AN ABSTRACT OF THE THESIS OF

Veerabhadra Rao Karanam for the degree of Doctor of Philosophy
in Electrical and Computer Engineering presented on March 7, 1978

Title: FUNCTIONAL EXPANSIONS IN BILINEAR SYSTEM ANALYSIS

Abstract approved: (R. R. Mohler)

Bilinear systems, encountered in many important applications of engineering, biological and socio-economic systems, represent the logical step in complexity between the linear systems which are inadequate in the modeling of various dynamic processes and the difficult realm of nonlinear systems. This dissertation studies two functional expansions in bilinear system analysis--the Volterra series approach and the Walsh-function approach.

The Volterra series for bilinear systems is developed by means of the Reversion technique. The uniform convergence of the series is proved and estimates on the bound of the output of the system and on the error due to the truncation of the series are obtained. The above development leads to a direct synthesis of the Volterra kernels which characterize the bilinear system.
This result is related to Bush's canonic forms and it is shown that synthesized kernels facilitate the computation of the Volterra transfer functions. The notions of "weakly" bilinear, which are systems that can be represented by a finite Volterra series, and "strongly" bilinear, which cannot be so represented, are introduced. A method is proposed to estimate the parameters of the system using discrete Volterra kernels. A neutron kinetics model is included as an application of the method.

In the second approach, a finite orthonormal set of functions is used in the expansion of input/output functions for parameter estimation and input/parameter functions for output computation of deterministic systems. Walsh functions form the basis of the expansions as a consequence of their group properties with respect to binary algebra and of their relationship with the integrals of Walsh functions. This method is applicable to time-variant bilinear systems and to systems with polynomial type nonlinearities as well. The effectiveness of the proposed method for parameter estimation of deterministic systems subject to modeling and measurement noise is examined. Several computational examples are presented to demonstrate the accuracy of the method.
Functional Expansions in Bilinear System Analysis

by

Veerabhadra Rao Karanam

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Doctor of Philosophy

June 1973
APPROVED:

Redacted for Privacy

Professor of Electrical and Computer Engineering
in charge of major

Redacted for Privacy

Head of Department of Electrical and Computer Engineering

Redacted for Privacy

Dean of Graduate School

Date thesis is presented  March 7, 1978

Typed by Lyndalu Sikes for  Veerabhadra Rao Karanam
ACKNOWLEDGEMENTS

I wish to express my sincere gratitude to Professor Ronald R. Mohler for his continued encouragement, guidance and support throughout my graduate studies and in the preparation of this thesis. I am also indebted to Professor Pieter A. Frick for his advice and assistance in my research.

The distracting pressures are alleviated by my association with Mary Langlois whose encouragement during the crucial stages of this thesis is deeply appreciated. And, then, there are my friends and fellow students, who by their work, have stimulated my own work; I am especially thankful to Chin S. Hsu for many useful discussions and for his critical remarks.

Finally, I would like to thank my parents whose unfailing support throughout the years has enabled me to continually pursue my educational goals.

The research leading to this dissertation was supported by the National Science Foundation under Grant Number ENG. 77-07027.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>I. INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Motivation and Objective</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Bilinear System Representation</td>
<td>3</td>
</tr>
<tr>
<td>1.3 Importance of Bilinear Models</td>
<td>4</td>
</tr>
<tr>
<td>1.4 Examples of Bilinear Systems</td>
<td>9</td>
</tr>
<tr>
<td>1.5 Outline of Thesis</td>
<td>13</td>
</tr>
<tr>
<td>II. THE VOLterra SERIES APPROACH</td>
<td>15</td>
</tr>
<tr>
<td>2.1 Introduction</td>
<td>15</td>
</tr>
<tr>
<td>2.2 Functional Series Expansion</td>
<td>21</td>
</tr>
<tr>
<td>2.3 Convergence of the Series</td>
<td>26</td>
</tr>
<tr>
<td>2.4 Bounds on Output and Truncation Error</td>
<td>28</td>
</tr>
<tr>
<td>2.5 Synthesis of Kernels</td>
<td>32</td>
</tr>
<tr>
<td>2.6 Finite Volterra Series</td>
<td>39</td>
</tr>
<tr>
<td>2.7 Parameter Estimation via Discrete Volterra Kernels</td>
<td>46</td>
</tr>
<tr>
<td>2.8 Application to a Neutron Kinetics Model</td>
<td>50</td>
</tr>
<tr>
<td>III. THE WALSH-FUNCTION APPROACH</td>
<td>57</td>
</tr>
<tr>
<td>3.1 Introduction</td>
<td>57</td>
</tr>
<tr>
<td>3.2 Walsh Functions</td>
<td>59</td>
</tr>
<tr>
<td>3.3 Some Basic Results</td>
<td>63</td>
</tr>
<tr>
<td>3.4 Parameter Estimation with Perfect Information</td>
<td>69</td>
</tr>
<tr>
<td>3.5 Solution to Nonlinear Equations</td>
<td>74</td>
</tr>
<tr>
<td>3.6 Walsh-function Expansion for White Gaussian Noise</td>
<td>76</td>
</tr>
<tr>
<td>3.7 Computational Examples</td>
<td>80</td>
</tr>
<tr>
<td>3.8 Application to an Immunological Model</td>
<td>89</td>
</tr>
<tr>
<td>IV. CONCLUSIONS</td>
<td>93</td>
</tr>
</tbody>
</table>

BIBLIOGRAPHY

APPENDIX A: Review of some Lie Algebraic Concepts 102
APPENDIX B: Computation of Walsh Coefficients 104
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Bilinear state diagram.</td>
<td>5</td>
</tr>
<tr>
<td>2.1</td>
<td>Bilinear system diagram</td>
<td>33</td>
</tr>
<tr>
<td>2.2</td>
<td>Synthesized structure for second-order kernel.</td>
<td>33</td>
</tr>
<tr>
<td>2.3</td>
<td>Synthesized structure for third-order kernel.</td>
<td>33</td>
</tr>
<tr>
<td>2.4</td>
<td>Synthesized structure for fourth-order kernel.</td>
<td>33</td>
</tr>
<tr>
<td>2.5</td>
<td>Canonic second-order system.</td>
<td>34</td>
</tr>
<tr>
<td>2.6</td>
<td>Canonic third-order system.</td>
<td>34</td>
</tr>
<tr>
<td>2.7</td>
<td>First canonic fourth-order system.</td>
<td>34</td>
</tr>
<tr>
<td>2.8</td>
<td>Second canonic fourth-order system.</td>
<td>34</td>
</tr>
<tr>
<td>2.9</td>
<td>Feedback-free structure of an abelian bilinear system.</td>
<td>42</td>
</tr>
<tr>
<td>2.10</td>
<td>The factorable Volterra system $\mathcal{T}_k$.</td>
<td>45</td>
</tr>
<tr>
<td>3.1</td>
<td>Walsh functions.</td>
<td>61</td>
</tr>
<tr>
<td>3.2</td>
<td>Computed results for example 3.3.</td>
<td>86</td>
</tr>
<tr>
<td>Table</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
<td>------</td>
</tr>
<tr>
<td>3.1</td>
<td>Parameter estimates for example 3.1</td>
<td>82</td>
</tr>
<tr>
<td>3.2</td>
<td>Parameter estimates with various noise variances for example 3.4</td>
<td>89</td>
</tr>
</tbody>
</table>
FUNCTIONAL EXPANSIONS IN BILINEAR
SYSTEM ANALYSIS

I. INTRODUCTION

1.1 Motivation and Objective

It is often required to determine a mathematical model of a nonlinear system from the system input and output so that the model closely approximates the system. Functional expansions have played an important role in the study of such a problem, both when the internal structure of the system, that is, the state space description of the system, is known and when only the external description of the system, such as the input-output data, is available. This dissertation considers both situations using two types of functional expansions: (i) the Volterra series which is used to represent explicitly input-output relationship for a large class of nonlinear systems, and (ii) the Walsh functions which due to their complete orthonormal property form a basis for the expansions of input-output functions. The class of systems considered here is bilinear systems, the interest in which is due to the fact that they model a variety of physical processes and is due to their mathematical tractability [1] - [3].
The Volterra series representation, which is a generalization of the convolution integral representation of the linear systems, has proved useful in the analysis of nonlinear systems [4] - [6]. Considerable attention has been given to it in the study of bilinear systems [7] - [9] and some work [10] has been done on the Volterra kernels, which are generalized impulse response functions and which characterize the system. There is a need for an approach which would provide an insight into the structure of the system and that would lead to the synthesis of the Volterra kernels, when the equations that govern the system are known. It is the intent of the first part of this dissertation to achieve this goal.

The set of sine and cosine functions are commonly used for complete orthogonal expansions in engineering practice. One other class that has received increased attention recently in many areas of engineering such as communications, signal processing and in the analysis of linear systems [11] - [13] is the set of Walsh functions. Walsh functions are piecewise constant, binary valued periodic functions which form a complete orthonormal set on the interval [0, 1), and possess most of the properties of the sine and cosine set, but are more amenable to computations. The second part of the dissertation examines the use of Walsh function expansions in the analysis of a class of nonlinear systems—in particular, that of bilinear systems. Specifically, a parameter
estimation method is developed using the input-output data and a
method is given for computing the solution of nonlinear equations by
reducing them to algebraic equations.

1.2 Bilinear System Representation

A system is said to be bilinear if it is linear in state and in
control but not jointly linear in both [1]. Thus, a bilinear system
is described by the following differential equation

\[ \dot{x}(t) = Ax(t) + N(x(t), u(t)) + Bu(t) \]  \hspace{1cm} (1.1)

\[ y(t) = Cx(t), \quad x(t_0) = x_0 \]  \hspace{1cm} (1.2)

where

\[ x(t) \in \mathbb{R}^n \] is the state of the system,
\[ u(t) \in \mathbb{R}^m \] is the control,
\[ y(t) \in \mathbb{R}^p \] is the output of the system,
\[ \mathbb{R}^n, \mathbb{R}^m, \mathbb{R}^p \] are \( n, m, p \) dimensional real
Euclidean spaces respectively,
\[ x_0 \] is the initial state of the system,
\[ A \] is a \( n \times n \) matrix,
\[ B \] is a \( n \times m \) matrix,
\[ C \] is a \( p \times n \) matrix, and
\[ N(\cdot) \] is a bilinear mapping from \( \mathbb{R}^n \times \mathbb{R}^m \) into \( \mathbb{R}^n \).
The bilinear term in (1.1) is commonly written as

\[ N(x(t), u(t)) = \sum_{k=1}^{m} N_k u_k(t) x(t) \]  

(1.3)

where \( N_k \) is a \( n \times n \) matrix and \( u_k(t) \)'s are the components of the control vector \( u(t) \).

Thus, a bilinear system is generally represented by

\[ \dot{x}(t) = A x(t) + \sum_{k=1}^{m} N_k u_k(t) x(t) + B u(t) \]  

(1.4)

\[ y(t) = C x(t), \quad x(t_0) = x_0 \]

In general, the matrices \( A, N_k, B \) and \( C \) are functions of time.

It may be pointed out that with the systems of type described by (1.4), the controls \( u_k(t) \) can be suitably manipulated to achieve certain control objectives. Due to this property bilinear systems are also referred to as variable structure systems.

A schematic block diagram for the bilinear system of (1.4) is shown in Figure 1.1.

1.3 The Importance of Bilinear Systems

For a broad class of engineering problems, until recently, engineers were content with linear analyses. However, the dynamics of many physical processes cannot be described satisfactorily by linear differential equations. Bilinear systems represent the
Figure 1.1. Bilinear state diagram.

\[ N(x, u) = \sum_{i=1}^{m} \bar{N}_i u_i x \]
next logical step in complexity from linear systems to the difficult realm of nonlinear systems, and closely approximate many dynamical processes. Such bilinear models, to varying degrees of accuracy, include cell division and generation of antibodies [14], nuclear-fission and heat-transfer dynamics [15], mammalian water balance [16], human temperature regulation, carbon dioxide regulation in lungs [1], microbial cell growth in waste treatment and fermentation systems [17], and productivity and capital formation [18]. While the modeling of above mentioned processes by bilinear systems is due to practical considerations, what follows is a brief review of theoretical results which show that a large class of nonlinear systems may be approximated by bilinear systems to any desired accuracy.

Consider a broad class of nonlinear processes modeled by the equations

\[
\begin{align*}
\dot{x}_1(t) &= f_0(x_1(t)) + u_1(t)f_1(x_1(t)), \quad x_1(t) \in \mathbb{R}^n \\
y_1(t) &= g_0(x_1(t)) + u_1(t)g_1(x_1(t)), \quad y_1(t) \in \mathbb{R}^k
\end{align*}
\]

(1.5)

\[x_1(0) = 0, \quad |u_1(t)| \leq 1, \quad 0 \leq t \leq T\]

where the functions \(f_0, f_1, g_0, g_1\) are assumed to be as smooth as needed and \(u_1(t)\) is a scalar. Using a method referred to as
"bilinearization around a point", which involves expansion of the nonlinear functions in (1.5) around a point by Taylor's series and the use of a linearization technique due to Carleman, Krener [19] has shown recently that the nonlinear system (1.5) can be approximated by a time-invariant bilinear system

\[
\dot{x}(t) = A x(t) + u(t) N x(t) \quad x(t) \in \mathbb{R}^m \quad (m \geq n)
\]

\[
y(t) = C x(t) + u(t) D x(t) \quad y(t) \in \mathbb{R}^k
\]

\[
x(0) = [1, 0, \ldots, 0]^T, \quad |u(t)| \leq 1, \quad 0 \leq t \leq T
\]

where \(u(t)\) is a scalar such that the error in the approximation grows like an arbitrary power of \(t\). Similarly, by bilinearizing around a reference trajectory, it is also shown in [19] that, with \(g_1 = 0\), the system (1.5) may be approximated by a time-variant bilinear system.

More recently, Sussmann[20] has proved, using Stone-Weierstrass theorem, a more startling result which states that every input-output map \(F\) can be approximated as closely as one wishes by maps which arise from bilinear systems, provided that \(F\) satisfies "continuity" and "causality" conditions which may be explained as follows:

Let \(T > 0\) be a fixed time and let there be a fixed bound \(M > 0\) for inputs. With the initial state of the system fixed, one can associate the output \(F_u : [0, T] \rightarrow \mathbb{R}\) for every input \(u\) defined on...
on \([0, T]\). Then \(F\) satisfies

i) **continuity** if the functions \(F_{u_k}\) converge uniformly to \(F_u\) whenever a sequence \(\{u_k\}\) of inputs converges weakly to an input \(u\); and

ii) **causality** if the functions \(F_u\) and \(F_v\) coincide on \([0, T']\) whenever two inputs \(u\) and \(v\) coincide on \([0, T']\) for some \(T' \leq T\).

Now the result of Sussmann may be stated as follows:

**Approximation Theorem (Sussmann[20])**

Suppose \(F\) is an arbitrary function which to every input \(u = (u', \ldots, u^m)\) defined on \([0, T]\) and bounded by \(M\) assigns a curve \(F_u : [0, T] \rightarrow \mathbb{R}\). Suppose, moreover, that \(F\) satisfies the continuity and causality conditions. Then for every \(\varepsilon > 0\) there is a bilinear system whose corresponding input-output map \(\Phi\) is \(\varepsilon\)-close to \(F\), in the sense that

\[
|F_u(t) - \Phi_u(t)| < \varepsilon
\]

for all \(t \in [0, T]\) and all \(u : [0, T] \rightarrow \mathbb{R}^m\) which are measurable and bounded by \(M\).

It may be pointed out that the above theorem does not give a constructive method for finding the bilinear system approximating a given nonlinear system.
While the broad practical significance of the work reviewed above has yet to be proven, it does indicate a growing realization of the generality of bilinear systems.

1.4 Examples of Bilinear Systems

In this section, two examples are given which illustrate how bilinear systems arise in the modeling of complex processes.

1.4.1 Immunological Model

The model considered here is that of the immune system's generation of antibodies in disease control and is due to Mohler, et al. [14].

Briefly, bone marrow is the basic source of cells for the immune process. These cells are sensitized according to specific antigens (such as associated with a virus or other alien material). The sensitized cells with a specific antigen affinity or association constant k are called immunocompetent cells. These in turn differentiate and divide, respectively, to form plasma cells and memory cells. Plasma cells are the primary source of antibodies which have the particular affinity for the antigen recognized. Antibody binds with antigen to form immuno complex which is removed by macrophages. Consequently, it is the purpose of the humoral immune process to generate antibody which binds antigen for
effective destruction and removal of alien material. This represents a control process, the dynamics of which is derived from classical birth-death equations to take the following bilinear form:

\[ x = A x + (n_1 u_1 + n_2 u_2) x_1 + n_3 u_3 x_3 + \frac{b}{4} u_4 + \frac{b}{5} u_5 \]  

(1.7)

where the state vector, \( x \in \mathbb{R}^5 \), and the control vector \( u \in \mathbb{R}^5 \), have the components:

- \( x_1 \), the population density of immunocompetent cells (ICC),
- \( x_2 \), the population density of plasma cells,
- \( x_3 \), the population density of free antibody sites,
- \( x_4 \), the population density of immune complex which individually includes antibody site and antigen,
- \( x_5 \), antigen concentration which triggers the response,
- \( u_1 \), ICC multiplication factor,
- \( u_2 \), plasma cell multiplication factor,
- \( u_3 \), binding multiplication factor,
- \( u_4 \), stem-cell source rate (from bone marrow), and
- \( u_5 \), inoculation rate of antigen.
\[ A = \begin{bmatrix}
-1/\tau_1 & 0 & 0 & 0 & 0 \\
0 & -1/\tau_2 & 0 & 0 & 0 \\
\alpha'' & \alpha' & -1/\tau_3 & 0 & 0 \\
0 & 0 & 0 & -(c_k + 1/\tau_4) & 0 \\
0 & 0 & 0 & -\omega c_k & -1/\tau_5 
\end{bmatrix} \] (1.8)

\[ n_1 = [\alpha, 0, 0, 0, 0]^T, \]
\[ n_2 = [0, 2\alpha, 0, 0, 0]^T, \]
\[ n_3 = [0, 0, -c_k, c_k, -Nc_k]^T, \]
\[ b_4 = [1, 0, 0, 0, 0]^T, \]
\[ b_5 = [0, 0, 0, 0, 1]^T, \]

\( \alpha \) is the birthrate constant of stimulated ICC,

\( c_k \) is the dissociation rate,

\( \omega \) is a weighting constant

\( \alpha' \) is the plasma-cell antibody production rate,

\( \alpha'' \) is the ICC antibody production rate,

\( \tau_1 \) is the mean lifetime of ICC, and similarly,

\( \tau_2, \tau_3, \tau_4, \tau_5 \) are appropriate lifetimes.
The additive controls \( u_4 \) and \( u_5 \) are significant in immuno-therapy and in disease prevention by vaccination, respectively.

A simulation of this bilinear model with state feedback which compares favorably with experimental data is presented in [14].

1.4.2 Neutron Kinetics Model

The following model of a nuclear fission process in a power reactor, considered in [15], assumes that the neutron time variations are independent of spatial variations and that the neutron energy spectrum is independent of neutron level. The neutron kinetics model is described by the equations

\[
\begin{align*}
\dot{n} &= \frac{u (1 - \beta)}{\tau} n + \lambda c \\
\dot{c} &= \frac{u \beta}{\tau} n - \lambda c
\end{align*}
\]  

(1.9)  

(1.10)

where \( n \) is the neutron population (approximately proportional to reactor power generation), \( c \) is the population of the average precursor groups (unstable fission products which emit delayed neutrons during decay), \( \lambda \) is the average decay constant, \( \beta \) is the portion of neutrons generated by the precursors, and \( u \) is the neutron multiplication which controls the bilinear process.

If \( u = 1 \), the fission chain is self-sustaining, and the reactor is said to be critical. The reactor is said to be delayed-critical
when the rate of change of neutron population is zero. With delayed neutrons present the reactor is ordinarily operated near delayed-critical, with \( u \) approximately one. Then the neutron kinetics can be approximated by [14]

\[
\begin{align*}
\dot{n} &= \frac{\delta u - \beta}{\ell} \ n + \lambda \ c \quad (1.11) \\
\dot{c} &= \frac{\beta}{\ell} \ n - \lambda \ c \quad (1.12)
\end{align*}
\]

where \( \delta u \), reactivity, is \( u - 1 \). The above model can be used to estimate the power output of a reactor with changes in reactivity input influenced by control rod position. Since the reactivity, \( \delta u \), controls the process, the neutron kinetics model described by (1.11) and (1.12) is a bilinear model as is (1.9) and (1.10).

1.5 Outline of Thesis

The next chapter, Chapter II, is concerned with the study of Volterra series of bilinear systems. A new method of deriving the Volterra series for a bilinear system is developed and the uniform convergence of the series is proved. Estimates on the bound of the output of the system and on the error due to the truncation of the series are obtained. The synthesis of the Volterra kernels that characterize the bilinear system is considered, and it is shown that the synthesized structures facilitate the computation of the Volterra
transfer functions. The notions of "strongly" and "weakly" bilinear are introduced and examples are presented to illustrate these concepts. Using discrete Volterra kernels, a parameter estimation method is developed. The chapter is ended with the neutron kinetics model as an example utilizing the method.

In Chapter III, the use of Walsh functions in the identification of bilinear systems is considered. Using Walsh functions, a parameter estimation method for time-invariant as well as time-variant bilinear systems with perfect information is proposed. A computational method for the solution of bilinear equations is also presented. Some results and comments on the Walsh function expansion of white gaussian noise is included. The estimation method is used in various computational examples that include a scalar general variable-structure system, a second-order bilinear system, a time-variant scalar bilinear system and a scalar bilinear system subjected to some modeling error and measurement noise. The chapter is concluded with an application to an immunological model.

Chapter IV, the conclusion chapter, offers a summary of results presented in the thesis, along with concluding remarks. Suggestions for future research are also given.
II. THE VOLterra SERIES APPROACH

2.1 Introduction

Functional power series expansions were first used for general functional relationships in the late nineteenth century by the Italian mathematician Vito Volterra whose work on the series, which bears his name, is summarized in [21]. The Volterra series was introduced into engineering nonlinear analysis by Wiener [22] to express the input-output relation of a nonlinear system. A fair, but by no means complete, view of the work on this topic since the pioneering work of Wiener can be obtained from [23] - [29].

Consider a continuous nonlinear system operating with a scalar input $u(t)$ that results in a scalar output $y(t)$. The Volterra series description of such a system is

$$ y(t) = \sum_{k=1}^{\infty} y_k(t) = \sum_{k=1}^{\infty} \mathcal{H}_k[u(t)] $$

(2.1)

in which

$$ y_k(t) = \mathcal{H}_k[u(t)] = \int_{-\infty}^{\infty} \ldots \int_{-\infty}^{\infty} h_k(\tau_1, \ldots, \tau_k) \prod_{j=1}^{k} u(t - \tau_j) d\tau_j $$

(2.2)

Here, $\mathcal{H}_k$ are called the $k^{th}$-order Volterra operator and $h_k$ are the Volterra kernels that characterize the system. If $h_k$ satisfies
\[ h_k(\tau_1, \ldots, \tau_k) = h_k(\pi(1), \ldots, \pi(k)) \]

where \( \pi \) denotes any permutation of the \( \tau_i \)'s then \( h_k \) is called a symmetric kernel. For example, the second-order kernel is symmetric if \( h_2(\tau_1, \tau_2) = h_2(\tau_2, \tau_1) \). If such is not the case, then \( h_2(\tau_1, \tau_2) \) in (2.2) can be replaced by the symmetric second-order kernel \( (h_2(\tau_1, \tau_2) + h_2(\tau_2, \tau_1))/2! \) without changing the input-output relation. This procedure generalizes to \( h_k \) and so it is generally assumed that the kernels are symmetric. If \( h_k \) has the property that

\[ h_k(\tau_1, \ldots, \tau_k) = 0 \text{ for } \tau_{i+1} < \tau_i, \ i = 1, 2, \ldots, k-1. \]

then \( h_k \) is called a triangular kernel. For a non-anticipatory system the Volterra kernels are zero for any \( \tau_i < 0 \).

The first term, \( y_1(t) \), of the Volterra series (2.1) may be recognized as the convolution integral of a linear system. So, the Volterra kernels can be considered to be the generalization of impulse response function of a linear system for nonlinear systems [23]. Associated with the first-order kernel \( h_1(t) \), is its Laplace transform \( H_1(s) \) which is the transfer function of a linear system. Therefore, the \( k \)-dimensional Laplace transform of the \( k \)-th-order Volterra kernel is called the \( k \)-th-order transfer function and is
given by

\[ H_k(s_1, \ldots, s_k) = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} h_k(\tau_1, \ldots, \tau_k) e^{-s_1 \tau_1 - \ldots - s_k \tau_k} d\tau_1 \ldots d\tau_k \]  

(2.3)

In order to take advantage of these concepts it is necessary to consider the multidimensional time function

\[ y_k(t_1, \ldots, t_k) = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} h_k(t_1, \ldots, t_k) \sum_{j=1}^{k} u(t_j - \tau_j) d\tau_j \]  

(2.4)

which reduces to \( y_k(t) \) of (2.2) when \( t_1 = \ldots = t_k = t \). The Laplace transform of (2.4) is given by

\[ Y_k(s_1, \ldots, s_k) = H_k(s_1, \ldots, s_k) U(s_1) \ldots U(s_k) \]  

(2.5)

By applying the "association of variables" technique introduced by George [23] and subsequently modified and extended in [30] - [33], \( Y_k(s_1, \ldots, s_k) \) is reduced to \( Y_k(s) \), the transform of \( y_k(t) \). Thus, the nonlinear system is characterized by the Volterra kernels or, equivalently, by the Volterra transfer functions. Once either of these is known, it is possible to determine the output of the system for arbitrary inputs. The series representation of (2.1) assumes zero initial conditions. Flake [34] has shown that nonzero initial conditions can be treated as pseudo-forcing functions and the
Volterra series for the system can be modified accordingly.

A remark on the scope of the Volterra series representation of nonlinear systems is now in order. It has been shown that a continuous mapping from the space of continuous functions, defined on a finite interval, into the real numbers can be approximated to any desired accuracy by (2.1) by properly choosing the number of terms in the series and the Volterra kernels [35].

From an engineering point of view a major advantage of the Volterra series representation is that it has an explicit input-output relation of the system under consideration. This often enables one to see qualitative effects which may not be clear from the differential equation form. For example, with the series representation the whole problem need not be solved again for each different input in contrast to the differential equation representation which has an implicit input-output relation. Also, it allows one to think of the system in terms of functional blocks, a concept with which many engineers feel comfortable. Further, it facilitates the treatment of the problem of interaction of cascaded nonlinear systems, such as a two transistor amplifier [36].

The Volterra kernels are not of a sterile abstract nature that simply give the output of the system when used in the functional series. They give not only a complementary view of a system but provide a new way of looking at a structurally unknown system.
For example, Marmarelis and Naka [37], by inspecting the shape of the kernels of a neuron chain in the catfish retina, were able to derive some characteristics and the topology of the system, thus enabling them further insight into the workings of a neurophysiological system. Stark [38] showed that the kernels give a better understanding of the pupil as a neurological control system. In view of modeling of physiological systems involving cell dynamics, immune response, etc., by bilinear systems the importance of the study of Volterra series of bilinear systems is apparent.

In recent years considerable effort has been devoted to both the theoretical and practical aspects of the Volterra series representation of bilinear systems. Bruni, et al., have shown in [7] that for a general bilinear system the input-output map is a uniformly convergent Volterra series. DiAlessandro, et al., [9] have given a set of necessary and sufficient conditions for a Volterra series to have a bilinear realization. In a series of papers [8], [39], [40], Brockett has made significant contributions to the study of Volterra series for variable structure systems in general and for bilinear systems in particular, and set this field of study on a firm mathematical foundation by using recent developments in geometric system theory, with a particular emphasis on Lie algebra. In [8] he has shown that linear analytic systems, i.e. systems described by differential equations which are analytic in
the state and linear in the control, have solutions which can be expressed in a Volterra series provided that there is no finite escape time. This result is extended in [39] to systems characterized by differential equations whose right-hand sides have a nonlinear control dependence. Further, the Volterra kernels are used to find higher order necessary conditions in singular optimal control for a class of nonlinear systems. The problem of realization of Volterra series by finite and infinite dimensional bilinear systems is examined in [40]. Very recently, Gilbert [41] considered the problem of functional expansions for the response of nonlinear differential systems in an abstract mathematical setting.

In this chapter, the Volterra series for a bilinear system is developed by means of the Reversion technique in section 2.2. This approach enables one to write down the series in a simple and straightforward manner and gives insight into the structure of the system. Also, this method is applicable to the case of finite Volterra series representation whereas the method in [7] is shown below to be not appropriate. The uniform convergence of the Volterra series is proved in section 2.3. In section 2.4, estimates on the bound of the output of the system and on the error due to the truncation of the series are derived. In section 2.5, the development method of section 2.2 is used for the synthesis of the
Volterra kernels which characterize the bilinear system. This synthesis result is related to Bush's canonic forms for nonlinear systems. Further, the synthesized structures facilitate the computation of the Volterra transfer functions. The notions of "strongly" and "weakly" bilinear are introduced in section 2.6. It is shown that the former cannot admit a finite Volterra series whereas the later can be so represented. Examples are presented to illustrate the significance of these concepts. In section 2.7, a parameter estimation method using discrete Volterra kernels is developed. Finally, in section 2.8, the neutron kinetics model is considered as an example utilizing this method.

2.2 Functional Series Expansion

With the Reversion technique [42] (1.4) is written as

\[ \dot{x} = A x + \sum_{k=1}^{m} \Sigma N_k \mu u_k + B \mu u \]  

(2.6)

where \( \mu \) is a numerical parameter introduced to facilitate solution. Ultimately, \( \mu \) is allowed to become unity. A solution for the above equation is sought in the form

\[ x = \mu x_1 + \mu^2 x_2 + \ldots + \mu^i x_i + \ldots \]  

(2.7)

where \( x_1, x_2, \ldots \) are functions of time \( t \), to be determined.
Substituting (2.7) into (2.6) and then equating terms of successive powers of \( \mu \) results in the following set of recursive equations:

\[
\begin{align*}
\mu^1 & : \quad \dot{x}_1 = A x_1 + Bu, \quad x_1(0) = x(0) = x_0 \\
\mu^2 & : \quad \dot{x}_2 = A x_2 + \sum_{k=1}^{m} N_k x_{k-1} u_k, \quad x_2(0) = 0 \\
& \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \ quadrant
\[ x_2(t) = \sum_{k=1}^{m} \int_{0}^{t} e^{A(t-\tau_1)} N_k e^{A \tau_1} x_0 u_k(\tau_1) d\tau_1 \]

\[ + \sum_{k_1=1}^{m} \sum_{k_2=1}^{m} \int_{0}^{t} \int_{0}^{\tau_1} e^{A(t-\tau_1)} N_{k_1} e^{A(\tau_1-\tau_2)} N_{k_2} b_{k_2} u_{k_1}(\tau_1) u_{k_2}(\tau_2) d\tau_1 d\tau_2 \]

and, in general,

\[ x_{i+1}(t) = \sum_{l=k_1, \ldots, k_i}^{m} \int_{0}^{t} \int_{0}^{\tau_1} \cdots \int_{0}^{\tau_{i-1}} e^{A(t-\tau_1)} N_{k_1} e^{A(\tau_1-\tau_2)} N_{k_2} \cdots e^{A(\tau_{i-1}-\tau_i)} N_{k_i} b_{k_{i+1}} u_{k_l}(\tau_l) d\tau_l \]

consequently, the output vector \( y \) is obtained from (1.2) and (2.7) with \( \mu = 1 \):

\[ y = C x_1 + C x_2 + \cdots + C x_{i+1} + \cdots \]

\[ = C e^{At} x_0 + \int_{0}^{t} C e^{A(t-\tau)} B u(\tau) d\tau \]

\[ + \sum_{i=1}^{m} \sum_{l=k_1, \ldots, k_i}^{m} \int_{0}^{t} \int_{0}^{\tau_1} \cdots \int_{0}^{\tau_{i-1}} C e^{A(t-\tau_1)} N_{k_1} \cdots N_{k_i} \]
\[
e^{A\tau_1}x_0 \approx \sum_{i=1}^{\infty} \frac{1}{i!} u_k(\tau_i) d\tau_i + \sum_{i=1}^{\infty} \sum_{l=kl, \ldots, k_{i+1}}^m t^{\tau_{i+1}} \int_0^{\tau_i} \varepsilon_{i} d\varepsilon_i \]

(2.10)

\[
Ce^{A(t-\tau_1)}N_{k_1} \ldots N_{k_i} e^{A(\tau_1-\tau_{i+1})} b_{k_{i+1}} \int_0^{\tau_i} \sum_{\ell=1}^{i+1} u_k(\tau_\ell) d\tau_\ell
\]

which is the Volterra series expansion of the output of the bilinear system of (1.4).

When \( x_0 = 0 \), the Volterra kernels given by

\[
h_i(k_1, \ldots, k_i)(\tau_1, \ldots, \tau_i) = C e^{A(t-\tau_1)}N_{k_1} e^{A(\tau_1-\tau_2)}N_{k_2} \ldots \]

\[
N_{k_i} e^{A(\tau_i-\tau_{i-1})} b_{k_{i-1}}
\]

(2.11)

i = 1, 2, \ldots; k_i = 1, 2, \ldots, m; \tau_i \geq \tau_{i-1} \geq 0, \text{ characterize the bilinear system in that knowledge of these kernels provides the means for determining the output corresponding to a given input.}

The \( i^{th} \) order term in (2.10) with \( x_0 = 0 \) is an \( i \)-fold convolution integral containing the \( i^{th} \) order product of the input. Thus, the Volterra series (2.10) may be considered as a generalization of the convolution integral for bilinear systems.

The above method of developing the Volterra series has two advantages over that of Bruni, et al. [7]. First, equations (2.8) give rise to the bilinear system diagram which facilitates the synthesis of Volterra kernels, as will be seen in section 2.5.
Second, the above development method is applicable in the case of finite Volterra series whereas the method used in [7] is not appropriate, as shown below.

Recall the equations considered in [7] to develop the Volterra series of (1.4):

\[
\mathbf{x}_0 = A \mathbf{x}_0 + B \mathbf{u}, \quad \mathbf{x}_0(0) = \mathbf{x}(0) 
\]

\[
\mathbf{x}_{i+1} = A \mathbf{x}_{i+1} + \sum_{k=1}^{m} N_k \mathbf{u}_k \mathbf{x}_i + B \mathbf{u}, \quad \mathbf{x}_i(0) = \mathbf{x}(0), \ i = 1, 2, \ldots
\]

The state vector \( \mathbf{x} \) is given in [7] by

\[
\mathbf{x} = \mathbf{x}_0 + \sum_{i=1}^{\infty} (\mathbf{x}_i - \mathbf{x}_{i-1})
\]

Consider a bilinear system which has a \( p \)-term Volterra series representation (\( p \) : finite). Then, from (2.13),

\[
\mathbf{x} = \mathbf{x}_p
\]

But \( \mathbf{x} \) and \( \mathbf{x}_p \) are given by (1.4) and (2.12) as

\[
\mathbf{x} = A \mathbf{x} + \sum_{k=1}^{m} N_k \mathbf{x}_k \mathbf{u}_k + B \mathbf{u} \quad \text{and}
\]

\[
\mathbf{x}_p = A \mathbf{x}_p + \sum_{k=1}^{m} N_k \mathbf{x}_{p-1} \mathbf{u}_k + B \mathbf{u}
\]

This implies that \( \mathbf{x}_p = \mathbf{x} \) for \( \mathbf{x}_p \) which further implies that \( \mathbf{x} = \mathbf{x}_i \) for \( i = 1, 2, \ldots, p \). This means that (2.12) and (2.13) are not
appropriate for generating a finite Volterra series. By the method
developed in this section, the finite Volterra series is simply
obtained from
\[ x = \sum_{i=1}^{p} x_i \]
where \( x_i \) are given by (2.8).

The finite Volterra series cases are further considered in
section 2.6.

2.3 Convergence of the Series

In this section, the convergence of the Volterra series is
established. The proof is similar to the one given in [7]. To
prove the convergence, it is required to show that the right-hand
side of (2.7), where \( x_i \)'s are given by (2.8), is indeed the solution
to (1.4). It is assumed that \( x_0 = 0 \) to simplify matters; the time
interval of interest is compact, that is \( t \in [0, T] \); the matrices \( N_k, A, B \) and \( C \) are bounded; the control \( u(t) \in C [0, T] \), the space
of continuous functions defined on \([0, T]\). The norm \( \| x \| = \sum_{i=1}^{n} | x_i | \)
is used for \( x \in \mathbb{R}^n \). This induces the norm of \( n \times n \) matrix
\( A = [ a_{ij} ] \) equal to \( | A | = \max \sum_{j=1}^{n} | a_{ij} | \). Now it is shown that
\[ \sum_{j=1}^{i} x_j(t) \] converges to \( \bar{x}(t) \) as \( i \to \infty \).
Define
\[ z_1(t) = x(t) - \sum_{j=1}^{i} x_j(t) \text{ and} \]

\[ z_0(t) = x(t) \quad (2.14) \]

where \( x(t) \) is given from (1.4) as

\[ x(t) = \int_0^t e^{A(t-\tau)} \left\{ \sum_{k=1}^{m} N_k u_k(\tau)x(\tau) + \sum_{k=1}^{m} b_k u_k(\tau) d\tau \right\} \quad (2.15) \]

Using (2.14), (2.15) and (2.9),

\[ z_1(t) = x(t) - x_1(t) \]

or

\[ z_1(t) = \int_0^t e^{A(t-\tau)} \sum_{k=1}^{m} N_k u_k(\tau) x(\tau) \ d\tau \]

Since \( u(t) \in C [0, T] \) and \( A \) and \( N_k \) are assumed to be bounded, there exists a constant \( M < \infty \) such that

\[ |e^{A(t-\tau)} \sum_{k=1}^{m} N_k u_k(\tau)| < M \ \forall t, \tau \in [0, T] \]

Further, since \( x(t) \) is bounded on \([0, T]\), there exists a constant \( K > 0 \) such that

\[ |x(t)| \leq K \ \forall t \in [0, T] \]

Therefore,

\[ |z_1(t)| \leq Mk \int_0^t d\tau = Mt \]
Similarly,

\[ z_2(t) = x(t) - (x_1(t) + x_2(t)) \]

\[ = \int_0^t e^{A(t-\tau)} \sum_{k=1}^{m} N_k u_k(\tau) z_1(\tau) \, d\tau \]

hence

\[ |z_2(t)| \leq \frac{(Mt)^2}{2!} K \]

and, in general,

\[ |z_i(t)| \leq \frac{(Mt)^i}{i!} K \quad (i=0, 1, 2, \ldots) \quad (2.16) \]

The right-hand side of (2.16) goes to zero as \( i \to \infty \). This implies that, from (2.14), \( \sum_{j=1}^{\infty} x_j(t) \) is indeed the state vector \( x(t) \) of (1.4), thus establishing the uniform convergence of the Volterra series (2.10).

### 2.4 Bounds on Output and Truncation Error

Using the equations (2.8), an estimation on the bound of the output of the bilinear system is derived here in terms of the norms of the matrices associated with the system and various other parameters. The norms of the vectors and matrices are defined as in the previous section. For finite \( t \), it is well-known that

\[ e^{At} = \sum_{i=0}^{\infty} \frac{(At)^i}{i!} \]

Hence
\[
\| e^{At} \| \leq \sum_{i=0}^{\infty} \frac{\| A \|^i t^i}{i!}
\]

which gives the inequality, for \( \alpha = \| A \| \),

\[
\| e^{At} \| \leq e^{\alpha t}
\] (2.17)

Further, it is assumed that

\[
\max_k \| N_k \| \leq M_1
\]

\[
\max_k \| B_k \| \leq M_2
\] (2.18)

\[
\max_k \| u_k \| \leq \delta
\]

It follows from the above that

\[
\| B \| \leq M_2 \quad \text{and}
\]

\[
\| u \| \leq m\delta
\] (2.19)

With the first equation of (2.8) and the above inequalities the following differential inequality is obtained:

\[
\frac{d}{dt} \| \chi_1(t) \| \leq \alpha \| \chi_1(t) \| + m\delta M_2
\]

which implies that
Similarly, the second equation of (2.8) gives rise to the differential inequality

\[ \frac{d}{dt} |x_2(t)| \leq \alpha |x_2(t)| + m \varepsilon M_1 |x_1(t)| \]

Using (2.20), the above inequality results in

\[ |x_2(t)| \leq m \varepsilon M_1 |x_0| e^{\alpha t} + (m \varepsilon)^2 M_1 M_2 \frac{1}{\alpha} e^{\alpha t} \left( t + \frac{1}{\alpha} (e^{\alpha t} - 1) \right) \]

(2.21)

Since \( \alpha > 0 \) and \( e^{-\alpha t} < 1 \), (2.21) simplifies to

\[ |x_2(t)| \leq m \varepsilon M_1 |x_0| e^{\alpha t} + (m \varepsilon)^2 M_1 M_2 \frac{1}{\alpha} e^{\alpha t} t \]

(2.22)

In general,

\[ |x_{i+1}(t)| \leq \left( m \varepsilon M_1 \right)^i |x_0| e^{\alpha t} i! + (m \varepsilon)^{i+1} M_1 M_2 \frac{1}{\alpha} e^{\alpha t} \]

(2.23)

for \( i = 1, 2, \ldots \)

Also a bound on the output vector of (1.4) is obtained so that

\[ |y(t)| \leq \sum_{i=1}^{\infty} |C_i| |x_i(t)| \]

which, using (2.20), (2.22) and (2.23), results in
\[ \| y(t) \| \leq \| C \| \left( \| x_0 \| + \frac{1}{\alpha} M_2 \right) e^{(\alpha + m M_1) t} - \| C \| \frac{1}{\alpha} m M_2 \]

(2.24)

\[ t \in [0, T] \]

A bound on the error in the output due to the truncation of the Volterra series is derived below. Let the series be truncated after the \( r \)th term. Then

\[ \| \sum_{i=r+1}^{\infty} x_i(t) \| \leq \sum_{i=r+1}^{\infty} \| x_i(t) \| \leq \left( \| x_0 \| + \frac{1}{\alpha} M_2 \right) e^{\alpha t} \sum_{i=r}^{\infty} \frac{(m M_1 t)^i}{i!} \]

using (2.23)

\[ \leq \left( \| x_0 \| + \frac{1}{\alpha} M_2 \right) e^{(\alpha + m M_1) t} \frac{(m M_1 t)^r}{r!} , \]

\[ r \geq 1 \quad (2.25) \]

The last step follows from Taylor's theorem:

\[ e^\theta = \sum_{i=0}^{r-1} \frac{\theta^i}{i!} + e^\phi \frac{\theta^r}{r!} \]

for some \( \phi \) between 0 and \( \theta \); therefore

\[ e^\theta \sum_{i=0}^{r-1} \frac{\theta^i}{i!} = \sum_{i=r}^{\infty} \frac{\theta^i}{i!} = e^\phi \frac{\theta^r}{r!} \leq e^\phi \frac{\theta^r}{r!} \]
which is used to arrive at (2.25). Thus, the output truncation error, \( \hat{y}(t) \), is bounded by

\[
\| \hat{y}(t) \| = \left\| \sum_{i=r+1}^{\infty} C x_i(t) \| \leq C \left( \| x_0 \| + \frac{1}{a} m \| M \| e^{(a+m \| M \|)t} \right) \frac{(m \| M \|)^r}{r!},
\]

\[ r \geq 1 \]  

(2.26)

### 2.5 Synthesis of Kernels

Equations (2.8) give a clue as to the form in which the Volterra kernels can be synthesized as can be conceived from the bilinear system diagram (Figure 2.1) suggested by (2.8) for the case \( m = 1 \). It is readily seen that the \( n \)th order kernel is synthesized in a hierarchical manner as illustrated by Figures 2.2-2.4 for the second-, third- and fourth-order kernels.

Bush [43] has given canonic forms for kernels of nonlinear systems which can be synthesized by a finite number of linear systems and multipliers. For example, the second-order kernel can be synthesized as a product of two first-order kernels and has only one canonic form. This is shown in Figure 2.5. Similarly, the third order kernel has only one canonic form (Figure 2.6). A fourth-order kernel can be thought of as a product of either first- and third-order kernels, which results in the first canonic form (Figure 2.7) or two second-order kernels which is the second
Figure 2.1. Bilinear system diagram.

Figure 2.2. Synthesized structure for second-order kernel.

Figure 2.3. Synthesized structure for third-order kernel.

Figure 2.4. Synthesized structure for fourth-order kernel.
Figure 2.3. Canonic second-order system.

Figure 2.6. Canonic third-order system.

Figure 2.7. First canonic fourth-order system.

Figure 2.8. Second canonic fourth-order system.
canonic form (Figure 2.8). It is to be noted that while only single input cases have been considered here, extensions to the general case are apparent.

Consider the synthesized structure of the second-order Volterra kernel of a bilinear system (Figure 2.2). This realization is essentially the same as the one outlined by Bush for a second-order system (Figure 2.5), with $k_a(t) = e^{At}$, $k_b(t) = e^{At}$, and $k_c(t) = 1$. Similarly, for the third-order synthesized kernel (Figure 2.3) $k_d(t) = 1, k_2(\tau_1, \tau_2)$ is represented by $K_2$ and $k_e(t) = e^{At}$ (Figure 2.5). The synthesized fourth-order kernel for a bilinear system corresponds to only the first canonic form for fourth-order system, (Figure 2.7), given by Bush, with $k_3(\tau_1, \tau_2, \tau_3) = K_3$, $k_f(t) = 1$ and $k_g(t) = e^{At}$. This is in contrast to certain polynomial type nonlinear systems for which the synthesized fourth-order kernel is the sum of both Bush's canonic forms of the fourth-order systems [44].

The synthesized structures for the kernels facilitate the computation of the Volterra transfer functions of a bilinear system. An important reason for the desirability of the s-domain approach is that bilinear systems in the frequency domain could be handled similarly to linear systems and, hence, one can draw on the experience gained from the frequency domain studies of linear systems. Further, Harper and Rugh [45] have developed a theory of
reachability, observability and realization for factorable Volterra systems using the transfer functions.

George [23] has shown that the system operation $k_c \ast (k_b \cdot k_a)$ of Figure 2.5 has the associated transform $K_c (s_1 + s_2) K_b (s_1) K_a (s_2)$ where $K_a (s)$, $K_b (s)$ and $K_c (s)$ are the Laplace transforms of the kernels $k_a (t)$, $k_b (t)$ and $k_c (t)$. For the second-order kernel of a bilinear system, in Figure 2.2, it is seen that $k_a (t) = e^{At}$, $k_b (t) = 1$ and $k_c (t) = e^{At} N$. Since the Laplace transform of $e^{At}$ is $(sI - A)^{-1}$, it follows that $K_2$, the dotted block in Figure 2.2 has the associated transform

$$((s_1 + s_2) I - A)^{-1} \frac{N}{s_1} (s_2 I - A)^{-1} b$$

and, finally,

$$Y_2 (s_1, s_2) = C ((s_1 + s_2) I - A)^{-1} \frac{N}{s_1} (s_2 I - A)^{-1} b U(s_1) U(s_2)$$  \hspace{1cm} (2.27)

Thus, the second-order transfer function is given by

$$H_2 (s_1, s_2) = C ((s_1 + s_2) I - A)^{-1} \frac{N}{s_1} (s_2 I - A)^{-1} b$$  \hspace{1cm} (2.28)

With $k_1 (t) = e^{At}$ and its associated transform $K_1 (s) = (sI - A)^{-1}$, (2.28) can be written as

$$H_2 (s_1, s_2) = C K_1 (s_1 + s_2) \frac{N}{s_1} K_1 (s_2) b$$  \hspace{1cm} (2.29)
Similarly, it follows from Figure 2.3, that the third-order transfer function is given by
\[ H_3(s_1, s_2, s_3) = C((s_1 + s_2 + s_3)I - A)^{-1} \frac{N}{s_1} ((s_2 + s_3)I - A)^{-1} \frac{N}{s_2} (s_3I - A)^{-1} b \]

which can be rewritten as
\[ H_3(s_1, s_2, s_3) = CK_1(s_1 + s_2 + s_3) \frac{N}{s_1} K_1(s_2 + s_3) \frac{N}{s_2} K_1(s_3)b \]  
(2.30)

In general, the \( \ell \)th order transfer function is given by
\[ H_\ell(s_1, s_2, \ldots, s_\ell) = CK_1(s_1 + \ldots + s_\ell) \frac{N}{s_1} K_1(s_2 + \ldots + s_\ell) \frac{N}{s_2} \ldots \frac{N}{s_{\ell-1}} K_1(s_\ell)b \]  
(2.31)

The following example illustrates the computation of the second-order transfer function of the bilinear system (2.32), which is considered in [46]:
\[ \dot{x}(t) = A \dot{x}(t) + N \dot{x}(t)u(t) + b u(t) \]
\[ y(t) = c^T x(t), \quad x(0) = 0 \]

where
\[ A = \begin{bmatrix} -\lambda & 0 \\ 0 & -(\lambda + \alpha) \end{bmatrix}, \quad N = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \text{and} \quad c = \begin{bmatrix} 0 \\ a \end{bmatrix} \]

Writing (2.32) explicitly and, after some manipulation, the output of the system (2.32) can be obtained as
\[ y(t) = \int_0^t \int_0^\tau \begin{bmatrix} t \\ \tau \\ \end{bmatrix} e^{A(t-\tau)N} e^{A(\tau_1-\tau_2)} b u(\tau_1) u(\tau_2) d\tau_1 d\tau_2 \]  

(2.33)

which implies that the bilinear system (2.32) is completely characterized by the second-order Volterra kernel. In the s-domain, (2.33) results in

\[ Y(s_1, s_2) = H_2(s_1, s_2) U(s_1) U(s_2) \]  

(2.34)

and the second-order transfer function of the system (2.32), using (2.29), is given by

\[ H_2(s_1, s_2) = \begin{bmatrix} t \\ \tau \\ \end{bmatrix} (s_1 + s_2 + \lambda + \alpha) (s_2 + \lambda) \begin{bmatrix} a \\ \alpha \\ \end{bmatrix} \begin{bmatrix} s_1 \\ s_1 \end{bmatrix} (s_1 - b) \begin{bmatrix} a \\ \alpha \\ \end{bmatrix} \]  

which simplifies to

\[ H_2(s_1, s_2) = \frac{a}{(s_1 + s_2 + \lambda + \alpha)} \frac{a}{(s_2 + \lambda)} \frac{s_1}{s_1} \]  

(2.35)

\[ \lambda = -\alpha \quad (2.36) \]

Once the second-order transfer function is known, the output of the bilinear system (2.32) can be computed, for any input, using (2.34) and the "association of variables" technique mentioned in section 2.1.
2.6 Finite Volterra Series

Associated with a bilinear system, as seen in section 2.2, there is, in general, an infinite number of Volterra kernels. In practice, one can handle only a finite number of terms in the series which leads to the problem of truncating accuracy. Wiener [22] suggested that the first few terms of the series may be sufficient to represent the output of a nonlinear system if the nonlinearities are not too violent. Not much was done about this truncation problem other than to assume, without justification, that the first few terms in the series suffice [47].

Recently, Brockett [8] derived the Lie algebraic and associative algebraic conditions for a bilinear system to have a finite Volterra series representation. Based upon these results, in this section, the notions of "strongly" and "weakly" bilinear systems are introduced. To facilitate discussion leading to these concepts a theorem due to Brockett and a few useful lemmas are recalled. A brief review of relevant Lie algebra concepts is given in Appendix A.

* This section is borne out of discussions with C. S. Hsu.
**Theorem 2.1** (Brockett [8])

Suppose that

$$
\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + u(t)N\mathbf{x}(t) \\
y(t) = c^T\mathbf{x}(t)$$

is a minimal realization [9]. Then the Volterra series is finite if and only if $S = \{N, \text{ad}_{A}^{k}N\}_{A\in\mathbb{R}^{n}}$ is nilpotent where $\text{ad}_{A}^{k}(\cdot) = [A, [A, \ldots[A, \cdot] \ldots]]$ and $k = 0, 1, \ldots$

**Lemma 2.1** [48]

If $T$ is a $n \times n$ matrix, and $X$ is any solution of $[T, [T, X]] = 0$, then $[T, X]$ is nilpotent.

**Lemma 2.2** [48]

A Lie algebra $L$ is solvable if and only if $[L, L]$ is nilpotent.

**Lemma 2.3** [48]

A Lie algebra of real matrices $L$ is solvable if and only if there exists a non-singular matrix $P$ such that $P^{-1}A_{i}P$ is upper triangular for all $A_{i} \in L$. 
In the above, \( \{C, D\}_{AA} \) denotes the smallest associative algebra that contains matrices \( C \) and \( D \). An associative algebra is nilpotent if every element in it is nilpotent. Consider the following examples:

Example 2.1

Consider a commutative (abelian) bilinear system which is a special class of system (2.36) with the property that \([A, N] = AN - NA = 0\). Then, the ideal \( S = \{N\} \). Hence, such a system has a finite Volterra series representation, by Theorem 2.1, if and only if \( N \) is nilpotent. Since \( A \) and \( N \) commute, there exists a common modal matrix that simultaneously upper triangularizes \( A \) and \( N \) and since \( N \) is nilpotent for finite Volterra series, it is transformable to a strictly upper triangular matrix. Thus, a second-order abelian bilinear system with a finite Volterra series is described by the equations:

\[
\begin{bmatrix}
  x_1 \\
  x_2
\end{bmatrix} = \begin{bmatrix}
  a_{11} & a_{12} \\
  0 & a_{22}
\end{bmatrix} \begin{bmatrix}
  x_1 \\
  x_2
\end{bmatrix} + u(t) \begin{bmatrix}
  0 & n \\
  0 & 0
\end{bmatrix} \begin{bmatrix}
  x_1 \\
  x_2
\end{bmatrix}
\]

\[
y(t) = \begin{bmatrix}
  c_1 & c_2
\end{bmatrix} \begin{bmatrix}
  x_1 \\
  x_2
\end{bmatrix}
\]
It can be seen that this system can be represented as two linear systems in cascade (Figure 2.9). Higher-order systems can be treated similarly.

![Feedback-free structure of an abelian bilinear system](image)

*Figure 2.9. Feedback-free structure of an abelian bilinear system.*

**Example 2.2**

A slightly more general class of bilinear systems is the quasi-abelian bilinear systems which possess the property that 

\[ [A, N] \neq 0, \text{ but } [A, [A, N]] = 0 \text{ and } N[A, N] = [A, N] N = 0. \]

Then, the ideal \( S \) contains only two elements: \( S = \{N, [A, N]\} \).

Since \([A, [A, N]] = 0\), by lemma 2.1, \([A, N]\) must be nilpotent. For the system to have a finite Volterra series representation, by Theorem 2.1, \(N\) is nilpotent (since \(S\) is nilpotent) and hence it is transformable to a strictly upper triangular matrix. Using lemmas 2.2 and 2.3 it follows that \(A\) is upper triangular transformable. Thus, quasi-abelian bilinear systems, with a finite Volterra series representation, also admit a feedback-free
structure in the form of linear systems in cascade.

Extending the above arguments, it is apparent that in order for the bilinear system (2.36) to have a finite Volterra series representation, the matrices $N$ and $A$ should be transformable, respectively, to strictly upper and upper triangular matrices. An immediate consequence of this observation is that if $N$ is nonsingular, then the system (2.36) has an infinite Volterra series representation.

Therefore, it is appropriate to define and classify bilinear systems as follows: A bilinear system is strongly bilinear if it cannot be represented by a finite Volterra series; otherwise it is called weakly bilinear. Such a distinction has various significant implications. First, since a strongly bilinear system does not admit a finite-number-of-Volterra-kernels characterization, the conventional technique of approximating with the first few kernels needs to be justified. Partial results, as shown in section 2.4, provide a preliminary step. Second, weakly bilinear systems can be reduced into interconnection of linear subsystems for which the linear system theory can be advantageously applied. An interesting class of linear composite systems, called factorable Volterra systems and developed by Harper and Rugh [49] for approximating nonlinear systems, are closely related to weakly bilinear systems. This is briefly elaborated as follows:
The factorable Volterra system \( \tau_k \) (Figure 2.10) composed of \( k \) linear dynamic subsystems connected in parallel with the outputs multiplied (in the time domain) are defined by

\[
\dot{x}(t) = A x(t) + b u(t)
\]

\[
y(t) = \sum_{i=1}^{k} c_i^T x_i(t)
\]

where

\[
x(t) = \begin{bmatrix}
    x_1(t) \\
    x_2(t) \\
    \vdots \\
    x_k(t)
\end{bmatrix},
\]

\[
A = \begin{bmatrix}
    A_1 & & \\
    & A_2 & \\
    & & \ddots & \\
    & & & A_k
\end{bmatrix},
\]

\[
b = \begin{bmatrix}
    b_1 \\
    b_2 \\
    \vdots \\
    b_k
\end{bmatrix}
\]

i.e.,

\[
\dot{x}_i(t) = A_i x_i(t) + b_i u(t), \quad A_i \in \mathbb{R}^{n_i}, \quad b_i \in \mathbb{R}^{n_i}, \quad i = 1, \ldots, k
\]

\[
y_i(t) = c_i^T x_i(t)
\]

and the vector \( \bar{x}(t) \) is of dimension \( n = n_1 + \ldots + n_k \).
Figure 2.10. The factorable Volterra system $\pi_k$ [49].

It can be seen that the system $\pi_k$ can be represented by a finite Volterra series and also can be realized as a bilinear system by defining an additional state variable $x_{k+1}(t) = 1$. As an example, consider the $\pi_2$ system [49] with transfer functions

$$H_1(s) = \frac{1}{s+1} \quad \text{and} \quad H_2(s) = \frac{1}{s+2}$$

A bilinear system which has the same input/output behavior as the $\pi_2$ system, with $x_3(t) = 1$, is as follows:

$$\dot{x}(t) = A \cdot x(t) + u \cdot B \cdot x(t), \quad x(t) \in \mathbb{R}^3$$

$$y(t) = x_1(t) \cdot x_2(t) \cdot x_3(t) = x_1(t) \cdot x_2(t)$$

(2.38)
where

\[
A = \begin{bmatrix}
-1 & 0 & 0 \\
0 & -2 & 0 \\
0 & 0 & 0
\end{bmatrix}
\quad \text{and} \quad
B = \begin{bmatrix}
0 & 0 & 1 \\
0 & 0 & 0 \\
0 & 0 & 0
\end{bmatrix}
\]

This system is a weakly bilinear system since \( A \) is upper triangular and \( B \) is strictly upper triangular. Thus, it is apparent that \( \pi_k \) systems are a class of weakly bilinear systems.

### 2.7 Parameter Estimation via Discrete Volterra Kernels

So far, the Volterra series aspects of continuous bilinear systems have been considered. Discrete systems can be treated similarly; for example, for the discrete bilinear system described by the equations

\[
x(k+1) = A x(k) + \sum_{j=1}^{m} N_j u_j(k) x(k) + B u(k)
\]

\[
y(k+1) = C x(k+1)
\]

the Volterra series may be derived in an analogous way as in section 2.2, resulting in

\[
y(k) = \sum_{i=1}^{\infty} \sum_{j=1}^{m-1} \sum_{i_1=0}^{k-1} \sum_{i_2=1}^{k-1} \cdots \sum_{i_j=1}^{k-1} \sum_{j=1}^{\infty} \sum_{i_{j-1}=0}^{k-1} \sum_{i_{j-2}=1}^{k-1} \cdots \sum_{i_1=0}^{k-1} CA_{i_1N_1} A_{i_2-i_1-1N_j} \cdots
\]
where \( b_j \) are the column vectors of the matrix \( B \). The discrete Volterra kernels given by

\[
V^2(j_1, \ldots, j_2; k_1, \ldots, k_2) = \sum \binom{k_1}{j_1} A^{k_2-k_1-1} b_{j_2}
\]

\[k_2 > k_1 + 1; j_1, j_2, k_1, k_2 \geq 0\]  

\[k_2 > k_1 + 1; j_1, j_2, k_1, k_2 \geq 0\]  

It is assumed that the measurements for \( V_1(j, k) \) and \( V_2(j_1, j_2; k_1, k_2) \) are available and that \( C \) is a known scalar constant, say \( c \).

Let the matrix \( A \) be nonsingular and be in the canonical form:
By the Cayley-Hamilton theorem,

\[ A^n = a_n A^{n-1} + a_{n-1} A^{n-2} + \ldots + a_1 I \]  

(2.42)

Multiplying both sides of (2.42) by \( A^k \) and then, pre-multiplication and post-multiplication by \( c \) and \( b_j \), respectively, yields

\[ cA^{n+k} = a_n cA^{n+k-1} b_j + a_{n-1} cA^{n+k-2} b_j + \ldots + a_1 cA b_j \]  

(2.43)

With (2.41), (2.43) can be written as

\[ v_1(j; n+k) = a_n v_1(j; n+k-1) + a_{n-1} v_1(j; n+k-2) + \ldots + a_1 v_1(j; k) \]  

(2.44)

(2.44) can be written in a compact form:

\[ v_1(j; n+k) = U_1(j) \beta \]  

(2.45)

where
and \( \mathbf{a} = (a_1, \ldots, a_n)^T \), the parameter vector. Solving for \( \mathbf{a} \) in (2.45) results in

\[
\mathbf{a} = U_{-1}^{-1}(j) \mathbf{v}_{1}(j; n+k) \tag{2.47}
\]

This determines the matrix \( A \). Once \( A \) is known and assumed to be nonsingular, \( b_j \) can be found from (2.48):

\[
b_j = \frac{1}{c} A^{-k} \mathbf{v}_{1}(j; k) \quad j = 1, 2, \ldots, m \tag{2.48}
\]

Thus, the knowledge of the first-order kernel is sufficient to determine the matrices \( A \) and \( B \). The second-order kernel is utilized to determine \( N_j \), \( j = 1, \ldots, m \). From (2.41),

\[
\mathbf{v}_2(j_1, j_2; k_1, k_2) = A^{k_1} N_{j_1} \mathbf{v}_{1}(j_2; k_2-k_1-1), \quad k_2 \geq k_1 + 1
\]

which can be rewritten as

\[
N_{j_1} \mathbf{v}_2(j_2; k_2-k_1-1) = A^{-k_1} \mathbf{v}_2(j_1, j_2; k_1, k_2), \quad k_2 \geq k_1 + 1 \tag{2.49}
\]

With \( k_2 = 2k_1 + 1, \ldots, 2k_1 + n \) which, of course, satisfy the condition
\[ k_2 \geq k_1 + 1, \quad (2.49) \text{ takes the form} \]

\[
N_j \left[ v_1(j_2; k_1) \cdots v_1(j_2; k_1 + 1) \cdots v_1(j_2; n + k_1 - 1) \right]
\]

\[ = A^{-k} \left[ v_2(j_1, j_2; k_1, 2k_1 + 1) \cdots v_2(j_1, j_2; k_1, 2k_1 + 2) \cdots v_2(j_1, j_2; k_1, 2k_1 + n) \right] j_1, j_2 = 1, 2, \ldots, m \quad (2.50)\]

which can be solved for \( N_j \):

\[
N_j = A^{-k_1} U_2(j_1, j_2) U_1^{-1}(j_2), \quad j_1, j_2 = 1, 2, \ldots, m \quad (2.51)
\]

where \( U_2(j_1, j_2) \) is the matrix multiplying \( A^{-k_1} \) in (2.50). It is worth noting that \( U_1^{-1}(j) \) and \( A^{-k_1} \) are already available from the estimation of \( a \) (2.47) and \( b_j \) (2.48); thus no matrix inversions are needed in the estimation of matrices \( N_j, j=1, \ldots, m \).

It is seen from the above that, irrespective of the value of \( m \), the knowledge of the first-order and second-order kernels is sufficient for identifying the matrices \( A, N_j, (j=1, \ldots, m) \), and \( B \), when the matrix \( A \) is in the canonical form and is nonsingular. The example in the next section shows that reasonable estimates are obtained using the estimation method proposed in this section.

2.8 Application to a Neutron Kinetics Model

The neutron kinetics model discussed in section 1.4 is considered here as an application of the method of the previous
section. A normalized version of the model (1.11) and (1.12) is given by [48]

\[
\dot{x} = \bar{A} x + \bar{N} x u + \bar{b} u
\]

(2.52)

\[
y = \bar{c}^T x,
\]

\[
x(0) = 0,
\]

\[
|u| \leq 1
\]

with

\[
\bar{A} = \begin{bmatrix}
-a_1 & a_1 \\
-a_2 & a_2
\end{bmatrix},
\bar{N} = \begin{bmatrix}
n & 0 \\
0 & 0
\end{bmatrix},
\bar{b} = \begin{bmatrix}
b \\
0
\end{bmatrix},
\bar{c} = \begin{bmatrix}
0 \\
1
\end{bmatrix}
\]

(2.53)

where it is assumed for convenience that the precursor population is available as the output. Typical values for the parameters in (2.53) with second \(^{-1}\) as units are [48]

\[
a_1 = 7.5
\]

\[
n = 0.675
\]

\[
a_2 = 0.1
\]

\[
b = 0.675
\]

(2.54)

Since the estimation method of section 2.7 is developed for discrete bilinear systems, it is required to discretize the system (2.52).

For \(u(t)\) constant in the interval \(k \Delta T \leq t \leq (k+1) \Delta T\), where \(\Delta T\) is the sampling interval, the Euler discretization of (2.52) results in the discrete bilinear system [50]:

\[
\dot{x} = \begin{bmatrix}
-a_1 & a_1 \\
-a_2 & a_2
\end{bmatrix} x + \begin{bmatrix}
n & 0 \\
0 & 0
\end{bmatrix} u + \begin{bmatrix}
b \\
0
\end{bmatrix}
\]

\[
y = \begin{bmatrix}
0 \\
1
\end{bmatrix} x,
\]

\[
x(0) = 0,
\]

\[
|u| \leq 1
\]
\[ x(k+1) = A_1 x(k) + N_1 x(k) u(k) + b_1 u(k) \]  
(2.55)

\[ y(k) = c^T x(k), \quad x(0) = 0, \quad |u| \leq 1 \]

where

\[ A_1 = I + \Delta T \]
\[ N_1 = N \Delta T \]  
(2.56)

\[ b_1 = \Delta T \]

With

\[ z_1(k) = x_2(k) \text{ and} \]
\[ z_2(k) = x_2(k+1) \]

equation (2.55) can be written as

\[ z(k+1) = A z(k) + N z(k) u(k) + b u(k) \]  
(2.57)

\[ y(k) = c^T z(k), \quad z(0) = 0, \quad |u| \leq 1 \]

where

\[
A = \begin{bmatrix}
0 & 1 \\
\alpha_1 & \alpha_2
\end{bmatrix}, \quad
N = \begin{bmatrix}
0 & 0 \\
\beta_1 & \beta_2
\end{bmatrix}, \quad
b = \begin{bmatrix}
0 \\
1
\end{bmatrix}, \quad
c = \begin{bmatrix}
1 \\
0
\end{bmatrix}
\]  
(2.58)
\[ a_1 = -(1-a_1 \triangle T - a_2 \triangle T) \]

\[ a_2 = 1 - a_1 \]

\[ \beta_1 = -\beta_2 (1-a_2 \triangle T) \quad (2.59) \]

\[ \beta_2 = n \triangle T \]

\[ \gamma = a_2 b (\triangle T)^2 \]

It is to be noted that the matrix A of (2.57) is in the desired canonical form for the estimation method. However, c in (2.57) is a vector whereas the technique of the previous section assumed c to be a scalar. So some modifications of the method are made as follows:

Dropping the argument j which is equal to 1 for the scalar system (2.57) and with n = 2, (2.44) becomes

\[ v_1(k+2) = a_2 v_1(k+1) + a_1 v_1(k) \]

which for \( k = 0 \) and 1 results in

\[
\begin{bmatrix}
  a_1 \\
  a_2
\end{bmatrix}
= \begin{bmatrix}
  v_1(0) & v_1(1) \\
  v_1(1) & v_1(2)
\end{bmatrix}^{-1}
\begin{bmatrix}
  v_1(2) \\
  v_1(3)
\end{bmatrix}
\]

(2.60)

Also, the first-order discrete Volterra kernel

\[ v_1(k) = c^T A^k b, \quad k \geq 0 \quad (2.61) \]

for \( k = 0 \) and 1 gives
\[ v_1(0) = b_1 \]  
\[ v_1(1) = b_2 \]

where \( b_1 \) and \( b_2 \) are the components of \( b \). And, finally, the second-order discrete kernel

\[ v_2(k_1, k_2) = c^T A^{k_1} N A^{k_2-k_1-1} b, \quad k_2 \geq k_1 + 1 > 0 \]  
(2.63)

for \( k_1 = 0 \) and 1 leads to

\[ v_2(0, k_2) = (n_{11} n_{12}) A^{k_2-1} b, \quad k_2 \geq 1 \]
\[ v_2(1, k_2) = (n_{21} n_{22}) A^{k_2-2} b, \quad k_2 \geq 2 \]

which can be solved for \( N \), with \( k_2 = 1 \) and 2, as

\[
N = \begin{bmatrix}
v_2(0, 1) & v_2(0, 2) \\
v_2(1, 2) & v_2(1, 3)
\end{bmatrix} [b \ Ab]^{-1}
\]  
(2.64)

Thus, the matrix \( N \) is determined since \( A \) and \( b \) are found from (2.60) and (2.62). It can be seen that \([b \ Ab]\) is invertible if the linear part of the system (2.52) (i.e., (2.52) with \( N = 0 \)) is completely controllable. Moreover, this implies that the matrix \( \bar{A} \) of (2.52) and the matrix \( A_1 \) of (2.55) are transformable to canonical forms so that only \( n \) (here, \( n = 2 \)) entries of \( \bar{A} \) need be identified, which is to
be compared with the fact that only \( n \) unknowns of the matrix \( A \), the canonical form of \( A_1 \), is determined by the estimation method.

With (2.53) and (2.54) the required data is generated from the first-order and second-order kernels of (2.52):

\[
h_1(t) = \frac{T}{e} e^{\frac{At}{b}} = \frac{0.675}{e^{7.6}} \left(1 - e^{-7.6t}\right), \quad t \geq 0
\]

\[
h_2(t_1, t_2) = \frac{T}{e} e^{\frac{At_1}{b}} e^{\frac{A(t_2-t_1)}{b}}
\]

\[
= \left(\frac{0.675}{e^{7.6}}\right)^2 \left(1 - e^{-7.6t_2}\right) \left(1 + 75e^{-7.6(t_2-t_1)}\right), \quad t_2 \geq t_1 \geq 0
\]

With \( t_i = k_i \Delta T \) and for \( \Delta T \ll 1 \) and \( k_i \) such that \( t_i \ll 1 \), it can be seen from (2.65), (2.61) and (2.63) that

\[
\Delta T h_1(k_1 \Delta T) \sim v_1(k_1) \quad (2.66)
\]

\[
(\Delta T)^2 h_2(k_1 \Delta T, k_2 \Delta T) \sim v_2(k_1, k_2)
\]

Using (2.65) and (2.66) and with \( \Delta T = 0.001 \) the following values are obtained from (2.60), (2.62) and (2.64):

\[
\alpha_1 = -0.992431 \quad \alpha_2 = 1.992429
\]

\[
b_1 = 0 \quad b_2 = \gamma = 6.724415 \times 10^{-5}
\]

\[
\beta_1 = n_{21} = -0.66989 \quad \beta_2 = n_{22} = 0.66996 \quad (2.67)
\]

\[
n_{11} = n_{12} = 0
\]
Since $a_1$ and $a_2$ differ by a constant (see (2.59)), (2.67) can be used to solve for only one parameter $a_1$ or $a_2$. The average decay constant for the precursors $\lambda (=a_2)$ is generally available and is assumed to be 0.1/second. With this value for $a_2$, the following are obtained from (2.67) in conjunction with (2.59):

$$a_1 = 7.471$$

$$b = 0.672$$

$$n = 0.670$$

which compare favorably with the values in (2.54).
III. THE WALSH-FUNCTION APPROACH

3.1 Introduction

The basis for development in many areas of engineering is a system of sine and cosine functions. This is due to the fact that the complete and orthogonal properties of such a system facilitate frequency domain solutions to many control, communication and stochastic problems encountered in the theoretical and practical aspects of engineering design. With the widespread use of digital computers and processors in engineering design, Walsh functions which are characterized by only two states, thus matching the behavior of digital logic and yet possessing many of the attractive manipulative properties of the sine and cosine functions, found many important applications in such diverse areas as pattern recognition, signal and image processing, digital filtering, sequency multiplexing, interferometric spectroscopy, speech synthesis and analysis, etc. [11], [51]. Such a broad interest in practical applications has stimulated important contributions to a firm mathematical theory of Walsh functions. It is well known that the sine and cosine functions evolve as the characteristic solution to certain linear differential equations. Gibbs [52] has shown that the Walsh functions evolve as solutions to what are known as the logical differential equations. However, as is shown in this chapter,
Walsh functions are very useful in the analysis of bilinear systems which are described by ordinary differential equations.

In recent years, Walsh functions have been used in the identification and optimal control of linear time-invariant systems [12], [53], [54]. Chen and Hsiao [53] deal with the problem of determining a suitable internal structure for a linear time-invariant system from its input-output data. Such a structure is obtained using the method of repeated integration and the operational matrix concept relating Walsh functions to their integrals. A similar identification problem is considered in [54] with unknown initial states. The design of piecewise constant gains for optimal control of linear time-invariant systems is examined in [12].

In this chapter, the convenient group properties of Walsh functions are exploited so as to develop a parameter estimation method for a class of nonlinear systems with perfect information and specifically for variable-structure systems and systems with polynomial type nonlinearities. This development also permits a method for computing the output of the class of systems under consideration. For purposes of discussion and to illustrate the developed method, a bilinear system is studied as an example. Specifically, the system (1.4), recalled here, is considered.

\[
\begin{align*}
\dot{x}(t) &= A(t) x(t) + \sum_{k=1}^{q} N_k(t) x(t) u_k(t) + B(t) u(t) \quad (3.1) \\
y(t) &= c(t) x(t), \quad x(0) = x_0 \in \mathbb{R}^n \quad (3.2)
\end{align*}
\]
The matrices $A(t)$, $N_k(t)$, $B(t)$ are assumed to be Lebesque measurable and $c(t)$ is assumed to be a known scalar valued function. The bilinear system of (3.1) and (3.2) can be represented by

$$y(t) = y(0) + \sum_{i=1}^{t} a_i(t) y_i(\tau) d\tau + \sum_{k=1}^{t} \sum_{i=1}^{n} n_k(t) y_i(t) u_k(t) d\tau$$

$$+ \sum_{j=1}^{q} b_j(t) c(t) u_j(t) d\tau$$

where $a_i$, $n_k$, and $b_j$ are the column vectors of matrices $A$, $N_k$ and $B$ respectively, and $y_i$ is the $i^{th}$ component of the output vector $y$.

The next section contains a brief review of the Walsh functions and their relevant properties.

3.2 The Walsh Functions

Following Fine [55] it is convenient to define Walsh functions in terms of Rademacher functions. The Rademacher functions $\{\varnothing_{m}\}$ form an incomplete set of orthogonal functions on $[0, 1]$ and are defined by

$$\varnothing_0(t) = \begin{cases} 1 & 0 \leq t < 1/2 \\ -1 & 1/2 \leq t < 1 \end{cases}$$

$$\varnothing_0(t+1) = \varnothing_0(t)$$

$$\varnothing_m(t) = \varnothing_0(2^m t) \ (m = 1, 2, \ldots)$$
and the Walsh functions \( \{ \psi_m \} \) are then obtained by defining

\[
\psi_0(t) = 1
\]

\[
\psi_m(t) = \phi_{m_1}(t) \phi_{m_2}(t) \cdots \phi_{m_r}(t)
\]

for \( m = 2^{m_1} + 2^{m_2} + \cdots + 2^{m_r} \), where the integers \( m_i \geq 0 \) are uniquely determined by \( m_{i+1} < m_i \). The first 16 Walsh functions are shown in Figure 3.1.

The Walsh functions form an orthonormal system on \( 0 < t < 1 [11] \). Thus

\[
\int_0^1 \psi_\varepsilon(t) \psi_m(t) \, dt = \begin{cases} 0, & \varepsilon \neq m \\ 1, & \varepsilon = m \end{cases}
\]

Any Lebesgue measurable function \( f(t) \) on \( [0, 1) \) may be represented by the Walsh series in the form:

\[
f(t) = a_0 \psi_0(t) + a_1 \psi_1(t) + \cdots + a_m \psi_m(t) + \cdots
\]

where the coefficients \( a_m \) are given by

\[
a_m = \frac{1}{2^m} \int_0^1 \psi_m(t) f(t) \, dt \quad (m=0,1,2,\ldots)
\]

The Walsh functions form a complete system since the mean-square error converges to zero with increasing number of terms, viz.
Figure 3.1. Walsh Functions.
\[
\lim_{m \to \infty} \int_{0}^{1} (f(t) - \sum_{i=0}^{m-1} a_i \psi_i(t))^2 \, dt = 0
\]

Fine [55] showed that if \( f(t) \) is continuous on \( 0 \leq t < 1 \), the series (3.7) converges uniformly to \( f(t) \) if the terms are grouped so that each group contains all of the Walsh functions designated by a given number of binary digits. If \( f(t) \) is not continuous, there is convergence in the mean. If the Walsh series is truncated at the end of a group represented by all of the terms given by \( m \) binary digits, the partial sum of the first \( 2^m \) terms of the series equals the average value of the function in each subinterval of length \( 2^{-m} \).

Two other properties of Walsh functions, particularly attractive for the purpose of this chapter, are the following:


Specifically, they are closed under multiplication, with the multiplication rule for Walsh functions expressed as

\[
\psi_\ell(t) \psi_m(t) = \psi_{\ell \oplus m}(t)
\]

(3.9)

The sign \( \oplus \) represents no carry modulo-2 addition. That is \( \ell \) and \( m \) are written as binary numbers and then they are added according to the rule \( 0 \oplus 1 = 1 \), \( 0 \oplus 0 = 0 \) and \( 1 \oplus 1 = 0 \) (no carry) e.g., \( 5 \oplus 7 = 2 \).

2) Fine [55] also showed that the integral of a Walsh function
may be represented in terms of Walsh functions. For an integral of the form

\[ J_k(t) = \int_0^t \psi_k(x) \, dx \quad (k = 0, 1, 2, \ldots) \quad (3.10) \]

Fine showed that, with \( k = 2^m + k', 0 \leq k' \leq 2^m \),

\[
J_0(t) = \frac{1}{2} \psi_0(t) - \frac{1}{4} \sum_{p=0}^{\infty} 2^{-p} \psi_{2^p}(t) \quad \text{and} \quad J_k(t) = 2^{-(m+2)} \{ \psi_{k'}(t) - \sum_{p=1}^{\infty} 2^{-p} \psi_{2^{m+p+k}}(t) \} \quad \text{for } k > 1 \quad (3.11)
\]

3.3 Some Basic Results

In this section, some mathematical tools which are used in the subsequent sections, are developed.

Equation (3.11) may be written in a compact matrix form.

Taking the terms containing the first four Walsh functions (that is, \( \psi_0, \psi_1, \psi_2, \text{ and } \psi_3 \)) \( J_0(t) \) is approximated by

\[
J_0(t) \approx \frac{1}{2} \psi_0(t) - \frac{1}{4} \psi_1(t) - \frac{1}{8} \psi_2(t) \quad (3.12)
\]

Let \( \psi_{(4)}^T(t) = [\psi_0(t), \psi_1(t), \psi_2(t), \psi_3(t)] \). Then (3.12) can be written as

\[
J_0(t) \approx \begin{bmatrix} \frac{1}{2} & -\frac{1}{4} & -\frac{1}{8} & 0 \end{bmatrix} \psi_{(4)}(t) \quad (3.13)
\]
Similarly, from (3.11) it can be seen that

\[ \mathbf{J}_1(t) \sim \begin{bmatrix} \frac{1}{4} & 0 & 0 & -\frac{1}{8} \end{bmatrix} \psi(4)(t) \]

\[ \mathbf{J}_2(t) \sim \begin{bmatrix} \frac{1}{8} & 0 & 0 & 0 \end{bmatrix} \psi(4)(t) \quad (3.14) \]

and \[ \mathbf{J}_3(t) \sim \begin{bmatrix} 0 & \frac{1}{8} & 0 & 0 \end{bmatrix} \psi(4)(t) \]

Combining (3.13) and (3.14), and writing \( J_1' \)'s explicitly,

\[
\begin{bmatrix}
\int_0^t \psi_0(x) \, dx \\
\int_0^t \psi_1(x) \, dx \\
\int_0^t \psi_2(x) \, dx \\
\int_0^t \psi_3(x) \, dx
\end{bmatrix}
\begin{bmatrix}
\frac{1}{2} & -\frac{1}{4} & -\frac{1}{8} & 0 \\
\frac{1}{4} & 0 & 0 & -\frac{1}{8} \\
\frac{1}{8} & 0 & 0 & 0 \\
0 & \frac{1}{8} & 0 & 0
\end{bmatrix}
\begin{bmatrix}
\psi_0(t) \\
\psi_1(t) \\
\psi_2(t) \\
\psi_3(t)
\end{bmatrix}
\]

or in matrix form

\[ \int_0^t \psi(4)(x) \, dx \sim \mathbf{P}(4) \psi(4)(t) \quad (3.15) \]

and in general
\[
\int_{0}^{t} \psi(x) \, dx = P \psi(t)
\]  
(3.16)

where \( \psi(x) \) is a column vector consisting of Walsh functions and the matrix \( P \) is termed "operational matrix" in [53]. \( \psi \) and \( P \) in (3.16) are of infinite dimension, but finite dimensional approximations are obtained by restricting \( \psi \) to contain only \( 2^n \) Walsh functions for some integer \( n \geq 1 \). The resulting square operational matrix \( P \) is, therefore, of order \( m = 2^n \) and (3.16) is replaced by

\[
\int_{0}^{t} \psi_{(m)}(x) \, dx \approx P_{(m)} \psi_{(m)}(t)
\]  
(3.17)

where the subscript \( m \) denotes the number of Walsh functions being used. A general expression for the operational matrix \( P \) of an arbitrary order \( m \) can be obtained and is given by

\[
P_{(1)} = \frac{1}{2}; \quad P_{(m)} = \begin{bmatrix}
P_{(\frac{m}{2})} & \cdots & \frac{1}{2m} & I_{(\frac{m}{2})} \\
\cdots & \cdots & \cdots & \cdots \\
\frac{1}{2m} & I_{(\frac{m}{2})} & \cdots & 0 \\
I_{(\frac{m}{2})} & 0 & \cdots & \frac{m}{2}
\end{bmatrix}
\]  
(3.18)

Here, \( I_{(\frac{m}{2})} \) and \( 0_{(\frac{m}{2})} \) denote the identity and null matrices of order \( \frac{m}{2} \), respectively.
Next, the outer product of the Walsh function vector $\psi_{(m)}(t)$ is examined. It is now shown that the matrix

$$M_{(m)}(t) = \psi_{(m)}(t) \psi_{(m)}^T(t) \quad t \in [0, 1)$$

(3.19)

is reducible to a simple algebraic form. Writing the vector $\psi_{(m)}(t)$ in terms of its components and carrying out the multiplication in left-hand side of the above equation, $M_{(m)}(t)$ can be written as

$$M_{m}(t) =
\begin{bmatrix}
\psi_0(t)\psi_0(t) & \psi_0(t)\psi_1(t) & \cdots & \psi_0(t)\psi_{m-1}(t) \\
\psi_1(t)\psi_0(t) & \psi_1(t)\psi_1(t) & \cdots & \psi_1(t)\psi_{m-1}(t) \\
.. & .. & .. & .. \\
\psi_{m-1}(t)\psi_0(t) & \psi_{m-1}(t)\psi_1(t) & \cdots & \psi_{m-1}(t)\psi_{m-1}(t)
\end{bmatrix}
$$

Employing (3.9) the multiplication rule for Walsh functions, the above matrix simplifies to the form

$$M_{m}(t) =
\begin{bmatrix}
\psi_0(t) & \psi_1(t) & \cdots & \psi_{m-1}(t) \\
\psi_1(t) & \psi_0(t) & \cdots & \psi_{m-2}(t) \\
.. & .. & .. & .. \\
\psi_{m-1}(t) & \psi_{m-2}(t) & \cdots & \psi_0(t)
\end{bmatrix}
$$

(3.20)
It is seen that the first column of the matrix $M_{(m)}(t)$ is $\psi_{(m)}(t)$, and the remaining columns are permutations of the first column elements. $M_{(m)}(t)$ can be represented in the composite matrix notation:

$$
M_{(m)}(t) = [ \psi_{(m)}(t), \Lambda_{1}^{(m)} \psi_{(m)}(t), \ldots, \Lambda_{m-1}^{(m)} \psi_{(m)}(t) ]
$$

where, for example, when $m = 4$.

$$
\Lambda_{1}^{(4)} = \begin{bmatrix}
0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0
\end{bmatrix}, \quad \Lambda_{2}^{(4)} = \begin{bmatrix}
0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0
\end{bmatrix}
$$

and

$$
\Lambda_{3}^{(4)} = \begin{bmatrix}
0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0
\end{bmatrix}
$$

In general, the $(m)$ matrices can be obtained from

$$
\Lambda_{i}^{(m)} = \begin{bmatrix}
\Lambda_{\frac{m}{2}}^{(i)} & \vdots & 0 & \frac{m}{2} \\
\vdots & \ddots & \ddots & \vdots \\
0 & \ldots & \Lambda_{\frac{m}{2}}^{(i)} & \frac{m}{2} \\
\frac{m}{2} & \ldots & \Lambda_{i}^{(m)} & \frac{m}{2}
\end{bmatrix}, \quad \text{and} \quad \Lambda_{i}^{(m)} = \begin{bmatrix}
0 & \frac{m}{2} \\
\frac{m}{2} & \ldots & \Lambda_{i}^{(m)} & \frac{m}{2} \\
\Lambda_{i}^{(m)} & \ldots & 0 & \frac{m}{2}
\end{bmatrix}
$$

for $i = 0, 1, 2, \ldots, \left(\frac{m}{2}\right) - 1$ and

$$
\Lambda_{0}^{(m)} = \begin{bmatrix}
0 & \frac{m}{2} \\
\frac{m}{2} & \ldots & \Lambda_{0}^{(m)} & \frac{m}{2} \\
\Lambda_{\frac{m}{2}}^{(m)} & \ldots & 0 & \frac{m}{2}
\end{bmatrix}
$$
Some interesting properties of $\Lambda_i$ matrices are established in the following three lemmas:

**Lemma 3.1**

$$\Lambda_i^{(m)} \Lambda_j^{(m)} = \Lambda_i^{(m)} \Lambda_j^{(m)}$$

Proof by induction. The lemma is trivially true for $m = 1$. For $m = 2$ four cases arise depending on whether $i = 0$ or 1 and $j = 0$ or 1.

Consider, for example, the case $i = j = 0$ giving

$$\Lambda_i^{(2)} \Lambda_j^{(2)} = \begin{bmatrix} \Lambda_i^{(1)} & 0^{(1)} \\ 0^{(1)} & \Lambda_i^{(1)} \end{bmatrix} \begin{bmatrix} \Lambda_j^{(1)} & 0^{(1)} \\ 0^{(1)} & \Lambda_j^{(1)} \end{bmatrix} = \begin{bmatrix} \Lambda_i^{(1)} & 0^{(1)} \\ 0^{(1)} & \Lambda_i^{(1)} \end{bmatrix}$$

$$= \Lambda_i^{(1)} \circ_j \text{ by (3.22)}$$

A similar simple calculation can be performed in the other three cases. Assume the lemma holds for $\frac{m}{2}$, $m = 2^n$ for some $n > 1$ (having already established this for $m = 2$) then, for $i < \frac{m}{2}$ and $j \geq \frac{m}{2}$ say $j = \ell + \frac{m}{2}$ where $\ell > 0$,

$$\Lambda_i^{(m)} \Lambda_j^{(m)} = \begin{bmatrix} \Lambda_i^{(\frac{m}{2})} & 0^{(\frac{m}{2})} \\ 0^{(\frac{m}{2})} & \Lambda_i^{(\frac{m}{2})} \end{bmatrix} \begin{bmatrix} 0^{(\frac{m}{2})} & \Lambda_i^{(\frac{m}{2})} \\ \Lambda_i^{(\frac{m}{2})} & 0^{(\frac{m}{2})} \end{bmatrix} = \begin{bmatrix} 0^{(\frac{m}{2})} & \Lambda_i^{(\frac{m}{2})} \\ \Lambda_i^{(\frac{m}{2})} & 0^{(\frac{m}{2})} \end{bmatrix}$$

$$= \Lambda_i^{(\frac{m}{2})} \circ_j \text{ by (3.22)}$$

Thus the lemma is true for all integer powers of 2.
Lemma 3.2 \( \Lambda^{(m)}_i \ (i = 0, 1, 2, \ldots, (m-1)) \) form an Abelian group under the multiplication operation.

Proof: Associativity holds for matrix multiplication. The identity element \( \Lambda^{(m)}_0 \) exists such that \( \Lambda^{(m)}_i \Lambda^{(m)}_0 = \Lambda^{(m)}_0 \Lambda^{(m)}_i = \Lambda^{(m)}_i \) for all \( i \). The inverse of each \( \Lambda^{(m)}_i \) is itself since

\[ \Lambda^{(m)}_i \Lambda^{(m)}_i = \Lambda^{(m)}_0 = \Lambda^{(m)}_0 \] using lemma 3.1.

This proves lemma 3.2.

Lemma 3.3 \( \sum_{i=1}^{2^{k-1}} \Lambda^{(m)}_i = \Lambda^{(m)}_0 \) for all \( k > 1 \)

Proof follows trivially from lemma 3.1 and the fact that \( 1 + 2 + 3 \ldots + (2^k - 1) = 0 \) for all \( k > 1 \).

3.4 Parameter Estimation with Perfect Information

In this section a method is developed for estimating the parameter vector-valued functions \( a_{i\tau}(t), n_{i\tau}(t) \) for \( i = 1, 2, \ldots, n \); and \( b_j(t) \) for \( j = 1, 2, \ldots, q \) of (3.3).

Employing \( m = 2^n \) (for some integer \( n \)) Walsh functions in \( \psi \), \( a_{i\tau}(t), b_{i\tau}(t), n_{i\tau}(t) \) and \( c(t) \) are approximated by their Walsh function expansions,
\[ a_i(t) \sim F_i \psi(m)(t) \quad b_j(t) \sim H_j \psi(m)(t) \]

\[ n_{k_i}(t) \sim G_{k_i} \psi(m)(t) \quad c(t) \sim \psi^T(m)(t) \]

thus reducing the identification problem to identifying \( F_i, G_{k_i}, \) and \( H_j \).

The known input \( u \) and the output \( y \) functions similarly are approximated by their Walsh function expansions

\[ y_i(t) \sim \frac{T}{e_i} \psi(m)(t) = \psi(m)(t)e_i \]

\[ u_k(t) \sim \frac{z_k}{e_i} \psi(m)(t) = \psi(m)(t)z_k \]

Substitution of (3.23) and (3.24) into (3.3) leads to

\[ E \psi(m)(t) \sim y(0) + \sum_{i=1}^{n} F_i \int_0^t \psi(m)(\tau) \psi^T(m)(\tau) d\tau e_i + \sum_{k=1}^{q} \sum_{i=1}^{n} G_{k_i} \int_0^t \psi(m)(\tau) \psi^T(m)(\tau) e_i z_k \psi^T(m)(\tau) d\tau \]

\[ + \sum_{k=1}^{q} H_k \int_0^t \psi(m)(\tau) \psi^T(m)(\tau) z_k \psi(m)(\tau) d\tau \]

where \( E \) represents the matrix formed by \( e_i^T \) as row vectors.
To fully exploit the simple algebraic structure outlined above in the system representation of (3.25) the following two lemmas are established:

**Lemma 3.4**

\[ \psi(m)^T \alpha = \Lambda_{\alpha} \psi(m) \]

where \( \Lambda_{\alpha} = [\alpha, \Lambda_{(m)}^{(1)} \alpha, \ldots, \Lambda_{(m)}^{(m-1)} \alpha] \) and \( \alpha \) is a constant vector.

**Proof:** From (3.21)

\[ \psi(m)^T = [\psi(m), \Lambda_{(m)}^{(1)} \psi(m), \ldots, \Lambda_{(m)}^{(m-1)} \psi(m)] \]

and since the matrices \( \psi(m)^T \psi(m) \) and \( \Lambda_{(m)}^{(i)} \)'s are symmetric

\[
\begin{bmatrix}
\psi(m)^T \\ \psi(m)^T \psi(m) \\ \vdots \\ \psi(m)^T \Lambda_{(m)}^{(m-1)} \psi(m)
\end{bmatrix}
\]

Noting that \( \psi(m)^T \Lambda_{(m)}^{(i)} \alpha \) is a scalar,

\[
\psi(m)^T \Lambda_{(m)}^{(i)} \alpha = (\psi(m)^T \Lambda_{(m)}^{(i)} \alpha)^T = \alpha \Lambda_{(m)}^{(i)} \psi(m)
\]

for \( i = 1, 2, \ldots, (m-1) \).

Hence
\[
\psi(m)^T \psi(m)^\alpha = \begin{bmatrix}
\alpha^T \psi(m) \\
\alpha \Lambda_1(m) \psi(m) \\
\vdots \\
\alpha \Lambda_{m-1}(m) \psi(m)
\end{bmatrix} = [\alpha, \Lambda_1(m), \ldots, \Lambda_{m-1}(m)]^T \psi(m) = \Lambda \psi(m)
\]

since \(\Lambda\) is symmetric.

**Lemma 3.5**

\[
\psi(m)^T \psi(m)^\alpha \beta^T \psi(m) = \Lambda \alpha \Lambda \psi(m)
\]

where \(\beta\) is a constant vector.

**Proof:** Using lemma 3.4, \(\psi(m)^T \psi(m)^\alpha \beta^T \psi(m) = \Lambda \alpha \psi(m) \beta^T \psi(m)\) and since \(\beta^T \psi(m)\) is a scalar and again using lemma 3.4, \(\psi(m)^T \psi(m)^\beta = \psi(m)^T \psi(m)^\beta = \Lambda \beta \psi(m)\) thus proving the lemma.

From (3.17) and lemmas 3.4 and 3.5, (3.25) can be written as

\[
E\psi(m)(t) = x(0) + \sum_{i=1}^{n} F_i \Lambda e_i P(m)\psi(m)(t) + \sum_{k=1}^{q} \sum_{i=1}^{n} G_{k_i} \Lambda e_i \Lambda z_k P(m)\psi(m)(t)
\]

\[
+ \sum_{k=1}^{q} H_k \Lambda \Lambda z_k P(m)\psi(m)(t)
\]

(3.26)

It is to be noted that the above derivation is carried out for arbitrary \(t \in [0, 1]\) and, therefore, holds for all \(t\) in this interval. It is now a relatively simple computational task to identify \(F_i, G_{k_i}\) and \(H_k\) for known \(E, z_k\) and \(\nu\) from input/output data. m linearly
independent equations can be generated from (3.26) by sampling the Walsh functions at \( t = i/2m \) for \( i = 1, 3, \ldots, (2m-1) \). These equations can be solved for the unknowns \( F_i \), \( G_k \), and \( H_k \). Situations may arise where the number of unknowns is greater than \( m \). For example, in the identification, using 8 Walsh functions, of a second order time-invariant bilinear system with a two-input vector, the unknown components of matrices \( A \), \( N_k \), and \( B \) total 16. Such situations can be handled by using an additional input/output data with a different initial condition or with a different input-vector. Another possible solution to such a problem is to increase the number of Walsh functions being used for the identification.

While the above derivation is carried through for bilinear systems, the method can, of course, be applied to a very large class of nonlinear systems. For example, the polynomial type nonlinear term, \( x^n \), (\( x \), a scalar) can be handled using lemmas 2.4 and 2.5 repeatedly. Let

\[
x(t) = e^T \psi(t)
\]

Then the repeated use of lemmas 3.4 and 3.5 gives

\[
x^n(t) = e^T (Ae)^{n-1} \psi(t).
\]

If the system is completely observable then the unknown initial state of the system, such as \( y(0) \) in (3.25), may be viewed as an
additional unknown parameter and hence can be estimated.

The proposed method, however, cannot be used in cases where \( \dim(y) \neq \dim(x) \) - a short-coming of a large class of parameter estimation methods. A recent result by Friedland [56], however, indicates that this difficulty can be avoided in cases where

\[
y(t) = H(t) x(t)
\]

and \( H(t) \) is a known time varying matrix-valued function. Also fractional powers in nonlinearities cannot be handled by the proposed method because lemmas 3.4 and 3.5 do not apply, and it is doubtful if analogous results in this framework could be obtained to handle such cases.

Computational results illustrating the proposed method are presented in section 3.7.

### 3.5 Solution to Nonlinear Equations

The lemmas developed in the previous section coupled with the results of section 3.3 provide a method for computing the solution of a class of nonlinear differential and integral equations with no finite escape time. For purposes of discussion, consider the time-invariant bilinear system

\[
\dot{x}(t) = Ax(t) + \sum_{k=1}^{q} N_k x(t) u_k(t) + \sum_{k=1}^{q} b_k u_k(t), x(t_0) = x_0 \quad (3.27)
\]
where the parameters $A$, $N_k$, and $b_k$, the initial state $x_0$, and the control $u_k(t)$ are known.

With

$$x(t) \sim S \psi_{(m)}(t)$$

(3.28)

$$u_k(t) \sim \psi^T_{(m)}(t) z_k$$

as in (3.24) and using lemma 3.4 and (3.17), (3.27) can be written as

$$S \psi_{(m)}(t) = x_0 + ASP_{(m)} \psi_{(m)}(t) + \sum_{k=1}^{q} N_k S \Lambda z^T_{k}(t) + \sum_{k=1}^{q} b_k z_k P_{(m)}(t)$$

With $F = [x_0', 0, \ldots, 0]$, and since the equation holds for all $t \in [0, 1]$, it follows that

$$S = ASP_{(m)} - \sum_{k=1}^{m} N_k S \Lambda z^T_{k}(m) = F + \sum_{k=1}^{m} b_k z^T_k P_{(m)}$$

(3.29)

Letting

$$\Lambda z^T_{k}(m) = T_k$$

$$F + \sum_{k=1}^{q} b_k z^T_k P_{(m)} = V$$

and using the symbol $\otimes$ for Kronecker product [57], (3.29) becomes...
\[ s - [A \otimes P^T_{(m)}] s - (\sum_{k=1}^{q} [N_k \otimes T_k^T]) s = v \]  \hspace{1cm} (3.30)

where \( s \) and \( v \) are np-vectors defined as

\[ s = [s_{11}, \ldots, s_{n1}; s_{12}, \ldots, s_{n2}; \ldots; s_{1p}, \ldots, s_{np}] \]

\[ v = [v_{11}, \ldots, v_{n1}; v_{12}, \ldots, v_{n2}; \ldots; v_{1p}, \ldots, v_{np}] \]

With \( I \) denoting np x np identity matrix, (3.30) can be solved for \( s \) as

\[ s = [I - [A \otimes P^T_{(m)}] - (\sum_{k=1}^{q} [N_k \otimes T_k^T])]^{-1} v \]  \hspace{1cm} (3.31)

Once \( s \) is determined, the Walsh series representation for \( x(t) \) is determined and hence, the solution to (3.27).

### 3.6 Walsh-function Expansion for White Gaussian Noise

The identification method developed in section 3.4 is applicable, as previously noted, only when perfect information is available; that is, the inputs and the observations are not corrupted by noise. However, in many physical systems some kind of external disturbance is present, which appears in the state and/or observation equation. Further, when a physical process is modeled by a differential equation, a "noise term"—usually white gaussian noise—is added to the state equation to reflect any possible error made in
the modeling of the dynamics of the process. To handle such problems within the framework of the previous sections, it is important to examine the Walsh-function expansion for white gaussian noise, which is considered below. The treatment here is similar to that in [58] where Haar functions are used.

Consider the Walsh function expansion of a zero mean, unit variance white gaussian process \( \xi_t \) on the interval \([0, 1)\):

\[
\xi_t = \sum_{k=0}^{\infty} c_k \psi_k(t) \quad (3.32)
\]

Associated with \( \xi_t \) is the Wiener process \( W_t \) which is defined as the integral of \( \xi_t \), that is,

\[
W_t = \int_{0}^{t} \xi_s \, ds, \quad W_0 = 0 \quad (3.33)
\]

The Wiener process thus defined is completely characterized by the first two moments [59]

\[
E[W_t] = 0 \quad (3.34)
\]

\[
E[(dW_t)^2] = dt
\]

Using (3.32), (3.33) results in

\[
W_t = \sum_{k=0}^{\infty} a_k(t) \quad (3.35)
\]

where
\alpha_k(t) = \int_0^t c_k \psi_k(\tau) \, d\tau = c_k J_k(t) \quad (3.36)

with \( J_k \) as defined by (3.10). The coefficients \( c_k \) in (3.32) and (3.36) can be obtained as

\[ c_k = \int_0^1 \psi_k(t) \, dW_t \quad (k = 0, 1, 2, \ldots) \quad (3.37) \]

This follows by multiplying both sides of (3.32) by \( \psi_k(t) \) and integrating by using the orthonormal property of Walsh functions (3.6). The random variables \( c_k \), \( (k = 0, 1, 2, \ldots) \), are Gaussian \[22\], and, by using (3.34) and (3.6), it can be seen that

\[ E[c_k] = 0 \]
\[ E[c_k c_\ell] = \begin{cases} 1 & k = \ell \\ 0 & k \neq \ell \end{cases} \tag{3.38} \]

for all \( k \) and \( \ell \). Thus \( c_k \), \( (k = 0, 1, 2, \ldots) \), are independent Gaussian random variables with zero mean and unit variance.

The convergence of the series (3.35) is now relatively simple to establish. Fine \[55\] showed that

\[
\max J_k(t) = 1/2^n
\]
\[
\min J_k(t) = -1/2^n
\tag{3.39}
\]
on \( t \in [0,1) \) for \( 2^{n-1} < k \leq 2^n \); \( n \geq 1 \) integer.
Consider
\[
\begin{align*}
    u_n &= \| \alpha_k(t) \|_\infty = \max_{t \in [0,1]} |\alpha_k(t)| \\
    &= \frac{1}{2^n} \max_k |c_k|, \quad 2^{n-1} < k < 2^n
\end{align*}
\]

Since \( c_k, k = 0, 1, 2, \ldots \) are independent, zero mean and unit variance Gaussian random variables, for arbitrary \( \theta > 1 \), it follows that

\[
\text{Prob} [u_n > \theta] = \text{Prob} \left[ \max_k |c_k| > 2^n \theta \right]
\]

\[
\leq \frac{1}{(2^n \theta)^2}
\]

from Chebychev's inequality [59]. With \( \theta > 1 \), the sequence \( \{\frac{1}{(2^n \theta)^2}\} \) converges to zero monotonically from above as \( n \to \infty \), and hence by the Borel-Cantelli lemma [58]

\[
\text{Prob} [u_n \leq \theta \mid n \to \infty] = 1
\]

thus proving the uniform convergence with probability one.

Now consider the scalar stochastic bilinear system

\[
x(t) = x(0) + \int_0^t a(\tau) x(\tau) \, d\tau + \int_0^t n(\tau) x(\tau) \xi_\tau \, d\tau \quad (3.40)
\]

where \( \xi_\tau \) is the above-mentioned white Gaussian process. While the series expansion of (3.32) would allow treatment of \( \xi_\tau \) in the form...
\[ \xi(t) = \mathbf{c}^T \psi(t) \quad \mathbf{c} = [c_0, c_1, \ldots]^T \]

This approach is useful in computations only if the series is truncated to \(m\) terms, say, \(m = 2^n\) for some finite integer \(n\). A truncated series produces, however, an immediate problem in that establishing convergence for the Walsh function approximated version of (3.40) as \(m \to \infty\) to the known closed form solution \([60]\) of (3.40) appears unfeasible. This difficulty can be attributed to the fact that the convergence of the series expansion of (3.33) to the sample functions of Wiener process has been established with probability one only and not in the mean-square sense.

In the light of the above remarks, computations for the stochastic case would not be meaningful. However, in the next section, a computational example is presented to illustrate the effectiveness of the estimation method of section 3.4 for deterministic cases that are subject to modeling and measurement errors.

### 3.7 Computational Examples

The effectiveness of any proposed method is determined by how well the method fares in solving real world problems. In the absence of handy experimental data, one sometimes resorts to data obtained from the simulation of the model on a digital computer.
which is the case with the examples in this section. The parameter estimation method of section 3.4 is used in the examples that include a general variable structure system, a two-input, second-order bilinear system and a scalar bilinear system with time-varying parameters. To illustrate the effectiveness of the method for deterministic cases that are subject to modeling and measurement errors, an example is included where, the input-output data is obtained from a simulated model wherein a small noise component is added to both the state and observation equations.

**Example 3.1:** Consider the following variable-structure system

\[ y(t) = ay(t) + by(t)^2u(t) + cy(t)u^2(t) + du(t) \]  
\[ (3.41) \]

with \( a = -1, \ b = 0.5, \ c = 0.3, \ d = 1.5, \ u(t) = e^{-0.5t} \) and zero initial condition. For simulation purposes, the output \( y(t) \) is generated using a fourth-order Runge-Kutta method with \( t = j \Delta \) where \( j = 0, 1, 2, \ldots, 8 \) and \( \Delta = 0.125 \). Using the method developed in Section 3.4, (3.41) can be written as

\[ _e^T \psi(t) = a_e^T P \psi(t) + b_e^T \Lambda_e \Lambda_z P \psi(t) + c_z^T \Lambda_e \Lambda_z P \psi(t) + d_z^T P \psi(t) \]  
\[ (3.42) \]

where \( e \) and \( z \) are the Walsh coefficient vectors of \( y(t) \) and \( u(t) \) respectively. To solve for the four parameters \( a, b, c, d \) four
equations are generated from (3.42) using four arbitrarily chosen but distinct time points. For the results tabulated in Table 3.1 time points \( t = \frac{3}{16}, \frac{7}{16}, \frac{11}{16}, \text{ and } \frac{15}{16} \) were used.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Total number of Walsh functions used</th>
<th>True Value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( m = 4 )</td>
<td>( m = 8 )</td>
</tr>
<tr>
<td>( a )</td>
<td>-0.9983</td>
<td>-0.9996</td>
</tr>
<tr>
<td>( b )</td>
<td>0.4963</td>
<td>0.4991</td>
</tr>
<tr>
<td>( c )</td>
<td>0.2987</td>
<td>0.2997</td>
</tr>
<tr>
<td>( d )</td>
<td>1.4951</td>
<td>1.4988</td>
</tr>
</tbody>
</table>

**Example 3.2:** Consider the second-order bilinear system described by

\[
\dot{y}(t) = Ay(t) + \sum_{k=1}^{2} N_k y(t) u_k(t) + Bu(t) \tag{3.43}
\]

where \( u(t) \) and \( y(t) \) are input and output vectors respectively and \( A, N_1, N_2 \) and \( B \) are constant 2x2 matrices. Again using a fourth-order Runge-Kutta method for generating the output data with

\[
A = \begin{bmatrix} 0 & 1 \\ -0.5 & -1 \end{bmatrix}, \quad N_1 = \begin{bmatrix} 0 & 1 \\ 0.5 & 0.2 \end{bmatrix}, \quad N_2 = \begin{bmatrix} 1 & 0 \\ 0.2 & 0.3 \end{bmatrix}, \quad B = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}
\]
\[ u_1(t) = e^{-0.5t}, \quad u_2(t) = e^{-t} \text{ and } t = j \Delta \text{ where } j = 0, 1, 2, \ldots, 8 \text{ and } \Delta = 0.125. \]

Two initial conditions \[ y_T^T(0) = (0, 0) \text{ and } y_T^T(0) = (0.2, 0.3) \] are used.

Using the technique of section 3.4, (3.43) may be manipulated to yield two equations:

\[ C \alpha_1 \simeq y_1 \quad (3.44) \]

and \[ C \alpha_2 \simeq y_2 \]

where \[ \alpha_i = (a_{i1} \quad a_{i2} \quad n_{i1}^{(1)} \quad n_{i2}^{(1)} \quad n_{i1}^{(2)} \quad n_{i2}^{(2)} \quad b_{i1} \quad b_{i2}) \]

\[ y_T^T = (e_{i1}^T \psi(m)(t_1), e_{i1}^T \psi(m)(t_2), \ldots, e_{i1}^T \psi(m)(t_m)) \]

for \( i = 1, 2; a_{ij}, n_{ij}^{(k)} \) and \( b_{ij} \) are the components of the matrices \( A, N_k \) and \( B; \) and \( C \) is a \( m \times 8 \) matrix with an \( i^{th} \) row as

\[ [e_{i1}^T \psi(m)(t_1), e_{i2}^T \psi(m)(t_1), e_{i1}^T \Lambda z_1^T P \psi(m)(t_1), e_{i2}^T \Lambda z_2^T P \psi(m)(t_1), e_{i1}^T \Lambda z_1 P \psi(m)(t_1), e_{i2}^T \Lambda z_2 P \psi(m)(t_1), \]

\[ e_{i1}^T \Lambda z_2 P \psi(m)(t_1), e_{i2}^T \Lambda z_2 P \psi(m)(t_1)] \]

With \( m = 8 \) Walsh functions, and since (3.44) is an approximate equation, to smooth out any errors 12 equations were chosen arbitrarily.
from the 16 generated with two different initial conditions to solve for
the 8 unknowns in each of equations (3.44) to yield the following values:

\[
A = \begin{bmatrix}
-0.005 & 0.979 \\
-0.500 & -1.004 \\
\end{bmatrix}, \quad N_1 = \begin{bmatrix}
0.016 & 1.033 \\
0.500 & 0.206 \\
\end{bmatrix}, \quad N_2 = \begin{bmatrix}
0.991 & -0.019 \\
0.200 & 0.296 \\
\end{bmatrix}, \quad B = \begin{bmatrix}
2.022 & 0.969 \\
1.001 & 0.992 \\
\end{bmatrix}
\]

**Example 3.3:** Consider the bilinear system with time-varying
parameters:

\[
y(t) = a(t)y(t) + n(t)y(t)u(t) + b(t)u(t)
\]

(3.45)

Using the technique of section 3.4, (3.26) corresponding to (3.45)
with

\[
a(t) = \int_f^T \psi(t)
\]

\[
n(t) = \int_g^T \psi(t)
\]

\[
b(t) = \int_h^T \psi(t)
\]

(3.46)

can be written as

\[
e^T \psi(t) = y(0) + \int_f^T \Lambda_e P \psi(t) + \int_g^T \Lambda_e \Lambda z P \psi(t) + \int_h^T \Lambda z P \psi(t)
\]

where \( e \) and \( z \) are the Walsh coefficient vectors of \( y(t) \) and \( u(t) \)
respectively. Using \( m \) Walsh functions, it is required to find \( 3m \)
components of \( f, g \) and \( h \). Four different inputs \( e^{-2t}, 0.2 \), \( 2t, 0.1 \),
\( te^{-t}, 4.1 \) and \( 1/(t+0.5), 2.6 \), (the numbers in the parenthesis being
the initial conditions) are used in (3.46) to generate 4m equations to solve for the 3m unknowns. The plots of a(t), n(t) and b(t) for m=4 and m=8 Walsh functions are illustrated in Figure 3.2.

An examination of the above computational examples reveals the following:

In example 3.1 the parameter estimation error with a 4 Walsh-function approximation is less than 0.8%, while for 8 Walsh functions, the error is less than 0.2%. Employing the norm

\[ ||A|| = \max_{||x||=1} ||Ax|| \]

the maximum estimation error (that of the matrix N1) is 0.034 in example 3.2. With the error defined as

\[ \frac{1}{m} \sum_{i=1}^{m} (p(t_i) - \hat{p}(t_i))^2 \]

where \( p(t_i) \) is the true parameter and \( \hat{p}(t_i) \) is the estimate at time \( t_i \), in example 3.3 the error is less than 0.3% for \( m = 4 \) and less than 0.1% for the 8 Walsh-function approximation.

The above observations illustrate the remarkable accuracy of the Walsh-function method.

Being a computational method, the remarks of section 3.6 seem to destroy the chances of using the Walsh-function method for bilinear systems with stochastic inputs. However, the
Figure 3.2. Computed results for example 3.3. The solid circles denote the values obtained using 8 Walsh functions and the open circles using 4 Walsh functions.
following example indicates that acceptable parameter estimates (error less than 10%) can be obtained even in cases where the bilinear system is subject to small modeling and measurement errors.

**Example 3.4:** This example illustrates the effect of modeling and measurement noise. Consider the scalar bilinear system

\[
\begin{align*}
\dot{x}(t) &= ax(t) + nx(t)u(t) + bu(t) + \xi(t) \\
y(t) &= x(t) + \eta(t)
\end{align*}
\] (3.47)

where \(\xi(t)\) and \(\eta(t)\) are independent Gaussian white noises with zero mean and variances given by

\[
E[\eta(t)\eta(\tau)] = r\delta(t - \tau)
\]

\[
E[\xi(t)\xi(\tau)] = q\delta(t - \tau)
\]

with the parameter values \(a = -2, \ n = 1, \ b = 3, \ u(t) = 2\exp(2(t-0.5))\), \(x(0) = 5.0\) and \(\Delta = 1/64\). Output data is generated using Euler's method for several values of \(r\) and \(q\).

In order to apply the technique of section 3.4, \(x(t)\) is to be estimated. With \(x(t) = \psi^T (m) (t) e\), (3.48) becomes

\[
y(t) = \psi^T (m) (t) e + \eta(t)
\] (3.49)
The least-squares estimate of $e$ for $m$ observations in $(3.49)$ is given by

$$e = \left[ V_m^T r_1^{-1} V_m \right]^{-1} V_m^T r_1^{-1} \gamma_m$$

where $V_m = [\psi(m)(t_1), \ldots, \psi(m)(t_m)]^T$ and

$$\gamma_m = [y(t_1), \ldots, y(t_m)]^T$$

With $t_i = (2i - 1)/2m$ for $i = 1, 2, \ldots, m$, it is easy to see that $V_m = W_m$, the Walsh matrix of order $m$. Noting that $W_m W_m^{-1} = mI_m$

where $I_m$ is the identity matrix of order $m$, and $W_m = W_m^{-1}$, $(3.50)$ becomes

$$e = \left( \frac{1}{m} \right) W_m \gamma_m = f$$

(see Appendix B)

where $f$ is the Walsh coefficient vector of $y(t)$. This implies that, since the Walsh coefficient vector for a function is unique, the least-squares estimate of $x(t)$ of $(3.48)$ is $y(t)$ which is as expected. Thus using the output generated for $x(t)$ and neglecting $\xi(t)$ (which is the reason for using very small values for $q$) in the parameter estimation scheme, the technique of section 3.4, with $m = 16$, is applied to $(3.47)$. In order to average out the effect of noise, all 16 equations generated are used to solve for the three unknown parameters. The results are tabulated in Table 3.2.
Table 3.2. Parameter estimates with various noise variances for example 3.4.

<table>
<thead>
<tr>
<th>Noise Covariance</th>
<th>Parameter Estimates</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>a</td>
</tr>
<tr>
<td>q = 0.01</td>
<td>0.09</td>
</tr>
<tr>
<td>r = 0.16</td>
<td></td>
</tr>
<tr>
<td>q = 0.01</td>
<td>0.25</td>
</tr>
<tr>
<td>r = 0.01</td>
<td>0.09</td>
</tr>
<tr>
<td>True value</td>
<td>-2.00</td>
</tr>
</tbody>
</table>

3.8 Application to an Immunological Model

As an application of the estimation method of section 3.4, a simplified version of the immunological model discussed in section 1.4 is considered here. The parameters which appear in the model fall into two categories: (i) those well established from animal experiments, and (ii) those that have yet to be estimated with any degree of accuracy. A parameter that falls into the latter category is $\alpha$, the birthrate constant of stimulated immunocompetent cells, which appears in the second component of the model equation (1.7). The estimation of such an unknown parameter provides, in some situations, a good basis for the control of the immune response.
Moreover, it is crucial to the development of a better immunological model—one that can play a key role in such areas as immunotherapy for the control of diseases.

In this section, the parameters $\alpha$ and $\tau_2$ in the second component of the immunological model (1.7)

\[
\frac{dx_2(t)}{dt} = \frac{1}{\tau_2} x_2(t) + 2\alpha \mu_2(t) x_1(t)
\]  

(3.51)

are estimated using the method of section 3.4. The data, used for parameter estimation, are taken from [14] wherein, a simulation of the model (1.7), which compares favorably with experimental data, is presented. In order to use the data which are given at time points that are ten hours apart, it is first necessary to normalize time to $[0, 1)$. Let

\[
\tau = (t - t_i) / (t_f - t_i)
\]

where $t_i$ and $t_f$ are the initial and final times of the data used for estimation. Then (3.51) becomes

\[
\dot{x}_2(\tau) = (t_f - t_i) \left\{ -\frac{1}{\tau_2} x_2(\tau) + 2\alpha \mu_2(\tau) x_1(\tau) \right\}, \quad 0 \leq \tau < 1
\]  

(3.52)

From the technique of section (3.4), (3.52) may be written as

\[
e^T \phi(\tau) = x_2(t_i) - \frac{(t_f - t_i)}{\tau_2} e^T P \psi(\tau) + 2\alpha(t_f - t_i) \int_0^T e^T \Lambda e^T P \psi(\tau)
\]  

(3.53)
where \( \underline{z} \), \( \underline{e} \) and \( \underline{f} \) are the Walsh-coefficient vectors of \( x_1(\tau) \), \( x_2(\tau) \) and \( \mu_0(\tau) \) and \( \Lambda_{\underline{z}} = [\underline{z}, \Lambda_{\underline{z}}^{(1)}, \underline{z}, \ldots, \Lambda_{\underline{z}}^{(p)}] \). With \( t_i = 160 \) hours and \( t_f = 240 \) hours in (3.53) and using an 8 Walsh-function approximation, four equations are generated to solve for the two parameters \( \alpha \) and \( \tau_2 \). The following values are obtained:

\[
\alpha = 0.0572
\]

\[
\tau_2 = 20.7
\]

In the simulation of the model, the values used were \( \alpha = 0.0578 \) and \( \tau_2 = 21 \), thus showing the remarkable accuracy of the estimation method in spite of the large spread in the values of the estimated parameters.

The results presented in the previous two sections illustrate the accuracy of the Walsh-function method of parameter estimation. Moreover, since Walsh functions are relatively easy to handle in computations, the estimation technique is quite simple to implement and has, therefore, obvious advantages in practical situations. Further, the initial estimates of the unknown parameters are not required and the system is not subjected to any probing inputs which is very important from a biological point of view. Of course, the method is not without any disadvantages. The number of Walsh functions that have to be employed are \( 2^n \) (\( n \) an integer) and as a consequence the computer storage requirements for higher
dimensional problems might become excessive if more than 8-16 Walsh functions are used. Also, equispaced data, though not necessarily from time $t = 0$, is required. This is the chief limitation on its applicability together with the cases where the measurement noise is significant.
IV. CONCLUSIONS

In this dissertation two functional expansions, the Volterra series and the Walsh-function expansion, are studied in the bilinear system analysis.

The Volterra series for bilinear systems is derived by Reversion technique. The uniform convergence of the series is proved and estimates on the bound of the system output and on the error due to the truncation of the series are obtained. The above development method has given an insight into the structure of the system, leading to the synthesis of the Volterra kernels that characterize the bilinear system. This synthesis result is related to the canonic forms given by Bush for nonlinear systems. Moreover, the synthesized kernels have facilitated the computation of the Volterra transfer functions. The notions of "weakly" bilinear, which are systems that can be represented by a finite Volterra series, and "strongly" bilinear, which cannot be so represented, are introduced. These concepts are illustrated with examples. A method is proposed to identify the parameters of the system employing discrete Volterra kernels. Using this method, reasonable estimates are obtained for the parameters of the neutron kinetics model.
In the second approach, Walsh functions are used as the basis in the finite orthonormal expansion of input/output functions for parameter estimation and input/parameter functions for system output computation. By exploiting the finite group properties of Walsh functions and the relationship between Walsh functions and their integrals, the system equations are reduced to algebraic equations which are conveniently solved for the parameters of the system output. This method is shown to be applicable to time-variant bilinear systems and to systems with polynomial type nonlinearities as well. To handle the presence of noise in the system equation, the Walsh function expansion of White Gaussian noise is considered and it is concluded that such cases cannot be handled within the framework of the above technique. However, the effectiveness of the proposed method for parameter estimation of deterministic systems subject to modeling and measurement noise is examined and it is found that reasonable parameter estimates can be obtained. The remarkable accuracy of the method for deterministic cases is demonstrated by other computational examples which include a time-variant bilinear system. Since Walsh functions are relatively easy to handle in computations, the proposed method is quite simple to implement and has, therefore, obvious advantages in practical situations.

For possible future study, some of the topics which remain to
be considered are the following:

1) the evaluation of the parameter estimation method with actual measurements of the Volterra kernels, and the generalization of the technique to cases where the matrix $A$ is not in the canonical form;

2) the extension of the Walsh-function method for systems with stochastic inputs and observation noise, and for systems with fractional powers in their nonlinearities; and

3) the evaluation of the performance of the Walsh-function method using data obtained from animal experiments so as to develop an accurate immunological model.
BIBLIOGRAPHY


APPENDIX A

Some Lie-Algebraic Concepts

To facilitate the reading of section 2.6 some relevant concepts of Lie algebra are recalled here. Detailed accounts can be found in [49].

1. A Lie algebra $L$ is a finite dimensional vector space over a field $F$ in which there is defined a multiplication operation that associates with each $x$ and $y$ in $L$ an element in $L$ denoted by $[x, y]$. For $x, y, z \in L$ and $a, b \in F$, this operation has the following properties:

   i) bilinearity, $[x, ay + bz] = a[x, y] + b[x, z]$

   ii) antisymmetry, $[x, y] = -[y, x]$

   iii) the Jacobi identity, $[[x, y], z] + [y, [z, x]] + [z, [x, y]] = 0$

   An important example of a Lie algebra is the set of all $n \times n$ matrices with the Lie product defined by $[A, B] = AB - BA$ where $AB$ is the ordinary matrix product of $A$ and $B$. This is called a matrix Lie algebra.

2. A subalgebra $S$ of the Lie algebra $L$ is a subspace which is closed under the bracket operation of $L$; that is, if $x$ and $y$ are in $S$, then $[x, y]$ is in $S$. 
3. An ideal $S$ of a Lie algebra $L$ is a subalgebra which satisfies the property that if $x$ is in $S$ and $y$ is in $L$, then $[x, y]$ is in $S$.

4. A Lie algebra $L$ is abelian if $[x, y] = 0$ for all $x, y$ in $L$.

5. The adjoint operator on a Lie algebra $L$ is defined by

\[
\text{ad}_x^0 y = y, \quad \text{ad}_x^1 y = \text{ad}_x y = [x, y], \quad \text{and} \quad \text{ad}_x^{k+1} y = [x, \text{ad}_x^k y]
\]

for $k = 1, 2, \ldots$.

6. A Lie algebra $L$ is nilpotent if and only if, for $x$ and $y$ in $L$, $\text{ad}_x^n y = 0$ for some $n$.

7. A Lie algebra $L$ is solvable if $L^{(n)} = \{0\}$. Here $L^{(n)}$ is defined inductively by

\[
L^{(1)} = [L, L]
\]

\[
L^{(k+1)} = [L^{(k)}, L^{(k)}]
\]

where $[L, L] = \{ [x, y], \text{ for any } x, y \in L \}$

Note that a Lie algebra $L$ is abelian if $L^{(1)} = \{0\}$ (see (4)).
APPENDIX B

Computation of Walsh Coefficients

In this appendix the procedure for expressing $u(t)$ and $y(t)$ in terms of Walsh functions is illustrated. Let

$$u(t) = h^T \psi(t) = h_0 \psi_0(t) + h_1 \psi_1(t) + h_2 \psi_2(t) + \ldots.$$  

$$y(t) = e^T \psi(t) = e_0 \psi_0(t) + e_1 \psi_1(t) + e_2 \psi_2(t) + \ldots.$$ 

The object is to find $h$ and $e$.

The input $u(t)$ and the output $y(t)$ are sampled with a sampling period $\triangle = \frac{1}{m}$ over the period $(0, 1)$ for $m = 2^n$, $n$ an integer.

Suppose the sampled values of $u(t)$ and $y(t)$ are available at $t = j\triangle$ for $j = 0, 1, 2, \ldots m$, and are averaged over each subinterval, $(j-1)\triangle \leq t \leq j\triangle$ to obtain

$$\overline{u}(i) \text{ and } \overline{y}(i) \text{ for } i = 1, 2, \ldots, m.$$ 

That is,

$$\overline{y}(i) = 0.5 \left( y((i-1)\triangle) + y(i\triangle) \right) \text{ and }$$

$$\overline{u}(i) = 0.5 \left( u((i-1)\triangle) + u(i\triangle) \right).$$

The Walsh coefficient vectors $h$ and $e$ are obtained from the following equations:
\[ e = \frac{1}{m} W_m \bar{y} \]

\[ h = \frac{1}{m} W_m \bar{u} \]

where \( W_m \) is the matrix of Walsh functions and \( \bar{y} \) and \( \bar{u} \) are the column vectors of averaged sampled values of \( y(t) \) and \( u(t) \) respectively. Because of the nature of Walsh functions, the Walsh matrix is simply a matrix of plus and minus ones. For example, for \( m = 4 \) the Walsh matrix is given by:

\[
W = \begin{bmatrix}
1 & 1 & 1 & 1 \\
1 & 1 & -1 & -1 \\
1 & -1 & 1 & -1 \\
1 & -1 & -1 & 1
\end{bmatrix}
\]