CHAPTER 1

INTRODUCTION

> *"In medicine, one must pay attention not to plausible theorizing* (*λογισμος*) *but to experience and reason* (*λογος*) *together. I agree that theorizing is to be approved, provided that it is based on facts and is systematically induced from what is observed; but conclusions drawn by the unaided reason can hardly be serviceable, only those drawn from observed facts."*
>
> *Hippocrates, "Precepts"*
> *Athens, 5th Century B.C.*

1.1. PURPOSE OF THIS BOOK

The purpose of this book is to bring to the attention of the biomedical community the fact that an effective methodology exists for quantifying the dynamic interrelationships among physiological variables of interest from natural observations (data).

The book presents the conceptual framework and the mathematical foundation of this methodology that render it *general* in its application and *rigorous* in its approach. This methodology yields mathematical models of the dynamic interrelationships among the observed variables in a *nonlinear and nonstationary* context that is appropriate for physiological systems. Unlike many previous approaches, the advocated approach resists the temptation of simplifying the model to fit the method but retains instead the full complexity depicted in the data.

The book is focused on the modeling of nonlinear time-invariant physiological systems; unlike most modeling studies to date that have focused on the limited class of linear time-invariant systems, due to the relative simplicity of the methods of estimation and analysis associated with the latter. Nonlinearities are ubiquitous in physiology and often essential in subserving critical aspects of physiological function. Although few will argue with the importance and necessity of addressing the nonlinear and dynamic aspects of physiological systems, most will view this task as a daunting challenge owing to its considerable complexity. The purpose of this book is to alter this confining view, so prevalent in the scientific community, by removing the perceived methodological barriers and offering the practicable prospect of analyzing nonlinear physiological systems within the prevailing experimental and computational constraints.

We seek to achieve this goal by presenting recently developed methodologies in a clear and reasoned manner that is rigorous but not unduly burdened by mathematical formalism. The emphasis is on key operational issues that allow the practical application of the advocated approach by interested investigators in a manner that avoids critical pitfalls and enhances the understanding of the physiological function of the system under study. Illustrative examples are used extensively to elucidate important methodological points and to assist the reader in understanding *why* and *how* the method works. The examples also instruct the reader in the manner through which the utility of the obtained results can be realized by elucidating their physiological interpretation.

The mathematical models are derived *inductively* (from the data) using a general and rigorous mathematical framework in order to secure methodological rigor and fidelity to the data. Unlike *deductive* methods previously used, the advocated methodology does not require *a priori* model postulates and avoids the potential pitfall of (possibly inaccurate) preconceived notions that may unduly constrain the achievable model form.

This point is most critical when limited experimental or natural data are used for the estimation and validation of the model, thereby limiting the modeling task to a portion of the operating space of the system. The advocated approach espouses the fundamental requirement of a broad ensemble of input-output data that covers as densely as possible the entire operating space of the system. Therefore, in the advocated approach, the inductively derived models are *"true to the data"* that are collected under broad experimental or natural operating conditions, and not *ad hoc* mathematical postulates (reflecting specific preconceived notions) fitted to experimental data that may probe only part of the total operating space of the system. Essential for the inductive approach is a general methodological framework (such as the one advocated herein) that does not constrain the possible form of the emerging model.

It is evident that the advocated methodology departs from the conventional way of thinking and practice in the study of physiological systems with regard to the undesirability of *a priori* model postulates and specialized/experimental input waveforms. It challenges the established deductive approach, especially when limited experimental data are used, as potentially misleading and inherently incapable of exploring the full complexity of the system. It rejects the notion that practical necessity dictates the simplification of the employed model to stay within the "tractable boundaries" of linear, stationary or static analysis, as unjustifiable in light of recent

developments that make nonlinear, nonstationary and dynamic analysis feasible. It insists on the use of a broad repertoire of data (be they experimental or natural, although the latter are preferable) instead of limited experimental data using specialized waveforms (e.g., pulses, sinusoids) that are often contrived to "simplify" the experiment and facilitate its interpretation. The risks of such experimental or methodological "simplifications" cannot be overstated, since they are likely to create the illusion of knowledge while providing results of limited utility (at best) or potentially misleading. For instance, the response to a pulse or sinusoidal stimulus cannot be used to extrapolate or infer the response of nonlinear dynamic systems to any other stimulus.

The validity of the aforementioned arguments rests on whether physiological systems (that have been shown to be almost always nonlinear and dynamic) can be represented meaningfully by approximate linear and/or static models. The definitive answer to this question can be given *only after* performing the complete nonlinear dynamic analysis of the system under broad operating conditions and, subsequently, assessing the adequacy of linear/static analysis. In other words, the linear/static analysis is inherently incapable of answering this question because it does not probe the nonlinear/dynamic alternative (i.e., it does not contain explicit information about the possible nonlinear dynamic characteristics of the system). Note that the assessment of adequacy (or not) of the linear/static model depends on the specific data ensemble used and the prevailing ambient noise. Therefore, nonlinear dynamic analysis with a broad data repertoire is the only way to reach the correct assessment of the adequacy of the linear/static model and obtain globally valid models (even if the latter end up being static and/or linear).

The expressed views seem almost self-evident but represent a strong challenge to the *status quo*. They seek to facilitate the advent of a new era in systems physiology that constitutes a quantum leap in the way physiological systems are modeled and understood. They are not meant to denigrate the efforts of past investigators who did not follow the advocated approach, since a review of history establishes the fact that the process of scientific progress requires a multitude of viewpoints and reveals that even unsuccessful efforts contribute to the advancement of knowledge by identifying dead ends.

The advocated approach cannot be the definitive view on the subject of physiological system modeling, since it simply represents another stage in the evolution of knowledge (with many improvements and refinements certain to follow). Nonetheless, it can advance the state of the art

by a "quantum leap", since it represents a drastic departure from conventional thinking and practice. It is interesting to note that the advocated approach is consistent with the basic tenets of the Hippocratic teachings that formed the foundation of medicine as a scientific discipline, separated from priestcraft and groundless speculation, in the 5th century B.C. (see Historical Note #1).

Hippocrates first asserted the cardinal importance of clinical observation against "theorizing" (speculation not supported by data) and established the fundamental concept of the "unity of organism" against the fragmented approach of the "Empiricists" and the "Anatomists". The latter distinction remains the key issue underpinning the ongoing debate between the integrative systems approach advocated herein and the reductionist approach prevalent in the physical and biological sciences to date. The static view of the Anatomists and the Empiricists was countered by the Hippocratic dynamic view of the "disease process" and the "recuperative faculties" of living organisms (an early version of the concepts of homeostasis and system dynamics). The Hippocratic views survived the challenge of the Empiricists, as well as the "Methodists" and the "Atomicists" of antiquity (all espousing reductionist views), and were restored to their rightful place of universal acceptance by the great Greek physician, scientist and philosopher Galen of Pergamos (Galenos or Γαληνος in Greek) in the 2nd century A.D. Galenos' extensive writings on human physiology defined medicine until the 16th century when William Harvey founded modern physiology with his seminal experiments in Padova (building on earlier seminal anatomical contributions by Vesalius and others in northern Italy).

Galenos fully espoused the key Hippocratic views on the "unity of the organism" and the cardinal importance of clinical observation of the "disease process" (as opposed to the static view of examining the composing parts and the isolated symptoms, espoused by all other schools of thought at his time) that led him to an *integrative* and *dynamic* view of physiology. These basic tenets form the foundation of the advocated approach of *dynamic system physiology* under natural operating conditions.

This approach stands in contrast to the reductionist and static viewpoints that remain dominant in biological sciences, and questions the validity (or even the purpose) of the oversimplified experimental preparations often used in designing experiments for conventional "hypothesis-driven" research. These strong statements do not seek to invalidate the significant contributions made by hypothesis-driven research or to reject the valuable knowledge acquired

over the centuries through the reductionist approach that have advanced scientific progress. The advocated viewpoint simply seeks to establish the proper balance between the two approaches in pursuing scientific knowledge, so that their synergistic contributions serve scientific progress and prevent the establishment of a mutually-impeding antagonism so often fostered in the past by an unwise inclination for polarized thinking.

It is the ambitious goal of this book to contribute to a new movement (with ancient roots) that will restore the key Hippocratic-Galenic tenets to their rightful position in scientific thinking and practice through adoption of the dynamic systems viewpoint, in order to bring about the desirable leap of progress in physiology and medicine.

1.2 ADVOCATED APPROACH

The advocated approach offers the effective methodological framework for obtaining reliable and objective (i.e., devoid of subjective modeling notions) descriptors of the system nonlinear dynamics based on the available experimental or natural data. This approach employs general model forms that do not require specific model postulates and yield inductively "data-true" models in the stochastic broadband context of natural operating conditions.

Due to the complexity of this fundamental problem, we have taken a gradualist step-by-step approach, building on the rigorous and general mathematical foundation of the Volterra-Wiener approach as extended and modified for various applications over the last thirty years. It is gratifying to note that our efforts have succeeded in developing a solid foundation for a general modeling approach capable of tackling this problem in a practical context.

This novel modeling methodology has been tested in pilot applications from the nervous, cardiovascular, renal, respiratory and metabolic/endocrine systems. These applications have showcased the efficacy of the developed methodology and have allowed important advances in systems physiology by assigning physiological significance to the obtained model components in a manner that deepens the scientific understanding of the system under study. This demonstrates the potential benefits of the advocated approach that are expected to enable improvements in diagnostic as well as therapeutic procedures.

Standing on this solid foundation, we are poised to address the next generation of challenges that pertain to the *multi-variate* and *highly interconnected* nature of physiological systems. This

direction represents the natural extension of our efforts in reaching our ultimate objective of modeling the *true* physiological systems, and not their simplified "surrogates" born of methodological inadequacy. This forward looking task is commenced with the important case of multiple inputs and multiple outputs that is discussed in Chapter 8. The complexity that emerges from the interconnections among multiple physiological variables of interest (often in *closed-loop* or *nested-loop* configurations) must be placed in a nonlinear dynamic context, with possible nonstationarities. Since we seek to study the physiological system under "natural" operating conditions (i.e., exposed to an ensemble of natural stimuli and unconstrained by arbitrary experimental manipulations), the ultimate modeling task must be placed in a multi-variate stochastic broadband context, without artificially imposed constraints (e.g., fixed operating points or specialized input waveforms) in order to achieve a globally valid model of the *real* system.

A measure of the posed challenge is attained when we note that proper study of the *real* physiological systems requires reliable modeling methods that are capable of dealing with:

- nonlinear dynamics (of arbitrary order)
- multiple variables of interest (observable/measurable)
- multiple interconnections (possibly in closed-loop)
- possible nonstationarities in system dynamics
- broadband stochastic input/output signals
- considerable measurement noise and systemic interference

Last, but not least, the obtained models must be amenable to meaningful physiological interpretation and offer the prospect for achieving significant diagnostic and/or therapeutic improvements.

The critical task of meaningful interpretation of the obtained models and their proper utilization in clinical practice is formidable, as much as it is important, because the wealth of information contained within these complicated models may be overwhelming and difficult to harness. Nonetheless, it is incumbent on us to perform this task in order to realize the benefits of the advocated new approach and achieve the ambitious goal of a quantum leap in the state of the art.

Interpretation of the obtained models will focus on relating their characteristics to specific physiological mechanisms that are either known qualitatively or can be explored experimentally. It will also examine the effects of changes in certain physiological variables (in a dynamic context) and the robustness of the overall system in the event of internal or external perturbations (homeostasis). The latter study may demarcate the bounds of normal vs. pathological states.

Utilization of the obtained models in a clinical context will seek to examine their use for improved diagnosis of disease (i.e., more and better clinically relevant information) and for the quantitative assessment of pharmaceutical treatment or therapeutic intervention. The latter will allow optimization of treatment (with regard to specific clinical goals) and the design of improved therapeutic procedures, consistent with the key Hippocratic exhortation: "*first, do no harm…*".

The progress made to date in the development of effective methodologies for nonlinear and/or nonstationary modeling of physiological systems is summarized in this book and has given rise to a new generation of issues associated with the study of greater system complexity and the analysis of expanded experimental/clinical databases - - both direct consequences of advances in the state of the art - - consistent with Socrates' aphorism: "*the more I learn, the more I realize how much I do not know*".

The advocated approach stands on the confluence of methodological (nonlinear and nonstationary) and technological (computational and experimental) advancements, and seeks to leverage on their synergistic utilization in order to tackle the formidable challenges in physiological system modeling from natural (i.e., random broadband) data. Pilot applications are selected from physiological domains that exhibit essential nonlinearities /nonstationarities (neural, cardiovascular, respiratory, renal, endocrine and metabolic systems) in order to demonstrate the wide applicability and unique capabilities of the advocated novel methodologies. In this sense, the advocated approach is at the cutting edge of scientific developments and has a universal appeal to all physiological domains in terms of scientific advancement, as well as potential impact across a broad swath of clinical applications.

The immense variety of nonlinear behavior makes it desirable that the developed methodologies retain a high degree of generality and the ability to gracefully transition into interpretable models for each particular application.

The complexity of nonlinear behavior also makes it imperative that the ensemble of input signals used for probing/observing the system behavior be spectrally and temporally rich, so that the maximum possible interactions among different values or frequencies of the input signal be observed at the recorded output. This is the fundamental reason why we favor *random broadband input signals* over specialized deterministic signals, following on the pivotal suggestion of Norbert Wiener regarding the use of Gaussian white noise inputs that is discussed in Section 2.2. For instance, the customary use of rectangular pulses or sinusoids as stimuli by experimental physiologists may be appealing to the eye of the observer and offer a comprehensible response waveform, but they are very poor probing signals in terms of the extracted amount of information for given experimentation time duration. This important point will be elaborated in 2.1.5, 2.2.1 and 5.1.

In addition, the practical modeling task is complicated by the fact that the observed operating conditions are often burdened by severe interference and noise, as well as nonstationarities in the system behavior. Since we will mainly address the case of stationary (time-invariant) models, the issue of systemic and measurement nonstationarities should be borne in mind as a complicating factor that may compromise the quality of the results or limit the record length of the data used for model estimation. The subject of explicit nonstationary modeling will be discussed in Chapter 9.

The practical challenge posed by the ubiquitous presence of noise and/or interference necessitates robust estimation algorithms that minimize the effect of noise/interference on the obtained model estimates for a given data record length. Increasing the data record length will normally improve the model estimates unless possible nonstationarities degrade the benefits of the extended data record. Repetition of identical experiments and proper averaging may mitigate the effects of noise/interference even for certain types of systemic nonstationarity but is more time-consuming and vulnerable to lack of ergodicity (see Chapter 5). In all cases, intelligent design of experiments can maximize the output signal-to-noise ratio by putting more input signal power at those frequency bands that are most critical for the system dynamics and are less contaminated by noise/interference. This is an important practical issue that has not received adequate attention in the literature and will be discussed in Chapter 5.

1.3 THE PROBLEM OF SYSTEM MODELING IN PHYSIOLOGY

The purpose of physiological system modeling is to advance our quantitative understanding of biological function and improve medical science and practice by utilizing the acquired quantitative knowledge. In modeling physiological systems, we seek to summarize all available experimental evidence about the functional characteristics of a system in the form of mathematical relations among variables of physiological interest. The resulting mathematical models ought to emulate the observed functional behavior of the system under natural operating conditions, when simulated on the computer.

Ideally, such a model must be accurate (i.e., reproduce precisely the observed data), global (i.e., be accurate under all natural operating conditions), compact (i.e., have the minimum mathematical and computational complexity) and interpretable (i.e., be amenable to physiological interpretation that advances our understanding of the mechanisms subserving the system function). Implicit in the first attribute is the robustness of the model which provides for stable behavior in the face of internal or external random perturbations. The latter, and the omnipresence of noise, require that our modeling efforts be cast in a stochastic context. Furthermore, the development of a global model (valid under all natural operating conditions) presumes our ability to observe and measure the spontaneous activity of the variables of interest with sufficient accuracy and sampling resolution over representative time intervals. These variables can be viewed as inputs, outputs or internal state variables depending on our specific modeling goals.

Such an "ideal" model would offer a succinct quantitative representation of the functional characteristics of the system and would allow the study of its behavior under arbitrary conditions through computer simulations (thus maximizing the "yield" of physiological research). In addition to being a complete "capsule of knowledge" that advances scientific understanding of how and why physiological systems function in the way they do, such a model can improve clinical diagnosis (by providing more and better relevant information) and treatment (by properly guiding therapeutic procedures and assessing their effects). The implications are immense and promise to usher a new era of advanced medical care.

The development of such an "ideal" model is a formidable task, because of the functional complexity of physiological systems which are typically dynamic, nonlinear, nonstationary, highly interconnected (often in closed-loop configurations) and subject to stochastic

perturbations and noise. Furthermore, the modeling process has been constrained in practice by limitations of the available experimental, computational and analytical methods, leading heretofore to the development of "less than ideal" models in the sense defined above, i.e., inability to reach the desired attributes of accuracy, globality, compactness and interpretability.

This sobering reality is put in perspective when one notes that many of the current modeling efforts are still confined to rudimentary methods of static and linear analysis. The course of methodological evolution started with static linear analysis (linear regression among variables of interest) and gradually progressed into dynamic (differential or integral equations) and nonlinear analysis. Although linear dynamic analysis has been increasingly used, the use of nonlinear dynamic analysis remains rather limited due to the scarcity of practical methods and the intrinsic complexity of the problem. This shortcoming would not be alarming, if it were not for the fact that most physiological systems exhibit significant and essential dynamic nonlinearities. This problem is compounded by the fact that physiological systems are also often nonstationary (i.e., their functional characteristics vary with time) necessitating models whose parameters also change with time. The need for proper modeling methodologies in this realistic context is increasingly pressing and provides the motivation for this book.

Physiological system modeling provides the means of summarizing vast amounts of experimental data into relatively compact mathematical (or computational) forms that allow the formulation and testing of scientific hypotheses regarding the functional properties of the system -- an iterative process that should lead to successive refinement and evolution of the system model. Thus, system modeling attains a central role in the scientific process of generation and dissemination of knowledge -- consistent with the credo "model or muddle." Models can be applied to arbitrary levels of system decomposition or integration depending on the availability of appropriate data -- hence providing the conceptual and methodological means for developing a hierarchical understanding of integrative systems physiology.

The fundamental question of the functional relationships between observed physiological variables drives the system modeling effort. The variables are observed experimentally over time (occasionally over space or wavelength) and are viewed as signals (or datasets) that are linked by a causal relationship. The direction of this causal relationship (cause-to effect sequence) defines some of them as inputs and some of them as outputs of a conceptual operator (the system) that transforms the inputs into the outputs (Figure 1.1). This transformation is

generally dynamic in the sense that the present values of the outputs depend not only on the present but also on the past values of the inputs. Another way of describing the same (defining) characteristic of dynamic systems is to state that the effect of an input change upon the output signal spreads over time.

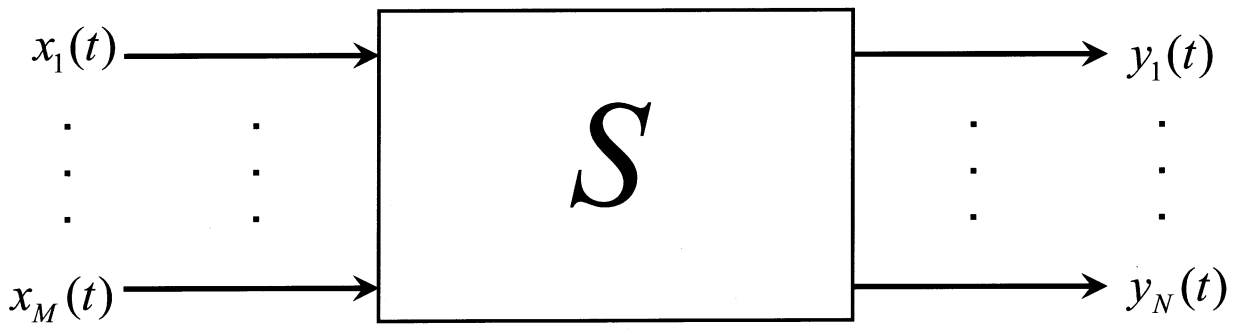
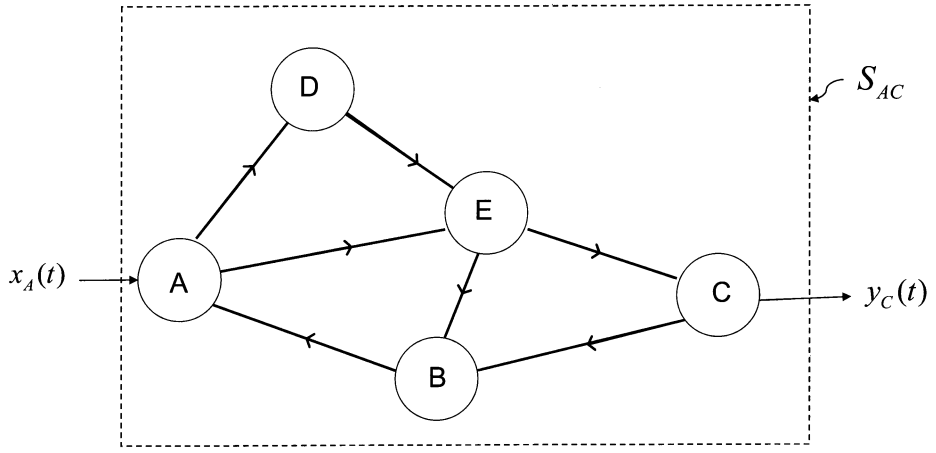


Figure 1.1
Schematic of a "black-box" system operator S transforming M input signals $\{x_1(t),\ldots,x_M(t)\}$ into N output signals $\{y_1(t),\ldots,y_N(t)\}$.

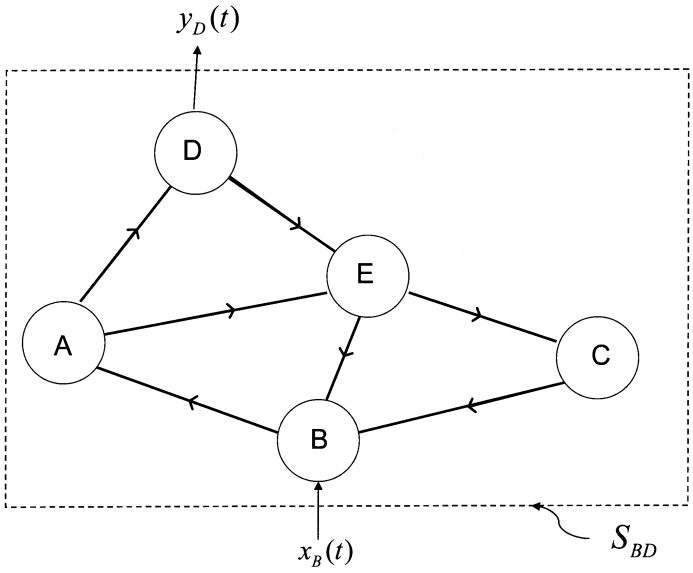
It is worth noting that the system is defined by means of its inputs and outputs (and not by a physical entity). Thus, we may define many different systems using the same physical entity by altering the selected inputs and outputs, as long as causality is not violated. This point is illustrated in Figure 1.2 where a physical entity is comprised of five components (A, B, C, D, E) and the designated connections (arrows) represent directional causal relationships. If we stimulate component A with input $x_A(t)$ and record output $y_C(t)$ from component C, then we define the system S_{AC} as the signal transformation from A to C. However, if we record from another component (e.g., $y_D(t)$ out of component D) and stimulate another component (e.g., $x_B(t)$ into component B), then we define a different system S_{BD} representing the signal transformation from input B to output D, even though the underlying physical entity remains the same.

The input-output signal transformation describes the functional properties of the system and may be linear or nonlinear, time-invariant (stationary) or time-varying (nonstationary), and deterministic or stochastic. A mathematical expression describing quantitatively this input-output signal transformation is the sought model of the system. For our purposes, the sought model will be deterministic, implying that possible stochastic variations of the system

characteristics will be relegated to the status of systemic noise or modeling errors and will be incorporated in the stochastic error term of the model. The latter will also incorporate other modeling errors and possible measurement noise or external/systemic interference.



(a)



(b)

Figure 1.2
Schematic diagram of causal connections (denoted by arrows) among five physical variables (A,B,C,D,E). The selected input(s) and output(s) define the particular system operator. For instance, an input $x_A(t)$ stimulating A and an output $y_C(t)$ recorded from C define a system operator S_{AC} (left panel) or an input $x_B(t)$ stimulating B and an output $y_D(t)$ recorded from D define a different system operator S_{BD} (right panel).

If we limit ourselves, at first, to the single-input/single-output case, we can use the convenient mathematical notation of a functional $S[\cdot]$ to denote the causal relationship between past and present values of the input *x* and the present value of the output *y* as:

$$y(t) = S[x(t'), t' \le t] + \varepsilon(t) \tag{1.1}$$

where the functional $S[\cdot]$ represents the deterministic system as it maps the input past and present values onto the output present value, and ε(t) represents the stochastic error term. The error term is assumed stochastic and additive – the latter is a common assumption that simplifies the estimation task but may not correspond to reality in some cases where the error may have a multiplicative or other modulatory effect on the input/output signals. The error term is also called the "residual" and may contain modeling errors (including possible stochastic variations of the system characteristics), systemic noise or interference and measurement noise. For analysis and estimation purposes, the residual is usually treated as a stationary random process (with zero mean) that is statistically independent from the input and the noise-free output signals (although it is contained in the output data measurements). Note that deviation from this assumption regarding the residual term may cause significant estimation errors.

The convenient mathematical notation of Equation (1.1) can be extended to the case of causal systems with multiple inputs and multiple outputs by adopting a vector notation (shown in bold) for the input/output signals, the residual terms and the functionals:

$$\boldsymbol{y}(t) = \boldsymbol{S}[\boldsymbol{x}(t'), t' \le t] + \boldsymbol{\varepsilon}(t) \tag{1.2}$$

Clearly each output signal must have its own distinct functional implicating (potentially) all inputs. The case of multi-input/multi-output systems is discussed extensively in Chapter 8. However, the main methodological developments are presented in Chapters 2-5 for the single-input/single-output case to avoid the burden of the inevitably increased complexity of mathematical expressions resulting from the multiple inputs and outputs.

1.3.1 Model Specification and Estimation

The goal of mathematical modeling is to obtain an explicit mathematical expression for the functional $S[\cdot]$ using experimental or natural input-output data and all other available knowledge about the system. This goal is generally achieved in two steps. At first, a suitable mathematical form is selected for $S[\cdot]$, containing unknown parameters and/or functions, which are subsequently estimated by use of input-output data in the second step. This two-step procedure is often referred to as "System Identification" in the engineering literature, and the two steps are termed "Model Specification" and "Model Estimation" respectively.

The *Model Specification* task is generally more challenging and the critical step to successful modeling. It utilizes all prior knowledge and information regarding the system under study and seeks to select the appropriate model form in each case. Typically the selection is made from among four classes of models: nonparametric, parametric, modular, and connectionist (see Table 1.1). The criteria for selection of the appropriate model class are discussed in Chapters 2-5 and constitute a critical issue.

Table 1.1 Nonlinear System Modeling Methodologies

	Nonparametric	Parametric	Modular	Connectionist
Model Specification	+	–	±	±
Interpretability	±	+	+	–
Robustness to Noise	+	-	+	±
Compactness	–	+	+	–
Adaptable to Time-Varying	+	±	±	+

The *Model Estimation* task employs estimation methods suitable for the specific characteristics of each case and seeks to maximize the accuracy of the resulting model prediction for given data types and noise conditions. Various measures and norms of model prediction accuracy can be used, but the most common is the mean-square error of the output prediction (the sum of the squared residuals).

The Model Estimation task is important because it yields the desired result, but it is meaningful only when the Model Specification task is performed successfully. The highly technical nature of the Model Estimation task has attracted most of the attention in the engineering literature, and the contents of this book reflect this fact by delving into the many technical details of the various estimation methods. Nonetheless, it is useful to remember that,

although the estimation methods are necessary tools for accomplishing the modeling task, the art of modeling and the impact of its scientific applications hinge primarily on the successful performance of the Model Specification task and the meaningful interpretation of the obtained model. It is for this reason that the overall modeling philosophy advocated by this book places the emphasis on securing first the best possible Model Specification results and then perform accurate Model Estimation.

Since system modeling seeks to find and quantify the causal functional relationships among observed variables, it occupies a central position in the process of scientific discovery. Although it addresses only the functional aspects of the system under study, structural information can be used to assist the Model Specification task by properly constraining the postulated model. Conversely, system models can be used to examine alternative hypotheses regarding the structural composition of a given system (e.g., interconnections of neuronal circuitry) and thus can advance our structural knowledge of physiological systems.

It is critical to note that the selection of the model form must not constrain unduly the range of functional possibilities of a system, lest it will lead to biased and inaccurate results. Thus extreme care must be taken to avoid such inappropriate constraining of the model either in terms of its mathematical form or in terms of its operational range (i.e., dynamic range and bandwidth). At the same time, the efficiency of the employed estimation methods and the utility of the obtained model in terms of scientific interpretability are compromised when the model is not adequately constrained. This key trade-off between model parsimony and its global validity pervades all modeling studies and is of fundamental importance, as discussed later in connection with the various classes of model forms.

Related to the Model Specification task is the issue of inductive versus deductive model development, discussed in Section 1.5. Deductive modeling is possible only in those rare occasions where sufficient knowledge exists about the detailed functional properties of the system, so that its internal workings can be described accurately by use of first physical and/or chemical principles. In these fortunate, but rare, occasions the system model can be reliably postulated in the form of precise mathematical expressions using a deductive process. The complexity of physiological systems rarely affords this kind of opportunity and, consequently, modeling of physiological systems is usually inductive (i.e., it is based on accumulated empirical evidence from experimental data). This distinction between inductive and deductive modeling

has been made before with the use of the terms "empirical or external" and "axiomatic or internal" models respectively [Arbib et al. 1969; Astrom & Eykhoff, 1971; Bellman & Astrom, 1969; Rosenblueth & Wiener, 1945; Yates, 1973; Zadeh, 1956].

Although prior knowledge about the system can be used to assist the Model Specification task, the modeling approaches presented herein are based entirely on experimental or natural input-output data and exclude the favorable -- but rare -- occasions where the model form can be derived from first principles or can be reliably postulated on the basis of prior knowledge about the system. Therefore, our approach to physiological system modeling is inductive and data driven.

The necessity of inductive modeling for most physiological systems elevates the importance of the type and quality of experimental or natural data needed for our modeling purposes. It is imperative, for instance, that the data cover the entire functional space of the system (e.g., the entire bandwidth and dynamic range of interest) and that the noise levels remain low relative to the power of the signal of interest. It must be noted in this connection that the systemic noise or interference (which is prevalent in physiological systems) is usually more harmful for the estimation process than the measurement noise.

1.3.2 Nonlinearity and Nonstationarity

A critical issue that complicates the modeling task is the presence of nonlinearities in the physiological system. Since this is the focus of the book, it is discussed in detail in the following section and in Chapters 2-7. Here, we limit ourselves to the definition of nonlinearities and nonstationarities with regard to the functional notation of Equation (1.1). A note should be made about static nonlinearities, whereby $y(t)$ depends only on the value of $x(t)$ and the functional $S[\cdot]$ reduces to a function. Static nonlinearities are easy to model (graphically or with simple numerical fitting procedures), leaving dynamic nonlinearities as the true challenge.

With reference to the functional notation of Equation (1.1), linearity of the system implies that $S[\cdot]$ obeys the "*superposition principle*", which states that if $y_1(t)$ and $y_2(t)$ are the system outputs for inputs $x_1(t)$ and $x_2(t)$ respectively, then for an input $x(t) = \lambda_1 x_1(t) + \lambda_2 x_2(t)$ the output is: $y(t) = \lambda_1 y_1(t) + \lambda_2 y_2(t)$. This can be expressed by the mathematical condition:

$$S\ \lambda_1 x_1(t) + \lambda_2 x_2(t)\ = \lambda_1 S\ x_1(t)\ + \lambda_2 S\ x_2(t) \tag{1.3}$$

where $x_1(t)$ and $x_2(t)$ are linearly independent signals, and λ_1 and λ_2 are nonzero scalar coefficients. The above condition can be tested experimentally with any pair of linearly independent inputs and scalar coefficients, and it must be satisfied by all such pairs. This condition for linear superposition can be extended to any number of linearly independent signals and remains by practical necessity a *necessary*, but not sufficient, condition (since all possible combinations cannot be practically tested). Appendix I provides the definition of linear independence between two or more signals. Elaboration on the experimental testing of system linearity is given in Section 5.2.4.

Returning to the functional notation of Equation (1.1), we can point out that stationarity (or time-invariance) of the system implies that the input-output mapping rule represented by $S[\cdot]$ remains invariant through time. It should be stressed that $S[\cdot]$ denotes the rule by which the system constructs the output at time t using the input values at time t and before. Thus, nonstationarity (or time-variance) should not be confused with the inevitable temporal changes in the output signal caused by temporal changes in the input signal, that occur whether the system is stationary or not. Experimental testing of the stationarity of a system requires the repetition of identical experiments at different times and the comparison of the obtained results. The test for stationarity of a system can be based on the following conditional statement: if the system output for input $x(t)$ is $y(t)$, then the output for input $x(t-\sigma)$ is $y(t-\sigma)$ for every time-shift σ. Since all experimental studies are subject to random influences and disturbances, this assessment is not straightforward in practice and requires statistical analysis of sufficient data, as discussed in Section 5.2.3. and in Chapter 9.

An experimental complication, often encountered in physiological systems, is the presence of possible nonstationarities in the experimental preparation caused by inadvertent injuries of surgical procedures. Thus an important distinction must be made between nonstationarties that are intrinsic to the system operation (e.g., endocrine/hormonal cycles or biological rhythms) and pertain to the actual physiological function of the system, and those that affect our measurement but are of no interest vis-à-vis the physiological function of the system under study. The former type of nonstationarity ought to be incorporated into the estimated model by proper methodological means, as discussed in Chapter 9. The latter type of nonstationarity degrades the

quality of the data and, although it can be often viewed as low-frequency noise, it may not have a simple additive effect on the observed output data (e.g., it may have a multiplicative or modulatory effect). Therefore, it is important to remember that the employed modeling methodologies for physiological systems must be robust in the presence of noise or systemic interference and must not require very long experimentation time over which measurement nonstationarities may develop or have significant effect.

Since the input-output data are collected and processed in sampled digital form, it is evident that the actual implementation of these modeling approaches takes place in discrete time. Thus, the continuous-time input-output signals $x(t)$ and $y(t)$ must be converted into discrete-time signals $x(n)$ and $y(n)$ using a fixed sampling interval T, where n denotes the discrete-time index (i.e., $t = nT$). Provided that proper attention is paid to the issue of aliasing (by sampling at a sufficiently high rate that secures a Nyquist frequency greater than the bandwidth of the sampled input-output signals, as discussed in Section 5.1.3), the presented mathematical methods generally transfer to the discrete-time case with minor modifications (e.g., converting an integral into summation). Both discrete and continuous cases are presented throughout the book, and we make special note in the few cases where the transition from continuous to discrete time requires special attention (e.g., from nonlinear differential to difference equations).

1.3.3 Definition of the Modeling Problem

With all these considerations in mind, the physiological system modeling problem is defined as:

Given a set of input-output experimental or natural data, find a mathematical model that describes the dynamic input-output relationship with sufficient accuracy (using a mean-square-error criterion for the output model prediction) under the following conditions:

- *no prior knowledge is available about the internal workings of the system*
- *nonlinearities and/or nonstationarities may be present in the system*
- *extraneous noise and/or systemic interference may be present in the data*
- *the obtained model must be amenable to physiological interpretation*
- *experimentation time and computational requirements may not be excessive*

A practical guide for the solution of the modeling problem in the context of physiological systems is provided in Chapter 5.

1.4. TYPES OF NONLINEAR MODELS OF PHYSIOLOGICAL SYSTEMS

The challenge of modeling nonlinear physiological systems derives from the immense variety of nonlinearities in living systems and the complex interactions among multiple mechanisms linked together within these systems.

We summarize in Table 1.1 the four main nonlinear modeling approaches and classes of nonlinear models used to date: *nonparametric, parametric, modular, and connectionist*.

In the *nonparametric* approach, the input-output relation is represented either analytically in integral equation form where the unknown quantities are kernel functions (e.g., Volterra-Wiener expansions) or computationally as input-output mapping combinations (e.g., look-up tables or operational surfaces /subspaces in phase-space). Nonparametric models are easy to postulate (because of their generality) but typically lack parsimony of representation. Of the various nonparametric approaches, the discrete-time Volterra-Wiener (kernel) formulation has been used most extensively for nonlinear modeling of physiological systems and will form the mathematical foundation of this book, along with its relations to the other modeling approaches.

In the *parametric* approach, algebraic or differential/difference equation models are typically used to represent the input-output relation for static or dynamic systems, respectively. These models contain typically a small number of unknown parameters that may be constant or time-varying depending on whether the model is stationary or nonstationary. The specific form of these parametric models is usually postulated *a priori* but the selection of certain structural parameters (e.g., degree/order of equation) are guided by the data.

The *modular* approach is a hybrid between the parametric and the nonparametric approach that makes use of *block-structured* models composed of parametric and/or nonparametric components properly connected to represent the input-output relation in a manner that reflects our evolving understanding of the functional organization of the system. The model specification task for this class of models is more demanding and may utilize previous parametric and/or nonparametric modeling results. A promising variant of this approach, which derives from the general Volterra-Wiener formulation, employs *principal dynamic modes* as a minimum set of filters to represent parsimoniously a nonlinear dynamic system of the broad Volterra class (see Section 4.1.1).

The *connectionist* approach has recently acquired considerable popularity and makes use of

generic model configurations/architectures, known as *artificial neural networks*, to represent input-output nonlinear mappings in discrete time. These connectionist models are fully parameterized, making this approach akin to parametric modeling, although typically lacking the parsimony and interpretability of parametric models. A hybrid nonparametric/connectionist approach is at the core of the modeling methodology that is advocated as the best overall option at present.

The relations among these four approaches are of critical practical importance, since considerable benefits may accrue from the combined use of these approaches in a cooperative manner. This synergistic use aims at securing the full gamut of advantages specific to each approach. The relative advantages and disadvantages in practical applications of the four modeling approaches will be discussed in Chapters 2-4.

The ultimate selection of a particular methodology (or combinations thereof) hinges upon the specific characteristics of the application at hand and the prioritization of objectives by the individual investigator. Nonetheless, it is appropriate to state that no single methodology is globally superior to all others (i.e., excelling with regard to all criteria and under all possible circumstances) and much can be gained by the synergistic use of the various modeling approaches in a combination that takes into account the specific characteristics/requirements of each application. Judgment must be exercised in each individual case to select the combination of methods that yields the greatest insight within the given experimental constraints. Since the general nonlinear system identification problem remains a challenge of considerable complexity, one must not be lulled into the risk-prone complacency of blind algorithmic processing. Many challenging issues remain that require vigilant attention, since there is no substitute for intelligence and educated judgment in resolving these issues. Four examples are given below to illustrate the various model forms employed by these approaches in different physiological domains.

Example 1.1

Vertebrate retina

The early stage of the vertebrate visual system (retina) is chosen as the first illustrative example to honor the historical fact that this was the first physiological system extensively studied with the Volterra-Wiener approach. A schematic of the neuronal architecture of the

retina is shown in Figure 1.3. The natural input to the retina is light intensity (photon flux per unit surface) impinging on the photoreceptor cells that convert the photon energy into intracellular potential through a chain of photochemical and biochemical reactions. The generated photoreceptor potential is synaptically transferred to downstream horizontal cells and bipolar cells through triadic synapses in the photoreceptor pedicle. The intracellular potentials thus generated within horizontal and bipolar cells are synaptically transferred to the postsynaptic ganglion cells and to the interneurons of the inner plexiform layer (various types of amacrine cells).

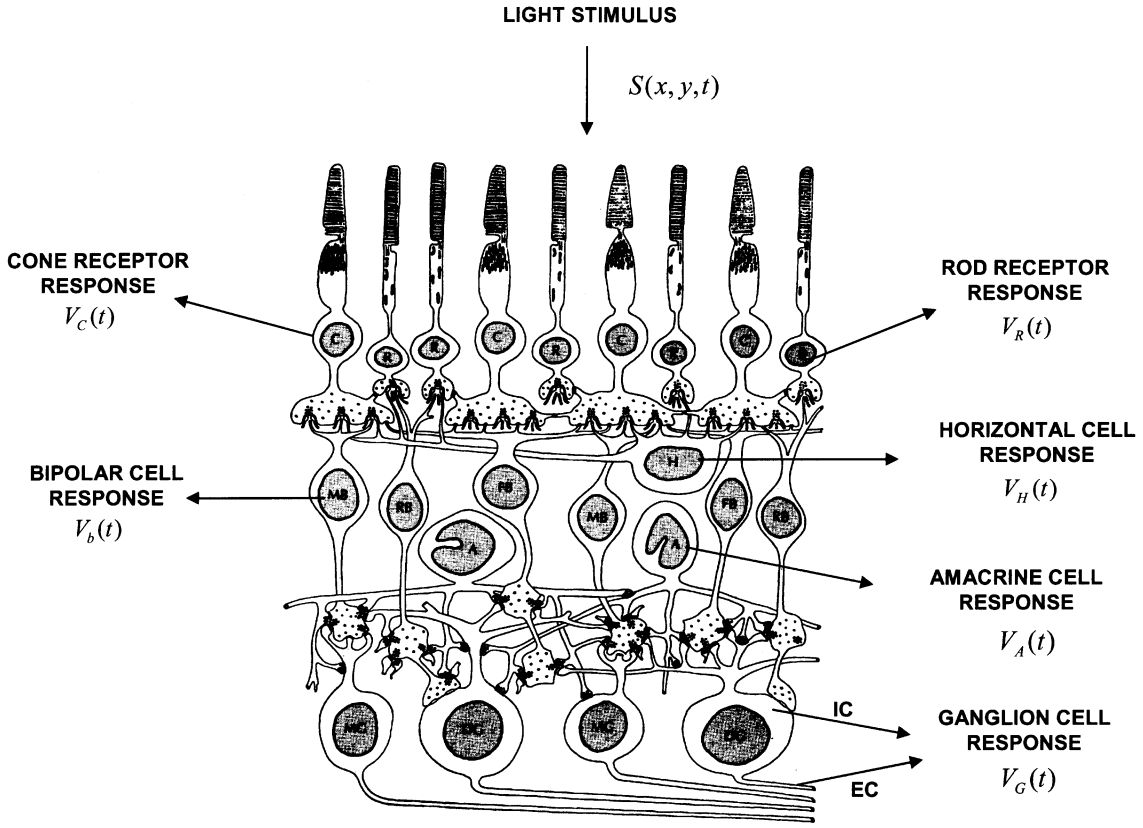


Figure 1.3
Schematic of the neuronal organization of the retina (Dowling & Boycott, 1966).

Thus, visual information conveyed by the time variations of light intensity is converted (encoded) into sequences of action potentials by the ganglion cells and transmitted to higher levels of the visual system via the optic nerve. This "encoding" process represents cascaded signal transformations effected by the various retinal neurons and their multiple interconnections. Therefore, one "system of interest" can be defined by considering as input signal the variations of light intensity impinging on the photoreceptors and as output signal the resulting sequence of action potentials generated by the ganglion cells. The induced intracellular potential in any other retinal neuron along this cascade of signal transformations can be used as an output signal in order to define a different "system of interest". A block diagram depicting the main neuronal interconnections in the retina is shown in Figure 1.4 that suggests many different possibilities for defining input/output signals and the corresponding "systems of interest".

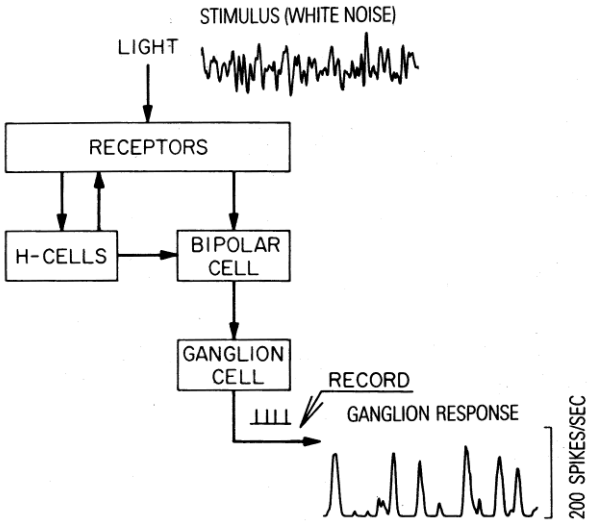


Figure 1.4
A block diagram depicting the main signal-flow pathways in the vertebrate retina and a light stimulus with the resulting ganglion cell response. Other stimulus-response pairs can be chosen experimentally (Marmarelis & Marmarelis, 1978).

In the foregoing, we described briefly the temporal sequence of causal effects from input photons impinging on the photoreceptor cell to the output action potentials generated by the ganglion cells. However, it is clear that complicated causal effects also occur in space through the multiple lateral interconnections among retinal neurons and interneurons. Thus, space can be viewed as another independent variable (in addition to time) to define retinal input-output signal transformations. This leads to the advanced spatio-temporal analysis of the visual system

discussed in Section 7.4.1. Note that the wavelength of the input photons can be viewed as yet another independent variable in studies of color vision.

The first successful application of the cross-correlation technique of modeling (see Section 2.2.3) on visual neuronal pathways using band-limited Gaussian white noise (GWN) test inputs was performed on the catfish retina [Marmarelis & Naka, 1972]. The recorded output signal was the sequence of action potentials generated by a ganglion cell (actually the "probability of firing" measured by superimposing the outputs of repeated trials with the same GWN input, also called the "peristimulus histogram"). The obtained nonparametric model took the form of a discrete-time second-order Wiener model:

$$y(n) = h_0 + \sum_m h_1(m)\,x(n-m) + \sum_{m_1}\sum_{m_2} h_2(m_1, m_2)\,x(n-m_1)\,x(n-m_2) - P\sum_m h_2(m,m) \qquad (1.4)$$

where h_0, h_1, h_2 denote the discretized Wiener kernels of zeroth, first and second order, P is the power level of the discretized GWN input $x(n)$ (light intensity), and the summation over m, m_1, and m_2 covers the entire memory of the system kernels. The discretized output $y(n)$ represents the probability of firing an action potential by a single ganglion cell at discrete-time index n ($t = nT$, where T is the sampling interval).

The first-order and second-order Wiener kernels of the horizontal-to-ganglion neuronal pathway, estimated through the cross-correlation technique, are shown in Figure 1.5. The interpretation of these kernels became a key issue from the very beginning, instigating intensive debates regarding the potential utility of this approach. Many arguments were voiced for and against this approach, some of which remain the fodder of lively debate to date. However, the fact remains that this general nonparametric approach represents a quantum leap of improvement over any other method used heretofore and, although some interpretation issues remain open, it has already elucidated immensely our understanding of retinal function. For instance, it is evident from the waveform of h_1 that the system is encoding both intensity and rate-of-change information (as discussed further in Section 6.1.1). It is also evident from the form of h_2 that the system exhibits rectifying nonlinearities, consistent with the presence of a threshold for the generation of action potentials (see the contribution of h_2 to the model prediction in Figure 1.6). The validation for this model is provided by its ability to predict the output signal to any given input signal, as demonstrated in Figure 1.6.

Many other "systems of interest" have been defined in the retina by considering the outputs of other retinal neurons (e.g., horizontal, bipolar, amacrine) and/or other inputs (e.g., light intensity, current injected into various cells, or light stimuli of different wavelengths). Extensive modeling studies have been reported for these systems by Naka and his associates following the nonparametric approach (see Section 6.1.1).

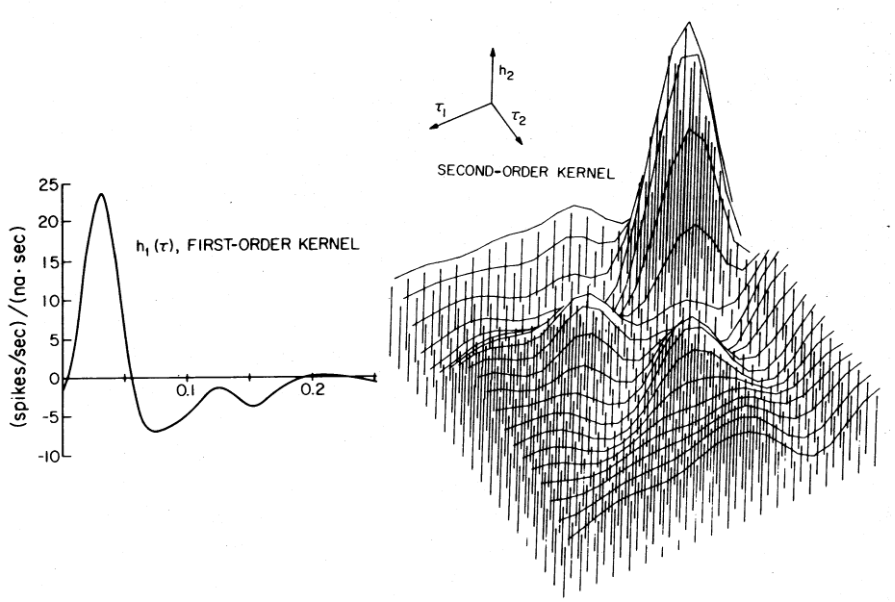


Figure 1.5
The first and second order Wiener kernel estimates of the horizontal-to-ganglion cell system in the catfish retina, obtained via the cross-correlation technique using band-limited GWN stimuli of current injected into the horizontal cell layer (Marmarelis & Naka, 1972).

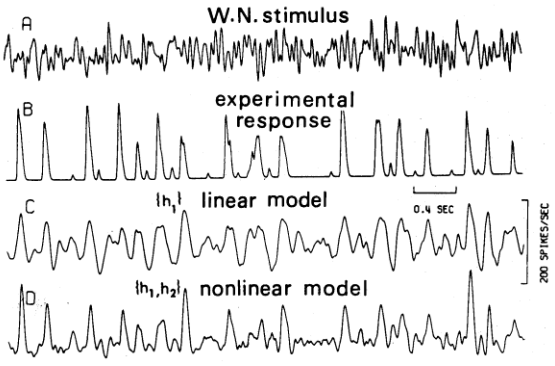


Figure 1.6
The recorded experimental response of the ganglion cell (second trace) represented as "frequency of firing" (or peristimulus histogram) for repeated trials of the band-limited GWN current stimulus shown in the top trace. The predictions of the linear (i.e., first-order) model and of the nonlinear (i.e., second-order) model using the Wiener kernels of Figure 1.5 are shown in the third and fourth traces respectively (Marmarelis & Naka, 1972).

Example 1.2

Invertebrate photoreceptor

As a second illustrative example, we consider the modular (block-structured) model derived for the photoreceptor of the fly eye (retinula cells 1-6 in an ommatidium of the composite eye of the fly *Calliphora erythrocephala*) using CSRS quasi-white stimuli under light-adapted conditions [Marmarelis & McCann, 1977]. We have found that the input-output relation of this system can be described by means of a modular model comprised of the cascade of a linear filter L followed by a quadratic static nonlinearity N (see Figure 1.7). If the impulse response function of the filter L is denoted by $g(m)$ and the static nonlinearity N is the quadratic function: $y = c_1 v + c_2 v^2$, then the equivalent discrete-time nonparametric model takes the form of a discrete Volterra model similar to the Wiener model of Equation (1.4), except for the first and last terms of the right-hand side that become zero in the Volterra model (see Section 2.2.1). The first and second order discrete Volterra kernels of this model (all other kernels are zero) are given by the expressions:

$$\tilde{k}_1(m) = c_1 g(m) \tilde{u}(m) \tag{1.5}$$

$$\tilde{k}_2(m_1, m_2) = c_2 g(m_1) g(m_2) \tilde{u}(m_1) \tilde{u}(m_2) \tag{1.6}$$

where $\tilde{u}(m)$ denotes the discrete step function (zero for $m < 0$ and 1 for $m \geq 0$) manifesting the causality of the model, and $g(m)$ is shown in Figure 1.7 (for the proof, see Section 4.1.2).

We can obtain the equivalent parametric discrete-time model through "parametric realization" of $g(m)$ (see Section 3.4), given by the two equations:

$$v(n) = \alpha_1 v(n-1) + \ldots + \alpha_K v(n-K) + \beta x(n) \tag{1.7}$$

$$y(n) = c_1 v(n) + c_2 v^2(n) \tag{1.8}$$

where the linear difference equation (1.7) describes the linear filter L (for properly chosen order K) and the algebraic equation (1.8) describes the static nonlinearity N. It is evident that if we seek to substitute v in terms of y in Equation (1.7) by solving Equation (1.8) with respect to v, we arrive at an irrational expression in terms of y (i.e., this system cannot be represented by a single rational nonlinear difference equation). In control engineering terminology, Equation

(1.7) can be viewed as a "state equation" and Equation (1.8) as the "output equation". This parametric model is also isomorphic to the L-N modular model in this case.

The equivalent connectionist model for this system requires an infinite number of hidden units, if the conventional sigmoidal activation functions are used (see Section 4.2.1). However, the use of polynomial activation functions allows for an equivalent connectionist model with a single hidden unit, as discussed in Section 4.2.1.

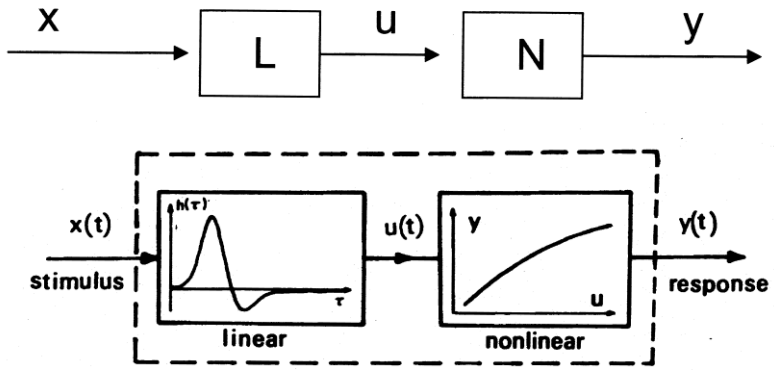


Figure 1.7
The L-N cascade model (a linear filter L followed by a static nonlinearity N) obtained for the photoreceptor of the fly *Calliphora erythrocephala* (Marmarelis & McCann, 1977).

Example 1.3

Volterra analysis of Riccati equation

The third example is chosen to be a parametric model of a nonlinear differential system described by the well-studied Riccati equation:

$$\frac{dy}{dt} + ay + by^2 = cx \tag{1.9}$$

where $x(t)$ is viewed as the input and $y(t)$ as the output of the system. This model exhibits a squared-output nonlinearity and can be also written as:

$$L(D)\,y = cx - by^2 \tag{1.10}$$

where $L(D)$ represents the differential operator: $D + a$, with D denoting differentiation over time. The form of Equation (1.10) implies that this model can be also viewed as a nonlinear feedback model with a linear feedforward component: $cL^{-1}(D)$, and a static nonlinear (square) negative feedback, as shown in Figure 1.8. This negative feedback formulation represents a

modular (block-structured) model, equivalent to the parametric model of Equation (1.9). In Figure 1.8, we also show an equivalent "circuit model", since this type of equivalent model form has been used extensively in physiology for parametric models described by differential equations. We consider the equivalent circuit model as another form of a parametric model (since it can be directly converted into a system of differential equations, and vice versa).

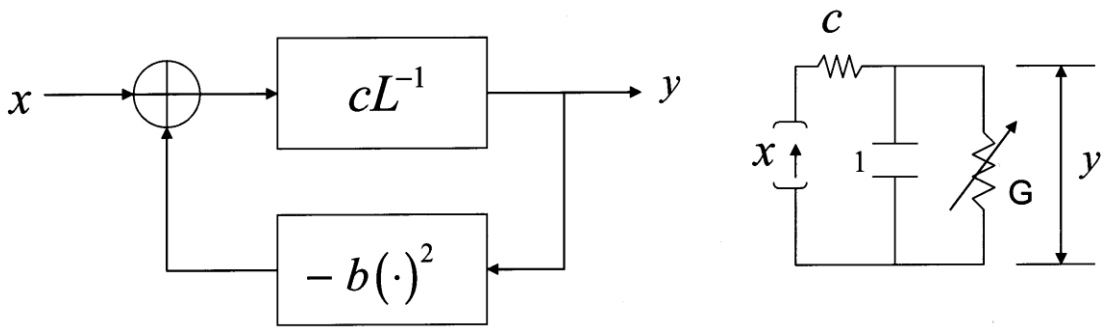


Figure 1.8
Equivalent modular (block-structured) model for the parametric model defined by the Riccati equation (1.9), depicting linear dynamic feedforward and nonlinear static feedback components (left panel). The equivalent "circuit model" (right panel) has a current source x flowing through a fixed conductance c, and the output is represented by the voltage y applied to a unit capacitance and a voltage-dependent conductance: G=a+by.

The equivalent nonparametric model for the Riccati equation is derived in Section 3.2 and corresponds to an infinite-order Volterra series. However, if we assume that $|b|$ is very small, then the higher order Volterra functional terms (higher than second order) can be neglected and the equivalent nonparametric Volterra model becomes approximately of second order, expressed in continuous time as:

$$y(t) \cong k_0 + \int_0^\infty k_1(\tau)\, x(t-\tau)\, d\tau + \int_0^\infty \int k_2(\tau_1,\tau_2)\, x(t-\tau_1)\, x(t-\tau_2)\, d\tau_1 d\tau_2 \tag{1.11}$$

where $x(t)$ and $y(t)$ denote the input and output signals respectively, and the Volterra kernels are given by the expressions:

$$k_0 = 0 \tag{1.12}$$

$$k_1(\tau) = ce^{-a\tau}u(\tau) \tag{1.13}$$

$$k_2(\tau_1,\tau_2) = -\frac{bc^2}{a} e^{-a(\tau_1+\tau_2)} \left[1 - e^{a\cdot\min(\tau_1,\tau_2)}\right] u(\tau_1)u(\tau_2) \tag{1.14}$$

where $u(\tau)$ denotes the continuous-time step function (0 for $\tau < 0$, 1 for $\tau \geq 0$). Note that the Volterra kernels depend on the Riccati parameters (as expected), but terms of order b^2 or higher have been neglected. In practice, these Volterra kernels are estimated from sampled input-output data (i.e., discretized signals) and they yield a discrete-time Volterra model. An equivalent continuous-time model can be obtained subsequently (if needed) by means of the "kernel invariance method" presented in Section 3.5.

The discrete-time Volterra kernels of first-order and second-order can be obtain by discretization of their continuous-time counterparts of Equations (1.13) and (1.14):

$$k_1(m) = \gamma \alpha^m \tilde{u}(m) \tag{1.15}$$

$$k_2(m_1, m_2) = \frac{\beta \gamma^2}{\ln \alpha} \alpha^{m_1 + m_2} \left[1 - \alpha^{-\min(m_1, m_2)} \right] \tilde{u}(m_1) \tilde{u}(m_2) \tag{1.16}$$

where the discrete-time parameters (α, β, γ) of the equivalent discrete-time parametric model that takes the form of a first-order nonlinear difference equation:

$$y(n) + \alpha y(n-1) + \beta y(n-1)^2 = \gamma x(n) \tag{1.17}$$

are distinct from the continuous-time parameters (a, b, c) and expressed in terms of the continuous parameters of the Riccati equation as: $\alpha = \exp[-aT], \beta = bT, \gamma = cT$, where T is the sampling interval.

The mathematical analysis of the equivalence between nonlinear differential and difference equation models is based on the "kernel invariance method" presented in Section 3.5. We note that the nonparametric model is not as compact as its parametric counterpart. On the other hand, the model specification task is greatly simplified in the nonparametric approach, and the estimation of the kernels of the nonparametric model can be accomplished by various methods using input-output data, as described in Chapter 2.

Example 1.4

Glucose-insulin minimal model

The fourth example is drawn from the metabolic/endocrine system and concerns the extensively studied dynamic interrelationship between blood glucose and insulin. The widely accepted "minimal model", used in connection with glucose tolerance tests, is a good example of

a parametric model for this nonlinear dynamic interrelationship [Bergman et al. 1981; Carson et al. 1983; Cobelli & Marmarelis, 1983]. This model is comprised of the following two first-order differential equations:

$$\frac{dG(t)}{dt} = -p_1\left[G(t) - G_b\right] - X(t)G(t) \tag{1.18}$$

$$\frac{dX(t)}{dt} = -p_2 X(t) + p_3\left[I(t) - I_b\right] \tag{1.19}$$

where $G(t)$ is the glucose plasma concentration (in *mg/dl*), $X(t)$ is the insulin action (in min^{-1}), $I(t)$ is the insulin plasma concentration (in $\mu U/ml$), G_b is the basal glucose plasma concentration (in *mg/dl*), I_b is the basal insulin plasma concentration (in $\mu U/ml$), p_1 and p_2 are two characteristic parameters describing the kinetics of glucose and insulin action respectively (in min^{-1}) and p_3 (in min^{-2} $ml/\mu U$) is a parameter determining the modulatory influence of insulin action on glucose uptake dynamics. It should be noted that this model does not take into consideration the pancreatic secretion of insulin induced by changes in glucose plasma concentration or the production of new glucose from internal organs (e.g., liver in response to elevation of plasma insulin), which can be described by separate differential equations (although this is far from a trivial task). The physiological parameters of glucose effectiveness $S_G = p_1$ (in min^{-1}) and insulin sensitivity $S_I = p_3 / p_2$ (in ml $\text{min}^{-1}/\mu U$) have been defined and used extensively in the literature for physiological/clinical purposes.

The above system is nonlinear, due to the bilinear term present in Equation (1.18) which gives rise to an equivalent nonparametric Volterra model of infinite order. However, it can be shown that, for the physiological range of the parameter values, a second-order Volterra model approximation is adequate for all practical purposes. Considering the variations of $I(t)$ around I_b as the input of the system and $G(t)$ as the output, we can derive the Volterra kernels of the system analytically using the generalized harmonic balance method (see Section 3.2). The resulting expressions for the zeroth, first, and second order kernels are (in first approximation, for $p_3 \gg p_1$):

$$k_0 = G_b \tag{1.20}$$

$$k_1(\tau) = -p_3 G_b h(\tau) \tag{1.21}$$

$$k_2(\tau_1, \tau_2) = \frac{G_b p_3^2}{2}\left\{ h(\tau_1)h(\tau_2) + p_1 \int_0^{\min(\tau_1,\tau_2)} \exp(-p_1\lambda)h(\tau_1-\lambda)h(\tau_2-\lambda)d\lambda \right\} \tag{1.22}$$

where:

$$h(\tau) = \frac{1}{p_2 - p_1}\left[\exp(-p_1\tau) - \exp(-p_2\tau)\right] \tag{1.23}$$

The rth-order Volterra kernel is proportional to p_3^r and can be neglected if $|p_3|$ is very small. Many other parametric models have been proposed for this system, typically comprising a larger number of "compartments" [Cobelli & Pacini, 1988; Vicini et al. 1999]. The equivalent first and second order Volterra kernels of the "minimal model" are shown in Figure 1.9 for typical parameters: $p_1 = 0.023$, $p_2 = 0.033$, $p_3 = 1.783 \cdot 10^{-5}$, $G_b = 80.25$. An equivalent modular model is shown in Figure 1.10, utilizing two linear filters, a multiplier, an adder and a feedback pathway. This modular (block-structured) model depicts the fact that the minimum model can be viewed as expressing a nonlinear (modulatory) control mechanism implemented by the feedback pathway into the multiplier.

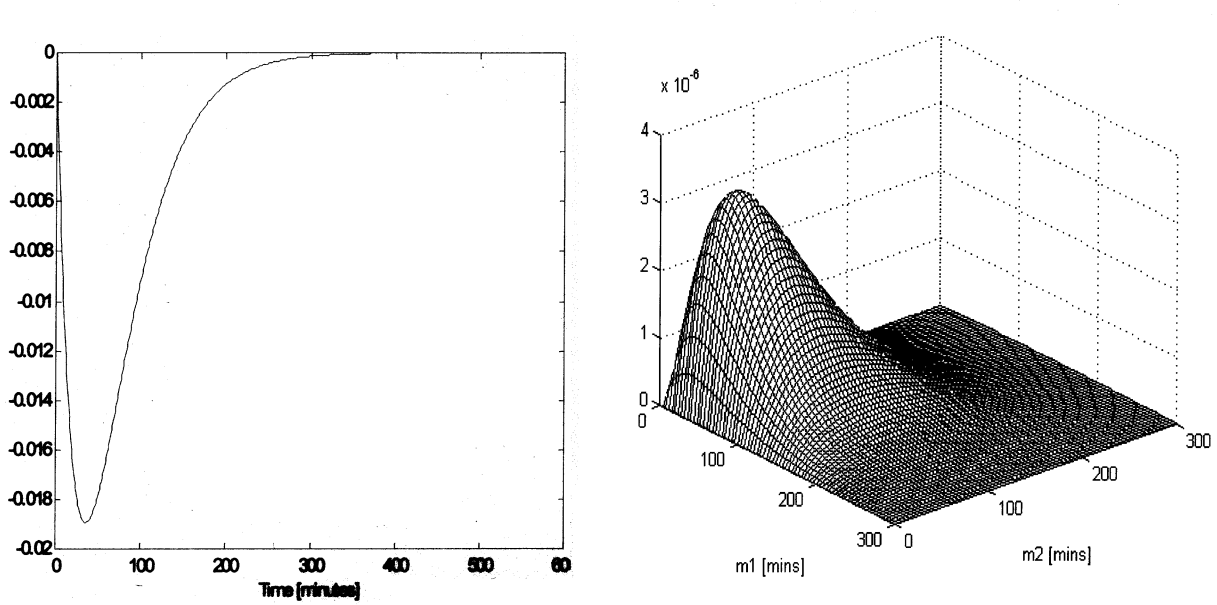


Figure 1.9
The equivalent first and second order Volterra kernels given by the Equations (1.21) and (1.22) for the insulin-glucose "minimal" model defined by Equations (1.18) – (1.19).

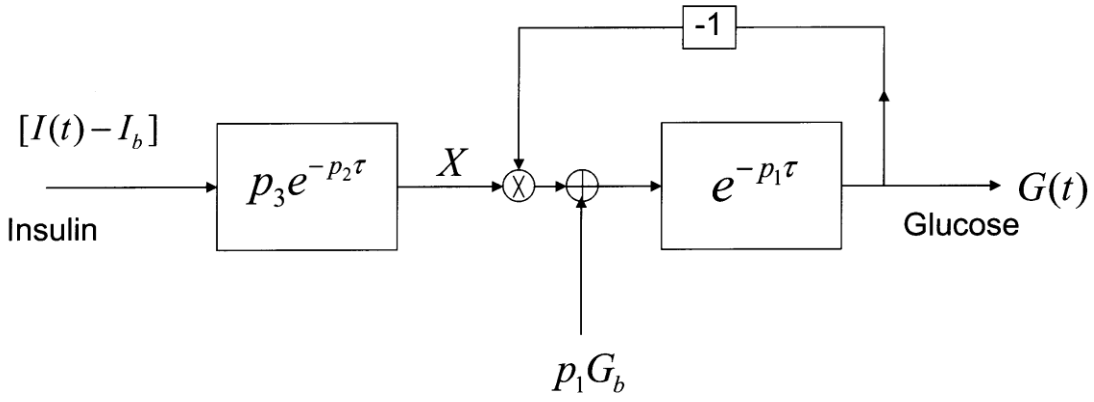


Figure 1.10

Equivalent modular (block-structured) model for insulin-glucose minimal model, utilizing two linear filters with impulse response functions: $p_3 \exp(-p_2\tau)$ and $\exp(-p_1\tau)$, an adder and a multiplier for negative multiplicative (modulatory) feedback (p_1G_b is a fixed reference defined level by the basal glucose value).

Example 1.5

Cerebral autoregulation

The fifth example is drawn from the cardiovascular system and concerns cerebral autoregulation. This is a challenging example of a physiological system with multiple closed (nested) loops that involve biomechanical, neural, endocrine and metabolic mechanisms interacting with each other. A simplified schematic of the protagonists in cerebral autoregulation is shown in Figure 1.11. A modular model obtained from real data is shown in Figure 1.12, depicting three parallel branches that correspond to the "principal dynamic modes" of this system (see Section 4.1.1). This modular model was obtained via the Laguerre-Volterra Network approach presented in Section 4.3, using mean arterial blood pressure data as input and mean cerebral blood flow velocity data as output. The model reveals the presence of (at least) three nonlinear dynamic mechanisms of cerebral autoregulation (see Section 6.2). Equivalent parametric, connectionist and nonparametric models can be obtained from this modular model (see Chapter 4).

The many technical details surrounding the derivation of these equivalent model forms are discussed in Chapters 2-4, along with the corresponding estimation methods and their relative performance characteristics. Important practical considerations for the successful application of these modeling methodologies and the required preliminary testing and error analysis in actual modeling applications are given in Chapter 5.

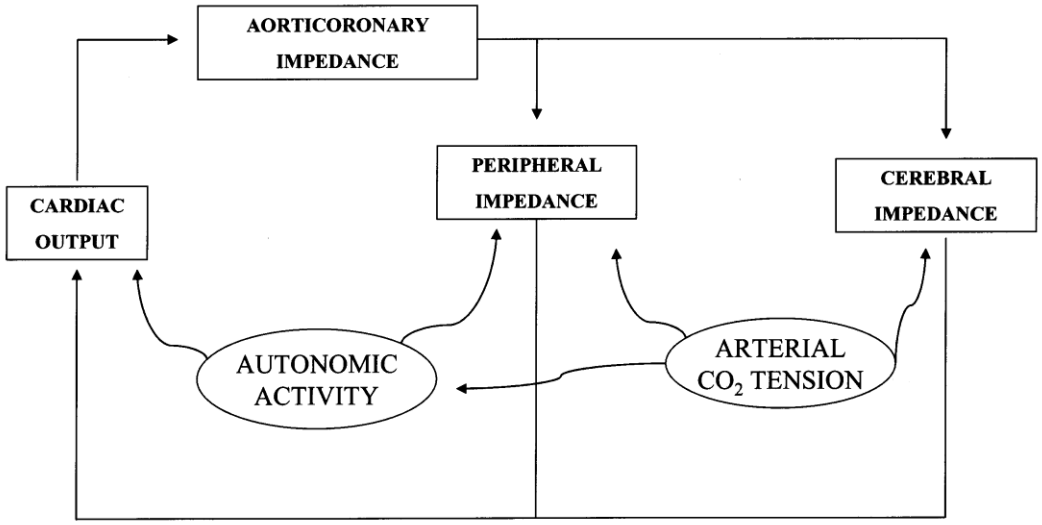


Figure 1.11
Schematic of the main protagonists in cerebral flow autoregulation.

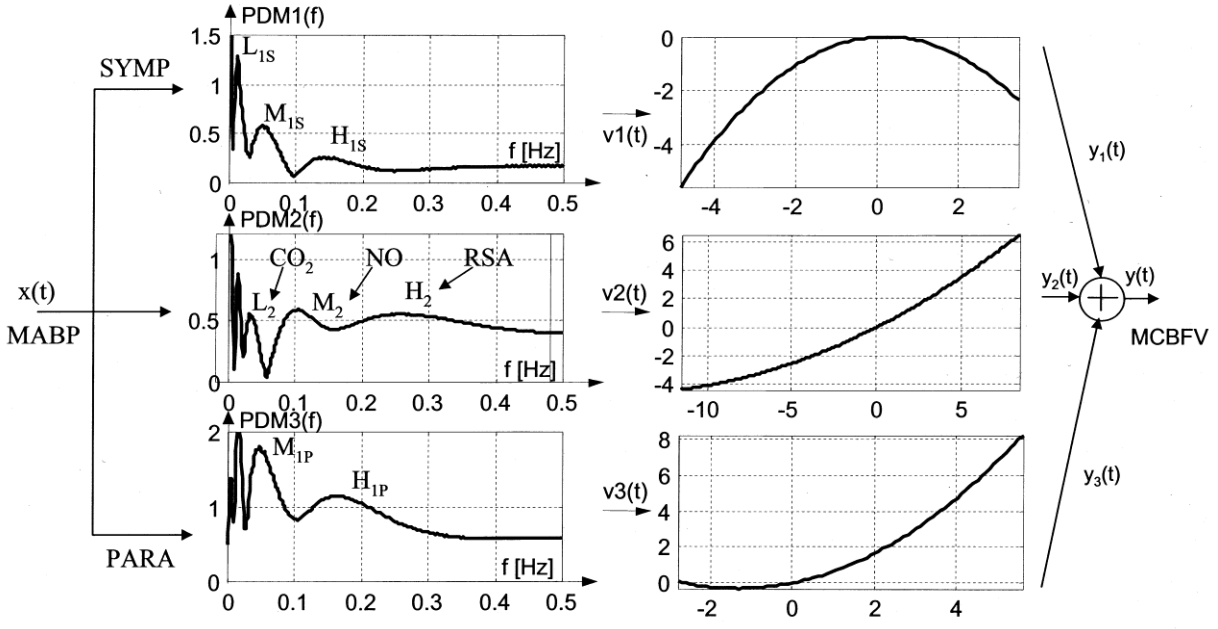


Figure 1.12
Modular model of cerebral flow autoregulation in a normal human subject, using the advocated methodology that starts with the general Volterra model and derives an equivalent "Principal Dynamic Mode" model of lower complexity. Each of the three branches is composed of a linear filter (a "principal dynamic mode" of the system) followed by a static nonlinearity. The model was obtained from 6 min long data of mean arterial blood pressure (input) and the corresponding mean cerebral blood flow velocity (output) sampled every 1 sec (for details, see Section 6.2).

1.5 DEDUCTIVE AND INDUCTIVE MODELING

The hypothesis-driven approach to scientific research offers a time-honored *deductive* path to scientific discovery that has been proven effective in the development of the physical sciences, closely associated with the reductionist viewpoint (i.e., proceeding deductively from first principles). However, this traditional approach encounters difficulties as the complexity of the problem under study increases, making the effective use of first principles unwieldy. This fact has given rise to a complementary approach that follows an *inductive* method based on the available data. In the inductive (data-driven) approach, priority is given to the data and the research effort is directed towards developing rigorous and robust mathematical and computational methods that extract the relevant information (models in our case) in a general context. This general context minimizes the *a priori* assumptions made about the system under study and, therefore, avoids possible "biasing" of the results by preconceived (and possibly restrictive) notions of the individual investigator. To the extent "unbiased knowledge" is acquired by this data-true inductive process, it can be incorporated in subsequent hypothesis-driven research to answer specific questions of interest and ultimately derive "general laws" that can be used deductively to advance scientific knowledge.

A synergistic approach is advocated in this book that commences with inductive (data-driven) modeling, following the methodologies presented in this book, and then formulates specific hypotheses that can be tested to answer unambiguously specific scientific questions of interest. In this manner, we secure the advantages of both approaches (inductive and deductive) and avoid their mutual shortcomings. In addition, this synergistic approach is time-efficient and cost-effective, because the inductive method places us quickly in the "neighborhood" of the correct model that can be further elaborated with regard to specific scientific questions of interest by use of hypothesis-driven research. The specific hypotheses depend, of course, on the goals of the study and therefore, cannot be prescribed beforehand, other than to indicate that they will be structured in a manner compatible with the available models.

Examples of the advocated synergistic approach are given in Chapter 6 where specific parametric or modular (block-structured) models are examined along with the previously obtained (data-true) nonparametric models in order to answer specific scientific questions and assist the interpretation of the models. Another class of examples pertains to the effects caused by the experimental change of a key controlling variable and the assessment of the resulting

quantitative changes in the model characteristics (e.g., effect of various drugs on cardiovascular, neural or metabolic function).

The advocated synergistic approach is appropriate for complex systems (because it obviates the need for vast reductionist/hierarchical structures) and protects the investigator from possible misleading results (when the assumptions made are restrictive or the testing conditions are not "natural"). It is hard to imagine a "downside" to this synergistic approach. In the worst case, when it is not necessary because the investigator is able to construct "perfect" hypothesis-driven tests, then we get definitive validation of the hypothesis-based results - - hardly a useless outcome. In all other cases, where existing reductionist knowledge and subjective intuition are either limited or yield unwieldy model postulates, the initial use of the data-driven approach can protect from potentially misleading results and can accelerate the pace of progress by placing us in the "neighborhood" of the correct answer. Further refinements/elaborations using the hypothesis-driven approach are subsequently possible and desirable.

One may wonder then why the advocated data-driven approach has not been used more extensively. The answer is that appropriate methodologies capable of tackling the *true complexity* of the problem (i.e., nonlinear, dynamic, nonstationary, multi-variate, nested-loop) have not been available heretofore. Lack of such methodologies forces investigators to use inadequate (restrictive) methods that are often unable to achieve the intended goals (i.e., the results are "biased" and often obtained under restrictive experimental conditions that do not place us in the "neighborhood" of the correct answer). As a result, the data-driven approach has not yet "proven" itself for lack of appropriate practicable methodologies, and investigators have seen no compelling reason yet to depart from the traditional hypothesis-driven approach. The overall goal of this book is to make available such appropriate methodologies to the biomedical research community at large and help usher a new era of advanced research on the *true* physiological systems - - and help the peer community avoid unrealistic simplifications, born of perceived necessity, that may breed misconceptions and perpetuate a state of studious confusion.

It is useful to remind the reader that the long debate between the reductionist and the integrative viewpoint (originating with Hippocrates in the 5^{th} century B.C. and lasting with undiminished intensity until the present time) is intertwined with the issue of hypothesis-driven research. The latter fosters a tendency towards fragmentation and static inquiry in a legitimized effort to construct and test clear and comprehensible hypotheses. This approach has borne

considerable benefits but also has placed serious limitations on those cases where multi-part dynamic interactions are of critical importance. For instance, when Erasistratus and the Anatomists were generating a wealth of anatomic knowledge in Alexandria of the 3rd century B.C., they were inadvertently diverting medical thought away from the fundamental Hippocratic concepts of the unity of organism and the dynamic disease process. This fact should not detract from the indisputable contributions of the Anatomists to medical science but should serve as a constructive reminder of the balance required in pursuing scientific research. Advancing knowledge is a multi-faceted endeavor and requires a multi-prong approach.

This point is not a mere historical curiosity, because a similar tag-of-ideas is taking place in our times (only in a much larger scale) between the reductionist approach espoused by molecular biology and the integrative approach advocated by systems biology/physiology. Nor is this debate an idle intellectual exercise, since it affects critically the direction of future research efforts. Although the integrative approach may follow either hypothesis-driven or data-driven methods, this book argues for a synergistic approach that gives priority to the data-driven methods in order to avoid self-entrapment within the comfortable confines of established viewpoints. A synergistic approach is also sensible in order to combine the benefits of the reductionist and integrative viewpoints. Although the desirability of this combination is self-evident, the historical record shows that even the best human minds have a tendency towards polarized binary thinking. The reasons for this tendency towards "binary thinking" are beyond the scope of this book, but certainly the root causes must be searched in the psycho-philosophical plexus of the human mind. The reader is urged to contemplate a way out of this "failing of the human mind" by taking into consideration the Galenic philosophical exhortations (see Historical Note #1 below).

Historical Note #1: Hippocratic and Galenic Views of Integrative Physiology [*]

Hippocrates is considered the founder of the medical profession, since he was the first to separate medicine from priestcraft and give it the independent status of a scientific discipline in Greece of the 5th century B.C. He was affiliated with the Asclepeion of Cos (a Greek island in the southeast Aegean) but also lived and worked in post-Periclean Athens. By providing a rational basis for medical practice and emphasizing the importance of clinical observation, Hippocrates did to medical thought what his contemporary Socrates did to thought in general: separated it from cosmological speculation. He gave the physician an independent status but held him to a high professional standard embodied in the "Hippocratic oath" that still defines the elevated duties of physicians worldwide.

Hippocrates observed that the human organism responds to external stresses/assaults (including disease) in a homeostatic manner (recuperation in the case of disease), i.e., the living organism possesses self-preserving powers and tends to maintain stable operation through complicated intrinsic mechanisms. He observed that each disease tends to follow a specific course through time and, therefore, it is a dynamic process (as opposed to a static state, which was the prevailing view at the time). Consequently, he emphasized *prognosis* over *diagnosis* and believed in the recuperative powers of the living organism. He advocated "giving nature a chance" to effect over time the adjustments that will cure the disease and restore health in most cases.

This broadly-defined homeostatic view led Hippocrates to the notion of the "unity of organism" that underpins integrative systems physiology to the present day. This integrative view tended to overlook the importance of the constituent parts and set him apart from the "reductionist" viewpoint espoused by the Anatomists of Alexandria. The labours of the latter in the 3rd century B.C. (especially Erasistratus) made outstanding contributions to medicine and gave rise to the sect of the Empiricists who proclaimed their concern with "the cure, and not the cause, of disease".

The reductionist viewpoint promulgated by the Empiricists was reinforced by the atomic theory of Leucippus and Democritus, as adapted to medicine by Asclepiades (1st century B.C.)

[*] Following A.J. Brock's introductory comments in his translation of Galen's "On the Natural Faculties", Harvard University Press, 1979 (first printed in 1916).

who introduced Greek medicine to Rome. A man of forceful personality and broad education, Asclepiades combined flamboyance with sharp intellect to achieve professional success and promote the view that physiological processes depend upon the particular way in which the indivisible particles of atomic theory come together. Although the validity of this fundamental view is indisputable (and self-evident), it did not lead to any constructive proposition for medical practice but served only as a public-relations vehicle for self-promotion in the intellectually shallow and "faddish prone" society of 1st-century Rome - - a phenomenon that is frequently recurring in history and all too familiar in our times as well. In fact, it can be argued that the disbelief of Asclepiades in the self-maintaining powers of the living organism and the intrinsic obstructionism of his maximalist approach caused a serious regression in the progress of medicine at that time. His views gave rise to the "Methodists" (founded by his pupil Themison) who espoused the simplistic pathological theory that all diseases belonged to two classes: one caused by constricted and the other by dilated pores traversing the molecular groups that compose all tissues. Another dubious trait established by the Methodists (and still prevalent to present time) is the tendency to invent a label for a perceived "disease" and then "treat the label" with no regard to the actual physiological processes that underpin the perceived disease.

The Empiricists and the Methodists were dominating Graeco-Roman medicine when Galen was born circa 131 A.D. as Claudius Galenos in Pergamos (a major Greek cultural center in Asia Minor during the Roman period). Galen, or more appropriately, Galenos ($\Gamma\alpha\lambda\eta\nu os$, meaning "tranquil" in Greek) had a benevolent and well-educated father, Nicon, who was a distinguished architect, mathematician and philosopher. Galenos received eclectic liberal education and studied the four main philosophical systems: Platonic, Aristotelian, Stoic and Epicurean. He pursued medical studies under the best teachers in Pergamos and afterwards in the other Hellenic centers of medical studies: Smyrna, Corinth and Alexandria. At the age of 27, he returned to Pergamos and was appointed surgeon to the gladiators. Four years later, driven by professional ambition, he went to Rome where he quickly achieved high distinction, rising to the coveted position of physician to the emperor Marcus Aurelius. Despite his broad acceptance and popularity, Galenos made no effort to conceal his contempt for the ignorance and charlatanism of most physicians in Rome. His courageous stand against corrupt medical practice, combined with professional envy, earned him many enemies in the medical circles of Rome who conspired against his life. To save his life, he fled Rome secretly at the age of 37 and retuned to his old

home in Pergamos, where he settled down to a literary life of philosophical contemplation and medical research. Even an imperial mandate a year later was not able to summon him back to Italy. Galenos pleaded vigorously to be excused and the emperor eventually consented, while trusting to his care the young prince Commodus. During the remaining 30 years of his life, Galenos wrote extensively on physiology, anatomy, and logic, providing the foundation for medieval medicine as the supreme authority (he was called the "Medical Pope of the Middle Ages") until Vesalius and Harvey disproved some of Galenos' basic cardiovascular premises with their seminal experiments and laid the foundation of modern anatomy and physiology, respectively, in the 16^{th} and 17^{th} century A.D.

In the six centuries that elapsed between Hippocrates and Galenos, the big debate in medicine revolved around the interrelated issues of integrative vs. reductionist viewpoint of physiology (Hippocrates' view of the unity of organism vs. the Atomists' view of decomposition into indivisible particles) and dynamic vs. static view of disease (Hippocrates' view of disease as a process vs. the anatomical view of the Empiricists). Galenos managed to put this debate to rest for 14 centuries by convincingly making the Hippocratic case. He re-established the Hippocratic ideas of the unity of the organism, the dynamic interdependence of its parts and its interaction with the environment (homeostasis). This constitutes the common conceptual foundation with our contemporary view of *integrative systems physiology* in a dynamic and homeostatic context that is espoused by this book. *The living system can only be understood as a dynamic whole* and not by static isolation of its component parts.

This fundamental principle is in direct opposition to the widespread view (even in our own times) that the whole can be understood by careful summation of the elaborated parts (reductionist viewpoint). The key difference concerns the emerging physiological properties from the dynamic interactions of the component parts and the interaction of the whole living system with its environment. In this, we stand today with Hippocrates and Galenos in uncompromising solidarity.

Galenos was not only a man of great intellect but also possessed a strong moral constitution. In his book "That the best Physician is also a Philosopher", he stipulated that a physician should have perfect self-control, should live laborious days and should be disinterested in money and the weak pleasures of the senses. Clearly, he would be a "hard sell" today. He calls on the physicians to be versed in: (a) *logic*, the science of how to think; (b) *physics*, the science of what

is nature; (c) *ethics*, the science of what to do. We must always remain aware of his concerns that medicine should not be allowed to fall into the hands of competing specialists without any organizing scientific, philosophical and moral principles. His recorded thoughts remain an inspiration forever and guide the constant evolution of medical science and practice.

CHAPTER 2

NONPARAMETRIC MODELING

INTRODUCTION

Nonparametric modeling constitutes, at present, the most general and mature (i.e., tested and reliable) methodology for nonlinear modeling of physiological systems. Its main strengths and weaknesses are summarized below.

The main *strengths* of nonparametric modeling are:

- it simplifies the model *specification* task and yields nonlinear dynamic "data-true" models for almost all physiological systems
- it yields *robust* model estimates from input-output data in the presence of ambient noise and systemic interference
- it allows derivation of equivalent parametric and modular model forms that facilitate physiological *interpretation* of the model
- it is extendable to nonlinear dynamic modeling of physiological systems with *multiple* inputs and outputs (including spatio-temporal, spectro-temporal and spike-train data)
- it is extendable to nonlinear dynamic modeling of many *nonstationary* physiological systems (including adapting and cyclical behavior)

The main *weaknesses* of nonparametric modeling are:

- it requires judicious attention to maintain the compactness of the model, lest it become unwieldy for highly nonlinear systems
- it requires appropriate input-output experimental or natural data (i.e., broadband and in sufficient quantity)
- physiological interpretation of the model may require derivation of equivalent parametric or modular (e.g., PDM model) forms

Nonparametric modeling employs the mathematical tool of a *functional* (a term due to Hadamard) that is a function of a function. As an alternative, the term "function of a line" was used by Volterra in his early work. A functional can represent mathematically the input-output transformation performed by

a causal *system*. The functional defines the mapping F of the past and present values of the *input* signal $x(t)$ (a function) onto the present value of the *output* signal $y(t)$ (a scalar):

$$y(t) = F\left[x(t'), t' \le t\right] \tag{2.1}$$

The objective of nonparametric modeling is to obtain an explicit mathematical representation of the functional F using input-output data (i.e., to derive *inductively* an empirical, true-to-the-data mathematical model of the system input-output transformation).

This conceptual/mathematical framework can be extended to the case of multiple inputs and multiple outputs, whereby each output in characterized by its own functional operating on all inputs that have a causal link to this specific output. Thus, the "system" is defined by its inputs and outputs, that are selected by the investigator to serve the objectives of the specific study. It is important to realize the immense flexibility afforded the investigators in defining the "system of interest" and the critical ramifications of this selection vis-à-vis the objectives of their study.

When the system is *stationary* (time-invariant), this mapping rule F remains fixed through time, facilitating the modeling task. The reader is alerted to avoid the pitfall of confusing an output signal varying through time (which is always the case) with temporal variation of the rule F. The case of *nonstationary* systems (whereby the mapping rule F varies through time) is far more challenging from the point of view of obtaining an explicit mathematical model from input-output data. This book focuses on stationary system modeling that has been the subject of most studies to date, although nonstationary modeling methods are also discussed in Chapter 9.

It is important to note that the mathematical formulation of Equation 2.1 applies to *dynamic* systems although the latter are often associated with differential equation models (parametric models). A system is dynamic (and causal) when the present value of the output depends on the present *and* past values of the input, as indicated by the functional notation of Equation 2.1. Another definition of dynamic systems, beyond the conventional differential equation formalism, can be based on whether or not the effects of an instantaneous (impulsive) input on the output spread over time. This is illustrated in Figure 2.1, where the distinction is also made between amplitude nonlinearity and dynamic nonlinearity.

The general approach to nonparametric modeling of nonlinear dynamic systems from input-output data is based on the Volterra functional expansion (or Volterra series) and its many elaborations or variants, including the orthogonal Wiener series for Gaussian white noise inputs. Because of its fundamental importance, we begin the chapter with a thorough discussion of the Volterra series and models (Section 2.1) in continuous and discrete time, as well as practical issues of Volterra kernel

estimation. We continue with the orthogonal Wiener series (Section 2.2), although its early importance is rapidly waning due to novel estimation methods, which do not require orthogonality. Methodologies for the estimation of the Wiener kernels (the unknown quantities in Wiener models) or kernels from other orthogonal functional expansions using quasi-white test inputs (e.g., CSRS) are also presented in Section 2.2. Efficient methodologies for the estimation of Volterra kernels, that are critically important in actual applications, are discussed in Section 2.3, including the most efficient method to date (Laguerre expansion technique). Emerging alternative methods for Volterra system modeling, based on equivalent network models and iterative estimation techniques, are discussed in Section 2.4. Chapter 2 concludes with a discussion of model estimation errors in a practical context using actual experimental data.

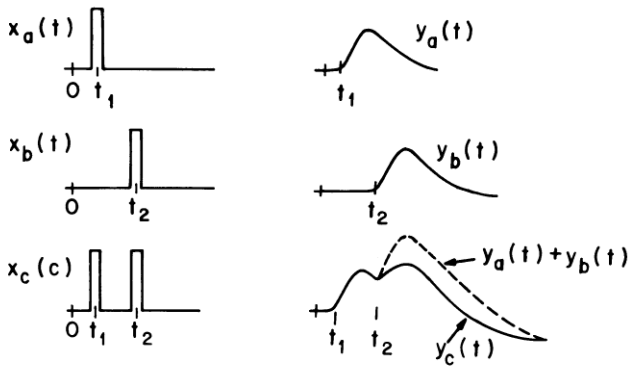


Figure 2.1

Illustration of dynamic effects of an impulsive input $x_a(t)$ at time t_1, manifested as the spread of the elicited response $y_a(t)$ beyond time t_1 (top trace). The first three traces illustrate the violation of the superposition principle by this system for two impulsive inputs that indicate the presence of dynamic nonlinearity in this system (in contradistinction to an amplitude nonlinearity that will be manifested as a violation of a linear scaling relation between input and output) [Marmarelis & Marmarelis, 1978].

The extension of these methodologies to the case of multiple inputs and multiple outputs is presented separately in Chapter 7 (including the case of spatio-temporal inputs and outputs), because it is attracting growing attention and importance. Likewise, the case of neural systems with spike inputs and/or outputs (sequences of action potentials) is discussed separately in Chapter 8, because of the significant methodological modifications required by this distinct signal modality.

2.1. VOLTERRA MODELS

The development of Volterra models relies on the mathematical notion of the Volterra series (a functional power series expansion) introduced by the distinguished Italian mathematician Vito Volterra[1] about a century ago [Volterra, 1930]. The term "Volterra series" is used to denote the (generally infinite) functional expansion of an analytic functional representing the input-output relation of any continuous and stable nonlinear dynamic system with *finite memory*.

The requirement of finite memory is necessary for the convergence of the Volterra functional series expansion (i.e., for the stability of the system) and excludes chaotic systems and autonomous (but not forced) nonlinear oscillators from this mathematical representation. The term "continuous" is appropriate for physiological systems and seeks to waive the mathematical requirement of analyticity of the system functional, because even non-analytic, but continuous, functionals (corresponding to systems with non-differentiable continuous input-output relations) can be approximated to any desired degree of accuracy by an analytic functional (and therefore a Volterra series) in a matter akin to the Weierstrass polynomial approximation theorem for non-analytic continuous functions. Thus the applicability of the Volterra series expansion to system modeling is very broad and requires a minimum of prior assumptions. We advocate its use because it offers practically a *universal* modeling framework for physiological systems.

It is unclear whether Volterra anticipated the profound implications of the proposed functional series expansion on the system modeling problem. At that time, his work on functionals and integro-differential equations was primarily motivated by the desire to understand phenomena in nonlinear mechanics (e.g., hereditary elasticity and plasticity extending to hysteresis) and later in population dynamics (e.g., prey-predator models). Since the conceptual framework of system science (i.e., the notion of a system as an operator transforming an input into an output) was not part of the scientific thinking in the beginning of the 20th century, it is likely that Volterra never conceived his functional expansion ideas in the context of the system modeling problem as we currently conceptualize it. This was evidently done in the 1940's by another great thinker and mathematician of the 20th century, Norbert Wiener, the father of cybernetics[2]. Wiener, in his seminal monograph [Wiener, 1958], made this critical connection between the nonlinear system modeling /identification problem and functional expansions

[1] For brief bio of Vito Volterra, see Historical Note #2 at the end of this chapter
[2] For brief bio of Norbert Wiener, see Historical Note #2 at the end of this chapter

(surprisingly, without acknowledgement of Volterra's preceding work). Wiener's fundamental contributions will be discussed in the following section.

Returning to Volterra's fundamental contributions to our subject, we note that the introduction of the Volterra "functional power series" (as he termed it) occupies only one page in his seminal monograph on the "Theory of Functionals and Integro-Differential Equations" (p. 21 in the Dover edition). Two more pages (pp. 19-20) are devoted to the introduction of the "regular homogeneous functionals" of higher degree (what we now call the Volterra functionals) and the extension of the Weierstrass theorem to continuous functionals – also a Frechet contribution around the same time [Frechet, 1928]. It is worth noting the disproportionate impact of these three pages (out of a 160-page monograph) on the future development of system modeling theory and on Volterra's posterity. Nonetheless, in acknowledging him, we honor the great impact of his entire scientific work, as well as the brilliance and overall intellectual stature of a remarkable individual.

Volterra's pivotal idea is the transition from a finite dimensional vector space to an enumerably infinite vector space and then to a continuous function space. In other words, the Volterra series may be viewed as a generalization of the well-known Taylor multi-variate series expansion of an analytic function, f, of m variables as $m \to \infty$. The multi-variate Taylor series expansion of an analytic function $f(x_1,...,x_m)$ about a reference point $(x_1^*,...,x_m^*)$ in the m-dimensional vector is space defined by these m variables as:

$$f(x_1,...,x_m) = f(x_1*,...,x_m*) + \sum_{i=1}^{m} \alpha_i (x_i - x_i^*) + \sum_{i_1=1}^{m} \sum_{i_2=2}^{m} \alpha_{i_1,i_2} (x_{i_1} - x_{i_1}^*)(x_{i_2} - x_{i_2}^*) + ... \qquad (2.2)$$

and evolves into the Volterra functional power series as $m \to \infty$, where the origin of the real axis is used as the reference point (i.e., $x_i^* = 0$). Then the vector $[x_1,...,x_m]$ turns into a continuous function $x(\lambda)$ for λ in the interval $[a,b]$, and the analytic function f turns into the analytic functional F that can be expressed as the Volterra series expansion:

$$F[x(\lambda)] = k_0 + \int_a^b k_1(\lambda) x(\lambda) d\lambda + \int_a^b \int k_2(\lambda_1,\lambda_2) x(\lambda_1) x(\lambda_2) d\lambda_1 d\lambda_2$$

$$+ ... + \int_a^b ... \int k_r(\lambda_1,...,\lambda_r) x(\lambda_1) ... x(\lambda_r) d\lambda_1 ... d\lambda_r + ... \qquad (2.3)$$

where k_r represents the limit of the multi-variate Taylor expansion coefficients $\alpha_{i_1 \ldots i_r}$ and is termed the "*Volterra kernel"* of r^{th} order. The multiple integrals are termed the "regular homogeneous functionals" or, simply, "*Volterra functionals*".

The coefficients of the Taylor series expansion in Equation (2.2) are determined by the partial derivatives of the analytic function f evaluated at the reference point. Likewise, the kernels of the Volterra series expansion in Equation (2.3) are determined by the partial derivatives of the analytic functional F as defined by Volterra and others. This does not necessarily provide the requisite physical or physiological insight into the meaning of the Volterra kernels for an actual system – an important issue that will be addressed later in the context of nonlinear system dynamics using illustrative examples.

With regard to the analyticity requirement of the Volterra series expansion, we already indicated that it can be relaxed to continuous system functionals based on an extension of the Weierstrass theorem due to Frechet and Volterra. This implies broad applicability of the Volterra expansion in practice, since it is difficult to disprove experimentally the continuity of a physiological system. Nonetheless, if discontinuity exists, then either separate Volterra representations can be sought for the different operational regions where the system is continuous or a satisfactory continuous model approximation may be obtained for the entire operational region of the system. After all, the difference between a "discontinuous" jump and a "very rapid" (but continuous) change is expected to be experimentally opaque in most cases. This remains a topic of open inquiry, although it is not expected to affect practically but a minute percentage of applications to physiological systems.

With regard to the issue of convergence of the Volterra series, we note that lack of convergence is experimentally manifested either as an instability of the system output for certain inputs or as the inability of the Volterra model to represent the system output with satisfactory accuracy. The former manifestation is experimentally evident, however the latter may be ambiguous since other factors (e.g., noise or inadequate model specification and/or estimation) may prevent the model from achieving satisfactory accuracy. The mathematical condition for convergence of the Volterra series can be expressed as follows for the case of uniformly bounded inputs (i.e., $|x(t)| \le B$):

$$\int_0^\infty \ldots \int |k_r(\tau_1, \ldots \tau_r)| d\tau_1 \ldots d\tau_r \le \frac{A_r}{B^r} \tag{2.4}$$

where $\{A_r\}$ is a convergent series of nonnegative scalars. Note that the absolute integrability condition of Equation (2.4) incorporates the requirement of finite memory that was placed earlier on the

applicability of the Volterra modeling approach. It is evident that the finite-memory requirement is satisfied both in the strict sense of the finite kernel support (i.e., finite domain of nonzero kernel values) and in the asymptotic sense of kernel absolute integrability.

The form of the Volterra series that we will use throughout the book is slightly different from Equation (2.3) to conform with established practice of how the input signal is represented. Connecting also with the system perspective depicted by Equation (2.1), we can state that the output $y(t)$ of a stationary stable causal system (and all physiological systems are assumed causal) can be expressed in terms of its input signal $x(t)$ by means of the Volterra series expansion as:

$$y(t) = k_0 + \int_0^\infty k_1(\tau)x(t-\tau)d\tau + \int_0^\infty\int k_2(\tau_1,\tau_2)x(t-\tau_1)x(t-\tau_2)d\tau_1 d\tau_2 + ... + \int_0^\infty ...\int k_r(\tau_1,...,\tau_r)x(t-\tau_1)...x(t-\tau_r)d\tau_1...d\tau_r + ... \quad (2.5)$$

where the range of integration (τ_i from 0 to ∞) indicates that the input past and present values affect the output present value in a manner determined by the Volterra kernels (i.e., the kernels should be viewed as weighting functions in this integration). Therefore, the kernels represent the system-specific patterns of input-output causal effects that practically extend over a finite time interval μ (i.e., τ_i takes values from 0 to μ) termed the "memory extent" of the system.

Note that the Volterra kernels are causal functions (i.e., zero for negative arguments) and symmetric with respect to their arguments (i.e., invariant to permutations of their arguments). They characterize completely the system dynamics and form a hierarchy of nonlinear terms (the Volterra functionals) representing system nonlinearity in ascending order. The order of each kernel corresponds to the multiplicity of distinct values of the input past and present (termed the input "epoch") that partake in forming the system present output. In practical terms, the kernels ought to be estimated over the finite memory extent of the system, using input-output data.

The zeroth-order Volterra kernel, k_0, is the value of F (and of the output) when the input is absent (i.e., the input is the null function). For a stationary system, k_0 is constant. However, in an actual physiological context, the system output will not be generally constant even in the absence of input, due to the effects of a multitude of unobservable factors that act as noise or systemic interference. In this practical context, the constant k_0, selected for a stationary Volterra model, becomes the average value of the output variations due to this noise/interference (or, in other words, the average of the spontaneous activity of the system in the absence of any input). Note that k_0 can be expressed explicitly as a function of time for a nonstationary system/model (along with all other kernels), as discussed in Chapter 9.

The first-order Volterra kernel, $k_1(\tau)$, is akin to the "impulse response function" of linear system theory, although it does not determine the response to an impulsive input in the nonlinear context (see Section 2.1.2). It should be viewed as the linear component of the nonlinear system, describing the pattern by which the system weighs the input past and present values (termed hereafter the "input epoch") in order to generate the linear portion of the output signal (i.e., no nonlinear interactions are included).

As an illustrative example, consider the first-order kernel of the fly photoreceptor model shown in Figure 1.7 (i.e., the impulse response function of the filter L). It depicts the fact that input values of light intensity impinging on the photoreceptor have maximum positive impact on the output value of intracellular potential generated at the photoreceptor soma about 12 ms later (lag) and maximum negative impact at about 20 ms lag (in a linear context). The total impact is quantified for all time lags between input and output by the values of the first-order kernel (in a linear context). The actual impact on the output depends, of course, on the input epoch and the static nonlinearity N. The linear component of the system output can be computed by means of the convolution between the first-order kernel and *any* given input signal, as indicated by the first-order Volterra functional. The first-order kernel provides a complete quantitative description of the linear dynamic characteristics of the system, and its Fourier transform provides the frequency response characteristics of the system in a linear context (akin to the "transfer function" of linear system theory). Most physiological systems exhibit linearized behavior for very small input signals, therefore the first-order kernel usually yields a good approximation of the entire system output for small input signals.

An illustration of the convolution operation performed by the first-order Volterra functional is given in Figure 2.2, where an arbitrary input signal $x(t)$ is approximated by contiguous rectangular pulses of narrow width Δt. As $\Delta t \to 0$, $r(t-t_m)$ tends to $\left[k_1(t-t_m)x(t_m)\Delta t\right]$ and the first-order Volterra functional computes for each time t the summation of all $r(t-t_m)$ for t_m between t and $t-\mu$, where μ is the extent of the first order kernel $k_1(\tau)$.

The second-order kernel, $k_2(\tau_1,\tau_2)$, represents the lowest order nonlinear interactions (i.e., between two values of the input epoch as they affect the output present value) in the Volterra modeling framework. It can be viewed as the two-dimensional pattern (a surface) by which the system weighs all possible pairwise product combinations of input epoch values in order to generate the second-order component of the system output (see Figure 2.3). For a second-order Volterra system, this can be

illustrated by the interaction term $\theta_{1,2}(t)$ depicted in Figrure 2.1, that represents the second-order interaction between the two impulses at the input and is equal to a slice of the second-order kernel of this system (parallel to the diagonal) as shown in Figure 2.4. For higher order systems, there are also higher order interactions between the two input impulses that are represented by "slices" of higher order kernels.

$$x(t) \simeq \sum_n x(n\Delta t)\, p(t - n\Delta t) \longrightarrow y(t) \simeq \sum_n x(n\Delta t)\, r(t - n\Delta t)$$

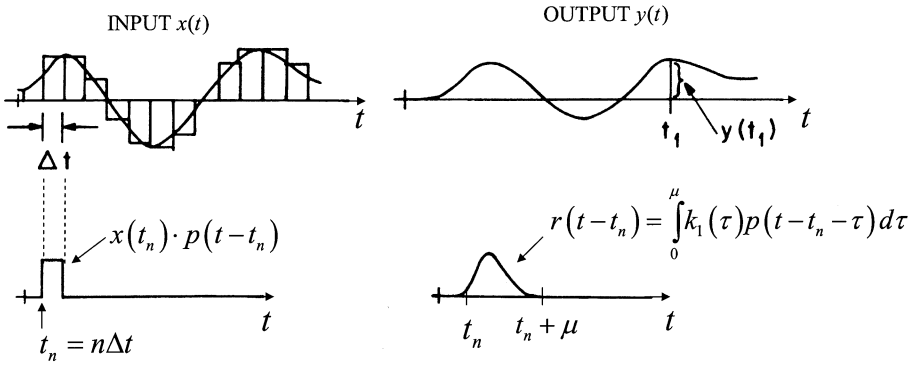


Figure 2.2
Illustration of the convolution operation performed by the first-order Volterra functional. The input signal $x(t)$ is approximated by contiguous rectangular pulses $p(t)$ of narrow width Δt and the resulting output signal $y(t)$ is the summation (superposition) of the individual responses $r(t - t_n)$ over the "memory" extent μ of the first-order kernel $k_1(\tau)$. Thus, the output at time t_1 contains contributions from all input pulses from $(t_1 - \mu)$ to t_1.

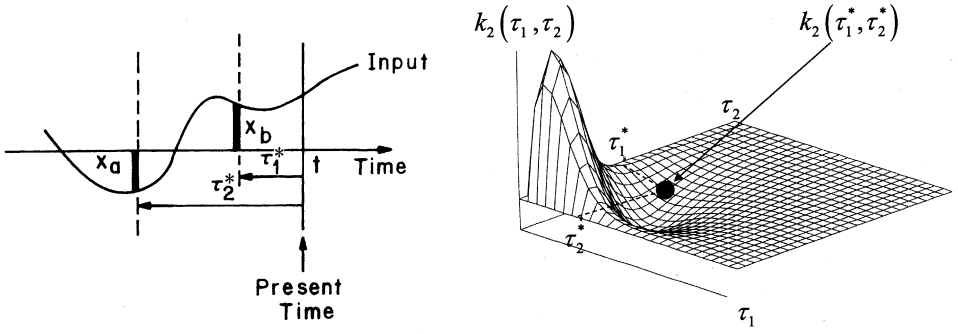


Figure 2.3
Second-order interaction of input epoch values $x(t - \tau_1^*)$ and $x(t - \tau_2^*)$ as they affect the output $y(t)$ in the manner determined by the value $k_2(\tau_1^*, \tau_2^*)$ of the second-order Volterra kernel. Contributions from all possible pairs of input epoch values are integrated to produce the second-order output component according to Equation (2.5).

As an example from a real second-order Volterra system, consider the aforementioned model of the fly photoreceptor. Its second-order kernel is shown in Figure 6.9 and depicts the fact that the maximum second-order effect on the present value of the system output is from the negative square of the input epoch values at about 12 ms lag. The maximum negative effect is from the product combination of input values at about 12 ms and 20 ms lags. The entire second-order impact of the input epoch values on the system present output is quantified by the second-order kernel for all possible pairwise combinations of input lagged values over the memory extent of the system.

The second-order component of the system output can be computed for *any* given input using the double convolution operation indicated by the second-order Volterra functional. The two-dimensional Fourier transform of the second-order kernel provides the complete bi-frequency response characteristics of the system, as discussed in Section 2.1.3. An illustration of the double convolution operation performed by the second-order Volterra functional is given in Figure 2.3 for a two-impulse input. The operational meaning of the second-order kernels is discussed further in Sections 2.1.2 and 2.1.3.

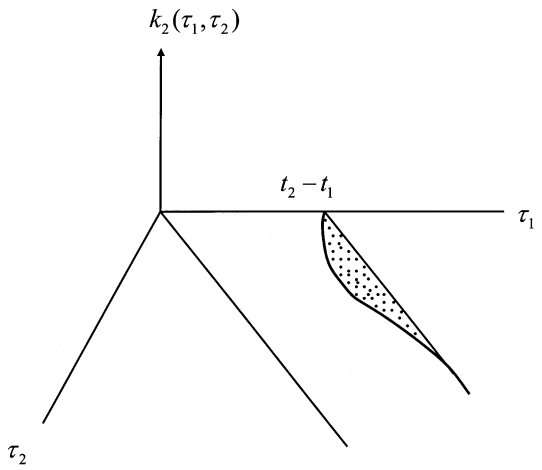


Figure 2.4
For a second-order Volterra system, the nonlinear interaction between two impulsive inputs (see Figure 2.1) is represented by a "slice" of the second-order Volterra kernel parallel to the diagonal (at $\tau_1 = t_2 - t_1$).

Higher order kernels represent the patterns of nonlinear interactions among a number of input epoch values equal to the order of the kernel. Their operational meaning will be further discussed in Sections 2.1.2 and 2.1.3. They are seldom obtained in actual applications (for order higher than second) since their estimation and interpretation becomes a formidable challenge due to the increasing dimensionality of the argument space. For this reason, we have developed an alternate approach to modeling higher order nonlinearities that facilitates estimation and interpretation (see Section 2.3.3).

Note that the Volterra functional expansion can be defined around any reference function (in addition to the null reference function for which it was initially defined) as long as we remain within regions of continuity and convergence. For instance, it is often the case in practice that a physical input signal cannot attain negative values (e.g., light intensity in visual stimulation). Then a non-zero (positive) reference level can be used to deliver physical stimuli with positive and negative deviations from this reference level that secures positive physical stimulus values. For stochastic stimuli (e.g., white noise), this reference level represents usually the mean of the random input signal and its value is greater than the maximum deviation from the mean.

This reference level can be used as a "computational zero" and the Volterra series expansion is defined around it. Of course, the obtained Volterra kernels depend on the specific reference level (in the same manner that the Taylor series coefficients depend on the reference point of the expansion) and the employed reference level must be explicitly reported when such kernels are published. For a nonzero reference level c, the associated set of Volterra kernels $\{k_r^c\}$ is related to the zero reference level set $\{k_r^0\}$ through the relation [Marmarelis & Marmarelis, 1978]:

$$k_r^c(\tau_1,...\tau_r) = \sum_{j=r}^{\infty} \binom{j}{r} c^{j-r} \int_0^{\infty} ... \int k_j(\tau_1,...,\tau_r,\tau_{r+1},...,\tau_j) d\tau_{r+1}...d\tau_j \qquad (2.6)$$

Some analytical examples of Volterra models are given below, followed by further discussion on the operational meaning of the Volterra kernels (Section 2.1.2) and the frequency-domain representation of Volterra models (Section 2.1.3) Since inductive modeling of physiological systems (i.e., directly from data) is performed in practice using sampled data, the mathematical expressions of the Volterra models must be adapted to discrete-time form, as discussed in Section 2.1.4. The practical estimation of the discrete-time Volterra kernels, that are involved in the discretized Volterra models, is initially discussed in Section 2.1.5 and elaborated further in Sections 2.3 and 2.4.

2.1.1. Examples of Volterra Models

To illustrate the use of Volterra models for nonlinear systems and their relation to equivalent parametric models, we use the following four examples.

Example 2.1

Static nonlinear system

We start with the simplest case of a static nonlinear system:

$$y = f(x) \tag{2.7}$$

If the function f is analytic, then we can use its Taylor series expansion:

$$y = \alpha_0 + \alpha_1 x + \alpha_2 x^2 + ... + \alpha_r x^r + ... \tag{2.8}$$

where, $\alpha_r = f^{(r)}(0)/r!$, with $f^{(r)}(0)$ denoting the r^{th} derivative of $f(x)$ at $x = 0$. This Taylor expansion determines the equivalent Volterra series with kernels $(r \geq 0)$:

$$k_r(\tau_1, ..., \tau_r) = \alpha_r \delta(\tau_1) ... \delta(\tau_r) \tag{2.9}$$

where $\delta(\tau)$ denotes the Dirac delta (impulse) function. [2]

If the function f is continuous (but not analytic), then a finite-degree polynomial approximation of any desired accuracy can be obtained based on the Weierstrass theorem:

$$f(x) \approx \alpha_0 + \alpha_1 x + ... + \alpha_Q x^Q \tag{2.10}$$

and the Volterra kernels are given by Equation (2.9) for $0 \leq r \leq Q$.

The evaluation of the coefficients $\{\alpha_r\}$ of the Weierstrass approximation can be made by use of polynomial expansions on bases defined over an expansion interval equal to the amplitude range of the system input (see Appendix I). This is illustrated in Figure 2.5 for one analytic and two non-analytic nonlinearities (one continuous and one discontinuous). Note that the expansion coefficients of polynomial approximations depend on the selected expansion interval, unlike the Taylor expansion that only depends on the derivatives of the (analytic) function at the reference point.

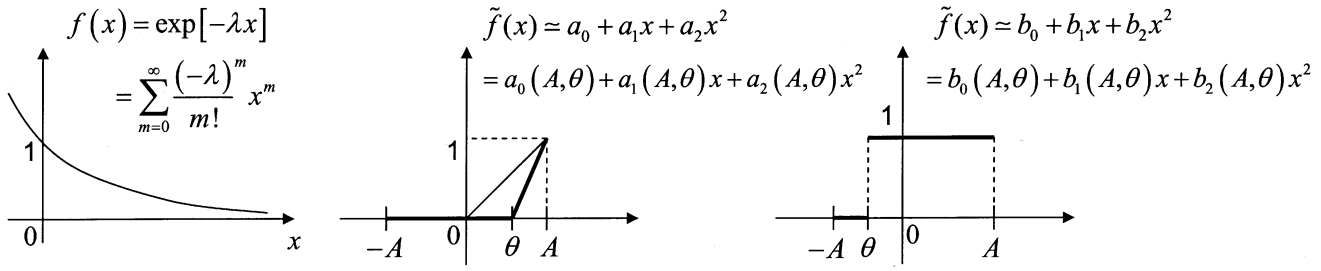


Figure 2.5
An analytic function and its Taylor series expansion valid for all x (exponential in left panel). Note that the coefficients of the second-degree polynomial (quadratic) approximations of the non-analytic continuous function (middle panel) and of the discontinuous function (right panel) in the interval $[-A, A]$ depend on A and θ (see Appendix I).

Example 2.2

L-N cascade system

We continue with a simple nonlinear dynamic system given by the cascade of a linear filter followed by a static nonlinearity (like the $L-N$ cascade model of the fly photoreceptor presented in Section 1.4 and shown in Figure 1.7). The equivalent Volterra model for this cascade system can be found by considering a polynomial representation of the static nonlinearity, as in Equation (2.8) or Equation (2.10):

$$y(t)=\sum_{r=0}^{Q}\alpha_r v^r(t) \tag{2.11}$$

where Q may be finite or infinite. Then, substitution of v given by Equation (2.12) as the convolution integral describing the input-output relation for the linear filter:

$$v(t)=\int_0^{\infty} g(\tau)\,x(t-\tau)\,d\tau \tag{2.12}$$

into Equation (2.11) yields the following analytical expression for the r^{th} order Volterra kernel of this $L-N$ cascade:

$$k_r(\tau_1,...,\tau_r)=\alpha_r g(\tau_1)...g(\tau_r) \tag{2.13}$$

---

[2] The Dirac delta function $\delta(t-t_0)$ (also known as the impulse function) is defined as zero whenever its argument is nonzero (i.e., $t\neq t_0$) and tends to infinity at $t=t_0$. The key defining relation is: $\int_{t_0-\varepsilon}^{t_0+\varepsilon} f(t)\delta(t-t_0)dt=f(t_0)$ for any continuous function $f(t)$ in the ε-neighborhood of t_0.

where $g(\tau)$ is the impulse response function of the linear filter L (for detailed derivation see Section 4.1.2). It is evident that the Volterra kernels of such a cascade system have a very particular structure that is amenable to interpretation and can be quickly ascertained by visual inspection, viz., for specific values of j tau arguments, the rth-order Volterra kernel is proportional to the $(r-j)$th-order Volterra kernel of this system. This is typically ascertained in practice by examining a possible scaling relation between a "slice" of the 2nd-order kernel estimate (along τ_1 or τ_2) and the 1st-order kernel estimate. This scaling relationship is illustrated in Figure 2.6 for the Volterra model of the fly photoreceptor, and it is only approximate for this real sensory system (under certain light-adapted conditions).

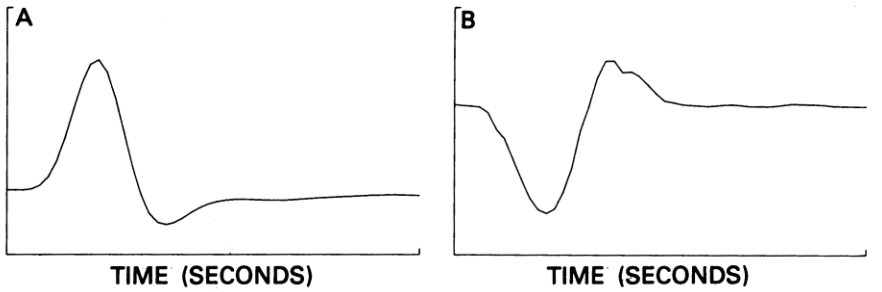


Figure 2.6
Comparison of the first-order kernel (A) with a "slice" of the second-order kernel (B) for the fly photoreceptor (Marmarelis & McCann, 1977).

Example 2.3

L-N-M "sandwich" system

Another modular cascade model that has been developed in the study of the visual system is the so-called "sandwich" model comprised of two linear filters L and M separated by a static nonlinearity N (see Figure 2.7). This cascade model has been extensively studied analytically and experimentally [Korenberg, 1973a; Marmarelis & Marmarelis, 1978; Korenberg & Hunter, 1986]. Its Volterra kernels can be derived by combining the results of the two previous examples and are given by (see Section 4.1 for detailed derivation):

$$k_r(\tau_1,...,\tau_r) = \alpha_r \int_0^{\min(\tau_1,...,\tau_r)} h(\lambda)\, g(\tau_1-\lambda)...g(\tau_r-\lambda)\, d\lambda \qquad (2.14)$$

where $g(\tau)$ and $h(\tau)$ are the impulse response functions of the prior filter L and the posterior filter M respectively and $\{a_r\}$ are the polynomial coefficients of the static nonlinearity N (as before).

We observe that the simple scaling relation established in the L-N cascade between "cuts" of Volterra kernels of various orders (at arbitrary values of τ_i) does not hold for the L-N-M cascade. However, in this case, the kernel values along any axes are proportional to each other (as long as they have the same dimensionality), because of the causality of the filters L and M [Chen et al. 1985]. For instance, if $\tau_r = 0$, then $g(\tau_r - \lambda) \neq 0$ only for $\lambda = 0$, and therefore:

$$k_r(\tau_1,...,\tau_{r-1},0) = \alpha_r h(0) g(0) g(\tau_1)...g(\tau_{r-1}) \tag{2.15}$$

which is proportional to: $k_{r+j}(\tau_1,...,\tau_{r-1},0,...,0) = \alpha_{r+j} h(0)\left[g(0)\right]^{j+1} g(\tau_1)...g(\tau_{r-1})$ for any j. In practice, this test is not easily applied to ascertain the possible suitability of the L-N-M cascade model, because it requires at least a third-order kernel estimate (i.e., it cannot be applied between the first-order and the second-order kernels that are typically estimated in practice). Therefore, an alternative test has been developed for this purpose, as described in Section 4.1 along with further elaboration on the analysis of the L-N-M cascade model.

One of the initial applications of the L-N-M cascade model was in the vertebrate retina, where it was found suitable for modeling the amacrine cells (see Section 6.1.1).

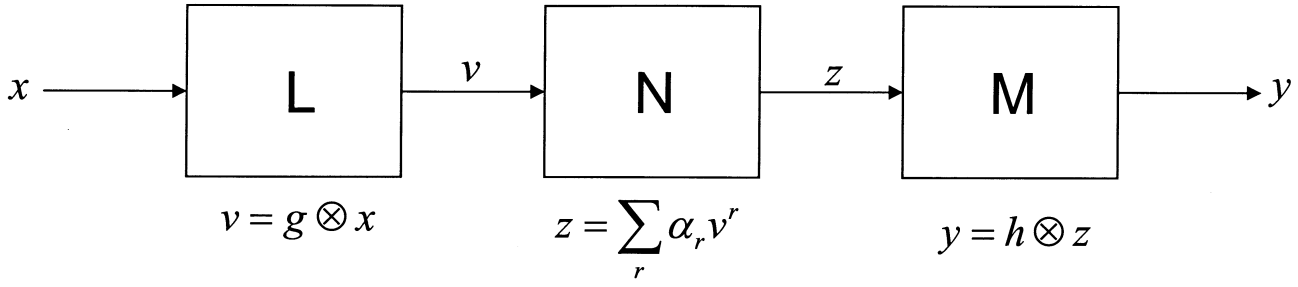


Figure 2.7
The $L-N-M$ cascade (sandwich) model, comprised of two linear filters (L and M) and a static nonlinearity (N) "sandwiched" between them ($\otimes$ denotes convolution).

Example 2.4

Riccati system

Consider now the simple parametric model presented as Example 1.3 in Section 1.4, where the relation between the input $x(t)$ and the output *y(t)* can be described by the first-order nonlinear differential equation:

$$\frac{dy}{dt} + ay + by^2 = cx \tag{2.16}$$

The homogeneous solution of this equation (i.e., in the absence of any input x) is the Bernoulli equation that yields upon integration the sigmoidal "logistic curve" (for non-zero initial conditions) encountered frequently in biology to describe saturation of growth processes. This equation, has been used also by Verhulst to describe the population dynamics of a species with internal competition.

With a forcing function on the right-hand side (represented by the input x), Equation (2.16) becomes the Riccati equation and has been used to describe nonlinear kinetics, where the kinetic constant depends linearly on the output (i.e., it is equal to: $a + by$). It constitutes a parametric model with three parameters *(a,b,c)* of a system that is nonlinear (because of the y^2 term), stationary (because all coefficients are time-invariant) and dynamic (because of the derivative term).

The equivalent nonparametric Volterra model can be obtained by use of the "generalized harmonic balance method" presented in Section 3.2. An infinite order Volterra model (Volterra series) is required for complete representation of this system. The first two Volterra kernels are derived to be:

$$k_1(\tau) = ce^{-a\tau}u(\tau) \tag{2.17}$$

$$k_2(\tau_1, \tau_2) = -\frac{bc^2}{a} e^{-a(\tau_1+\tau_2)} \left[1 - e^{a \cdot \min(\tau_1, \tau_2)}\right] u(\tau_1) u(\tau_2) \tag{2.18}$$

where $u(\tau)$ denotes the step function (i.e., zero for $\tau < 0$ and unity elsewhere). The higher order Volterra kernels have more complicated expressions and are omitted in the interest of space. Note, however, that the r-th order kernel is proportional to b^{r-1} and, thus, terms higher than second order are negligible if $|b|$ is very small.

2.1.2. Operational Meaning of the Volterra Kernels

In this section, we seek to provide an operational meaning to the Volterra kernels, so that they do not remain abstract mathematical objects but become useful instruments for enhancing our understanding of the functional properties of physiological systems.

Let us begin by pointing out that the zeroth-order kernel is a simple reference constant (for stationary systems) representing the output value when no input is applied to the system. The 1st-order Volterra functional is the well-known convolution integral that represents the input-output relation of linear time-invariant (LTI) systems. In this convolution, the 1st-order kernel plays the role of the "impulse response function" for LTI systems, which is the system response (output) to an impulsive input for LTI systems but not for nonlinear systems. As will be seen below, the response of a nonlinear system to an impulsive input involves additionally the diagonal values of all higher order kernels. The 1st-order kernel represents the pattern by which the systems weight the input epoch values (past and present) to generate the present value of the system output through linear superposition (weighted summation in discrete time or integration in continuous time). An illustration of this is presented in Figure 2.2.

The nonlinear behavior is represented by the multiple convolution integrals of the Volterra functionals of order second and higher. The r^{th}-order Volterra functional:

$$V_r\left[x(t)\right] = \int_0^\infty \ldots \int k_r(\tau_1, \ldots, \tau_r)\, x(t-\tau_1) \ldots x(t-\tau_r)\, d\tau_1 \ldots d\tau_r \tag{2.19}$$

is an r-tuple convolution of r time-shifted versions of the input signal with an r-dimensional function k_r, termed the r^{th}-order Volterra kernel. The latter describes the weighting pattern of r^{th}-order nonlinear interactions that the system uses in order to generate the present value of the system output through integration of all product combinations of r input epoch values.

An illustration of this is given in Figure 2.3 for the second-order kernel. Each value of this kernel (depicted as a surface), for instance $k_2(\lambda_1, \lambda_2)$ at lags λ_1 and λ_2, represents the weight (i.e., the relative importance) of the product $x(t-\lambda_1)x(t-\lambda_2)$ in constructing the system output $y(t)$. All such weighted product combinations are integrated in order to construct the second order nonlinear component of the system output described by the second-order Volterra functional. A large positive value of $k_2(\lambda_1, \lambda_2)$ implies strong mutual facilitation of the input lagged values $x(t-\lambda_1)$ and $x(t-\lambda_2)$ in the way they affect the system output $y(t)$. Note that this often occurs at the diagonal points $(\lambda_1 = \lambda_2)$, as in the case of the retinal ganglion cell whose second-order kernel peaks around $\lambda_1 = \lambda_2 = 35\text{ms}$ as shown in Figure

1.5. Conversely, a negative value of $k_2(\lambda_1, \lambda_2)$ implies mutual inhibition between the input lagged values $x(t-\lambda_1)$ and $x(t-\lambda_2)$ in the way they affect the system output $y(t)$-- see, for instance, the negative "trough" along the diagonal of the 2nd-order kernel of the ganglion cell between a lag of 80 ms and 160 ms (shown in Figure 1.5). Small kernel values imply that the corresponding combinations of input lagged values do not affect significantly the present output value.

It is critical to note that the Volterra kernel values are fixed (for a given stationary system) and represent characteristic system *signatures* (i.e., they are unique in the Volterra context). Therefore, they can be used for unambiguous classification of nonlinear physiological systems and hold great promise for *improved diagnostic purposes*. The Volterra kernels of a given system are also complete descriptors of the system nonlinear dynamics of the corresponding order (i.e., for 2nd-order interactions, there is nothing left out of the 2nd-order Volterra kernel pertaining to the manner in which pairs of input lagged values affect the output of the system). Therefore, the Volterra kernels contain complete and reliable information regarding the system function at the respective order of nonlinear interactions (nothing missing, nothing spurious) and offer the ultimate tools for proper understanding of physiological function—upon successful interpretation.

Thus, the Volterra kernels of the system form a hierarchy of system nonlinearities (according to multiplicity rank of input interactions) and constitute a canonical representation of the system nonlinear dynamics. They are the complete and reliable descriptors of the system function (i.e., allow accurate prediction of the system output for *any* given input). Their estimation from input-output data is the objective of the system identification task in the nonparametric context. Methodologies to this purpose are discussed in Sections 2.1.5 and 2.3.

The elegant and insightful hierarchical organization of the Volterra functionals is clearly depicted in Example 2.2, where the r^{th}-degree term of the polynomial static nonlinearity gives rise to the r^{th}-order Volterra functional (involving the associated r^{th}-order Volterra kernel). An instructive look into the operational meaning of the r^{th}-order Volterra functional (and kernel) is provided by the use of sinusoidal and impulsive inputs, as discussed below.

Impulsive Inputs

For an impulsive input $x(t) = A\delta(t)$, the output of the Volterra model is:

$$y(t) = k_0 + Ak_1(t) + A^2k_2(t,t) + ... + A^rk_r(t,...,t) + ... \tag{2.20}$$

i.e., it involves all the main diagonal values of all kernels in the system and forms a power series (or polynomial, if the model is finite) in terms of the amplitude A of the impulsive input. This fact draws the clear distinction between the impulse response of a nonlinear system (which contains the main diagonal values of all kernels) and the impulse response of a linear system (which corresponds to the 1st-order Volterra kernel). Thus, we must avoid the use of the term "impulse response function" to denote the 1st-order kernel of a nonlinear system, since this is clearly misleading.

Let us now consider a pair of impulses in the input: $x(t) = A\delta(t) + B\delta(t - t_0)$. Then the Volterra model output is:

$$\begin{aligned} y(t) &= k_0 + Ak_1(t) + Bk_1(t - t_0) \\ &+ A^2k_2(t,t) + B^2k_2(t - t_0, t - t_0) + 2ABk_2(t, t - t_0) \\ &+ A^3k_3(t,t,t) + B^3k_3(t - t_0, t - t_0, t - t_0) + 3A^2Bk_3(t,t,t - t_0) + 3AB^2k_3(t, t - t_0, t - t_0) \\ &+ ... \end{aligned} \tag{2.21}$$

where the first three terms are the 1st-order ("linear") component of the output, the following three terms are the 2nd-order component of the output, the following four terms are the 3rd-order component of the output, etc. Note that the expression of Equation (2.21) is condensed by using the fact that the Volterra kernels are symmetric about the diagonals (i.e., invariant for any permutation of their arguments).

Clearly, for the pair-of-impulses input, the high-order kernels generate interaction terms in the output (e.g., the term $2ABk_2(t, t - t_0)$ for second-order interaction) that represent the nonlinear "cross-talk" between the two impulses in generating the output. This is the effect of "*dynamic nonlinearity*" that is manifested in the output as interactions of input values at different (lagged) points in time (e.g., at time-points t and $(t - t_0)$ in this example). This effect of "dynamic nonlinearity" can be contrasted to the "*amplitude nonlinearity*" effect manifested in this example by the terms involving only the main diagonal values of the kernels and the powers of only A or only B. Note that both effects spread over time in a manner determined by the kernel values, as illustrated in Figure 2.1.

It is evident that if the output components due to each impulse (applied separately) get subtracted from the pair-of-impulses output (the test for the superposition principle), then the residual corresponds strictly to the "dynamic nonlinearity" terms and depends on the off-diagonal values of the kernels. This

is further elaborated in Section 2.1.5, in connection with the issue of Volterra kernel estimation using multiple impulses as experimental test inputs.

It is also evident from Equation (2.21) that "amplitude nonlinearities" cannot be detected with the pair-of-impulses test, unless the amplitude of the impulses is varied. The same is true for any sequence of impulses (including random or pseudorandom sequences) that maintain fixed magnitude. This implies that the main diagonal values of the kernels cannot be estimated by use of impulses with fixed magnitude (although the remaining kernel values can be estimated), a fact that has important implications in modeling studies of neuronal systems with spike-train inputs (as elaborated in Chapter 8). This limitation can be overcome, of course, if the impulses are allowed to vary in amplitude or a judicious form of interpolation can be used.

This type of analysis can be extended to three or more impulses and can yield Volterra kernel estimates for finite-order systems, as discussed for a limited experimental context in Section 2.1.5, or be extended to arbitrary impulse sequences in a general methodological context germane to neuronal systems, as discussed in Chapter 8. Due to its relative simplicity, the study of the system response to impulsive stimuli can provide useful preliminary information about the system memory and dynamics, as discussed in Section 5.2.

Sinusoidal Inputs

For a sinusoidal input of frequency ω_0, the r^{th}-order Volterra functional V_r generates an r^{th} harmonic (i.e., a sinusoidal signal of frequency $(r \cdot \omega_0)$) and lower harmonics of the same parity (i.e., odd or even), namely the harmonics $r,(r-2),(r-4),...,(r-2q)$, where q is the integer part of $r/2$, as can be shown by use of trigonometric formulae [Bedrosian & Rice, 1971]. The amplitude and phase of these harmonics is determined by the values of the r^{th}-order Volterra kernel k_r. This is demonstrated below for the 2nd-order case.

For a sinusoidal input $x(t) = \cos \omega_0 t$, the 2nd-order Volterra functional is:

$$V_2\left[x(t)\right] = \int_0^\infty \int k_2(\tau_1, \tau_2) \cos \omega_0 (t - \tau_1) \cos \omega_0 (t - \tau_2)\, d\tau_1 d\tau_2$$

$$= \frac{1}{2} \int_0^\infty \int k_2(\tau_1, \tau_2) \cos \omega_0 (\tau_1 - \tau_2)\, d\tau_1 d\tau_2 + \frac{1}{2} \int_0^\infty \int k_2(\tau_1, \tau_2) \cos \omega_0 (2t - \tau_1 - \tau_2)\, d\tau_1 d\tau_2 \qquad (2.22)$$

Clearly the first term is constant over time (zeroth harmonic) and the second term yields the second harmonic:

$$\frac{1}{2}\cos 2\omega_0 t\iint k_2(\tau_1,\tau_2)\cos\omega_0(\tau_1+\tau_2)\,d\tau_1 d\tau_2+\frac{1}{2}\sin 2\omega_0 t\iint k_2(\tau_1,\tau_2)\sin\omega_0(\tau_1+\tau_2)\,d\tau_1 d\tau_2$$

$$=\pi^2\,\mathrm{Re}\left\{K_2(\omega_0,\omega_0)\right\}\cos 2\omega_0 t+\pi^2\,\mathrm{Im}\left\{K_2(\omega_0,\omega_0)\right\}\sin 2\omega_0 t \qquad (2.23)$$

where $K_2(\omega_1,\omega_2)$ is the two-dimensional Fourier transform of the 2nd-order Volterra kernel. It is evident from Equation (2.23) that the amplitude and the phase of the second harmonic depend on the value of the two-dimensional Fourier transform (2D-FT) of the 2nd-order Volterra kernel at the bi-frequency (ω_0,ω_0). It can be further shown that, if two sinusoidal frequencies are used at the input: $x_2(t)=\cos\omega_1 t+\cos\omega_2 t \quad (\omega_1,\omega_2)$, then the 2nd-order Volterra functional will generate sinusoidal components at frequencies $(2\omega_1)$, $(2\omega_2)$, $(\omega_1+\omega_2)$, $(\omega_1-\omega_2)$, in addition to constant terms; indicating that the complex values of $K_2(\omega_i\pm\omega_j)$, where $i,j=1,2$, determine the 2nd-order response of the system. This result can be generalized for any number M of sinusoids in the input signal by letting the indices i and j take all integer values from 1 to M. Specifically:

$$V_2\left[x_2(t)\right]=const.+\pi^2\sum_{i=1}^{M}\sum_{j=1}^{M}\mathrm{Re}\left\{K_2(\omega_i\pm\omega_j)\right\}\cos(\omega_i\pm\omega_j)t+\mathrm{Im}\left\{K_2(\omega_i\pm\omega_j)\right\}\sin(\omega_i\pm\omega_j)t \qquad (2.24)$$

This expression (2.24) governs the 2nd-order response of any Volterra system to an arbitrary input waveform expressed in terms of its Fourier decomposition. Thus, 2nd-order nonlinear interactions in the frequency domain (intermodulation effects) involve all possible pair combinations $(\omega_i\pm\omega_j)$ of sinusoidal components of the input signal weighted by the values of the 2D-FT of the 2nd-order kernel at the respective bi-frequency points $(\omega_i\pm\omega_j)$.

Following the same line of analysis, we can show that the r th-order Volterra functional generates output components at all frequencies $(\omega_{i_1}\pm\omega_{j_2}\pm\ldots\pm\omega_{i_r})$ weighted by the respective values of $K_r(\omega_{i_1}\pm\omega_{j_2}\pm\ldots\pm\omega_{i_r})$, where the indices i_1 through i_r take all integer values from 1 to M. The frequency response characteristics of the Volterra functionals are discussed more broadly in the following section dealing with frequency-domain representations of the Volterra models.

Remarks on the Meaning of Volterra Kernels

From the foregoing discussion, it is evident that the Volterra kernels (of any order) can be viewed as the multi-dimensional weighing patterns by which the system weighs all product combinations of input lagged values or sum/difference combinations of multi-sinusoid input frequencies in order to produce the system output through weighted integration (or summation, in discrete time). These patterns of nonlinear interactions among different values of the input signal (as they are encapsulated by the system kernels) allow prediction of the system output to *any* given input and constitute a complete representation of the system functional properties, as well as characteristic "signatures" of the system function. As such, they can be used to simulate and analyze the functional properties of the system, as well as to characterize it for classification or diagnostic purposes.

The far-reaching implications for physiology and medicine are evident (physiological understanding, hypothesis testing, clinical diagnosis and monitoring, closed-loop treatment, therapy assessment, design of prosthetics and implants, tissue characterization, physiological control and regulation, etc.), if we can only harness this modeling power in an experimental and clinical context [Brillinger, 1970; Bedrosian & Rice, 1971].

It should be emphasized that the Volterra kernels representation is <u>not</u> an *ad hoc* scheme based on intuition or serendipitous inspiration but a complete, rigorous, canonical representation of the system functional properties that possesses the requisite credibility and reliability for critical, life-affecting applications.

2.1.3. Frequency-Domain Representation of the Volterra Models

The useful insight gained by frequency-domain analysis provide the motivation for studying the Volterra models in the frequency domain. This is accomplished with the use of multi-dimensional Fourier transforms for the high-order kernels. It has been found that the Volterra series can be expressed in the frequency domain as [Brillinger, 1970; Rugh, 1981]:

$$Y(\omega) = 2\pi k_0 \delta(\omega) + K_1(\omega) X(\omega) + \frac{1}{2\pi} \int_{-\infty}^{\infty} K_2(\omega, \omega - u) X(\omega) X(\omega - u) du + ...$$

$$... + \frac{1}{(2\pi)^{r-1}} \iint \int_{-\infty}^{\infty} K_r(u_1, ..., u_{r-1}, \omega - u_1 ... - u_{r-1}) X(u_1) ... X(u_{r-1}) ... X(\omega - u_1 ... - u_{r-1}) du_1 ... du_{r-1} + ... \quad (2.25)$$

for deterministic inputs and kernels that have proper Fourier transforms. The latter are guaranteed because the kernels must satisfy the absolute integrability condition (Dirichlet condition) for purposes of

Volterra series convergence and system stability (asymptotic finite-memory requirement), as indicated by Equation (2.4). Although certain input signals may not have proper Fourier transforms (e.g., stationary random signals such as white noise), the use of finite data-records in practice makes this mathematical issue moot. Note that ω and u_i denote frequency in rad/sec giving rise to the powers of (2π) scaling terms in Equation (2.25). If frequency is measured in Hz, then these scaling factors are eliminated.

Equation (2.25) indicates that for a generalized sinusoidal input $x(t) = Ae^{j\omega_0 t}$, the r^{th}-order Volterra functional generates at the system output the r^{th} harmonic:

$$Y_r(\omega) = 2\pi A^r K_r(\omega_0, \omega_0, ..., \omega_0, \omega_0)\delta(\omega - r\omega_0) \tag{2.26}$$

since $X(u_i) = 2\pi A\delta(u_i - \omega_0)$ in this case. Note that no lower harmonics are generated here because of the complex analytic form (phasor) of the generalized sinusoidal input that simplifies the mathematical expressions. However, in practice, the input is not complex analytic and the resulting output components include lower harmonics of the same parity (odd or even). For instance, the 5^{th}-order Volterra functional will give rise to a first, third and fifth harmonic. This odd/even separation of the Volterra functionals can be used in practice to gain additional insight into the possible odd/even symmetries of the system nonlinearity.

If we consider an input comprised of a pair of complex analytic sinusoids: $x_2(t) = Ae^{j\omega_1 t} + Be^{j\omega_2 t}$, then: $X_2(\omega) = 2\pi\left[A\delta(\omega - \omega_1) + B\delta(\omega - \omega_2)\right]$, and intermodulation terms are generated by the Volterra functionals due to nonlinear interactions. For instance, the 2^{nd}-order functional contributes the following three terms to the system output in the frequency domain:

$$Y_2(\omega) = 2\pi A^2 K_2(\omega_1, \omega_1)\delta(\omega - 2\omega_1) + 2\pi B^2 K_2(\omega_2, \omega_2)\delta(\omega - 2\omega_2) + 4\pi AB K_2(\omega_1, \omega_2)\delta(\omega - \omega_1 - \omega_2) \tag{2.27}$$

that represent second harmonics at frequencies $(2\omega_1)$ and $(2\omega_2)$, as well as an intermodulation term at frequency $(\omega_1 + \omega_2)$. In the time domain, the second-order Volterra functional for this input is:

$$V_2\left[x_2(t)\right] = A^2 K_2(\omega_1, \omega_1)e^{j2\omega_1 t} + B^2 K_2(\omega_2, \omega_2)e^{j2\omega_2 t} + 2AB K_2(\omega_1, \omega_2)e^{j(\omega_1 + \omega_2)t} \tag{2.28}$$

The resulting second-order output component has three generalized sinusoidal terms at the frequencies $(2\omega_1)$, $(2\omega_2)$, $(\omega_1 + \omega_2)$ with amplitudes and phases determined by the values of K_2 at the respective frequencies, as illustrated in Figure 2.8 for a second-order kernel from renal autoregulation.

The expressions for inputs with multiple sinusoids and higher-order functionals are, of course, more complicated. For an input with M complex analytic sinusoids:

$$x_M(t) = A_1 e^{j\omega_1 t} + \ldots + A_M e^{j\omega_M t} \tag{2.29}$$

the r th order Volterra functional contributes in the frequency domain the r th-order output component:

$$Y_r(\omega) = 2\pi \sum_{m_1=1}^{M} \ldots \sum_{m_r=1}^{M} A_{m_1} \ldots A_{m_r} K_r\left(\omega_{m_1}, \ldots, \omega_{m_r}\right) \delta\left(\omega - \omega_{m_1} - \ldots - \omega_{m_r}\right) \tag{2.30}$$

which yields in the time domain complex analytic sinusoidal components at all possible r^M sums of r frequencies (with repetitions) from the M frequencies present in the input signal:

$$V_r\left[x_M(t)\right] = \sum_{m_1=1}^{M} \ldots \sum_{m_r=1}^{M} A_{m_1} \ldots A_{m_r} K\left(\omega_{m_1}, \ldots, \omega_{m_r}\right) e^{j\left(\omega_{m_1} + \ldots + \omega_{m_r}\right)t} \tag{2.31}$$

The main point is that the harmonics and intermodulation terms generated by the system nonlinearities can be predicted explicitly and accounted quantitatively by the system kernels.

Equation (2.25) also shows that a broadband input may lead to an output with even broader bandwidth, depending on the spectral characteristics of the kernels. For instance, a static nonlinearity will generally lead to broadening of the output bandwidth (relative to the input bandwidth) but this may not happen for certain types of kernels. Another interesting possibility raised by Equation (2.25) is that the various orders of kernels of the same system may have different bandwidths and, therefore, different nonlinearities may be activated by higher or lower input frequencies.

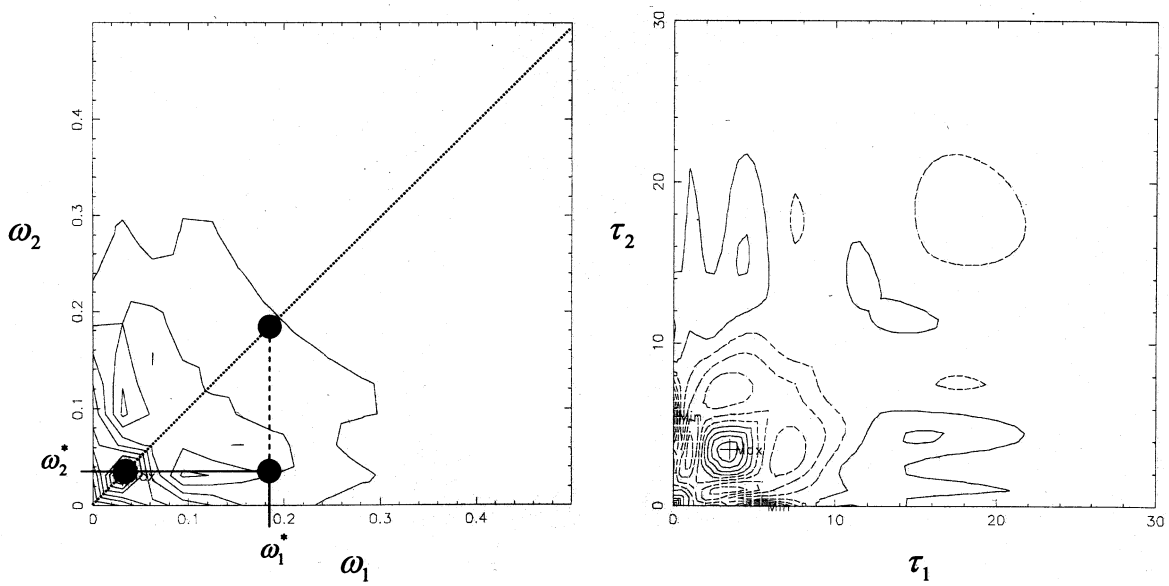


Figure 2.8

The magnitude (left) of the 2-D Fourier transform of the second-order kernel shown in the right panel (from renal autoregulation). For given pair of stimulation frequencies $\left(\omega_1^*, \omega_2^*\right)$, the resulting second-order output component has three terms at frequencies $\left(2\omega_1^*\right), \left(2\omega_2^*\right), \left(\omega_1^* + \omega_2^*\right)$ as indicated in Equation (2.28). The corresponding magnitudes depend on the values of $K_2\left(\omega_1^*, \omega_1^*\right), K_2\left(\omega_2^*, \omega_2^*\right), K_2\left(\omega_1^*, \omega_2^*\right)$ as marked in the figure with solid circles.

2.1.4. Discrete-Time Volterra Models

In actual applications, the input-output signals are sampled at a fixed sampling rate that must exceed the bandwidths of both signals (Nyquist frequency). These sampled data constitute discrete-time signals, also referred to as time-series data. These discrete-time signals are used to model the system in practice and to estimate the requisite Volterra models in discrete form, as discussed in Sections 2.1.5 and 2.3. This gives rise to the discrete-time Volterra model of the form:

$$y(n) = k_0 + T\sum_{m} k_1(m)\,x(n-m) + T^2\sum_{m_1}\sum_{m_2} k_2(m_1,m_2)\,x(n-m_1)\,x(n-m_2) + \ldots \qquad (2.32)$$

where n represents the discrete-time index $(n = t/T)$, m denotes the discrete-time lag $(m = \tau/T)$, and T is the sampling interval. The discretized kernels $k_1(m)_1\, k_2(m_1,m_2),\ldots$ are sampled versions of the true continuous-time Volterra kernels of the system. Thus, the discretization of the Volterra model is straightforward as long as T is sufficiently small relative to the bandwidth B_s of the system (i.e., $T \le 1/(2B_s)$). A note of caution is in order, regarding the selection of T, when the bandwidth of the input signal is narrower than the system bandwidth. The reader should be alerted that kernel estimation is not possible in this case (without aliasing) if T is selected on the basis of the input Nyquist frequency. Therefore, T must be selected according to the maximum bandwidth of the system *and* of the input-output signals.

It is evident from Equation (2.32) that the discrete-time Volterra series expansion depends on the sampling interval T. Since the latter is not usually incorporated into the discrete Volterra model used for kernel estimation, the resulting discrete kernel estimates are scaled by a power of T equal to the order of the kernel. In order to remove the dependence of the estimated discrete kernel values on T, we must either include T explicitly in the discrete Volterra model as shown in Equation (2.32) or divide the obtained kernel values by T^r where r is the order of the kernel. We adopt the latter convention, which is also important in order to maintain the proper physical units for the estimated kernel values. Under this convention, the physical units for the rth-order kernel remain what they ought to be: (output units)/[(input units)$^r \times$(time units)^r]. Note, however, that under this convention the estimated values of the discrete Volterra kernels must be divided by T^r, in order to retain their physical units

An important practical attribute of the discrete Volterra kernels is the number of samples (lags) along each dimension of the kernel that are required in order to represent properly the kernel in discrete time. This number M is determined as the ratio of the effective kernel memory μ (i.e., the domain of τ over

which the kernel has significant values) to the sampling interval T. Since the latter may not exceed the inverse of twice the system bandwidth B_s (in Hz) to avoid aliasing, we conclude that the minimum M is equal to twice the *memory-bandwidth product* of the system. In general:

$$M \geq 2B_s\mu \tag{2.33}$$

where μ is measured in sec and B_s in Hz. The parameter M is critical in practice because it determines the number of unknowns that need be estimated when the kernel is represented by its discrete-time samples. This number increases geometrically with the order r of the kernel (i.e., as M^r). Therefore, high-order kernels with large memory-bandwidth product are difficult to estimate. This problem is mitigated by the introduction of appropriate kernel expansions that reduce the number of unknowns to be estimated, as discussed in Section 2.3.

The equivalence between discrete Volterra models (kernels) and discretized parametric models of differential equations (i.e., nonlinear difference equations) is discussed in Section 3.3. In general, discrete Volterra kernels of difference-equation models can be derived analytically using the "generalized harmonic balance" method. This provides a methodological bridge between parametric and nonparametric models in discrete time. However, this methodology tends to be rather cumbersome in the general case and may prove practical only in a limited number of applications—an issue that deserves further investigation, since it has received only a rudimentary treatment to date.

The frequency-domain analysis of the discrete Volterra models/kernels employs discrete Fourier transforms (DFT) or its computationally efficient version, the fast Fourier transform (FFT). The DFT or FFT of the discrete Volterra kernels are approximations of the kernel Fourier transforms discussed in Section 2.1.3, within the frequency resolution defined by the fundamental frequency of the DFT/FFT (i.e., the inverse of the kernel length). The minimum frequency resolution is established by the kernel memory, but higher resolution can be had by increasing the length of the kernel estimate with zero-packing (i.e., decreasing the fundamental frequency of the DFT/FFT), if so desired. A parsimonious approach is always recommended, due to the dimensionality of high-order kernels (e.g., a doubling of frequency resolution quadruples the number of points in the 2D-FFT of the 2^{nd}-order kernel).

For analytical manipulations in discrete time, the z-transform can be used as indicated in Sections 3.3-3.5. Note that the primary utility of the z-transform is in solving analytically difference equations.

2.1.5. Estimation of Volterra Kernels

From the point of view of system modeling, the critical issue is the practical and accurate estimation of the discrete-time Volterra kernels from input-output experimental data. It is evident from Equation (2.32) that the unknown kernel sampled values enter linearly in this estimation problem (although the model is nonlinear in terms of the input-output relation), thus facilitating the solution of the set of linear equations written in matrix form as:

$$\mathbf{y} = \mathbf{X}\mathbf{k} + \boldsymbol{\varepsilon} \tag{2.34}$$

where $\mathbf{y}' = \left[y(1), y(2), ..., y(N) \right]$ is the output data vector (with prime denoting "transpose"), $\boldsymbol{k}' = \left[k_0, Tk_1(0), Tk_1(1), ..., Tk_1(M), T^2K_2(0,0), 2T^2k_2(1,0), T^2k_2(1,1), 2T^2k_2(2,0), 2T^2k_2(2,1), T^2k_2(2,2), ..., 2T^2k_2(M, M-1), T^2k_2(M,M) \right]$ is the vector of unknown kernel values (to be estimated) for a 2nd-order discrete-time Volterra model with memory-bandwidth product $M(\mu = M \cdot T)$, $\mathbf{X}$ is the input data matrix constructed according to Equation (2.32) using the above definition of the kernel vector $\boldsymbol{k}$ that takes into account the kernel symmetries, and $\boldsymbol{\varepsilon}$ is the error vector $\left[\varepsilon(1), \varepsilon(2), ..., \varepsilon(N) \right]$ defined from Equation (2.32) for each discrete time n as the difference between the model-predicted and the measured output value (the error terms are also called "residuals"). Note that the estimated discrete kernel values are scaled by constants lT^r dependent on the specific kernel order r and point location, where l accounts for the intrinsic kernel symmetries (e.g., $l = r!$ for all off-diagonal points but $l = 1$ for full-diagonal points). As discussed in the previous section, the estimated discrete kernel values depend on the sampling interval T, which is determined by the system bandwidth and the sampling requirements of the input-output signals. The size of P of the unknown kernel vector $\boldsymbol{k}$ (which determines the computational burden of this estimation problem) depends on the system memory-bandwidth product M and on the nonlinear order of the system $Q = \max\{r\}$, roughly increasing as M^Q.

There are two confounding issues in practice. The first issue concerns the rapid increase of discrete kernel values that need be estimated as the model order and/or the system memory-bandwidth product increase. For a kernel with M discrete sampled values in each dimension (representing a fixed characteristic of the system determined by the product of the system bandwidth with the system memory) the number of estimated discrete values for the rth-order kernel is: $\left[M(M+1)...(M+r-1) \right] / r!$, when the kernel symmetries are taken into account. By summing all kernel orders from 0 to Q, we find that the total number of discrete kernel values for a Qth-order system is:

$$P = (M+Q)(M+Q-1)...(M+1)/Q! \tag{2.35}$$

For $Q \gg M$, this number is approximately $M^Q/Q!$, indicating a geometric increase with Q and an exponential dependence of the total number of estimated kernel values (free parameters) on the model order and the log of M.

This "curse of dimensionality" (as it is often lamented by investigators confronted with this problem) represents the most serious limitation in the practical application of the Volterra modeling approach to nonlinear systems of high order and/or large memory-bandwidth product (MBP). This limitation has motivated the introduction of kernel expansions (see Section 2.3) which mitigate the effects of the dimensionality problem by compacting the kernel representation for systems with large MBP. However, the kernel expansion still faces limitations for high-order systems. This problem can be effectively addressed only by use of equivalent *structured* models that constrain the number of free parameters, such as the network models discussed in Section 2.3.3. The latter represent our best answer to this problem at the present time.

The second confounding issue arises from the practical necessity of selecting a finite-order Volterra model, even if the system is actually of infinite order. This implies that there exists some correlation among the residuals of the estimation (fitting) procedure due to the model truncation errors that depend on the input signal. This correlation of the residuals leads to biases in the kernel estimates obtained via least-squares estimation procedures. Of course, this estimation bias becomes significant only when the model truncation error is significant. Therefore, the severity of this problem depends on the specific application and becomes serious only for high-order systems that are not represented with satisfactory model order approximation.

Note that the adequacy of the model order approximation also depends on the dynamic range (or power level) of the input signals used or anticipated in each application. Naturally, the greater the dynamic range (or power level) of the input signal, the greater the relative importance of the higher order kernels. Furthermore, the form of the resulting estimation bias depends on the spectral characteristics of the particular input data used for kernel estimation. Most applications to date have been limited to the estimation of up to 2nd-order kernels and are liable to this "model truncation" problem (i.e., the possible presence of significant higher order kernels will cause estimation biases in the obtained 1st-order and 2nd-order kernels). However, the advocated use of equivalent high-order structured (network) models alleviates the model truncation problem by allowing estimation of high-order nonlinearities with small computational cost (see Section 2.3.3).

We describe below several methodologies that have been used thus far for the estimation of the Volterra kernels. These methodologies can be clustered in two groups: one is employing specialized experimental inputs (e.g., multiple impulses or sums of sinusoids) and the other is applicable for arbitrary input signals. These methods are not recommended for efficacy but are presented for completeness of methodological background

Specialized Test Inputs

Since impulsive and sinusoidal inputs have been used extensively in the study of linear systems, it was natural that early attempts to estimate Volterra kernels employed similar test input waveforms. In order to account for multiple time or frequency interactions, these inputs took the form of sequences of impulses (with variable inter-impulse intervals) and sums of sinusoids (with incommensurate frequencies), as discussed below.

In the case of impulse sequences, the order Q of the Volterra model has to be determined first from single-impulse experiments using the variable impulse strength input $x_A(t) = A\delta(t)$ that elicits the output [Schetzen, 1965a]:

$$y_A(t) = Ak_1(t) + A^2k_2(t,t) + ... + A^Qk_Q(t,...,t) \tag{2.36}$$

The model order Q is determined by finding the polynomial dependence of y_A on Q for some t (usually near the peak output value).

The diagonal values of the kernels can be estimated, along with $k_1(t)$, from single-impulse elicited output data by solving a system of linear simultaneous equations given by Equation (2.36) for various A values. The number of A values ought to be at least Q times the memory-bandwidth product of the system to have a critically determined set of equations, although a larger number of A values is welcome since it offers noise-suppressing possibilities through an overdetermined set of equations.

In order to estimate the off-diagonal values of the kernels, a sequence of two, three,…, up to Q impulses of variable timing and amplitude is presented to the system in a manner that covers all timing and amplitude combinations of interest. Upon completion of this long sequence of such experiments, the various kernels are estimated in ascending order starting with the second-order kernel through the proper subtraction of multiple experimental outputs, as demonstrated below.

For a second-order system, a two-impulse sequence:

$$x_{AB}(t) = A\delta(t) + B\delta(t - t_0) \tag{2.37}$$

is used as experimental input to produce the output (omitting k_0 for simplicity):

$$y_{AB}(t) = Ak_1(t) + Bk_1(t-t_0) + A^2k_2(t,t) + B^2k_2(t-t_0,t-t_0) + 2ABk_2(t,t-t_0) \tag{2.38}$$

for all values of t_0 from T (the sampling interval) to μ (the system memory). Note that the second-order kernel is symmetric about its diagonal (i.e., $k_2(t-t_0,t) = k_2(t,t-t_0)$). Although only single values of A and B are required in theory, several values may be used to cover the input amplitude range of interest (in order to secure the global validity of the model) and improve its accuracy in the presence of inevitable noise by affording some averaging. The values A and B can be randomly selected according to a prescribed probability distribution representing our understanding of the likelihood of input amplitudes occurring under the real operating conditions of the system. Then, it is evident that:

$$y_{AB}(t) - y_A(t) - y_B(t-t_0) = 2ABk_2(t,t-t_0) \tag{2.39}$$

which yields an estimate of a slice $k_2(t,t-t_0)$ of the second-order kernel parallel to the diagonal for given values of A and B (which can be divided out). Using different values of A and B, we can obtain multiple estimates of the same para-diagonal slice of the second-order kernel and suppress inevitable errors (from measurements and noise) through averaging. By varying t_0, we cover the entire second-order kernel slice by slice (up to the system memory, for $t_0 = \mu$) except for its diagonal.

The apparent simplicity of this approach conceals the onerous experimentation required, especially for high-order systems, and its vulnerability to noise. Nonetheless, it may represent an attractive option in certain cases. A variant of this procedure can be used in the study of neural systems with spike inputs (action potentials), as discussed in Chapter 8.

Another specialized test input that was introduced in the early years of this approach is the sum of sinusoids of incommensurate frequencies [Victor et al. 1977; Victor, 1979]:

$$x(t) = \sum_i A_i e^{j\omega_i t} \tag{2.40}$$

where the frequencies $\{\omega_i\}$ are <u>not</u> multiples of the fundamental frequency ω_0 defined by the inverse of the record length R $(\omega_0 = 2\pi/R)$. The summation index i takes the same positive and negative values, and $A_i = A_{-i}^*$ because the input signal is real.

When this input is presented to a nonlinear (Volterra) system, the r^{th}-order Volterra kernel will give rise to sinusoidal terms at the output that have frequencies defined by the sums or differences of these

incommensurate frequencies (see Section 2.1.3). For instance, a second-order system will generate the output:

$$y(t) = \sum_i A_i \cdot K_1(\omega_i) e^{j\omega_i t} + \sum_{i_1} \sum_{i_2} A_{i_1} \cdot A_{i_2} \cdot K_2(\omega_{i_1}, \omega_{i_2}) e^{j(\omega_{i_1} + \omega_{i_2})t} \tag{2.41}$$

where $K_1(\omega_i)$ and $K_2(\omega_{i_1}, \omega_{i_2})$ are the 1-D and 2-D Fourier transforms of the first and second order Volterra kernels respectively, sampled at the input frequencies. Note that the summation indices in Equation (2.41) account for sums and differences of input frequencies, since the summation indices take symmetric positive and negative values. Because the input-output signals are real, the differences between these input frequencies also arise (e.g., $\omega_{i_1} + \omega_{-i_2} = \omega_{i_1} - \omega_{i_2}$).

Based on Equation (2.41), the Fourier transform of $y(t)$ will reveal the values of the Fourier transforms of the Volterra kernels at the input frequencies and their sum/difference combinations thereof. The fact that the input frequencies are incommensurate secures that no overlapping frequencies will exist at the output because of high-order sum/difference combinations (i.e., the sum or the difference of two or more input frequencies will not coincide with another input frequency). A key practical issue is the ability to reconstruct accurately the values of the kernels at these incommensurate frequencies using FFT computed values that are found by numerical necessity at commensurate frequencies (leakage correction).

Note also that the highest significant harmonic of each of the input frequencies indicates the order of the system, since the Q^{th}-order kernel will give rise to a maximum Q^{th} harmonic. This is an important practical advantage of this method, since it obviates the need for selecting *a priori* the order of the Volterra model. Another advantage of this method is that it is rather robust in the presence of noise since it concentrates the signal power at a few specific frequencies. In addition, it is experimentally and computationally efficient.

The main drawback of this method is the leakage correction problem and the fact that it estimates the Volterra kernels of the system only at a few specific points in the frequency domain. Thus, if the kernels are relatively smooth in the frequency domain, this approach can be very efficient. However, if the kernels have multiple resonances and/or dissonances, then this approach is rendered ineffective for a limited number of input frequencies.

Arbitrary Inputs

The adjective "arbitrary" implies input signals that are not specially designed to attain specific waveforms, but they are input signals that occur naturally during the normal operation of the system. The only desirable (and, in fact, necessary) feature of these inputs is to be relatively broadband (but not necessarily white) so that they cover the entire range of frequencies of interest which determines with the system bandwidth. Naturally, the special case of random inputs (white or non-white) is covered by these methods.

The direct approach to the estimation of the discrete Volterra kernels using arbitrary (sampled) inputs, $x(n)$, and the resulting outputs, $y(n)$, is to formulate the problem in the vector-matrix form of Equation. (2.34). Then the simplest solution to this estimation problem is given by the ordinary least-squares (OLS) estimate [Eykhoff, 1963,1974; Hsieh, 1964]:

$$\hat{\mathbf{k}}_{OLS} = \left[\mathbf{X}'\mathbf{X}\right]^{-1}\mathbf{X}'\mathbf{y} \tag{2.42}$$

where the prime denotes transpose. This OLS estimate is unbiased, consistent and minimum-variance if the error terms in the vector $\boldsymbol{\varepsilon}$ are uncorrelated and zero-mean Gaussian—an assumption that cannot be made easily, especially for truncated Volterra models where the residuals are correlated and possibly non-Gaussian. In addition, physiological noise may exhibit outliers or other non-Gaussian statistical characteristics that call for robust estimation methods, as discussed later in this section.

This OLS estimate represents the most basic and direct approach to the Volterra kernel estimation problem and may yield satisfactory results if the $\left[P \times P\right]$ "Gram matrix" $\left[\mathbf{X}'\mathbf{X}\right]$ is not ill-conditioned (i.e., a numerically stable inverse exists) and the error terms (also termed "residuals") are approximately uncorrelated and Gaussian. Let us examine the estimation errors associated with the OLS estimator of Equation (2.42). By substitution of the OLS estimate of Equation (2.42) into the model of Equation (2.34), the estimated model output becomes:

$$\mathbf{y} = \mathbf{X}\left[\mathbf{X}'\mathbf{X}\right]^{-1}\mathbf{X}'\mathbf{y} \tag{2.43}$$

and the estimated output residuals are:

$$\begin{aligned}\hat{\boldsymbol{\varepsilon}} &= \mathbf{y} - \mathbf{y} \\ &= \left[\mathbf{I} - \mathbf{X}\left[\mathbf{X}'\mathbf{X}\right]^{-1}\mathbf{X}'\right]\mathbf{y}\end{aligned} \tag{2.44}$$

where $\mathbf{I}$ is the $\left[N \times N\right]$ identity matrix. Then the following measure of the OLS estimation errors can be derived that is given by the covariance matrix of the estimated parameter vector:

$$E\left[\left(\mathbf{k}-\mathbf{k}\right)\left(\mathbf{k}-\mathbf{k}\right)'\right]=\sigma^2\left[\mathbf{X}'\mathbf{X}\right]^{-1} \tag{2.45}$$

where σ^2 is the computed variance of the output residuals. It is evident that the kernel estimation variance is determined by the Gram matrix, which relates to the *input autocorrelation*. The key practical problem is the possible ill-conditioning of the Gram matrix $\left[\mathbf{X}'\mathbf{X}\right]$, which requires robust inversion methods such as singular value decomposition (SVD) or other generalized inverse methods [Fan & Kalaba, 2003; Kalaba & Tesfatsion, 1990; Udwadia & Kalaba, 1996]. In cases where the ill-conditioning may be caused by insufficient input bandwidth, the issue of proper testing or observation of the system must be addressed. This is particularly important for nonlinear systems that are prone to pitfalls regarding the adequacy of the input ensemble and deserves careful attention in practice.

Note that when the input vectors are orthogonal, then the Gram matrix $\left[\mathbf{X}'\mathbf{X}\right]$ becomes diagonal and the matrix inversion problems are avoided. This apparently attractive situation exists when the input signal is white (or quasi-white) and the Volterra functional expansion is orthogonalized, as discussed in Section 2.2 for the Wiener modeling approach.

If the residuals are correlated, then they can be pre-whitened with a linear transformation and the kernel estimate becomes the "generalized least squares" (GLS) solution given by:

$$\mathbf{k}_{GLS}=\left[\mathbf{X}'\mathbf{S}^{-1}\mathbf{X}\right]^{-1}\mathbf{X}'\mathbf{S}^{-1}\mathbf{y} \tag{2.46}$$

where $\mathbf{S}$ is the covariance matrix of the residuals. Note that this GLS estimate will minimize the estimation variance under the Gaussian assumption and remains unbiased and consistent. The GLS estimate of Equation (2.46) implies a change in the coordinate system defined by the columns of the input matrix $\mathbf{X}$:

$$\mathbf{Z}=\mathbf{B}\mathbf{X} \tag{2.47}$$

where the coordinate transformation matrix $\mathbf{B}$ is based on the residual covariance matrix as:

$$\mathbf{B}'\mathbf{B}=\mathbf{S}^{-1} \tag{2.48}$$

so that the residuals become uncorrelated. Note that the application of the GLS method requires knowledge of the residual covariance matrix--not a simple requirement in practice.

If the residuals are not Gaussian, then an estimate with smaller variance can be obtained by utilizing the log-likelihood function of the residuals (if such can be evaluated) as the cost function which is distinct from the quadratic cost function used in least-squares estimation methods (see below). This cost function is minimized through iterative procedures using gradient descent, as discussed in Section 4.2, or

other minimization methods. This case has been receiving increasing attention in recent years, especially when the residuals exhibit outliers (caused by impulsive noise or spurious measurements) that affect significantly the quadratic cost function (robust estimation).

It is evident from the foregoing that the size $[N \times P]$ of the matrix $\mathbf{X}$ must remain within computationally feasible bounds. This may become a serious problem for high-order systems (large Q) with large memory-bandwidth product (M). This problem is compounded by the fact that N must be much larger than P in order to obtain an estimate of reasonable accuracy in the presence of noise. The latter constraint impacts the length of the experimental data record and prolongs the experiment as M and/or Q increase.

For instance, a typical physiological system will have $M \sim 10^2$ and, therefore, the data-record must be much larger than $\sim 10^{2Q}/Q!$ for a Q^{th}-order model. Thus, if $Q = 3$, the minimum number of input-output samples is $N \sim 10^6$ and probably closer to 10^7, depending on the noise level. This requirement can be experimentally onerous or even infeasible (due, for instance, to system nonstationarities that prevent the collection of long data-records under the stationary assumption).

In order to address the practically serious problem of required record length, we advocate the key idea of kernel expansions on properly chosen bases to reduce the size of the unknown coefficient vector and, consequently, reduce the length of the required experimental data-record. This idea was originally proposed by Wiener and various implementations have been explored by several investigators [Bose, 1956; Lee, 1964; Amorocho & Brandstetter, 1971; Watanabe & Stark, 1975]. We have developed and advocate a variant of demonstrated efficacy that employs discrete-time Laguerre expansions [Marmarelis, 1993].

The kernel expansion approach represents the core of the advocated modeling methodology, based on extensive experience with various physiological systems. It usually reduces the required record length by a factor of 10^Q and results in significant benefits in terms of estimation accuracy and reduction in the experimental, as well as the computational, burden. Because of its importance, this methodology is discussed in detail in Section 2.3.

Fast Exact Orthogonalization and Parallel-Cascade Methods

Solution of the least-squares problem of Equation (2.34) can be also achieved with a variant of the Cholesky (QR) decomposition that is computationally efficient [Korenberg, 1988]. This technique,

termed "fast exact orthogonalization", develops the kernels estimates by successive orthogonalization of input data vectors (having the structure of a column of matrix $\mathbf{X}$) with respect to each other and to the output residuals. This method has been applied successfully to various biological systems but remains computationally intensive when the system memory-bandwidth product is large or when kernels of order higher than second need be estimated. It also remains vulnerable to noise in the data, especially noise corrupting the input data.

In order to make the practical estimation of high-order kernels more efficient, the "parallel cascade" approach was introduced by the same investigator, whereby the system model is developed by adding successive parallel branches of $L-N$ cascades (a linear filter followed by a static nonlinearity) until a satisfactory prediction error criterion is met [Korenberg, 1991]. The linear filters of these parallel cascades are estimated by "cross-correlation slices" and the unknown parameters of the static nonlinearities are determined by fitting procedures. This method is computationally efficient, even for high-order models with large memory-bandwidth product, but remains sensitive to noise in the data and usually yields a very large number of parallel cascades (in the hundreds). Although the latter can be consolidated in the form of equivalent kernels computed from the parameters of the parallel cascades, this "parallel-cascade" model does not lend itself to physiological interpretation--unlike the method of "principal dynamic modes" discussed in Section 4.2 that employs a small number of parallel cascades.

Iterative Cost-Minimization Methods for Non-Gaussian Residuals

We begin with the linear estimation problem described by Equation (2.34) for the case of residuals with arbitrary joint probability density function (joint PDF) $p(\boldsymbol{\varepsilon})$. Following the maximum likelihood framework, that has been proven to yield optimal (i.e., minimum variance) estimates, we seek to maximize over the parameter vector $\boldsymbol{k}$ the likelihood function:

$$L(\mathbf{k}) = p(\mathbf{y} - \mathbf{X}\mathbf{k}) \tag{2.49}$$

In practice, we often minimize the negative log-likelihood function (viewed as a cost function) because many PDFs take a decaying exponential form. Furthermore, the joint PDF is avoided as impractical by performing "pre-whitening" of the model residuals (if necessary) with the transformation indicated in Equation (2.47). It is understood that the latter transformation simply provides uncorrelated residuals, which implies true "pre-whitening" (i.e., statistical independence) only in the Gaussian case. Nonetheless, in the spirit of pragmatic accommodation, we replace in practice the joint PDF with the product of the single-variable PDFs for all residuals (assuming statistical independence of the residuals after pre-whitening).

For instance, in the Gaussian case, the likelihood function becomes under the assumption of white (i.e., statistically independent) residuals:

$$L(\mathbf{k}) = \left(2\pi\sigma^2\right)^{-N/2} \exp\left[-\frac{1}{2\sigma^2}\sum_{n=1}^{N}\left[y(n) - \mathbf{x}_n^{'}\mathbf{k}\right]^2\right] \tag{2.50}$$

where σ^2 is the residual variance, N is the number of samples and $\boldsymbol{x}_n'$ is the transpose of the nth row of the matrix $\mathbf{X}$. Instead of maximizing L over $\boldsymbol{k}$, we can minimize $(-\log L)$ over $\boldsymbol{k}$. This leads to a quadratic cost function:

$$C(\mathbf{k}) \propto -\log L = \frac{N}{2}\log\left(2\pi\sigma^2\right) + \frac{1}{2\sigma^2}\sum_{n=1}^{N}\left[y(n) - \mathbf{x}_n^{'}\mathbf{k}\right]^2 \tag{2.51}$$

which can be minimized in closed form by the ordinary least-squares estimate of Equation (2.42). Note that the GLS estimate of Equation (2.46) is the closed form solution of this cost minimization problem for correlated residuals with covariance matrix $\mathbf{S}$. Of course, Equation (2.51) can be also solved iteratively using gradient descent (or any other minimization) methods.

Let us now consider a case of *non-Gaussian*, white residuals with PDF:

$$p(\varepsilon) = \lambda \exp\left[-\alpha|\varepsilon|^{\beta}\right] \tag{2.52}$$

where $\alpha, \beta > 0$ are dispersion and shape parameters respectively, and the value of λ that satisfies the PDF normalization condition is:

$$\lambda = \frac{\alpha\beta}{2\Gamma\left(1/\beta\right)} \tag{2.53}$$

where Γ denotes the Gamma function. This class of PDFs includes the Gaussian $(\beta = 2)$ and the Laplacian $(\beta = 1)$, and yields the cost function:

$$C(\mathbf{k}) = -\log\lambda + \alpha\sum_{n=1}^{N}\left|y(n) - \mathbf{x}_n'\mathbf{k}\right|^{\beta} \tag{2.54}$$

which can be minimized over $\boldsymbol{k}$ through gradient-descent iterative methods, since it is differentiable except at $y(n) = \mathbf{x}_n^{'}\mathbf{k}$. Note that the gradient components are given by:

$$\frac{\partial Q}{\partial k_i} = -\alpha\beta \cdot \sum_{n=1}^{N}\mathrm{sgn}\left[\varepsilon(n)\right] x_{n,i}\left|\varepsilon(n)\right|^{\beta-1} \tag{2.55}$$

when $\varepsilon(n) \neq 0$ (the gradient should be set to zero if $\varepsilon(n) = 0$), $\mathrm{sgn}[\cdot]$ denotes the signum function and $x_{n,i}$ is the ith element of the vector $\boldsymbol{x}_n'$ (for $i = 1,...,M$).

The estimation errors associated with iterative cost-minimization procedures haven been studied extensively in the literature in connection with a variety of procedures. Since a host of available procedures exists (based on gradient descent or random search), we will defer to the vast literature on the subject [Eykhoff, 1974; Haykin, 1994; Hassoun, 1995]. We simply note that the key issues are: (a) avoidance of local minima; and (b) rapid convergence of the iterative algorithm.

These iterative cost-minimization procedures can be used also to solve the daunting nonlinear regression problem, where the nonlinearity arises from the model form and not from the non-Gaussian residuals. An example is the training (i.e., the iterative parameter estimation) of the network models discussed in Section 2.3.3 that are equivalent to the Volterra models of nonlinear systems. In these network models, certain unknown parameters enter nonlinearly and, therefore, the simple formulation of Equation (2.34) is not applicable. The chain rule of differentiation has to be used in this context (refer to as "error back propagation") for iterative estimation of the unknown network parameters. Although this iterative method has been used extensively, it still offers challenges in some applications.

2.2. WIENER MODELS

The motivation for the introduction of the Wiener series (and the associated Wiener models) is found in the desire to diagonalize the Gram matrix $[\mathbf{X}'\mathbf{X}]$ of the previous section by orthogonalizing the "input vectors". This also addresses the "model truncation" problem by decoupling the various kernels through orthogonalization of their corresponding functionals, and subsequently facilitates their separate estimation and reduces the size of the estimation problem. This is similar to the procedure followed in order to facilitate the estimation of the expansion coefficients of a function expansion on a basis of functions by orthogonalizing the expansion basis over the selected domain of the independent variable (see Appendix I).

Wiener proposed this approach in the context of functionals (systems) by orthogonalizing the Volterra functionals for a Gaussian white noise (GWN) input using a Gram-Schmidt orthogonalization procedure (see Appendix III). The basic properties of GWN are discussed in Appendix II. The GWN input *power level* defines the region of functional orthogonality (i.e., the range of input power for which orthogonality holds) in a manner akin to the role of the domain of the independent variable in defining orthogonal basis functions. Wiener studied extensively the stochastic process of Brownian motion and the mathematical properties of its "derivative" (the GWN), including its stochastic integrals that led him

to the introduction of what he termed the "homogeneous chaos" – a hierarchy of stochastic integrals involving GWN that was a forerunner of the Wiener series [Wiener, 1938].

Wiener's idea extends to functional spaces the logic established in function spaces by the introduction of orthogonal function bases to facilitate the evaluation of the expansion coefficients of square-integrable functions. This logic entails the decoupling of simultaneous equations through orthogonalization and was extended by Wiener to *functional* expansions of unknown system functionals by combining Volterra's key idea of extending the mathematical formalism from enumerably infinite vector spaces to continuous function spaces on one hand, with the statistical properties of GWN and its integrals (homogeneous chaos) on the other hand. It is critical for the comprehension of the functional expansions to view a function as a "vector" with enumerably infinite number of dimensions.

If one draws the analogy between the Volterra series expansion of an analytic functional and a Taylor series expansion of an analytic function, then the analogy can be also drawn between the Wiener series of orthogonal functionals with GWN input and a Hermite orthogonal expansion of a square-integrable function, because the latter employs a Gaussian weighting function. In fact, the structure of the Wiener functionals resembles the structure of the Hermite polynomials. It must be noted again that the Wiener kernels of a system are generally different from its Volterra kernels, although specific analytical relations exist between the two sets that are presented below.

Even though Wiener's ideas had great influence and shaped constructively our thinking on nonlinear system identification/modeling, the practical relevance of the orthogonal Wiener series (for GWN inputs) has diminished in recent years due to the advent of superior kernel estimation methodologies that are applicable for non-GWN inputs, and the practical necessity of utilizing non-GWN inputs in the study of physiological system under natural operating conditions. Nonetheless, we will present Wiener's seminal ideas in this section, because they still exert considerable influence and are instructive in understanding the evolution of this field.

Wiener's critical contributions to the problem of nonlinear system identification/modeling are two: (1) the suggestion that GWN is an effective test input for identifying nonlinear dynamic systems of a very broad class, and (2) the introduction of specific procedures for the estimation of the unknown system kernels from input-output data in the framework of the orthogonal Wiener series. Even though better kernel estimation procedures (that do not require orthogonalization of the functional expansion or white-noise inputs) have been developed in recent years, Wiener's seminal contributions gave tremendous initial impetus to the field and "blazed the trail" for many investigators who followed his

lead and advanced the state of the art. For this, he is properly considered a pioneer and a prominent founder of the field.

The idea that GWN is an effective test input for nonlinear system identification and modeling (the same way the impulse function is an effective test input for linear time-invariant system identification) is of particular importance and interest. Aside of the mathematical properties of GWN that facilitate the Wiener kernel estimation, the idea engenders the notion that the nonlinear system must be tested by all possible input waveforms that are expected to stimulate the system under normal operation or by a dense, representative subset of this "natural input ensemble". This fundamental idea is revisited throughout the book in a context broader than the original Wiener suggestion (i.e., only a subset of GWN comprises the "natural input ensemble" and, therefore, GWN--even band-limited--may exhibit unnecessary redundancy). The concept is clear but the practical implications depend on the degree of redundancy of GWN relative to the natural input ensemble of the system.

In principle, the Volterra kernels of a system cannot be directly determined from input-output data unless the Volterra expansion is of finite order. For a finite-order Volterra expansion, kernel measurement methods through least-squares fitting procedures or by use of specialized inputs (e.g., multiple impulses or multiple sinusoids) were discussed in the previous section. These methods have numerical or experimental limitations and potential pitfalls, related primarily to the effects of the model truncation error (correlated residuals leading to estimation biases) and the "richness" of the utilized input ensemble (misleading results, if the system functional space is not probed densely by the input signals).

These two fundamental limitations motivated Wiener to introduce the GWN as an "effective test input" (i.e., an input signal that probes densely the operational space of all systems) and to propose the orthogonalization of the Volterra functional expansion (i.e., the orthogonal Wiener expansion makes the residuals orthogonal to the estimated model prediction for a GWN input). The latter results in a new set of kernels (Wiener kernels), which are distinct from the Volterra kernels of the system, in general. This can be viewed as a "structural bias" of the Wiener kernels relative to the Volterra kernels of a system, since the residuals of a truncated Wiener model remain correlated (i.e., non-white). The difference is that the "structural bias" of the Wiener kernels is determined by the GWN input power level (one parameter), whereas the estimation bias of the Volterra kernels (in a truncated model) depend on the utilized input ensemble that can be different from case to case--thus introducing a source of inconsistency in the obtained results (estimated kernels).

For these reasons, Wiener suggested the orthogonalization of the Volterra series for a GWN test input (see Appendix III and the Historical Note #2 at the end of this chapter). The resulting orthogonal functional series is termed the "*Wiener series*" and exhibits the aforementioned advantages. Additional advantages, due to its orthogonality, are the "finality" of the Wiener kernel estimates (i.e., they do not change if additional higher-order terms are added) and the rapid convergence of the expansion for a GWN input (i.e., least truncation error for given model order). Note, however, that the latter advantage is true only for GWN inputs (as discussed later).

The functional terms of the Wiener series are termed the "*Wiener functionals*" and are constructed on the basis of a Gram-Schmidt orthogonalization procedure requiring that the covariance between any two Wiener functionals be zero for a GWN input, as detailed in Appendix III. The resulting Wiener series expansion of the output signal takes the form:

$$y(t) = \sum_{n=0}^{\infty} G_n[h_n; x(t'), t' \le t]$$

$$= \sum_{n=0}^{\infty} \sum_{m=0}^{[n/2]} \frac{(-1)^m n! P^m}{(n-2m)! m! 2^m} \int_0^{\infty} \cdots \int_0^{\infty} h_n(\tau_1, ..., \tau_{n-2m}, \lambda_1, \lambda_1, ..., \lambda_m, \lambda_m) x(t-\tau_1) \cdots x(t-\tau_{n-2m}) d\tau_1 \cdots d\tau_{n-2m} d\lambda_1 \cdots d\lambda_m \qquad (2.56)$$

where [n/2] is the integer part of n/2 and P is the *power level* of the GWN input. The leading integral term of the nth-order Wiener functionals has the form of the nth-order Volterra functional (of course with a different kernel). The Wiener kernel is integrated in the non-leading integral terms (of lower homogeneous order) for each Wiener functional to reduce appropriately the dimensionality and secure the orthogonality of the Wiener functionals. Note that the nth order Wiener functional has $\lfloor n/2 \rfloor$ integral terms that contain the same Wiener kernel convolved with the input n, $(n-2)$,..., $\lceil (n-1)/2 \rceil$ times (i.e., each of these integral terms has the form of a homogeneous functional of order equal to the number of convolved inputs).

The Wiener functionals $\{G_n(t)\}$ are constructed orthogonal in the statistical sense of zero co-variance: $E[G_n(t) G_m(t')] = 0$, for $m \neq n$ and for all values of t and t'; where $E[\cdot]$ denotes the "expected value" operator which forms the statistical average of the random quantity within the brackets over the entire ensemble of this random quantity. For ergodic and stationary random processes, this ensemble average can be replaced by a time average over the entire time axis (from $-\infty$ to $+\infty$). In practice, of course, these averages (both over ensemble and over time) form incompletely, because of the

inevitably finite ensemble and/or time record of data, leading to inaccuracies that are discussed in detail in Section 2.4.2.

The orthogonality of the Wiener functionals is also compromised in practice by the necessity of using *band-limited* GWN inputs (instead of the ideal GWN that has infinite bandwidth and is, therefore, not physically realizable). This situation is akin to the common approximation of the Dirac delta function (a mathematical idealization that is not physically realizable) with an impulse waveform of finite time-support (width) that is sufficiently small for the requirements of each specific application. In the same vein, the ideal GWN input is approximated in practice by a band-limited GWN signal with sufficiently broad bandwidth as to cover the bandwidth of the system under study.

2.2.1. Relation Between Volterra and Wiener Models

The set of Wiener kernels $\{h_n\}$ is, in general, different from the set of Volterra kernels $\{k_n\}$ of the system and dependent on the GWN input power level P. Specific mathematical relations exist between the two sets of kernels (when they both exist) that can be derived by equating the two series expansions. These relations are given in the time domain by:

$$h_n(\tau_1,...,\tau_n) = \sum_{m=0}^{\infty} \frac{(n+2m)!P^m}{n!m!2^m} \int_0^{\infty}...\int_0^{\infty} k_{n+2m}(\tau_1,...,\tau_n,\lambda_1,\lambda_1,...,\lambda_m,\lambda_m)\,d\lambda_1...d\lambda_m \tag{2.57}$$

or in the frequency domain:

$$H_n(\omega_1,...,\omega_n) = \sum_{m=0}^{\infty} \frac{(n+2m)!P^m}{n!m!2^m(2\pi)^m} \int_{-\infty}^{\infty}...\int_{-\infty}^{\infty} K_{n+2m}(\omega_1,...,\omega_n,u_1,-u_1,...,u_m,-u_m)\,du_1...du_m \tag{2.58}$$

where ω_i or u_i denote frequency in rad/sec. It is evident from Equation (2.57) that the nth-order Wiener kernel depends, not only on the nth-order Volterra kernel, but also on all higher order Volterra kernels of the same parity. Note that the parity (odd/even) separation in the expressions of the Wiener kernels provides that the even/odd order Wiener kernels are polynomials in P with coefficients depending on all the higher even/odd order Volterra kernels. Thus, a system with an even symmetric (or odd-symmetric) nonlinearity will have only even-order (or odd-order) Volterra and Wiener kernels.

Similar expressions can be derived for the Volterra kernels of the system in terms of the Wiener kernels of higher (and equal) order and the respective power level P, by collecting the terms in the Wiener functional with the same number of input product terms from all the Wiener functionals [Marmarelis, 1976]. Note that, for finite order models, the Volterra and Wiener kernels of the two

highest orders (odd and even) are identical, because of the absence of higher order kernels of the same parity.

As an illustrative example, consider the $L-N$ cascade system of Example 2.2, but with a cubic nonlinearity. Its Volterra kernels are given by Equation (2.13) for $r=1,2,3$, with $k_0=0$. According to Equation (2.57), the equivalent Wiener kernels of this system for GWN input power level P are:

$$h_0 = P\int_0^\infty k_2(\lambda,\lambda)\,d\lambda = P\alpha_2\int_0^\infty g^2(\lambda)\,d\lambda \tag{2.59}$$

$$h_1(\tau) = k_1(\tau) + 3P\int_0^\infty k_3(\tau,\lambda,\lambda)\,d\lambda = \alpha_1 g(\tau) + 3P\alpha_3 g(\tau)\int_0^\infty g^2(\lambda)\,d\lambda \tag{2.60}$$

$$h_2(\tau_1,\tau_2) = k_2(\tau_1,\tau_2) = \alpha_2 g(\tau_1)\,g(\tau_2) \tag{2.61}$$

$$h_3(\tau_1,\tau_2,\tau_3) = k_3(\tau_1,\tau_2,\tau_3) = \alpha_3 g(\tau_1)\,g(\tau_2)\,g(\tau_3) \tag{2.62}$$

If we wish to express the Volterra kernels in terms of the Wiener kernels of this system, then:

$$k_0 = h_0 - P\int_0^\infty h_2(\lambda,\lambda)\,d\lambda \tag{2.63}$$

$$k_1(\tau) = h_1(\tau) - 3P\int_0^\infty h_3(\tau,\lambda,\lambda)\,d\lambda \tag{2.64}$$

since the first three Wiener functionals have the structure:

$$G_1(t) = \int_0^\infty h_1(\tau)\,x(t-\tau)\,d\tau \tag{2.65}$$

$$G_2(t) = \int_0^\infty\int h_2(\tau_1,\tau_2)\,x(t-\tau_1)\,x(t-\tau_2)\,d\tau_1 d\tau_2 - P\int_0^\infty h_2(\lambda,\lambda)\,d\lambda \tag{2.66}$$

$$G_3(t) = \iiint_0^\infty h_3(\tau_1,\tau_2,\tau_3)\,x(t-\tau_1)\,x(t-\tau_2)\,x(t-\tau_3)\,d\tau_1 d\tau_2 d\tau_3 - 3P\iint_0^\infty h_3(\tau,\lambda,\lambda)\,x(t-\tau)\,d\tau\,d\lambda \tag{2.67}$$

One interesting implication of the derived relation between the Volterra and the Wiener kernels of the system is that the first-order Wiener kernel of a nonlinear system is, in general, different from the linear part of the system (the first-order Volterra kernel), and it is actually dependent on all higher odd-order Volterra kernels, i.e., contains some of the odd-order system nonlinearities. This demonstrates the faster convergence of the Wiener orthogonal expansion, where even the first-order functional term reflects some of the nonlinear characteristics of the system (see Equation (2.57) for $n=1$ or Equation (2.60) in the example). At the same time, this "projection" of higher odd-order Volterra kernels on the first-order Wiener kernel may obscure the interpretation of "linearized approximations" obtained in the

Wiener framework. This point has important practical implications on "apparent transfer function" measurements often used in practice and the corresponding coherence measurements, as discussed in Section 2.2.5.

With regard to possible pitfalls in the interpretation of Wiener kernels, the reader must be reminded that the Wiener series is constructed orthogonal with respect to GWN inputs of certain power level P which determines the range of validity of the orthogonality between any two Wiener functionals. This "range of orthogonality" in function space is determined by the product of the input bandwidth and variance. Therefore, since P determines the range of the orthogonal "coordinate system" represented by the Wiener functionals, the obtained Wiener kernel estimates depend on the specific P value of the utilized white (or quasi-white) input, as indicated by Equation (2.57), and should be expected to provide good model predictions for input signals with bandwidth-variance products comparable to P (if the model is truncated). If the model is complete, then the predictions will be good for *any* input signal. Since the estimated Wiener kernels are generally different for different P values, they should be reported in the literature with reference to the P value for which they were obtained.

The reader may wonder why orthogonality is sought. As mentioned briefly earlier, there are three main reasons why orthogonality is desirable. The first reason is that an orthogonal basis spans the functional space (within the range of its validity) most efficiently. That is, the Wiener series is expected to have faster convergence than the Volterra series for a GWN input (i.e., smaller output prediction error for given order of truncated model). However, this cannot be guaranteed for an arbitrarily chosen input signal. Recall that GWN is an ergodic random process with the same power over all frequencies, and thus constitutes an exhaustive test input, i.e., it tests the system with all possible input waveforms, given sufficient time. Consequently, it can be expected to provide a better truncated model of the system over all possible inputs. This is the rationale for Wiener's suggestion of using GWN test inputs for kernel estimation. However, this fact does not exclude the possibility of the opposite to be true for certain specific input signals, raising the important issue of defining a "*natural ensemble*" of inputs for each specific system, that should be used for kernel estimation in order to provide the truncated model with the best convergence for the system at hand (the issue is moot for a complete model).

The second reason for seeking orthogonality is that, if the expansion basis is orthogonal, then the truncated model can be extended to include higher-order terms without affecting the lower-order terms already estimated (finality of orthogonal expansions). The third reason is that the orthogonality allows the estimation of the system kernels in a relatively simple way using cross-correlation (or co-variance)

when the input is GWN (as discussed in Section 2.2.3) or through diagonalization of the Gram matrix (as discussed in Section 2.1.5). This is analogous to the determination of the expansion coefficients of a given vector or function on an orthogonal vector or function basis (discussed in Appendix I).

This last advantage of orthogonality has been the primary motivation for the initial use of the Wiener series in the actual identification of nonlinear systems, although the first advantage of orthogonality (faster convergence) was the primary motivation for its introduction in connection with GWN test inputs.

It must be noted that the GWN input is not the only signal with respect to which the Volterra functional series can be orthogonalized. The orthogonalization can be achieved for other classes of input signals that possess suitable autocorrelation properties, such as the CSRS class of quasi-white signals discussed in Section 2.2.4. For each such signal class, a corresponding orthogonal functional series can be constructed and the associated set of kernels can be estimated.

The Wiener Class of Systems

The class of nonlinear time-invariant systems for which a Wiener series expansion exists is different from the Volterra class defined by the absolute integrability condition of Equation (2.4). The Wiener class is comprised of systems that generate outputs with finite variance in response to a GWN input.

Since the output variance of the Wiener model for a GWN input is (see Appendix III for derivation):

$$\sigma_y^{\,2} = \sum_{r=0}^{\infty} r!P^r \int \ldots \int_0^{\infty} h_r^2(\tau_1,\ldots,\tau_r)\, d\tau_1 \ldots d\tau_r \tag{2.69}$$

the condition for finite output variance becomes the square-integrability condition on the Wiener kernels:

$$\int \ldots \int_0^{\infty} h_r^2(\tau_1,\ldots,\tau_r)\, d\tau_1 \ldots d\tau_r \le \frac{c_r}{r!P^r} \tag{2.70}$$

where $\{c_n\}$ is a convergent series of non-negative scalars.

Therefore, the Wiener class of systems is defined by a square integrability condition for the Wiener kernels with a radius of convergence determined by the power level of the GWN input (unlike the absolute integrability condition for the Volterra kernels that has a radius of convergence determined by the uniform bound on the amplitude of the input signal). This square-integrability condition excludes kernels with delta functions from the Wiener class (e.g., solitary static nonlinearities) which are admissible in the Volterra class of systems.

We should note at this point that the output of a Wiener model is a stationary random process – in general, non-white. Therefore the analysis of such output signals has to be statistical and make use of probability density functions, correlation functions, and spectra (including high-order ones because of the system nonlinearities). Thus, it is deemed useful to provide a brief overview of the basic tools for the characterization and analysis of stationary random processes (signals) in Appendix IV. Because of its pivotal importance, we discuss the basic properties of the GWN process separately in Appendix II.

Examples of Wiener Models

As illustrative examples, let us examine the equivalent Wiener models for the systems used previously as Examples 2.1-2.4 for Volterra models.

Example 2.1 of a static nonlinear system has no formal equivalent Wiener model because its kernels are not square integrable (being composed of delta functions) and do not satisfy the condition (2.70).

Example 2.2 of the $L-N$ cascade system has Volterra kernels given by Equation (2.13). The Wiener kernels for this system (in the general case of infinite order nonlinearity and not the case of cubic nonlinearity discussed earlier) are found by use of Equation (2.57) as:

$$h_r(\tau_1,...,\tau_r)=\sum_{m=0}^{\infty}\frac{(r+2m)!P^m}{r!m!2^m}\alpha_{r+2m}\left[\int_0^{\infty}g^2(\lambda)d\lambda\right]^m g(\tau_1)...g(\tau_r) \tag{2.71}$$

This illustrates again the general observation that the Wiener series has a different pattern of convergence than the Volterra series and, therefore, truncated models will yield different prediction accuracies depending on the specific input. The closer the input signal comes to the GWN input that was used to estimate the Wiener kernels, the better the relative performance of the Wiener model. However, this performance advantage may turn into a deficit for certain inputs that deviate significantly from the aforementioned GWN input.

Example 2.3 of the $L-N-M$ cascade has Wiener kernels given by combining Equation (2.14) with Equation (2.57) as:

$$h_r(\tau_1,...,\tau_r)=\sum_{m=0}^{\infty}\frac{(r+2m)!P^m}{r!m!2^m}\alpha_{r+2m}\int_0^{\min(\tau_1,...,\tau_r)}\left\{\int_{\lambda}^{\infty}g^2(\lambda')d\lambda'\right\}^m h(\lambda)g(\tau_1-\lambda)...g(\tau_r-\lambda)d\lambda \tag{2.72}$$

Note that as the posterior filter $h(\lambda)$ tends to a delta function (i.e., its memory decreases or its bandwidth increases), the Wiener kernels for this "sandwich" model tend to their Volterra counterparts of Equation (2.14), as expected, since the L-N-M cascade tends to the L-N cascade in this case.

Example 2.4 presents a greater challenge in evaluating its Wiener kernels because of the complexity of the expressions for its high-order Volterra kernels. This subject will be examined in Section 3.2, when we analyze the relation between Volterra models and nonlinear differential equations, in connection with studies of nonlinear feedback.

Comparison of Volterra/Wiener Model Predictions

It is important to re-emphasize that the Wiener kernels depend on the GWN input power level, whereas the Volterra kernels are independent of any input characteristics. This is due to the fact that the Wiener kernels are associated with an orthogonal functional expansion (when the input is GWN of some power level P), whereas the Volterra kernels are associated with an analytic functional expansion that only depends on the functional derivatives, which are characteristic of the system but independent of the specific input. This situation can be likened to the difference between the coefficients of an orthogonal and an analytic expansion of a function – where the coefficients of the orthogonal expansion depend on the interval of expansion whereas the coefficients of the analytic expansion depend only on the derivatives of the function at the reference point (see Appendix I). It is therefore imperative that Wiener kernel estimates be reported in the literature with reference to the GWN input power level that was used to estimate them. On the other hand, the Volterra kernels are fixed for a given system and their estimates are input-invariant for a complete model.

When a complete set of Wiener kernels is obtained for a given system, then the complete set of Volterra kernels of the system can be evaluated using the following relationship:

$$k_n(\tau_1,...,\tau_n) = \sum_{m=0}^{\infty} \frac{(-1)^m (n+2m)! P^m}{n! m! 2^m} \int_0^{\infty} ... \int h_{n+2m}(\tau_1,...,\tau_n,\lambda_1,\lambda_1,...,\lambda_m,\lambda_m)\, d\lambda_1 ... d\lambda_m \qquad (2.73)$$

which bears an astonishing resemblance with the reverse relationship (expressing the Wiener kernels in terms of the Volterra kernels) given by Equation (2.57), i.e., the only difference is the $(-1)^m$ term in the series.

When a complete set of Wiener kernels cannot be obtained, then approximations of Volterra kernels can be obtained from Wiener kernels of the same order measured with various input power levels, utilizing the polynomial dependence of the Wiener kernels on the GWN input power level P, as described by Equation (2.57). For instance, the 1st-order Wiener kernel as a function of different values of P is given by Equation (2.57) as:

$$h_1(\tau;P)=\sum_{m=0}^{\infty}\frac{(2m+1)!P^m}{m!2^m}\int_0^{\infty}k_{2m+1}(\tau,\lambda_1,\lambda_1,...,\lambda_m,\lambda_m)d\lambda_1...d\lambda_m \tag{2.74}$$

which can be used to eliminate the contribution of k_3 from two measurements of h_1 for two different P_1 and P_2 values (following a form of Gaussian elimination):

$$h_1(\tau;P_1)-\frac{P_1}{P_2}h_1(\tau;P_2)=\left(\frac{P_2-P_1}{P_2}\right)k_1(\tau)$$

$$+\sum_{m=2}^{\infty}\frac{(2m+1)!P_2^m}{m!2^m}\left(\frac{P_1^m}{P_2^m}-\frac{P_1}{P_2}\right)k_{2m+1}(\tau,\lambda_1,\lambda_1,...,\lambda_m,\lambda_m)d\lambda_1...d\lambda_m \tag{2.75}$$

This procedure can be continued with a third measurement for $P=P_3$ in order to eliminate the contribution of k_5 by computing the expression:

$$\left[h_1(\tau;P_1)-\frac{P_1}{P_2}h_1(\tau;P_2)\right]-\frac{P_1-P_2}{P_1-P_3}\left[h_1(\tau;P_1)-\frac{P_1}{P_3}h_1(\tau;P_3)\right]=$$

$$=\frac{(P_2-P_1)(P_3-P_2)}{P_2P_3}k_1(\tau)+\{\text{Terms involving } k_7 \text{ and higher order Volterra kernels}\} \tag{2.76}$$

Therefore, this procedure can be continued until the contribution of all significant Volterra kernels is eliminated, yielding a good estimate of $k_1(\tau)$. This procedure can be used for any order of estimated Wiener kernel and can be formulated mathematically as an inversion of a Vandermonde matrix defined by the various values of the GWN input power level used [Marmarelis & Sams, 1982].

Complete sets of either the Wiener or the Volterra kernels can be used to predict the system output to any given input (for which the series converge). However, if the obtained Wiener or Volterra model is incomplete (truncated), then the accuracy of the predicted system output will be, in general, different for the two models and for each different input signal.

For instance, using the cascade example above, the complete 3rd-order Volterra or Wiener models will predict precisely the output for any given input. However, if an incomplete model (e.g., truncated at the 2nd-order) is used, then the difference in output prediction between the 2nd-order Wiener $(\hat{y}_w)$ and Volterra $(\hat{y}_v)$ models is:

$$y_v(t)-y_w(t)=P\int_0^{\infty}k_2(\lambda,\lambda)d\lambda+3P\int_0^{\infty}k_3(\tau,\lambda,\lambda)x(t-\tau)d\lambda$$

$$= P\left(\alpha_2 + 3\alpha_3\right)_2 \int_0^\infty g^2(\lambda)\, d\lambda \cdot \int_0^\infty g(\tau)\, x(t-\tau)\, d\tau \qquad (2.77)$$

i.e., the difference depends on the 2nd-order and 3rd-order Volterra kernels (because of the lower order projections of higher order terms in the Wiener functionals) which reduces to a simpler relation in this specific example, given by Equation (2.77) as proportional to the 1st-order Volterra functional. This model prediction difference depends generally on P and the specific input $x(t)$, as expected. The truncated Wiener model will have the minimum prediction mean-square error for a GWN input with power level P, due to its orthogonality. However, for arbitrary input signals, the relative prediction accuracy of the two truncated models (of the same order) will vary.

The proper way of comparing the Volterra/Wiener model predictions is to evaluate the mean-square errors in the two cases for a certain ensemble of inputs. This task tends to be rather cumbersome analytically, when we go beyond the 2nd-order functionals. Therefore, we will use here a 2nd-order example for simplicity of derivations, with the understanding that the essential conclusions hold for higher order cases as well.

We will compare the mean-square errors (MSEs) of the two types of 1st-order model predictions for arbitrary random inputs. For the 1st-order Volterra model, the MSE is:

$$Q_v \equiv E\left[V_2^2(t)\right] = \int_0^\infty\int_0^\infty\int_0^\infty\int_0^\infty k_2(\tau_1,\tau_2)\, k_2\left(\tau_1',\tau_2'\right) E\left[x(t-\tau_1)\, x(t-\tau_2)\, x\left(t-\tau_1'\right) x\left(t-\tau_2'\right)\right] d\tau_1 d\tau_2 d\tau_1' d\tau_2' \qquad (2.78)$$

and depends on the fourth-order autocorrelation function of the input. The MSE of the 1st-order Wiener model prediction is:

$$\begin{aligned} Q_w \equiv E\left[G_2^2(t)\right] = & \int_0^\infty\int_0^\infty\int_0^\infty\int_0^\infty h_2(\tau_1,\tau_2)\, h_2\left(\tau_1',\tau_2'\right) E\left[x(t-\tau_1)\, x(t-\tau_2)\, x\left(t-\tau_1'\right) x\left(t-\tau_2'\right)\right] d\tau_1 d\tau_2 d\tau_1' d\tau_2' \\ & -2P\int_0^\infty h_2(\lambda,\lambda)\, d\lambda \cdot \int_0^\infty\int_0^\infty h_2(\tau_1,\tau_2)\, E\left[x(t-\tau_1)\, x(t-\tau_2)\right] d\tau_1 d\tau_2 \\ & +\left\{P\int_0^\infty h_2(\lambda,\lambda)\, d\lambda\right\}^2 \end{aligned} \qquad (2.79)$$

and depends on the fourth-order and second-order autocorrelation functions of the input.

It is evident from these expressions that a comparison between the two MSEs would not be a simple matter for an arbitrary ensemble of inputs, if it were not for the fact that $k_2\left(\tau_1,\tau_2\right) \equiv h_2\left(\tau_1,\tau_2\right)$ for a 2nd-order system. Therefore, for this example of a 2nd-order system, we have:

$$Q_v - Q_w = 2P\int_0^\infty h_2(\lambda,\lambda)\,d\lambda \cdot \iint_0^\infty h_2(\tau_1,\tau_2)\,\phi(\tau_1-\tau_2)\,d\tau_1 d\tau_2 - \left\{P\int_0^\infty h_2(\lambda,\lambda)\,d\lambda\right\}^2 \qquad (2.80)$$

where ϕ denotes the second-order autocorrelation function of the input signal.

For an arbitrary input ensemble, Equation (2.80) indicates that a reduction in prediction MSE will occur for the Wiener model $(\Delta Q > 0)$ if:

$$\int_0^\infty\int h_2(\tau_1,\tau_2)\,\phi(\tau_1-\tau_2)\,d\tau_1 d\tau_2 > \frac{1}{2}P\int_0^\infty h_2(\lambda,\lambda)\,d\lambda \qquad (2.81)$$

Therefore, the improvement of the Wiener model prediction depends on the autocorrelation properties of the input ensemble and its relation to the 2nd-order kernel (i.e., the nonlinear characteristics of the system). It is conceivable that for some inputs the Wiener model prediction may be worse than the Volterra model prediction of the same order. However, as the input ensemble tends to GWN (i.e., $\phi(\tau_1-\tau_2) \to P\delta(\tau_1-\tau_2)$), the improvement of the Wiener model prediction becomes guaranteed, since the left-hand side of (2.81) becomes twice the right-hand side. It must be emphasized that these statements apply only to truncated models, and the complete models (Volterra or Wiener) give the same output prediction.

For an alternate GWN test input of power level P', the MSE difference between the two-model predictions is given by:

$$\Delta Q(P') = (2P' - P)P\left\{\int_0^\infty h_2(\lambda,\lambda)\,d\lambda\right\}^2 \qquad (2.82)$$

which indicates that the reduction in the MSE of the Wiener model prediction increases as the alternate GWN input power level P' increases (relative to the power level P of the GWN input for which the Wiener series was orthogonalized) but it may become negative if $P' < P/2$. For $P = P'$, a reduction in prediction MSE occurs $(\Delta Q > 0)$ as expected. Note that the rate of this reduction is proportional to the square of the integral of the 2nd-order kernel diagonal (i.e., it depends on the system nonlinear characteristics). This illustrative result is strictly limited to 2nd-order systems and must not be generalized to higher order systems and models.

2.2.2. Wiener Approach to Kernel Estimation

The estimation of the unknown system kernels is the key task in nonparametric system modeling and identification. As discussed earlier, the primary motivation for the introduction of the Wiener series has been the facilitation of the kernel estimation task.

In the general case of an infinite Volterra series, the accurate (unbiased) estimation of the Volterra kernels from input-output data using the methods of Section 2.1.5 is not possible, in principle, due to the unavoidable truncation of the Volterra series that results in correlated residuals. The severity of this kernel estimation bias depends on the size of the correlated residuals relative to the prediction of the truncated Volterra model. If more high-order terms are included in the truncated model in order to reduce the size of the residuals, then the estimation task becomes more cumbersome and often impractical. Although a practicable solution for this important problem has been recently proposed in the form of trainable network models of high-order systems (see Sections 2.3 and 4.3), this serious limitation gave initial impetus to the Wiener approach and its variants that were developed to overcome specific practical shortcomings of the initial Wiener methodology. The Wiener approach addresses the problem of biased kernel estimation in the general context of an infinite series expansion by decoupling the various Wiener kernels through orthogonalization of the respective Wiener functionals. Nonetheless, the resulting Wiener kernels are distinct from the Volterra kernels of the system and can be viewed as having a "structured bias" related to the employed GWN input, as discussed previously.

In this section, we present Wiener's original approach to kernel estimation in order to provide historical perspective and allow the reader to appreciate the many subtleties of this modeling problem, although Wiener's approach is not deemed the best choice at present, in light of recent developments that have made available powerful methodologies for the estimation of Volterra (not Wiener) models with superior performance in a practical context.

Following presentation of the rudiments of the Wiener approach, we will elaborate on its most popular implementation (the cross-correlation technique) in Section 2.2.3 and discuss its practically useful variants using quasi-white test inputs in Section 2.2.4. We note that the recommended and most promising estimation methods at present (using kernel expansions, iterative estimation techniques and equivalent network models) are presented in Sections 2.3 and 2.4, and they yield Volterra (not Wiener) models.

The orthogonality of the Wiener series allows decoupling of the Wiener functionals through covariance computations and estimation of the Wiener kernels from the GWN input and the

corresponding output data. Since the orthogonality of the Wiener functionals is independent of the specific kernel functions involved, a known "instrumental" Wiener functional can be used to isolate each term in the Wiener series (by computing its covariance with the system output) and subsequently obtain the corresponding kernel. For instance, if an m^{th}-order instrumental functional $Q_m\left[q_m; x(t'), t' \le t\right]$, constructed with a known kernel $q_m\left(\tau_1,...,\tau_m\right)$, is used to compute the covariance with the output signal $y(t)$, then:

$$\begin{aligned} E\left[y(t)Q_m(t)\right] &= \sum_{n=0}^{\infty} E\left[G_n(t)Q_m(t)\right] \\ &= E\left[G_m(t)Q_m(t)\right] \\ &= m!P^m \int_0^{\infty}\!\!\ldots\!\int_0^{\infty} h_m\left(\tau_1,...,\tau_m\right) q_m\left(\tau_1,...,\tau_m\right) d\tau_1 \ldots d\tau_m \end{aligned} \tag{2.83}$$

since Q_n is orthogonal (i.e., it has zero covariance) with all G_n functionals for $m \neq n$. Note that the instrumental functional $Q_m(t)$ has the form of the mth-order Wiener functional given by Equation (2.57) with the kernel q_m replacing the kernel h_m; hence it can be computed for a given input signal $x(t)$.

The "instrumental" kernel $q_m(\tau_1,...\tau_m)$ is judiciously chosen in order to facilitate the evaluation of the unknown kernel $h_m(\tau_1,...\tau_m)$, after the left-hand-side of Equation (2.83) is evaluated from input-output data. Wiener suggested the use of a multi-dimensional orthonormal basis for defining the instrumental kernels. So, if $\left\{b_j(\tau)\right\}$ is a complete orthonormal (CON) basis over the range of system memory, $\tau \in [0, \mu]$, then instrumental kernels of the form:

$$q_m(\tau_1,...,\tau_m) = b_{j_1}(\tau_1)...b_{j_m}(\tau_m) \tag{2.84}$$

can be used to obtain the expansion coefficients $\left\{a_{j_1,...,j_m}\right\}$ of the unknown kernel over the specified CON basis as:

$$a_{j_1,...,j_m} = \frac{1}{m!P^m} E\left[y(t)Q_m(t)\right] \tag{2.85}$$

where,

$$h_m(\tau_1,...,\tau_m) = \sum_{j_1}\ldots\sum_{j_m} a_{j_1,\ldots j_m} b_{j_1}(\tau_1)...b_{j_m}(\tau_m) \tag{2.86}$$

Note that in this case:

$$Q_m(t) = \sum_{l=0}^{[m/2]} \frac{(-1)^l P^l m!}{(m-2l)!\,l!\,2^l} v_{j_1}(t)...v_{j_{m-2l}}(t)\delta_{j_{m-2l+1},j_{m-2l+2}}...\delta_{j_{m-1},j_m} \tag{2.87}$$

where,

$$v_j(t) = \int_0^{\mu} b_j(\tau)x(t-\tau)d\tau \tag{2.88}$$

and μ and $\delta_{i,j}$ denote the system memory and the Kronecker delta[3] respectively. Since the input $x(t)$ is GWN and the basis functions $\{b_j\}$ are orthonormal, the signals $v_j(t)$ are independent Gaussian (non-white) random processes with zero mean and variance P [Marmarelis, 1979b]. Therefore, the instrumental functionals $\{Q_m\}$ can be seen as orthogonal multinomials in the variables $(v_1,...,v_m)$, with a structure akin to multi-variate Hermite polynomials.

Motivated by this observation, Wiener proposed a general model (for the Wiener class of systems) comprised of a known set of parallel linear filters with impulse response functions $\{b_j(\tau)\}$ (i.e., comprising an orthonormal complete basis of functions, such as the Laguerre set) receiving the GWN input signal and feeding the filter-bank outputs into a multi-input static nonlinearity, $y = f(v_1, v_2, ...)$, that he decomposed into the cascade of a known "Hermite-polynomial" component and an unknown "coefficients" component to be determined from the data, as shown in Figure 2.9. The latter decomposition was viewed as an efficient way of implementing the mathematical structure of the functionals $\{Q_m\}$, although one may question this assertion of efficiency in light of the complexity introduced by the Hermite expansion. The system identification/modeling task then reduces to evaluating the "coefficients" component from input-output data for a GWN input, since all other components are known and fixed.

[3] The Kronecker delta $\delta_{i,j}$ is the discrete-time equivalent of the Dirac delta function (impulse), defined as 1 for $i = j$ and as zero elsewhere.

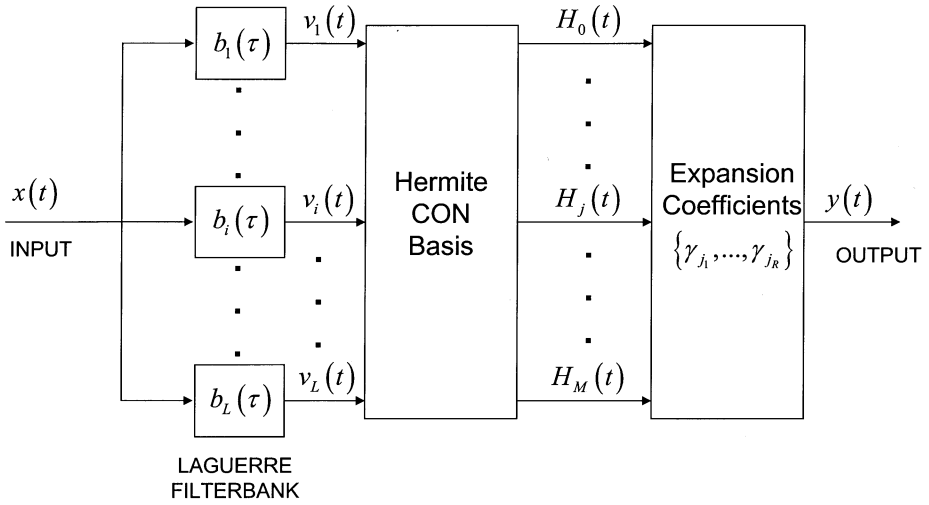


Figure 2.9
The block-structured Wiener model that is equivalent to the Wiener series for a GWN input when L and M tend to infinity. The suggested filterbank $\{b_i\}$ by Wiener was the complete othonormal (CON) Laguerre basis of functions. Since the Laguerre and Hermite bases are known (selected), the problem reduces to determining the expansion coefficients $\{\gamma_{j1},\ldots,\gamma_{jR}\}$ from input—output data when the input is GWN.

In the original Wiener formulation, the output values of the filterbank $\{v_i\}$ are viewed as variables completely describing the input signal from $-\infty$ up to the present time $[x(\tau);\tau \le t]$. Wiener chose the Laguerre set of functions to expand the past (and present) of the input signal because these functions form a complete orthonormal (CON) basis in the semi-infinite interval $[0,\infty)$ and have certain desirable mathematical properties (which will be described in Section 2.3.2). In addition, the outputs of the Laguerre "filterbank" can be easily generated in analog form by linear ladder R-C circuits (an important issue at that time).

When we employ L filters in the filterbank, we use L variables $\{v_i(t)\}$ to describe the past (and present) of the input signal at each time t. Thus, the system output can be considered a function of L variables and it can be expanded in terms of the CON basis of Hermite polynomials $\{H_j\}$ as:

$$y(t) = \lim_{\substack{M\to\infty \\ R\to\infty}} \sum_{j_1=0}^{M} \ldots \sum_{j_R=0}^{M} \gamma_{j_i \ldots j_R} H_{j_1}(v_1) \ldots H_{j_R}(v_L) + h_0 \qquad (2.89)$$

Clearly, in practice, both M and R must be finite and determine the maximum order of nonlinearity approximated by the finite-order model, where M is the maximum order of Hermite polynomials used in the expansion. Note that the nonlinear order r is defined by the sum of the indices: $(j_1 + \ldots + j_R)$. The Hermite polynomial of j th order is given by:

$$H_j(v) = e^{-\beta v^2} \frac{d^j}{dv^j}\left[e^{\beta v^2} \right] \tag{2.90}$$

where the parameter β determines the Gaussian weighting function $\exp\left[-2\beta v^2\right]$ that defines the orthogonality of the Hermite polynomials as:

$$\int_{-\infty}^{\infty} e^{-2\beta v^2} H_{j_1}(v) H_{j_2}(v)\, dv = \delta_{j_1, j_2} \tag{2.91}$$

The proper selection of the parameter β is important in practice, because it determines the convergence of the Hermite expansion of the static nonlinearity $f(v_1,...,v_L)$ in conjunction with the GWN input power level P, since the variance of each $v_i(t)$ process is P.

For any GWN input, the terms of the multi-dimensional Hermite expansion in Equation (2.89) are statistically orthogonal (i.e., have zero covariance) and are normalized to unity Euclidean norm because the joint PDF of the $\{v_i\}$ processes has the same form as the Hermite weighting function (i.e., multi-variate Gaussian). Therefore, the expansion coefficients in Equation (2.89) can be evaluated through the ensemble average:

$$\gamma_{i_1 \ldots i_R} = E\left\{ \left[y(t) - h_0 \right] H_{i_1}\left[v_1(t) \right] \ldots H_{i_R}\left[v_R(t) \right] \right\} \tag{2.92}$$

where all terms of the expansion of Equation (2.89) with indices $(j_i,..., j_R)$ distinct from the indices $(i_i,...,i_R)$ vanish. The indices i_1 through i_R take values from 0 to M and add up to the order of the estimated Wiener functional component. Note that the multi-index subscript ($i_1 \ldots i_R$) is ordered (i.e., it is non-permutable) and all possible combinations of L functions $\{v_i\}$ taken by R must be considered in the expansion of Equation (2.89). The mean h_0 of the output $y(t)$ must be subtracted in Equation (2.92) because it is separated out in Equation (2.89).

According to the Wiener approach, the coefficients $\gamma_{i_1 \ldots i_R}$ characterize the system completely and the identification problem reduces to the problem of determining these coefficients through the averaging operation indicated in Equation (2.92). The ensemble average of Equation (2.92) can be implemented by time-averaging in practice, due to the ergodicity and stationarity of the input-output processes. Once these coefficients have been determined, they can be used to synthesize (predict) the output of the nonlinear model for any given input, according to Equation (2.89). Of course, the output prediction for any given input will be accurate only if the model is (nearly) complete, as discussed earlier.

This approach, which was deciphered for Wiener's MIT colleagues in the 50's by Amar Bose (see Historical Note #2), is difficult to apply to physiological systems in a practical context for the following reasons:

(a) The form of the output expression is alienating to many physiologists, because it is difficult to assign some physiological meaning to the characterizing coefficients that would reveal some functional features of the system under study.

(b) The experimentation and computing time required for the evaluation of the characterizing coefficients is long, because long data-records are required in general for reducing the variance of the estimates down to acceptable levels.

For these reasons, this original Wiener approach has been viewed by most investigators as impractical, and has not found any applications to physiology in the originally proposed form. However, variants of this approach have found many applications, primarily by means of the cross-correlation technique discussed in Section 2.2.3 and multinomial expansions of the multi-input static nonlinearity in terms of the orthogonal polynomials $\{Q_m\}$ given by Equation (2.87) to yield estimates of the expansion coefficients $\{a_{j_1,j_2,\ldots}\}$ of the Wiener kernels, as described by Equation (2.85) [Marmarelis, 1987].

To avoid the complexity introduced by the Hermite expansion, Bose proposed the use of an orthogonal class of functions that he called "gate" functions, which are simply square unit pulses that are used to partition the output function space into non-overlapping cells (hence orthogonality of the gate functions) [Bose, 1956]. This formulation is conceptually simple and appears suitable for systems that have strong saturating elements. Nonetheless, it has found very limited applications, due to the still cumbersome model form and the demanding requirement for long input-output data records.

It must be emphasized that the main contribution of Wiener's formulation is in suggesting the decomposition of the general nonlinear model into a linear filter-bank (using a complete set of filters that span the system functional space) and a multi-input static nonlinearity receiving the outputs of the filter-bank and producing the system output (see Figure 2.9). This is a powerful statement in terms of nonlinear dynamic system modeling, because it separates the dynamics (the filter-bank stage) from the nonlinearities and reduces the latter to static form that is much easier to represent/estimate, for any given application.

In the many possible variants of the Wiener approach, different orthogonal or non-orthogonal bases of functions can be used both for the linear filter-bank and the static nonlinearity. We have found that the Laguerre basis (in discrete time) is a good choice for filter-bank in general, as discussed in Section

2.3.2. We have also found that polynomial nonlinearities are good choices in the general case, although combinations with specialized forms (e.g., sigmoidal) may also be suitable in certain cases. These important issues are discussed further in Section 2.3 and constitute the present state of the art, in connection with iterative (gradient-based) estimation methods and equivalent network structures (see Section 4.2).

It should be noted that a general, yet rudimentary, approach in discretized, input-output space (having common characteristics with the Bose approach), may utilize a grid of the discrete input values that cover the memory of the system and the dynamic range of amplitudes of the input. At any discrete time t, the present and past values of the input are described by a vector of real numbers $(x_0, x_1, ..., x_M)$ that can be put in correspondence with the value, y_0, of the system output at this time, thus forming the "mapping" input-output vector $(x_0, x_1, ..., x_M, y_0)$. As the system is being tested with an ergodic input (e.g., white noise), input-output vectors are formed until the system is densely tested by combinations of values of the input vectors. All these input-output vectors define an input-output mapping that represents a "digital model" of the system in a most rudimentary form.

In the next two sections, we complete the traditional approaches to Wiener kernel estimation that have found many applications to date but have seen their utility eclipsed by more recent methodologies presented in Sections 2.3 and 4.2.

2.2.3. The Cross-Correlation Technique for Wiener Kernel Estimation

Lee and Schetzen (1965) proposed a different implementation of Wiener's original idea for kernel estimation that has been widely used because of its relative simplicity. The Lee and Schetzen method, termed the "cross-correlation technique", is based on the observation that the product of m time-shifted versions of the GWN input can be written in the form of the leading term of the m^{th}-order Wiener functional using delta functions:

$$x(t-\tau_1)...x(t-\tau_m) = \int_0^\infty ... \int_0^\infty \delta(\tau_1-\lambda_1)...\delta(\tau_m-\lambda_m)\; x(t-\lambda_1)...x(t-\lambda_m) d\lambda_1...d\lambda_m \qquad (2.93)$$

The expression of Equation (2.93) has the form of a homogeneous (Volterra) functional of m th-order and, therefore, it is orthogonal to all Wiener functionals of higher order. Based on this observation, they were able to show that, using this product as the leading term of an "instrumental functional" in connection with Equation (2.83), the Wiener kernel estimation is possible through input-output cross-

correlations of the respective order. The resulting expression for the estimation of the mth-order Wiener kernel is:

$$h_m(\tau_1,...,\tau_m) = \frac{1}{m!P^m} E\left[y_m(t)\, x(t-\tau_1)...x(t-\tau_m) \right] \tag{2.94}$$

where $y_m(t)$ is the m^{th}-order output residual defined by the expression:

$$y_m(t) = y(t) - \sum_{n=0}^{m-1} G_n(t) \tag{2.95}$$

The use of the output residual in the cross-correlation formula of Equation (2.94) is necessitated by the fact that the input-output expression of Equation (2.93), having the form of a homogeneous (Volterra) functional of mth-order, is orthogonal to all higher order Wiener functionals but not to the lower order ones whose contributions must be subtracted. It is seen later that this subtraction is required only by the lower order terms of the same parity (odd/even) in principle. Failure to use the output residual in Equation (2.94) leads to severe misestimation of the Wiener kernels at the diagonal values giving rise to impulse-like errors along the kernel diagonals. In practice, this output residual is computed on the basis of the previously estimated lower-order Wiener kernels and functionals. Thus, the use of the output residual in Equation (2.94) implies the application of the cross-correlation technique in ascending order of Wiener kernel estimation.

The statistical ensemble average denoted by the "expected value" operator $E\,\cdot$ in Equation (2.94) can be replaced in practice by time-averaging over the length of the data record, assuming stationarity of the system. Since the data record is finite, these time-averages form with certain statistical variance (i.e., they are not precise). This variance depends on the record length and the GWN input power level, in addition to the ambient noise and the system characteristics, as detailed in Section 2.4.2. At the risk of becoming somewhat pedantic, we detail below the successive steps in the actual implementation of the cross-correlation technique for Wiener kernel estimation [Marmarelis & Marmarelis, 1978; Schetzen 1980].

Estimation of h_0

The expected value of each Wiener functional $G_n[h_n; x(t)]$ (for $n \geq 1$) is zero if $x(t)$ is GWN, since the Wiener functionals for $n \geq 1$ are constructed orthogonal to the constant zeroth-order Wiener functional h_0. Therefore, taking the expected value of the output signal $y(t)$ yields:

$$E\ y(t)\ = h_0 \tag{2.96}$$

which indicates that h_0 is the ensemble average (mean) or the time average of the output signal for a GWN input.

Estimation of $h_1(\tau)$

The shifted input $x(t-\sigma)$ is a first-order homogeneous (Volterra) functional according to Equation (2.93) where it is written as a convolution of the GWN input with a delta function. Since $x(t-\sigma)$ can be also viewed as a Wiener functional of first order (no other terms are required in the first-order case), its covariance with any other order of Wiener functional will be zero. Thus:

$$E\left[y(t)\,x(t-\sigma)\right] = E\left[x(t-\sigma)\int_0^\infty h_1(\tau)\,x(t-\tau)\,d\tau\right]$$

$$= \int_0^\infty h_1(\tau) E\left[x(t-\sigma)\,x(t-\tau)\right] d\tau$$

$$= \int_0^\infty h_1(\tau) P\delta(\tau-\sigma)\,d\tau \qquad (2.97)$$

since the second-order autocorrelation function of GWN is a delta function with strength P. Therefore, the first-order Wiener kernel is given by the cross-correlation between the GWN input and the respective output, normalized by the input power level P:

$$h_1(\sigma) = (1/P)\,E\left[y(t)\,x(t-\sigma)\right] \qquad (2.98)$$

Note that the cross-correlation formula (2.98) for the estimation of h_1 does not require in principle the use of the 1st-order output residual prescribed by Equation (2.94). The reason is the parity (odd/even) separation of the homogeneous functionals (and, by extension, of the Wiener functionals) due to the fact that the odd-order autocorrelation functions of GWN are uniformly zero (see Appendix II). Nonetheless, since the orthogonality between Wiener functionals is only approximate in practice due to the finite data records, it is advisable to use always the output residual as prescribed by Equation (2.94). This implies subtraction of the previously estimated h_0 from $y(t)$ prior to cross-correlation with $x(t-\sigma)$ for the practical estimation of $h_1(\sigma)$.

Estimation of $h_2(\tau_1,\tau_2)$

Since $x(t-\sigma_1)x(t-\sigma_2)$ is a 2nd-order homogeneous (Volterra) functional of $x(t)$ (see Equation (2.93)), it is orthogonal to all Wiener functionals of higher order, i.e., $E\left[G_i(t)x(t-\sigma_1)x(t-\sigma_2)\right]=0$ for $i>2$. Thus, the cross-correlation between $\left[x(t-\sigma_1)x(t-\sigma_2)\right]$ and the output $y(t)$ eliminates the contributions of all Wiener functionals, except G_0, G_1, and G_2. Furthermore:

$$E\ \ G_0x(t-\sigma_1)x(t-\sigma_2)\ \ =h_0P\delta(\sigma_1-\sigma_2) \tag{2.99}$$

indicating that the estimate of h_0 ought to be subtracted from $y(t)$ prior to cross-correlation, and:

$$E\left[G_1(t)\,x(t-\sigma_1)\,x(t-\sigma_2)\right]=\int_0^\infty h_1(\tau)\,E\left[x(t-\sigma_1)\,x(t-\sigma_2)\,x(t-\tau)\right]d\tau=0 \tag{2.100}$$

since all the odd-order autocorrelation functions of GWN are zero (see Appendix II). For the second-order Wiener functional, we have:

$$E\left[G_2(t)\,x(t-\sigma_1)\,x(t-\sigma_2)\right]$$

$$=P^2\int_0^\infty\int h_2(\tau_1,\tau_2)\left[\delta(\tau_1-\tau_2)\delta(\sigma_1-\sigma_2)+\delta(\tau_1-\sigma_1)\delta(\tau_2-\sigma_2)+\delta(\tau_2-\sigma_1)\delta(\tau_1-\sigma_2)\right]d\tau_1d\tau_2-P^2\delta(\sigma_1-\sigma_2)\int_0^\infty h_2(\tau_1,\tau_1)d\tau_1$$

$$=2P^2h_2(\sigma_1,\sigma_2) \tag{2.101}$$

using the Gaussian decomposition property (see Appendix II) and the symmetry of the second-order kernel. Thus, the cross-correlation between the output $y(t)$ and $\left[x(t-\sigma_1)x(t-\sigma_2)\right]$ yields:

$$E\left[y(t)\,x(t-\sigma_1)\,x(t-\sigma_2)\right]=Ph_0\delta(\sigma_1-\sigma_2)+2P^2h_2(\sigma_1,\sigma_2) \tag{2.102}$$

which demonstrates the previously stated fact of impulse-like estimation errors along the kernel diagonals when the output residual is not used in the cross-correlation formula of Equation (2.94). This example also demonstrates the fact that there is an odd/even separation of the Wiener functionals (i.e., h_1 is not present in Equation (2.102) but h_0 is). In order to eliminate the contribution of h_0 to this second-order cross-correlation, it suffices in practice to subtract from the output signal $y(t)$ the previously estimated value of h_0 (which is the average value of the output signal). However, it is generally advisable to subtract all lower order functional contributions from the output signal, as prescribed by Equation (2.95), because the theoretical orthogonality between $\left[x(t-\sigma_1)x(t-\sigma_2)\right]$ and $G_1\left[h_1;x(t)\right]$ is only approximate for finite data-records in practice. Therefore, in order to minimize the

estimation errors due to the finite data records, we cross-correlate the 2nd-order output residual (the "hats" denote estimates of the Wiener kernels):

$$y_2(t) = y(t) - h_0 - \int_0^\infty h_1(\tau)\, x(t-\tau) d\tau \tag{2.103}$$

with $\left[x(t-\tau_1)x(t-\sigma_2)\right]$ to obtain the 2nd-order Wiener kernel estimate as:

$$h_2(\sigma_1, \sigma_2) = \left(1/2P^2\right) E\left[y_2(t)\, x(t-\sigma_1)\, x(t-\sigma_2)\right] \tag{2.104}$$

It should be noted that a possible error Δh_0 in the estimate of h_0 used for the computation of the output residual causes some error Δh_2 at the diagonal points of the 2nd-order Wiener kernel estimate that takes the form of a delta function along the diagonal:

$$\Delta h_2(\sigma_1, \sigma_2) = \left(1/2P^2\right) \Delta h_0 \delta(\sigma_1 - \sigma_2) \tag{2.105}$$

Estimation of $h_3(\tau_1, \tau_2, \tau_3)$

Following the same reasoning as for h_2, we first compute the third-order output residual $y_3(t)$:

$$y_3(t) = y(t) - G_2(t) - G_1(t) - G_0 \tag{2.106}$$

using the previously estimated $h_0, h_1,$ and h_2 (even though the subtraction of G_2 and G_0 is not theoretically required due to the odd/even separation), and then we estimate the third-order Wiener kernel by computing the third-order cross-correlation:

$$h_3(\sigma_1, \sigma_2, \sigma_3) = \left(1/6P^3\right) E\left[y_3(t)\, x(t-\sigma_1)\, x(t-\sigma_2)\, x(t-\sigma_3)\right] \tag{2.107}$$

It must be noted that any imprecision Δh_1 in the estimation of h_1, that is used for the computation of the output residual, will cause some error Δh_3 along the diagonals of the h_3 estimate that will have the impulse form:

$$\Delta h_3(\sigma_1, \sigma_2, \sigma_3) = (1/6P)\left[\Delta h_1(\sigma_1)\delta(\sigma_2 - \sigma_3) + \Delta h_1(\sigma_2)\delta(\sigma_3 - \sigma_1) + \Delta h_1(\sigma_3)\delta(\sigma_1 - \sigma_2)\right] \tag{2.108}$$

because:

$$E\left[G_1(t)\, x(t-\sigma_1)\, x(t-\sigma_2)\, x(t-\sigma_3)\right] = P^2\left[h_1(\sigma_1)\delta(\sigma_2 - \sigma_3) + h_1(\sigma_2)\delta(\sigma_3 - \sigma_1) + h_1(\sigma_3)\delta(\sigma_1 - \sigma_2)\right] \tag{2.109}$$

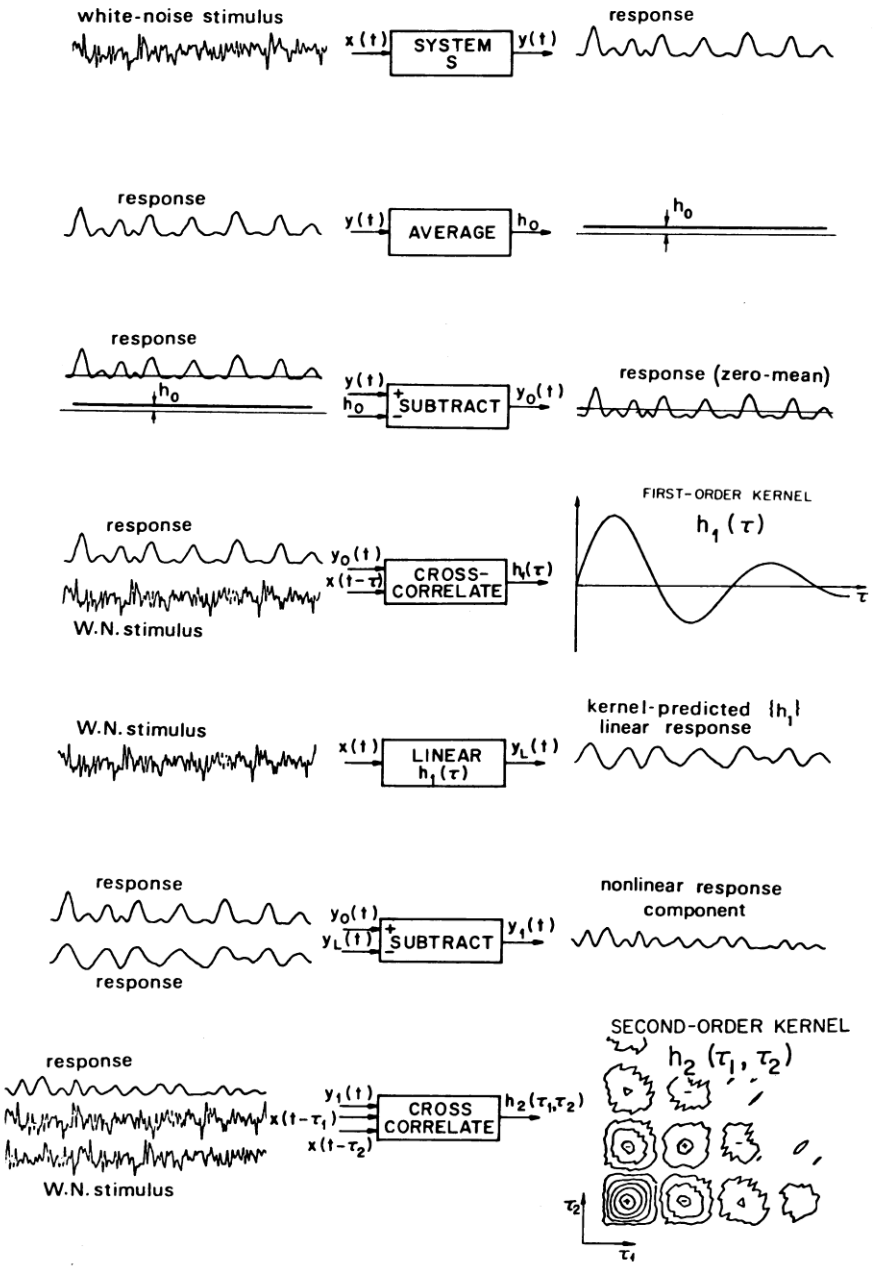


Figure 2.10
Illustration of the successive steps for the estimation of zero-, first- and second-order Wiener kernels via the cross-correlation technique [Marmarelis & Marmarelis, 1978].

The successive steps of Wiener kernel estimation using the cross-correlation technique are illustrated in Figure 2.10. The aforementioned errors due to the finite data records (see Equations (2.105) and (2.108)) occur along the diagonals, but additional estimation errors occur throughout the kernel space for a variety of reasons detailed in Section 2.4.2 and become more severe as the order of kernel estimation increases. A complete analysis of estimation errors associated with the cross-correlation technique and their dependence on the input and system characteristics is given in Section 2.4.2.

The computation of the output residuals in ascending order also provides in practice a measure of model adequacy at each order of Wiener kernel estimation. This measure of model adequacy is typically the normalized mean-square error (NMSE) Q_r of the r th-order output prediction, defined as the ratio of the variance of the r th-order output residual $y_r(t)$ to the output variance (note that the 1st-order output residual is simply the de-meaned output):

$$Q_r = E\left[y_r^2(t)\right] \Big/ E\left[y_1^2(t)\right] \tag{2.110}$$

The NMSE measure Q_r lies between 0 and 1 (for $r \geq 1$) and quantifies the portion of the output signal power that is not predicted by the model. If this quantity Q_r drops below a selected critical value (e.g., a value $Q_r = 0.1$ would correspond to 10% NMSE of the r th-order model prediction, indicating that 90% of the output signal power is explained/predicted by the model) then the kernel estimation procedure can be terminated and a truncated Wiener model of r th order is obtained. Obviously, the selection of this critical NMSE value depends on the specific requirements of each application and on the prevailing signal-to-noise ratio (that provides a lower bound for this value). In applications to date, this value has ranged from 0.02 to 0.20.

As a practical matter, the actual estimation of Wiener kernels has been limited to 2nd-order to date (with rare occasions of 3rd-order Wiener kernel estimation but no further) due to the multi-dimensional structure of high-order kernels. This is dictated by practical limitations of data-record length, computational burden and estimation accuracy that become rather severe for multi-dimensional high-order kernels.

Some Practical Considerations

Wiener kernel estimation through the cross-correlation technique has several advantages over the original Wiener formulation, because: (1) it directly estimates the Wiener kernels, which can reveal interpretable features of the system under study; (2) it is much simpler computationally, since it does not involve the concatenated Laguerre and Hermite expansions.

Because of its apparent simplicity, the cross-correlation technique has found many applications in physiological system modeling and fomented the initial interest in the Wiener approach. Nonetheless, the application of the original cross-correlation technique exposed a host of practical shortcomings that were addressed over time by various investigators. These shortcomings concerned primarily issues of

estimation errors regarding the dependence of estimation variance on the input length and bandwidth, as well as the generation and application of physically realizable approximate GWN inputs. This prompted the introduction of a broad class of quasi-white test input signals that address some of these applicability issues, as discussed in Section 2.2.4.

Among the important practical issues that had to be explored in actual applications of the cross-correlation technique are: the generation of appropriate quasi-white test signals (since ideal GWN is not physically realizable); the choice of input bandwidth; the accuracy of the obtained kernel estimates as a function of input bandwidth and record length; the effect of extraneous noise and experimental transducer errors. A comprehensive and detailed study of these practical issues was first presented in Marmarelis & Marmarelis (1978). A summary of the types of estimation errors associated with the cross-correlation technique is provided in Section 2.4.2, since the cross-correlation technique is a "legacy methodology" but not the best choice at present time.

It is evident that, in actual applications, the ideal GWN input has to be approximated in terms of practical limitations on bandwidth and amplitude. Since the latter are both infinite in the ideal GWN case, we have to accept in practice a band-limited and amplitude-truncated GWN input.

In light of these inevitable approximations, non-Gaussian quasi-white input signals were introduced that exhibited whiteness within a given finite bandwidth and remained within a specified finite amplitude range. In addition, these quasi-white test signals were easy to generate and offered computational advantages in certain cases (e.g., binary or ternary signals). A broad family of such quasi-white test input signals was introduced in 1975 by the author during his Ph.D. studies. These signals can be used in connection with the cross-correlation technique for Wiener kernel estimation and give rise to their own orthogonal functional series, as discussed in Section 2.2.4.

We must note that a fair amount of variance is introduced into the Wiener kernel estimates (obtained via cross-correlation) because of the stochastic nature of the (quasi-) white noise input. This has prompted the use of pseudorandom m-sequences which, however, present some problems in their high-order autocorrelation functions (see Section 2.2.4). The use of deterministic inputs (such as multiple impulses or sums of sinusoids of incommensurate frequencies) alleviates this problem but places us in the context of Volterra kernels, as discussed in Section 2.1.5.

An interesting estimation method, that reduces the estimation variance for arbitrary stochastic inputs, is based on exact orthogonalization of the expansion terms for the given input data record [Korenberg, 1988]. However, this method (and any other method based on least-squares fitting for

arbitrary input signals) yields a hybrid between the Wiener and the Volterra kernels of the system, that depends on the spectral characteristics of the specific input signal. This hybrid can be viewed as a biased estimate of the Wiener and/or Volterra kernels, if significant higher order terms exist beyond the order of the estimated truncated model.

Another obstacle in the broader use of the cross-correlation technique has been the heavy computational burden associated with the estimation of *high-order* kernels. The amount of required computations increases geometrically with the order of estimated kernel, since the estimated kernel values are roughly proportional to M^Q where Q is the kernel order and M is the kernel memory-bandwidth product. This prevents the practical estimation of kernels above a certain order, depending on the kernel memory-bandwidth product (i.e., the number of sample values per kernel dimension). An additional practical limitation is imposed by the fact that the kernels with more than three dimensions are difficult to represent, inspect or interpret meaningfully. As a result, successful application of the cross-correlation technique has been limited to weakly nonlinear systems (typically second-order or rarely third-order) to date.

Illustrative Example

To illustrate the application of the cross-correlation technique on a real system, we present below one of its first successful applications to a physiological system that transforms light intensity variations, impinging upon the catfish retina, into a variable ganglion cell discharge rate [Marmarelis & Naka, 1972,1973b]. The latter is measured by superimposing multiple spike-train responses (sequences of action potentials generated by the ganglion cell) to the same (repeated) light-intensity stimulus signal. The physical stimulus is monochromatic light intensity modulated in band-limited GWN fashion around a reference level of mean light intensity (light-adapted preparation). The response is the extracellularly recorded ganglion cell response (spike trains superimposed and binned to yield a measure of instantaneous firing frequency). The stimulus-response data are processed using the cross-correlation technique to estimate the first-order and second-order Wiener kernels of the light-to-ganglion cell system shown in Figure 2.11.

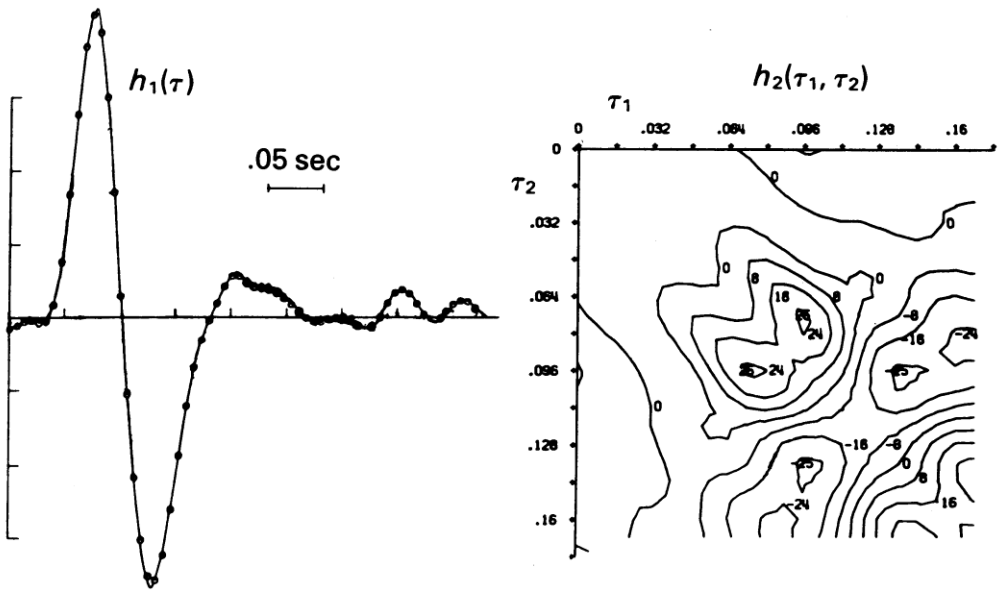


Figure 2.11
The Wiener kernel estimates of first-order (left) and second-order (right) obtained via the cross-correlation technique for the light-to-ganglion cell system in the catfish retina [Marmarelis & Naka, 1973b].

Frequency-Domain Estimation of Wiener Kernels

Wiener kernel estimation is also possible in the frequency domain through the evaluation of high-order cross-spectra [Brillinger, 1970; French & Butz, 1974; Barker & Davy, 1975; French, 1976] which represent the frequency-domain counterparts of the high-order cross-correlations. The problem of estimation variance due to the stochastic nature of the GWN input persists in this approach as well.

The efficient implementation of the frequency-domain approach calls for the use of Fast Fourier Transforms (FFT) with a data-length size equal to the smallest power of 2 that is greater than or equal to twice the memory-bandwidth product of the system. Thus, if the FFT size is M (a power of two by algorithmic necessity), then the input-output data record is segmented into K contiguous segments of M data points each. An estimate of the r th-order cross-spectrum:

$$S_{r,i}(\omega_1,...,\omega_r) = Y_{r,i}(\omega_1+...+\omega_r)\,X_i^*(\omega_1)...X_i^*(\omega_r) \tag{2.111}$$

can be obtained from the i th segment, where $X_i^*(\omega)$ denotes the FFT conjugate of the i th input segment and $Y_{r,i}(\omega)$ denotes the FFT of the corresponding i th segment of the r th output residual. Then, the r th-order Wiener kernel can be estimated in the frequency-domain by averaging the various segment estimates of Equation (2.111):

$$H_r(\omega_1,...,\omega_r) = \frac{1}{r!P^rK}\sum_{i=1}^{K} S_{r,i}(\omega_1,...,\omega_r) \tag{2.112}$$

where P is the input power level and (KM) is the total number of input-output data points. The r th-order Wiener kernel estimate can be converted into the time domain via an r -dimensional inverse FFT.

This approach offers some computational advantages over direct cross-correlation (especially for systems with large memory-bandwidth product) but exhibits the same vulnerabilities in terms of kernel estimation errors.

2.2.4. Quasi-White Test Inputs

The aforementioned quasi-white random signals, that can be used as test inputs in the context of Wiener modeling, have been termed "Constant-switching-pace Symmetric Random Signals" (CSRS) and they are defined by the following generation procedure [Marmarelis, 1975,1977]:

(1) select the bandwidth of interest, B_x (in Hz), and the desirable finite-range symmetric amplitude distribution (with zero-mean), $p(x)$, for the discrete-time CSRS input signal *x(n);*

(2) draw an independent sample *x(n)* at every time step $\Delta t = 1/(2B_x)$ (in sec) from a random number generator according to the probability distribution *p(x).*

It has been shown [Marmarelis, 1975,1977] that a quasi-white signal thus generated possesses the appropriate autocorrelation properties of all orders that allow its use for Wiener-type modeling of nonlinear systems with bandwidths: $B \le 1/(2\Delta t)$, via the cross-correlation technique.

In analog form, the CSRS remains constant at the selected value, *x(n),* for the duration of each interval Δt (i.e., for $n\Delta t \le t < (n+1)\Delta t$) and switches to the statistically independent value $x(n+1)$, where it stays constant until the next switching time $t = (n+2)\Delta t$. The fundamental time interval Δt is called the "step" of the CSRS, and it directly determines the bandwidth of the signal, within which whiteness is secured: $B_x = 1/(2\Delta t)$.

It has been shown that the statistical independence of any two steps of a CSRS is sufficient to guarantee its whiteness within the bandwidth determined by the specified step size Δt. The symmetric probability density function (PDF) $p(x)$ with which a CSRS is generated has zero mean, and, consequently, all its odd-order moments are zero. The even-order moments of $p(x)$ are non-zero and

finite, because the CSRS has finite amplitude range. The power level of the CSRS is ($M_2 \cdot \Delta t$), where M_2 is the second moment of $p(x)$--also the variance since $p(x)$ has zero mean.

This random quasi-white signal approaches the ideal white-noise process as Δt tends to zero. Note that in order to maintain constant CSRS power level as Δt decreases, the signal amplitude must be divided by $\sqrt{\Delta t}$. This implies that the distribution of the asymptotic white-noise process as $\Delta t \to 0$ is $\sqrt{\Delta t}\, p\left(x \cdot \sqrt{\Delta t}\right)$ and has infinite variance. For instance, if the standardized normal/Gaussian distribution is used to generate the CSRS (zero-mean and unit variance), then the asymptotic PDF is: $\lim_{\Delta t \to 0} \sqrt{\Delta t/(2\pi)} \exp\left[-x^2 \Delta t/2\right]$, having variance equal to $1/\Delta t$.

We note that every member of the CSRS family is by construction a stationary and ergodic process. For a given step size Δt and a proper amplitude PDF $p(x)$, an ensemble of random processes is defined within the CSRS family. Because of the ergodicity and stationarity of the CSRS, the ensemble-through and time-through statistics of the signal are invariant over the ensemble and over time respectively. This allows us to define its autocorrelation functions in both the temporal and the statistical (ensemble) sense.

In practice, we have finite-length signals and we are practically restricted to obtaining only estimates of the autocorrelation functions, usually by time averaging over the record length R, as:

$$\hat{\phi}_n(\tau_1,...,\tau_{n-1}) = \frac{1}{R-T_m} \int_{T_m}^{R} x(t)\, x(t-\tau_1)...x(t-\tau_{n-1})\, dt \tag{2.113}$$

where $T_m = \max\{\tau_1,...,\tau_{n-1}\}$ and stationarity has suppressed one tau argument (e.g., no shift for the first term of the product). For discrete-time (sampled) data, the integral of Equation (2.113) becomes a summation.

The estimate $\hat{\phi}_n(\tau_1,...,\tau_{n-1})$ is a random variable itself and its statistical properties must be studied in order to achieve an understanding of the statistical properties of the kernel estimates obtained by the cross-correlation technique. It was found that the expected value of $\hat{\phi}_n(\tau_1,...,\tau_{n-1})$ is $\phi_n(\tau_1,...,\tau_{n-1})$, which makes it an unbiased estimate, and the variance of $\hat{\phi}_n(\tau_1,...,\tau_{n-1})$ at all points tends to zero asymptotically with increasing record length R, which makes it a consistent estimate [Marmarelis, 1977].

It has been shown [Marmarelis, 1975,1977] that the autocorrelation functions of a CSRS are those of a quasi-white signal; viz., the odd-order autocorrelation functions are uniformly zero, while the even-

order ones are zero everywhere except at the diagonal strips. Note that the diagonal strips are areas within $\pm\Delta t$ (the step size of a CSRS) around every *full diagonal* of the argument space (i.e., where all the arguments form pairs of identical values).

If we visualize the autocorrelation function estimate as a surface in n-dimensional space, then $\hat{\phi}_n(\tau_1,...,\tau_{n-1})$ appears to have apexes corresponding to the *nodal* points of the n-dimensional space (i.e., the points with coordinates that are multiples of Δt). These apexes are connected with n-dimensional surface segments which are of first degree (planar) with respect to each argument τ_i. The direct implication of this morphology is that the "extrema" of the surface $\hat{\phi}_n(\tau_1,...,\tau_{n-1})$ must be sought among its apexes (i.e., among the nodal points of the argument space). This simplifies the analysis of this multi-dimensional surface that determines the statistical characteristics of the CSRS kernel estimates.

To illustrate this, we consider the second-order autocorrelation function of a CSRS:

$$\phi_2(\tau_1)=\begin{cases} M_2\left(1-\dfrac{|\tau_1|}{T}\right) & \text{for } |\tau_1|\le\Delta t \\ 0 & \text{for } |\tau_1|\ge\Delta t \end{cases} \tag{2.114}$$

shown in Figure 2.12 along with a cross-section of the fourth-order autocorrelation function. The quasi-white autocorrelation properties of the CSRS family are manifested by the impulse-like structure of their even-order autocorrelation functions and justify their use in kernel estimation via the cross-correlation technique. However, the kernel estimates that are obtained through the use of CSRS test inputs correspond to an orthogonal functional series that is slightly different in structure from the original Wiener series. This structural difference is due to the statistical properties of the CSRS, as expressed by the moments of its amplitude PDF, which are different in general from the moments of the Gaussian distribution (although the latter is included in the CSRS family).

The decomposition property of even products of Gaussian random variables (see Appendix II) results in great simplification of the expressions describing the orthogonal Wiener functionals. In the general case of a non-Gaussian CSRS, however, the decomposition property does not hold and complete description of the amplitude PDF may require several of its moments, which results in greater complexity of the form of the CSRS orthogonal functionals. Nevertheless, the construction of the CSRS functionals can be made routinely on the basis of an orthogonalization (Gram-Schmidt) procedure

similar to the one that was used in the construction of the Wiener series, under the assumption that the CSRS bandwidth is broader than the system bandwidth.

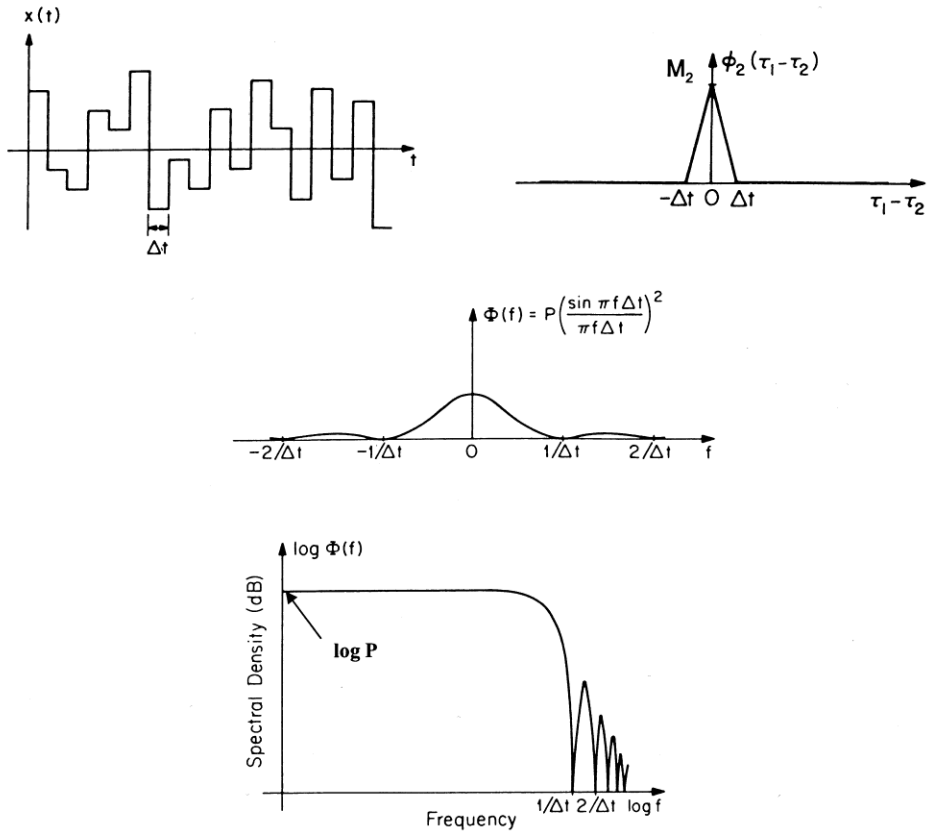


Figure 2.12
Portion of a CSRS quasi-white signal (top left), its second-order autocorrelation function (top right) and its power spectrum in linear and logarithmic scales (bottom) [Marmarelis & Marmarelis, 1978].

In the special case where a Gaussian amplitude PDF is chosen for the CSRS, the CSRS functional series takes the form of the Wiener series, where the power level of the quasi-white input is equal to the product of the second moment with the step size of the CSRS $\left(P = M_2 \Delta t\right)$. Thus, it becomes clear that the CSRS functional series is a more general orthogonal functional expansion than the Wiener series, extending the basic idea of orthogonal expansions using Volterra-type functionals throughout the space of symmetric probability distributions. The advantages of such a generalization are those that accrue in any optimization problem where the parameter space is augmented. This generality is achieved at the expense of more complexity in the expressions for the orthogonal functionals, if a non-Gaussian PDF is chosen.

Under the assumption that the CSRS bandwidth is broader than the system bandwidth, the first four orthogonal functionals $\{G_r^*\}$ that correspond to a CSRS quasi-white input take the form [Marmarelis, 1977]:

$$G_0^*[g_0; x(t'), t' \le t] = g_0 \tag{2.115}$$

$$G_1^*[g_1(\tau_1); x(t'), t' \le t] = \int_0^\infty g_1(\tau_1)\, x(t-\tau_1)\, d\tau_1 \tag{2.116}$$

$$G_2^*[g_2(\tau_1,\tau_2); x(t'), t' \le t] = \int_0^\infty \int_0^\infty g_2(\tau_1,\tau_2) x(t-\tau_1) x(t-\tau_2) d\tau_1 d\tau_2 - (M_2 \Delta t) \int_0^\infty g_2(\tau_1,\tau_1) d\tau_1 \tag{2.117}$$

$$G_3^*[g_3(\tau_1,\tau_2,\tau_3); x(t'), t' \le t] = \int_0^\infty \int_0^\infty \int_0^\infty g_3(\tau_1,\tau_2,\tau_3) x(t-\tau_1) x(t-\tau_2) x(t-\tau_3) d\tau_1 d\tau_2 d\tau_3$$

$$-3(M_2 \Delta t) \int_0^\infty \int_0^\infty g_3(\tau_1,\tau_2,\tau_2) x(t-\tau_1) d\tau_1 d\tau_2 - \left[\left(\frac{M_4}{M_2} - 3M_2 \right) \Delta t^2 \right] \int_0^\infty g_3(\tau_1,\tau_1,\tau_1) x(t-\tau_1) d\tau_1 \tag{2.118}$$

where $x(t)$ is a CSRS, Δt is its step size, $\{g_i\}$ are its associated CSRS kernels, and M_2, M_4, etc are the second, fourth, etc. moments of its amplitude PDF $p(x)$. It is evident that the deviation of the CSRS functionals (and associated kernels) from their Wiener counterparts diminishes as the CSRS amplitude distribution approaches the Gaussian profile, since then $M_4 = 3M_2^2$ and G_3^* attains exactly the form of a 3rd-order Wiener functional with power level $P = M_2 \Delta t$.

The expressions for higher-order functionals become quite complicated since they involve all the higher even moments of the CSRS input, but their derivation can be made routinely on the basis of a Gram-Schmidt orthogonalization procedure. Notice that the basic even/odd separation in the structural form of the CSRS functionals is the same as in the Wiener functionals, i.e., the odd- (even-) order functionals consist solely of all odd- (even-) order homogeneous functionals of equal and lower order.

The CSRS functionals should be viewed as slightly modified Wiener functionals. The integral terms (i.e., the homogeneous functionals) of the CSRS functionals that contain higher even moments (>2) contain also a higher power of Δt as a factor. This makes them significantly smaller than the terms containing only the second moment, since Δt attains small values. Therefore, the CSRS functionals become (for all practical purposes) the same as the Wiener functionals for very small Δt. This implies in turn that, whenever Δt is very small, the CSRS kernels are approximately the same as the Wiener

kernels except possibly at the diagonal points where the finite number of CSRS amplitude levels (e.g., binary, ternary, etc.) limits estimability, as discussed below.

The power spectrum of a CSRS with step size Δt and second moment M_2 is shown in Fig. 2.12 and is independent of the amplitude PDF. The bandwidth of the signal is inversely proportional to Δt, and it approaches the ideal white noise (infinite bandwidth) as Δt approaches zero (provided that the power level $P = M_2 \Delta t$ remains finite). The orthogonality of the CSRS functionals is satisfactory when the bandwidth of the CSRS exceeds the bandwidth of the system under study.

In order to estimate the r^{th}-order CSRS kernel, we cross-correlate the r^{th}-order output residual $y(t)$ with r time-shifted versions of the CSRS input (as in the GWN input case) and scale the outcome:

$$\hat{g}_r(\sigma_1,...,\sigma_r) = C_r E\left[y_r(t)\,x(t-\sigma_1)...x(t-\sigma_r)\right] \tag{2.119}$$

where:

$$y_r(t) = y(t) - \sum_{i=0}^{r-1} G_i^*\left[g_i(\tau_1,...,\tau_i); x(t'), t' \le t\right] \tag{2.120}$$

and C_r is the proper scaling factor that depends on the even moments and the step size of the CSRS, as well as the location of the estimated point in the kernel (e.g., diagonal vs. non-diagonal).

It was shown earlier that, in the case of GWN inputs, the scaling factor C_r is $\left(r!P^r\right)^{-1}$. However, in the case of CSRS inputs, the scaling factors differ for the diagonal and non-diagonal points of the kernels. For the non-diagonal points (i.e., when all σ_i are distinct) the scaling factor is the same as in the GWN case, where $P = M_2 \Delta t$. However, the determination of the appropriate scaling factors for the diagonal points involves higher even moments. For example, in the second-order case the scaling factor for the diagonal points $(\sigma_1 = \sigma_2)$ is found to be:

$$C_2 = \frac{1}{\left(M_4 - M_2^2\right)\Delta t^2} \tag{2.121}$$

CSRS and Volterra Kernels

It is instructive to examine the relation between the Volterra and the CSRS kernels of a system (at the non-diagonal points, for simplicity of expression) in order to demonstrate the dependence of the CSRS kernels upon the even moments and the step size of the specific CSRS that is used to estimate the kernels. Recall that the Volterra kernels of a system are independent of input characteristics, unlike the

Wiener or CSRS kernels that depend on the GWN or CSRS input characteristics (i.e., the power level or the step site and even moments respectively) [Marmarelis & Marmarelis, 1978]. The resulting expressions for the CSRS kernels in terms of the Volterra kernels $\{k_i\}$ of the system are more complicated than their Wiener counterparts given by Equation (2.57). The general expression for the non-diagonal points of the even-order CSRS kernels is:

$$g_{2n}(\sigma_1,...,\sigma_{2n}) = \sum_{m=n}^{\infty} C_{m,n}(P_1) \int_0^{\infty} ... \int_0^{\infty} k_{2n}(\tau_1,...,\tau_{2m-2n},\sigma_1,...,\sigma_{2n})\, d\tau_1 ... d\tau_{2m-2n}$$

$$+ \sum_{l=1}^{m-n} \Delta t^l D_{l,m,n}(P_1,...,P_{l+1}) \int_0^{\infty} ... \int_0^{\infty} k_{2m}(\tau_1,...,\tau_{2m-2n-2l},\sigma_1,...,\sigma_{2n}) d\tau_1 ... d\tau_{2m-2n-2l} \quad (2.122)$$

where P_l is a "generalized power level of l th-order" for CSRS inputs, defined as:

$$P_l = M_{2l} \Delta t^l \quad (2.123)$$

The function $C_{m,n}$ depends only on the conventional power level $P = P_1$, but the function $D_{l,m,n}$ is a rational expression of the generalized power levels. Note, however, that the terms of the second summation in the expression (2.122) are negligible in comparison with the terms of the first summation, since Δt is very small. Furthermore, the function $C_{m,n}(P_1)$ tends to the coefficient found in the relation between the Wiener and the Volterra kernels given by Equation (2.57) as Δt tends to zero. Hence, the CSRS kernels at the non-diagonal points are approximately the same as the Wiener kernels (as long as they have the same power level $P = M_2 \Delta t$). The only significant difference between the CSRS and the Wiener kernels of a system is found at some diagonal points whenever the CSRS attains a finite number of amplitude levels, as discussed below.

The Diagonal Estimability Problem

Every CSRS waveform attains (by construction) a finite number L of amplitude levels. This limits our ability to estimate via cross-correlation the kernel values at diagonal points that have dimension L or above. For instance, a binary CSRS input $x(t)$ attains the values of $\pm A$. Therefore, $x^2(t)$ is a constant A^2 for this binary CSRS input and the cross-correlation formula of Equation (2.119) yields at the diagonal points $(\sigma_1 = \sigma_2 = \sigma)$ of the 2nd-order binary kernel:

$$g_2^b(\sigma,\sigma) = C_2 A^2 E[y_2(t)] = 0 \quad (2.124)$$

since the 2nd-order output residual has zero mean by construction. The same zero value is obtained for all diagonal points of higher order binary kernels with even parity (i.e., when an even number of arguments are the same) because $x^{2n}(t) = A^{2n}$ and the high-order cross-correlations reduce to lower order ones that are orthogonal to the output residual.

By the same token, the estimated values of binary kernels at diagonal points with odd parity are zero, because all the odd-order cross-correlations reduce to a first-order one that is zero for all residuals of order higher than first. For instance, the 3rd-order binary kernel estimate at the diagonal of parity two (i.e., $\sigma_1 = \sigma'$, $\sigma_2 = \sigma_3 = \sigma$) is:

$$g_3^b(\sigma', \sigma, \sigma) = C_3 A^2 E\left[y_3(t)\, x(t - \sigma')\right] = 0 \tag{2.125}$$

since the 3rd-order output residual is orthogonal to $x(t - \sigma')$. The estimates at the diagonal points of parity three $(\sigma_1 = \sigma_2 = \sigma_3)$ also yield zero result, because the 3rd-order cross-correlation reduces to the 1st-order one indicated by Equation (2.125).

It can be stated, in general, that the binary CSRS input "folds" all diagonal kernel estimates of dimension two or above (obtained via cross-correlation) into lower order projections and yields a zero estimated value (inability to estimate these diagonal kernel values) because of the orthogonality between the output residual and the reduced order of the instrumental homogeneous functional (i.e., the number of cross-correlated input product terms is smaller than the order of the output residual).

The same can be stated for a ternary CSRS input with regard to the diagonal of the 3rd-order ternary kernel estimate or all diagonal values of dimension three and above in higher order ternary kernels.

In general, a CSRS input with L amplitude levels cannot yield kernel estimates (via cross-correlation) at diagonal points of dimension L or above. The contributions of these diagonal points to the system output are folded into lower order functional terms and the resulting cross-correlation estimates of the CSRS kernels for these diagonal points are zero. An analytical example is given below that elucidates these points.

Since kernel estimation via cross-correlation has been limited in practice to second order (and rarely extending to third order), the practical implications of the aforementioned fact of "diagonal estimability" attain importance only for binary (and occasionally ternary) quasi-white inputs. The binary test inputs have been rather popular among investigators, in either the CSRS form or in the pseudorandom m-sequence form (discussed below). Therefore, the issue of "diagonal estimability" attains practical importance, since many investigators have been perplexed in the past by the so-called "diagonal

problem" when binary test inputs are used to estimate 2nd-order kernels. This discussion elucidates the problem and provides useful guidance in actual applications.

An illustrative example from a real physiological system (the fly photoreceptor) is given in Figure 2.13, where the 2nd-order kernel estimates obtained via cross-correlation using binary and ternary CSRS test inputs are shown. The differences along the diagonal points are evident and point to the potential pitfalls of the common (but questionable) practice of interpolating the diagonal values of the binary kernel estimate to overcome the "diagonal problem". It is evident from Figure 2.13 that diagonal interpolation of the binary kernel estimate would not reproduce the correct kernel values (represented by the ternary estimate) especially along two segments of the diagonal (the part between the two positive humps and for latencies shorter than 8 msec). There may be certain cases where interpolation may yield reasonable approximations along the diagonal (depending on the true kernel morphology) but this cannot be asserted in general. However, even in those cases where interpolation in justifiable, adjustments must be made to the lower order kernel estimates of the same parity (e.g., the zero-order kernel estimate when the second-order kernel is interpolated along the diagonal) in order to balance properly the contributions of these kernel values to the system output.

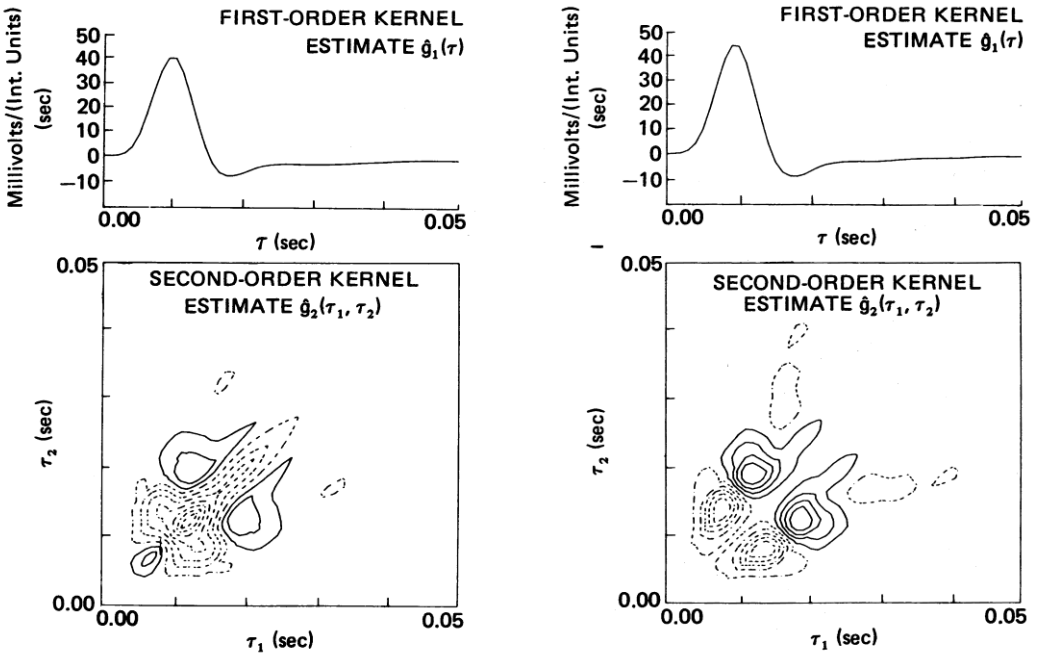


Figure 2.13
The first and second order CSRS kernel estimates of the fly photoreceptor obtained with the use of binary (right) and ternary (left) test inputs. The differences along the diagonal of the second-order kernel are evident [Marmarelis & McCann, 1977].

An Analytical Example

As an analytical example of the relation between CSRS and Volterra kernels, consider a third-order Volterra system (i.e., a system for which $k_n(\tau_1, \tau_2, ..., \tau_n) = 0$ for $n > 3$). If $k_0 = 0$, the system response to a CSRS $x(t)$ is:

$$y(t) = \int_0^\infty k_1(\tau_1)\, x(t-\tau_1)\, d\tau_1 + \int_0^\infty \int k_2(\tau_1, \tau_2)\, x(t-\tau_1)\, x(t-\tau_2)\, d\tau_1 d\tau_2$$

$$+ \int_0^\infty \int \int k_3(\tau_1, \tau_2, \tau_3)\, x(t-\tau_1)\, x(t-\tau_2)\, x(t-\tau_3)\, d\tau_1 d\tau_2 d\tau_3 \qquad (2.126)$$

If we consider a CSRS input with more than three amplitude levels (to avoid the aforementioned problem of "diagonal estimability"), then the CSRS kernels of the system (for the non-diagonal points) are:

$$g_0 = (M_2 \Delta t) \int_0^\infty k_2(\tau_1, \tau_1)\, d\tau_1 \qquad (2.127)$$

$$g_1(\sigma_1) = k_1(\sigma_1) + 3(M_2 \Delta t) \int_0^\infty k_3(\sigma_1, \tau_1, \tau_1) d\tau_1 + \left[\left(\frac{M_4}{M_2} - 3M_2 \right) \Delta t^2 \right] k_3(\sigma_1, \sigma_1, \sigma_1) \qquad (2.128)$$

$$g_2(\sigma_1, \sigma_2) = k_2(\sigma_1, \sigma_2) \qquad (2.129)$$

$$g_3(\sigma_1, \sigma_2, \sigma_3) = k_3(\sigma_1, \sigma_2, \sigma_3) \qquad (2.130)$$

The second-order and third-order CSRS kernels are identical to the Volterra kernels because of the absence of nonlinearities of order higher than third and the multi-level structure of the CSRS input (more than three levels). The zeroth-order CSRS kernel depends upon the second-order Volterra kernel, and the first-order CSRS kernel depends upon the first-order and third-order Volterra kernels in a manner similar to the dependence on the Wiener kernels as Δt tends to zero.

If a binary CSRS input is used, then the diagonal values of the estimated 2nd-order and 3rd-order binary kernels are zero. The "nulling" of the diagonal values of g_2^b and g_3^b simplifies the form of the CSRS functionals given by Equations (2.115) – (2.118), because only the leading term of each functional remains non-zero for a binary CSRS input:

$$G_2^b\left[g_2^b; x(t'), t' \le t \right] = \int_0^\infty \int g_2^b(\tau_1, \tau_2)\, x(t-\tau_1)\, x(t-\tau_2)\, d\tau_1 d\tau_2 \qquad (2.131)$$

$$G_3^b\left[g_3^b; x(t'), t' \le t \right] = \int_0^\infty \int \int g_3^b(\tau_1, \tau_2, \tau_3)\, x(t-\tau_1)\, x(t-\tau_2)\, x(t-\tau_3)\, d\tau_1 d\tau_2 d\tau_3 \qquad (2.132)$$

where the superscript "b" denotes "binary" functionals or kernels. This simplification applies to all higher order functionals that may be present in a system. This modified form of the CSRS functional series for binary inputs will be revisited in the study of neuronal systems with spike-train inputs (see Chapter 8).

If a ternary CSRS input is used, then only the main diagonal values of g_3 become zero and only the 3rd-order CSRS functional is simplified by elimination of its last term (for this example of a 3rd-order system). For a high-order system, the kernel values for all diagonal points of parity three or above attain zero values, with concomitant simplifications of the respective ternary functionals. Since most applications have been limited to 2nd-order models thus far, the ternary CSRS inputs appear to be a better choice than their more popular binary counterparts in terms of avoiding the "diagonal problem" in the 2nd-order kernel estimate.

It is instructive to explore also the form of the Wiener kernels of this 3rd-order system as an illustrative example. The second-order and third-order Wiener kernels are identical to their Volterra (and non-diagonal CSRS) counterparts for this system. The zeroth-order and first-order Wiener kernels are given by:

$$h_0 = P\int_0^\infty k_2(\tau_1,\tau_1)\,d\tau_1 \tag{2.133}$$

$$h_1(\sigma_1) = k_1(\sigma_1) + 3P\int_0^\infty k_3(\sigma_1,\tau_1,\tau_1)\,d\tau_1 \tag{2.134}$$

where P is the power level of the GWN input. In other words, the Wiener kernels can be viewed as a special case of CSRS kernels, when $M_4 = 3M_2^2$ (i.e., when the amplitude distribution is Gaussian), or when Δt is very small (i.e., when the third term of g_1 in Equation (2.128) becomes negligible relative to the second term).

Comparison of Model Prediction Errors

Of importance in system modeling is the accuracy (in the mean-square error sense) of the model-predicted response to a given stimulus. This issue was discussed previously with regard to Wiener models and is examined here with regard to the predictive ability of a CSRS model relative to a Volterra model of the same order. To simplify the mathematical expressions (without loss of conceptual

generality), we consider the zeroth-order model of the system in the previous analytical example. The mean-square error (MSE) of the zeroth-order Volterra model prediction is:

$$Q_v = E\left[y^2(t)\right] \tag{2.135}$$

because we considered $k_0 = 0$, while the MSE of the zeroth-order CSRS model prediction is:

$$\begin{aligned} Q_c &= E\left[\left[y(t) - g_0\right]^2\right] \\ &= E\left[y^2(t)\right] + g_0^2 - 2g_0 E\left[y(t)\right] \end{aligned} \tag{2.136}$$

Therefore, the improvement in accuracy of the zeroth-order model prediction, using a CSRS model instead of a Volterra model for an arbitrary input, is:

$$\begin{aligned} i_0 &= Q_v - Q_c \\ &= 2g_0 E\left[y(t)\right] - g_0^2 \end{aligned} \tag{2.137}$$

If the input signal is the CSRS with which the model was estimated, then:

$$\begin{aligned} i_0 &= g_0^2 \\ &= \left[P_1 \int_0^\infty k_2(\tau_1, \tau_1)\, d\tau_1\right]^2 \geq 0 \end{aligned} \tag{2.138}$$

As expected, we always have improvement in predicting the system output for a CSRS input of the same power level with which kernel estimation was performed.

If other CSRS inputs of different power level P_1^* are used to evaluate the zeroth-order model, then:

$$i_0 = P_1\left(2P_1^* - P_1\right)\left[\int_0^\infty k_2(\tau_1, \tau_1)\, d\tau_1\right]^2 \tag{2.139}$$

which is similar to the result for the Wiener model (see Equation (2.82)). Thus, we can have improvement or deterioration in the accuracy of the zeroth-order model prediction, depending on the relative size of the power levels.

If the input $x(t)$ is an arbitrary signal, then:

$$i_0 = P_1 \int_0^\infty k_2(\tau_1, \tau_1)\, d\tau_1 \left\{ 2\mu \int_0^\infty k_1(\tau_1)\, d\tau_1 + 2\int_0^\infty\int\int k_3(\tau_1, \tau_2, \tau_3)\phi_3(\tau_1, \tau_2, \tau_3)\, d\tau_1 d\tau_2 d\tau_3 + \int_0^\infty\int k_2(\tau_1, \tau_2)\left[2\phi_2(\tau_1, \tau_2) - P_1\delta(\tau_1 - \tau_2)\right] d\tau_1 d\tau_2 \right\} \tag{2.140}$$

where ϕ_2 and ϕ_3 are the second and third order autocorrelation functions of the input signal, and μ is the input mean. Equation (2.140) clearly demonstrates the fact that the improvement (or deterioration)

in the case of an arbitrary input signal depends upon the autocorrelation functions of this signal. This establishes the important fact that the performance of models obtained with quasi-white test inputs (including band-limited GWN) depends crucially on the relation of the autocorrelation functions of the specific input with the ones of the quasi-white test signal used to obtain the model.

Discrete-Time Representation of the CSRS Functional Series

Since sampled data are used in practice to perform the modeling task, it is useful to examine the form of the CSRS functional series in discrete time. This form is simplified when the sampling interval T is equal to the CSRS step size Δt. Because of aliasing considerations, T cannot be greater than Δt and, if $T < \Delta t$, the integrals of the continuous-time representation of the CSRS functionals attain complicated discretized forms. Therefore, we adopt the convention that $T = \Delta t$ for actual applications, which allows the conversion of the integrals of the CSRS functionals into summations to yield the discrete-time representation (for $\Delta t = T$):

$$G_1^*(n) = T\sum_m g_1(m)\,x(n-m) \tag{2.141}$$

$$G_2^*(n) = T^2\sum_{m_1}\sum_{m_2} g_2(m_1,m_2)\,x(n-m_1)\,x(n-m_2) - M_2T^2\sum_m g_2(m,m) \tag{2.142}$$

$$G_3^*(n) = T^3\sum_{m_1}\sum_{m_2}\sum_{m_3} g_3(m_1,m_2,m_3)\,x(n-m_1)\,x(n-m_2)\,x(n-m_3)$$

$$-3M_2T^3\sum_m\sum_{m'} g_3(m,m',m')\,x(n-m) - \left(\frac{M_4}{M_2}-3M_2\right)T^3\sum_m g_3(m,m,m)\,x(n-m\ ) \tag{2.143}$$

where the tilde denotes the discrete-time (sampled) counterparts of the continuous-time variables. Aside of possible scaling factors involving T, the discretized values of the kernels and input variables are simply the corresponding sampled values. These discretized forms should be used in practice to interpret the CSRS kernels and functionals, when the cross-correlation technique is used for CSRS kernel estimation.

Pseudorandom Signals Based on m-Sequences

In order to reduce the natural redundancy of the random quasi-white signals, while still preserving quasi-white autocorrelation properties, one may employ specially crafted pseudorandom signals(PRS) based on *m*-sequences, which are deterministic periodic signals with quasi-white autocorrelation properties within a period of the PRS. These PRS signals are generated with linear auto-recursive

relations designed to yield sequences with maximum period (see below) [Zierler, 1959; Gyftopoulos & Hooper, 1964; Barker, 1967; Davies, 1970; Moller, 1973; Sutter, 1975].

An important advantage of the PRS is the fact that their second-order autocorrelation function is zero outside the neighborhood of the origin (zero lag) and within, of course, the period of the signal (since the autocorrelation functions are also periodic). This is an advantage over random quasi-white signals (such as the CSRS), which exhibit small nonzero values in this region of their second-order autocorrelation function. The latter cause some statistical error in CSRS kernel estimation (see Section 2.4.2). However, the PRS exhibit significant imperfections in their higher even-order autocorrelation functions, which offset their superiority in the second-order autocorrelation properties and may cause significant errors in the estimation of high-order kernels. For this reason, the PRS are most advantageous in identification of linear systems, while the presence of nonlinearities in the system makes the choice between random and pseudorandom quasi-white test signals dependent upon the specific characteristics of the system at hand.

The PRS exhibit the same stair-like form as the CSRS, i.e., they remain constant within small time intervals defined by the step size Δt and switch abruptly at multiples of Δt. Their values at each step are determined by a linear recurrence formula of the form

$$x_i = a_1 \otimes x_{i-1} \oplus a_2 \otimes x_{i-2} \oplus \ldots \oplus a_m \otimes x_{i-m} \tag{2.144}$$

where the coefficients a_j and the signal values x_i correspond to the elements of a finite Galois field (i.e., a finite set of integer values equal to a power of a prime number). The operations $\{\otimes \oplus\}$ are defined to be internal operations of multiplication and addition for the specified Galois field. For example, in the case of a binary pseudorandom signal, the Galois field has two elements (0 and 1) and the operations $\oplus$ and $\otimes$ are defined to be modulo 2 (i.e., corresponding to "AND" and "OR" Boolean logic operations), so that the outcome of the recurrence formula (2.144) is also an element of same Galois field (i.e., binary).

It is evident that a sequence $\{x_i\}$ constructed on the basis of the linear recurrence formula (2.144) is periodic, because the root string of m consecutive values of x_i will repeat after a finite number of steps. The length of this period depends on the specific values of the coefficients a_j and the order m of the recurrence formula (for a given Galois field). Among all the sequences $\{x_i\}$ constructed from the L members of a certain Galois field (a prime number) and with linear recurrence formulae of order *m*, there

are some that have the maximum period. Since L^m is the number of all possible distinct arrangements with repetitions of L elements in strings of *m*, this maximum period is $\left(L^m - 1\right)$, where L is a prime number and the null string is excluded.

These maximum period sequences are called "maximum-length" or "*m*-sequences," and they correspond to a special choice of the coefficients $\{a_1, ..., a_m\}$ that coincide with the coefficients of a primitive (or irreducible) polynomial of degree $(m-1)$ in the respective Galois field [cf. Zierler, 1959]. Thus, we can always select the number of elements L and the order of the recurrence formula m in such a way that we get an m-sequence with a desirable period (within the limitations imposed by the integer nature of m and prime L).

The generation of PRS in the laboratory is a relatively simple task. Suppose we have decided upon the prime number L of amplitude values that the signal will attain and the required maximum period $\left(L^m - 1\right)$ (i.e., the order m of the linear recurrence formula (2.144)). Now, we only need the coefficients of a primitive polynomial of degree $(m-1)$ in the respective L-element Galois field. If such a primitive polynomial is available (such polynomials have been tabulated, cf. Church, 1935), then we choose an initial string of values and construct the corresponding *m*-sequence. Any initial string (except the null one) will give the same *m*-sequence (for a given set of coefficients a_j), merely shifted.

Specialized pieces of hardware can be used for real-time generation. For example, a binary *m*-sequence can be generated through a digital shift-register, composed of a cascade of flip-flops (0 or 1) and an "exclusive OR" feedback connection, as shown in Figure 2.14. Upon receipt of a shift (or clock) pulse, the content of each flip-flop is transferred to its neighbor and the input to the first stage is being received from the output of the "exclusive OR" gate (that generates a 0 when the two inputs are same, and 1 otherwise).

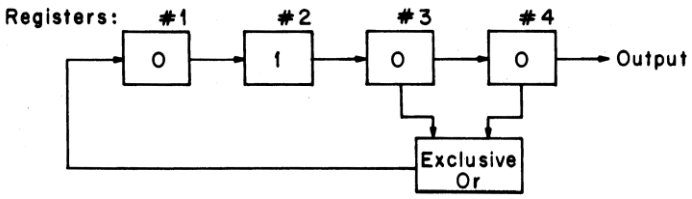


Figure 2.14
Shift-register with "exclusive OR" feedback for the generation of pseudorandom sequences [Marmarelis & Marmarelis, 1978].

We note that a 15-bit m-sequence can be generated by this four-stage shift-register, corresponding to the maximum number $\left(2^4-1\right)$ of possible four-bit binary numbers, except for the null one (since if 0000 ever occurred, the output thereafter would be zero).

Such pseudorandom sequences, produced by shift-registers with feedback, have been studied extensively cf. Davies, 1970; Golomb, 1976; Gold, 1984 in connection with many engineering applications, especially in communication systems (spread-spectrum communications and CDMA protocols of wireless telephony). Table 2.1 gives the possible stage numbers in the shift-register from which the output, along with the output from the last stage, could be fed into the "exclusive-OR" gate and fed back into the first stage, in order to obtain a maximum-length binary sequence [Davies, 1970].

The quasi-whiteness of a pseudorandom signal (and consequently its use for Wiener/CSRS kernel estimation in connection with the cross-correlation technique) is due to the shift-and-add property of the *m*-sequences (cf. Ream, 1970). According to this property, the product (of the proper modulo) of any number of sequence elements is another sequence element:

$$x_{k-j_1} \otimes x_{k-j_2} \otimes ... \otimes x_{k-j_m} = x_{k-j_0} \tag{2.145}$$

where j_0 depends on $j_1, j_2, ..., j_m$ but not on *k*. A slight modification must be made in the *m*-sequences with even numbers of levels in order to possess the anti-symmetric property, which entails the inversion of every other bit of the *m*-sequence (doubling the period of the sequence) and makes the odd-order autocorrelation functions uniformly zero. As a result of the shift-and-add property and the basic structural characteristics of the *m*-sequences (i.e., maximum period and anti-symmetry), the odd-order autocorrelation functions are uniformly zero everywhere (within a period of the signal) and the even-order ones approximate quasi-whiteness.

Note that the even-order autocorrelation functions of order higher than second exhibit some serious imperfections (termed "anomalies" in the literature), which constitute a serious impediment in the use of PRS test inputs for nonlinear system identification using the cross-correlation technique [Barker *et al*. 1972]. These anomalies, first observed by Gyftopoulos and Hooper (1964, 1967), have been studied extensively by Barker and Pradisthayon (1970). Barker *et al*. (1972) studied several PRS (binary, ternary, and quinary) and compared their relative performance, showing that these anomalies are due to existing linear relationships among the elements of the m-sequence and that their exact position and magnitude can be determined from the generating recurrence equation through a laborious algorithm related to polynomial division. Since these anomalies are proven to be inherent characteristics of the *m*-

sequences related to their mathematical structure, their effect can be anticipated and potentially mitigated through an elaborate iterative scheme.

The kernels estimated by the use of a PRS and the corresponding functional series are akin to the CSRS kernels of the associated multi-level amplitude distribution, corresponding to of the specific PRS.

Comparative Use of GWN, PRS, and CSRS

The quasi-white test input signals that have been used so far to estimate the Wiener/CSRS kernels of nonlinear systems through cross-correlation are: band-limited GWN, PRS, and CSRS. Each one of these classes of quasi-white input signals exhibits its own characteristic set of advantages and disadvantages, summarized below.

(I) For the GWN. The main advantage of band-limited GWN derives from its Gaussian nature that secures the simplest expressions for the orthogonal functional series (the Wiener series) because of the decomposition property of Gaussian random variables (which allows all the even-order autocorrelation functions to be expressed in terms of the second-order one). Additionally, the use of a GWN test input avoids the estimation problems at diagonal kernel points, associated with the use of binary or ternary PRS/CSRS.

The main disadvantages of GWN are the actual generation and application in the laboratory (including the unavoidable truncation of the Gaussian amplitude distribution), as well as the imperfections in the autocorrelation functions due to its stochastic nature and the finite data-records.

(II) For the PRS. The main advantages of binary or ternary PRS are two:

(1) easy generation and application;

(2) short records required to form the desirable autocorrelation functions.

The main disadvantages of PRS are two:

(1) the anomalies in their higher (>2) even-order autocorrelation functions, which may induce considerable kernel estimation errors if the system contains significant nonlinearities;

(2) the inability to estimate the diagonal kernel values using binary PRS.

(III) For the CSRS. The main advantages of the CSRS are four:

(1) easy generation and application;

(2) their autocorrelation functions do not exhibit any “anomalies” as in the case of PRS;

(3) error analysis is facilitated by the simple structure of CSRS, allowing the design of an optimum test input;

(4) the user is given flexibility in choosing the signal with the number of levels and amplitude PDF that fits best the specific case at hand.

The main disadvantages of the CSRS are three:

(1) they require fairly long records in order to reduce the statistical error in the kernel estimates (as in the case of GWN);

(2) the analytical expressions concerning the corresponding functional series and kernels are fairly complicated (e.g., relation of CSRS kernels with Volterra kernels, normalizing factors of the cross-correlation estimates, etc.);

(3) the inability to estimate the diagonal kernel values using binary CSRS. Note that for a ternary test input (CSRS or PRS), this inability concerns the estimation of the 3^{rd}-order kernel main diagonal (or higher order diagonals).

Besides these basic advantages and disadvantages of GWN, PRS, and CSRS, there may be other factors that become important in a specific application because of particular experimental or computational considerations.

2.2.5. Apparent Transfer Function and Coherence Measurements

One of the questionable habits forced upon investigators by the lack of effective methodologies for the study of nonlinear systems is the tendency to "linearize" physiological systems with intrinsic nonlinearities by applying uncritically linear modeling methods in the frequency domain. An "apparent transfer function" measurement is typically sought in those cases, often accompanied by "coherence function" measurements in order to test the validity of the linear assumption or establish the extent of the "linearized" approximation. Specifically, the "coherence function" is computed over the entire frequency range of interest using Equation (2.149), and the proximity of its values to unity is examined (coherence values are by definition between 0 and 1). If the coherence values are found to be close to unity over the frequency range of interest, then the inference is made that the linear time-invariant assumption is valid and the noise content of the experimental data is low. In the opposite case, the reduced coherence is thought to indicate the presence of either system nonlinearities (and/or nonstationarities) and/or high noise content in the data. Distinguishing among those possible culprits for the reduction in coherence values requires specialized testing and analysis (e.g., repetition of identical

experiments and averaging to reduce possible noise effects, or application of nonlinear/nonstationary analysis to assess the respective effects). As a rule of thumb, coherence values above 0.8 are thought to validate the linearized approximation.

In this section, we examine the apparent transfer function and coherence function measurements in the general framework of nonlinear systems with GWN inputs following the Wiener approach [Marmarelis, 1988], thus offering a rigorous guide for the proper interpretation of these two popular experimental measurements.

In the Wiener series representation of nonlinear systems/models, the Wiener functionals are constructed to be *mutually orthogonal* (or have zero covariance) for a GWN input with power level P. Thus, using this statistical orthogonality (zero covariance) and the basic properties of high-order autocorrelation functions of GWN summarized in Appendix II, we can find the output spectrum to be:

$$S_y(f) = h_0^2 \delta(f) + P|H_1(f)|^2 + \sum_{r=2}^{\infty} r! P^r \int_{-\infty}^{\infty} \ldots \int |H_r(u_1, \ldots, u_{r-1}, f - u_1 - \ldots - u_{r-1})|^2 du_1 \ldots du_{r-1} \tag{2.146}$$

where H_r is the r-dimensional Fourier transform of the r^{th}-order Wiener kernel of the system h_r, and f denotes frequency in Hz. Since the input-output cross-correlation is the 1st-order Wiener kernel scaled by P (see Equation (2.98)), its Fourier transform yields the cross-spectrum when the input is GWN:

$$S_{yw}(f) = PH_1(f) \tag{2.147}$$

Therefore, the commonly evaluated "apparent transfer function" (ATF) is the 1st-order Wiener kernel in the frequency domain:

$$H_{app}(f) \triangleq \frac{S_{yw}(f)}{S_w(f)} = H_1(f) \tag{2.148}$$

since $S_w(f) = P$ for a GWN input. The coherence function becomes:

$$\gamma^2(f) \triangleq \frac{|S_{yw}(f)|^2}{S_w(f) S_y(f)} = \frac{|H_1(f)|^2}{|H_1(f)|^2 + \sum_{r=2}^{\infty} r! P^{r-1} \int \ldots \int |H_r(u_1, \ldots, u_{r-1}, f - u_1 - \ldots - u_{r-1})|^2 du_1 \ldots du_{r-1}} \tag{2.149}$$

for all frequencies other than $f = 0$. Since the summation in the denominator of Equation (2.149) includes only nonnegative terms, it is clear that the coherence function will be less than unity to the extent determined by the GWN input power level and the indicated integrals of the high-order kernels of the system. Equation (2.149) shows that for a linear system or very small P, the coherence function is close to unity.

Note that the coherence function is further reduced in the presence of noise. For instance, in the presence of output-additive noise $n(t)$ the output signal is:

$$y(t) = y(t) + n(t) \tag{2.150}$$

which leaves the input-output cross-spectrum unchanged ($S_{yw} \equiv S_{yw}$) if the noise has zero mean and is statistically independent from the output signal, but the output spectrum becomes:

$$S_y(f) = S_y(f) + S_n(f) \tag{2.151}$$

where $S_n(f)$ is the noise spectrum. Thus, the coherence function for the noise-contaminated output data is reduced according to the relation:

$$\gamma^2(f) = \gamma^2(f)\frac{S_y(f)}{S_y(f) + S_n(f)} \tag{2.152}$$

Since the ATF equals the Fourier transform of the 1st-order Wiener kernel, we can express it in terms of the Volterra kernels of the system:

$$H_{app}(f) = \sum_{m=0}^{\infty} \frac{(2m+1)!P^m}{m!2^m} \int_{-\infty}^{\infty} \ldots \int K_{2m+1}(f, u_1, -u_1, \ldots, u_m, -u_m) du_1 \ldots du_m \tag{2.153}$$

which indicates that the ATF of a nonlinear system depends on all odd-order Volterra kernels of the system and the GWN input power level. Therefore, measurements of $H_{app}(f)$ with GWN inputs *differ from the linear transfer function* of the system (represented by the 1st-order Volterra kernel $K_1(f)$) and vary with the power level of the GWN input as a power series, according to Equation (2.153).

In many studies of physiological systems, experimental considerations dictate that GWN test inputs of various nonzero mean levels be used (e.g., in the visual system the input light intensity can assume, physically, only positive values). In those cases, the computation of the Wiener kernels requires de-meaning of the input and the resulting kernel estimates depend on the input mean level.

Thus, the ATF in this case becomes:

$$H_{app}^{\mu}(f) = \sum_{m=0}^{\infty} \sum_{l=0}^{\infty} \frac{(2m+\ell+1)!}{m!\ell!} \left(\frac{P}{2}\right)^m (\mu - \mu_0)^{\ell} \int_{-\infty}^{\infty} \ldots \int K_{2m+\ell+1}^{0}(f, u_1, -u_1, \ldots, u_m, -u_m, 0, \ldots, 0) du_1 \ldots du_m \tag{2.154}$$

where μ is the nonzero mean of the GWN test input and μ_0 is the reference mean level (possibly but not necessarily, zero) for the nominal Volterra kernels $\{K_i^0\}$. The coherence function measurement is also affected, because the Wiener kernel estimates depend on the GWN input μ used in the experiment.

Example 2.5

L-N cascade system

As an illustrative example, we consider the case of a simple $L-N$ cascade of a linear filter, with impulse response function $g(\tau)$, followed by a static nonlinearity that is represented by a Taylor series expansion with coefficients $\{a_r\}$. The system output is:

$$y(t)=\sum_{r=0}^{\infty}a_r v^r(t)=\sum_{r=0}^{\infty}a_r\left[\int_0^{\infty}g(\tau)x(t-\tau)d\tau\right]^r \tag{2.155}$$

where $v(t)$ is the output of the linear filter.

The Volterra kernels of this system in the frequency domain are (see Section 4.1):

$$K_r(f_1,...,f_r)=a_r G(f_1)...G(f_r) \tag{2.156}$$

and the ATF can be found using Equation (2.53) to be:

$$H_{app}(f)=\left\{\sum_{m=0}^{\infty}\frac{(2m+1)!P^m}{m!2^m}a_{2m+1}\left[\int_{-\infty}^{\infty}|G(u)|^2\,du\right]^m\right\}G(f)$$

$$=c\cdot G(f) \tag{2.157}$$

i.e., it is a scaled version of $G(f)$ which is the transfer function of the linear component of the cascade. The scaling factor c depends on the static nonlinearity, the input power level and the Euclidean norm of $G(f)$.

The coherence function in this example is found from Equations (2.57) and (2.149) to be (for $f\neq 0$):

$$\gamma^2(f)=\frac{c^2|G(f)|^2}{c^2|G(f)|^2+\sum_{r=2}^{\infty}r!P^{r-1}A_r^2\int...\int|G(u_1)...G(u_{r-1})G(f-u_1-...-u_{r-1})|^2du_1...du_{r-1}} \tag{2.158}$$

where:

$$A_r=\sum_{m=0}^{\infty}\frac{(r+2m)!P^m}{r!m!2^m}a_{r+2m}\left[\int|G(u)|^2\,du\right]^m \tag{2.159}$$

This indicates that, even in this relatively simple case, the coherence function is a rather complicated expression, dependent on the system nonlinearities and the input power level in the manner described by the denominator of Equation (2.158). To reduce the complexity, let us consider a quadratic nonlinearity (i.e., $a_r=0$ for $r>2$). Then,

$$H_{app}(f)=a_1G(f) \tag{2.160}$$

and:

$$\gamma^2(f)=\left[\frac{|G(f)|^2}{|G(f)|^2+2P\left(\frac{a_2}{a_1}\right)^2\int|G(u)G(f-u)|^2\,du}\right] \tag{2.161}$$

for $f \neq 0$. Equation (2.161) indicates clearly that the quadratic nonlinearity reduces the coherence values more as P and/or $(a_2/a_1)^2$ increase.

An illustration of this is given in Figure 2.15 where part (a) shows the first and second order kernels of a quadratic cascade system in the frequency domain (gain functions only) and part (b) shows the ATF measurements for three different values of GWN input power level ($P = 0.25$, 1, and 4) obtained from input-output records of 4,096 data points each. We observe slight variations in the ATF measurements due to the finite data-record which leads to imperfectly forming averages. Part (c) of Figure 2.15 shows the coherence function measurements obtained from the same input-output data records and values of P. The observed effect of P on the coherence function measurement is in agreement with Equation (2.161).

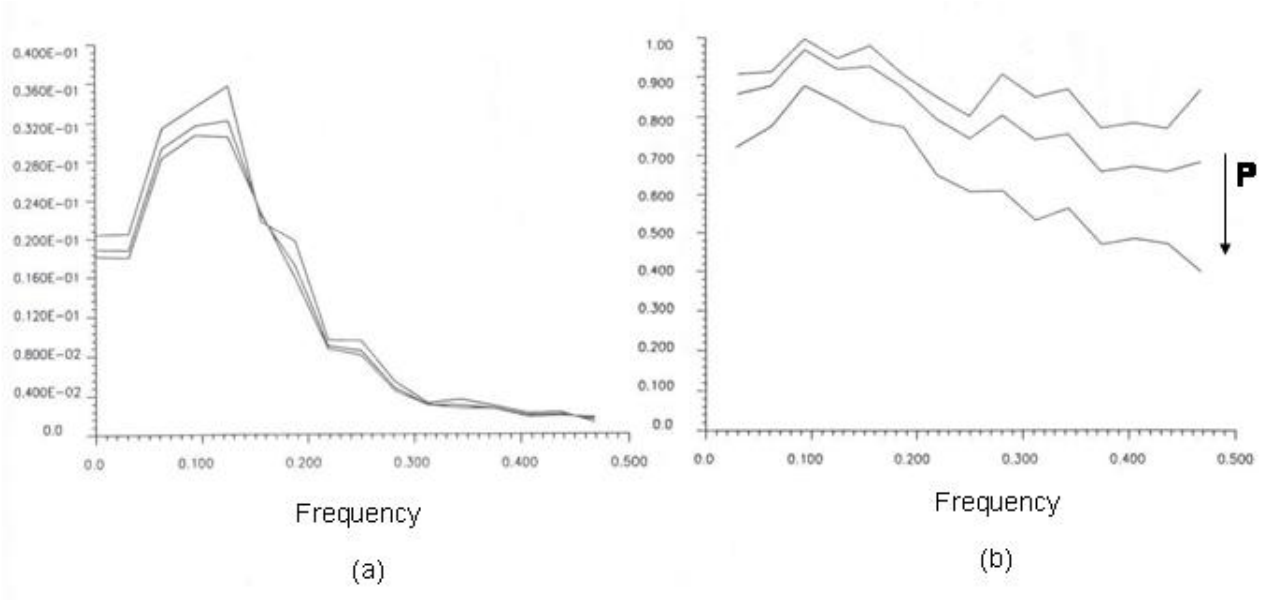


Figure 2.15
(a) The first and second order kernels of the quadratic cascade system in the frequency domain (magnitude only); (b) the gain of the apparent transfer function for three different power levels of GWN input ($P = 0.25, 1$ and 4); (c) the coherence function measurements for the three GWN power input levels [Marmarelis, 1988].

Example 2.6

Quadratic Volterra system

Since quadratic nonlinear systems are quite common in physiology (e.g., in the visual system), we use them in a second example which is more general than the first in that the system is described by two general kernels (K_1, K_2) and is not limited to a cascade arrangement. Note that these kernels are the same for the Volterra and the Wiener expansions--a fact that holds for all 2nd-order systems ($K_r \equiv 0,$ for $r > 2$). For this general 2nd-order system, we have (for $f \neq 0$):

$$H_{app}(f) = K_1(f) \tag{2.162}$$

$$\gamma^2(f) = \frac{|K_1(f)|^2}{|K_1(f)|^2 + 2P\int |K_2(u, f-u)|^2 \, du} \tag{2.163}$$

Equation (2.162) shows that the ATF in this case is identical to the linear portion (1st-order Volterra kernel) of the system. However, if a nonzero mean GWN test input is used, then we can find from Equation (2.154) that:

$$H^{\mu}_{app}(f) = K_1^0(f) + 2(\mu - \mu_0)K_2^0(f, 0) \tag{2.164}$$

i.e., the ATF measurement is affected by the 2nd-order kernel and the non-zero mean of the GWN test input signal.

Equation (2.163) shows that the coherence function values are reduced as the relative nonlinear contribution increases. The latter is different for different GWN input mean level μ, since the corresponding $K_1^{\mu}(f)$ or $H_1^{\mu}(f)$ will vary with μ as:

$$K_1^{\mu}(f) = K_1^0(f) + 2(\mu - \mu_0)K_2^0(f, 0) \tag{2.165}$$

while K_2 (or H_2) remains the same for all values of μ.

Example 2.7

Non-white Gaussian inputs

We can use the case of quadratic nonlinear systems to demonstrate also the effect of nonlinearities on coherence and apparent transfer function measurements when the input $x(t)$ is zero-mean Gaussian but **non-white**. In this case, the cross-spectrum is:

$$S_{yx}(f) = K_1(f) S_x(f) \tag{2.166}$$

and the output spectrum is (for $f \neq 0$):

$$S_y(f) = |K_1(f)|^2 S_x(f) + 2\int_{-\infty}^{\infty} |K_2(u, f-u)|^2 S_x(u) S_x(f-u)\, du \tag{2.167}$$

where S_x is the input spectrum.

Consequently, the ATF for quadratic systems is exactly the linear component of the system, $K_1(f)$, regardless of whiteness of the input. On the other hand, the coherence function is (for $f \neq 0$):

$$\gamma^2(f) = \frac{1}{1 + 2\int_{-\infty}^{\infty} \left| \frac{K_2(u, f-u)}{K_1(f)} \right|^2 \left[\frac{S_x(u)}{S_x(f)} \right] S_x(f-u)\, du} \tag{2.168}$$

which is clearly less than unity to the extent determined by the degree of nonlinearity and the input spectrum, as indicated in the second term of the denominator.

Example 2.8

Duffing system

As a final example, let us consider a system described by a relatively simple nonlinear differential equation:

$$Ly + \alpha y^3 = x \tag{2.169}$$

where L is a linear differential operator of qth-order with constant coefficients (i.e., $L(D) = c_q D^q + ... + c_1 D + c_0$, where D denotes the differential operator $\frac{d(\cdot)}{dt}$). When L is of second-order, Equation (2.169) is the Duffing equation, popular in nonlinear mechanics because it describes a mass-spring system with cubic elastic characteristics.

The Volterra kernels of this nonlinear differential system are [Marmarelis et al. 1979]:

$$K_1(f) = 1/L(f) \tag{2.170}$$

$$K_2(f_1, f_2) = 0 \tag{2.171}$$

$$K_3(f_1, f_2, f_3) = -\alpha K_1(f_1) K_1(f_2) K_1(f_3) K_1(f_1 + f_2 + f_3) \tag{2.172}$$

The general expression for the odd-order Volterra kernels of this system is:

$$K_{2r+1}(f_1,...,f_{2r+1}) = \frac{3(-\alpha)^r}{r(4r^2-1)} K_1(f_1)...K_1(f_{2r+1}) K_1(f_1+...+f_{2r+1}) \sum_{j_1,j_2,j_3} K_1(f_{j_1}+f_{j_2}+f_{j_3}) \qquad (2.173)$$

where the summation is taken over all combinations of (j_1, j_2, j_3) indices from the integers 1 through $(2r+1)$. All the even-order Volterra kernels of this system are zero. Since the $(2r+1)$th-order Volterra kernel is proportional to α^r, we can simplify the equivalent Volterra model when $\alpha \ll 1$ by neglecting the odd-order Volterra kernels of order higher than third. Then:

$$H_{app}(f) \cong K_1(f) - \lambda K_1^2(f) \qquad (2.174)$$

and:

$$\gamma^2(f) = \frac{\left|1-\lambda K_1(f)\right|^2}{\left|1-\lambda K_1(f)\right|^2 + 6\alpha^2 P^2 \iint \left|K_1(u_1) K_1(u_2) K_1(f-u_1-u_2)\right|^2 du_1 du_2} \qquad (2.175)$$

where:

$$\lambda = 3P\alpha \int \left|K_1(u)\right|^2 du \qquad (2.176)$$

Note that Equation (2.174) provides a practical tool for exploring a class of nonlinear feedback systems, as discussed in Section 4.1.5.

Concluding Remarks

In conclusion, it has been shown that for the case of GWN inputs:

1. The coherence function measurements reflect the presence of high-order (nonlinear) kernels and depend on the GWN input power level. Specifically, the coherence values are reduced from unity as the input power level P and/or the degree of nonlinearity increase.
2. The apparent transfer function is identical to the Fourier transform of the 1st-order Wiener kernel of the system, which is not, in general, the same as the linear component of the system (formally represented by the 1st-order Volterra kernel). The apparent transfer function depends on the system odd-order nonlinearities and on the power level of the GWN input.
3. In those physiological studies where the physical test input is a GWN perturbation around a nonzero mean μ (e.g., in vision), the apparent transfer function and the coherence function measurements depend on μ. Therefore, they will vary as μ may vary from experiment to experiment.

4. The same observations hold true for non-white broadband inputs, where additionally the specific form of the input spectrum affects the apparent transfer function (unless it is a second-order system) and the coherence measurements.

It is hoped that the presented analysis will assist investigators of physiological systems in interpreting their experimental results, obtained with these traditional frequency-domain measurements of the apparent transfer function and the coherence function, in the presence of intrinsic system nonlinearities.

2.3. EFFICIENT VOLTERRA KERNEL ESTIMATION

In Section 2.2, we described methods for the estimation of the Wiener kernels of a nonlinear system that receives a GWN (or quasi-white) input. For such inputs, the most widely used method to date has been the cross-correlation technique in spite of its numerous limitations and inefficiencies. Part of the reason is that, although the Wiener functional terms are decoupled (i.e., they are orthogonal for GWN inputs), their orthogonality (zero covariance) is approximate in practice, since their covariance is not exactly zero for finite data-records and causes estimation errors in the Wiener kernel estimates. Additional estimation errors associated with the cross-correlation technique are caused by the finite bandwidth of the input and the presence of noise/interference. These errors depend on the system characteristics and decrease with increasing data-record length, as discussed in Section 2.4.2.

The most important limitations of the cross-correlation technique are: (a) the stringent requirement of a band-limited white-noise input; (b) the input-dependence of the estimated Wiener (instead of Volterra) kernels; (c) the experimental and computational burden of long data-records; (d) the considerable estimation variance of the obtained kernel estimates (especially of order higher than first). The latter two limitations are an inevitable consequence of the stochastic nature of the employed GWN (or quasi-white) input and the fact that the cross-correlation estimates are computed from input-output data-records with finite length. These estimates converge to the true values at a rate proportional to the square-root of the record length. Note that this limitation is also incumbent on the initially proposed Wiener implementation using time-averaging computation of the covariance between any two Wiener functionals over finite data-records [Marmarelis & Marmarelis, 1978]. Thus, long data-records are required to obtain cross-correlation or covariance estimates of satisfactory accuracy, resulting in heavy experimental and computational burden.

In addition to the various estimation errors, the Wiener kernels are not input-independent and, therefore, the Volterra kernels are deemed more desirable in actual applications from the viewpoint of physiological interpretation. Methods for Volterra kernel estimation were discussed in Section 2.1.5, but they were also shown to exhibit practical limitations and inefficiencies. More efficient methods for Volterra kernel estimation are discussed in this section. These methods can be applied to systems with arbitrary (broadband) inputs typical of spontaneous/natural operation, thus removing a serious practical limitation of the Wiener approach (i.e., the requirement for white or quasi-white inputs) or the need for specialized inputs (e.g., impulses or sinusoids).

It is evident that, in practice, we can obtain unbiased estimates of the Volterra kernels only for systems of finite-order. These Volterra kernel estimates do not depend on input characteristics, when a complete (non-truncated) model is obtained for the system at hand. However, the Volterra kernel estimates for a truncated model generally have biases resulting from the correlated residuals which are dependent of the specific input used for kernel estimation. Thus, the attraction of obtaining directly the Volterra kernels of a system has been tempered by the fact that model truncation is necessitated in the practical modeling of high-order systems, which in turn introduces input-dependent estimation biases because the residuals are correlated and input-dependent since the functional terms of the Volterrra models are coupled.

Naturally, the biases and other estimation errors of the Volterra kernels depend on the specific characteristics of each application (e.g., system order, data record length, input characteristics etc.) and should be minimized in each particular case. To accomplish this error minimization, we must have a thorough understanding of the methodological issues regarding kernel estimation in a general, yet realistic, context.

The emphasis here will be placed on the most promising methods of Volterra kernel estimation that are applicable for nearly arbitrary inputs and high-order models. These methods employ Volterra kernel expansions, direct inversion and iterative estimation methods in connection with equivalent network model structures.

2.3.1. Volterra Kernel Expansions

The introduction of the kernel expansion approach was prompted by the fact that the use of the cross-correlation technique for kernel estimation revealed serious practical problems with the estimation of

multi-dimensonal (high-order) kernels. These problems are rooted in the unwieldy multi-dimensional representation of high-order kernels and the statistical variation of covariance estimates.

The Volterra kernel expansion approach presented in this section mitigates some of the problems of kernel estimation by compacting the kernel representation and avoiding the computation of cross-correlation (or covariance) averages, performing instead least-squares fitting of the actual data to estimate the expansion coefficients of the Volterra kernels. This alternative strategy also removes the whiteness requirements of the experimental input, although broadband inputs remain desirable. The difference between this approach and the direct least-squares estimation of Volterra kernels discussed in Section 2.1.5 is that the representation of the Volterra kernels in expansion form is more compact than the discrete-time formulation of Section 2.1.5, especially for systems with large memory-bandwidth products, as discussed below. The more compact kernel representations result in higher estimation accuracy and reduced computational burden, especially in high-order kernel estimation.

The basic kernel expansion methodology employs a properly selected basis of L causal functions $\{b_j(\tau)\}$ defined over the kernel memory, which can be viewed as the impulse response functions of a linear filterbank receiving the input signal $x(t)$. This basis is assumed to span completely and efficiently (i.e., with fast convergence) the kernel function space over $[0, \mu]$. The outputs $\{v_j(t)\}$ of this filterbank ($j = 1,..., L$) are given by:

$$v_j(t) = \int_0^{\mu} b_j(\tau)\, x(t-\tau)\, d\tau \tag{2.177}$$

and can be used to express the system output $y(t)$ according to the Volterra model of Equation (2.5), through a multinomial expression:

$$y(t) = k_0 + \sum_{r=1}^{Q} \sum_{j_1=1}^{L} \cdots \sum_{j_r=1}^{L} a_r(j_1,..., j_r)\, v_{j_1}(t) \cdots v_{j_r}(t) \tag{2.178}$$

which results from substituting the kernels in Equation (2.5) with their expansions:

$$k_r(\tau_1,..., \tau_r) = \sum_{j_1=1}^{L} \cdots \sum_{j_r=1}^{L} a_r(j_1,..., j_r)\, b_{j_1}(\tau_1) \cdots b_{j_r}(\tau_r) \tag{2.179}$$

where $\{a_r\}$ are the coefficients of the r th-order kernel expansion. Note that this modified expression of the Volterra model (using the kernel expansions) is isomorphic to the block-structured model of Figure 2.16, which is equivalent to the Volterra class of systems if and only if the kernel expansions of

Equation (2.179) hold for all r. In other words, the selected basis $\{b_j(\tau)\}$ must be complete for the expansion of the particular kernels of the system under study or, at least, provide adequate approximations for the requirements of the study. The kernel estimation problem thus reduces to the estimation of the unknown expansion coefficients using the expression (2.178) that is *linear* in terms of the unknown coefficients. Note that the signals $v_j(t)$ are known as convolutions of the input signal with the selected filterbank (see Equation (2.177)), but are available only in sampled form (discretized) in practice.

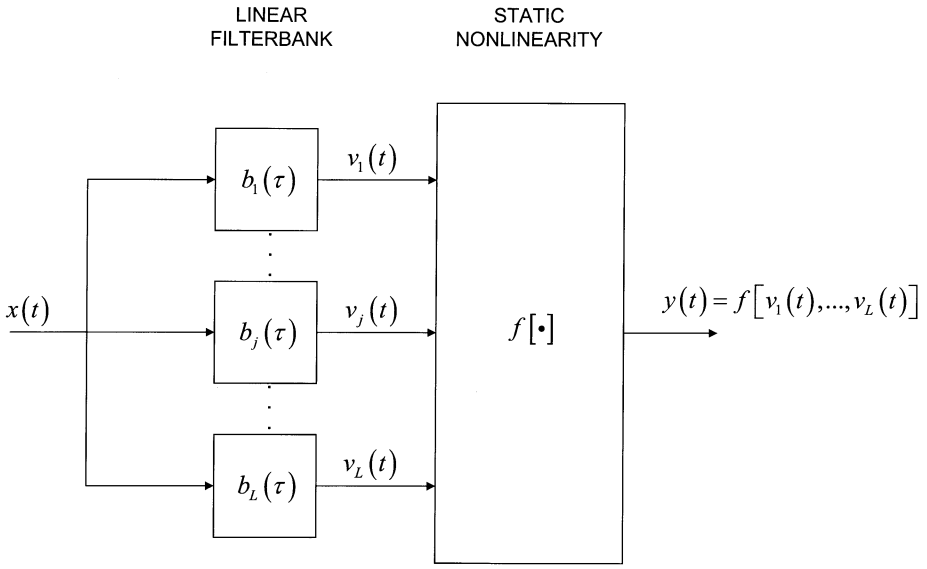


Figure 2.16
The modular form of the Volterra model, akin to the modular Wiener model of Figure 2.9. This modular model is isomorphic to the modified Volterra model of Equation (2.178), where the multi-input static nonlinearity $f[\square]$ is represented/ approximated by a multinomial expansion.

For this reason, the advocated methodology is cast in a discrete-time context by letting n replace t, m replace τ, and summation replace integral. This leads to the *modified discrete Volterra* (MDV) model:

$$y(n) = c_0 + \sum_{r=1}^{Q} \sum_{j_1=1}^{L} \cdots \sum_{j_r=1}^{j_{r-1}} c_r(j_1,...,j_r) v_{j_1}(n) \cdots v_{j_r}(n) + \varepsilon(n) \tag{2.180}$$

where the expansion coefficients $\{c_r\}$ take into account the symmetries of the Volterra kernels, i.e.,

$$c_r(j_1,...,j_r) = \lambda_r a_r(j_1,...,j_r) \tag{2.181}$$

where λ_r depends on the multiplicity of the specific indices $(j_1,...,j_r)$. For instance, if all indices are distinct, then $\lambda_r = 1$; but if the indices form p groups with multiplicities $m_i \left(i = 1,...,p, \text{ and } m_1 + ... + m_p = r\right)$, then $\lambda_r = m_1!...m_p!$. The error term $\varepsilon(n)$ incorporates possible model truncation errors and noise/interference in the data. The filterbank outputs are:

$$v_j(n) = T\sum_{m=0}^{M-1} b_j(m)x(n-m) \tag{2.182}$$

where M denotes the memory-bandwidth product of the system and T is the sampling interval.

For estimation efficiency, the number of required basis functions L must be much smaller than the memory-bandwidth product M of the system, which is the number of discrete samples required for representing each kernel dimension. With regard to the estimation of the unknown expansion coefficients, the methods discussed in Section 2.1.5 apply here as well, since the unknown coefficient vector $\boldsymbol{c}$ must be estimated from the matrix equation:

$$\mathbf{y} = \mathbf{V}\mathbf{c} + \boldsymbol{\varepsilon} \tag{2.183}$$

where the matrix $\mathbf{V}$ is constructed with the outputs of the filterbank according to Equation (2.180). Note that the symmetries of the kernels have been taken into account by requiring that $j_i \le j_{i-1}$ in Equation (2.180). For instance, for a second-order system, the nth row of the matrix $\mathbf{V}$ is: $\left\{1, v_1(n),...,v_L(n), v_1^2(n), v_2(n)v_1(n),...,v_L(n)v_1(n), v_2^2(n), v_3(n)v_2(n),...,v_L(n)v_{L-1}(n), v_L^2(n)\right\}$.

The number of columns in matrix $\mathbf{V}$ for a Qth-order MDV model is:

$$P = (L+Q)!/(L!Q!) \tag{2.184}$$

and the number of rows is equal to the number of output samples N. Solution of the estimation problem formulated in Equation (2.183) can be achieved by direct inversion of the square Gram matrix $\mathbf{G} = \left[\mathbf{V}'\mathbf{V}\right]$ if it is non-singular, or through pseudo-inversion of the rectangular matrix $\mathbf{V}$ if $\mathbf{G}$ is singular (or ill-conditioned), as discussed in Section 2.1.5. Iterative schemes based on gradient descent can also be used instead of matrix inversion, especially if the residuals are non-Gaussian, as discussed in Section 2.1.5. The number of columns of the matrix $\mathbf{V}$ determines the computational burden in the direct inversion approach and depends on the parameters L and Q, as indicated n Equation (2.184). Therefore, it is evident that practical solution of this estimation problem is subject to the "curse of dimensionality" for high-order systems, since the number of columns P in matrix $\mathbf{V}$ increases

geometrically with Q or L. For this reason, our efforts should aim at minimizing L by judicious choice of the expansion basis, since Q is an invariant system characteristic.

If the matrix $\mathbf{V}$ is full-rank, then the coefficient vector can be estimated by means of ordinary least-squares as:

$$\hat{\mathbf{c}} = \left[\mathbf{V}'\mathbf{V}\right]^{-1}\mathbf{V}'\mathbf{y} \tag{2.185}$$

This unbiased and consistent estimate is also efficient (i.e., it has minimum variance among all linear estimators) if the residuals are white (i.e., statistically independent with zero mean) and Gaussian. If the residuals are not white, then the generalized least-squares estimate of Equation (2.46) can be used, as discussed in Section 2.1.5. In that same section, an iterative procedure was described for obtaining efficient estimates when the residuals are not Gaussian, by minimizing a cost function defined by the log-likelihood function (not repeated here in the interest of space).

A practical problem arises when the matrix $\mathbf{V}$ is not full-rank or, equivalently, when the Gram matrix $\mathbf{G}$ is singular. In this case, a generalized inverse (or pseudo-inverse) $\mathbf{V}^{+}$ can be used to obtain the coefficient estimates as [Fan & Kalaba, 2003]:

$$\hat{\mathbf{c}} = \mathbf{V}^{+}\mathbf{y} \tag{2.186}$$

Another problematic situation arises when the Gram matrix $\mathbf{G}$ is ill-conditioned (a frequent occurrence in practice). In this case, a generalized inverse can be used or a reduced rank inverse can be found by means of singular-value decomposition to improve numerical stability. The resulting solution in the latter case depends on the selection of the threshold used for determining the "significant" singular values.

Model Order Determination

Of particular practical importance is the determination of the structural parameters L and Q of the MDV model, which determine the size of matrix $\mathbf{V}$. A statistical criterion for MDV model order selection has been developed by the author that proceeds sequentially in ascending order, as described below for the case of Gaussian white output-additive noise (i.e., the residual vector $\boldsymbol{\varepsilon}$ for the true model order). If the residuals are not white, then the prewhitening indicated by Equation (2.47) must be applied.

First we consider a sequence of MDV models of increasing order "r", where r is a sequential index sweeping through the double loop of increasing L and Q values (L being the fast loop). Thus, starting with ($Q=1,\ L=1$) for $r=1$, we go to ($Q=1,\ L=2$) for $r=2$ etc. After ($Q=1,\ L=L_{\max}$) for $r=L_{\max}$, we have ($Q=2,\ L=1$) for $r=L_{\max}+1$ etc. until we reach ($Q=Q_{\max},\ L=L_{\max}$) for $r=Q_{\max}\cdot L_{\max}$. For the true model order $r=R$ (corresponding to the true Q and L values), we have:

$$\mathbf{y}=\mathbf{V}_R\mathbf{c}_R+\boldsymbol{\varepsilon}_R \tag{2.187}$$

where the residual vector $\boldsymbol{\varepsilon}_R$ is white and Gaussian, and the coefficient vector $\boldsymbol{c}_R$ reconstructs the true Volterra kernels of the system according to:

$$k_q\left(m_1,...,m_q\right)=\sum_{j_1=1}^{L}...\sum_{j_q=1}^{j_{q-1}}c_q\left(j_1,...,j_q\right)b_{j_1}\left(m_1\right)...b_{j_q}\left(m_q\right) \tag{2.188}$$

However, for an incomplete order r, we have the truncated model:

$$\mathbf{y}=\mathbf{V}_r\mathbf{c}_r+\boldsymbol{\varepsilon}_r \tag{2.189}$$

where the residual vector contains portion of the input-output relationship and is given by:

$$\begin{aligned}\boldsymbol{\varepsilon}_r&=\left[\mathbf{I}-\mathbf{V}_r\left[\mathbf{V}_r'\mathbf{V}_r\right]^{-1}\mathbf{V}_r'\right]\mathbf{y}\\&\triangleq\mathbf{H}_r\mathbf{y}\end{aligned} \tag{2.190}$$

where the "projection" matrix $\mathbf{H}_r$ is idempotent (i.e., $\mathbf{H}_r^2=\mathbf{H}_r$) and of rank: ($N-P_r$), with P_r denoting the number of free parameters given by Equation (2.184) for the respective L and Q. Note that:

$$\begin{aligned}\boldsymbol{\varepsilon}_r&=\mathbf{H}_r\mathbf{H}_{r-1}^{+}\boldsymbol{\varepsilon}_{r-1}\\&\triangleq\mathbf{S}_r\boldsymbol{\varepsilon}_{r-1}\end{aligned} \tag{2.191}$$

which relates the residuals at successive model orders through the matrix $\mathbf{S}_r$ that depends on the input data. Since $\boldsymbol{\varepsilon}_0$ is the same as $\mathbf{y}$ (by definition), the r th-order residual vector can be expressed as:

$$\begin{aligned}\boldsymbol{\varepsilon}_r&=\mathbf{S}_r\mathbf{S}_{r-1}...\mathbf{S}_1\mathbf{y}\\&=\mathbf{H}_r\mathbf{y}\end{aligned} \tag{2.192}$$

because the concatenation of the linear operators $\left[\mathbf{S}_r\mathbf{S}_{r-1}...\mathbf{S}_1\right]$ simply reduces to $\boldsymbol{H}_r$. The residuals of an incomplete model r are composed of an input-dependent term $\boldsymbol{u}_r$ (the unexplained part of the system output) and a stochastic term $\boldsymbol{w}_r$ that represents the output-additive noise after being transformed by the matrix $\mathbf{H}_r$. Therefore:

$$\boldsymbol{\varepsilon}_r = \boldsymbol{u}_r + \boldsymbol{w}_r \tag{2.193}$$

where $\boldsymbol{u}_r$ is input-dependent and:

$$\boldsymbol{w}_r = \mathbf{H}_r \boldsymbol{w}_0 \tag{2.194}$$

$$\boldsymbol{u}_r = \mathbf{H}_r \boldsymbol{u}_0 \tag{2.195}$$

Note that $\boldsymbol{w}_0$ denotes the output-additive noise in the data and $\boldsymbol{u}_0$ represents the noise-free output data. For the true order R, we must have $\boldsymbol{u}_R = \mathbf{0}$ since all the input-dependent information in the output data has been captured and explained by the MDV model of order R.

The expected value of the sum of the squared residuals (SSR) at order r is given by:

$$\begin{aligned} E\left[\Omega_r\right] &= E\left[\boldsymbol{\varepsilon}_r' \boldsymbol{\varepsilon}_r\right] \\ &= \boldsymbol{u}_r' \boldsymbol{u}_r + \sigma_0^2 \mathrm{Tr}\left\{\mathbf{H}_r' \mathbf{H}_r\right\} \end{aligned} \tag{2.196}$$

where σ_0^2 is the variance of the initial white residuals, and $Tr\{\cdot\}$ denotes the trace of the subject matrix. The latter is found to be:

$$\begin{aligned} \mathrm{Tr}\left\{\mathbf{H}_r' \mathbf{H}_r\right\} &= N - P_r \\ &= N - \left(L_r + Q_r\right)! / \left(L_r ! Q_r !\right) \end{aligned} \tag{2.197}$$

where L_r and Q_r are the structural parameter values that correspond to model order r.

To test whether r is the true order of the system, we postulate the null hypothesis that *it is* the true order, which implies that $\boldsymbol{u}_r = \mathbf{0}$, and an estimate of the initial noise variance can be obtained from the computed SSR as:

$$\sigma_0^2 = \Omega_r / \left(N - P_r\right) \tag{2.198}$$

Then the expected value of the reduction in the computed SSR for the next order $(r+1)$ is:

$$E\left[\Omega_r - \Omega_{r+1}\right] = \sigma_0^2 \cdot \left(P_{r+1} - P_r\right) \tag{2.199}$$

under this null hypothesis. Therefore, using the estimate of σ_0^2 given by Equation (2.198), we see that this reduction in computed SSR ought to be greater than the critical value:

$$\theta_r = \lambda \Omega_r \frac{\left(P_{r+1} - P_r\right)}{\left(N - P_r\right)} \tag{2.200}$$

to reject this null hypothesis and continue the search to higher orders. We can select: $\lambda = 1 + 2\sqrt{2}$, because the SSR follows approximately a chi-square distribution. If the reduction in computed SSR is

smaller than the critical value given by Equation (2.200), then the order r is accepted as the true order of the MDV model for this system; otherwise we proceed with the next order ($r+1$) and repeat the test.

As an alternative approach, we can employ a constrained configuration of the MDV model in the form of equivalent feedforward networks, that reduce significantly the number of free parameters for *high-order* systems regardless of the value of L, as discussed in Section 2.3.3. However, the estimation problem ceases to be linear in terms of the unknown parameters and estimation of the network parameters is achieved with iterative schemes based on gradient descent, as discussed in Chapter 4. The use of gradient-based iterative estimation of system kernels was first attempted with "stochastic approximation" methods [Groussard et al. 1991], but it became more attractive and more widely used with the introduction of the equivalent network models discussed in Section 2.3.3. It should be noted that these methods offer computational advantages when the size of the Gram matrix $\mathbf{G}$ becomes very large, but they are exposed to problems of convergence (speed and local minima). They also allow robust estimation in the presence of noise outliers (i.e., when the model residuals are not Gaussian and exhibit occasional outliers) by utilizing non-quadratic cost functions compatible with the log-likelihood function of the residuals, since least-squares estimation methods are prone to serious errors in those cases (as discussed in Sections 2.1.5 and 2.4.4).

The key to the efficacy of the kernel expansion approach is in finding the proper basis $\{b_j\}$ that reduces the number L to a minimum for a given application (since Q is fixed for a given system, the number of free parameters depends only on L). The search for the putative "minimum set" of such basis functions may commence with the use of a general complete orthonormal basis (such as the Laguerre basis discussed in Section 2.3.2 below) and advance with the notion of "principal dynamic models" (discussed in Section 4.1.1) in order to extract the significant dynamic components of the specific system and eliminate spurious or insignificant terms.

2.3.2 THE LAGUERRE EXPANSION TECHNIQUE

To date the best implementation of the kernel expansion approach (discussed in the previous section) has been made with the use of discrete Laguerre functions. The Laguerre basis (in continuous time) was Wiener's original suggestion for expansion of the Wiener kernels, because the Laguerre functions are orthogonal over a domain from zero to infinity (consistent with the kernel domain) and have a built-in exponential (consistent with the relaxation dynamic characteristics of physical systems). Furthermore, the Laguerre expansions can be easily generated in continuous time with analog means (ladder R-C

network) that enhanced their popularity at that time when analog processing was in vogue. With the advancing digital computer technology, the data processing focus shifted to discretized signals and, consequently, to sampled versions of the continuous-time Laguerre functions. The first applications of Laguerre kernel expansions were made in the early seventies on the eye pupil reflex [Watanabe & Stark, 1975] and on hydrological rainfall-runoff processes [Amorocho & Branstetter, 1971] using sampled versions of the continuous-time Laguerre functions. Note that these discretized Laguerre functions are distinct from the "discrete Laguerre functions" (DLFs) advocated herein, which are constructed to be orthogonal in discrete time [Ogura, 1985] (the discretized versions are not generally orthogonal in discrete time but tend to orthogonality as the sampling interval tends to zero). Wide application of the Laguerre kernel expansion approach commenced in the nineties with the successful introduction of the DLFs discussed below [Marmarelis, 1993].

The Laguerre expansion technique (LET) for Volterra kernel estimation is cast in discrete time by the use of the orthonormal set of discrete Laguerre functions (DLFs) given by [Ogura, 1985]:

$$b_j(m) = \alpha^{(m-j)/2}(1-\alpha)^{1/2}\sum_{k=0}^{j}(-1)^k\binom{m}{k}\binom{j}{k}\alpha^{j-k}(1-\alpha)^k \qquad (2.201)$$

where $b_j(m)$ denotes the jth-order orthonormal DLF, the integer m ranges from 0 to $M-1$ (the memory-bandwidth product of the system), and the real positive number α $(0<\alpha<1)$ is the critical DLF parameter which determines the rate of exponential (asymptotic) decline of these functions. We consider a DLF filter-bank (for $j=0,1,...,L-1$) receiving the input signal $x(n)$ and generating at the output of each filter the key variables $\{v_j(n)\}$ as the discrete-time convolutions given by Equation (2.182) between the input and the respective DLF.

Since the sampled versions of the traditional continuous-time Laguerre functions, that were originally proposed by Wiener and used by Watanabe and Stark in the first known application of Laguerre kernel expansions to physiological systems [Watanabe & Stark, 1975], are not necessarily orthogonal in discrete time (depending on the sampling interval), Ogura constructed the DLFs to be orthogonal in discrete time and introduced their use in connection with Wiener-type kernel estimation that involves the computation of covariances by time-averaging over the available data-records (with all the shortcomings of the covariance approach discussed earlier) [Ogura, 1985]. The advocated LET combines Ogura's DLFs (which are easily computed using an auto-recursive relation) with least-squares fitting (instead of covariance computation) as was initially done by Stark and Watanabe. An important

enhancement of the LET was introduced by the author in terms of adaptive estimation of the critical Laguerre parameter α from the input-output data, as elaborated below. This is a critical issue in actual applications of the Laguerre expansion approach because it determines the rate of convergence of the DLF expansion and the efficiency of the method.

It should be noted that LET was introduced in connection with Volterra kernel estimation but could be also used for Wiener kernel estimation if the Wiener series is employed as the model structure. However, there is no apparent reason to use the Wiener series formulation, especially when a complete (non-truncated) model is employed. Of course, the Volterra formulation *must* be used when the input signal is non-white.

When the discretized Volterra kernels of the system are expanded on the DLF basis as:

$$k_r(m_1,...,m_r) = \sum_{j_1=0}^{L-1} ... \sum_{j_r=0}^{j_{r-1}} c_r(j_1,...,j_r) b_{j_1}(m_1) ... b_{j_r}(m_r) \tag{2.202}$$

then the resulting MDV model of Qth-order is given by Equation (2.180), where a finite number L of DLFs is used to represent the kernels as in Equation (2.202).

The task of Volterra system modeling now reduces to estimating the unknown kernel expansion coefficients $\{c_r(j_1,...,j_r)\}$ from Equation (2.180), where the output data $\{y(n)\}$ and the transformed input data $\{v_j(n)\}$ (which are the outputs of the DLF filter-bank) are known. This task can be performed either through direct inversion of the matrix $\mathbf{V}$ in the formulation of Equation (2.183) or using iterative (gradient based) techniques that can be more robust when the errors are non-Gaussian (with possible outliers) and/or the size of the $\mathbf{V}$ matrix is very large.

The total number of free parameters P that must be estimated (which is equal to the number of columns of matrix $\mathbf{V}$ given by Equation (2.184) remains the critical consideration with regard to estimation accuracy and computational burden. Estimation accuracy generally improves as the ratio P/N decreases, where N is the total number of input-output data points. The computational burden increases with P or N, but is more sensitive to increases in P. Thus, minimizing L is practically important in order to minimize P, since the model order Q is dictated by the nonlinear characteristics of the system (beyond our control). A complete model (i.e., when Q is the highest-order of significant nonlinearity in the system) is required in order to avoid estimation biases in the obtained Volterra kernel estimates. This also alleviates the strict requirement of input whiteness, and thus naturally occurring data can be used for kernel estimation.

The computations required for given L and Q can be reduced by computing the key variables $\{v_j(n)\}$ using the auto-recursive relation [Ogura, 1985]:

$$v_j(n) = \sqrt{\alpha} v_j(n-1) + \sqrt{\alpha} v_{j-1}(n) - v_{j-1}(n-1) \tag{2.203}$$

which is due to the particular form of the DLF. Computation of this auto-recursive relation must be initialized by the following auto-recursive equation of first order that yields $v_0(n)$ for given stimulus $x(n)$:

$$v_0(n) = \sqrt{\alpha} v_0(n-1) + T\sqrt{1-\alpha}\, x(n) \tag{2.204}$$

where T is the sampling interval. These computations can be preformed rather fast for $n = j,...,N$ and $j = 0,1,...,L-1$; where L is the total number of DLF used in the kernel expansion. Note also the symmetry between j and n, i.e., $v_j(n) = v_n(j)$.

The choice of the Laguerre parameter α is critical in achieving efficient kernel expansions (i.e., minimize L) and, consequently, fast and accurate kernel estimation. Its judicious selection was initially made on the basis of the parameter values L and M [Marmarelis, 1993] or by successive trials. However, this author recently proposed its estimation through an iterative adaptive scheme based on the auto-recursive relation of Equation (2.203). This estimation task is embedded in the broader estimation procedure for the kernel expansion coefficients using iterative gradient-based methods.

Specifically, we seek the minimization of a selected non-negative cost function F (usually the square of the residual term in Equation (2.180) by means of the iterative gradient-based expression:

$$c_r^{(i+1)}(j_1,...j_r) = c_r^{(i)}(j_1,...,j_r) - \gamma \left. \frac{\partial F(\varepsilon)}{\partial c_r(j_1,...,j_r)} \right|_{\varepsilon=\varepsilon^{(i)}} \tag{2.205}$$

where γ denotes the adaptation step (learning constant), i is the iteration index, and the gradient $\partial F / \partial c_r$ is evaluated for the ith-iteration error (i.e., the model residual or prediction error is computed from Equation (2.180) for the ith-iteration parameter estimates). In the case of the Laguerre expansion, the iterative estimation of the expansion coefficients based on the expressions (2.205) can be combined with the iterative estimation of the square-root of the DLF parameter $\beta = \alpha^{1/2}$:

$$\beta^{(i+1)} = \beta^{(i)} - \gamma_\beta \left. \frac{\partial F(\varepsilon)}{\partial \beta} \right|_{\varepsilon=\varepsilon^{(i)}} \tag{2.206}$$

Clearly, changes in β (or $\alpha = \beta^2$) alter the values of the key variables $\{v_j(n)\}$ according to the auto-recursive relation (2.202):

$$\frac{\partial v_j(n)}{\partial \beta} = v_j(n-1) + v_{j-1}(n) \tag{2.207}$$

This simple gradient relation (with respect to $\beta = \alpha^{1/2}$) allows iterative estimation of α along with the expansion coefficients in a practical context. For instance, in the case of a quadratic cost function: $F = \varepsilon^2$ (corresponding to least-squares fitting), the gradient components are:

$$\frac{\partial F(\varepsilon)}{\partial c_r(j_1,...,j_r)} = -2\varepsilon(n) v_{j_1}(n)...v_{j_r}(n) \tag{2.208}$$

and:

$$\frac{\partial F}{\partial \alpha^{1/2}} = -2\varepsilon(n) \sum_{r=1}^{Q} \sum_{j_1=0}^{L} ... \sum_{j_r=0}^{j_{r-1}} \sum_{p=j_1}^{jr} c_r(j_1,...,j_r) v_{j_1}(n)...\left[v_p(n-1) + v_{p-1}(n)\right]...v_{j_r}(n) \tag{2.209}$$

Although this procedure appears to be laborious for high-order models, it becomes drastically accelerated when an equivalent network structure is used to represent the high-order model, as discussed in Section 2.3.3.

It is instructive to illustrate certain basic properties of the DLFs. The first 5 DLFs for $\alpha = 0.2$ are shown in Figure 2.17 $(T = 1)$. We note that the number of zero-crossing (roots) of each DLF equals its order. Furthermore, the higher the order the longer the significant values of a DLF spread over time, and the time separation between zero-crossing increases. This is further illustrated in Figure 2.18, where the DLFs of order 4, 8, 12, and 16 are shown for $\alpha = 0.2$.An interesting illustration of the first 50 DLFs for $\alpha = 0.2$is given in Figure 2.19, plotted as a square matrix over 50 time lags in 3-D perspective (top display) and contour plot (bottom display). We note the symmetry of this matrix and the fact that higher order DLFs are increasingly "delayed" in their significant values. Note that Volterra kernels with a pure delay may require use of "associated DLFs" of appropriate order for efficient kernel representation [Ogura, 1985].

In the frequency domain, the FFT magnitude of all DLFs for a given α is identical, as illustrated in Figure 2.20 for the 5 DLFs of Figure 2.17. However, the FFT phase of these DLFs is different, as illustrated also in Figure 2.20. We note that the nth-order DLF exhibits a maximum phase shift of $(n\pi)$, with the odd-order ones commencing at π radians and the even order ones commencing at 0 radians (at zero frequency). Thus, the "minimum-phase" DLF is always for order zero.

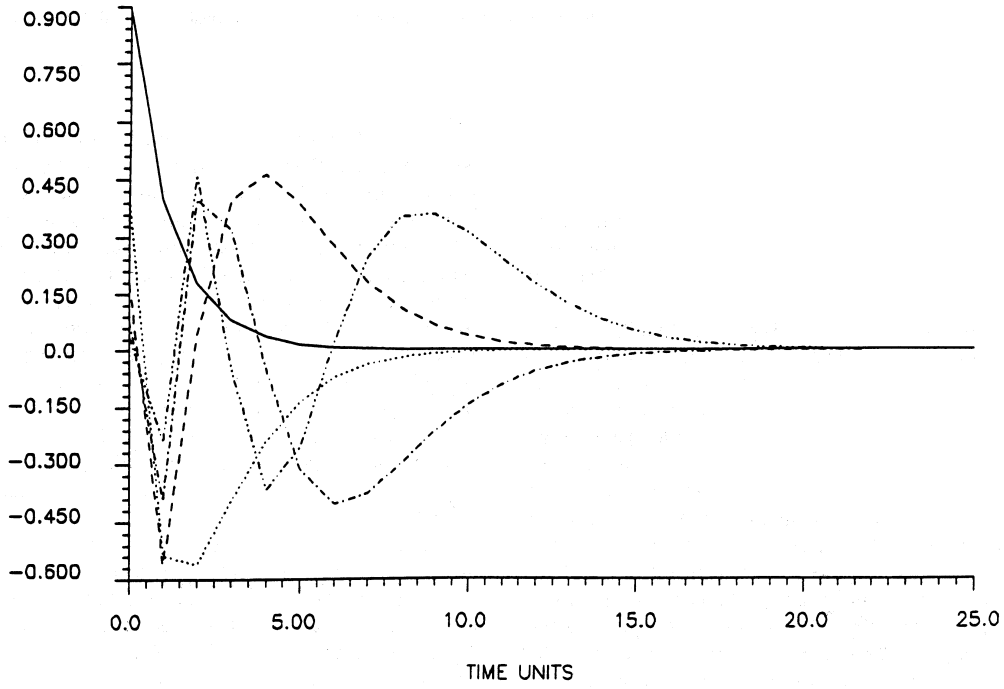


Figure 2.17
Discrete-time Laguerre functions (DLF) of order 0 (solid), 1 (dotted), 2 (dashed), 3 (dot-dash), 4 (dot-dot-dot-dash) for $\alpha = 0.2$, plotted over the first 25 lags [Marmarelis, 1993].

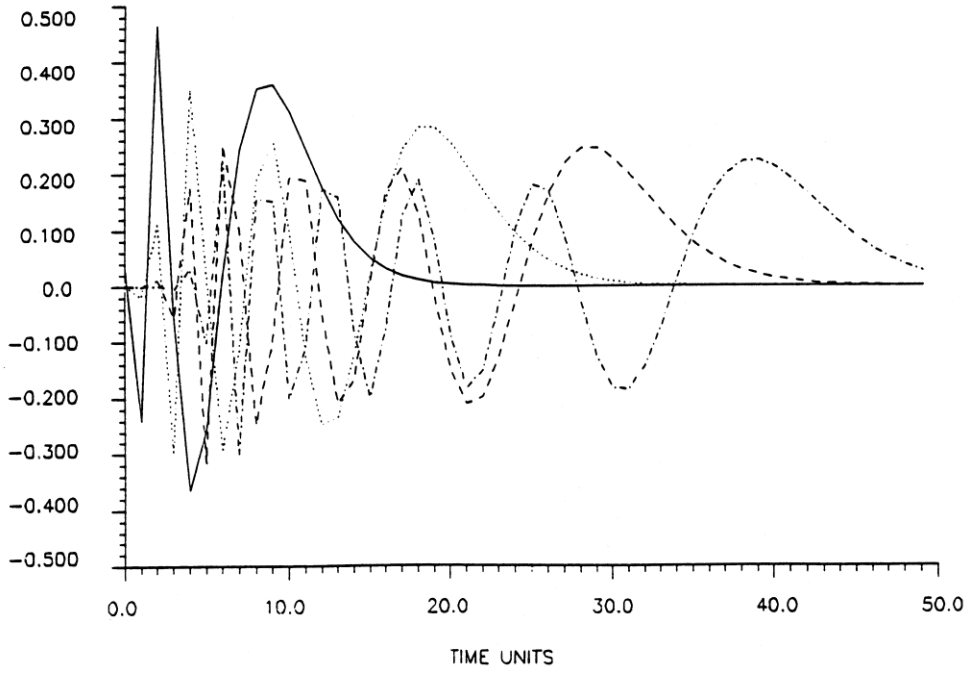


Figure 2.18
Discrete-time Laguerre functions (DLF) of order 4 (solid), 8 (dotted), 12 (dashed), 16 (dot-dash) for $\alpha = 0.2$, plotted over the first 25 lags [Marmarelis, 1993].

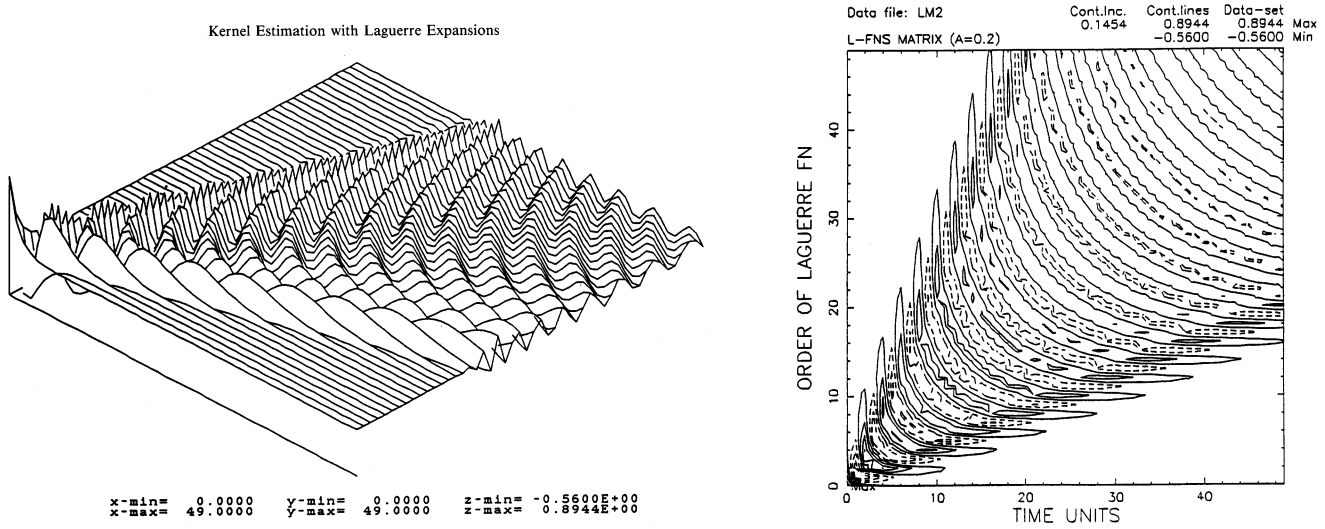
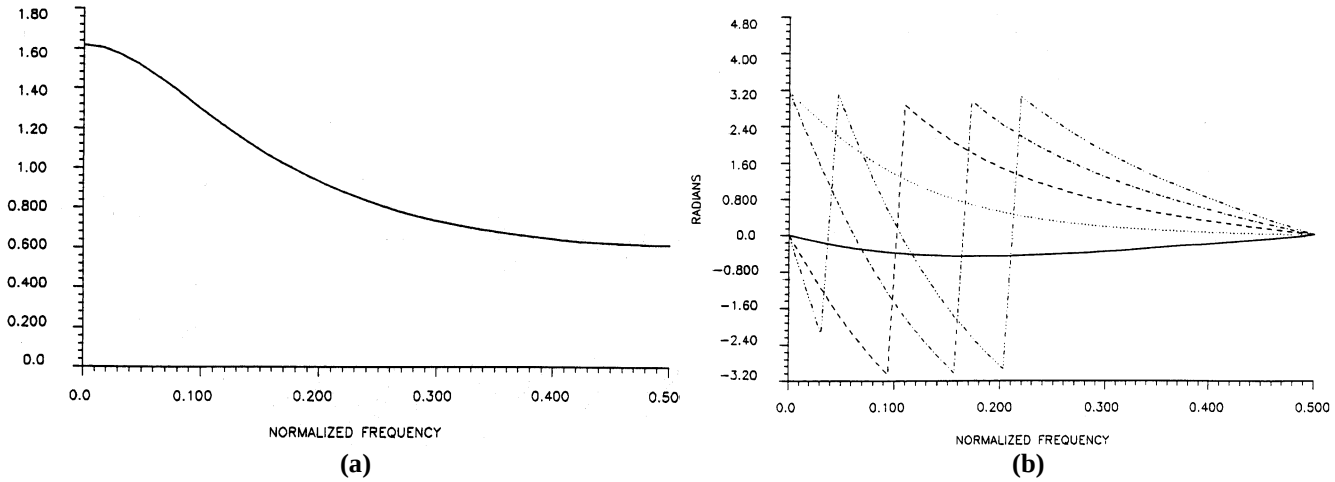


Figure 2.19
The first 50 DLFs for $\alpha = 0.2$, plotted form 0 to 49 lags in 3-D perspective plot (left panel) and contour plot (right panel) [Marmarelis, 1993].



(a) **(b)**

Figure 2.20
(a)FFT magnitude of the first 5 (orders 0 to 4) DLFs (shown in Figure 2.17) for $\alpha = 0.2$, plotted up to the normalized Nyquist frequency of 0.5 [Marmarelis, 1993]. (b) FFT phase of the first 5 (orders 0 to 4) DLFs (shown in Figure 2.17) for $\alpha = 0.2$, plotted up to the normalized Nyquist frequency of 0.5 [Marmarelis, 1993].

In order to examine the effect of α on the form of the DLFs, we show in Figure 2.21 the 4^{th}-order DLF for $\alpha = 0.1,\ 0.2,$ and 0.4. We observe that increasing α results in longer spread of significant values and zero-crossings. Thus, kernels with longer memory may require a larger α for efficient representation.

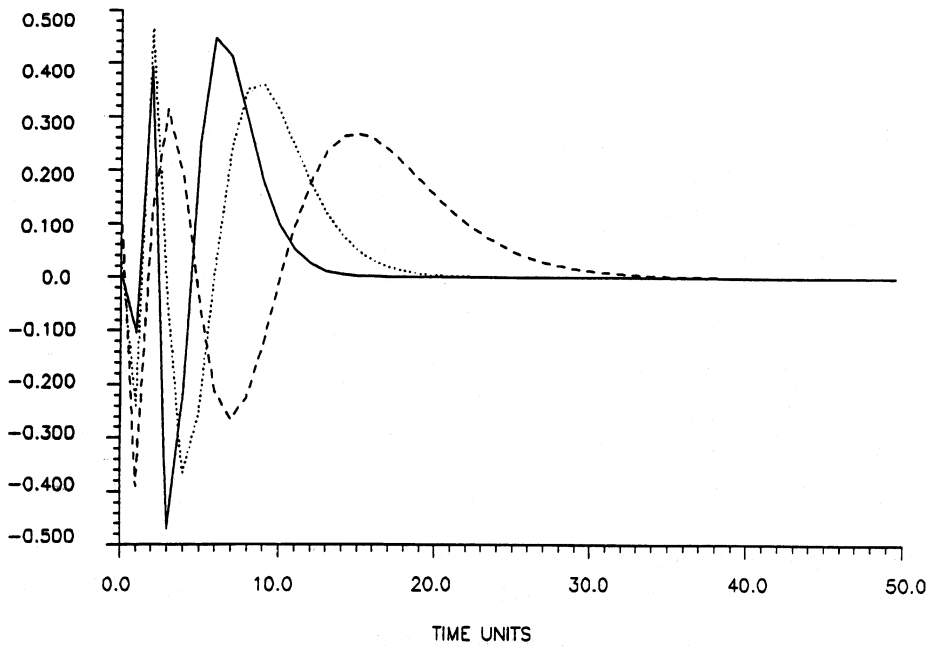
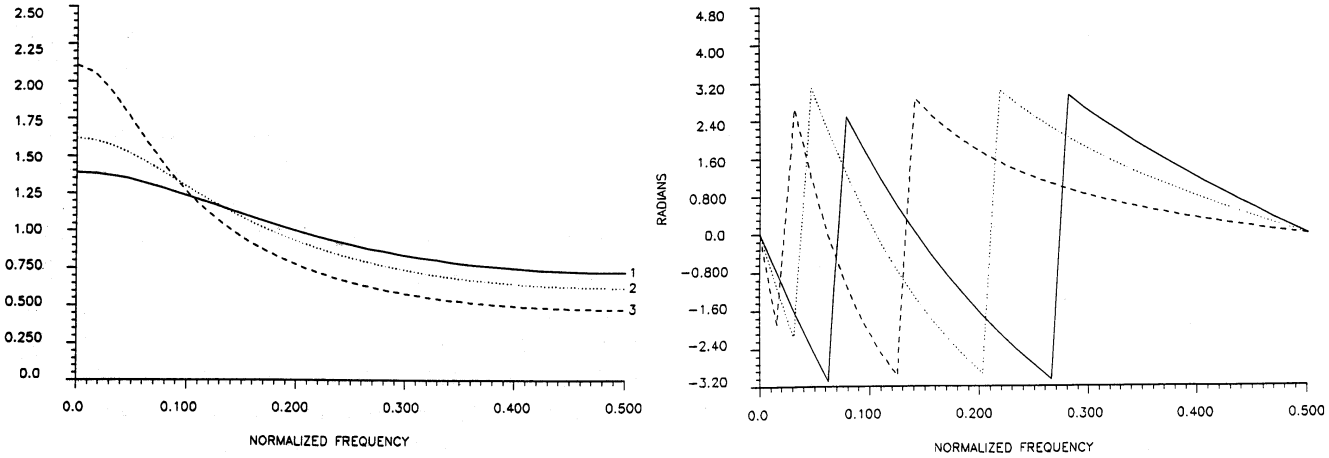


Figure 2.21
The 4th-order DLFs for $\alpha = 0.2$ (solid), 0.2n (dotted), 0.4 (dashed), plotted from 0 to 49 lags [Marmarelis, 1993].

In the frequency domain, the FFT magnitudes and phases of the DLFs of Figure 2.21 are shown in Figure 2.22. We observe that for larger α, the lower frequencies are emphasized more and the phase lags more rapidly, although the total phase shift is the same (4π for all 4th-order DLFs).



(a) (b)

Figure 2.22
(a) FFT magnitude of the 4th-order DLFs shown in Figure 2.21, for $\alpha = 0.1$ (solid), $\alpha = 0.2$ (dotted), $\alpha = 0.4$ (dashed) [Marmarelis, 1993]. (b) FFT phase of the 4th-order DLFs shown in Figure 2.21, for $\alpha = 0.1$ (solid), $\alpha = 0.2$ (dotted), $\alpha = 0.4$ (dashed) [Marmarelis, 1993].

A more complete illustration of the effect of α is given in Figure 2.23, where the matrix of the first 50 DLFs for $\alpha = 0.1$ is plotted in the same fashion as in Figure 2.19 (for $\alpha = 0.2$). It is clear from these figures that increasing α increases the separation between the zero crossings (ripple in 3-D perspective plots) and broadens the "fan" formation evident in the contour plots.

These observations provide an initial guide in selecting an approximate value of α for given kernel memory-bandwidth product (M) and number of DLFs (L). For instance, we may select the value α for which the significant values of the DLFs extend from 0 to M while, at the same time, the DLFs values are diminished for $m \square M$ (in order to secure their orthogonality). In other words, for given values M and L, α must be chosen so that the point (M,L) in the contour plane be near the edge of the "fan formation" but outside this "fan".

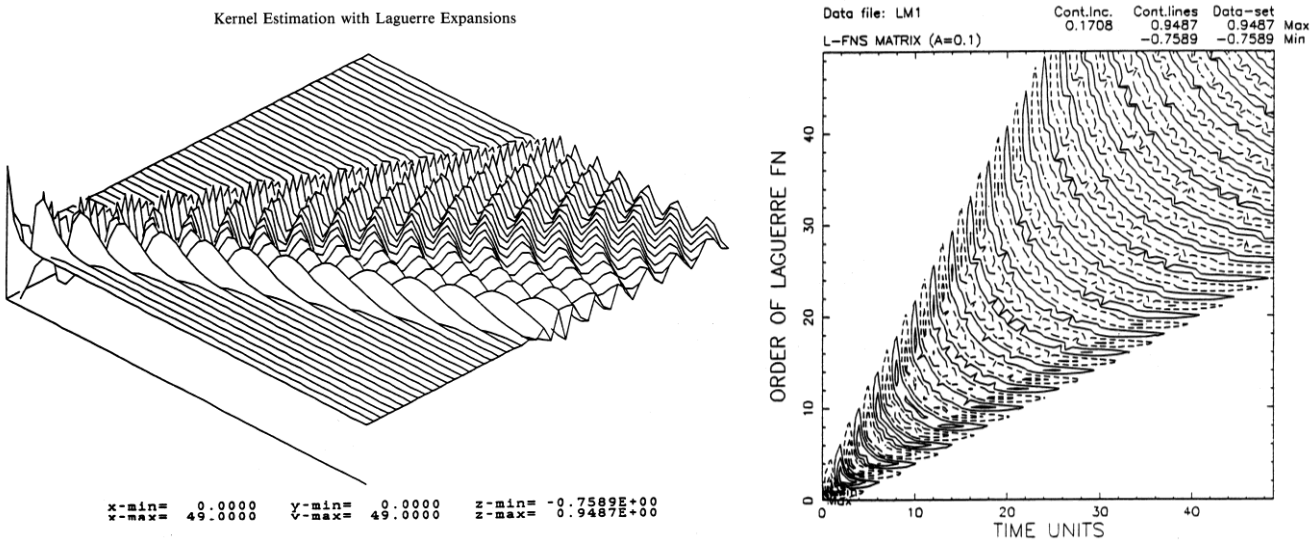
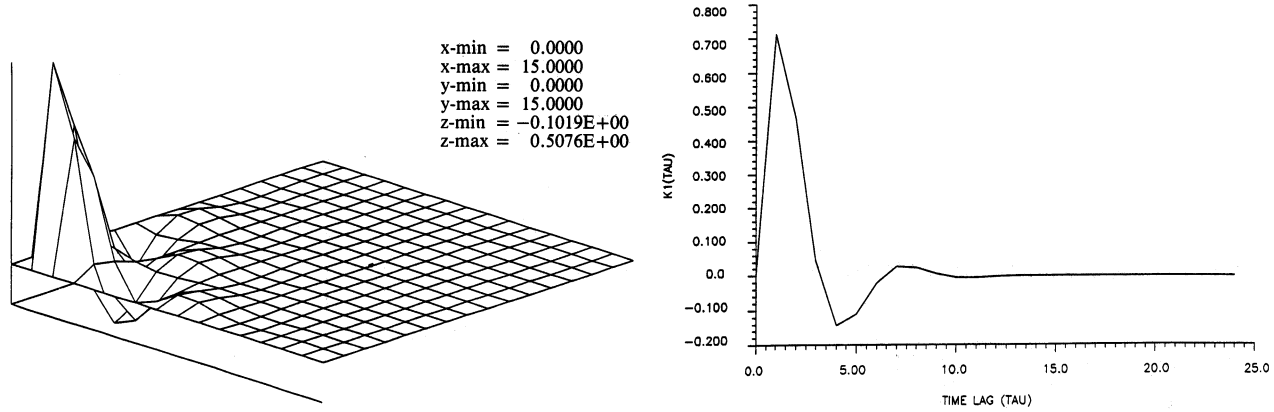


Figure 2.23
The first 50 DLFs for $\alpha = 0.1$, plotted from 0 to 49 lags in 3-D perspective plot (top panel) and contour plot (bottom panel) [Marmarelis, 1993].

Illustrative Examples

We now demonstrate the use of DLF expansions for kernel estimation. Consider first a second-order nonlinear system with the 1st- and 2nd-order Volterra kernels shown in Figure 2.24, corresponding to a simple $L-N$ cascade. This system is simulated for a band-limited GWN input of 512 datapoints, and the kernels are estimated using both the cross-correlation technique (CCT) and the advocated Laguerre expansion technique (LET). The 1st-order kernel estimates are shown in Figure 2.25, where the LET estimate is plotted with solid line (almost exactly identical to the true kernel) and the CCT estimate is

plotted with dashed line. The 2nd-order kernel estimates are shown in Figure 2.26, demonstrating the superiority of the LET estimates. Note that for a second-order system the Volterra and Wiener kernels are identical (except for the zeroth order one). Thus, the Volterra kernel estimates obtained via LET can be directly compared with Wiener kernel estimates obtained via CCT.



(a) (b)

Figure 2.24
(a) Exact 1st-order kernel of simulated system [Marmarelis, 1993]. (b) Exact 2nd-order kernel of simulated system [Marmarelis, 1993].

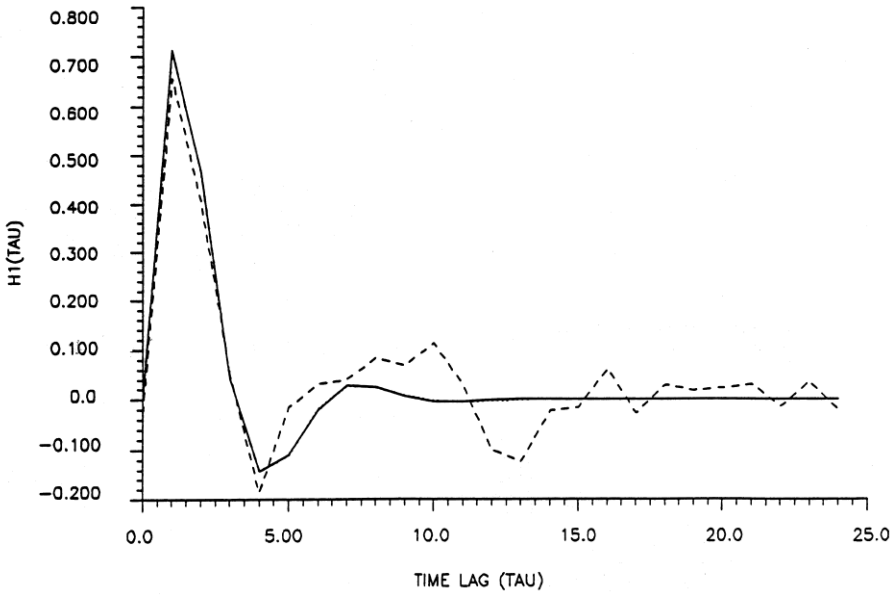
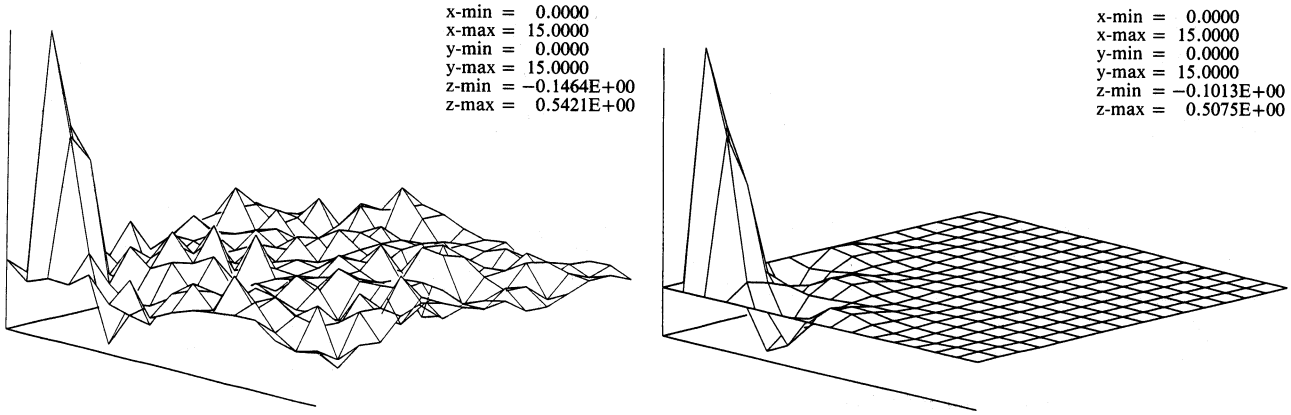


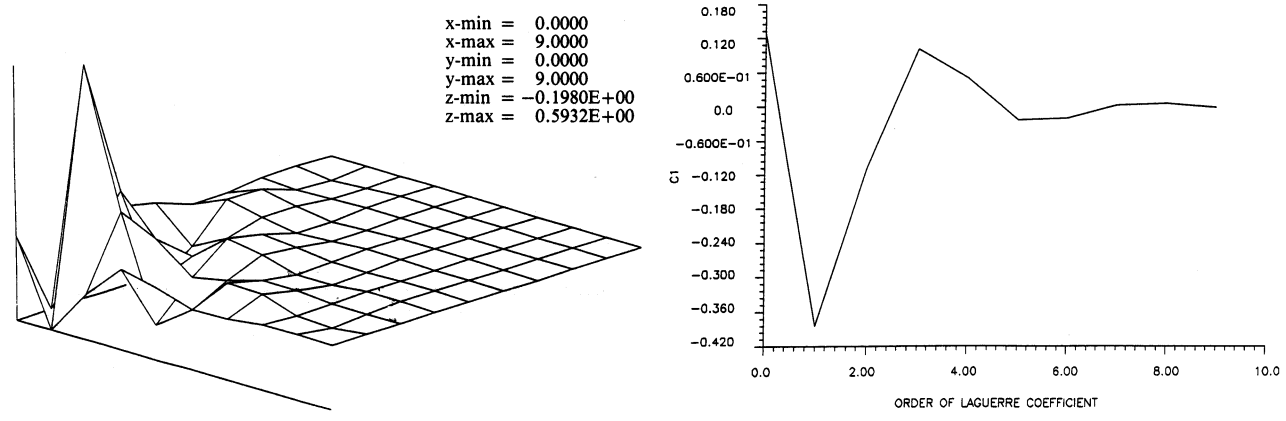
Figure 2.25
The estimated first-order kernel via LET (solid line) and CCT (dashed line) [Marmarelis, 1993].

These LET estimates were obtained for $L = 10$ and $\alpha = 0.1$, and the estimates of the Laguerre coefficients in this case are shown in Figure 2.27 for the 1st-order and 2nd-order kernels. Note that the 2nd-order coefficients are plotted as a symmetric matrix, even though only the entries of one triangular region are estimated. It is evident from Figure 2.27 that an adequate estimation of these kernels can also be accomplished with $L = 8$ (due to the very small values of the 9th and 10th coefficients), which demonstrates the significant compactness in kernel representation accomplished by the Laguerre expansion (for $M = 50$ and $L = 8$, the savings factor is about 35 for a 2nd-order model and grows rapidly for higher-order models).



(a) (b)

Figure 2.26
(a) 2nd-order kernel estimated via LET [Marmarelis, 1993]. (b) 2nd-order kernel estimated via CCT [Marmarelis, 1993].



(a) (b)

Figure 2.27
(a) Estimates of the first 10 (order 0 to 9) Laguerre expansion coefficients for the 1st-order kernel [Marmarelis, 1993].
(b) Estimates of the Laguerre expansion coefficients (order 0 to 9 in each dimension) for the 2nd-order kernel, plotted as a symmetric 2-D array (total number of distinct coefficients is 55) [Marmarelis, 1993].

Recall that the number L of required DLFs for accurate representation of the kernels of a given system critically affects the computational burden associated with this method. In general, the total number of resulting expansion coefficients (free parameters of the model) for a system of order Q is: $(L+Q)!/(L!Q!)$. Note that this number includes all kernels up to order Q and takes into account the symmetries in high-order kernels.

The required L in a given application can be determined by the statistical procedure described in the previous section (see Equation (2.200)) or can be empirically selected by estimating the 1st-order kernel for a large L, and then by inspecting the obtained coefficient estimates we can select the minimum number that corresponds to significant values. The same reasoning can be applied to high-order kernels (e.g., the 2nd-order coefficients shown in Figure 2.27), although the pattern of convergence of the Laguerre expansion depends on α -- a fact that prompted the introduction of the iterative scheme (discussed above) for estimating α (along with the expansion coefficients) on the basis of the data. Selection of the required maximum kernel order Q can also be made using the statistical procedure of Equation (2.200) or can be empirically based on preliminary tests or on the adequacy of the output prediction accuracy of a given model order (as in all previous applications of the Volterra-Wiener approach). We strongly recommend the selection of the key model parameters L and Q using the statistical criterion of Equation (2.200) and successive trials in ascending order, as discussed above. The use of the popular criteria (e.g., Akaike Information Criterion or Minimum Description Length), although rather common in established practice, is discouraged as potentially misleading.

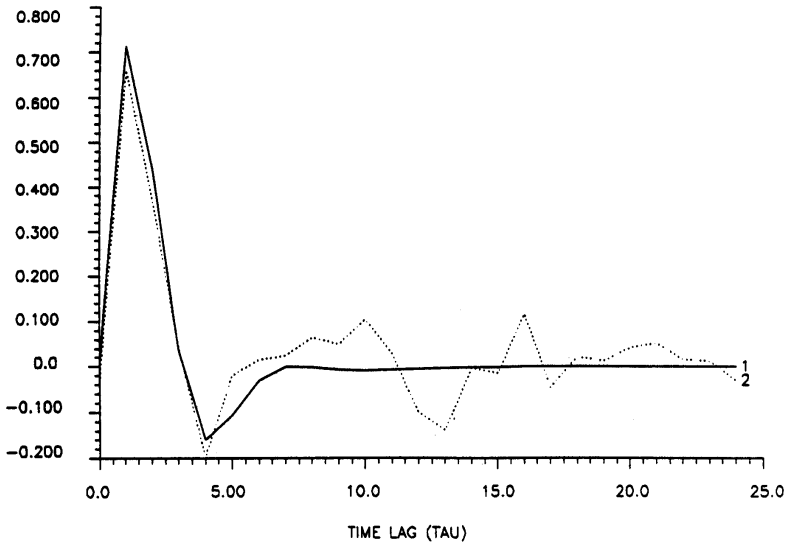


Figure 2.28
1st-order kernel estimates obtained via LET (solid) and CCT (dotted) for noisy data (SNR=10dB) [Marmarelis, 1993].

We now examine the effect of noise on the obtained kernel estimates by adding independent GWN to the output signal for a signal-to-noise ratio (SNR) of 10 dB. The 1st-order kernel estimates obtained by LET and CCT are shown in Figure 2.28 in solid line for LET and in dotted line for CCT. The corresponding 2nd-order kernel estimates are shown in Figure 2.29 and demonstrate the superiority of the LET approach in terms of robustness under noisy conditions. Note that the LET estimates in this demonstration were computed with $L = 8$, and required very short computing time. The fact that kernels estimates of this quality can be obtained from such short data-records (512 input-output data points), even in cases with significant noise (SNR=10 dB), can have important implications in actual applications of the advocated modeling approach to physiological systems.

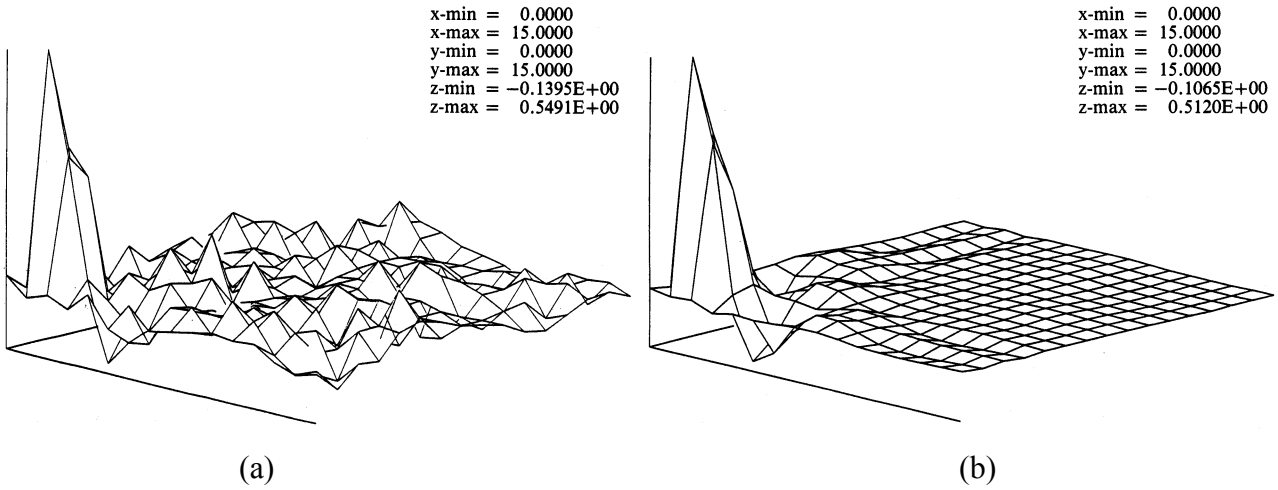


Figure 2.29
(a) 2nd-order kernel estimated obtained via LET for noisy data [Marmarelis, 1993].
(b) 2nd-order kernel estimated obtained via CCT for noisy data [Marmarelis, 1993].

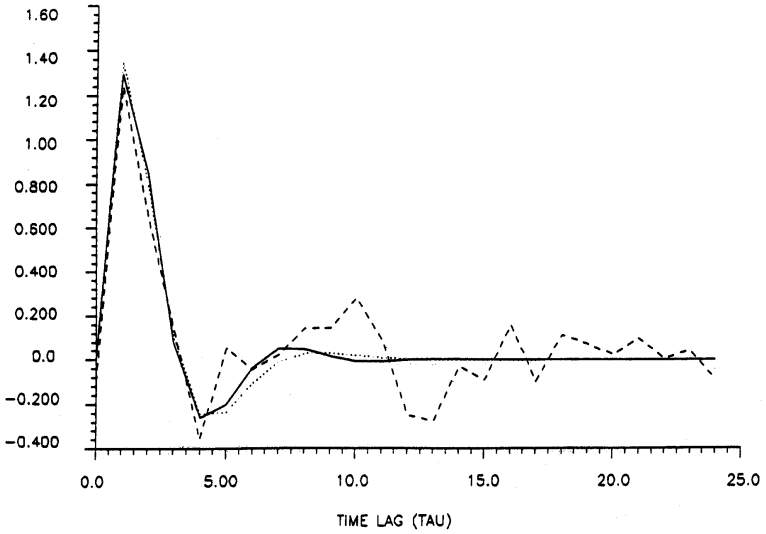


Figure 2.30

1st-order kernel estimates for 4th-order simulated system obtained via LET (dotted) and CCT (dashed). The exact kernel is plotted with solid line and is almost identical with the LET estimate [Marmarelis, 1993].

Another important issue in actual applications is the effect of higher order nonlinearities on the obtained lower order kernel estimates when a truncated (incomplete) Volterra model is used. Since most applications limit themselves to the first two kernels, the presence of higher-order (>2) Volterra functionals acts as a source of "correlated noise" which is dependent on the input. To illustrate this effect, we add 3rd-order and 4th-order nonlinearities to the previous system and recompute the kernel estimates. Note that the simulated system is a simple $L-N$ cascade of a linear filter followed by a static nonlinearity of the form: $y = z + z^2$ (for the 2nd-order system) and $y = z + z^2 + z^3/3 + z^4/4$ (for the 4th-order system), where $z(t)$ is the output of the linear filter. The obtained 1st-order kernel estimates are shown in Figure 2.30, where the exact Volterra kernel is plotted with solid line, the LET estimate with dotted line, and the CCT estimate with dashed line. The LET estimate is much better than its CCT counterpart, but it exhibits certain minor deviations from the exact Volterra kernel due to the presence of the 3rd-order and 4th-order terms. The exact 2nd-order Volterra kernel of the 4th-order system has the same form as in Figure 2.24, but its 2nd-order Wiener kernel is scaled by a factor of 2.23 because of the presence of the 4th-order nonlinearity (see Equation (2.57) for $P=1$). The 2nd-order kernel estimates obtained via LET and CCT are shown in Figure 2.31 and demonstrate that the LET estimate is better than its CCT counterpart. Note that the LET estimate closely resembles the exact Wiener or Volterra kernel in form (since both have the same shape in this simulated system), but has the size of the Wiener kernel because the input is GWN and the estimated model is truncated to second order.

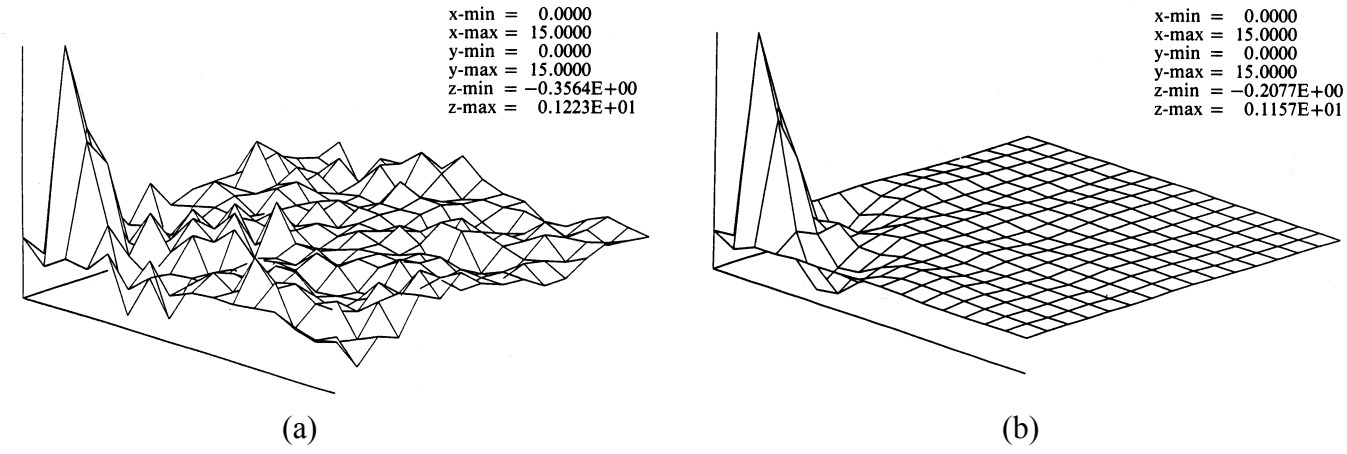


Figure 2.31
(a) 2nd-order kernel estimated for the 4th-order simulated system obtained via LET [Marmarelis, 1993].

(b) 2nd-order kernel estimated for the 4th-order simulated system obtained via CCT [Marmarelis, 1993].

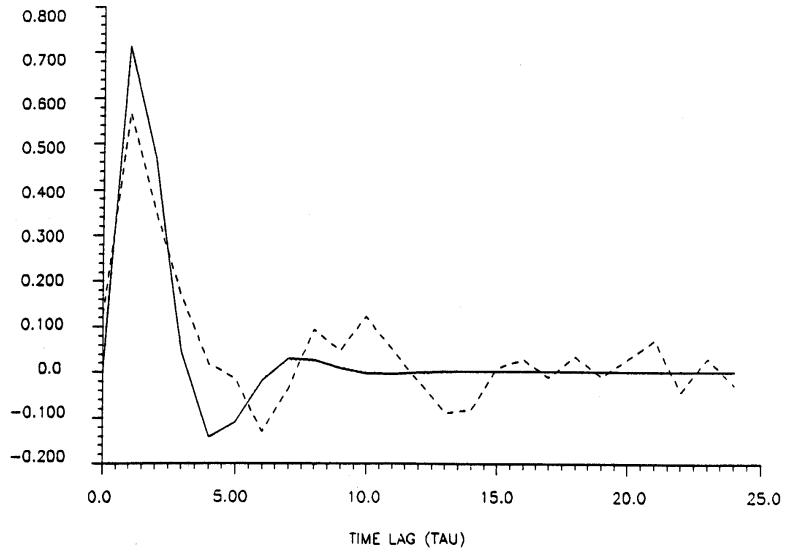


Figure 2.32
1st-order kernel estimates for non-white stimulus obtained via LET (dotted) and CCT (dashed). The exact kernel in solid line is nearly superimposed on the LET estimate [Marmarelis, 1993].

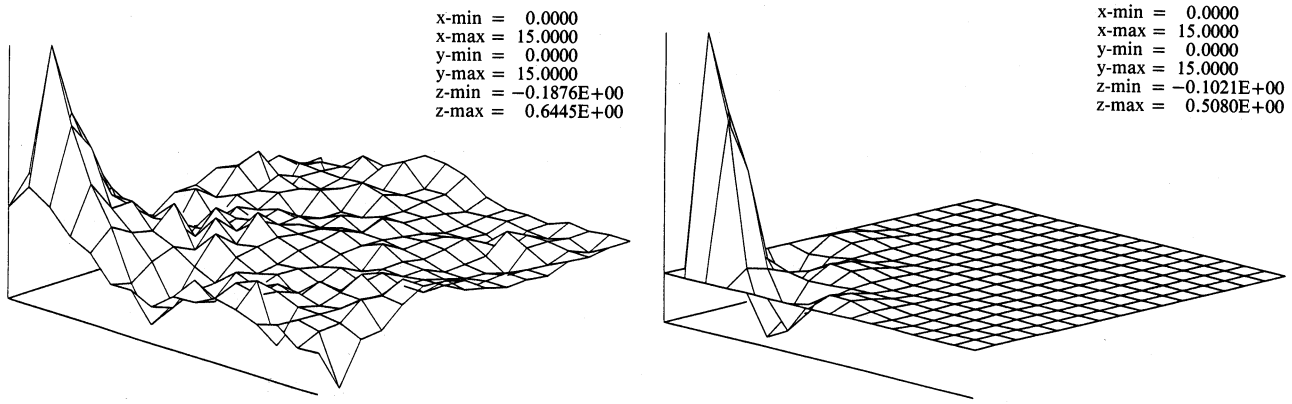


Figure 2.33
(a) 2nd-order kernel estimated for non-white stimulus via LET [Marmarelis, 1993].
(b) 2nd-order kernel estimated for non-white stimulus via CCT [Marmarelis, 1993].

An important advantage of the advocated technique (LET) is its ability to yield accurate kernel estimates even when the input signal deviates from white noise (as long as the model order is correct). This is critically important in experimental studies where a white-noise (or quasi-white) input cannot be easily secured. As an illustrative example, consider the previous 2nd-order system being simulated for a non-white (broad-band) input with two resonant peaks [Marmarelis, 1993]. The 1st-order kernel estimates obtained via LET and CCT are shown in Figure 2.32, where the LET estimate (solid line) is almost identical to the exact kernel (solid line), while the CCT estimate (dashed line) shows the effects

of the non-white stimulus in terms of estimation bias (in addition to the anticipated estimation variance). The 2nd-order kernel estimates obtained via LET and CCT are shown in Figure 2.33 and clearly demonstrate that the LET estimate is far better than the CCT estimate and is not affected by the non-white spectral characteristics of the input. Note, however, that the LET estimates will be affected by the spectral characteristics of a non-white input in the presence of higher order nonlinearities. This is illustrated by simulating the previous 4th-order system with the non-white input. The obtained 1st-order kernel estimates via LET and CCT are shown in Figure 2.34, along with the exact kernel. The LET estimate is affected by showing some additional estimation bias relative to the previous case of white input for the 4th-order system (see Figure 2.30), but still remains much better than its CCT counterpart. The same is true for the 2nd-order kernel estimates shown in Figure 2.35. It is interesting to note that the overall form of the CCT 2nd-order kernel estimates for the non-white input (see Figures 2.33 and 2.35) is not affected much by the presence of higher order terms, even though the estimates are rather poor in both cases.

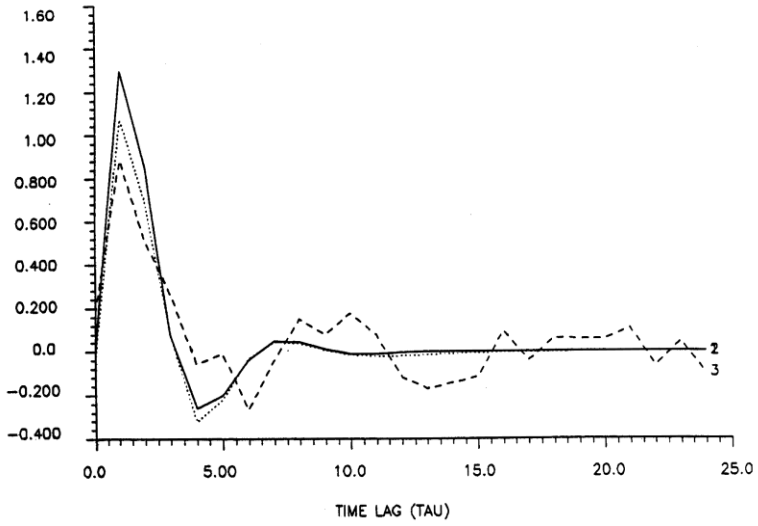


Figure 2.34
1st-order kernel estimates for non-white stimulus and 4th-order system obtained via LET (dotted) and CCT (dashed). The exact kernel is plotted with a solid line [Marmarelis, 1993].

These results demonstrate the fact that the advocated LET approach yields accurate kernel estimates from short experimental data-records, even for non-white (but broadband) inputs, when there are no significant nonlinearities higher than the estimated ones (non-truncated models). However, these kernel estimates may be seriously affected when the experimental input is non-white and significant higher

order nonlinearities exist beyond the estimated model order (truncated model). This is due to the fact that the model residuals are correlated resulting in certain estimation bias owing to lower-order "projections" from the omitted higher order terms (that are distinct from the projections associated with the structure of the Wiener series). Of course, this problem is alleviated when all significant nonlinearities (kernels) are included in the estimated model. This is illustrated below with the accurate estimation of 3rd-order kernels from short data-records (made possible by the ability of the LET approach to reduce the number of required parameters for kernel representation). Although this compactness of kernel representation cannot be guaranteed in every case, the structure of the DLFs (i.e., exponentially weighted polynomials) makes it likely for most physiological systems, since their kernels usually exhibit asymptotically exponential structure.

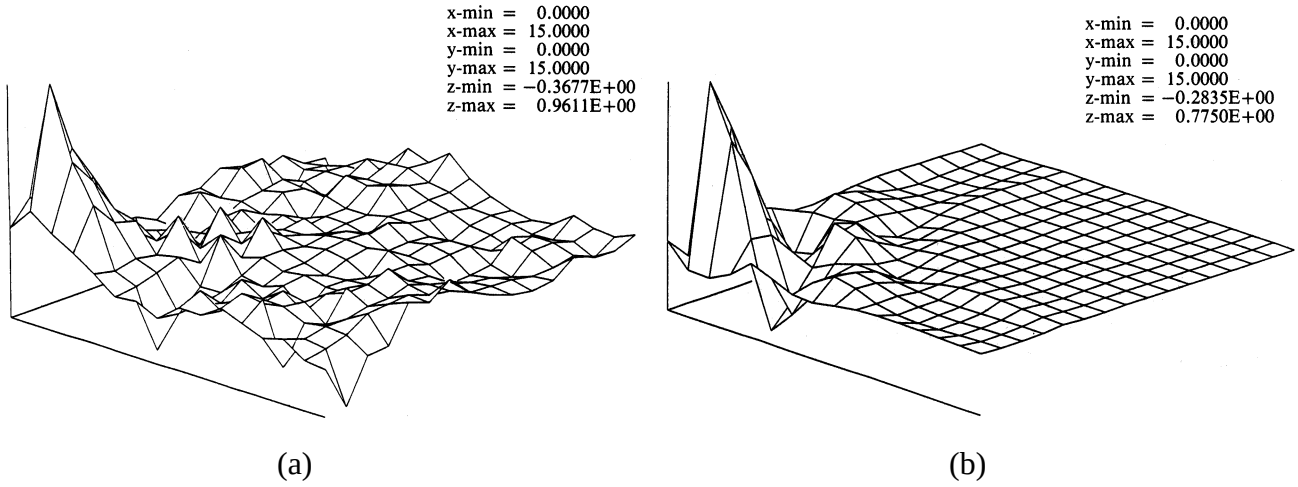


Figure 2.35
(a)The 2nd-order kernel estimate for the non-white stimulus and 4th-order system obtained via LET [Marmarelis, 1993].
(b)The 2nd-order kernel estimate for the non-white stimulus and 4th-order system obtained via CCT [Marmarelis, 1993].

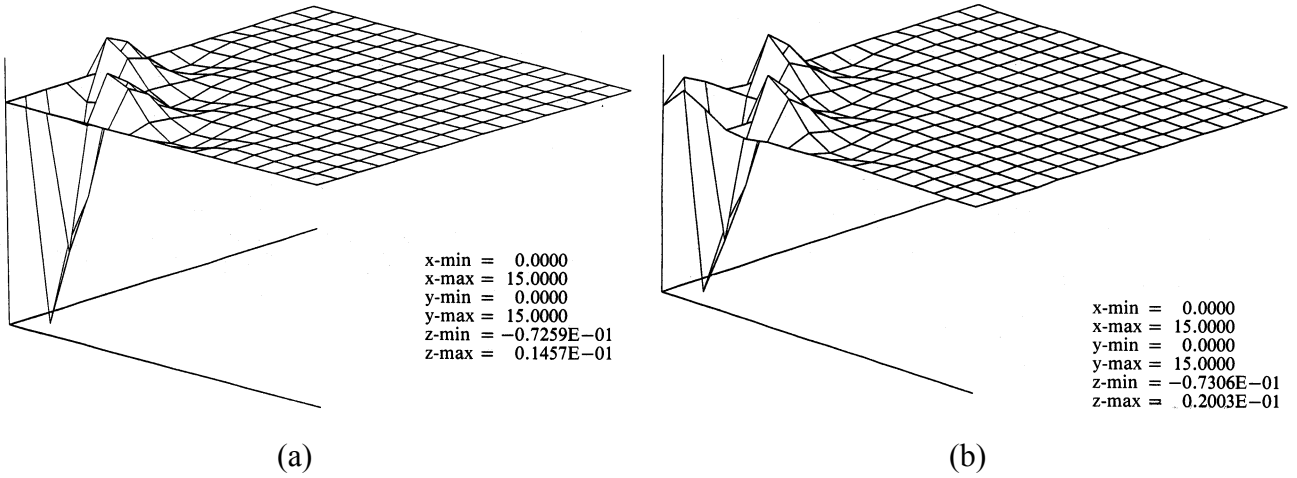


Figure 2.36

(a) 3rd-order kernel estimate (2-D "slice" at $m_3 = 4$) obtained via LET [Marmarelis, 1993].

(b) Exact 3rd-order kernel "slice" at $m_3 = 4$ [Marmarelis, 1993].

Let us now consider the previously simulated system extending to the 3rd-order nonlinearity: $y = z + z^2 + z^3$, receiving a band-limited GWN input of 1024 data points. The resulting 3rd-order kernel estimate via LET is shown in Figure 2.36, as a 3-D "slice" for $m_3 = 4$ (note that visualization of 3rd-order kernels requires taking "3D-slices" for specific values of m_3). Comparison with the exact 3rd-order kernel "slice" at $m_3 = 4$, shown also in Figure 2.36, indicates the efficacy of the LET technique for 3rd-order kernel estimation. Results of similar quality were obtained for all other values of m_3. It has been shown that 3rd-order kernel estimate obtained via the traditional CCT in this case is rather poor [Marmarelis, 1993].

The ability of the advocated LET approach to make the estimation of 3rd-order kernels practical and accurate from short data-records creates a host of exciting possibilities for the effective nonlinear analysis of physiological systems with significant 3rd-order nonlinearities. This can give good approximations of nonlinearities with an inflection point (such as sigmoid-type nonlinearities that have gained increasing prominence in recent years) that are expected to be rather common in physiology because of the requirement for bounded operation for very large positive or negative input values. The efficacy of the advocated kernel estimation technique (LET) is further demonstrated in Chapter 6 with results obtained from actual experimental data in various physiological systems.

2.3.3. High-Order Volterra Modeling with Equivalent Networks

The use of a basis of functions for kernel expansion (discussed in the previous two sections) is equivalent to using a linear filterbank to pre-process the input in order to arrive at a static nonlinear relation between the system output and the multiple outputs of the filterbank, in accordance with Wiener's original suggestion (see Figure 2.9).

As discussed in Section 2.2.2, Wiener further suggested the expansion of the multi-input static nonlinearity into a set of Hermite functions, giving rise to a network of adjustable coefficients of these functions that are characteristic of each system and must be estimated from the input-output data (i.e., a parametric description of the system). This model structure (often referred to as the Wiener-Bose model and shown in Figure 2.9) is an early network configuration that served as an equivalent model for the Wiener class of nonlinear dynamic systems. For reasons discussed in Section 2.2.2, this particular

model form was never adopted by users. However, Wiener's idea has been adapted successfully in the Volterra context, where the multi-input static nonlinear relation can be expressed in multinomial form, as shown in Equation (2.178) for the general Volterra model representation (see Figure 2.16). This modular structure of the general Volterra model can be also cast as an equivalent feedforward network configuration that may offer certain practical/methodological advantages presented below.

Various architectures can be utilized to define equivalent networks for Volterra models of physiological systems. Different types of input pre-processing can be used (i.e., different filter-banks) depending on the dynamic characteristics of the system, and different numbers or types of hidden units (e.g., activation functions) can be used depending on the nonlinear characteristics of the system. The selected architecture affects the performance of the model in terms of parsimony, robustness and prediction accuracy, as elaborated in Chapter 4 where the connectionist modeling approach is discussed further. In this section, we will limit ourselves to discussing the network architectures that follow most directly from the Volterra model forms discussed in Section 2.3.1.

Specifically, we focus on the use of filterbanks for input pre-processing and polynomial-type decompositions of the static nonlinearity (i.e., polynomial activation functions of the hidden units) transforming the multiple outputs of the filterbank into the system output. One fundamental feature of these basic architectures is that there are no recurrent connections in these equivalent networks and, therefore, they attain forms akin to feedforward "artificial neural networks" that have received considerable attention in recent years. Nonetheless, recurrent connections may offer significant advantages in certain cases and are discussed in Chapter 10. Due to the successful track record of the Laguerre expansion technique (presented in Section 2.3.2), the particular feed-forward network architecture that utilizes the discrete-time Laguerre filterbank, termed the "Laguerre-Volterra network", has found many successful applications in recent years and is elaborated in Section 4.3.

The modular representation of the general Volterra model shown in Figure 2.16 is equivalent to a feedforward network architecture that can be used to obtain accurate and robust *high-order* Volterra models using relatively short data-records. One particular implementation that has been used successfully in several applications to physiological system modeling is the aforementioned Laguerre-Volterra Network (discussed in Section 4.3). In this section, we discuss the general features of the feedforward "Volterra-Wiener Network" (VWN) which implements directly the block-structured model of Figure 2.16 with a single hidden layer having polynomial activation functions. The VWN employs a discrete-time filter-bank with trainable parameter(s) for input pre-processing and is equivalent to a

"separable Volterra network", discussed earlier. The applicability of the VWN is premised on the separability of the multi-input static nonlinearity f into a sum of polynomial transformations of linear combinations of the L outputs $\{v_j\}$ of the filterbank as:

$$y(n) = f(v_1,...,v_L) \cong c_0 + \sum_{h=1}^{H}\sum_{q=1}^{Q} c_{h,q}\left\{\sum_{j=1}^{L} w_{h,j} v_j(n)\right\}^q \tag{2.210}$$

where $\{w_{h,i}\}$ are the connection weights from the L outputs of the filterbank to the H hidden units and $\{c_{h,q}\}$ are the coefficients of the polynomial activation functions of the hidden units.

Although this separability cannot be guaranteed in the general case for finite *H*, it has been found empirically that such separability is possible (with satisfactory approximation accuracy) for *all* cases of physiological systems modeling attempted to date. In fact, it has been found that reasonable approximation accuracy is achieved *even for small* H, leading to the notion of "principal dynamic modes" discussed in Section 4.1.1.

It can be shown that *exact* representation of *any* Volterra system can be achieved by letting L and H approach infinity [Marmarelis & Zhao, 1994,1997]. However, this mathematical result is of no practical utility as we must keep L and H to small values by practical necessity.

Estimation of the unknown parameters of the VWN (i.e., the w's and c's) is achieved by iterative schemes based on gradient descent and the chain rule of differentiation for error back-propagation, as discussed in detail in Section 4.2. Here we limit ourselves to discussing the main advantage of the VWN model (i.e., the reduction in the number of unknown parameters) relative to conventional Volterra or MDV models, as well as the major considerations governing their applicability to physiological systems.

In terms of model compactness, the VWN has $\left[H(L+Q)+1\right]$ free parameters (excluding any trainable parameters of the filterbank), while the MDV model of Equation (2.180) based on kernel expansions has $\left[(L+Q)!/(L!Q!)\right]$ free parameters (in addition to any filterbank parameters, such as the DLF parameter α). The conventional discrete Volterra model of Equation (2.42) with memory-bandwidth product M has $\left[(M+Q)!/(M!Q!)\right]$ free parameters. Since typically $M \gg L$, the conventional discrete Volterra model is much less compact than the MDV model and we will focus on comparing the MDV with the VWN. It is evident that the VWN is more compact when:

$$H < (L+Q-1)...(L+1)/Q!$$

Clearly, the potential benefits of using the VWN increase for larger Q, given that H is expected to retain small values in practice (less than five). Since L is typically less than ten, the VWN is generally expected to be more compact than the MDV model (provided the separability property applies to our system). As an example, for $H=3, L=10, Q=3$, we have a compactness ratio of 7 in favor of VWN, which grows to about 67 for $Q=5$. This implies significant savings in data-record length requirements when the VWN is used, especially for high-order models. Note that the savings ratio is much higher relative to conventional discrete Volterra models (about 4,000 for $Q=3$).

Another network model architecture emerges when the separability concept used in VWN is applied to the conventional discrete Volterra model, without a filterbank for pre-processing the input. This leads to the "separable Volterra network" (SVN) that exhibits performance characteristics between VWN and MDV, since its number of free parameters is $[H(M+Q)+1]$. For instance, the VWN is about 8 times more compact than the SVN when $M=100, L=10, Q=3$, regardless of the number of hidden units. The SVN is also useful in demonstrating the equivalence between discrete Volterra models and three-layer perceptrons or feedforward artificial neural networks [Marmarelis & Zhao, 1997] as discussed in Section 4.2.1.

It is evident that a variety of filterbanks can be used in given applications to achieve the greatest model compactness. Naturally, the performance improvement in the modeling task depends on how well the chosen filterbank matches the dynamics of the particular system under study. This has given rise to the concept of the "principal dynamic modes", which represent the optimal (minimal) set of filters in terms of MDV model compactness for a given system, as discussed in Section 4.1.1.

It is also worth noting that different types of activation functions (other than polynomial) can be used in these network architectures, as long as they retain the equivalence with the Volterra class of models. Specifically, any complete set of analytic functions will retain this equivalence (e.g., exponential, trigonometric, polynomial and combinations thereof). Of course the key practical criterion in assessing suitability for a specific system is the resulting model compactness for a given level of prediction accuracy. The latter is determined by the combination of the selected filterbank and activation functions (in form and numbers). A promising approach to near-optimal selection of a network model for physiological systems is presented in Section 4.4.

The model compactness determines the required length of experimental data and the robustness of the obtained parameter estimates (i.e., estimation accuracy) for given signal-to-noise ratio (SNR) -- both

important practical considerations. In general, more free parameters and/or lower SNR in the data imply longer data-record requirements -- a fact that impacts critically the experiment design and our fundamental ability to achieve our scientific goal of understanding the system function under the prevailing experimental/operational conditions. For instance, issues of nonstationarity of the experimental preparation may impose strict limits on the length of the experimental data that can be collected in a stationary context, with obvious implications on model parameter estimation. In addition, lower SNR in the data raises the specter of unintentional overfitting when the model is not properly constrained (i.e., having the minimum necessary number of free parameters). This risk is mitigated by the use of statistical criteria for model order selection, like the one presented in Section 2.3.1 for MDV models. Finally, it must be noted that the model compactness may affect our ability to properly interpret the results in a physiologically meaningful context.

It is conceivable that the introduction of more "hidden layers" in the network architecture (i.e., more layers of units with nonlinear activation functions) may lead to greater overall model compactness by allowing reduction in the number of hidden units in each hidden layer (so that the total number of parameters is reduced). The introduction of additional hidden layers also relaxes the aforementioned requirement of separability, expressed by Equation (2.210). The estimation/training procedure is minimally impacted by the presence of multiple hidden layers. The case of multiple hidden layers will be discussed in Sections 4.2 and 4.4 and may attain critical importance in connection with multi-input and spatio-temporal models (see Chapter 7).

2.4. ANALYSIS OF ESTIMATION ERRORS

In this section, we elaborate on the specific estimation errors associated with the use of each of the presented kernel estimation methods: (1) the cross-correlation technique (Section 2.4.2); (2) the direct inversion methods (Section 2.4.3); and (3) the iterative cost minimization methods (Section 2.4.4). We begin with an overview of the various sources of estimation errors in the following section.

2.4.1. Sources of Estimation Errors

The main sources of estimation errors in nonparametric modeling can be classified into three categories:

(1) *model specification* errors, due to mismatch between the system characteristics and the specified model structure;

(2) *estimation method* errors, due to imperfections of the selected estimation method for the given input and model structure;

(3) *noise/interference* errors, due to the presence of ambient noise (including measurement errors) and systemic interference.

In the advocated modeling approach that employs kernel expansions or equivalent network structures, the model specification errors arise from incorrect selection of the key structural parameters of the model. For instance, in the Laguerre expansion technique (LET), the key structural parameters are: the number L of discrete-time Laguerre functions (DLFs), the order of nonlinearity Q, and the DLF parameter α. In the case of a Volterra-Wiener Network (VWN), the key structural parameters are: the number L of filters in the filterbank, the number H of hidden units, and the order of nonlinearity Q in the polynomial activation functions. Misspecification of these structural parameters will cause modeling errors whose severity will depend on the specific input signals and, of course, on the degree of misspecification. This type of error is very important, as it has broad ramifications both on the accuracy and the interpretation of the obtained models. It can be avoided (or at least minimized) by the use of a statistical search procedure utilizing successive trials, where the parameter values are gradually increased until a certain statistical criterion on the incremental improvement of the output prediction (typically qualified by the sum of the squared residuals) is met. This procedure is described in detail in Section 2.3.1.

The errors associated with the selected estimation method depend also on the specific input signals as they relate to the selected model structure. In the case of LET, the direct inversion method is typically used, since the estimation problem is linear in the unknown parameters. The required matrix inversion (or pseudo-inversion if the matrix is ill-conditioned) is subject to all the subtleties and pitfalls of this well-studied subject. In short, the quality of the result depends on the input data and the structural parameters of the model that determine the Gram matrix to be inverted. Generally, if the input is a broadband random signal and the model structure is adequate for the system at hand, then the results are expected to be very good—provided that the effects of noise/interference are not excessive (see third type of error below). However, if the model structure is inadequate or the input signal is a simple waveform with limited information content (e.g., pulse, impulse, sinusoid or narrowband random), then

the resulting Gram matrix is likely to be ill-conditioned and the results are expected to be poor unless a pseudo-inverse of reduced rank is used.

In the case of VWN, the iterative cost-minimization method is typically used, since the estimation problem is nonlinear in some of the unknown parameters (i.e., the weights of the filterbank outputs). A variety of algorithms are available for this purpose, most of them based on gradient descent. Generally, the key issues with these algorithms are the rate of convergence and the search for the global minimum (avoidance of local minima)—both of which depend on the model structure and the input characteristics. The proper definition of the minimized cost function depends on the prediction error statistics that include the noise/interference contaminating the data. Although a quadratic cost function is commonly used, the log-likelihood function of the model residuals can define the cost function in order to achieve estimates with minimum variance. Many important details concerning the application of these algorithms are discussed in Sections 4.2 and 4.3, or can be found in the extensive literature on this subject that has received a lot of attention in recent years in the application context of artificial neural networks (see, for instance, Haykin 1994 and Hassoun 1995).

The last type of estimation error is due to the effects of ambient noise and/or systemic interference contaminating the input-output data. The extent of this error depends on the statistical and spectral characteristics of the noise/interference as they relate to the respective data characteristics, the selected model structure and the employed estimation method. For instance, if there is considerable noise/interference at high frequencies but little noise-free output power at these high frequencies, then the resulting estimation errors will be significant if the selected model structure allows high-frequency dynamics. On the other hand, these errors will be minimal if the selected model structure constrains the high-frequency dynamics (e.g., a large DLF parameter α and/or small L in the LET method) or if the input-output data are low-pass filtered prior to processing in order to eliminate the high-frequency noise/interference. It should be noted that the iterative cost-minimization methods allow for explicit incorporation of the specific noise/interference statistics (if known or estimable) in the minimized cost function to reduce the estimation variance and achieve robust parameter estimates (see Sections 2.1.5, 4.2 and 4.3).

Naturally, there is a tremendous variety in the possible noise/interference characteristics that can be encountered in practice, and many possible interrelationships with the model/method characteristics to provide general practical guidelines for this type of error. However, it can be generally said that the noise/interference characteristics should be carefully studied and intelligently managed by proper pre-

processing of the data (e.g., filtering) and judicious selection of the model structural parameters for the chosen estimation method and available data. This general strategy is summarized in Chapter 5 and illustrated with actual applications in Chapter 6. It is accurate to say that the proper management of the effects of noise/interference is of paramount importance in physiological system modeling and often determines the success of the whole undertaking. Note that, unlike man-made systems where high-frequency noise is often dominant, physiological systems are often burdened with low-frequency noise/interference that has been given traditionally less attention.

The modeling errors that result from possible random fluctuations in system characteristics are viewed as being part of the systemic noise, since the scope of the advocated methodology is limited to deterministic models. Such stochastic system variations can be studied statistically on the basis of the resulting model residuals.

The effects of noise and interference are intrinsic to the experimental preparation and do not depend on the analysis procedure. However, the specification and estimation errors depend on the selected modeling and estimation methodology. The issue of model specification errors runs through the book as a central issue best addressed in the nonparametric context (minimum prior assumptions about the model structure). A major thrust of this book is to emphasize this point and develop the synergistic aspects with other modeling approaches (parametric, modular and connectionist).

The model specification and estimation errors are discussed in the following section for the various nonparametric methods presented in this book: the cross-correlation technique, the direct-inversion methods, and the iterative cost minimization methods.

2.4.2. Estimation Errors Associated with the Cross-Correlation Technique

In this section, we concentrate on the estimation errors associated with the use of the cross-correlation technique for the estimation of the Wiener or CSRS kernels of a nonlinear system (see Sections 2.2.3 and 2.2.4). Although the introduction of the cross-correlation technique is theoretically connected to GWN inputs, it has been extended (by practical necessity) to the case of quasi-white (CSRS or PRS) inputs used in actual applications, as discussed in Section 2.2.4. Since we are interested in studying the estimation errors during actual applications of the cross-correlation technique, we focus here on the broad class of CSRS quasi-white inputs (that includes the band-limited GWN as a special case) and the discrete-time formulation of the problem, as relevant to practice. Detailed analysis was first provided in [Marmarelis & Marmarelis, 1978].

The estimation of the r th-order CSRS kernel, g_r, requires the computation of the r th-order cross-correlation between sampled input-output data of finite record length N, and its subsequent scaling as:

$$\hat{g}_r(m_1,...,m_r) = \frac{C_r}{N}\sum_{n=1}^{N} y(n)\, x(n-m_1)...x(n-m_r) \tag{2.211}$$

where C_r is the scaling constant, $x(n)$ is the discretized CSRS input, and $y(n)$ is the discretized output, used here instead of the r th-order output residual (as properly defined in Section 2.2.4) to facilitate the analytical derivations, i.e., the analysis applies to the nondiagonal points of the kernel argument space. It is evident that the estimation error is more severe at the diagonal points, because of the presence of low order terms in the CSRS orthogonal functionals; however, the number of diagonal points in a kernel is much smaller than the number of nondiagonal points. Thus the overall kernel estimation error will be determined primarily by the cumulative errors at the nondiagonal points. We note that the study of the errors at diagonal points of the kernels makes use of the higher moments of the input CSRS, thus complicating considerably the analytical derivations. The important issues related to CSRS kernel estimation at the diagonal points are discussed in Section 2.2.4.

The estimation errors for the r th-order CSRS kernel estimate, $\hat{g}_r$, given by Equation (2.211) depend on the step size Δt of the particular quasi-white CSRS input $x(n)$ and on the record length N. Note that Δt may not be equal to the sampling interval T $(T \le \Delta t)$, and that the initial portion of the actual experimental data record (equal to the extent of the computed kernel memory) is not included in the summation interval of Equation (2.211) in order to prevent estimation bias resulting from null input values.

The use of Equation (2.211) for the estimation of the CSRS results in two main types of errors, which are due to the finite record length and the finite input bandwidth. These two limitations of finite record and bandwidth are imposed in any actual application of the method, and consequently the study of their effect on the accuracy of the obtained kernel estimates becomes of paramount importance.

In addition to the aforementioned sources of estimation error, there are errors due to the finite rise-time of input transducers, discretization and digitization errors can be made negligible in practice [Marmarelis & Marmarelis, 1978]. Of particular importance is the sampling rate, which must be higher than the Nyquist frequency of the input-output signals in order to alleviate the aliasing problem $(T \le \Delta t)$ and

must exceed the system bandwidth B_s $\left(T \le \left(2B_s\right)^{-1}\right)$. Also, digitization length of at least 12 bits is needed to provide the numerical accuracy sufficient for most applications.

We concentrate on the two main types of errors caused by the finite record length and bandwidth of the CSRS family of quasi-white test signals [Marmarelis, 1975,1977,1979]. The relatively simple statistical structure of the CSRS facilitates the study of their autocorrelation properties (as discussed in Section 2.2.4), which are essential in the analysis of the kernel estimation errors.

Substituting $y(n)$ in terms of its Volterra expansion in Equation (2.211), we obtain the following expression for the nondiagonal points of the r th-order CSRS kernel estimate:

$$\hat{g}_r\left(m_1,...,m_r\right) = C_r \sum_{i=0}^{\infty} \sum_{\sigma_1} ... \sum_{\sigma_i} k_i\left(\sigma_1,...,\sigma_i\right) \hat{\phi}_{r+i}\left(\sigma_1,...,\sigma_i,m_1,...,m_r\right) \tag{2.212}$$

where the factor T^i has been incorporated in the discretized Volterra kernel k_i, and:

$$\hat{\phi}_{r+i}(\sigma_1,...,\sigma_i,m_1,...,m_r) = \frac{1}{N}\sum_{n=1}^{N} x\left(n-\sigma_1\right)...x(n-\sigma_i)x(n-m_1)...x(n-m_r) \tag{2.213}$$

is the estimate of the $\left(r+i\right)$th autocorrelation function of the CSRS input $x(n)$.

Clearly, the error analysis for the obtained CSRS kernel estimate $\hat{g}_r$ necessitates the study of the statistical properties of the autocorrelation function estimate $\hat{\phi}_{r+i}$, as they affect the outcome of the summation in Equation (2.212). The estimation error in $\hat{g}_r$ consists of two parts: the bias and the variance.

Estimation Bias

The bias in the CSRS kernel estimate is due to the finite bandwidth of the quasi-white input (i.e., the finite step size Δt). The possible deviation of the CSRS input amplitude distribution from Gaussian may also cause a bias (with respect to the Wiener kernels) but it is not actually an error, since it represents the legitimate difference between the Wiener kernels and the CSRS kernels of the system. Therefore, this type of bias will not be included in the error analysis and was studied separately in Section 2.2.4. Note that the difference between Wiener and CSRS kernels vanishes as Δt tends to zero (see, for instance, Equations (2.128) and (2.134)).

The estimation bias due to the finite input bandwidth amounts to an attenuation of high frequencies in the kernel estimate. The extent of this attenuation can be studied in the time domain by considering that the $2r$ th-order autocorrelation function ϕ_{2r} of a CSRS at the nondiagonal points $(m_1,...,m_r)$, which was found to be [Marmarelis, 1977]:

$$\phi_{2r}(\sigma_1,...,\sigma_r,m_1,...,m_r)=\begin{cases} M_2^r\prod_{i=1}^{r}\left(1-\frac{|m_i-\sigma_i|}{\Delta t}\right); \text{for } |m_i-\sigma_i|\le\Delta t \\ 0 \qquad\qquad\qquad\qquad \text{elsewhere.}\end{cases} \tag{2.214}$$

Assuming the r th-order Volterra kernel k_r is analytic in the neighborhood of the nondiagonal point $(m_1,...,m_r)$, we can expand it in a multi-variate Taylor series about that point. With the use of Equation (2.214), we can obtain:

$$E\left[\hat{g}_r(m_1,...,m_r)\right]=k_r(m_1,...,m_r)+\sum_{l=1}^{\infty}\Delta t^{2l}\sum_{j_1,...,j_l=1}^{r}D_l(j_1,...,j_l)\,k_r^{(2l)}(m_1,...,m_r) \tag{2.215}$$

where $k_r^{(2l)}(m_1,...,m_r)$ denotes the $2l$ -th partial derivative of k_r with respect to each of its arguments twice, evaluated at the discrete point $(m_1,...,m_r)$, and $D_l(j_1,...,j_l)$ depends on the multiplicity of the sets of indices $(j_1,...,j_l)$. More specifically, if a certain combination $(j_1,...,j_l)$ consists of I distinct groups of identical indices, and p_i is the population number (multiplicity) of the i th group, then:

$$D_l(j_1,...,j_l)=\prod_{i=1}^{I}\frac{1}{(2p_i)!(p_i+1)(2p_i+1)}. \tag{2.216}$$

In all practical situations, Δt is much smaller than unity, which allows us to obtain a simpler approximate expression for the expected value of the kernel estimate given by:

$$E\left[\hat{g}_r(m_1,...,m_r)\right]\cong k_r(m_1,...,m_r)+\frac{\Delta t^2}{12}\sum_{j=1}^{r}\left.\frac{\partial^2 k_r(\tau_1,...,\tau_r)}{\partial\tau_j^2}\right|_{\tau_j=m_j}. \tag{2.217}$$

Therefore, if higher order Volterra functionals are present in the system (note that the lower order ones will vanish for nondiagonal points), then we can show that, in first approximation, they give rise to terms of the form [Marmarelis, 1979]:

$$\begin{aligned}E\left[\hat{g}_r(m_1,...,m_r)\right]\cong\sum_{l=0}^{\infty}\frac{(r+2l)!(M_2\Delta t)^l}{r!l!2^l}\Delta t^l\sum_{m_{r+1}}...\sum_{m_{r+1}}\Big\{&k_{r+2l}(m_1,...,m_r,m_{r+1},m_{r+1},...,m_{r+l},m_{r+l})\\&+\frac{\Delta t^2}{12}\sum_{j=1}^{r+l}\left.\frac{\partial^2 k_{r+2l}(\tau_1,...,\tau_r,\tau_{r+1},\tau_{r+1},...,\tau_{r+l},\tau_{r+l})}{\partial\tau_j^{\,2}}\right|_{\tau_j=m_j}\Big\}\end{aligned} \tag{2.218}$$

where $(r+2l)$ represents the order of each existing higher order Volterra kernel. We conclude that, for small Δt, the bias of the rth-order kernel estimate at the nondiagonal points can be approximately expressed as:

$$E\left[\hat{g}_r\left(m_1,...,m_r\right)\right]-g_r\left(m_1,...,m_r\right)\cong A_r\left(m_1,...,m_r\right)\Delta t^2 \tag{2.219}$$

The function $A_r\left(m_1,...,m_r\right)$ depends on the second partial derivative of the Volterra kernels of the same or higher order as indicated by Equations (2.217) and (2.218). We must note that the validity of the derived expressions is limited to the analytic regions of the kernel (a point of only theoretical interest).

Estimation Variance

The variance of the CSRS kernel estimate $\hat{g}_r$ is due to the statistical variation of the estimates of the input autocorrelation functions for finite data-records. This variance diminishes as N increases, but it remains significant for values of N used in practice. Thus, if we consider the random deviation γ_i of the ith-order autocorrelation function, then the variance of the rth order kernel estimate at the nondiagonal points, is equal to the second moment of the random quantity:

$$s_r\left(m_1,...,m_r\right)=C_r\sum_{i=0}^{\infty}\int_0^{\mu}...\int_i k_i\left(\sigma_1,...,\sigma_i\right)\gamma_{r+i}\left(\sigma_1,...,\sigma_i,m_1,...,m_r\right)d\sigma_1...d\sigma_i$$

$$=\sum_{i=0}^{\infty}u_{r,i}\left(m_1,...,m_r\right) \tag{2.220}$$

where the scaling factor C_r for nondiagonal points is [Marmarelis, 1977]

$$C_r=\frac{1}{r!\left(M_2\Delta t\right)^r} \tag{2.221}$$

Each of the random functions $u_{r,i}$ can be evaluated by utilizing the basic properties of the random deviation functions γ_{r+i}. Namely, that γ_{r+i} is of first degree (i.e., piecewise linear) with respect to each of its arguments and that its values at the nodal points (i.e., the points where each argument is a multiple of Δt) determine uniquely its values over the whole argument space through multi-linear interpolation [Marmarelis, 1975,1977; Marmarelis & Marmarelis, 1978]. Furthermore, the probability density function of the γ_{r+i} random values at nodal points will tend to the zero-mean Gaussian due to the

Central Unit Theorem and the statistical properties of the CSRS input. If we expand the kernel k_i into a multi-variate Taylor series about the nodal points and evaluate the integral over each elementary segment of the space within which γ_{r+i} remains analytic (and piecewise linear), then the variance of each random quantity $u_{r,i}$ (for small Δt) is approximately [cf. Marmarelis, 1979]:

$$\mathrm{var}\left[u_{r,i}\left(m_1,...,m_r\right)\right] = \frac{P^{\mathrm{i}}}{N\Delta t^r} U_{r,i}\left(m_1,...,m_r\right) \tag{2.222}$$

where $P = M_2\Delta t$ is the power level of the input CSRS, and $U_{r,i}$ is a square-integral quantity dependent on the i th-order Volterra kernel of the system. Combining Equation (2.222) with (2.220), we find that the variance of the r th order CSRS kernel estimate at the nondiagonal points is approximately:

$$\mathrm{var}\left[\hat{g}_r\left(m_1,...,m_r\right)\right] \cong \frac{B_r^2\left(P;m_1,...,m_r\right)}{N\Delta t^r} \tag{2.223}$$

where:

$$B_r^2\left(P;m_1,...,m_r\right) = \sum_{i=0}^{\infty} P^i U_{r,i}\left(m_1,...,m_r\right) \tag{2.224}$$

depends on the Volterra kernels of the system and the input power level (for details, see Marmarelis, 1979).

Optimization of Input Parameters

The combined error, due to the estimation variance and bias, can be minimized by judicious choice of the CSRS input parameters Δt and N. This optimization procedure is based on Equations (2.219) and (2.223). Therefore, it is valid for very small values of Δt --a condition that is always satisfied in actual applications.

We seek to minimize the total mean-square error Q_r of the r th-order CSRS kernel estimate that can be found by summation of A_r^2 and B_r^2 over the effective memory extent of the kernel:

$$Q_r = \alpha_r \Delta t^4 + \frac{\beta_r\left(P\right)}{N\Delta t^r} \tag{2.225}$$

where α_r and β_r are the summations of A_r^2 and B_r^2, respectively, over $\left(m_1,...,m_r\right)$ that cover the entire kernel memory. Note that Q_r decreases monotonically with increasing N but, in practice, limitations are imposed on N by experimental or computational considerations. Therefore, the optimization task

consists of selecting Δt as to minimize Q_r for a given value of N (set at the maximum possible for a given application).

The function Q_r has always a single minimum with respect to Δt, because α_r and β_r are positive constants. The position of this minimum defines the optimum Δt for each order r and is given by

$$\left(\Delta t_{opt}\right)_r = \left[\frac{r}{4N}\frac{\beta_r(P)}{\alpha_r}\right]^{1/(r+4)} \tag{2.226}$$

Consideration must be given to the fact that the optimum Δt depends implicitly through β_r on the power level P, which is proportional to Δt. Therefore, the optimization of Δt for a fixed P necessitates the adjustment of the second moment of the CSRS input signal as Δt changes (since $P = M_2 \cdot \Delta t$) by dividing its amplitude with $\sqrt{\Delta t}$.

Note also that Equation (2.226) gives a different optimum Δt for each order of kernel. The determination of each optimum Δt requires the knowledge of the corresponding constants α_r and β_r, which depend on the (unknown) Volterra kernels of the system. Therefore, these constants must be estimated through properly designed preliminary tests, on the basis of the analysis given above. For instance, if we obtain several r th order kernel estimates for various values of Δt, while keeping the input power level constant and varying the record length as to keep the second term of (2.225) constant, then α_n can be estimated through a least-squares regression procedure. A regression procedure can be also used to estimate β_n by varying the input record length, while keeping Δt and P constant. These procedures have been illustrated with computer simulated examples [Marmarelis, 1977] and elaborated extensively in the first monograph by the Marmarelis brothers [Marmarelis & Marmarelis, 1978].

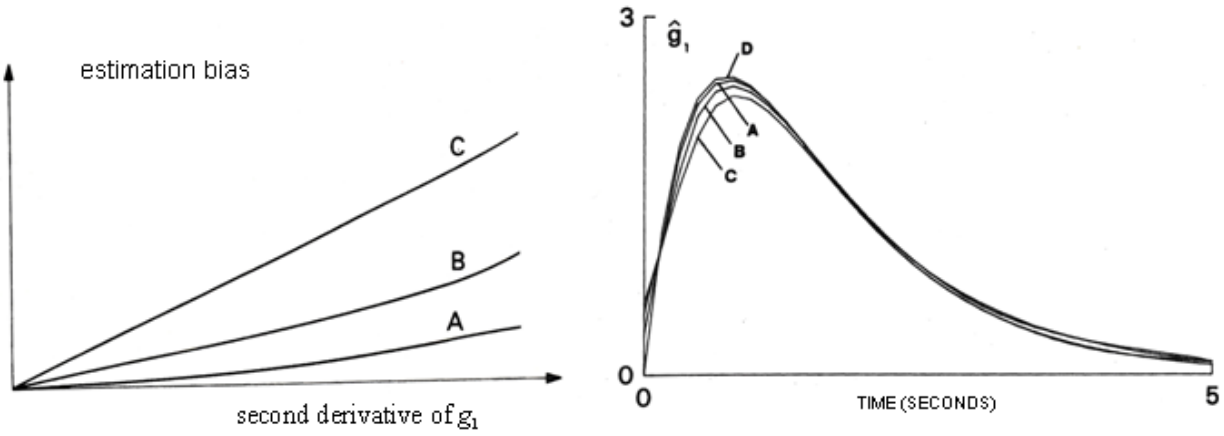


Figure 2.37
Dependence of first-order CSRS kernel estimation bias on CSRS step size (left) for the simulated example defined by the first-order Volterra kernel shown as a curve D in the right panel, along with the CSRS kernel estimates (right) for Δt equal to 0.1 (A), 0.2 (B) and 0.4 (C) [Marmarelis & Marmarelis, 1978].

We illustrate below the findings of the presented analysis through computer simulated examples and demonstrate the proposed optimization method in an actual application. The dependence of the estimation bias upon the step size Δt of the CSRS input (cf. Equation (2.219)) is illustrated in Figure 2.37, where the kernel estimates obtained for three different step sizes are shown. The bias of those estimates at each point is approximately proportional to the second derivative of the kernel and the square of the respective step size, confirming Equation (2.219).

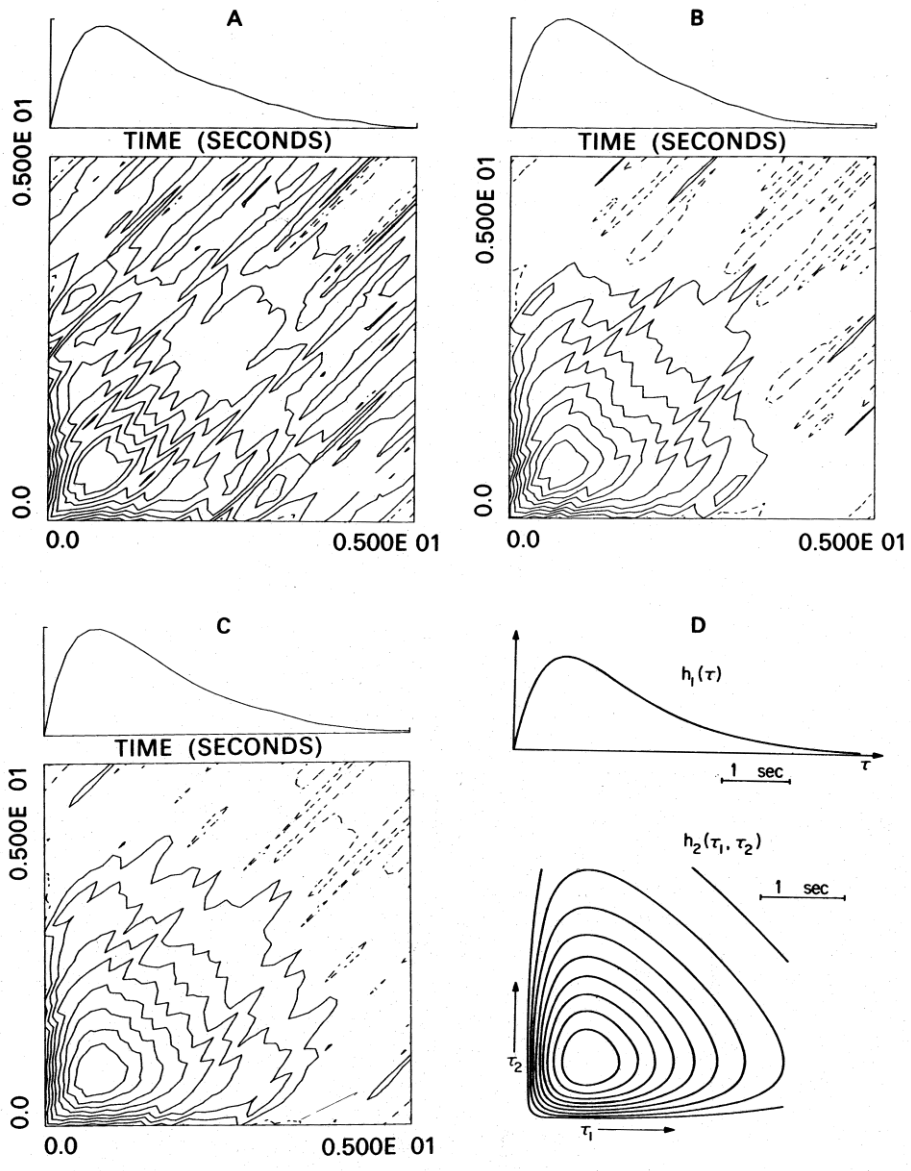


Figure 2.38
First and second order CSRS kernel estimates for various data-record lengths: 500 (A), 1000 (B), and 2000 (C) data points. The exact kernel is shown in (D) [Marmarelis & Marmarelis, 1978].

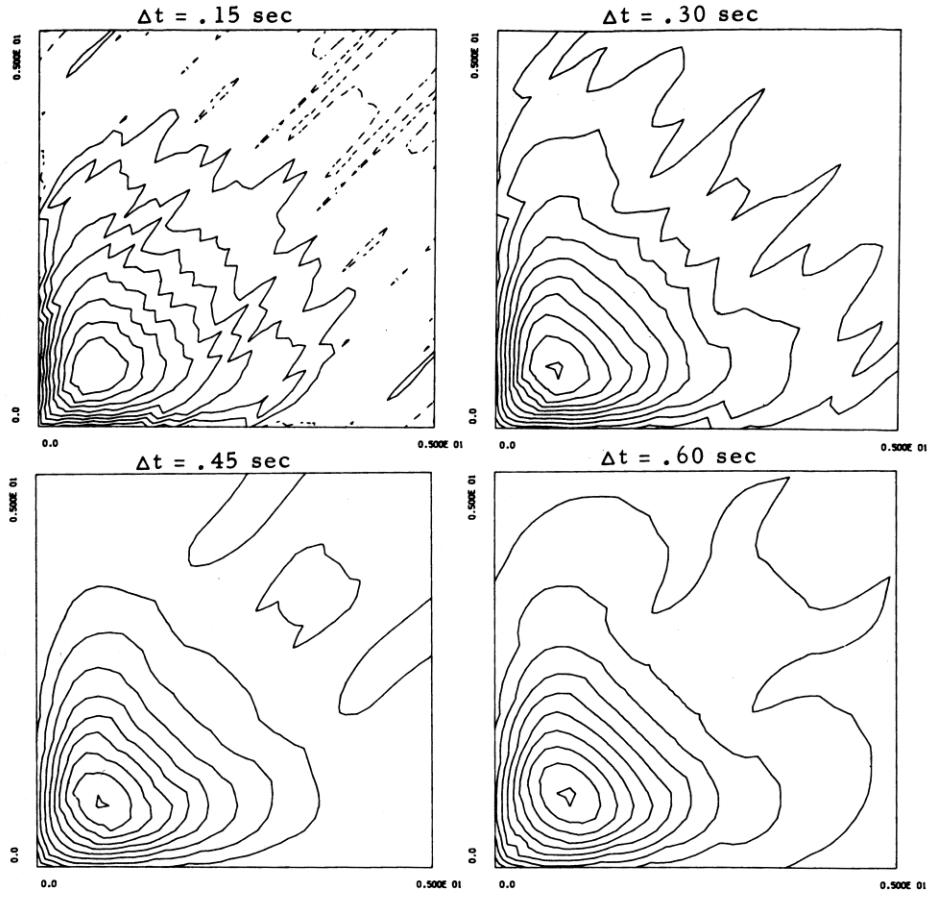


Figure 2.39
Second-order CSRS kernel estimates for various step sizes of the CSRS input. The exact kernel is shown in Figure 2.38 (D) [Marmarelis & Marmarelis, 1978].

The dependence of the estimation variance upon the record length (cf. Equation (2.223)) is illustrated in Figure 2.38, where the first and second order CSRS kernel estimates, obtained for three different record lengths, are shown. The step size Δt of the CSRS input in all three cases is 0.1 sec. The normalized mean-square errors of those estimates (as percentages of the respective summed squared kernels) are given in Table 2.2. These numerical values confirm, within some statistical deviation, the theoretical relation given by Equation (2.223).

The dependence of the total estimation error (bias and variance) upon the step size Δt of the CSRS input (see Equation (2.223)) is illustrated in Figure 2.39, where the second-order CSRS kernel estimates obtained for three different step sizes are shown. The data-record length in all three cases is 500 CSRS steps (i.e., $N = 500$). The statistical error (estimation variance) for multiple kernel estimates of first and second-order is plotted in Figure 2.40 for various step sizes. Note that the same data-record length is used for these estimates and the input power level is kept constant while varying Δt by adjusting the variance of the CSRS input signal. The total mean-square errors of several kernel estimates, obtained for various CSRS step sizes and record lengths, are shown in Figure 2.41, along with the curves representing the theoretical relation described by Equation (2.225).

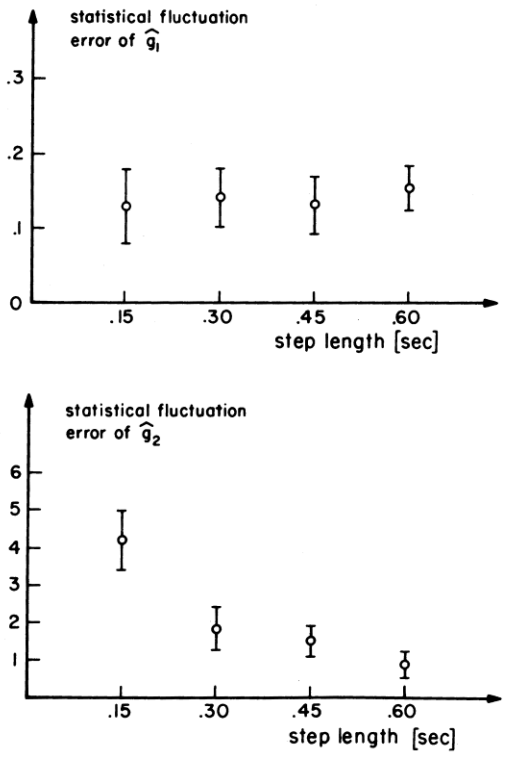


Figure 2.40
The statistical error (estimation variance) of first and second-order CSRS kernel estimates for various CSRS step sizes (mean and standard deviation bars) [Marmarelis & Marmarelis, 1978].

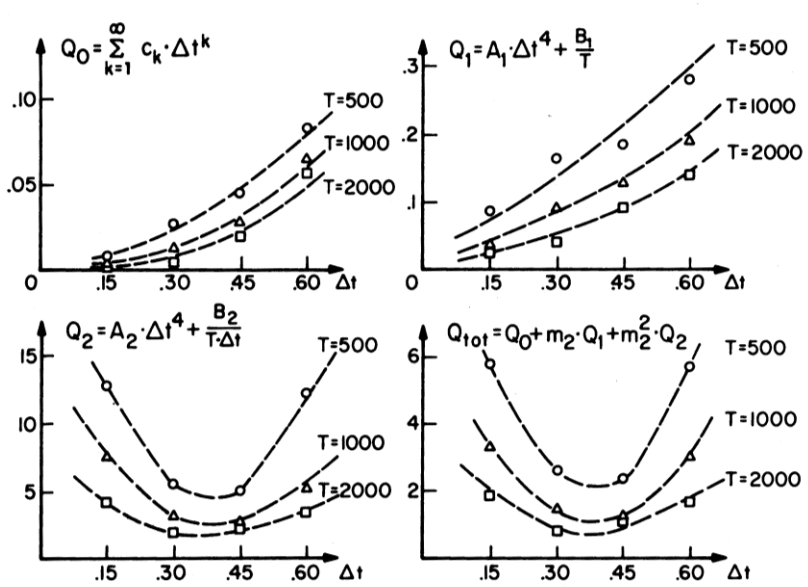


Figure 2.41
The total mean-square error of CSRS kernel estimates for various CSRS step sizes Δt and record lengths T: Q_0, (zeroth-order), Q_1 (first-order), Q_2 (second-order), Q_{tot} (all orders together) [Marmarelis & Marmarelis, 1978].

Noise Effects

The effect of noise in the data is examined in the context of the cross-correlation technique by considering first the most common case of output-additive noise:

$$\tilde{y}(n) = y(n) + \varepsilon(n) \tag{2.227}$$

then the r^{th} –order CSRS kernel estimate obtained via cross-correlation becomes:

$$\tilde{g}_r(m_1,...,m_r) = \hat{g}_r(m_1,...,m_r) + \frac{C_r}{N}\sum_{n=1}^{N}\varepsilon(n)x(n-m_1)...x(n-m_r) \tag{2.228}$$

The last term of Equation (2.228) represents the effect of output-additive noise on the CSRS kernel estimate and will diminish as the record length N increases, because the noise $\varepsilon(n)$ is assumed to have zero mean and finite variance.

In the case of input-additive noise, the effect is more complicated since substitution of $\tilde{x} = x + \varepsilon$ into the cross-correlation formula (2.211) for CSRS kernel estimation gives rise to many multiplicative terms involving the noise term and the input signal. Furthermore, some of the input-additive noise (if not simply measurement errors) may propagate through the system and give rise to noise-generated deviations in the output signal. This further complicates matters and causes multiplicative terms that may not have zero mean (e.g., products of correlated noise values) whose effect will not diminish as fast with increasing N. Fortunately, in practice, input-additive noise is not significant in most cases and care

must be taken to remain minimized by paying proper attention to the way the input is applied to the system (e.g., careful design of D-to-A devices and transducers).

The effect of systemic noise or interference (i.e., random variations of the system characteristics due to interfering factors) is potentially the most detrimental and most difficult to alleviate in practice, since its effect on the output signal is generally dependent on the input and may result in significant estimation errors when the cross-correlation technique is used for kernel estimation. Nonetheless, if the systemic noise/interference has zero mean, then sufficiently long data-records may alleviate its effect considerably. Careful attention must be given to the nature of the systemic noise/interference and experimental care must be taken to minimize its effects.

Erroneous Scaling of Kernel Estimates

In the cross-correlation method, we obtain the r th-order CSRS kernel estimate by scaling the r th-order cross-correlation function estimate with the appropriate factors that depend on the even moments of the CSRS, its step size Δt and the location of the estimated kernel point (i.e., multiplicity of indices for diagonal points).

The problem that is encountered in practice, with regard to the accurate scaling of the CSRS kernel estimates, derives from the fact that the actual input signal, which stimulates the system under study, deviates somewhat from the exact theoretical CSRS waveform that is intended to be delivered as test-input to the system. This deviation is usually caused by the finite response time of the experimental transducers, which convert the string of numbers generated by the digital computer into an analog physical signal that stimulates the system under study. This does not affect the quasi-white autocorrelation properties of the test input signal but it causes some changes in the values of the even moments (which define the scaling factors) from the ones which are theoretically anticipated.

If the measurement of the actual even moments of the applied input signal is not possible, then a final correctional procedure can be used that is based on a least-squares fit between the actual system output and the estimated contributions of the estimated CSRS orthogonal functionals, in order to determine the accurate scaling of the CSRS kernel estimates [Marmarelis & Marmarelis, 1978].

2.4.3. Estimation Errors Associated with Direct Inversion Methods

The direct inversion methods for Volterra kernel estimation employ the Modified Discrete Volterra (MDV) model of Equation (2.180) in its matrix formulation of Equation (2.183). Although pseudo-inversion may be required when the matrix $\mathbf{V}$ is not full-rank or simply ill-conditioned (as discussed in Section 2.3.1), the error analysis will be performed below for the case when the matrix $\mathbf{V}$ is full-rank and, equivalently, the Gram matrix $\mathbf{G} = [\mathbf{V}'\mathbf{V}]$ is non-singular. In this case, we can define the "projection" matrix:

$$\mathbf{H}_r = \mathbf{I} - \mathbf{V}_r \left[\mathbf{V}_r'\mathbf{V}_r\right]^{-1} \mathbf{V}_r' \tag{2.229}$$

for an MDV model or order r that corresponds to structural parameter values of L basis functions for the expansion of the Volterra kernels of highest order Q. The matrix $\mathbf{H}_r$ is idempotent and has rank $(N - P_r)$, where N is the number of output data points and $P_r = (Q+L)!/(Q!L!)$ is the total number of free parameters in the MDV model (i.e., the total number of kernel expansion coefficients that need be estimated).

In Section 2.3.1, we showed that that residual vector for model order r is given by:

$$\begin{aligned} \boldsymbol{\varepsilon}_\mathbf{r} &= \mathbf{H}_\mathbf{r} \boldsymbol{y} \\ &= \mathbf{u}_r + \mathbf{w}_r \end{aligned} \tag{2.230}$$

where $\boldsymbol{u}_r$ is input-dependent (the unexplained part of the system output that vanishes for the correct model order), and $\boldsymbol{w}_r$ is the transformation of the input-independent noise $\boldsymbol{w}_0$ (that is added to the output data and assumed to be white and Gaussian) using the "projection" matrix:

$$\boldsymbol{w}_r = \mathbf{H}_r \boldsymbol{w}_0 \tag{2.231}$$

Note that the matrix $\mathbf{H}_r$ depends on the input data and the MDV model structure of order r. Therefore, even if $\boldsymbol{w}_0$ is white, the residuals for $r \neq 0$ are not generally white. Furthermore, we note that the residuals for an incomplete model order r contain a term:

$$\mathbf{u}_r = \mathbf{H}_r \mathbf{u}_0 \tag{2.232}$$

that represents the model "truncation error" of order r (because it is the part of the noise-free output that is not captured by the model order r), where $\boldsymbol{u}_0$ represents the noise-free output data.

Thus for a MDV model of order r, the "model specification" error is represented by $\boldsymbol{u}_r$ and vanishes for the time system order R, i.e., $\boldsymbol{u}_R = \mathbf{0}$. The size of this error in the estimated expansion coefficients for a truncated model ($r < R$) is given by:

$$\boldsymbol{\theta}_r \triangleq E\left[\hat{\mathbf{c}}_r\right] - \mathbf{c}_r = \mathbf{A}_r \mathbf{A}_R^{+} \mathbf{c}_R - \mathbf{c}_r \tag{2.233}$$

where $\mathbf{c}_r$ and $\mathbf{c}_R$ denote the kernel expansion coefficients for the model orders r and R respectively (R is the true system order), $\mathbf{A}_R^{+}$ denotes the generalized inverse of the $\left[P_R \times N\right]$ matrix $\mathbf{A}_R$, and $\mathbf{A}_r$ is the input-dependent matrix that yields the coefficient vector estimate for model order r:

$$\hat{\mathbf{c}}_r = \mathbf{A}_r \mathbf{y} \tag{2.234}$$

where:

$$\mathbf{A}_r = \left[\mathbf{V}_r' \mathbf{V}_r\right]^{-1} \mathbf{V}_r' \tag{2.235}$$

Note that the coefficient vector estimate is composed of two parts:

$$\hat{\mathbf{c}}_r = \mathbf{A}_r \mathbf{u}_0 + \mathbf{A}_r \mathbf{w}_0 \tag{2.236}$$

where the first term represents the deterministic bias $\boldsymbol{\theta}_r$, of the estimate $\hat{\mathbf{c}}_r$ (due to the model truncation), and the second term represents the random variation in the estimate due to the output-additive noise $\boldsymbol{w}_0$ (statistical error). Since:

$$\begin{aligned} E\left[\hat{\mathbf{c}}_r\right] &= \mathbf{A}_r \mathbf{u}_0 \\ &= \mathbf{A}_r \mathbf{A}_R^{+} \mathbf{c}_R \end{aligned} \tag{2.237}$$

which yields the result of Equation (2.233), quantifying the estimation bias due to the truncation of the model order. It is evident form Equation (2.233) that the model specification error depends on the model structure *and* the input data.

The statistical estimation error $\mathbf{A}_r \mathbf{w}_0$ has zero mean and covariance matrix:

$$\begin{aligned} \mathrm{cov}\left[\hat{\mathbf{c}}_r\right] &\triangleq E\left[\mathbf{A}_r \boldsymbol{w}_0 \boldsymbol{w}_0' \mathbf{A}_r'\right] = \sigma_0^2 \cdot \mathbf{A}_r \mathbf{A}_r' \\ &= \sigma_0^2 \cdot \left[\mathbf{V}_r' \mathbf{V}_r\right]^{-1} \end{aligned} \tag{2.238}$$

when the output-additive noise is white with variance σ_0^2 (note that the SNR is: $\boldsymbol{u}_0' \boldsymbol{u}_0 / \sigma_0^2$). Thus the coefficient estimates are correlated with each other and their covariance depends on the input data and

the structure of the model. The estimation variance will be minimum for Gaussian noise (for given input data). The statistics of the estimation error are multi-variate Gaussian if $\boldsymbol{w}_0$ is Gaussian, or will tend to multi-variate Gaussian even if $\boldsymbol{w}_0$ is not Gaussian (but reasonably concentrated) because of the Central Limit Theorem.

When the output-additive noise $\mathbf{w}_0$ is not white but has some covariance matrix $\mathbf{S}$, then we can perform pre-whitening of the output data in the manner similar to the one described earlier by Equations (2.46)-(2.48). This results in alteration of the fundamental matrix $\mathbf{A}_r$ as:

$$\mathbf{A}_r = \left[\mathbf{V}_r' \mathbf{S}^{-1} \mathbf{V}_r \right]^{-1} \mathbf{V}_r' \mathbf{S}^{-1} \tag{2.239}$$

that determines the estimation bias in Equation (2.233) and the estimation variance in Equation (2.238). Thus, the estimation bias and variance of the kernel expansions are affected by the covariance matrix of the output-additive noise, when the latter is not white.

The case of non-Gaussian output-additive noise is discussed in the following section in connection with iterative cost-minimization methods, since the estimation problem becomes nonlinear in the unknown expansion coefficients when non-quadratic cost functions are used.

2.4.4. Estimation Errors Associated with Iterative Cost-Minimization Methods

These iterative estimation methods are used typically in connection with the network models introduced in Section 2.3.3 and elaborated further in Sections 4.2-4.4, because the estimation problem becomes nonlinear with respect to the unknown network parameters in those cases. The elimination problem also becomes nonlinear when non-quadratic cost functions are used (see Section 2.1.5), even if the model remains linear with respect to the unknown parameters. It is also plausible that these methods can be used for practical advantage in certain cases with quadratic cost function where the model is linear with respect to the unknown parameters (e.g., for very large and/or ill-conditioned Gram matrix) resulting in the "iterative least-squares" method [Goodwin & Sin, 1984; Ljung, 1987; Haykin, 1994].

The model specification error for these methods relates to the selection of the structural parameters (e.g., number L of basis filters, number H of hidden units, number Q of nonlinear order). This error can be minimized by use of a statistical selection criterion such as the one described in Section 2.3.1 in connection with the modified discrete Volterra (MDV) model. However, the size of the resulting specification error depends on the particular iterative method used. The same is true for the parameter estimation error, which depends on the ability of the selected iterative method to converge to the global

minimum of the cost function (for the specified model). Since there is a great variety of such iterative methods (from gradient-based to random search or genetic algorithms), a comprehensive error analysis is not possible within the bounds of this section. The reader can find ample information on this subject in the extensive literature on the training of "artificial neural networks" that are recently in vogue, or on general minimization methods dating all the way back to the days of Newton [Eykhoff, 1974; Haykin, 1994; Hassoun, 1995].

For the modest objectives of this section, we limit our discussion to simple gradient-based methods in connection to the MDV model (see Equations (2.204) and (2.207)). Generally speaking, the efficacy of all these iterative cost-minimization methods relies on the ability of the employed algorithm to find the global minimum (thus avoiding being trapped in the possible local minima) within a reasonable number of iterations (fast convergence). Clearly this task depends on the "morphology of the surface defined by the cost function" in the parameter space (to use a geometric notion). If this surface is convex with a single minimum, the task is trivial. However, this is seldom the case in practice, where the task is further complicated by the fact that the "surface morphology" changes at each iteration. In addition, the presence of noise introduces additional "wrinkles" onto this surface. The presence of multiple local minima (related to noise or not) makes the choice of the initialization point rather critical. Therefore, this task is formidable and remains an open challenge in its general formulation (although considerable progress has been made in many cases with the introduction of clever algorithms for specific classes of problems).

As an example, we consider the case of the MDV model of Equation (2.180) and the gradient-descent method of Equation (2.204) with a quadratic cost function $F(\varepsilon)$ as in Equation (2.207). Then, the "update rule" of the iterative procedure for the expansion coefficient $c_r(j_1,...,j_r)$ is:

$$c_r^{(i+1)}(j_1,...,j_r) = c_r^{(i)}(j_1,...,j_r) + 2\gamma\varepsilon^{(i)}(n)v_{j_1}^{(i)}(n)...v_{j_r}^{(i)}(n) \tag{2.240}$$

where i is the iteration index and $v_j^{(i)}(n)$ denotes the i th-iteration estimate of $v_j(n)$. It is evident from Equation (2.240) that the method utilizes an *estimate* of the gradient of the cost-function surface at each iteration. This estimate can be rather poor and further degraded by the presence of output-additive noise (i.e., a term could be added at the end of Equation (2.240) representing the gradient of the input-dependent residual w_r in Equation (2.231). Therefore, the reader should be disabused of any simplistic

notions of reliable gradient evaluation of the putative cost-function surface or of the certitude of smooth convergence in general.

It is evident that, even in this relatively simple case where the model is linear with respect to the unknown parameters and the cost function is quadratic, the convergence of the gradient-descent algorithm raises a host of potential pitfalls. Nonetheless, experience has shown that gradient-descent methods perform (on the average) at least as well as any existing alternative search methods that have their own set of potential pitfalls.

It should be also noted that the three types of error defined in the introduction of this section (model specification, model estimation and noise related) are intertwined in their effects following the iterative cost-minimization approach. For instance, the use of the proper model order does not guarantee elimination of the model specification error (since entrapment in a local minimum is still possible), or the use of very long data records does not guarantee diminution of the model estimation error (no claim of "consistent estimation" can be made). Likewise, it is not possible to quantify separately the effects of noise based on a measure of signal-to-noise ratio in the data, as these effects depend on the unpredictable distortion of the cost-function surface in the *neighborhood* of the global minimum. For all these reasons, reliable quantitative analysis of estimation errors is not generally possible for the iterative cost-minimization methods.

Historical Note #2: Vito Volterra and Norbert Wiener

The field of mathematical modeling of nonlinear dynamic systems was founded on the pioneering ideas of the Italian mathematician Vito Volterra and of his American counterpart Nobert Wiener, the "father of cybernetics". *Vito Volterra* was born at Ancona, Italy in 1860 and was raised in Florence. His strong interest in mathematics and physics convinced his reluctant middle-class family to let him attend the university, receiving a Doctorate in Physics from the University of Pisa in 1882. He rose quickly in mathematical prominence to become the youngest Professor of the University of Pisa in 1883 at the age of 23. In 1890, he was invited to assume the Chair of Mathematical Physics in the University of Rome, reaching the highest professional status in Italian academe.

In his many contributions to mathematical physics and nonlinear mechanics, Volterra emphasized a method of analysis that he described as "passing from the finite to the infinite". This allowed him to extend the mathematical body of knowledge on vector spaces to the study of functional spaces. In doing so, he created a general theory of functionals (a function of a function) as early as 1883 and applied it to a class of problems that he termed "the inversion of definite integrals". This allowed solution of long-standing problems in nonlinear mechanics (e.g., hereditary elasticity and hysteresis) and other fields of mathematical physics in terms of integral and integro-differential equations. In his seminal monograph "Theory of Functionals and of Integral and Integro-differential Equations", Volterra discusses the conceptual and mathematical extension of the Taylor series expansion to the theory of analytic functionals, resulting in what we now call the Volterra series expansion, which is at the core of our subject matter.

The introduction of the Volterra series as a general explicit representation of a functional (output) in terms of a causally related arbitrary function (input) occupies only one page in Volterra's seminal monograph (p. 21 in the Dover edition of 1930) and the definition of the homogeneous Volterra functionals occupies only two pages (pp. 19-20). Nonetheless, the impact of these seminal ideas on the development of methods for nonlinear system modeling from input-output data has been immense, starting with the pivotal influence on Wiener's ideas on this subject (as discussed below).

The breadth of Volterra's intellect and sociopolitical awareness led him to become a Senator of the Kingdom of Italy in 1905 and to take active part in World War I (1914-18). He was instrumental in bringing Italy to the side of the Allies (Entente) and, although in his mid-50s, he joined the Air Force

and developed its capabilities flying himself with youthful enthusiasm in the Italian skies. He promoted assiduously scientific and technical collaboration with the French and English allies.

At the end of WWI in 1918, Volterra returned to university teaching and commenced research on mathematical biology regarding population dynamics of competitive species in a shared environment. This work was stimulated by extensive interactions with Umberto D'Ancona, a Professor of Biology in the University of Siena, and remained his primary research interest until the end of his life in 1940.

Volterra's later years were shadowed by the oppressive grip of fascism rising over Italy in the 1920s. He was one of the few Senators who had the courage to oppose vocally the rise of fascism at great personal risk--offering to all of us a life model of scientific excellence coupled with moral convictions and liberal thinking. When democracy was completely abolished by the fascists in 1930, he was forced to leave the University of Rome and resign from all positions of influence. His life was saved by the international renown of his scientific stature. Since 1930 and until his death in 1940, he lived mostly in Paris and other European cities, returning only occasionally to his country house in Ariccia where he took refuge from the turmoil of a tormented Europe.

Volterra lived a productive and noble life. He remains an inspiration not only for his pioneering scientific contributions but also for his highly ethical and principled conduct in life. In the words of Griffith Evans from his Preface to Volterra's monograph (Dover edition): "*His career gives us confidence that the Renaissance ideal of a free and widely ranging knowledge will not vanish, however great the presence of specialization*". Let me add to that: "*Volterra's noble life offers a model of the struggle of enlightened intellect against the dark impulses of human nature, that remains the only means for sustaining the advancement of civilization against reactionary forces*".

It is intriguing that with Vito Volterra's passing in 1940, and as the world was getting embroiled in the savagery of World War II, Norbert Wiener was joining the war effort against the Axis on the other side of the Atlantic and was assuming the scientific mantle of advancing Volterra's pivotal ideas (regrettably without explicit acknowledgement) with regard to the problem of nonlinear system modeling from input-output data.

Norbert Wiener was born in 1894 and demonstrated an early aptitude in mathematics (a child prodigy) receiving his Ph.D. at Harvard at the age of 19 and becoming a Professor of mathematics at MIT at the age of 25, where he remained until the end of his life in 1964. Among his many contributions, the ones with high scientific impact have been the solution of the "optimal linear filtering" problem (the Wiener-Hopf equation) and its extension to nonlinear filtering and system identification

(the Wiener functional series) that is germane to our subject. However, the most influential and best known of his intellectual contributions has been the introduction of "*cybernetics*" in 1948 as the science of "communication and control in the animal and the machine".

Wiener authored several books and monographs on the mathematical analysis of data, including "The Fourier Integral and Certain of its Applications" (1933) and "Extrapolation and Interpolation and Smoothing of Stationary Time Series with Engineering Applications" (1949). He also authored a two-volume autobiography: "Ex-Prodigy: My Childhood and Youth" (1953) and "I Am a Mathematician" (1956); as well as a novel, "The Temper" (1959). In the last year of his life, he published "God and Golem" (1964) which received posthumously the National Book Award in 1965.

Wiener was an original thinker and a grand intellectual force throughout his life. His ideas had broad impact that extended beyond science as it shaped the forward perspective of our society with regard to cybernetics and ushered in the "information age". Germane to our subject matter is Wiener's fundamental view "…that the physical functioning of the living individual and the operation of some of the newer communication machines are precisely parallel in their analogous attempts *to control entropy through feedback*" (from "The Human Use of Human Beings: Cybernetics and Society" (1950) p. 26). This view forms the foundation for seeking mathematical or robotic emulators of living systems and asserts the fundamental importance of feedback in physiological system function (i.e., homeostatic mechanisms constraining disorganization, which is analogous to "controlling entropy"). This view is the conceptual continuation of the Hippocratic views on "the unity of organism" and "the recuperative mechanisms against the disease process" with regard to the feasibility of prognosis. Furthermore, Wiener's specific mathematical ideas for modeling nonlinear dynamic systems in a stochastic context, presented in his seminal monograph "Nonlinear Problems in Random Theory" (1958), have provided great impetus to our collective efforts for modeling physiological systems.

Wiener's seminal ideas on systems and cybernetics have shaped the entire field of systems science and have inspired its tremendous growth in the last 50 years. This development was greatly assisted by his interactions with the MIT "statistical communication theory group" under the stewardship of Y.W. Lee (Wiener's former doctoral student). Through these interactions, Wiener's ideas were adapted to the practical context of electrical engineering and were elaborated for the benefit of the broader research community by Lee's graduate students (primarily Amar Bose who first expounded on Wiener's theory of nonlinear systems in a 1956 RLE Technical Report). Lee himself worked with Wiener (initially for his doctoral Thesis) on the use of Laguerre filters for the synthesis of networks/systems (a pivotal Wiener

idea that has proven very useful in recent applications) and the use of high-order cross-correlations for estimation of Wiener kernels (in collaboration with Lee's doctoral student Martin Schetzen). Wiener's ideas on nonlinear system modeling were presented in a series of lectures to Lee's group in 1956 and were transcribed by this group into the seminal monograph "Nonlinear Problems in Random Theory".

Wiener's ideas on nonlinear system identification were first recorded in a brief 1942 paper where the Volterra series expansion was applied to a nonlinear circuit with Gaussian white noise excitation. Following on the orthogonalization ideas of Cameron and Martin (1947), and combining them with his previously published ideas on the "homogeneous chaos" (1938), he developed the orthogonal Wiener series expansion for Gaussian white noise inputs that is the main thrust of the aforementioned seminal monograph and his main contribution to our subject matter (see Section 2.2).

Wiener's ideas were further developed, elaborated and promulgated by Lee's group in the 1950s (Bose, 1956; Brilliant, 1958; George, 1959) and in the 1960s with the Lee-Schetzen cross-correlation technique (1965) having the greatest impact on future developments. However, the overall impetus of the MIT group began to wane in the 1970s, while at the same time the vigor of Wiener's ideas sprouted considerable research activity at Caltech on the subject of visual system modeling, spearheaded by my brother Panos in collaboration with Ken Naka and Gilbert McCann (see Prologue). As these ideas were put to the test of actual physiological system modeling, the weaknesses of the Wiener approach became evident (as well as its strengths) and stimulated an intensive effort for the development of variants that offered more effective methodologies in a practical context. This effort spread nationally and bore fruit in the form of more effective methodologies by the late 1970s, when the Caltech group was dissolved through administrative fiat in one of the least commendable moments of the school administration. The state of the art at that time was imprinted on the monograph "Analysis of Physiological Systems: The White-Noise Approach" authored by the Marmarelis brothers

The torch was passed on to a nationwide community of researchers who undertook to apply this methodology to a variety of physiological systems and further explore its efficacy. It became evident by this time that only some core elements of the original Wiener ideas remained essential. For instance, it is essential to have broadband (preferably stochastic) inputs and kernel expansions for improved efficiency of estimation, but there is no need to orthogonalize the Volterra series or use necessarily Gaussian white noise inputs. This process culminated in the early 1990s, with the development of efficient methodologies that are far more powerful than the original methods in a practical application context (see Section 2.3). On this research front, a leading role was played by the Biomedical

Simulations Resource at the University of Southern California (a research center funded by the National Institutes of Health since 1985) through the efforts of the author and his many collaborators worldwide.

As we pause on the threshold of the new century to reflect on this scientific journey that started with Volterra and Wiener, we can assert with a measure of satisfaction that the roots planted by Hippocrates and Galen grew through numerous efforts all the way to the point where we can confidently avail the present generation of scientists with the powerful methodological tools to realize the "quantum leap" in systems physiology through reliable models that capture the essential complexity of living systems.

CHAPTER 4

MODULAR AND CONNECTIONIST MODELING

INTRODUCTION

In this chapter, we review the other two modeling approaches (modular and connectionist) that supplement the traditional parametric and nonparametric approaches discussed in the previous two chapters. The potential utility of these approaches depends on the characteristics of each specific application (as discussed in Section 1.4). Generally, the connectionist approach offers methodological advantages in a synergistic context, especially in conjunction with the nonparametric modeling approach as discussed in Sections 4.2-4.4. The primary motivation for connectionist (network-structured) modeling is the desire for greater efficiency in extracting the causal relationships between input and output data without the benefit of prior knowledge (as discussed in Sections 4.2 and 4.3). On the other hand, the primary motivation for modular (block-structured) modeling is the desire to compact the estimated nonparametric models and facilitate their physiological interpretation (as discussed in Sections 4.1 and 4.4).

The equivalence among these model forms is established through detailed mathematical analysis in order to allow their synergistic use, since each approach exhibits its own blend of advantages and disadvantages vis-à-vis the specific requirements of a given application.

Some modular forms that have been proven useful in practice and derive from nonparametric models are discussed in Section 4.1. The connectionist modeling approach and its relation to nonparametric (Volterra) models is discussed in Section 4.2, and Section 4.3 presents its most successful implementation to date (the Laguerre-Volterra network). The chapter concludes with a general model form that is advocated as the ultimate tool for open-loop physiological system modeling from input-output data (the VWM model).

4.1. MODULAR FORM OF NONPARAMETRIC MODELS

In this section, we present a general modular form of nonparametric (Volterra) models employing the concept of "principal dynamic modes," as well as specific modular forms (cascades and feedback) derived from nonparametric models, that have been found useful in actual applications as they assist physiological interpretation of the obtained models. We begin with the general modular form that is equivalent to a Volterra model and employs the concept of "principal dynamic modes" in order to facilitate model interpretation.

4.1.1. Principal Dynamic Modes

The general modular form that is equivalent to a Volterra model derives from the block-structured model of Figure 2.16 which results from the modified discrete Volterra model of Equation (2.180). The discrete impulse response functions $\{b_j(m)\}$ of the filter-bank constitute generally a complete basis for the functional space of the system kernels and is selected *a priori* as a general "coordinate system" for kernel representation. However, this is not generally the most efficient representation of the system in terms of parsimony.

The pursuit of parsimony raises the important practical issue of finding the "minimum set" of linear filters $\{p_i(m)\}$ that yield an adequate approximation of the system output and can be used in connection with the modular form of Figure 2.16 as an efficient/parsimonious model of the system. This "minimum set" is termed the "*principal dynamic modes*" (PDMs) of the system and defines a parsimonious modular model in conjunction with its respective static nonlinearity that is equivalent to the discrete Volterra model and shown schematically in Figure 4.1 [Marmarelis, 1997].

It is evident that the reduction in dimensionality of the functional space defined by the filter-bank requires a criterion regarding the adequacy of the PDM model prediction for the given ensemble of input-output data. This criterion must also take into account the possible presence of noise in the data.

The general approach to this question can be based on the matrix formulation of the modified discrete Volterra (MDV) model of Equation (2.180) that is given in Equation (2.183). For the correct model order R, the MDV model is:

$$\mathbf{y} = \mathbf{V}_R \mathbf{c}_R + \mathbf{H}_R \mathbf{w}_0 \tag{4.1}$$

where $\mathbf{c}_R$ is the true vector of Volterra kernel expansion coefficients for the general basis employed in the filter-bank, $\mathbf{V}_R$ is the input-dependent matrix of model order R, $\mathbf{H}_R$ is the "projection" matrix defined in Equation (2.229), and $\mathbf{w}_0$ is the output-additive noise vector.

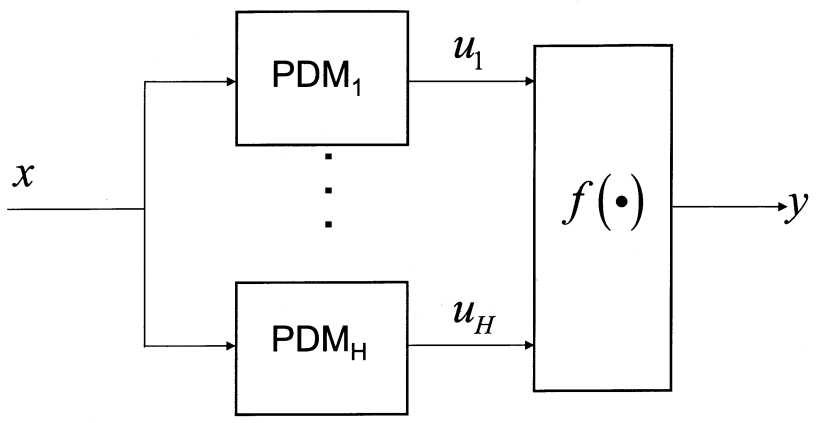


Figure 4.1
Structure of the PDM model composed of a filterbank of H filters (PDMs) that span the dynamics of the system and whose outputs $\{u_j\}$ feed into an H -input static nonlinearity $f(u_1,...,u_H)$ to generate the output of the model.

We seek a parsimonious representation in terms of the size P_R of the coefficient vector $\mathbf{c}_R$, so that the sum of the squared residuals of the model output prediction remains below an acceptable level Ω_0. Let us use the following notation for the "parsimonious" model:

$$\mathbf{y} = \mathbf{\Psi}_s \boldsymbol{\gamma}_s + \boldsymbol{\zeta}_s \tag{4.2}$$

where the input-dependent matrix $\mathbf{\Psi}_s$ is composed of the outputs of the H PDMs in the MDV model formulation ($H < L$), $\boldsymbol{\gamma}_s$ is the vector of the respective expansion coefficients that can reconstruct the Volterra kernel estimates using the PDMs $\{p_i(m)\}$ ($i = 1,...,H$), and $\boldsymbol{\zeta}_s$ is the resulting residual vector whose Euclidean norm cannot exceed Ω_0.

The task is to find a $[P_R \times P_s]$ matrix $\mathbf{\Gamma}$ that transforms $\mathbf{V}_R$ into $\mathbf{\Psi}_s$:

$$\mathbf{V}_R \Gamma = \mathbf{\Psi}_s \tag{4.3}$$

where $P_R = (Q+L)!/(Q!L!)$ and $P_s = (Q+H)!/(Q!H!)$, so that the resulting prediction error remains below the acceptable level Ω_0. Note that the residual vector $\boldsymbol{\zeta}_s$ contains generally a deterministic input-dependent component that results from the reduction in dimensionality of the filter-bank and a stochastic component that is due to the output-additive noise but is different (in general) from $\mathbf{H}_r \mathbf{w}_0$.

The transformation of Equation (4.3) gives rise to a different "projection" matrix for the PDM model:

$$G_s = I - \mathbf{\Psi}_s \left[\mathbf{\Psi}_s' \mathbf{\Psi}_s \right]^{-1} \mathbf{\Psi}_s'$$

$$= I - \mathbf{V}_R \boldsymbol{\Gamma} \left[\boldsymbol{\Gamma}' \mathbf{V}_R' \mathbf{V}_R \boldsymbol{\Gamma} \right]^{-1} \boldsymbol{\Gamma}' \mathbf{V}_R' \tag{4.4}$$

which defines the deterministic error in the output prediction as $\mathbf{G}_s \mathbf{V}_R \mathbf{c}_R$, and the stochastic component of $\boldsymbol{\zeta}_s$ as: $\mathbf{G}_s \mathbf{w}_0$. Therefore, the prediction error for the PDM model is:

$$\boldsymbol{\zeta}_s = \mathbf{G}_s \mathbf{V}_R \mathbf{c}_R + \mathbf{G}_s \mathbf{w}_0 \tag{4.5}$$

which implies that the mean of the Euclidean norm of this prediction error is:

$$E\ \boldsymbol{\zeta}_s' \boldsymbol{\zeta}_s\ = \mathbf{c}_R' \mathbf{V}_R' \mathbf{G}_s' \mathbf{G}_s \mathbf{V}_R \mathbf{c}_R + \sigma_0^2 \cdot \mathrm{Tr}\left\{ \mathbf{G}_s' \mathbf{G}_s \right\} \tag{4.6}$$

where σ_0^2 is the variance of the output-additive noise (assumed white for simplification) and $\mathrm{Tr}\{\cdot\}$ denotes the trace of the subject matrix. Thus, the task becomes one of finding a matrix $\boldsymbol{\Gamma}$ (which determines the matrix $\mathbf{G}_s$ from Equation (4.4)) so that the right-hand side of Equation (4.6) is no more than Ω_0 for a given system (c_R), input data ($\mathbf{V}_R, \mathbf{H}_R$) and output-additive noise variance σ_0^2.

It is evident from Equations (4.4) and (4.6) that the general solution to this problem is rather complicated, however an approximate solution can be obtained with equivalent model networks (see Section 4.2.2), and a practical solution has been proposed for second-order systems [Marmarelis & Orme, 1993; Marmarelis, 1994,1997], which is discussed below.

For a second order MDV model, the output signal can be expressed as a quadratic form:

$$y(n) = \mathbf{v}'(n) \mathbf{C} \mathbf{v}(n) \tag{4.7}$$

where $\mathbf{v}(n)$ is the augmented vector of the filter-bank outputs $\{\mathbf{v}_j(n)\}$ at each discrete time n:

$$\mathbf{v}'(n) = \begin{bmatrix} 1 & v_1(n) & v_2(n) & \cdots & v_L(n) \end{bmatrix} \tag{4.8}$$

and $\mathbf{C}$ is the symmetric coefficient matrix:

$$\mathbf{C} = \begin{bmatrix} c_0 & \frac{1}{2}c_1(1) & \frac{1}{2}c_1(2) & \cdots & \frac{1}{2}c_1(L) \\ \frac{1}{2}c_1(1) & c_2(1,1) & c_2(1,2) & \cdots & c_2(1,L) \\ \frac{1}{2}c_1(2) & c_2(2,1) & c_2(2,2) & \cdots & c_2(2,L) \\ \vdots & \vdots & \vdots & & \vdots \\ \frac{1}{2}c_1(L) & c_2(L,1) & c_2(L,2) & \cdots & c_2(L,L) \end{bmatrix} \tag{4.9}$$

Because of the symmetry of the square matrix $\mathbf{C}$, there exists always an orthonormal (unitary) matrix $\mathbf{R}$ (composed of the eigenvectors of $\mathbf{C}$) such that:

$$\mathbf{C} = \mathbf{R}'\Lambda\mathbf{R} \tag{4.10}$$

where Λ is the diagonal matrix of the eigenvalues (i.e., a diagonal matrix with the $(L+1)$ distinct eigenvalues $\{\lambda_i\}$ of matrix $\mathbf{C}$ as diagonal elements). Therefore:

$$\begin{aligned} y(n) &= \mathbf{v}'(n)\mathbf{R}'\Lambda\mathbf{R}\mathbf{v}(n) \\ &= \mathbf{u}'(n)\Lambda\mathbf{u}(n) \\ &= \sum_{i=0}^{L} \lambda_i u_i^2(n) \end{aligned} \tag{4.11}$$

where:

$$\mathbf{u}(n) = \mathbf{R}\mathbf{v}(n) \tag{4.12}$$

Therefore, each component $u_i(n)$ is the inner product between the ith eigenvector $\boldsymbol{\mu}_i$ of matrix $\mathbf{C}$ (i.e., the ith column of matrix $\mathbf{R}$) and the vector $\mathbf{v}(n)$ which is defined by the outputs of the filter-bank at each discrete time n. The first element, $\mu_{i,0}$, of each eigenvector $\boldsymbol{\mu}_i$ corresponds to the constant which is the first element of the vector $\mathbf{v}(n)$. The other eigenvector elements define the transformed filter-bank outputs:

$$u_i(n) = \mu_{i,0} + \sum_{j=1}^{L} \mu_{i,j} v_j(n) \tag{4.13}$$

Because the eigenvectors have unity Euclidean norm, the eigenvalues $\{\lambda_i\}$ in Equation (4.11) quantify the relative contribution of the components $u_i^2(n)$ to the model output. Thus, inspection of the relative magnitude of the ordered eigenvalues (by absolute value) allows us to determine which components $u_i^2(n)$ in Equation (4.11) make significant contributions to the output $y(n)$.

Naturally, the selection of the "significant" eigenvalues calls for a criterion, such as $|\lambda_i|$ being greater than a set percentage (e.g., 5%) of the maximum absolute eigenvalue $|\lambda_0|$. Having made the selection of H "significant" eigenvalues, the model output signal is approximated as:

$$\begin{aligned} y(n) &= \sum_{i=0}^{H-1} \lambda_i u_i^2(n) \\ &= \sum_{i=0}^{H-1} \lambda_i \left[\mu_{i,0} + \sum_{m=0}^{M-1} p_i(m) x(n-m) \right]^2 \end{aligned}$$

$$= \sum_{i=0}^{H-1} \lambda_i \left\{ \mu_{i,0}^2 + \sum_{m=0}^{M-1} 2\mu_{i,0} p_i(m)\, x(n-m) \right.$$

$$\left. + \sum_{m_1=0}^{M-1} \sum_{m_2=0}^{M-1} p_i(m_1)\, p_i(m_2)\, x(n-m_1)\, x(n-m_2) \right\} \tag{4.14}$$

where:

$$p_i(m) = \sum_{j=1}^{L} \mu_{i,j} b_j(m) \tag{4.15}$$

is the i th "*principal dynamic mode*" (PDM) of the system. It is evident that the selection of H PDMs compacts the MDV model representation by reducing the number of filter-bank outputs and, consequently, the dimensionality of the static nonlinearity that generates the model output. The resulting reduction in the number of free parameters for the second-order MDV model is: $(L-H)(L+H+3)/2$. This reduction in the number of free parameters has multiple computational and methodological benefits that become even more dramatic for higher order models (provided that the PDMs can be determined for higher order models). As a matter of practice, the PDMs thus determined by the second-order model can be used for higher order models as well, on the premise that the PDMs of a system are reflected on all kernels. However, this may not be always true and the issue of determining the PDMs of a higher order model is addressed more properly in the context of equivalent network models, as discussed in Section 4.2.

The resulting Volterra kernels of the PDM model are:

$$k_0 = \sum_{i=0}^{H-1} \lambda_i \mu_{i,0}^2 \tag{4.16}$$

$$k_1(m) = \sum_{i=0}^{H-1} 2\lambda_i \mu_{i,0} p_i(m)$$

$$= \sum_{i=0}^{H-1} \sum_{j=1}^{L} 2\lambda_i \mu_{i,0} \mu_{i,j} b_j(m) \tag{4.17}$$

$$k_2(m_1, m_2) = \sum_{i=0}^{H-1} \lambda_i p_i(m_1)\, p_i(m_2)$$

$$= \sum_{i=0}^{H-1} \sum_{j_1=1}^{L} \sum_{j_2=1}^{L} \lambda_i \mu_{i,j_1} \mu_{i,j_2} b_{j_1}(m_1) b_{j_2}(m_2) \tag{4.18}$$

which indicates the effect of the elimination of the "insignificant" eigenvalues by the PDM analysis on the original expansion coefficients of the Volterra kernels. Specifically, the kernel expansion coefficients after the PDM analysis are:

$$\tilde{c}_1(j) = \sum_{i=0}^{H-1} 2\lambda_i \mu_{i,0} \mu_{i,j} \tag{4.19}$$

$$\tilde{c}_2(j_1, j_2) = \sum_{i=0}^{H-1} \lambda_i \mu_{i,j_1} \mu_{i,j_2} \tag{4.20}$$

Note that another criterion for selecting the PDMs (i.e., the "significant" eigenvalues) can be based on the mean-square value of the model output for a GWN input, after the zero-order Volterra kernel is subtracted (this represents a measure of output signal power for the broadest possible ensemble of input epochs). Implementation of this criterion requires an analytical expression of this mean-square value θ in terms of the expansion coefficients for the orthonormal basis $\{b_j(m)\}$ for any H from 1 to L. This is found to be:

$$\theta(H,P) \equiv E\left[y(n) - k_0 \right]^2 = P\sum_{j=1}^{L} \tilde{c}_1^2(j) + 2P^2 \sum_{j_1=1}^{L}\sum_{j_2=1}^{L} \tilde{c}_2^2(j_1, j_2) + \left\{ P\sum_{j=1}^{L} \tilde{c}_2(j,j) \right\}^2 \tag{4.21}$$

where P is the power level of the GWN input, and $\tilde{c}_1, \tilde{c}_2$ depend on H as indicated by Equations (4.19) and (4.20) respectively. The quantity $\theta(H,P)$ is evaluated for $H = 1, ..., L$, and the ratio $\theta(H,P)/\theta(L,P)$ (that is between 0 and 1) is compared to a critical threshold value that is slightly less than unity (e.g., 0.95), representing the percentage of output signal power captured by the H PDMs for the input power level of interest. The minimum H for which the ratio exceeds the threshold value determines the number of PDMs

The presented PDM analysis can be performed directly in the discrete-time representation of the kernels without employing any expansion on a discrete-time basis [Marmarelis, 1997]. However, the use of a general orthonormal filter-bank (i.e., a complete basis of expansion) usually improves the computational efficiency of this task.

Note that a similar type of analysis can be performed through eigen-decomposition of the second-order kernel alone and selection of its "significant" eigenvalues that determine the PDMs in the form of the respective eigenvectors [Westwick & Kearney, 1994]. The first-order kernel is then included as another (separate) PDM in this approach, unless it can be represented as a linear combination of the PDMs selected from the eigen-decomposition of the second-order kernel.

This approach can be extended to higher order kernels, whereby the eigen-decomposition is replaced by singular-value decomposition of rectangular matrices, properly constructed to represent the contribution of the respective discrete Volterra functional to the output of the model. The column vectors that correspond to the "significant" singular values are the selected PDMs for each order of kernel.

The fundamental tradeoff in the PDM modeling approach regards the compactness versus the accuracy of the model as the number of PDMs changes. This issue can be addressed in a rigorous mathematical framework by considering the following matrix-vector representation of the input-output relation:

$$y(n) = y_0 + \mathbf{x}_1(n)\mathbf{k}_1 + \mathbf{x}_1'(n)\boldsymbol{K}_2\mathbf{x}_1(n) + \mathbf{x}_1'(n)\boldsymbol{K}_3\mathbf{x}_2(n) + ... + \mathbf{x}_1'(n)\boldsymbol{K}_R\mathbf{x}_{R-1}(n) \tag{4.22}$$

where $\mathbf{x}_1'(n) = \begin{bmatrix} x(n) & x(n-1) & \cdots & x(n-M+1) \end{bmatrix}$ is the vector of input epoch values affecting the output at discrete time n, $\mathbf{k}_1' = \begin{bmatrix} k_1(0) & k_1(1) & \cdots & k_1(M-1) \end{bmatrix}$ is the vector of first-order kernel values, $\boldsymbol{K}_2$ is a symmetric matrix defined by the values of the second-order Volterra kernel, and generally the vector $\mathbf{x}_{r-1}(n)$ and the matrix $\boldsymbol{K}_r$ are constructed so that they represent the contribution of the r th-order Volterra functional term. For instance, if $\mathbf{x}_1'(n) = \begin{bmatrix} x(n) & x(n-1) \end{bmatrix}$, then:

$$\mathbf{x}_1'(n)\boldsymbol{K}_3\mathbf{x}_2(n) = \begin{bmatrix} x(n) & x(n-1) \end{bmatrix} \begin{bmatrix} k_3(0,0,0) & 3k_3(0,0,1) & 0 \\ 0 & 3k_3(0,1,1) & k_3(1,1,1) \end{bmatrix} \begin{bmatrix} x^2(n) \\ x(n)x(n-1) \\ x^2(n-1) \end{bmatrix} \tag{4.23}$$

Following this formulation, we may pose the mathematical question of how to reduce the rank of the matrices $\boldsymbol{K}_2$, $\boldsymbol{K}_3$,…, $\boldsymbol{K}_R$ for a certain threshold of significant "singular values" when a canonical input (such as GWN) is used. Then, singular-value decomposition (SVD) of these matrices:

$$\boldsymbol{K}_R = \boldsymbol{U}_R'\boldsymbol{S}_R\boldsymbol{V}_R \tag{4.24}$$

where $\boldsymbol{S}_R$ is the diagonal matrix of singular values, can yield the singular column vectors of matrix $\boldsymbol{V}_R$ corresponding to the "significant" singular values that are greater than the specified threshold (rank reduction). All these "singular vectors" thus selected from all matrices $\boldsymbol{K}_2$ through $\boldsymbol{K}_R$ (and the vector $\mathbf{k}_1$) can form a new rectangular matrix that can be subjected again to rank reduction through SVD in order to arrive at the final PDMs of the system. To account for the relative importance of different nonlinear orders, we can weigh each singular vector by the square-root of the product of the respective

singular value with the input power level. This process represents a rigorous way of determining the PDMs of a system (or the structural parameters H in Volterra-equivalent network models) but requires knowledge of the Volterra kernels of the system--which is impractical. It is presented here only as a general mathematical framework to assist the reader in understanding the meaning and the role of the PDMs in Volterra-type modeling.

A more practical approach to the problem of selecting the PDMs is to use the estimated coefficients from the direct inversion of the MDV model estimation to form a rectangular matrix $\mathbf{C}$ that comprises all the column vectors of all estimated kernel expansions (i.e., the first-order kernel expansion has one column vector, the second-order kernel expansion has L column vectors, the third-order kernel expansion has $L(L+1)/2$ column vectors, etc.). Then application of SVD on the matrix $\mathbf{C}$ yields the PDMs of the system as the singular vectors corresponding to the most significant singular values. To account for the different effect of input power on the various orders of nonlinearity, we may weigh the column vectors of the rth-order kernel by the rth power of the root-mean-square value of the de-meaned input signal.

Illustrative Examples

Two simulated examples are given below to illustrate the PDM analysis using a second-order and a fourth-order Volterra system. Examples of PDM analysis using real data from physiological systems are given in Chapter 6.

In the first example, a second-order system with two PDMs is described by the output equation:

$$y(n) = 1 + v_1(n) + v_2(n) + v_1(n)v_2(n) \tag{4.25}$$

where $y(n)$ is the system output and (v_1, v_2) are the convolutions of the system input $x(n)$--a 1,024-point GWN signal in this simulation--with the two impulse response functions, g_1 and g_2, respectively, shown in Figure 4.2. The first-order and second-order kernels of this system are shown in Figure 4.3 and can be precisely estimated from these data via the Laguerre expansion technique (LET) (see Section 2.3.2). As anticipated by theory, these kernels can be expressed as (note that k_0 =1):

$$k_1(m) = g_1(m) + g_2(m) \tag{4.26}$$

$$k_2(m_1, m_2) = \frac{1}{2}\left[g_1(m_1)g_2(m_2) + g_1(m_2)g_2(m_1)\right] \tag{4.27}$$

Equation (4.25) can be viewed as the static nonlinearity of a system with two PDMs g_1 and g_2 (and their corresponding outputs v_1 and v_2) having a bilinear cross-term: $v_1 \cdot v_2$. This nonlinearity can also be expressed without cross-terms using the "decoupled" PDMs: $(g_1 + g_2)$ and $(g_1 - g_2)$, and their corresponding outputs: $u_1 = (v_1 + v_2)$ and $u_2 = (v_1 - v_2)$, with offsets $\beta_1 = 2$ and $\beta_2 = 0$, respectively (see Equation (4.13)) as:

$$y = \frac{1}{4}(u_1 + 2)^2 - \frac{1}{4}u_2^2 = 1 + u_1 + \frac{1}{4}u_1^2 - \frac{1}{4}u_2^2 \tag{4.28}$$

Note that, if we apply the convention of normalizing the PDMs to unity Euclidean norm, the resulting normalized PDMs are: $p_1 = 0.60(g_1 + g_2)$, $p_2 = 0.92(g_1 - g_2)$, in this case, and have an associated nonlinearity:

$$y = 1 + 1.67u_1 + 0.69u_1^2 - 0.30u_2^2 \tag{4.29}$$

Application of the eigen-decomposition (ED) approach based on LET kernel estimates (outlined above) yields the two PDMs shown in Figure 4.4. Note that the first PDM (solid line) has the same form as the first-order kernel in this case. As indicated previously, these two PDMs are the normalized sum and difference of g_1 and g_2 of Figure 4.2. The estimated nonlinear function (through regression) for these PDM estimates is precisely the one given by Equation (4.29).

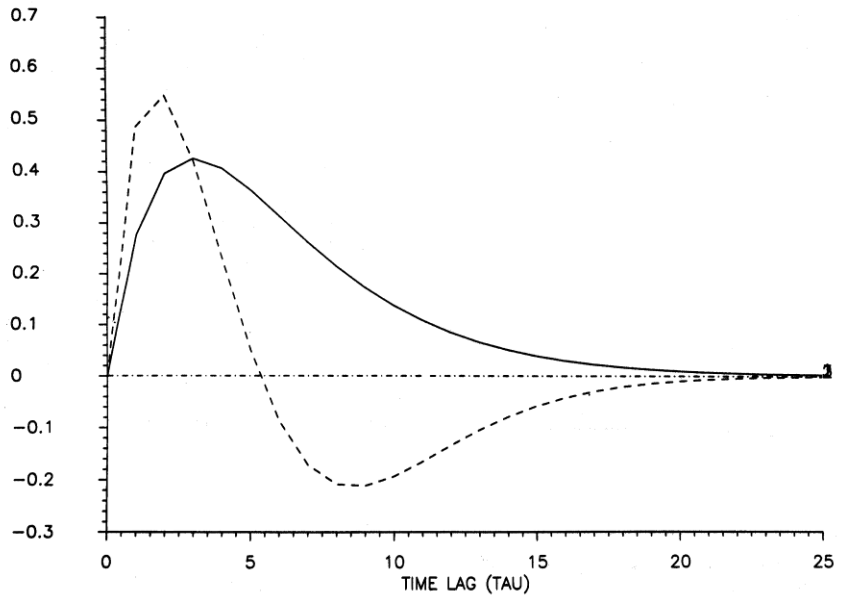
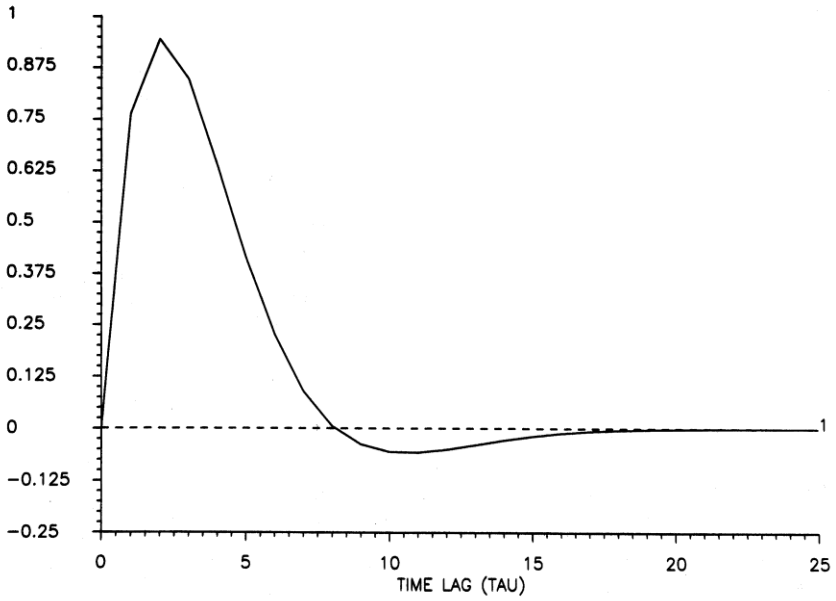
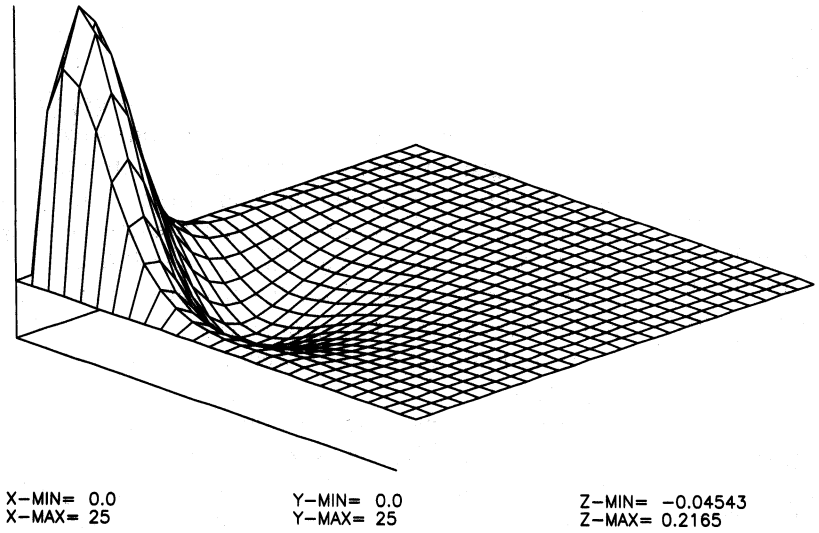


Figure 4.2
Impulse response functions of the two filters $\{g_1, g_2\}$ used in the simulation example [Marmarelis, 1997].



(a)



(b)

Figure 4.3
First-order (left) and second-order (right) kernels of the simulated system described by Equation (4.25) [Marmarelis, 1997].

The presence of contaminating noise may introduce estimation errors (random fluctuations) in the obtained PDMs. The point is illustrated by adding independent GWN to the output signal for an SNR of 6dB and, subsequently, estimating the PDMs using LET-ED that are shown in Figure 4.5. We observe excellent noise resistance of the LET-ED approach, especially for the first PDM (solid line) that corresponds to the highest eigenvalue. The obtained output nonlinearity associated with the normalized PDMs estimated via LET-ED is: $y = 1.02 + 1.66u_1 + 0.70u_1^2 - 0.32u_2^2$, demonstrating the robustness of this method by comparing with the precise output nonlinearity of Equation (4.29). Although these

estimates are somewhat affected by the noise in the data, the estimation accuracy of the resulting nonlinear model in the presence of considerable noise (SNR=6dB) demonstrates the robustness of the proposed approach.

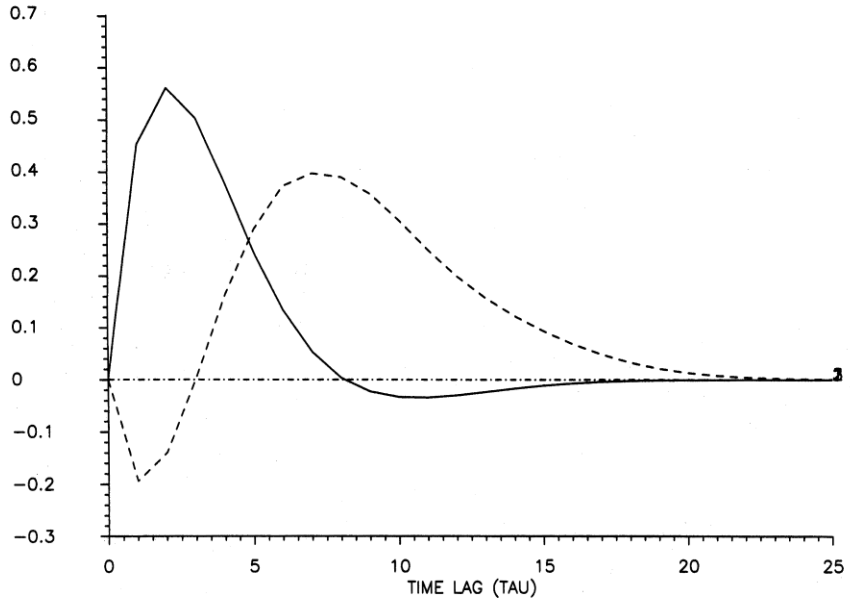


Figure 4.4
Two PDMs obtained via eigen-decomposition (ED) using the first two LET kernel estimates. They correspond to the normalized functions: p_1=0.60(g_1+g_2) and p_2=0.92(g_1-g_2), associated with the nonlinearity of Equation (4.29) [Marmarelis, 1997].

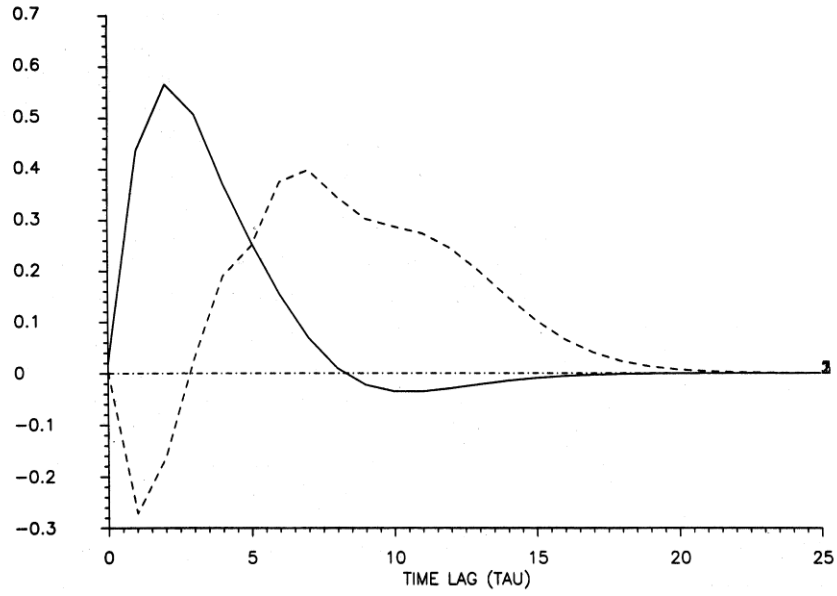


Figure 4.5
Estimated PDMs from noisy data (SNR=6dB) using LET-ED [Marmarelis, 1997].

Next, we examine the efficacy of this approach for high-order systems by considering the fourth-order system described by the output nonlinearity:

$$y = v_1 + v_2 + v_1 v_2 - \frac{1}{3} v_1^3 + \frac{1}{4} v_2^4 \tag{4.30}$$

where (v_1, v_2) are as defined previously. This example serves the purpose of demonstrating the performance of the proposed method in a high-order case where it is not feasible to estimate all of the system kernels in a conventional manner, whereas the proposed method may yield a complete model (i.e., one containing all nonlinear terms present in the system).

Note that the quadratic part of Equation (4.30) is identical to the output nonlinearity of the previous example given by Equation (4.25). The addition of the third-order and fourth-order terms will introduce some bias into the first-order and second-order kernel estimates obtained for the truncated second-order model. This bias is likely to prevent precise estimation of the previous PDMs: $(g_1 + g_2)$ and $(g_1 - g_2)$, via LET-ED based on the first two kernel estimates. Furthermore, more than two PDMs will be required for a model of this system without cross-terms in the output nonlinearity. For instance, the use of three PDMs corresponding to g_1, g_2 and a linear combination $(g_1 + \alpha g_2)$ should be adequate, because g_1 and g_2 give rise to v_1 and v_2, respectively, and the cross-term $(v_1 v_2)$ can be expressed in terms of these three PDMs as: $\left[(v_1 + \alpha v_2)^2 - v_1^2 - \alpha^2 v_2^2 \right] \Big/ 2\alpha$. Thus, three separate polynomial functions corresponding to these three PDMs can be found to fully represent the system output. The LET-ED analysis based on the first two kernel estimates obtained from the simulated data still yields two PDMs corresponding to two significant eigenvalues: $\lambda_1 = 2.38$ and $\lambda_2 = -0.65$ (with the subsequent eigenvalues being: $\lambda_3 = 0.14$, $\lambda_4 = -0.12$, etc.). Note that the sign of the eigenvalues signifies how the respective PDM output contributes to the system output (excitatory or inhibitory).

The obtained PDMs via LET-ED for λ_1 and λ_2 are shown in Figure 4.6, and resemble the PDMs of the previous example, although considerable distortion (estimation bias) is evident due to the aforementioned influence of the high-order nonlinearities. These two PDMs correspond to a bivariate output nonlinearity that only yields an approximation of the system output. This distortion can be removed by increasing the order of estimated nonlinearities. This can be done by means of equivalent network models discussed in Section 4.2. In this simulated example, the use of a fourth-order model yields the three PDMs shown in Figure 4.7 that correspond precisely to g_1, g_2 and a linear combination of g_1 and g_2, as discussed above [Marmarelis, 1997].

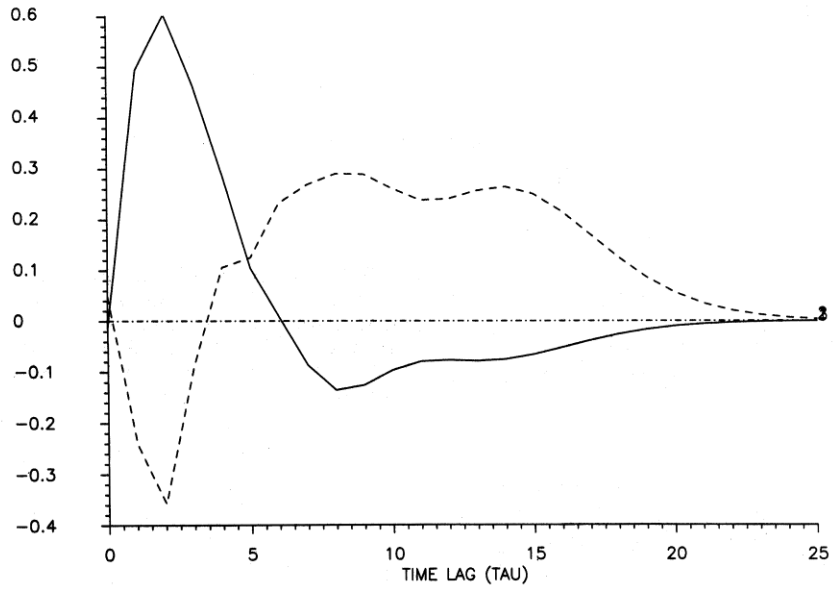


Figure 4.6
The two estimated PDMs for the fourth-order system described by Equation (4.30) using the LET-ED method, corresponding to two significant eigenvalues: $\lambda_1 = 2.38$ and, $\lambda_2 = -0.65$. Considerable distortion relative to the previous two PDMs is evident, due to the influence of the higher order terms (third and fourth order) [Marmarelis, 1997].

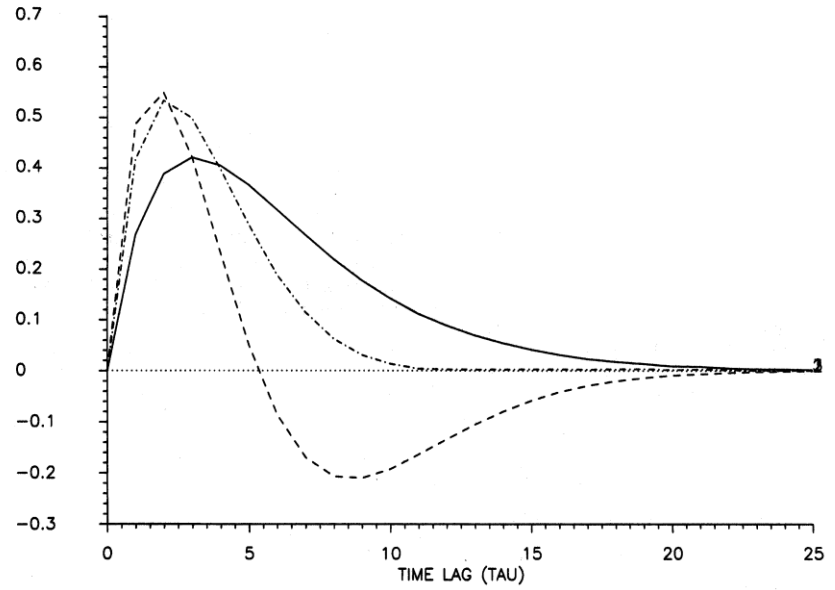


Figure 4.7
The three estimated PDMs for the fourth-order system described by Equation (4.30) using a fourth-order model. The obtained PDMs correspond to g_1, g_2, and a linear combination of g_1 and g_2, as anticipated by theory [Marmarelis, 1997].

These results are extendable to systems of arbitrary order of nonlinearity with multiple PDMs, endowing this approach with unprecedented power of modeling applications of highly nonlinear systems. An illustrative example of infinite-order Volterra system is given for a simulated system described by the output nonlinearity:

$$y = \exp[v_1]\sin\left[(v_1 + v_2)/2\right] \tag{4.31}$$

where v_1 and v_2 are as defined previously. This Volterra system has kernels of all orders with declining magnitudes as the order increases. This becomes evident when the Taylor series expansions of the exponential and trigonometric functions are used in Equation (4.31). Application of the LET-ED method yields only two PDMs (i.e., only two significant eigenvalues: $\lambda_1 = 1.86$ and $\lambda_2 = 1.09$, with the remaining eigenvalues being at least one order of magnitude smaller). The prediction of the fifth-order PDM model (based on two PDMs) is shown in Figure 4.8 along with the exact output of the infinite-order system described by Equation (4.31). Note that the corresponding normalized mean-square error (NMSE) of the model prediction is only 6.8% for the fifth-order PDM model, demonstrating the potential of the PDM modeling approach for high-order Volterra systems.

In closing this section, we must emphasize that the use of the PDM modeling approach is not only motivated by our desire to achieve an accurate model (of possibly high-order Volterra systems) but also by the need to provide meaningful physiological interpretation for the obtained nonlinear models, as demonstrated in Chapter 6.

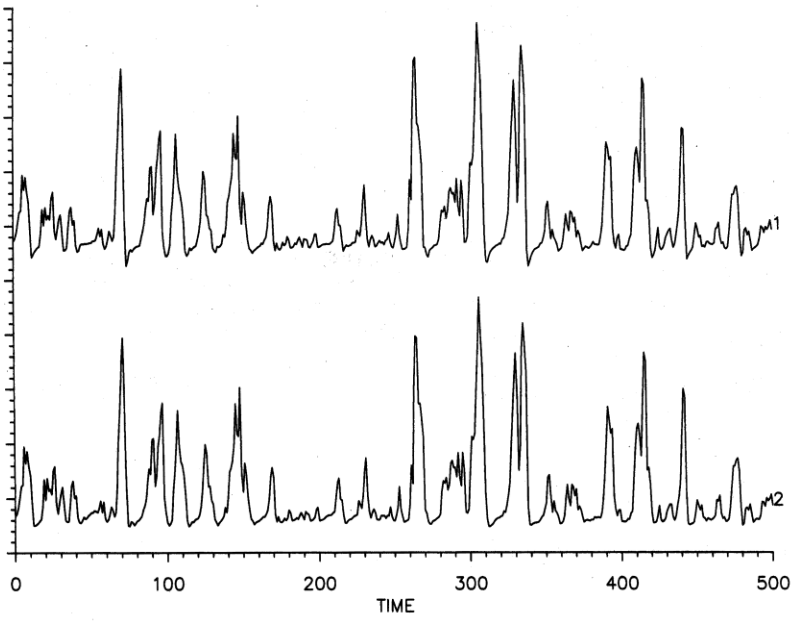


Figure 4.8
Actual output of infinite-order Volterra system (trace 1) and output prediction by a fifth-order PDM model using two estimated PDMs (trace 2). The resulting NMSE of output prediction is only 6.8% [Marmarelis, 1997].

4.1.2. Volterra Models of System Cascades

The cascade of two Volterra systems A and B (shown in Figure 4.9) with Volterra kernels $\{a_0, a_1, a_2, ...\}$ and $\{b_0, b_1, b_2, ...\}$, respectively, has Volterra kernels that can be expressed in terms of the Volterra kernels of the cascade components. The analytical expressions can be derived by the following procedure where the short-hand notation: $k_r \otimes x^r$, is employed to denote the r-tuple convolution of the r-th order kernel k_r with the signal $x(t)$. It is evident that the output of the A-B cascade system can be expressed by the Volterra series expansion:

$$y = b_0 + b_1 \otimes z + b_2 \otimes z^2 + ... \tag{4.32}$$

where $z(t)$ is the output of the first cascade component A that can be expressed in terms of the input $x(t)$ as:

$$z = a_0 + a_1 \otimes x + a_2 \otimes x^2 + ... \tag{4.33}$$

By substitution of z from Equation (4.33) into Equation (4.32), we have the input-output expression for the cascade system:

$$\begin{aligned} y &= b_0 + b_1 \otimes \left[a_0 + a_1 \otimes x + a_2 \otimes x^2 + ... \right] + b_2 \otimes \left[a_0 + a_1 \otimes x + a_2 \otimes x^2 + ... \right]^2 \\ &= \left[b_0 + b_1 \otimes a_0 + b_2 \otimes a_0^2 + ... \right] + \\ &\quad + \left[b_1 \otimes a_0 + b_2 \otimes \left(a_0 \otimes a_1 + a_1 \otimes a_0 \right) + ... \right] \otimes x + \\ &\quad + \left[b_1 \otimes a_2 + b_2 \otimes \left(a_1^2 + a_0 \otimes a_2 + a_2 \otimes a_0 \right) + ... \right] \otimes x^2 + ... \end{aligned} \tag{4.34}$$

which directly determines the Volterra kernels $\{k_0, k_1, k_2, ...\}$ of the cascade system by equating functional terms of a given order. Thus:

$$k_0 = b_0 + a_0 \int_0^\infty b_1(\lambda)\, d\lambda + a_0^2 \int_0^\infty \int b_2(\lambda_1, \lambda_2)\, d\lambda_1 d\lambda_2 + ... + a_0^r \int ... \int_0^\infty b_r(\lambda_1, ..., \lambda_r)\, d\lambda_1 ... d\lambda_r + ... \tag{4.35}$$

$$\begin{aligned} k_1(\tau) &= \int_0^\infty a_1(\tau - \lambda) u(\tau - \lambda) b_1(\lambda)\, d\lambda + 2a_0 \int_0^\infty \int a_1(\tau - \lambda_1) u(\tau - \lambda_1) b_2(\lambda_1, \lambda_2)\, d\lambda_1 d\lambda_2 + \\ &\quad ... + r a_0^{r-1} \int_0^\infty \int a_1(\tau - \lambda_1) u(\tau - \lambda_1) b_r(\lambda_1, ..., \lambda_r)\, d\lambda_1 ... d\lambda_r + ... \end{aligned} \tag{4.36}$$

$$k_2(\tau_1,\tau_2)=\int_0^\infty\int_0^\infty a_2(\tau_1-\lambda_1,\tau_2-\lambda_2)u(\tau_1-\lambda_1)u(\tau_2-\lambda_2)b_1(\lambda_1)\delta(\lambda_1-\lambda_2)d\lambda_1 d\lambda_2$$

$$+\int_0^\infty\int_0^\infty a_1(\tau_1-\lambda_1)a_1(\tau_2-\lambda_2)u(\tau_1-\lambda_1)u(\tau_2-\lambda_2)b_2(\lambda_1,\lambda_2)d\lambda_1 d\lambda_2$$

$$+a_0\int_0^\infty\int_0^\infty\left[a_2(\tau_1-\lambda_1,\tau_2-\lambda_2)+a_2(\tau_1-\lambda_2,\tau_2-\lambda_1)\right]u(\tau_1-\lambda_1)u(\tau_2-\lambda_2)b_2(\lambda_1,\lambda_2)d\lambda_1 d\lambda_2$$

$$+... \tag{4.37}$$

where $u(\tau-\lambda)$ denotes the step function (1 for $\tau\geq\lambda$, 0 otherwise). These expressions can also be given in terms of the multi-dimensional Fourier (or Laplace) transforms of these causal Volterra kernels (denoted with capital letters):

$$k_0=b_0+a_0B_1(0)+a_0^2B_2(0,0)+...+a_0^rB_r(0,...,0)+... \tag{4.38}$$

$$K_1(\omega)=A_1(\omega)B_1(\omega)+2a_0A_1(\omega)B_2(\omega,0)+...+ra_0^{r-1}A_1(\omega)B_r(\omega,0,...,0)+... \tag{4.39}$$

$$K_2(\omega_1,\omega_2)=A_1(\omega_1)A_1(\omega_2)B_2(\omega_1,\omega_2)+A_2(\omega_1,\omega_2)B_1(\omega_1+\omega_2)+2a_0A_2(\omega_1,\omega_2)B_2(\omega_1,\omega_2)+... \tag{4.40}$$

It is evident that these expressions are simplified considerably when $a_0=0$ (i.e., when the first subsystem has no output basal value), especially the expressions for higher order kernels. In this case, the third-order Volterra kernel of the cascade is given in the frequency domain by:

$$K_3(\omega_1,\omega_2,\omega_3)=A_1(\omega_1)A_1(\omega_2)A_1(\omega_3)B_3(\omega_1,\omega_2,\omega_3)$$

$$+\frac{2}{3}\left[A_1(\omega_1)A_2(\omega_2,\omega_3)B_2(\omega_1,\omega_2+\omega_3)+A_1(\omega_2)A_2(\omega_3,\omega_1)B_2(\omega_2,\omega_3+\omega_1)\right.$$

$$\left.+A_1(\omega_3)A_2(\omega_1,\omega_2)B_2(\omega_3,\omega_1+\omega_2)\right]+A_3(\omega_1,\omega_2,\omega_3)B_1(\omega_1+\omega_2+\omega_3) \tag{4.41}$$

The frequency-domain expressions of the Volterra kernels indicate how the frequency-response characteristics of the two cascade components combine at multiple frequencies in a nonlinear context. For instance, the interaction between input power at two frequencies ω_1 and ω_2 is reflected on the cascade output in a manner determined by the first-order response characteristics of A at these two frequencies combined in product with the second-order response characteristics of B at the at the bi-frequency point (ω_1,ω_2) (i.e., the term $A_1(\omega_1)A(\omega_2)B_2(\omega_1,\omega_2)$) etc.

For cascade systems with more than two components, the associative property can be applied, whereby the kernels of the first two components are evaluated first and then the result is combined with

the third component and so on. The reverse route can be followed in decomposing a cascade into its constituent components.

For finite-order cascade components, the order of the cascade is the product of the orders of the cascade components. For instance, in the A-B cascade, if Q_A and Q_B are the finite orders of A and B respectively, then the order of the overall cascade is $Q_A \cdot Q_B$, as can be easily derived from Equations (4.32) and (4.33).

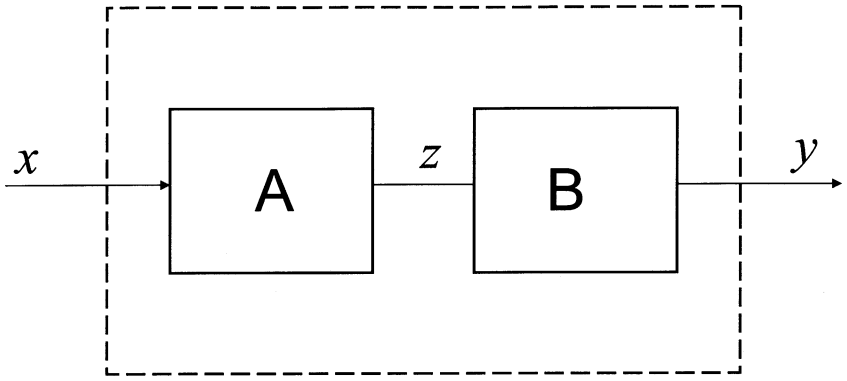


Figure 4.9
The cascade configuration of two Volterra Systems A and B.

The L-N-M, L-N and N-M Cascades

The most widely studied cascade systems to date are the L-N-M, L-N and N-M cascades, where L and M represent linear filters and N is a static nonlinearity. These three types of cascades have been used extensively to model physiological systems in a manner that lends itself to interpretation and control (see Chapter 6). Note that the L-N-M cascade (also called "the *sandwich model*") was initially used for the study of the visual system in the context of "describing functions" [Spekreije, 1969] and several interesting results were obtained outside the context of Volterra-Wiener modeling. In the Volterra-Wiener context, pioneering was the work of Korenberg (1973b).

The expressions for the Volterra kernels of the L-N-M cascade can be easily adapted to the other two types of cascades by setting one filter to unity transfer function (all-pass). Thus, we will start by deriving the kernel expressions for the L-N-M cascade first, and then the expressions for the other two cascades L-N and N-M can immediately follow.

The L-N-M cascade is composed of two linear filters L and M with impulse response functions $g(\tau)$ and $h(\tau)$ respectively, separated by a static nonlinearity N defined by the function $f(\cdot)$ that can be represented as a polynomial or a power series. The latter can be viewed as the Taylor series expansion for any analytic nonlinearity. If the nonlinearity is not analytic, a polynomial approximation of arbitrary accuracy can be obtained over the domain of values defined by the output of the first filter L using any polynomial basis in the context of function expansions detailed in Appendix I. The constitutive equations for the L-N-M cascade shown in Figure 4.10 are:

$$v(t) = \int_0^\infty g(\tau) x(t-\tau)\, d\tau \tag{4.42}$$

$$z(t) = f\left[v(t)\right] = \sum_{r=0}^{Q} \alpha_r v^r(t) \tag{4.43}$$

$$y(t) = \int_0^\infty h(\lambda) z(t-\lambda)\, d\lambda \tag{4.44}$$

where Q may tend to infinity for a general analytic nonlinearity. Combining these three equations to eliminate $v(t)$ and $z(t)$, we can obtain the input-output relation in the form of a Volterra model of order Q as follows. Substitution of $z(t)$ from Equation (4.43) into Equation (4.44) yields:

$$y(t) = \sum_{r=0}^{Q} \alpha_r \int_0^\infty h(\lambda) v^r(t-\lambda)\, d\lambda \tag{4.45}$$

and substitution of $v(t)$ from Equation (4.42) into Equation (4.45) yields the input-output relation:

$$y(t) = \sum_{r=0}^{Q} \alpha_r \int_0^{\min(\tau_1,\ldots,\tau_r)} h(\lambda)\, d\lambda \int_0^\infty \cdots \int_0^\infty g(\tau_1-\lambda)\ldots g(\tau_r-\lambda)\, x(t-\tau_1)\ldots x(t-\tau_r)\, d\tau_1 \ldots d\tau_r \tag{4.46}$$

Therefore, the r th-order Volterra kernel of the L-N-M cascade is:

$$k_r^{\mathrm{LNM}}(\tau_1,\ldots,\tau_r) = \alpha_r \int_0^{\min(\tau_1,\ldots,\tau_r)} h(\lambda)\, g(\tau_1-\lambda)\ldots g(\tau_r-\lambda)\, u(\tau_1-\lambda)\ldots u(\tau_r-\lambda)\, d\lambda \tag{4.47}$$

This expression yields the Volterra kernels of the L-N and N-M cascades by letting $h(\lambda) = \delta(\lambda)$ in the former case and $g(\lambda) = \delta(\lambda)$ in the latter case. Thus, the r th-order Volterra kernel for the L-N cascade (sometimes called the "Wiener model"--somewhat confusingly, since Wiener's modeling concept is far broader than this extremely simple model) has the r th-order Volterra kernel:

$$k_r^{\mathrm{LN}}(\tau_1,\ldots,\tau_r) = \alpha_r g(\tau_1)\ldots g(\tau_r)\, u(\tau_1)\ldots u(\tau_r) \tag{4.48}$$

and the N-M cascade (also called the "Hammerstein model") has the r th-order Volterra kernel:

$$k_r^{\mathrm{NM}}(\tau_1,...,\tau_r) = \alpha_r \left\{ \frac{1}{r} \sum_{(j_1,...,j_r)} h(\tau_{j_1}) \delta(\tau_{j_2} - \tau_{j_1}) ... \delta(\tau_{j_r} - \tau_{j_1}) \right\} u(\tau_1) ... u(\tau_r) \qquad (4.49)$$

where the summation over $(j_1,...,j_r)$ takes place for $j_1 = 1,...,r$ and $(j_2,...,j_r)$ being all other $(r-1)$ indices except j_1. This rotational summation is necessary to make the Volterra kernel invariant to any permutation of its arguments (i.e., symmetric by definition).

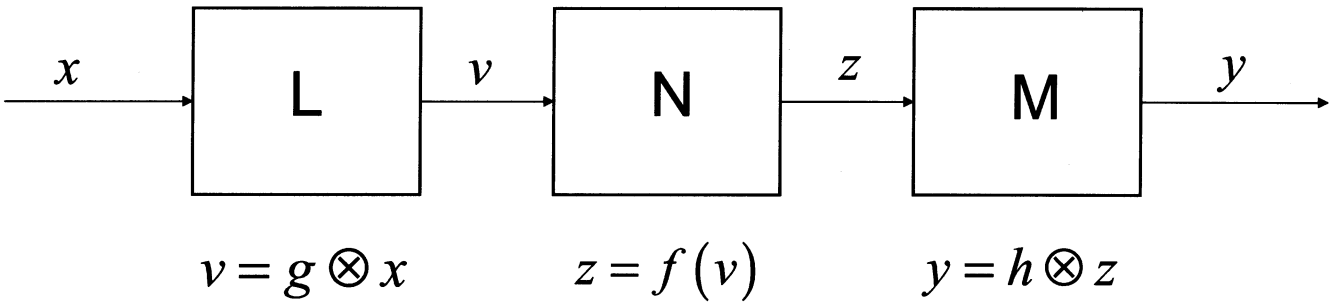


Figure 4.10
The L-N-M configuration composed of two linear filters L and M separated by a static nonlinearity N (see text).

The relative simplicity of these kernel expressions has made the use of these cascade models rather popular. Some illustrative examples were given in Sections 1.4 and 2.1. Additional applications of these cascade models are given in Chapter 6 and more can be found in the literature [Hunter & Korenberg, 1986; Korenberg & Hunter, 1986; Naka et al. 1988; Sakai et al. 1985,1987; Sakuranaga & Naka, 1985].

It is evident from the Volterra kernel expressions (4.47), (4.48), (4.49), that the different cascade structures can be distinguished by simple comparisons of the first and second (or higher) order Volterra kernels. For instance, any "slice" of the second-order kernel at a fixed τ_2 value in the L-N model is proportional to the first-order kernel--an observation used in the fly photoreceptor model discussed in Sections 1.4 and 2.1. The same is true for slices of higher order kernels of the L-N model. Furthermore, the first-order Volterra kernel is proportional to the impulse response function $g(\tau)$ of the linear filter L.

Likewise, the N-M model is distinguished by the fact that the second-order kernel is zero everywhere except at the main diagonal ($\tau_1 = \tau_2$), where the second-order kernel values are proportional to the first-order kernel, which is in turn proportional to the impulse response function $h(\tau)$ of the linear

filter M. Similar facts hold for the higher order kernels of the N-M cascade (i.e., zero everywhere except at the main diagonal, $\tau_1 = \tau_2 = ... = \tau_r$, where the kernel values are proportional to the first-order kernel). Note that kernels of order higher than second have rarely been estimated in practice and, therefore, these useful observations have been used so far primarily in comparisons between first and second order kernels.

For the L-N-M cascade, the relation between the first-order and the second-order Volterra kernels is a bit more complicated in the time domain but becomes simplified in the frequency domain where:

$$K_1^{\mathrm{LNM}}(\omega) = \alpha_1 G(\omega) H(\omega) \tag{4.50}$$

$$K_2^{\mathrm{LNM}}(\omega_1, \omega_2) = \alpha_2 G(\omega_1) G(\omega_2) H(\omega_1 + \omega_2) \tag{4.51}$$

and consequently:

$$K_2^{\mathrm{LNM}}(\omega, 0) = \frac{\alpha_2}{\alpha_1} G(0) \cdot K_1^{\mathrm{LNM}}(\omega) \tag{4.52}$$

This relation can be used to distinguish the L-N-M structure, provided that $G(0) \neq 0$. The frequency-domain expression for the rth-order Volterra kernel of the L-N-M cascade is:

$$K_r(\omega_1, ..., \omega_r) = \alpha_r G(\omega_1) ... G(\omega_r) H(\omega_1 + ... + \omega_r) \tag{4.53}$$

A time-domain relation can also be found for the L-N-M model, when we observe that the values of the rth-order kernel along any axis (i.e., when any τ_i is zero) become proportional to the values of the $(r+1)$th-order kernel along any two axes, due to the causality of $g(\tau)$ [Chen et al. 1985]:

$$\begin{aligned} k_{r+1}^{\mathrm{LNM}}(\tau_1, ..., \tau_{r-1}, 0, 0) &= \alpha_{r+1} h(0) g^2(0) \cdot g(\tau_1) ... g(\tau_{r-1}) \\ &= \frac{\alpha_{r+1}}{\alpha_r} g(0) \cdot k_r^{\mathrm{LNM}}(\tau_1, ..., \tau_{r-1}, 0) \end{aligned} \tag{4.54}$$

This relation, however, requires at least third-order kernel estimates for meaningful implementation, provided also that $g(0) \neq 0$ and $h(0) \neq 0$. These requirements have limited its practical use to date. Nonetheless, the kernel values along an axis offer an easy way of estimating the prior filter $g(\tau)$ (within a scalar), provided that $g(0)h(0) \neq 0$, since:

$$k_2^{\mathrm{LNM}}(\tau_1, 0) = \alpha_2 h(0) g(0) \cdot g(\tau_1) \tag{4.55}$$

Subsequently, the posterior filter $h(\tau)$ can be estimated (within a scalar) through deconvolution of the estimated $g(\tau)$ from the first-order Volterra kernel (see Equation (4.50)).

It is evident that the prior filter L and/or the posterior filter M can be estimated (within a scalar) from the first-order and second-order Volterra kernels of any of these cascades (L-N-M, L-N, N-M), following the steps described above. Subsequently, the static nonlinearity can be estimated by plotting (or regressing) the reconstructed internal signal $z(t)$ on the reconstructed internal signal $v(t)$ in the case of the L-N-M cascade, or the corresponding signals in the other two cases (e.g., plotting the output $y(t)$ versus the reconstructed internal signal $v(t)$ in the case of the L-N cascade). "Reconstructed" $v(t)$ signal implies convolution of the input signal with the estimated prior filter $g(\tau)$, and "reconstructed" $z(t)$ signal implies deconvolution of the estimated posterior filter $h(\tau)$ from the output signal.

Clearly, there is a scaling ambiguity between the filters and the static nonlinearity (i.e., the filters can be multiplied by an arbitrary nonzero scalar and the input-output relation of the cascade remains intact by adjusting properly the scale(s) of the static nonlinearity). For instance, let us assume that the prior and the posterior filters of the L-N-M cascade are multiplied by the scalars c_L and c_M respectively. Then the static nonlinearity $z = f(v)$ is adjusted to the nonlinearity:

$$f(v) = \frac{1}{c_M} f(c_L v) \tag{4.56}$$

in order to maintain the same input-output relation in the overall cascade model. This fact implies that the estimated filters can be normalized without any loss of generality.

The foregoing discussion elucidates the way of how to achieve complete identification of these three types of cascade systems, provided that their first-order and second-order Volterra kernels can be estimated accurately (within a scalar). This raises the issue of possible estimation bias due to higher order terms when a truncated second-order Volterra model is used for kernel estimation. For such bias to be avoided, either a complete Volterra model (with all the significant nonlinear terms) ought to be used *or a Wiener model of second order* that offers an unbiased solution in this particular case. It is this latter case that deserves proper attention for these cascade systems, because it offers an attractive practical solution to the problem of high-order model estimation. The reason for this is found in the fact that the Wiener kernels of first-order and second-order are proportional to their Volterra counterparts for these cascades systems [Marmarelis & Marmarelis, 1978]. Specifically, the Wiener kernel expressions for the L-N-M cascade are found using Equation (2.57) to be:

$$h_1(\tau) = k_1(\tau) \cdot \sum_{m=0}^{\infty} \frac{(2m+1)!}{m!2^m} \alpha_{2m+1} \left[P \int_0^{\infty} g^2(\lambda) d\lambda \right]^m \tag{4.57}$$

$$h_2(\tau_1, \tau_2) = k_2(\tau_1, \tau_2) \cdot \sum_{m=1}^{\infty} \frac{(2m)!}{(m-1)!2^m} \alpha_{2m} \left[P \int_0^{\infty} g^2(\lambda) d\lambda \right]^{m-1} \tag{4.58}$$

The proportionality factor between the Volterra and the Wiener kernels depends on the coefficients of the polynomial (or Taylor series) static nonlinearity of the same parity (i.e., odd for h_1 and even for h_2). The proportionality factor also depends on the variance of the prior filter L for a GWN input with power level P, which is given by :

$$\mathrm{Var}\left[v(t) \right] = P \int_0^{\infty} g^2(\lambda) d\lambda \tag{4.59}$$

Thus, estimation of the normalized first-order and second-order Volterra kernels can be achieved in practice for an L-N-M cascade system *of any order* by means of estimation of their Wiener counterparts when a GWN input is available. Subsequently, the estimation of each cascade component separately can be achieved by the aforementioned methods for *any order* of nonlinearity.

The feasibility of estimating such cascade models of arbitrary order of nonlinearity has contributed to their popularity. Some illustrative examples are given in Chapter 6.

Because of its relative simplicity and the fact that cascade operations appear natural for information processing in the nervous system, the L-N-M cascade (or "sandwich model") received early attention in the study of sensory systems by Spekreijese and his colleagues, who used a variant of the "describing function" approach employing a combination of sinusoidal and noise stimuli [Spekreije, 1969]. A few years later, Korenberg analyzed the sandwich model in the Volterra-Wiener context [Korenberg, 1973b]. This pioneering work was largely ignored until it was properly highlighted in [Marmarelis & Marmarelis, 1978], leading to a number of subsequent applications to physiological systems.

We conclude this section by pointing out that the aforementioned three types of cascade systems cannot be distinguished by means of the first-order kernel alone (Volterra or Wiener), but the second-order kernel is necessary if the static nonlinearity has an even component, or the third-order kernel is required if the static nonlinearity is odd. Longer cascades can also be studied using the general results on cascaded Volterra systems presented earlier; however, the attractive simplifications of the L-N-M cascade (and its L-N or N-M offsprings) are lost when more nonlinearities are appended to the cascade.

4.1.3. Volterra Models of Systems with Lateral Branches

The possible presence of lateral feedforward branches in a system may take the form of additive parallel branches (the simplest case) or modulatory feedforward branches that either multiply the output of another branch or affect the characteristics (parameters or kernels) of another system component (see Figure 4.11).

In the simple case of additive parallel branches (see Figure 4.11a), the Volterra kernels of the overall system are simply the sum of the component kernels of the respective order:

$$k_r(\tau_1,...,\tau_r) = a_r(\tau_1,...,\tau_r) + b_r(\tau_1,...,\tau_r) \tag{4.60}$$

where $\{a_r\}$ and $\{b_r\}$ are the r th-order Volterra kernels of A and B respectively.

In the case of a multiplicative branch (see Figure 4.11b), the system output is given by:

$$y = \left[a_0 + a_1 \otimes x + a_2 \otimes x^2 + ...\right]\left[b_0 + b_1 \otimes x + b_2 \otimes x^2 + ...\right] \tag{4.61}$$

Thus, the Volterra kernels of the overall system are:

$$k_0 = a_0 b_0 \tag{4.62}$$

$$k_1(\tau) = a_0 b_1(\tau) + b_0 a_1(\tau) \tag{4.63}$$

$$k_2(\tau_1,\tau_2) = a_0 b_2(\tau_1,\tau_2) + b_0 a_2(\tau_1,\tau_2) + \frac{1}{2}\left[a_1(\tau_1) b_1(\tau_2) + a_1(\tau_2) b_1(\tau_1)\right] \tag{4.64}$$

$$\vdots$$

$$k_r(\tau_1,...,\tau_r) = \sum_{j=0}^{r} a_j(\tau_1,...,\tau_j) b_{r-j}(\tau_{j+1},...,\tau_r) \tag{4.65}$$

for $\tau_1 \geq \tau_2 \geq ... \geq \tau_r$, so that the general expression for the r th-order kernel need not be symmetrized with respect to the arguments $(\tau_1,...,\tau_r)$.

In the case of the "regulatory" branch B of Figure 4.11c, the Volterra kernel expressions for the overall system will depend on the specific manner in which the output z of component B influences the internal characteristics of the output-generating component A. For instance, if $z(t)$ multiplies (modulates) the first-order kernel of component A, then:

$$y = a_0 + \left[b_0 + b_1 \otimes x + b_2 \otimes x^2 + ...\right] a_1 \otimes x + a_2 \otimes x^2 + ... \tag{4.66}$$

and the Volterra kernels of this system are:

$$k_r(\tau_1,...,\tau_r) = a_r(\tau_1,...,\tau_r) + a_1(\tau_1) b_{r-1}(\tau_1,...,\tau_r) \tag{4.67}$$

for $\tau_1 \geq \tau_2 \geq ... \geq \tau_r$, to avoid symmetrizing the last term of Equation (4.67).

This latter category of "regulatory" branches may attain numerous diverse forms that will define different kernel relations. However, the method by which the kernel expressions are derived in all cases remains the same and relies on expressing the output in terms of the input using Volterra representations.

Note that the component subsystems may also be expressed in terms of parametric models (e.g., differential equations). Then, the equivalent nonparametric model of each component must be used to derive the Volterra kernels of the overall system in terms of the component kernels. An example of this was given in Section 1.4 for the "minimal model" of insulin-glucose interactions.

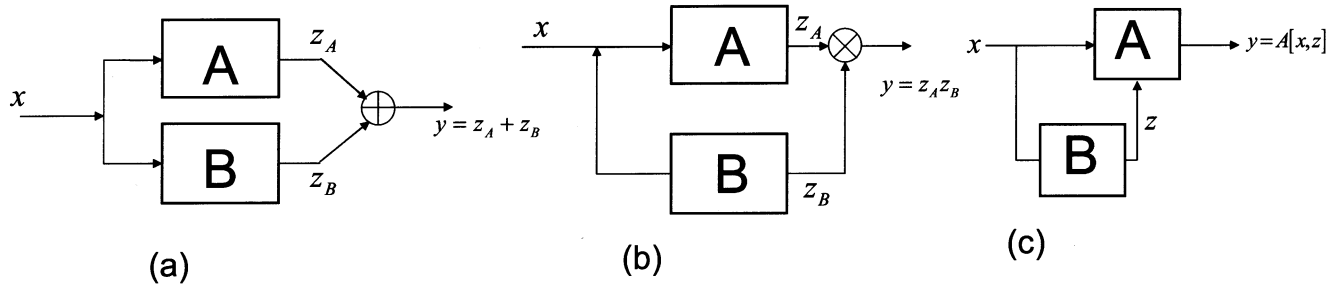


Figure 4.11
Configurations of modular models with lateral branches (a) two parallel branches converging at an adder; (b) two parallel branches converging at a multiplier; (c) a lateral branch B modulating component A.

4.1.4. Volterra Models of Systems with Feedback Branches

Systems with feedback branches constitute a very important class of physiological systems because of the critical role of feedback mechanisms in maintaining stable operation under normal or perturbed conditions (homeostasis and autoregulation). Feedback mechanisms may attain diverse forms, including closed-loop and nested-loop configurations discussed in Chapter 10. In this section, we derive the Volterra kernels for certain basic feedback configurations depicted in Figure 4.12.

The simplest case of Figure 4.12a exhibits additive feedback that can be expressed as the integral input-output equation:

$$y = a_1 \otimes \left[x + b_1 \otimes y + b_2 \otimes y^2 + ... \right] + a_2 \otimes \left[x + b_1 \otimes y + b_2 \otimes y^2 + ... \right]^2 + ... \tag{4.68}$$

where we have assumed that $a_0 = 0$ and $b_0 = 0$ to simplify matters (which implies that $k_0 = 0$). This integral equation contains Volterra functionals of the input *and* output, suggesting the rudiments of the general theory presented in Chapter 10. The explicit solution of this integral equation (i.e., expressing the output signal as a Volterra series of the input signal) is rather complicated but it may be achieved by balancing Volterra terms of the same order. Thus, balancing terms of first-order yields:

$$k_1 \otimes x = a_1 \otimes x + a_1 \otimes b_1 \otimes k_1 \otimes x \tag{4.69}$$

which can be solved in the frequency domain to yield:

$$K_1(\omega) = \frac{A_1(\omega)}{1 - A_1(\omega) B_1(\omega)} \tag{4.70}$$

Equation (4.70) is well known from linear feedback system theory. Balancing terms of second order, we obtain:

$$k_2 \otimes x^2 = a_1 \otimes \left[b_1 \otimes k_2 \otimes x^2 + b_2 \otimes (k_1 \otimes x)^2 \right] + a_2 \otimes \left[x^2 + (b_1 \otimes k_1 \otimes x)^2 + x(b_1 \otimes k_1 \otimes x) \right] \tag{4.71}$$

which can be solved in the frequency domain to yield the second-order Volterra kernel of the feedback system:

$$K_2(\omega_1, \omega_2) = \{ A_1(\omega_1 + \omega_2) B_2(\omega_1, \omega_2) K_1(\omega_1) K(\omega_2) + A_2(\omega_1, \omega_2) [1 + B_1(\omega_1) B_1(\omega_2) K_1(\omega_1) K_1(\omega_2) +$$

$$+ \frac{1}{2} [B_1(\omega_1) K_1(\omega_1) + B(\omega_2) K_1(\omega_2)] \} [1 - A_1(\omega_1 + \omega_2) B_1(\omega_1 + \omega_2)]^{-1} \tag{4.72}$$

This approach can be extended to any order, resulting in kernel expressions of increasing complexity. Obviously, these expressions are simplified when either A or B is linear. For instance, if the forward component A is linear, then the second-order kernel becomes:

$$K_2(\omega_1, \omega_2) = A_1(\omega_1 + \omega_2) B_2(\omega_1, \omega_2) K_1(\omega_1) K_1(\omega_2) [1 - A_1(\omega_1 + \omega_2) B_1(\omega_1 + \omega_2)]^{-1} \tag{4.73}$$

and the third-order kernel is given by:

$$K_3(\omega_1, \omega_2, \omega_3) = \left\{ \frac{2}{3} A_1(\omega_1 + \omega_2 + \omega_3) [B_2(\omega_1, \omega_2) K_2(\omega_1, \omega_2) K_1(\omega_3) + \right.$$

$$+ B_2(\omega_2, \omega_3) K_2(\omega_2, \omega_3) K_1(\omega_1) + B_2(\omega_3, \omega_1) K_2(\omega_3, \omega_1) K_1(\omega_2)]$$

$$\left. + B_3(\omega_1, \omega_2, \omega_3) K_1(\omega_1) K_1(\omega_2) K_1(\omega_3) \right\} [1 - A_1(\omega_1 + \omega_2 + \omega_3) B_1(\omega_1 + \omega_2 + \omega_3)]^{-1} \tag{4.74}$$

This case of the linear forward and nonlinear feedback is discussed again in the following section in connection with nonlinear differential equation models.

We examine now the multiplicative feedback of Figure 4.12b. The input-output integral equation (assuming $a_0 = k_0 = 0$ but $b_0 \neq 0$) is:

$$y = a_1 \otimes \left[x \left(b_0 + b_1 \otimes y + b_2 \otimes y^2 + \ldots \right) \right] + a_2 \otimes \left[x \left(b_0 + b_1 \otimes y + b_2 \otimes y^2 + \ldots \right) \right]^2 + \ldots \tag{4.75}$$

which yields the first-order balance equation:

$$k_1 \otimes x = a_1 \otimes (b_0 x) \tag{4.76}$$

from which the first-order Volterra kernel of the multiplicative feedback system is derived to be:

$$K_1(\omega) = b_0 A_1(\omega) \tag{4.77}$$

The second-order balance equation is:

$$k_2 \otimes x^2 = a_1 \otimes \left[x\left(b_1 \otimes k_1 \otimes x\right)\right] + a_2 \otimes \left(b_0 x\right)^2 \tag{4.78}$$

which yields the second-order Volterra kernel:

$$K_2(\omega_1+\omega_2) = b_0^2 A_2(\omega_1,\omega_2) + \frac{b_0}{2} A_1(\omega_1+\omega_2)\left[B_1(\omega_1)A_1(\omega_1) + A_1(\omega_2)A_1(\omega_2)\right] \tag{4.79}$$

Note that the kernel expressions for multiplicative nonlinear feedback are simpler than their counterparts for additive nonlinear feedback. The case of "regulatory" feedback of Figure 4.12c depends on the specific manner in which the feedback signal $z(t)$ influences the characteristics of the forward component A and will not be discussed further in the interest of space.

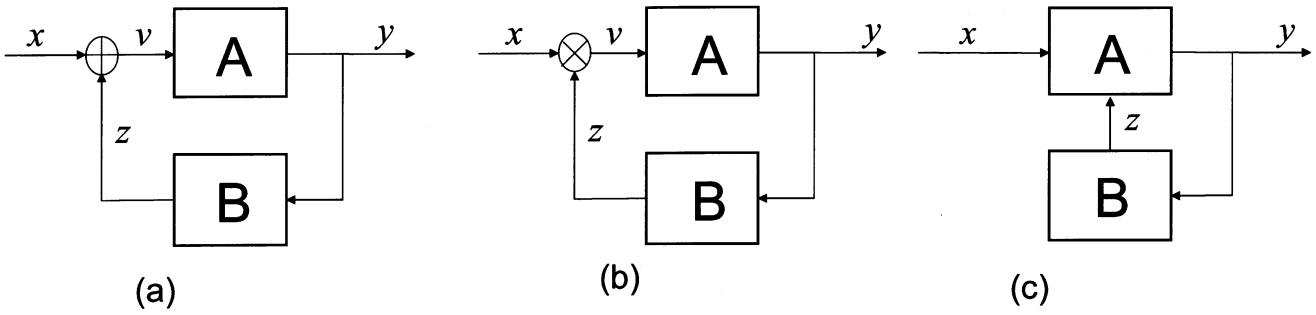


Figure 4.12
Configurations of modular models with feedback branches: (a) additive feedback branch B; (b) multiplicative feedback branch B; (c) modulatory feedback branch B; all acting on the forward component A.

4.1.5. Nonlinear Feedback Described by Differential Equations

This case was first discussed in Section 3.2 and is revisited here in order to elaborate on the relationship between parametric models described by nonlinear differential equations and modular feedback models. It is evident that any of the component subsystems (A and/or B), discussed above in connection with modular feedback systems/models, can be described equivalently by a parametric model and converted into an equivalent nonparametric model using the methods presented in Section

3.4. In this section, we will elaborate further on the case of a system with a linear forward and weak nonlinear feedback shown in Figure 4.13, that is described by the differential equation:

$$L(D)y + \in f(y) = M(D)x \tag{4.80}$$

where $|\in| \ll 1$, and $L(D)$, $M(D)$ are polynomials in the differential operator $D \equiv \frac{d(\cdot)}{dt}$. If the function $f(\cdot)$ is analytic or can be approximated to an arbitrary degree of accuracy by a power series (note that the linear term is excluded since it can be absorbed into L) as [Marmarelis, 1991]:

$$f(y) = \sum_{n=2}^{\infty} \alpha_n y^n \tag{4.81}$$

then the resulting Volterra kernels are:

$$K_1(s) = \frac{M(s)}{L(s)} \tag{4.82}$$

$$K_n(s_1, ..., s_n) = -\in \alpha_n K_1(s_1) \ldots K_1(s_n) / L(s_1 + \ldots + s_n) \tag{4.83}$$

where terms of order $\in^2$ or higher have been considered negligible.

The first-order Wiener kernel in this case is:

$$H_1(j\omega) = K_1(j\omega)\left\{1 - \frac{\in}{L(j\omega)} \sum_{m=1}^{\infty} \frac{(2m+1)!}{m!} \left(\frac{P_\kappa}{2}\right)^m \alpha_{2m+1}\right\}$$

$$= K_1(j\omega)\left[1 - \frac{\in}{L(j\omega)} C_1(P)\right] \tag{4.84}$$

where κ is the integral of the square of k_1 and $P_\kappa = (P\kappa)$. The second-order Wiener kernel is:

$$H_2(j\omega_1, j\omega_2) = -\in \frac{K_1(j\omega_1) K_1(j\omega_2)}{L(j\omega_1, j\omega_2)} \sum_{m=0}^{\infty} \frac{(2m+2)!}{m!2} \left(\frac{P_\kappa}{2}\right)^m \alpha_{2m+2}$$

$$= -\in \frac{K_1(j\omega_1) K_1(j\omega_2)}{L(j\omega_1 + j\omega_2)} C_2(P) \tag{4.85}$$

We observe that, as the input power level varies, the waveform of the first-order Wiener kernel changes but the second-order Wiener kernel remains unchanged in shape and changes only in scale. Note that the functions $C_1(P)$ and $C_2(P)$ are power series (or polynomials) in (P_κ) and characteristic

of the system nonlinearities. The Wiener kernels approach their Volterra counterparts as the input power level diminishes (as expected).

These results indicate that, for a system with linear forward and weak nonlinear feedback (i.e., $\left|\in \alpha_i\right|$ 1), the first-order Wiener kernel in the time domain will be:

$$h_1(\tau) = k_1(\tau) - \in C_1(P) \int_0^{\tau} k_1(\tau - \lambda)\, g(\lambda)\, d\lambda \tag{4.86}$$

and the second-order Wiener kernel will be:

$$h_2(\tau_1, \tau_2) = - \in C_2(P) \int_0^{\min(\tau_1, \tau_2)} k_1(\tau_1 - \lambda) k_1(\tau_2 - \lambda)\, g(\lambda)\, d\lambda \tag{4.87}$$

where $g(\lambda)$ is the inverse Fourier transform of $1/L(j\omega)$.

A companion issue to that of changing input power level is the effect of changing mean level of the experimental input (with white-noise or other perturbations superimposed on them) in order to explore different ranges of the system function. The resulting kernels for each different mean level μ of the input will vary, if the input data are defined as the deviations from the mean level each time. To reconcile these different measurements, we can use a reference mean level μ_0 in order to refer the kernels $\left\{k_n^{\mu}\right\}$ obtained from different mean levels μ to the reference kernels $\left\{k_n^{0}\right\}$ according to the relation:

$$k_n^{\mu}(\tau_1, ..., \tau_n) = \sum_{i=0}^{\infty} \frac{(n+i)!}{n!\, i!} (\mu - \mu_0)^i \int_0^{\infty} ... \int k_{n+i}^{0}(\tau_1, ..., \tau_n, \sigma_1, ..., \sigma_i)\, d\sigma_1 ... d\sigma_i \tag{4.88}$$

The first-order Wiener kernel for this class of systems with static nonlinear feedback is given in terms of the reference Volterra kernels (when $\mu_0 = 0$) by the expression:

$$h_1^{\mu}(\tau) = k_1^{0}(\tau) - \in A \int_0^{\tau} g(\lambda)\, k_1^{0}(\tau - \lambda)\, d\lambda \tag{4.89}$$

where:

$$A = \left\{ \sum_{\substack{m=0 \\ m+i \geq 1}}^{\infty} \sum_{i=0}^{\infty} \frac{(2m+i+1)!}{m!\, i!} \left(\frac{P_{\kappa}}{2} \right)^m (\mu\gamma)^i\, \alpha_{2m+i+1} \right\} \tag{4.90}$$

and γ is the integral of k_1^{0}. Note that the first-order Wiener kernel for $\mu \neq 0$ is also affected by the even-order terms of the nonlinearity in this case, unlike the case of $\mu = 0$ where it is affected only by the odd-order terms of the nonlinearity.

We use below computer simulations of systems with cubic and sigmoidal feedback to demonstrate the effect of changing GWN input power level and/or mean level on the waveform of the first-order Wiener kernel. This will help us explain the changes observed in the first-order Wiener kernels of some sensory systems when the GWN input power level and/order mean level is varied experimentally.

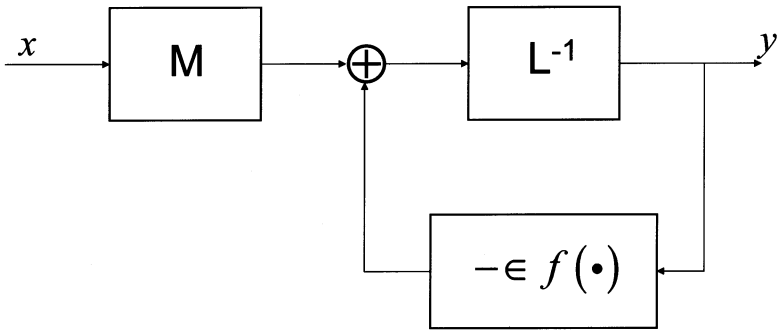


Figure 4.13
The modular form of the nonlinear feedback system described by Equation (4.80): L and M are linear dynamic operators and $f(\square)$ is an analytic function (or a function approximated by a polynomial). The factor $-\in$ in the feedback component denotes weak negative feedback.

Example 1

Cubic feedback systems

First, we consider a system with a low-pass forward linear subsystem (L^{-1}) and a cubic negative feedback $-\in y^3$, as shown in Figure 4.13 (for $M \equiv 1$). For $|\in| \square$ 1, the first-order Wiener kernel is given by Equation (4.86) as:

$$h_1(\tau) = g(\tau) - 3 \in P_\kappa \int_0^\tau g(\lambda) g(\tau - \lambda) d\lambda \tag{4.91}$$

where the first-order Volterra kernel $k_1(\tau)$ is identical to the impulse response function $g(\tau)$ of the low-pass linear forward subsystem in this case. For a zero-mean GWN input with power levels of $P = 1, 2, 4$ and cubic feedback coefficient $\in = 0.01$, the first-order Wiener kernel estimates are shown in Figure 4.14 along with the estimate for $\in = 0$ (i.e., no cubic feedback) which corresponds to $k_1(\tau) \equiv g(\tau)$. We observe a gradual decrease of damping (i.e., emergence of an increasing "undershoot") in the kernel estimates as P increases, consistent with Equation (4.91). This corresponds to a gradual increase of their bandwidth as P increases, as shown in Figure 4.15, where the FFT magnitudes of these kernel estimates are shown up to normalized frequency 0.1Hz (Nyquist frequency is

0.5Hz). We observe the gradual transition from an overdamped to an underdamped mode and a companion decrease of zero-frequency gain as P increases, similar to what has been observed in certain low-pass sensory systems such as retinal horizontal cells. Note that this system becomes unstable when P increases beyond a certain value.

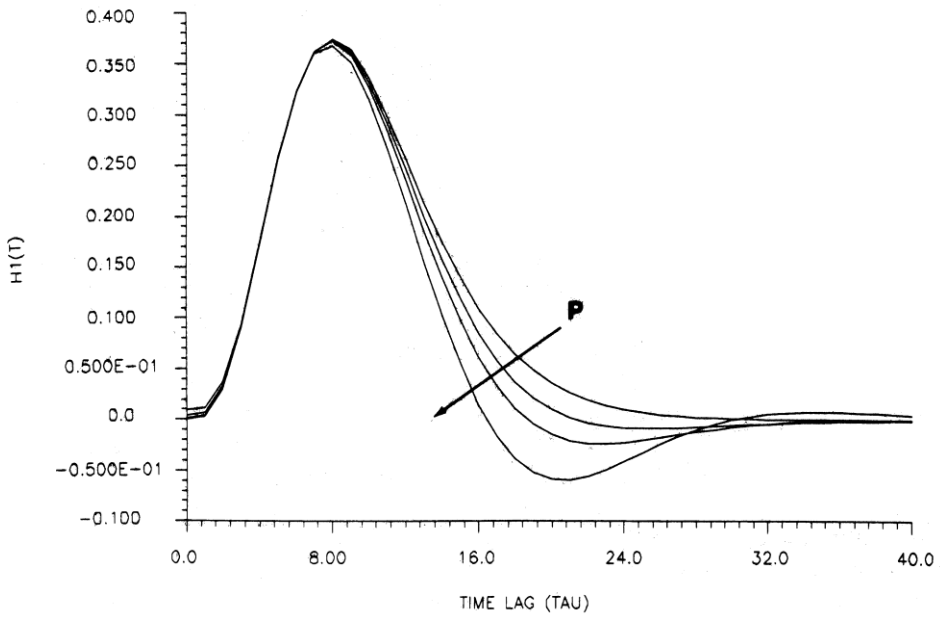


Figure 4.14
First-order Wiener kernel estimates of the system with negative cubic feedback ($\in = 0.001$) and a low-pass forward subsystem $g(\tau)$, obtained for P=1,2, and 4, along with the first-order Volterra kernel of the system ($P \to 0$) which is identical to $g(\tau)$ in this case. Observe the increasing undershoot in the kernel waveform as P increases [Marmarelis, 1991].

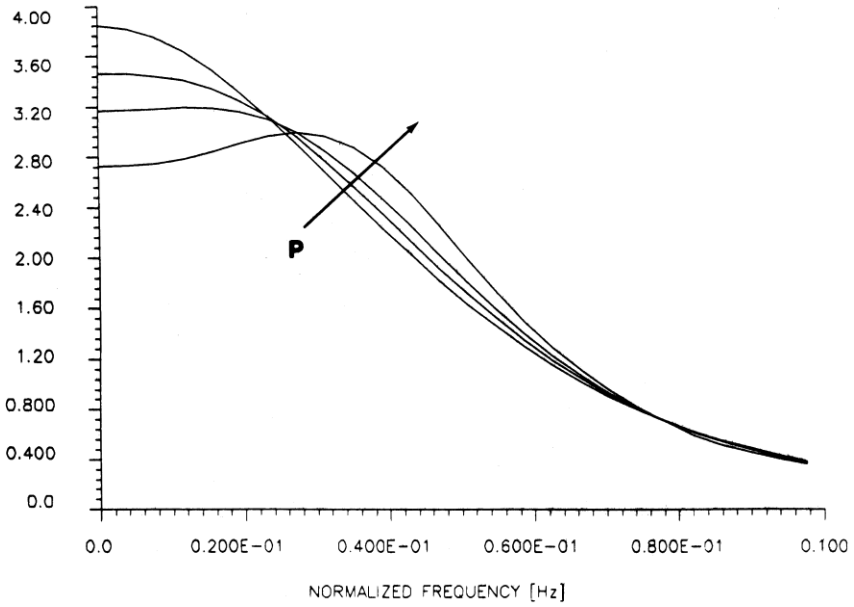


Figure 4.15
FFT magnitudes of the first-order kernels in Figure 4.14, plotted up to normalized frequency of 0.1 Hz. Observe the gradual transition from overdamped to underdamped mode and the increase of bandwidth as P increases [Marmarelis, 1991].

Next we explore the effect of varying the GWN input mean level μ while keeping $\in$ and P constant ($\in = 0.001$ and $P = 1$) using input mean levels of $\mu = 0, 1, 2$, and 3, successively. The obtained first-order Wiener kernel estimates are shown in Figure 4.16 and exhibit changes in their waveform as μ increases that are qualitatively similar to the ones induced by increasing P (i.e., increasing bandwidth and decreasing damping). According to the general expression of Equation (4.89), we have for this system:

$$h_1^{\mu}(\tau) = g(\tau) - 3 \in \left[P_{\kappa} + (\mu\gamma)^2 \right] \int_0^{\tau} g(\lambda)\, g(\tau - \lambda)\, d\lambda \tag{4.92}$$

We see that the effect of increasing P is similar to the effect of increasing μ^2 and the differential effect is proportional to κ and γ^2, respectively. Another point of practical interest is the difference between the first-order kernel (Volterra or Wiener) and the system response to an impulse. This point is often the source of confusion due to misconceptions ingrained by linear system analysis. For a third-order system, such as in this example for small $\in$, the response to an impulse input $x(t) = A\delta(t)$ is:

$$r_{\delta}(t) = Ag(t) - \in A^3 \int_0^t g(\lambda)\, g^3(t - \lambda)\, d\lambda \tag{4.93}$$

which is clearly different from the first-order Volterra kernel $k_1(t) \equiv g(t)$, or its Wiener counterpart given by Equation (4.91). Another point of practical interest is the response of this nonlinear feedback system to a step/pulse input $x(t) = Au(t)$, since pulse inputs have been used extensively in physiological studies. The system response to the pulse input is:

$$r_u(t) = A \int_0^t g(\tau)\, d\tau - \in A^3 \int_0^t g(\tau) \left\{ \int_{\tau}^t g(\lambda - \tau)\, d\lambda \right\}^3 d\tau \tag{4.94}$$

and the changes in the response waveforms as the pulse amplitude increases are demonstrated in Figure 4.17, where the responses of this system are shown for pulse amplitudes of 1, 2, and 4. The observed changes are qualitatively consistent with the previous discussion (i.e., the responses are less damped for stronger pulse inputs). However, the reader must note that we cannot obtain the first-order kernel (Volterra or Wiener) or the response to an impulse by differentiating the pulse response over time, as in the linear case. Observe also the sharp difference between on-set and off-set transient response, characteristic of nonlinear systems and so often seen in physiological systems.

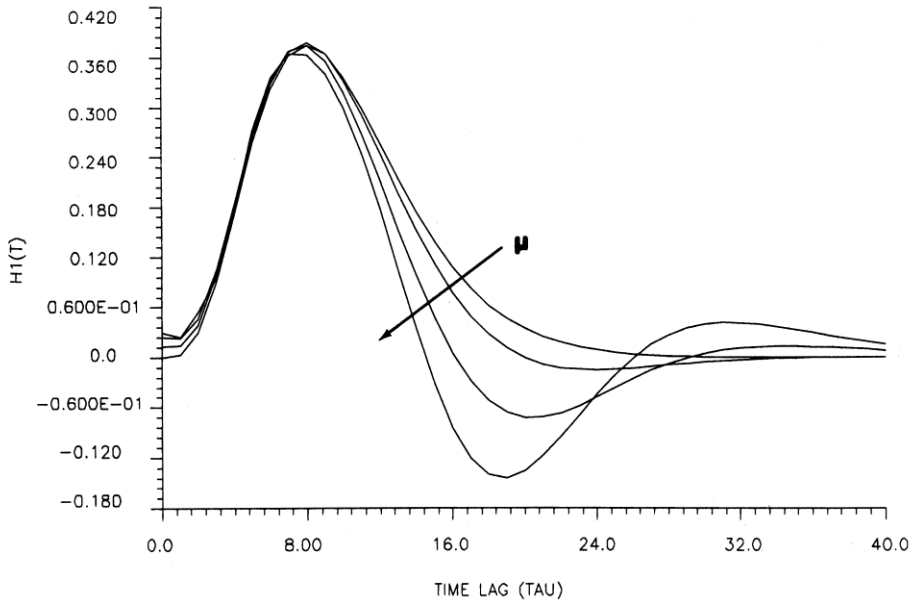


Figure 4.16
First-order Wiener kernel estimates of system with negative cubic feedback and a low-pass forward subsystem, obtained for $\mu = 0,1,2$, and 3 ($P = 1$, $\in = 0.001$ in all cases). The changes in kernel waveform follow the same qualitative pattern as in Figure 4.14 [Marmarelis, 1991].

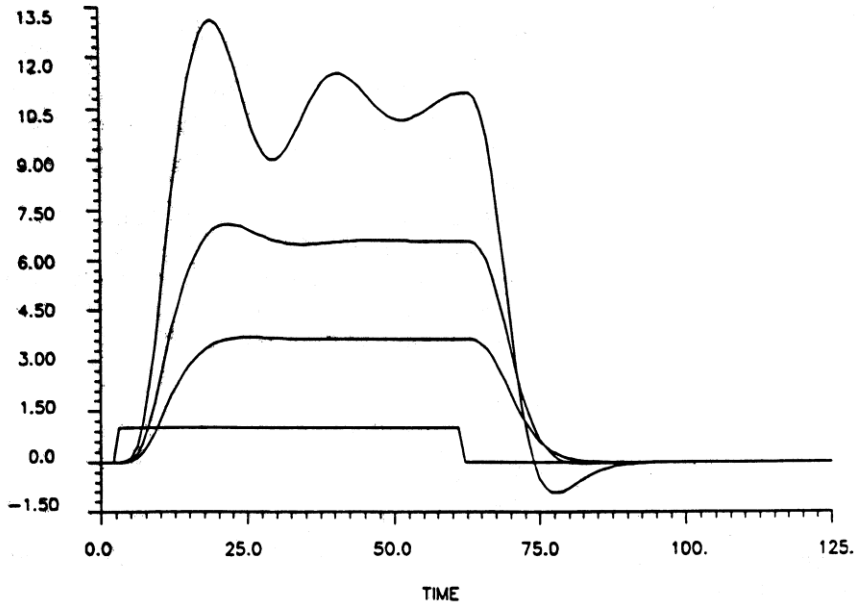


Figure 4.17
Responses of negative cubic feedback system ($\in = 0.001$) to pulse inputs of different amplitudes (1,2, and 4). Observe the gradual decrease of damping and latency of the on-set response as the pulse amplitude increases, as well as the difference between on-set and off-set transient responses [Marmarelis, 1991].

The steady-state value of the step response for various values of A is given by:

$$L(0)\,y + \in y^3 = A \tag{4.95}$$

(in the region of stability of this system) where $L(0) = 1/K_1(0)$ for this system. The steady-state values of the pulse response as a function of pulse amplitude are shown in Figure 4.18. Note that these values are different, in general, from the mean response values when the GWN input has nonzero mean.

Having examined the behavior of this nonlinear feedback system with a low-pass (overdamped) forward subsystem, we now examine the case of a band-pass (underdamped) forward subsystem with negative cubic feedback of $\in = 0.008$. The resulting first-order Wiener kernel estimates for increasing GWN input power level (viz., $P = 1, 2,$ and 4) are shown in Figure 4.19, along with the first-order Volterra kernel of the system (which is the same as the impulse response function of the linear forward subsystem) that corresponds to the case of $P = 0$. We observe a gradual deepening of the undershoot portion of the band-pass kernel accompanied by a gradual shortening of its duration as P increases (i.e., we see a gradual broadening of the system bandwidth and upward shift of the resonance frequency as P increases). This is demonstrated in Figure 4.20 in the frequency domain, where the FFT magnitudes of the kernels of Figure 4.19 are shown. The changes in the waveform of these kernels with increasing P are consistent with our theoretical analysis and mimic changes observed experimentally in band-pass systems (e.g. primary auditory fibers).

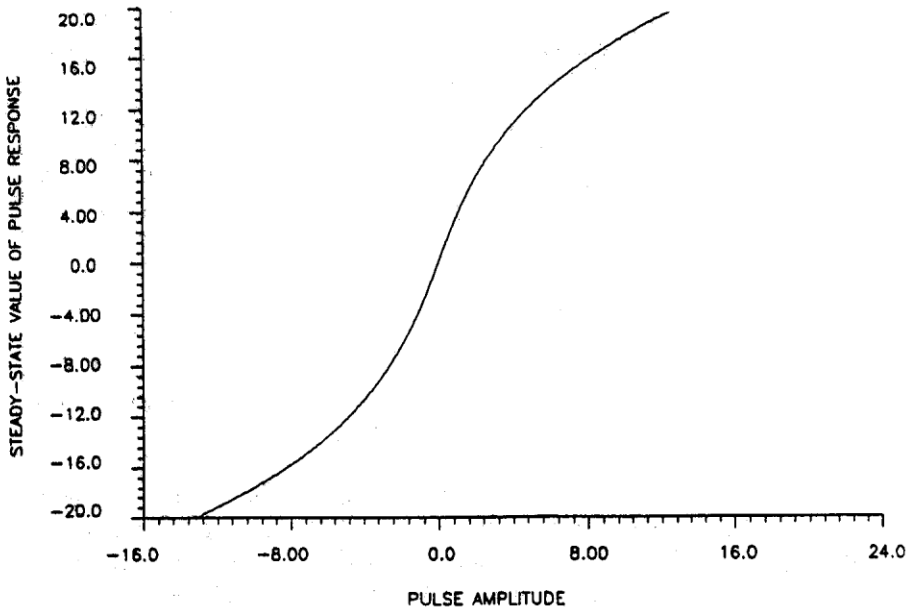


Figure 4.18
The steady-state values of the pulse responses as a function of input pulse amplitude for the negative cubic feedback system ($\in = 0.001$) [Marmarelis, 1991].

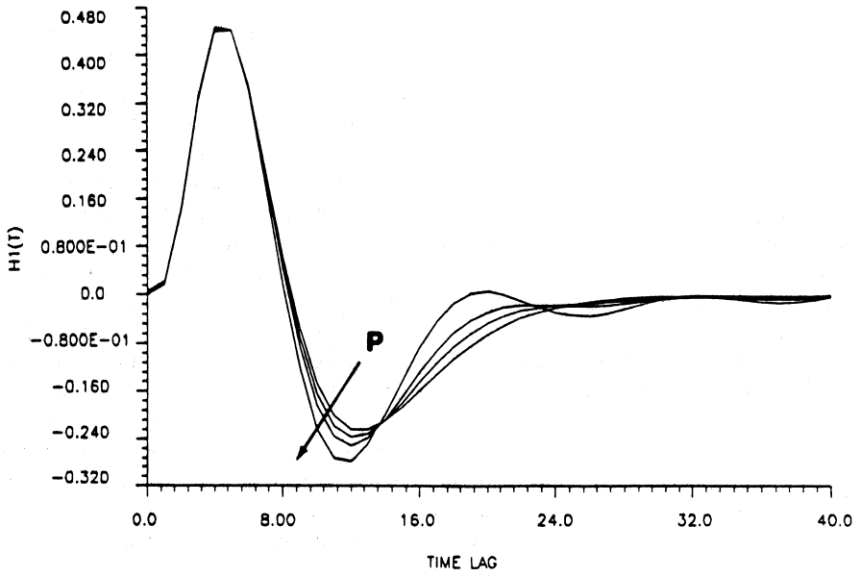


Figure 4.19
First-order Wiener kernels of the negative cubic feedback system ($\in = 0.008$) with the band-pass (underdamped) linear forward subsystem (corresponding to $P = 0$), for $P = 0.1, 1, 2,$ and 4. Observe the increasing undershoot as P increases [Marmarelis, 1991].

Note that the effect of increasing GWN input mean level on the first-order Wiener kernels is not significant, due to the fact that γ (i.e., the integral of k_1) is extremely small in this case—cf. Equation (4.92). Finally, the system response to input pulses of increasing amplitude ($A = 1, 2,$ and 4) are shown in Figure 4.21 and demonstrate increasing resonance frequency and decreasing damping in the pulse response as A increases. Note also that the steady-state values of the pulse responses are extremely small, and the on-set/off-set response waveforms are similar (with received polarity), due to the very small value of $\gamma = K_1(0)$ (cf. Equation (4.95)).

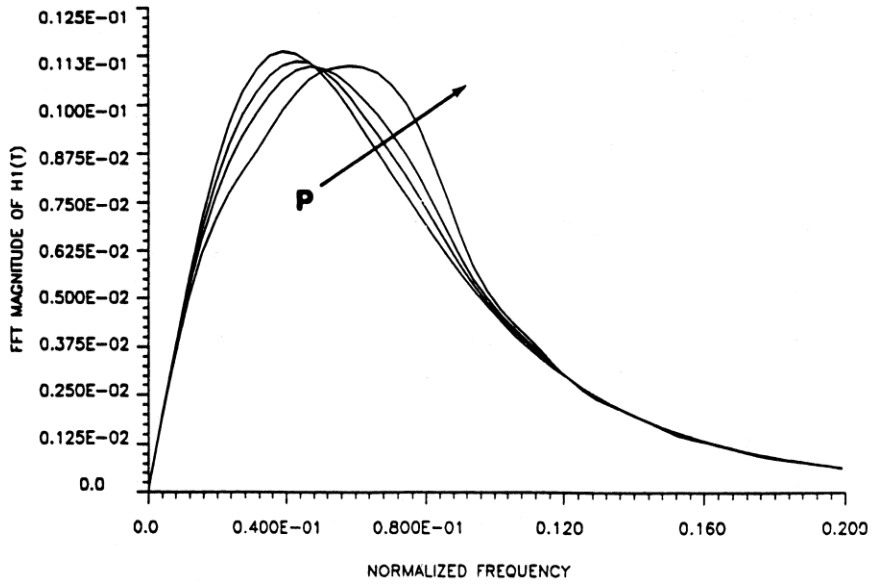


Figure 4.20
FFT magnitudes of the first-order Wiener kernels shown in Figure 4.19. Observe the gradual increase of bandwidth and upward shift of resonant frequency as P increases [Marmarelis, 1991].

Example 2

Sigmoid feedback systems

The next example deals with a sigmoid feedback nonlinearity which, unlike the cubic one, is bounded for any response signal amplitude. The *arctangent* function:

$$f(y) = \frac{2}{\pi}\arctan(\alpha y) \tag{4.96}$$

was used in the simulations ($\alpha = 0.25$) with the previous low-pass forward subsystem and the resulting first-order Wiener kernels for $P = 1$ and $\in = 0, 0.125,$ and 0.5 are shown in Figure 4.22. The qualitative changes in waveform are similar to the cubic feedback case for increasing input power level P or feedback strength $\in$. However, for fixed sigmoid feedback strength ($\in$) the kernels resulting from increasing GWN input power level P follow the *reverse* transition in waveform, as demonstrated in Figure 4.23, where the kernels obtained for $P = 1$, 4, 16, and 64 are shown for $\in = 0.25$ in all cases.

Bear in mind that the first-order Volterra kernel of this sigmoid feedback system is not the same as the impulse response function of the forward subsystem, but it is the impulse response function of the overall linear feedback system when the linear term of the sigmoid nonlinearity (i.e., its slope at zero) is incorporated in the (negative) feedback loop. Thus, the kernel waveform follows the previously described gradual changes from the impulse response function of the linear feedback system to that of the linear forward subsystem as P increases (i.e., the kernel waveform changes gradually from underdamped to overdamped as P increases and the gain of the equivalent linearized feedback decreases).

Because of the bounded nature of the (negative) sigmoid nonlinearity, large values of $\in$ and/or P do not lead to system instabilities as in the case of cubic feedback. Increasing values of $\in$ result in decreasing damping, eventually leading to *oscillatory* behavior. This is demonstrated in Figure 4.24, where the kernels for $\in = 0.5,$ 1, 2, and 4 are shown ($P = 1$). The oscillatory behavior of this system, for large values of $\in$, is more dramatically demonstrated in Figure 4.25, where the actual system responses $y(t)$ for $\in = 100$ and 1000 are shown ($P = 1$). The system goes into perfect oscillation regardless of the GWN input, due to the overwhelming action of the negative sigmoid feedback that is bounded and symmetric about the origin. The amplitude of this oscillation is proportional to $\in$, but is independent of the input power level. In fact, the oscillatory response remains the same in amplitude and frequency for *any* input signal (regardless of its amplitude and waveform) as long as the value of $\in$ is much larger than the maximum value of the input. The initial transient and the phase of the oscillation, however,

may vary according to the input power and waveform. The frequency of oscillation depends on the dynamics (time constants) of the linear forward subsystem. For instance, a low-pass subsystem with shorter memory (i.e., shorter time constants) leads to higher frequency of oscillation, and so does an underdamped system with the same memory extent.

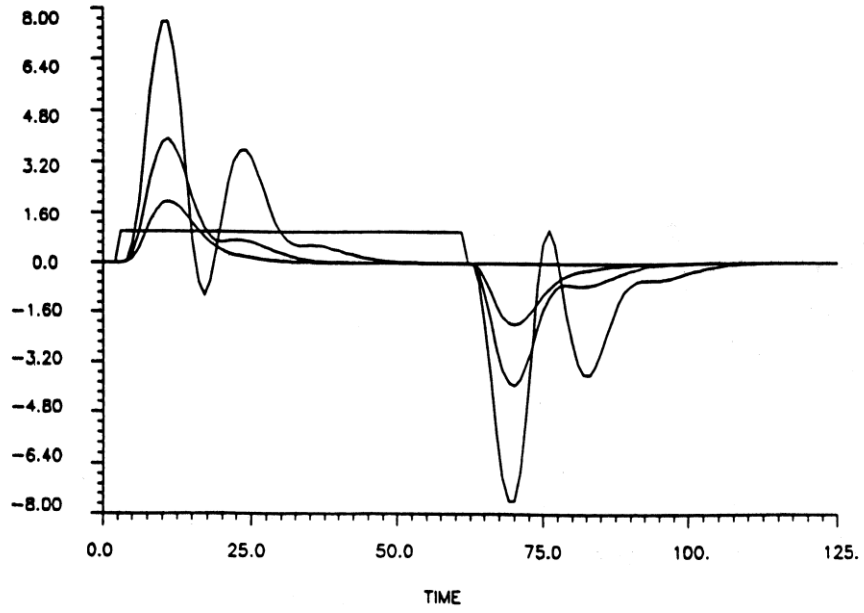


Figure 4.21
Response of negative feedback system ($\in = 0.008$) with underdamped forward to pulse inputs of different amplitudes $A = 1$, 2, and 4. Observe the increasingly underdamped response as A increases, and the negligible steady-state responses [Marmarelis, 1991].

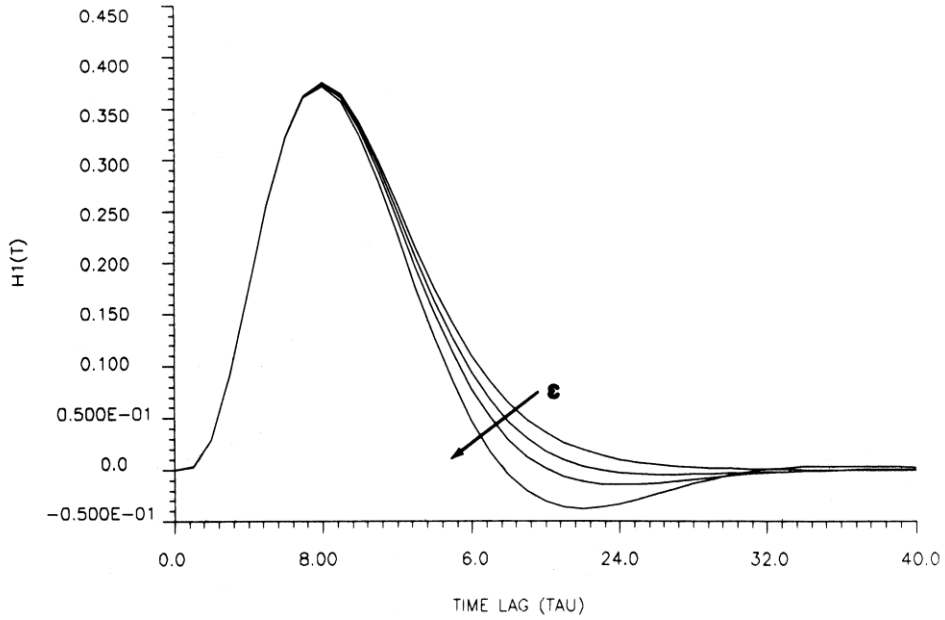


Figure 4.22
First-order Wiener kernel estimates of negative sigmoid feedback system with the previous low-pass forward subsystem for $\in = 0, 0.125,$ *and* 0.5 ($P = 1$, $\alpha = 0.25$ in all cases). Observe the similarity in changes of kernel waveform with the ones shown in Figure 4.14 [Marmarelis, 1991].

Although the case of oscillatory behavior is not covered formally by the Volterra-Wiener analysis because it violates the finite-memory requirement, it is of great interest in physiology because of the numerous and functionally important physiological oscillators. Therefore, it is a subject worthy of further exploration in the context of large negative compressive (e.g., sigmoid) feedback, because the foregoing observations are rather intriguing. For instance, can an oscillation of fixed frequency be initiated by a broad ensemble of stimuli that share only minimal attributes irrespective of waveform (e.g., having bandwidth and dynamic range within certain bounds) as long as the feedback gain is large?

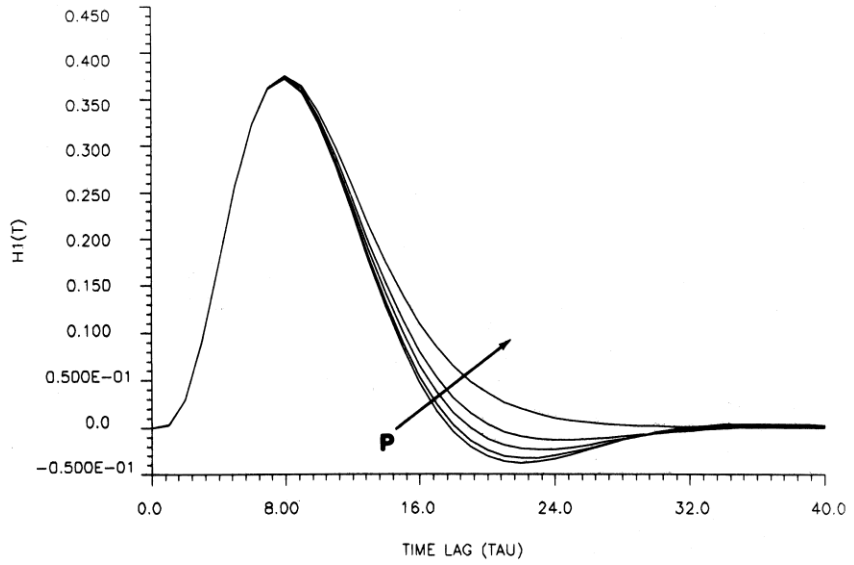


Figure 4.23
First order Wiener kernel estimates of negative sigmoid feedback system with the previous low-pass forward subsystem for $P = 1, 4, 16,$ and 64 ($\in = 0.25$, $\alpha = 0.25$ in all cases). Observe reverse pattern of kernel waveform changes from the ones in Figure 4.22 or 4.14 [Marmarelis, 1991].

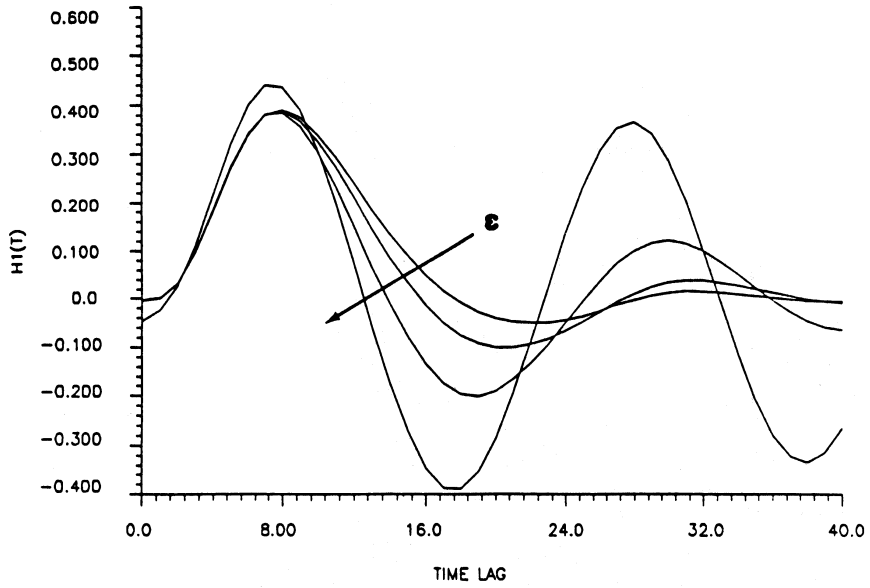


Figure 4.24
First-order Wiener kernels of negative sigmoid feedback system with low-pass (overdamped) forward, for $\in = 0.5,\ 1,\ 2,$ and 4. Observe transition to oscillatory behavior as $\in$ increases [Marmarelis, 1991].

The effect of varying slope of the sigmoid nonlinearity was also studied and a gradually decreasing damping with increasing slope was observed [Marmarelis, 1991]. This transition reaches asymptotically a limit in both directions of changing α values, as expected. For $\alpha \to \infty$, the sigmoid nonlinearity becomes the signum function and leads to perfect oscillations; and for $\alpha \to 0$ the gain of the feedback loop diminishes leading to a kernel identical to the impulse response function of the forward linear subsystem.

The effect of nonzero GWN input mean μ is similar to the effect of increasing P, i.e., the first-order Wiener kernels become more damped as μ increases, which indicates decreasing gain of the equivalent linearized feedback.

In the case of the underdamped forward subsystem and negative sigmoid feedback, the changes in the kernel waveform undergo a gradual transition from the linearized feedback system to the forward linear subsystem as the GWN input power level P increases. The two limit waveforms (for $P \to 0$ and $P \to \infty$) of the first-order Wiener kernel are shown in Figure 4.26 for $\in = 1$, $\alpha = 0.25$. The effect of the negative sigmoid feedback is less dramatic in this case, since the kernel retains its underdamped mode for all values of P. There is, however, a downward shift of resonance frequency and increase of damping when P increases, as indicated by the FFT magnitudes of the "limit" kernel waveforms shown in the right panel of Figure 4.26.

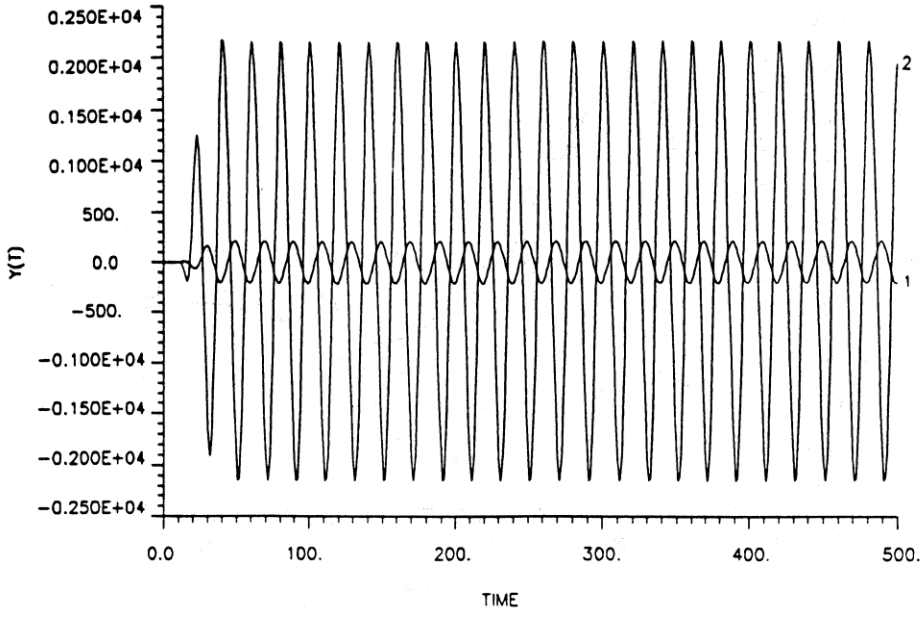


Figure 4.25
Oscillatory response of negative sigmoid feedback system for very large feedback gain $\in = 100$ and 1000, and GWN input ($P = 1$) [Marmarelis, 1991].

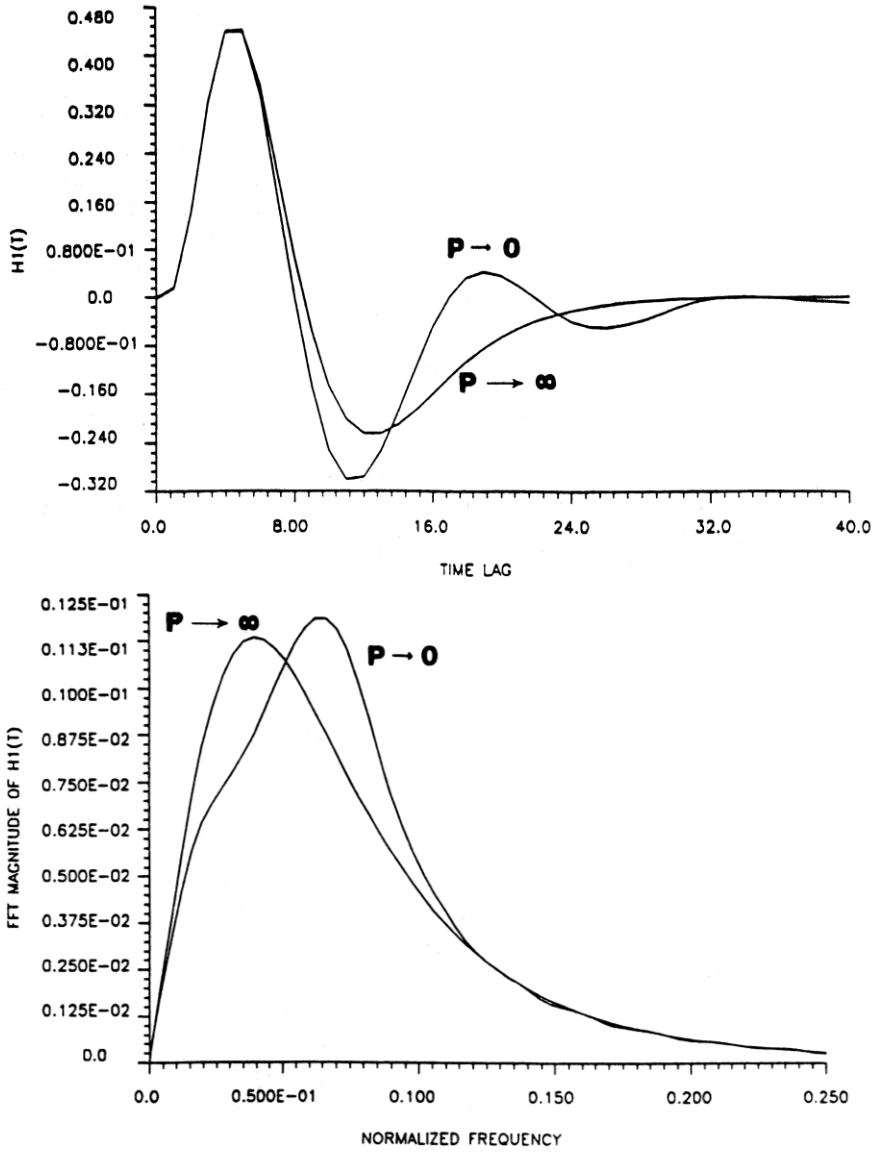


Figure 4.26
The two limit waveforms of the first-order Wiener kernel for the negative sigmoid feedback system ($\in=1$, $\alpha=0.258$) with underdamped forward, obtained for $P\to 0$ and $P\to\infty$ (left panel), and their FFT magnitudes (right panel). Observe the lower resonance frequency and increased damping for $P\to\infty$ [Marmarelis, 1991].

Example 3

Positive nonlinear feedback

The reverse transition in the first-order Wiener kernel waveform is observed when the polarity of the weak nonlinear feedback is changed, as dictated by Equation (4.91). Positive decompressive (e.g., cubic) feedback leads to a decrease in resonance frequency and higher gain values in the resonant region. Also, the reverse transition in kernel waveform occurs (i.e., upward shift of resonance frequency and decrease of damping with increasing P values) when the compressive (e.g., sigmoid) feedback becomes positive.

The great advantage of sigmoid versus cubic feedback is that stability of the system behavior is retained over a broader range of the input power level. For this reason, sigmoid feedback is an appealing candidate for plausible models of physiological feedback systems. For those systems that exhibit transitions to broader bandwidth and decreased damping as P increases, candidate models may include either negative decompressive (e.g., cubic) or positive compressive (e.g., sigmoid) feedback. For those systems that exhibit the reverse transition pattern (i.e., to narrower bandwidth and increased damping as P increases), candidate models may include either positive decompressive or negative compressive feedback.

Example 4

Second-order kernels of nonlinear feedback systems

Our examples so far have employed nonlinear feedback with odd symmetry (cubic and sigmoid), and our attention has focused on first-order Wiener kernels because these systems do not have even-order kernels. However, if the feedback nonlinearity is not odd-symmetric, then even-order kernels exist. An example of this is given for negative quadratic feedback of the form $\in y^2$ (for $\in = 0.08$, $P = 1$) where the previous band-pass (underdamped) forward subsystem is used. The resulting second-order Wiener kernel is shown in Figure 4.27 and has the form and size predicted by the analytical expression of Equation (4.87). The first-order Wiener kernel is not affected by the quadratic feedback for small values of $\in$.

It is important to note that Wiener analysis with nonzero GWN input mean yields even-order Wiener kernels (dependent on the nonzero input mean μ) even for cubic or sigmoid feedback systems, because a nonzero input mean defines an "operating point" that breaks the odd symmetry of the cubic or sigmoid nonlinearity. For instance, a negative cubic feedback system, where only K_1 and K_3 are assumed to be significant for small values of $\in$, has the second-order Wiener kernel:

$$\begin{aligned} H_2^{\mu}(\omega_1, \omega_2) &= 3\mu K_3(\omega_1, \omega_2, 0) \\ &= -3 \in \mu\gamma K_1(\omega_1) K_1(\omega_2) K_1(\omega_1 + \omega_2) \end{aligned} \tag{4.97}$$

Equation (4.97) implies that the second-order Wiener kernel will retain its shape but increase linearly in size with increasing μ (provided, of course, that $\in$ is small).

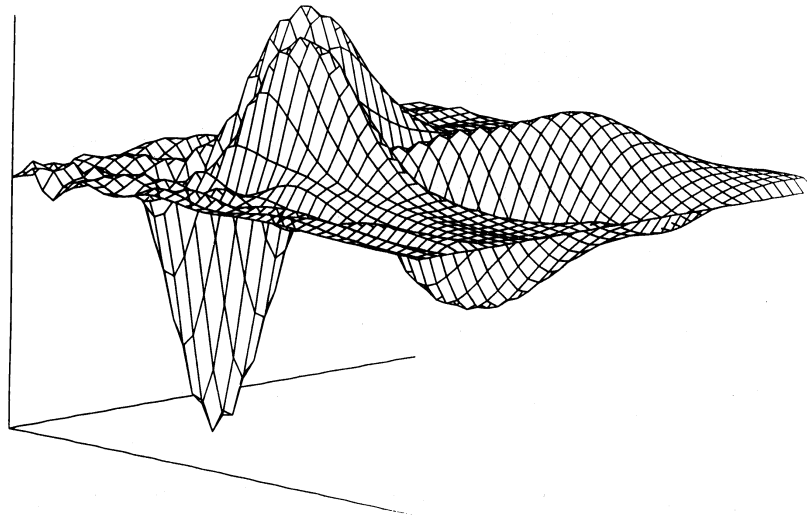


Figure 4.27
Second-order Wiener kernel of the negative quadratic feedback system with a band-pass (underdamped) forward subsystem ($\in = 0.08$, $P = 1$) [Marmarelis, 1991].

Nonlinear Feedback in Sensory Systems

The presented analysis of nonlinear feedback systems is useful in interpreting the Wiener kernel measurements obtained for certain visual and auditory systems under various conditions of GWN stimulation, as discussed below.

The Wiener approach has been applied extensively to the study of r*etinal cells* using band-limited GWN stimuli [e.g., Marmarelis & Naka, 1972,1973abcd,1974abcd]. In these studies, the experimental stimulus consists of band-limited GWN modulation of light intensity about a constant level of illumination, and the response is the intracellular or extracellular potential of a certain retinal cell (receptor, horizontal, bipolar, amacrine, ganglion). Wiener kernels (typically of first and second order) are subsequently computed from the stimulus-response data. The experiments are typically repeated for different levels of mean illumination (input mean) and various GWN input power levels in order to cover the entire physiological range of interest. It has been observed that the waveform of the resulting kernels generally varies with different input mean and/or power level. We propose that these changes in waveform may be explained by the presence of a nonlinear feedback mechanism, in accordance with the previous analysis. Note that these changes cannot be explained by the simple cascade models discussed in Section 4.1.2.

The first such observation was made in the early 1970s [Marmarelis & Naka, 1973b] on the changing waveform of first-order Wiener kernel estimates of horizontal cells in the catfish retina, obtained for two different levels of stimulation (low and high mean illumination levels with proportional

GWN modulation). The kernel corresponding to high level of stimulation was less damped and had shorter latency (shorter peak-response time) as shown in Figure 6.2. This observation was repeated later (e.g., [Naka et al. 1988; Sakai & Naka, 1985,1987]) for graduated values of increasing P and μ. The observed changes are qualitatively similar to the ones observed in our simulations of negative decompressive (cubic) feedback systems with an overdamped forward subsystem. However, the changes in latency time and kernel size are much more pronounced in the experimental kernels than in our simulations of negative cubic feedback systems presented earlier.

To account for the greater reduction in kernel size observed experimentally, we may introduce a compressive (static) nonlinearity in cascade with the overall feedback system that leads to an additional reduction of the gain of the overall cascade system as P and/or μ increase. On the other hand, a greater reduction in the peak-response (latency) time may require the introduction of another dynamic component in cascade with the feedback system.

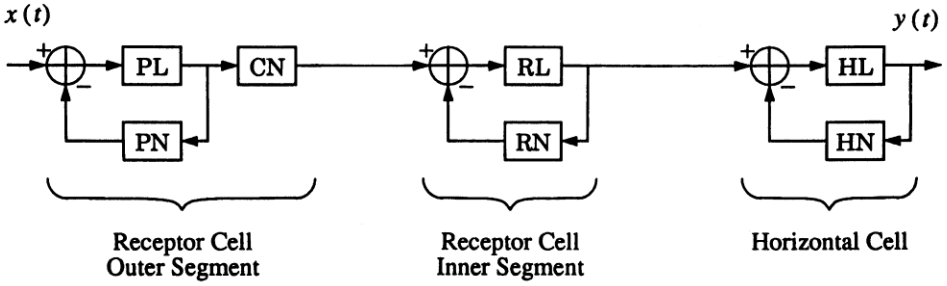


Figure 4.28

Schematic of the modular (block-structured) model of light→ horizontal cell system. Input $x(t)$ represents the light stimulus and output $y(t)$ the horizontal cell response. Each of the three cascaded segments of the model contains negative decompressive feedback and an overdamped forward component (shown in Figure 4.29). The static nonlinearity CN between the outer and inner segment of the receptor model component is compressive (sigmoidal) [Marmarelis, 1991].

Led by these observations, we propose the modular (block-structured) model, shown in Figure 4.28, for the light-to-horizontal cell system [Marmarelis, 1987]. This model is comprised of the cascade of three negative decompressive (cubic) feedback subsystems with different overdamped forward components (shown in Figure 4.29) and a compressive (sigmoidal) nonlinearity between the outer and the inner segments of the photoreceptor model component. The first part of this cascade model, comprised of the PL/PN feedback loop and the compressive nonlinearity CN, corresponds to the transformations taking place in the outer segment of the photoreceptor and represents the nonlinear dynamics of the phototransduction process. The second part, comprised of the RL/RN feedback loop, represents the nonlinear dynamic transformations taking place in the inner segment of the photoreceptors (including the receptor terminals). The third part, comprised of the HL/HN feedback

loop, represents the nonlinear dynamic transformations taking place in the horizontal cell and its synaptic junction with the receptor. Note that this model does not differentiate between cone/rod receptors and does not take into account spatial interactions or the triadic synapse with bipolar cells (see below).

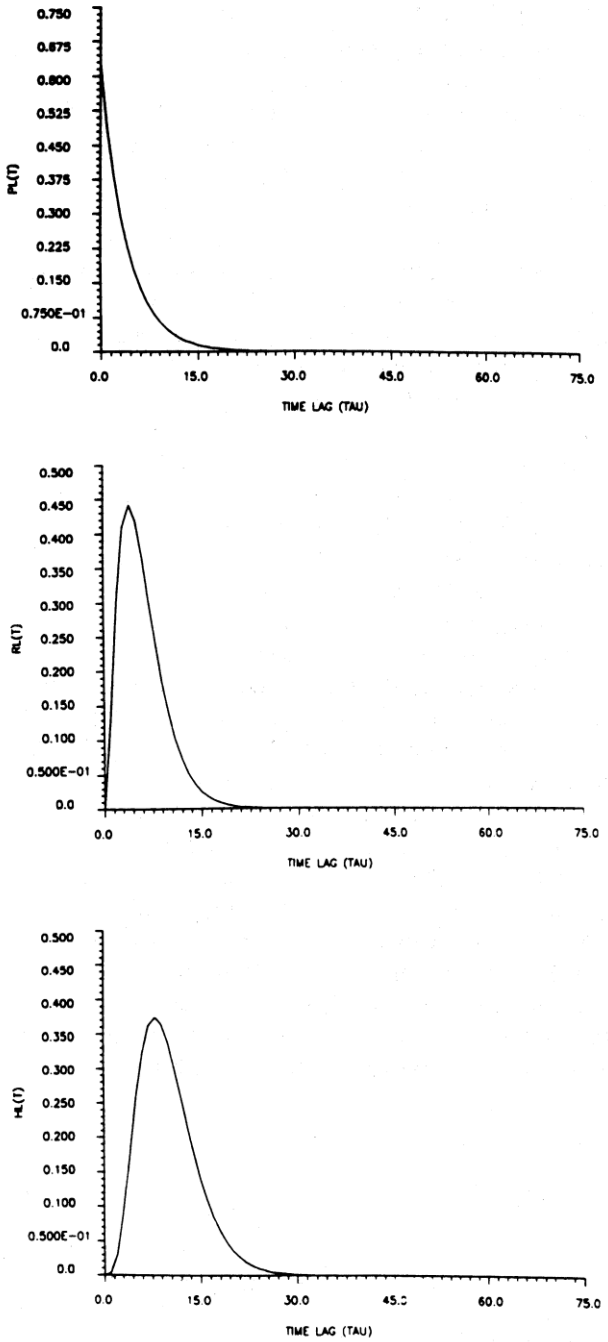
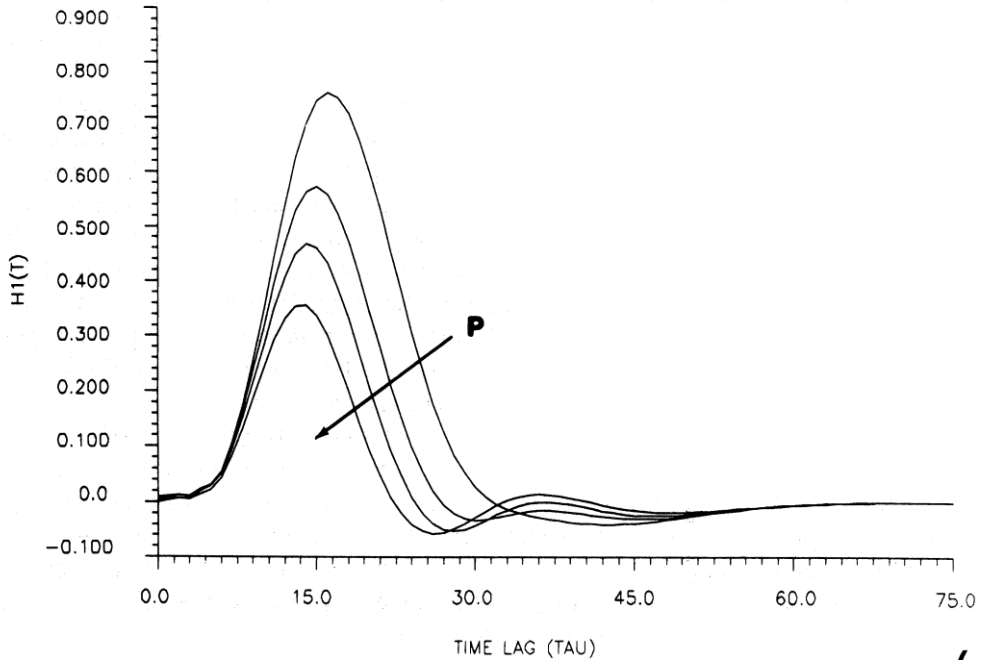
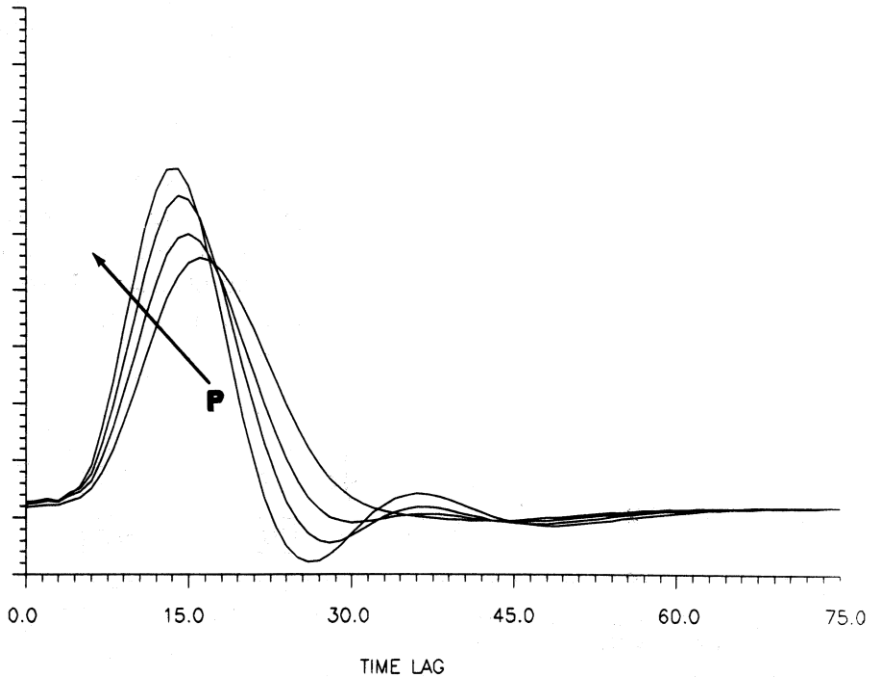


Figure 4.29
Impulse response functions of the linear overdamped forward components PL, RL, and HL used in the model of Figure 4.28. Note that the feedback nonlinearities PN, RN, and HN used in the model are decompressive (cubic) with coefficients $\in = 0.05, 0.10,$ and 0.01, respectively. The static nonlinearity CN is compressive (sigmoidal) of the form described by Equation (4.96) with $\alpha = 0.2$ ($\in = 2$) [Marmarelis, 1991].



a)



b)

Figure 4.30
(a) First-order Wiener kernels of the light-to-horizontal cell model shown in Figure 4.28, for $P = 0.5, 1, 2,$ and 4. Observe the gradual transition in kernel waveform akin to the experimentally observed; (b) the same kernels plotted in contrast sensitivity scale (i.e., each kernel scaled by its corresponding power level) [Marmarelis, 1991].

The first-order Wiener kernels of this model are shown in Figure 4.30 for GWN input power levels $P = 0.5,\ 1,\ 2,$ and 4, for the parameter values indicated in the caption of Figure 4.29. We observe kernel waveform changes that resemble closely the experimentally observed changes (note that hyperpolarization is plotted as a positive deflection) that are discussed in Section 6.1. Since experimentally obtained horizontal-cell kernels are usually plotted in the contrast sensitivity scale (i.e., scaled by the respective GWN input power level), we show in the same figure the kernels plotted in contrast sensitivity scale. The purpose of this demonstration is to show that the experimentally observed kernel waveform changes can be reproduced fairly well by a model of this form employing nonlinear feedback. The selected model components and parameters are dependent on the specific species, and the precise parameter values can be determined by repeated experiments (for different values of P and μ) and kernel analysis in the presented context for each particular physiological preparation.

We can extend the light-to-horizontal cell model to explain the experimentally observed changes in the waveform of first-order Wiener kernels of the light-to bipolar cell system (for increasing GWN input power level) [Marmarelis, 1987]. As shown in Figure 4.31, the response of the horizontal cell is subtracted from the response of the receptor (inner segment), and the resulting signal is passed through a nonlinear feedback component representing the triadic synapses (from the receptor terminals to the horizontal processes and bipolar dendrites) as well as the transformation of the postsynaptic potential through the bipolar dendrites. The resulting first-order Wiener kernels are similar to the experimentally observed ones (i.e., shorter latency, increased bandwidth, and increased sensitivity with increasing P) that are presented in Section 6.1.

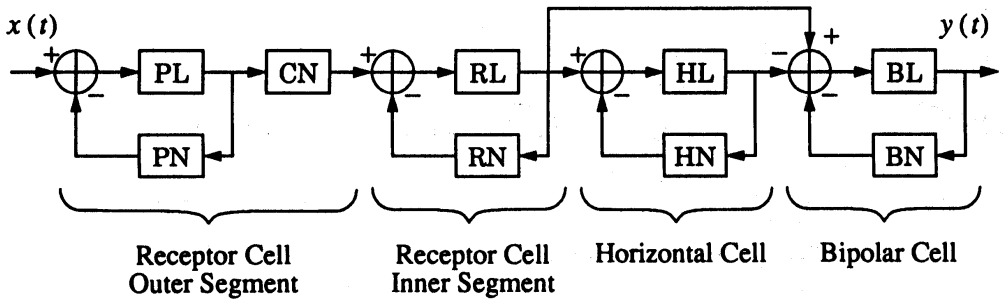


Figure 4.31
Schematic of the modular (block-structured) model of the light→ bipolar cell system described in the text [Marmarelis, 1991].

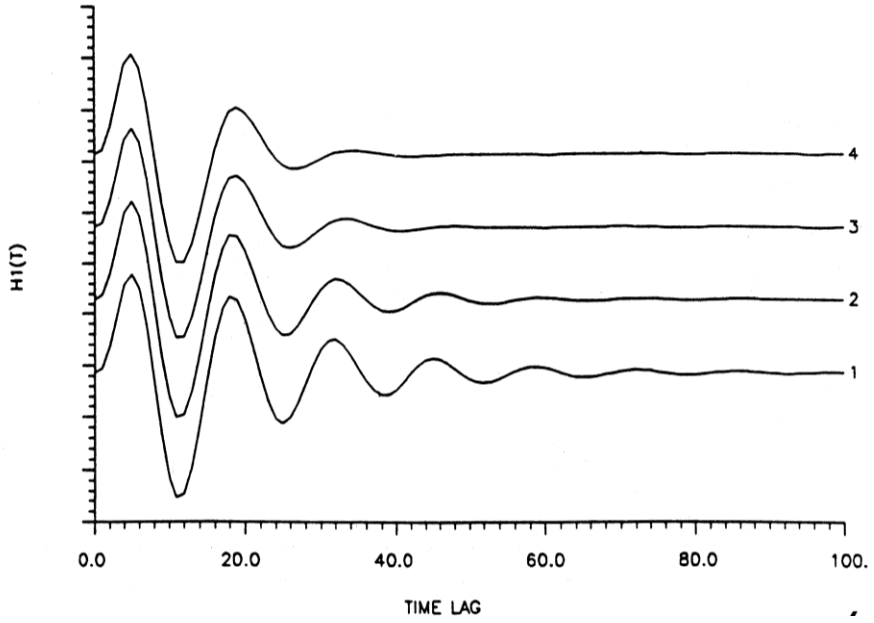
Beyond these mechanistic explanations, an important scientific question can be posed about the *teleological* reasons for the existence of decompressive feedback in retinal cells, in tandem with compressive nonlinearities. The presented analysis suggests that this is an effective functional design that secures sensory transduction over a very broad range of stimulus intensities while, at the same time, provides adequate (practically undiminishing) dynamic range of operation about a dynamically changing operating point (attuned to changing stimulus conditions). Furthermore, the gradual transition of the system functional characteristics towards faster response when the stimulus intensity and temporal changes are greater, would be a suitable attribute for a sensory system that has evolved under the requirements of rapid detection of changes in the visual field (threat detection) for survival purposes.

Another interesting example of a sensory system is the *auditory nerve fibers*, whose band-pass first-order Wiener kernel undergoes a transition to lower resonance frequencies as the input power level increases [Marmarelis, 1991]. This has been observed experimentally in primary auditory nerve fibers that have center (resonance) frequencies between 1.5 and 6KHz [Moller, 1983; Lewis et al. 2002b], as discussed in Section 6.1.

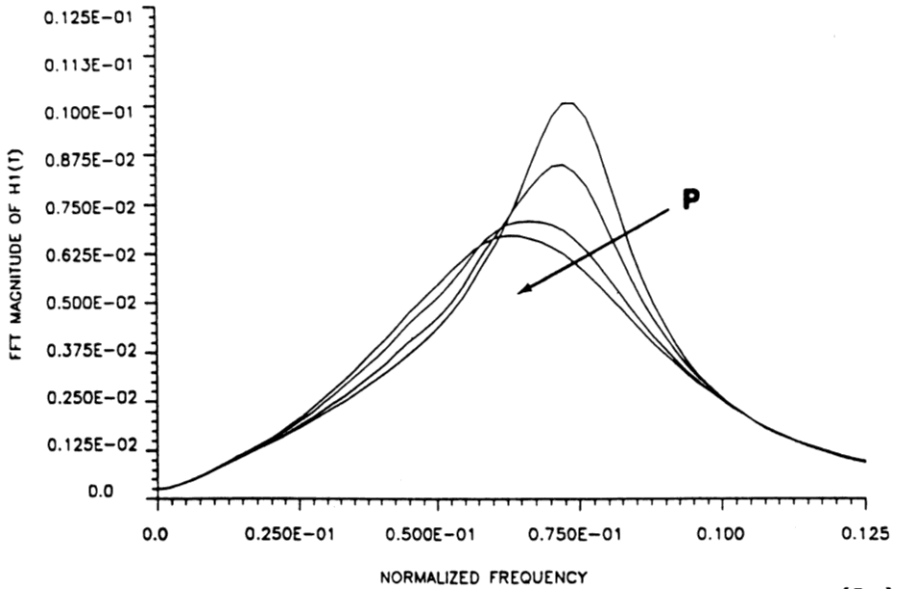
To explore whether nonlinear feedback may constitute a plausible model in this case, we consider a band-pass linear forward subsystem and negative sigmoid feedback, like the one discussed earlier. The obtained first-order Wiener kernel estimates for GWN input power level $P = 1,\ 16,\ 256$, and 4096 are shown in Figure 4.32 (with appropriate plotting offsets to allow easier visual inspection) and replicate the gradual shift in resonance frequency and contraction of the "envelope" of the kernel with increasing P, which were also observed experimentally [Moller, 1975ab,1976ab]. Since these changes are more easily seen in the frequency domain (the preferred domain in auditory studies), we also show the FFT magnitudes of these kernels in Figure 4.32 (right panel). Note that the latter are approximate "inverted tuning curves" exhibiting decreasing resonance frequency and broadening fractional bandwidth as P increases, similar to the experimental observations in auditory nerve fibers.

This nonlinear feedback model appears to capture the essential functional characteristics of primary auditory nerve fibers that have been observed experimentally. The negative compressive feedback can be thought as intensity-reduced stiffness, which has been observed in studies of the transduction properties of cochlear hair cells. Accurate quantitative measures of the functional components and parameters of this feedback system (e.g., the precise form of the feedback nonlinearity) can be obtained on the basis of the analysis presented earlier and will require a series of properly designed experiments for various values of P. Furthermore, the presence of a negative compressive feedback in the auditory fiber response characteristics may provide a plausible explanation for the onset of pathological

conditions such as tinnitus, as a situation where the strength of the negative compressive feedback increases beyond normal values and leads to undiminishing oscillatory behavior irrespective of the specific auditory input waveform, as demonstrated earlier (see Figure 4.25).



a)



b)

Figure 4.32
(a) First-order Wiener kernels of negative sigmoid feedback system ($\in = 1$, $\alpha = 0.25$) with a band-pass forward component, for $P = 1$ (trace 1), 16 (trace 2), 256 (trace 3), and 4096 (trace 4) emulating primary auditory fibers. Observe the contracting envelope and decreasing resonance frequency as P increases. (b) FFT magnitudes of the kernels shown in (a). We observe decreasing resonance frequency and gain as P increases, as well as broadening of the tuning curve in reverse relation to the envelope of the kernel. When these curves are plotted in contrast sensitivity scale (i.e., each scaled by its corresponding P value), then the resonance-frequency gain will appear increasing with increasing P [Marmarelis, 1991].

Concluding Remarks on Nonlinear Feedback

Nonlinear feedback has been long thought to exist in many important physiological systems and be of critical importance for maintaining proper physiological function. However, its systematic and rigorous study has been hindered by the complexity of the subject matter and the inadequacy of practical methods of analysis. The study of Volterra-Wiener expansions of nonlinear differential equations has led to some analytical results that begin to shed light on the analysis of nonlinear feedback systems in a manner that advances our understanding of the system under study. The results obtained for a class of nonlinear feedback systems relate Wiener kernel measurements with the effects of nonlinear feedback under various experimental conditions. Explicit mathematical expressions were derived that relate Wiener kernel measurements to the characteristics of the feedback system and the stimulus parameters. The theoretical results were tested with simulations, and their validity was demonstrated in a variety of cases (cubic and sigmoid feedback with overdamped or underdamped forward subsystem). These test cases were chosen as to suggest possible interpretations of experimental results, including results that have been published in recent years for two types of sensory systems: retinal horizontal and bipolar cells, and primary auditory nerve fibers. It was shown that relatively simple nonlinear feedback models can reproduce the qualitative changes in kernel waveforms observed experimentally in these sensory systems. Precise quantitative determination of the parameters of the feedback models requires analysis (in the presented context) of data from a series of properly designed experiments.

Specifically, it was shown that negative decompressive feedback (e.g., cubic) or positive compressive feedback (e.g., sigmoid) result in gradually decreasing damping (increasing bandwidth) of the first-order Wiener kernel as the GWN input power level and/or mean level increase. Conversely, positive decompressive or negative compressive feedback result in the reverse pattern of changes. The extent of these effects depends, of course, on the exact type of feedback nonlinearity and/or the dynamics of the linear forward subsystem. It was demonstrated through analysis and computer simulations that the experimentally observed changes in the waveform of the first-order Wiener kernel measurements for retinal horizontal and bipolar cells can be explained with the use of negative decompressive (cubic) feedback and low-pass forward subsystems (viz., the gradual transition from an overdamped to an underdamped mode as the GWN stimulus power and/or mean level increase). In the case of auditory nerve fibers, it was shown that the use of negative compressive (sigmoid) feedback and a band-pass forward subsystem can reproduce the effects observed experimentally on their "tuning

curves" for increasing stimulus intensity (viz., a gradual downward shift of the resonance frequency and broadening of the bandwidth of the tuning curve with increasing stimulus power level).

It is hoped that this work will inseminate an interest among systems physiologists to explore the possibility of nonlinear feedback models in order to explain changes in response characteristics when the experimental stimulus conditions vary. This is critical when such changes cannot be explained by simple cascade models of linear and static nonlinear components (like the ones discussed earlier), which are currently popular in efforts to construct equivalent block-structured models from kernel measurements. For instance, in the case of the auditory nerve fibers, the suggested model of negative sigmoid feedback may offer a plausible explanation for pathological states of the auditory system, such as tinnitus. Likewise, in the case of retinal cells, negative decompressive feedback in tandem with compressive nonlinearites may explain the ability of the "front end" of the visual system to accommodate a very broad range of visual stimulus intensities while preserving adequate dynamic range for effective information processing, as well as retain the ability to respond rapidly to changes in stimulus intensity.

4.2. CONNECTIONIST MODELS

The idea behind connectionist models is that the relationships among variables of interest can be represented in the form of connected graphs with generic architectures, so that claims of universal applicability can be supported for certain broad classes of problems. The most celebrated example of this approach has been the class of "artificial neural networks" (ANN) with forward and/or recurrent (feedback) interconnections. The latter types with recurrent interconnections have found certain specialized applications (e.g., Hopfield nets solution to the notorious "traveling salesman problem") and have made, largely unsubstantiated, claims of affinity to biological neural networks; however it is fair to say that they have not lived up to their promise (until now) and their current use in modeling applications is rather limited. The former types of ANN with forward interconnections have found (and continue to find) numerous applications and have demonstrated considerable utility in various fields. These types of ANN can be used for modeling purposes by representing arbitrary input-output mappings, and they derive their scientific pedigree from Hilbert's "13^{th} Problem" and Kolmogorov's "representation theorem" in the early part of the 20^{th} century.

The fundamental mathematical problem concerns the mapping of a multivariate function onto a univariate function by means of a reference set of "activation functions" and interconnection weights.

Kolmogorov's constructive theorem provided theoretical impetus to this effort, but the practical solution of this problem came through the methodological evolution of the concept of a "perceptron" that was proposed by Rosenblatt. The field was further advanced through the pioneering work of Widrow, Grossberg, and the contributions of numerous others, leading to the burgenoning field of feedforward ANN (for review see [Haykin, 1994;Hassoun, 1995; Rumellhard & McCleland, 1987]).

The adjective "neural" is used for historical reasons, since some of the pioneering work alluded to similarities with information processing in the central nervous system--a point that remains conjectural and largely wishful rather than corroborated by real data in a convincing manner (yet). Nonetheless, the mere allusion to analogies with natural "thinking processes" seems to magnetize people's attention and to lower the threshold of initial acceptance of related ideas. Although this practice was proven to offer promotional advantages, the tenuous connection with reality and certain distaste for promotional hype has led us to dispense with this adjective in the sequel and refer to this type of connectionist model as "*Volterra-equivalent networks*" (VEN).

For our purposes, the problem of nonlinear system modeling from input-output data relates intimately with the problem of mapping multivariate functions onto a univariate function, when the input-output data are discretized (sampled). Since the latter is always the case in practice, we have explored the use of Volterra-equivalent network architectures as an alternative approach to achieve nonlinear system modeling in a practical context. We have found that certain architectures offer practical advantages in some cases, as discussed below.

4.2.1. Equivalence Between Connectionist and Volterra Models

We start by exploring the conditions for equivalence between feedforward connectionist models and discrete Volterra models [Marmarelis, 1993; Marmarelis & Zhao, 1994,1997]. The latter generally represent a mapping of the input epoch vector: $\mathbf{x}(n) = \left[x(n), x(n-1), ..., x(n-M+1) \right]'$ onto the output present scalar value $y(n)$, where M is the memory-bandwidth product of the system. This mapping of the $[M \times 1]$ input epoch vector onto the output scalar can be expressed in terms of the discrete Volterra series expansion of Equation (2.32). On the other hand, this mapping can be implemented by means of a feedforward network architecture that receives as input the input epoch vector and generates as output the scalar value $y(n)$. The general architecture of a "Volterra-equivalent network" (VEN) is shown in Figure 4.33 and employs an input layer of M units (introducing the input epoch values into the

network) and two hidden-layers of units (that apply nonlinear transformations on weighted sums of the input values). The VEN output is formed by the sum of the outputs of the hidden units of the second-layer and an offset value.

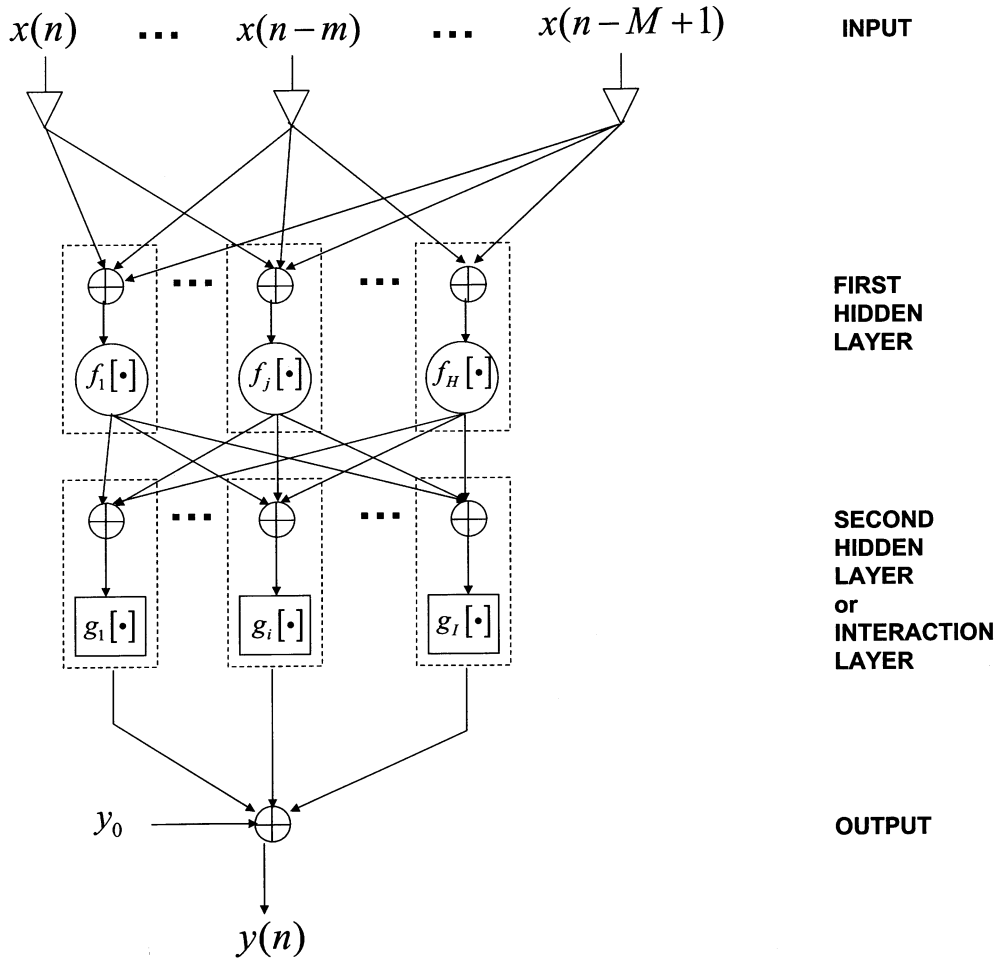


Figure 4.33

The general architecture of a "Volterra-equivalent network" (VEN) receiving the input epoch and generating the corresponding system output after processing through two hidden layers. Each hidden unit performs a polynomial transformation weighted sum of its inputs. Arbitrary activation functions can be used and be approximated by polynomial expressions within the range of interest. The output unit performs a simple summation of the outputs of the second hidden layer (also called "interaction layer") and an output offset.

A more restricted but practically useful VEN class follows the general architecture of a "three-layer perceptron" (TLP) with a "tapped-delay" input and a single hidden layer (shown in Figure 4.34) that utilizes polynomial activation functions instead of the conventional sigmoidal activation functions employed by TLP and other ANN. The output may have its own nonlinearity (e.g., a hard-threshold in the initial perceptron architecture that generated binary outputs, consistent with the all-or-none data modality of action potentials in the nervous system).

The TLP architecture shown in Figure 4.34 corresponds to the class of "separable Volterra networks" (SVN) whose basic operations are described below [Marmarelis & Zhao, 1997]. The weights $\{w_{j,m}\}$ are used to form the input $u_j(n)$ into the nonlinear "activation function" f_j of the j th hidden unit as:

$$u_j(n) = \sum_{m=0}^{M-1} w_{j,m} x(n-m) \tag{4.98}$$

leading to the output of the j th hidden unit:

$$z_j(n) = f_j\left[u_j(n)\right] \tag{4.99}$$

where the activation function f_j is a static nonlinear function. For SVN or VEN models, the activation functions are chosen to be polynomials:

$$f_j(u_j) = \sum_{q=1}^{Q} c_{j,q} u_j^q \tag{4.100}$$

although any analytic function or non-analytic function approximated by polynomials (or power series) can be used.

The SVN/VEN output unit is a simple adder (i.e., no output weights are necessary) that sums the outputs $\{z_j\}$ of the hidden units and an offset y_0 as:

$$y(n) = y_0 + \sum_{j=1}^{H} z_j(n) \tag{4.101}$$

Combining Equations (4.98), (4.99), (4.100) and (4.101), we obtain the SVN/VEN input-output relation:

$$y(n) = y_0 + \sum_{q=1}^{Q}\sum_{j=1}^{H} c_{j,q} \sum_{m_1=0}^{M-1} \cdots \sum_{m_q=0}^{M-1} w_{j,m_1} \cdots w_{j,m_q} x(n-m_1) \cdots x(n-m_q) \tag{4.102}$$

which is isomorphic to the discrete Volterra model (DVM) of order Q as H tends to infinity. Equation (4.102) demonstrates the equivalence between SVN/VEN and DVM, which is expected to hold in practice for infinite H. It has been found empirically that satisfactory DVM approximations can be obtained with small H for many physiological systems.

It is evident that the discrete Volterra kernels can be estimated by means of the SVN/VEN parameters as:

$$k_q(m_1, \ldots, m_q) = \sum_{j=1}^{H} c_{j,q} w_{j,m_1} \cdots w_{j,m_q} \tag{4.103}$$

offering an alternative for Volterra kernel estimation through SVN/VEN training that has proven to have certain practical advantages [Marmarelis & Zhao, 1997]. Naturally, the same equivalence holds for the broader class of VEN models shown in Figure 4.33 as H and I tend to infinity. However the Volterra kernel expressions become more complicated for the general form of the VEN model in Figure 4.33.

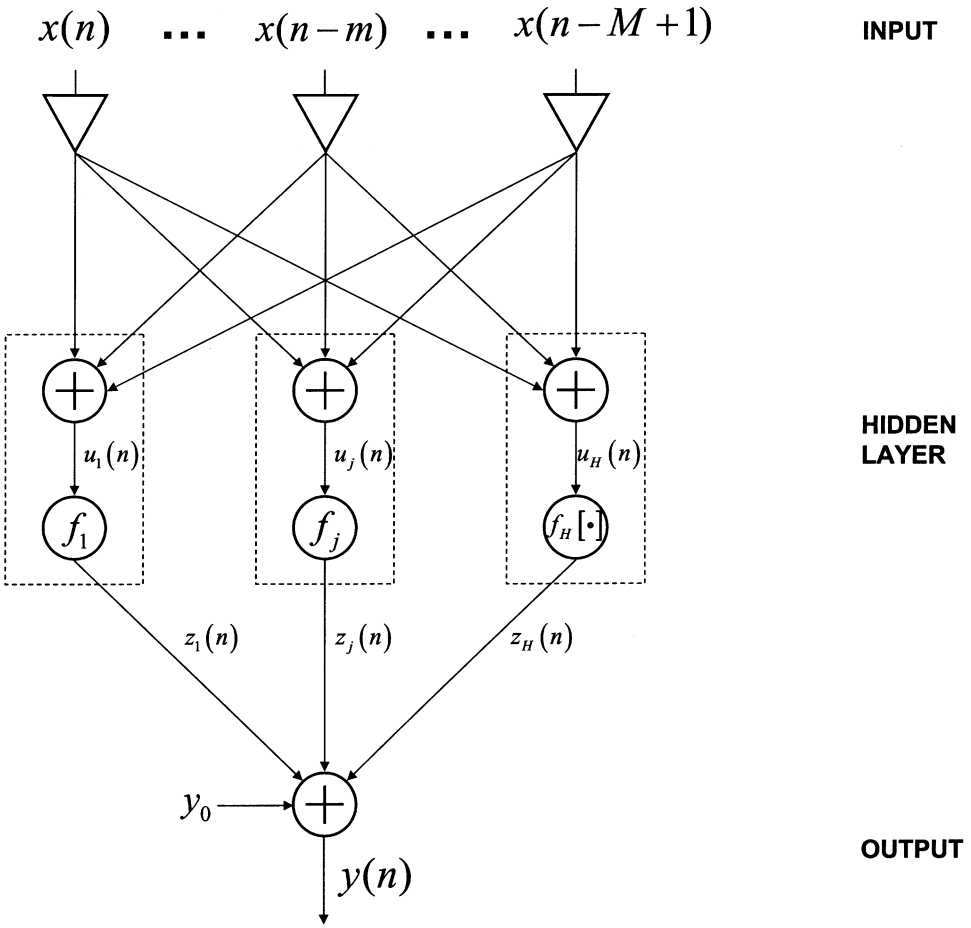


Figure 4.34
The single-layer architecture of the special class of VEN that corresponds to the "separable Volterra network" (SVN). The output unit is a simple adder. This network configuration is similar to the traditional "three-layer perceptron" (TLP) albeit with *polynomial* activation functions $\{f_i\}$ in the hidden units, instead of the conventional sigmoidal activation functions used in TLP.

The use of polynomial activation functions in connection with SVN/VEN models directly maintains the mathematical affinity with the Volterra models [Marmarelis, 1994,1997], and typically reduces the required number H of hidden units (relative to perceptron-type models). This is demonstrated below by comparing the two classes of models.

The activation functions used for conventional TLP architectures are selected to have the sigmoidal shape of the "logistic" function:

$$S_j\left(u_j\right)=\frac{1}{1+\exp\left[-\lambda\left(u_j-\theta_j\right)\right]} \tag{4.104}$$

or the "hyperbolic tangent" function:

$$S_j\left(u_j\right)=\frac{1-\exp\left[-\lambda\left(u_j-\theta_j\right)\right]}{1+\exp\left[-\lambda\left(u_j-\theta_j\right)\right]} \tag{4.105}$$

depending on whether we want a unipolar (between 0 and 1) or a bipolar (between -1 and $+1$) output, as u_j tends to $\pm\infty$. The parameter λ defines the slope of this sigmoidal curve at the inflection point $u_j=\theta_j$ and is typically specified by the user in the conventional TLP architectures (i.e., it is not estimated from the data). However, the offset parameter θ_j is estimated from the data for each hidden unit separately, during the "training" process of the TLP. Since the specific value of λ can affect the stability and the convergence rate of the training algorithm (e.g., through error back-propagation discussed in Section 4.2.2), we recommend that λ be trained along with the other network parameters (contrary to established practice). As λ increases, the sigmoidal function approaches a "hard threshold" at θ_j and can be used in connection with binary output variables. In the latter case, it is necessary to include a hard-threshold θ_0 in the output unit, as:

$$y\left(n\right)=T_0\left[\sum_{j=1}^{H} r_j z_j\left(n\right)\right] \tag{4.106}$$

where $\left\{r_j\right\}$ are the output weights and T_0 denotes a hard-threshold operator (i.e., equal to 1 when its argument is greater than a threshold value θ_0, and 0 otherwise).

In order to simplify the comparison and study of equivalence between the SVN/VEN and the TLP class of models, we consider a linear TLP output unit:

$$y\left(n\right)=\sum_{j=1}^{H} r_j z_j\left(n\right) \tag{4.107}$$

Then, combining Equations (4.98) and (4.107), we obtain the input-output relation of the TLP model:

$$y\left(n\right)=\sum_{j=1}^{H} r_j S_j\left[\sum_{m=0}^{M-1} w_{j,m} x\left(n-m\right)\right] \tag{4.108}$$

which can be put in a DVM form by representing each sigmoidal activation function S_j with its respective Taylor expansion:

$$S_j\left(u_j\right)=\sum_{i=0}^{\infty}\alpha_i\left(\theta_j\right)u_j^i\left(n\right) \tag{4.109}$$

where the Taylor expansion coefficients depend on the offset parameter θ_j of the sigmoidal activation function (and on the slope parameter λ, if it is allowed to be trained). If the selected activation function is not analytic (e.g., a hard-threshold function that does not have a proper Taylor expansion), then a polynomial approximation of arbitrary accuracy can be obtained according to the Weierstrass theorem, following the method presented in Appendix I.

Combining Equations (4.108) and (4.109), we obtain the equivalent DVM:

$$y\left(n\right)=\sum_{i=0}^{\infty}\sum_{j=1}^{H}r_j\alpha_i\left(\theta_j\right)\sum_{m_1=0}^{M-1}...\sum_{m_i=0}^{M-1}w_{j,m_1}...w_{j,m_i}x\left(n-m_1\right)...x\left(n-m_i\right) \tag{4.110}$$

for this class of TLP models, where the i th-order discrete Volterra kernel is given by:

$$k_i\left(m_1,...,m_i\right)=\sum_{j=1}^{H}r_j\alpha_i\left(\theta_j\right)w_{j,m_1}...w_{j,m_i} \tag{4.111}$$

Therefore, an SVN/VEN or a TLP model has an equivalent DVM whose kernels are defined by Equation (4.103) or (4.111) respectively. The possible presence of an activation function at the output unit (e.g., a hard threshold) does not alter this fundamental fact but it makes the analytical expressions for the equivalent Volterra kernels more complicated. The fundamental question remains as to the relative efficiency of an equivalent SVN/VEN or TLP representation of a given DVM. This question can be answered by considering the total number of free parameters for each model type that yields the same approximation of the kernel values for a Q th-order DVM. It has been found that the TLP model requires generally a much larger number of hidden units and therefore more free parameters.

This important point can be elucidated geometrically by introducing a hard threshold at the output of both models and let $\lambda\to\infty$ in the sigmoidal activation functions of the TLP so that each S_j becomes a hard-threshold operator $T_j\left(u_j-\theta_j\right)$. Furthermore, to facilitate the demonstration, consider the simple case of $Q=2$ and $M=2$, where the input-output relation before the application of the output threshold is simply given by:

$$y\left(n\right)=x^2\left(n\right)+x^2\left(n-1\right) \tag{4.112}$$

Application of a hard threshold $\theta_0=1$ at the output of this DVM yields a circular binary boundary defining the input-output relation, as shown in Figure 4.35. To approximate this input-output relation

with a TLP architecture, we need to use a very large number H of hidden units, since each hidden unit yields a rectilinear segment after network training for the best mean-square approximation of the output, according to the binary output approximation:

$$\sum_{j=1}^{H} r_j T_j\left(u_j - \theta_j\right) \cong \theta_0 \tag{4.113}$$

where $T_j\left(u_j - \theta_j\right) = 1$ when $w_{j,0}x(n) + w_{j,1}x(n-1) \geq \theta_j$, otherwise T_j is zero. Thus, the linear equation:

$$w_{j,0}x(n) + w_{j,1}x(n-1) = \theta_j \tag{4.114}$$

defines the rectilinear segment of the output approximation due to the j th hidden unit, and the resulting TLP polygonal approximation is defined by the output Equation (4.113)). For instance, if H is only 3, then the TLP triangular approximation is shown in Figure 4.35 with dotted line. If the training of the TLP is perfect, then the shaded area is minimized in a mean-square sense and the resulting polygonal approximation is canonical (i.e., symmetric for a symmetric boundary). This, of course, is seldom the case in practice because the TLP training is imperfect due to noise in the data or incomplete training convergence, and an approximation similar to the one depicted in Figure 4.34 typically emerges. Naturally, as H increases, the polygonal TLP approximation of the circle improves, reaching asymptotically a perfect representation when $H \to \infty$ and if the training data set is noise-free and fully representative of the actual input ensemble of the system. Note however that a *perfect SVN/VEN* representation of this input-output relation (for noise-free data) requires *only* two hidden units with quadratic activation functions!

This simple illustrative example punctuates a very important point that forms the foundation of understanding the relative efficacy of SVN/VEN and TLP models in representing nonlinear dynamic input-output relationships/mappings. The key point is that the use of sigmoidal activation functions unduly constrains the ability of the network to represent a broad variety of input-output mappings with a small number of hidden units. This is true even when multiple hidden layers are used, because the aforementioned rectilinear constraint remains in force. On the other hand, if polynomial activation functions are used, then a much broader variety of input-output mappings can be represented by a small number of hidden units.

This potential parsimony in the complexity of the required network architecture (in terms of the number of hidden units) is of critical importance in practice, because the complexity of the modeling task is directly related to the number of hidden units (both in terms of estimation and interpretation). This holds true for single or multiple hidden layers. However, the use of polynomial activation

functions may give rise to some additional problems in the network training process by introducing more local minima during minimization of the cost function. It is also contrary to Kolmogorov's constructionist approach in his representation theorem (requiring monotonic activation functions) that retains a degree of reverence within the peer community. With all due respect to Kolmogorov's seminal contributions, it is claimed herein that non-monotonic activation functions (such as polynomials) offer, on balance, a more efficient approach to the problem of modeling arbitrary input-output mappings with feedforward network architectures.

This proposition gives rise to a new class of feedforward Volterra-equivalent network architectures that employ polynomial (or generally non-monotomic) activation functions and can be efficient models of nonlinear input-output mappings. Note that this is consistent with Gabor's proposition of a "universal" input-output model, akin in form to a discrete Volterra model [Eykhoff, 1974].

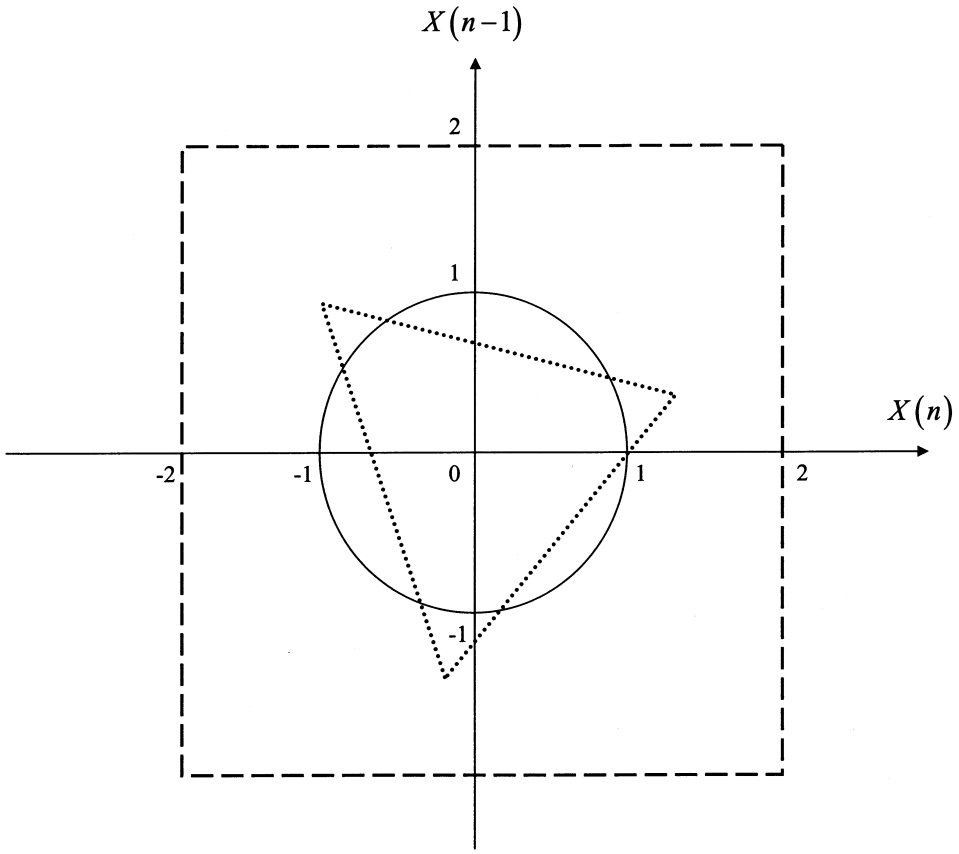


Figure 4.35
Illustrative example of a circular output "trigger boundary" (solid line) being approximated by a three-layer perceptron (TLP) with three hidden units defining the piecewise rectilinear (triangular) approximation of the "trigger boundary" marked by the dotted lines. The training set is generated by 500 datapoints of uniform white noise input that defines the square domain demarcated by dashed lines. The piecewise rectilinear approximation improves with increasing number of hidden units of the TLP, assuming polygonal form and approaching asymptotically more precise representations of the circular boundary. Nonetheless, a VEN with two hidden units having quadratic activation functions yields a precise and parsimonious representation of the circular boundary.

Relation with PDM Modeling

If a feedforward network architecture with polynomial activation functions is shown to be appropriate for a certain system, then the number of hidden units in the first hidden layer defines the number of PDMs of this system. This is easily proven for the single hidden-layer SVN/VEN models shown in Figure 4.34 by considering as the j th PDM output the "internal variable" $u_j(n)$ of the j th hidden unit given by the convolution of Equation (4.98), where the j th PDM $p_j(m)$ is defined by the respective weights $\{w_{j,m}\}$ that form a discrete "impulse response function". This internal variable $u_j(n)$ is subsequently transformed by the polynomial activation function $f_j(u_j)$ to generate the output of the j th hidden unit $z_j(n)$ according to Equation (4.100), where the polynomial coefficients $\{c_{j,q}\}$ are estimated from the data during SVN/VEN training.

It is evident that the equivalent Volterra kernels of the SVN/VEN model are given by:

$$k_q(m_1,...,m_q) = \sum_{j=1}^{H} c_{j,q}\, p_j(m_1)...p_j(m_q) \tag{4.115}$$

Therefore, this network model corresponds to the case of a "separable PDM model" where the static nonlinearity associated with the PDMs can be "separated" into individual polynomial nonlinearities corresponding to each PDM and defined by the respective activation functions, as shown in Figure 4.36. This separable PDM model can be viewed as a special case of the general PDM model of Figure 4.1 and corresponds to "separable Volterra network" (SVN) architecture. The PDMs capture the system dynamics in a most efficient manner (although other filterbanks can be also used) and the nonlinearities may or may not be separable. In the latter case, the general VEN model of Figure 4.33 must be used to represent the general PDM model of Figure 4.1.

The SVN architecture is obviously very convenient in practice but cannot be proposed as having general applicability. Even though it has been found to be appropriate for many actual applications to date (see Chapter 6), the general VEN model would require multiple hidden layers that can represent static nonlinearities of arbitrary complexity in the PDM model. The additional hidden layers may incorporate other analytic functions (such as sigmoidal, Gaussian, etc.), although the polynomial functions would yield directly the familiar multinomial form of the modified Volterra model with cross-terms.

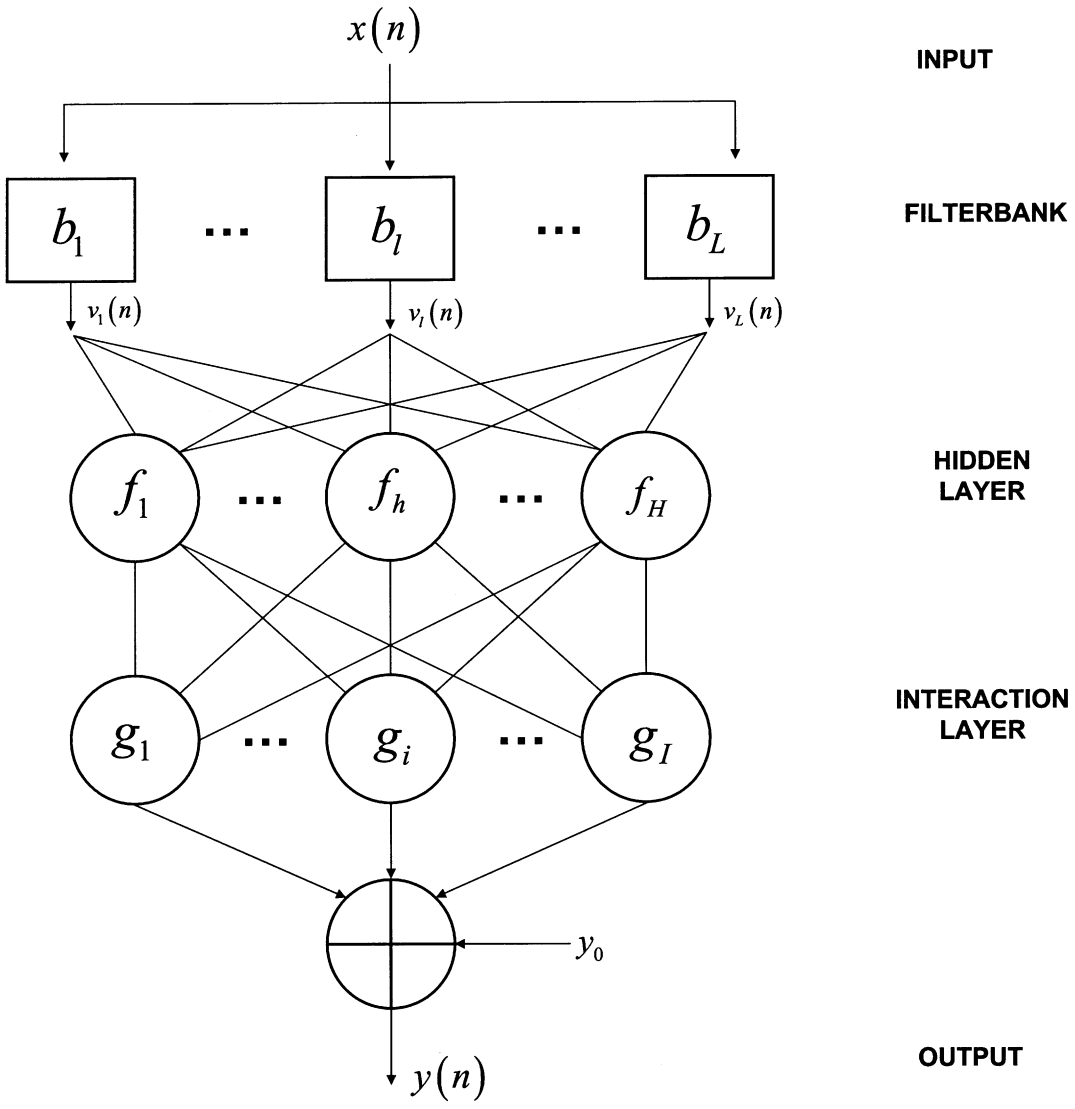


Figure 4.36
The VEN model architecture for input pre-processing by the filterbank $\{b_l\}$.

The relation of PDM models with the VEN architectures also points to the relation with the modified discrete Volterra (MDV) models that employ kernel expansions on selected bases. In this case, the input can be viewed as being pre-processed through the respective filterbank prior to weighting and processing by the hidden layer, resulting in the architecture of Figure 4.36. The only difference with the previous case of the VEN model shown in Figure 4.33 is that the internal variables of the first hidden layer are now weighted sums of the filterbank outputs $\{v_i(n)\}$:

$$u_j(n) = \sum_{l=1}^{L} w_{j,l} v_l(n) \tag{4.116}$$

instead of the weighted sum of the input lags shown in Equation (4.98). The critical point is that when $L \square M$, VEN model parsimony results. By definition, the use of the PDMs in the filterbank yields the most parsimonious VEN model (minimum L).

In order to establish a clear terminology, we will use the term SVN for the VEN model of Figure 4.34 with a single hidden layer and polynomial activation functions, and the term TLP when the activation functions are sigmoidal. Note that the VEN model can generally have multiple hidden layers to represent arbitrary nonlinearities (not necessarily separable), as shown in Figure 4.33.

The important practical issue of how we determine the appropriate memory-bandwidth product M and degree of polynomial nonlinearity Q in the activation functions of the SVN model is addressed by preliminary experiments discussed in Section 5.2. For instance, the degree of polynomial nonlinearity can be established by preliminary testing of the system with sinusoidal inputs and subsequent determination of the highest harmonic in the output via discrete Fourier transform, or by varying the power level of a white-noise input and fitting the resulting output variance to a polynomial expression. On another critical issue, the number of hidden units can be determined by successive trials in ascending order and application of the statistical criterion of Section 2.3.1 on the resulting reduction of residual variance. Note that the input weights in the SVN model are normalized to unity Euclidean norm for each hidden unit so that the polynomial coefficients give a direct measure of the relative importance of each hidden unit.

Illustrative examples are given below for a second-order and an infinite-order simulated system with two PDMs [Marmarelis, 1997; Marmarelis & Zhao, 1997].

Illustrative Examples

First we consider a second-order Volterra system with memory-bandwidth product $M = 25$, having the first-order kernel shown in Figure 4.37 with solid line and the second-order kernel shown in the top panel of Figure 4.38. This system is simulated using a uniform white-noise input of 500 datapoints. We estimate the first-order and second-order Volterra kernels of this system using TLP and SVN models, as well as LET which was introduced in Section 2.3.2 to improve Volterra kernel estimation by use of Laguerre expansions of the kernels and least-squares estimation of the expansion coefficients [Marmarelis, 1993]. In the noise-free case, the LET and SVN approaches yield precise first-order and second-order Volterra kernel estimates, although at considerably different computational cost (LET is

about 20 times faster than SVN in this case). Note that LET approach requires five discrete Laguerre functions (DLFs) in this example (i.e., 21 free parameters need be estimated) while the SVN approach needs only one hidden unit with a second-degree activation function (resulting in 28 free parameters).

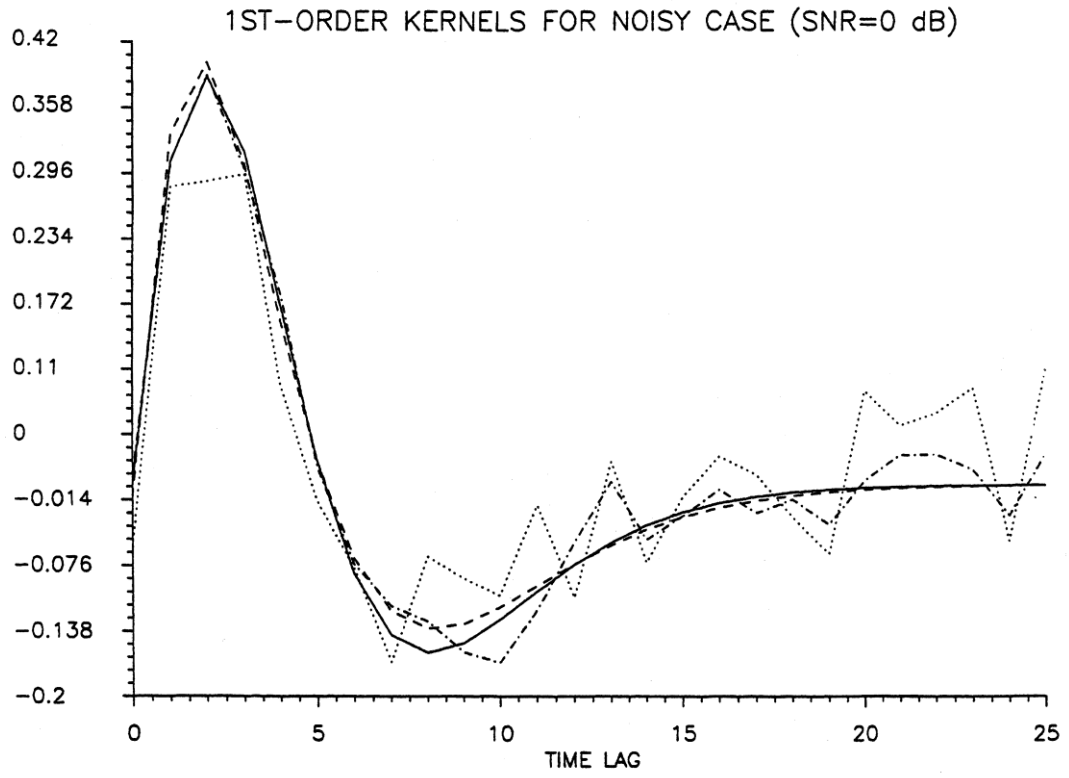


Figure 4.37
The exact first-order kernel (solid line) and the three estimates obtained in the noisy case (SNR=0 dB) via LET (dashed line), SVN (dot-dashed line), and TLP (dotted line). The LET estimate is the best in this example, followed closely by the SVN estimate in terms of accuracy. The TLP estimate (obtained with four hidden units) is the worst in accuracy and computationally most demanding [Marmarelis & Zhao, 1997].

As expected, the TLP model requires more free parameters in this example, (i.e., more hidden units) and its predictive accuracy is rather inferior, although it incrementally improves with increasing number H of hidden units. This incremental improvement gradually diminishes, because of the finite data record. Since the computational burden for network training increases with increasing H, we are faced with an important tradeoff: incremental improvement in accuracy versus additional computational burden. By varying H, we determine a reasonable compromise for a TLP model with four hidden units, where the number of free parameters is 112 and the required training time is about 20 times longer than SVN (or 400 times longer than LET). The resulting TLP kernel estimates are not as accurate as their SVN or LET counterparts, as illustrated in Figures 4.37 and 4.38 for the first-order and second-order

kernels, respectively, for a signal-to-noise ratio of 0 dB in the output data (i.e., the output-additive independent GWN variance is equal to the noise-free de-meaned output mean-square value). Note that the SVN training required 200 iterations in this example versus 2000 iterations required for TLP training. Thus, SVN appears preferable to TLP in terms of accuracy and computational effort in this example of a second-order Volterra system.

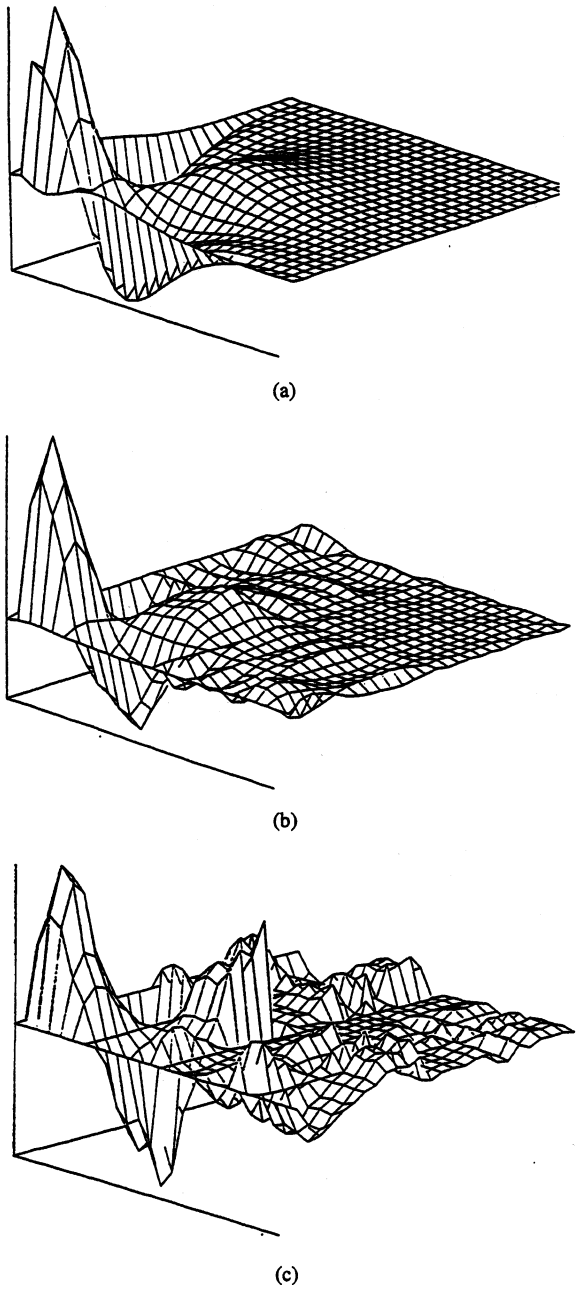


Figure 4.38
The second-order Volterra kernel estimates obtained in the noisy case (SNR=0 dB) via LET (a), SVN (b), TLP (c). The relative performance of the three methods is the same as described in the case of first-order kernels (see caption of Figure 4.37) [Marmarelis & Zhao, 1997].

The obtained Volterra kernel estimates via the three methods (LET, SVN, TLP) demonstrate that the LET estimates are the most accurate and quickest to obtain, followed by the SVN estimates in terms of accuracy and computation—although SVN requires longer computing time (by a factor of 20). The TLP estimates are clearly inferior to either LET or SVN estimates in this example and require longer computing time (about 20 times longer than SVN for $H=4$). These results demonstrate the considerable benefits of using SVN configurations instead of TLP for Volterra system modeling purposes, although there may be some cases where the TLP configuration has a natural advantage (e.g., systems with sigmoidal output nonlinearities).

Although LET appears to yield the best kernel estimates, its application is practically limited to low-order kernels (up to third) and, therefore, it is the preferred method only for systems with low-order nonlinearites. On the other hand, SVN offers not only an attractive alternative for low-order kernel estimation and modeling, but also a *unique practical solution when the system nonlinearities are of high order*. The latter constitutes the primary motivation for introducing the SVN configuration for nonlinear system modeling.

To demonstrate the efficacy of SVN modeling for high-order systems, we consider an infinite-order nonlinear system described by the output equation:

$$y=\left(v_1+0.8v_2^2-0.6v_1^2v_2\right)\sin\left[\left(v_1+v_2\right)/5\right] \tag{4.117}$$

where the sine function can be expressed as a Taylor series expansion, and the "internal" variables (v_1, v_2) are given by the difference equations:

$$v_1(n)=1.2v_1(n-1)-0.6v_1(n-2)+0.5x(n-1) \tag{4.118}$$

$$\begin{aligned} v_2(n)=&\,1.8v_2(n-1)-1.1v_2(n-2)+0.2x(n-3)\\ &+0.1x(n-1)+0.1x(n-2) \end{aligned} \tag{4.119}$$

The discrete-time input signal $x(n)$ is chosen in this simulation to be a 1024-point segment of GWN with unit variance. Use of LET with six DLFs to estimate the truncated second-order and third-order Volterra models yields output predictions with normalized mean-square errors (NMSE) of 47.2% and 34.7%, respectively. Note that the obtained kernel estimates are seriously biased because of the presence of higher order terms in the output equation that are treated by LET as correlated residuals in least-squares estimation. Use of the SVN approach (employing five hidden units of seventh-degree) yields a model of improved prediction accuracy (NMSE=6.1%) and mitigates the problem of kernel estimation bias by allowing estimation of nonlinear terms up to *seventh-order*. Note that, although the

selected system is of infinite order, the higher order Volterra kernels are of gradually diminishing size consistent with the Taylor series expansion of the sine function. Training of a TLP model with these data yields less prediction accuracy than the SVN model for comparable numbers of hidden units. For instance, a TLP model with $H = 8$ yields an output prediction NMSE of 10.3% (an error that can be gradually, but slowly, reduced by increasing the number of hidden units) corresponding to 216 free parameters, which compares with 156 free parameters for the aforementioned SVN model with $H = 5$ and $Q = 7$ that yields an output prediction NMSE of 6.1%.

4.2.2. Volterra-Equivalent Network Architectures for Nonlinear System Modeling

This section discusses the basic principles and methods that govern the use of Volterra-equivalent network (VEN) architectures for nonlinear system modeling. The previously established Volterra modeling framework will remain the mathematical foundation for evaluating the performance of alternative network architectures. The key principles that will be followed are:

(1) the network architecture must retain equivalence to the Volterra class of models;

(2) generality and parsimony will be sought, so that the model is compact but not unduly constrained.

The study will be limited here to feedforward network architectures of single-input/single-output models for which broadband time-series data are available. The case of multiple inputs and outputs, as well as the case of auto-regressive models with recurrent network connections, will be discussed in Chapters 7 and 10 respectively. The underlying physiological system is assumed to be stationary and belong to the Volterra class. The nonstationary case will be discussed in Chapter 9.

The network architectures considered herein may have multiple hidden layers and arbitrary activation function (as long as the latter can be expressed or approximated by polynomials or Taylor series expansions in order to maintain equivalence with the Volterra class of models). For simplicity of representation, all considered networks will have a filterbank for input preprocessing, which can be replaced by the basis of sampling functions if no preprocessing is desired. In general, the selection of the filterbank will be assumed "judicious" in order to yield compact representations of the system kernels (see Section 4.1.1). Although the filterbank may incorporate trainable parameters (see next section on the Laguerre-Volterra network), this will not be a consideration here. With these stipulations in mind, we consider the general VEN architecture shown in Figure 4.36 that has two hidden layers, $\{f_j\}$ and $\{g_i\}$, and a filterbank $\{b_l\}$ for input convolutional preprocessing according to:

$$v_l(n) = \sum_{m=0}^{M-1} b_l(m)\, x(n-m) \tag{4.120}$$

The first hidden layer has H units with activation functions $\{f_j\}$ transforming the internal variables:

$$u_j(n) = \sum_{l=1}^{L} w_{j,l} v_l(n) \tag{4.121}$$

into the hidden unit output:

$$\begin{aligned} z_j(n) &= f_j\left[u_j(n)\right] \\ &= \sum_{q=1}^{Q} c_{j,q} u_j^q(n) \end{aligned} \tag{4.122}$$

The outputs $\{z_j(n)\}$ of the first hidden layer are the inputs to the second hidden layer (also termed the "interaction layer") that has I units with activation functions $\{g_i\}$ transforming the ith internal variable:

$$\phi_i(n) = \sum_{j=1}^{H} \rho_{i,j} z_j(n) \tag{4.123}$$

into the ith interaction unit output:

$$\begin{aligned} \psi_i(n) &= g_i\left[\phi_i(n)\right] \\ &= \sum_{r=1}^{R} \gamma_{i,r} \phi_j^r(n) \end{aligned} \tag{4.124}$$

Note that R and/or Q may tend to infinity if the activation function is expressed as a Taylor series expansion. Therefore, activation functions other than polynomials (e.g., sigmoidal, exponential, sinusoidal) are admissible under this network architecture (note that monotonicity is not a requirement in contrast to the conventional approach). For instance, a sensible choice might involve polynomial activation functions in the first hidden layer (for the reason expounded in the previous section), but cascaded with sigmoidal activation functions to secure stability of the model output, as discussed in Section 4.4.

It is evident that the presence of a second hidden layer distinguishes this architecture from the separable Volterra networks (SVN) discussed in the previous section and endows them with broader applicability. However, as previously for the SVN model, the "principal dynamic modes" (PDMs) corresponding to this VEN model architecture remain the equivalent filters generating the internal variables $u_j(n)$ of the first hidden layer, i.e., the jth PDM is:

$$p_j(m) = \sum_{l=1}^{L} w_{j,l} b_l(m) \tag{4.125}$$

and the PDM outputs:

$$u_j(n) = \sum_{m=0}^{M-1} p_j(m) x(n-m) \tag{4.126}$$

are fed into a multi-input static nonlinearity that maps the H PDM outputs $\{z_1(n),...,z_H(n)\}$ onto the VEN output $y(n)$ after transformation through the interaction layer. Thus, the non-separable nonlinearity of the equivalent PDM model is represented by the cascaded operations of the hidden and interaction layers, yielding the input-output relation:

$$y(n) = y_0 + \sum_{i=1}^{I} g_i \left\{ \sum_{j=1}^{H} \rho_{i,j} f_j \left[\sum_{l=1}^{L} w_{j,l} \sum_{m=0}^{M-1} b_l(m) x(n-m) \right] \right\} \tag{4.127}$$

which provides guidance for the application of the chain rule of differentiation for network training through the error back-propagation method (see below). When the quadratic cost function:

$$J(n) = \frac{1}{2} \in^2 (n) \tag{4.128}$$

is sought to be minimized for all n in the training set of data, where:

$$\in(n) = y(n) - y(n) \tag{4.129}$$

is the output prediction error (y denotes the output measurements), we need to evaluate the gradient of the cost function with respect to all network parameters. The network parameter space which is composed of the weights $\{w_{j,l}\}$ and $\{\rho_{i,j}\}$, as well as the parameters of the activation functions $\{f_j\}$ and $\{g_i\}$. For instance, the gradient component with respect to the weight $w_{k,s}$ is:

$$\frac{\partial J(n)}{\partial w_{k,s}} = \in(n) \frac{\partial \in(n)}{\partial w_{k,s}} \tag{4.130}$$

by application of the chain rule of differentiation we have:

$$\frac{\partial \in(n)}{\partial w_{k,s}} = \frac{\partial y(n)}{\partial w_{k,s}} = \sum_{i=1}^{I} g_i' \{\phi_i(n)\} \frac{\partial \phi_i(n)}{\partial w_{k,s}}$$

$$= \sum_{i=1}^{I} g_i' \{\phi_i(n)\} \sum_{j=1}^{H} \rho_{i,j} f_j' [u_j(n)] \frac{\partial u_j(n)}{\partial w_{k,s}}$$

$$= \sum_{i=1}^{I} g_i'\left\{\phi_i(n)\right\} \rho_{i,k} f_k'\left[u_k(n)\right] v_s(n) \tag{4.131}$$

where f' and g' denote the derivatives of f and g respectively. These gradient components are evaluated, of course, for the current parameter values that are continuously updated through the training procedure. For instance, the value of the weight $w_{k,s}$ is updated at the i th iteration as:

$$w_{k,s}^{(i+1)} = w_{k,s}^{(i)} - \gamma_w \left[\frac{\partial \in (n)}{\partial w_{k,s}}\right]^{(i)} \in^{(i)}(n) \tag{4.132}$$

where the gradient component is given by Equation (4.131), γ_w denotes the "training step" or "learning constant" for the weights $\left\{w_{j,l}\right\}$, and the superscript (i) denotes quantities evaluated for the i th-iteration parameter values. The update schemes that are based on local gradient information usually employ a "momentum" term that reduces the random variability from iteration to iteration by performing first-order low-pass filtering (exponentially weighted smoothing).

Analogous expressions can be developed for the other network parameters using the chain rule of differentiation. A specific example is given in the following section for the Laguerre-Volterra network that has been used extensively in actual applications to date. In this section, we will concentrate on three key issues:

(1) equivalence with Volterra kernels/models;
(2) selection of the structural parameters of the network model;
(3) convergence and accuracy of the training procedure.

Note that the training of the network is based on the training dataset (using either single error/residual points or summing many squared residuals in batch form), however the cost function computation is based on the testing dataset (different residuals than the training dataset) that has been randomly selected according to the method described earlier in order to reduce possible correlations among the residuals. Note also that, if the actual prediction errors are not Gaussian, then a non-quadratic cost function can be used to attain efficient estimates of the network (i.e., minimum estimation variance). The appropriate cost function in this case is determined by the minus log-likelihood function of the actual prediction errors, as described in Section 2.1.5.

Equivalence with Volterra Kernels/Models

The input-output relation of the VEN model shown in Figure 4.36 is given by Equation (4.127). The equivalent Volterra model, when the activation functions are expressed either as polynomials or as Taylor series expansions, yields the Volterra kernel expressions:

$$k_0 = y_0 \tag{4.133}$$

$$k_1(m) = \sum_{i=1}^{I} \gamma_{i,1} \sum_{j=1}^{H} \rho_{i,j} c_{j,1} \sum_{l=1}^{L} w_{j,l} b_l(m) \tag{4.134}$$

$$\begin{aligned} k_2(m_1, m_2) = & \sum_{i=1}^{I} \gamma_{i,1} \sum_{j=1}^{H} \rho_{i,j} c_{j,2} \sum_{l_1=1}^{L} \sum_{l_2=1}^{L} w_{j,l_1} w_{j,l_2} b_{l_1}(m_1) b_{l_2}(m_2) \\ & + \sum_{i=1}^{I} \gamma_{i,2} \sum_{j_1=1}^{H} \sum_{j_2=1}^{H} \rho_{i,j_1} \rho_{i,j_2} c_{j_1,1} c_{j_2,1} \sum_{l_1=1}^{L} \sum_{l_2=1}^{L} w_{j_1,l_1} w_{j_2,l_2} b_{l_1}(m_1) b_{l_2}(m_2) \end{aligned} \tag{4.135}$$

The expressions for the higher order kernels grow more complicated but are not needed in practice, since the interpretation of high-order nonlinearities will rely on the PDM model form and not on individual kernels. It is evident that the complete representation of the general Volterra system will require infinite number of hidden units and filterbank basis functions. However, we posit that, for most physiological systems, finite numbers of L, H and I will provide satisfactory model approximations. The same is posited for the order of nonlinearity which is determined by the product (QR) in the network of Figure 4.36.

Selection of the Structural Parameters of the VEN Model

The selection of the structural parameters (L, H, Q, I, R) that define the architecture of the VEN model in Figure 4.36, is a very crucial matter because it determines the ability of the network structure to approximate the function of the actual physiological system (with regard to the input-output mapping) for properly selected parameter values and for a broad ensemble of inputs. It should be clearly understood that the ability of a given network model to achieve a satisfactory approximation of the input-output mapping (with properly selected parameter values) is critically constrained by the selected network structure.

In the case of the VEN models, this selection task is as formidable as it is crucial, because some of the model parameters enter nonlinearly in the estimation process--unlike the case of the discrete Volterra model, where the parameters enter linearly and a model order selection criterion can be rigorously applied (see Section 2.3.1). Therefore, even if one assumes that proper convergence can be achieved in

the iterative cost-minimization procedure (discussed below), the issue of rigorously assessing the significance of residual reduction with increasing model complexity remains a formidable challenge.

To address this issue, we establish the following guiding principles:

(1) the assessment of significance of residual reduction must be statistical, since the data-contaminating noise/interference is expected to be stochastic (at least in part);

(2) the simplest measure of residual reduction is the change in the sum of the squared residuals (SSR) as the model complexity increases (i.e., for increasing values of L, H, Q, I, R);

(3) proceeding in ascending model complexity, a statistical hypothesis test is performed at each step, based on the "null hypothesis" that the current model structure is the right one and examining the residual reduction in the next step (i.e., the next model order) using a statistical criterion constructed under the null hypothesis;

(4) to maximize the statistical independence of the model residuals used for the SSR computation (a fact that simplifies the construction of the statistical criterion by assuming whiteness of these residuals), we evaluate the SSR from randomly selected data-points of the output (the "testing dataset") while using the remaining output data-points for network training (the "training dataset");

(5) the statistics of the residuals used for the SSR computation are assumed to be approximately Gaussian in order to simplify the statistical derivations and justify the use of a quadratic cost function.

Based on these principles, we examine a sequence of network model structures $\{S_k\}$ in ascending order of complexity, starting with $L=H=Q=I=R=1$ and incrementing each structural parameter sequentially in the presented rightward order (i.e., first we increment L all the way to $L_{\max}$ and then increment H, etc.). At the k th step, the network structure S_k is trained with the "training dataset" and the resulting SSR J_k is computed from the "testing dataset". Because the residuals are assumed Gaussian and white (see #4 and #5 above), J_k follows a chi-square distribution with degrees of freedom equal to the size of the "testing dataset" minus the number of free parameters in S_k. Subsequently, an F statistic can be used to test the ratio J_k/J_{k+1} against a statistical threshold for a specified level of confidence. If the threshold is not exceeded, then the null hypothesis is accepted and the network structure S_k is anointed the "right one" for this system; otherwise the statistical testing procedure continues with the next network structure of higher complexity.

This selection procedure appears straightforward but is subject to various pitfalls, rooted primarily in the stated assumptions regarding the nature of the actual residuals and their interrelationship with the specific input data used in this procedure. It is evident that the pitfalls are minimized when the input data are close to band-limited white noise (covering the entire bandwidth and dynamic range of the system) and the actual residuals are truly white and Gaussian, as well as totally independent from both the input and the output.

The application of this procedure is demonstrated is Section 4.3 in connection with the Laguerre-Volterra network, which is the most widely used Volterra-equivalent network model to date.

Convergence and Accuracy of the Training Procedure

Having selected the structural parameters (L, H, Q, I, R) of the VEN model, we must "train" it using the "training set" of input-output data. The verb "train" is used to indicate the iterative estimation of the VEN model parameters through minimization of a cost function defined by the "testing set" of the input-output data.

As indicated above, the available input-output data are divided into a "training set" (typically about 80% of the total) and a complementary "testing set" using random sampling to maximize the statistical independence of the model prediction errors/residuals at the points of the training set. This random sampling is also useful in mitigating the effects of possible nonstationarities in the system, as discussed in Chapter 9. Note that the input data is comprised of the vector of preprocessed data of the filterbank outputs: $\mathbf{v}(n) = \left[v_1(n), ..., v_L(n) \right]'$, which are contemporaneous with the corresponding output $y(n)$. Thus the random sampling selects about 20% of the time indices n for the testing set prior to commencing the training procedure on the basis of the remaining 80% samples of the training set.

For each data-point in the training set, the output prediction residual $\in(n)$ is computed for the current values of the network parameters. This residual error is used to update the values of the VEN parameter estimates, based on a gradient-descent procedure, such as the one shown in Equation (4.132) or a variant of it, as discussed below. Search procedures, either deterministic or stochastic (e.g., genetic algorithms), are also possible candidates for this purpose but are typically more time consuming. The reader is urged to explore the multitude of interesting approaches and algorithms that are currently available in the extensive literature on artificial neural networks [Haykin, 1994; Hassoun, 1995] on the

classic problem of nonlinear cost minimization that has been around since the days of Newton and still defies a "definitive solution".

In this section, we will touch on some of the key issues germane to the training of feedforward Volterra-equivalent network models of the type depicted in Figure 4.36. These issues are:

(1) selection of the training and testing data sets;

(2) network initialization;

(3) enhanced convergence for fixed step;

(4) variable step algorithms

The *selection of the training and testing data sets* entails, in addition to the aforementioned random sampling, the sorting of the input data vectors $\{\mathbf{v}(n)\}$ so that if their Euclidean distance in the L-dimensional space is shorter than a specified "minimal proximity" value, then the data can be consolidated by using the vector averages within each "proximity cell". The rationale for this consolidation is that proximal input vectors $\mathbf{v}(n)$ are expected to have small differential effect on the output (in which case their "training value" is small) or, if the respective observed outputs are considerably different, then this difference is likely due to noise/interference and will be potentially misleading in the training context. This "data consolidation" is beneficial in the testing context as well, because it improves the signal-to-noise ratio and makes the measurements of the quadratic cost function more robust.

The "proximity cells" can be defined either through a Cartesian grid in the L-dimensional space or through a clustering procedure. In both cases, a "minimal proximity" value Δv must be specified that quantifies our assessment of the input signal-to-noise ratio (which determines the input vector "jitter") and the differential sensitivity of the input-output mapping for the system at hand. This value Δv defines the spacing of the grid or the cluster size. Note that this "minimal proximity" value Δv may vary depending on an estimate of the gradient of the output at each specific location in the L-dimensional space. Since this gradient is not known *a priori*, a conservative estimate can be used or the "data consolidation" procedure can be applied iteratively. The downside risk of this procedure is the possibility of excessive smoothing of the surface that defines the mapping of the input vector $\mathbf{v}(n)$ onto the output $y(n)$. Finally, the random sampling of the consolidated data for the selection of the testing data set is subject to a minimum time-separation between selected datapoints in order to minimize the probability of correlated residuals. The remaining datapoints form the training data set.

The *network initialization* concerns the critical issue of possible entrapment in local minima during the training procedures. This is one of the fundamental pitfalls of gradient-based iterative procedures, since unfortunate initialization may lead to a "stable" local minimum (i.e., a locally "deep" trough of the cost function surface that remains much higher than the global minimum). This risk is often mitigated by selecting multiple initialization points (that "sample" the parameter space with sufficient density either randomly or with a deterministic grid) and comparing the resulting minima in order to select the global minimum. This procedure is sound but can be very time-consuming when the parameter space is multidimensional. This problem also gave impetus to search algorithms (including genetic algorithms that make random "mutation" jumps) which, however, remain rather time-consuming.

In general, there are no definitive solutions to this problem. However, for Volterra-type models of physiological systems, one may surmise that the higher order Volterra terms/functionals will be of gradually declining importance relative to the first two orders in most cases. Consequently, one may obtain initial second-order Volterra approximations (using direct inversion or iterative methods) and use these approximations to set many of the initial network parameter values in the "neighborhood" of the global minimum. Subsequently, the correct model order can be selected without second-order limitation and the training procedure can be properly performed to yield the final parameter estimates with minimal risk of "local minimum" entrapment.

It is worth noting that the aforementioned "data consolidation" procedure is expected to alleviate some of the local minima that are due to noise in the data (input and output). However, the morphology of the minimized cost function depends on the current estimates of the network parameters and, therefore, *changes continuously* throughout the training process. The general morphology also changes for each different datapoint in the training set, and the location of the global minimum may shift depending on the actual residual/noise at each datapoint of the testing set. It is expected that the global minimum for the entire testing set will be very close to the location defined by the true parameter values of the network model. The basic notion of the changing surface morphology of the cost function during the training process is not widely understood or appreciated, although its implications for the training process can be very important (e.g., appearance and disappearance of local minima during the iterative process). A static notion of the cost function can be seriously misleading as it supports an unjustifiable faith in the chancy estimates of the gradient which is everchanging. When the cost function is formed by the summation of all squared residuals in the testing set, then this morphology remains invariant at least with respect to the individual datapoints, but still changes with respect to the continuously updated parameter values. Note that these updates are based on gradient estimates of the everchanging cost

function surfaces for the various datapoints in the training set. Although one may be tempted to combine many training datapoints in batch form in order to make the cost function surface less variable in this regard, this has been shown empirically to retard the convergence of the training algorithm. Somewhat counter-intuitively, the individual training datapoints seem to facilitate the convergence speed of the gradient-descent algorithm.

Enhanced convergence algorithms for fixed training step have been extensively studied, starting with the Newton-Raphson method that employs "curvature information" by means of the second partial derivatives forming the Hessian matrix [Eykhoff, 1974; Haykin, 1994]. This approach has also led to the so-called "natural gradient" method that takes into account the coupling between the updates of the various parameters during the training procedure using eigendecomposition of the Hessian matrix in order to follow a "most efficient" path to cost minimization. Generally, the gradient-based update of the parameter p_k at the $(i+1)$ iteration step is given by:

$$\Delta p_k^{(i)} \equiv p_k^{(i+1)} - p_k^{(i)} = -\gamma \frac{\partial J^{(i)}(n)}{\partial p_k} \tag{4.136}$$

However, this update of parameter p_k changes the cost function surface that is used for the update of the next parameter p_{k+1} by approximately:

$$\Delta J_{k,i+1}(n) \cong \frac{\partial J^{(i)}(n)}{\partial p_k} \Delta p_k^{(i)} \cong -\gamma \left\{ \frac{\partial J^{(i)}(n)}{\partial p_k} \right\}^2 \tag{4.137}$$

Thus, the update of the p_{k+1} parameter should be based on the gradient of the "new" cost function $J^{(i)}(n)$:

$$\frac{\partial J^{(i)}}{\partial p_{k+1}} = \frac{\partial J^{(i)}}{\partial p_{k+1}} + \frac{\partial^2 J^{(i)}}{\partial p_{k+1} \partial p_k} \Delta p_k^{(i)} \cong \frac{\partial J^{(i)}}{\partial p_{k+1}} - \gamma \frac{\partial}{\partial p_{k+1}} \left\{ \frac{\partial J^{(i)}}{\partial p_k} \right\}^2 \tag{4.138}$$

It is evident that the "correction" of the cost-function gradient depends on the second partial derivative (curvature) of the ith update of the cost function (i.e., depends on the Hessian matrix of the cost function update, if the entire parameter vector is considered), leading to the second-order ith update of p_{k+1}:

$$\Delta p_{k+1}^{(i)} = -\gamma \frac{\partial J^{(i)}}{\partial p_{k+1}} + \gamma^2 \frac{\partial}{\partial p_{k+1}} \left\{ \frac{\partial J^{(i)}}{\partial p_k} \right\}^2 \tag{4.139}$$

which reduces back to the first-order ith update of the type indicated in Equation (4.136) when γ is very small.

Because of the aforementioned fundamental observation regarding the changeability of the cost-function surface during the training process, it appears imprudent to place high confidence in these gradient estimates (or their Hessian-based corrections). Nonetheless, the gradient-based approaches have found many useful applications and their refinement remains an active area of research. Since these requirements are elaborate and deserve more space than we can dedicate here, we refer the reader to numerous excellent sources in the extensive bibliography on this subject. We note the current popularity of the Levenberg-Marquardt algorithm (in part because it is available in MATLAB) and the useful notion, embedded in the "normalized least mean-squares" method, that the fixed step size may be chosen inversely proportional to the mean-square value of the input. The use of a momentum term in the update formula has also been found useful, whereby the update indicated by Equation (4.136) is not directly applied but is subject to first-order autorecursive filtering. The choice of the fixed step value remains a key practical issue in this iterative gradient-based approach.

The *variable step* algorithms for enhanced convergence deserve a brief overview because they were found to perform well in certain cases were convergence with fixed step proved to be problematic [Haykin, 1994]. Of these algorithms, some use alternate trials (e.g., the "beta rule") and others use previous updates of the parameters to adjust the stepsize (e.g., the "delta-bar-delta" rule). Note that the idea of reducing the step as a monotonic function of the iteration index, originating in "stochastic approximation" methods, was found to be of very limited utility.

The "beta rule" provides that the step size for the training of a specific parameter is either multiplied or divided by a fixed scalar β depending on which of the alternate trials yields a smaller cost function. Thus, both options are evaluated at each step and the one that leads to greater reduction of the cost function is selected. The proper value of the fixed scalar β has been determined empirically to be about 1.7 in the case of artificial neural networks.

The "delta-bar-delta" rule is rooted on two heuristic observations by Jacobs who suggested that if the value of the gradient retains its algebraic sign for several consecutive iterations then the corresponding step size should be increased. Conversely, if the algebraic sign of the gradient alternates over several successive iterations, then the corresponding stepsize should be decreased. These ideas were first implemented in the "delta-delta rule" that changes the stepsize according to the product of the last two gradient values. However, observed deficiencies in the application of the "delta-delta rule" led to the variant of the "delta-bar-delta" rule, that increases the step size by a small fixed quantity κ if the

gradient has the same sign with a low-pass filtered (smoothed) measure of previous gradient values, otherwise it decreases the step size by a quantity proportional to its current value (so that the step size remains always positive but may diminish asymptotically) [Haykin, 1994].

Breaking altogether with the conventional thinking of incremental updates, we propose a method that uses successive parabolic fits (based on local estimates of first and second derivatives) to define variable "leaps" in search of the global minimum. According to this method, the morphology of the cost-function surface with respect to a specific parameter p_k may be one of the three types shown in Figure 4.39. The parameter change (leap) is defined by the iterative relation:

$$p_k^{(i+1)} = p_k^{(i)} - \frac{J'\left(p_k\right)}{\left|J''\left(p_k\right)\right| + \varepsilon} \tag{4.140}$$

where J' and J'' denote the first and second partial derivatives of the cost function evaluated at $p_k^{(i)}$, and ε is a very small positive "floor value" used to avoid numerical instabilities when J'' approaches zero. Alternatively, the parameter is not changed when J'' is very close to zero, since the cost-function surface is continuously altered by the updates of the other parameters and, consequently, J'' is likely to attain non-zero values at the next iteration(s) (moving away from an inflection point on the surface). Clearly this method is a slight modification of the classic Newton-Raphson method that extends to concave morphologies and inflection points. The rationale of this approach is depicted in Figure 4.39.

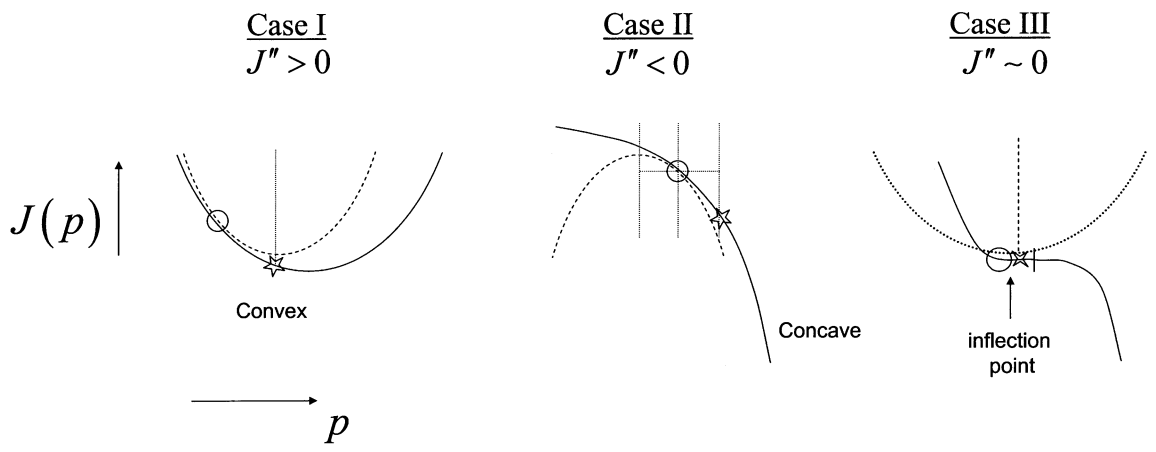


Figure 4.39
Illustration of the three main cases encountered in the use of the "parabolic leap" method of cost minimization. The simplified 2-D drawings show the cost function $J(p)$ with solid line and the parabolic local fits with dashed line. The abscissa is the value of the trained parameter p, whose starting value p_r at the r th iteration is marked with a circle and "landing" value (after the "leap") is marked with an asterisk. Case I of a convex morphology (left) illustrates the efficiency and reliable convergence of defining the "leap" using the minimum of the local parabolic fit (the classic Newton-Raphson method). Case II of a concave morphology (middle) illustrates the use of the symmetric point relative to the maximum of the local parabolic fit. Case III of convex/concave morphology with an inflection point (right) illustrates the low likelihood of a potential pitfall at the inflection point. The local parabolic fit has the same first and second derivative values as the cost function at the pivot point p_r, and consequently has the analytical form:

$$f(p) = \frac{1}{2} J''(p_r)(p-p_r)^2 + J'(p_r)(p-p_r) + J(p_r)$$

which defines a "leap landing" point: $p^* = p_r - J'(p_r) \big/ \left| J''(p_r) \right|$ as long as $J''(p_r) \neq 0$ (i.e., avoiding inflection points). This is a slightly modified form of the Newton-Raphson method that covers concave morphologies.

The Pseudo-Mode-Peeling Method

One practical way of addressing the problem of local minima in the training of separable VEN (SVN) models is the use of the "pseudo-mode-peeling" method, whereby a single-hidden-unit SVN model is initially fitted to the input-output data and, subsequently, another single-hidden-unit SVN model is fitted to the input-residual data, and so on until the residual is minimized (i.e., meets our criterion for model-order selection given in Section 2.3.1). Each of the single-hidden-unit SVN "sub-models" thus obtained corresponds to a "pseudo-PDM" of the system (termed the "pseudo-mode") with its associated nonlinearity. All the obtained SVN "sub-models" are combined by summation of their outputs to form the output of the overall SVN model of the system.

Although this method protects the user from getting "trapped" in a local minimum (by virtue of its ability to repeat the training task on the residual error and "peel" new pseudo-modes as needed), it is generally expected to yield different pseudo-modes for different initializations of the successive

"peeling" steps. Also, there is no guarantee that this procedure will always converge to the correct overall model, because it is still subject to the uncertainties of iterative minimization methods. Nonetheless, it is expected to reach the neighborhood of the correct overall model in most cases.

Another practical advantage of this method is that it simplifies the model-order selection task by limiting it to the structural parameters L and Q at each "peeling" step. Note that the overall model (comprised of all the "peeled" pseudo-modes) can be used to construct the *unique* Volterra kernels of the system, which can be used subsequently to determine the PDMs of the system using the singular-value decomposition method presented in Section 4.1.1. In this manner, the proposed approach is expected to arrive at similar overall PDM models regardless of the particular initializations used during the pseudo-mode-peeling procedure.

An alternative utilization of the overall model resulting from the pseudo-mode-peeling method is to view it as a "de-noising" tool, whereby the output noise is equated with the final residuals of the overall model. The "de-noised" data can be subsequently used to perform any other modeling or analysis procedure at a higher output SNR. This can offer significant practical advantages in low-SNR cases.

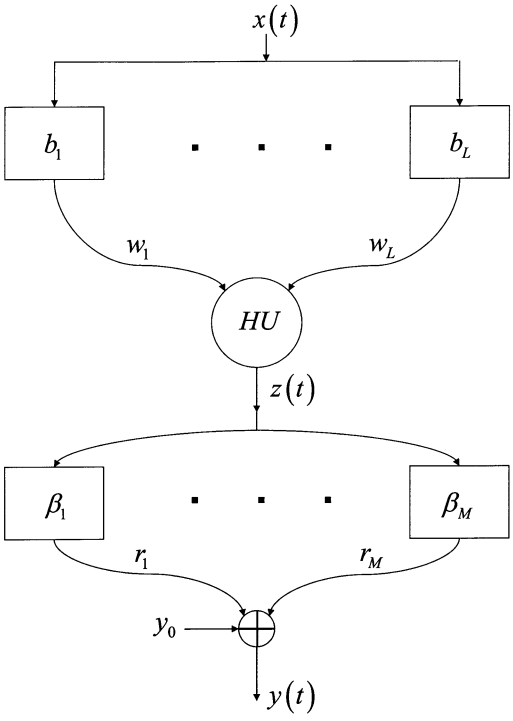


Figure 4.40
The L-N-M equivalent network model for a "peeling mode", employing two filterbanks $\{b_l\}$ and $\{\beta_m\}$ where $l = 1,\ldots,L$ and $m = 1,\ldots,M$. The second filterbank processes the output $z(t)$ of the hidden unit (HU) and captures the "posterior" filter operation through the output weights $\{r_m\}$.

Another variant of the pseudo-mode-peeling method, that may be useful in certain cases, involves the use of a "posterior" filter in each mode branch (akin to an L-N-M cascade). This method can be implemented by the network model architecture shown in Figure 4.40, whereby two filterbanks are employed to represent the two filtering operations: one pre-processes the input signal (prior filter) and the other processes the output of the hidden unit for each "peeling pseudo-mode" (posterior filter). It is evident that this network model architecture gives rise to an overall model of parallel L-N-M cascades that deviates from the basic Volterra-Wiener modular form or from the equivalent PDM model form of Figure 4.1. This model form can be more efficient (i.e., less modes required) in certain cases.

Nonlinear Auto-Regressive Modeling (Open-Loop)

The Volterra-equivalent network models discussed above can be also deployed in an auto-regressive context without recurrent connections (i.e., open-loop configurations), whereby the "input" is defined as the past epoch of the signal from discrete time ($n-1$) to ($n-M$) and the output is defined as the value of the signal at discrete time n. Thus, we can obtain the nonlinear mapping of the past epoch of a signal $\left[y(n-1),...,y(n-M)\right]$ onto its present value $y(n)$ in the form of a feedforward network model that constitutes an "open loop" nonlinear auto-regressive (NAR) model of the signal $y(n)$. Clearly, this NAR model attains the form of an auto-regressive discrete Volterra model (with well-defined auto-regressive kernels) that is equivalent to a nonlinear difference equation and describes the "internal dynamics" of a single process/signal. The residual of this NAR model can be viewed as the "innovations process" of conventional auto-regressive terminology (i.e., the unknown signal that drives the generation of the observed signal through the NAR model). It is evident that this class of models can have broad utility in physiology, since it pertains to the characterization of single processes (e.g., heart rate variability, neuronal rhythms, endocrine cycles, etc.).

The estimation approach for this NAR model form is determined by the criterion imposed on the properties of the residuals. For instance, if we seek to minimize the variance of these residuals, then the estimation approach is similar to the aforementioned ones (i.e., least-squares estimation). However, residual variance minimization may not be a sensible requirement in the auto-regressive case, because the residuals represent an "innovation process" and do not quantify a "prediction error" as before. Thus, we may require that the residuals be white (i.e., statistically independent innovations that yield the "maximum entropy" solution) or that they have minimum correlations with certain specified variables

(maximizing the "mutual information" criterion) or that they exhibit specific statistical and/or spectral properties (as prescribed by previous knowledge about the process). This important issue is revisited in Chapter 10 in connection wit the modeling of multiple interconnected physiological variables operating in closed-loop or nested-loop configurations that represent the ultimate modeling challenge in systems physiology.

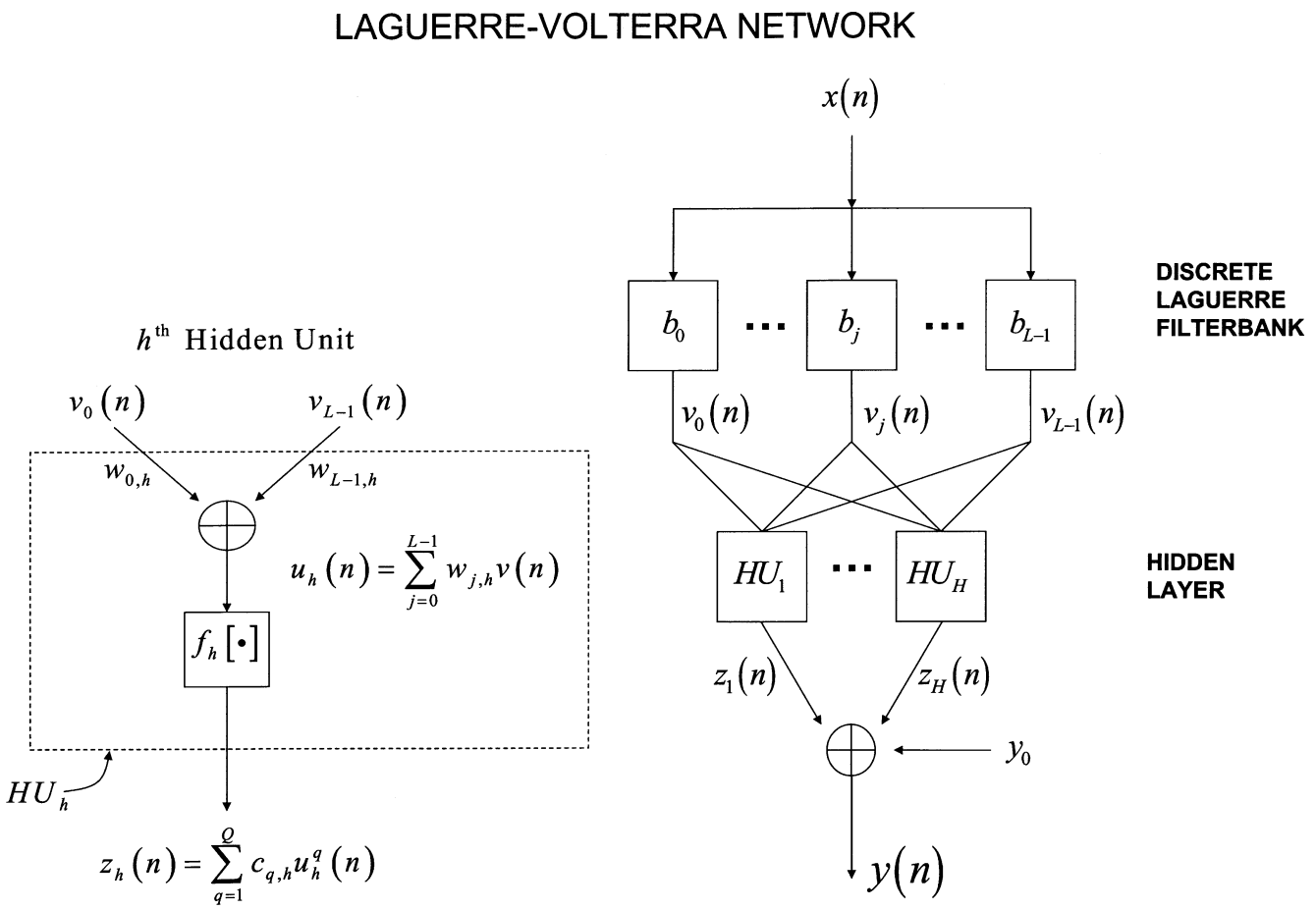


Figure 4.41
The Laguerre-Volterra Network (LVN) architecture employing a discrete Laguerre filterbank for input pre-processing and a single hidden layer with polynomial activation functions distinct for each hidden unit HU_h (see text).

4.3. THE LAGUERRE-VOLTERRA NETWORK

The most widely used Volterra-equivalent network model to date has been the so-called "Laguerre-Volterra Network" (LVN) which employs a filterbank of discrete-time Laguerre functions (DLFs) to preprocess the input signal (as discussed in Section 2.3.2 in connection with the Laguerre expansion technique) and a single hidden layer with polynomial activation functions, as shown in Figure 4.41 [Marmarelis, 1997; Alataris et al. 2000; Mitsis & Marmarelis, 2002]. The LVN can be viewed as a network-based implementation of the Laguerre expansion technique (LET) for Volterra system modeling that employs iterative cost minimization algorithms (instead of direct least-squares inversion employed by LET) to estimate the unknown kernel expansion coefficients (through the estimates of the LVN parameters) *and* the DLF parameter α. The latter has proven to be extremely important in actual applications, because it determines the efficiency of the Laguerre expansion of the kernels and constitutes a critical contribution by the author to the long evolution of Laguerre expansion methods. The LVN approach has yielded accurate and reliable models of various physiological systems since its recent introduction, using relatively short input-output data-records (see Chapter 6).

The basic architecture of the LVN is shown in Figure 4.41 and follows the standard architecture of a single-layer fully-connected feedforward artificial neural network, with three distinctive features that place it in the SVN/VEN class of models: (1) the DLF filterbank that pre-processes the input with trainable parameter α; (2) the polynomial (instead of the conventional sigmoidal) activation functions in the hidden units; (3) the non-weighted summative output unit (no output weights are necessary because the coefficients of the polynomial activation functions in the hidden units are trainable). Note that the polynomial activation functions do not have constant terms, but the output unit has a trainable offset value. The LVN has been shown to be equivalent to the Volterra class of finite-order models [Marmarelis, 1997] and yields interpretable PDM models (see Chapter 6).

As discussed in Section 2.3.2, the output $v_j(n)$ of the j-th DLF can be computed by means of the recursive relation (2.203) that improves the computational efficiency of the DLF expansion, provided the parameter α can be properly selected. A major advantage of the LVN approach over the Laguerre expansion technique is the iterative estimation of α (along with the other LVN parameters).

The input of the h-th hidden unit is the weighted sum of the DLF filterbank outputs:

$$u_h(n) = \sum_{j=0}^{L-1} w_{j,h} v_j(n) \tag{4.141}$$

where $h = 1, 2, ..., H$ and the DLF index j ranges from 0 to $L-1$, instead of the conventional range from 1 to L used in Section 2.3.1 (since the zero-order DLF is defined as the first term in the filterbank). The output of the h-th hidden unit is given by the polynomial activation function:

$$z_h(n) = \sum_{q=1}^{Q} c_{q,h} u_h^q(n) \tag{4.142}$$

where Q is the nonlinear order of the equivalent Volterra model. The LVN output is given by the non-weighted summation of the hidden-unit outputs, including a trainable offset value y_0:

$$y(n) = \sum_{h=1}^{H} z_h(n) + y_0 \tag{4.143}$$

The parameter a is critical for the efficacy of this modeling procedure because it defines the form of the DLFs and it determines the convergence of the DLF expansion, which in turn determines the computational efficiency of this approach. In the original introduction of the Laguerre expansion technique, the key parameter a was specified beforehand and remained constant throughout the model estimation process. This presented a serious practical impediment in the application of the method, because it required tedious and time-consuming trials to select the proper a value that yielded an efficient Laguerre expansion (i.e., with rapid convergence and capable of accurate kernel representation). As discussed in Section 2.3.2, the recursive relation (2.203) allows the computationally efficient evaluation of the prediction error gradient with respect to α, so that the iterative estimation of the parameter a from the data is feasible, thus removing a serious practical limitation of the original Laguerre expansion approach.

It is evident from Equations (2.203) and (2.207) that the parameter $\beta = \sqrt{\alpha}$ can be estimated using the iterative expression:

$$\beta^{(r+1)} = \beta^{(r)} - \gamma_\beta \varepsilon^{(r)}(n) \sum_{h=1}^{H} f_h'^{(r)}(u_h) \sum_{j=0}^{L-1} w_{h,j} \left[v_j(n-1) + v_{j-1}(n) \right] \tag{4.144}$$

where $\varepsilon^{(r)}(n)$ is the output prediction error at the r-th iteration, γ_β is the fixed learning constant (update stepsize) and $f_h'^{(r)}(u_h)$ is the derivative of the polynomial activation function of the h-th hidden unit at the r-th iteration:

$$f_h'^{(r)}\left[u_h^{(r)}(n) \right] = \sum_{q=1}^{Q} q c_{q,h}^{(r)} \left[u_h^{(r)}(n) \right]^{q-1} \tag{4.145}$$

where the superscript (r) denotes the value of the subject variable/parameter at the r th iteration.

The iterative relations for the estimation of the other trainable parameters of the LVN are:

$$w_{h,j}^{(r+1)} = w_{h,j}^{(r)} + \gamma_w \varepsilon^{(r)}(n) f_h'^{(r)}\left[u_h^{(r)}(n)\right] v_j^{(r)}(n) \tag{4.146}$$

$$c_{q,h}^{(r+1)} = c_{q,h}^{(r)} + \gamma_c \varepsilon^{(r)}(n)\left[u_h^{(r)}(n)\right]^q \tag{4.147}$$

$$y_0^{(r+1)} = y_0^{(r)} + \gamma_y \varepsilon^{(r)}(n) \tag{4.148}$$

where $j = 0,1,...,L-1;\ h = 1,2,...,H;\ \ q = 1,2...,Q;\ \ \gamma_w,\ \gamma_c$ and γ_y denote the respective learning constants for the weights, the polynomial coefficients and the output offset.

In order to assist the reader in making the formal connection between the LVN and the Volterra models, we note that successive substitution of the variables $\{z_h\}$ in terms of $\{u_h\}$ and, in turn, $\{v_j\}$, yields an expression of the output in terms of the input that is equivalent to the discrete Volterra model. Then it can be seen that the Volterra kernels can be expressed in terms of the LVN parameters as:

$$k_0 = y_0 \tag{4.149}$$

$$k_1(m_1) = \sum_{h=1}^{H} c_{1,h} \sum_{j=0}^{L-1} w_{h,j} b_j(m_1) \tag{4.150}$$

$$k_2(m_1, m_2) = \sum_{h=1}^{H} c_{2,h} \sum_{j_1=0}^{L-1} \sum_{j_2=0}^{L-1} w_{h,j_1} w_{h,j_2} b_{j_1}(m_1) b_{j_2}(m_2) \tag{4.151}$$

…

$$k_Q(m_1,...,m_Q) = \sum_{h=1}^{H} c_{Q,h} \sum_{j_1=0}^{L-1} ... \sum_{j_Q=0}^{L-1} w_{h,j_1} ... w_{h,j_Q} b_{j_1}(m_1) ... b_{j_Q}(m_Q) \tag{4.152}$$

If so desired, once the training of the LVN is performed, the Volterra kernels can be evaluated from Equations (4.149)-(4.152). However, it is recommended that the physiological interpretation of the obtained model be based on the equivalent PDM model, and thus the Volterra kernels serve only as a general framework of reference and a means of validation/evaluation.

A critical practical issue is the selection of the LVN structural parameters, namely the number L of DLFs, the number H of hidden units and the degree Q of the polynomial activation functions, so that the LVN model is not underspecified or overspecified. This is done by successive trials in ascending order (i.e., moving from lower to higher parameter values, starting with $L = 1,\ H = 1,\ Q = 1$ and incrementing from left to right) using a statistical criterion to test the reduction in the computed mean-square error of the output prediction achieved by the model, properly balanced against the total number

of free parameters. To this purpose, we have developed the Model Order Selection (MOS) criterion that is described in Section 2.3.1.

Concerning the number L of DLFs, it was found that it can affect the convergence of the iterative estimation of the a parameter in certain cases where more DLFs in the LVN make a converge to a smaller value. The reason is that increasing DLF order results in a longer spread of significant values and increased distance between zero crossings in each DLF, which is also what happens when a is increased. Note that Q is independent of the L selection since it represents the intrinsic nonlinearity of the system. Likewise, the selection of L does not affect (in principle) the selection of H, which determines the number of required PDMs in the system under study. We must emphasize that the selection of these structural parameters of the LVN model is not unique in general, but the equivalent Volterra model ought to be unique.

Illustrative Example of LVN Modeling

To illustrate the efficacy of the LVN modeling approach, we simulate a 5th-order system described by the following difference and algebraic equations:

$$v(n) + A_1 v(n-1) A_2 v(n-2) = B_0 x(n) + B_1 x(n-1) \tag{4.153}$$

$$y(n) = v(n) + \frac{1}{2}v^2(n) + \frac{1}{3}v^3(n) + \frac{1}{4}v^4(n) + \frac{1}{5}v^5(n) \tag{4.154}$$

where the coefficient values are arbitrarily chosen to be: $A_1 = -1.77$, $A_2 = 0.78$, $B_0 = 0.25$, $B_1 = -0.27$. This system is equivalent to the cascade of a linear filter with two poles (defined by Equation (4.153)) followed by a fifth-degree polynomial static nonlinearity (defined by Equation (4.154)). In this example, we know neither the appropriate value of a nor the appropriate number L of DLFs, but we do know the appropriate values of the structural parameters: $H = 1$, $Q = 5$ -- knowledge that can be used as "ground-truth" for testing and validation purposes.. The first-order Volterra kernel can be found analytically by solving the difference equation (4.153) that yields:

$$k_1(m) = \frac{1}{4}\left(2p_1^m - p_2^m\right) \tag{4.155}$$

where $p_1 = \exp(-0.2)$, $p_2 = \exp(-0.05)$. The high-order Volterra kernels of this system are expressed in terms of the first-order kernel as:

$$k_r(m_1,...,m_r) = \frac{1}{r} k_1(m_1)...k_1(m_r) \tag{4.156}$$

for $r = 2,3,4,5$ (of course k_r is zero for $r > 5$).

The simulation of this system is performed with a unit-variance 1024-point Gaussian CSRS input and the selected LVN model has structural parameters: $L = 4$, $H = 1$, $Q = 5$. The obtained Volterra kernel estimates are identical to the true Volterra kernels of the system given by Equations (4.155) and (4.156). In order to examine the performance of the LVN approach in the presence of noise, the simulation is repeated for noisy output data, whereby an independent GWN signal is added to the output for a signal-to-noise ratio (SNR) equal to 0dB (i.e., the output signal power is equal to the noise variance). The a learning curves for both the noise-free and noisy cases are shown in Figure 4.42. We can see that a converges to nearby values: 0.814 in the noise-free case and 0.836 in the noisy case. Its convergence is not affected significantly by the presence of noise. The large values of a in this example reflect the fact that this system has slow dynamics (the spread of significant values of the first-order kernel is about 100 lags, which corresponds to the memory-bandwidth product of the system).

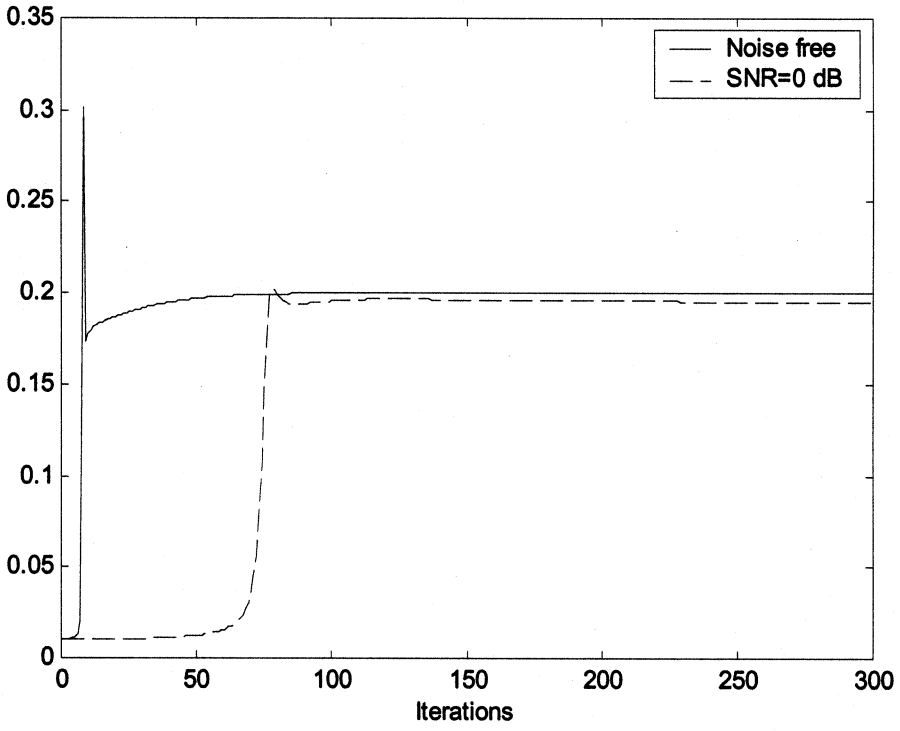
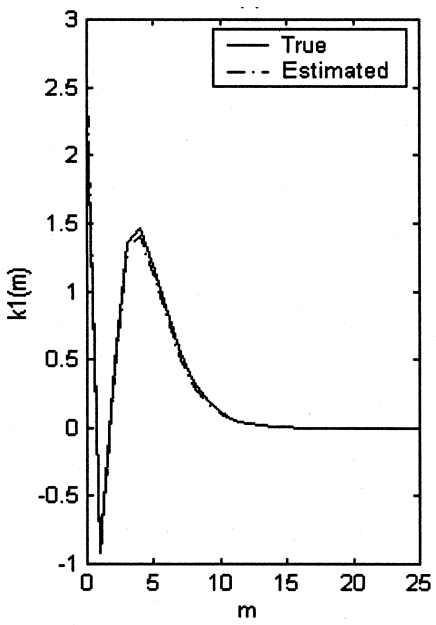
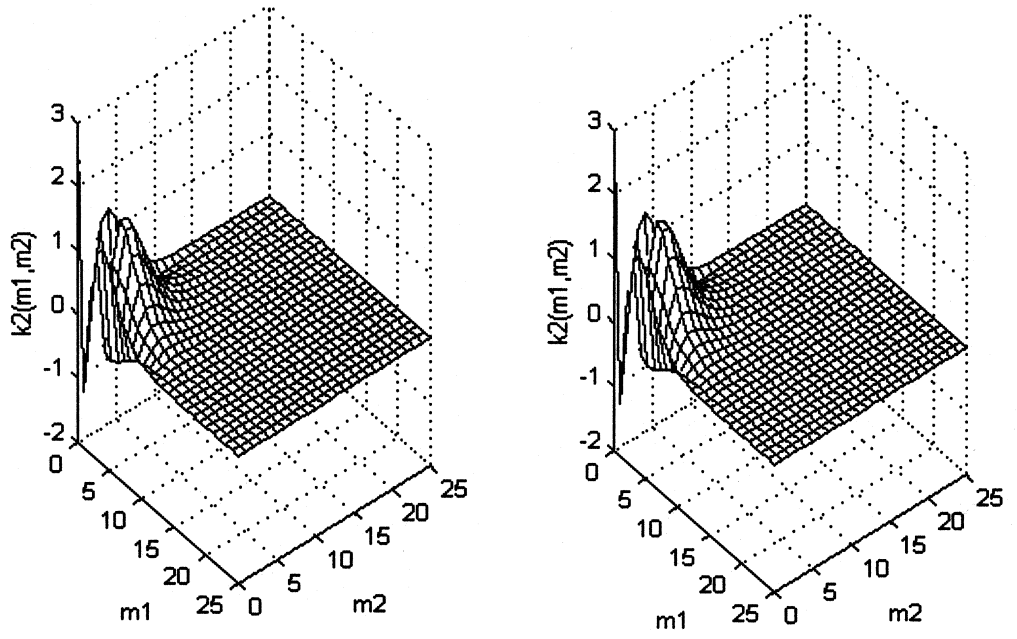


Figure 4.42
The learning curves of a for the simulated 5th-order system for noise-free and noisy outputs. Note that the learning constant for the noisy case is ten times smaller [Mitsis & Marmarelis, 2002].



a)



b)

Figure 4.43

(a) The true and estimated first-order Volterra kernel of the simulated 5th-order system using LVN ($L=4, H=1, Q=5$) for the noisy case of SNR=0 db [Mitsis & Marmarelis, 2002].

(b) The true (left) and estimated (right) second-order Volterra kernel of the simulated 5th-order system using LVN ($L=4, H=1, Q=5$) for the noisy case of SNR=0 db [Mitsis & Marmarelis, 2002].

The estimated first-order and second-order kernels in the noisy case are shown along with the true ones (which are identical to the noise-free estimate) in Figure 4.43. The noisy estimates exhibit excellent resemblance to the noise-free estimates (which are identical to the true kernels) despite the low-SNR data and the relatively short data-record of 1024 samples. However, it is evident that the second-order kernel estimate is affected more than its first-order counterpart by the presence of noise. The NMSE values of the LVN model prediction for a different GWN input and output data (out-of-sample prediction) are 0.2% and 49.41% for the noise-free and noisy cases respectively. Note that a perfect output prediction in the noisy case for SNR=0dB corresponds to 50% NMSE value. These results demonstrate the efficacy and the robustness of the LVN modeling approach, even for high-order systems (5^{th}-order in this example), low SNR (0dB) and short data-records (1024 samples).

In actual applications to physiological systems, the SNR rarely drops below 10dB and almost never below 0dB. Likewise, it is very rare to require a model order higher than fifth and it is not unusual to have data records of size comparable to 1024 (in fact, in the early days of the cross-correlation technique the data records had typically tens of thousands of samples). Therefore, this illustrative example offers a realistic glimpse at the quality of the modeling results achievable by the LVN approach.

The application of this modeling approach to actual physiological systems is illustrated in Chapter 6. It is accurate to say that the LVN approach has yielded the best Volterra modeling results to date with real physiological data and, therefore, points to a promising direction for future modeling efforts--without precluding further refinements or enhanced variants of this approach in the future.

Modeling Systems with Fast and Slow Dynamics (LVN-2)

Of the many possible extensions of the LVN approach, the most immediate concerns the use of multiple Laguerre filterbanks (with distinct parameters α) in order to capture multiple time-scales of dynamics intrinsic to a system. This is practically important because many physiological systems exhibit vastly different scales of fast and slow dynamics which may also be interdependent—a fact that makes their simultaneous estimation a serious challenge in a practical context. Note that fast dynamics require high sampling rates and slow dynamics necessitate long experiments, resulting in extremely long data-records (with all the burdensome experimental and computational ramifications). A practical solution to this problem can be achieved by a variant of the LVN approach with two filterbanks (one for fast and one for slow dynamics) discussed below [Mitsis & Marmarelis, 2002].

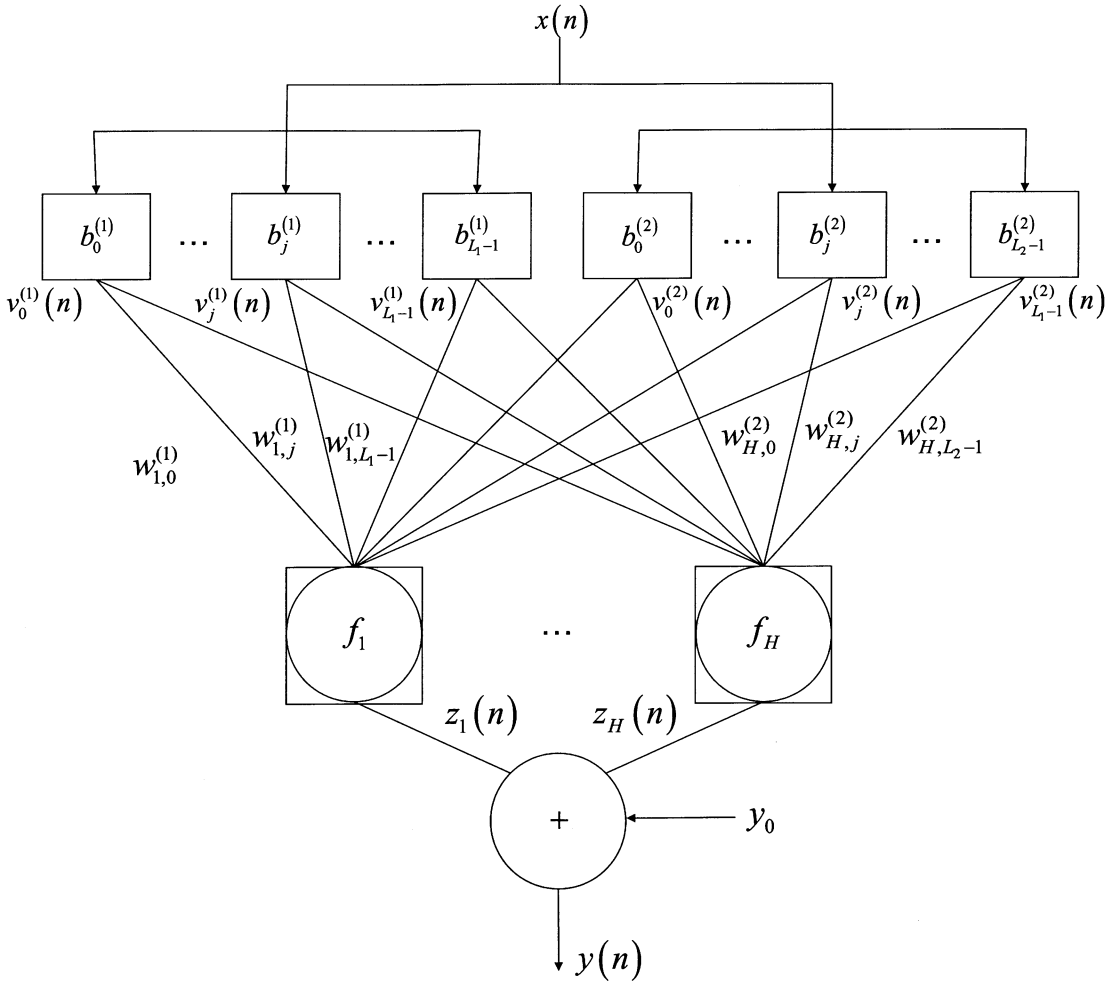


Figure 4.44

The LVN-2 model architecture with two Laguerre filterbanks $\left\{b_j^{(1)}\right\}$ and $\left\{b_j^{(2)}\right\}$ that preprocess the input $x(n)$. The hidden units in the hidden layer have polynomial activation functions $\left\{f_h\right\}$ and receive input from the outputs of both filterbanks. The output $y(n)$ is formed by summation of the outputs of the hidden units $\left\{z_h\right\}$ and the output offset y_0 [Mitsis & Marmarelis, 2002].

The proposed architecture of the LVN variant with two filter-banks (LVN-2) is shown in Figure 4.44. The two filterbanks preprocess the input separately and are characterized by different Laguerre parameters $(\alpha_1$ and $\alpha_2)$ corresponding generally to different numbers of DLFs (L_1 and L_2). A small value of α_1 for the first filterbank and a large value for α_2 for the second filterbank allows the simultaneous modeling of the fast and the slow components of a system, as well as their interaction.

As was discussed in Section 2.3.2, the asymptotically exponential structure of the DLFs makes them a good choice for modeling physiological systems, since the latter often exhibit asymptotically exponential structure in their Volterra kernels. However, one cannot rule out the possibility of system

kernels that do not decay smoothly—a situation that will require either a large number of DLFs or an alternate (more suitable) filterbank. The reader must be reminded that the parameter α defines the exponential relaxation rate of the DLFs and determines the convergence of the Laguerre expansion for a given kernel function. Larger α values result in longer spread of significant values (slow dynamics). Therefore the choice of the DLF parameters $(\alpha_1$ and $\alpha_2)$ for the two filterbanks of the LVN-2 model must not be arbitrary and is critical in achieving an efficient model representation of a system with fast and slow dynamics. This choice is made automatically by an iterative estimation procedure using the actual experimental data, as discussed earlier for the LVN model. For the LVN-2 model, the iterative estimation formula is:

$$\beta_i^{(r+1)} = \beta_i^{(r)} - \gamma_i \varepsilon^{(r)}(n) \sum_{h=1}^{H} \sum_{q=1}^{Q} \sum_{j=0}^{L_i-1} q \left[c_{q,h} w_{h,j}^{(i)} u_h^{q-1}(n) \left[v_j^{(i)}(n-1) + v_{j-1}^{(i)}(n) \right] \right]_r \tag{4.157}$$

where $i = 1,2$ is the filterbank index, $\varepsilon^{(r)}(n)$ is the output prediction error at the r th iteration, $\beta_i = \sqrt{\alpha_i}$, and $\{\gamma_i\}$ are fixed positive learning constants. The notation $[\cdot]_r$ means that the quantity in the brackets is evaluated for the parameter estimates at the r -th iteration.

The remaining variables and parameters are defined in a manner similar to the LVN case, as indicated in Figure 4.44. Note, however, that the filterbank index i appears as a subscript of L (since the two filter-banks may have different numbers of DLFs in general) and as a superscript of the weights $\left\{w_{h,j}^{(i)}\right\}$ and of the DLF outputs $\left\{v_j^{(i)}(n)\right\}$, indicating their dependence on the respective filterbank. The inputs to the polynomial activation functions of the hidden units are:

$$u_h(n) = \sum_{i=1}^{2} \sum_{j=0}^{L_i-1} w_{h,j}^{(i)} v_j^{(i)}(n) \tag{4.158}$$

and the LVN-2 output is given by:

$$y(n) = y_0 + \sum_{h=1}^{H} f_h\left[u_h(n)\right] \tag{4.159}$$

where each polynomial activation function performs the polynomial transformation:

$$f_h[u_h] = \sum_{q=1}^{Q} c_{q,h} u_h^q \tag{4.160}$$

that scrambles the contributions of the two filterbanks and generates output components that mix (or solely retain, if appropriate) the characteristics of the two filterbanks.

For instance, the equivalent first-order Volterra kernel of the LVN-2 model is composed of a fast component (denoted by subscript " f " and corresponding to $i=1$):

$$k_f(m)=\sum_{h=1}^{H} c_{1,h} \sum_{j=0}^{L_1-1} w_{h,j}^{(1)} b_j^{(1)}(m) \tag{4.161}$$

and a slow component (denoted by subscript " s " and corresponding to $i=2$):

$$k_s(m)=\sum_{h=1}^{H} c_{2,h} \sum_{j=0}^{L_2-1} w_{h,j}^{(2)} b_j^{(2)}(m) \tag{4.162}$$

where:

$$k_1(m)=k_f(m)+k_s(m) \tag{4.163}$$

The equivalent higher order Volterra kernels of the LVN-2 model contain also components that mix the fast with the slow dynamics (cross-terms). For instance, the 2nd-order kernel is composed of three components: a fast k_{ff}, a slow k_{ss} and a fast-slow cross-term k_{fs}, that are given by the expressions:

$$k_{ff}(m_1,m_2)=\sum_{h=1}^{H} c_{2,h} \sum_{j_1=0}^{L_1-1} \sum_{j_2=0}^{L_1-1} w_{h,j_1}^{(1)} w_{h,j_2}^{(1)} b_{j_1}^{(1)}(m_1) b_{j_2}^{(1)}(m_2) \tag{4.164}$$

$$k_{ss}(m_1,m_2)=\sum_{h=1}^{H} c_{2,h} \sum_{j_1=0}^{L_2-1} \sum_{j_2=0}^{L_2-1} w_{h,j_1}^{(2)} w_{h,j_2}^{(2)} b_{j_1}^{(2)}(m_1) b_{j_2}^{(2)}(m_2) \tag{4.165}$$

$$k_{fs}(m_1,m_2)=\sum_{h=1}^{H} c_{2,h} \sum_{j_1=1}^{L_1} \sum_{j_2=1}^{L_2} \left[w_{h,j_1}^{(1)} w_{h,j_2}^{(2)} b_{j_1}^{(1)}(m_1) b_{j_2}^{(2)}(m_2) + w_{h,j_1}^{(2)} w_{h,j_2}^{(1)} b_{j_1}^{(1)}(m_2) b_{j_2}^{(2)}(m_1) \right] \tag{4.166}$$

The second-order kernel is the summation of these three components. In general, the equivalent q th-order Volterra kernel can be reconstructed from the LVN-2 parameters as:

$$k_q(m_1,\ldots,m_q)=\sum_{h=1}^{H} c_{q,h} \sum_{i_1=1}^{2} \cdots \sum_{i_q=1}^{2} \sum_{j_1=0}^{L_{i_1}-1} \cdots \sum_{j_q=0}^{L_{i_q}-1} w_{h,j_1}^{(i_1)} \cdots w_{h,j_q}^{(i_q)} b_{j_1}^{(i_1)}(m_1) \cdots b_{j_q}^{(i_q)}(m_q) \tag{4.167}$$

It is evident that there is a wealth of information in these kernel components that cannot be retrieved with any other existing method. This approach can be extended to any number of filterbanks.

A critical practical issue for the successful application of the LVN-2 model is the proper selection of its structural parameters, i.e., the size of the DLF filterbanks L_1 and L_2, the number H of hidden units, and the degree Q of the polynomial activation functions. As in the LVN case discussed previously, this selection can be made by successive trials in ascending order (i.e., moving from lower to higher values of the parameters) until a proper criterion is met (e.g., the statistical MOS criterion presented in Section 2.3.1) that defines statistically the minimum reduction in the normalized mean-square error (NMSE) of

the output prediction achieved by the model for an increment in the structural parameters. Specifically, we commence the LVN-2 training with structural parameter values: $L_1 = L_2 = 1, H = 1, Q = 1$ and increment the structural parameters sequentially (starting with L_1 and L_2, and continuing with H and Q) until the MOS criterion is met. The training of the LVN-2 is performed in a manner similar to the training of the LVN (based on gradient descent) and has not presented any additional problems.

Illustrative Examples of LVN-2 Modeling

To illustrate the performance of the LVN-2 modeling approach, we use three simulated nonlinear systems: the first system is isomorphic to the LVN-2 model shown in Figure 4.44, (used as a "ground-truth" validation test), the second is a high-order modular system, and the third is a nonlinear parametric model defined by the differential equations (4.176)-(4.178) that are frequently used to describe biophysical and biochemical processes [Mitsis & Marmarelis, 2002].

The first simulated system has Volterra kernels that are composed of linear combinations of the first three DLFs with two distinct Laguerre parameters: $\alpha_1 = 0.2$ and $\alpha_2 = 0.8$. It can also be represented by the modular model of Figure 4.45 with two branches of linear filters (h_1 and h_2 representing the fast and slow dynamics respectively) feeding into the output static nonlinearity N given by the second-order expression:

$$y(n) = u_1(n) + u_2(n) + u_1^2(n) - u_2^2(n) + u_1(n)u_2(n) \tag{4.168}$$

where u_1 and u_2 are the outputs of h_1 and h_2 respectively, given by:

$$h_1(m) = b_0^{(1)}(m) + 2b_1^{(1)}(m) + b_2^{(1)}(m) \tag{4.169}$$

$$h_2(m) = b_0^{(2)}(m) - b_1^{(2)}(m) + 2b_2^{(2)}(m) \tag{4.170}$$

where $b_j^{(i)}(m)$ denotes the j th-order DLF with parameter α_i.

By substituting the convolutions of the input with h_1 and h_2 for u_1 and u_2, respectively, in Equation (4.168), the first-order and second-order Volterra kernels of this system are found to be:

$$k_1(m) = h_1(m) + h_2(m) \tag{4.171}$$

$$k_2(m_1, m_2) = h_1(m_1)h_1(m_2) - h_2(m_1)h_2(m_2) + \frac{1}{2}\left[h_1(m_1)h_2(m_2) + h_1(m_2)h_2(m_1)\right] \tag{4.172}$$

These kernels contain fast and slow components corresponding to the two distinct Laguerre parameters, as indicated in Equations (4.171) and (4.172).

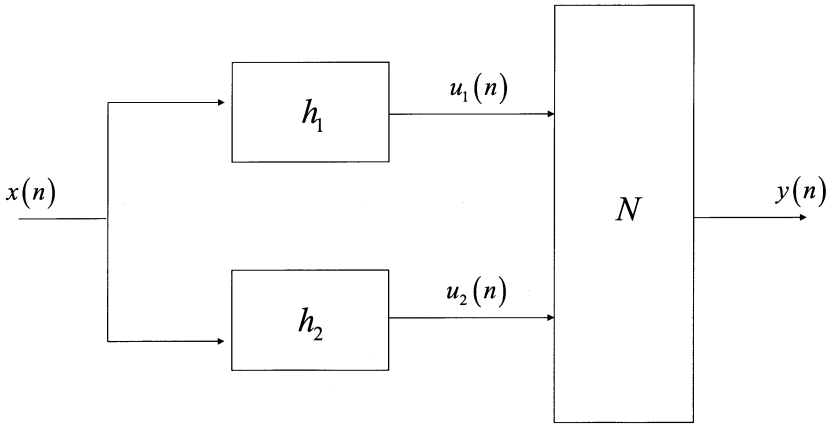


Figure 4.45
The modular model of the first simulated example of a second-order system [Mitsis & Marmarelis, 2002].

The system is simulated with a Gaussian CSRS input of unit variance and length of 1024 data points $(\Delta t = T = 1)$. Following the ascending-order MOS procedure described earlier, we determine that three DLFs in each filterbank $(L_1 = L_2 = 3)$ and three hidden units $(H = 3)$ with distinct second-degree polynomial activation functions $(Q = 2)$ are adequate to model the system, as theoretically anticipated. The number of free parameters of this LVN-2 model is: $(L_1 + L_2 + Q)H + 3 = 27$. Note that the obtained structural parameter values in the selected LVN-2 model are not unique in general, although in this example they match the values anticipated by the construction of simulated system. Different structural parameters (i.e., LVN-2 model configurations) may be selected for different data from the same system, and the corresponding LVN-2 parameter values (e.g., weights and polynomial coefficients) will be generally different as well. However, what remains constant is the equivalent Volterra representation of the system (i.e., the Volterra kernels of the system), regardless of the specific LVN-2 configuration selected or the corresponding parameter values.

In the noise-free case, the estimated LVN-2 kernels of first and second order are identical to their true counterparts, given by Equations (4.171) and (4.172) respectively, and the normalized mean-square error (NMSE) of the output prediction achieved by the LVN-2 model is on the order of 10^{-3}, demonstrating the excellent performance of this modeling procedure. The first-order kernel is shown in Figure 4.46, along with its slow and fast components estimated by LVN-2. The estimated second-order

kernel and its three components are shown in Figure 4.47, demonstrating the ability of the LVN-2 to capture slow and fast dynamics.

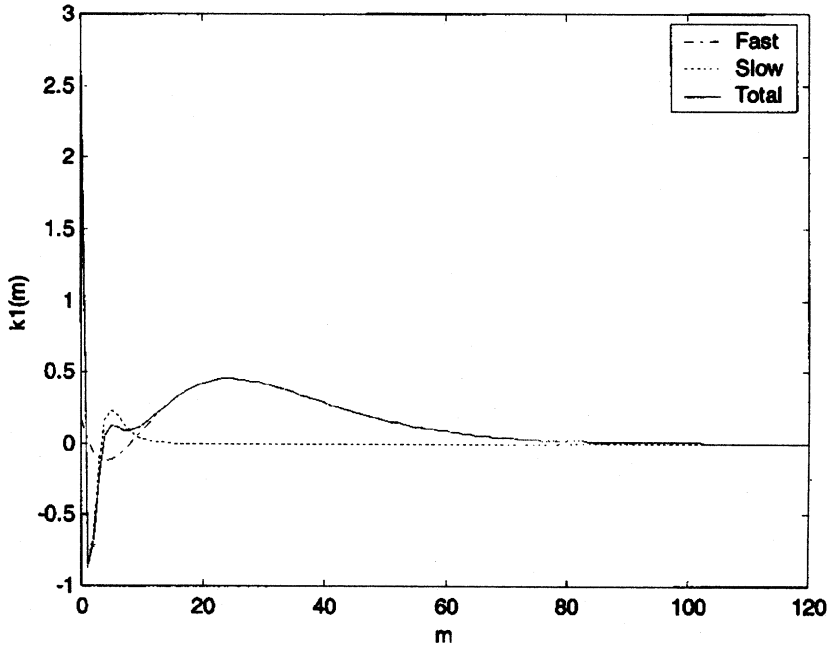


Figure 4.46
Estimated first-order Volterra kernel for the first simulated system and its slow/fast components [Mitsis & Marmarelis, 2002].

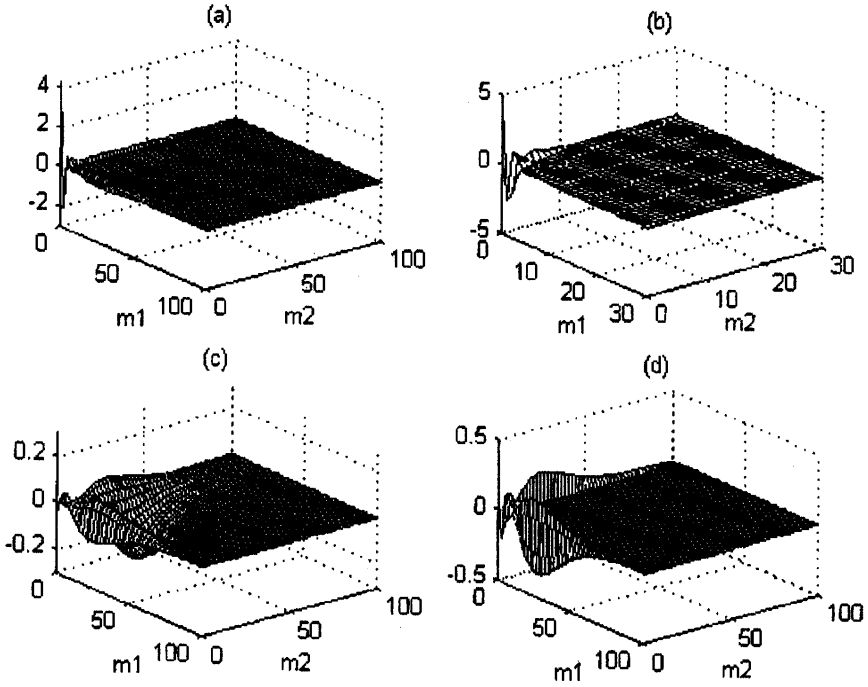


Figure 4.47
Estimated second-order Volterra kernel for the first simulated system (a), and its three components: (b) fast component, (c) slow component , and (d) fast-slow cross component [Mitsis & Marmarelis, 2002].

The utility of employing two filterbanks in modeling systems with fast and slow dynamics can be demonstrated by comparing the performance of the LVN-2 model with an LVN model (one filter-bank) having the same total number of free parameters $(L=6, H=3, Q=2)$. In order to achieve comparable performance with a single filterbank, it was found that we have to almost double the total number of free parameters to 44 from 27 [Mitsis & Marmarelis, 2002].

In order to examine the effect of noise on the performance of the LVN-2 model, we add independent GWN to the system output for a signal-to-noise ratio (SNR) equal to 0 dB (i.e., the noise variance equals the mean-square value of the de-meaned noise-free output). Convergence occurs in about 600 iterations (peculiarly, faster than in the noise-free case) and the final estimates of α_1, α_2 are not affected much by the presence severe noise (α_1=0.165 and α_2=0.811). The resulting NMSE value for the LVN-2 model prediction is 53.78 % in the noisy case, which is deemed satisfactory, since the ideal NMSE level is 50% for SNR=0dB. The NMSEs of the estimated first-order and second-order Volterra kernels in the noisy case are given in Table 4.1 along with the estimates of α_1 and α_2 that corroborate the previous favorable conclusion regarding the efficacy of the LVN approach, especially when compared to the estimates obtained via the conventional cross-correlation technique (also given in Table 4.1).

Table 4.1 Normalized mean-square errors (NMSEs) for model prediction and kernel estimates using LVN-2 and conventional cross-correlation.

			Prediction	Kernel NMSEs	
	α_1	α_2	NMSE (%)	$k_1(m)$ (%)	$k_2(m_1, m_2)$ (%)
LVN-2	0.165	0.811	53.78	7.69	4.10
Cross-correlation	⋯	⋯	86.38	421	1919

The second simulated system also has the modular architecture shown in Figure 4.45 but with a fourth-order nonlinearity given by:

$$y(n) = u_1(n) + 2u_1(n) + 4u_1^2(n) - 4u_2^2(n) + 4u_1(n)u_2(n) + \frac{1}{3}u_1^3(n) + \frac{1}{2}u_2^3(n) + \frac{3}{4}u_1^4(n) + \frac{1}{2}u_2^4(n) \quad (4.173)$$

and linear filter impulse responses that are not linear combinations of DLFs but given by:

$$h_1(m) = \exp\left(-\frac{m}{3}\right)\sin\left(\frac{\pi m}{5}\right) \quad (4.174)$$

$$h_2(m) = \exp\left(-\frac{m}{20}\right) - \exp\left(\frac{m}{10}\right) \quad (4.175)$$

Employing the ascending-order MOS procedure for the selection of the structural parameters of the LVN-2 model, we select: $L_1 = L_2 = 7, H = 4, Q = 4$ (a total of 75 LVN-2 model parameters). The results for a Gaussian CSRS input of 4096 data points are excellent in the noise-free case, as before, and demonstrate the efficacy of the method for this high-order system.

The effect of output-additive noise on the performance of the LVN-2 model is examined for this system by adding 20 different independent GWN signals to the output for an SNR of 0 dB. The resulting NMSE values (computed over the 20 independent trials) for the output prediction and for the estimated kernels [mean value $\pm$ standard deviation] are: $[48.42 \pm 2.64\%]$ for the output prediction, $[3.72 \pm 2.37\%]$ for the k_1 estimate, and $[6.22 \pm 3.82\%]$ for the k_2 estimate. The robustness of the method is evident since the prediction NMSE is close to 50% (ideal NMSE for SNR=0 dB) and the kernel NMSEs are low compared to the variance of the output-additive noise. In fact, the NMSE of the kernel estimates can be used to define an SNR measure for the kernel estimates as: $-10\log(\text{NMSE})$, which yields about 14 dB for the k_1 estimate and about 12 dB for the k_2 estimate.

Finally, a third system of different structure is simulated, that is described by the following differential equations (the input-output data are properly discretized after continuous-time simulation of these differential equations):

$$\frac{dy(t)}{dt} + b_0 y(t) = \left[c_1 z_1(t) - c_2 z_2(t)\right] y(t) + y_0 \tag{4.176}$$

$$\frac{dz_1(t)}{dt} + b_1 z_1(t) = x(t) \tag{4.177}$$

$$\frac{dz_2(t)}{dt} + b_2 z_2(t) = x(t) \tag{4.178}$$

where y_0 is the output baseline (basal) value and $z_1(t)$, $z_2(t)$ are internal state variables whose products with the output $y(t)$ in the bilinear terms of Equation (4.176) constitute the nonlinearity of this system, which gives rise to an equivalent Volterra model of infinite order. The nonlinearities of this system (i.e., the bilinear terms of Equation (4.176)) may represent modulatory effects that are often encountered in physiological regulation (neural, endocrine, cardiovascular, metabolic, immune systems) or in intermodulatory interactions of cellular/molecular mechanisms (including voltage-dependent conductances of ion channels in neuronal membranes or ligand-dependent conductances in synapses--see Chapter 8).

The contribution of the q th-order Volterra kernel of this system is proportional to the q th powers of Rc_1 and Rc_2, where R is the root-mean-square value of the input. When the magnitudes of c_1 and c_2 are smaller than one, a truncated Volterra model can be used to approximate the system. For the parameter values of $b_0 = 0.5$, $b_1 = 0.2$, $b_2 = 0.02$, $c_1 = 0.3$ and $c_2 = 0.1$, it was found that a fourth-order LVN-2 model was sufficient to represent the system output for a Gaussian CSRS input with unity power level and length of 2048 data points.

Following the proposed MOS search procedure for the selection of the structural parameters of the model, an LVN-2 with $L_1 = L_2 = 5, H = 3, Q = 4$ was selected (a total of 45 free parameters). The obtained results for the noise-free and noisy (SNR=0dB) conditions are given in Table 4.2, demonstrating the excellent performance of the LVN-2 model for this system as well. It should be noted that the estimated zeroth-order kernel was equal to 1.996, very close to its true value of 2.

Table 4.2 LVN-2 model performance for the system described by Equations (4.176)-(4.178).

	α_1	α_2	Output Prediction NMSE (%)	1st-Order Kernel NMSE (%)
Noise-free output	0.505	0.903	0.28	0.10
Noisy output (SNR = 0 dB)	0.366	0.853	47.49	4.13

The equivalent Volterra kernels for this system can be analytically derived by using the generalized harmonic balance method described in Section 3.2. The resulting analytical expressions for the zeroth and first-order kernels are:

$$k_0 = \frac{y_0}{b_0} \tag{4.179}$$

$$k_1(m) = \frac{y_0}{b_0}\left\{\frac{c_1}{b_1 - b_0}\left[\exp(-b_0 m) - \exp(-b_1 m)\right] - \frac{c_2}{b_2 - b_0}\left[\exp(-b_0 m) - \exp(-b_2 m)\right]\right\} \tag{4.180}$$

The analytical forms of the higher-order kernels are rather complex and are not given here in the interest of space, but the general expression for the second-order Volterra kernel is given by Equation (3.71). The fast component of the first-order kernel corresponds to the first exponential difference in

Equation (4.180), whereas the slow component corresponds to the second exponential difference in Equation (4.180) (recall that b_1 is ten times bigger than b_2 in this simulated example).

We close this section with the conclusion that the use of LVN offers powerful means for efficient modeling of nonlinear physiological systems from short input-output data records. The problem of efficient modeling of nonlinear systems with fast and slow dynamics can also be addressed by employing two filterbanks characterized by distinct Laguerre parameters (the LVN-2 model). The efficiency and robustness of this approach were demonstrated in the presence of severe output-additive noise.

4.4. THE VWM MODEL

The basic rationale for the use of the Volterra-equivalent network models is twofold: (1) the separation of the dynamics from the nonlinearities occurring at the first hidden layer, and (2) the compact representation of the dynamics with a "judiciously chosen" filterbank and of the nonlinearities with a "properly chosen" structure of activation functions in the hidden layer(s).

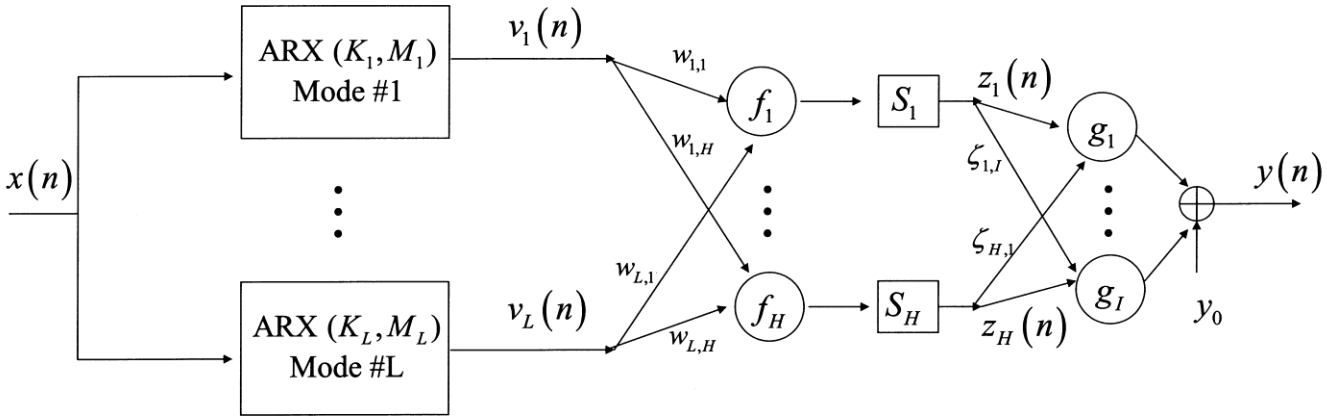


Figure 4.48
Schematic of the VWM model. Each input-preprocessing Mode # l ($l = 1,...,L$) is an Auto-Regressive with eXogenous variable (ARX) difference equation of order (K_l, M_l), the activation functions $\{f_h\}$ and $\{g_i\}$ are polynomials of degree Q_h and R_i respectively as shown in Equations (4.182) and (4.184), $\{S_h\}$ are sigmoidal functions given by Equation (4.183), and the weights $\{w_{l,h}\}$ and $\{\zeta_{h,i}\}$ are applied according to Equations (4.182) and (4.184) respectively.

This basic rationale is encapsulated in the proposed Volterra-Wiener-Marmarelis (VWM) model in a manner considered both general and efficient that obviates the need for a pre-selected basis in the filterbank and provides additional stability of operation with the use of a trainable compressive transformation cascaded with the polynomial activation functions. The overall structure of the VWM model exhibits close affinity with the Volterra-equivalent network models shown in Figure 4.36 and discussed in Section 4.2.2.

Specifically, the VWM model is composed of cascaded and lateral operators (both linear and nonlinear) forming a layered architecture, as shown in Figure 4.48 and described below.

The previously employed filterbank for input preprocessing is replaced in the VWM model with a set of linear difference equations of the form:

$$v_j(n) = \alpha_{j,1} v_j(n-1) + \ldots + \alpha_{j,K_j} v_j(n-K_j) + x(n) + \beta_{j,1} x(n-1) + \ldots + \beta_{j,M_j} x(n-M_j) \tag{4.181}$$

which define the "modes" of the system, where $x(n)$ is the input and $v_j(n)$ is the output of the jth mode. Note that $\beta_{j,0}$ is set equal to 1, because subsequent weighting of the mode outputs in Equation (4.182) makes these coefficients redundant. The modes defined in Equation (4.181) for $j = 1, \ldots, L$, serve as the input pre-processing filterbank but take the form of *trainable ARX models* instead of the pre-selected filterbank basis in the VEN architecture. The mode outputs are fed into the units of the first hidden layer (after proper weighting) and are transformed first polynomially and subsequently sigmoidally to generate the output of the hth hidden unit as:

$$z_h(n) = S_h\left\{\sum_{q=1}^{Q_h} c_{h,q}\left\{\sum_{j=1}^{L} w_{j,h} v_j(n)\right\}^q\right\} \tag{4.182}$$

where the sigmoidal transformation (denoted by S_h) is distinct for each hidden unit (with trainable slope λ_h and offset θ_h) and is given by the bipolar sigmoidal expression of the hyperbolic tangent:

$$S_h\{u\} = \frac{1-\exp\left[-\lambda_h(u-\theta_h)\right]}{1+\exp\left[-\lambda_h(u-\theta_h)\right]} \tag{4.183}$$

The outputs of these hidden units are combined linearly with weights $\{\zeta_{h,i}\}$ and entered into the units of a second hidden layer that will be termed the "*interaction layer*" for clarity of communication. Each "interaction unit" generates an output:

$$\psi_i(n) = \sum_{r=1}^{R_i} \gamma_{i,r} \left[\sum_{h=1}^{H} \zeta_{h,i} z_h(n) \right]^r \tag{4.184}$$

by means of a polynomial transformation of degree R_i.

The VWM model output is formed simply by summing the outputs of the interaction units and an output offset y_0:

$$y(n) = y_0 + \sum_{i=1}^{I} \psi_i(n) \tag{4.185}$$

No output weights are needed because the scaling of the contributions of the interaction units is absorbed by the coefficients $\{\gamma_{i,r}\}$.

The equivalence of the VWM model with the discrete Volterra model is evident from the discussion of Section 4.2.1. Note that the cascaded polynomial-sigmoidal nonlinear transformation in the hidden layer endows the VWM model with the potential capabilities of both polynomial and sigmoidal activation functions in a data-adaptive manner, so that the combined advantages can be secured. We expounded the advantages of polynomial activation function previously. The additional sigmoidal operation is meant to secure stability of bounded outputs in the hidden units (which is a potential drawback of polynomial transformations) and facilitate the convergence of the training process. However, it may not be needed, in which case the trained values of the slopes $\{\lambda_h\}$ become very small (being reduced to a linear scaling transformation) and the respective $\{\zeta_{h,i}\}$ values become commensurably large to compensate for the scaling of the small slope values. The key practical questions are: (1) how to select appropriately the structural parameters of this model; (2) how to estimate the model parameters from input-output data (via cost function minimization, since many parameters enter nonlinearly in this estimation problem).

The selection of the structural parameters follows the guidelines of the search procedure presented in Section 4.2.2. A simplification occurs when we accept uniform structure for the same type of units in each layer (i.e., $K_j = K$, $M_j = M$, $Q_h = Q$, $R_i = R$). Then the structural parameters of the VWM model are: L, K, M, H, Q, I, R. Certain additional simplifications are advisable in practice. For instance, K and M can be fixed to 4 or 6, since most physiological systems to date have exhibited modes with no more than two or three resonances (from the PDM analysis of actual physiological

systems). Although this is not asserted as a general rule and there are no guarantees of general applicability, one has to maintain a practicable methodological sense rooted in the accumulated experience. Naturally, if this rule appears inappropriate in a certain case, different values of K and M should be explored. Following the same reasoning, R can be limited to 2, since the purpose of the interaction units is to introduce the "cross-terms" missing in the "separable Volterra network" architecture and in achieving a better approximation of the multi-input static nonlinearity of the system. This is largely achieved by setting $R = 2$. Finally, Q can be set to 3 on the basis of the accumulated experience to date regarding the observable order of the actual physiological systems in a practical context. Recall that in the presence of the cascaded polynomial-sigmoidal transformations, the nonlinear order of the VWM model becomes infinite. However, lower order models are possible when the trained slopes of the sigmoidal transformations become very small (effectively a linear transformation with the dynamic range of the input). Note that the inclusion of the sigmoidal transformation (of trainable slope) at the outputs of the hidden units is motivated by the desire to avoid possible numerical instabilities (due to the polynomial transformations) by bounding the outputs of the hidden units. Because of the trainable slope of the sigmoidal functions, no such "bounding" is applied if not needed (corresponding to a very small value of λ_h, compensated by the trained values of $\zeta_{h,i}$).

With these practical simplifications in the structure of the VWM model, the remaining structural parameters are L, H and I. The model order selection procedure presented in Section 4.2.2 (utilizing the statistical criterion presented in Section 2.3.1) can be applied for these three structural parameters in the prescribed rightward ascending order (i.e., starting with $L = H = I = 1$, continue with increasing L up to $L_{\max}$ and then increase H and I in successively slower "looping speeds").

Having determined the structure of the VWM model, the parameter estimates are obtained via the training procedures discussed in the previous section by application of the chain rule of differentiation in error back-propagation. It should be emphasized that the training of the mode equations obviates the need for "judicious selection" of the input filterbank basis (a major practical advantage) and is expected to yield the minimum set of filters for input preprocessing based on guidance from the input-output data. One cannot ask for more--provided the training procedure converges properly.

The equivalent Volterra kernels for the VWM model can be obtained by expressing the sigmoidal functions in terms of Taylor series expansions or finite polynomial approximations within the range of their abscissa values. For instance, the analytical expression for the first-order Volterra kernel of the VWM model is:

$$k_1(m) = \sum_{i=1}^{I} \gamma_{i,1} \sum_{h=1}^{H} \zeta_{h,i} \alpha_{h,1}(\theta_h, \lambda_h) c_{h,1} \sum_{j=1}^{L} p_j(m) \tag{4.186}$$

where $p_h(m)$ denotes the hth PDM defined by Equation (4.187) and $\alpha_{h,1}$ is the first-order Taylor coefficient of S_h that depends on the respective slope λ_h and the offset θ_h. The resulting analytical expressions for the high-order kernels are rather complex in the general case and are omitted in the interest of space. Furthermore, their practical utility is marginal because the physiological interpretation of the VWM model ought to be based on the equivalent PDM model shown in Figure 4.1. Note that the hth PDM is given by:

$$p_h(m) = \sum_{j=1}^{L} w_{j,h} g_j(m) \tag{4.187}$$

where $g_j(m)$ denotes the impulse response function of the ARX (mode) Equation (4.181).

The number of free parameters in the VWM model is:

$$P = L(K+M) + H(L+Q) + I(H+R+3) + 1 \tag{4.188}$$

which indicates that the VWM model complexity is linear with respect to each structural parameter separately, although it is bilinear with respect to pairs of parameters. This fact is of fundamental practical importance, especially with regard to the nonlinear order Q and R, and constitutes the primary advantage (model compactness) of the use of the VWM model and generally Volterra-equivalent networks for nonlinear dynamic system modeling. In most actual applications, we expect $K = M = 4$, $Q = 3$, $R = 2$ and $L \le 4$, $H \le 3$, $I \le 2$. Therefore, the *maximum* number of free parameters in a practical context is expected to be about 70 (for $L = 4$, $H = 3$, $I = 2$). In the few cases where $K = M = 6$, the maximum number of free parameters becomes 86. In most cases we expect P to be between 30 and 50. This implies that effective modeling is possible with a few hundred input-output datapoints.

The structural parameter H is the most critical because it defines the number of "principal dynamic modes" (PDMs) in the Volterra-equivalent PDM model that is used for physiological interpretation (i.e., each hidden unit corresponds to a distinct PDM). A smaller number of PDMs facilitates the physiological interpretation of the VWM model and affects significantly the computational complexity of the VWM model because $\partial P / \partial H = L + Q + I$. Likewise, the computational complexity is affected significantly by the necessary number of modes L, because $\partial P / \partial L = K + M + H$. Each of the PDMs

corresponds to the linear combination of the VWM modes given by Equation (4.187), that defines the input entering into each hidden unit in the VWM model.

In closing, we note that the mode equations of the VWM model can be viewed also as "state equations", since the mode outputs $\{v_j(n)\}$ describe the dynamic state of the system at time n. This dynamic state is subject to a static nonlinear transformation to generate the system output. The nonlinear transformation is implemented (approximated) by means of the activation functions of the hidden and interaction layers in the combination prescribed by the VWM model structures of Figure 4.48.

Finally, we note that the VWM approach can be extended to the case of multiple inputs and multiple outputs with immense scalability advantages through the use of the interaction layer, as discussed in Chapter 7.