UNIVERSITY OF Southampton
School of Engineering Sciences

Aerodynamics & Flight Mechanics Research Group

Ground Resonance Modelling using Appell's Equations

S. J. Newman

Technical Report AFM-11/21

January 2011

UNIVERSITY OF SOUTHAMPTON

SCHOOL OF ENGINEERING SCIENCES

AERODYNAMICS AND FLIGHT MECHANICS RESEARCH GROUP

Ground Resonance Modelling using Appell's Equations

by

S. J. Newman

AFM Report No. AFM 11/21

January 2011

© School of Engineering Sciences, Aerodynamics and Flight Mechanics Research Group

COPYRIGHT NOTICE

(c) SES University of Southampton All rights reserved.

SES authorises you to view and download this document for your personal, non-commercial use. This authorization is not a transfer of title in the document and copies of the document and is subject to the following restrictions: 1) you must retain, on all copies of the document downloaded, all copyright and other proprietary notices contained in the Materials; 2) you may not modify the document in any way or reproduce or publicly display, perform, or distribute or otherwise use it for any public or commercial purpose; and 3) you must not transfer the document to any other person unless you give them notice of, and they agree to accept, the obligations arising under these terms and conditions of use. This document, is protected by worldwide copyright laws and treaty provisions.

Preamble

The following analysis derives the equations of motion for an N bladed rotor with the rotor hub fixed to earth by a linear spring and damper in parallel (Figure 1). The rotor blades are modeled by light rods with a concentrated mass at the outer extremity (Figure 2). The blades are attached to the hub via an offset lag hinge to which is fitted linear rotating spring and damper in parallel.

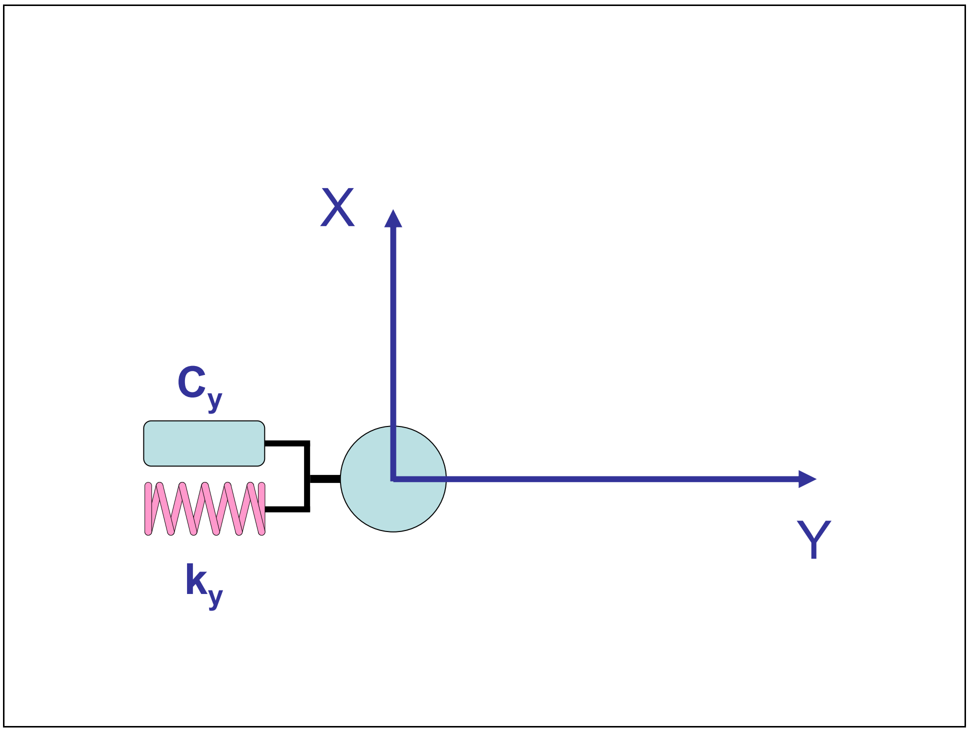

Figure 1 – Rotor Hub Motion Coordinates & Constraints

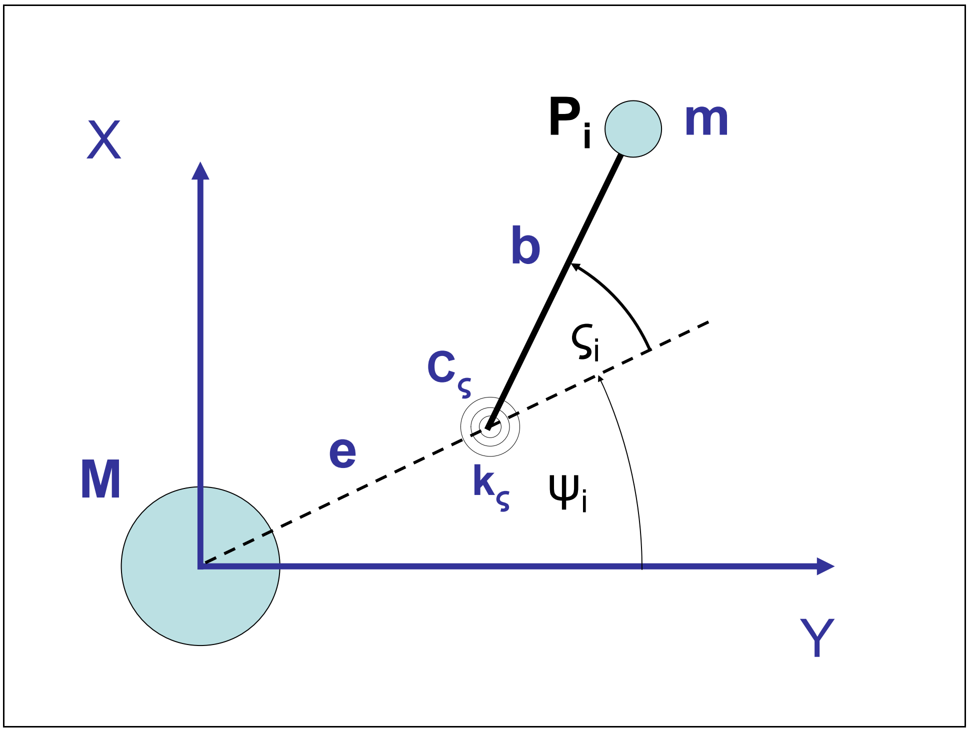

Figure 2 – Blade Lag Motion Coordinates & Constraints

The analysis is linearised from a derivation using the Gibbs-Appell equations.

Nomenclature

X	Coordinate Normal to Spring
Y	Coordinate in Line with Spring
ψ_i	Azimuth Angle of i^{th} Blade
ζ_i	Lag Angle of i^{th} Blade
y	Rotor Head Translation Variable
k_ζ	Lag Spring Rate
C_ζ	Lag Damper Rate
$\overline{C}_\zeta$	Normalised Lag Damper Rate
k_y	Head Translation Spring Rate
C_y	Head Translation Damper Rate
$\overline{C}_y$	Normalised Head Translation Damper Rate
P_i	Tip Mass of i^{th} Blade
M	Effective Head Mass
m	Tip Mass
e	Lag Hinge Offset
b	Blade Length
X_i	Coordinates of P_i
Y_i	
R	Rotor Radius
Ω	Rotor Speed
θ, ϕ	Angles used in Free Lag Motion Analysis
R’	Dimension used in Free Lag Motion Analysis
p, q, r, s	Coleman Variables
N	Number of Blades
ω_ζ	Lag Frequency
ω_y	Head Translational Frequency
Ω_L, α, β, λ	Simplifying Variables

Analysis

Position of P_i is given by:

$$X_i = (e + b\cos\zeta_i)\sin\psi_i + b\sin\zeta_i\cos\psi_i \tag{1a}$$

$$Y_i = y + (e + b\cos\zeta_i)\cos\psi_i - b\sin\zeta_i\sin\psi_i \tag{1b}$$

Defining:

$$R = e + b$$

$$\begin{aligned} \zeta_i &\ll 1 \\ \cos\zeta_i &\approx 1 - \frac{1}{2}\zeta_i^{\,2} \\ \sin\zeta_i &\approx \zeta_i \end{aligned}$$

$$X_i = \left(R - \frac{1}{2}b\zeta_i^2\right)\sin\psi_i + b\zeta_i\cos\psi_i \tag{2a}$$

$$Y_i = y + \left(R - \frac{1}{2}b\zeta_i^2\right)\cos\psi_i - b\zeta_i\sin\psi_i \tag{2b}$$

Differentiating with respect to time gives:

$$\begin{aligned} \dot{X}_i = &-b\zeta_i\dot{\zeta}_i\sin\psi_i + \Omega\left(R - \frac{1}{2}b\zeta_i^2\right)\cos\psi_i \\ &+ b\dot{\zeta}_i\cos\psi_i - \Omega b\zeta_i\sin\psi_i \end{aligned} \tag{3a}$$

$$\ddot{X}_i = -b\left(\zeta_i\ddot{\zeta}_i + \dot{\zeta}_i^{\,2}\right)\sin\psi_i - 2b\zeta_i\dot{\zeta}_i\Omega\cos\psi_i - \Omega^2\left(R - \frac{1}{2}b\zeta_i^2\right)\sin\psi_i$$
$$+ b\left(\ddot{\zeta}_i\cos\psi_i - 2\Omega\dot{\zeta}_i\sin\psi_i - \Omega^2\zeta_i\cos\psi_i\right) \tag{3a}$$

$$\ddot{Y}_i = \ddot{y} - b\left(\zeta_i\ddot{\zeta}_i + \dot{\zeta}_i^{\,2}\right)\cos\psi_i + 2b\zeta_i\dot{\zeta}_i\Omega\sin\psi_i - \Omega^2\left(R - \frac{1}{2}b\zeta_i^2\right)\cos\psi_i$$
$$- b\left(\ddot{\zeta}_i\sin\psi_i + 2\Omega\dot{\zeta}_i\cos\psi_i - \Omega^2\zeta_i\sin\psi_i\right) \tag{3b}$$

Noting that:

$$\Omega = \dot{\psi}_i$$

Now:

$$\frac{\partial X_i}{\partial \zeta_k} = \left\{-b\zeta_k\, sin\psi_k + b\, cos\psi_k\right\}\cdot\delta_{ik}$$
$$\frac{\partial X_i}{\partial y} = 0$$
$$\frac{\partial Y_i}{\partial \zeta_k} = \left\{-b\zeta_k\, cos\psi_k - b\, sin\psi_k\right\}\cdot\delta_{ik}$$
$$\frac{\partial Y_i}{\partial y} = 1$$

We now form the Appel Function A:

$$\mathrm{A} = \sum_i \frac{1}{2}m\left(\ddot{X}_i^2 + \ddot{Y}_i^2\right) + \frac{1}{2}M\ddot{y}^2$$

It can be proved that:

$$\frac{\partial \ddot{X}_i}{\partial \ddot{q}} = \frac{\partial X_i}{\partial q}$$
$$\frac{\partial \ddot{Y}_i}{\partial \ddot{q}} = \frac{\partial Y_i}{\partial q}$$

We have:

$$\frac{\partial \mathrm{A}}{\partial \zeta_k} = m\sum_i \left\{ \ddot{X}_i \frac{\partial X_i}{\partial \zeta_k} + \ddot{Y}_i \frac{\partial Y_i}{\partial \zeta_k} \right\}$$

$$\frac{\partial \mathrm{A}}{\partial y} = m\sum_i \left\{ \ddot{X}_i \frac{\partial X_i}{\partial y} + \ddot{Y}_i \frac{\partial Y_i}{\partial y} \right\} + M\ddot{y}$$

Linearising these expressions:

$$\ddot{X}_i = -\Omega^2 R \sin\psi_i + b\left(\ddot{\zeta}_i \cos\psi_i - 2\Omega\dot{\zeta}_i \sin\psi_i - \Omega^2 \zeta_i \cos\psi_i\right)$$

$$\ddot{Y}_i = \ddot{y} - \Omega^2 R \cos\psi_i - b\left(\ddot{\zeta}_i \sin\psi_i + 2\Omega\dot{\zeta}_i \cos\psi_i - \Omega^2 \zeta_i \sin\psi_i\right)$$

$$\frac{\partial X_i}{\partial \zeta_k} = b\cdot\{-\zeta_k \, sin\psi_k + cos\psi_k\}\cdot\delta_{ik}$$

$$\frac{\partial X_i}{\partial y} = 0$$

$$\frac{\partial Y_i}{\partial \zeta_k} = -b\cdot\{\zeta_k \, cos\psi_k + sin\psi_k\}\cdot\delta_{ik}$$

$$\frac{\partial Y_i}{\partial y} = 1$$

Whence

$$\frac{\partial \mathrm{A}}{\partial \zeta_k} = m\left[\begin{array}{l} \left\{-\left(\Omega^2 R + 2\Omega b\dot{\zeta}_k\right)\sin\psi_k + b\left(\ddot{\zeta}_k - \Omega^2\zeta_k\right)\cos\psi_k\right\}\bullet\left\{b\left(\cos\psi_k - \zeta_k \sin\psi_k\right)\right\} \\ + \left\{\ddot{y} - \left(\Omega^2 R + 2\Omega b\dot{\zeta}_k\right)\cos\psi_k - b\left(\ddot{\zeta}_k - \Omega^2\zeta_k\right)\sin\psi_k\right\}\bullet\left\{-b\left(\sin\psi_k + \zeta_k \cos\psi_k\right)\right\} \end{array}\right.$$

$$\frac{\partial \mathrm{A}}{\partial y} = m\sum_i \left\{\ddot{y} - \left(\Omega^2 R + 2\Omega b\dot{\zeta}_i\right) cos\psi_i - b\left(\ddot{\zeta}_i - \Omega^2\zeta_i\right) sin\psi_i\right\} + M\,\ddot{y}$$

$$V = \frac{1}{2}k_y y^2 + \sum_i \frac{1}{2}k_\zeta \zeta_i^2$$

Whence:

$$\frac{\partial V}{\partial \zeta_k} = k_\zeta \zeta_k$$

$$\frac{\partial V}{\partial y} = k_y y$$

The Appell equations are thus:

$$\frac{\partial \mathrm{A}}{\partial \ddot{\zeta}_k} = Q_{\zeta_k} = -\frac{\partial V}{\partial \zeta_k} - C_\zeta \dot{\zeta}_k$$

$$\frac{\partial \mathrm{A}}{\partial y} = Q_y = -\frac{\partial V}{\partial y} - C_y \dot{y}$$

∀k

$$m\left[\begin{array}{l}\left\{-\left(\Omega^2 R + 2\Omega b \dot{\zeta}_k\right)\sin\psi_k + b\left(\ddot{\zeta}_k - \Omega^2 \zeta_k\right)\cos\psi_k\right\} \bullet \left\{b\left(\cos\psi_k - \zeta_k \sin\psi_k\right)\right\} \\ + \left\{\ddot{y} - \left(\Omega^2 R + 2\Omega b \dot{\zeta}_k\right)\cos\psi_k - b\left(\ddot{\zeta}_k - \Omega^2 \zeta_k\right)\sin\psi_k\right\} \bullet \left\{-b\left(\sin\psi_k + \zeta_k \cos\psi_k\right)\right\}\end{array}\right]$$
$$= -k_\zeta \zeta_k - C_\zeta \dot{\zeta}_k$$

&

$$m\sum_i \left\{\ddot{y} - \left(\Omega^2 R + 2\Omega b \dot{\zeta}_i\right) cos\psi_i - b\left(\ddot{\zeta}_i - \Omega^2 \zeta_i\right) sin\psi_i\right\} + M\,\ddot{y} = -k_y y - C_y \dot{y}$$

Expanding to first order:

∀k

$$
mb\begin{bmatrix}\left\{-\left(\Omega^2 R + 2\Omega b\dot{\zeta}_k\right) sin\,\psi_k\, cos\,\psi_k + b\left(\ddot{\zeta}_k - \Omega^2\zeta_k\right) cos^2\psi_k + \Omega^2 R\zeta_k\, sin^2\psi_k - \ddot{y}\, sin\,\psi_k\right\} \\ +\Omega^2 R\, cos\,\psi_k\left(sin\,\psi_k + \zeta_k\, cos\,\psi_k\right) + 2\Omega b\dot{\zeta}_k\, sin\,\psi_k\, cos\,\psi_k + b\left(\ddot{\zeta}_k - \Omega^2\zeta_k\right) sin^2\psi_k\end{bmatrix}
$$
$$
= -k_\zeta\zeta_k - C_\zeta\dot{\zeta}_k
$$

Whence:

$$
b\ddot{\zeta}_k + \zeta_k\left\{-\Omega^2 b + \Omega^2 R\right\} - \ddot{y}\sin\psi_k = -\frac{k_\zeta}{mb}\zeta_k - \frac{C_\zeta}{mb}\dot{\zeta}_k
$$

&

$$
m\sum_i\left\{-b\ddot{\zeta}_i\sin\psi_i - 2\Omega b\dot{\zeta}_i\cos\psi_i + \Omega^2 b\zeta_i\sin\psi_i - \Omega^2 R\cos\psi_i\right\} + \left(M + Nm\right)\ddot{y}
$$
$$
= -k_y y - C_y\dot{y}
$$

Noting that:

$$
\sum_i \cos\psi_i = 0
$$
$$
e = R - b
$$

&

$$\frac{\partial^2}{\partial t^2}(\zeta_i \sin\psi_i) = \ddot{\zeta}_i \sin\psi_i + 2\Omega\dot{\zeta}_i \cos\psi_i - \Omega^2\zeta_i \sin\psi_i$$

$$\frac{\partial^2}{\partial t^2}(\zeta_i \cos\psi_i) = \ddot{\zeta}_i \cos\psi_i - 2\Omega\dot{\zeta}_i \sin\psi_i - \Omega^2\zeta_i \cos\psi_i$$

Equations of motion:

ζ k (∀k)

$$b\ddot{\zeta}_k + \Omega^2 e\zeta_k - \ddot{y}\sin\psi_k = -\frac{k_\zeta}{mb}\zeta_k - \frac{C_\zeta}{mb}\dot{\zeta}_k$$

y

$$-mb\sum_i \frac{\partial^2}{\partial t^2}(\zeta_i \sin\psi_i) + (M + Nm)\ddot{y} = -k_y y - C_y \dot{y}$$

Free Lag Motion

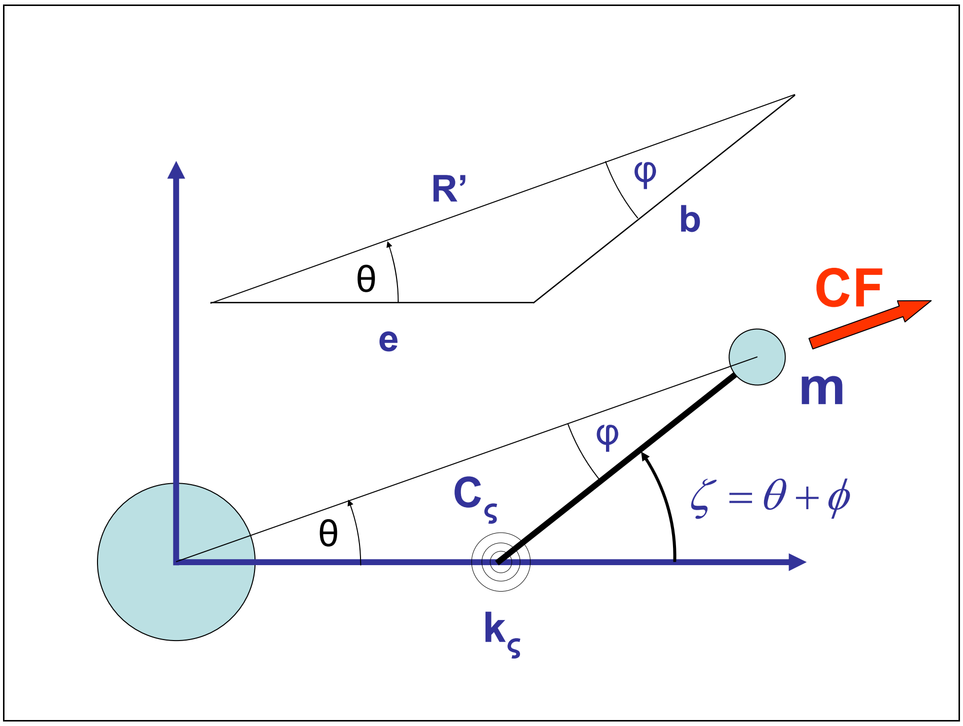

Figure 3 - Free Lagging Geometry

Consider now the free lagging motion of the blade with the rotor hub fixed:

$$C.F. = m\Omega^2 R'$$

$$(C.F.) \cdot b \cdot \sin\phi + k_\zeta \zeta$$

The restoring moment of the Centrifugal Force about the lag hinge is given by:

By the Sine Rule:

$$\frac{R'}{\sin\zeta} = \frac{e}{\sin\phi}$$

From which we obtain:

$$\sin\phi = \frac{e}{R'}\sin\zeta$$

Whence the restoring moment becomes:

$$m\Omega^2 R' \cdot b \cdot \frac{e}{R'}\sin\zeta + k_\zeta \zeta$$

The lag inertia of the blade is:

$$mb^2$$

The lag equation of motion can now be assembled (assuming that ς is small):

$$mb^2\ddot{\zeta} = -\left\{m\Omega^2 be\zeta + k_\zeta \zeta\right\}$$

From which:

$$\ddot{\zeta} = -\left\{\Omega^2 \frac{e}{b} + \frac{k_\zeta}{mb^2}\right\}\cdot\zeta$$

Which is Simple Harmonic Motion of circular frequency:

$$\omega_L^2 = \Omega^2 \frac{e}{b} + \frac{k_\zeta}{mb^2}$$

If the rotor is stationary, the lag frequency is given by:

$$\omega_{L0}^2 = \frac{k_\zeta}{mb^2}$$

From which we find:

$$\omega_L^2 = \Omega^2 \frac{e}{b} + \omega_{L0}^2$$

Free Hub/Blades Translational Motion

If the blades are now fixed in lag and the rotor allowed to oscillate on the spring k_y the equation of motion for y is given by:

$$(M + Nm)\ddot{y} = -k_y y$$

Which is Simple Harmonic Motion of circular frequency:

$$\omega_y^2 = \frac{k_y}{(M + Nm)}$$

Substituting these results into the full equations of motion we find:

$$\zeta \; k \;\; (\forall k)$$

$$\ddot{\zeta}_k + 2\omega_L \overline{C}_\zeta \dot{\zeta}_k + \zeta_k\left(\omega_L^2\right) - \frac{\ddot{y}}{b} sin\psi_k = 0$$

y

$$\ddot{y} + 2\omega_y \overline{C}_y \dot{y} + \omega_y^2 y - \frac{mb}{M + Nm} \sum_i \frac{\partial^2}{\partial t^2}\left(\zeta_i \, sin\psi_i\right) = 0$$

Where the damping terms have been replaced by the damping ratio to critical, i.e.

$$C_\zeta = 2\omega_L \cdot mb^2 \cdot \overline{C}_\zeta$$
$$C_y = 2\omega_y \cdot (M + Nm) \cdot \overline{C}_y$$

Coleman Transformation

The Coleman variables are defined by:

$$p = \sum_i \zeta_i \cdot sin\psi_i$$
$$q = \sum_i \zeta_i \cdot cos\psi_i$$
$$r = \sum_i \zeta_i$$
$$s = \sum_i (-1)^{i+1} \zeta_i$$

The s variable only applies if the rotor has an even number of blades.

Differentiating these expressions gives the following:

$$\sum_i \dot{\zeta}_i \cdot sin\psi_i = \dot{p} - \Omega q$$
$$\sum_i \dot{\zeta}_i \cdot cos\psi_i = \dot{q} + \Omega p$$
$$\sum_i \ddot{\zeta}_i \cdot sin\psi_i = \ddot{p} - 2\Omega\dot{q} - \Omega^2 p$$
$$\sum_i \ddot{\zeta}_i \cdot cos\psi_i = \ddot{q} + 2\Omega\dot{p} - \Omega^2 q$$

Multiplying the ζ_k equations by $\sin\psi_k$ and summing over k gives *(the summing variable k is replaced by i)*;

$$\sum_i \ddot{\zeta}_i \, sin\psi_i + 2\omega_L \overline{C}_\zeta \sum_i \dot{\zeta}_i \, sin\psi_i + \omega_L^2 \sum_i \zeta_i \, sin\psi_i - \frac{\ddot{y}}{b} \sum_i sin^2\psi_i = 0$$

Similarly multiplying by $\cos\psi_k$ and summing over k gives:

$$\sum_i \ddot{\zeta}_i \, cos\psi_i + 2\omega_L \overline{C}_\zeta \sum_i \dot{\zeta}_i \, cos\psi_i + \omega_L^2 \sum_i \zeta_i \, cos\psi_i - \frac{\ddot{y}}{b} \sum_i sin\psi_i \cdot cos\psi_i = 0$$

As the blades are equispaced around the azimuth we have:

$$\sum_i sin^2\psi_i = \frac{N}{2}$$
$$\sum_i sin\psi_i \cdot cos\psi_i = 0$$

Whence we find:

$$\ddot{p} + 2\omega_L \overline{C}_\zeta \dot{p} - 2\Omega\dot{q} + p\left(\omega_L{}^2 - \Omega^2\right) - 2\omega_L \overline{C}_\zeta \Omega q - \frac{N}{2b}\ddot{y} = 0$$
$$\ddot{q} + 2\omega_L \overline{C}_\zeta \dot{q} + 2\Omega\dot{p} + q\left(\omega_L{}^2 - \Omega^2\right) + 2\omega_L \overline{C}_\zeta \Omega p = 0$$

Also the y equation becomes:

$$\ddot{y} - \left(\frac{mb}{M + Nm}\right)\ddot{p} + 2\omega_y \overline{C}_y \cdot \dot{y} + \omega_y^2 \cdot y = 0$$

The equation for p, q and y can now be expressed in matrix form:

$$\begin{bmatrix} 1 & 0 & -\frac{N}{2b} \\ 0 & 1 & 0 \\ -\lambda & 0 & 1 \end{bmatrix}\begin{bmatrix} \ddot{p} \\ \ddot{q} \\ \ddot{y} \end{bmatrix} + \begin{bmatrix} 2\omega_L \overline{C}_\zeta & -2\Omega & 0 \\ 2\Omega & 2\omega_L \overline{C}_\zeta & 0 \\ 0 & 0 & 2\omega_y \overline{C}_y \end{bmatrix}\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{y} \end{bmatrix} + \begin{bmatrix} \Omega_L^2 & -2\omega_L \overline{C}_\zeta \Omega & 0 \\ 2\omega_L \overline{C}_\zeta \Omega & \Omega_L^2 & 0 \\ 0 & 0 & \omega_y^2 \end{bmatrix}\begin{bmatrix} p \\ q \\ y \end{bmatrix}$$

Where:

$$\Omega_L^2 = \omega_L^2 - \Omega^2$$
$$\lambda = \frac{mb}{M + Nm}$$

If we assume SHM then the solution can be expressed as:

$$p = Pe^{\omega t}$$
$$q = Qe^{\omega t}$$
$$y = Ye^{\omega t}$$

Putting:

$$\alpha = 2\omega_L \overline{C}_\zeta$$
$$\beta = 2\omega_y \overline{C}_y$$

We find:

$$\left\{ \omega^2 \cdot \begin{bmatrix} 1 & 0 & -\dfrac{N}{2b} \\ 0 & 1 & 0 \\ -\lambda & 0 & 1 \end{bmatrix} + \omega \cdot \begin{bmatrix} \alpha & -2\Omega & 0 \\ 2\Omega & \alpha & 0 \\ 0 & 0 & \beta \end{bmatrix} \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{y} \end{bmatrix} + \begin{bmatrix} \Omega_L^2 & -\Omega\alpha & 0 \\ \Omega\alpha & \Omega_L^2 & 0 \\ 0 & 0 & \omega_y^2 \end{bmatrix} \right\} \begin{bmatrix} P \\ Q \\ Y \end{bmatrix} = 0$$

Which will have a solution only if:

$$\begin{vmatrix} \omega^2 + \omega\alpha + \Omega_L^2 & -\Omega(2\omega + \alpha) & -\dfrac{N\omega^2}{2b} \\ \Omega(2\omega + \alpha) & \omega^2 + \omega\alpha + \Omega_L^2 & 0 \\ -\lambda\omega^2 & 0 & \omega^2 + \omega\beta + \omega_y^2 \end{vmatrix} = 0$$

Putting:

$$\mu = \frac{N\lambda}{2b}$$
$$= \frac{NM}{2(M + Nm)}$$

We have for the frequency ω:

$$a_1\omega^6 + a_2\omega^5 + a_3\omega^4 + a_4\omega^3 + a_5\omega^2 + a_6\omega^1 + a_7 = 0$$

Where the coefficients are given by:

$$a_1 = 1 - \mu$$
$$a_2 = 2\alpha + \beta - \mu\alpha$$
$$a_3 = \alpha^2 + 2\Omega_L^2 + 2\alpha\beta + \omega_y^2 + 4\Omega^2 - \mu\Omega_L^2$$
$$a_4 = 2\alpha\Omega_L^2 + \beta\left(\alpha^2 + 2\Omega_L^2\right) + 2\alpha\omega_y^2 + 4(\alpha + \beta)\Omega^2$$
$$a_5 = \Omega_L^4 + 2\alpha\beta\Omega_L^2 + \left(\alpha^2 + 2\Omega_L^2\right)\omega_y^2 + \left(\alpha^2 + 4\alpha\beta + 4\omega_y^2\right)\Omega^2$$
$$a_6 = \Omega_L^4\beta + 2\alpha\Omega_L^2\omega_y^2 + \left(\alpha^2\beta + 4\alpha\omega_y^2\right)\Omega^2$$
$$a_7 = \omega_y^2\left(\Omega_L^4 + \alpha^2\Omega^2\right)$$

MATLAB Files – Suite of Programs

Program Summary

M File Name	Function
GRSET	*Set up the basic default rotor and fuselage data.*
DAMPSET	*Set up basic lag & fuselage damping pairs.*
GRES0	*Basic data – Frequency v Rotor Speed*
GRES1	*Basic Data – Instability v Rotor Speed*
GRES2	*Effective Head Mass Data – Instability v Rotor Speed*
GRES3	*Fuselage Frequency Data – Instability v Rotor Speed*
GRES4	*Lag Frequency Data – Instability v Rotor Speed*
GRES5	*Lag Frequency Data – Contour Plot with Lag & Fuselage Damping*
GRES6	*Set of Lag Frequency Data – Contour Plot with Lag & Fuselage Damping*
GRES7	*Lag Frequency Data – Surface Plot with Lag & Fuselage Damping*

GRSET

```
%     Program GRSET
%
%     This routine sets up the basic set of values to run ground resonance
calculations.
%
%     Dr. S. J. Newman - 13th January 2003
%
wl0=input('Static Rotor Lag Frequency (rads/s)');
wy=input('Fuselage Frequency (rads/s)');
e=input('Lag Hinge Offset (m)');
r=input('Rotor Radius (m)');
fusm=input('Fuselage Mass (kg)');
bladm=input('Blade Mass (kg)');
nblad=input('Number of Blades');

clpc=input('Critical Damping Lag (%)');

cypc=input('Critical Damping Fuselage (%)');

omst=input('Rotor Speed Start Value (rads/s)');
omfin=input('Rotor Speed Final Value (rads/s)');
ominc=input('Rotor Speed Increment Value (rads/s)');

effhdms=linspace(500,1500,3);

fusfrq=linspace(8,16,3);

lagfrq=linspace(0,20,5);
```

DAMPSET

```
%      Program DAMPSET
%
%      This routine sets up damping pairings to run ground resonance
calculations.
%
%      Dr. S. J. Newman - 13th January 2003
%
damp=[0 0;5 0;10 0;0 5;0 10;5 5;10 10];
dampy=linspace(0,100,21);
dampl=linspace(0,100,21);
```

GRES0

```
%     Program GRES0
%
%     CALCULATION OF THE FREQUENCIES OF GROUND RESONANCE V ROTOR SPEED -
Standard Frequency Sweep
%
% Shows the frequency variation with rotor speed
% apply damping on both rotor(clpc) and fuselage(cypc)as %ge critical
%
%     Dr. S. J. Newman - 27th April 2004
%
     b=r-e;
      xnm=nblad*bladm;
      xmu=xnm/2./(fusm+xnm);
      wl02=wl0^2;
      wy2=wy^2;
      bet=2.*wy*cypc/100.;

nom=round((omfin-omst)/ominc)+1;
freq=zeros(nom,6);
omg=zeros(nom,1);
%
% Commence Rotor Speed Sweep
%
for ii=1: nom; % Start of Rotor Frequency Loop

      om=omst+(ii-1)*ominc;
      omg(ii,1)=om;
      om2=om^2;
      wl2=wl02+om2*e/b;
      wl=sqrt(wl2);
      tmm=wl2-om2;
      alp=2.*wl*clpc/100.;
      alp2=alp^2;
      t1=alp2+2.*tmm;
      t2=tmm^2+alp2*om2;
      t3=tmm+2.*om2;
      n=6;
      a(1)=1.-xmu;
      a(2)=alp*(2.-xmu)+bet;
      a(3)=t1+2.*alp*bet+wy2+4.*om2-xmu*tmm;
      a(4)=bet*t1+2.*alp*t3+2.*(alp*wy2+2.*bet*om2);
      a(5)=t2+2.*alp*bet*t3+wy2*(alp2+2.*t3);
      a(6)=bet*t2+2.*alp*wy2*t3;
      a(7)=wy2*t2;
      rts=roots(a);
stby=imag(rts);

freq(ii,:)=stby';

end; % End of Rotor Frequency Loop
%
% set up plot
%
xmin=omst;
```

```
xmax=omfin;
ymin=0;
ymax=100;
%
% plot
%
plot(omg,freq);
% title plot
title('Ground Resonance Frequencies - Effect of Rotor Speed');
xlabel('Rotor Speed (rads/sec)');
ylabel('Frequencies (Imaginary Part) Roots');
grid on;
```

GRES1

```
%       Program GRES1
%
%       CALCULATION OF THE GROUND RESONANCE INSTABILITY V ROTOR SPEED - FOR
VARIOUS DAMPING PAIRINGS - Standard Frequency Sweep
%
% Shows the need to apply damping on both rotor and fuselage
%
%
%       Dr. S. J. Newman - 27th April 2004
%

[ndamp,drivel]=size(damp);

nom=round((omfin-omst)/ominc)+1;
bad=zeros(nom,ndamp);
omg=zeros(nom,1);

      for iicnt=1:ndamp; %  Commence Damping Loop

cypc=damp(iicnt,1);
clpc=damp(iicnt,2);

% Commence Main Inner Ground Resonance Loop

      b=r-e;
      xnm=nblad*bladm;
      xmu=xnm/2./(fusm+xnm);
      wl02=wl0^2;
      wy2=wy^2;
      bet=2.*wy*cypc/100.;

for ii=1: nom; %  Commence Rotor Speed Loop
      om=omst+(ii-1)*ominc;
      omg(ii)=om;
      om2=om^2;
      wl2=wl02+om2*e/b;
      wl=sqrt(wl2);
      tmm=wl2-om2;
      alp=2.*wl*clpc/100.;
      alp2=alp^2;
      t1=alp2+2.*tmm;
      t2=tmm^2+alp2*om2;
      t3=tmm+2.*om2;
      n=6;
      a(1)=1.-xmu;
      a(2)=alp*(2.-xmu)+bet;
      a(3)=t1+2.*alp*bet+wy2+4.*om2-xmu*tmm;
      a(4)=bet*t1+2.*alp*t3+2.*(alp*wy2+2.*bet*om2);
      a(5)=t2+2.*alp*bet*t3+wy2*(alp2+2.*t3);
      a(6)=bet*t2+2.*alp*wy2*t3;
      a(7)=wy2*t2;
      rts=roots(a);
stby=real(rts);
bad(ii,iicnt)=max(stby);
```

```
end;%  End Rotor Speed Loop


end;%  End Damping Loop

plot(omg,bad);
legm=num2str(damp);
legend(legm,0);

title('Ground Resonance - Effect of Head & Fuselage Damping ');
xlabel('Rotor Speed (rads/sec)');
ylabel('Instability (Real Part) Roots');

grid on;
```

GRES2

```
%      Program GRES2
%
%      CALCULATION OF THE GROUND RESONANCE INSTABILITY V ROTOR SPEED - FOR
VARIOUS EFFECTIVE HEAD MASSES - Standard Frequency Sweep
%
%      Dr. S. J. Newman - 27th April 2004
%
[drivel,nfusm]=size(effhdms);

nom=round((omfin-omst)/ominc)+1;


bad=zeros(nom,nfusm);
omg=zeros(nom,1);

      for iicnt=1:nfusm;% Commence Effective Head Mass Loop

% Commence Main Inner Ground Resonance Loop

fusm=effhdms(iicnt);

      b=r-e;
      xnm=nblad*bladm;
      xmu=xnm/2./(fusm+xnm);
      wl02=wl0^2;
      wy2=wy^2;
      bet=2.*wy*cypc/100.;

for ii=1: nom; % Start of Rotor Frequency Loop

      om=omst+(ii-1)*ominc;
      omg(ii)=om;
      om2=om^2;
      wl2=wl02+om2*e/b;
      wl=sqrt(wl2);
      tmm=wl2-om2;
      alp=2.*wl*clpc/100.;
      alp2=alp^2;
      t1=alp2+2.*tmm;
      t2=tmm^2+alp2*om2;
      t3=tmm+2.*om2;
      n=6;
      a(1)=1.-xmu;
      a(2)=alp*(2.-xmu)+bet;
      a(3)=t1+2.*alp*bet+wy2+4.*om2-xmu*tmm;
      a(4)=bet*t1+2.*alp*t3+2.*(alp*wy2+2.*bet*om2);
      a(5)=t2+2.*alp*bet*t3+wy2*(alp2+2.*t3);
      a(6)=bet*t2+2.*alp*wy2*t3;
      a(7)=wy2*t2;
      rts=roots(a);
stby=real(rts);
bad(ii,iicnt)=max(stby);

end; % End of Rotor Frequency Loop
```

```
end; % End of Effective Head Mass Loop

plot(omg,bad);
legm=num2str(effhdms');
legend(legm,0);

title('Ground Resonance - Variation of Effective Head Mass ');
xlabel('Rotor Speed (rads/sec)');
ylabel('Instability (Real Part) Roots');

grid on;
```

GRES3

```
%      Program GRES3
%
%      CALCULATION OF THE INSTABILITY OF GROUND RESONANCE V ROTOR SPEED -
FOR VARIOUS FUSELAGE FREQUENCIES - Standard Frequency Sweep
%
% Shows the effect of fuselage frequency
%
%      Dr. S. J. Newman - 27th April 2004
%
 [drivel,nfusfrq]=size(fusfrq);

nom=round((omfin-omst)/ominc)+1;
bad=zeros(nom,nfusfrq);
omg=zeros(nom,1);

      for iicnt=1:nfusfrq;% Commence Fuselage Frequency Loop


wy=fusfrq(iicnt);

% Commence Main Inner Ground Resonance Loop

      b=r-e;
      xnm=nblad*bladm;
      xmu=xnm/2./(fusm+xnm);
      wl02=wl0^2;
      wy2=wy^2;
      bet=2.*wy*cypc/100.;

for ii=1:nom;% Commence Rotor Speed Loop

      om=omst+(ii-1)*ominc;
      omg(ii)=om;
      om2=om^2;
      wl2=wl02+om2*e/b;
      wl=sqrt(wl2);
      tmm=wl2-om2;
      alp=2.*wl*clpc/100.;
      alp2=alp^2;
      t1=alp2+2.*tmm;
      t2=tmm^2+alp2*om2;
      t3=tmm+2.*om2;
      n=6;
      a(1)=1.-xmu;
      a(2)=alp*(2.-xmu)+bet;
      a(3)=t1+2.*alp*bet+wy2+4.*om2-xmu*tmm;
      a(4)=bet*t1+2.*alp*t3+2.*(alp*wy2+2.*bet*om2);
      a(5)=t2+2.*alp*bet*t3+wy2*(alp2+2.*t3);
      a(6)=bet*t2+2.*alp*wy2*t3;
      a(7)=wy2*t2;
      rts=roots(a);
stby=real(rts);
bad(ii,iicnt)=max(stby);

end;% End Rotor Speed Loop
```

```
end;% End Fuselage Frequency Loop

plot(omg,bad);
legm=num2str(fusfrq');
legend(legm,0);

title('Ground Resonance - Effect of Fuselage Frequency ');
xlabel('Rotor Speed (rads/sec)');
ylabel('Instability (Real Part) Roots');

grid on;
```

GRES4

```
%     Program GRES4
%
%     CALCULATION OF THE GROUND RESONANCE INSTABILITY  - FOR VARIOUS LAG
FREQUENCIES - Standard Frequency Sweep
%
%   Shows the effect of lag frequency
%
%     Dr. S. J. Newman - 27th April 2004
%
 [drivel,nlagfrq]=size(lagfrq);

nom=round((omfin-omst)/ominc)+1;
bad=zeros(nom,nlagfrq);
omg=zeros(nom,1);

      for iicnt=1:nlagfrq;% Commence Lag Frequency Loop

      wl=lagfrq(iicnt);
      wl2=wl^2;

% Commence Main Inner Ground Resonance Loop

      b=r-e;
      xnm=nblad*bladm;
      xmu=xnm/2./(fusm+xnm);
      wy2=wy^2;
      bet=2.*wy*cypc/100.;

for ii=1:nom;% Commence Rotor Speed Loop

      om=omst+(ii-1)*ominc;
      omg(ii)=om;
      om2=om^2;
      tmm=wl2-om2;
      alp=2.*wl*clpc/100.;
      alp2=alp^2;
      t1=alp2+2.*tmm;
      t2=tmm^2+alp2*om2;
      t3=tmm+2.*om2;
      n=6;
      a(1)=1.-xmu;
      a(2)=alp*(2.-xmu)+bet;
      a(3)=t1+2.*alp*bet+wy2+4.*om2-xmu*tmm;
      a(4)=bet*t1+2.*alp*t3+2.*(alp*wy2+2.*bet*om2);
      a(5)=t2+2.*alp*bet*t3+wy2*(alp2+2.*t3);
      a(6)=bet*t2+2.*alp*wy2*t3;
      a(7)=wy2*t2;
      rts=roots(a);
stby=real(rts);
bad(ii,iicnt)=max(stby);

end;% End Rotor Speed Loop


end;% End Lag Frequency Loop
```

```
plot(omg,bad);

legm=num2str(lagfrq');
legend(legm,0);

title('Ground Resonance - Effect of Lag Frequency ');
xlabel('Rotor Speed (rads/sec)');
ylabel('Instability (Real Part) Roots');

grid on;
```

GRES5

```
%	Program GRES5
%
%	CALCULATION OF DAMPING CONTOURS FOR GROUND RESONANCE INSTABILITY - 
Standard Frequency Sweep
%
%	Dr. S. J. Newman - 27th April 2004
%
[drivel,ndampy]=size(dampy);
[drivel,ndampl]=size(dampl);

worst=zeros(1);
grcont=zeros(ndampy,ndampl);

	for iiy=1:ndampy;% Start of Fuselage Damping Loop


cypc=dampy(iiy);

	for jjl=1:ndampl;

clpc=dampl(jjl);% Start of Lag Damping Loop



% Commence Main Inner Ground Resonance Loop

	b=r-e;
	xnm=nblad*bladm;
	xmu=xnm/2./(fusm+xnm);
	wl02=wl0^2;
	wy2=wy^2;
	bet=2.*wy*cypc/100.;

nom=round((omfin-omst)/ominc);
bad=zeros(nom,1);

for ii=1: nom;% Start of Rotor Speed Loop

	om=omst+(ii-1)*ominc;
	omg(ii)=om;
	om2=om^2;
	wl2=wl02+om2*e/b;
	wl=sqrt(wl2);
	tmm=wl2-om2;
	alp=2.*wl*clpc/100.;
	alp2=alp^2;
	t1=alp2+2.*tmm;
	t2=tmm^2+alp2*om2;
	t3=tmm+2.*om2;
	n=6;
	a(1)=1.-xmu;
	a(2)=alp*(2.-xmu)+bet;
	a(3)=t1+2.*alp*bet+wy2+4.*om2-xmu*tmm;
	a(4)=bet*t1+2.*alp*t3+2.*(alp*wy2+2.*bet*om2);
	a(5)=t2+2.*alp*bet*t3+wy2*(alp2+2.*t3);
```

```
      a(6)=bet*t2+2.*alp*wy2*t3;
      a(7)=wy2*t2;
      rts=roots(a);
stby=real(rts);
bad(ii)=max(stby);

end;% End of Rotor Speed Loop

worst=max(bad);
grcont(iiy,jjl)=worst;

end;% End of Lag Damping Loop


end;% End of Fuselage Damping Loop

% Set Up Contour Values

contour(dampy,dampl,grcont,11);
colormap(jet);
colorbar('vert');
title('Ground Resonance - Maximum Instability ');
xlabel ('Fuselage %ge Critical Damping');
ylabel ('Lag %ge Critical Damping');
grid on;
```

GRES6

```
%     Program GRES6
%
%     CALCULATION OF DAMPING CONTOURS FOR GROUND RESONANCE STABILITY -
Standard Frequency Sweep - Variation of Lag Frequency
%
%     Dr. S. J. Newman - 27th April 2004
%
[drivel,nlagfrq]=size(lagfrq);
[drivel,ndampy]=size(dampy);
[drivel,ndampl]=size(dampl);

      for iiwl=1:nlagfrq;% Start of Lag Frequency Loop

      wl=lagfrq(iiwl);
      wl2=wl^2;

      figure(iiwl);

      worst=zeros(1);
      grcont=zeros(ndampy,ndampl);

      for iiy=1:ndampy;% Start of Fuselage Damping Loop


cypc=dampy(iiy);

      for jjl=1:ndampl;% Start of Lag Damping Loop


clpc=dampl(jjl);


% Commence Main Inner Ground Resonance Loop

      b=r-e;
      xnm=nblad*bladm;
      xmu=xnm/2./(fusm+xnm);
      wy2=wy^2;
      bet=2.*wy*cypc/100.;

nom=round((omfin-omst)/ominc);
bad=zeros(nom,1);

for ii=1: nom;% Start of Rotor Speed Loop

      om=omst+(ii-1)*ominc;
      omg(ii)=om;
      om2=om^2;
      tmm=wl2-om2;
      alp=2.*wl*clpc/100.;
      alp2=alp^2;
      t1=alp2+2.*tmm;
      t2=tmm^2+alp2*om2;
      t3=tmm+2.*om2;
      n=6;
```

```
      a(1)=1.-xmu;
      a(2)=alp*(2.-xmu)+bet;
      a(3)=t1+2.*alp*bet+wy2+4.*om2-xmu*tmm;
      a(4)=bet*t1+2.*alp*t3+2.*(alp*wy2+2.*bet*om2);
      a(5)=t2+2.*alp*bet*t3+wy2*(alp2+2.*t3);
      a(6)=bet*t2+2.*alp*wy2*t3;
      a(7)=wy2*t2;
      rts=roots(a);
stby=real(rts);
bad(ii)=max(stby);

end;% End of Rotor Speed Loop

worst=max(bad);
grcont(iiy,jjl)=worst;

      end;% End of Lag Damping Loop


      end;% End of Fuselage Damping Loop

% Set Up Contour Values

heading=['Ground Resonance - Maximum Instability (Lag Frequency =
',num2str(wl),')'];

contour(dampy,dampl,grcont,11);
colormap(jet);
colorbar('vert');
xlabel ('Fuselage %ge Critical Damping');
ylabel ('Lag %ge Critical Damping');
title(heading);
grid on;

      end;% End of Lag Frequency Loop
```

GRES7

```
%       Program GRES7
%
%       CALCULATION OF DAMPING SURFACE FOR GROUND RESONANCE STABILITY -
Standard Frequency Sweep
%
%       Dr. S. J. Newman - 27th April 2004
%
[drivel,ndampy]=size(dampy);
[drivel,ndampl]=size(dampl);

worst=zeros(1);
grcont=zeros(ndampy,ndampl);

      for iiy=1:ndampy;% Start of Fuselage Damping Loop


cypc=dampy(iiy);

      for jjl=1:ndampl;

clpc=dampl(jjl);% Start of Lag Damping Loop



% Commence Main Inner Ground Resonance Loop

      b=r-e;
      xnm=nblad*bladm;
      xmu=xnm/2./(fusm+xnm);
      wl02=wl0^2;
      wy2=wy^2;
      bet=2.*wy*cypc/100.;

nom=round((omfin-omst)/ominc);
bad=zeros(nom,1);

for ii=1: nom;% Start of Rotor Speed Loop

      om=omst+(ii-1)*ominc;
      omg(ii)=om;
      om2=om^2;
      wl2=wl02+om2*e/b;
      wl=sqrt(wl2);
      tmm=wl2-om2;
      alp=2.*wl*clpc/100.;
      alp2=alp^2;
      t1=alp2+2.*tmm;
      t2=tmm^2+alp2*om2;
      t3=tmm+2.*om2;
      n=6;
      a(1)=1.-xmu;
      a(2)=alp*(2.-xmu)+bet;
      a(3)=t1+2.*alp*bet+wy2+4.*om2-xmu*tmm;
      a(4)=bet*t1+2.*alp*t3+2.*(alp*wy2+2.*bet*om2);
      a(5)=t2+2.*alp*bet*t3+wy2*(alp2+2.*t3);
```

```
      a(6)=bet*t2+2.*alp*wy2*t3;
      a(7)=wy2*t2;
      rts=roots(a);
stby=real(rts);
bad(ii)=max(stby);

end;% End of Rotor Speed Loop

worst=max(bad);
grcont(iiy,jjl)=worst;

end;% End of Lag Damping Loop


end;% End of Fuselage Damping Loop

surf(dampy,dampl,grcont);
colormap(jet);
colorbar('vert');
view(160,20);
title ('Ground Resonance - Surface of Maximum Instability');
xlabel ('Fuselage %ge Critical Damping');
ylabel ('Lag %ge Critical Damping');
zlabel ('Instability');

rotate3d on;
```

Results

Using the following data:

Parameter	MATLAB Variable Name	Value
Static Rotor Lag Frequency (rads/s)	wl0	15.22
Fuselage Frequency (rads/s)	wy	12
Lag Hinge Offset (m)	e	1.22
Rotor Radius (m)	r	6.4
Blade Mass (kg)	bladm	24.80
Number of Blades	nblad	4
Fuselage Mass (kg)	fusm	500

The above programs produce the following ground resonance behaviour:

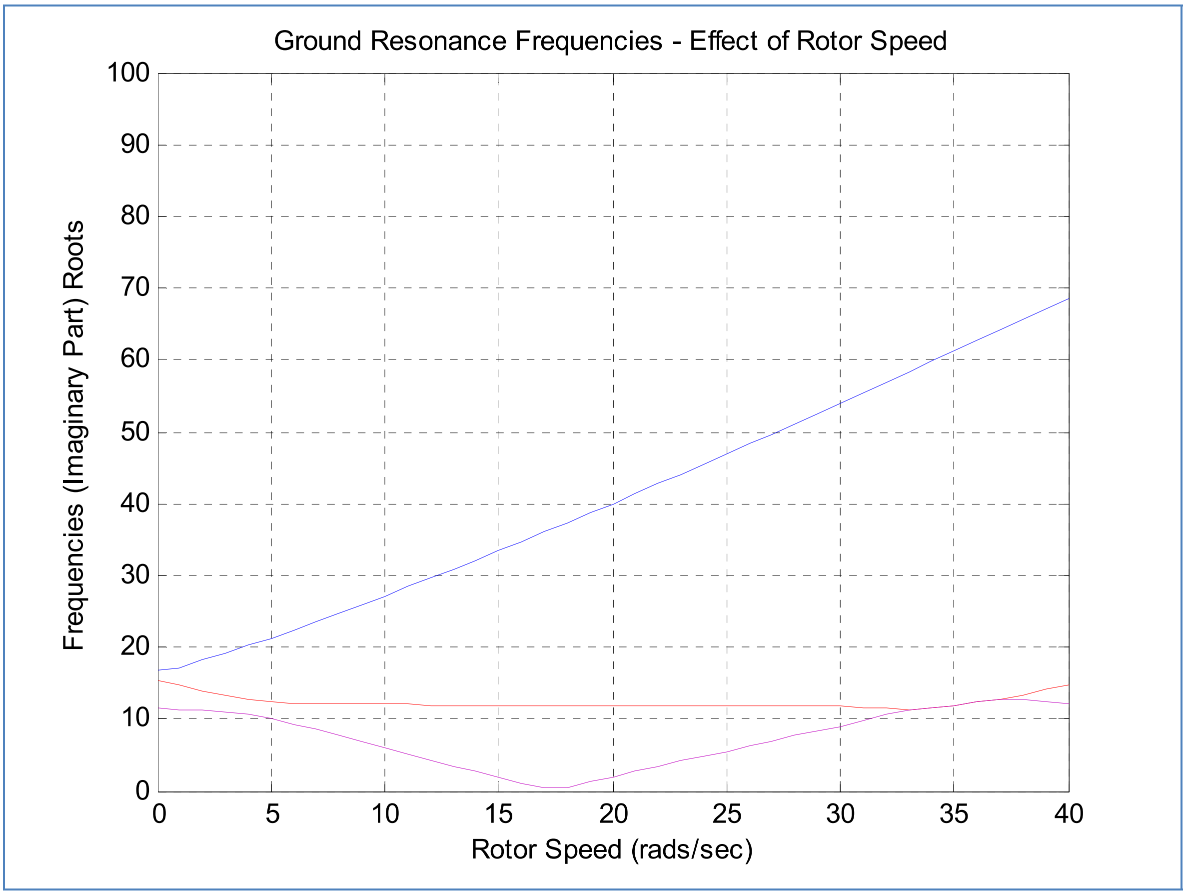

Figure 4 - Basic Frequency Variation (Imaginary Part) with Rotor Speed

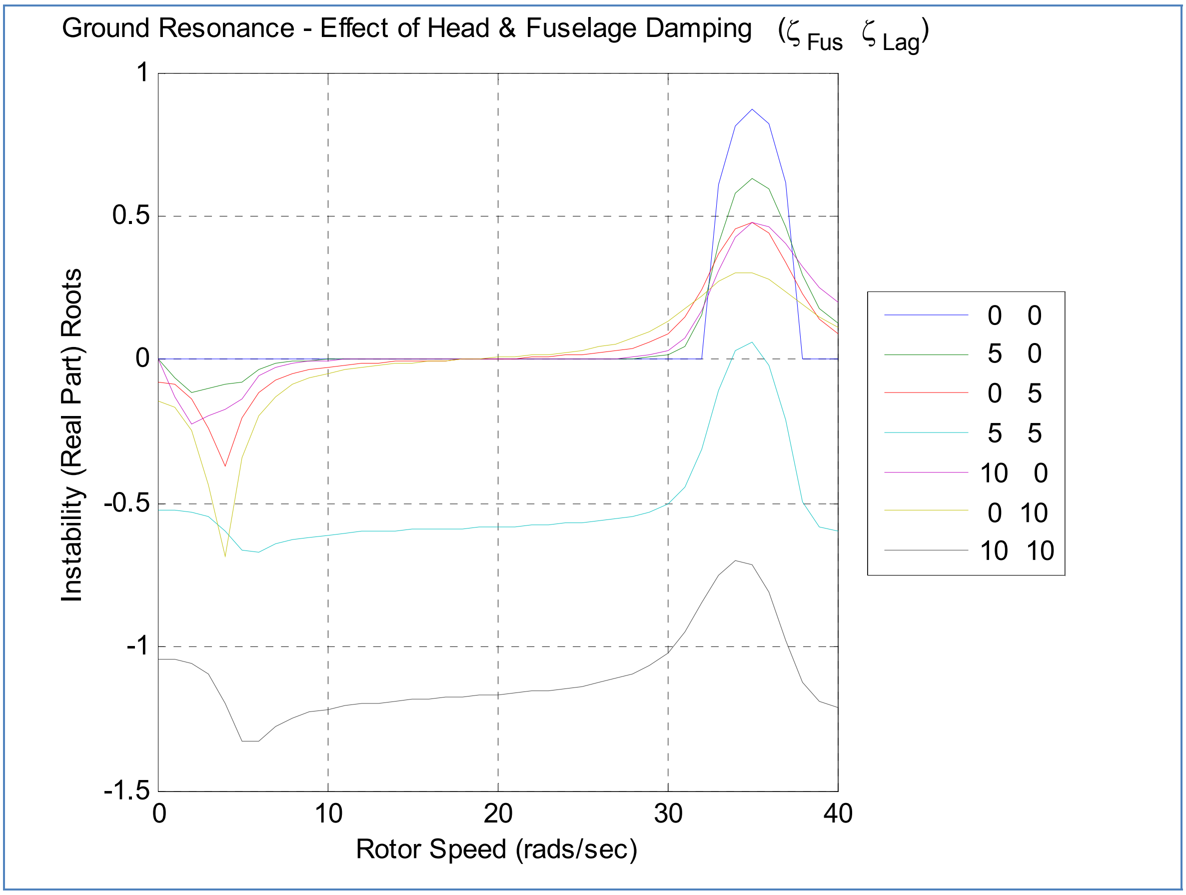

Figure 5 - Stability Variation (Real Part) with Rotor Speed

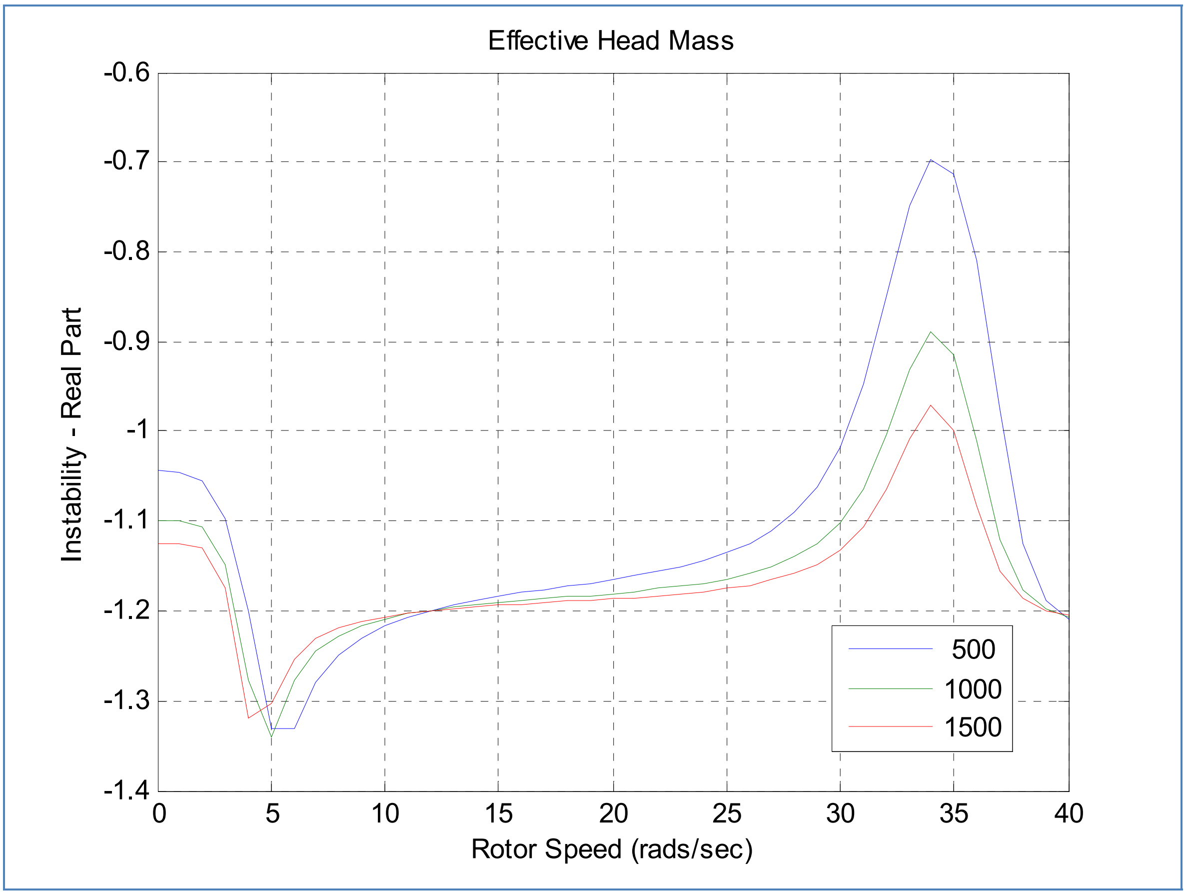

Figure 6 – Stability Variation – Influence of Effective Head Mass

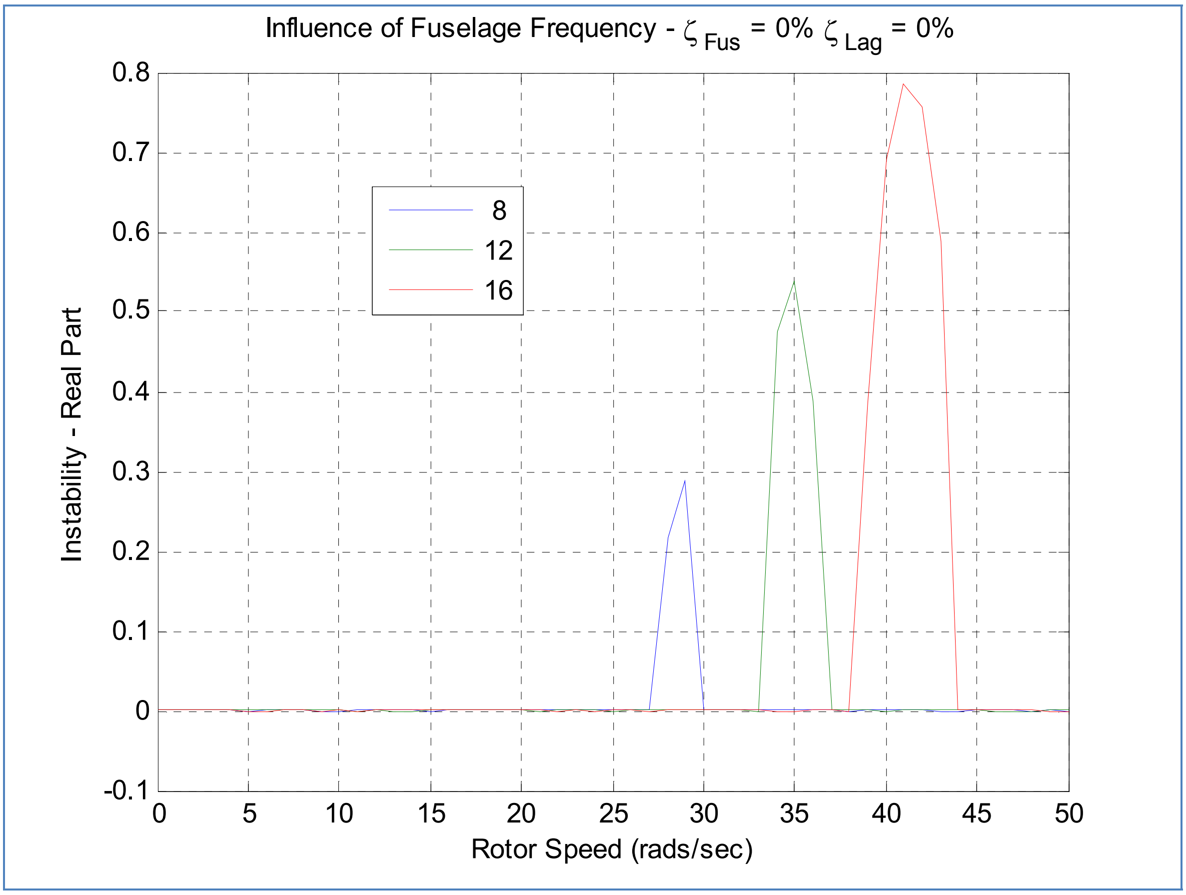

Figure 7 - Stability Variation - Influence of Fuselage Frequency

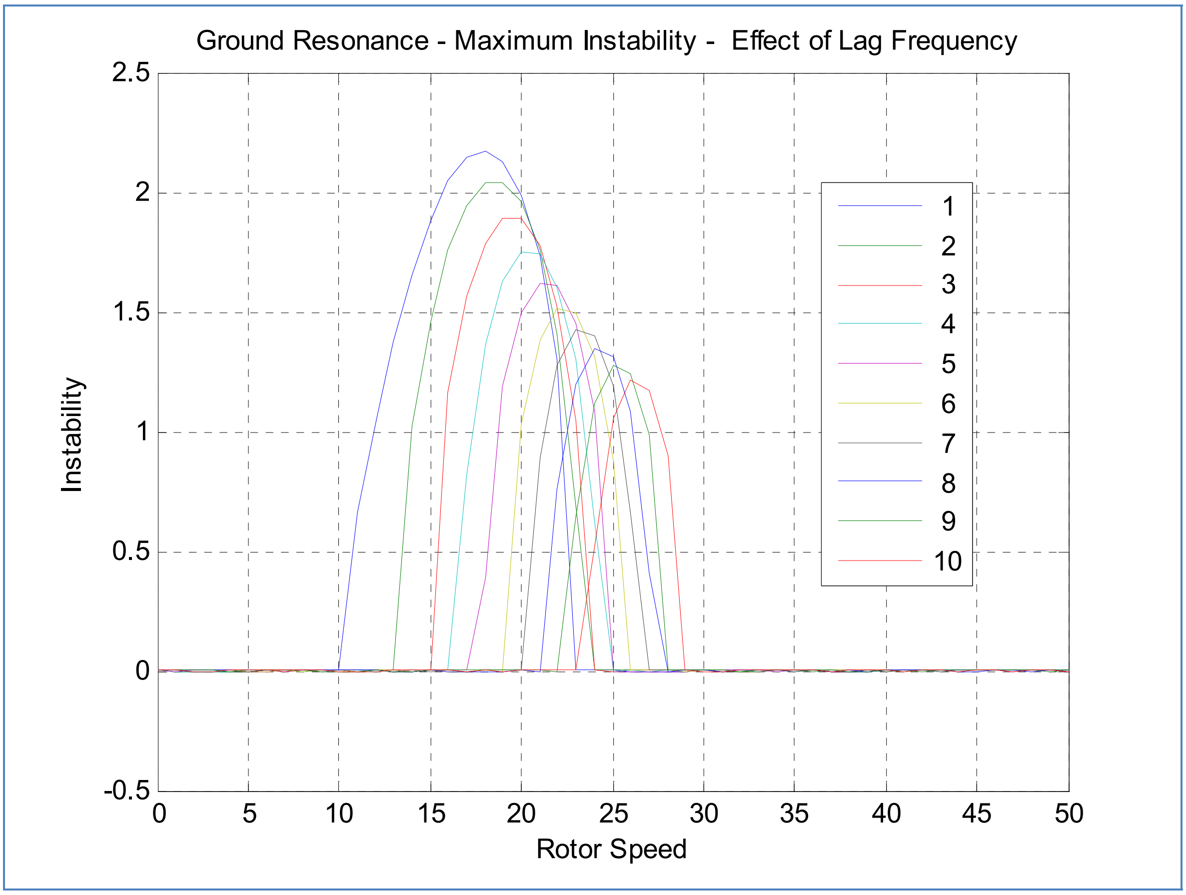

Figure 8 - Stability Variation – Influence of Lag Frequency

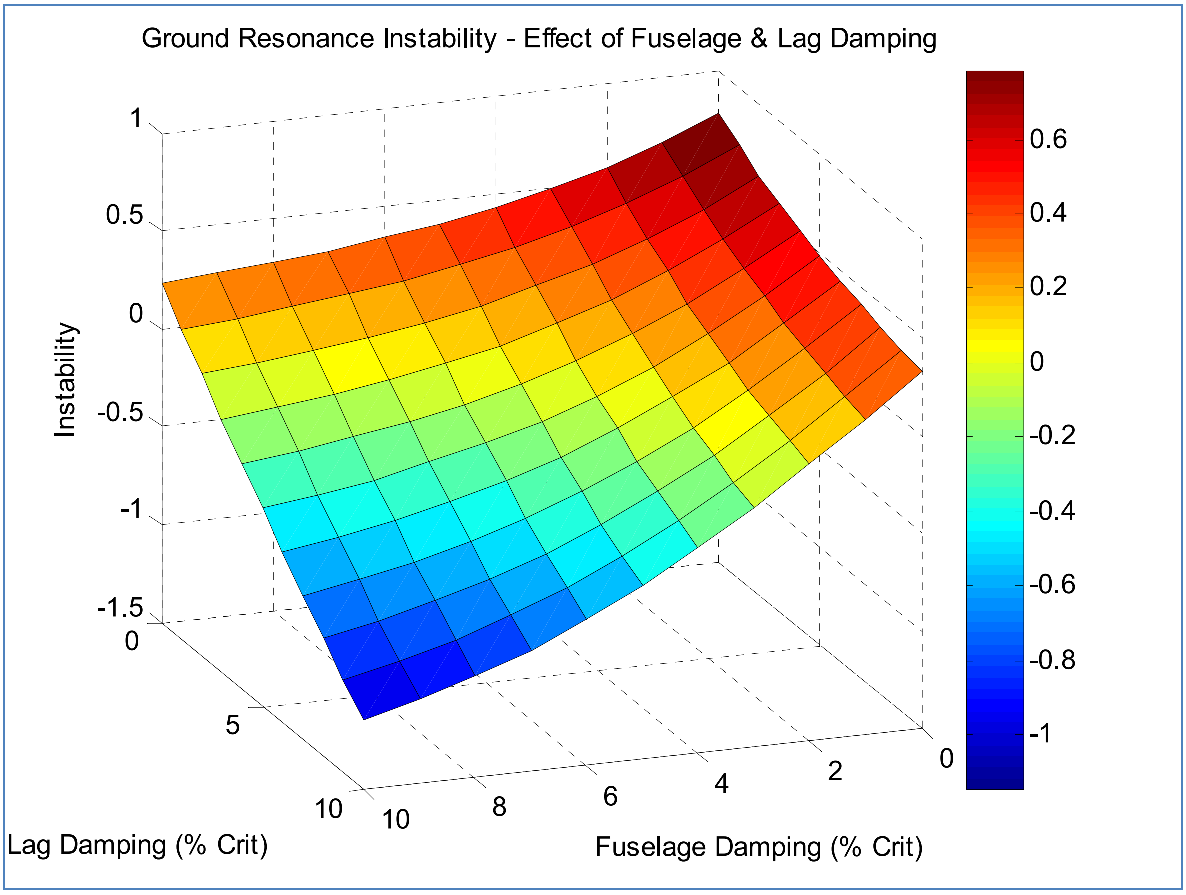

Figure 9 - Stability Variation - Influence of Fuselage & Lag Damping