AEROSERVOELASTIC ANALYSIS, DESIGN AND WIND TUNNEL TESTING OF A THREE DEGREE-OF-FREEDOM BINARY FLUTTER MODEL

Alan Neville Sutherland

A dissertation submitted to the Faculty of Engineering and the Built Environment, School of Mechanical, Industrial and Aeronautical Engineering, University of the Witwatersrand, in fulfilment of the requirements for the degree of Master of Science in Engineering.

Johannesburg, March 2011

Declaration

I declare that except where indicated, this dissertation is my own unaided work. It is being submitted for the Degree of Master of Science in Engineering to the School of Mechanical, Industrial and Aeronautical Engineering in the University of the Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination to any other University.

....
A.N. Sutherland

....
Date

Abstract

Aeroelastic problems are typically limiting factors in the performance and flight envelopes of aircraft. Aeroelastic constraints can be relaxed through active control of the flutter modes at and above the uncontrolled flutter speed of aircraft using primary flight control surfaces. To demonstrate the concept and associated benefits of aeroservoelasticity, a low speed, three degree-of-freedom binary flutter model incorporating a full-span trailing edge control surface was designed, modelled mathematically and wind tunnel tested. Open-loop flutter test results agreed well with predictions from classical flutter theory, and gentle low speed flutter was demonstrated. Closed-loop wind tunnel testing using a linear quadratic Gaussian controller proved that flutter could be suppressed successfully, allowing the flutter boundary of the model to be increased significantly. Flutter suppression was demonstrated up to a speed 54*m/s,* an increase of 116% over the model's open-loop flutter speed of 25*m*/*s*, using a controller optimised for an airspeed of 25*m*/*s*.

"When once you have tasted flight, you will forever walk the earth with your eyes turned skyward, for there you have been, and there you will always long to return."

Leonardo da Vinci (1452 – 1519)

Acknowledgements

The following people and institutions are thanked and acknowledged in appreciation of their assistance with this research:

- My supervisors, Dr. Craig Law and Prof. Brian Wigdorowitz for their advice and guidance
- Mr. Louw van Zyl, Mr. John Monk and other CSIR colleagues for their help and technical inputs
- My father Mr. Andrew Sutherland, whom I am especially grateful to for his continued support, interest and help (technical or otherwise) throughout this research. He is also recognised for planting the seeds derived from his own research at the then National Institute for Aeronautics and Systems Technology (NIAST) of South Africa, that would later become the success of this research
- The Aeronautic Systems Competency (ASC) of the Council for Scientific and Industrial Research (CSIR) for their continued and generous funding of this work and for the extensive use of their facilities and wind tunnels, without which the research would have not been possible
- The National Aerospace Centre (NAC) for funding the data acquisition system and MathWorks™ software

Table of Contents

List of Figures

Appendix Figures

List of Tables

Appendix Tables

List of Symbols

Constants

a	-	Non-dimensional distance between aerofoil mid-chord and elastic axis
b	$[m]$	Aerofoil semi-chord
c	-	Non-dimensional distance between aerofoil mid-chord and control surface hinge point
c_h	$[Ns/m]$	Aerofoil plunge viscous damping coefficient
c_α	$[Ns]$	Aerofoil pitch viscous damping coefficient
c_β	$[Ns]$	Aerofoil control surface viscous damping coefficient
d	$[m]$	Flexure diameter
D'	$[J]$	Modal damping/dissipation function per unit wingspan
E	$[N/m^2]$	Young's modulus
f	$[Hz.]$	Frequency
f_p	$[Hz.]$	Peak frequency
F	$[N]$	Generalised force
F_p	-	Peak amplitude
g	$[m/s^2]$	Acceleration due to gravity
g_a	$[Ns/m]$	Artificial damping
g_h	$[Ns/m]$	Aerofoil plunge structural damping coefficient
g_α	$[Ns]$	Aerofoil pitch structural damping coefficient
g_β	$[Ns]$	Aerofoil control structural viscous damping coefficient
G	$[N/m^2]$	Shear modulus
G_A	-	Strain gauge amplifier gain
h	$[m]$	Aerofoil plunge displacement
i	-	$\sqrt{-1}$
I	$[kgm^2]$	Mass inertia
I_R	$[kgm^2]$	Mass inertia about R
I_{xx}	$[kgm^2]$	Mass inertia about *x-x* axis
I_{yy}	$[kgm^2]$	Mass inertia about *y-y* axis
I_α	$[kgm^2]$	Mass inertia of a typical aerofoil section
I_β	$[kgm^2]$	Mass inertia of the control surface of a typical aerofoil section
J	$[m^4]$	Polar second moment of area

J_{LQR}	-	LQR feedback control cost function
k	-	Reduced frequency (Strouhal number)
k_h	$[N/m]$	Aerofoil plunge stiffness coefficient
k_α	$[Nm]$	Aerofoil pitch stiffness coefficient
k_β	$[Nm]$	Aerofoil control surface stiffness coefficient
k_{SGB}	-	Strain gauge bridge factor
ℓ	-	Unsteady aerodynamic lag state
L	$[m]$	Length
L	$[N]$	Unsteady aerofoil lift force
m	$[kg]$	Aerofoil mass
M_α	$[Nm]$	Unsteady aerofoil pitching moment
M_β	$[Nm]$	Unsteady aerofoil control surface hinge moment
N	-	Number of flexures
P	-	Complex eigenvalue
q	-	Generalised coordinate
Q	-	Unsteady aerodynamic circulatory flow parameter
R	$[m]$	Pitch circle diameter radius
R_G	$[\Omega]$	Amplifier gain resistor value
s	-	Laplace domain parameter
s_w	$[m]$	Wing span
t	$[m]$	Flexure thickness
t	$[s]$	Time
T'	$[J]$	Aerofoil kinetic energy per unit wingspan
$T_1 \cdots T_{13}$	-	Theodorsen's functions
u	-	Control input variable
U	$[m/s]$	Airspeed
U'	$[J]$	Aerofoil potential energy per unit wingspan
V	$[V]$	Voltage
w	$[m]$	Flexure width
x_α	-	Non-dimensional distance between aerofoil mid-chord and cg
x_β	-	Non-dimensional distance between control surface hinge point and cg
Z	-	Zimmerman flutter margin
Z_s	-	Simplified Zimmerman flutter margin
$\bar{Z}_s$	-	Simplified Normalised Zimmerman flutter margin
α	$[rad.]$	Aerofoil pitch angle

α_e	-	State estimator characteristic equation
α_r	-	State regulator characteristic equation
β	$[rad.]$	Aerofoil control surface angle
β_a	$[rad.]$	Control surface actuator position
β_c	$[rad.]$	Commanded aerofoil control surface angle
Δ_α	$[rad.]$	Pitch SGB calibration surface offset term
Δ_h	$[m]$	Plunge SGB calibration surface offset term
Δf	$[Hz.]$	Frequency bandwidth
ε	$[mm/mm]$	Strain
Γ	$[m^3/s]$	Fluid circulation
λ	-	Complex eigenvalue
μ	-	Normal probability distribution mean value
θ	$[rad.]$	Flexure angle
ρ	$[kg/m^3]$	Air density
σ	-	Standard deviation
τ	$[s]$	Time
ω	$[rad./s]$	Coupled structural frequency
ζ	$[N/m]$	Modal decay parameter

Vectors and Matrices

$\mathbf{A}$	State-space system matrix
$\mathbf{A}'$	Augmented state-space system matrix
$\mathbf{B}$	State-space input matrix
$\mathbf{B}'$	Augmented state-space input matrix
$\mathbf{C}$	State-space measurement (output) matrix
$\mathbf{C}'$	Augmented state-space measurement (output) matrix
$\mathbf{C}_o$	Controllability matrix
$\mathbf{D}$	Direct feed-through state-space matrix; Damping matrix
$\bar{\mathbf{D}}$	Combined damping matrix
$\mathbf{e}$	Estimation error vector
$\mathbf{F}$	Force Matrix
$\mathbf{g}$	Matrix of modal structural dampings
$\mathbf{G}$	Process noise coefficient matrix
$\mathbf{I}$	Identity matrix

$\mathbf{K}$	Stiffness matrix
$\mathbf{K}_{LQR}$	LQR feedback vector
$\bar{\mathbf{K}}$	Combined stiffness matrix
$\mathbf{L}$	Observer feedback matrix
$\mathbf{L}_{Kalman}$	Kalman filter gain matrix
$\mathbf{L}_{\delta}$	Unsteady aerodynamic lag state δ coefficient matrix
$\mathbf{L}_{\lambda}$	Unsteady aerodynamic lag state λ coefficient matrix
$\mathbf{M}$	Mass matrix
$\bar{\mathbf{M}}$	Combined mass matrix
$\mathbf{O}_{b}$	Observability matrix
$\mathbf{P}$	Vector of eigenvalues
$\mathbf{P}_{Kalman}$	Kalman filter state error covariance matrix
$\mathbf{q}$	Generalised modal vector
$\mathbf{Q}$	Unsteady aerodynamic circulatory flow parameter matrix
$\mathbf{Q}_{Kalman}$	Kalman filter process noise covariance matrix
$\mathbf{Q}_{LQR}$	Linear quadratic regulator state weighting matrix
$\mathbf{R}_{Kalman}$	Kalman filter measurement noise covariance matrix
$\mathbf{R}_{LQR}$	Linear quadratic regulator control cost matrix
v	Measurement noise vector
w	Process noise vector
$\mathbf{V}$	Strain gauge bridge calibration input voltage vector
$\mathbf{X}$	State-space system states vector
$\mathbf{X}'$	Augmented state-space system states vector
$\hat{\mathbf{X}}$	State-space system estimated states vector
$\mathbf{Y}$	State-space measurements vector
$\boldsymbol{\delta}$	Real part of the complex coefficient matrix
$\boldsymbol{\Delta}$	Strain gauge bridge calibration intercept matrix
$\boldsymbol{\varepsilon}$	Strain gauge bridge calibration slope matrix
$\boldsymbol{\Phi}$	Strain gauge bridge calibration modal displacement vector
$\boldsymbol{\eta}$	Imaginary part of the complex coefficient matrix
$\boldsymbol{\lambda}$	Complex coefficient matrix
$\boldsymbol{\Omega}$	Unsteady aerodynamic force matrix
$\boldsymbol{\omega}$	Matrix of modal frequencies
$\boldsymbol{\zeta}$	Matrix of modal decay rates

Mathematical Functions

$C(k)$	Theodorsen's function
$D(t)$	Duhamel integral
$E\{\mathbf{X}\}$	Expected value operator
$f*g$	Convolution of the functions *f* and *g*
$\mathcal{F}$	Fourier Transform
$\mathcal{F}^{-1}$	Inverse Fourier Transform
$H(k)$	Hänkel function
$p(A)$	Probability function
$N\left(\mu,\sigma^2\right)$	Normal (Gaussian) probability distribution function
$\Phi(U,t,b)$	Wagner's function
$\dot{}$	First derivative with respect to time (velocity)
$\ddot{}$	Second derivative with respect to time (acceleration)

Subscripts

a	Acceleration; Aerodynamic; Actuator
c	Circulatory; Control
CL	Closed-loop
d	Discrete
D	Design
f	Flutter
h	Related to the plunge degree-of-freedom
i	Indexing number
nc	Non-circulatory
OL	Open-loop
p	Peak
R	Rayleigh
s	Structural
v	Velocity
w	Wing
α	Associated with the pitch degree-of-freedom
β	Associated with control surface degree-of-freedom

Superscripts

(C)	Compressive
T	Matrix transpose
(T)	Tensile
(0)	Function evaluated at zero

Nomenclature

2D	Two-Dimensional
3D	Three-Dimensional
3DOF	Three Degree-of-Freedom
ac	Aerodynamic Center
BACT	Benchmark Active Controls Technology
BNC	Bayonet Nut Coupling
cg	Centre of Gravity
CAD	Computer Aided Design
CCV	Control Configured Vehicle
CL	Closed-loop
CSIR	Council for Scientific and Industrial Research
CWT	Calibration Wind Tunnel
DAQ	Data Acquisition System
DC	Direct Current
DIFF	Differential
DLM	Doublet Lattice Method
ea	Elastic Axis
FBE	Flutter Boundary Extension
FAST	Flutter with Active Suppression Technology
FFT	Fast Fourier Transform
FMC	Flutter Mode Control
FSS	Flutter Suppression System
KCAS	Knots Calibrated Airspeed
KIAS	Knots Indicated Airspeed
LCO	Limit Cycle Oscillation
LMI	Linear Matrix Inequality
LPV	Linear Parameter Varying
LQG	Linear Quadratic Gaussian
LQR	Linear Quadratic Regulator
LSWT	Low Speed Wind Tunnel
LTI	Linear Time Invariant
LTR	Loop Transfer Recovery

MIMO	Multi-Input Multi-Output
NACA	National Advisory Committee for Aeronautics
NASA	National Aeronautics and Space Administration
NRSE	Non-Referenced Single-Ended
ODE	Ordinary Differential Equation
OL	Open-loop
PCB	Printed Circuit Board
PID	Proportional Integral Derivative
PCD	Pitch Circle Diameter
PSD	Power Spectral Density
PWM	Pulse Width Modulation
RH	Relative Humidity
RSE	Referenced Single-Ended
SGB	Strain Gauge Bridge
SISO	Single-Input Single-Output
SQL	Structured Query Language
UAS	Unmanned Aerial System
USB	Universal Serial Bus

Chapter 1

Introduction

Airframe structural design trends have shown a steady increase in flexibility, slenderness ratio and maximum operating speeds over time (Thompson and Kass, 1971). Combined, these factors can lead to flutter and destruction of the airframe. Consequently, aeroelastic problems are invariably the limiting factors in the performance of all aircraft (Leishman and Nguyen, 1990). Traditional methods of avoiding flutter are sufficiently stiff structural design, modification of the aircraft mass distribution or placing of limitations on the operation of the aircraft. These methods are termed passive flutter suppression methods, and are undesirable as they can lead to expensive airframe structural modifications and often introduce weight penalties that restrict the flight envelope of the aircraft (Bradshaw *et al.*, 1986). A modern method of avoiding flutter is to actively prevent or damp out divergent airframe structural oscillations using primary aircraft flight control surfaces. The concept of active flutter suppression is attractive because reducing aeroelastic constraints through this means allows lighter airframe design and wider aircraft operating envelopes than is currently possible. In addition, active flutter suppression control systems allow military aircraft to operate with store configurations that would otherwise cause flutter. Active flutter suppression systems are therefore an important consideration in the design of modern high performance aircraft and unmanned aerial systems (UASs).

1.1 Background and Motivation

Active flutter control merges the engineering disciplines of aeroelasticity (a combination of elastic, inertial and aerodynamic forces) and active feedback control. The combination of these two fields is commonly termed *aeroservoelasticity* (Collar, 1946). The idea of aeroservoelasticity is not a new one. In the chapter “An Elementary Explanation of the Flutter Mechanism” in Flomenhoft (1997), Samuel Pines suggested that with more advanced computers and improved theories of aerodynamic and structural analysis available now (and when the chapter was written), compared to those available to early aeroelasticians, it is possible to actively suppress flutter. In Pines’ words; “We may even be able to design feedback control loops that permit flutter-free flight throughout the entire desired flight regime”. Thompson and Kass (1971) summarised some of the earliest research efforts in active aeroelastic control of aircraft structures. They highlighted that up until the time of their publication (1971) much of the work on active aeroelastic control focused on suppression of unstable structural modes, citing examples from Boeing, Honeywell, Lockheed and other aerospace companies. They concluded that active flutter suppression research is technically feasible and active flutter suppression control systems offer significant weight saving and performance benefits for aircraft.

An effective means of investigating and demonstrating aeroservoelasticity is by designing, building and testing wind tunnel models. Much of the active flutter control research conducted to date used simple two-dimensional (pitch-plunge) flutter models, as these facilitated simple mathematical modelling and safe testing. This approach is still used as it allows the emphasis of the research to be placed on the design and implementation of different control laws, making pitch-plunge flutter models useful test-beds for assessing new control laws. Control laws developed using pitch-plunge wind tunnel flutter models can then be applied in active flutter control of three-dimensional aircraft structures. Two common binary[1] wind tunnel flutter model design approaches are reported in the literature. The first approach employs flexures, whilst the second uses coil springs to achieve the desired pitch and plunge modal behaviour. Most models actively suppress flutter by altering aerodynamic forces through a trailing edge control surface, although in some cases both leading and trailing edge control surfaces are used. Flutter suppression is typically achieved through linear time-domain control systems using state-space design techniques. Wind tunnel testing of these models involves first assessing their open-loop (control system deactivated, i.e. uncontrolled) dynamics and then evaluating the control system effectiveness at the observed open-loop flutter speed. Thereafter, the amount by which the flutter boundary is extended with the application of control can be determined.

1.2 Research Hypothesis

The hypothesis of this research was that it is feasible to design, mathematically model and wind tunnel test a small-scale, three degree-of-freedom binary flutter model to demonstrate sustained open-loop flutter and active flutter suppression at and significantly beyond this uncontrolled flutter speed.

1.3 Aims and Objectives

The aim of this research was to physically investigate and demonstrate active flutter suppression of a small-scale pitch-plunge wind tunnel flutter model (representative of an aircraft wing) in the Calibration Wind Tunnel (CWT) and Low Speed Wind Tunnel (LSWT) of the Council for Scientific and Industrial Research (CSIR). To achieve this, the following objectives were met:

- The design and manufacture of a model wing incorporating a trailing edge control surface, control surface actuation mechanism and appropriate sensors for open- and closed-loop (control system activated, i.e. controlled) flutter testing on an existing, small-scale, pitch-plunge flexure mount system
- The complete mathematical modelling of the three degree-of-freedom binary flutter model and the design of an active flutter suppression controller using a linear quadratic Gaussian (LQG) control law

---

[1] Flutter that results from the coupling of three structural modes is termed ternary flutter whilst flutter resulting from the coupling between two dominant modes is referred to as binary flutter (Fung, 1955).

- The integration of the wind tunnel flutter model and mount into the CWT and LSWT of the CSIR
- Physical implementation of the LQG control law by means of a suitable data acquisition system (DAQ) and custom software
- Demonstration of flutter suppression of a developed flutter cycle in under 5*s* for different model center of gravity (cg) position configurations, using the trailing edge control surface to maintain the plunge displacement and pitch angle of the model within ±0.5*mm* and ±0.25°, respectively
- Flutter boundary extension (FBE) demonstration with the model in its baseline configuration to speeds significantly in excess (at least 100%) of the open-loop flutter speed

1.4 Delineation

The active flutter control research conducted in this study was carried out using a rigid wing model on an existing flexible mount that allows only pitch-plunge motion (Sutherland, 2006; 2008). The structural dynamics of the mount were modelled using Lagrange's equation (Tse *et al.*, 1978) with a Rayleigh dissipation function to account for viscous structural damping. Under the assumption of small oscillations of all vibration modes, the equations of motion were linearised about the neutral point (trim position) of the binary flutter system. Theodorsen (1935) showed that full three-dimensional unsteady aerodynamic modelling is not necessary for two-dimensional binary flutter models designed for low flutter speeds. Consequently, two-dimensional unsteady aerodynamic theory (Theodorsen, 1935) with appropriate analytical approximations of certain terms (Fung, 1955) was used. The underlying assumptions of this theory are that the aerofoil is thin with an infinite aspect ratio, modal oscillations are small and the flow over the wing remains potential and unseparated (Fung, 1955). Several standard linear control laws were considered, but in this work only a single, constant, linear quadratic Gaussian control law was implemented. It was not intended that this implementation be fully optimised or completely robust at off-design points, and external disturbance inputs (such as wind tunnel turbulence) were not directly included in the control law design, but accounted for to some extent in the Kalman filter used. Nonlinearities in the system were minimised and/or eliminated by careful design, so they too were excluded from the mathematical model of the active flutter suppression control system. It was assumed that the mass inertia of the wing about its elastic axis did not change significantly when the control surface was actuated, a reasonable assumption given its small size, low mass and small angular deflections.

1.5 Equipment and Experimental Method

This section gives brief details of the wind tunnel flutter model, instrumentation and data acquisition hardware. Overviews of the wind tunnels in which the flutter tests were done and descriptions of the test procedure, data acquired and data analysis are also provided.

1.5.1 Active Flutter Control Model, Instrumentation and Wind Tunnels

The active flutter suppression model comprised an existing pitch-plunge cantilevered flexure mount and a newly designed rigid wing with a full-span trailing edge control surface. The flexible mount consisted of four rectangular flexures that permitted only pitch and plunge motion of the rigid wing attached to it. The flexures were instrumented with one full strain gauge bridge (SGB) to measure the plunge displacement of the wing and one to measure its pitch angle. The pitch and plunge modal frequencies could be tailored by changing the angles of the rectangular flexures. The flutter characteristics of the model/mount could be modified by changing its mass inertia using sliding ballast masses on arms attached to the flexure mount. The control surface could be driven by either an internal model aircraft servo via control rods fixed on a torsion spring, or by a more powerful externally mounted linear actuator and crank mechanism. The modular design of the wing allowed relatively easy fitting and/or exchange of these actuation systems. The servo was driven by a pulse width modulated (PWM) signal, whilst an analogue voltage signal was used to drive the linear actuator. A Hall-effect magnetic sensor attached to the pivot shaft of the control surface was used to measure the control surface angle. A National Instruments™ NI USB-6211 multi-function data acquisition (DAQ) module was used to acquire all data and generate the actuation signal for the control surface actuator. Most of the wind tunnel tests were done in the CSIR CWT ($800mm \times 600mm$), which has a maximum speed of nominally $35m/s$. In order to test the model and flutter suppression controller at higher speeds, additional tests (up to a maximum speed of $53.9m/s$) were done in the CSIR LSWT ($2.1m \times 1.5m$). The model system was integrated into each of these wind tunnels by means of custom designed and built support structures. Each of these wind tunnels is fully instrumented to measure ambient temperature, pressure and airspeed within the test section. Full details of the active flutter control model, instrumentation and control system software are given in Chapter 4. A description of the two wind tunnels is given in Section 5.2.

1.5.2 Test Procedure

Open-loop testing was conducted to verify and validate the mathematical model of aeroservoelastic system and so ensure an effective control law design for the wind tunnel model. Having verified the open-loop response of the model, closed-loop testing was conducted to demonstrate active flutter suppression of the model and the extent to which the flutter boundary could be extended. A brief overview of the test procedures is given below.

Open-loop Testing

Open-loop testing involved safely approaching the anticipated flutter speed (calculated using classical unsteady aerodynamic theory) of the model by analysing its response at subcritical airspeeds. The simplified Zimmerman flutter margin, peak-hold and half-power bandwidth methods were initially used for this purpose. Due to the simplicity of the wind tunnel model and its well defined structural modes, the simplified Zimmerman flutter margin technique proved reliable and was used extensively in subsequent wind tunnel testing. Starting well below the analytically predicted flutter speed, a flutter margin based on the subcritical response at each test point of the model was calculated and plotted against wind tunnel speed. A curve fitted through the data was then extrapolated to zero to give the predicted flutter speed, prior to incrementing the wind tunnel speed. Open-loop flutter testing was conducted on the model in a baseline, forward and aft cg position configuration.

Closed-loop Testing

Initial closed-loop testing involved activating the control algorithm at the experimentally determined open-loop flutter speed of the model in its baseline, forward and aft cg position configurations and demonstrating that flutter could not be induced either by wind tunnel turbulence or by giving the model an initial excitation. The model was then allowed to flutter at the open-loop flutter speed with the control system off. When the flutter cycle was established the controller was activated to demonstrate that flutter at this critical open-loop flutter speed could be suppressed. For the baseline cg configuration, in both the CWT and LSWT, the wind tunnel speed was gradually increased from the open-loop flutter speed, with the control system active. This was done to demonstrate the effectiveness of the control system in preventing flutter and extending the flutter boundary of the model.

1.5.3 Measured Data

To enable flutter predictions, certain physical properties of the model were required prior to wind tunnel testing. Wind tunnel flutter test data were measured to validate the mathematical model of the system and demonstrate the effectiveness of the feedback controller.

Preliminary Model Data

Prior to wind tunnel testing of the active flutter control model, various model parameters were measured or estimated for input to the mathematical model of the aeroservoelastic system. This was to enable flutter prediction calculations and the design of an effective flutter suppression controller. These data were:

- The masses of the wing and moving parts of the flexure mount, measured with an electronic scale
- The mass inertia and cg position of the wing and moving part of the flexure mount, estimated from the parametric three-dimensional (3D) computer aided design (CAD) model using the measured component masses
- The mass inertia and cg position of the control surface, estimated from the parametric 3D CAD model using the measured component masses
- The stiffness and structural damping of the pitch and plunge vibration modes of the wing/flexure mount, estimated from the power spectral densities (PSDs) of the wind-off response of the system as mounted in the wind tunnel

- The transfer function of the control surface and actuation mechanism obtained from the measured response of the control surface to a swept-sine input

Wind Tunnel Data

Wind tunnel flutter test data were measured to validate the mathematical model of the system and demonstrate the effectiveness of the feedback controller. During open-loop wind tunnel testing the pitch and plunge displacements of the model were measured using calibrated strain gauge bridges located on the flexure mount. The frequency and damping of the pitch and plunge modes was determined from these data as described in Section 1.5.4. For closed-loop testing, in addition to the pitch angle and plunge displacement of the model, the control surface angle was measured with a calibrated Hall-effect rotary sensor. Since a full state feedback controller was designed, the remaining unmeasured model states (plunge rate, pitch rate, control surface rotation rate and two aerodynamic lag states) were estimated in real-time using the measured states and a Kalman filter implemented in the controller software.

1.5.4 Data Analysis and Application

Data from the aeroservoelastic wind tunnel model's transducers were acquired by a National Instruments™ NI USB-6211 multi-function DAQ module and custom developed C# software. The measured pitch and plunge time series data were transformed to the frequency domain and the PSD of each was calculated, using the MATLAB™ routine listed in Appendix C. Subcritical modal frequencies and dampings obtained from the PSDs of the time series data were used to predict the flutter onset speed to ensure safe testing of the wind tunnel model. During closed-loop testing, signals from the strain gauge bridges and control surface angle sensor were input in real-time to a Kalman filter with a fixed gain to estimate the remaining unmeasured system states and provide full state vector information. The full state vector was then multiplied by a fixed linear quadratic regulator (LQR) feedback gain to provide the control surface angle required to stabilise the model.

1.6 Outcomes

This research resulted in:

- The design and manufacture of a cantilevered pitch-plunge flexure/rigid wing model system that exhibits gentle flutter at low speed, allows easy adjustment of model mass, mass inertia, cg position and modal frequencies, and is easy to integrate into a wind tunnel
- A detailed understanding and appraisal of various control laws suitable for implementation to suppress flutter
- Design and physical implementation (in terms of both hardware and software) of the selected constant LQG constant control law

- A comprehensive set of wind tunnel flutter test results for different model configurations, showing good agreement between the measured flutter speed, the flutter speed calculated using two-dimensional unsteady aerodynamic theory and that predicted from the subcritical response of the model using real time methods during testing
- A comprehensive set of wind tunnel test results for different model configurations demonstrating active flutter control to a speed well in excess of the open-loop flutter speed, thus proving the effectiveness and robustness of the constant LQG controller in spite of mechanical, electrical, computational and design constraints

1.7 Dissertation Overview

Chapter 2 begins with a brief overview of two active flutter suppression flight test programs on full-scale aircraft to illustrate the benefits of such technology and the importance of binary wind tunnel flutter models in active flutter suppression research. Thereafter, a literature review of existing pitch-plunge wind tunnel flutter models and prominent active flutter suppression research conducted using them is presented. The chapter concludes with an overview of accepted flight flutter testing techniques to ensure safe testing of the wind tunnel model. The development of a full aeroservoelastic mathematical model in a state-space representation is given in Chapter 3, along with the design of a LQG feedback controller. Chapter 4 is devoted to the mechanical, electrical and software design of the aeroservoelastic wind tunnel model. The experimental procedures followed in this research are given in Chapter 5, in which the open- and closed loop wind tunnel testing of the active flutter control wind tunnel model is described. The CSIR CWT and LSWT in which testing was conducted, are also described in Chapter 5. Representative preliminary, open- and closed-loop wind tunnel test results are presented and discussed in Chapter 6. Finally, conclusions from the research are drawn and recommendations for future work are made in Chapter 7. Additional details and relevant software listings are given in the appendices.

Chapter 2

Literature Review

A brief overview of two active flutter suppression flight test programs is given in Section 2.1 to show the benefits of active flutter suppression controllers and to provide the context for this research. A review of existing pitch-plunge binary flutter wind tunnel models follows in Section 2.2, as these models are typically used to demonstrate flutter suppression and investigate different active flutter suppression control laws for full-scale aircraft. A literature survey of active flutter control applied to two-dimensional, binary pitch-plunge flutter models is then presented in Section 2.3. In this section, the mathematical modelling techniques used, the control law designs and wind tunnel test results (where applicable) of these aeroservoelastic systems are briefly discussed. Finally, techniques to estimate the flutter speed based on the system's sub-critical response during flight (wind tunnel) flutter testing, used to ensure safe testing of the model, are presented in Section 2.4. The mathematical formulation of a full three degree-of-freedom binary aeroservoelastic model is not discussed here because of its complexity, but rather outlined separately in Chapter 3.

2.1 Active Flutter Suppression Flight Test Programs

To demonstrate the importance and effectiveness of active flutter suppression control systems, examples of two active flutter suppression flight test programs are discussed in this section. Typically flutter suppression control systems are applied to high performance military jet aircraft that fly close to the flutter boundary and are thus more likely to experience flutter. These aircraft usually have existing full command and stability augmentation systems with fast responding control surface actuators, making the implementation of flutter suppression systems on them relatively easy (Haidl *et al.*, 1981). The flutter suppression flight test programs outlined below are the B-52 flutter suppression program and the F-4F flutter suppression program.

2.1.1 B-52 Flutter Suppression Program

The Wichita Division of the Boeing aircraft company conducted a study on a control configured vehicle (CCV) B-52 aircraft. The purpose of the study was to demonstrate the benefits of applying flight control technology to a large flexible aircraft. Full details can be found in Johannes and Thompson (1973). Of interest to this research is the flutter mode control (FMC) system of the CCV program that used aileron and flaperon control surfaces to stabilise symmetric wing flutter (Roger *et al.*, 1975). The primary function of the FMC system was to analytically extend the flutter boundary of the NB-52E (Figure 2.1) aircraft by 30.0% and thereafter demonstrate stable flight 10 knots above the uncontrolled flutter speed of the aircraft. To achieve a flutter speed within the operating envelope of the aircraft, the aircraft port and starboard drop fuel tanks were modified to carry lead in their forward sections. In this configuration, it was predicted using the doublet lattice method (DLM) that flutter would occur at 315.0*KCAS* at an altitude of 21 000*ft*, which was below the normal level flight limit of 400 knots. The flutter mode was gentle (2.4*Hz*.; 0.01*g* per 10.0*KIAS*) and comprised a second wing bending, first wing torsion, in-plane wing bending, outboard engine strut bending, fuselage nose bending and vertical tail bending coupled structural mode. Mild flutter characteristics where important because it made demonstration of the FMC 10 knots above the flutter speed of the aircraft feasible, and permitted a linear and low-cost controller design.

Figure 2.1 Air Force Flight Dynamics Laboratory NB-52E CCV Aircraft (U.S. Air Force)

Various synthesis constraints (such as sensor locations and turbulence limits) were imposed on the FMC control system to ensure adequate performance would be provided despite variations between the actual system and those predicted by the nominal mathematical model of the system. Nonlinearities such as backlash in the actuator attachments, servo valve threshold and control surface limit and rate saturation were accounted for and modelled in the control system. The control system design comprised two independent control loops that were both capable of augmenting structural damping to achieve performance objectives. These loops were an outboard aileron and an outboard flaperon control loop. Signals from a wing vertical acceleration sensor placed near the external flutter (fuel) tanks were used to drive the outboard ailerons whilst signals from a sensor measuring the wing vertical acceleration between the engine pods was used to drive the outboard flaperon. Redundancy was built into the FMC module to ensure flight safety. This was done by using independent sensors, electronics and secondary power for each control loop. Additional safety features included the ability to jettison the tanks and telemetry to provide real-time monitoring of the system. Given the mild flutter characteristics and low flutter frequency of the B-52 aircraft, flutter could also be arrested by the pilot by closing the throttle and applying the airbrakes. Control of CCV equipment was accomplished through panels at the pilot and flight engineer stations, where both the pilot and flight engineer needed to engage a system to make it operative. Before commencing flight tests the CCV equipment was tested extensively on the ground by determining the hysteresis, step response and frequency response of both the ailerons and flaperons. Flight testing commenced by establishing the baseline (FMC-off) aircraft flutter mode where it was shown that the actual flutter speed was 7.0% higher than predicted. With the FMC system in operation, the objective of flying the aircraft 10.0 knots past the demonstrated flutter speed was met and exceeded for two different flight test configurations. At 12 knots above the FMC-off flutter speed, large pilot inputs could not induce flutter with the FMC system operative (Roger *et al.*, 1975).

2.1.2 F-4F Flutter Suppression Program

Based on the success of previous research, a flutter suppression research program on a German Air Force Test Center F-4F aircraft was jointly initiated by Bundesamt für Wehrtechnik und Beschaffung and the U.S. Air Force Flight Dynamic Laboratory (Haidl *et al.*, 1981). A study based on store mass and mass inertia was conducted to determine a store configuration that would cause flutter within the flight envelope of the aircraft, to be used to demonstrate a flutter suppression system (FSS). The calculated modal properties of the aircraft with the chosen store (flutter tank) configuration were validated by ground resonance tests. A FSS control box was fitted to the aircraft and all other flight test instrumentation already existed on the aircraft. Four sensors were fitted to each wing to measure the first wing bending and first wing torsion/store pitch aircraft modes. Initially a damping controller was designed to introduce damping into the critical flutter mode whilst keeping all other dynamic characteristics unchanged. A frequency controller was also investigated to change the aerodynamic stiffness of the flutter mode. Both controllers were investigated around a design speed of 600 knots. An approximate transfer function of the whole FSS, which included actuator dynamics and the control electronics band pass filter, was introduced into the mathematical model of the aircraft. The FSS output was fed into the roll channel of the aircraft stability and augmentation system to actuate both the ailerons and spoilers. Redundancy was built into the FSS to ensure safe flight testing beyond the supercritical flutter speed of the aircraft. Each wing had an independent FSS that could be used to suppress flutter. In addition, each store had trim weights fitted that could be released within 0.5s (three flutter cycles) to change store mass inertia to result in a flutter free aircraft configuration. The stores could also be jettisoned by the pilot in extreme cases. Initial flight testing began with the FSS inoperative to determine the flutter speeds of the safe and critical store configurations. Open-loop tests were then performed to substantiate and optimise the flutter control law design before closed-loop testing to demonstrate flutter mode suppression. The open-loop flight flutter tests were conducted using aileron excitation to determine the damping and frequencies of the aircraft modes. Subcritical flight test results predicted that flutter would occur at 600$KIAS$ with the stores in their critical configuration. In comparison, 4.0% structural damping was observed at

600*KIAS* with the stores in their safe configuration (trim weights released). Subcritical closed-loop tests were conducted to demonstrate elastic mode suppression were it was shown that the FSS was able to considerably increase the damping of critical modes at high subcritical speeds. Extrapolated damping curves indicated an anticipated flutter speed of 700*KIAS* with the FSS operative.

Figure 2.2 McDonnell Douglas F-4F Phantom Aircraft (www.airliners.net)

2.2 Binary Wind Tunnel Flutter Models

Although wind tunnel testing of full three-dimensional aeroservoelastic wind tunnel models has been conducted successfully (Borglund and Kuttenkeuler (2002), Peloubet *et al.* (1981), Perry and Cole (1995) for example), aeroservoelastic research is typically conducted on two-dimensional binary flutter models that exhibit classical pitch-plunge flutter. These models are safer to test and simpler to model mathematically, allowing them to act as test-beds for active flutter suppression control laws that will ultimately be applied to full-scale aircraft.

Two common pitch-plunge flutter model design approaches are reported in the literature. One approach employs flexures such as shown in Figure 2.3. The second uses coil springs, for example as shown in Figure 2.4 and Figure 2.5, to achieve the desired pitch and plunge modal frequencies and motion. Farmer (1982) describes the design of a large pitch-plunge flutter model in which a rigid wing is attached to a splitter plate mounted on four circular rods and a central horizontal rectangular drag strut, all cantilevered from the side wall of the wind tunnel. The function of the drag strut is to make the in-plane stiffness of the mount significantly higher than its transverse stiffness, whilst contributing minimally to its torsional stiffness. The rods are fixed and constrained by plates at either end to ensure that the end deflection slopes remain zero as the splitter plate moves, resulting in simple pitch-plunge movement. An advantage of flexure constrained flutter models is that structural damping is very low (i.e. it can be considered negligible) and remains constant as loads on the model vary. Aerodynamic damping is therefore essentially the only damping influencing the motion of the wing. This enables investigation of changes in aerodynamic damping with air flow over the wing (Farmer, 1982). Dansberry *et al.* (1993) improved Farmer's mount (1982) by reducing the size of the splitter plate and using it only as a means to mount the flexures to the wing. A second splitter plate, unconnected to the model, was used to separate the airflow over the model and the mount.

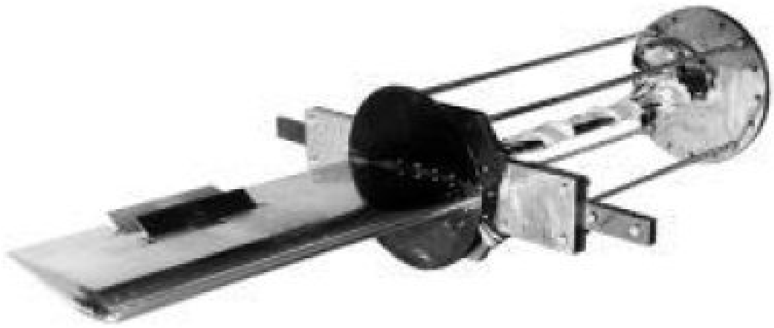


Figure 2.3 Pitch-Plunge Cantilevered Flexure Flutter Model (Waszak, 1997)

The alternative approach to the design of a pitch-plunge flutter model, using an arrangement of coil springs to constrain the motion of a rigid wing, is described in O'Neil and Strganac (1998) and Chowdhury and Sarkar (2003). This system allows pitch and plunge stiffnesses to be varied *independently* by changing the arrangement of the springs. These systems can also be used to investigate the effect of a nonlinear restoring force on the flutter characteristics of a rigid wing (O'Neil and Strganac, 1998).

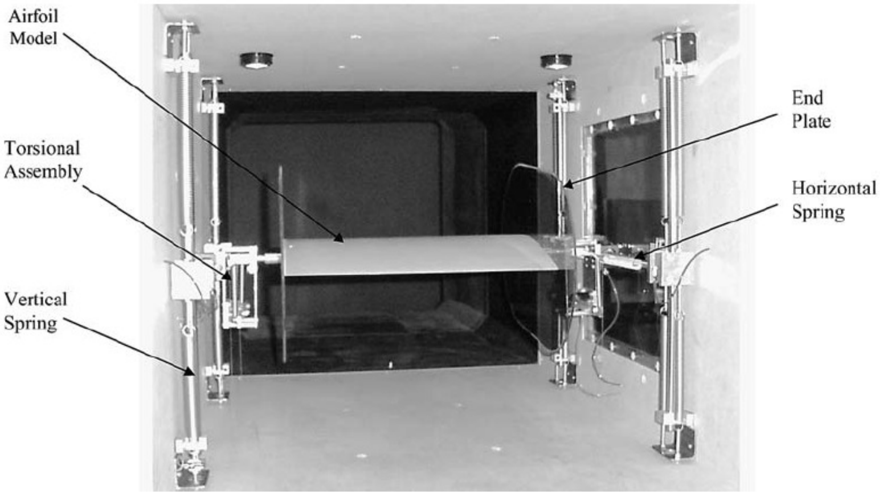


Figure 2.4 Typical Spring Mount Rigid Wing Flutter Model (Chowdhury and Sarkar, 2003)

The main disadvantages of the spring systems are their complexity and relatively high cost, and the significant structural damping they introduce into the flutter model. An important feature in the design of a rigid wing flutter model is the inclusion of mechanical stops to limit the models deflection should flutter at a super-critical airspeed be encountered (Cole, 1986). The amount of movement allowed must be set by an evaluation of the strength of the flexure and the requirement to keep the deflections within its elastic range. The systems mentioned above have flutter characteristics which are qualitatively similar to those of more complex three-dimensional elastic wing models with bending-torsion deflections. Consequently, these systems are very useful for many flutter research studies because they adequately represent parameters affecting flutter (Farmer, 1982). No other different type of model/mount system for pitch-plunge motion could be found in the literature.

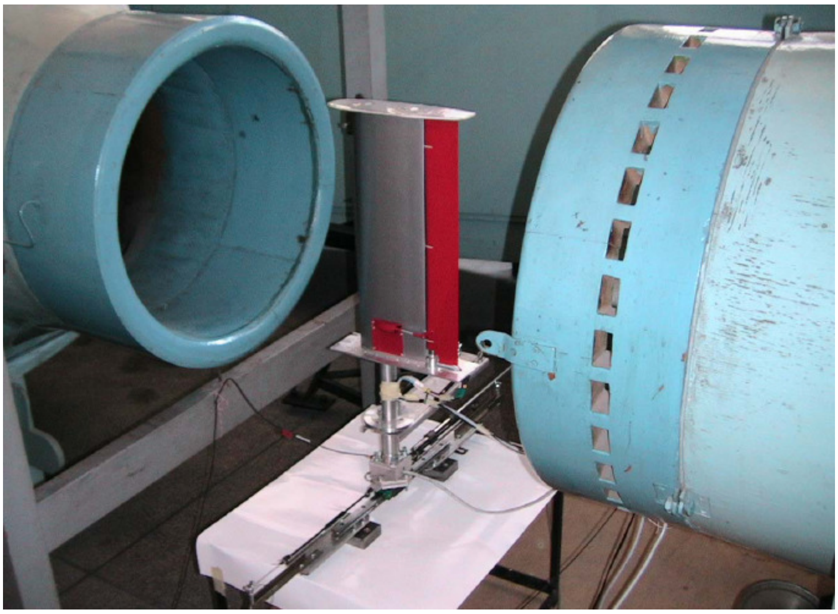


Figure 2.5 Cantilever Spring Mount Rigid Wing Flutter Model (Predoiu *et al.*, 2007)

2.3 Binary Flutter Suppression Research

The literature reviewed and presented here highlights two-dimensional (2D) binary pitch-plunge flutter suppression research. The control laws designed for these models are discussed and where applicable, wind tunnel results are quoted. In the absence of wind tunnel test results, the results from numerical simulations are quoted.

Horikawa and Dowell (1979) discuss the application of a parameter design technique to active flutter control of a wing section with a trailing edge control surface using a root locus method. In their research, it was assumed that the inertia and damping effects of the control surface do not affect the dynamics of the primary system and were thus omitted from the mathematical model of the system. Static aerodynamic theory was used (in that the lift and moment were assumed proportional to geometric angle of attack of the wing or control surface) so unsteady effects were neglected. Four feedback control rules to activate the control surface are briefly outlined, these being a bending displacement feedback, a bending acceleration feedback, a torsion displacement feedback and torsion acceleration feedback control law. A parametric stability analysis of the centre of mass, centre of pressure and elastic axis position is briefly outlined for each of the feedback control cases. As a numerical example, the authors calculated the stability boundaries of the system for each feedback control rule at various elastic axis positions. They concluded by stating that a bending acceleration feedback rule can give rise to a ceiling on the flutter boundary, even if the feedback gain is increased to infinity. A torsion feedback rule was seen to result in a simpler stability boundary.

Karpel (1982) describes an analytical design technique for active flutter suppression and gust load alleviation using a state-space aeroelastic model. To achieve matrix equations with constant coefficients for the state-space model, a rational approximation of unsteady aerodynamics loads in the Laplace domain was made. The focus of the research was on optimising a partial feedback control law over a wide range of aerodynamic parameters (e.g. gust response) that can suppress flutter over the entire intended flight envelope using a pole assignment technique. Karpel (1982) suggested that the control technique is dependant on the accuracy of the mathematical model of the system, the available control means, the measurement accuracy of sensors and the capacity of the computer used. A numerical simulation of a typical wing section was performed to show how the flutter boundary can be extended using active control.

Heeg (1993) used a novel method to actively suppress pitch-plunge flutter. Her model consisted of a rigid wing mounted to spring tines that permitted translational and rotational degrees-of-freedom. Heeg's model is the smallest model that could be found described in the open literature, having a chord length of 4.25" (108*mm*) and wingspan of 4.00" (102*mm*). The structural dynamics of the model were calculated using a finite element method and laminated plate theory, and a DLM, with rational function approximations, was used to characterise the unsteady aerodynamics forces. The structural and unsteady aerodynamic models were then combined in a state-space form to complete the full aeroservoelastic mathematical model. An important design feature of the wind tunnel model was that mechanical stops were incorporated to limit the amplitude of the pitch and plunge oscillations above the critical flutter speed, and thus prevent destruction of the model. A SGB and an accelerometer were used to measure model parameters. Flutter suppression was achieved by the use of piezoelectric actuators attached to the bending (plunge) leaf spring. When energised these actuators altered the stiffness and damping characteristics of the model mount. Due to the simplicity of the model and the only design requirement being that the system be stable, a straightforward single-input, single-output (SISO) controller was designed and implemented. Open-loop flutter of the model was encountered at 580.0*ft*/*s* (177.0*m*/*s*), whilst closed-loop flutter was encountered at 697.0*ft*/*s* (212.0*m*/*s*), implying a maximum increase in the flutter boundary of 20.2%. Above 697*ft*/*s* (212.0*m*/*s*) there was insufficient control authority to suppress flutter.

Block and Gilliatt (1997) describe a rigid-wing, flexible-mount system for active flutter control research that can be used to provide either linear or nonlinear restoring forces. This was achieved by designing a spring-cam system, to which a rigid wing was mounted vertically in the wind tunnel, with the spring-cam mechanism outside the wind tunnel, away from the airflow. The full span control surface used to suppress flutter was 20.0% of the full chord length. Mechanical stops were incorporated in the model to limit the plunge deflection of the wing to ±40*mm* and the pitch deflection to ±28.0˚. Optical encoders placed on the pitch and plunge cams were used to measure pitch and plunge deflections respectively. A Futaba™ FPS-134 model aircraft servo was used to actuate the control surface because of its high torque and small size, allowing it to be embedded in the wing model. A two degree-of-freedom linear structural mathematical model of the system that included viscous structural damping was developed. Under the assumption that the damped free vibration of the system was logarithmic, the log-decrement method was used to calculate the damping ratio for each degree-of-freedom. The unsteady lift and pitching moment were modelled using Theodorsen's theory (Theodorsen, 1935) whilst it was assumed that control surface dynamics did not affect the primary (two degree-of-freedom) system and was thus modelled as a second order transfer function. Feedback gains were obtained using LQR theory, and system states that could not be measured were obtained using a state observer based upon the Kalman estimator. Process noise was estimated to be proportional to the single value decomposition of the eigenvalues of the system. Measurement noise was calculated from the squares of the smallest possible measurements of the plunge deflection and pitch angle of the model, determined from the resolution of the optical encoders used. Feedback gains were chosen to make the dynamics of the state observer three to four times faster than the dynamics of the original system to correct for any error between the initial conditions of the observer and actual system states. With the model in its linear restoring force configuration, open-loop flutter was encountered at 15.5*m*/*s* at a frequency of 2.1*Hz*. Closed-loop flutter testing, using a controller derived at 25% above the open-loop flutter speed of the model, demonstrated that the control system was able to suppress developed flutter cycles within 3*s* with maximum control surface deflections of ±10.0°. Tests were performed up to a wind tunnel speed of 31.4*m*/*s* after which they were stopped to avoid exceeding the limitations of the servo driving the control surface. This translated to an increase in the flutter speed of the model of 102.6%. The open-loop limit cycle oscillation[2] (LCO) behaviour of the model was then investigated. These results are not relevant to this work and are therefore not discussed here.

Waszak (1997) describes how robust multivariable control design methods (H^∞ and μ-synthesis) were used to demonstrate flutter suppression of the Benchmark Active Controls Technology

---

[2] Limit cycle oscillations occur when a *nonlinear* restoring force is applied to the wing. Instead of the amplitude of the oscillation increasing, as in the case of flutter, the amplitude of a LCO remains constant.

(BACT) model at the National Aeronautics and Space Administration (NASA) Langley research centre. The BACT project focused on active flutter control in the transonic flow regime. The main objectives of the research were to maintain stability over the entire operating range (robust stability), reduce pitch and plunge accelerations (robust performance) at all operating conditions, minimise control surface deflections and their respective rates and use a single, low order, controller to simplify implementation and reduce risk. The model used consisted of a rigid wing (NACA0012 section) and flexible mount system that permitted pitch and plunge degrees of freedom. The wing was fitted with hydraulically actuated upper and lower spoilers and a trailing edge control surface to achieve control. The BACT wing was instrumented with pressure transducers, accelerometers, control surface position sensors and hydraulic pressure transducers. The flexible mount was instrumented with strain gauges for measurement of pitch and plunge deflections. It was isolated from the rigid wing with a splitter plate, and from the airflow with a fairing between the splitter plate and wind tunnel wall. A mathematical model of the system that included structural dynamics, unsteady aerodynamics and actuator dynamics was developed. The unsteady aerodynamic forces were characterised using an aerodynamic coefficient based method, deemed suitable because of the low reduced frequency of the model. A simple model of wind tunnel turbulence, based on experimental data, was also developed. Both of these models were combined into a single state-space model that had ten states, parameterised by dynamic pressure. The control law design was performed in two stages. The first stage covered the design and testing of SISO controllers to assess the effectiveness of the spoilers in controlling flutter and to provide a benchmark for evaluating the benefits of multi-input, multi-output (MIMO) control. The second stage of the research involved the design of MIMO controllers to demonstrate the potential for enhanced performance and robustness that can be achieved by using robust multivariable design methods. The design strategy for both the SISO and MIMO controllers was to use a single fixed rather than a scheduled control law to suppress flutter over the anticipated range of wind tunnel operating conditions. This was to simplify controller implementation and data analysis, and is representative of how active flutter suppression techniques can be applied to real aircraft. All control laws designed and tested were successful in suppressing flutter above the open-loop critical flutter speed. The H^{∞} and μ-synthesis robust multivariable design methods were shown to have better stability and performance robustness than traditional SISO designs. It was further shown that a constant control law was able to maintain closed loop stability over a wide range (open-loop stable and unstable) of wind tunnel operating conditions.

Waszak (1998) gives the detailed formulation of the mathematical model of the dynamic behaviour of the BACT wind tunnel model described by Waszak (1997). The equations of motion of the BACT model were combined with models of the actuators and a model of wind tunnel turbulence, to give a complete model of the system. The trailing edge control surface was assumed to have a very large stiffness such that deformation due to hinge load is negligible. Under this assumption, a system having only pitch and plunge degrees-of-freedom i.e. a two degree-of-freedom system was modelled. Structural damping and inertial coupling between the wing structure and control surfaces were included in the equations of motion. Waszak (1998) used a simple coefficient-based aerodynamic model rather than the theory of Theodorsen (1935) to model unsteady aerodynamics, because the BACT model operated in the transonic flow regime where Theodorsen's theory (1935) is not applicable and the reduced frequency of the model was low. Numerical values based on experimentation and computational aerodynamics were substituted into the final equations of motion of the BACT model to form the numerical model of the system. To validate the theory, the BACT numerical model was compared to actual BACT wind tunnel data. In general, the BACT numerical model was in good agreement with BACT experimental data in terms of both static and dynamic properties.

Vipperman *et al.* (1998) used an H_2 optimal control scheme to actively suppress flutter of a rigid wing (NACA0012 profile) with a trailing edge control surface. The control surface span was the same as that of the wing, its chord was 33.0% of the wing chord and it was driven by a linear actuator. The wing was mounted to guided cantilever beams to provide the plunge stiffness and a wire spring mounted at the elastic axis of the wing provided the pitch stiffness. The goal of the research was to create a compensator that would provide gust alleviation at low velocities but still extend the flutter boundary above the critical open-loop flutter boundary. The control system design was based on experimental system identification of the transfer function of the measured variables (plunge deflection, pitch angle and control surface angle in this case) and an output control signal to drive the control surface at each test velocity. The control surface angle was measured by using a linear variable displacement transducer to measure the displacement of the drive actuator. The pitch and plunge displacements were measured using rotational variable displacement transducers. As the poles and zeros of the dynamic system were dependant on the velocity of the airflow over the wing, only limited robustness was achieved. Even so, it was demonstrated that the flutter boundary of the system could be extended by 12.4% and that the controller stabilised the system at speeds within ±10.0% of the controller design speed.

Block and Strganac (1998) further describe the work of Block and Gilliatt (1997) in which active control of a rigid wing with linear and nonlinear pitch and plunge restoring forces was investigated. The authors discuss inherent nonlinearities that have been assumed negligible in most other research into active pitch-plunge flutter control. These nonlinearities in the system include controller saturation, free-play nonlinearities when there is backlash in control linkages and hysteresis that occurs when friction loads affect linkage dynamics. They go on to describe how stiffness nonlinearities can also affect the dynamics of a system. They made use of Wagner and Jones' (1945) approximation to Theodorsen's (1935) function to model unsteady aerodynamics, and assumed that the control surface dynamics behaved as a second order system that did not affect the coupling of the plunge and pitch degrees-of-freedom. The dynamics of the control surface actuator were neglected since it was assumed that the actuator reacted exactly as specified as long as it operated below its maximum rate. The final system model developed consisted of twelve states, of which only the pitch and plunge displacement were measured. A state observer (Kalman filter) was used to estimate the ten remaining system states. The system was stabilised, and the feedback gain optimised, using a LQR controller. All control laws implemented were derived at 25.0% above the flutter velocity. Three types of experiments were conducted. The first allowed control and state estimation to begin while the system response was growing, the second allowed control and state estimation to commence once the structure was released from rest and the third was to implement control and state estimation before the structure was released from rest or excited. The system was most stable when control started before the structure was released from rest. The controller was only effective when implemented while the response was growing if the motion of the structure had not caused the wing to stall. It was found that if the controller was implemented once the structure was released from rest, less control surface motion was required to stabilise it. Block and Strganac (1998) also highlighted that a linear controller was ineffective in controlling LCOs at speeds higher that the flutter speed of the model. Strganac *et al*. (2000) addressed this problem by designing a nonlinear controller for the model, but this is not discussed here as the focus of this research is on flutter suppression, not suppression of LCOs.

Lau and Krener (1999) used a standard linear model with three degrees-of-freedom (pitch, plunge and control surface angle) to investigate active flutter control. They assumed incompressible and irrotational flow for modelling of unsteady aerodynamics and made use of Theodorsen's function characterised by Jones' approximation, as described by Küssner and Schwarz (1941). They developed a linear model of the system that was realised by an eight dimensional state-space model that included two aerodynamic lag states. Having developed a control surface hinge moment LQG controller and highlighted problems associated with LQR control, they discuss the benefits of using linear parameter varying (LPV) control. Lau and Krener (1999) did not actually implement the LPV control on a wind tunnel model, but only outlined its development.

Barker *et al.* (1999) used a gain scheduled controller to actively suppress flutter of the BACT wind tunnel model using linear fractional control. The primary objectives of their research were to improve disturbance rejection characteristics and increase the flutter boundary of the wing. To facilitate linear fractional control, linear time invariant models of the BACT wind tunnel model were developed for various wind tunnel operating conditions, using the DLM to calculate three-dimensional aerodynamic forces on the wing. The final controller was successful in suppressing flutter and stabilising the wing over a wide range of operating conditions, and increased the flutter boundary of the system in excess of 50%.

Norlander *et al.* (2000) used a rigid rectangular wing with a NACA64A010 profile attached to a mount that provided two degrees-of-freedom for active flutter suppression research. Unlike other rigid wing flutter models, their model was given a root bending (plunge) degree-of-freedom together with the pitch (torsion) degree of freedom, which were both tailored using torsional springs. Three accelerometers were mounted in the wing to measure modal frequencies and three laser triangulation sensors were mounted in the roof of the wind tunnel to measure the displacement of the wing. The trailing edge control surface was actuated hydraulically and its position measured using a potentiometer. The controller design criteria were to suppress flutter with the least possible effort and to be robust to changes in wind tunnel conditions. Initially, a LQG controller was used and shown to be successful in suppressing flutter. The controller operated with low gain over the frequency range tested, implying that only small control surface movements were required to suppress flutter. They then investigated the design of a less complex controller to have the same performance as the tested LQG controller. To achieve this, a parametric linear quadratic method was used to design a low order linear quadratic controller. The parametric linear quadratic controller they implemented had nearly equal performance to the LQR controller, but required far less computer power. They also showed that although both controllers were designed for one wind tunnel operating point, both were robust to changes in wind tunnel operating conditions. Although flutter boundary extension tests were not conducted, flutter could not be induced at 2.5% above the demonstrated open-loop flutter speed of the model.

Motivated by the limited success achieved by Block and Gilliatt (1997), Block and Strganac (1998) and Strganac *et al*. (2000) in using a trailing edge control surface to suppress LCOs, Platanitis and Strganac (2004) investigated the use of both a leading and trailing edge control surfaces for LCO and flutter suppression. They assumed that although the model had four degrees-of-freedom, the high frequency dynamics of both the leading and trailing edge control surfaces were far removed from the primary system. This allowed them to construct a new two degree-of-freedom (pitch and plunge motion) model to be used on the mount described by Block and Gilliatt (1997). The leading edge control surface was 15.0% of the full wing chord whilst the trailing edge control surface was 20.0% of the full wing chord. Two Futaba™ S9402 model aircraft servos were used to actuate these control surfaces. The unsteady lift and pitching moment of the wing were modelled using quasi-steady aerodynamic theory and aerodynamic coefficients proven by Ko *et al*. (1997) through wind tunnel testing to be adequate for low reduced frequency, subsonic flow. Adaptive control and feedback linearisation techniques were used to design an effective flutter suppression controller. Wind tunnel tests showed that the controller worked only for a limited range of free stream velocities, and that unmodelled nonlinear aerodynamic and unsteady effects became problematic at higher test velocities. The authors concluded by saying that control effectiveness was limited by hardware design at higher test velocities because of larger aerodynamic loads acting on the control surfaces, an increase in sensitivity to nonlinear and unsteady aerodynamic effects and control surface dynamics.

Bhoir and Singh (2004) investigated active control of a two-dimensional wing with structural nonlinearities. They made use of Theodorsen's (1935) theory, appropriately approximated by Jones' (1945) approximation, to model the unsteady aerodynamics. They developed a model of the system containing eight states. The LQR controller they designed used three measured system states (plunge deflection, pitch angle and control surface angle) and a state observer to estimate the remaining unmeasured system states. The controller was shown to be effective at suppressing flutter only through numerical simulations. Wind tunnel tests on a flutter model were not conducted.

De Marqui *et al.* (2005) conducted active flutter suppression research on a pitch-plunge flutter model with the purpose of using a control scheme to suppress flutter and maintain stability of the closed-loop system over a wide range of test velocities. They developed a two degree-of-freedom (pitch, plunge) mathematical model of the system using a state-space technique, to facilitate the design of a state feedback controller. Unsteady aerodynamic forces were modelled using an aerodynamic coefficient method and wake effects were ignored because of the low reduced frequency of the model. Neither the control surface nor control surface actuator dynamics were modelled in the state-space system. A pole assignment with state feedback control scheme was implemented to suppress flutter. A rigid wing with a NACA0012 profile fixed to a flexible mount system based on Farmer's (1982) concept was designed for wind tunnel testing. A flexure mount was used because it was considered a simple and safe way of achieving flutter. Model instrumentation comprised three accelerometers, two SGBs and an encoder. One accelerometer was mounted in the centre of the moving plate (i.e. along the elastic axis of the wing) to measure plunge accelerations and the other two near the leading and trailing edges of the wing to measure pitch accelerations. The SGBs were used to measure pitch and plunge deflections. A low-pass Butterworth filter with a cut-off frequency of 5.0*Hz.* was applied to both the accelerometer and SGB signals to avoid high frequency noise. A brushless DC electrical motor was used to drive the trailing edge control surface and an encoder was used to measure the control surface angle. A dSPACE™ DS 1103 processor board was used to simultaneously acquire data and apply feedback control to the wind tunnel model. Initial wind tunnel tests focused on determining the open-loop (uncontrolled) system dynamics, more details of which are given in De Marqui *et al.* (2006). Flutter was encountered at 23.0*m/s* at a frequency of 1.6*Hz*. Once the instability of the uncontrolled system was established, new sets of stable poles were assigned and a feedback control matrix calculated. Since the calculated feedback control matrix was not unique, one calculated to give the best system performance was then implemented experimentally. Closed-loop wind tunnel tests demonstrated that flutter could be suppressed up to 10% above the open-loop critical flutter speed of the system. Beyond this, divergence of the model that the control system was unable to prevent, was encountered. The authors concluded by emphasising the importance of controller simulations to verify the response characteristics of the system before experimental implementation because of the destructive oscillations characteristic of flutter.

da Silva and Júnior (2006) investigated the use of Linear Matrix Inequalities (LMIs) as a flutter suppression control law. The aim of their research was to design a LMI control methodology that could be robust to parametric uncertainties (such as pitch stiffness) in a two degree-of-freedom model described by De Marqui *et al.* (2005). They assumed that plunge deflection, pitch angle and control surface angle and their respective rates were measurable, whilst other system states could be obtained from a state estimator. The regulator (controller and state estimator) they designed was shown via numerical simulations to successfully suppress flutter above the predicted critical open-loop flutter speed and to be robust to parametric uncertainties. Actual wind tunnel tests were not performed.

Marretta and Marino (2007) describe the design process of an active flutter suppression controller with two feedback control loops based on classical control system design techniques for use on the NASA BACT wing (Waszak, 1997). The authors highlighted that typical flutter suppression controllers have three functions, the first is to sense the flutter mode, the second to compensate the feedback signal and the third to apply the calculated control force to the aircraft. A linearised three degree-of-freedom (pitch angle, plunge displacement and control surface rotation) mathematical model was developed. Unsteady aerodynamic forces were modelled using Theodorsen's theory (1935) with a rational function approximation used to account for aerodynamic hysteresis. The BACT model control surface actuator dynamics were modelled by a second order oscillator derived from experimental data generated by Waszak and Fung (1996). The second order oscillator representing the control surface actuator dynamics was then substituted into the control surface equation of motion including unsteady aerodynamic forces. Although the full control surface dynamics were modelled, Marretta and Marino (2007) noted that because the control surface stiffness was much greater than the control surface actuator stiffness, the control surface dynamics were primarily governed by the control surface actuator. A full twelve state state-space model of the system incorporating two aerodynamic lag states per degree-of-freedom was developed using Jones' approximation (1945) to model Theodorsen's function (1935). A flutter suppression feedback controller using classical theory based on a Nyquist stability criterion was designed. A linear combination of the BACT model pitch and plunge accelerations was used as the output variable and the control surface angle as the input variable in the control system design. A computational delay of 1/200*s* and an actuator dead-band of 0.02° were modelled in the control system with MathWorks™ Simulink™. Although wind tunnel tests of the controller were not performed, the authors concluded that a dual control-loop flutter suppression controller has better performance and greater robustness than classical SISO flutter suppression controllers.

McEver *et al*. (2007) described active flutter control and closed-loop identification of a binary pitch-plunge flutter model. The authors argued that because the dynamics of aeroelastic systems change dramatically with airspeed, fixed parameter controllers may only stabilise aircraft flutter modes over a small airspeed range within the full flight envelope of the aircraft, thus multiple controllers may be required to fully suppress flutter over the entire flight regime. They state that the actual flutter dynamics of an aeroelastic system can be determined using system identification techniques, but highlight that because the system is open-loop unstable, these techniques must be applied to a closed-loop system using a nominal controller design. In view of this, they described a method of designing active flutter suppression controllers based on a *Q* parameterised aeroservoelastic model identified at supercritical airspeeds. Initially a nominal controller was designed to stabilise a wind tunnel model over a limited airspeed range above its open-loop flutter speed, to allow its dynamics to be identified. Subsequent controllers were then designed based on the identified unstable aeroservoelastic model over the specified airspeed range. Using this model identification/controller design technique, McEver *et al*. (2007) suggest that flutter can be suppressed over the full operating range of an aircraft if sufficient controller overlap is ensured and an adaptive control scheme that operates on airspeed is implemented. The wind tunnel model used by McEver *et al*. (2007) to demonstrate this concept comprised a NACA0015 section wing with an aluminium/steel/balsa wood internal structure covered with an aluminium skin. The wing was fixed to a mount that permitted pitch and plunge degrees-of-freedom. The wingspan of the model was 520*mm* and the chord was 360*mm*, giving it an aspect ratio of 1.44. A piezoelectric actuator mounted in the middle of the wing was used to drive a trailing edge control surface to maximum deflections of ±5.5° through a slider-crank lever mechanism. Rotary sensors mounted to the pitch axis and control surface pivot pins were used to measure these angles. The plunge displacement of the model was measured with a rotary sensor connected to a rod that rotated as the wing moved in its plunge degree-of-freedom. Control was implemented through a dSPACE™ digital signal processor board at a frequency of 500*Hz*. Open-loop wind tunnel tests showed the onset of flutter at 23.0*m/s* at a frequency of 4.3*Hz*. The frequency response of the aeroservoelastic system was estimated using a swept-sine control surface input of 2.0° from 1.0*Hz*. to 6.0*Hz*. at both subcritical and supercritical speeds. Frequency domain system identification software was used to fit continuous state-space models to the estimated frequency responses of the model. Having identified the aeroservoelastic model using a nominal controller, a new controller using an Evans root-locus technique (Evans, 1950) was designed using the pitch angle of the model as a measure of the aerofoil state. Pitch angle was chosen because of its high interaction with the airflow and because of the higher pitch/control surface coupling than the plunge /control surface coupling of the model. A SISO controller was developed because the coupling of the two aerofoil modes meant that only one degree-of-freedom had to be suppressed to stabilise the model. McEver *et al*. (2007) demonstrated that the original nominal controller was able to suppress flutter up to 30.0*m/s*, 30.4% above the open-loop flutter speed of the model. At higher wind tunnel speeds the model was more unstable and difficult to control, but the ability to accurately identify its dynamics enabled the redesign of a more suitable controller. For example, a controller designed using a mathematical model identified at 28.0*m/s* was able to increase the flutter boundary of the model by 52.2%, suppressing flutter up to an airspeed of 35.0*m/s*. These results illustrated the appeal of using a *Q* parameterisation technique to identify an aeroservoelastic system above its open-loop flutter speed to facilitate the design of an appropriate, effective flutter suppression controller.

Prime *et al.* (2009) discussed the LPV control of an improved three degree-of-freedom mathematical model to describe the wind tunnel model developed by Platanitis and Strganac

(2004). The authors argued that in the original formulation of the mathematical model of the system (Platanitis and Strganac, 2004), the inertia of model and dynamics of the control surfaces were not properly accounted for. This was apparent in the original wind tunnels tests in which the controller was only effective over a limited wind tunnel speed range. To improve the original model of Platanitis and Strganac (2004), Prime *et al.* (2009) modelled the dynamics of the trailing edge servo with a second order system. The aerodynamic model of the system remained unchanged. Using a H_2 representation of the standard LQR regulator, a state feedback controller was synthesised using LMIs as a generalised LPV control problem (Prime *et al.*, 2009). Wind tunnel tests of the system demonstrated its ability to suppress developed LCOs in approximately 1.0*s* at varying airspeeds. This was a significant improvement on what was achieved by Platanitis and Strganac (2004) and demonstrated the importance of accurate mathematical modelling

2.4 Wind Tunnel and Flight Flutter Testing Techniques

Wind tunnel and flight flutter testing is inherently dangerous and entails the risk of damage to models/aircraft because of violent oscillations that occur at flutter (Ruhlin *et al.*, 1983). A flutter mode can suddenly become unstable with only a small increase in airspeed (Bennett, 1982). Consequently, it is desirable and necessary to predict the onset of flutter by analysing the behaviour of a model in the subcritical region below the flutter boundary, which is referred to as the subcritical response. Analysis of the subcritical response of the model entails tracking the modal dampings and frequencies of the model as a function of increasing airspeed. Once sufficient data has been recorded, the damping trend of the model can be extrapolated to a flutter condition of zero damping (Ruhlin *et al.*, 1983). Various flight flutter testing techniques to analyse subcritical response data and safely approach the flutter speed exist, some of which are briefly described below. In terms of binary flutter models, the methods described are typically applied to the pitch time series data, as this is usually the likely mode of flutter. It should be noted that no single technique is sufficient to reliably predict the onset of flutter, so ideally several methods should be used simultaneously during flutter testing (Bennett, 1982). Ruhlin *et al.* (1983) and Zimmerman and Weissenburger (1964) described different methods of subcritical response analysis for on-line prediction of flutter speed during wind tunnel flutter tests. Some of these methods are simply stated below as presented in the literature. Ruhlin *et al.* (1983) found that all the methods listed below gave reliable predictions of flutter onset, but recommended the peak-hold and cross-spectrum methods for on-line use because they are the easiest and quickest to implement. With all the methods they found it necessary to conduct the flutter tests to within 7.0% to 10.0% of the flutter speed to obtain a reliable prediction.

2.4.1 Peak-Hold Method

The peak-hold method relates the peak amplitude F_p or peak amplitude squared F_p^2, in the frequency domain of one of the vibration modes to its damping. That is (Ruhlin *et al.,* 1983):

$$g_a \approx \frac{1}{F_p} \tag{2.1}$$

$$g_a \approx \frac{1}{F_p^2} \tag{2.2}$$

2.4.2 Half-Power Bandwidth (Power Spectral Density) Method

The PSD method relates the damping of one of the modes to its peak frequency f_p and the bandwidth Δf of the peak at half its amplitude, as given by (Ruhlin *et al.*, 1983):

$$g_a \approx \frac{\Delta f}{f_p} \tag{2.3}$$

2.4.3 Cross-Spectrum Method

The cross-spectrum method attempts to quantitatively measure the coupling between pitch and plunge motions as flutter is approached. As with the peak-hold method, the amplitude of a response peak was taken to be indicative of damping in that mode. It was found that a logarithmic relationship between damping and cross-spectrum amplitude gave the best prediction of flutter speed (Ruhlin *et al.*, 1983). Mathematically:

$$g_a \approx \log_{10}\left(\frac{1}{F_p}\right) \tag{2.4}$$

2.4.4 Zimmerman Flutter Margin Method

Zimmerman and Weissenburger (1964) defined a flutter margin Z as a function of the decay rates ζ and frequencies ω of the system modes, given by:

$$Z = \left[1 - \left(\frac{\zeta_2 - \zeta_1}{\zeta_2 + \zeta_1}\right)^2\right]\left\{\left(\frac{\omega_2^2 - \omega_1^2}{2}\right)^2 + \left(\zeta_2 + \zeta_1\right)^2\left[\left(\frac{\omega_2^2 + \omega_1^2}{2}\right) + \left(\frac{\zeta_2 + \zeta_1}{2}\right)^2\right]\right\} \tag{2.5}$$

At flutter, the decay rate of one of the modes (pitch or plunge) of the system becomes zero and hence the flutter margin becomes zero. Since the decay rate of a system is often difficult to measure, a simplified flutter margin calculation was proposed by Bennett (1982) that does not include damping. That is, by setting ζ_1 and $\zeta_2 = 0$ in Equation 2.5, the simplified flutter margin is expressed as:

$$Z_s \approx \left(\frac{\omega_2^2 - \omega_1^2}{2}\right)^2 \tag{2.6}$$

Both the Zimmerman and simplified Zimmerman flutter margins can be normalised by $Z_s^{(0)}$, which is defined as (Bennett, 1982):

$$Z_s^{(0)} \triangleq \left(\frac{\omega_2^2 - \omega_1^2}{2} \right)_{U=0}^2 \qquad \text{2.7}$$

Thus, the normalised, simplified Zimmerman flutter margin is expressed as:

$$\overline{Z}_s = \frac{Z_s}{Z_s^{(0)}} \qquad \text{2.8}$$

Reported results indicate that the simplified Zimmerman flutter margin typically gives a non-conservative flutter speed prediction (Bennett, 1982).

2.5 Conclusion

Two successful full-scale active flutter suppression flight test programs were outlined here to highlight the benefits of active flutter suppression control systems and provide the context and motivation for binary aeroservoelastic wind tunnel models. Aeroservoelastic mathematical models of binary pitch-plunge flutter models comprise structural equations of motion and equations characterising unsteady aerodynamic forces that are typically combined into a state-space model. In all cases, the structural equations of the aeroservoelastic systems are linearised about the aerofoil trim position as small modal displacements are assumed. Theodorsen's (1935) linearised unsteady aerodynamic theory with rational approximations developed by Jones (1945) is used almost exclusively to model aerodynamic forces. Where it is not used, either an aerodynamic coefficient based method or a DLM is used. Because of the low reduced frequencies in many of the reported wind tunnel flutter model tests, Theodorsen's (1935) theory is often further simplified by omitting the terms that account for unsteady aerodynamic forces that result from vortex shedding from the wing. These simplified unsteady aerodynamic models are termed quasi-steady models and wind tunnel test results have proved that they can adequately model binary aeroservoelastic systems. In some instances, an analytical model of the aeroservoelastic systems is not developed at all, but rather the dynamics of the system are determined experimentally using system identification techniques after initial wind tunnel tests. Since the dynamics of binary three degree-of-freedom systems are strongly coupled at and above their flutter speeds (by definition), a single control input is sufficient to suppress flutter. Consequently, flutter suppression is typically achieved through augmentation of the control surface hinge moment by changing the control surface angle of a full span trailing edge control surface. The addition of a control surface introduces a further degree-of-freedom (control surface rotation) into binary pitch-plunge flutter models that is not always modelled, although the effect of a control surface rotation on the unsteady lift and pitching moment is. In some of the previous active flutter control research reviewed, control surface dynamics are accounted for to some extent with an experimentally determined transfer function (typically second order), but these functions still do not model the unsteady control surface hinge moment. In other of the works reviewed, more complete models that account for both the unsteady control surface hinge moment and the control surface actuator dynamics were implemented. Time-domain control systems are typically used to suppress flutter. Optimal control theory is preferred, with a fixed LQR feedback control scheme using a Kalman filter to estimate unmeasured system states invariably being implemented to form a LQG flutter suppression controller. Other active flutter suppression control systems designed used H^{∞}, H_2, μ-synthesis or pole placement techniques. The literature highlights that because of the nature of flutter, control systems with fixed gains calculated for a specific flutter speed are not always robust over a range of open-loop supercritical flutter speeds. It is therefore suggested that various control laws be designed for different flight conditions that can be implemented using a gain scheduled function. Based on the representative literature reviewed, pitch-plunge wind tunnel models are suitable active control research tools, but only a few different low speed, subsonic active flutter suppression wind tunnel models are described in the literature. All of these models were mounted to either a flexure or coil spring mechanism to provide pitch-plunge dynamics. The models were generally designed to have low frequency (typically < 5*Hz*.), gentle flutter characteristics at low speeds (typically < 25*m*/*s*). The control surfaces used on these models to suppress flutter were actuated by model aircraft servos, linear actuators, piezoelectric actuators or small DC electric motors. Flutter could be suppressed on all the models described in the literature, and on average the flutter boundaries of these models were extended by nominally 40%. The smallest demonstrated increase in flutter speed through active control of the wind tunnel model was 10.0% and the largest was 102.6%. For flight (wind tunnel) flutter testing of aeroservoelastic wind tunnel modes, the peak-hold, half-power bandwidth, cross-spectrum and Zimmerman (or simplified Zimmerman) techniques are shown to be suitable to safely estimate the flutter speed based on the system's sub-critical response.

In view of these findings, in this research a complete linearised aeroservoelastic model including control surface dynamics and vortex shedding effects was developed, and a constant LQG flutter suppression controller was designed and implemented. Theoretical details of the aeroservoelastic mathematical model developed are given in Chapter 3. The half-power bandwidth and simplified Zimmerman flutter margin techniques were considered suitable for use in the wind tunnel flutter tests of this work.

Chapter 3

Aeroservoelastic Mathematical Model

This chapter details the development of a mathematical model to describe the flutter dynamics of a binary three degree-of-freedom (3DOF) flutter system and the design of a LQG controller to suppress flutter. The 3DOF aeroelastic model and LQG controller were combined to form an aeroservoelastic model which was implemented on the wind tunnel model used to demonstrate active flutter suppression. The structural equations of the 3DOF binary flutter model are outlined in Section 3.1.1 and the aerodynamic equations describing the unsteady lift, pitching moment and control surface hinge moment are developed in Section 3.1.2. Both sets of equations are combined in a state-space formulation in Section 3.1.3 to create a model that incorporates the full dynamics of the fluttering aerofoil. Section 3.1.4 outlines a method of including actuator dynamics into the aeroservoelastic model and Section 3.1.5 discusses ways of simplifying and reducing the order of the mathematical model, to minimise computer overheads as required for practical implementation of the control system. Section 3.1.6 describes an eigenvalue analysis of the aeroservoelastic model to calculate the anticipated open-loop flutter speed of the model. Having developed and simplified an appropriate state-space model that accurately models the oscillatory behaviour of a 3DOF binary flutter system, the LQR regulator and Kalman filter observer (and their application to the aeroservoelastic model) are described in Sections 3.2.1 and 3.2.2 respectively. The Kalman filter was used to estimate system states that could not be measured, to account for uncertainties and simplifications in the mathematical model and to model noise on the measured signals. Section 3.2.3 shows how the LQR regulator and Kalman filter are combined to form a LQG compensator, while Sections 3.2.4 and 3.2.5 highlight issues that were considered to enable practical implementation of the LQG flutter suppression controller on the wind tunnel model.

3.1 Aeroelastic Model

Flutter is a function of inertial, aerodynamic and elastic forces (Collar, 1946) thus equations that describe the structural response of each of the vibration modes and the unsteady aerodynamic forces acting on the aerofoil must be developed and combined to model the full flutter dynamics of the aerofoil. With reference to Figure 3.1 the three degrees-of-freedom in the aeroelastic model developed here are a plunge deflection h (positive downward), an aerofoil pitch angle α (positive nose-up) and a control surface deflection β (positive for a downwards rotation). The aerofoil definitions shown in Figure 3.1 and used in the development of the mathematical model of the aeroservoelastic system are (Fung, 1955):

- b [m] - the aerofoil semi-chord
- ab [m] - the distance between the mid-chord and the elastic axis (positive aft of the mid-chord)
- bx_α [m] - the distance between the elastic axis and the aerofoil centre of gravity (positive aft of the elastic axis)
- cb [m] - the distance between the mid-chord and the control surface hinge point (positive aft of the mid-chord)
- bx_β [m] - the distance between the control surface hinge point and the control surface centre of gravity (positive aft of the control surface hinge)

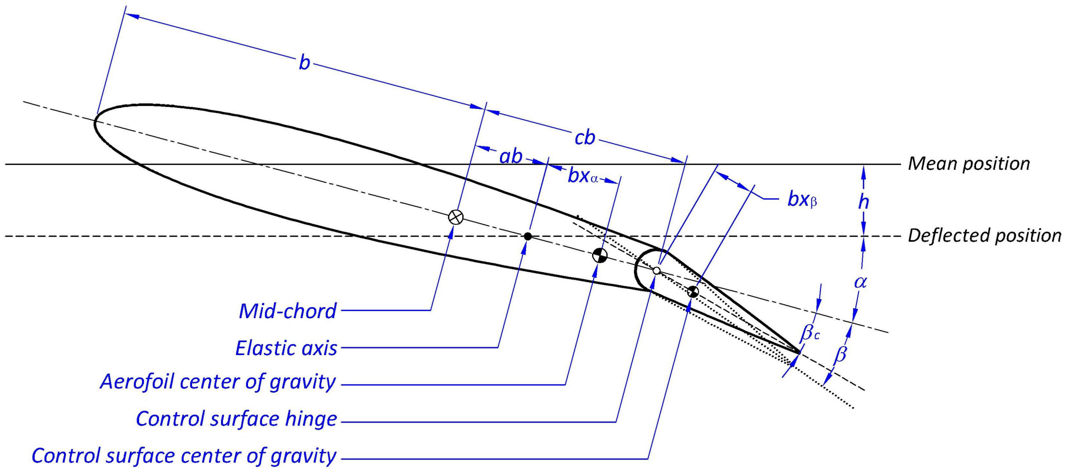


Figure 3.1 Aerofoil Notation (Derived from Fung, 1955)

3.1.1 Structural Equations

The equations of motion of the three degree-of-freedom aeroelastic model are obtained by applying Lagrange's equation (adapted to include Rayleigh's dissipation function for damping) to the wing section depicted in Figure 3.2. That is (Tse *et al.*, 1978):

$$\frac{d}{dt}\left(\frac{\partial T'}{\partial \dot{q}_i}\right)-\frac{\partial T'}{\partial q_i}+\frac{\partial D'}{\partial \dot{q}_i}+\frac{\partial U'}{\partial q_i}=F_i \quad \text{...3.1}$$

where q_i is the generalised co-ordinate and F_i the associated generalised force. T' denotes the kinetic energy and U' the potential energy of the system. Rayleigh's dissipation function (to account for damping in the system) is denoted by D'. In each case the prime denotes "per unit wingspan". The generalised co-ordinates of the fluttering aerofoil are h, α and β (plunge deflection, pitch angle and control surface deflection) and the associated generalised forces are L, M_α and M_β (lift, pitching moment and control surface hinge moment).

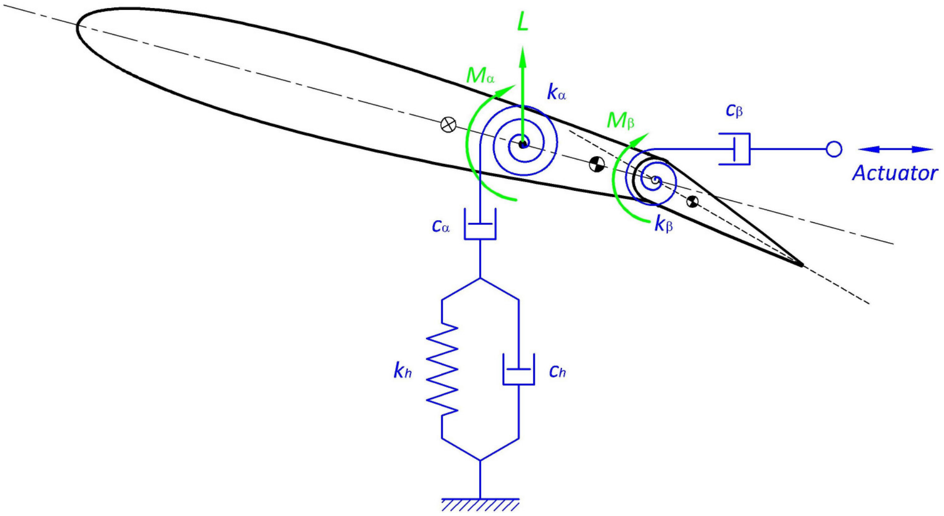


Figure 3.2 Fluttering Aerofoil Free Body Diagram (Derived from Fung, 1955)

If small modal oscillations and control surface deflections are assumed, then h, α and β are small and the structural equations of the model can be linearised by taking $\cos\alpha \approx 1$, $\sin\alpha \approx \alpha$, $\cos\beta \approx 1$ and $\sin\beta \approx \beta$ (see Figure 3.1). Under these assumptions, the total kinetic energy of the system per unit wingspan is given by (Scanlan and Rosenbaum, 1968):

$$T' = \tfrac{1}{2}m\dot{h}^2 + \tfrac{1}{2}I_\alpha\dot{\alpha}^2 + \tfrac{1}{2}I_\beta\dot{\beta}^2 + mx_\alpha b\dot{h}\dot{\alpha} + mx_\beta b\dot{h}\dot{\beta} + \left[(c-a)b^2 mx_\beta + I_\beta\right]\dot{\alpha}\dot{\beta} \quad \text{.................3.2}$$

and the total potential energy of the system per unit wingspan is given by (Scanlan and Rosenbaum, 1968):

$$U' = \tfrac{1}{2}k_h h^2 + \tfrac{1}{2}k_\alpha\alpha^2 + \tfrac{1}{2}k_\beta\left(\beta-\beta_c\right)^2 \quad \text{...3.3}$$

where the term β_c is the commanded control surface deflection. Assuming viscous damping, the Rayleigh dissipation function per unit wingspan of the aeroelastic system is given as[3]:

$$D_R' = \tfrac{1}{2}c_h\dot{h}^2 + \tfrac{1}{2}c_\alpha\dot{\alpha}^2 + \tfrac{1}{2}c_\beta\dot{\beta}^2 \quad \text{..3.4}$$

Applying Equation 3.1 to Equations 3.2, 3.3 and 3.4 for the three vibration modes of the aerofoil, the structural equations of motion, per unit wing span, of the binary flutter model are (Scanlan and Rosenbaum, 1968):

$$m\ddot{h} + mx_\alpha b\ddot{\alpha} + mx_\beta b\ddot{\beta} + c_h h + k_h h = L \quad \text{...3.5}$$

$$mx_\alpha b\ddot{h} + I_\alpha\ddot{\alpha} + \left[(c-a)b^2 mx_\beta + I_\beta\right]\ddot{\beta} + c_\alpha\dot{\alpha} + k_\alpha\alpha = M_\alpha \quad \text{..............................3.6}$$

$$mx_\beta b\ddot{h} + \left[(c-a)b^2 mx_\beta + I_\beta\right]\ddot{\alpha} + I_\beta\ddot{\beta} + c_\beta\dot{\beta} + k_\beta\left(\beta-\beta_c\right) = M_\beta \quad \text{........................3.7}$$

---

[3] See Appendix A for a formulation of Equation 3.4 using structural damping.

3.1.2 Unsteady Aerodynamic Forces

The lift, pitching moment and control surface hinge moment must be expressed in terms of the generalised co-ordinates of the system and substituted into the structural equations (Equations 3.5, 3.6 and 3.7) in order to obtain the full dynamics of the system. Simplified two-dimensional aerodynamic theory was used to model the unsteady aerodynamic forces acting on an aerofoil with plunge, pitch and control surface degrees of freedom. Various two-dimensional methods have been developed by several researchers to calculate these unsteady aerodynamic forces and moments acting on an oscillating aerofoil (Scanlan and Rosenbaum, 1968). The most common of these methods, and the method used in this study, is that of Theodorsen[4] (1935).

Theodorsen's theory is limited to aerofoils with thin sections and infinite aspect ratios, undergoing small oscillations in all vibration modes. Small oscillations are assumed so that that the flow over the aerofoil can be regarded to remain potential and unseparated (Fung, 1955). It is also assumed that the control surface is aerodynamically balanced and pivots about its leading edge (Theodorsen and Garrick, 1942). With these assumptions, under steady-state conditions and at constant velocity, the aerodynamic forces acting on the aerofoil are dependant on the circulation Γ of the fluid around the aerofoil, which is itself a function of the aerofoil's angle of attack, chord length and forward velocity, i.e.:

$$\Gamma = f\left(\alpha,\, c,\, U\right) \quad \text{.........} 3.8$$

For unsteady motion the aerodynamic forces L, M_α and M_β are no longer a function of Γ only, but also depend on the motion, instantaneous configuration (acceleration, rate and position) and circular frequency ω of the aeroelastic system (Scanlan and Rosenbaum, 1968). That is:

$$\left.\begin{matrix} L \\ M_\alpha \\ M_\beta \end{matrix}\right\} = f\left(\dot{h},\, \ddot{h},\, \alpha,\, \dot{\alpha},\, \ddot{\alpha},\, \beta,\, \dot{\beta},\, \ddot{\beta},\, U,\, \omega\right) \quad \text{.........} 3.9$$

The circular frequency of the fluttering aerofoil accounts for an additional and important contribution to the unsteady aerodynamic forces generated by the aerofoil due to its motion. A change in state of the aerofoil results in a change of circulation around it, accompanied by a shed vortex at the trailing edge. These shed vortices contribute to the aerodynamic forces generated by the aerofoil because of the vertical airflow they create near its trailing edge, before dissipating downstream. A useful parameter in assessing the contribution that this vortex shedding (wake circulation) makes to the overall aerodynamic forces acting on the aeroelastic system is the non-dimensional *reduced frequency* or *Strouhal* number given by Equation 3.10. It characterises the

---

[4] Zeiler (2000) points out that due to limitations in computing power at the time their work was done, the reports of Theodorsen and Garrick contain some errors which have been noted during this research. The theory is however faultless, and the reported errors do not in any way detract from its validity.

variation of flow with time and the mutual influence between the motion at various points of the oscillating aerofoil by the way in which a disturbance at one point on the aerofoil is felt at other points as a function of the frequency at which vortices are shed (Fung, 1955). The inverse of the reduced frequency (Equation 3.10) is the *reduced speed* (Fung, 1955).

$$k=\frac{b\omega}{U} \qquad 3.10$$

Some theories (termed *quasi-steady* solutions) omit the contribution that this wake circulation makes to the overall aerodynamic forces generated by the aerofoil. These solutions are, however, only valid when the reduced frequency of the system is low (Bisplinghoff *et al.*, 1957). In this study circulatory lift caused by vortex shedding was accounted for. Qualitatively, the unsteady aerodynamic forces are solved by setting up a (velocity) potential function for each of the parameters listed in Equation 3.9. A thorough formulation of these velocity potential functions is given in Theodorsen (1935). Having set up potential functions for each parameter, Bernoulli's theorem is used to calculate the pressure distribution around the aerofoil, and hence the aerodynamic forces it generates. Theodorsen (1935) shows that the unsteady lift, pitching moment and control surface hinge moment (per unit wingspan) of a typical aerofoil section are:

$$\begin{aligned}L(t)=&-\rho b^2\left(\pi\ddot{h}-\pi ba\ddot{\alpha}-bT_1\ddot{\beta}+U\pi\dot{\alpha}-UT_4\dot{\beta}\right)-\\&2\pi\rho bC(k)\left[\dot{h}+b\left(\tfrac{1}{2}-a\right)\dot{\alpha}+\tfrac{1}{2\pi}bT_{11}\dot{\beta}+U\alpha+\tfrac{1}{\pi}T_{10}U\beta\right]\end{aligned} \qquad 3.11$$

$$\begin{aligned}M_\alpha(t)=&-\rho b^2\left\{-\pi ab\ddot{h}+\pi b^2\left(\tfrac{1}{8}+a^2\right)\ddot{\alpha}-b^2\left[T_7+(c-a)T_1\right]\ddot{\beta}+\right.\\&\left.\pi bU\left(\tfrac{1}{2}-a\right)\dot{\alpha}+bU\left[T_1-T_8-(c-a)T_4+\tfrac{1}{2}T_{11}\right]\dot{\beta}+\left(T_4+T_{10}\right)U^2\beta\right\}+\\&2\pi\rho b^2U\left(\tfrac{1}{2}+a\right)C(k)\left[\dot{h}+b\left(\tfrac{1}{2}-a\right)\dot{\alpha}+\tfrac{1}{2\pi}bT_{11}\dot{\beta}+U\alpha+\tfrac{1}{\pi}T_{10}U\beta\right]\end{aligned} \qquad 3.12$$

$$\begin{aligned}M_\beta(t)=&-\rho b^2\left\{-bT_1\ddot{h}+2b^2T_{13}\ddot{\alpha}-\tfrac{1}{\pi}b^2T_3\ddot{\beta}-Ub\left[2T_9+T_1-T_4\left(\tfrac{1}{2}-a\right)\right]\dot{\alpha}-\right.\\&\left.\tfrac{1}{2\pi}UbT_4T_{11}\dot{\beta}+\tfrac{1}{\pi}U^2\left(T_5-T_4T_{10}\right)\beta\right\}-\rho b^2UT_{12}C(k)\left[\dot{h}+b\left(\tfrac{1}{2}-a\right)\dot{\alpha}+\right.\\&\left.\tfrac{1}{2\pi}bT_{11}\dot{\beta}+U\alpha+\tfrac{1}{\pi}T_{10}U\beta\right]\end{aligned} \qquad 3.13$$

where *C*(*k*) is known as Theodorsen's function and accounts for the circulatory lift caused by vortex shedding of the fluttering aerofoil (Fung, 1955). Theodorsen's function is expressed in terms of either Bessel or Hänkel functions (Equation 3.14) and is dependant on the reduced frequency *k* of the aerofoil. The *T* terms in Equations 3.11, 3.12 and 3.13 are functions of the chord-wise position of the control surface as defined by Garrick (1938) and listed in Appendix B.

$$C(k)=\frac{H_1^{(2)}(k)}{H_1^{(2)}(k)+H_0^{(2)}(k)} \qquad 3.14$$

Since each generalised aerodynamic force in Equations 3.11, 3.12 and 3.13 result from circulatory and non-circulatory flow, the equations can each be split into two parts, with elements denoted by the subscripts *c* and *nc* as follows[5]:

$$L = L_c + L_{nc} \quad \text{3.15}$$

$$M_\alpha = M_{\alpha_c} + M_{\alpha_{nc}} \quad \text{3.16}$$

$$M_\beta = M_{\beta_c} + M_{\beta_{nc}} \quad \text{3.17}$$

Considering only the circulatory flow contribution to the unsteady aerodynamic forces, i.e. those terms containing *C*(*k*) in Equations 3.11, 3.12 and 3.13, it is apparent that:

$$L_c(t) = -2\pi\rho bC(k)\left[\dot{h} + b\left(\tfrac{1}{2}-a\right)\dot{\alpha} + \tfrac{1}{2\pi}bT_{11}\dot{\beta} + U\alpha + \tfrac{1}{\pi}T_{10}U\beta\right] \quad \text{3.18}$$

$$M_{\alpha_c}(t) = 2\pi\rho b^2U\left(\tfrac{1}{2}+a\right)C(k)\left[\dot{h} + b\left(\tfrac{1}{2}-a\right)\dot{\alpha} + \tfrac{1}{2\pi}bT_{11}\dot{\beta} + U\alpha + \tfrac{1}{\pi}T_{10}U\beta\right] \quad \text{3.19}$$

$$M_{\beta_c}(t) = -\rho b^2UT_{12}C(k)\left[\dot{h} + b\left(\tfrac{1}{2}-a\right)\dot{\alpha} + \tfrac{1}{2\pi}bT_{11}\dot{\beta} + U\alpha + \tfrac{1}{\pi}T_{10}U\beta\right] \quad \text{3.20}$$

These equations for the circulatory lift, pitching moment and control surface hinge moment contain a common element *Q*, given by:

$$Q(t) = \dot{h} + b\left(\tfrac{1}{2}-a\right)\dot{\alpha} + \tfrac{1}{2\pi}bT_{11}\dot{\beta} + U\alpha + \tfrac{1}{\pi}T_{10}U\beta \quad \text{3.21}$$

Using Equation 3.21, Equations 3.18, 3.19 and 3.20 can be expressed as:

$$L_c(t) = -2\pi\rho bC(k)Q(t) \quad \text{3.22}$$

$$M_{\alpha_c}(t) = 2\pi\rho b^2C(k)Q(t) \quad \text{3.23}$$

$$M_{\beta_c}(t) = -\rho b^2UT_{12}C(k)Q(t) \quad \text{3.24}$$

To further develop the aerodynamic forces acting on the aerofoil it is assumed that all vibration modes exhibit simple harmonic motion. Under this assumption, the positions, rates and accelerations of the respective degrees-of-freedom of the system can be expressed in the frequency domain in terms of the circular frequency ω by:

$$\left.\begin{array}{lll} h = f(t) = h_0e^{i\omega t}; & \dot{h} = i\omega h; & \ddot{h} = -\omega^2 h \\ \alpha = f(t) = \alpha_0e^{i\omega t}; & \dot{\alpha} = i\omega\alpha; & \ddot{\alpha} = -\omega^2\alpha \\ \beta = f(t) = \beta_0e^{i\omega t}; & \dot{\beta} = i\omega\beta; & \ddot{\beta} = -\omega^2\beta \end{array}\right\} \quad \text{3.25}$$

Using Equation 3.25, Equation 3.21 is temporally converted from the time domain (transient solution) to the frequency domain (oscillatory solution) as:

---

[5] Marretta and Marino (2007) describe the reduced frequency dependant circulatory unsteady aerodynamic forces as self-excited forces.

$$Q(\omega)=i\omega h+b\left(\tfrac{1}{2}-a\right)i\omega\alpha+\tfrac{1}{2\pi}bT_{11}i\omega\beta+U\alpha+\tfrac{1}{\pi}T_{10}U\beta \quad 3.26$$

to give Equations 3.18, 3.19 and 3.20 in an oscillatory form. By removing a factor of $1/i\omega$, Equation 3.26 becomes:

$$Q(\omega)=\frac{1}{i\omega}\left[-\omega^2 h-b\left(\tfrac{1}{2}-a\right)\omega^2\alpha-\tfrac{1}{2\pi}bT_{11}\omega^2\beta+Ui\omega\alpha+\tfrac{1}{\pi}T_{10}Ui\omega\beta\right] \quad 3.27$$

which can be written as:

$$Q(\omega)=\frac{1}{i\omega}Q'(\omega) \quad 3.28$$

where:

$$Q'(\omega)=-\omega^2 h-b\left(\tfrac{1}{2}-a\right)\omega^2\alpha-\tfrac{1}{2\pi}bT_{11}\omega^2\beta+Ui\omega\alpha+\tfrac{1}{\pi}T_{10}Ui\omega\beta \quad 3.29$$

Equations 3.22, 3.23 and 3.24 are now converted into the frequency domain by substituting Equation 3.29 into them, to give:

$$L_c(\omega)=-2\pi\rho b\frac{C(k)}{i\omega}Q'(\omega) \quad 3.30$$

$$M_{\alpha_c}(\omega)=2\pi\rho b^2\frac{C(k)}{i\omega}Q'(\omega) \quad 3.31$$

$$M_{\beta_c}(\omega)=-\rho b^2UT_{12}\frac{C(k)}{i\omega}Q'(\omega) \quad 3.32$$

The reason for writing Equation 3.26 in the equivalent form of Equation 3.28, to yield Equations 3.30, 3.31 and 3.32, is to obtain a rational expression for the circulatory contribution to the lift, pitching moment and control surface hinge moment. This is done with Wagner's function (Wagner, 1925). Wagner's function Φ is related to Theodorsen's function by the inverse Fourier transform, as given by (Rodden and Stahl, 1969):

$$\Phi=\mathcal{F}^{-1}\left\{\frac{C(k)}{i\omega}\right\} \quad 3.33$$

An explicit expression does not exist for Wagner's function, but a common and accurate empirical exponential approximation, assuming $\Phi = f(U, t, b)$, is (Rodden and Stahl, 1969):

$$\Phi=1-\sum\delta_n e^{\frac{-\lambda_n U}{b}t} \quad 3.34$$

A suitable two term approximation for Equation 3.34 at low subsonic speeds was given by W.P. Jones from Rodden and Stahl (1969):

$$\Phi = 1 - \delta_1 e^{\frac{-\lambda_1 U}{b}t} - \delta_2 e^{\frac{-\lambda_2 U}{b}t} \quad \text{3.35}$$

where $\delta_1 = 0.165$, $\lambda_1 = 0.041$, $\delta_2 = 0.335$ and $\lambda_2 = 0.320$. Since the oscillatory unsteady aerodynamic loading may be regarded as the Fourier transform of the transient unsteady aerodynamic loading (Rodden and Stahl, 1969), Equations 3.30 to 3.32 can be converted back to the time domain using Fourier inversion and convolution theory. The inverse Fourier transform of two functions multiplied together (i.e. $\mathcal{F}^{-1}\{F(\omega)G(\omega)\}$) is their convolution[6]. The convolution of two functions is (Stroud, 1987):

$$f * g = \int_0^t f(t-\tau)g(\tau)d\tau \quad \text{3.36}$$

where *f* and *g* are the inverse Fourier transforms of the functions $F(\omega)$ and $G(\omega)$ respectively. Substituting the functions *f* and *g* in Equation 3.36 with the appropriate inverse Fourier transforms ($\mathcal{F}^{-1}\{C(k)/i\omega\}$ and $\mathcal{F}^{-1}\{Q'\}$) of the two non-constant terms in Equations 3.30, 3.31 and 3.32, the Duhamel integral for each of the circulatory parts of the unsteady aerodynamic forces is defined as:

$$D(t) = \int_0^t \Phi(t-\tau)\dot{Q}(\tau)d\tau \quad \text{3.37}$$

since:

$$\mathcal{F}^{-1}\{Q'\} = \dot{Q} \quad \text{3.38}$$

where:

$$\dot{Q} = \ddot{h} + b\left(\tfrac{1}{2} - a\right)\ddot{\alpha} + \tfrac{1}{2\pi} b T_{11}\ddot{\beta} + U\dot{\alpha} + \tfrac{1}{\pi} T_{10} U \dot{\beta} \quad \text{3.39}$$

and the inverse Fourier transform of Theodorsen's function has been related to Wagner's function in Equation 3.33. Substitution of Equation 3.35 into Equation 3.37 yields:

$$D(t) = \int_0^t \left[1 - \delta_1 e^{\frac{-\lambda_1 U}{b}(t-\tau)} - \delta_2 e^{\frac{-\lambda_2 U}{b}(t-\tau)}\right]\dot{Q}(\tau)d\tau \quad \text{3.40}$$

which can be multiplied out to give:

---

[6] The convolution function of two functions is also known as the Duhamel integral (James *et al.*, 1999).

$$D(t)=\int_0^t \dot{Q}(\tau)d\tau-\delta_1\int_0^t e^{\frac{-\lambda_1 U}{b}(t-\tau)}\dot{Q}(\tau)d\tau-\delta_2\int_0^t e^{\frac{-\lambda_2 U}{b}(t-\tau)}\dot{Q}(\tau)d\tau \quad \ldots\ldots 3.41$$

Thus the circulatory flow contribution to the unsteady forces generated by the fluttering aerofoil given by Equations 3.30, 3.31 and 3.32, can now be expressed as:

$$L_c(t)=-2\pi\rho bD(t) \quad \ldots\ldots 3.42$$

$$M_{\alpha_c}(t)=2\pi\rho b^2D(t) \quad \ldots\ldots 3.43$$

$$M_{\beta_c}(t)=-\rho b^2UT_{12}D(t) \quad \ldots\ldots 3.44$$

To obtain a rational solution to Equations 3.42, 3.43 and 3.44, Equation 3.41 must be solved. The first term in Equation 3.41 is easily integrated as:

$$\int_0^t \dot{Q}(\tau)d\tau=Q(t) \quad \ldots\ldots 3.45$$

whilst the remaining two terms require further manipulation to be solved. Initially, the integrals in these two terms are defined as ℓ_1 and ℓ_2 and expressed by:

$$\ell_n \triangleq \int_0^t e^{\frac{-\lambda_n U}{b}(t-\tau)}\dot{Q}(\tau)d\tau \quad \ldots\ldots 3.46$$

which can be partially solved and simplified to:

$$\ell_n=e^{\frac{-\lambda_n U}{b}t}\int_0^t e^{\frac{\lambda_n U}{b}\tau}\dot{Q}(\tau)d\tau \quad \ldots\ldots 3.47$$

The term ℓ_n is defined as the n^{th} aerodynamic lag state, and is a measure of the lag in the induced aerodynamic loads in following the motion of the aerofoil[7] (Rodden and Stahl, 1969). This definition together with Equation 3.45 allows Equation 3.41 to be written as:

$$D(t)=Q(t)-\sum\delta_n\ell_n(t) \quad \ldots\ldots 3.48$$

In this work n = 2 was used. Equation 3.48 can now be substituted into Equations 3.42, 3.43 and 3.44 to provide explicit approximations for the circulatory contribution to the unsteady aerodynamic forces, now written as:

---

[7] A practical example of shed vortices creating aerodynamic lag states that augment the lift forces of a fluttering wing is that of birds flying in a "V" formation, where the shed vortices off each bird as it flaps its wings induce lift for the bird behind it.

$$L_c(t) = -2\pi\rho b\left(Q(t) - \delta_1\ell_1 - \delta_2\ell_2\right) \quad \text{3.49}$$

$$M_{\alpha_c}(t) = 2\pi\rho b^2\left(Q(t) - \delta_1\ell_1 - \delta_2\ell_2\right) \quad \text{3.50}$$

$$M_{\beta_c}(t) = -\rho b^2 U T_{12}\left(Q(t) - \delta_1\ell_1 - \delta_2\ell_2\right) \quad \text{3.51}$$

To solve these equations, a solution for ℓ_n must be obtained. To achieve this, the derivatives of both sides of Equation 3.47 are calculated using the product rule (Stroud, 1987) as:

$$\dot{\ell}_n = \frac{-\lambda_n U}{b} e^{\frac{-\lambda_n U}{b}t} \int_0^t e^{\frac{\lambda_n U}{b}\tau} \dot{Q}(\tau)\,d\tau + e^{\frac{-\lambda_n U}{b}t} \frac{d}{dt}\left[\int_0^t e^{\frac{\lambda_n U}{b}\tau} \dot{Q}(\tau)\,d\tau\right] \quad \text{3.52}$$

where the first term in the expression for $\dot{\ell}_n$ is simply $\left(-\lambda_n U/b\right)\ell_n$ from Equation 3.47 and by the Fundamental Theorem of Calculus (Stroud, 1987) the second term can be simplified to:

$$\frac{d}{dt}\left[\int_0^t e^{\frac{\lambda_n U}{b}\tau} \dot{Q}(\tau)\,d\tau\right] = e^{\frac{\lambda_n U}{b}t} \dot{Q}(t) \quad \text{3.53}$$

Thus the final expression for $\dot{\ell}_n$ becomes:

$$\dot{\ell}_n(t) = \frac{-\lambda_n U}{b} \ell_n(t) + \dot{Q}(t) \quad \text{3.54}$$

This solution is elegant in that it eliminates the need to solve the Duhamel integral and allows ℓ_n to be solved with two (assuming *n* = 2) simultaneous linear differential equations. Thus, the complete aerodynamic model of the system can now to be solved simultaneously by a system of first order linear[8] differential equations. That is, once factorised, the final equations describing the total (circulatory and non-circulatory) unsteady aerodynamic forces acting on the aerofoil are:

$$\begin{aligned} L(t) = & -\pi\rho b^2\ddot{h} + \pi\rho b^3 a\ddot{\alpha} + \rho b^3 T_1\ddot{\beta} - 2\pi\rho bU\dot{h} - 2\pi\rho b^3(1-a)\dot{\alpha} + \rho b^2 U(T_4 - T_{11})\dot{\beta} - \\ & 2\pi\rho bU^2\alpha - 2\rho bU^2 T_{10}\beta + 2\pi\rho bU\delta_1\ell_1 + 2\pi\rho b^2 U\delta_2\ell_2 \end{aligned} \quad \text{3.55}$$

$$\begin{aligned} M_\alpha(t) = & \pi\rho b^3 a\ddot{h} - \pi\rho b^4\left(\tfrac{1}{8} + a^2\right)\ddot{\alpha} + \rho b^4\left[T_7 + (c-a)T_1\right]\ddot{\beta} - 2\pi\rho b^2 U\left(\tfrac{1}{2} + a\right)\dot{h} + \\ & 2\pi\rho b^3 Ua\left(\tfrac{1}{2} - a\right)\dot{\alpha} + \rho b^3 U\left[T_8 - T_1 + (c-a)T_4 + aT_{11}\right]\dot{\beta} + 2\pi\rho b^2 U^2\left(\tfrac{1}{2} + a\right)\alpha + \\ & \rho b^2 U^2\left(2aT_{10} - T_4\right)\beta - 2\pi\rho b^2 U\left(\tfrac{1}{2} + a\right)\delta_1\ell_1 - 2\pi\rho b^2 U\left(\tfrac{1}{2} + a\right)\delta_2\ell_2 \end{aligned} \quad \text{3.56}$$

$$\begin{aligned} M_\beta(t) = & \rho b^3 T_1\ddot{h} - 2\rho b^4 T_{13}\ddot{\alpha} + \rho b^4 \tfrac{1}{\pi} T_3\ddot{\beta} - \rho b^2 U T_{12}\dot{h} + \\ & \rho b^3 U\left[2T_9 + T_1 + (T_4 - T_{12})\left(\tfrac{1}{2} - a\right)\right]\dot{\alpha} + \rho b^3 U\tfrac{1}{2\pi}T_{11}\left(T_4 - T_{12}\right)\dot{\beta} - \rho b^2 U^2 T_{12}\alpha - \\ & \tfrac{1}{\pi}\rho b^2 U^2\left[T_5 - T_{10}\left(T_4 - T_{12}\right)\right]\beta + \rho b^2 U T_{12}\delta_1\ell_1 + \rho b^2 U T_{12}\delta_2\ell_2 \end{aligned} \quad \text{3.57}$$

---

[8] Linear differential equations result if ρ and *U* are held fixed in Equations 3.55, 3.56 and 3.57.

where ℓ_1 and ℓ_2 are obtained from the solution of:

$$\dot{Q}(t) = \dot{\ell}_1 + \frac{\lambda_1 U}{b}\ell_1 \quad \text{3.58}$$

$$\dot{Q}(t) = \dot{\ell}_2 + \frac{\lambda_2 U}{b}\ell_2 \quad \text{3.59}$$

The fluttering aerofoil unsteady aerodynamic force equations (Equations 3.55 to 3.57) are combined with the structural equations (Equations 3.5 to 3.7) in a state-space formulation to give the full equations of motion of the aeroservoelastic system. The development of the full state-space model of this three degree-of-freedom binary flutter model is presented in Section 3.1.3.

3.1.3 State-Space Model Formulation

To form a complete aeroservoelastic model of the system, the structural and aerodynamic sets of linear differential equations are combined in a *state-space* formulation. In this method, dynamic systems are described by a set of first order linear differential equations in variables called the state (Franklin *et al.*, 1991), shown in matrix form in Equation 3.60 (without measurement or process noise).

$$\left.\begin{aligned} \dot{\mathbf{X}} &= \mathbf{AX} + \mathbf{B}u \\ \mathbf{Y} &= \mathbf{CX} + \mathbf{D}u \end{aligned}\right\} \quad \text{3.60}$$

The vector **X** is the state vector and **Y** the measurement vector. The matrix **A** is known as the *system* matrix, **B** the *input* matrix, **C** the *output* or *measurement* matrix and **D** the *direct transmission* or *feed-through* matrix (Franklin *et al.*, 1991). The control variable *u* is used to control the system. Representing the aeroservoelastic model in state-space form allows the complete mathematical model to be solved with matrix methods and is convenient for the subsequent flutter suppression control law design. The following subsections briefly outline the derivation of the system, input and measurement matrices for the aeroelastic model. Direct feed-through control was not investigated or performed in this work, so the feed-through matrix (**D**) was omitted from Equation 3.60.

State Vector

Prior to development of the state-space model matrices of the binary three degree-of-freedom flutter model, a state vector must be defined. The development of a full state vector for the aeroservoelastic system under investigation requires the formulation of two partial state vectors. The initial partial state vector describing the modal positions of the three degrees-of-freedom of the flutter model is defined as:

$$\mathbf{X}_s \triangleq \begin{bmatrix} h & \alpha & \beta \end{bmatrix}^T \quad \text{3.61}$$

where the subscript *s* denotes “structural”. Using this partial state vector $\mathbf{X}_s$, the structural equations of the flutter model given by Equations 3.5, 3.6 and 3.7 are written in matrix form as:

$$\mathbf{M}_s\ddot{\mathbf{X}}_s + \mathbf{D}_s\dot{\mathbf{X}}_s + \mathbf{K}_s\mathbf{X}_s = \mathbf{F}_a + \mathbf{B}u \quad 3.62$$

where $\mathbf{F}_a$ is a matrix of unsteady aerodynamic forces generated by the aerofoil. The structural mass $\mathbf{M}_s$, damping $\mathbf{D}_s$ and stiffness $\mathbf{K}_s$ matrices of the system are:

$$\mathbf{M}_s = \begin{bmatrix} m & mx_\alpha b & mx_\beta b \\ mx_\alpha b & I_\alpha & (c-a)b^2 mx_\beta + I_\beta \\ mx_\beta b & (c-a)b^2 mx_\beta + I_\beta & I_\beta \end{bmatrix} \quad 3.63$$

$$\mathbf{D}_s = \begin{bmatrix} c_h & 0 & 0 \\ 0 & c_\alpha & 0 \\ 0 & 0 & c_\beta \end{bmatrix} \quad 3.64$$

$$\mathbf{K}_s = \begin{bmatrix} k_h & 0 & 0 \\ 0 & k_\alpha & 0 \\ 0 & 0 & k_\beta \end{bmatrix} \quad 3.65$$

The unsteady aerodynamic forces must now also be converted to matrix form. It was shown in Section 3.1.2 that the unsteady aerodynamic forces and moments generated by an oscillating aerofoil have circulatory and non-circulatory components (Equations 3.15, 3.16 and 3.17). The non-circulatory components can be written wholly as a function of the state vector $\mathbf{X}_s$ (Equation 3.61), while the circulatory contributions to the unsteady aerodynamics are written partly as a function of $\mathbf{X}_s$ and partly as a function of the two aerodynamic lag states (Equations 3.58 and 3.59). To account for the two aerodynamic lag states and to express the total unsteady aerodynamic forces in matrix notation, an additional partial state vector must be defined as:

$$\mathbf{X}_a \triangleq \begin{bmatrix} \ell_1 & \ell_2 \end{bmatrix}^T \quad 3.66$$

where the subscript *a* denotes “aerodynamic”. This state definition, together with the definition of the partial state vector $\mathbf{X}_s$, allows the unsteady aerodynamic forces to be written in matrix notation as:

$$\mathbf{F}_a = \mathbf{M}_a\ddot{\mathbf{X}}_s + \mathbf{D}_a\dot{\mathbf{X}}_s + \mathbf{K}_a\mathbf{X}_s + \mathbf{L}_\delta\mathbf{X}_a \quad 3.67$$

The two aerodynamic lag sates are accounted for in:

$$\mathbf{L}_\delta = \rho b^2 U \begin{bmatrix} \dfrac{2\pi\delta_1}{b} & \dfrac{2\pi\delta_2}{b} \\ -2\pi\left(\frac{1}{2}+a\right)\delta_1 & -2\pi\left(\frac{1}{2}+a\right)\delta_2 \\ T_{12}\delta_1 & T_{12}\delta_2 \end{bmatrix} \quad \text{......} \; 3.68$$

as derived from Equations 3.55, 3.56 and 3.57. The non-circulatory and remaining circulatory contributions to the unsteady lift, pitching moment and control surface hinge moment are accounted for in the "aerodynamic mass", "aerodynamic damping" and "aerodynamic stiffness" matrices, given by Equations 3.69, 3.70 and 3.71 respectively[9].

$$\mathbf{M}_a = \rho b^3 \begin{bmatrix} -\dfrac{\pi}{b} & \pi a & T_1 \\ \pi a & -\pi b\left(\frac{1}{8}+a^2\right) & b\left[T_7+(c-a)T_1\right] \\ T_1 & -2bT_{13} & \dfrac{1}{\pi}bT_3 \end{bmatrix} \quad \text{......} \; 3.69$$

$$\mathbf{D}_a = \rho b^2 U \begin{bmatrix} -\dfrac{2\pi}{b} & -2\pi(1-a) & T_4 - T_{11} \\ 2\pi\left(\frac{1}{2}+a\right) & 2\pi b a\left(\frac{1}{2}-a\right) & b\left[T_8 - T_1 + (c-a)T_4 + aT_{11}\right] \\ -T_{12} & b\left[2T_9 + T_1 + (T_4 - T_{12})\left(\frac{1}{2}-a\right)\right] & \dfrac{1}{2\pi}bT_{11}(T_4 - T_{12}) \end{bmatrix} \quad \text{......} \; 3.70$$

$$\mathbf{K}_a = \rho b^2 U^2 \begin{bmatrix} 0 & -\dfrac{2\pi}{b} & -\dfrac{2T_{10}}{b} \\ 0 & 2\pi\left(\frac{1}{2}+a\right) & 2aT_{10} - T_4 \\ 0 & -T_{12} & -\dfrac{1}{\pi}\left[T_5 - T_{10}(T_4 - T_{12})\right] \end{bmatrix} \quad \text{......} \; 3.71$$

The generalised forces in the structural equations of motion of the system (Equation 3.62) can now be substituted with the matrix representations of the unsteady aerodynamic forces generated by the fluttering aerofoil (Equation 3.67) to form a complete matrix representation of the whole binary aeroservoelastic system. That is:

$$\mathbf{M}_s\ddot{\mathbf{X}}_s + \mathbf{D}_s\dot{\mathbf{X}}_s + \mathbf{K}_s\mathbf{X}_s = \mathbf{M}_a\ddot{\mathbf{X}}_s + \mathbf{D}_a\dot{\mathbf{X}}_s + \mathbf{K}_a\mathbf{X}_s + \mathbf{L}_\delta\mathbf{X}_a + \mathbf{B}u \quad \text{......} \; 3.72$$

which can be factorised to:

$$\left(\mathbf{M}_s - \mathbf{M}_a\right)\ddot{\mathbf{X}}_s + \left(\mathbf{D}_s - \mathbf{D}_a\right)\dot{\mathbf{X}}_s + \left(\mathbf{K}_s - \mathbf{K}_a\right)\mathbf{X}_s = \mathbf{L}_\delta\mathbf{X}_a + \mathbf{B}u \quad \text{......} \; 3.73$$

and further simplified to:

---

[9] From Appendix B: $T_{13} = -\frac{1}{2}\left[T_7 + (c-a)T_1\right]$, so the aerodynamic mass matrix in Equation 3.69 is symmetric as required.

$$\bar{\mathbf{M}}\ddot{\mathbf{X}}_s + \bar{\mathbf{C}}\dot{\mathbf{X}}_s + \bar{\mathbf{K}}\mathbf{X}_s = \mathbf{L}_\delta \mathbf{X}_a + \mathbf{B}u \tag{3.74}$$

by defining:

$$\left.\begin{aligned} \bar{\mathbf{M}} &\triangleq \mathbf{M}_s - \mathbf{M}_a \\ \bar{\mathbf{D}} &\triangleq \mathbf{D}_s - \mathbf{D}_a \\ \bar{\mathbf{K}} &\triangleq \mathbf{K}_s - \mathbf{K}_a \end{aligned}\right\} \tag{3.75}$$

To convert Equation 3.74 to a state-space form, a complete state vector was defined by combining the partial state vectors $\mathbf{X}_s$ and $\mathbf{X}_a$ into one full state vector given as:

$$\mathbf{X} = \begin{bmatrix} \dot{\mathbf{X}}_s & \mathbf{X}_s & \mathbf{X}_a \end{bmatrix}^T \tag{3.76}$$

which expands to:

$$\mathbf{X} = \begin{bmatrix} \dot{h} & \dot{\alpha} & \dot{\beta} & h & \alpha & \beta & \ell_1 & \ell_2 \end{bmatrix}^T \tag{3.77}$$

With this definition of the full state vector, the system matrix **A**, input matrix **B** and measurement matrix **C** of the state-space model of the binary flutter system (Equation 3.60) were derived as shown below.

System Matrix

The system matrix of the state-space model of the three degree-of-freedom binary flutter system was obtained in elements, as defined in Equation 3.78.

$$\mathbf{A} \triangleq \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} & \mathbf{A}_{13} \\ \mathbf{A}_{21} & \mathbf{A}_{22} & \mathbf{A}_{23} \\ \mathbf{A}_{31} & \mathbf{A}_{32} & \mathbf{A}_{33} \end{bmatrix} \tag{3.78}$$

Substituting Equations 3.76 and 3.78 into the state-space model of the system (Equation 3.60) and omitting the control input **B***u* for now, it can be shown that:

$$\begin{bmatrix} \ddot{\mathbf{X}}_s \\ \dot{\mathbf{X}}_s \\ \dot{\mathbf{X}}_a \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} & \mathbf{A}_{13} \\ \mathbf{A}_{21} & \mathbf{A}_{22} & \mathbf{A}_{23} \\ \mathbf{A}_{31} & \mathbf{A}_{32} & \mathbf{A}_{33} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{X}}_s \\ \mathbf{X}_s \\ \mathbf{X}_a \end{bmatrix} \tag{3.79}$$

which was multiplied out to:

$$\ddot{\mathbf{X}}_s = \mathbf{A}_{11}\dot{\mathbf{X}}_s + \mathbf{A}_{12}\mathbf{X}_s + \mathbf{A}_{13}\mathbf{X}_a \tag{3.80}$$

$$\dot{\mathbf{X}}_s = \mathbf{A}_{21}\dot{\mathbf{X}}_s + \mathbf{A}_{22}\mathbf{X}_s + \mathbf{A}_{23}\mathbf{X}_a \tag{3.81}$$

$$\dot{\mathbf{X}}_a = \mathbf{A}_{31}\dot{\mathbf{X}}_s + \mathbf{A}_{32}\mathbf{X}_s + \mathbf{A}_{33}\mathbf{X}_a \tag{3.82}$$

To obtain the elements $\mathbf{A}_{11}$ to $\mathbf{A}_{13}$ of the system matrix, Equation 3.74 was solved for $\ddot{\mathbf{X}}_s$ as:

$$\ddot{\mathbf{X}}_s = -\bar{\mathbf{M}}^{-1}\bar{\mathbf{D}}\dot{\mathbf{X}}_s - \bar{\mathbf{M}}^{-1}\bar{\mathbf{K}}\mathbf{X}_s + \bar{\mathbf{M}}^{-1}\mathbf{L}_\delta\mathbf{X}_a + \bar{\mathbf{M}}^{-1}\mathbf{B}u \tag{3.83}$$

By comparing Equation 3.83 with Equation 3.80 it is apparent that:

$$\mathbf{A}_{11} = -\bar{\mathbf{M}}^{-1}\bar{\mathbf{D}} \tag{3.84}$$

$$\mathbf{A}_{12} = -\bar{\mathbf{M}}^{-1}\bar{\mathbf{K}} \tag{3.85}$$

$$\mathbf{A}_{13} = \bar{\mathbf{M}}^{-1}\mathbf{L}_\delta \tag{3.86}$$

The remaining term ($\bar{\mathbf{M}}^{-1}\mathbf{B}u$) in Equation 3.83 is accounted for in the derivation of the input matrix. To obtain the elements $\mathbf{A}_{21}$ to $\mathbf{A}_{23}$ in Equation 3.81, it is deduced from Equation 3.79 that:

$$\mathbf{A}_{21} = \mathbf{I}_{3\times 3} \tag{3.87}$$

$$\mathbf{A}_{22} = \mathbf{0}_{3\times 3} \tag{3.88}$$

$$\mathbf{A}_{23} = \mathbf{0}_{3\times 2} \tag{3.89}$$

where $\mathbf{I}_{n\times n}$ is an identity matrix of dimension *n* by *n* and $\mathbf{0}_{n\times m}$ is a matrix of zeros with *n* rows and *m* columns. To find the remaining system matrix elements $\mathbf{A}_{31}$, $\mathbf{A}_{32}$ and $\mathbf{A}_{33}$, which account for the two aerodynamic lag states in the unsteady aerodynamic model of the system, Equation 3.39 was written in matrix notation as:

$$\dot{\mathbf{Q}} = \begin{bmatrix} 1 & b\left(\tfrac{1}{2}-a\right) & \tfrac{1}{2\pi}bT_{11} \end{bmatrix}\ddot{\mathbf{X}}_s + U\begin{bmatrix} 0 & 1 & \tfrac{1}{\pi}T_{10} \end{bmatrix}\dot{\mathbf{X}}_s \tag{3.90}$$

which becomes:

$$\dot{\mathbf{Q}} = \mathbf{Q}_a\ddot{\mathbf{X}}_s + \mathbf{Q}_v\dot{\mathbf{X}}_s \tag{3.91}$$

if $\mathbf{Q}_a$ and $\mathbf{Q}_v$ are defined as:

$$\mathbf{Q}_a \triangleq \begin{bmatrix} 1 & b\left(\tfrac{1}{2}-a\right) & \tfrac{1}{2\pi}bT_{11} \end{bmatrix} \tag{3.92}$$

$$\mathbf{Q}_v \triangleq \begin{bmatrix} 0 & U & \tfrac{1}{\pi}UT_{10} \end{bmatrix} \tag{3.93}$$

Since $\mathbf{Q}_a$ is a vector of terms multiplied by the modal accelerations of the system and $\mathbf{Q}_v$ a vector of terms multiplied by the modal velocities of the system, the subscript *a* refers to “acceleration” and *v* to “velocity”. By solving Equations 3.58 and 3.59 for $\dot{\ell}_1$ and $\dot{\ell}_2$ respectively, and using Equation 3.66 it can be shown that:

$$\dot{\mathbf{X}}_a = \mathbf{L}_\lambda \mathbf{X}_a + \begin{bmatrix} \dot{\mathbf{Q}} & \dot{\mathbf{Q}} \end{bmatrix}^T \tag{3.94}$$

where $\mathbf{L}_\lambda$ is defined as:

$$\mathbf{L}_\lambda \triangleq \begin{bmatrix} -\dfrac{\lambda_1 U}{b} & 0 \\ 0 & -\dfrac{\lambda_2 U}{b} \end{bmatrix} \tag{3.95}$$

To obtain the system matrix elements $\mathbf{A}_{31}$, $\mathbf{A}_{32}$ and $\mathbf{A}_{33}$, Equation 3.80 was substituted into Equation 3.91 to yield:

$$\dot{\mathbf{Q}} = \mathbf{Q}_a \left[\mathbf{A}_{11}\dot{\mathbf{X}}_s + \mathbf{A}_{12}\mathbf{X}_s + \mathbf{A}_{13}\mathbf{X}_a \right] + \mathbf{Q}_v \dot{\mathbf{X}}_s \tag{3.96}$$

Equation 3.96 was then substituted into Equation 3.94, to give:

$$\dot{\mathbf{X}}_a = \mathbf{L}_\lambda \mathbf{X}_a + \mathbf{Q}_a \begin{bmatrix} \mathbf{A}_{11}\dot{\mathbf{X}}_s + \mathbf{A}_{12}\mathbf{X}_s + \mathbf{A}_{13}\mathbf{X}_a \\ \mathbf{A}_{11}\dot{\mathbf{X}}_s + \mathbf{A}_{12}\mathbf{X}_s + \mathbf{A}_{13}\mathbf{X}_a \end{bmatrix} + \mathbf{Q}_v \begin{bmatrix} \dot{\mathbf{X}}_s \\ \dot{\mathbf{X}}_s \end{bmatrix} \tag{3.97}$$

which, once factorised, becomes:

$$\dot{\mathbf{X}}_a = \begin{bmatrix} \mathbf{Q}_a\mathbf{A}_{11} + \mathbf{Q}_v \\ \mathbf{Q}_a\mathbf{A}_{11} + \mathbf{Q}_v \end{bmatrix} \dot{\mathbf{X}}_s + \begin{bmatrix} \mathbf{Q}_a\mathbf{A}_{12} \\ \mathbf{Q}_a\mathbf{A}_{12} \end{bmatrix} \mathbf{X}_s + \begin{bmatrix} \mathbf{Q}_a\mathbf{A}_{13} \\ \mathbf{Q}_a\mathbf{A}_{13} \end{bmatrix} \mathbf{X}_a + \mathbf{L}_\lambda \mathbf{X}_a \tag{3.98}$$

where, using Equation 3.82, the matrix elements $\mathbf{A}_{31}$, $\mathbf{A}_{32}$ and $\mathbf{A}_{33}$ were deduced as:

$$\mathbf{A}_{31} = \begin{bmatrix} \mathbf{Q}_a\mathbf{A}_{11} + \mathbf{Q}_v \\ \mathbf{Q}_a\mathbf{A}_{11} + \mathbf{Q}_v \end{bmatrix} \tag{3.99}$$

$$\mathbf{A}_{32} = \begin{bmatrix} \mathbf{Q}_a\mathbf{A}_{12} \\ \mathbf{Q}_a\mathbf{A}_{12} \end{bmatrix} \tag{3.100}$$

$$\mathbf{A}_{33} = \begin{bmatrix} \mathbf{Q}_a\mathbf{A}_{13} \\ \mathbf{Q}_a\mathbf{A}_{13} \end{bmatrix} + \mathbf{L}_\lambda \tag{3.101}$$

Thus the final system matrix of the three degree-of-freedom binary flutter model is given by Equation 3.102. This equation was used in an eigenvalue analysis of Equation 3.60 to calculate the flutter speed, modal dampings, modal frequencies and time domain response as a function of airspeed U and air density ρ of the active flutter suppression wind tunnel model.

$$\mathbf{A}=\left[\begin{array}{c:c:c} -\bar{\mathbf{M}}^{-1}\bar{\mathbf{D}} & -\bar{\mathbf{M}}^{-1}\bar{\mathbf{K}} & \bar{\mathbf{M}}^{-1}\mathbf{L}_{\delta} \\ \hdashline \mathbf{I}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 2} \\ \hdashline \begin{matrix} -\mathbf{Q}_{a}\bar{\mathbf{M}}^{-1}\bar{\mathbf{D}}+\mathbf{Q}_{v} \\ -\mathbf{Q}_{a}\bar{\mathbf{M}}^{-1}\bar{\mathbf{D}}+\mathbf{Q}_{v} \end{matrix} & \begin{matrix} -\mathbf{Q}_{a}\bar{\mathbf{M}}^{-1}\bar{\mathbf{K}} \\ -\mathbf{Q}_{a}\bar{\mathbf{M}}^{-1}\bar{\mathbf{K}} \end{matrix} & \begin{matrix} \mathbf{Q}_{a}\bar{\mathbf{M}}^{-1}\mathbf{L}_{\delta} \\ \mathbf{Q}_{a}\bar{\mathbf{M}}^{-1}\mathbf{L}_{\delta} \end{matrix}+\mathbf{L}_{\lambda} \end{array}\right] \quad \text{.........................} \; 3.102$$

Input Matrix

The input matrix for the state-space model of the binary flutter system was derived so that control to suppress flutter could be applied to the model through the trailing edge control surface. The control surface was used to control divergent oscillations that occur above the critical flutter speed of the model by replacing u in Equation 3.60 with a commanded control surface angle β_c to regulate the control surface hinge moment. This hinge moment applied to the control surface can be expressed by:

$$M_{\beta,\,c}=k_{\beta}\beta_{c} \quad \text{...} \; 3.103$$

From Equation 3.103 and Equation 3.83 the state-space model input matrix for the binary flutter system is then:

$$\mathbf{B}=\begin{bmatrix} \bar{\mathbf{M}}^{-1}k_{\beta} \\ \mathbf{0}_{5\times 1} \end{bmatrix} \quad \text{...} \; 3.104$$

The dynamics of the actuator driving the control surface were included in the state-space model of the complete aeroservoelastic system to ensure an effective flutter controller design, as outlined in Section 3.1.4.

Output Matrix

The output matrix of the system relates the state variables to the measured variables of the system. Not all system states were measured, either to reduce the number of sensors required (and hence reduce the cost of the wind tunnel model) or because they could not be measured. In this research the two aerodynamic lag states (Section 3.1.2) were unmeasurable. It was decided to measure only the plunge displacement, pitch angle and control surface angle and estimate the remaining system states using a suitable state estimator as discussed in Section 3.2.2. The measurement matrix for the complete aeroelastic system was therefore:

$$\mathbf{C}=\begin{bmatrix} \mathbf{0}_{3\times 3} & \mathbf{I}_{3\times 3} & \mathbf{0}_{3\times 2} \end{bmatrix} \quad \text{..} \; 3.105$$

3.1.4 Full Aeroservoelastic Model

To complete the full aeroservoelastic mathematical model, the dynamics of the control surface and its actuation system were characterised and added to the state-space model of the flutter system. It was important to include these dynamics so that the control input did not become out of phase with the pitch and plunge oscillations of the flutter model, and so enhance an unsteady oscillation rather than suppress it. In this work it was assumed that the control surface actuator had linear servo dynamics described by:

$$\dot{\beta}_c = \mathbf{a}\beta_c + \mathbf{b}\beta_a \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ 3.106$$

where β_a is the commanded control surface actuator position and β_c is the commanded control surface angle. Under this assumption of linear first order control surface actuator dynamics, it can be shown that the control surface actuator dynamics can be augmented to the state-space model of the aeroservoelastic system as follows:

$$\mathbf{A}' = \left[\begin{array}{c:c} \mathbf{A} & \mathbf{B} \\ \hdashline 0 & \mathbf{a} \end{array}\right] \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ 3.107$$

$$\mathbf{B}' = \left[\begin{array}{c} \mathbf{0}_{8\times 1} \\ \hdashline \mathbf{b} \end{array}\right] \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ 3.108$$

$$\mathbf{C}' = \left[\begin{array}{c:c} \mathbf{C} & \mathbf{0}_{3\times 1} \end{array}\right] \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ 3.109$$

where the new system, input and measurements matrices are **A**′, **B**′ and **C**′ respectively and the new state vector of the augmented state-space model is defined as:

$$\mathbf{X}' = \left[\begin{array}{cccccccc:c} \dot{h} & \dot{\alpha} & \dot{\beta} & h & \alpha & \beta & \ell_1 & \ell_2 & \beta_c \end{array}\right]^T \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ 3.110$$

The control input of the new state-space model of the complete active flutter suppression system incorporating actuator dynamics is an actuator position or angle β_a and no longer an actual commanded control surface angle β_c.

3.1.5 Simplified Aeroservoelastic Model

Although a full aeroelastic state-space model was developed, some simplifications were made to the model in light of the low anticipated flutter speed and inherent low structural damping of the flexure mount. Low structural damping implies that the viscous damping terms could be omitted from Equations 3.5, 3.6 and 3.7. The dynamics of the control surface were simplified by assuming that $k_\beta >> k_h$ and $k_\beta >> k_\alpha$ in Equations 3.5, 3.6 and 3.7. In practice this was valid and implied that coupling between the control surface mode and the pitch and plunge modes was negligible. In addition, the state-space flutter model could have been simplified even further by omitting the aerodynamic lag terms if their effects on the flutter dynamics of the model were seen to be negligible after initial wind tunnel tests. It would be valid to omit these terms because of the low flutter speed, hence the low reduced frequency of the model (Bisplinghoff *et al.*, 1957). However, in this work the only simplifications made to the aeroservoelastic state-space model were to simplify the control surface dynamics and omit the aerodynamic forces acting on the control surface. This was reasonable given that the primary dynamics of the control surface were a function of the actuator and actuation mechanism driving it. That is, assuming no control surface coupling with other modes, the equation describing the dynamics of the control surface (Equation 3.5) was substituted with a second order equation approximating the dynamics of the control surface actuator and actuation mechanism. The second order equation was required to ensure that the mass matrix of the aeroservoelastic system stayed invertible and non-singular. The second order equation used in this work was:

$$m_a\ddot{\beta}+c_a\dot{\beta}+k_a\beta=k_a\beta_c \quad \text{.......} \quad 3.111$$

where the subscript a denotes "actuator". Equation 3.111 can be expressed more intuitively as:

$$\ddot{\beta}+2\zeta_a\omega_a\dot{\beta}+\omega_a^2\beta=\omega_a^2\beta_c \quad \text{.......} \quad 3.112$$

if:

$$\left.\begin{aligned} 2\zeta_a\omega_a &\triangleq \frac{c_a}{m_a} \\ \omega_a^2 &\triangleq \frac{k_a}{m_a} \end{aligned}\right\} \quad \text{.......} \quad 3.113$$

With this assumption, the aeroservoelastic model of the binary flutter system was simplified by replacing Equation 3.7 with Equation 3.112 and modifying the state-space model of the system accordingly. This reduced the dimension of the state-space model (incorporating actuator dynamics) by one term and so in turn the computation effort required by the controller, which allowed faster feedback control. The structural and "aerodynamic" mass matrices of the simplified state-space model then became:

$$\mathbf{M}_s=\begin{bmatrix} m & mx_\alpha b & mx_\beta b \\ mx_\alpha b & I_\alpha & (c-a)b^2mx_\beta+I_\beta \\ 0 & 0 & m_a \end{bmatrix} \quad \text{.......} \quad 3.63$$

$$\mathbf{M}_a=\rho b^3\begin{bmatrix} -\frac{\pi}{b} & \pi a & T_1 \\ \pi a & -\pi b\left(\frac{1}{8}+a^2\right) & b\left[T_7+(c-a)T_1\right] \\ 0 & 0 & 0 \end{bmatrix} \quad \text{.......} \quad 3.64$$

The structural and “aerodynamic” damping matrices became:

$$\mathbf{D}_s = \begin{bmatrix} c_h & 0 & 0 \\ 0 & c_\alpha & 0 \\ 0 & 0 & c_a \end{bmatrix} \tag{3.114}$$

$$\mathbf{D}_a = \rho b^2 U \begin{bmatrix} -\frac{2\pi}{b} & -2\pi(1-a) & T_4 - T_{11} \\ 2\pi\left(\frac{1}{2}+a\right) & 2\pi ba\left(\frac{1}{2}-a\right) & b\left[T_8 - T_1 + (c-a)T_4 + aT_{11}\right] \\ 0 & 0 & 0 \end{bmatrix} \tag{3.115}$$

and the new simplified structural and “aerodynamic” stiffness matrices became:

$$\mathbf{K}_s = \begin{bmatrix} k_h & 0 & 0 \\ 0 & k_\alpha & 0 \\ 0 & 0 & k_a \end{bmatrix} \tag{3.116}$$

$$\mathbf{K}_a = \rho b^2 U^2 \begin{bmatrix} 0 & -\frac{2\pi}{b} & -\frac{2T_{10}}{b} \\ 0 & 2\pi\left(\frac{1}{2}+a\right) & 2aT_{10} - T_4 \\ 0 & 0 & 0 \end{bmatrix} \tag{3.117}$$

The system input matrix (Equation 3.104) and state vector (Equation 3.77) remained the same as for the full aeroservoelastic system without actuator dynamics, and the solution of the simplified state-space model followed that of the full state-space model. The simplified mathematical model described in this section was used during initial wind tunnel testing of the active flutter suppression wind tunnel model to allow a faster control system to be designed.

3.1.6 Open-Loop Aeroservoelastic Flutter Analysis

The method of performing an open-loop flutter analysis of the state-space model presented above is known as the *P* flutter analysis method (Hassig, 1971). The state-space system matrix in Equation 3.60 is a function of both airspeed *U* and air density ρ, so omitting the feedback control matrix, for an open-loop flutter analysis, enabled it to be written as:

$$\dot{\mathbf{X}}(t) = \mathbf{A}(U,\rho)\mathbf{X}(t) \tag{3.118}$$

The transfer function of the aeroelastic system governed by Equation 3.118 was obtained by calculating its Laplace transform. That is:

$$s\mathbf{X}(s) = \mathbf{A}(U,\rho)\mathbf{X}(s) \tag{3.119}$$

which once factorised, was simplified to:

$$\left[\mathbf{A}(U,\rho)-s\mathbf{I}\right]\mathbf{X}(s)=0 \quad \text{3.120}$$

The eigenvalues of **A** are the roots of the characteristic flutter equation (Abel, 1979) and were obtained from:

$$\mathbf{P}(U,\rho)=\det\left|\mathbf{A}(U,\rho)-s\mathbf{I}\right| \quad \text{3.121}$$

which, for a given airspeed and air density, has the complex solution (Turner, 1975):

$$\mathbf{P}=-\boldsymbol{\zeta}\boldsymbol{\omega}+i\boldsymbol{\omega} \quad \text{3.122}$$

where the terms $\boldsymbol{\zeta}$ and $\boldsymbol{\omega}$ are matrices of the modal decay rates and frequencies of the aeroelastic system, respectively. Thus:

$$\omega_j=imag\left(P_j\right) \quad \text{3.123}$$

$$g_{a,j}=-real\left(P_j\right)/\omega_j \quad \text{3.124}$$

The subscript j in Equations 3.123 and 3.124 corresponds to the j^{th} structural mode of the aeroelastic system. In this research j = 0, 1, 2.

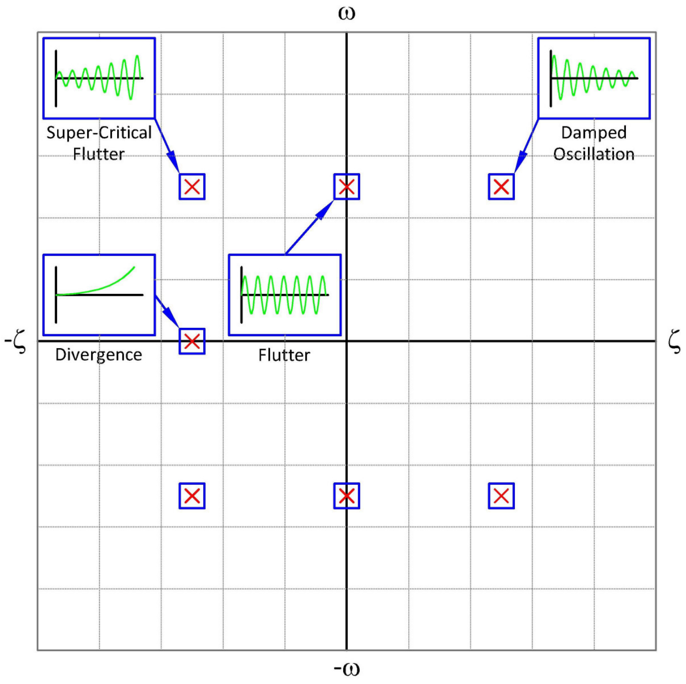


Figure 3.3 Aeroservoelastic Model Root Locus Diagram (Derived from Nachtigal, 1990)

The flutter speed and frequency of the system described by Equation 3.118 were determined by solving Equation 3.121 over the range of airspeeds in which flutter was expected. The modal dampings and frequencies of each mode were plotted as a function of airspeed. An interpolated curve fitted through these points and evaluated at zero damping gave the flutter speed of the aeroelastic model in the flutter mode whose damping passed through zero. It was also useful to plot the eigenvalues (**P**) of Equation 3.120 on a root locus diagram, to determine the aeroelastic behaviour of the system. Poles lying on the positive frequency axis correspond to critical flutter, whereas poles located on the negative decay rate axis correspond to torsional divergence (Rodden and Stahl, 1969). Poles in the positive frequency/negative decay rate region relate to supercritical flutter. This is shown schematically in Figure 3.3. The advantage of this eigenvalue flutter analysis method is that various system dynamics (such as actuator and sensor dynamics) can be accounted for, and their effect on the flutter characteristics of the complete aeroservoelastic system determined.

3.2 Feedback Control System Design

An overview of the feedback controller designed to suppress flutter of a binary pitch-plunge flutter model is given here. Section 3.2.1 presents the LQR controller used to calculate feedback gains and apply control inputs to the binary flutter system, and the Kalman filter used to estimate unmeasured system states used in the feedback control algorithm is described in Section 3.2.2. Combined, the LQR feedback controller and the Kalman state estimator formed the LQG flutter suppression controller used in this work, given in Section 3.2.3. For the LQG flutter controller to be effective, the system being modelled must inherently be both controllable and observable. That is, all the system state variables must be affected by the control input and able to affect the output of the system (Tewari, 2002), so methods of determining whether a system is controllable and observable are also briefly described in Section 3.2.4. Finally, discretisation of the LQG controller is outlined in Section 3.2.5. This was required to apply the continuous LQG controller to the physical system, which becomes non-continuous when implemented on a personal computer and data are acquired at a finite sampling rate.

3.2.1 Optimal Control Theory and the Linear Quadratic Regulator

A full explanation of optimal control theory is well beyond the scope of this work and unnecessary, so only a holistic overview and a method to design an appropriate optimal controller is described here. For a thorough treatise on optimal control theory, refer to Kirk (1970) and Bryson and Ho (1975). The idea of optimal control is to design the best possible control system for a given set of performance objectives. Typically the design of such a control system requires the minimum control energy (energy to generate control inputs) to satisfy the maximum overshoot and settling time constraints placed on it. This can be achieved by defining a system's transient energy as its total energy during a transient response, and the control energy as the energy required to generate a control input (Tewari, 2002). The maximum value of the transient energy of the system determines its maximum overshoot, whilst the time taken for the transient energy to decay to zero determines it's settling time. For optimal control it is then required to formulate an objective (cost) function that minimises both the control and transient energies of the system. The total control energy and transient energy of the system is determined by integrating expressions for these energies over a given time interval. The cost function for an optimal controller design is therefore the time integral of the sum of the control and transient energies (expressed as a function of time) of the system over a given control period (Tewari, 2002). One such optimal controller is the linear quadratic regulator that minimises the cost function J_{LQR} given by the infinite[10] time integral (Tewari, 2002):

$$J_{LQR} = \int_0^\infty \left[\mathbf{X}^T(t)\mathbf{Q}_{LQR}\mathbf{X}(t) + u^T(t)\mathbf{R}_{LQR}u(t) \right] dt \quad \text{......} \quad 3.125$$

of the state-space system (Equation 3.60, repeated here for convenience):

$$\left.\begin{aligned} \dot{\mathbf{X}} &= \mathbf{AX} + \mathbf{B}u \\ \mathbf{Y} &= \mathbf{CX} \end{aligned}\right\} \quad \text{......} \quad 3.60$$

that has the linear feedback control input calculated from:

$$u = -\mathbf{K}_{LQR}\mathbf{X} \quad \text{......} \quad 3.126$$

The control energy of the system is given by $u^T\mathbf{R}_{LQR}u$ and its transient energy by $\mathbf{X}^T\mathbf{Q}_{LQR}\mathbf{X}$. The square symmetric weighting matrices $\mathbf{R}_{LQR}$ and $\mathbf{Q}_{LQR}$ are defined as the *control cost* matrix and *state weighting* matrix respectively. The control cost matrix weights the control effort required to stabilise the system and the state weighting matrix weights the performance objective of the feedback controller. The objective of the linear quadratic regulator is then to solve for a feedback gain vector $\mathbf{K}_{LQR}$ that minimises the cost function (Equation 3.125) subject to the constraint that $\mathbf{X}(t)$ remains the solution to the system's state-space equation (Tewari, 2002). Methods of specifying the control cost and state weighting matrices to achieve this are given in Bryson and Ho (1975) and Kwakernaak and Sivan (1972), as summarised in Franklin and Powell (1980). In this research the cost function was solved and an appropriate feedback gain calculated using one of the LQR solvers in MATLAB™ (see Appendix C). Different control cost and state weighting matrices were used in the MATLAB™ routines until a controller with an appropriate response (one that created a balance between minimising the total control energy and total transient energy of the system) was achieved. That is, the maximum overshoots of the system states and the settling time of the controlled system were adjusted by selecting $\mathbf{R}_{LQR}$ and $\mathbf{Q}_{LQR}$ until a flutter suppression controller design that established a trade-off between the closed-loop performance of the system and its limitations, and met that the requirements listed in Section 1.3, was achieved. A block diagram of the linear quadratic regulator is shown in Figure 3.4 in which $\dot{\mathbf{X}}$ is denoted as **X'**.

---

[10] Since the steady-state response of the system is of interest, the integration limits are $t = 0$ to $t = \infty$.

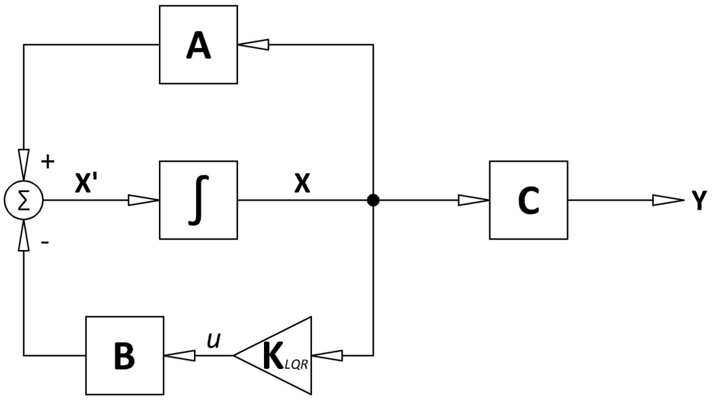


Figure 3.4 Linear Quadratic Regulator Block Diagram (Adapted from Chen, 1999)

3.2.2 State Estimation and the Kalman Filter

In Section 3.2.1 it was assumed that all state variables are available for the design of a LQR feedback controller. In practice this is often not the case. For instance, it may be impossible to measure certain system states, some modelled states may not be physical quantities or the cost of sensors and instrumentation is prohibitive. In such cases, state estimators (state observers) are used to estimate unmeasured system states to provide the full set of state estimates required to calculate feedback gains. Unmeasured system states can be estimated by observing the output of a system over a finite time interval, for a known input, and then reconstructing the state vector from the record of the output (Tewari, 2002). Only the basic mathematics used to implement the state estimator used in this work is included here. It was desired that a state estimate be made based on the matrices **A**, **B**, **C**, the system input *u* and the system output **Y** in the state-space equation of the system (Equation 3.60), as these are were the known parameters. Consequently, the original system of Equation 3.60 was duplicated as (Chen, 1999):

$$\dot{\hat{\mathbf{X}}} = \mathbf{A}\hat{\mathbf{X}} + \mathbf{B}u \quad \text{......} \quad 3.127$$

where $\hat{\mathbf{X}}$ is the vector of state estimates of the system states in **X**. Equation 3.127 alone could be used as an open-loop state estimator, but has the disadvantage that an initial state estimate must be calculated for each time step. More importantly, if the eigenvalues of the system matrix **A** have positive real parts, the differences between the state estimates and the actual system states increase with each time step calculation (Chen, 1999). To eliminate this, a closed-loop estimator was used where the system measurements **Y** = **CX** were compared with $\mathbf{C}\hat{\mathbf{X}}$. The difference between these two terms, multiplied by a constant gain vector **L,** was used as a correcting term to drive the differences between the actual and estimated states to zero. Mathematically, the closed-loop state estimate of the system described by Equation (3.60) is then given as (Chen, 1999):

$$\dot{\hat{\mathbf{X}}} = \mathbf{A}\hat{\mathbf{X}} + \mathbf{B}u + \mathbf{L}\left(\mathbf{Y} - \mathbf{C}\hat{\mathbf{X}}\right) \quad \text{......} \quad 3.128$$

which was factorised to:

$$\dot{\hat{\mathbf{X}}} = (\mathbf{A} - \mathbf{LC})\hat{\mathbf{X}} + \mathbf{B}u + \mathbf{LY} \quad \text{.......... } 3.129$$

This is shown as a block diagram in Figure 3.5 (in which $\hat{\mathbf{X}}$ is denoted as $\mathbf{X}_e$ and $\dot{\hat{\mathbf{X}}}$ as $\mathbf{X'}_e$). It is apparent from Equation 3.128 that if the measured and estimated states are equal, the closed-loop estimator becomes an open-loop estimator (Equation 3.127) that estimates the actual system states exactly.

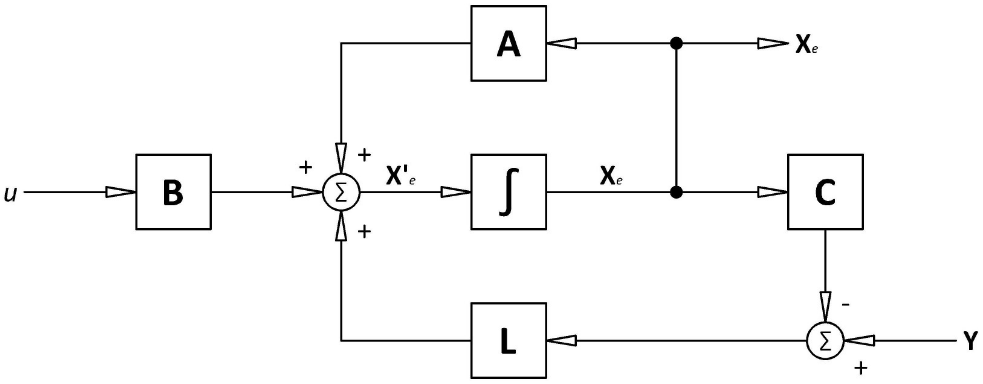


Figure 3.5 State Observer Block Diagram (Adapted from Chen, 1999)

The performance of the closed-loop estimator was determined by defining an estimation error vector as:

$$\mathbf{e} \triangleq \mathbf{X} - \hat{\mathbf{X}} \quad \text{.......... } 3.130$$

The estimation error of the state observer should reduce to zero in a steady state (Tewari, 2002). The dynamics of the estimation error of the observer were investigated by differentiating Equation 3.130 with respect to time and substituting Equations 3.60 and 3.129 into the resulting expression (Equation 3.131) to give the estimation error rate as a function of the state-space matrices (Equation 3.132).

$$\dot{\mathbf{e}} = \dot{\mathbf{X}} - \dot{\hat{\mathbf{X}}} \quad \text{.......... } 3.131$$

$$\dot{\mathbf{e}} = \mathbf{AX} + \mathbf{B}u - \left[(\mathbf{A} - \mathbf{LC})\hat{\mathbf{X}} + \mathbf{B}u + \mathbf{LY}\right] \quad \text{.......... } 3.132$$

Equation 3.132 was simplified and factorised to:

$$\dot{\mathbf{e}} = (\mathbf{A} - \mathbf{LC})\left(\mathbf{X} - \hat{\mathbf{X}}\right) \quad \text{.......... } 3.133$$

which from Equation 3.130 is equivalent to:

$$\dot{\mathbf{e}} = (\mathbf{A} - \mathbf{LC})\mathbf{e} \quad \text{3.134}$$

Equation 3.134 is the governing equation of the estimation error (Chen, 1999). The rate at which the estimation error decays to zero, or equivalently the rate at which the estimated states approach measured states, is governed by the eigenvalues of [**A**-**LC**] (Chen, 1999). By carefully selecting the observer gain, a suitable closed-loop estimator can be designed. There is no need to calculate the initial system states as the estimated states approach the actual system states rapidly, so the initial state estimate can be assigned arbitrarily. However, initial state estimates should be made as close to the actual states as possible to minimise initial transient errors and speed up the response of the observer. With regard to this study, only three states *viz*. h, α and β, were measured. The remaining five states were estimated with an optimal state observer (Kalman filter).

Optimal Estimation and the Kalman Filter

As with the closed-loop control system and the LQR optimal controller gain, an optimal observer gain vector can be determined for a given state estimator. In addition to it not always being possible to measure all system states, most physical systems cannot be modelled accurately using only a deterministic model because of the presence of modelling uncertainties (process noise) and uncertain state measurements (measurement noise) (Tewari, 2002). An optimal observer that is able to both estimate unmeasured system states and account for process and measurement noise is the Kalman filter (Kalman, 1960). In this work, only the mathematics and method of implementing a constant, linear, time-invariant Kalman filter are presented. To account for external disturbances and measurement errors, the state-space equations of the system given by Equation 3.60 were modified to (Tewari, 2002):

$$\left.\begin{aligned}\dot{\mathbf{X}} &= \mathbf{AX} + \mathbf{B}u + \mathbf{G}w \\ \mathbf{Y} &= \mathbf{CX} + v\end{aligned}\right\} \quad \text{3.135}$$

where *w = f(t)* is the process noise vector which may arise due to modelling errors, in this application through neglecting the effect of turbulence and nonlinear or high frequency dynamics of the system. The term *v = f(t)* is the measurement noise vector which accounts for errors in the measured signals (Tewari, 2002). It was assumed that both of these noise vectors are white[11] (Gaussian random) noise vectors, such that (Welch and Bishop, 2006):

$$p(w) \sim N(0, \mathbf{Q}_{Kalman}) \quad \text{3.136}$$

$$p(v) \sim N(0, \mathbf{R}_{Kalman}) \quad \text{3.137}$$

---

[11] White noise is defined as a signal with zero mean and a normal probability distribution (Teweri, 2002).

where (Franklin and Powell, 1980)[12]:

$$E\left\{ww^T\right\}=\mathbf{Q}_{Kalman} \quad 3.138$$

$$E\left\{vv^T\right\}=\mathbf{R}_{Kalman} \quad 3.139$$

Cross-correlation between the process and measurement noise was assumed to be zero, implying that system disturbances were uncorrelated with measurement noise. The term $\mathbf{Q}_{Kalman}$ in Equation 3.138 is the process noise covariance matrix, whilst $\mathbf{R}_{Kalman}$ in Equation 3.139 is the measurement noise covariance matrix (Welch and Bishop, 2006). The design objective of the Kalman filter is to establish a balance between the confidence in the mathematical model of the system and the confidence in the measured system states by appropriately defining the $\mathbf{Q}_{Kalman}$ and $\mathbf{R}_{Kalman}$ matrices. The optimal Kalman filter aims to minimise the covariance of the estimation error (given by Equation 3.130) in the presence of process and measurement noise (Tewari, 2002). That is, the Kalman filter minimises (Welch and Bishop, 2006):

$$\mathbf{P}_{Kalman}=E\left\{\left(\mathbf{X}-\hat{\mathbf{X}}\right)\left(\mathbf{X}-\hat{\mathbf{X}}\right)^T\right\} \quad 3.140$$

which is equivalent to:

$$\mathbf{P}_{Kalman}=E\left\{\mathbf{e}\mathbf{e}^T\right\} \quad 3.141$$

The optimal observer gain (Kalman gain in this case) is given by:

$$\mathbf{L}_{Kalman}=\mathbf{P}_{Kalman}\mathbf{C}^T\mathbf{Q}^{-1}_{Kalman} \quad 3.142$$

which satisfies the constraints of Equation 3.143 and Equation 3.144, assuming the process and measurement noises are stationary (Murray, 2008).

$$\dot{\mathbf{P}}=\mathbf{A}\mathbf{P}_{Kalman}+\mathbf{P}_{Kalman}\mathbf{A}^T-\mathbf{P}_{Kalman}\mathbf{C}\mathbf{Q}_{Kalman}\mathbf{C}\mathbf{P}_{Kalman}+\mathbf{G}\mathbf{R}_{Kalman}\mathbf{G}^T \quad 3.143$$

$$\mathbf{P}_{Kalman}\left(0\right)=E\left\{\mathbf{X}\left(0\right)\mathbf{X}^T\left(0\right)\right\} \quad 3.144$$

For the active flutter suppression wind tunnel model of this work, the measurement noise covariance matrix was expressed as:

---

[12] The notation $E\left\{\mathbf{X}\right\}$ is the expected value operator and is equal to $\sum_{i=1}^{n} p_i x_i$ for *n* probable outcomes $x_1 \ldots x_n$ and corresponding probabilities $p_1 \ldots p_n$ (Welch and Bishop, 2006).

$$\mathbf{R}_{Kalman} = \begin{bmatrix} \sigma_h^2 & 0 & 0 \\ 0 & \sigma_\alpha^2 & 0 \\ 0 & 0 & \sigma_\beta^2 \end{bmatrix} \quad \text{3.145}$$

where σ_h, σ_α and σ_β represent the standard deviation of the model's measured wind-off plunge displacement, pitch angle and control surface position signals respectively. In this research, where a well defined and accurate mathematical model was developed to characterise the aeroservoelastic system, the Kalman filter designed was robust to fairly large measurement uncertainties. Consequently, the terms in Equation 3.145 were specified to be larger than they actually were. The process noise covariance matrix ($\mathbf{Q}_{Kalman}$) was judged based on the confidence in the mathematical model of the system after initial open-loop wind tunnel testing. As with the LQR controller gain described in Section 3.2.1, the Kalman gain used in the control system implementation were determined using the Control System Toolbox in MATLAB™ (see Appendix C).

3.2.3 Linear Quadratic Gaussian Control

In this section the optimal regulator (Section 3.2.1) and the optimal estimator (Section 3.2.2) are combined to form an optimal compensator. These components were simply combined such that the control system feedback gain was computed from the system state estimates obtained from the Kalman filter (Equation 3.129), and not the actual system states (Equation 3.126). Mathematically:

$$u = -\mathbf{K}_{LQR}\hat{\mathbf{X}} \quad \text{3.146}$$

which results in Equation 3.127 becoming:

$$\dot{\hat{\mathbf{X}}} = \left(\mathbf{A} - \mathbf{L}_{Kalman}\mathbf{C} - \mathbf{B}\mathbf{K}_{LQR}\right)\hat{\mathbf{X}} + \mathbf{L}_{Kalman}\mathbf{Y} \quad \text{3.147}$$

as shown in block diagram form in Figure 3.6 where $\dot{\mathbf{X}}$ is denoted as **X'**, $\hat{\mathbf{X}}$ as $\mathbf{X}_e$ and $\dot{\hat{\mathbf{X}}}$ as $\mathbf{X'}_e$. The combined LQR regulator and Kalman filter observer is referred to as a linear quadratic Gaussian compensator (Tewari, 2002). Although the LQR controller was designed assuming the true state **X** of the system was available for feedback, the estimated state $\hat{\mathbf{X}}$ was substituted for **X**, without having to redesign the LQR controller. This can be proved by using $\hat{\mathbf{X}}$ in place of **X** in Equation 3.60 and substituting *u* with $-\mathbf{K}_{LQR}\hat{\mathbf{X}}$ from Equation 3.126 to give:

$$\dot{\mathbf{X}} = \mathbf{A}\mathbf{X} - \mathbf{B}\mathbf{K}_{LQR}\hat{\mathbf{X}} \quad \text{3.148}$$

which was written in terms of the state estimation error (Equation 3.130) as:

$$\dot{\mathbf{X}} = \mathbf{A}\mathbf{X} - \mathbf{B}\mathbf{K}_{LQR}\left(\mathbf{X} - \mathbf{e}\right) \quad \text{3.149}$$

The expression that models the dynamics of the estimation error (Equation 3.134) was combined with Equation 3.149 to form the two coupled equations which described the complete behaviour of the LQG compensator (Franklin and Powell, 1980). Written in augmented matrix form (once simplified):

$$\begin{bmatrix} \dot{\mathbf{X}} \\ \dot{\mathbf{e}} \end{bmatrix} = \begin{bmatrix} \mathbf{A}-\mathbf{BK}_{LQR} & \mathbf{BK}_{LQR} \\ 0 & \mathbf{A}-\mathbf{LC} \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ \mathbf{e} \end{bmatrix} \quad \text{..} \; 3.150$$

which has the characteristic equation (from the Laplace transform of Equation 3.150):

$$\det\left|\begin{bmatrix} \mathbf{A}-\mathbf{BK}_{LQR} & \mathbf{BK}_{LQR} \\ 0 & \mathbf{A}-\mathbf{LC} \end{bmatrix} - \lambda\mathbf{I}\right| = 0 \quad \text{...} \; 3.151$$

As Equation 3.151 is block triangular, it was simplified to the product of the two determinants (Franklin and Powell, 1980):

$$\left(\det\left|\left[\mathbf{A}-\mathbf{BK}_{LQR}\right]-\lambda\mathbf{I}\right|\right)\left(\det\left|\left[\mathbf{A}-\mathbf{L}_{Kalman}\mathbf{C}\right]-\lambda\mathbf{I}\right|\right) = \alpha_r\alpha_e = 0 \quad \text{.......................} \; 3.152$$

The functions α_r and α_e in Equation 3.152 are respectively the independent characteristic equations of the state regulator and estimator. That is, the roots of the combined system are a sum of the estimator and regulator roots. Consequently, the LQR regulator and Kalman filter (estimator) can be designed separately and then combined to form a complete LQG compensator. This ability to independently design a regulator and estimator and then combine them to form a compensator is known as the *separation principle* (Chen, 1999). In the design of the LQG compensator it was important to select the poles of the Kalman filter to be much faster than those of the LQR regulator. This was to ensure that the estimation error was minimised as quickly as possible so that the initial (inaccurate) state estimates did not cause the LQR controller to be ineffective initially.

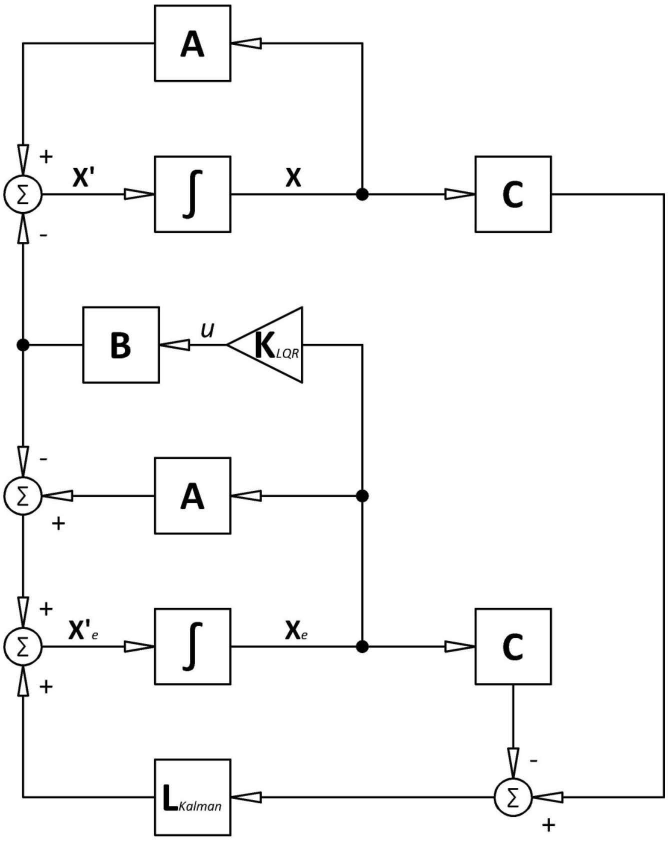


Figure 3.6 Linear Quadratic Gaussian Compensator Block Diagram (Adapted from Chen, 1999)

3.2.4 Controllability and Observability

The effectiveness of the LQG compensator had to be determined before its practical implementation. For a LQR controller to be effective the system it regulates must be controllable, whilst the Kalman filter can only be effective in estimating system states if the system is observable. Controllability refers to the ability to move any system state in a desired direction using a suitable control signal (Franklin *et al.*, 1991). Mathematically, a system is controllable if the matrix $\mathbf{C}_o$, defined as:

$$\mathbf{C}_o \triangleq \begin{bmatrix} \mathbf{B} & \mathbf{AB} & \mathbf{A}^2\mathbf{B} & \dots & \mathbf{A}^{n-1}\mathbf{B} \end{bmatrix} \quad \text{.. 3.153}$$

has full rank. The controllability matrix was obtained in MATLAB™ using the "`rank(ctrb(B, C))`" command in the Control System Toolbox. Observability refers to the ability to deduce system states from only the measured output of a system (Franklin *et al.*, 1991). Mathematically, a system is observable if the matrix $\mathbf{O}_b$, defined as:

$$\mathbf{O}_b \triangleq \begin{bmatrix} \mathbf{C} & \mathbf{CA} & \mathbf{CA}^2 & \dots & \mathbf{CA}^{n-1} \end{bmatrix}^T \quad \text{.. 3.154}$$

has full rank. The observability matrix was obtained in MATLAB™ using the "`rank(obsv(A, C))`" command in the Control System Toolbox.

3.2.5 Digital Control Systems

The derivations of the LQR controller and Kalman filter assume continuous operation, i.e. that when calculating the dynamic feedback gains and state estimates there is no time gap between one instance of a system state and the next. In practice it is impossible to have a continuous compensator because of the time constants of the transducers used to measure system states and the finite sampling rate. Consequently, a discrete (digital) controller was used to account for the time gap between one time step calculation and the next (Franklin *et al.*, 1991). The state-space model of the aeroservoelastic system was discretised using the Control System Toolbox in MATLAB™ with the command:

```
[A_d, B_d, C_d, D_d] = c2d(ss([A, B, C, D]), 1/f_s)
```

where $\mathbf{A}_d$, $\mathbf{B}_d$, $\mathbf{C}_d$ and $\mathbf{D}_d$ represent the discrete state-space matrices calculated from the corresponding continuous matrices **A**, **B**, **C** and **D** in Equation 3.60 and f_s is the anticipated sample and control frequency of the physical control system. The control system design and calculation of the LQR controller and Kalman filter gains then proceeded using the discrete state-space model of the aeroservoelastic system.

3.3 Conclusion

The structural equations governing the dynamics of a fluttering aerofoil and the associated physics of the unsteady aerodynamic forces driving the flutter have been described in this chapter. The structural and unsteady aerodynamic equations including actuator dynamics were combined to form a full aeroservoelastic model to describe the system. Simplifications to the aeroservoelastic model to reduce the system states and minimise control system hardware overheads were suggested for actual implementation of the active flutter suppression controller on the wind tunnel model. A holistic overview of feedback controllers and state estimators was given. It was decided to use a LQR regulator to suppress flutter and a Kalman filter to estimate unmeasured system states. Together, these form the combined LQG active flutter suppression compensator which was described. It was shown via the separation principle that the LQR controller and Kalman observer could be designed independently, without either of their dynamics being adversely influenced when combined into the LQG compensator. Controllability and observability of the system was discussed to illustrate that the LQG controller is theoretically capable of suppressing flutter. Finally, digital control, whereby a continuous state-space system is discretised to account for finite sample and calculation periods of a real control system, was briefly discussed.

Chapter 4

Aeroservoelastic Wind Tunnel Model

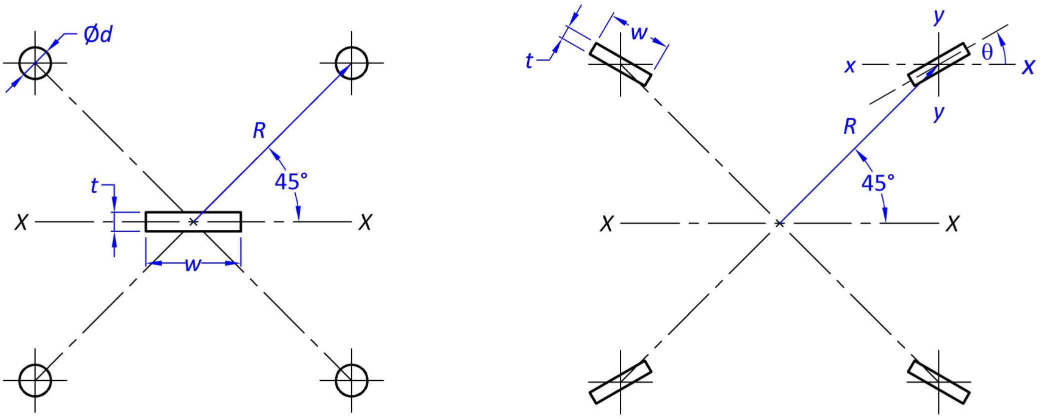
An outline of the design and complexities of the various components (mechanical, electrical and software), of the active flutter suppression wind tunnel model is given in this chapter. The complete system was designated as the **F**lutter with **A**ctive **S**uppression **T**echnology (FAST) wind tunnel model. The mechanical design of a flexure mount and wing is outlined first in Section 4.1, followed by a description of the electronic components and circuitry in Section 4.2. Section 4.3 discusses the hardware used to acquire data and drive the control surface actuator. Finally, the main features of software developed to simulate the open- and closed-loop response of the FAST model, apply feedback control to it, and post process wind tunnel data are given in Section 4.4.

4.1 Mechanical Design

The mechanical design of the FAST model consisted of the design of a flexure mount system, a rigid wing incorporating a movable control surface and the control surface actuation mechanism. The flexure mount system was designed to have structural dynamic characteristics which allowed gentle and well defined binary flutter of the rigid wing to be demonstrated.

4.1.1 Flexure Mount Design

In this work a new flexure mount design was developed based on Farmer's (1982) design (Figure 4.1 (a)) using *rectangular* instead of round flexures. This eliminated the need for the central drag strut, made the small size mount feasible and provided more freedom to tailor the pitch and plunge frequencies of the wing/mount system. A cross-section through the new mount design is shown in Figure 4.1(b). This design is thought to be novel as no similar implementation could be found in the literature. An earlier full parametric analysis (Sutherland, 2006) established the effect of the number of flexures N and their respective thickness t, width w, orientation θ, length L and placement R, for a given flexure material, on the flutter dynamics of the complete model. In this research the number of equally spaced flexures and their respective thickness, width and length were kept fixed. Only the flexure angles θ were variable, for frequency tuning during wind tunnel testing. The new flexure mount comprises four flat flexures, shown and numbered in Figure 4.2, that permit pure pitch and plunge motion through differential bending, as in the original design of Farmer (1982). The pitch and plunge modal frequencies of the flutter model are adjusted within set frequency ranges by independently changing the angles of each of the flexures to one of four discrete positions. The flexures are locked at the chosen angle with circular keys. The mass and mass inertia of the mount are changed by changing and/or repositioning sliding masses on the ballast arms (see Figure 4.16).



(a) Traditional Flexure Mount (Farmer, 1982) **(b)** FAST Model Flexure Mount

Figure 4.1 Cross-Section of Traditional and FAST Model Flexure Mounts ($N = 4$)

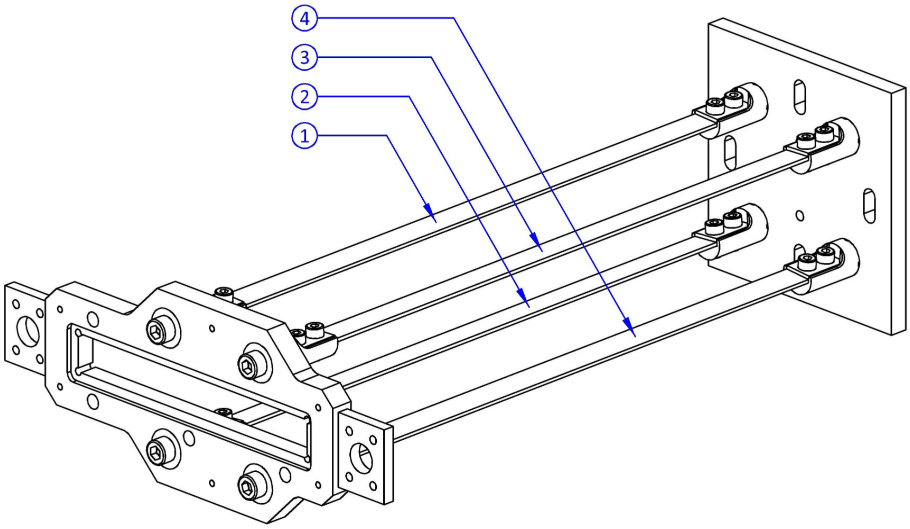


Figure 4.2 FAST Model Flexure Mount

Flexure Mount Design Equations

The FAST model flexure mount is described in detail in Sutherland (2006; 2008), but for convenience and completeness the design equations are repeated here. These equations relate the geometric and material properties of the flexures to the pitch and plunge frequencies of the complete flutter model. Referring to Figure 4.1(b) the stiffness of the plunge degree of freedom of the mount is given by Equation 3.155 and that of the pitch degree of freedom by Equation 3.156, that is (Roark and Young, 1975):

$$k_h = \frac{12EI_{xx}}{L^3} \quad \text{......} \quad 3.155$$

$$k_\alpha = N\frac{GJ}{L} + \frac{12EI_R}{L^3}R^2 \quad \text{......} \quad 3.156$$

The polar second moment of area *J* in Equation 3.156 is given by (Roark and Young, 1975):

$$J = wt^3\left(\frac{1}{3} - 0.21\left(\frac{t}{w}\right)\left[1 - \frac{1}{12}\left(\frac{t}{w}\right)^4\right]\right) \quad \text{3.157}$$

and the area moments I_{xx}, I_{yy} and I_R by (Roark and Young, 1975):

$$I_{xx} = \frac{wt}{12}\left[\left(w\sin\theta\right)^2 + \left(t\cos\theta\right)^2\right] \quad \text{3.158}$$

$$I_{yy} = \frac{wt}{12}\left\{\left[w\sin\left(90^\circ - \theta\right)\right]^2 + \left[t\cos\left(90^\circ - \theta\right)\right]^2\right\} \quad \text{3.159}$$

$$I_R = \frac{wt}{12}\left\{\left[w\sin\left(45^\circ - \theta\right)\right]^2 + \left[t\cos\left(45^\circ - \theta\right)\right]^2\right\} \quad \text{3.160}$$

The natural plunge and pitch frequencies of the binary flutter model with mass *m* and mass inertia I_α about its elastic axis, are then given by (Farmer, 1982):

$$f_h = \frac{1}{2\pi}\sqrt{\frac{k_h}{m}} \quad \text{3.161}$$

$$f_\alpha = \frac{1}{2\pi}\sqrt{\frac{k_\alpha}{I_\alpha}} \quad \text{3.162}$$

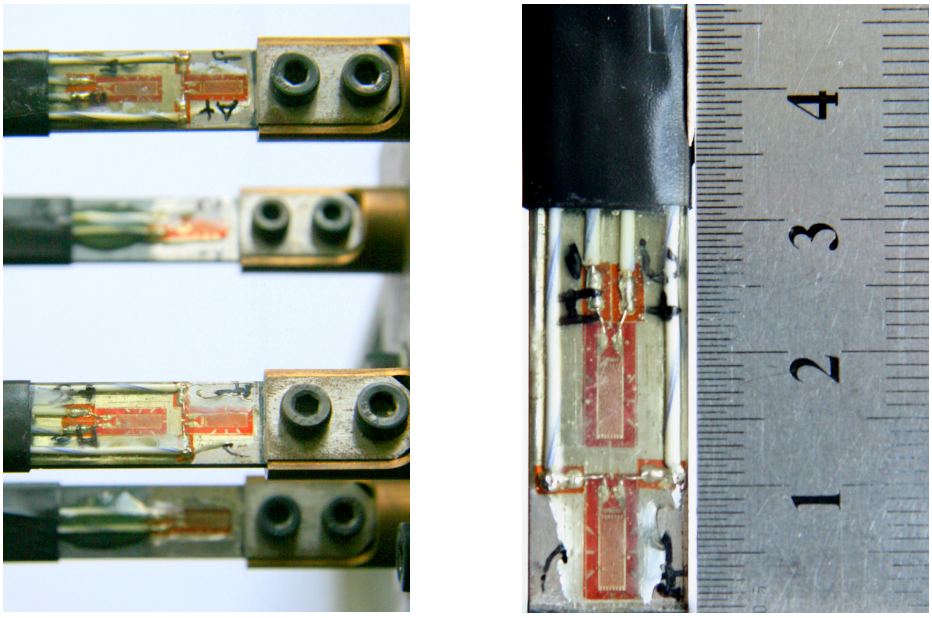
The final parameters of the flexure mount, obtained from a parametric analysis of the above equations, are listed in Table 4.1.

Table 4.1 FAST Model Flexure Mount Parameters

Parameter	**Value**	**Units**
Number of flexures (*N*)	4	-
PCD of flexures (2*R*)	70.7	*mm*
Effective flexure length (*L*)	270	*mm*
Flexure width (*w*)	12	*mm*
Flexure thickness (*t*)	1.5	*mm*
Flexure angles (θ)	2.5, 3.5, 4.5, 5.5	*deg.*
Flexure material	AISI Class 1 heat treated ground flat stock tool steel	-

Plunge Deflection and Pitch Angle Measurements

To measure the plunge displacement and pitch angle of the FAST model wing, needed for the design of the flutter suppression control law, the flexure mount was instrumented with two full Wheatstone strain gauge bridges. The SGBs were placed at the root of the FAST model mount (see Figure 4.3) to sense maximum strain. Parameters of the strain gauges used are listed in Table 4.2.



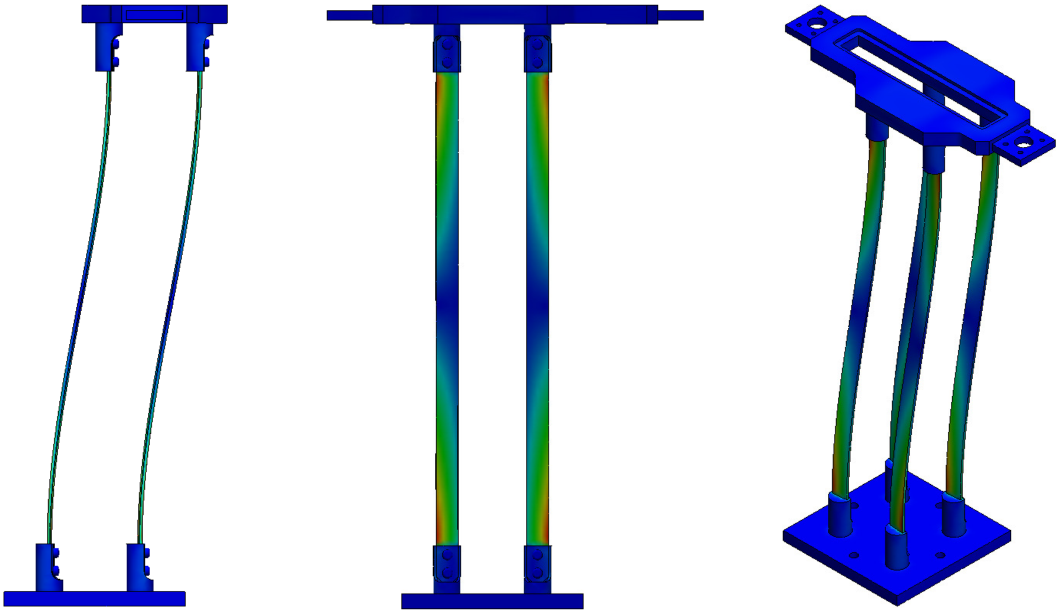
(a) Flexure Mount Strain Gauges **(b)** Strain Gauge Detail

Figure 4.3 Strain Gauges Bonded to the FAST Model Mount Flexures

Table 4.2 FAST Model Strain Gauge Parameters

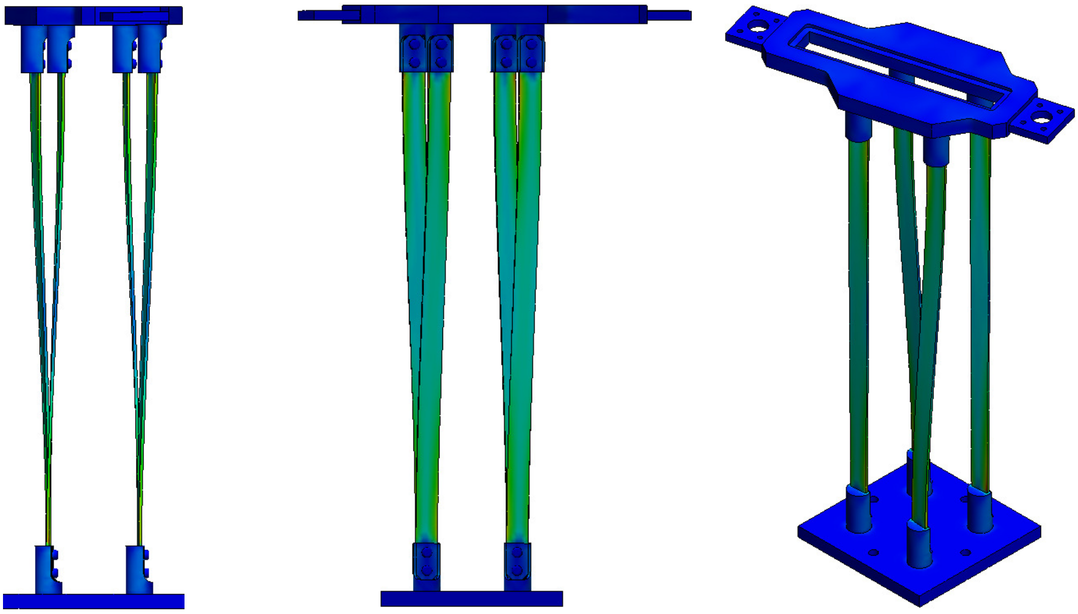
Parameter	**Value**	**Units**
Manufacturer	Kyowa	-
Gauge Factor (24°C, 50% RH)	2.10 ± 1.0%	-
Gauge Length	5	*mm*
Gauge Resistance (24°C, 50% RH)	346 ± 0.6	Ω

A simple finite element analysis was performed to determine the deflection shapes of the mount when subjected to a pure bending (Figure 4.4) or a pure torsion (Figure 4.5) load. This was done to establish how the SGBs should be located and connected to ensure that their outputs remained isolated from each other during pure bending or pure torsion of the mount.



(a) Side View **(b)** Top View **(c)** Isometric View

Figure 4.4 FAST Model Mount Bending Mode Deflection and Stress Distribution



(a) Side View **(b)** Top View **(c)** Isometric View

Figure 4.5 FAST Model Mount Torsion Mode Deflection and Stress Distribution

Based on the finite element analysis, the pitch SGB was set up to measure differential bending in the flexures (rather than the actual torsion of each flexure) so it would output a null voltage during pure bending of the FAST model mount.

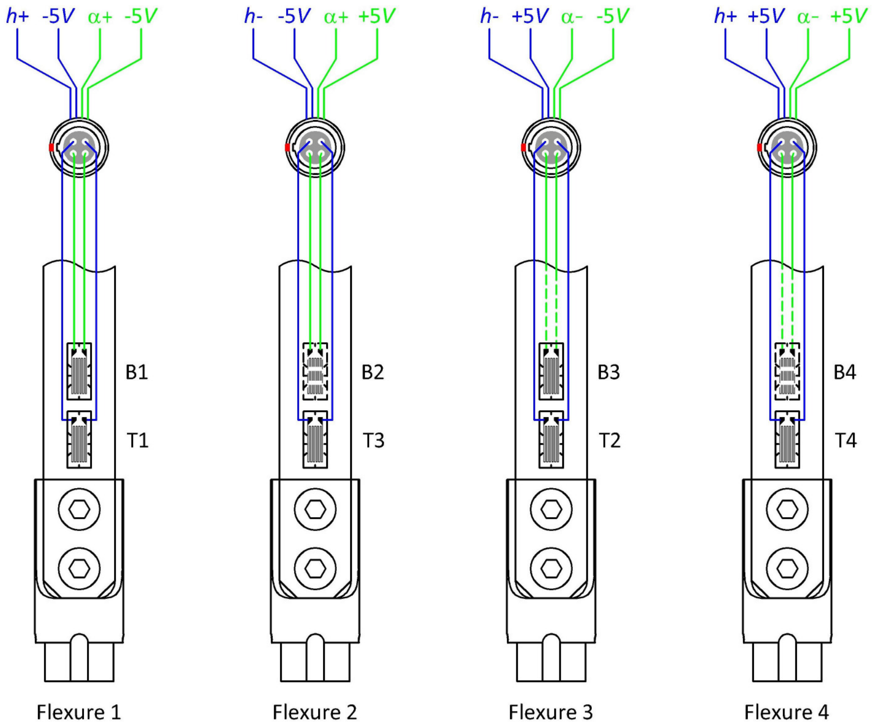
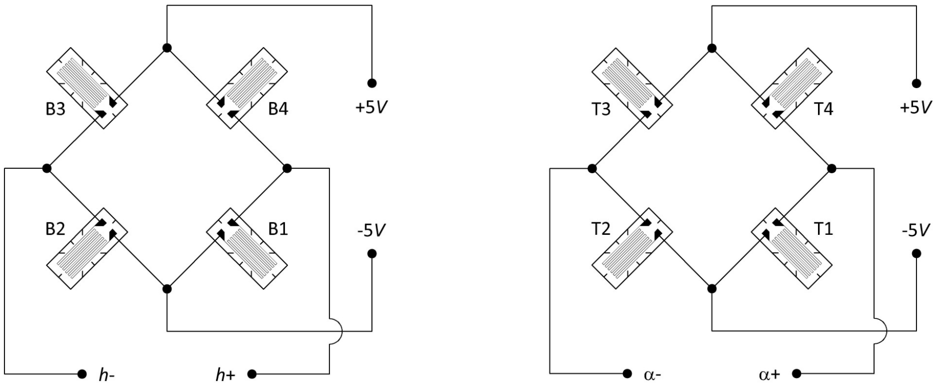


Figure 4.6 Flexure Plunge and Pitch Strain Gauge Locations

The locations of the strain gauges on the flexures are shown in Figure 4.6 and the connections of the plunge and pitch SGBs in Figure 4.7(a) and Figure 4.7(b), respectively. The dotted lines in Figure 4.6 indicate strain gauges on the underside of the flexures.



(a) Plunge Strain Gauge Bridge **(b)** Pitch Strain Gauge Bridge

Figure 4.7 Plunge and Pitch Strain Gauge Bridge Connection Diagram

To show how the SGBs are isolated from each other in this arrangement, the output voltage of a full strain gauge bridge is given by (Window and Holister, 1982):

$$V = k_{SGB}\left(\varepsilon_1 - \varepsilon_2 + \varepsilon_3 - \varepsilon_4\right)V_S \quad \text{3.163}$$

where ε_i is the strain experienced by the i^{th} strain gauge and V_S is the SGB supply voltage. Referring to the bending modeshape of the mount shown in Figure 4.4 and assuming equal strain in each strain gauge, the two SGB outputs for pure bending in one direction are:

$$\left.\begin{aligned} V_h &= k_{SGB}\left(\varepsilon_{\mathrm{B1}}^{(C)} - \varepsilon_{\mathrm{B2}}^{(T)} + \varepsilon_{\mathrm{B3}}^{(C)} - \varepsilon_{\mathrm{B4}}^{(T)}\right)V_S \\ V_h &= k_{SGB}\left[\varepsilon_{\mathrm{B1}}^{(C)} - \left(-\varepsilon_{\mathrm{B2}}^{(C)}\right) + \varepsilon_{\mathrm{B3}}^{(C)} - \left(-\varepsilon_{\mathrm{B4}}^{(C)}\right)\right]V_S \\ V_h &= 4k_{SGB}\varepsilon^{(C)}V_S \end{aligned}\right\} \quad \text{3.164}$$

$$\left.\begin{aligned} V_\alpha &= k_{SGB}\left(\varepsilon_{\mathrm{T1}}^{(C)} - \varepsilon_{\mathrm{T2}}^{(C)} + \varepsilon_{\mathrm{T3}}^{(C)} - \varepsilon_{\mathrm{T4}}^{(C)}\right)V_S \\ V_\alpha &= 0 \end{aligned}\right\} \quad \text{3.165}$$

where the $\varepsilon^{(T)}$ terms represent tensile strain and the $\varepsilon^{(C)}$ terms represent compressive strain. Similarly, referring to the torsion modeshape of the mount shown in Figure 4.5 and again assuming equal strain in each strain gauge, the two SGB outputs for pure torsion of the model mount in one direction are:

$$\left.\begin{aligned} V_h &= k_{SGB}\left(\varepsilon_{\mathrm{B1}}^{(T)} - \varepsilon_{\mathrm{B2}}^{(C)} + \varepsilon_{\mathrm{B3}}^{(C)} - \varepsilon_{\mathrm{B4}}^{(T)}\right)V_S \\ V_h &= 0 \end{aligned}\right\} \quad \text{3.166}$$

$$\left.\begin{aligned} V_\alpha &= k_{SGB}\left(\varepsilon_{\mathrm{T1}}^{(T)} - \varepsilon_{\mathrm{T2}}^{(C)} + \varepsilon_{\mathrm{T3}}^{(T)} - \varepsilon_{\mathrm{T4}}^{(C)}\right)V_S \\ V_\alpha &= k_{SGB}\left[\varepsilon_{\mathrm{T1}}^{(T)} - \left(-\varepsilon_{\mathrm{T2}}^{(T)}\right) + \varepsilon_{\mathrm{T3}}^{(T)} - \left(-\varepsilon_{\mathrm{T4}}^{(T)}\right)\right]V_S \\ V_\alpha &= 4k_{SGB}\varepsilon^{(T)}V_S \end{aligned}\right\} \quad \text{3.167}$$

It is therefore apparent that the plunge SGB outputs its maximum signal and the pitch SGB outputs a null signal when the mount experiences pure translation, and *vice versa* for pure rotation. The actual Wheatstone bridge connections were made on a custom designed printed circuit board (PCB) described in Section 4.2. In practice, interference between the pitch and plunge SGBs arising from slight misalignments of the strain gauges and unaccounted for torsion in the flexures was calibrated out. This ensured both SGBs gave the desired measurements accurately, even for coupled pitch-plunge motion.

4.1.2 FAST Model Wing Structure Design

A new wing model with a NACA0012 aerofoil section incorporating a trailing edge control surface was designed to bolt to the flexure mount. The design is similar to that of a full-scale aircraft wing in that it is a semi-monocoque structure (once assembled) with spars, ribs and a load carrying skin. This design was chosen to allow the control surface actuator and instrumentation to be fitted inside the wing, out of the airflow. The internal structure of the wing comprises aluminium and carbon fibre ribs, and carbon fibre spars. Machined aluminium fittings were bonded onto the ends of the carbon fibre spars and each spar was attached to the wing root and wing tip aluminium sections with machine screws. The internal structure is shown in Figure 4.8. A one-piece carbon fibre skin fitted over the whole framework, fixed in several places to the aluminium ribs with brass shear pins, to provide additional stiffness and the aerodynamic shape.

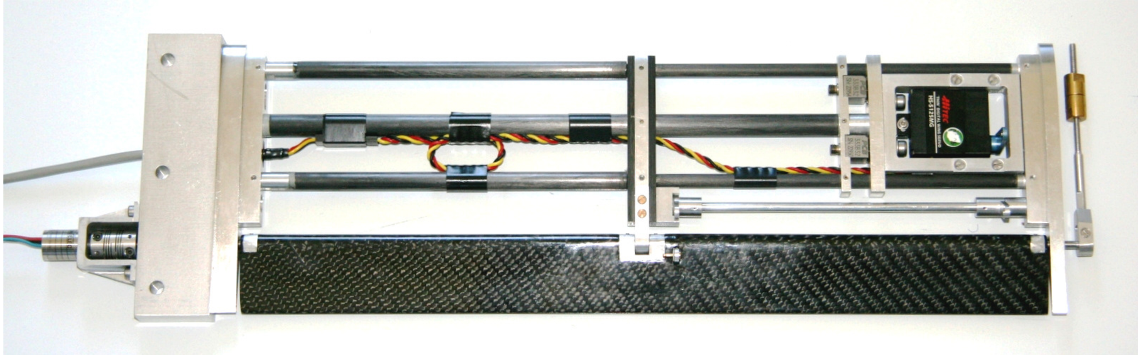


Figure 4.8 FAST Model Wing Internal Structure and Movable Control Surface

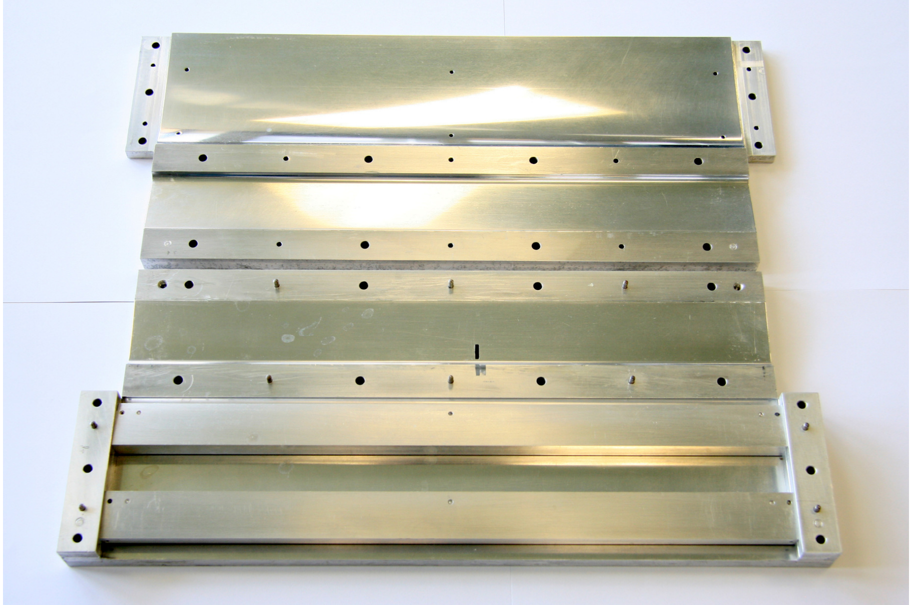


Figure 4.9 Wing and Control Surface Aluminium Mould and Inner Inserts

The wing skin was made from four layers of uni-directional carbon fibre layed up at +90°/+45°/-45°/-90° in the aluminium mould shown in Figure 4.9. A 3.0*mm* thick carbon shear web towards the trailing edge of the wing skin and a carbon D-Section at its leading edge provided additional stiffness. Aluminium inserts were required in the skin mould to correctly locate these components to ensure that the finished skin slid correctly over the internal wing structure. The mould incorporated accurately positioned holes to guide the drilling of holes in the cured upper and lower skin surfaces, whilst still in the mould. This was necessary to precisely align the holes in the wing skin with the shear pin positions in the wing ribs.

4.1.3 Model Control Surface Design and Actuator Selection

A full-span control surface with a chord length of 25.0% of the wing chord was incorporated in the FAST model wing to actively suppress flutter, by altering the unsteady aerodynamic forces. It comprised a carbon fibre skin, carbon fibre and aluminium ribs and a carbon fibre rod for the rear spar. The control surface skin was made from four layers of uni-directional carbon in a +90°/+45°/-45°/-90° lay-up. A ϕ4.0*mm* full-span steel shaft acted as both the front spar and the hinge axis of the control surface. This shaft slid into the leading edge of the control surface and was fixed in place with grub screws through the machined aluminium end and centre ribs. It rotates in brass bushes in the wing root fitting, wing tip fitting and wing centre rib.

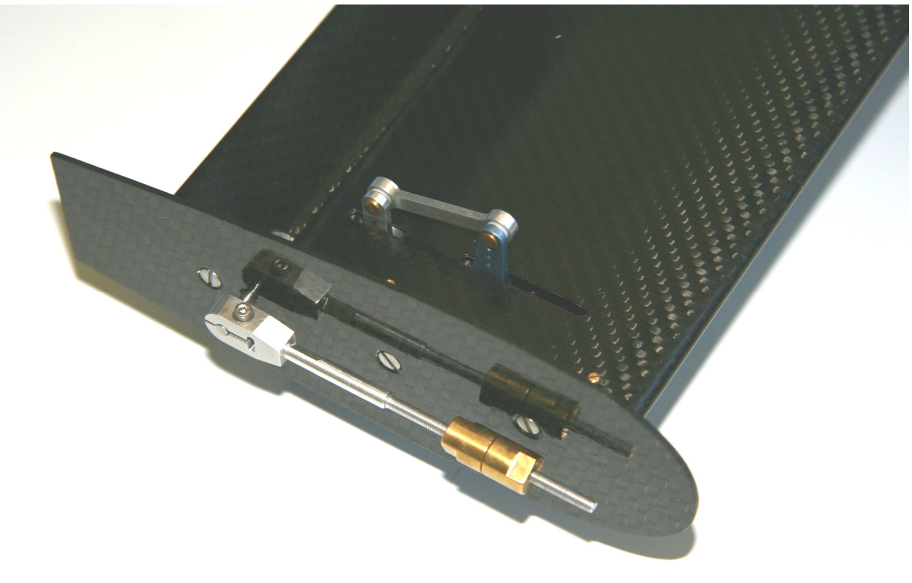
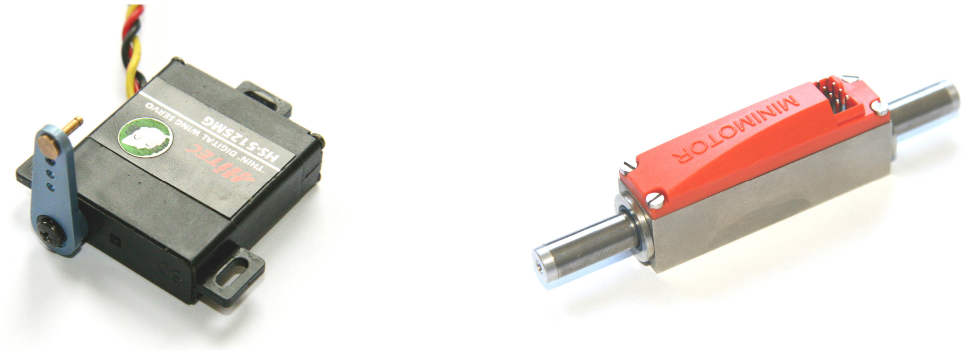


Figure 4.10 FAST Model Control Surface Ballast Arm and Movable Ballast Mass

Careful alignment and assembly of these components ensured that the control surface rotated freely. An aluminium control horn was built into the control surface to connect it to the actuation system. An adjustable ballast mass (Figure 4.10) clamped to the control surface shaft could change the centre of mass of the control surface and be used in conjunction with a torsion spring to tune the flutter characteristics of the control surface itself. For this work though the ballast mass was positioned such that the centre of mass of the control surface lay on its rotation axis.

FAST Model Control Surface Actuators and Drive Mechanisms

Initially it was planned to use a model aircraft servo to drive the control surface. This approach is not new and has been adopted by Block and Gilliatt (1997) and Predoiu *et al*. (2007) for example. A Hitec™ HS-5125 digital model aircraft servo, shown in Figure 4.11(a), was chosen because of its small size (low profile), ease of control and relatively good quoted performance characteristics. The original actuation system was designed around this servo.



(a) Hitec™ HS-5125 Digital Servo **(b)** Faulhaber™ LM1247-020-01 Actuator

Figure 4.11 FAST Model Control Surface Actuators

This actuation system included a torsion spring between the servo and control surface control horn, to enable future investigation into the dynamics of the control surface and its own flutter characteristics. The servo was connected to the torsion spring with a custom aluminium control rod using 1.5*mm* ball bearings in the rod-ends. The torsion spring was connected to the control surface control horn in a similar manner. Small (ϕ1*mm*) circlips were used to retain the pivot pins. The Hitec™ HS-5125 servo and the original control surface actuation system are shown in Figure 4.12. Once this actuation system was assembled, initial testing showed that in fact the servo had limited performance. This was attributed to servo dead-band, the large time constant of the servo, free-play in the gears and the relatively high control surface inertia, all of which introduced a significant time lag into the control system. Although the Hitec™ HS-5125 servo was probably powerful enough to *prevent* flutter with an appropriate controller design, it is unlikely that it was powerful enough to suppress a developed flutter cycle. In view of this, and to prevent having to design an active flutter suppression controller with limited performance, the decision was made to use a more powerful actuator instead. This also provided additional scope for future control algorithm testing. A Faulhaber™ 1247-020-01 linear DC-Servomotor (Figure 4.11(b)) was chosen because of its high performance in terms of load carrying capability and acceleration. This required modification of the FAST model, since the larger size and linear motion of this actuator required it to be mounted outside the wing.

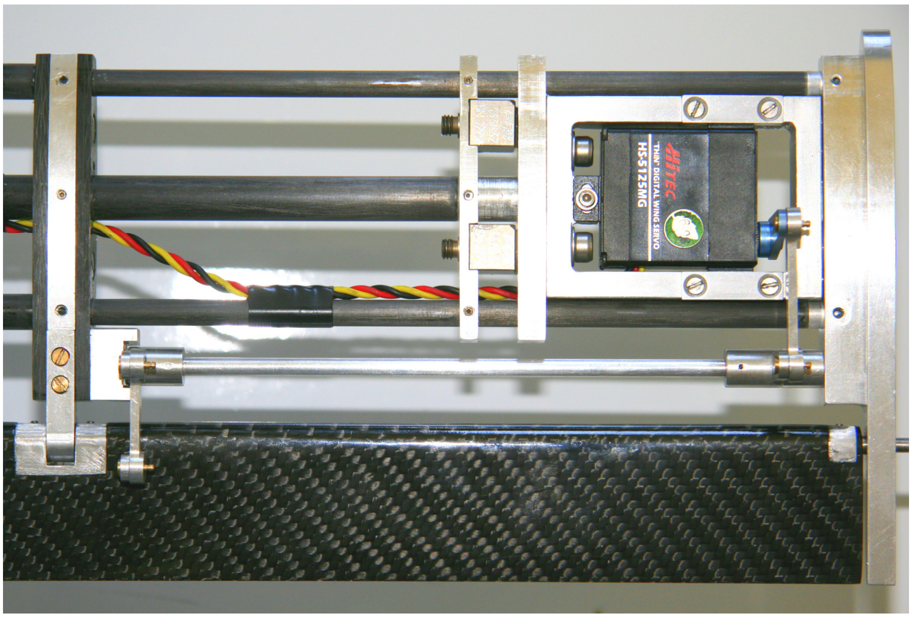
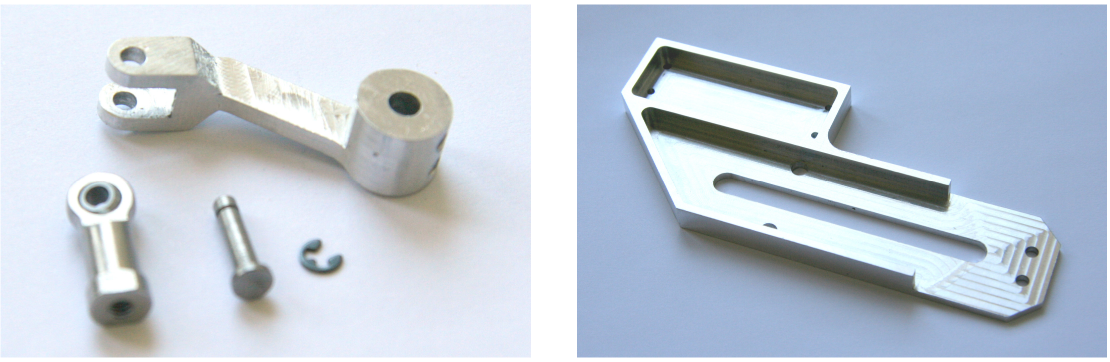


Figure 4.12 Initial Control Surface Actuation Mechanism

A new mount bracket for the actuator was designed, manufactured and fitted to the existing FAST model wing root mount plate. A clevis, miniature rod-ends and crank connected the actuator to the control surface shaft. The crank was held in place on the control surface shaft by a grub screw bearing on a flat. The linkage components and actuator mounting bracket are shown in Figure 4.13(a) and (b) respectively, and the assembled system as fitted to the FAST model is shown in Figure 4.14.



(a) Actuator Crank and Components

(b) Actuator Mounting Bracket

Figure 4.13 Control Surface Crank and Faulhaber™ LM1247-020-01 Actuator Mounting Bracket

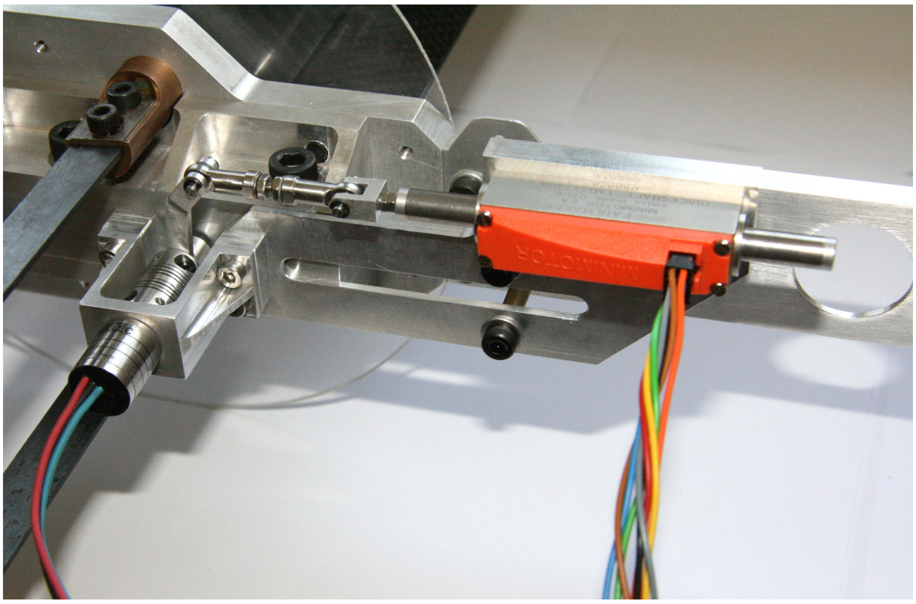


Figure 4.14 Control Surface Actuation Mechanism as used in Wind Tunnel Testing

Control Surface Angle Measurement

A Contelec™ Vert-X 13 rotary sensor attached to the control surface shaft was used to measure control surface angle. This sensor uses the Hall-effect principle to measure angles from 0.0° to 360.0° with 12 bit (0.09°) resolution. It has very low rotational friction and so did not resist the motion of the control surface. The sensor was attached to the control surface shaft by a flexible torsion coupling and to the FAST model by a U-shaped bracket as seen in Figure 4.14 and Figure 4.15. The flexible coupling tolerated some misalignment between the control surface and transducer shafts by being able to bend, but remained stiff in torsion. The sensor was shielded with braided metal sheathing, connected to the common ground of the instrumentation, to minimise electromagnetic interference from the nearby linear actuator.

Figure 4.15 Control Surface Angle Contelec™ Vert-X 13 rotary sensor and Flexible Coupling

Summary of FAST Model Mount and Wing Structure

Figure 4.16 shows the complete, assembled FAST wind tunnel mount and model. The geometric parameters of the model are listed in Table 4.3.

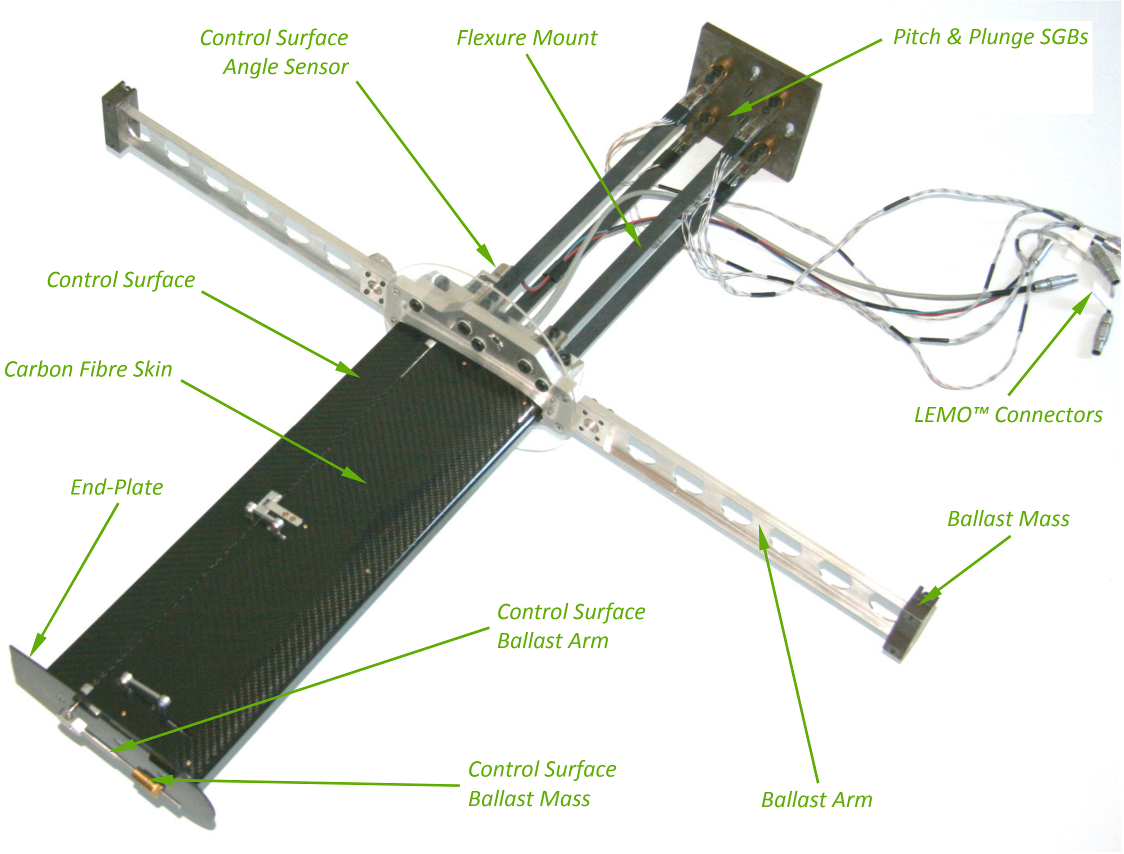


Figure 4.16 Complete FAST Model and Mount

Table 4.3 FAST Model Wing Parameters

FAST Model Wing Parameter	**Value**	**Units**
Wingspan	360	*mm*
Chord	120	*mm*
Wing Aspect Ratio	3	-
Flap Span	346	*mm*
Flap Chord	30	*mm*
Wing Taper Ratio	1	-
Wing Twist	0	*deg.*
Wing Aerofoil Section	NACA0012	-

4.2 Electrical System Design

A simple electronic circuit was designed and manufactured to amplify and balance signals from the plunge displacement and pitch angle SGBs. The circuit diagram is shown in Figure 4.17. Two AD620 instrumentation amplifiers (manufactured by Analog Devices™) powered directly from the ±12.0V supply voltage were used for this purpose. From the AD620 data sheet, the amplifier gain was calculated as:

$$G_A = \frac{49400}{R_G} + 1 \ldots\ldots\ldots\ldots 3.168$$

Referring to Figure 4.17 the 250Ω variable gain resistor in the pitch SGB amplifier was set to 50Ω which gave a gain of 989 on the pitch signal. In the plunge SGB amplifier the 250Ω variable gain resistor was set to 75Ω for a gain of 660 on the plunge signal. The strain gauges on each flexure were connected directly to the PCB via LEMO™ plugs and screw terminals and both Wheatstone bridge circuits were completed on the PCB.

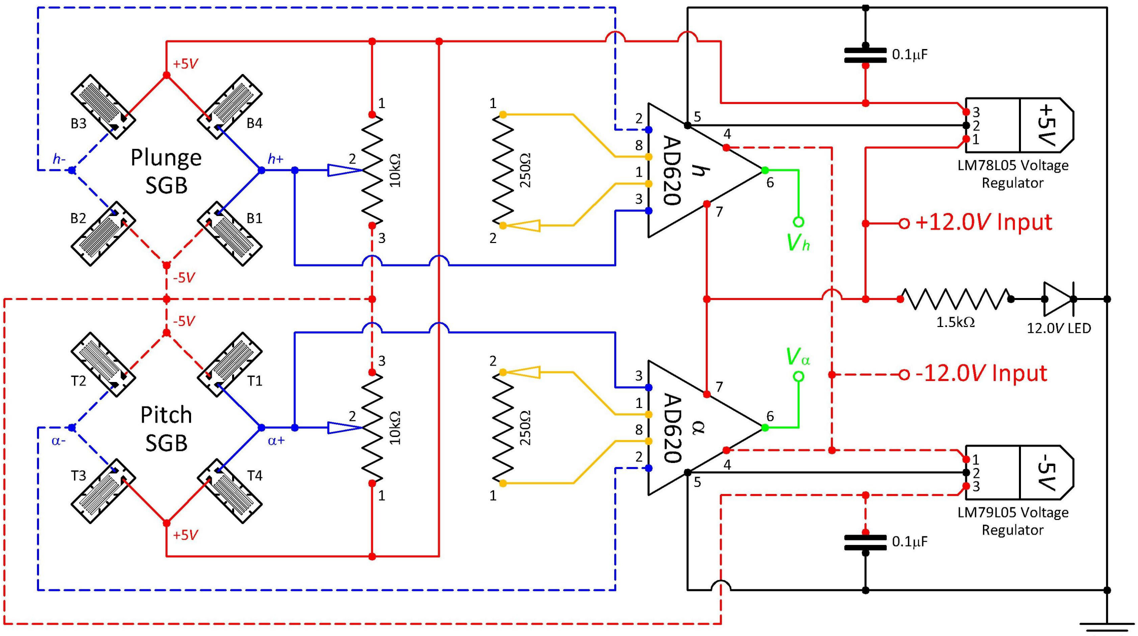
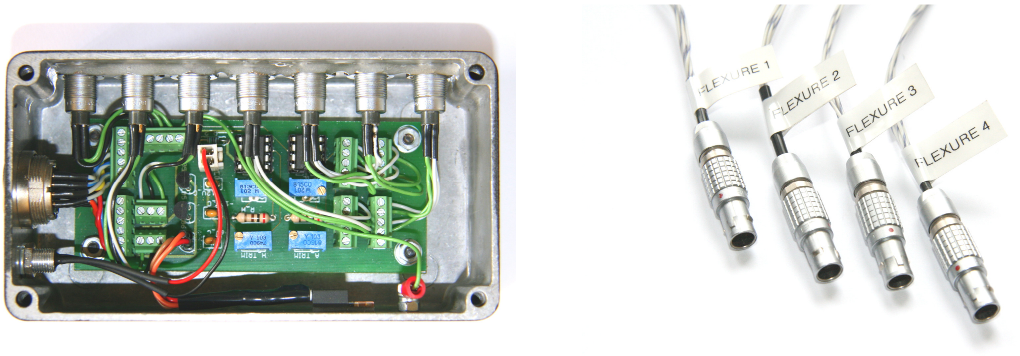


Figure 4.17 FAST Model Strain Gauge Bridge Amplifier Circuit Diagram

The two 10kΩ screw type potentiometers shown in Figure 4.17 were required to balance (or trim) the plunge and pitch Wheatstone bridges by changing their DC offset voltages so that each SGB output a null voltage when the model/mount was in its trim position. The circuit also included the means to regulate the ±12.0V supply to ±5.0V to power the two SGBs and the Contelec™ Vert-X 13 sensor. The PCB was housed in an aluminium box (Figure 4.18 (a)) which acted as a Faraday cage and minimised electrical noise on the strain gauge and Contelec™ Vert-X 13 sensor signals.

LEMO™ connectors (Figure 4.18 (b)) were used to connect all the strain gauges, Contelec™ Vert-X 13 sensor and the original Hitec™ servo to the circuit board.



(a) SGB Amplifier Circuit Board Design

(b) LEMO™ Connectors

Figure 4.18 FAST Model Electronics and Instrumentation Connectors

4.3 Data Acquisition and Instrumentation

The FAST wind tunnel model instrumentation included a ±12.0*V* power supply, a data acquisition system and the Faulhaber™ MCLM 3006 S motion controller, each of which are described in more detail below. The main ±12.0*V* power supply was regulated to ±5.0*V* to power the pitch and plunge SGB amplifiers, the pitch and plunge SGBs and the control surface angle sensor. The DAQ was used to acquire the model's response signals and to drive the control surface actuator through the Faulhaber™ MCLM 3006 S motion controller.

Figure 4.19 FAST Model NI USB-6211 DAQ and Instrumentation System

4.3.1 Data Actuation System

A National Instruments™ NI USB-6211 16-Bit, 250kS/s, M Series Multifunction, USB Bus Powered DAQ (Figure 4.19) was used to acquire data from the FAST model transducers and drive the control surface actuator. It can be configured to have sixteen single-ended (non-referenced or referenced) or eight differential analogue input channels and has four digital input channels, four digital output channels, two analogue output channels and two onboard 32-bit counters. One of the onboard counters, together with one of the digital output channels, was configured to generate the PWM signal required to actuate the Hitec™ HS-5125 servo. Similarly, one of the analogue output voltage channels was used to drive the Faulhaber™ linear actuator that was configured to operate in its analogue input mode.

4.3.2 Model Instrumentation Power Supply System

The ±12.0V power supply was housed in an aluminium box with an integrated cooling fan, as shown in Figure 4.20. The FAST model transducer output signal and control surface actuator signal cables were connected to the DAQ via internal wiring in the aluminium box. These signals were separated to minimise interference between the two, which might otherwise have caused excessive noise on the DAQ signals. Provision was also made in the aluminium box to allow for two accelerometer signals (via BNC cables) to be routed to the DAQ and two rubber tubes to be fitted to the front of the box to be connected to a pressure transducer inside the box, for dynamic pressure, and hence wind tunnel speed, measurements. In this work, the wind tunnel speeds were obtained directly from the wind tunnel instrumentation, so an independent pressure transducer was not required.

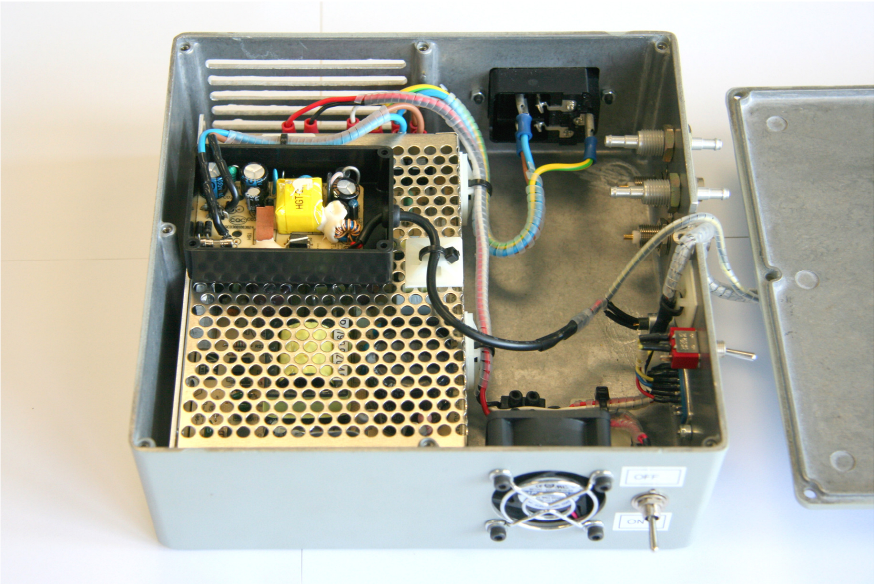


Figure 4.20 FAST Model Power Supply and Instrumentation System Internal Wiring

4.3.3 Control Surface Actuator Controller

The Faulhaber™ 1247-020-01 linear DC-Servomotor used to actuate the FAST model control surface has its own proportional integral derivative (PID) controller, designated the MCLM3006S linear actuator motion controller, and shown in Figure 4.21. The proportional, integral and differential gains of the controller, as well as other controller and actuator parameters, could be changed using proprietary software. It was important that these values be programmable as the actuator had to be tuned, along with the actuation system, to represent the transfer function (Equation 3.112) modelled in the FAST model LQG controller design. The controller was set up in its analogue input mode to accept an external voltage as a control input. In this mode, a +10.0*V* or -10.0*V* input moved the actuator to its maximum deflections in the positive or negative directions respectively. The position of the actuator arm varies linearly as a function of input voltage over its full 20.0*mm* stroke. All parameters and operating modes were downloaded and saved to the controller hardware, so it did not need to be connected to the computer during wind tunnel testing.

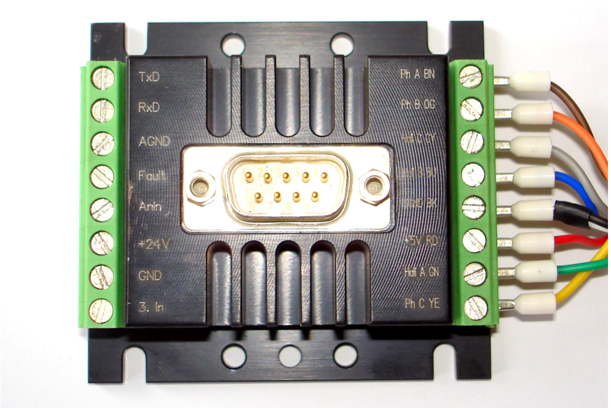


Figure 4.21 Faulhaber™ MCLM 3006 S Linear Actuator Motion Controller

4.4 Software Development

Software for various purposes was written during this research, these being to:

- Model the open- and closed-loop dynamics of the FAST wind tunnel model
- Evaluate the response of the active flutter suppression controller to various inputs
- Implement the LQG controller on the FAST wind tunnel model
- Post-process simulation and wind tunnel data

This software was written using MathWorks™ MATLAB™/Simulink™ and Microsoft Visual™ C# (2008 Express Edition). MATLAB™ was used to develop the state-space model of the complete closed-loop flutter system to perform a linear analysis, and to post-process and analyse simulation and wind tunnel data. This is outlined in Section 4.4.1. A complete block diagram model of the flutter system that was developed in Simulink™ to determine the effects of nonlinearities in the system that were not included in the MATLAB™ state-space model is discussed in Section 4.4.2. The software used to acquire data, calculate feedback gains and control inputs and drive the control surface actuator through the DAQ was written in C# and is discussed in Section 4.4.3.

4.4.1 MATLAB™ Software

State-Space MATLAB™ Model

A complete state-space model was developed in MATLAB™ to model the flutter dynamics of the FAST model and active flutter suppression controller. A full listing of this code and a brief description of each of the functions is included in Appendix C. Only the main features of the code are described here. Most of the controller design and mathematical modelling of the system was done using this MATLAB™ model. The MATLAB™ code was written to calculate modal parameters and the open- and closed-loop responses of the FAST model, and has the following features:

- A routine to set up the state-space matrices (given in Section 3.1.3) of a binary flutter system at a given airspeed and air density. Aerofoil parameters were read from a separate MATLAB™ file
- A routine to perform an eigenvalue flutter analysis over a range of airspeeds for a given air density. This determined the modal damping and frequency of the flutter system as a function of airspeed, and was used to predict the flutter speed of the system in either its open- or closed-loop configuration
- An ordinary differential equation (ODE) solver to calculate the time domain response of the flutter system in either its open- or closed-loop configuration
- A function to calculate the LQR feedback gain and Kalman filter matrices. Weightings were adjusted in this function so that different feedback gain matrices could be calculated and their effect on flutter suppression evaluated
- Various functions to plot the simulated data, for example modal damping and frequencies as a function of airspeed

MATLAB™ Data Analysis Software

Software to analyse the FAST model simulation and wind tunnel data was written in MATLAB™, a listing of which is included in Appendix C. The main features of the data analysis software are:

- A function to apply a window to the time series data to minimise spectral leakage when calculating the Fast Fourier Transforms (FFTs) and PSDs
- A routine to calculate the FFTs and PSDs of the time series data, required to estimate the modal frequencies and dampings of each structural mode of the FAST model
- A peak finding algorithm to search for peaks within the FFTs or PSDs of the time series data. The search method and the number of peaks to be located could be specified. The “curve fit off” method finds the data point nearest to the peak. The “curve fit on” option

fits a quadratic function through the three data points nearest the peak and then calculates the peak amplitude and frequency from this fit

- A routine to calculate the width at half the amplitude of the PSD peaks of the time series data, to enable use of the half power bandwidth method for on-line flutter speed predictions during wind tunnel testing
- Various functions to plot the calculated FFTs, PSDs, peak widths and peak frequencies

4.4.2 Simulink™ Software

A Simulink™ model of the whole aeroservoelastic system was developed to verify the MATLAB™ eigenvalue solution of the state-space model. Nonlinearities such as free-play and dead-band in the control surface actuation mechanism, which were not included in the MATLAB™ state-space model, were included in the Simulink™ model. Using this model, the effect of these nonlinearities on the flutter dynamics and control effectiveness of the controller was investigated with relative ease. Simulink™ block diagrams representing the components of the full aeroservoelastic system are described and given in Appendix D.

4.4.3 Microsoft Visual™ C# Software

The data acquisition and control software used to apply control to the FAST model was written in Microsoft Visual™ C# 2008 Express Edition. This compiler was chosen because it is freely available, has built in database functionality, supports multi-threaded applications and is relatively easy to use. The design requirements of the software were to acquire data from the National Instruments™ DAQ, calculate Kalman state estimates, apply feedback gains and calculate control inputs, drive the control surface actuator, plot measured FAST model parameters and log all data to file. Multi-threading was required to enable all components of the program to run simultaneously at the relatively high frequencies required. These main components of the software and the way in which they interact are shown in the flow chart in Figure 4.22. In this figure, the “Control” thread is shown in blue, the “Record Data” thread in green and the “Plot Data” thread in red. The software has a graphical user interface with four tabs for the different operations, and various menu items with options to set parameters or perform functions.

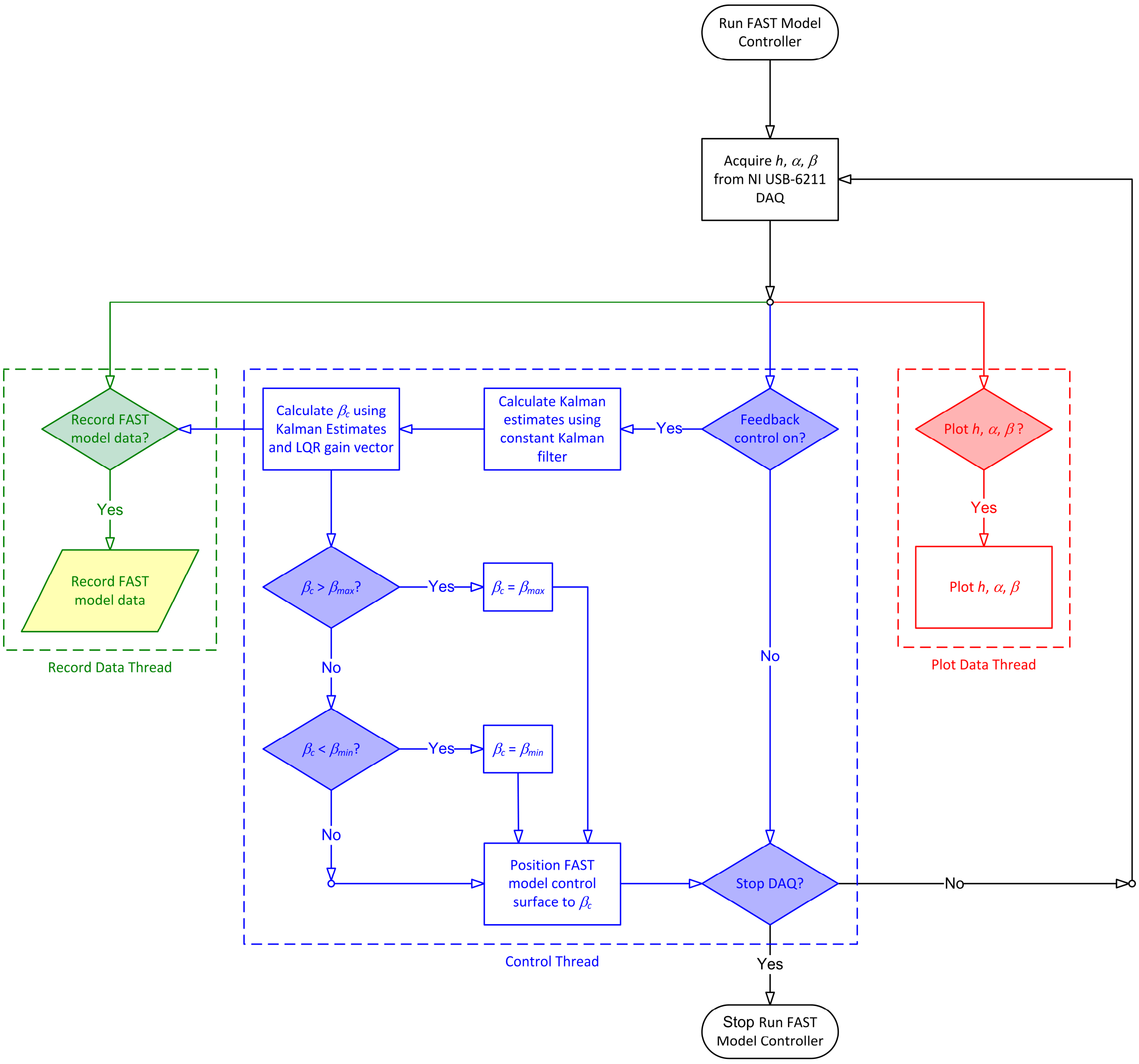


Figure 4.22 FAST Model Controller Software Overview Flowchart

The FAST model controller software featured the following main components[13]:

- A “Flutter Control” tab that shows primary FAST model data, *viz*. model plunge displacement, pitch angle and control surface deflection, recorded during wind tunnel testing. Controls to display and plot wind tunnel data, to apply feedback control and to reset the controller are also featured on this tab. When the “Feedback ON” checkbox is checked the control algorithm is activated and the control surface is actively controlled. Checking the “Plot data” checkbox in the “Flutter Control” tab opens a MATLAB™ figure in which data is plotted in real-time. When the “Record Data” checkbox is checked real-time data is recorded to a “~.csv” file. The “Reset Controller” button resets the Kalman filter estimates to a null vector and the commanded control input to zero. The control frequency (the frequency at which the feedback gains and control input are

---

[13] See Appendix E for screenshots of these components.

calculated, and at which the control surface is moved) can be adjusted in the "Control Freq." spinbox

- A "Flap Control" tab, on which are controls and settings to manually adjust the control surface position by commanding either a specific angle or an actuator signal (analogue voltage or PWM signal). This is done by setting values in spinboxes on the tab or by dragging a trackbar to manually move the control surface. The "Pulse Type" groupbox has radio buttons that are used to specify a single or continuous PWM signal when a model aircraft servo is being used to drive the control surface (digital servos require a single pulse for control, whilst older analogue servos require a continuous pulse train). The fine adjustment checkbox on this tab allows the control surface to be controlled in small increments with the trackbar
- A "Flutter Data" tab where recorded data is displayed in a table for quick reference, to give an indication of the data without having to open the data file. The user selects a menu item to populate the table with data from the last wind tunnel run, retrieved from memory
- A "Status Dialogue" tab that displays a history of all user inputs and other important data which can be saved to a file for later reference. For example, the tab lists a history of which checkboxes were checked, where data was saved, what control surface actuator settings were chosen and what the data acquisition parameters (start time, stop time, sample period, sample frequency, etc.) were for a given test run
- Multi-threading to allow the various components of the software to run simultaneously to achieve the required flutter suppression control system speed. A record data worker is used to record data, a control worker to calculate the Kalman state estimates and feedback control gains and position the control surface and a plot worker to plot the data in a separate MATLAB™ figure. Priority is given to the control thread to ensure that flutter will be prevented or suppressed when necessary, even while other software processes are running
- A SQL database to which important FAST model and control system data is written. The database was implemented to save any software settings or FAST model parameters (e.g. calibration and control system data) that changed during wind tunnel testing and to ensure that these parameters are available each time the FAST Model Controller software or computer is restarted
- A bottom tool strip, configured as a status bar to indicate if the DAQ has been initialised, if a link to MATLAB™ has been established, if feedback control is on or off and if the DAQ is running (shown by a pulsing progress bar)

Menu items (some of which can also be accessed via buttons in the software's top tool strip) provide the following:

- Options to save recorded data to a comma separated value (~.csv) file and the command feed listing to a text (~.txt) file

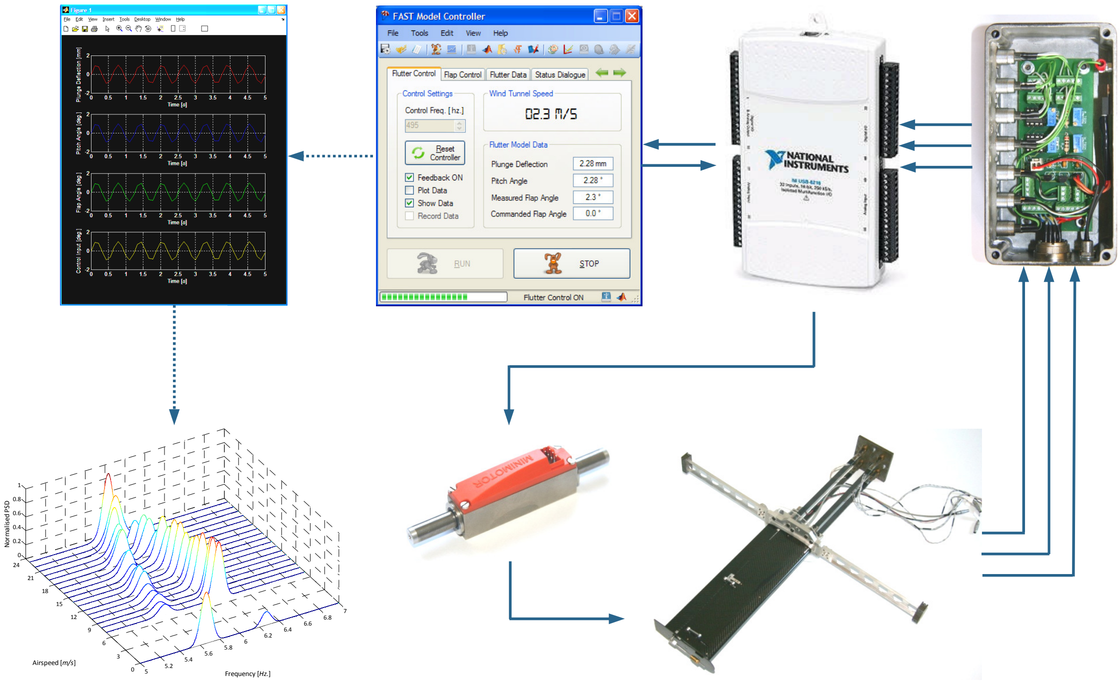
- Functions to load control system data *viz*. discrete state-space model, LQR and Kalman gain matrices and calibration data from files with fixed the extensions "~.csd" and "~.cal" respectively
- The option of linking the software to MATLAB™, which allows the program to access MATLAB™ functions and plotting routines. MATLAB™ was used mainly to display a strip chart of the data, but scope is provided to link more MATLAB™ routines (to post-process recorded wind tunnel data for example) to the FAST Model Controller software
- The ability to select the number format for the data table in the "Flutter Data" tab
- A feature to simulate flutter suppression control by calculating Kalman state estimates and feedback gains in real-time from recorded data previously generated in MATLAB™. If the DAQ is present in the system and initialised, this "Simulate Flutter Control" function will also drive the control surface
- A step input function to investigate the real-time response of the control surface actuator and actuation mechanism to estimate its transfer function. This requires specification of a step amplitude (control surface angle), a time at which to command the control surface angle and an end time to stop recording the response of the system. This function can be used with either the linear actuator or the model aircraft servo

The following dialogue boxes can be opened in the FAST Model Controller software, either from the menu or via an icon on the top tool strip:

- "Actuator Parameters" to specify the maximum and minimum actuator limits and the respective control signals to achieve these limits (analogue voltage or PWM signal). A linear interpolation of this data enables calculation of the signal required to move the control surface to any specified angle within the specified range
- "Plot Parameters" where parameters such as how many data points to plot in a given period or auto-scaling of the vertical axis for example, can be edited
- "DAQ Settings" to set the DAQ input channels, their voltage ranges and their configuration (referenced single ended (RSE), non-referenced single ended (NRSE) or differential (DIFF) input), and to select the use of a PWM or analogue voltage signal to drive the control surface actuator
- "Calibration Parameters" to edit the calibration data of the control surface angle sensor and the plunge and pitch SGBs. Calibration data loaded from the calibration file is automatically shown in this dialogue box. A "Flutter Model Trim Data" groupbox provides the option to adjust for the static deflection of the model due to steady airflow (i.e. to adjust sensor readings to match the trim position of the model)
- "Control Gains" to edit the LQR feedback and Kalman filter gains
- "State-Space Matrices" to edit the discrete state-space matrices that are a function of the configuration of the FAST model

4.5 Active Flutter Control Model Design Summary

A flutter model with the capability to actively prevent flutter or suppress an existing flutter state was successfully designed and manufactured. The model was designated as the Flutter with Active Suppression Technology (FAST) wind tunnel model. The complete model design incorporated mechanical, electronic and software components that interface with each other as shown schematically in Figure 4.23. The model mount in particular is considered a novel feature of the entire design as it is much smaller and structurally simpler than the large mount on which its design is based. The overall size of the complete FAST model is significantly smaller than any other similar models described in the literature. Custom software completes the system and allows for easy wind tunnel testing and demonstration of flutter and flutter suppression with the model.



Control Freq. [hz.] 495

Wind Tunnel Speed 02.3 M/S

☑ Feedback ON
☐ Plot Data
☑ Show Data
☐ Record Data

Plunge Deflection 2.28 mm
Pitch Angle 2.28 °
Measured Flap Angle 2.3 °
Commanded Flap Angle 0.0 °

Flutter Control ON

Figure 4.23 Complete FAST Wind Tunnel Model and Instrumentation

Chapter 5

Research Method

The equipment, instrumentation, wind tunnel facilities and open- and closed-loop wind tunnel testing procedures of the active flutter suppression wind tunnel model are outlined in this chapter. Initial open- and closed-loop flutter tests were done in the CSIR CWT. Further testing that focused on significantly extending the flutter boundary of the model was performed in the LSWT of the CSIR, which has a much higher maximum operating speed. The test procedures are documented and shown in the form of flow charts to enable future wind tunnel test campaigns to achieve similar results by following these same procedures. The chapter ends with some precautions to adhere to for safe testing of the aeroservoelastic model.

5.1 Data Required and Data Analysis

Preliminary Data Required

Before open- and closed-loop wind tunnel testing of the FAST model could be conducted, various model parameters required to design an appropriate controller had to be measured and where this was not possible, estimated. These parameters were:

- The mass of the wing and moving section of the flexure mount
- The mass inertia of the wing and moving section of the flexure mount, estimated from the CAD drawing of the complete model
- The cg position of the wing and moving section of the flexure mount, estimated from the CAD drawing of the complete model
- The mass inertia of the control surface, estimated from the CAD drawing of the complete model
- The cg position of the control surface, estimated from the CAD drawing of the complete model
- The pitch and plunge natural frequencies of the model, determined from the PSDs of the measured wind-off time series response of the model (as mounted in the wind tunnel)
- The damping coefficients associated with the plunge, pitch and control surface modes of the wind tunnel model, estimated by observing the damping of the wind-off response

Wind Tunnel Data and Data Analysis

Open-loop wind tunnel data was analysed after each wind tunnel run (at a given airspeed), to assess the subcritical response of the FAST model for on-line flutter speed prediction. Power spectral densities of each of the measured FAST model states were calculated using software developed in MATLAB™ (see Appendix C). These were used in the on-line flutter prediction, which was performed using the half power bandwidth and simplified Zimmerman flutter margin techniques outlined in Section 2.4. Although the half power bandwidth method was implemented (in MATLAB™), the simplified Zimmerman flutter margin technique (Section 2.4.4) was used predominately because of the simplicity of the FAST model and its well defined structural modes. Modal frequencies were read directly off the PSDs of the time series data using an automatic peak finding routine developed in MATLAB™. Calculated flutter margins based on the subcritical response of the model were plotted and extrapolated to zero to give a predicted flutter onset speed, up until the actual flutter speed of the model. During closed-loop wind tunnel testing, the raw signals from the FAST model transducers were multiplied in real-time by calibration coefficients to give the modal positions of the model. The plunge displacement and pitch angle were measured using calibrated strain gauge bridges located on the FAST model flexure mount (see Section 4.1.1), whilst the control surface angle was measured using a calibrated Hall-effect rotary sensor (see Section 4.1.3) on the control surface shaft. A Kalman filter was applied to the data in real-time to estimate the remaining unmeasured FAST model system states for input to the LQR feedback controller which was used to calculate the control surface angular positions required to suppress flutter.

5.2 Test Equipment

5.2.1 FAST Wind Tunnel Model and Instrumentation

The mechanical, electrical and software components of the FAST wind tunnel model are fully described in Chapter 4 so are not discussed further here. The DAQ described in Section 4.3.1 was used in its differential analogue input mode to eliminate ground loop problems and minimise electrical noise. This mode is set in the "Edit" tab under the "DAQ Settings..." option of the software described in Section 4.4.3 and shown in Appendix E.

5.2.2 Calibration Wind Tunnel

Complete testing of the FAST model was conducted in the CSIR CWT. This is a wooden blow-down wind tunnel with a (nominally) $600mm$ long $800mm \times 600mm$ hexagonal test section (see Figure 5.1). The CWT was ideal for flutter testing of the FAST model because of its small size, low turbulence and excellent velocity resolution. The CWT has an overall speed range of $0.5m/s$ to $35.0m/s$. The maximum speed can be restricted by placing various gauzes behind the test section, but for these tests only the open configuration was used. A new cantilever mount was developed to accommodate the FAST model in the CWT. This mount was made very stiff, having a much higher first natural frequency than both the plunge and pitch frequencies of the FAST model. This was achieved by using thick steel channel sections and suitable geometric design. Cut-outs were made in the mount to reduce its mass without compromising its stiffness, and it incorporated slotted holes to allow the FAST model to be adjusted in all three axes for accurate positioning in the CWT test section. The mount clamped to existing columns supporting the CWT test section (see Figure 5.1 and Figure 5.2). The FAST model is shown mounted in the CWT in Figure 5.3.

Figure 5.1 The CSIR's Calibration Wind Tunnel

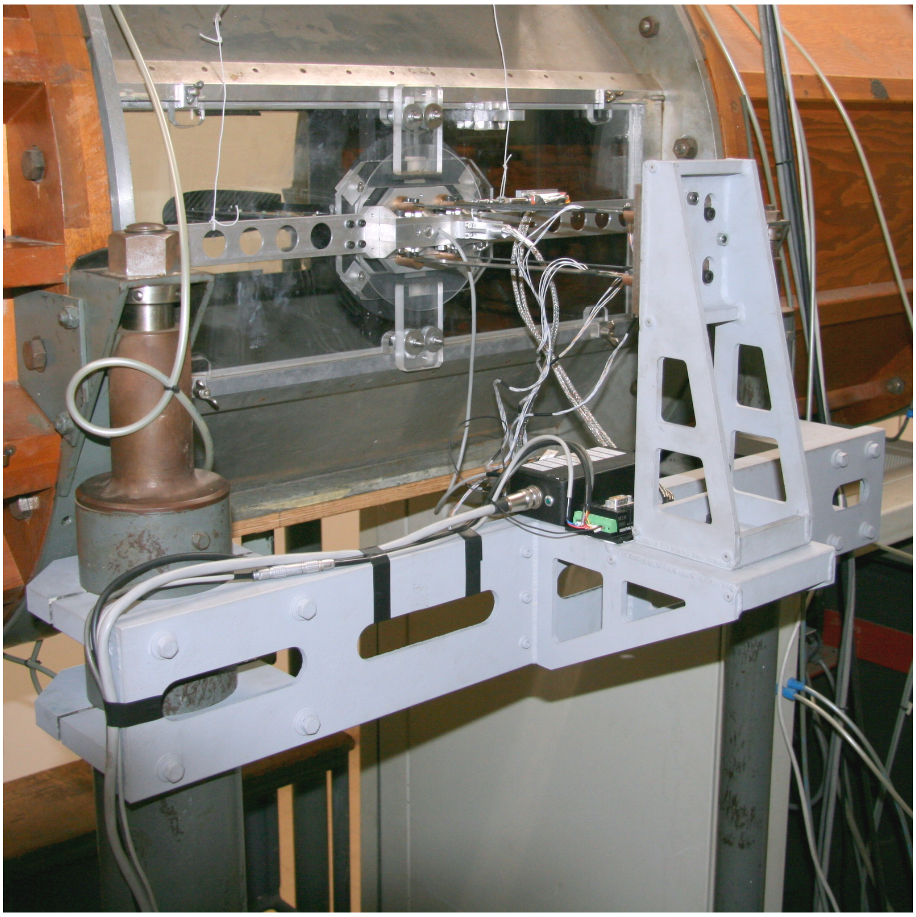


Figure 5.2 CWT FAST Model Adjustable Cantilever Mount

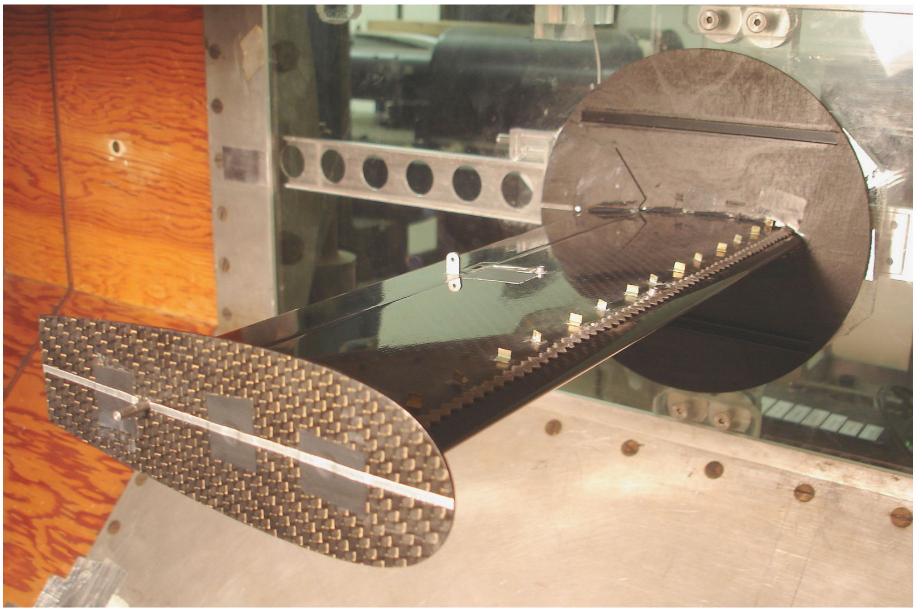


Figure 5.3 FAST Model Mounted in the CWT

5.2.3 Low Speed Wind Tunnel

Limited testing of the FAST model was performed in the CSIR LSWT, which is a continuous wind tunnel, shown schematically in Figure 5.4. It has an adjustable atmospheric slot that is used to vary the pressure within the test section, which is nominally 1.5*m* (4.9*ft*) × 2.1*m* (6.9*ft*) in cross-section and 2.1*m* (6.9*ft*) long. The minimum operating speed of the LSWT is 5.0*m*/*s* and the maximum is 135.0*m*/*s*. Another cantilever mount was made for the LSWT testing of the FAST model, similar to the mount used in the CWT. This mount is shown in Figure 5.5 and the FAST model mounted in the LSWT is shown in Figure 5.6.

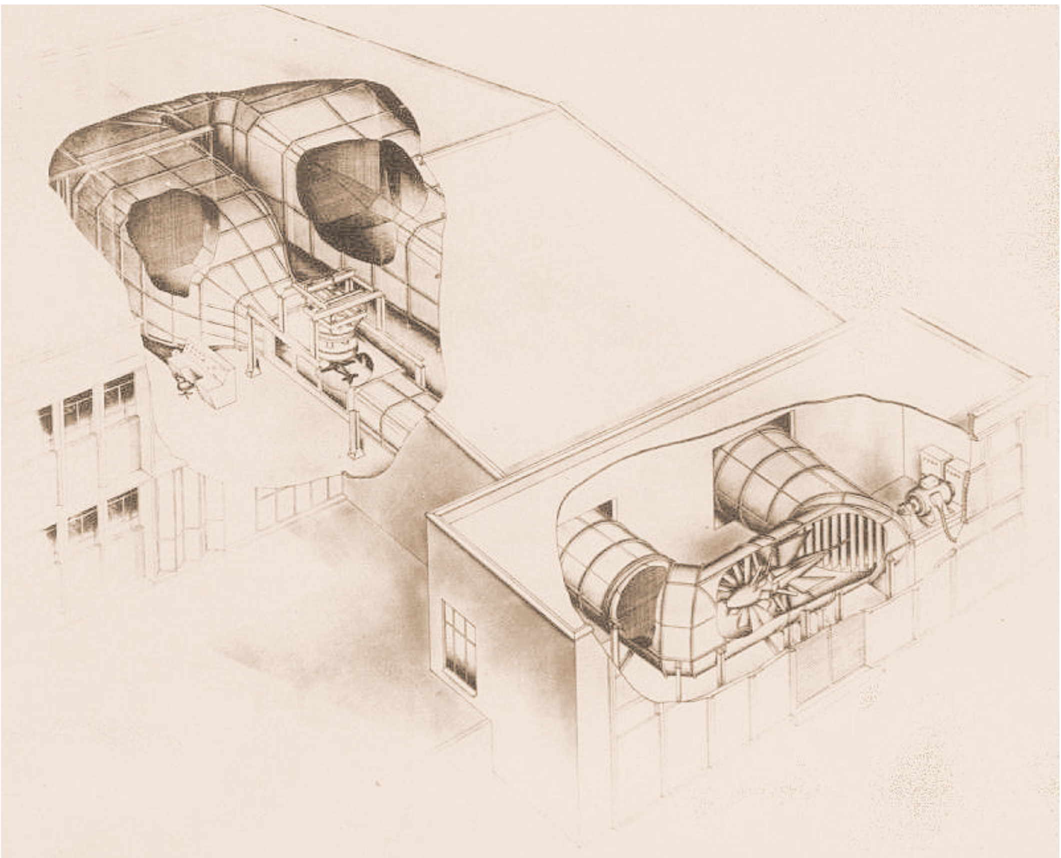


Figure 5.4 The CSIR's Low Speed Wind Tunnel

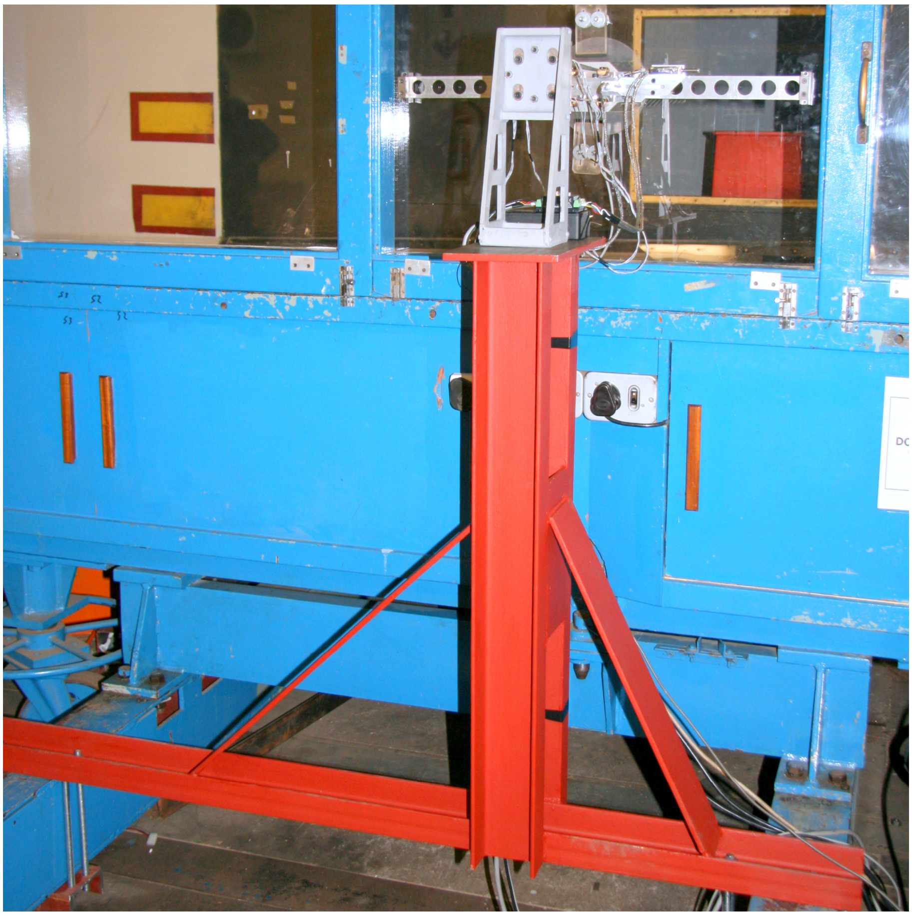


Figure 5.5 LSWT FAST Model Adjustable Cantilever Mount

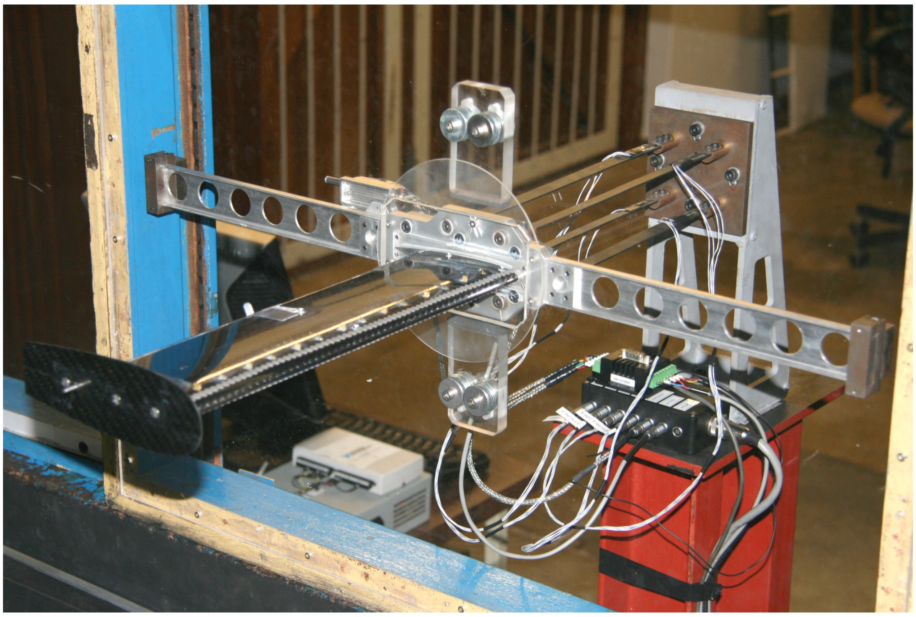


Figure 5.6 FAST Model Mounted in the LSWT

5.3 Test Matrix and Summary

5.3.1 Test Matrix

The FAST model was designed to have several variable parameters (outlined in Chapter 4) to provide scope for variations in wind tunnel testing. In this work though, testing was limited to validating the numerical model of the aeroservoelastic system and demonstrating the open- and closed-loop response of the wind tunnel model, within the planned project objectives and budget. To achieve this, the test matrix shown in Table 5.1 was completed. Three FAST model configurations, each with the control system deactivated (open-loop tests) and then activated (closed-loop tests) were evaluated. These were a baseline cg position configuration, a forward cg position configuration and an aft cg position configuration.

Table 5.1 FAST Model Wind Tunnel Test Matrix

FAST Model Configuration (cg Position)	**Wind Tunnel Test Type**		
	Open-Loop	**Closed-Loop**	**Flutter Boundary Extension**
Baseline	✓ $U = 7.0m/s \rightarrow U_f$	✓ $U = 7.0m/s \rightarrow U_f$	✓ $U = U_f \rightarrow U_{max,\,CWT}$
Forward	✓ $U = 7.0m/s \rightarrow U_f$	✓ $U = U_f$	×
Aft	✓ $U = 10.0m/s \rightarrow U_f$	✓ $U = U_f$	×

5.3.2 Open-Loop Wind Tunnel Test Summary

Initially, open-loop wind tunnel flutter tests were conducted to establish the flutter dynamics and speed of the FAST model, and to assess how well the mathematical model matched the measured response of the system. The subcritical response (modal frequencies and dampings) of the model was tracked and used to perform an online flutter prediction during wind tunnel testing. Given the well defined structural modes and simplicity of the FAST model, the simplified Zimmerman flutter margins evaluated at each subcritical wind tunnel test speed were sufficient to predict its flutter speed. For each cg position configuration tested the CWT was run at speeds from 7.0m/s up to the actual critical flutter speed of FAST model, in increments of 1.0m/s, which characterised its dynamics. The mathematical model of the system was then tailored to match the physical response, to ensure an effective flutter suppression control system design. This was done by slightly adjusting the estimated modal dampings, wing mass inertia, wing cg position and the control surface mass inertia until the measured FAST model modeshapes matched those predicted by the mathematical model.

5.3.3 Closed-Loop Wind Tunnel Test Summary

During initial closed-loop wind tunnel testing the LQG controller was activated prior to reaching the critical open-loop flutter speeds U_f of the FAST model, to demonstrate that flutter could not be induced with the control system operating. Active suppression of a developed flutter cycle was demonstrated, for each cg position configuration, by switching the control system off at these critical open-loop flutter speeds and allowing flutter to occur, then switching it on again. Control gains were tuned during wind tunnel testing to achieve a suitable and robust controller. Further closed-loop wind tunnel testing was conducted with the FAST model in its baseline cg position configuration over the speed range 7.0*m/s* to U_f to evaluate its subcritical closed-loop response. Finally, FBE tests were performed with the FAST model in its baseline cg position configuration in both the CWT and LSWT to establish what increase in the flutter boundary could be achieved with the active flutter control system operative. The first series of FBE tests were done in the CWT at speeds from U_f to 35.6*m/s* in increments of nominally 1.0*m/s*. Additional (limited) FBE tests were then performed in the LSWT. These tests were stopped at an arbitrarily selected speed of 53.9*m/s* at which point the LQG controller was still effective in preventing flutter, but the risk of failure of the model due to high static loads and consequent damage to the wind tunnel was considered too great.

5.4 Setup and Testing Procedures

5.4.1 FAST Model Instrumentation Calibration Procedures

The FAST model instrumentation had to be calibrated before wind tunnel testing could commence. Transducers that required calibrating were the flexure mount plunge and pitch strain gauge bridges, the control surface angle sensor and the control surface actuator. The processes followed to calibrate each of these transducers are described below.

Plunge Displacement and Pitch Angle Strain Gauge Bridges

The two strain gauge bridges at the root of the FAST model flexure mount were calibrated using a specially designed and manufactured jig shown in Figure 5.8 and Figure 5.8. The purpose of the jig was to fix the pitch degree of freedom of the mount whilst allowing motion in the plunge degree-of-freedom, and *vice versa*. The model mount was calibrated with the static deflection of the wing-mount combination as the zero point (i.e. h = 0*mm*; α = 0°) in both the plunge and pitch degrees-of-freedom. These values were measured to be h = +7.5*mm* (positive downwards) and α = +0.0° (positive leading edge up) from the level horizontal position of the model mount. With the model in this position, the DC voltage offset potentiometers on the strain gauge bridge amplifier circuit (see Figure 4.17) were adjusted until each SGB gave an output of 0.0*V*. With the pitch degree-of-freedom held fixed, the plunge displacement SGB was calibrated over the range h = +1.5*mm* to h = +13.5*mm* (i.e. h = 6.0*mm* either side of the mean) in increments of 0.5*mm.* Thereafter, the pitch angle SGB was calibrated over the range α = -6.0° to α = +6.0° in increments of 0.5° with the FAST model prevented from moving in its plunge degree-of-freedom. Finally, the model mount was moved to various combined pitch and plunge positions (within the same limits) to generate cross-terms for the calibration matrix. In summary, the two SGBs were calibrated by (i) loading the model with the pitch angle held fixed and measuring the response from each SGB to the pure plunge displacement (ii) fixing the plunge displacement whilst measuring the response from each SGB to a change in pitch angle and (iii) measuring cross terms with the model allowed to pitch and plunge simultaneously. A 0.0*mm* - 500.0*mm* digital height gauge was used to measure the plunge displacement and a Kell-Strom™ PRO 3600 digital inclinometer to measure the pitch angle of the mount. It should be noted that if the modal frequencies of the FAST model mount are adjusted by changing the mount flexure orientations, the root strain gauge bridges will need to be re-calibrated.

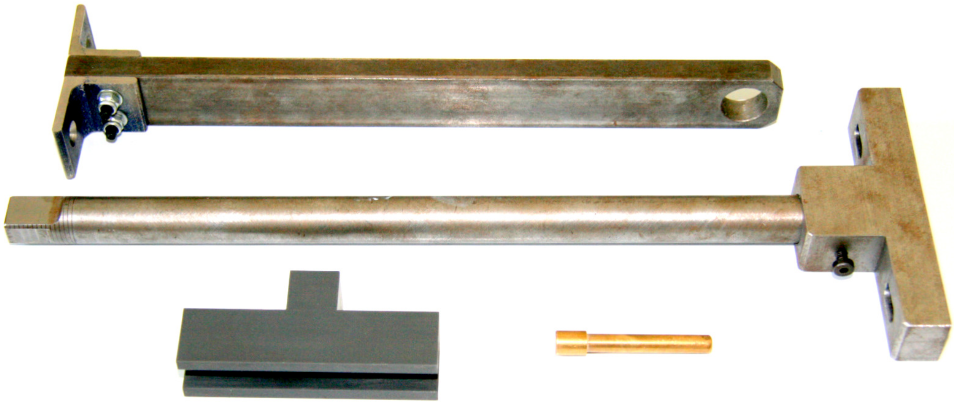


Figure 5.7 Disassembled Flexure Mount Calibration Jig

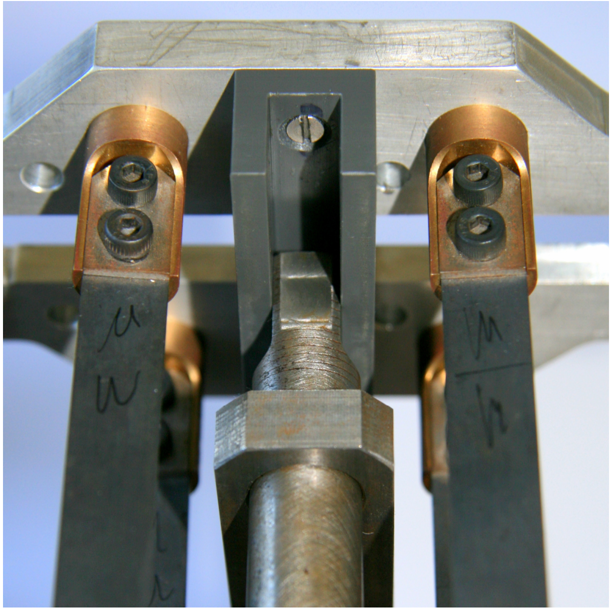


Figure 5.8 Calibration Jig Permitting Only Plunge Motion

Control Surface Angle Sensor

Calibration of the Contelec™ Vert-X 13 Hall-effect control surface angle rotary sensor was done by manually moving the control surface to known angular positions between +35° and -35° and plotting this measured angle against sensor output voltage. The set control surface angle was measured with a Kell-Strom™ PRO 3600 digital inclinometer. This calibration was done prior to calibration of the control surface actuator.

Control Surface Linear Actuator

The Faulhaber™ 1247-020-01 control surface actuator was calibrated to relate its voltage input to the angle at which it positioned the control surface. This was done by driving the actuator with voltage inputs from -9.0*V* to +9.0*V* in increments of 0.5*V* and measuring the corresponding control surface angle with the already calibrated Contelec™ Vert-X 13 Hall-effect rotary sensor.

5.4.2 Identification of the FAST Model Control Surface Dynamics

As discussed in Section 3.1.5, it was desired that the control surface and its actuation mechanism have similar dynamics to a second order oscillator function (Equation 3.112) to allow for the use of a simplified aeroservoelastic model in the design of the flutter suppression control law. To determine the transfer function of the control surface and actuation mechanism experimentally, the actuator was driven with a swept-sine voltage signal from 1.0*Hz*. to 12.0*Hz*. using a *TTi* function generator (Model Number: TG2000) whilst recording the control surface angle. The recorded control surface response was evaluated with a GenRad Computer-Aided Test System (Model Number: 2515) and plotted on a Bode diagram.

5.4.3 FAST Model Setup Procedures

FAST Model Calibration Wind Tunnel Installation Procedure

The FAST wind tunnel model was mounted on an existing flexible mount that permits both pitch and plunge displacements (Sutherland, 2006; 2008). The entire system was bolted to the steel backup structure, which clamped to the CWT test section supports (see Figure 5.2). Slotted holes in the mount system allowed the model to be moved in all three axes and accurately positioned in the CWT test section. A wingtip end-plate and a wing root splitter-plate either side of the wind tunnel side wall were attached to the FAST model, once it was mounted in the CWT. Mechanical stops (see Figure 5.9(a)) were then attached to the wind tunnel side wall, to prevent large model oscillations and destruction of the model in the event of a control system failure or occurrence of violent flutter.

FAST Model Instrumentation Setup Procedure

Since the FAST model had its own data acquisition and control system (as opposed to the wind tunnel equipment), some points considered with respect to setup of this instrumentation were:

- Electromagnetic interference between the linear motor driving the control surface and the control surface rotary sensor added significant noise to the control surface angle measurements (because both employ Hall-effect magnetic sensors, in close proximity to each other). This problem was eliminated by shielding both the rotary sensor and actuator cables with braided metal sheathing connected to the FAST model instrumentation common ground
- The three phase cables (Phase A, Phase B and Phase C) of the Faulhaber™ linear actuator were shielded separately from the other controller wires to ensure that the actuator operated properly (as directed in the manufacturer's manual)

5.4.4 Calibration and Low Speed Wind Tunnel Setup Procedures

In preparation for testing of the FAST model and before operating the CWT:

- The test section was checked thoroughly to ensure that no loose fittings, model fasteners, tools etc. were left behind, which could be blown down the tunnel and damage the fan blades
- All gauzes behind the test section were removed to allow the CWT to be used in its open configuration
- The ambient pressure and temperature were measured in the CWT room and recorded in the CWT software, to ensure that the correct wind tunnel speed readouts were given by the CWT software
- All removable panels in the CWT test section were checked and tightly secured in position

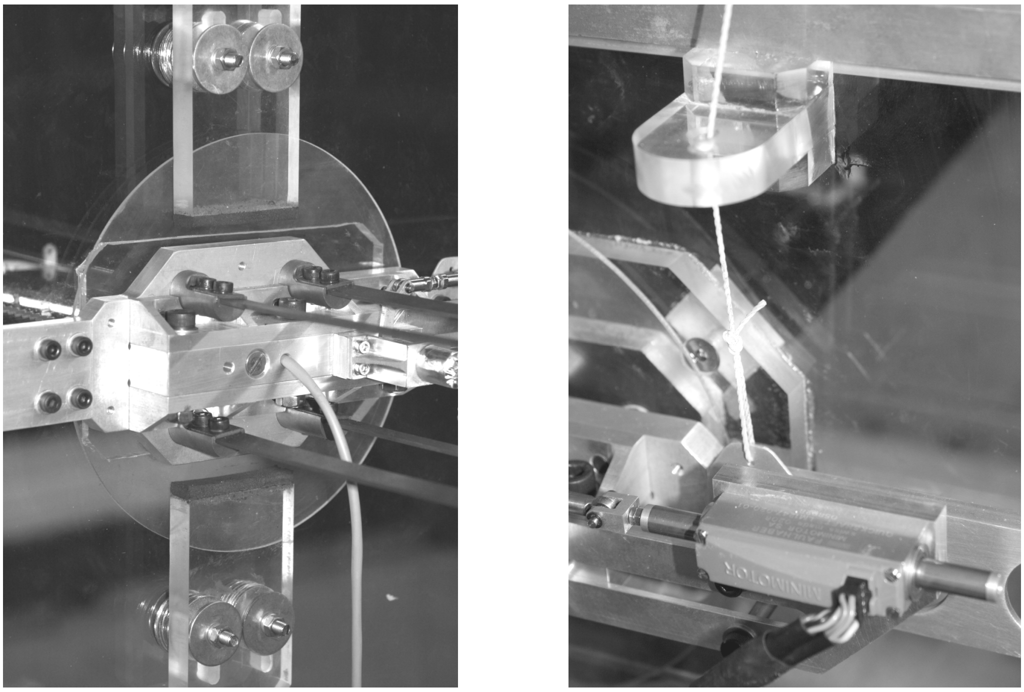
Similarly, before operating the LSWT for flutter boundary extension tests using the FAST model:

- The pressure within the test section was matched to the ambient air pressure using the adjustable slot behind the wind tunnel test section
- The test section was checked thoroughly to ensure that no loose fittings, model fasteners, tools etc. were left behind, which could be blown down the tunnel and damage the fan blades
- All removable panels in the LSWT test section were checked and tightly secured in position

5.4.5 Wind Tunnel Test Procedures

FAST Model Excitation Procedure

During wind tunnel flutter testing excitation of the structural modes of the model was required to evaluate their dampings and frequencies as a function of airspeed. When tunnel turbulence was insufficient to provide this excitation the model was given an initial displacement and then allowed to oscillate freely in the airflow, whilst its response was recorded. The model was displaced by pulling a string attached to a plate on the model mount as shown in Figure 5.9 (b). Close to and above the flutter speed of the FAST model, turbulence in the wind tunnel was sufficient to excite the two natural modes of the model and initiate flutter in both the CWT and LSWT.



(a) Adjustable Mechanical Stops **(b)** Model Excitation System

Figure 5.9 FAST Model Adjustable Mechanical Stops and Excitation System

Open-Loop Wind Tunnel Test Procedure

The procedure followed for open-loop wind tunnel flutter testing of the FAST model followed the sequence described below and indicated in the flow diagram in Figure 5.10:

a) The wind tunnel was run at an initial subcritical flutter speed of 7.0*m/s*

b) The model was excited by giving it an initial displacement. Its damped response was measured and recorded while it oscillated freely in the airflow

c) Modal dampings and frequencies were calculated from the PSDs of the recorded time series data

d) The simplified Zimmerman flutter margin (or other appropriate flutter margin) based on the subcritical response of the model was calculated

e) The calculated flutter margin for this speed was then added to a plot of flutter margin versus wind tunnel speed. Once subcritical tests at a sufficient number of test speeds were completed, an on-line prediction of the flutter speed was made. This was done by fitting a polynomial curve through the points in the plot and extrapolating this to zero, to update the on-line flutter speed prediction of the model

f) If the model did not flutter, the wind tunnel speed was increased by nominally 1.0m/s (or less when close to the expected flutter speed) and steps (b) to (e) were repeated. In this way the actual flutter speed of the model was approached safely

g) When the model fluttered the speed was recorded and the wind tunnel was stopped

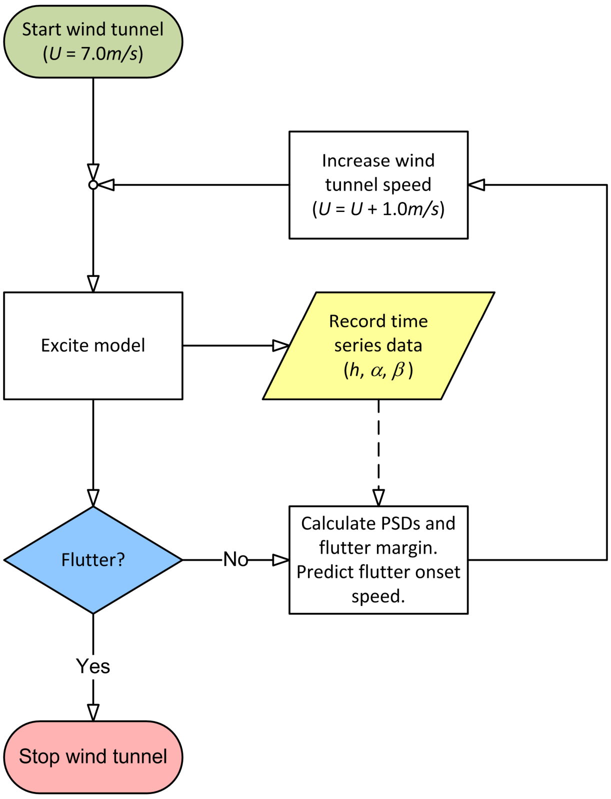


Figure 5.10 FAST Model Open-Loop Wind Tunnel Test Procedure Flow Chart

Closed-Loop Wind Tunnel Test Procedure

Initial closed-loop tests were required to evaluate the performance of the LQG controller and its effectiveness in preventing flutter at the critical open-loop flutter speed. The closed-loop wind tunnel testing was performed by following the procedure described below and shown schematically in Figure 5.11:

a) With the model constrained so it was unable to move freely, the flutter suppression control system was activated

b) The wind tunnel speed was set to the observed open-loop critical flutter speed of the model for the cg position configuration under investigation

c) The model was then allowed to move freely and its closed-loop response to wind tunnel turbulence was observed and recorded

d) If turbulence did not excite the model sufficiently, it was then given an initial displacement and allowed to oscillate freely from this position while its closed-loop response was observed and recorded

e) If flutter was still not induced by this manual excitation of the FAST model at its open-loop flutter speed with the control system active, then the flutter boundary extension tests were commenced

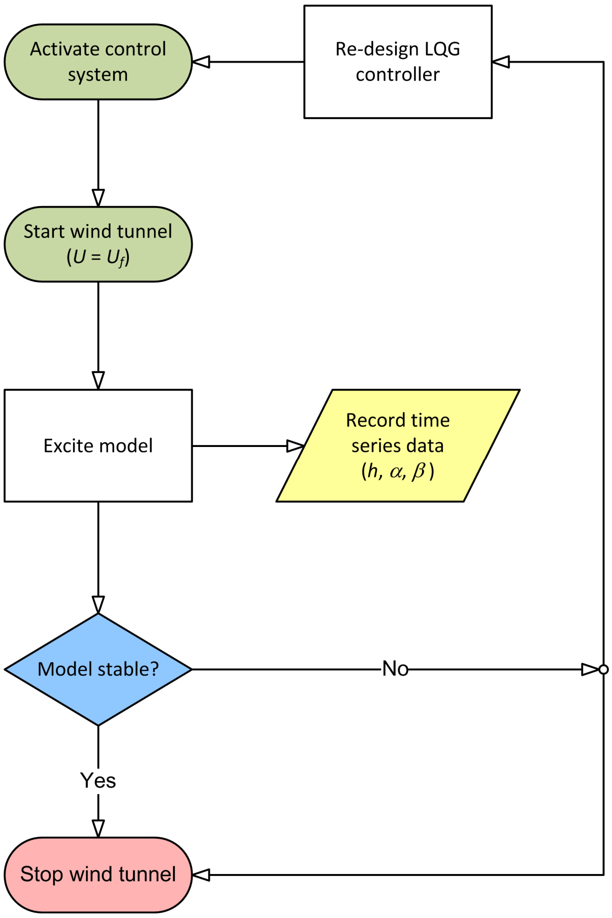


Figure 5.11 FAST Model Closed-Loop Test Procedure Flow Chart

Although it was not necessary in this work, if after following the steps outlined above, the LQG controller design is ineffective in suppressing flutter, the control system would need to be re-designed and steps (a) to (e) repeated until an effective control system design is achieved.

Flutter Boundary Extension Wind Tunnel Test Procedure

Once closed-loop wind tunnel flutter testing at the open-loop flutter speed of each cg position configuration was demonstrated successfully, flutter boundary extension tests commenced with the model in the baseline cg configuration. The purpose of these tests was to investigate the robustness of the controller at off-design points, and to determine to what extent the flutter boundary of the model could be increased by using active feedback control. The test procedures followed are described below and shown schematically in Figure 5.12(a) for CWT tests and in Figure 5.12(b) for LSWT tests. The procedure followed for flutter boundary extension tests (baseline cg position configuration) in the CWT was:

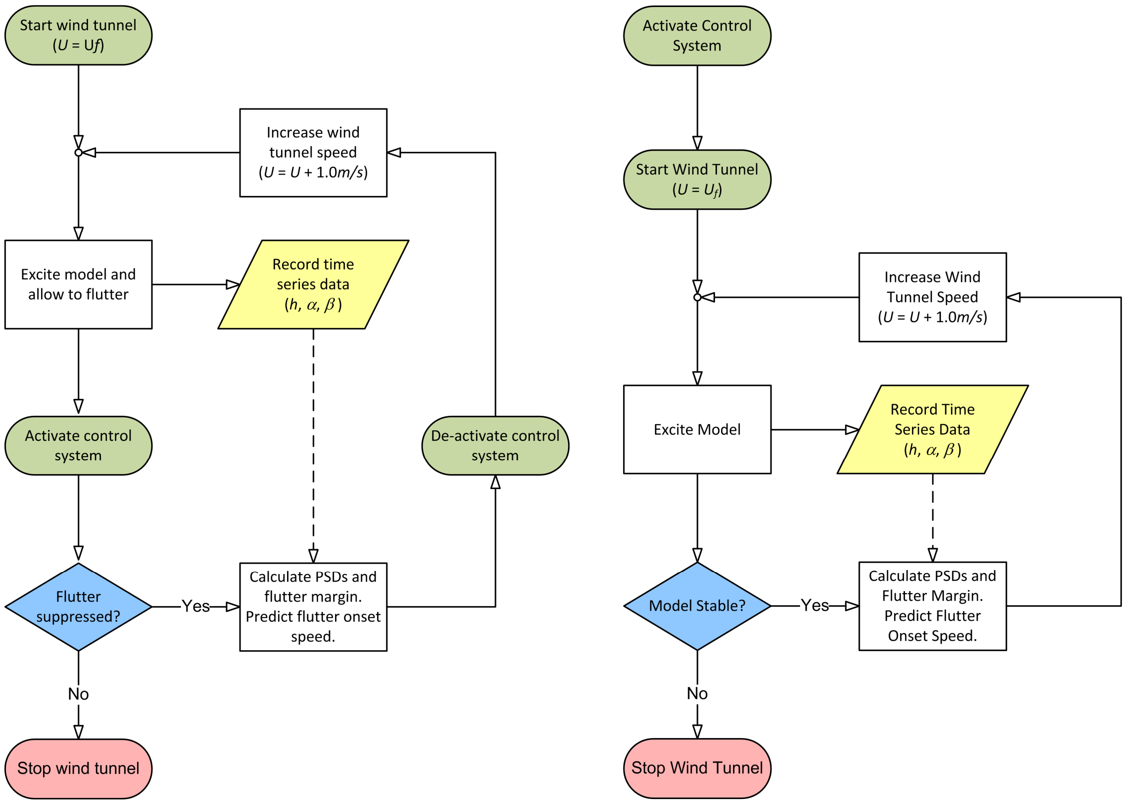
a) The wind tunnel speed was set to the open-loop critical flutter speed

b) With the active control system off, the model was allowed to oscillate freely after being excited by an initial displacement

c) Once the model entered a sustained flutter cycle, the controller was activated to evaluate the effectiveness of the feedback loop and control surface in suppressing the flutter

d) If flutter could not be induced by manually exciting the model with the control system active, the control system was deactivated and wind tunnel speed was increased by nominally 1.0m/s and steps (b) and (c) were repeated

e) Steps (b) to (d) were repeated until the flutter suppression controller became ineffective or the maximum operating speed of the wind tunnel was reached



(a) CWT Test Procedure **(b)** LSWT Test Procedure

Figure 5.12 FAST Model Flutter Boundary Extension Test Procedure Flow Chart

Due to the much higher operating speeds of the LSWT and since flutter boundary extension tests are inherently dangerous, the method for these tests in the LSWT was slightly different, and proceeded as follows:

a) With the FAST wind tunnel model constrained to prevent it from moving freely, the feedback control system was activated
b) The wind tunnel speed was set to the open-loop critical flutter speed for the baseline cg position configuration
c) The flutter model was manually excited by giving it an initial displacement
d) If flutter could not be induced by manually exciting the model with the control system active, the wind tunnel speed was increased by nominally 1.0*m*/*s* and step (c) was repeated

Flutter boundary extension tests must be stopped when the controller becomes ineffective or the static forces acting on the model are considered too high.

5.5 Precautions

Given the inherent danger of flutter testing, certain precautions were taken before (Section 5.5.1) and during (Section 5.5.2) the wind tunnel testing (which should again be taken in any future testing) to prevent destruction of the FAST model and damage to the wind tunnels.

5.5.1 FAST Model Setup Precautions

Before wind tunnel testing commenced:

- Mechanical hard stops were suitably positioned to prevent large oscillations of the model at critical and supercritical wind tunnel speeds in case the control system failed
- A length of string was fastened to the model (off a modal point) that could be pulled to excite the model or restrain it during flutter
- The wind tunnel operator was well briefed to rapidly reduce the wind tunnel speed to stop uncontrolled flutter if necessary

5.5.2 Wind Tunnel Testing Precautions

During wind tunnel flutter testing of the FAST model, the following precautions were taken:

- During closed-loop testing the wind tunnel operator was on standby to shut down the wind tunnel immediately and restrain the model using the attached string if it entered a divergent oscillation that could not be stopped by the flutter suppression control system
- The expected flutter speed was approached slowly from below, using a suitable online flutter speed prediction technique based on subcritical model response

5.6 Conclusion

The research equipment, a description of the two wind tunnels used for the flutter testing and an overview of the flutter test procedures followed were given in this chapter, to ensure that the results obtained can be repeated in any future wind tunnel tests. Calibrations required prior to wind tunnel testing were described, and a wind tunnel test matrix is given to show that sufficient data were gathered to demonstrate the hypothesis and meet the objectives of this research. Methods for open- and closed-loop testing, as well as flutter boundary extension testing of the FAST model are listed in a procedural manner and shown schematically in flow diagrams. Finally, simple but important precautions adhered to for safe testing of the model were outlined, which can serve as a guide in future testing to minimise the likelihood of destroying the model and damaging the wind tunnel.

Chapter 6

Results and Discussion

Observations and results of the calibration, structural characterisation of the model/mount system, simulation and wind tunnel tests of the FAST wind tunnel model are presented and discussed in this chapter. Calibration of the plunge displacement and pitch angle strain gauge bridges is presented in Section 6.1.1, of the control surface angle sensor in Section 6.1.2 and of the control surface linear actuator in Section 6.1.4. The measured transfer function and fitted 2nd order Bessel filter of the control surface actuator/actuation mechanism is shown and discussed in Section 6.1.5. These data were needed for the state-space model used in the design of the LQG controller. Section 6.2 presents selected representative plots from both open- and closed-loop simulations, for a given model configuration, done using MATLAB™ and Simulink™ prior to the wind tunnel tests.

The majority of the wind tunnel tests were performed in the CSIR CWT, which has a nominal maximum operating speed of 35*m*/*s*. Three configurations, a baseline cg position, a forward cg position and an aft cg position were tested in this wind tunnel. A set of open- and closed-loop tests were conducted for each cg position configuration, after which further more comprehensive closed-loop testing was done with the FAST model in its baseline cg position configuration. Selected data and representative plots from the open- and closed-loop wind tunnel tests in both the CWT and LSWT are presented in Section 6.3. During initial wind tunnel testing in the CWT, unexpected LCO type behaviour of the FAST model was encountered. This is briefly discussed in Section 6.3.1, where it is shown how these limit cycle oscillations were stopped so that open- and closed-loop binary flutter testing could continue. Section 6.3.2 gives observations and results from the CWT open-loop tests. Pitch and plunge responses of the FAST model and corresponding PSDs for each configuration tested, at their respective critical flutter speeds, are presented and discussed. Plots of calculated and measured open-loop modal frequencies and the calculated simplified Zimmerman flutter margins as a function of airspeed are also shown for each cg position configuration tested. In Section 6.3.3 the closed-loop wind tunnel data from CWT wind tunnel tests of the FAST model are presented. These data are the measured responses (plunge displacement, pitch angle and control surface angle), Kalman state estimates and the commanded control input for each cg position configuration at its critical flutter speed. Additional plots of the controlled response and Kalman state estimates of the baseline cg position configuration at airspeeds above the open-loop critical flutter speed from both the CWT and LSWT tests are also included. Limited additional closed-loop testing of the FAST model in its baseline cg position was conducted in the CSIR LSWT in which higher operating airspeeds, up to a maximum of 135*m*/*s*, can be achieved. The focus of these LSWT tests was on assessing the robustness of the flutter suppression controller in arresting flutter of the FAST model over a much wider speed range than could be achieved in the CWT. Selected results from these LSWT tests are shown and discussed in Section 6.3.4.

6.1 FAST Model Transducer and Actuator Calibration Data

The transducers used to measure the plunge displacement, pitch angle and control surface angle of the FAST model were calibrated before wind tunnel testing commenced. As discussed in Section 4.1, two full strain gauge bridges were used to measure the plunge displacement and pitch angle of the FAST model wing, whilst a Hall-effect rotary sensor was used to measure the control surface angle. The Faulhaber™ LM1247-020-01 linear actuator used to drive the control surface was operated in its analogue input voltage mode (see Section 4.3.1), so it too was calibrated to relate the actuator control signal from the DAQ to the driven control surface angle. The results of these calibrations and plots of their residuals (and their 95% confidence error bounds) are given and discussed in this section.

6.1.1 Plunge Displacement and Pitch Angle Strain Gauge Bridges

A linear least squares method was applied to the measured calibration data (obtained as outlined in Section 5.4.1) to determine the calibration equations for both the plunge and pitch SGBs. The plunge SGB data and fitted linear surface are shown in Figure 6.1, and corresponding calculated residuals and 95% error bounds in Figure 6.2. The pitch SGB calibration data and fitted calibration surface are shown in Figure 6.3 and its calculated residuals and 95% error bounds in Figure 6.4. The data used to generate these surfaces and a more detailed explanation of the calibration and surface fitting process is given in Appendix F. Calibration results show that the outputs from both SGBs are linear over a relatively large range of movement of the mount. The points and error bounds in red in Figure 6.2 and Figure 6.4 indicate points whose error bars fall outside the mean. This was the case for 5 of the 70 plunge SGB calibration points and 3 of the 70 pitch SGB calibration points. Whilst these few outliers were undesirable, they did not detract from the otherwise excellent calibration data and linearity of the FAST model mount. Calibration coefficients were checked by moving the mount to arbitrary positions and comparing the plunge displacement and pitch angle calculated from the SGB outputs with those measured using a digital height gauge and inclinometer. In all cases the agreement was excellent.

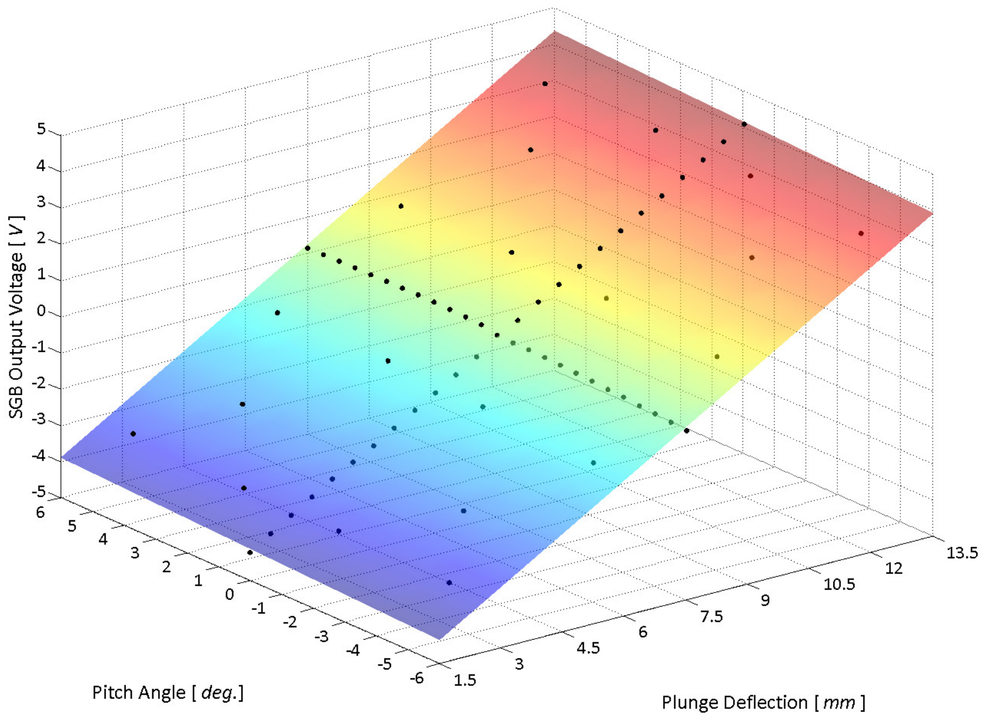


Figure 6.1 Plunge SGB Calibration Data and Surface

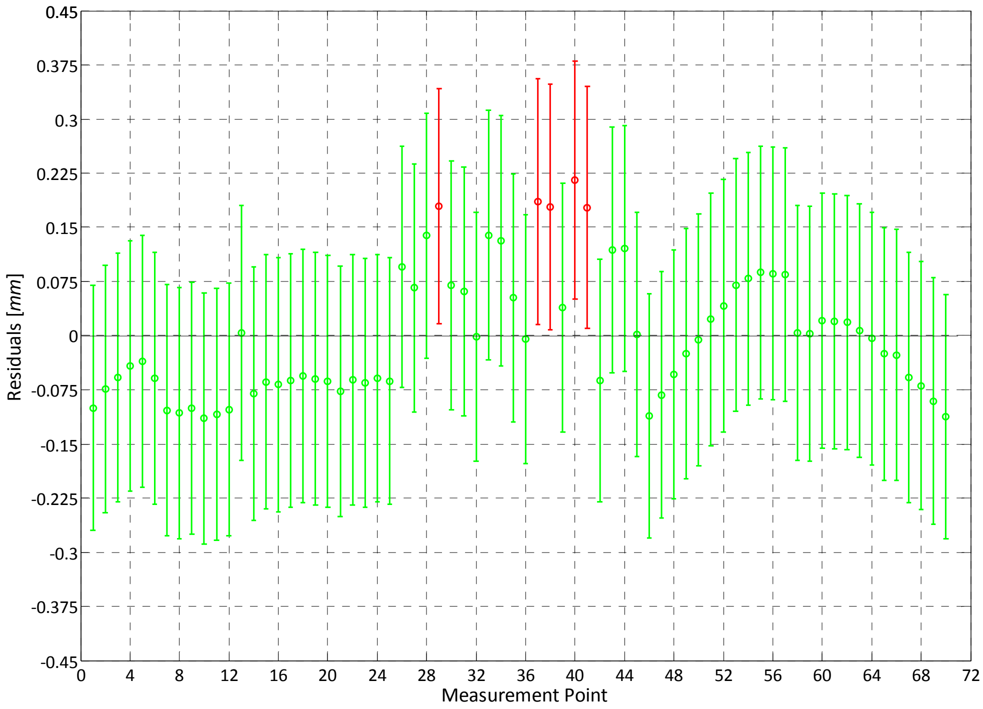


Figure 6.2 Plunge SGB Calibration Surface Residuals and 95% Error Bounds

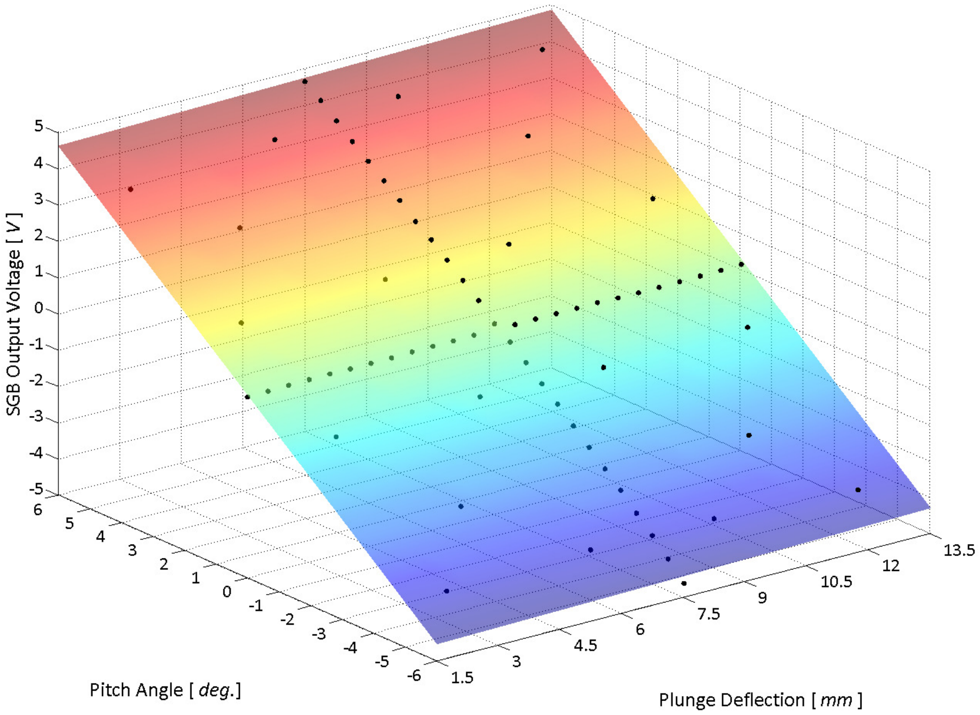


Figure 6.3 Pitch SGB Calibration Data and Surface

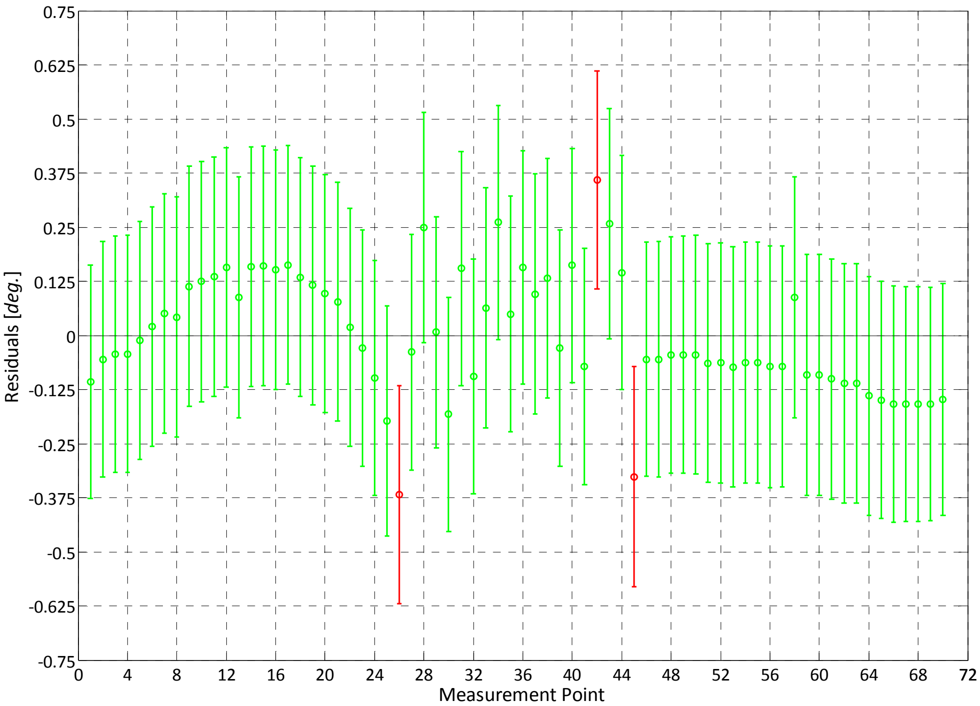


Figure 6.4 Pitch SGB Calibration Surface Residuals and 95% Error Bounds

6.1.2 Control Surface Angle Sensor

The measured points from the calibration of the control surface angle sensor and linear fit through these data are shown in Figure 6.5. Figure 6.6 shows the residual and its 95% error bound at each of the calibration points. The small scatter in the data and low (within ±0.6°) residuals indicate a good fit. Although the control surface angle sensor was calibrated over a wide range (-35.0° to +28.0°), the control system software used to drive the control surface limited its movement to within a -20.0° to +20.0° band (See Section 4.4.3).

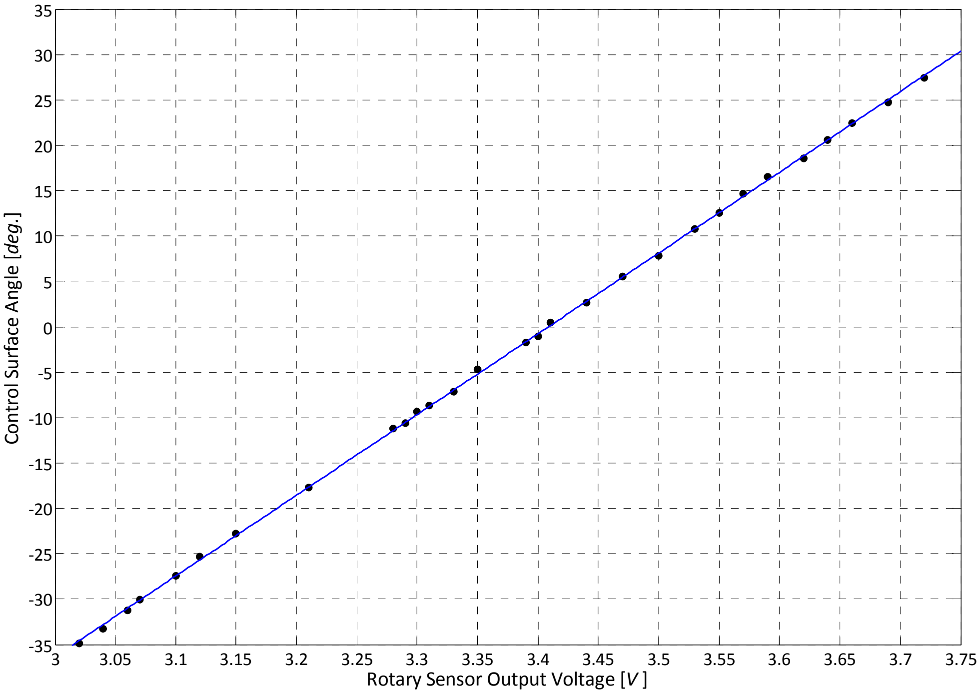


Figure 6.5 Control Surface Angle Sensor Calibration Data and Curve

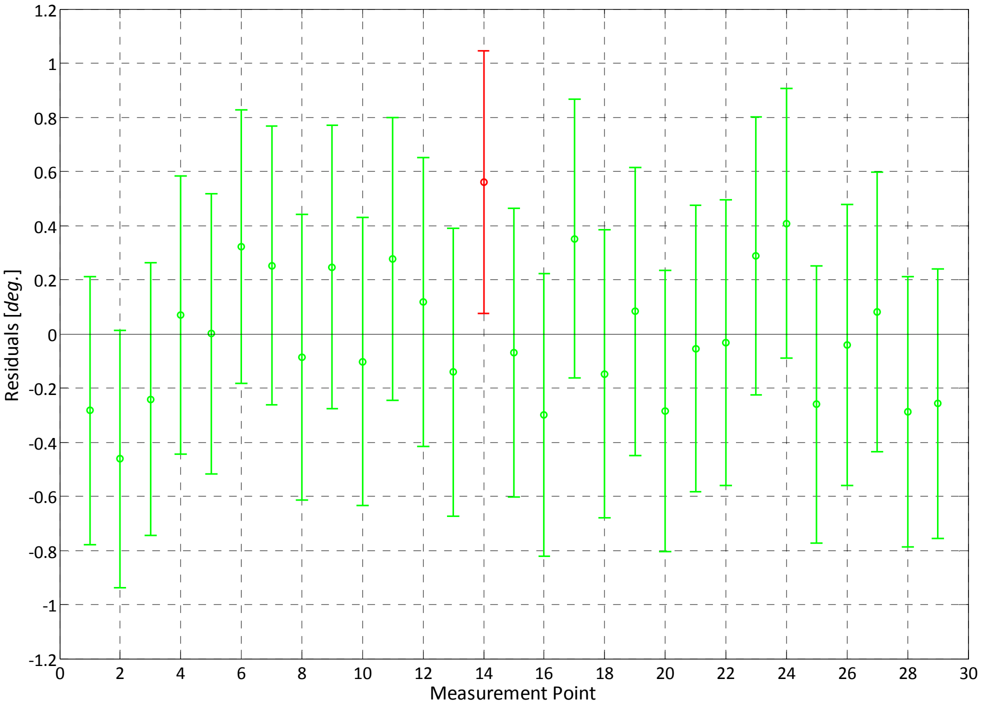


Figure 6.6 Control Surface Angle Calibration Curve Residuals and 95% Error Bounds

6.1.3 Transducer Noise Bands

The measurement noise covariance matrix $\mathbf{R}_{Kalman}$ (Equation 3.145) required for the design of the Kalman filter needed a representative estimate of the standard deviation of the noise on the measured signals. This was determined by holding the FAST model and control surface static and measuring the response (h, α, β) from each sensor over a 10.0*s* period. MATLAB™ was used to plot these responses and determine the mean, standard deviation and range of each, as given in Table 6.1.

Table 6.1 Calibrated FAST Model Transducer Noise Statistical Data

Parameter	**Plunge Displacement**	**Pitch Angle**	**Control Surface Angle**
	mm	***deg.***	***deg.***
Mean Value	0.0000	0.0100	0.0000
Standard Deviation	0.0095	0.0229	0.0764
Range	0.1114	0.2849	0.6125

Figure 6.7 shows a subset of typical data from which these estimates were obtained, where the dashed red lines indicate the mean and standard deviation of each signal. The standard deviation of the signal on the plunge deflection channel was determined as ±0.0095*mm.* The noise on the

pitch angle channel had a standard deviation of ±0.0229°, and the standard deviation of the signal recorded on the control surface angle channel was ±0.0764°. It is apparent that the largest variation in the noise data was present in the control surface angle sensor signal. This was expected as the control surface actuator would have induced noise on the measured control surface angle whilst attempting to hold it in a fixed position. For closed-loop wind tunnel testing of the FAST model, the measured transducer variances listed in Table 6.1 were increased significantly in calculations (see Appendix C) to ensure a conservative Kalman filter and control system design.

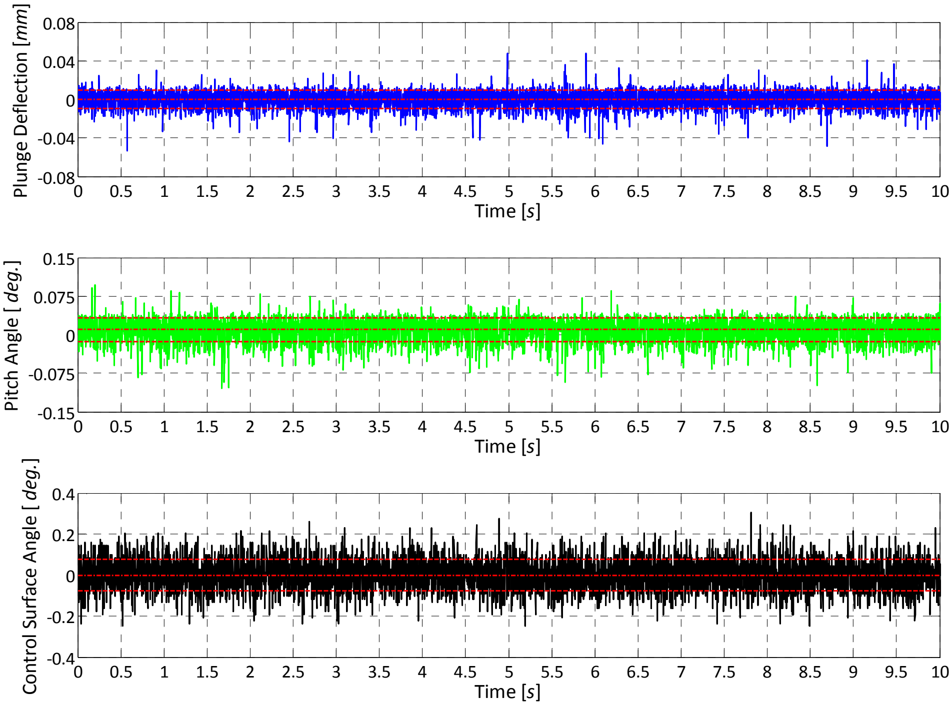


Figure 6.7 FAST Model Transducer Noise Bands and Standard Deviations

6.1.4 Control Surface Linear Actuator

The measured data and linear least squares fit from the calibration of the control surface actuator are presented in Figure 6.8. It can be seen in this figure that a voltage input of 0.0*V* did not correspond to a control surface angle of 0.0° because of the geometry of the crank used to move the control surface. The residuals and 95% error bounds of the linear least squares fit are shown in Figure 6.9. Referring to Figure 6.8, it could be argued that there is some small oscillation of the measured data points about the fitted linear curve, hence deviation from the straight line, particularly at the high positive control surface angles. An apparent trend in the residuals in Figure 6.9 confirms that a higher order fit to the data may have been more appropriate and resulted in reduced residuals without any evident trend, but it was considered unnecessary in this application. Within the anticipated operating range (±10.0°) of the control surface, the linear fit matches the data well enough. Using the lower order fit enabled simplification and faster operation of the feedback control algorithm used to position the control surface. The controller design proceeded on this basis and used the linear coefficients from the fit in Figure 6.8 in the control system software.

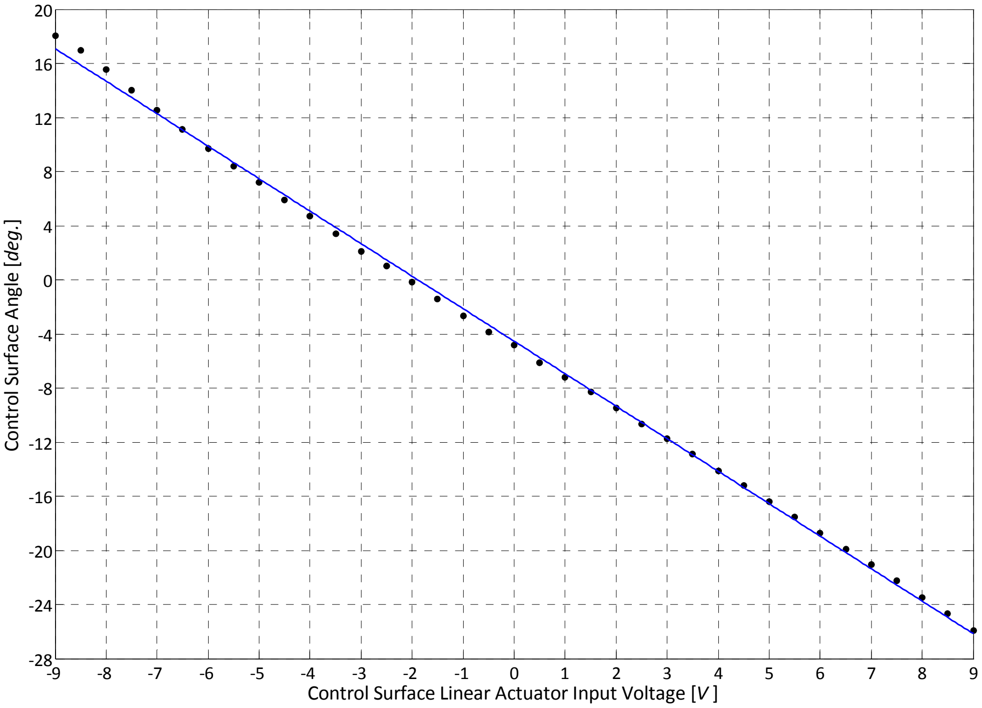


Figure 6.8 Control Surface Linear Actuator Calibration Data and Curve

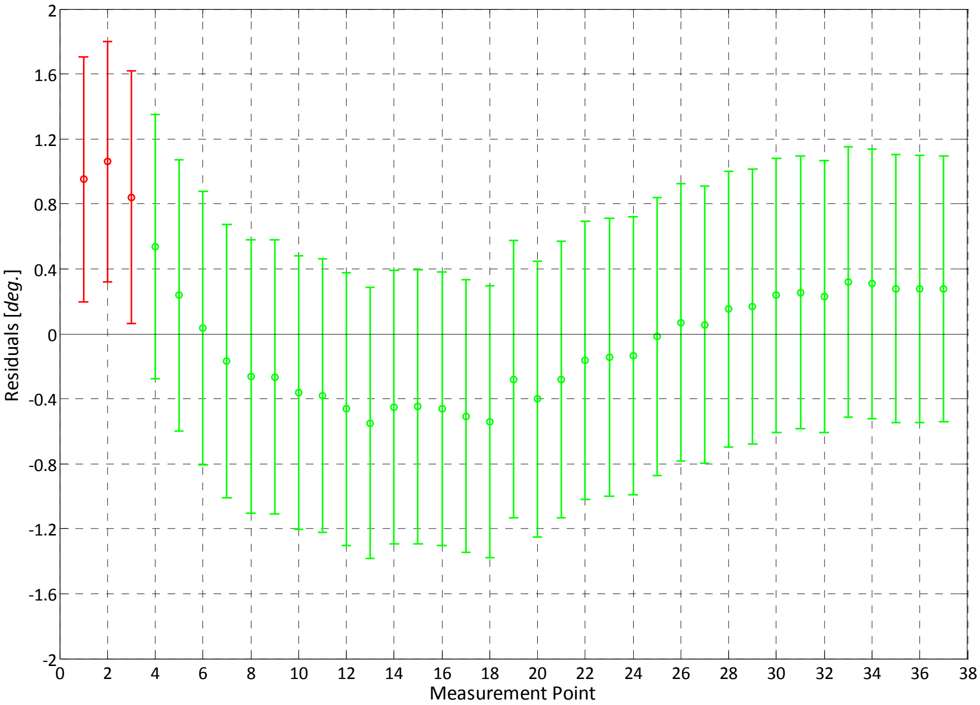


Figure 6.9 Control Surface Linear Actuator Calibration Curve Residuals and 95% Error Bounds

6.1.5 Control Surface and Actuation Mechanism Transfer Function

The control surface and actuation mechanism transfer function was identified using the method outlined in Section 5.4.2. Controller parameters of the Faulhaber™ 1247-020-01 linear DC-Servomotor were tuned until a transfer function that best suited this application (Equation 3.112) was achieved. It was found that magnitude and phase data obtained using the controller parameters listed in Table 6.2 most closely resembled a 2nd order transfer function. Accordingly, a 2nd order Bessel filter was fitted to this measured data to determine an approximate analytical transfer function for the control surface and actuation mechanism. This was done by specifying a Bessel filter frequency and plotting its phase and magnitude response until a fit that best matched the measured magnitude and phase was achieved. Figure 6.10 shows the measured normalised magnitude and phase of the sub-system and the fitted transfer function. Some deviation in the data at the higher frequencies, as seen in Figure 6.10, was expected as the Bessel filter fit was optimised around the flutter frequency (nominally 6.0*Hz*.) of the FAST model. With reference to Equation 3.112, it was determined from the Bessel filter approximation of the data that ω_a = 5.75*Hz*. and ζ_a = 0.91. To validate the calculated analytical transfer function of the subsystem, the measured response of the control surface to a step input was compared to the step response predicted by the 2nd order Bessel Filter. Figure 6.11 shows the good agreement between the analytical transfer function and the actual response of the control surface. It can be seen that the 2nd order Bessel Filter is slightly conservative, as the actual step response of the control surface is faster than the response predicted using the analytical transfer function. This result was desirable to ensure a safe and robust control system design. It was also important to quantify any system lag, which would reduce the effectiveness of the controller. From the measured response of the system to a step input it was determined that the system had a rise time of 72m*s* and a settling time of 133m*s*, and importantly that there was no appreciable lag in the system in responding to the step input.

Table 6.2 Faulhaber™ LM1247-020-01 Linear Motor Settings used During Testing

Parameter	Value	Units
Proportional Term (POR)	25	-
Integral Term (I)	25	-
Peak Current Limit (LPC)	3000	*mA*
Continuous Current Limit (LCC)	1500	*mA*
Acceleration (AC)	32000	mm/s^2
Deceleration (DEC)	32000	mm/s^2
Maximum Speed (SP)	30000	*mm/s*
Gain of Proportional Controller (PP)	30	-
D Term of Position Controller (PD)	10	-
I Term of Current Controller (CI)	1	-
Sampling Rate (SR)	1	1/10m*s*

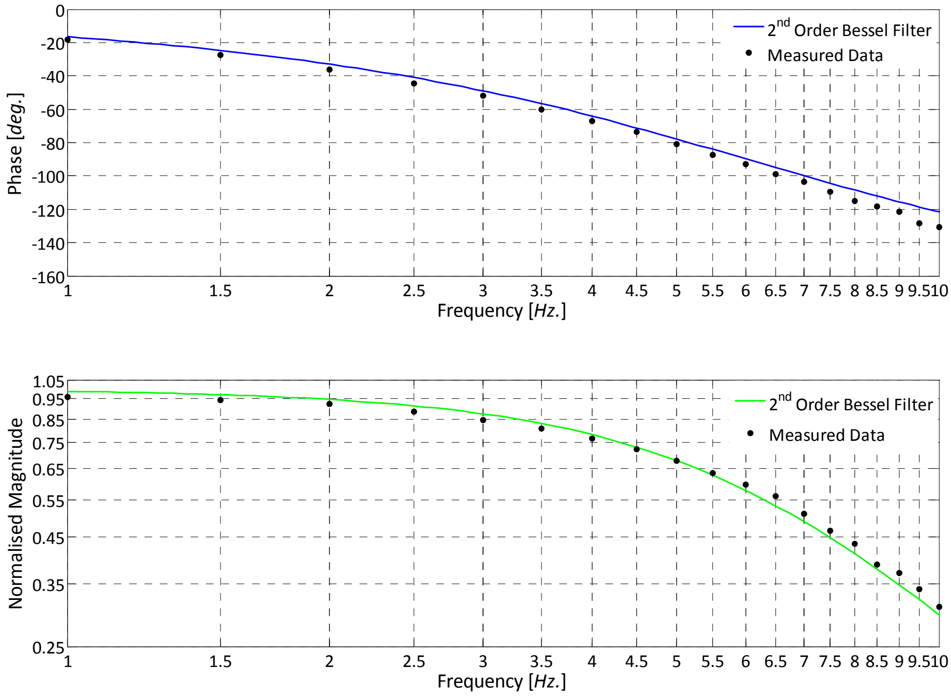


Figure 6.10 Control Surface and Control Surface Actuation System Bode Diagram

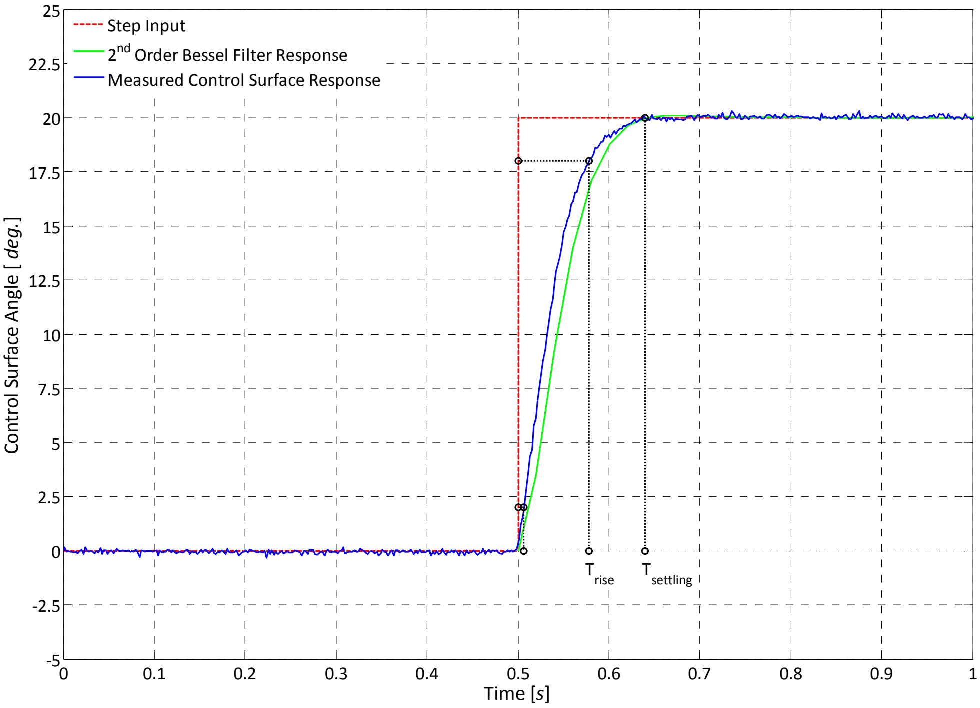


Figure 6.11 Step Response of Control Surface and Control Surface Actuation Mechanism

6.2 Simulated Open- and Closed-Loop FAST Model Responses

Prior to conducting wind tunnel tests, extensive simulations to determine the open- and closed-loop response of the FAST model were done using models developed in both MATLAB™ and Simulink™. The full MATLAB™ and Simulink™ models are given in Appendix C and Appendix D respectively. The purpose of these simulations was to establish the open-loop dynamics of the model, and thereafter to evaluate the effect and robustness of the control law in suppressing flutter. The majority of simulations were done using a state-space model in MATLAB™, as this was the model used in the control law design. The Simulink™ model was used as an additional check once a feedback and Kalman gain matrix had been calculated using the MATLAB™ state-space model. The advantage of the Simulink™ model was that it could be used to investigate the effect of nonlinearities such as free-play, dead-band and time delays, that could not be modelled easily in the state-space system. Extensive simulations were performed during the initial design of the FAST model controller, but only selected representative examples of the final controller design for the FAST model in a baseline cg position configuration are included here.

Open-Loop FAST Model Modal Damping and Frequency Response

Initially, the open-loop modal dampings and frequencies of the FAST model for the three cg position configurations were calculated by performing an eigenvalue analysis of the MATLAB™ state-space model of the aeroservoelastic system. They were calculated and plotted over an airspeed range in which flutter was anticipated at a fixed air density. The theoretical open-loop flutter speeds were established from the points at which the plots of modal damping passed through zero. The theoretical flutter speed for the baseline cg position configuration was calculated as 23.51*m/s* at a frequency of 5.98*Hz*., as shown in Figure 6.12. For the forward cg position the calculated flutter speed was 20.18*m/s* at a frequency of 6.06*Hz*., shown in Figure 6.13, and for the aft cg position the calculated flutter speed was 29.75*m/s* at a frequency of 6.12*Hz*. shown in Figure 6.14. A *U-g* flutter analysis, outlined in Appendix A, was also performed for comparison with the *P* flutter analysis. The results of this analysis are not included here, but simply listed in Table 6.6.

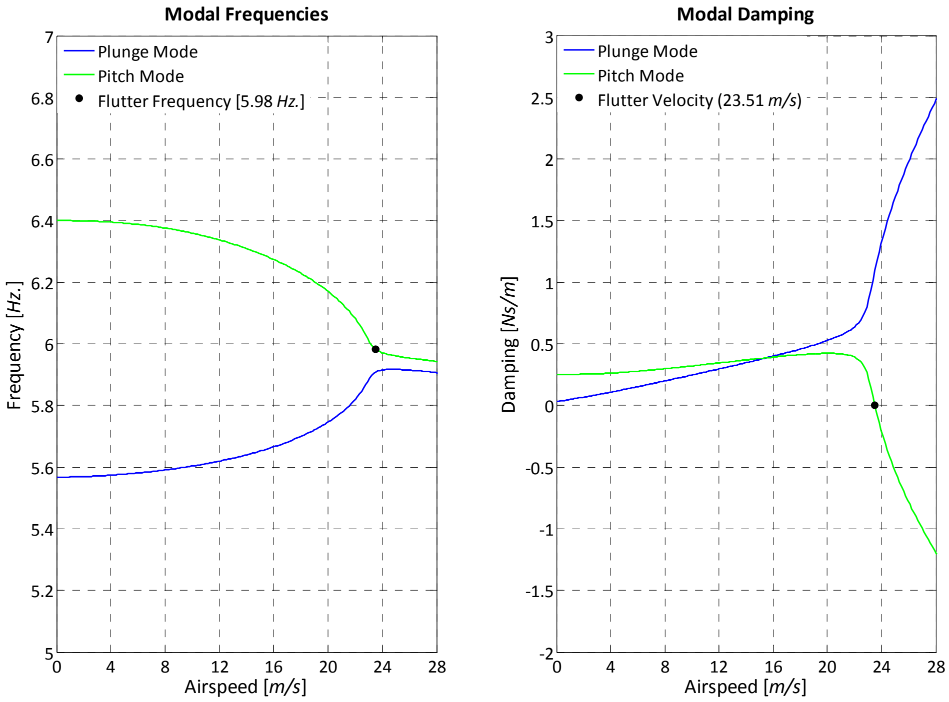


Figure 6.12 Calculated Open-Loop Damping and Frequency Response (Baseline cg Position)

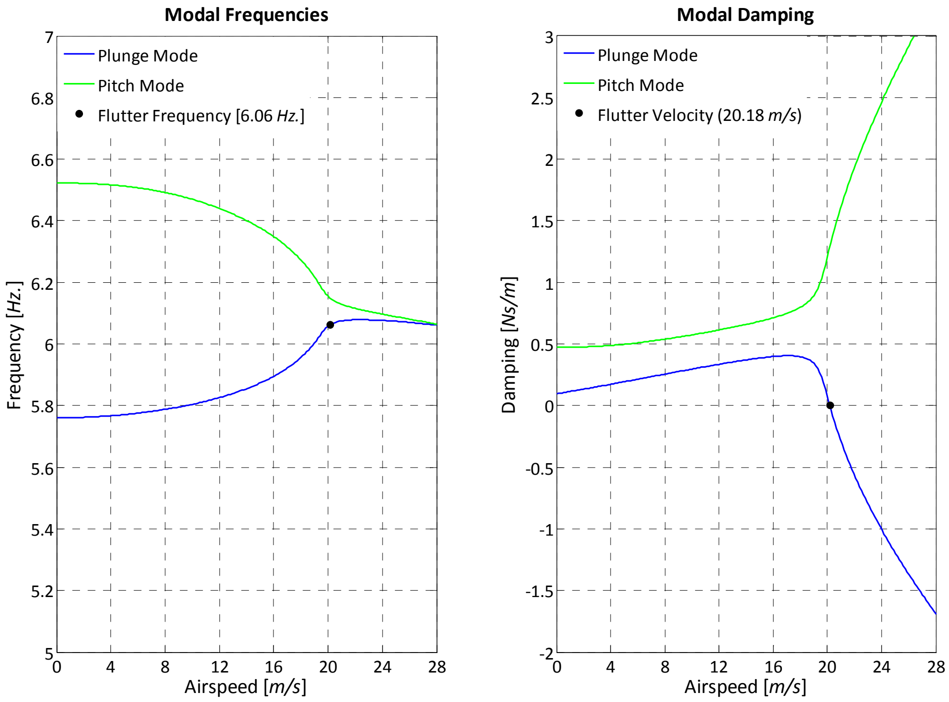


Figure 6.13 Calculated Open-Loop Damping and Frequency Response (Forward cg Position)

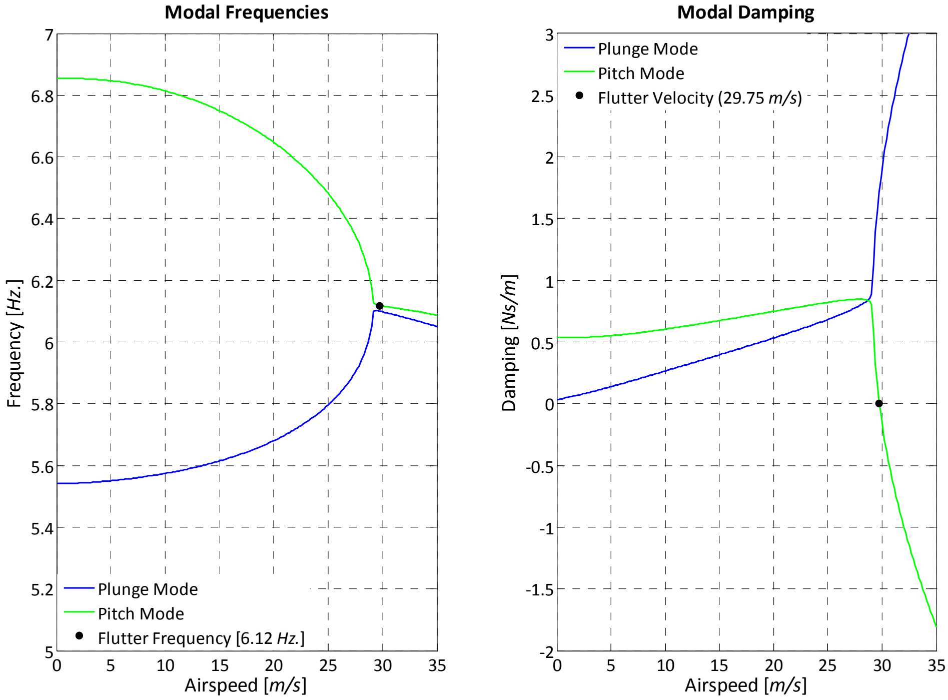


Figure 6.14 Calculated Open-Loop Damping and Frequency Response (Aft cg Position)

Closed-Loop FAST Model Simulations

To evaluate the effectiveness and robustness of the controller at the critical flutter speed of the model, its responses to an initial displacement, an impulse input and a step input were simulated with the MATLAB™ state-space model. Representative closed-loop simulations of the FAST model in the baseline cg position configuration are shown in Figure 6.15 for the initial displacement, Figure 6.16 for the impulse input and Figure 6.17 for the step input. The simulations indicated that the system stabilises in approximately 1.0*s* (< 6 cycles) and that the Kalman state estimates matched the simulated measured states extremely well. The anticipated response of the system to the application of the control during a developed flutter cycle at the critical flutter speed, calculated with the Simulink™ model, is given in Figure 6.18, which predicts that flutter can be completely suppressed in under 1.0*s* with large initial control surface movements. Further simulations were performed in Simulink™ to determine the effect of nonlinearities on the response and robustness of the FAST model controller. Nonlinearities investigated were the effect of actuator saturation (±20.0°), dead-band (±0.25°) and rate limitations (±120.0°/*s*) on the closed-loop response of the model. The effect of a fixed time delay between the actuator command signal and the control surface response on the controller performance was also investigated using the Simulink™ model. It was shown that the controller was robust to small time delays and was able to effectively suppress flutter. Larger time delays (> 100m*s*) caused the controller to become out of phase with the flutter and enhance unsteady oscillations of the system at supercritical airspeeds. It was thus ensured during wind tunnel testing that the control system software operated at its maximum possible speed to minimise any inherent time delays in the FAST model.

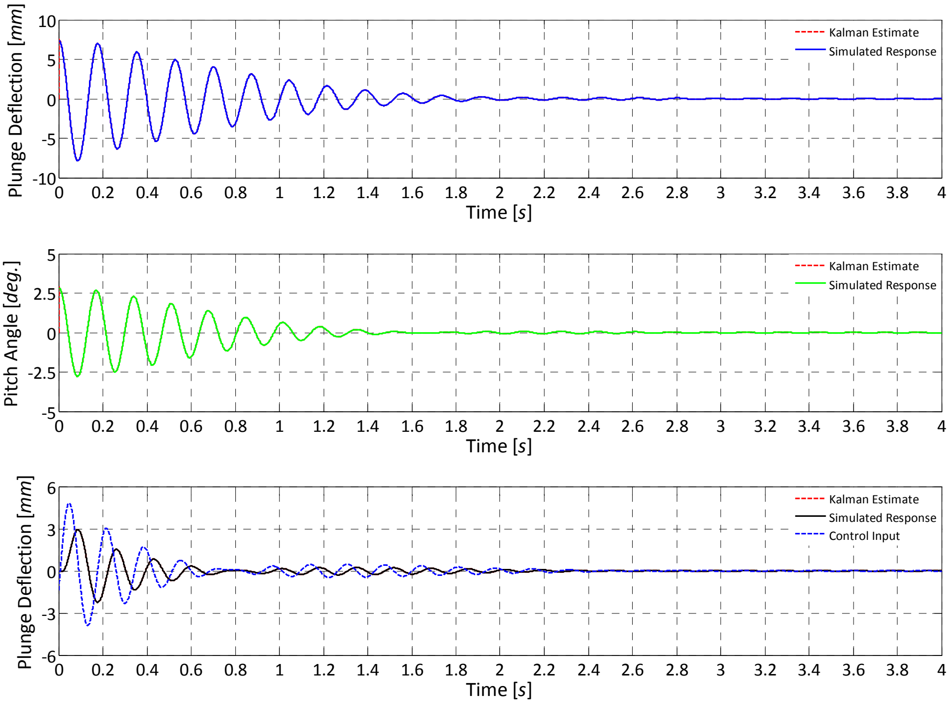


Figure 6.15 MATLAB™ Simulated FAST Model Response to an Initial Displacement

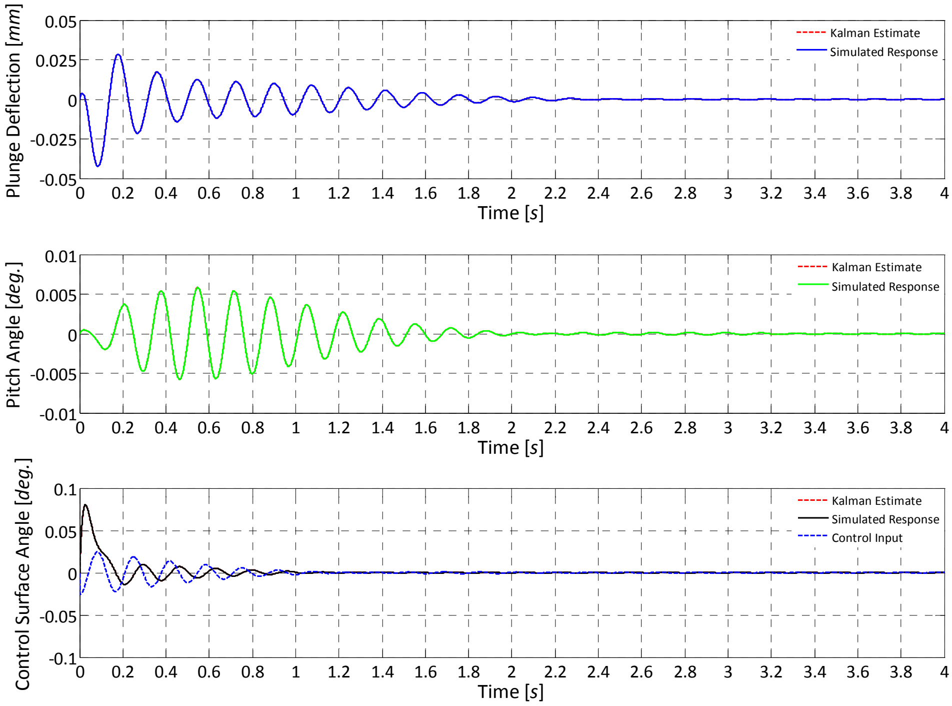


Figure 6.16 MATLAB™ Simulated FAST Model Response to an Impulse Input

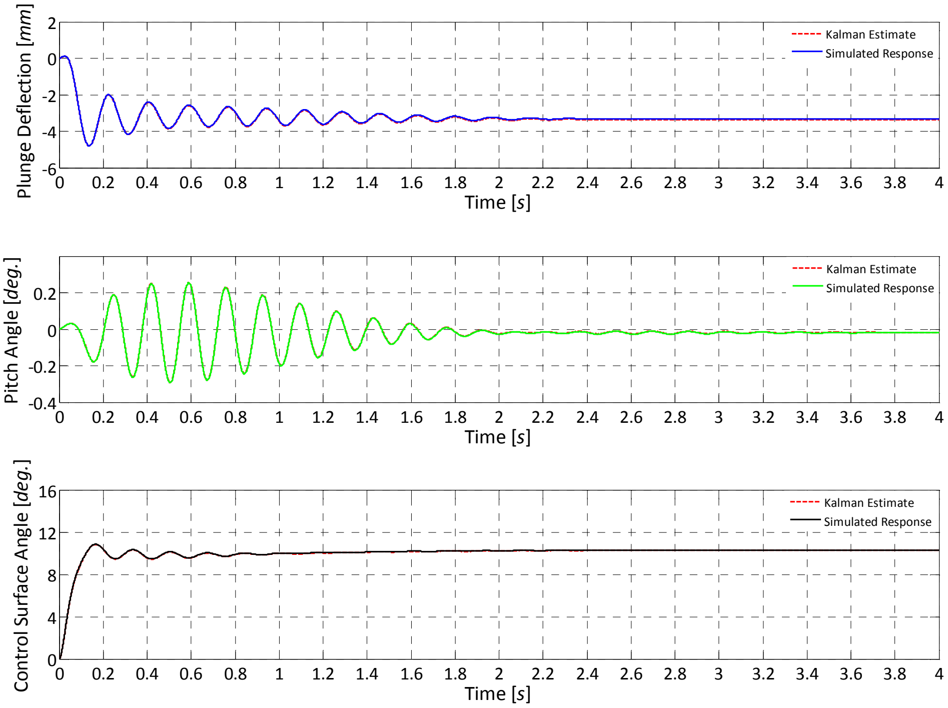


Figure 6.17 MATLAB™ Simulated FAST Model Response to a Step Input

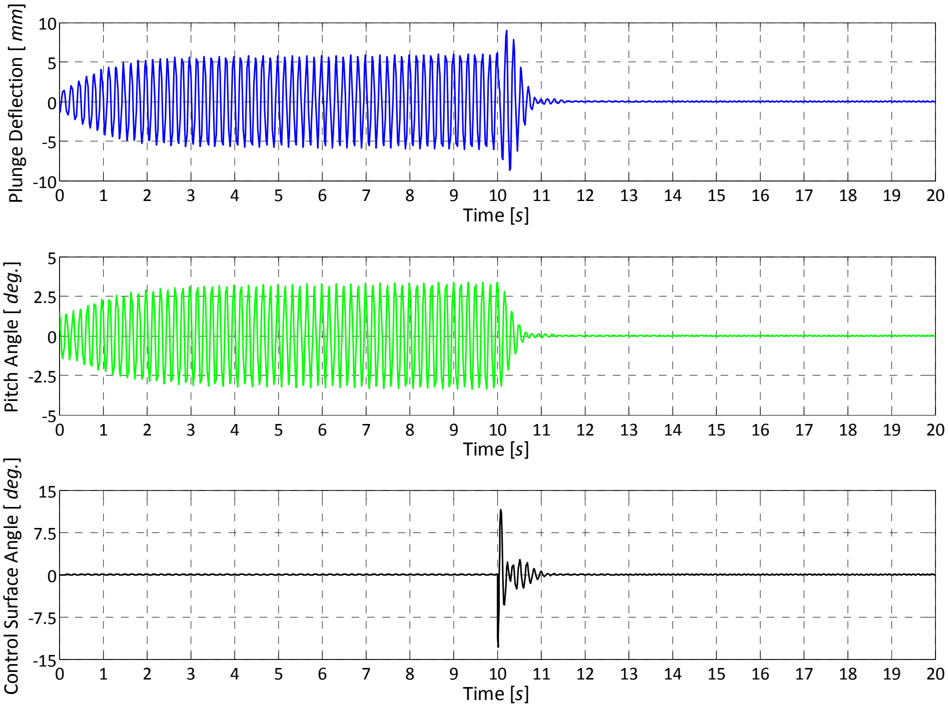


Figure 6.18 Simulink™ Model Output of Control Applied During Critical Flutter

6.3 Wind Tunnel Test Results

Results of wind tunnel tests with the FAST model in both the CWT and LSWT of the CSIR are presented and discussed in this section. Unexpected behaviour of the model during the initial CWT tests is discussed in Section 6.3.1, followed by presentation and discussion of CWT open-loop test results, for the model in each of the cg position configurations, in Section 6.3.2. CWT closed-loop flutter test results, again for the model in each of the cg position configurations, are presented and discussed in Section 6.3.3. Limited open- and closed-loop LSWT test results with the model in the baseline cg position are given and discussed in Section 6.3.4.

6.3.1 Initial Wind Tunnel Results

During initial wind tunnel runs in the CWT an unexpected nonlinear aerodynamic phenomenon was observed, in which the wing displayed a type of LCO at an airspeed slightly below its calculated flutter speed. An example of this response, at an airspeed of 23.0m/s, can be seen in Figure 6.19. It was apparent that both the plunge and pitch modes of the FAST model were excited at their natural frequencies, as indicated by the two widely spaced peaks in the derived PSD plots shown in Figure 6.20. The almost constant plunge and pitch amplitudes of the FAST model wing are characteristic of a LCO. The LCO behaviour of the model indicates a nonlinearity in the system that is not consistent with binary flutter, where only one coupled structural mode at a single frequency would be observed. Nonlinear structural dynamics were eliminated based on the results of the SGB calibrations which showed the flexure mount to be highly linear within the displacement amplitudes of the wing under the applied aerodynamic loading. It was thought that low Reynolds number flow resulting from the small chord and smooth surface of the model wing, combined with the low wind tunnel speeds, was causing laminar flow separation. Under these conditions a situation can arise where laminar separation bubbles (alternating between the top and bottom surfaces of the wing) set up an unsteady LCO with very different dynamics to two-dimensional binary flutter (Poirel *et al.*, 2008). To verify the assertion of laminar flow separation, a mixture of oil, oleic acid and titanium oxide (TiO_2) was applied to the upper surface of the model to visualise the flow to establish if it was separating, and if so, where. With the wing held fixed, this technique revealed that the flow was separating at approximately 25% of the chord length and not re-attaching, as seen in Figure 6.21. This result eliminated laminar separation bubbles as the cause of the observed model oscillations, but revealed that the flow over the model was not as modelled, an observation that otherwise may have been overlooked. To re-attach the flow over the whole wing, serrated tape trip strips were added to the top and bottom surfaces of the wing at 5% of chord length across the whole span. In addition, vortex generators in the form of small bent metal tabs were added to the top and bottom surfaces of the wing to ensure that the flow over the control surface remained attached. The trip strip and vortex generators on the upper surface of the wing are shown in Figure 6.22.

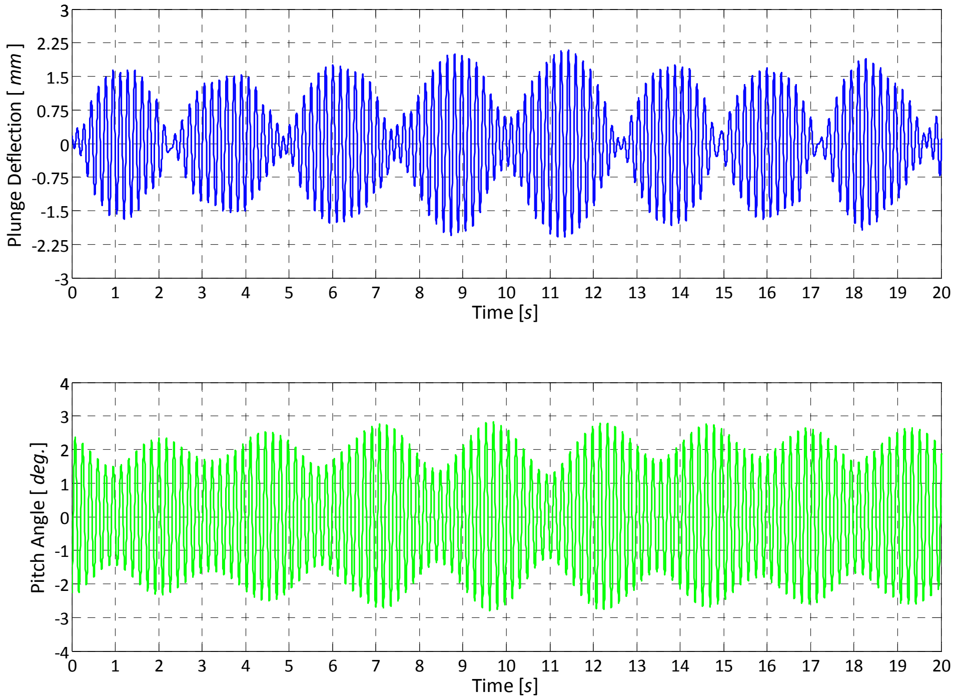


Figure 6.19 Unsteady Aerodynamics Induced Limit Cycle Oscillation ($U_{LCO} \approx 23.0m/s$)

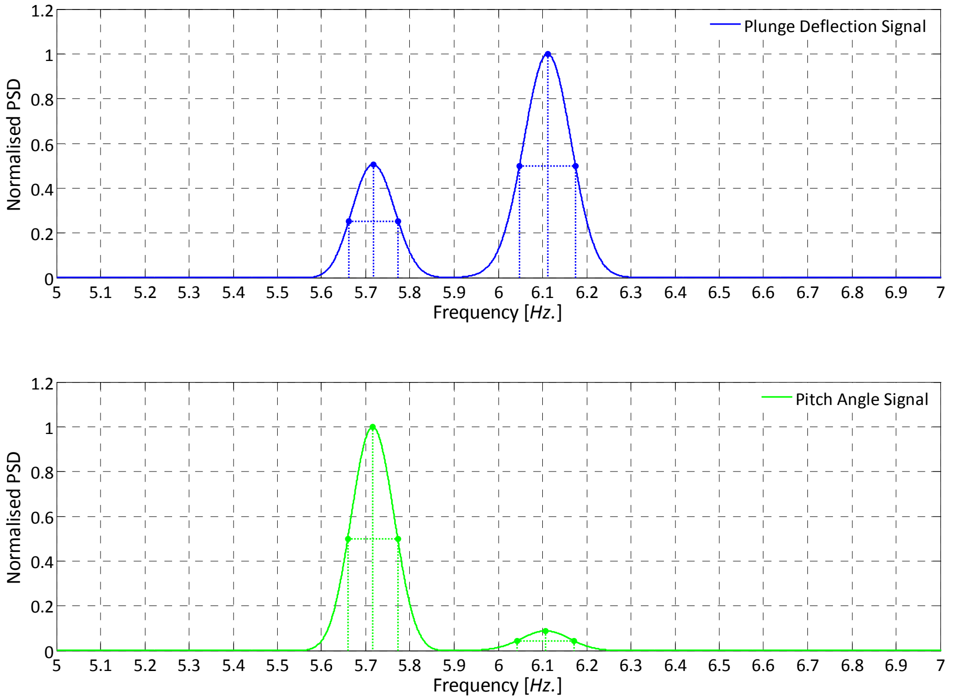


Figure 6.20 Normalised PSD of Limit Cycle Oscillation ($U_{LCO} \approx 23.0m/s$)

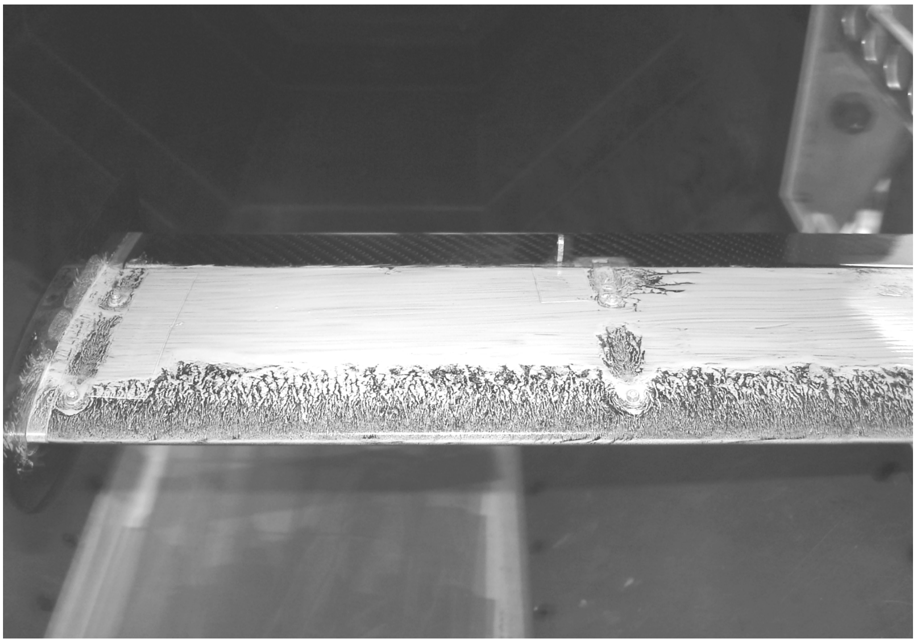


Figure 6.21 FAST Model Leading Edge Laminar Flow Separation

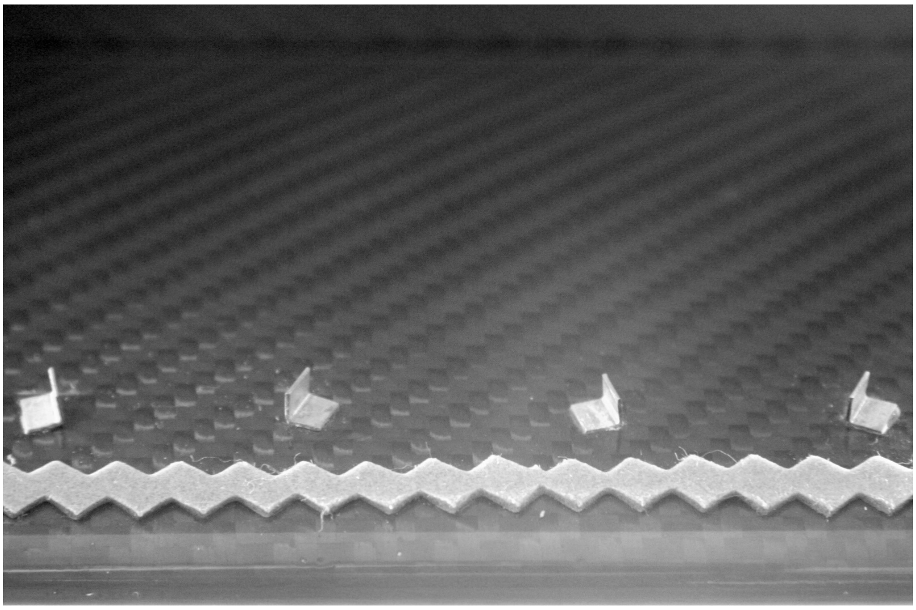


Figure 6.22 Leading Edge Trip Strip and Vortex Generators

Using the oil flow visualisation technique again after the modifications established that the trip strips attached the flow well, as shown in Figure 6.23, but during subsequent wind tunnel testing the LCO type oscillation persisted. The only likely system nonlinearity left to investigate was a small in-flow of air over the wing root, revealed by the oil flow visualisation technique and shown in Figure 6.24. This flow originated from a cut out made in the wind tunnel side wall to allow clearance for free movement of the FAST model mount flexures. It was originally thought that this in-flow of air was small and that its effect on the flutter dynamics of the model would be negligible, because of a large endplate mounted to the wing root. However, upon re-investigation it was deduced that this was not the case. It was found that the motion of the wing, coupled with the additional airflow over the wing root, set up unsteady pressure fluctuations that were exciting the natural modes of the FAST model and causing the observed LCO.

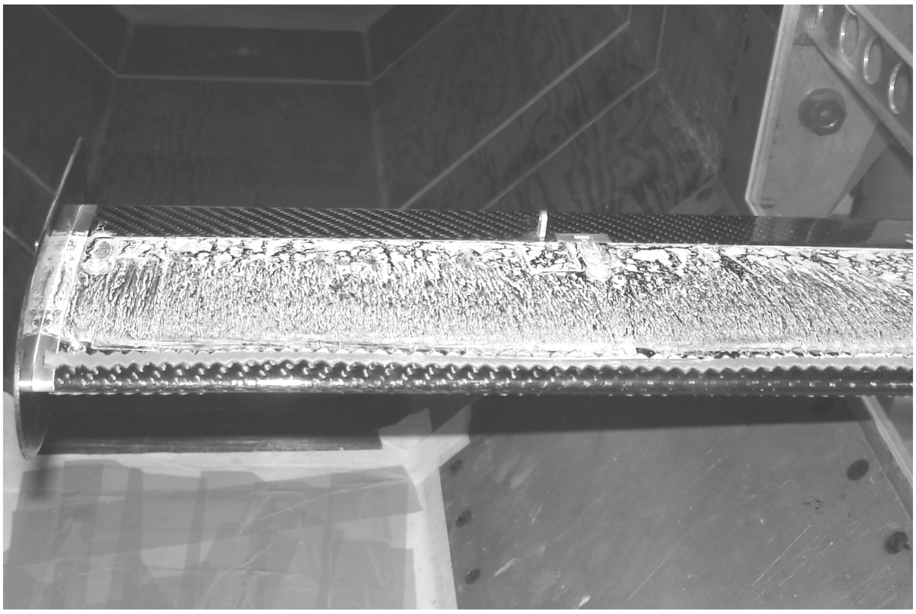


Figure 6.23 FAST Model Leading Edge Trip Strip and Attached Flow

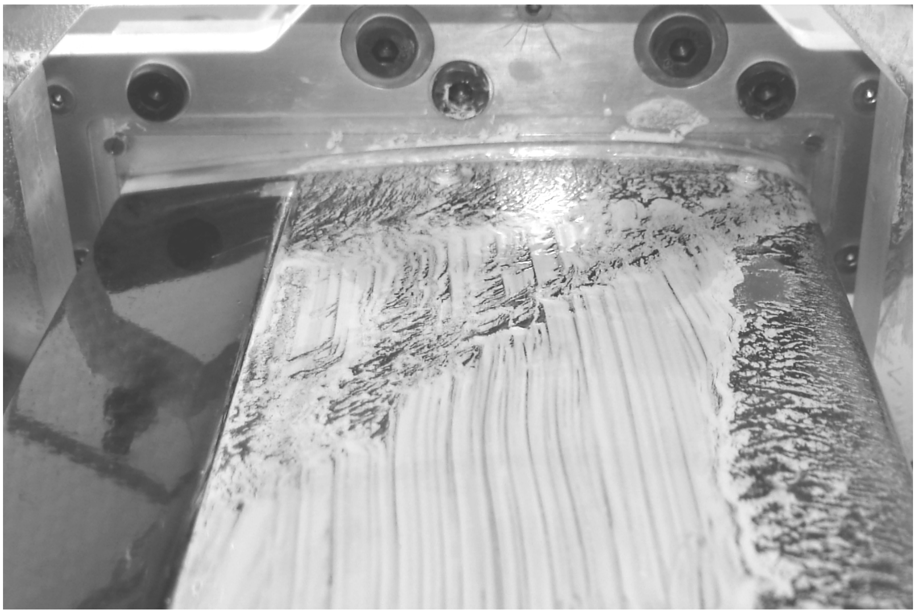


Figure 6.24 Air Flow over FAST Model Wing Root

Two methods of preventing this additional airflow from entering the wind tunnel test section and so eliminating its undesired effect on the model were considered. The first was to construct a plenum chamber around the model mount, to seal the whole model off from the ambient air to ensure no air could be drawn into the wind tunnel test section over the model. However, this idea was rejected because of its complexity, cost and limited time available for subsequent wind tunnel testing. Instead, a simpler double splitter plate and control surface end-plate method was implemented to eliminate the undesired airflow and pressure fluctuations.

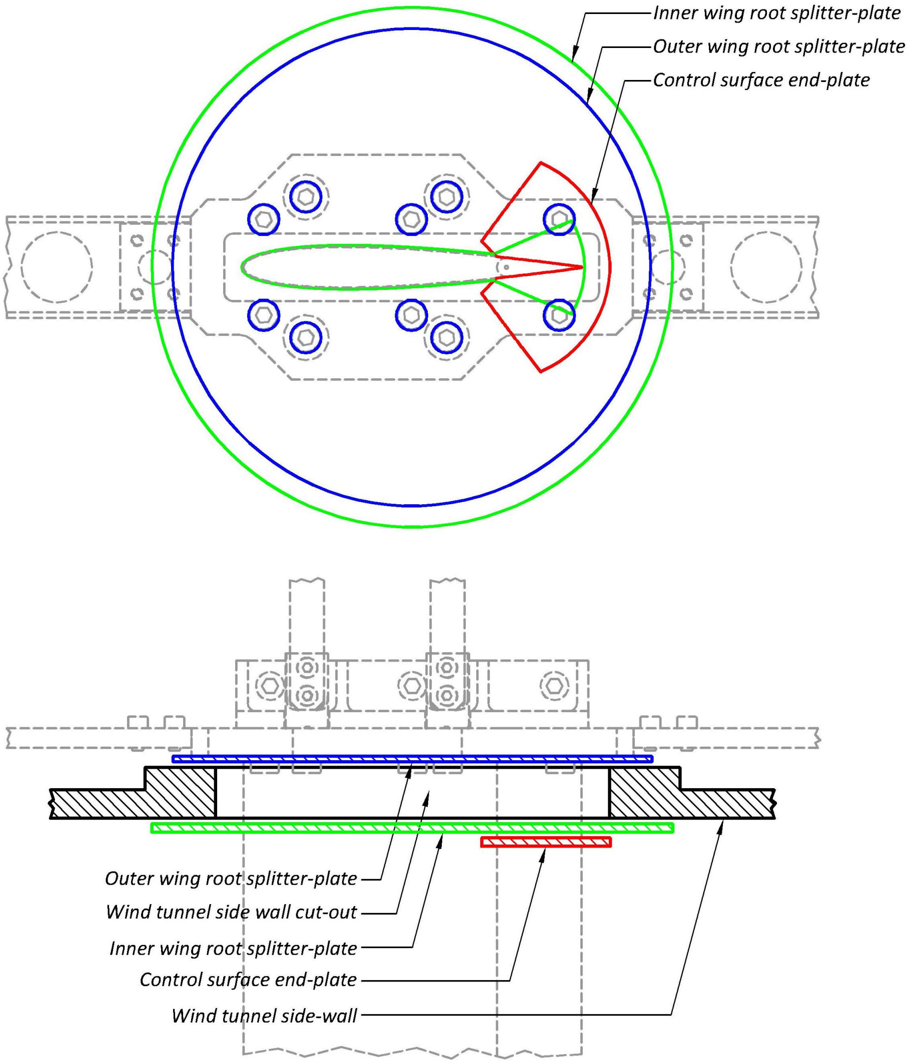


Figure 6.25 FAST Model Wing Root Splitter-Plates and Control Surface End-Plate

Referring to Figure 6.25, this solution was achieved by attaching an additional splitter-plate to the model at the wing root, such that the wind tunnel side wall was sandwiched between two splitter-plates. This ensured that outside air flowing into the wind tunnel test section did not pass over the wing but was forced to flow down the boundary layer along the wind tunnel side wall. To make this solution effective, the gaps between wind tunnel side wall and both the inner and outer splitter-plates were minimised to limit the amount of ambient air entering the test section. A cut-out was made in the new inner splitter-plate to allow the control surface to move freely. This cut-out, although small, still allowed some ambient air to enter the test section over the wing root. To eliminate this, a small end-plate was attached to the control surface to ensure that as it moved no gaps would open in the inner splitter-plate attached to the wing root. Subsequent wind tunnel testing showed that the new wing root splitter-plate was very effective and eliminated the unsteady pressure fluctuations, and the FAST model behaved exactly as originally anticipated. The model, as tested in the CWT with the wing root splitter plate, control surface end-plate and wing tip end-plate attached is shown in Figure 5.3.

6.3.2 Open-Loop Calibration Wind Tunnel Tests

Once the undesired effects of the pressure fluctuations on the FAST model had been eliminated, wind tunnel flutter testing of the model in the CWT continued. The purpose of the open-loop wind tunnel tests was to validate the mathematical model of the system, as this was the basis of the design of an appropriate controller. The open-loop results also allowed semi-empirical modification of the mathematical model of the system to ensure a robust flutter suppression controller design for closed-loop testing. Extensive testing was performed for the baseline cg position configuration, whilst less comprehensive testing was done for the forward and aft cg position configurations.

In the baseline cg position configuration both ballast masses were attached to the ends of the ballast arms, giving $m = 1.85kg$, $I_\alpha = 30\text{x}10^{-3}kgm^2$ and $x_\alpha = 0.200$. For the forward cg position configuration the rear ballast mass was removed which reduced the mass of the FAST model by 2.7% to 1.80*kg*, shifted the cg position to $x_\alpha = 0.185$ (-7.5%) and reduced the mass inertia by 25.0% to nominally $22.5\text{x}10^{-3}kgm^2$. For the aft cg position configuration the forward ballast mass was repositioned back along the ballast arm to the leading edge of the wing to move the cg position to $x_\alpha = 0.215$ (+7.5%) whilst again reducing the mass inertia by 25.0% from the baseline value, to nominally $22.5\text{x}10^{-3}kgm^2$. These changes in mass properties and cg position altered the modal frequencies and the gap between them, thus altering the flutter dynamics and speed from those of the baseline cg position configuration of the FAST model. It is apparent that it was not strictly a pure change in cg position that was investigated as the mass and mass inertias of the FAST model were different for the cg position configurations. That said, the principle was simply to change model parameters to demonstrate the ability to mathematically model the aeroservoelastic system and suppress flutter with the model in different configurations. For simplicity, the three configurations have been referred to as the baseline, forward and aft cg configurations as the cg position was the primary FAST model parameter that changed.

In initial open-loop testing the PSDs of the wind-off response of the model in each cg position configuration were measured to determine the natural pitch and plunge frequencies of the model on the wind tunnel mount. These values were used in the flutter predictions and mathematical modelling of the system. In the results that follow the measured control surface angle is not presented, since for the open-loop tests the control surface was held fixed in a neutral position by the control surface actuator. In conducting the tests, the predicted open-loop flutter speed was approached safely by measuring the modal frequencies at subcritical wind tunnel speeds, calculating the simplified Zimmerman flutter margins and plotting them as a function of airspeed. This plot was updated and extrapolated to zero after each test point, to give the predicted flutter onset airspeed, which was compared with the calculated flutter speed as the test progressed. Once deemed safe, the wind tunnel speed was increased to the critical flutter speed and the FAST model was allowed to flutter unrestrained whilst data was recorded. Selected results from these CWT flutter tests are presented below for each cg position configuration in turn. The section ends with a comparison in Table 6.6 between the open-loop measured wind tunnel results and the predictions of the *P* (Section 3.1.6) and *U-g* (Appendix A) flutter analysis solutions.

Baseline Center of Gravity Position Configuration

The baseline cg position configuration of the FAST model, for which measured and estimated parameters are listed in Table 6.3, was tested first. The natural plunge and pitch frequencies and dampings of the FAST model as mounted in the CWT were established from the PSDs (Figure 6.27) of the recorded wind-off time response (Figure 6.26) of the model after it had been given an initial displacement and then allowed to oscillate freely.

Table 6.3 CWT FAST Model Parameters (Baseline cg Position Configuration)[14]

Flutter Model Parameter	**Symbol**	**Value**	**Units**
Effective wingspan	s_w	0.335	*m*
Distance between mid-chord and elastic axis in semi-chords	a	-0.2	-
Reference semi-chord	b	0.06	*m*
Control surface position in semi-chords	c	0.5	-
Distance between aerofoil cg and elastic axis in semi-chords	x_α	(0.2)	-
Distance between control surface cg and hinge axis in semi-chords	x_β	-0.075	-
Wing mass	m	1.85	*kg*
Mass inertia of wing (about wing elastic axis)	I_α	(30.0×10^{-3})	kgm^2
Mass inertia of control surface (about hinge axis)	I_β	(45.0×10^{-6})	kgm^2
Natural plunge frequency	f_h	5.67	*Hz.*
Natural pitch frequency	f_α	6.27	*Hz.*
Natural control surface frequency	f_β	5.75	*Hz.*
Plunge damping coefficient	c_h	(0.025)	-
Pitch damping coefficient	c_α	(0.05)	-
Control surface damping coefficient	c_β	(0.05)	-

---

[14] Bracketed data indicates estimated parameters.

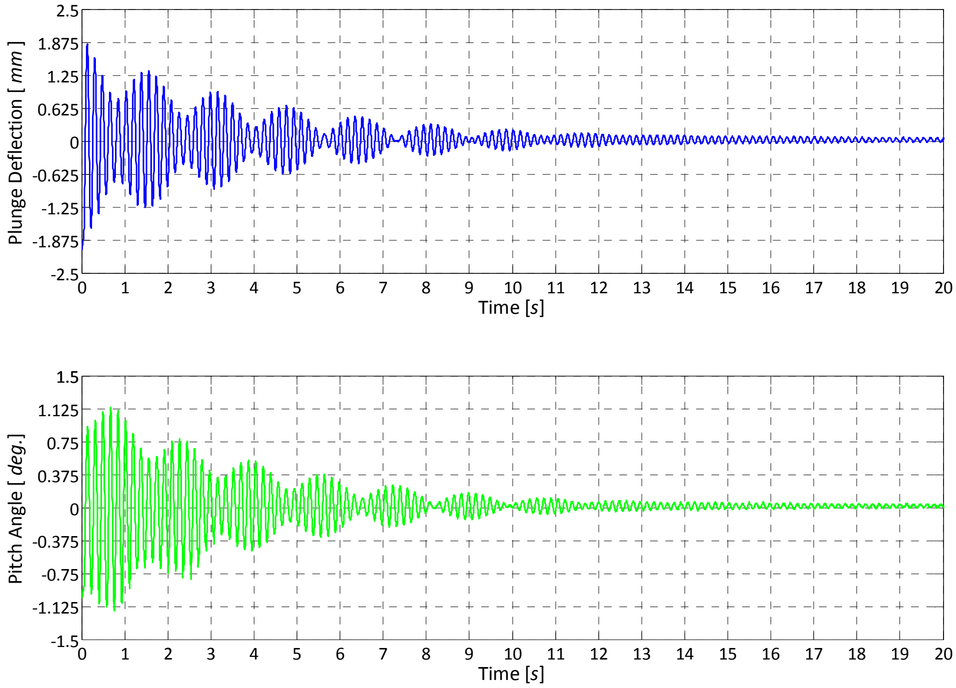


Figure 6.26 FAST Model Open-Loop Time Response (Baseline cg Position; *U* = 0.0*m*/*s*)

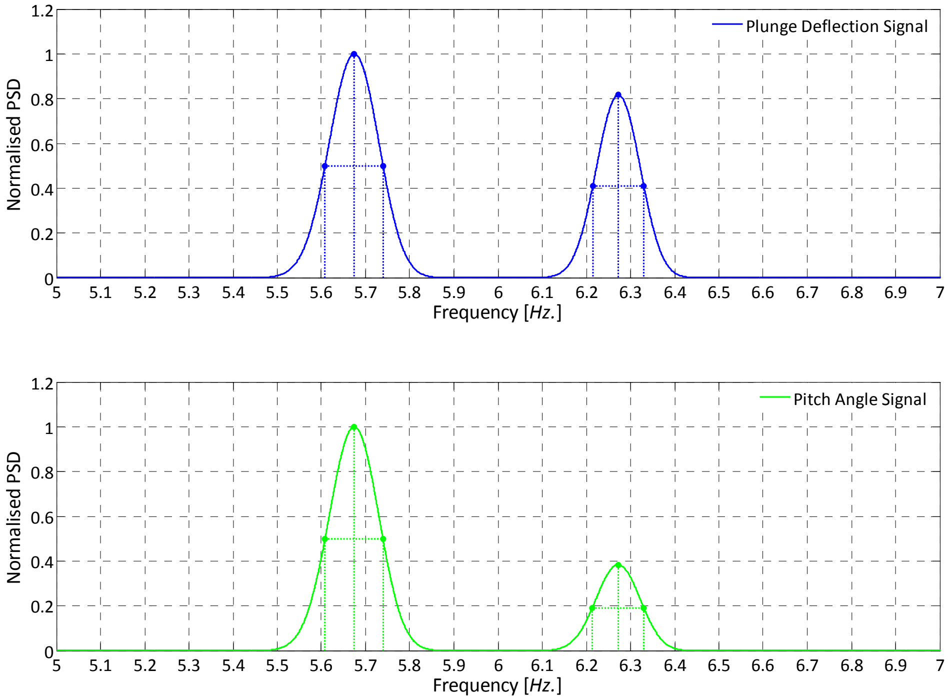


Figure 6.27 Normalised PSD of Open-Loop Response (Baseline cg Position; *U* = 0.0*m*/*s*)

At a wind tunnel speed of 23.0*m/s* gentle flutter occurred and was sustained at a frequency of 5.8*Hz*., as shown in the photograph sequence in Figure 6.28 and by the model time response and PSD plots in Figure 6.29 and Figure 6.30 respectively. Although the plunge displacement and pitch angle of the FAST model was small when it fluttered, it is still apparent in Figure 6.28 that the FAST model exhibited classical binary pitch-plunge flutter. With reference to the red and green dotted lines in Figure 6.28 (which respectively indicate the trim positions of the FAST model wing and ballast arms before flutter was encountered) it is evident that the model pitched about its elastic axis whilst simultaneously translating vertically at a single frequency. This response is characteristic of classic binary pitch-plunge flutter.

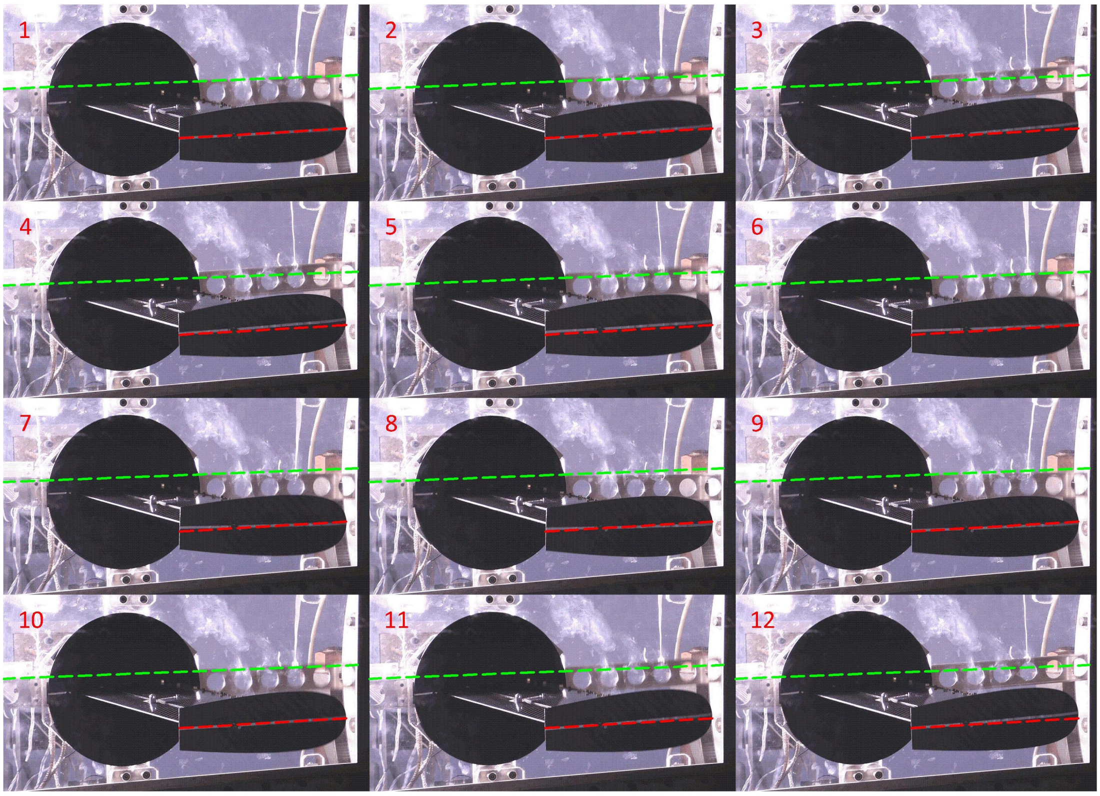


Figure 6.28 Open-Loop Flutter Cycle (U_f = 23.0*m/s*, *f* = 5.8*Hz.*)

The measured flutter speed and frequency agreed well with the prediction of 23.5*m/s* at a frequency of 6.0*Hz*. from the *P* flutter analysis method (see Table 6.6). In comparison, the predicted flutter speed and frequency using the *U-g* flutter analysis method were 21.7*m/s* and 6.0*Hz*. respectively. At the critical flutter speed the flutter was gentle, modal oscillations were small (±1.0*mm* in the plunge mode and ±2.3° in the pitch mode) and showed no tendency to diverge. Combined power spectral density plots from the plunge SGB signals (Figure 6.31) and from the pitch SGB signals (Figure 6.32) illustrate how the modal damping changed and the modal frequencies converged to the flutter frequency of the model. Since the FAST model fluttered in its pitch mode, it was expected that frequency peaks in the pitch signal PSDs would be clearer than those in the plunge signal PSDs, as is evident in Figure 6.31 and Figure 6.32.

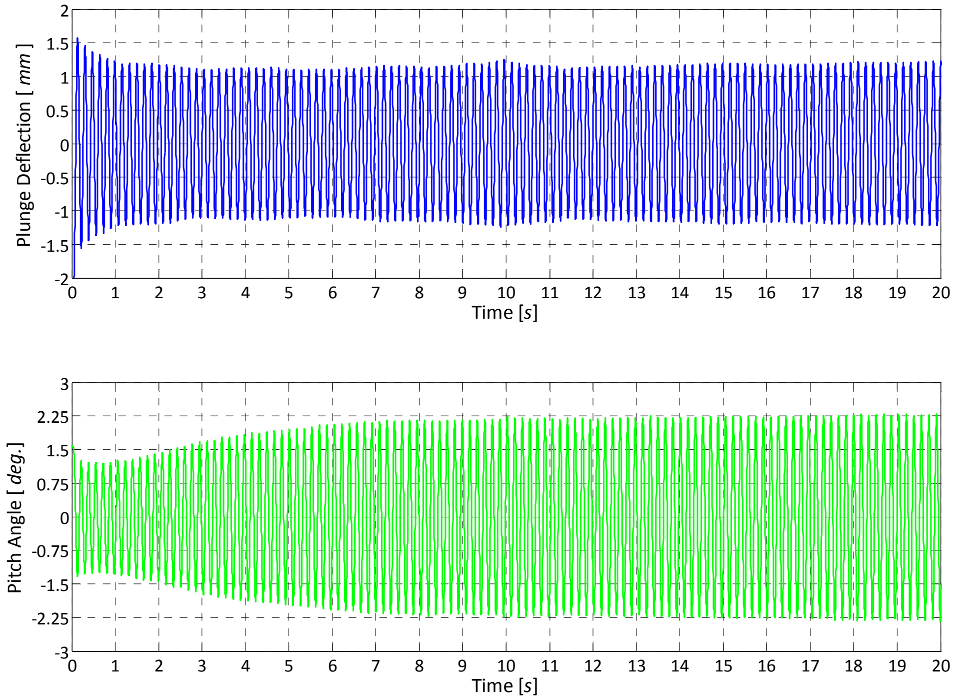


Figure 6.29 FAST Model Open-Loop Flutter (Baseline cg Position; U_f = 23.0m/s)

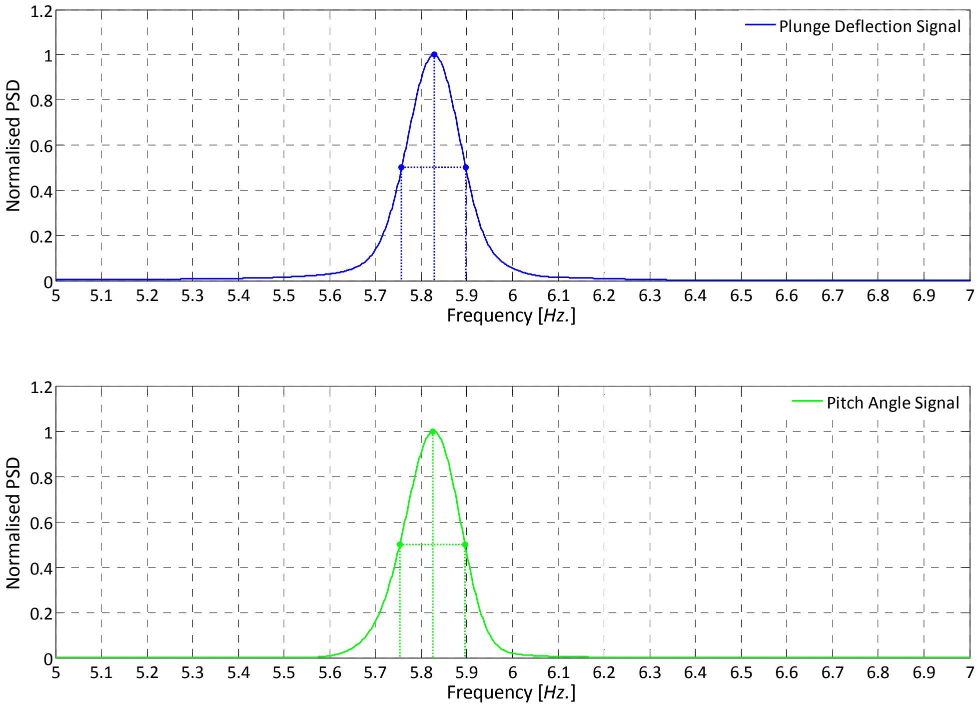


Figure 6.30 Normalised PSD of Open-Loop Flutter (Baseline cg Position; U_f = 23.0m/s)

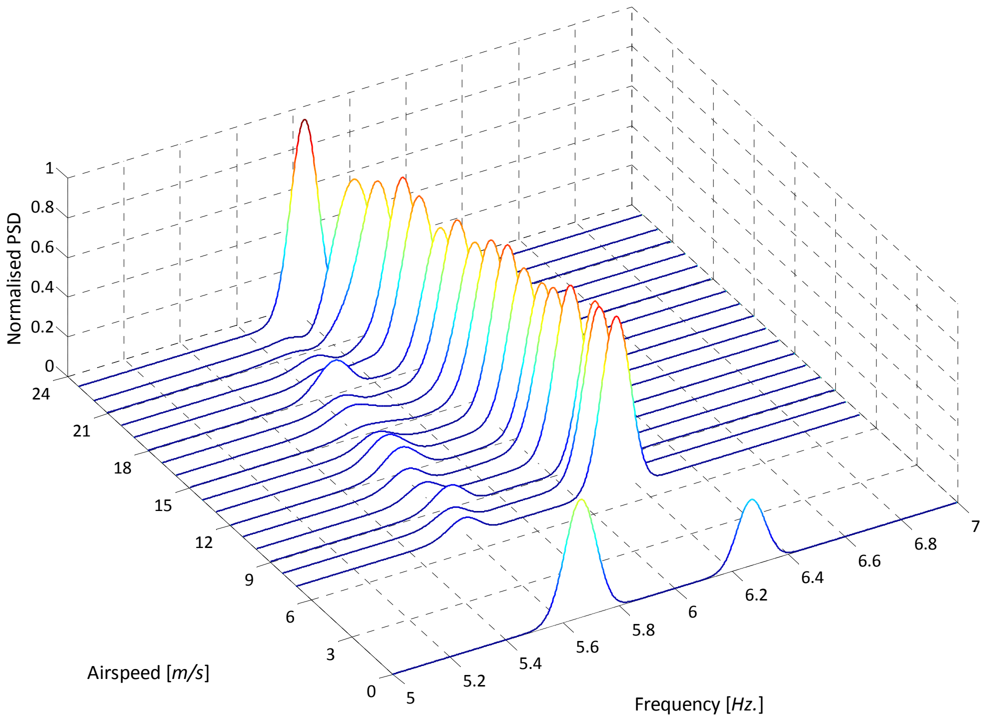


Figure 6.31 Combined Normalised Open-Loop PSDs from Plunge SGB (Baseline cg Position)

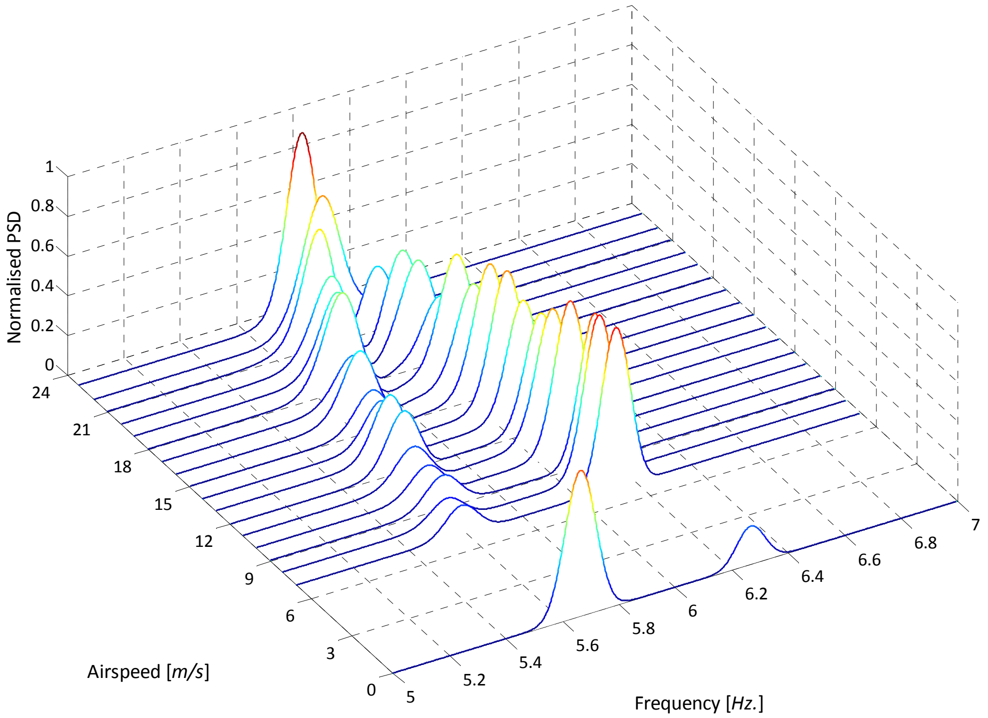


Figure 6.32 Combined Normalised Open-Loop PSDs from Pitch SGB (Baseline cg Position)

Predicted and measured modal frequencies corresponding to the combined and averaged peaks in Figure 6.31 and Figure 6.32, plotted as a function of airspeed, are presented in Figure 6.33. This figure shows the slow convergence of the two modal frequencies up until coalescence, indicating the gentle nature of the flutter. The measured modal frequencies matched the predicted values well, with the measured plunge frequencies being within 0.9% and pitch frequencies within 3.2% of the predicted values. The simplified Zimmerman flutter margins, calculated for each wind tunnel test point, are plotted as a function of airspeed in Figure 6.34. As the test progressed, a continuously updated polynomial curve fit through these points was extrapolated to a zero flutter margin to safely approach the flutter speed, which was finally predicted as 25.7*m*/*s*. This over prediction of the flutter speed by 11.7% was expected as the simplified Zimmerman flutter margin criterion excludes modal structural damping and therefore typically gives a non-conservative flutter speed prediction (Bennett, 1982). Furthermore, the closest subcritical simplified Zimmerman flutter margin calculated was at an airspeed nominally 2.5*m*/*s* below the actual observed flutter speed of the model. This implies that actual flutter occurred 12.2% beyond the last data point used in the extrapolation of the polynomial curve used to predict the onset of flutter.

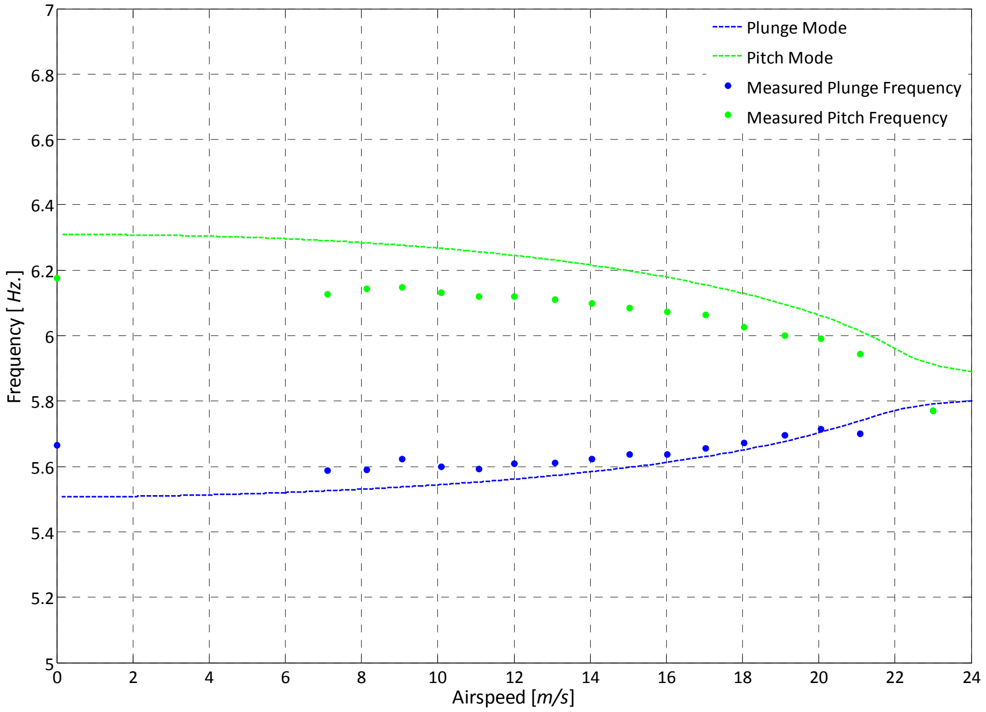


Figure 6.33 Predicted and Measured Open-Loop Modal Frequencies (Baseline cg Position)

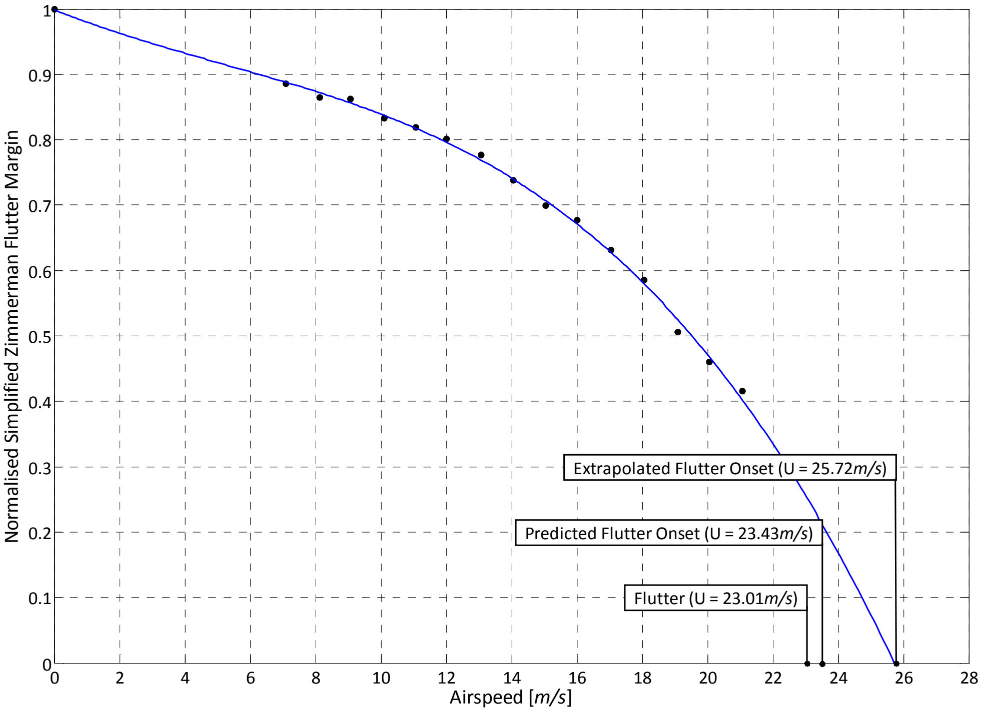


Figure 6.34 Open-Loop Simplified Zimmerman Flutter Margins (Baseline cg Position)

Forward Center of Gravity Position Configuration

The measured and estimated model parameters used in the design of the control law, simulations and tests for the forward cg position configuration are listed in Table 6.4. The wind-off time response and the corresponding PSD for each mode, from which the natural plunge and pitch frequencies of the model/mount were obtained for use in the flutter prediction, are given in Appendix G. In this configuration the model fluttered at 21.4*m/s* at a frequency of 6.1*Hz*. which was in reasonable agreement with the flutter speed of 20.2*m/s* at 6.1*Hz*. predicted by the eigenvalue solution (Figure 6.13) of the state-space model of the system. The *U-g* flutter analysis method predicted flutter to occur at 19.6*m/s* at a frequency of 6.2*Hz*., which was also in acceptable agreement with the actual measured flutter parameters. The conservative *U-g* flutter speed prediction was expected as structural damping in the FAST model mount was not accounted for in the *U-g* flutter analysis. The time response of the FAST model whilst fluttering in its forward cg position configuration is shown in Figure 6.35 and the corresponding normalised plunge and pitch PSDs in Figure 6.36. In this configuration, the modal amplitudes of the FAST model in flutter were ±1.9*mm* in the plunge degree-of-freedom and ±2.2° in the pitch degree-of-freedom. Combined PSD plots of the subcritical signals are given in Figure 6.37 for the plunge SGB and in Figure 6.38 for the pitch SGB. These plots show how much closer the two modal frequencies of the model were in this configuration than in the baseline cg position configuration. At wind tunnel speeds above 10.0*m/s* the PSDs from the pitch SGB signal did not show the two modal frequencies distinctly, probably because they were so close to each other in this configuration. The PSDs from the plunge SGB signals, however, do show a clear distinction between the two modal frequencies and their convergence with increasing airspeed. The predicted and measured modal frequencies versus airspeed are presented in Figure 6.39, which together with Figure 6.35 again demonstrates the stable and benign flutter behaviour of the model and the good agreement between measured and predicted modal frequencies. The absence of a sharp change in either the plunge or pitch subcritical frequencies to the observed flutter frequency as the flutter speed was reached confirms that the flutter was gentle. Although the agreement between the predicted and measured modal frequencies of the model appears poor in Figure 6.39, quantitatively the plunge frequencies of the model were within 3.5% and pitch frequencies within 3.2% of predicted values. The simplified Zimmerman Flutter margins for each tunnel test point are plotted against airspeed in Figure 6.40. Extrapolation of a curve fit through these points to a zero flutter margin was again used to safely approach the flutter speed of the model. In this case, flutter was predicted at 21.9*m/s*, which was 0.5*m/s* (or 2.2%) above the actual flutter speed. The actual flutter speed of the FAST model in the forward cg position configuration was 7.0% (1.6*m/s*) lower than that of the baseline cg position configuration.

Table 6.4 CWT FAST Model Parameters (Forward cg Position Configuration)

Flutter Model Parameter	**Symbol**	**Value**	**Units**
Effective wingspan	s_w	0.335	*m*
Distance between mid-chord and elastic axis in semi-chords	a	-0.2	-
Reference semi-chord	b	0.06	*m*
Control surface position in semi-chords	c	0.5	-
Distance between aerofoil cg and elastic axis in semi-chords	x_α	(0.185)	-
Distance between control surface cg and hinge axis in semi-chords	x_β	-0.075	-
Wing mass	m	1.80	*kg*
Mass inertia of wing (about wing elastic axis)	I_α	(22.5×10^{-3})	kgm^2
Mass inertia of control surface (about hinge axis)	I_β	(45.0×10^{-3})	kgm^2
Natural plunge frequency	f_h	5.90	*Hz.*
Natural pitch frequency	f_α	6.35	*Hz.*
Natural control surface frequency	f_β	5.75	*Hz.*
Plunge damping coefficient	c_h	(0.075)	-
Pitch damping coefficient	c_α	(0.075)	-
Control surface damping coefficient	c_β	(0.05)	-

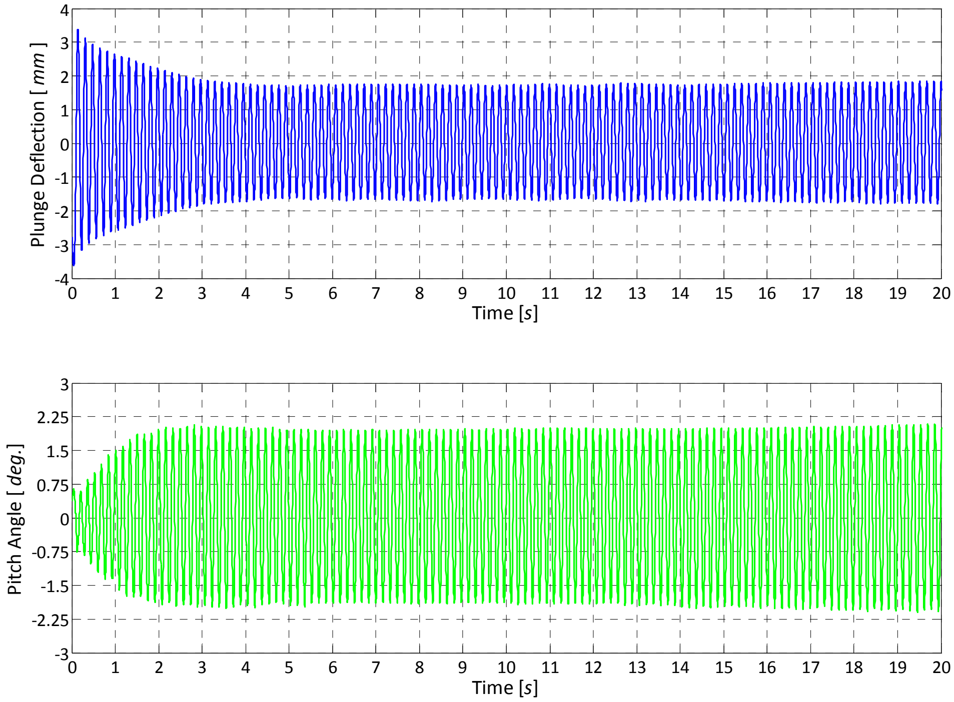


Figure 6.35 FAST Model Open-Loop Flutter (Forward cg Position; U_f = 21.4*m*/*s*)

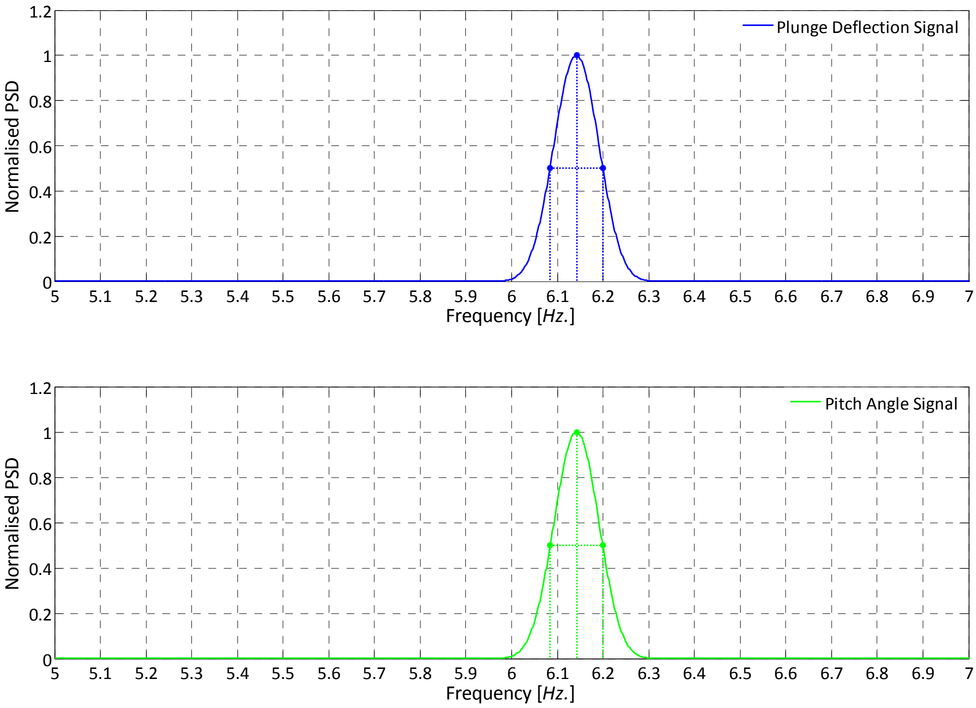


Figure 6.36 Normalised PSD of Open-Loop Flutter (Forward cg Position; U_f = 21.4*m*/*s*)

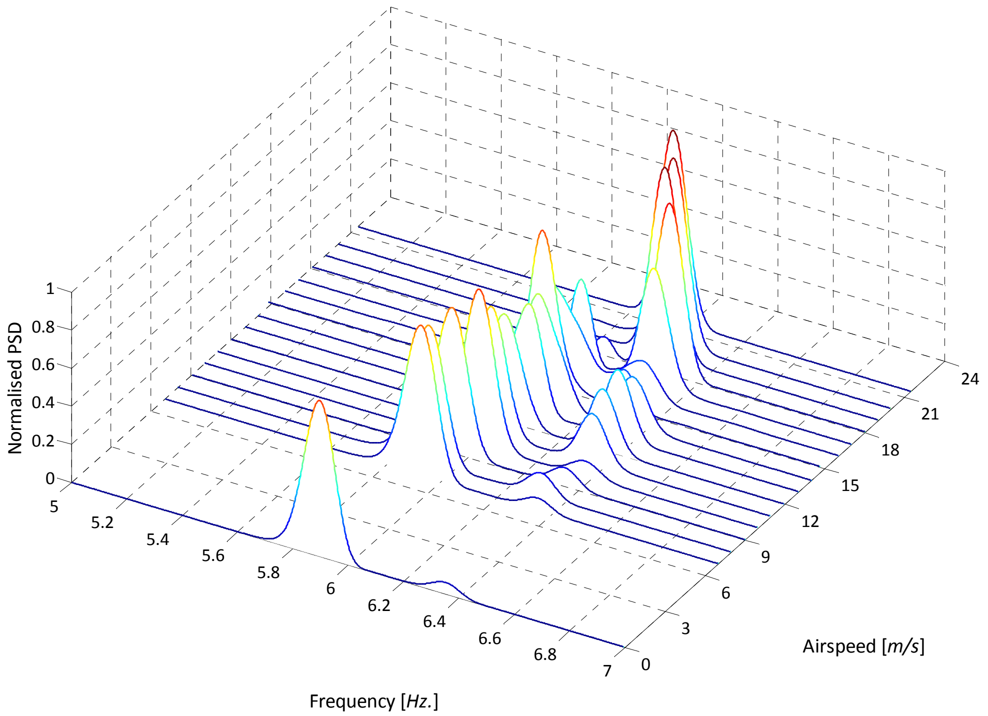


Figure 6.37 Combined Normalised Open-Loop PSDs from Plunge SGB (Forward cg Position)

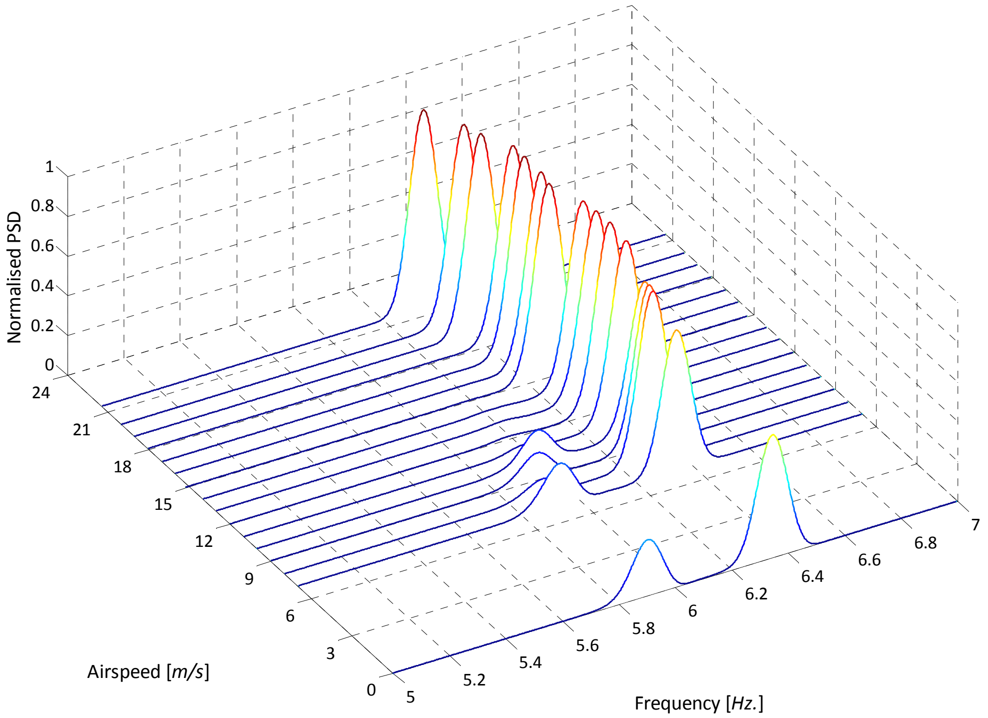


Figure 6.38 Combined Normalised Open-Loop PSDs from Pitch SGB (Forward cg Position)

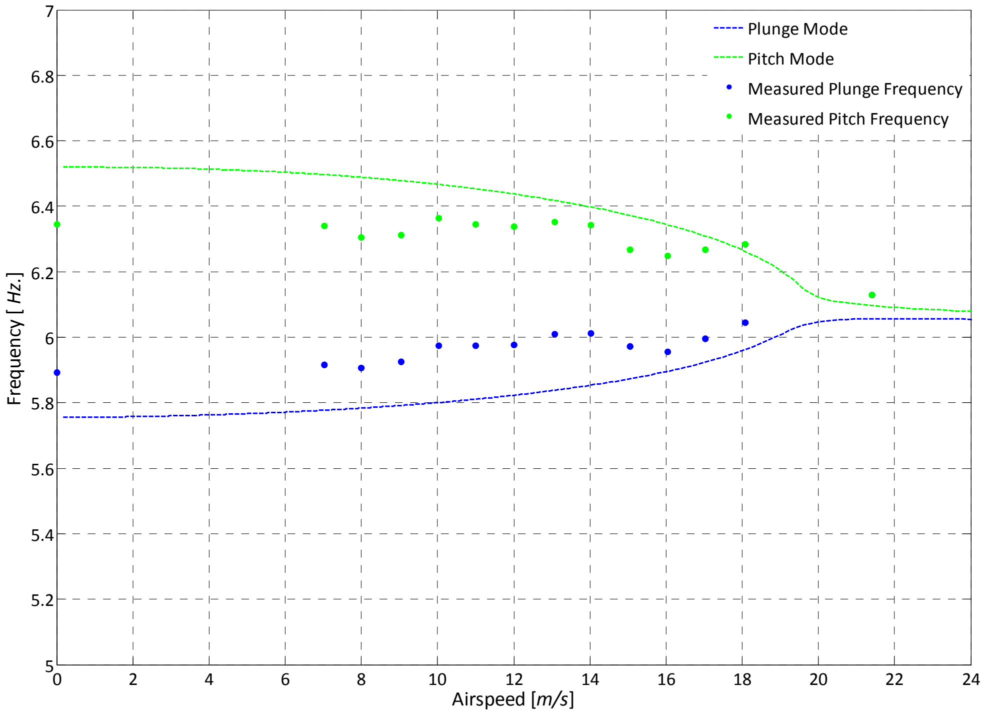


Figure 6.39 Predicted and Measured Open-Loop Modal Frequencies (Forward cg Position)

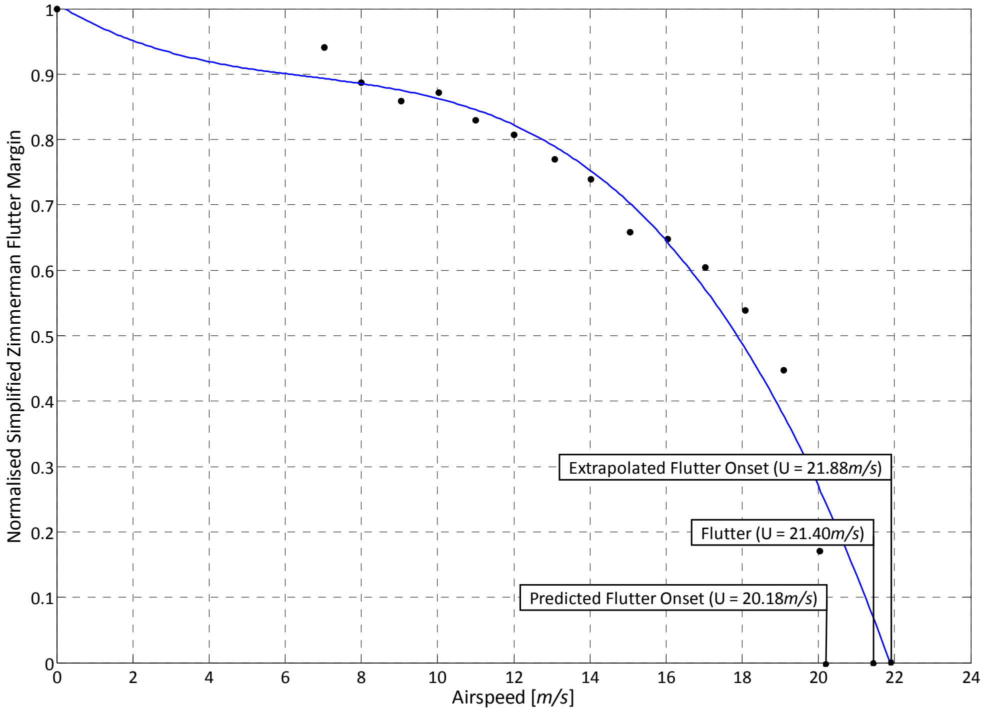


Figure 6.40 Open-Loop Simplified Zimmerman Flutter Margins (Forward cg Position)

Aft Center of Gravity Position Configuration

Table 6.5 lists the model parameters used in the design of the control law, simulations and tests for the FAST model aft cg position configuration. The wind-off time responses and associated PSDs used to establish the natural plunge and pitch frequencies of the model/mount for this configuration can be found in Appendix G.

Table 6.5 CWT FAST Model Parameters (Aft cg Position Configuration)

Parameter	**Symbol**	**Value**	**Units**
Effective wingspan	s_w	0.335	*m*
Distance between mid-chord and elastic axis in semi-chords	a	-0.2	-
Reference semi-chord	b	0.06	*m*
Control surface position in semi-chords	c	0.5	-
Distance between aerofoil cg and elastic axis in semi-chords	x_α	(0.215)	-
Distance between control surface cg and hinge axis in semi-chords	x_β	-0.075	-
Wing mass	m	1.85	*kg*
Mass inertia of wing (about wing elastic axis)	I_α	(22.5×10^{-3})	kgm^2
Mass inertia of control surface (about hinge axis)	I_β	(45.0×10^{-3})	kgm^2
Natural plunge frequency	f_h	5.63	*Hz.*
Natural pitch frequency	f_α	6.72	*Hz.*
Natural control surface frequency	f_β	5.75	*Hz.*
Plunge damping coefficient	c_h	(0.025)	-
Pitch damping coefficient	c_α	(0.075)	-
Control surface damping coefficient	c_β	(0.05)	-

In this configuration, flutter was first excited at an airspeed of 30.0*m/s* (Figure 6.41) and frequency of 6.0*Hz.* (Figure 6.42). This compared well with the calculated values of 29.8*m/s* and 6.1*Hz*. of the *P* flutter analysis method and 27.9*m/s* and 6.1*Hz*. of the *U-g* flutter analysis method. It is likely though that the actual critical flutter speed was slightly below 30.0*m/s*, since both pitch and plunge oscillations of the model were divergent, as seen in Figure 6.41, which is characteristic of flutter at supercritical speeds. In practice it was difficult to achieve the exact critical flutter speed of the model in the aft cg position configuration because of the sharp decrease in aerodynamic damping as the flutter speed was reached. This much more sudden onset of flutter is apparent from the calculated modal damping (combined structural and aerodynamic) of the model shown in Figure 6.14 and the steep change in the measured pitch modal frequency near to the flutter frequency, as is evident in Figure 6.45. The combined PSD plot from the plunge SGB is shown in Figure 6.43 while that from the pitch SGB is given in Figure 6.44. These plots clearly illustrate the slow convergence of the modal frequencies followed by the sudden onset of flutter. Figure 6.45 shows good agreement between the predicted and measured pitch and plunge frequencies of the FAST model as a function of airspeed. The largest difference between the measured and calculated plunge frequencies was 1.8% and between the measured and calculated pitch frequencies was 2.0%. Simplified Zimmerman Flutter margins from these tests are plotted in Figure 6.46. As for the other two cg position configurations tested, the plot was extrapolated forward to zero to predict the onset of flutter and so approach it safely. The subcritical wind tunnel data provided a good on-line estimate of the flutter speed, within 2.0% of the actual value as shown in Figure 6.46. The observed flutter speed of the FAST model in this configuration was 0.8% higher than the calculated value using the *P* flutter analysis method and 7.1% higher than the prediction of the *U-g* flutter analysis method. Again, the conservative flutter speed prediction using the *U-g* flutter analysis method was consistent with the fact that structural damping of the mount was omitted from these calculations. The flutter speed of the FAST model in this configuration was 30.4% higher than the baseline cg position configuration flutter speed, as expected for the larger x_α value and greater separation of the two wind-off modal frequencies.

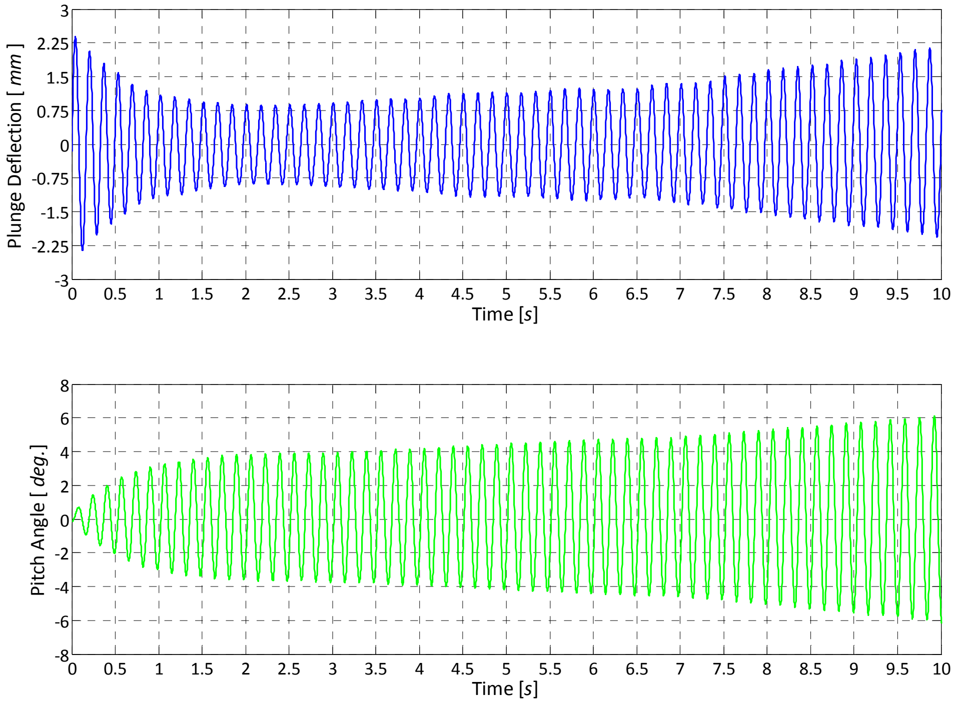


Figure 6.41 FAST Model Open-Loop Flutter (Aft cg Position; U_f = 30.0m/s)

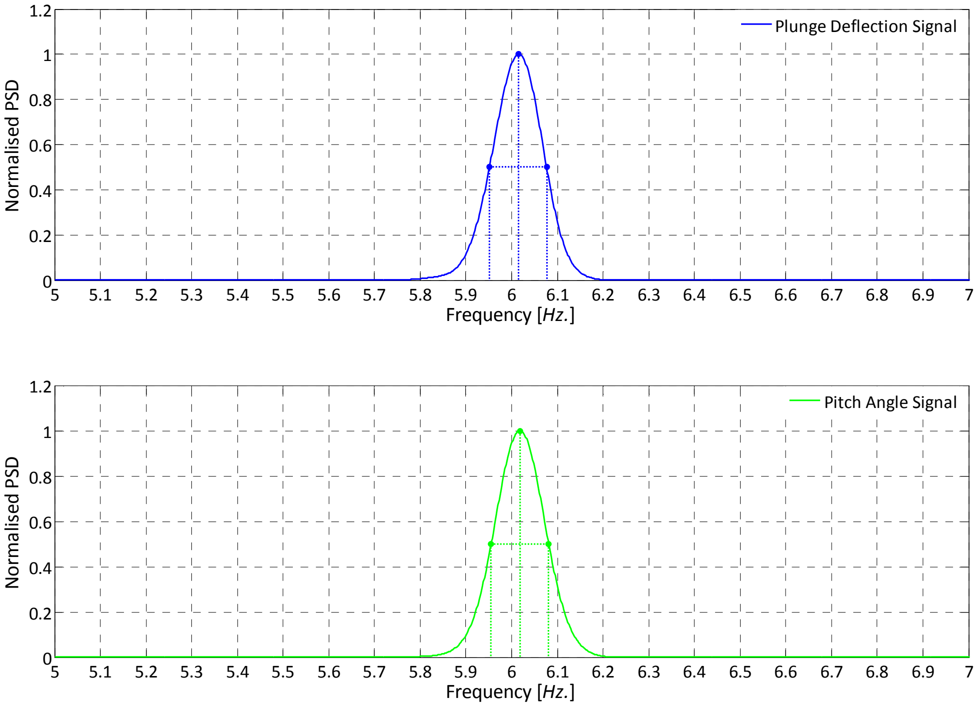


Figure 6.42 Normalised PSD of Open-Loop Flutter (Aft cg Position; U_f = 30.0m/s)

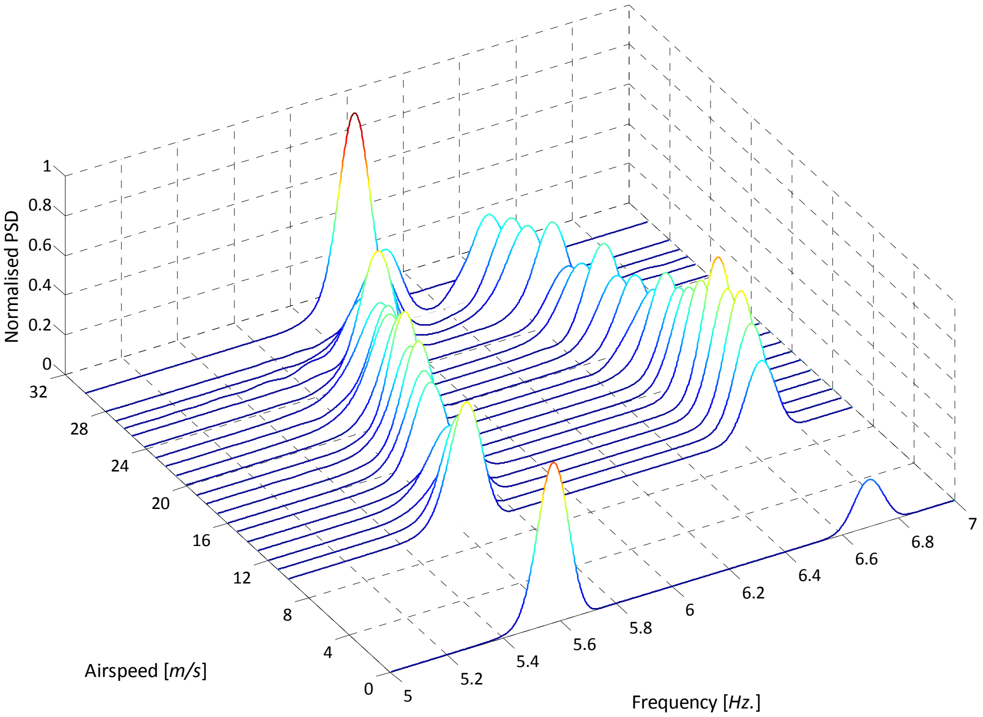


Figure 6.43 Combined Normalised Open-Loop PSDs from Plunge SGB (Aft cg Position)

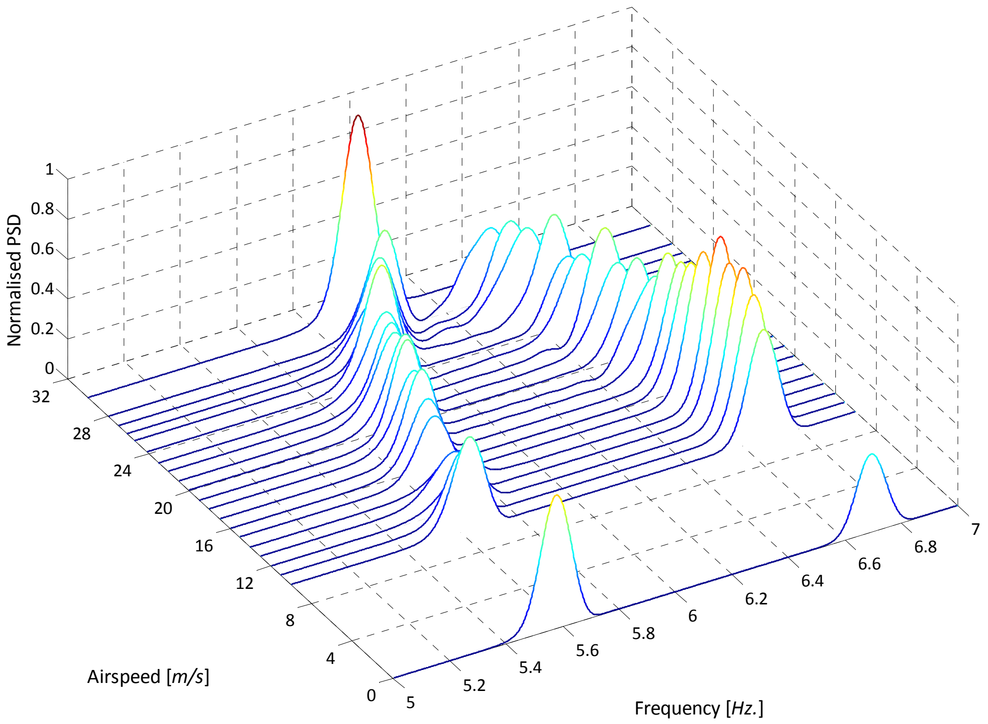


Figure 6.44 Combined Normalised Open-Loop PSDs from Pitch SGB (Aft cg Position)

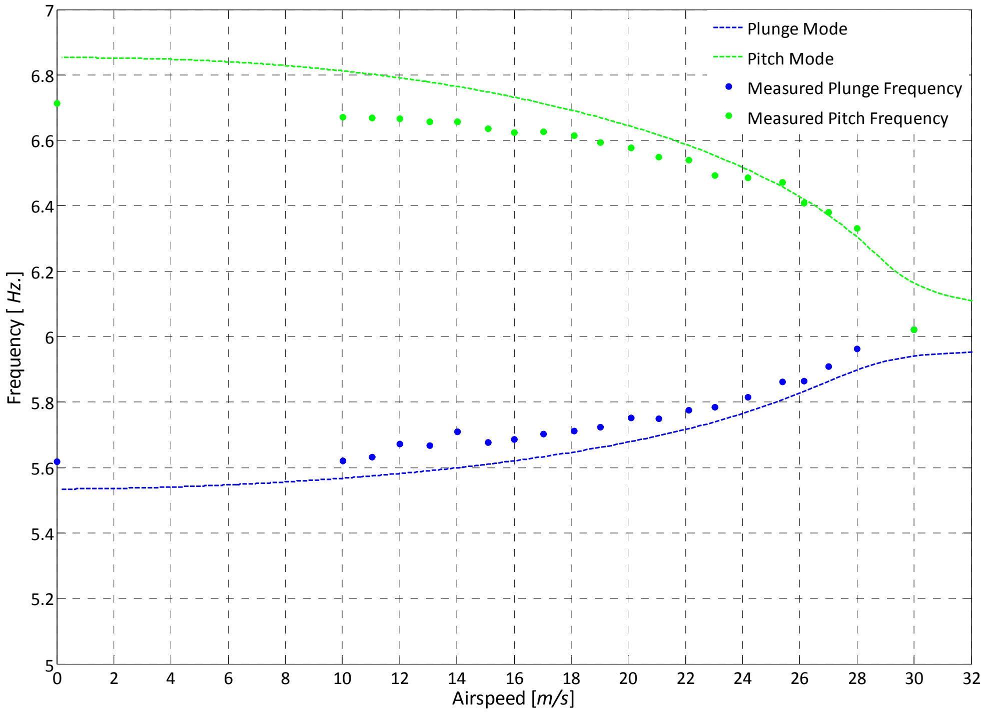


Figure 6.45 Predicted and Measured Open-Loop Modal Frequencies (Aft cg Position)

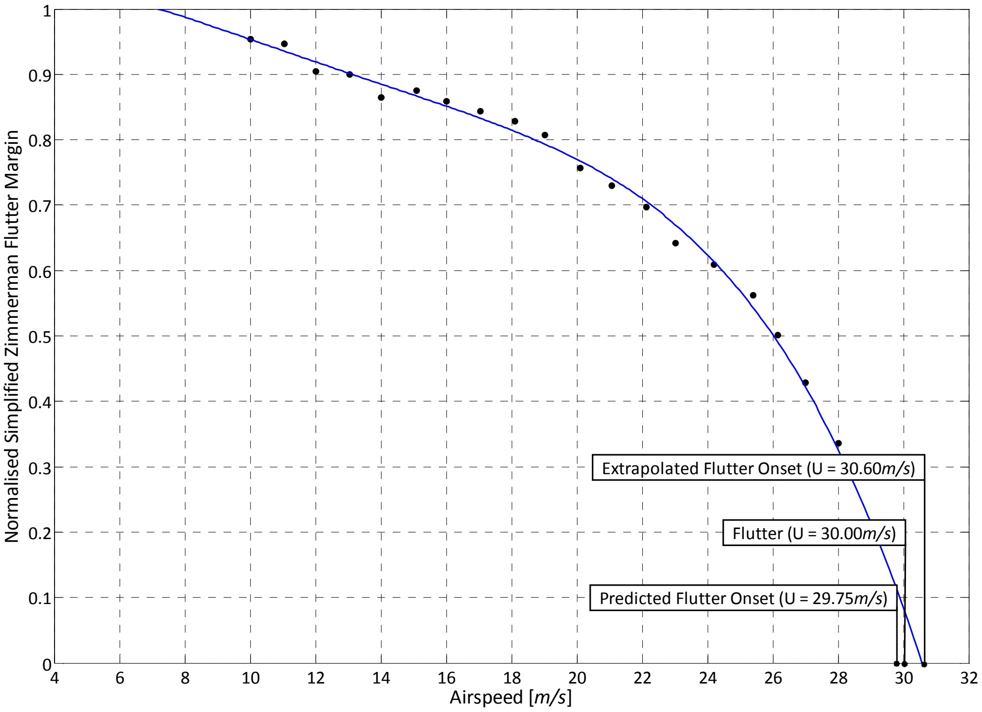


Figure 6.46 Open-Loop Simplified Zimmerman Flutter Margins (Aft cg Position)

Summary of Open-Loop Calibration Wind Tunnel Results

A summary of the predicted flutter speeds and frequencies and the actual results from the open-loop CWT flutter testing of the FAST model in each cg position configuration is presented in Table 6.6. For each cg position tested the flutter speed and frequency calculated with the *P* flutter analysis method (Section 3.1.6), the *U-g* flutter analysis method (Appendix A) and the actual measured values are listed. In each case, it is apparent that the observed flutter speeds and frequencies of the FAST model closely matched the theoretical predictions.

Table 6.6 FAST Model Open-Loop Flutter Parameter Predictions and CWT Test Results

FAST Model Configuration (cg Position)	**Flutter Speed**			**Flutter Frequency**		
	m/s			***Hz.***		
	***P* Analysis**	***U-g* Analysis**	**Actual Value**	***P* Analysis**	***U-g* Analysis**	**Actual Value**
Baseline	23.51	21.70	23.01	5.98	5.98	5.83
Forward	20.18	19.58	21.40	6.06	6.15	6.13
Aft	29.75	27.87	30.00	6.12	6.32	6.02

The *U-g* flutter analysis method assumes zero structural damping while the *P* method includes structural damping, consistent with the conservative flutter predictions of the *U-g* method when compared to the *P* method.

After gentle open-loop flutter of the FAST model was safely demonstrated and before closed-loop testing commenced, the mathematical model of the system was verified. Although the original mathematical model predicted the flutter speeds and frequencies accurately, as shown in the above results, the predicted modeshapes did not match the measured modeshapes entirely. This was expected as modal damping used in the flutter prediction was difficult to measure and the cg position and mass inertia were not measured exactly, but rather estimated from the CAD drawing of the FAST model and mount. To better match the actual and the predicted dynamics, and to improve the effectiveness of the closed-loop control law design, certain parameters that were originally estimated were adjusted within a reasonable margin (±10%), based on the open-loop wind tunnel data. These parameters were the modal dampings, wing mass inertia, wing cg position and the control surface mass inertia. The estimates of the unmeasured parameters were improved semi-empirically by matching the measured modeshapes and their amplitude ratios (ratio of the plunge modeshape peak-to-peak amplitude to the pitch modeshape peak-to-peak amplitude) to those predicted by the mathematical model. The final estimated model parameters did not change significantly from the CAD predictions, but the new calculated modeshapes for each cg position configuration matched the measured modeshapes much better. The values of model parameters listed in Table 6.3, Table 6.4 and Table 6.5 were used in the controller designs for the baseline, forward and aft cg position configurations respectively.

6.3.3 Closed-Loop Calibration Wind Tunnel Tests

Closed-loop wind tunnel testing of the FAST model in the CWT was conducted once the state-space model of the aeroservoelastic system was validated from the open-loop wind tunnel results. The purpose of these tests was to demonstrate the principle of active flutter suppression by showing the effectiveness of an intelligently activated control surface in preventing flutter at, and above, the open-loop flutter speed of the model. The same three cg position configurations evaluated and tested in open-loop wind tunnel testing (Section 6.3.2) were used in closed-loop wind tunnel testing. Extensive closed-loop tests were performed with the FAST model in its baseline cg position configuration. Limited testing was done with the model in the other two cg position configurations.

Closed-loop testing of the FAST model in the CWT was conducted conservatively by activating the control system and then evaluating the subcritical response by giving the model an initial displacement and allowing it to oscillate freely. These tests started at a wind tunnel speed below the previously observed open-loop flutter speed of the model in the cg position configuration under investigation, and continued until the critical flutter speed was reached. Once it was established that flutter could not be excited at the open loop flutter speed with the control system active, the control was switched off and the model was allowed to enter a sustained flutter cycle. The control system was then re-engaged and the effectiveness of the control surface in actively damping out the unsteady oscillations was observed. The closed-loop results presented in this section show the time response of the model at its critical flutter speed (for each model cg configuration tested), before and after control was applied. The eight system states of the mathematical model as estimated by the Kalman filter and used to calculate the commanded control surface angle are also given.

Flutter boundary extension tests were only conducted with the FAST model in its baseline cg position configuration. In these tests the model did not flutter and was kept stable by the control system up to the maximum operating speed of the CWT. The state-space matrices and control system gains used during this testing are also given in this section, so that the closed-loop testing can be replicated.

Baseline Center of Gravity Position Configuration

Initial closed-loop testing was conducted on the FAST model in its baseline cg position configuration. A controller designed for an airspeed of 25.0*m/s* (slightly *above* the demonstrated open-loop flutter speed) was implemented on the model in this configuration. The discrete system, input and measurement matrices ($\mathbf{A}_d$, $\mathbf{B}_d$, $\mathbf{C}_d$) calculated in MATLAB™ from measured and estimated FAST model parameters, and the calculated LQR feedback ($\mathbf{K}_{LQR}$) and Kalman filter ($\mathbf{K}_{Kalman}$) gains are given in Table 6.7.

Table 6.7 FAST Model LQG Controller Parameters (Baseline cg Position; U_D = 25.0*m/s*)

$\mathbf{A}_d$	0.9985	-0.0001	-0.0002	-0.8356	-0.0171	-0.0213	0.0002	0.0004
	0.0023	0.9993	-0.0003	0.6150	-0.9625	0.0046	-0.0003	-0.0007
	0.0000	0.0000	0.9587	0.0000	0.0000	-0.8550	0.0000	0.0000
	0.0007	0.0000	0.0000	0.9997	0.0000	0.0000	0.0000	0.0000
	0.0000	0.0007	0.0000	0.0002	0.9997	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0007	0.0000	0.0000	0.9997	0.0000	0.0000
	-0.0013	0.0165	0.0092	-0.8000	-0.0652	-0.0359	0.9888	0.0003
	-0.0013	0.0159	0.0088	-0.7697	-0.0628	-0.0346	0.0002	0.9150
$\mathbf{B}_d$	0.0038	0.0058	0.8550	0.0000	0.0000	0.0003	0.0041	0.0040
$\mathbf{K}_{LQR}$	-0.62	0.32	0.03	38.77	4.07	2.26	-0.01	-0.01
$[\mathbf{L}_{Kalman}]^T$	46.03	3.26	0.00	0.98	0.00	0.00	-0.15	-0.07
	0.08	-0.12	0.00	0.00	0.66	0.00	0.22	0.01
	-0.01	0.00	-0.33	0.00	0.00	0.60	-0.02	-0.01

A developed flutter cycle was successfully suppressed by the LQG flutter suppression controller, as is evident in Figure 6.47 which shows the effectiveness of the control system in suppressing flutter at a wind tunnel speed of 24.5*m*/*s* after control was initiated at t = 4.0*s*. The divergence in the plunge displacement and pitch angle before feedback control was applied is consistent with the fact that the wind tunnel speed of 24.5*m*/*s* was higher than the demonstrated open-loop flutter speed of 23.0*m*/*s.* The LQG flutter suppression controller was able to damp out both structural modes and the model was fully stabilised within 5.0*s* with $h < \pm 0.5mm$ and $\alpha < \pm 0.25°$. It is apparent from the closed-loop response of the model that the pitching moment of the model was directly regulated and all other states were stabilised because of the close-coupling between system states. The plunge response of the model shown in Figure 6.47 demonstrates this. Initially the plunge displacement peak-to-peak amplitudes decreased, then after a short period increased before decreasing again. This is explained by the downward (positive) control surface deflection increasing the nose-down pitching moment whilst simultaneously creating additional lift, thus causing an initial increase in the plunge peak-to-peak amplitudes. In comparison, the control system continually decreased the pitch angle peak-to-peak amplitude, as shown by the pitch response in Figure 6.47. Figure 6.47 and Figure 6.49 illustrate that the control surface actuator became saturated at about t = 4.1*s*, although only for a short time (≈ 0.02*s*). In spite of this the controller was robust enough to still suppress flutter effectively. The reason for the actuator saturation was that control surface angle limitations were implemented in the controller software to avoid excessive control surface deflections during wind tunnel testing. In this case, the commanded control surface angle was larger than the imposed limit of 20.0°.

The Kalman state estimates as calculated by the Kalman filter and used by the LQR regulator to determine the required control surface angle are shown in Figure 6.48. The response was fast and the Kalman estimates matched measured states almost exactly. Figure 6.49 gives a 1.0*s* window of the plunge and pitch time responses and control surface deflection of the model at an airspeed of 24.5 *m*/*s*, and shows the degree to which system states estimated by the Kalman filter matched the actual (measured) system states. The phase shift between the commanded and measured control surface angle seen in Figure 6.49 was expected and is accounted for by the transfer function of the control surface and actuation system discussed in Section 6.1.5. Figure 6.50 shows an overlay of the open- and closed-loop (Figure 6.48) time responses of the model at an airspeed of 24.0*m*/*s*, and clearly illustrates the effectiveness of the system in controlling the flutter. Finally, Figure 6.51 and Figure 6.52 show the combined PSDs of the subcritical time series data. They clearly illustrate the effectiveness of the control in separating the two modal frequencies (given by the PSD peaks) when compared with corresponding open-loop data plotted in Figure 6.31 and Figure 6.32. The damping added to the system by the LQG controller is evident from the greater width of the closed-loop PSD peaks near the open-loop flutter speed in comparison with the corresponding narrower open-loop PSD peaks in Figure 6.31 and Figure 6.32.

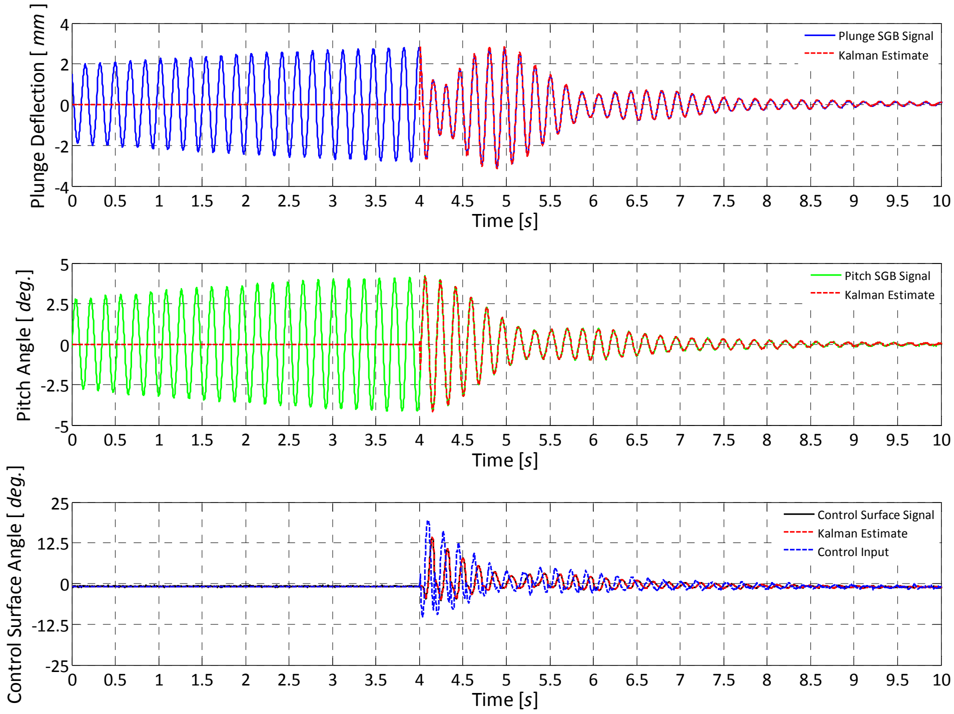


Figure 6.47 FAST Model Closed-Loop Time Response (Baseline cg Position; *U* = 24.52*m*/*s*)

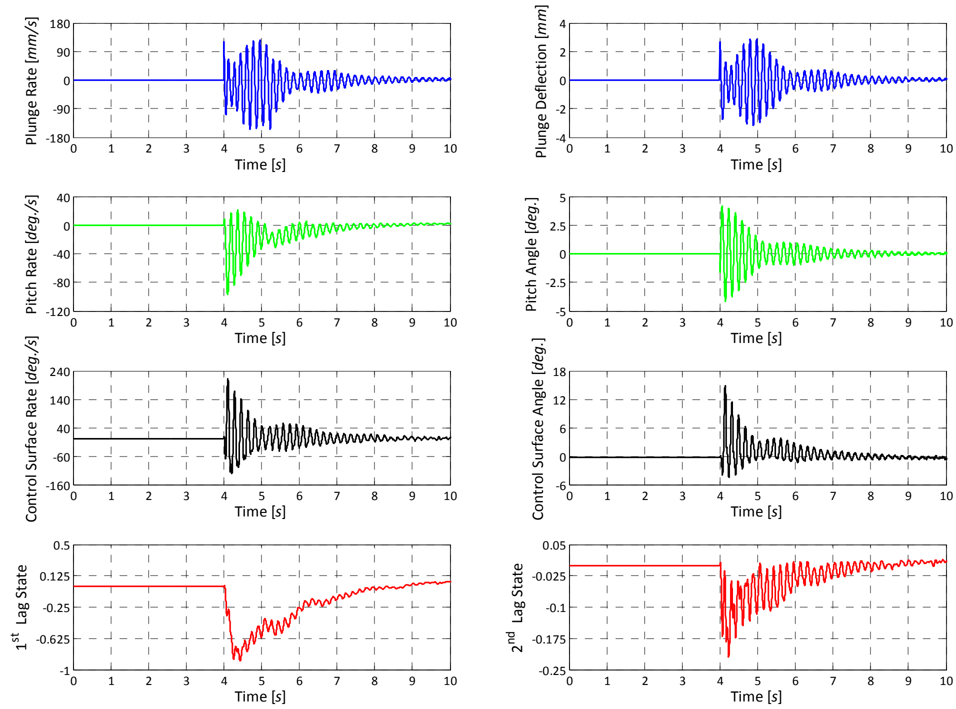


Figure 6.48 FAST Model Kalman Estimates (Baseline cg Position; *U* = 24.5*m*/*s*)

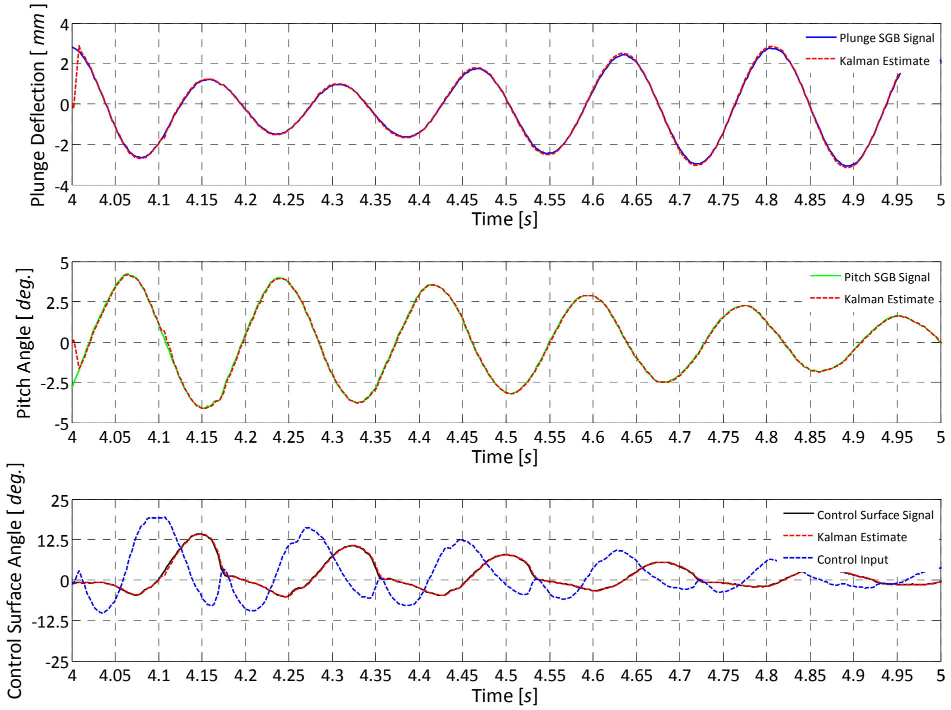


Figure 6.49 Segment of Closed-Loop Time Response (Baseline cg Position; *U* = 24.5*m*/*s*)

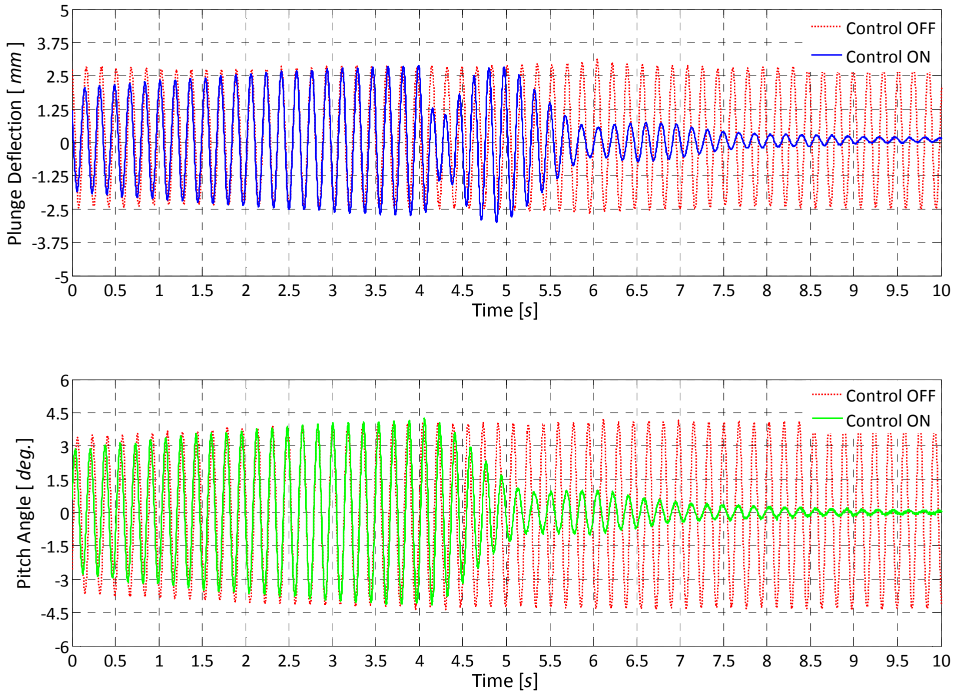


Figure 6.50 Overlaid Open- and Closed-Loop Responses (Baseline cg Position; $U \approx 24.0$ *m*/*s*)

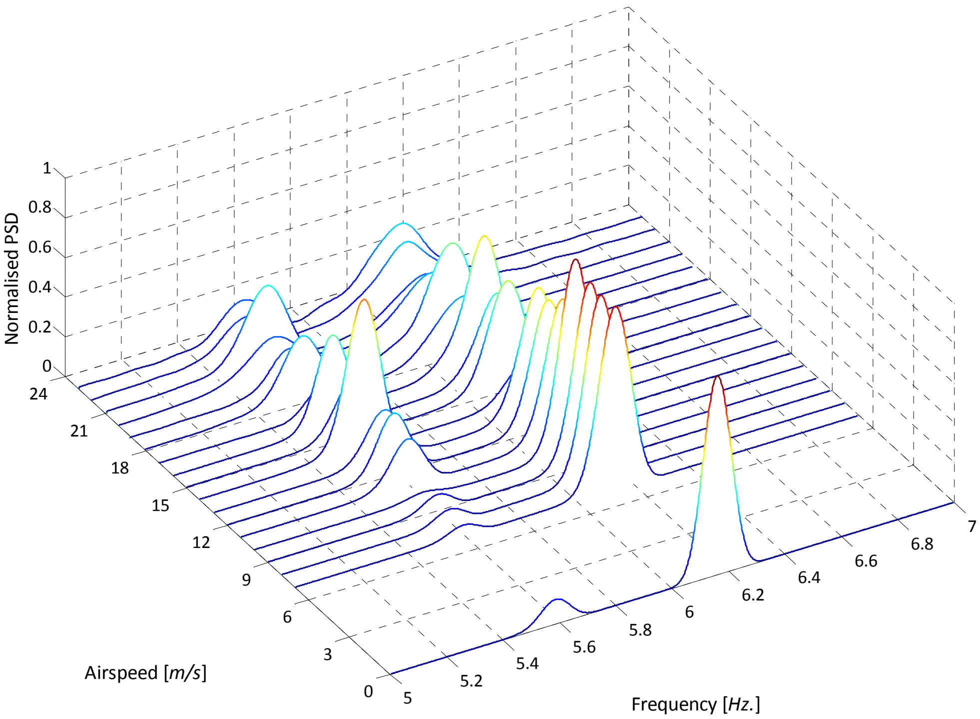


Figure 6.51 Combined Normalised Closed-Loop PSDs from Plunge SGB (Baseline cg Position)

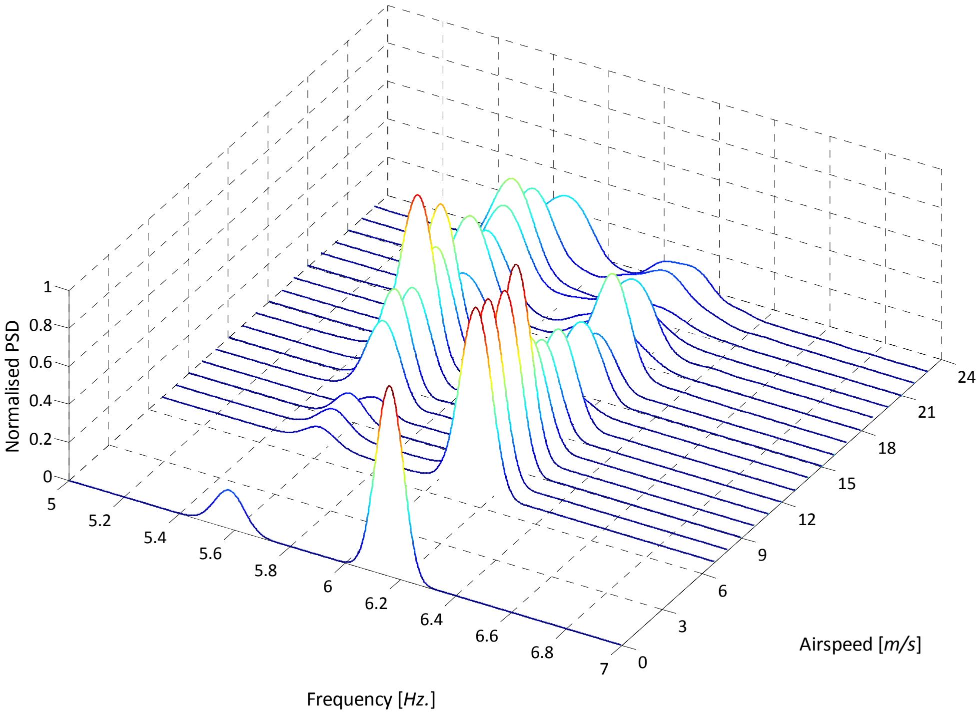


Figure 6.52 Combined Normalised Closed-Loop PSDs from Pitch SGB (Baseline cg Position)

Forward Center of Gravity Position Configuration

For the forward cg position configuration the controller was modified to comply with the different dynamics of the model. The discreet system matrix, input matrix, LQR gain matrix and Kalman filter gain matrix calculated for the forward cg configuration are given in Table 6.8. Closed-loop testing followed the same sequence as that for the baseline cg position configuration. The LQG controller was effective in suppressing a fully developed flutter cycle at a wind tunnel speed of 21.5*m/s*, the critical flutter speed of the model in this configuration. Figure 6.53 shows a typical closed-loop time response of the FAST model before and after activation of the control system at *t* = 7.0*s*. The corresponding Kalman state estimates for this configuration are shown in Figure 6.54. Again, Kalman estimates matched measured states almost exactly as is evident in Figure 6.53. In this configuration the flutter was fully suppressed in approximately 10*s*, 5*s* longer than the time required to suppress flutter in the baseline cg position configuration. The plunge displacement and pitch angle of the model where reduced to within ±0.5*mm* and ±0.25° respectively in approximately 5*s*. The reduced performance of the forward cg position configuration controller, when compared with the baseline cg position controller, was expected as the feedback gains calculated for this configuration were not optimised, as was the case with the baseline cg position controller gains. At the lower wind tunnel speed, aerodynamic loads generated by the control surface were not as high as in the baseline cg case, which also contributed to the slower response of the control system in the forward cg case.

Table 6.8 FAST Model LQG Controller Parameters (Forward cg Position; U_D = 21.0*m/s*)

$\mathbf{A}_d$	0.9986	-0.0001	-0.0002	-0.9239	-0.0115	-0.0177	0.0002	0.0003
	0.0027	0.9989	-0.0004	0.8215	-1.0189	0.0019	-0.0004	-0.0008
	0.0000	0.0000	0.9587	0.0000	0.0000	-0.8550	0.0000	0.0000
	0.0007	0.0000	0.0000	0.9997	0.0000	0.0000	0.0000	0.0000
	0.0000	0.0007	0.0000	0.0003	0.9997	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0007	0.0000	0.0000	0.9997	0.0000	0.0000
	-0.0012	0.0145	0.0080	-0.8790	-0.0615	-0.0319	0.9901	0.0003
	-0.0012	0.0140	0.0077	-0.8495	-0.0595	-0.0309	0.0001	0.9248
$\mathbf{B}_d$	0.0037	0.0079	0.8550	0.0000	0.0000	0.0003	0.0036	0.0035
$\mathbf{K}_{LQR}$	-0.72	0.26	0.03	46.76	4.69	2.15	-0.01	-0.01
$[\mathbf{L}_{Kalman}]^T$	46.13	3.75	0.00	0.98	0.00	0.00	-0.06	-0.03
	0.09	-0.16	0.00	0.00	0.66	0.00	0.18	0.01
	-0.01	0.00	-0.33	0.00	0.00	0.60	-0.01	-0.01

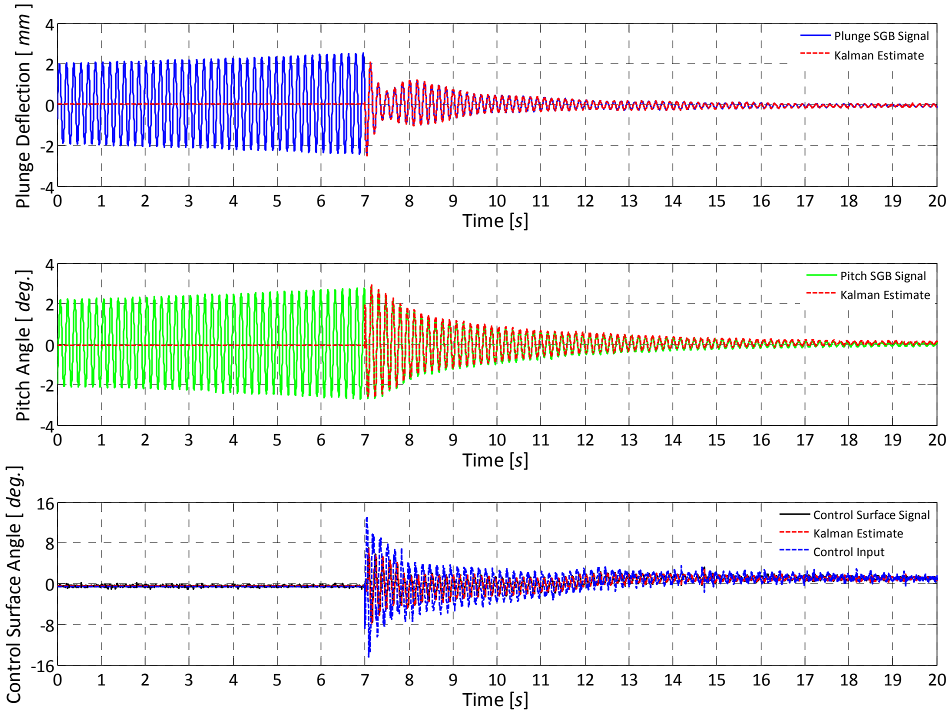


Figure 6.53 FAST Model Closed-Loop Time Response (Forward cg Position; *U* = 21.5*m*/*s*)

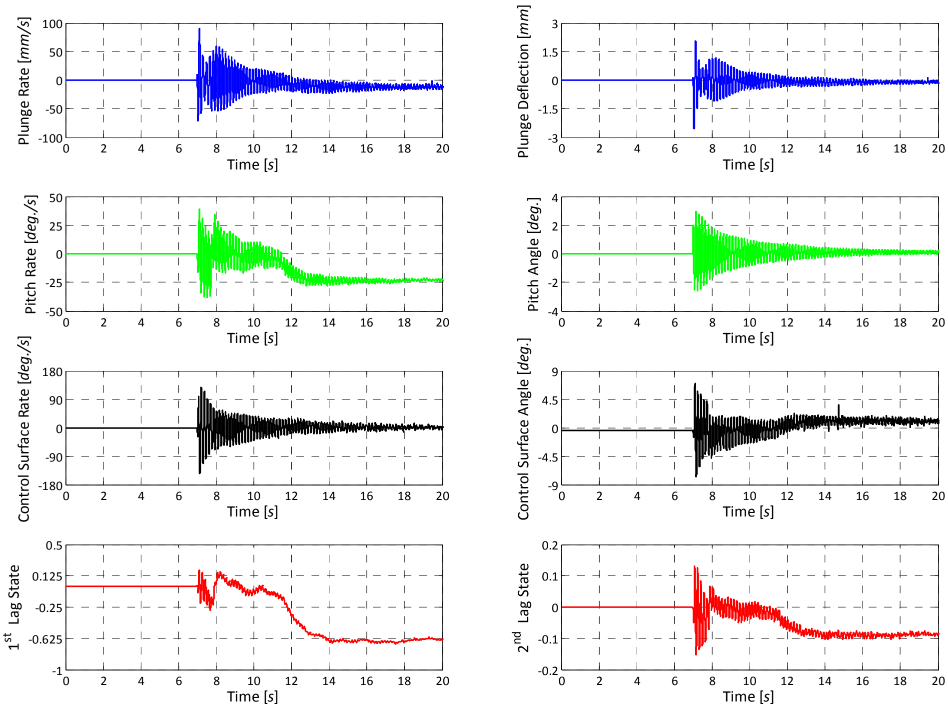


Figure 6.54 FAST Model Kalman Estimates (Forward cg Position; *U* = 21.5*m*/*s*)

Aft Center of Gravity Position Configuration

The discrete system, input, LQR gain and Kalman filter gain matrices calculated for the FAST model in an aft cg position configuration using a design airspeed of 30.0*m/s* are listed in Table 6.9. The closed-loop response of the FAST model in its aft cg position configuration is presented in Figure 6.55. This figure shows, for an airspeed of 30.4*m/s*, the closed-loop time response of the FAST model before and after activation of the feedback control at t = 7.0*s*. Figure 6.56 gives the corresponding Kalman state estimates. It is evident from the divergence of the response before control was applied that that the wind tunnel speed was above the critical open-loop speed of the model, but the controller was robust enough to suppress this unsteady oscillation easily. In this configuration modal oscillations were reduced to within ±0.5*mm* in the plunge mode and ±0.25° in the pitch mode in under 5.0*s* and flutter was fully suppressed in under 6.0*s*, 1.0*s* slower than for the baseline cg position and 4.0*s* faster than for the forward cg position configuration. The higher wind tunnel speed meant that the control surface had more authority than in the baseline and forward cg position configurations and was therefore more responsive in suppressing flutter. The aft cg position configuration closed-loop flutter tests demonstrated the robustness of the flutter suppression controller and its ability to suppress the more violent and sudden flutter (Figure 6.45) of this case.

Table 6.9 FAST Model LQG Controller Parameters (Aft cg Position; U_D = 30.0*m/s*)

$\mathbf{A}_d$	0.9983	-0.0001	-0.0002	-0.8421	-0.0285	-0.0298	0.0002	0.0005
	0.0038	0.9992	-0.0004	0.8496	-1.0995	0.0140	-0.0006	-0.0011
	0.0000	0.0000	0.9587	0.0000	0.0000	-0.8550	0.0000	0.0000
	0.0007	0.0000	0.0000	0.9997	0.0000	0.0000	0.0000	0.0000
	0.0000	0.0007	0.0000	0.0003	0.9996	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0007	0.0000	0.0000	0.9997	0.0000	0.0000
	-0.0015	0.0198	0.0111	-0.7924	-0.0852	-0.0447	0.9866	0.0004
	-0.0015	0.0189	0.0106	-0.7565	-0.0815	-0.0427	0.0002	0.8989
$\mathbf{B}_d$	0.0037	0.0074	0.8550	0.0000	0.0000	0.0003	0.0050	0.0048
$\mathbf{K}_{LQR}$	-0.52	0.32	0.03	39.30	1.59	2.66	-0.01	-0.01
$[\mathbf{L}_{Kalman}]^T$	45.63	5.19	0.01	0.98	0.00	0.00	0.71	0.22
	0.13	-0.10	0.00	0.00	0.66	0.00	0.28	0.02
	-0.01	0.01	-0.33	0.00	0.00	0.60	-0.02	-0.02

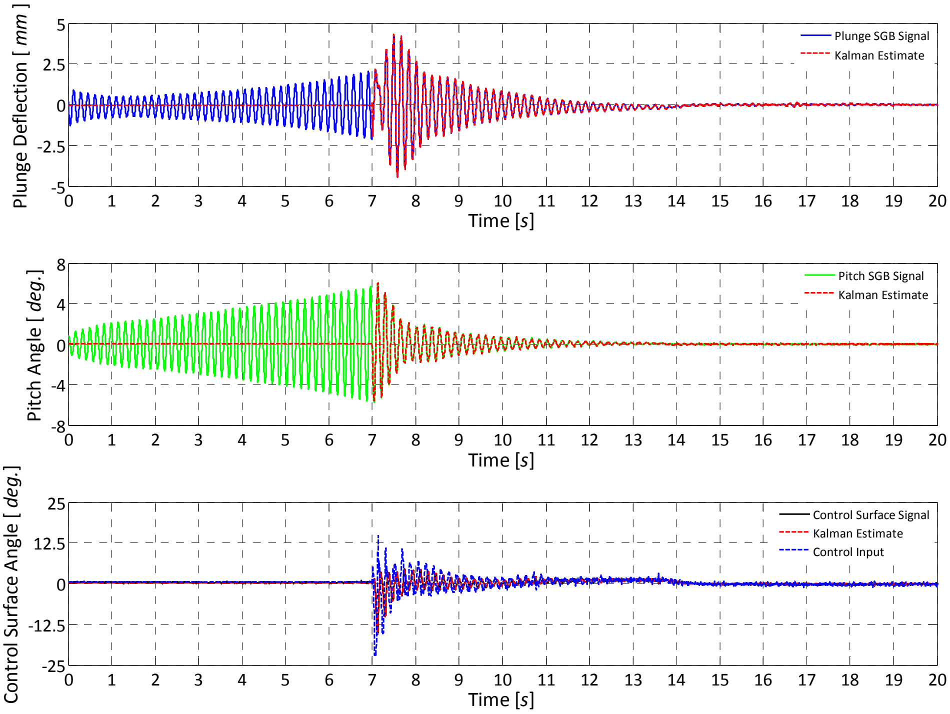


Figure 6.55 FAST Model Closed-Loop Time Response (Aft cg Position; *U* = 30.4*m*/*s*)

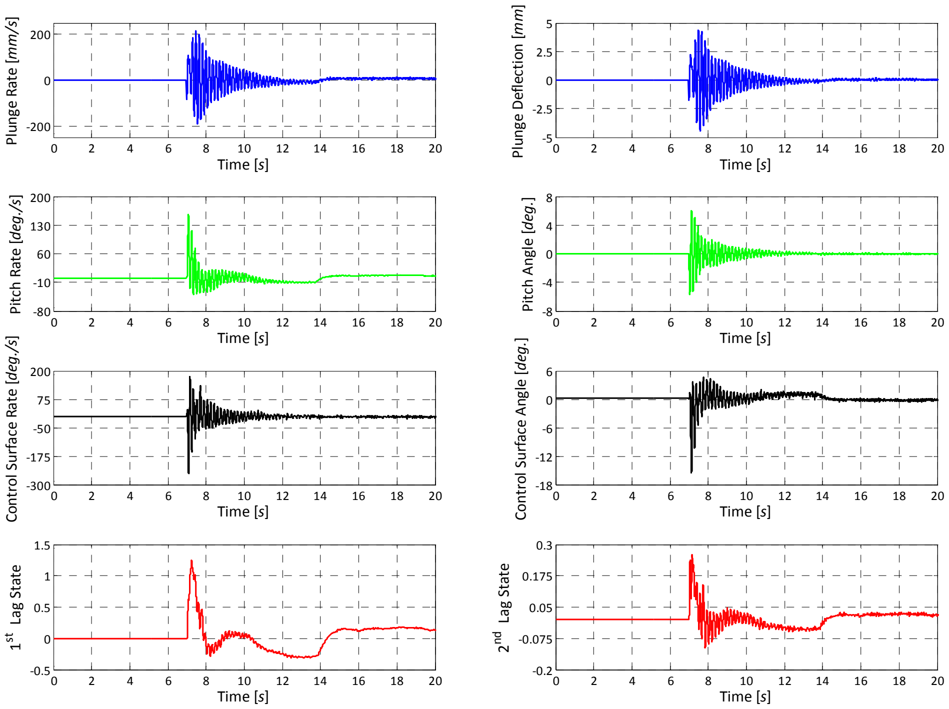


Figure 6.56 FAST Model Kalman Estimates (Aft cg Position; *U* = 30.4*m*/*s*)

Flutter Boundary Extension (Baseline Center of Gravity Position Configuration)

Based on the success of closed-loop testing of the FAST model at the critical flutter speed for each cg position configuration, flutter boundary extension tests were performed in the CWT with the model in its baseline cg position configuration. The purpose of these tests was to demonstrate the increase in speed that could be achieved by active flutter suppression and to determine what margin the LQG controller was capable of achieving. The flutter boundary of the FAST model was extended by conducting closed-loop tests similar to those performed at the critical open-loop flutter speed, at speeds up to the maximum operating speed of the CWT. With the control system active the wind tunnel speed was increased incrementally from the open-loop flutter speed of the model and the response of the model observed. Flutter could not be induced for all tests right up to the maximum operating speed of the CWT (nominally 35*m/s*), either through wind tunnel turbulence or by applying an initial displacement or impulse to the model and allowing it to oscillate freely. In view of this, it was then decided to allow the model to enter a sustained flutter cycle with the control system inoperative at the maximum operating speed of the CWT before re-engaging the control system and observing it's effectiveness in suppressing flutter.

Figure 6.57 presents the closed-loop time responses of the FAST model at the maximum speed of 35.6*m/s* of the CWT, using a control law designed for 25.0*m/s*, before and after activating the control at t = 3.0*s*. The effectiveness of the controller in fully suppressing this supercritical flutter within 4.0*s* is clear. The corresponding Kalman state estimates and how they too were stabilised by the feedback controller is shown in Figure 6.58. The excellent agreement between the Kalman state estimates and the actual measured system states is evident in Figure 6.57. The stable closed-loop response of the model at this speed represents an increase of 54.8% (or 1.5 times) in the open-loop flutter speed of the model. More importantly, this translates to an increase of 139.6% (or 2.40 times) in dynamic pressure which is the main driver of flutter. This result was impressive as it was achieved with a controller designed for a wind tunnel speed of only 25.0*m/s* (Table 6.7), and demonstrated the robustness and effectiveness of the controller well beyond its design point. With all other controller parameters remaining the same, a new control law was then designed for a wind tunnel speed of 35.0*m/s* to investigate the effect of designing for the correct speed. The calculated discrete system, input, LQR gain and Kalman filter gain matrices for this LQG controller are listed in Table 6.10. The closed-loop response of the FAST model using this "optimal" LQG controller is shown in Figure 6.59 and the Kalman estimates of system states in Figure 6.60. Under the same wind tunnel and ambient conditions, the new controller was able to damp out divergent oscillations within 2.5*s*, which was 1.5*s* (or 37.5%) faster than the controller designed for a wind tunnel speed of 25.0*m/s*. This result indicates the appeal of designing a gain scheduled LQG controller, discussed further in Chapter 7.

Table 6.10 FAST Model LQG Controller Parameters (Optimised FBE Case; $U_D = 35.0m/s$)

A$_d$	0.0998	-0.0001	-0.0002	-.08355	-0.0452	-0.0382	0.0003	0.0005
	0.0032	0.9993	-0.0003	0.6189	-0.9108	0.0146	-0.0005	-0.0009
	0.0000	0.0000	0.9587	0.0000	0.0000	-0.8550	0.0000	0.0000
	0.0007	0.0000	0.0000	0.9997	0.0000	0.0000	0.0000	0.0000
	0.0000	0.0007	0.0000	0.0002	0.9997	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0007	0.0000	0.0000	0.9997	0.0000	0.0000
	-0.0017	0.0231	0.0131	-0.7959	-0.0934	-0.0537	0.9844	0.0005
	-0.0017	0.0219	0.0124	-0.7541	-0.0887	-0.0510	0.0002	0.8831
B$_d$	0.0037	0.0059	0.8550	0.0000	0.0000	0.0003	0.0058	0.0056
K$_{LQR}$	-0.56	0.35	0.03	32.48	8.81	2.84	-0.01	-0.01
$[\mathbf{L}_{Kalman}]^T$	45.39	4.41	0.01	0.98	0.00	0.00	0.48	0.13
	0.01	-0.08	0.00	0.00	0.66	0.00	0.24	0.01
	-0.01	0.01	-0.33	0.00	0.00	0.60	-0.02	-0.02

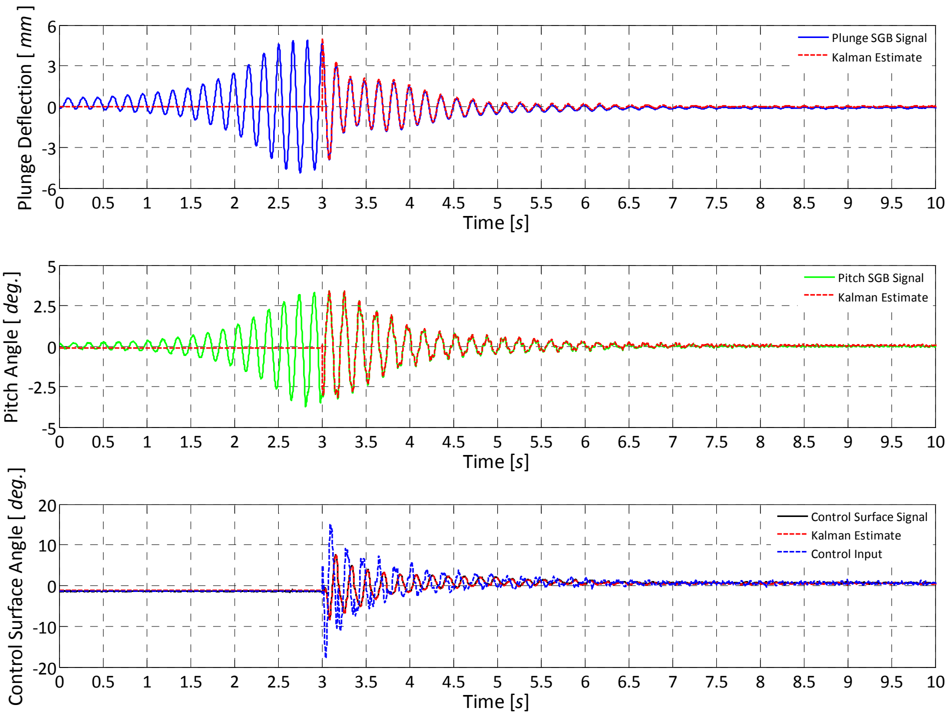


Figure 6.57 Un-optimised LQG Controller Closed-Loop Time Response ($U = 35.6m/s$)

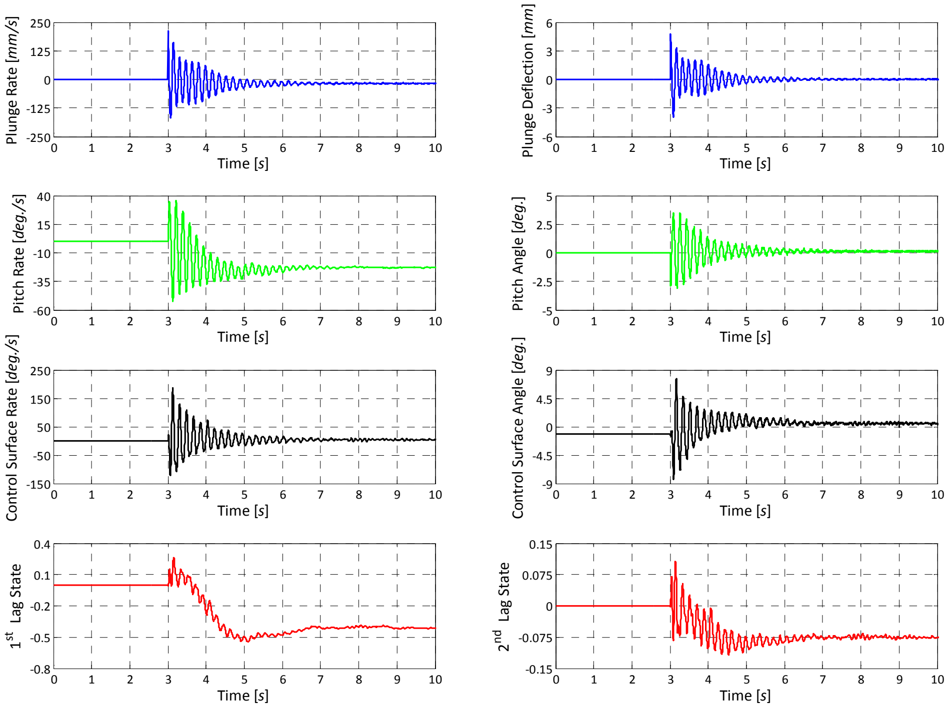


Figure 6.58 Un-optimised LQG Controller Kalman Estimates (*U* = 35.6*m/s*)

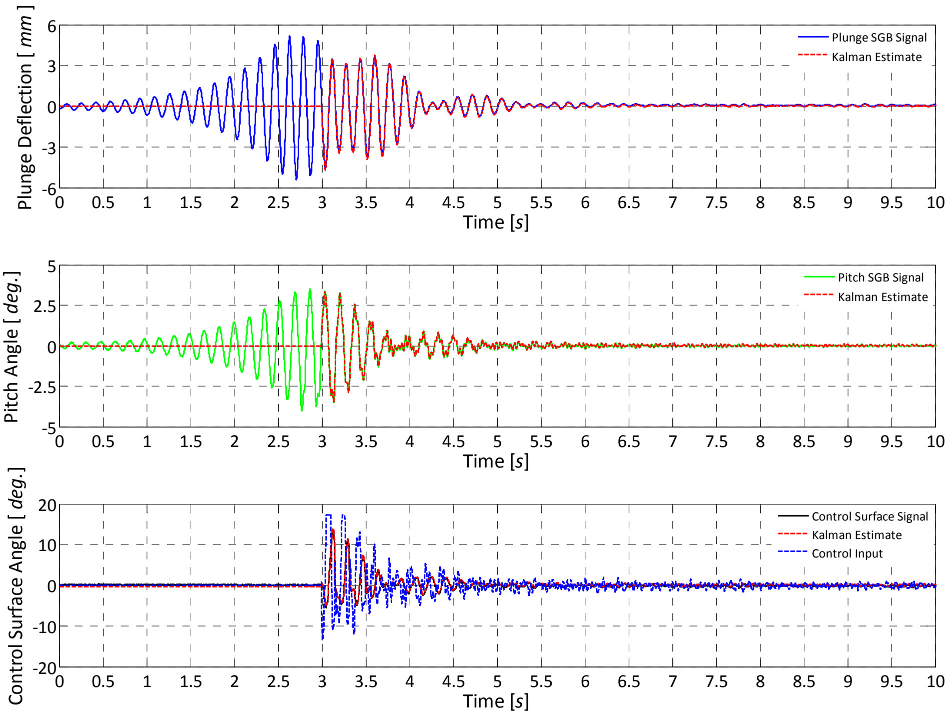


Figure 6.59 Optimised LQG Controller Closed-Loop Time Response (*U* = 35.6*m/s*)

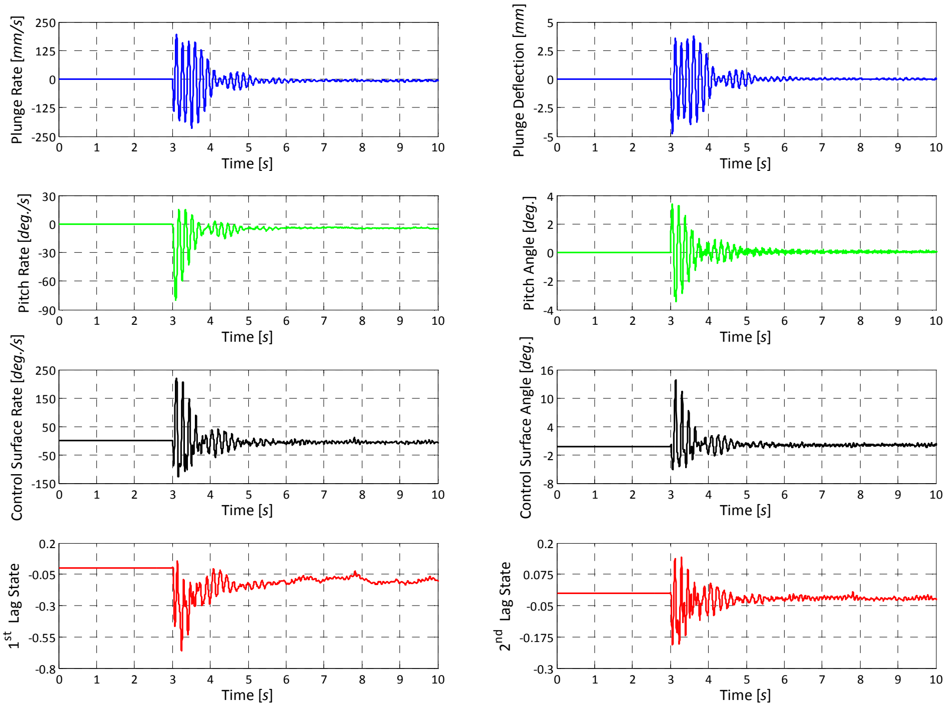


Figure 6.60 Optimised LQG Controller Kalman Estimates (U = 35.6m/s)

6.3.4 Low Speed Wind Tunnel Results

Since the FAST model was very stable at the maximum operating speed of the CWT (Section 6.3.3) it was set up for further testing in the CSIR LSWT with its higher maximum operating speed of nominally 135.0m/s. Initial open-loop tests were run to confirm that the model's behaviour in the LSWT and CWT were similar. Testing then focused on extending the flutter boundary of the FAST model in its baseline cg configuration using a LQG controller designed for a wind tunnel speed of 25.0m/s (Table 6.7).

Open-Loop Results

The LSWT wind-off open-loop model time responses and normalised PSDs are given in Figure G.5 and Figure G.6 in Appendix G, which can be compared with the corresponding results from the CWT, Figure 6.26 and Figure 6.27 in Section 6.3.2. The comparison shows that the model had similar dynamic characteristics in both wind tunnels. The measured natural plunge and pitch frequencies of 5.6Hz. and 6.2Hz. respectively of the FAST model on the LSWT mount were slightly lower than the corresponding frequencies of 5.7Hz. and 6.3Hz. measured in the CWT. This was due to the CWT cantilever mount being slightly stiffer than the LSWT floor mount. Open-loop flutter of the model in the LSWT occurred at 25.0m/s at a frequency of 5.9Hz., as seen in Figure G.7 and Figure G.8, compared with 23.0m/s at a frequency of 5.8Hz. in the CWT. The 2.0m/s discrepancy in the two flutter speeds was accounted for by the variation in the effective wingspans of the model in the LSWT (343mm) and CWT (335mm) and the slight differences in the natural frequencies of the pitch and plunge modes on the different wind tunnel mounts.

Closed-Loop Results

Once open-loop tests had established the flutter speed in the LSWT and demonstrated that the FAST model behaviour was the same in the LSWT as it was in the CWT, the control system was activated and the wind tunnel was run at speeds from 25.0*m/s* to 53.9*m/s* in increments of nominally 2.0*m/s*. At each test point the FAST model was given an initial displacement after which its closed-loop response was observed and recorded. The control system was robust and did not allow any unstable oscillations at any of the test points. The closed-loop response of the FAST model at 53.9*m/s* (the maximum test speed in the LSWT) is shown Figure 6.61 and the corresponding Kalman state estimates are shown in Figure 6.62. The stable response of the model at this speed represents a 115.6% increase (or 2.16 times) in the flutter speed and a 364.8% increase (or 4.65 times) in the dynamic pressure over the open-loop flutter point.

Whilst the FAST model was still stable at 53.9*m/s* and the controller showed no signs of saturation, flutter boundary extension tests did not continue past this speed. It was deemed too risky to continue at such high wind tunnel speeds and dynamic pressures, and the active control had already been proven well beyond project requirements. At this speed static loads on the model and mount were very high due to the high dynamic pressure, and the risk of structural failure was great even though the flutter was suppressed. Any malfunction of the control system at this speed or higher would have resulted in catastrophic flutter, total (almost instantaneous) destruction of the model and damage to the wind tunnel. To demonstrate the anticipated violently rapid and divergent nature of supercritical open-loop flutter of the FAST model, the simulated open-loop response of the model at a wind tunnel speed of 53.9m/s is shown in Figure 6.63. In a very short time, just 0.75*s*, the predicted plunge displacement exceeds -450*mm* and the pitch angle exceeds 50°. This of course is meaningless as the model would break up long before reaching these amplitudes, but highlights the importance of having adequate mechanical stops incorporated in the wind tunnel side-wall to limit model deflections should the control system become ineffective during supercritical closed-loop testing. The red dotted lines in Figure 6.63 indicate the limitations the installed mechanical stops would impose on the FAST model displacements.

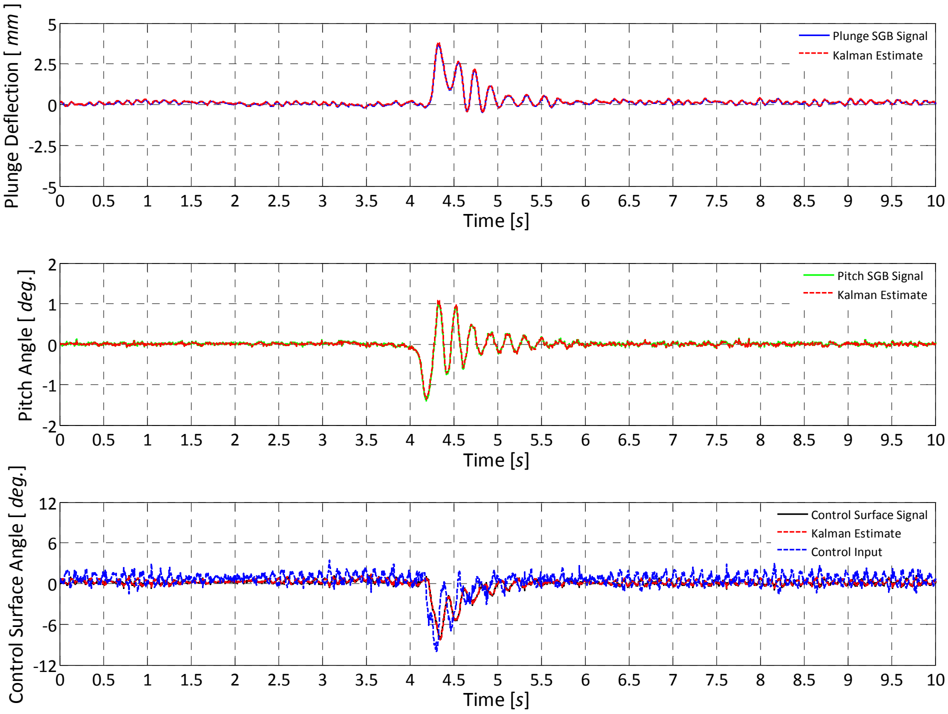


Figure 6.61 FAST Model LSWT Closed-Loop Response (Baseline cg Position; *U* = 53.9*m/s*)

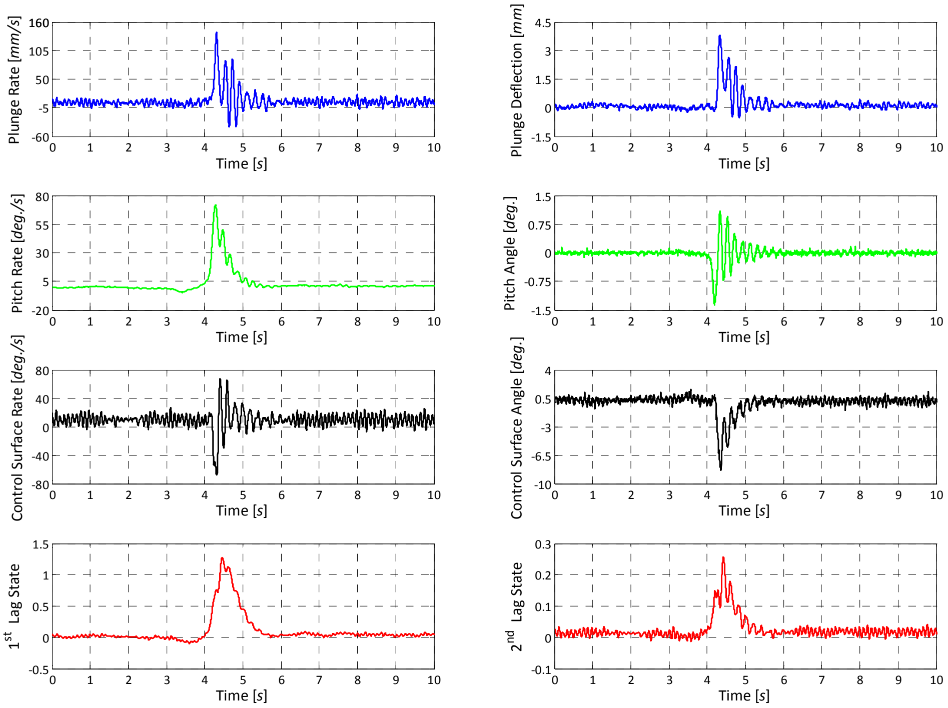


Figure 6.62 FAST Model LSWT Kalman Estimates (Baseline cg Position; *U* = 53.9*m/s*)

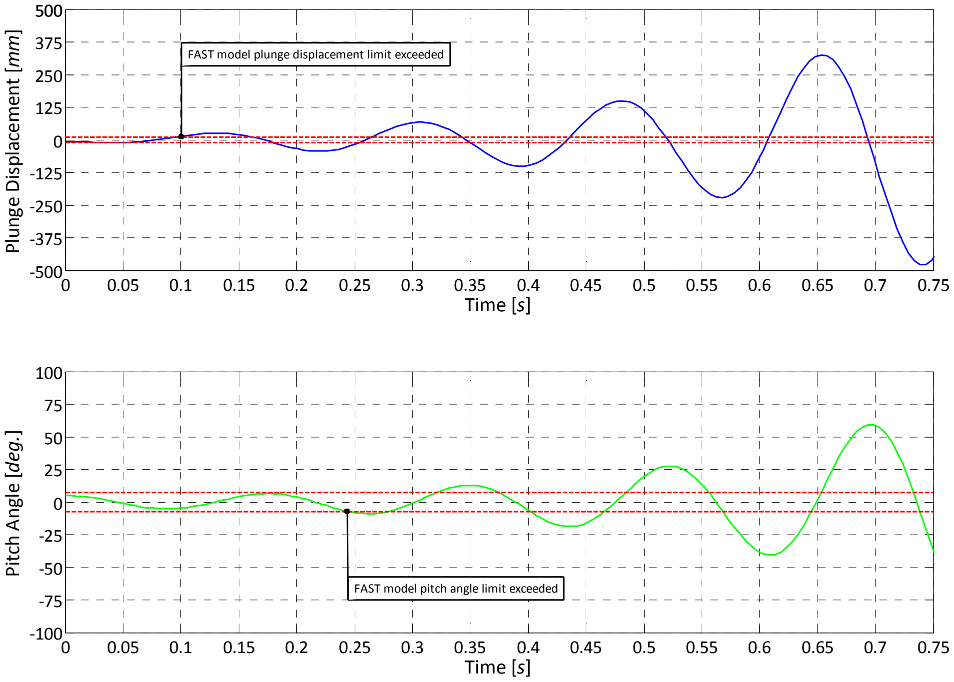


Figure 6.63 Simulated Model Open-Loop Response (Baseline cg Position; *U* = 53.9*m*/*s*)

6.4 Summary of Results

Careful calibration and characterisation of the FAST model transducers and control surface actuator allowed for accurate measurement of model states during open- and closed-loop wind tunnel testing. Some setbacks and unexpected model behaviour during initial wind tunnel testing of the FAST model in the CWT were resolved and subsequent open- and closed-loop testing produced results that clearly demonstrated the principle of active flutter suppression using a trailing edge control surface. Excellent agreement between theory and experiment was achieved through careful design and setup of the FAST wind tunnel model. Open-loop results matched two different theoretical predictions well and were repeatable, even when testing the model in two different wind tunnels. The closed-loop response of the FAST model far exceeded expectations, with the LQG feedback controller designed for an airspeed of 25.0*m*/*s* proving to be highly effective and robust over a large wind tunnel speed range. For each configuration tested, developed flutter cycles were suppressed by the LQG controller and model oscillations were limited to within ±0.5*mm* in the plunge degree-of-freedom and ±0.25° in the pitch degree-of-freedom, in under 5*s*. Only small control surface deflections were required at the higher wind tunnel speeds because of the high aerodynamic loads generated by the control surface at these speeds. Flutter boundary extension tests demonstrated that flutter could not be induced at an airspeed of 35.6*m*/*s* in the CWT and 53.9*m*/*s* in the LSWT, which translates to an increase of 54.8% and 115.6% respectively in the FAST model open-loop flutter speeds of 23.0*m*/*s* and 25.0*m*/*s* in each tunnel.

Chapter 7

Conclusions and Recommendations

Conclusions from the research described in this dissertation are drawn in Section 7.1 and recommendations for future work are made in made in Section 7.2. The chapter concludes with a brief summary and some thoughts on broader research goals to work towards in applying active flutter suppression technology to unmanned and full-scale aircraft.

7.1 Conclusions

The design of a small-scale active flutter suppression wing model with a movable trailing edge control surface for wind tunnel testing on a flexible mount in low speed wind tunnels is feasible. This was proven by the manufacture of a 120*mm* chord, 360*mm* span model wing which was flutter tested on a unique pitch-plunge flexible mount. The model incorporated a 25*mm* chord full-span trailing edge control surface which could be driven by an internally mounted model aircraft servo via a torsion spring or by an external linear actuator and crank mechanism. Instrumentation required for successful testing was minimal, comprising full bending and torsion strain gauge bridges to measure the model dynamics and a rotary Hall-effect sensor to measure control surface rotation.

A full eight state ($\dot{h}$, $\dot{\alpha}$, $\dot{\beta}$, h, α, β, ℓ_1, ℓ_2) aeroservoelastic state-space mathematical model of a three degree-of-freedom binary flutter model was developed in MATLAB™. This mathematical model included the linearised structural dynamics of the model/mount, the linearised unsteady aerodynamic forces acting on the wing and the control surface and the actuation mechanism dynamics. A Kalman filter to estimate unmeasured system states and a LQR feedback regulator to suppress flutter of the FAST model through a trailing edge control surface were designed. Control surface commands were calculated from a fixed LQR feedback gain multiplied by the three measured states (h, α, β) and five Kalman estimated states ($\dot{h}$, $\dot{\alpha}$, $\dot{\beta}$, ℓ_1, ℓ_2). Combined, the LQR regulator and Kalman observer formed an effective LQG flutter suppression controller. A block diagram representation of the aeroservoelastic model was developed in Simulink™, which enabled the effect of unmodeled nonlinearties in the system (such as free-play, dead-band, actuator saturation and a fixed time delay) to be investigated. The response of the flutter suppression controller in the presence of these nonlinearities was assesed, and it was shown that the system was robust, provided these nonlinearities remained small.

The model and mount were integrated into both the CWT and the LSWT of the CSIR by means of custom designed adjustable steel brackets and new wind tunnel test section windows. In the CWT the model mount structure clamped directly to the columns supporting the wind tunnel working section, whilst in the LSWT the model support bracket was bolted to a stand mounted on the wind tunnel test section support beams on the floor outside the working section.

Software for data acquisition and implementation of the LQG flutter suppression controller using the National Instruments™ NI USB-6211 DAQ module was designed and written in Microsoft™ C# 2008 Express Edition. Gentle open-loop flutter was demonstrated successfully in the CWT with the FAST model in baseline, forward and aft cg position configurations. The open-loop flutter speeds in these three configurations were 23.0*m/s*, 21.4*m/s* and 30.0*m/s* respectively and the corresponding flutter frequencies were 5.8*Hz.*, 6.1*Hz.* and 6.0*Hz*..

Closed-loop testing of the FAST model in each cg position configuration was completed in the CWT. In each case a sustained, developed flutter cycle was established at the open-loop flutter speed of the configuration and then suppressed within 5*s* of activation of the control. Control surface deflections were initially of the order of ±15.0° to catch the flutter, but decreased to ±1.0° within 2*s*. The control system was able to limit the model's plunge displacement to ±0.5*mm* and its pitch angle to ±0.25° at these open-loop flutter speeds.

After the demonstration of flutter suppression of the FAST model in the different cg position configurations, at their respective critical open-loop flutter speeds, flutter boundary extension tests with the FAST model in its baseline cg position configuration were conducted in the CWT. In these tests with the control system active, flutter could not be induced up to the CWT's maximum operating speed of 35.0*m/s*, even though the controller was optimised for an airspeed of only 25.0*m/s*. This extension of the flutter boundary of the FAST model represented a speed increase of 54.8% and a dynamic pressure increase of 139.6%. Further flutter boundary extension tests with the FAST in its baseline cg position configuration were successfully conducted in the LSWT using the 25.0*m/s* controller design. These tests were arbitrarily stopped at a wind tunnel speed of 53.9*m/s*, at which point flutter could still not be induced. This extension of the flutter boundary of the FAST model represents a speed increase of 115.6% and a dynamic pressure increase of 364.8%. Similar flutter boundary extension margins achieved with other binary wind tunnel flutter models described in the literature are all considerably lower than this (on average typically < 40% increase in the flutter boundary).

It is clear that the FAST model and LQG control system is an ideal platform for demonstration of open-loop binary flutter and closed-loop flutter control, and that the model can be used to research different active control laws. Some recommendations for such follow-on work are given in Section 7.2.

7.2 Recommendations

A significant amount of work was done in the course of this research to successfully develop an aeroservoelastic analysis tool for a three degree-of-freedom binary flutter system and a simple, low-cost, robust aeroservoelastic wind tunnel model. Given the results achieved and the hardware and software now in place, scope exists to further develop this aeroservoelastic capability. It is suggested that future work include:

- Further validation of the FAST model's open-loop flutter characteristics through investigation of the effects of a change in mass, mass inertia and model frequencies on its flutter dynamics
- Implementation of gain scheduling into the controller software to demonstrate how more optimal control can be realised by dynamically calculating the LQR and Kalman feedback gains as a function airspeed
- Investigation of the response of a reduced controller that calculates the required control surface angle using only the measured plunge deflection, pitch angle and control surface angle
- Investigation of the effectiveness of other control laws such as loop transfer recovery (LTR) or PID controllers
- Development of a stand-alone onboard electronic controller for flutter suppression

7.3 Summary

Although this research focused on a simple three degree-of-freedom binary flutter model, it demonstrated the benefits of applying active control to an aeroelastic system. Simple mathematical modelling and time-domain control system design techniques were shown to be sufficient for the design of a robust active flutter suppression control system. To achieve the ultimate goal of applying active flutter suppression technology to an unmanned aerial system or full-scale aircraft, a more complex mathematical model and controller would need to be developed. Redundancy would have to be incorporated into the control system design as failure of a UAS or aircraft active flutter suppression control system would be catastrophic. Other factors to be considered in the design of an aircraft flutter suppression system include maintaining the trim configuration of the aircraft, accounting for changing aircraft mass and inertial properties due to fuel burn during flight and store release implications. Active flutter suppression technology is thought to be an integral consideration in new aircraft design as airframes continue to become lighter and operational requirements demand increased performance and wider flight envelopes.

References

Abel, I., 1979, An Analytical Technique for Predicting the Characteristics of a Flexible Wing Equipped With an Active Flutter-Suppression System and Comparison With Wind-Tunnel Data, NASA TP-1367, National Aeronautics and Space Administration.

Barker, J.M., Balas, G.J. and Blue, P.A., 1999, Active Flutter Suppression via Gain-Scheduled Linear Fractional Control, Proceedings of the American Control Conference, San Diego, California.

Bennett, R.M., 1982, Application of Zimmerman Flutter-margin Criterion to a Wind-Tunnel Model, NASA TM-84545, National Aeronautics and Space Administration.

Bhoir, N. and Singh, S.N., 2004, Output Feedback Nonlinear Control of an Aeroelastic System with Unsteady Aerodynamics, Journal of Aerospace Science and Technology, Vol. 8, Issue 3, pp. 195 - 205.

Bisplinghoff, R.L., Ashley, H. and Halfman, R.L., 1957, *Aeroelasticity*, Addison-Wesley Publishing Company, Inc., Reading, Massachusetts.

Block, J.J. and Gilliatt, H., 1997, Active Control of an Aeroelastic Structure, AIAA-97-0016, American Institute of Aeronautics and Astronautics.

Block, J.J. and Strganac, T.W., 1998, Applied Active Control for a Nonlinear Aeroelastic Structure, Journal of Guidance, Control and Dynamics, Vol. 21, No. 6, pp. 838 - 845.

Borglund, D. and Kuttenkeuler, J., 2002, Active wing flutter suppression using a trailing edge flap, Journal of Fluids and Structures, Vol. 16, Issue 3, pp. 271 - 294.

Bradshaw, A., Ruhlan, T. and Woodhead, M.A., 1986, Active Flutter Suppression, Proceedings of the 15th Congress of the International Council of the Aeronautical Sciences, London, England.

Bryson, A.E. and Ho, Y., 1975, *Applied Optimal Control: Optimization, Estimation and Control*, Blaisdell, Waltham, Massachusetts.

Chen, C., 1999, *Linear System Theory and Design*, Oxford University Press, New York.

Chowdhury, A.G. and Sarkar, P.P., 2003, A new technique for identification of eighteen flutter derivatives using a three-degree-of-freedom section model, Journal of Engineering Structures, Vol. 25, No. 12, pp. 1763 - 1772.

Cole, S.R., 1986, Exploratory Flutter Test in a Cryogenic Wind Tunnel, Journal of Aircraft, Vol. 23, No. 12, pp. 904 - 911.

Collar, A.R., 1946, The Expanding Domain Of Aeroelasticity, The Journal of the Royal Aeronautical Society, Vol. 50 (1946), pp. 613 - 636.

Dansberry, B.E., Durham, M.H., Bennett, R.M., Turnock, D.L., Silva, W.A. and Rivera, J.A., 1993, Physical Properties of the Benchmark Models Program Supercritical Wing, NASA TM-4457, National Aeronautics and Space Administration.

De Marqui, C., Belo, E.M. and Marques, F.D., 2005, A Flutter Suppression Active Controller, Proceedings of the I MECH E Part G Journal of Aerospace Engineering, Vol. 219, No. 1, pp. 19 - 33.

De Marqui, C., Rebolho, D.C., Belo, E.M. and Marques, F.D., 2006, Identification of Flutter Parameters for a Wing Model, Journal of the Brazilian Society of Mechanical Sciences and Engineering, Vol. 28, No. 3, pp. 339 - 346.

da Silva, S. and Júnior, V.L., 2006, Active Flutter Suppression in a 2-D Aerofoil using Linear Matrix Inequalities Techniques, Journal of the Brazilian Society of Mechanical Sciences and Engineering, Vol. 28, No. 1, pp. 84 - 93.

Evans, W. R., 1950, Control Systems Synthesis by Root Locus Method, AIEE Transactions, Vol. 69, Part 2, pp. 66 - 69.

Farmer, M.G., 1982, A Two-Degree of Freedom Mount System with Low Damping for Testing Rigid Wings at Different Angles of Attack, NASA TM-83302, Langley Research Centre.

Flomenhoft, H.I., 1997, *The Revolution in Structural Dynamics: The Big "Shake-up" in Airframe Design*, Dynaflow Press, Florida.

Franklin G.F., Powell, J.D. and Emami-Naeini, A., 1991, *Feedback Control of Dynamic Systems*, Addison-Wesley Publishing Company, Menlo Park, California.

Franklin G.F. and Powell, J.D., 1980, *Digital Control of Dynamic Systems*, Addison-Wesley Publishing Company, Menlo Park, California.

Fung, Y.C., 1955, *An Introduction to the Theory of Aeroelasticity*, John Wiley and Sons, Inc., New York.

Garrick, I.E., 1938, On Some Fourier Transforms in the Theory of Non-Stationary Flows, Proceedings of the 5th Congress on Applied Mechanics, Wiley, New York, pp. 590 - 593.

Haidl, G., Hönlinger, H. and Lotze, A., 1981, The F-4 Flutter Suppression Program, Proceedings of the International Symposium on Aeroelasticity, Nuremberg, Germany.

Hassig, H.J., 1971, An Approximate True Damping Solution of the Flutter Equation by Determinant Iteration, Journal of Aircraft, Vol. 8, No. 11, pp. 885 - 889.

Heeg, J., 1993, Analytical and Experimental Investigation of Flutter Suppression by Piezoelectric Actuation, NASA-TP-3241, National Aeronautics and Space Administration, Langley Research Center, Hampton, Virginia.

Horikawa, H. and Dowell, E.H., 1979, An Elementary Explanation of the Flutter Mechanism with Active Feedback Controls, Journal of Aircraft, Vol. 16, No. 4, pp. 225 - 232.

James, G., Burley, D., Dyke, P., Searl, J., Steele, N. and Wright, J., 1999, *Advanced Engineering Mathematics*, *Second Edition*, Prentice-Hall, Inc., Singapore.

Johannes, R.P. and Thompson, G.O., 1973, B-52 Control Configured Vehicles Program, Proceedings of the AGARD 17th Meeting of the Guidance and Control Panel, Geilo, Norway.

Jones, W.P., 1945, Aerodynamic Forces on Wings in Non-Uniform Motion, R. & M. 2117, British Aeronautical Research Council.

Kalman, R.E., 1960, A New Approach to Linear Filtering and Prediction Problems, ASME Journal of Basic Engineering, No. 82, pp. 35 - 45.

Karpel, M., 1982, Design for Active Flutter Suppression and Gust Alleviation Using State-Space Aeroelastic Modelling, Journal of Aircraft, Vol. 19, No. 3, pp. 221 - 227.

Kirk, D.E., 1970, *Optimal Control Theory: An Introduction*, Prentice-Hall, Inc., New Jersey.

Ko, J., Strganac, T. W. and Kurdila, A. J., 1997, Nonlinear Control of a Prototypical Wing Section with Torsional Nonlinearity, Journal of Guidance, Control and Dynamics, Vol. 20, No. 6, pp. 1181 - 1189.

Küssner, H.G. and Schwarz, I., 1941, The Oscillating Wing with Aerodynamically Balanced Elevator, NACA TM-991, National Advisory Commission for Aeronautics.

Kwakernaak, H. and Sivan, R., 1972, *Linear Optimal Control Systems*, Wiley Interscience, New York City, New York.

Lau, E. and Krener, A.J., 1999, LPV Control of Two Dimensional Wing Flutter, Proceedings of the 38th Conference on Decision and Control, Phoenix, Arizona.

Leishman, J.G. and Nguyen, 1990, State-Space representation on Unsteady Airfoil Behavior, AIAA Journal, Vol. 28, No. 5, pp. 836 - 844.

Marretta, R.M.A. and Marino, F., 2007, Wing flutter suppression enhancement using a well-suited active control model, Proceedings of the I MECH E Part G Journal of Aerospace Engineering, Vol. 221, No. 3, pp. 441 - 452.

McEver, M.A., Ardelean, E.V., Cole, D.G., Clark, R.L., 2007, Active Control and Closed-Loop Identification of Flutter Instability in Typical Section Airfoil, Journal of Guidance, Control and Dynamics, Vol. 30, No. 3, pp. 733 - 740.

Murray, R. M., 2008, Optimization-Based Control, Department of Control and Dynamical Systems California Institute of Technology, Pasadena, California.

Nachtigal, C.H., 1990, *Instrumentation and Control: Fundamentals and Applications*, John Wiley and Sons, Inc., New York.

Norlander, T., Nilsson, B., Ring, D. and Johansson, U., 2000, A Study on Active Flutter Detection and Control, Proceedings of the National Aerospace and Electronics Conference, Dayton, Ohio.

O'Neil, T. and Strganac T.W., 1998, Aeroelastic Response of a Rigid Wing Supported by Nonlinear Springs, Journal of Aircraft, Vol. 35, No. 4, pp. 616 - 622.

Peloubet, R.P., Haller, R.L. and Bolding, R.M., 1981, F-16 Active Flutter Suppression Program, Proceedings of the International Symposium on Aeroelasticity, Nuremberg, Germany.

Perry, B. and Cole, S.R., 1995, Summary of an Active Flexible Wing Program, Journal of Aircraft, Vol. 32, No. 1, pp. 10 - 15.

Platanitis, G. and Strganac, T.W., 2004, Control of a Nonlinear Wing Section Using Leading- and Trailing-Edge Surfaces, Journal of Guidance, Control and Dynamics, Vol. 27, No. 1, pp. 52 - 58.

Poirel, D., Harris, Y. and Benaissa, 2008, Self-sustained aeroelastic oscillations of a NACA0012 airfoil at low-to-moderate Reynolds numbers, Journal of Fluids and Structures, Vol. 24, Issue 5, pp. 700 - 719.

Predoiu, I., Stoia-Djeska, M., Frunzulică, F., Popescu, I. and Silisteanu, P., 2007, The Synthesis of Controllers for Flutter Suppression - Theory And Experiment, Proceedings of the SISOM 2007 and Homagial Session of the Commission of Acoustics, Bucharest, Romania.

Prime, Z., Cazzolato, B., Doolan, C. and Strganac, T., 2009, Linear-Parameter-Varying Control of an Improved Three-Degree-of-Freedom Aeroelastic Model, Journal of Guidance, Control and Dynamics, Vol. 33, No. 2, pp. 615 - 619.

Roark, R.J. and Young, W.C., 1975, *Formulas for Stress and Strain*, *Fifth Edition*, McGraw-Hill International Book Company, Tokyo, Japan.

Rodden, W.P. and Stahl, B., 1969, A Strip Method for Prediction of Damping in Subsonic Wind Tunnel and Flight Flutter Tests, Journal of Aircraft, Vol. 6, No. 1, pp. 9 - 17.

Roger, K.L., Hodges, G.E. and Felt, L., 1975, Active Flutter Suppression - A Flight Test Demonstration, Journal of Aircraft, Vol. 12, No. 6, pp. 551 - 556.

Rhulin, C.L., Watson, J.J., Ricketts, R.H. and Dogget, R.V., 1983, Evaluation of Four Subcritical Response Methods for On-line Prediction of Flutter Onset in Wind Tunnel Tests, Journal of Aircraft, Vol. 20, No. 10, pp. 835 - 840.

Scanlan, R.H. and Rosenbaum, R., 1968, *Introduction to the Study of Aircraft Vibration and Flutter*, Dover Publications, Inc., New York.

Strganac, T.W., Ko, J., Thompson, D.E. and Kurdila, A.J., 2000, Identification and Control of Limit Cycle Oscillations in Aeroelastic Systems, Journal of Guidance, Control and Dynamics, Vol. 23, No. 6, pp. 1127 - 1133.

Stroud, K.A., 1987, *Engineering Mathematics, Third Edition*, MacMillan Education Ltd., London.

Sutherland, A.N., 2006, The Aeroelastic Response of a Rigid Wing: A Pitch-Plunge Flutter Investigation, BSc. (Eng.) (Aero.) Research Report, University of the Witwatersrand, Johannesburg, South Africa.

Sutherland, A.N., 2008, A Small Scale Pitch-Plunge Flutter Model for Active Flutter Control Research, Proceedings of the 26th Congress of the International Council of the Aeronautical Sciences, Anchorage, Alaska.

Tewari, A., 2002, *Modern Control Design with MATLAB and SIMULINK*, John Wiley and Sons, Ltd., West Sussex, England.

Theodorsen, T., 1935, General Theory of Aerodynamic Instability and the Mechanism of Flutter, NACA TR-496, National Advisory Commission for Aeronautics.

Theodorsen, T. and Garrick, I.E., 1942, Flutter Calculations in Three Degrees of Freedom, NACA TR-741, National Advisory Commission for Aeronautics.

Thompson, G.O. and Kass, G.J., 1971, Active Flutter Suppression - An Emerging Technology, Journal of Aircraft, Vol. 9, No. 3, pp. 230 - 235.

Tse, F.S., Morse, I.E. and Hinkle, R.T., 1978, *Mechanical Vibrations: Theory and Applications*, Allyn and Bacon, Inc., Boston, Massachusetts.

Turner, M.R., 1975, Active Flutter Suppression, Proceedings of the 40th Meeting of the Structures and Materials Panel, Specialist Meeting on Flutter Suppression and Structural Load Alleviation, Brussels, Belgium.

Vipperman, J.S., Clark, R.L., Conner, M. and Dowell, E.H., 1998, Experimental Active Control of a Typical Section Using a Trailing-Edge Control surface, Journal of Aircraft, Vol. 35, No. 2, pp. 224 - 229.

Wagner, H., 1925, Über die Entstehen des Dynamischen Auftriebes von Tragflügeln, Zeitschrift für angewandte Mathematik und Mechanik, Vol. 5, No. 1, pp. 17 - 35.

Waszak, M.R., 1997, Robust Multivariable Flutter Suppression for the Benchmark Active Control Technology (BACT) Wind Tunnel Model, Proceedings of the Eleventh Symposium on Structural Dynamics and Control, Virginia Polytechnic Institute and State University, Blacksburg, Virginia.

Waszak, M.R., 1998, Modelling the Benchmark Active Control Technology Wind-Tunnel Model for Active Control Design Applications, NASA-TP-1998-206270, National Aeronautics and Space Administration.

Waszak, M.R. and Fung, J., 1996, Parameter Estimation and Analysis of Actuators for the BACT Wind-Tunnel Model, Proceedings of the AIAA Atmospheric Flight Mechanics Conference, San Diego, California.

Welch, G. and Bishop, G., 2006, An Introduction to the Kalman Filter, University of North Carolina at Chapel Hill, Department of Computer Science, Chapel Hill, North Carolina.

Window, A. L. and Holister, G. S., 1982, *Strain Gauge Technology*, Applied Science Publishers, London.

Zeiler, T.A., 2000, Results of Theodorsen and Garrick Revisited, Journal of Aircraft, Vol. 37, No. 5, pp. 918 - 920.

Zimmerman, N.H. and Weissenburger, J.T., 1964, Prediction of Flutter Onset Speed Based on Flight Testing at Subcritical Speeds, Journal of Aircraft, Vol. 1, No. 4, pp. 190 - 202.

Zwaan, R.J., 1990, Aeroelasticity of Aircraft, Faculteit der Luchtvaart en Ruimtevaarttechniek, Technische Universiteit Delft, The Netherlands.

Bibliography

Adams, W.M. and Tiffany, S.H., 1985, Development of a Flutter Suppression Control Law by use of Linear Quadratic Gaussian and Constrained Optimization Design Techniques, Proceedings of the 2nd International Symposium on Aeroelasticity and Structural Dynamics, Aachen, Germany.

Adams, W.M., 1995, Design and Multifunction Tests of a Frequency Domain-Based Active Flutter Suppression System, Journal of Aircraft, Vol. 32, No. 1, pp. 52 - 60.

Balakrishnan, A.V., 1978, Active Control of Airfoils in Unsteady Aerodynamics, Journal of Applied Mathematics and Optimisation, Vol. 4 (1978), pp. 171 - 195.

Barker, J.M. and Balas, G.J., 1999, Gain-Scheduled Linear Fractional Control for Active Flutter Suppression, Journal of Guidance, Control and Dynamics, Vol. 22, No. 4, pp. 507 - 512.

Bergmann, G.E. and Sevart, F.D., 1975, Design and Evaluation of Miniature Control Surface Actuation Systems for Aeroelastic Models, Journal of Aircraft, Vol. 12, No. 3, pp. 129 - 134.

Billingley, J., 1989, *Controlling with Computers: Control Theory and Practical Digital Systems*, McGraw-Hill Book Company (UK) Limited, England.

Bishop, R.H., 1993, *Modern Control Systems Analysis and Design using MATLAB*, Addison-Wesley Publishing Company, Inc, Reading, Massachusetts.

Bolton, W., 2003, *Mechatronics: Electronic Control Systems in Mechanical and Electrical Engineering*, Third Edition, Pearson Education Limited, England.

Botez, R.H., Doin, A., Biskri, D.E., Cotoi, I., Hamza, D. and Parvu, P., 2003, Method for Flutter Aero-servoelastic Open Loop Analysis, Canadian Aeronautics and Space Journal, Vol. 49, No. 4, pp. 179 - 190.

Broadbent, E.G., 1954, The Elementary Theory of Aero-Elasticity: A Series of Articles Written from the Standpoint of a Structural Engineer for Students and Junior Members of Aircraft Design Teams, Journal of Aircraft Engineering and Aerospace Technology, Vol. 26, No. 5, pp. 145 - 153.

Chang, S., 1984, Modelling and Control of Aircraft Flutter Problem, Proceedings of the 23rd Conference on Decision and Control, Las Vegas, Nevada.

Copley, J.C., Skingle, C.W. and Turner, M.R., 1983, Design and Demonstration of a Digital Flutter Suppression System on a Wind Tunnel Model, Proceedings of the 24th AIAA/ASME/ASCE/AHS Structures, Structural Dynamics and Materials Conference, Lake Tahoe, Nevada.

de Silva, C.W., 2005, *Mechatronics: An Integrated Approach*, CRC Press, Boca Raton, Florida.

Destuynder, R., 1981, Recent Developments in Flutter Suppression of Wings with Stores, Proceedings of the International Symposium on Aeroelasticity, Nuremberg, Germany.

Destuynder, R., 1985, Active Control of the Buffeting Response on a Large Modern Civil Airplane Configuration in Wind Tunnel, Proceedings of the 2nd International Symposium on Aeroelasticity and Structural Dynamics, Aachen.

Destuynder, R., 1987, Multi-Control System in Unsteady Aerodynamics using Spoilers, Proceedings of the 28th AIAA/ASME/ASCE/AHS Structures, Structural Dynamics and Materials Conference, Monterey, California.

Dowell, E.H., Crawley, E.F., Curtiss, H.C., Peters, D.A., Scanlan, R.H. and Sisto, F., 1995, *A Modem Course in Aeroelasticity*, *Third Edition*, Kluwer Academic Publishers, The Netherlands.

Edwards, J.W., 1977, Unsteady Aerodynamic Modelling and Active Aeroelastic Control, Stanford University Guidance and Control Laboratory Department of Aeronautics and Astronautics, Stanford, California.

Garrard, W.L., Liebst, B.S. and Farm, T.A., 1987, Eigenspace Techniques for Active Flutter Suppression, NASA CR-4071, National Aeronautics and Space Administration.

Ghiringhelli, G.L. and Mantegazza, M.L., 1988, Active Flutter Suppression of a Wing Model, Proceedings of the 16th Congress of the International Council of the Aeronautical Sciences, Jerusalem, Israel.

Gilbert, M.G., 1990, An Analytical Sensitivity Method for use in Integrated Aeroservoelastic Aircraft Design, Journal of Mechanical Systems and Signal Processing, Vol. 4, No. 20, pp. 215 - 231.

Han, J., Tani, J. and Qiu, J., 2006, Active flutter suppression of a lifting surface using piezoelectric actuation and modern control theory, Journal of Sound and Vibration, Vol. 291 (2006), pp. 706 - 722.

Hariharan, N. and Leishman, J.G., 1996, Unsteady Aerodynamics of a Flapped Airfoil in Subsonic Flow by Indicial Concepts, Journal of Aircraft, Vol. 33, No. 5, pp. 855 - 868.

Horowitz, P. and Hill, W., 1995, *The Art of Electronics*, *Second Edition*, Cambridge University Press, Cape Town, South Africa.

Irving, A. and Noll, T.E., 1988, Research and Applications in Aeroservoelasticity at the NASA Langley Research Center, Proceedings of the 16th Congress of the International Council of the Aeronautical Sciences, Jerusalem, Israel.

Jones, B.L., 2004, *Sams Teach Yourself the C# Language in 21 Days*, Sams Publishing, Indianapolis, Indiana.

Katz, J. and Plotkin, A., 2001, *Low-Speed Aerodynamics*, *Second Edition*, Cambridge University Press, London, United Kingdom.

Lanz, M. and Mantegazza, P., 1982, Design of Compensated Flutter Suppression Systems, Proceedings of the 13th Congress of the International Council of the Aeronautical Sciences, Seattle, Washington.

Ledin, J., 2004, *Embedded Control Systems in C/C++: An Introduction for Software Developers Using MATLAB*, CMP Books, San Francisco.

Librescu, L., Marzocca, P. and Silva, W.A., 2005, Aeroelasticity of 2-D lifting surfaces with time-delayed feedback control, Journal of Fluids and Structures, Vol. 20, Issue 2, pp. 197 - 215.

Livne, E., 1999, Integrated Aeroservoelastic Optimization: Status and Direction, Journal of Aircraft, Vol. 36, No. 1, pp. 122 - 145.

Livne, E., 2003, Future of Airplane Aeroelasticity, Journal of Aircraft, Vol. 40, No. 6, pp. 1066 - 1092.

Lodge, C.G. and Baldock, J.C.A., 1981, Identification and Control of Flutter on Military Combat Aircraft, Proceedings of the International Symposium on Aeroelasticity, Nuremberg, Germany.

Lottati, I., 1986, The Role of Structural and Aerodynamic Damping on the Aeroelastic Behavior of Wings, Journal of Aircraft, Vol. 23, No. 7, pp 606 - 608.

Moudgalya, K.M., 2007, *Digital Control*, John Wiley and Sons, Ltd., West Sussex, England.

Mukhopadhyay, V., 1995, Flutter Suppression Control Law Design and Testing for the Active Flexible Wing, Journal of Aircraft, Vol. 32, No. 1, pp. 45 - 51.

Mukhopadhyay, V., 1999, Transonic Flutter Suppression Control Law Design, Analysis and Wind-Tunnel Results, Proceedings of the International Forum on Aeroelasticity and Structural Dynamics, Williamsburg, Virginia.

Newsom, J.R. and Pototzky, A.S., 1981, Analysis and Flight Data for a Drone Aircraft with Active Flutter Suppression, Journal of Aircraft, Vol. 19, No. 11, pp. 1012 - 1018.

Niblett, L.T., 1988, A Guide to Classical Flutter, The Aeronautical Journal, Vol. 92, No. 919, pp. 339 - 354.

Ohta, H., Fujimori, A., Nikiforuk, P.N. and Gupta, M.M., 1989, Active Flutter Suppression for Two-Dimensional Airfoils, Journal of Guidance, Control and Dynamics, Vol. 12, No.2, pp. 188 - 194.

Otnes, R.K. & Enochson, L., 1978, *Applied Time series Analysis, Volume 1: Basic Techniques*, John Wiley and Sons, New York.

Palaniappan, K., Sahu, P., Alonso, J. and Jameson, A., 2006, Active Flutter Control using an Adjoint Method, Proceedings of the 44th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada.

Patil, M.J. and Hodges, D.H., 2002, Output Feedback Control of the Nonlinear Aeroelastic Response of a Slender Wing, Journal of Guidance, Control and Dynamics, Vol. 25, No. 2, pp. 302 - 308.

Patil, M.J. 2003, Energy Perspectives in Aeroelastic Control, Proceedings of the 44th AIAA/ASME/ASCE/AHS Structures, Structural Dynamics and Materials Conference, Norfolk, Virginia.

Pendleton, E.W., Bessette, D., Field, P.B., Miller, G.D. and Griffin, K.E., 2000, Active Aeroelastic Wing Flight Research Program: Technical Program and Model Analytical Development, Journal of Aircraft, Vol. 37, No. 4, pp. 554 - 561.

Perisho, C.H., Triplett, W.E., and Mykytow, W.J., 1975, Design Considerations for an Active Flutter Suppression System for Fighter Wing/Store Flutter, Proceedings of the 40th Meeting of the Structures and Materials Panel, Specialist Meeting on Flutter Suppression and Structural Load Alleviation, Brussels, Belgium.

Poyneer, R.E., 1975, Design and Evaluation of a Multi-Surface Control System for the CCV B-52, Journal of Aircraft, Vol. 12, No. 3, pp. 135 - 138.

Rae, W.H. and Pope, A., 1984, *Low-Speed Wind Tunnel Testing*, John Wiley and Sons, Inc., New York.

Raja, S. and Upadhya, A.R., 2007, Active Control of Wing Flutter using Piezoactuated Surface, Journal of Aircraft, Vol. 44, No. 1, pp. 71 - 80.

Reed, W.H. 1981, Aeroelasticity Matters: Some Reflections on Two Decades of Testing in the NASA Langley Transonic Dynamics Tunnel, Proceedings of the International Symposium on Aeroelasticity, Nuremberg, Germany.

Sandford, M.C., Abel, I. and Gray, D.L., 1975, Transonic Study of Active Flutter Suppression Based on an Energy Concept, Journal of Aircraft, Vol. 12, No. 2, pp. 72 - 77.

Schweiger, J. and Kramer, J., 1999, Active Aeroelastic Aircraft and its Impact on Structure and Flight Control System Design, Proceedings of the RTO AVT Specialists' Meeting on Structural Aspects of Flexible Aircraft Control, Ottawa, Canada.

Singh, S.N. and Yim, W., 2003, State feedback control of an aeroelastic system with structural nonlinearity, Journal of Aerospace Science and Technology, Vol. 7, No. 1, pp. 23 - 31.

Stoltz, P.M., 1981, Unsteady Aeroelastic Modelling and Trailing-Edge Flap Control of an Experimental Wing in a Two-Dimensional Wind Tunnel, PhD. Thesis, Stanford University, Palo Alto, California.

Suleman, A. and Costa, A.P., 2004, Adaptive control of an aeroelastic flight vehicle using piezoelectric actuators, Journal of Computers and Structures, Vol. 82 (2004), pp. 1303 - 1314.

Sutherland, A.P.N., 1987. The use of Dynamically- and Elastically-Scaled Wind Tunnel Models for Aeroelastic Analysis: A Demonstration of Flutter Prediction and Measurement using a Generic Low Speed Cantilever Wing Model, MSc. (Eng.) Dissertation, University of the Witwatersrand, Johannesburg, South Africa.

Sutherland, A.N., 2010, A Demonstration of Pitch-Plunge Flutter Suppression using LQG Control, Proceedings of the 27th Congress of the International Council of the Aeronautical Sciences, Nice, France.

Theodorsen, T. and Garrick, I.E., 1942, Nonstationary Flow About a Wing-Aileron-Tab Combination Including Aerodynamic Balance, NACA TR-736, National Advisory Commission for Aeronautics.

Thompson, G.O. and Sevart, F.D., 1975, Wind Tunnel Investigation of Control Configured Vehicle Systems, Proceedings of the 40th Meeting of the Structures and Materials Panel, Specialist Meeting on Flutter Suppression and Structural Load Alleviation, Brussels, Belgium.

Toker, O. and Özbay, H., 1995, Robustness analysis of controllers designed for active flutter suppression, Proceedings of the American Control Conference, Seattle, Washington.

Trickey, S.T., Virgin, L.N. and Dowell, E.H., 2002, The Stability of Limit-Cycle Oscillations in a Nonlinear Aeroelastic System, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, Vol. 458, No. 2025, pp. 2203 – 2226.

Triplett, W.E., 1972, A Feasibility Study of Active Wing/Store Flutter Control, Journal of Aircraft, Vol. 9, No. 6, pp. 438 - 444.

Triplett, W.E., Kappus, H.F. and Landy, R.J., 1973, Active Flutter Control - An Adaptable Application to Wing/Store Flutter, Journal of Aircraft, Vol. 10, No. 11, pp. 669 - 678.

Turner, M.R., 1981, Active Control of Near Frequency Coalescence Flutter, Proceedings of the International Symposium on Aeroelasticity, Nuremberg, Germany.

van Zyl, L.H., 1993, Use of Eigenvectors in the Solution of the Flutter Equation, Journal of Aircraft, Vol. 30, No. 4, pp. 553 - 554.

Yang, Y.C., Zhao, L.C. and Jiang. J.S., 1995, A Semi-Active Flutter Control Scheme for a Two-Dimensional Wing, Journal of Sound and Vibration, Vol. 184, No. 1, pp. 1 - 7.

Zeiler, T.A. and Weisshaart, T.A., 1988, Integrated Aeroservoelastic Tailoring of Lifting Surfaces, Journal of Aircraft, Vol. 25, No. 1, pp. 76 - 83.

Zhu, D.M., Gu, Z.Q., Can, M.H., Chen, Z.F. and Wang, W.P., 1986, Research on Active Suppression Technology for Wing/Aileron Flutter, Proceedings of the 15th Congress of the International Council of the Aeronautical Sciences, London, England.

Zimmermann, H., 1991, Aeroservoelasticity, Computer Methods in Applied Mechanics and Engineering, Vol. 90 (1991), pp. 719 - 735.

Appendices

Appendix A *U-g* Flutter Analysis Method

A *U-g* flutter analysis method[15] was used to verify the results from an open-loop *P* flutter analysis performed using the state-space representation (developed in Section 3.1) of the FAST wind tunnel model. The fundamental difference between the two methods is that the structural dynamics model of the *P* flutter analysis method assumes viscous damping, whilst the *U-g* method assumes artificial structural damping (where $\mathbf{D}_s \propto \mathbf{K}_s$)[16]. Artificial structural damping is only valid for harmonic motions, and is thus assumed in the *U-g* method to enforce harmonic motion in the mathematical representation of the 3DOF binary flutter model (Zwaan, 1990). In comparison to Equation 3.4 for viscous damping, the complete structural damping (per unit wingspan) of a 3DOF binary flutter system, as a function of its three vibration modes and their coupled frequency ω, can be expressed by (Scanlan and Rosenbaum, 1968):

$$D_s' = \tfrac{1}{2}\frac{mg_h\omega_h^2}{\omega}\dot{h}^2 + \tfrac{1}{2}\frac{I_\alpha g_\alpha \omega_\alpha^2}{\omega}\dot{\alpha}^2 + \tfrac{1}{2}\frac{I_\beta g_\beta \omega_\beta^2}{\omega}\dot{\beta}^2 \quad \ldots\ldots\ldots\ldots \text{A.1}$$

Applying Lagrange's equation (Equation 3.1) to Equation A.1 and under the assumption of sinusoidal motion of all vibration modes (Equation 3.25), the equations of motion of the 3DOF binary flutter system (given by Equations 3.5, 3.6 and 3.7) can be re-written as:

$$-\omega^2 mh - \omega^2 m x_\alpha b\alpha - \omega^2 m x_\beta b\beta + i\omega\frac{mg_h\omega_h^2}{\omega}h + m\omega_h^2 h = L \quad \ldots\ldots\ldots\ldots \text{A.2}$$

$$-\omega^2 m x_\alpha bh - \omega^2 I_\alpha - \omega^2\left[(c-a)b^2 m x_\beta + I_\beta\right]\beta + i\omega\frac{I_\alpha g_\alpha\omega_\alpha^2}{\omega}\alpha + I_\alpha\omega_\alpha^2\alpha = M_\alpha \quad \ldots\ldots\ldots\ldots \text{A.3}$$

$$-\omega^2 m x_\beta bh - \omega^2\left[(c-a)b^2 m x_\beta + I_\beta\right]\alpha - \omega^2 I_\beta\beta + i\omega\frac{I_\beta g_\beta\omega_\beta^2}{\omega}\beta + I_\beta\omega_\beta^2\beta = M_\beta \quad \ldots\ldots\ldots\ldots \text{A.4}$$

The feedback control term in Equation 3.7 is omitted (i.e. $\beta_c = 0$), since only an open-loop analysis of the system is of interest when using the *U-g* method. In Equations A.2, A.3 and A.4, the modal stiffness terms are expressed as a function of modal frequencies, to be consistent with the structural damping terms, where:

$$\left.\begin{aligned} k_h &= \omega_h^2 m \\ k_\alpha &= \omega_\alpha^2 I_\alpha \\ k_\beta &= \omega_\beta^2 I_\beta \end{aligned}\right\} \quad \ldots\ldots\ldots\ldots \text{A.5}$$

---

[15] This method is also less commonly referred to as the *k* flutter analysis method (Zwaan, 1990).

[16] Structural damping can be thought of as a force of magnitude proportional to the elastic restoring force and in phase with the velocity of an oscillation (Scanlan and Rosenbaum, 1968).

Since the modes of the 3DOF binary flutter model are a plunge displacement h, pitch angle α and control surface rotation β, the modal vector ($\mathbf{q}$) is defined as:

$$\mathbf{q} \triangleq \begin{bmatrix} h & \alpha & \beta \end{bmatrix}^T \tag{A.6}$$

which then allows Equations A.2, A.3 and A.4 to be simplified and written in matrix form as:

$$-\omega^2 \mathbf{M}_s \mathbf{q} + i\mathbf{D}_s \mathbf{q} + \mathbf{K}_s \mathbf{q} = \mathbf{F}_a \tag{A.7}$$

where the structural mass, damping and stiffness matrices of the aeroelastic model are:

$$\mathbf{M}_s = \begin{bmatrix} m & mx_\alpha b & mx_\beta b \\ mx_\alpha b & I_\alpha & (c-a)b^2 mx_\beta + I_\beta \\ mx_\beta b & (c-a)b^2 mx_\beta + I_\beta & I_\beta \end{bmatrix} \tag{A.8}$$

$$\mathbf{D}_s = \begin{bmatrix} mg_h\omega_h^2 & 0 & 0 \\ 0 & I_\alpha g_\alpha \omega_\alpha^2 & 0 \\ 0 & 0 & I_\beta g_\beta \omega_\beta^2 \end{bmatrix} \tag{A.9}$$

$$\mathbf{K}_s = \begin{bmatrix} m\omega_h^2 & 0 & 0 \\ 0 & I_\alpha \omega_\alpha^2 & 0 \\ 0 & 0 & I_\beta \omega_\beta^2 \end{bmatrix} \tag{A.10}$$

It is apparent from Equations A.9 and A.10 (and by the definition of structural damping), that the damping matrix is directly proportional to the stiffness matrix. Mathematically:

$$\mathbf{D}_s = \mathbf{g}\mathbf{K}_s \tag{A.11}$$

where:

$$\mathbf{g} = \begin{bmatrix} g_h & 0 & 0 \\ 0 & g_\alpha & 0 \\ 0 & 0 & g_\beta \end{bmatrix} \tag{A.12}$$

Equation A.12 now allows Equation A.7 to be written as:

$$-\omega^2 \mathbf{M}_s \mathbf{q} + (1 + i\mathbf{g})\mathbf{K}_s \mathbf{q} = \mathbf{F}_a \tag{A.13}$$

Scanlan and Rosenbaum (1968) show that the unsteady aerodynamic forces L, M_α and M_β contained in the generalised force matrix $\mathbf{F}_a$ are complex functions that include circulatory and non-circulatory force and moment contributions, given by:

$$L=\pi\rho b^3\omega^2\left(\frac{L_h}{b}h+\left[L_\alpha-\left(\tfrac{1}{2}+a\right)L_h\right]\alpha+L_\beta\beta\right) \tag{A.14}$$

$$M_\alpha=\pi\rho b^4\omega^2\left(\begin{array}{l}\left[M_h-\left(\tfrac{1}{2}+a\right)L_h\right]\dfrac{1}{b}h+\left[M_\alpha-\left(\tfrac{1}{2}+a\right)\left(L_h+M_h\right)L_h+\left(\tfrac{1}{2}+a\right)^2L_h\right]\alpha\\ +\left[M_\beta-\left(\tfrac{1}{2}+a\right)L_\beta\right]\beta\end{array}\right) \tag{A.15}$$

$$M_\alpha=\pi\rho b^4\omega^2\left(\frac{T_h}{b}h+\left[T_\alpha-\left(\tfrac{1}{2}+a\right)T_h\right]\alpha+T_\beta\beta\right) \tag{A.16}$$

where the aerodynamic coefficients L_h, L_α, L_β, M_h, M_α, M_β, T_h, T_α and T_β in these equations are expressed in terms of the real and imaginary parts of Theodorsen's function (Equation 3.14). The aerodynamic coefficients can be obtained from Scanlan and Rosenbaum (1968). These unsteady aerodynamic forces and moments acting on the 3DOF binary flutter model (given by Equations A.14, A.15 and A.16 respectively) can be compiled in matrix form as:

$$\mathbf{F}_a=\omega^2\mathbf{\Omega}\mathbf{q} \tag{A.17}$$

if an unsteady aerodynamic force matrix, denoted by $\mathbf{\Omega}$, is expressed as:

$$\mathbf{\Omega}=\pi\rho b^4\left[\begin{array}{c:c:c}\dfrac{L_h}{b^2} & \dfrac{1}{b}\left[L_\alpha-\left(\tfrac{1}{2}+a\right)L_h\right] & \dfrac{L_\beta}{b}\\ \hdashline \dfrac{1}{b}\left[M_h-\left(\tfrac{1}{2}+a\right)L_h\right] & M_\alpha-\left(\tfrac{1}{2}+a\right)\left(L_h+M_h\right)L_h+\left(\tfrac{1}{2}+a\right)^2L_h & M_\beta-\left(\tfrac{1}{2}+a\right)L_\beta\\ \hdashline \dfrac{T_h}{b} & T_\alpha-\left(\tfrac{1}{2}+a\right)T_h & T_\beta\end{array}\right] \tag{A.18}$$

The generalised forces in Equation A.13 can now be replaced by the unsteady aerodynamic forces given in Equation A.17. That is:

$$-\omega^2\mathbf{M}_s\mathbf{q}+\left(1+i\mathbf{g}\right)\mathbf{K}_s\mathbf{q}=\omega^2\mathbf{\Omega}\mathbf{q} \tag{A.19}$$

Equation A.19 can be multiplied by $\mathbf{K}_s^{-1}$ and factorised to:

$$\left[-\omega^2\mathbf{K}_s^{-1}\mathbf{M}_s+\left(1+i\mathbf{g}\right)\right]\mathbf{q}=\omega^2\mathbf{K}_s^{-1}\mathbf{\Omega}\mathbf{q} \tag{A.20}$$

which simplifies to:

$$\mathbf{K}_s^{-1}\left[\mathbf{M}_s+\mathbf{\Omega}\right]\mathbf{q}=\frac{\left(1+i\mathbf{g}\right)}{\omega^2}\mathbf{q} \tag{A.21}$$

Finally, if the complex coefficient of **q** in Equation A.21 is defined as:

$$\boldsymbol{\lambda} \triangleq \frac{(1+i\mathbf{g})}{\omega^2} \qquad \text{A.22}$$

then Equation A.21 can be written as:

$$\mathbf{K}_s^{-1}\left[\mathbf{M}_s + \boldsymbol{\Omega}\right]\mathbf{q} = \boldsymbol{\lambda}\mathbf{q} \qquad \text{A.23}$$

which is an eigenvalue equation in the generalised coordinate **q** of the aeroelastic model, given by Equation A.24.

$$\det\left|\mathbf{K}_s^{-1}\left[\mathbf{M}_s + \boldsymbol{\Omega}\right] - \boldsymbol{\lambda}\mathbf{I}\right| = 0 \qquad \text{A.24}$$

The complex eigenvalue solution (Equation A.25) of this equation yields the modal damping and frequencies of the aeroelastic system as a function of reduced frequency *k*.

$$\boldsymbol{\lambda} = \boldsymbol{\delta} + i\boldsymbol{\eta} \qquad \text{A.25}$$

That is, the j^{th} modal damping and frequency (for modes j = 1, 2, 3) can be related to the eigenvalue solution through Equation A.22, where (Zwaan, 1990):

$$\left.\begin{aligned} \omega_j &= \frac{1}{\sqrt{\eta_j}} \\ g_{a,j} &= \omega_j^2 \delta_j \end{aligned}\right\} \qquad \text{A.26}$$

The *U-g* flutter analysis for the 3DOF system is done by calculating the unsteady aerodynamic force matrix (Equation A.18) and associated eigenvalues (Equation A.25) of the complete aeroelastic model over a range of reduced frequencies, corresponding to a range of different airspeeds, and at a specified air density. The calculated modal dampings and frequencies (Equation A.26) are then related to airspeed through Equation 3.10. The calculated artificial dampings are plotted against airspeed, and a curve fitted through these points is interpolated to zero damping in each mode to determine a calculated flutter speed. Sufficient values of *k* are used to ensure that the flutter speed of the model is calculated accurately.

Appendix B Theodorsen's Functions

Theodorsen's functions (Section 3.1.2) required to include the effect of a control surface on the aerodynamics and hence flutter dynamics of a three degree-of-freedom binary flutter system, are listed below (Theodorsen and Garrick, 1942).

$$T_1 = -\tfrac{1}{3}\sqrt{1-c^2}\left(2+c^2\right)+c\cos^{-1}\left(c\right) \tag{B.1}$$

$$T_2 = c\left(1-c^2\right)-\sqrt{1-c^2}\left(1+c^2\right)\cos^{-1}\left(c\right)+c\left[\cos^{-1}\left(c\right)\right]^2 \tag{B.2}$$

$$T_3 = -\left(\tfrac{1}{8}+c^2\right)\left[\cos^{-1}\left(c\right)\right]^2+\tfrac{1}{4}c\sqrt{1-c^2}\cos^{-1}\left(c\right)\left(7+2c^2\right)-\tfrac{1}{8}\left(1-c^2\right)\left(5c^2+4\right) \tag{B.3}$$

$$T_4 = -\cos^{-1}\left(c\right)+c\sqrt{1-c^2} \tag{B.4}$$

$$T_5 = -\left(1-c^2\right)-\left[\cos^{-1}\left(c\right)\right]^2+2c\sqrt{1-c^2}\cos^{-1}\left(c\right) \tag{B.5}$$

$$T_6 = T_2 \tag{B.6}$$

$$T_7 = -\left(\tfrac{1}{8}+c^2\right)\cos^{-1}\left(c\right)+\tfrac{1}{8}c\sqrt{1-c^2}\left(7+2c^2\right) \tag{B.7}$$

$$T_8 = -\tfrac{1}{3}\sqrt{1-c^2}\left(2c^2+1\right)+c\cos^{-1}\left(c\right) \tag{B.8}$$

$$T_9 = \tfrac{1}{2}\left[\tfrac{1}{3}\left(1-c^2\right)^{\frac{3}{2}}+aT_4\right]=\tfrac{1}{2}\left[\tfrac{1}{3}\sqrt{\left(1-c^2\right)^3}+aT_4\right] \tag{B.9}$$

$$T_{10} = \sqrt{1-c^2}+\cos^{-1}\left(c\right) \tag{B.10}$$

$$T_{11} = \cos^{-1}\left(c\right)\left(1-2c\right)+\sqrt{1-c^2}\left(2-c\right) \tag{B.11}$$

$$T_{12} = \sqrt{1-c^2}\left(2+c\right)-\cos^{-1}\left(c\right)\left(1+2c\right) \tag{B.12}$$

$$T_{13} = \tfrac{1}{2}\left[-T_7-\left(c-a\right)T_1\right] \tag{B.13}$$

$$T_{14} = \tfrac{1}{16}+\tfrac{1}{2}ac \tag{B.14}$$

Appendix C Flutter and Controller Analysis Software

The MATLAB™ software developed to analyse and simulate the open- and closed-loop dynamics of the FAST model is listed here. Functions of the main program are briefly described in the order in which they are called by it, after which a listing of each is provided. The various functions are:

- *FAST_Model_Analysis.m:* The main open- and closed-loop binary flutter analysis program from which all other functions are called
- *Aerofoil_Data.m:* A MATLAB™ file containing the aerofoil parameters. When called, this function saves aerofoil parameters to a '~.mat' file that is subsequently read by all other functions in the program
- *Modal_Data.m:* A routine to solve the eigenvalue flutter solution and calculate the modal dampings and frequencies of the flutter system at a given airspeed and air density
- *State_Space_Matrices.m:* A function that sets up the state-space matrices of the binary flutter system as a function of airspeed and air density
- *T_Functions.m:* A function to calculate Theodorsen's functions (required in the state-space matrices of the flutter system) from aerofoil parameters based on the control surface parameters listed in "*Aerofoil_Data.m*"
- *Sort_Eigenvalues.m:* A routine to sort the eigenvalues calculated in "*Modal_Data.m*". This is required so that the real (modal damping) and imaginary (modal frequency) parts of the eigenvalues can be arranged according to airspeed to allow the theoretical flutter speed of the system to be calculated
- *Flutter_Speed.m:* A function to calculate the flutter speed of a given model configuration by linearly interpolating between a point of positive and negative damping from vectors of airspeed, modal damping and modal frequencies
- *Plot_Modal_Data.m:* A function that plots the pitch, plunge and control surface modal damping and frequency as a function of airspeed
- *Time_Series.m:* A function that calculates the time dependant solution of the state-space matrices of a flutter system
- *Solve_ODE.m:* A sub-routine of the "*Time_Series.m*" function, used to solve the ordinary differential flutter equation
- *Plot_Time_Series.m:* A routine that plots the plunge, pitch and control surface response of the binary flutter system as a function of time

• *LQG_Control.m:*	The controller function of the main flutter analysis software that calculates the LQR and Kalman gains and the closed-loop response of the binary flutter system

Additional MATLAB™ functions developed to process and present results are also described here. Functions were written to generate input data for the Simulink™ model (Appendix D) of the system, and to post-process open- and closed-loop simulation and wind tunnel data. This code calls other functions that search for peaks within PSDs of signals, calculate PSD peak widths at a given amplitude and plot all signal processing data. These functions are:

• *FAST_Model_Data.m:*	The function used to specify constants required in the FAST Simulink™ model
• *Analyse_Data.m:*	The main program used for post-processing of FAST model simulation and wind tunnel data
• *Signal_Processing.m:*	A function used to calculate the FFT and PSD of a given signal after applying zero padding and an appropriate window function
• *Find_Peaks.m:*	A function that searches for a specified number of peaks in either the FFT or PSD of a given signal and returns the peak frequency and amplitude
• *Peak_Width.m:*	A function that calculates the width of a PSD peak at half its height. This was required for the half power bandwidth method (Section 2.4.2) of evaluating the subcritical response of the FAST model
• *Plot_FFT.m:*	The function to plot the FFT and PSD data of a given signal
• *Plot_Peak_Width.m:*	The function to plot the calculated PSD peak widths over a plot of the calculated PSD of a signal

```
%=========================================================================%
%================== FAST MODEL AEROSERVOELASTIC ANALYSIS =================%
%=========================================================================%

clear all
close all
clc

%-------------------------------------------------------------------------%
%------------------------------- Input Data ------------------------------%
%-------------------------------------------------------------------------%

rho = 1.0062;               % Ambient air density [kg/m^3]
U_ol = linspace(0, 25, 50); % Airspeed range to analyse [m/s]
U_cl = linspace(0, 80, 50); % Airspeed range to analyse [m/s]
T_Samples = 500;            % Number of time steps in flutter analysis
T_end = 10;                 % Time period [s]

%-------------------------------------------------------------------------%
%-------------------- Generate and Save Aerofoil Data --------------------%
%-------------------------------------------------------------------------%

% To change aerofoil parameters (mass, mass inertia, cg position etc.),
% edit the MATLAB script: 'Aerofoil_Data.m'.

Aerofoil_Data;              % Save aerofoil parameters to MATLAB ~.mat file
load('Aerofoil_Data');      % Load aerofoil parameters from ~.mat file

%-------------------------------------------------------------------------%
%----------------- Open-Loop Modal Frequencies and Damping ---------------%
%-------------------------------------------------------------------------%

[U_f_ol, f_f_ol, f_ol, g_ol] = Modal_Data(U_ol, rho, zeros(1, 8));
U_f_ol = U_f_ol(1); f_f_ol = f_f_ol(1);

% U_f_ol:  Open-loop flutter speed [m/s]
% f_f_ol:  Open-loop flutter frequency [Hz.]
% f_ol:    Open-loop modal frequencies as a function of airspeed
% g_ol:    Open-loop modal damping as a function of airspeed

figure, plot_Modal_Data(f_ol, g_ol, U_ol, U_f_ol, f_f_ol)

%-------------------------------------------------------------------------%
%---------------------- Open-Loop Time Series Analysis -------------------%
%-------------------------------------------------------------------------%

% Time vector to plot modal response over:
T_Series = linspace(0, T_end, T_Samples);

X_0 = [0;                   % Initial plunge rate
       0;                   % Initial pitch rate
       0;                   % Initial control surface rate
       -7.5/1000;           % Initial plunge displacement [m]
       5/(180/pi);          % Initial pitch displacement [rad.]
       0;                   % Initial control surface angle [rad.]
       0;                   % Initial aerodynamic lag (1st state)
       0]';                 % Initial aerodynamic lag (2nd state)

[t_ol, X_ol] = Time_Series(U_f_ol, rho, X_0, zeros(1, 8), T_Series);
figure, plot_Time_Series(t_ol, X_ol, [])

%-------------------------------------------------------------------------%
%---------------- Closed-Loop Modal and Time Series Analysis -------------%
%-------------------------------------------------------------------------%

% Initial Kalman states:
X_e_0 = [0 0 0 0 0 0 0 0];

[K_LQR, L_Kalman, t_cl, X_cl, X_hat, sigma_m] = LQG_Control(T_Series,...
                                                X_0, X_e_0, U_f_ol, rho);

[U_f_cl, f_f_cl, f_cl, g_cl] = Modal_Data(U_cl, rho, K_LQR);
```

```
U_f_cl = U_f_cl(1); f_f_cl = f_f_cl(1);

% U_f_cl:   Closed-loop flutter speed [m/s]
% f_f_cl:   Closed-loop flutter frequency [Hz.]
% f_cl:     Closed-loop modal frequencies as a function of airspeed
% g_cl:     Closed-loop modal damping as a function of airspeed

figure, plot_Modal_Data(f_cl, g_cl, U_cl, U_f_cl, f_f_cl)
figure, plot_Time_Series(t_cl, X_cl, X_hat)

%-------------------------------------------------------------------------%
%------------- Generate and Save Parameters for Simulink Model -----------%
%-------------------------------------------------------------------------%

FAST_Model_Data(U_f_ol, rho, K_LQR, L_Kalman, X_0, X_e_0, sigma_m);
load('Flutter_Model_Data')

%=========================================================================%
```

```
%=========================================================================%
function Aerofoil_Data
%=========================================================================%

span = 0.335;           % Wingspan [m]

a = -0.2;               % Distance between mid-chord and ea in semi-chords
b = 0.06;               % Reference semi-chord [m]
c = 0.5;                % Flap position in semichords
x_a = 0.2;              % Distance between aerofoil ea and cg in
                        % cg in semi-chords
x_B = -0.075;           % Distance between control surface hinge axis and
                        % cg in semi-chords
I_a = 30e-3/span;       % Mass inertia of wing about ea [kgm^2]
I_B = 45e-6/span;       % Mass inertia of flap about hinge axis [kgm^2]
m = 1.85/span;          % Mass of wing [kg]

f_nh = 5.6651;          % Uncoupled plunge frequency [Hz.]
f_na = 6.2688;          % Uncoupled pitch frequency [Hz.]
f_nB = 100;             % Uncoupled control surface frequency [Hz.]
w_nh = 2*pi*f_nh;       % Uncoupled plunge frequency [rad/s]
w_na = 2*pi*f_na;       % Uncoupled pitch frequency [rad/s]
w_nB = 2*pi*f_nB;       % Uncoupled control surface frequency [rad/s]
c_h = 0.025;            % Plunge damping coefficient [Ns/m]
c_a = 0.05;             % Pitch damping coefficient [Ns/m]
c_B = 0.05;             % Control surface damping coefficient [Ns/m]

k_h = (w_nh^2)*m;       % Plunge stiffness coefficient [N/m]
k_a = (w_na^2)*I_a;     % Pitch stiffness coefficient [N/m]
k_B = (w_nB^2);         % Pitch stiffness coefficient [N/m]

save('Aerofoil_Data')

%=========================================================================%
```

```
%===========================================================================%
function [U_f, f_f, f_s, g_s] = Modal_Data(U, rho, K)
%===========================================================================%

load('Aerofoil_Data');        % Load aerofoil parameters from ~.mat file

%---------------------------------------------------------------------------%
%--------------------- Calculate System Eigenvalues ------------------------%
%---------------------------------------------------------------------------%

i_end = length(U);
lambda = zeros(i_end, 8);

for i = 1: i_end

    [A, B, C, D] = State_Space_Matrices(U(i), rho);
    lambda(i, :) = eig(A - B*K);

end

lambda = Sort_Eigenvalues(lambda);

g = real(lambda);
w = imag(lambda);
f = w/(2*pi);

[U_f, f_f] = Flutter_Speed(U, g, f);

%---------------------------------------------------------------------------%
%---------- Extract Plunge, Pitch and Control Surface Angle Data ---------%
%---------------------------------------------------------------------------%

i = 0;
k_end = length(lambda(1, :));

for k =1: k_end

    for j = k: k_end - 1

        if f(end, k) == -f(end, j + 1) && f(end, k) ~= 0

            i = i + 1;

            if sign(w(end, k)) == 1

                g_r(:, i) = g(:, k);
                f_r(:, i) = f(:, k);

            else

                g_r(:, i) = g(:, j + 1);
                f_r(:, i) = f(:, j + 1);

            end

            break

        else

            break

        end

    end

end

while i < 3 && k == k_end

    for j = 1: k_end

        if sign(g(end, j)) == 1 && f(end, j) == 0
```

```
            i = i + 1;
            g_r(:, i) = g(:, j);
            f_r(:, i) = f(:, j);

        end

    end

end

%-------------------------------------------------------------------------%
%----------- Sort Plunge, Pitch and Control Surface Angle Data -----------%
%-------------------------------------------------------------------------%

f_rs = [f_r(1, 1) f_r(1, 2) f_r(1, 3)];
f_n = [f_nh f_na f_nB];
s_matrix = zeros(3, 3);
f_haB = zeros(length(f_r(:, 1)), 3);
g_haB = zeros(length(g_r(:, 1)), 3);

for i = 1: 3

    for j = 1: 3

        s_matrix(j, i) = abs(f_rs(i) - f_n(j));

    end

end

[s_vector, r_index] = min(s_matrix);
[ignore, c_index] = sort(s_vector);

for k = 1: 3

    f_haB(:, k) = f_r(:, c_index(k));
    g_haB(:, k) = g_r(:, c_index(k));

end

%-------------------------------------------------------------------------%
%----------- Save Plunge, Pitch and Control Surface Angle Data -----------%
%-------------------------------------------------------------------------%

f_s.h = f_haB(:, 1); g_s.h = g_haB(:, 1);
f_s.a = f_haB(:, 2); g_s.a = g_haB(:, 2);
f_s.B = f_haB(:, 3); g_s.B = g_haB(:, 3);

%=========================================================================%
```

```
%=========================================================================%
function [A, B, C, D] = State_Space_Matrices(U, rho)
%=========================================================================%

load('Aerofoil_Data')

%-------------------------------------------------------------------------%
%------------------------ Theodorsen's Coefficients ----------------------%
%-------------------------------------------------------------------------%

[p, T] = T_Functions;

%-------------------------------------------------------------------------%
%------------------------ W.P. Jones' Approximation ----------------------%
%-------------------------------------------------------------------------%

delta_1 = 0.165;
delta_2 = 0.335;
lambda_1 = 0.041;
lambda_2 = 0.320;

%-------------------------------------------------------------------------%
%-------------------------- Structural Matrices --------------------------%
%-------------------------------------------------------------------------%

M_s = [m m*x_a*b m*x_B*b;
       m*x_a*b I_a (c - a)*(b^2)*m*x_B + I_B;
       m*x_B*b (c - a)*(b^2)*m*x_B + I_B I_B];

C_s = [c_h 0 0; 0 c_a 0; 0 0 c_B];

K_s = [k_h 0 0; 0 k_a 0; 0 0 k_B];

%-------------------------------------------------------------------------%
%-------------------------- Aerodynamic Matrices -------------------------%
%-------------------------------------------------------------------------%

M_a = [-pi*rho*(b^2) pi*rho*(b^3)*a rho*(b^3)*T(1);
       pi*rho*(b^3)*a -pi*rho*(b^4)*((1/8) + a^2) rho*(b^4)*(T(7) +...
       (c - a)*T(1));
       rho*(b^3)*T(1) -2*rho*(b^4)*T(13) rho*(b^4)*(T(3)/pi)];

C_a = [-2*pi*rho*b*U -2*pi*rho*(b^2)*U*(1 - a) rho*(b^2)*U*(T(4) - T(11));
       2*pi*rho*(b^2)*U*(a + (1/2)) pi*rho*(b^3)*U*a*(1 - 2*a) rho*...
       (b^3)*U*(T(8) - T(1) + (c - a)*T(4) + a*T(11));
       -rho*(b^2)*U*T(12) rho*(b^3)*U*(2*T(9) + T(1) + (T(4) - T(12))*...
       ((1/2) - a)) rho*(b^3)*U*(T(11)/(2*pi))*(T(4) - T(12))];

K_a = [0 -2*pi*rho*b*(U^2) -2*rho*b*(U^2)*T(10);
       0 2*pi*rho*(b^2)*(U^2)*((1/2) + a) rho*(b^2)*(U^2)*(2*a*T(10) -...
       T(4));
       0 -rho*(b^2)*(U^2)*T(12) -(1/pi)*rho*(b^2)*(U^2)*(T(5) - T(10)*...
       (T(4) - T(12)))];

L_delta = [2*pi*rho*U*b*delta_1 2*pi*rho*U*b*delta_2;
           -2*pi*rho*U*(b^2)*(a + (1/2))*delta_1 -2*pi*rho*(b^2)*U*(a +...
           (1/2))*delta_2;
           rho*U*(b^2)*T(12)*delta_1 rho*U*(b^2)*T(12)*delta_2];

%-------------------------------------------------------------------------%
%------------------------ Aerodynamic Lag Matrices -----------------------%
%-------------------------------------------------------------------------%

L_lambda = [(-lambda_1*U)/b 0; 0 (-lambda_2*U)/b];

Q_a = [1 b*((1/2) - a) (T(11)*b)/(2*pi)];
Q_v = [0 U (T(10)*U)/pi];

%-------------------------------------------------------------------------%
%----------------------------- System Matrix -----------------------------%
%-------------------------------------------------------------------------%
```

```
A_11 = -inv(M_s - M_a)*(C_s - C_a);
A_12 = -inv(M_s - M_a)*(K_s - K_a);
A_13 = inv(M_s - M_a)*L_delta;

A_21 = eye(3, 3);
A_22 = zeros(3, 3);
A_23 = zeros(3, 2);

A_31 = [Q_a*A_11 + Q_v; Q_a*A_11 + Q_v];
A_32 = [Q_a*A_12; Q_a*A_12];
A_33 = [Q_a*A_13; Q_a*A_13] + L_lambda;

A = [A_11 A_12 A_13; A_21 A_22 A_23; A_31 A_32 A_33];

%------------------------------------------------------------------------%
%------------------------------ Input Matrix ----------------------------%
%------------------------------------------------------------------------%

B = [inv(M_s - M_a)*[0; 0; k_B]; 0; 0; 0; 0; 0];

%------------------------------------------------------------------------%
%------------------------------ Output Matrix ---------------------------%
%------------------------------------------------------------------------%

C = zeros(3, 8);
C(1, 4) = 1;                    % Plunge displacement measurable
C(2, 5) = 1;                    % Pitch angle measurable
C(3, 8) = 1;                    % Control surface angle measurable

%------------------------------------------------------------------------%
%-------------------- Input-Output Transmission Matrix ------------------%
%------------------------------------------------------------------------%

D = zeros(3, 1);

%========================================================================%
```

```
%=========================================================================%
function [p, T] = T_Functions
%=========================================================================%

load('Aerofoil_Data')

%-------------------------------------------------------------------------%
%------------------------ Theodorsen's Coefficients ----------------------%
%-------------------------------------------------------------------------%

p = -(1/3)*(sqrt(1 - c^2))^3;

T_1 = -(1/3)*sqrt(1 - c^2)*(2 + c^2) + c*acos(c);
T_2 = c*(1 - c^2) - sqrt(1 - c^2)*(1 + c^2)*acos(c) + c*(acos(c))^2;
T_3 = -((1/8) + c^2)*(acos(c))^2 + (1/4)*c*sqrt(1 - c^2)*acos(c)*...
      (7 + 2*c^2) - (1/8)*(1 - c^2)*(5*c^2 + 4);
T_4 = -acos(c) + c*sqrt(1 - c^2);
T_5 = -(1 - c^2) - (acos(c))^2 + 2*c*sqrt(1 - c^2)*acos(c);
T_6 = T_2;
T_7 = -((1/8) + c^2)*acos(c) + (1/8)*c*sqrt(1 - c^2)*(7 + 2*c^2);
T_8 = -(1/3)*sqrt(1 - c^2)*(2*c^2 + 1) + c*acos(c);
T_9 = (1/2)*(-p + a*T_4);
T_10 = sqrt(1 - c^2) + acos(c);
T_11 = acos(c)*(1 - 2*c) + sqrt(1 - c^2)*(2 - c);
T_12 = sqrt(1 - c^2)*(2 + c) - acos(c)*(2*c + 1);
T_13 = (1/2)*(-T_7 - (c - a)*T_1);
T_14 = (1/16) + (1/2)*a*c;
T = [T_1 T_2 T_3 T_4 T_5 T_6 T_7 T_8 T_9 T_10 T_11 T_12 T_13 T_14];

save('T_Functions')

%=========================================================================%
```

```
%=========================================================================%
function lambda_s = Sort_Eigenvalues(lambda)
%=========================================================================%

lambda_t = lambda;
[r_end, j_end] = size(lambda_t);

for r = 1: r_end - 1

    e = zeros(j_end, j_end);

    for i = 1: j_end

        for j = 1: j_end

            e(i, j) = abs(lambda_t(r, i) - lambda_t(r + 1, j));

        end

    end

    s_flag = 0;
    m_flag = 1;

    while s_flag < j_end

        min_ij = min(min(e));

        if m_flag == 1

            max_ij = max(max(e));
            m_flag = 0;

        end

        [i_min, j_min] = find(e == min_ij);
        i_min = i_min(1);
        j_min = j_min(1);
        lambda_t((r + 1), i_min) = lambda(r + 1, j_min);
        e(i_min, :) = 10*max_ij;
        e(:, j_min) = 10*max_ij;

        s_flag = s_flag + 1;

    end

end

lambda_s = lambda_t;

%=========================================================================%
```

```
%==========================================================================%
function [U_f, w_f] = Flutter_Speed(U, g, w)
%==========================================================================%

i = 1;
U_f_flag = 0;

for r = 1:length(U) - 1

    for c = 1: length(g(1, :))

        if g(r, c) <= 0 & g((r + 1), c) >= 0

            g_i = [g(r, c), g((r + 1), c)];
            w_i = [w(r, c), w((r + 1), c)];
            U_i = [U(r), U(r + 1)];

            U_f_us(i) = interp1(g_i, U_i, 0);
            w_f_us(i) = interp1(U_i, w_i, U_f_us(i));

            i = i + 1;
            U_f_flag = 1;

        end

    end

end

if U_f_flag == 0

    U_f = 0;
    w_f = 0;

else

    [U_f, I] = sort(U_f_us, 'ascend');

    for j = 1: length(I)

        w_f(j) = w_f_us(I(j));

    end

end

%==========================================================================%
```

```
%=========================================================================%
function plot_Modal_Data(f, g, U, U_f, f_f)
%=========================================================================%

subplot(2,2,[1 3]), plot(U, f.h, 'b', 'LineWidth', 1.5), hold on
subplot(2,2,[1 3]), plot(U, f.a, 'g', 'LineWidth', 1.5), grid on
subplot(2,2,[1 3]), plot(U, f.B, 'k', 'LineWidth', 1.5)
subplot(2,2,[1 3]), plot(U_f, f_f, 'or', 'LineWidth’, 1.5, 'MarkerSize', 6)

title('\bfModal Frequencies', 'FontName', 'Calibri', 'FontSize', 14)
xlabel('Airspeed [\itm/s\rm]', 'FontName', 'Calibri', 'FontSize', 12)
ylabel('Frequency [\itHz.\rm]', 'FontName', 'Calibri', 'FontSize', 12)
set(gca, 'FontName', 'Calibri', 'FontSize', 10)

legend('Plunge Mode',...
       'Pitch Mode',...
       'Control Surface Mode',...
       ['Flutter Frequency [', num2str(f_f, '%.2f'), ' \itHz.\rm]'], 1)

%-------------------------------------------------------------------------%

subplot(2,2,[2 4]), plot(U, -g.h, 'b', 'LineWidth', 1.5), hold on
subplot(2,2,[2 4]), plot(U, -g.a, 'g', 'LineWidth', 1.5), grid on
subplot(2,2,[2 4]), plot(U, -g.B, 'k', 'LineWidth', 1.5)
subplot(2,2,[2 4]), plot(U_f, 0, 'ok', 'LineWidth’, 1.5, 'MarkerSize', 6)

title('\bfModal Damping', 'FontName', 'Calibri', 'FontSize', 14)
xlabel('Airspeed [\itm/s\rm]', 'FontName', 'Calibri', 'FontSize', 12)
ylabel('Damping [\itNs/m\rm]', 'FontName', 'Calibri', 'FontSize', 12)
set(gca, 'FontName', 'Calibri', 'FontSize', 10)

legend('Plunge Mode',...
       'Pitch Mode',...
       'Control Surface Mode',...
       ['Flutter Velocity (', num2str(U_f, '%.2f'), ' \itm/s\rm)'], 2);

%=========================================================================%
```

```
%=========================================================================%
function [t, X] = Time_Series(U, rho, X_0, K, T_Series)
%=========================================================================%

global A_BK

%-------------------------------------------------------------------------%
%----------------------------- Additional Data ---------------------------%
%-------------------------------------------------------------------------%

[A, B, C, D] = State_Space_Matrices(U, rho);
A_BK = (A - B*K);

%-------------------------------------------------------------------------%
%-------------------------- Differential Equation ------------------------%
%-------------------------------------------------------------------------%

options = odeset('Refine', 10, 'RelTol', 0.00005, 'MaxStep',...
                 0.005*abs(T_Series(end) - T_Series(1)));

[t, X_i] = ode113('Solve_ODE', T_Series, X_0, options);

%-------------------------------------------------------------------------%
%------------------------------ System States ----------------------------%
%-------------------------------------------------------------------------%

X.h_dot = X_i(:, 1)';
X.a_dot = X_i(:, 2)';
X.B_dot = X_i(:, 3)';
X.h = X_i(:, 4)';
X.a = X_i(:, 5)';
X.B = X_i(:, 6)';
X.l_1 = X_i(:, 7)';
X.l_2 = X_i(:, 8)';

X.h_dot_dot = A(1, 1)*X.h_dot + A(1, 2)*X.a_dot + A(1, 3)*X.B_dot +...
              A(1, 4)*X.h + A(1, 5)*X.a + A(1, 6)*X.B + A(1, 7)*X.l_1 +...
              A(1, 8)*X.l_2;
X.a_dot_dot = A(2, 1)*X.h_dot + A(2, 2)*X.a_dot + A(2, 3)*X.B_dot +...
              A(2, 4)*X.h + A(2, 5)*X.a + A(2, 6)*X.B + A(2, 7)*X.l_1 +...
              A(2, 8)*X.l_2;
X.B_dot_dot = A(3, 1)*X.h_dot + A(3, 2)*X.a_dot + A(3, 3)*X.B_dot +...
              A(3, 4)*X.h + A(3, 5)*X.a + A(3, 6)*X.B + A(3, 7)*X.l_1 +...
              A(3, 8)*X.l_2;

%=========================================================================%
```

```
%=========================================================================%
function dX_dt = Solve_ODE(T, X_0)
%=========================================================================%

global A_BK

dX_dt = A_BK*X_0;

%=========================================================================%
```

```
%=========================================================================%
function plot_Time_Series(t, X, X_hat)
%=========================================================================%

if ~isempty(X_hat)

    subplot(3,1,1), plot(t, 1000*X_hat.h, '--r', 'LineWidth', 1.5)
    hold on
    subplot(3,1,2), plot(t, (180/pi)*X_hat.a, '--r', 'LineWidth', 1.5)
    hold on
    subplot(3,1,3), plot(t, (180/pi)*X_hat.B, '--r', 'LineWidth', 1.5)
    hold on

end

%-------------------------------------------------------------------------%

subplot(3,1,1), plot(t, 1000*X.h, 'b', 'LineWidth', 1.5)
grid on

title('\bfPlunge Response', 'FontName', 'Calibri', 'FontSize', 14)
xlabel('Time [\its\rm]', 'FontName', 'Calibri', 'FontSize', 12)
ylabel('Plunge Displacement [\itmm\rm]', 'FontName', 'Calibri',...
       'FontSize', 12)
set(gca, 'FontName', 'Calibri', 'FontSize', 10)

%-------------------------------------------------------------------------%

subplot(3,1,2), plot(t, (180/pi)*X.a, 'g', 'LineWidth', 1.5)
grid on

title('\bfPitch Response', 'FontName', 'Calibri', 'FontSize', 14)
xlabel('Time [\its\rm]', 'FontName', 'Calibri', 'FontSize', 12)
ylabel('Pitch Angle [\itdeg.\rm]', 'FontName', 'Calibri', 'FontSize', 12)
set(gca, 'FontName', 'Calibri', 'FontSize', 10)

%-------------------------------------------------------------------------%

subplot(3,1,3), plot(t, (180/pi)*X.B, 'k', 'LineWidth', 1.5)
grid on

title('\bfControl Surface Response', 'FontName', 'Calibri', 'FontSize', 14)
xlabel ('Time [\its\rm]', 'FontName', 'Calibri', 'FontSize', 12)
ylabel ('Control Surface Angle [\itdeg.\rm]', 'FontName', 'Calibri',...
        'FontSize', 12)
set(gca, 'FontName', 'Calibri', 'FontSize', 10)

%=========================================================================%
```

```
%==========================================================================%
function [K_LQR, L_Kalman, t_cl, X_cl, X_hat, sigma_m] =...
          LQG_Control(T, X_0, X_e_0, U_f, rho)
%==========================================================================%

load('Aerofoil_Data');

%--------------------------------------------------------------------------%
%------------------------------ Sample Settings ---------------------------%
%--------------------------------------------------------------------------%

f_sample = 1495;
T_sample = 1/f_sample;
t_period = T(end);
delta_t = 1/f_sample;

%--------------------------------------------------------------------------%
%------------------------ System Model Construction -----------------------%
%--------------------------------------------------------------------------%

[A, B, C, D] = State_Space_Matrices(U_f, rho);

% Continuous state-space model
FlutterModel = ss(A, B, C, D);
% Discrete state-space model
FlutterModel_d = c2d(FlutterModel, T_sample, 'zoh');
[A_d, B_d, C_d, D_d] = ssdata(FlutterModel_d);

%--------------------------------------------------------------------------%
%-------------------------- Measurement Noise Data ------------------------%
%--------------------------------------------------------------------------%

sigma_h = 0.15/1000;
% Plunge displacement measurement variance of +/- 0.15 mm
sigma_alpha = 0.25*(pi/180);
% Pitch angle measurement variance of +/- 0.25 deg.
sigma_beta = 0.30*(pi/180);
% Control surface measurement variance of +/- 0.30 deg.

sigma_m = [sigma_h 0 0; 0 sigma_alpha 0; 0 0 sigma_beta];

%--------------------------------------------------------------------------%
%-------------------- Controllability and Observability -------------------%
%--------------------------------------------------------------------------%

P = rank(ctrb(FlutterModel_d));          % Controllability matrix
N = rank(obsv(FlutterModel_d));          % Observability matrix

%--------------------------------------------------------------------------%
%------------------------------ LQR Controller ----------------------------%
%--------------------------------------------------------------------------%

Q_LQR = zeros(8, 8);
Q_LQR(4, 4) = 150;
Q_LQR(5, 5) = 150;
Q_LQR(6, 6) = 5;
R_LQR = 1;

[K_LQR, S_LQR, E_LQR] = dlqr(A_d, B_d, Q_LQR, R_LQR);

%--------------------------------------------------------------------------%
%------------------------------- Kalman Filter ----------------------------%
%--------------------------------------------------------------------------%

G_Kalman = eye(8);
Q_Kalman = [(0.05)^2 0 0 0 0 0 0 0;
            0 (0.5e-5)^2 0 0 0 0 0 0;
            0 0 (0.5e-5)^2 0 0 0 0 0;
            0 0 0 (1e-3)^2 0 0 0 0;
            0 0 0 0 (0.005)^2 0 0 0;
            0 0 0 0 0 (0.005)^2 0 0;
            0 0 0 0 0 0 (0.02)^2 0;
```

```
            0 0 0 0 0 0 0 (0.05)^2];
R_Kalman = (C*C')*sigma_m^2;

L_Kalman = dlqe(A_d, G_Kalman, C_d, Q_Kalman, R_Kalman);

%-------------------------------------------------------------------------%
%----------------------------- LQG Controller ----------------------------%
%-------------------------------------------------------------------------%

LQGCompensator = reg(FlutterModel_d, K_LQR, L_Kalman);
FlutterModel_CL = feedback(FlutterModel_d, LQGCompensator, 1);

[Y, t_cl, X] = initial(FlutterModel_CL, [X_0 X_e_0],...
                      [0: delta_t: t_period]);

%-------------------------------------------------------------------------%
%------------------- System States and Kalman Estimates ------------------%
%-------------------------------------------------------------------------%

X_cl.h_dot = X(:, 1)';
X_cl.a_dot = X(:, 2)';
X_cl.B_dot = X(:, 3)';
X_cl.h = Y(:, 1)';
X_cl.a = Y(:, 2)';
X_cl.B = Y(:, 3)';
X_cl.l_1 = X(:, 7)';
X_cl.l_2 = X(:, 8)';

X_hat.h_dot = X(:, 9)';
X_hat.a_dot = X(:, 10)';
X_hat.B_dot = X(:, 11)';
X_hat.h = X(:, 12)';
X_hat.a = X(:, 13)';
X_hat.B = X(:, 14)';
X_hat.l_1 = X(:, 15)';
X_hat.l_2 = X(:, 16)';

%=========================================================================%
```

```
%==========================================================================%
function FAST_Model_Data(U, rho, K_LQR, L_Kalman, X_0, X_0_e, sigma_m)
%==========================================================================%
%
% Data for 'FAST_Model.mdl'
%
%--------------------------------------------------------------------------%
%------------------------------ Aerofoil Data -----------------------------%
%--------------------------------------------------------------------------%

load('Aerofoil_Data')

%--------------------------------------------------------------------------%
%------------------------ W.P. Jones' Approximation -----------------------%
%--------------------------------------------------------------------------%

delta_1 = 0.165;
delta_2 = 0.335;
lambda_1 = 0.041;
lambda_2 = 0.320;

%--------------------------------------------------------------------------%
%------------------------ Theodorsen's Coefficients -----------------------%
%--------------------------------------------------------------------------%

p = -(1/3)*(sqrt(1 - c^2))^3;

T_1 = -(1/3)*sqrt(1 - c^2)*(2 + c^2) + c*acos(c);
T_2 = c*(1 - c^2) - sqrt(1 - c^2)*(1 + c^2)*acos(c) + c*(acos(c))^2;
T_3 = -((1/8) + c^2)*(acos(c))^2 + (1/4)*c*sqrt(1 - c^2)*acos(c)*...
       (7 + 2*c^2) - (1/8)*(1 - c^2)*(5*c^2 + 4);
T_4 = -acos(c) + c*sqrt(1 - c^2);
T_5 = -(1 - c^2) - (acos(c))^2 + 2*c*sqrt(1 - c^2)*acos(c);
T_6 = T_2;
T_7 = -((1/8) + c^2)*acos(c) + (1/8)*c*sqrt(1 - c^2)*(7 + 2*c^2);
T_8 = -(1/3)*sqrt(1 - c^2)*(2*c^2 + 1) + c*acos(c);
T_9 = (1/2)*(-p + a*T_4);
T_10 = sqrt(1 - c^2) + acos(c);
T_11 = acos(c)*(1 - 2*c) + sqrt(1 - c^2)*(2 - c);
T_12 = sqrt(1 - c^2)*(2 + c) - acos(c)*(2*c + 1);
T_13 = (1/2)*(-T_7 - (c - a)*T_1);
T_14 = (1/16) + (1/2)*a*c;

%--------------------------------------------------------------------------%
%--------------------------- State-Space Matricies ------------------------%
%--------------------------------------------------------------------------%

[A, B, C, D] = State_Space_Matrices(U, rho);

%--------------------------------------------------------------------------%
%-------------------- Initial States and Controller Gains -----------------%
%--------------------------------------------------------------------------%

% Put inputs into the MATLAB workspace:

X_0 = X_0;
X_0_e = X_0_e;
K_LQR = K_LQR;
L_Kalman = L_Kalman;
sigma_m = sigma_m;

%--------------------------------------------------------------------------%
%---------------------------------- Save Data -----------------------------%
%--------------------------------------------------------------------------%

save('Flutter_Model_Data')

%==========================================================================%
```

```
%==========================================================================%
%====================== ANALYSIS OF FAST MODEL DATA =======================%
%==========================================================================%

clear all
close all
clc

%--------------------------------------------------------------------------%
%------------------------ Plot Modal Time Responses -----------------------%
%--------------------------------------------------------------------------%

Path = 'C:\FAST Model Wind Tunnel Data';

[File, Path] = uigetfile('*.csv', 'Open', Path);
DataFile = [Path File];
FlutterData = csvread(DataFile, 11, 0);
T_End = csvread(DataFile, 8, 0, [8, 0, 8, 0]);
DataPoints = length(FlutterData(:, 1));
T = linspace(0, T_End, DataPoints)';
U = str2double(File(9:10)) + str2double(File(12:13))/100;

%--------------------------------------------------------------------------%
%------------------------ Plot Modal Time Responses -----------------------%
%--------------------------------------------------------------------------%

h = FlutterData(:, 3)';
h = smooth(h, 'sgolay')';
h_hat = FlutterData(:, 10)';
h_hat = smooth(h_hat, 'sgolay')';

alpha = FlutterData(:, 4)';
alpha = smooth(alpha, 'sgolay')';
alpha_hat = FlutterData(:, 11)';
alpha_hat = smooth(alpha_hat, 'sgolay')';

beta = FlutterData(:, 5)';
beta = smooth(beta, 'sgolay')';
beta_hat = FlutterData(:, 12)';
beta_hat = smooth(beta_hat, 'sgolay')';
u = FlutterData(:, 6)';
u = smooth(u, 'sgolay')';

%--------------------------------------------------------------------------%

f_range = [5 7];
N_Peaks = 2;

[f_s(1), f_h, h_FFT, h_PSD] = Signal_Processing(T, h);
[f_s(2), f_a, a_FFT, a_PSD] = Signal_Processing(T, alpha);

f_h_int = cumtrapz(f_h, h_PSD); f_h_int = f_h_int(end);
f_a_int = cumtrapz(f_a, a_PSD); f_a_int = f_a_int(end);
h_PSD = h_PSD/f_h_int;
a_PSD = a_PSD/f_a_int;

h_PSD_Peaks = Find_Peaks(f_h, h_PSD, N_Peaks, 'curve fit on');
a_PSD_Peaks = Find_Peaks(f_a, a_PSD, N_Peaks, 'curve fit on');
[h_Width, h_Height] = Peak_Width(h_PSD_Peaks, h_PSD, f_h);
[a_Width, a_Height] = Peak_Width(a_PSD_Peaks, a_PSD, f_a);

%--------------------------------------------------------------------------%
%------------------------ Plot Modal Time Responses -----------------------%
%--------------------------------------------------------------------------%

subplot(3, 1, 1), plot(T, 1000*h, '-b', 'LineWidth', 1.5); hold on
subplot(3, 1, 1), plot(T, 1000*h_hat, '--r', 'LineWidth', 1.5);

grid on
title('\bfPlunge Response', 'FontName', 'Calibri', 'FontSize', 14)
xlabel('Time [\its\rm]', 'FontName', 'Calibri', 'FontSize', 12)
ylabel('Plunge Displacement [\itmm\rm]', 'FontName', 'Calibri',...
```

```
       'FontSize', 12)
set(gca, 'FontName', 'Calibri', 'FontSize', 10)
legend('Plunge SGB Signal', 'Kalman Estimate');

%-----------------------------------------------------------------------%

subplot(3, 1, 2), plot(T, (180/pi)*alpha, '-g', 'LineWidth', 1.5); hold on
subplot(3, 1, 2), plot(T, (180/pi)*alpha_hat, '--r', 'LineWidth', 1.5);

grid on
title('\bfPitch Response', 'FontName', 'Calibri', 'FontSize', 14)
xlabel('Time [\its\rm]', 'FontName', 'Calibri', 'FontSize', 12)
ylabel('Pitch Angle [\itdeg.\rm]', 'FontName', 'Calibri', 'FontSize', 12)
set(gca, 'FontName', 'Calibri', 'FontSize', 10)
legend('Pitch SGB Signal', 'Kalman Estimate');

%-----------------------------------------------------------------------%

subplot(3, 1, 3), plot(T, (180/pi)*beta, 'k', 'LineWidth', 1.5); hold on
subplot(3, 1, 3), plot(T, (180/pi)*beta_hat, '--r', 'LineWidth', 1.5);
subplot(3, 1, 3), plot(T, (180/pi)*u, '--b', 'LineWidth', 1.5);

grid on
title('\bfControl Surface Response', 'FontName', 'Calibri', 'FontSize', 14)
xlabel ('Time [\its\rm]', 'FontName', 'Calibri', 'FontSize', 12)
ylabel ('Control Surface Angle [\itdeg.\rm]', 'FontName', 'Calibri',...
        'FontSize', 12)
set(gca, 'FontName', 'Calibri', 'FontSize', 10)
legend('Control Surface Signal', 'Kalman Estimate', 'Control Input');

%-----------------------------------------------------------------------%
%----------------------------- Plot Modal PSD's --------------------------%
%-----------------------------------------------------------------------%

figure

plot_FFT(f_h, h_PSD, f_range, 1, 2, 'Plunge Signal PSD',...
                                    'Normalised PSD', 'b')
plot_FFT(f_a, a_PSD, f_range, 2, 2, 'Pitch Signal PSD',...
                                    'Normalised PSD', 'g')

plot_Peak_Width(h_Width, h_Height, h_PSD_Peaks, 1, 2, 'b')
plot_Peak_Width(a_Width, a_Height, a_PSD_Peaks, 2, 2, 'g')

%=======================================================================%
```

```
%==========================================================================%
function [f_s, f_Range, Y_FFT, Y_PSD] = Signal_Processing(T, y)
%==========================================================================%
%
% T - Time period of samples [s]
% y - Samples
%
%--------------------------------------------------------------------------%
%-------------------------- Time Series Parameters -----------------------%
%--------------------------------------------------------------------------%

f_s = 1/(T(end)/length(T));          % Sample f_Range [Hz.]
m = floor(log(length(T))/log(2));    % Data points indice

%--------------------------------------------------------------------------%
%------------------------------ FFT Parameters ---------------------------%
%--------------------------------------------------------------------------%

Data_Points = 2^m;
Zeros = 32;
FFT_Points = Zeros*Data_Points;
df = f_s/FFT_Points;
Window_sf = 1.85;
f_Range = (0:(FFT_Points/2 - 1))*df;

%--------------------------------------------------------------------------%
%------------------------------- FFT Analysis ----------------------------%
%--------------------------------------------------------------------------%

Window = blackmanharris(Data_Points)';

% Other possible window functions to consider:
%
% Window = barthannwin(Data_Points)';
% Window = bartlett(Data_Points)';
% Window = bohmanwin(Data_Points)';
% Window = chebwin(Data_Points, 50)';
% Window = gausswin(Data_Points)';
% Window = hamming(Data_Points)';
% Window = hanning(Data_Points)';
% Window = kaiser(Data_Points, 5)';
% Window = nuttallwin(Data_Points)';
% Window = rectwin(Data_Points)';
% Window = tukeywin(Data_Points, 50)';

y_FFT = [Window.*y(1: Data_Points) zeros(1, FFT_Points - Data_Points)];
Y_FFT_F = (fft(y_FFT(1: FFT_Points))/Data_Points)*(2*Window_sf);
Y_FFT = abs(Y_FFT_F(1: FFT_Points/2));

Y_PSD_F = Y_FFT_F.* conj(Y_FFT_F);
Y_PSD = Y_PSD_F(1: FFT_Points/2);

%==========================================================================%
```

```
%=========================================================================%
function Peaks = Find_Peaks(x, y, N, Method)
%=========================================================================%

warning off all

j = 0;
i_end = (length(y) - 2);

for i = 1: i_end

    if (y(i) <= y(i + 1)) && (y(i + 1) >= y(i + 2)) && y(i + 1) > 0

        j = j + 1;

        if strcmp(Method, 'curve fit on')

            x_fit = [x(i) x(i + 1) x(i + 2)];
            y_fit = [y(i) y(i + 1) y(i + 2)];
            P_fit = polyfit(x_fit, y_fit, 2);
            x_max = -P_fit(2)/(2*P_fit(1));
            y_max = P_fit(1)*x_max^2 + P_fit(2)*x_max + P_fit(3);
            x_Peak(j) = -P_fit(2)/(2*P_fit(1));
            y_Peak(j) = P_fit(1)*x_max^2 + P_fit(2)*x_max + P_fit(3);

        elseif strcmp(Method, 'curve fit off')

            x_Peak(j) = x(i + 1);
            y_Peak(j) = y(i + 1);

        end

        index(j, 1) = i + 1;

    end

end

Peaks_temp = [y_Peak' x_Peak'];
Peaks_temp = [Peaks_temp index];
Peaks_temp = sortrows(Peaks_temp);
Peaks_temp((1: end - N), :) = [];
Peaks_temp = flipud(Peaks_temp);

[Peaks.frequency, s_index] = sort(Peaks_temp(:, 2), 'ascend');

for j = 1: N

    Peaks.index(j) = Peaks_temp(s_index(j), 3);
    Peaks.amplitude(j) = Peaks_temp(s_index(j), 1);

end

%=========================================================================%
```

```
%==========================================================================%
function [Width, Height] = Peak_Width(Peaks, Amplitude, Frequency)
%==========================================================================%

j_end = length(Peaks.frequency);
Width = zeros(j_end, 2);
Height = zeros(j_end, 1);

for j = 1: j_end

    start_pt = Peaks.index(j);
    Height(j) = 0.5*Peaks.amplitude(j);

    for i = start_pt: -1: 2

        if (Amplitude(i) >= Height(j)) && (Amplitude(i - 1) <= Height(j))

            x = [Frequency(i) Frequency(i - 1)];
            y = [Amplitude(i) Amplitude(i - 1)];
            Width(j, 1) = interp1(y, x, Height(j));
            break

        end

    end

    clear x
    clear y

    for i = start_pt: (length(Amplitude) - 1)

        if (Amplitude(i) >= Height(j)) && (Amplitude(i + 1) <= Height(j))

            x = [Frequency(i) Frequency(i + 1)];
            y = [Amplitude(i) Amplitude(i + 1)];
            Width(j, 2) = interp1(y, x, Height(j));
            break

        end

    end

    clear x
    clear y

end

%==========================================================================%
```

```
%=========================================================================%
function plot_FFT(X_FFT, Y_FFT, f_range, i_plot, N_plots, Title, YAxis,...
                   Colour)
%=========================================================================%

subplot(N_plots, 1, i_plot), plot (X_FFT, Y_FFT, Colour, 'LineWidth', 1.5)
grid on

title (['\bf', Title], 'FontName', 'Calibri', 'FontSize', 14)
xlabel ('Frequency [\itHz.\rm]', 'FontName', 'Calibri', 'FontSize', 12)
ylabel (YAxis, 'FontName', 'Calibri', 'FontSize', 12)
set(gca, 'XLim', f_range, 'FontName', 'Calibri', 'FontSize', 10)

%=========================================================================%
```

```
%==========================================================================%
function plot_Peak_Width(Width, Height, Peaks, i_plot, N_plots, colour)
%==========================================================================%

for i = 1: length(Peaks.frequency)

    f_Peak = Peaks.frequency(i);
    A_Peak = Peaks.amplitude(i);

    subplot(N_plots, 1, i_plot), hold on

    plot(f_Peak, A_Peak, strcat('.', colour), 'MarkerFaceColor', colour,...
                                              'MarkerSize', 18)
    plot([f_Peak, f_Peak], [A_Peak 0], strcat(':', colour),...
         'LineWidth', 1.5)
    plot([Width(i, 1), Width(i, 2)], [Height(i), Height(i)],...
         strcat(':.', colour), 'LineWidth', 1.5, 'MarkerFaceColor',...
         colour, 'MarkerSize', 18)
    plot([Width(i, 1), Width(i, 1)], [Height(i), 0],...
          strcat(':', colour), 'LineWidth', 1.5)
    plot([Width(i, 2), Width(i, 2)], [Height(i), 0],...
          strcat(':', colour), 'LineWidth', 1.5)

end

%==========================================================================%
```

Appendix D Aeroservoelastic Model and LQG Controller Block Diagrams

Simulink™ block diagrams of the model developed in Chapter 3 to simulate the open- and closed-loop response of a three degree-of-freedom binary flutter model and LQG feedback flutter controller are given here. Fixed parameters in the model were obtained from the "*FAST_Model_Data.m*" function listed in Appendix C. The Simulink™ model of the three degree-of-freedom binary flutter model was made up of the following block diagrams:

- *Complete FAST Model:* Main block diagram to represent the FAST model in its closed-loop configuration
- *Measurements and Noise:* The block diagram to output the measured variables to the complete FAST model block diagram and apply the anticipated noise to these signals
- *Kalman Filter:* The block diagram used to estimate the unmeasured states of the flutter model and provide an estimate of all system states to the LQR feedback controller which is used to calculate the control input
- *Actuator Constraints:* Block diagram to account for control surface actuator limitations and constraints (dead-band, rate limitations and saturation)
- *Structural Equations:* Block diagram to model the structural dynamics of a three degree-of-freedom binary flutter model
- *Unsteady Aerodynamics:* Block diagram to model the unsteady aerodynamics of a three degree-of-freedom binary flutter model
- *Aerodynamic Lag States:* The block diagram that calculates the aerodynamic lag states of a three degree-of-freedom binary flutter model
- *Aeroelastic Model:* Model that combines the structural equations and unsteady aerodynamic forces of a three degree-of-freedom binary flutter model

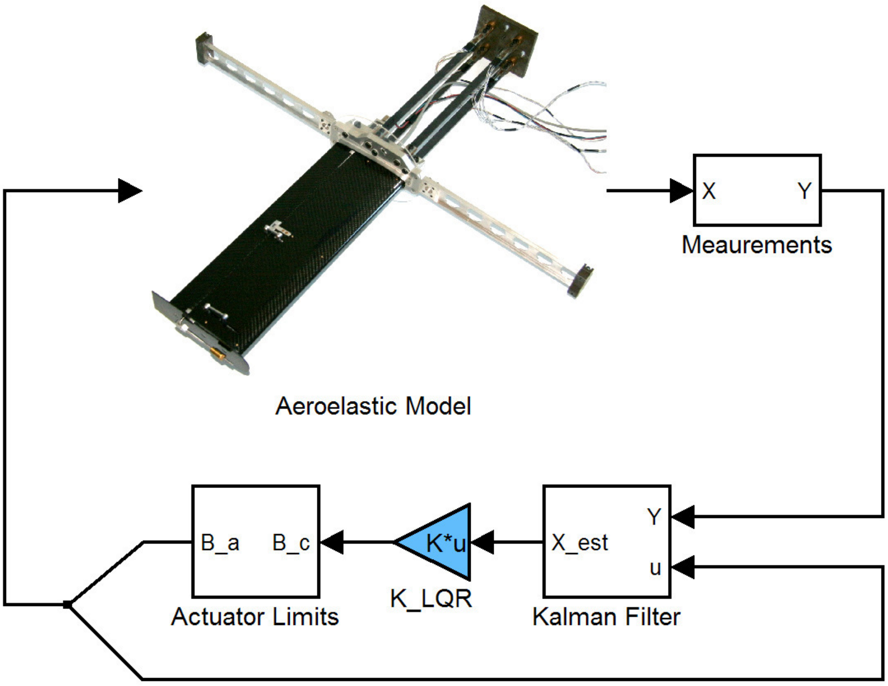


Figure D.1 Complete FAST Model Block Diagram

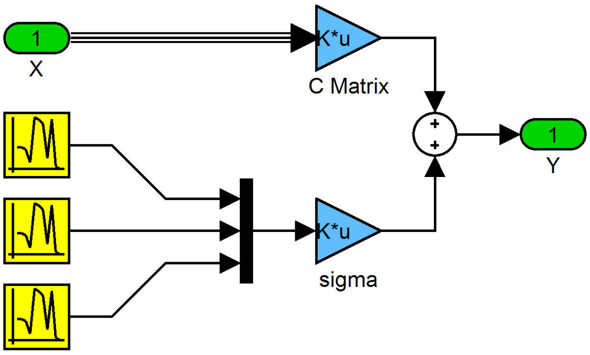


Figure D.2 Measurements and Noise Block Diagram

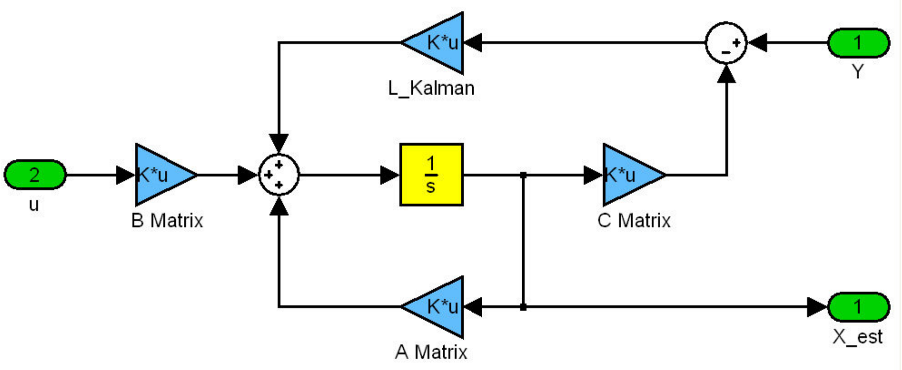


Figure D.3 Kalman Filter Block Diagram

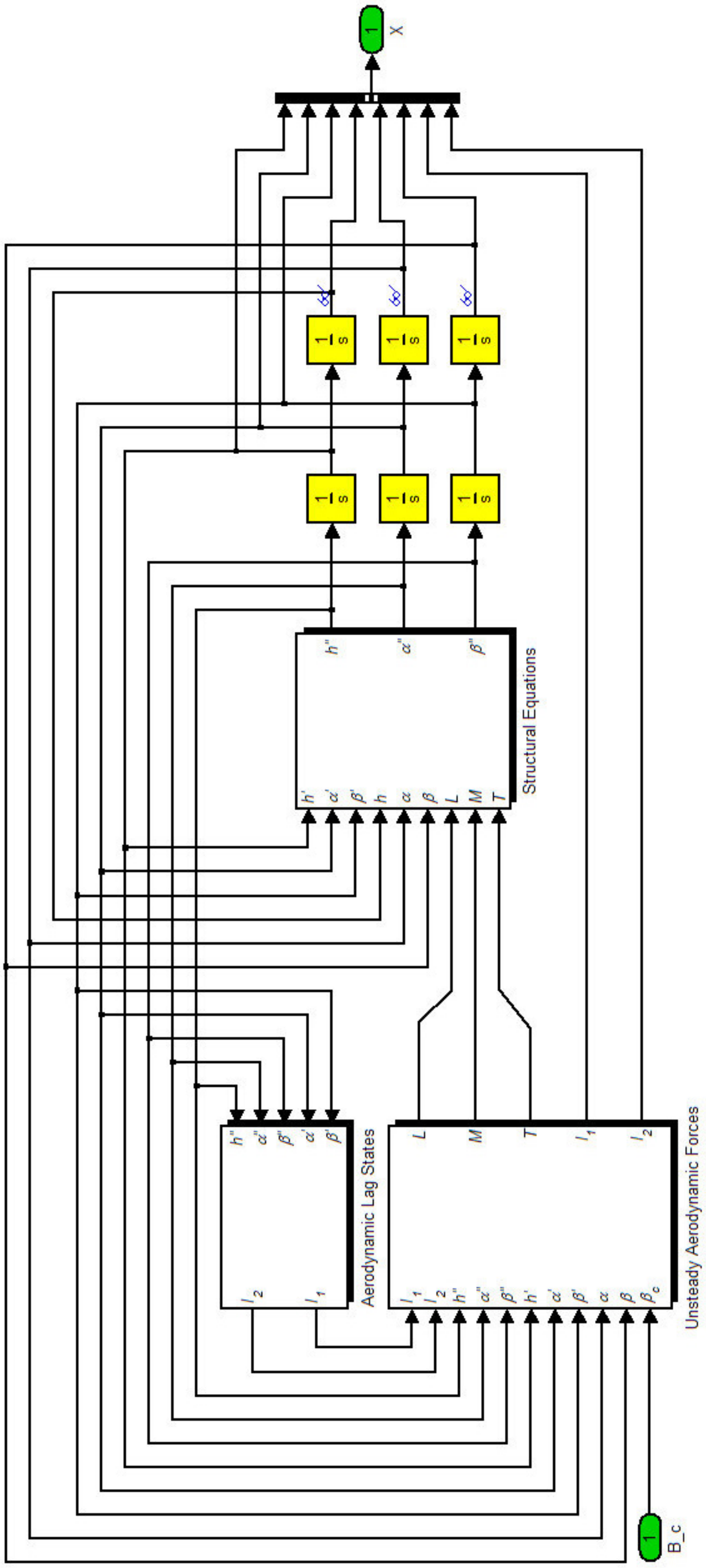


Figure D.4 Aeroelastic Model Block Diagram

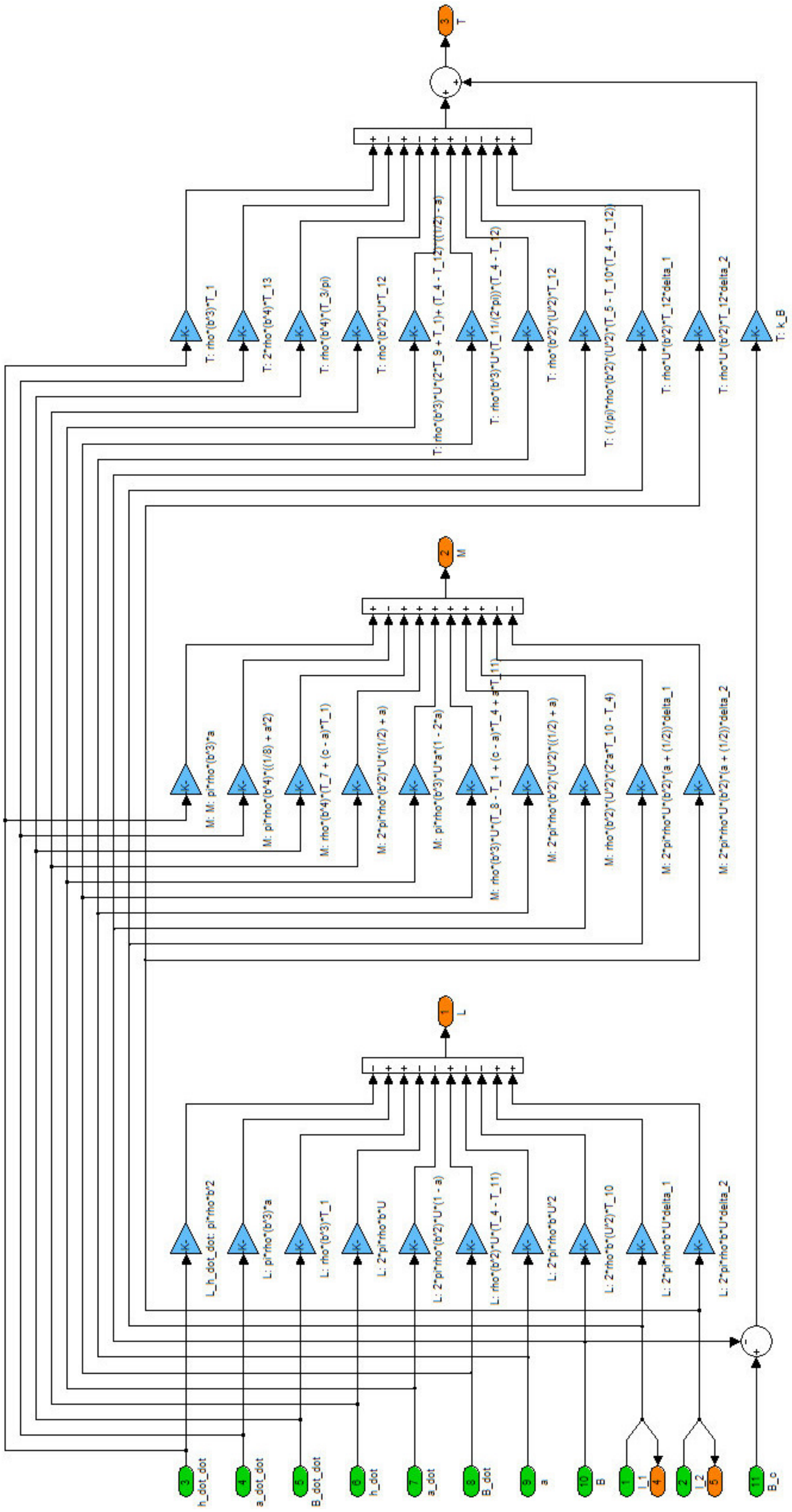


Figure D.5 Unsteady Aerodynamics Block Diagram

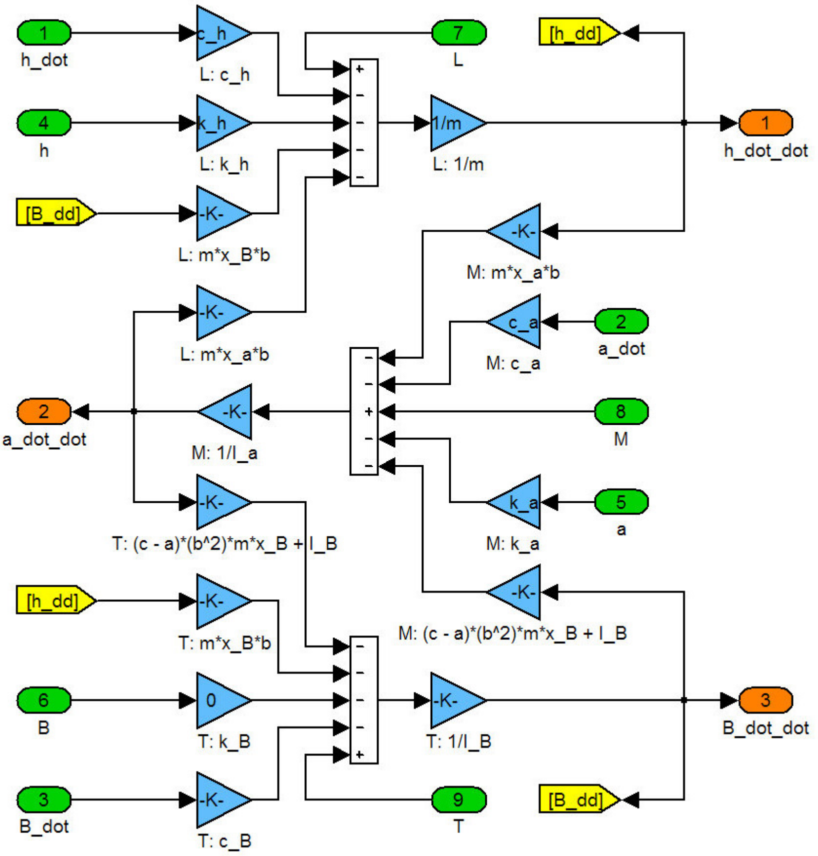


Figure D.6 Structural Equations Block Diagram

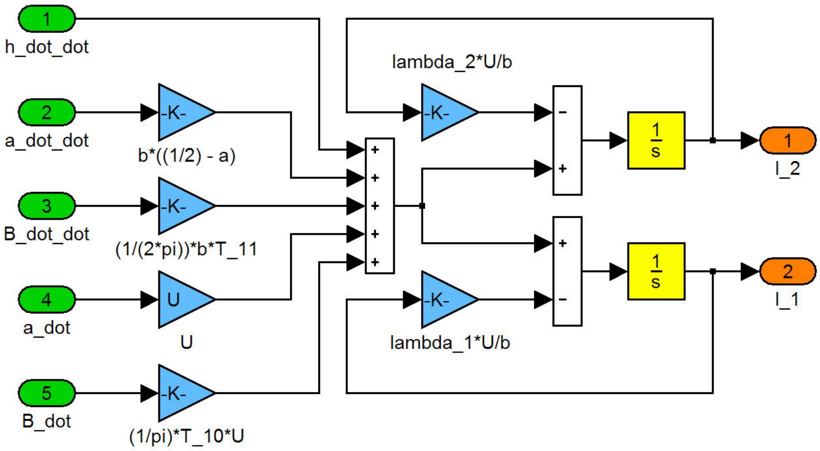


Figure D.7 Aerodynamic Lag States Block Diagram

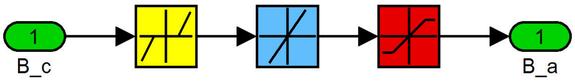
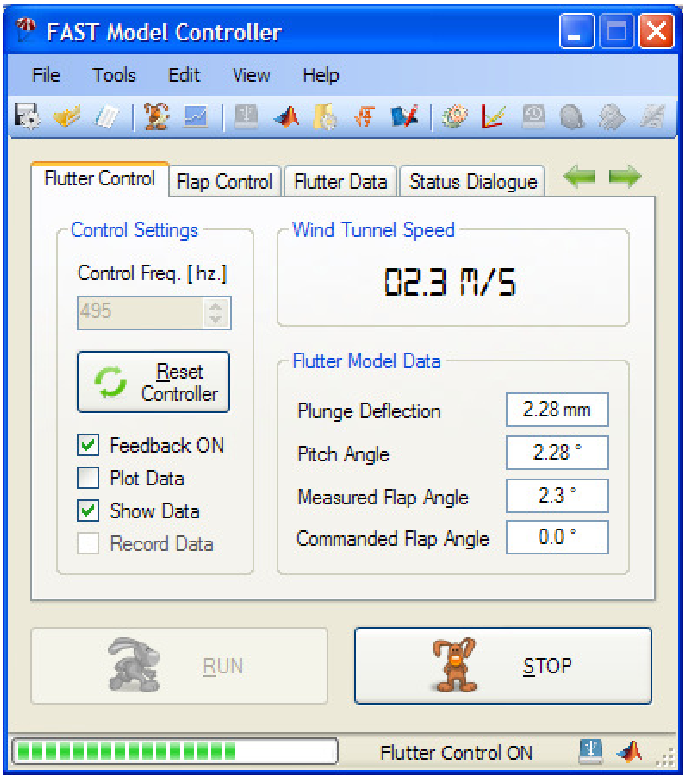


Figure D.8 Actuator Constraints Block Diagram

Appendix E FAST Model Controller Software

Screen shots of the FAST model controller software described in Section 4.4.3 are given here, to convey the functionality and main features of the software.



Control Freq. [hz.] 495

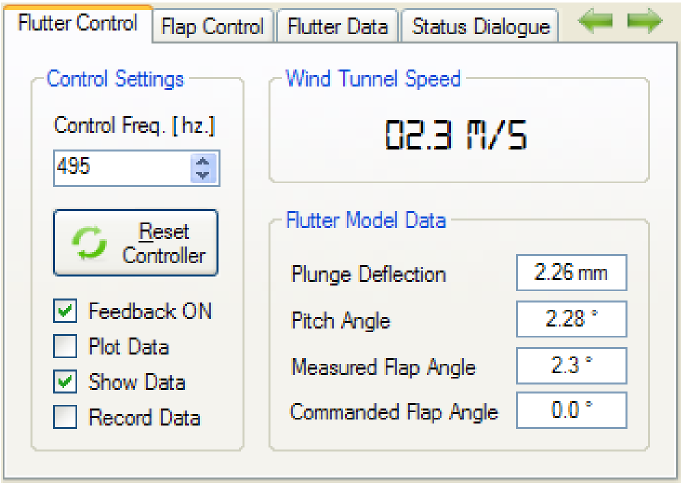
02.3 M/S

☑ Feedback ON
☐ Plot Data
☑ Show Data
☐ Record Data

Plunge Deflection 2.28 mm
Pitch Angle 2.28 °
Measured Flap Angle 2.3 °
Commanded Flap Angle 0.0 °

Flutter Control ON

Figure E.1 FAST Model Controller Main User Interface



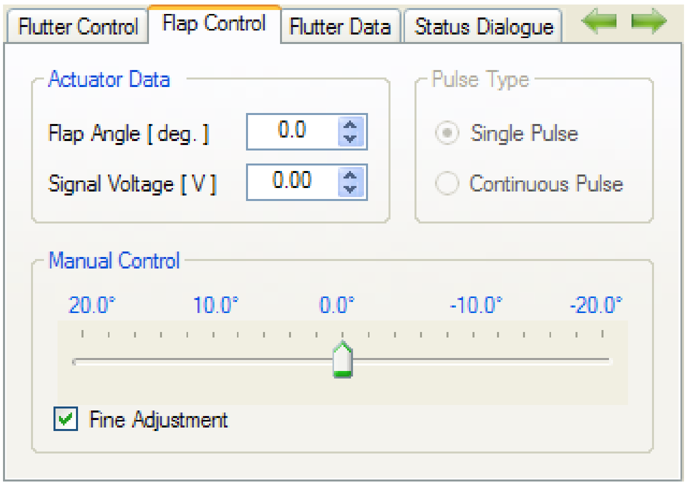
Control Freq. [hz.] 495

02.3 M/S

☑ Feedback ON
☐ Plot Data
☑ Show Data
☐ Record Data

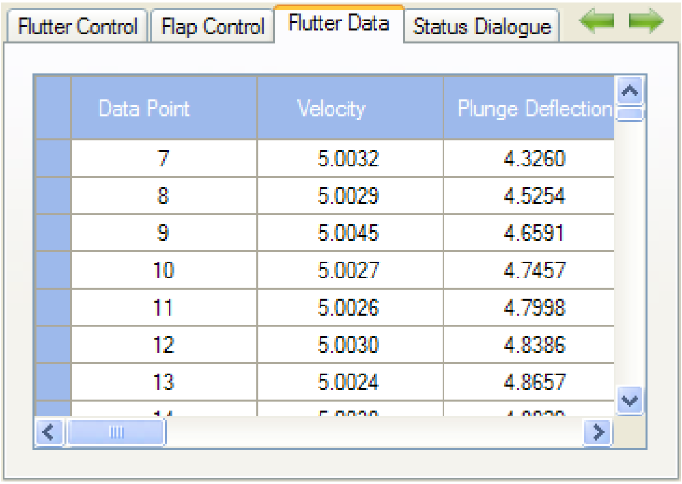
Plunge Deflection 2.26 mm
Pitch Angle 2.28 °
Measured Flap Angle 2.3 °
Commanded Flap Angle 0.0 °

Figure E.2 FAST Model Controller Flutter Control Tab



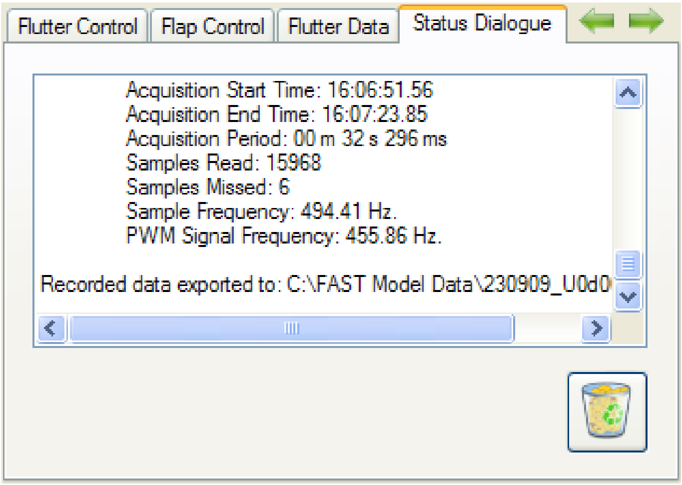
Flap Angle [deg.] 0.0
Signal Voltage [V] 0.00
● Single Pulse
○ Continuous Pulse
☑ Fine Adjustment

Figure E.3   FAST Model Controller Manual Control Tab



Data Point	Velocity	Plunge Deflection
7	5.0032	4.3260
8	5.0029	4.5254
9	5.0045	4.6591
10	5.0027	4.7457
11	5.0026	4.7998
12	5.0030	4.8386
13	5.0024	4.8657

Figure E.4   FAST Model Controller Data Tab



```
        Acquisition Start Time: 16:06:51.56
        Acquisition End Time: 16:07:23.85
        Acquisition Period: 00 m 32 s 296 ms
        Samples Read: 15968
        Samples Missed: 6
        Sample Frequency: 494.41 Hz.
        PWM Signal Frequency: 455.86 Hz.

Recorded data exported to: C:\FAST Model Data\230909_U0d0
```

Figure E.5   FAST Model Controller Status Tab

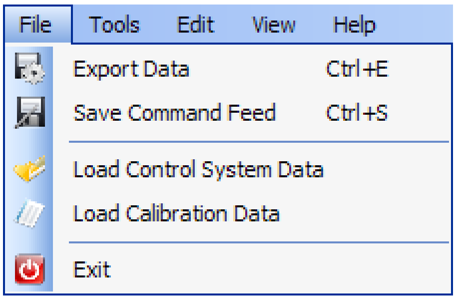


Figure E.6 FAST Model Controller File Menu

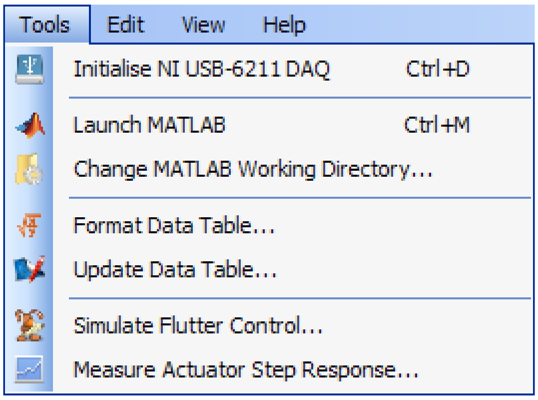


Figure E.7 FAST Model Controller Tools Menu

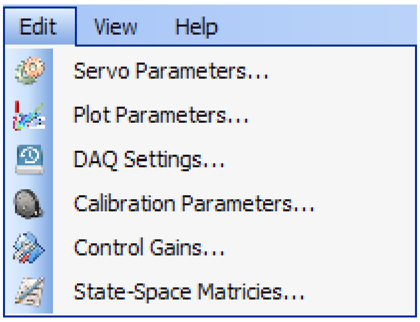


Figure E.8 FAST Model Controller Edit Menu

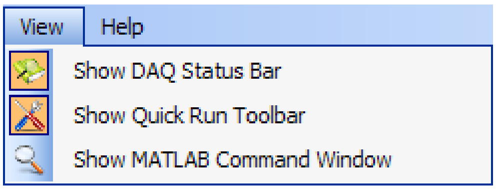


Figure E.9 FAST Model Controller View Menu

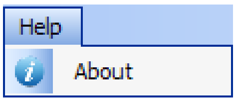
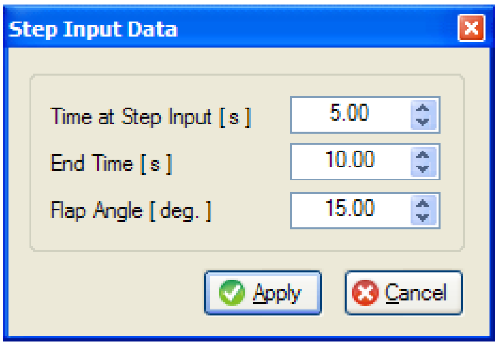
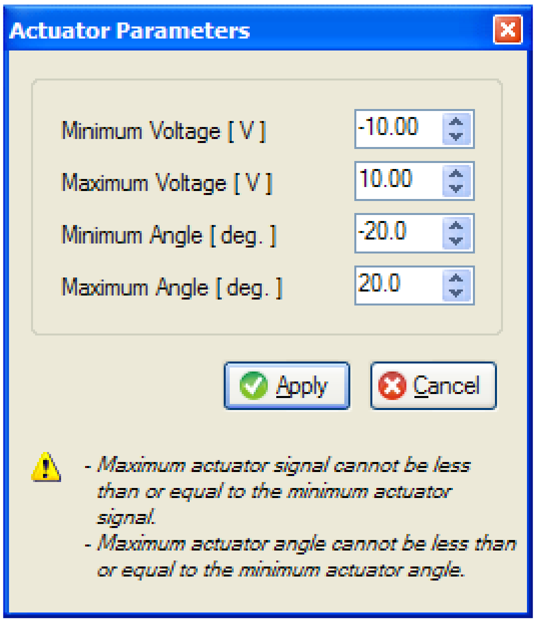


Figure E.10 FAST Model Controller Help Menu



Time at Step Input [s] 5.00
End Time [s] 10.00
Flap Angle [deg.] 15.00

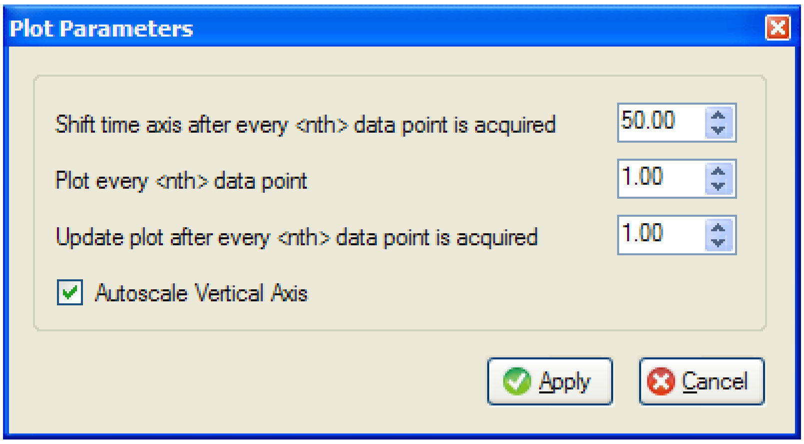
Figure E.11 FAST Model Controller Step Input Data Dialogue Box



Minimum Voltage [V] -10.00
Maximum Voltage [V] 10.00
Minimum Angle [deg.] -20.0
Maximum Angle [deg.] 20.0

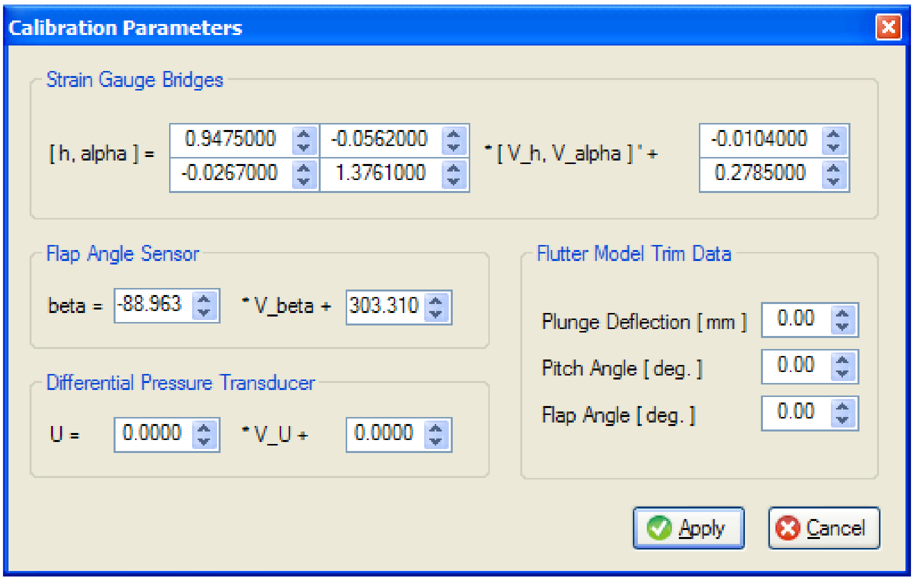
- *Maximum actuator signal cannot be less than or equal to the minimum actuator signal.*
- *Maximum actuator angle cannot be less than or equal to the minimum actuator angle.*

Figure E.12 FAST Model Controller Actuator Parameters Dialogue Box



Shift time axis after every <nth> data point is acquired 50.00
Plot every <nth> data point 1.00
Update plot after every <nth> data point is acquired 1.00
☑ Autoscale Vertical Axis

Figure E.13 FAST Model Controller Plot Parameters Dialogue Box



Calibration Parameters

Strain Gauge Bridges

[h, alpha] = 0.9475000 -0.0562000 / -0.0267000 1.3761000 * [V_h, V_alpha]' + -0.0104000 / 0.2785000

Flap Angle Sensor

beta = -88.963 * V_beta + 303.310

Differential Pressure Transducer

U = 0.0000 * V_U + 0.0000

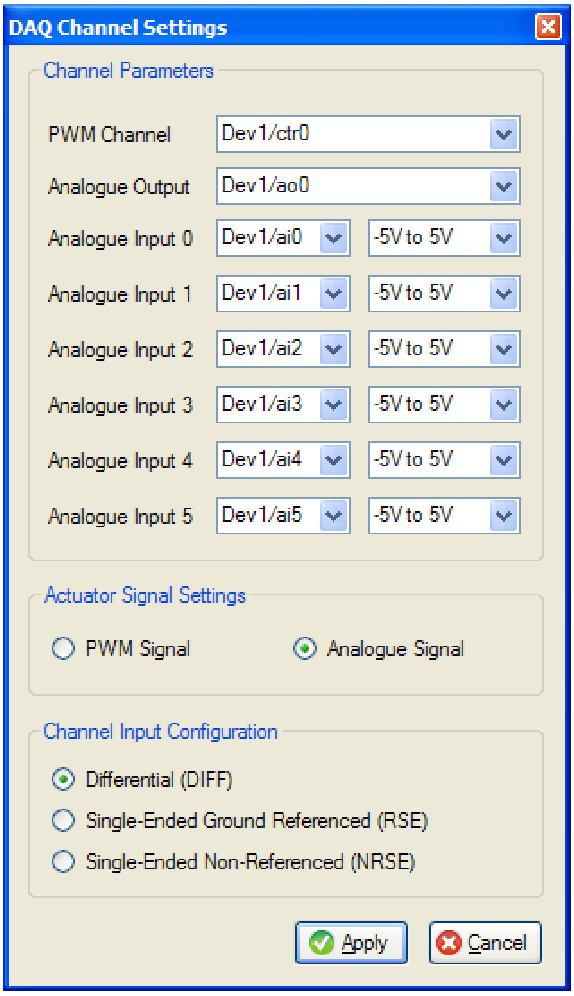
Flutter Model Trim Data

Plunge Deflection [mm] 0.00

Pitch Angle [deg.] 0.00

Flap Angle [deg.] 0.00

Figure E.14 FAST Model Controller Calibration Data Dialogue Box



DAQ Channel Settings

Channel Parameters

PWM Channel: Dev1/ctr0

Analogue Output: Dev1/ao0

Analogue Input 0: Dev1/ai0, -5V to 5V

Analogue Input 1: Dev1/ai1, -5V to 5V

Analogue Input 2: Dev1/ai2, -5V to 5V

Analogue Input 3: Dev1/ai3, -5V to 5V

Analogue Input 4: Dev1/ai4, -5V to 5V

Analogue Input 5: Dev1/ai5, -5V to 5V

Actuator Signal Settings

○ PWM Signal ● Analogue Signal

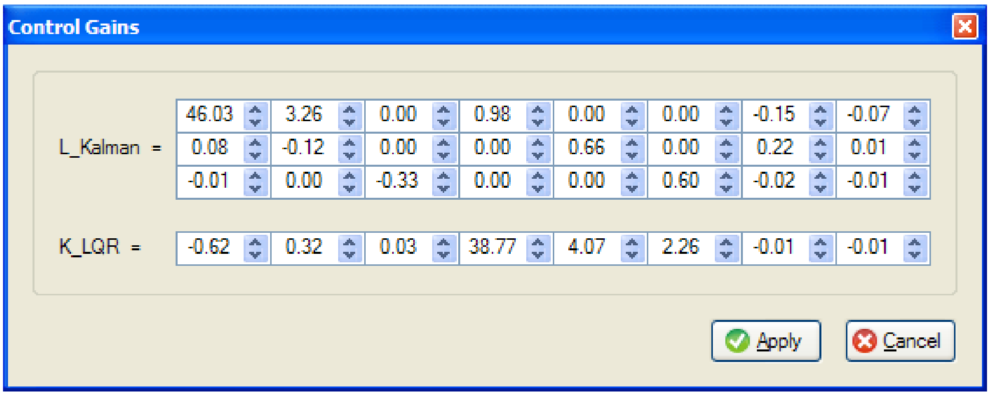
Channel Input Configuration

● Differential (DIFF)

○ Single-Ended Ground Referenced (RSE)

○ Single-Ended Non-Referenced (NRSE)

Figure E.15 FAST Model Controller DAQ Settings Dialogue Box



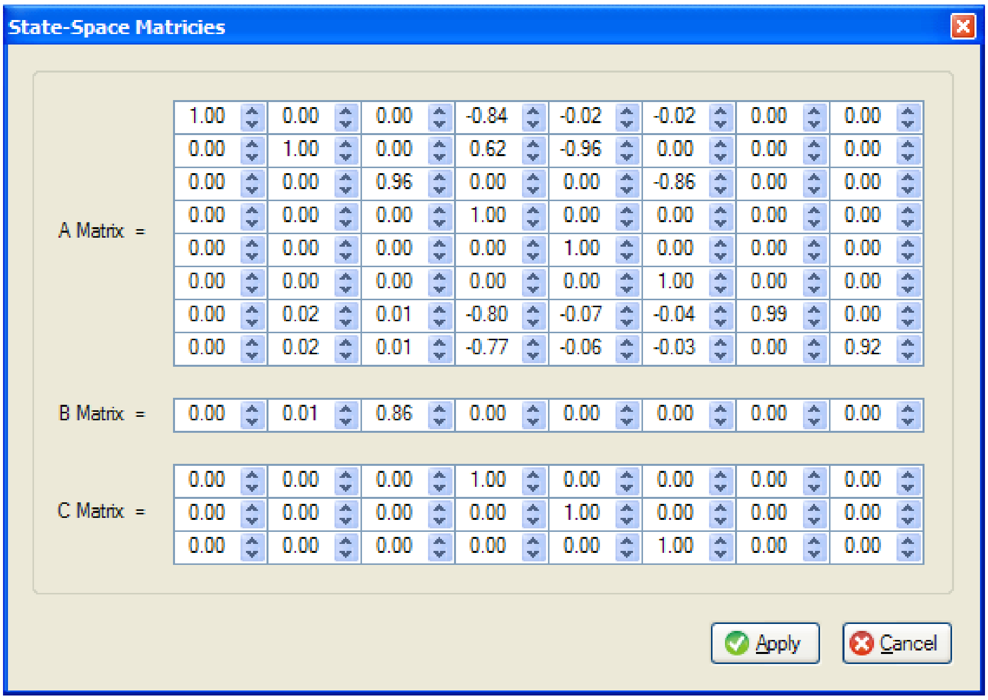
L_Kalman =

46.03	3.26	0.00	0.98	0.00	0.00	-0.15	-0.07
0.08	-0.12	0.00	0.00	0.66	0.00	0.22	0.01
-0.01	0.00	-0.33	0.00	0.00	0.60	-0.02	-0.01

K_LQR =

-0.62	0.32	0.03	38.77	4.07	2.26	-0.01	-0.01

Figure E.16 FAST Model Controller Control Gains Dialogue Box



A Matrix =

1.00	0.00	0.00	-0.84	-0.02	-0.02	0.00	0.00
0.00	1.00	0.00	0.62	-0.96	0.00	0.00	0.00
0.00	0.00	0.96	0.00	0.00	-0.86	0.00	0.00
0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00
0.00	0.02	0.01	-0.80	-0.07	-0.04	0.99	0.00
0.00	0.02	0.01	-0.77	-0.06	-0.03	0.00	0.92

B Matrix =

0.00	0.01	0.86	0.00	0.00	0.00	0.00	0.00

C Matrix =

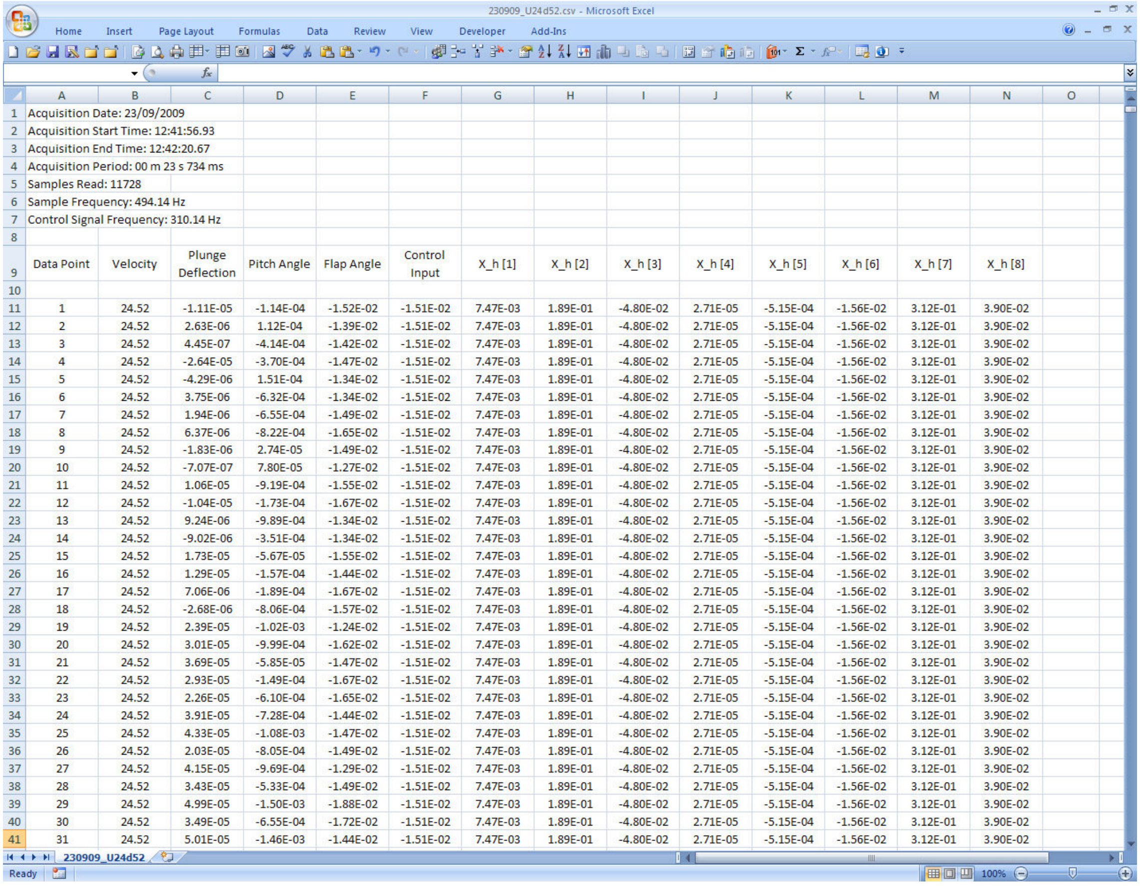
0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00

Figure E.17 FAST Model Controller State-Space Matrices Dialogue Box

FAST Model Controller
Version 1.0.0.0
Copyright © 2009
CSIR

Software to control the FAST (Flutter; with Active Suppression Technology) wind tunnel model developed by the

Figure E.18 FAST Model Controller About Dialogue Box



Acquisition Date: 23/09/2009
Acquisition Start Time: 12:41:56.93
Acquisition End Time: 12:42:20.67
Acquisition Period: 00 m 23 s 734 ms
Samples Read: 11728
Sample Frequency: 494.14 Hz
Control Signal Frequency: 310.14 Hz

Data Point	Velocity	Plunge Deflection	Pitch Angle	Flap Angle	Control Input	X_h [1]	X_h [2]	X_h [3]	X_h [4]	X_h [5]	X_h [6]	X_h [7]	X_h [8]
1	24.52	-1.11E-05	-1.14E-04	-1.52E-02	-1.51E-02	7.47E-03	1.89E-01	-4.80E-02	2.71E-05	-5.15E-04	-1.56E-02	3.12E-01	3.90E-02
2	24.52	2.63E-06	1.12E-04	-1.39E-02	-1.51E-02	7.47E-03	1.89E-01	-4.80E-02	2.71E-05	-5.15E-04	-1.56E-02	3.12E-01	3.90E-02
3	24.52	4.45E-07	-4.14E-04	-1.42E-02	-1.51E-02	7.47E-03	1.89E-01	-4.80E-02	2.71E-05	-5.15E-04	-1.56E-02	3.12E-01	3.90E-02
4	24.52	-2.64E-05	-3.70E-04	-1.47E-02	-1.51E-02	7.47E-03	1.89E-01	-4.80E-02	2.71E-05	-5.15E-04	-1.56E-02	3.12E-01	3.90E-02
5	24.52	-4.29E-06	1.51E-04	-1.34E-02	-1.51E-02	7.47E-03	1.89E-01	-4.80E-02	2.71E-05	-5.15E-04	-1.56E-02	3.12E-01	3.90E-02
6	24.52	3.75E-06	-6.32E-04	-1.34E-02	-1.51E-02	7.47E-03	1.89E-01	-4.80E-02	2.71E-05	-5.15E-04	-1.56E-02	3.12E-01	3.90E-02
7	24.52	1.94E-06	-6.55E-04	-1.49E-02	-1.51E-02	7.47E-03	1.89E-01	-4.80E-02	2.71E-05	-5.15E-04	-1.56E-02	3.12E-01	3.90E-02
8	24.52	6.37E-06	-8.22E-04	-1.65E-02	-1.51E-02	7.47E-03	1.89E-01	-4.80E-02	2.71E-05	-5.15E-04	-1.56E-02	3.12E-01	3.90E-02
9	24.52	-1.83E-06	2.74E-05	-1.49E-02	-1.51E-02	7.47E-03	1.89E-01	-4.80E-02	2.71E-05	-5.15E-04	-1.56E-02	3.12E-01	3.90E-02
10	24.52	-7.07E-07	7.80E-05	-1.27E-02	-1.51E-02	7.47E-03	1.89E-01	-4.80E-02	2.71E-05	-5.15E-04	-1.56E-02	3.12E-01	3.90E-02
11	24.52	1.06E-05	-9.19E-04	-1.55E-02	-1.51E-02	7.47E-03	1.89E-01	-4.80E-02	2.71E-05	-5.15E-04	-1.56E-02	3.12E-01	3.90E-02
12	24.52	-1.04E-05	-1.73E-04	-1.67E-02	-1.51E-02	7.47E-03	1.89E-01	-4.80E-02	2.71E-05	-5.15E-04	-1.56E-02	3.12E-01	3.90E-02
13	24.52	9.24E-06	-9.89E-04	-1.34E-02	-1.51E-02	7.47E-03	1.89E-01	-4.80E-02	2.71E-05	-5.15E-04	-1.56E-02	3.12E-01	3.90E-02
14	24.52	-9.02E-06	-3.51E-04	-1.34E-02	-1.51E-02	7.47E-03	1.89E-01	-4.80E-02	2.71E-05	-5.15E-04	-1.56E-02	3.12E-01	3.90E-02
15	24.52	1.73E-05	-5.67E-05	-1.55E-02	-1.51E-02	7.47E-03	1.89E-01	-4.80E-02	2.71E-05	-5.15E-04	-1.56E-02	3.12E-01	3.90E-02
16	24.52	1.29E-05	-1.57E-04	-1.44E-02	-1.51E-02	7.47E-03	1.89E-01	-4.80E-02	2.71E-05	-5.15E-04	-1.56E-02	3.12E-01	3.90E-02
17	24.52	7.06E-06	-1.89E-04	-1.67E-02	-1.51E-02	7.47E-03	1.89E-01	-4.80E-02	2.71E-05	-5.15E-04	-1.56E-02	3.12E-01	3.90E-02
18	24.52	-2.68E-06	-8.06E-04	-1.57E-02	-1.51E-02	7.47E-03	1.89E-01	-4.80E-02	2.71E-05	-5.15E-04	-1.56E-02	3.12E-01	3.90E-02
19	24.52	2.39E-05	-1.02E-03	-1.24E-02	-1.51E-02	7.47E-03	1.89E-01	-4.80E-02	2.71E-05	-5.15E-04	-1.56E-02	3.12E-01	3.90E-02
20	24.52	3.01E-05	-9.99E-04	-1.62E-02	-1.51E-02	7.47E-03	1.89E-01	-4.80E-02	2.71E-05	-5.15E-04	-1.56E-02	3.12E-01	3.90E-02
21	24.52	3.69E-05	-5.85E-05	-1.47E-02	-1.51E-02	7.47E-03	1.89E-01	-4.80E-02	2.71E-05	-5.15E-04	-1.56E-02	3.12E-01	3.90E-02
22	24.52	2.93E-05	-1.49E-04	-1.67E-02	-1.51E-02	7.47E-03	1.89E-01	-4.80E-02	2.71E-05	-5.15E-04	-1.56E-02	3.12E-01	3.90E-02
23	24.52	2.26E-05	-6.10E-04	-1.65E-02	-1.51E-02	7.47E-03	1.89E-01	-4.80E-02	2.71E-05	-5.15E-04	-1.56E-02	3.12E-01	3.90E-02
24	24.52	3.91E-05	-7.28E-04	-1.44E-02	-1.51E-02	7.47E-03	1.89E-01	-4.80E-02	2.71E-05	-5.15E-04	-1.56E-02	3.12E-01	3.90E-02
25	24.52	4.33E-05	-1.08E-03	-1.47E-02	-1.51E-02	7.47E-03	1.89E-01	-4.80E-02	2.71E-05	-5.15E-04	-1.56E-02	3.12E-01	3.90E-02
26	24.52	2.03E-05	-8.05E-04	-1.49E-02	-1.51E-02	7.47E-03	1.89E-01	-4.80E-02	2.71E-05	-5.15E-04	-1.56E-02	3.12E-01	3.90E-02
27	24.52	4.15E-05	-9.69E-04	-1.29E-02	-1.51E-02	7.47E-03	1.89E-01	-4.80E-02	2.71E-05	-5.15E-04	-1.56E-02	3.12E-01	3.90E-02
28	24.52	3.43E-05	-5.33E-04	-1.49E-02	-1.51E-02	7.47E-03	1.89E-01	-4.80E-02	2.71E-05	-5.15E-04	-1.56E-02	3.12E-01	3.90E-02
29	24.52	4.99E-05	-1.50E-03	-1.88E-02	-1.51E-02	7.47E-03	1.89E-01	-4.80E-02	2.71E-05	-5.15E-04	-1.56E-02	3.12E-01	3.90E-02
30	24.52	3.49E-05	-6.55E-04	-1.72E-02	-1.51E-02	7.47E-03	1.89E-01	-4.80E-02	2.71E-05	-5.15E-04	-1.56E-02	3.12E-01	3.90E-02
31	24.52	5.01E-05	-1.46E-03	-1.44E-02	-1.51E-02	7.47E-03	1.89E-01	-4.80E-02	2.71E-05	-5.15E-04	-1.56E-02	3.12E-01	3.90E-02

Figure E.19 FAST Model Controller Recorded Data Output File

Appendix F Pitch and Plunge Strain Gauge Bridge Calibration Process

The strain gauge bridges on the FAST model flexure mount were calibrated to determine the outputs corresponding to pitch angles and plunge deflections of FAST model wing. Section 6.1.1 discusses the calibration process of both the plunge and pitch SGBs using the specially designed and manufactured calibration rig. This appendix presents the mathematics of the calibration process outlined in Section 6.1.1. Assuming a linear relationship between the mount deflections (pitch and plunge) and the output voltage of the corresponding pitch and plunge SGBs then:

$$\begin{bmatrix} V_h \\ V_\alpha \end{bmatrix} = \begin{bmatrix} \varepsilon_h & \varepsilon_{h\alpha} \\ \varepsilon_{\alpha h} & \varepsilon_\alpha \end{bmatrix} \begin{bmatrix} h \\ \alpha \end{bmatrix} + \begin{bmatrix} \varepsilon_h^{(0)} \\ \varepsilon_\alpha^{(0)} \end{bmatrix} \quad \text{..........F.1}$$

where the voltage output from the plunge SGB is V_h and that from the pitch SGB is V_α. The off-diagonal terms in the strain matrix in Equation F.1 account for cross-coupling between the plunge and pitch modes of the FAST model mount. Since plunge and pitch deflections of the FAST model mount must be related to the plunge and pitch SGB signals respectively, Equation F.1 was solved for h and α as:

$$\begin{bmatrix} h \\ \alpha \end{bmatrix} = \begin{bmatrix} \varepsilon_h & \varepsilon_{h\alpha} \\ \varepsilon_{\alpha h} & \varepsilon_\alpha \end{bmatrix}^{-1} \begin{bmatrix} V_h \\ V_\alpha \end{bmatrix} - \begin{bmatrix} \varepsilon_h & \varepsilon_{h\alpha} \\ \varepsilon_{\alpha h} & \varepsilon_\alpha \end{bmatrix}^{-1} \begin{bmatrix} \varepsilon_h^{(0)} \\ \varepsilon_\alpha^{(0)} \end{bmatrix} \quad \text{..........F.2}$$

The terms in the second matrix multiplication in Equation F.2 are all constant, so it was multiplied out to give:

$$\begin{bmatrix} \Delta_h \\ \Delta_\alpha \end{bmatrix} = \begin{bmatrix} \varepsilon_h & \varepsilon_{h\alpha} \\ \varepsilon_{\alpha h} & \varepsilon_\alpha \end{bmatrix}^{-1} \begin{bmatrix} \varepsilon_h^{(0)} \\ \varepsilon_\alpha^{(0)} \end{bmatrix} \quad \text{..........F.3}$$

which then allows Equation F.2 to be written as:

$$\begin{bmatrix} h \\ \alpha \end{bmatrix} = \begin{bmatrix} \varepsilon_h & \varepsilon_{h\alpha} \\ \varepsilon_{\alpha h} & \varepsilon_\alpha \end{bmatrix}^{-1} \begin{bmatrix} V_h \\ V_\alpha \end{bmatrix} - \begin{bmatrix} \Delta_h \\ \Delta_\alpha \end{bmatrix} \quad \text{..........F.4}$$

If the modal deflections of the mount are written as a matrix $\mathbf{\Phi}$, defined as:

$$\mathbf{\Phi} \triangleq \begin{bmatrix} h & \alpha \end{bmatrix}^T \quad \text{..........F.5}$$

then Equation F.4 can be written in matrix form as:

$$\mathbf{\Phi} = \boldsymbol{\varepsilon}^{-1}\mathbf{V} - \mathbf{\Delta} \quad \text{..........F.6}$$

where $\boldsymbol{\varepsilon}^{-1}$ is the "slope" calibration matrix and $\boldsymbol{\Delta}$ the "intercept" calibration matrix. This allows the two SGB output voltages to be related to the pitch and plunge deflections of the mount. Thus the terms $\boldsymbol{\varepsilon}$ and $\boldsymbol{\Delta}$ in Equation F.6 need to be determined from a calibration process of the FAST model mount SGBs. To achieve this, the mount was loaded in pure plunge, pure pitch and coupled pitch-plunge modes whilst recording the pitch and plunge SGB signals, to generate the deflection matrix $\boldsymbol{\Phi}$ and the voltage output matrix $\mathbf{V}$. A surface fitted through these data yields $\boldsymbol{\varepsilon}^{-1}$ and $\boldsymbol{\Delta}$. The calibration data used to determine these terms is given in Table F.1 and plotted in Figure 6.1 (plunge SGB) and Figure 6.3 (pitch SGB), along with the corresponding calibration surfaces.

Table F.1 Flexure Mount Strain Gauge Bridges Calibration Data Matrix

	Pitch Angle [deg.]																								
Plunge Deflection [mm]	6.00	5.50	5.00	4.50	4.00	3.50	3.00	2.50	2.00	1.50	1.00	0.50	0.00	-0.50	-1.00	-1.50	-2.00	-2.50	-3.00	-3.50	-4.00	-4.50	-5.00	-5.50	-6.00
1.50													-4.23 -0.02												
2.00													-3.86 -0.01												
2.50			-3.16 3.50							-3.32 1.16			-3.50 0.01			-3.36 -0.84							-3.45 -3.75		
3.00													-3.14 0.02												
3.50													-2.79 0.03												
4.00						-2.21 2.57							-2.47 0.02							-2.48 -2.43					
4.50													-2.17 0.03												
5.00													-1.83 0.03												
5.50													-1.48 0.05												
6.00			-0.85 3.84							-0.84 1.33			-1.15 0.06			-0.96 -0.76							-1.17 -3.64		
6.50													-0.80 0.06												
7.00													-0.45 0.07												
7.50	0.11 4.62	0.12 4.29	0.13 3.92	0.14 3.54	0.14 3.19	0.15 2.84	0.15 2.49	0.16 2.10	0.15 1.79	0.14 1.42	0.12 1.05	0.1 0.69	0.00 0.24	-0.02 -0.07	-0.02 -0.45	-0.04 -0.84	-0.06 -1.21	-0.09 -1.62	-0.12 -2.02	-0.16 -2.42	-0.18 -2.82	-0.23 -3.26	-0.26 -3.69	-0.30 -4.14	-0.34 -4.62
8.00													0.26 0.07												
8.50													0.62 0.08												
9.00			1.21 4.15							1.27 1.42			0.96 0.08			1.15 -0.83							0.88 -3.66		
9.50													1.31 0.08												
10.00													1.66 0.09												
10.50													2.00 0.07												
11.00						2.75 3.05							2.34 0.07							2.45 -2.52					
11.50													2.67 0.07												
12.00													3.03 0.08												
12.50			3.56 4.42							3.61 1.65			3.37 0.09			3.50 -0.75							3.25 -3.89		
13.00													3.72 0.10												
13.50													4.06 0.12												

Appendix G Additional Open- and Closed-Loop Wind Tunnel Test Results

Included here are some representative results described in Chapter 6, but not presented there. These are the CWT wind-off open-loop time responses and corresponding power spectral densities for the model forward and aft cg position configurations (Section 6.3.2), the baseline cg position LSWT open-loop wind-off time response and PSDs and the baseline cg position LSWT critical flutter time response and corresponding PSDs (Section 6.3.4).

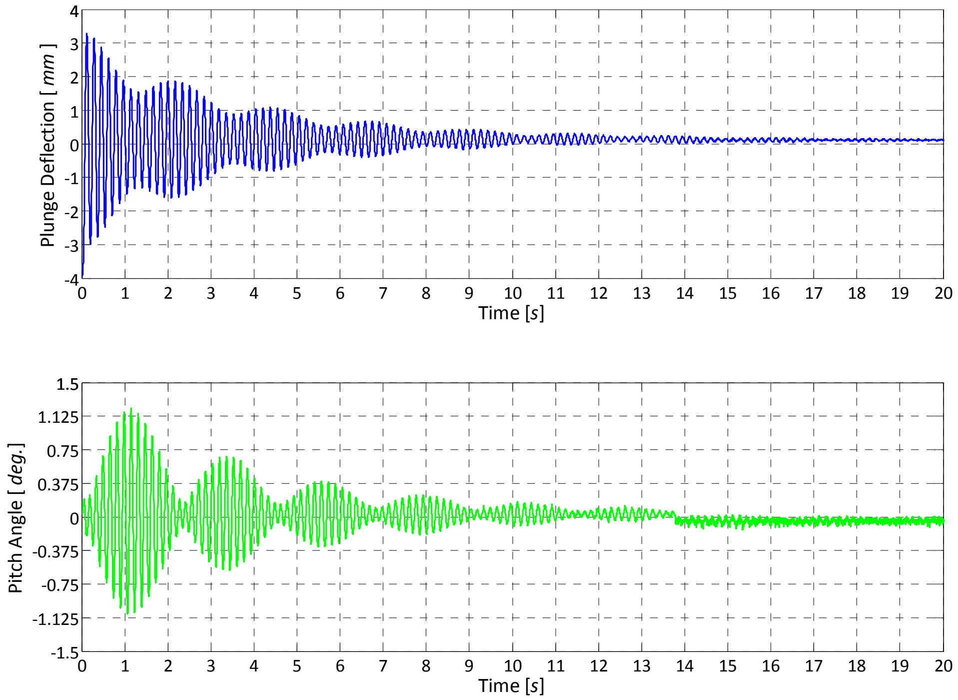
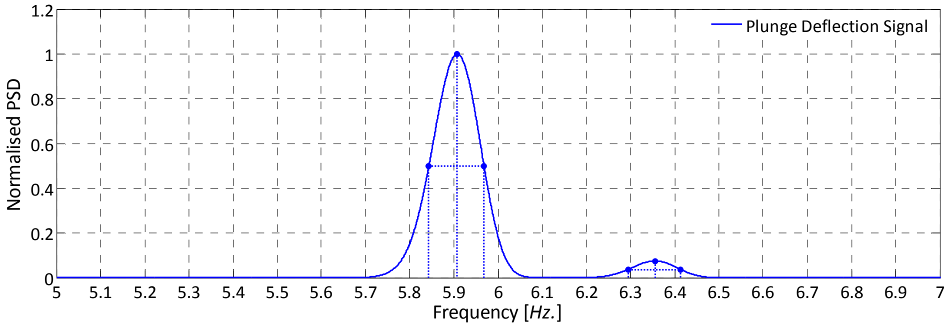


Figure G.1 FAST Model CWT Open-Loop Time Response (Forward cg Position; $U = 0.0 m/s$)



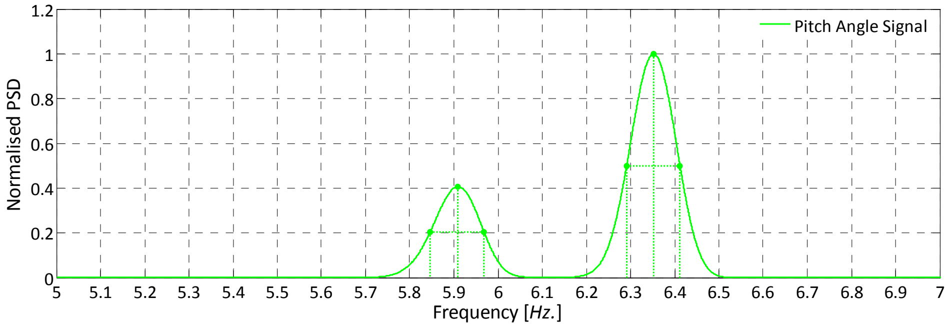
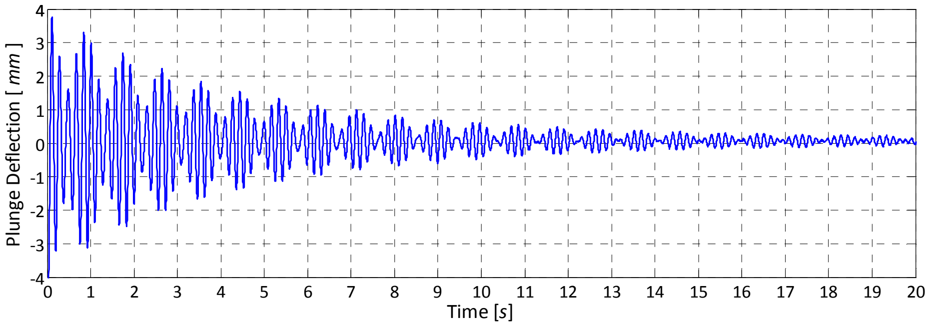


Figure G.2 Normalised PSD of CWT Open-Loop Response (Forward cg Position; $U = 0.0m/s$)



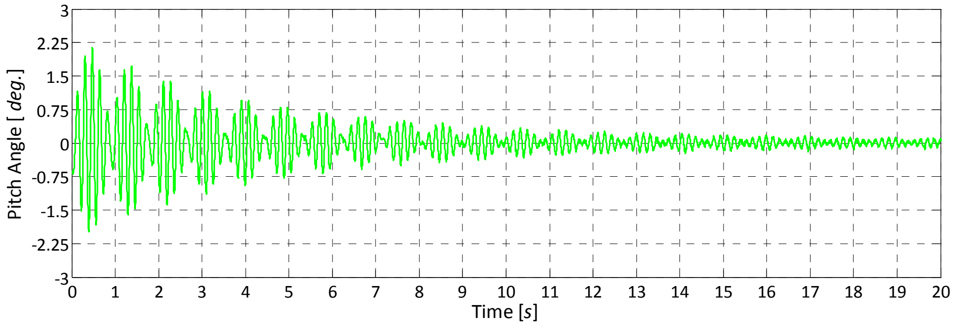


Figure G.3 FAST Model CWT Open-Loop Time Response (Aft cg Position; $U = 0.0m/s$)

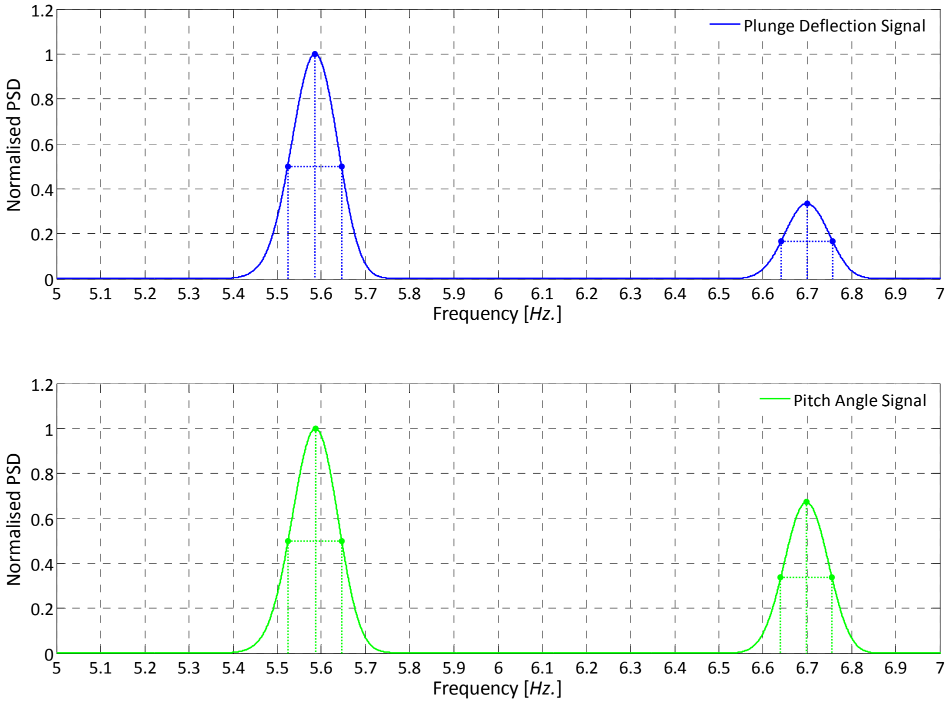


Figure G.4 Normalised PSD of CWT Open-Loop Response (Aft cg Position; $U = 0.0m/s$)

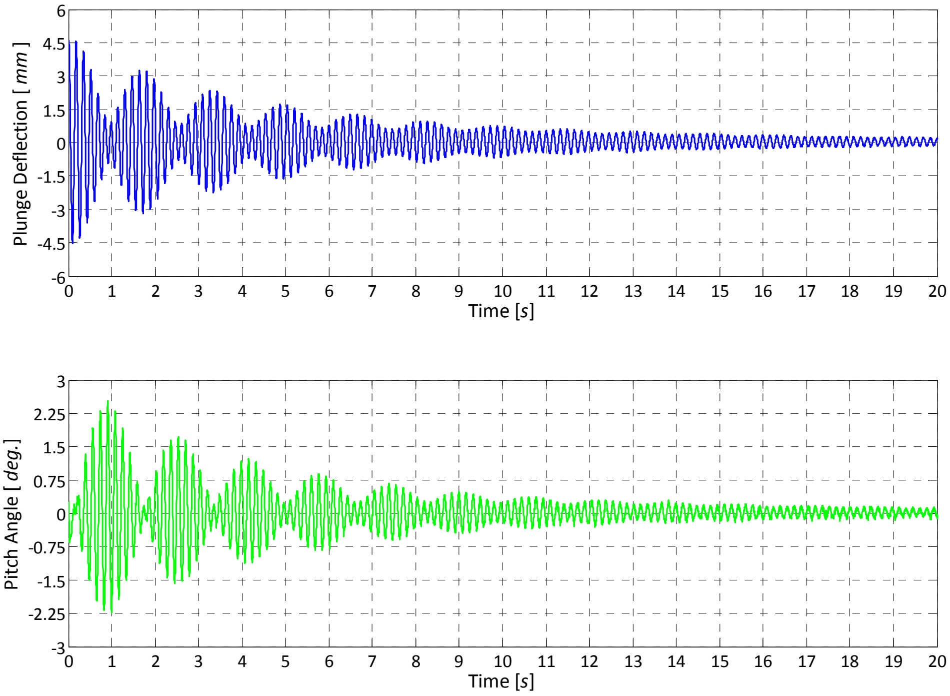


Figure G.5 FAST Model LSWT Open-Loop Time Response ($U = 0.0m/s$)

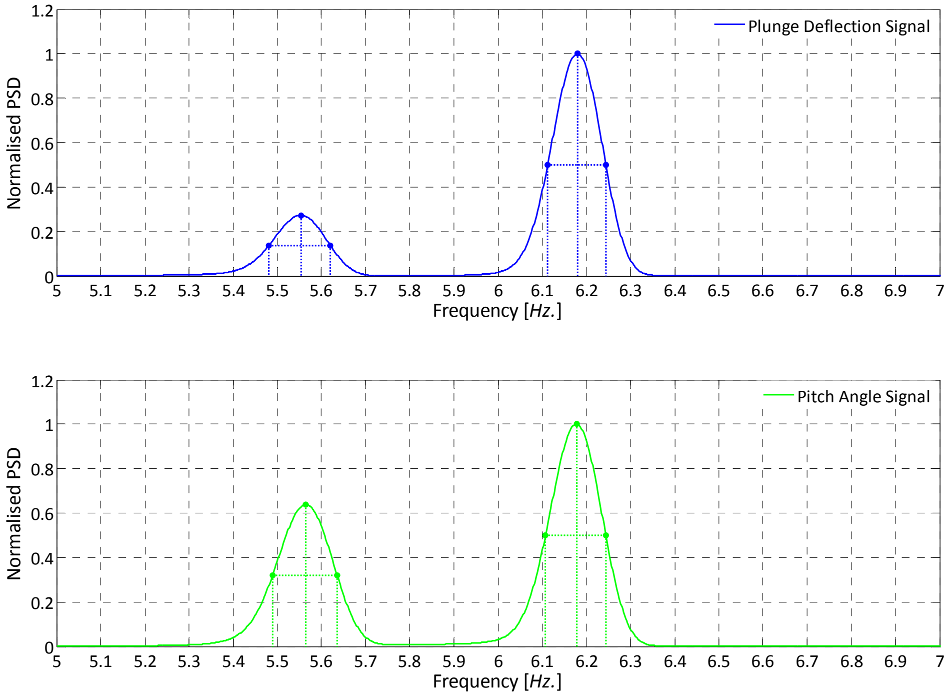


Figure G.6   Normalised PSD of LSWT Open-Loop Response (U = 0.0m/s)

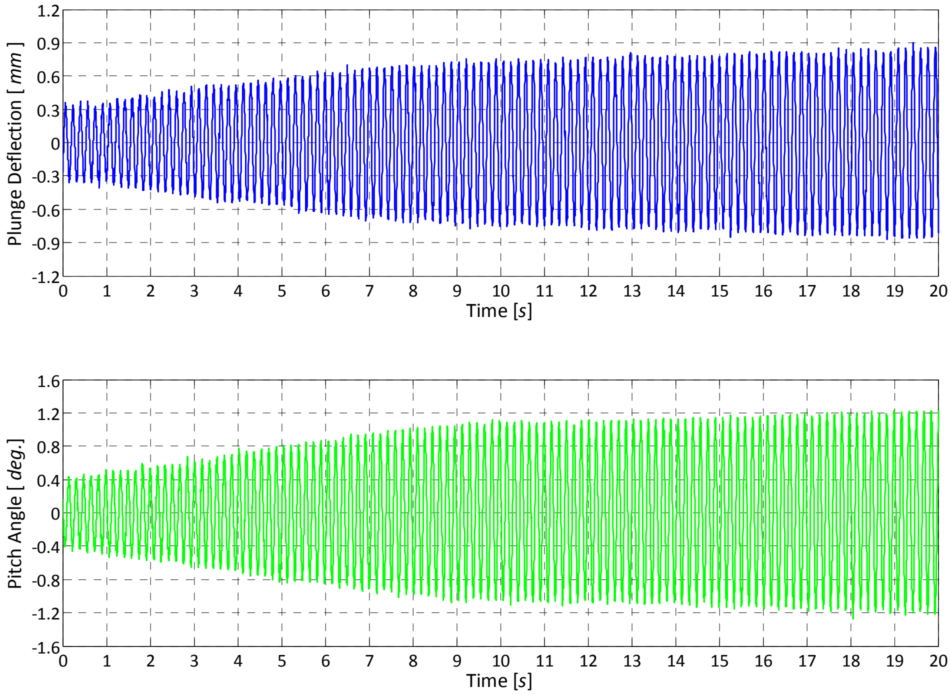


Figure G.7   FAST Model LSWT Open-Loop Flutter Time Response (U_f = 25.0m/s)

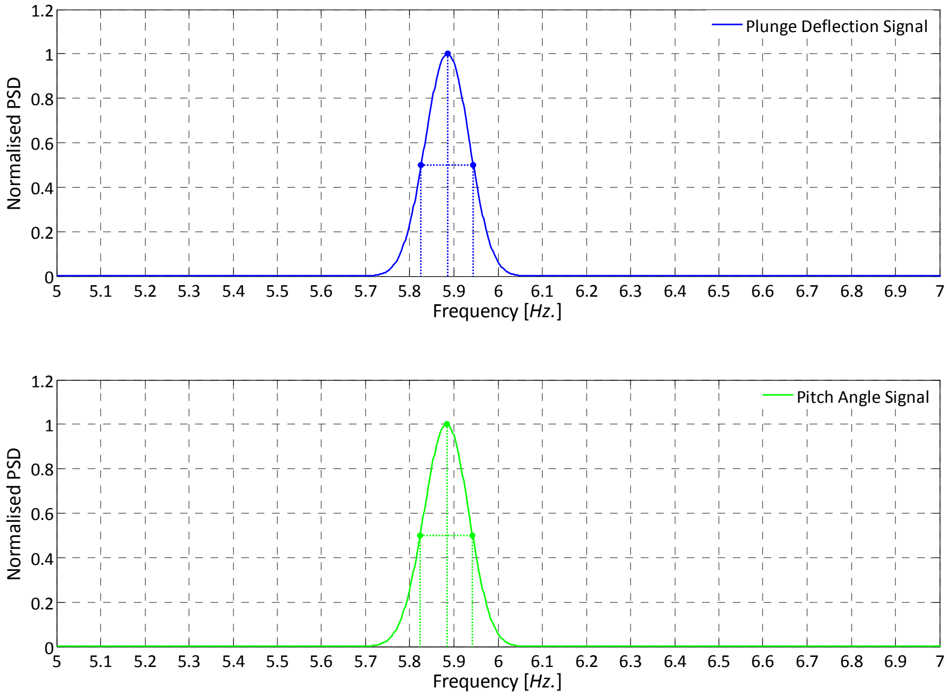


Figure G.8 Normalised PSD of LSWT Open-Loop Flutter (U_f = 25.0m/s)