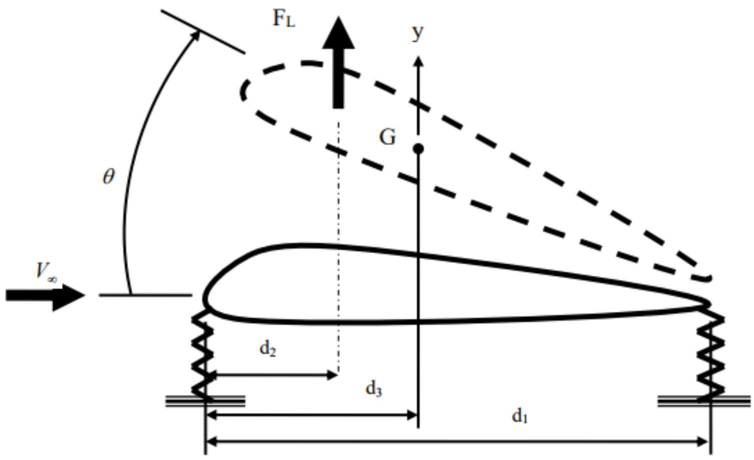
University of Sussex



Dynamic Aeroelasticity of an Aero-foil

Candidate Number: 146694

Tutor: Professor Julian Dunne

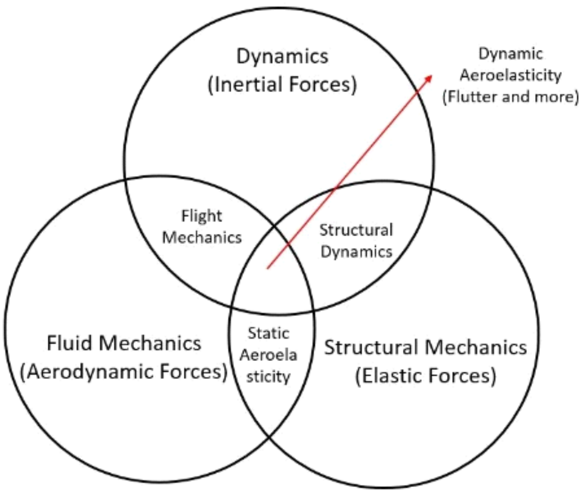
Jonathan Edwards

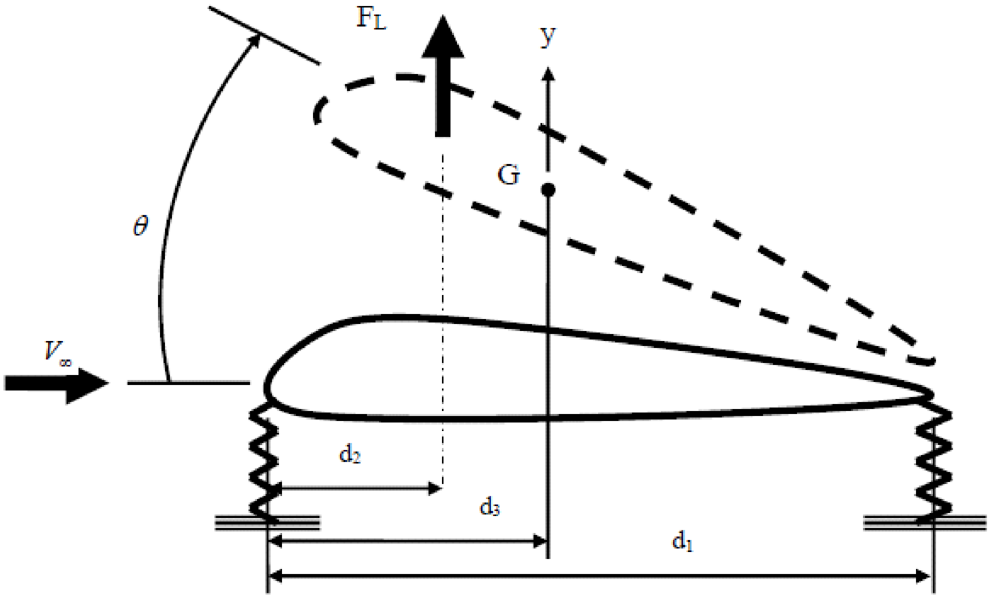
Introduction

Flutter or unstable oscillatory divergence is a form of Dynamic Aeroelasticity. It is an instability caused by the amalgamation of aerodynamic, inertial and elastic forces. In classical flutter analysis we consider the system to be linear and therefore we can state that 'flutter point' is the point in time at which the aerofoil undergoes simple harmonic motion. As the system has zero damping anything past the flutter point will result in a self-oscillation and eventual catastrophic failure.

Although this is a phenomenon which dominates the aerospace industry, it also has a significant effect on aerodynamic devices on race cars due to the high speeds and their main design aim of being as light as possible whilst still producing a significant amount of downforce.

This project will attempt to model the dynamic aeroelasticity of the aerofoil and predict the points of divergence, non-oscillatory divergence and stability. The mathematics will be worked out and then a program will be written MATLAB to aid with the computing.





Parameter	Value	Unit
D_1	0.393	m
D_2	0.100	m
D_3	0.165	m
b	1.250	m
Surface area	1.058	m
Air Density	1.225	Kg/m^3

Parameter	Value	Unit
V_∞	26.820	m/s
m	4.830	kg
I_G	0.6 Appendix	$Kg.m^2$
θ	10	°

Mathematical Calculations

3.1 Free Body Diagram

The free body diagram has been modelled

The displacement variables in this model are y which deals with translation and θ deals with rotation.

These motions are caused by the downforce (F_D) generated by the wing which can be calculated using equation

$$F_d = \frac{1}{2} C_L \rho A V^2$$

Where A is equal to the area of the wing

$$A = bd_1$$

B is the span of the wing. C_L is equal to the coefficient of lift and for small angles it can be approximated as 10θ.

Newton's second Law states

$$\sum F_R = \mathrm{m}\overline{a} = R_1 + R_2$$

$$\sum M = I\ddot{\theta}$$

3.2 Sum of the Forces

The sum of the force is equal to the sum of the reaction forces at each of the springs because these are the only forces that oppose the downforce.

Assuming linearity we can say the spring acts in accordance to Hooke's Law.

$$F = kx$$

Where F is the Force, x is the displacement and k is the spring constant

R_1 and R_2 can be deduced

$$R_1 \; = \; -k_2(y + \; \theta d_3\,)$$

$$R_1 \; = \; -k_2(y - \theta(d_1 - d_3\,))$$

R1 and R2 are negative due to the sign convention of the free body diagram. The displacement is taken as the summation of the y component displacement and the arc displacement.

The rc displacement can be calculated by multiplying the angle change with the distance from each respective spring to the centre of downforce.

$$\sum F_R = -k_1(\mathrm{y}+\theta d_3)\text{-}\; k_2\; (\mathrm{y}\text{-}\theta(d_1\text{-}d_3))$$

The function is then simplified to give the function in terms of y and theta

$$\sum F_R = \text{-}\mathrm{y}k_1 - \theta d_3 k_1 - \mathrm{y}k_2 + \theta k_2\; (d_1\text{-}d_3)$$

$$\sum F_R = y(-k_1 - k_2) + \theta(-d_3k_1 + k_2(d_1 - d_3))$$

3.3 Sum of Moments

Equations of the sum of the moments are now formulated

$$\sum M = I\ddot{\theta} = -(y+\theta d_3)k_1 d_3 + (y-(d_1-d_3)\,\theta)\,k_2\,(d_1-d_3)$$

And then simplified to give the function in terms of y and θ

$$\sum M = I\ddot{\theta} = -yd_3 - k_1\theta d{k_3}^2 + yk_2(d_1-d_3) - \theta k_2(d_1-d_3))^2$$

$$\sum M = I\ddot{\theta} = y(-k_1 d_3 + k_2(d_1 - d_3)) + \theta(-k_1{d_3}^2 - k_2(d_1-d_3)^2)$$

3.4 Simultaneous Equations

The governing equations are rearranged in order to be placed into matrix form. From Newton's second law two equations can be stated.

$$\sum F_R + F_D = \mathrm{m}\ddot{y}$$

$$\sum M + F_D(d_3 - d_2) = I\ddot{\theta}$$

After rearrangement

$$\mathrm{m}\ddot{y} - \sum F_R - F_D = 0$$

$$I\ddot{\theta} - \sum M - F_D(d_3 - d_2) = 0$$

After substitution and once the small angle approximation has been made

$$\mathrm{m}\ddot{y} - \sum F_R - \frac{1}{2}10\theta\rho A V^2 = 0$$

$$I\ddot{\theta} - \sum M - F_D\frac{(d_3 - d_2)}{2}10\rho A V^2 = 0 = 0$$

Simultaneous equations are now formed via substitution of the two governing equations.

$$(m\ddot{y}) - \Big(y(-k_1 - k_2) + \ \theta\big(-d_3k_1 + k_2(d_1 - d_3)\big)\Big) - (5\theta\rho AV^2) = 0$$

$$(I\ddot{\theta}) - (I\ddot{\theta} = y(-k_1 - d_3 + k_2(-d_1 - d_3) + \theta\left(-k_1{d_3}^2 - k_2(d_1 - d_3)^2\right)\Big) - \left((d_3 - d_2)5\theta\rho AV^2\right) = 0$$

3.5 Matrix

The simultaneous equations in the previous section can now be put into matrix form

$$\begin{bmatrix} M & 0 \\ 0 & I \end{bmatrix}\begin{bmatrix} \ddot{y} \\ \ddot{\theta} \end{bmatrix} + \begin{bmatrix} (k_1 + k_2) & (d_3k_1 - k_2(d_1 - d_3) - 5\theta\rho AV^2) \\ (k_1d_3 - \ k_2(d_1 - d_3)) & \left(k_1{d_3}^2 + k_2(d_1 - d_3)^2\right) - 5(d_3 - d_2)\theta\rho AV^2) \end{bmatrix}\begin{bmatrix} y \\ \theta \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Then divided through by the mass matrix

$$\begin{bmatrix} \ddot{y} \\ \ddot{\theta} \end{bmatrix} + \begin{bmatrix} \frac{(k_1+k_2)}{m} & \frac{(d_3k_1-k_2(d_1-d_3)-5\theta\rho AV^2)}{m} \\ \frac{(k_1d_3- \ k_2(d_1-d_3))}{I} & \frac{\left(k_1{d_3}^2+k_2(d_1-d_3)^2\right)-5(d_3-d_2)\theta\rho AV^2)}{I} \end{bmatrix}\begin{bmatrix} y \\ \theta \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

The above equation is equivalent to the one below

$$\begin{bmatrix} \ddot{y} \\ \ddot{\theta} \end{bmatrix} + \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}\begin{bmatrix} y \\ \theta \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$a_{11} = \frac{(k_1 + k_2)}{m}$$

$$a_{12} = \frac{(d_3k_1-k_2(d_1-d_3)-5\theta\rho AV^2)}{m}$$

$$a_{21} = \frac{(k_1d_3- \ k_2(d_1-d_3))}{I}$$

$$a_{22} = \frac{\left(k_1{d_3}^2+k_2(d_1-d_3)^2\right)-5(d_3-d_2)\theta\rho AV^2)}{I}$$

State Space Analysis

State space takes the form

$$\underline{\ddot{x}} = A\underline{x}$$

Here

$$\mathrm{A} = \begin{bmatrix} 0 & I \\ -[m]^{-1}[k] & 0 \end{bmatrix}$$

If

$$-[m]^{-1}[k] = \begin{bmatrix} -a_{11} & -a_{12} \\ -a_{21} & -a_{22} \end{bmatrix}$$

Then

$$\mathrm{A} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -a_{11} & -a_{21} & 0 & 0 \\ -a_{12} & -a_{22} & 0 & 0 \end{bmatrix}$$

The full state space model can be written

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \\ \dot{x_3} \\ \dot{x_4} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -a_{11} & -a_{21} & 0 & 0 \\ -a_{12} & -a_{22} & 0 & 0 \end{bmatrix}$$

Matlab code

```
%Wing Properties
d1=0.393;%Aerofoil length
d2=0.1; %Length to Lift Force
d3=0.165; %length to Centre of Mass
b=1.25; %aerofoil depth (NOT USED)
a=1.058; %Surface Area
m=4.830; %Mass of Aerofoil
theta=10; %Angle of Attack
Ig=0.6;  %Moment of Inertia

%Dynamic and Atmospheric Properties
Vi=26.82; %Car Velocity
Cl=10*theta; %Lift Coefficient
rho=1.225; %Density of air at sea level

%mesh grid Setup
xx=0:10:15000;
yy=0:10:15000;
[k1,k2]=meshgrid(xx,yy);

Non_Oscillatory_k1=zeros(1500);
Non_Oscillatory_k2=zeros(1500);
stable_k1=zeros(1500);
stable_k2=zeros(1500);
flutter_k1=zeros(1500);
flutter_k2=zeros(1500);

%stability for-loop
for i=1:length(k1);
    for j=1:length(k2)
   k1_i=k1(i,j);
   k2_i=k2(i,j);

%State Space Model
a11=(k1_i+k2_i)/m;
a12=(((k1_i*d3)-(k2_i*(d1-d3)))/m)-((5*rho*a*(Vi^2))/m);
a21=(((k1_i*d3)-(k2_i*(d1-d3)))/Ig);
a22=(((k1_i*d3^2)+(k2_i*(d1-d3)^2))/Ig)-(((5*rho*a*(Vi^2))*(d3-d2))/Ig);

Amatrix=[a11 a12;a21 a22];
Mmatrix=[m 0;0 Ig];%Mass Matrix
Kmatrix=[(k1_i+k2_i) (k1_i*(d1-d3))-(k2_i-(d1-d3));(k1_i*d3)-(k2_i*(d1-
d3)) ((k1_i*d3)^2)+((k2_i*(d1-d3))^2)];%Stiffness matrix
zero=zeros(2); %zero matrix
unit=eye(2); %unit matrix
Stability_Matrix=[zero unit;Amatrix zero]; %State Space matrix
Eigen_Values=eig(Stability_Matrix);%eigen values of the X_dot matrix

%Stability
A=((a11+a22 ))/2;
B=(a11*a22)-(a12*a21);
AA=A^2;
if (B<=0)
    Non_Oscillatory_k1(i,j)=k1_i;
```

```
    Non_Oscillatory_k2(i,j)=k2_i;

elseif (0<B)&&(B<=AA)
    disp('Stable Oscillation')
    stable_k1(i,j)=k1_i;
    stable_k2(i,j)=k2_i;

elseif (B>(AA))
    disp('Oscillatory Divergence "Flutter"')
 flutter_k1(i,j)=k1_i;
 flutter_k2(i,j)=k2_i;

end
    end
end
%Stability condition counters
No_of_non_Oscillatory=nnz(Non_Oscillatory_k1)
No_of_stable=nnz(stable_k1)
No_of_flutter=nnz(flutter_k1)

%Plotting commands
hold on
plot(stable_k1,stable_k2,'green')
plot(Non_Oscillatory_k1,Non_Oscillatory_k2,'yellow')
plot(flutter_k1,flutter_k2,'red')
hold off
```