a_{m_1}\delta(n)] \\ \mathcal{H}^{(\ell)}[a_{m_1}\delta(n)] &= 0, \ \text{for } \ell = 2, \ldots, M \end{aligned} \quad (3.7)$$

This relation in (3.7) is shown pictorially in Fig. 3.1. We can deduce the relation in (3.7) from the input vectors $\mathbf{x}^{(\ell)}(q_1, \ldots, q_{\ell-1}; n)$ defined in (3.4). When we form the input vectors $\mathbf{x}^{(\ell)}(q_1, \ldots, q_{\ell-1}; n)$, $\ell = 1, \ldots, M$ for an input signal with

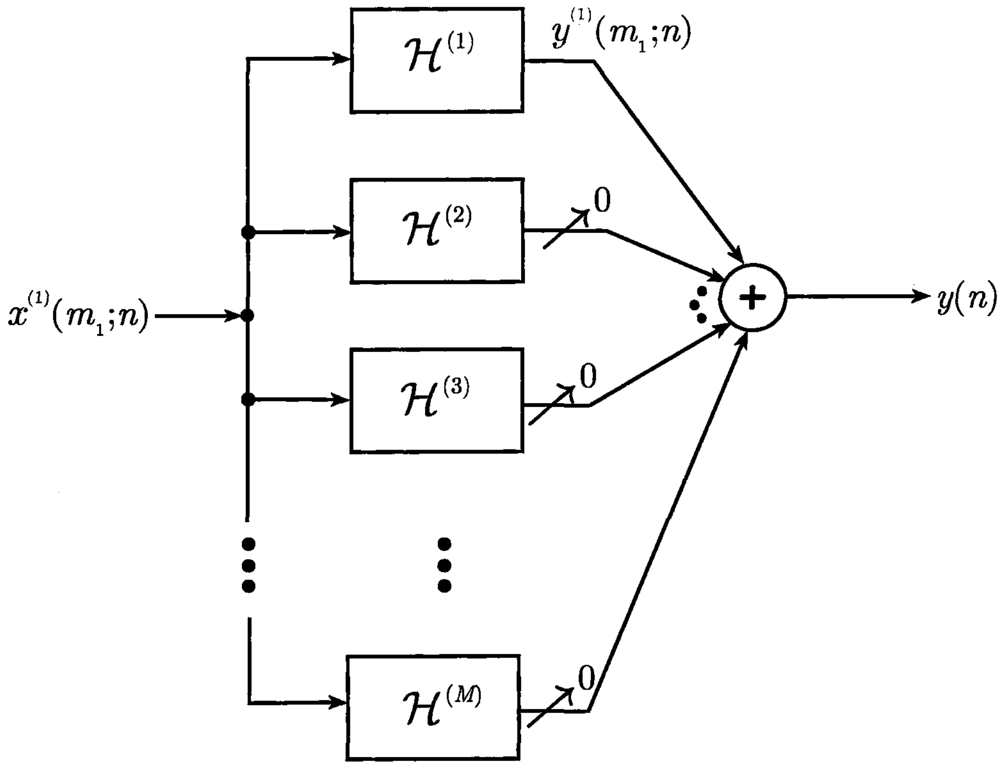


Figure 3.1: Pictorial description for (3.7). The output of the nonlinear system $\mathcal{N}$ for $x(n) = a_{m_1}\delta(n)$ is equal to the output of the subsystem $\mathcal{H}^{(1)}$.

a single impulse, we see that all the input vectors for $\ell > 1$ are exactly zero vectors.

$$\begin{aligned} &\mathbf{x}^{(\ell)}(q_1, \ldots, q_{\ell-1}; n) = \mathbf{0}, \text{ for } \ell = 2, \ldots, M \\ &\quad \text{when } x(n) = x^{(1)}(m_1; n) = a_{m_1}\delta(n) \end{aligned} \tag{3.8}$$

The reason for this is that, these input vectors consist of products of cross-terms with different delays in the form as given in (3.5). For a single impulse signal, these products will always come out as zero for $\ell = 2, 3, \ldots M$. Hence, the output of the M^{th}-order nonlinear system, when the input is $x(n) = x^{(1)}(m_1; n)$, is given by

$$\begin{aligned} y(n) =& \mathcal{N}\left[x^{(1)}(m_1; n)\right] = y(m_1; n) \\ =& \mathcal{H}^{(1)}\left[x^{(1)}(m_1; n)\right] = y^{(1)}(m_1; n) \\ =& \sum_{i=0}^{N} \mathbf{h}^{(1)^T}(i)\mathbf{u}^{(1)}(m_1; n-i) \end{aligned} \tag{3.9a}$$

$$\mathbf{u}^{(1)}(m_1; n) \equiv \begin{bmatrix} x^{(1)}(m_1; n) \\ x^{(1)^2}(m_1; n) \\ \vdots \\ x^{(1)^M}(m_1; n) \end{bmatrix} = \begin{bmatrix} a_{m_1} \\ a_{m_1}^2 \\ \vdots \\ a_{m_1}^M \end{bmatrix} \delta(n) \tag{3.9b}$$

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:

$$\mathbf{y}_e^{(1)}(n) = \mathcal{N}\big[\mathbf{x}_e^{(1)}(n)\big] = \mathcal{H}^{(1)}\big[\mathbf{x}_e^{(1)}(n)\big] = \sum_{i=0}^{N} \mathbf{U}_e^{(1)}(n-i)\,\mathbf{h}^{(1)}(i) \tag{3.10}$$

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

$$\mathbf{x}_e^{(1)}(n) \equiv \begin{bmatrix} x^{(1)}(1;n) \\ x^{(1)}(2;n) \\ \vdots \\ x^{(1)}(M;n) \end{bmatrix} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_M \end{bmatrix} \delta(n) \tag{3.11}$$

$$\mathbf{y}_e^{(1)}(n) \equiv \begin{bmatrix} \mathcal{N}\left[x^{(1)}(1;n)\right] \\ \mathcal{N}\left[x^{(1)}(2;n)\right] \\ \vdots \\ \mathcal{N}\left[x^{(1)}(M;n)\right] \end{bmatrix} = \begin{bmatrix} y^{(1)}(1;n) \\ y^{(1)}(2;n) \\ \vdots \\ y^{(1)}(M;n) \end{bmatrix} \tag{3.12}$$

$$\mathbf{U}_e^{(1)}(n) \equiv \begin{bmatrix} \mathbf{u}^{(1)^T}(1;n) \\ \mathbf{u}^{(1)^T}(2;n) \\ \vdots \\ \mathbf{u}^{(1)^T}(M;n) \end{bmatrix} \tag{3.13}$$

Here, $\mathbf{u}^{(1)}(m_1;n)$ is as defined in (3.9b). Therefore, $\mathbf{U}_e^{(1)}(n-i)$ in (3.10) can be replaced with $\mathbf{U}_e^{(1)}(n-i) = \mathbf{U}_e^{(1)}\delta(n-i)$, and the matrix $\mathbf{U}_e^{(1)}$ is written as

$$\mathbf{U}_e^{(1)} = \begin{bmatrix} a_1 & a_1^2 & \cdots & a_1^M \\ a_2 & a_2^2 & \cdots & a_2^M \\ \vdots & \vdots & \ddots & \vdots \\ a_M & a_M^2 & \cdots & a_M^M \end{bmatrix} \tag{3.14}$$

Hence, we get

$$\mathbf{y}_e^{(1)}(n) = \sum_{i=0}^{N} \mathbf{U}_e^{(1)}\,\mathbf{h}^{(1)}(i)\,\delta(n-i) = \mathbf{U}_e^{(1)}\,\mathbf{h}^{(1)}(n) \tag{3.15}$$

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

$$\mathbf{h}^{(1)}(n) = \left[\mathbf{U}_e^{(1)}\right]^{-1} \mathbf{y}_e^{(1)}(n), \quad \text{for } n = 0, 1, \ldots, N \tag{3.16}$$

Fig. 3.2 depicts the identification method for 1-D kernels as outlined in this section and finalized in (3.16). From (3.16), it is possible to write the solution

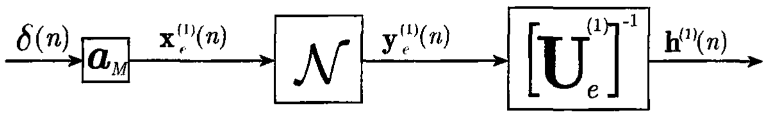


Figure 3.2: Method used for identification of 1-D Volterra kernels, $\mathbf{h}^{(1)}(n)$.

for all 1-D kernel vectors together in the matrix form as follows:

$$\mathbf{H}^{(1)} = \left[\mathbf{U}_e^{(1)}\right]^{-1}\mathbf{Y}_e^{(1)} \tag{3.17}$$

Here,

$$\begin{aligned}
\mathbf{H}^{(1)} &= \left[\mathbf{h}^{(1)}(0) \quad \mathbf{h}^{(1)}(1) \quad \cdots \quad \mathbf{h}^{(1)}(N)\right] \\
\mathbf{Y}_e^{(1)} &= \left[\mathbf{y}_e^{(1)}(0) \quad \mathbf{y}_e^{(1)}(1) \quad \cdots \quad \mathbf{y}_e^{(1)}(N)\right]
\end{aligned}$$

This result shows that all $\big(M(N+1)\big)$ 1-D kernels $h_j^{(1)}(i)$ can be determined by using only the inverse of an $M \times M$ ensemble matrix $\mathbf{U}_e^{(1)}$ times the ensemble output matrix $\mathbf{Y}_e^{(1)}$ as shown in (3.17). Note that the linear FIR filter identification via the impulse response is covered by this method as the special case $M = 1$.

3.1.3 Condition for Distinctness of the Input Signal Levels

In order to guarantee the Vandermonde like input matrix $\mathbf{U}_e^{(1)}$ in (3.14) 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 $a_i \neq a_j$, $\forall\ i \neq j$. In the following sections we will deal with the inverses of input matrices for the higher order cross-terms, which have a form similar to $\mathbf{U}_e^{(1)}$. The invertibility of the input matrix $\mathbf{U}_e^{(1)}$ is sufficient to assure the invertibility of the higher order input matrices. However, in addition to the nonsingularity constraint, we have to take into consideration the condition number of these input matrices $\mathbf{U}_e^{(\ell)}$, because these condition numbers will determine the effect of the observation noise on the accuracy of the kernel estimates. Hence, when we have the freedom to choose the impulse amplitudes, the values of the parameter vector, $\boldsymbol{a}_M \triangleq \left[a_1 \quad a_2 \quad \cdots \quad a_M\right]^T$ will be determined to optimize a collective criterion for the condition numbers of $\mathbf{U}_e^{(\ell)}$ for $\ell = 1, 2, \ldots, M$.

Example 3.1: Consider as an example the nonlinear system with $M = 3$ and $N = 2$, given by

$$\begin{aligned} y(n) = & \; x(n) + 2x^2(n) + 3x^3(n) \\ & - x(n-1) - 2x^2(n-1) - 3x^3(n-1) \\ & + 4x(n-2) + 5x^2(n-2) + 6x^3(n-2) \\ & - 4x(n)x(n-1) - 5x^2(n)x(n-1) - 6x(n)x^2(n-1) \\ & + 7x(n-1)x(n-2) + 8x^2(n-1)x(n-2) + 9x(n-1)x^2(n-2) \\ & - 7x(n)x(n-2) - 8x^2(n)x(n-2) - 9x(n)x^2(n-2) \\ & + 10x(n)x(n-1)x(n-2); \end{aligned} \tag{3.18}$$

For this section, we are interested in calculating the 1-D kernel vectors, i.e.

$$\mathbf{h}^{(1)}(0) = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} ; \quad \mathbf{h}^{(1)}(1) = \begin{bmatrix} -1 \\ -2 \\ -3 \end{bmatrix} ; \quad \mathbf{h}^{(1)}(2) = \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix}$$

To this end we will employ the 1-D input ensemble as in (3.11) and the corresponding output ensemble. We choose the impulse amplitudes as $a_1 = 1$, $a_2 = -1$ and $a_3 = 2$ without loss of generality, since the only condition is that the amplitudes should be distinct and nonzero. Hence, the 1-D input ensemble to the system is given as

$$\mathbf{x}_e^{(1)}(n) = \begin{bmatrix} 1 \\ -1 \\ 2 \end{bmatrix} \delta(n) \tag{3.19}$$

The output ensemble produced by the Volterra filter for this input ensemble is determined as

$$\mathbf{y}_e^{(1)}(n) = \mathbf{v}_e^{(1,1)}(n) = \begin{bmatrix} 6 \\ -2 \\ 34 \end{bmatrix} \delta(n) + \begin{bmatrix} -6 \\ 2 \\ -34 \end{bmatrix} \delta(n-1) + \begin{bmatrix} 15 \\ -5 \\ 76 \end{bmatrix} \delta(n-2) \tag{3.20}$$

From (3.14) and (3.19), the matrix $\mathbf{U}_e^{(1)}$ is written as,

$$\mathbf{U}_e^{(1)} = \begin{bmatrix} 1 & 1 & 1 \\ -1 & 1 & -1 \\ 2 & 4 & 8 \end{bmatrix} \tag{3.21}$$

Applying (3.16) for (3.20) and (3.21) yields,

$$
\begin{aligned}
\mathbf{h}^{(1)}(0) &= \left[\mathbf{U}_e^{(1)}\right]^{-1} \begin{bmatrix} 6 \\ -2 \\ 34 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \\
\mathbf{h}^{(1)}(1) &= \left[\mathbf{U}_e^{(1)}\right]^{-1} \begin{bmatrix} -6 \\ 2 \\ -34 \end{bmatrix} = \begin{bmatrix} -1 \\ -2 \\ -3 \end{bmatrix} \\
\mathbf{h}^{(1)}(2) &= \left[\mathbf{U}_e^{(1)}\right]^{-1} \begin{bmatrix} 15 \\ -5 \\ 76 \end{bmatrix} = \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix}
\end{aligned}
\tag{3.22}
$$

□

3.2 2-D KERNEL VECTORS AND ENSEMBLE INPUT FORMATION MATRICES

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

$$
\begin{aligned}
x^{(2)}\big((m_1, m_2), q_1; n\big) &= x^{(1)}(m_1; n) + x^{(1)}(m_2; n - q_1) \\
&\qquad \text{for } m_1 = 1, \ldots, M-1;\ m_2 = m_1 + 1, \ldots, M
\end{aligned}
\tag{3.23}
$$

where $x^{(1)}(m_i; n) = a_{m_i}\delta(n),\ i = 1, 2$ and the amplitudes are assigned by using the parameters of the 1-D input ensemble sequences, $a_{m_1}, a_{m_2} \in \{a_1, \ldots, a_M\}$.

It is possible to show that the 2-D input signal in (3.23) does not excite the Volterra kernels having more than two cross-terms, i.e.,

$$
\mathcal{H}^{(\ell)}\left[x^{(2)}\big((m_1; m_2), q_1; n\big)\right] = 0 \text{ for } \ell \geqslant 3. \tag{3.24}
$$

(3.24) is depicted in Fig. 3.3.

When we form the input vectors $\mathbf{x}^{(\ell)}(q_1, \ldots, q_{\ell-1}; n)$, $\ell = 1, \ldots, M$ for an input signal with two distinct impulses, we see that all the input vectors for $\ell > 2$ are exactly zero vectors.

$$
\begin{aligned}
\mathbf{x}^{(\ell)}(q_1, \ldots, q_{\ell-1}; n) &= \mathbf{0}, \text{ for } \ell = 3, \ldots, M \\
\text{when } x(n) &= x^{(2)}\big((m_1, m_2), q_1; n\big)
\end{aligned}
\tag{3.25}
$$

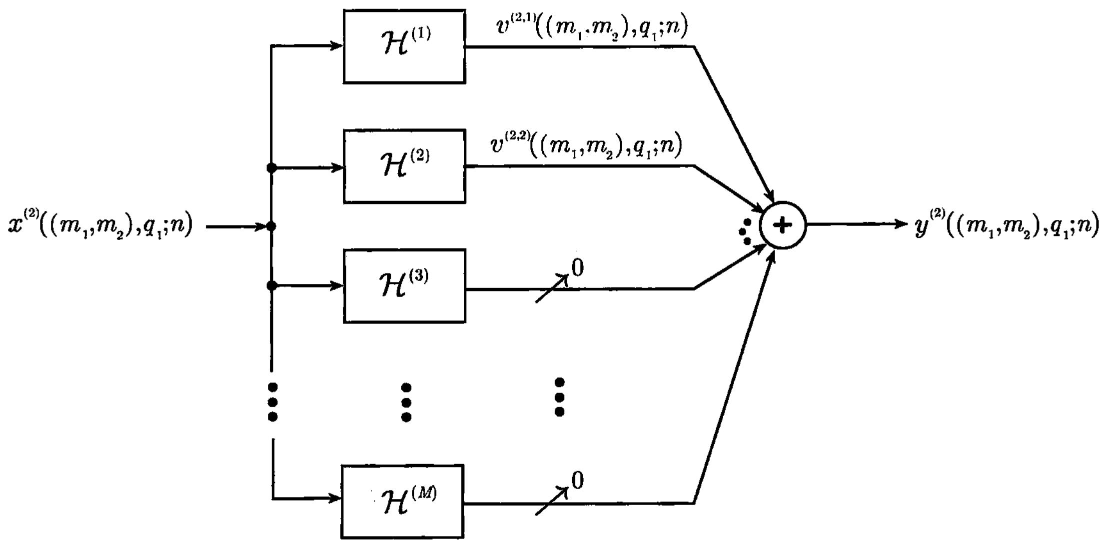


Figure 3.3: Pictorial description for (3.24). The output of the nonlinear system $\mathcal{N}$, for $x(n) = a_{m_1}\delta(n) + a_{m_2}\delta(n-q_1)$, is equal to the sum of the outputs of subsystems $\mathcal{H}^{(1)}$ and $\mathcal{H}^{(2)}$.

The reason for this is that, these input vectors consist of products of three or more cross-terms as given in (3.4c). For a double impulse signal, these products will always come out as zero for $\ell = 3, 4, \ldots M$.

Hence, the output for the input in (3.23) can be written as

$$\begin{aligned}\mathcal{N}\Big[x^{(2)}\big((m_1,m_2),q_1;n\big)\Big] &= y^{(2)}\big((m_1,m_2),q_1;n\big)\\ &=\mathcal{H}^{(1)}\Big[x^{(2)}\big((m_1,m_2),q_1;n\big)\Big] + \mathcal{H}^{(2)}\Big[x^{(2)}\big((m_1,m_2),q_1;n\big)\Big]\end{aligned} \tag{3.26}$$

where

$$\mathcal{H}^{(2)}\left[x^{(2)}\big((m_1,m_2),q_1;n\big)\right] = v^{(2,2)}\big((m_1,m_2),q_1;n\big) \tag{3.27}$$

and

$$\begin{aligned}\mathcal{H}^{(1)}\left[x^{(2)}\big((m_1,m_2),q_1;n\big)\right] &= v^{(2,1)}\big((m_1,m_2),q_1;n\big)\\ &= v^{(1,1)}(m_1;n) + v^{(1,1)}(m_2;n-q_1)\end{aligned} \tag{3.28}$$

with

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

Remark 3.1: The notation $v^{(i,j)}\big((m_1,\ldots,m_i),q_1,\ldots,q_{i-1};n\big)$ will denote the output of the j-D subsystem for an input signal consisting of i distinct impulses,

$$v^{(i,j)}\big((m_1,\ldots,m_i),q_1,\ldots,q_{i-1};n\big) = \mathcal{H}^{(j)}\left[x^{(i)}\big((m_1,\ldots,m_i),q_1,\ldots,q_{i-1};n\big)\right] \tag{3.29}$$

The output of the 2-D subsystem can be obtained from (3.26), (3.27) and (3.28) as

$$v^{(2,2)}\big((m_1,m_2),q_1;n\big) = \mathcal{N}\Big[x^{(2)}\big((m_1,m_2),q_1;n\big)\Big] - \Big[v^{(1,1)}(m_1;n) + v^{(1,1)}(m_2;n-q_1)\Big] \quad (3.30)$$

3.2.1 Ensemble Input Formation Matrices

(3.30) can be verified by direct substitution in (3.1). It is also interesting to note that the output of the 2-D subsystem is obtained by subtracting the appropriate previously computed 1-D outputs from the overall nonlinear system output. Since the 2-D kernel vectors have $\binom{M}{2}$ components, we must use a total of $\binom{M}{2}$ distinct ensemble inputs as given in (3.23). The main idea of the proposed method is based upon the observation of the outputs for these particular 2-D ensemble inputs which are given in the matrix form as follows:

$$\mathbf{x}_e^{(2)}(q_1;n) = \mathbf{T}_{2,1}^{(M)}\mathbf{x}_e^{(1)}(n) + \mathbf{T}_{2,2}^{(M)}\mathbf{x}_e^{(1)}(n-q_1) \quad (3.31)$$

where $\mathbf{x}_e^{(1)}(n) = \begin{bmatrix} a_1 & a_2 & \cdots & a_M \end{bmatrix}^T \delta(n)$ and 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. A recursive algorithm is given in Table A.1 for constructing these $\mathbf{T}_{\ell,k}^{(M)}$ matrices for $\ell = 1, 2, \ldots, M$ and $k = 1, 2, \ldots, \ell$. Each row of these ensemble input formation matrices has only one nonzero unity element. Similar to the single-input single-output case in (3.26), using the 2-D ensemble input vector in (3.31), we can determine the corresponding output vectors of 1-D and 2-D subsystems, $\mathbf{v}_e^{(2,1)}(q_1;n) = \mathcal{H}^{(1)}\Big[\mathbf{x}_e^{(2)}(q_1;n)\Big]$ and $\mathbf{v}_e^{(2,2)}(q_1;n) = \mathcal{H}^{(2)}\Big[\mathbf{x}_e^{(2)}(q_1;n)\Big]$, respectively.

Remark 3.2: 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.

$$\mathbf{v}_e^{(i,j)}(q_1,\ldots,q_{i-1};n) = \mathcal{H}^{(j)}\Big[\mathbf{x}_e^{(i)}(q_1,\ldots,q_{i-1};n)\Big] \quad (3.32)$$

Fig. 3.4 shows this definition.

Using (3.26) and (3.31), it is not difficult to show that the response of the 1-D system to the 2-D input ensemble can be decomposed in terms of the 1-D

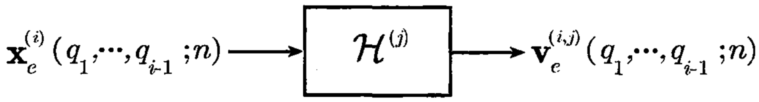


Figure 3.4: The definition for $\mathbf{v}_e^{(i,j)}(q_1,\ldots,q_{i-1};n)$ as given in Remark 3.2.

responses as

$$\begin{aligned}\mathbf{v}_e^{(2,1)}(q_1;n) &= \mathcal{H}^{(1)}\left[\mathbf{T}_{2,1}^{(M)}\mathbf{x}_e^{(1)}(n)\right] + \mathcal{H}^{(1)}\left[\mathbf{T}_{2,2}^{(M)}\mathbf{x}_e^{(1)}(n-q_1)\right] \\ &= \mathbf{T}_{2,1}^{(M)}\mathbf{v}_e^{(1,1)}(n) + \mathbf{T}_{2,2}^{(M)}\mathbf{v}_e^{(1,1)}(n-q_1)\end{aligned} \tag{3.33}$$

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

$$\begin{aligned}\mathbf{v}_e^{(2,2)}(q_1;n) &= \mathbf{y}_e^{(2)}(q_1;n) - \mathbf{v}_e^{(2,1)}(q_1;n) \\ &= \mathbf{y}_e^{(2)}(q_1;n) - \left[\mathbf{T}_{2,1}^{(M)}\mathbf{v}_e^{(1,1)}(n) + \mathbf{T}_{2,2}^{(M)}\mathbf{v}_e^{(1,1)}(n-q_1)\right]\end{aligned} \tag{3.34}$$

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

$$\mathbf{v}_e^{(2,2)}(q_1;n) = \sum_{i=0}^{N-q_1} \mathbf{U}_e^{(2)}(q_1;n-i)\mathbf{h}^{(2)}(q_1;i) \tag{3.35}$$

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, $a_1,\ a_2,\ldots,a_M$.

$$\mathbf{U}_e^{(2)} = \begin{bmatrix} \mathbf{u}^{(2)^T}(1,1) \\ \mathbf{u}^{(2)^T}(1,2) \\ \vdots \\ \mathbf{u}^{(2)^T}(M-1,M) \end{bmatrix} \tag{3.36a}$$

where

$$\mathbf{u}^{(2)^T}(i,j) = \left[u_2^{(2)}(i,j) \,\vdots\, \mathbf{u}_3^{(2)^T}(i,j) \,\vdots\, \cdots \,\vdots\, \mathbf{u}_M^{(2)^T}(i,j)\right] \tag{3.36b}$$

with

$$\begin{aligned} u_2^{(2)}(i,j) &= [a_i a_j]_{1\times 1} \\ \mathbf{u}_3^{(2)^T}(i,j) &= \left[a_i a_j^2 \quad a_i^2 a_j\right]_{1\times 2} \\ \mathbf{u}_4^{(2)^T}(i,j) &= \left[a_i a_j^3 \quad a_i^2 a_j^2 \quad a_i^3 a_j\right]_{1\times 3} \\ &\ \vdots \\ \mathbf{u}_M^{(2)^T}(i,j) &= \left[a_i a_j^{M-1} \quad a_i^2 a_j^{M-2} \quad \cdots \quad a_i^{M-1} a_j\right]_{1\times M} \end{aligned} \tag{3.36c}$$

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

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

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

The 2-D Volterra kernel vectors are obtained by using (3.34) and (3.35),

$$\mathbf{h}^{(2)}(q_1; n - q_1) = \left[\mathbf{U}_e^{(2)}\right]^{-1} \mathbf{v}_e^{(2,2)}(q_1; n) \tag{3.37}$$

for $q_1 = 1, \ldots, N$ and $n = q_1, q_1 + 1, \ldots, N$, provided the inverse exists. Fig. 3.5 depicts the identification method for 2-D kernels as outlined in this section and in (3.37).

Example 3.2: We continue with the identification problem we started in Example 3.1. We consider again the nonlinear system given in (3.18). For this section, the problem is to determine the 2-D kernel vectors

$$\mathbf{h}^{(2)}(1;0) = \begin{bmatrix} -4 \\ -5 \\ -6 \end{bmatrix} \; ; \quad \mathbf{h}^{(2)}(1;1) = \begin{bmatrix} 7 \\ 8 \\ 9 \end{bmatrix} \; ; \quad \mathbf{h}^{(2)}(2;0) = \begin{bmatrix} -7 \\ -8 \\ -9 \end{bmatrix}$$

Using the 1-D input ensemble vector $\mathbf{x}_e^{(1)}(n)$ in (3.19) and Table A.1, we can construct the ensemble input formation matrices $\mathbf{T}_{2,1}^{(3)}$ and $\mathbf{T}_{2,2}^{(3)}$

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

The detailed derivation of these particular matrices is given as an example after Table A.1. Having found these matrices, the first 2-D input ensemble in (3.31) for $q_1 = 1$ can be written as,

$$\begin{aligned} \mathbf{x}_e^{(2)}(1;n) &= \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \\ 2 \end{bmatrix} \delta(n) + \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \\ 2 \end{bmatrix} \delta(n-1) \\ &= \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix} \delta(n) + \begin{bmatrix} -1 \\ 2 \\ 2 \end{bmatrix} \delta(n-1) \end{aligned} \tag{3.38}$$

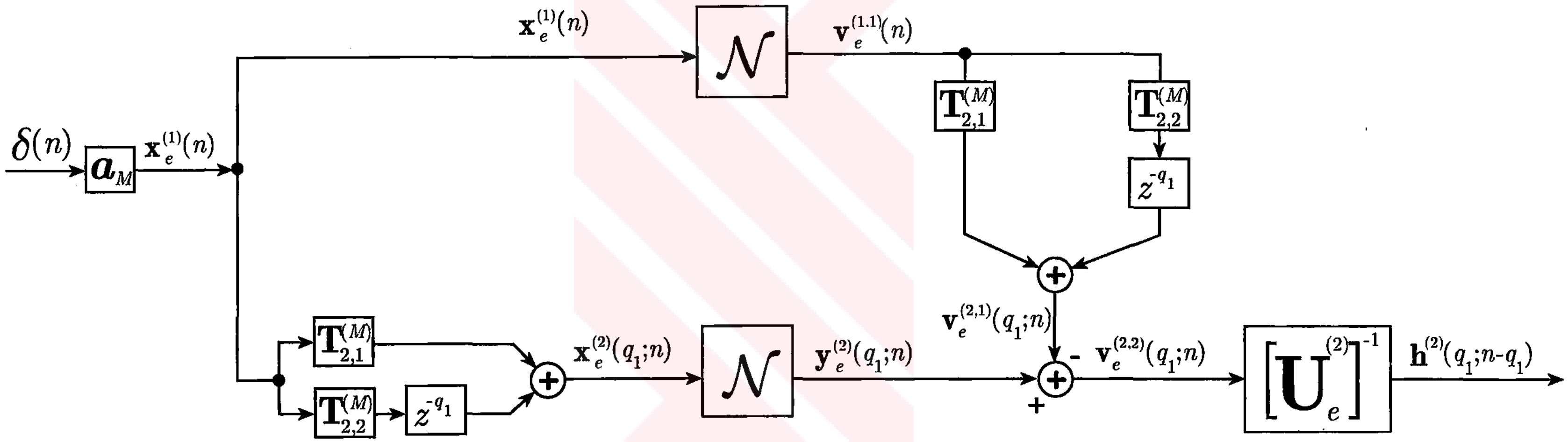


Figure 3.5: Method used for identification of 2-D Volterra kernels, $\mathbf{h}^{(2)}(q_1;n)$.

For this particular 2-D input ensemble in (3.38), the output of the nonlinear system in (3.18) is calculated as

$$\mathbf{y}_e^{(2)}(1;n) = \begin{bmatrix} 6 \\ 6 \\ -2 \end{bmatrix} \delta(n) + \begin{bmatrix} -3 \\ -12 \\ 52 \end{bmatrix} \delta(n-1) + \begin{bmatrix} 9 \\ 45 \\ -67 \end{bmatrix} \delta(n-2) + \begin{bmatrix} -5 \\ 76 \\ 76 \end{bmatrix} \delta(n-3) \quad (3.39)$$

It is possible to determine the response of the 1-D subsystem to the 2-D input ensemble in (3.38). This is accomplished by applying (3.33) to the 1-D output ensemble given in (3.20) as follows:

$$\begin{aligned}
\mathbf{v}_e^{(2,1)}(1;n) &= \mathbf{T}_{2,1}^{(3)}\mathbf{y}_e^{(1)}(n) + \mathbf{T}_{2,2}^{(3)}\mathbf{y}_e^{(1)}(n-1) \\
&= \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \left(\begin{bmatrix} 6 \\ -2 \\ 34 \end{bmatrix} \delta(n) + \begin{bmatrix} -6 \\ 2 \\ -34 \end{bmatrix} \delta(n-1) + \begin{bmatrix} 15 \\ -5 \\ 76 \end{bmatrix} \delta(n-2) \right) \\
&+ \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \left(\begin{bmatrix} 6 \\ -2 \\ 34 \end{bmatrix} \delta(n-1) + \begin{bmatrix} -6 \\ 2 \\ -34 \end{bmatrix} \delta(n-2) + \begin{bmatrix} 15 \\ -5 \\ 76 \end{bmatrix} \delta(n-3) \right) \\
&= \left(\begin{bmatrix} 6 \\ 6 \\ -2 \end{bmatrix} \delta(n) + \begin{bmatrix} -6 \\ -6 \\ 2 \end{bmatrix} \delta(n-1) + \begin{bmatrix} 15 \\ 15 \\ -5 \end{bmatrix} \delta(n-2) \right) \\
&+ \left(\begin{bmatrix} -2 \\ 34 \\ 34 \end{bmatrix} \delta(n-1) + \begin{bmatrix} 2 \\ -34 \\ -34 \end{bmatrix} \delta(n-2) + \begin{bmatrix} -5 \\ 76 \\ 76 \end{bmatrix} \delta(n-3) \right)
\end{aligned}$$

$$\mathbf{v}_e^{(2,1)}(1;n) = \begin{bmatrix} 6 \\ 6 \\ -2 \end{bmatrix} \delta(n) + \begin{bmatrix} -8 \\ 28 \\ 36 \end{bmatrix} \delta(n-1) + \begin{bmatrix} 17 \\ -19 \\ -39 \end{bmatrix} \delta(n-2) + \begin{bmatrix} -5 \\ 76 \\ 76 \end{bmatrix} \delta(n-3) \quad (3.40)$$

Then, the response of the 2-D subsystem can be obtained by subtracting $\mathbf{v}_e^{(2,1)}(1;n)$ from the nonlinear system output $\mathbf{y}_e^{(2)}(1;n)$ in (3.39)

$$\begin{aligned}
\mathbf{v}_e^{(2,2)}(1;n) &= \mathbf{y}_e^{(2)}(1;n) - \mathbf{v}_e^{(2,1)}(1;n) \\
&= \begin{bmatrix} 5 \\ -40 \\ 16 \end{bmatrix} \delta(n-1) + \begin{bmatrix} -8 \\ 64 \\ -28 \end{bmatrix} \delta(n-2)
\end{aligned} \quad (3.41)$$

The desired 2-D Volterra kernel vectors $\mathbf{h}^{(2)}(1;i)$ can be calculated by substituting (3.41) into (3.37)

$$\begin{aligned}
\mathbf{h}^{(2)}(1;0) &= \left[\mathbf{U}_e^{(2)}\right]^{-1}\mathbf{v}_e^{(2,2)}(1;1) \\
&= \begin{bmatrix} -1 & 1 & -1 \\ 2 & 4 & 2 \\ -2 & -4 & 2 \end{bmatrix}^{-1} \begin{bmatrix} 5 \\ -40 \\ 16 \end{bmatrix} = \begin{bmatrix} -4 \\ -5 \\ -6 \end{bmatrix}
\end{aligned} \quad (3.42)$$

$$\begin{aligned}\mathbf{h}^{(2)}(1;1) &= \left[\mathbf{U}_e^{(2)}\right]^{-1}\mathbf{v}_e^{(2,2)}(1;2)\\ &= \begin{bmatrix}-1 & 1 & -1\\ 2 & 4 & 2\\ -2 & -4 & 2\end{bmatrix}^{-1}\begin{bmatrix}-8\\ 64\\ -28\end{bmatrix} = \begin{bmatrix}7\\ 8\\ 9\end{bmatrix}\\ \mathbf{h}^{(2)}(1;\ell) &= \mathbf{0} \quad \ell \geq 2\end{aligned} \tag{3.43}$$

Here, the constant matrix $\mathbf{U}_e^{(2)}$ is obtained from $\mathbf{x}_e^{(1)}(n)$ and (3.36).

The second required 2-D input ensemble for $q_1 = 2$ can be found by using (3.31)

$$\begin{aligned}\mathbf{x}_e^{(2)}(2;n) &= \begin{bmatrix}1 & 0 & 0\\ 1 & 0 & 0\\ 0 & 1 & 0\end{bmatrix}\begin{bmatrix}1\\ -1\\ 2\end{bmatrix}\delta(n) + \begin{bmatrix}0 & 1 & 0\\ 0 & 0 & 1\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}1\\ -1\\ 2\end{bmatrix}\delta(n-2)\\ &= \begin{bmatrix}1\\ 1\\ -1\end{bmatrix}\delta(n) + \begin{bmatrix}0\\ 0\\ 0\end{bmatrix}\delta(n-1) + \begin{bmatrix}-1\\ 2\\ 2\end{bmatrix}\delta(n-2)\end{aligned} \tag{3.44}$$

For this particular 2-D input ensemble in (3.44), the output of the nonlinear system in (3.18) is calculated as

$$\begin{aligned}\mathbf{y}_e^{(2)}(2;n) = &\begin{bmatrix}6\\ 6\\ -2\end{bmatrix}\delta(n) + \begin{bmatrix}-6\\ -6\\ 2\end{bmatrix}\delta(n-1)\\ &+ \begin{bmatrix}21\\ -15\\ 57\end{bmatrix}\delta(n-2) + \begin{bmatrix}2\\ -34\\ -34\end{bmatrix}\delta(n-3) + \begin{bmatrix}-5\\ 76\\ 76\end{bmatrix}\delta(n-4)\end{aligned} \tag{3.45}$$

It is possible to determine the response of the 1-D subsystem to the 2-D input ensemble in (3.44). This is accomplished by applying (3.33) to the 1-D output ensemble given in (3.20) as follows:

$$\begin{aligned}\mathbf{v}_e^{(2,1)}(2;n) &= \mathbf{T}_{2,1}^{(3)}\mathbf{y}_e^{(1)}(n) + \mathbf{T}_{2,2}^{(3)}\mathbf{y}_e^{(1)}(n-2)\\ &= \begin{bmatrix}1 & 0 & 0\\ 1 & 0 & 0\\ 0 & 1 & 0\end{bmatrix}\left(\begin{bmatrix}6\\ -2\\ 34\end{bmatrix}\delta(n) + \begin{bmatrix}-6\\ 2\\ -34\end{bmatrix}\delta(n-1) + \begin{bmatrix}15\\ -5\\ 76\end{bmatrix}\delta(n-2)\right)\\ &+ \begin{bmatrix}0 & 1 & 0\\ 0 & 0 & 1\\ 0 & 0 & 1\end{bmatrix}\left(\begin{bmatrix}6\\ -2\\ 34\end{bmatrix}\delta(n-2) + \begin{bmatrix}-6\\ 2\\ -34\end{bmatrix}\delta(n-3) + \begin{bmatrix}15\\ -5\\ 76\end{bmatrix}\delta(n-4)\right)\\ &= \left(\begin{bmatrix}6\\ 6\\ -2\end{bmatrix}\delta(n) + \begin{bmatrix}-6\\ -6\\ 2\end{bmatrix}\delta(n-1) + \begin{bmatrix}15\\ 15\\ -5\end{bmatrix}\delta(n-2)\right)\\ &+ \left(\begin{bmatrix}-2\\ 34\\ 34\end{bmatrix}\delta(n-2) + \begin{bmatrix}2\\ -34\\ -34\end{bmatrix}\delta(n-3) + \begin{bmatrix}-5\\ 76\\ 76\end{bmatrix}\delta(n-4)\right)\end{aligned}$$

$$\mathbf{v}_e^{(2,1)}(2;n) = \begin{bmatrix} 6 \\ 6 \\ -2 \end{bmatrix} \delta(n) + \begin{bmatrix} -6 \\ -6 \\ 2 \end{bmatrix} \delta(n-1)$$

$$+ \begin{bmatrix} 13 \\ 49 \\ 29 \end{bmatrix} \delta(n-2) + \begin{bmatrix} 2 \\ -34 \\ -34 \end{bmatrix} \delta(n-3) + \begin{bmatrix} -5 \\ 76 \\ 76 \end{bmatrix} \delta(n-4) \quad (3.46)$$

Then, the response of the 2-D subsystem can be obtained by subtracting $\mathbf{v}_e^{(2,1)}(1;n)$ from the nonlinear system output $\mathbf{y}_e^{(2)}(1;n)$ in (3.45)

$$\begin{aligned} \mathbf{v}_e^{(2,2)}(2;n) &= \mathbf{y}_e^{(2)}(2;n) - \mathbf{v}_e^{(2,1)}(2;n) \\ &= \begin{bmatrix} 8 \\ -64 \\ 28 \end{bmatrix} \delta(n-2) \end{aligned} \quad (3.47)$$

The desired 2-D Volterra kernel vectors $\mathbf{h}^{(2)}(1;i)$ can be calculated by substituting (3.41) into (3.37)

$$\begin{aligned} \mathbf{h}^{(2)}(2;0) &= \left[\mathbf{U}_e^{(2)}\right]^{-1} \mathbf{v}_e^{(2,2)}(2;2) \\ &= \begin{bmatrix} -1 & 1 & -1 \\ 2 & 4 & 2 \\ -2 & -4 & 2 \end{bmatrix}^{-1} \begin{bmatrix} 8 \\ -64 \\ 28 \end{bmatrix} = \begin{bmatrix} -7 \\ -8 \\ -9 \end{bmatrix} \end{aligned} \quad (3.48)$$

Hence, we have determined all the 2-D kernel vectors, namely $\mathbf{h}^{(2)}(1;0)$, $\mathbf{h}^{(2)}(1;1)$ and $\mathbf{h}^{(2)}(2;0)$. □

3.3 3-D KERNEL VECTORS AND OUTPUT PICK-UP MATRICES

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

$$\mathbf{x}_e^{(3)}(q_1, q_2; n) = \mathbf{T}_{3,1}^{(M)} \mathbf{x}_e^{(1)}(n) + \mathbf{T}_{3,2}^{(M)} \mathbf{x}_e^{(1)}(n-q_2) + \mathbf{T}_{3,3}^{(M)} \mathbf{x}_e^{(1)}(n-q_1-q_2) \quad (3.49)$$

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. The output of the nonlinear

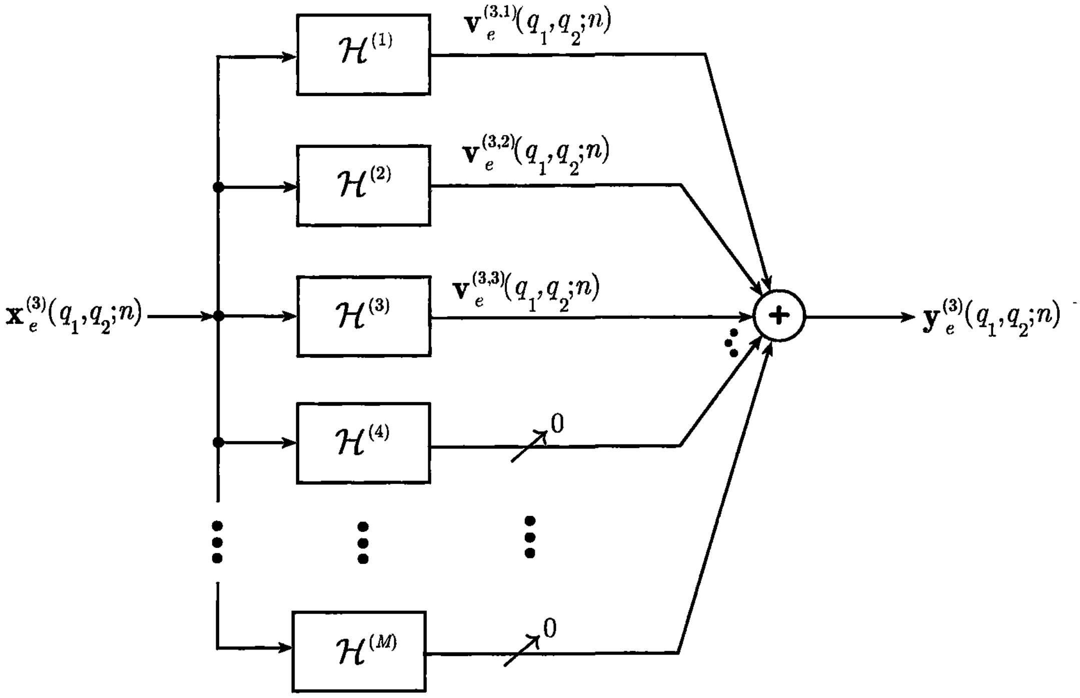


Figure 3.6: Pictorial description for (3.50). The output of the nonlinear system $\mathcal{N}$ for the 3-D input ensemble $\mathbf{x}_e^{(3)}(q_1, q_2; n)$, is equal to the sum of the outputs of subsystems $\mathcal{H}^{(1)}$, $\mathcal{H}^{(2)}$ and $\mathcal{H}^{(3)}$.

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

$$\begin{aligned}\mathbf{y}_e^{(3)}(q_1, q_2; n) =& \mathcal{N}\left[\mathbf{x}_e^{(3)}(q_1, q_2; n)\right] \\ =& \sum_{i=1}^{3} \mathcal{H}^{(i)}\left[\mathbf{x}_e^{(3)}(q_1, q_2; n)\right] \\ =& \sum_{i=1}^{3} \mathbf{v}_e^{(3,i)}(q_1, q_2; n)\end{aligned} \tag{3.50}$$

The input-output relationship in (3.50) is depicted pictorially in Fig. 3.6.

3.3.1 Output Pick-Up Matrices

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.

$$\mathbf{v}_e^{(3,1)}(q_1, q_2; n) = \sum_{j=1}^{\binom{3}{1}} \mathbf{S}_{31,j}^{(M)} \mathbf{v}_e^{(1,1)}(n - n_j^{(3,1)}) \tag{3.51}$$

Here, the matrices $\mathbf{S}_{31,j}^{(M)}$ for $j = 1, 2, 3$, which have $\binom{M}{3}$ rows and $\binom{M}{1}$ columns, are used to pick up the appropriate 1-D ensemble output values. Each row of these matrices has only one nonzero unity element. The superscript M denotes the order of nonlinearity for the nonlinear system $\mathcal{N}$. The subscripts ℓ, k and j in $\mathbf{S}^{(M)}_{\ell k,j}$ indicate the dimension of the ensemble input, the dimension of the subsystem, and the index number of the excited k-D outputs, respectively. We call these matrices as the output pick-up matrices. A recursive algorithm is given in Table B.1 for constructing the $\mathbf{S}^{(M)}_{\ell k,j}$ matrices for $\ell = 1, \ldots, M, \quad k = 1, \ldots, \ell - 1$ and $j = 1, \ldots, \binom{\ell}{k}$. The necessary delays $n_j^{(3,1)}$ in (3.51) for $j = 1, 2, 3$ can be obtained by using the formation matrix $\mathbf{T}^{(3)}_{1,1} = \mathbf{I}_{\binom{3}{1}} = \mathbf{I}_3$.

$$\mathbf{n}^{(3,1)} = \mathbf{T}^{(3)}_{1,1}\mathbf{q}^{(3)} \tag{3.52}$$

Here, $\mathbf{n}^{(3,1)} = \begin{bmatrix} n_1^{(3,1)} & n_2^{(3,1)} & n_3^{(3,1)} \end{bmatrix}^T$ and $\mathbf{q}^{(3)} = \begin{bmatrix} 0 & q_2 & q_1 + q_2 \end{bmatrix}^T$.

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

$$\mathbf{v}_e^{(3,2)}(q_1, q_2; n) = \sum_{j=1}^{\binom{3}{2}} \mathbf{S}^{(M)}_{32,j}\mathbf{v}_e^{(2,2)}(q_j^{(3,2)}; n - n_j^{(3,2)}) \tag{3.53}$$

where the output pick up matrices $\mathbf{S}^{(M)}_{32,j}$ 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 $\mathbf{x}_e^{(3)}(q_1, q_2; n)$. The necessary 2-D parameters $q_j^{(3,2)}$ and $n_j^{(3,2)}$ can be obtained by using the input formation matrices $\mathbf{T}^{(3)}_{2,1}$ and $\mathbf{T}^{(3)}_{2,2}$ as follows

$$\mathbf{q}^{(3,2)} = \left[\mathbf{T}^{(3)}_{2,2} - \mathbf{T}^{(3)}_{2,1}\right]\mathbf{q}^{(3)} \quad \text{and} \quad \mathbf{n}^{(3,2)} = \mathbf{T}^{(3)}_{2,1}\mathbf{q}^{(3)} \tag{3.54}$$

where $\mathbf{q}^{(3,2)} = \begin{bmatrix} q_1^{(3,2)} & q_2^{(3,2)} & q_3^{(3,2)} \end{bmatrix}^T$ and $\mathbf{n}^{(3,2)} = \begin{bmatrix} n_1^{(3,2)} & n_2^{(3,2)} & n_3^{(3,2)} \end{bmatrix}^T$.

In a similar fashion to the equations for the 1-D and 2-D subsystems ((3.15), (3.35)), the output of the 3-D subsystem $\mathbf{v}_e^{(3,3)}(q_1, q_2; n)$ can be written as

$$\mathbf{v}_e^{(3,3)}(q_1, q_2; n) = \mathbf{U}_e^{(3)}\mathbf{h}^{(3)}(q_1, q_2; n - q_1 - q_2) \tag{3.55}$$

for $q_1 = 1, \ldots, N-1$, $q_2 = 1, \ldots, N - q_1$ and $n = q_1 + q_2, q_1 + q_2 + 1, \ldots, N$. Here, the matrix $\mathbf{U}_e^{(3)}$ has the dimensions $\binom{M}{3} \times \binom{M}{3}$, and it can be written in terms of the input levels in a way analogous to (3.13) and (3.36).

$$\mathbf{U}_e^{(3)} = \begin{bmatrix} \mathbf{u}^{(3)^T}(1, 2, 3) \\ \mathbf{u}^{(3)^T}(1, 2, 4) \\ \vdots \\ \mathbf{u}^{(3)^T}(M-2,\ M-1,\ M) \end{bmatrix} \tag{3.56a}$$

where

$$\mathbf{u}^{(3)^T}(i,j,k) = \left[u_3^{(3)}(i,j,k) \,\vdots\, \mathbf{u}_4^{(3)^T}(i,j,k) \,\vdots\, \cdots \,\vdots\, \mathbf{u}_M^{(3)^T}(i,j,k) \right] \tag{3.56b}$$

with

$$\begin{aligned}
u_3^{(3)}(i,j,k) &= [a_i a_j a_k]_{1\times\binom{2}{2}} \\
\mathbf{u}_4^{(3)^T}(i,j,k) &= \left[a_i a_j a_k^2 \quad a_i a_j^2 a_k \quad a_i^2 a_j a_k\right]_{1\times\binom{3}{2}} \\
\mathbf{u}_5^{(3)^T}(i,j,k) &= \left[a_i a_j a_k^3 \quad a_i a_j^2 a_k^2 \quad \cdots \quad a_i^3 a_j a_k\right]_{1\times\binom{4}{2}} \\
&\vdots \\
\mathbf{u}_M^{(3)^T}(i,j,k) &= \left[a_i a_j a_k^{M-2} \quad a_i a_j^2 a_k^{M-3} \quad \cdots \quad a_i^{M-2} a_j a_k\right]_{1\times\binom{M-1}{2}}
\end{aligned} \tag{3.56c}$$

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

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

From (3.50)-(3.55), we get the desired calculation formula for the 3-D kernel vectors.

$$\mathbf{h}^{(3)}(q_1,q_2;n-q_1-q_2) = \left[\mathbf{U}_e^{(3)}\right]^{-1} \mathbf{v}_e^{(3,3)}(q_1,q_2;n) \tag{3.57}$$

where,

$$\begin{aligned}
\mathbf{v}_e^{(3,3)}(q_1,q_2;n) &= \mathbf{y}_e^{(3)}(q_1,q_2;n) \\
&\quad - \left(\sum_{j=1}^{3} \mathbf{S}_{31,j}^{(M)} \mathbf{v}_e^{(1,1)}(n-n_j^{(3,1)}) + \sum_{j=1}^{3} \mathbf{S}_{32,j}^{(M)} \mathbf{v}_e^{(2,2)}(q_j^{(3,2)};n-n_j^{(3,2)}) \right)
\end{aligned} \tag{3.58}$$

Fig. 3.7 depicts the identification method for 3-D kernels as described by (3.57) and (3.58).

Example 3.3: We conclude the identification experiment we started in Example 3.1. We consider again the nonlinear system given in (3.18). Here, the problem is to determine the only nonzero 3-D kernel vector, namely

$$\mathbf{h}^{(3)}(1,1;0) = [10]$$

Using Table A.1, we can construct the ensemble input formation matrices $\mathbf{T}_{3,1}^{(3)}$, $\mathbf{T}_{3,2}^{(3)}$ and $\mathbf{T}_{3,3}^{(3)}$.

$$\mathbf{T}_{3,1}^{(3)} = [1 \quad 0 \quad 0] \qquad \mathbf{T}_{3,2}^{(3)} = [0 \quad 1 \quad 0] \qquad \mathbf{T}_{3,3}^{(3)} = [0 \quad 0 \quad 1]$$

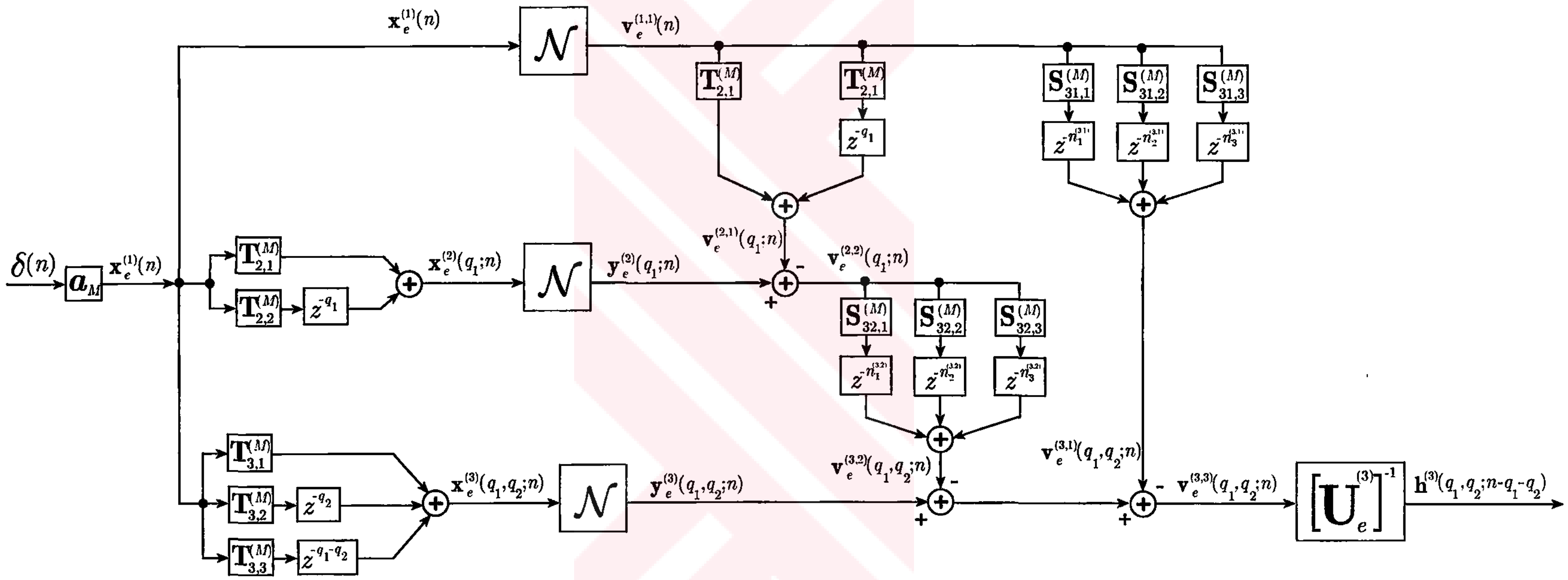


Figure 3.7: Method used for identification of 3-D Volterra kernels, $\mathbf{h}^{(3)}(q_1, q_2; n)$.

The derivation of these matrices is detailed after Table A.1. Having found these matrices, the 3-D input ensemble in (3.49) can be written via the 1-D input ensemble vector $\mathbf{x}_e^{(1)}(n)$ in (3.19).

$$\begin{aligned}
\mathbf{x}_e^{(3)}(1,1;n) &= \begin{bmatrix}1 & 0 & 0\end{bmatrix}\begin{bmatrix}1\\-1\\2\end{bmatrix}\delta(n) + \begin{bmatrix}0 & 1 & 0\end{bmatrix}\begin{bmatrix}1\\-1\\2\end{bmatrix}\delta(n-1)\\
&+ \begin{bmatrix}0 & 0 & 1\end{bmatrix}\begin{bmatrix}1\\-1\\2\end{bmatrix}\delta(n-2)\\
&= \delta(n) - \delta(n-1) + 2\delta(n-2)
\end{aligned} \tag{3.59}$$

For this 3-D input ensemble in (3.59), the output of the nonlinear system in (3.18) is calculated as

$$\mathbf{y}_e^{(3)}(1,1;n) = 6\delta(n) - 3\delta(n-1) - 25\delta(n-2) - 67\delta(n-3) + 76\delta(n-3) \tag{3.60}$$

It is possible to determine the response of the 1-D subsystem to the 3-D input ensemble in (3.59). This is accomplished by applying (3.51) to the 1-D output ensemble given in (3.20).

$$\begin{aligned}
\mathbf{v}_e^{(3,1)}(1,1;n) &= \sum_{j=1}^{3}\mathbf{S}_{31,j}^{(3)}\mathbf{v}_e^{(1,1)}(n-n_j^{(3,1)})\\
&= \mathbf{S}_{31,1}^{(3)}\mathbf{v}_e^{(1,1)}(n) + \mathbf{S}_{31,2}^{(3)}\mathbf{v}_e^{(1,1)}(n-1) + \mathbf{T}_{31,3}^{(3)}\mathbf{v}_e^{(1,1)}(n-2)\\
&= \begin{bmatrix}1 & 0 & 0\end{bmatrix}\left(\begin{bmatrix}6\\-2\\34\end{bmatrix}\delta(n) + \begin{bmatrix}-6\\2\\-34\end{bmatrix}\delta(n-1) + \begin{bmatrix}15\\-5\\76\end{bmatrix}\delta(n-2)\right)\\
&+ \begin{bmatrix}0 & 1 & 0\end{bmatrix}\left(\begin{bmatrix}6\\-2\\34\end{bmatrix}\delta(n-1) + \begin{bmatrix}-6\\2\\-34\end{bmatrix}\delta(n-2) + \begin{bmatrix}15\\-5\\76\end{bmatrix}\delta(n-3)\right)\\
&+ \begin{bmatrix}0 & 0 & 1\end{bmatrix}\left(\begin{bmatrix}6\\-2\\34\end{bmatrix}\delta(n-2) + \begin{bmatrix}-6\\2\\-34\end{bmatrix}\delta(n-3) + \begin{bmatrix}15\\-5\\76\end{bmatrix}\delta(n-4)\right)
\end{aligned}$$

$$\mathbf{v}_e^{(3,1)}(1,1;n) = 6\delta(n) - 8\delta(n-1) + 51\delta(n-2) - 39\delta(n-3) + 76\delta(n-4) \tag{3.61}$$

It is possible to determine the response of the 2-D subsystem to the 3-D input ensemble in (3.59). This is accomplished by applying (3.53) for the 2-D output ensembles given in (3.41) and (3.47).

$$\begin{aligned}
\mathbf{v}_e^{(3,2)}(1,1;n) &= \sum_{j=1}^{3} \mathbf{S}_{32,j}^{(3)} \mathbf{v}_e^{(2,2)}(q_j^{(3,2)}; n - n_j^{(3,2)}) \\
&= \mathbf{S}_{32,1}^{(3)} \mathbf{v}_e^{(2,2)}(1,n) + \mathbf{S}_{32,2}^{(3)} \mathbf{v}_e^{(2,2)}(2,n) + \mathbf{S}_{32,3}^{(3)} \mathbf{v}_e^{(2,2)}(1,n-1) \\
&= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \left(\begin{bmatrix} 5 \\ -40 \\ 16 \end{bmatrix} \delta(n-1) + \begin{bmatrix} -8 \\ 64 \\ -28 \end{bmatrix} \delta(n-2) \right) \\
&\quad + \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \left(\begin{bmatrix} 8 \\ -64 \\ 28 \end{bmatrix} \delta(n-2) \right) \\
&\quad + \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \left(\begin{bmatrix} 5 \\ -40 \\ 16 \end{bmatrix} \delta(n-2) + \begin{bmatrix} -8 \\ 64 \\ -28 \end{bmatrix} \delta(n-3) \right)
\end{aligned}$$

$$\mathbf{v}_e^{(3,2)}(1,1;n) = 5\delta(n-1) - 56\delta(n-2) - 28\delta(n-3) \tag{3.62}$$

The recursive algorithm to calculate the $\mathbf{S}_{\ell k,j}^{(3)}$ matrices is given in Table B.1. The necessary variables $n_j^{(3,1)}$, $n_j^{(3,2)}$ and $q_j^{(3,2)}$ can be obtained by using (3.52) and (3.54).

$$\mathbf{n}^{(3,1)} = \mathbf{T}_{1,1}^{(3)} \mathbf{q}^{(3)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix}$$

Here, $\mathbf{n}^{(3,1)} = \begin{bmatrix} n_1^{(3,1)} & n_2^{(3,1)} & n_3^{(3,1)} \end{bmatrix}^T$ and $\mathbf{q}^{(3)} = \begin{bmatrix} 0 & q_2 & q_1 + q_2 \end{bmatrix}^T$. Using (3.54), the necessary 2-D parameters $q_j^{(3,2)}$ and $n_j^{(3,2)}$ can be obtained as

$$\begin{aligned}
\mathbf{q}^{(3,2)} = \left[\mathbf{T}_{2,2}^{(3)} - \mathbf{T}_{2,1}^{(3)} \right] \mathbf{q}^{(3)} &= \left(\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \right) \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix} \\
&= \begin{bmatrix} -1 & 1 & 0 \\ -1 & 0 & 1 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}
\end{aligned}$$

and

$$\mathbf{n}^{(3,2)} = \mathbf{T}_{2,1}^{(3)} \mathbf{q}^{(3)} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

where $\mathbf{q}^{(3,2)} = \begin{bmatrix} q_1^{(3,2)} & q_2^{(3,2)} & q_3^{(3,2)} \end{bmatrix}^T$ and $\mathbf{n}^{(3,2)} = \begin{bmatrix} n_1^{(3,2)} & n_2^{(3,2)} & n_3^{(3,2)} \end{bmatrix}^T$.

After these calculations, the response of the 3-D subsystem can be obtained by subtracting $\mathbf{v}_e^{(3,1)}(1,1;n)$ and $\mathbf{v}_e^{(3,2)}(1,1;n)$ from the nonlinear system output

$\mathbf{y}_e^{(3)}(1,1;n)$ in (3.60).

$$\begin{aligned}\mathbf{v}_e^{(3,3)}(1,1;n) =&\mathbf{y}_e^{(3)}(1,1;n) - \mathbf{v}_e^{(3,1)}(1,1;n) - \mathbf{v}_e^{(3,2)}(1,1;n)\\ =& \left[-20\right]\delta(n-2)\end{aligned} \tag{3.63}$$

The desired 3-D Volterra kernel vector $\mathbf{h}^{(3)}(1,1;0)$ can be calculated by substituting (3.63) into (3.57)

$$\begin{aligned}\mathbf{h}^{(3)}(1,1;0) &= \left[\mathbf{U}_e^{(3)}\right]^{-1}\mathbf{v}_e^{(3,3)}(1,1;0)\\ &= \left[-2\right]^{-1}\left[-20\right] = 10\end{aligned} \tag{3.64}$$

Here, the constant matrix $\mathbf{U}_e^{(3)}$ is obtained from $\mathbf{x}_e^{(1)}(n)$ and (3.56). □

3.4 ℓ-D KERNEL VECTORS

We assume that all k-D kernel vectors $\mathbf{h}^{(k)}(q_1,\ldots,q_{k-1};i)$ for $k = 1,2,\ldots,\ell-1$ are determined or that all the k-D output vectors $\mathbf{v}_e^{(k,k)}(q_1,\ldots,q_{k-1};n)$ for $k = 1,2,\ldots,\ell-1$ are available. Now, we try to identify the ℓ-D kernel vectors by using the response of the nonlinear system for the ℓ-D ensemble input vector and all the previously computed subsystem outputs, $\mathbf{v}_e^{(k,k)}(q_1,\ldots,q_{k-1};n)$, for $k = 1,2,\ldots,\ell-1$. Similar to 1-, 2-, and 3-D ensemble input vectors in (3.11), (3.31) and (3.49), the ℓ-D input ensemble vector can be written as

$$\mathbf{x}_e^{(\ell)}(q_1,\ldots,q_{\ell-1};n) = \sum_{i=1}^{\ell}\mathbf{T}_{\ell,i}^{(M)}\mathbf{x}_e^{(1)}(n-n_i^{(\ell)}) \tag{3.65}$$

where, $n_1^{(\ell)} = 0$ and $n_i^{(\ell)} = \sum_{j=1}^{i-1} q_{\ell-j}$ for $i = 2,\ldots,\ell$. The input formation matrices, $\mathbf{T}_{\ell,i}^{(M)}$'s can be constructed from the Table A.1. The response of the nonlinear system to the ensemble input in (3.65) can be written in terms of the outputs of the subsystems,

$$\begin{aligned}\mathbf{y}_e^{(\ell)}(q_1,\ldots,q_{\ell-1};n) =&\mathcal{N}\left[\mathbf{x}_e^{(\ell)}(q_1,\ldots,q_{\ell-1};n)\right]\\ =&\sum_{k=1}^{\ell}\mathcal{H}^{(k)}\left[\mathbf{x}_e^{(\ell)}(q_1,\ldots,q_{\ell-1};n)\right]\\ =&\sum_{k=1}^{\ell}\mathbf{v}_e^{(\ell,k)}(q_1,\ldots,q_{\ell-1};n)\end{aligned} \tag{3.66}$$

Fig. 3.8 draws a picture of (3.66), by showing that for an ℓ-D input ensemble, the outputs of all subsystems $\mathcal{H}^{(k)}$, for $k > \ell$, are equal to zero.

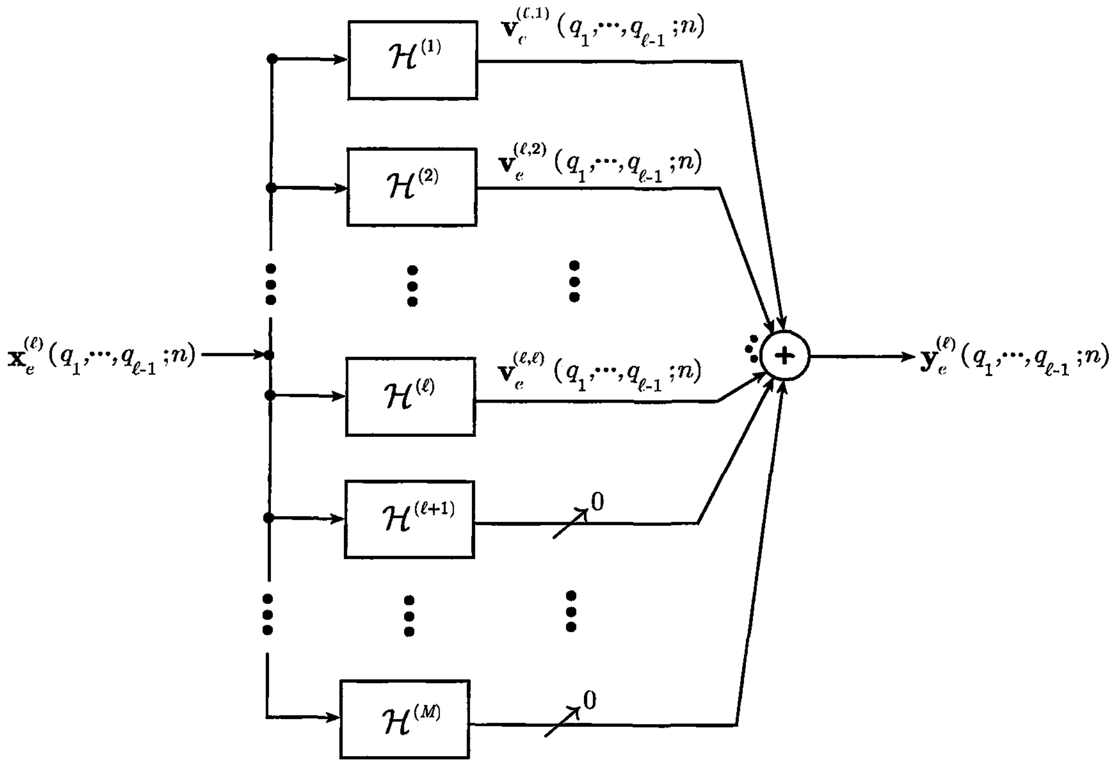


Figure 3.8: Pictorial description for (3.66). The output of the nonlinear system $\mathcal{N}$ for the ℓ-D input ensemble $\mathbf{x}_e^{(\ell)}(q_1,\ldots,q_{\ell-1};n)$, is equal to the sum of the outputs of subsystems $\mathcal{H}^{(1)}$ through $\mathcal{H}^{(\ell)}$.

In (3.66) $\mathbf{v}_e^{(\ell,k)}(q_1,\ldots,q_{\ell-1};n)$ for $k=1,\ldots,\ell-1$ can be obtained from the previous subsystem outputs.

$$\begin{aligned}
\mathbf{v}_e^{(\ell,1)}(q_1,\ldots,q_{\ell-1};n) &= \sum_{j=1}^{\binom{\ell}{1}} \mathbf{S}_{\ell 1,j}^{(M)} \mathbf{v}_e^{(1,1)}(n-n_j^{(\ell,1)}) \\
&\;\vdots \\
\mathbf{v}^{(\ell,k)}(q_1,\ldots,q_{\ell-1};n) &= \sum_{j=1}^{\binom{\ell}{k}} \mathbf{S}_{\ell 1,j}^{(M)} \mathbf{v}_e^{(k,k)}(q_{j,1}^{(\ell,k)},\ldots,q_{j,k-1}^{(\ell,k)};n-n_j^{(\ell,k)})
\end{aligned} \tag{3.67}$$

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

$$\mathbf{q}_j^{(\ell,k)} = \begin{bmatrix} q_{j,1}^{(\ell,k)} & \cdots & q_{j,k-1}^{(\ell,k)} \end{bmatrix}^T \text{ and } \mathbf{n}^{(\ell,k)} = \begin{bmatrix} n_1^{(\ell,k)} & \cdots & n_{\binom{\ell}{k}}^{(\ell,k)} \end{bmatrix}^T \tag{3.68}$$

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

$$\begin{aligned}\mathbf{q}_j^{(\ell,k)} &= \left[\mathbf{T}_{k,j+1}^{(\ell)} - \mathbf{T}_{k,j}^{(\ell)}\right]\mathbf{q}^{(\ell)} \quad \text{for } j = 1, \ldots, \tbinom{\ell}{k} \\ \mathbf{n}^{(\ell,k)} &= \mathbf{T}_{k,1}^{(\ell)}\mathbf{q}^{(\ell)}\end{aligned} \tag{3.69}$$

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

$$\begin{aligned}\mathbf{v}_e^{(\ell,\ell)}(q_1, \ldots, q_{\ell-1}; n) &= \mathcal{H}^{(\ell)}\left[\mathbf{x}_e^{(\ell)}(q_1, \ldots, q_{\ell-1}; n)\right] \\ &= \sum_{i=0}^{N-\bar{q}_{\ell-1}} \mathbf{U}_e^{(\ell)}(q_1, \ldots, q_{\ell-1}; n-i)\mathbf{h}^{(\ell)}(q_1, \ldots, q_{\ell-1}; i)\end{aligned} \tag{3.70}$$

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 $a_1, a_2, \ldots, a_M$ as,

$$\mathbf{U}_e^{(\ell)} = \begin{bmatrix} \mathbf{u}^{(\ell)^T}(1, 2, \ldots, \ell-1, \ell) \\ \mathbf{u}^{(\ell)^T}(1, 2, \ldots, \ell-1, \ell+1) \\ \vdots \\ \mathbf{u}^{(\ell)^T}(M-\ell+1,\ M-\ell+2,\ \ldots, M) \end{bmatrix} \tag{3.71a}$$

where

$$\mathbf{u}^{(\ell)^T}(i_1, \ldots, i_\ell) = \left[u_\ell^{(\ell)}(i_1, \ldots, i_\ell) \vdots \mathbf{u}_{\ell+1}^{(\ell)^T}(i_1, \ldots, i_\ell) \vdots \cdots \vdots \mathbf{u}_M^{(\ell)^T}(i_1, \ldots, i_\ell)\right] \tag{3.71b}$$

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

$$\mathbf{u}_k^{(\ell)}(i_1, i_2, \ldots, i_\ell) = \left[a_{i_1}^{p_1} a_{i_2}^{p_2} \cdots a_{i_\ell}^{p_\ell}\right]_{\sigma(p_1, p_2, \ldots, p_\ell)} \tag{3.72}$$

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

$$\begin{aligned}
u_\ell^{(\ell)}(i_1, i_2, \ldots, i_\ell) &= [a_{i_1} a_{i_2} \cdots a_{i_\ell}]_{1\times\binom{\ell-1}{\ell-1}} \\
\mathbf{u}_{\ell+1}^{(\ell)^T}(i_1, i_2, \ldots, i_\ell) &= \left[\left(a_{i_1} \cdots a_{i_{\ell-1}} a_{i_\ell}^2\right) \ \cdots \ \left(a_{i_1} a_{i_2}^2 a_{i_3} \cdots a_{i_\ell}\right) \ \left(a_{i_1}^2 a_{i_2} \cdots a_{i_\ell}\right)\right]_{1\times\binom{\ell}{\ell-1}} \\
\mathbf{u}_{\ell+2}^{(\ell)^T}(i_1, i_2, \ldots, i_\ell) &= \left[\left(a_{i_1} \cdots a_{i_{\ell-1}} a_{i_\ell}^3\right) \ \cdots \ \left(a_{i_1}^2 a_{i_2}^2 a_{i_3} \cdots a_{i_\ell}\right) \ \left(a_{i_1}^3 a_{i_2} \cdots a_{i_\ell}\right)\right]_{1\times\binom{\ell+1}{\ell-1}} \\
&\ \ \vdots \\
\mathbf{u}_M^{(\ell)^T}(i_1, i_2, \ldots, i_\ell) &= \left[\left(a_{i_1} \cdots a_{i_{\ell-1}} a_{i_\ell}^{M-\ell+1}\right) \ \cdots \ \left(a_{i_1}^{M-\ell} a_{i_2}^2 a_{i_3} \cdots a_{i_\ell}\right)\right. \\
&\qquad \left.\left(a_{i_1}^{M-\ell+1} a_{i_2} a_{i_3} \cdots a_{i_\ell}\right)\right]_{1\times\binom{M-1}{\ell-1}}
\end{aligned} \tag{3.73}$$

As an example, for $M=6$ and $\ell=4$, the vectors $\mathbf{u}_k{}^{(4)^T}(i_1,i_2,i_3,i_4)$ $(k=4,5,6)$ will be given as follows:

$$
\begin{aligned}
u_4^{(4)}(i_1,i_2,i_3,i_4) =& \left[a_{i_1}a_{i_2}a_{i_3}a_{i_4}\right]_{1\times\binom{1}{1}} \\
\mathbf{u}_5^{(4)^T}(i_1,i_2,i_4,i_4) =& \left[\left(a_{i_1}a_{i_2}a_{i_3}a_{i_4}^2\right)\ \left(a_{i_1}a_{i_2}a_{i_3}^2a_{i_4}\right)\ \left(a_{i_1}a_{i_2}^2a_{i_3}a_{i_4}\right)\ \left(a_{i_1}^2a_{i_2}a_{i_3}a_{i_4}\right)\right]_{1\times\binom{4}{3}} \\
\mathbf{u}_6^{(4)^T}(i_1,i_2,i_3,i_4) =& \Big[\left(a_{i_1}a_{i_2}a_{i_3}a_{i_4}^3\right)\quad \left(a_{i_1}a_{i_2}a_{i_3}^2a_{i_4}^2\right)\quad \left(a_{i_1}a_{i_2}^2a_{i_3}a_{i_4}^2\right) \\
& \left(a_{i_1}^2a_{i_2}a_{i_3}a_{i_4}^2\right)\ \left(a_{i_1}a_{i_2}a_{i_3}^3a_{i_4}\right)\ \left(a_{i_1}a_{i_2}^2a_{i_3}^2a_{i_4}\right)\ \left(a_{i_1}^2a_{i_2}a_{i_3}^2a_{i_4}\right) \\
& \left(a_{i_1}a_{i_2}^3a_{i_3}a_{i_4}\right)\ \left(a_{i_1}^2a_{i_2}^2a_{i_3}a_{i_4}\right)\ \left(a_{i_1}^3a_{i_2}a_{i_3}a_{i_4}\right)\Big]_{1\times\binom{5}{3}}
\end{aligned}
\tag{3.74}
$$

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

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

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

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

$$
\mathbf{h}^{(\ell)}(q_1,\ldots,q_{\ell-1};n-\bar{q}_{\ell-1}) = \left[\mathbf{U}_e^{(\ell)}\right]^{-1}\mathbf{v}_e^{(\ell,\ell)}(q_1,\ldots,q_{\ell-1};n) \tag{3.75a}
$$

Here,

$$
\begin{aligned}
\mathbf{v}_e^{(\ell,\ell)}(q_1,\ldots,q_{\ell-1};n) = \mathbf{y}_e^{(\ell)}(q_1,\ldots,q_{\ell-1};n) &- \sum_{j=1}^{\binom{\ell}{1}} \mathbf{S}_{\ell 1,j}^{(M)}\mathbf{v}_e^{(1,1)}(n-n_j^{(\ell,1)}) \\
&- \sum_{k=2}^{\ell-1}\sum_{j=1}^{\binom{\ell}{k}} \mathbf{S}_{\ell k,j}^{(M)}\mathbf{v}_e^{(k,k)}(q_{j,1}^{(\ell,k)},\ldots,q_{j,k-1}^{(\ell,k)};n-n_j^{(\ell,k)})
\end{aligned}
\tag{3.75b}
$$

(3.75) shows that our algorithm can form the estimate for any Volterra kernel independent from other kernels. (3.75a) only includes the inverse of a matrix ($\mathbf{U}_e^{(\ell)}$) which can be calculated beforehand and output values ($\mathbf{v}_e^{(\ell,\ell)}$) which should be obtained for certain multilevel input sequences. In this equation there is no reference to estimates of other kernel values. Hence, any error in previous kernel estimates cannot leak into current kernel estimate calculations. This equation proves the no-error propagation property of our algorithm.

3.4.1 Output Pick-Up Operators

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

$$\mathbb{S}_{\ell,k}^{(M)}\left[\mathbf{v}_e^{(k,k)}(q_1, \ldots, q_{k-1}; n)\right] = \sum_{j=1}^{\binom{\ell}{k}} \mathbf{S}_{\ell k,j}^{(M)} \mathbf{v}_e^{(k,k)}(q_{j,1}^{(\ell,k)}, \ldots, q_{j,k-1}^{(\ell,k)}; n - n_j^{(\ell,k)}) \quad (3.76)$$

With this definition, (3.75b) can be rewritten in a more compact form.

$$\mathbf{v}_e^{(\ell,\ell)}(q_1, \ldots, q_{\ell-1}; n) = \mathbf{y}_e^{(\ell)}(q_1, \ldots, q_{\ell-1}; n) - \sum_{k=1}^{\ell-1} \mathbb{S}_{\ell,k}^{(M)}\left[\mathbf{v}_e^{(k,k)}(q_1, \ldots, q_{k-1}; n)\right] \quad (3.77)$$

Fig. 3.9 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.

3.5 IMPLEMENTATION OF THE IDENTIFICATION ALGORITHM

Let us discuss the practical implementation of the algorithm under the guidance of Fig. 3.9. The system to be modelled is assumed to have a known order of nonlinearity M. The memory length N might be assumed to be known, or it might get estimated during the identification procedure on the fly. The steps of the measurement process might be listed as follows:

- The impulse levels $a_1, a_2, \ldots, a_M$ are chosen, possibly using numerical optimization.

- Multilevel sequences composed of a single impulse ($\mathbf{x}_e^{(1)}(n)$ in Fig. 3.9) are applied to the system and the observed output vectors are stored ($\mathbf{y}_e^{(1)}(n)$ in Fig. 3.9).

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

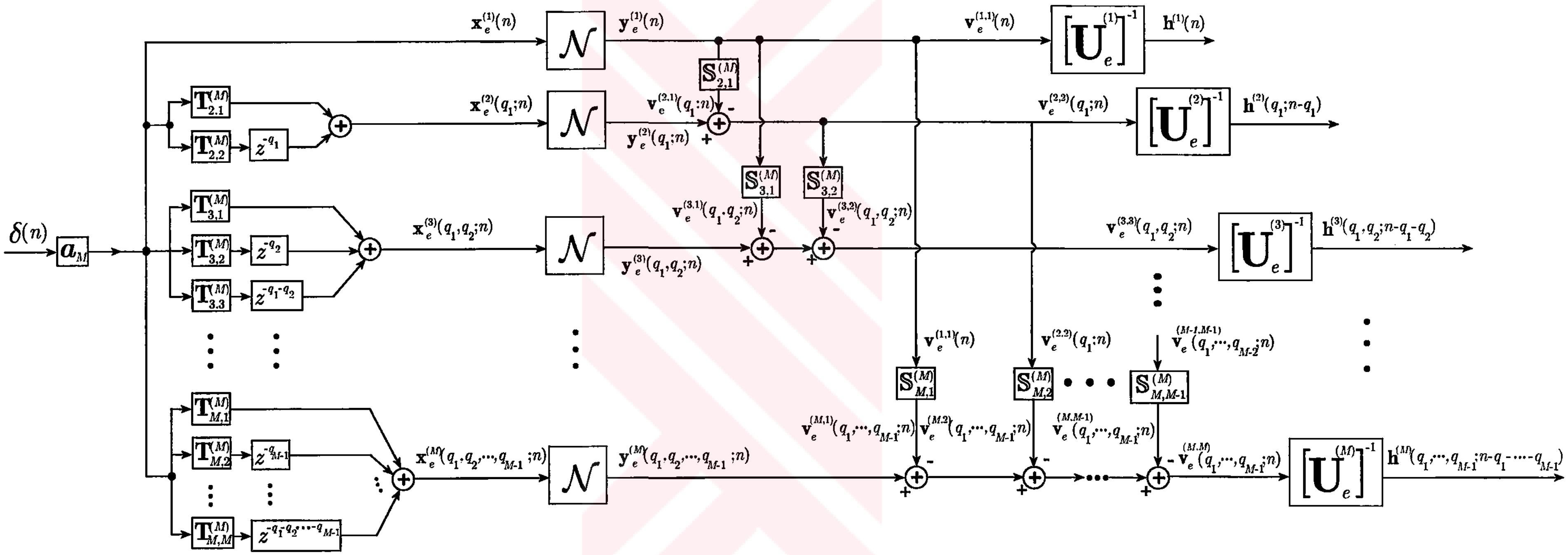


Figure 3.9: Proposed Volterra kernel identification method using multilevel deterministic sequences as inputs.

- $\mathbf{y}_e^{(1)}(n) = \mathbf{v}_e^{(1,1)}(n) = \mathcal{N}\Big[\mathbf{x}_e^{(1)}(n)\Big]$

- for $\ell = 2, \ldots, M$:

  Multilevel sequences composed of ℓ distinct impulses $\Big(\mathbf{x}_e^{(\ell)}(q_1, \ldots, q_{\ell-1}; n)$ in Fig. 3.9$\Big)$ are applied to the system, and the observed output vectors are stored $\Big(\mathbf{y}_e^{(\ell)}(q_1, \ldots, q_{\ell-1}; n)$ in Fig. 3.9$\Big)$. The $\mathbb{S}_{\ell,k}^M$ operators $(k = 1, 2, \ldots, \ell - 1)$ form sums of appropriate previously stored output vectors. The outputs of the $\mathbb{S}_{\ell,k}^M$ operators are subtracted from $\mathbf{y}_e^{(\ell)}(q_1, \ldots, q_{\ell-1}; n)$. Hence, $\mathbf{v}_e^{(\ell,\ell)}(q_1, \ldots, q_{\ell-1}; n)$ is formed.

  - $\mathbf{x}_e^{(\ell)}(q_1, \ldots, q_{\ell-1}; n) = \sum\limits_{i=1}^{\ell} \mathbf{T}_{\ell,i}^{(M)} \mathbf{x}_e^{(1)}(n - n_i^{(\ell)})$
  - $\mathbf{y}_e^{(\ell)}(q_1, \ldots, q_{\ell-1}; n) = \mathcal{N}\left[\mathbf{x}_e^{(\ell)}(q_1, \ldots, q_{\ell-1}; n)\right]$
  - $\mathbf{v}_e^{(\ell,\ell)}(q_1, \ldots, q_{\ell-1}; n) = \mathbf{y}_e^{(\ell)}(q_1, \ldots, q_{\ell-1}; n) - \sum\limits_{k=1}^{\ell-1} \mathbb{S}_{\ell,k}^{(M)}\left[\mathbf{v}_e^{(k,k)}(q_1, \ldots, q_{k-1}; n)\right]$

- for $\ell = 1, \ldots, M$:

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

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

3.6 LENGTH OF THE REQUIRED INPUT SEQUENCE

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

$$\begin{aligned} L =& \sum_{\ell=1}^{M} (\bar{q}_{\ell-1} + N + 1) \binom{M}{\ell} \binom{N}{\ell - 1} \\ &\leqslant (2N+1) \sum_{\ell=1}^{M} \binom{M}{\ell} \binom{N}{\ell-1} = (2N+1) \binom{N+M}{M-1} \end{aligned} \tag{3.78}$$

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

Example 3.4: We consider the identification of a Volterra filter with $M = 3$ and $N = 2$ as discussed in Examples 3.1, 3.2 and 3.3. The overall deterministic input sequence which should be applied to identify the kernels of this system is shown in Fig. 3.10. This figure depicts all the input ensembles utilized in the Examples 3.1, 3.2 and 3.3 for the identification of the nonlinear system. The overall input sequence is assumed to be applied to a single system box, and there is a guarding interval of length $N = 2$ between all individual input ensembles. The distinct input amplitude levels are chosen as $a_1 = 1$, $a_2 = -1$ and $a_3 = 2$ as in the examples. The length of the overall sequence is seen from the figure as $L = 39$. This value is indeed less than the upper-bound $(2N+1)\binom{N+M}{M-1} = (5)\binom{5}{2} = 50$ from (3.78). $\square$

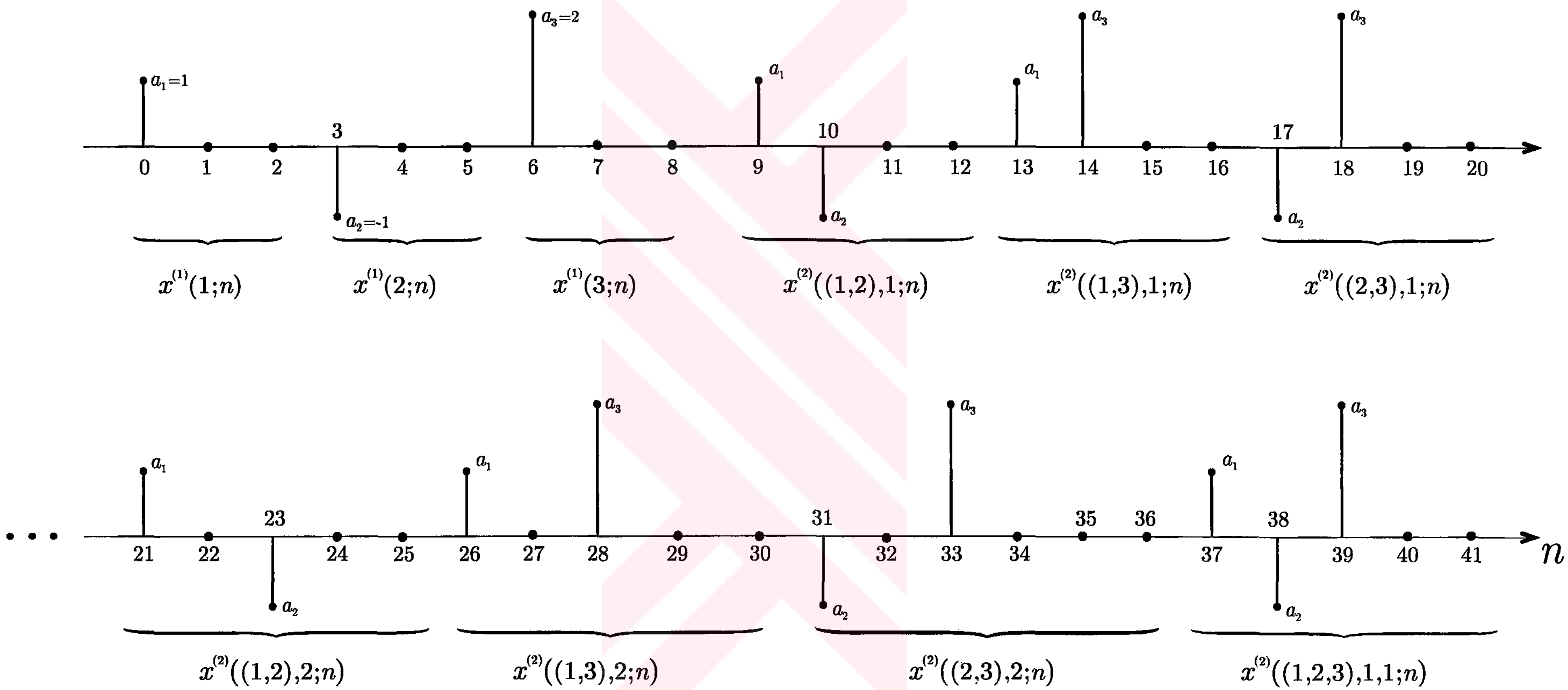


Figure 3.10: Deterministic multilevel input sequence for Example 3.4.

4. PERSISTENCE OF EXCITATION, NUMERICAL ISSUES AND RESULTS FOR THE MULTILEVEL INPUT SIGNALS

In this chapter, we prove that the multilevel input signals as defined in Chapter 3 persistently excite a Volterra filter. We also show that the identification algorithm of Chapter 3 actually gives the optimal least squares estimate of the Volterra kernels. We deal with the problem of uniform excitation for the defined identification algorithm using multilevel probing signals. Ideally, an input signal excites all modes of the system which is to be identified equally. However, it is known that polynomial systems suffer from severe ill-conditioning if the nonlinearity order of the system is high. In this chapter, the condition numbers of the utilized input matrices are quantified to assess the uniformity of excitation to the system for the proposed identification algorithm.

4.1 PERSISTENCE OF EXCITATION FOR THE MULTILEVEL DETERMINISTIC INPUTS

4.1.1 Least-Squares Solution for $\mathcal{H}^{(\ell)}$

We start by rewriting the input-output relation of the ℓ-D cross-term subsystem, $\mathcal{H}^{(\ell)}$, but this time in a slightly different, more condensed manner. The ℓ-D multivariate cross-term subsystem of (3.1c) can be rewritten as

$$y^{(\ell)}(n) = \mathcal{H}^{(\ell)}\big[x(n)\big] = \sum_{p=1}^{\binom{N}{\ell-1}} \sum_{i=0}^{N-\mathbf{sum}(\boldsymbol{q}_p)} \mathbf{h}^{(\ell)^T}(\boldsymbol{q}_p;i)\,\mathbf{x}^{(\ell)}(\boldsymbol{q}_p;n-i), \tag{4.1}$$

where the sum over p is a sum over all possible delay structures. Hence, we replaced all the $\ell-1$ summations over the q_i variables in (3.1c), with a single summation over the single variable p. The number of possible delay structures was calculated in (2.42) as $\binom{N}{\ell-1}$, and this number is utilized here. $\boldsymbol{q}_p = \begin{bmatrix} q_1 & q_2 & \dots & q_{\ell-1}\end{bmatrix}_p$ is an $(\ell-1)$-D delay vector with $\mathbf{sum}(\boldsymbol{q}_p) \leqslant N$. $\mathbf{sum}(\boldsymbol{a})$

denotes the summation of the elements of the vector $\boldsymbol{a}$. The length of the vector $\mathbf{h}^{(\ell)}(\boldsymbol{q}_p; i)$ is $\binom{M}{\ell}$ using (2.21).

We can reformulate the output in (4.1) as

$$y^{(\ell)}(n) = \mathcal{H}^{(\ell)}\big[x(n)\big] = \sum_{p=1}^{\binom{N}{\ell-1}} \mathbf{h}^{(\ell)^T}(\boldsymbol{q}_p)\mathbf{x}_n^{(\ell)}(\boldsymbol{q}_p) \tag{4.2}$$

where, $\mathbf{h}^{(\ell)}(\boldsymbol{q}_p)$ and $\mathbf{x}_n^{(\ell)}(\boldsymbol{q}_p)$ are column vectors. $\mathbf{h}^{(\ell)}(\boldsymbol{q}_p)$ is a concatenation of the kernel vectors $\mathbf{h}^{(\ell)}(\boldsymbol{q}_p; i)$, whereas $\mathbf{x}_n^{(\ell)}(\boldsymbol{q}_p)$ is a concatenation of the expanded input vectors $\mathbf{x}^{(\ell)}(\boldsymbol{q}_p; n-i)$.

$$\mathbf{h}^{(\ell)}(\boldsymbol{q}_p) = \begin{bmatrix} \mathbf{h}^{(\ell)}(\boldsymbol{q}_p; 0) \\ \mathbf{h}^{(\ell)}(\boldsymbol{q}_p; 1) \\ \vdots \\ \mathbf{h}^{(\ell)}\big(\boldsymbol{q}_p; N - \mathrm{sum}(\boldsymbol{q}_p)\big) \end{bmatrix} \quad \mathbf{x}_n^{(\ell)}(\boldsymbol{q}_p) = \begin{bmatrix} \mathbf{x}^{(\ell)}(\boldsymbol{q}_p; n) \\ \mathbf{x}^{(\ell)}(\boldsymbol{q}_p; n-1) \\ \vdots \\ \mathbf{x}^{(\ell)}\Big(\boldsymbol{q}_p; n - \big(N - \mathrm{sum}(\boldsymbol{q}_p)\big)\Big) \end{bmatrix} \tag{4.3}$$

We can rewrite the linear combination in (4.2) as a single vector product,

$$y^{(\ell)}(n) = \mathcal{H}^{(\ell)}\big[x(n)\big] = \mathbf{h}^{(\ell)^T}\mathbf{x}^{(\ell)}(n) \tag{4.4}$$

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.

$$\mathbf{h}^{(\ell)} = \begin{bmatrix} \mathbf{h}^{(\ell)}(\boldsymbol{q}_1) \\ \mathbf{h}^{(\ell)}(\boldsymbol{q}_2) \\ \vdots \\ \mathbf{h}^{(\ell)}\big(\boldsymbol{q}_{\binom{N}{\ell-1}}\big) \end{bmatrix} \qquad \mathbf{x}^{(\ell)}(n) = \begin{bmatrix} \mathbf{x}_n^{(\ell)}(\boldsymbol{q}_1) \\ \mathbf{x}_n^{(\ell)}(\boldsymbol{q}_2) \\ \vdots \\ \mathbf{x}_n^{(\ell)}\big(\boldsymbol{q}_{\binom{N}{\ell-1}}\big) \end{bmatrix} \tag{4.5}$$

The length of the vector $\mathbf{h}^{(\ell)}$ is $\binom{M}{\ell}\binom{N+1}{\ell}$, using (2.21) and (2.39).

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.

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

The output vector is related to the input by

$$\mathbf{y}^{(\ell)}(n) = \mathbf{X}^{(\ell)}(n)\mathbf{h}^{(\ell)^*} \tag{4.6}$$

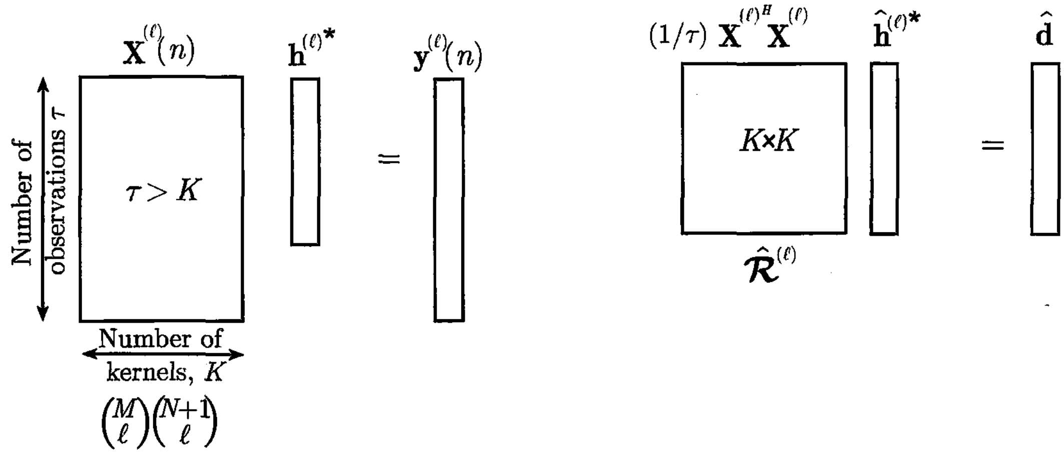


Figure 4.1: Graphical description for (4.6) and (4.8).

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

$$\mathbf{X}^{(\ell)}(n) = \begin{bmatrix} \mathbf{x}^{(\ell)}(n)^H \\ \mathbf{x}^{(\ell)}(n+1)^H \\ \vdots \\ \mathbf{x}^{(\ell)}(n+\tau)^H \end{bmatrix} \tag{4.7}$$

$(\,\cdot\,)^H$ denotes the Hermitian transpose, and $(\,\cdot\,)^*$ denotes complex conjugation.

(4.6) defines the input-output relation of the ℓ-D subsystem $\mathcal{H}^{(\ell)}$ as a single matrix product. 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 (4.6) can be written as below (Manolakis et al., 2000).

$$\hat{\mathbf{h}}^{(\ell)*} = \hat{\mathcal{R}}^{(\ell)^{-1}} \hat{\mathbf{d}} \tag{4.8}$$

(4.6) and (4.8) are portrayed in Fig. 4.1. In (4.8), $\hat{\mathcal{R}}^{(\ell)}$ 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.

$$\hat{\mathcal{R}}^{(\ell)} = \frac{1}{\tau} \mathbf{X}^{(\ell)^H}(n)\, \mathbf{X}^{(\ell)}(n) = \frac{1}{\tau} \sum_{m=n}^{n+\tau} \mathbf{x}^{(\ell)}(m)\, \mathbf{x}^{(\ell)^H}(m) \tag{4.9}$$

$$\hat{\mathbf{d}} = \frac{1}{\tau} \mathbf{X}^{(\ell)^H}(n)\, \mathbf{y}^{(\ell)}(n) = \frac{1}{\tau} \sum_{m=n}^{n+\tau} \mathbf{x}^{(\ell)}(m)\, y^*(m) \tag{4.10}$$

Hence, (4.8) can be rewritten as

$$\hat{\mathbf{h}}^{(\ell)*} = \left(\mathbf{X}^{(\ell)^H}(n)\, \mathbf{X}^{(\ell)}(n)\right)^{-1} \mathbf{X}^{(\ell)^H}(n)\, \mathbf{y}^{(\ell)}(n) \tag{4.11}$$

The least squares solution as formulated in (4.8) and (4.11) has a unique solution if the autocorrelation matrix $\hat{\mathcal{R}}^{(\ell)}$ is invertible. This brings us to the definition of persistence of excitation.

4.1.2 Definition of Persistence of Excitation for $\mathcal{H}^{(\ell)}$

Basically signals, for which the time-average autocorrelation matrix is nonsingular for all times, are said to persistently excite a system (Mathews and Sicuranza, 2000). The definition of persistence of excitation for Volterra filters with deterministic input sequences is given in Nowak and Van Veen (1994b). Following this definition, we formulate the persistence of excitation condition for the ℓ-D cross-term subsystem $\mathcal{H}^{(\ell)}$:

Definition 4.1 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 $\hat{\mathcal{R}}^{(\ell)}$, as defined in (4.9). If there exist positive constants $\rho_1, \rho_2 > 0$ such that for every time instant k

$$\rho_1 \leqslant \lambda_{\min} \leqslant \lambda_{\max} \leqslant \rho_2 \tag{4.12}$$

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

Theorem 4.1 [Persistence of excitation for the deterministic sequence] *The deterministic multilevel input sequence is persistently exciting for a Volterra filter of order M, when the impulse amplitude levels $a_1, a_2, \ldots, a_M$ take distinct and nonzero values.*

Proof. We will first prove that our deterministic ℓ-D ensemble input signals $\mathbf{x}_e^{(\ell)}$, as discussed in Section 3.4, are persistently exciting for the ℓ-D cross-term subsystem $\mathcal{H}^{(\ell)}$. The ℓ-D ensemble input signals do not excite the higher dimensional subsystems, i.e., $\mathbf{v}_e^{(\ell,k)} = \mathcal{H}^{(k)}\big[\mathbf{x}_e^{(\ell)}\big] = \mathbf{0},\ \forall\ k > \ell$, and $\mathcal{N}\big[\mathbf{x}_e^{(\ell)}\big] = \sum_{k=1}^{\ell} \mathbf{v}_e^{(\ell,k)}$. By subtracting the appropriate previously stored lower dimensional outputs, we can obtain the response of the ℓ-D subsystem alone. With this subtraction we single out the input-output ensembles for the ℓ-D cross-term subsystem $\mathcal{H}^{(\ell)}$. If we can prove that the ℓ-D input ensembles are PE for $\mathcal{H}^{(\ell)}$, then all the input ensembles together will be PE for the composite nonlinear structure $\mathcal{N}$ of order M.

Let us assume that we apply our total ℓ-D ensemble input signal periodically. Suppose we begin observing the output of the ℓ-D system at some time n and collect data over an observation period $\tau > 0$. Without loss of generality, we take τ as the length of all the ℓ-D input ensembles together, i.e., as the length of the overall ℓ-D input sequence which we assume is applied periodically to a single system box $\mathcal{N}$.

We first decompose the vector $\mathbf{x}^{(\ell)}(m)$ as given in (4.5). If at the time point m the current ensemble input has a delay structure given by $\boldsymbol{q}_p$, of all the subvectors of $\mathbf{x}^{(\ell)}(m)$ only $\mathbf{x}_m^{(\ell)}(\boldsymbol{q}_p)$ will be nonzero. To put it differently, $\mathbf{x}_m^{(\ell)}(\boldsymbol{q}_k) = \mathbf{0}$, for all $1 \leq k \leq \binom{N}{\ell-1}$ except for $k = p$. Hence,

$$\mathbf{x}^{(\ell)}(m) = \begin{bmatrix} \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \mathbf{x}_m^{(\ell)}(\boldsymbol{q}_p) \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix}$$

We continue further and decompose $\mathbf{x}_m^{(\ell)}(\boldsymbol{q}_p)$ as defined in (4.3). All the subvectors of $\mathbf{x}_m^{(\ell)}(\boldsymbol{q}_p)$, i.e., $\mathbf{x}^{(\ell)}(\boldsymbol{q}_p; m-i)$ for $i = 0 \ldots N - \mathbf{sum}(\boldsymbol{q}_p)$, will be zero vectors except for only one subvector for a certain shift $\bar{\imath}$.

$$\mathbf{x}_m^{(\ell)}(\boldsymbol{q}_p) = \begin{bmatrix} \mathbf{0}_{\binom{M}{\ell}\times 1} \\ \vdots \\ \mathbf{0}_{\binom{M}{\ell}\times 1} \\ \mathbf{x}^{(\ell)}(\boldsymbol{q}_p; m - \bar{\imath}) \\ \mathbf{0}_{\binom{M}{\ell}\times 1} \\ \vdots \\ \mathbf{0}_{\binom{M}{\ell}\times 1} \end{bmatrix}$$

Hence, the time-average correlation matrix, $\hat{\boldsymbol{\mathcal{R}}}^{(\ell)}$ in (4.9), will become a block diagonal matrix. For each ensemble input sequence with a delay structure given by $\boldsymbol{q}_p$, $\mathbf{x}_m^{(\ell)}(\boldsymbol{q}_p)$ will have the nonzero subvector at the position $\bar{\imath} = 0$ at some time m, and at every next point in time the nonzero subvector will travel down until it reaches $\bar{\imath} = N - \mathbf{sum}(\boldsymbol{q}_p)$. The total ensemble input sequence includes all possible

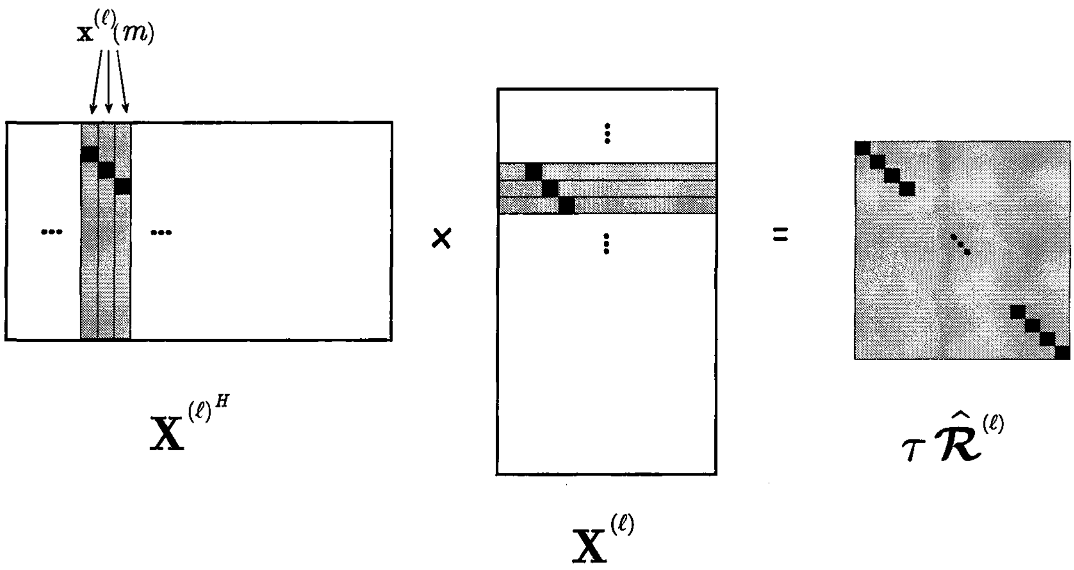


Figure 4.2: For the ℓ-D ensemble input, the time average correlation matrix $\hat{\mathcal{R}}^{(\ell)}$ becomes block diagonal (4.13).

ℓ-level combinations for all possible delay structures. Therefore, when we form the block diagonal time-average correlation matrix, the matrices on the diagonal will be equal, i.e., for the multilevel deterministic ensemble input sequence

$$\hat{\mathcal{R}}^{(\ell)} = \frac{1}{T}\begin{bmatrix} \left[\hat{\mathbf{R}}^{(\ell)}\right] & \left[\mathbf{0}_{\binom{M}{\ell}\times\binom{M}{\ell}}\right] & \cdots & \left[\mathbf{0}_{\binom{M}{\ell}\times\binom{M}{\ell}}\right] \\ \left[\mathbf{0}_{\binom{M}{\ell}\times\binom{M}{\ell}}\right] & \left[\hat{\mathbf{R}}^{(\ell)}\right] & \ddots & \vdots \\ \vdots & \ddots & \ddots & \left[\mathbf{0}_{\binom{M}{\ell}\times\binom{M}{\ell}}\right] \\ \left[\mathbf{0}_{\binom{M}{\ell}\times\binom{M}{\ell}}\right] & \cdots & \left[\mathbf{0}_{\binom{M}{\ell}\times\binom{M}{\ell}}\right] & \left[\hat{\mathbf{R}}^{(\ell)}\right] \end{bmatrix} \tag{4.13}$$

The generation of the block diagonal correlation matrix is shown graphically in Fig. 4.2. $\hat{\mathbf{R}}^{(\ell)}$ in (4.13) has a size of $\binom{M}{\ell} \times \binom{M}{\ell}$. $\hat{\mathbf{R}}^{(\ell)}$ can be represented as

$$\hat{\mathbf{R}}^{(\ell)} = \sum_{j=1}^{\binom{M}{\ell}} \mathbf{u}^{(\ell)}(\boldsymbol{i}_j)\mathbf{u}^{(\ell)^H}(\boldsymbol{i}_j) \tag{4.14}$$

Here, the vector $\mathbf{u}^{(\ell)}(\boldsymbol{i}_j)$ is as defined in (3.71b). $\boldsymbol{i}_j = [i_1\ i_2\ \cdots\ i_\ell]_j$ denotes which impulse amplitudes to pick, and the index $j = 1 \ldots \binom{M}{\ell}$ is such that all possible impulse amplitude ensembles are covered.

$\hat{\mathbf{R}}^{(\ell)}$ in (4.14) can be rewritten as

$$\hat{\mathbf{R}}^{(\ell)} = \left[\mathbf{U}_e^{(\ell)^H}\mathbf{U}_e^{(\ell)}\right]^T \tag{4.15}$$

where $\mathbf{U}_e^{(\ell)}$ is the ℓ-D ensemble input matrix as given in (3.71a). Hence, for our deterministic input ensemble the eigenvalues of the sample correlation matrix $\hat{\boldsymbol{\mathcal{R}}}^{(\ell)} = \frac{1}{\mathcal{T}}\mathbf{X}^{(\ell)^H}(n)\mathbf{X}^{(\ell)}(n)$ are equal to the eigenvalues of the matrix $\hat{\mathbf{R}}^{(\ell)} = \left[\mathbf{U}_e^{(\ell)^H}\mathbf{U}_e^{(\ell)}\right]^T$.

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 (4.12). In this case $\rho_1 = \lambda_{\min}(\hat{\mathbf{R}}^{(\ell)})$ and $\rho_2 = \lambda_{\max}(\hat{\mathbf{R}}^{(\ell)})$, where $\lambda_{\min}(\hat{\mathbf{R}}^{(\ell)})$ and $\lambda_{\max}(\hat{\mathbf{R}}^{(\ell)})$ are the smallest and largest eigenvalues of $\hat{\mathbf{R}}^{(\ell)}$, respectively. It is a known fact that 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 (Strang, 1998). 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. 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. It follows from our discussion in Section 3.1.1, that the input ensemble is assured to be PE when we choose distinct and nonzero amplitude levels $a_1, a_2, \ldots, a_M$. □

4.1.3 The Equivalence of the Identification Algorithm to the Optimal Least-Squares Estimate

Let $\texttt{blkdiag}(\mathbf{A}, p)$ denote the block diagonal matrix which results from $p-$times block diagonal concatenation of the matrix $\mathbf{A}$. Using this definition (4.13) can be rewritten as

$$\hat{\boldsymbol{\mathcal{R}}}^{(\ell)} = \frac{1}{\mathcal{T}}\,\texttt{blkdiag}\Big(\hat{\mathbf{R}}^{(\ell)}, \tbinom{N+1}{\ell}\Big) \tag{4.16}$$

Using (4.15), (4.16) can be reformulated as,

$$\hat{\boldsymbol{\mathcal{R}}}^{(\ell)} = \frac{1}{\mathcal{T}}\,\texttt{blkdiag}\left(\mathbf{U}_e^{(\ell)^T}, \tbinom{N+1}{\ell}\right)\texttt{blkdiag}\left(\mathbf{U}_e^{(\ell)^*}, \tbinom{N+1}{\ell}\right) \tag{4.17}$$

From (4.8), (4.11) and (4.17), the least squares estimate of the kernel vector $\mathbf{h}^{(\ell)}$ for the ℓ-D input ensemble is given as follows.

$$\hat{\mathbf{h}}^{(\ell)^*} = \texttt{blkdiag}\Big(\big[\mathbf{U}_e^{(\ell)^*}\big]^{-1}, \tbinom{N+1}{\ell}\Big)\,\texttt{blkdiag}\Big(\big[\mathbf{U}_e^{(\ell)^T}\big]^{-1}, \tbinom{N+1}{\ell}\Big)\,\mathbf{X}^{(\ell)^H}(n)\,\mathbf{y}^{(\ell)}(n) \tag{4.18}$$

The matrix

$$\left[\texttt{blkdiag}\Big(\big[\mathbf{U}_e^{(\ell)^T}\big]^{-1}, \tbinom{N+1}{\ell}\Big)\, \mathbf{X}^{(\ell)^H}(n)\right] \tag{4.19}$$

in (4.18), is simply a permutation matrix, which changes the order of the output terms in $\mathbf{y}^{(\ell)}(n)$ and regroups them. The reordered output vector

$$\mathbf{v}^{(\ell)}(n)^* = \texttt{blkdiag}\Big(\big[\mathbf{U}_e^{(\ell)^T}\big]^{-1}, \tbinom{N+1}{\ell}\Big)\mathbf{X}^{(\ell)^H}(n)\mathbf{y}^{(\ell)}(n)$$

is a concatenation of the output ensemble vectors $\mathbf{v}_e^{(\ell,\ell)}(q_1, \ldots, q_{\ell-1}; n)$ which were defined in (3.75b). The least squares solution in (4.18) becomes

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

This result is equivalent to the ℓ-D kernel vector calculation equation in (3.75a). Hence, equation (3.75a) is indeed the least-squares solution for $\mathbf{h}^{(\ell)}(q_1, \ldots, q_{\ell-1}; n)$.

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.

4.2 UNIFORM EXCITATION OF VOLTERRA SYSTEM

The least squares solution necessitates the calculation of the inverse of the time-average correlation matrix. If the input is PE for the Volterra filter, than this inverse exists. However, even when the input is PE, this does not assure that all parts of the Volterra filter will get excited in a uniform manner. The input may excite some modes of the Volterra filter stronger than other modes. This difference in the strength of the excitation applied to particular modes is hidden in the singular values of the data matrix $\mathbf{X}^{(\ell)}(n)$ or the eigenvalues of the correlation matrix $\hat{\boldsymbol{\mathcal{R}}}^{(\ell)}$. Practically, the estimation error is determined by the noise power at the output of the system and the size of the eigenvalues of the time-average correlation matrix. Here we examine rather the ability of the deterministic sequences to uniformly excite all the modes of the Volterra filter. The condition to uniformly excite all modes of the ℓ-D substructure $\mathcal{H}^{(\ell)}$ requires to find an input level ensemble such that the difference among the sizes of the singular values of the data matrix is minimized. This difference in the the sizes of the singular values is quantified by the condition number of the data matrix. The condition number for a matrix $\mathbf{A}$ is defined as below.

$$\mathsf{cond}_2\ \mathbf{A} = \sqrt{\frac{\lambda_{\max}(\mathbf{A}^H\mathbf{A})}{\lambda_{\min}(\mathbf{A}^H\mathbf{A})}} \tag{4.20}$$

Hence, the condition number for $\mathbf{X}^{(\ell)}(n)$ will become,

$$\mathsf{cond}_2\ \mathbf{X}^{(\ell)}(n) = \sqrt{\frac{\lambda_{\max}(\hat{\boldsymbol{\mathcal{R}}}^{(\ell)})}{\lambda_{\min}(\hat{\boldsymbol{\mathcal{R}}}^{(\ell)})}}$$

We know that for the ℓ-D input sequence the eigenvalues of the time-average correlation matrix $\hat{\boldsymbol{\mathcal{R}}}^{(\ell)}$ are equal to the eigenvalues of the matrix $\hat{\mathbf{R}}^{(\ell)}$. Hence, to uniformly excite all modes of the ℓ-D substructure $\mathcal{H}^{(\ell)}$, the condition number of the matrix $\mathbf{U}_e^{(\ell)}$ should be minimized. Since we want uniformity of excitation for all the cross-term subsystems $\mathcal{H}^{(\ell)}$, $\ell = 1, \ldots, M$, the condition numbers for all the input matrices $\mathbf{U}_e^{(\ell)}$, $\ell = 1, \ldots, M$, should be minimized simultaneously. A collective criterion should be chosen to define what the optimally conditioned input matrices would be. The criterion we used to optimize the input level values was to minimize the maximum condition number among all the condition numbers. Other criteria, for example the mean-square of the condition numbers, could as well be used. The maximum condition numbers and the optimal level values are given in Table 4.1 for nonlinearity orders $M = 1, \ldots, 6$. Here the optimization was unconstrained; however, for the simulations in next section the input levels should also fulfill certain power constrains. Therefore, in the case of the simulations the optimal levels which are subject to certain average power constrains are found by constrained optimization.

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

4.3 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). To compare the performance with the existing algorithms, we have to use input lengths equal to the setups given in these. However, our required input ensemble length is much shorter than the required lengths for the PRMS and PSK inputs. In order to retain comparable input lengths, we resend our input ensembles until the total input length is close to those given in Zhou and Giannakis (1997) and Nowak and Van Veen (1994b) and we take the mean of the estimates coming from individual input cycles. The input impulse levels are chosen optimally as to satisfy the average power constrains of their respective examples, and to minimize the condition number criterion as defined in Section IV. The optimal level values are given in Table 4.2.

Simulation 4.1: We simulate a second order Volterra filter with memory length $N = 2$.

$$y(n) = \sum_{i_1=0}^{2} b_1(i_1)x(n-i_1) + \sum_{i_1=0}^{2}\sum_{i_2=i_1}^{2} b_2(i_1,i_2)x(n-i_1)x(n-i_2) \tag{4.21}$$

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. This setup is equivalent to the simulation in Nowak and Van Veen (1994b) with the difference that in the example there an extended Volterra filter had to be used, which is a drawback of using the Kronecker product representation. In Table 4.3 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

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

For PRMS the results are taken from Table IV in Nowak and Van Veen (1994b), and the average is over 200 independent trials. For our algorithm the average is also taken over 200 independent trials.

From the results in Table 4.3, it is clear that our algorithm uses less operations and gives better results than the PRMS method (Nowak and Van Veen, 1994b). To get comparable data lengths, we simply resent our input of length 15 and averaged over the estimates. This is reflected in the linear decrease of the mean square error for our algorithm. The length of the PRMS sequences are calculated as q^{N+1}, where $q > M$ is the order of the PRMS sequence. As we go down the Table 4.3, the order of the PRMS sequence used is increased as $q = 3, 4, 5, 7$, and the improvement in the condition number of the PRMS sample correlation matrix gets reflected in the error performance of the PRMS. Only after $q = 7$, PRMS has better error performance than our algorithm. However, it should not be overlooked that this example has a limited memory length $N = 2$, and for larger N using higher order PRMS would make the input length q^{N+1} prohibitively large. The reason why our algorithm performs better than the PRMS method in Table 4.3 is that our algorithm has an input diversity better than the PRMS input. The PRMS has to spend some of the allowed input power on input sequences, which are redundant for the identification of the regular Volterra filter, whereas our algorithm only includes the necessary input signals.

There are examples for the use of the PRMS in Volterra system identification also in (Nowak and Van Veen, 1994a). However, these examples illustrate the errors associated with the use of finite data in the absence of observation noise. In the case of PRMS, to get an exact least squares solution in the absence of observation noise, a full period of the PRMS, i.e., a sequence of length $(M+1)^{N+1}$ has to be used. For example, for a third-order filter ($M = 3$) with memory length 11 ($N = 11$), to get the exact solution with PRMS a data record of length 4^{12} is required. Use of shorter data records removes the PE property, and estimate errors occur as studied in Nowak and Van Veen (1994a). However, our algorithm is much more effective in the input diversity than PRMS, because our multilevel deterministic input completely eliminates those input combinations which are required for the identification of the extended Volterra filter but not for the regular Volterra system, which are included in the PRMS. For example, using (3.78) we can show that for the Volterra filter with ($M = 3$) and ($N = 11$) to get the exact kernel estimates our algorithm needs a data record of length less than $(2N+1)\binom{N+M}{M-1} = 2093$. This is a radical improvement over the required length 4^{12} for PRMS.

Simulation 4.2: We simulate a fourth-order Volterra filter with memory length $N = 3$ after the example 2 in Zhou and Giannakis (1997).

$$\begin{aligned} y(n) = x(n)^4 + 4.8x(n)^3x(n-1) + 4.8x(n)^2x(n-1)^2 \\ - 6x(n)^2x(n-1)x(n-2) + 14.4x(n)x(n-1)x(n-2)x(n-3) \end{aligned}$$

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. The optimal level values used in our multilevel sequence are given in Table 4.2. Table 4.4 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.

The PSK input method uses higher-order moments. In Example 1 of Zhou and Giannakis (1997), a third order filter with $N = 4$ is simulated and the mean and deviations of the estimates are examined in the absence of noise for an input length of 4096. Since our algorithm is an exact algorithm, for this filter our input sequence gives the *exact* kernel values for an input length as short as 155 in the absence of noise.

4.4 CONCLUDING REMARKS

In Chapters 3 and 4 we have developed a novel algorithm for input-output nonlinear system identification. The optimal estimation of Volterra kernels requires the solution of a large system of simultaneous linear equations. Our algorithm avoids the direct computation of the least squares estimate. By using a certain deterministic multilevel sequence and a novel partitioning of the Volterra kernels, the data correlation matrix reduces to a block diagonal matrix with relatively smaller dimensional matrices on the diagonal. These matrices depend only on the order of nonlinearity M and the input levels, hence they and their inverses can be calculated and stored beforehand. Our algorithm is in closed-form, exact, non-iterative and a computationally efficient implementation is also presented. It is shown that the input sequence is PE and the problem of ill-conditioning is also discussed. It is demonstrated with simulations that the algorithm can produce better parameter estimates than some existing algorithms (Nowak and Van Veen, 1994b; Zhou and Giannakis, 1997).

There are some other merits of this input sequence, which it shares with the PRMS. In some identification problems the normal operating input of the nonlinear system may be restricted to a fixed amplitude range. In this case the deterministic input sequence can utilize the full dynamic amplitude range of the normal operating interval of the system input. Another benefit of our algorithm is that it gives the Volterra kernel estimates directly. Some methods in the literature first calculate coefficients which have no directly interpretable results (i.e. the Wiener coefficients in Reed and Hawksford (1996)), and then obtain the Volterra kernels from these. An additional advantage is that our deterministic input sequence is suited to low rank approximations. The eigenvectors and the eigenvalues of the sample correlation matrix of the deterministic input sequence can be computed a priori as in the case of the PRMS.

Table 4.1: Optimal Deterministic Input Levels

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

Table 4.2: Optimal Input Levels for Simulations

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

Table 4.3: Averaged Squared Error of Estimates for Simulation 4.1

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}

Table 4.4: 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 $\hat{b}_4$ for PSK	0.9944	4.7927	4.8006	-6.0084	14.3892
std of $\hat{b}_4$ for PSK	0.2013	0.1940	0.1745	0.1766	0.1004
mean of $\hat{b}_4$ for our alg.	1.0002	4.7999	4.8000	-6.0003	14.4001
std of $\hat{b}_4$ for our alg.	0.0024	0.0042	0.0021	0.0038	0.0069

5. NONLINEAR CHANNEL IDENTIFICATION USING MULTILEVEL INPUT SIGNALS

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

Nonlinear channel identification is important in mitigating the effects of nonlinear distortions and for the equalization of the nonlinear communication channels (Agazzi et al., 1982; Benedetto et al., 1979; Falconer, 1978; Lazzarin et al., 1994). The performance of the efforts for compensation of nonlinearities and channel equalization are highly dependent on the accuracy of the nonlinear channel estimate. Hence, nonlinear channel identification has been a subject of significance (Ching and Powers, 1993; Özden et al., 1998; Tseng and Powers, 1993). In satellite communications, the amplifiers utilized are driven past their saturation region, due to power consumption considerations (Saleh, 1981). This inherently introduces nonlinearities into the satellite communication channel, and hence satellite communication has been a natural research area for nonlinear channel identification.

The Volterra series representation has been widely utilized to describe the input-output relationship of nonlinear systems. The truncated (or "doubly finite") Volterra filter representation can provide an approximation for a large class of nonlinear systems, and it is appropriate for modelling the nonlinearities encountered in communication systems. Hence, the characterization of nonlinear communication channels as finite order discrete Volterra filters has been studied in the literature.

The determination of the Volterra kernels of the nonlinear communication system model has been largely based on the use of input-output relations for random inputs (Mathews and Sicuranza, 2000). In the case of zero-mean Gaussian input, the closed form estimates for the Volterra kernels can be formed using the cross cumulants between the input and the output (Koukoulas and Kalouptsidis, 1995). However, the Gaussian assumption is far fetched for most communication applications. Moreover, the estimates are found in a sequential manner, which causes error propagation between kernel estimates. In Zhou and Giannakis (1997), PSK (phase shift keying) modulated input signals, which have vanishing higher order moments, have been studied as probing signals for nonlinear channels. A similar method has been given in Petrochilos and Comon (2000) for more general input sequences. Another Volterra kernel estimation algorithm based on cyclostationarity of communication signals is presented in Cheng and Powers (2001) where the kernels for a bandpass nonlinear channel are identified under i.i.d. M-QAM (quadrature amplitude modulation) and M-PSK inputs (Cheng, 1998). All of the methods presented depend on higher-order moments and cross correlations between input and output sequences.

In this work, we consider deterministic excitation sequences for the identification of nonlinear channels modelled using the third-order truncated Volterra series representation. The algorithm developed in this chapter is based on the novel Volterra system identification method presented in Chapter 3. However, in Chapter 3 only real input data was considered. Here, we will allow for complex inputs and complex Volterra channel kernels. Our algorithm provides exact, closed form solutions for the Volterra kernels. The results do not depend on statistics of random sequences; rather the algorithm utilizes specially designed deterministic sequences. Therefore, the algorithm shows better performance than the above random-input based methods even for input sequences of shorter length. The kernels are estimated separately, thus eliminating error propagation amongst the estimates. Complex inputs are allowed, hence PSK or QAM modulated communication signals can be utilized as the deterministic input sequence. In the context of a narrowband communication channel, the nonlinear system gets modelled by a "bandpass" Volterra series (Benedetto et al., 1987; Biglieri et al., 1988; Cheng and Powers, 2001; Raz and Van Veen, 1998). We will modify our identification algorithm for the identification of bandpass Volterra channels and present the corresponding simulation results.

5.1 NOVEL VOLTERRA CHANNEL REPRESENTATION

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

$$\begin{aligned} y(n) &= \mathcal{C}[x(n)] \\ &= \sum_{i_1=0}^{N} b_1(i_1)\,x(n-i_1) + \sum_{i_1=0}^{N}\sum_{i_2=i_1}^{N} b_2(i_1,i_2)\,x(n-i_1)x(n-i_2) \\ &+ \sum_{i_1=0}^{N}\sum_{i_2=i_1}^{N}\sum_{i_3=i_2}^{N} b_3(i_1,i_2,i_3)\,x(n-i_1)x(n-i_2)x(n-i_3) \end{aligned} \tag{5.1}$$

Here, $\mathcal{C}[\cdot]$ represents the nonlinear channel. N is the memory length of the nonlinear channel and $b_k(i_1,\ldots,i_k)$, $k=1,2,3$ is the Volterra kernel of degree k (Mathews and Sicuranza, 2000). We introduce a novel representation for the Volterra channel by rearranging the Volterra kernels. We reformulate the third-order discrete Volterra channel in terms of the multivariate cross-term complexity. The output $y(n)$ can be rewritten as the sum of the outputs of 3 different multivariate cross-term nonlinear subsystems, $\mathcal{H}^{(\ell)}$.

$$y(n) = \sum_{\ell=1}^{3} y^{(\ell)}(n) = \sum_{\ell=1}^{3} \mathcal{H}^{(\ell)}[x(n)] \tag{5.2a}$$

where

$$\mathcal{H}^{(1)}[x(n)] = \sum_{i=0}^{N} \mathbf{h}^{(1)^T}(i)\,\mathbf{x}^{(1)}(n-i) \tag{5.2b}$$

$$\mathcal{H}^{(2)}[x(n)] = \sum_{q_1=1}^{N}\sum_{i=0}^{N-q_1} \mathbf{h}^{(2)^T}(q_1;i)\,\mathbf{x}^{(2)}(q_1;n-i) \tag{5.2c}$$

$$\mathcal{H}^{(3)}[x(n)] = \sum_{q_1=1}^{N-1}\sum_{q_2=1}^{N-q_1}\sum_{i=0}^{N-q_1-q_2} \mathbf{h}^{(3)^T}(q_1,q_2;i)\mathbf{x}^{(3)}(q_1,q_2;n-i) \tag{5.2d}$$

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

$$\mathbf{h}^{(1)}(i) = \begin{bmatrix} b_1(i) & b_2(i,i) & b_3(i,i,i) \end{bmatrix}^T \tag{5.3a}$$

$$\mathbf{x}^{(1)}(n-i) = \begin{bmatrix} x(n-i) & x^2(n-i) & x^3(n-i) \end{bmatrix}^T \tag{5.3b}$$

$$\mathbf{h}^{(2)}(q_1;i) = \begin{bmatrix} b_2(i,i+q_1) \\ b_3(i,i,i+q_1) \\ b_3(i,i+q_1,i+q_1) \end{bmatrix} \tag{5.4a}$$

$$\mathbf{x}^{(2)}(q_1;n-i) = \begin{bmatrix} x(n-i)x\big(n-(i+q_1)\big) \\ x\big(n-i\big)^2 x\big(n-(i+q_1)\big) \\ x\big(n-i\big)x\big(n-(i+q_1)\big)^2 \end{bmatrix} \tag{5.4b}$$

$$\mathbf{h}^{(3)}(q_1,q_2;i) = \big[b_3\big(i,i+q_2,i+q_1+q_2\big)\big] \tag{5.5a}$$

$$\mathbf{x}^{(3)}(q_1,q_2;n-i) = \Big[x(n-i)x\big(n-i-q_1\big)x\big(n-i-(q_1+q_2)\big)\Big] \tag{5.5b}$$

where the limits for q_1, q_2 and i are as given in (5.2). We can see that the ℓ-D kernels correspond to an input vector, which can be factored into powers of ℓ distinct input terms with ℓ-distinct delays. Hence by cross-term complexity we consider the number of distinct input powers utilized in forming the terms of the input vectors and group the kernels accordingly. In the regular Volterra representation (5.1) on the other hand, the Volterra kernels are grouped together using the degree of the nonlinearity of the corresponding input terms. The introduced novel representation enables us to formulate an exact closed form algorithm to identify the Volterra kernels of a third-order Volterra channel utilizing deterministic input sequences.

5.2 IDENTIFICATION OF THE VOLTERRA CHANNEL USING DETERMINISTIC INPUT SIGNALS

In this section, we derive an efficient algorithm to identify the kernel vectors $\mathbf{h}^{(\ell)}$, $\ell = 1,2,3$ as defined in (5.2). The algorithm will utilize deterministic input sequences with 3 distinct levels (other than zero). All kernel vectors can be determined by using only the output of the system.

5.2.1 Identification of the 1-D Kernel Vectors

We will prove that an ensemble of three input sequences composed of single impulses with distinct values, $x^{(1)}(j;n) = a_j\delta(n)$, for $j = 1,2,3$ is adequate to obtain the 1-D kernel vectors in (5.3a). Looking at the cross-term representation in (5.2), we can see that for these single impulses the outputs of the 2-D and 3-D subsystems are zero, i.e., $\mathcal{H}^{(2)}[a_j\delta(n)] = 0$ and $\mathcal{H}^{(3)}[a_j\delta(n)] = 0$. Hence, the output of the overall nonlinear channel when the input is $x^{(1)}(j;n)$ is given by

$$\begin{aligned} \mathcal{C}\left[x^{(1)}(j;n)\right] &= \mathcal{H}^{(1)}\left[x^{(1)}(j;n)\right] \\ &= v^{(1,1)}(j;n) \\ &= \sum_{i=0}^{N} \mathbf{h}^{(1)^T}(i)\ \mathbf{u}^{(1)}(j;n-i) \end{aligned} \tag{5.6a}$$

where

$$\mathbf{u}^{(1)}(j;n) = \begin{bmatrix} a_j & a_j^2 & a_j^3 \end{bmatrix}^T \delta(n) \tag{5.6b}$$

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

$$\mathbf{y}_e^{(1)}(n) = \mathcal{H}^{(1)}[\mathbf{x}_e^{(1)}] = \sum_{i=0}^{N} \mathbf{U}_e^{(1)}(n-i)\ \mathbf{h}^{(1)}(i) \tag{5.7}$$

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

$$\begin{aligned} \mathbf{y}_e^{(1)}(n) &= \mathbf{v}_e^{(1,1)}(n) = \begin{bmatrix} v^{(1,1)}(1;n) \\ v^{(1,1)}(2;n) \\ v^{(1,1)}(3;n) \end{bmatrix} \\ \mathbf{x}_e^{(1)}(n) &= \begin{bmatrix} a_1\delta(n) \\ a_2\delta(n) \\ a_3\delta(n) \end{bmatrix} \\ \mathbf{U}_e^{(1)}(n) &= \begin{bmatrix} \mathbf{u}^{(1)^T}(1;n) \\ \mathbf{u}^{(1)^T}(2;n) \\ \mathbf{u}^{(1)^T}(3;n) \end{bmatrix} \end{aligned} \tag{5.8}$$

Here $\mathbf{u}^{(1)}(j;n)$ is as defined in (5.6b). We can replace $\mathbf{U}_e^{(1)}(n-i)$ in (5.7) with $\mathbf{U}_e^{(1)}\delta(n-i)$, where the matrix $\mathbf{U}_e^{(1)}$ is given as

$$\mathbf{U}_e^{(1)} = \begin{bmatrix} a_1 & a_1^2 & a_1^3 \\ a_2 & a_2^2 & a_2^3 \\ a_3 & a_3^2 & a_3^3 \end{bmatrix} \tag{5.9}$$

We can rewrite (5.7) as

$$\mathbf{y}_e^{(1)}(n) = \sum_{i=0}^{N} \mathbf{U}_e^{(1)}\ \mathbf{h}^{(1)}(i)\ \delta(n-i) = \mathbf{U}_e^{(1)}\ \mathbf{h}^{(1)}(n) \tag{5.10}$$

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

$$\mathbf{h}^{(1)}(n) = \left[\mathbf{U}_e^{(1)}\right]^{-1} \mathbf{y}_e^{(1)}(n); \quad \text{for } n = 0, 1, \ldots, N \tag{5.11}$$

The input matrix $\mathbf{U}_e^{(1)}$ in (5.9) is invertible if and only if the levels of the single impulse input sequences are distinct and nonzero, i.e., $a_i \neq 0$ and $a_i \neq a_j$, $\forall\, i \neq j$.

5.2.2 Identification Of The 2-D Kernel Vectors

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

$$\begin{aligned} \mathbf{x}_e^{(2)}(q_1; n) &= \begin{bmatrix} x^{(2)}\big((1,2), q_1; n\big) \\ x^{(2)}\big((1,3), q_1; n\big) \\ x^{(2)}\big((2,3), q_1; n\big) \end{bmatrix} \\ &= \begin{bmatrix} a_1\delta(n) + a_2\delta(n-q_1) \\ a_1\delta(n) + a_3\delta(n-q_1) \\ a_2\delta(n) + a_3\delta(n-q_1) \end{bmatrix} \end{aligned} \tag{5.12}$$

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

$$\begin{aligned} \mathbf{y}_e^{(2)}(q_1; n) &= \mathcal{H}^{(1)}\left[\mathbf{x}_e^{(2)}(q_1; n)\right] + \mathcal{H}^{(2)}\left[\mathbf{x}_e^{(2)}(q_1; n)\right] \\ &= \mathbf{v}_e^{(2,1)}(q_1; n) + \mathbf{v}_e^{(2,2)}(q_1; n) \end{aligned} \tag{5.13}$$

Using (5.2b) and (5.12), it is not difficult to show that the response of the 1-D subsystem to the 2-D input ensemble can be decomposed in terms of the 1-D responses as

$$\begin{aligned} \mathbf{v}_e^{(2,1)}(q_1; n) &= \mathcal{H}^{(1)} \begin{bmatrix} a_1\delta(n) \\ a_1\delta(n) \\ a_2\delta(n) \end{bmatrix} + \mathcal{H}^{(1)} \begin{bmatrix} a_2\delta(n-q_1) \\ a_3\delta(n-q_1) \\ a_3\delta(n-q_1) \end{bmatrix} \\ &= \begin{bmatrix} v^{(1,1)}(1; n) \\ v^{(1,1)}(1; n) \\ v^{(1,1)}(2; n) \end{bmatrix} + \begin{bmatrix} v^{(1,1)}(2; n-q_1) \\ v^{(1,1)}(3; n-q_1) \\ v^{(1,1)}(3; n-q_1) \end{bmatrix} \end{aligned} \tag{5.14}$$

Note that all the terms in (5.14) are output sequences which were found in section 3.1 in (5.6). We can obtain the response of the 2-D subsystem alone by subtracting the response of the 1-D subsystem from the nonlinear channel output vector.

$$\begin{aligned}\mathbf{v}_e^{(2,2)}(q_1;n) &= \begin{bmatrix} v^{(2,2)}\big((1,2),q_1;n\big) \\ v^{(2,2)}\big((1,3),q_1;n\big) \\ v^{(2,2)}\big((2,3),q_1;n\big)\end{bmatrix} \\ &= \mathbf{y}_e^{(2)}(q_1;n) - \mathbf{v}_e^{(2,1)}(q_1;n)\end{aligned} \tag{5.15}$$

On the other hand using (5.2c) we can write the 2-D subsystem output for our ensemble input as,

$$\mathbf{v}_e^{(2,2)}(q_1;n) = \mathbf{U}_e^{(2)}\mathbf{h}^{(2)}(q_1;n-q_1) \tag{5.16}$$

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

$$\mathbf{U}_e^{(2)} = \begin{bmatrix} a_1a_2 & a_1a_2^2 & a_1^2a_2 \\ a_1a_3 & a_1a_3^2 & a_1^2a_3 \\ a_2a_3 & a_2a_3^2 & a_2^2a_3 \end{bmatrix} \tag{5.17}$$

We can identify the 2-D Volterra kernel vectors by using (5.15) and (5.16),

$$\mathbf{h}^{(2)}(q_1;n-q_1) = \big[\mathbf{U}_e^{(2)}\big]^{-1}\mathbf{v}_e^{(2,2)}(q_1;n) \tag{5.18}$$

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

5.2.3 Identification Of The 3-D Kernel Vectors

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

$$x^{(3)}(q_1,q_2;n) = a_1\delta(n) + a_2\delta(n-q_2) + a_3\delta(n-q_1-q_2) \tag{5.19}$$

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

$$y^{(3)}(q_1,q_2;n) = \sum_{i=1}^{3} v^{(3,i)}(q_1,q_2;n) \tag{5.20}$$

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

$$v^{(3,1)}(q_1,q_2;n) = v^{(1,1)}(1;n) + v^{(1,1)}(2;n-q_2) + v^{(1,1)}(3;n-q_1-q_2) \tag{5.21}$$

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

$$\begin{aligned}v^{(3,2)}(q_1,q_2;n) = v^{(2,2)}&\big((1,2),q_2;n\big) + \\ &v^{(2,2)}\big((1,3),q_1+q_2;n\big) + v^{(2,2)}\big((2,3),q_1;n-q_2\big)\end{aligned} \tag{5.22}$$

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

$$v^{(3,3)}(q_1,q_2;n) = y^{(3)}(q_1,q_2;n) - v^{(3,1)}(q_1,q_2;n) - v^{(3,2)}(q_1,q_2;n) \tag{5.23}$$

On the other hand, in a similar fashion to the equations for the 1-D and 2-D subsystems $\big((5.10),(5.16)\big)$, the output of the 3-D subsystem $v^{(3,3)}(q_1,q_2;n)$ can be written as,

$$v^{(3,3)}(q_1,q_2;n) = \mathbf{U}_e^{(3)}\mathbf{h}^{(3)}(q_1,q_2;n-q_1-q_2) \tag{5.24}$$

where $\mathbf{U}_e^{(3)} = a_1a_2a_3$. Hence, we get

$$\mathbf{h}^{(3)}(q_1,q_2;n-q_1-q_2) = \left[\mathbf{U}_e^{(3)}\right]^{-1} v^{(3,3)}(q_1,q_2;n) \tag{5.25}$$

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

Fig. 5.1 depicts the identification of the Volterra channel using the proposed algorithm. In Fig. 5.1 bold lines symbolize vector signals, whereas plain lines suggest non-vector signals. The matrices $\mathbf{T}_{i,j}$ and the operators $\mathbb{S}_{i,j}[\,\cdot\,]$ shown in this figure are utilized to form the input ensembles and to choose the past output ensembles which get subtracted as in (5.15) and (5.23). The $\mathbf{T}_{i,j}$ matrices are given as,

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

The $\mathbb{S}_{i,j}[\,\cdot\,]$ operators are defined as

$$\begin{aligned}
\mathbb{S}_{2,1}\left[\mathbf{v}_e^{(1,1)}(n)\right] &= \mathbf{T}_{2,1}^{(M)}\mathbf{v}_e^{(1,1)}(n) + \mathbf{T}_{2,2}^{(M)}\mathbf{v}_e^{(1,1)}(n-q_1) \\
\mathbb{S}_{3,1}\left[\mathbf{v}_e^{(1,1)}(n)\right] &= v^{(1,1)}(1;n) + v^{(1,1)}(2;n-q_2) + v^{(1,1)}(3;n-q_1-q_2) \\
\mathbb{S}_{3,2}\left[\mathbf{v}_e^{(2,2)}(n)\right] &= v^{(2,2)}\big((1,2),q_2;n\big) + v^{(2,2)}\big((1,3),q_1+q_2;n\big) + v^{(2,2)}\big((2,3),q_1;n-q_2\big)
\end{aligned}$$

Using the approach in Chapter 3, the identification algorithm developed here can be extended to the identification of channels with nonlinearities higher than third order.

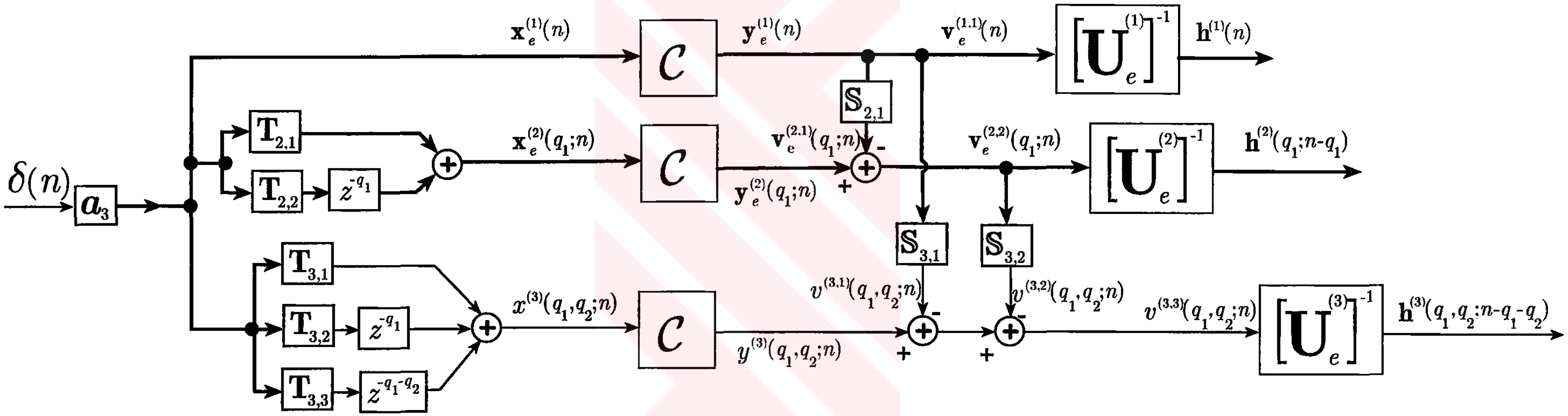


Figure 5.1: Proposed nonlinear communication channel identification method, utilizing third-order Volterra series model with multilevel deterministic sequences as inputs.

Example 5.1: We consider the third order nonlinear channel with $N = 2$ given by

$$\begin{aligned}y(n) = x(n) + 2x(n-1) - x^2(n) - 4x(n)x(n-1)- \\ 2x^2(n-1) + 3x^3(n) + 4x^2(n)x(n-1) + 5x(n)x^2(n-1)- \\ 3x^3(n-1) - 5x(n)x(n-2) + 6x(n)x(n-1)x(n-2)\end{aligned} \quad (5.26)$$

We want to find the Volterra kernels using the method outlined in this section. First we are interested in calculating the 1-D kernel vectors

$$\mathbf{h}^{(1)}(0) = \begin{bmatrix} b(0) \\ b(0,0) \\ b(0,0,0) \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \\ 3 \end{bmatrix}$$

and

$$\mathbf{h}^{(1)}(1) = \begin{bmatrix} b(1) \\ b(1,1) \\ b(1,1,1) \end{bmatrix} = \begin{bmatrix} 2 \\ -2 \\ -3 \end{bmatrix}^T$$

from the input and the output. We choose the impulse levels a_1, a_2, a_3 from the QPSK signal set $e^{j(2\pi k/4)}, k = 0, 1, 2, 3$. Hence, we define the 1-D input ensemble to the channel as

$$\mathbf{x}_e^{(1)}(n) = \begin{bmatrix} 1 \\ -1 \\ j \end{bmatrix} \delta(n) \quad (5.27)$$

The 1-D output ensemble from (5.8) becomes

$$\mathbf{y}_e^{(1)}(n) = \begin{bmatrix} 3 \\ -5 \\ 1-2j \end{bmatrix} \delta(n) + \begin{bmatrix} -3 \\ -1 \\ 2+5j \end{bmatrix} \delta(n-1) \quad (5.28)$$

From (5.9) and (5.27), the matrix $\mathbf{U}_e^{(1)}$ is written as,

$$\mathbf{U}_e^{(1)} = \begin{bmatrix} 1 & 1 & 1 \\ -1 & 1 & -1 \\ j & -1 & -j \end{bmatrix} \quad (5.29)$$

Applying (5.28) and (5.29) to (5.11) yields,

$$\begin{aligned}\mathbf{h}^{(1)}(0) = \left[\mathbf{U}_e^{(1)}\right]^{-1} \begin{bmatrix} 3 \\ -5 \\ 1-2j \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \\ 3 \end{bmatrix} \\ \mathbf{h}^{(1)}(1) = \left[\mathbf{U}_e^{(1)}\right]^{-1} \begin{bmatrix} -3 \\ -1 \\ 2+5j \end{bmatrix} = \begin{bmatrix} 2 \\ -2 \\ -3 \end{bmatrix}\end{aligned} \quad (5.30)$$

Next, we want to determine the 2-D kernel vectors

$$\mathbf{h}^{(2)}(1;0) = \begin{bmatrix} b(0,1) \\ b(0,0,1) \\ b(0,1,1) \end{bmatrix} = \begin{bmatrix} -4 \\ 4 \\ 5 \end{bmatrix}$$

and

$$\mathbf{h}^{(2)}(2;0) = \begin{bmatrix} b(0,2) \\ b(0,0,2) \\ b(0,2,2) \end{bmatrix} = \begin{bmatrix} -5 \\ 0 \\ 0 \end{bmatrix}$$

of the nonlinear channel given in (5.26). Using the 1-D input ensemble vector $\mathbf{x}_e^{(1)}(n)$ in (5.27), the 2-D input ensembles in (5.12) can be written as

$$\mathbf{x}_e^{(2)}(1;n) = \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix} \delta(n) + \begin{bmatrix} -1 \\ j \\ j \end{bmatrix} \delta(n-1) \tag{5.31}$$

$$\mathbf{x}_e^{(2)}(2;n) = \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix} \delta(n) + \begin{bmatrix} -1 \\ j \\ j \end{bmatrix} \delta(n-2) \tag{5.32}$$

For the 2-D input ensemble in (5.31), the output of the nonlinear channel in (5.26) is calculated as

$$\mathbf{y}_e^{(2)}(1;n) = \begin{bmatrix} 3 \\ 3 \\ -5 \end{bmatrix} \delta(n) + \begin{bmatrix} -5 \\ -6-j \\ 4+7j \end{bmatrix} \delta(n-1) + \begin{bmatrix} -1 \\ 2+5j \\ 2+5j \end{bmatrix} \delta(n-2) \tag{5.33}$$

It is possible to determine the response of the 1-D subsystem to the 2-D input ensemble in (5.31). To accomplish this we use the 1-D output ensemble given in (5.28) and (5.14).

$$\mathbf{v}_e^{(2,1)}(1;n) = \begin{bmatrix} 3 \\ 3 \\ -5 \end{bmatrix} \delta(n) + \begin{bmatrix} -8 \\ -2-2j \\ -2j \end{bmatrix} \delta(n-1) + \begin{bmatrix} -1 \\ 2+5j \\ 2+5j \end{bmatrix} \delta(n-2) \tag{5.34}$$

Using this result, the response of the 2-D subsystem can be obtained by subtracting $\mathbf{v}_e^{(2,1)}(1;n)$ from the nonlinear channel output $\mathbf{y}_e^{(2)}(1;n)$ in (5.33).

$$\begin{aligned} \mathbf{v}_e^{(2,2)}(1;n) &= \mathbf{y}_e^{(2)}(1;n) - \mathbf{v}_e^{(2,1)}(1;n) \\ &= \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \delta(n) + \begin{bmatrix} 3 \\ -4+j \\ 4+9j \end{bmatrix} \delta(n-1) + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \delta(n-2) \end{aligned} \tag{5.35}$$

The desired 2-D Volterra kernel $\mathbf{h}^{(2)}(1;0)$ can be calculated by substituting (5.35) into (5.18).

$$\begin{aligned}\mathbf{h}^{(2)}(1;0) &= \left[\mathbf{U}_e^{(2)}\right]^{-1}\mathbf{v}_e^{(2,2)}(1;1)\\ &= \begin{bmatrix} -1 & 1 & -1 \\ j & -1 & j \\ -j & 1 & j \end{bmatrix}^{-1} \begin{bmatrix} 3 \\ -4+j \\ 4+9j \end{bmatrix} = \begin{bmatrix} -4 \\ 4 \\ 5 \end{bmatrix}\end{aligned} \tag{5.36}$$

where the constant matrix $\mathbf{U}_e^{(2)}$ is obtained from $\mathbf{x}_e^{(1)}(n)$ and (5.17).

For the 2-D input ensemble in (5.32), the output of the nonlinear channel is calculated as

$$\mathbf{y}_e^{(2)}(2;n) = \begin{bmatrix} 3 \\ 3 \\ -5 \end{bmatrix} \delta(n) + \begin{bmatrix} -3 \\ -3 \\ -1 \end{bmatrix} \delta(n-1) + \begin{bmatrix} 0 \\ 1-7j \\ 1+3j \end{bmatrix} \delta(n-2) + \begin{bmatrix} -1 \\ 2+5j \\ 2+5j \end{bmatrix} \delta(n-3) \tag{5.37}$$

It is possible to determine the response of the 1-D subsystem to the 2-D input ensemble in (5.32). To accomplish this we use the 1-D output ensemble given in (5.28) and (5.14).

$$\mathbf{v}_e^{(2,1)}(2;n) = \begin{bmatrix} 3 \\ 3 \\ -5 \end{bmatrix} \delta(n) + \begin{bmatrix} -3 \\ -3 \\ -1 \end{bmatrix} \delta(n-1) + \begin{bmatrix} -5 \\ 1-2j \\ 1-2j \end{bmatrix} \delta(n-2) + \begin{bmatrix} -1 \\ 2+5j \\ 2+5j \end{bmatrix} \delta(n-3) \tag{5.38}$$

Using this result, the response of the 2-D subsystem can be obtained by subtracting $\mathbf{v}_e^{(2,1)}(2;n)$ from the nonlinear channel output $\mathbf{y}_e^{(2)}(2;n)$ in (5.37).

$$\begin{aligned}\mathbf{v}_e^{(2,2)}(2;n) &= \mathbf{y}_e^{(2)}(2;n) - \mathbf{v}_e^{(2,1)}(2;n)\\ &= \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \delta(n) + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \delta(n-1) + \begin{bmatrix} 5 \\ -5j \\ 5j \end{bmatrix} \delta(n-2) + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \delta(n-3)\end{aligned} \tag{5.39}$$

The desired 2-D Volterra kernel $\mathbf{h}^{(2)}(2;0)$ can be calculated by substituting (5.39) into (5.18).

$$\begin{aligned}\mathbf{h}^{(2)}(2;0) &= \left[\mathbf{U}_e^{(2)}\right]^{-1}\mathbf{v}_e^{(2,2)}(2;2)\\ &= \begin{bmatrix} -1 & 1 & -1 \\ j & -1 & j \\ -j & 1 & j \end{bmatrix}^{-1} \begin{bmatrix} 5 \\ -5j \\ 5j \end{bmatrix} = \begin{bmatrix} -5 \\ 0 \\ 0 \end{bmatrix}\end{aligned} \tag{5.40}$$

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

$$x^{(3)}(1,1;n) = 1 \cdot \delta(n) + (-1) \cdot \delta(n-1) + j \cdot \delta(n-2) \tag{5.41}$$

For the 3-D input ensemble in (5.41), the output of the nonlinear channel in (5.26) is calculated as

$$y^{(3)}(1,1;n) = 3\delta(n) - 5\delta(n-1) + (4-4j)\delta(n-2) + (2+5j)\delta(n-3) \quad (5.42)$$

It is possible to determine the response of the 1-D subsystem to the 3-D input ensemble in (5.41). The output of the 1-D subsystem is found by utilizing (5.21) and the 1-D output ensemble given in (5.28).

$$v^{(3,1)}(1,1;n) = 3\delta(n) - 8\delta(n-1) - 2j\delta(n-2) + (2+5j)\delta(n-3) \quad (5.43)$$

It is also possible to determine the response of the 2-D subsystem to the 3-D input in (5.41). We find the response of the 2-D subsystem by using (5.22) and the 1-D output ensemble given in (5.28).

$$v^{(3,2)}(1,1;n) = 3\delta(n) - 8\delta(n-1) - 2j\delta(n-2) + (2+5j)\delta(n-3) \quad (5.44)$$

Using these results, the response of the 3-D subsystem alone can be obtained by subtracting $v^{(3,1)}(1,1;n)$ and $v^{(3,2)}(1,1;n)$ from the nonlinear channel output $y^{(3)}(1;n)$ in (5.42).

$$\begin{aligned} v^{(3,3)}(1,1;n) &= y^{(3)}(1,1;n) - v^{(3,1)}(1;n) - v^{(3,2)}(1;n) \\ &= -6j\delta(n-2) \end{aligned} \quad (5.45)$$

The desired 3-D Volterra kernel $\mathbf{h}^{(3)}(1,1;0)$ can be calculated by substituting (5.45) into (5.25):

$$\begin{aligned} \mathbf{h}^{(3)}(1,1;0) &= \left[\mathbf{U}_e^{(3)}\right]^{-1} v^{(3,3)}(1,1;2) \\ &= (-j)^{-1}(-6j) = 6 \end{aligned} \quad (5.46)$$

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

5.2.4 Length of the Required Probing Signal

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

$$L = 3(N+1) + 3\sum_{q_{1,2}=1}^{N}(q_{1,2}+N+1) + \sum_{q_{1,3}=1}^{N-1}\sum_{q_{2,3}=1}^{N-q_{1,3}} q_{1,3}+q_{2,3}+N+1 \qquad (5.47)$$

Here, $q_{1,2}$ takes values between 1 and N as suggested by (5.2c). Similarly, the sum $q_{1,3}+q_{2,3}$ takes values between 2 and N, as suggested by (5.2d). Hence, both of these terms are upper-bounded by N, and we can write from (5.47)

$$\begin{aligned} L &\leqslant 3(N+1) + 3N(2N+1) + \frac{(N^2-N)}{2}(2N+1) \\ \Rightarrow \quad L &\leqslant (2N+1)\binom{N+3}{2} - 3N \end{aligned} \qquad (5.48)$$

Therefore, the required length of our total input sequence is upper-bounded by $(2N+1)\binom{N+3}{2}-3N$. For the channel given in Example 5.1 ($N=2$) the length of the overall input sequence was 41. This length is indeed less than $(2N+1)\binom{N+3}{2}-3N = 44$.

5.3 IDENTIFICATION OF BANDPASS VOLTERRA CHANNELS

The bandpass Volterra series is employed in the baseband representation of narrow-band communication channels. For bandpass communication signals, where the carrier frequency is much larger than the modulated channel bandwidth, the complex envelope of the nonlinear channel output signal gets described by a bandpass Volterra series rather than the regular Volterra filter representation as in (5.1) (Tseng, 1993). The even-order terms in the regular representation disappear, since they generate spectral components which fall outside the channel bandwidth and hence can be filtered by bandpass filter (Benedetto and Biglieri, 1983; Benedetto et al., 1987). The bandpass Volterra filter including nonlinearities up to third order is given as:

$$y(n) = \sum_{i_1=0}^{N} b_1(i_1)x(n-i_1) + \sum_{i_1=0}^{N}\sum_{i_2=0}^{N}\sum_{i_3=i_2}^{N} b_3(i_1,i_2,i_3)x^*(n-i_1)x(n-i_2)x(n-i_3) \tag{5.49}$$

Here, $(\cdot)^*$ denotes complex conjugation. N is the memory length of the bandpass nonlinear channel. $b_1(i_1)$ and $b_3(i_1,i_2,i_3)$ are the complex-valued linear and cubic bandpass Volterra kernels, respectively (Cheng and Powers, 2001).

We can easily modify the identification method we developed for the regular Volterra filter to the bandpass Volterra channel case. We will first reformulate the input-output relationship for the bandpass Volterra filter. The new representation will be similar to (5.2). The output $y(n)$ for the bandpass Volterra channel in (5.50) can be considered as the sum of the outputs of three different nonlinear subsystems, $\mathcal{H}^{(\ell)}$; that is

$$y(n) = \sum_{\ell=1}^{3} y^{(\ell)}(n) = \sum_{\ell=1}^{3} \mathcal{H}^{(\ell)}[x(n)] \tag{5.50a}$$

where

$$\mathcal{H}^{(1)}[x(n)] = \sum_{i=0}^{N} \mathbf{h}^{(1)^T}(i)\ \mathbf{x}^{(1)}(n-i) \tag{5.50b}$$

$$\mathcal{H}^{(2)}[x(n)] = \sum_{q_1=1}^{N}\sum_{i=0}^{N-q_1} \mathbf{h}^{(2)^T}(q_1;i)\ \mathbf{x}^{(2)}(q_1;n-i) \tag{5.50c}$$

$$\mathcal{H}^{(3)}[x(n)] = \sum_{q_1=1}^{N-1}\sum_{q_2=1}^{N-q_1}\sum_{i=0}^{N-q_1-q_2} \mathbf{h}^{(3)^T}(q_1,q_2;i)\mathbf{x}^{(3)}(q_1,q_2;n-i) \tag{5.50d}$$

The kernel vectors and the input vectors utilized in this representation can be given in terms of the bandpass Volterra channel kernels (5.50) and the input signal $x(n)$, respectively.

$$\mathbf{h}^{(1)}(i) = \begin{bmatrix} b_1(i) \\ b_3(i,i,i) \end{bmatrix} \tag{5.51}$$

$$\mathbf{x}^{(1)}(n) = \begin{bmatrix} x(n) \\ |x(n)|^2 x(n) \end{bmatrix} \tag{5.52}$$

$$\mathbf{h}^{(2)}(q_1;i) = \begin{bmatrix} b_3(i,i,i+q_1) \\ b_3(i+q_1,i,i+q_1) \\ b_3(i,i+q_1,i+q_1) \\ b_3(i+q_1,i,i) \end{bmatrix} \tag{5.53}$$

$$\mathbf{x}^{(2)}(q_1;n) = \begin{bmatrix} |x(n-i)|^2 x(n-(i+q_1)) \\ |x(n-(i+q_1))|^2 x(n-i) \\ x(n-i)^* x^2(n-(i+q_1)) \\ x(n-(i+q_1))^* x^2(n-i) \end{bmatrix} \tag{5.54}$$

$$\mathbf{h}^{(3)}(q_1,q_2;i) = \begin{bmatrix} b_3(i, i+q_1, i+q_1+q_2) \\ b_3(i+q_1, i, i+q_1+q_2) \\ b_3(i+q_1+q_2, i, i+q_1) \end{bmatrix} \tag{5.55}$$

$$\mathbf{x}^{(3)}(q_1,q_2;n) = \begin{bmatrix} x(n-i)^* x(n-(i+q_1)) x(n-(i+q_1+q_2)) \\ x(n-(i+q_1))^* x(n-i) x(n-(i+q_1+q_2)) \\ x(n-(i+q_1+q_2))^* x(n-i) x(n-(i+q_1)) \end{bmatrix} \tag{5.56}$$

Here, the limits for q_1, q_2 and i are as given in (5.50). This representation enables us to form an algorithm for identifying the bandpass Volterra kernels using deterministic sequences. The algorithm as we detailed in Section 3 can be used again for identification, however this time for the bandpass Volterra channel kernels. The sole difference will be in the matrices $\mathbf{U}_e^{(1)}$, $\mathbf{U}_e^{(2)}$ and $\mathbf{U}_e^{(3)}$. For the identification of the bandpass Volterra channel, these matrices will be given as

$$\mathbf{U}_e^{(1)} = \begin{bmatrix} a_1 & |a_1|^2 a_1 \\ a_2 & |a_2|^2 a_2 \end{bmatrix} \tag{5.57}$$

$$\mathbf{U}_e^{(2)} = \begin{bmatrix} |a_1|^2 a_2 & |a_2|^2 a_1 & a_1^* a_2^2 & a_2^* a_1^2 \\ |a_1|^2 a_3 & |a_3|^2 a_1 & a_1^* a_3^2 & a_3^* a_1^2 \\ |a_1|^2 a_4 & |a_4|^2 a_1 & a_1^* a_4^2 & a_4^* a_1^2 \\ |a_2|^2 a_3 & |a_3|^2 a_2 & a_2^* a_3^2 & a_3^* a_2^2 \end{bmatrix} \tag{5.58}$$

$$\mathbf{U}_e^{(3)} = \begin{bmatrix} a_1^* a_2 a_3 & a_2^* a_1 a_3 & a_3^* a_1 a_2 \\ a_1^* a_2 a_4 & a_2^* a_1 a_4 & a_4^* a_1 a_2 \\ a_2^* a_3 a_4 & a_3^* a_2 a_4 & a_4^* a_2 a_3 \end{bmatrix} \tag{5.59}$$

Other than these changes in the utilized matrices, the algorithm as detailed in Section 5.2 works also for the identification of the bandpass Volterra channel.

5.4 SIMULATIONS

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

$$y(n) = x(n) + 0.5x(n-1) - 0.8x(n-2) + x(n)^2 + 0.6x(n)x(n-1) - 0.3x(n-1)^2 + x(n)^3$$
$$+ 1.2x(n)^2 x(n-1) + 0.8x(n)x(n-1)^2 - 0.5x(n-1)^3 + x(n)x(n-1)x(n-2)$$

We use QPSK modulated signals as the input, where the deterministic input levels are chosen from the set $4e^{j(2\pi k/4+\pi/4)}, k = 0, 1, 2, 3$. Additive independent GWN observation noise with unit variance is present. Our required deterministic sequence is of length 41. In order to compare the performance of our algorithm, we also simulate the method given in Zhou and Giannakis (1997) for this setup. The PSK data length used for the method given in Zhou and Giannakis (1997) is 4096. Table 5.1 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). Note that the non-redundant kernels given in Table 5.1 are the triangular kernels. In the simulations in Zhou and Giannakis (1997), symmetric kernel values are used (Mathews and Sicuranza, 2000, 34). However, we utilized triangular kernel representation in our simulations to concord with our notation in (5.1). There are 11 nonzero Volterra kernels. The results for both methods are calculated over 400 independent trials. The results for our algorithm are better than those for the method of Zhou and Giannakis (1997) even though our method uses an input sequence of length almost 100 times smaller (41 vs. 4096). Our estimates are very accurate despite the presence of noise and the short length of input utilized.

The formulas given in Zhou and Giannakis (1997) for the calculation of the Volterra kernels of orders 1 through 4 are irrespective of the other nonlinear kernels present in the channel. Hence, they would be valid even if there are higher order kernels present. However, our algorithm presented here is based on the premise that the channel is of third order. This is one advantage of the algorithm in Zhou and Giannakis (1997) over our algorithm. However, as stated earlier it is possible to extend our algorithm to channels with nonlinearity order higher than 3 using the results in Chapter 3. On the other hand, the method developed in Zhou and Giannakis (1997) utilizes higher-order moments. This introduces errors in the estimates for finite length input sequences even in the absence of noise. In Example 1 of Zhou and Giannakis (1997), a third order filter with $N = 4$ is simulated and the mean and deviations of the estimates are examined in the absence of noise for an input length of 4096. Since our algorithm is exact, for this filter our input sequence gives the *exact* kernel values for an input length as short as 155 in the absence of noise.

Simulation 5.2: We simulate a linear-cubic "bandpass" Volterra filter, where the input-output relationship for the bandpass Volterra filter is given in (5.49).

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

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

We also realized the method for bandpass Volterra kernel identification as given in Cheng and Powers (2001) for the simulation setup given above. The PSK data length used for the method of Cheng and Powers (2001) is 4096. When PSK data is utilized, the bandpass Volterra filter simplifies considerably, since some of the third order kernels degenerate into first order kernels. Table 5.2 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 non-redundant kernels are the kernels given as $b_3(i, j, k)$, $b_3(i, k, k)$ and $b_1(i)$ (Cheng and Powers, 2001). There are a total of 10 Volterra kernels. The results for both methods are calculated over 400 independent trials. The results for our algorithm are better than those for the method of Cheng and Powers (2001) even though our method employed an input sequence of shorter length.

We presented a novel method for input-output identification of the Volterra kernels of a nonlinear channel modelled as a third-order Volterra filter. Our method utilizes carefully designed deterministic sequences as the probing signal and avoids the shortcomings of the use of random signals and correlation methods. The algorithm works also for complex inputs, allowing the use of complex baseband communication signals. We give simulation examples demonstrating the performance of the algorithm compared to methods available in the literature. These methods can be easily extended to the identification of nonlinear channels with higher-order nonlinearities following the results given in Chapter 3.

Table 5.1: 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

(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

Table 5.2: 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

(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

(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

6. ADAPTIVE VOLTERRA FILTERING USING COMPLETE LATTICE ORTHOGONALIZATION

This chapter presents 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. Experimental results indicate that the performance of this method is better than previously published adaptive Volterra filter realization structures.

The truncated Volterra series expansion provides a commonly used nonlinear model for adaptive nonlinear filtering applications (Kajikawa, 2000; Mathews, 1991). In the so called Volterra filter model, the output of a causal, discrete-time, time-invariant nonlinear system is expressed as a function of the input using the Volterra filter expansion. However, polynomial regression models such as the truncated Volterra expansion suffer from ill-conditioning, especially if the degree of the polynomial is high. It is known that even when the input signal is white, the presence of the nonlinear terms in the input vector will cause the eigenvalue spread of the data correlation matrix to be more than one. It is important to seek for alternate Volterra filter realizations that have better numerical properties. Orthogonal realizations are structures with good numerical behavior (Chen et al., 1989; Korenberg and Paarman, 1991; Ling and Proakis, 1984). Lattice models provide viable orthogonalized structures for nonlinear Volterra filtering (Koh and Powers, 1985; Lee and Mathews, 1993; Mathews, 1991; Syed and Mathews, 1993, 1994). An order recursive solution for the 2-D normal equations has been recently developed by Kayran (1996b). This solution has lead to an efficient orthogonal lattice predictor for two-dimensional (2-D) filtering. This structure has been previously utilized to develop methods for 2-D FIR Wiener filtering

(Kayran and Ekşioğlu, 2000) and 2-D ARMA modelling (Kayran, 1996a). In this chapter, we apply the fully orthogonal 2-D lattice structure of Kayran (1996b) to the realization and identification of truncated Volterra series expansion based nonlinear systems. This model can be applied to Volterra systems with arbitrary input signals and arbitrary supports for the Volterra kernels, as opposed to some proposed models which are orthogonalizing only for special, e.g. Gaussian input signals and special shapes for the Volterra kernel support (Mathews, 1991).

6.1 2-D LATTICE ANALYSIS MODEL

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

$$d(n) = \sum_{i_1=0}^{N-1} b_1(i_1;n)\; x(n-i_1) + \sum_{i_1=0}^{N-1}\sum_{i_2=i_1}^{N-1} b_2(i_1,i_2;n)\; x(n-i_1)x(n-i_2) \quad (6.1)$$

Here, the nonlinear system is possibly time-variant, because the Volterra kernels depend on the time index n as opposed to the representation as given in (2.8). N is the memory length of the nonlinear system. $b_1(i_1;n)$ and $b_2(i_1,i_2;n)$ are respectively the linear and quadratic Volterra kernels of the second order Volterra filter. It is possible to describe the input-output relationship given in (6.1) as a pseudo-linear regression in the form of a vector product.

$$d(n) = \mathbf{X}_2^T(n)\mathbf{B}_2(n) \quad (6.2)$$

Hence, we reformulated (6.1) in the form of a single vector product. Here, $\mathbf{X}_2(n)$ will be a vector containing all the single input samples and products of pairs of input samples, as included in (6.1). $\mathbf{X}_2(n)$ will be of the form as described in the following equation.

$$\mathbf{X}_2(n) = \begin{bmatrix} x(n) \\ x(n-1) \\ \vdots \\ x(n-N+1) \\ x(n)^2 \\ x(n)x(n-1) \\ \vdots \\ x(n-N+1)^2 \end{bmatrix} \quad (6.3)$$

$\mathbf{B}_2(n)$ is a vector which contains all the Volterra kernels as required in (6.1). The kernels are arranged in the vector $\mathbf{B}_2(n)$ such that the location of the kernels are identical of the position of the corresponding input sample product term. Hence, $\mathbf{B}_2(n)$ will be given as,

$$\mathbf{B}_2(n) = \begin{bmatrix} b_1(0;n) \\ b_1(1;n) \\ \vdots \\ b_1(N-1;n) \\ b_2(0,0;n) \\ b_2(0,1;n) \\ \vdots \\ b_2(N-1,N-1;n) \end{bmatrix} \tag{6.4}$$

The total number of Volterra kernels is given by

$$M = N + \binom{N+1}{2} = \frac{N(N+3)}{2} \tag{6.5}$$

The direct form realization as indicated by (6.1) and (6.2) 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; Özden et al., 1996a; Syed and Mathews, 1994). Both methods in Syed and Mathews (1994) and Özden et al. (1996a) are based on the multichannel lattice structure as developed in Ling and Proakis (1984). These methods convert the input signal vector as given in (6.3) into a multichannel signal and apply orthogonalization onto the multichannel signal while calculating the nonlinear output. The orthogonalization procedure results in improved performance and good numerical properties in adaptive filtering applications.

In this chapter we restructure the expanded input signal vector $\mathbf{X}_2(n)$ into a 2D array as shown in Fig. C.1, rather than using a multichannel setup. The ordering of the M (6.5) data points in the support region can be made in various ways. Fig. C.2 shows the indexing arrangement we chose for the input array. 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 (6.1). Fig. C.3 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. In this figure, the lattice section is a 2D lattice predictor for the underlying expanded input signal array $\mathbf{X}_2(n)$ and the ladder section is a linear combination of the orthogonalized backward prediction errors (Kayran, 1996b). We use the ladder-section of the joint-process estimator to form an estimate of some primary input $d(n)$ using the backward prediction errors. In the system identification mode this input will be the output from a second-order Volterra filter. The internal structure of the basic lattice module utilized in this lattice orthogonalization-based novel nonlinear structure is shown in Fig. C.4.

We reshape the vector $\mathbf{X}_2(n)$ into a 2D array using the proposed ordering as in Fig. C.1. We can observe from the ordering scheme in Fig. C.1 that the 2D array exhibits a 1D shift-invariance in the horizontal direction for each row. This ordering was chosen on purpose to eliminate some lattice sections in the full-complexity 2D lattice filter. Using this shift-invariance property along the rows, we can perform the orthogonalization within each row utilizing regular 1D FIR lattice filter. The resulting 2D lattice predictor with significantly decreased number of lattice sections is the reduced complexity 2D orthogonal lattice filter (Kayran, 1996b). Fig. C.5 depicts the nonlinear joint-process estimator, complete with the reduced complexity lattice predictor and ladder section for the case $N = 4$. The description for the lattice blocks $\boldsymbol{\Pi}(N, K)$ utilized in this reduced complexity structure is shown in Fig. C.6.

It is possible to calculate the corresponding Volterra kernels for a given lattice model following the results as described in Kayran (1996b). Hence, we can calculate the Volterra kernels from the reflection coefficients and the ladder coefficients of the 2D lattice based structure. Since the expanded input vector is reshaped into a 2D array as given in Fig. C.1, we can calculate the Volterra kernels as the coefficients of a 2D transfer function relating the input $x(n)$ to the output $y(n) = \hat{d}(n)$. We first need to calculate the transfer functions

$$G_p^{(p)}(z_1, z_2) = \frac{B_p^p(z_1, z_2)}{X(z_1, z_2)}$$

These are the transfer functions between the input and the backward prediction errors $b_p^{(p)}$. After calculating these transfer functions, the overall transfer function will be found by adding the ladder section. To this end we will multiply these transfer functions with the corresponding ladder coefficients and add them together. The overall algorithm for calculating the Volterra kernels from the 2D lattice filter coefficients will be outlined as below. Here, $\mathbf{g}_p^{(m)}$ and $\mathbf{a}_p^{(m)}$ define the coefficient vectors for the transfer functions $G_p^{(m)}(z_1, z_2) = \dfrac{B_p^{(m)}(z_1, z_2)}{X(z_1, z_2)}$ and $A_p^{(m)}(z_1, z_2) = \dfrac{F_p^{(m)}(z_1, z_2)}{X(z_1, z_2)}$, respectively (Kayran, 1996b).

Algorithm 6.1 Algorithm for calculating the Volterra kernels from the 2D lattice filter coefficients:

1. Set $m = 0$. Define the initial coefficient vectors as

$$\mathbf{a}_p^{(0)} = \mathbf{g}_p^{(0)} = [1] \quad \text{for } p = 0, 1, 2, \cdots, M$$

2. Increasing m by one, we calculate $\hat{\mathbf{a}}_{p-m}^{(m)}$ and $\hat{\mathbf{g}}_p^{(m)}$ from $\hat{\mathbf{a}}_{p-m}^{(m-1)}$ and $\hat{\mathbf{g}}_p^{(m-1)}$ using the augmented vectors

$$\hat{\mathbf{a}}_{p-m}^{(m)} = \begin{bmatrix} \hat{\mathbf{a}}_{p-m}^{(m-1)} \\ 0 \end{bmatrix};$$

$$\hat{\mathbf{g}}_p^{(m)} = \begin{bmatrix} 0 \\ \hat{\mathbf{g}}_p^{(m-1)} \end{bmatrix}; \quad \text{for } p = m, m+1, \cdots, M$$

3. The $m^{\text{th}}-$stage forward and backward 2D transfer function coefficient vectors are calculated as

$$\mathbf{a}_{p-m}^{(m)} = \hat{\mathbf{a}}_{p-m}^{(m)} + \Gamma_{b_p}^{(m)} \hat{\mathbf{g}}_p^{(m)}$$

$$\mathbf{g}_p^{(m)} = \hat{\mathbf{g}}_p^{(m)} + \Gamma_{f_{p-m}}^{(m)} \hat{\mathbf{a}}_{p-m}^{(m)}$$

This step is continued stage by stage for $p = m, m+1, \cdots, M$.

4. We form the coefficient vector for the overall transfer function by multiplying the proper backward 2D transfer function coefficient vectors with the corresponding ladder coefficients and adding them together. Here v will be the resulting vector which gives the Volterra kernels.

$$\mathbf{v} = \sum_{p=0}^{M} c_p \, \mathbf{g}_p^{(p)}$$

6.2 NONLINEAR SYSTEM IDENTIFICATION USING RLS ADAPTATION

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 in gradient descent type of algorithms such as (normalized) LMS. However, here we develop an RLS-type adaptive algorithm for our novel structure to permit comparison with the algorithms in (Syed and Mathews, 1994).

The orthogonalization within each row of the input array along the horizontal direction utilizes 1D FIR lattice filters. Hence, for the adaptation of the reflection coefficients in each of this 1D lattice filters we used the regular RLS lattice adaptation using a posteriori estimation errors (Manolakis et al., 2000). This provides for fast, computationally simple and numerically robust adaptation for the 1D lattice sections. For the reflection coefficients in the lattice module sections which provide for the inter-row orthogonalization, the shift invariance is no longer valid. However, the 2D lattice predictor ensures that the backward prediction and forward prediction errors, which are the inputs to the lattice modules, are fully orthogonalized. We can employ this decoupling property of the lattice structure and develop RLS adaptation algorithms for each single lattice stage inside the lattice modules separately. The RLS time update equations for the backward and forward prediction reflection coefficients are given in Table C.1 where n is the time index. We set $P(0) = \delta^{-1}$, where δ is a small positive constant and $e(0) = 0$. The decoupling property of the lattice filter is valid for both the backward and forward predictors, and the ladder section. Hence, we use similar update equations for the backward prediction, forward prediction and ladder coefficients.

In Table C.1, $\Gamma_{f_{p-m}}^{(m)}(n)$ and $\Gamma_{b_p}^{m}(n)$ are the real-valued m^{th} stage reflection coefficients for the forward and backward predictors, respectively (Kayran, 1996b).

The decoupling property of the lattice filter is also valid for the backward prediction errors $b_0^{(0)}(n)$, $b_1^{(1)}(n)$, $\ldots$, $b_M^{(M)}(n)$ which are fed into the ladder section. Hence, the backward prediction errors are orthogonal to each other. Therefore, the update equations for the ladder coefficients $c_m(n)$ are similar to the equations used for the reflection coefficients.

6.3 SIMULATIONS

The setting for the simulations is shown in Fig. C.7. 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 values of the kernels are taken from system in the example in Syed and Mathews (1994) by truncating the system given in Syed and Mathews (1994) to $N = 4$. The kernel values are given in Table C.2. The input signal $x(n)$ to the unknown system was colored Gaussian noise obtained as given in Syed and Mathews (1994) by filtering white Gaussian random noise using an FIR filter with the impulse response $h(n) = [0.9045, \underset{\uparrow}{1}, 0.9045]$. 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. We present the learning curves in Fig. C.8, 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. C.8. The error curves are mean squared for 500 cycles and $\lambda = 0.9975$. Our structure maintains the excellent numerical behavior of the lattice models. There is some deterioration in the convergence speed of the multichannel lattice model of Syed and Mathews (1994) when compared to the performance of the transversal RLS filter, which is due to the increased complexity of the multichannel lattice structure (Syed and Mathews, 1994). However, our structure exhibits better performance than both the transversal realization and the multichannel lattice structure.

We also did calculate the direct form Volterra coefficients from the lattice reflection coefficients using using Algorithm 6.1 outlined in the previous section. Fig.C.9 plots the norm of the coefficient error vector for the linear terms $b_1(i_1)$. Fig. C.10 shows the time-evolution of the norm of the coefficient error vector for the quadratic terms $b_2(i_1, i_2)$. These norms are calculated as follows, where $\hat{b}_1(i_1; n)$ and $\hat{b}_2(i_1, i_2; n)$ denote the Volterra kernel estimates at time n.

$$\|e_1(n)\| = \sqrt{\sum_{i_1=0}^{N-1}\left[b_1(i_1) - \hat{b}_1(i_1; n)\right]^2}$$

$$\|e_2(n)\| = \sqrt{\sum_{i_1=0}^{N-1}\sum_{i_2=i_1}^{N-1}\left[b_2(i_1, i_2) - \hat{b}_2(i_1, i_2; n)\right]^2}$$

Fig. C.9 and Fig. C.10 show that the lattice structure converges to the correct Volterra filter coefficients in addition to the convergence in the mean square error sense.

In this chapter, a new method is developed for the realization of the second-order Volterra filters utilizing the 2-D orthogonal lattice structure (Kayran, 1996b). An RLS adaptive nonlinear filter based on this structure is presented. The performance of our structure in the adaptive system identification setting is superior to previous models (Mathews, 1991; Syed and Mathews, 1994). We plan to apply the proposed approach also to the realization of nonlinear Volterra filters with feedback, especially the bilinear system model (Baik and Mathews, 1993). This will be the subject of forthcoming work. We restricted our discussion in this chapter to adaptive quadratic Volterra filters. However, the method presented can be applied in the realization of finite memory Volterra filters with higher-order nonlinearities.

7. 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. We introduced a novel fully orthogonal realization for Volterra filters utilizing the 2D lattice structure.

The work on the identification of the Volterra filters using deterministic impulse sequences seems rather complete. 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. The identification algorithm presented in this dissertation may facilitate better Volterra kernel estimates for short input sequences. This will help in devising better Volterra-based compensators, because the quality of such compensators depend on the accuracy of the Volterra kernel estimates for the nonlinear system.

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.

APPENDIX A

Table A.1: Recursive Algorithm for the Calculation of the Ensemble Input Formation Matrices.

Define:

$$\mathbf{T}_{1,1}^{(p)} = \mathbf{I}_p$$

$$\boldsymbol{i}_k = [1 \ \cdots \ 1]_{1\times k}^T$$

$$\boldsymbol{\Pi}_{k,i} = \begin{bmatrix} 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \end{bmatrix}_{1\times k},$$

where only i^{th}-entry is unity.

---

for $\ell = 2, 3, \ldots, M$

for $i = 1, 2, \ldots, \ell$

$$\mathbf{T}_{\ell,i}^{(\ell)} = \boldsymbol{\Pi}_{\binom{\ell}{1},i}$$

endfor

if $\ell \leq M - 1$

for $p = \ell, \ldots, M - 1$

$$\mathbf{T}_{\ell,1}^{(p+1)} = \begin{bmatrix} \boldsymbol{i}_{\binom{p}{\ell-1}} & \mathbf{0}_{\binom{p}{\ell-1}\times p} \\ \mathbf{0}_{\binom{p}{\ell}\times 1} & \mathbf{T}_{\ell,1}^{(p)} \end{bmatrix}_{\binom{p+1}{\ell}\times(p+1)}$$

for $i = 2, 3, \ldots, \ell$

$$\mathbf{T}_{\ell,i}^{(p+1)} = \begin{bmatrix} \mathbf{0}_{\binom{p}{\ell-1}\times 1} & \mathbf{T}_{\ell-1,i-1}^{(p)} \\ \mathbf{0}_{\binom{p}{\ell}\times 1} & \mathbf{T}_{\ell,i}^{(p)} \end{bmatrix}_{\binom{p+1}{\ell}\times(p+1)}$$

endfor

endfor

endif

endfor

Table A.2: All Ensemble Input Formation Matrices for $M = 4$. The matrices are generated using the algorithm in Table A.1.

$$\mathbf{T}_{1,1}^{(1)} = \mathbf{I}_1 = \begin{bmatrix} 1 \end{bmatrix}$$

$$\mathbf{T}_{1,1}^{(2)} = \mathbf{I}_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\mathbf{T}_{1,1}^{(3)} = \mathbf{I}_3 \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{T}_{1,1}^{(4)} = \mathbf{I}_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{T}_{2,1}^{(2)} = \boldsymbol{\Pi}_{\binom{2}{1},1} = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

$$\mathbf{T}_{2,2}^{(2)} = \boldsymbol{\Pi}_{\binom{2}{1},2} = \begin{bmatrix} 0 & 1 \end{bmatrix}$$

$$\mathbf{T}_{2,1}^{(3)} = \begin{bmatrix} \boldsymbol{i}_{\binom{2}{1}} & \mathbf{0}_{\binom{2}{1}\times 2} \\ \mathbf{0}_{\binom{2}{2}\times 1} & \mathbf{T}_{2,1}^{(2)} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

$$\mathbf{T}_{2,2}^{(3)} = \begin{bmatrix} \mathbf{0}_{\binom{2}{1}\times 1} & \mathbf{T}_{1,1}^{(2)} \\ \mathbf{0}_{\binom{2}{2}\times 1} & \mathbf{T}_{2,2}^{(2)} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{T}_{2,1}^{(4)} = \begin{bmatrix} \boldsymbol{i}_{\binom{3}{1}} & \mathbf{0}_{\binom{3}{1}\times 3} \\ \mathbf{0}_{\binom{3}{3}\times 1} & \mathbf{T}_{3,1}^{(3)} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\mathbf{T}_{2,2}^{(4)} = \begin{bmatrix} \mathbf{0}_{\binom{3}{1}\times 1} & \mathbf{T}_{1,1}^{(3)} \\ \mathbf{0}_{\binom{3}{3}\times 1} & \mathbf{T}_{2,2}^{(3)} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{T}_{3,1}^{(3)} = \boldsymbol{\Pi}_{\binom{3}{1},1} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$$

$$\mathbf{T}_{3,2}^{(3)} = \boldsymbol{\Pi}_{\binom{3}{1},2} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}$$

$$\mathbf{T}_{3,3}^{(3)} = \boldsymbol{\Pi}_{\binom{3}{1},3} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{T}_{3,1}^{(4)} = \begin{bmatrix} \boldsymbol{i}_{\binom{3}{2}} & \mathbf{0}_{\binom{3}{2}\times 3} \\ \mathbf{0}_{\binom{3}{3}\times 1} & \mathbf{T}_{3,1}^{(3)} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

$$\mathbf{T}_{3,2}^{(4)} = \begin{bmatrix} \mathbf{0}_{\binom{3}{2}\times 1} & \mathbf{T}_{2,1}^{(3)} \\ \mathbf{0}_{\binom{3}{3}\times 1} & \mathbf{T}_{3,2}^{(3)} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\mathbf{T}_{3,3}^{(4)} = \begin{bmatrix} \mathbf{0}_{\binom{3}{2}\times 1} & \mathbf{T}_{2,2}^{(3)} \\ \mathbf{0}_{\binom{3}{3}\times 1} & \mathbf{T}_{3,3}^{(3)} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{T}_{4,1}^{(4)} = \boldsymbol{\Pi}_{\binom{4}{1},1} = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{T}_{4,2}^{(4)} = \boldsymbol{\Pi}_{\binom{4}{1},2} = \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix}$$

$$\mathbf{T}_{4,3}^{(4)} = \boldsymbol{\Pi}_{\binom{4}{1},3} = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\mathbf{T}_{4,4}^{(4)} = \boldsymbol{\Pi}_{\binom{4}{1},4} = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}$$

APPENDIX B

Table B.1: Recursive Algorithm for the Calculation of the Ensemble Output Pick Up Matrices.

Define Initial Matrices:

for $p = 2, 3, \ldots, M$
  for $\ell = 1, 2, \ldots, p$
    for $k = 1, 2, \ldots, \ell$
      for $i = 1, 2, \ldots, \binom{\ell}{1}$
        $\mathbf{S}^{(p)}_{\ell 1,i} = \mathbf{T}^{(p)}_{\ell,i}$
      endfor
      for $j = 1, 2, \ldots, \binom{p}{k}$
        $\mathbf{S}^{(p)}_{pk,j} = \boldsymbol{\Pi}_{\binom{p}{k},j}$
      endfor
    endfor
    $\mathbf{S}^{(p)}_{\ell\ell,1} = \mathbf{I}_{\binom{p}{\ell}}$
  endfor
endfor

---

for $p = 4, 5, \ldots, M$
  for $\ell = 3, 4, \ldots, p-1$
    for $k = 2, 3, \ldots, \ell - 1$
      for $i = 1, 2, \ldots, \binom{\ell-1}{k-1}$

$$\mathbf{S}^{(p)}_{\ell k,i} = \begin{bmatrix} \mathbf{S}^{(p-1)}_{(\ell-1)(k-1),i} & \mathbf{0}_{\binom{p-1}{\ell-1}\times\binom{p-1}{k}} \\ \mathbf{0}_{\binom{p-1}{\ell}\times\binom{p-1}{k-1}} & \mathbf{S}^{(p-1)}_{\ell k,i} \end{bmatrix}_{\binom{p}{\ell}\times\binom{p}{k}}$$

      endfor
      for $j = 1, 2, \ldots, \binom{\ell-1}{k}$

$$\mathbf{S}^{(p)}_{\ell k,\binom{\ell-1}{k-1}+j} = \begin{bmatrix} \mathbf{0}_{\binom{p-1}{\ell-1}\times\binom{p-1}{k}} & \mathbf{S}^{(p-1)}_{(\ell-1)k,j} \\ \mathbf{0}_{\binom{p-1}{\ell}\times\binom{p-1}{k-1}} & \mathbf{S}^{(p-1)}_{\ell k,\binom{\ell-1}{k-1}+j} \end{bmatrix}_{\binom{p}{\ell}\times\binom{p}{k}}$$

      endfor
    endfor
  endfor
endfor

Table B.2: All Ensemble Output Pick Up Matrices for $M = 4$, generated using Table B.1.

$$\mathbf{S}^{(2)}_{11,1} = \mathbf{T}^{(2)}_{1,1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\mathbf{S}^{(2)}_{21,1} = \mathbf{T}^{(2)}_{2,1} = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

$$\mathbf{S}^{(2)}_{21,2} = \mathbf{T}^{(2)}_{2,2} = \begin{bmatrix} 0 & 1 \end{bmatrix}$$

$$\mathbf{S}^{(2)}_{22,1} = \mathbf{I}_{\binom{2}{2}} = \begin{bmatrix} 1 \end{bmatrix}$$

$$\mathbf{S}^{(3)}_{11,1} = \mathbf{T}^{(3)}_{1,1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{S}^{(3)}_{31,1} = \boldsymbol{\Pi}_{\binom{3}{1},1} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$$

$$\mathbf{S}^{(3)}_{31,2} = \boldsymbol{\Pi}_{\binom{3}{1},2} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}$$

$$\mathbf{S}^{(3)}_{31,3} = \boldsymbol{\Pi}_{\binom{3}{1},3} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{S}^{(3)}_{21,1} = \mathbf{T}^{(3)}_{2,1} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

$$\mathbf{S}^{(3)}_{21,2} = \mathbf{T}^{(3)}_{2,2} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{S}^{(3)}_{32,1} = \boldsymbol{\Pi}_{\binom{3}{2},1} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$$

$$\mathbf{S}_{32,2}^{(3)} = \boldsymbol{\Pi}_{\binom{3}{2},2} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}$$

$$\mathbf{S}_{32,3}^{(3)} = \boldsymbol{\Pi}_{\binom{3}{2},3} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{S}_{33,1}^{(3)} = \mathbf{I}_{\binom{3}{3}} = \begin{bmatrix} 1 \end{bmatrix}$$

$$\mathbf{S}_{11,1}^{(4)} = \mathbf{T}_{1,1}^{(4)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{S}_{41,1}^{(4)} = \boldsymbol{\Pi}_{\binom{4}{1},1} = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{S}_{41,2}^{(4)} = \boldsymbol{\Pi}_{\binom{4}{1},2} = \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix}$$

$$\mathbf{S}_{41,3}^{(4)} = \boldsymbol{\Pi}_{\binom{4}{1},3} = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\mathbf{S}_{41,4}^{(4)} = \boldsymbol{\Pi}_{\binom{4}{1},4} = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{S}_{21,1}^{(4)} = \mathbf{T}_{2,1}^{(4)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\mathbf{S}_{21,2}^{(4)} = \mathbf{T}_{2,2}^{(4)} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{S}^{(4)}_{42,1} = \boldsymbol{\Pi}_{\binom{4}{2},1} = \begin{bmatrix}1 & 0 & 0 & 0 & 0 & 0\end{bmatrix}$$

$$\mathbf{S}^{(4)}_{42,2} = \boldsymbol{\Pi}_{\binom{4}{2},2} = \begin{bmatrix}0 & 1 & 0 & 0 & 0 & 0\end{bmatrix}$$

$$\mathbf{S}^{(4)}_{42,3} = \boldsymbol{\Pi}_{\binom{4}{2},3} = \begin{bmatrix}0 & 0 & 1 & 0 & 0 & 0\end{bmatrix}$$

$$\mathbf{S}^{(4)}_{42,4} = \boldsymbol{\Pi}_{\binom{4}{2},4} = \begin{bmatrix}0 & 0 & 0 & 1 & 0 & 0\end{bmatrix}$$

$$\mathbf{S}^{(4)}_{42,5} = \boldsymbol{\Pi}_{\binom{4}{2},5} = \begin{bmatrix}0 & 0 & 0 & 0 & 1 & 0\end{bmatrix}$$

$$\mathbf{S}^{(4)}_{42,6} = \boldsymbol{\Pi}_{\binom{4}{2},6} = \begin{bmatrix}0 & 0 & 0 & 0 & 0 & 1\end{bmatrix}$$

$$\mathbf{S}^{(4)}_{22,1} = \mathbf{I}_{\binom{4}{2}} = \begin{bmatrix}1 & 0 & 0 & 0 & 0 & 0\\ 0 & 1 & 0 & 0 & 0 & 0\\ 0 & 0 & 1 & 0 & 0 & 0\\ 0 & 0 & 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 0 & 0 & 1\end{bmatrix}$$

$$\mathbf{S}^{(4)}_{31,1} = \mathbf{T}^{(4)}_{3,1} = \begin{bmatrix}1 & 0 & 0 & 0\\ 1 & 0 & 0 & 0\\ 1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\end{bmatrix}$$

$$\mathbf{S}^{(4)}_{31,2} = \mathbf{T}^{(4)}_{3,2} = \begin{bmatrix}0 & 1 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 1 & 0\end{bmatrix}$$

$$\mathbf{S}^{(4)}_{31,3} = \mathbf{T}^{(4)}_{3,3} = \begin{bmatrix}0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 0 & 1\\ 0 & 0 & 0 & 1\end{bmatrix}$$

$$\mathbf{S}_{43,1}^{(4)} = \boldsymbol{\Pi}_{\binom{4}{3},1} = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{S}_{43,2}^{(4)} = \boldsymbol{\Pi}_{\binom{4}{3},2} = \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix}$$

$$\mathbf{S}_{43,3}^{(4)} = \boldsymbol{\Pi}_{\binom{4}{3},3} = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\mathbf{S}_{43,4}^{(4)} = \boldsymbol{\Pi}_{\binom{4}{3},4} = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{S}_{33,1}^{(4)} = \mathbf{I}_{\binom{4}{3}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{S}_{44,1}^{(4)} = \mathbf{I}_{\binom{4}{4}} = \begin{bmatrix} 1 \end{bmatrix}$$

$$\mathbf{S}_{32,1}^{(4)} = \begin{bmatrix} \mathbf{S}_{21,1}^{(3)} & \mathbf{0}_{\binom{3}{2}\times\binom{3}{2}} \\ \mathbf{0}_{\binom{3}{3}\times\binom{3}{1}} & \mathbf{S}_{32,1}^{(3)} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

$$\mathbf{S}_{32,2}^{(4)} = \begin{bmatrix} \mathbf{S}_{21,2}^{(3)} & \mathbf{0}_{\binom{3}{2}\times\binom{3}{2}} \\ \mathbf{0}_{\binom{3}{3}\times\binom{3}{1}} & \mathbf{S}_{32,2}^{(3)} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\mathbf{S}_{32,3}^{(4)} = \begin{bmatrix} \mathbf{0}_{\binom{3}{2}\times\binom{3}{2}} & \mathbf{S}_{22,1}^{(3)} \\ \mathbf{0}_{\binom{3}{3}\times\binom{3}{1}} & \mathbf{S}_{32,3}^{(3)} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

APPENDIX C

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

Figure C.1: Ordering scheme for the 2-D input array.

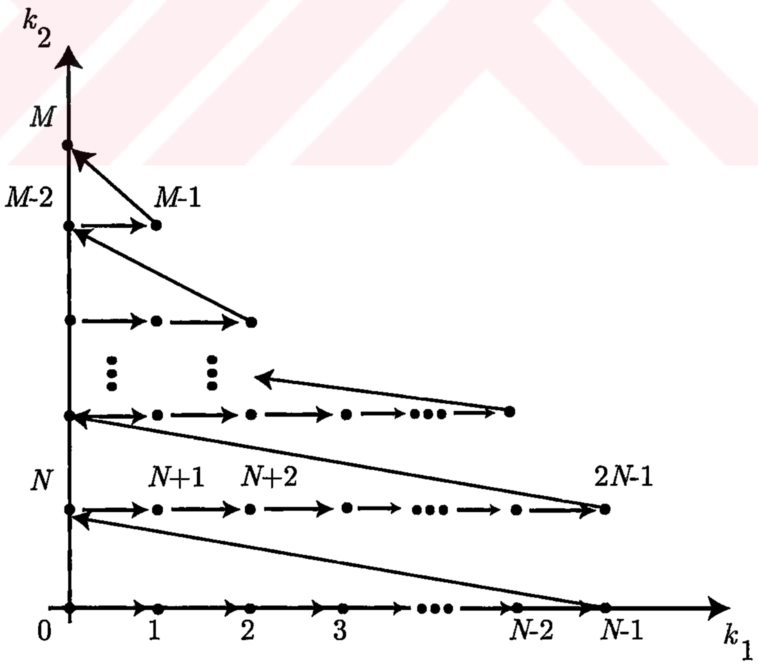


Figure C.2: Indexing scheme for the 2-D input array.

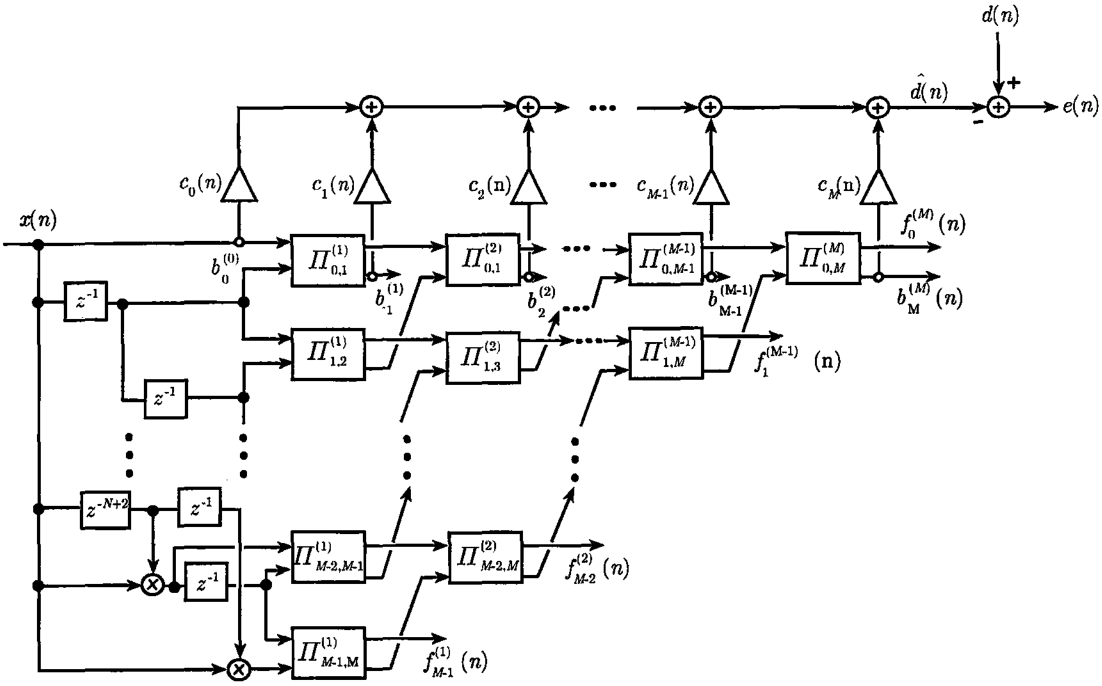


Figure C.3: Full complexity nonlinear joint-process estimator.

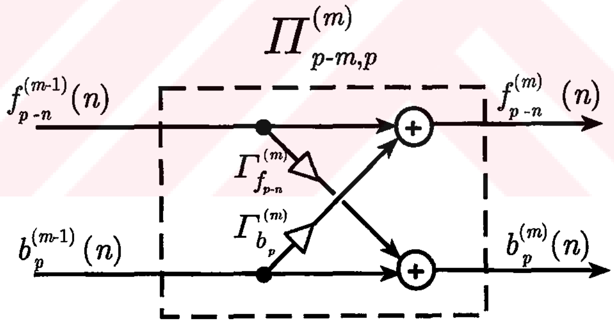


Figure C.4: Internal structure of the basic lattice module utilized in the nonlinear joint process estimator.

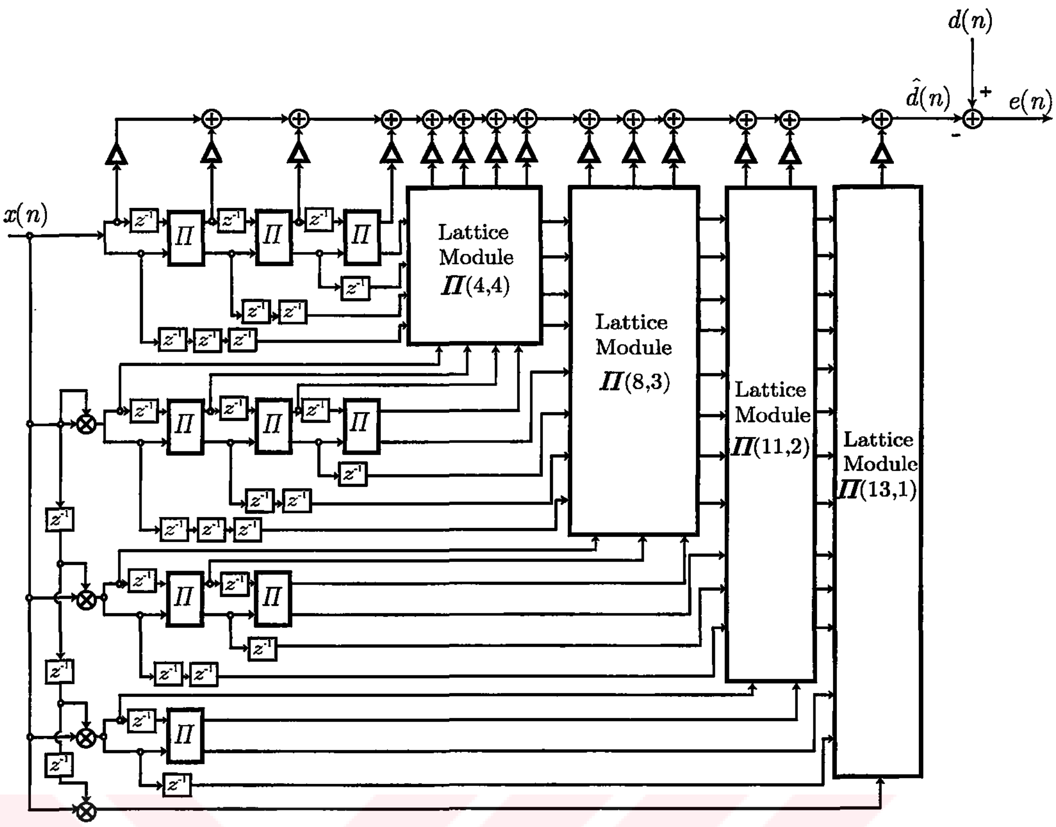


Figure C.5: Reduced complexity nonlinear joint-process estimator for $N = 4$.

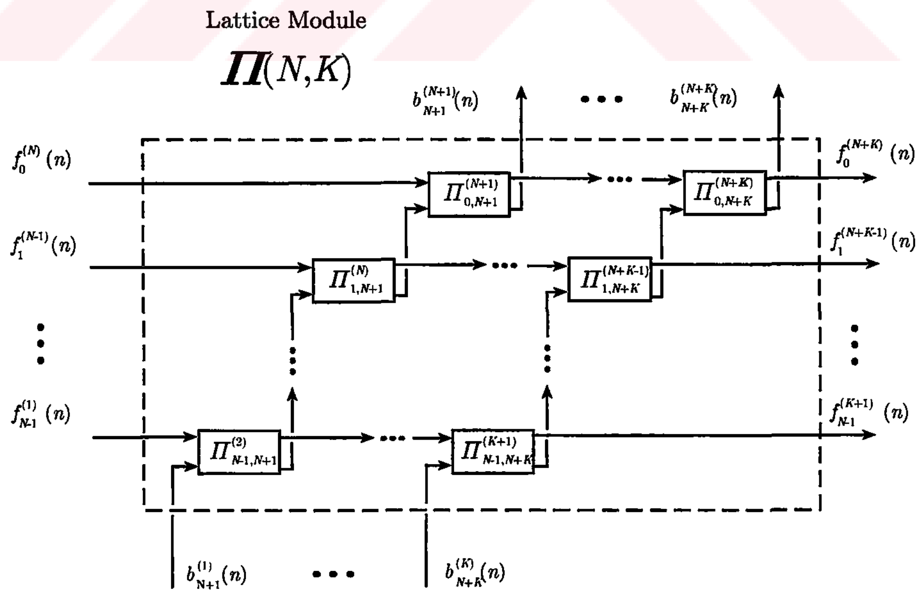


Figure C.6: Internal structure of the lattice block $\boldsymbol{\Pi}(N, K)$ utilized in the reduced complexity nonlinear joint process estimator.

Table C.1: Practical implementation of the RLS adaptation algorithm for the reflection coefficients inside the lattice modules.

forward prediction coefficients adaptation	backward prediction coefficients adaptation
$\bar{g} = P_f(n-1) f_{p-m}^{(m-1)}(n)$	$\bar{g} = P_b(n-1) b_p^{(m-1)}(n)$
$\alpha = \lambda + \bar{g} f_{p-m}^{(m-1)}(n)$	$\alpha = \lambda + \bar{g} b_p^{(m-1)}(n)$
$g = \frac{\bar{g}}{\alpha}$	$g = \frac{\bar{g}}{\alpha}$
$P_f(n) = \lambda^{-1}\big(P_f(n-1) - g\bar{g}\big)$	$P_b(n) = \lambda^{-1}\big(P_b(n-1) - g\bar{g}\big)$
$e = b_p^{(m-1)}(n) + \Gamma_{f_{p-m}}^{m}(n-1) f_{p-m}^{(m-1)}(n)$	$e = f_{p-m}^{(m-1)}(n) + \Gamma_{b_p}^{m}(n-1) b_p^{(m-1)}(n)$
$\Gamma_{f_{p-m}}^{m}(n) = \Gamma_{f_{p-m}}^{m}(n-1) + g\, e$	$\Gamma_{b_p}^{m}(n) = \Gamma_{b_p}^{m}(n-1) + g\, e$

Table C.2: Linear and quadratic coefficients of the unknown Volterra filter used in the simulations.

	$b_1(i_1)$		$b_2(i_1, i_2)$			
i_1		$i_2 =$	0	1	2	3
0	-0.052		1.020	-1.812	-1.138	-0.592
1	0.723			1.389	-2.608	-1.486
2	0.435				-0.635	-0.468
3	-0.196					-1.044

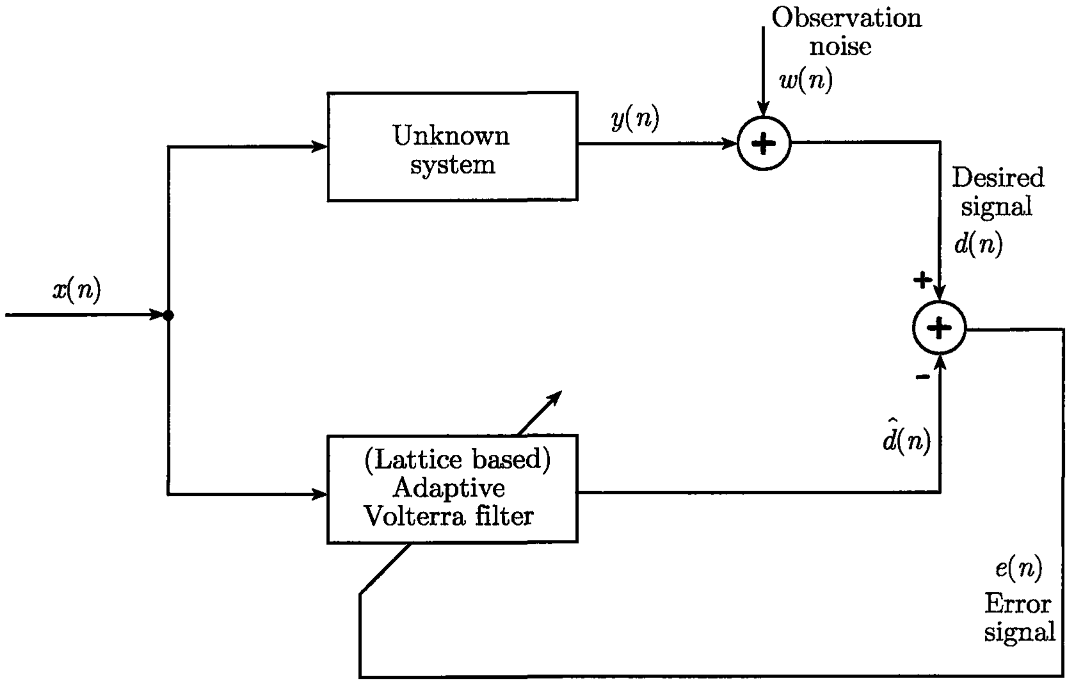


Figure C.7: The general setup for the adaptive second-order Volterra filter identification simulations.

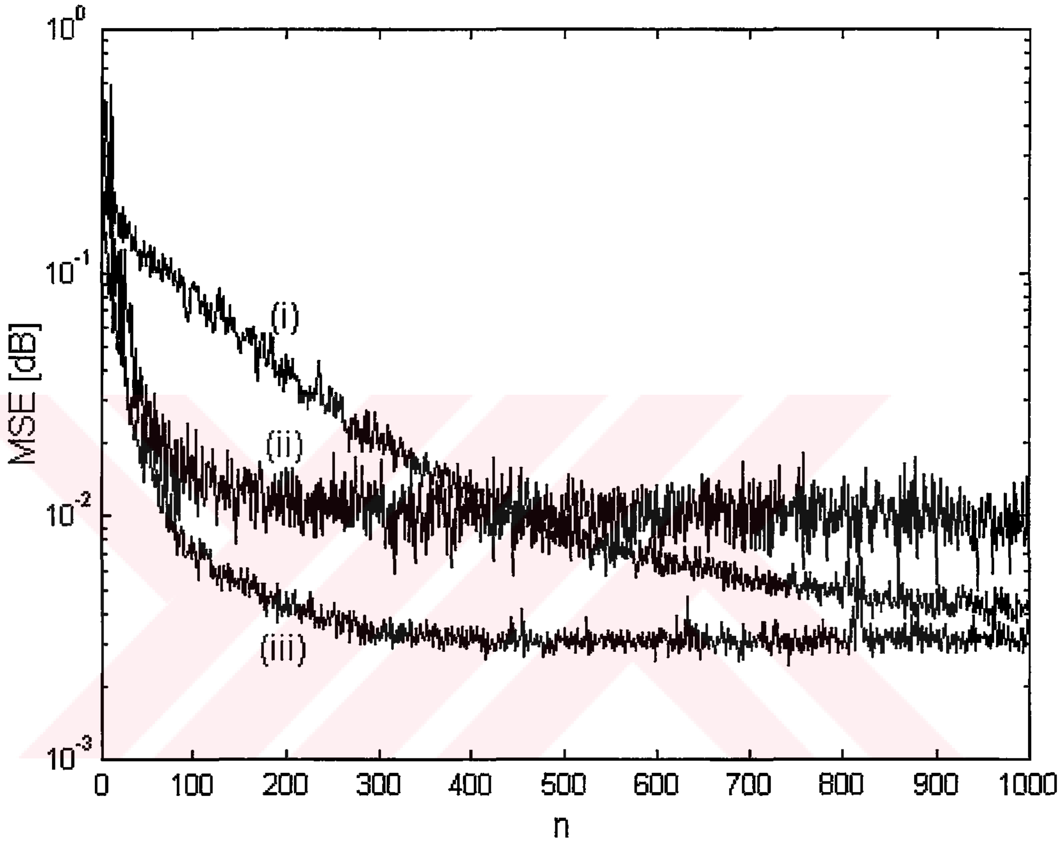


Figure C.8: Learning curves for different models; (i) multichannel lattice structure, (ii) transversal direct-form realization, (iii) model based on 2-D lattice structure.

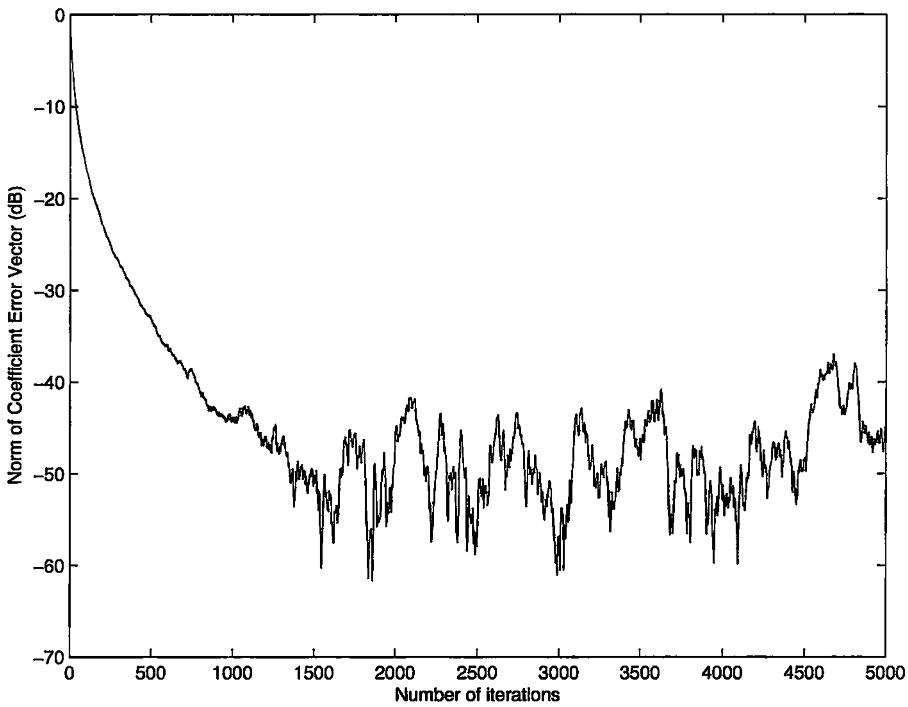


Figure C.9: Norm of coefficient error vector for linear coefficients $b_1(i_1)$ ($\|e_1(n)\|$).

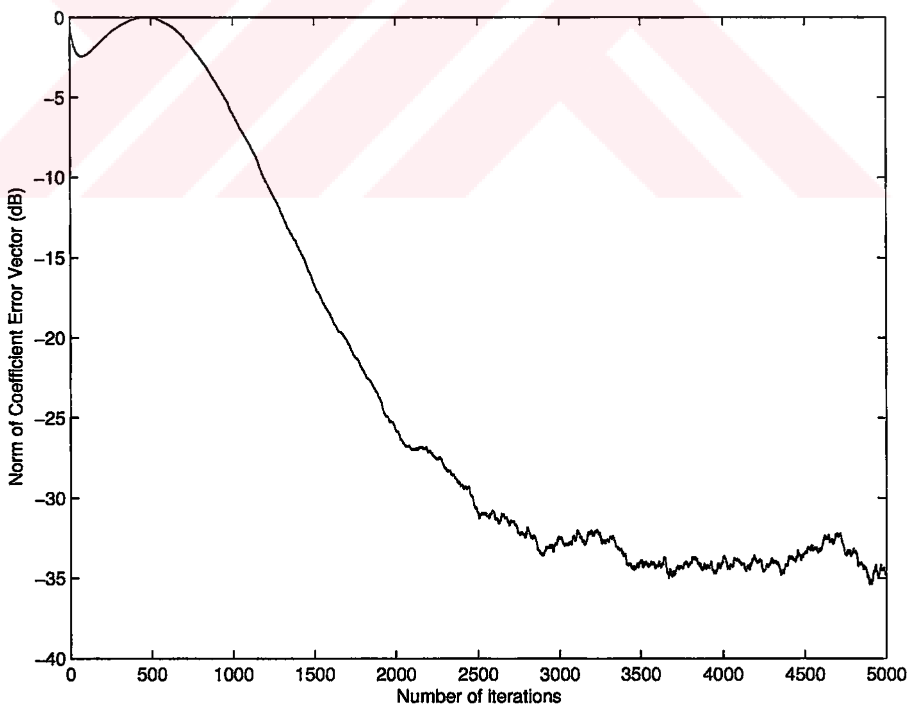


Figure C.10: Norm of coefficient error vector for quadratic coefficients $b_2(i_1, i_2)$ ($\|e_2(n)\|$).

REFERENCES

Agazzi, O., Messerschmitt, D. G. and Hodges, D. A., 1982. Nonlinear echo cancelation of data signals, *IEEE Trans. Communications*, **COM-30**(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 crosscorrelation, *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.

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

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

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

Özden, 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.

BIOGRAPHY

Ender M. Ekşioğlu was born in Istanbul, Turkey in 1976. He graduated from İstanbul Lisesi in 1993. He received the B.Sc. and M.Sc. degrees in electrical engineering from the University of Michigan, Ann Arbor, USA, in 1997 and 1999, respectively. Since September, 1999 he has been studying towards a Ph.D. degree in electrical engineering at the Istanbul Technical University, Istanbul, Turkey. He has been working as a teaching assistant at the same university.