On the Application of Multi-Input Volterra Theory to Nonlinear Multi-Degree-of-Freedom Aerodynamic Systems

By

Maciej Balajewicz
B. Eng., Carleton University

A thesis submitted to The Faculty of Graduate Studies and Research in partial fulfilment of the degree requirements of
Master of Applied Science

Ottawa-Carleton Institute for Mechanical and Aerospace Engineering

Department of Mechanical and Aerospace Engineering
Carleton University
Ottawa, Ontario, Canada
Aug 2007

Copyright ©
2007 – Maciej Balajewicz
NOTICE:
The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.
Abstract

The following thesis presents a reduced-order-modeling approach for nonlinear, multi-degree-of-freedom aerodynamic systems using multi-input Volterra theory. The method is applied to a two-dimensional, two-degree-of-freedom transonic airfoil undergoing simultaneous forced pitch and heave harmonic oscillations. The so called Volterra cross-kernels are identified and are shown to successfully model the lift and moment nonlinearities associated with the simultaneous pitch and heave motions. The improvements in accuracy over previous approaches which effectively ignored the cross-kernel by using superposition are demonstrated. The proposed method is shown to be easily extendable to higher degree-of-freedom aerodynamic systems including three-dimensional wings, where modal coordinates would serve as the degrees of freedom of the system.
"You can know the name of a bird in all the languages of the world, but when you're finished, you'll know absolutely nothing whatever about the bird... So let's look at the bird and see what it's doing — that's what counts. I learned very early the difference between knowing the name of something and knowing something"

Richard Feynman


"An expert is a man who has made all the mistakes which can be made in a very narrow field"

Niels Bohr

Quoted in Alan Mackay, The Harvest of a Quiet Eye (1977)
Acknowledgments

I would like to thank my supervisors Dr. F. Nitzsche and Dr. D. Feszty for their invaluable support both financially and intellectually. I would also like to thank my parents and brother for their unconditional encouragement and support. And last, but certainly no least, my sounding board and partner-in-crime, Demet Ulker.
Table of Contents

Abstract ..................................................................................................................................... iii
Table of Contents ..................................................................................................................... vi
List of Figures ........................................................................................................................ viii
Nomenclature ........................................................................................................................... x

1. Introduction .......................................................................................................................... 1
   1.1 Unsteady Transonic Airfoil Aerodynamics ............................................................ 1
   1.2 Volterra Model of the Unsteady Transonic Airfoil ................................................. 1
   1.3 Organization of the Thesis ........................................................................................ 3

2. An Introduction to the Volterra Theory of Nonlinear Systems ..................................... 4
   2.1 Single-Input Volterra Theory ................................................................................... 4
      2.1.1 Input-Output Characteristics of SDOF Systems ...................................... 4
      2.1.2 Volterra Method for SDOF Nonlinear Systems ....................................... 6
      2.1.3 Numerical Example of a Nonlinear SDOF System ................................. 11
   2.2 Multi-Input Volterra Theory ................................................................................... 15
      2.2.1 Input-Output Characteristics of MDOF Systems .................................... 15
      2.2.2 Volterra Method for MDOF nonlinear systems ..................................... 16
      2.2.3 Numerical Example of MDOF Nonlinear System ................................. 20
   2.3 Multi-Input Volterra Model of the Transonic Two-Dimensional Airfoil ............. 23
   2.4 Description of Test Case ......................................................................................... 24
   2.5 CFD Using the Parallel Multi Block Solver ......................................................... 27

3. Results and Discussion .................................................................................................... 29
   3.1 AGARD Test Case, CFD Validation ................................................................... 29
   3.2 Volterra Kernel Identification ............................................................................... 30
   3.3 Unsteady Results, \( k_h/k_a = 1 \) ......................................................................... 32
   3.4 Unsteady Results, \( k_h/k_a = 0.8 \) ....................................................................... 42
List of Figures

Figure 1: Symmetric and Anti-symmetric Nonlinearities .......................................................... 5
Figure 2: Volterra Kernels of SDOF System ............................................................................. 12
Figure 3: Finite Difference Output of SDOF System ................................................................. 12
Figure 4: Single-Input Volterra Model Errors ........................................................................ 13
Figure 5: Frequency Content of SDOF System Volterra Models Due to Unit Sinusoidal
   at \( \omega = 1 \text{rad/s} \) ........................................................................................................ 14
Figure 6: Volterra Cross-Kernel of MDOF System ................................................................. 21
Figure 7: Finite Difference Output of MDOF System ............................................................... 21
Figure 8: Multi-Input Volterra Model Errors .......................................................................... 22
Figure 9: Frequency Content of MDOF System Due to Unit Sinusoidal Inputs at
   \( \omega_0 = 1 \text{rad/s} \) and \( \omega_1 = 0.8 \text{rad/s} \) ...................................................................... 22
Figure 10: Airfoil Input Coordinate System ............................................................................ 23
Figure 11: Airfoil Output Coordinates ..................................................................................... 24
Figure 12: CFD Validation of AGARD Test Case .................................................................... 29
Figure 13: Volterra Kernels of the Transonic Airfoil ............................................................... 31
Figure 14: Individually Applied Pitch and Heave Motions, \( k_p/k_a = 1 \) ............................... 32
Figure 15: Dynamic Pitch and Heave Inputs .......................................................................... 33
Figure 16: Surface Pressure Coefficients of Independent Pitch and Heave Motion CFD
   Outputs ............................................................................................................................... 34
Figure 17: Surface Pressure Coefficients of Simultaneous Pitch and Heave Motion CFD
   Outputs ............................................................................................................................... 35
Figure 18: Superimposed and Simultaneous Pitch and Heave Motion CFD Outputs,
   \( k_p/k_a = 1 \) ...................................................................................................................... 36
Figure 19: Frequency Content of Superimposed and Simultaneous Pitch and Heave CFD
   Outputs, \( k_p/k_a = 1 \) ........................................................................................................... 37
Figure 20: Volterra Model Using Direct-Kernels Only Against Superimposed and
   Simultaneous Pitch and Heave CFD Outputs, \( k_p/k_a = 1 \) ................................................. 38
Figure 21: Volterra Model Using Direct-Kernels Only, and Direct and Cross-Kernels
   Against Simultaneous Pitch and Heave CFD Output, \( k_p/k_a = 1 \) ................................. 39
Figure 22: Absolute Errors of Volterra Models Using Direct-Kernels Only, and both Direct and Cross-Kernels, $k_h/k_a=1$ ................................................................. 40

Figure 23: Frequency Content of Simultaneous CFD Output and Volterra Models Using Direct-Kernels Only, and Both Direct and Cross-Kernels, $k_h/k_a=1$ ....... 41

Figure 24: Individually Applied Pitch and Heave Motions, $k_h/k_a=0.8$ .............................. 42

Figure 25: Superimposed and Simultaneous Pitch and Heave CFD Outputs, $k_h/k_a=0.8$. 43

Figure 26: Volterra Model Using Direct-Kernels Only Against Superimposed and Simultaneous Pitch and Heave Motion CFD Outputs, $k_h/k_a=0.8$ .................. 44

Figure 27: Volterra Model Using Direct-Kernels Only, and Both Direct and Cross-Kernels Against Simultaneous Pitch and Heave CFD Outputs, $k_h/k_a=0.8$ .... 44

Figure 28: Absolute Errors of Volterra Models Using Direct-Kernels, and Both Direct and Cross-Kernels, $k_h/k_a=0.8$ ................................................................. 45

Figure 29: Frequency Content of Combined CFD Output and Volterra Kernel Outputs, $k_h/k_a=0.8$ ........................................................................... 46

Figure A1: CMB Input File for Volterra Cross-kernel Identification .................. 52

Figure AII: CMB Input File, Two Combined Signals ............................................. 53
Nomenclature

\(\alpha\) Airfoil pitch [°]
\(\bar{\alpha}\) Static airfoil pitch [°]
\(\alpha_0\) Dynamic airfoil pitch [°]
\(\beta\) Roll [rad]
\(\dot{\beta}\) Time derivative of roll [rad/s]
\(\Psi\) Yaw [rad]
\(\theta\) Heave motion phase shift [°]
\(\xi_p\) Modal coordinate corresponding to the \(p^{th}\) mode
\(c\) Airfoil chord
\(C_l\) Lift coefficient
\(C_m\) Moment coefficient
\(C_p\) Pressure Coefficient
\(\delta(t)\) Discrete Dirac delta function
\(\delta_p(t)\) Discrete Dirac delta signal applied to \(p^{th}\) degree-of-freedom
\(f\) Forcing function of example problems
\(f_p\) Forcing function corresponding to \(p^{th}\) degree-of-freedom of example problems
\(h\) Airfoil heave [% of chord]
\(h_0\) Dynamic airfoil heave [% of chord]
\(\dot{h}\) Time derivative of airfoil heave [% of chord/s]
\(h_{p0,1\ldots p}(\cdot)\) \(p^{th}\)-order Volterra kernel
\(H_{p0,1\ldots p}[\cdot]\) \(p^{th}\)-order Volterra operator
\(k_{0,1\ldots p}\) Stiffness of example problems
\(k_a\) Pitch oscillation reduced frequency \([\omega c/2U_\infty]\)
\(k_h\) Heave oscillation reduced frequency \([\omega c/2U_\infty]\)
\(M\) Mach number
\(N\) Summation limit
\(N_p[\cdot]\) \(p^{th}\)-order Volterra system
\(\Delta t\) Time step [s]
\(T_{p0,1\ldots p}[\cdot]\) \(p^{th}\)-order operator
\( T_{p,1^{\ldots p}} \) \( p \text{-linear operator} \\
\( U_\infty \) Airfoil forward velocity \\
\( \omega \) Frequency of oscillation [rad/s] \\
\( \omega_p \) Frequency of oscillation corresponding to \( p \text{th degree-of-freedom} \) [rad/s] \\
\( x(t) \) Arbitrary physically realizable function \\
\( x_p(t) \) Arbitrary physically realizable function corresponding to \( p \text{th degree-of-freedom} \) \\
\( y(t) \) Output coordinate of example problems \\
\( y_p(t) \) Output coordinate corresponding to \( p \text{th degree-of-freedom} \) of example problems

\textbf{Acronyms}

CFD Computational Fluid Dynamics \\
FD Finite Difference \\
MDOF Multi-Degree-of-Freedom \\
PMB Parallel Multi Block \\
SDOF Single-Degree-of-Freedom
1. Introduction

1.1 Unsteady Transonic Airfoil Aerodynamics

The determination of lift and moment characteristics of an airfoil experiencing forced pitch and heave oscillations at transonic conditions is a complex task. The main culprit of this complexity is the appearance of nonlinear effects resulting from the presence of a shock wave at the airfoils surface. These nonlinearities, especially in the field of aeroelasticity, have been shown to have strong and often dangerous consequences. Since most modern civilian and military aircraft spend a significant duration of their life span at transonic conditions, the interest in this field of research has been significant. For a detailed discussion of transonic and nonlinear aeroelasticity refer to Dowell et al. (2003).

Due to the inherent complexities in transonic aerodynamics, Computation Fluid Dynamics (CFD) has become the most reliable tool for its analysis. However, the large computational resources required for such high fidelity analysis renders the approach undesirable; especially during the initial and conceptual design stages. Hence, there has been a great deal of interest in reduced-order-models (ROMs) of the transonic nonlinear system. A ROM is a simplified mathematical model that captures most of the physics of the more complex system under investigation. A popular linear ROM approach uses the concept of the impulse response together with the convolution integral. Such approaches have proven extremely accurate and reliable for both the subsonic and supersonic flight regimes. Transonic aerodynamics however, are inherently nonlinear and require more elaborate methods of modeling. A recent survey of the various methods available is presented by Clark (2004) and Lucia et al. (2004)

1.2 Volterra Model of the Unsteady Transonic Airfoil

Volterra theory has been successfully applied to nonlinear engineering problems since the early 1940's, with the first application to the unsteady transonic airfoil problem being carried out by Silva (1993). Since this first publication, the method has been applied by other researches including Raveh (2000), Marzocca (2001) and Munteanu (2005). Its solid mathematical foundations and similarity to familiar engineering concepts such as convolution and power series expansion have only served to increase
the methods popularity in the engineering community. For a detailed survey of the Volterra method as applied to nonlinear aerodynamics refer to Silva (2005).

However, the application of the Volterra theory to multi-degree-of-freedom (MDOF) systems such as the two-dimensional, two-degree-of-freedom transonic airfoil has been incomplete. This is because the classical Volterra theory is applicable to single-degree-of-freedom (SDOF) systems only. In its applications to MDOF aerodynamic systems, the analysis has always been limited to the identification of aerodynamic nonlinearities resulting from individual perturbations of structural modes. Determination of total lift and moment for simultaneous motions required the superposition of the individual nonlinear responses. However, as suggested by Silva (2005) and Lucia et al. (2004), the nonlinear nature of the system renders the principle of superposition invalid. Hence, a more complete model of the system would include the effects of modal interactions resulting from the nonlinear aerodynamics. An extension of the classical Volterra theory called multi-input Volterra theory is well suited for this purpose and has been successfully utilized by Grzywacz and Koch (1987), Agarossi et al (1998) and Dimoka et al. (2007) in other engineering disciplines.

The multi-input Volterra theory differs from the classical single-input Volterra theory through its inclusion of Volterra cross-kernels. These cross-kernels capture the coupling dynamics between the degrees of freedom of the nonlinear system.

The work summarized in this thesis illustrates the applicability of multi-input Volterra theory to nonlinear aerodynamic systems by modeling the transonic, unsteady, two-dimensional, two-degree-of-freedom airfoil.
1.3 Organization of the Thesis

The following is a short summary of the contents of this thesis.

Section 2. An Introduction to the Volterra Theory of Nonlinear Systems – This section deals with the theoretical derivations of the Volterra theory. Subsection 2.1 introduces the concepts of single-input Volterra theory. Subsection 2.2 extends the principles to two-degree-of-freedom systems using the multi-input Volterra theory. Both sections conclude with illustrative numerical examples.

Section 3. Multi-input Volterra Model of the Transonic Airfoil – The concepts of the previous section are applied to the transonic airfoil system using the pitch and heave degrees of freedom. A description of the specific test case and CFD software used in its analysis is included.

Section 4. Results and Discussion – This section presents the Volterra model and CFD outputs of various simultaneous forced pitch and heave motions. Significant gains in modeling accuracy through the application of the multi-input Volterra theory are demonstrated.

Section 5. Conclusions – Important results and their implications are discussed and summarized.

Section 6. Future Work – Potential future extensions to the work presented in this thesis are discussed.
2. An Introduction to the Volterra Theory of Nonlinear Systems

This section presents a detailed derivation of the Volterra theory. The section is divided into two subsections; the first deals with single-input Volterra theory while the second section deals with the MDOF theory. Each subsection summarizes the general input-output characteristics of SDOF and MDOF nonlinear systems both in the time and frequency domains. Each subsection concludes with an illustrative numerical example.

2.1 Single-Input Volterra Theory

2.1.1 Input-Output Characteristics of SDOF Systems

In the time domain, arbitrary signals can be constructed using a series of impulse functions. In the frequency domain, arbitrary signals can be constructed using a series of sinusoidals. It is possible to classify a system as linear or nonlinear based upon its output to a series of impulses or sinusoidals. In the time domain a linear system is one which operates on each individual impulse from the input signal independently from all the others; the total output being a superposition of the individual responses. In the frequency domain, a linear system is one whose output to a sinusoidal input of frequency \( \omega \) is a sinusoidal of the same frequency \( \omega \). In the time domain, a nonlinear system is one whose operation on each individual impulse from the input signal is a function of past impulses. In the frequency domain, a nonlinear system, is one whose output to a sinusoidal input of frequency \( \omega \) contains frequencies other than the input frequency \( \omega \).

An important category, which encompasses a significant portion of nonlinear systems of engineering interest, is the category of weakly nonlinear system. In the time domain, a weakly nonlinear system is one whose operation on each individual impulse from the input signal is a weak function of past impulses. In the frequency domain, a weakly nonlinear system is one whose output due to a sinusoidal input of frequency \( \omega \) contains only integer harmonics (\( 0\omega, 2\omega, 3\omega, 4\omega, 5\omega, \ldots \)). In general, the relative magnitude and, hence, significance of these harmonics tends to drop off quickly with increasing frequency \( \omega \). Hence, for most engineering purposes only the first two or three harmonics need to be considered. Although no general methods of analysis exist for
nonlinear systems, the techniques are numerous when dealing with weakly nonlinear systems; the Volterra method being just one of them. For a detailed survey of the various methods available refer to Neyfeh and Mook (1979).

To illustrate the characteristics of weakly nonlinear systems in more detail, consider the following nonlinear system:

\[
y'' + y' + k_1y + k_{11}y^2 + k_{111}y^3 = f(t)
\]

(2.1)

where:

\[
k_i >> k_{11}, k_{111}
\]

(2.2)

Equation (2.1) contains two different types of nonlinearities, which can be roughly categorized as symmetric and anti-symmetric. If we consider Equation (2.1) as a mass-damper-spring system we notice that the nonlinear restoring force term, \(k_{111}y^3\), increases the spring’s stiffness when the spring element is elongated \(y>0\) and also when it is compressed \(y<0\). Hence, such a nonlinearity is called symmetric. Alternatively, the nonlinear restoring force term, \(k_{11}y^2\), increases the spring’s stiffness when elongated \(y>0\) while decreasing the spring’s stiffness when compressed \(y<0\). Hence, such a nonlinearity is called anti-symmetric. Figure 1 shows the values of the linear and nonlinear restoring force terms for \(-2<y<2\) when \(k_1=1\) and \(k_{11}=k_{111}=0.5\).

**Figure 1: Symmetric and Anti-symmetric Nonlinearities**

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
In the frequency domain, a symmetric nonlinearity adds odd harmonics \((3\omega, 5\omega, 7\omega, \ldots)\) to the output signal. An anti-symmetric nonlinearity on the other hand, adds both even and odd harmonics \((0\omega, 2\omega, 3\omega, 4\omega, 5\omega, \ldots)\) to the output signal.

Now, if conditions of Equation (2.2) are not met, no known general output characteristics exist. Also, more complicated and less understood phenomenon such as chaos, sub-harmonic \((1/2\omega, 1/3\omega, 2/3\omega \ldots)\) and non-integer harmonic \((3/2\omega, 4/3\omega, 5/2\omega, \ldots)\) oscillations can result.

### 2.1.2 Volterra Method for SDOF Nonlinear Systems

The backbone of the Volterra theory of nonlinear systems is the Volterra series. The Volterra series is an infinite series of Volterra operators. The first operator is called first-order operator, the second is called second-order operator, and so on. The Volterra series is named in honor of the Italian mathematician Vito Volterra who contributed much to its development in the late 1800’s. However, its first application to the modeling of nonlinear systems was by Wiener (1942).

A nonlinear system containing an anti-symmetric nonlinearity can be modeled using an infinite sum of odd and even ordered Volterra operators. A nonlinear system containing symmetric nonlinearities can be modeled using an infinite sum of odd ordered Volterra operators only.

The following section contains a thorough derivation of the Volterra series and the Volterra theory of nonlinear SDOF systems. Starting from the concept of the impulse function, a detailed description of mathematical operators, which are fundamental to the understanding of the Volterra series, is presented.

The building block of all physically realizable discrete signals is the impulse function:

\[
\delta(t) = \begin{cases} 
1 & \text{for } |t| < 1/2\Delta t \\
0 & \text{elsewhere} 
\end{cases} \tag{2.3}
\]

Any function can be represented by a series of impulse functions from \(t=0\) to \(t=N\Delta t\):

\[
x(t) = \sum_{n=1}^{N} x(n\Delta t)\delta(t-n\Delta t) \tag{2.4}
\]
When a general dynamic system receives such an input signal $x(t)$, it will process or operate upon these individual impulses yielding the output signal $y(t)$. Hence, a system can be interpreted as a collection of mathematical operators which map the input signal on to the output space. We define such an operator as:

$$y(t) = T[x(t)]$$  \hspace{1cm} (2.5)

Operators are classified based on how they operate on the individual impulses which make up the input signal.

The most fundamental operator, called the first-order operator, designated as $T_1$ is one whose response to a combination of impulses is a linear operation on the individual impulses:

$$y(t) = T_1[x(t)]$$

$$= T_1[\sum_{n=1}^{N} x(n\Delta t)\delta(t-n\Delta t)]$$

$$= \sum_{n=1}^{N} x(n\Delta t)T_1[\delta(t-n\Delta t)]$$  \hspace{1cm} (2.6)

where $T_1{}$ is called a linear operator because the operator obeys both the scaling and superposition principles.

A second-order operator, designated as $T_2$, is one whose output to a set of impulses is a bilinear operation on the individual impulses:

$$y(t) = T_2[x(t)]$$

$$= T_2[\sum_{n=1}^{N} x(n\Delta t)\delta(t-n\Delta t)]$$

$$= \sum_{n_1=1}^{N} \sum_{n_2=1}^{N} T_2{\{x(n_1\Delta t)\delta(t-n_1\Delta t), x(n_2\Delta t)\delta(t-n_2\Delta t)\}}$$

$$= \sum_{n_1=1}^{N} \sum_{n_2=1}^{N} x(n_1\Delta t)x(n_2\Delta t)T_2{\{\delta(t-n_1\Delta t), \delta(t-n_2\Delta t)\}}$$  \hspace{1cm} (2.7)

The operator $T_2{}$ is called bilinear because $T_2{\{n_1,n_2\}}$ is linear in $n_1$ for any given $n_2$ and linear in $n_2$ for any given $n_1$; also $T_2{\{n_1,n_1\}} = T_2[n_1]$
Hence, a pth-order operator which we designate \( T_p \) is one whose output due to a set of impulses is a pth-linear operation on the individual impulses:

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

\[
y(t) = T_0[\sum_{n=1}^{N} x(n\Delta t)\delta(n\Delta t)]
\]

\[
= \sum_{n_1=1}^{N} \sum_{n_2=1}^{N} \cdots \sum_{n_p=1}^{N} \left\{ x(n_1\Delta t)\delta(t-n_1\Delta t), x(n_2\Delta t)\delta(t-n_2\Delta t), \ldots, x(n_p\Delta t)\delta(t-n_p\Delta t) \right\}
\]

\[
= \sum_{n_1=1}^{N} \sum_{n_2=1}^{N} \cdots \sum_{n_p=1}^{N} x(n_1\Delta t)x(n_2\Delta t)\cdots x(n_p\Delta t)T_2\{\delta(t-n_1\Delta t), \delta(t-n_2\Delta t), \ldots, \delta(t-n_p\Delta t)\}
\]

\[
(2.8)
\]

The operator \( T_p \) is called pth-linear because \( T_p\{n_1,n_2,\ldots,n_p\} \) is linear in each \( n_i \) if all other \( n_j (j \neq i) \) are held fixed; also \( T_p\{n_1,n_2,\ldots,n_p\} = T_p[n_1,n_2,\ldots,n_p] \).

If all such operators \( T_1[x(t)], T_2[x(t)], \ldots, T_p[x(t)] \) are time-invariant, they are called Volterra operators and are designated by \( H_1[x(t)], H_2[x(t)], \ldots, H_p[x(t)] \). By assembling the operators in a series, the pth-order Volterra series, \( N_p \), is created:

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

\[
y(t) = H_1[x(t)] + H_2[x(t)] + \cdots + H_p[x(t)]
\]

\[
y(t) = \sum_{n=1}^{N} x(n\Delta t)H_1\{\delta(t-n\Delta t)\}
\]

\[
+ \sum_{n=1}^{N} \sum_{n_2=1}^{N} x(n_1\Delta t)x(n_2\Delta t)H_2\{\delta(t-n_1\Delta t), \delta(t-n_2\Delta t)\}
\]

\[
\vdots
\]

\[
+ \sum_{n_1=1}^{N} \sum_{n_2=1}^{N} \cdots \sum_{n_p=1}^{N} x(n_1\Delta t)x(n_2\Delta t)\cdots x(n_p\Delta t)H_p\{\delta(t-n_1\Delta t), \delta(t-n_2\Delta t), \ldots, \delta(t-n_p\Delta t)\}
\]

\[
(2.9)
\]

A pth-order Volterra series \( N_p \) used to model a nonlinear system is called a pth-order Volterra system. As stated previously, odd and even ordered harmonics of a nonlinear system can be modeled using odd and even ordered Volterra operators respectively. To demonstrate this property, let's investigate the output of a nonlinear system modeled by Equation (2.9) to a sinusoidal input of frequency \( \omega_0 \). Observing Equation (2.9) closely, we notice that input \( x(t) \) appears \( p \) times in the pth-order operator. Hence, if our input is \( \cos(\omega t) \) the frequency content from the pth-order operator will be equivalent to \( \cos(\omega t)^p \).
Hence, the first-order operator $p=1$, will have a frequency content of $\omega$, the second-order operator $p=2$ will have a frequency content of $0\omega$ and $2\omega$, and the third-order operator $p=3$, will have a frequency content of $1\omega$ and $3\omega$ and so on.

Because the relative magnitude and hence significance of the harmonics tends to drop off quickly with increasing frequency $\omega$, we can approximate the output of a nonlinear system using a finite, low-ordered Volterra system. Since the identification procedure for the trilinear operator is computationally intensive, the application of the Volterra theory to nonlinear aerodynamic systems is commonly limited to second-order Volterra systems $N_2$:

$$y(t) = N_2[x(t)]$$

$$y(t) = \mathbf{H}_2[x(t)] + \mathbf{H}_2[x(t)]$$

$$y(t) = \sum_{n=1}^{N} x(n\Delta t)\mathbf{H}_1[\{\delta(t-n\Delta t)\}]$$

$$+ \sum_{n_1=1}^{N} \sum_{n_2=1}^{N} x(n_1\Delta t)x(n_2\Delta t)\mathbf{H}_2[\{\delta(t-n_1\Delta t), \delta(t-n_2\Delta t)\}]$$

(2.10)

In other words, the application is limited to problems where anti-symmetric nonlinearities dominate since odd ordered Volterra operators ($\mathbf{H}_3, \mathbf{H}_5, \mathbf{H}_7, \ldots$) would be required to model symmetric nonlinearities. Observing Equation (2.10) the key to determine the output of a second-order Volterra system $N_2$ is the determination of its linear $\mathbf{H}_1[\{\delta(t-n\Delta t)\}]$ and bilinear operators $\mathbf{H}_2[\{\delta(t-n_1\Delta t), \delta(t-n_2\Delta t)\}]$. For simplicity, let’s consider a Volterra system composed of the second-order operator only:

$$y(t) = \mathbf{H}_2[x(t)]$$

$$y(t) = \sum_{n_1=1}^{N} \sum_{n_2=1}^{N} x(n_1\Delta t)x(n_2\Delta t)\mathbf{H}_2[\{\delta(t-n_1\Delta t), \delta(t-n_2\Delta t)\}]$$

(2.11)

Since the bilinear operator is a function of two impulses $\delta(t-n_1\Delta t)$ and $\delta(t-n_2\Delta t)$, let’s investigate its output to two such impulses:

$$x(t) = \delta(t-\tau_1\Delta t) + \delta(t-\tau_2\Delta t)$$

(2.12)
Hence:

\[
y(t) = H_1[\delta(t-n_1\Delta t) + \delta(t-n_2\Delta t)] \\
= \sum_{n_1=1}^{N} \sum_{n_2=1}^{N} x(n_1\Delta t)x(n_2\Delta t)H_2[\delta(t-n_1\Delta t) + \delta(t-n_2\Delta t)] \\
= H_2[\delta(t-n_1\Delta t) + \delta(t-n_2\Delta t)] + 2H_2[\delta(t-n_1\Delta t),\delta(t-n_2\Delta t)] \\
+ H_2[\delta(t-n_2\Delta t),\delta(t-n_2\Delta t)] \\
\]

(2.13)

Isolating for the bilinear operator alone and noting that \( T_2[n_1,n_1] = T_2[n_1] \) we get:

\[
2H_2[\delta(t-n_1\Delta t),\delta(t-n_2\Delta t)] = H_2[\delta(t-n_1\Delta t) + \delta(t-n_2\Delta t)] \\
- H_2[\delta(t-n_1\Delta t)] - H_2[\delta(t-n_2\Delta t)] \\
\]

(2.14)

This is the relation for the bilinear operator in terms of the second-order Volterra operator. In other words, we can identify the bilinear operator by applying two impulse signals, each with time delays of \( n_1\Delta t \) and \( n_2\Delta t \), one after the other to the system and from this response subtract the responses of the system due to each impulse signal applied individually. Hence the first term of the bilinear operator corresponds to two impulses separated by \( 0\Delta t \), the second term corresponds to two impulses separated by \( 1\Delta t \), the third term corresponding to two impulses separated by \( 2\Delta t \) and so on.

However, as we know, a general second-order Volterra system \( N_2 \) also contains the first-order operator. It is easy to see that this identification procedure is unaffected by the presence of the first-order operator since:

\[
H_1[\delta(t-n_1\Delta t) + \delta(t-n_2\Delta t)] - H_1[\delta(t-n_1\Delta t)] - H_1[\delta(t-n_2\Delta t)] = 0 \\
\]

(2.15)

Hence, we can say that:

\[
2H_2[\delta(t-n_1\Delta t),\delta(t-n_2\Delta t)] = N_2[\delta(t-n_1\Delta t) + \delta(t-n_2\Delta t)] \\
- N_2[\delta(t-n_1\Delta t)] - N_2[\delta(t-n_2\Delta t)] \\
\]

(2.16)

The linear operator can then be identified by applying a single impulse to the system and subtracting from it the response due to the second-order operator.

The linear, bilinear and \( p^{th} \)-linear operators \( H_1[\delta(t-n_1\Delta t)] \), \( H_2[\delta(t-n_1\Delta t),\delta(t-n_2\Delta t)] \), \( \ldots \), \( H_p[\delta(t-n_1\Delta t),\delta(t-n_2\Delta t), \ldots,\delta(t-n_p\Delta t)] \) are commonly referred to as Volterra kernels and designated as \( h_1(t-n_1\Delta t),h_2(t-n_1\Delta t,t-n_2\Delta t),\ldots,h_p(t-n_1\Delta t,t-n_2\Delta t,\ldots,t-n_p\Delta t) \) respectively.
Hence:

\begin{align*}
  y(t) &= H_1[x(t)] + H_2[x(t)] + \cdots + H_p[x(t)] \\
  y(t) &= \sum_{n=1}^{N} x(n\Delta t) h_1(t - n\Delta t) \\
  &\quad + \sum_{n_1=1}^{N} \sum_{n_2=1}^{N} x(n_1\Delta t) x(n_2\Delta t) h_2(t - n_1\Delta t, t - n_2\Delta t) \\
  &\quad \vdots \\
  &\quad + \sum_{n_1=1}^{N} \cdots \sum_{n_p=1}^{N} x(n_1\Delta t) x(n_2\Delta t) \cdots x(n_p\Delta t) h_p(t - n_1\Delta t, t - n_2\Delta t, \cdots, t - n_p\Delta t) \\
\end{align*}

(2.17)

### 2.1.3 Numerical Example of a Nonlinear SDOF System

To illustrate the Volterra modeling principles of SDOF nonlinear systems, consider the following second-order anti-symmetrically nonlinear system:

\begin{equation}
  \ddot{y} + \dot{y} + y + 0.1y^2 = f(t)
\end{equation}

The equation was discretized using a second-order backward in time finite difference (FD) scheme using a time step of $2\pi/30$:

\begin{equation}
  y(t_\ell) = \frac{1}{\Delta t} \left[ -y(t_{\ell-3}) + 4y(t_{\ell-2}) - 5y(t_{\ell-1}) \right] - \frac{1}{2\Delta t} \left[ y(t_{\ell-2}) - 4y(t_{\ell-1}) \right] + f(t_\ell) \\
  \frac{2}{\Delta t^2} + \frac{3}{2\Delta t} + 1 + 0.1y(t_{\ell-1})
\end{equation}

(2.19)

Although such a large time step results in fairly high numerical damping, for our purposes of demonstrating the modeling principles of the Volterra series, the selected time step is satisfactory. The finite difference equation of Equation (2.19) including all subsequent finite difference equations were solved using MathCAD version 13.1
We model the output of this nonlinear system using a second-order Volterra system $N_2$. Figure 2 below shows the first-order $h_1$ and second-order $h_2$ Volterra kernels of this system computed using the identification procedure outlined in Section 2.1.2. Only three terms of the second-order kernel are plotted: the 1st, 5th and 10th terms, which correspond to inputs $f(t) = \delta(t-0\Delta t) + \delta(t-0\Delta t) = 2\delta(t-0\Delta t)$, $f(t) = \delta(t-0\Delta t) + \delta(t-4\Delta t)$, and $f(t) = \delta(t-0\Delta t) + \delta(t-9\Delta t)$ respectively. Figure 3 shows the output of the nonlinear system due to a unit sinusoidal input at $\omega=1$ rad/s.

![Figure 2: Volterra Kernels of SDOF System](image1)

![Figure 3: Finite Difference Output of SDOF System](image2)
Figure 4 shows the absolute errors of the Volterra models from the finite difference outputs. Two curves are plotted; the thick gray line plots the error of the Volterra model, which used the first-order kernel $h_1$ only, and the thick black line, which plots the error of the Volterra model, which used both the first and second-order kernels $h_1$ and $h_2$. The improvement in accuracy is substantial.

![Figure 4: Single-Input Volterra Model Errors](image)

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
In Figure 5, the frequency content of the finite difference and Volterra steady-state outputs are shown. As expected, the finite difference output contains integer harmonics of the input frequency, $\omega=1$ rad/s. Also plotted are the frequencies modeled by the Volterra model. The first-order kernel $h_1$ captures the fundamental, while the second-order kernel $h_2$ captures the $\omega=0$ rad/s (nonlinear contribution to the static response) and $\omega=2$ rad/s harmonics.

![Figure 5: Frequency Content of SDOF System Volterra Models Due to Unit Sinusoidal at $\omega=1$ rad/s](image)

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
2.2 Multi-Input Volterra Theory

2.2.1 Input-Output Characteristics of MDOF Systems

Because our application of the multi-input Volterra theory is limited to the two-degree-of-freedom airfoil, we carry out the following discussion and theoretical derivations in two degrees of freedom.

For illustration purposes, let’s consider the following two-degree-of-freedom nonlinear system:

\[
\begin{align*}
y_0 + y_0 + k_0 y_0^2 + k_{00} y_0^2 + k_{01} y_1 = f_0(t) \\
y_1 + y_1 + k_0 y_1^2 + k_{11} y_1^2 + k_{01} y_0 = f_1(t)
\end{align*}
\]

(2.20)

where:

\[k_0, k_1 >> k_{00}, k_{11}, k_{01}\]

(2.21)

For simplicity we set the total output equal to:

\[y = y_0 + y_1\]

(2.22)

The two-degree-of-freedom system of Equation (2.20) contains two types of nonlinear terms. The terms \(k_{00}y_0^2\) and \(k_{11}y_1^2\) are called direct nonlinear terms, while the \(k_{01}y_0\) and \(k_{01}y_1\) are called cross nonlinear terms. In the time domain, such a two-degree-of-freedom nonlinear system is one whose operation on each individual impulse from the input signal is a function of past impulses from both degrees of freedom. In the frequency domain, such a two-degree-of-freedom nonlinear system is one whose output due to inputs \(f_0(t) = \cos(\omega_0 t)\) and \(f_1(t) = \cos(\omega_1 t)\) contains both integer harmonics and intermodulation harmonics. Integer harmonics are caused by the direct nonlinear terms \(k_{00}y_0^2\) and \(k_{11}y_1^2\) and behave in the same way as the harmonics of SDOF nonlinear systems. Intermodulation harmonics on the other hand are caused by the cross nonlinear terms \(k_{01}y_0\) and \(k_{01}y_1\), and are of frequencies \(\omega_0 + \omega_1, \omega_0 - \omega_1, 2\omega_0 - \omega_1, \omega_0 - 2\omega_1, \text{etc.}\)

Similar to the integer harmonics of SDOF systems, only the first few intermodulation harmonics tend to be significant. In most instances, the \(\omega_0 + \omega_1, \omega_0 - \omega_1\) harmonics dominate. If the conditions of equation (2.21) are not met, no known general output characteristics exist.
2.2.2 Volterra Method for MDOF nonlinear systems

The extension of the Volterra theory to MDOF systems is credited to Worden et al (1997). The multi-input Volterra series is an infinite series of direct and cross Volterra operators. A nonlinear MDOF system containing direct nonlinearities and a cross nonlinearity can be modeled using an infinite sum of even and odd ordered direct Volterra operators and an infinite sum of cross Volterra operators.

Similarly to the theoretical derivation of single-input Volterra theory, this section begins with a description of impulse functions and mathematical operators. Although the derivation of the theory is carried out for a two-degree-of-freedom system, once the fundamentals are understood, the extension to arbitrary degrees of freedom will become trivial.

Recall that any physically realizable function can be approximated by a summation of discrete impulses from \( t=0 \) to \( t=N\Delta t \):

\[
x(t) = \sum_{n=1}^{N} x(n\Delta t)\delta(t-n\Delta t)
\]

(2.23)

However, now let's consider a two-degree-of-freedom system where two input signals exist. These are designated as \( x_0(t) \) and \( x_1(t) \):

\[
x_0(t) = \sum_{n=1}^{N} x_0(n\Delta t)\delta_0(t-n\Delta t)
\]

\[
x_1(t) = \sum_{n=1}^{N} x_1(n\Delta t)\delta_1(t-n\Delta t)
\]

(2.24)

where \( \delta_0 \) and \( \delta_1 \) are impulse functions corresponding to input signals \( x_0(t) \) and \( x_1(t) \). When a general two-degree-of-freedom system receives such input signals, \( x_0(t) \) and \( x_1(t) \), it will operate upon these individual impulses yielding the output signal, \( y_0(t) \) and \( y_1(t) \). However, for simplicity, let's consider the output \( y(t)=y_0(t)+y_1(t) \). Hence, a system can be interpreted as a collection of mathematical operators which map the input signals onto the output space. The procedure for defining these operators is identical to the previous section. However, now we need to remember that the input is composed of two separate inputs.
The most fundamental operator called the first-order operator (designated as $T_1$) is one whose response to a combination of impulses is a linear operation on the individual impulses:

$$y(t) = T_1[x_0(t) + x_1(t)]$$

$$= T_1\left[\sum_{n=1}^{N} x_0(n\Delta t)\delta_0(t-n\Delta t) + \sum_{n=1}^{N} x_1(n\Delta t)\delta_1(t-n\Delta t)\right]$$  \hspace{1cm} (2.25)

Now we designate that part of the first-order operator which operates on $x_0(t)$ as $T_1^0$, and the part which operates on $x_1(t)$ we designate as $T_1^1$:

$$y(t) = T_1^0\left[\sum_{n=1}^{N} x_0(n\Delta t)\delta_0(t-n\Delta t) + \sum_{n=1}^{N} x_1(n\Delta t)\delta_1(t-n\Delta t)\right]$$

$$= T_1^0\left[\sum_{n=1}^{N} x_0(n\Delta t)\delta_0(t-n\Delta t) + T_1^1\left[\sum_{n=1}^{N} x_1(n\Delta t)\delta_1(t-n\Delta t)\right]\right]$$  \hspace{1cm} (2.26)

$$= \sum_{n=1}^{N} x_0(n\Delta t)T_1^0\{\delta_0(t-n\Delta t)\} + \sum_{n=1}^{N} x_1(n\Delta t)T_1^1\{\delta_1(t-n\Delta t)\}$$

A second-order operator, which we designate as $T_2$, is one whose output to a set of impulses is a bilinear operation on the individual impulses:

$$y(t) = T_2[x_0(t) + x_1(t)]$$

$$= T_2\left[\sum_{n=1}^{N} x_0(n\Delta t)\delta_0(t-n\Delta t) + \sum_{n=1}^{N} x_1(n\Delta t)\delta_1(t-n\Delta t)\right]$$  \hspace{1cm} (2.27)

As we know from the previous section, the bilinear operator has two inputs. In a SDOF system the only combinations of two inputs the operator could process were impulses applied one after the other separated by varying time intervals. However, in the two-degree-of-freedom system the operator has a choice of operating on two separate impulses; both from $x_0(t)$, both from $x_1(t)$ or alternatively, one from $x_0(t)$ and one from $x_1(t)$. Hence, we divide the operator into three parts. $T_2^{00}$ is used to designate the part of the second-order operator which operates on both impulses from the $x_0(t)$ input. $T_2^{11}$ is used to designate the part of the second-order operator which operates on both impulses from the $x_1(t)$ input. Finally, $T_2^{01}$ is used to designate the part of the second-order operator which operates on inputs from both $x_0(t)$ and $x_1(t)$. 

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
\[ y(t) = T_2 \left[ \sum_{n=1}^{N} x_0(n \Delta t) \delta_0(t - n \Delta t) + \sum_{n=1}^{N} x_1(n \Delta t) \delta_1(t - n \Delta t) \right] 
\]
\[ = \sum_{n_1=1}^{N} \sum_{n_2=1}^{N} T_{2}^{00} \{ x_0(n_1 \Delta t) \delta_0(t - n_1 \Delta t), x_0(n_2 \Delta t) \delta_0(t - n_2 \Delta t) \} 
\]
\[ + \sum_{n_1=1}^{N} \sum_{n_2=1}^{N} T_{2}^{11} \{ x_1(n_1 \Delta t) \delta_1(t - n_1 \Delta t), x_1(n_2 \Delta t) \delta_1(t - n_2 \Delta t) \} 
\]
\[ + \sum_{n_1=1}^{N} \sum_{n_2=1}^{N} T_{2}^{01} \{ x_0(n_1 \Delta t) \delta_0(t - n_1 \Delta t), x_1(n_2 \Delta t) \delta_1(t - n_2 \Delta t) \} 
\]
\[ = \sum_{n_1=1}^{N} \sum_{n_2=1}^{N} x_0(n_1 \Delta t) x_0(n_2 \Delta t) T_{2}^{00} \{ \delta_0(t - n_1 \Delta t), \delta_0(t - n_2 \Delta t) \} 
\]
\[ + \sum_{n_1=1}^{N} \sum_{n_2=1}^{N} x_1(n_1 \Delta t) x_1(n_2 \Delta t) T_{2}^{11} \{ \delta_1(t - n_1 \Delta t), \delta_1(t - n_2 \Delta t) \} 
\]
\[ + \sum_{n_1=1}^{N} \sum_{n_2=1}^{N} x_0(n_1 \Delta t) x_1(n_2 \Delta t) T_{2}^{01} \{ \delta_0(t - n_1 \Delta t), \delta_1(t - n_2 \Delta t) \} 
\]
\[ \tag{2.28} \]

We can continue and generalize the process for a p\textsuperscript{th}-order operator; however, the arithmetic gets tedious. Because we will be modeling the transonic airfoil as a second-order, multi-input Volterra system, we only need the second-order operator. However, the basics behind this procedure become obvious. It is worthwhile to stress that the p\textsuperscript{th}-order operator will have p+1 combinations of the p\textsuperscript{th}-linear operation.

When all operators \( T_1^{0}, T_1^{1}, T_2^{00}, T_2^{11}, T_2^{01}, \ldots, T_p^{p-p} \) are time-invariant, they are called Volterra operators and are designated by \( H_1^{0}, H_1^{1}, H_2^{00}, H_2^{11}, H_2^{01}, \ldots, H_p^{p-p} \). By assembling these operators in a series, the p\textsuperscript{th}-order multi-input Volterra series \( N_p \) is created.
Using the previous argument, we only require the second-order Volterra series, \( N_2 \):

\[
y(t) = N_2 \{ x_0(t) + x_1(t) \}
\]

\[
= H_0^0[x_0(t)] + H_1^1[x_1(t)] + H_2^{00}[x_0(t)] + H_2^{11}[x_1(t)] + H_2^{01}[x_0(t) + x_1(t)]
\]

\[
= \sum_{n=1}^{N} x_0(n\Delta t)H_0^0(\delta_0(t-n\Delta t)) + \sum_{n=1}^{N} x_1(n\Delta t)H_1^1(\delta_1(t-n\Delta t))
\]

\[
+ \sum_{n_1=1}^{N} \sum_{n_2=1}^{N} x_0(n_1\Delta t)x_0(n_2\Delta t)H_2^{00}(\delta_0(t-n_1\Delta t), \delta_0(t-n_2\Delta t))
\]

\[
+ \sum_{n_1=1}^{N} \sum_{n_2=1}^{N} x_1(n_1\Delta t)x_1(n_2\Delta t)H_2^{11}(\delta_1(t-n_1\Delta t), \delta_1(t-n_2\Delta t))
\]

\[
+ \sum_{n_1=1}^{N} \sum_{n_2=1}^{N} x_0(n_1\Delta t)x_1(n_2\Delta t)H_2^{01}(\delta_0(t-n_1\Delta t), \delta_1(t-n_2\Delta t))
\]

(2.29)

A model of a MDOF nonlinear system using the \( p^{th} \)-order multi-input Volterra series \( N_p \) is called a \( p^{th} \)-order multi-input Volterra system. As stated previously, both the integer and intermodulation harmonics can be modeled using multi-input Volterra operators. This property for the integer harmonics has been demonstrated in the previous section. Here we demonstrate how the intermodulation harmonics are captured. Observing Equation (2.9) closely, we see that the \( H_2^{01} \) operator contains the term \( x_0(t)x_1(t) \). Hence for inputs \( x_0(t) = \cos(\omega_0 t) \) and \( x_1(t) = \cos(\omega_1 t) \) the frequency content of the output of this operator will be generated by the product \( \cos(\omega_0 t)\cos(\omega_1 t) \), in other words, \( \omega_1 + \omega_0 \) and \( \omega_1 - \omega_0 \).

The identification procedure of the bilinear operator is similar to the method presented in the previous section. However, now the bilinear operator comes in three configurations:

\[
2H_2^{00} \{ \delta_0(t-\tau_1\Delta t), \delta_0(t-\tau_2\Delta t) \} = N_2 \{ \delta_0(t-\tau_1\Delta t) + \delta_0(t-\tau_2\Delta t) \}
\]

\[
- N_2 \{ \delta_0(t-\tau_1\Delta t) + \delta_0(t-\tau_2\Delta t) \}
\]

(2.30)

\[
2H_2^{11} \{ \delta_1(t-\tau_1\Delta t), \delta_1(t-\tau_2\Delta t) \} = N_2 \{ \delta_1(t-\tau_1\Delta t) + \delta_1(t-\tau_2\Delta t) \}
\]

\[
- N_2 \{ \delta_1(t-\tau_1\Delta t) + \delta_1(t-\tau_2\Delta t) \}
\]

(2.31)

\[
2H_2^{01} \{ \delta_0(t-\tau_1\Delta t), \delta_1(t-\tau_2\Delta t) \} = N_2 \{ \delta_0(t-\tau_1\Delta t) + \delta_1(t-\tau_2\Delta t) \}
\]

\[
- N_2 \{ \delta_0(t-\tau_1\Delta t) + \delta_1(t-\tau_2\Delta t) \}
\]

(2.32)

The linear, bilinear and \( p^{th} \)-linear operators \( H_1^0, H_1^1, H_2^{00}, H_2^{11}, H_2^{01}, \ldots, H_p^{p\ldots p} \) are often called Volterra kernels; designated as \( h_1^0, h_1^1, h_2^{00}, h_2^{11}, h_2^{01}, \ldots, h_p^{p\ldots p} \). Volterra kernels \( h_p^{p\ldots p} \) where \( p \ldots p \) are all either 0 or 1 are called direct-kernels while Volterra
kernels \( h_p \) where \( p \) are combinations of 0 and 1 are called cross-kernels. Hence, a second-order Volterra system \( N_2 \) is composed of two first-order direct-kernels \( h_1^0, h_1^1 \), two second-order direct-kernels \( h_2^{00}, h_2^{11} \) and one second-order cross-kernel \( h_2^{01} \).

\[
y(t) = N_2[x_0(t) + x_1(t)]
\]
\[
= H_1^0[x_0(t)] + H_1^1[x_1(t)] + H_2^{00}[x_0(t)] + H_2^{11}[x_1(t)] + H_2^{01}[x_0(t) + x_1(t)]
\]
\[
= \sum_{n=1}^{N} x_0(n \Delta t) h_1^0(t - n \Delta t) + \sum_{n=1}^{N} x_1(n \Delta t) h_1^1(t - n \Delta t)
\]
\[
+ \sum_{n_1=1}^{N} \sum_{n_2=1}^{N} x_0(n_1 \Delta t) x_0(n_2 \Delta t) h_2^{00}(t - n_1 \Delta t, t - n_2 \Delta t)
\]
\[
+ \sum_{n_1=1}^{N} \sum_{n_2=1}^{N} x_1(n_1 \Delta t) x_1(n_2 \Delta t) h_2^{11}(t - n_1 \Delta t, t - n_2 \Delta t)
\]
\[
+ \sum_{n=1}^{N} \sum_{n_2=1}^{N} x_0(n \Delta t) x_1(n_2 \Delta t) h_2^{01}(t - n_1 \Delta t, t - n_2 \Delta t)
\]
\[ (2.33) \]

2.2.3 Numerical Example of MDOF Nonlinear System

To illustrate the multi-input Volterra theory of nonlinear systems, consider the following two-degree-of-freedom, second-order nonlinear system:

\[
\begin{align*}
\dot{y}_0 + y_0 + y_0 + 0.1 y_0^2 + 0.1 y_1 &= f_0(t) \\
\dot{y}_1 + y_1 + y_1 + 0.1 y_1^2 + 0.1 y_0 &= f_1(t)
\end{align*}
\]
\[ (2.34) \]

with the output:

\[
y = y_0 + y_1
\]
\[ (2.35) \]

As in the SDOF numerical example, the equation was discretized using a second-order backward in time finite difference scheme with a time step of \( 2\pi/30 \):

\[
y_0(t_i) = \frac{1}{\Delta t^2} \left[ -y_0(t_{i-2}) + 4y_0(t_{i-1}) - 5y_0(t_{i-2}) - 4y_0(t_{i-1}) + f_0(t_i) \right] - \frac{1}{2\Delta t} \left[ y_0(t_{i-2}) - 4y_0(t_{i-1}) \right] + f_0(t_i)
\]
\[
\frac{2}{\Delta t^2} + \frac{3}{2\Delta t} + 1 + 0.1 y_0(t_{i-1}) + 0.1 y_1(t_{i-1})
\]
\[
y_1(t_i) = \frac{1}{\Delta t^2} \left[ -y_1(t_{i-2}) + 4y_1(t_{i-1}) - 5y_1(t_{i-2}) - 4y_1(t_{i-1}) + f_1(t_i) \right] - \frac{1}{2\Delta t} \left[ y_1(t_{i-2}) - 4y_1(t_{i-1}) \right] + f_1(t_i)
\]
\[
\frac{2}{\Delta t^2} + \frac{3}{2\Delta t} + 1 + 0.1 y_1(t_{i-1}) + 0.1 y_0(t_{i-1})
\]
\[ (2.36) \]

20
We model the output of this two-degree-of-freedom nonlinear system using a second-order, multi-input Volterra system $N_2$. Since both equations of motion for $y_0(t)$ and $y_1(t)$ under the individually applied inputs $f_0(t)$ and $f_1(t)$ are identical to the SDOF nonlinear system studied in the previous sections, our first and second-order direct-kernels are identical to the kernels shown in Figure 2 and Figure 3. Hence $h_1^0=h_1^1=h_1$ and $h_2^{00}=h_2^{11}=h_2$. However, the present nonlinear system also contains a cross nonlinearity which couples the two inputs. Because of this term, the Volterra system now contains a cross-kernel $h_2^{01}$. Figure 6 shows three components of the Volterra cross-kernel calculated using the procedure outlined in Section 2.2.2:

![Figure 6: Volterra Cross-Kernel of MDOF System](image)

As stated previously, the following results were computed using MathCAD 13.1. Now, let's investigate how a second-order multi-input Volterra system models the output of this two-degree-of-freedom nonlinear equation. Figure 7 below shows the finite difference output due to inputs $f_0(t)=\cos(\omega t)$ and $f_1(t)=\cos(0.8\omega t)$ where $\omega=1\text{rad/s}$:

![Figure 7: Finite Difference Output of MDOF System](image)

Figure 8 illustrates the absolute errors of the Volterra models from the finite difference (FD) results. Three curves are plotted; the thick gray line plots the error of the Volterra model which used the first-order kernels $h_1^0, h_1^1$ only. The dotted line shows the error of
the Volterra model using both the first and second-order direct-kernels \( h_1^0, h_1^1, h_2^{00}, h_2^{11} \), while the thin black line depicts the errors of the Volterra model which utilizes both the direct and cross-kernels \( h_1^0, h_1^1, h_2^{00}, h_2^{11}, h_2^{01} \). The improvement in accuracy through the addition of the cross-kernel is clearly substantial.

![Figure 8: Multi-Input Volterra Model Errors](image)

Figure 8: Multi-Input Volterra Model Errors

Frequency content of the finite difference and Volterra model outputs are shown in Figure 9. As expected, the first-order kernels \( h_1^0 \) and \( h_1^1 \) capture the two fundamentals \( 1\omega \) and \( 0.8\omega \), respectively. The second-order direct-kernels \( h_2^{00} \) and \( h_2^{11} \) capture the \( 0\omega \) (nonlinear contribution to the static response), \( 1.6\omega \) and \( 2\omega \) harmonics while the cross-kernel \( h_2^{01} \) captures the intermodulation harmonics \( 0.2\omega \) and \( 1.8\omega \).

![Figure 9: Frequency Content of MDOF System Due to Unit Sinusoidal Inputs at \( \omega_0=1\text{rad/s} \) and \( \omega_1=0.8\text{rad/s} \)](image)

Figure 9: Frequency Content of MDOF System Due to Unit Sinusoidal Inputs at \( \omega_0=1\text{rad/s} \) and \( \omega_1=0.8\text{rad/s} \)

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
2.3 Multi-Input Volterra Model of the Transonic Two-Dimensional Airfoil

In this section we present the second-order multi-input Volterra system model of the two dimensional, two-degree-of-freedom unsteady airfoil. The equations are a direct application of the theoretical derivations of the previous sections.

A two dimensional airfoil can produce lift through two degrees of freedom; pitch $\alpha$, and a heave velocity $\dot{h}$:

![Airfoil Input Coordinate System](image)

**Figure 10: Airfoil Input Coordinate System**

The main weakness of previous Volterra models of the transonic airfoil is the fact that only direct-kernels $h_1^a$, $h_1^h$, $h_2^{aa}$, $h_2^{hh}$ were considered. In other words, the Volterra model of the transonic airfoil was as follows:

$$y(t) = N_2[\alpha(t) + \dot{h}(t)]$$

$$= \mathbf{H}_1^a[\alpha(t)] + \mathbf{H}_1^h[\dot{h}(t)] + \mathbf{H}_2^{aa}[\alpha(t)] + \mathbf{H}_2^{hh}[\dot{h}(t)]$$

$$= \sum_{n=1}^{N} \alpha(n\Delta t)h_1^a(t - n\Delta t) + \sum_{n=1}^{N} \dot{h}(n\Delta t)h_1^h(t - n\Delta t)$$

$$+ \sum_{n_1=1}^{N} \sum_{n_2=1}^{N} \alpha(n_1\Delta t)\alpha(n_2\Delta t)h_2^{aa}(t - n_1\Delta t, t - n_2\Delta t)$$

$$+ \sum_{n_1=1}^{N} \sum_{n_2=1}^{N} \dot{h}(n_1\Delta t)\dot{h}(n_2\Delta t)h_2^{hh}(t - n_1\Delta t, t - n_2\Delta t)$$

(3.1)

Hence, the model was simply a superposition of single-input Volterra systems with respect to each degree-of-freedom. As we have seen from discussion in the previous sections, this approach is considered a major simplification.
Hence, the aim of the work summarized in this thesis is the application of multi-input Volterra theory to the present aerodynamic system. The second-order multi-input Volterra system model of the transonic airfoil is as follows:

\[
\begin{align*}
  y(t) &= N_2[\alpha(t) + \dot{h}(t)] \\
  &= \mathbf{H}_1^{\alpha}[\alpha(t)] + \mathbf{H}_1^{\dot{h}}[\dot{h}(t)] + \mathbf{H}_2^{\alpha\alpha}[\alpha(t)] + \mathbf{H}_2^{\alpha\dot{h}}[\alpha(t), \dot{h}(t)] + \mathbf{H}_2^{\dot{h}\dot{h}}[\dot{h}(t)] \\
  &= \sum_{n=1}^{N} \alpha(n\Delta t)h_2^{\alpha}(t-n\Delta t) + \sum_{n=1}^{N} \dot{h}(n\Delta t)h_2^{\dot{h}}(t-n\Delta t) \\
  &\quad + \sum_{n_1=1}^{N} \sum_{n_2=1}^{N} \alpha(n_1\Delta t)\alpha(n_2\Delta t)h_2^{\alpha\alpha}(t-n_1\Delta t, t-n_2\Delta t) \\
  &\quad + \sum_{n_1=1}^{N} \sum_{n_2=1}^{N} \dot{h}(n_1\Delta t)\dot{h}(n_2\Delta t)h_2^{\dot{h}\dot{h}}(t-n_1\Delta t, t-n_2\Delta t) \\
  &\quad + \sum_{n_1=1}^{N} \sum_{n_2=1}^{N} \alpha(n_1\Delta t)\dot{h}(n_2\Delta t)h_2^{\alpha\dot{h}}(t-n_1\Delta t, t-n_2\Delta t) \\
  &\quad + \sum_{n_1=1}^{N} \sum_{n_2=1}^{N} \alpha(n_1\Delta t)h(n_2\Delta t)h_2^{\alpha h}(t-n_1\Delta t, t-n_2\Delta t)
\end{align*}
\]

The output \( y(t) \) of the pitching and heaving airfoil has two components, the lift coefficient \( C_l \) and moment coefficient \( C_m \) as illustrated in Figure 11.

![Airfoil Output Coordinates](image)

**Figure 11: Airfoil Output Coordinates**

### 2.4 Description of Test Case

As discussed in the previous section, we are modeling our transonic airfoil as a second-order multi-input Volterra system. Hence, as discussed in Section 2.2.2, we must ensure that the test case we investigate contains predominantly anti-symmetric nonlinearities. In terms of a transonic airfoil, a geometrically asymmetric airfoil oscillating around a zero initial angle of attack or a symmetric airfoil oscillating around a nonzero angle of attack will both contain predominantly anti-symmetric nonlinearities. This is because both instances will result in anti-symmetric shockwave distributions about the top and bottom
surfaces of the airfoil. For our purposes we choose a symmetric NACA 0012 airfoil oscillating about a non-zero static angle of attack. Hence, we define our motions as:

\[
\begin{align*}
\alpha(t) &= \bar{\alpha} + \alpha_0 \sin(\omega_{\alpha} t) \\
\h(t) &= h_0 \sin(\omega_{h} t + \theta)
\end{align*}
\]

where \( \bar{\alpha} \) is static pitch and \( \alpha_0 \) and \( h_0 \) are the amplitudes of the dynamic pitching and heaving motions. \( \theta \) is a phase shift of the heave motion and the angular frequencies \( \omega_{\alpha} \) and \( \omega_{h} \) are related to the reduced frequency \( k \) by:

\[
k = \frac{\omega c}{2U_\infty}
\]

We define the constant forward velocity of the airfoil in terms of the Mach number \( M \). When assigning values to parameters in Equations (3.3) and (3.4), several requirements must be satisfied:

1. For the prescribed motion, reliable experimental data should exist in order to validate the CFD output.
2. The nonlinear lift and moment response due to these motions must be weakly nonlinear. In other words, only low transonic conditions (weak shockwaves) will be effectively modeled using the second-order multi-input Volterra system.

To the author's knowledge, no experimental data for simultaneous pitch and heave motions at transonic conditions exists. However, the experimental data is numerous for forced pitch motions alone. An often cited experimental data set of lift and moment values for forced pitch motions at transonic conditions is the AGARD (1982) database. The CT2 test case used the following forced pitch motion:

\[
\alpha(t) = 3.16^\circ + 4.59^\circ \sin(\omega_{\alpha} t)
\]

Due to the large pitch oscillations, a Mach number of only 0.6 at a pitch reduced frequency of \( k_{\alpha}=0.081 \) is required to create transonic flow conditions. However, the prescribed AGARD test case motion creates a strong shockwave at maximum pitch-up conditions resulting in a strongly nonlinear aerodynamic system. Since the present Volterra model can only handle weak nonlinearities, we must decrease the dynamic pitch motion so that only weak shock waves appear. Observing the pressure contours from the AGARD reports, this condition is satisfied at dynamic pitch magnitudes of about \( \alpha_0=1^\circ \).
Now, since the present airfoil motion also includes heave, we must also define the dynamic heave amplitude. As stated previously, the aim of this thesis is to present a method of modeling MDOF aerodynamic systems using multi-input Volterra theory. The main difference between the application of multi-input Volterra theory over the single-input Volterra theory is the identification of the cross-kernel. Since we want to focus on this cross-kernel as much as possible, it would be helpful if both the pitch and heave direct-kernel magnitudes were approximately equal. In other words, if the relative strength of the aerodynamic response harmonics to the fundamental would be equal for both motions, thereby allowing for direct comparisons between the cross-kernel and the direct-kernels, including their relative contributions to the aerodynamic output. Since no experimental data is available for the present case, we will define instead a CFD data set running simulation cases with several heave amplitudes and compare their harmonic components to the corresponding harmonics generated by pitch oscillations alone. Through this process, it was found that a dynamic heave magnitude of $h_0 = 14\%$ of the chord, at $k_h = k_a$, would achieve this requirement. Hence, the following dynamic amplitudes have been chosen:

\[
\begin{align*}
\alpha(t) &= 3.16^\circ + 1^\circ \sin(\omega_a t) \\
h(t) &= 0.14c \sin(\omega_h t + \theta)
\end{align*}
\] (3.6)

For these dynamic motions, two pitch and heave frequency ratios were analyzed $k_h/k_a = 1$ and $k_h/k_a = 0.8$, where $k_a = 0.081$.

The only parameter remaining is the heave phase offset $\theta$. However this choice is not arbitrary and interesting results can be obtained at different phase shifts. This is because the associated nonlinearities resulting from the combined motions are a function of this phase difference. To understand the reason for this, we must realize that the nonlinearities in this system are primarily due to the shockwave; the greater the shockwave strength the greater the nonlinearities and vice versa. Likewise, the shockwave strength is directly proportional to the total lift of the airfoil. Since both pitch and heave motions can increase or decrease the lift of the airfoil, combining the motions so that the lift from each motion is in phase will tend to amplify the lift, shockwave strength and total nonlinearities in the system. On the other hand, combining the motions so that the lift produced from each motion is $180^\circ$ out of phase will tend to decrease the

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
lift, shockwave strength and total nonlinearities in the system. However, since the moment coefficient is in general not in phase with the lift, the magnitude of the moment coefficient will not follow the amplification trends of the lift values just described. However, because the nonlinearities in the moment coefficient are also due to the shockwaves strength, motions which tend to amplify the lift will tend to increase the moment coefficient nonlinearities, while motions that tend to cancel out the lift will tend to decrease the moment coefficient nonlinearities.

For illustration purposes consider a quasi-steady sinusoidal motion of both pitch and heave degrees of freedom at the same frequency. In the pitch case, the lift due to pitch will be in phase with the pitch motion. However, since lift is produced by a heave velocity, the lift due to heave lags the heave motion by 90°. Hence, for the lift of both motions to be in phase, the heave motion must lead the pitch motion by 90°. Conversely, for the lift to be 180° out of phase, heave must lag the pitch motion by 90°. Since the chosen reduced frequency is low, the considerations taken from this quasi-steady model remain applicable for the fully unsteady flow. Hence, to analyze the cross-kernels ability to predict the coupling nonlinearities for both referred scenarios, for the $k_h/k_a=1$ case, two heave phase shifts, $\theta=90^\circ$ and $\theta=-90^\circ$ are utilized. On the other hand, for the $k_h/k_a=0.8$ case as the motions will experience both in-phase and out-of-phase lift components, no matter what the heave offset, we set $\theta=0^\circ$ for convenience.

### 2.5 CFD Using the Parallel Multi Block Solver

The synthesis of the second-order Volterra model was carried out using a CFD code called Parallel Multi Block (PMB) developed at the University of Glasgow specifically for the purposes of transonic time marching aeroelastic analysis. The code is based on a cell-centered finite volume method. Osher upwind Finite Difference Splitting scheme together with a MUSCL variable interpolation is used to discretize the convective terms to provide third-order spatial accuracy. The method uses an implicit real-time discretization, and at each time step the solution marches in pseudo time using implicit methods; resulting in second-order accuracy in time. The aerodynamics of the airfoil is modeled using the inviscid Euler equations. For further details, refer to Dubuc et al (1997).
The NACA0012 airfoil domain is discretized using a C type 180x33 Euler grid with 130 nodes on the airfoil. The surface nodes are at a distance of approximately 0.001c off the airfoil surface. The mesh extends into the far field approximately 10c in all directions. The unsteady solutions are solved using 20 time steps per each period of the airfoil pitch oscillation. This choice of mesh and time step was based on several studies on mesh refinement done by Dubuc et al. (1997), which showed that no significant accuracy is gained at higher spatial or temporal mesh densities.
3. Results and Discussion

This section presents the various CFD and Volterra system model outputs for the unsteady transonic airfoil. Section 3.1 is a validation of the CFD software using the AGARD CT2 test case for comparison. Section 3.2 presents the identified Volterra kernels, including the Volterra cross-kernel. Finally, Section 3.3 and 3.4 contain all the Volterra system model and CFD outputs for the various simultaneous pitch and heave motions.

3.1 AGARD Test Case, CFD Validation

Figure 12 compares the experimental and CFD outputs of the AGARD test case as described by Equation (3.5); very good agreement is obtained. Errors are likely associated with the neglect of viscous forces and uncertainties in the experimental data.

![Figure 12: CFD Validation of AGARD Test Case](image.png)
3.2 Volterra Kernel Identification

As stated in the theoretical derivation of the Volterra series and further clarified in Appendix A, to identify the second-order kernels, one must apply impulses in both the pitch and heave velocity degrees of freedom. The outputs of the system due to these applied impulses are lift and moment. However, it was verified that only the moment part of the output is significantly nonlinear. Hence, for the sake of brevity we present here only the moment components of the kernels. Figure 13 shows the various kernels identified in the present work. Note that only the first three terms of the second-order kernel are shown, which, for the second-order cross-kernel $h_2^{\alpha h}$ for example, correspond to inputs $\delta_\alpha(t-0\Delta t)+\delta_h(t-0\Delta t)$, $\delta_\alpha(t-0\Delta t)+\delta_h(t-1\Delta t)$ and $\delta_\alpha(t-0\Delta t)+\delta_h(t-2\Delta t)$. The magnitudes of the $\alpha$ impulse and $h$ step (i.e. impulse in $\dot{h}$) inputs were taken as 0.5° and 0.02$c$, respectively.
Figure 13: Volterra Kernels of the Transonic Airfoil

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
3.3 Unsteady Results, $k_h/k_\alpha=1$

Figure 14 shows the moment coefficient due to individually applied pitch and heave motions:

$$\alpha(t) = 3.16^\circ + 1^\circ \sin(\omega_n t)$$
$$h(t) = 0.14c \sin(\omega_h t)$$

(3.7)

where $M=0.6$, $k_\alpha=0.081$ and $k_h/k_\alpha=1$. We notice that the Volterra model using both the first-order $h_1^x$ and second-order $h_2^{xx}$ kernels performs very well at modeling the nonlinear output. Please note that superscript $x$ in $h_1^x$ and $h_2^{xx}$ refers to either $\alpha$ and $\alpha\alpha$ or $h$ and $h\dot{h}$.

![Pitch and Heave Motions](image)

Figure 14: Individually Applied Pitch and Heave Motions, $k_h/k_\alpha=1$
Figure 15 is an illustration of the sinusoidal pitch and heave inputs. T identifies the period of the pitch and heave inputs. The A, B, C and D markings identify the points in time at which the pressure contours of Figure 16 were plotted.

Observing Figure 16 the lift due to pitch is in phase with the pitch motion as expected. We arrive to this conclusion because the pressure difference between the upper and lower surfaces is the largest at location B and the smallest at location D. In other words, lift is largest when pitch is largest and lift is smallest when pitch is smallest. As for the heave motion, as expected its lift response lags the heave motion by 90°. In the same way, we arrive at this conclusion because the pressure difference between the upper and lower surfaces is the largest at location A and the smallest at C. In other words, lift is largest 90° behind maximum heave position B and lift is smallest 90° behind minimum heave position D. This is because lift is a function of the time derivative of heave motion, $\dot{h}$, not the heave motion, $h$, itself. For both motions the appearance of a weak shock wave during their respective maximum lift periods is evident.
Figure 16: Surface Pressure Coefficients of Independent Pitch and Heave Motion CFD Outputs
Figure 17 shows the surface pressure coefficients at four instances of the airfoils simultaneous pitch and heave oscillations. However, this time the four points at which the pressure distributions are plotted are with respect to the pitch motion. As predicted in Section 2.4, when heave lags pitch by 90° the respective motions tend to cancel out each other’s produced lift, while when heave leads pitch by 90° the respective motions tend to amplify each other’s produced lift. This creates the overall reduction and amplification of the shockwave strength respectively, as is visible in Figure 17.
Figure 18 compares the moment coefficient produced by a superposition of individually applied motions and the simultaneous motion. We can clearly verify that due to the nonlinear nature of the aerodynamic response, the superposition of the individual responses does not accurately predict the simultaneous response.

![Graph showing the comparison between superimposed and simultaneous pitch and heave motions.](image)

**Figure 18: Superimposed and Simultaneous Pitch and Heave Motion CFD Outputs, \(k_h/k_\alpha=1\)**
Figure 19 compares the frequency content of the superimposed pitch and heave CFD outputs to the simultaneous pitch and heave CFD outputs. One can clearly see how a heave motion, which lags the pitch motion by 90°, tends to decrease the shockwave strength and the output harmonics (i.e. nonlinearities). While a heave motion, which leads the pitch motion by 90°, tends to amplify the shockwave strength and the output harmonics (i.e. nonlinearities).

![Diagram showing frequency content comparison](image)

**Figure 19: Frequency Content of Superimposed and Simultaneous Pitch and Heave CFD Outputs, $k_h/k_{\alpha}=1$**
Now the question becomes how well the cross-kernel can model this coupling nonlinearity. Figure 20 compares the superimposed and simultaneous outputs with the Volterra model using only the direct-kernels. As expected, the superimposed response is well predicted, but the simultaneous motion is not.

![Graph](image)

**Figure 20: Volterra Model Using Direct-Kernels Only Against Superimposed and Simultaneous Pitch and Heave CFD Outputs, $k_d/k_a=1$**

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Figure 21 illustrates how the inclusion of the cross-kernel significantly improves the accuracy of model of the simultaneous pitch and heave motions.

Figure 21: Volterra Model Using Direct-Kernels Only, and Direct and Cross-Kernels Against Simultaneous Pitch and Heave CFD Output, $k_h/k_a=1$
Figure 22 shows the absolute errors of the Volterra models in comparison with the simultaneous motion predicted by CFD. Although the improvement in accuracy is considerable for the case where heave lags pitch by 90°, the cross-kernel only manages to half the absolute error for the case where heave leads pitch by 90°. This behavior is to be expected however, because when heave leads pitch, the combined motions amplify the shockwave’s strength and, hence, amplify the nonlinearities as well. Since the second-order Volterra system model can only handle weakly nonlinear systems, an increase in nonlinearity will tend to increase the Volterra models errors.

Figure 22: Absolute Errors of Volterra Models Using Direct-Kernels Only, and both Direct and Cross-Kernels, $k_h/k_e=1$
Figure 23 compares the frequency content of the studied Volterra models and the CFD output reference. In both cases, the addition of the cross-kernel in the Volterra model increases the accuracy at which both the \(0k/ka\) (nonlinear contribution to the static response) and \(2k/ka\) harmonics are predicted.

**Figure 23: Frequency Content of Simultaneous CFD Output and Volterra Models Using Direct-Kernels Only, and Both Direct and Cross-Kernels, \(k_h/ka=1\)**
3.4 Unsteady Results, $k_h/k_\alpha=0.8$

Figure 24 shows the moment coefficient due to individually applied pitch and heave motions. We notice that the Volterra model using both the first and second-order kernels performs well at modeling the nonlinear outputs.

![Figure 24: Individually Applied Pitch and Heave Motions, $k_h/k_\alpha=0.8$](image)
Figure 25 below compares the superimposed pitch and heave CFD outputs and the simultaneous pitch and heave motion CFD output. However, in this case, the pitch and heave motions are at different frequencies, $k_h/k_a=0.8$. As the two excitation frequencies are close to each other, we can clearly notice the introduction of a beat frequency. From the previous analysis, we concluded that when the heave motion leads the pitch motion by 90°, the simultaneous CFD($\alpha+h$) output tend to peak higher than the superimposed CFD($\alpha$) + CFD($h$) output. Hence, we can conclude that at the time steps of approximately 80, 180 and 280 the heave motion leads the pitch motion by 90°. Conversely, we can conclude that heave lags the pitch motion by 90° at time steps approximately 130 and 230.

![Graph showing CFD outputs](image)

Figure 25: Superimposed and Simultaneous Pitch and Heave CFD Outputs, $k_h/k_a=0.8$
Figure 26 and Figure 27 illustrate how well the Volterra models capture the superimposed and simultaneous pitch and heave moment outputs. Although good at modeling the superimposed moment outputs, the direct-kernels fail to model the nonlinearities associated with the simultaneous motion.

Figure 26: Volterra Model Using Direct-Kernels Only Against Superimposed and Simultaneous Pitch and Heave Motion CFD Outputs, $k_p/k_a=0.8$

Figure 27: Volterra Model Using Direct-Kernels Only, and Both Direct and Cross-Kernels Against Simultaneous Pitch and Heave CFD Outputs, $k_p/k_a=0.8$
Figure 28 shows the absolute errors of the Volterra models. As in the $k_h/k_a=1$ case, the improvements in accuracy from the inclusion of the cross-kernel are greatest in regions where heave lags the pitch motion by 90°, and least around regions where heave leads by 90°. Overall, however, the improvement may be considered significant.

![Figure 28: Absolute Errors of Volterra Models Using Direct-Kernels, and Both Direct and Cross-Kernels, $k_h/k_a=0.8$](image)

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Figure 29 shows the frequency content of the simultaneous pitch and heave CFD output against the various frequencies captured by the individual Volterra kernels. We see that the first-order kernels successfully model the fundamentals, $1k/k_a$ and $0.8k/k_a$. The direct-kernels successfully capture the $0k/k_a$ (nonlinear contribution to the static response), $1.6k/k_a$ and $2k/k_a$ harmonics. On the other hand the cross-kernel identifies the intermodulation harmonics $0.2k/k_a$ and $1.8k/k_a$.

![Figure 29: Frequency Content of Combined CFD Output and Volterra Kernel Outputs, $k_h/k_a=0.8$](image)

Also, because the intermodulation harmonics have higher magnitudes compared to the integer harmonics $1.6k/k_a$ and $2k/k_a$, we can conclude that for this specific test case, the inclusion of the cross kernel was more beneficial then the inclusion of the second-order direct-kernels.
4. Conclusions

Previous applications of the Volterra theory of nonlinear systems to aerodynamic problems utilized superposition of the nonlinear modal responses. However, it is well known that such a linear approach to a nonlinear problem is generally incorrect. This thesis presented a method of modeling nonlinear aerodynamic systems using multi-input Volterra theory. The method was applied to a two-dimensional, two-degree-of-freedom transonic airfoil undergoing simultaneous forced pitch and heave harmonic oscillations. The so called Volterra cross-kernels were identified and shown to successfully model the lift and moment nonlinearities associated with the simultaneous motion. The improvements in accuracy over previous approaches which effectively ignored the cross-kernel by using the superposition principle were demonstrated.

Although the present work utilized the inviscid Euler aerodynamic models, the multi-input Volterra theory is entirely general and other models such as the Reynolds averaged Navier-Stokes models could be easily facilitated.
5. Future Work

The proposed method is also easily generalized to higher-order, MDOF problems including three-dimensional wings, where the modal coordinates would replace the degrees of freedom of the system. The only limiting factor of such an extension would be the availability of computational resources since a system with $n$ structural modes would require $n$ first-order kernels, $n$ second-order direct-kernels and $\sum_{i=1}^{n} (i-1)$ cross-kernels. However, since the flutter phenomenon in general involves a limited number of modes, the present method may still be considered attractive for flutter prediction at transonic speeds. For example, a system with three structural modes, the multi-input Volterra model would be:

$$y(t) = N_2[\xi_1(t) + \xi_2(t) + \xi_3(t)]$$

$$= H_1^{\xi_1}[\xi_1(t)] + H_1^{\xi_2}[\xi_2(t)] + H_1^{\xi_3}[\xi_3(t)]$$

$$+ H_2^{\xi_1}[\xi_1(t)] + H_2^{\xi_2}[\xi_2(t)] + H_2^{\xi_3}[\xi_3(t)]$$

$$+ H_2^{\xi_1}[\xi_1(t), \xi_2(t)] + H_2^{\xi_2}[\xi_1(t), \xi_3(t)] + H_2^{\xi_3}[\xi_2(t), \xi_3(t)]$$

(3.8)
References


Appendix A. Volterra Kernel Identification Procedure Using PMB

Step 1. Airfoil Mesh Generation

Due to the Euler inviscid aerodynamic model, a medium density spatial discretization is satisfactory. An implicit temporal discretization scheme used in the PMB CFD code grants similar coarseness capabilities in the temporal direction. For the purpose of the analysis summarized in this thesis, the NACA0012 airfoil domain was discretized using a C type 180x33 Euler grid with 130 nodes on the airfoil. The surface nodes were a distance of approximately 0.001c off the airfoil surface. The mesh extended to the far field approximately 10c in all directions. The unsteady solutions were solved using 20 time steps per each period of the airfoil pitch oscillation.

Because the PMB code is specifically tailored towards three-dimensional aircraft applications, the coordinates through which motions can be enforced is limited to rotations about the standard, pitch α, roll β and yaw ψ coordinate system. In other words, it is not possible to displace the mesh in a purely linear fashion as required for the heave motion. To overcome this limitation, the roll axis was offset from the airfoil mesh by a distance large enough to generate almost pure heave motions when rotations about the roll axis were applied. An offset of 1000c proved adequate for the required heave amplitudes of 0.14c.

Finally, in order to simulate two-dimensional flow conditions using the three-dimensional control volume formulization of the PMB code, periodic boundary conditions were applied to the one element thick mesh planes.
Step 2. Unsteady Input File Generation

For illustration purposes, an input file for one of the required inputs, \( \delta_a(t-0\Delta t) + \delta_h(t-2\Delta t) \), required in the Volterra cross-kernel identification procedure is shown in Figure AI.

![Figure AI: CMB Input File for Volterra Cross-kernel Identification](image)

As visible in Figure AI, columns B, C, E, I and K contain the \( t, \alpha, \beta, \dot{\alpha} \) and \( \dot{\beta} \) parameters respectively. As discussed in the previous step, the roll axis was offset by 1000c from the airfoil mesh in order to simulate pure heave motions. Hence, in order to achieve a step magnitude of 0.02c in \( h \) (i.e. an impulse in \( \dot{h} \), see Section 3), a roll deflection of approximately 0.02/1000rad is required. Also, please note that the required time separation of \( 2\Delta t \) in the input signal, \( \delta_a(t-0\Delta t) + \delta_h(t-2\Delta t) \), was achieved by applying rotations in \( \alpha \) and \( \dot{\beta} \) in rows 7 and 9 respectively.

Although it is possible to generate individual input files for the other required inputs, such as \( \delta_a(t-0\Delta t) + \delta_h(t-3\Delta t) \), \( \delta_a(t-0\Delta t) + \delta_h(t-4\Delta t) \), etc., the approach is inefficient. A more desirable method combines all the required inputs into one input file as shown in Figure AI.
After the flow is allowed to reach steady-state through several time intervals, the second set of impulses, \( \delta_a(t-3\Delta t) \), is applied. To ensure that the second set of impulses responses, \( \delta_a(t-0\Delta t)+\delta_a(t-3\Delta t) \), are not influenced by the first set, \( \delta_a(t-0\Delta t)+\delta_a(t-2\Delta t) \), a relaxation period in between is required. In Figure AII, the relaxation period is identified by the left brace, and includes rows 25-29. The relaxation period is initiated by returning the airfoil to its original \( h=0 \), heave position. After the flow is allowed to reach steady-state through several time intervals, the second set of impulses, \( \delta_a(t-0\Delta t)+\delta_a(t-3\Delta t) \), is applied.

<table>
<thead>
<tr>
<th>1</th>
<th>1000</th>
<th>0.001</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1000</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.001</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>none</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( t )</th>
<th>( \alpha )</th>
<th>( \beta )</th>
<th>( \dot{\alpha} )</th>
<th>( \dot{\beta} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1.963495</td>
<td>0</td>
<td>-0.004444</td>
</tr>
<tr>
<td>3</td>
<td>2.392693</td>
<td>0</td>
<td>-2.00E-05</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>7.953986</td>
<td>0</td>
<td>-2.00E-05</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>11.79097</td>
<td>0</td>
<td>-2.00E-05</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>13.74447</td>
<td>0</td>
<td>-2.00E-05</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>15.70996</td>
<td>0</td>
<td>-2.00E-05</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>17.87146</td>
<td>0</td>
<td>-2.00E-05</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>19.93495</td>
<td>0</td>
<td>-2.00E-05</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>21.98645</td>
<td>0</td>
<td>-2.00E-05</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>23.66194</td>
<td>0</td>
<td>-2.00E-05</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>25.52544</td>
<td>0</td>
<td>-2.00E-05</td>
<td>0</td>
</tr>
<tr>
<td>13</td>
<td>27.88093</td>
<td>0</td>
<td>-2.00E-05</td>
<td>0</td>
</tr>
<tr>
<td>14</td>
<td>29.45433</td>
<td>0</td>
<td>-2.00E-05</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>31.41692</td>
<td>0</td>
<td>-2.00E-05</td>
<td>0</td>
</tr>
<tr>
<td>16</td>
<td>33.37942</td>
<td>0</td>
<td>-2.00E-05</td>
<td>0</td>
</tr>
<tr>
<td>17</td>
<td>35.100</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>18</td>
<td>39.200</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>19</td>
<td>43.00</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>20</td>
<td>46.00</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>21</td>
<td>50.00</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>22</td>
<td>60.00</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure AII: CMB Input File, Two Combined Signals

The table above represents the time, \( t \), and the values of \( \alpha \), \( \beta \), \( \dot{\alpha} \), and \( \dot{\beta} \) at each time step. The values are expressed in a format that indicates the change in these parameters over time. The table includes up to 50 rows, with each row representing a time step.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
**Step 3. CFD Output File Modifications**

As discussed in Section 3, the output coordinates of the unsteady airfoil are $C_I$ and $C_m$. Although the CMB code includes several subroutines capable of outputting these parameters, they have proved unreliable because the surface pressure values at nodes located at the intersection of any 2 blocks and the airfoils surface are erroneous. After several attempts at fixing this problem, it was concluded that the error was isolated to those nodes only, and hence, a manual integration of surface pressures ignoring those troublesome nodes proved successful.

**Step 4. Volterra Model Output**

Once the Volterra kernels are identified using the procedure described in Section 2.2.2, $C_I$ and $C_m$ for arbitrary pitch and heave motions and time intervals can be modeled using Equation (3.2); partially reproduced in Equation (A1) below

$$y(t) = \sum_{n=1}^{N} \alpha(n\Delta t) h_i^a(t - n\Delta t) + \sum_{n=1}^{N} \dot{h}(n\Delta t) h_i^b(t - n\Delta t)$$
$$+ \sum_{n_1=1}^{N} \sum_{n_2=1}^{N} \alpha(n_1\Delta t) \alpha(n_2\Delta t) h_2^{aa}(t - n_1\Delta t, t - n_2\Delta t)$$
$$+ \sum_{n_1=1}^{N} \sum_{n_2=1}^{N} \dot{h}(n_1\Delta t) \dot{h}(n_2\Delta t) h_2^{bb}(t - n_1\Delta t, t - n_2\Delta t)$$
$$+ \sum_{n_1=1}^{N} \sum_{n_2=1}^{N} \alpha(n_1\Delta t) \dot{h}(n_2\Delta t) h_2^{ab}(t - n_1\Delta t, t - n_2\Delta t)$$

(A1)