aerospace

MDPI

Article

Continuation Methods for Nonlinear Flutter

Edward E. Meyer

Boreal Racing Shells, Seattle Rowing Center, 1116 West Ewing Street, Seattle, WA 98119, USA;
edward.e.meyer@borealracingshells.com; Tel.: +1-206-453-8778

Academic Editor: Konstantinos Kontis
Received: 27 October 2016; Accepted: 5 December 2016; Published: 9 December 2016

Abstract: Continuation methods are presented that are capable of treating frequency domain flutter equations, including multiple nonlinearities represented by describing functions. A small problem demonstrates how a series of continuation processes can find all limit-cycle oscillations within a specified region with a reasonable degree of confidence. Curves of the limit-cycle amplitude variation with velocity, indicating regions of stability and instability with colors, give a compact view of the nonlinear behavior throughout the flight regime. A continuation technique for reducing limit-cycle amplitudes by adjusting various system parameters is presented. These processes are economical enough to be a routine part of aircraft design and certification.

Keywords: aeroelasticity; multiple nonlinearity flutter; continuation methods; describing functions; bifurcation; continuation optimization; controlling LCO amplitudes

1. Introduction

In the commercial airplane industry certification that an aircraft is free from flutter instabilities is a major aspect of design and modification, requiring numerous analyses to cover the entire range of flight conditions and design parameters. Because it is impossible to analyze all flight conditions, such as altitude, speed, fuel loading and payload, limited sets of conditions are analyzed, often relying on engineering judgment to pick the important conditions. To do the many analyses required, techniques are typically limited to linear, frequency domain methods, using finite-element models reduced with low-frequency free-vibration modes and linear unsteady aerodynamics from the doublet-lattice method.

Linear flutter equations assume infinitesimally small displacements; nonlinear flutter equations are far more realistic, allowing for limit-cycle oscillation (LCO), self-excited, constant amplitude oscillations that may be harmful or merely ride-quality problems depending on the amplitude. The additional information provided by nonlinear analyses is important to the design of commercial aircraft and should become a routine part of industrial flutter analyses.

Continuation methods, a class of methods for solving parameterized nonlinear equations [1], when applied to linear flutter equations have proven advantageous for covering the range of flight conditions by allowing for continuous variations in parameters, thereby reducing the number of discrete conditions necessary [2]. For example, interpolating mass matrices at several fuel loadings allows tracing the variation of flutter points with fuel loading in a continuous fashion. The success of continuation methods in treating linear flutter equations is due to the fact that in spite of the term linear flutter, the equations are nonlinear in several parameters, but linear in the generalized coordinates (g.c.). Thus, it is a small step to extend continuation methods to treat nonlinear flutter equations, identifying LCOs and tracing the variation of amplitude with parameters, such as velocity.

Single nonlinearities, for example free play or bilinear stiffness in one discrete degree-of-freedom, have been studied for many years [3,4]. A single nonlinearity modeled with a describing function is readily treated with linear flutter techniques, since the generalized coordinates can be normalized to satisfy the amplitude of the single generalized coordinate. Multiple nonlinearities present a conceptually more difficult task because of the need to simultaneously adjust the generalized coordinates so that they and the resulting nonlinearities satisfy the flutter equations. Various schemes have been proposed to address this problem [5,6]. Here, it is shown how a continuation method designed for linear flutter analysis can also be used with multiple nonlinearities with minor modifications. The result is a method that is straightforward, efficient and familiar to flutter engineers.

Following a brief summary of the continuation method presented in [2] and its application to nonlinear flutter equations, a small problem is used to illustrate the search for limit cycle oscillations. With a series of continuation processes, the region of interest is covered with a grid of curves that find almost all LCOs in the region. Because of the sparsity of the grid, an LCO is missed, but is discovered using a continuation method that traces an optimal path toward a limit cycle. Starting from the limit cycles encountered, curves of LCO amplitude versus velocity are traced with continuation. From one of these curves at a specified velocity, the optimal-path continuation method is used to reduce the LCO amplitude by adjusting gains and phases in a simple control system.

2. Continuation Method

Continuation methods solve systems of nonlinear equations that are functions of more variables than equations, at discrete points over a range of the variables while maintaining continuity in the nonlinear functions. Continuity in flutter equations is important because there are always multiple solutions, known as aeroelastic modes, which occasionally become close and difficult to distinguish.

The problem to solve is:

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0} \in \mathbb{R}^m, \quad \boldsymbol{x} \in \mathbb{R}^n, \quad n > m \tag{1}$$

starting from a known solution $\boldsymbol{x}_0$ and varying the independent variables through the desired range. $n - m$ is the dimensionality of the solution: curves (1); surfaces (2); volumes (3), etc. The focus here is on curves: $n = m + 1$.

The continuation method chosen to solve this is known as a pseudo-arc length method [1] with a minimum-norm corrector. At step j, the next point is predicted using the tangent to the curve and the known solution $\boldsymbol{x}_j$:

$$\boldsymbol{x}_{j+1}^0 = \boldsymbol{x}_j + \alpha \boldsymbol{t}_j \tag{2}$$

where the stepsize α is an approximate distance along the curve, hence the name pseudo-arc length. Starting with the prediction $\boldsymbol{x}_{j+1}^0$, the corrector iterates to find the solution $\boldsymbol{x}_{j+1}$ using a Newton-like method, computing corrections $\boldsymbol{h}^i$ satisfying:

$$\boldsymbol{J}(\boldsymbol{x}^i)\boldsymbol{h}^i = -\boldsymbol{f}(\boldsymbol{x}^i) \tag{3}$$

and setting:

$$\boldsymbol{x}^{i+1} = \boldsymbol{x}^i + \boldsymbol{h}^i, \tag{4}$$

where:

$$\boldsymbol{J}(\boldsymbol{x}) = \boldsymbol{f}'(\boldsymbol{x}) = \left[\frac{\partial f_i}{\partial x_j}\right] \in \mathbb{R}^{m \times n} \tag{5}$$

is the Jacobian matrix of derivatives of $\boldsymbol{f}$ with respect to $\boldsymbol{x}$. The tangent vector ($\boldsymbol{t}$) is characterized by:

$$\boldsymbol{J}\boldsymbol{t} = \boldsymbol{0}, \quad \|\boldsymbol{t}\|_2 = \sqrt{\boldsymbol{t}^T\boldsymbol{t}} = 1, \quad \boldsymbol{t} = \frac{\partial \boldsymbol{x}}{\partial \tau} \tag{6}$$

where τ is the arc length along the curve.

Equation (3) is an underdetermined linear system that has an infinity of solutions; the smallest such solution can be computed by factoring the transposed Jacobian into an (n, n) orthogonal matrix $\boldsymbol{Q}$ times an (n, m) upper-triangular matrix $\boldsymbol{R}$:

$$\boldsymbol{J}^T = \boldsymbol{QR} = [\boldsymbol{Q}_1 \ \ \boldsymbol{Q}_2] \begin{bmatrix} \boldsymbol{R}_1 \\ \boldsymbol{0} \end{bmatrix}, \tag{7}$$

known as a QR factorization. Substituting Equation (7) into Equation (3) yields the minimum-norm solution [7] (p. 300):

$$\boldsymbol{h}^i = -\boldsymbol{Q}_1 \boldsymbol{R}_1^{-T} \boldsymbol{f}(\boldsymbol{x}^i) \tag{8}$$

Corrector iterations continue using Equations (3)–(8) until suitable convergence criteria are met, for example for some absolute tolerance ϵ_a and relative tolerance ϵ_r:

$$\|\boldsymbol{f}(\boldsymbol{x}^{i+1})\|_2 < \epsilon_a \quad \text{and} \quad \|\boldsymbol{h}^i\|_2 < \epsilon_a + \epsilon_r \|\boldsymbol{x}^{i+1}\|_2 \tag{9}$$

In preparation for the next predictor step, the tangent is computed from the $(n, n-m)$ matrix $\boldsymbol{Q}_2$ in Equation (7), a by-product of the QR factorization. $\boldsymbol{Q}_2$ is an orthogonal matrix spanning the null space of the Jacobian,

$$\begin{aligned} \boldsymbol{J}\boldsymbol{Q}_2 &= \boldsymbol{0} \ \in \mathbb{R}^{m \times n-m} \\ \boldsymbol{Q}_2^T \boldsymbol{Q}_2 &= \boldsymbol{I} \ \in \mathbb{R}^{n-m \times n-m} \end{aligned} \tag{10}$$

An arbitrary vector $\boldsymbol{w}$ is projected onto the null space with:

$$\bar{\boldsymbol{t}} = \boldsymbol{Q}_2 \boldsymbol{Q}_2^T \boldsymbol{w} = \boldsymbol{Q} \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_2 \end{bmatrix} \boldsymbol{Q}^T \boldsymbol{w} \ \in \mathbb{R}^n \tag{11}$$

where $\boldsymbol{I}_2$ is the $(n-m, n-m)$ identity and $\boldsymbol{t} = \bar{\boldsymbol{t}} / \|\bar{\boldsymbol{t}}\|_2$ to satisfy Equation (6). If $n = m + 1$, $\boldsymbol{Q}_2$ consists of one column, the projection only determines the sign of the tangent, and a logical choice for the vector to project is the tangent at the previous continuation step to keep the curve tracking in the right direction. In this case, it is not absolutely necessary to do the projection, but it avoids forming $\boldsymbol{Q}_2$ (see Section 8) and is necessary for the optimal-path technique (Sections 5.6, 5.8 and 6).

3. Flutter Equations

Treating the flutter equation in the frequency domain has computational advantages over the time domain: the characteristic equation is a system of nonlinear algebraic equations instead of a system of differential equations, which must be integrated at each flight condition. By contrast, the algebraic equations can be solved over a continuous range of flight conditions. The disadvantage is that nonlinearities must be approximated to conform to the harmonic motion assumption, Equation (12).

3.1. Assumed Motion

Linear, frequency domain flutter analyses are based on the assumption that the actual motion of the structure ($\boldsymbol{z}$) as a function of time is related to a set of complex generalized coordinates ($\hat{\boldsymbol{q}}$) by:

$$\boldsymbol{z}(t) = \boldsymbol{\Phi} \Re(\hat{\boldsymbol{q}} e^{st}) \tag{12}$$

where $\boldsymbol{\Phi}$ is typically a matrix of vibration modes, t is time, $s = \sigma + i\omega$ is the Laplace variable, ω is the frequency and oscillations are growing, neutral or decaying if the growth factor σ is positive, zero or negative, respectively.

The actual motion in a nonlinear frequency domain flutter analysis does not in general follow this assumption; retaining this assumption results in an a quasi-linear approximation [8].

Describing functions and harmonic balance methods are examples of quasi-linear approximations. Any approximation conforming to these assumptions should work with the methods presented here.

3.2. Equations

Various formulations of the flutter equations have been proposed [9–13]; any of them can be used with the continuation method presented above. A general form of the frequency domain flutter characteristic equations is:

$$\left[s^2\boldsymbol{M} + s\boldsymbol{G} + s\boldsymbol{V} + (1+id)\boldsymbol{K} - q\boldsymbol{A}(p,M) + \boldsymbol{T}\right]\hat{\boldsymbol{q}} = \hat{\boldsymbol{D}}\hat{\boldsymbol{q}} = \boldsymbol{0}, \tag{13}$$

where $\boldsymbol{M}$, $\boldsymbol{K}$, $\boldsymbol{G}$, $\boldsymbol{V}$, $\boldsymbol{A}$ and $\boldsymbol{T}$ are the (n_s, n_s) mass, stiffness, gyroscopic, viscous damping, unsteady aerodynamic and control-system matrices, respectively, d is the structural damping coefficient, $q = \rho V^2/2$ is the dynamic pressure, $p = s/V$ is the complex reduced frequency, V is the free-stream velocity, $M = V/a$ is the free-stream Mach number, a is the sonic velocity and $\hat{\boldsymbol{D}}$ is the complex dynamic matrix.

The related equation:

$$\hat{\boldsymbol{D}}(s)\hat{\boldsymbol{y}} = \boldsymbol{0}, \quad \|\hat{\boldsymbol{y}}\|_2 = \sqrt{\hat{\boldsymbol{y}}^*\hat{\boldsymbol{y}}} = 1, \quad \Im(\hat{y}_k) = 0 \tag{14}$$

is a nonlinear eigenvalue problem with n_s solution eigenpair $(s_i, \boldsymbol{y}_i)$, $i = 1, \ldots n_s$, normalized with the 2-norm of $\hat{\boldsymbol{y}}$ 1.0 and the k-th component real. Contrast this with the generalized-coordinate vector $\hat{\boldsymbol{q}}$, which is a factor, η times $\hat{\boldsymbol{y}}$:

$$\hat{\boldsymbol{q}} = \eta\hat{\boldsymbol{y}}, \quad \|\hat{\boldsymbol{q}}\|_2 = \eta \tag{15}$$

This distinction is important when the dynamic matrix is a function of generalized-coordinate amplitudes, particularly when η is small or zero.

3.3. Conversion to Real

The continuation method presented is for real equations and variables; the flutter equations are converted from a set of n_s complex equations to $2n_s$ real equations by setting:

$$\boldsymbol{D} = \begin{bmatrix} \Re(\hat{D}_{ij}) & -\Im(\hat{D}_{ij}) \\ \Im(\hat{D}_{ij}) & \Re(\hat{D}_{ij}) \end{bmatrix} \quad \boldsymbol{y} = \begin{Bmatrix} \Re(\hat{y}_i) \\ \Im(\hat{y}_i) \end{Bmatrix} \quad \boldsymbol{q} = \eta\boldsymbol{y}, \tag{16}$$

so that:

$$\boldsymbol{D}\boldsymbol{y} = \boldsymbol{D}\boldsymbol{q} = \boldsymbol{0} \quad \in \mathbb{R}^{2n_s} \tag{17}$$

and the 2-norms of the eigenvectors and generalized-coordinate vectors are unchanged:

$$\begin{aligned} \|\hat{\boldsymbol{y}}\|_2 &= \sqrt{\hat{\boldsymbol{y}}^*\hat{\boldsymbol{y}}} = \|\boldsymbol{y}\|_2 = \sqrt{\boldsymbol{y}^T\boldsymbol{y}} = 1 \\ \|\hat{\boldsymbol{q}}\|_2 &= \sqrt{\hat{\boldsymbol{q}}^*\hat{\boldsymbol{q}}} = \|\boldsymbol{q}\|_2 = \sqrt{\boldsymbol{q}^T\boldsymbol{q}} = \eta \end{aligned} \tag{18}$$

where $\hat{(\ )}^*$ is the conjugate transpose.

3.4. Matrix Parameterizations

The matrices in Equation (13) are, in general, not constant; rather, they are assumed to be functions of problem-dependent parameters using various matrix parameterizations. For example, aircraft mass matrices are often parameterized by payload and fuel loading by interpolating matrices at several parameter values. More important for this study, the matrices might be parameterized by generalized-coordinate amplitudes in various ways depending on the matrix and type of nonlinearity.

The dynamic matrix is therefore a function of the generalized coordinates and a vector $\boldsymbol{p}$ of parameters, such as V, σ, and ω: $\boldsymbol{D} = \boldsymbol{D}(\boldsymbol{p}, \boldsymbol{q})$. Common examples of nonlinearities include structural free play, nonlinear stiffnesses and control systems.

Nonlinear structural elements, such as free play and nonlinear stiffness, can be approximated using describing functions, which conform to the harmonic motion assumption, Equation (12). These stiffness elements are usually modeled with a single degree-of-freedom with no stiffness coupling between it and other degrees-of-freedom [4]. A describing function for bilinear stiffness k_{jj} associated with generalized coordinate j, with stiffness k_0 when the displacement amplitude $|\hat{q}_j|$ is below δ and k_1 when it is above, is:

$$\begin{aligned} k_{jj}(\gamma, r) &= c(\gamma, r)k_0 \\ c(\gamma, r) &= \begin{cases} r + \frac{2}{\pi}(1-r)\left[\sin^{-1}(\gamma) + \gamma\sqrt{1-\gamma^2}\right] & \text{if } \gamma \le 1 \\ 1 & \text{otherwise} \end{cases} \\ r &= \frac{k_1}{k_0}, \qquad \gamma = \frac{\delta}{|\hat{q}_j|} \end{aligned} \tag{19}$$

Nonlinear controls equations can be treated by describing functions [8]; one of the first multiple- nonlinearity flutter solutions was with structural nonlinearities in the control system [6].

3.5. Normalization

To make solutions unique, a phase normalization on the generalized coordinates is necessary, for example by constraining the k-th component of $\hat{\boldsymbol{q}}$ to be real:

$$f_{2n_s+1} = \Im(\hat{q}_k) = q_{2k} = 0 \tag{20}$$

If in addition η is to be held to a constant value η_0, another equation is added:

$$f_{2n_s+2} = \boldsymbol{q}^T\boldsymbol{q} - \eta_0^2 = 0 \tag{21}$$

Therefore, to trace curves $2n_s + 2$, independent variables must be used if η is allowed to vary, or $2n_s + 3$ if it is held constant.

3.6. Continuation Formulations

The independent variables ($\boldsymbol{x}$) comprise the $2n_s$ variables $\boldsymbol{y}$ or $\boldsymbol{q}$, plus enough additional parameters to give an underdetermined system. There are two cases to consider: constant η with $2n_s + 3$ variables and variable η with $2n_s + 2$ variables.

In what follows, various combinations of velocity, frequency, σ and η are used to demonstrate the solution technique; other parameters are possible, but these three serve to illustrate typical analyses. Each combination of parameters will be named according to which three of these parameters vary; the fourth is implicitly constant. For example, a V-σ-ω analysis traces the variation in σ and ω with velocity, holding η constant, as in a traditional linear flutter analysis, and a V-ω-η analysis traces LCO boundaries, the goal of this study.

3.6.1. Constant η

Solving the flutter equations with the generalized-coordinate norm held to a constant value η_0 presents a problem when $\eta_0 = 0$ and possibly numerical difficulties when η_0 is small. The vector $\boldsymbol{q} = \boldsymbol{0}$ is known as the trivial solution because even though it is technically a solution, it provides no real information about the behavior of the structure. Yet solutions at $\boldsymbol{q} = \boldsymbol{0}$ are important for determining the stability of the trivial solution, the basis of linear flutter solutions. For this reason, when η is held constant, eigenvectors are used as variables instead of the generalized coordinates.

The eigenvector is constrained to unit 2-norm with the addition of an equation like (21): $\boldsymbol{y}^T\boldsymbol{y} - 1 = 0$, so there are $m = 2n_s + 2$ equations, and three more parameters are necessary for $n = m + 1 = 2n_s + 3$ independent variables.

For the demonstration parameters, the equations are:

$$\begin{aligned}
\boldsymbol{f}(\boldsymbol{x}) &= \left\{ \begin{array}{c} \boldsymbol{D}\boldsymbol{y} \\ y_{2k} \\ \boldsymbol{y}^T\boldsymbol{y} - 1 \end{array} \right\} \in \mathbb{R}^m \\
\boldsymbol{x} &= \left\{ \begin{array}{c} V \\ \sigma \\ \omega \\ \boldsymbol{y} \end{array} \right\} \in \mathbb{R}^{m+1} \\
\boldsymbol{q} &= \eta_0 \boldsymbol{y} \\
\boldsymbol{J} &= \left[\begin{array}{cccc} \frac{\partial \boldsymbol{D}}{\partial V}\boldsymbol{y} & \frac{\partial \boldsymbol{D}}{\partial \sigma}\boldsymbol{y} & \frac{\partial \boldsymbol{D}}{\partial \omega}\boldsymbol{y} & \eta_0 \frac{\partial \boldsymbol{D}}{\partial \boldsymbol{q}}\boldsymbol{y} + \boldsymbol{D} \\ 0 & 0 & 0 & \boldsymbol{I}_{2k:} \\ 0 & 0 & 0 & 2\boldsymbol{y}^T \end{array} \right] \in \mathbb{R}^{m \times m+1} \qquad (22)
\end{aligned}$$

where $\boldsymbol{I}_{2k:}$ is row $2k$ of the order $2n_s$ identity matrix.

3.6.2. Variable η

If η is allowed to vary, the additional normalization Equation (21) is not used, so there are $m = 2n_s + 1$ equations: Equation (17) plus the phase normalization (Equation (20)),

$$\boldsymbol{f}(\boldsymbol{x}) = \left\{ \begin{array}{c} \boldsymbol{D}\boldsymbol{q} \\ q_{2k} \end{array} \right\} \in \mathbb{R}^m \qquad (23)$$

and two of the three parameters (V, σ, ω) are needed.

Two combinations of the three parameters will be of interest, σ-ω-η at constant velocity (used to search for $\sigma = 0$) with:

$$\boldsymbol{x} = \left\{ \begin{array}{c} \sigma \\ \omega \\ \boldsymbol{q} \end{array} \right\} \in \mathbb{R}^{m+1} \qquad \boldsymbol{J} = \left[\begin{array}{ccc} \frac{\partial \boldsymbol{D}}{\partial \sigma}\boldsymbol{q} & \frac{\partial \boldsymbol{D}}{\partial \omega}\boldsymbol{q} & \frac{\partial \boldsymbol{D}}{\partial \boldsymbol{q}}\boldsymbol{q} + \boldsymbol{D} \\ 0 & 0 & \boldsymbol{I}_{2k:} \end{array} \right] \in \mathbb{R}^{m \times m+1} \qquad (23a)$$

and V-ω-η at constant σ (used to trace LCO boundaries):

$$\boldsymbol{x} = \left\{ \begin{array}{c} V \\ \omega \\ \boldsymbol{q} \end{array} \right\} \in \mathbb{R}^{m+1} \qquad \boldsymbol{J} = \left[\begin{array}{ccc} \frac{\partial \boldsymbol{D}}{\partial V}\boldsymbol{q} & \frac{\partial \boldsymbol{D}}{\partial \omega}\boldsymbol{q} & \frac{\partial \boldsymbol{D}}{\partial \boldsymbol{q}}\boldsymbol{q} + \boldsymbol{D} \\ 0 & 0 & \boldsymbol{I}_{2k:} \end{array} \right] \in \mathbb{R}^{m \times m+1} \qquad (23b)$$

If the first step in either of these processes starts from a known solution at $\eta = 0$ with:

$$\boldsymbol{x}_0 = \left\{ \begin{array}{c} \sigma_0 \\ \omega_0 \\ \boldsymbol{0} \end{array} \right\} \quad \text{or} \quad \boldsymbol{x}_0 = \left\{ \begin{array}{c} V_0 \\ \omega_0 \\ \boldsymbol{0} \end{array} \right\} \qquad (24)$$

a modification in the tangent calculation is necessary, because the Jacobian and its null space vectors are then:

$$\boldsymbol{J} = \left[\begin{array}{ccc} 0 & 0 & \boldsymbol{D} \\ 0 & 0 & \boldsymbol{I}_{2k:} \end{array} \right], \quad \boldsymbol{Q}_2 = \left[\begin{array}{ccc} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{y} \end{array} \right] \qquad (25)$$

The third column of $\boldsymbol{Q}_2$ gives the desired greatest increase in η, so the first predicted solution for the σ-ω-η process is:

$$\boldsymbol{x}_1^0 = \boldsymbol{x}_0 + \alpha \boldsymbol{t}_0 = \left\{ \begin{array}{c} \sigma_0 \\ \omega_0 \\ \boldsymbol{0} \end{array} \right\} + \alpha \left\{ \begin{array}{c} 0 \\ 0 \\ \boldsymbol{y} \end{array} \right\} = \left\{ \begin{array}{c} \sigma_0 \\ \omega_0 \\ \alpha \boldsymbol{y} \end{array} \right\} \tag{26}$$

In addition, an equation should be added to prevent converging to the trivial solution ($\boldsymbol{q} = \boldsymbol{0}$),

$$f_{2n_s+2} = \boldsymbol{q}^T \boldsymbol{q} - \eta^0 = 0 \tag{27}$$

where η^0 is the 2-norm of the predicted $\boldsymbol{q}$. For the first step, $\eta^0 = \alpha$.

4. Demonstration Problem

A small, well-known problem illustrates the use of these continuation processes. The model is Example 9-1 in [14], implemented as Example HA145B in finite-element program MSC-NASTRAN [13], with a generalized-coordinate basis of $n_s = 10$ free-vibration modes and doublet-lattice aerodynamics. Bilinear nonlinearities are applied to the first four diagonals of the (diagonal) stiffness matrix using the describing function of Equation (19) with $\delta = 0.05$, $r = 2$ (hardening) for diagonals 1 and 2, $r = 0.5$ (softening) for diagonal 3 and $r = 0.3$ (softening) for diagonal 4. Figure 1 shows the model wing planform and unsteady aerodynamic grid.

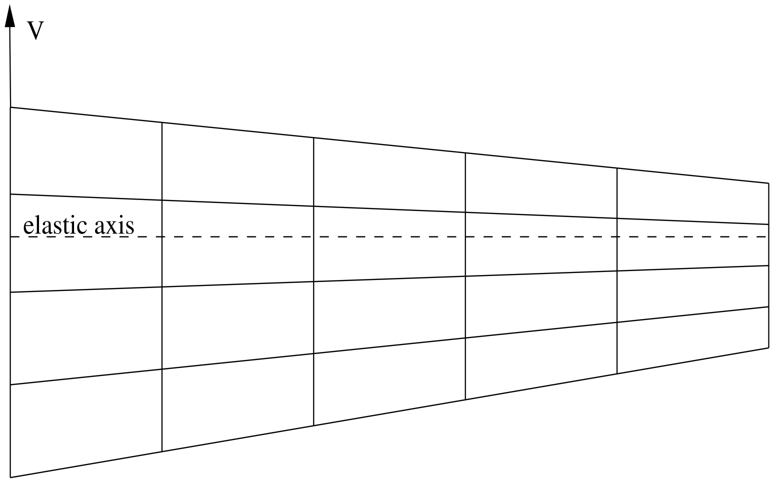

Figure 1. HA145B planform, elastic axis (dashed line), and doublet-lattice grid.

5. Searching for Limit Cycle Oscillations

Limit cycle oscillation is a condition where the structure is oscillating with an amplitude that is neither growing nor decaying; it is in dynamic equilibrium and $\sigma = 0$. On the other hand, $\boldsymbol{q} = \boldsymbol{0}$ is a static equilibrium with stability determined by the sign of σ. The goal then is to find all points within the region of interest where $\sigma = 0$.

The strategy is to find several points where $\sigma = 0$ using the formulations of Sections 3.6.1 and 3.6.2, then use these points to start V-ω-η processes to trace the variation of LCO amplitude with velocity.

An example of the results of an LCO analysis is shown in Figure 2 where η is plotted against velocity normalized with respect to the maximum velocity of interest, $v = V/V_{max}$. Ordinarily, η would not be the plot variable of choice; instead, one or more of the nonlinear generalized-coordinate amplitudes would be plotted against velocity, but this choice is problem dependent, so η is used for generality. Of the two curves in Figure 2, only the bottom one extends to $\eta = 0$, so it is the only one that would be detected in a linear flutter analysis. Detecting the upper curve is why the processes of Sections 3.6.1 and 3.6.2 are necessary.

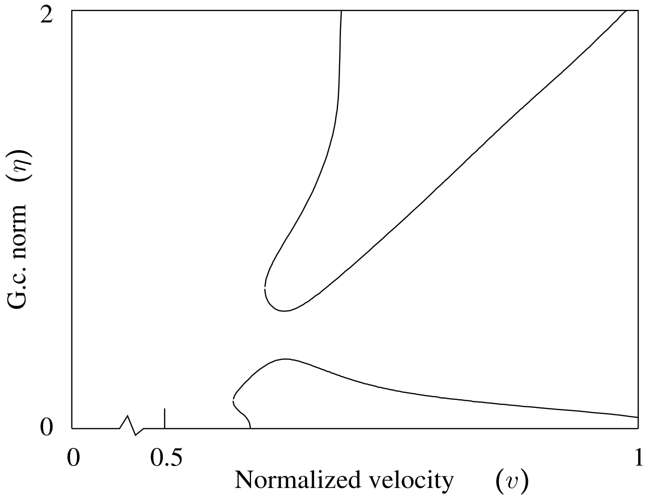

Figure 2. Typical limit-cycle oscillation (LCO) plot from a V-ω-η analysis.

A systematic method for this search is to cover the range with a grid of V-σ-ω and σ-ω-η processes, any of which might encounter $\sigma = 0$ points. Figure 3 shows a minimal grid of six lines where horizontal lines are V-σ-ω processes (numbered 1, 3 and 4), and vertical lines are σ-ω-η processes (numbered 2, 5 and 6). Arrows indicate the direction of the continuation processes. Each of these lines represents one or more modes chosen to cover the frequency range of interest.

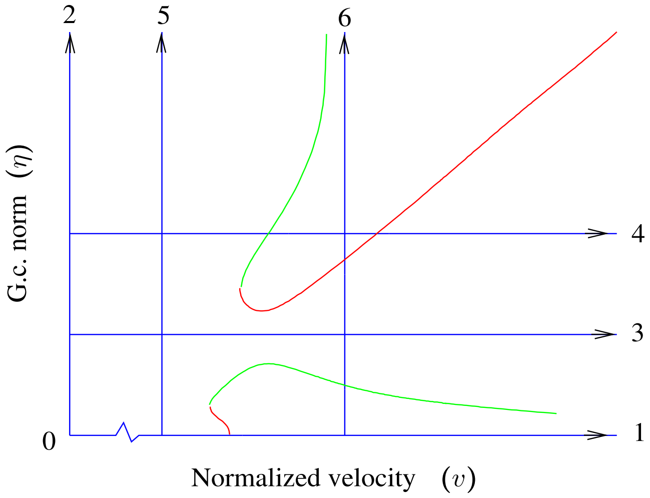

Figure 3. Grid (blue) for finding limit cycles. Green means stable, red means unstable.

This set of processes is started from a free-vibration solution, at the origin of Figure 3 (Point 0), with $v = \eta = 0$. Two processes are started from this free-vibration solution: Process 1, a V-σ-ω process along the v axis, and Process 2, a σ-ω-η process along the η axis. The remaining vertical and horizontal grid lines are started from these two by interpolating to the desired points.

5.1. Point 0: Free-Vibration

At $V = 0$, the aerodynamic term drops out, and with $\eta = 0$, $\boldsymbol{q}$ is zero for the purposes of evaluating the dynamic matrix $\boldsymbol{D}$. The problem to be solved then is in general the complex nonlinear eigenvalue problem, Equation (14). Under certain conditions, Equation (14) reduces to a polynomial eigenvalue problem [15], or to a generalized eigenvalue problem [16]. For example, in the absence of controls equations, viscous and structural damping and gyroscopics, the eigenvalue problem reduces to the real, generalized eigenvalue problem:

$$\left[-\omega^2 \boldsymbol{M} + \boldsymbol{K}\right] \hat{\boldsymbol{y}} = \boldsymbol{0}, \quad \boldsymbol{y} = \left\{ \begin{array}{c} \hat{y}_i \\ 0 \end{array} \right\} \tag{28}$$

Otherwise, it is usually necessary to resort to nonlinear eigenvalue solution techniques [15].

Using the results of this eigensolution, representing the origin in Figure 2, the LCO search is begun with two continuation processes corresponding to the two axes of Figure 2: a V-σ-ω analysis at $\eta = 0$ (the horizontal or V axis) and a σ-ω-η analysis at $V = 0$ (the vertical or η axis).

5.2. Process 1: V-σ-ω with $\eta_0 = 0$

A continuation process along the horizontal axis of Figure 2 is a V-σ-ω process at $\eta_0 = 0$, solving Equation (22), starting with select eigenvalues and eigenvectors from the free-vibration solution. This is equivalent to a linear flutter solution. Figure 4 shows results for the first four lowest frequency modes with green indicating a stable equilibrium and red unstable.

From these curves σ-ω-η Processes 5 and 6 in Figure 3 are started at the desired velocities by interpolating among solution points. Additionally, a V-ω-η processes could be started from the point where a curve crosses the $\sigma = 0$ axis, tracing the bottom curve in Figure 2.

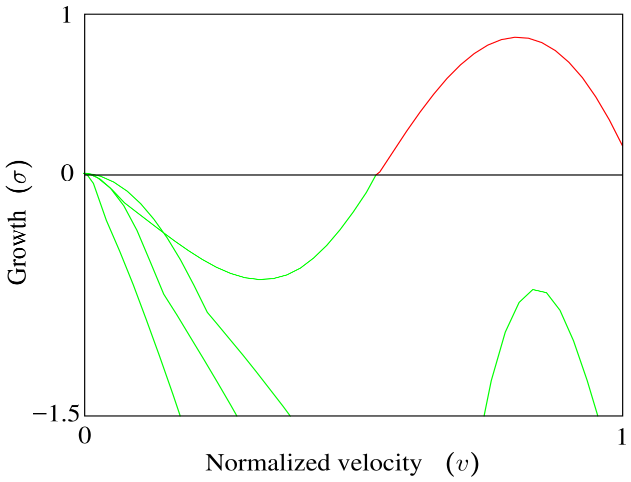

Figure 4. Process 1: V-σ-ω analysis with $\eta_0 = 0$ (Green means stable, red means unstable).

5.3. Process 2: σ-ω-η at $V = 0$

In a σ-ω-η process η is allowed to vary; velocity is held constant; and the Equations are (23) and (23a). With $V = 0$, this process corresponds to the vertical axis of Figure 2. Start points for this process are again select eigenvalues and eigenvectors from the free-vibration solution. At zero velocity, the aerodynamic term drops out, and without viscous or structural damping, $\sigma = 0$ for each vibration mode. Only the frequency changes with increasing η, as shown in Figure 5.

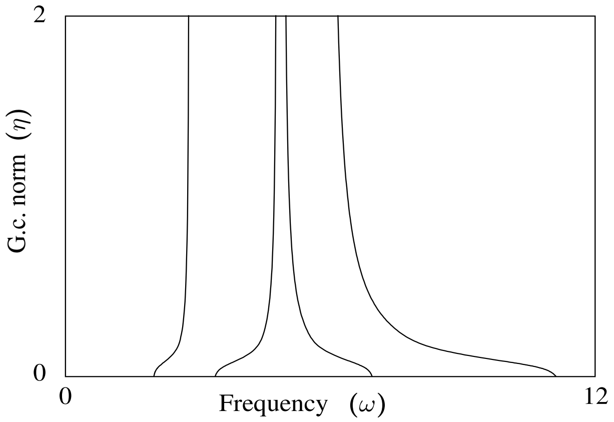

Figure 5. Process 2: σ-ω-η analysis with $V = 0$.

5.4. Processes 3 and 4: V-σ-ω

Horizontal lines 3 and 4 in Figure 3 are V-σ-ω processes starting from Process 2 interpolated to the desired values of η. Figure 6 shows the result of these processes; as before, green indicates stability (negative σ), and red indicates instability. Three levels of η are highlighted for the only mode that crosses $\sigma = 0$. These crossings will be used in the V-ω-η processes of Section 3.6.2 to trace LCO boundaries.

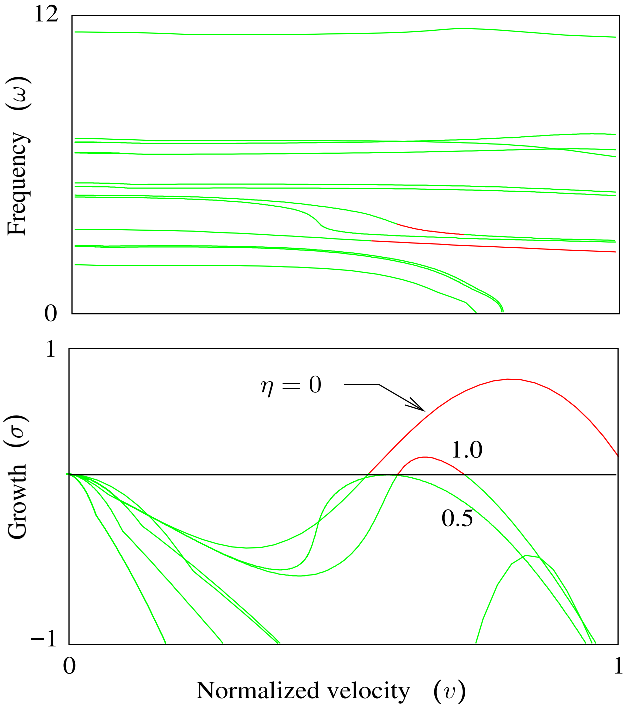

Figure 6. Processes 1, 3 and 4: V-σ-ω at three η (Green means stable, red means unstable).

5.5. Processes 5 and 6: σ-ω-η

σ-ω-η Processes 5 and 6 are started by interpolating the curves for each mode traced in Process 1 to the desired velocity. Figure 7 shows σ versus η at three velocities; for clarity, only one of the four modes traced in Process 1 is shown, the only one that encounters LCO.

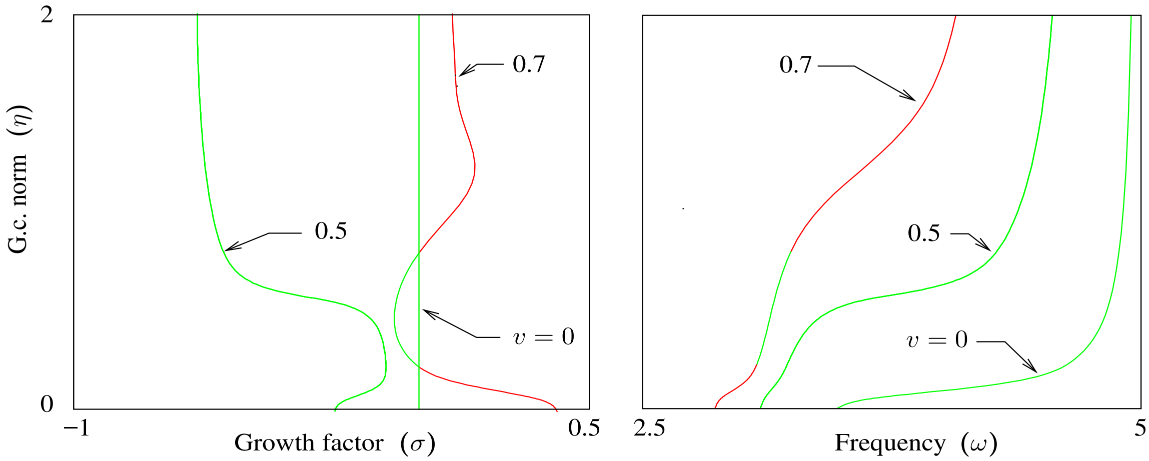

Figure 7. Processes 2, 5 and 6: σ-ω-η at 3 normalized velocities (Green means stable, red means unstable).

5.6. Stability of Equilibria

At an LCO, if a small perturbation in the generalized coordinates decreases η and η continues to decrease with time, or if the perturbation increases η and η continues to increase with time, the equilibrium is unstable. On the other hand, if after the perturbation, the generalized coordinates return to the LCO amplitudes, the equilibrium is said to be stable. Mathematically, this means that stability is determined by the sign of the derivative of σ with respect to η: positive indicates instability, and negative indicates stability. This derivative can be computed from the components of the tangent vector corresponding to σ and $\boldsymbol{q}$, recalling Equation (6),

$$\frac{\partial \sigma}{\partial \eta} = \frac{\partial \sigma}{\partial \tau}\frac{\partial \tau}{\partial \eta} = \frac{t_\sigma}{t_\eta}, \quad t_\eta = \frac{\partial \eta}{\partial \tau} = \frac{1}{\eta}\sum_{i=1}^{2n_s} q_i \frac{\partial q_i}{\partial \tau} = \frac{1}{\eta}\sum_{i=1}^{2n_s} q_i t_{q_i} \tag{29}$$

where τ again is the arc length along the curve. These tangent components are available at every step of a σ-ω-η process, but this information is only relevant at $\sigma = 0$. More important is the stability of each point of LCO boundaries, created with V-ω-η processes and Equation (23b). That Jacobian then must be converted to the Jacobian of Equation (23a) by replacing the first column with $\partial \boldsymbol{f}/\partial \sigma$,

$$\bar{\boldsymbol{J}} = \boldsymbol{J} + \boldsymbol{v}\boldsymbol{u}^T \tag{30}$$

where:

$$\boldsymbol{u} = \begin{Bmatrix} 1 \\ 0 \\ \vdots \end{Bmatrix} \in \mathbb{R}^n \qquad \boldsymbol{v} = \begin{Bmatrix} \frac{\partial \boldsymbol{f}}{\partial \sigma} \end{Bmatrix} - \boldsymbol{J}_{:1} \quad \in \mathbb{R}^m \tag{31}$$

It is not necessary to actually make this replacement and re-factorize the Jacobian; only the vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ are of interest because with them, the QR factorization of the transposed Jacobian can be updated [7] (p. 334):

$$\bar{\boldsymbol{J}}^T = \boldsymbol{J}^T + \boldsymbol{u}\boldsymbol{v}^T = \boldsymbol{Q}\boldsymbol{R} + \boldsymbol{u}\boldsymbol{v}^T = \bar{\boldsymbol{Q}}\bar{\boldsymbol{R}} = [\bar{\boldsymbol{Q}}_1 \;\; \bar{\boldsymbol{Q}}_2] \begin{bmatrix} \bar{\boldsymbol{R}}_1 \\ \boldsymbol{0} \end{bmatrix} \tag{32}$$

Updating the factorization requires a small fraction of the work required to factorize the Jacobian. From the updated Jacobian, the tangent used in Equation (29) is computed by projecting $\boldsymbol{u}$ onto the updated null space with Equation (11):

$$\boldsymbol{t} = \bar{Q}_2 \bar{Q}_2^T \boldsymbol{u} = \bar{Q} \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{I}_2 \end{bmatrix} \bar{Q}^T \boldsymbol{u} \quad \in \mathbb{R}^n \tag{33}$$

As η increases or decreases from an unstable equilibrium, the next equilibrium may be stable or unstable. Often, an unstable equilibrium will be followed by a stable one; Figure 7 shows why this is true when the equilibriums are associated with the same aeroelastic mode. That this is not always true will be shown in Section 5.8, where an LCO is discovered associated with a different aeroelastic mode.

5.7. Limit Cycles: V-ω-η Analysis at $\sigma = 0$

Among the six sets of continuation curves in Figures 6 and 7, there are five points where $\sigma = 0$, at the intersection of the green and red line segments in Figure 8.

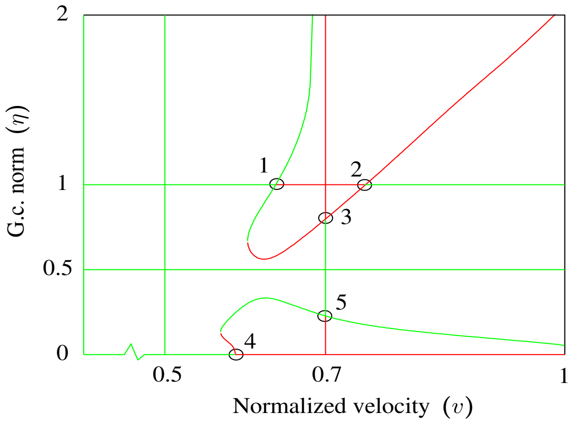

Figure 8. Five start points for LCO curves (Green means stable, red means unstable).

These points can be used to start V-ω-η processes, the goal in this exercise, shown as dotted curves in Figure 8. The upper dotted curve could be started from Point 1, 2 or 3; either start point will trace the same curve. Likewise, the lower dotted curve could be started from Point 4 or 5.

5.8. Optimal Path: V-σ-ω-η Analysis

Two of the grid lines in Figure 3 do not cross the $\sigma = 0$ axis ($\eta = 0.5$ in Figure 6 and $v = 0.5$ in Figure 7), but come close. If all four parameters were allowed to vary instead of holding one fixed and varying the other three, those two continuation processes might have also found limit cycle points. A continuation process where the number of unknowns is greater than the number of equations plus one requires only a minor modification of the continuation process and is in a sense an optimization procedure.

The idea is to allow any number of independent variables greater than the number of equations (m) and to trace a special curve on the surface (or volume, or higher-dimension manifold): the shortest path toward some goal, for example maximizing or minimizing some parameter β. The key to tracing this curve is the tangent to the curve, computed by projecting the vector:

$$\boldsymbol{w} = \left\{ \frac{\partial \beta}{\partial x_i} \right\} \tag{34}$$

onto the null space of the Jacobian (Equation (11)). Aside from this modification, the continuation process is unchanged.

Here, the goal is $\sigma = 0$, a limit-cycle oscillation, and the equations are:

$$f(x) = \left\{ \begin{array}{c} \boldsymbol{Dq} \\ q_{2k} \end{array} \right\} \in \mathbb{R}^{2n_s+1} \qquad \boldsymbol{x} = \left\{ \begin{array}{c} V \\ \sigma \\ \omega \\ \boldsymbol{q} \end{array} \right\} \in \mathbb{R}^{2n_s+3} \tag{35}$$

Assuming the start point is at a negative σ, the tangent vector is computed by projecting:

$$\boldsymbol{w} = \left\{ \frac{\partial \sigma}{\partial x_i} \right\} = \left\{ \begin{array}{c} 0 \\ 1 \\ 0 \\ \boldsymbol{0} \end{array} \right\} \tag{36}$$

onto the null space of the Jacobian (Equation (11)).

Figures 9 and 10 show the result of such a process, starting at a mode in Figure 4 that peaks around $v = 0.9$ and $\sigma = -0.7$ and continuing until $\sigma = 0$ is encountered.

Then, starting from the point where this curve meets $\sigma = 0$, a V-ω-η process traces a limit-cycle curve shown in Figure 11.

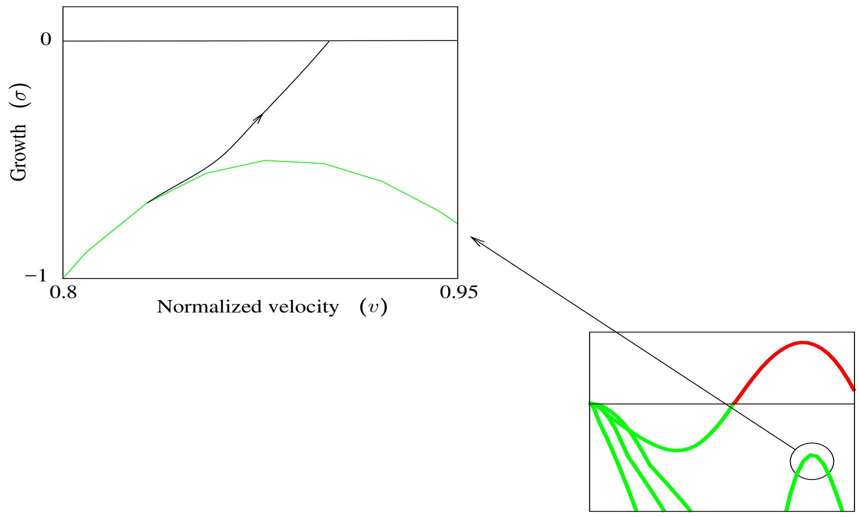

Figure 9. Optimal path starting from V-σ-ω at $\eta = 0$ (black curve with arrow indicating tracking direction) (Green means stable, red means unstable).

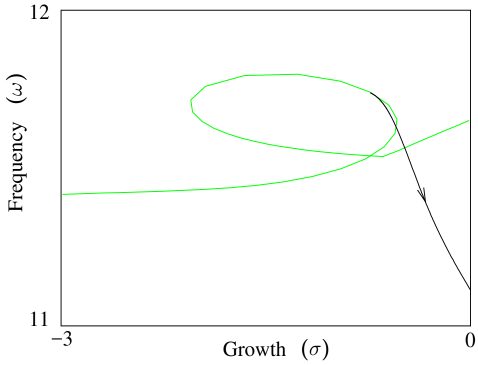

Figure 10. Optimal path: σ-ω, black curve with arrow indicating tracking direction (Green means stable).

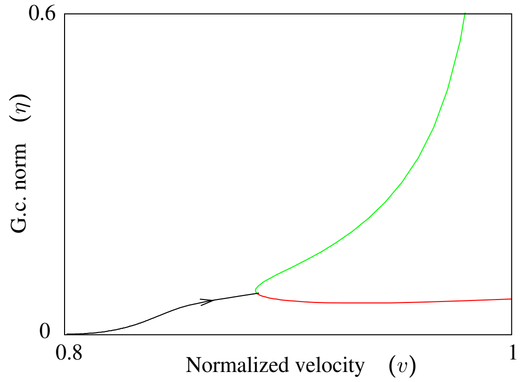

Figure 11. LCO from the optimal path (black curve with arrow indicating tracking direction) (Green means stable, red means unstable).

The complete set of LCO curves, with stable and unstable LCO again shown in green and red, is shown in Figure 12.

Note that at $v = 0.9$ in Figure 12, as η increases from the unstable (static) equilibrium at zero, the next equilibrium encountered is also unstable, followed by two stable equilibriums and finally an unstable LCO. A σ-ω-η process would show continuity in two different modes, a three-Hertz mode and a 10-Hertz mode, and each of these would have the usual pattern of stable followed by unstable and vice versa.

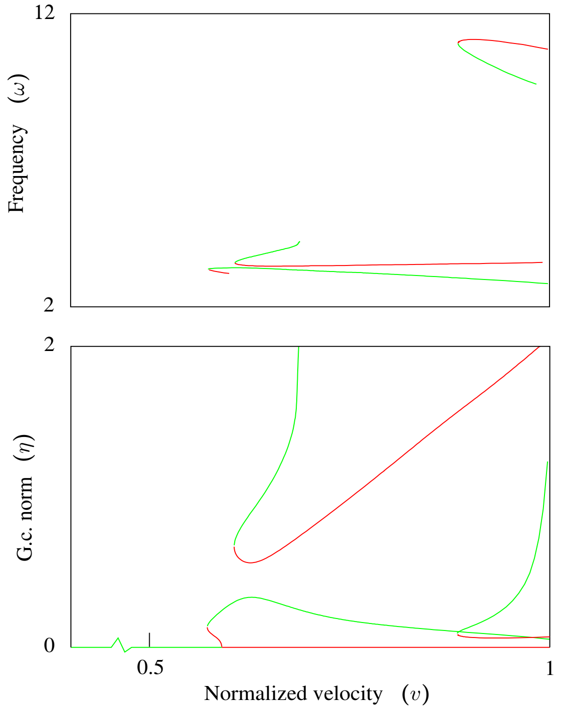

Figure 12. Complete LCO boundaries, v-η and v-ω views (Green means stable, red means unstable).

6. Decreasing LCO Amplitudes

It is often desirable to reduce the amplitude of limit cycles, whether to avoid structural damage or to improve ride quality. Sometimes, this can be done with structural changes, such as reducing free play or increasing damping; in other situations, such changes might be impractical, and it may be necessary to resort to control systems. In either case, the technique of Section 5.8 can be used to decrease the amplitude of a generalized coordinate undergoing LCO.

For example, including a simple control system (matrix $\boldsymbol{T}$ in Equation (13)) can be used to reduce the amplitude of generalized coordinate 1 in the unstable LCO at normalized velocity 0.8 in Figure 12. Details of the control system are unimportant, as it is just for illustration; it simply feeds back displacement in generalized coordinates 1 and 2 to add stiffness to these coordinates with two complex gains: $g_1 e^{i\phi_1}$ and $g_2 e^{i\phi_2}$. The equation to be solved is as before, Equation (23), with $m = 2n_s + 2$, but now with three more unknowns than equations:

$$\boldsymbol{x} = \begin{Bmatrix} g_1 \\ \phi_1 \\ g_2 \\ \phi_2 \\ \omega \\ \boldsymbol{q} \end{Bmatrix} \in \mathbb{R}^{2n_s+5} \tag{37}$$

A projection vector pointing towards decreasing $|\hat{q}_1|$ is:

$$\boldsymbol{w} = -\left\{ \frac{\partial|\hat{q}_1|}{\partial x_i} \right\} = -\begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \boldsymbol{w}_q \end{Bmatrix} \in \mathbb{R}^{2n_s+5} \qquad \boldsymbol{w}_q = \frac{1}{\sqrt{q_1^2 + q_2^2}} \begin{Bmatrix} q_1 \\ q_2 \\ 0 \\ \vdots \end{Bmatrix} \tag{38}$$

Beginning at $v = 0.8$ on the upper curve in Figure 12, where $\eta = 1.18$ and $|\hat{q}_1| = 1.16$, the projection vector guides the solution toward $|\hat{q}_1| = 0$, as shown in Figure 13. The stability of the LCO at each point on the curves is computed as in Section 5.6.

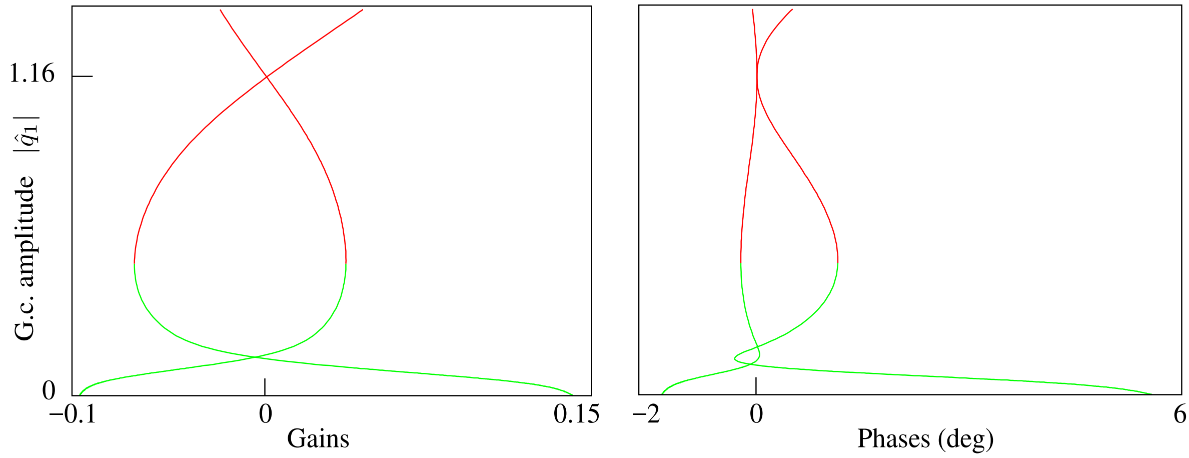

Figure 13. Reduction of an LCO amplitude with control system parameters (Green means stable, red means unstable).

7. Bifurcation

When tracing a curve with a continuation method, it is possible to encounter a bifurcation, where two curves cross [1,17]. Other characteristics of a bifurcation, useful in detecting their presence, are a rank-deficient Jacobian matrix and a change in the sign of the determinant of the Jacobian matrix augmented with the tangent vector. It can be shown [1] that a change in the sign of:

$$\mu(\boldsymbol{x}) = \det \begin{bmatrix} \boldsymbol{J} \\ \boldsymbol{t}^T \end{bmatrix} = \det \begin{bmatrix} \boldsymbol{J}^T & \boldsymbol{t} \end{bmatrix} \tag{39}$$

indicates that a bifurcation has been passed. A method for computing μ is shown in Section 8 that is economical enough that it could (and should) be done at every step.

One of the curves in the V-σ-ω process shown in Figure 6 encounters a bifurcation, although it is not evident at that scale. When plotted σ versus ω with an expanded scale, the bifurcation becomes more evident, as shown in Figure 14. Arrows indicate the direction of the positive determinant. Had the bifurcation not been detected, the process would have tracked only the green curve. Detecting the bifurcation and computing the bifurcation tangent allows the blue curve to be traced.

A brief summary of the steps necessary to trace the bifurcation branch is given here; for more detail, see [1,17].

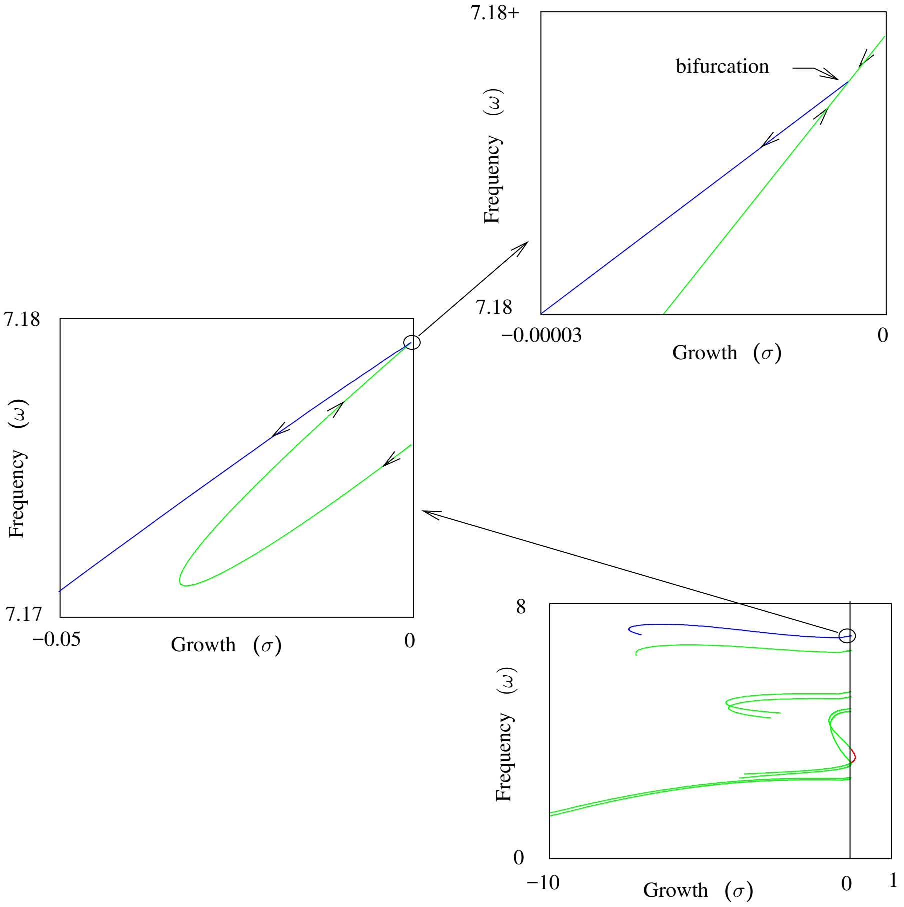

Figure 14. Bifurcation in a V-σ-ω process. Arrows indicate the direction of positive μ, the blue curve arises from the bifurcation (Green means stable, red means unstable).

Branching from the bifurcation requires the tangent to the branch curve. For a simple bifurcation at $\boldsymbol{x}_b$, detected by a change in the sign of $\mu(\boldsymbol{x})$, the Jacobian will have one rank deficiency, and the $\boldsymbol{Q}_2$ matrix has two columns, some combination of which gives the branch tangent; and another combination continues the original curve.

A singular value decomposition (SVD) [7] (p. 76) of the Jacobian at $\boldsymbol{x}_b$,

$$\boldsymbol{J} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T, \quad \boldsymbol{U} \in \mathbb{R}^{m\times m}, \quad \boldsymbol{V} \in \mathbb{R}^{m+1\times m+1} \tag{40}$$

reveals the rank and is necessary for branching from the bifurcation. The last column of $\boldsymbol{U}$ ($\boldsymbol{u}_m$) and the last two columns of $\boldsymbol{V}^T$ ($\boldsymbol{v}_m$ and $\boldsymbol{v}_{m+1}$) are left and right null vectors of the Jacobian, and the two tangents emanating from the bifurcation are combinations of these:

$$\boldsymbol{t}_k = \alpha_k \boldsymbol{v}_m + \beta_k \boldsymbol{v}_{m+1}, \quad k = 1, 2 \tag{41}$$

where:

$$\begin{aligned}
\alpha_k &= \frac{c_k}{\sqrt{c_k^2 + a_{11}^2}} = \frac{a_{22}}{\sqrt{c_k^2 + a_{22}^2}} \\
\beta_k &= \frac{c_k}{\sqrt{c_k^2 + a_{22}^2}} = \frac{a_{11}}{\sqrt{c_k^2 + a_{11}^2}} \\
a_{11} = \frac{\partial^2 g}{\partial \xi_1^2} &\approx \frac{1}{\epsilon^2}\left[g(\epsilon, 0) - 2g(0,0) + g(-\epsilon, 0)\right] \\
a_{12} = \frac{\partial^2 g}{\partial \xi_1 \partial \xi_2} &\approx \frac{1}{4\epsilon^2}\left[g(\epsilon, \epsilon) + g(-\epsilon, -\epsilon) - g(\epsilon, -\epsilon) - g(-\epsilon, \epsilon\right] \\
a_{22} = \frac{\partial^2 g}{\partial \xi_2^2} &\approx \frac{1}{\epsilon^2}\left[g(0, \epsilon) - 2g(0,0) + g(0, -\epsilon)\right] \\
\epsilon &\approx 10^{-6} \\
g(\xi_1, \xi_2) &= \boldsymbol{u}_m^T \boldsymbol{f}(\boldsymbol{x}_b + \xi_1 \boldsymbol{v}_m + \xi_2 \boldsymbol{v}_{m+1})
\end{aligned} \tag{42}$$

One of these tangents is aligned with the original curve, the other with the bifurcation curve. Comparing with the previous tangent ($\boldsymbol{t}_j$) determines which is the primary tangent ($\boldsymbol{t}_c$) and the bifurcation tangent ($\boldsymbol{t}_b$):

$$t_1 = \left|\boldsymbol{t}_j^T \boldsymbol{t}_1\right|, \quad t_2 = \left|\boldsymbol{t}_j^T \boldsymbol{t}_2\right|$$

$$\boldsymbol{t}_c, \boldsymbol{t}_b = \begin{cases} \boldsymbol{t}_1, \boldsymbol{t}_2 & \text{if } t_1 > t_2 \\ \boldsymbol{t}_2, \boldsymbol{t}_1 & \text{otherwise} \end{cases} \tag{43}$$

8. Programming Considerations

Much of the computational effort in the continuation method is in the factorization of the Jacobian matrix in Equation (7), where QR factorizations are used to solve the underdetermined system. Some continuation methods [1,18] add an equation in the corrector phase to constrain corrections, for example to hold one of the variables constant, creating a square set of linear equations that can be solved using an LU factorization instead of QR. Theoretically, LU is faster than QR; but QR is more stable, and modern computer architectures eliminate the speed advantage [7] (p. 299). QR is also necessary for the optimal-path technique of Section 5.8.

The linear algebra library LAPACK [19] provides several variants of QR factorizations; the preferred routine for efficiency and stability is SGEQP3 [16] (Section 2.4.2.3). None of the QR routines in LAPACK return the $\boldsymbol{Q}$ and $\boldsymbol{R}$ matrices directly; instead, all of the information to create them is returned in the input matrix and a separate array named TAU. The $\boldsymbol{R}$ matrix is contained in the upper triangle of the input matrix, and subroutine STRSM solves a set of equations with it. Subroutine SORMQR is provided to multiply $\boldsymbol{Q}$ or $\boldsymbol{Q}^T$ by a vector.

Projecting a vector onto the null space of the Jacobian can be done in LAPACK without actually forming the $\boldsymbol{Q}_2$ matrix with two calls to subroutine SORMQR:

$$\boldsymbol{Q}_2 \boldsymbol{Q}_2^T \boldsymbol{w} = \boldsymbol{Q} \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_2 \end{bmatrix} \boldsymbol{Q}^T \boldsymbol{w} \tag{44}$$

that is, call SORMQR to multiply the vector by $\boldsymbol{Q}^T$, zero the first m elements of the result, then call it again to multiply by $\boldsymbol{Q}$.

Updating the QR factorization to determine stability with Equation (32) can be done efficiently with the qrupdate package available at [20].

Evaluating the determinant in Equation (39) is simplified by recognizing that the tangent came from the QR factorization, that the determinant of an orthogonal matrix is ±1 and that the determinant of a triangular matrix is the product of the diagonals. Furthermore, the tangent and the QR factorization of the Jacobian it was computed from are available from the last corrector iteration, so the desired determinant is:

$$\begin{aligned}
\det\left[\boldsymbol{J}^T \;\; \boldsymbol{t}\right] &= \det\left[\boldsymbol{QR} \;\; \boldsymbol{t}\right] = \det\left[\boldsymbol{Q}_1 \;\; \boldsymbol{Q}_2\right]\begin{bmatrix} \boldsymbol{R} & 0 \\ 0 & l \end{bmatrix} \\
&= l \det \boldsymbol{Q} \det \boldsymbol{R} = (-1)^k l \prod_{i=1}^{m} R_{ii} \qquad (45)
\end{aligned}$$

where k is the number of non-zero elements in the TAU array [21] for LAPACK QR routines, and l is $+1$ if $\boldsymbol{Q}^T\boldsymbol{t}$ is positive, -1 otherwise.

Derivatives are an essential part of continuation methods. Accurate derivatives of parameters and matrices with respect to the independent variables are necessary for computing the Jacobian matrix that is used both to iterate to solutions and predict the next solution. Deriving formulas for complicated expressions like Equation (19) can be tedious and error-prone. A technique that eliminates the need for closed expressions for the derivatives and can reduce coding errors is automatic differentiation [22]. This programming technique gives exact derivatives at a cost on the order of evaluating the function itself, in contrast with difference approximations, which are not exact and much more expensive to compute. The autodiff.org [23] website is devoted to automatic differentiation resources.

9. Conclusions

A continuation method capable of treating frequency domain flutter equations with multiple nonlinearities approximated with describing functions has been developed. With it, a set of continuation processes can find with reasonable certainty all limit-cycle oscillations within a desired flight envelope, tracing curves of limit-cycle amplitude versus velocity and indicating stability with colors.

With a small modification to the continuation method, curves can be traced taking an optimal path with respect to any number of parameters toward some goal. This technique was used to seek out a limit cycle that was missed and to reduce limit-cycle amplitudes by adjusting control-system parameters.

Tracing curves with continuation methods, it is always possible to encounter bifurcation. Detecting bifurcation is inexpensive enough that it can be done at every step when tracing a curve; missing a bifurcation risks missing an important curve. When a bifurcation is detected, a straightforward calculation gives the two tangents, which are then used to start continuation processes to trace each curve from the bifurcation.

Conflicts of Interest: The author declares no conflict of interest.

Nomenclature

a	sonic velocity
$\hat{\boldsymbol{D}}, \boldsymbol{D}$	complex and equivalent real dynamic matrices (Equations (13) and (16))
d	structural damping coefficient
$\boldsymbol{f}$	vector of n real, nonlinear equations
$\boldsymbol{J}$	Jacobian matrix of partial derivatives (Equation (5))
$\boldsymbol{J}_{i:}, \boldsymbol{J}_{:j}$	row i and column j of matrix $\boldsymbol{J}$
$\boldsymbol{M}, \boldsymbol{K}, \boldsymbol{G},$ $\boldsymbol{V}, \boldsymbol{A}, \boldsymbol{T}$	mass, stiffness, gyroscopic, viscous damping, unsteady aero and control-system matrices (Equation (13))

M	V/a, Mach number
n_s	the number of generalized coordinates
p	s/V, complex reduced frequency
$\\|\boldsymbol{x}\\|_2$	2-norm of vector $\boldsymbol{x} = \sqrt{\boldsymbol{x}^T\boldsymbol{x}}$
q	$\rho V^2/2$, dynamic pressure
$\hat{\boldsymbol{q}}, \boldsymbol{q}$	complex and equivalent real generalized coordinates (g.c.) (Equations (12) and (16))
$\boldsymbol{Q}, \boldsymbol{R}$	QR factors of the transposed Jacobian (Equation (7))
$\mathbb{R}^n, \mathbb{R}^{m\times n}$	the real n-vectors and m by n matrices
$\Re(), \Im()$	real and imaginary parts of a complex variable
r, γ, δ	bilinear stiffness describing-function variables (Equation (19))
s	Laplace variable $\sigma + i\omega$
$\boldsymbol{t}$	tangent vector
V	velocity (true airspeed)
V_{max}	maximum velocity of interest (true airspeed)
v	normalized velocity V/V_{max}
$\boldsymbol{w}$	projection vector (Equation (11))
$\boldsymbol{x}$	independent variables
$\boldsymbol{x}_j^i$	ith iteration at the jth continuation step (Equation (2))
x_i	ith element of vector $\boldsymbol{x}$
$\hat{\boldsymbol{y}}, \boldsymbol{y}$	complex and equivalent real eigenvector
$\boldsymbol{z}(t)$	motion of points of the structure (Equation (12))
η	2-norm of the generalized coordinates (Equation (15))
ω	oscillation frequency (imaginary part of the Laplace variable)
$\boldsymbol{\Phi}$	transformation from generalized to physical coordinates
ρ	fluid density
σ	growth factor (real part of the Laplace variable)
τ	arc length along a curve
$()^T$	transpose

References

1. Allgower, E.L.; Georg, K. *Numerical Continuation Methods: An Introduction*; Springer Series in Computational Mathematics; Springer: Berlin, Germany, 1990; Volume 13.
2. Meyer, E.E. Unified Approach to Flutter Equations. *AIAA J.* **2014**, *52*, 627–633.
3. Shen, S.F. An Approximate Analysis of Nonlinear Flutter Problems. *J. Aerosp. Sci.* **1959**, *26*, 25–32.
4. Gordon, J.T.; Meyer, E.E.; Minogue, R.L. Nonlinear Stability Analysis of Control Surface Flutter with Freeplay Effects. *J. Aircr.* **2008**, *45*, 1904–1916.
5. Lee, C.L. An Iterative Procedure for Nonlinear Flutter Analysis. *AIAA J.* **1986**, *24*, 833–840.
6. Breitbach, E.J. *Flutter Analysis of an Airplane with Multiple Structural Nonlinearities in the Control System*; NASA Langley Research Center: Hampton, VA, USA, 1980.
7. Golub, G.H.; Van Loan, C.F. *Matrix Computations*, 4th ed.; The Johns Hopkins University Press: Baltimore, MD, USA, 2013.
8. Gelb, A.; Vander Velde, W.E. *Multiple-Input Describing Functions and Nonlinear System Design*; McGraw-Hill: New York, NY, USA, 1968.
9. Hassig, H.J. An Approximate True Damping Solution of the Flutter Equation by Determinant Iteration. *J. Aircr.* **1971**, *8*, 885–889.
10. Chen, P.C. Damping Perturbation Method for Flutter Solution: The g-Method. *AIAA J.* **2000**, *38*, 1519–1524.
11. Van Zyl, L.H.; Maserumule, M.S. Divergence and the p-k Flutter Equation. *J. Aircr.* **2001**, *38*, 584–586.
12. Edwards, J.W.; Wieseman, C.D. Flutter and Divergence Analysis Using the Generalized Aeroelastic Analysis Method. *J. Aircr.* **2008**, *45*, 906–915.
13. Rodden, W.P. *MSC/NASTRAN Handbook for Aeroelastic Analysis*; The MacNeal-Schwendler Corporation: Santa Ana, CA, USA, 2009; pp. 357–372.
14. Bisplinghoff, R.L.; Ashley, H.; Halfman, R.L. *Aeroelasticity*; Addison-Wesley: Reading, MA, USA, 1955.
15. Ruhe, A. Algorithms for the Nonlinear Eigenvalue Problem. *SIAM J. Numer. Anal.* **1973**, *10*, 674–689.

16. Anderson, E.; Bai, Z.; Bischof, C.; Blackford, S.; Dongarra, J.; Du Croz, J.; Greenbaum, A.; Hammarling, S.; McKenney, A.; Sorensen, D. *LAPACK Users' Guide*, 3rd ed.; Siam: Philadelphia, PA, USA, 1999.
17. Meyer, E.E. Continuation and Bifurcation in Linear Flutter Equations. *AIAA J.* **2015**, *53*, 3113–3116.
18. Rheinboldt, W.C.; Burkardt, J.V. A Locally Parameterized Continuation Process. *ACM Trans. Math. Softw.* **1983**, *9*, 215–235.
19. LAPACK—Linear Algebra PACKage. Available online: https://www.netlib.org/lapack (accessed on 18 June 2016).
20. Hajek, J. Qrupdate. Available online: https://sourceforge.net/projects/qrupdate/ (accessed on 2 September 2016).
21. Langou, J. Determinant in QR factorization. Available online: http://icl.cs.utk.edu/lapack-forum/viewtopic.php?f=2&t=1741 (accessed on 14 May 2011).
22. Griewank, A.; Walther, A. *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*, 2nd ed.; Siam: Philadelphia, PA, USA, 2008.
23. Community Portal for Automatic Differentiation. Available online: http://www.autodiff.org (accessed on 2 September 2016).