EXPERIMENTAL AND ANALYTICAL TECHNIQUES FOR EXTRACTING THE AERODYNAMIC DAMPING-IN-PITCH PARAMETER FOR ORBITAL RE-ENTRY VEHICLES: AN INVESTIGATION OF STABILITY FOR SPACE CAPSULES

BY

Bradley Torgler

Submitted to the graduate degree program in Aerospace Engineering
and the Faculty of the Graduate School of the University of Kansas
In partial fulfillment of the requirements for the degree of
Master's of Science

_____
Chairperson

Committee Members _____

_____

Date defended: _____

The Thesis Committee for Bradley Torgler certifies
That this is the approved version of the following thesis:

EXPERIMENTAL AND ANALYTICAL TECHNIQUES FOR EXTRACTING THE AERODYNAMIC DAMPING-IN-PITCH PARAMETER FOR ORBITAL RE-ENTRY VEHICLES: AN INVESTIGATION OF STABILITY FOR SPACE CAPSULES

Committee:

_____
Chairperson

_____

_____

Date defended: _____

ABSTRACT

Aerodynamic damping is investigated in the subsonic, transonic and supersonic flight regimes for the Apollo and Expert orbital re-entry vehicles. Static moment and forced oscillation experiments are carried out on replica models in the von Karman Institute (VKI) S-1 Transonic-Supersonic wind tunnel. This is the first investigation to be done using the forced oscillation experimental technique in the VKI-S1. Oscillations are induced to match the reduced frequencies of flight vehicles based on their natural frequencies. A variety of mathematical models are developed to extract the aerodynamic damping-in-pitch parameter from the forced oscillation data. The results are compared to one another as well as to published data (where applicable). Free-to-tumble experimental data obtained for an Apollo model prior to this investigation is also evaluated using various mathematical models. Results show that classical analysis methods for extracting the aerodynamic damping-in-pitch parameter, which have been used many times prior to this investigation, continue to yield the most accurate measure for aerodynamic damping extraction. These include an outer-extremity envelope distribution method, which assumes a constant damping from one peak of pitch attitude to the next for free-to-tumble analysis and an energy conservation method based on hysteresis loops of pitching moment against pitch attitude for forced oscillation analysis.

TABLE OF CONTENTS

TABLE OF CONTENTS (CONT.)

TABLE OF CONTENTS (CONT.)

LIST OF FIGURES

LIST OF FIGURES (CONT.)

LIST OF FIGURES (CONT.)

LIST OF FIGURES (CONT.)

LIST OF TABLES

LIST OF SYMBOLS

Symbol	Description	Unit
Basic		
a	Speed of sound; or damping coefficient term	m/s; Hz
b	Torque balance slope; or stiffness coefficient term	N-m/V; $1/s^2$
C	Coefficient	~
C_m	Moment coefficient	~
$\left(C_{m_q} + C_{m_{\dot{\alpha}}}\right)$	Aerodynamic damping-in-pitch parameter	1/rad
$\partial C_m / \partial \theta$	Derivative of moment coefficient w.r.t. incidence angle (aerodynamic stiffness parameter)	1/rad
D	Diameter	m
f	Frequency	Hz
i	$\sqrt{-1}$	~
I	Moment of inertia about axis of rotation	kg-m^2
k	Mechanical stiffness	N-m/rad
L	Length	m
m	Mass	kg
M	Moment; or Mach number	N-m; ~
P	Pressure	N/m^2
q	Dynamic pressure	N/m^2
Re	Reynold's number	~
S	Reference area	m^2
St	Strouhal number	~
t	Time	s
T	Period; or temperature	s; K
u	Velocity in the x-direction	m/s
U	Velocity; or voltage	m/s; V
v	Velocity in the y-direction	m/s
V	Voltage	V
W	Work	N-m
x	x-direction; or axial direction	
y	y-direction; or moment direction	
z	z-direction; or normal direction	

LIST OF SYMBOLS (CONT.)

Symbol	Description	Unit
Greek		
α	Angle of attack	deg
γ	Flight path angle; or mechanical damping contribution	deg; kg-m^2/s
ϕ	Meridian plane angle	deg
μ	Phase angle	rad
ρ	Density	kg/m^3
θ	Oscillation angle; or pitch attitude angle	deg; deg
$\dot{\theta}$	Angular velocity	rad/s
$\ddot{\theta}$	Angular acceleration	rad/s^2
$\dddot{\theta}$	Angular jerk	rad/s^3
ω	Frequency	rad/s
ξ	Pitch aerodynamic damping	1/rad
Subscripts		
0	Offset; or where moment is zero	
1	Amplitude peak 1	
2	Amplitude peak 2	
f	Flow conditions	
ft	Flight vehicle	
m	Mean	
n	Natural (frequency)	
nf	No flow conditions	
o	Oscillation	
ref	Reference	
sys	System	
t	Total	
Superscripts		
–	Magnitude	
Acronyms		
AoA	Angle of attack	

LIST OF SYMBOLS (CONT.)

Symbol	Description	Unit
Acronyms (cont.)		
c.g.	Center of gravity	
CFD	Computational fluid dynamics	
DAQ	Data acquisition	
DoF	Degree of freedom	
FO	Forced oscillation	
FTT	Free-to-tumble	
LDV	Laser Doppler Velocimetry	
NASA	National Aeronautics and Space Administration	
PIV	Particle Image Velocimetry	
RMS	Root mean square	
SIO	Shock induced oscillation	
VKI	von Karman Institute for Fluid Dynamics	
w.r.t.	With respect to	

1. INTRODUCTION

1.1 Project Objective

Oscillatory motion is a dynamic phenomenon experienced by space capsules upon re-entry to the earth's atmosphere. This behavior needs to be well understood for specific geometries to avoid unstable flight. Proper characterization of aerodynamic damping for stability evaluation can allow drogue chute deployment at lower Mach number. Such findings may permit smaller drogue chute designs, thus enabling a payload volume and weight increase as well as a vast decrease in drag penalties. The nature of this project is to ascertain the dynamic stability characteristics of two orbital re-entry vehicles (the Apollo and Expert) by evaluating the aerodynamic damping-in-pitch parameter and static restoring moment characteristics for various flight regimes ranging from subsonic (Mach < 1) to supersonic speeds (at Mach ~ 2). The Apollo investigation is merely a test-bed to compare with published data for either validating or invalidating the forced oscillation experiment and data post-processing methods.

1.2 Experimental Overview

Experimental investigations are carried out on stereolithographic models (Figure 1 and Figure 2) using the forced oscillation (FO) technique about each models' trim angle of attack.

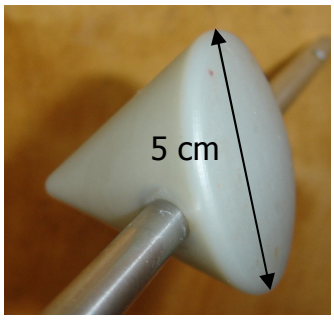

Figure 1 Apollo stereolithography model

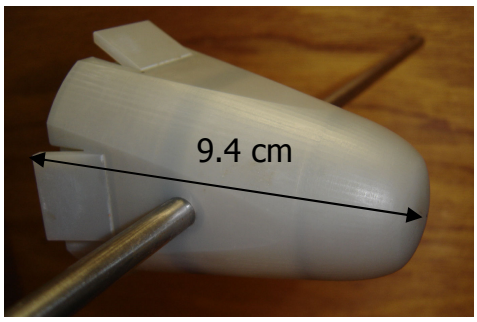

Figure 2 Expert stereolithography model

Static moments are also extracted over a range of incidence angles. Measurements are taken in the von Karman Institute (VKI) S1 Transonic-Supersonic wind tunnel (Figure 3).

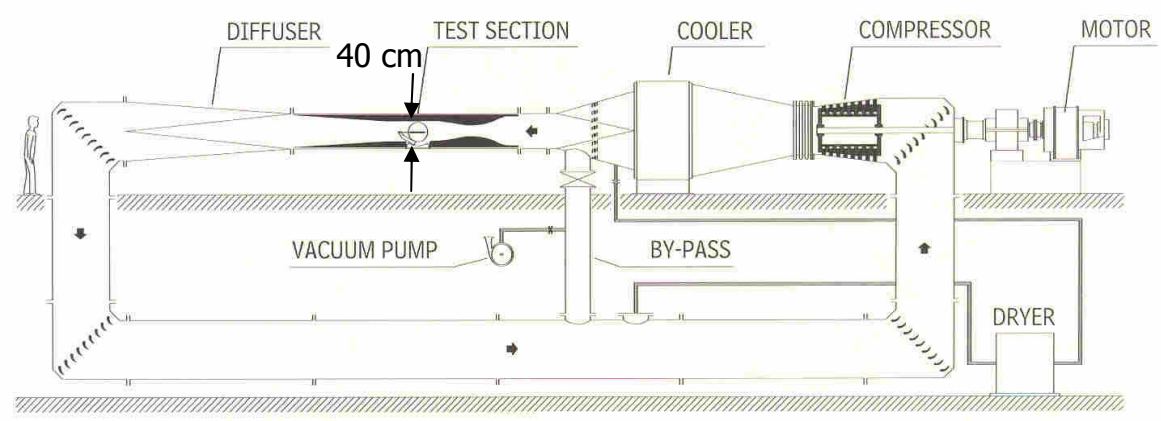

Figure 3 VKI-S1 wind tunnel schematic [22]

The VKI-S1 wind tunnel is a continuous closed circuit super/trans/sub-sonic wind tunnel of the Ackeret type and driven by a 615 kW axial flow compressor. Three 40 cm x 36 cm test sections are available with contoured nozzles capable of setting flows from the subsonic to transonic regime (Mach 0.4 to 1.05) and others for setting flow in the supersonic regime (Mach 2 to 2.25). A typical unit Reynold's number is 4×10^6/m [21].

Each model is glued to a steel, lateral support axis rod and mounted spanning the wind tunnel test section side walls. Actual flight vehicle trajectory data is used to determine wind tunnel conditions to retain Mach number relations between the flight vehicles and models. Pressure transducer calibrations are performed to accurately deduce and adjust wind tunnel conditions during testing, accordingly. Oscillations of approximately ±5 degrees in magnitude are induced at frequencies chosen to preserve the flight vehicle reduced frequencies (Strouhal numbers) using a motor and crankshaft system. Static moments are obtained at varying incidences ranging from approximately -20 to +20 degrees for the Expert model and approximately +135 to +225 degrees for the Apollo.

Two separate balance systems are calibrated: one for aerodynamic static testing and the other for aerodynamic static moment with forced oscillation testing. A two-component (axial and normal force) and three-component (axial, normal and pitching moment forces) strain gage-with-torque balance are placed in a mock test section set-up and mounted on the walls opposite one another for aerodynamic static testing calibrations. The static test calibrations will be utilized to obtain aerodynamic loads in future experimental investigation not covered under the scope of this project. Results will also be used in analysis of future experiments using the free-to-tumble (FTT)

technique. The second balance system is a strain gage torque balance internally housed in a shielded portion of the axis rod near one side of the wind tunnel walls for forced oscillation dynamic testing. The torque balance is also used for aerodynamic static moment testing. Moments experienced by the models are translated to the balance system, which is connected to a digital data acquisition system. An encoder is mounted externally to the wind tunnel to measure the angle of attack of each model and is also connected to the digital data acquisition system.

The experiment is divided into three parts:

- Calibrations – static balance system (for FTT), torque balance (for FO), and pressure transducers
- Experiment – static pitching moment tests and forced oscillation tests
- Post-processing – using forced oscillation test data and previously acquired free-to-tumble test data to extract the aerodynamic damping-in-pitch parameter

1.3 Analytical Overview

Analytical post-processing methods are developed and used to extract the aerodynamic damping-in-pitch parameter for the forced oscillation technique. In addition to post-processing the data from the forced oscillation experiments, analytical methods are also developed for the free-to-tumble experimental method. Though this method is not performed experimentally for this project, previous data was obtained at the VKI using the free-to-tumble method by Paris and Charbonnier, [7] and Emilie, [17]. An analytical method to determine the aerodynamic damping-in-pitch parameter using this technique had yet to be successfully developed at the VKI, thus the task of determining such a method fell under the responsibilities of this project. Assuring that Mach and Strouhal number relations are maintained allows for accurate comparison of results between the flight vehicle re-entry data with the model experimental data where applicable.

1.4 Related Works

Prior folk have put many hours into investigation of aerodynamic damping characteristics for orbital re-entry vehicles such as Karatekin, [1], Hanff, [10], Paris et. al, [7] and others. This work is in part an ongoing investigation of the research carried out by Karatekin, [1] as well as Underhill, [16] and Emilie, [17]. All investigate oscillatory behavior of space capsules as they relate to aerodynamic influences on damping and trajectory behavior.

Karatekin's research was conducted at the von Karman Institute and lead to a doctorate in philosophy from the Université Libre de Bruxelles. The concentration of his work was to better understand dynamic instabilities that act on orbital re-entry vehicles at low, subsonic speeds. His approach focused on examination of unsteady flow field effects on dynamic stability for the Apollo. The research carried out included characterization of aerodynamic static and dynamic stability, and then performing experiments based on these characterizations. Mean and unsteady flow field surface pressure measurements on an Apollo model were performed in the VKI-L1 wind tunnel in conjunction with LDV and PIV to study the mean and instantaneous velocity fields over the model and in the wake. Surface pressure measurements and moments were obtained for forced oscillations at free stream flows of 30 m/s. Energy exchange between the flow field and the model was investigated by analyzing moment hysteresis. He found that the main reason for dynamically unstable behavior is due to time-lag effects on surface pressure variation. Thus, even for stable regions of angle of attack, a divergent oscillatory motion (due to negative damping) could be created due to these time-lags.

Emilie's research was conducted at the VKI as well. The objective of her experiment was to determine stability characteristics and aerodynamic damping of the Apollo using the free-to-tumble technique. Tests were conducted at Mach 2. A Runge Kutta approach was used to curve fit data results to extract aerodynamic damping, but lead to no conclusive results except for steady state angles of attack and phase plane plots for different release angles of the model.

Underhill's research, conducted at the VKI, was done in part to develop a methodology for the forced oscillation technique in the VKI-S1 wind tunnel and to evaluate the limitations and behaviors of the equipment that are associated with the methodology developed. She cultivated a method that uses a motor with driver and implemented this on a mock test section ("dummy tunnel") of the VKI-S1 test section. A literature survey was made to determine the required capabilities of motor frequency and balance system maximum allowables. Based on her findings and on the system developed, she concluded that mechanical friction of the system could "unduly affect measurements, especially for small angles" [16] and that vibrations of the motor at high rpm (over 1000) may affect angular measurements.

2. THEORY

The equations of motion of an object in flight have been characterized using various methods such as Euler's body-attached frame of reference and quaternion frame of reference rotation approaches. Rather than an in-depth re-derivation of motion analysis, this section will concentrate on the motion of the experimental system only and its use for deriving analytical methods for data post-processing. Further information on the kinematics of flight can be found in Karatekin, [1], Etkin, [2] and Loh, [3].

2.1 Motion of the System

Both the forced oscillation and free-to-tumble experimental set-ups contain only one rotational degree of freedom (DoF) and no translational degrees of freedom.

2.1.1 Governing Physics

Due to the system DoF constraints, the motion of the model can be simplified to two-dimensional, planar motion analysis. This simplification is also performed for actual flight vehicles and outlined in Karatekin, [1] and Chapman, [8].

The general coordinate system for the simplified planar motion is depicted for the Apollo and given in Figure 4. The same coordinate system is also applicable for the Expert vehicle.

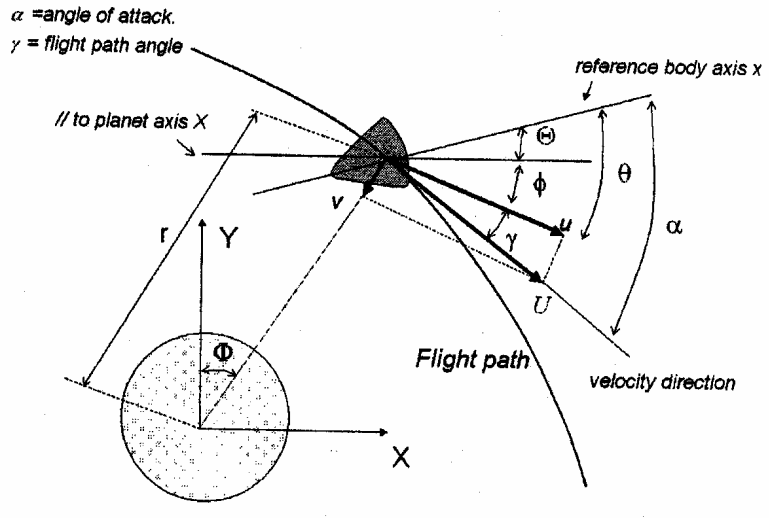

Figure 4 Planar motion coordinate system [1]

The simplified equation for only one degree of freedom in rotation for planar motion is given in Equation 1.

$$I\ddot{\theta} = C_m q_\infty SD + C_{m_q} q_\infty SD \frac{D\dot{\theta}}{U_\infty} + C_{m_{\dot{\alpha}}} q_\infty SD \frac{D\dot{\alpha}}{U_\infty} \qquad \textbf{Equation 1}$$

where: I is the vehicle moment of inertia about the axis of rotation
θ is the of pitch attitude angle
α is the angle of attack
C_m is the pitch moment coefficient
q_∞ is the free stream dynamic pressure
S is the vehicle reference area
D is the vehicle characteristic length
U_∞ is the total free stream velocity
$\dot{}$ is the derivative with respect to time, (d/dt)

$$C_{m_q} = \frac{\partial C_m}{\partial(\dot{\theta}D/U_\infty)}, \qquad C_{m_{\dot{\alpha}}} = \frac{\partial C_m}{\partial(\dot{\alpha}D/U_\infty)},$$

Recall that Equation 1 is simplified for an actual vehicle in flight. For forced oscillation and free-to-tumble testing, the inertia, I, includes the model inertia as well as the lateral support axis upon which the model rests and any other component coupled with the entire rotating system. Also, for wind tunnel tests, the pitch attitude angle is equal to the angle of attack (meaning no flight path angle). Thus, Equation 1 for can be re-written to yield Equation 2.

$$I_{sys}\ddot{\alpha} - q_\infty SD\left[\left(C_{m_q} + C_{m_{\dot{\alpha}}}\right)\frac{D}{2U_\infty}\dot{\alpha} + \frac{\partial C_m}{\partial \alpha}\left(\alpha - \alpha_0\right)\right] = M(t) \qquad \textbf{Equation 2}$$

where: I_{sys} is the moment of inertia of the entire rotating system
α_0 is the trim angle of attack (C_m is zero)
$M(t)$ is an external forcing moment as a function of time

A moment forcing function, $M(t)$ is introduced in Equation 2, and it is not in Equation 1. This is due to the experimental nature of the investigation. The only assumed forces come from aerodynamic loads with no external forces acting on the vehicle during actual flight. This is not the case for the forced oscillation technique; thus, the added forcing moment function, $M(t)$.

Although there is a small amount of friction in the system between the lateral support axis and the ball-bearing supports for the free-to-tumble technique, it is assumed negligible; and the same assumption of no externally added forces is implemented.

To sum up: $M(t) = M(t)$ for FO technique $M(t) = 0$ for FTT technique

Another term introduced in Equation 2 that is not shown in Equation 1 is the trim angle of attack, α_0. The value of C_m is replaced by $\frac{\partial C_m}{\partial \alpha}(\alpha - \alpha_0)$ because the two are equivalent and the slope $\frac{\partial C_m}{\partial \alpha}$ is determined from static moment testing. By redefining θ as the oscillation angle in Equation 3 and letting $I_{sys} = I$, the final form of the experimental system (both FO and FTT) equation of motion is given in Equation 4.

$$\theta = \alpha - \alpha_m \qquad \textbf{Equation 3}$$

where: α_m is the mean angle of attack about which dynamic oscillation occurs and $\alpha_m = \alpha_0$ when oscillating about the trim angle of attack

$$I\ddot{\theta} - \left(C_{m_q} + C_{m_{\dot{\alpha}}}\right)\frac{q_\infty SD^2}{2U_\infty}\dot{\theta} - q_\infty SD\frac{\partial C_m}{\partial \theta}\theta = M(t) \qquad \textbf{Equation 4}$$

The benefit of Equation 4 is that its general form is that of a 2nd order ordinary differential equation 'mass-spring-damper' type. There exists a theoretical solution to such types of equations, and it is the starting point for the processing methods for the experimental techniques.

2.1.2 Non-Dimensional Parameters

It is important to match certain non-dimensional parameters to ensure equivalent wind tunnel data relations between model and true flight vehicle geometries. Besides an accurate geometric scaling between the model and flight vehicle (this includes point of rotation location: center of gravity {vehicle} = rotation point {model}), tests should be made at the same Mach number and Reynold's number when possible as well as at reduced frequency number when oscillations are forced.

2.1.2.1 Mach Number and Reynold's Number

The Mach number, M, is the ratio of free stream velocity to the speed of sound (Equation 5). More importantly, for this investigation where extracting aerodynamic damping is the project objective, it is the ratio of inertial (aerodynamic) forces to elastic forces.

$$M = \frac{U_\infty}{a} \qquad \textbf{Equation 5}$$

where: a is the speed of sound in the fluid medium and a function of temperature

It is very important that wind tunnel tests preserve this number for models to accurately compare with flight vehicles, especially in the supersonic regime due to the higher ratio of aerodynamic forces to elastic forces in the fluid medium. The transonic regime is also important, but the reduced ratio in aerodynamic-to-elastic forces and unpredictable behavior of shock induced oscillations (SIO) that generally occur leaves room for slightly more bandwidth in Mach number matching. The same (excluding SIO) is the case for the subsonic flight regime.

The Reynold's number (Re) is the ratio of inertial forces to viscous forces (Equation 6).

$$\mathrm{Re}_x = \frac{\rho_\infty U_\infty x}{\mu_\infty} \qquad \textbf{Equation 6}$$

where: x is the characteristic length of the vehicle
ρ is the density of the fluid medium
μ is the fluid dynamic viscosity and a function of temperature

This parameter is much more difficult to match the model with the actual flight vehicle for forced oscillation testing. This is not only due to the reduced size of the model relative to the flight vehicle, but also due to the superseding importance to match reduced frequencies and Mach number during tests. All three non-dimensional numbers are defined using the free stream velocity, U_∞ and Re_x and M are indirect functions of temperature. Thus, changing one would change another. Also, the typical Reynold's number in the VKI-S1 facility is 4×10^6/m. There is no means to preserve flight vehicle Reynold's numbers unless the air is super-cooled given the characteristic

lengths of the models being tested. This is not a capability of the VKI-S1 facility. Although the Reynold's number cannot be preserved between the flight vehicle and the testing model for this investigation, it is important to attempt a similarity in flow regime (laminar or turbulent) where transition to turbulence is typically around a Reynold's number of the order 10^5 or 10^6.

2.1.2.2 Reduced Frequency (Strouhal Number)

The reduced frequencies of both the model and the flight vehicle for the forced oscillation tests should be maintained to equate the dissipation or input of energy from the surrounding flow field (aerodynamic influences) on the model during testing with the flight vehicle. It is defined by the Strouhal number (St); this number describes the frequency of vortex shedding. Oscillations occur at the natural frequencies of the actual flight vehicle upon atmospheric re-entry [1] (includes the range of flight regime investigated). Thus, during forced oscillation tests on the models, the Strouhal number based on the natural frequency and characteristic length of the flight vehicle is used. The general formulation is given in Equation 7.

$$St = \frac{fD}{U_\infty} \qquad \textbf{Equation 7}$$

where: f is the frequency in Hz

Extraction of the natural frequency of the flight vehicle, ω_n, is obtained from the 'spring' constant and 'damping' constant in Equation 4.

$$(\omega_n)_{ft} = \left(\sqrt{\frac{q_\infty SD\left|\left(\partial C_m / \partial \theta\right)\right|}{I} - \left(\frac{q_\infty SD(C_{m_q} + C_{m_\alpha})}{2I} \right)^2} \right)_{ft} \qquad \textbf{Equation 8}$$

where: ω_n is the natural frequency in rad/s

The aerodynamic damping (right end term of Equation 8) can be neglected due to its magnitude relative to the static restoring moment contribution (left end term of Equation 8). A system is statically stable only when $\partial C_m / \partial \theta$ is negative. When $\partial C_m / \partial \theta > 0$, then the system is statically unstable and

oscillation magnitudes grow over time. Thus, the Strouhal number of the flight vehicle is given (Equation 9).

$$(St)_{ft} = \left(\frac{\omega_n D}{2\pi U_\infty}\right)_{ft} = \left(\sqrt{\frac{\rho_\infty S D^3 \left|\left(\partial C_m / \partial \theta\right)\right|}{8\pi^2 I}}\right)_{ft}$$ **Equation 9**

2.2 Free-to-Tumble Technique

This method exploits free range of rotational motion about an axis (1 DoF). A model is held fixed until flow conditions in the wind tunnel are set, and is then released to freely tumble. The model incidence angles are recorded over time.

2.2.1 Analytical Processing Methods

Equation 4 is re-written to incorporate the free-to-tumble technique and yields Equation 10.

$$\ddot{\theta} + a\dot{\theta} + b\theta = 0$$ **Equation 10**

where: $a = -\left(C_{m_q} + C_{m_{\dot{\alpha}}}\right)\frac{q_\infty S D^2}{2U_\infty I}$; $b = -\frac{q_\infty S D}{I}\frac{\partial C_m}{\partial \theta}$

The objective of this project is to ascertain the aerodynamic damping-in-pitch parameter, $\left(C_{m_q} + C_{m_{\dot{\alpha}}}\right)$, for the models investigated. The following methods use solution techniques to solve Equation 10, and from the solution, extract the aerodynamic damping-in-pitch parameter. Note that the coefficients a and b are generally constant for a well-posed 2nd order system, but for this case, both a and b are functions of incidence (oscillation) angle, θ.

$a = f(\theta)$; $b = g(\theta)$

Thus, the system becomes highly coupled and non-linear. The only unknown of the system from the test data is the a coefficient. All flow conditions and geometry are known. The derivatives of incidence with respect to time are found using central difference schemes. Analytical results using the methods in Section 2.2.1 can be found in Section 4.2.

2.2.1.1 Algebraic Method

This is the most basic method. A Matlab function *func_ftt_algebra.m* is written for this method (given in the appendix). Since the only unknown is a, which contains the aerodynamic damping-in-pitch parameter, Equation 10 can be re-arranged to yield Equation 11.

$$a = -\frac{\left(\ddot{\theta} + b\theta\right)}{\dot{\theta}} \qquad \textbf{Equation 11}$$

With all else known, the aerodynamic damping coefficient can be extracted.

2.2.1.2 Envelope Method

This method is an adaptation of one of the older approaches used for aerodynamic damping investigation from oscillatory motion of flight vehicles. A Matlab function *func_ftt_envelope.m* is written for this method (given in the appendix). Mention or practice of this method is given in Redd et al., [9] and others. This method assumes that neither a or b is known and that the only known variables are $\ddot{\theta}, \dot{\theta}, \theta$. It also assumes that both a and b are constant over one full oscillation in the oscillation data signal.

The general solution to Equation 10 is given by the following:

$$\theta(t) = e^{\sigma t}\left(C_1 \cos(\omega t) + C_2 \sin(\omega t)\right) \qquad \textbf{Equation 12}$$

$$\dot{\theta}(t) = \sigma\theta + \omega e^{\sigma t}\left(C_2 \cos(\omega t) - C_1 \sin(\omega t)\right) \qquad \textbf{Equation 13}$$

$$\ddot{\theta}(t) = 2\sigma\dot{\theta} - \theta\left(\omega^2 + \sigma^2\right) \qquad \textbf{Equation 14}$$

where: C_1, C_2 are 'constant' coefficients based on initial and boundary conditions

$a = -2\sigma$; $b = \left(\sigma^2 + \omega^2\right)$

Based on the definition of symbols from Figure 5, a and b from one oscillation period to the next can be extracted over the entire data signal.

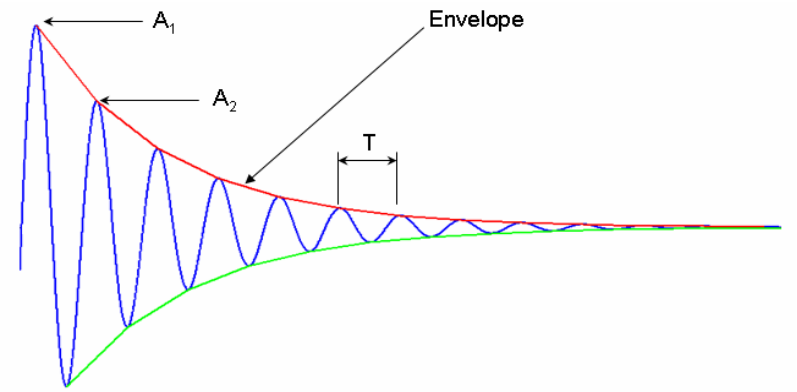

Figure 5 Symbolic definitions for envelope method

$$a_{1\to 2} = \frac{2}{T_{1\to 2}} \ln \frac{A_1}{A_2} \qquad \textbf{Equation 15}$$

$$b_{1\to 2} = \left(\frac{2\pi}{T_{1\to 2}}\right)^2 + \left(\frac{a_{1\to 2}}{2}\right)^2 \qquad \textbf{Equation 16}$$

2.2.1.3 Linearized Algebra Method

This method branches off the equation solutions of the envelope method, but has no assumptions except that the general solution is valid. A Matlab function *func_ftt_lin_alg.m* is written for this method (given in the appendix). The known variables are $\ddot{\theta}, \dot{\theta}, \theta$ for this method. In addition to Equation 12 through Equation 14, Equation 17 is used.

$$\dddot{\theta}(t) = 2\sigma\ddot{\theta} - \dot{\theta}\left(\omega^2 + \sigma^2\right) \qquad \textbf{Equation 17}$$

From the system of the four equations mentioned, it can be seen that only Equation 14 and Equation 17 are required to define both a and b. An algebraic system combining the two equations is given in Equation 18.

$$\begin{bmatrix} \ddot{\theta} \\ \dddot{\theta} \end{bmatrix} = \begin{bmatrix} 2\dot{\theta} & -\theta \\ 2\ddot{\theta} & -\dot{\theta} \end{bmatrix} \begin{Bmatrix} \sigma \\ \left(\omega^2 + \sigma^2\right) \end{Bmatrix} = \begin{bmatrix} -\dot{\theta} & -\theta \\ -\ddot{\theta} & -\dot{\theta} \end{bmatrix} \begin{Bmatrix} a \\ b \end{Bmatrix} \qquad \textbf{Equation 18}$$

Using matrix algebra inversion methods for Equation 18, the aerodynamic damping-in-pitch parameter can be evaluated for each incidence angle. Thus, $a = f(\theta)$ is met.

2.3 Forced Oscillation Technique

Model vehicles are oscillated about an axis of rotation over a fixed range of oscillation angles in a periodic pattern. The forced oscillations induce moments on the model in addition to the aerodynamic loads. Moments and incidence angles are recorded over time.

2.3.1 Analytical Processing Methods

The resulting moments recorded during forced oscillation tests embodies all moments acting on the model [1], [10] and [11]. This includes:

- Aerodynamic moments
  - static restoring moments (3rd term of Equation 4)
  - moment hysteresis from oscillatory motion and unsteady flow field effects (related to energy transfer from model to surrounding flow and vice-versa, corresponding to aerodynamic damping (2nd term of Equation 4))
- Mechanical system moments
  - torsion spring and damping resistive moments from the forced oscillation torque balance
- Forced moments
  - forced moments incurred due to forced oscillations

The complete angular differential equation of motion for the forced oscillations is expressed in Equation 19.

$$I\ddot{\theta} = M \qquad \textbf{Equation 19}$$

where: M represents the total moments acting on the model

2.3.1.1 Algebraic Method

This method extracts the aerodynamic damping by subtracting resultant total moments of induce forced oscillations under no flow (nf) conditions from the resultant total moments of the induced forced oscillation data under flow (f) conditions for the same mean angle of attack. A Matlab function *func_fo_algebra.m* is written for this method (given in the appendix). No flow conditions yield mechanical and forced total moments whereas flow conditions yield the mechanical and forced moments as well as the aerodynamic moments. Hence, the aerodynamic restoring moment contributions are completely separated.

$$\left(C_{m_q} + C_{m_{\dot\alpha}}\right)\frac{q_\infty SD^2}{2U_\infty}\dot\theta_f + q_\infty SD\frac{\partial C_m}{\partial\theta}\theta_f = \left(I\ddot\theta = M\right)_f - \left(I\ddot\theta = M\right)_{nf} \quad \textbf{Equation 20}$$

The only remaining unknown in Equation 20 following static moment testing (to extract $\frac{\partial C_m}{\partial\theta}$) and dynamic forced oscillation testing (to extract all flow and no flow conditions and data) is the aerodynamic damping-in-pitch parameter. Note that in this method, there are two possible solution directions/methods: either $M_f - M_{nf}$ {'moment'} or $I(\ddot\theta_f - \ddot\theta_{nf})$ {'inertia'} could be used and both are investigated in this project.

2.3.1.2 Phase-Difference Amplitude Method

This method is derived from Hanff, [10]. A Matlab function *func_fo_phase_diff_amp.m* is written for this method (given in the appendix). The total moments are separated into their respective parts in Equation 21 through Equation 23.

$$I\ddot\theta + c\dot\theta + K\theta = M(t) \quad \textbf{Equation 21}$$

where: $M(t)$ is the moment forcing function

$$c = \gamma - M_{\dot\theta} \quad \textbf{Equation 22}$$

where: γ is the mechanical moment damping contribution

$M_{\dot\theta}$ is the aerodynamic moment damping-in-pitch contribution

$$\left(= \left(C_{m_q} + C_{m_{\dot\alpha}}\right)\frac{q_\infty SD^2}{2U_\infty}\right)$$

$$K = k - M_\theta \quad \textbf{Equation 23}$$

where: k is the mechanical stiffness

M_θ is the aerodynamic stiffness-in-pitch $\left(= q_\infty SD\frac{\partial C_m}{\partial\theta}\right)$

It is assumed with this method that under steady-state conditions, the forced oscillation angle can be written as a pure sinusoidal function and the moment

forcing function also as a sinusoidal function with a phase shift and an additional unsteady flow component.

$$\theta = \bar{\theta} \sin \omega t \qquad \textbf{Equation 24}$$

where: $\bar{\theta}$ is the oscillation angle amplitude

$$M(t) = \bar{M} \sin(\omega t + \mu) + n(t) \qquad \textbf{Equation 25}$$

where: $\bar{M}$ is the forcing function moment amplitude
μ is phase angle between the forcing moment and oscillation angle
$n(t)$ is the moment contribution from unsteady flow

By assuming the RMS value of the random moment contribution from unsteady flow is zero, it is disregarded and Equation 21 is written in vectorial form.

$$I\omega^2 \bar{\theta} e^{i\omega t} + i\omega c \bar{\theta} e^{i\omega t} + K\bar{\theta} e^{i\omega t} = \bar{M} e^{i(\omega t + \mu)} \qquad \textbf{Equation 26}$$

where: i is the imaginary unit number ($= \sqrt{-1}$)

Dividing through by a common factor on both sides of Equation 26, the equations for flow and no flow conditions are obtained.

$$-I\omega_{nf}{}^2 \bar{\theta}_{nf} + i\omega_{nf} \gamma \bar{\theta}_{nf} + k\bar{\theta}_{nf} = \bar{M}_{nf} e^{i\mu_{nf}} \qquad \textbf{Equation 27}$$

$$-I\omega_f{}^2 \bar{\theta}_f + i\omega_f \left(\gamma - M_{\dot{\theta}}\right)\bar{\theta}_f + \left(k - M_\theta\right)\bar{\theta}_f = \bar{M}_f e^{i\mu_f} \qquad \textbf{Equation 28}$$

The aerodynamic stiffness in pitch is extracted from the real parts of Equation 27 and Equation 28.

$$-I\omega_{nf}{}^2 \bar{\theta}_{nf} + k\bar{\theta}_{nf} = \bar{M}_{nf} \cos(\mu_{nf}) \qquad \textbf{Equation 29}$$

$$-I\omega_f{}^2 \bar{\theta}_f + \left(k - M_\theta\right)\bar{\theta}_f = \bar{M}_f \cos(\mu_f) \qquad \textbf{Equation 30}$$

Substitution of k from Equation 29 into Equation 30, M_θ is found.

$$M_\theta = \frac{\overline{M}_{nf}\cos(\mu_{nf})}{\overline{\theta}_{nf}} - \frac{\overline{M}_f\cos(\mu_f)}{\overline{\theta}_f} + I\left(\omega_{nf}{}^2 - \omega_f{}^2\right) \quad \textbf{Equation 31}$$

In contrast to using the real parts of Equation 27 and Equation 28, the imaginary parts are used to extract the aerodynamic moment damping-in-pitch contribution.

$$\omega_{nf}\gamma\overline{\theta}_{nf} = \overline{M}_{nf}\sin(\mu_{nf}) \quad \textbf{Equation 32}$$

$$\omega_f\left(\gamma - M_{\dot{\theta}}\right)\overline{\theta}_f = \overline{M}_f\sin(\mu_f) \quad \textbf{Equation 33}$$

Again, using substitution, Equation 34 is obtained.

$$M_{\dot{\theta}} = \frac{\overline{M}_{nf}\sin(\mu_{nf})}{\omega_{nf}\overline{\theta}_{nf}} - \frac{\overline{M}_f\sin(\mu_f)}{\omega_f\overline{\theta}_f} \quad \textbf{Equation 34}$$

For this method discussed thus far, the amplitude of the forcing moment in no flow conditions, $\overline{M}_{nf}$, must be known. This means that the contribution of moments from the mechanical system (torsion spring and damping resistive moments from the forced oscillation torque balance) are known prior to testing. The experiments carried out in this project do not resolve the separation of the two. Thus, to use this method as it stands, the theoretical forcing function set at the same testing frequency for no flow conditions and same amplitude under the no flow conditions is determined and used. Determining the function is based on the mechanical set-up for induced oscillations and the inertial characteristics of the rotating system (model with support axis). This sub-method is later referred to as {'unlumped'}.

An alternate to using the theoretical forcing function is to simply use the no flow condition results and lump the mechanical system moment contributions with the forcing moment function contributions. With this alternate scheme and using the same approach that was used to resolve Equation 31 and Equation 34 from Equation 26, the following equations result. This sub-method is later referred to as {'lumped'}.

$$M_\theta = \left(\omega_f{}^2\right)\frac{\overline{M}_{nf}\cos(\mu_{nf})}{\omega_{nf}{}^2\overline{\theta}_{nf}} - \frac{\overline{M}_f\cos(\mu_f)}{\overline{\theta}_f} \quad \textbf{Equation 35}$$

$$M_{\dot{\theta}} = -\frac{\overline{M}_f\sin(\mu_f)}{\omega_f\overline{\theta}_f} \quad \textbf{Equation 36}$$

2.3.1.3 Moment Hysteresis-Energy Method

This method is based on Equation 19 and is adapted from Tang et al., [4] and Lan et al., [5]. A Matlab function *func_fo_energy.m* is written for this method (given in the appendix). The exchange of energy for a rotational, single degree of freedom system is represented by the moment hysteresis relative to oscillation angle. Figure 6 depicts the energy exchange for one period of oscillation.

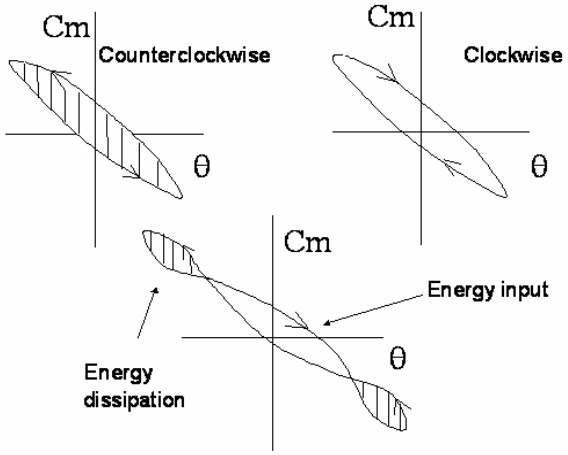

Figure 6 Moment hysteresis showing energy exchange over one period of oscillation [1]

The energy exchange is a measure for aerodynamic damping and thus a measure of local dynamic stability. It is shown that energy is inputted to the system for a clockwise path in the moment hysteresis loop. The added energy to the system is a result of added moment in the direction of motion. Thus, overall damping is negative, yielding a positive result for the aerodynamic damping-in-pitch parameter. Likewise, for a counter-clockwise path, energy is dissipated resulting in positive damping of the system, and a negative value for damping-in-pitch parameter. A triple-loop in moment hysteresis represents a system undergoing limit cycle oscillations. Summing up the instantaneous product of angular velocity with applied torque over one period yields the work done by the system.

$$W = \int_0^T I\ddot{\theta}\dot{\theta}dt \qquad \textbf{Equation 37}$$

where: T is the period of oscillation

Substituting Equation 19 and non-dimensionalizing the work by dividing by ($q_\infty SD$), the following relation is made.

$$W_{ND} = \int_0^T C_m \dot{\theta} dt = \int_{\theta_0}^{2\pi+\theta_0} C_m d\theta \qquad \textbf{Equation 38}$$

where: θ_0 is the lower bound oscillation angle chosen for integration

In order to maintain energy balance, the work done by the system added to the restoring aerodynamic work must be zero. The static restoring moments yield no contribution to the aerodynamic restoring work. Thus, all restoring work is found from aerodynamic damping and the following relation is made.

$$W = -W_{damping} \qquad \textbf{Equation 39}$$

The following excerpt comes from Tang et al., [4]:

> "A quantity that represents the degree of hysteresis is given by the pitch aerodynamic damping, ξ. … The pitch aerodynamic damping term represents the transfer of energy between the … pitching motion and the surrounding unsteady flow environment. When ξ is positive, the airstream makes the … pitch motion stable. Conversely, a negative value of ξ represents unstable motion."

Tang defines the pitch aerodynamic damping as

$$\xi = -\oint C_m d\theta \,/\, \pi \bar{\theta}^2 \qquad \textbf{Equation 40}$$

This definition can be directly correlated with the general solution of Equation 10, where $a = 2\xi\omega_n$ and $b = \omega_n{}^2$. Therefore, the following correlation can be made for the pitch aerodynamic damping, ξ, and the aerodynamic damping-in-pitch parameter, $\left(C_{m_q} + C_{m_{\dot{\alpha}}}\right)$.

$$\left(C_{m_q} + C_{m_{\dot{\alpha}}}\right) = -\frac{2U_\infty I}{q_\infty SD^2}\left(2\xi\sqrt{\frac{q_\infty SD\left|\left(\partial C_m / \partial\theta\right)\right|}{I}}\right) \qquad \textbf{Equation 41}$$

3. EXPERIMENT

Experiments are carried out on the Apollo and Expert stereolithography models using the forced oscillation technique. Balance systems are calibrated for the forced oscillation investigation at hand and for future investigations to obtain the basic aerodynamic response properties of the vehicles as well as free-to-tumble results. The investigation is in response to acquiring forced oscillation data for meeting the requirements of the RFQ 3-11702/06/NL/PA proposal [12].

3.1 Static Test Balance System Calibration

The calibration results for this balance system will be used in future investigations (aerodynamic force and moment measurements and free-to-tumble measurements).

3.1.1 Apparatus and Set-Up

The bulk of mechanical equipment used for the calibration is shown in Figure 7. It is set-up in a replica test section of the VKI-S1 wind tunnel test section.

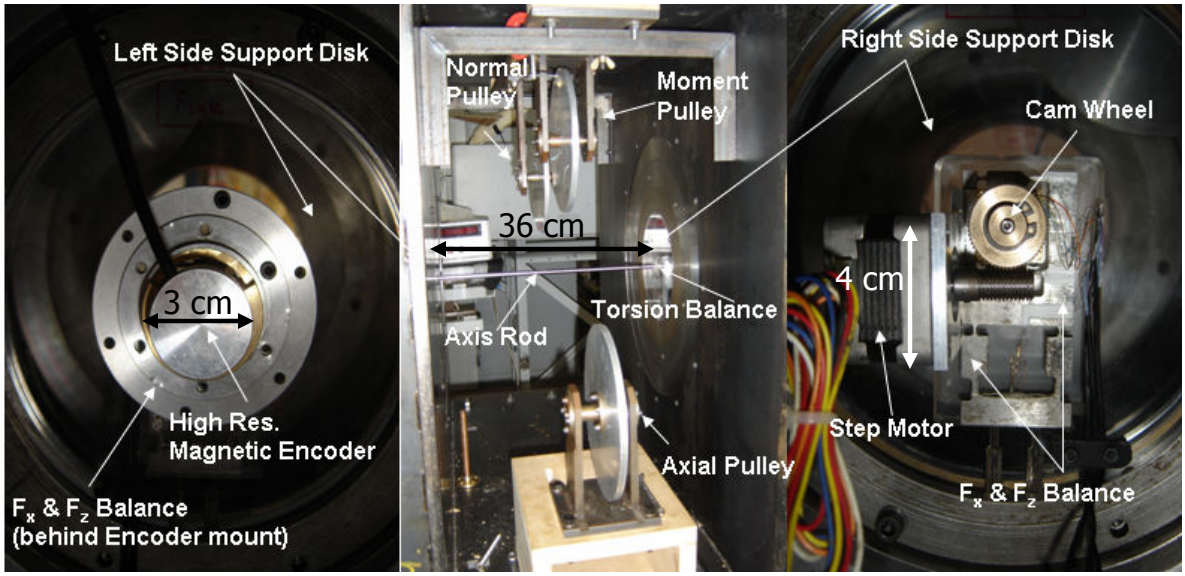

Figure 7 Static test balance system calibration set-up

The axial and normal force (F_x and F_z) balances (one on each side of the wind tunnel walls) and torsion (M_y) balance use strain gages to detect deformations, thereby creating a voltage potential that can be related to the

amount of force or torsion applied, respectively. The maximum allowed forces are 50 N in the axial and normal directions and 0.20 Nm in torsion. The step motor threading is mechanically linked to the cam wheel's gear teeth, and the cam wheel is fixed to the axis rod. An external step motor driver with manual switch powers the step motor to rotate its threaded shaft causing the entire axis rod to rotate on command. The magnetic encoder measures the angular position of the axis rod relative to an initial state with a sensitivity of 4096 steps/revolution. Note that all of the system components described and shown in Figure 7 are placed outside the mock wind tunnel test section with the exception of the axis rod, torsion balance and pulleys. The system of pulleys is used for inducing pure axial, pure normal or pure moment forces so as to reduce coupling between each force component during calibration.

3.1.2 Data Acquisition and Procedure

3.1.2.1 Acquisition Measurement Chain

Data is acquired by the given measurement chain (Figure 8).

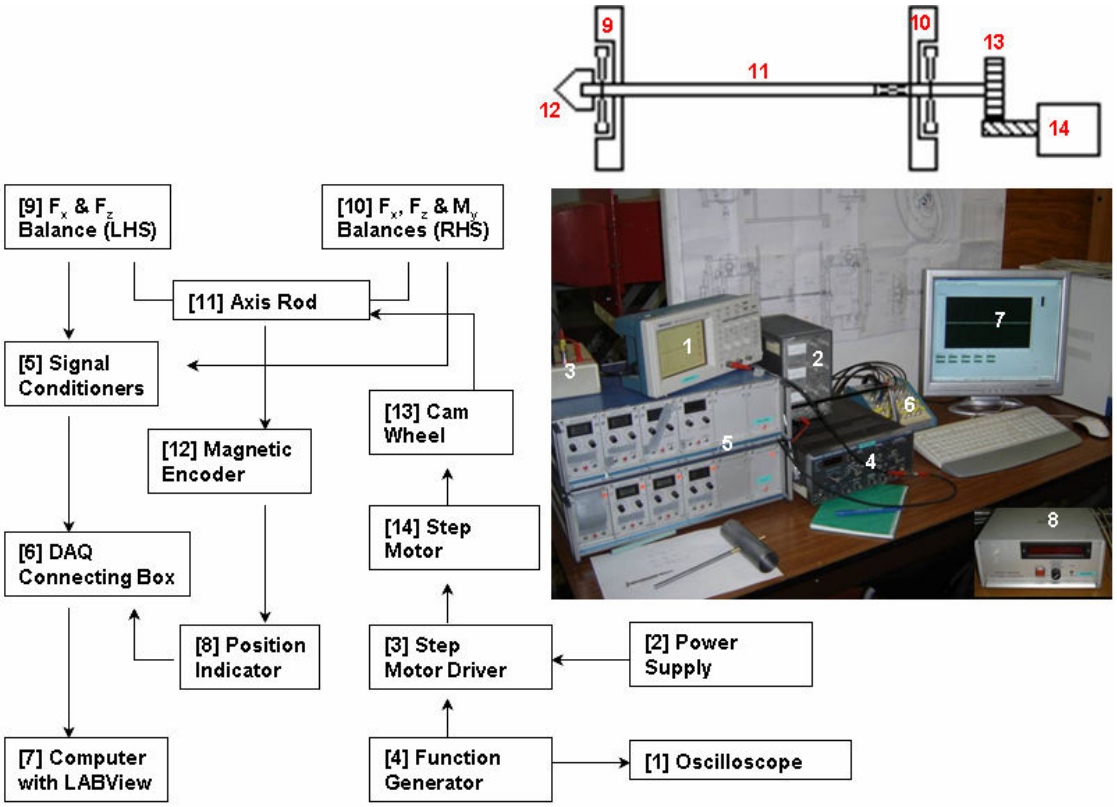

Figure 8 Static test balance system calibration measurement chain

LabVIEW is used for acquiring all data signals that stream through the DAQ connecting box. The DAQ connecting box houses BNC male ports and a LabVIEW data acquisition card (Data ACQ Card PCI-6034E); this is the mechanism used to link the voltage potentials coming from the balances and position indicator to be digitally recorded on computer. The signal conditioners are made at the VKI and are used to condition the signals coming from the balances. The function generator induces a square-tooth wave function for generating the signal from the step motor driver to the step motor.

3.1.2.2 Procedure

The following outlines the procedures following proper installment of all components as shown by Figure 7 and Figure 8.

1. Choose a component of the balance system to calibrate (axial force, normal force, or torsion moment). Note that for axial and normal force measurements, the applied load is shared between the two support disk balances. Thus, the output voltage from both must be added together for producing calibration curves.
2. Apply loads using standardized calibration weights from 0 to the maximum allowable. Obtain 10 or more data points at nearly equal load spacing over this range. Record voltages of all three components corresponding to the applied force or moment and the axis rod angle from the position indicator using the LabVIEW data acquisition program. Perform at axis rod angles of -180 degrees to 180 degrees in steps of 30 degrees.
3. Repeat step 2 until all axial, normal and moment components have been measured.

3.1.3 Results

For the balance system calibration, the voltage output of each strain gage is related to the force or moment applied at that time. These relations can be written as

$$U_x = A_{xx} \cdot F_x + A_{xz} \cdot F_z + A_{xm} \cdot M_y$$
$$U_z = A_{zx} \cdot F_x + A_{zz} \cdot F_z + A_{zm} \cdot M_y$$
$$U_m = A_{mx} \cdot F_x + A_{mz} \cdot F_z + A_{mm} \cdot M_y$$

Or written in matrix form as

$$\begin{pmatrix} U_x \\ U_z \\ U_m \end{pmatrix} = \begin{pmatrix} A_{xx} & A_{xz} & A_{xm} \\ A_{zx} & A_{zz} & A_{zm} \\ A_{mx} & A_{mz} & A_{mm} \end{pmatrix} \cdot \begin{pmatrix} F_x \\ F_z \\ M_y \end{pmatrix}$$

In which one can define $\begin{pmatrix} U_x \\ U_z \\ U_m \end{pmatrix}$ being the voltage outputs, $\begin{pmatrix} A_{xx} & A_{xz} & A_{xm} \\ A_{zx} & A_{zz} & A_{zm} \\ A_{mx} & A_{mz} & A_{mm} \end{pmatrix}$ being the sensitivity (in the principal diagonal) and the other terms interferences and $\begin{pmatrix} F_x \\ F_z \\ M_y \end{pmatrix}$ being the applied forces. After employing the steps of the procedure, the A matrix terms are determined over a complete revolution of axis rotation at 30 degree intervals. The global A matrix is established by averaging the individual values over all axis angles. The uncertainty associated with each A matrix terms comes from the maximum absolute difference between the averaged global value and each individual angle value. Uncertainty plots verify that there is no predictable trend for each A term over the entire range of rod axis angle. Thus, defining the global A matrix by averaging is deemed acceptable and the criterion for determining uncertainty of each term yields complete confidence. The A matrix terms and associated uncertainties are summarized in Table 1.

Table 1 Summary of static test balance system A matrix terms and uncertainties

	Value	***Δ***
A_{xx}	0.1245	5.94E-04
A_{zx}	-8.60E-04	1.02E-04
A_{mx}	1.94E-04	0.0013
A_{xz}	4.94E-04	7.44E-05
A_{zz}	0.1925	1.69E-04
A_{mz}	-1.15E-04	7.36E-04
A_{xm}	-0.0206	0.016
A_{zm}	0.0491	0.0235
A_{mm}	33.955	0.3376

3.1.4 Precision of Measurements

The precision of the magnetic encoder is 360 degrees for every 4096 steps. Thus, the uncertainty of angle is approximately 0.1 degrees. The uncertainty in voltage reading from each balance component is 0.02V.

The general equation for establishing uncertainty for a function dependent on multiple variables, $f = g(x_1, x_2, \ldots)$, is given by Equation 42.

$$\Delta f = \sqrt{\left(\frac{\partial f}{\partial x_1}\right)^2 \Delta x_1^{\,2} + \left(\frac{\partial f}{\partial x_2}\right)^2 \Delta x_2^{\,2} + \ldots} \qquad \textbf{Equation 42}$$

For the coupled 3x3 system, the uncertainties in axial, normal and moment forces during tests can be found using Equation 43 [13].

$$\begin{Bmatrix} (\Delta U_x)^2\left(\frac{\partial F_x}{\partial U_x}\right)^2 + (\Delta A_{xx})^2\left(\frac{\partial F_x}{\partial A_{xx}}\right)^2 + (\Delta A_{xz})^2\left(\frac{\partial F_x}{\partial A_{xz}}\right)^2 + (\Delta A_{xm})^2\left(\frac{\partial F_x}{\partial A_{xm}}\right)^2 \\ (\Delta U_z)^2\left(\frac{\partial F_z}{\partial U_z}\right)^2 + (\Delta A_{zx})^2\left(\frac{\partial F_z}{\partial A_{zx}}\right)^2 + (\Delta A_{zz})^2\left(\frac{\partial F_z}{\partial A_{zz}}\right)^2 + (\Delta A_{zm})^2\left(\frac{\partial F_z}{\partial A_{zm}}\right)^2 \\ (\Delta U_m)^2\left(\frac{\partial M}{\partial U_m}\right)^2 + (\Delta A_{mx})^2\left(\frac{\partial M}{\partial A_{mx}}\right)^2 + (\Delta A_{mz})^2\left(\frac{\partial M}{\partial A_{mz}}\right)^2 + (\Delta A_{mm})^2\left(\frac{\partial M}{\partial A_{mm}}\right)^2 \end{Bmatrix} = \qquad \textbf{Equation 43}$$

$$\begin{bmatrix} 1 & -\left(\frac{\partial F_x}{\partial F_z}\right)^2 & -\left(\frac{\partial F_x}{\partial M}\right)^2 \\ -\left(\frac{\partial F_z}{\partial F_x}\right)^2 & 1 & -\left(\frac{\partial F_z}{\partial M}\right)^2 \\ -\left(\frac{\partial M}{\partial F_x}\right)^2 & -\left(\frac{\partial M}{\partial F_z}\right)^2 & 1 \end{bmatrix} \begin{Bmatrix} (\Delta F_x)^2 \\ (\Delta F_z)^2 \\ (\Delta M)^2 \end{Bmatrix}$$

3.2 Forced Oscillation Torque Balance Calibration

The calibration results of the torque balance are used for the static pitching moment test campaign and the forced oscillation test campaign.

3.2.1 Apparatus and Set-Up

The apparatus used to calibrate this balance is the same as that for the other balance system (seen in Figure 7) minus the actual balances of the other system. Rather than support disks housing axial and normal force balances (which are free to flex in translational motion), they house outer and inner bearing ring spacers with ball bearings between the two. The inner ring's inner diameter is the same as the axis rod diameter. This ensures a "perfect" fit for the axis rod, which ensures translation motion constraints and only one degree of freedom in rotational motion. Also, the axial and normal component pulleys are removed as they are not needed for the torque balance calibration. This torque balance is similar in design to the other torque balance, but slightly thicker in order to endure higher moments for its maximum allowable. The balance is depicted in Figure 9. Finite element analysis is used to determine the maximum allowable moment that can be placed on the torque balance. The resultant value is 0.25 Nm.

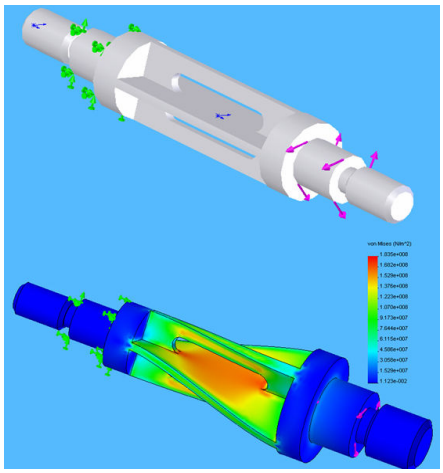

Figure 9 Forced oscillation torsion balance with FEM results for deformation and stress analysis

3.2.2 Procedure and Acquisition

The calibration procedures and acquisition measurement chain is equivalent to the other balance system. The only difference is that axial and normal force calibrations are not performed. Because the geometry of the static test balance system's torsion balance is similar to the forced oscillation torque balance, and the results of the moment calibrations from the static test balance show independence of the variation in moment-to-balance-voltage

slope with angle orientation, only two random angles are calibrated at (denoted as run1 and run2).

3.2.3 Results

The complexity of the forced oscillation torque balance calibration equations are reduced immensely compared to the 3x3 classification of the static test balance system. The moment is linearly related to voltage, as shown in the resolved calibration curve of Figure 10.

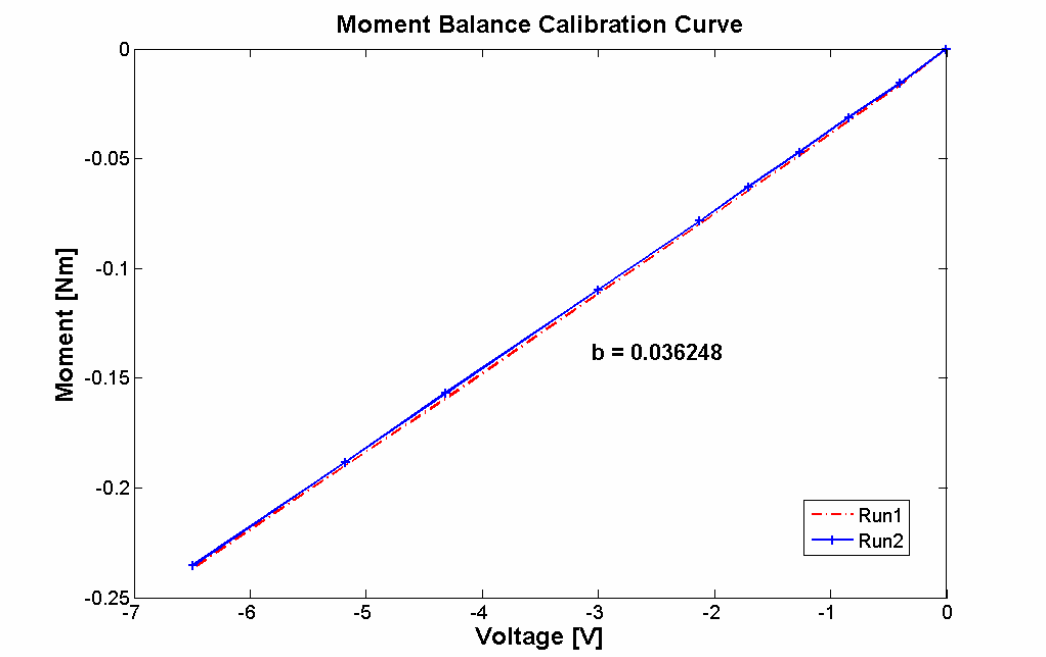

Figure 10 Forced oscillation torsion balance calibration curve

The resulting calibration curve is given as $M_y = b(V - V_0)$, where V_0 is the voltage offset for zero moment.

3.2.4 Precision of Measurements and Uncertainty

The acquisition system and angle uncertainties are 0.02V and 0.1 degrees, respectively, as stated before. The uncertainty of the torsion balance slope, b, is found by taking the maximum absolute difference of each individual run's slope with the averaged, global slope. The resulting uncertainty is Δb = 7.78e-5 Nm/V. Utilizing Equation 42, the uncertainty in moment spanning the moment range for the torque balance is plotted in Figure 11.

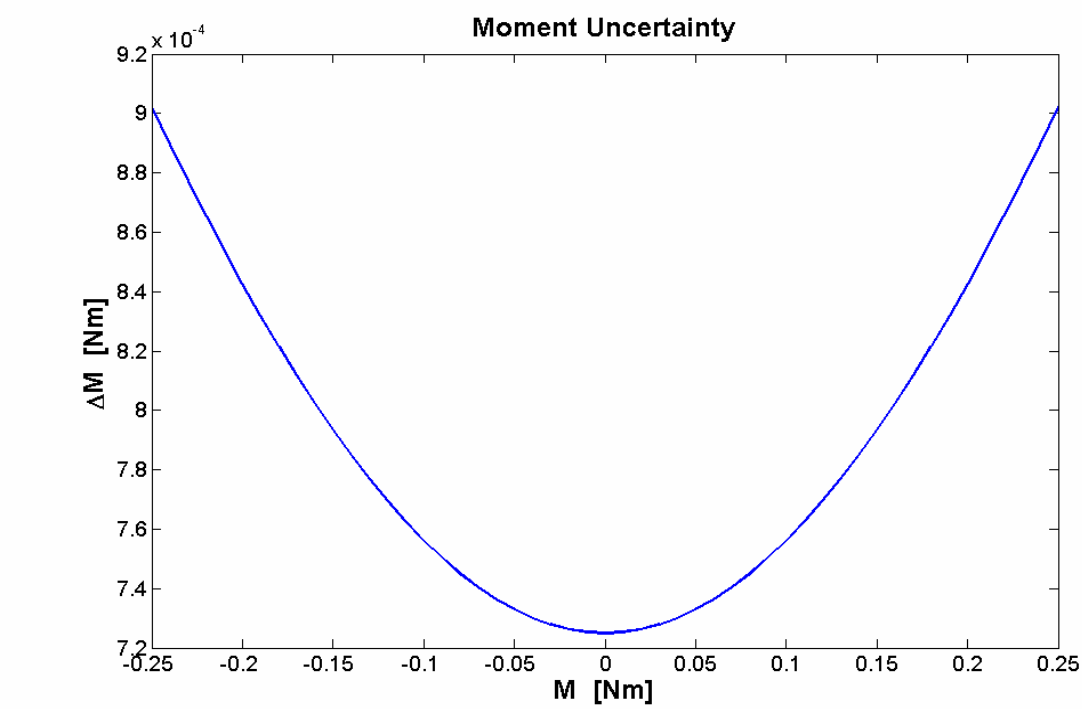

Figure 11 Uncertainty in moment spanning over the range of torque balance maximum allowable moment

3.3 Pressure Transducer Calibrations

Two pressure transducers are calibrated to measure dynamic pressure and total pressure (actually a total minus reference pressure where the reference value is known from a reservoir) during wind tunnel testing. Obtaining these values during testing and knowing total temperature of the flow conditions (which is indicated on the VKI-S1 facility control panel) and using perfect gas and Mach number (isentropic) relations, all other characteristics of the flow are attainable.

3.3.1 Apparatus and Set-Up

Validyne pressure transducers are used. The transducer membrane for the dynamic pressure can withstand up to 14 kPa before possible plastic deformation whereas the membrane for the measurement of total-minus-reference pressure can withstand up to 86 kPa. Demodulators are used to set the voltage output gain and zero offset. A digital pressure transducer calibrator is used to induce pressures for calibration with a sensitivity of 1 millibar. The pressure transducer calibration device is connected to one port of the pressure transducer by a plastic vein tube. The other port of the

transducer is open to the atmosphere. The pressure transducer is connected to the demodulator, which is connected to a computer via the BNC connection box.

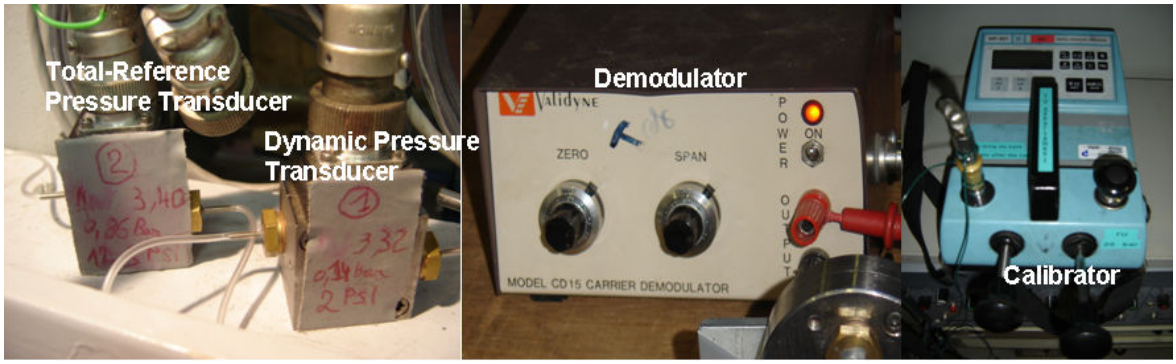

Figure 12 Equipment used for the pressure transducer calibrations

3.3.2 Procedure and Acquisition

The LabVIEW data acquisition system is used for data acquisition. The listed procedure steps are used for the pressure transducer calibration following the complete set-up for the acquisition chain.

1. Set the zero with the demodulator when no pressure is present.
2. Close the calibration device and apply near the maximum allowable pressure for the given transducer membrane being calibrated.
3. Adjust the demodulator gain using the span for higher resolution of voltage output relative to pressure input.
4. Relieve the maximum pressure back to zero and check the value. Adjust the zero accordingly. Repeat step 3 and this step until the desired gain is set and the zero does not move.
5. Apply pressures ranging from zero to the maximum allowable for approximately 10 equivalently spaced data points.

3.3.3 Results

The calibration slope results from the pressure transducer calibrations are given in Figure 13 and Figure 14.

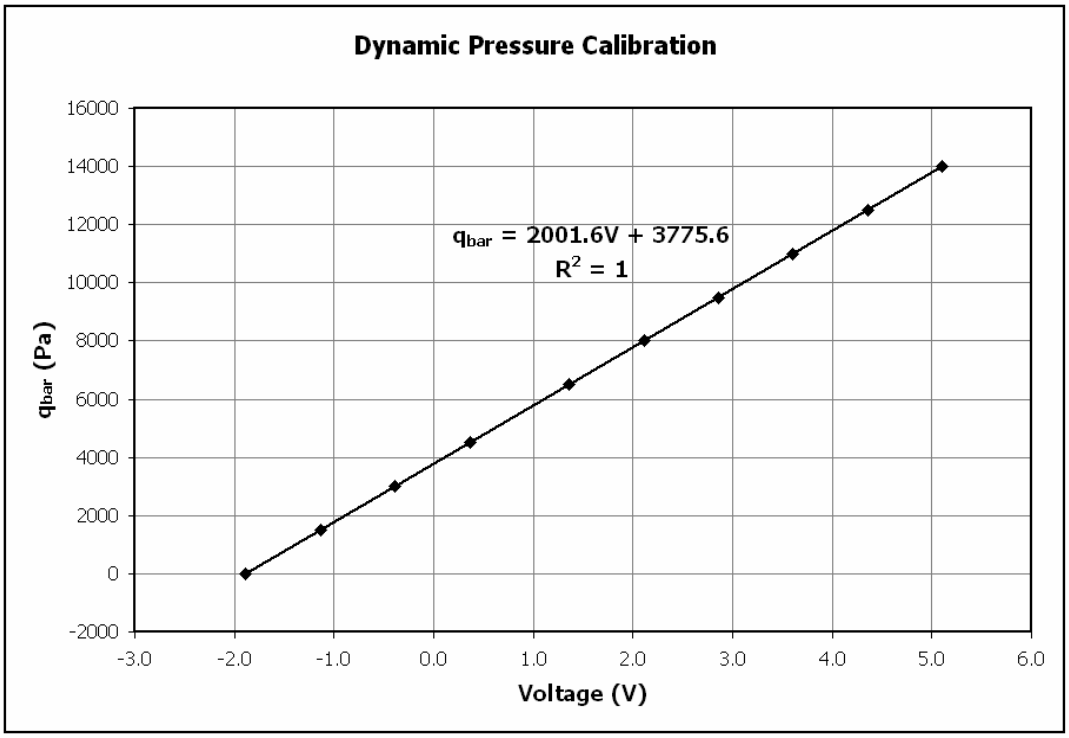

Figure 13 Calibration curve for the dynamic pressure transducer

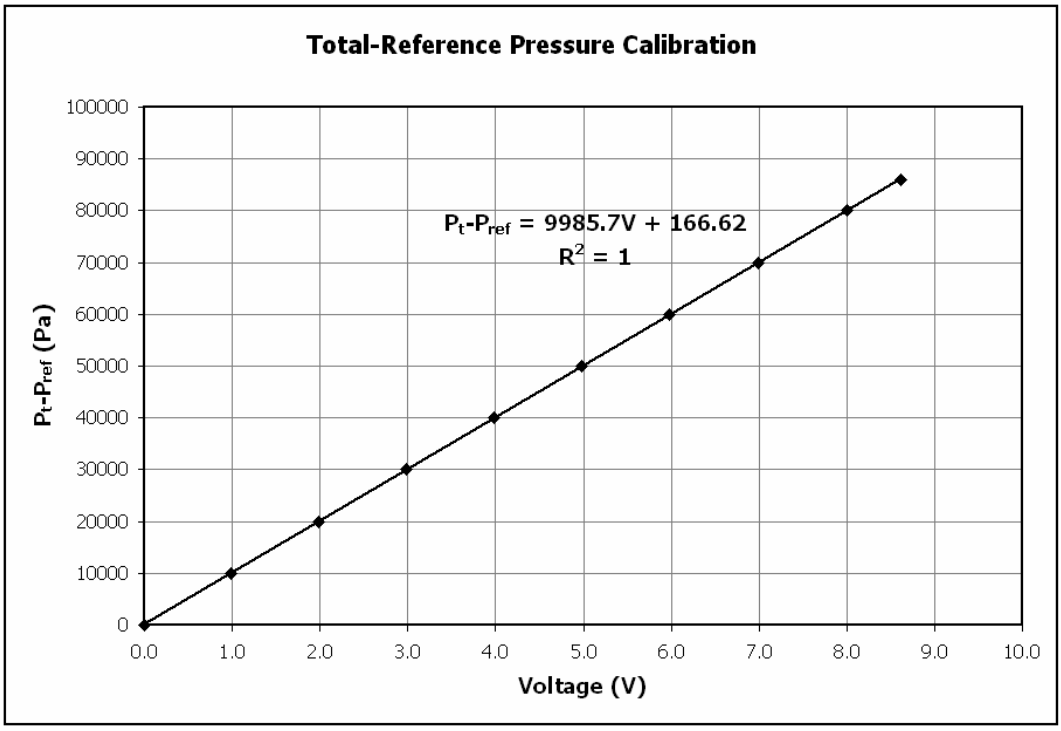

Figure 14 Calibration curve for the total-reference pressure transducer

3.3.4 Precision of Measurements and Uncertainty

The calibrator device has a sensitivity of 1 mbar. Yet, due to random fluctuations of ±3 mbar about the mean voltage value, this can be assumed as the pressure sensitivity of the calibrator in conjunction with the pressure transducer membrane. Pressure uncertainty plots are given in Figure 15 and Figure 16.

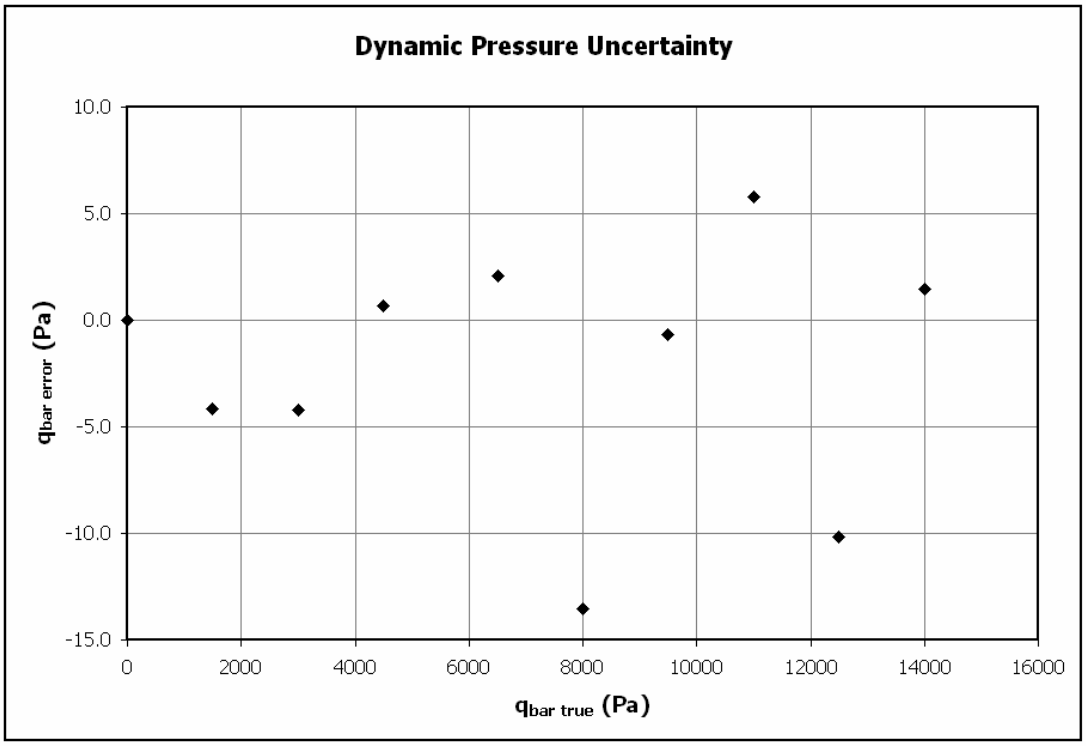

Figure 15 Dynamic pressure uncertainty for the dynamic pressure transducer

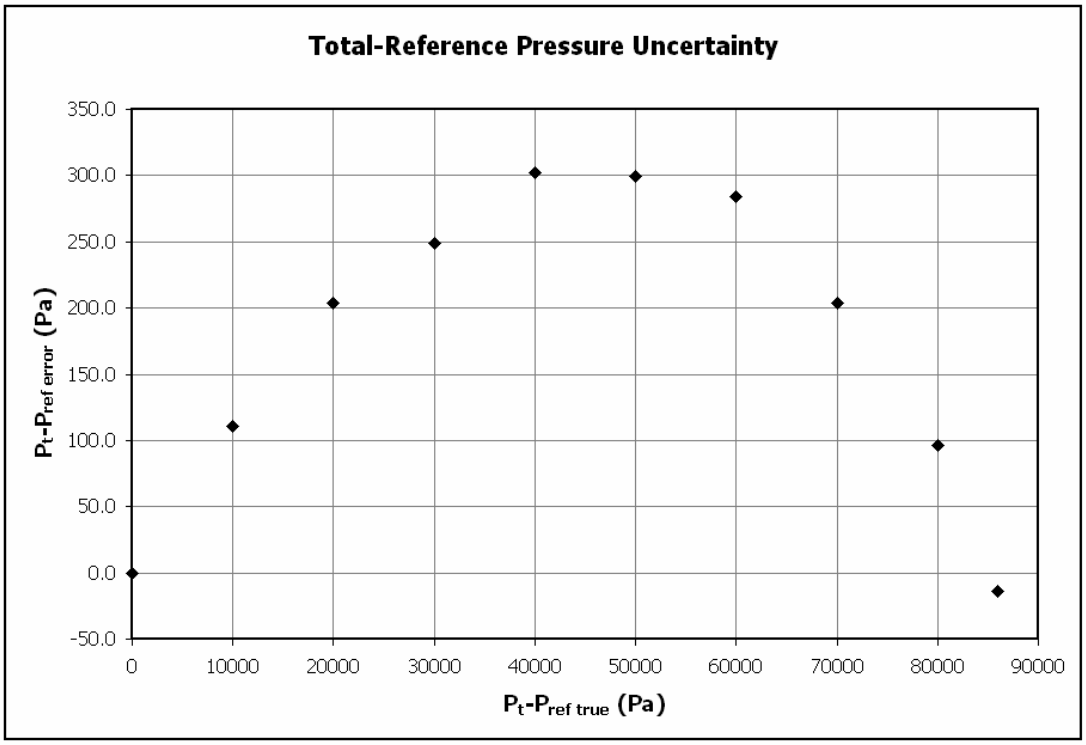

Figure 16 Total-reference pressure uncertainty for the total-reference pressure transducer

3.4 Static Pitching Moment Tests

The sole purpose for the static pitching moment tests is to extract the Strouhal number based on the contribution of $\left(\partial C_m / \partial \theta\right)$ in the aerodynamic stiffness-in-pitch for each model at each test Mach number. The resulting values are used for determining oscillation frequencies for the model during the forced oscillation testing. The values of $\left(\partial C_m / \partial \theta\right)$ between the flight vehicle and model are understood to be the same for same Mach number and same angle of attack by assuming the model and true flight vehicle share the same static response characteristics. The basis of this postulation comes from the assumption that under static conditions, only geometric and axis of rotation location similarities are required for similitude of static moment response characteristics. Though this neglects the surface roughness differences between the flight vehicle and the model as well as Reynold's number effects, it is a credible method for estimating Strouhal numbers based on the natural frequency of the flight vehicle, especially in instances where no published data is available.

3.4.1 Apparatus and Set-up

Testing is conducted in the VKI-S1 wind tunnel (Figure 3). Apollo and Expert models are made using the stereolithography process (models shown in Figure 1 and Figure 2). The model material is the DSM Somos NanoForm 15120 nanocomposite resin, which has a density of only 1.38 g/cm^3 when hardened. The light density of the model (compared to past models investigated at the VKI, using mostly Plexiglas or metals) keeps the moment of inertia down. This is important because it allows the models to oscillate at higher forced frequencies while reducing inertial damping forces. Model geometries are given in Figure 17 and Figure 18.

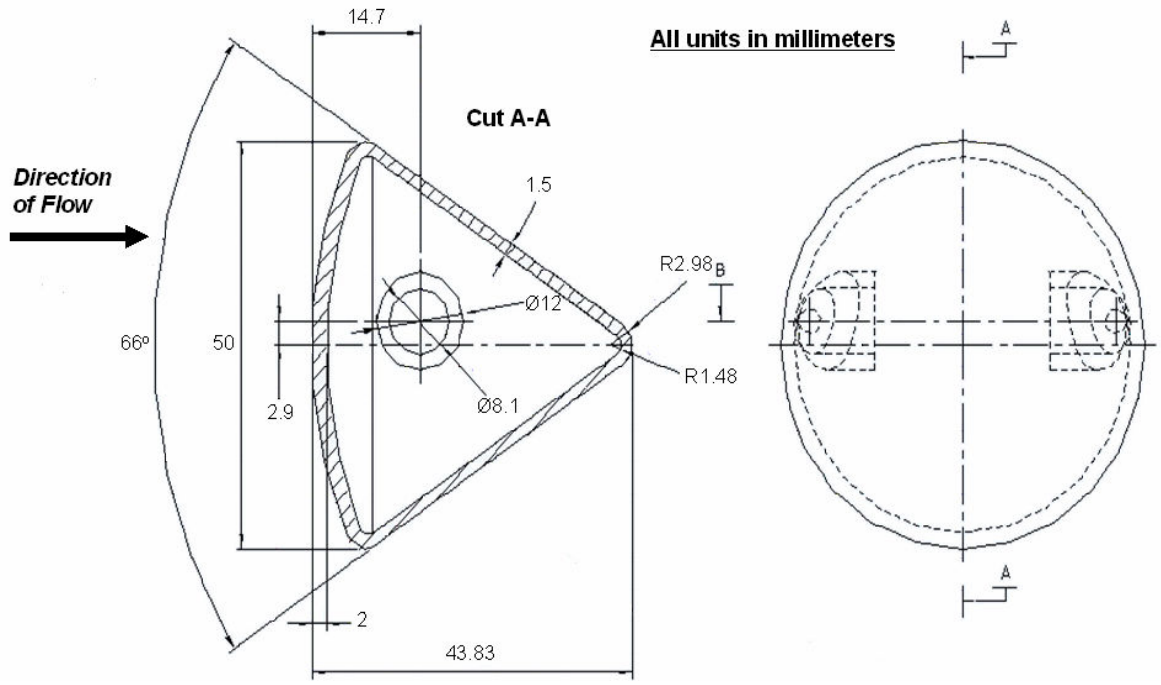

Figure 17 Apollo model geometry at α = 180° based on flow direction

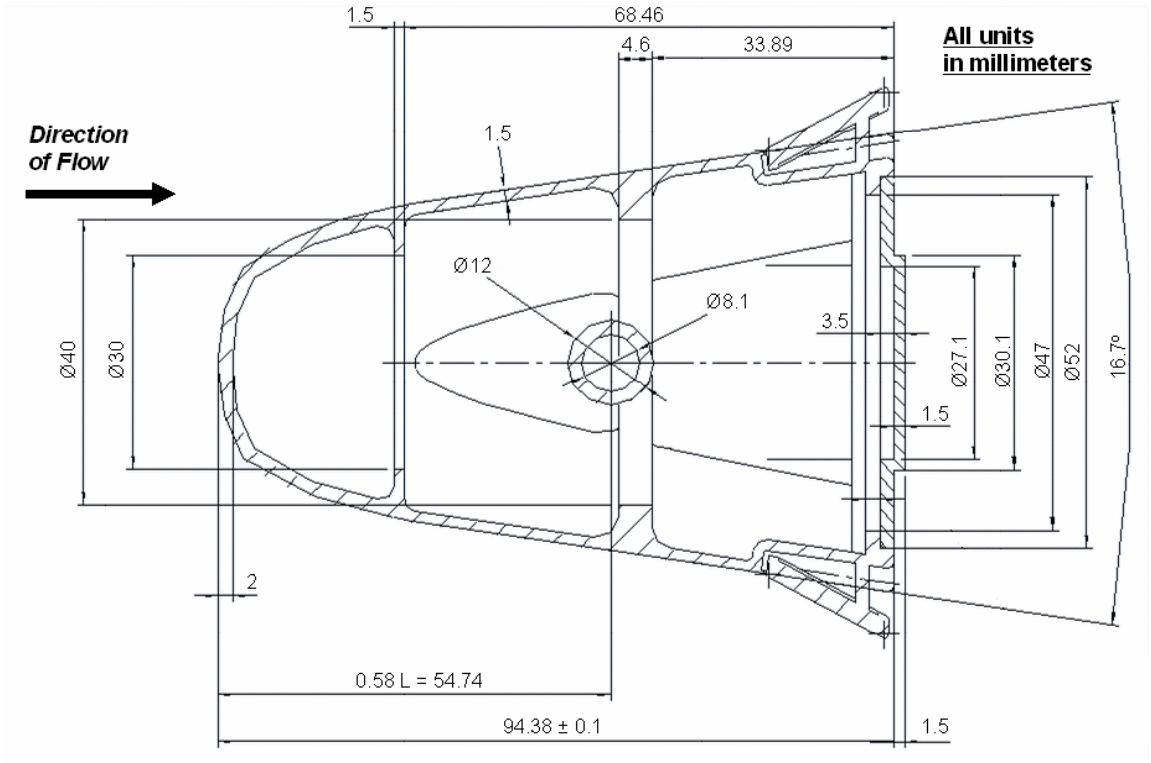

Figure 18 Expert model geometry at α = 0° based on flow direction

The body-fixed coordinate system used for the Apollo model is depicted in Figure 19 and in Figure 20 for the Expert model (Expert model is the 4.4B). Center of gravity and rotation point locations are also defined in the figures.

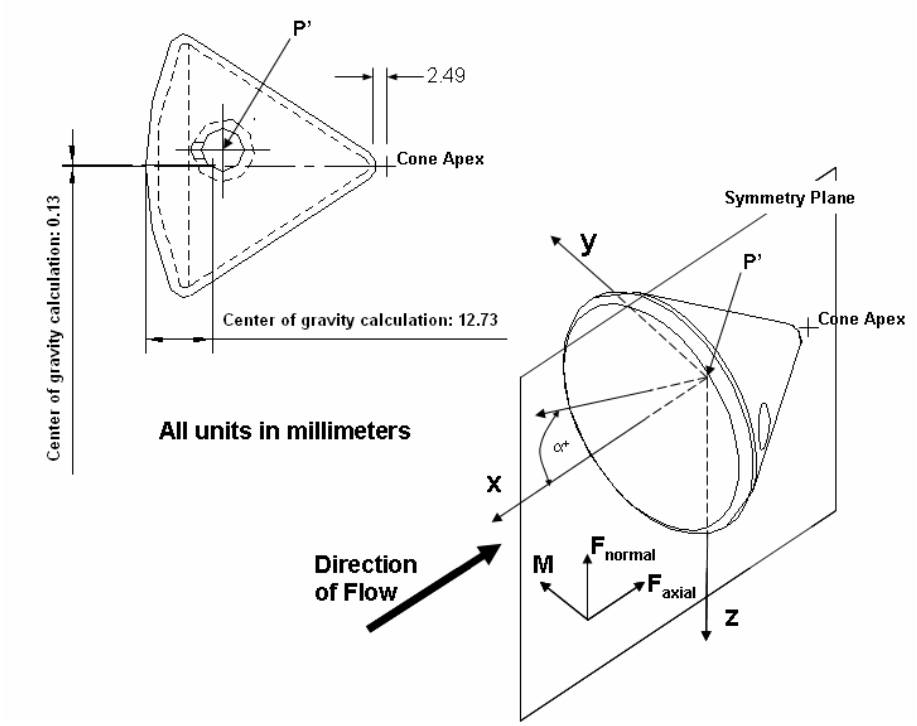

Figure 19 Body-fixed coordinates and c.g. location for the Apollo model

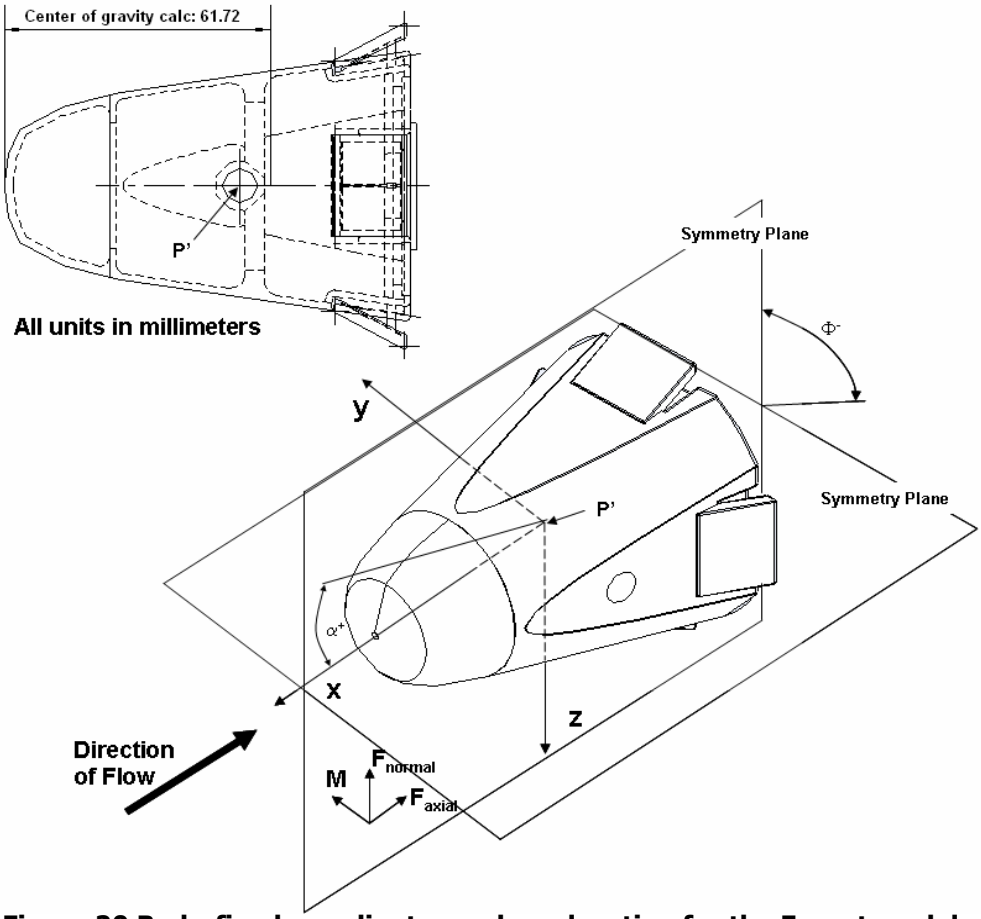

Figure 20 Body-fixed coordinates and c.g. location for the Expert model

The complete torque balance system is transferred from the calibration set-up to the VKI-S1 wind tunnel. Note that due to electromagnetic interference caused by the VKI plasmatron test facility when running, the encoder was changed from magnetic to optical between calibrations and the actual test campaigns. The optical encoder has a higher sensitivity (8000 steps/revolution). The models are glued onto the lateral support axis rod such that the vertical plane cutting through each model for symmetry is in-line with the center of the wind tunnel test section.

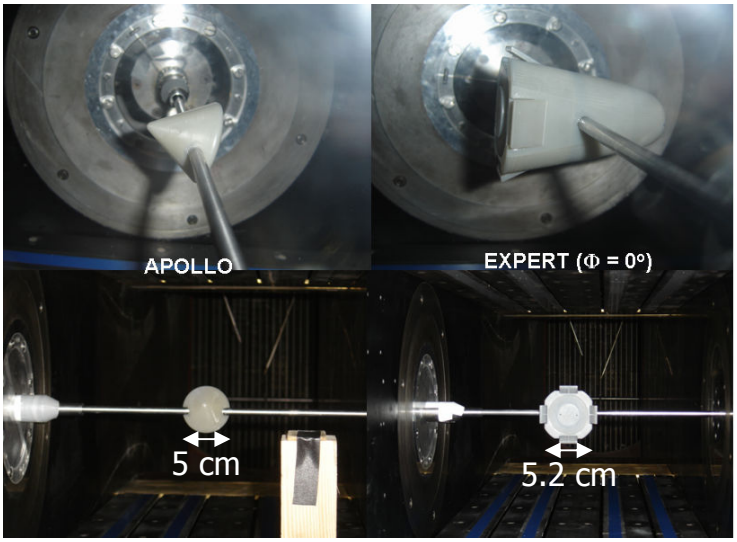

Figure 21 Apollo and Expert model placement in the VKI-S1 wind tunnel

A fitted sheath is rigidly fixed to the support disks (which are also laterally-rigidly fixed as part of the wind tunnel test section side walls). This allows the torsion balance to be shielded from the flow while not interfering with the lateral support axis rod. A manual crankshaft is outfitted on the VKI-S1 that allows the entire system (model and axis) to be rotated to different incidences without inducing any external moments on the torsion balance (this is because the torsion balance rotates with the system as a whole). This is used as a replacement for the step motor and cam wheel set-up, which are not utilized for this or the forced oscillation test campaign.

Although the axis rod is well-fitted with the inner bearing ring spacer, flow during testing could leak out because there is no perfect seal with the set-up as described thus far. Therefore, large metal casings with windows for viewing are attached external to the wind tunnel test section walls to provide a seal. These casings house the components that are part of the system,

which are also external to the wind tunnel test section and out of the way of the flow.

3.4.2 Procedure and Acquisition

Data is acquired using the LabVIEW acquisition system. Data sampling rate is set to 100 Hz for static moment testing. The programming block diagram for this test is depicted in Figure 22 and Figure 23. The LabVIEW front panel associated with the block diagram is given in Figure 24.

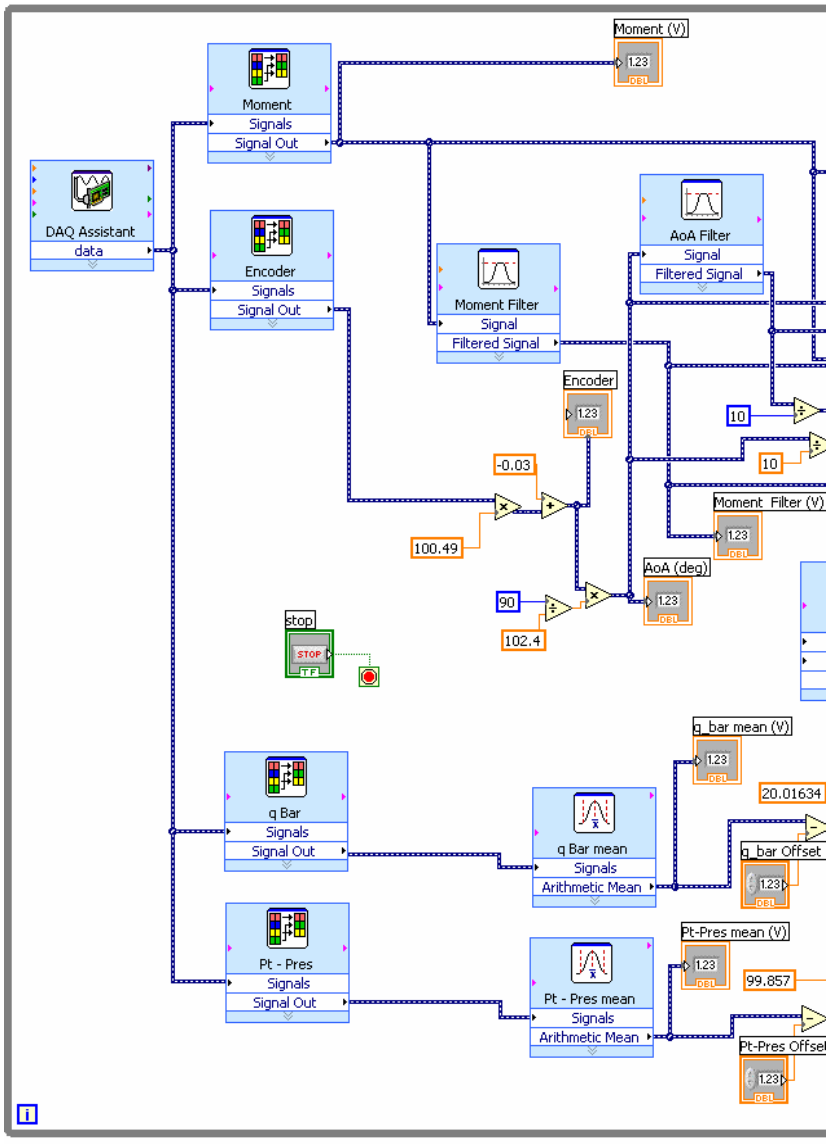

Figure 22 LabVIEW programming block diagram for the static moment test campaign – Part 1

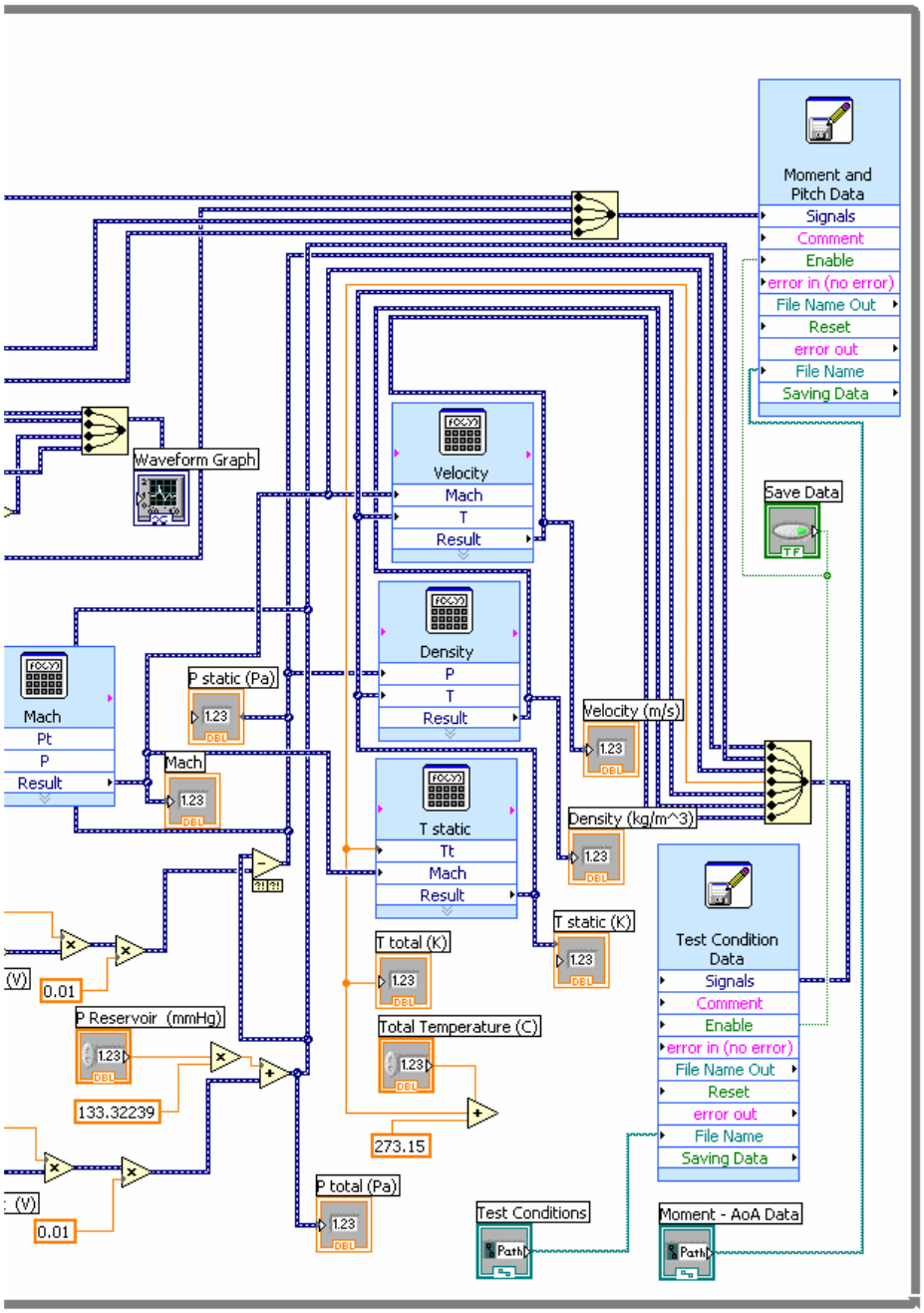

Figure 23 LabVIEW programming block diagram for the static moment test campaign – Part 2

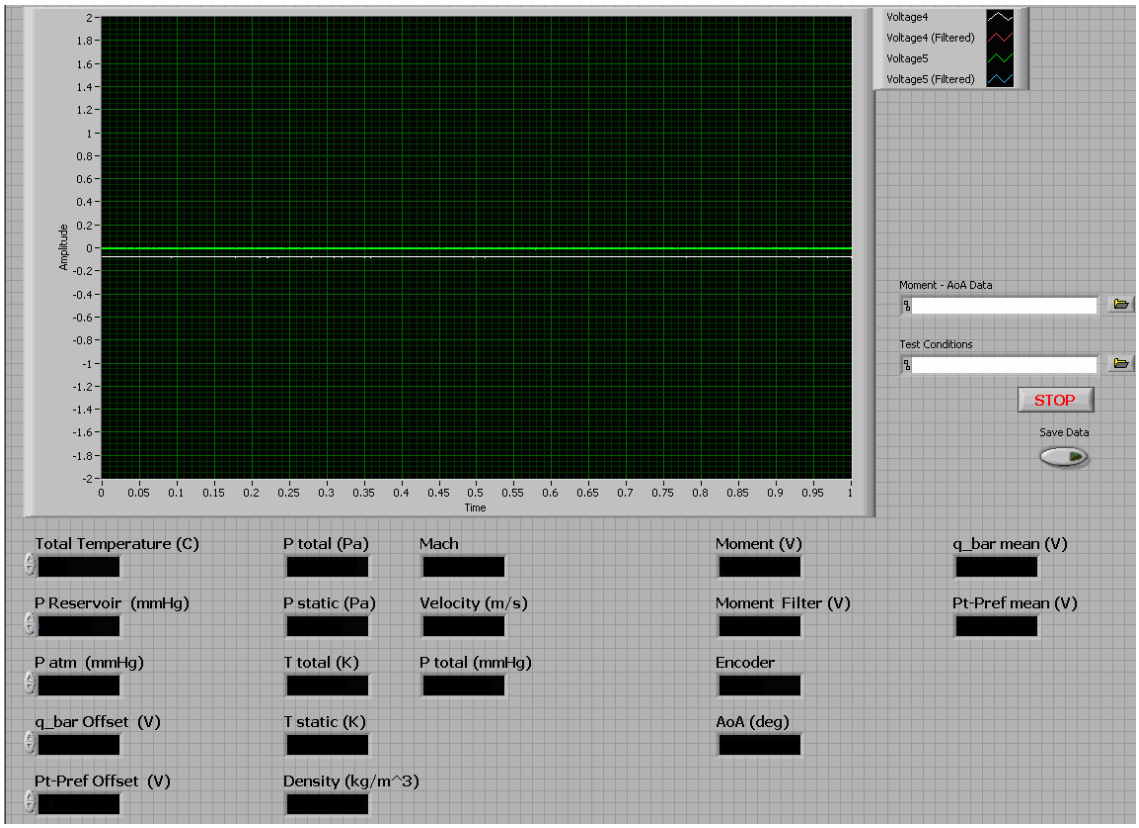

Figure 24 LabVIEW user interface front panel for the static moment test campaign

The procedural steps for this test are outlined as follows:

1. Establish the zero degree angle of attack of the model relative to flow direction before creating vacuum or flow conditions. With this set, zero the position indicator. For precautionary measure and redundancy, use a stationary telescope to denote the zero angle position of the model being tested.
2. Create vacuum in the wind tunnel. The start up of the wind tunnel compressor can cause the encoder values on the position indicator to drift. Once vacuum is complete and stabilized, double-check the zero with the stationary telescope and re-zero the position indicator.
3. Record approximately 30 to 40 moment voltage data points over a specified range of incremental angles of attack (approximately $+135^{o}$ to $+225^{o}$ for the Apollo and approximately -20^{o} to $+20^{o}$ for the Expert) in vacuum. These are the moment voltage offset values (see Section 3.2.3).

4. Set desired Mach number and test conditions. Once the flow has stabilized, repeat the same procedure as done in step 3. Record data at static, incremental steps in angle of attack.

3.4.3 Results

Test procedures are the same for all Mach numbers investigated.

3.4.3.1 Apollo

Test conditions and important characteristics for the Apollo model are outlined in

Table 2 and Table 3, respectively.

Table 2 Apollo test conditions for static moment measurements

Mach	*P (Pa)*	*T (K)*	*U_∞ (m/s)*	*q_∞ (Pa)*	*Re (10^6/m)*
0.50	22420	285	170	3940	2.58
0.70	19270	281	234	6530	3.15
0.89	16310	266	291	9030	3.65
2.0	1040	166	517	2920	1.00

Table 3 Geometric and mass characteristics of the Apollo model

Coordinate Convention	*Rotation Pt. (x/D)*	*Rotation Pt. (z/D)*	*m (kg)*	*I (kg-m^2) x10^{-6}*	*Characteristic Length, D (m)*	*Reference Area, S (m^2)*
Torgler Karatekin, [1]	0 0.632	0 0.058	0.014	3.364	0.05	0.0019635

Axis rod geometry is given in Table 4.

Table 4 Geometric and mass characteristics of the lateral support axis

m (kg)	*I (kg-m^2) x10^{-6}*	*Diameter (m)*	*Length (m)*
0.176	1.405	0.008	0.445

A Matlab function *func_static_moment_fo.m* (given in the appendix) is used to process the moment voltage offset data and moment test voltage values to

yield the static pitch moment coefficient against angle of attack. The results of the testing are represented in Figure 25.

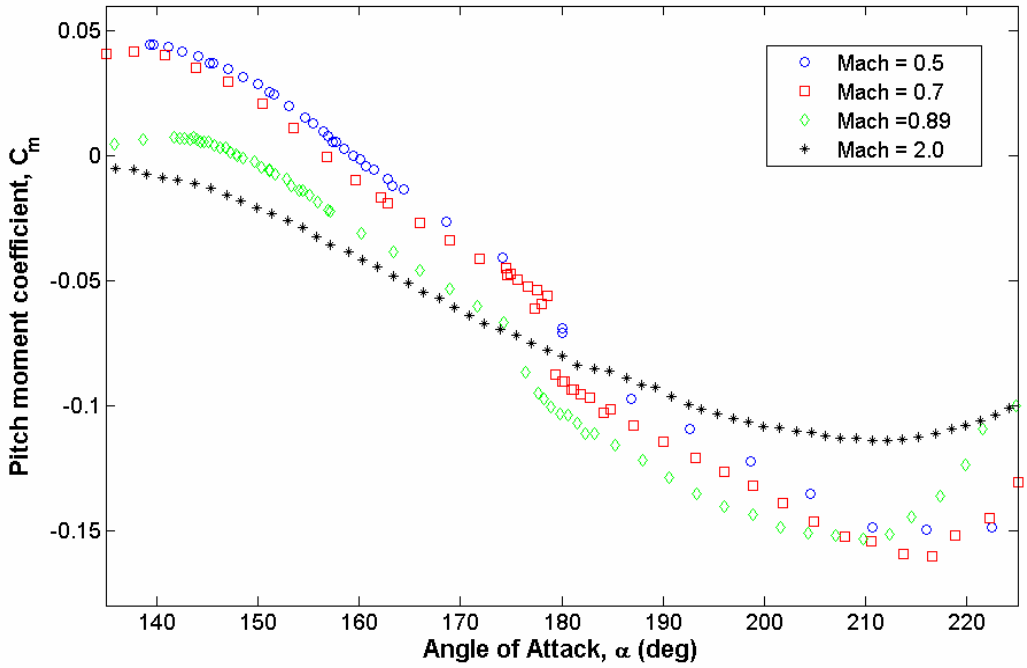

Figure 25 Variation of Apollo model pitch moment coefficient against attack angle

3.4.3.2 Expert at Φ = 0°

Test conditions and important characteristics for the Expert model are outlined in Table 5 and Table 6, respectively.

Table 5 Expert test conditions for static moment measurements

Mach	*P (Pa)*	*T (K)*	*U_∞ (m/s)*	*q_∞ (Pa)*	*Re (10^6/m)*
0.50	22050	286	169	3830	2.52
0.70	18890	280	235	6490	3.12
0.88	16180	276	293	8770	3.41
2.09	1730	163	535	5310	1.79

Table 6 Geometric and mass characteristics of the Expert model

Coordinate Convention	*Rotation Pt. (x/L)*	*Rotation Pt. (z/L)*	*m (kg)*	*I (kg-m^2) $x10^{-6}$*	*Characteristic Length, L (m)*	*Reference Area, S (m^2)*
Torgler Design, [15]	0 0.580	0 0	0.065	68.878	0.09438	0.0044036

The results of the testing are represented in Figure 26.

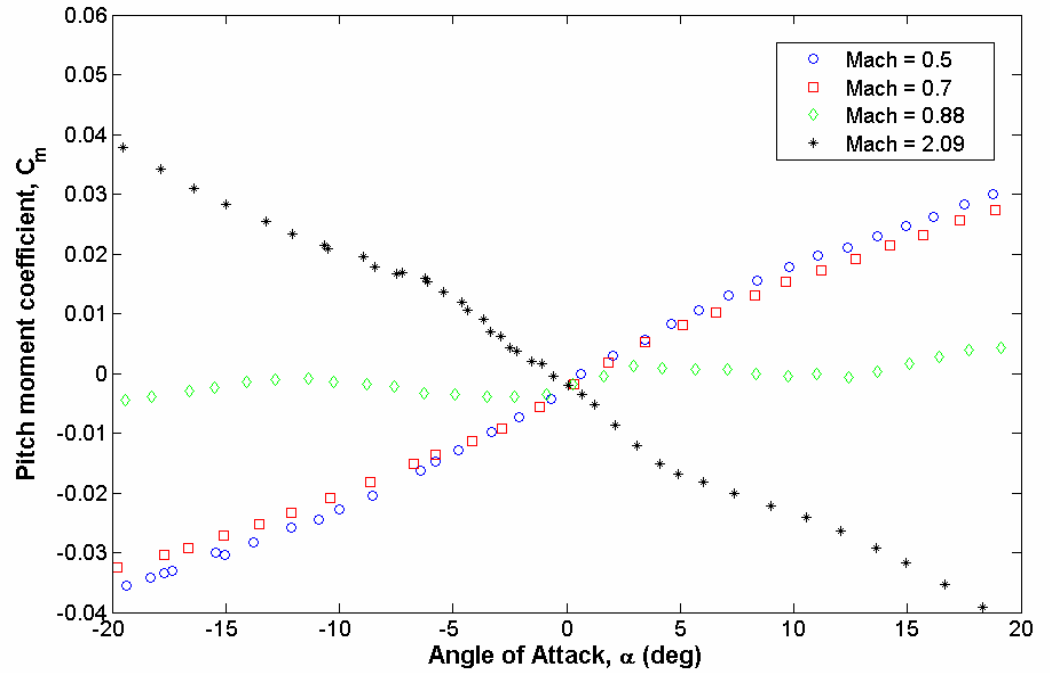

Figure 26 Variation of Expert model pitch moment coefficient against attack angle for $\phi = 0^o$

3.4.4 Discussion and Data Comparison

3.4.4.1 Vehicle Trajectories

Apollo flight vehicle trajectories are based on flight data obtained in the post-launch report of the AS-202 mission [14]. For comprehensive data comparison purposes, geometric characteristics given in Karatekin, [1] are used. Expert flight trajectories and geometric and mass characteristics are obtained from data used for the Alcatel Alenia Space's Expert design and configuration report [15].

Table 7 Apollo flight vehicle trajectories

Mach	*P (Pa)*	*T (K)*	*U_∞ (m/s)*	*Re (10^6)*
0.50	17236	217	148	11.27
0.70	10688	217	207	9.79
0.89	6628	217	266	7.80
2.0	1251	226	603	3.09

Table 8 Expert flight vehicle trajectories

Mach	P (Pa)	T (K)	U_∞ (m/s)	Re (10^6)
0.50	20329	224	150	4.98
0.70	16246	223	210	5.64
0.88	14431	222	262	6.28
2.09	9904	222	624	10.29

3.4.4.2 Strouhal Numbers

For the forced oscillation testing, oscillations are induced on the models about a trim angle of attack where pitching moment is zero (used at that angle when possible). The exception to this is the Apollo supersonic case, where an angle that can be compared with published data is chosen. The local pitch moment slope with respect to angle of attack, which is equivalent to the value of $\left(\partial C_m / \partial \theta\right)$, is evaluated at that angle. As stated previously in Section 3.4, values of $\left(\partial C_m / \partial \theta\right)$ between the flight vehicle and model are understood to be the same for same Mach number and same angle of attack. The local slopes are evaluated based on the data shown in Figure 25 and Figure 26. These values are then used in accordance with Equation 7, Equation 8, and Equation 9 to yield the data given in Table 9 and Table 10.

Table 9 Apollo flight vehicle frequency characteristics for trajectory Mach number values with geometric and mass characteristics

Mach	f_n (Hz)	St	$dC_m/d\theta$ (rad^{-1})	I ($kg\text{-}m^2$)	Characteristic Length, D (m)	Reference Area, S (m^2)
0.50	0.328	0.00878	-0.180	6200	3.949	12.249
0.70	0.376	0.00720	-0.196			
0.89	0.270	0.00406	-0.100			
2.0	0.279	0.00183	-0.113			

Table 10 Expert flight vehicle frequency characteristics for trajectory Mach number values with geometric and mass characteristics

Mach	f_n (Hz)	St	$dC_m/d\theta$ (rad^{-1})	I ($kg\text{-}m^2$)	Characteristic Length, L (m)	Reference Area, S (m^2)
0.50	0.536	0.00553	0.153	88.5	1.55	1.188
0.70	0.644	0.00476	0.141			
0.88	0.516	0.00305	0.065			
2.09	1.616	0.00401	-0.164			

3.5 Forced Oscillation Tests

The Strouhal numbers obtained for the actual flight vehicles are used in conjunction with Equation 7, each model's characteristic length, and the static moment test free stream velocities given in Table 2 and Table 5 to yield the required oscillation frequency to preserve Strouhal numbers between the actual flight vehicles and wind tunnel models. The oscillation frequencies for forced oscillation testing are given in Table 11 and Table 12.

Table 11 Required Apollo model oscillation frequencies for Strouhal number preservation based on static moment test Mach number values

Mach	f_o *(Hz)*
0.50	29.8
0.70	33.6
0.89	23.6
2.0	18.9

Table 12 Required Expert model oscillation frequencies for Strouhal number preservation based on static moment test Mach number values

Mach	f_o *(Hz)*
0.50	9.9
0.70	11.8
0.88	9.5
2.09	22.8

The amplitude of oscillation angle is set to approximately 5 degrees for each model tested. The reason for this value is to maintain within the linear range of $\partial C_m / \partial \theta$, so that slope values are (near) constant over the range. This yields constant Strouhal numbers.

3.5.1 Apparatus and Set-Up

In addition to the apparatus and set-up used for the static moment tests, a motor and drive inverter mechanism is added to the system. The specifications of the motor indicate crankshaft rotation speeds up to 1500 rpm. The drive inverter used is the SEW Eurodrive MDV60A0030-5A3-4-00 [16]. A resistance mechanism is shielded and used in conjunction with the drive inverter. An internal cooling fan is enclosed in the motor casing for cooling. The motor is installed on the side of the wind tunnel opposing the encoder location. Its crankshaft is mechanically linked to the lateral support axis rod via gearing and rotary belts. The ratio of motor rotation frequency

to oscillation frequency of the axis rod and model system is given by the ratio of gear diameters, which 72 to 26. Thus, for every 26 Hz of motor crankshaft rotational frequency, the system oscillates at a frequency of 72 Hz. The motor system and details of the mechanical linking are depicted in Figure 27 and Figure 28.

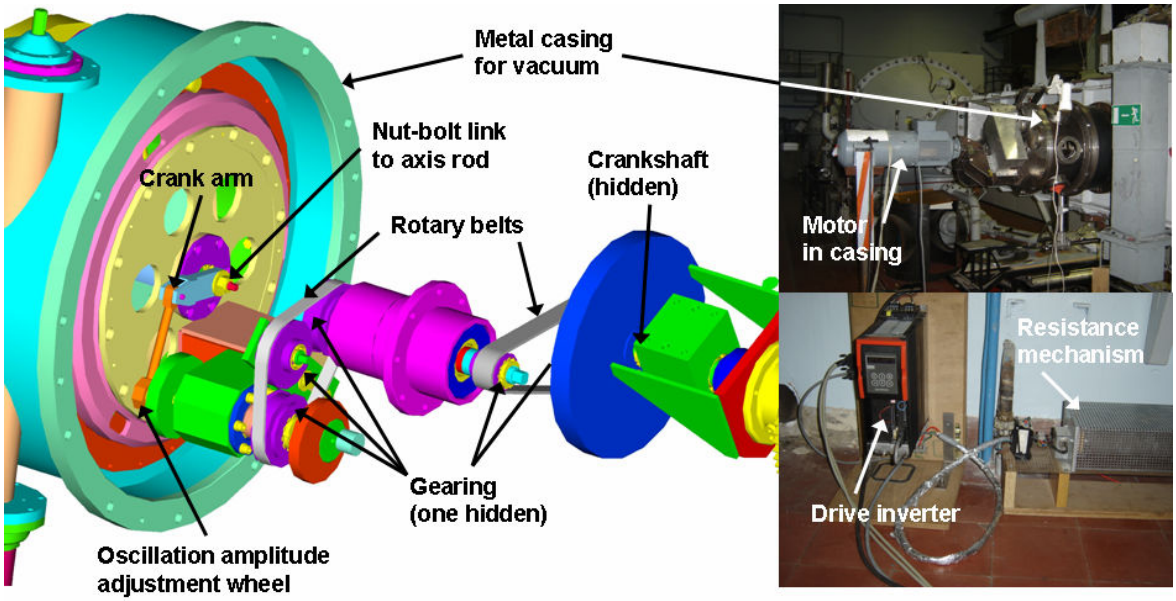

Figure 27 Forced oscillation motor and drive inverter system with mechanical linking

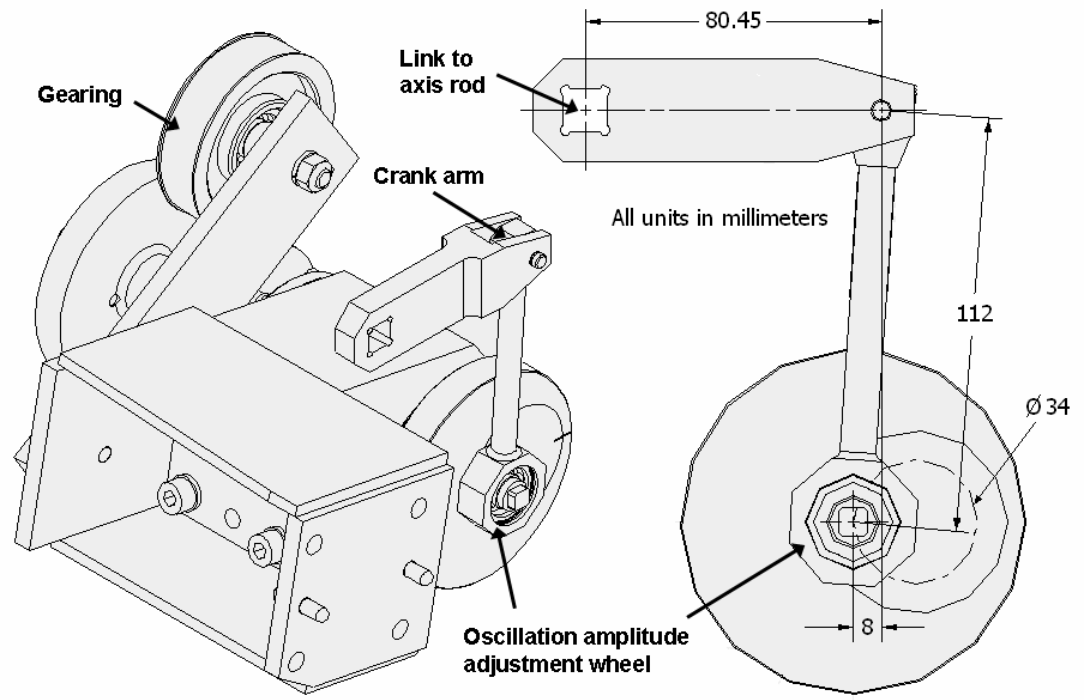

Figure 28 Details of the crank arm and amplitude adjustment wheel of the forced oscillation motor system

As seen from Figure 28, the adjustment wheel for setting the amplitudes of oscillation is done by mechanically adjusting a nut and bolt. This creates difficulty in acquiring precise desired oscillation angle amplitudes; this is the reason why the amplitude of oscillation angle is set to approximately 5 degrees and not exactly 5 degrees.

3.5.2 Procedure and Acquisition

Data is acquired using the LabVIEW acquisition system. The programming block diagram for this test is the same as used for the static moment tests (depicted in Figure 22). The Nyquist criterion states that the sampling frequency must be more than twice the signal frequency to avoid aliasing. Standard practice is to have data sampled at even more than four times the signal frequency. For this test, the data sampling rate is set to 10 kHz, which is well above four times the oscillation frequencies given in Table 11. This guarantees any and all frequencies are captured.

The procedural steps for this test are outlined as follows:

1. Immediately following the static moment test with the flow conditions still set, perform all the following steps.
2. At a low motor speed (about 50 rpm), adjust the mean oscillation angle to the trim angle of attack of the model.
3. For a given model and Mach number, set motor rpm values to ensure match with Strouhal numbers. Test at rpm values different from the Strouhal numbers as well (left to discretion). Record torque balance moment voltages and encoder signal angle positions for approximately 10 seconds.
4. Follow the same procedure until all Mach numbers (both for static moment tests and forced oscillation tests) are completed and stop the wind tunnel flow but leave the system in vacuum.
5. While in vacuum with no flow, repeat steps 2 and 3. This produces no flow data.

3.5.3 Results

Though wind tunnel conditions should remain the same for forced oscillation testing as for the static moment tests, there is likely to be some slight change in the flow conditions. This is due to the VKI-S1 facility being a continuous circuit wind tunnel – density and temperature (and thus static and dynamic pressure) of the flow will change slightly over lengthy periods of testing.

3.5.3.1 Apollo

The flow conditions for the Apollo forced oscillation test campaign are given in Table 13.

Table 13 Apollo test conditions for forced oscillation measurements

Mach	*P (Pa)*	*T (K)*	U_∞ *(m/s)*	q_∞ *(Pa)*	*Re* (10^6/*m*)
0.50	22270	287	169	3880	2.54
0.69	19220	281	233	6450	3.12
0.89	16400	281	299	9070	3.42
2.1	860	159	531	2650	0.92

A Matlab function *func_dynamic_fo.m* (given in the appendix) is used to process the data. Moment and angle positions for each tests' duration and the range of oscillation angles about the mean angle of attack are generated. Processing techniques to smooth the data while preserving the true oscillatory signal are performed. A Matlab MathWorks written function using

the Savitzky-Golay smoothing process routine is applied for all acquired data sets. A second Matlab routine *func_phase_lock.m* (in the appendix) is written to extract the ensemble average of the smoothed signals over one period of oscillation. Only data for the closest match of vehicle and model Strouhal numbers at each test Mach number are given in this section along with corresponding no flow data (refer to Table 11). All other results are given in the appendix.

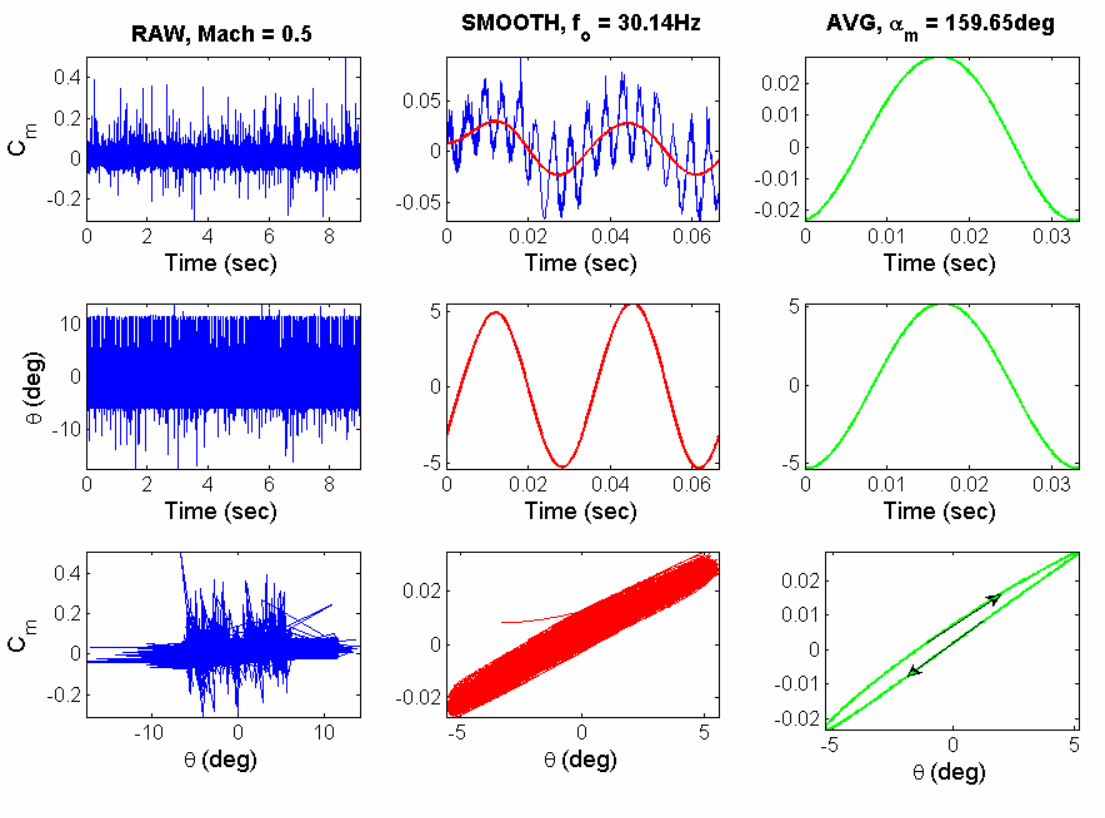

Figure 29 Apollo forced oscillations (Mach = 0.5, f_o = 30.1 Hz)

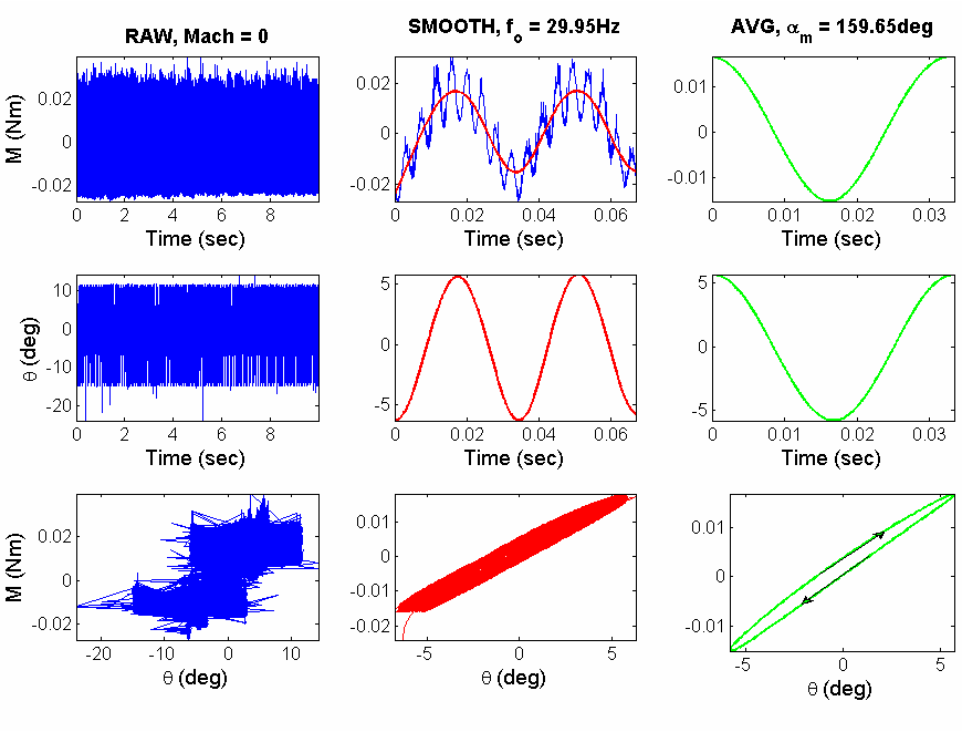

Figure 30 Apollo forced oscillations (no flow, f_o = 30 Hz)

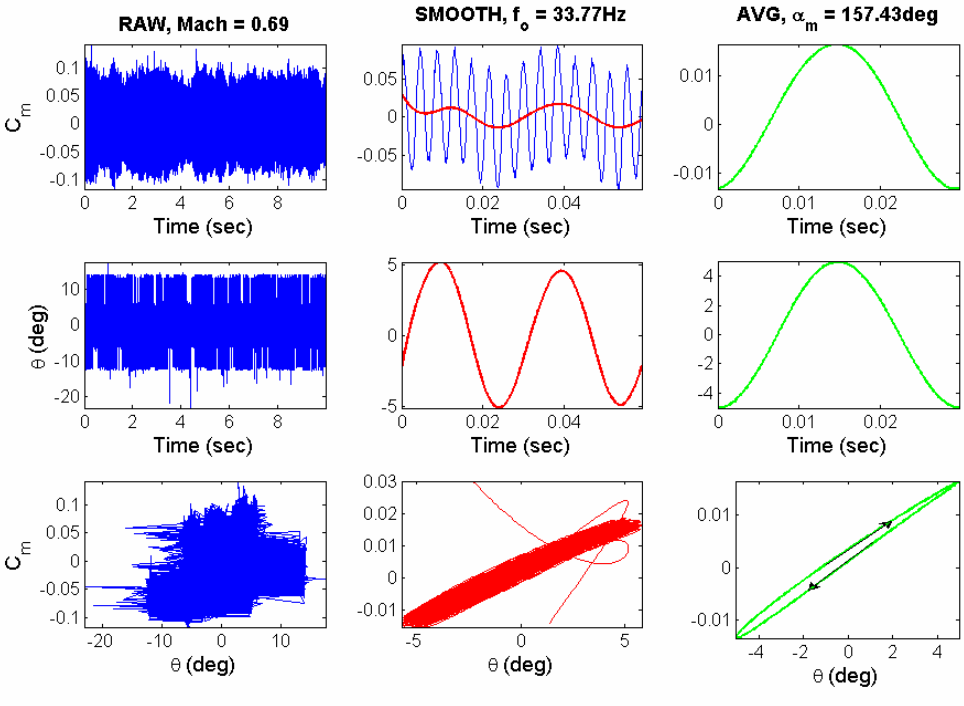

Figure 31 Apollo forced oscillations (Mach = 0.69, f_o = 33.8 Hz)

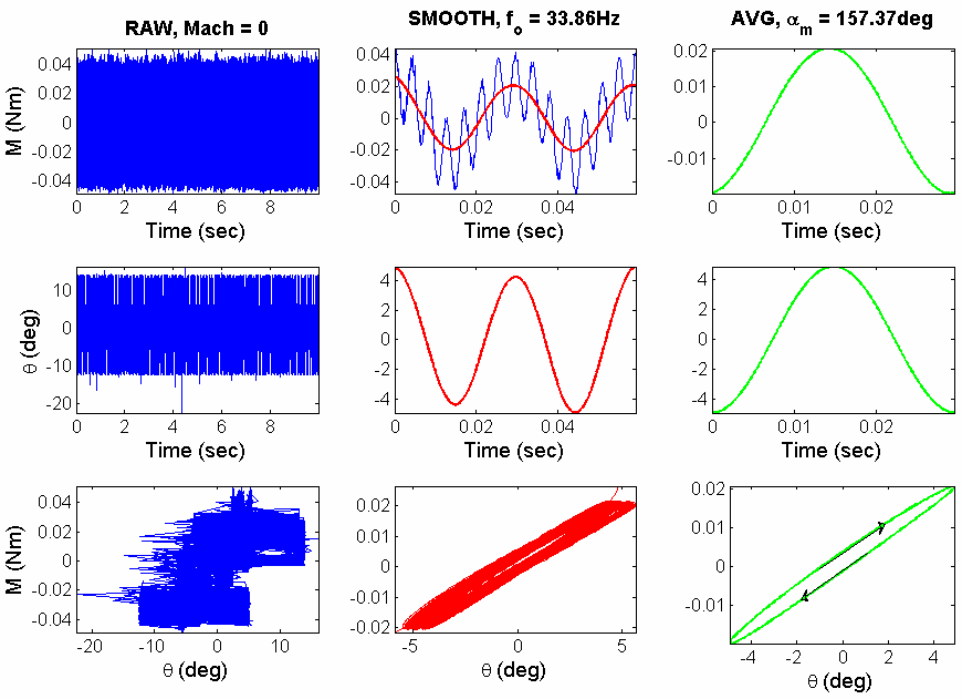

Figure 32 Apollo forced oscillations (no flow, f_o = 33.9 Hz)

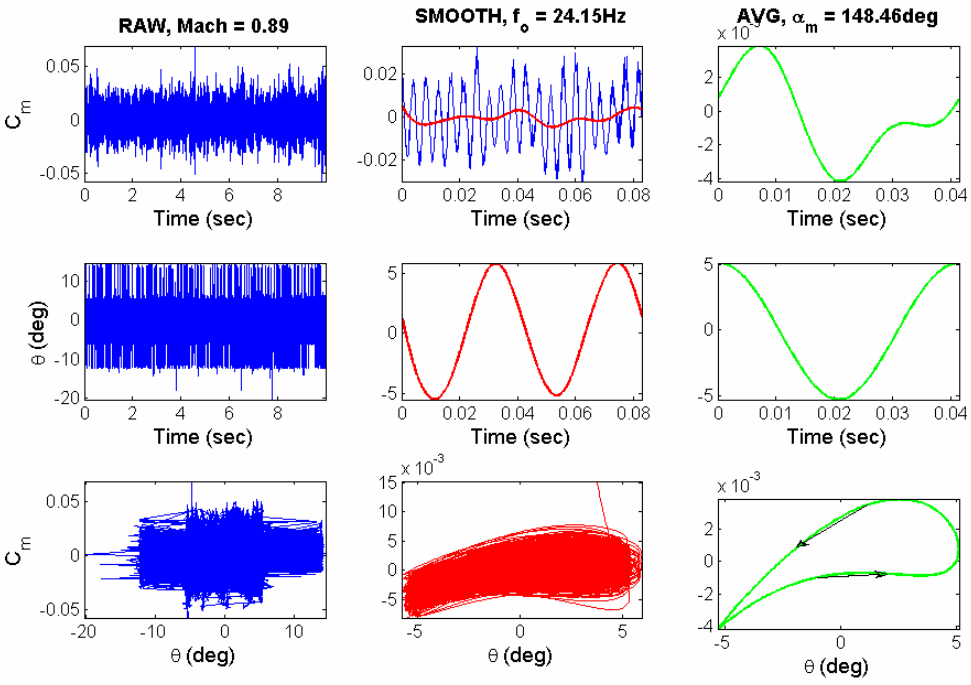

Figure 33 Apollo forced oscillations (Mach = 0.89, f_o = 24.2 Hz)

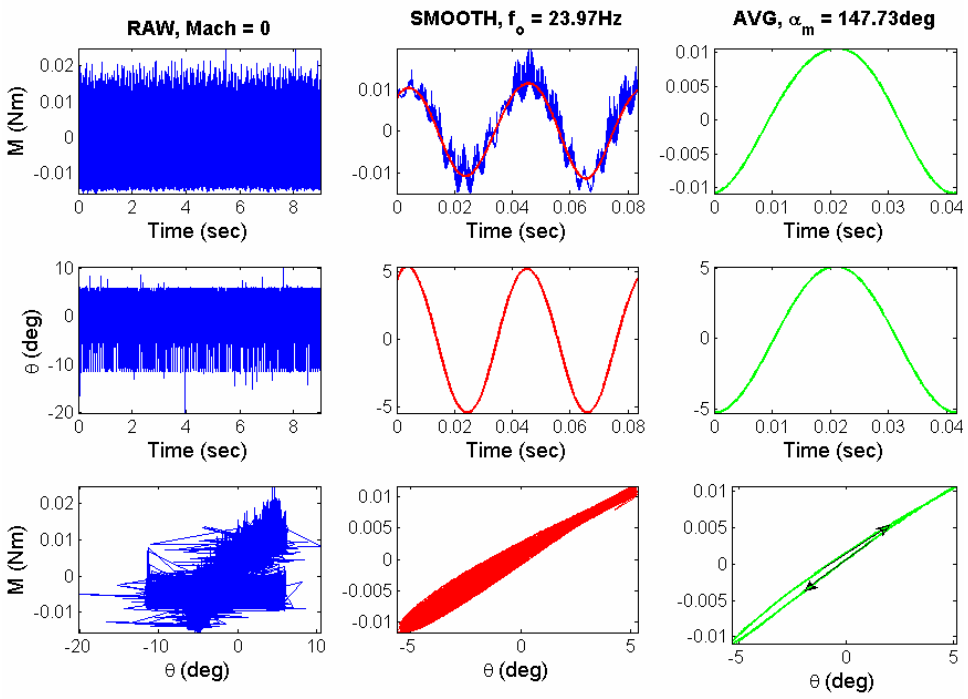

Figure 34 Apollo forced oscillations (no flow, f_o = 24 Hz)

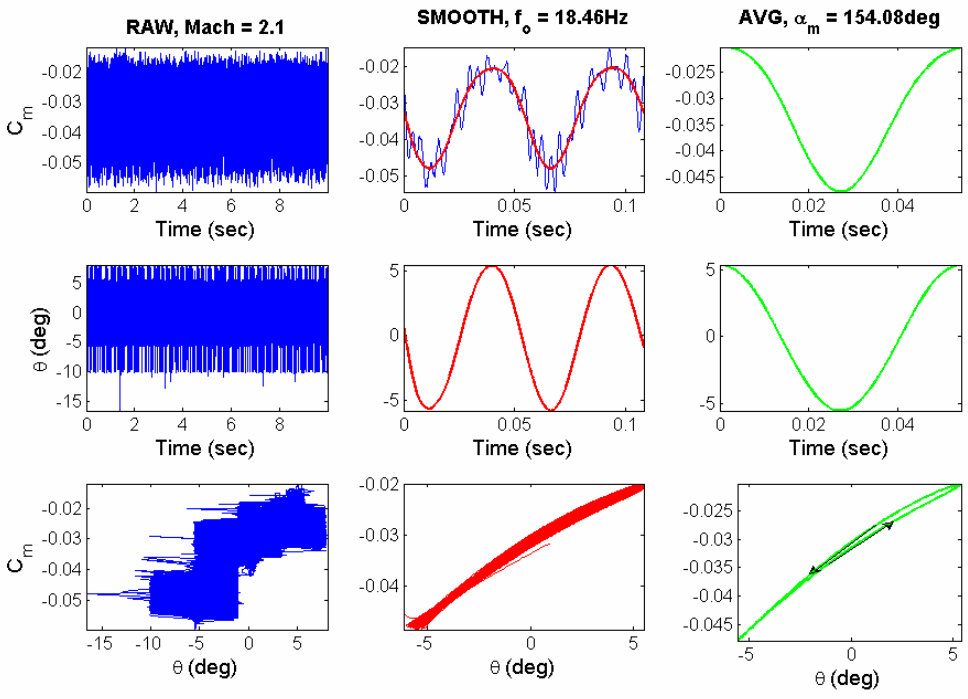

Figure 35 Apollo forced oscillations (Mach = 2.1, f_o = 18.5 Hz)

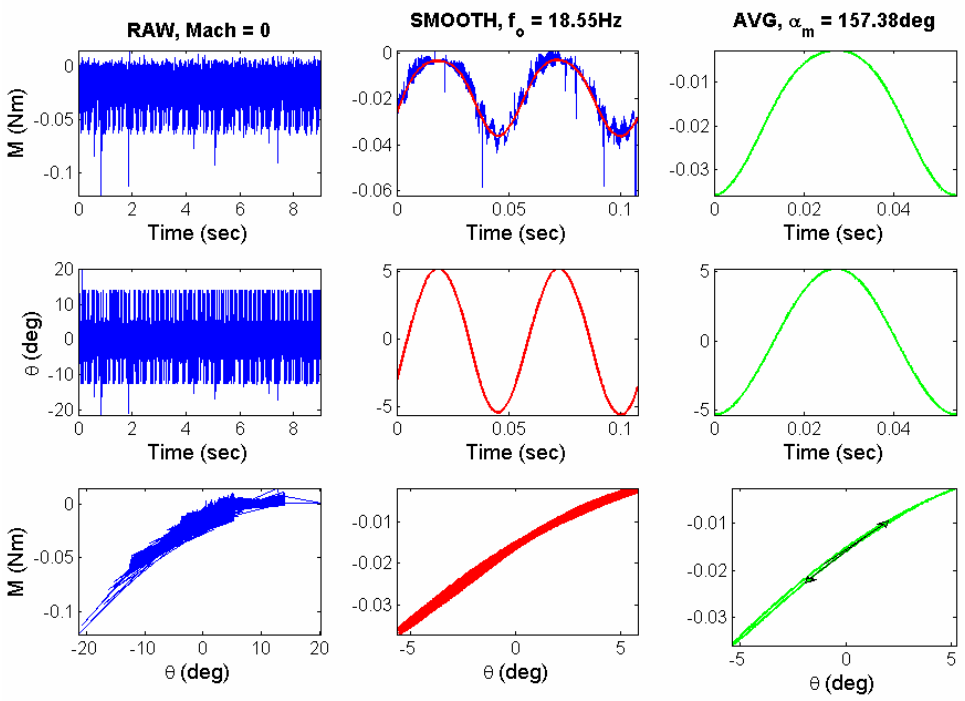

Figure 36 Apollo forced oscillations (no flow, f_o = 18.6 Hz)

The resulting data set shows the chaotic raw signal transfer to smoothed signal (shown for two periods) and the smoothed signal transfer to the ensemble average values. In going from the raw signal to the smoothed signal, frequencies on the order of 250-280 Hz and 2500-3000 Hz are attenuated. The 2500-3000 Hz signal was attributed to electromagnetic interference caused by the motor drive inverter. The source of the 250-280 Hz signal is unknown but believed to be due to the interaction of vortex streaks from the model with those from the support axis rod, which induces natural resonation of the model and support axis set-up based on the system's inertia and mass properties.

The Apollo target frequencies for Strouhal number matching and closest matched values representing the previous figure data set are given in Table 14.

Table 14 Mismatch between Apollo target Mach number and target forced oscillation frequency with actual forced oscillation tests

Target		FO Test	
Mach	***f_o (Hz)***	***Mach***	***f_o (Hz)***
0.5	29.8	0.5	30.1
0.7	33.6	0.69	33.8
0.89	23.6	0.89	24.2
2.0	18.9	2.1	18.5

3.5.3.2 Expert at $\Phi = 0^o$

The flow conditions for the Expert at $\Phi = 0^o$ forced oscillation test campaign are given in Table 15.

Table 15 Expert at $\Phi = 0^o$ test conditions for forced oscillation measurements

Mach	***P (Pa)***	***T (K)***	***U_∞ (m/s)***	***q_∞ (Pa)***	***Re (10^6/m)***
0.50	22240	287	168	3820	2.51
0.70	18790	281	236	6470	3.08
0.88	16420	279	294	8870	3.41
2.12	1640	161	539	5170	1.75

Only data for the closest match of vehicle and model Strouhal numbers at each test Mach number are given in this section along with corresponding no flow data (refer to Table 12). The exception to this is the data for the Mach = 2.12 case. The motor could only withstand up to a certain oscillation frequency given the size of the model along with the desire to preserve the motor. All other results are given in the appendix.

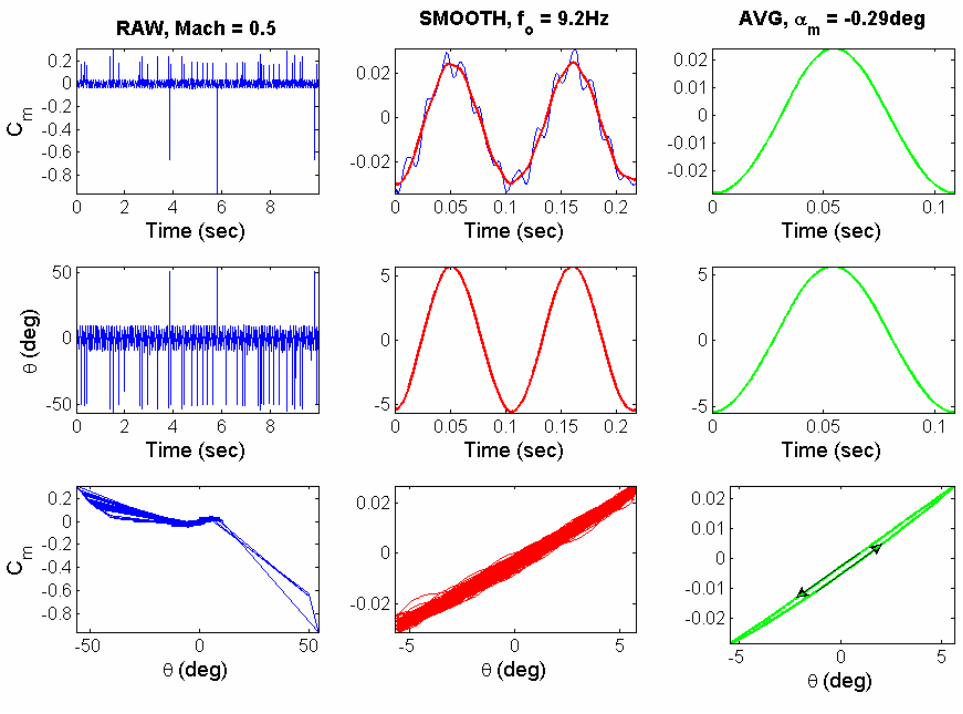

Figure 37 Expert at Φ = 0° forced oscillations (Mach = 0.5, f_o = 9.2 Hz)

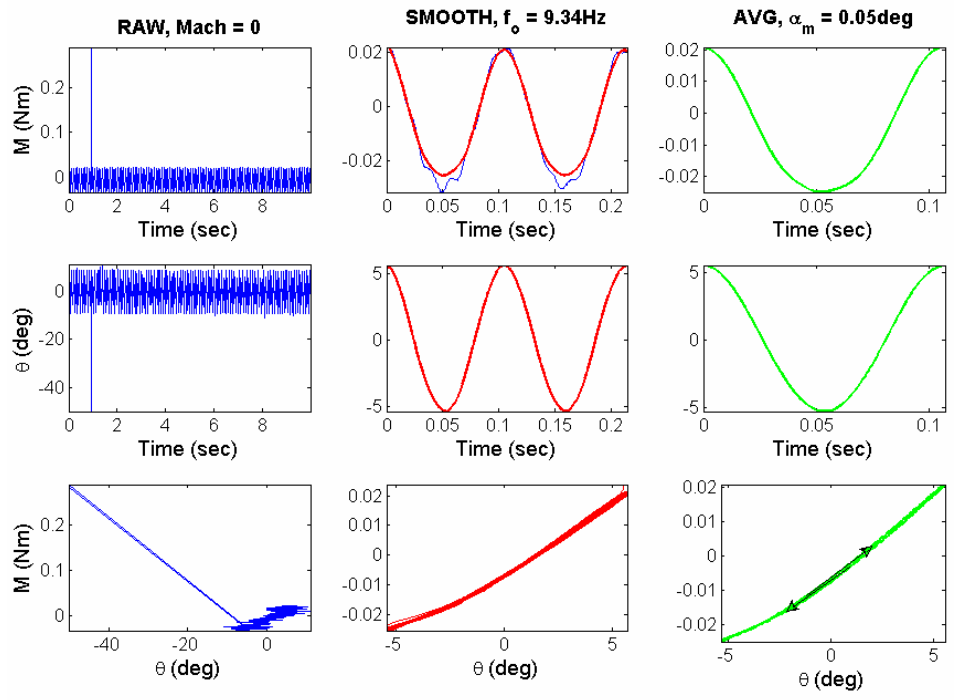

Figure 38 Expert at Φ = 0° forced oscillations (no flow, f_o = 9.3 Hz)

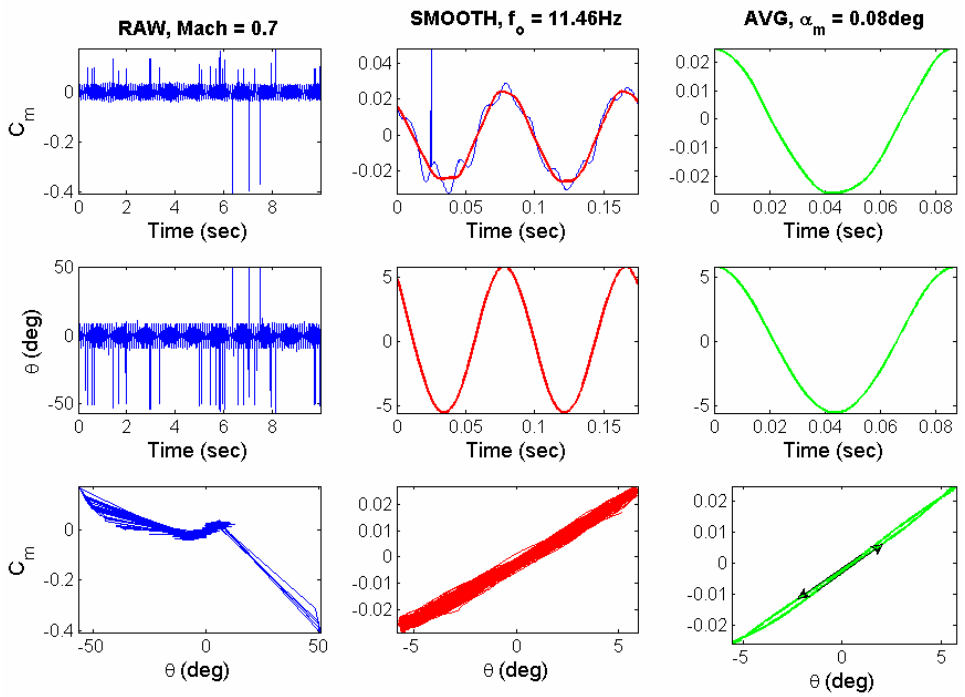

Figure 39 Expert at Φ = 0° forced oscillations (Mach = 0.7, f_o = 11.5 Hz)

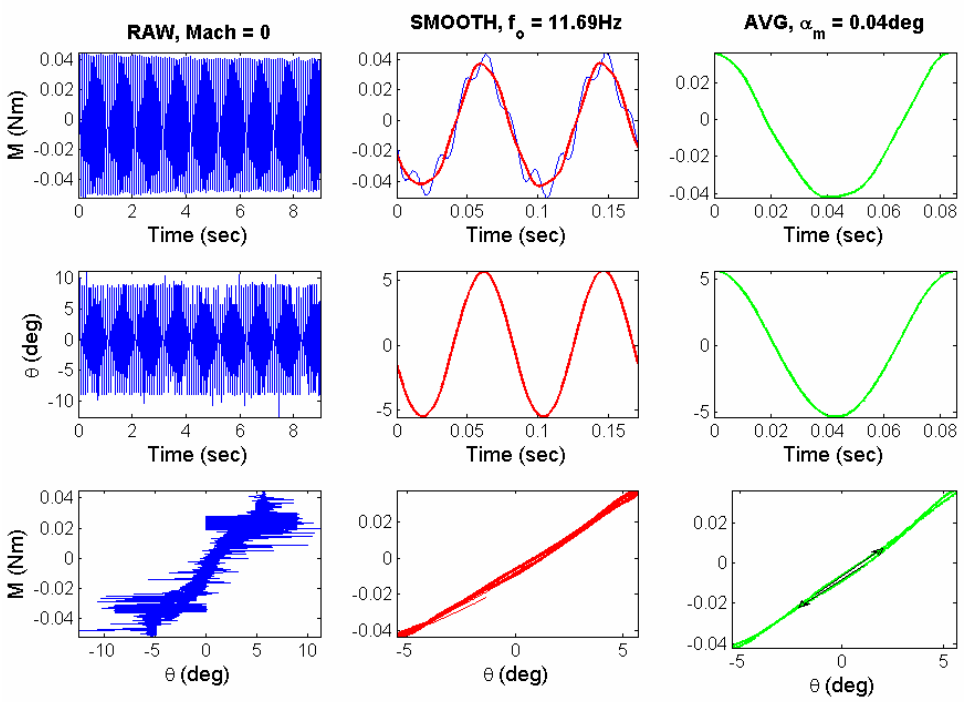

Figure 40 Expert at Φ = 0° forced oscillations (no flow, f_o = 11.7 Hz)

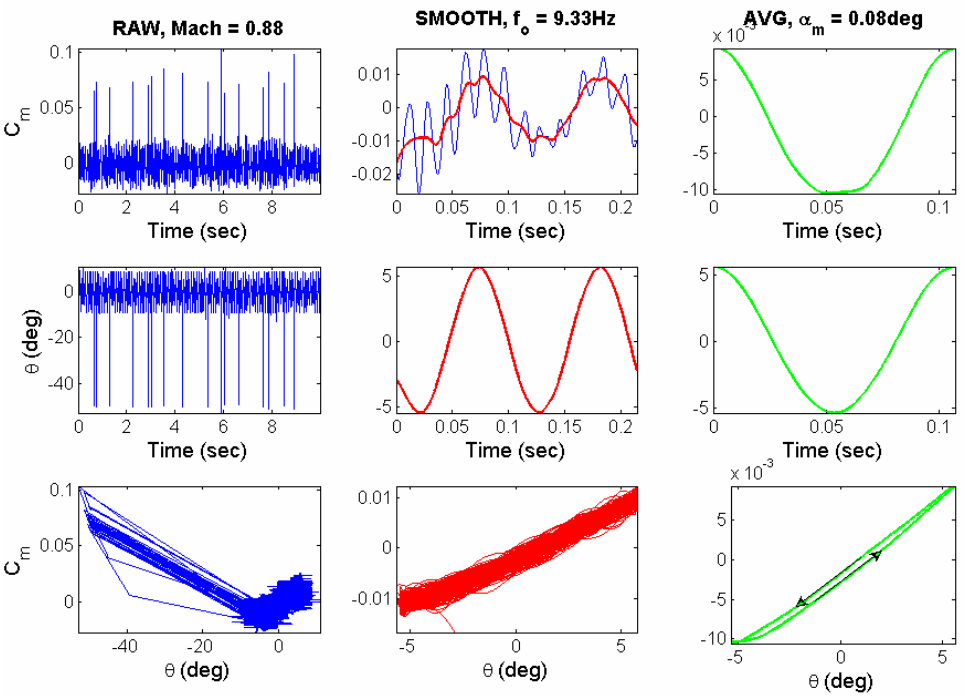

Figure 41 Expert at Φ = 0° forced oscillations (Mach = 0.88, f_o = 9.3 Hz)

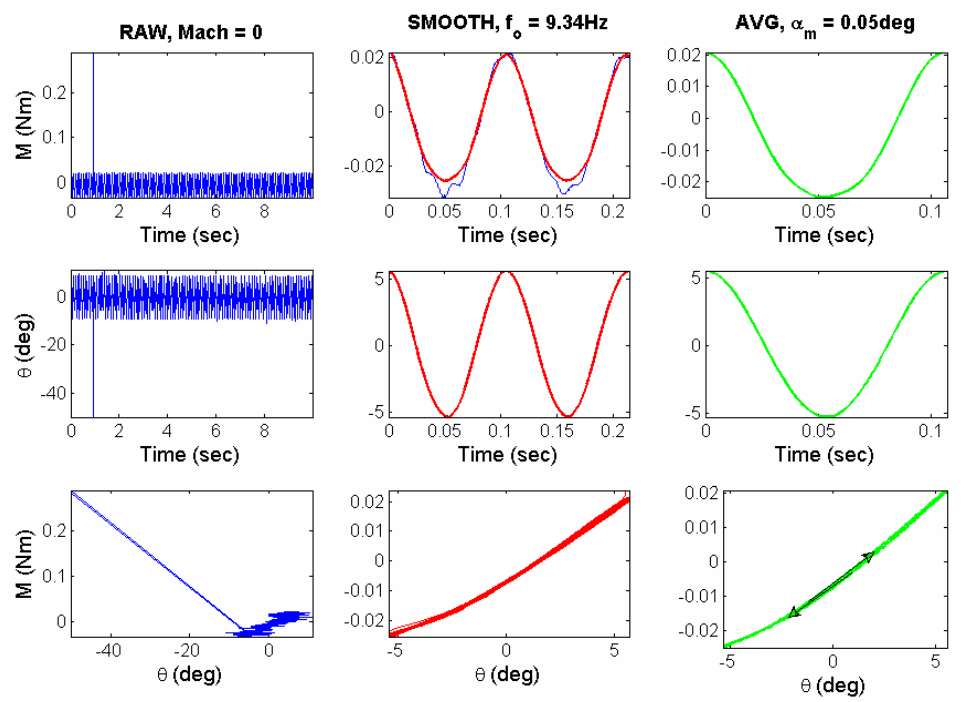

Figure 42 Expert at Φ = 0° forced oscillations (no flow, f_o = 9.3 Hz)

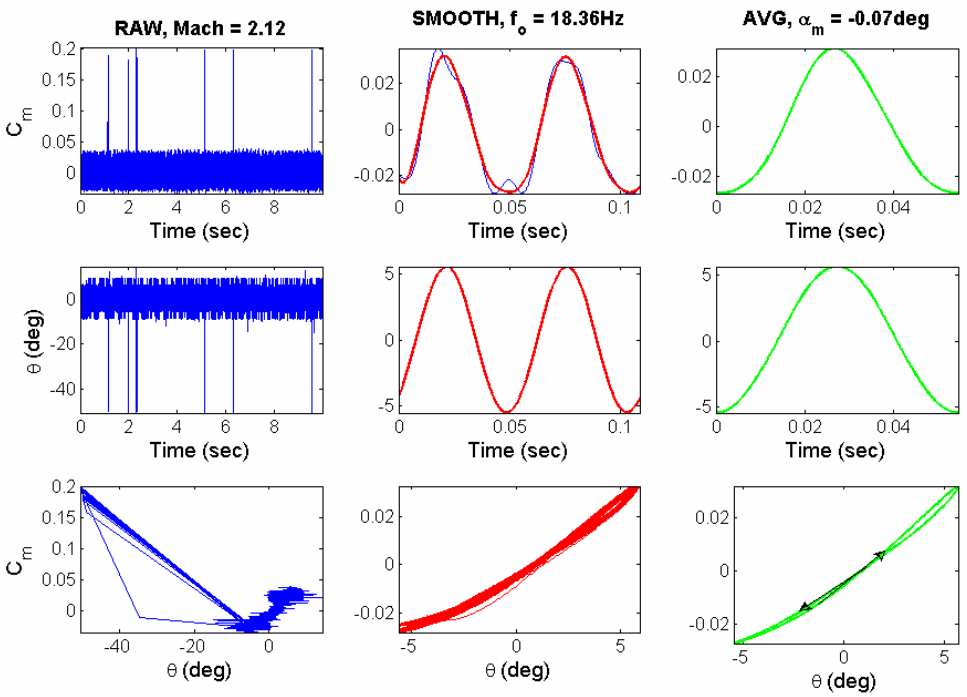

Figure 43 Expert at Φ = 0° forced oscillations (Mach = 2.12, f_o = 18.4 Hz)

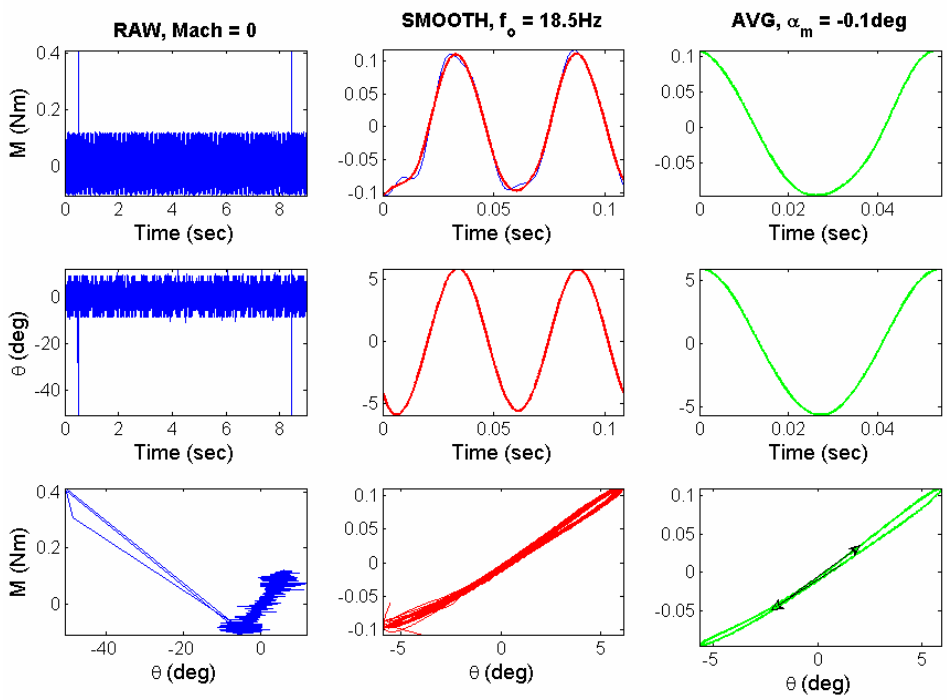

Figure 44 Expert at Φ = 0° forced oscillations (no flow, f_o = 18.5 Hz)

Similar to the Apollo, the resulting Expert data set shows the transfer from the raw signal to the ensemble averaged signal. In going from the raw signal to the smoothed signal, interference frequencies on the order of 70-80 Hz are attenuated. It is believed that the 70-80 Hz signal is a result of the interaction of vortex streaks from the model with those from the support axis rod, which induces natural resonation of the model and support axis set-up based on the system's inertia and mass properties. This same postulation was made for the Apollo. The difference in interference frequencies comes from the difference in model sizes. The inertia and physical size of the Expert model is greater than the Apollo model, leading to lower frequencies in vortex streak shedding as well as lower natural frequency for the Expert and consequently the lower interference signal frequencies. This is proven by examining the ratio of the resonation frequencies of the Apollo (250-280 Hz) to the Expert (70-80 Hz) and comparing with the square root of the ratio of the Expert with support axis inertia (70.283E-6 kg-m^2) to Apollo with support axis inertia (4.769E-6 kg-m^2). The result is: 270Hz / 70Hz = 3.86 and (70.283E-6 kg-m^2 / 4.769E-6 kg-m^2)^(1/2) = 3.84. As shown by comparing the two resulting values, they are nearly identical. Thus, the postulation that there is interference caused by the natural frequency resonation of the model and support axis set-up is valid.

The Expert target frequencies for Strouhal number matching and closest matched values representing the previous figure data set are given in Table 16.

Table 16 Mismatch between Expert at Φ = 0° target Mach number and target forced oscillation frequency with actual forced oscillation tests

Target		FO Test	
Mach	***f_o (Hz)***	***Mach***	***f_o (Hz)***
0.5	9.9	0.5	9.2
0.7	11.8	0.7	11.5
0.88	9.5	0.88	9.3
2.09	22.8	2.12	18.4

4. ANALYSIS OF RESULTS

This section addresses the post-processing of data obtained from the forced oscillation tests to extract the aerodynamic damping-in-pitch parameter. The methods as described in Section 2.3.1 are utilized. A single test case using the free-to-tumble technique obtained by Emilie, [17], along with a simulated test case are also analyzed based on the methods described in Section 2.2.1.

4.1 Forced Oscillations

As seen in Section 3.5.3, the raw data is not well-posed (due to high levels of noise) for carrying out the methods of Section 2.3.1, but the ensemble averaged values are. Thus, they are used for analytical purposes. Only near equivalent test model and flight vehicle Strouhal number cases are used for analysis to obtain comprehensive results.

4.1.1 Algebraic Method Analysis

This method is based on Equation 20 and broken into two sub-methods as described in Section 2.3.1.1. The benefit of this method is that the aerodynamic damping-in-pitch parameter is calculated over the range of oscillation angle about the model trim angle of attack whereas the other two methods are evaluated at the trim angle only. The ensemble averaged C_m moment coefficient values are converted back to actual moments for flow results. Using the Matlab function *func_nth_derivative.m* (given in the appendix), the values of oscillation angular velocity, $\dot{\theta}_f$, and oscillation angular acceleration, $\ddot{\theta}_f$, are determined over the entire oscillation range. Once evaluated, all required variables including model and axis geometries and test conditions to extract the aerodynamic damping-in-pitch parameter in Equation 20 are known. Results of the two sub-method approaches are plotted against one another. The results are given in Figure 45.

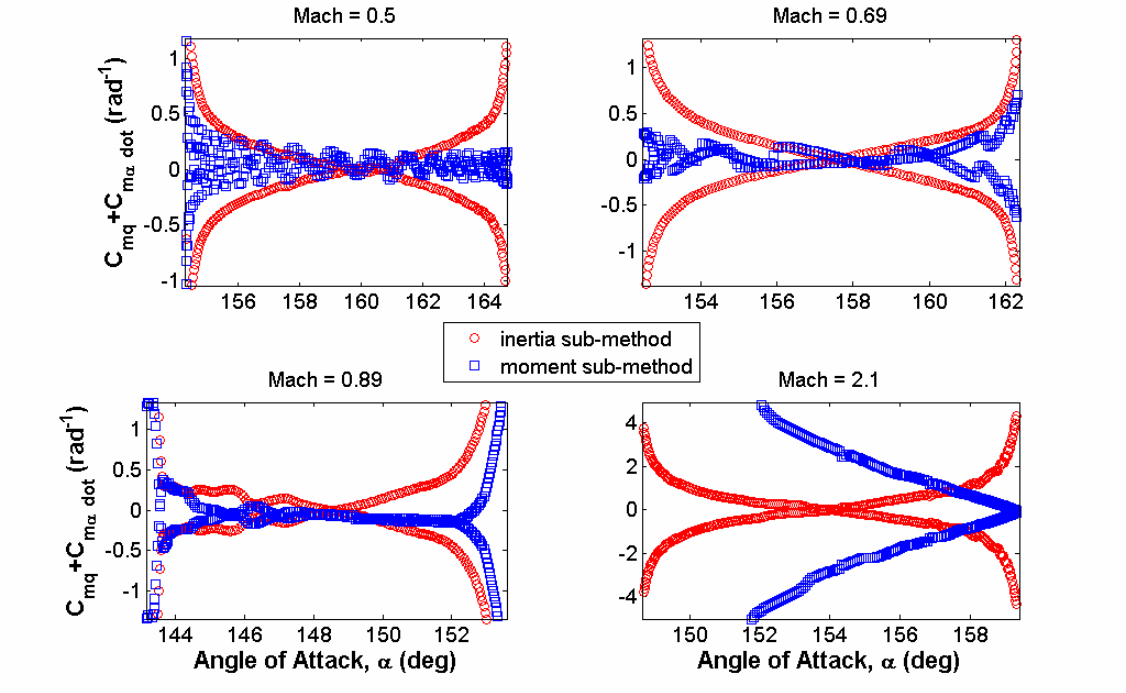

Figure 45 Apollo forced oscillation algebraic method results

4.1.1.1 Discussion

The results in Figure 45 do not show an exact correlation between the moment and inertia sub-methods for the algebra method. An explanation for this comes from moment uncertainty. Looking at Figure 11 and Figure 30 as an example, the no flow smoothed moment data shows magnitudes of approximately 0.015 Nm. This corresponds to an approximate uncertainty of 0.000725 Nm. This is near a 5% uncertainty, whereas the uncertainty associated with the inertia method (based on the incidence angle) correlating to the encoder accuracy, smoothing accuracy, and derivative accuracy is approximately 10%. Given this along with dynamic pressure uncertainties (see Figure 15), the error bands from each method would overlap one another (except at points near the extremities of attack angles). Another interesting aspect in Figure 45 is that the inertia curves (and the moment curves in somewhat like-manner) are almost symmetric about zero aerodynamic damping-in-pitch. The fact that they are not perfectly symmetric is due to the moment hysteresis. Thus, all one can truly infer from this method is the amount of hysteresis present in the system without any definitive conclusions about the actual aerodynamic damping-in-pitch parameter. Thus, this method should not be used for future investigations

and is not used to compare against published data, nor is it used to analyze the resulting Expert forced oscillation data.

4.1.2 Phase-Difference Amplitude Method Analysis

This method is described in detail in Section 2.3.1.2. Either the sub-method using theoretical forcing function based on the forced oscillation set-up (see Figure 28), {'unlumped'}, or the alternate sub-method using no flow condition data, {'lumped'}, can be used. Both sub-methods are investigated here with a sample case from the data given in Figure 29. A Matlab routine *func_forcing_moment.m* is created to evaluate oscillation angle and theoretical forcing moment based on the forced oscillation mechanical system. Inputs of system inertia, oscillation frequency, and oscillation amplitude are required. The frequency of oscillation input is chosen to be the same as the no flow condition oscillation frequency. The oscillation amplitude input is the same as the resulting amplitude from the actual no flow condition tests.

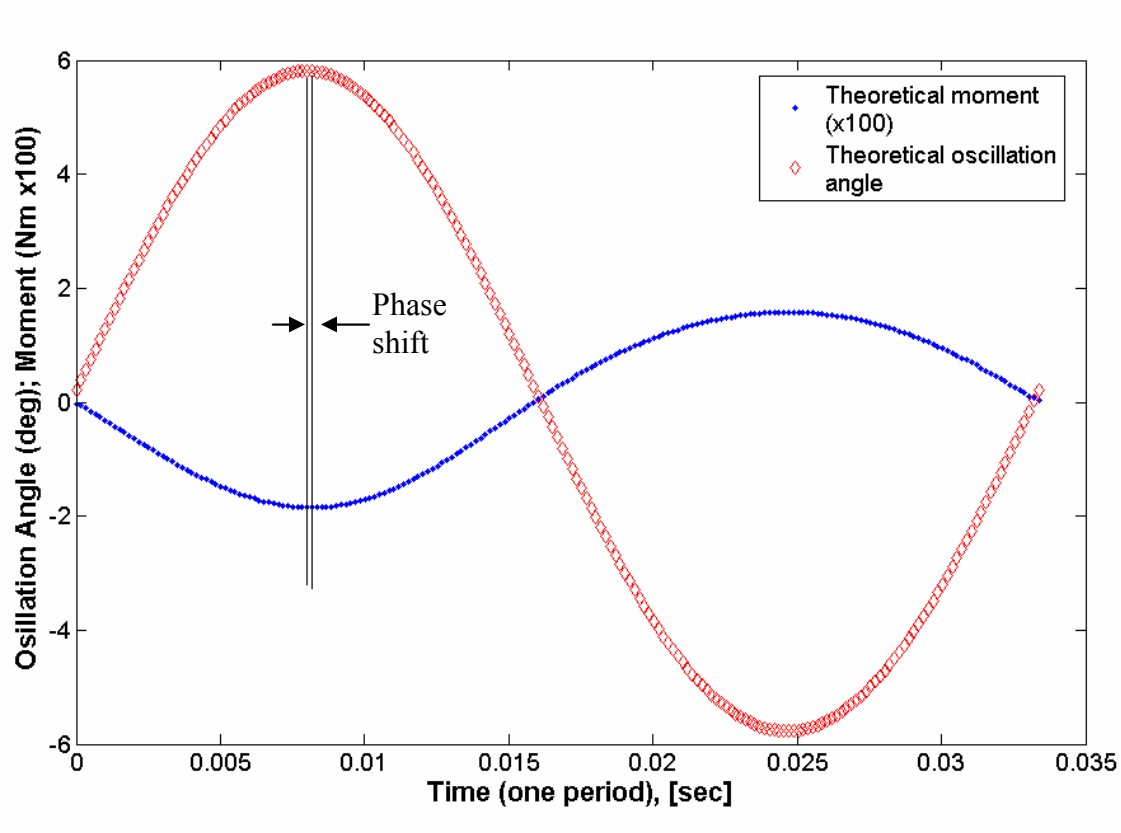

Figure 46 Theoretical forcing moment function and oscillation angle for Figure 29 test conditions based on mechanical set-up

The ensemble averaged oscillation angle and moment data in Figure 29 is plotted against one another in Figure 47 to show as an example.

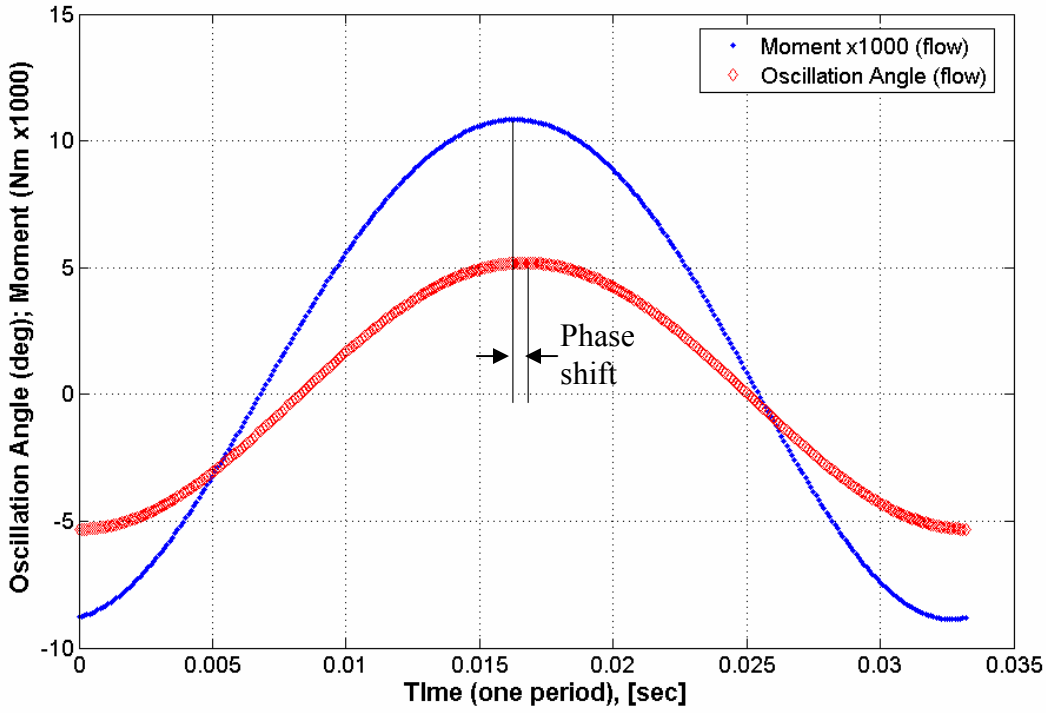

Figure 47 Apollo forced oscillation averages of moment and oscillation angle (Mach = 0.5, f_o = 30.1 Hz)

Using Equation 31 and Equation 34, the resulting values based on theoretical forcing moment are obtained {'unlumped'}. Using Equation 35 and Equation 36, the resulting values based on flow and no flow forced oscillation test results are obtained {'lumped'}. The benefit of this method is that the aerodynamic stiffness-in-pitch is obtained. From this, the theoretical value for $\left(\partial C_m / \partial \theta\right)$ can be extracted and compared with the values obtained from the static moment tests. Results yielding similar values to the test values will validate or invalidate the method. The phase-difference amplitude method results for all test cases (at vehicle/model matched Strouhal numbers) are tabulated in Table 17 and Table 18.

Table 17 Forced oscillation phase-difference amplitude method results for Apollo

Method	*Mach*	*α (deg)*	$dC_m/d\theta$ (rad^{-1})	$C_{mq}+C_{m\,\alpha\_dot}$ (rad^{-1})
Lumped	0.50	159.6	-0.140	0.840
Unlumped			0.720	0.589
Lumped	0.69	157.4	-0.200	0.409
Unlumped			0.513	0.174
Lumped	0.89	148.5	-0.172	1.836
Unlumped			0.083	1.684
Lumped	2.1	154.1	-0.826	1.537
Unlumped			0.107	0.823

Table 18 Forced oscillation phase-difference amplitude method results for Expert at Φ = 0°

Method	*Mach*	*α (deg)*	$dC_m/d\theta$ (rad^{-1})	$C_{mq}+C_{m\,\alpha\_dot}$ (rad^{-1})
Lumped	0.50	-0.3	-0.418	0.533
Unlumped			-0.114	0.387
Lumped	0.70	0.1	0.111	1.303
Unlumped			0.402	1.151
Lumped	0.88	0.1	-0.168	-0.580
Unlumped			-0.038	-0.690
Lumped	2.12	-0.1	-0.767	-0.345
Unlumped			0.148	-1.030

4.1.2.1 Discussion

The results shown in Table 17 show some promising outcomes as well as some discrepancies. Data in Table 18 will be credited or discredited base on example analysis for the Apollo. Take the Apollo, Mach = 0.5 data from Figure 29 along with the corresponding test conditions (Table 13) and the geometry (Table 3) as an example. It is seen that converting back to moment values from the C_m ensemble averaged data that the moment magnitudes are on the order of 0.01 Nm. According to the uncertainty curve given in Figure 11, this yields an approximate uncertainty of 0.000725 Nm, which is nearly 7.5% uncertainty. And yet, for the case of Mach = 0.89 (Figure 33), the moment magnitudes are on the order of 0.0035 Nm. This yields an approximate uncertainty of 0.000725 Nm. This is a 21% uncertainty! This is certainly not on the same order as the uncertainty for the Mach = 0.5 case. The same can be done for the other Mach value cases as

well. The Mach = 0.69 leads to a moment magnitude of about 0.009 Nm, which corresponds to 0.000725 Nm uncertainty, which is 8%. The Mach = 2.1 leads to a moment magnitude of about 0.0034 Nm, which corresponds to 0.000725 Nm uncertainty, which is 21.5%. The discrepancies in the results come in the comparison of the $\left(\partial C_m / \partial \theta\right)$ values to the static moment test result values (given in Table 9 and Table 10 for a mean angle of attack). For the Apollo, the 'lumped' sub-method results show at least commonality in the same sign to the test results. The value for Mach = 0.7 is almost the exact same as the test results, the Mach = 0.5 value is 22% off the test value, the Mach = 0.9 value is 72% off the test value, and the Mach = 2.12 is more than 600% off the test value. The 'unlumped' results don't even share the same sign as the test results. The explanation for the poorly matched data for this sub-method is likely due to the theoretical forcing moment function and that it does not truly match the actual forcing moment that occurred during forced oscillation testing. Just a slight difference between the two can make a sign change, and this is evident from what is given in Table 17. Examination of Expert data provided in Table 18 shows large discrepancies of the $\left(\partial C_m / \partial \theta\right)$ values to the static moment test result values (given in Table 10). There is overwhelming evidence that suggests the invalidity of this method. Finalized results given in Section 4.3 depict this method's resultant values for the Apollo only for mere visual comparison purposes to published data.

4.1.3 Moment Hysteresis-Energy Method Analysis

Details of this method are described in Section 2.3.1.3. Ensemble averaged hysteresis loops for the Apollo and Expert at $\Phi = 0^o$ corresponding to the flow condition data from Figure 29 to Figure 44 are depicted in the following figures.

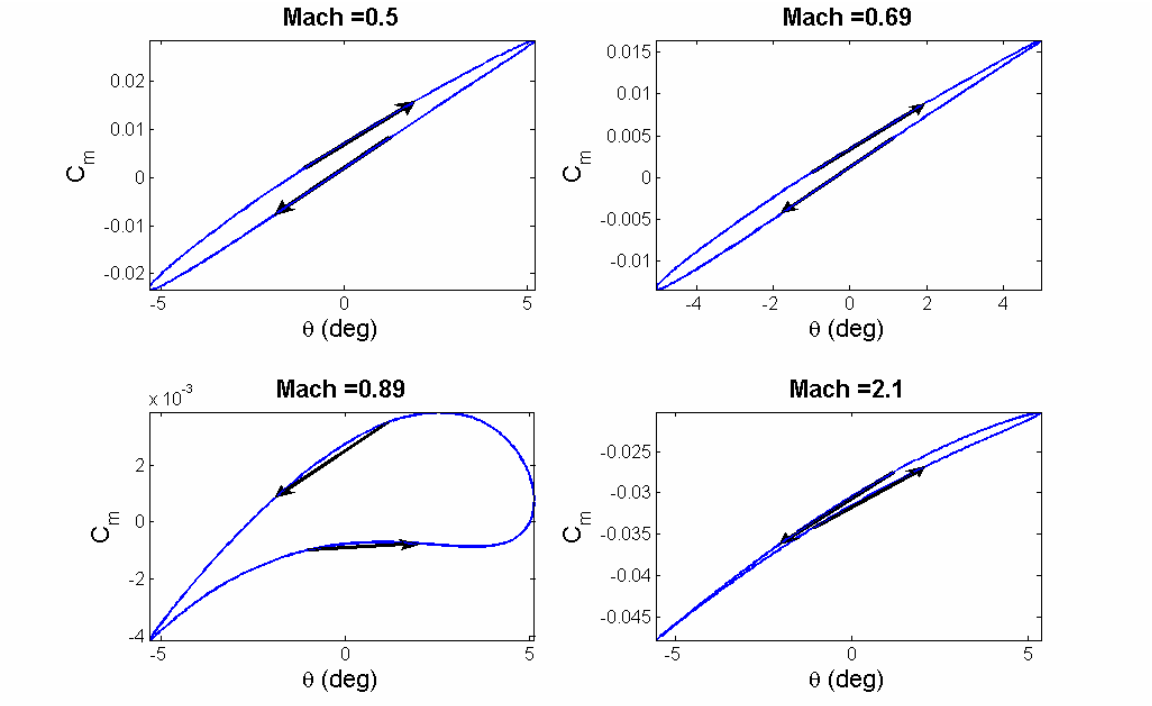

Figure 48 Moment hysteresis for Apollo forced oscillation tests

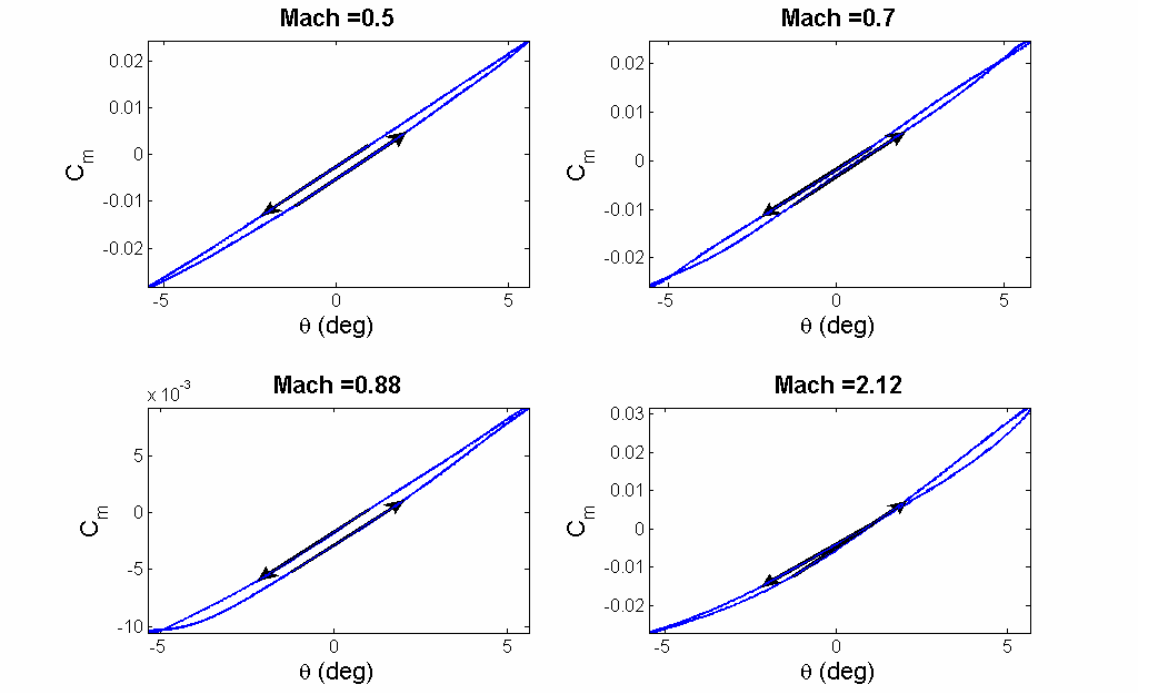

Figure 49 Moment hysteresis for Expert at Φ = 0° forced oscillation tests

The aerodynamic damping-in-pitch parameter for this method is determined using the *func_fo_energy.m* Matlab routine (defined by Equation 40 and Equation 41) in conjunction with test conditions, model and axis geometries, and static test results. The results are tabulated in Table 19 and Table 20.

Table 19 Forced oscillation moment hysteresis-energy method results for Apollo

Mach	α *(deg)*	$C_{mq}+C_{m\,\alpha\_dot}$ *(rad^{-1})*
0.50	159.7	0.559
0.69	157.4	0.280
0.89	148.5	-0.338
2.1	154.1	-0.362

Table 20 Forced oscillation moment hysteresis-energy method results for Expert at $\Phi = 0^{o}$

Mach	α *(deg)*	$C_{mq}+C_{m\,\alpha\_dot}$ *(rad^{-1})*
0.50	-0.3	-0.251
0.70	0.1	-0.162
0.88	0.1	-0.086
2.12	-0.1	0.002

4.1.3.1 Discussion

Uncertainties associated with the moments (and thus directly related with moment coefficients) are addressed in Section 4.1.2.1 for flow onset conditions. From the Apollo results given in Table 19, it is seen that the aerodynamic damping-in-pitch coefficient is positive for Mach = 0.5 and 0.69 (corresponding to negative damping), and is negative for Mach = 0.89 and Mach = 2.1 (corresponding to positive damping). The reason for this is due to the direction of the hysteresis loops shown in Figure 48. From the Expert results given in Table 20 and corresponding hysteresis loops in Figure 49, it is seen that the aerodynamic damping-in-pitch parameter is negative for all Mach numbers except for near zero at Mach = 2.12. Recall though, the data depicted in the tables are taken at the closest oscillation frequency required to match Strouhal numbers. For the Expert case of Mach = 2.12, the required oscillation frequency is 22.8 Hz and only 18.4 Hz was achieved. Thus, an extrapolation technique must be used to more accurately evaluate the aerodynamic damping-in-pitch parameter, and this is addressed in detail in Section 4.3.2.2. Overall, the results show agreement with the information given in Section 2.3.1.3. An important characteristic in the hysteresis loops is the median slope of the loops. The slope of every loop given in Figure 48

and Figure 49 is positive. Yet, examination of the slopes from the static moment tests (see Figure 25 and Figure 26) show that the Apollo's should all be negative along with the Expert's for the supersonic test condition. The reason that all slopes in the hysteresis loops are positive and do not match the slopes depicted in the static moment tests is due to the impact of moments from inertial forces of the models at high frequencies. Data provided in the appendix shows that the hysteresis loop slopes for very low oscillation frequencies are in correct agreement with the slopes based on static moment test data because data acquisition for oscillations at very low frequencies is essentially the same as acquiring during static testing. Though the hysteresis slopes do not match the static moment data, it does not invalidate this method. The direction of the hysteresis loop slope does not matter because it is merely the clockwise or counter-clockwise line path direction along the loop that defines flow field energy input or dissipation to the system. This is the basis for measuring the amount of aerodynamic damping imbued on the system and the cornerstone for this method.

4.2 Free-to-Tumble

A theoretical simulated case is fabricated and used to validate or invalidate the methods as outlined in Section 2.2.1. Prior investigation of this method by Emilie, [17] used a Runge Kutta analytical technique to extract the aerodynamic damping-in-pitch parameter. This attempt proved unsuccessful at the time and is the reason alternate methods are investigated here.

4.2.1 Simulated Case

The simulated case replicates the governing differential equation form as depicted in Equation 10 and general solution to the equation as depicted in Equation 12 through Equation 14. The coefficients of *a* and *b* are set as non-linear functions of incidence angle, θ. The functions are shown plotted against incidence angle along with the resulting theoretical data curve of incidence angle over time.

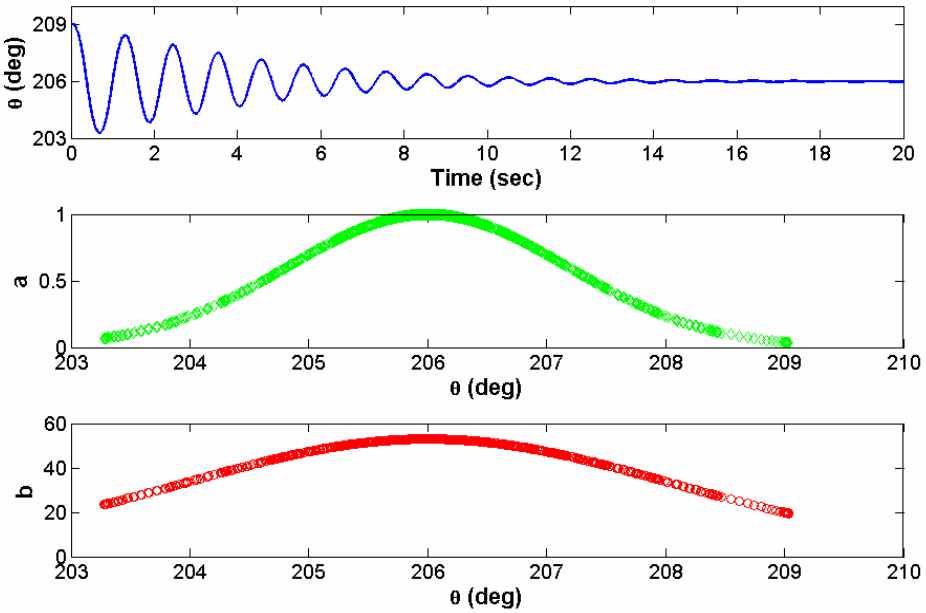

Figure 50 Free-to-tumble simulated case incidence curve with 'a' and 'b' coefficients as functions of θ

The methods outlined in Section 2.2.1 are utilized. The angular velocity, angular acceleration, and angular jerk, $\dddot{\theta}$, are extracted from the simulated incidence curve using the *func_nth_derivative.m* Matlab routine (since this is what would be done for an actual test case to obtain the derivatives). The *func_smooth.m* Matlab function (from MathWorks) is used to smooth out any random noise from the derivative calculation results (this method is also what would be done for analyzing an actual test case). Resulting values for the *a* and *b* coefficients as functions of incidence are depicted in Figure 51 and Figure 52. Note that for the algebraic method, only the *a* coefficient is considered an unknown, thus the *b* coefficient is not plotted (because it would fall directly on the simulated case's curve).

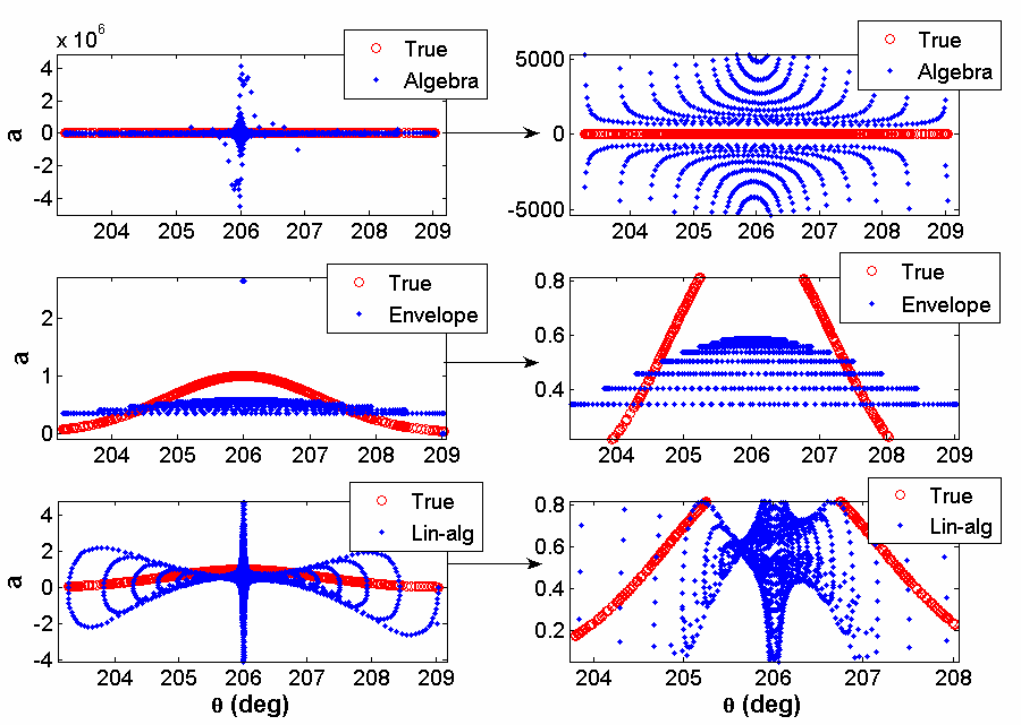

Figure 51 Free-to-tumble analysis method results for the 'a' coefficient values of the test case

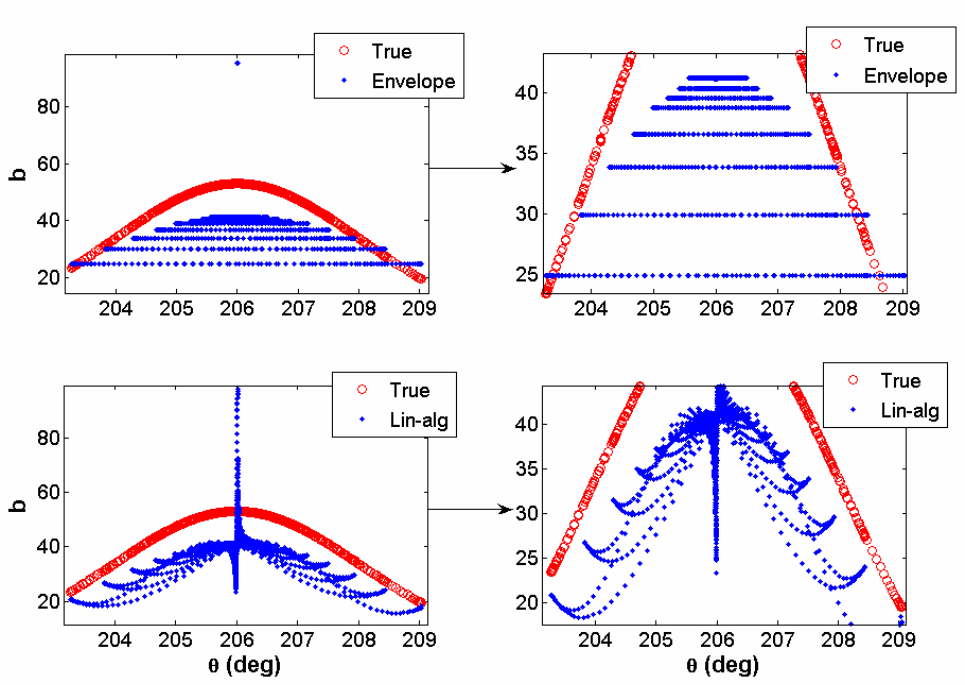

Figure 52 Free-to-tumble analysis method results for the 'b' coefficient values of the test case

4.2.1.1 Discussion

The results show that only the envelope method has some type of agreement with the correct *a* and *b* coefficient values, and even it does not match directly. The algebra method yields atrocious results for the 'a' coefficient. Reasons for this are attributed to the fact that $\dot{\theta}$ is zero when θ is either a minimum or maximum. This value lies in the denominator of Equation 11. Thus, to satisfy the equation for this method (to retain 'a'), the values of $\dot{\theta}$ and $\ddot{\theta}$ must be exact! Yet, these derivatives are calculated using a central difference scheme, leaving truncation errors. This is the reason the values 'blow up'. The linear algebra method has the same problem as the algebra method. The truncation errors cause poor numerical results. Though the envelope method does not directly correlate with the true 'a' and 'b' coefficient values, the trends are somewhat similar and are on the same order of magnitude as the true values. Looking at the envelope method results in Figure 51 and Figure 52, it is believed that the distribution of outer point values can be stretched or squished (while preserving the area under the distribution curve for energy conservation purposes) in such a manner that the results would fall on the true coefficient values. How to apply this kind of manipulation is not known and would make for a good future investigation.

4.2.2 Test Data Case

The free-to-tumble test data examined comes from Emilie, [17]. The model tested is the Apollo at Mach 2 at 50 degree release incidence angle. Geometry between the current investigation's Apollo model and the test data case model are the same. The angular coordinate convention used is reversed (180 degrees out of phase) from the one used in this project and Karatekin's experimental tests. The test data case model is made from a different material than the nanocomposite resin of the stereolithography models used in this project. Thus, the model's moment of inertia is different ($I = 8.829 \times 10^{-6}$ kg-m^2). The axis of rotation location is given in reference to model diameter, D, as x/D = 0.632 and z/D = 0.058, where x is measured from the cone apex and z from the model centerline (same as Karatekin's coordinate convention). The test conditions are given in Table 21.

Table 21 Free-to-tumble test case conditions from Emilie, [17] tests

Mach	*P (Pa)*	T_t *(K)*	U_∞ *(m/s)*	ρ *(kg/m^3)*	q_∞ *(Pa)*
2	1237	305	522	0.0254	3463

Incidence angle is recorded over time. Pitch moment data against incidence angle is translated to the given c.g. location based on NASA data. Test results and moment-incidence curves are depicted in Figure 53 and Figure 54.

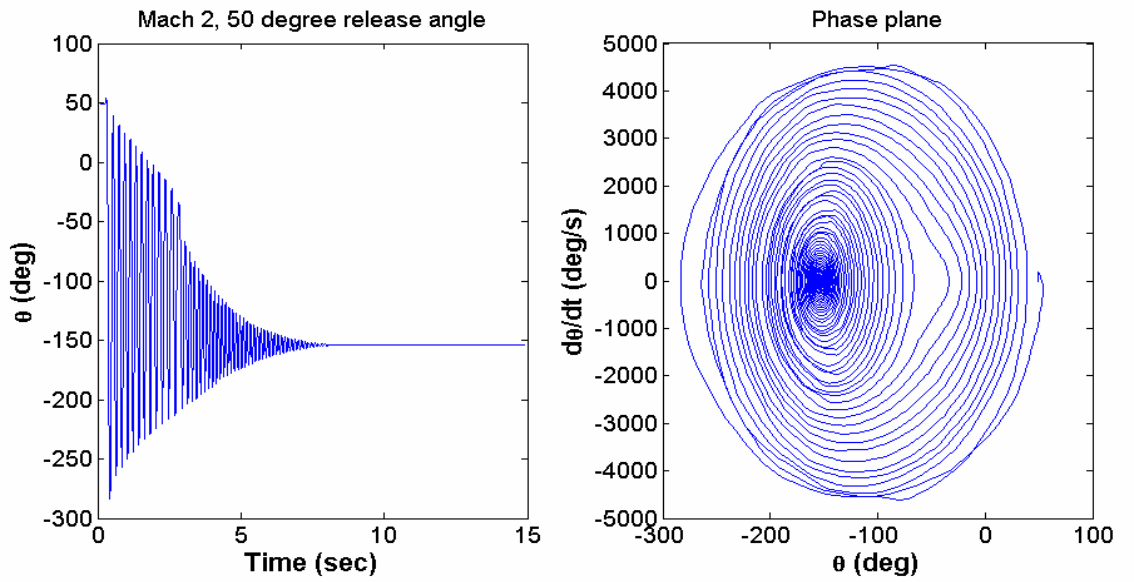

Figure 53 Free-to-tumble test case results (left) and phase plane plot (right) [17]

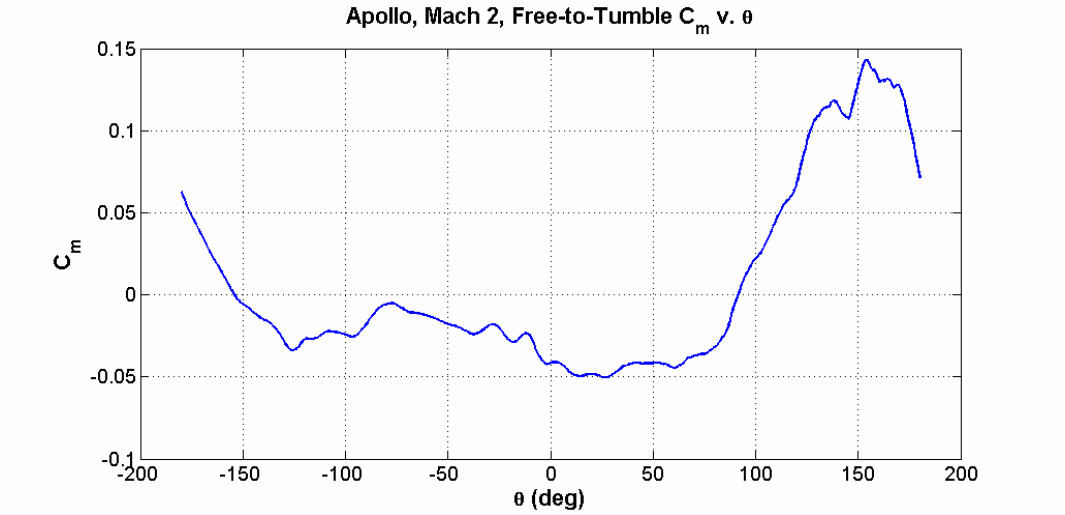

Figure 54 Free-to-tumble test case variation of pitch moment against incidence angle [17]

The test case curves are used in performing the free-to-tumble analysis methods described by Section 2.2.1. Resulting values for the aerodynamic damping-in-pitch parameter are given in Figure 55.

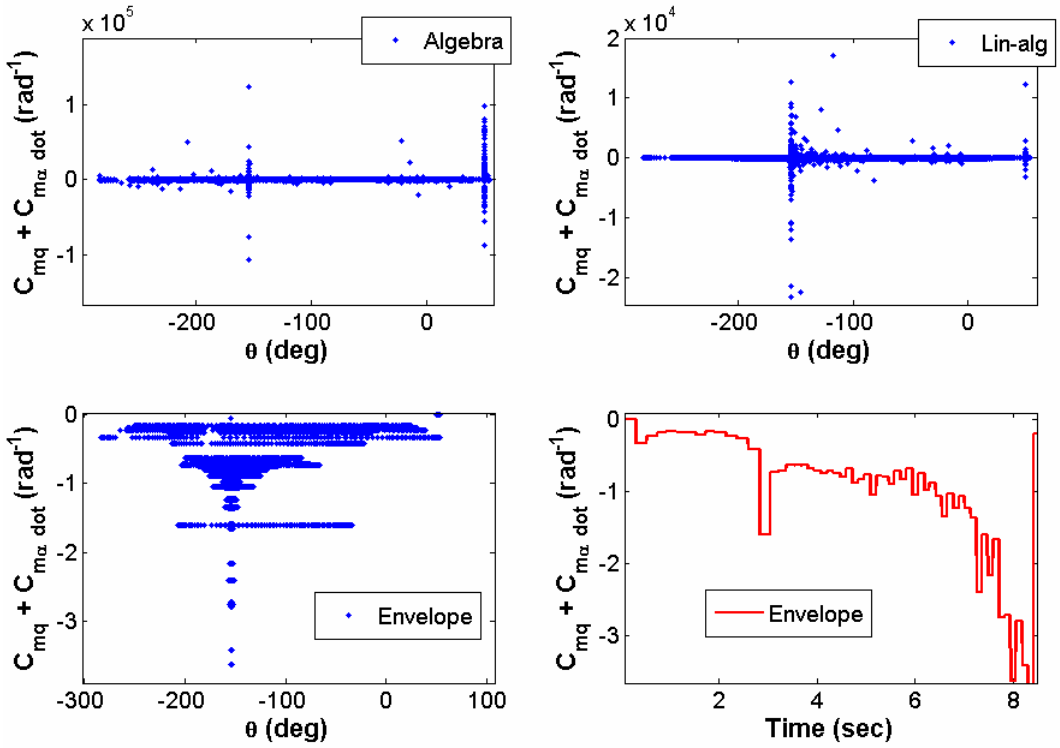

Figure 55 Free-to-tumble test case results for aerodynamic damping-in-pitch parameter as a function of incidence angle using test case angle convention

4.2.2.1 Discussion

The phase plane plot in Figure 53 shows the focal point where steady state occurs. This is also shown where C_m is zero with negative slope in Figure 54. The presence of asymmetry is evident from the test results and shown in the phase plane plots. It is caused by the eccentricity of the axis of rotation to c.g. location. Physically, the model is released and impacted with a large amount of energy. This energy allows the model to 'ride' the C_m vs. θ curve until it comes to rest at the stable angle of attack of -154 degrees (this is +206 degrees according to the coordinate convention of this report). As seen by the results and discussed in Section 4.2.1.1, both the algebra and linear algebra methods produce poor results. The envelope method shows the trend of increased damping (larger negative values for aerodynamic damping-in-pitch) as time reaches steady state at the stable angle of attack, which makes perfect sense. The amplitude jump comes from the ratio of peaks

around the time of 3 seconds and can be ignored, as this is not due to a sudden increase in aerodynamic damping but to the static restoring moments and energy left in the system at that instant.

4.3 Comparison of Results and Discussion

The comparison of results for the static moment tests and forced oscillation tests are focused on in this section. The most significant deductions for this research are presented here. The FO algebra post-processing method (Section 2.3.1.1) proved to yield inconclusive results, so it is not presented here. The results are compared to past investigations where applicable.

4.3.1 Apollo

4.3.1.1 Static Moment Tests

Based on Figure 25 and the data in Table 9, the slope of pitch moment with respect to incidence angle for all test cases (given over the investigated range of angle of attack) is seen to be similar for the subsonic flight conditions (Mach = 0.5 and 0.7), but drastically different for transonic flight condition (Mach = 0.89) as well as the supersonic condition (Mach = 2). In the transonic region, this is likely due to the highly unsteady flow environment that occurs from shock induced oscillations, which causes pressure pockets to move about the model surface (length in 2-D analysis) in a random manner. But this is only a hypothesis, and future experimental investigation would be needed to verify this phenomenon. The presence of a detached shock wave off the heat shield surface in the supersonic flow test establishes a vast characteristic change in slope compared with the other test cases. The published data for low-speed, subsonic tests at Mach < 0.15 in Karatekin, [1] gives $\partial C_m / \partial \theta$ = -0.15 rad^{-1} for the Apollo and is assumed independent of Mach number. The assumption of Mach number independence is verified by physical observation of the data depicted in Figure 25. Yet, there is a slight difference between Karatekin's value and the values obtained at the trimmed angles of attack during this test campaign, but these variances are quite minor in significance. Discrepancy is likely due to air taking on compressibility effects, which is attained when Mach values are greater than 0.3. Regardless, both values show static stability of the system. Published data from Moseley et al., [20] shows zero pitching moment for Mach = 0.9 at approximately 150 degrees, which is fairly well correlated with the data obtained in this investigation at 148.5 degrees. Also, Moseley's data of $\partial C_m / \partial \theta$ passing through this point is around -0.095 rad^{-1},

and this is also very well correlated with the static moment test value of -0.100 rad^{-1}. Lastly, the data set for Mach = 2 is compared with the results from Emilie, [17] and Moseley et al., [20]. There is no actual Mach = 2 test from Moseley, so the data is interpolated based on the given set of information. The mean angle of attack used during forced oscillation testing is used for comparison, which is 154.1 degrees. Moseley's value of pitch moment coefficient for this angle at correlated Mach number is approximately -0.025 and Emilie's value is approximately -0.023, and the static moment test data from this experiment is evaluated at -0.025. The value of $\partial C_m / \partial \theta$ is approximately -0.115 rad^{-1} for both Moseley's and Emilie's data, and the value obtained during experiments is -0.113 rad^{-1} for the given angle of attack. The static moment test data is well correlation with published data, and this provides conclusive evidence that testing was done properly. Therefore, one may infer this conclusion for the Expert static moment test cases as well.

4.3.1.2 Forced Oscillation Tests

The analysis of results for the Apollo forced oscillation tests are summarized in Table 22. Data that correlates the closest for matching Strouhal number between the flight vehicle and model is used. Recall that the algebra method is not presented here. Data from the table is then compared with published data in Figure 56.

Table 22 Comprehensive Apollo forced oscillation results (and data description table for Figure 56)

Model	*Method*	*Mach*	*St*	f_o *(Hz)*	α *(deg)*	$C_{mq}+C_{m\,\alpha\_dot}$ (rad^{-1})
Apollo	*Lumped*	0.50	0.00890	30.14	159.7	0.840
	Unlumped					0.589
	Energy					0.559
	Lumped	0.69	0.00725	33.77	157.4	0.409
	Unlumped					0.174
	Energy					0.280
	Lumped	0.89	0.00404	24.15	148.5	1.836
	Unlumped					1.684
	Energy					-0.338
	Lumped	2.1	0.00174	18.46	154.1	1.537
	Unlumped					0.823
	Energy					-0.362

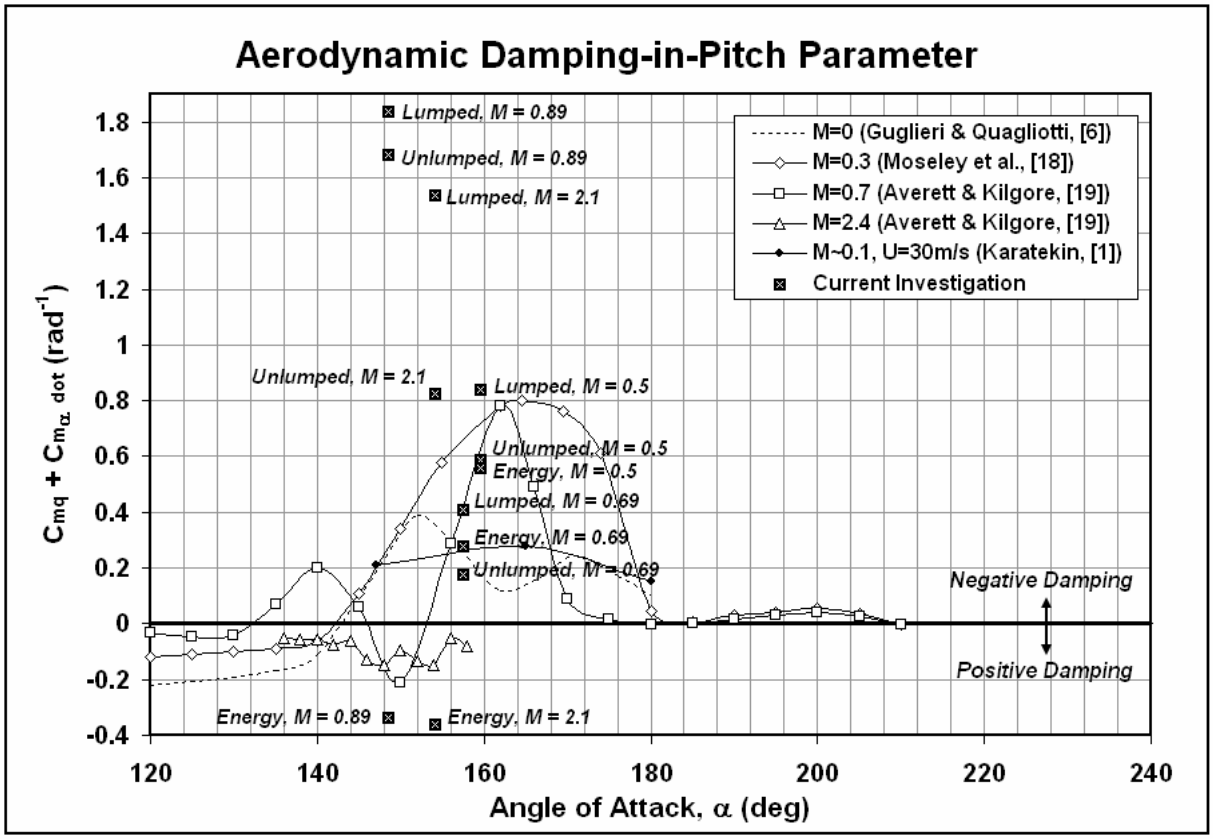

Figure 56 Comparison forced oscillation results for the Apollo aerodynamic damping-in-pitch parameter with published data based on closest Strouhal number matching

The representation of data given in Figure 56 shows reasonable correlation between all methods for the Mach = 0.5 and Mach = 0.69 cases. The 'lumped' and 'unlumped' values for Mach = 0.89 and Mach = 2.1 are quite far off from published data. Reasons for this are explained in Section 4.1.2.1. The moment hysteresis-energy method seems to show the best agreement for all subsonic, transonic, and supersonic test cases. From the prescribed results, it is believed that the moment hysteresis-energy post-processing method for forced oscillation testing provides the most accurate prediction for aerodynamic damping (at least in comparison with published data).

Although the moment hysteresis-energy method provides the best results, there are uncertainties associated with it. One is allocated to the smoothing of the raw data signal. Because phase-locking for obtaining ensemble average test curves is done on smoothed signals, performing the smoothing over a certain number of data points compared to a different number of data points may yield a slightly different curve for ensemble average. This is

investigated using the Apollo Mach = 0.5 test case and represented by Figure 57. From the figure, it can be assumed that there is a 10% uncertainty in the value of the aerodynamic damping-in-pitch parameter as a worst case scenario.

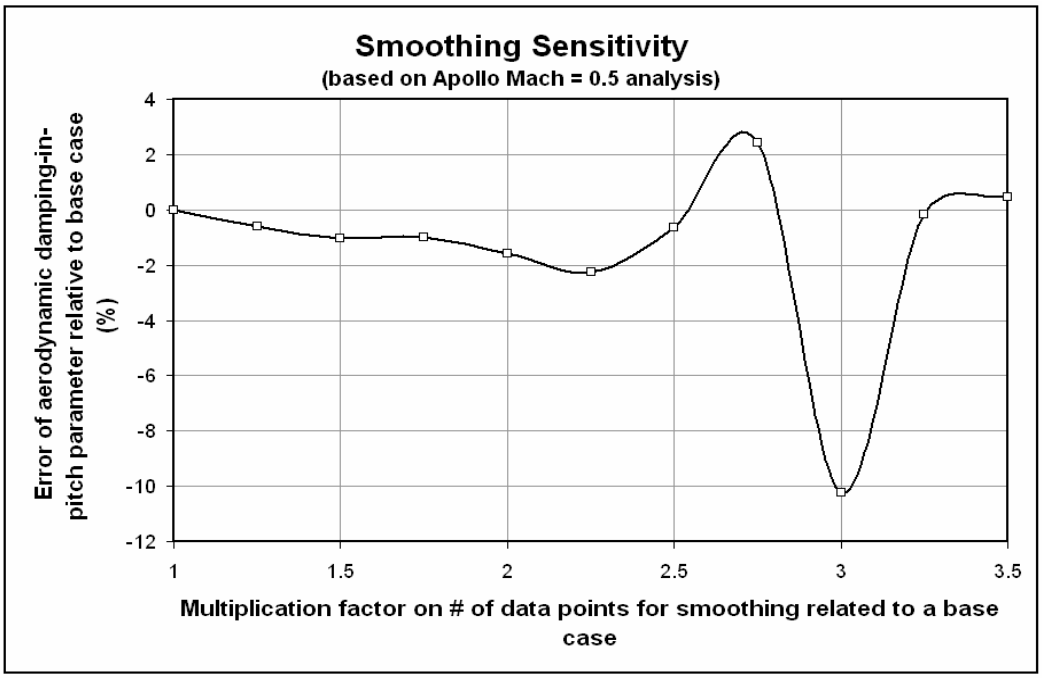

Figure 57 Sensitivity of the aerodynamic damping-in-pitch parameter to smoothing using the Apollo Mach = 0.5 forced oscillation test as case study

The other sources for uncertainty are given in the uncertainty of angle of attack based on optical encoder sensitivity and noise as well as uncertainty associated with the resulting test magnitudes of moment relative to moment uncertainty associated with the torque balance (given in Figure 11). This uncertainty contributes to uncertainty of the pitch aerodynamic damping (see Equation 40) and thus finally to uncertainty of the aerodynamic damping-in-pitch parameter (see Equation 41). Looking further at Equation 41, there is possible uncertainty associated with dynamic pressure as well as the static pitch moment slope. Looking at Figure 15, the dynamic pressure uncertainty is extremely small, thus it can be neglected. The uncertainty in pitch moment slope is also very small due to the proximity of data points taken over the range of angle of attack examined. Thus, this is neglected. Therefore, the main contributions of uncertainty are from smoothing and the test moment magnitudes that define the pitch aerodynamic damping. Both can be linearly related to the aerodynamic damping-in-pitch parameter uncertainty. Lastly, the information given in Table 14 comparing target oscillation frequencies

and Mach numbers to what was actually achieved is investigated. Due to mismatches in both, interpolation or extrapolation of the curves shown in Figure 58 are performed to ensure matches. Note that only frequencies are attempted for matching and not Mach number because there are not enough Mach number test cases to make a reasonable interpolation scheme, especially for the jump from transonic to supersonic test cases.

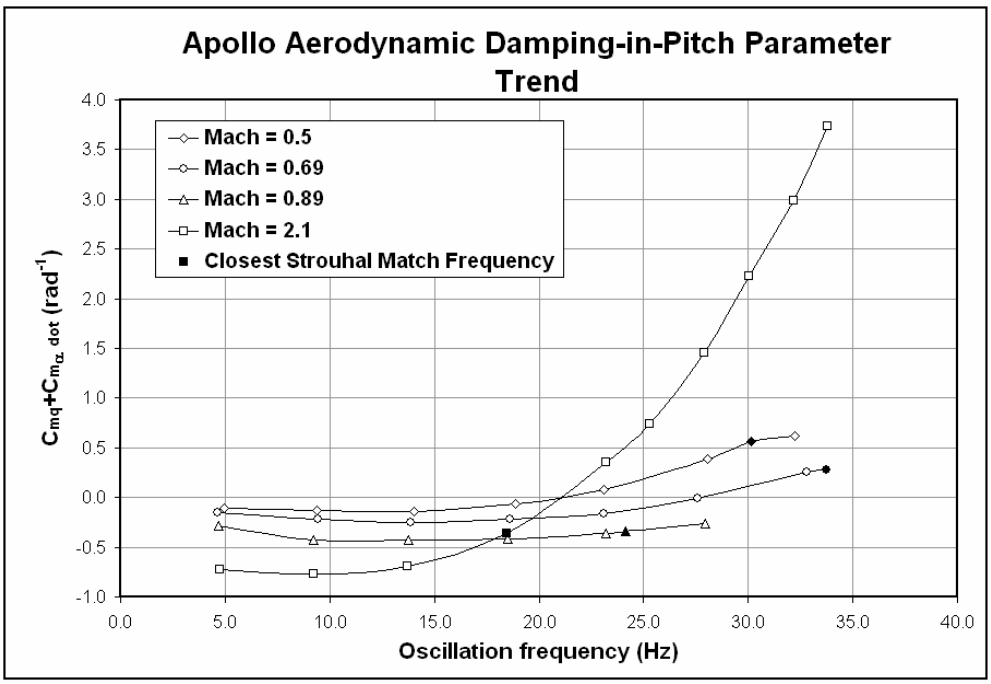

Figure 58 Apollo trend in oscillation frequency with resulting aerodynamic damping-in-pitch parameter values using the moment hysteresis-energy method

The finalized results using the interpolated values from Figure 58 along with error bands showing the range of influence of uncertainty are depicted in Figure 59.

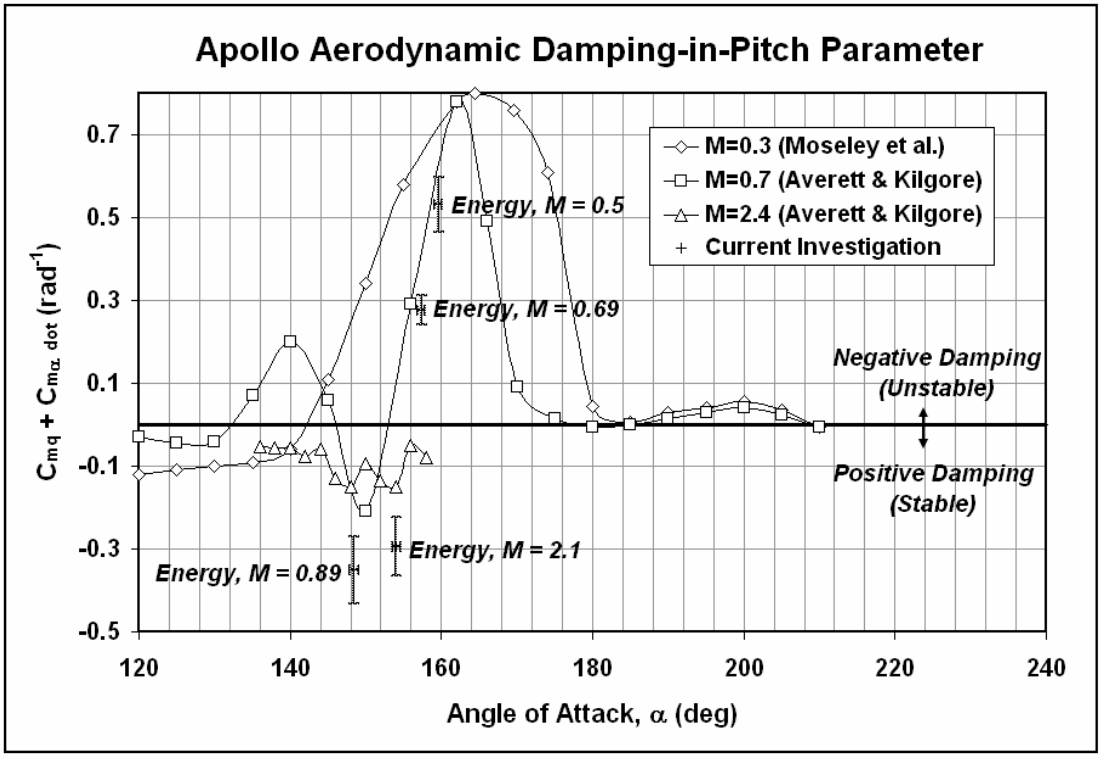

Figure 59 Apollo finalized forced oscillation results using moment hysteresis-energy method with error bands and oscillation frequency adjustment for Strouhal number matching

The finalized results depicted in Figure 59 show a good correlation with published data for all flow regimes investigated. The static moment test results for all flow conditions are also well correlated with published data. Thus, experimental investigation is deemed as a success.

4.3.2 Expert at $\Phi = 0^o$

4.3.2.1 Static Moment Tests

Based on Figure 26 and the data provided in Table 10, the slope of pitch moment with respect to incidence angle is seen to be somewhat similar in the subsonic flight conditions (Mach = 0.5 and 0.7), but like with the Apollo, there is a drastic difference for the transonic flight condition (Mach = 0.88) as well as the supersonic condition (Mach = 2.09). A noticeable trend in the figure is the symmetry of the data rotated about the origin point (at $C_m = 0$, AoA = 0). This is due to the symmetry of the vehicle and the rotation point located at the model centerline. The slopes in the subsonic and transonic regions are positive; this means that the vehicle is statically unstable for the

given vehicle configuration. Yet, there is evidence of a restoring moment that provides static stability in the supersonic regime due to the negative slope of pitch moment with respect to incidence angle. The answers to the reasons for these trends lie in the physiological topology of the flow over the vehicle. The interaction of a shock front in the supersonic flow creates a stronger pressure influence on the vehicle, especially towards the rear of the vehicle where the flaps are located. This causes the center of pressure to shift rearward behind the rotation point for the model and thus enables static stability. As the strong influence of shock interaction dissipates for the transonic regime and is no longer present for the subsonic regimes, the center of pressure is shifted fore of the rotation point yielding static instability. Based on these trends, there is a flow regime, likely in the very low supersonic speeds, where the center of pressure overlaps the rotation point which would yield a zero moment slope. There is no published static data found for the subsonic and transonic test cases to compare with, but there is data based on CFD calculations at Mach = 2.0 from the Expert design report [15]. Based on the data in the report, the pitch moment coefficient for given angle of attack is as follows: $C_m = 0$ for AoA = 0^o, $C_m = -0.039$ for AoA = 2^o, and $C_m = -0.0982$ for AoA = 5^o. The experimentsl test results are $C_m \sim 0$ for AoA = 0^o, $C_m \sim -0.008$ for AoA = 2^o, and $C_m \sim -0.017$ for AoA = 5^o. The resulting static moment test data is nearly one full order of magnitude off from the published data given in the design report. It is not clear as to where this discrepancy is stemming from. The comparison of resulting data from the Apollo test campaign to published data has already validated the static moment test methods. The extraction of the pitch moment coefficient from the static moment tests has been checked various times to ensure that there is no mathematical error. It is also highly unlikely that there was an error in the CFD computations, but whether or not this is certain is unknown. It is suggested that further static tests be conducted to absolutely verify the test data obtained.

4.3.2.2 Forced Oscillation Tests

The method providing the most accurate evaluation for aerodynamic damping was discovered as the moment hysteresis-energy method based on the analysis provided by the Apollo test results compared with published data. Consequently, the finalized results for the Expert are depicted using this method. In accordance with the data given in Table 16 comparing target oscillation frequencies and Mach numbers to what was actually achieved is investigated. Due to mismatches in both, interpolation or extrapolation of the curves shown in Figure 60 are performed to ensure at least a match in oscillation frequency. The only true required extrapolation is for the Mach =

2.12 case, where the required oscillation frequency for Strouhal number preservation is 22.8 Hz but only 18.4 Hz was achieved. It is hypothesized that the trends depicted in Figure 60 are periodic and reflect a symmetric nature, but further testing would have to be done to validate or invalidate this supposition. It is with this hypothesis that the extrapolation for the Mach = 2.12 test case is made. In addition to the extrapolation and interpolation of data, uncertainty associated with smoothing (using data in Figure 57 as a basis), the optical encoder test data, and torque balance moment test data is evaluated and depicted in the same manner as was done for the Apollo. The final resulting values are depicted in Figure 61, followed by a discussion of the results.

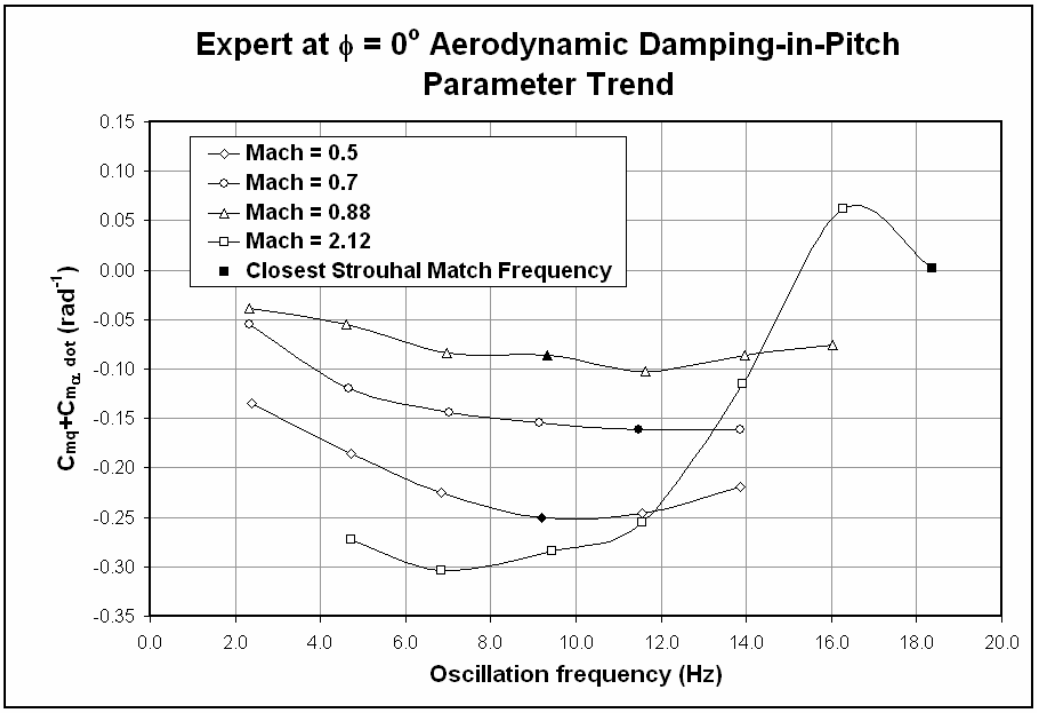

Figure 60 Expert at Φ = 0° trend in oscillation frequency with resulting aerodynamic damping-in-pitch parameter values using the moment hysteresis-energy method

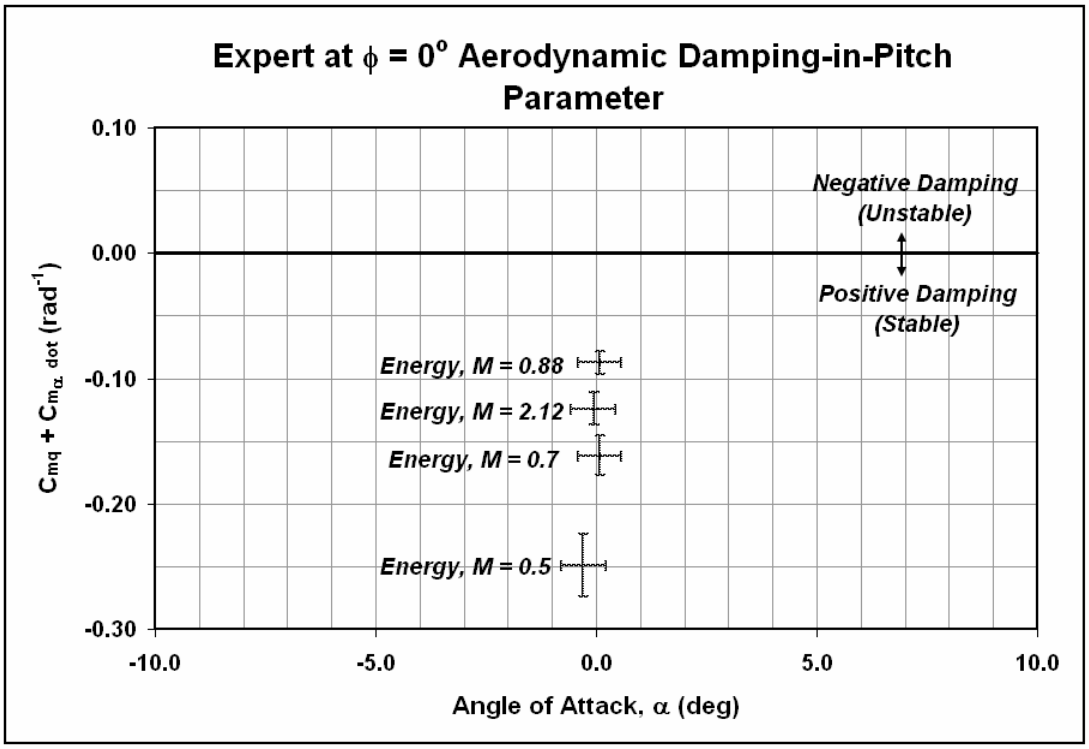

Figure 61 Expert at Φ = 0° finalized forced oscillation results using moment hysteresis-energy method with error bands and oscillation frequency adjustment for Strouhal number matching

The Expert's finalized results, depicted in Figure 61, show that damping is positive for all flow regimes investigated. Whether or not this data is accurate can not be verified as of yet. The fact that the damping is positive, which promotes stability, seems counter-intuitive to the subsonic and transonic static moment test results. The static tests generated positive pitch moment coefficient slopes over incidence range. Such occurrences mean that the vehicle is statically unstable. Although this remains true, a re-examination of the static moment and aerodynamic damping results for the Apollo shows the same counter-intuitive argument for the subsonic tests. The static moment results produced negative slopes (thus static stability), but the aerodynamic damping was found to be negative (thus damping instability). Therefore, the results from the Expert tests are deemed reasonable. The exception is that the situation is reversed from the Apollo subsonic test cases, where the static test results show static instability for the subsonic and transonic flow regimes, yet a damping stabilizing effect based on the forced oscillation testing.

4.3.3 Physical Interpretation of Results

Let us first look at the Apollo static moment and aerodynamic damping results. For the supersonic case (at Mach ~ 2 and a 154.1 degree angle of attack), there is both static stability and positive aerodynamic damping, which is a stabilizing phenomenon. Thus, in this flight regime at the given orientation, the Apollo is completely stable and will begin to orient towards an angle of attack with zero moment if not forced to maintain the given orientation. This is due to the slope of the pitch moment coefficient with respect to incidence angle. For the transonic case (at Mach ~ 0.9 and a 148.5 degree angle of attack), there is also complete stability and the same phenomenon occurs. For the subsonic cases (at Mach ~ 0.5 and 0.7 and a 159.7 and 157.4 degree angle of attack, respectively), there is static stability but a negative aerodynamic damping, which is a destabilizing effect. To physically interpret what happens in this situation, imagine the Apollo coming towards the earth at the given orientation and Mach number. The capsule appears stable, but only for an instant. Because of the negative damping, it begins to oscillate and the oscillations grow in pitch attitude. This will continue until an angle of attack is reached where damping becomes positive. From here, depending on the static moment slope, angle of attack orientation, and the actual value of pitching moment, a variety of phenomena could occur. One, the vehicle may stabilize, but likely only for a moment; two, the vehicle undergoes limit cycle oscillations; or three, the vehicle begins to freely tumble and is incapable to recover unless there is some sort of advanced control system to regain control of the vehicle.

Now, let's examine the Expert. For the supersonic case (at Mach ~ 2.1 and a -0.1 degree angle of attack), there is both static stability and positive aerodynamic damping. Given the flight orientation, the vehicle is completely stable. For the other cases (at Mach ~ 0.88, 0.7, and 0.5 and a 0.1, 0.1, and -0.3 degree angle of attack, respectively), the vehicle is statically unstable, but has a positive aerodynamic damping. The physical interpretation for this situation is overall instability at the given orientation even though there is the presence of positive aerodynamic damping. Due to the static instability, any slight perturbation away from zero moment will cause the vehicle to re-orient until it can find another stable point, and the effect of the positive damping will simply allow this to occur in a smooth manner (meaning without oscillations), but only within the vicinity of the given angle of attack. If a stable point can not be reached, then either limit cycle oscillations may occur or a complete divergence (tumbling motion) of the vehicle's orientation attitude might result. It is greatly desired to avoid such an incidence.

5. CONCLUSIONS

In conclusion, two stereolithographic atmospheric re-entry capsule models, the Apollo and Expert at $\Phi = 0^{o}$, were subjected to static moment and forced oscillation tests in the VKI-S1 wind tunnel. The models were exposed to subsonic, transonic and supersonic flows. The experiment and results analysis processes were split into three major sections: calibrations, experimental testing, and post-processing of results. Calibrations of pressure transducers enabled the accurate measure of wind tunnel flow conditions. Accurate measures for moments experienced by the models and associated uncertainty with those measures were determined with balance calibrations. An optical encoder was used to obtain model pitch attitude. Static moment tests were conducted for determining trim angles of attack and the associated slope of pitching moment with respect to incidence angle over a range of attack angle. These findings were then used to produce the correlation of reduced frequencies between the model and actual flight vehicle corresponding to the flight vehicle natural frequency using the Strouhal number. Forced oscillations at related test conditions to the static moment tests were performed. Oscillation amplitudes were defined by the angle of attack range where there was a constant pitch moment slope. This was found from the static moment tests (about the trim/mean angle of attack) at frequencies for ensuring a match of Strouhal number between the models and flight vehicles. Data from the forced oscillation tests were processed and results on the stability characteristics of each model proceeded.

The Apollo test results depict negative dynamic damping for Mach 0.5 and 0.69 flows at angles of attack of 159.7 and 157.4 degrees, respectively. The corresponding aerodynamic damping-in-pitch parameter values are 0.532 rad^{-1} and 0.276 rad^{-1}, respectively. Positive dynamic damping resulted for Mach 0.89 and 2.1 flows at angles of attack of 148.5 and 154.1 degrees, respectively. The corresponding aerodynamic damping-in-pitch parameter values are -0.351 rad^{-1} and -0.294 rad^{-1}, respectively. The Expert test results depict positive dynamic damping for all Mach numbers tested, which were Mach 0.5, 0.7, 0.88 and 2.12 flows at angles of attack of -0.3, 0.1, 0.1 and -0.1 degrees, respectively. The corresponding aerodynamic damping-in-pitch parameter values are -0.249, -0.161, -0.087 and -0.124 rad^{-1}, respectively.

In addition to this, previously obtained data from a free-to-tumble experiment on an Apollo model at Mach 2.0 flow was analyzed. Major conclusions are drawn from this investigation in the following sections.

5.1 Experimental Method Conclusions and Recommendations

The experimental data chain of going from calibrations, to testing, to post-processing is valid. There are a couple issues that were present in the experiment though that ought to be addressed. The first is the presence of higher frequency signal content than the forced oscillation frequencies. This produced high levels of random noise in the moment acquisition from the torque balance, and one source of these frequencies was attributed to electromagnetic interference from the motor drive inverter (FO testing). These frequencies had to be attenuated by smoothing the signal. Rather, it would be beneficial to shield the drive inverter so as to reduce interference with true signal content. The other is the uncertainty associated with resulting moment values (at least for the Apollo). The moments during forced oscillation testing were quite low in comparison with the maximum allowable of the torque balance used. Using a higher sensitivity torque balance would provide lower uncertainties in acquired moment values, especially for higher Mach number tests.

5.2 Post-Processing Conclusions and Recommendations

Three post-processing methods were examined to analyze forced oscillation data obtained and three others for analyzing free-to-tumble data from a previous experiment. The end-all goal of the processing was to determine the aerodynamic damping-in-pitch parameter.

5.2.1 Free-to-Tumble

5.2.1.1 Algebraic Method

The algebraic method, outlined in Section 2.2.1.1, proved to yield absolute inconclusive results in relation to evaluating the aerodynamic damping-in-pitch parameter. This was due to truncation errors associated with derivative evaluations from experimental data. Until there is a numeric method out there that can guarantee precise derivative calculations (which has proven to be highly unlikely thus far), it is the opinion of this author that this method is a waste of time and should not be used in future investigations.

5.2.1.2 Linearized Algebra Method

This method, outlined in Section 2.2.1.3, yields the same inconclusive results as the algebraic method. This method promised very good results when it was originally developed, but the reason was because exact derivative values

were being used rather than numerically based derivatives. The only way to extract derivative values based on experimental data is by using numerical methods, which as stated in the algebraic method conclusions, leads to truncation error. Thus, matrix inversion processes associated with this method became ill-posed and led to unsatisfying results. Future investigation using a least-squares process to reduce truncation error may be useful. It is the opinion of this author not to use this method for future investigations.

5.2.1.3 Envelope Method

This method, outlined in Section 2.2.1.2, is one that has been used by many others before. The results yielded the trend of increased aerodynamic damping as the model reached a stable, steady state angle of attack. This correlates well with common reason. Though not a precise method, it proved to be the best measure for aerodynamic damping-in-pitch evaluation for the free-to-tumble tests. An interesting note to the results of the simulated case study, given in Section 4.2.1.1, is that the method's resulting distribution curve (based on the outer points of the constant damping lines) of aerodynamic damping-in-pitch relative to model angle of attack could possibly be elongated or squished (while preserving area under the distribution curve for energy conservation) in a manner such that the results would fall on the true values. Whether or not this is true is unknown and would be good to examine for future investigations.

5.2.2 Forced Oscillation

5.2.2.1 Algebraic Method

The results from this method, outlined in Section 2.3.1.1, did not provide a measure for determining the aerodynamic damping-in-pitch parameter, but merely as a validation to the presence of moment hysteresis in a forced oscillation system. No other conclusions are drawn from this method except that its use in future investigations is recommended against.

5.2.2.2 Phase-Difference Amplitude Method

The results of this method, outlined in Section 2.3.1.2, provided inconclusive results (for both sub-methods, based on Apollo data) for the aerodynamic damping-in-pitch parameter. The reason for this is due to the magnitude of measured moment during tests. The higher Mach number(s) yielded extremely low magnitudes of oscillation moments. Based on the uncertainty curve of moment for a certain measured value, this resulted in high levels of

uncertainty. Therefore, this method is deemed as one that should not be used for future investigations, and it is suggested that higher sensitivity equipment (torque balance) be used so as to reduce uncertainty.

5.2.2.3 Moment Hysteresis-Energy Method

This method, outlined in Section 2.3.1.3, yielded the best results in comparing the Apollo model results with published data (see Figure 59). A conclusion that even for stable regions of angle of attack, a divergent oscillatory motion (due to negative damping) could be created due to surface pressure variation time-lags was drawn by Karatekin, [1] in his low speed investigations on an Apollo model. The results of this method verify his conclusion for the subsonic Mach values tested on the Apollo. Also, the prediction of positive aerodynamic damping (thus negative value of aerodynamic damping-in-pitch parameter) for the transonic and supersonic test case in a certain region of angle of attack is verified with published data. Thus, it is deemed that the highest priority in further expanding and developing experimental forced oscillation techniques to correlate well with accurate results should be centered on this method.

5.2.3 Comparing Free-to-Tumble and Forced Oscillation Methods

The results show that the moment hysteresis-energy method using the forced oscillation test technique yields the most verifiable results in comparing with published data. Yet, this method only provides a single data point at a time for each complete test carried out. It would take a great deal of time in testing and post-processing as well as money in order to obtain an entire aerodynamic damping profile for a vehicle. The free-to-tumble test technique would allow one to do this using just a few test campaigns. The drawback is that there is no definitive post-processing method, as of yet, that provides results with near absolute confidence for this test technique. It is suggested that for future work and investigation on aerodynamic damping-in-pitch extraction that methods adapted with the envelope method for free-to-tumble test results be a focus for research. It is believed that in the long-run, this will end up being the most effective approach to solving this problem and will ultimately save time and money.

6. REFERENCES

1. Karatekin, O.: Aerodynamics of a planetary entry capsule at low speeds. von Karman Institute, 2002.

2. Etkin, B.: Dynamics of atmospheric flight. John Wiley and Sons Inc., 1972.

3. Loh, W.: Dynamics and thermodynamics of planetary entry. Prentice Hall Inc., 1963.

4. Tang, D.M. and Dowell, E.H.: Experimental investigation of three-dimensional dynamic stall model oscillation in pitch. Journal of Aircraft, Vol. 32, No. 5,1995, pgs. 1062-1070.

5. Lan, E.C. and Chen, Y.: Experimental and analytical investigation of transonic limit-cycle oscillations of a flaperon. Journal of Aircraft, Vol. 32, No. 5,1995, pgs. 905-910.

6. Guglieri, G. and Quagliotti, F.: Low speed dynamic tests on a capsule configuration. Aerospace Sciences and Technology, pgs. 390-393, 2000.

7. Paris, S. and Charbonnier, J.M.: Aerodynamic force and moment measurements and dynamic characterization of an Apollo capsule in the VKI-S1 wind tunnel. von Karman Institute, TN-107, 1997.

8. Chapman, D.R.: An approximate analytical method for studying entry into planetary atmospheres. National Advisory Committee for Aeronautics, TN 4276, 1958.

9. Redd, B., Olsen, D. and Barton, R.: Relationship between the aerodynamic damping derivatives measured as a function of instantaneous angular displacement and the aerodynamic damping derivatives measured as a function of oscillation amplitude. National Aeronautics and Space Administration, TN D-2855, 1965.

10. Hanff, E.S.: Direct forced-oscillation techniques for the determination of stability derivatives in wind tunnels.
Unsteady Aerodynamics Laboratory, National Research Council of Canada.

11. Billingsley, J.P. and Norman, W.S.: Relationship between local and effective aerodynamic pitch-damping derivatives as measured by a forced-oscillation balance for preliminary Viking configurations.
von Karman Gas Dynamics Facility, ARO, Inc., 1972.

12. Fletcher, D.G. and Asma, C.O.: Tools for the study of dynamic stability of reentry vehicles in the transonic/supersonic regime.
von Karman Institute, Proposal, August 2006.

13. Torgler, B., Beloki, J. and Neves, S.: Calculation of aerodynamic forces and moments on an ogive body.
von Karman Institute, AMTA Lab Report, 2006.

14. Post-launch report for mission AS-202.
National Aeronautics and Space Administration, Post-Launch Report, 1966.

15. Expert design report: vehicle design definition and configuration file.
Alcatel Alenia Space Co., Design Report, Dec. 2005.

16. Underhill, K.: Experimental set-up for stability experiments on planetary entry vehicles.
von Karman Institute, Stagiaire Report 2005-40, Sept. 2005.

17. Emilie, R.: Dynamic characterization of an Apollo capsule and flow calibration in the VKI-S1 wind tunnel.
von Karman Institute, Stagiaire Report 2006-43, Sept. 2006.

18. Moseley, W.C., Moore, R.H. and Hughes, J.E.: Stability characteristics of the Apollo command module.
NASA, Manned Spacecraft Center, TN D-3890, 1966.

19. Averett, B.T. and Kilgore, R.A.: Dynamic stability characteristics of models of proposed Apollo configurations at Mach number from 0.3 to 1.2.
NASA, Langley Research Center, TM X-912, 1964.

20. Moseley, W.C., Graham, R.E. and Hughes, J.E.: Aerodynamic stability characteristics of the Apollo command module.
NASA, Manned Spacecraft Center, TN D-4688, 1968.

21. Paris, S.: Investigation of dynamic stability in transonic/supersonic regime.
von Karman Institute, 106th STAI Conference Meeting, 2006.

22. Technical Group. von Karman Institute.

A. APPENDICES

A.1 Figures

Forced oscillation data

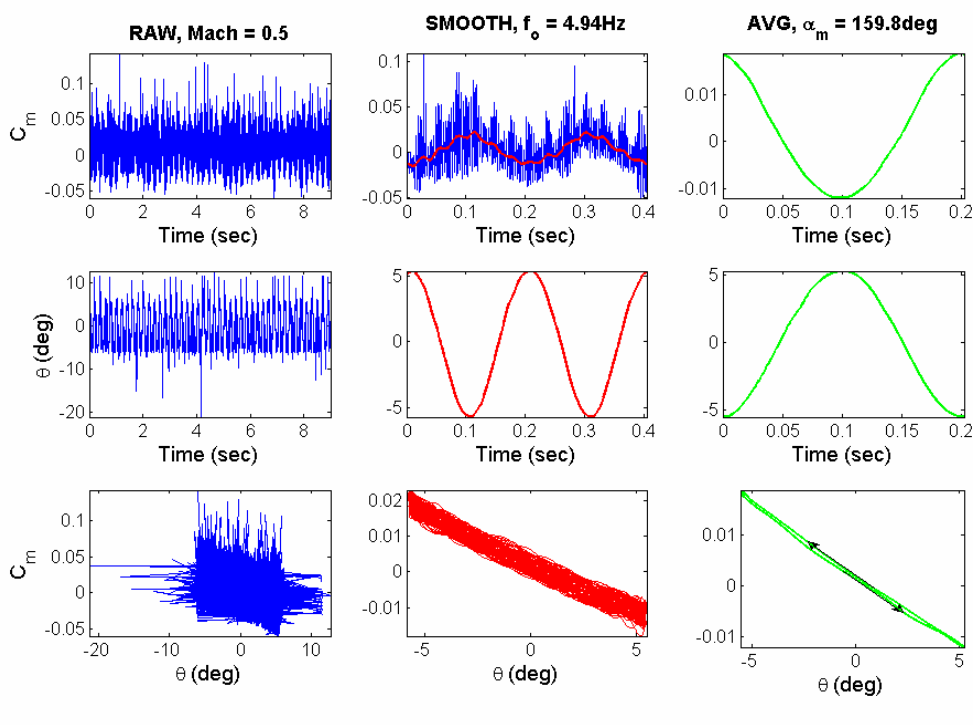

Figure 62 Apollo forced oscillations (Mach = 0.5, f_o = 4.9 Hz)

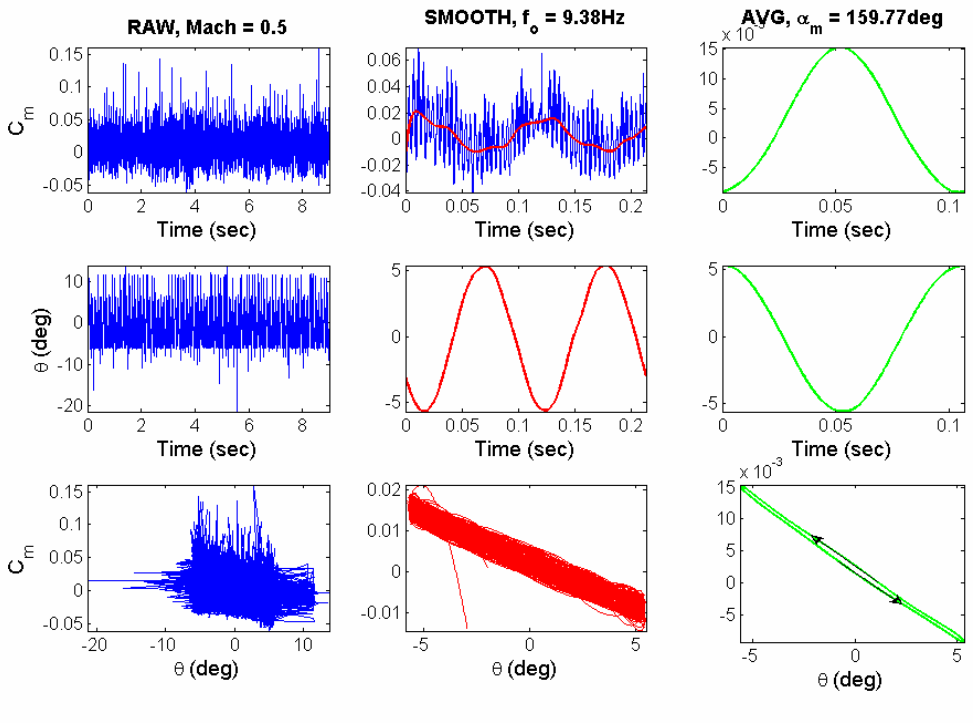

Figure 63 Apollo forced oscillations (Mach = 0.5, f_o = 9.4 Hz)

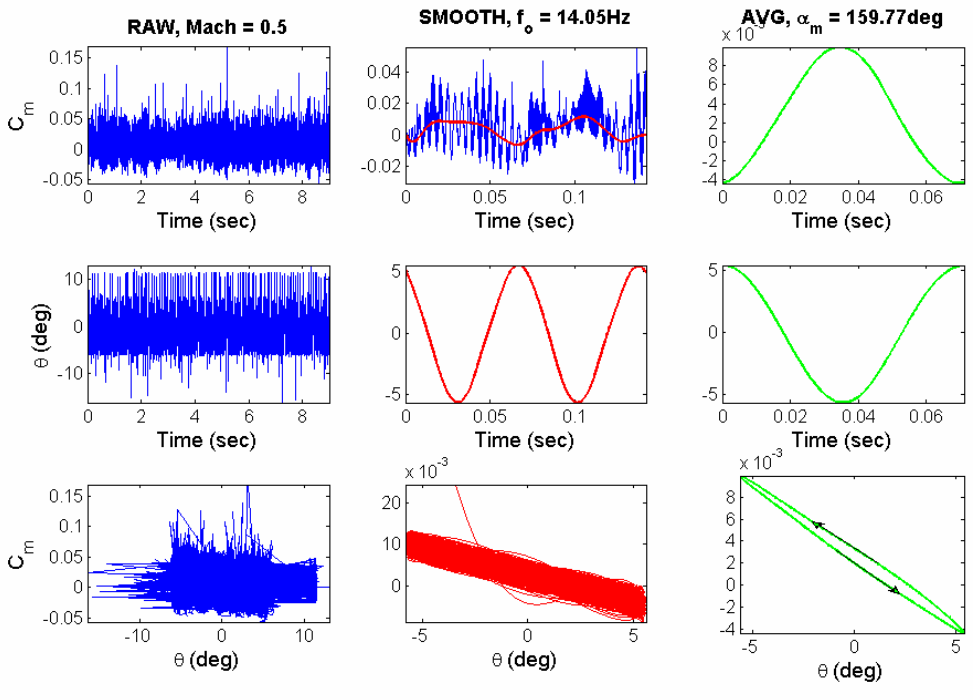

Figure 64 Apollo forced oscillations (Mach = 0.5, f_o = 14.1 Hz)

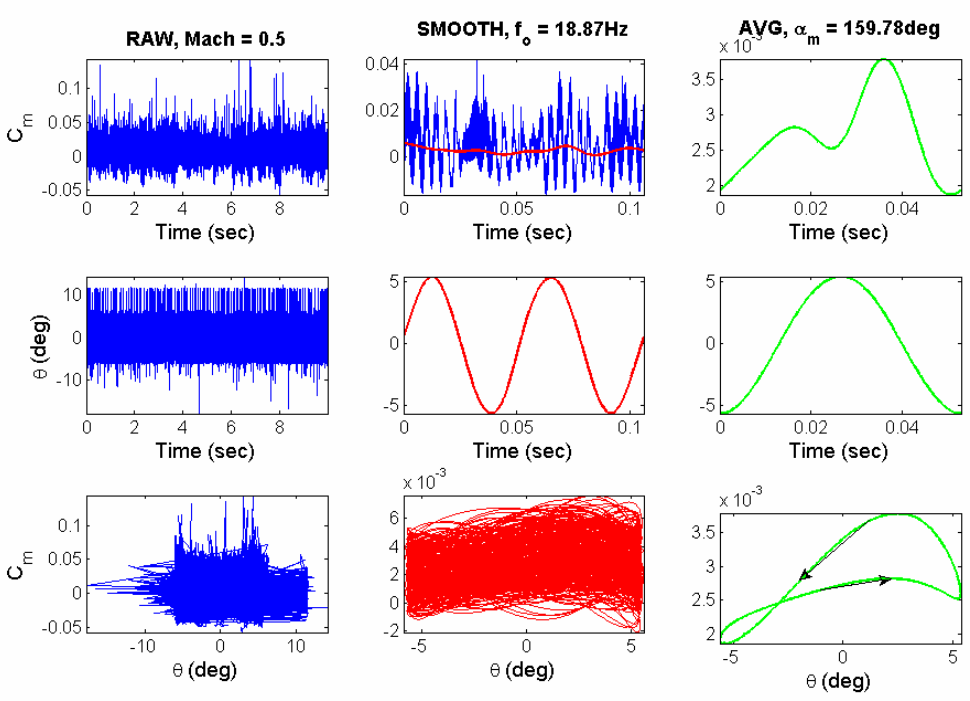

Figure 65 Apollo forced oscillations (Mach = 0.5, f_o = 18.9 Hz)

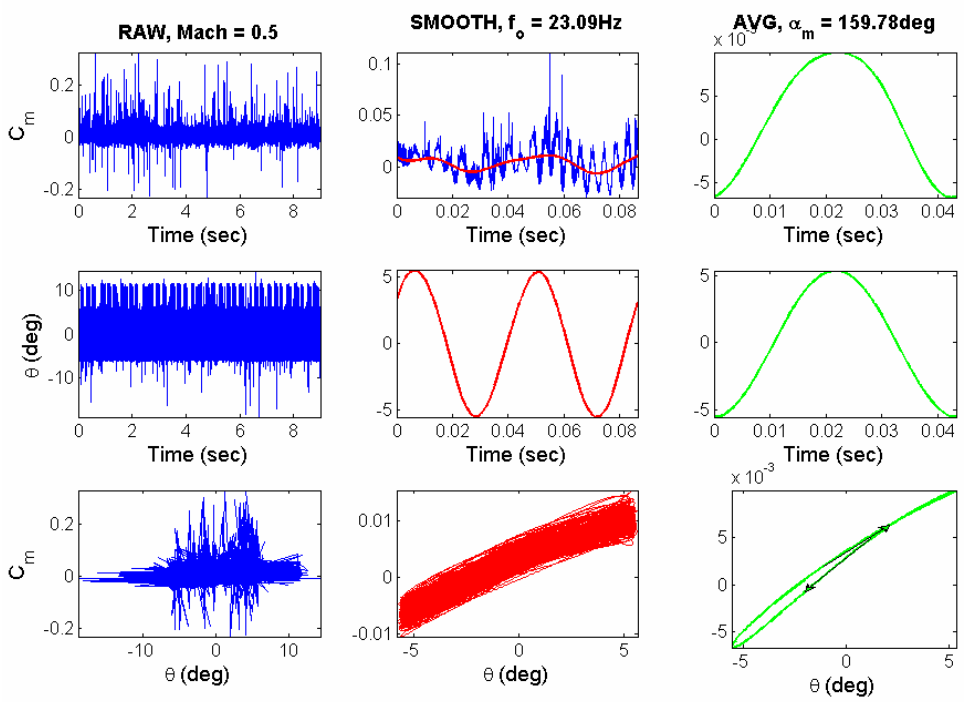

Figure 66 Apollo forced oscillations (Mach = 0.5, f_o = 23.1 Hz)

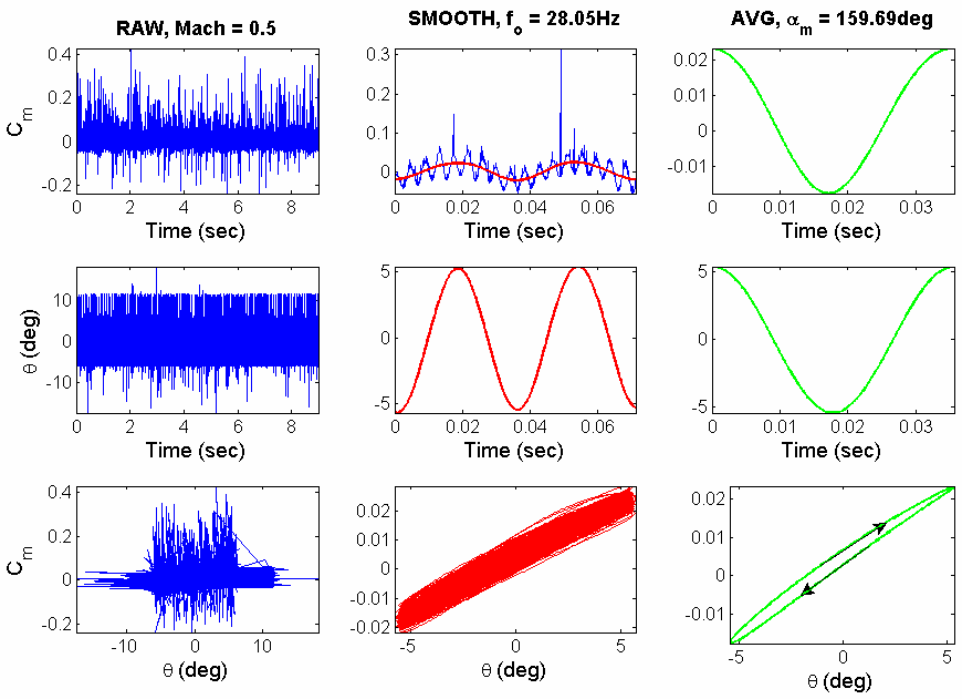

Figure 67 Apollo forced oscillations (Mach = 0.5, f_o = 28.1 Hz)

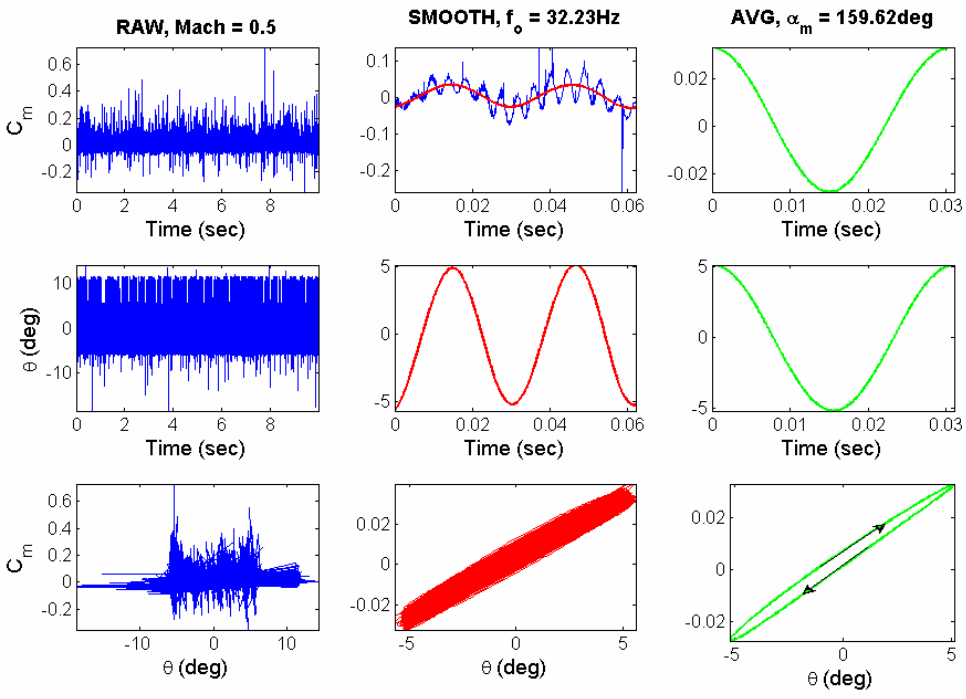

Figure 68 Apollo forced oscillations (Mach = 0.5, f_o = 32.2 Hz)

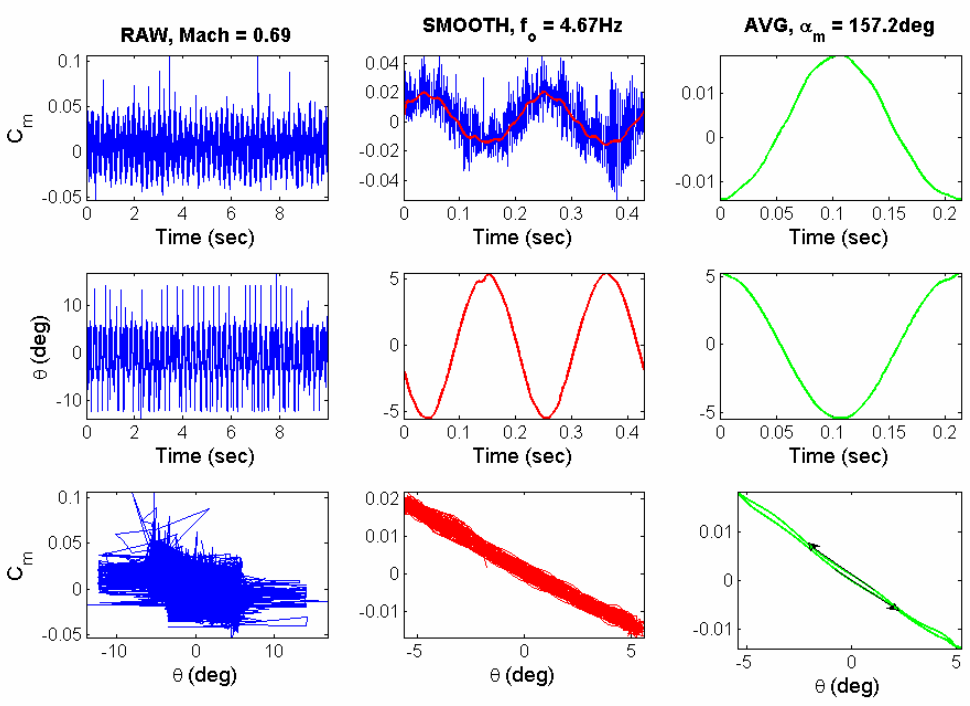

Figure 69 Apollo forced oscillations (Mach = 0.69, f_o = 4.7 Hz)

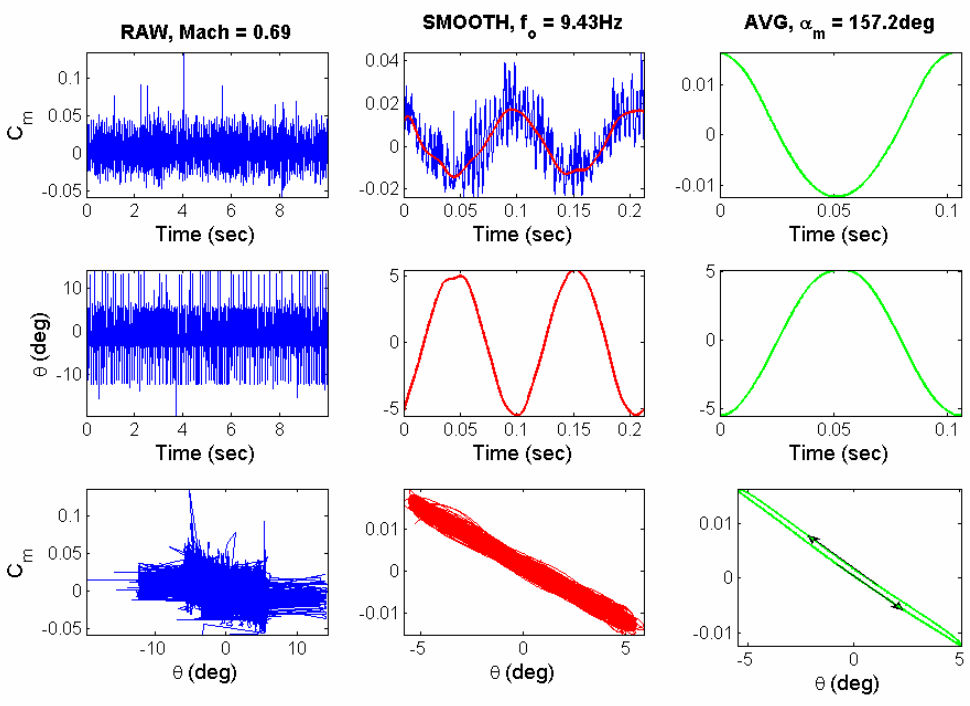

Figure 70 Apollo forced oscillations (Mach = 0.69, f_o = 9.4 Hz)

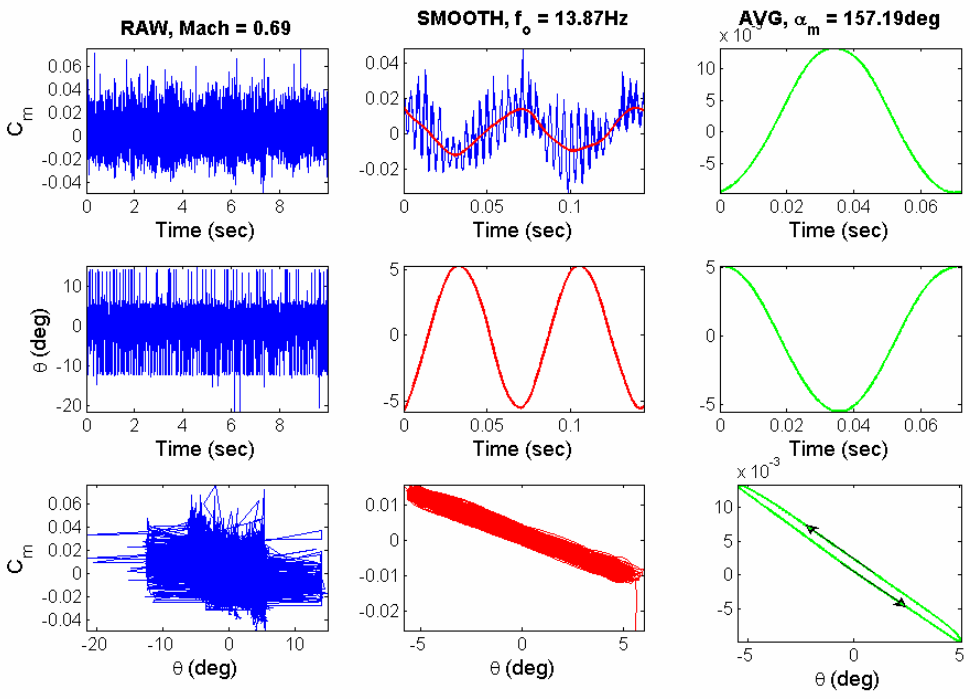

Figure 71 Apollo forced oscillations (Mach = 0.69, f_o = 13.9 Hz)

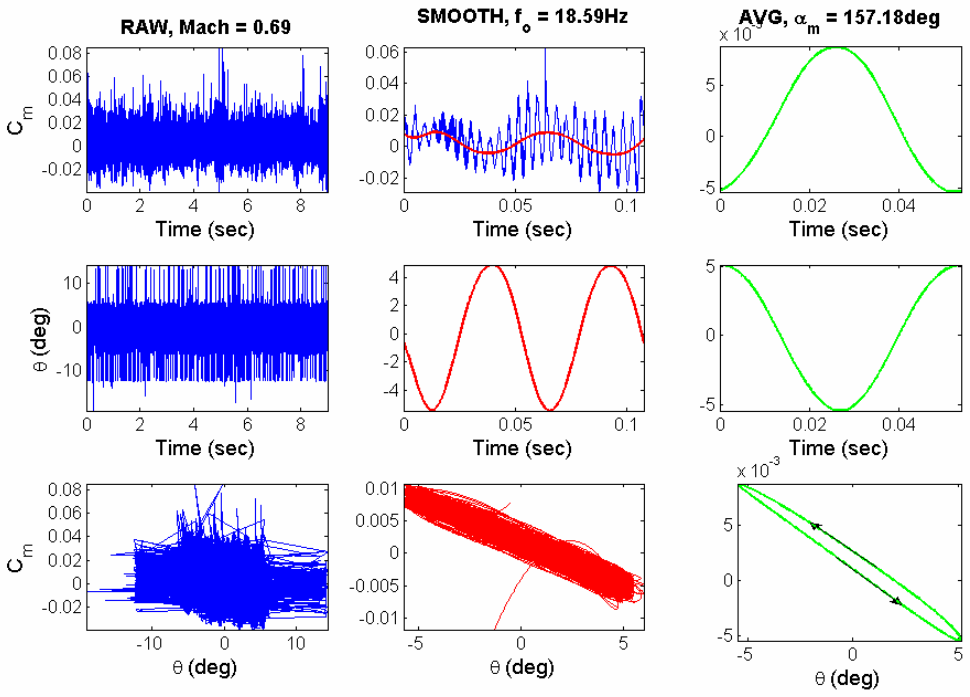

Figure 72 Apollo forced oscillations (Mach = 0.69, f_o = 18.6 Hz)

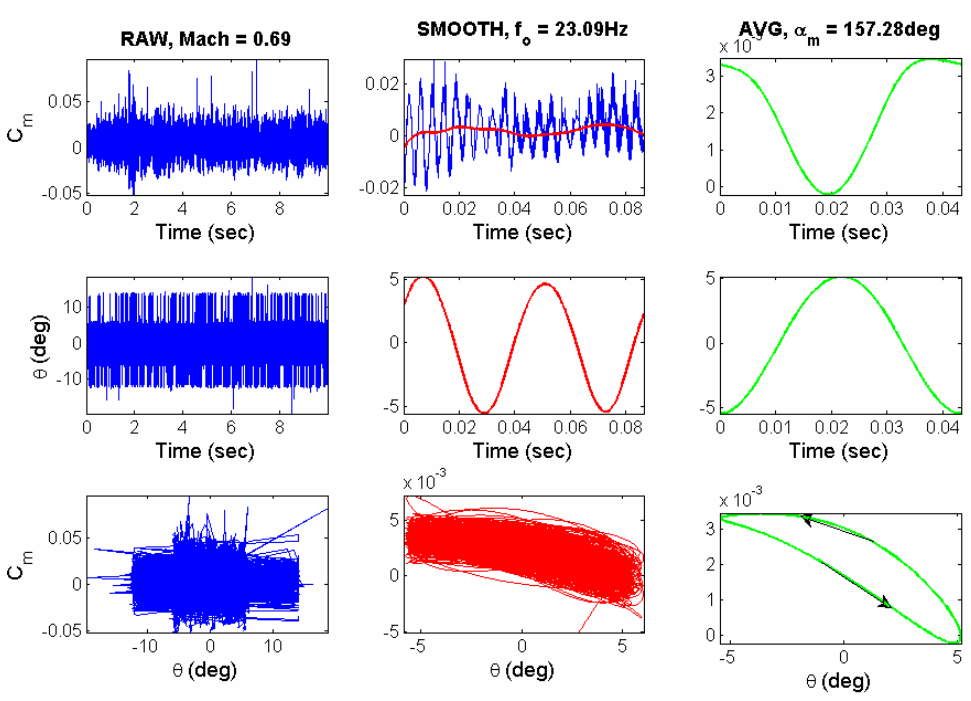

Figure 73 Apollo forced oscillations (Mach = 0.69, f_o = 23.1 Hz)

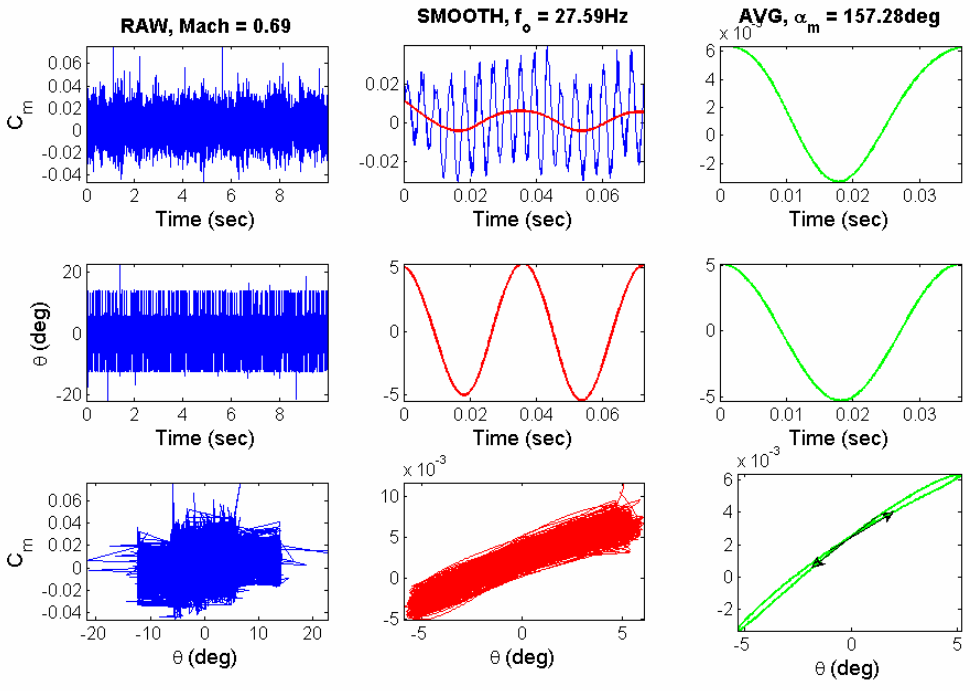

Figure 74 Apollo forced oscillations (Mach = 0.69, f_o = 27.6 Hz)

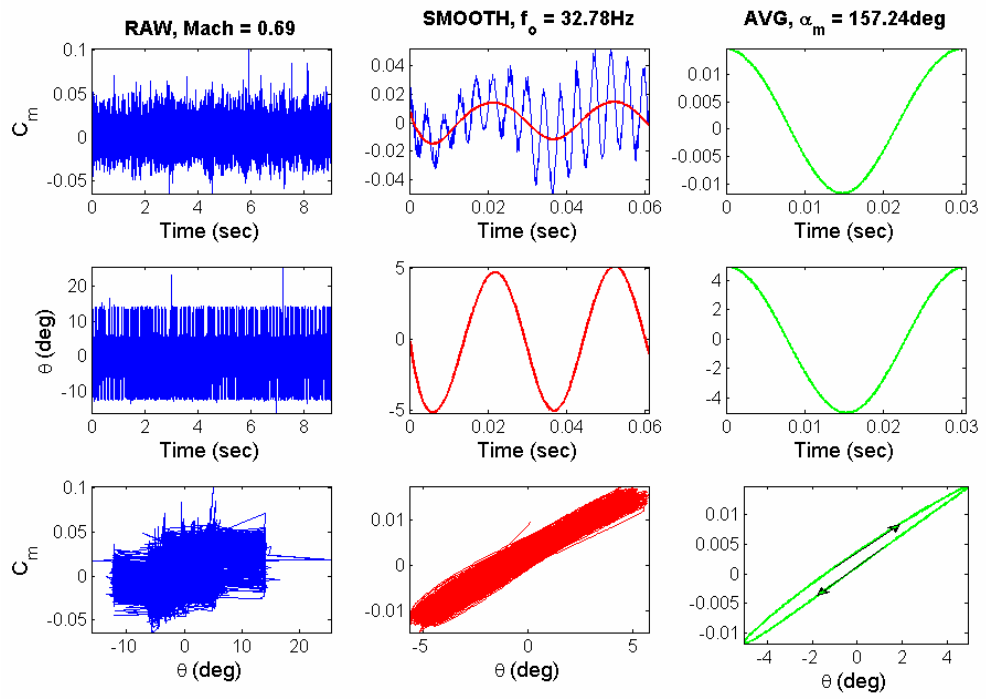

Figure 75 Apollo forced oscillations (Mach = 0.69, f_o = 32.8 Hz)

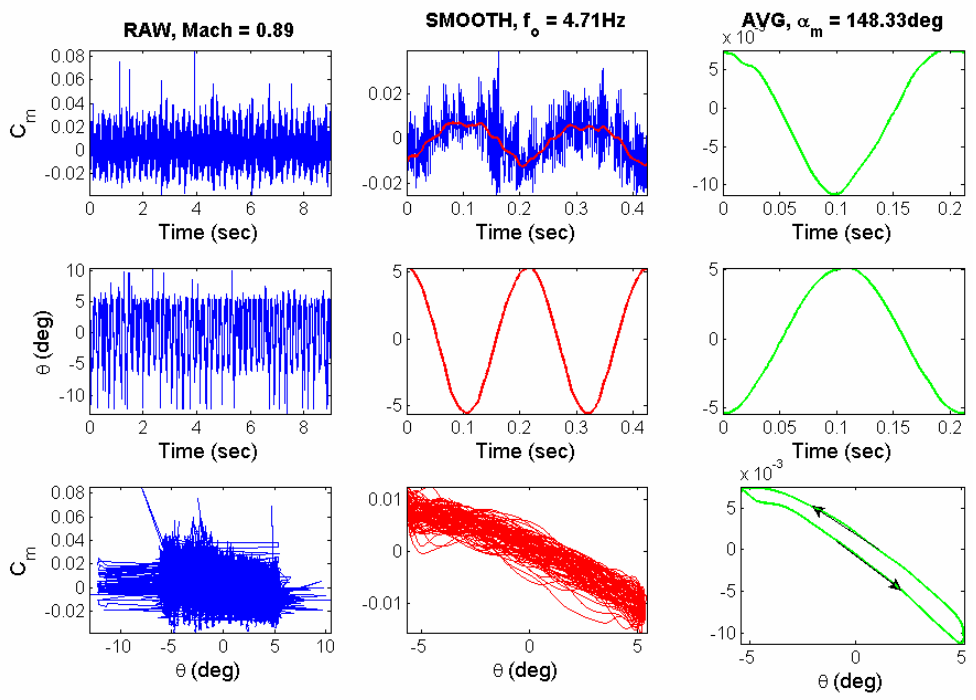

Figure 76 Apollo forced oscillations (Mach = 0.89, f_o = 4.7 Hz)

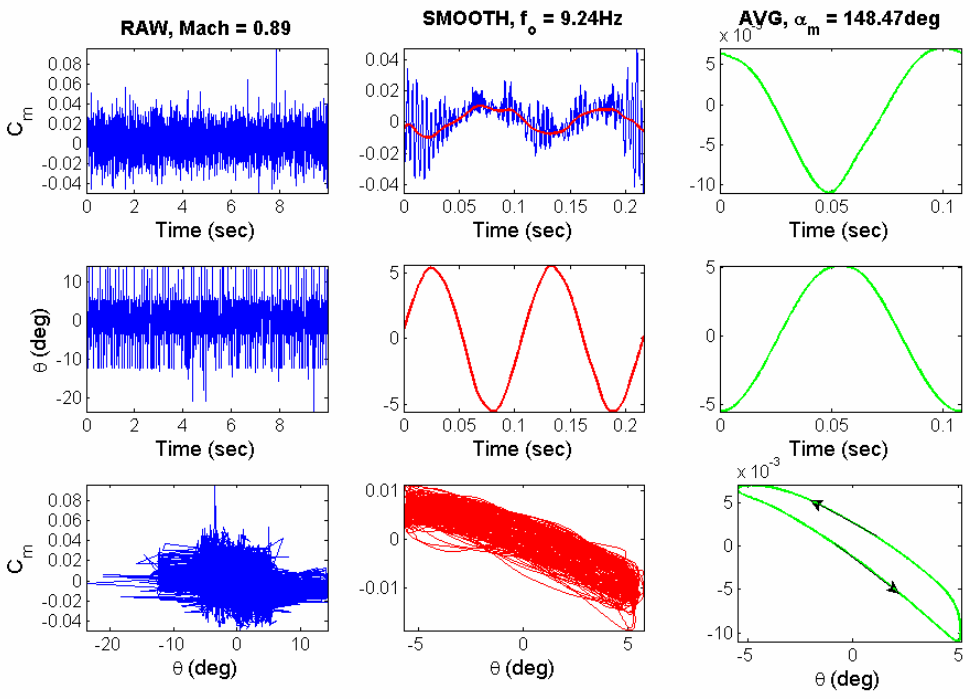

Figure 77 Apollo forced oscillations (Mach = 0.89, f_o = 9.2 Hz)

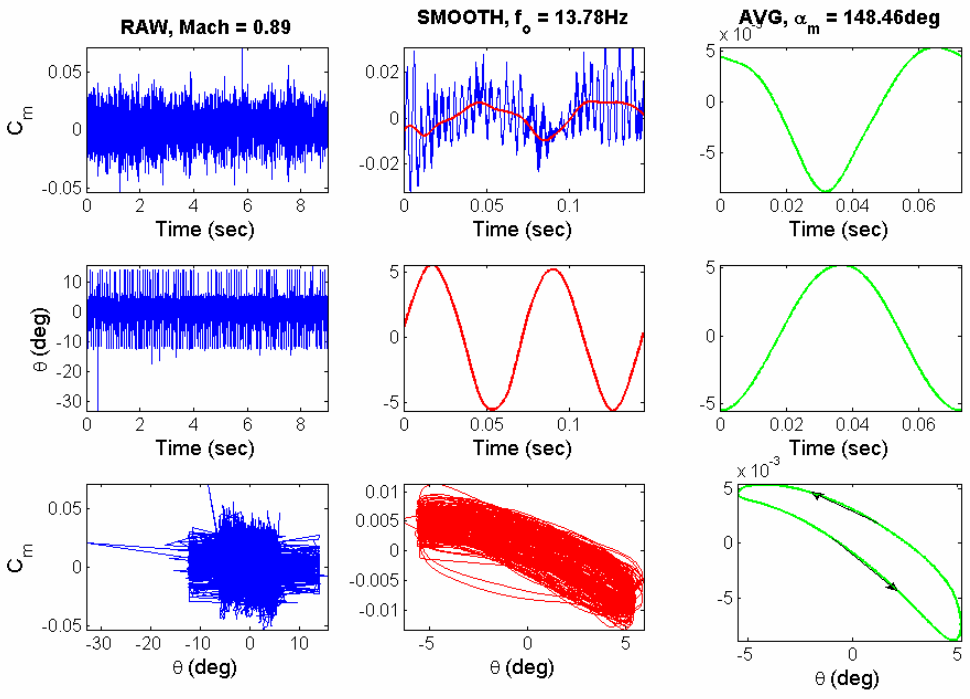

Figure 78 Apollo forced oscillations (Mach = 0.89, f_o = 13.8 Hz)

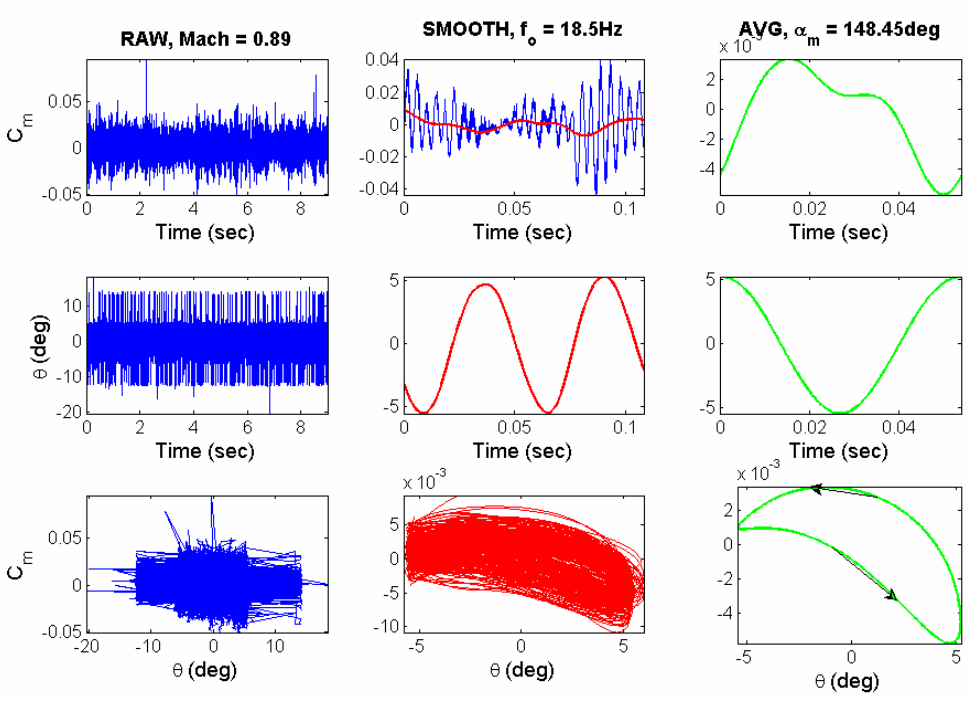

Figure 79 Apollo forced oscillations (Mach = 0.89, f_o = 18.5 Hz)

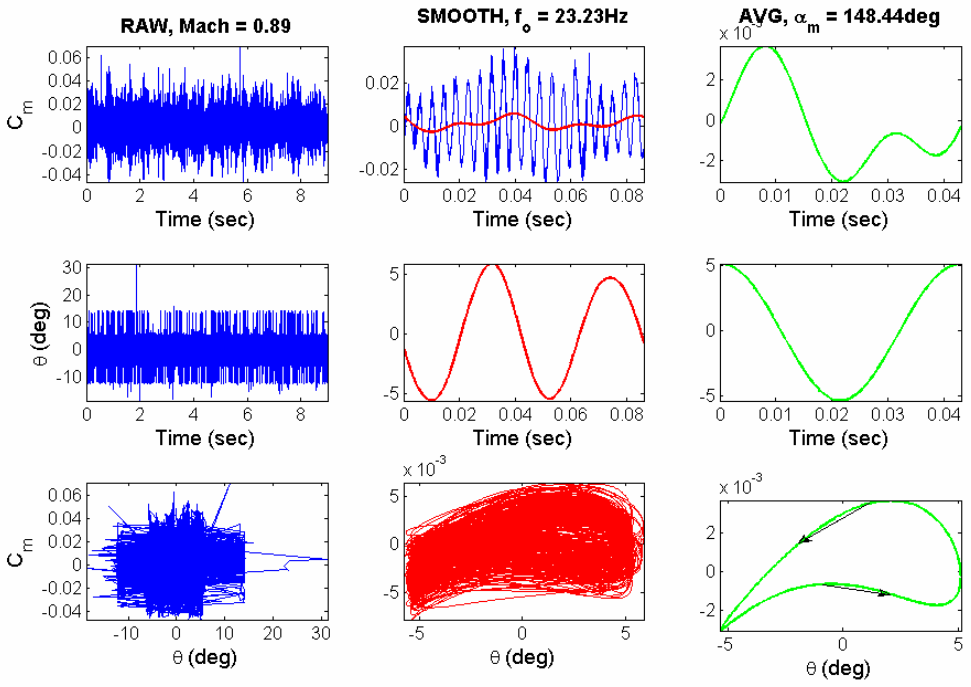

Figure 80 Apollo forced oscillations (Mach = 0.89, f_o = 23.2 Hz)

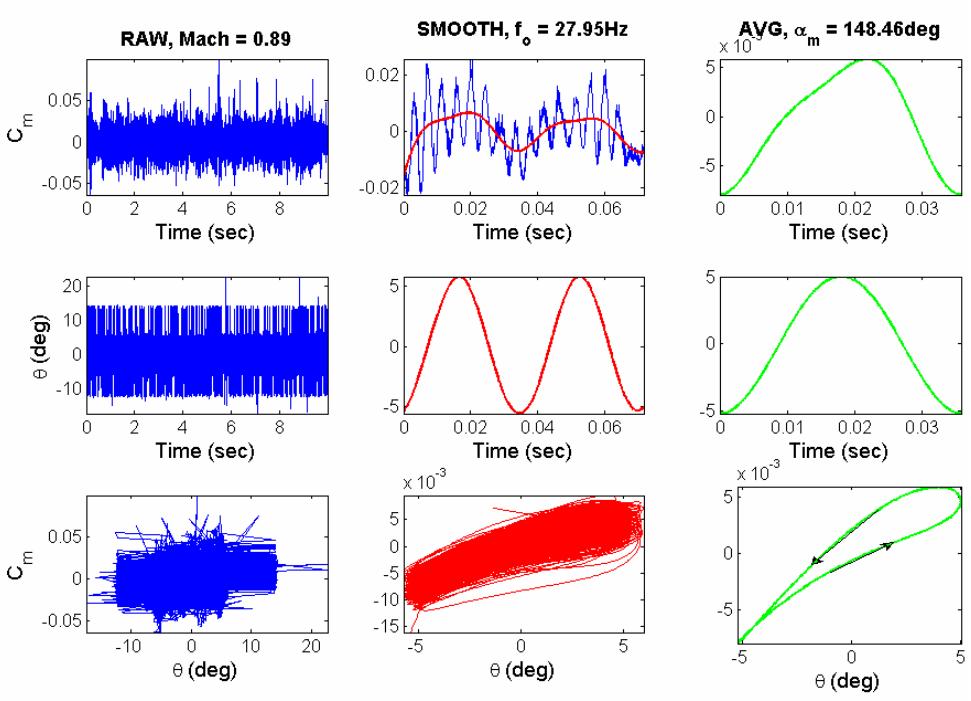

Figure 81 Apollo forced oscillations (Mach = 0.89, f_o = 28 Hz)

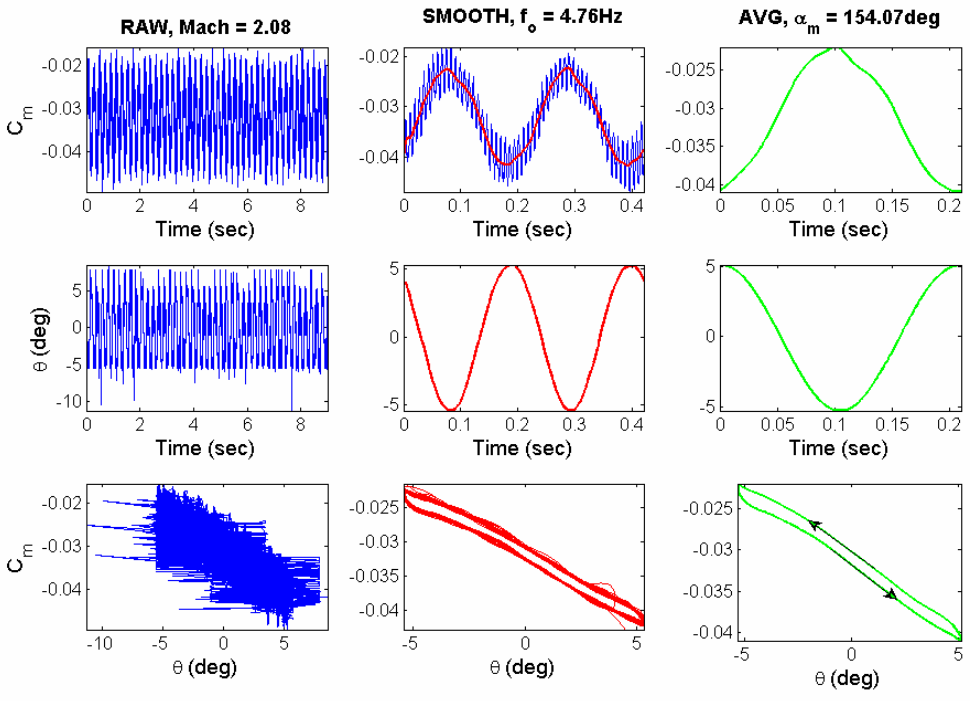

Figure 82 Apollo forced oscillations (Mach = 2.08, f_o = 4.8 Hz)

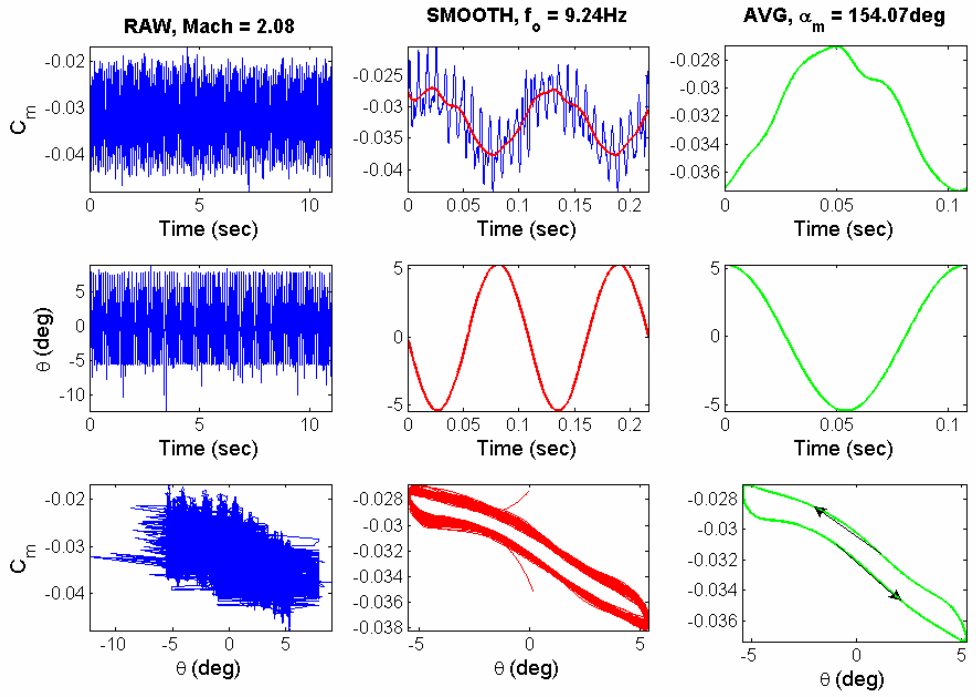

Figure 83 Apollo forced oscillations (Mach = 2.08, f_o = 9.2 Hz)

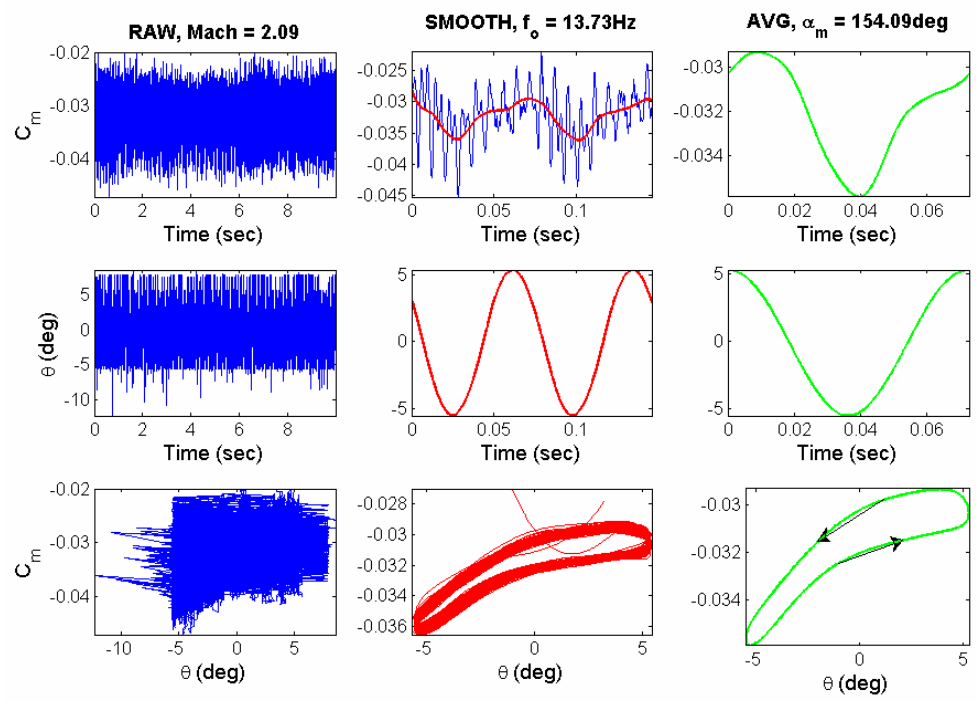

Figure 84 Apollo forced oscillations (Mach = 2.09, f_o = 13.7 Hz)

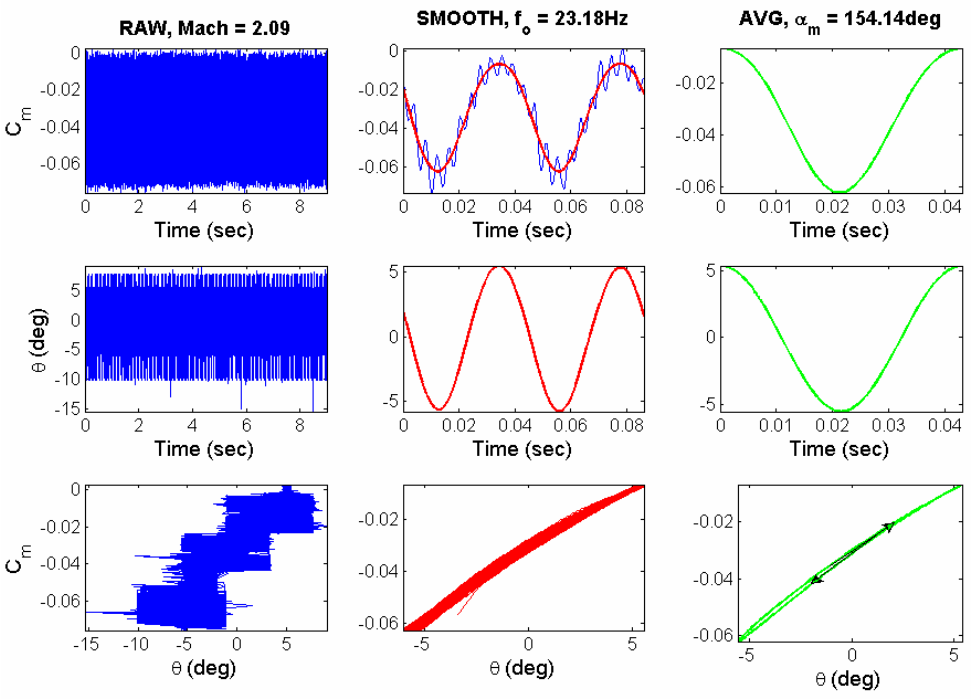

Figure 85 Apollo forced oscillations (Mach = 2.09, f_o = 23.2 Hz)

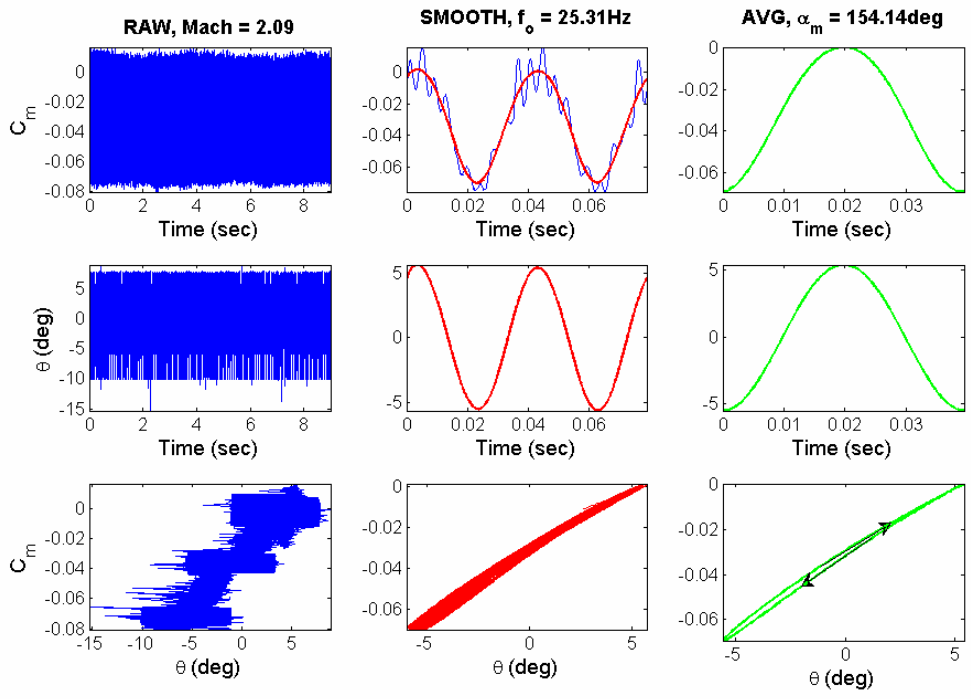

Figure 86 Apollo forced oscillations (Mach = 2.09, f_o = 25.3 Hz)

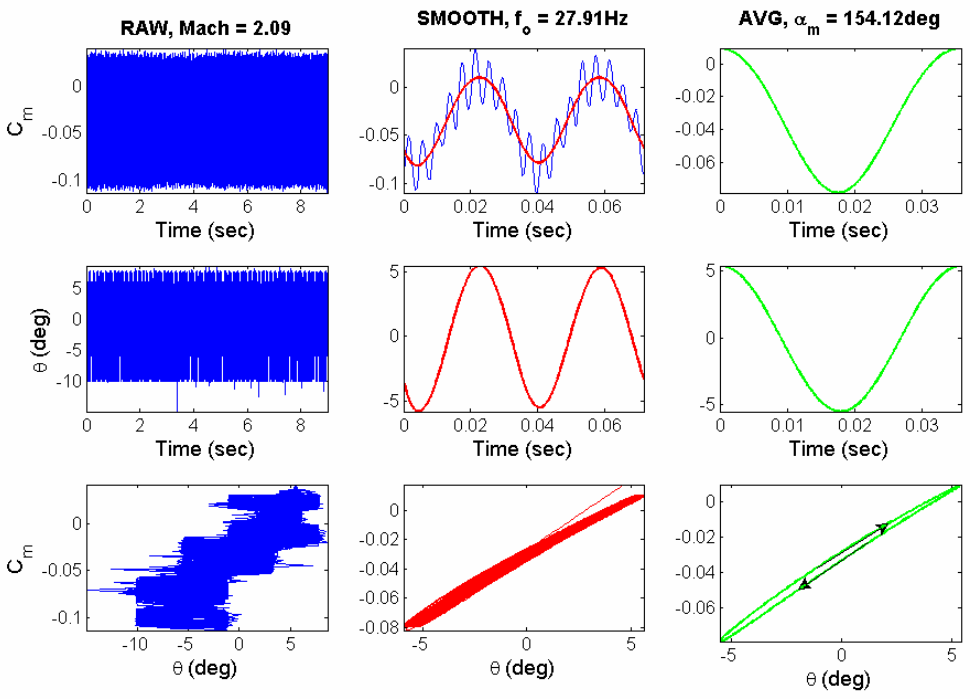

Figure 87 Apollo forced oscillations (Mach = 2.09, f_o = 27.9 Hz)

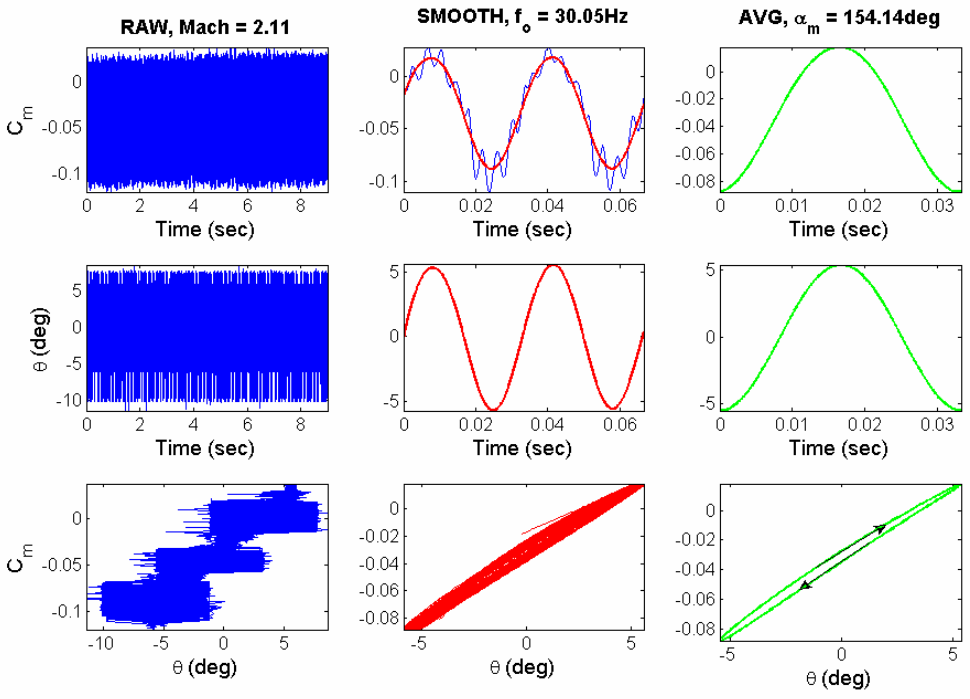

Figure 88 Apollo forced oscillations (Mach = 2.11, f_o = 30.1 Hz)

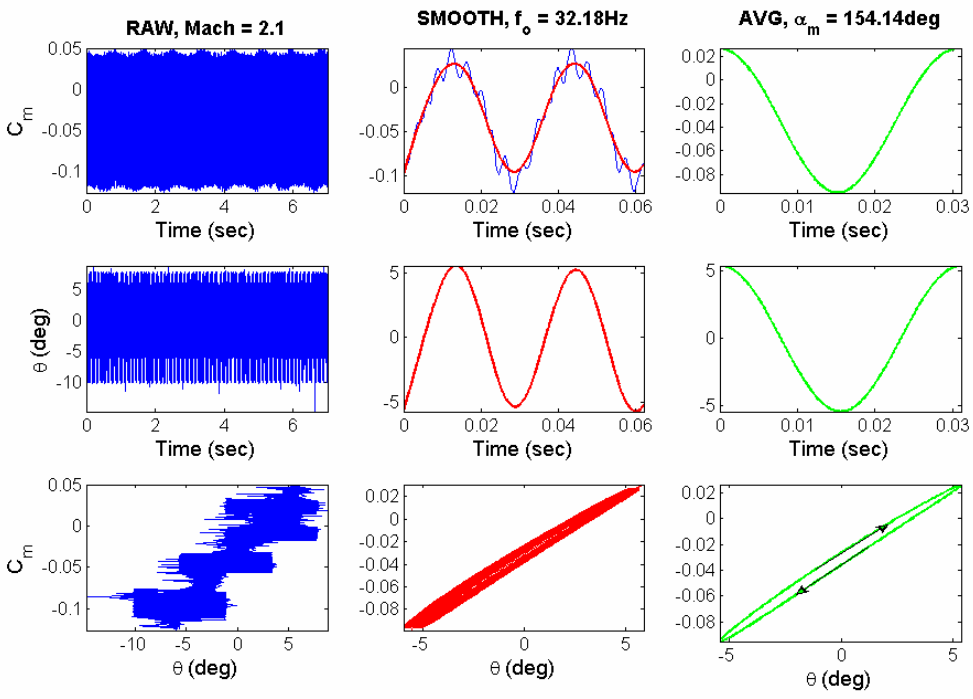

Figure 89 Apollo forced oscillations (Mach = 2.1, f_o = 32.2 Hz)

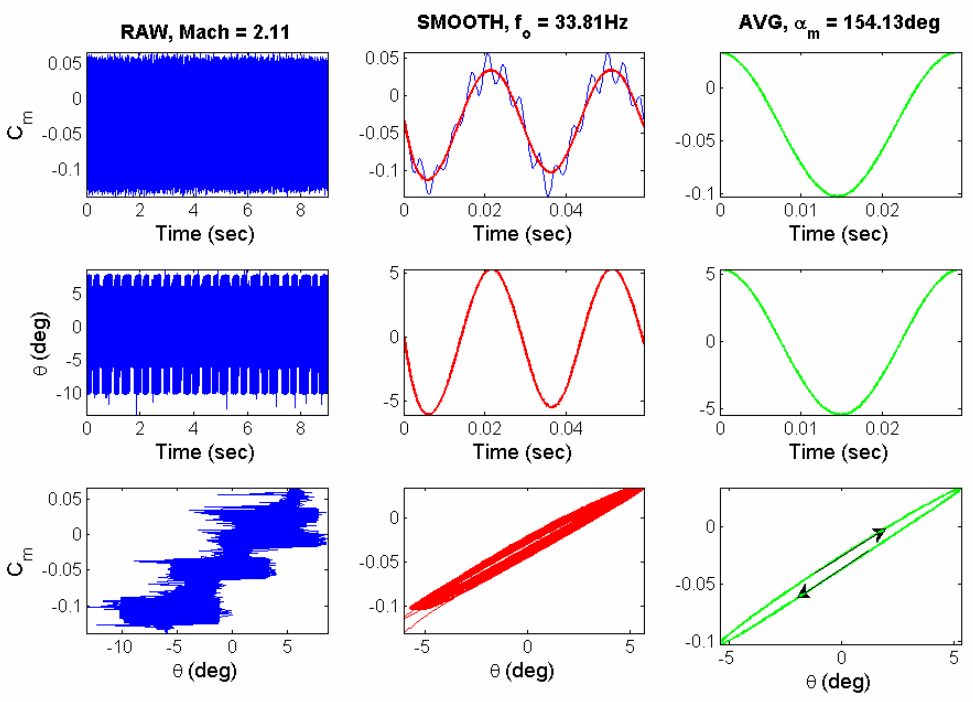

Figure 90 Apollo forced oscillations (Mach = 2.11, f_o = 33.8 Hz)

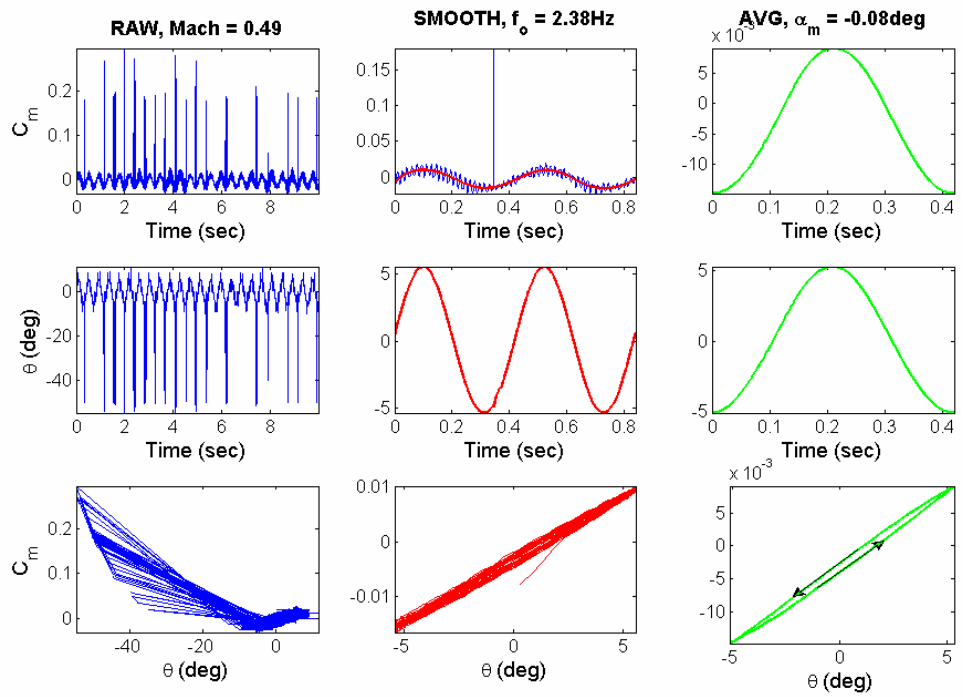

Figure 91 Expert at Φ = 0° forced oscillations (Mach = 0.49, f_o = 2.4 Hz)

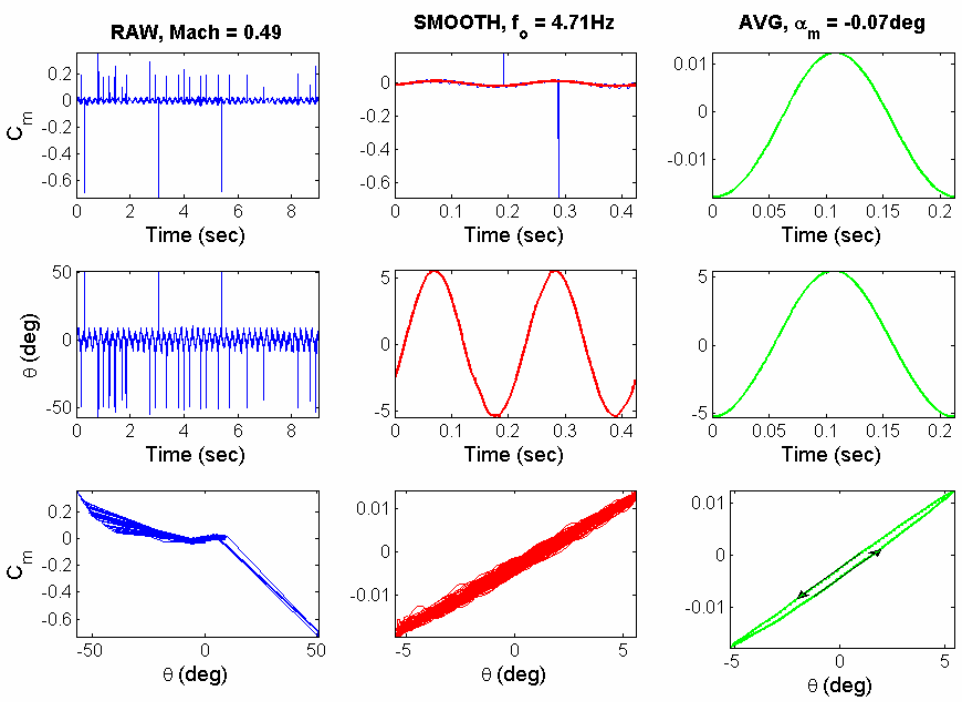

Figure 92 Expert at Φ = 0° forced oscillations (Mach = 0.49, f_o = 4.7 Hz)

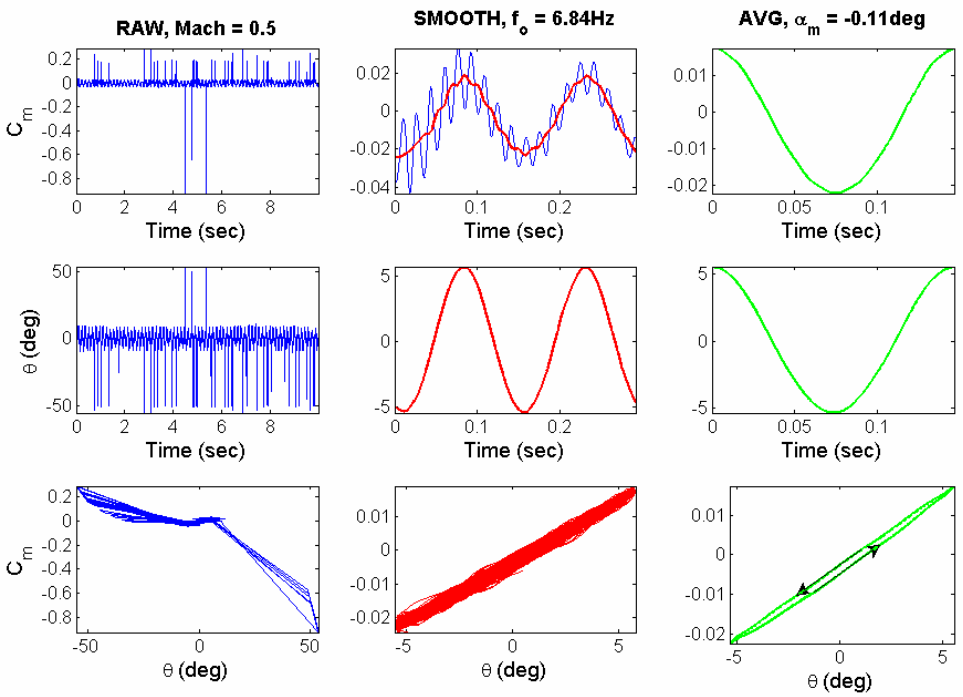

Figure 93 Expert at Φ = 0° forced oscillations (Mach = 0.5, f_o = 6.8 Hz)

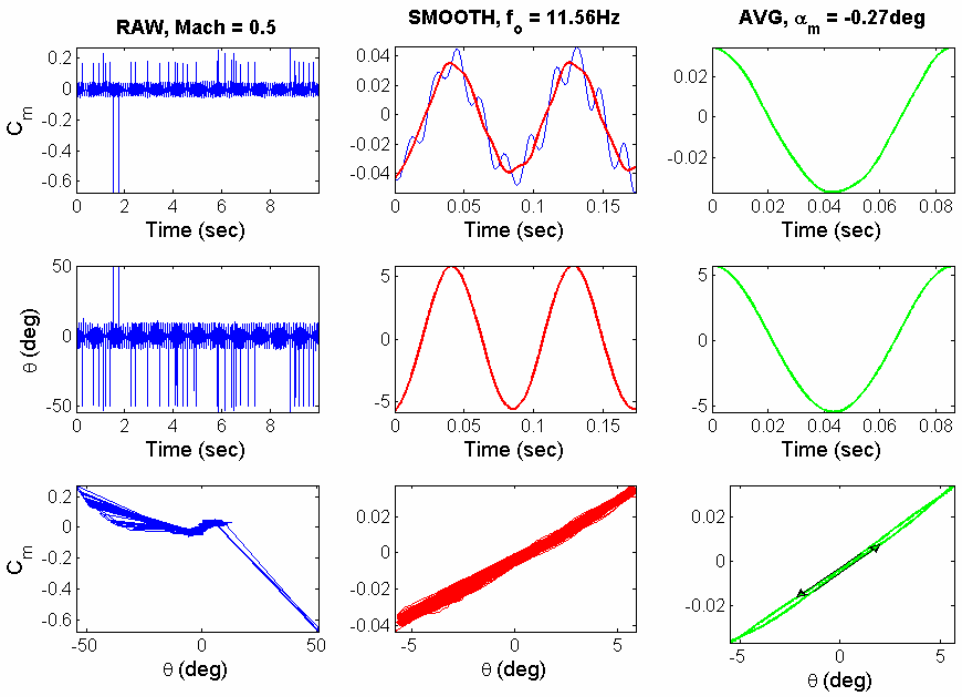

Figure 94 Expert at Φ = 0° forced oscillations (Mach = 0.5, f_o = 11.6 Hz)

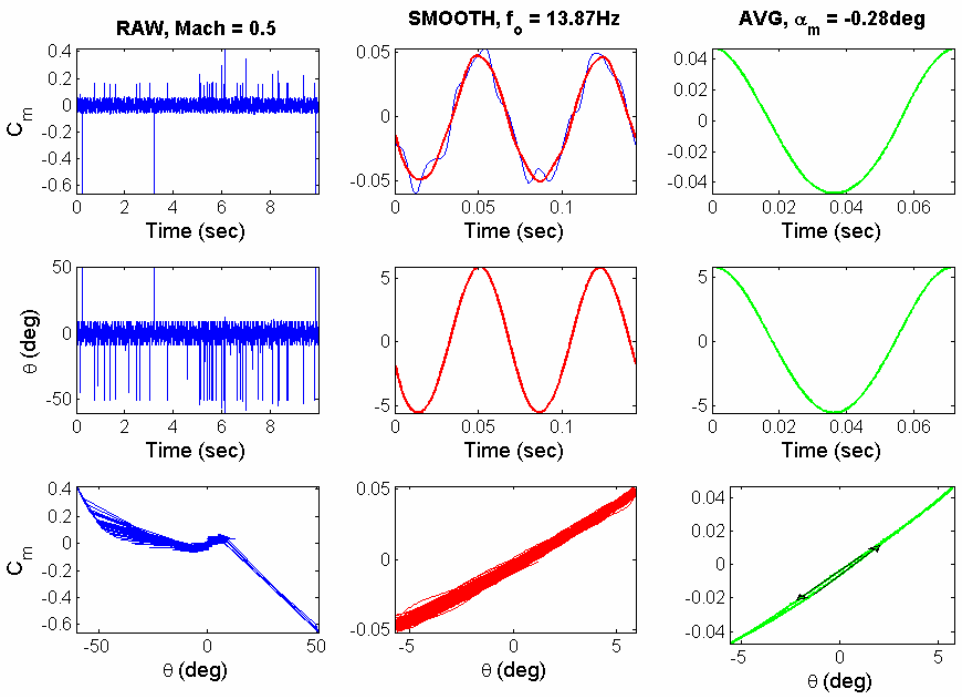

Figure 95 Expert at Φ = 0° forced oscillations (Mach = 0.5, f_o = 13.9 Hz)

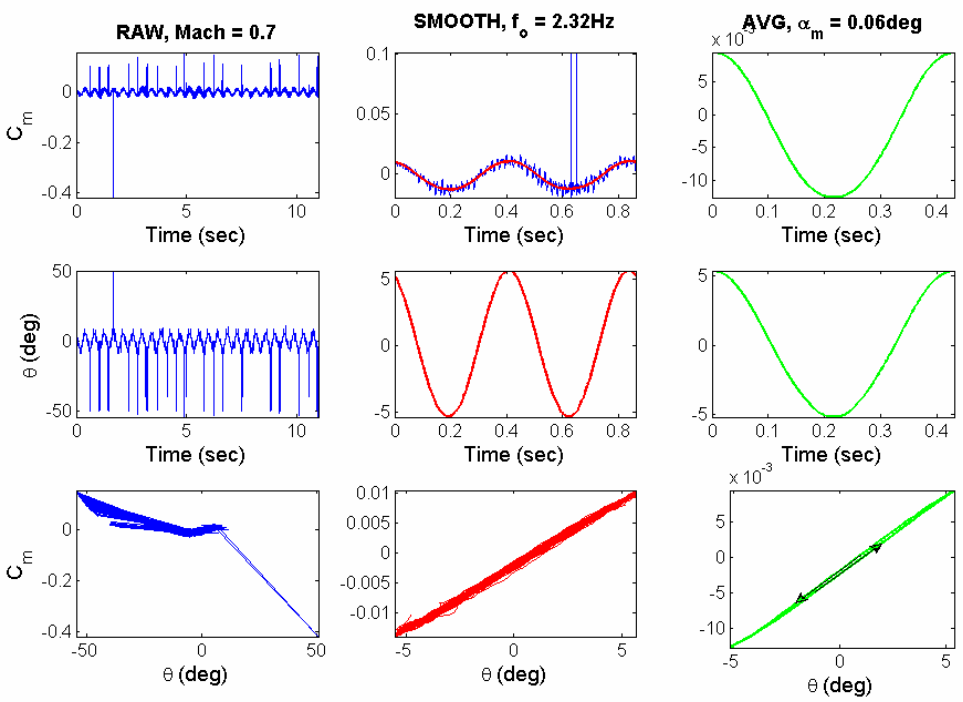

Figure 96 Expert at Φ = 0° forced oscillations (Mach = 0.7, f_o = 2.3 Hz)

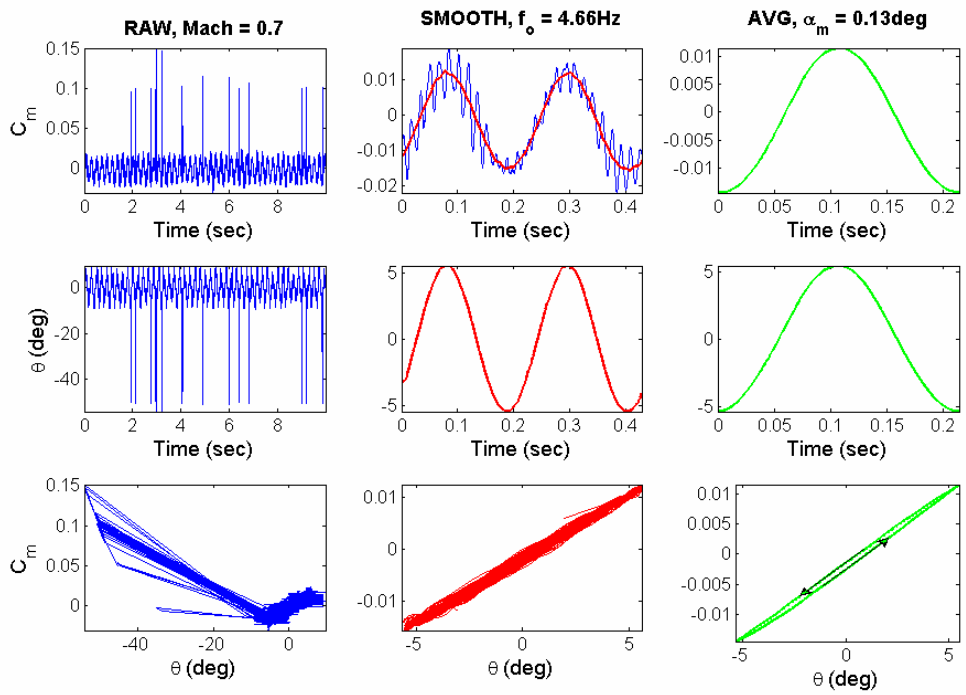

Figure 97 Expert at Φ = 0° forced oscillations (Mach = 0.7, f_o = 4.7 Hz)

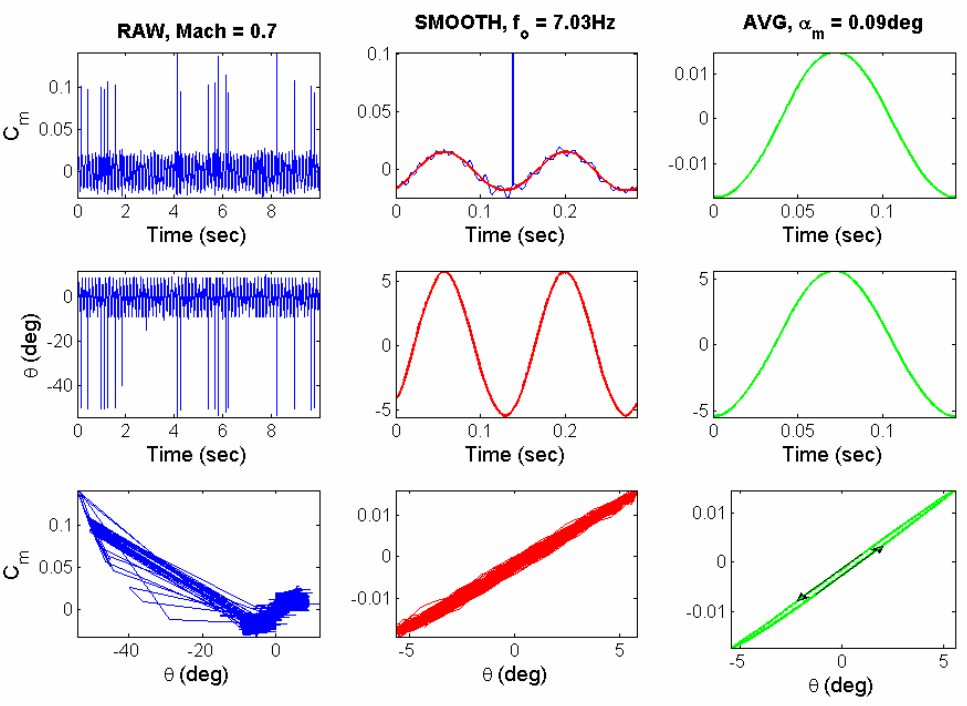

Figure 98 Expert at Φ = 0° forced oscillations (Mach = 0.7, f_o = 7 Hz)

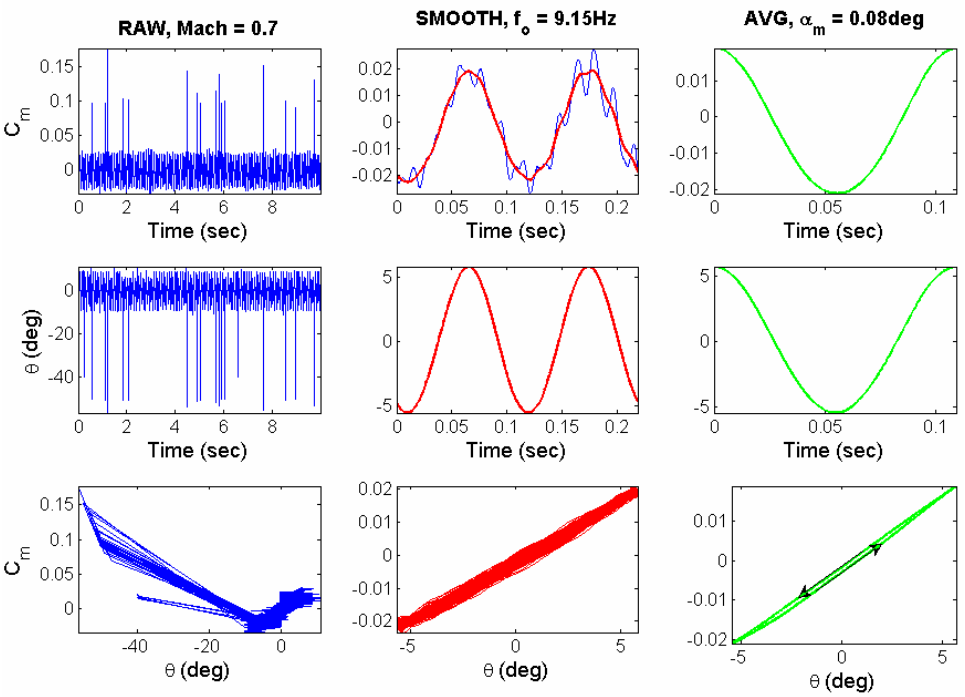

Figure 99 Expert at Φ = 0° forced oscillations (Mach = 0.7, f_o = 9.2 Hz)

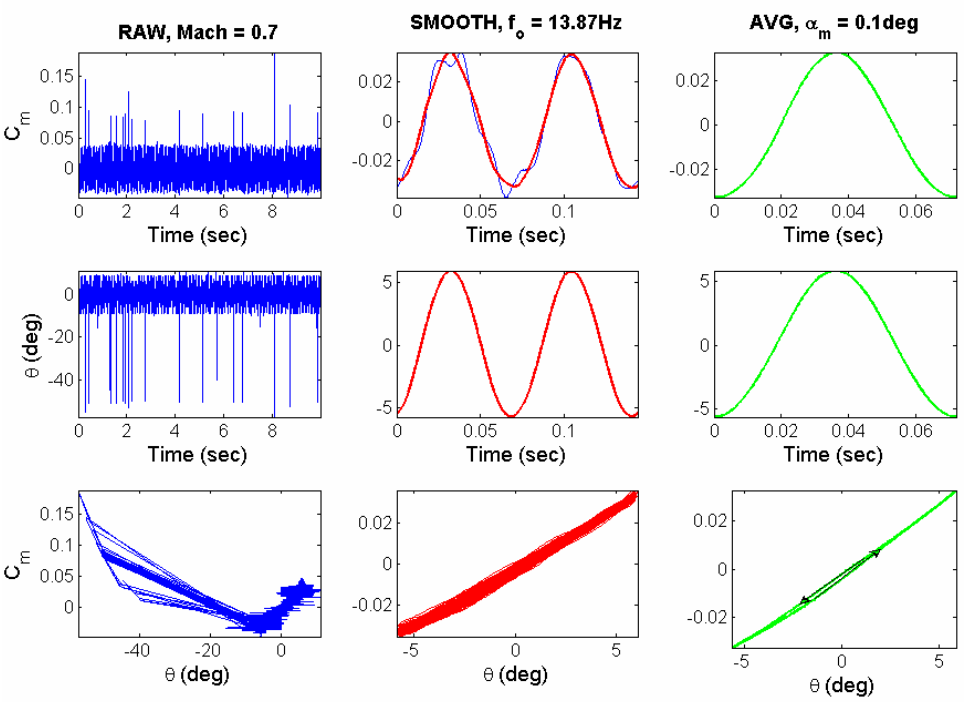

Figure 100 Expert at Φ = 0° forced oscillations (Mach = 0.7, f_o = 13.9 Hz)

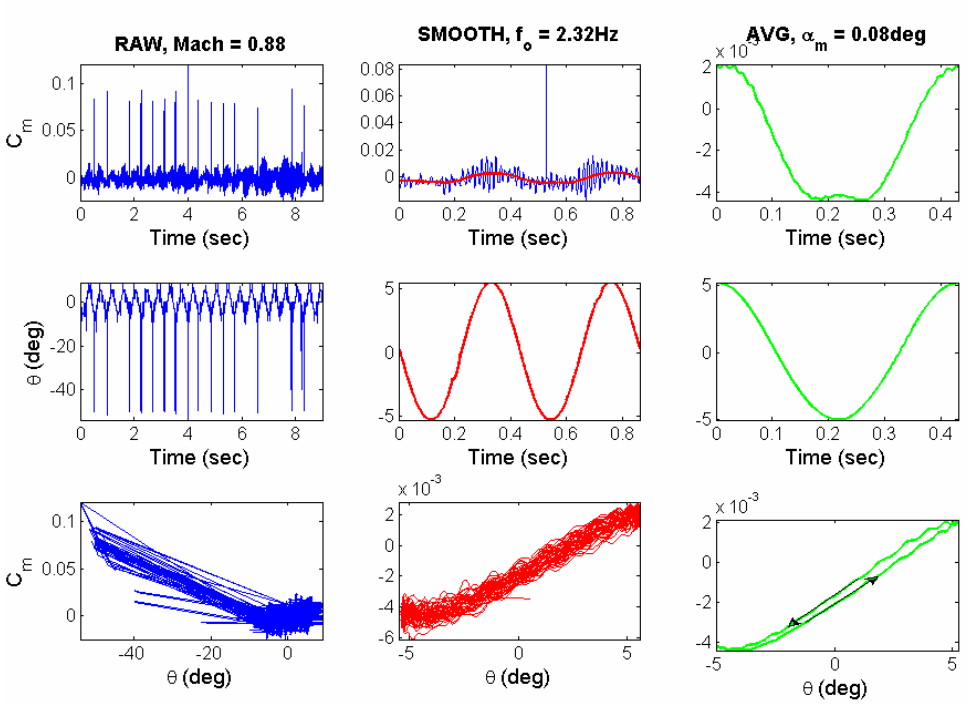

Figure 101 Expert at Φ = 0° forced oscillations (Mach = 0.88, f_o = 2.3 Hz)

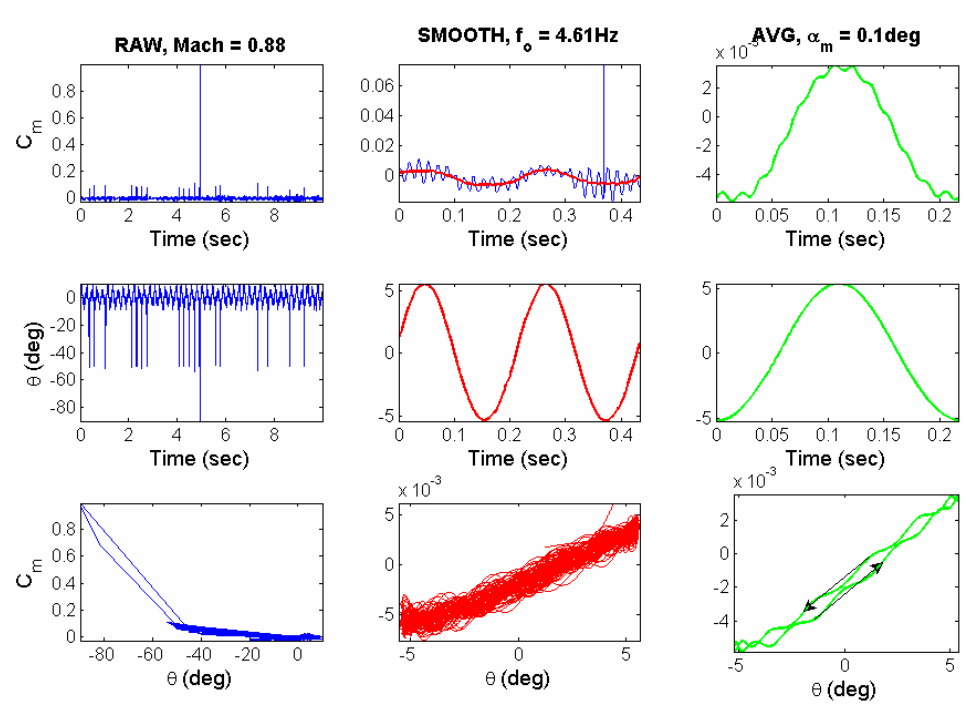

Figure 102 Expert at Φ = 0° forced oscillations (Mach = 0.88, f_o = 4.6 Hz)

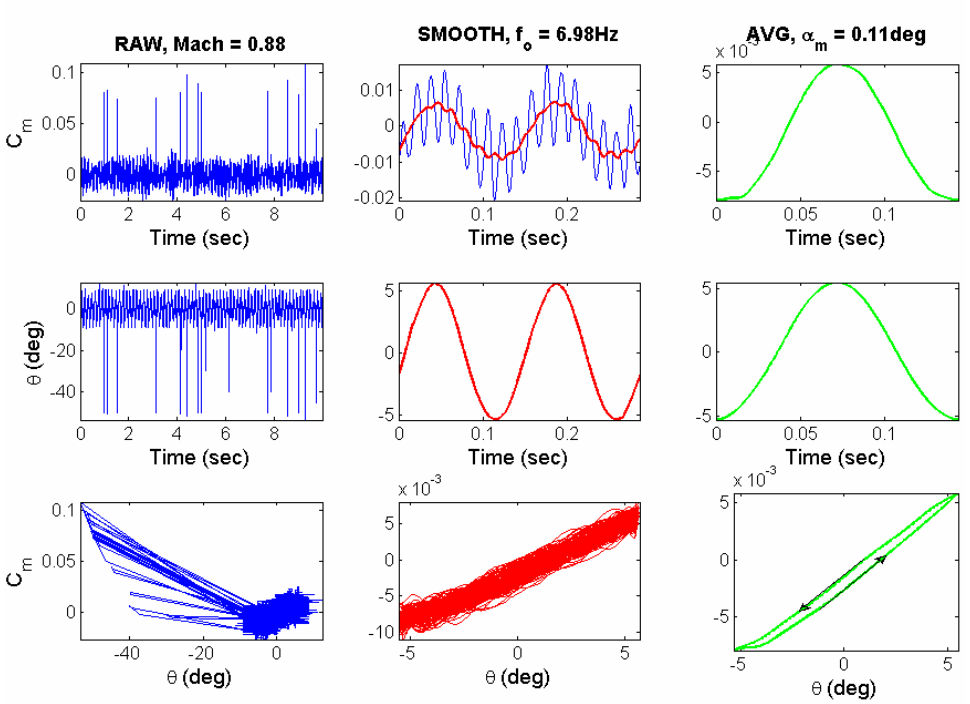

Figure 103 Expert at Φ = 0° forced oscillations (Mach = 0.88, f_o = 7 Hz)

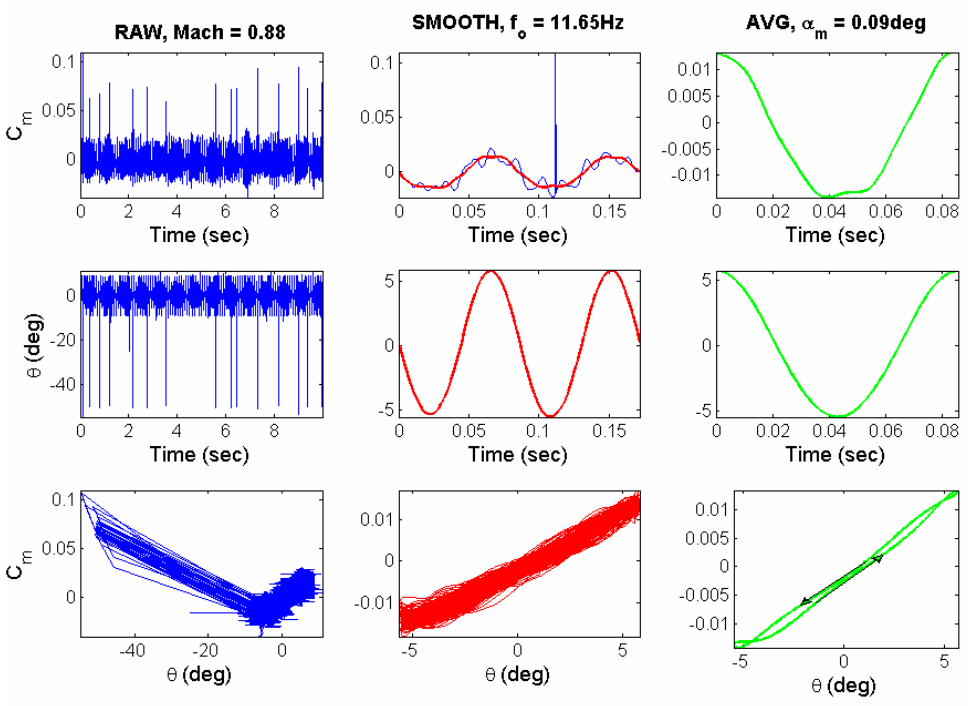

Figure 104 Expert at Φ = 0° forced oscillations (Mach = 0.88, f_o = 11.7 Hz)

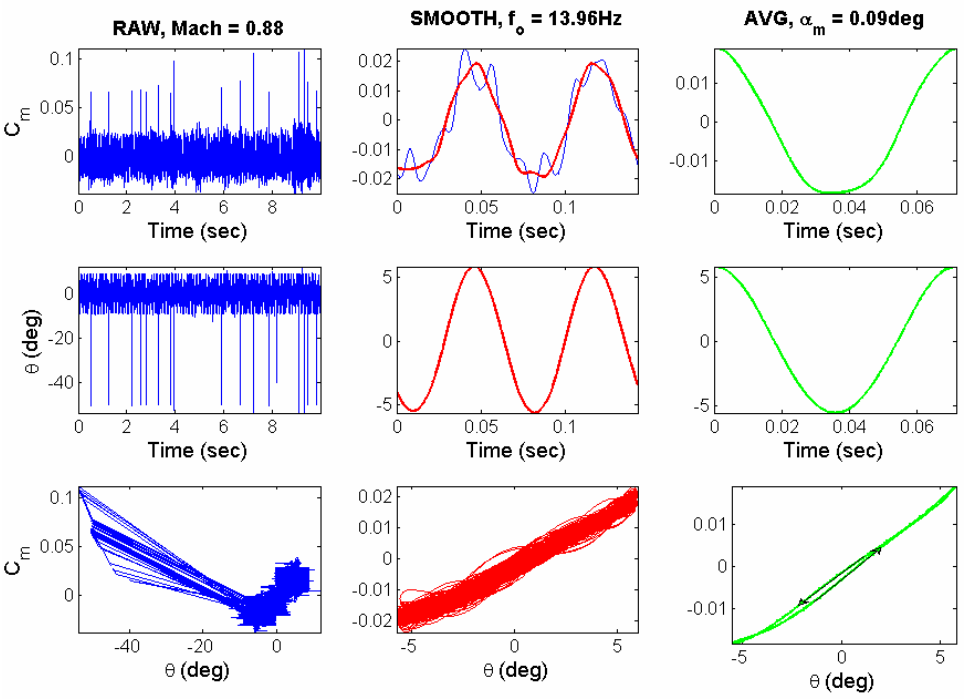

Figure 105 Expert at Φ = 0° forced oscillations (Mach = 0.88, f_o = 14 Hz)

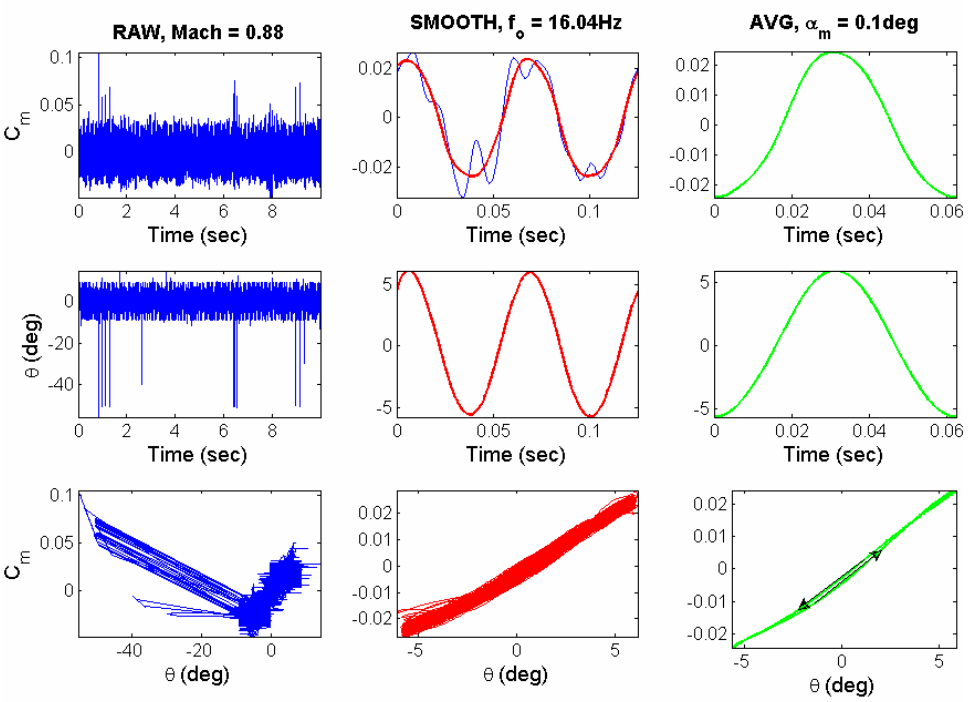

Figure 106 Expert at Φ = 0° forced oscillations (Mach = 0.88, f_o = 16 Hz)

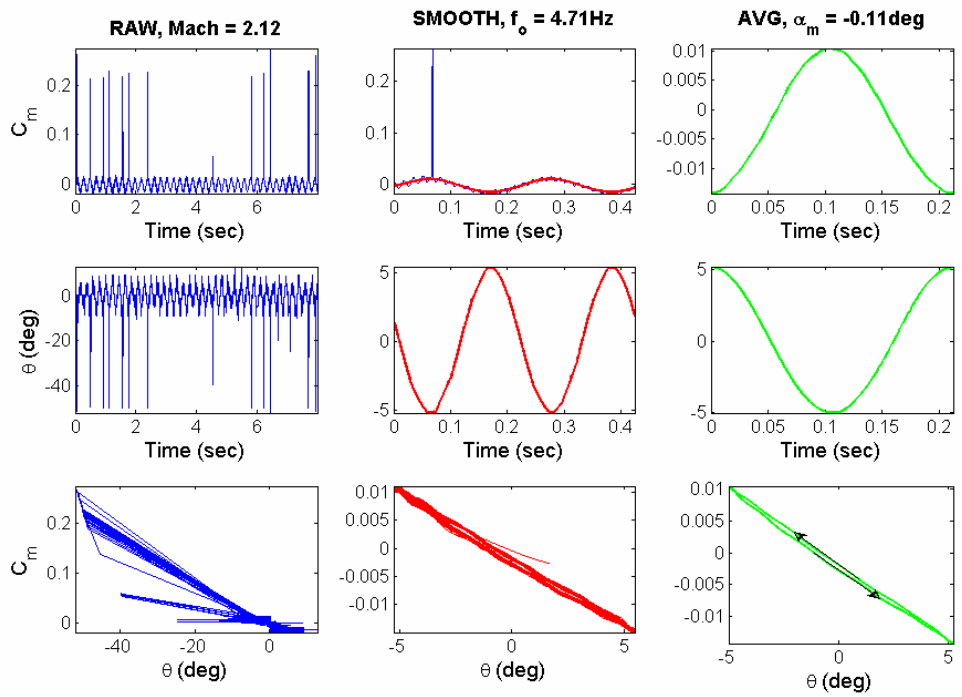

Figure 107 Expert at Φ = 0° forced oscillations (Mach = 2.12, f_o = 4.7 Hz)

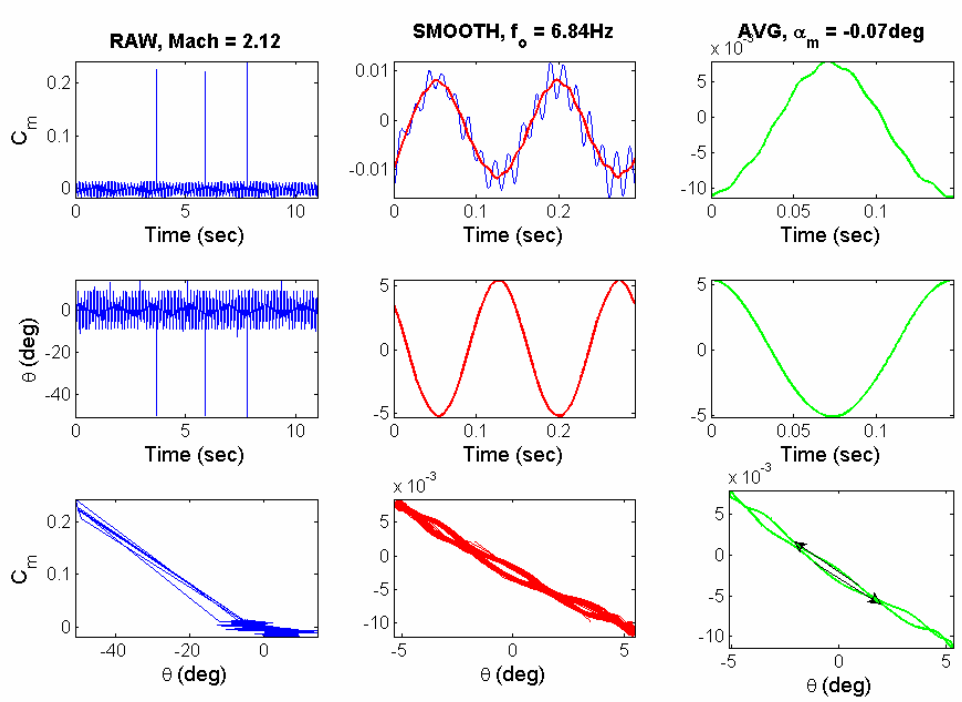

Figure 108 Expert at Φ = 0° forced oscillations (Mach = 2.12, f_o = 6.8 Hz)

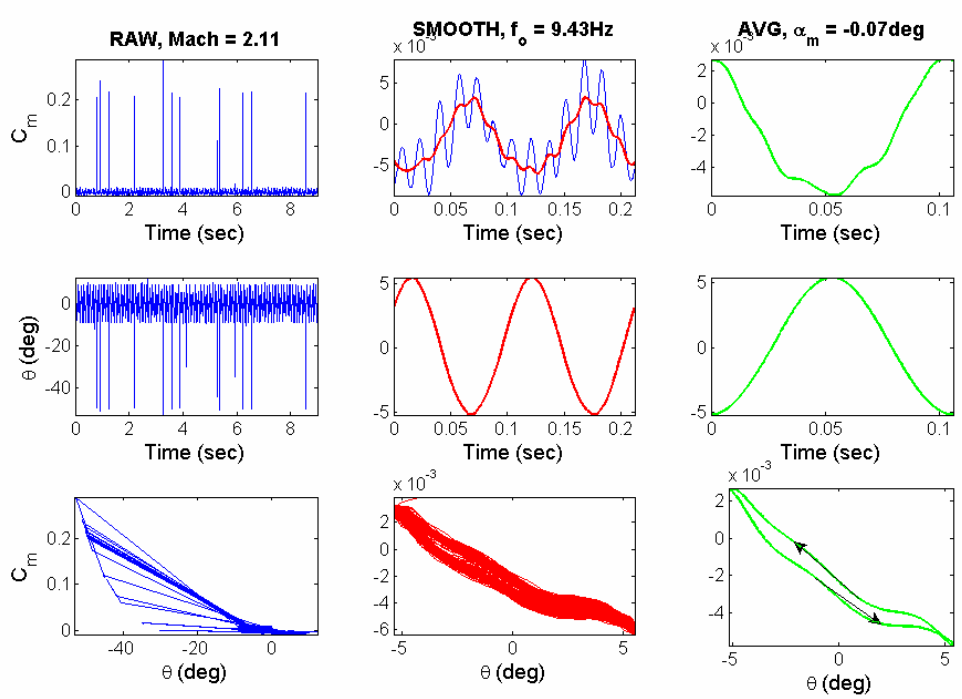

Figure 109 Expert at Φ = 0° forced oscillations (Mach = 2.11, f_o = 9.4 Hz)

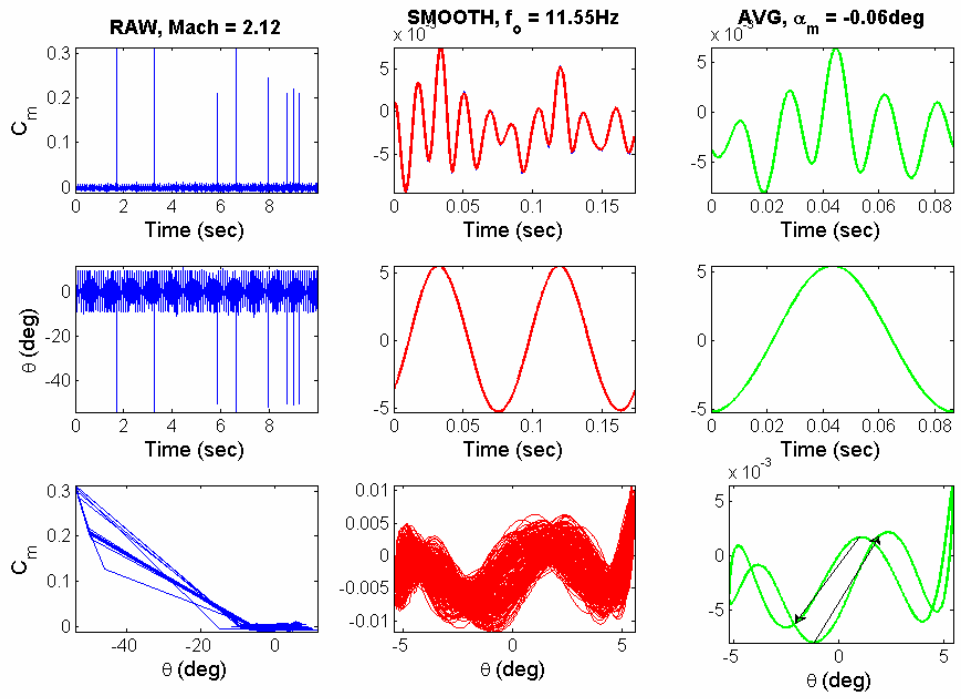

Figure 110 Expert at Φ = 0° forced oscillations (Mach = 2.12, f_o = 11.6 Hz)

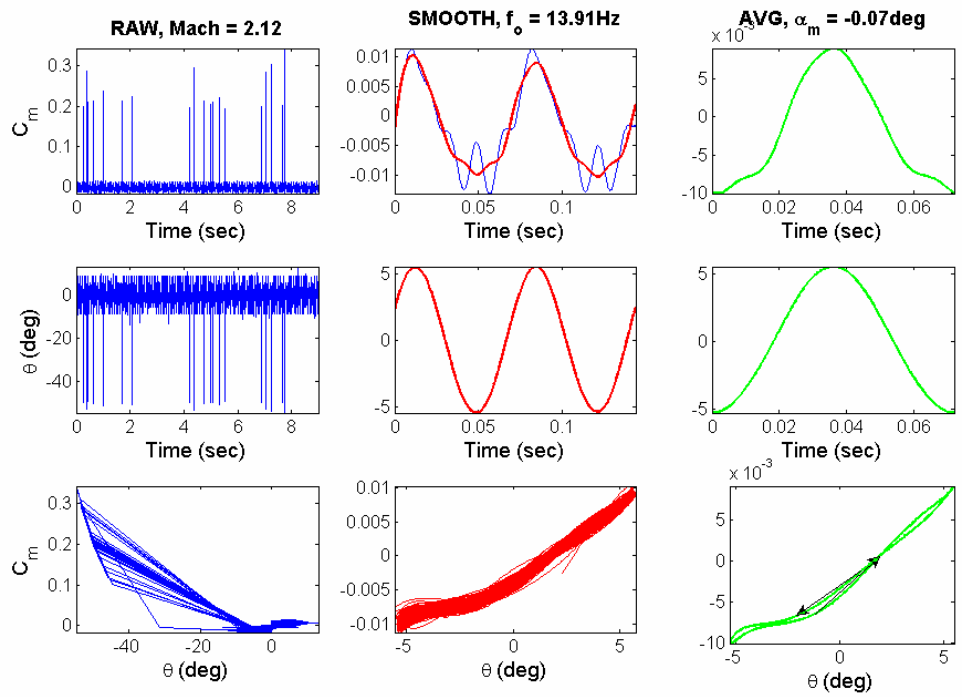

Figure 111 Expert at Φ = 0° forced oscillations (Mach = 2.12, f_o = 13.9 Hz)

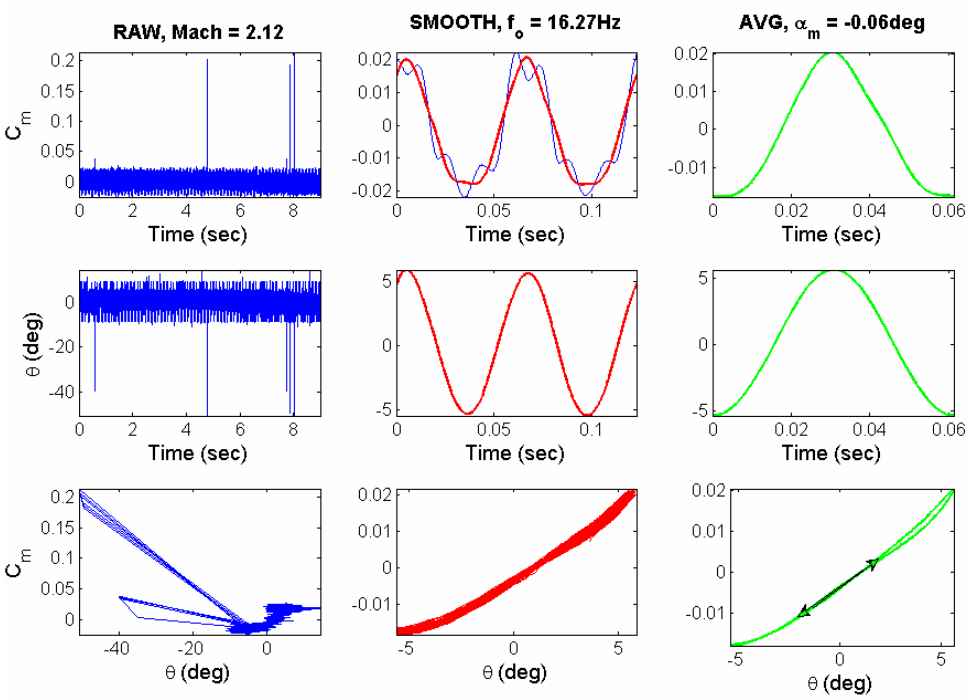

Figure 112 Expert at Φ = 0° forced oscillations (Mach = 2.12, f_o = 16.3 Hz)

A.2 Matlab Codes

func_ftt_algebra.m

```
% Takes the data from Free-to-Tumble (FTT) experiment and extracts the
% damping parameter using the ALGEBRA METHOD
%
% Solve: theta_dbldot + a*theta_dot + b*(theta-theta_0) = 0
%
% Notation:
% [CmqCmalphadot,a,b] = func_ftt_algebra(t,theta,theta_dot,theta_dbldot, ...
%                       q,S,D,V,I,Cm,dCm_dtheta,Cm_Angles)
%
% INPUTS:
% t = TIME [sec]
% theta = ANGLE OF ATTACK [rad]
% theta_dot = 1st derivative of theta w.r.t. time [rad/s]
% theta_dbldot = 2nd derivative of theta w.r.t. time [rad/s^2]
% theta_0 = Steady-state ANGLE OF ATTACK [rad] --- Not an actual input!
% q = DYNAMIC PRESSURE during testing [N/m^2]
% S = REFERENCE AREA of the model [m^2]
% D = CHARACTERISTIC LENGTH of the model [m]
% V = FREESTREAM VELOCITY during FO flow conditions [m/s]
% Cm = PITCHING MOMENT COEFFICIENT over range of Cm_Angles [~]
% dCm_dtheta = 1st derivative of Cm w.r.t. theta [rad-1]
% Cm_Angles = ANGLES OF ATTACK corresponding to dCm_dtheta vector [rad]

function [CmqCmalphadot,a,b] =
func_ftt_algebra(t,theta,theta_dot,theta_dbldot,q,S,D,V,I,Cm,dCm_dtheta,Cm_A
ngles)

theta_0 = theta(end);

for i = 1:length(t)
    if theta(i) < -pi
        row = find(theta(i)+2*pi < Cm_Angles);
        b(i) = -(q*S*D/I)*interp1(Cm_Angles(min(row)-
1:min(row)),dCm_dtheta(min(row)-1:min(row)),theta(i)+2*pi,'cubic');
    else
        row = find(theta(i) < Cm_Angles);
        b(i) = -(q*S*D/I)*interp1(Cm_Angles(min(row)-
1:min(row)),dCm_dtheta(min(row)-1:min(row)),theta(i),'cubic');
    end
    if theta_dot(i) == 0
        a(i) = a(i-1) - log(theta(i)) + log(theta(i-1));
    else
        a(i) = -(theta_dbldot(i) + b(i)*(theta(i)-theta_0))/theta_dot(i);
    end
end

for i = 1:length(a)
    CmqCmalphadot(i) = -a(i)*((2*V*I)/(q*S*D^2));
end
```

func_ftt_envelope.m

```
% Free to tumble analysis using the ENVELOPE METHOD
%
```

```
% Using the values of x vs. t (peaks) to solve 2nd order
% differential equation of the following form: x_dbldot + a*x_dot +
% b*x = 0
%
% Notation:
% [xnew,sigma,omega,C1,C2] = func_ftt_envelope(t,x,x_dot)
%
% INPUTS:
% t = Time [sec]
% x = Signal
% x_dot = Derivative of x signal w.r.t. time
%
% OUTPUTS:
% xnew = Derived signal based on this method
% sigma = Exponential constant of the outer envelope [sec-1], a = -2*sigma
% omega = Frequency of the signal [rad/s], b = (sigma^2 + omega^2)
% C1 = Boundary constant to meet equation solution
% C2 = Initial value constant to meet equation solution

function [xnew,sigma,omega,C1,C2] = func_ftt_envelope(t,x,x_dot)


index = 1;
i = 1;

% Evaluates the peaks and valleys
while index < length(x)
    hi = find(x == max(x(index:end)));
    if length(hi) == 1
        high(i) = hi;
    else
        j = 1;
        while hi(j) < index
            j = j+1;
        end
        high(i) = hi(j);
    end
    index = high(i);
    lo = find(x == min(x(index:end)));
    if length(lo) == 1
        low(i) = lo;
    else
        j = 1;
        while lo(j) < index
            j = j+1;
        end
        low(i) = lo(j);
    end
    index = low(i);
    i = i+1;
end

% Evaluates the period from one extreme to the next like-extreme
for i = 1:length(high)-1
    T_high(i) = t(high(i+1))-t(high(i));
    T_low(i) = t(low(i+1))-t(low(i));
end

% Determines the steady-state condition
for i = 1:length(x)
```

```
    mn(i) = mean(x(i:end));
end

[count,val] = hist(mn,length(mn));
p = find(count == max(count));

avg = val(p);

% Evaluates the amplitude ratio based on steady-state condition
for i = 1:length(high)-1
    ratio_high(i) = abs((x(high(i))-avg))/abs((x(high(i+1))-avg));
    ratio_low(i) = abs((avg-x(low(i))))/abs((avg-x(low(i+1))));
end

% Evaluates the value of a and b
a_high = 2./T_high.*log(ratio_high);
a_low = 2./T_low.*log(ratio_low);

b_high = (2*pi./T_high).^2 + (a_high./2).^2;
b_low = (2*pi./T_low).^2 + (a_low./2).^2;

% ************************************************************************
j = 1;

while j < length(high)
    for i = high(j):high(j+1)
        a(i) = a_high(j);
        b(i) = b_high(j);
    end
    j = j + 1;
end

for i = 1:length(a)
    sigma(i) = -a(i)/2;
    omega(i) = sqrt(4*b(i) - a(i)^2)/2;
end

for i = 1:length(a)
    k1(i) = x(i)/exp(sigma(i)*t(i));
    k2(i) = (x_dot(i) - sigma(i)*x(i))/(omega(i)*exp(sigma(i)*t(i)));
    C1(i) = k1(i)*cos(omega(i)*t(i)) - k2(i)*sin(omega(i)*t(i));
    C2(i) = k2(i)*cos(omega(i)*t(i)) + k1(i)*sin(omega(i)*t(i));
end

for i = 1:length(a)
    xnew(i) = exp(sigma(i)*t(i))*(C1(i)*cos(omega(i)*t(i)) +
C2(i)*sin(omega(i)*t(i)));
end
```

func_ftt_lin_alg.m

```
% This is the free to tumble analysis using linear algebra
%
% Notation:
% [xnew,sigma,omega,C1,C2] =
func_ftt_lin_alg(time,x,x_dot,x_dbldot,x_trpdot)
%
% Note that x and corresponding derivatives must be in radians or unitless
%
% INPUTS:
```

```
% time = Time [sec]
% x = Signal
% x_dot = Derivative of x signal w.r.t. time
% x_dbldot = 2nd "          "
% x_trpdot = 3rd "          "
%
% OUTPUTS:
% xnew = Derived signal based on this method
% sigma = Exponential constant of the outer envelope [sec-1], a = -2*sigma
% omega = Frequency of the signal [rad/s], b = (sigma^2 + omega^2)
% C1 = Boundary constant to meet equation solution
% C2 = Initial value constant to meet equation solution

function [xnew,sigma,omega,C1,C2] =
func_ftt_lin_alg(t,x,x_dot,x_dbldot,x_trpdot)

for i = 1:length(x)
    Y1(:,:,i) = [x_dbldot(i); x_trpdot(i)];
    A1(:,:,i) = [2*x_dot(i) -x(i); 2*x_dbldot(i) -x_dot(i)];
    X1(:,:,i) = inv(A1(:,:,i))*Y1(:,:,i);

    sigma(i) = X1(1,1,i);
    omega(i) = sqrt(X1(2,1,i) - sigma(i)^2);
end

for i = 1:length(x)
    k1(i) = x(i)/exp(sigma(i)*t(i));
    k2(i) = (x_dot(i) - sigma(i)*x(i))/(omega(i)*exp(sigma(i)*t(i)));
    Y1(:,:,i) = [k1(i); k2(i)];
    A1(:,:,i) = [cos(omega(i)*t(i)) sin(omega(i)*t(i)); -sin(omega(i)*t(i))
cos(omega(i)*t(i))];
    X1(:,:,i) = inv(A1(:,:,i))*Y1(:,:,i);

    C1(i) = X1(1,1,i);
    C2(i) = X1(2,1,i);
end

for i = 1:length(sigma)
    xnew(i) = exp(sigma(i)*t(i))*(C1(i)*cos(omega(i)*t(i)) +
C2(i)*sin(omega(i)*t(i)));
end
```

func_fo_algebra.m

```
% ALGEBRAIC METHOD for forced oscillation analysis
%
% Notation:
% [CmqCmalphadot] = func_fo_algebra(AoA_st,dCm_dAoA_st,AoAlock_f,timelock_f,
...
%
Mlock_f,AoAlock_nf,timelock_nf,Mlock_nf,I,q,S,D,V,method)
%
% Solution:
% INERTIA METHOD
% CmqCmalphadot = (I*(d2theta_dt2_f - d2theta_dt2_nf2) -
((q*S*D)*(dCm_dtheta_st2)*theta_f)) ...
%                     / (((q*S*D)/(2*V))*dtheta_dt_f);
%
% MOMENT METHOD
```

```
% CmqCmalphadot = ((Mlock_f - Mlock_nf2) -
((q*S*D)*(dCm_dtheta_st2)*theta_f)) ...
%                    / (((q*S*D)/(2*V))*dtheta_dt_f);
%
% INPUTS:
% AoA_st = Static moment data ANGLE OF ATTACK [deg]
% dCm_dAoA_st = SLOPE of Cm w.r.t ANGLE OF ATTACK [rad-1]
% AoAlock_f = Phase averaged ANGLE OF ATTACK during FO flow conditions [deg]
% timelock_f = TIME over one period of AoAlock_f [sec]
% Mlock_f = MOMENT over one period of AoAlock_f [N-m]
% AoAlock_nf = Phase averaged ANGLE OF ATTACK durin no flow conditions [deg]
% timelock_nf = TIME over one period of AoAlock_nf [sec]
% Mlock_nf = MOMENT over one period of AoAlock_nf [N-m]
% I = INERTIA of the oscillation system (model + support) [kg-m^2]
% q = DYNAMIC PRESSURE during the FO flow conditions [N/m^2]
% S = REFERENCE AREA of the model [m^2]
% D = CHARACTERISTIC LENGTH of the model [m]
% V = FREESTREAM VELOCITY during FO flow conditions [m/s]
% method = 'inertia' or 'moment'
%
% OUTPUTS:
% CmqCmalphadot = AERODYNAMIC DAMPING IN PITCH parameter [rad-1]
%                   also equals [dCm_d(qD/2V)+dCm_d(alphadotD/2V)]


function [CmqCmalphadot] =
func_fo_algebra(AoA_st,dCm_dAoA_st,AoAlock_f,timelock_f, ...
             Mlock_f,AoAlock_nf,timelock_nf,Mlock_nf,I,q,S,D,V,method)

% Flow Conditions
theta_f = (AoAlock_f-mean(AoAlock_f))*(pi/180);                    %[rad]
dtheta_dt_f = func_nth_derivative(timelock_f,theta_f,1);           %[rad/s]
d2theta_dt2_f = func_nth_derivative(timelock_f,dtheta_dt_f,1);
%[rad/s^2]

% No Flow Conitions
theta_nf = (AoAlock_nf-mean(AoAlock_nf))*(pi/180);                 %[rad]
dtheta_dt_nf = func_nth_derivative(timelock_nf,theta_nf,1);        %[rad/s]
d2theta_dt2_nf = func_nth_derivative(timelock_nf,dtheta_dt_nf,1);
%[rad/s^2]

theta_st = (AoA_st-mean(AoAlock_f))*(pi/180);                      %[rad]
dCm_dtheta_st = dCm_dAoA_st;                                       %[rad-1]

q = mean(q);                                                       %[N/m^2]
V = mean(V);                                                       %[m/s]

sorttheta_nf = sort(theta_nf);
sorttheta_st = sort(theta_st);

j = 1;
while j <= length(sorttheta_nf)
    hold = find(sorttheta_nf(j) == theta_nf);
    if length(hold) == 1
        row_nf(j) = hold;
        j = j+1;
    else
        row_nf(j) = hold(1);
        row_nf(j+1) = hold(2);
        j = j+2;
```

```
    end
end

j = 1;
while j <= length(sorttheta_st)
    hold = find(sorttheta_st(j) == theta_st);
    if length(hold) == 1
        row_st(j) = hold;
        j = j+1;
    else
        row_st(j) = hold(1);
        row_st(j+1) = hold(2);
        j = j+2;
    end
end

% Evaluation no flow and static conditions for theta_nf = theta_f!!!!!
d2theta_dt2_nf2 =
interp1(theta_nf(row_nf),d2theta_dt2_nf(row_nf),theta_f,'cubic')
Mlock_nf2 = interp1(theta_nf(row_nf),Mlock_nf(row_nf),theta_f,'cubic');
dCm_dtheta_st2 =
interp1(theta_st(row_st),dCm_dtheta_st(row_st),theta_f,'cubic')

% ************************************************************************
% ****************          ALGEBRAIC METHOD        **********************
% ************************************************************************
if method == 'inertia'

    for i = 1:length(d2theta_dt2_f)
        CmqCmalphadot(i) = (I*(d2theta_dt2_f(i) - d2theta_dt2_nf2(i)) -
((q*S*D).*(dCm_dtheta_st2(i)).*theta_f(i))) ...
                    / (((q*S*D)/(2*V))*dtheta_dt_f(i));
    end
end

if method == 'moment'

    for i = 1:length(Mlock_f)
        CmqCmalphadot(i) = ((Mlock_f(i) - Mlock_nf2(i)) -
((q*S*D).*(dCm_dtheta_st2(i))*theta_f(i))) ...
                    / (((q*S*D)./(2*V)).*dtheta_dt_f(i));
    end
end
```

func_fo_phase_diff_amp.m

```
% Uses the PHASE DIFFERENCE AMPLITUDE METHOD to extract
% dCm_dq+dCm_dalphadot [rad-1] (aerodynamic damping) at a given AoA [deg]
%
% Notation:
% [CmqCmalphadot,dCm_dalpha,alpha] =
func_fo_phase_diff_amp(timelock_f,AoAlock_f, ...
%
Mlock_f,period_f,freq_f,timelock_nf,AoAlock_nf,Mlock_nf,period_nf,freq_nf,I,
q,S,D,V,method)
%
% INPUTS:
% timelock_f = Phase averaged time over FO tests during flow    [sec]
% AoAlock_f = Phase averaged AoA corresponding to timelock_f    [deg]
% Mlock_f = "       " Moment "      "                           [N-m]
```

```
% period_f = Period for FO tests during flow                    [sec]
% freq_f = Frequency for FO tests ""    ""                      [Hz]
% timelock_nf = Phase averaged time over FO tests during no flow    [sec]
% AoAlock_nf = Phase averaged AoA corresponding to timelock_nf      [deg]
% Mlock_nf = "       " Moment "      "                           [N-m]
% period_nf = Period for FO tests during no flow                 [sec]
% freq_nf = Frequency for FO tests ""    ""                      [Hz]
% I = INERTIA of the oscillation system (model + support)        [kg-m^2]
% q = DYNAMIC PRESSURE during the FO flow conditions             [N/m^2]
% S = REFERENCE AREA of the model                                [m^2]
% D = CHARACTERISTIC LENGTH of the model                         [m]
% V = FREESTREAM VELOCITY during FO flow conditions              [m/s]
% method = 'lumped' - coupled flexure (from balance) and forcing moment
%       or 'unlumped' - uncoupled with theoretical forcing moment base on
%       mechanical system
%
% OUTPUTS:
% CmqCmalphadot = AERODYNAMIC DAMPING IN PITCH parameter [rad-1]
%                   also equals [dCm_d(qD/2V)+dCm_d(alphadotD/2V)]
% dCm_dalpha = Static Moment-in-Pitch derivative at alpha [rad-1]
% alpha = Angle of Attack corresponding to the other two outputs!!! [deg]

function [CmqCmalphadot,dCm_dalpha,alpha] = ...
    func_fo_phase_diff_amp(timelock_f,AoAlock_f,Mlock_f,period_f,freq_f, ...
    timelock_nf,AoAlock_nf,Mlock_nf,period_nf,freq_nf,I,q,S,D,V,method)


q = mean(q);                                                    %[N/m^2]
V = mean(V);                                                    %[m/s]

% ************************************************************************
% ************************************************************************
t_f = timelock_f;                                               %[sec]
theta_f = AoAlock_f-mean(AoAlock_f);                            %[deg]
M_f = Mlock_f-mean(Mlock_f);                                    %[N-m]

theta_fp = ((max(theta_f)-min(theta_f))/2)*sin(2*pi*freq_f*t_f); %[deg]
M_fp = ((max(M_f)-min(M_f))/2)*sin(2*pi*freq_f*t_f);            %[N-m]

muM_f_max=(2*pi)*((t_f(find(max(M_fp)==M_fp))-
t_f(find(max(M_f)==M_f)))/period_f); %[rad]
muM_f_min=(2*pi)*((t_f(find(min(M_fp)==M_fp))-
t_f(find(min(M_f)==M_f)))/period_f); %[rad]
mutheta_f_max=(2*pi)*((t_f(find(max(theta_fp)==theta_fp))-
t_f(find(max(theta_f)==theta_f)))/period_f);      %[rad]
mutheta_f_min=(2*pi)*((t_f(find(min(theta_fp)==theta_fp))-
t_f(find(min(theta_f)==theta_f)))/period_f);      %[rad]

mu_f_max = muM_f_max - mutheta_f_max;                           %[rad]
mu_f_min = muM_f_min - mutheta_f_min;                           %[rad]

% ************************************************
mu_f = (mu_f_max+mu_f_min)/2                                    %[rad]
Mbar_f = (max(M_f)-min(M_f))/2                                  %[N-m]
thetabar_f = ((max(theta_f)-min(theta_f))/2)*pi/180             %[rad]
omega_f = 2*pi*freq_f                                           %[rad/s]
% ************************************************
% Only no flow (nf) conditions are determined by the lumped and unlumped
methods
if method == 'lumped'
```

```
    t_nf = timelock_nf;                                          %[sec]
    theta_nf = AoAlock_nf-mean(AoAlock_nf);                      %[deg]
    M_nf = Mlock_nf-mean(Mlock_nf);                              %[N-m]

    theta_nfp = ((max(theta_nf)-min(theta_nf))/2)*sin(2*pi*freq_nf*t_nf);
%[deg]
    M_nfp = ((max(M_nf)-min(M_nf))/2)*sin(2*pi*freq_nf*t_nf); %[N-m]

    muM_nf_max=(2*pi)*((t_nf(find(max(M_nfp)==M_nfp))-
t_nf(find(max(M_nf)==M_nf)))/period_nf);              %[rad]
    muM_nf_min=(2*pi)*((t_nf(find(min(M_nfp)==M_nfp))-
t_nf(find(min(M_nf)==M_nf)))/period_nf);              %[rad]
    mutheta_nf_max=(2*pi)*((t_nf(find(max(theta_nfp)==theta_nfp))-
t_nf(find(max(theta_nf)==theta_nf)))/period_nf);  %[rad]
    mutheta_nf_min=(2*pi)*((t_nf(find(min(theta_nfp)==theta_nfp))-
t_nf(find(min(theta_nf)==theta_nf)))/period_nf);  %[rad]

    mu_nf_max = muM_nf_max - mutheta_nf_max;                  %[rad]
    mu_nf_min = muM_nf_min - mutheta_nf_min;                  %[rad]

    % ***************************************************
    mu_nf = (mu_nf_max+mu_nf_min)/2                           %[rad]
    Mbar_nf = (max(M_nf)-min(M_nf))/2                         %[N-m]
    thetabar_nf = ((max(theta_nf)-min(theta_nf))/2)*pi/180    %[rad]
    omega_nf = 2*pi*freq_nf                                   %[rad/s]
    % ***************************************************

    Mtheta = ((omega_f^2/(thetabar_nf*omega_nf^2))*Mbar_nf*cos(mu_nf))-
((Mbar_f*cos(mu_f))/thetabar_f);
    Mtheta_dot = -(Mbar_f*sin(mu_f))/(omega_f*thetabar_f);
    dCm_dalpha = Mtheta/(q*S*D);                               %[rad-1]
    CmqCmalphadot = Mtheta_dot*2*V/(q*S*D^2);                  %[rad-1]
end
if method == 'unlumped'
    osc_amp = (max(AoAlock_nf)-min(AoAlock_nf))/2;             %[deg]
    [M_nf,t_nf,theta_nf,theta_dot_nf,theta_dbldot_nf] =
func_forcing_moment(I,osc_amp,freq_nf);

    theta_nfp = ((max(theta_nf)-min(theta_nf))/2)*sin(2*pi*freq_nf*t_nf);
%[deg]
    M_nfp = ((max(M_nf)-min(M_nf))/2)*sin(2*pi*freq_nf*t_nf);   %[N-m]

    muM_nf_max=(2*pi)*((t_nf(find(max(M_nfp)==M_nfp))-
t_nf(find(max(M_nf)==M_nf)))/period_nf);              %[rad]
    muM_nf_min=(2*pi)*((t_nf(find(min(M_nfp)==M_nfp))-
t_nf(find(min(M_nf)==M_nf)))/period_nf);              %[rad]
    mutheta_nf_max=(2*pi)*((t_nf(find(max(theta_nfp)==theta_nfp))-
t_nf(find(max(theta_nf)==theta_nf)))/period_nf);  %[rad]
    mutheta_nf_min=(2*pi)*((t_nf(find(min(theta_nfp)==theta_nfp))-
t_nf(find(min(theta_nf)==theta_nf)))/period_nf);  %[rad]

    mu_nf_max = muM_nf_max - mutheta_nf_max;                  %[rad]
    mu_nf_min = muM_nf_min - mutheta_nf_min;                  %[rad]

    % ***************************************************
    mu_nf = (mu_nf_max+mu_nf_min)/2                           %[rad]
    Mbar_nf = (max(M_nf)-min(M_nf))/2                         %[N-m]
    thetabar_nf = ((max(theta_nf)-min(theta_nf))/2)*pi/180    %[rad]
    omega_nf = 2*pi*freq_nf                                   %[rad/s]
    % ***************************************************
```

```
    Mtheta = ((Mbar_nf*cos(mu_nf))/thetabar_nf)-
((Mbar_f*cos(mu_f))/thetabar_f)+(I*(omega_nf^2-omega_f^2));
    Mtheta_dot = ((Mbar_nf*sin(mu_nf))/(omega_nf*thetabar_nf))-
((Mbar_f*sin(mu_f))/(omega_f*thetabar_f));
    dCm_dalpha = Mtheta/(q*S*D);                              %[rad-1]
    CmqCmalphadot = Mtheta_dot*2*V/(q*S*D^2);                 %[rad-1]
end
alpha = mean(AoAlock_f);                                      %[deg]
```

func_fo_energy.m

```
% ENERGY METHOD for forced oscillation analysis
%
% Notation:
% [CmqCmalphadot,alpha] = func_fo_energy(AoA_st,dCm_dAoA_st,AoAlock_f, ...
%                                    timelock_f,Cmlock_f,I,q,S,D,V)
%
% INPUTS:
% AoA_st = Static moment data ANGLE OF ATTACK [deg]
% dCm_dAoA_st = SLOPE of Cm w.r.t ANGLE OF ATTACK [rad-1]
% AoAlock_f = Phase averaged ANGLE OF ATTACK during FO flow conditions [deg]
% timelock_f = TIME over one period of AoAlock_f [sec]
% Cmlock_f = PITCH MOMENT COEFFICIENT over one period of AoAlock_f [~]
% I = INERTIA of the oscillation system (model + support) [kg-m^2]
% q = DYNAMIC PRESSURE during the FO flow conditions [N/m^2]
% S = REFERENCE AREA of the model [m^2]
% D = CHARACTERISTIC LENGTH of the model [m]
% V = FREESTREAM VELOCITY during FO flow conditions [m/s]
%
% OUTPUTS:
% CmqCmalphadot = AERODYNAMIC DAMPING IN PITCH parameter [rad-1]
%                   also equals [dCm_d(qD/2V)+dCm_d(alphadotD/2V)]
% alpha = Angle of Attack corresponding to CmqCmalphadot!!! [deg]

function [CmqCmalphadot,alpha] =
func_fo_energy(AoA_st,dCm_dAoA_st,AoAlock_f, ...
                                    timelock_f,Cmlock_f,I,q,S,D,V)
q = mean(q);
V = mean(V);

Cm = Cmlock_f;                                                    %[~]
dCm_dtheta = interp1(AoA_st,dCm_dAoA_st,mean(AoAlock_f),'cubic');  %[rad-1]
theta = (AoAlock_f-mean(AoAlock_f))*pi/180;                       %[rad]
thetabar = (max(theta)-min(theta))/2;                             %[rad]
zeta = -(trapz(theta,Cm))/(pi*thetabar^2);                        %[~]

alpha = mean(AoAlock_f);                                          %[deg]
CmqCmalphadot = -((2*V*I)/(q*S*D^2))*(2*zeta*sqrt(-(q*S*D/I)*dCm_dtheta));
%[rad-1]
```

func_static_moment_fo.m

```
% Static Moment Test Data Processing
% Returns test conditions, moments and incidences
%
% Notation:
%
[M,AoA,Pt,P,q,Mach,Tt,T,rho,V,Cm,dCm_dAoA,slope,AoA_NoM,Mach_NoM,T_NoM,P_NoM
,rho_NoM,V_NoM] = ...
```

```
%
func_static_moment_fo(Model,MachNo,MachNo2,D,S,no_of_smo_files,no_of_static_
files)
%
% Where each input is:
%
% Model = 'Apollo' or 'Expert0'
% MachNo = '05' for Mach 0.5 or '09' for Mach 0.9 and so on ...
% MachNo2 = "" when MachNo is '00' (use as MachNo when MachNo is not '00')
% D = Characteristic length for the model (also L)  [m]
% S = Reference Area of the model                   [m^2]
% no_of_smo_files = The total number of static moment offset files for that
%                   particular model at the designated MachNo
% no_of_static_files = The total number of static moment test files for that
%                      particular model at the designated MachNo
%
% Where each output is:
%
% M = PITCHING MOMENT [N-m]
% AoA = ANGLE OF ATTACK [deg]
% Pt = TOTAL PRESSURE during test conditions [Pa]
% P = STATIC PRESSURE during test conditions [Pa]
% q = DYNAMIC PRESSURE during test conditions [Pa]
% Mach = MACH NUMBER during test conditions [~]
% Tt = TOTAL TEMPERATURE during test conditions [K]
% T = STATIC TEMPERATURE during test conditions [K]
% rho = DENSITY during test conditions [kg/m^3]
% V = VELOCITY during test conditions [m/s]
% Cm = PITCHING MOMENT COEFFICIENT [~]
% dCm_dAoA = Derivative of PITCH MOMENT COEFFICIENT w.r.t. ANGLE OF ATTACK
[rad-1]
% slope = dCm_dAoA(Cm = 0) [rad-1]
% Mach_NoM = Mach(Cm = 0) [~]
% T_NoM = T(Cm = 0) [K]
% P_NoM = P(Cm = 0) [Pa]
% rho_NoM = rho(Cm = 0) [kg/m^3]
% V_NoM = V(Cm = 0) [m/s]

function
[M,AoA,Pt,P,q,Mach,Tt,T,rho,V,Cm,dCm_dAoA,slope,AoA_NoM,Mach_NoM,T_NoM,P_NoM
,rho_NoM,V_NoM] ...
          =
func_static_moment_fo(Model,MachNo,MachNo2,D,S,no_of_smo_files,no_of_static_
files)

% *************** Torque Balance Calibrations ***************
[Anew,Anew_unc,CL] = func_torque_bal_cal(0.9);

% ************************************************************************
% STATIC MOMENT OFFSETS - Loading Documents
% ************************************************************************
Jump_SMO=0;
Jump_Data=0;
for i = 1:no_of_smo_files
    document = strcat('Static Moment Offset
Data\',Model,'_Mach',MachNo,'\',Model,'_Mach',MachNo2,'_smo_',num2str(i),'.d
at');
    X1 = load(document);

% ***********************REMOVING ENCODER SPIKES************************
```

```
% Check correct angle of the encoder by comparing the value with the mean
% value of the file loaded with a tolerence of 2 degres

    SMO_mean = mean(X1);
    for j=1:size(X1,1)
        if ( abs( X1(j,4)-SMO_mean(4) ) < 2 )  % tolerence
            SMO_temp(j,:) = X1(j,:);
        else
            Jump_SMO=Jump_SMO + 1;
        end
    end

    SMO(i,:)=mean(SMO_temp);
    clear SMO_temp;
end

%## Use this line for Apollo model
Offset_Voltage = -SMO(:,2);
Offset_Angle = 180-SMO(:,4);

% %## Use this line for other models
% Offset_Voltage = SMO(:,2);
% Offset_Angle = SMO(:,4);

% ************************************************************************
% STATIC MOMENTS - Loading Documents
% ************************************************************************

for i = 1:no_of_static_files
    document = strcat('Static Moment Test
Data\',Model,'_Mach',MachNo,'\',Model,'_Mach',MachNo2,'_Condition_',num2str(
i),'.dat');
    X1 = load(document);
    Condition(i,:) = mean(X1);
    document2 = strcat('Static Moment Test
Data\',Model,'_Mach',MachNo,'\',Model,'_Mach',MachNo2,'_Data_',num2str(i),'.
dat');
    X2 = load(document2);

% ************************REMOVING ENCODER SPIKES*************************
% Check correct angle of the encoder by comparing the value with the mean
% value of the file loaded with a tolerence of 2 degres

    Data_mean = mean(X2);

    for j=1:size(X2,1)

        if (abs(X2(j,4)-Data_mean(4))<2)
            Data_temp(j,:) = X2 (j,:);

        else
            Jump_Data = Jump_Data+1;
        end
    end

    Data(i,:)=mean(Data_temp);
    clear Data_temp Data_mean ;
end

%## Use this line for Apollo model
```

```
Moment_Voltage = -Data(:,2);
Moment_Angle = 180-Data(:,4);

% %## Use this line for other models
% Moment_Voltage = Data(:,2);
% Moment_Angle = Data(:,4);

% ****************** EVALUATION **********************
M = Anew*(Moment_Voltage-
(interp1(Offset_Angle,Offset_Voltage,Moment_Angle,'cubic')))  ;    %[N-m]
AoA = Moment_Angle;                                  %[deg]
sortAoA = sort(AoA);                                %[deg]

for i = 1:length(sortAoA)
    row(i) = find(sortAoA(i) == AoA);
end

gamma = 1.4;
Pt = Condition(:,2)*0 + mean(Condition(:,2));
%[N/m^2]
Mach = Condition(:,4)*0 + mean(Condition(:,4));                          %[~]
P = Pt./((1+(((gamma-1)/2)*Mach.^2)).^(gamma/(gamma-1)));
%[N/m^2]
q = (gamma/2).*P.*Mach.^2;
%[N/m^2]
Tt = Condition(:,5)*0 + mean(Condition(:,5));                           %[K]
T = Tt./(1+(((gamma-1)/2)*Mach.^2));                                    %[K]
rho = P./(287.*T);
%[kg/m^3]
V = Mach.*((gamma*287.*T).^0.5);                                        %[m/s]
Cm = M./(q*S*D);                                                        %[~]

[dCm_dAoA] = func_nth_derivative(AoA,Cm,1);
dCm_dAoA = dCm_dAoA'*180/pi;                                            %[rad-
1]

% Use this for the Apollo model
slope = interp1(Cm(row),dCm_dAoA(row),0);
AoA_NoM = interp1(Cm(row),AoA(row),0);
Mach_NoM = interp1(Cm(row),Mach(row),0);
T_NoM = interp1(Cm(row),T(row),0);
P_NoM = interp1(Cm(row),P(row),0);
rho_NoM = interp1(Cm(row),rho(row),0);
V_NoM = interp1(Cm(row),V(row),0);

% % Use this for the Expert model
% slope = interp1(AoA(row),dCm_dAoA(row),0);
% AoA_NoM = interp1(AoA(row),AoA(row),0);
% Mach_NoM = interp1(AoA(row),Mach(row),0);
% T_NoM = interp1(AoA(row),T(row),0);
% P_NoM = interp1(AoA(row),P(row),0);
% rho_NoM = interp1(AoA(row),rho(row),0);
% V_NoM = interp1(AoA(row),V(row),0);
```

func_dynamic_fo.m

```
% Forced Oscillation Test Data Processing and Smoothing
% Returns test conditions, moments and incidences along with smoothed
% and phase-locked values
%
```

```
% Notation:
% [t,Pt,P,Mach,Tt,T,rho,V,q,time,timelock,AoA,AoAsmooth,AoAlock,M, ...
% Msmooth,Mlock,Cm,Cmsmooth,Cmlock,period,freq] = ...
%
func_dynamic_fo(Conditionfile,Datafile,Model,MachNo,MachNo2,D,S,no_of_smo_fi
les)
%
% Where each input is:  (EXAMPLES)
%
% Conditionfile = 'FO Test
Data\Apollo_Mach05\Apollo_Mach05_Condition_700.dat'
% Datafile = 'FO Test Data\Apollo_Mach05\Apollo_Mach05_Data_700.dat'
% Model = 'Apollo' or 'Expert0'
% MachNo = '05' for Mach 0.5 or '09' for Mach 0.9 and so on ...
% MachNo2 = "" when MachNo is '00' (use as MachNo when MachNo is not '00')
% D = Characteristic length for the model (also L)
% S = Characteristic surface for the model
% no_of_smo_files = The total number of static moment offset files for that
%                   particular model at the designated MachNo
%
% Where each output is:
%
% t = TIME during test conditions [sec]
% Pt = TOTAL PRESSURE during test conditions [Pa]
% P = STATIC PRESSURE during test conditions [Pa]
% Mach = MACH NUMBER during test conditions [~]
% Tt = TOTAL TEMPERATURE during test conditions [K]
% T = STATIC TEMPERATURE during test conditions [K]
% rho = DENSITY during test conditions [kg/m^3]
% V = VELOCITY during test conditions [m/s]
% q = DYNAMIC PRESSURE during test conditions [Pa]
% time = TIME for data acquisition [sec]
% timelock = TIME for one period of testing [sec]
% AoA = ANGLE OF ATTACK for data acquisition [deg]
% AOAsmooth = SMOOTHED ANGLE OF ATTACK using 'sgolay' method [deg]
% AOAlock = PHASE-LOCKED AVERAGE ANGLE OF ATTACK [deg]
% period_AoA = PERIODIC TIME OF ANGLE OF ATTACK oscillations [sec]
% freq_AoA = OSCILLATION FREQUENCY OF ANGLE OF ATTACK [Hz]
% M = PITCHING MOMENT for data acquisition [N-m]
% Msmooth = SMOOTHED PITCHING MOMENT using 'sgolay' method [N-m]
% Mlock = PHASE-LOCKED AVERAGE PITCHING MOMENT [N-m]
% period_M = PERIODIC TIME OF PITCHING MOMENT oscillations [sec]
% freq_M = OSCILLATION FREQUENCY OF PITCHING MOMENT [Hz]
% Cm = PITCHING MOMENT COEFFICIENT for data acquisition [~]
% Cmsmooth = SMOOTHED PITCHING MOMENT COEFFICIENT using 'sgolay' method [~]
% Cmlock = PHASE-LOCKED AVERAGE PITCHING MOMENT COEFFICIENT [~]
% period_Cm = PERIODIC TIME OF PITCHING MOMENT COEFFICIENT oscillations
[sec]
% freq_Cm = OSCILLATION FREQUENCY OF PITCHING MOMENT COEFFICIENT [Hz]

function [t,Pt,P,Mach,Tt,T,rho,V,q,time,timelock,AoA,AoAsmooth,AoAlock, ...
    M,Msmooth,Mlock,Cm,Cmsmooth,Cmlock,period,freq] = ...

func_dynamic_fo(Conditionfile,Datafile,Model,MachNo,MachNo2,D,S,no_of_smo_fi
les)

% *************** Torque Balance Calibrations ***************
[Anew,Anew_unc,CL] = func_torque_bal_cal(0.9);

% ********************************************************************
```

```
% Dynamic Data and Conditions
% ********************************************************************

Cond = load(Conditionfile);
Dat = load(Datafile);

Jump_SMO=0;
for i = 1:no_of_smo_files
    document = strcat('Static Moment Offset
Data\',Model,'_Mach',MachNo,'\',Model,'_Mach',MachNo2,'_smo_',num2str(i),'.d
at');
    X1 = load(document);

% ************************REMOVING ENCODER SPIKES*************************
% Check correct angle of the encoder by comparing the value with the mean
% value of the file loaded with a tolerence of 2 degres

    SMO_mean = mean(X1);
    for j=1:size(X1,1)
        if ( abs( X1(j,4)-SMO_mean(4) ) < 2 )  % tolerence
            SMO_temp(j,:) = X1(j,:);
        else
            Jump_SMO=Jump_SMO + 1;
        end
    end

    SMO(i,:)=mean(SMO_temp);
    clear SMO_temp;
end

%## Use this line for Apollo model
Offset_Voltage = -SMO(:,2);
Offset_Angle = 180-SMO(:,4);

% %## Use this line for Expert model
% Offset_Voltage = SMO(:,2);
% Offset_Angle = SMO(:,4);

% Testing Condition Data
gamma = 1.4;
t = Cond(:,1)-Cond(1,1);
Pt = Cond(:,2)*0 + mean(Cond(:,2));
Mach = Cond(:,4)*0 + mean(Cond(:,4));
P = Pt./((1+(((gamma-1)/2)*Mach.^2)).^(gamma/(gamma-1)));
q = (gamma/2).*P.*Mach.^2;
Tt = Cond(:,5)*0 + mean(Cond(:,5));
T = Tt./(1+(((gamma-1)/2)*Mach.^2));
rho = P./(287.*T);
V = Mach.*((gamma*287.*T).^0.5);

% Use this line for the Apollo model
% Resultant Moment and AoA data
time = Dat(:,1)-Dat(1,1);
M_volt = -Dat(:,2);
M_filt = -Dat(:,3);
AoA = 180-Dat(:,4);
AoA_filt = 180-Dat(:,5);

% % Use this line for the Expert model
% % Resultant Moment and AoA data
```

```
% time = Dat(:,1)-Dat(1,1);
% M_volt = Dat(:,2);
% M_filt = Dat(:,3);
% AoA = Dat(:,4);
% AoA_filt = Dat(:,5);

M = Anew*(M_volt-(interp1(Offset_Angle,Offset_Voltage,AoA,'cubic')));

for i = 1:size(q,1)
    q2((i-1)*(length(time)/length(q))+1:i*(length(time)/length(q))) = q(i);
end

q2 = q2';

if q(i) == 0
    Cm = M;
else
    Cm = M./(q2*S*D);
end



% This section is for M

[count3,value3] = hist(M,length(M));
[row3,col3] = find(count3 < 1);
cutoff3 = min(abs(value3(col3)));

Msmooth = M;
for i = 2:length(Msmooth)-1
    if abs(Msmooth(i)) > cutoff3
        Msmooth(i) = (Msmooth(i+1)+Msmooth(i-1))/2;
    end
end

Msmooth = func_smooth(Msmooth,4,'sgolay',3);
for i = 1:5
    Msmooth = func_smooth(Msmooth,i*40,'sgolay',3);
end



% This section is for Cm

[count,value] = hist(Cm,length(Cm));
[row,col] = find(count < 1);
cutoff = min(abs(value(col)));

Cmsmooth = Cm;
for i = 2:length(Cmsmooth)-1
    if abs(Cmsmooth(i)) > cutoff
        Cmsmooth(i) = (Cmsmooth(i+1)+Cmsmooth(i-1))/2;
    end
end

Cmsmooth = func_smooth(Cmsmooth,4,'sgolay',3);
for i = 1:5
    Cmsmooth = func_smooth(Cmsmooth,i*40,'sgolay',3);
end
```

```
% This section is for AoA

[count2,value2] = hist(AoA,length(AoA));
[row2,col2] = find(count2 < 1);
cutoff2 = min(abs(value2(col2)));

AoAsmooth = AoA;
for i = 2:length(AoAsmooth)-1
    if abs(AoAsmooth(i)) > cutoff2
        AoAsmooth(i) = (AoAsmooth(i+1)+AoAsmooth(i-1))/2;
    end
end

AoAsmooth = func_smooth(AoAsmooth,4,'sgolay',3);
for i = 1:5
    AoAsmooth = func_smooth(AoAsmooth,i*40,'sgolay',3);
end


[AoAlock,Cmlock,period,freq] = func_phase_lock(time,AoAsmooth,Cmsmooth);
[AoAlock2,Mlock,period2,freq2] = func_phase_lock(time,AoAsmooth,Msmooth);
% [Cmlock_NO,AoAlock_NO,period_Cm,freq_Cm] =
func_phase_lock(time,Cmsmooth,AoAsmooth);
% [Mlock_NO,AoAlock_NO,period_M,freq_M] =
func_phase_lock(time,Msmooth,AoAsmooth);

AoAlock = AoAlock(1:length(Cmlock));
Mlock = Mlock(1:length(Cmlock));
timelock = time(1:length(Cmlock));

AoAlock = AoAlock+(mean(AoAsmooth)-mean(AoAlock));
```

func_phase_lock.m

```
% Perodic Phase-Locking Averaging
%
% [signal_locked,period,frequency] = func_phase_lock(time,signal)
%
% Returns the dominant frequency of the signal and then calculates an
% averaged signal over one period based on the frequency calculated for
% the entire length of the signal duration.

function [signal1_locked,signal2_locked,period,frequency] =
func_phase_lock(time,signal1,signal2)

% function [signal_locked,period,frequency] =
func_phase_lock(time,signal1,signal2preserve)

y = fft(signal1);
Y = y.*conj(y);

row = find(Y == max(Y(2:end)));

no_waves = row(1)-1;
time_index = round(length(time)/(no_waves+1));

checkmin = find(min(signal1(1:time_index)) == signal1);
checkmax = find(max(signal1(1:time_index)) == signal1);
```

```
if checkmin > checkmax
    index_min(1) = checkmin;
    time_index_new(1) = time_index;
    for i = 1:no_waves-2
        index_max(i) =
find(max(signal1(index_min(i):index_min(i)+time_index_new(i))) == signal1);
        if index_max(i)+time_index_new(i) > length(signal1)
            index_min(i+1) = find(min(signal1(index_max(i):length(signal1)))
== signal1);
        else
            index_min(i+1) =
find(min(signal1(index_max(i):index_max(i)+time_index_new(i))) == signal1);
        end
        time_index_new(i+1) = index_min(i+1)-index_min(i);
    end
else
    index_max(1) = checkmax;
    time_index_new(1) = time_index;
    for i = 1:no_waves-2
        index_min(i) =
find(min(signal1(index_max(i):index_max(i)+time_index_new(i))) == signal1);
        if index_min(i)+time_index_new(i) > length(signal1)
            index_max(i+1) = find(max(signal1(index_min(i):length(signal1)))
== signal1);
        else
            index_max(i+1) =
find(max(signal1(index_min(i):index_min(i)+time_index_new(i))) == signal1);
        end
        time_index_new(i+1) = index_max(i+1)-index_max(i);
    end
end

for i = 1:length(index_min)-1
    valley(i) = index_min(i+1)-index_min(i);
end
for i = 1:length(index_max)-1
    peak(i) = index_max(i+1)-index_max(i);
end

true_index = (mean(valley)+mean(peak))/2;
period = interp1(1:length(time),time,true_index,'cubic'); %Global period
frequency = 1/period;                                     %Global freq
initial_time_index = time_index;
time_index = round(true_index);

dt = time(2)-time(1);
if checkmax > checkmin
    for i = 1:length(time_index_new)-1
        for j = index_max(i):index_max(i+1)-1
            signalindex(j-index_max(i)+1) = ((time_index_new(i+1)-
time_index)/time_index_new(i+1))*(j-index_max(i))+j;
        end
        if length(signalindex) < time_index
            for k = 1:(time_index-length(signalindex))
                signalindex(end+1) =
interp1(1:length(signalindex),signalindex,length(signalindex)+1,'cubic');
            end
        end
        index(1+(i-1)*time_index:i*time_index) = signalindex(1:time_index);
```

```
        clear signalindex;
    end
    timenew = time(checkmax:checkmax+length(index)-1);
else
    for i = 1:length(time_index_new)-1
        for j = index_min(i):index_min(i+1)-1
            signalindex(j-index_min(i)+1) = ((time_index_new(i+1)-
time_index)/time_index_new(i+1))*(j-index_min(i))+j;
        end
        if length(signalindex) < time_index
            for k = 1:(time_index-length(signalindex))
                signalindex(end+1) =
interp1(1:length(signalindex),signalindex,length(signalindex)+1,'cubic');
            end
        end
        index(1+(i-1)*time_index:i*time_index) = signalindex(1:time_index);
        clear signalindex;
    end
    timenew = time(checkmin:checkmin+length(index)-1);
end

signal1new = interp1(time,signal1,dt*(index-1));
signal2new = interp1(time,signal2,dt*(index-1));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for j = 1:time_index
    signal1_locked(j) = signal1new(j);
    signal2_locked(j) = signal2new(j);
end
for i = 1:length(time_index_new)-2
    for j = 1:time_index
        signal1_locked(j) = signal1_locked(j) +
signal1new(j+((i)*time_index));
        signal2_locked(j) = signal2_locked(j) +
signal2new(j+((i)*time_index));
        verif1(i,j) = signal1new(j+((i)*time_index));
        verif2(i,j) = signal2new(j+((i)*time_index));
    end
end

for j = 1:time_index
    signal1_locked(j) = signal1_locked(j)/length(time_index_new);
    signal2_locked(j) = signal2_locked(j)/length(time_index_new);
end
```

func_nth_derivative.m

```
% Calculates the derivative of one vector with respect to another vector
% with a 2nd order truncation error
%
% function [dny_dxn] = func_nth_derivative(x,y,order)
%
% Specify what order of derivative desired (nth order)
% Works well up to the 4th order

function [dny_dxn] = func_nth_derivative(x,y,n)

xfirst(1) = x(1) - n*(x(2)-x(1));
yfirst(1) = interp1(x,y,xfirst(1),'cubic','extrap');
xlast(1) = x(end) + (x(end)-x(end-1));
ylast(1) = interp1(x,y,xlast(1),'cubic','extrap');
```

```
if n > 1
    for i = 2:n
        xfirst(i) = xfirst(i-1) + (x(2)-x(1));
        yfirst(i) = interp1(x,y,xfirst(i),'cubic','extrap');
        xlast(i) = xlast(i-1) + (x(end)-x(end-1));
        ylast(i) = interp1(x,y,xlast(i),'cubic','extrap');
    end
end

if size(x,1) == 1
    xnew = [xfirst x xlast];
    xnew = xnew';
elseif size(x,2) == 1
    xnew = [xfirst; x; xlast];
else
    disp('x variable is not a vector and needs to be one');
end

if size(y,1) == 1
    ynew = [yfirst y ylast];
    ynew = ynew';
elseif size(y,2) == 1
    ynew = [yfirst; y; ylast];
else
    disp('y variable is not a vector and needs to be one');
end

const = diag(sort(pascal(n+1),'descend'));

k = 1;
for i = n+1:length(xnew)-n
    dx(i) = (xnew(i+1)-xnew(i-1))/2;
    middle = 0;
    if n > 1
        for j = 1:n-1
            middle = middle + (-1)^j*ynew(i+n-2*j)*const(j+1);
        end
    end
    dny_dxn(k) = (ynew(i+n) + middle + (-1)^n*ynew(i-n)) / ((2*dx(i))^n);
    k = k+1;
end

if n >= 5
    dny_dxn = func_smooth(dny_dxn,floor(0.05*length(x)),'sgolay',3);
end
```

func_forcing_moment.m

```
% Defines the theoretical forcing function based on forced oscillation
% set-up in the VKI S1 Wind Tunnel
%
% Notation:
% function [M,alpha,alpha_dot,alpha_dbldot] =
% func_forcing_moment(I,Oscillation_Amplitude,Frequency)
%
% Function INPUTS:
%
% I = System moment of inertia about the axis of rotation [kg-m^2]
% Oscillation_Amplitude = The maximum amplitude of forced oscillation [deg]
```

```
% Frequency = The frequency of forced oscillation [Hz]
%
% Function OUTPUTS:
% M = The theoretical moment forcing function [N-m]
% t = Time over one full period of oscillation [sec]
% alpha = Oscillation angle {not angle of attack} [deg]
% alpha_dot = Derivative of alpha w.r.t. time [rad/s]
% alpha_dbldot = 2nd derivative of alpha w.r.t. time [rad/s^2]

function [M,t,alpha,alpha_dot,alpha_dbldot] =
func_forcing_moment(I,Oscillation_Amplitude,Frequency)

r = 0;                  %[mm]
alpha = 0;              %[deg]

while (max(alpha)-min(alpha))/2 < Oscillation_Amplitude
r = r + 0.1;            %[mm]
R = 80.45;              %[mm]
L = 112;                %[mm]
s = 8;                  %[mm]
x0 = s - R;             %[mm]
y0 = sqrt(L^2 - s^2);   %[mm]

% Maximum moment that the torque balcance can withstand
T = 1/Frequency;             %[s]
om = 2*pi*Frequency;         %[rad/s]
t = 0:0.005*T:T;          %[s]
theta = om*t;           %[rad]
x1 = r*cos(theta);
y1 = r*sin(theta);

val = 0.0001;           % Iteration precision
R2 = R + 10;
x2 = s + val;

for i = 1:length(theta)
    while R2-R > 0
        x2 = x2 - val;
        y2 = (2*y1(i) + sqrt((2*y1(i))^2 - 4*(y1(i)^2 + (x2-x1(i))^2 -
L^2)))/2;
        R2 = sqrt((x2-x0)^2 + (y2-y0)^2);
    end
    x_2(i) = x2;
    y_2(i) = y2;
    R2 = 100;
    x2 = s + val;
end

for i = 1:length(x_2)
    alpha(i) = atan((y_2(i)-y0)/(x_2(i)-x0));
end
alpha_dot = func_nth_derivative(t,alpha,1);
alpha_dbldot = func_nth_derivative(t,alpha_dot,1);

for i = 1:length(alpha)
    M(i) = I*alpha_dbldot(i);
end
alpha = alpha*180/pi;
end
```