JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 26, No. 2, March–April 2003

Uncertainty Estimation from Volterra Kernels for Robust Flutter Analysis

Richard J. Prazenica,* Rick Lind,† and Andrew J. Kurdila‡
University of Florida, Gainesville, Florida 32611-6250

The flutterometer is a tool used for predicting the onset of flutter during flight testing. This tool uses robust flutter analysis to consider a model with an associated uncertainty description. The flutterometer is particularly useful because the uncertainty description is determined by flight data. However, the standard method of uncertainty estimation is somewhat suspect because of the effects of nonlinearities in the flight data. A method is introduced to estimate uncertainties by considering only the linear component of the flight data. The linear component is extracted by representing the system in terms of Volterra kernels. The first-order kernel describes the linear component of the data and, thus, can be used by the flutterometer. Flight data from the aerostructures test wing is used to demonstrate this procedure. The analysis using the first-order kernel is shown to generate a more accurate description of the modeling error than standard analysis of the measured flight data.

Nomenclature

A	=	area of domain
a	=	scaling function filter
b	=	wavelet filter
D	=	length of Volterra kernel
g	=	function
h	=	Volterra kernel
$\mathbb{K}$	=	set of multi-indices
$[P]$	=	matrix of integral values
T	=	operator
$[T]$	=	matrix form of wavelet transform operator
t	=	time
$[U_1]$	=	matrix of discrete inputs
$[U_2]$	=	matrix of products of discrete inputs
u	=	input
V	=	scaling function approximation space
W	=	wavelet detail space
y	=	response
$\boldsymbol{y}$	=	vector of discrete output values
α	=	coefficient of scaling function
$\boldsymbol{\alpha}$	=	vector of single-scale kernel coefficients
β	=	coefficient of wavelet
$\boldsymbol{\beta}$	=	vector of multiscale kernel coefficients
γ	=	mapping
η, ξ	=	time
ϕ	=	orthonormal scaling function
φ	=	scaling function
χ	=	characteristic function
ψ	=	wavelet
Ω	=	domain of second-order kernel
$\oplus$	=	summation of vector spaces
$\ominus$	=	subtraction of vector spaces

Subscripts and Superscripts

i, P	=	counter
j	=	discretization level
k	=	value in {0, 1, 2, 3}
r	=	value in {1, 2, 3}
κ	=	multi-index

Received 11 March 2002; presented as Paper 2002-1650 at the AIAA/ASME/ASCE/AHS/ASC 43rd Structures, Structural Dynamics, and Materials Conference, Denver, CO, 22–25 April 2002; revision received 22 November 2002; accepted for publication 25 November 2002.

*Graduate Student, Department of Mechanical and Aerospace Engineering; currently National Research Council Fellow, NASA Dryden Flight Research Center, Edwards, CA, 93523; chad.prazenica@dfrc.nasa.gov.

†Assistant Professor, Department of Mechanical and Aerospace Engineering; rick@aero.ufl.edu. Senior Member AIAA.

‡Professor, Department of Mechanical and Aerospace Engineering; ajk@aero.ufl.edu. Associate Fellow AIAA.

Introduction

THE analysis of flight data is obviously important for any flight test. The measurements are usually corrupted by noise and imperfections; however, these data are often the best indicator of the true dynamics of the aircraft. The dependency on data exists for all types of flight testing, but it is especially prevalent when flight flutter testing for envelope expansion.

A tool called the flutterometer has been developed for predicting the onset of flutter during a flight test.[1] This tool is a model-based utility, but it is directly dependent on flight data. The flutterometer computes a flutter speed for an analytical model that is robust with respect to an uncertainty description.[2] The tool uses flight data to generate that uncertainty description. Essentially, the uncertainty is a mathematical operator that describes differences between transfer functions of the model and data. The flutterometer predicts a flutter speed dependent on characteristics of the uncertainty description and consequently dependent on characteristics of the flight data.

A particular concern for testing with the flutterometer is the quality of the uncertainty description. A description that does not consider a sufficient level of modeling error may overpredict the flutter speed. Conversely, a description that considers too much modeling error may underpredict the flutter speed. Either situation is adverse to conducting a safe and efficient flight test.

An accurate assessment of modeling error, using the flutterometer approach, can only result from comparing the transfer function of the model to a transfer function that accurately represents the aircraft dynamics. Such an accurate transfer function is difficult to compute. The flight data used to generate that transfer function often contains components that violate assumptions, such as linearity and statistical properties, associated with standard spectral analysis.

A technique was developed to analyze flight data and assess an accurate measure of modeling error.[3] This technique actually identified model parameters and their associated variances simultaneously. The approach used wavelets for the signal analysis and a min–max optimization for the estimation. This method was shown to generate reasonable results using flight data; however, the results are somewhat limited in that uncertainty is only associated with the observation matrix of the model.

This paper introduces a new technique for estimating uncertainty descriptions. The technique is also a wavelet-based approach, but it

actually relies on the theory associated with Volterra kernels. The concept associated with Volterra kernels is that separate kernels exist that describe dynamics of increasing order.[4] The approach uses wavelets to obtain reduced-order models of these kernels and allows the coefficients describing the dynamics to be computed as a least-squares problem.[5] The theory of Volterra holds for any order of nonlinearity, but this paper will limit consideration to a pair of kernels that describe the linear and quadratic dynamics.

The approach is to identify Volterra kernels to represent the data and then to use those kernels to identify modeling uncertainty. Specifically, levels of uncertainty are formulated by comparing transfer functions of the analytical model and the first-order kernel that represents the linear component of the data. The use of Volterra modeling is essential to this process because it separates the linear and nonlinear parts of the data into separate kernels. The objective is to analyze a linear model and determine linear stability properties; therefore, the optimal data set for analysis is actually the linear part of the measured data.

The uncertainty determined by comparing the model to the first-order kernel should be more accurate than uncertainties determined by directly considering the measured data. This does not necessarily mean that the uncertainty will be less; rather, it simply means the uncertainty will be a better indication of modeling errors. The nonlinearities in the measured data may exaggerate or hide the true errors in the linear model when comparing transfer functions from that data and model. The first-order kernel will give a more realistic indication of the linear dynamics of the aircraft by removing the effects of these nonlinearities.

System Identification for Volterra Kernels

Volterra Series Representations

The use of Volterra kernels for modeling system dynamics is considered in this paper. Such representation will be built on assumptions that the system under consideration is causal and time invariant. The system is assumed to have finite memory so that the response $y(t)$ at any time is caused by the excitations $u(t)$ from a finite length of time previously. Also, the system will be assumed to be weakly nonlinear in the sense that the dynamics are dominated by first-order and second-order terms in the Volterra series.

The Volterra theory states that a nonlinear system can be expressed in terms of an infinite sum of integral operators of increasing order. Correspondingly, the response $y(t)$ of that system can be expressed as a sum of components $y_i(t)$ that are responses from each operator:

$$y(t) = y_1(t) + y_2(t) + \cdots + y_\infty(t) \tag{1}$$

Here, $y_1(t)$ is the response of the first-order operator, and $y_2(t)$ is the response of the second-order operator. These components can be formally expressed in terms of the first-order and second-order kernels, h_1 and h_2, and the input $u(t)$ that generated the response[4,6]:

$$y_1(t) = \int_0^t h_1(\xi)u(t-\xi)\,\mathrm{d}\xi \tag{2}$$

$$y_2(t) = \int_0^t \int_0^\eta h_2(\xi,\eta)u(t-\xi)u(t-\eta)\,\mathrm{d}\xi\,\mathrm{d}\eta \tag{3}$$

The kernels defined in Eqs. (2) and (3) are functions of ξ and η parameters. The domains of these parameters can be quite large and so it is desired to obtain reduced-order representations of the kernels. Several approaches have been developed for obtaining these representations using different techniques; however, this paper will adopt an approach based on wavelets.[5]

The definition of $y_2(t)$ clearly indicates that the second-order Volterra operator is dependent on the kernel h_2 over the triangular domain defined by $0 \le \xi \le \eta$ and $0 \le \eta \le t$. This kernel is actually a decaying function because of the assumption of finite memory. Thus, the kernel is effectively zero after some period of time D and is supported over the triangular domain Ω, as shown in Fig. 1.

A wavelet basis is constructed over the domain of Ω for the representation of the triangular form of the second-order Volterra kernel.

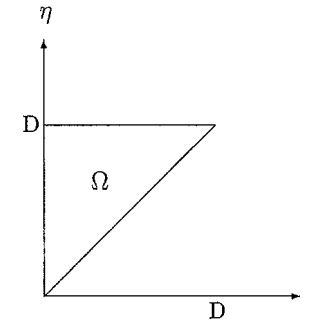

Fig. 1 Triangular domain Ω over which the second-order Volterra kernel is defined.

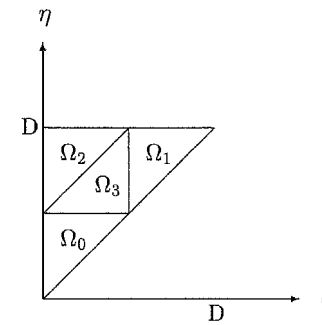

Fig. 2 Subsets $\Omega_i \subset \Omega$.

The wavelet coefficients in this expansion are then determined by analyzing flight data. Of course, the basis could actually be constructed over the square domain bounded by $0 \le \xi \le D$ and $0 \le \eta \le D$, corresponding to the symmetric form of the second-order kernel, but the triangular approach is more computationally efficient.

Wavelets Supported on Triangular Domains

The construction of wavelets on invariant sets has been discussed in detail in the literature.[7–9] This construction is used to derive a wavelet-based multiresolution analysis over the domain of Ω. The derivation of this analysis requires four contractive and affine mappings, $\gamma_i : \mathbb{R}^2 \to \mathbb{R}^2$, to be defined:

$$\gamma_0(\xi,\eta) = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix} \begin{Bmatrix} \xi \\ \eta \end{Bmatrix}$$

$$\gamma_1(\xi,\eta) = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix} \begin{Bmatrix} \xi \\ \eta \end{Bmatrix} + \begin{Bmatrix} D/2 \\ D/2 \end{Bmatrix}$$

$$\gamma_2(\xi,\eta) = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix} \begin{Bmatrix} \xi \\ \eta \end{Bmatrix} + \begin{Bmatrix} 0 \\ D/2 \end{Bmatrix}$$

$$\gamma_3(\xi,\eta) = \begin{bmatrix} -\frac{1}{2} & 0 \\ 0 & -\frac{1}{2} \end{bmatrix} \begin{Bmatrix} \xi \\ \eta \end{Bmatrix} + \begin{Bmatrix} D/2 \\ D \end{Bmatrix} \tag{4}$$

These invertible operators effectively map the domain of Ω into four subdomains given as $\Omega_i := \gamma_i(\Omega)$. These subdomains are shown in relation to Ω in Fig. 2.

Define χ_Ω to be the characteristic function over the domain of Ω:

$$\chi_\Omega(\xi,\eta) := \begin{cases} 1, & (\xi,\eta) \in \Omega \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

An orthonormal scaling function ϕ is then defined using this characteristic function and the area A_0 of the domain Ω:

$$\phi(\xi,\eta) := \left(1/\sqrt{A_0}\right)\chi_\Omega(\xi,\eta) \tag{6}$$

Clearly, ϕ is a constant function over Ω. Consider $\mathcal{L}_2(\Omega)$ as the space of square-integrable functions in Ω. The one-dimensional space spanned by ϕ can be defined as V_0 such that $V_0 \subset \mathcal{L}_2(\Omega)$.

The scaling functions that span a finer-resolution approximation space V_1 are defined as normalized characteristic functions over the subdomains Ω_i:

$$\phi_{1,i}(\xi,\eta) = \left(2\Big/\sqrt{A_0}\right)\chi_{\Omega_i}(\xi,\eta), \qquad i = 0, 1, 2, 3 \tag{7}$$

Subscript 1 denotes the resolution level of the scaling function, whereas index i specifies the subdomain Ω_i over which the function is supported. Define the operators

$$(T_i g)(\xi,\eta) = g\left[\gamma_i^{-1}(\xi,\eta)\right]\chi_{\Omega_i}(\xi,\eta), \qquad i = 0, 1, 2, 3 \tag{8}$$

where g is any function in $\mathcal{L}_2(\Omega)$. These operators simply restrict functions to the subdomains Ω_i in the same manner that the mappings γ_i restrict points in Ω to Ω_i. The scaling functions that span V_1 can then be expressed in terms of the operators T_i acting on the scaling function ϕ:

$$\phi_{1,i}(\xi,\eta) = 2(T_i\phi)(\xi,\eta), \qquad i = 0, 1, 2, 3 \tag{9}$$

The space V_1, defined as

$$V_1 := \text{span}\{\phi_{1,0}, \phi_{1,1}, \phi_{1,2}, \phi_{1,3}\} \tag{10}$$

can then be written in terms of the operators T_i acting on the coarser-resolution space V_0:

$$V_1 = T_0V_0 \oplus T_1V_0 \oplus T_2V_0 \oplus T_3V_0 \tag{11}$$

A series of nested spaces, $V_{j+1} \supset V_j$, of increasing resolution can be constructed recursively via the relationship

$$V_{j+1} = T_0V_j \oplus T_1V_j \oplus T_2V_j \oplus T_3V_j \tag{12}$$

A set of multi-index parameters $\mathbb{K}_j$ is defined for each level of resolution. A multi-index, $\kappa \in \mathbb{K}_j$, is basically a group of indices. Each group has j elements for the set defined on the resolution level of j such that $\kappa = \{k_1, k_2, \ldots, k_j\}$. Furthermore, each index in this group can take on values of 0, 1, 2, or 3. This set can be expressed mathematically, but it is essential to note that $\mathbb{K}_j$ is only defined in association with an implied j that represents the level of resolution:

$$\mathbb{K}_j = \{\{k_1, k_2, \ldots, k_j\} : k_i \in \{0, 1, 2, 3\}, \qquad i = 1, \ldots, j\} \tag{13}$$

The set of multi-index parameters grows more complex as the level of resolution is increased. The increase in complexity results from the corresponding increase in the number of indices contained in each multi-index in the set. For example, each multi-index has a single value for level 1 resolution so that $\mathbb{K}_1$ is the set of $\{0, 1, 2, 3\}$ and has four elements. Similarly, each multi-index is a group of 2 indices for level 2 resolution so that $\mathbb{K}_2$ is the set of $\{(0, 0), (0, 1), \ldots, (3, 3)\}$ and has 16 elements.

The concept of a multi-index simplifies the presentation of the multiresolution analysis. In particular, the operators and vectors associated with spaces of increasing resolution can be presented in terms of a $\mathbb{K}_j$.

Scaling functions are defined for each level of resolution. Logically, the number of these functions increases as the resolution is increased. Thus, the level 1 resolution has 4 scaling functions and level 2 resolution has 16 scaling functions. The scaling functions for any level are computed by applying a sequence of operators to the scaling function ϕ defined at level 0 resolution. The sequence of operators corresponds to a multi-index. Thus, each scaling function can be written as $\phi_{j,\kappa}$:

$$\phi_{j,\kappa}(\xi,\eta) := 2^j\left(T_{k_j}\cdots T_{k_1}\phi\right)(\xi,\eta) \tag{14}$$

where $\kappa = \{k_1, k_2, \ldots, k_j\} \in \mathbb{K}_j$.

Each space V_j is easy to define using the multi-index set,

$$V_j := \text{span}\{\phi_{j,\kappa} : \kappa \in \mathbb{K}\} \tag{15}$$

The multiresolution analysis can be completed by deriving the wavelets that span the orthogonal complement spaces W_j defined as the differences between adjacent approximation spaces V_j and V_{j+1}:

$$W_j := V_{j+1} \ominus V_j \tag{16}$$

Because V_0 is a one-dimensional space and V_1 is a four-dimensional space, W_0 is a three-dimensional subspace of V_1. The following orthonormal wavelets form a basis for W_0:

$$\begin{aligned}
\psi^1(\xi,\eta) &:= \left(1\Big/\sqrt{2}\right)\phi_{1,0}(\xi,\eta) - \left(1\Big/\sqrt{2}\right)\phi_{1,1}(\xi,\eta)\\
\psi^2(\xi,\eta) &:= \left(1\Big/\sqrt{2}\right)\phi_{1,2}(\xi,\eta) - \left(1\Big/\sqrt{2}\right)\phi_{1,3}(\xi,\eta)\\
\psi^3(\xi,\eta) &:= \tfrac{1}{2}\phi_{1,0}(\xi,\eta) + \tfrac{1}{2}\phi_{1,1}(\xi,\eta)\\
&\quad - \tfrac{1}{2}\phi_{1,2}(\xi,\eta) - \tfrac{1}{2}\phi_{1,3}(\xi,\eta)
\end{aligned} \tag{17}$$

Each wavelet is a piecewise-constant function over Ω. By inspection, one can verify that these wavelets are orthogonal to each other and to the scaling function ϕ that spans V_0. Define the space

$$W_0 := \text{span}\{\psi^1, \psi^2, \psi^3\} \tag{18}$$

The wavelet spaces W_j are constructed recursively using the operators T_i:

$$W_{j+1} = T_0W_j \oplus T_1W_j \oplus T_2W_j \oplus T_3W_j \tag{19}$$

Each space W_j is defined as

$$W_j := \text{span}\left\{\psi^r_{j,\kappa} : \kappa \in \mathbb{K}_j, r = 1, 2, 3\right\} \tag{20}$$

where each wavelet $\psi^r_{j,\kappa}$ is given by

$$\psi^r_{j,\kappa}(\xi,\eta) := 2^j\left(T_{k_j}T_{k_{j-1}}\cdots T_{k_1}\psi^r\right)(\xi,\eta) \tag{21}$$

A multiresolution analysis is now formulated for $\mathcal{L}_2(\Omega)$. Recall that the space V_j can be expressed as the sum of the two coarser-resolution spaces V_{j-1} and W_{j-1}:

$$V_j = V_{j-1} \oplus W_{j-1} \tag{22}$$

A recursive application of this relationship admits the following multiscale decomposition:

$$V_j = V_0 \oplus W_0 \oplus W_1 \oplus \cdots \oplus W_{j-2} \oplus W_{j-1} \tag{23}$$

A single-scale representation of a function $f \in \Omega$ in terms of V_j can be written as

$$f_j(\xi,\eta) = \sum_{\kappa\in\mathbb{K}_j}\alpha_{j,\kappa}\phi_{j,\kappa}(\xi,\eta) \tag{24}$$

where the $\{\alpha_{j,\kappa}\}$ are constant scaling function coefficients. From Eq. (23), an equivalent multilevel expansion of f is given by

$$f_j(\xi,\eta) = \alpha\phi(\xi,\eta) + \sum_{r=1}^{3}\sum_{i=0}^{j-1}\sum_{\kappa\in\mathbb{K}_i}\beta^r_{i,\kappa}\psi^r_{i,\kappa}(\xi,\eta) \tag{25}$$

where the $\{\beta^r_{i,\kappa}\}$ are constant wavelet coefficients. Fast algorithms can be derived by which the multiscale coefficients in Eq. (25) are calculated from the single-scale coefficients in Eq. (24). First, note that the scaling functions and wavelets satisfy the following two-scale relationships:

$$\phi(\xi,\eta) = \sum_{i=0}^{3}a_i\phi_{1,i}(\xi,\eta) \tag{26}$$

$$\psi^r(\xi,\eta) = \sum_{i=0}^{3}b^r_i\phi_{1,i}(\xi,\eta), \qquad r = 1, 2, 3 \tag{27}$$

where the scaling function filters $\{a_i\}$ and the wavelet filters $\{b_i^r\}$ for $i \in \{0, 1, 2, 3\}$ are defined as

$$\{a_0, a_1, a_2, a_3\} = \left\{\tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2}\right\}$$

$$\left\{b_0^1, b_1^1, b_2^1, b_3^1\right\} = \left\{1/\sqrt{2}, -1/\sqrt{2}, 0, 0\right\}$$

$$\left\{b_0^2, b_1^2, b_2^2, b_3^2\right\} = \left\{0, 0, 1/\sqrt{2}, -1/\sqrt{2}\right\}$$

$$\left\{b_0^3, b_1^3, b_2^3, b_3^3\right\} = \left\{\tfrac{1}{2}, \tfrac{1}{2}, -\tfrac{1}{2}, -\tfrac{1}{2}\right\} \tag{28}$$

The two-scale relationships and the orthogonality properties of the functions enable the derivation of decomposition formulas of the form

$$\alpha_{j-1,\kappa} = \sum_{i=0}^{3} a_i \alpha_{j,i|\kappa} \tag{29}$$

$$\beta_{j-1,\kappa}^r = \sum_{i=0}^{3} b_i^r \alpha_{j,i|\kappa}, \qquad r = 1, 2, 3 \tag{30}$$

where $i|\kappa$ denotes appending i to the multi-index κ. Similarly, obtain the reconstruction formula,

$$\alpha_{j,m|\kappa} = a_m \alpha_{j-1,\kappa} + \sum_{r=1}^{3} b_m^r \beta_{j-1,\kappa}^r \tag{31}$$

Equations (29) and (30) can be applied recursively to compute the multilevel coefficients in Eq. (25) from the single-scale coefficients in Eq. (24). The reverse can be accomplished through a recursive application of Eq. (31).

Volterra Kernel Identification

The identification of the Volterra kernels is obviously of primary importance to their use for signal analysis. Identification should address both the accuracy of the resulting model and the computational efficiency of the process. Thus, this topic has received much attention in the literature using approaches ranging from statistical arguments to genetic algorithms. A least-squares approach is utilized here that is formulated to be especially efficient for identifying the Volterra kernels defined in this paper.

The input and output measurements need to be discretized for the analysis. This step is performed by using a standard zero-order hold. The discretization step is 2^{-j}, and N is the number of discrete input and output values in the data set. The zero-order hold of the input takes the form

$$u_j(t) = \sum_{i=0}^{N-1} u_{j,i} \chi_{j,i}(t) \tag{32}$$

where $\chi_{j,i}$ is the characteristic function over the interval $[2^{-j}i, 2^{-j}(i+1)]$. The first-order Volterra kernel can be approximated in terms of scaling functions on level j using any arbitrary one-dimensional scaling function φ as

$$h_{1,j}(\xi) = \sum_{i=0}^{N-1} \alpha_{j,i} \varphi_{j,i}(\xi) \tag{33}$$

This kernel $h_{1,j}$ can be formulated equivalently as a multiscale representation:

$$h_{1,j}(\xi) = \sum_{i} \alpha_{j_0,i} \varphi_{j_0,i}(\xi) + \sum_{p=j_0}^{j-1} \sum_{i} \beta_{p,i} \psi_{p,i}(\xi) \tag{34}$$

where j_0 is the coarsest level used in the approximation. If the scaling function and wavelet pair φ and ψ are chosen such that the scaling functions $\varphi_{j,k}$ are orthogonal to the characteristic functions $\chi_{j,k}$, the following expression is obtained for the output of the first-order Volterra operator[10]:

$$\boldsymbol{y}_{1,j} = [U_1]\boldsymbol{\alpha}_j = [U_1][T_1]^{-1}\boldsymbol{\beta}_1 \tag{35}$$

In Eq. (35), $\boldsymbol{y}_{1,j}$ is a vector of discrete output values, $[U_1]$ is a matrix of discrete inputs, and $\boldsymbol{\alpha}_j$ is a vector composed of the kernel coefficients from Eq. (33). $[T_1]$ is the invertible matrix operator that decomposes the single-scale coefficients in Eq. (33) into the multiscale coefficients in Eq. (34). The vector $\boldsymbol{\beta}_1$ is the vector of multiscale kernel coefficients.

The second-order kernel can be approximated in terms of the wavelet basis derived in the preceding section. The kernel is expressed as a single-scale representation:

$$h_{2,j_2}(\xi, \eta) = \sum_{k \in \mathbb{K}_{j_2}} \alpha_{j_2,\kappa} \phi_{j_2,\kappa}(\xi, \eta) \tag{36}$$

and a multilevel expansion of the form

$$h_{2,j_2}(\xi, \eta) = \alpha\phi(\xi, \eta) + \sum_{r=1}^{3} \sum_{p=0}^{j_2-1} \sum_{\kappa \in \mathbb{K}_p} \beta_{p,\kappa}^r \psi_{p,\kappa}^r(\xi, \eta) \tag{37}$$

Note that the discretization level j_2 for the kernel is not necessarily chosen to be the same as that for the input and output. Denote $[T_2]$ to be the matrix operator that transforms the vector of single-scale coefficients into the vector of multiscale coefficients. When the zero-order hold approximation of the input and the expression for the second-order kernel in Eq. (36) are used, the following expression is obtained for the discrete outputs of the second-order Volterra operator:

$$y_{2,j}(2^{-j}n) = \sum_{m,p,\kappa} \left(u_{j,n-m-1} u_{j,n-p-1} \alpha_{j_2,\kappa}\right) \times \left(\int_{\Omega} \phi_{j_2,\kappa}(\xi, \eta) \chi_{j,m}(\xi) \chi_{j,p}(\eta)\, \mathrm{d}\Omega\right) \tag{38}$$

for $n = 1, \ldots, N$. This can be written in matrix form as

$$\boldsymbol{y}_{2,j} = [U_2][P]\boldsymbol{\alpha}_{j_2} = [U_2][P][T_2]^{-1}\boldsymbol{\beta}_2 \tag{39}$$

where $[U_2]$ is a matrix of products of discrete input values and $[P]$ is a matrix of values of the integrals in Eq. (38). The vector $\boldsymbol{\alpha}_{j_2}$ is composed of the single-scale coefficients from Eq. (36), and $\boldsymbol{\beta}_2$ is a vector of multiscale coefficients from Eq. (37).

In this paper, the Volterra series representation of the system is truncated to include only the first- and second-order operators. Then, the vector of discrete outputs, $\boldsymbol{y}_j$ can be written as

$$\boldsymbol{y}_j = \boldsymbol{y}_{1,j} + \boldsymbol{y}_{2,j} \tag{40}$$

When Eqs. (35) and (39) are combined, the total model takes the form

$$\boldsymbol{y}_j = \left[[U_1][T_1]^{-1} \quad [U_2][P][T_2]^{-1}\right] \begin{Bmatrix} \boldsymbol{\beta}_1 \\ \boldsymbol{\beta}_2 \end{Bmatrix} \tag{41}$$

The kernel identification problem is reduced to solving, in a least-squares sense, for the first- and second-order kernel coefficients in Eq. (41). The wavelet representations of the kernels admit reduced-order representations of the kernels through the truncation of the multiscale vectors $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$. In this manner, the goal is to obtain accurate models for the first- and second-order kernels in terms of a relatively small number of wavelet coefficients.

Robust Flutter Analysis

Uncertainty Estimation

Robust flutter analysis considers an analytical model with an associated uncertainty description. The model is a linear model that relates the coupled structural dynamics and aerodynamics as represented by rational function approximations. The uncertainty description indicates levels of possible errors and variations in the different parameters of that model.

The standard approach to estimate uncertainty is to compare transfer functions of the model and data. The effect of the uncertainty is that the model could actually produce a range of transfer functions. The levels of uncertainty are determined by increasing the size of the parameters until the resulting range of transfer functions for the

model completely bounds any transfer functions from data. In effect, this is a model validation condition that determines whether that model could reproduce the flight data.

The new approach to estimate uncertainty is to compare transfer functions of the model and the first-order Volterra kernels. The data are actually respresented by a set of Volterra kernels; however, the linear dynamics are purely described by the first-order kernel. The procedure for estimating uncertainty is to compare transfer functions of the model and the first-order kernel. The standard model validation condition is then applied such that the uncertainty levels are chosen to ensure the range of the uncertain model bounds the transfer function of the first-order Volterra kernel.

The estimation of uncertainty is intended to describe linear errors in a linear model and, consequently, be used for analysis of linear stability; therefore, the uncertainty estimation should only consider the first-order Volterra kernel. Any differences between the linear model and the second-order Volterra kernel are not of interest for flutter analysis. The second-order kernel is of obvious interest for limit- cycle analysis, but it is optimal to consider only the first-order kernel for purposes of flutter analysis.

Predicting the Onset of Flutter

The prediction of flutter for the flutterometer is based on μ-method analysis.[2] This type of analysis computes a stability measure that is robust with respect to an uncertainty description. Thus, the flutter speed is computed as the largest increase in airspeed for which the theoretical model remains robustly stable with respect to the uncertainty.

The flutterometer operates by computing a robust flutter speed at every test point during an envelope expansion. The initial step is to compute an uncertainty description for the model at that flight condition. The next step is to compute the robust flutter speed by applying μ-method analysis to the uncertain model. In this way, the flutterometer predicts a realistic flutter speed that is more beneficial than theoretical predictions because the robust speed directly accounts for flight data.

Note the actual operation of the flutterometer is not altered by this paper; instead, the data supplied to the flutterometer are altered. The original approach was to analyze the measured data using the flutterometer. The new approach is to first identify Volterra kernels from the data and then analyze the response data from the first-order kernel using the flutterometer.

Flight-Test Example

Aerostructures Test Wing

The aerostructures test wing (ATW) was developed at NASA Dryden Flight Research Center. This testbed was specifically designed for testing methods to analyze aeroelastic stability and to predict the onset of flutter. The ATW was essentially a flexible wing and boom assembly, as shown in Fig. 3.

The wing was formulated based on a NACA-65A004 airfoil shape. The wing had a span of 18.0 in. with root chord length of 13.2 in. and tip chord length of 8.7 in. The boom was a 1-in.-diam hollow tube of length 21.5 in. The total weight of the ATW was 2.66 lb.

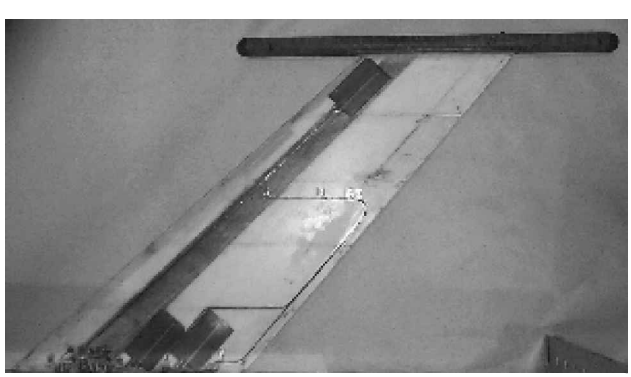

Fig. 3 ATW.

This assembly was flown by using an F-15 aircraft and associated flight-test fixture. The ATW was mounted horizontally to the fixture and the resulting system attached to the undercarriage of the F-15 fuselage.[11] Previous testing indicated that the airflow is relatively smooth around the system and so the F-15 fuselage and wings are assumed to have minimal interference with the ATW.

A measurement and excitation system was incorporated into the wing. The measurement system consisted of 18 strain gauges placed throughout the airfoil structure and 3 accelerometers placed at fore, aft, and midlocations in the boom. The excitation system was six patches of piezoelectric material, three patches mounted on the upper surface that are out of phase with three patches on the lower surface, that acted as a single distributed actuator. Sinusoidal sweeps of energy from 5 to 35 Hz were commanded to these patches.

Ground-vibration tests were conducted to determine the structural dynamics of the wing. The main modes of the system were the first-bending mode at 13.76 Hz and the first-torsion mode at 20.76 Hz. Tests were conducted for the wing on a test stand and also attached to the flight-test fixture to ensure that these modal properties are not affected for the flight testing.

Flight Data

The ATW was flown at several different flight conditions. Measurements were recorded at each of these points. The measurements used for flutter analysis were the voltage commanded to the piezoelectric excitation systems and the gravitational acceleration levels from the accelerometers in the boom. In each case, the commanded signal was a sine-sweep chirp from 5 to 35 Hz for 60 s.

The flight data were measured at 800 Hz and so indicate energy at frequencies considerably greater than the modes of interest for flutter predictions. Thus, the input and output measurements were filtered to a sampling rate of 256 Hz. This filtering was performed using the Haar wavelet transform. This paper will only consider data from a single test point. That test point occurred at flight conditions of Mach 0.80 and altitude of 20,000 ft. The accelerometer measurement at the trailing edge of the boom in response to the chirp excitation is shown in Fig. 4.

Volterra Kernels

Volterra kernels were computed from the data shown in Fig. 4. The order of the kernels included in the Volterra model generally depends on the degree of nonlinearity that is to be identified from the data. Certain nonlinear systems have been shown to require kernels of third order and higher[12]; however, the ATW was essentially a linear system. Thus, only first-order and second-order kernels were identified to reflect properties associated with linear and quadratic dynamics.

The number of points in each kernel representation affects the time-domain and frequency-domain resolution of the kernel. Each kernel was chosen to have a resolution of 256 points/s to match the sampling rate of the data.

Another parameter that needed to be chosen was the time duration of the kernels. This time duration represents the memory length of

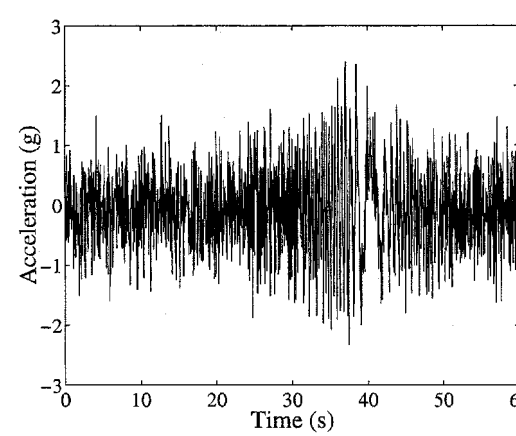

Fig. 4 Response of the trailing-edge boom accelerometer to a chirp excitation.

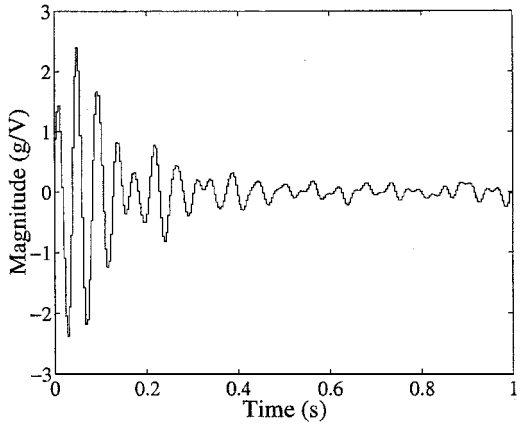

Fig. 5 Identified first-order Volterra kernel.

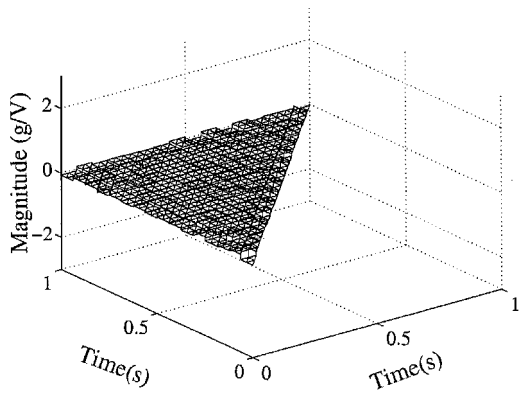

Fig. 6 Identified second-order Volterra kernel.

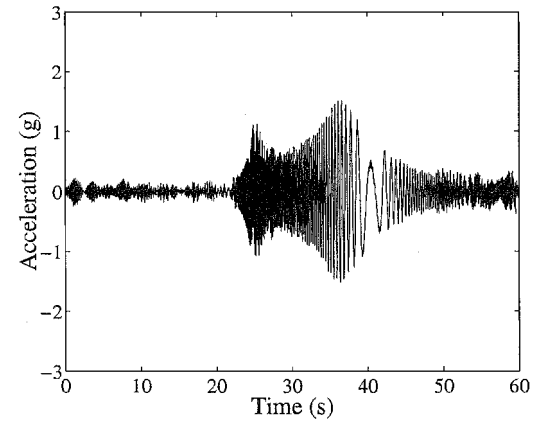

Fig. 7 Simulated response to chirp input from first-order Volterra kernel.

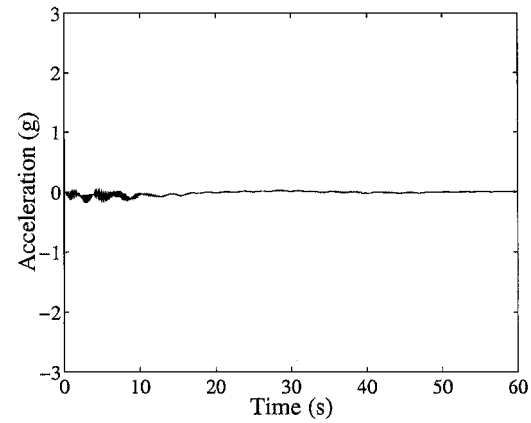

Fig. 8 Simulated response to chirp input from second order.

the system. Alternatively, the time duration can be interpreted as the length of time into the future that a current input will still have an influence on the output. Most physical systems effectively have finite memory so the time duration of the kernel should also be finite. The effect of finite memory is that the Volterra kernels should decay to zero in a finite period of time.

Volterra kernels were identified from the flight data using time durations of 1, 2, and 4 s. The results consistently indicated that the first-order kernel almost completely decays after 1 s. Thus, the data analysis proceeded using only kernels with this length. Figure 5 shows the first-order kernel that was identified from the data. This kernel is represented in terms of 256 wavelet coefficients and has a time duration of 1 s. The triangular form of the identified second-order kernel is shown in Fig. 6. This kernel also has a time duration of 1 s and was identified using 256 nonzero terms.

An obvious feature in Figs. 5 and 6 is that the first-order kernel is an order of magnitude larger than the second-order kernel. This difference in size indicates the flight data was essentially generated by a linear system. These results agree with previous analysis that noted the ATW responded linearly to a range of inputs.[13] Thus, the first-order kernel dominates the identified dynamics and demonstrates the ATW was primarily a linear system.

Another feature of Fig. 5 is that the first-order kernel resembles an impulse response. The first-order kernel is theoretically predicted to be an impulse response when analyzing data from a linear system. Therefore, this feature also demonstrates the ATW was primarily a linear system.

The linear component of the flight data can be estimated by simulating the response of the first-order Volterra kernel. Such a response is generated by applying the chirp signal commanded to the ATW during flight. The resulting response is shown in Fig. 7.

The response of the first-order kernel is different from the measured flight data shown in Fig. 4; however, the response and data do have some similarities. Notably, the response and the measurements indicate similar frequency-response characteristics. Both Figs. 4 and 7 show increased magnitude near 25 s, at which time the excitation signal is about 18 Hz. Similarly, both show increased magnitude near 37 s at which time the excitation signal is about 24 Hz. This indicates the systems that generated the simulated response and measured data both have modes near 18 and 24 Hz.

The simulated response and measured data are notably different at times when the excitation is not exciting the aeroelastic modes. This is particularly evident at the beginning of the excitation. The measured data contains a large amount of noise but only a small amount is reproduced by the simulated response. The identification procedure noted this noise did not correspond to the response of a linear system and so the first-order kernel does not significantly represent this component of the data.

The response of the second-order kernel can also be computed. This response, as shown in Fig. 8, indicates the nonlinear component of the flight data. The magnitude of this data is quite small and corresponds to the small magnitude of the kernel shown in Fig. 6. This response is actually assumed to represent nonlinear characteristics of the noise rather than nonlinear characteristics of the ATW. For instance, note the magnitude is largest near the beginning of the response that corresponds to excitation far from the modal frequencies.

Finally, a frequency-domain representation of the first-order Volterra kernel is shown in Fig. 9. This representation clearly shows that the identified dynamics are dominated by modes near 18 and 24 Hz.

Uncertain Model

An analytical model of the ATW was developed that combined a finite element model with data from the ground-vibration testing. A finite element model was initially used to generate a set of mass values at locations throughout the structure. Correspondingly, the test data indicated the frequencies and responses at these locations for

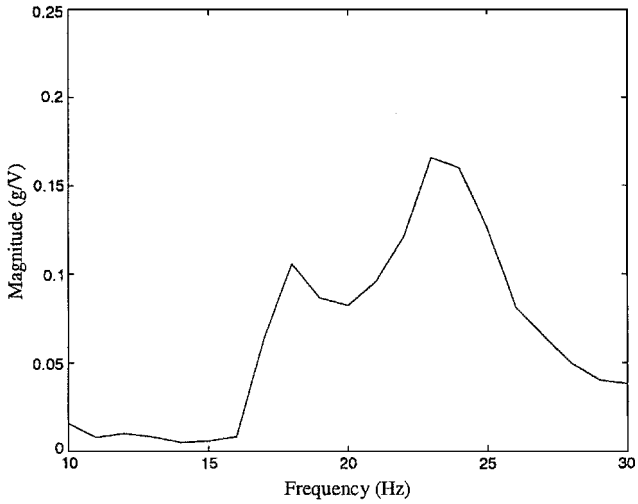

Fig. 9 Fast Fourier transform of first-order Volterra kernel.

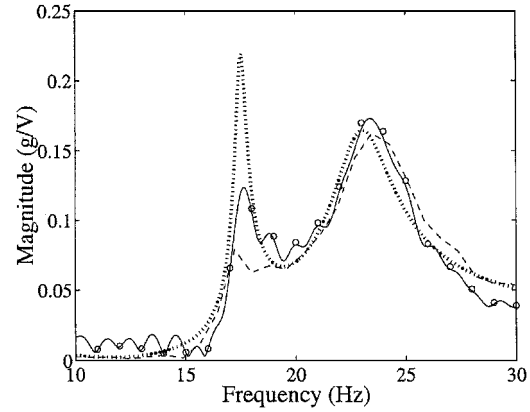

Fig. 10 Transfer functions: ····, nominal model; – – –, flight data; ——, response of first-order Volterra kernel; and ○, exact first-order Volterra kernel.

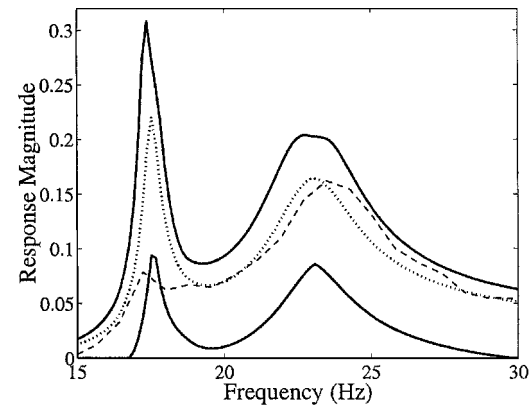

Fig. 11 Transfer functions: – – –, flight data; ····, nominal model; and ——, upper and lower bounds of uncertain model.

modes of the structure. An equivalent model was then formulated with natural frequencies and mode shapes that were determined by the data, mass values that were purely analytical, and stiffness values that resulted from relating the analytical mass and experimental natural frequencies. Thus, this equivalent model was representative of both analytical and experimental results. This model was formulated using the ZAERO package.[14]

A state-space model of the ATW is generated by the equivalent model formulated using ZAERO. This model includes the structural dynamics and the associated aerodynamic forces caused by aeroelastic coupling. The structural dynamics are realized as standard mass, stiffness, and damping matrices. The aerodynamic forces are realized as a rational function approximation.[15] The input to this model is the voltage to the excitation system, and the output from this model is the acceleration at the trailing edge of the boom.

The introduction of uncertainties actually made use of both the structural and aerodynamic representations. Parametric uncertainty was introduced to admit variations directly in the stiffness and damping matrices of the structural dynamics. Dynamic uncertainty was introduced to admit variations in magnitude and phase of the aerodynamic forces. Also, dynamic uncertainty was associated with the excitation and sensing signals to account for the effects of unmodeled dynamics and mode shape errors.

Uncertainty Estimation

The magnitudes of uncertainties in the model need to be determined to relate size of errors. These errors are computed by applying the model validation procedure that compares transfer functions from a model and data. The data that are considered are the accelerometer data measured during flight and the simulated data produced by the Volterra representation of the data.

The aeroelastic instability to be predicted for the ATW is assumed to be a flutter mechanism caused by linear dynamics. Thus, the uncertainty estimation only considers the first-order Volterra kernel. This kernel will indicate the error in the linear model for describing the linear dynamics of the ATW.

The transfer function describing the first-order Volterra kernel needs to be determined. This transfer function is actually shown as the frequency-domain kernel shown in Fig. 9. That kernel is an exact representation of the transfer function; however, it is not suitable for uncertainty estimation. The kernel was formulated using only 256 points and so the resolution in the frequency domain is somewhat coarse.

The first-order Volterra kernel is used to simulate response in Fig. 7, which shows the linear component of the flight data. A transfer function is then determined from these high-rate simulated data. The resulting transfer function is shown in Fig. 10 along with the transfer functions from the model and the measured flight data.

The transfer functions in Fig. 10 immediately demonstrate information about the quality of the model. The initial analysis of the flight data noted much less response of the bending mode near 18 Hz than was predicted by the model. The analysis using the Volterra kernel actually indicates the differences between the model and data for this mode are not as great as initially indicated. A visual inspection shows the model predicts a response magnitude of about 0.23 g/V, whereas the original data indicates a magnitude of about 0.08 g/V and the first-order Volterra kernel indicates a magnitude of about 0.13 g/V.

Also, the values of the coarse-resolution exact representation of the kernel are shown in comparison with the transfer function computed from the simulated data. Clearly the approach using simulated data results in a transfer function that matches the exact representation at the coarse-resolution frequencies but also provides more resolution at other frequencies.

The actual estimation of the uncertainty associated with the model uses the mathematical procedure for model validation. This procedure increases the amount of uncertainty associated with the model until that model does not invalidate the data. Essentially, the concept is to increase the uncertainty until the flight data lie within upper and lower bounds resulting from analysis of the uncertain model. The uncertainty associated with the model is estimated using the accelerometer data measured during flight. The resulting uncertainty levels produce upper and lower bounds on the range of transfer functions from the model as shown in Fig. 11.

One feature of note in Fig. 11 is that the flight data lie outside the upper and lower bounds of the model near 18 Hz. This feature would seem to indicate that the data are invalidated by the model so that more uncertainty is needed. Actually, this feature is a result of the resolution of the frequency-domain data. There is no data point right at this frequency and so the appearance of invalidation is an artifact of connecting discrete points with a continuous line.

An uncertainty description associated with the model is also computed by analyzing the data simulated from the first-order Volterra

Table 1 Magnitudes of uncertainty affecting the model

Uncertainty	Original data	First-order kernel
Structural stiffness	28	18
Structural damping	51	30
Aerodynamic forces	1	1
Command input	11	8
Measured output	11	8

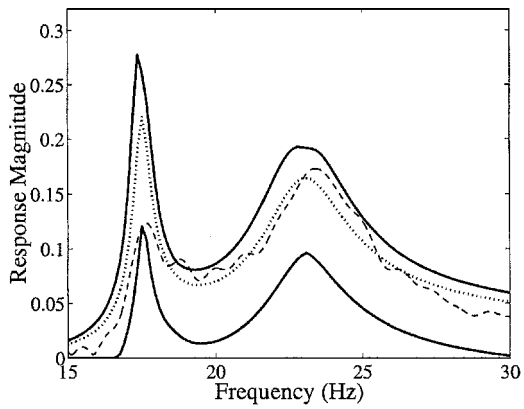

Fig. 12 Transfer functions: – – –, first-order Volterra kernel; · · · ·, nominal model; and ——, upper and lower bounds of uncertain model.

kernel. This description includes uncertainty levels that produced upper and lower bounds on the range of transfer functions from the model as shown in Fig. 12. The upper and lower bounds presented in Figs. 11 and 12 demonstrate the advantage of using the Volterra representation of the data. Namely, the uncertainty description associated with linear errors in the model is more accurate when using the first-order Volterra kernel for model validation as compared to using the measured data. Thus, the linear model more closely resembles the response of the linear kernel as compared to the noisy flight data.

The actual percentages resulting from model validation with each type of transfer function are shown in Table 1 for the uncertainties affecting different parameters of the model. Note that the uncertainties are associated with the parameter in a multiplicative fashion so that, for example, the stiffness in the model needs to vary by $+/-$ 28% when comparing with the original data but only needs to vary by $+/-$ 18% when comparing with the response of the first-order kernel. Clearly the model needs less uncertainty to ensure the transfer function from the first-order Volterra kernel is not invalidated as compared to the transfer function from the measured data. This reduction in uncertainty directly results from the visual features noted in Figs. 11 and 12.

Flutter Prediction

Flutter speeds are computed for the model to predict the flight conditions associated with the onset of the instability. These values correspond to the airspeed associated with flight at Mach 0.80 for the ATW. The flutter speeds may be easily converted to flutter altitudes using match-point relationships, but it is sufficient to discuss speeds for the purpose of this paper.

Several values of flutter speeds are computed. A speed is computed by analysis of the nominal model with no consideration of uncertainty. A speed is computed by analysis of the model that indicates a prediction of flutter that is robust with respect to the uncertainty set that is estimated from the flight data. Similarly, a speed is computed by analysis of the model that indicates a prediction of flutter that is robust with respect to the uncertainty set that is estimated from the response of the first-order Volterra kernel. Also, the actual flutter speed of the ATW is computed as the speed at which the test article experienced flutter during flight testing.[16] These speeds are presented in Table 2. The main feature of Table 2 is that the flutter speed is closer to the true value and, therefore, less conservative, when computed with respect to uncertainty determined by the Volterra kernel as opposed to the flight data. This feature directly results from the decreased size of the uncertainty description resulting from analyzing the data simulated by the Volterra kernel. In effect, the reduced uncertainty description has limited the possible mechanisms that can cause flutter and so the worst-case speed is closer to the nominal speed.

Table 2 Flutter speeds for Mach 0.80

Method of determination	Flutter speed, knots of equivalent airspeed
Nominal model	431
Uncertainty from flight data	403
Uncertainty from Volterra kernel	413
Flight test	460

Note that the reduction in conservatism is not guaranteed to always result from utilizing the first-order Volterra kernel. The transfer functions representing the ATW show the magnitude of the response of the bending mode is greater for the Volterra kernel than for the data. In this case, the high-order components of the noise caused underestimation of the size of the bending mode response. The transfer function of the data is affected by these high-order components, but the transfer function of the first-order Volterra kernel only indicates linear components of the measurements.

Also, the true speed at which the ATW experienced flutter is noticeably higher than either of the robust predictions. The flutter speed of the nominal model is less than the true speed and so any robust speed must also be less than the true speed. The main issue is that the robust speeds based on Volterra responses are less conservative than those directly based on measured data.

Conclusions

This paper has introduced a method to determine uncertainty descriptions for use with robust flutter analysis. The use of Volterra kernels to represent flight data is an integral component of the method. The uncertainty is meant to describe errors in a linear model; therefore, the estimation of those errors should be based on differences between a linear model and linear data. This linear component of the flight data is computed by the first-order Volterra kernel. The uncertainty description is estimated using only this first-order kernel and so is not affected by nonlinearities. The removal of nonlinearities ensures the uncertainty only represents variations in the model that could affect the linear stability, or flutter, margins. This method is applied to flight data from the ATW. The method is able to generate more accurate descriptions of uncertainty in an analytical model and, thus, reduce conservatism in the robust flutter analysis. This paper indicates the benefits of using Volterra theory for flutter analysis and indicates similar benefit may be obtained by considering higher-order Volterra kernels for limit-cycle analysis.

References

[1] Lind, R., and Brenner, M., "Flutterometer: An On-Line Tool to Predict Robust Flutter Margins," *Journal of Aircraft*, Vol. 37, No. 6, 2000, pp. 1105–1112.

[2] Lind, R., and Brenner, M., *Robust Aeroservoelastic Stability Analysis*, Springer-Verlag, London, 1999, pp. 183–194.

[3] Brenner, M. J., "Aeroservoelastic Uncertainty Model Identification from Flight Data," *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 4, 2002, pp. 748–754.

[4] Schetzen, M., *The Volterra and Wiener Theories of Nonlinear Systems*, Wiley, New York, 1980, pp. 11–49.

[5] Prazenica, R., Kurdila, A., and Silva, W., "Multiresolution Methods for Representation of Volterra Series and Dynamical Systems," AIAA Paper 2000-1754, April 2000.

[6] Rugh, W. J., *Nonlinear System Theory: The Volterra-Wiener Approach*, Wiley, New York, 1980, pp. 50–90.

[7] Chen, Z., Micchelli, C., and Xu, Y., "A Construction of Interpolating Wavelets on Invariant Sets," *Mathematics of Computation*, Vol. 68, No. 228, 1999, pp. 1569–1587.

[8] Micchelli, C., and Xu, Y., "Using the Matrix Refinement Equation for the Construction of Wavelets on Invariant Set," *Applied and Computational Harmonic Analysis*, Vol. 1, Jan. 1994, pp. 391–401.

[9]Micchelli, C., and Xu, Y., "Reconstruction and Decomposition Algorithms for Biorthogonal Multiwavelets," *Multidimensional Systems and Signal Processing*, Vol. 8, Sept. 1997, pp. 31–69.

[10]Kurdila, A. J., Prazenica, R. J., Rediniotis, O., and Strganac, T., "Multiresolution Methods for Reduced-Order Models for Dynamic Systems," *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 2, 2001, pp. 193–200.

[11]Lind, R., "Flight Testing with the Flutterometer," AIAA Paper 2001-1654, April 2001.

[12]Marzocca, P., Librescu, L., and Silva, W. A., "Aeroelastic Response of Nonlinear Wing Sections Using a Functional Series Technique," *AIAA Journal*, Vol. 40, No. 5, April 2002, pp. 813–824.

[13]Voracek, D., Reaves, M., Horta, L., and Potter, S., "Piezoelectric Actuators for Ground and Flight Test Structural Excitation," AIAA Paper 2002-1349, 2002.

[14]"ZAERO Users Guide," Zona Technology, Scottsdale, AZ, 2000.

[15]Karpel, M., "Design for Active Flutter Suppression and Gust Load Alleviation Using State-Space Aeroelastic Modeling," *Journal of Aircraft*, Vol. 19, No. 3, 1982, pp. 221–227.

[16]Lind, R., and Brenner, M., "Flight Test Evaluation of Flutter Prediction Methods," AIAA Paper 2002-1649, April 2002.