ASWING 5.99 Technical Description — Unsteady Extension

Mark Drela

10 March 2015

Summary

This addendum describes the additional variables and unsteady loads which appear when full dynamics are considered in ASWING. Other extensions include an unsteady liftting-line model and a structural damping model. A general unsteady atmospheric gust velocity field is also considered.

The governing discrete equations for the extended model form a system of coupled 1st-order ODE's in time. These are numerically solved via implicit time marching. Both 1st-order and 2nd-order accurate time discretizations are considered.

In addition to the nonlinear time marching, linearized unsteady analyses are derived, both for the forced-response case (Bode analysis), and for the natural-response case (Eigenmode analysis). A Reduced Order Model (ROM) is constructed using the eigenmodes as a basis, for the purpose of approximate but very fast forced-response calculations, and for control-law synthesis.

Contents

1 Unsteady State Variables

To treat the general case with unsteady deformations, the structural state vector $\vec{r}_i, \vec{\theta}_i, \vec{M}_i, \vec{F}_i$ is now augmented with the local rates $\vec{u}_i$ and $\vec{\omega}_i$. These are the linear and angular velocities of the local csn beam axis system relative to the xyz body axes, as shown in Figure 1. The augmenting equations are then simply the rate definitions with the appropriate transformations included.

$$\dot{\vec{r}}_i = \vec{u}_i \tag{1}$$

$$\dot{\vec{\theta}}_i = \bar{\bar{C}}\,\vec{\omega}_i \tag{2}$$

$$\bar{\bar{C}} = \left[\bar{\bar{T}}^T \bar{\bar{K}}\right]^{-1} = \begin{bmatrix} 1 & \cos\varphi\,\tan\psi & \sin\varphi\,\tan\psi \\ 0 & \cos\varphi/\cos\psi & \sin\varphi/\cos\psi \\ 0 & -\sin\varphi & \cos\varphi \end{bmatrix} \qquad \text{(surface beam)} \tag{3}$$

$$\bar{\bar{C}} = \left[\bar{\bar{T}}^T \bar{\bar{K}}\right]^{-1} = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi/\cos\varphi & \cos\psi/\cos\varphi & 0 \\ \sin\psi\,\tan\varphi & -\cos\psi\,\tan\varphi & 1 \end{bmatrix} \qquad \text{(fuselage beam)} \tag{4}$$

The overall state vector is also augmented with the aircraft-frame position $\vec{R}$ in inertial space, and the aircraft-frame orientation Euler angles Φ, Θ, Ψ relative to inertial space.

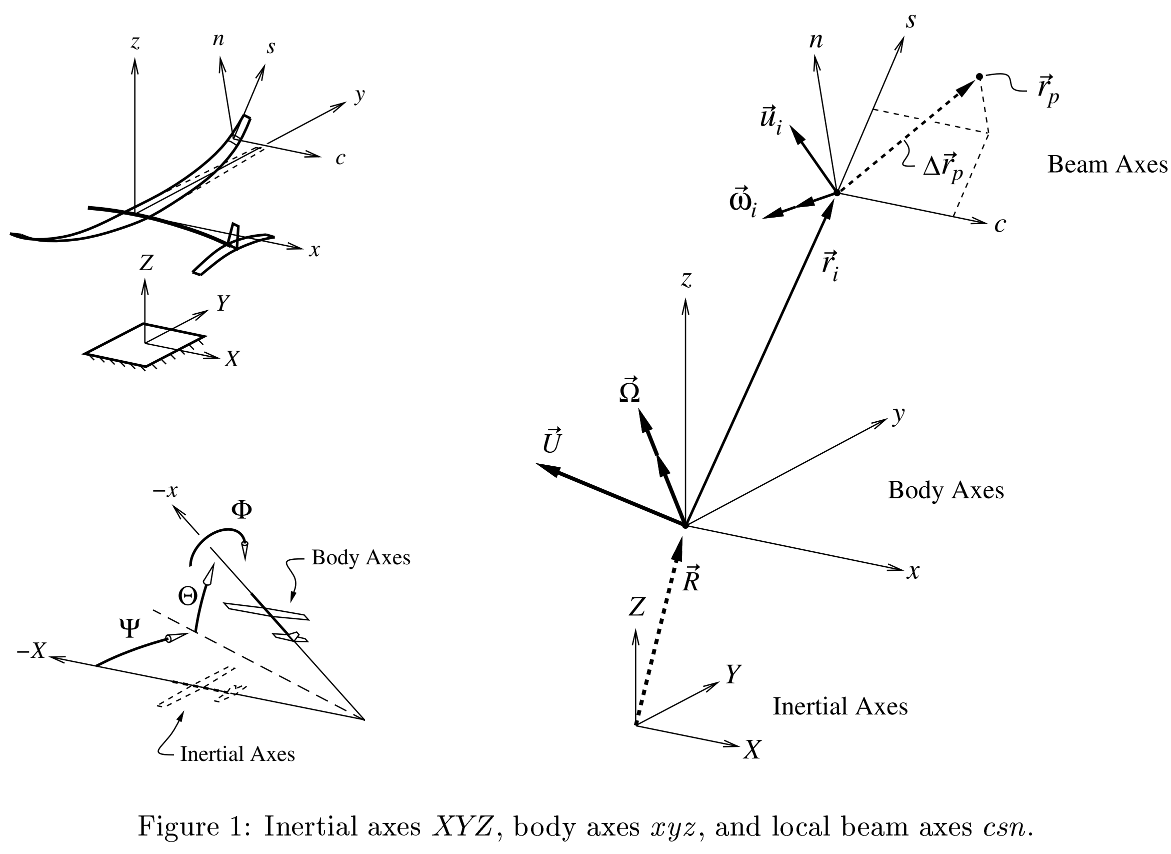

Figure 1: Inertial axes XYZ, body axes xyz, and local beam axes csn.

2 Frame Kinematics

A hierarchy of three axis systems, shown in Figure 1, is used to describe the overall aircraft motions and deformations. The relative positions and motions of the axis-system origins are defined by the following quantities.

$\vec{R}$	xyz position relative to XYZ	$\vec{r}_i$	csn_i position relative to xyz
$\vec{U}$	xyz velocity relative to XYZ	$\vec{u}_i$	csn_i velocity relative to xyz
$\vec{\Omega}$	xyz rotation rate relative to XYZ	$\vec{\omega}_i$	csn_i rotation rate relative to xyz
$\vec{\Theta}$	xyz Euler angles relative to XYZ	$\vec{\theta}_i$	csn_i Euler angles relative to xyz

Note that there is only one XYZ and one xyz system, but a multitude of csn_i systems, one for each beam node i.

The Euler angle triplet $\vec{\Theta} = \{\Phi, \Theta, \Psi\}^T$ is not a proper vector, and the symbol $\vec{\Theta}$ will be used only as a convenient terse notation. The following quantities are also convenient to describe local beam-section dynamics.

$\vec{r}_p$	a point fixed to local csn system
$\Delta\vec{r}_p$	offset of point relative to csn origin

All vectors are normally expressed in terms of their xyz body-axes components. The $(\;)_E$ subscript denotes a vector given in terms of the "Earth" inertial axes XYZ. The component transformation of any vector $\vec{A}$ from XYZ to xyz is via the three standard aircraft Euler angles, applied in the sequence $-\Psi$, Θ, $-\Phi$ about z, y, x, respectively. The negated rotations account for the fact that Ψ and Φ are rotations about the standard axes [1] which have x and z reversed from the present system. Figure 1 shows the rotation directions. The inverse transformation of any vector $\vec{A}$ from xyz to XYZ is then defined via the backward sequence Φ, $-\Theta$, Ψ.

$$\left\{ \begin{array}{c} A_X \\ A_Y \\ A_Z \end{array} \right\} = \left[\quad \bar{\bar{T}}_E \quad \right] \left\{ \begin{array}{c} A_x \\ A_y \\ A_z \end{array} \right\} \tag{5}$$

$$\bar{\bar{T}}_E = \begin{bmatrix} \cos\Psi & \sin\Psi & 0 \\ -\sin\Psi & \cos\Psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\Theta & 0 & \sin\Theta \\ 0 & 1 & 0 \\ -\sin\Theta & 0 & \cos\Theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\Phi & \sin\Phi \\ 0 & -\sin\Phi & \cos\Phi \end{bmatrix} \tag{6}$$

$$\bar{\bar{T}}_E = \begin{bmatrix} \cos\Theta\cos\Psi & -\sin\Phi\sin\Theta\cos\Psi + \cos\Phi\sin\Psi & \cos\Phi\sin\Theta\cos\Psi + \sin\Phi\sin\Psi \\ -\cos\Theta\sin\Psi & \sin\Phi\sin\Theta\sin\Psi + \cos\Phi\cos\Psi & -\cos\Phi\sin\Theta\sin\Psi + \sin\Phi\cos\Psi \\ -\sin\Theta & -\sin\Phi\cos\Theta & \cos\Phi\cos\Theta \end{bmatrix} \tag{7}$$

The aircraft rotation rate relative to the inertial axes is related to the time rate of change of the transformation tensor by

$$\dot{\bar{\bar{T}}}_E = \bar{\bar{\Omega}}_E \bar{\bar{T}}_E \quad \longrightarrow \quad \bar{\bar{\Omega}}_E \equiv \begin{bmatrix} 0 & -\Omega_Z & \Omega_Y \\ \Omega_Z & 0 & -\Omega_X \\ -\Omega_Y & \Omega_X & 0 \end{bmatrix} = \dot{\bar{\bar{T}}}_E \bar{\bar{T}}_E^T \tag{8}$$

which gives explicit expressions for the three rotation rate components.

$$\left\{ \begin{array}{c} \Omega_X \\ \Omega_Y \\ \Omega_Z \end{array} \right\} = \left[\quad \bar{\bar{K}}_E \quad \right] \left\{ \begin{array}{c} \dot{\Phi} \\ \dot{\Theta} \\ \dot{\Psi} \end{array} \right\} \tag{9}$$

$$\bar{\bar{K}}_E = \left[\begin{array}{ccc} -\cos\Psi \, \cos\Theta & \sin\Psi & 0 \\ \sin\Psi \, \cos\Theta & \cos\Psi & 0 \\ \sin\Theta & 0 & -1 \end{array} \right] \tag{10}$$

The rotation rate components in the body axes are then related to the Euler angle rates as follows.

$$\left\{ \begin{array}{c} \Omega_x \\ \Omega_y \\ \Omega_z \end{array} \right\} = \left[\quad \bar{\bar{T}}_E^T \quad \right] \left\{ \begin{array}{c} \Omega_X \\ \Omega_Y \\ \Omega_Z \end{array} \right\} = \left[\quad \bar{\bar{T}}_E^T \quad \right] \left[\quad \bar{\bar{K}}_E \quad \right] \left\{ \begin{array}{c} \dot{\Phi} \\ \dot{\Theta} \\ \dot{\Psi} \end{array} \right\} \tag{11}$$

$$\left\{ \begin{array}{c} \Omega_x \\ \Omega_y \\ \Omega_z \end{array} \right\} = \left[\begin{array}{ccc} -1 & 0 & \sin\Theta \\ 0 & \cos\Phi & \sin\Phi \, \cos\Theta \\ 0 & \sin\Phi & -\cos\Phi \, \cos\Theta \end{array} \right] \left\{ \begin{array}{c} \dot{\Phi} \\ \dot{\Theta} \\ \dot{\Psi} \end{array} \right\} \tag{12}$$

3 Trajectory Equations

In the quasi-steady formulation, the aircraft position $\vec{R}_E$ and orientation $\vec{\Theta}$ are prescribed explicitly. In the unsteady case, $\vec{R}_E$ evolves via its rate equation.

$$\frac{d\vec{R}_E}{dt} - \bar{\bar{T}}_E \vec{U} = 0 \tag{13}$$

Note that $\vec{R}_E(t)$ does not directly couple to the aircraft structural dynamics. It is needed only to interrogate an imposed gust velocity field $\vec{V}_{E_{\rm gust}}(\vec{r}_E)$ implemented in the user-supplied subroutine `VGUSTE`, and of course it conveys trajectory information. The rate equation residual for the Euler angles is simply the inverse of equation (12).

$$\frac{d\vec{\Theta}}{dt} - \bar{\bar{C}}_E \vec{\Omega} = 0 \tag{14}$$

$$\bar{\bar{C}}_E = \left[\bar{\bar{T}}_E^T \bar{\bar{K}}_E \right]^{-1} = \left[\begin{array}{ccc} -1 & \sin\Phi \, \tan\Theta & -\cos\Phi \, \tan\Theta \\ 0 & \cos\Phi & \sin\Phi \\ 0 & \sin\Phi / \cos\Theta & -\cos\Phi / \cos\Theta \end{array} \right] \tag{15}$$

Normally only Φ and Θ couple to the structural dynamics by their influence on the gravity vector $\vec{g}$ in body axes. If a gust velocity field $\vec{V}_{E_{\rm gust}}(\vec{r}_E)$ is being imposed, then all three Euler angles will come into play via the necessary transformation $\vec{R}_E, \vec{r} \rightarrow \vec{r}_E$.

The aircraft-frame velocity $\vec{U}$ and rotation rate $\vec{\Omega}$ evolve according to their kinematic rate equations.

$$\frac{d\vec{U}}{dt} + \vec{\Omega} \times \vec{U} - \vec{a}_o = 0 \tag{16}$$

$$\frac{d\vec{\Omega}}{dt} \; - \; \vec{\alpha}_o \; = \; 0 \tag{17}$$

The inclusion of the frame-rotation term $\vec{\Omega}\times\vec{U}$ makes $\vec{a}_o$ the absolute acceleration vector relative to the inertial XYZ earth system. The simple rate $d\vec{U}/dt$ is merely the relative acceleration, equivalent to $(\dot{V}_\infty\,,\, V_\infty\dot{\beta}\,,\, V_\infty\dot{\alpha})$ for the case $\alpha_0 = \beta_0 = 0$.

The absolute acceleration $\vec{a}_o$ and angular acceleration $\vec{\alpha}_o$ in equations (16) and (17) are additional variables included in the state vector, which therefore must have additional constraining equations. For an aircraft anchored to a mount having known motion (e.g. wind tunnel model), simple direct constraints are appropriate:

$$\vec{a}_o \; - \; \vec{a}_{\rm spec} \;\; = \;\; 0 \tag{18}$$

$$\vec{\alpha}_o \; - \; \vec{\alpha}_{\rm spec} \;\; = \;\; 0 \tag{19}$$

For an aircraft in free flight, the following indirect constraints are used, like in the quasi-steady case.

$$\vec{\mathcal{F}} \;\; = \;\; 0 \tag{20}$$

$$\vec{\mathcal{M}} \;\; = \;\; 0 \tag{21}$$

The $\vec{\mathcal{F}}$ and $\vec{\mathcal{M}}$ forces and moments include gravity and inertial reaction loads. Hence, they are equivalent to "$F - ma = 0$", and therefore are correct even for a maneuvering aircraft.

4 Inertial velocity, acceleration

The absolute velocity $\vec{v}$ and acceleration $\vec{a}$ relative to the inertial frame have the following forms at the point $\vec{r}_i$ on the beam s-axis.

$$\vec{v}(\vec{r}_i) \;\equiv\; \vec{v}_i \;\; = \;\; \vec{U} \; + \; \vec{u}_i \; + \; \vec{\Omega}\times\vec{r}_i \tag{22}$$

$$\vec{a}(\vec{r}_i) \;\equiv\; \vec{a}_i \;\; = \;\; \vec{a}_o \; + \; \dot{\vec{u}}_i \; + \; \vec{\alpha}_o\times\vec{r}_i \; + \; \vec{\Omega}\times\left(\vec{\Omega}\times\vec{r}_i\right) \; + \; 2\vec{\Omega}\times\vec{u}_i \tag{23}$$

The centrifugal and Coriolis acceleration terms appear explicitly. The usual frame-rotation acceleration term $\vec{\Omega}\times\vec{U}$ is contained in $\vec{a}_o$ by virtue of relation (16), as discussed previously.

The offset vector $\Delta\vec{r}_p$ is fixed to the local c, s, n axes which have the relative rotation rate $\vec{\omega}_i$. Hence the relative rate of change of the offset vector is due only to this rotation.

$$\frac{d}{dt}\Delta\vec{r}_p \;\; = \;\; \vec{\omega}_i\times\Delta\vec{r}_p \tag{24}$$

The absolute velocity and acceleration of $\vec{r}_p$ then have the following form.

$$\vec{r}_p \;\; = \;\; \vec{r}_i \; + \; \Delta\vec{r}_p \tag{25}$$

$$\vec{v}(\vec{r}_p) \;\equiv\; \vec{v}_p \;\; = \;\; \vec{v}_i \; + \; \left(\vec{\Omega}+\vec{\omega}_i\right)\times\Delta\vec{r}_p \tag{26}$$

$$\begin{aligned} \vec{a}(\vec{r}_p) \;\equiv\; \vec{a}_p \;\; = \;\; & \vec{a}_i \; + \; \left(\vec{\alpha}_o+\dot{\vec{\omega}}_i\right)\times\Delta\vec{r}_p \\ & + \; \vec{\Omega}\times\left(\vec{\Omega}\times\Delta\vec{r}_p\right) \; + \; \vec{\omega}_i\times(\vec{\omega}_i\times\Delta\vec{r}_p) \; + \; 2\vec{\Omega}\times(\vec{\omega}_i\times\Delta\vec{r}_p) \end{aligned} \tag{27}$$

5 Aerodynamic loads

Unsteady aerodynamic loads are represented by a bound circulation $\Gamma(s,t)$ which is now time-dependent. The usual Fourier expansion along the surface beam can still be employed, but it now has time-dependent coefficients.

$$\Gamma(\theta,t) \;=\; \sum_{k=1}^{K} A_k(t)\,\sin(k\theta) \tag{28}$$

Alternatively, a piecewise-constant spanwise distribution $A_i(t)$ can be employed, which is equivalent to the usual Vortex-Lattice treatment where the strength of each horseshoe vortex is an unknown.

The inertial-frame velocity $\vec{v}_{c/4}$ of the bound vortex at the quarter-chord location $\vec{r}_{c/4}$ is given by equation (26).

$$\Delta\vec{r}_{c/4} \;=\; (\bar{c}/4 - \bar{x}_o)\,\hat{c} \tag{29}$$

$$\vec{v}_{c/4} \;=\; \vec{v}_i \;+\; \left(\vec{\Omega} + \vec{\omega}_i\right) \times \Delta\vec{r}_{c/4} \tag{30}$$

The velocity relative to the bound vortex is then given by the usual kinematic velocity summation.

$$\vec{V}(\vec{r}_{c/4},t) \;=\; \vec{V}_{\rm ind}(\vec{r}_{c/4},t) \;+\; \vec{V}_{\rm gust}(\vec{r}_{c/4},t) \;-\; \vec{v}_{c/4} \tag{31}$$

$$\vec{V}_{\rm ind}(\vec{r}_{c/4},t) \;=\; \sum_{k=1}^{K} \vec{v}_k(\vec{r}_{c/4},t)\,A_k(t) \;+\; \vec{w}_{\rm vol}(\vec{r}_{c/4})\,V_\infty(t) \tag{32}$$

The induced velocity $\vec{V}_{\rm ind}$ is still given in terms of influence functions, which in effect neglects the shed vorticity in the wake. This shed vorticity will be approximated later when the flow-tangency constraints on the A_k coefficients are formulated. The volume influence function $\vec{w}_{\rm vol}$ is assumed to be unaffected by unsteady effects except via the time dependence of the instantaneous freestream, which is in effect a quasi-steady approximation. The gust velocity $\vec{V}_{\rm gust}$ will be treated in more detail in a later section.

The aerodynamic lift vector is taken to be the sum of steady and unsteady parts, computed using the steady and unsteady vector form of the Kutta-Joukowsky theorem, with the total relative velocity $\vec{V}$ taken at $\vec{r}_{c/4}$.

$$\vec{f}_{\rm lift} \;=\; \vec{f}_S \;+\; \vec{f}_U \tag{33}$$

$$\vec{f}_S \;=\; \rho\,\Gamma\,\vec{V} \times \hat{s} \tag{34}$$

$$\vec{f}_U \;=\; \rho\,\frac{\partial\Gamma}{\partial t}\,\frac{\bar{c}}{|\vec{V}_\perp|}\,\vec{V} \times \hat{s} \tag{35}$$

The profile moment vector $\vec{m}_{\rm lift}$ is constructed to be consistent with Theodorsen's theory [2].

$$\vec{m}_{\rm lift} \;=\; \Delta\vec{r}_{c/4} \times \vec{f}_S \;+\; \Delta\vec{r}_{c/4} \times \vec{f}_U \;+\; \frac{1}{2}\rho\,|\vec{V}_\perp|^2\,\bar{c}^2\,c_m\,\hat{s} \tag{36}$$

Note that both parts of the lift act at the quarter-chord, which gives Theodorsen's result at both the low and the high reduced-frequency limits. The profile drag loads $\vec{f}_{\rm drag}$ are assumed to be unchanged from the steady case.

6 Loads

6.1 Inertial and gravity loads

The absolute acceleration at the local mass centroid of a beam element is given by equation (27), with $\Delta\vec{r}_{cg}$ being the offset vector from the s axis.

$$\begin{aligned} \Delta\vec{r}_{cg} &= c_{cg}\,\hat{c} \;+\; n_{cg}\,\hat{n} && (37) \\ \vec{a}_{cg} &= \vec{a}_i \;+\; \left(\vec{\alpha}_o + \dot{\vec{\omega}}_i\right) \times \Delta\vec{r}_{cg} \\ &\quad +\; \vec{\Omega}\times\left(\vec{\Omega}\times\Delta\vec{r}_{cg}\right) \;+\; \vec{\omega}_i\times(\vec{\omega}_i\times\Delta\vec{r}_{cg}) \;+\; 2\vec{\Omega}\times(\vec{\omega}_i\times\Delta\vec{r}_{cg}) && (38) \end{aligned}$$

The local gravitational, inertial-reaction, and precession loads retain their quasi-steady form, except that the local beam section rotations now also include $\vec{\omega}$ and $\dot{\vec{\omega}}$.

$$\begin{aligned} \vec{f}_{\rm acc} &= \mu\,(\vec{g} - \vec{a}_{cg}) && (39) \\ \vec{m}_{\rm acc} &= \Delta\vec{r}_{cg} \times \vec{f}_{\rm acc} \;-\; \bar{\bar{T}}^T\,\bar{\bar{\imath}}\,\bar{\bar{T}}\,\left(\vec{\alpha}_o + \dot{\vec{\omega}}_i\right) \;-\; \left(\vec{\Omega} + \vec{\omega}_i\right) \times \left\{\bar{\bar{T}}^T\,\bar{\bar{\imath}}\,\bar{\bar{T}}\,\left(\vec{\Omega} + \vec{\omega}_i\right)\right\} && (40) \end{aligned}$$

6.2 Apparent-mass loads

The surface apparent-mass loads still depend only on the normal component of the local acceleration $\vec{a}_{c/2}$ evaluated at midchord, per Theodorsen's theory [2].

$$\begin{aligned} \Delta\vec{r}_{c/2} &= (\bar{c}/2 - \bar{x}_o)\,\hat{c} && (41) \\ \vec{a}_{c/2} &= \vec{a}_i \;+\; \left(\vec{\alpha}_o + \dot{\vec{\omega}}_i\right) \times \Delta\vec{r}_{c/2} \\ &\quad +\; \vec{\Omega}\times\left(\vec{\Omega}\times\Delta\vec{r}_{c/2}\right) \;+\; \vec{\omega}_i\times\left(\vec{\omega}_i\times\Delta\vec{r}_{c/2}\right) \;+\; 2\vec{\Omega}\times\left(\vec{\omega}_i\times\Delta\vec{r}_{c/2}\right) && (42) \end{aligned}$$

Like the acceleration loads, the apparent mass loads include the effects of the local beam section $\vec{\omega}$ and $\dot{\vec{\omega}}$.

$$\begin{aligned} \vec{f}_{\rm am} &= \frac{\pi}{4}\,\rho\,\bar{c}^2\,\left[\vec{V}\times\left(\vec{\Omega} + \vec{\omega}_i\right)\cdot\hat{n} \;-\; \vec{a}_{c/2}\cdot\hat{n}\right]\hat{n} && \text{(surface beam)} \quad (43) \\ \vec{m}_{\rm am} &= -\frac{\pi}{4}\,\rho\,\bar{c}^2\,\frac{\bar{c}}{4}\left[\vec{V}\times\left(\vec{\Omega} + \vec{\omega}_i\right)\cdot\hat{n} \;+\; \frac{\bar{c}}{8}\left(\vec{\alpha}_o + \dot{\vec{\omega}}_i\right)\cdot\hat{s}\right]\hat{s} \;+\; \Delta\vec{r}_{c/2}\times\vec{f}_{\rm am} && (44) \end{aligned}$$

The apparent-mass loads on a fuselage beam have the same form as the quasi-steady case.

$$\vec{f}_{\rm am} = -2\pi\,\rho\,R^2\left(\vec{a}_i - (\vec{a}_i\cdot\hat{s})\,\hat{s}\right) \qquad \text{(fuselage beam)} \qquad (45)$$

The unsteady effects are hidden in the local acceleration $\vec{a}_i$ given by (23), which now includes the local beam velocity $\vec{u}_i$ and acceleration $\dot{\vec{u}}_i$,

6.3 Point-mass loads

The local relative air velocity $\vec{V}_p$ and the absolute acceleration $\vec{a}_p$ of the point mass follow from relations (26) and (27) with the usual pylon offset vector $\Delta\vec{r}_p$. The force and moment applied to the beam is then the same as the quasi-steady case.

$$\Delta\vec{F}_{\text{pmass}} = m_p\,(\vec{g} - \vec{a}_p) \;+\; \frac{1}{2}\rho\,|\vec{V}_p|\,\vec{V}_p\,(C_D A)_p \;+\; \vec{F}_{\text{eng}} \tag{46}$$

$$\begin{aligned}\Delta\vec{M}_{\text{pmass}} &= \Delta\vec{r}_p \times \Delta\vec{F}_{\text{pmass}} \;-\; \left(\vec{\Omega} + \vec{\omega}_i\right) \times \vec{H}_p \\ &\quad - \bar{\bar{I}}_p\left(\vec{\alpha}_o + \dot{\vec{\omega}}_i\right) \;-\; \left(\vec{\Omega} + \vec{\omega}_i\right) \times \left(\bar{\bar{I}}_p\,(\vec{\Omega} + \vec{\omega}_i)\right) \;+\; \vec{M}_{\text{eng}}\end{aligned} \tag{47}$$

Again, the unsteady effects are buried in $\vec{a}_p$ and $\vec{V}_p$, which now include the effects of $\vec{u}_i$, $\vec{\omega}_i$, and their rates $\dot{\vec{u}}_i$, $\dot{\vec{\omega}}_i$.

6.4 Strut and joint loads

The strut and joint loads use a static formulation and are unaffected by dynamic effects. Hence, they retain their steady form without change.

7 Velocity Influence Coefficients

7.1 Locally-2D approximation

ASWING does not explicitly track shed vorticity. Instead, to compute the total velocity at the beam points, it uses a simplified approach where only horseshoe vortices are placed on the surface beams, and their velocity influences at all control and beam points are computed via the usual horseshoe vortex influence coefficients $\vec{v}_k$, together with the instantaneous (unsteady) vortex circulations. The source and doublet elements on the fuselage beams are similarly computed via their influences $\vec{w}_{\text{vol}}$. The influence of the shed vorticity is assumed to act only locally, which is quantified by an empirical lag term added to the following overall induced velocity summation. This added term directly modifies the unsteady flow-tangency circulation constraint.

$$\vec{V}_{\text{ind}}(\vec{r}_{c.p.}, t) = \sum_{k=1}^{K} \vec{v}_k(\vec{r}_{c.p.})A_k(t) \;+\; \vec{w}_{\text{vol}}(\vec{r}_{c.p.})V_\infty(t) \;-\; \frac{b}{V_\perp}\frac{\partial\Gamma}{\partial t}\,\hat{n}_{c.p.} \tag{48}$$

$$\left(\vec{V}_{c.p.}\cdot\hat{n}_{c.p.}\right)_{\text{unsteady}} = \left(\vec{V}_{c.p.}\cdot\hat{n}_{c.p.}\right)_{\text{steady}} \;-\; \frac{b}{V_\perp}\frac{\partial\Gamma}{\partial t} \;=\; 0 \tag{49}$$

The new lag term accounts for the downwash of the spanwise vorticity being shed at a rate of $\partial\Gamma/\partial t$, as shown in Figure 2, with $b = 2/\pi$ being a suitable lag constant. This value is obtained by comparing this approximate treatment with Theodorsen's theory.

In the small-disturbance 2D thin-airfoil limit, with the control point at the 3/4 chord location, the unsteady flow tangency constraint (49) becomes

$$V\vartheta \;-\; \dot{z} \;+\; \frac{\bar{c}}{4}\dot{\vartheta} \;-\; \frac{1}{\pi\bar{c}}\Gamma \;-\; \frac{b}{V}\frac{d\Gamma}{dt} \;=\; 0 \tag{50}$$

$$\text{or} \qquad \Gamma \;+\; \pi\bar{c}\,\frac{b}{V}\frac{d\Gamma}{dt} \;=\; \pi\,\bar{c}\left[V\vartheta \;-\; \dot{z} \;+\; \frac{\bar{c}}{4}\dot{\vartheta}\right] \tag{51}$$

where $\vartheta(t)$ and $z(t)$ are the pitch and plunge motions. With the sinusoidal motion

$$\vartheta(t) = \tilde{\vartheta}e^{i\omega t} \qquad\qquad z(t) = \tilde{z}e^{i\omega t} \qquad\qquad \Gamma(t) = \tilde{\Gamma}e^{i\omega t}$$

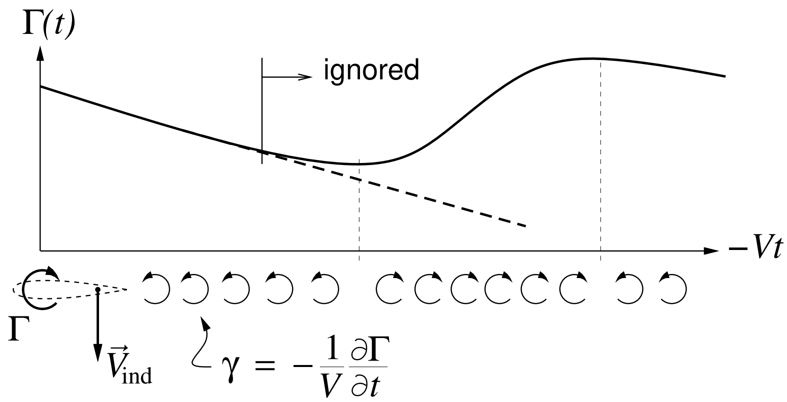

Figure 2: Unsteady downwash due to recently-shed vortex sheet strength γ related to current circulation rate of change.

the flow tangency equation reduces to

$$\tilde{\Gamma}\,(1 \,+\, 2\pi\, ik\, b) \;=\; \pi\,\bar{c}\,V\left[\left(1+\frac{1}{2}ik\right)\tilde{\vartheta} \,-\, ik\,\frac{\tilde{z}}{\bar{c}/2}\right] \tag{52}$$

$$k \;=\; \frac{\omega\bar{c}}{2V} \tag{53}$$

where k is the standard reduced frequency based on the half-chord. The resulting circulation-related unsteady complex lift computed via equations (34,35) is

$$f_{\rm lift} \;=\; \rho V\Gamma \,+\, \rho\bar{c}\,\frac{\partial\Gamma}{\partial t} \tag{54}$$

$$\begin{aligned}\tilde{f}_{\rm lift} &= \rho V\tilde{\Gamma}\,(1 \,+\, 2ik)\\ &= \frac{1}{2}\rho V^2\,\bar{c}\,2\pi\left[\left(1+\frac{1}{2}ik\right)\tilde{\vartheta} \,-\, ik\,\frac{\tilde{z}}{\bar{c}/2}\right]\frac{1+2ik}{1+2\pi ik\,b}\end{aligned} \tag{55}$$

For a perfect match with Theodorsen's result, the complex fraction on the righthand side *should* be equal to the lag function $C(k) = F(k) + iG(k)$, which has the asymptotic behavior $C(k) \rightarrow 1/2$ as $k \rightarrow \infty$. This indicates that $b = 2/\pi$ is the appropriate choice for this empirical lag constant. This defines an implied effective lag function.

$$C_{\rm implied}(k) \;=\; F_{\rm implied}(k) + iG_{\rm implied}(k) \;=\; \frac{1+2ik}{1+4ik} \tag{56}$$

In summary, when the empirically-lagged flow-tangency circulation constraint (49) is used for a nearly-2D oscillatory case, the oscillatory lift and moment computed using expressions (34,35) will be the same as if a standard Theodorsen analysis were performed, but with the approximate $C_{\rm implied}(k)$ function (56) being used in lieu of the actual exact $C(k)$. As shown in Figure 3, the two compare quite reasonably. In fact, it is doubtful whether a better match could be obtained with a "correct" treatment of the unsteady wake, given the lifting-line representation of the surface.

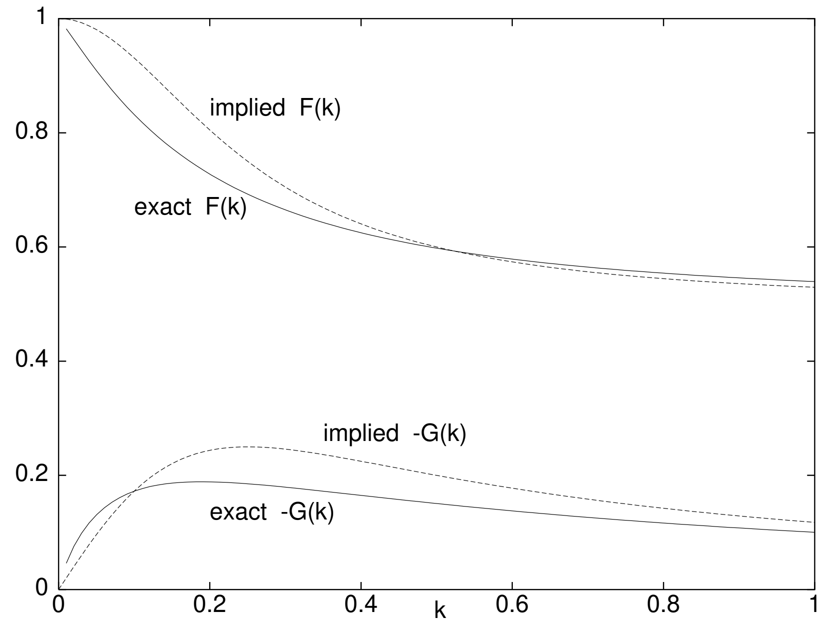

Figure 3: Comparison of approximate lag function components implied by empirical lagged flow tangency treatment, compared with exact Theodorsen lag function components.

8 Structural Damping

8.1 Formulation

Structural damping is modeled by assuming that material stresses and loads are proportional not only to the strain, but also to the strain rate. For example, a simple uniaxial stress/strain relation for the structural material would have the form

$$\sigma_{ss} \;=\; E\,(\epsilon_{ss} \,+\, t_d \dot{\epsilon}_{ss})$$

where t_d is a specified characteristic damping time. Any simple harmonic oscillator which uses this material as the restoring spring will have a damping ratio, frequency, and damping time constant given by

$$\zeta \;=\; \frac{1}{2}\,\omega_0 t_d \qquad\qquad \omega \;=\; \omega_0\sqrt{1-\zeta^2} \qquad\qquad \sigma \;=\; -\omega_0\,\zeta$$

where ω_0 is the oscillator's undamped natural frequency. Figure 4 shows the effect of the added damping on the characteristic roots. For $\omega_0 t_d \ll 1$, there is little effect. Critical damping ($\zeta = 1$) occurs when $\omega_0 t_d = 2$.

The damping is incorporated into the beam strain/load and curvature/moment relations.

$$\left\{ \begin{array}{c} \gamma_c \,+\, t_{\gamma_c}\dot{\gamma}_c \\ \epsilon_s \,+\, t_{\epsilon_s}\dot{\epsilon}_s \\ \gamma_n \,+\, t_{\gamma_n}\dot{\gamma}_n \end{array} \right\} = \left\{ \begin{array}{c} F_c/GK_c \\ F_s/EA \\ F_n/GK_n \end{array} \right\} + \left[\begin{array}{ccc} 0 & -n_{ea} & 0 \\ n_{ta} & 0 & -c_{ta} \\ 0 & c_{ea} & 0 \end{array} \right] \left[\begin{array}{c} \\ \bar{\bar{E}}^{-1} \\ \\ \end{array} \right] \left\{ \begin{array}{c} M'_c \\ M'_s \\ M'_n \end{array} \right\} \tag{57}$$

$$\left\{ \begin{array}{c} \kappa_c - \kappa_{c0} \,+\, t_{\kappa_c}\dot{\kappa}_c \\ \kappa_s - \kappa_{s0} \,+\, t_{\kappa_s}\dot{\kappa}_s \\ \kappa_n - \kappa_{n0} \,+\, t_{\kappa_n}\dot{\kappa}_n \end{array} \right\} = \left[\begin{array}{c} \\ \bar{\bar{E}}^{-1} \\ \\ \end{array} \right] \left\{ \begin{array}{c} M'_c \\ M'_s \\ M'_n \end{array} \right\} \tag{58}$$

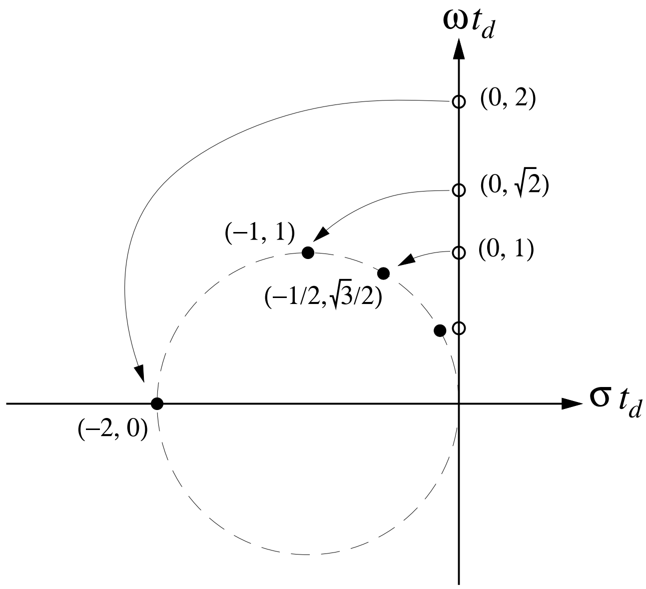

Figure 4: Modification of undamped roots by the addition of damping time t_d.

A different damping time has been defined for each component, although this level of detail may not be necessary in practice. It is reasonable to assume that the surface or beam has one damping time t_ϵ characterizing the beam material which resists spanwise strains, such as the sparcaps. Another damping time t_γ is assumed to characterize the material resisting shearing, such as the torsion skin. We can then define all six damping times by specifying only t_ϵ and t_γ.

$$\left\{ \begin{array}{c} t_{\gamma_c} \\ t_{\epsilon_s} \\ t_{\gamma_n} \end{array} \right\} = \left\{ \begin{array}{c} t_\gamma \\ t_\epsilon \\ t_\gamma \end{array} \right\} \qquad\qquad \left\{ \begin{array}{c} t_{\kappa_c} \\ t_{\kappa_s} \\ t_{\kappa_n} \end{array} \right\} = \left\{ \begin{array}{c} t_\epsilon \\ t_\gamma \\ t_\epsilon \end{array} \right\} \tag{59}$$

8.2 Implementation

8.2.1 Shear and extension strain rates

To implement equations (57) and (58), it is necessary to express the time rates of the strains and curvatures, $\dot{\gamma}$, $\dot{\epsilon}$, $\dot{\kappa}$ in terms of the state variables. First we rearrange the three previously derived deflection/strain relations, and take the time derivative.

$$\left\{ \begin{array}{c} \gamma_c \\ 1+\epsilon_s \\ \gamma_n \end{array} \right\}_a \Delta s_0 \;=\; \bar{\bar{T}}_a \, \Delta\vec{r} \qquad \rightarrow \qquad \left\{ \begin{array}{c} \dot{\gamma}_c \\ \dot{\epsilon}_s \\ \dot{\gamma}_n \end{array} \right\}_a \Delta s_0 \;=\; \bar{\bar{T}}_a \, \Delta\dot{\vec{r}} \;+\; \dot{\bar{\bar{T}}}_a \, \Delta\vec{r}$$

Using $\dot{\vec{r}} = \vec{u}$ and $\dot{\bar{\bar{T}}} = -\bar{\bar{T}}\bar{\bar{\omega}}$ then gives a suitable expression for the strain rates in terms of the beam element positions, velocities, and rotation rates.

$$\left\{ \begin{array}{c} \dot{\gamma}_c \\ \dot{\epsilon}_s \\ \dot{\gamma}_n \end{array} \right\}_a \Delta s_0 \;=\; \bar{\bar{T}}_a \left\{ \Delta\vec{u} \;-\; \vec{\omega}_a \times \Delta\vec{r} \right\} \tag{60}$$

Starting again with the previously derived deflection/strain compatibility relations

$$\begin{Bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{Bmatrix} = \begin{bmatrix} & \bar{\bar{T}}_a^T & \end{bmatrix} \begin{Bmatrix} \gamma_c \\ 1+\epsilon_s \\ \gamma_n \end{Bmatrix}_a \Delta s_0$$

we use (57) and (60) to eliminate the strains and strain rates in terms of the beam state variables. This gives the following final discrete deflection/strain compatibility equations.

$$\begin{Bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{Bmatrix} = \begin{bmatrix} & \bar{\bar{T}}_a^T & \end{bmatrix} \left\{ \begin{Bmatrix} F_c/GK_c \\ 1+F_s/EA \\ F_n/GK_n \end{Bmatrix} + \begin{bmatrix} 0 & -n_{ea} & 0 \\ n_{ta} & 0 & -c_{ta} \\ 0 & c_{ea} & 0 \end{bmatrix} \begin{bmatrix} & \bar{\bar{E}}^{-1} & \end{bmatrix} \begin{Bmatrix} M_c' \\ M_s' \\ M_n' \end{Bmatrix} \right\} \Delta s_0$$
$$- \begin{bmatrix} t_{\gamma_c} & & \\ & t_{\epsilon_s} & \\ & & t_{\gamma_n} \end{bmatrix} \{\Delta\vec{u} \, - \, \vec{\omega}_a \times \Delta\vec{r}\} \tag{61}$$

8.2.2 Curvature strain rates

Taking the time derivative of the rearranged angle/curvature relation gives

$$\begin{Bmatrix} \kappa_c \\ \kappa_s \\ \kappa_n \end{Bmatrix} \Delta s \, = \, \bar{\bar{K}}_a \, \Delta\vec{\theta} \qquad \rightarrow \qquad \begin{Bmatrix} \dot{\kappa}_c \\ \dot{\kappa}_s \\ \dot{\kappa}_n \end{Bmatrix} \Delta s \, = \, \bar{\bar{K}}_a \, \Delta\dot{\vec{\theta}} \, + \, \dot{\bar{\bar{K}}}_a \, \Delta\vec{\theta} \tag{62}$$

where the extension-rate contribution from $(\dot{\Delta s})$ has been neglected since it's expected to be very small. The rate of the curvature-definition matrix $\bar{\bar{K}}(\varphi, \vartheta, \psi)$ is expanded using its partial derivatives,

$$\dot{\bar{\bar{K}}} \, = \, \bar{\bar{K}}_\varphi \, \dot{\varphi} \, + \, \bar{\bar{K}}_\vartheta \, \dot{\vartheta} \, + \, \bar{\bar{K}}_\psi \, \dot{\psi}$$

so that

$$\dot{\bar{\bar{K}}} \, \Delta\vec{\theta} \, = \, \left[\bar{\bar{K}}_\varphi \, \dot{\varphi} + \bar{\bar{K}}_\vartheta \, \dot{\vartheta} + \bar{\bar{K}}_\psi \, \dot{\psi} \right] \Delta\vec{\theta} \, = \, \left\{ \bar{\bar{K}}_\varphi \Delta\vec{\theta} \right\} \dot{\varphi} \, + \, \left\{ \bar{\bar{K}}_\vartheta \Delta\vec{\theta} \right\} \dot{\vartheta} \, + \, \left\{ \bar{\bar{K}}_\psi \Delta\vec{\theta} \right\} \dot{\psi} \, \equiv \, \Delta\bar{\bar{K}} \, \dot{\vec{\theta}}_a$$

The $\Delta\bar{\bar{K}}$ matrix has the following forms for surface and fuselage beams.

$$\Delta\bar{\bar{K}} \, = \begin{bmatrix} 0 & -\cos\psi \, \sin\vartheta \, \Delta\varphi - \cos\vartheta \, \Delta\psi & -\sin\psi \, \cos\vartheta \, \Delta\varphi \\ 0 & 0 & -\cos\psi \, \Delta\varphi \\ 0 & \cos\psi \, \cos\vartheta \, \Delta\varphi - \sin\vartheta \, \Delta\psi & -\sin\psi \, \sin\vartheta \, \Delta\varphi \end{bmatrix} \qquad \text{(surface beam)}$$

$$\Delta\bar{\bar{K}} \, = \begin{bmatrix} \sin\varphi \, \sin\vartheta \, \Delta\psi & -\sin\vartheta \, \Delta\varphi - \cos\varphi \, \cos\vartheta \, \Delta\psi & 0 \\ -\cos\varphi \, \Delta\psi & 0 & 0 \\ -\sin\varphi \, \cos\vartheta \, \Delta\psi & \cos\vartheta \, \Delta\varphi - \cos\varphi \, \sin\theta \, \Delta\psi & 0 \end{bmatrix} \qquad \text{(fuselage beam)}$$

Equation (62) can now be written as

$$\begin{Bmatrix} \dot{\kappa}_c \\ \dot{\kappa}_s \\ \dot{\kappa}_n \end{Bmatrix} \Delta s \, = \, \bar{\bar{K}}_a \, \Delta\dot{\vec{\theta}} \, + \, \Delta\bar{\bar{K}} \, \dot{\vec{\theta}}_a \tag{63}$$

and substituting $\dot{\vec{\theta}} = \bar{\bar{K}}^{-1}\bar{\bar{T}}\vec{\omega}$ then gives the curvature rates in terms of the state variables.

$$\left\{ \begin{array}{c} \dot{\kappa}_c \\ \dot{\kappa}_s \\ \dot{\kappa}_n \end{array} \right\} \Delta s \;=\; \bar{\bar{K}}_a \, \Delta\!\left(\bar{\bar{K}}^{-1}\bar{\bar{T}}\vec{\omega}\right) \;+\; \Delta\bar{\bar{K}} \left(\bar{\bar{K}}^{-1}\bar{\bar{T}}\vec{\omega}\right)_a \tag{64}$$

Starting again from the angle/curvature compatibility relations

$$\bar{\bar{K}}_a \, \Delta\vec{\theta} \;=\; \left\{ \begin{array}{c} \kappa_c \\ \kappa_s \\ \kappa_n \end{array} \right\} \Delta s$$

we use (58) and (63) to eliminate the curvatures and curvature rates in terms of the beam state variables. This gives the final discrete angle/curvature compatibility equations.

$$\begin{aligned} \left[\;\; \bar{\bar{K}}_a \;\; \right] \left\{ \begin{array}{c} \Delta\varphi \\ \Delta\vartheta \\ \Delta\psi \end{array} \right\} - \left[\;\; \bar{\bar{K}}_{a0} \;\; \right] \left\{ \begin{array}{c} \Delta\varphi_0 \\ \Delta\vartheta_0 \\ \Delta\psi_0 \end{array} \right\} &= \left[\;\; \bar{\bar{E}}^{-1} \;\; \right] \left\{ \begin{array}{c} M'_c \\ M'_s \\ M'_n \end{array} \right\} \Delta s \\ &- \left[\begin{array}{ccc} t_{\kappa_c} & & \\ & t_{\kappa_s} & \\ & & t_{\kappa_n} \end{array} \right] \left\{ \bar{\bar{K}}_a \, \Delta\!\left(\bar{\bar{K}}^{-1}\bar{\bar{T}}\vec{\omega}\right) \;+\; \Delta\bar{\bar{K}} \left(\bar{\bar{K}}^{-1}\bar{\bar{T}}\vec{\omega}\right)_a \right\} \end{aligned} \tag{65}$$

In practice, the second damping term involving $\Delta\bar{\bar{K}}$ is negligible compared to the first, and can be omitted in this level of modeling.

9 Gust Field

9.1 Formulation

The atmospheric gust field is specified in Earth coordinates and time: $\vec{V}_{E_{\rm gust}}(\vec{r}_E, t)$, which for time-domain calculations is arbitrary and is provided via a subroutine. The resulting velocity change $\Delta\vec{V}$ seen by a body-axes location $\vec{r}$ on the aircraft is evaluated by

$$\Delta\vec{V}(\vec{r}, t) \;=\; \bar{\bar{T}}_E^T \, \vec{V}_{E_{\rm gust}}(\vec{r}_E, t) \tag{66}$$

$$\text{where} \qquad \vec{r}_E(\vec{r}, t) \;=\; \vec{R}_E \;+\; \bar{\bar{T}}_E \vec{r} \tag{67}$$

For the Bode analysis and mode-excitation analysis considered henceforth, the $\vec{V}_{E_{\rm gust}}(\vec{r}_E, t)$ field is treated via its spatial Fourier coefficients. It is also assumed to be sufficiently weak to be treatable via linearization. Any z_E variation in the gust field is also neglected for simplicity, although this is not a hard requirement. We first define conveniently normalized forms of the streamwise and spanwise coordinates x_E, y_E,

$$\xi \;=\; 2\, x_E / c'_{\rm ref} \tag{68}$$

$$\eta \;=\; 2\, y_E / b'_{\rm ref} \tag{69}$$

where the streamwise and spanwise reference lengths $c'_{\rm ref}$,$b'_{\rm ref}$ are arbitrary. The streamwise and spanwise Fourier series expansion of the gust field is then defined as follows.

$$\begin{aligned} \vec{V}_{E_{\rm gust}}(x_E, y_E, t) \;=\; \sum_{n=0}^{N} \sum_{m=0}^{M} \Big[& \vec{A}^c_{mn} \cos(m\pi\xi) \cos(n\pi\eta) \;+\; \vec{A}^s_{mn} \sin(m\pi\xi) \cos(n\pi\eta) \\ & + \vec{B}^c_{mn} \cos(m\pi\xi) \sin(n\pi\eta) \;+\; \vec{B}^s_{mn} \sin(m\pi\xi) \sin(n\pi\eta) \Big] \end{aligned} \tag{70}$$

It should be noted that $\vec{A}_{mn}$ represent the spanwise-symmetric part of the gust field, and $\vec{B}_{mn}$ represent the spanwise-antisymmetric part. These series coefficients are in general functions of t, and are defined by a standard Fourier analysis.

$$\vec{A}^c_{mn}(t) = \frac{1}{c_m c_n} \int_{-1}^{1} \int_{-1}^{1} \vec{V}_{E_{\text{gust}}} \; \cos(m\pi\xi) \; \cos(n\pi\eta) \; d\xi \; d\eta \tag{71}$$

$$\vec{A}^s_{mn}(t) = \frac{1}{c_m c_n} \int_{-1}^{1} \int_{-1}^{1} \vec{V}_{E_{\text{gust}}} \; \sin(m\pi\xi) \; \cos(n\pi\eta) \; d\xi \; d\eta \tag{72}$$

$$\vec{B}^c_{mn}(t) = \frac{1}{c_m c_n} \int_{-1}^{1} \int_{-1}^{1} \vec{V}_{E_{\text{gust}}} \; \cos(m\pi\xi) \; \sin(n\pi\eta) \; d\xi \; d\eta \tag{73}$$

$$\vec{B}^s_{mn}(t) = \frac{1}{c_m c_n} \int_{-1}^{1} \int_{-1}^{1} \vec{V}_{E_{\text{gust}}} \; \sin(m\pi\xi) \; \sin(n\pi\eta) \; d\xi \; d\eta \tag{74}$$

$$c_k = \begin{cases} 2 & \text{if} \quad k = 0 \\ 1 & \text{if} \quad k > 0 \end{cases}$$

Some of the coefficients in (70), specifically $\vec{A}^s_{0n}$, $\vec{B}^c_{m0}$, $\vec{B}^s_{00}$, have zero mode shapes and hence have no effect. They are simply omitted from the expansion. The four basic modes in (70) are shown in Figure 5 for $m = 1$ and $n = 1$.

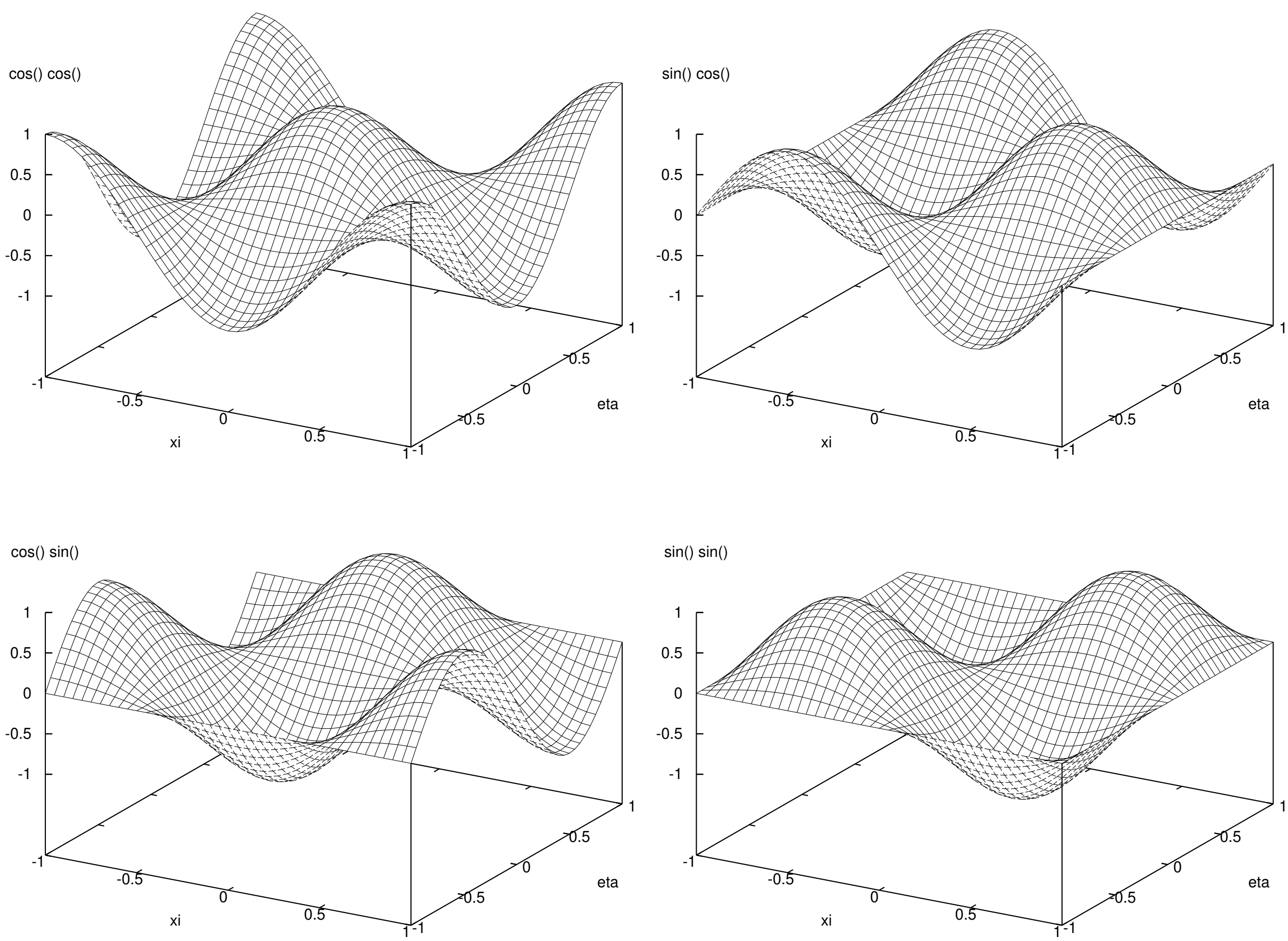

Figure 5: Four basic gust modes for $m = 1$ and $n = 1$.

We define streamwise and spanwise wavenumbers by

$$k_E = 2\pi\, m/c'_{\rm ref} \tag{75}$$
$$\ell_E = 2\pi\, n/b'_{\rm ref} \tag{76}$$

so that the sinusoidal function arguments in (70) can be alternatively written as follows.

$$m\pi\xi = k_E x_E \tag{77}$$
$$n\pi\eta = \ell_E y_E \tag{78}$$

This allows specifying the mode shapes either by the mode indices m, n and corresponding reference lengths $c'_{\rm ref}, b'_{\rm ref}$, or simply by k_E, ℓ_E alone.

9.2 Translation to aircraft frame

For algebraic simplicity, it is assumed that the inertial coordinates $\vec{r}_E = \{x_E, y_E, z_E\}^T$ are oriented so that x_E is along the flight path, and hence $\vec{U}_E = \{U_E, 0, 0\}^T$. Assuming the aircraft flies through the field at a nearly uniform velocity, we have

$$\vec{R}_E = \vec{U}_E\, t = \{U_E t\, ,\, 0\, ,\, 0\}^T \tag{79}$$
$$\vec{r}_E = \vec{U}_E\, t + \bar{\bar{T}}_E \vec{r} = \{U_E t\, ,\, 0\, ,\, 0\}^T + \bar{\bar{T}}_E \vec{r} \tag{80}$$
$$m\pi\xi = k_E x_E = \vec{k}\cdot\vec{r} + \omega t \tag{81}$$
$$n\pi\eta = \ell_E y_E = \vec{\ell}\cdot\vec{r} \tag{82}$$

where

$$\vec{k} = \bar{\bar{T}}_E^{\,T} \{k_E, 0, 0\}^T \tag{83}$$
$$\vec{\ell} = \bar{\bar{T}}_E^{\,T} \{0, \ell_E, 0\}^T \tag{84}$$
$$\omega = k_E U_E = \vec{k}\cdot\vec{U} \tag{85}$$

9.3 Frozen gust assumption

A real gust field is typically unsteady, as expressed by the time dependence of the $\vec{A}_{mn}$ and $\vec{B}_{mn}$ coefficients. For the purpose of computing an aircraft's linearized dynamic response, the gust will be assumed to be "frozen" in time. This is justified if the time scales of the dynamic interaction are much smaller than the time scales of the gust field itself, which is usually the case. Hence, the $\vec{A}_{mn}$ and $\vec{B}_{mn}$ mode coefficients will be assumed constant. Together with relations (81) and (82), the Fourier expansion (70) then becomes

$$\begin{aligned}\vec{V}_{E_{\rm gust}}(x_E, y_E) \;=\; \sum_{n=0}^{N}\sum_{m=0}^{M} \Big[\, & \vec{A}^c_{mn} \cos(\omega t + \vec{k}\cdot\vec{r}) \cos(\vec{\ell}\cdot\vec{r}) \;+\; \vec{A}^s_{mn} \sin(\omega t + \vec{k}\cdot\vec{r}) \cos(\vec{\ell}\cdot\vec{r}) \\ & + \vec{B}^c_{mn} \cos(\omega t + \vec{k}\cdot\vec{r}) \sin(\vec{\ell}\cdot\vec{r}) \;+\; \vec{B}^s_{mn} \sin(\omega t + \vec{k}\cdot\vec{r}) \sin(\vec{\ell}\cdot\vec{r}) \,\Big]\end{aligned} \tag{86}$$

The unsteadiness in the aircraft frame is then entirely due to its motion through the frozen gust, as shown in Figure 6. Equation (85) relates the perceived frequency ω to the streamwise wavenumber k_E and flight speed U_E.

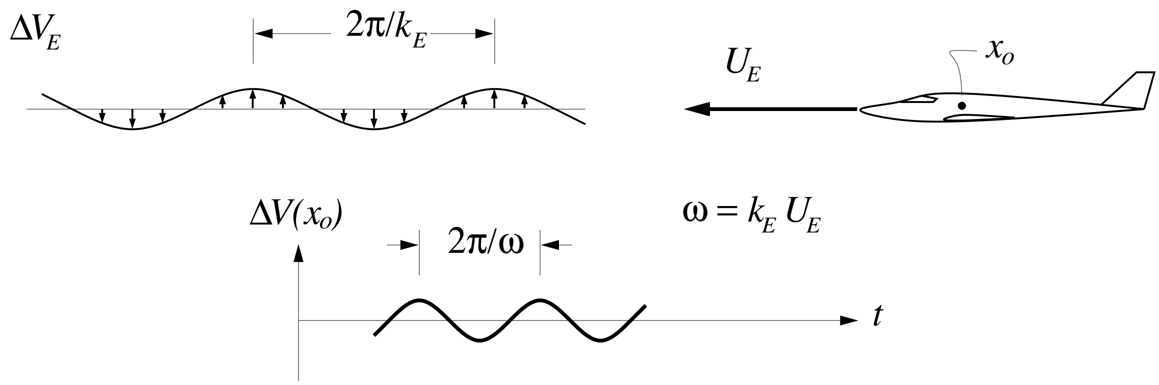

Figure 6: Frozen sinusoidal gust, as seen at some observation point x_o on aircraft flying through the gust.

9.4 Bode Response Implementation

To compute the Bode response of the aircraft to atmospheric turbulence, each cartesian component and each mode of the gust velocity in (86) is separately treated as the linearized complex forcing quantity. For example, for a vertical gust we assume a small gust field of the form

$$\vec{V}_{E_{\rm gust}} = \{0, 0, \delta W_E\}^T \tag{87}$$

$$\delta W_E = \hat{W} \exp(i\omega t) \tag{88}$$

$$\hat{W} = \mathcal{A}_{Z_{mn}} \exp(i\vec{k}\cdot\vec{r}) \cos(\vec{\ell}\cdot\vec{r}) + \mathcal{B}_{Z_{mn}} \exp(i\vec{k}\cdot\vec{r}) \sin(\vec{\ell}\cdot\vec{r}) \tag{89}$$

with $\mathcal{A}_{Z_{mn}}$ and $\mathcal{B}_{Z_{mn}}$ being arbitrary amplitudes of the symmetric and antisymmetric modes. A practical complication here is that this imposed $\hat{W}$ is a complex quantity, and hence it is not immediately suitable for implementation in real arithmetic. Instead, we impose the real velocity field

$$\begin{aligned} \bar{W}_E = & \ \Delta_{g_1} \cos(\vec{k}\cdot\vec{r}) \cos(\vec{\ell}\cdot\vec{r}) + \Delta_{g_2} \sin(\vec{k}\cdot\vec{r}) \cos(\vec{\ell}\cdot\vec{r}) \\ & + \Delta_{g_3} \cos(\vec{k}\cdot\vec{r}) \sin(\vec{\ell}\cdot\vec{r}) + \Delta_{g_4} \sin(\vec{k}\cdot\vec{r}) \sin(\vec{\ell}\cdot\vec{r}) \end{aligned} \tag{90}$$

where $\Delta_{g_{1\ldots4}}$ are "gust control variables". These are set to zero for evaluation of the residuals, but the (real!) Jacobians to their perturbations are computed as usual. To correctly compute the linearized response to the gust mode amplitude δW_E, it will then be necessary to impose

$$\Delta_{g_1} = \mathcal{A}_{Z_{mn}} \tag{91}$$

$$\Delta_{g_2} = i\,\mathcal{A}_{Z_{mn}} \tag{92}$$

$$\Delta_{g_3} = \mathcal{B}_{Z_{mn}} \tag{93}$$

$$\Delta_{g_4} = i\,\mathcal{B}_{Z_{mn}} \tag{94}$$

so that the imposed $\bar{W}_E$ given by (90) gives the physically correct δW_E via (88).

9.5 Gust specification

For given gust wavenumbers ℓ_E,k_E, or equivalently m,n, the gust field amplitude is specified by the vectors $\vec{\mathcal{A}}_{mn}$, $\vec{\mathcal{B}}_{mn}$ (only their Z-components are shown in (89)). In the actual implementation, only one specified gust direction and one symmetry/antisymmetry pair is considered at one time, e.g.

$$\vec{\mathcal{A}}_{mn} = \{\,0\,,\,0\,,\,\mathcal{A}_{Z_{mn}}\}^T \qquad\qquad \vec{\mathcal{B}}_{mn} = \{\,0\,,\,0\,,\,\mathcal{B}_{Z_{mn}}\}^T$$

Some number of indices $n = 1 \ldots n_{\max}$ are sampled, which correspond to the spanwise wavenumbers $\ell = 2\pi/b'_{\rm ref} \ldots 2\pi n_{\max}/b'_{\rm ref}$. For each of these spanwise wavenumbers, some range of frequencies $\omega = \omega_{\min} \ldots \omega_{\max}$ is then sampled, which imply the streamwise wavenumbers $k = \omega_{\min}/U_E \ldots \omega_{\max}/U_E$. This in effect gives a 2-dimensional Bode gain and phase versus the two parameters n and ω. The process can be repeated for streamwise and sideways gusts, if deemed appropriate.

10 Sensors

10.1 Sensor description

A sensor is an ASWING program feature which allows interrogation of the current solution state or velocity field at a chosen point on a beam or in space. As shown in Figure 7, the sensor consists of a rigid pylon, mounted on a beam. The intent is to mimic an actual sensor implementation, where

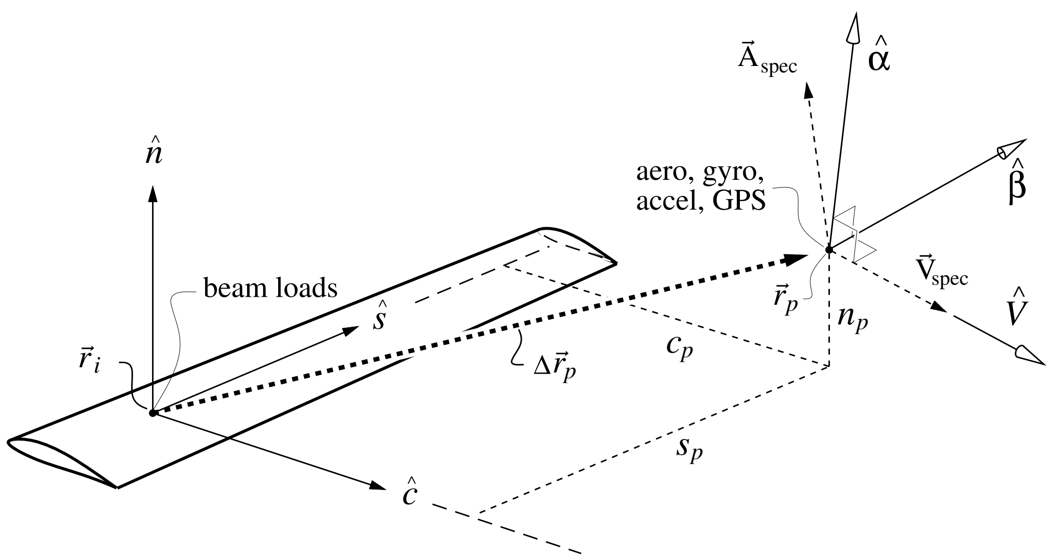

Figure 7: Sensor on pylon cantilevered from beam. $\hat{V}$, $\hat{\beta}$, $\hat{\alpha}$ are sensor axes.

the sensor's output is affected by the movement of its location due to structural deformations, and by the aerodynamic influence of nearby surfaces.

10.2 Sensor outputs

The pylon attachment point at discrete node i is where the beam's relative location and velocity, and structural loads are sampled:

$\vec{r}_i$	relative position
$\vec{u}_i$	relative velocity
$\vec{F}_i$	beam forces
$\vec{M}'_i$	beam moments

The tip of the pylon is imagined to contain an anemometer, alpha and beta vanes, position gyros, rate gyros, accelerometers, and GPS sensors. It delivers the data listed below. The $()_S$ subscript denotes that the quantity is in *sensor axes*, described later.

V_S local air velocity
α_S local alpha
β_S local beta

$\vec{\Omega}_S$ absolute rotation rate
$\vec{a}_S$ absolute linear acceleration
$\vec{\alpha}_S$ absolute angular acceleration

Φ_S roll angle
Θ_S elevation angle
Ψ_S heading angle

$\vec{r}_E$ position in earth axes (GPS)
$\vec{v}_E$ velocity in earth axes (GPS)

The sensor also delivers the following error-integral signals.

$\int (V_S - V_c) dt$
$\int (\alpha_S - \alpha_c) dt$
$\int (\beta_S - \beta_c) dt$

$\int (\Phi_S - \Phi_c) dt$
$\int (\Theta_S - \Theta_c) dt$
$\int (\Psi_S - \Psi_c) dt$

$\int (\vec{\Omega}_S - \vec{\Omega}_c) dt$
$\int (\vec{a}_S - \vec{a}_c) dt$

10.3 Sensor location and sensor axes

The sensor position is specified in the same way as a point mass, with the anchor point location $\vec{r}_{i_0}$ and sensor location $\vec{r}_{p_0}$ are both specified for the undeformed geometry. This defines the *csn* components of the connecting pylon $\Delta\vec{r}_{p_0}$ which are subsequently held fixed as the geometry deforms.

$$\Delta\vec{r}_{p_0} \equiv \vec{r}_{p_0} - \vec{r}_{i_0} \tag{95}$$

$$\{ c_p \; s_p \; n_p \}^T = \bar{\bar{T}}_0 \, \Delta\vec{r}_{p_0} \tag{96}$$

As the beam deforms, the pylon will rotate along with the beam and become the new pylon vector

$$\Delta\vec{r}_p = \bar{\bar{T}}^T \{ c_p \; s_p \; n_p \}^T = \bar{\bar{T}}^T \bar{\bar{T}}_0 \, \Delta\vec{r}_{p_0} \tag{97}$$

with the original pylon attachment point $\vec{r}_{p_0}$ also moving to a new location $\vec{r}_i$. The sensor will therefore move to a new location

$$\vec{r}_p = \vec{r}_i + \Delta\vec{r}_p$$

which is the same as equation (25) for the point mass. The sensor will likewise have a corresponding absolute velocity $\vec{v}_p$ given by equation (26), and absolute acceleration $\vec{a}_p$ given by equation (27), as derived for the point mass case.

The initial flow sensor axes are conveniently defined by the two vectors $\vec{\mathrm{V}}_{\mathrm{spec}}$ and $\vec{\mathrm{A}}_{\mathrm{spec}}$, shown in Figure 7.

$$\begin{aligned}
\hat{V}_0 &= \vec{\mathrm{V}}_{\mathrm{spec}} / \left|\vec{\mathrm{V}}_{\mathrm{spec}}\right| \\
\hat{\beta}_0 &= \vec{\mathrm{A}}_{\mathrm{spec}} \times \vec{\mathrm{V}}_{\mathrm{spec}} / \left|\vec{\mathrm{A}}_{\mathrm{spec}} \times \vec{\mathrm{V}}_{\mathrm{spec}}\right| \\
\hat{\alpha}_0 &= \hat{V}_0 \times \hat{\beta}_0
\end{aligned}$$

Note that $\hat{\alpha}_0$ is not necessarily parallel to $\vec{\mathrm{A}}_{\mathrm{spec}}$, but instead is forced to lie in the plane defined by $\vec{\mathrm{V}}_{\mathrm{spec}}$ and $\vec{\mathrm{A}}_{\mathrm{spec}}$. The sensor axes are analogous to the engine axes, except that here the orientation of the off-axial vectors $\hat{\alpha}$ and $\hat{\beta}$ is not arbitrary.

The rotation of the attachment point and pylon will also give the sensor axes a new orientation.

$$\begin{aligned}
\hat{V} &= \bar{\bar{T}}^T \bar{\bar{T}}_0 \, \hat{V}_0 \\
\hat{\beta} &= \bar{\bar{T}}^T \bar{\bar{T}}_0 \, \hat{\beta}_0 \\
\hat{\alpha} &= \bar{\bar{T}}^T \bar{\bar{T}}_0 \, \hat{\alpha}_0
\end{aligned}$$

These three unit vectors define a convenient sensor-axis transformation tensor,

$$\bar{\bar{T}}_S \equiv \begin{bmatrix} - & \hat{V} & - \\ - & \hat{\beta} & - \\ - & \hat{\alpha} & - \end{bmatrix} = \begin{bmatrix} - & \hat{V}_0 & - \\ - & \hat{\beta}_0 & - \\ - & \hat{\alpha}_0 & - \end{bmatrix} \begin{bmatrix} & & \\ & \bar{\bar{T}}^T \bar{\bar{T}}_0 & \\ & & \end{bmatrix}^T \tag{98}$$

which depends on the specified $\vec{\mathrm{V}}_{\mathrm{spec}}$ and $\vec{\mathrm{A}}_{\mathrm{spec}}$, and also on the local beam Euler angles φ_i, ϑ_i, and ψ_i for the deformed and undeformed geometry.

An alternative "fixed" sensor definition is simply

$$\vec{r}_p = \vec{r}_{p_0} \tag{99}$$

$$\hat{V} = \hat{V}_0 \tag{100}$$

$$\hat{\beta} = \hat{\beta}_0 \tag{101}$$

$$\hat{\alpha} = \hat{\alpha}_0 \tag{102}$$

so that the sensor stays at its original specified location and orientation. This may be more suitable for some applications, although such a sensor might be difficult to physically implement on a real flexible aircraft. In the program, a fixed sensor is specified by simply attaching its pylon at the ground point. At the ground point we have $\bar{\bar{T}}^T \bar{\bar{T}}_0 = \bar{\bar{\mathrm{I}}}$, which simplifies all the previous sensor quantities to the fixed-sensor case.

10.4 Sensor output definitions

At the sensor location, the relative air velocity seen by the sensor is

$$\vec{V} = \vec{V}_{\mathrm{ind}} + \vec{V}_{\mathrm{gust}} - \vec{v}_p \tag{103}$$

and the sensed velocity and flow angles are determined as follows.

$$\{u_S \;\; v_S \;\; w_S\}^T = \bar{\bar{T}}_S \, \vec{V} \tag{104}$$

$$V_S = \sqrt{u_S^2 + v_S^2 + w_S^2} = \left|\vec{V}\right| \tag{105}$$

$$\alpha_S = \arctan\left(w_S / u_S\right) \tag{106}$$

$$\beta_S = \arctan\left(-v_S / \sqrt{u_S^2 + w_S^2}\right) \tag{107}$$

The absolute rotation and accelerations of the sensor, reported as components along the sensor axes, are

$$\vec{\Omega}_S = \bar{\bar{T}}_S \left(\vec{\Omega} + \vec{\omega}_i\right) \tag{108}$$

$$\vec{a}_S = \bar{\bar{T}}_S \,(\vec{a}_p - \vec{g}) \quad , \qquad \vec{g} = \bar{\bar{T}}_E^T \{0 \;\; 0 \; -g\}^T \tag{109}$$

$$\vec{\alpha}_S = \bar{\bar{T}}_S \left(\vec{\alpha}_o + \dot{\vec{\omega}}_i\right) \tag{110}$$

where the absolute sensor acceleration $\vec{a}_p$ is given by equation (27). Note that the sensor linear acceleration $\vec{a}_S$ includes gravity, in order to mimic the signal from an actual accelerometer. The aircraft-frame's linear acceleration $\vec{a}_o$ (which is a component of $\vec{a}_p$), and the aircraft-frame's angular acceleration $\vec{\alpha}_o$, are defined by equations (16) and (17).

The sensor axes in earth coordinates are given by

$$\begin{bmatrix} - & \hat{V}_E & - \\ - & \hat{\beta}_E & - \\ - & \hat{\alpha}_E & - \end{bmatrix} \equiv \begin{bmatrix} V_X & V_Y & V_Z \\ \beta_X & \beta_Y & \beta_Z \\ \alpha_X & \alpha_Y & \alpha_Z \end{bmatrix} = \begin{bmatrix} - & \hat{V} & - \\ - & \hat{\beta} & - \\ - & \hat{\alpha} & - \end{bmatrix} \begin{bmatrix} & & \\ & \bar{\bar{T}}_E & \\ & & \end{bmatrix}^T \tag{111}$$

and the apparent aircraft Euler angles as seen by the sensor are then determined using its axes' earth components as follows.

$$\Psi_S = \arctan\left(-V_Y/V_X\right) \tag{112}$$

$$\Theta_S = \arctan\left(-V_Z/\sqrt{V_X^2 + V_Y^2}\right) \tag{113}$$

$$\Phi_S = \arctan\left(-\beta_Z/\alpha_Z\right) \tag{114}$$

A GPS unit at the sensor location will report its earth-based position and velocity with no effect from the sensor orientation.

$$\vec{r}_E = \vec{R}_E + \bar{\bar{T}}_E \,\vec{r}_p \tag{115}$$

$$\vec{v}_E = \bar{\bar{T}}_E \,\vec{v}_p \tag{116}$$

11 Equation system

11.1 Variables

The variables in the overall unsteady problem are

$$\begin{aligned} \mathbf{x} = \Big(& \vec{r}_i \;\; \vec{\theta}_i \;\; \vec{M}_i \;\; \vec{F}_i \;\; \vec{u}_i \;\; \vec{\omega}_i \\ & \Delta\vec{r}_J \;\; \Delta\vec{\theta}_J \;\; \vec{M}_J \;\; \vec{F}_J \;\; A_1 \;\; A_2 \ldots A_K \;\; \vec{R}_E \;\; \vec{\Theta} \;\; \vec{U} \;\; \vec{\Omega} \;\; \vec{a}_o \;\; \vec{\alpha}_o \\ & \delta_{F_1} \;\; \delta_{F_2} \ldots \;\; \Delta_{e_1} \;\; \Delta_{e_2} \ldots \;\; \Delta_{g_1} \;\; \Delta_{g_2} \ldots \;\; \mathbf{e} \Big) \end{aligned}$$

$$\begin{aligned} \dot{\mathbf{x}} = \Big(& \dot{\vec{r}}_i \;\; \dot{\vec{\theta}}_i \;\; \dot{\vec{u}}_i \;\; \dot{\vec{\omega}}_i \\ & \dot{A}_1 \;\; \dot{A}_2 \ldots \dot{A}_K \;\; \dot{\vec{R}}_E \;\; \dot{\vec{\Theta}} \;\; \dot{\vec{U}} \;\; \dot{\vec{\Omega}} \;\; \dot{\mathbf{e}} \Big) \end{aligned}$$

$$\mathbf{y} = \begin{pmatrix} V_S & \alpha_S & \beta_S & \Phi_S & \Theta_S & \Psi_S & \ldots \end{pmatrix}$$

$$\mathbf{u} = \begin{pmatrix} V_c & \alpha_c & \beta_c & \Phi_c & \Theta_c & \Psi_c & (\delta_{F1})_c & (\delta_{F2})_c \ldots \end{pmatrix}$$

where $\mathbf{y}(\mathbf{x}, \dot{\mathbf{x}}, \mathbf{u})$ are dependent sensor variables, and $\mathbf{u}$ are commanded variables. In principle, any variable which can be evaluated as a function $f_c(\mathbf{x}, \dot{\mathbf{x}})$, either directly or via $\mathbf{y}(\mathbf{x}, \dot{\mathbf{x}}, \mathbf{0})$, can appear in $\mathbf{u}$. The ones shown in $\mathbf{u}$ above are the ones most likely to be actually needed in order to implement aircraft control laws (e.g. hold airspeed at V_c, hold heading at Ψ_c). The additional vector $\mathbf{e}$ which augments $\mathbf{x}$ is the error-integral vector, defined as

$$\mathbf{e} = \int^t \left\{ \begin{array}{c} V(\mathbf{x}) - V_c \\ \alpha(\mathbf{x}) - \alpha_c \\ \vdots \end{array} \right\} dt$$

for some or all of the components of $\mathbf{u}$. It is used to implement PI or PID control laws.

It should be pointed out the state rate vector $\dot{\mathbf{x}}$ is smaller than the state vector $\mathbf{x}$, simply because not all the components in $\mathbf{x}$ appear as rates in the governing equations. For example, the governing beam equations do not depend on the bending moment rates $\dot{\vec{M}}_i$ in any way, so these are omitted from $\dot{\mathbf{x}}$.

11.2 Governing equations

11.2.1 Beam and global equations

Most of the governing equations for the variables in $\mathbf{x}$ have been developed previously. The beam variables $\vec{r}_i\ \vec{\theta}_i\ \vec{M}_i\ \vec{F}_i$, joint variables $\Delta\vec{r}_J\ \Delta\vec{\theta}_J\ \vec{M}_J\ \vec{F}_J$, and circulation variables $A_1\ A_2 \ldots A_K$ are associated with their usual steady equations, but which now include inertia and unsteady aero force terms. The beam node velocities and rates $\vec{u}_i\ \vec{\omega}_i$, are associated with their simple definitions (2). Similarly, the global position and rate variables $\vec{R}_E\ \vec{\Theta}\ \vec{U}\ \vec{\Omega}$ are associated with their kinematic relations and definitions (13), (14), (16), (17). The absolute accelerations $\vec{a}_o$ and $\vec{\alpha}_o$ are constrained by either the direct forced-motion constraints (18) and (19), or the indirect free-flight constraints (20) and (21), as appropriate for the case at hand.

11.2.2 Error-integral equations

The unsteady equations associated with the error-integral vector $\mathbf{e}$ follow directly from its definition.

$$\dot{\mathbf{e}} - \left\{ \begin{array}{c} V(\mathbf{x}) - V_c \\ \alpha(\mathbf{x}) - \alpha_c \\ \vdots \end{array} \right\} = \mathbf{0} \tag{117}$$

11.2.3 Control equations

The control variables in the state vector, namely $\delta_F, \Delta_e, \Delta_g$, are associated with control-law equations. A general form for these is the following nonlinear residual expression.

$$\mathbf{c}(\mathbf{x}, \mathbf{y}; \mathbf{u}) = \mathbf{0} \tag{118}$$

The simplest example, with one control variable present, is a direct "commanded-stick" constraint,

$$\mathbf{c} \;\equiv\; \delta \,-\, \delta_{\mathrm{c}}(t) \;=\; 0 \tag{119}$$

where δ represents any of the control variables in $\mathbf{x}$. A more elaborate example might be a state-feedback law, such as

$$\mathbf{c} \;\equiv\; \delta \,+\, f(\mathbf{y};\mathbf{u}) \;=\; 0 \tag{120}$$

where $f()$ implements a controller which tries to make some output in $\mathbf{y}$ (e.g. sensed airspeed) follow its commanded value in $\mathbf{u}$.

For the gust variables Δ_g, the current implementation uses the explicit contraint (119), which is appropriate for prescribed-gust encounter simulations. For the remaining flap and engine control variables δ_F and Δ_e, one can select either the same explicit constraint (119), or the general form (118). In the latter case, the control law function $\mathbf{c}(\mathbf{x},\mathbf{y};\mathbf{u})$ must be provided by the user in the form of a black-box subroutine `CONTROL`, which returns numerical values of $\mathbf{c}$, $\partial\mathbf{c}/\partial\mathbf{x}$, and $\partial\mathbf{c}/\partial\mathbf{y}$. The derivatives are needed to construct the Jacobians for the overall Newton system, which includes the linearized form of (118). The forcing vector $\mathbf{u}(t)$ is imposed interactively at runtime, for both types of control equations. This allows rapid investigation of the system response to specified forcing (e.g. step inputs, doublet inputs, etc), for both the open-loop and closed-loop cases.

11.3 Linearization

All the beam, global, and control equations discussed in the previous section can be written in residual form as follows.

$$\mathbf{r}(\mathbf{x},\dot{\mathbf{x}},\mathbf{u}) \;=\; \mathbf{0} \tag{121}$$

In principle, $\mathbf{r}()$ also depends on the sensor variables $\mathbf{y}$, via the control equations (118) which are contained in $\mathbf{r}()$. But this direct dependence on $\mathbf{y}$ is now treated as a dependence on $\mathbf{x}$,$\mathbf{u}$ using the $\mathbf{y}(\mathbf{x},\mathbf{u})$ definitions and the chain rule. The general perturbation of the residual is given via three Jacobian matrix terms.

$$\delta\mathbf{r} \;=\; \left[\frac{\partial\mathbf{r}}{\partial\mathbf{x}}\right]\delta\mathbf{x} \;+\; \left[\frac{\partial\mathbf{r}}{\partial\dot{\mathbf{x}}}\right]\delta\dot{\mathbf{x}} \;+\; \left[\frac{\partial\mathbf{r}}{\partial\mathbf{u}}\right]\delta\mathbf{u} \tag{122}$$

This linearized form is used in a number of ways in the subsequent sections.

12 Time-Domain Calculation

12.1 Time discretization

A time-marching calculation consists of a sequence of solutions $\mathbf{x}_0$, $\mathbf{x}_1 \ldots \mathbf{x}_{n-1}$, $\mathbf{x}_n$ at discrete times t_0, $t_1 \ldots t_{n-1}$, t_n. The solution time derivative at any time level is then defined using a suitable backward difference in time, e.g.

$$\dot{\mathbf{x}}_n \;=\; k_0\,\mathbf{x}_n \;+\; k_1\,\mathbf{x}_{n-1} \;+\; k_2\,\mathbf{x}_{n-2} \tag{123}$$

The differencing coefficients are

$$k_0 \;=\; \frac{1}{t_n - t_{n-1}}$$

$$
\begin{aligned}
k_1 &= \frac{-1}{t_n - t_{n-1}} \\
k_2 &= 0
\end{aligned}
\tag{124}
$$

for first-order differencing, and

$$
\begin{aligned}
k_0 &= \frac{1}{t_n - t_{n-1}} + \frac{1}{t_n - t_{n-2}} \\
k_1 &= \frac{-1}{t_n - t_{n-1}} + \frac{-1}{t_n - t_{n-2}} - \frac{t_n - t_{n-1}}{(t_n - t_{n-2})(t_{n-1} - t_{n-2})} \\
k_2 &= \qquad\qquad\qquad\qquad\qquad\quad \frac{t_n - t_{n-1}}{(t_n - t_{n-2})(t_{n-1} - t_{n-2})}
\end{aligned}
\tag{125}
$$

for second-order differencing. Note that setting

$$
k_0 \;=\; 0 \qquad\qquad k_1 \;=\; 0 \qquad\qquad k_2 \;=\; 0
$$

recovers the steady-state case. This is also equivalent to taking an "infinite time step", or $t_n \to \infty$.

12.2 Time-marching calculation

The objective at each time level is to find the $\mathbf{x}_n$ which satisfies the residual equations (121).

$$
\mathbf{r}\left(\mathbf{x}_n, \dot{\mathbf{x}}_n, \mathbf{u}_n\right) \;=\; \mathbf{0} \tag{126}
$$

This is accomplished by using Newton iteration via the linearized form (122). We first define $\delta\mathbf{x}$ to be the change in $\mathbf{x}_n$ from one Newton iteration to the next. Denoting the Newton iteration level by a superscript i, we then have

$$
\mathbf{x}_n^{i+1} \;=\; \mathbf{x}_n^i \;+\; \delta\mathbf{x}
$$

and determine $\delta\mathbf{x}$ so as to drive the residual vector at the next iteration to zero.

$$
\mathbf{r}(\mathbf{x}_n^{i+1}, \dot{\mathbf{x}}_n^{i+1}, \mathbf{u}_n) \;\simeq\; \mathbf{r}^i \;+\; \left[\frac{\partial \mathbf{r}}{\partial \mathbf{x}}\right]^i \delta\mathbf{x} \;+\; \left[\frac{\partial \mathbf{r}}{\partial \dot{\mathbf{x}}}\right]^i \delta\dot{\mathbf{x}} \;+\; \left[\frac{\partial \mathbf{r}}{\partial \mathbf{u}}\right]^i \delta\mathbf{u} \;=\; \mathbf{0} \tag{127}
$$

From (123), we have $\delta\dot{\mathbf{x}}_n = k_0\, \delta\mathbf{x}_n$, which then allows combining of the unsteady term in (127) with the steady term. Also, with the commanded state $\mathbf{u}_n$ being set to its prescribed value at each time level, its correction is zero, or $\delta\mathbf{u} = 0$. Hence, we can discard the commanded-state Jacobian term from (127). The final resulting Newton system is

$$
\left[\frac{\partial \mathbf{r}}{\partial \mathbf{x}} + k_0 \frac{\partial \mathbf{r}}{\partial \dot{\mathbf{x}}}\right]^i \delta\mathbf{x} \;=\; -\mathbf{r}^i \tag{128}
$$

Using the state at the previous time level as the initial guess,

$$
\mathbf{x}_n^0 \;=\; \mathbf{x}_{n-1}
$$

system (128) is set up and solved for $\delta\mathbf{x}$, which is then used to obtain an improved guess for the current state.

$$
\mathbf{x}_n^{i+1} \;=\; \mathbf{x}_n^i \;+\; \delta\mathbf{x} \tag{129}
$$

Convergence of the Newton iterates $\mathbf{x}_n^0, \mathbf{x}_n^1, \mathbf{x}_n^2 \ldots \mathbf{x}_n$ is usually very rapid, requiring no more than 3 or 4 iterations for $\delta\mathbf{x}$ to reach machine zero for most cases. However, up to 10 or even 20 iterations may be required if a very large time step is used, although temporal accuracy is likely to be suspect in this case.

13 Perturbation analyses

13.1 General perturbation form

We now assume that the Newton iteration has been performed to convergence, producing a quasi-steady or unsteady (time-marched) state $\mathbf{x}$, with some prescribed state $\mathbf{u}$. These satisfy the governing residual equations, considered previously.

$$\mathbf{r}(\mathbf{x}, \dot{\mathbf{x}}, \mathbf{u}) \;=\; \mathbf{0} \tag{130}$$

Small unsteady perturbations $\delta\mathbf{x}, \delta\dot{\mathbf{x}}, \delta\mathbf{u}$ about this state are now considered. These are governed by the general linearizing relation (122). By requiring that the residuals remain at zero, or setting $\delta\mathbf{r} = \mathbf{0}$, all the governing equations remain satisfied, and hence the perturbations are physical. Relation (122) then becomes

$$\mathbf{M}\,\delta\dot{\mathbf{x}} \;=\; \mathbf{A}\,\delta\mathbf{x} \,+\, \mathbf{B}\,\delta\mathbf{u} \tag{131}$$

where

$$\mathbf{M} \;=\; -\frac{\partial\mathbf{r}}{\partial\dot{\mathbf{x}}} \qquad\qquad \mathbf{A} \;=\; \frac{\partial\mathbf{r}}{\partial\mathbf{x}} \qquad\qquad \mathbf{B} \;=\; \frac{\partial\mathbf{r}}{\partial\mathbf{u}} \tag{132}$$

The system Jacobian matrices, evaluated at the converged state, are already available from the last iteration of the Newton solver. They are renamed $\mathbf{M}, \mathbf{A}, \mathbf{B}$ to match traditional notation. We also linearize the sensor output $\mathbf{y}(\mathbf{x}, \dot{\mathbf{x}}, \mathbf{u})$ relation,

$$\delta\mathbf{y} \;=\; \mathbf{C}\,\delta\mathbf{x} \,+\, \mathbf{Q}\,\delta\dot{\mathbf{x}} \,+\, \mathbf{D}\,\delta\mathbf{u} \tag{133}$$

where

$$\mathbf{C} \;=\; \frac{\partial\mathbf{y}}{\partial\mathbf{x}} \qquad\qquad \mathbf{Q} \;=\; \frac{\partial\mathbf{y}}{\partial\dot{\mathbf{x}}} \qquad\qquad \mathbf{D} \;=\; \frac{\partial\mathbf{y}}{\partial\mathbf{u}} \tag{134}$$

The linearized governing equation system (131) and linearized sensor output relation (133) will be the starting points for the subsequent analyses.

Strictly speaking, the perturbation $\delta\mathbf{x}$ computed from (131) is physically valid only if $\mathbf{x}$ itself is a *trimmed* state, defined as a state which has time-invariant system matrices $\mathbf{M}$,$\mathbf{A}$,$\mathbf{B}$. This requires a steady vertical earth velocity U_Z, and a constant turn radius as seen projected onto the earth X-Y plane. The most general case of this type is a helical flight path at constant velocity, constant climb rate, and constant bank angle. The simplest case of this type is straight and level flight, with zero climb rate and zero bank angle.

Nevertheless, it is possible to compute $\delta\mathbf{x}(t)$ at any instant in time during a non-trimmed flight, but this perturbation must be interpreted with care. In general, this perturbation will be meaningful only for time scales significantly shorter than the time scale over which the $\mathbf{M}$,$\mathbf{A}$,$\mathbf{B}$ system matrices are changing. Roughly speaking, the "trimmed condition" must be changing much more slowly than the perturbation itself. A more rigorous treatment of this situation would require a WKB analysis, which is beyond scope here.

13.2 Frequency-domain analysis

The commanded forcing vector $\delta\mathbf{u}$ and resulting response $\delta\mathbf{x}$ are now assumed to have infinitesimal oscillatory perturbations of the form

$$\delta\mathbf{u} \;=\; \hat{\mathbf{u}}\,\exp(i\omega t) \qquad\qquad \delta\mathbf{x} \;=\; \hat{\mathbf{x}}\,\exp(i\omega t) \qquad\qquad \delta\dot{\mathbf{x}} \;=\; i\omega\,\hat{\mathbf{x}}\,\exp(i\omega t)$$

with the response $\delta\mathbf{x}$ being obtained by solving the perturbed system (131).

$$[\,i\omega\,\mathbf{M} - \mathbf{A}]\,\hat{\mathbf{x}} \;=\; \mathbf{B}\,\hat{\mathbf{u}} \tag{135}$$

Sensor output perturbations $\delta\mathbf{y} = \hat{\mathbf{y}}\,\exp(i\omega t)$ are then computed from (133).

$$\hat{\mathbf{y}} \;=\; [\mathbf{C} + i\omega\mathbf{Q}]\,\hat{\mathbf{x}} \;+\; \mathbf{D}\hat{\mathbf{u}} \tag{136}$$

In practice, only one component of $\hat{\mathbf{u}}$ is chosen to be nonzero at a time, typically one of the control vector variables, e.g.

$$\begin{aligned}\hat{\mathbf{u}} \;&=\; \{\ldots 0\,,\,0\,,\,\hat{\delta}_F\,,\,0\,,\,0\,,\,0\,,\,0\ldots\}^T \\ \hat{\mathbf{u}} \;&=\; \{\ldots 0\,,\,0\,,\,0\,,\,\hat{\Delta}_e,\,0\,,\,0\,,\,0\ldots\}^T\end{aligned}$$

In each case, the magnitude of the single nonzero component is arbitrary, and can be conveniently set to unity, e.g. $\hat{\delta}_F = 1$ or $\hat{\Delta}_e = 1$.

The case of the harmonic gust is slightly different, since this has a spatial influence which is complex. As described in a previous section, this was implemented via the gust control variables $\Delta_{g_{1\ldots4}}$ which were assumed independent. Here they must be combined appropriately. Taking the symmetric vertical gust mode as an example, the harmonic unsteady forcing quantity appearing in the gust mode field (88) is

$$\mathcal{A}_{Z_n}\;\exp(i\omega t) \;=\; \hat{\mathcal{A}}_{Z_n}\;\exp(i\omega t)$$

so that

$$\hat{\mathbf{u}} \;=\; \{\ldots 0\,,\,0\,,\,0\,,\,0\,,\,\hat{\mathcal{A}}_{Z_n},\,0\,,\,0\ldots\}^T$$

with a unit amplitude $\hat{\mathcal{A}}_{Z_n} = 1$ set as usual. According to the constraints (91) – (94) we must then set the gust control variable changes as follows.

$$\begin{aligned}\delta\Delta_{g_1} \;&=\; \hat{\mathcal{A}}_{Z_n}\;\exp(i\omega t) && (137)\\ \delta\Delta_{g_2} \;&=\; i\,\hat{\mathcal{A}}_{Z_n}\;\exp(i\omega t) && (138)\\ \delta\Delta_{g_3} \;&=\; 0 && (139)\\ \delta\Delta_{g_4} \;&=\; 0 && (140)\end{aligned}$$

The gust then produces the residual perturbation

$$\begin{aligned}\delta\mathbf{r} \;&=\; \frac{\partial\mathbf{r}}{\partial\Delta_{g_1}}\,\delta\Delta_{g_1} \;+\; \frac{\partial\mathbf{r}}{\partial\Delta_{g_2}}\,\delta\Delta_{g_2} \;+\; \frac{\partial\mathbf{r}}{\partial\Delta_{g_3}}\,\delta\Delta_{g_3} \;+\; \frac{\partial\mathbf{r}}{\partial\Delta_{g_4}}\,\delta\Delta_{g_4} && (141)\\ &=\; \left[\frac{\partial\mathbf{r}}{\partial\Delta_{g_1}} \;+\; i\,\frac{\partial\mathbf{r}}{\partial\Delta_{g_2}}\right]\hat{\mathcal{A}}_{Z_n}\;\exp(i\omega t) && (142)\end{aligned}$$

By implication, the quantity in the brackets must be

$$\frac{\partial\mathbf{r}}{\partial\mathcal{A}_{Z_n}} \;=\; \frac{\partial\mathbf{r}}{\partial\Delta_{g_1}} \;+\; i\,\frac{\partial\mathbf{r}}{\partial\Delta_{g_2}} \tag{143}$$

This is the complex $\hat{\mathcal{A}}_{Z_n}$ column in the $\mathbf{B}$ matrix in (132) and (135), given in terms of the real $\partial\mathbf{r}/\partial\Delta_{g_1}$ and $\partial\mathbf{r}/\partial\Delta_{g_2}$ vectors which are actually computed. The $\mathbf{B}$ columns for the other forcing variables such as $\hat{\delta}_F$ are normally real. Whether the matrices are real or complex doesn't matter at this stage, since the entire Bode system (135) is implemented in complex arithmetic anyway.

The dynamic control response versus frequency is concisely summarized for any $\hat{\mathbf{x}}$ component $\hat{x}_i$ by the gain and phase relative to the nonzero forcing variable $\hat{u}_j$.

$$\left|\frac{\hat{x}_i}{\hat{u}_j}\right| \qquad , \qquad \arg\left(\frac{\hat{x}_i}{\hat{u}_j}\right)$$

These are defined for each perturbed component $\hat{u}_j$ being considered. An alternative way to display the frequency response is with broken-loop gain and phase, defined as follows.

$$\left|1 - \frac{\hat{u}_j}{\hat{x}_i}\right| \qquad , \qquad \arg\left(1 - \frac{\hat{u}_j}{\hat{x}_i}\right)$$

The structure of the response vector $\hat{\mathbf{x}}$ is most readily displayed by using it to perturb the current base solution $\mathbf{x}$.

$$\mathbf{x}' = \mathbf{x} + c\,\Re\{\hat{\mathbf{x}}\exp(i\phi)\}$$

The scaling factor c and phase ϕ are provided by the user at runtime, with $c = 1$ corresponding to a unity $\hat{u}_j$ forcing magnitude. Setting $\phi = \omega t$ produces a real-time movie of the response.

13.3 Eigenmode analyses

We now consider (131) for the unforced case, with $\delta\mathbf{u} = 0$. We assume a perturbation solution of the form

$$\delta\mathbf{x}(t) = \hat{\mathbf{x}}\exp(\lambda t)$$

which simplifies (131) to

$$\mathbf{A}\,\hat{\mathbf{x}} = \mathbf{M}\,\hat{\mathbf{x}}\,\lambda \tag{144}$$

Eigenvalues and eigenvectors (or eigenmodes) are nontrivial solutions λ_k,$\mathbf{v}_k$ to the unforced perturbed system (144),

$$\mathbf{A}\,\mathbf{v}_k = \mathbf{M}\,\mathbf{v}_k\,\lambda_k \tag{145}$$

where $k = 1, 2...$ is the solution mode index. For each eigenvalue λ_k, it is also useful to determine the system's left-eigenvector $\mathbf{w}_k$ from

$$\mathbf{w}_k^*\,\mathbf{A} = \mathbf{w}_k^*\,\mathbf{M}\,\lambda_k \tag{146}$$

where $(\;)^*$ denotes the complex transpose operation (transpose together with complex conjugation).

13.3.1 Eigenmode computation

The eigenvalues λ_k and corresponding eigenvectors $\mathbf{v}_k$ are generated by recasting system (145) in the equivalent "shifted power-iteration" form.

$$\mathbf{H}\,\mathbf{v}_k = \mathbf{v}_k\,\nu_k \tag{147}$$

$$\mathbf{H} \equiv [\mathbf{A} - \mu\mathbf{M}]^{-1}\,\mathbf{M} \tag{148}$$

$$\nu_k \equiv \frac{1}{\lambda_k - \mu} \tag{149}$$

An Arnoldi iteration method using $\mathbf{H}$ will converge to the largest $|\nu_k|$, which corresponds to the λ_k which is closest to the shift μ in the complex plane. The corresponding eigenvector $\mathbf{v}_k$ is also obtained as a by-product. Hence, one can selectively choose μ to focus the search on any nearby eigenvalues. Typically, one would choose $\mu \simeq 0 + 0i$ to search for low-frequency modes such as the phugoid and spiral modes. Similarly, one would choose $\mu = 0 + i\omega$ to search weakly-damped or weakly-unstable (e.g. flutter) modes at an anticipated frequency ω. ASWING currently employs the ARPACK [3] sparse-system eigenvalue package to rapidly compute the eigenpairs. The matrix inversion in (148) is implemented via LU factorization. This is performed only once, and only back-substitution operations are required for all subsequent Arnoldi iterations.

The left-eigenvector $\mathbf{w}_k$ corresponding to each $\lambda_k, \mathbf{v}_k$ pair is obtained via inverse power iteration. Starting with a random vector $\tilde{\mathbf{w}}^0$, one iteration step is executed as follows:

$$\mathbf{w}' = \left[\mathbf{A}^T - \lambda_k^* \mathbf{M}^T\right]^{-1} \left\{\mathbf{M}^T \tilde{\mathbf{w}}^i\right\} \tag{150}$$

$$\tilde{\mathbf{w}}^{i+1} = \frac{\mathbf{w}'}{\mathbf{w}'^* \, \mathbf{M} \, \mathbf{v}_k} \tag{151}$$

The matrix inversion in (150) is implemented via a complex LU factorization, performed once and then re-used for each iteration. The factored matrix is very nearly singular, making $\mathbf{w}'$ typically very large. But because of the normalization operation (151), the $\tilde{\mathbf{w}}^0, \tilde{\mathbf{w}}^1 \ldots \mathbf{w}_k$ sequence converges stably. In fact, convergence is extremely rapid, with only 2 or 3 iterations being needed to reach machine zero accuracy. The normalization operation (151) also makes the resulting set of $\mathbf{v}_k$,$\mathbf{w}_k$ pairs satisfy the following $\mathbf{M}$-orthonormality condition.

$$\mathbf{w}_k^* \, \mathbf{M} \, \mathbf{v}_\ell = \begin{cases} 1 \;, & \text{if } \; k = \ell \\ 0 \;, & \text{if } \; k \neq \ell \end{cases} \tag{152}$$

13.3.2 Forced response analysis

The availability of eigenmodes allows deriving a concise expression for the solution to the Bode forced-response system (135). Although direct solution of (135) is what's actually employed in ASWING, having an explicit form for the solution aids in physical interpretations. We define

$$\mathbf{V} = \begin{bmatrix} | & | & \\ \mathbf{v}_1 & \mathbf{v}_2 & \ldots \\ | & | & \end{bmatrix} \qquad \mathbf{W} = \begin{bmatrix} | & | & \\ \mathbf{w}_1 & \mathbf{w}_2 & \ldots \\ | & | & \end{bmatrix} \qquad \mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & & \\ & \lambda_2 & \\ & & \ddots \end{bmatrix} \qquad \mathbf{z} = \begin{Bmatrix} z_1 \\ z_2 \\ \vdots \end{Bmatrix} \tag{153}$$

so that all the eigensolutions can be combined into single matrix relations.

$$\mathbf{A}\,\mathbf{V} = \mathbf{M}\,\mathbf{V}\,\mathbf{\Lambda} \tag{154}$$

$$\mathbf{W}^*\mathbf{A} = \mathbf{\Lambda}\mathbf{W}^*\mathbf{M} \tag{155}$$

$$\mathbf{W}^*\mathbf{M}\,\mathbf{V} = \mathbf{I} \tag{156}$$

We now expand the harmonic solution response vector $\delta\mathbf{x}$ in terms of the eigenvectors as follows.

$$\delta\mathbf{x} = \mathbf{V}\,\mathbf{z} \tag{157}$$

$$\hat{\mathbf{x}} = \mathbf{V}\,\hat{\mathbf{z}} \tag{158}$$

This allows the following manipulation of the forced-response system (135).

$$
\begin{aligned}
i\omega \mathbf{M}\mathbf{V}\hat{\mathbf{z}} \;-\; \mathbf{A}\mathbf{V}\hat{\mathbf{z}} &= \mathbf{B}\,\hat{\mathbf{u}} && (159)\\
\mathbf{M}\mathbf{V}[\,i\omega\mathbf{I} - \boldsymbol{\Lambda}]\,\hat{\mathbf{z}} &= \mathbf{B}\,\hat{\mathbf{u}} && (160)
\end{aligned}
$$

This is then diagonalized by premultiplying by $\mathbf{W}^*$ and using relation (156).

$$
\begin{aligned}
[\,i\omega\mathbf{I} - \boldsymbol{\Lambda}]\,\hat{\mathbf{z}} &= \mathbf{W}^*\mathbf{B}\,\hat{\mathbf{u}} && (161)\\
\hat{\mathbf{z}} &= [\,i\omega\mathbf{I} - \boldsymbol{\Lambda}]^{-1}\,\mathbf{W}^*\mathbf{B}\,\hat{\mathbf{u}} && (162)\\
\hat{z}_k &= \frac{1}{i\omega - \lambda_k}\,\mathbf{w}_k^*\mathbf{B}\,\hat{\mathbf{u}} && (163)
\end{aligned}
$$

The gain and phase of each state vector component relative to a forcing vector component is then given by summing of the eigenmodes in (158), using the $\hat{z}_k$ solution (163).

$$
\frac{\hat{x}_i}{\hat{u}_j} = \sum_k \hat{z}_k\,\frac{(v_k)_i}{\hat{u}_j} \qquad (164)
$$

Resonance is expected whenever any of the eigenvalues λ_k is close to $i\omega$ on the imaginary axis. In equation (163) this will give a large $1/(i\omega - \lambda_k)$ and hence a large $\hat{z}_k$, which will then dominate the modal summation in (164).

13.3.3 Stability analysis

Consider now the time-evolution equation (131) as an initial-value problem with no forcing.

$$
\begin{aligned}
\mathbf{M}\,\delta\dot{\mathbf{x}} &= \mathbf{A}\,\delta\mathbf{x} && (165)\\
\text{at } t=0: \quad \delta\mathbf{x} &= \delta\mathbf{x}_0 && (166)\\
\text{for all } t: \quad \delta\mathbf{u} &= 0 && (167)
\end{aligned}
$$

The resulting unsteady evolution is again expanded in terms of eigenmodes,

$$
\begin{aligned}
\delta\mathbf{x}(t) &= \mathbf{V}\,\mathbf{z}(t) && (168)\\
z_k(t) &= \hat{z}_k\,\exp(\lambda_k t) && (169)
\end{aligned}
$$

which automatically satisfies the governing equation (165).

$$
\begin{aligned}
\mathbf{M}\,\mathbf{V}\dot{\mathbf{z}} &= \mathbf{A}\,\mathbf{z} && (170)\\
\mathbf{M}\,\mathbf{V}\boldsymbol{\Lambda}\hat{\mathbf{z}} &= \mathbf{A}\,\mathbf{V}\,\hat{\mathbf{z}} && (171)
\end{aligned}
$$

The final expression (171) is satisfied by virtue of (154).

The initial eigenmode amplitudes $z_{k_0} = z_k(t{=}0) = \hat{z}_k$ are determined from the initial state $\delta\mathbf{x}_0$, making use of the left eigenvectors and the normalizing condition (152).

$$
\begin{aligned}
\delta\mathbf{x}_0 &= \mathbf{V}\,\mathbf{z}_0 && (172)\\
\mathbf{W}^*\,\mathbf{M}\,\delta\mathbf{x}_0 &= \mathbf{W}^*\,\mathbf{M}\,\mathbf{V}\,\mathbf{z}_0 && (173)\\
\mathbf{W}^*\,\mathbf{M}\,\delta\mathbf{x}_0 &= \mathbf{z}_0 && (174)\\
\mathbf{w}_k^*\,\mathbf{M}\,\delta\mathbf{x}_0 &= z_{k_0} && (175)
\end{aligned}
$$

The time evolution of the state perturbation can then be explicitly written as follows.

$$\delta\mathbf{x}(t) \;=\; \mathbf{V}\,\mathbf{z}(t) \;=\; \sum_k \mathbf{v}_k\, z_k(t) \;=\; \sum_k \mathbf{v}_k\, z_{k_0}\, \exp(\lambda_k t) \tag{176}$$

Dynamic instability of any eigenmode k is indicated if $\Re(\lambda_k) > 0$. The corresponding eigenvector $\mathbf{v}_k$ indicates the nature of the instability. Unstable aircraft stability modes (e.g. spiral mode) have their eigenvectors dominated by rigid-body motion components such as $\hat{R}_E$, $\hat{\Theta}$, etc. Unstable structural modes (e.g. flutter) have their eigenvectors dominated by beam-variable components $\hat{r}$, $\hat{\theta}$, $\hat{M}$, $\hat{F}$. In highly flexible aircraft, unstable modes can easily have significant components of both types, blurring the distinction between "aircraft instability" and "flutter instability".

13.3.4 Reduced-order model construction

We now use the eigenmodes to simplify a control-law design problem. The state vector perturbation is first expanded in terms of the eigenmodes.

$$\mathbf{x}(t) \;=\; \delta\mathbf{x}(t) \,+\, \mathbf{x}_o \tag{177}$$

$$\delta\mathbf{x}(t) \;=\; \mathbf{V}\,\mathbf{z}(t) \tag{178}$$

This is the same approach used in expansion (157), except that the $\mathbf{z}$ vector is now an arbitrary function of time rather being assumed to be harmonic. The baseline state is also added separately, and is now denoted by $\mathbf{x}_o$. With the eigenvector matrix $\mathbf{V}$ assumed frozen and of full rank, $\mathbf{z}(t)$ can be considered as an alternative state vector equivalent to $\mathbf{x}(t)$.

In practice, the dimension of $\mathbf{z}$ can be made much smaller by ignoring most of the eigenmodes. This mode truncation is one example of *Reduced-Order Modeling* (ROM). A rational choice is to retain only those "slow" modes which have the smallest $|\lambda_k|$, since these are the ones most likely to dominate the dynamics. One exception is fast modes which have very little damping or are unstable (e.g. flutter modes). If these are to be actively stabilized, they would also be included in expansion (157). With most of the modes ignored, the $\mathbf{V}$ and $\mathbf{W}$ matrices become rectangular, with many fewer columns (eigenmodes) than rows (state vector components).

Substituting (178) into the linearized state equation (131) gives

$$\mathbf{M}\,\mathbf{V}\,\dot{\mathbf{z}} \;=\; \mathbf{A}\,\mathbf{V}\,\mathbf{z} \,+\, \mathbf{B}\,\delta\mathbf{u} \tag{179}$$

which is then diagonalized by premultiplying by $\mathbf{W}^*$.

$$\mathbf{W}^*\,\mathbf{M}\,\mathbf{V}\,\dot{\mathbf{z}} \;=\; \mathbf{W}^*\,\mathbf{A}\,\mathbf{V}\,\mathbf{z} \,+\, \mathbf{W}^*\,\mathbf{B}\,\delta\mathbf{u} \tag{180}$$

$$\dot{\mathbf{z}} \;=\; \mathbf{\Lambda}\,\mathbf{z} \,+\, \mathbf{W}^*\,\mathbf{B}\,\delta\mathbf{u} \tag{181}$$

It is also of interest to compute the perturbation of the output sensor vector $\mathbf{y}(\mathbf{x}, \dot{\mathbf{x}}, \mathbf{u})$.

$$\delta\mathbf{y} \;=\; \mathbf{C}\,\delta\mathbf{x} \,+\, \mathbf{Q}\,\delta\dot{\mathbf{x}} \,+\, \mathbf{D}\,\delta\mathbf{u} \tag{182}$$

Recasting this using (178), we have

$$\delta\mathbf{y} \;=\; \mathbf{C}\,\mathbf{V}\,\mathbf{z} \,+\, \mathbf{Q}\mathbf{V}\,\dot{\mathbf{z}} \,+\, \mathbf{D}\,\delta\mathbf{u} \tag{183}$$

$$\delta\mathbf{y} \;=\; [\mathbf{C}\,\mathbf{V} \,+\, \mathbf{Q}\mathbf{V}\mathbf{\Lambda}]\;\mathbf{z} \,+\, [\mathbf{D} \,+\, \mathbf{Q}\mathbf{V}\,\mathbf{W}^*\mathbf{B}]\;\delta\mathbf{u} \tag{184}$$

Equations (181) and (184) can be put in the concise form

$$\dot{\mathbf{z}} = \mathbf{\Lambda}\,\mathbf{z} + \mathbf{B}'\,\delta\mathbf{u} \tag{185}$$

$$\delta\mathbf{y} = \mathbf{C}'\,\mathbf{z} + \mathbf{D}'\,\delta\mathbf{u} \tag{186}$$

where

$$\mathbf{B}' = \mathbf{W}^*\,\mathbf{B} \tag{187}$$

$$\mathbf{C}' = \mathbf{C}\,\mathbf{V} + \mathbf{Q}\mathbf{V}\mathbf{\Lambda} \tag{188}$$

$$\mathbf{D}' = \mathbf{D} + \mathbf{Q}\mathbf{V}\,\mathbf{W}^*\mathbf{B} \tag{189}$$

which is a good starting point for designing a suitable control law $\mathbf{u}(\mathbf{x}, \dot{\mathbf{x}})$ in order to obtain a desirable output behavior $\mathbf{y}(t)$.

13.3.5 Forced response from reduced-order model

One application of the ROM is an approximate but very rapid computation of the Bode response $\hat{\mathbf{x}}$ from $\hat{\mathbf{u}}$ for a range of forcing frequencies ω. This is simply the application of the response relations (162),(163),(164) with only the chosen ROM modes included in the mode summation. This method is fast because it eliminates the time-consuming inversion of the Bode coefficient matrix $[i\omega\mathbf{M} - \mathbf{A}]$ in (135) for each frequency ω. However, this ROM-based Bode calculation method requires special care to compute the response of the accelerations $\vec{a}_o$ and $\vec{\alpha}_o$. Although these are components of $\mathbf{x}$, they typically are not properly captured by the eigenmode expansion (158). Specifically, each mode $\mathbf{v}_k$ in this expansion has the following form.

$$\mathbf{v}_k = \left\{ \ldots \hat{U}_k, \hat{\Omega}_k \ldots \hat{a}_{o_k}, \hat{\alpha}_{o_k} \ldots \right\}^T \tag{190}$$

The individual components satisfy the linearized form of the acceleration definitions (16) and (17), restated as

$$\vec{a}_o = \dot{\vec{U}} + \vec{\Omega} \times \vec{U} \tag{191}$$

$$\vec{\alpha}_o = \dot{\vec{\Omega}} \tag{192}$$

and hence

$$\hat{a}_{o_k} = \lambda_k \hat{U}_k + \vec{\Omega} \times \hat{U}_k + \hat{\Omega}_k \times \vec{U} \tag{193}$$

$$\hat{\alpha}_{o_k} = \lambda_k \hat{\Omega}_k \tag{194}$$

The accelerations extracted directly from the overall approximate ROM state (178) will then be

$$\delta\vec{a}_o = \sum_k \left[\lambda_k \hat{U}_k + \vec{\Omega} \times \hat{U}_k + \hat{\Omega}_k \times \vec{U} \right] \hat{z}_k \exp(i\omega t) \tag{195}$$

$$\delta\vec{\alpha}_o = \sum_k \left[\lambda_k \hat{\Omega}_k \right] \hat{z}_k \exp(i\omega t) \tag{196}$$

However, this is not the same as the following alternative response accelerations, computed from (191) and (192) using only the $\hat{U}_k$ and $\hat{\Omega}_k$ components.

$$\delta\vec{a}_o = \sum_k \left[i\omega \hat{U}_k + \vec{\Omega} \times \hat{U}_k + \hat{\Omega}_k \times \vec{U} \right] \hat{z}_k \exp(i\omega t) \tag{197}$$

$$\delta\vec{\alpha}_o = \sum_k \left[i\omega \hat{\Omega}_k \right] \hat{z}_k \exp(i\omega t) \tag{198}$$

The alternative forms (195,196) and (197,198) are seen to be comparable only when $i\omega$ is near one of the λ_k's and that mode dominates the summation, i.e. near resonance, which is what's observed. In general, the second forms (197,198) are observed to be much more accurate across a wide ω range, and are what's actually computed in ASWING's ROM Bode implementation.

As with the accelerations, the Bode responses of the Earth position $\vec{R}_E$ and the heading angle Ψ are also not well captured by their components in the ROM mode set. Again, the reason is that each eigenmode's $\hat{R}_E$ and $\hat{U}$ components are related by a factor of λ_k, while in the actual frequency response they are related by a factor of $i\omega$. The latter option is again chosen, using only the $\hat{U}$ and $\hat{\Omega}$ components in the eigenmode expansion.

$$\delta\vec{U} = \sum_k \hat{U}_k \, \hat{z}_k \, \exp(i\omega t) \tag{199}$$

$$\delta\vec{\Omega} = \sum_k \hat{\Omega}_k \, \hat{z}_k \, \exp(i\omega t) \tag{200}$$

These are then put into the linearized forms of the $\vec{R}_E$ and Ψ evolution equations (13) and (14).

$$\dot{\vec{\Theta}} = \bar{\bar{C}}_E \, \vec{\Omega} \tag{201}$$

$$i\omega \, \delta\vec{\Theta} = \bar{\bar{C}}_E \, \delta\vec{\Omega} + \frac{\partial \bar{\bar{C}}_E}{\partial \vec{\Theta}} \, \delta\vec{\Theta} \, \vec{\Omega} \tag{202}$$

$$\delta\vec{\Theta} = \left[i\omega \, \bar{\bar{\mathrm{I}}} + \frac{\partial \bar{\bar{C}}_E}{\partial \vec{\Theta}} \, \vec{\Omega} \right]^{-1} \bar{\bar{C}}_E \, \delta\vec{\Omega} \tag{203}$$

$$\dot{\vec{R}}_E = \bar{\bar{T}}_E^T \vec{U} \tag{204}$$

$$i\omega \, \delta\vec{R}_E = \bar{\bar{T}}_E^T \, \delta\vec{U} + \frac{\partial \bar{\bar{T}}_E^T}{\partial \vec{\Theta}} \delta\vec{\Theta} \, \vec{U} \tag{205}$$

$$\delta\vec{R}_E = \frac{1}{i\omega} \left[\bar{\bar{T}}_E^T \, \delta\vec{U} + \frac{\partial \bar{\bar{T}}_E^T}{\partial \vec{\Theta}} \delta\vec{\Theta} \, \vec{U} \right] \tag{206}$$

14 Unsteady Propeller Model

14.1 Propeller axes

As described in the Steady ASWING document, three engine-axis unit vectors are defined, $\hat{x}_e$, $\hat{y}_e$, $\hat{z}_e$. The $\hat{x}_e$ vector is along the propeller axis, while $\hat{y}_e$, $\hat{z}_e$ lie in the propeller disk. These define a transformation tensor between the body axes and the engine axes.

$$\bar{\bar{T}}_e \equiv \begin{bmatrix} - & \hat{x}_e & - \\ - & \hat{y}_e & - \\ - & \hat{z}_e & - \end{bmatrix} \tag{207}$$

These axis vectors will in general vary as the engine moves due to structural deformation.

14.2 Propeller velocities and rotation rates

The center of the propeller sees local relative air velocity $\vec{V}_p$, and rotates at some rate $\vec{\Omega}_p$. The corresponding vectors in the propeller axes are

$$\vec{V}_e = \bar{\bar{T}}_e \vec{V}_p = \{V_{\rm eng}\,,\, v_y\,,\, v_z\}^T \tag{208}$$

$$\vec{\Omega}_e = \bar{\bar{T}}_e \vec{\Omega}_p = \{\omega_{\rm eng}\,,\, \omega_y\,,\, \omega_z\}^T \tag{209}$$

which are used in the propeller model to compute the thrust forces and moments. Note that $\omega_{\rm eng}$ is the axial rotation rate of the propeller mount, and is not used here. Specifically, $\omega_{\rm eng}$ is *not* the rotation rate of the propeller itself, which is denoted by $\Omega_{\rm eng}$.

14.3 Primary propeller force and moment

The basic propeller axial thrust and torque are assumed to be the same as in the steady model, and depend only on the axial velocity component $V_{\rm eng}$.

$$F_{\rm eng} = F_{\rm eng}(\Delta_e, \rho, V_{\rm eng}) \tag{210}$$

$$M_{\rm eng} = M_{\rm eng}(\Delta_e, \Omega_{\rm eng}) = -P_{\rm eng}/\Omega_{\rm eng} \tag{211}$$

A possible choice for the engine control variable is the engine shaft power $\Delta_e \equiv P_{\rm eng}$, or alternatively throttle position or battery voltage. These distinctions are not important here.

The remaining velocity and rotation rate components v_y, v_z, ω_y, ω_z determine side loads and moments on the propeller, commonly referred to a "P-factor" forces. These are derived in the following sections.

14.4 Propeller blade element forces

Figure 8 shows a propeller blade element at some radius r, and its velocity and force triangles. The velocity V_d is the axial velocity at the disk, which consists of the axial component of $\vec{V}_p$, plus half of the propeller's slipstream velocity increment $\Delta V_{\rm eng}$, developed in the Steady ASWING document. Only the propeller rotation rate magnitude Ω will be considered for now, and the appropriate force and moment signs will be assigned at the end.

$$V_d = V_{\rm eng} + \frac{1}{2}\Delta V_{\rm eng} \tag{212}$$

$$\Omega = |\Omega_{\rm eng}| \tag{213}$$

This V_d and the relative tangential velocity Ωr give the flow angle ϕ. With the profile drag force neglected, the lift element dL has axial and tangential components dT, dS (Thrust, Sideforce) whose perturbations are given as follows.

$$W = \sqrt{V_d^2 + (\Omega r)^2} \tag{214}$$

$$\phi = \arctan \frac{V_d}{\Omega r} \tag{215}$$

$$dT = dL\, \cos\phi \tag{216}$$

$$dS = dL\, \sin\phi \tag{217}$$

$$\Delta\, dT = \Delta\, dL\, \cos\phi \,-\, dL\, \sin\phi\, \Delta\phi \tag{218}$$

$$\Delta\, dS = \Delta\, dL\, \sin\phi \,+\, dL\, \cos\phi\, \Delta\phi \tag{219}$$

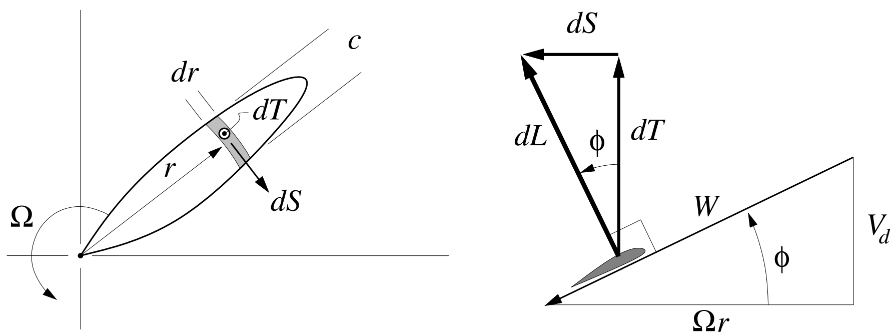

Figure 8: Blade element velocities and forces.

The blade lift perturbation is obtained via some assumed local lift-curve slope C_{L_α}.

$$dL = \frac{1}{2}\rho W^2\, c_\ell\, c\, dr \tag{220}$$

$$\Delta c_\ell = C_{L_\alpha}\,\Delta\alpha = -C_{L_\alpha}\,\Delta\phi \tag{221}$$

$$\Delta\, dL = \frac{1}{2}\rho\,\Delta\Big(W^2\, c_\ell\Big)\; c\, dr = \frac{1}{2}\rho\left(-W^2 C_{L_\alpha}\Delta\phi \;+\; 2W\Delta W\, c_\ell\right)\; c\, dr \tag{222}$$

Substituting these into (218), (219) gives the axial and tangential force perturbations in terms of $\Delta\phi$ and ΔW.

$$\Delta\, dT = \frac{1}{2}\rho\left[W^2\left(-C_{L_\alpha}\cos\phi - c_\ell\,\sin\phi\right)\Delta\phi \;+\; 2W\Delta W c_\ell\,\cos\phi\right] c\, dr \tag{223}$$

$$\Delta\, dS = \frac{1}{2}\rho\left[W^2\left(-C_{L_\alpha}\sin\phi + c_\ell\,\cos\phi\right)\Delta\phi \;+\; 2W\Delta W c_\ell\,\sin\phi\right] c\, dr \tag{224}$$

14.5 Propeller force and moment due to transverse velocity

Figure 9 shows the propeller blade element subjected to a transverse velocity v along either the y_e or z_e direction, which adds an apparent tangential velocity $\Omega r \rightarrow \Omega r + v\,\cos\psi$ seen by the blade element. This produces corresponding changes in the relative flow angle and velocity.

$$W^2\Delta\phi = W^2\frac{\partial\phi}{\partial(\Omega r)}\, v\,\cos\psi = -v\,V_d\,\cos\psi \tag{225}$$

$$W\Delta W = W\frac{\partial W}{\partial(\Omega r)}\, v\,\cos\psi = v\,\Omega r\,\cos\psi \tag{226}$$

The axial and tangential velocity changes are then obtained from (223), (224).

$$\Delta\, dT = \frac{1}{2}\rho\left[\left(C_{L_\alpha}\cos\phi + c_\ell\,\sin\phi\right)V_d \;+\; 2c_\ell\,\cos\phi\,\Omega r\right] v\;\cos\psi\; c\, dr \tag{227}$$

$$\Delta\, dS = \frac{1}{2}\rho\left[\left(C_{L_\alpha}\sin\phi - c_\ell\,\cos\phi\right)V_d \;+\; 2c_\ell\,\sin\phi\,\Omega r\right] v\;\cos\psi\; c\, dr \tag{228}$$

When ΔS and ΔT are appropriately averaged over the azimuthal angle ψ, and integrated over the radius of B blades, the result is a sideforce F_v and moment M_v due to a unit v perturbation.

$$\Delta\, dF_v = \frac{\Delta\, dS}{v}\;\cos\psi = \frac{1}{2}\rho\left[\left(C_{L_\alpha}\sin\phi - c_\ell\,\cos\phi\right)V_d \;+\; 2c_\ell\,\sin\phi\,\Omega r\right]\;\cos^2\psi\; c\, dr \tag{229}$$

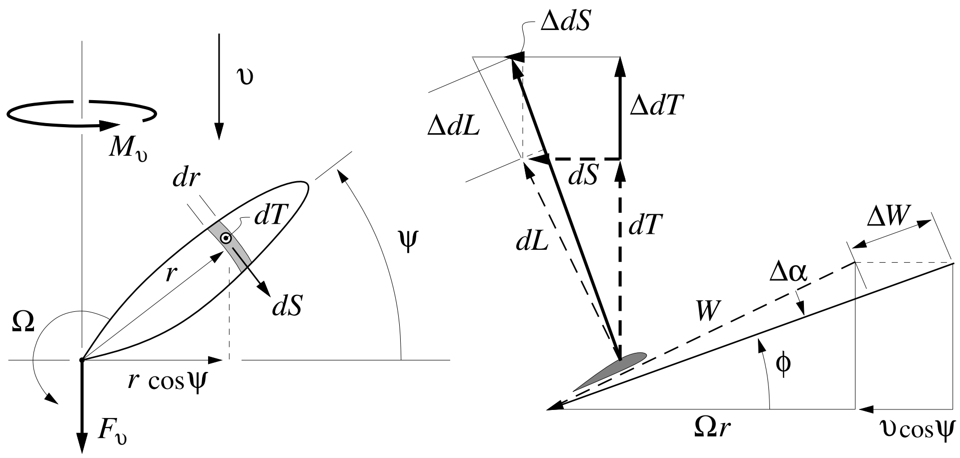

Figure 9: Effect of side velocity v on propeller blade element.

$$dF_v = \frac{1}{2\pi}\int_0^{2\pi} \Delta\, dF_v\, d\psi = \frac{1}{4}\rho\left[(C_{L_\alpha}\sin\phi - c_\ell\,\cos\phi)\, V_d + 2c_\ell\,\sin\phi\,\Omega r\right]\, c\, dr \tag{230}$$

$$F_v = B\int_0^R \frac{dF_v}{dr}\, dr = \frac{1}{4}\rho R^2\left[(C_{L_\alpha} S_0 - c_\ell\, C_0)\, V_d + 2c_\ell\, S_1\,\Omega R\right] \tag{231}$$

$$\Delta\, dM_v = \frac{-\Delta\, dT}{v}\, r\,\cos\psi = -\frac{1}{2}\rho\left[(C_{L_\alpha}\cos\phi + c_\ell\,\sin\phi)\, V_d + 2c_\ell\,\cos\phi\,\Omega r\right]\,\cos^2\psi\, c\, r\, dr \tag{232}$$

$$dM_v = \frac{1}{2\pi}\int_0^{2\pi} \Delta\, dM_v\, d\psi = -\frac{1}{4}\rho\left[(C_{L_\alpha}\cos\phi + c_\ell\,\sin\phi)\, V_d + 2c_\ell\,\cos\phi\,\Omega r\right]\, c\, r\, dr \tag{233}$$

$$M_v = B\int_0^R \frac{dM_v}{dr}\, dr = -\frac{1}{4}\rho R^3\left[(C_{L_\alpha} C_1 + c_\ell\, S_1)\, V_d + 2c_\ell\, C_2\,\Omega R\right] \tag{234}$$

The radial integrals S_0, C_0, S_1 … will be defined at the end of the next section.

14.6 Propeller blade element forces due to pitch rate

Figure 10 shows the propeller blade element subjected to a pitch rate ω, which adds an apparent axial velocity $V_d \rightarrow V_d - \omega r\,\cos\psi$ seen by the blade element. This produces corresponding changes in the relative flow angle and velocity.

$$W^2\Delta\phi = W^2\frac{\partial\phi}{\partial V_d}(-\omega r\,\cos\psi) = -\omega r^2\,\Omega\,\cos\psi \tag{235}$$

$$W\,\Delta W = W\frac{\partial W}{\partial V_d}(-\omega r\,\cos\psi) = -\omega r\, V_d\,\cos\psi \tag{236}$$

The axial and tangential velocity changes are then obtained from (223), (224).

$$\Delta\, dT = \frac{1}{2}\rho\left[(C_{L_\alpha}\cos\phi + c_\ell\,\sin\phi)\,\Omega r - 2c_\ell\,\cos\phi\, V_d\right]\omega r\,\cos\psi\, c\, dr \tag{237}$$

$$\Delta\, dS = \frac{1}{2}\rho\left[(C_{L_\alpha}\sin\phi - c_\ell\,\cos\phi)\,\Omega r - 2c_\ell\,\sin\phi\, V_d\right]\omega r\,\cos\psi\, c\, dr \tag{238}$$

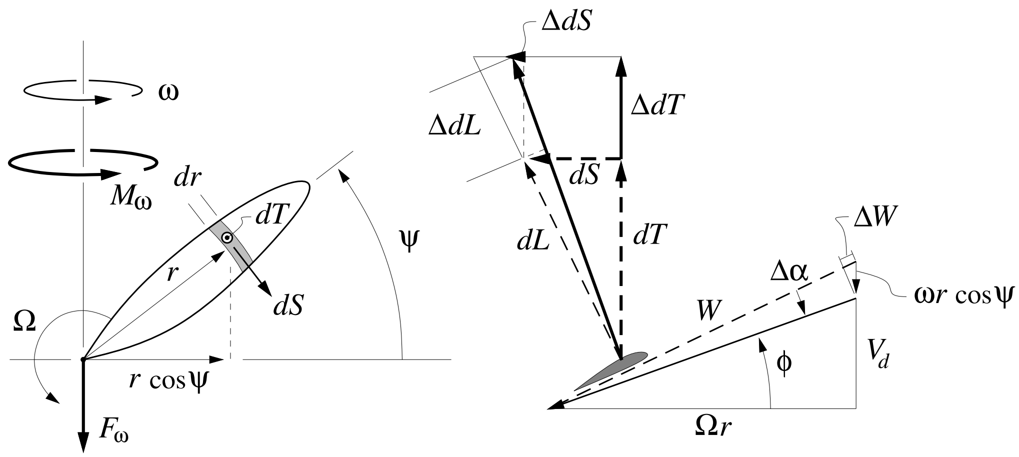

Figure 10: Effect of pitch rate ω on propeller blade element.

Integrating over the azimuth and radius as before, gives a sideforce F_ω and moment M_ω due to a unit ω perturbation.

$$\Delta\, dF_\omega \;=\; \frac{\Delta\, dS}{\omega}\cos\psi \;=\; \frac{1}{2}\rho\left[\left(C_{L_\alpha}\sin\phi - c_\ell\cos\phi\right)\Omega r \;-\; 2c_\ell\sin\phi\, V_d\right] r\,\cos^2\psi\; c\, dr \tag{239}$$

$$dF_\omega \;=\; \frac{1}{2\pi}\int_0^{2\pi}\Delta\, dF_\omega\, d\psi \;=\; \frac{1}{4}\rho\left[\left(C_{L_\alpha}\sin\phi - c_\ell\cos\phi\right)\Omega r \;-\; 2c_\ell\sin\phi\, V_d\right] r\, c\, dr \tag{240}$$

$$F_\omega \;=\; B\int_0^R \frac{dF_\omega}{dr}\, dr \;=\; \frac{1}{4}\rho R^3\left[\left(C_{L_\alpha} S_2 - c_\ell C_2\right)\Omega R \;-\; 2c_\ell S_1 V_d\right] \tag{241}$$

$$\Delta\, dM_\omega \;=\; \frac{-\Delta\, dT}{\omega}\, r\cos\psi \;=\; -\frac{1}{2}\rho\left[\left(C_{L_\alpha}\cos\phi + c_\ell\sin\phi\right)\Omega r \;-\; 2c_\ell\cos\phi\, V_d\right] r^2\,\cos^2\psi\; c\, dr \tag{242}$$

$$dM_\omega \;=\; \frac{1}{2\pi}\int_0^{2\pi}\Delta\, dM_\omega\, d\psi \;=\; -\frac{1}{4}\rho\left[\left(C_{L_\alpha}\cos\phi + c_\ell\sin\phi\right)\Omega r \;-\; 2c_\ell\cos\phi\, V_d\right] r^2\, c\, dr \tag{243}$$

$$M_\omega \;=\; B\int_0^R \frac{dM_\omega}{dr}\, dr \;=\; -\frac{1}{4}\rho R^4\left[\left(C_{L_\alpha} C_3 + c_\ell S_3\right)\Omega R \;-\; 2c_\ell C_2 V_d\right] \tag{244}$$

14.7 Propeller blade radial integrals

The radial integrals appearing in the force and moment perturbation expressions are defined as follows, along with suitable approximations.

$$S_0 \;=\; B\int_0^1 \frac{c}{R}\sin\phi\; d\!\left(\frac{r}{R}\right) \;\simeq\; 1.20B\left(\frac{c}{R}\sin\phi\right)_{0.75R} \tag{245}$$

$$C_0 \;=\; B\int_0^1 \frac{c}{R}\cos\phi\; d\!\left(\frac{r}{R}\right) \;\simeq\; 0.80B\left(\frac{c}{R}\cos\phi\right)_{0.75R} \tag{246}$$

$$S_1 \;=\; B\int_0^1 \frac{c}{R}\sin\phi\,\left(\frac{r}{R}\right) d\!\left(\frac{r}{R}\right) \;\simeq\; 0.70B\left(\frac{c}{R}\sin\phi\right)_{0.75R} \tag{247}$$

$$C_1 \;=\; B\int_0^1 \frac{c}{R}\cos\phi\,\left(\frac{r}{R}\right) d\!\left(\frac{r}{R}\right) \;\simeq\; 0.45B\left(\frac{c}{R}\cos\phi\right)_{0.75R} \tag{248}$$

$$S_2 \;=\; B\int_0^1 \frac{c}{R}\sin\phi\,\left(\frac{r}{R}\right)^2 d\!\left(\frac{r}{R}\right) \;\simeq\; 0.35B\left(\frac{c}{R}\sin\phi\right)_{0.75R} \tag{249}$$

$$C_2 = B\int_0^1 \frac{c}{R}\cos\phi\left(\frac{r}{R}\right)^2 d\left(\frac{r}{R}\right) \simeq 0.30B\left(\frac{c}{R}\cos\phi\right)_{0.75R} \tag{250}$$

$$S_3 = B\int_0^1 \frac{c}{R}\sin\phi\left(\frac{r}{R}\right)^3 d\left(\frac{r}{R}\right) \simeq 0.20B\left(\frac{c}{R}\sin\phi\right)_{0.75R} \tag{251}$$

$$C_3 = B\int_0^1 \frac{c}{R}\cos\phi\left(\frac{r}{R}\right)^3 d\left(\frac{r}{R}\right) \simeq 0.20B\left(\frac{c}{R}\cos\phi\right)_{0.75R} \tag{252}$$

14.8 Overall engine force and moment

Superposition of the primary and perturbation forces and moments gives the following total engine force and moment, in engine axes. The sign factor s accounts for the direction of the prop rotation rate Ω, defined positive about the x axis (a conventional right-handed prop has $s = -1$).

$$s = \mathrm{sign}(\Omega) \tag{253}$$

$$\vec{F}_e = \left\{ \begin{array}{c} -F_{\mathrm{eng}} \\ F_v v_y + sF_\omega \omega_y \\ F_v v_z + sF_\omega \omega_z \end{array} \right\} \tag{254}$$

$$\vec{M}_e = \left\{ \begin{array}{c} -M_{\mathrm{eng}} \\ sM_v v_y + M_\omega \omega_y \\ sM_v v_z + M_\omega \omega_z \end{array} \right\} \tag{255}$$

Using the inverse engine-axis transformation tensor

$$\vec{F}_{\mathrm{eng}} = \bar{\bar{T}}_e^T \vec{F}_e \tag{256}$$

$$\vec{M}_{\mathrm{eng}} = \bar{\bar{T}}_e^T \vec{M}_e \tag{257}$$

then gives the corresponding body-axis force and moment which are applied as beam loads.

15 Real and Apparent Mass and Inertia

Although the total mass and inertias are not required to compute a solution in ASWING, knowing these quantities is of considerable interest in applications. These are computed as described below. The inertias will be derived from the fundamentals, which will clarify the somewhat unusual form of the apparent inertia. The real mass and inertia quantities will be denoted by $(\,)_{\mathrm{rm}}$, and the apparent mass and inertia quantities will be denoted by $(\,)_{\mathrm{am}}$.

15.1 Radius and inertia tensors

For the purpose of deriving the moment-of-inertia tensors, it's convenient to first define an anti-symmetric tensor $\bar{\bar{r}}$ which is formed from the three components of the corresponding vector $\vec{r}$,

$$\vec{r} = x\hat{\imath} + y\hat{\jmath} + z\hat{k} \tag{258}$$

$$\bar{\bar{r}} \equiv \vec{r} \times \bar{\bar{1}} = \left[\begin{array}{ccc} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{array} \right] \tag{259}$$

where $\bar{\bar{1}}$ is the identity matrix. This allows a cross product of $\vec{r}$ with any vector $\vec{A}$ to be expressed as a matrix-vector product which is simpler to manipulate.

$$\vec{r} \times \vec{A} \;=\; \bar{\bar{r}} \vec{A} \tag{260}$$

$$\vec{A} \times \vec{r} \;=\; -\vec{r} \times \vec{A} \;=\; -\bar{\bar{r}} \vec{A} \tag{261}$$

The negative square of the radius tensor is the familiar unit-mass inertia tensor.

$$-\bar{\bar{r}}\,\bar{\bar{r}} \;=\; \begin{bmatrix} y^2+z^2 & -xy & -xz \\ -yx & x^2+z^2 & -yz \\ -zx & -zy & x^2+y^2 \end{bmatrix} \tag{262}$$

15.2 Integrated Force and Moment Equations

The real and apparent mass derivation begins with the discrete beam force equation, summed over all the beam intervals.

$$\sum_i \left\{ \vec{F}_{i+1} - \vec{F}_i + \vec{f}_a \,\Delta s + \Delta\vec{F} \right\} \;=\; 0 \tag{263}$$

$$\sum_i \vec{f}_a \,\Delta s + \sum_p \Delta\vec{F} \;=\; 0 \tag{264}$$

The second simplified form (264) results from the zero-force boundary conditions $\vec{F}_i = 0$ on all the beam ends in free flight.

The discrete beam moment equation is also summed over all beam intervals.

$$\sum_i \left\{ \vec{M}_{i+1} - \vec{M}_i + \vec{m}_a \,\Delta s + \Delta\vec{M} + \Delta\vec{r} \times \vec{F}_a \right\} \;=\; 0 \tag{265}$$

The last force term is first expanded as

$$\begin{aligned}
\Delta\vec{r} \times \vec{F}_a &= \tfrac{1}{2} \left(\bar{\bar{r}}_{i+1} - \bar{\bar{r}}_i\right) \left(\vec{F}_{i+1} + \vec{F}_i\right) \\
&= \bar{\bar{r}}_{i+1}\vec{F}_{i+1} - \bar{\bar{r}}_i\vec{F}_i - \tfrac{1}{2} \left(\bar{\bar{r}}_{i+1} + \bar{\bar{r}}_i\right) \left(\vec{F}_{i+1} - \vec{F}_i\right) \\
&= \bar{\bar{r}}_{i+1}\vec{F}_{i+1} - \bar{\bar{r}}_i\vec{F}_i + \bar{\bar{r}}_a \left(\vec{f}_a \,\Delta s + \Delta\vec{F}\right)
\end{aligned} \tag{266}$$

which gives the following summed moment equation.

$$\sum_i \left(\vec{m} + \bar{\bar{r}}\,\vec{f}\right)_a \Delta s + \sum_p \left(\Delta\vec{M} + \bar{\bar{r}}\,\Delta\vec{F}\right) \;=\; 0 \tag{267}$$

As before, beam-end boundary conditions $\vec{M}_i = 0$, $\vec{F}_i = 0$, have been applied to cancel the summed difference terms.

15.3 Applied force

The applied force term $\vec{f}_a$ appearing in (264) and (267) has the following contributions, treated earlier. Drag contributions are neglected here.

$$\vec{f} \;=\; \vec{f}_S + \vec{f}_U + \vec{f}_{\text{am}} + \vec{f}_{\text{acc}} \tag{268}$$

$$
\begin{aligned}
\vec{f}_S &= \rho\, \Gamma\, \vec{V} \times \hat{s} && (269)\\
\vec{f}_U &= \rho\, \frac{\partial \Gamma}{\partial t}\, \frac{\bar{c}}{|\vec{V}_\perp|}\, \vec{V} \times \hat{s} && (270)\\
\vec{f}_{\mathrm{am}} &= \frac{\pi}{4}\, \rho\, \bar{c}^2 \left[\vec{V} \times \left(\vec{\Omega} + \vec{\omega}_i \right) \cdot \hat{n} \;-\; \vec{a}_{c/2} \cdot \hat{n} \right] \hat{n} && (271)\\
\vec{f}_{\mathrm{acc}} &= \mu\, (\vec{g} - \vec{a}_{cg}) && (272)
\end{aligned}
$$

$$
\begin{aligned}
\vec{m} &= \vec{m}_S \;+\; \vec{m}_U \;+\; \vec{m}_{\mathrm{am}} \;+\; \vec{m}_{\mathrm{acc}} && (273)\\
\vec{m}_S &= \Delta \vec{r}_{c/4} \times \vec{f}_S \;+\; \frac{1}{2} \rho\, |\vec{V}_\perp|^2\, \bar{c}^2\, c_m\, \hat{s} && (274)\\
\vec{m}_U &= \Delta \vec{r}_{c/4} \times \vec{f}_U && (275)\\
\vec{m}_{\mathrm{am}} &= -\frac{\pi}{4}\, \rho\, \bar{c}^2\, \frac{\bar{c}}{4} \left[\vec{V} \times \left(\vec{\Omega} + \vec{\omega}_i \right) \cdot \hat{n} \;+\; \frac{\bar{c}}{8} \left(\vec{\alpha}_o + \dot{\vec{\omega}}_i \right) \cdot \hat{s} \right] \hat{s} \;+\; \Delta \vec{r}_{c/2} \times \vec{f}_{\mathrm{am}} && (276)\\
\vec{m}_{\mathrm{acc}} &= \Delta \vec{r}_{cg} \times \vec{f}_{\mathrm{acc}} \;-\; \bar{\bar{T}}^T\, \bar{\bar{\imath}}\, \bar{\bar{T}} \left(\vec{\alpha}_o + \dot{\vec{\omega}}_i \right) \;-\; \left(\vec{\Omega} + \vec{\omega}_i \right) \times \left\{ \bar{\bar{T}}^T\, \bar{\bar{\imath}}\, \bar{\bar{T}} \left(\vec{\Omega} + \vec{\omega}_i \right) \right\} && (277)
\end{aligned}
$$

The moment sum terms in (267) can be combined as follows.

$$
\begin{aligned}
\vec{m}_S \;+\; \bar{\bar{r}}\, \vec{f}_S &= \bar{\bar{r}}_{c/4}\, \vec{f}_S \;+\; \frac{1}{2} \rho\, |\vec{V}_\perp|^2\, \bar{c}^2\, c_m\, \hat{s} && (278)\\
\vec{m}_U \;+\; \bar{\bar{r}}\, \vec{f}_U &= \bar{\bar{r}}_{c/4}\, \vec{f}_U && (279)\\
\vec{m}_{\mathrm{am}} \;+\; \bar{\bar{r}}\, \vec{f}_{\mathrm{am}} &= \frac{\pi}{4}\, \rho\, \bar{c}^2 \left\{ -\frac{\bar{c}}{4} \ldots \right\} && (280)\\
\vec{m}_{\mathrm{acc}} \;+\; \bar{\bar{r}}\, \vec{f}_{\mathrm{acc}} &= \mu\, \bar{\bar{r}}_{cg} (\vec{g} - \vec{a}_{cg}) \;-\; \bar{\bar{I}} \left(\vec{\alpha}_o + \dot{\vec{\omega}}_i \right) \;-\; \left(\vec{\Omega} + \vec{\omega}_i \right) \times \left\{ \bar{\bar{I}} \left(\vec{\Omega} + \vec{\omega}_i \right) \right\} && (281)
\end{aligned}
$$

References

[1] B. Etkin. *Dynamics of Flight — Aircraft Stability and Control.* John Wiley and Sons, New York, 1972.

[2] T. Theodorsen. General theory of aerodynamic instability and the mechanism of flutter. TR 490, NACA, 1935.

[3] R.B. Lehoucq, D.C. Sorensen, and C. Yang. ARPACK users's guide. Report 1, Computational and Applied Mathematics, Rice University, Houston, TX, 1997.