

Bifurcation Analysis Using Rigorous Branch and Bound Methods

Andrew P. Smith and Luis G. Crespo
National Institute of Aerospace
Hampton, Virginia 23666
Email: andrew.smith@nianet.org
luis.g.crespo@nasa.gov

César A. Muñoz
NASA Langley Research Center
Hampton, Virginia 23681
Email: cesar.a.munoz@nasa.gov

Mark H. Lowenberg
Department of Aerospace Engineering
University of Bristol
Bristol BS8 1TR, UK
Email: m.lowenberg@bristol.ac.uk

Abstract—For the study of nonlinear dynamic systems, it is important to locate the equilibria and bifurcations occurring within a specified computational domain. This paper proposes a new approach for solving these problems and compares it to the numerical continuation method. The new approach is based upon branch and bound and utilizes rigorous enclosure techniques to yield outer bounding sets of both the equilibrium and local bifurcation manifolds. These sets, which comprise the union of hyper-rectangles, can be made to be as tight as desired. Sufficient conditions for the existence of equilibrium and bifurcation points taking the form of algebraic inequality constraints in the state-parameter space are used to calculate their enclosures directly. The enclosures for the bifurcation sets can be computed independently of the equilibrium manifold, and are guaranteed to contain all solutions within the computational domain. A further advantage of this method is the ability to compute a near-maximally sized hyper-rectangle of high dimension centered at a fixed parameter-state point whose elements are guaranteed to exclude all bifurcation points. This hyper-rectangle, which requires a global description of the bifurcation manifold within the computational domain, cannot be obtained otherwise. A test case, based on the dynamics of a UAV subject to uncertain center of gravity location, is used to illustrate the efficacy of the method by comparing it with numerical continuation and to evaluate its computational complexity.

I. INTRODUCTION

A state-space representation of an autonomous dynamic system is given by the set of ordinary differential equations

$$\begin{aligned} \dot{x}_1 &= f_1(x_1, \dots, x_{n_x}, p_1, \dots, p_{n_p}), \\ &\vdots \\ \dot{x}_{n_x} &= f_{n_x}(x_1, \dots, x_{n_x}, p_1, \dots, p_{n_p}), \end{aligned} \quad (1)$$

where the dot notation indicates differentiation with respect to time, t , $x \in \mathbb{R}^{n_x}$ is the state vector, $p \in \mathbb{R}^{n_p}$ is the model's parameter vector, and $f_i, i = 1, \dots, n_x$, are in general nonlinear functions of the indicated arguments.

The equilibria of the system are given by the state-parameter combinations for which $\dot{x}_1, \dots, \dot{x}_{n_x} = 0$. These pairs will be denoted as (\hat{x}, \hat{p}) . Hence, for a fixed value of p , the equilibria are given by the solution of a system of n_x equations in n_x unknowns. A zero of this system of equations is called an equilibrium point or fixed point of the dynamic system. The vector field associated with (1) in the vicinity of an equilibrium point determines if such a point is locally stable or unstable

[1]. The locus of the equilibrium points is parameterized by the value of p . Variation in p may change the location, number, and local stability of equilibrium points. A local bifurcation occurs when the number of equilibrium points or their local stability changes due parameter variations. The range of these variations, which prescribe the computational domain of interest, will be given by $x_i \in [\underline{x}_i, \bar{x}_i] \subset \mathbb{R}$, $i = 1, \dots, n_x$, and $p_j \in [\underline{p}_j, \bar{p}_j] \subset \mathbb{R}$, $j = 1, \dots, n_p$.

Bifurcation analysis is a popular tool for researching the behaviour of nonlinear dynamic systems, including aircraft flight dynamics. In this particular application, bifurcation analysis has been used to study the flight upset tendencies of aircraft [2], the resilience of control laws to faults, time delays and uncertainties, and to evaluate the effectiveness and robustness of control laws as individual gains are varied.

Conventional bifurcation analysis typically requires computation of the equilibrium manifold within the computational domain. In the numerical continuation method [3], state trajectories starting from a grid of initial conditions are computed via integration. These trajectories are used to identify a stable equilibrium point on the equilibrium manifold. Numerical continuation methods reconstruct the equilibrium manifold by performing a search that starts in this state-parameter point. This search yields points on the equilibrium manifold along with the corresponding local stability analysis and bifurcation points. Numerical continuation methods do not require an analytical representation of (1) and yield both local and global bifurcation analyses. However, they are usually restricted to low-dimensional parameters, e.g., $n_p \leq 2$, and yield results that might be incomplete, e.g., they might miss unconnected branches of the equilibrium manifold and their corresponding bifurcation points.

This paper proposes a new approach for outer bounding the equilibrium and local bifurcation manifolds of (1) within a fixed computational domain. The approach is based upon branch and bound and utilizes rigorous enclosure techniques based on interval arithmetic and the Bernstein polynomial basis. While the method is restricted to cases in which (1) is available analytically, it renders solutions that are both correct and complete, i.e., it will not miss any point of the equilibrium and bifurcation manifolds within the computational domain.

The organization of the paper is as follows: A set of

sufficient conditions for identifying and classifying local bifurcation points is presented in Section II. Enclosure methods are introduced in Section III and theory and methodology of the branch and bound solver is described in Section IV. The definition and formulation required to calculate exclusion sets are provided in Section V. In Section VI the main example is presented and used to illustrate the performance of the method and compare it to numerical continuation. The paper concludes with directions for future work.

II. CATEGORIES OF BIFURCATION

A dynamic system may exhibit both global and local bifurcations; this paper focuses on the identification and classification of local bifurcations of (1). These bifurcations, which require finding the fixed points of the system, linearizing it at such points, and performing an eigenvalue analysis, lend themselves more naturally to a branch and bound approach.

In non-degenerate cases, the equilibrium and bifurcation manifolds of the dynamic system (1) have n_p and $n_p - 1$ degrees of freedom, respectively. For instance, when $n_p = 1$, a few isolated bifurcation points may exist; when $n_p = 2$, there may be bifurcation line segments; when $n_p = 3$, there may be bifurcation surfaces, and so on. In a practical bifurcation analysis, only a small number of parameters are allowed to vary simultaneously, since the complexity of characterizing the bifurcation set increases exponentially with n_p . The types of local bifurcations considered here are classified as follows:

A. Steady-State Bifurcations

Consider the order n_x Jacobian matrix $J_f(\hat{x}, \hat{p})$ of $f = (f_1, \dots, f_{n_x})$ in the system (1). A steady-state bifurcation arises at (\hat{x}, \hat{p}) wherever $J_f(\hat{p})$ is singular, i.e., it has a zero eigenvalue. This category includes pitchfork and saddle-node bifurcations [1].

B. Hopf Bifurcations

A Hopf bifurcation at (\hat{x}, \hat{p}) occurs when the characteristic polynomial, $\det(J_f(\hat{x}, \hat{p}) - zI) = 0$, has a conjugate pair of of complex solutions with zero real part and all other roots have a negative real part. Each such bifurcation originates a limit cycle. Let the characteristic polynomial be given by

$$q(z) = a_0 z^{n_x} + a_1 z^{n_x-1} + \dots + a_{n_x}, \quad (2)$$

where the coefficients are functions of (\hat{x}, \hat{p}) . The associated Hurwitz matrix, which is order n_x and square, is given by

$$H = \begin{pmatrix} a_1 & a_3 & a_5 & \cdots & \cdots \\ a_0 & a_2 & a_4 & \cdots & \cdots \\ 0 & a_1 & a_3 & a_5 & \cdots \\ 0 & a_0 & a_2 & a_4 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}. \quad (3)$$

The i th Hurwitz determinant, Δ_i , is equal to the determinant of the i th principal minor of H .

Theorem 1 (Routh-Hurwitz, cf. [4]): All the roots of q are in the left-hand half-plane if and only if

$$\Delta_{n_x}(p) > 0, \quad \dots, \quad \Delta_1(p) > 0. \quad (4)$$

Theorem 2 (cf. [4]): Let $a_0 > 0$. Then q has a pair of distinct roots, $i\omega$ and $-i\omega$, on the imaginary axis and all other roots are in the left-hand half-plane if and only if $a_{n_x}(p) > 0$ and

$$\Delta_{n_x-1}(p) = 0, \quad \Delta_{n_x-2}(p) > 0, \quad \dots, \quad \Delta_1(p) > 0. \quad (5)$$

The criteria of Theorem 2 may be used to detect Hopf bifurcations.

III. ENCLOSURE METHODS

In order to determine the behavior of a nonlinear function $f_i : \mathbb{R}^{n_x} \rightarrow \mathbb{R}$ over interval ranges for variables and parameters, techniques are required that, given such a function, enable the construction of an interval-valued function $F_i : \mathbb{I}\mathbb{R}^{n_x} \rightarrow \mathbb{I}\mathbb{R}$, where $\mathbb{I}\mathbb{R}$ denotes the set of closed non-empty real intervals, such that $\forall \mathbf{X} \in \mathbb{I}\mathbb{R}^{n_x}, x \in \mathbf{X} \implies f_i(x) \in F_i(\mathbf{X})$.

A. Interval Arithmetic

A natural interval extension for a real-valued function may be obtained by taking a particular symbolic representation and replacing each of the component operators and functions (chiefly elementary operations and logarithmic and trigonometric functions) by their interval equivalents. Basic definitions for such interval operators are given in terms of the interval endpoints, for example, if $\mathbf{a} = [\underline{a}, \bar{a}]$, $\mathbf{b} = [\underline{b}, \bar{b}] \in \mathbb{I}\mathbb{R}$, then $\mathbf{a} + \mathbf{b} = [\underline{a} + \underline{b}, \bar{a} + \bar{b}]$.

These operational definitions obey inclusion isotonicity, i.e., if $\mathbf{a}_1, \mathbf{b}_1 \in \mathbb{I}\mathbb{R}$ with $\mathbf{a}_1 \subseteq \mathbf{a}$ and $\mathbf{b}_1 \subseteq \mathbf{b}$, then $\mathbf{a}_1 \circ \mathbf{b}_1 \subseteq \mathbf{a} \circ \mathbf{b}$, if $\mathbf{a}_1 \circ \mathbf{b}_1$ is defined. However, some relations known to be true in \mathbb{R} are not valid in $\mathbb{I}\mathbb{R}$, e.g., the distributive law. Instead there is the weaker subdistributive law

$$\mathbf{a} \cdot (\mathbf{b} + \mathbf{c}) \subseteq \mathbf{a}\mathbf{b} + \mathbf{a}\mathbf{c} \text{ for } \mathbf{a}, \mathbf{b}, \mathbf{c} \in \mathbb{I}\mathbb{R}.$$

Interval extensions for non-trivial functions can therefore exhibit a significant amount of excess width, which is caused by each separate occurrence of the same variable being treated as if it were an independent variable. This phenomenon is known as the dependency problem. As a result, the performance of an interval algorithm is characterized not just by its speed and correctness, but also the quality of the result in terms of minimizing such excess width. A straightforward adaption of an algorithm designed for floating-point arithmetic is therefore usually unsuitable, and customized algorithms may be used instead.

Numerous software implementations of interval arithmetic exist for C++, MATLAB, and for other programming environments. In this work, the C++ library *filib++* [5] is used, which utilizes floating point rounding modes to deliver rigorous interval results. Further operator definitions and an introduction to interval arithmetic are given in [6].

B. Bernstein Expansion

High-quality interval enclosures of multivariate polynomial functions can be computed by rewriting the polynomial in terms of the Bernstein basis with respect to a particular set of variable ranges, instead of in the usual power form. The

interval hull of the coefficients of such an expansion, the so-called Bernstein coefficients, provide an enclosure which is in general tighter than that provided by interval arithmetic, and exhibits superior convergence. A detailed description and an efficient representation scheme for sparse polynomials may be found in [7]. A formally-verified treatment of Bernstein polynomials is given in [8]. Bernstein expansion has previously been applied to the stability and bifurcation analysis of bivariate polynomial dynamic systems [9].

IV. A GENERIC BRANCH AND BOUND SOLVER

Branch and bound is a computation scheme for problems commonly specified over a domain consisting of a Cartesian product of intervals, i.e., a hyper-rectangle, referred to here as a *box*. This starting domain is recursively subdivided into sub-boxes, typically by performing a bisection of one of the component intervals. Over each box, rigorous enclosures for the range of a real-valued function may be computed by employing a suitable enclosure method. Whereas sub-boxes that are proven not to contain a solution to the problem are discarded, those that might contain a solution are subdivided further, resulting in tighter function enclosures.

The solution to many problems can be given as a guaranteed outer approximation of the exact solution set within the computational starting domain. Here, a *paving* for a solution consists of the union of not necessarily disjoint boxes, with each box possibly containing part of the exact solution. This collection of boxes can be progressively refined until a terminally small size for a sub-box is reached.

The tool used in this work, *Kodiak*, is a software package in C++ which facilitates formally-verified branch and bound computation. It allows for the implementation of several types of branch and bound algorithm using generic routines. Constraints may be formulated as Boolean expressions of predicates involving one or more nonlinear functions, which are represented symbolically, and relational operators. There are specific instantiations of the generic algorithm for optimization problems, systems of nonlinear equations, and Boolean problems. The two main enclosure methods currently available for nonlinear functions are interval arithmetic and Bernstein expansion. Input functions are currently written in software and can be symbolically manipulated, e.g., symbolic partial differentiation can be performed.

A formal verification of a related algorithm was presented in [10]. A similar algorithm for systems of polynomial equations, utilizing the Bernstein expansion, was given in [11].

A. Systems of Nonlinear Equations (Equilibrium Sets)

One of the main instantiations of the generic branch and bound algorithm is an algorithm for paving the equilibrium manifold of (1), given by all the pairs (\hat{x}, \hat{p}) . Following the notation of (1), let n_x real-valued functions f_i , $i = 1, \dots, n_x$, in the variables x_1, \dots, x_{n_x} and parameters p_1, \dots, p_{n_p} , and a box $\mathbf{Z} := [\underline{x}_1, \bar{x}_1] \times \dots \times [\underline{x}_{n_x}, \bar{x}_{n_x}] \times [\underline{p}_1, \bar{p}_1] \times \dots \times [\underline{p}_{n_p}, \bar{p}_{n_p}]$ in $\mathbb{IR}^{n_x+n_p}$ be given. In general, the solution set $S := \{x \in \mathbb{R}^{n_x}, p \in \mathbb{R}^{n_p} : f_i(x, p) = 0, i = 1, \dots, n_x\}$ cannot be

described algebraically. Instead a paving S^* , a union of boxes with $S^* \supseteq S \cap \mathbf{Z}$, is computed.

In non-degenerate cases for which p is fixed ($n_p = 0$), zero or more point solutions may exist, and the paving consists of one or more boxes of terminal width enclosing each individual solution. For underdetermined systems where $n_p > 0$, the paving consists of many boxes; those intersecting the boundary of S should be of terminal width. The interpretation of the paving is that it is proven that no solutions in $\mathbf{Z} \setminus S^*$ can exist, but only that solutions in S^* possibly exist — the algorithm is sound with respect to the exclusion of solutions, but complete with respect to their inclusion.

In outline, the processing of a sub-box \mathbf{Z}^* deriving from \mathbf{Z} proceeds as follows:

- 1) Compute enclosures for ranges of each f_i over \mathbf{Z}^* , using the methods in Section III. As soon as an i is found for which $0 \notin f_i(\mathbf{Z}^*)$, one can be sure that the box does not admit a solution, and discard it. Otherwise, proceed:
- 2) If \mathbf{Z}^* is of terminal width, place it into the final paving S^* . Otherwise, proceed:
- 3) Choose a variable or parameter in which to perform a subdivision. Various heuristics are available, e.g. round-robin, widest interval, use of partial derivatives.
- 4) Bisect (branch) \mathbf{Z}^* into two sub-boxes, by bisecting the corresponding variable or parameter interval, and recurse.

This algorithm can be used to compute a paving for the equilibrium set of (1).

B. Bifurcation Sets

The above algorithm can be extended to compute a paving for the bifurcation set (steady-state and Hopf bifurcations) of (1) directly, without first needing to determine the equilibrium set. To this end, proceed as follows. First, the Jacobian, the coefficients of the characteristic polynomial (2), and the determinants of the Hurwitz matrix (3) are computed symbolically. The substitution of these expressions into the sufficient conditions for bifurcation yields a set of inequality constraints. Recall that these constraints are $a_{n_x}(\hat{x}, \hat{p}) = 0$ for a steady-state bifurcation, and $a_{n_x}(\hat{x}, \hat{p}) > 0$ and (5) for a Hopf bifurcation. Note that the exact state-parameter points at which the constraints are to be evaluated are unknown (i.e., we will only have their enclosure).

A system of equations consisting of the equality conditions for equilibrium and the extra constraints representing each type of bifurcation can then be solved. A box may either not contain either type of bifurcation point (in which case it is discarded), it may possibly contain only a steady-state bifurcation, it may possibly contain only a Hopf bifurcation, or it may possibly contain both.

For non-degenerate problems, the resultant paving for the bifurcation set is a subset of the paving for the equilibrium set. The bifurcation set itself has one fewer degree of freedom.

The paving of the solution set will be given as a union of boxes. Membership in the corresponding outer bounding set can be determined by checking whether any given point is a

member of any of the sub-boxes comprising the paving. The separation between any given state-parameter point and the bifurcation manifold can be safely underestimated by finding the minimum distance between the point and the center of all sub-boxes after half of each sub-box's diagonal is subtracted.

V. GUARANTEED EXCLUSION BOXES

The algorithm for paving bifurcation sets also enables the calculation of a box in \mathbf{Z} of near-maximal size not containing any bifurcation point. Denote by $z_0 = [x, p] \in \mathbb{R}^{n_x+n_p}$ an arbitrary point in the state-parameter domain \mathbf{Z} outside the bifurcation manifold B . The m -scaled infinity norm is instrumental for calculating the desired box. For a vector $m \in \mathbb{R}^{n_x+n_p}$ with positive components, the m -scaled infinity norm of $a \in \mathbb{R}^{n_x+n_p}$ is defined as $\|a\|_m^\infty = \max_k \{|a_k|/m_k\}$. The vector m , called the *aspect vector*, is set to half of the positive diagonal of \mathbf{Z} . The desired box is given by $\mathbf{Z}^\dagger(z_0, m, r) = \{z : z_0 - rm \leq z \leq z_0 + rm\}$, where the inequalities apply componentwise and the worst-case perturbation radius is specified as $r = \min_z \{\|z - z_0\|_m^\infty : z \in B \cap \mathbf{Z}\}$.

A value of z at which the minimum of r occurs, \tilde{z} , is the worst-case state-parameter combination associated with z_0 and m . Instead of trying to solve the optimization problem for r , the bifurcation set paving algorithm is used to generate a sequence of increasingly-close inner bounding boxes for \mathbf{Z}^\dagger , as well as a paving for \tilde{z} . These bounding boxes are centered on z_0 , which is chosen to be a point on the equilibrium manifold S . The radius r measures the separation between z_0 (which could be chosen to build a linear approximation to (1) for control design purposes), and the region of \mathbf{Z} where the open-loop dynamics are qualitatively different from those at z_0 (thus, where a controller designed for the linear approximation will not perform well). Furthermore, the computational effort required to generate a close inner bounding box for \mathbf{Z}^\dagger is typically much lower than that for generating a paving for B . This enables the consideration of dynamic systems for which $n_x + n_p$ exceeds the feasible limit for the computation of a whole paving.

VI. EXAMPLE: NASA GTM LONGITUDINAL DYNAMICS

Bifurcation analysis is a popular tool for researching the behavior of nonlinear dynamic systems, including aircraft flight dynamics. The example below studies the dynamics of the Generic Transport Model (GTM). The GTM is a mathematical representation of the AirStar vehicle, a 5.5% dynamically scaled remotely-operated twin-engine jet airliner developed by the NASA Langley Research Center. A high-fidelity model of the vehicle, using nonlinear aerodynamic models extracted from wind tunnel and system identification experiments for conditions extending beyond the normal flight envelope, is available. This example problem focuses on the dependence of longitudinal dynamics on variations in the location of the aircraft's center of gravity (CG). Such variations might, in general, result from fuel consumption, structural damage, load shifts, and uncertainty.

A. Mathematical Model

The open-loop aircraft dynamics is given by

$$\begin{aligned}\dot{\theta} &= q, \\ \dot{\alpha} &= q + \frac{g \cos(\theta - \alpha)}{V} - \frac{L + T_x \sin(\alpha) - T_z \cos(\alpha)}{(m_1 + m_2)V}, \\ \dot{V} &= -g \sin(\theta - \alpha) + \frac{T_x \cos(\alpha) + T_z \sin(\alpha) - D}{m_1 + m_2}, \\ \dot{q} &= [M + T_m - dx(T_z + D \sin(\alpha) + L \cos(\alpha)) - \\ &\quad dz(T_x - D \cos(\alpha) + L \sin(\alpha))] / \gamma,\end{aligned}$$

where $\gamma = 0.2564(m_1 + m_2) + m_1 m_2(dx^2 + dz^2)/(m_1 + m_2)$. The state x comprises α (attack angle), V (velocity), θ (pitch angle), and q (pitch rate), while the parameter p contains δ_e (elevator deflection), δ_{th} (throttle input), m_2 (supplemental mass), dx (offset in the location of the CG relative to a nominal location in the longitudinal axis), and dz (offset in the location of the CG relative to a nominal location in the z -direction in body axes). The lift, drag, and pitching moments are given by $L = \rho V^2 S C_L(\delta_e, \alpha, q)/2$, $D = \rho V^2 S C_D(\delta_e, \alpha, q)/2$, and $M = \rho V^2 S \bar{c} C_m(\delta_e, \alpha, q)/2$, where \bar{c} is the mean aerodynamic chord, ρ is the air density, S is the planform area, and the non-dimensional terms C_L , C_D , and C_m are polynomial functions of their arguments given in [12]. Likewise, the thrust components T_x , T_z , and T_m are given by cubic polynomial functions of δ_{th} . The values of the parameters not specified in this paper are available in [12].

The fixed mass of the aircraft, m_1 , is 18 kg; the supplemental mass m_2 might vary. The nominal values of the parameters are $\bar{\delta}_e = 5.368$ deg (i.e., elevator deflection required for horizontal flight at maximum throttle input), $\bar{\delta}_{th} = 1$, $\bar{m}_2 = 5$ kg, $\bar{dx} = 0$, $\bar{dz} = 0$. The ranges of x and p used in the analyses that follow are $\alpha \in [-11.5, 103]$ deg, $\theta \in [-86, 86]$ deg, $V \in [1, 200]$ m/s, $q = 0$ deg/s, $\delta_e \in [-10, 20]$ deg, $\delta_{th} \in [0, 1]$, $m_2 \in [0, 5]$ kg, $dx, dz \in [-0.15, 0.2]$ m.

B. Example Problems

Parameters that are not allowed to vary in any particular problem take on their nominal values. Numerical continuation was performed using the Dynamical Systems Toolbox [13], which implements the Fortran AUTO code [14] in the MATLAB environment. A 2.4 GHz Intel i5 PC with 4 Gb of RAM was used. The times given are for computation alone, and do not reflect the additional modeling effort and time required of the user, in order to generate a suitable set of starting solutions. Even an expert may easily miss some solutions that are necessary in order not to miss any branches. The branch and bound pavings were computed using *Kodiak* on a 3 GHz Intel Xeon PC with 4 Gb of RAM.

1) *Single-parameter varying, 5D paving*: In the first analysis, only dx is allowed to vary. In this setting, pavings for the equilibrium surface and bifurcation points are computed, and are compared with the solution obtained via numerical continuation. The graphs of the 2D projections of the continuation solution and pavings are presented in Figure 1; the numbers of boxes in each paving category are listed in the key.

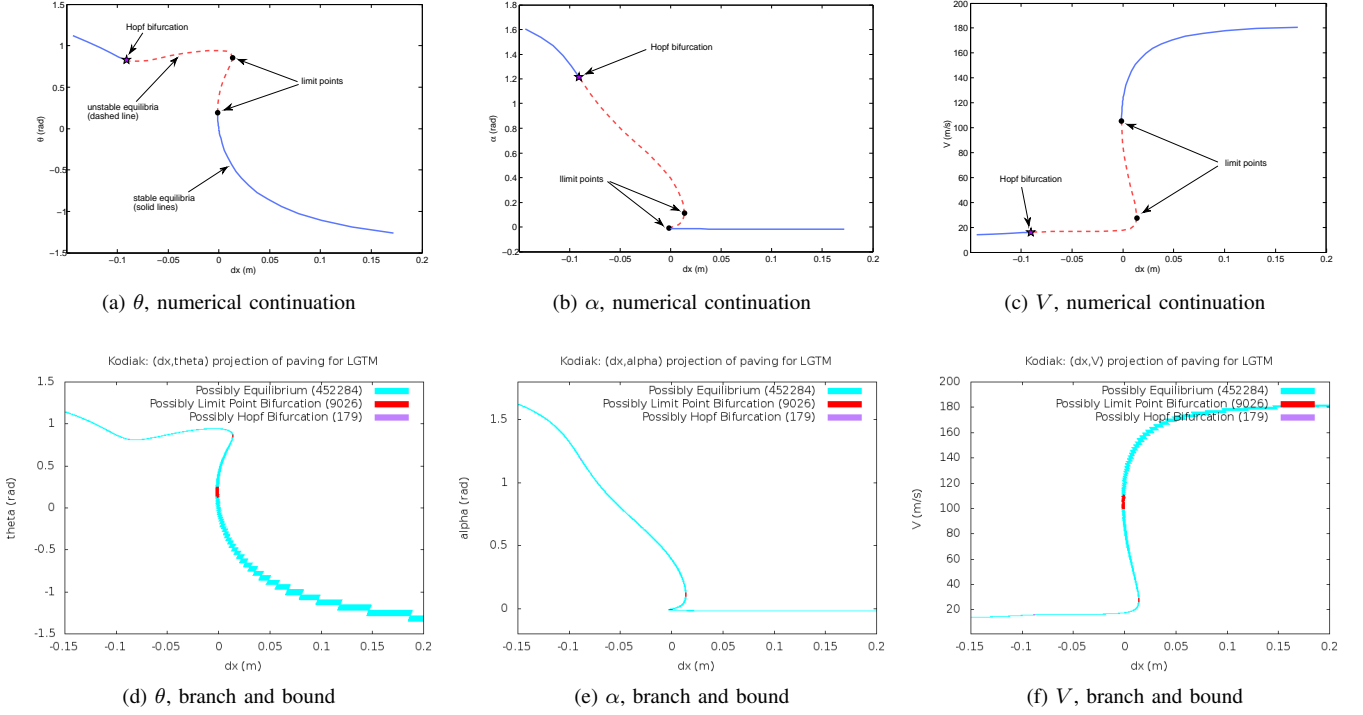


Fig. 1. Single-parameter varying: results of numerical continuation compared with paving projections with respect to each variable and dx .

The agreement in the results of both methods is apparent. The numerical continuation solution was computed in 0.4 s. At the depicted resolution, the paving for the equilibrium set required 4,700,699 bisections and took 44 min 45 s. The bifurcation pavings, however, being much smaller, only required 139,793 bisections and took 1 min 49 s. It is seen that the system remains stable for all positive values of dx , i.e., where the CG is ahead of its nominal position. This is expected, as static stability varies with CG variation, forward positions increasing the stability margin and aft positions reducing it. Here, stability is seen to be marginal at the nominal CG position: a very small negative value of dx induces a fold bifurcation (at $V \approx 105$ m/s) and stability is lost. It is evident, therefore, that high thrust values which induce nose-up moments act to reduce static stability (pitch stiffness). The equilibria remain unstable until the Hopf bifurcation at $dx \approx -0.091$ m. Whilst solutions are stable again at more negative dx values, they represent a very deep stall condition at extremely high angle of attack. In practice, where lateral-directional effects are also modeled, it is likely that these equilibria may translate into a spin condition. Note that the polynomial approximations to the aerodynamic data were not weighted for accuracy at such high angles of attack, so that the model realism is degraded here.

2) *Two-parameter varying, 6D paving:* Now both δ_e and dx are permitted to vary simultaneously. Some of the graphs of the 2D projections of the continuation solution and pavings for the bifurcation surfaces are presented in Figure 2. The loci of limit point and Hopf bifurcations in the parameter space delimit regions of stable and unstable behavior for the

system. For both pavings, 7,000,167 bisections were required and the computation took 116 min 46 s. In the parameter space, two separate branches of each bifurcation type can clearly be identified. The first runs with numerical continuation produced three of these branches after a total computation time of 15.8 s. The missing branch (the Hopf branch to the right) was only identified after checking with the paving; the extra runs took an additional 1.3 s. The paving for the left-most limit point branch where $\delta_e \approx 0.1$ rad terminates where the corresponding value of θ moves outside the search domain.

3) *Five-parameter varying, 9D paving: guaranteed exclusion box:* It is desired to compute the near-maximal state-parameter box of fixed proportions (set to be the same as the overall state-parameter domain) centered on a point in the equilibrium manifold not containing any bifurcation of the considered types. Here $\bar{\delta}_{th} = 0.2$ is taken, for which $\bar{\delta}_e = 3.561$ deg, and the other nominal parameter values are unchanged. These values correspond to horizontal flight. This problem, in which the bifurcation volume is four-dimensional, makes numerical continuation methods inapplicable.

The value of r is searched for by considering progressively larger boxes until the paving of the bifurcation manifold is no longer empty. Recall that obtaining a non-empty paving does not guarantee the existence of a bifurcation point. Empty pavings with $r = 0.06$ (after 3,195 bisections and 3.2 s) and $r = 0.12$ (after 415,293 bisections and 7 min 43 s) were found. The latter proves that no bifurcations of the considered types can exist in the box given by ranges of x and p as follows: $\alpha \in [-5.09, 8.66]$ deg, $\theta \in [-8.52, 12.10]$ deg, $V \in [40.40, 64.27]$ m/s, $q = 0$ deg/s, $\delta_e \in [1.77, 5.36]$ deg,

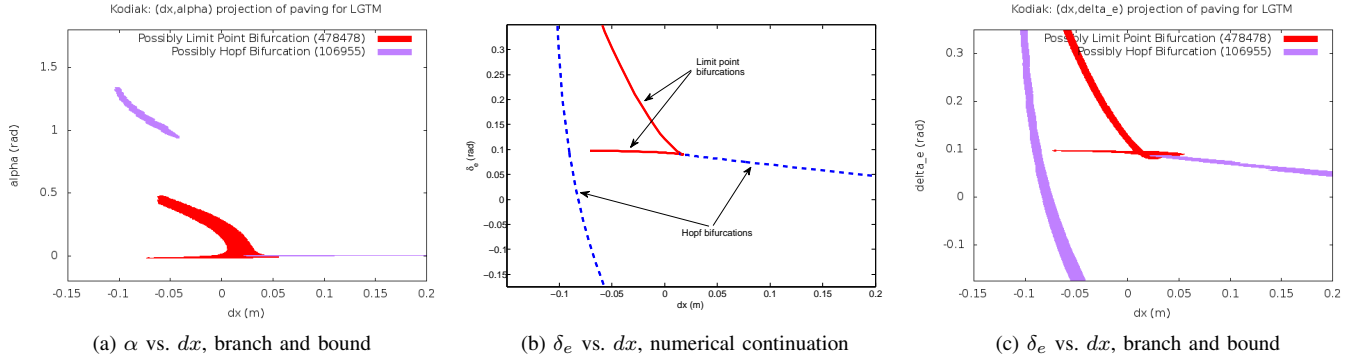


Fig. 2. Two-parameter varying: results of numerical continuation compared with paving projections with respect to α , δ_e , or dx .

$\delta_{th} \in [0.14, 0.26]$, $m_2 \in [4.7, 5]$ kg, $dx, dz \in [-0.021, 0.021]$ m. Further investigations strongly suggest that the worst-case perturbation radius for this equilibrium point is less than 0.17.

VII. CONCLUSION

There are two main advantages of the proposed branch and bound techniques for bifurcation analysis. Firstly, rigorous enclosure methods are employed and the entirety of the computational domain is considered, meaning that all branches of the equilibrium or bifurcation manifolds therein are guaranteed to be included; empty pavings may be considered equivalent to proofs. Secondly, statements can be made about the bifurcation set directly, without needing to compute the whole equilibrium manifold. Depending on the desired quality (fine or coarse) of the paving, the computational effort may be considerable, even for problems of moderate size (where $n_x + n_p \leq 10$). This effort increases exponentially with respect to the number of degrees of freedom in the paving, e.g., pavings for points are less intensive than pavings for line segments. The computed pavings can be used in concert with numerical continuation methods, either at the beginning, to compute guaranteed starting domains, or at the end, to formally verify the completeness of a solution. The guaranteed exclusion boxes can be computed much more quickly, since, if successful, the branch and bound method yields an empty paving. Therefore this particular technique is applicable for higher-dimensional problems in which many parameters are permitted to vary simultaneously.

In the future, the method will be tested on a wider range of aircraft models, with additional dynamics or design parameters. Numerous planned improvements to the Kodiak software tool, including additional enclosure methods and improved heuristics, may further extend the range of problems that can be solved.

ACKNOWLEDGMENT

Funding of the first author's research under NASA Cooperative Agreement NNL09AA00A is gratefully acknowledged. Collaboration between NIA and the University of Bristol was supported by Royal Society International Exchanges Scheme grant no. IE121367.

REFERENCES

- [1] H. Khalil, *Nonlinear Systems*, 3rd ed. Prentice Hall, 2002.
- [2] S. J. Gill, M. H. Lowenberg, S. A. Neild, B. Krauskopf, G. Puyou, and E. Coetzee, "Upset dynamics of an airliner model: A nonlinear bifurcation analysis," *Journal of Aircraft*, vol. 50, no. 6, pp. 1832–1842, 2013.
- [3] E. L. Allgower and K. Georg, *Numerical Continuation Methods*, ser. Computational Mathematics. Springer-Verlag, 1980.
- [4] M. El Kahoui and A. Weber, "Deciding Hopf bifurcations by quantifier elimination in a software-component architecture," *J. Symbolic Computation*, vol. 30, pp. 161–179, 2000.
- [5] M. Lerch, G. Tischler, J. Wolff von Gudenberg, W. Hofschuster, and W. Krämer, "filib++, a fast interval library supporting containment computations," *ACM Trans. on Mathematical Software*, vol. 32, no. 2, pp. 299–324, 2006.
- [6] R. E. Moore, R. B. Kearfott, and M. J. Cloud, *Introduction to Interval Analysis*. Philadelphia: SIAM, 2009.
- [7] A. P. Smith, "Fast construction of constant bound functions for sparse polynomials," *J. Global Optimization*, vol. 43, no. 2–3, pp. 445–458, 2009.
- [8] C. Muñoz and A. Narkawicz, "Formalization of a representation of Bernstein polynomials and applications to global optimization," *Journal of Automated Reasoning*, vol. 51, no. 2, pp. 151–196, August 2013. [Online]. Available: <http://dx.doi.org/10.1007/s10817-012-9256-3>
- [9] A. Sadrpour, L. G. Crespo, and S. P. Kenny, "Analysis of nonlinear systems via Bernstein expansions," in *AIAA Guidance, Navigation, and Control (GNC) Conference*. American Institute of Aeronautics and Astronautics, August 2013.
- [10] A. Narkawicz and C. Muñoz, "A formally verified generic branching algorithm for global optimization," in *Fifth Working Conference on Verified Software: Theories, Tools and Experiments (VSTTE)*, ser. Lecture Notes in Computer Science, E. Cohen and A. Rybalchenko, Eds., vol. 8164, 2014, pp. 326–343.
- [11] J. Garloff and A. P. Smith, "Investigation of a subdivision based algorithm for solving systems of polynomial equations," *J. of Nonlinear Analysis: Series A Theory and Methods*, vol. 47, no. 1, pp. 167–178, 2001.
- [12] A. Chakraborty, P. Seiler, and G. J. Balas, "Nonlinear region of attraction analysis for flight control verification and validation," *Control Engineering Practice*, vol. 19, pp. 335–345, 2011.
- [13] E. Coetzee, B. Krauskopf, and M. Lowenberg, "The Dynamical Systems Toolbox: Integrating AUTO into Matlab," in *16th U.S. National Congress of Theoretical and Applied Mechanics*. U.S. National Congress of Theoretical and Applied Mechanics, 2010, paper USNCTAM2010-827.
- [14] E. J. Doedel and E. Oldeman Bart, *AUTO-07P: Continuation and Bifurcation Software for Ordinary Differential Equations*, January 2012. [Online]. Available: <http://sourceforge.net/projects/auto-07p/files/auto07p/>