Aerospace Science and Technology ••• (••••) •••–•••

Contents lists available at ScienceDirect

Aerospace Science and Technology

www.elsevier.com/locate/aescte

Automatic mode tracking for flight dynamic analysis using a spanning algorithm

C.S. Beaverstock [a,*], M.I. Friswell [a], S. Adhikari [a], T.S. Richardson [b], J.L. Du Bois [c]

[a] Swansea University, College of Engineering, Singleton Park, Swansea, SA2 8PP, UK
[b] University of Bristol, Queen's Building, University Walk, Clifton, Bristol, BS8 1TR, UK
[c] University of Bath, Department of Mechanical Engineering, Claverton Down, Bath, BA2 7AY, UK

ARTICLE INFO

Article history:
Received 13 April 2015
Received in revised form 3 July 2015
Accepted 29 August 2015
Available online xxxx

ABSTRACT

Identifying and tracking dynamic modes in a multi-dimensional parameter space is a problem that presents itself in many engineering disciplines. In a flight dynamics context, the dynamic modes refer to the modes of motion obtained from a linearisation of the aircraft system about a known operating point. Typically dynamic results derived from these linear models are unsorted, where mode indices are unrelated from one operating point to the next. When varying the parameters, or in this case operating point, difficulties in automating the process of relating modes from a linear system derived at one parameter set to the next exists. This paper builds on the work in tracking modes in a structural context, using the Modal Assurance Criterion (MAC) to numerically relate modes from two comparable linear systems. The (MAC) is deployed within a spanning algorithm to discover and identify all modes within all conditions, with their relationship to adjacent/neighbouring conditions. This is tested on a 1-, 2- and 3-dimensional parameter space, twelve state system.

1. Introduction

For a nonlinear dynamic system, the modal characteristics and shapes are likely to vary considerably throughout the operational envelope of the system. Observing how specific modes evolve in a system can be beneficial when analysing the stability properties, and in the development of controllers. The linear dynamic properties of these nonlinear systems are typically obtained by linearising the system about an operating point, and using linear systems analysis methods, the modal properties can be extracted. However, the analysis method typically generates unsorted eigenstructures of the system dynamics, such that there is no relation between the data from one operating condition to another.

Identifying and tracking modes is also a problem in structural dynamics. Structural dynamicists are concerned with, for example, validating a model to represent one or more modes of a system from experimental data, at various operating points. The Modal Assurance Criterion (MAC) has been widely used to address problems in structural dynamics and vibrations [20], providing a measure to correlate modes from two comparable systems. In this case, the system is a structure, where the system dynamics of a model is being compared to the experimental data at the equivalent operating point. It is conceivable that using the same algorithm for the model at varying operating points, the dynamic results can be compared to relate the modes from one operating point to another.

In this paper, the MAC is applied to adjacent/neighbouring (graph theory) operating points, to ascertain dynamic correspondence, and track a mode through this parameter space. The paper deploys this MAC in a spanning algorithm to identify and link modes. Where previous papers address deployment of the MAC for comparison of two systems, or tracking over a single dimension, this paper proposes an algorithm suitable for comparisons over an N-dimensional parameter space of the operating condition. The algorithm is deployed using an 'embarrassingly parallel' approach to span modes. Although this enables multiple cores to be used to reduce the time to solution, it is also shown that the computational resources are not being exploited to their full potential, resulting in significant processor idle time. This is applied to a flight dynamics example, for a model of a novel Unmanned Aerial Vehicle (UAV) platform, simulated using a standard 6 Degrees of Freedom (DoF) Equation of Motion (EoM), using the Euler angle description for orientation.

Section 2 provides the motivation behind the proposed framework for the modal identifying and tracking algorithm, and a de-

* Principal corresponding author.
E-mail addresses: c.s.beaverstock@swansea.ac.uk (C.S. Beaverstock), m.i.friswell@swansea.ac.uk (M.I. Friswell), s.adhikari@swansea.ac.uk (S. Adhikari), t.s.richardson@bristol.ac.uk (T.S. Richardson), j.l.du.bois@bath.ac.uk (J.L. Du Bois).

http://dx.doi.org/10.1016/j.ast.2015.08.013
1270-9638/

Nomenclature

A	Linear time invariant system plant matrix
B	Linear time invariant system control matrix
C	Linear time invariant system output matrix
D	Linear time invariant system feed forward matrix
x	System state vector
u	System input vector
v	System eigenvector matrix
λ	System diagonal eigenvalue matrix
$\mathbf{\Phi}_{c,m}$	Eigenvector of the mth mode of the cth system
$\text{MAC}_{c,d,[m,n]}$	MAC value comparing the eigenvector of the mth mode of the cth system, to the eigenvector of the nth mode of the dth system
α and β	Angles of incidence: angle of attack and sideslip angle respectively
V_t	Aircraft flight speed
$\bar{p}, \bar{q}$ and $\bar{r}$	Body axis rotation rates: Roll, pitch and yaw respectively
ϕ, θ and ψ	Orientation expressed in Euler angles: Roll, pitch and yaw respectively
$-z$ or h	Altitude in an Earth frame of reference
ΔX_{CoG}	Increment in aircraft Centre of Gravity (CoG) from nominal position
η	Span retraction parameter
ζ	Damping ratio

Superscript

$*$	Complex conjugate
T	Transpose of a matrix

scription of the methods deployed to perform the modal analysis and categorisation.

Section 3 presents a structure for the proposed framework, with a description of the required components. Section 4 presents the theory of the methods used in the components for the framework.

Section 5 presents the aircraft model example which will be used to generate results from this algorithm.

Section 6 presents results for the aircraft example, and a discussion of the algorithm performance, followed by conclusions and recommendations presented in Section 7.

2. Background

Many engineering systems are nonlinear, where the linear dynamic characteristics will vary over its operational envelope. These dynamic systems can be represented by a set of dynamic first order system state equations, where the dynamic state is a function of time, system state and inputs [28]. Furthermore, this model can be idealised to a linear model about a specific system state and input [26], from which the linearized systems model can be derived/generated. From this, traditional linear dynamic analysis tools can be used to analyse the system [10]. These systems can be represented in state-space form or by transfer functions [25]. Linear modes can be found through the system's eigenstructure [16], where the eigenvalues define the modal characteristics and the eigenvector the mode shape.

Algorithms used to extract the eigenstructure generally output an unstructured set of eigenvalues and eigenvectors [24]. This means that for a nonlinear system, where varying the operating point modifies the linear description of the system, the modes output from the algorithm indexed with the same number are generally unconnected. For systems of greater than two modes, this makes it difficult to trace how the characteristics of the modes change over the system's operational envelope. For this, a robust method is required to relate modes from varying eigenstructure for a system over its operational envelope. Furthermore, to design controllers to modify the behaviour of specific modes requires the distinction of these modes, such that the controller can be modified (feedback gains) to appropriately control [14] the response for varying operational conditions.

The MAC is a correlation parameter [1], computed by comparison of two comparable eigenvectors. The value of this parameter compares the similarity of one mode shape to another, and is bounded between 0 to 1. A value of 0 represents no correlation, where values increasing approaching unity signify the increasing similarity, and hence correlation, of mode shapes. It has been extensively deployed to analyse structural engineering system modes, to track how vibration modes change due to variation in a design or operational parameter [21], or using an improved formulation of the MAC to small changes, for example to measure the effect of salt accumulation to a turbine blade on its dynamics [18]. Furthermore, for comparisons between experimental and analytical models, the MAC is deployed to provide information of the modal response, and hence is an indication of correspondence between the analytic and experimental models. Amsallem and Farhat [3] presents a framework to interpolate Linear Parameter Reduced Order Model (LPROMs) that uses the MAC parameter to identify potential cross-over points, or behaviour associated with the mode veering phenomenon. Because these systems can be large, with multiple interacting modes, a robust algorithm is required to ensure accurate tracking of each individual mode's eigenvalue. When extending this problem into multiple dimensions, a simple single dimensional spanning algorithm is no longer viable to track modes in this multi-dimensional space.

Other parameters that are used to correlate modes includes the Modal Observability Criterion (MOC) parameter outlined by Yaghoubi and Abrahamsson [27], and a series of parameters that extend the original MAC presented by Allemang [1]. These use alternative formulations or adaptations of the MAC correlation parameter. Some of these formulations include the mass or stiffness matrix to ensure orthogonality between the modes.

The framework developed in this paper builds on the MAC parameter, which succeeded in its application to demonstrate the benefits of applying a correlation parameter alongside a spanning algorithm to track the dynamic modes present in flight dynamic systems. The MAC within this framework can be replaced by other correlation parameters, where the application permits, along with the spanning algorithm proposed to track the modes across several parameters. Alternately, other numerical methods that potentially involve mode tracking as a part of the framework include numerical continuation [2] and mode veering [3,11,8].

3. Algorithm overview and structure

In this section, an outline of the algorithm used to identify, assign a unique index, and link associated modes through an N-dimensional parameter space is given. This includes defining the input to the framework, the functions and algorithms used to drive the analysis, identify and track the modes in the parameter space. Fig. 1 provides an illustration of the proposed framework used to develop the algorithm.

The schematic of the software framework shows the data flow through the various components. The letter 'D' indicates a database, where 'F' is used to refer to a function or algorithm.

Please cite this article in press as: C.S. Beaverstock et al., Automatic mode tracking for flight dynamic analysis using a spanning algorithm, Aerosp. Sci. Technol. (2015), http://dx.doi.org/10.1016/j.ast.2015.08.013

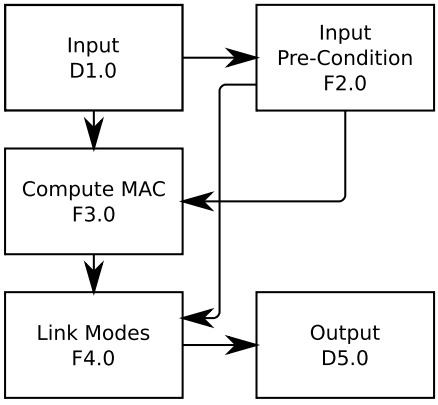


Fig. 1. Software framework hierarchy.

These components are marked by '#.0', where the '.0' refers to the base level component, with appropriate values assigned for sub-level components in the hierarchy. Essentially the framework allows for a structured input of linear systems for pre-defined operating points, where these undergo pre-processing to condition the input for comparison of conditions, followed by linking each condition. This produces a set of modes, where each mode is related to any connecting operating condition mode with maximum correspondence. The framework components are described in the following sections.

3.1. Input interface

The framework requires a structured input according to the independent parameter space. For each parameter set that defines an operating point, a linear state-space system is required. Linear systems analysis tools can then be used to compute the system's eigenstructure, which describes the linear dynamic state at that operating point. Generally, the output from algorithms used to compute the eigenstructure are unstructured [4], hence there is no relation between the modes computed at one operating point to another. This necessitates a method to relate modes, such that the dynamic behaviour and modes can be tracked through a multi-dimensional parameter set.

The driver to compare and relate modes from the linear systems at two different parameter sets is the MAC, which is a correlation parameter that compares mode shapes to quantify their similarity. To deploy this effectively, the search space for comparisons is limited to operating points that are local, or in Graph Theory [15] terminology, only neighbouring nodes (operating points) are compared. This reduces both the memory requirement and number of executions of the MAC function. This section describes components D1.0 and F2.0 from Fig. 1. Fig. 2 illustrates this using terminology derived in Graph Theory [15], and shows the nodes that are investigated, or interrogated by the algorithm.

Essentially, D1.0 is a database of all available parameters for a set of systems. As a minimum this includes the state-space description of the system at each operating point. The preconditioning algorithm (F2.0 in Fig. 1) is then used to supplement data which is required to track the modes, which is as follows:

1. Calculate eigenstructure at each node (eigenvalues and eigenvectors)
2. Declare connectivity between node pairs (requires minimal information)
3. Identify node pairs
4. Prepare data-structures for storing run data

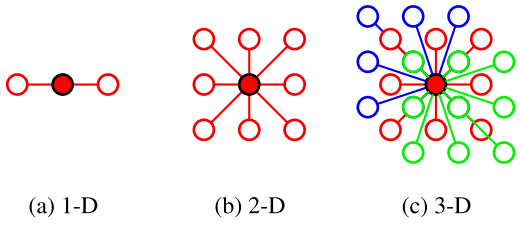


Fig. 2. Node network visited by the algorithm (connectivity).

From this, the primary algorithms can be executed to identify and track modes. The eigenstructure for a standard form linear state-space system can be computed by solving to find the roots of the system plant matrix (see Sections 4.1.1 and 4.1.2). This can be addressed relatively trivially using the *eig* command within MATLAB, which uses QZ-decomposition [13] to compute the eigenstructure.

3.2. Correlation parameter: the Modal Assurance Criterion (MAC)

At the core of this framework is the MAC function, which drives the algorithm to identify, index and track modes in an N-dimensional parameter space. The F3.0 component in Fig. 1 represents the function to compute the MAC parameter, and through using the connectivity between node pairs information, the MAC between each pair is computed and stored. Each pair is only compared once, and only compared to its adjacent neighbours from the connectivity array. This is followed by an algorithm to pick the maximum MAC for each mode being compared with another comparable mode. Modes can only be linked to a single mode from another condition.

The basis of the MAC is that it compares the mode shapes of two comparable modes through their eigenvectors. This compares the relative magnitudes and angles between the vectors of each of the states, to generate a correlation parameter, that quantifies the relative difference in mode shape. In the context of two linear systems, A and B, each mode in system A can be compared to all modes in system B, where the maximum MAC value relates the mode from A to the corresponding mode in B that produces this value.

3.3. Spanning/linking algorithm

The MAC provides the function to compare modes from two parameter sets, where the input pre-conditioning provided a method for only comparing modes from connected operating points. From this, all modes in connected operating points can be compared, identified and linked if necessary. It is then required that all unique modes can be identified and tracked. This requires a spanning algorithm, where modes from each operating point can be spanned until a boundary is met in the parameter space. Modes may not span the entire operating point parameter space. For operating points where a spanning mode cannot be connected, this operating point can be flagged to begin a new unique mode to span. The failure of a mode to span the space generally occurs when either a complex mode encounters the real axis, or two aperiodic modes interact and form a complex mode, when undergoing parameter variation for two comparable operating points. When a mode undergoes this transition, the mode/s are no longer comparable, and so a protocol is used to generate separate modes. This is observed on the root locus where varying a parameter causes a modification to the character of a system, a bifurcation, to the dynamics where a complex conjugate pair becomes a real pair of modes (or visa versa). This continues until all modes have been indexed and connected. This is represented by the F4.0 component within the framework presented in Fig. 1.

Please cite this article in press as: C.S. Beaverstock et al., Automatic mode tracking for flight dynamic analysis using a spanning algorithm, Aerosp. Sci. Technol. (2015), http://dx.doi.org/10.1016/j.ast.2015.08.013

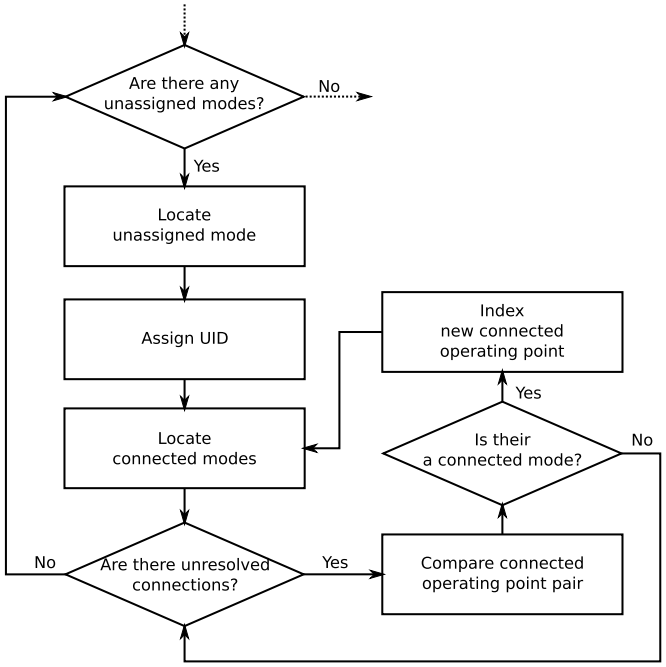


Fig. 3. Spanning algorithm structure to span node space.

Fig. 3 presents a structure to develop a pseudo-code (see Section 4.3) for the spanning algorithm. The first part of the algorithm must identify a mode at a particular operating point which is yet to be assigned. If a mode is found, a new thread is spawned, and the spanning algorithm begins. Each iteration of the spanning algorithm compares all operating points connected to the current active operating point that have been identified as connected to the mode of interest. If the operating point is connected, then all connected operating points that have not been investigated for connectivity are appended to a list of operating points to investigate. This continues until all operating points have been investigated, or no more connected operating points can be found. This algorithm is repeated until all modes have been indexed, mapping the connectivity of modes. The output of this algorithm is a complete mapping of indexed modes.

4. Methods and tools

Section 3 presented the outline of the required framework for this work, this section will present the mathematical tools and pseudo code required to construct functions and algorithms in the framework. This includes the input preconditioning algorithms, MAC function and finally the algorithms used to link the modes.

4.1. Linearized systems

Although the systems being analysed are essentially nonlinear, linear analysis at progressing operating points is used to infer nonlinear behaviour, and can be considered a form of grid-based Linear Parameter Varying (LPV) system when observing these linear systems over the parameter space of interest. The dynamic representation of a linear system obtained through linearising a nonlinear system, is a valid description of the dynamics for only small relative perturbations or amplitudes of the systems state. The tracking algorithm assumes small perturbations to the structure of the plant matrix (**A** matrix) for the variation in operating point, such that the eigenvectors observed are from the same system. The data set uses a grid that is sufficiently dense that for modes whose eigenvectors are similar the eigenvalues must also be close. For the linear system generated at each operating point, the system plant matrix (**A** matrix) is used to track the modes.

4.1.1. State-space model

Linear systems can be represented by both time and frequency domain forms. The general system state-space form representation is:

$$\dot{\mathbf{x}} = \mathbf{A}(\mathbf{x}, \mathbf{u})\mathbf{x} + \mathbf{B}(\mathbf{x}, \mathbf{u})\mathbf{u}\,, \tag{1}$$

$$\mathbf{y} = \mathbf{C}(\mathbf{x}, \mathbf{u})\mathbf{x} + \mathbf{D}(\mathbf{x}, \mathbf{u})\mathbf{u}\,. \tag{2}$$

With this description, the systems dynamic modes can be computed from the systems plant matrix (**A**).

4.1.2. Eigenstructure: eigenvectors and eigenvalues

The dynamic of the system can be determined by first taking a Laplacean of Equation (1), putting the system in the frequency domain, and solving the following equation:

$$\mathbf{A}\mathbf{v} = \lambda\mathbf{v}\,. \tag{3}$$

By solving this equation, the eigenvalues, λ, and corresponding eigenvectors, $\mathbf{v}$, can be computed. The eigenvalues describe the modal properties (frequency and damping), where the eigenvectors describe the mode shape. When identifying a mode with parameter variations, it is the eigenvectors that are used to identify the mode. Comparison of the eigenvectors of two comparable systems should reveal the correspondence of the modes from one system to another.

In the context of flight dynamics, the Short Period refers in general, to the complex conjugate pair, with the poles located in positions of relative high frequency and moderate damping when compared to other longitudinal flight modes. The Short Period would be identified by observing the eigenvector with relatively large contributions from the angle of incidence and pitch rate in the dynamics of the mode.

4.2. Modal Assurance Criterion (MAC)

The MAC is a correlation parameter, which compares two eigenvectors to quantify the similarity (or difference) in the mode shapes. The MAC is defined as:

$$\mathrm{MAC}_{c,d,[m,n]} = \frac{\left|\{\boldsymbol{\Phi}_{c,m}\}^T\{\boldsymbol{\Phi}^{\star}_{d,n}\}\right|^2}{\left(\{\boldsymbol{\Phi}_{c,m}\}^T\{\boldsymbol{\Phi}^{\star}_{c,m}\}\right).\left(\{\boldsymbol{\Phi}_{d,n}\}^T\{\boldsymbol{\Phi}^{\star}_{d,n}\}\right)}\,, \tag{4}$$

where $\boldsymbol{\Phi}_{c,m}$ defines the eigenvector corresponding to mode index m of linear system c, $\boldsymbol{\Phi}_{d,n}$ similarly mode index n of linear system d, $\boldsymbol{\Phi}^*$ indicates the complex conjugate of $\boldsymbol{\Phi}$, $\boldsymbol{\Phi}^T$ indicates the transpose of $\boldsymbol{\Phi}$ and $\mathrm{MAC}_{c,d,[m,n]}$ the MAC value corresponding to the comparison of mode m with n, from system c and d respectively.

The following example is of a longitudinal flight dynamic model, with two complex conjugate modes. The state vector consists of angle of incidence, flight speed, pitch rate and pitch angle ($[\alpha, V_t, q, \theta]$). The modes of the aircraft are calculated for two trimmed flight conditions at two different speeds. System X is at 50 kph and Y at 55 kph; the system's plant matrix and eigenstructures are:

System X:

$$\mathbf{A}|_X = \begin{bmatrix} -2.2244 & -0.0593 & 0.9228 & -0.2171 \\ -9.6156 & 0.1185 & 0.3094 & 8.6460 \\ -53.4078 & -1.7931 & -2.8787 & 5.5391 \\ 0 & 0 & 1.0000 & 0 \end{bmatrix}$$

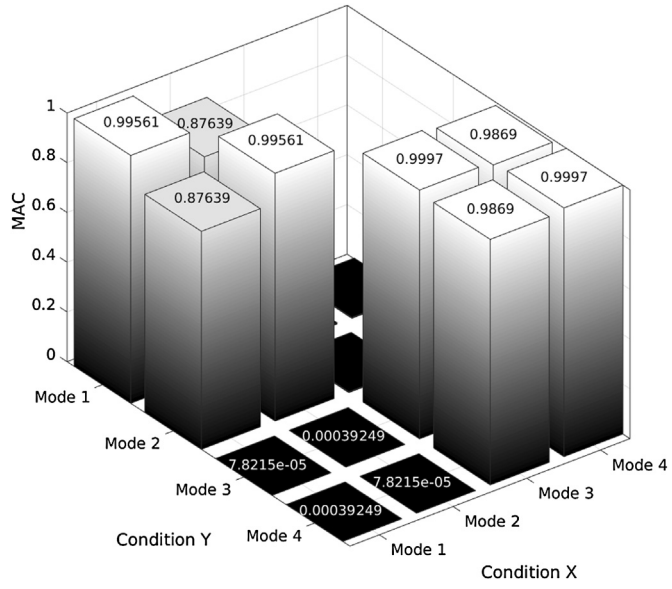


Fig. 4. MAC example.

$$\lambda_{1,2}|_X = -2.8092 \pm 6.6991i \quad \lambda_{3,4}|_X = 0.3169 \pm 0.4675i$$

$$\mathbf{v}_{1,2}|_X = \begin{bmatrix} -0.0078 \pm 0.1358i \\ 0.0328 \pm 0.0034i \\ 0.9809 \pm 0.0000i \\ -0.0522 \pm 0.1245i \end{bmatrix}$$

$$\mathbf{v}_{3,4}|_X = \begin{bmatrix} 0.0327 \mp 0.0057i \\ -0.9970 \mp 0.0000i \\ 0.0319 \mp 0.0131i \\ 0.0124 \mp 0.0598i \end{bmatrix}$$

System Y:

$$\mathbf{A}|_Y = \begin{bmatrix} -2.5732 & -0.0495 & 0.9229 & -0.1698 \\ -6.0271 & 0.1692 & 0.4361 & 8.6874 \\ -64.7566 & -1.6061 & -3.1510 & 5.8961 \\ 0 & 0 & 1.0000 & 0 \end{bmatrix}$$

$$\lambda_{1,2}|_Y = -3.0391 \pm 7.4236i \quad \lambda_{3,4}|_Y = 0.2616 \pm 0.4222i$$

$$\mathbf{v}_{1,2}|_Y = \begin{bmatrix} -0.0052 \mp 0.1233i \\ -0.0304 \pm 0.0055i \\ 0.9843 \pm 0.0000i \\ -0.0465 \mp 0.1136i \end{bmatrix}$$

$$\mathbf{v}_{3,4}|_Y = \begin{bmatrix} -0.0239 \pm 0.0042i \\ 0.9981 \pm 0.0000i \\ -0.0227 \pm 0.0113i \\ -0.0048 \pm 0.0508i \end{bmatrix}$$

The MACs matrix computed is as:

$$\mathbf{MAC}|_{X,Y} = \begin{bmatrix} 0.9956 & 0.8764 & 0.0047 & 0.0033 \\ 0.8764 & 0.9956 & 0.0033 & 0.0047 \\ 0.0001 & 0.0004 & 0.9997 & 0.9869 \\ 0.0004 & 0.0001 & 0.9869 & 0.9997 \end{bmatrix}$$

This is graphically represented in Fig. 4.

All values are between 0 and 1, values of zero representing no correlation, and 1 representing equivalent modes. As can be seen here, the modes that are highlighted on the diagonal of the MAC array are the modes that are correlated from condition X to condition Y. Also observed from the array and Fig. 4, the correlated mode and its complex conjugate are generally closer than the other complex conjugate pair. Note that if two modes are correlated to a single mode from the other condition, it is the maximum of these that is selected.

The two modes identified are the Short Period $(1, 2)$ and phugoid $(3, 4)$. By inspection, the Short Period and phugoid in both systems are easily identifiable by their eigenvalues. Also, by inspection of the eigenvectors, these modes can be identified by their distinct mode shapes. As is highlighted on the diagonal of the MAC matrix for the comparisons of these systems, the correct eigenvalues are identified by the algorithm. Note that the implementation of the algorithm ignores comparisons of real with complex, regarding these as incomparable modes. Therefore the tracking algorithm only follows purely real or purely complex modes.

4.3. Spanning algorithms

To deploy this effectively to identify all simply connected modes, a spanning algorithm is required to identify and index new modes, linking these modes to all connected modes from other parameter sets. In the simplest case of 1-dimensional spanning, the algorithm is only required to interrogate modes from one or two adjacent parameter sets. In N-dimensions, to span a mode over a more general parameter space, a more complex algorithm is required, which can work regardless of the dimensionality of the problem. Essentially the dimensions are the degree of the nodes (the number of neighbours connected to each node), for example, when expanding over 3 parameters, there are up to 26 connected neighbours (see Fig. 2c). Because the connectivity need not be structured, the algorithm has been designed to manage over arbitrary dimensions at each node.

With this code, driven by the MAC, a mode index can be spawned and span within the parameter space where the mode exists. From this, all modes can be assigned an index relating it to all similar modes in the parameter space. The algorithm presented in Appendix A can be parallelised, although with its current structure this is limited to the number of modes at any one node using simple parallelisation methods.

5. Tracking flight dynamic modes

The following example is of a flight dynamics problem, where the model was used first by Beaverstock et al. [7,5,6] to investigate the effect of span morphing on the flight modes. The aircraft model is based on the Tekever AR4, and modified to accommodate span morphing (image with illustration presented in Fig. 5).

The unmorphed and morphed wing parameters that define the aircraft are summarised in Table 1 [6].

For this model, b_η is the span position with morphing applied, b_0 is the nominal full span, η is the span retraction parameter (1 for fully retracted and 0 for no retraction), $b_\eta|_{\eta=1}$ is the span position when fully retracted and $\bar{c}$ is the mean aerodynamic chord. The model contains a 6-DoF Equation of Motion (EoM). The model typically produces five modes broken down as follows:

- Longitudinal
  - Short Period: complex conjugate pair typified by angle of attack (α) and pitch rate ($\bar{q}$)
  - Phugoid: complex conjugate pair, long period mode typified by variations in velocity (V_t), attitude (θ) and altitude ($-z$ or h)

Table 1
Varying properties of the reference UAV based on the Tekever AR4 presented by Beaverstock et al. [6].

Parameter	Unmorphed	Morphed asymmetric	Morphed symmetric
AR	6.67	5.83	5
H-Tail Volume ($\frac{S_t l_t}{S\bar{c}}$)	0.2966	0.3389	0.3954
V-Tail Volume ($\frac{S_v l_v}{Sb}$)	0.0258	0.0336	0.0458
CoG Y Pos (% Span)	0	±0.28	0
$[I_{xx}, I_{yy}, I_{zz}]$ (kg m^2)	[7.54,6.88,13.14]	[6.68, 6.88, 12.28]	[5.83,6.88,11.42]
$[I_{xy}, I_{xz}, I_{yz}]$ (kg m^2)	[0,0.28,0]	[±0.02, 0.28, ±0.05]	[0, 0.28, 0]

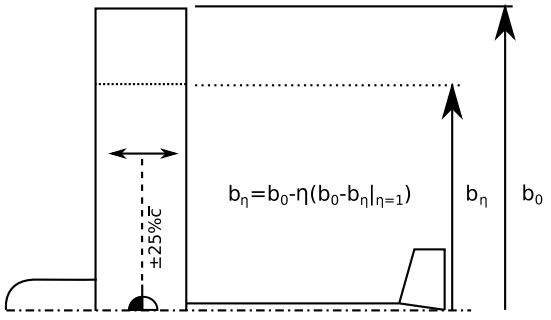


Fig. 5. Modified AR4 span morphing.

- Lateral
  - Dutch Roll: complex conjugate pair characterised by the side-slip (β) and yaw rate ($\bar{r}$), and also roll rate ($\bar{p}$)
  - Roll Convergence/Subsidence: a real mode dominated by roll rate ($\bar{p}$) and roll angle (ϕ)
  - Spiral: A slow mode characterised by side-slip (β), roll angle (ϕ)

These modes are described in many textbooks [12,17,19,22,23,9], along with methods to estimate/ predict both the modal characteristics (eigenvalue) and mode shape. Although these modes are generally observed, changes in aircraft design and flight condition affect the contribution of each state to the modes. This leads to subtle changes in the mode, where, for example, the characteristic roll-yaw rate ratio of the Dutch Roll may vary. The literature also suggests that in more extreme flight regimes, or unconventional designs, atypical modes can be present and identified.

The following example shows changes in the mode due to parameter changes in the velocity (V_t), span morphing parameter (η) and CoG position (ΔX_{CoG}). The parameter ranges and increments are summarised in Table 2.

6. Results

In the following section, results showing the unprocessed dynamic data, and numerical outcomes of applying the algorithm are presented. This is followed by an analysis into the variation in dynamic characteristics, consisting of modal characteristics, across the parameter space under consideration. These are presented alongside the maximum MAC values for the mode at each parameter set.

6.1. Unsorted dynamic results

To emphasise the problem, all modes are plotted in Fig. 6, for the case of a 1-, 2- and 3-dimensional problem. As can be observed, trends in modes increase in complexity, in their interactions and number as the dimensions increase. In Fig. 6, the variation of each individual parameter is highlighted, holding the other two parameters at a nominal value.

Table 2
Summary of parameter ranges and increments.

Parameter	Range		Increment
	min	max	
V_t (kph)	55	150	5
η %	0	100	5
ΔX_{CoG} % $\bar{c}$	−25	25	5

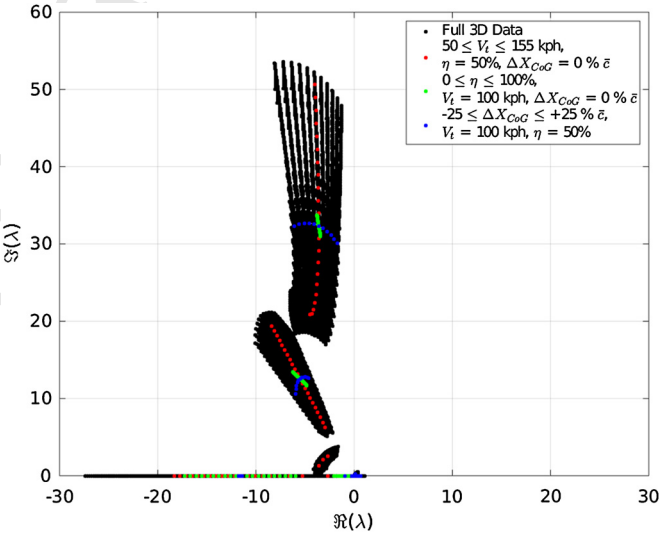


Fig. 6. Unsorted modes varying across 3 parameters.

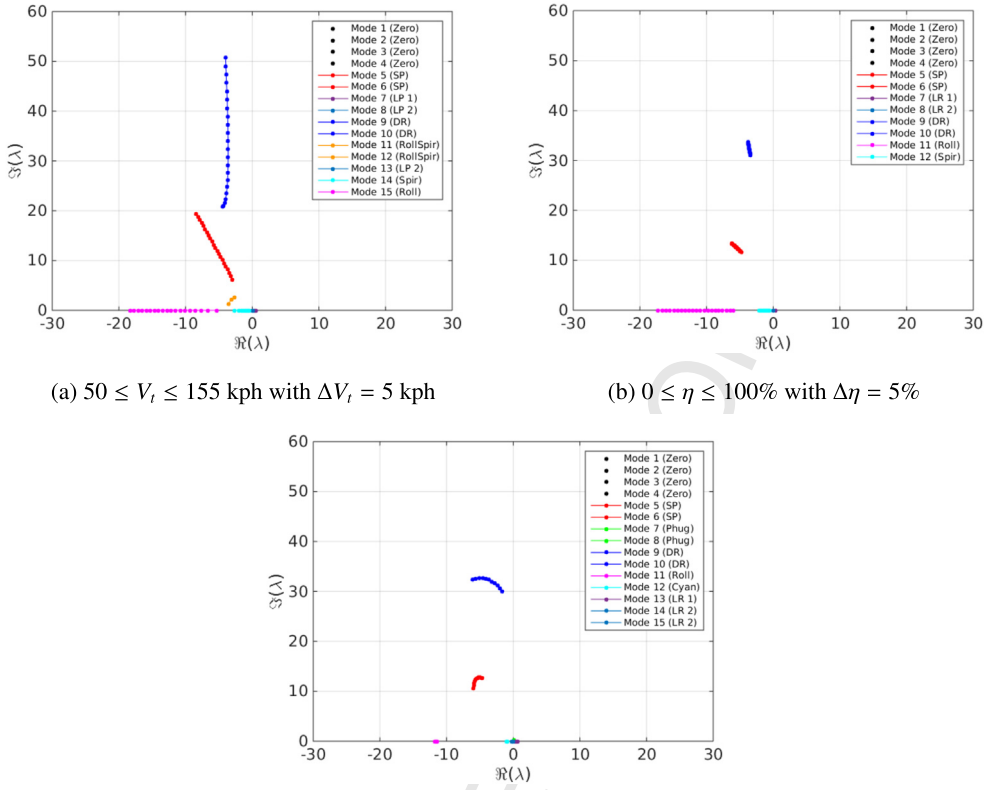
In some cases, the related modes are clearly observable graphically, and as can be seen, varying parameters individually, the behaviour of these clearly defined modes are relatively simple/predictable. Although these evaluations/observations can easily be made visually, automating this to sort the modes can be a labour intensive and cumbersome process to manage manually. Over large data sets spanning multiple dimensions, this may not be practical or even feasible in a reasonable time frame. Furthermore, more complex interactions, where modes vary between aperiodic and harmonic, maybe more difficult to analyse manually over a multi-dimensional space.

With this unsorted data, dynamic characteristics cannot be illustrated whilst varying parameters, as there is no structure to determine which modes are related to which in each flight condition.

6.1.1. Identified modes

The first test for the algorithm is to observe the tracking of a mode/s through the variation of a single parameter. Fig. 7 presents the variation in a single parameter for each of the 3 parameters. These are varied as the single parameters are varied in Fig. 6.

Fig. 7 shows that more than 12 modes are identified. Essentially because the algorithm spans and terminates at nodes where a neighbouring node has no comparable mode; a new mode is



(a) $50 \leq V_t \leq 155$ kph with $\Delta V_t = 5$ kph

(b) $0 \leq \eta \leq 100\%$ with $\Delta\eta = 5\%$

(c) $-25 \leq \Delta X_{CoG} \leq 25\%\bar{c}$ with $\Delta\left(\Delta X_{CoG}\right) = 5\%\bar{c}$

Fig. 7. Sorted modes varied over 3 parameters individually.

Table 3
Summary of run information for varying 3 parameters individually (see Fig. 7).

Parameter	No. of modes	No. of cycles	Time to completion (s)
V_t	15	2	0.1222
η	12	2	0.0732
ΔX_{CoG}	15	3	0.1197

spawned at the unidentified mode in neighbouring node. Also in a single parameter variation example, because no other parameter exists to navigate around these nodes, disjointed sets where two modes are related may exist. Table 3 summarises the running of each set, setting the number of modes identified, the number of parallel cycles of the algorithm and the time taken to complete execution of the algorithm. In each cycle, a single node is selected, the node with the maximum number of unidentified modes, where each of these modes represents a thread on a single processor. This means that if there are fewer jobs (modes available) than processors available, those processors without jobs will be idle for this cycle. Furthermore, because the number of maximum available modes at a node diminishes with each cycle, generally the number of idle processors increases as the number of cycles increase. This was performed on a Dell Precision T5500, using 2 Intel Xeon X5650 @ 2.67 GHz (6 physical cores per processor). This enables 12 modes per flight condition to be sorted for each cycle of the algorithm. This means all 12 modes can be followed in parallel.

Where a single parameter is varied, it can be observed that more than 12 modes are tracked through the span of the parameter. These plots in Fig. 7, along with Table 3, show that some modes are observed for all values of a parameter, where others are only observed partially over the envelope. The disjoint is typically caused by a mode behaviour changing between aperiodic and harmonic.

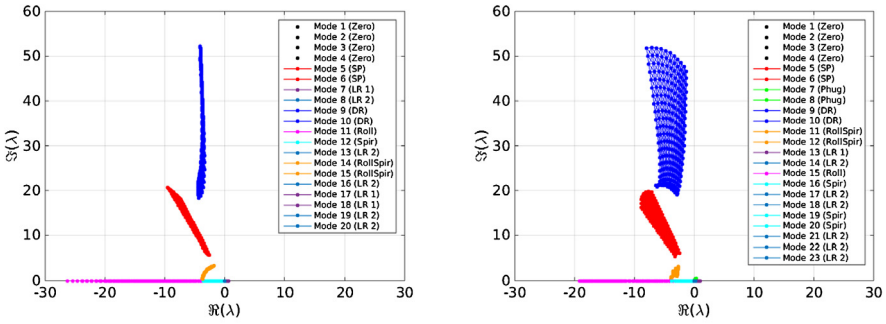
A similar analysis to that which is presented in Fig. 7, where two parameters are varied concurrently, is presented in Fig. 8.

From Fig. 8, the algorithm has identified a greater number of unique modes when compared to the single parameter case. In general, the modes outlined in Section 5 were clearly identified, with the variation approximately following the trend of combining the appropriate two results from Fig. 7. Thus, each mode can be individually extracted and analysed to investigate its behaviour, and how it varies over each parameter combination. Fig. 9 highlights the effect of two dimensional variations of the parameters on the Short Period mode. This mode was selected automatically from the 12 modes available at each flight condition, and linked through the two-dimensions.

Again, additional modes to the initial 12 are identified, with complex interactions with the real axis. The algorithm, as can be seen in the timing results from Table 4, performs this for all cases in under 30 seconds. Performing this manually would require sorting through in excess of 1000–1500 pairwise comparisons to link the modes, and repeated for however many sets of data. Furthermore, the code developed can be used over an arbitrary number of dimensions, negating the necessity of developing bespoke code for data for varying dimensionality.

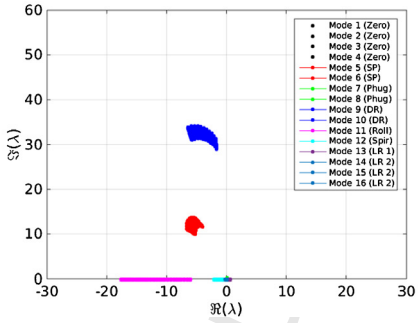
As can be observed, running over increased dimensions increases the time to solve and complete the algorithm. The added dimension means that the algorithm complexity does not scale solely on the increased number of flight conditions, but also the additional dimension increases the complexity as a function of both conditions and dimensions. Fig. 10 presents the graph connectivity of some example modes, showing the effect of the spanning algorithm for a complete graph for a mode at all flight conditions, along with the graph that does not fully span the space.

Please cite this article in press as: C.S. Beaverstock et al., Automatic mode tracking for flight dynamic analysis using a spanning algorithm, Aerosp. Sci. Technol. (2015), http://dx.doi.org/10.1016/j.ast.2015.08.013



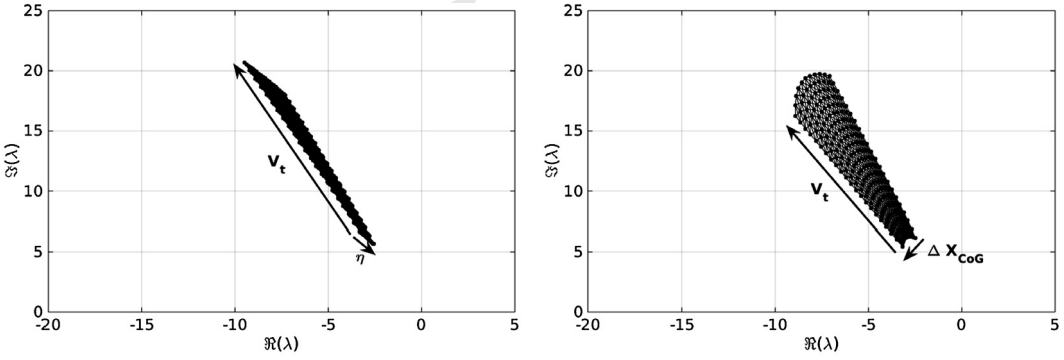
(a) $50 \leq V_t \leq 155$ kph with $\Delta V_t = 5$ kph & $0 \leq \eta \leq 100\%$ with $\Delta\eta = 5\%$

(b) $50 \leq V_t \leq 155$ kph with $\Delta V_t = 5$ kph & $-25 \leq \Delta X_{CoG} \leq 25\%\bar{c}$ with $\Delta(\Delta X_{CoG}) = 5\%\bar{c}$



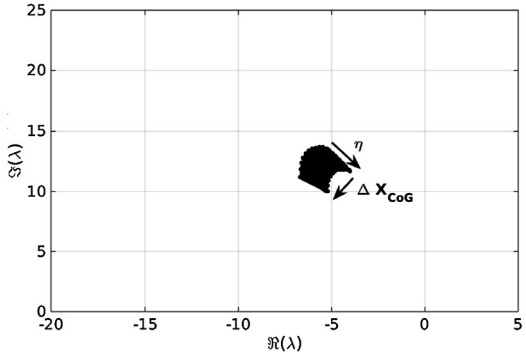
(c) $0 \leq \eta \leq 100\%$ with $\Delta\eta = 5\%$ & $-25 \leq \Delta X_{CoG} \leq 25\%\bar{c}$ with $\Delta(\Delta X_{CoG}) = 5\%\bar{c}$

Fig. 8. Sorted modes varied over the 2 parameters for 3 parameter combinations.



(a) $50 \leq V_t \leq 155$ kph with $\Delta V_t = 5$ kph & $0 \leq \eta \leq 100\%$ with $\Delta\eta = 5\%$

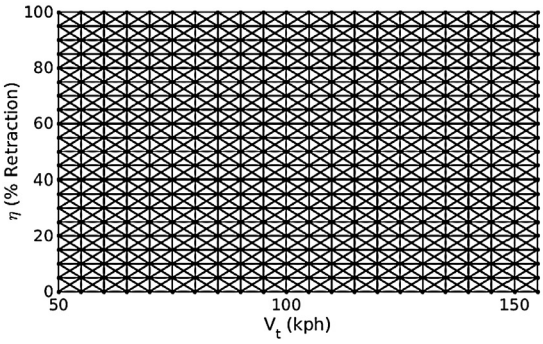
(b) $50 \leq V_t \leq 155$ kph with $\Delta V_t = 5$ kph & $-25 \leq \Delta X_{CoG} \leq 25\%\bar{c}$ with $\Delta(\Delta X_{CoG}) = 5\%\bar{c}$



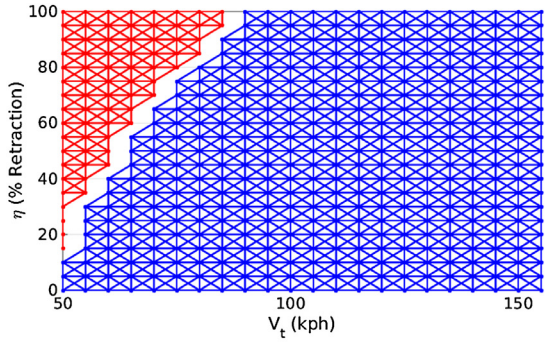
(c) $0 \leq \eta \leq 100\%$ with $\Delta\eta = 5\%$ & $-25 \leq \Delta X_{CoG} \leq 25\%\bar{c}$ with $\Delta(\Delta X_{CoG}) = 5\%\bar{c}$

Fig. 9. Sorted Short Period mode varied over the 2 parameters for 3 parameter combinations.

Please cite this article in press as: C.S. Beaverstock et al., Automatic mode tracking for flight dynamic analysis using a spanning algorithm, Aerosp. Sci. Technol. (2015), http://dx.doi.org/10.1016/j.ast.2015.08.013



(a) Example connectivity of the Dutch Roll showing a complete graph



(b) Example connectivity of the roll convergence mode showing an encounter with the Spiral mode to give a roll-spiral complex mode

Fig. 10. Graph of modal connectivity in two-dimensions.

Table 4
Summary of run information for varying parameter pairs for the mode behaviour over 2-dimensions (see Fig. 8).

Parameter	No. of modes	No. of cycles	Time to completion (s)
V_t and η	20	7	29.22
V_t and ΔX_{CoG}	23	9	13.62
η and ΔX_{CoG}	16	4	9.94

Typically, a mode does not span the entire space as the mode modifies its behaviour between aperiodic and harmonic. On occasion, the algorithm failed to span the space as two comparable modes from one flight condition were similar to one from another condition, leading to one linked and one unlinked mode. This was typically associated to aperiodic modes. Additional strategies to avoid this can be as follows:

- For each cycle of spanning a mode, each condition is compared only once to an already identified condition. An additional loop is required to ensure that where a disjoint appears, that the disjointed condition is compared to all of its connected nodes that have identified the mode.
- Where modes are similar, with MAC values that are close, these should be tested for modal veering, to ensure the correct mode is tracked, following two similar modes interacting.

It is suggested that following completion of the algorithm, the finite number of connected regions should be investigated to form disjointed sets of modes that are potentially related. Currently the algorithm is only concerned with spanning a connected set of nodes, and disjointed sets should be connected through analysis by the user.

In the final run, all parameters are varied simultaneously. The full modal Argand diagram is presented in Fig. 11 over three-dimensions. Each 2-D Argand plane represents variation over velocity and span retraction parameter, where the third dimension is used to represent variation in the CoG position.

Fig. 11 illustrates that in excess of 40 modes are identified. The Short Period and Dutch roll are present in all conditions, Roll subsidence and Spiral are present in most, and the phugoid in a more limited number of operating points in the parameter space.

Other modes are formed when a complex conjugate pair, for example, the phugoid mode, encounters the real axis, and splits into two real longitudinal modes. Or two real modes, for example, the roll convergence and spiral, encounter one another and combine into a complex conjugate pair. On inspection, some of these

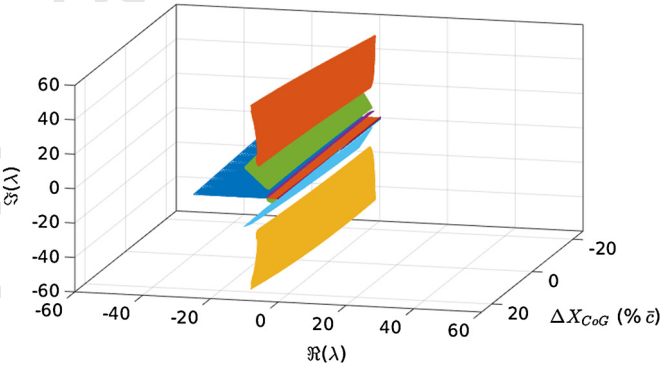


Fig. 11. Argand diagram varying parameters over 3 dimensions, 3^{rd} dimension represents ΔX_{CoG}.

Table 5
Summary of run information over three-dimensions.

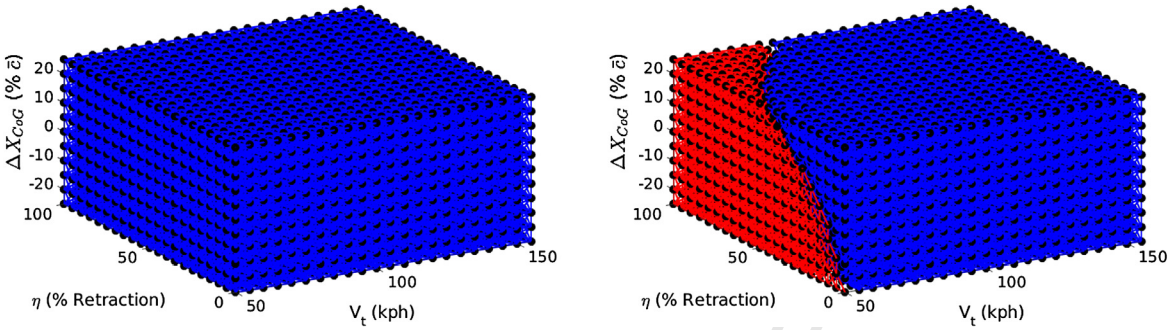
Parameter	No. of modes	No. of cycles	Time to completion (s)
V_t, η and ΔX_{CoG}	42	7	15,978

Table 6
Cycle time summary for three-dimensional case.

Cycle index	No. of modes	Time to complete
1	12	7682 s
2	4	6566 s
3–28	1	14 cycles at 10–15 s/cycle 5 cycles at 15–20 s/cycle 2 cycles at 20–30 s/cycle 4 cycles at >60 s/cycle

minor modal groups should actually be considered as members of a larger primary group. These have been disconnected as the condition that they were tested at is similar to another mode that has already been identified. Changing the initial condition, alters the connectivity of some of these modes. However, analysing many of these modes, some of these modes are only present in a handful of conditions. A summary of the run information is presented in Table 5.

As can be observed, with increased dimensions, there is an increase in the time to solve the entire space, and also the time for each cycle. Table 6 provides a breakdown of a cycle and the time to solution for sections of the pseudo code.



(a) Example connectivity of mode presented in Figure 10a extended to three-dimensions

(b) Example connectivity of mode presented in Figure 10b extended to three-dimensions

Fig. 12. Graph of modal connectivity in three-dimensions. (For interpretation of the references to colour in this figure, the reader is referred to the web version of this article.)

Table 7
Summary of model properties of identified primary mode sets.

Mode	Frequency (Hz)		Damping ζ	
	min	max	min	max
Short period	0.5	4	0.3	0.6
Phugoid	0.01	0.1	−0.2	−1
Dutch roll	2	9	0	0.35

This reveals the complexity of the code, and demonstrates how the complexity scales with the size of a problem, and the number of dimensions of the problem. The addition of a third dimension results in a greatly increased time to solution. From this table, firstly the maximum time to complete a cycle increases by more than two orders of magnitude. For this problem, the increase in dimensions from two to three results in 8 node comparisons in two dimensions to 26 in three dimensions. This increase is directly related to the increase in time to compute. Cycle 2 shows that although only a third of the modes are being computed, the time taken for the solution is the same order of magnitude. The algorithm runs in parallel, with a maximum of 12 active modes tracked. The algorithm requires that all of these modes be from the same node. Therefore, when there are fewer than 12 modes available in the node under investigation, or a processor finishes its work before other processors, the result is idle time for some processors. This represents an inefficiency in the parallelisation of the problem, and requires investigation. Fig. 12 presents a polytope that shows the connectivity of a complete and an incomplete graph.

For the incomplete graph, this could be a member of a disjointed set of similar modes, or in some cases, a member of a connected group, where the algorithm has failed to identify the connecting edge using the MAC parameter. In the context of this example, the complete graph represents the Short Period or Dutch roll, which shows that a mode in every node is connected to a mode from at least one of the other connected nodes. Fig. 12b presents an example where a mode does not span all conditions, where the blue points represent nodes where both a spiral and roll convergence mode have been identified, and the red points are where the mode has become a complex conjugate pair that represent a roll-spiral mode.

6.1.2. Modal properties (characteristics and shapes)

Figs. 13 to 17 show the eigenvalue characteristics are shown, namely the frequency and damping for the identifiable flight modes. Tables 7 and 8 present a summary of the information.

The Short Period, which spans the parameter space, shows generally that at high speed, maximum span and CoG aft position (unstable) the frequency is highest. The damping ratio is at its

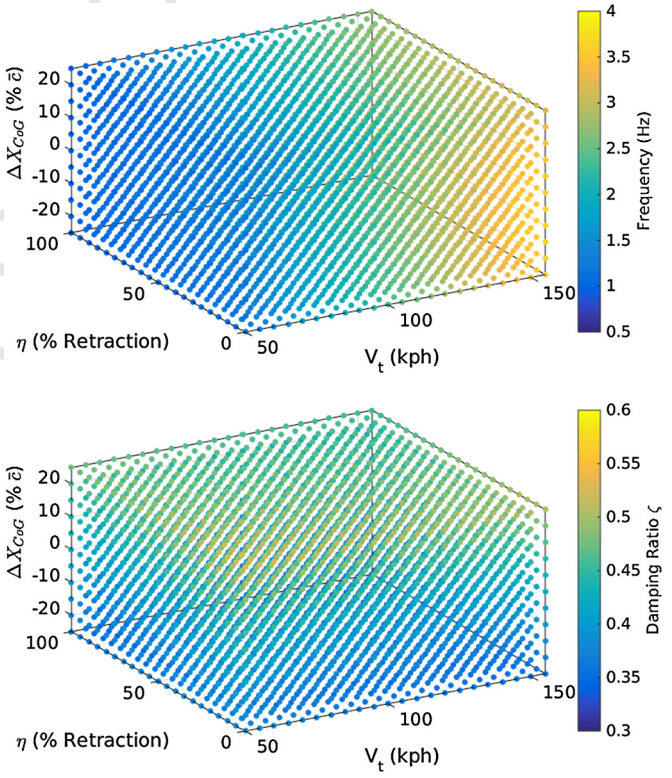


Fig. 13. Short period modal characteristics. (For interpretation of the colours in this figure, the reader is referred to the web version of this article.)

greatest when the span is at its maximum, and the CoG at its most fore position (stable). The damping appears to be relatively insensitive to variations in flight speed.

When the Short Period spans the entire parameter space, the other longitudinal complex conjugate pair associated with the phugoid only spans a limited portion of the parameter space. The mode only appears present in the unstable CoG range, upon which the damping indicates that the mode encounters the real axis, changing its behaviour to two real modes. The mode is unstable throughout, the frequency varying from 0.01 Hz at low speed, increasing one order of magnitude to approximately 0.1 Hz at high speed. The damping indicates that it is sensitive to all three parameters, where its sensitivity to each parameter is coupled to the others.

The complex lateral mode, the Dutch Roll, like the Short Period, spans the parameter space. According to Fig. 15 the frequency is dominated by its relationship to flight speed, varying from 2 Hz to

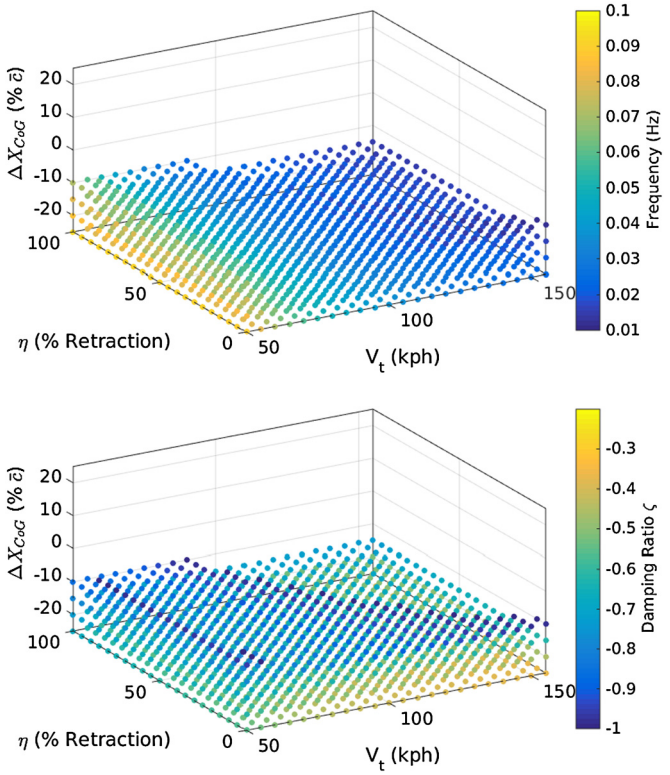


Fig. 14. Phugoid modal characteristics. (For interpretation of the colours in this figure, the reader is referred to the web version of this article.)

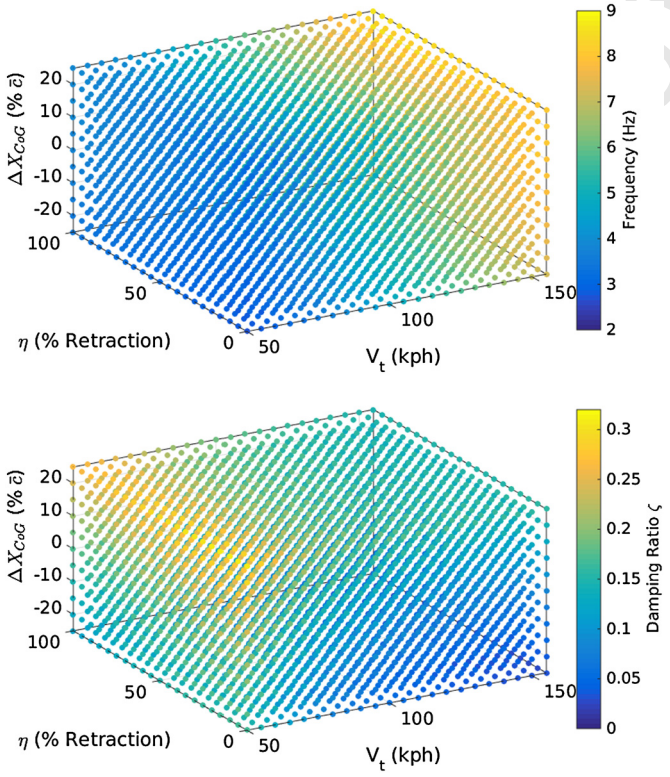


Fig. 15. Dutch Roll modal characteristics. (For interpretation of the colours in this figure, the reader is referred to the web version of this article.)

9 Hz when increasing the speed. The mode damping is primarily dependent on the speed and the CoG position, and is minimum for high speed and unstable CoG position.

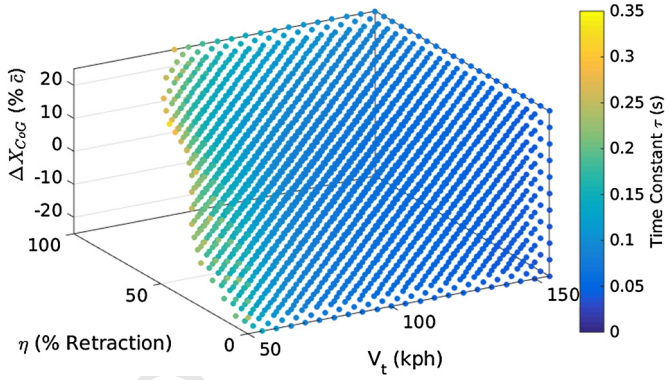


Fig. 16. Roll subsidence/convergence modal characteristics. (For interpretation of the colours in this figure, the reader is referred to the web version of this article.)

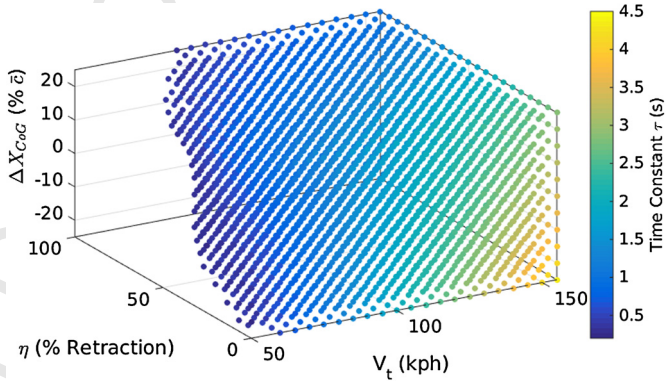


Fig. 17. Spiral modal characteristics. (For interpretation of the colours in this figure, the reader is referred to the web version of this article.)

Table 8
Summary of model properties of identified primary mode sets.

Mode	Time constant (s)	
	min	max
Roll convergence	0	0.35
Spiral	0.3	4.5

The two real lateral modes, Roll Subsidence (Fig. 16) and Spiral (Fig. 17) are present throughout the speed and CoG range. Retracting the span at low speeds and unstable CoG positions causes these two modes to form a complex conjugate pair. The figures indicate at high speed and unstable CoG position, the spiral time constant is at its highest, and the Roll subsidence is at its lowest.

In summary of these modal characteristics, this data represents the characteristics related to a root locus of the modes in a parameter space of three dimensions. The interactions shown in these figures show the effect of varying parameters on the dynamic characteristics, and the potential interactions and modifications that can occur in a dynamic system. This shows that traditional idealised models of dynamic systems are modified and even breakdown within a multi-dimensional parameter space. With respect to the algorithm, the nodes are connected by their adjacency to modes from a neighbouring system, which are essentially autonomous. With simple deployment of the MAC parameter, relationships of modes in a multidimensional space can quickly be related for further analysis.

One problem that is highlighted by this investigation is the scaling problem. In this physical example, the eigenvector is composed of both translational and rotational state parameters. These can be angles, linear translational displacements and rates. As the

Table 9
Summary of cycle run times.

Test case	No. of nodes	No. of comparisons	Total cycle time (s)	Time per comparison (s)	Time per comparison per node (s)
1st Dim	22	21	7.4397×10^{-2}	3.5427×10^{-3}	1.6103×10^{-4}
2nd Dim	21	20	7.3151×10^{-2}	3.6576×10^{-3}	1.7417×10^{-4}
3rd Dim	11	10	5.8274×10^{-2}	5.8274×10^{-3}	5.2976×10^{-4}
1st Dim and 2nd Dim	$22 \times 21 = 462$	1721	2.3129×10^{1}	1.3439×10^{-2}	2.9089×10^{-5}
1st Dim and 3rd Dim	$22 \times 11 = 242$	871	6.0365×10^{0}	6.9305×10^{-3}	2.8638×10^{-5}
2nd Dim and 3rd Dim	$21 \times 11 = 231$	830	5.6725×10^{0}	6.8343×10^{-3}	2.9586×10^{-5}
(1st Dim, 2nd Dim and 3rd Dim)	$22 \times 21 \times 11 = 5082$	57,971	7.6824×10^{3}	1.3252×10^{-1}	2.6077×10^{-5}

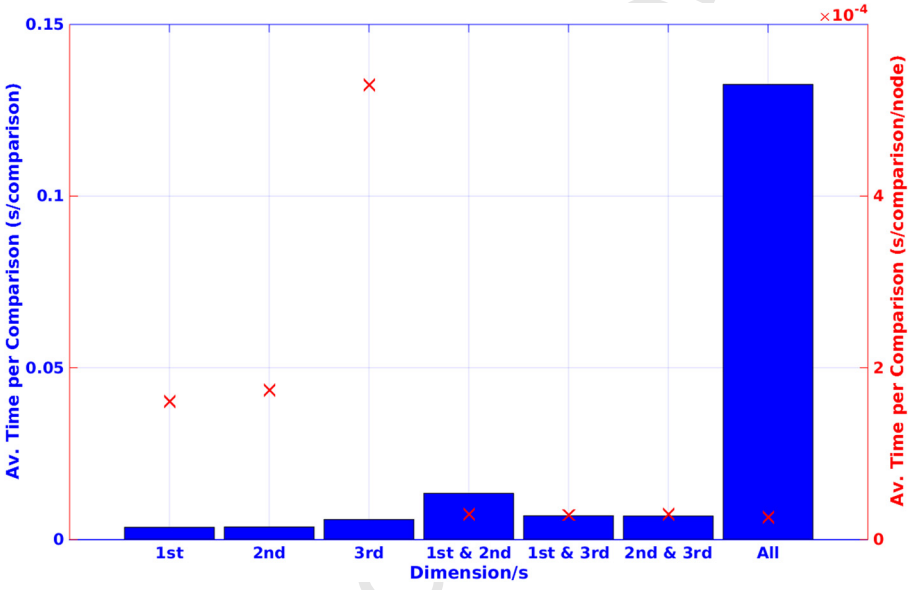


Fig. 18. Average timings on initial cycle per comparison and per node. (For interpretation of the references to colour in this figure, the reader is referred to the web version of this article.)

MAC parameter is dependent on the relative magnitude and angle in the eigenspace, dominant state parameters will affect the magnitude, in addition to the ability to reliably track a mode. Additionally, comparing systems with comparable dynamic structure and modal response, but with significantly different systems configuration (larger variation in mass, speed or flight configuration) can be difficult to compare two similar modes, due to the different interacting scales of a dimensional eigenvector. This may require the eigenvector to be re-scaled, and non-dimensionalised to normalise the result. Modifying the systems description to a mass-stiffness (**M** and **K**) representation can be used to normalise the result, such that the scale of the states in the eigenvector are comparable.

6.2. Performance

Deploying this practically by observing the performance over 1-, 2- and 3-dimensions (Tables 3, 4 and 5) shows that increasing the dimensionality of the data-set has a more significant effect than the number of flight conditions. Therefore, the greater the dimensional connectivity at a node, the more comparisons have to be interrogated, and so increases the time spent interrogating a node. The algorithm tested using MATLAB provides a practical solution for a 22 by 21 by 11 grid of points, in 1-, 2- and 3-dimensions within a few hours. It is expected that the algorithm deployed more efficiently could yield an order of magnitude improvement, enabling larger grids and more dimensions to be feasible.

Table 9 presents a summary of the timing data for the first cycle of each test case. The first 3 test cases are a single dimension, followed by 3 two dimensional runs and the final run is performed over all three dimensions. The table shows the total number of node-to-node comparisons, total number of nodes, total time to complete the cycle, average time per comparison and the average time per comparison normalised by the number of nodes.

The results show that the time to complete increases with both the number of nodes as well as the number of comparisons. The number of comparisons is a function of the number of nodes in addition to the degree of each node to determine the number of unique node-to-node comparisons. For a structured uniform grid, Fig. 2 showed that the maximum number of connected nodes connected to a single node increases from 2 to 26 as the dimensions are increased from one to three. The effect of this can be observed with the number of comparisons, where the factor by which the number of comparisons increases is greater than the equivalent factor of increase in number of nodes. Fig. 18 presents the average time taken per comparison (blue bars), and the average time taken per comparison normalised to the number of nodes in the space (red crosses).

The two and three dimensional test cases in Fig. 18 (supported by the numerical results presented in Table 9) shows that the average time per comparison normalised by the number of nodes is constant. This implies that the maximum time to complete a single cycle is the product of the number of nodes and the number of comparisons, multiplied by a constant time factor. Fig. 18 also shows the effect of dimensionality, where time per compari-

Please cite this article in press as: C.S. Beaverstock et al., Automatic mode tracking for flight dynamic analysis using a spanning algorithm, Aerosp. Sci. Technol. (2015), http://dx.doi.org/10.1016/j.ast.2015.08.013

son increases rapidly when progressing from the two to the three dimensional problem. The one dimensional test cases contradict these assertions. The reason for this is explained by an initial overhead which is used to initiate each cycle. Because the initial overhead is of the same order of magnitude as the one dimensional test cases, this results in greater average time per comparison, and average time per comparison normalised by number of nodes. This is also supported by these times increasing with decreasing number of nodes.

Parallelisation for the comparisons of system eigenstructures to compute the MAC parameter was quick, where all comparisons for a 3 dimensional parameter space were completed within 10 seconds (for 5000+ comparisons). This was largely due to the simple parallelisation of the problem, which enabled each processor to be used for the entire duration of the loop. For the spanning algorithm, the definition of the problem restricts the utilisation of the processors. Because each processor can only span a single mode, and this mode must be from the seed node that spawned the mode, a processor is only being utilised if the mode is still active. If a processor becomes inactive whilst other modes are still being processed, this processor will remain inactive until a new flight condition is investigated. Furthermore, if there are fewer modes than processors available, a number of processors will be inactive for that run. It is suggested that for improvement in the total time to solution, modifications to the parallelisation is necessary that utilise all processors throughout the run, reducing the total idle time of any single processor, for example in searching for the connectivity of nodes.

7. Conclusions and recommendations

In this paper, integration of the MAC parameter within a spanning algorithm, applied to a flight dynamics problem was used to track dynamic multiple modes in 1-, 2- and 3-dimensional parameter space. Although the algorithm was only demonstrated for up to 3 dimensions, its application can be extended to automatically track modes over N-dimensions. It was shown that the complexity is affected by both the number of nodes/flight conditions to investigate, the number of dimensions these are distributed over and the number of modes found. It was shown that for the problem presented in this paper, that the time to solution was primarily dependent on the number of dimensions due to the functional relationship with the algorithm complexity. This meant using the algorithm for increasing dimensions resulted in a much greater run time. The test case had 12 modes, and so utilisation of the computational processing power was maximum while all of these solved over the entire parameter space. However, where fewer than 12 modes (and thus processors) were being solved, or where some modes did not span the entire space, results in larger idle time of computational resources, and increasing number of cycles to solve the entire space. It is suggested that an alternative spanning algorithm, which enables all processors to be utilised, and to minimise the potential of processors be idle, should be implemented. This may involve operating processors to solve single subspaces, where the solutions are then stitched. Another problem highlighted with the spanning algorithm used is that each node is only investigated with one of its neighbours. This leads to possibly connected mode spaces not being found due to failure in the connecting edges MAC comparison. Two suggested methods are either: 1) re-investigate failed connected nodes, re-running the algorithm if a connection can be established, 2) or to compare border nodes of completed identified sets at connected points.

The MAC was computed using raw, dimensional eigenvectors. As such, scaling between different states in the eigenvector can lead to the MAC deriving its "similarity" based on the most significant components numerically, with no real physical interpretation imposed. In reality, these eigenvectors are composed of variables with varying scales, and units used. Therefore, it is suggested to ensure proper handling of modal comparisons, that possible non-dimensionalising and normalisation of this vector is performed, such that the MAC is representative of the relative dynamic state. Such an approach allows comparisons of the dynamic state for aircraft of varying scales, for example, comparing the dynamic spectrum of a large and relatively fast aircraft, to that of a small UAV scale aircraft, such that a "Short Period" at one scale can be found at another.

Future work with this algorithm should focus on two primary aspects; 1) Reducing the complexity, or the effect of dimensionality of the problem on the time to solution for a single processor. The current algorithm can solve a mode across the full span of the space in hours for a three dimensional problem, where it takes less than a minute across two. Reducing the effect of dimensionality on the algorithm complexity, will enable the algorithm to extend feasible its range of application to problems greater than 3 dimensions. 2) Modifying the parallelisation approach to ensure maximum utilisation of the computational resources, and minimise processor idle time.

Conflict of interest statement

None declared.

Acknowledgements

The work presented herein has been partially funded by the European Community's Seventh Framework Programme (FP7) under Grant Agreement 314139, and the European Research Council under the European Union's Seventh Framework Programme (FP/2007–2013)/ERC Grant Agreement No. [247045]. The CHANGE project (Combined morphing assessment software using flight envelope data and mission based morphing prototype wing development) is a Level 1 project funded under the topic AAT.2012.1.1-2 involving 9 partners. The project started on August 1st 2012.

Appendix A. Spanning algorithm pseudo-code

Algorithm 1 Spanning algorithm: main algorithm.

```
function SpanningAlgol
    while Modes Unassigned do
        SeedMode()
        TrackMode()
    end while
end function
```

Algorithm 2 Spanning algorithm: seed mode.

```
function SeedMode
    for i = 1 : noOfFltCons do                          ▷ Go through flight conditions
        for j = 1 : noOfModes do ▷ Go through modes in flight condition index i
            if modeIdxFlag(i, j) = false then           ▷ Test if mode j in flight condition i has been indexed
                unModeIdx = unModeIdx + 1               ▷ Incrament mode index by 1
                modeIdx(i, j) = unModeIdx               ▷ Assign mode index to unassigned mode
                modeIdxFlag(i, j) = true                ▷ Flag mode as assigned
                ConnectCons(i, cmpModeIdx, openForInspFlag)  ▷ Locate connected mode in other flight conditions
            end if
        end for
    end for
end function
```

Algorithm 3 Spanning algorithm: connect conditions.

```
function ConnectCons(i,cmpModeIdx,openForInspFlag)
    for m = 1 : noOfFltCons do                ▷ Search flight conditions for connected
        if connIdxFlag(i, m) = true & connRunFlag(m) = false then   ▷ Test if flight conditions are connected and if flight conditions has already bee run for this instance
            openForInspFlag(m) = true          ▷ Set conditions as open for inspection
            cmpModeIdx(m, 1) = i               ▷ Give position of condition to compare flight condition m with
            cmpModeIdx(m, 2) = j               ▷ Give mode index j in position i to compare modes in flight condition m
        end if
    end for
    return cmpModeIdx, openForInspFlag
end function
```

Algorithm 4 Spanning algorithm: track mode.

```
function TrackMode
    while openForInspFlag(:) = true do ▷ While at least one condition is open for inspection keep looping
        for i = 1 : noOfFltCons do             ▷ For all conditions test if open and then perform model comparison
            if openForInspFlag(i) = true then  ▷ Test if condition i is open for inspection
                cmpModeIdx(m, 1)
                cmpModeIdx(m, 2)
                CompareMAC(i),                 ▷ If open then compare condition i with the condition that was connected to it
                openForInspFlag(i) = false     ▷ Close mode for inspection
                connRunFlag(i) = true          ▷ Mark condition as run
            end if
        end for
    end while
end function
```

Algorithm 5 Spanning algorithm: compare MAC.

```
function CompareMAC(i, m, n)
    for j = 1 : noOfModes do
        if MACIdxCon(i, m)(j, n) = true then
            modeIdx(i, j) = unModeIdx
            modeIdxFlag(i, j) = true
            ConnectCons(i, cmpModeIdx, openForInspFlag)
        end if
    end for
end function
```

Appendix B. Supplementary material

Supplementary material related to this article can be found online at http://dx.doi.org/10.1016/j.ast.2015.08.013.

References

[1] R.J. Allemang, The modal assurance criterion – twenty years of use and abuse, J. Sound Vib. (2003).
[2] E.L. Allgower, K. Georg, Introduction to Numerical Continuation Methods, Class. Appl. Math., Society for Industrial and Applied Mathematics, 1987.
[3] D. Amsallem, C. Farhat, An online method for interpolating linear parametric reduced-order models, SIAM J. Sci. Comput. (2011).
[4] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J.D. Croz, A. Greenbaum, S. Hammarling, A. McKenney, D. Sorensen, Lapack user's guide, 1999.
[5] C. Beaverstock, R. Ajaj, M. Friswell, R. deBreuker, N. Wetter, Optimising mission performance for a morphing MAV, in: Ankara International Aerospace Conference, 2013.
[6] C. Beaverstock, R. Ajaj, M. Friswell, R. deBreuker, N. Wetter, Effect of span morphing on the symmetric & asymmetric flight dynamics, in: AIAA SciTech Conference, Maryland, 2014.
[7] C. Beaverstock, R. Ajaj, M. Friswell, W. Dettmer, R. deBreuker, N. Wetter, Effect of span-morphing on the flight modes, stability & control, in: AIAA Guidence Navigation and Control Conference, Boston, 2013.
[8] A. Chandrashaker, S. Adhikari, M.I. Friswell, Quantification of vibration localization in periodic structures, J. Vib. Acoust. (2015), submitted for publication.
[9] M.V. Cook, Flight Dynamics Principles: A Linear Systems Approach to Aircraft Stability and Control, 3rd edition, Butterworth–Heinemann, 2013.
[10] R.C. Dorf, R.H. Bishop, Modern Control Systems, Pearson, 2004.
[11] J.L. du Bois, S. Adhikari, N.A.J. Lieven, On the quantification of eigenvalue curve veering: a veering index, J. Appl. Mech. (2011).
[12] B. Etkin, L.D. Reid, Dynamics of Flight: Stability and Control, 3rd edition, John Wiley & Sons, 1995.
[13] G. Golub, C.V. Loan, Matrix Computations, 3rd edition, Johns Hopkins University Press, 1996.
[14] T.A. Johansen, T. Fossen, Control allocation – a survey, Automatica 49 (5) (2013) 1087–1103.
[15] D. Jungnickel, Graphs, Networks and Algorithms, 4th edition, Springer, 2012.
[16] G.P. Liu, R.J. Patton, Eigenstructure Assignment for Control System Design, Wiley–Blackwell, 1998.
[17] B. McCormic, Aerodynamics, Aeronautics, and Flight Mechanics, John Wiley & Sons, 1995.
[18] L.D. Mitchell, Increasing the sensitivity of the modal assurance criteria to small mode shape changes, in: International Modal Analysis Conference, 2001, pp. 64–69.
[19] R. Nelson, Flight Stability and Automatic Control, 2nd edition, McGraw–Hill Education, 1997.
[20] M. Pastor, M. Binda, T. Harčarik, Modal assurance criterion, Proc. Eng. 48 (2012) 543–548.
[21] C. Schedlinski, F. Wagner, K. Bohnert, J. Frappier, A. Irrgang, R. Lehmann, A. Müller, Experimental modal analysis and computational model updating of a car body in white, in: ISMA2004 International Conference on Noise and Vibration Engineering, 2004.
[22] L.V. Schmidt, Introduction to Aircraft Flight Dynamics, American Institute of Aeronautics and Astronautics, Inc., 1998.
[23] R.F. Stengel, Flight Dynamics, Princeton University Press, 2004.
[24] G. Strang, Introduction to Linear Algebra, Wellesley–Cambridge Press, 2009.
[25] J. Wilkie, M. Johnson, R. Katebi, Control Engineering, Palgrave Macmillan, 2001.
[26] R.L. Williams II, D.A. Lawrence, Linear State-Space Control Systems, John Wiley & Sons, 2007.
[27] V. Yaghoubi, T. Abrahamsson, Automated modal analysis based on statistical evaluation of frequency responses, in: Topics in Modal Analysis: Proceedings of the 31st IMAC, A Conference on Structural Dynamics, in: Conf. Proc. Soc. Exp. Mech. Ser., vol. 7, Springer, New York, 2014.
[28] L.A. Zadeh, C.A. Desoer, Linear System Theory: The State Space Approach, Dover Civ. Mech. Eng., Dover Publications, 2008.