GUST LOAD ALLEVIATION OF AN AEROELASTIC SYSTEM

USING NONLINEAR CONTROL

A Thesis

by

AMY MARIE LUCAS

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

August 2009

Major Subject: Aerospace Engineering

GUST LOAD ALLEVIATION OF AN AEROELASTIC SYSTEM

USING NONLINEAR CONTROL

A Thesis

by

AMY MARIE LUCAS

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Approved by:

Chair of Committee,   Thomas W. Strganac
Committee Members,   John Valasek
                                       Alan Palazzolo
Head of Department,   Dimitris Lagoudas

August 2009

Major Subject: Aerospace Engineering

ABSTRACT

Gust Load Alleviation of an Aeroelastic System

Using Nonlinear Control. (August 2009)

Amy Marie Lucas, B.S., Lipscomb University

Chair of Advisory Committee: Dr. Thomas W. Strganac

The author develops a nonlinear longitudinal model of an aircraft modeled by rigid fuselage, tail, and wing, where the wing is attached to the fuselage with a torsional spring. The main focus of this research is to retain the full nonlinearities associated with the system and to perform gust load alleviation for the model by comparing the impact of a proportional-integral-filter nonzero setpoint linear controller with control rate weighting and a nonlinear Lyapunov-based controller. The four degree of freedom longitudinal system under consideration includes the traditional longitudinal three degree of freedom aircraft model and one additional degree of freedom due to the torsion from the wing attachment. Computational simulations are performed to show the aeroelastic response of the aircraft due to a gust load disturbance with and without control. Results presented in this thesis show that the linear model fails to capture the true nonlinear response of the system and the linear controller based on the linear model does not stabilize the nonlinear system. The results from the Lyapunov-based control demonstrate the ability to stabilize the nonlinear response, including the presence of an LCO, and emphasize the importance of examining the fully nonlinear system with a nonlinear controller.

To my husband and best friend, Matt.

"I sought the Lord, and he answered me, and delivered me from all my fears."

Psalm 34:4

ACKNOWLEDGMENTS

The work that is presented in this thesis would not have been possible without the guidance and help of many people. First, I would like to thank my advisor, Dr. Thomas Strganac for his guidance and endless patience. I appreciate all the nudgings in the right direction and support along the way.

I would also like to thank my committee members. Dr. Palazzolo, for agreeing to be on my committee. Dr. Valasek, for his immense knowledge of control systems, his willingness to answer my questions, and for the use of his Matlab codes, *qpmcalc.m* and *lqrdjv.m*, which are used to calculate the inverse of the QPM matrix and the LQR gains for the linear controller. I am also very thankful for the guidance of Dr. Hurtado for lending his ear when I had a dynamics problem and for the use of his Matlab code for a constant time step Runga-Kutta solver, *runge.m*. Especially, I would like to thank NASA Graduate Student Researchers Program for helping fund my studies and research this past year. I appreciate Dr. Walt Silva's insight into my research problem and his suggestion to look into gust load alleviation and a nonlinear controller.

I am grateful to my family for their neverending love and support. I also owe many thanks to some of my colleagues and friends. Thanks to Julie Parish and Leslie Weitz for their dynamical insight and willingness to listen, with special thanks to Leslie Weitz for her help with the Lyapunov function. I am grateful to Matt Fritz for sharing some of his Matlab debugging knowledge. Also, I would like to thank Myounghee Kim, for always being there to listen, and for her insight into potential gain choices for the Lyapunov controller.

And last, but not least, I would like to thank my husband, Matt. For his patience and support, without which I would not have made it this far.

NOMENCLATURE

$\hat{b}$ Fuselage fixed reference frame unit vector

$\hat{s_t}$ Tail stability axes fixed reference frame unit vector

$\hat{s_w}$ Wing stability axes fixed reference frame unit vector

$\hat{t}$ Tail fixed reference frame unit vector

$\hat{w}$ Wing fixed reference frame unit vector

AC Aerodynamic center

CM Center of mass

CRW Control rate weighting

d Distance between CM_f and wing hinge in $\hat{b}_1$ direction

d_c Distance between wing hinge and CM_f in $\hat{w}_1$ direction

D_t Drag at tail quarter chord

D_w Drag at wing quarter chord

e Distance between elastic axis and aerodynamic center of wing in $\hat{w}_1$ direction

f Distance between tail support and CM_f in $\hat{b}_1$ direction

g Acceleration of gravity

h Altitude

H_{CM} Angular momentum of a body about its CM

I_f Moment of inertia of fuselage about CM_f

I_w Moment of inertia of wing about CM_w

k Wing stiffness

L_t Lift at tail quarter chord

L_w Lift at wing quarter chord

m Mass of vehicle, $m_f + m_w$

M_{ac} Moment at aerodynamic center

m_f Mass of fuselage

m_w Mass of wing

$NZSP$ Nonzero setpoint

PIF Proportional-integral-filter

QPM Quad partition matrix

S_t Tail area

S_w Wing area

T Thrust

u Velocity of system CM in $\hat{b}_1$ direction

u_d Desired trim velocity of system CM in $\hat{b}_1$ direction

V Freestream velocity of aircraft

w Velocity of system CM in $\hat{b}_3$ direction

α_r Initial wing displacement angle with zero spring displacement

α_t Angle of attack of tail, $\alpha_w - \epsilon - \alpha_r$

α_w Angle of attack of wing, $\phi + \theta - \gamma_w$

δ_e Deflection of elevator

δ_f Deflection of flaperon

ϵ downwash, $\frac{d\epsilon}{d\alpha}\alpha_w + \epsilon_0$

γ_t Flight path angle for tail surface

γ_w Flight path angle for wing surface

ω Frequency

ω_w Wing natural frequency

ϕ Wing displacement relative to the fuselage fixed reference frame

ρ Density of air

τ_{δ_e} Elevator effectiveness parameter

τ_{δ_f} Flaperon effectiveness parameter

θ Vehicle pitch attitude relative to horizontal

$\dot{}$ First time derivative

$\ddot{}$ Second time derivative

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Aeroelasticity is the study of the interaction between inertial, structural, and aerodynamic forces. This area of study is complex due to the coupling nature of these different forces. Additional complexity is added to due the inherent nonlinearities in the equations of motion for the system. All real physical responses are nonlinear and the structure-aerodynamic systems have an infinite number of degrees of freedom (DOF). Usually, simplifications of the system are made that may allow for a linearized model and/or a finite number of DOF. The system of interest in this paper is the longitudinal model of an aircraft with a flexible wing. This system is a finite DOF problem, and the nonlinearities of the system are retained. If the aeroelastic nonlinear response of an aircraft is more fully understood, then it may be possible to have a more optimally designed wing that minimizes weight by of the structure and, in turn, reduces fuel consumption and improves performance.

Nonlinearities in the equations of motion for an aeroelastic system have been of much interest in recent years. These nonlinearities have been shown to lead to Limit Cycle Oscillations (LCOs) and other nonlinear aeroelastic responses[1]. Wing flexibility in particular is inherently nonlinear and may greatly affect the aeroelastic response. Patil, Hodges, and Cesnik[2] showed that wing flexibility affected both the trim solution and values obtained for the short-period and phugoid modes. Areas of interest for aeroelasticity are multifaceted, incorporating structural modeling, aerodynamic modeling, control approaches, and estimation of unknown parameters. Of

—————

This thesis follows the style of *IEEE Transactions on Automatic Control.*

particular interests to the author, and discussed in this paper, are the determination of the importance of nonlinearities associated with a torsionally flexible wing on a longitudinal aircraft model and a viable method to control the aircraft during gust loading. For the following work, estimation of parameters is not included.

An example, demonstrating the need for further investigation of nonlinear aeroelastic systems, is the loss of the NASA Helios aircraft, a powered uninhabited flying wing with a wingspan of approximately 247 feet. The Helios encountered normal turbulence which caused a high dihedral angle for the wing which transformed into growing pitching oscillations, and resulted in the loss of the aircraft. One of the recommendations for future work was noted to be a need to "develop multidisciplinary (structures, aerodynamic, controls, etc) models, which may describe the nonlinear dynamic behavior of aircraft modifications"[3]. It is important to note that it was not extreme unusual disturbances that caused the failure of the Helios; rather there was a lack of understanding of the aeroelastic model of the flexible structure.

Shearer and Cesnik[4] investigated nonlinear flight dynamics for a very flexible aircraft. They determined the 6 DOF equations of motion for an aircraft coupled with the aeroelastic equations that govern the geometrically nonlinear structural response of the vehicle. Their analysis focused on analyzing the differences between rigid body, linearized aeroelastic, and nonlinear aeroelastic aircraft dynamic responses for various flight maneuvers and aircraft conditions. They found that for simple symmetric maneuvers, the results indicated that a linearized solution gave acceptable results. However, for a heavy weight asymmetric maneuver, results showed significant differences in the responses. They concluded that a nonlinear analysis approach is required to properly capture the vehicle response for asymmetric maneuvering of a flexible aircraft. Though the current research is a rigid wing with a torsionally flexible attachment, this approach provides a simplified model of a continuous flexible wing

undergoing torsion. Similar to Shearer and Cesnik's[4] work, a comparison of the linear and nonlinear response of the vehicle is investigated in this paper.

Numerous studies have been conducted concerning gust load alleviation (GLA) for various types of aircraft. Gust loadings disturb the system and may excite the system, leading to a nonlinear aeroelastic response. Increased flexibility leads to wing structure and vehicle rigid body modes being in the same frequency range, allowing the nonlinear structural response to have a greater impact on the system as a whole. Vartio et al.[5] documented the results of a wind tunnel test at NASA Langley in the Transonic Dynamics Tunnel of a gust load alleviation system for the Sensorcraft concept. Through the experiment, they demonstrated that a leading edge/outboard trailing edge control scheme may successfully control the first bending and pitch modes for a SensorCraft concept. Matsuzaki et al.[6] both analytically and experimentally demonstrated that a control system using a feedback filter is effective for both sinusoidal gust loadings and atmospheric turbulence for a transport-type wing.

Penning et al.[7] investigated GLA and flutter suppression for a Sensorcraft concept using system identification. They included flexible dynamics of wing bending for a high aspect ratio blended wing-body tailless aircraft and found that deriving control laws based on the combination system identification technology and flight control law development achieves GLA and flutter suppression. They verified these results experimentally. The experiment included wind tunnel tests, conducted at NASA Langley TDT, on a half span aeroservoelastic model with allowed pitch and plunge DOF. The results showed that their control laws were able to reduce peak wing bending moments by more than 50% in many cases, and that body freedom flutter was suppressed at speeds greater than 20% above the flutter speed.

The current research is building on the previous work in the field of aeroelasticity.

Block and Strganac[8] showed the effectiveness of multivariable feedback control on a wing section with nonlinear stiffness. They showed that although the linear controller suppressed the LCOs, linear control was not robust enough to be a global controller. More recently, work by Platanitis and Strganac in adaptive control[9] on the nonlinear aeroelastic test apparatus (NATA) with a leading- and trailing-edge control effectors was performed. In this study, an experiment on NATA was performed in the 2' by 3' wind tunnel at Texas A&M University. It was determined that, through simulation and experimentation, a globally stabilizing control law may be achieved by only using the leading- and trailing-edge control effectors for the 2 DOF system. The interest in controlling nonlinear aeroelastic responses of a 2 DOF model led to interest in the nonlinear control of a 4 DOF nonlinear model of an aircraft with a torsionally flexible wing.

Variations of optimal nonzero setpoint (NZSP) controllers have been used for many different aircraft applications. Stauffer et al.[10] designed a proportional-integral-filter controller using nonzero setpoint formulation (PIF-NZSP) for integrated control of a vertical take-off and landing (VTOL) airplane in hover. Valasek et al.[11] and Kimmett et al.[12] investigated an optimal nonzero setpoint control rate weighting (NZSP-CRW) control structure to track and dock with a stationery drogue for in-flight aerial refueling with turbulence. Extending this work, Valasek et al.[13] derived and implemented a proportional-integral-filter optimal nonzero setpoint controller with control rate weighting (PIF-NZSP-CRW) to aid in their vision-based sensor and navigation system for autonomous air refueling. The linear controller used in this research for comparison with the nonlinear controller is based on the same PIF-NZSP-CRW method derived by Valasek et al.[13].

Specific interest lies in the implementation of a nonlinear Lyapunov-based controller. The Lyapunov approach has been used to control a vast number of nonlinear

systems, and has been found to work particularly well with flexible structures. de Paiva et al.[14] investigated three nonlinear control methods to develop a global nonlinear controller for an autonomous unmanned airship. They found that the backstepping technique, a Lyapunov-based nonlinear control, was a robust tool that enabled them to simulate a representative flight test of a complete airship mission, including path tracking under turbulent wind conditions. Also, Suk et al.[15] used Lyapunov control theory and time-domain finite element analysis to control a slew maneuver for a flexible space structure. Bhatta and Leonard[16] also implemented Lyapunov based control for a longitudinal underwater glider model. They used the phugoid-mode model, a Hamiltonian model of longitudinal dynamics originally studied by Lanchester[17, 18], as the foundation for their composite Lyapunov function and proved exponential stability for the model with buoyancy and elevator controls.

The main contribution of this research is to develop a nonlinear 4 DOF longitudinal aircraft model, use the model to determine the adverse consequences produced by linearizing the equations of motion for an aircraft, and develop nonlinear Lyapunov-based control laws to perform GLA for the nonlinear system. The research demonstrates the importance of using both a nonlinear model and nonlinear control method. Results show that the linear model fails to capture the true nonlinear response of the system and the linear controller does not stabilize the nonlinear system. The equations of motion are general, allowing one to consider models of various other types of aircraft.

This thesis is organized in the following manner. First the 4 DOF longitudinal model and important reference frames are presented. Second, the equations of motion are developed using both Newton-Euler methods and Gibbs-Appell methods to verify the derivation. Third, the aerodynamic model, gust load model, and control effectors used are discussed. A section is included to explain the simulation algorithm. Next,

outlines for the development and implementation of both the linear PIF-NZSP-CRW controller and the nonlinear Lyapunov-based controller, respectively, are introduced. Then, the results are discussed and a comparison between the effectiveness of the linear and nonlinear controllers are given, elaborating on their respective advantages and disadvantages. Lastly, future areas of research are presented.

CHAPTER II

4 DOF LONGITUDINAL MODEL

The foundation for the current work is the development of a 4 DOF analytical model of a vehicle in trimmed flight with a rigid wing mounted on a flexible support. This model is useful since it is a simplified representation of a fully flexible wing. This base model simulates motion along a flight path, a flight path change, a change in pitch attitude, and an additional degree of freedom due to the torsion from the wing attachment. The degrees of freedom are the velocities of the center of mass (CM) of the system within a plane, u and w respectively, the change in pitch attitude, θ, and a degree of freedom associated with the torsion of the wing, ϕ. The simplified representation including reference frame locations is shown in Fig. 1.
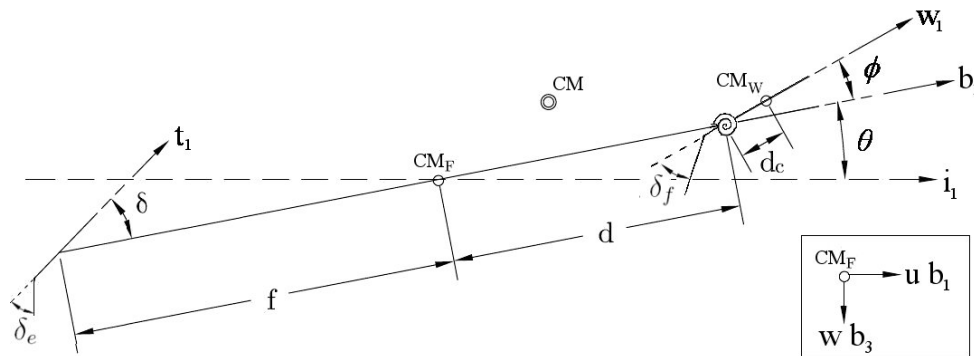
Fig. 1. Simplified Representation

Three degrees of freedom capture classical longitudinal rigid body motion. The fourth

degree of freedom is associated with a rigid wing attached to the vehicle with a flexible support. A simple quasi-steady aerodynamic model is used. Nonlinear terms are retained which permit the onset of an LCO at specific flight conditions. The derivation of the 4DOF equations of motion was performed using both Newton-Euler Method and Gibbs-Appell Method to check validity of the development.

For the derivation of the equations of motion reference frames are defined. The body reference frames are the body axis frame, the wing axis frame, and the tail axis frame.
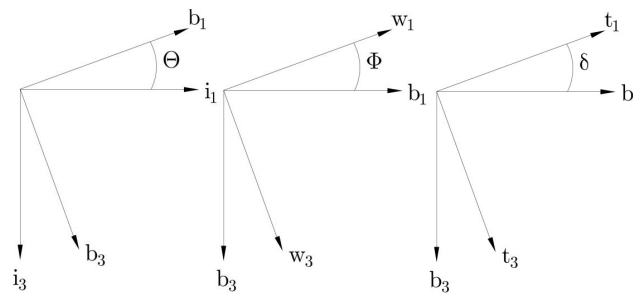
Fig. 2. Body Reference Frames: Body (b), Wing (w), Tail (t)

In addition to the body reference frames (Fig. 2), stability frames (Fig. 3) are needed for lift and drag moment calculations. The lift and drag of the vehicle is divided into the lift and drag for the wing and tail respectively.

The angle of attack is defined as the angle between the wing body frame axes
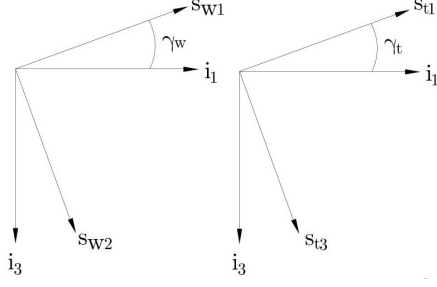
Fig. 3. Stability Frames: Wing (w), Tail (t)

and the local velocity vector. The equations for the wing and tail angle of attack are

$$\alpha_w = \theta + \phi - \gamma_w \tag{2.1}$$

$$\alpha_t = \alpha_w - \epsilon - \alpha_r \tag{2.2}$$

where $\epsilon = \epsilon_0 + \alpha_w \frac{d\epsilon}{d\alpha}$ is the downwash.

A.   Development of 4 DOF Model

The Gibbs-Appell Method and the Newton-Euler Method are used to derive and confirm the equations of motion for the system. Lagrange's equations are not used because some of the states are quasi-velocities, i.e. functions of some of the generalized coordinates. In the current system, two of the states are the velocity of the CG for the system in the body frame, which are functions of the inertial velocity of the CM for the system, and the fuselage pitch rate. Quasi-velocities are used to compare with the same quasi-velocities used in published results[19, 20].

1.   Gibbs-Appell Method

The Gibbs-Appell method uses a technique similar to Lagrange's equations. In Lagrange's equations a scalar function in velocity terms, i.e. the kinetic energy, is used

to derive the equations of motion. For the Gibbs-Appell methods, a scalar function in acceleration terms, known as the energy of acceleration, is used. The energy of acceleration equation for the system is

$$\mathbf{S} = \frac{1}{2}\left(m\mathbf{a}_{cg/i} \boldsymbol{.} \mathbf{a}_{cg/i} + \frac{1}{2}\left(\boldsymbol{\alpha} \boldsymbol{.} \dot{\mathbf{H}}_{cg}\right) + \boldsymbol{\alpha} \boldsymbol{.} \left(\boldsymbol{\omega} \times \mathbf{H}_{cg}\right)\right) \tag{2.3}$$

where the slash notation $'a\diagup b'$ denotes the vector from b to a, and $\boldsymbol{\alpha}$ is the angular acceleration. Also, $\boldsymbol{\omega}_{cg/i} = \dot{\theta}\hat{b}_2$ and $\mathbf{H}_{cg} = I_{cg}\boldsymbol{\omega}_{cg/i}$; therefore, $(\boldsymbol{\omega} \times \mathbf{H}_{cg}) = 0$.

To obtain the equations of motion, one considers

$$\frac{\partial \mathbf{S}}{\partial \dot{u}_i} = U_i \tag{2.4}$$

where $i = 1 : m$ with 'm' equaling the total number of generalized coordinates, i.e. there are 'm' second order equations of motion. Also, u_i represents the generalized coordinates, using the quasi-velocities, and U_i represents the generalized forces of the system. The generalized forces are determined using the virtual work equation,

$$\begin{aligned}\overline{\delta W} &= \sum \mathbf{F}_j \boldsymbol{.} \partial \mathbf{p}_j + \sum \mathbf{M}_k \partial \alpha_k \\ &= \sum U_i \, \partial \mathbf{u}_i\end{aligned} \tag{2.5}$$

In the above equation, α_k represents the angle of rotation for the body, not to be confused with angle of attack or angular acceleration. Also, the notation uses the subscript $j = 1 : n$ where 'n' equals the number of forces acting on the system and $k = 1 : o$ where 'o' equals the number of moments acting on the system. Note that the $\partial \mathbf{p}_j$ term is derived from the inertial velocity vectors of the forces. Time derivatives in the velocity vector, such as $\dot{\theta}$, are replaced with the respective partial, such as $\partial \theta$. For full details of derivation, see Appendix A.

2. Newton-Euler Method

Newton-Euler derivation of the equations of motion uses Newton's Second Law for the derivation of the translational equations of motion and Euler's equations for a rotating body for the derivation of the rotational equations of motion.

Newton's Second law states

$$\sum \mathbf{F} = m\mathbf{a}_{CM_s/i} \tag{2.6}$$

where $\mathbf{F}$ represents the forces on the system. To develop the rotational equations of motion, Euler's equations for a rotational body about an arbitrary point, '$*$', are used[21].

$$\mathbf{H}_* = \mathbf{H}_{CM} + m\left(\mathbf{r}_{CM/i} - \mathbf{r}_{*/i}\right) \times \left(\mathbf{v}_{CM/i} - \mathbf{v}_{*/i}\right) \tag{2.7}$$

$$\dot{\mathbf{H}}_* = \mathbf{l}_* + m\mathbf{a}_{*/i} \times \left(\mathbf{r}_{CM/i} - \mathbf{r}_{*/i}\right) \tag{2.8}$$

where, $\mathbf{H}$ is the angular momentum, CM is the center of mass for each individual body, $\mathbf{r}$ is the inertial position, $\mathbf{v}$ is the inertial velocity, and $\mathbf{l}$ is the sum of moments on the system. The total angular momentum for the full system is determined by adding the contributions from wing and fuselage angular momentums about the system CM.

$$\mathbf{H}_{CM_s} = \mathbf{H}_{CM_w/CM_s} + \mathbf{H}_{CM_f/CM_s} \tag{2.9}$$

Similarly,

$$\dot{\mathbf{H}}_{CM_s} = \dot{\mathbf{H}}_{CM_w/CM_s} + \dot{\mathbf{H}}_{CM_f/CM_s} \tag{2.10}$$

For full details of derivation, see Appendix B.

3. Gust Load

A vertical gust load is imposed on the system and acts as a perturbation of the system. Depending on the initial velocity and spring stiffness for the wing attachment, the perturbation could cause a nonlinear aeroelastic response, such as an LCO.

The gust load is imposed in the manner of an additional lift on the wing surface. Some assumptions made are that the wing is the only surface that encounters the gust and that the gust impacts the total surface of the wing instantaneously.

$$L_{vg} = QS_wC_{L_\alpha}\frac{V_{vg}}{V} \tag{2.11}$$

and

$$L_{hg} = \frac{\rho}{2}\left(2V_{hg}V + V_{hg}^2\right)S_wC_L \tag{2.12}$$

where V_{vg} and V_{hg} are the velocities of a horizontal and vertical gust respectively, and V is the velocity of the aircraft. The lift due to the horizontal gust acts at the CM for the wing, and the lift due to the vertical gust acts at the aerodynamic center[22]. This study will only implement a vertical gust, but the equations are written in a manner that would allow for both a vertical and a horizontal gust. The velocity of the gust is modeled by a one-minus-cosine idealization.

$$V_g = \frac{1}{2}V_{max}\left(1 - \cos\frac{2\pi t}{2dm}\right) \tag{2.13}$$

where V_{max} is the total maximum velocity for the gust and dm is the time to reach the maximum gust velocity. See Fig. 4 for a diagram depicting the one-minus cosine
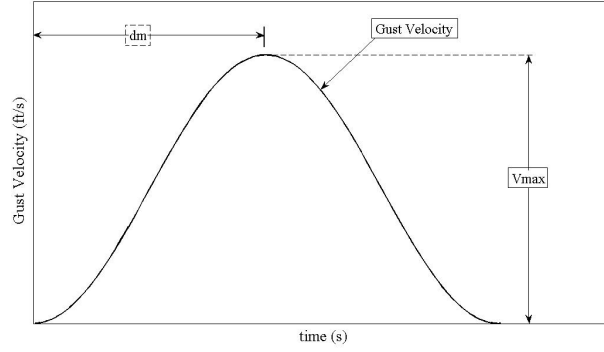
Fig. 4. One-minus-cosine Gust Idealization

idealization for the gust loading.

Thus, using Eqs. (2.3), (2.4), and (2.5) and including the gust load, the following equations of motion are constructed

Translational Equation of Motion (in the $\hat{b}_1$ frame):

$$m\left(\dot{u} + w\dot{\theta}\right) = -mg\sin\theta - D_w\cos\left(\theta - \gamma_w\right) + \left(L_w + L_{vg} + L_g\right)\sin\left(\theta - \gamma_w\right)$$
$$+ L_t\sin\left(\theta - \gamma_t\right) - D_t\cos\left(\theta - \gamma_t\right) + T \tag{2.14}$$

Translational Equation of Motion (in the $\hat{b}_3$ frame):

$$m\left(\dot{w} - u\dot{\theta}\right) = mg\cos\theta - D_w\sin\left(\theta - \gamma_w\right) - \left(L_w + L_{vg} + L_{hg}\right)\cos\left(\theta - \gamma_w\right)$$
$$- L_t\cos\left(\theta - \gamma_t\right) - D_t\sin\left(\theta - \gamma_t\right) \tag{2.15}$$

System Rotational Equation of Motion:

$$I_f\ddot{\theta} + \frac{m_w m_f}{m}\left(d_c^2\left(\ddot{\theta}+\ddot{\phi}\right) - dd_c\left(2\dot{\theta}\dot{\phi}+\dot{\phi}^2\right)\sin\phi + d^2\ddot{\theta} + dd_c\left(2\ddot{\theta}+\ddot{\phi}\right)\cos\phi\right)$$
$$+ I_w\left(\ddot{\theta}+\ddot{\phi}\right) = (L_w + L_{vg})\left(\left(e - \frac{m_w}{m}d_c\right)\cos\alpha_w + \frac{m_f}{m}d\cos\left(\theta-\gamma_w\right)\right)$$
$$+ L_{hg}\frac{m_f}{m}\left(d_c\cos\alpha_w + d\cos\left(\theta-\gamma_w\right)\right)$$
$$+ D_w\left(\left(e - \frac{m_w}{m}d_c\right)\sin\alpha_w + \frac{m_f}{m}d\sin\left(\theta-\gamma_w\right)\right)$$
$$- L_t\left(\frac{m_w}{m}d_c\cos\alpha_t + \left(\frac{m_w}{m}d + f\right)\cos\left(\theta-\gamma_t\right)\right)$$
$$- D_t\left(\frac{m_w}{m}d_c\sin\alpha_t + \left(\frac{m_w}{m}d + f\right)\sin\left(\theta-\gamma_t\right)\right)$$
$$+ T\frac{m_w}{m}d_c\sin\phi + M_{ac}$$

$$(2.16)$$

Wing Rotational Equation of Motion:

$$I_w\left(\ddot{\theta}+\ddot{\phi}\right) + \frac{m_w m_f}{m}\left(dd_c\ddot{\theta}\cos\phi + d_c^2\left(\ddot{\theta}+\ddot{\phi}\right) + dd_c\dot{\theta}^2\sin\phi\right) =$$
$$- k\left(\phi-\alpha_r\right) + (L_w + L_{vg})\left(e - \frac{m_w}{m}d_c\right)\cos\alpha_w + L_{hg}\frac{m_f}{m}d_c\cos\alpha_w$$
$$+ D_w\left(e - \frac{m_w}{m}d_c\right)\sin\alpha_w - L_t\frac{m_w}{m}d_c\cos\alpha_t - D_t\frac{m_w}{m}d_c\sin\alpha_t$$
$$+ T\frac{m_w}{m}d_c\sin\phi + M_{ac}$$

$$(2.17)$$

The equations contain nonlinear rotation rate terms that are caused by the rotation of the fuselage and wing components about their respective mass centers. Additionally, there are nonlinear terms that arise from the lift and drag terms. Note M_{ac} is the zero-lift aerodynamic moment, if it exists. One component of the research is to perform simulations using the nonlinear equations of motion to determine the nonlinear aeroelastic response of the vehicle.

The equations of motion for the 4 DOF model have been verified using published data for a 3 DOF case. The airplane data used to validate the current model is the Cessna 182, mainly chosen for the availability of published data[20, 23] for

comparisons. Published stability and control derivatives are used to determine the unknown required input values, such as length and mass ratios. Note, the set of input values used in the following analysis does not lead to a unique solution, due in part to the nondimensional nature of the stability and control derivatives, as well as nonlinearities.

4. Control Effectors

The longitudinal aircraft model uses two control effectors, an elevator and a flaperon. The elevator is located on the horizontal tail and the flaperon is located on the wing. Both are trailing edge control effectors and are constrained to deflect symmetrically in such a way as to not impose a roll moment on the aircraft. Limitations on the elevator restrict movement to $+28°$ and $-21°$ [23]. The flaperon control effector is added to examine alleviation of the wing oscillations from the torsional spring attachment. Since this control effector has been introduced by the author, the limitations and size requirements must be defined. The limitations of the flaperon movement are chosen to be $+22°$ and $-15°$, which falls within the general guidelines limiting control effectors to $\pm30°$[24, 25].

The control effectors impact the lift/drag characteristics for the wing and tail. The equations for lift and drag of the wing and tail, including the terms originating from the controls, are

$$
\begin{aligned}
L_w &= Q_w S_w C_L \\
&= Q_w S_w \left(C_{L_0} + C_{L_\alpha} \alpha_w + C_{L\delta_f} \delta_f \right)
\end{aligned}
\tag{2.18}
$$

$$
\begin{aligned}
D_w &= Q_w S_w C_D \\
&= Q_w S_w \left(C_{D_0} + C_{D_\alpha} \alpha_w + C_{D\delta_f} \delta_f \right)
\end{aligned}
\tag{2.19}
$$

$$L_t = Q_t S_t C_{L_t}$$
$$= Q_t S_t \left(C_{L0_t} + C_{L\alpha_t}\alpha_t + C_{L\delta_e}\delta_e + C_{L\dot{\theta}}\dot{\theta} \right)$$

$$(2.20)$$

$$D_t = Q_t S_t C_{D_t}$$
$$= Q_t S_t \left(C_{D0_t} + C_{D\alpha_t}\alpha_t + C_{D\delta_e}\delta_e \right)$$

$$(2.21)$$

where Q is the dynamic pressure, S is the wing area, δ_f is the deflection angle of the flaperon, and δ_e is the deflection angle of the elevator. The subscripts t and w represent the tail and wing respectively. The term $C_{L\dot{\theta}}$ is the induced lift coefficient due to the induced angle of attack at the horizontal tail from the pitch rate, $\dot{\theta}$. When there is no canard contribution, i.e. a conventional aircraft, the wing contribution to $C_{L\dot{\theta}}$ is negligible but the horizontal tail contribution is important to include due to its large moment arm[20].

B. Special Cases

To further prove the validity of the equations of motion derived in this paper some special cases are discussed.

1. Case 1: Traditional Longitudinal Equations of Motion

The case where the CM of the wing is placed at the wing hinge attachment, i.e. $d_c = 0$ is examined. The system rotational equation of motion, Eq. (2.16), reduces to

$$
\begin{aligned}
\left(I_f + I_w + \frac{m_w m_f}{m} d^2\right) \ddot{\theta} + I_w \ddot{\phi} = \\
(L_w + L_{vg}) \left(e \cos \alpha_w + \frac{m_f}{m} d \cos (\theta - \gamma_w)\right) \\
+ L_{hg} \frac{m_f}{m} \left(d \cos (\theta - \gamma_w)\right) \\
+ D_w \left(e \sin \alpha_w + \frac{m_f}{m} d \sin (\theta - \gamma_w)\right) \\
- L_t \left(\frac{m_w}{m} d + f\right) \cos (\theta - \gamma_t) \\
- D_t \left(\frac{m_w}{m} d + f\right) \sin (\theta - \gamma_t) + M_{ac}
\end{aligned}
\tag{2.22}
$$

And, the wing rotational equation of motion, Eq. (2.17), reduces to

$$
I_w \left(\ddot{\theta} + \ddot{\phi}\right) = -k \left(\phi - \alpha_r\right) + (L_w + L_{vg}) e \cos \alpha_w + D_w e \sin \alpha_w + M_{ac}
\tag{2.23}
$$

Also, assume that the wing is rigidly attached and cannot rotate, i.e. $\ddot{\phi} = \dot{\phi} = \phi = 0$. Therefore, using the translational equations of motion and the system rotational equation of motion, the longitudinal pitch equations of motion for a rigid aircraft are obtained,

$$
m \left(\dot{u} + w\dot{\theta}\right) = -mg \sin \theta + X
\tag{2.24}
$$

$$
m \left(\dot{w} - u\dot{\theta}\right) = mg \cos \theta + Z
\tag{2.25}
$$

$$
I_y \ddot{\theta} = M
\tag{2.26}
$$

where X and Z are the aerodynamic forces in the body frame, M is the total moment of the system about its CM, and I_y is the total moment of inertia for the system about

the axis of rotation. Additionally the wing rotational equation of motion reduces to

$$I_w \left(\ddot{\theta} + \ddot{\phi} \right) = T \frac{m_w}{m} d_c \sin \phi + M_{ac} + \left(L_w + L_{vg} \right) e \cos \alpha_w + D_w e \sin \alpha_w \qquad (2.27)$$

which is the equation of motion for a rigid wing, rigidly attached.

2. Case 2: Equation of Motion for Wing Constrained by Torsional Spring

A second case to consider is where the wing rotational equation of motion, Eq. (2.17), reduced to the simple static case. Again assume the CM of the wing is placed at the wing hinge attachment, i.e. $d_c = 0$. And assume that the fuselage is not rotating, $\ddot{\theta} = \dot{\theta} = 0$. Using these assumptions, the system rotational equation of motion reduces to

$$\sum M_{CM} = 0 \qquad (2.28)$$

which makes since because sum of moments would equal zero if there is no rotation of the body. The translational equations of motion reduce to

$$m\dot{u} = -mg \sin \theta + X \qquad (2.29)$$

$$m\dot{w} = mg \cos \theta + Z \qquad (2.30)$$

And the wing rotational equation of motion reduces to

$$I_w \ddot{\phi} = -k \left(\phi - \alpha_r \right) + \left(L_w + L_{vg} \right) e \cos \alpha_w + D_w e \sin \alpha_w + M_{ac} \qquad (2.31)$$

which is the familiar static equation of motion for a wing section with pitch spring restraints[26].

C.   Simulations

The response of the 4 DOF system is obtained from an algorithm written in Matlab. The program uses the initial conditions specified for the system and determines initial trim condition estimates for the states of the system. The trim condition estimate values are used in *fmincon.m*, a Matlab function that solves the nonlinear equations of motion to find a stable solution. The equilibrium values, obtained from *fmincon.m*, are used for the ordinary differential equations (ODEs) to show that the time derivatives of all the states are zero, verifying that it is a stable solution. It is important that the initial guess is a good approximation and that appropriate constraints are imposed on the system since there is not a single unique trim solution. For example, straight and level trim flight requires the flight path angle, γ, equal to zero. There are equilibrium conditions where the constraint of $\gamma = 0$ is not valid. Therefore, to obtain the straight and level trim condition, $\gamma = 0$ must be a constraint on the system.

Next, the trim solution obtained from *fmincon.m* is used in a linearization routine. This routine provides the eigenvalues/eigenvectors and provides stability information for the system. If the real part of any eigenvalue is positive, then the system is unstable. Also, this information is used in the verification of the model. If the torsional spring attaching the wing to the fuselage is rigid, then the 4 DOF equations reduce to the 3 DOF equations. There is published data[20] on the stability characteristics of the Cessna 182 for the 3 DOF case.

Then, the trim solution is input into *ode45.m*, a Matlab continuous ordinary differential equation solver, to obtain the time history for the states, using the nonlinear equations of motion. Depending on the initial conditions of the system, the wing torsional spring stiffness, k, and the magnitude of the gust disturbance velocity, V_g, the system's response is stable at an equilibrium state or develops into a stable

LCO. See section IV for representative plots of the system's response due to a vertical gust. Recall, that the degrees of freedom are the velocities of the CM of the system in the body frame, u and w, the change in pitch attitude, θ, and a degree of freedom associated with the torsion of the wing, ϕ.

CHAPTER III

CONTROL METHODS

A.   PIF-NZSP-CRW Control

To control the 4 DOF system, a proportional-integral-filter (PIF) nonzero setpoint (NZSP) controller with control rate weighting (CRW), a type of linear quadratic regulator (LQR) controller, is used. The development of the controller, described below, is based on the PIF-NZSP-CRW formulation described by Valasek et al.[13]. The proportional integral part of the controller allows the control signal to be changed at a rate proportional to the magnitude of the error signal. By doing this, the controller operates with zero steady state error. Control rate weighting allows for weights to be put on the control rates, which gives the designer the ability to limit control weights, and avoid control saturation. Control saturation occurs when the controller requires more control authority than the system is capable of contributing.

1.   Optimal NZSP

Optimal nonzero setpoint (NZSP) is defined as a command structure that forces the plane to a terminal steady-state condition, with guaranteed tracking properties. For discrete linear time invariant (LTI) systems, the state space representation for the states is

$$\mathbf{x}_{k+1} = \Phi\mathbf{x}_k + \Gamma\mathbf{u}_k \tag{3.1}$$

and for the outputs,

$$\mathbf{y}_k = H\mathbf{x}_k + D\mathbf{u}_k \tag{3.2}$$

The desired outcome is to command select outputs, $\mathbf{y}$, to steady-state values, y_m, and to hold that value as $t \to \infty$. In the current case, the selected outputs are the velocity of the CM in the $\hat{b}_1$ direction, u, and the fuselage pitch angle, θ. The commanded values are the trim condition for the states, denoted by $*$. Therefore the steady-state system is described as

$$\mathbf{x}^*_{k+1} = \Phi \mathbf{x}^*_k + \Gamma \mathbf{u}^*_k \equiv 0 \tag{3.3}$$

and for the outputs,

$$\mathbf{y}_m = H\mathbf{x}^*_k + D\mathbf{u}^*_k \tag{3.4}$$

For guaranteed tracking the commanded outputs, $\mathbf{y}_m$ must be less than or equal to the number of controls. In this case there are only two controls, therefore only two outputs are commanded. The commanded outputs in this work are chosen to be the velocity of the system CM in the $\hat{b}_1$ direction, u, and the fuselage pitch angle, θ, which are the states normally associated with the phugoid, long period, mode in 3 DOF longitudinal dynamics. The errors between the current state and control, and their respective desired state and control are

$$\tilde{\mathbf{x}} = \mathbf{x} - \mathbf{x}^* \tag{3.5}$$

and

$$\tilde{\mathbf{u}} = \mathbf{u} - \mathbf{u}^* \tag{3.6}$$

where $\tilde{\mathbf{x}}$ is the error state and $\tilde{\mathbf{u}}$ is the error control. Therefore, the new state equation is represented as

$$\tilde{\mathbf{x}}^*_{k+1} = \Phi \tilde{\mathbf{x}}_k + \Gamma \tilde{\mathbf{u}}_k \tag{3.7}$$

with the respective quadratic function,

$$J = \frac{1}{2} \int_0^\infty \left(\tilde{\mathbf{x}}^T Q \tilde{\mathbf{x}} + \tilde{\mathbf{u}}^T R \tilde{x} u + \dot{\tilde{\mathbf{u}}}^T S \dot{\tilde{\mathbf{u}}} \right) dt \tag{3.8}$$

To determine the optimal control, which minimizes Eq. (3.8), the solution of the discrete matrix algebraic Riccati Equation, Eq. 3.9, is used.

$$P = Q + (\Phi - \Gamma K)^T P (\Phi - \Gamma K) + K^T R K \tag{3.9}$$

where

$$K = \left(\Gamma^T P \Gamma + R\right)^{-1} \left(\Gamma^T P \Phi\right) \tag{3.10}$$

The optimal control, $\tilde{u}_k$ that minimizes Eq. (3.9) is

$$\tilde{\mathbf{u}}_k = -K\tilde{\mathbf{x}}_k \tag{3.11}$$

which makes the feedback control law,

$$\begin{aligned}
\mathbf{u}_k &= \tilde{\mathbf{u}}_k + \mathbf{u}_k^* \\
&= \mathbf{u}_k^* - K\tilde{\mathbf{x}}_k \\
&= (\mathbf{u}_k^* + K\mathbf{x}_k^*) - K\mathbf{x}_k
\end{aligned} \tag{3.12}$$

where the constants $\mathbf{x}_k^*$ and $\mathbf{u}_k^*$ are solved for directly by inverting the quad partition matrix (QPM) obtained from Eqs. (3.3) and (3.4),

$$\begin{bmatrix} (\Phi - I) & \Gamma \\ H & D \end{bmatrix}^{-1} = \begin{bmatrix} \pi_{11} & \pi_{12} \\ \pi_{21} & \pi_{22} \end{bmatrix} \tag{3.13}$$

Using Eqs. (3.3) and (3.4) and assuming the desired output, $\mathbf{y}_*$, is constant, i.e. $\mathbf{x}_*$ and $\mathbf{u}_*$ are constant and implying $\dot{\mathbf{x}}_* = 0$,

$$\begin{bmatrix} \mathbf{x}_k^* \\ \mathbf{u}_k^* \end{bmatrix} = \begin{bmatrix} \pi_{11} & \pi_{12} \\ \pi_{21} & \pi_{22} \end{bmatrix} \begin{bmatrix} 0 \\ y_m \end{bmatrix} \tag{3.14}$$

Therefore, $\mathbf{x}_k^* = \pi_{12}\mathbf{y}_m$ and $\mathbf{u}_k^* = \pi_{22}\mathbf{y}_m$. Substituting these values into Eq. (3.12),

$$\mathbf{u} = (\pi_{12} + K\pi_{12})\,\mathbf{y}_m - K\mathbf{x} \tag{3.15}$$

2. PIF-NZSP-CRW

Building on the development of the NZSP controller, the proportional-integral-filter nonzero setpoint with control rate weighting (PIF-NZSP-CRW) controller adds the integral of the commanded error

$$\dot{\mathbf{y}}_I = \mathbf{y}_k - \mathbf{y}_m \tag{3.16}$$

which, using Eqs. (3.2) and (3.4), becomes

$$
\begin{aligned}
\dot{\mathbf{y}}_I &= H\mathbf{x}_k + D\mathbf{u}_k - H\mathbf{x}_k^* - D\mathbf{u}_k^* \\
&= H\tilde{\mathbf{x}}_k + D\tilde{\mathbf{u}}_k
\end{aligned}
\tag{3.17}
$$

The gains for the PIF system are designed through the LQR cost function, with a modified weighting matrix. The unmodified cost function, including the integral of the commanded error and control rate weighting, is

$$J = \frac{1}{2} \int_0^\infty \left(\tilde{\mathbf{x}}_k^T Q_1 \tilde{\mathbf{x}}_k + \tilde{\mathbf{u}}_k^T R \tilde{\mathbf{u}}_k + \dot{\tilde{\mathbf{u}}}_k^T S \dot{\tilde{\mathbf{u}}}_k + \mathbf{y}_I^T Q_2 \mathbf{y}_I \right) dt \tag{3.18}$$

Using an augmented state vector $\tilde{\mathbf{x}}_{1k} = [\tilde{\mathbf{x}}_k, \tilde{\mathbf{u}}_k, \mathbf{y}_I + I]^T$ and letting $\tilde{\mathbf{u}}_{1k} = \dot{\tilde{\mathbf{u}}}_k$, leads to

$$J = \int_0^\infty \left(\tilde{\mathbf{x}}_{1k}^T Q_{mod} \tilde{\mathbf{x}}_{1k} + \tilde{\mathbf{u}}_{1k}^T S \tilde{\mathbf{u}}_{1k} \right) dt \tag{3.19}$$

where Q_{mod} is the modified weighting matrix,

$$Q_{mod} = \begin{bmatrix} Q_1 & 0 & 0 \\ 0 & R & 0 \\ 0 & 0 & Q_2 \end{bmatrix} \tag{3.20}$$

To determine the optimal control, which minimizes Eq. (3.19), the solution of the discrete matrix algebraic Riccati Equation, Eq. 3.21, is used.

$$P = Q_{mod} + (\Phi - \Gamma K)^T P (\Phi - \Gamma K) + K^T S K \qquad (3.21)$$

where

$$K = \left(\Gamma^T P \Gamma + S\right)^{-1} \left(\Gamma^T P \Phi\right) \qquad (3.22)$$

The optimal control, $\tilde{\mathbf{u}}_k$ that minimizes Eq. (3.21) is

$$\tilde{\mathbf{u}}_{1k} = -K\tilde{\mathbf{x}}_{1k}$$
$$= -\left(K_1\tilde{\mathbf{x}}_k + K_2\tilde{\mathbf{u}}_k + K_3\mathbf{y}_I\right) \qquad (3.23)$$

which makes the actual feedback control law,

$$\mathbf{u}_{1k} = \tilde{\mathbf{u}}_{1k} + \mathbf{u}_{1k}^*$$
$$= \mathbf{u}_k^* - K_1\tilde{\mathbf{x}}_k - K_2\tilde{\mathbf{u}}_k - K_3\mathbf{y}_I \qquad (3.24)$$
$$= \left(\mathbf{u}_{1k}^* + K_1\mathbf{x}_k^* + K_2\mathbf{u}_k^*\right) - K_1\mathbf{x}_k - K_2\mathbf{u}_k - K_3\mathbf{y}_I$$

As before, the inverse of the QPM leads to $\mathbf{x}_k^* = \pi_{12}\mathbf{y}_m$ and $\mathbf{u}_k^* = \pi_{22}\mathbf{y}_m$. Recall, from the control rate weighting, $\mathbf{u}_{1k} = \dot{\mathbf{u}}_k$; therefore, $\mathbf{u}_{1k}^* = 0$. Therefore, the control law for the PIF-NZSP-CRW is

$$\mathbf{u}_{1k} = \left(K_1\pi_{12} + K_2\pi_{22}\right)\mathbf{y}_m - K_1\mathbf{x}_k - K_2\mathbf{u}_k - K_3\mathbf{y}_I \qquad (3.25)$$

The gains, K, are chosen using linear quadratic methods, which provide optimal gains. The model of the actuator dynamics, as assumed for the PIF-NZSP-CRW controller, are of the form

$$G_c = \frac{\omega_c^2}{s^2 + 2\zeta\omega_c + \omega_c^2} \qquad (3.26)$$

where G_c is the transfer function, ω_c is the natural frequency, and ζ is the damping for the actuator. For the elevator and flaperon, the actuators are assumed to have $\omega_c = 10 \frac{rad}{s}$ and $\zeta = 0.6$.

B. Nonlinear Controller

Particular interests are the determination of the adverse consequences of linearizing the equations of motion for the implementation of a linear controller and the benefits of using a nonlinear controller. Linearized idealizations of nonlinear systems may not accurately represent the dynamic response of the system. For example, nonlinear phenomena such as LCOs cannot be modeled by the linear system. In addition, in some cases the nonlinear system may be controllable where the linear system is uncontrollable. Dixon et al.[27] give a simple example, based on work by Canudas de Wit et al.[28] modeling the motion of a wheeled mobile robot, that demonstrates where a controllable nonlinear system becomes uncontrollable when linearized.

The simple nonlinear system is represented as follows

$$
\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} \cos x_3 & 0 \\ \sin x_3 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{3.27}
$$

Linearizing the equations about the point $x_3(t) = 0$, the linearized system becomes

$$
\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{3.28}
$$

This demonstrates that the dynamics for the state x_2 are now uncontrollable since the controls may have no effect on it. Also, there is the potential for the linear controller

to destabilize the nonlinear system, thereby exacerbating the response instead of alleviating it.

The nonlinear control method chosen to control the current system is based on the Lyapunov direct method. This method is not restricted to local motion about an equilibrium point, as is Lyapunov's linearization method. Rather, it determines the system's stability based on the time derivatives of the Lyapunov function, $\mathcal{V}$, which is similar to the energy of the system[29]. Using the additional insight of Mukherjee and Chen[30], global asymptotic stability may be confirmed even when the first derivative of the Lyapunov function is negative semi-definite.

The Lyapunov function chosen for the system is

$$\mathcal{V} = \frac{1}{2}\mathbf{e}^T P_1 \mathbf{e} + \frac{1}{2}\dot{\mathbf{e}}^T P_2 \dot{\mathbf{e}} \tag{3.29}$$

where P_1 and P_2 are symmetric positive definite gain matrices and $\mathbf{e}$ is the error state vector between the current state and the desired equilibrium state.

$$\mathbf{e} = \begin{bmatrix} \theta - \theta_{eq} \\ \phi - \phi_{eq} \end{bmatrix} \tag{3.30}$$

To satisfy the positive definite requirement for a Lyapunov function, $\mathcal{V}$ equals zero only when at equilibrium, i.e. $\mathbf{e} = \dot{\mathbf{e}} = 0$, and is positive everywhere else. It should be noted that, since there are only two control effectors, only two generalized coordinates may be controlled. In this work the fuselage and wing angles of rotation, θ and ϕ, are chosen to be controlled. Due to the coupled nature of the system, and the stable nature of the body fixed quasi-velocities, u and w, the response of u and w should stabilize as well.

For the Lyapunov controller method, it is necessary to determine the time deriva-

tive of the Lyapunov function.

$$\dot{\mathcal{V}} = \dot{\mathbf{e}}^T P_1 \mathbf{e} + \ddot{\mathbf{e}}^T P_2 \dot{\mathbf{e}}$$
$$= \dot{\mathbf{e}}^T (P_1 \mathbf{e} + P_2 \ddot{\mathbf{e}})$$

(3.31)

Assuming a negative semi-definite form for $\dot{\mathcal{V}}$,

$$\dot{\mathcal{V}} = -\dot{\mathbf{e}}^T \dot{\mathbf{e}}$$

(3.32)

The above model is always negative due to the negative quadratic nature of the function, but $\dot{\mathcal{V}}$ is only negative semi-definite because it is not a function of all the controlled states, $\dot{\mathbf{e}}$ and $\mathbf{e}$. Thus, $\dot{\mathcal{V}}$ may only be said to stabilize, not asymptotically stabilize, the system. This issue will be revisited later.

Using Eqs. (3.31) and (3.32),

$$-\dot{\mathbf{e}} = (P_1 \mathbf{e} + P_2 \ddot{\mathbf{e}})$$

(3.33)

Let,

$$\ddot{\mathbf{e}} = G + H \begin{bmatrix} \delta_e \\ \delta_f \end{bmatrix}$$

(3.34)

where G is the vector containing the contribution to $\dot{\mathbf{e}}$ from the non-control terms and H is the matrix containing the control term contributions. Using Eqs. (3.33) and (3.34), the nonlinear control law for stability of the system is

$$\begin{bmatrix} \delta_e \\ \delta_f \end{bmatrix} = -H^{-1}[P_2^{-1} (\dot{\mathbf{e}} + P_1 \mathbf{e}) + G]$$

(3.35)

P_1 and P_2 are chosen by the control designer. To simplify the control law, one assumes P_1 is the identity matrix and defines a new gain matrix of the following form, $P = P_2^{-1}$.

Therefore the simplified control law is

$$\begin{bmatrix} \delta_e \\ \delta_f \end{bmatrix} = -H^{-1}[P\left(\dot{\mathbf{e}} + \mathbf{e}\right) + G] \tag{3.36}$$

This control law will be implemented on the nonlinear 4DOF system. Comparing this control law to the linear control law obtained from the PIF-NZSP-CRW method will allow for conclusions to be made regarding the effectiveness of each controller.

Recall that the derivative of the Lyapunov function is negative semi-definite because it is not a function of all the states, $\mathbf{e}$ and $\dot{\mathbf{e}}$. Using the method described by Mukherjee and Chen[30], one identifies a set, $\mathcal{Z}$, that makes $\dot{\mathcal{V}} = 0$. Thus, for Eq. (3.32),

$$\mathcal{Z} : \{\mathbf{e} \, \epsilon \, \Re, \dot{\mathbf{e}} = 0\} \tag{3.37}$$

Mukherjee and Chen state that if the higher order derivatives of $\mathcal{V}$ evaluated over the set $\mathcal{Z}$ are zero, and if the first nonzero odd derivative evaluated on $\mathcal{Z}$ is negative definite, then the system is asymptotically stable. Along these lines, the second time derivative of $\mathcal{V}$ evaluated on $\mathcal{Z}$ is

$$\ddot{\mathcal{V}}|_{\mathcal{Z}} = \left(-2\ddot{\mathbf{e}}^T\dot{\mathbf{e}}\right)|_{\mathcal{Z}} = 0 \tag{3.38}$$

The third time derivative of $\mathcal{V}$ evaluated on $\mathcal{Z}$ is

$$\dddot{\mathcal{V}}|_{\mathcal{Z}} = \left(-4\left(\dddot{\mathbf{e}}^T\dot{\mathbf{e}} + \ddot{\mathbf{e}}^T\ddot{\mathbf{e}}\right)\right)|_{\mathcal{Z}} \tag{3.39}$$

Recall the assumption that P_1 is the identity matrix and the definition of a new gain matrix of the following form, $P = P_2^{-1}$. Rearranging Eq. (3.33), one finds

$$\ddot{\mathbf{e}} = -P\left(\dot{\mathbf{e}} + \mathbf{e}\right) \tag{3.40}$$

Substituting Eq. (3.40) into Eq. (3.39),

$$
\begin{aligned}
\dddot{\mathcal{V}}|_{\mathcal{Z}} &= -4 \left(\ddot{\mathbf{e}}^T \dot{\mathbf{e}} + (P(\dot{\mathbf{e}} + \mathbf{e}))^T (P(\dot{\mathbf{e}} + \mathbf{e})) \right) |_{\mathcal{Z}} \\
&= -4 \left((P\mathbf{e})^T (P\mathbf{e}) \right)
\end{aligned}
\tag{3.41}
$$

which shows that $\dddot{\mathcal{V}}$ is negative definite because it is zero only at $\mathbf{e} = \dot{\mathbf{e}} = 0$, and is negative elsewhere. Therefore, the control law derived in Eq. (3.36) is asymptotically stabilizing for the system.

CHAPTER IV

RESULTS

In this section, results from three main areas of interest will be presented: open-loop response for both linear and nonlinear models, closed-loop response with a linear controller for both linear and nonlinear models, and closed-loop response with a non-linear controller for the nonlinear model. Following the results is a discussion of assumptions.

Results are investigated using the one-minus-cosine gust disturbance, as described in section 3. The maximum vertical gust velocity is chosen to be $0.03u_d$. Though the equations of motion, Eqs. (2.14) - (2.17), include a lift due to both a horizontal and vertical gust, this research will focus solely on a vertical gust. The gust disturbance begins at time equal to 10 seconds for the simulations and has a one second duration, i.e. $dm = 0.5$. The gust disturbance is delayed to show that the system is at trim initially. System parameters are shown in Table I.

A.   Open-Loop Response

The first case is the response of the open-loop linear model of the system in comparison to the full nonlinear model. The linear and nonlinear model responses for the fuselage pitch angle, θ, and wing pitch angle, ϕ, and the velocities of the system CM in the b_1 direction, u, and in the b_3 direction, w, are shown in Figs. 5 and 6.

The main input parameters for this simulation are shown in Table II, where k is the stiffness for the torsional attachment for the wing, and u_d is the desired horizontal trim velocity for the system.

Table I. System Parameters

$Variable$	$Value$	$Units$	$Description$
W	2650	lb	Weight
I_f	1143.8	$lb - ft^2$	Fuselage Moment of Inertia
I_w	201.9	$lb - ft^2$	Wing Moment of Inertia
b	36	ft	Span
S_w	174	ft^2	Wing Area
c	4.9	ft	Wing Chord
x_{CM}	26.3	$\%$	System CM location (% chord)
$S_{\delta f}$	15	ft^2	Flaperon Area
$S_{\delta e}$	16.61	ft^2	Elevator Area
h	5000	ft	Altitude

It is apparent by the response, of both the linear and nonlinear models, that this is a stable condition. Note that the linear model is more oscillatory than the nonlinear model. This response is expected since the nonlinear and higher order terms are neglected in the linear model, lowering the overall damping in the system. The linear and nonlinear models have been compared by applying a small disturbance, of the order of a maximum vertical gust velocity of $(-2 \times 10^{-5})u_d$, and showing both systems produce the same response.
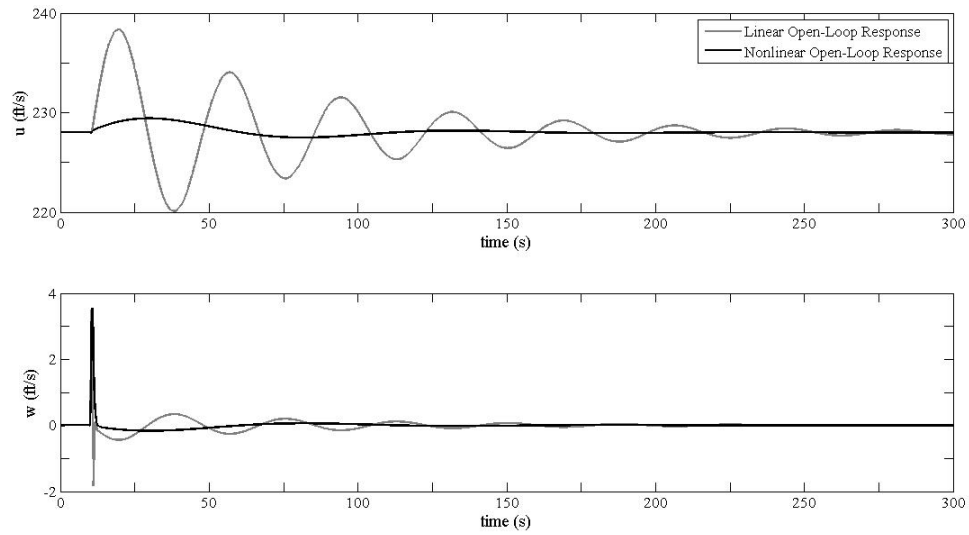
Fig. 5. Velocities of the System CM in the b_1 Direction (u) and the b_3 Direction (w): Linear/Nonlinear Open-Loop Response
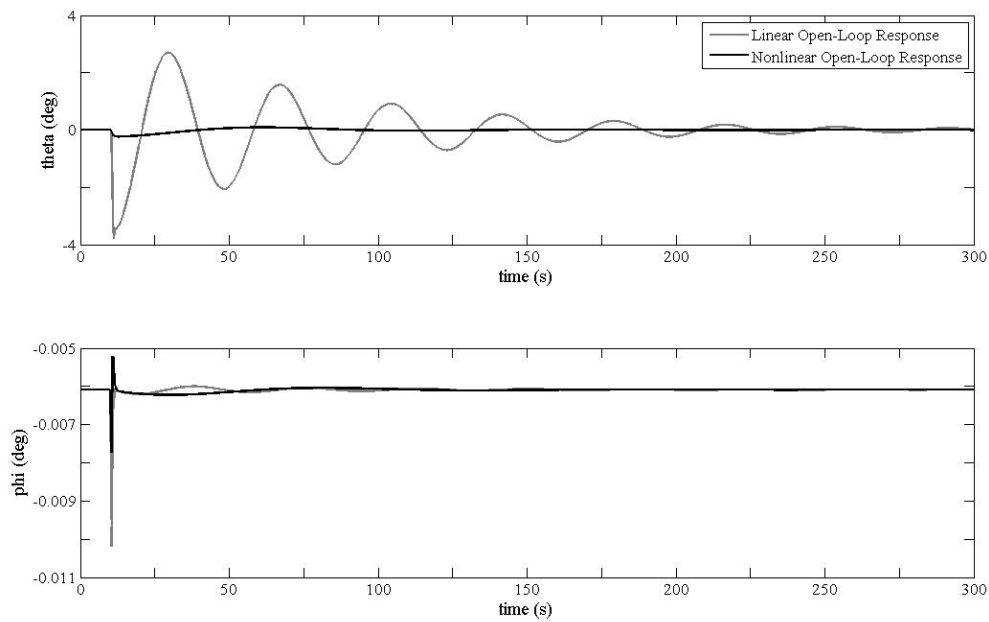
Fig. 6. Fuselage (θ) and Wing (ϕ) Pitch Angles: Linear/Nonlinear Open-Loop Response

Table II. Input Parameters: Case 1

$Variable$	$Value$	$Units$
k	10^7	$\frac{lb-ft}{rad}$
u_d	228	$\frac{ft}{s}$
ω_w	222.54	$rad-s^{-1}$

B.   PIF-NZSP-CRW Linear Controller

For the next set of simulations, a second case (see Table III) of input parameters is used to trigger an LCO response. The short duration of the gust disturbance is chosen in order to excite the elastic mode from the torsional wing[22].

Table III. Input Parameters: Case 2

$Variable$	$Value$	$Units$
k	10^5	$\frac{lb-ft}{rad}$
u_d	244.5	$\frac{ft}{s}$
ω_w	22.25	$rad-s^{-1}$

The system is sensitive to the values of the wing torsional stiffness, k, and the desired trim velocity, u_d. The plots that follow show both the closed and open-loop responses for the linear and nonlinear models with a linear PIF-NZSP-CRW controller imposed on the systems. It is shown (see Figs. 7 and 8) that the linear controller controls the linear system. The gains, see section A chapter III, are chosen to achieve stability and to ensure the control rate and angle limitations are not violated. See Figs. 9 and 10 for the time history for both controllers. Figures 11 and 12 show the linear controller applied to the nonlinear system. For this implementation the

exact same control law from the PIF-NZSP-CRW linear formulation is imposed on the nonlinear system.

Fig. 7. Velocities of the System CM in the b_1 Direction (u) and the b_3 Direction (w): Linear Response with and without Linear Control
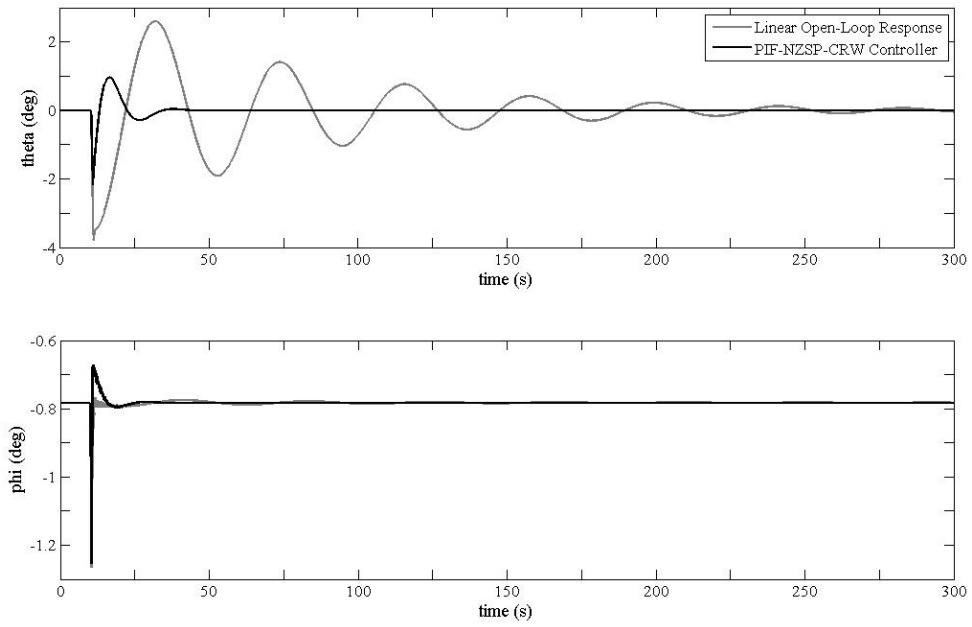
Fig. 8. Fuselage (θ) and Wing (ϕ) Pitch Angles: Linear Response with and without Linear Control
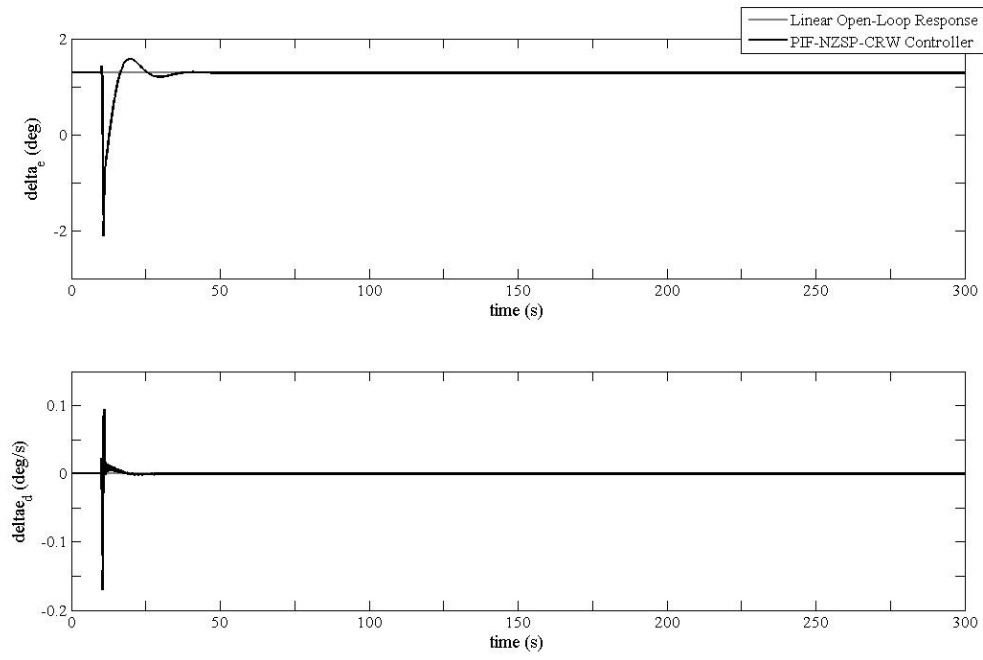
Fig. 9. Elevator Deflection Angle, δ_e, and rate, $\dot{\delta}_e$
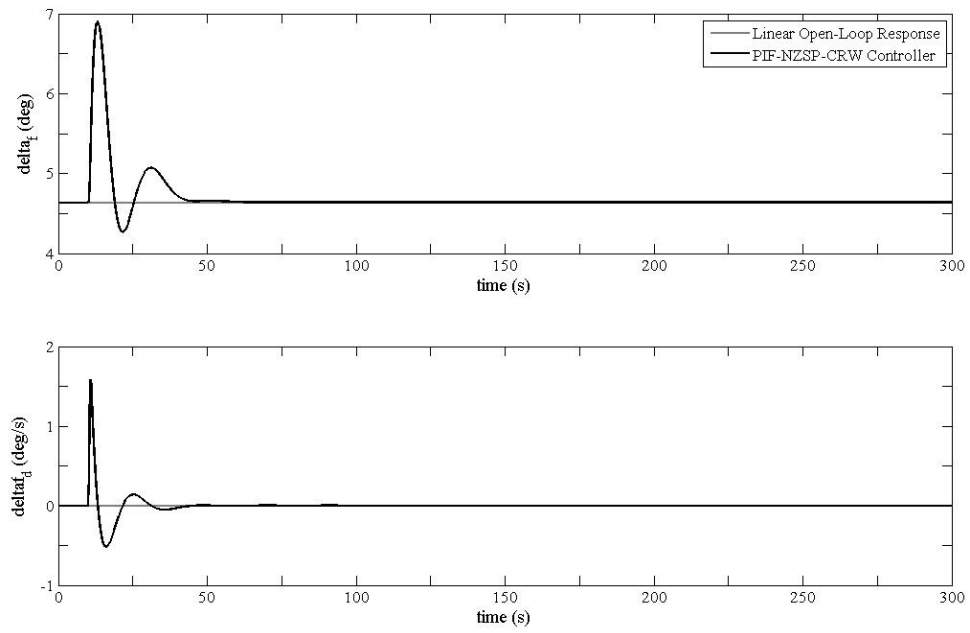
Fig. 10. Flaperon Deflection Angle, δ_f, and rate, $\dot{\delta}_f$

Fig. 11. Velocities of the System CM in the b_1 Direction (u) and the b_3 Direction (w): Nonlinear Response with and without Linear Control
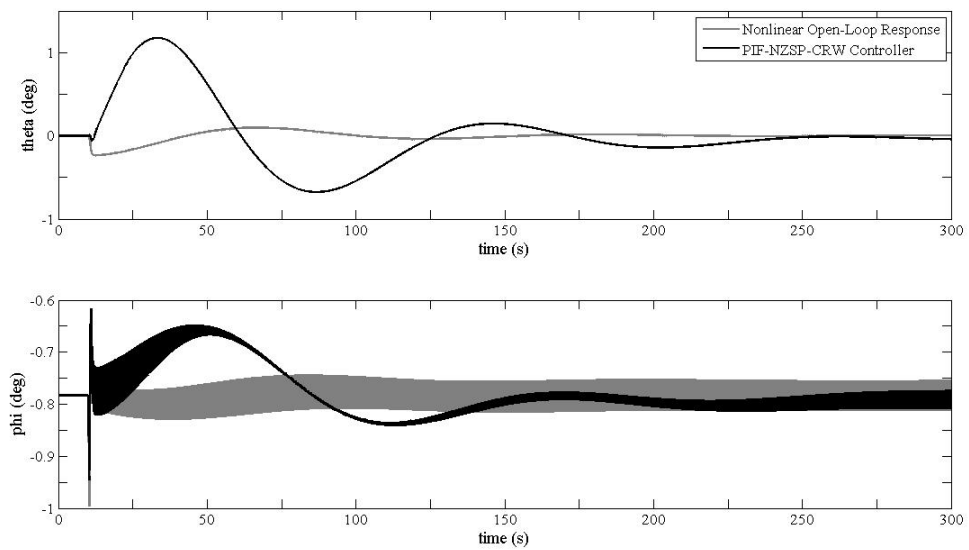
Fig. 12. Fuselage (θ) and Wing (ϕ) Pitch Angles: Nonlinear Response with and without Linear Control

Figures 11 and 12 show that the input values produce a LCO for the nonlinear model. The LCO is not present in the linear response (see Figs. 7 and 8) since the LCO is strictly a nonlinear phenomena. It should be noted that a linear controller implemented on a nonlinear model can produce an LCO response. This response is shown in Fig. 12. It is found that the linear controller is unable to stabilize the LCO, though it is able to reduce the LCO's amplitude. As seen in Fig. 11, the linear controller worsens the response of the system, making the amplitude of the oscillatory response higher than the open-loop response.
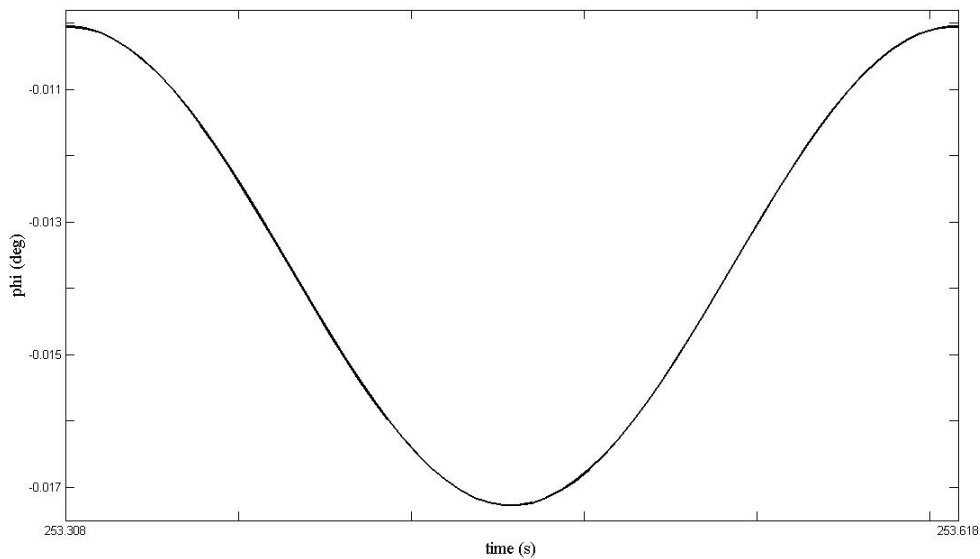
Fig. 13. Close Up of One Period of Oscillation for the Wing Pitch Angle, ϕ, LCO

Figure 13 is a close up of one period of oscillation for the LCO for the nonlinear open-loop solution. From this the frequency of LCO is determined to be approximately $20.3 \frac{rad}{s}$.

C.   Nonlinear Lyapunov-Based Control

The input parameters of Table III are used in the simulations provided in Figs. 14 and 15. These cases implement the nonlinear Lyapunov-based controller on the nonlinear system. The gain matrix, P, for the nonlinear controller, see section B, is chosen to be of the form

$$P = \begin{bmatrix} c_1\omega_w & 0 \\ 0 & c_2\omega_w \end{bmatrix} \tag{4.1}$$

where ω_w is the natural frequency for the wing, see Table I, and c_1 and c_2 are arbitrary constants chosen by the control designer. As shown in Figs. 14 and 15, the nonlinear controller compensates for the LCOs in the system and stabilizes the system to the original equilibrium condition.
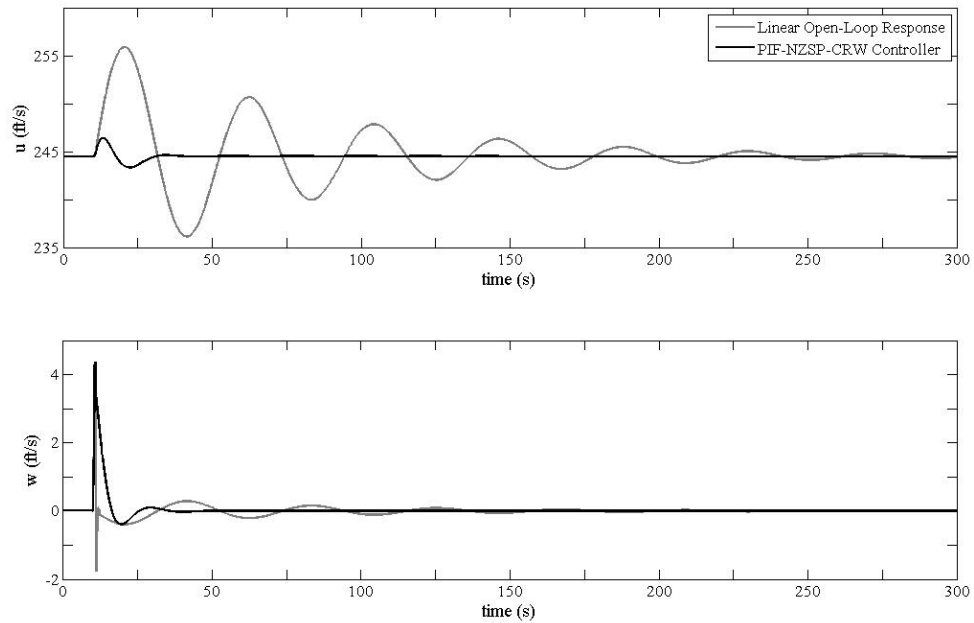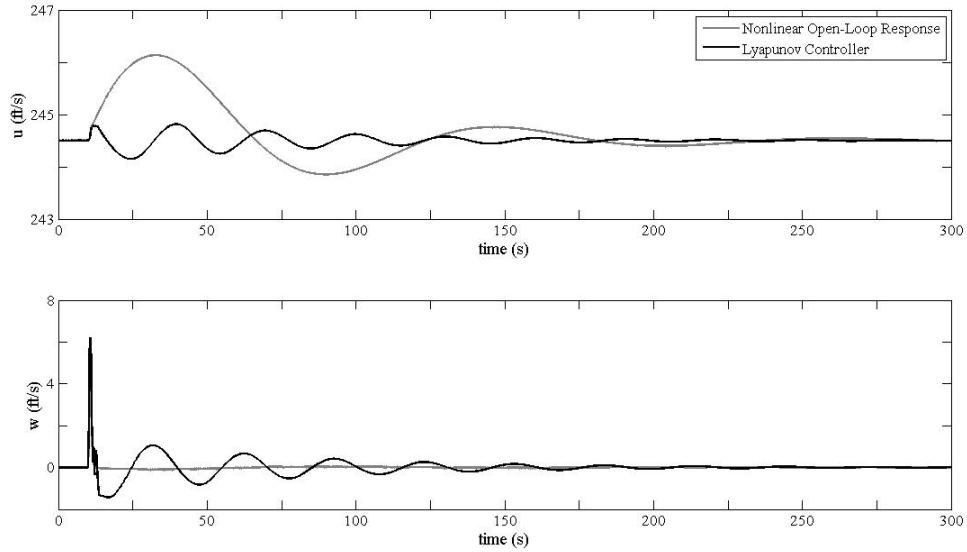

Fig. 14. Velocities of the System CM in the b_1 Direction (u) and the b_3 Direction (w): Nonlinear Response with and without Nonlinear Control

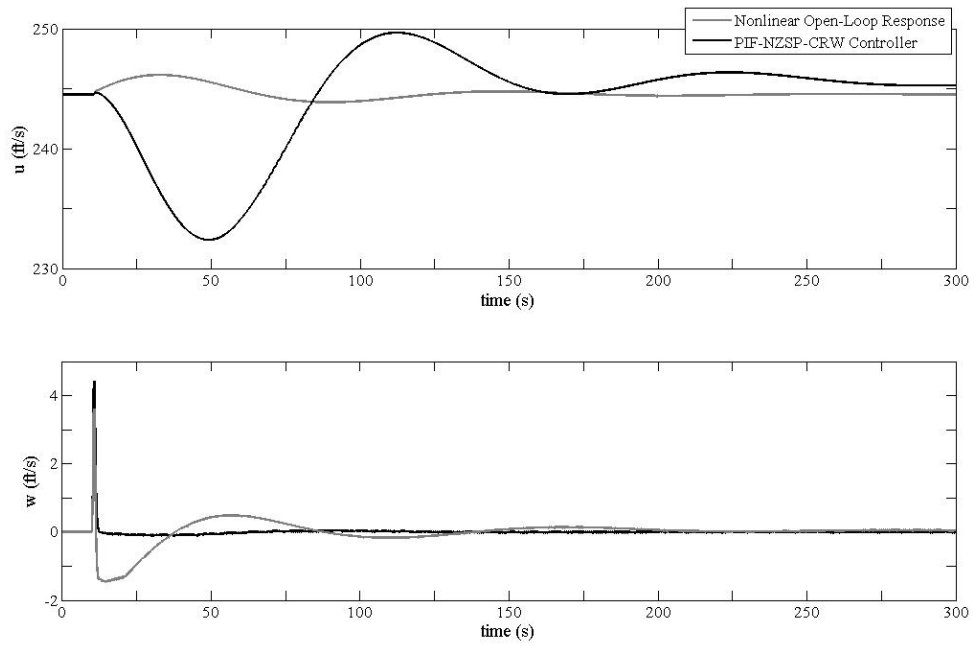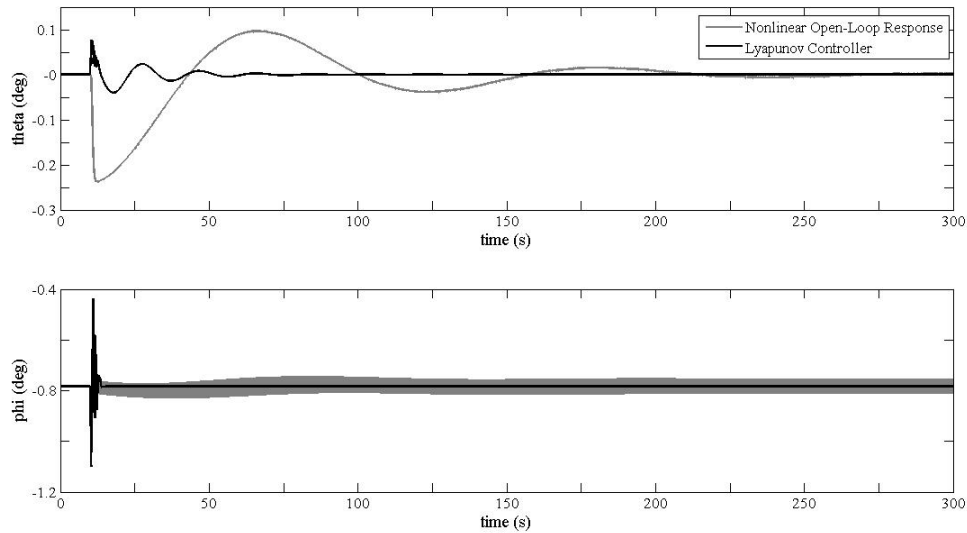Fig. 15. Fuselage (θ) and Wing (ϕ) Pitch Angles: Nonlinear Response with and without Nonlinear Control

D. Assumptions

As is the case in all analysis, assumptions are required to reduce the problem being investigated to a manageable level. For the current research, assumptions are made with respect to the aerodynamics, structure, and controls.

With respect to the aerodynamics, the steady-flow aerodynamics model used is a simplification of the true aerodynamic model. This is done to focus attention to the nonlinear dynamics of the structure and to reduce complexity of analysis. In addition, fuselage aerodynamic effects are neglected. This is done due to lack of information concerning the aerodynamics of the fuselage for the aircraft, and to further simplify the model. It is also assumed that the coefficient of drag with respect to the control surfaces is negligible. This is consistent with other longitudinal aircraft studies[20].

Assumptions in the structural dynamic model partially lie in the reduction of the full 6 DOF aircraft model to the current 4 DOF model that only includes the

longitudinal model. The current model constrains the aircraft from roll and yaw motions, and translation in the $\hat{b}_2$ direction. In addition, the model assumes a rigid wing attached to the aircraft with a torsional spring. While this is a simplified model of a flexible wing in torsion, the true aircraft would have flexibility in many structural modes including both bending and torsion, and will add many DOFs to the analysis.

Assumptions, made in the controls, allow for a continuous controller that updates every timestep. In reality, the controller would be unable to update continuously due to computational and structural limitations.

The author feels that the current assumptions of this research allow for a more detailed analysis of a simpler problem, and makes the analysis more trackable. The current model forms a basis for the addition of more complicated aerodynamics and structural considerations, and allows for a transition between the fully flexible and the rigid model, which would aid in the validation of flexible models.

E. Frequency Analysis

It is of interest to identify and discuss the amplitudes and frequencies for each of the states. Frequency characteristics of the linear system will be analyzed and compared with frequency analysis obtained from the time histories of the state from the simulation. The eigenvalues for the linear system using the physical properties of Table III are shown in Table IV. The frequencies and damping ratios, see Table V, are calculated from the eigenvalues.

In order to obtain the plots for amplitude and frequency, a fast fourier transform (FFT) is performed on the time history for each state. The FFT function in Matlab assumes a constant sampling size for the data, which makes it important to have a constant step size for integration solver. For the linear analysis a constant step size

Table IV. Eigenvalues

$Eigenvalues$
-0.203 ± 24.3
-3.36 ± 4.62
$-1.46 \times 10^{-2} \pm 0.150$

Table V. Frequencies and Damping Ratios

$Frequency(rad/s)$	$Damping Ratio$
24.3	8.36×10^{-3}
5.70	0.587
0.151	9.71×10^{-2}

is already implemented, for the nonlinear model, a constant step size Runga Kutta solver, such as *runge.m*, should be used instead of a variable time step solver, *ode45.m*. The step size is chosen by the designer, but care must be taken to make sure the step size is small enough to capture the nonlinear response.

In Figs. 16 - 21 the frequency response for the open-loop linear response, using the second set of physical properties (Table III), are shown. The wing natural frequency is 22.54 $rad - s^{-1}$ for this set of initial conditions.

The vertical lines in Figs. 16 - 21 are the system frequencies obtained through eigenvalue analysis (see Table V). The sharpness of the peaks for each frequency vary according to the respective damping ratios. The highest frequency has the lowest damping ratio (see Table V), which is why the peak for the highest frequency is the most pronounced. The middle frequency, normally referred to as the short period mode in 3 DOF longitudinal analysis, has the greatest damping ratio, and thus has the least pronounced peak of the three frequencies. And the lowest frequency, normally

referred to as the phugoid mode, has a damping ratio in between the two other frequencies, and the sharpness of its peak reflects this relationship. These results show that the simulation of the linear system matches with results from an eigenvalue analysis, validating the linear simulation. As previously stated, for small disturbances, the linear and nonlinear simulations of the response are the same, therefore the current validation for the linear simulation also validates the nonlinear simulation.
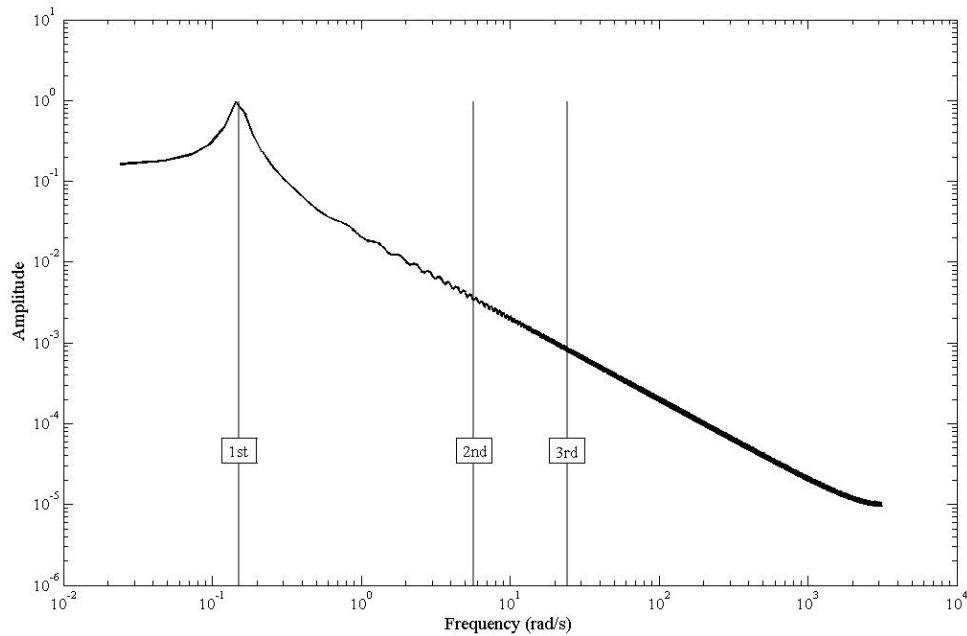
Fig. 16. Frequency Response as Derived from Simulation of Velocity, u
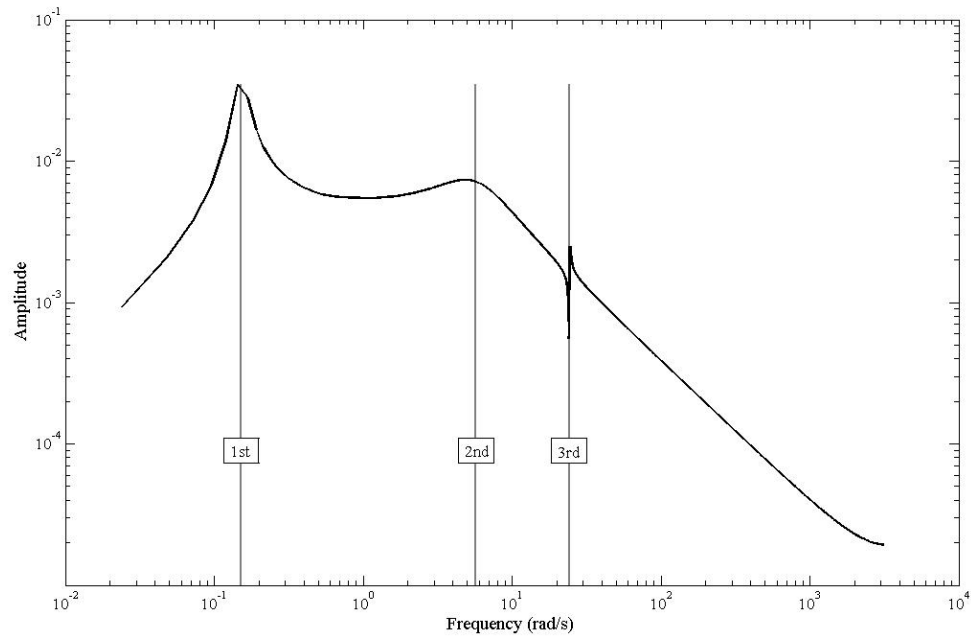
Fig. 17. Frequency Response as Derived from Simulation of Velocity, w
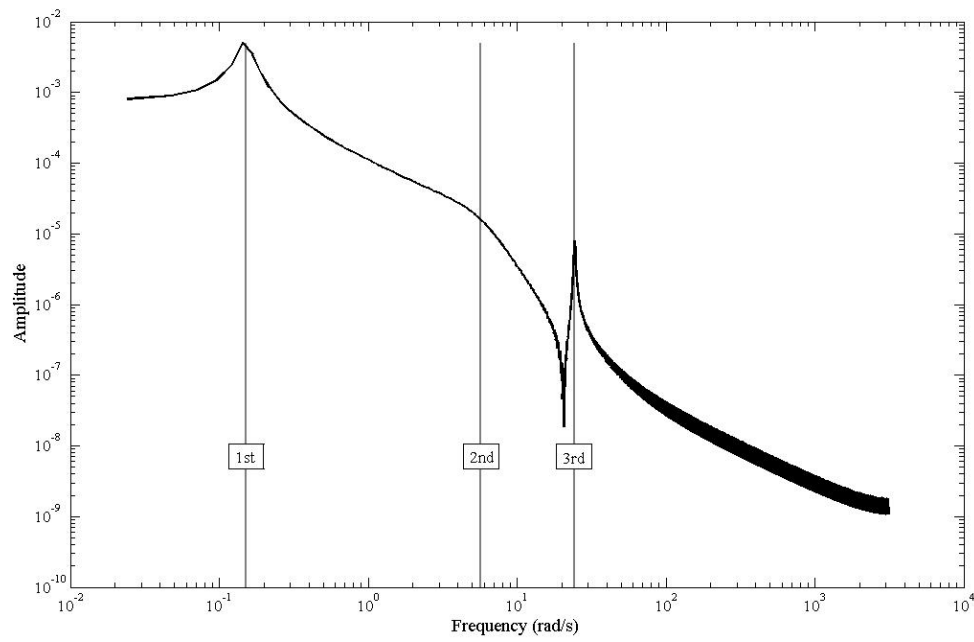
Fig. 18. Frequency Response as Derived from Simulation of Fuselage Pitch Angle, θ
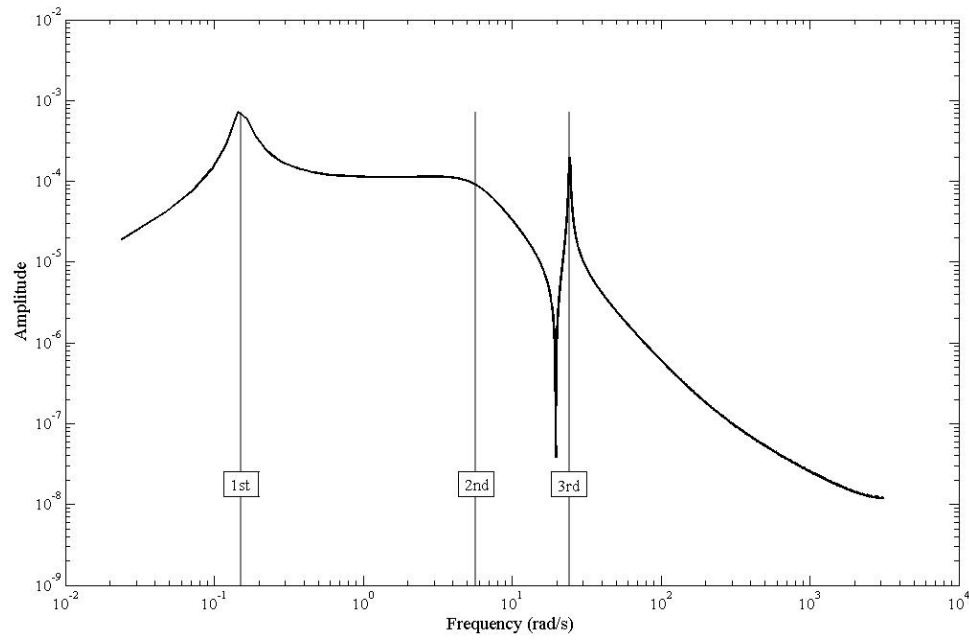
Fig. 19. Frequency Response as Derived from Simulation of Fuselage Pitch Rate, $\dot{\theta}$
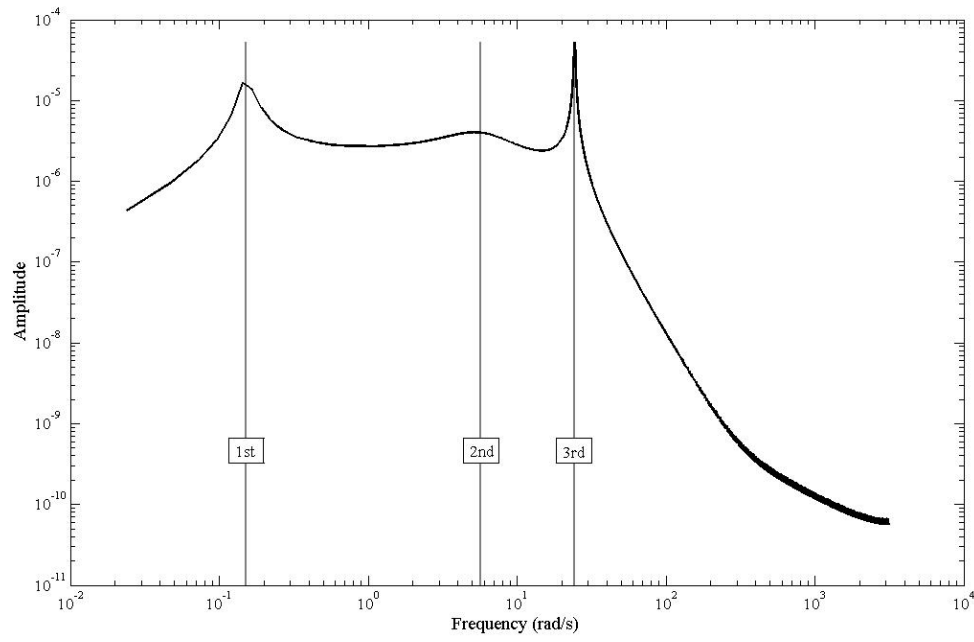
Fig. 20. Frequency Response as Derived from Simulation of Wing Pitch Angle, ϕ
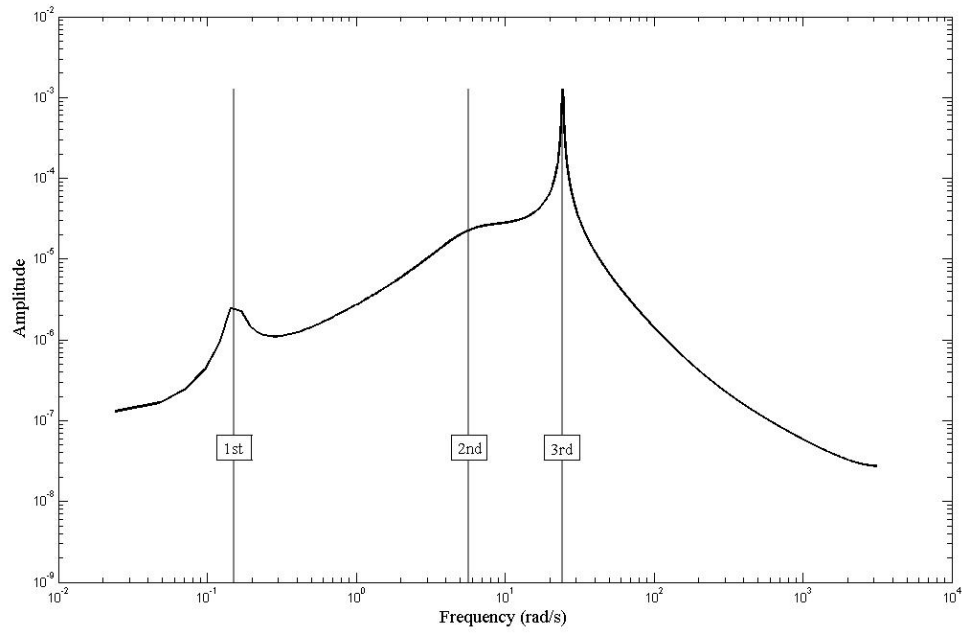
Fig. 21. Frequency Response as Derived from Simulation of Wing Pitch Rate, $\dot{\phi}$

It should be noted that for the nonlinear system, the amplitudes and the frequencies are dependent on the initial conditions and the disturbance imposed. For a complete analysis of the system, bifurcation analysis tools should be used to obtain the complete frequency-response curves for the nonlinear system.

CHAPTER V

CONCLUSIONS

A primary contribution of this research is to examine the adverse consequences produced in linearizing the equations of motion and in implementing a linear controller on a nonlinear system. The importance of using a nonlinear model instead of a linearized model is evident in Figs. 5 and 6. The simulations show that even for a relatively minor velocity disturbance, the responses for the linear and nonlinear models differ substantially.

Not only is the nonlinear model important, but also is the implementation of a nonlinear controller. As seen in Figs. 11 and 12, the linear controller worsens the response for the velocities and fuselage pitch angle for the system. It is interesting that the linear controller did help alleviate the amplitude of the LCO that the wing pitch angle, ϕ, experiences, see Fig. 12, but linear control cannot completely control the system because it does not fully compensate for the LCO in the nonlinear model. As seen in Fig. 8, the linear system does not experience the nonlinear LCO phenomena. The nonlinear model with the nonlinear Lyapunov-based controller stabilizes the system for the LCO. Figure 14 shows that the uncontrolled states, u and w, also stabilize to their original equilibrium values.

The results included in this paper show that the linear model fails to capture the true nonlinear response of the system, for example, the presence of an LCO. Further, the linear controller based on the linear model does not stabilize the true nonlinear system. The results from the Lyapunov-based control on the nonlinear model emphasize the importance of treating the fully nonlinear system with a nonlinear controller.

CHAPTER VI

FUTURE WORK

One aspect that is of great interest is bifurcation analysis of the system. The bifurcation analysis is useful in the rapid and complete determination the characteristics of the system, including unstable LCOs, stable LCOs, catastrophic divergence, flutter, etc. and will identify responses where control is necessary. Also, it should be of interest to perform bifurcation on the closed loop solution of the equations of motion to determine the stability characteristics of the controlled system.

The ideal next step of this research is to implement a rigidly attached flexible wing onto the longitudinal model that would allow for torsion and bending. The wing should be modeled as a cantilever beam with one end rigidly attached and the other free[31]. The fixed end must be allowed to translate and rotate, which will allow for the fuselage motion to be imposed onto the wing. One of the complexities of this problem is due to the inherently coupled nature of the aerodynamics and the structure. The aerodynamic forces depend on the angle of attack of the wing, but the wing twist angle also depends on the aerodynamic forces. This is a classic aeroelastic problem, in which iteration is necessary to obtain a solution.

One issue that must be addressed is the point at which the inertial velocities of the system are defined. It will be necessary to redefine the inertial velocities for system as that of the attachment point of the wing instead of the CM of the system, as is currently used. This is an issue because the CM of the system depends on the wing configuration and is therefore not fixed. If the attachment point for the wing is chosen as the defined value, then the fixed end of the cantilever is defined. Otherwise the fixed end would be a function of the moving CM of the system, which is in turn

a function of the twist and bending of the wing, which adds unnecessary complexity to the problem.

Some additional areas of interest include implementing different nonlinear control models and a more rigorous aerodynamic model of unsteady flow. The work focused on a Lyapunov nonlinear control method, which is a fairly robust method, but there are two additional nonlinear control methods of interest: adaptive control and sliding mode control. The adaptive control design accounts for the modeling uncertainties and corrects for the uncertainties caused by the inexact cancellation of nonlinear terms for the feedback linearization. The control laws developed from the use of adaptive methods allow for the LCO and flutter to be suppressed during normal flight operations, with the goal of creating a globally stabilizing controller. Sliding mode control allows for a stable mode to be achieved and then provides control to constrain the response to remain at the stable mode. Both methods should be explored to identify stability characteristics and their effectiveness for flutter and LCO suppression, as well as gust load alleviation. Also there is the opportunity to investigate adaptive nonlinear Lyapunov control. This would all the control law to adapt to better control the response.

The aerodynamic model currently implemented is a steady flow model. For future work, it is desired to implement a more robust model, such as the unsteady model of Theodorsen's unsteady thin-airfoil theory[26]:

$$L = 2\pi\rho_\infty UbC(k)\left[\dot{h} + U\theta + b\left(\frac{1}{2} - a\right)\dot{\theta}\right] + \pi\rho_\infty b^2\left(\ddot{h} + U\dot{\theta} - ba\ddot{\theta}\right) \qquad (6.1)$$

where the function C(k) is a function of the reduced frequency, $k = \frac{b\omega}{U}$, b' is the semi-chord, a is a dimensionless parameter for the length of the midchord to the hinge location, U is the velocity, and θ is the pitch angle of the wing.

An additional area of interest could be a more realistic gust model. The one-minus-cosine gust is good for modeling one discrete gust, but a more accurate model would allow for the gust to be continuous and irregular. An approach for this could be idealization of the gust as a stationary gaussian random process. Stationary-gaussian idealization allows for an infinite possible variations in the shape of the individual gusts, variation of gust magnitude and duration, and superposition of short duration gusts that excite elastic modes with longer duration gusts that give the largest rigid airplane loads[22].

One asset of the University is the Nonlinear Aeroelastic Testbed Apparatus (NATA). NATA is a pitch plunge apparatus that may be used to implement the nonlinear control algorithms and experimentally validate results obtained through simulations.

REFERENCES

[1] E. Dowell, J. Edwards, and T. Strganac, "Nonlinear aeroelasticity," *Journal of Aircraft*, vol. 40, no. 5, pp. 857–874, 2003.

[2] M. J. Patil, D. H. Hodges, and C. E. S. Cesnik, "Nonlinear aeroelasticity and flight dynamics of high-altitude long-endurance aircraft," *AIAA Paper 99-1470*, 1999.

[3] T. E. Noll, J. M. Brown, M. E. Perez-Davis, S. D. Ishmael, G. C. Tiffany, and M. Gaier, "Investigation of the helios prototype aircraft mishap," vol. 1, January 2004, [online article], http://www.nasa.gov/pdf/64317main_helios.pdf [retrieved 14 April. 2009].

[4] C. M. Shearer and C. E. S. Cesnik, "Nonlinear flight dynamics of very flexible aircraft," *Journal of Aircraft*, vol. 44, no. 5, pp. 1528–1545, 2007.

[5] E. Vartio, A. Shimko, C. P. Tilmann, and P. M. Flick, "Structural modal control and gust load alleviation for sensorcraft concept," in *Proc. of the 46th AIAA/ASME/ASCE/AHS/ASC Structural Dynamics and Materials Conference*, ser. AIAA-2005-1946, Austin, TX, April 2005.

[6] Y. Matsuzaki, T. Ueda, Y. Miyazawa, and H. Matsushita, "Gust load alleviation of a trasport-type wing: Test and analysis," *Journal of Aircraft*, vol. 26, no. 4, pp. 322–327, 1989.

[7] K. B. Penning, P. S. Zink, P. Wei, A. P. De La Garza, M. H. Love, and J. Martinez, "Gla and flutter supression for a sensorcraft class concept using system identification," in *Proc. of the 26th AIAA Applied Aerodynamics Conference*, ser. AIAA-2008-7188, Honolulu, HI, August 2008.

[8] J. Block and T. W. Strganac, "Applied active control for a nonlinear aeroelastic stucture," *Journal of Guidance, Control, and Dynamics*, vol. 21, no. 6, pp. 838–845, 1998.

[9] G. Platanitis and T. Strganac, "Control of a nonlinear wing section using leading- and trailing-edge surfaces," *Journal of Guidance, Control, and Dynamics*, vol. 27, no. 1, pp. 52–58, 2004.

[10] B. L. Stauffer, R. A. Stuever, and J. L. Vian, "Application of a design method for integrated control to a vtol airplane in hover," in *Proc. of the AIAA Guidance, Navigation and Control Conference*, ser. AIAA-90-3334-CP, Portland, OR, August 1990.

[11] J. Valasek, J. Kimmett, D. Hughes, K. Gunnam, and J. L. Junkins, "Vision based sensor and navigation system for autonomous aerial refueling," in *Proc. of the 1st UAV Conference*, ser. AIAA Paper 2002-3441, Portsmouth, VA, May 2002.

[12] J. Kimmett, J. Valasek, and J. L. Junkins, "Autonomous aerial refueling utilizing a vision based navigation system," in *Proc. of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, ser. AIAA Paper 2002-4469, Monterey, CA, August 2002.

[13] J. Valasek, K. Gunnam, J. Kimmett, M. Tandale, and J. L. Junkins, "Vison-based sensor and navigation system for autonomous air refueling," *Journal of Guidance, Control, and Dynamics*, vol. 28, no. 5, pp. 979–989, 2005.

[14] E. de Pavia, F. Benjovengo, S. Bueno, J. Azinheira, and A. Moutinho, "Nonlinear control approaches for an autonomous unmanned robotic airship," in *Proc. of*

the 7th AIAA Aviation Technology, Integration and Operations Conference, ser. AIAA-2007-7782, Belfast, Northern Ireland, September 2007.

[15] J. Suk, S. Boo, and Y. Kim, "Lyapunov control law for slew maneuver using time finite element analysis," *Journal of Guidance, Control, and Dynamics*, vol. 24, no. 1, pp. 87–94, 2001.

[16] P. Bhatta and N. E. Leonard, "Nonlinear gliding stabiliy and control for vehicles with hydrodynamic forcing," *Automatica*, vol. 44, pp. 1240–1250, 2008.

[17] F. Lanchester, *Aerodonetics*. London: A. Constable & Company, 1908.

[18] R. von Mises, *Theory of Flight*. New York, NY: Dover, 1959.

[19] R. C. Nelson, *Flight Stability and Automatic Control*. New York, NY: McGraw-Hill Higher Education, 1998.

[20] J. Roskam, *Airplane Flight Dynamics and Automatic Flight Controls: Part 1*. Lawrence, KS: DARcorporation, 1995, pp. 480–486.

[21] J. Hurtado, *Kinematic and Kinetic Principles*. College Station, TX: John E. Hurtado/Lulu, 2007, p. 113.

[22] F. M. Hoblit, *Gust Loads on Aircraft: Concepts and Applications*. Washington, DC: AIAA Education Series, 1988, pp. 1–8.

[23] P. Jackson, Ed., *Jane's All the World's Aircraft 2007-2008*. Alexandria, VA: Jane's Information Group Inc., 2007, pp. 718–719.

[24] C. D. Perkins and R. E. Hage, *Airplane Performance Stability and Control*. New York, NY: John Wiley & Sons, Inc., 1949, p. 329.

[25] J. B. Rathbun, *Aeroplane Construction and Operation.* Stanton and Van Vliet, 1918, p. 321.

[26] D. H. Hodges and G. A. Pierce, *Introduction to Structural Dynamics and Aeroelasticity.* New York, NY: Cambridge University Press, 2002.

[27] W. E. Dixon, A. Behal, D. M. Dawson, and S. P. Nagarakatti, *Nonlinear Control of Engineering Systems.* Boston, MA: Birkäuser, 2003.

[28] C. Canudas de Wit, K. Khennouf, C. Samson, and O. J. Sordalen, "Nonlinear control for mobile robots," in *Recent Trends in Mobile Robots*, Y. Zheng, Ed. River Edge, NJ: World Scientific, 1993, pp. 121–156.

[29] J. E. Slotine and W. Li, *Applied Nonlinear Control.* Englewood Cliffs, NJ: Prentice Hall, 1991, p. 40.

[30] M. Mukherjee and D. Chen, "Asymptotic stability theorem for autonomous systems," *Journal of Guidance, Control, and Dynamics*, vol. 16, no. 5, pp. 961–963, 1993.

[31] S. S. Rao, *Vibration of Continuous Systems.* Hoboken, NJ: John Wiley and Sons, Inc, 2007.

APPENDIX A

GIBBS-APPELL DERIVATION

For the Gibbs-Appell methods, a scalar function in acceleration terms, known as the energy of acceleration, is used. The energy of acceleration equation for the system is

$$\mathbf{S} = \frac{1}{2} \left(m\mathbf{a}_{cg/i} \cdot \mathbf{a}_{cg/i} + \frac{1}{2} \left(\boldsymbol{\alpha} \cdot \dot{\mathbf{H}}_{cg} \right) + \boldsymbol{\alpha} \cdot \left(\boldsymbol{\omega} \times \mathbf{H}_{cg} \right) \right) \tag{A.1}$$

where the slash notation 'a/b' denotes the vector from b to a. The term $\boldsymbol{\alpha}$ is the angular acceleration for the body, not to be confused with angle of attack. Also, $\boldsymbol{\omega}_{cg/i} = \dot{\theta}\hat{b}_2$ and $H_{cg} = I_{cg}\omega_{cg/i}$; therefore, $(\boldsymbol{\omega} \times \mathbf{H}_{cg}) = 0$.

A.   Energy of Acceleration

As shown in Eq. (A.1) the acceleration terms for each of the bodies, wing and fuselage, are needed. In order to obtain the acceleration components of the fuselage and wing, their relationship with the CM of the system must be established. Given the definition of the CM,

$$\sum_{i=1}^{N} m_i \mathbf{r}_i = 0 \tag{A.2}$$

In order to obtain the velocity relationships, differentiate Eq. (A.2)

$$\sum_{i=1}^{N} m_i \mathbf{v}_i = 0 \tag{A.3}$$

Note, v_i is the inertial velocity of each body. Expanding Eq. (A.3),

$$m\mathbf{v}_{CMs} = m_w\mathbf{v}_{CMf} + m_w\mathbf{v}_{CMw} \tag{A.4}$$

where the subscript s stands for the system. Investigating the Eq. (A.3) for

velocities of the bodies with respect to the CM of the fuselage,

$$m\mathbf{v}_{CMs/CMf} = m_w\mathbf{v}_{CMf/CMf} + m_w\mathbf{v}_{CMw/CMf} \tag{A.5}$$

where $v_{f/f}$ equals zero. Thus,

$$\mathbf{v}_{CMs/CMf} = \frac{m_w}{m}\mathbf{v}_{CMw/CMf} \tag{A.6}$$

and similarly,

$$\mathbf{v}_{CMs/CMw} = \frac{m_f}{m}\mathbf{v}_{CMf/CMw} \tag{A.7}$$

Recall that for any body fixed vector,

$$\mathbf{v}_{a/b} = -\mathbf{v}_{b/a} \tag{A.8}$$

Therefore,

$$\mathbf{v}_{CMw/CMs} = \frac{m_f}{m}\mathbf{v}_{CMw/CMw} \tag{A.9}$$

and

$$\mathbf{v}_{CMf/CMs} = -\frac{m_w}{m}\mathbf{v}_{CMw/CMw} \tag{A.10}$$

Taking the time derivatives of both Eq. A.9 and A.10 leads to

$$\mathbf{a}_{CMw/CMs} = \frac{m_f}{m}\mathbf{a}_{CMw/CMf} \tag{A.11}$$

and

$$\mathbf{a}_{f/CMs} = -\frac{m_w}{m}\mathbf{a}_{CMw/CMf} \tag{A.12}$$

Thus, to find both $\mathbf{a}_{CMf/CMs}$ and $\mathbf{a}_{CMw/CMs}$, $\mathbf{a}_{CMw/CMf}$ is needed. Recall,

$$\mathbf{r}_{CMw/CMf} = \left(d\hat{b_1} + d_c\hat{w_1}\right)$$
$$= (d + d_c\cos\phi)\,\hat{b_1} - d_c\sin\phi\hat{b_3} \tag{A.13}$$

with the time derivative

$$
\begin{aligned}
\mathbf{v}_{CMw/CMf} &= {}^b\frac{d}{dt}\left(r_{CMw/CMf}\right) + \boldsymbol{\omega}_{b/i} \times \left(r_{CMw/CMf}\right) \\
&= -\left(\dot{\theta}+\dot{\phi}\right)d_c\sin\phi\,\hat{b}_1 - \left(d\dot{\theta}+\left(\dot{\theta}+\dot{\phi}\right)d_c\cos\phi\right)\hat{b}_3
\end{aligned}
\tag{A.14}
$$

and the derivative of A.14, the acceleration term,

$$
\begin{aligned}
\mathbf{a}_{CMw/CMf} &= {}^b\frac{d}{dt}\left(\mathbf{v}_{CMw/CMf}\right) + \boldsymbol{\omega}_{b/i} \times \left(\mathbf{v}_{CMw/CMf}\right) \\
&= -\left(\left(\ddot{\theta}+\ddot{\phi}\right)d_c\sin\phi + \left(\dot{\theta}+\dot{\phi}\right)d_c\cos\phi - d\dot{\theta}^2\right)\hat{b}_1 \\
&\quad - \left(d\ddot{\theta}+\left(\ddot{\theta}+\ddot{\phi}\right)d_c\cos\phi - \left(\dot{\theta}+\dot{\phi}\right)^2 d_c\sin\phi\right)\hat{b}_3
\end{aligned}
\tag{A.15}
$$

Therefore, combining Eqs. (A.15) and (A.11),

$$
\begin{aligned}
\mathbf{a}_{CMw/CMs} &= -\frac{m_f}{m}\left(\left(\left(\ddot{\theta}+\ddot{\phi}\right)d_c\sin\phi + \left(\dot{\theta}+\dot{\phi}\right)d_c\cos\phi - d\dot{\theta}^2\right)\hat{b}_1 \right. \\
&\quad \left. + \left(d\ddot{\theta}+\left(\ddot{\theta}+\ddot{\phi}\right)d_c\cos\phi - \left(\dot{\theta}+\dot{\phi}\right)^2 d_c\sin\phi\right)\hat{b}_3\right)
\end{aligned}
\tag{A.16}
$$

And, combining Eqs. (A.15) and (A.12),

$$
\begin{aligned}
\mathbf{a}_{CMf/CMs} &= \frac{m_w}{m}\left(\left(\left(\ddot{\theta}+\ddot{\phi}\right)d_c\sin\phi + \left(\dot{\theta}+\dot{\phi}\right)d_c\cos\phi - d\dot{\theta}^2\right)\hat{b}_1 \right. \\
&\quad \left. + \left(d\ddot{\theta}+\left(\ddot{\theta}+\ddot{\phi}\right)d_c\cos\phi - \left(\dot{\theta}+\dot{\phi}\right)^2 d_c\sin\phi\right)\hat{b}_3\right)
\end{aligned}
\tag{A.17}
$$

The inertial velocity of the CM for the system is defined as

$$
\mathbf{v}_{CMs/i} = u\hat{b}_1 + w\hat{b}_3
\tag{A.18}
$$

with the time derivative,

$$
\begin{aligned}
\mathbf{a}_{CMs/i} &= {}^b\frac{d}{dt}\left(\mathbf{v}_{CMs/i}\right) + \boldsymbol{\omega}_{b/i} \times \left(\mathbf{v}_{CMs/i}\right) \\
&\quad \left(\dot{u}+w\dot{\theta}\right)\hat{b}_1 + \left(\dot{w}+u\dot{\theta}\right)\hat{b}_3
\end{aligned}
\tag{A.19}
$$

Therefore, using Eqs. (A.16) and (A.17), the inertial velocities of the fuselage and

wing are

$$\mathbf{a}_{CMw/i} = \left(\dot{u} + w\dot{\theta} - \frac{m_f}{m}\left(\left(\ddot{\theta} + \ddot{\phi}\right)d_c\sin\phi + \left(\dot{\theta} + \dot{\phi}\right)d_c\cos\phi - d\dot{\theta}^2\right)\right)\hat{b_1}$$
$$\left(\dot{w} - u\dot{\theta} - \frac{m_f}{m}\left(d\ddot{\theta} + \left(\ddot{\theta} + \ddot{\phi}\right)d_c\cos\phi - \left(\dot{\theta} + \dot{\phi}\right)^2 d_c\sin\phi\right)\right)\hat{b_3}$$

(A.20)

and

$$\mathbf{a}_{CMf/i} = \left(\dot{u} + w\dot{\theta} + \frac{m_w}{m}\left(\left(\ddot{\theta} + \ddot{\phi}\right)d_c\sin\phi + \left(\dot{\theta} + \dot{\phi}\right)d_c\cos\phi - d\dot{\theta}^2\right)\right)\hat{b_1}$$
$$\left(\dot{w} - u\dot{\theta} + \frac{m_w}{m}\left(d\ddot{\theta} + \left(\ddot{\theta} + \ddot{\phi}\right)d_c\cos\phi - \left(\dot{\theta} + \dot{\phi}\right)^2 d_c\sin\phi\right)\right)\hat{b_3}$$

(A.21)

Recall, from Eq. (A.1), that the dot product of the inertial acceleration terms for each body is needed. For simplicity, let

$$\mathbf{S} = \mathbf{S}_w + \mathbf{S}_f$$

(A.22)

where the subscripts w and f represent the wing and fuselage bodies respectively.

$$\mathbf{S}_w = \frac{1}{2}m_w\mathbf{a}_{w/i} \cdot \mathbf{a}_{w/i} + \frac{1}{2}\left(\boldsymbol{\alpha} \cdot \dot{\mathbf{H}}_w\right)$$

(A.23)

The angular momentum for the wing about its CM is defined as

$$\mathbf{H}_w = I_w\left(\dot{\theta} + \dot{\phi}\right)\hat{b_2}$$

(A.24)

taking the time derivative

$$\dot{\mathbf{H}}_w = I_w\left(\ddot{\theta} + \ddot{\phi}\right)\hat{b_2}$$

(A.25)

where the angular acceleration for the wing is identified as

$$\boldsymbol{\alpha}_w = \left(\ddot{\theta} + \ddot{\phi}\right)\hat{b_2}$$

(A.26)

Now using Eq. (A.23), the energy of acceleration for the wing in the $\hat{b_2}$ direction

is

$$
\begin{aligned}
\mathbf{S}_w &= \frac{1}{2} m_w \mathbf{a}_{w/i} \cdot \mathbf{a}_{w/i} + \frac{1}{2} \left(\boldsymbol{\alpha}_w \cdot \dot{\mathbf{H}}_w \right) \\
&= \frac{1}{2} m_w \left(\dot{u}^2 + 2\dot{u}w\dot{\theta} - 2\frac{m_f}{m}\dot{u}\left(d_c\left(\ddot{\theta}+\ddot{\phi}\right)\sin\phi \right. \right. \\
&\quad \left. + d_c\left(\dot{\theta}+\dot{\phi}\right)^2\cos\phi + d\dot{\theta}^2 \right) + w^2\dot{\theta}^2 \\
&\quad - 2w\dot{\theta}\frac{m_f}{m}\left(d_c\left(\ddot{\theta}+\ddot{\phi}\right)\sin\phi + d_c\left(\dot{\theta}+\dot{\phi}\right)^2\cos\phi + d\dot{\theta}^2 \right) \\
&\quad + \left(\frac{m_f}{m}\right)^2 d_c^2\left(\ddot{\theta}+\ddot{\phi}\right) + 2\left(\frac{m_f}{m}\right)^2 d_c d\dot{\theta}^2\left(\ddot{\theta}+\ddot{\phi}\right)\sin\phi \\
&\quad + 2\left(\frac{m_f}{m}\right)^2 d_c^2\left(\dot{\theta}+\dot{\phi}\right)^4 + 2\left(\frac{m_f}{m}\right)^2 d_c d\dot{\theta}^2\left(\dot{\theta}+\dot{\phi}\right)^2\cos\phi \\
&\quad + \left(\frac{m_f}{m}\right)^2 d^2\dot{\theta}^4 + \dot{w}^2 - 2\dot{w}u\dot{\theta} - 2\left(\frac{m_f}{m}\right)^2 d_c d\ddot{\theta}\left(\dot{\theta}+\dot{\phi}\right)^2\sin\phi \\
&\quad - 2\frac{m_f}{m}\dot{w}\left(d\ddot{\theta} + d_c\left(\ddot{\theta}+\ddot{\phi}\right)\cos\phi - d_c\left(\dot{\theta}+\dot{\phi}\right)^2\sin\phi \right) \\
&\quad + u^2\dot{\theta}^2 + 2\frac{m_f}{m}u\dot{\theta}\left(d\ddot{\theta} + d_c\left(\ddot{\theta}+\ddot{\phi}\right)\cos\phi - d_c\left(\dot{\theta}+\dot{\phi}\right)^2\sin\phi \right) \\
&\quad \left(\frac{m_f}{m}\right)^2 d^2\ddot{\theta}^2 + 2\left(\frac{m_f}{m}\right)^2 d_c d\ddot{\theta}\left(\ddot{\theta}+\ddot{\phi}\right)\cos\phi + \frac{1}{2} I_w\left(\ddot{\theta}+\ddot{\phi}\right)^2
\end{aligned}
$$

$$(\text{A.27})$$

The energy of acceleration for the fuselage in the $\hat{b}_2$ direction is

$$
\mathbf{S}_f = \frac{1}{2} m_f \mathbf{a}_{f/i} \cdot \mathbf{a}_{f/i} + \frac{1}{2} \left(\boldsymbol{\alpha} \cdot \dot{\mathbf{H}}_f \right) \tag{A.28}
$$

The angular momentum for the fuselage about its CM is defined as

$$
\mathbf{H}_f = I_f \dot{\theta} \hat{b}_2 \tag{A.29}
$$

with the time derivative

$$
\dot{\mathbf{H}}_f = I_f \ddot{\theta} \hat{b}_2 \tag{A.30}
$$

The angular acceleration for the fuselage is

$$
\boldsymbol{\alpha}_f = \dddot{\theta} \hat{b}_2 \tag{A.31}
$$

Therefore, using Eqs. (A.21) - A.31 and (A.28), the energy of acceleration for the fuselage in the $\hat{b}_2$ direction is

$$
\begin{aligned}
\mathbf{S}_f ={}& \frac{1}{2}m_f \mathbf{a}_{f/i} \boldsymbol{.} \mathbf{a}_{f/i} + \frac{1}{2}\left(\boldsymbol{\alpha}_f \boldsymbol{.} \dot{\mathbf{H}}_f\right) \\
={}& \frac{1}{2}m_f\Big(\dot{u}^2 + 2\dot{u}w\dot{\theta} + 2\frac{m_w}{m}\dot{u}d_c\left(\ddot{\theta}+\ddot{\phi}\right)\sin\phi \\
& + 2\dot{u}\frac{m_w}{m}d_c\left(\dot{\theta}+\dot{\phi}\right)^2\cos\phi + 2\dot{u}\frac{m_w}{m}d\dot{\theta}^2 + w^2\dot{\theta}^2 + 2w\dot{\theta}\frac{m_w}{m}d\dot{\theta}^2 \\
& + 2w\dot{\theta}\frac{m_w}{m}d_c\left(\ddot{\theta}+\ddot{\phi}\right)\sin\phi + 2w\dot{\theta}\frac{m_w}{m}d_c\left(\dot{\theta}+\dot{\phi}\right)^2\cos\phi + \left(\frac{m_w}{m}\right)^2 d_c^2\left(\ddot{\theta}+\ddot{\phi}\right) \\
& + 2\left(\frac{m_w}{m}\right)^2 d_cd\dot{\theta}^2\left(\ddot{\theta}+\ddot{\phi}\right)\sin\phi + \left(\frac{m_w}{m}\right)^2 d_c^2\left(\dot{\theta}+\dot{\phi}\right)^4 \\
& + 2\left(\frac{m_w}{m}\right)^2 d_cd\dot{\theta}^2\left(\dot{\theta}+\dot{\phi}\right)^2\cos\phi + \left(\frac{m_w}{m}\right)^2 d^2\dot{\theta}^4 + \dot{w}^2 - 2\dot{w}u\dot{\theta} \\
& + 2\frac{m_w}{m}\dot{w}d\ddot{\theta} + 2\dot{w}\frac{m_w}{m}d_c\left(\ddot{\theta}+\ddot{\phi}\right)\cos\phi - 2\dot{w}\frac{m_w}{m}d_c\left(\dot{\theta}+\dot{\phi}\right)^2\sin\phi \\
& + u^2\dot{\theta}^2 - 2u\dot{\theta}]fracm_wmd\ddot{\theta} - 2\frac{m_w}{m}u\dot{\theta}d_c\left(\ddot{\theta}+\ddot{\phi}\right)\cos\phi \\
& + 2u\dot{\theta}d_c\frac{m_w}{m}\left(\dot{\theta}+\dot{\phi}\right)^2\sin\phi + \left(\frac{m_w}{m}\right)^2 d^2\ddot{\theta}^2 \\
& + 2\left(\frac{m_w}{m}\right)^2 d_cd\ddot{\theta}\left(\ddot{\theta}+\ddot{\phi}\right)\cos\phi - 2\left(\frac{m_w}{m}\right)^2 d_cd\ddot{\theta}\left(\dot{\theta}+\dot{\phi}\right)^2\sin\phi\Big) + \frac{1}{2}I_f\ddot{\theta}^2
\end{aligned}
$$

$$(A.32)$$

The partial derivatives of S_w with respect to each of the generalized accelerations are

$$
\frac{\partial \mathbf{S}_w}{\partial \dot{u}} = m_w\left(\dot{u} + w\dot{\theta} - \frac{m_f}{m}d_c\left(\ddot{\theta}+\ddot{\phi}\right)\sin\phi - \frac{m_f}{m}d_c\left(\dot{\theta}+\dot{\phi}\right)^2\cos\phi - \frac{m_f}{m}d\dot{\theta}^2\right)
$$

$$(A.33)$$

$$
\frac{\partial \mathbf{S}_w}{\partial \dot{w}} = m_w\left(\dot{w} - u\dot{\theta} - \frac{m_f}{m}d_c\left(\ddot{\theta}+\ddot{\phi}\right)\cos\phi + \frac{m_f}{m}d_c\left(\dot{\theta}+\dot{\phi}\right)^2\sin\phi - \frac{m_f}{m}d\ddot{\theta}\right) \quad (A.34)
$$

$$
\begin{aligned}
\frac{\partial \mathbf{S}_w}{\partial \ddot{\theta}} = m_w\Big(& -\frac{m_f}{m}\left(\dot{u}d_c\sin\phi + \dot{w}d - u\dot{\theta}d - u\dot{\theta}d_c\cos\phi\right. \\
& \left. + w\dot{\theta}d_c\sin\phi + \dot{w}d_c\cos\phi\right) + \left(\frac{m_f}{m}\right)^2\left(d_c^2\left(\ddot{\theta}+\ddot{\phi}\right)\right. \\
& \left. - dd_c\left(2\dot{\theta}\dot{\phi}+\dot{\phi}^2\right)\sin\phi + dd_c\left(2\ddot{\theta}+\ddot{\phi}\right)\cos\phi\right)\Big) + I_w\ddot{\theta}
\end{aligned}
$$

$$(A.35)$$

and

$$\frac{\partial \mathbf{S}_w}{\partial \ddot{\phi}} = m_w \Big(-\frac{m_f}{m} \Big(\dot{w} d_c \cos \phi - u \dot{\theta} d_c \cos \phi$$
$$+ w \dot{\theta} d_c \sin \phi + \dot{u} d_c \sin \phi \Big) + \Big(\frac{m_f}{m} \Big)^2 \Big(d_c^2 \big(\ddot{\theta} + \ddot{\phi} \big) \qquad \text{(A.36)}$$
$$+ d d_c \dot{\theta}^2 \sin \phi + d d_c \ddot{\theta} \cos \phi \Big) \Big) + I_w \ddot{\phi}$$

And, the partial derivatives of $\mathbf{S}_f$ with respect to each of the generalized accelerations are

$$\frac{\partial \mathbf{S}_f}{\partial \dot{u}} = m_f \left(\dot{u} + w\dot{\theta} + \frac{m_w}{m} d_c \big(\ddot{\theta} + \ddot{\phi} \big) \sin \phi + \frac{m_w}{m} d_c \big(\dot{\theta} + \dot{\phi} \big)^2 \cos \phi + \frac{m_w}{m} d\dot{\theta}^2 \right)$$

$$\text{(A.37)}$$

$$\frac{\partial \mathbf{S}_f}{\partial \dot{w}} = m_f \left(\dot{w} - u\dot{\theta} + \frac{m_w}{m} d_c \big(\ddot{\theta} + \ddot{\phi} \big) \cos \phi - \frac{m_w}{m} d_c \big(\dot{\theta} + \dot{\phi} \big)^2 \sin \phi + \frac{m_w}{m} d\ddot{\theta} \right) \quad \text{(A.38)}$$

$$\frac{\partial \mathbf{S}_f}{\partial \ddot{\theta}} = m_f \Big(\frac{m_w}{m} \Big(\dot{u} d_c \sin \phi + \dot{w} d - u\dot{\theta} d - u\dot{\theta} d_c \cos \phi$$
$$+ w \dot{\theta} d_c \sin \phi + \dot{w} d_c \cos \phi \Big) + \Big(\frac{m_w}{m} \Big)^2 \Big(d_c^2 \big(\ddot{\theta} + \ddot{\phi} \big) \qquad \text{(A.39)}$$
$$- d d_c \big(2\dot{\theta}\dot{\phi} + \dot{\phi}^2 \big) \sin \phi + d d_c \big(2\ddot{\theta} + \ddot{\phi} \big) \cos \phi \Big) \Big) + I_f \ddot{\theta}$$

and

$$\frac{\partial \mathbf{S}_f}{\partial \ddot{\phi}} = m_f \Big(\frac{m_w}{m} \Big(\dot{w} d_c \cos \phi - u\dot{\theta} d_c \cos \phi$$
$$+ w \dot{\theta} d_c \sin \phi + \dot{u} d_c \sin \phi \Big) + \Big(\frac{m_w}{m} \Big)^2 \Big(d_c^2 \big(\ddot{\theta} + \ddot{\phi} \big) \qquad \text{(A.40)}$$
$$+ d d_c \dot{\theta}^2 \sin \phi + d d_c \ddot{\theta} \cos \phi \Big) \Big)$$

Obtain the total partial derivatives for the energy of acceleration of the system with respect to the generalized accelerations by adding the respective components from the partial derivatives obtained from the wing and fuselage in Eqs. (A.33) - (A.40).

$$\frac{\partial \mathbf{S}}{\partial \dot{u}} = m \big(\dot{u} + w\dot{\theta} \big) \qquad \text{(A.41)}$$

$$\frac{\partial \mathbf{S}}{\partial \dot{w}} = m\left(\dot{w} - u\dot{\theta}\right) \tag{A.42}$$

$$\begin{aligned}
\frac{\partial \mathbf{S}}{\partial \ddot{\theta}} = \frac{m_f m_w}{m}\left(d_c^2\left(\ddot{\theta} + \ddot{\phi}\right) - dd_c\left(2\dot{\theta}\dot{\phi}\right.\right. \\
\left.\left. + \dot{\phi}^2\right)\sin\phi + dd_c\left(2\ddot{\theta} + \ddot{\phi}\right)\cos\phi\right) + I_f\ddot{\theta} + I_w\left(\ddot{\theta} + \ddot{\phi}\right)
\end{aligned} \tag{A.43}$$

$$\begin{aligned}
\frac{\partial \mathbf{S}}{\partial \ddot{\phi}} = \frac{m_f m_w}{m}\left(d_c^2\left(\ddot{\theta} + \ddot{\phi}\right) + dd_c\dot{\theta}^2\sin\phi\right. \\
\left. + dd_c\ddot{\theta}\cos\phi\right) + I_w\left(\ddot{\theta} + \ddot{\phi}\right)
\end{aligned} \tag{A.44}$$

Note, several terms in $\frac{\partial S_w}{\partial \dot{u}_i}$ and $\frac{\partial S_f}{\partial \dot{u}_i}$ cancel each other. With $\frac{\partial S}{\partial \dot{u}_i}$, the generalized forces U_i must be derived.

B. Generalized Forces

The generalized forces are determined using the virtual work equation,

$$\begin{aligned}
\overline{\delta W} &= \sum F_j \boldsymbol{.} \partial \mathbf{p}_j + \sum M_k \partial \alpha_k \\
&= \sum U_i \, \partial \mathbf{u}_i
\end{aligned} \tag{A.45}$$

In the above equation, α_k is representative of the angle of rotation for the body, not to be confused with angle of attack. Also, the notation uses the subscript $j = 1 : n$ where 'n' equals the number of forces acting on the system and $k = 1 : o$ where 'o' equals the number of moments acting on the system. Note, that the $\partial \mathbf{p}_j$ term is derived from the inertial velocity vector for the force. Time derivatives in the velocity vector, such as $\dot{\theta}$, are replaced with their respective partial, such as $\partial \theta$.

Note that there are six forces acting on the system and two moments. The six forces are the lift and drag for each the wing and the tail, weight of the system, and thrust. The two moments are from the torsional spring and the zero-lift aerodynamic moment, if it exists.

Let the equivalent of

$$\dot{\theta} = \partial\theta \tag{A.46}$$

for the generalized velocities be represented as

$$u = \partial\tilde{u} \tag{A.47}$$

and

$$w = \partial\tilde{w} \tag{A.48}$$

Inertial velocity vector for the weight of the system,

$$\mathbf{v}_{CMs/i} = u\hat{b}_1 + w\hat{b}_3 \tag{A.49}$$

$$\mathbf{v}_{CMs/i} = \partial\tilde{u}\hat{b}_1 + \partial\tilde{w}\hat{b}_3 \tag{A.50}$$

Inertial velocity vector for the aerodynamic center of the wing,

$$
\begin{aligned}
\mathbf{v}_{ACw/i} &= \mathbf{v}_{ACw/CMw} + \mathbf{v}_{CMw/CMs} + \mathbf{v}_{CMs/i} \\
&= \left(u - \left(\dot{\theta} + \dot{\phi}\right)\left(e - \frac{m_w}{m}d_c\right)\sin\phi \right)\hat{b}_1 \\
&+ \left(w - \left(\dot{\theta} + \dot{\phi}\right)\left(e - \frac{m_w}{m}d_c\right)\cos\phi - \frac{m_f}{m}d\dot{\theta} \right)\hat{b}_3
\end{aligned} \tag{A.51}
$$

$$
\begin{aligned}
\partial\mathbf{p}_{CMf/i} &= \left(\partial\tilde{u} - (\partial\theta + \partial\phi)\left(e - \frac{m_w}{m}d_c\right)\sin\phi \right)\hat{b}_1 \\
&+ \left(\partial\tilde{w} - (\partial\theta + \partial\phi)\left(e - \frac{m_w}{m}d_c\right)\cos\phi - \frac{m_f}{m}d\partial\theta \right)\hat{b}_3
\end{aligned} \tag{A.52}
$$

Since thrust is aligned with the body frame, it does not matter which point is selected if it is aligned with the $\hat{b}_1$ direction. Also, note that the lift and drag on the tail are assumed to act through the attachment point for the tail since the moment of these loads about the CM of the system is large compared to the moment about

the tail attachment. The inertial velocity vector for the attachment point of the tail,

$$\mathbf{v}_{t/i} = \mathbf{v}_{t/CMf} + \mathbf{v}_{CMf/CMs} + \mathbf{v}_{CMs/i}$$

$$= \left(u + \frac{m_w}{m}\left(\dot{\theta} + \dot{\phi}\right) d_c \sin\phi\right)\hat{b}_1 \tag{A.53}$$

$$+ \left(w + \frac{m_w}{m}\left(\dot{\theta} + \dot{\phi}\right) d_c \cos\phi + \left(\frac{m_w}{m}d + f\right)\dot{\theta}\right)\hat{b}_3$$

$$\partial\mathbf{p}_{t/i} = \left(\partial\tilde{u} + \frac{m_w}{m}\left(\partial\theta + \partial\phi\right) d_c \sin\phi\right)\hat{b}_1 \tag{A.54}$$

$$+ \left(\partial\tilde{w} + \frac{m_w}{m}\left(\partial\theta + \partial\phi\right) d_c \cos\phi + \left(\frac{m_w}{m}d + f\right)\partial\theta\right)\hat{b}_3$$

Now, the respective moments are defined.

For the weight of the system,

$$mg\hat{i_3} \cdot \partial\mathbf{p}_{CMs/i} = mg\left(\cos\theta\partial\tilde{w} - \sin\theta\partial\tilde{u}\right) \tag{A.55}$$

For the lift of the wing,

$$L_w\left(-\hat{s}_3\right) \cdot \partial\mathbf{p}_{ACw/i} = L_w\Big(\partial\tilde{u}\sin\left(\theta - \gamma_w\right) - \partial\tilde{w}\cos\left(\theta - \gamma_w\right)$$

$$+ \left(\partial\theta + \partial\phi\right)\left(e - \frac{m_w}{m}d_c\right)\cos\alpha_w + \frac{m_f}{m}d\partial\theta\cos\left(\theta - \gamma_w\right)\Big) \tag{A.56}$$

For the drag of the wing,

$$D_w\left(-\hat{s}_1\right) \cdot \partial\mathbf{p}_{ACw/i} = D_w\Big(-\partial\tilde{u}\cos\left(\theta - \gamma_w\right) - \partial\tilde{w}\sin\left(\theta - \gamma_w\right)$$

$$+ \left(\partial\theta + \partial\phi\right)\left(e - \frac{m_w}{m}d_c\right)\sin\alpha_w + \frac{m_f}{m}d\partial\theta\sin\left(\theta - \gamma_w\right)\Big) \tag{A.57}$$

For the lift of the tail,

$$L_t\left(-\hat{s}_3\right) \cdot \partial\mathbf{p}_{t/i} = L_t\Big(\partial\tilde{u}\sin\left(\theta - \gamma_t\right) - \partial\tilde{w}\cos\left(\theta - \gamma_t\right)$$

$$- \left(\partial\theta + \partial\phi\right)\frac{m_w}{m}d_c\cos\alpha_t - \left(f + \frac{m_f}{m}d\right)\partial\theta\cos\left(\theta - \gamma_t\right)\Big) \tag{A.58}$$

For the drag of the tail,

$$D_t\left(-\hat{s}_1\right) \cdot \partial \mathbf{p}_{t/i} = D_t\Big(-\partial\tilde{u}\cos\left(\theta-\gamma_t\right) - \partial\tilde{w}\sin\left(\theta-\gamma_t\right)$$
$$- \left(\partial\theta + \partial\phi\right)\frac{m_w}{m}d_c\sin\alpha_t - \left(f + \frac{m_f}{m}d\right)\partial\theta\sin\left(\theta-\gamma_t\right)\Big) \tag{A.59}$$

For the thrust of the vehicle,

$$T\hat{b}_1 \cdot \partial \mathbf{p}_{t/i} = T\left(\partial\tilde{u} + \frac{m_w}{m}d_c\sin\phi\left(\partial\theta + \partial\phi\right)\right) \tag{A.60}$$

For the torsional spring moment,

$$M\partial\alpha = -k\left(\phi - \alpha_r\right)\partial\phi \tag{A.61}$$

And, for the aerodynamic moment of the wing, if it exists at zero lift,

$$M\partial\alpha = M_{ac}\left(\partial\theta + \partial\phi\right) \tag{A.62}$$

Next, combining terms into Eq. (A.45), one obtains the four generalized forces.

$$U_u = -mg\sin\theta + L_w\sin\left(\theta-\gamma_w\right) - D_w\cos\left(\theta-\gamma_w\right)$$
$$+ L_t\sin\left(\theta-\gamma_t\right) - D_t\cos\left(\theta-\gamma_t\right) + T \tag{A.63}$$

$$U_w = mg\cos\theta - L_w\cos\left(\theta-\gamma_w\right) - D_w\sin\left(\theta-\gamma_w\right)$$
$$+ L_t\cos\left(\theta-\gamma_t\right) - D_t\sin\left(\theta-\gamma_t\right) \tag{A.64}$$

$$U_{\dot{\theta}} = L_w\left(\left(e - \frac{m_w}{m}d_c\right)\cos\alpha_w + \frac{m_f}{m}d\cos\left(\theta-\gamma_w\right)\right)$$
$$+ D_w\left(\left(e - \frac{m_w}{m}d_c\right)\sin\alpha_w + \frac{m_f}{m}d\sin\left(\theta-\gamma_w\right)\right)$$
$$- L_t\left(\frac{m_w}{m}d_c\cos\alpha_t + \left(f + \frac{m_w}{m}d\right)\cos\left(\theta-\gamma_t\right)\right)$$
$$- D_t\left(\frac{m_w}{m}d_c\sin\alpha_t + \left(f + \frac{m_w}{m}d\right)\sin\left(\theta-\gamma_t\right)\right)$$
$$+ T\frac{m_w}{m}d_c\sin\phi + M_{ac} \tag{A.65}$$

and

$$
\begin{aligned}
U_{\dot{\phi}} = {} & L_w \left(e - \frac{m_w}{m} d_c \right) \cos \alpha_w + D_w \left(e - \frac{m_w}{m} d_c \right) \sin \alpha_w \\
& + L_t \frac{m_w}{m} d_c \cos \alpha_t - D_t \frac{m_w}{m} d_c \sin \alpha_t \\
& + T \frac{m_w}{m} d_c \sin \phi + M_{ac} - k \left(\phi - \alpha_r \right)
\end{aligned}
\tag{A.66}
$$

C.   Gibbs-Appell Equations of Motion

The equations of motion are derived using

$$
\frac{\partial S}{\dot{u}_i} = U_i
\tag{A.67}
$$

Thus, the equations of motion from Gibbs-Appell are:

$$
\begin{aligned}
m \left(\dot{u} + w\dot{\theta} \right) = {} & -mg \sin \theta + L_w \sin \left(\theta - \gamma_w \right) - D_w \cos \left(\theta - \gamma_w \right) \\
& + L_t \sin \left(\theta - \gamma_t \right) - D_t \cos \left(\theta - \gamma_t \right) + T
\end{aligned}
\tag{A.68}
$$

,

$$
\begin{aligned}
m \left(\dot{w} - u\dot{\theta} \right) = {} & mg \cos \theta - L_w \cos \left(\theta - \gamma_w \right) - D_w \sin \left(\theta - \gamma_w \right) \\
& + L_t \cos \left(\theta - \gamma_t \right) - D_t \sin \left(\theta - \gamma_t \right)
\end{aligned}
\tag{A.69}
$$

,

$$
\begin{aligned}
I_f \ddot{\theta} + I_w \left(\ddot{\theta} + \ddot{\phi} \right) & + \frac{m_f m_w}{m} \left(d_c^2 \left(\ddot{\theta} + \ddot{\phi} \right) - d d_c \left(2\dot{\theta}\dot{\phi} + \dot{\phi}^2 \right) \sin \phi \right. \\
& \left. + d d_c \left(2\ddot{\theta} + \ddot{\phi} \right) \cos \phi \right) = T \frac{m_w}{m} d_c \sin \phi + M_{ac} \\
& + L_w \left(\left(e - \frac{m_w}{m} \right) \cos \alpha_w + \frac{m_f}{m} d \cos \left(\theta - \gamma_w \right) \right) \\
& + D_w \left(\left(e - \frac{m_w}{m} \right) \sin \alpha_w + \frac{m_f}{m} d \sin \left(\theta - \gamma_w \right) \right) \\
& - L_t \left(\frac{m_w}{m} d_c \cos \alpha_t + \left(f + \frac{m_w}{m} d \right) \cos \left(\theta - \gamma_t \right) \right) \\
& - D_t \left(\frac{m_w}{m} d_c \sin \alpha_t + \left(f + \frac{m_w}{m} d \right) \sin \left(\theta - \gamma_t \right) \right)
\end{aligned}
\tag{A.70}
$$

, and

$$
\begin{aligned}
I_w \left(\ddot{\theta} + \ddot{\phi} \right) + \frac{m_w m_f}{m} & \left(d_c^2 \left(\ddot{\theta} + \ddot{\phi} \right) + d d_c \ddot{\theta} \cos \phi + d d_c \dot{\theta}^2 \sin \phi \right) = \\
& - k \left(\phi - \alpha_r \right) + L_w \left(e - \frac{m_w}{m} d_c \right) \cos \alpha_w + D_w \left(e - \frac{m_w}{m} d_c \right) \sin \alpha_w \\
& - L_t \frac{m_w}{m} d_c \cos \alpha_t - D_t \frac{m_w}{m} d_c \sin \alpha_t \\
& + T \frac{m_w}{m} d_c \sin \phi + M_{ac}
\end{aligned}
\tag{A.71}
$$

APPENDIX B

NEWTON-EULER DERIVATION

The Newton-Euler derivations of the equations of motion use Newton's Second Law for the derivation of the translational equations of motion and Euler's equation for a rotating body for the derivation of the rotational equations of motion.

A.   Translational Equations of Motion

Newton's Second law states

$$\sum \mathbf{F} = m\mathbf{a}_{CM_s/i} \tag{B.1}$$

where $\mathbf{F}$ represents the forces on the system.

Recall, that in Appendix A that the inertial accelerations for the wing and fuselage are derived, refer to Eqs. (A.20) and (A.21). Equation (B.1), using the wing and fuselage components is written as,

$$\sum \mathbf{F} = m_f\mathbf{a}_{CM_f/i} + m_w\mathbf{a}_{CM_w/i} \tag{B.2}$$

Since the inertial accelerations of the fuselage and wing are obtained in Appendix A, the only information needed is the sum of the forces. There are six forces that act on the system: lift and drag due to each the wing and the tail, weight of the system, and thrust of the vehicle. See the free body diagram for the system in Fig. 22.
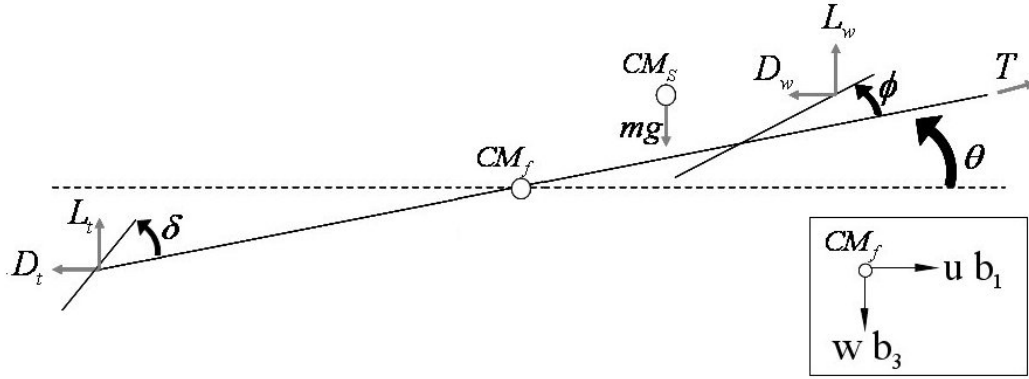
Fig. 22. System Free Body Diagram

Therefore,

$$\sum \mathbf{F} = mg\hat{i_3} + T\hat{b_1} + L_w\left(-\hat{s_{w3}}\right) + D_w\left(-\hat{s_{w1}}\right)$$
$$+ L_t\left(-\hat{s_{t3}}\right) + D_t\left(-\hat{s_{t1}}\right) \tag{B.3}$$

Transforming these equations so that they are in the body frame,

$$\sum \mathbf{F} = \Big(T - mg\sin\theta + L_w\sin\left(\theta - \gamma_w\right) + L_t\sin\left(\theta - \gamma_t\right)$$
$$- D_w\cos\left(\theta - \gamma_w\right) - D_t\cos\left(\theta - \gamma_t\right)\Big)\hat{b_1}$$
$$+ \Big(mg\cos\theta - L_w\cos\left(\theta - \gamma_w\right) - L_t\cos\left(\theta - \gamma_t\right)$$
$$- D_w\sin\left(\theta - \gamma_w\right) - D_t\sin\left(\theta - \gamma_t\right)\Big)\hat{b_3} \tag{B.4}$$

Therefore, using Eqs. (A.20), (A.21), and (B.4), the Newton's equations of motion for the system in the body frame are obtained.

Translational Equation of Motion (in the $\hat{b_1}$ frame):

$$m\left(\dot{u} + w\dot{\theta}\right) = T - mg\sin\theta + L_w\sin\left(\theta - \gamma_w\right) + L_t\sin\left(\theta - \gamma_t\right)$$
$$- D_w\cos\left(\theta - \gamma_w\right) - D_t\cos\left(\theta - \gamma_t\right) \tag{B.5}$$

Translational Equation of Motion (in the $\hat{b}_3$ frame):

$$m\left(\dot{w} - u\dot{\theta}\right) = mg\cos\theta - L_w\cos\left(\theta - \gamma_w\right) - L_t\cos\left(\theta - \gamma_t\right)$$
$$- D_w\sin\left(\theta - \gamma_w\right) - D_t\sin\left(\theta - \gamma_t\right)$$

(B.6)

B. Rotational Equations of Motion

To develop the rotational equations of motion, Euler's equation for a rotational body about an arbitrary point, '*', are used[21].

$$\mathbf{H}_* = \mathbf{H}_{CM} + m\left(\mathbf{r}_{CM/i} - r_{*/i}\right) \times \left(\mathbf{v}_{CM/i} - \mathbf{v}_{*/i}\right) \tag{B.7}$$

$$\dot{\mathbf{H}}_* = \mathbf{l}_* + m\mathbf{a}_{*/i} \times \left(\mathbf{r}_{CM/i} - \mathbf{r}_{*/i}\right) \tag{B.8}$$

where, $\mathbf{H}$ is the angular momentum, CM is the center of mass for each individual body, $\mathbf{r}$ is the inertial position, $\mathbf{v}$ is the inertial velocity, and $\mathbf{l}$ is the sum of moments on the system.

1. System Rotational Equation of Motion

For the system rotational equation of motion, the arbitrary point, '*', is chosen as the CM of the system, which reduces Eq. (B.8) to

$$\dot{\mathbf{H}}_{CM} = \mathbf{l}_{CM} + m\mathbf{a}_{*/i} \times \left(\mathbf{r}_{CM/i} - \mathbf{r}_{*/i}\right) \tag{B.9}$$

where $\mathbf{r}_{CM/i} - \mathbf{r}_{*/i} = 0$ at the CM of the system.

The total angular momentum for the full system is determined by adding the contributions from wing and fuselage angular momentums about the system CM, refer to Fig. 22 for free body diagram. Recall, the velocities of the CM of the wing and fuselage with respect to the CM of the system are derived in Appendix A, refer

to Eqs. (A.9) and (A.10). Thus, the angular momentum is

$$\begin{aligned}
\mathbf{H}_{CMw/CMs} &= \mathbf{H}_w + m_w \left(\mathbf{r}_{CMw/CMs} \times \mathbf{v}_{CMw/CMs} \right) \\
&= I_w \left(\dot{\theta} + \dot{\phi} \right) + m_w \left(\frac{m_f}{m} \right)^2 \left(\mathbf{r}_{CMf/CMw} \times \mathbf{v}_{CMf/CMw} \right)
\end{aligned} \tag{B.10}$$

and

$$\begin{aligned}
\mathbf{H}_{CMf/CMs} &= \mathbf{H}_f + m_f \left(\mathbf{r}_{CMf/CMs} \times \mathbf{v}_{CMf/CMs} \right) \\
&= I_f \dot{\theta} + m_f \left(\frac{m_w}{m} \right)^2 \left(\mathbf{r}_{CMf/CMw} \times \mathbf{v}_{CMf/CMw} \right)
\end{aligned} \tag{B.11}$$

Therefore, the total angular momentum, $\mathbf{H}_{CMt/CMs}$, of the system about its CM is

$$\mathbf{H}_{CMt/CMs} = \left(I_f \dot{\theta} + I_w \left(\dot{\theta} + \dot{\phi} \right) + \frac{m_w m_f}{m} \left(r_{CMf/CMw} \times \mathbf{v}_{CMf/CMw} \right) \right) \hat{b}_2 \tag{B.12}$$

Considering the relationships for $\mathbf{r}_{CMf/CMw}$ and $\mathbf{r}_{CMf/CMw}$ (see Eqs. (A.13) and (A.14)), one finds

$$\begin{aligned}
\mathbf{H}_{CMt/CMs} = \Big(I_f \dot{\theta} + I_w \left(\dot{\theta} + \dot{\phi} \right) + \frac{m_f m_w}{m} \Big(d^2 \dot{\theta} \\
+ \left(\dot{\theta} + \dot{\phi} \right) dd_c \cos \phi + dd_c \dot{\theta} \cos \phi + \left(\dot{\theta} + \dot{\phi} \right) d_c^2 \Big) \Big) \hat{b}_2
\end{aligned} \tag{B.13}$$

with the time derivative

$$\begin{aligned}
\dot{\mathbf{H}}_{CMt/CMs} = \Big(f \ddot{\theta} + I_w \left(\ddot{\theta} + \ddot{\phi} \right) + \frac{m_f m_w}{m} \Big(d^2 \ddot{\theta} + dd_c \left(2\ddot{\theta} + \ddot{\phi} \right) \cos \phi \\
- dd_c \left(2\dot{\theta}\dot{\phi} + \dot{\phi}^2 \right) \sin \phi + d_c^2 \left(\ddot{\theta} + \ddot{\phi} \right) \Big) \Big) \hat{b}_2
\end{aligned} \tag{B.14}$$

The moments on the system are determined. Note, as with the Gibbs-Appell formulation in Appendix A, the lift and drag of the wing are assumed to be at the aerodynamic center, and the vehicle thrust, and the lift and drag of the tail are assumed to be at the attachment point of the tail. There are 6 moments acting on the system from the lift and drag of the wing, lift and drag of the tail, vehicle thrust, and the aerodynamic center (due to an asymmetric airfoil).

Recall, the following relationship from the Gibbs-Appell formulation,

$$\mathbf{r}_{CMw/CMs} = \frac{m_f}{m} \mathbf{r}_{CMw/CMf} \tag{B.15}$$

and

$$\mathbf{r}_{CMf/CMs} = -\frac{m_w}{m} \mathbf{r}_{CMw/CMf} \tag{B.16}$$

where

$$\mathbf{r}_{CMw/CMf} = (d + d_c \cos\phi)\,\hat{b}_1 - d_c \sin\phi \hat{b}_3 \tag{B.17}$$

Moments arms from the forces to the CM of the system are

$$
\begin{aligned}
\mathbf{r}_{ACw/CMs} &= \mathbf{r}_{ACw/CMw} + \mathbf{r}_{CMw/CMs} \\
&= (e - d_c)\,\hat{w}_1 + \frac{m_f}{m}\left((d + d_c \cos\phi)\,\hat{b}_1 - d_c \sin\phi\right)\hat{b}_3 \\
&= \left(\left(e - \frac{m_w}{m}d_c\right)\cos\phi + \frac{m_f}{m}d\cos\phi\right)\hat{b}_1 \\
&\quad - \left(e - \frac{m_w}{m}d_c\right)\sin\phi\hat{b}_3
\end{aligned} \tag{B.18}
$$

$$
\begin{aligned}
\mathbf{r}_{t/CMs} &= \mathbf{r}_{t/CMf} + \mathbf{r}_{CMf/CMs} \\
&= -\left(f + \frac{m_w}{m}(d + d_c \cos\phi)\right)\hat{b}_1 + \frac{m_w}{m}(d_c \sin\phi)\,\hat{b}_3
\end{aligned} \tag{B.19}
$$

Now, all respective moments are determined.

For the lift on the wing,

$$\mathbf{r}_{ACw/CMs} \times L_w\,(-\hat{s_{w3}}) = L_w\left(\left(e - \frac{m_w}{m}d_c\right)\cos\alpha_w + \frac{m_f}{m}d\cos\left(\theta - \gamma_w\right)\right)\hat{b}_2 \tag{B.20}$$

For the drag on the wing,

$$\mathbf{r}_{ACw/CMs} \times D_w\,(-\hat{s_{w1}}) = D_w\left(\left(e - \frac{m_w}{m}d_c\right)\sin\alpha_w + \frac{m_f}{m}d\sin\left(\theta - \gamma_w\right)\right)\hat{b}_2 \tag{B.21}$$

For the lift on the tail,

$$\mathbf{r}_{t/CMs} \times L_t \left(-\hat{s}_{t3}\right) = -L_t \left(\frac{m_w}{m} d_c \cos \alpha_t + \left(f + \frac{m_w}{m} d\right) \cos \left(\theta - \gamma_t\right)\right) \hat{b}_2 \qquad \text{(B.22)}$$

For the drag on the tail,

$$\mathbf{r}_{t/CMs} \times D_t \left(-\hat{s}_{t1}\right) = -D_t \left(\frac{m_w}{m} d_c \sin \alpha_t + \left(f + \frac{m_w}{m} d\right) \sin \left(\theta - \gamma_t\right)\right) \hat{b}_2 \qquad \text{(B.23)}$$

And, for the thrust of the vehicle,

$$\mathbf{r}_{t/CMs} \times T\hat{b}_1 = T\frac{m_w}{m} d_c \sin \phi \hat{b}_2 \qquad \text{(B.24)}$$

We must to include the aerodynamic moment at the aerodynamic center, M_{ac}, if it exists at zero lift.

Therefore, using Euler's Equation, see Eq. (B.9), the rotational equation of motion for the system about the CM is

$$\begin{aligned}
I_f\ddot{\theta} + I_w\left(\ddot{\theta} + \ddot{\phi}\right) &+ \frac{m_f m_w}{m}\left(d^2\ddot{\theta} + dd_c\left(2\ddot{\theta} + \ddot{\phi}\right)\cos\phi\right. \\
&\left. - dd_c\left(2\dot{\theta}\dot{\phi} + \dot{\phi}^2\right)\sin\phi + d_c^2\left(\ddot{\theta} + \ddot{\phi}\right)\right) = \\
&L_w\left(\left(e - \frac{m_w}{m}d_c\right)\cos\alpha_w + \frac{m_f}{m}d\cos\left(\theta - \gamma_w\right)\right) \\
&+ D_w\left(\left(e - \frac{m_w}{m}d_c\right)\sin\alpha_w + \frac{m_f}{m}d\sin\left(\theta - \gamma_w\right)\right) \\
&- L_t\left(\frac{m_w}{m}d_c\cos\alpha_t\left(f + \frac{m_w}{m}d\right)\cos\left(\theta - \gamma_t\right)\right) \\
&- D_t\left(\frac{m_w}{m}d_c\sin\alpha_t\left(f + \frac{m_w}{m}d\right)\sin\left(\theta - \gamma_t\right)\right) \\
&+ T\frac{m_w}{m}d_c\sin\phi + M_{ac}
\end{aligned} \qquad \text{(B.25)}$$

2. Wing Rotational Equation of Motion

This equation of motion may be determined in many ways. Note, that the equations of motion appear differently depending on the reference. For this equation, it is

possible to take the rotation of the fuselage about the CM, or the rotation of the wing about the CM, or the rotation of the wing about an arbitrary point, etc. It may be difficult to match equations of motion derived using different approaches because of the reference. This was an obstacle of this research. To match the fourth equation derived using the Gibbs-Appell Method, Eq. (A.71), it is necessary to derive the rotational equation of motion for the system about an arbitrary point, to be defined later, and then subtract the rotational equation of motion for the fuselage about the hinge. This is shown to be a valid method of deriving an equation of motion, and is necessary to develop and compare with the same equation of motion derived in the Gibbs-Appell derivation.

a.   Part 1: Rotational Equation of Motion of System about a Reference Point

To develop the rotational equations of motion, Euler's equation for a rotational body about an arbitrary reference point, '*', are used[21].

$$\mathbf{H}_* = \mathbf{H}_{CM} + m\left(\mathbf{r}_{CM/i} - \mathbf{r}_{*/i}\right) \times \left(\mathbf{v}_{CM/i} - \mathbf{v}_{*/i}\right) \tag{B.26}$$

$$\dot{\mathbf{H}}_* = \mathbf{l}_* + m\mathbf{a}_{*/i} \times \left(\mathbf{r}_{CM/i} - \mathbf{r}_{*/i}\right) \tag{B.27}$$

where, $\mathbf{H}$ is the angular momentum, CM is the center of mass for each individual body, $\mathbf{r}$ is the inertial position, $\mathbf{v}$ is the inertial velocity, and $\mathbf{l}$ is the sum of moments on the system. In this case, the '*' location is assumed to be the point defined as

$$\mathbf{r}_{*/h} = \frac{m_w}{m} d_c \hat{w}_1 \tag{B.28}$$

where h is the hinge attachment point for the wing, see Fig. 23 where the triangle represents the hinge. Recall, the free body diagram for the system is shown in Fig. 22.
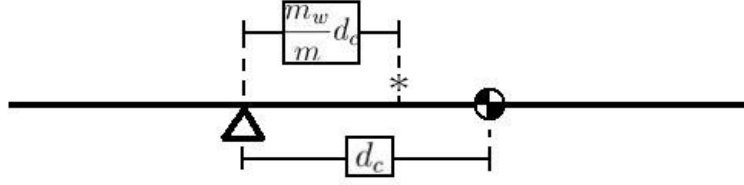
Fig. 23. Simplified Wing Model

The position of the wing CM with respect to the star location is defined as

$$\mathbf{r}_{CMw/*} = \frac{m_f}{m} d_c \hat{w}_1 \tag{B.29}$$

with the time derivative shown as

$$
\begin{aligned}
\mathbf{v}_{CMw/*} &= {}^w\frac{d}{dt}\left(\mathbf{r}_{CMw/*}\right) + \boldsymbol{\omega}_{w/i} \times \left(\mathbf{r}_{CMw/*}\right) \\
&= -\frac{m_f}{m} d_c \left(\dot{\theta} + \dot{\phi}\right) \hat{w}_3
\end{aligned}
\tag{B.30}
$$

$$
m_w \left(\mathbf{r}_{CMw/*} \times \mathbf{v}_{CMw/*}\right) = \left(m_w \left(\frac{m_f}{m}\right)^2 d_c^2 \left(\dot{\theta} + \dot{\phi}\right)\right)\hat{b}_2 \tag{B.31}
$$

The position of the fuselage CM with respect to the * reference is defined as

$$
\begin{aligned}
\mathbf{r}_{CMf/*} &= d\hat{b}_1 + \frac{m_w}{m} d_c \hat{w}_1 \\
&= \left(d + \frac{m_w}{m} d_c \cos\phi\right)\hat{b}_1 - \frac{m_w}{m} d_c \sin\phi\,\hat{b}_3
\end{aligned}
\tag{B.32}
$$

with the time derivative shown as

$$
\begin{aligned}
\mathbf{v}_{CMf/*} &= {}^b\frac{d}{dt}\left(\mathbf{r}_{CMf/*}\right) + \boldsymbol{\omega}_{b/i} \times \left(\mathbf{r}_{CMf/*}\right) \\
&= \left(-\frac{m_w}{m} d_c \left(\dot{\theta} + \dot{\phi}\right)\sin\phi\right)\hat{b}_1 - \left(d\dot{\theta} + \frac{m_w}{m} d_c \left(\dot{\theta} + \dot{\phi}\right)\cos\phi\right)\hat{b}_3
\end{aligned}
\tag{B.33}
$$

$$
\begin{aligned}
m_f \left(\mathbf{r}_{CMf/*} \times \mathbf{v}_{CMf/*}\right) = &\left(m_f d^2\dot{\theta} + \frac{m_w m_f}{m}\left(dd_c\left(\dot{\theta} + \dot{\phi}\right)\cos\phi + dd_c\dot{\theta}\cos\phi\right)\right. \\
&\left. + m_f\left(\frac{m_w}{m}\right)^2 d_c^2\left(\dot{\theta} + \dot{\phi}\right)\right)\hat{b}_2
\end{aligned}
\tag{B.34}
$$

With Eq. (B.26), one finds

$$\mathbf{H}_{CMw/*} = \mathbf{H}_{CMw} + m_w \left(\mathbf{r}_{CMw/*} \times \mathbf{v}_{CMw/*} \right) \hat{b}_2 \tag{B.35}$$

and

$$\mathbf{H}_{CMf/*} = \mathbf{H}_{CMw} + m_f \left(\mathbf{r}_{CMf/*} \times \mathbf{v}_{CMf/*} \right) \hat{b}_2 \tag{B.36}$$

where

$$\mathbf{H}_{CMw} = I_w \left(\dot{\theta} + \dot{\phi} \right) \hat{b}_2 \tag{B.37}$$

and

$$\mathbf{H}_{CMf} = I_f \dot{\theta} \hat{b}_2 \tag{B.38}$$

Therefore, the total angular momentum of the system about the * reference is

$$
\begin{aligned}
\mathbf{H}_{CMs/*} &= \mathbf{H}_{CMw/*} + \mathbf{H}_{CMf/*} \\
&= \left(I_f \dot{\theta} + I_w \left(\dot{\theta} + \dot{\phi} \right) + m_f d^2 \dot{\theta} \right. \\
&\quad \left. + \frac{m_w m_f}{m} \left(d d_c \left(2\dot{\theta} + \dot{\phi} \right) \cos\phi + d_c^2 \left(\dot{\theta} + \dot{\phi} \right) \right) \right) \hat{b}_2
\end{aligned}
\tag{B.39}
$$

with the time derivative shown as

$$
\begin{aligned}
\dot{\mathbf{H}}_{CMs/*} &= \left(I_f \ddot{\theta} + I_w \left(\ddot{\theta} + \ddot{\phi} \right) + m_f d^2 \ddot{\theta} \right. \\
&\quad \left. + \frac{m_w m_f}{m} \left(d d_c \left(2\ddot{\theta} + \ddot{\phi} \right) \cos\phi - d d_c \left(2\dot{\theta}\dot{\phi} + \dot{\phi}^2 \right) \sin\phi + d_c^2 \left(\ddot{\theta} + \ddot{\phi} \right) \right) \right) \hat{b}_2
\end{aligned}
\tag{B.40}
$$

Now, to determine the rotational equation of motion for the system about the star location,

$$\dot{\mathbf{H}}_{CMs/*} = \mathbf{l}_* + m\mathbf{a}_{*/i} \times \mathbf{r}_{CMs/*} \tag{B.41}$$

, the inertial acceleration of the star location is needed.

$$\mathbf{r}_{*/CMs} = \mathbf{r}_{*/CMw} + \mathbf{r}_{CMw/CMs}$$
$$= \frac{m_f}{m}\hat{b}_1$$
(B.42)

with the time derivative shown as

$$\mathbf{v}_{*/CMs} = {}^b\frac{d}{dt}\left(\mathbf{r}_{*/CMs}\right) + \boldsymbol{\omega}_{b/i} \times \left(\mathbf{r}_{*/CMs}\right)$$
$$= -\frac{m_f}{m}d\dot{\theta}\hat{b}_3$$
(B.43)

$$\mathbf{v}_{*/i} = \mathbf{v}_{*/CMs} + \mathbf{v}_{CMs/i}$$
$$= u\hat{b}_1 + \left(w - \frac{m_f}{m}d\dot{\theta}\right)\hat{b}_3$$
(B.44)

and, with the time derivative of Eq. (B.43) shown as

$$\mathbf{a}_{*/i} = {}^b\frac{d}{dt}\left(\mathbf{v}_{*/i}\right) + \boldsymbol{\omega}_{b/i} \times \left(\mathbf{v}_{*/i}\right)$$
$$= \left(\dot{u} + w\dot{\theta} - \frac{m_f}{d}\dot{\theta}^2\right)\hat{b}_1 + \left(\dot{w} - u\dot{\theta} - \frac{m_f}{m}d\ddot{\theta}\right)\hat{b}_3$$
(B.45)

Recall, Eq. (B.29) for the position of the wing CM with respect to the star location.

$$m\mathbf{a}_{*/i} \times \mathbf{r}_{CMs/*} = -m_f d\left(\dot{w} - u\dot{\theta} - \frac{m_f}{m}d\ddot{\theta}\right)\hat{b}_2$$
(B.46)

The moments about the * reference, $\mathbf{l}_*$, must be found. As stated previously, there are 6 moments acting on the system: moments due to the lift and drag of the wing, lift and drag of the tail, vehicle thrust, and the aerodynamic moment (due to an asymmetric airfoil).

Distances from the forces to the * reference are,

$$\mathbf{r}_{ACw/*} = \left(e - \frac{m_w}{m}d_c\right)\hat{w}_1$$
(B.47)

and

$$\mathbf{r}_{ACw/*} = \left(-(f+d) - \frac{m_w}{m}d_c\cos\phi\right)\hat{b}_1 + \frac{m_w}{m}d_c\sin\phi\hat{b}_3$$
(B.48)

Moments are as follows,

From the lift on the wing,

$$\mathbf{r}_{ACw/*} \times L_w\left(-\hat{s_{w3}}\right) = L_w\left(e - \frac{m_w}{m}d_c\right)\cos\alpha_w\hat{b_2} \tag{B.49}$$

From the drag on the wing,

$$\mathbf{r}_{ACw/*} \times D_w\left(-\hat{s_{w1}}\right) = D_w\left(e - \frac{m_w}{m}d_c\right)\sin\alpha_w\hat{b_2} \tag{B.50}$$

From the lift on the tail,

$$\mathbf{r}_{t/*} \times L_t\left(-\hat{s_{t3}}\right) = -L_t\left(\frac{m_w}{m}d_c\cos\alpha_t + (f+d)\cos\left(\theta-\gamma_t\right)\right)\hat{b_2} \tag{B.51}$$

From the drag on the tail,

$$\mathbf{r}_{t/*} \times D_t\left(-\hat{s_{t1}}\right) = -D_t\left(\frac{m_w}{m}d_c\sin\alpha_t + (f+d)\sin\left(\theta-\gamma_t\right)\right)\hat{b_2} \tag{B.52}$$

For the thrust of the vehicle,

$$\mathbf{r}_{t/*} \times T\hat{b_1} = T\frac{m_w}{m}d_c\sin\phi\hat{b_2} \tag{B.53}$$

And, the aerodynamic moment at zero lift, M_{ac}, must be included, if it exists.

Therefore, using Euler's Equation (see Eq. (B.9)) the rotational equation of

motion for the system about the $*$ reference, in the $\hat{b}_2$ direction, is

$$
\begin{aligned}
I_f\ddot{\theta} + I_w\left(\ddot{\theta} + \ddot{\phi}\right) &+ m_f d^2\ddot{\theta} + \frac{m_w m_f}{m}\left(d_c^2\left(\ddot{\theta} + \ddot{\phi}\right)\right. \\
&+ \left. dd_c\left(2\ddot{\theta} + \ddot{\phi}\right)\cos\phi - dd_c\left(2\dot{\theta}\dot{\phi} + \dot{\phi}^2\right)\sin\phi\right) = \\
&+ L_w\left(e - \frac{m_w}{m}d_c\right)\cos\alpha_w + D_w\left(e - \frac{m_w}{m}d_c\right)\sin\alpha_w \\
&- L_t\left(\frac{m_w}{m}d_c\cos\alpha_t + (f+d)\cos\left(\theta - \gamma_t\right)\right) \\
&- D_t\left(\frac{m_w}{m}d_c\sin\alpha_t + (f+d)\sin\left(\theta - \gamma_t\right)\right) \\
&+ T\frac{m_w}{m}d_c\sin\phi + M_{ac} - m_f d\left(\dot{w} - u\dot{\theta} - \frac{m_f}{m}d\ddot{\theta}\right)
\end{aligned}
\tag{B.54}
$$

b.   Part 2: Rotational Equation of Motion of Fuselage about Wing Hinge

Using Euler's Equation for rotation of a body about an arbitrary point, the angular momentum for the rotation of the CM of the fuselage about the wing hinge is

$$
\mathbf{H}_{CMf/h} = \mathbf{H}_{CMf} + m_f\left(\mathbf{r}_{CMf/i} - \mathbf{r}_{h/i}\right) \times \left(\mathbf{v}_{CMf/i} - \mathbf{v}_{h/i}\right)
\tag{B.55}
$$

and

$$
\dot{\mathbf{H}}_{CMf/h} = \mathbf{l}_h + m_f\mathbf{a}_{h/i} \times \mathbf{r}_{CMf/h}
\tag{B.56}
$$

where, $\mathbf{H}$ is the angular momentum, CM is the center of mass for each individual body, $\mathbf{r}$ is the inertial position, $\mathbf{v}$ is the inertial velocity, h is the wing hinge location, and $\mathbf{l}$ is the sum of moments on the system. The free body diagram for the fuselage is shown in Fig. 24. The corresponding free body diagram of the wing is presented, see Fig. 25, to highlight the equal and opposite relationship of the reaction forces, R_x and R_z, and moment, $k(\phi - \alpha_r)$, between the fuselage and the wing. Note, that the reaction forces, R_x and R_z, will not contribute to the sum of moments for the fuselage about the wing attachment point because they act through the point about which the moments are being taken.
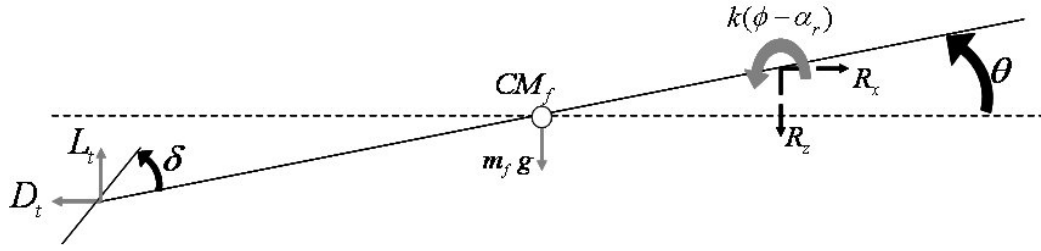
Fig. 24. Fuselage Free Body Diagram
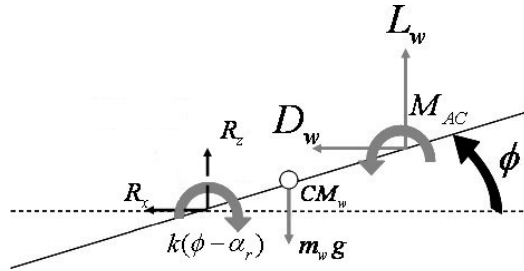
Fig. 25. Wing Free Body Diagram

The position of the fuselage CM with respect to the wing hinge is

$$\mathbf{r}_{CMf/h} = -d\hat{b_1} \tag{B.57}$$

with the time derivative shown as

$$\mathbf{v}_{CMf/h} = {}^{b}\frac{d}{dt}\left(\mathbf{r}_{CMf/h}\right) + \boldsymbol{\omega}_{b/i} \times \left(\mathbf{r}_{CMf/h}\right)$$
$$= d\dot{\theta}\hat{b_3} \tag{B.58}$$

$$m_f\left(\mathbf{r}_{CMf/h} \times \mathbf{v}_{CMf/h}\right) = m_f d^2 \ddot{\theta}\hat{b_2} \tag{B.59}$$

The angular momentum for the fuselage CM about the wing hinge is,

$$\begin{aligned}
\mathbf{H}_{CMf/h} &= \mathbf{H}_{CMf} + m_f \left(\mathbf{r}_{CMf/i} - \mathbf{r}_{h/i} \right) \times \left(\mathbf{v}_{CMf/i} - \mathbf{v}_{h/i} \right) \\
&= \left(I_f \dot{\theta} + m_f d^2 \ddot{\theta} \right) \hat{b}_2
\end{aligned} \tag{B.60}$$

The rotational equation of motion for the fuselage CM about the wing hinge is,

$$\dot{\mathbf{H}}_{CMf/h} = \mathbf{l}_h + m\mathbf{a}_{h/i} \times \mathbf{r}_{CMf/h} \tag{B.61}$$

The inertial acceleration of the wing hinge is needed. Starting with the velocities,

$$\begin{aligned}
\mathbf{v}_{h/i} &= \mathbf{v}_{h/CMs} + \mathbf{v}_{CMs/i} \\
&= \left(u + \frac{m_w}{m} d_c \left(\dot{\theta} + \dot{\phi} \right) \sin \phi \right) \hat{b}_1 \\
&+ \left(w + \frac{m_w}{m} d_c \left(\dot{\theta} + \dot{\phi} \right) \cos \phi - \frac{m_f}{m} d\dot{\theta} \right) \hat{b}_3
\end{aligned} \tag{B.62}$$

the time derivative of Eq. (B.62) is

$$\begin{aligned}
\mathbf{a}_{h/i} &= {}^b\frac{d}{dt} \left(\mathbf{v}_{h/i} \right) + \boldsymbol{\omega}_{b/i} \times \left(\mathbf{v}_{h/i} \right) \\
&= \left(\dot{u} + w\dot{\theta} + \frac{m_w}{m} \left(\ddot{\theta} + \ddot{\phi} \right) \sin \phi + \frac{m_w}{m} d_c \left(\dot{\theta} + \dot{\phi} \right)^2 \cos \phi - \frac{m_f}{m} d\dot{\theta}^2 \right) \hat{b}_1 \\
&+ \left(\dot{w} - u\dot{\theta} + \frac{m_w}{m} d_c \left(\ddot{\theta} + \ddot{\phi} \right) \cos \phi - \frac{m_w}{m} d_c \left(\dot{\theta} + \dot{\phi} \right)^2 \sin \phi - \frac{m_f}{d} \ddot{\theta} \right) \hat{b}_3
\end{aligned} \tag{B.63}$$

Recall, Eq. (B.29) for the position of the wing CM with respect to the * reference is

$$\begin{aligned}
m\mathbf{a}_{h/i} \times \mathbf{r}_{CMf/h} &= -m_f d \Big(\dot{w} - u\dot{\theta} - \frac{m_f}{m} d\ddot{\theta} + \frac{m_w}{m} \left(\ddot{\theta} + \ddot{\phi} \right) \cos \phi \\
&- \frac{m_w}{m} d_c \left(\dot{\theta} + \dot{\phi} \right)^2 \sin \phi \Big) \hat{b}_2
\end{aligned} \tag{B.64}$$

The moments about the wing hinge, $\mathbf{l}_h$, are obtained. There are 5 moments acting on the fuselage: moments due to the lift and drag of the tail, vehicle thrust, torsional spring, and the aerodynamic center (due to an unsymmetric airfoil). Note, in this instance the spring torsional moment is included instead of the individual moments due to the lift and drag of the wing.

The distance from the tail forces to the wing hinge is

$$\mathbf{r}_{t/h} = -d\hat{b}_1 \tag{B.65}$$

All moments are calculated,

From the lift on the tail,

$$\mathbf{r}_{t/h} \times L_t \left(-\hat{s}_{t3}\right) = -L_t \left(f + d\right) \cos\left(\theta - \gamma_t\right) \hat{b}_2 \tag{B.66}$$

From the drag on the tail,

$$\mathbf{r}_{t/h} \times D_t \left(-\hat{s}_{t1}\right) = -D_t \left(f + d\right) \sin\left(\theta - \gamma_t\right) \hat{b}_2 \tag{B.67}$$

From the thrust of the vehicle,

$$\mathbf{r}_{t/h} \times T\hat{b}_1 = 0 \tag{B.68}$$

The moments due to the aerodynamic moment at zero lift, M_{ac}, and to the torsional spring, $k\left(\phi - \alpha_r\right)$ must be included.

Therefore, using Euler's Equation (see Eq. (B.9)) the rotational equation of motion, in the $\hat{b}_2$ direction, for the fuselage rotating about the wing hinge is

$$\begin{aligned}
I_f\ddot{\theta} + m_f d^2\ddot{\theta} + \frac{m_w m_f}{m}&\left(\left(\ddot{\theta} + \ddot{\phi}\right)\cos\phi\right. \\
&\left. - \frac{m_w}{m}d_c\left(\dot{\theta} + \dot{\phi}\right)^2\sin\phi\right) = k\left(\phi - \alpha_r\right) \\
&- L_t\left(f + d\right)\cos\left(\theta - \gamma_t\right) \\
&- D_t\left(f + d\right)\sin\left(\theta - \gamma_t\right) + M_{ac} \\
&- m_f d\left(\dot{w} - u\dot{\theta} - \frac{m_f}{m}d\ddot{\theta}\right)
\end{aligned} \tag{B.69}$$

The rotational equation of motion for the wing, that matches with the previous Gibbs-Appell formulation, is found by subtracting Eq. (B.54) from Eq. (B.69).

c. Wing Rotational Equation of Motion

$$I_w \left(\ddot{\theta} + \ddot{\phi} \right) + \frac{m_w m_f}{m} \left(d_c^2 \left(\ddot{\theta} + \ddot{\phi} \right) + d d_c \ddot{\theta} \cos \phi + d d_c \dot{\theta}^2 \sin \phi \right) =$$
$$- k \left(\phi - \alpha_r \right) + L_w \left(e - \frac{m_w}{m} d_c \right) \cos \alpha_w + D_w \left(e - \frac{m_w}{m} d_c \right) \sin \alpha_w$$
$$- L_t \frac{m_w}{m} d_c \cos \alpha_t - D_t \frac{m_w}{m} d_c \sin \alpha_t$$
$$+ T \frac{m_w}{m} d_c \sin \phi + M_{ac} \tag{B.70}$$

3. Newton-Euler Equations of Motion

Translational Equation of Motion (in the $\hat{b}_1$ frame):

$$m \left(\dot{u} + w \dot{\theta} \right) = T - mg \sin \theta + L_w \sin \left(\theta - \gamma_w \right) + L_t \sin \left(\theta - \gamma_t \right)$$
$$- D_w \cos \left(\theta - \gamma_w \right) - D_t \cos \left(\theta - \gamma_t \right) \tag{B.71}$$

Translational Equation of Motion (in the $\hat{b}_3$ frame):

$$m \left(\dot{w} - u \dot{\theta} \right) = mg \cos \theta - L_w \cos \left(\theta - \gamma_w \right) - L_t \cos \left(\theta - \gamma_t \right)$$
$$- D_w \sin \left(\theta - \gamma_w \right) - D_t \sin \left(\theta - \gamma_t \right) \tag{B.72}$$

System Rotational Equation of Motion:

$$I_f \ddot{\theta} + I_w \left(\ddot{\theta} + \ddot{\phi} \right) + \frac{m_f m_w}{m} \left(d^2 \ddot{\theta} + d d_c \left(2 \ddot{\theta} + \ddot{\phi} \right) \cos \phi \right.$$
$$\left. - d d_c \left(2 \dot{\theta} \dot{\phi} + \dot{\phi}^2 \right) \sin \phi + d_c^2 \left(\ddot{\theta} + \ddot{\phi} \right) \right) =$$
$$L_w \left(\left(e - \frac{m_w}{m} d_c \right) \cos \alpha_w + \frac{m_f}{m} d \cos \left(\theta - \gamma_w \right) \right)$$
$$+ D_w \left(\left(e - \frac{m_w}{m} d_c \right) \sin \alpha_w + \frac{m_f}{m} d \sin \left(\theta - \gamma_w \right) \right) \tag{B.73}$$
$$- L_t \left(\frac{m_w}{m} d_c \cos \alpha_t \left(f + \frac{m_w}{m} d \right) \cos \left(\theta - \gamma_t \right) \right)$$
$$- D_t \left(\frac{m_w}{m} d_c \sin \alpha_t \left(f + \frac{m_w}{m} d \right) \sin \left(\theta - \gamma_t \right) \right)$$
$$+ T \frac{m_w}{m} d_c \sin \phi + M_{ac}$$

Wing Rotational Equation of Motion:

$$
\begin{aligned}
I_w \left(\ddot{\theta} + \ddot{\phi} \right) + \frac{m_w m_f}{m} \left(d_c^2 \left(\ddot{\theta} + \ddot{\phi} \right) + dd_c \ddot{\theta} \cos \phi + dd_c \dot{\theta}^2 \sin \phi \right) = \\
- k \left(\phi - \alpha_r \right) + L_w \left(e - \frac{m_w}{m} d_c \right) \cos \alpha_w + D_w \left(e - \frac{m_w}{m} d_c \right) \sin \alpha_w \\
- L_t \frac{m_w}{m} d_c \cos \alpha_t - D_t \frac{m_w}{m} d_c \sin \alpha_t \\
+ T \frac{m_w}{m} d_c \sin \phi + M_{ac}
\end{aligned}
\tag{B.74}
$$

APPENDIX C

TEST CASE AND MATLAB CODE

The following sections include the Matlab files used to run the simulations for the 4 DOF model. Recall, the basic algorithm for the simulations is written in section C chapter II. For a test case, the main inputs (shown in Table VI) were used. Using the main inputs from Table VI and the inputs from Table VII, Figs. 5-6 were obtained. Similarly, using the main inputs from Table VI and the inputs from Table VIII, Figs. 7-12 were obtained. And, using the main inputs from Table VI and the inputs from Table IX, Figs. 14-15 were obtained.

Table VI. System Parameters

$Variable$	$Value$	$Units$	$Description$
W	2650	lb	Weight
dm	0.5	s	Half-period for Gust
ts	10	s	Time to Start Gust
$t0$	0	s	Initial Time
tf	300	s	End Time
I_f	1143.8	$lb - ft^2$	Fuselage Moment of Inertia
I_w	201.9	$lb - ft^2$	Wing Moment of Inertia
ρ	2.0495×10^{-3}	$slug/ft^3$	Air Density
g	32.2	ft/s^2	Gravity
m_f	69.95	$slug$	Fuselage Mass
m_w	12.34	$slug$	Wing Mass
$span$	36	ft	Wing Span
$span_t$	10.49	ft	Tail Span
S_w	174	ft^2	Wing Area
$S_{\delta f}$	15	ft^2	Flaperon Area
$S_{\delta e}$	16.61	ft^2	Elevator Area
c	4.9	ft	Wing Chord
c_t	2.19	ft	Tail Chord
e	0.075c	ft	Distance from AC to Hinge
d_c	0.0078c	ft	Distance from Wing CM to Hinge
d	-0.042c	ft	Distance from Fuselage CM to Hinge
f	23.6	ft	Distance from Tail to Fuselage CM
x_{CM}	26.3	$\%$	System CM location (% chord)
$C_L\alpha w$	4.3		Wing Lift Coefficient with respect to (wrt) Angle of Attack (AOA)
$C_L\alpha t$	1.87		Tail Lift Coefficient wrt AOA
$C_D\alpha w$	0.121		Wing Drag Coefficient wrt AOA
$C_D\alpha t$	0		Tail Drag Coefficient wrt AOA
C_L0w	0.307		Wing Lift Coefficient with Zero AOA
C_L0t	0		Tail Lift Coefficient with Zero AOA
C_D0w	0.027		Wing Drag Coefficient with Zero AOA
C_D0t	0.0355		Tail Drag Coefficient with Zero AOA
$C_L\delta$	0.43		Lift Coefficient wrt Elevator Deflection
$C_M\delta$	-1.122		Moment Coefficient wrt Elevator Deflection
C_Lq	-12.4		Lift Coefficient due to Pitching Moment
τ_f	0.25		Flaperon Effectiveness Parameter

Table VII. Open-loop Parameters for Low Speed, High Stiffness Case

$Variable$	$Value$	$Units$	$Description$
$case$	1		Case to Evaluate
k	1×10^7	$\frac{lb-ft}{rad}$	Spring Stiffness
u_d	228	$\frac{ft}{s}$	Desired Velocity

Table VIII. Linear Closed-loop Parameters for Low Speed, High Stiffness Case

$Variable$	$Value$	$Units$	$Description$
$case$	2		Case to Evaluate
k	1×10^5	$\frac{lb-ft}{rad}$	Spring Stiffness
u_d	244.5	$\frac{ft}{s}$	Desired Velocity

A.   Main Matlab File: *main_total.m*

```
%% 4DOF Longitudinal Model
%% use for open loop responses for linear and nonlinear
close all
clear all
clc
format short g

global xeq rootsA I o E K phi_big gamma_big index od

%% Load Inputs
tic
input4dof_cessna182
load fdinputs

index = 0;
%% Trim Calculation

% Set initial guesses
x0 = initguess(I.u0, I.w0, I)
disp('Initial Guesses:')
disp(x0')

for i = 1:length(x0)
    if abs(x0(i)) <= 1e-7
        x0(i) = 0;
    else
        x0(i) = x0(i);
    end
end

disp('Initial Guesses:')
disp(x0')
```

Table IX. Linear Closed-loop Parameters for High Speed, Low Stiffness Case

$Variable$	$Value$	$Units$	$Description$
$case$	3		Case to Evaluate
k	1×10^5	$\frac{lb-ft}{rad}$	Spring Stiffness
u_d	244.5	$\frac{ft}{s}$	Desired Velocity

```
o = 0;           % value that triggers the gust (see liftdrag and dof4)

% define options for Fmincon

options = optimset('MaxFunEvals',1000000,'Display','iter','TolFun',1e-8,...
    'TolCon',1e-8,'MaxIter',600000);

[xeq, fval, exitflag] = fmincon(@trimfourdof, x0, [], [], [], [], [], [],...
    'nonlincon', options);

%% Manually set equilibrium values to zero if smaller than 1e-4
% avoids machine error

for i = 1:length(xeq)
    if abs(xeq(i)) <= 1e-7
        xeq(i) = 0;
    else
        xeq(i) = xeq(i);
    end
end

%% Linearization

[A,B] = linize(xeq,I,o);

[eigVc_A,eigVal_A] = eig(A);     %find eigenvectors and eigenvalues

rootsA = eig(A);         %find eigenvalues only, save to different variable
[wn,damps] = damp(A);    %find freqs and damping ratios from eigenvalues
damp(A)

disp('System CG Location (percent of chord):')
disp(100*(0.3 - (I.mw/I.m*(I.d + I.dc*cos(I.alfr)) - I.d)/I.c))

disp('Aerodynamic Center Location (percent of chord):')
disp(100*(0.3*I.c - I.e)/I.c)

%% Model

if I.case == 1
    openloop
elseif I.case == 2
    linearcontrol
else
    nonlinearcontrol
end

% Get Frequency Values, could be used for fft analysis
[wn,zeta] = damp(A);
```

```
I.one = wn(1);
I.two = wn(3);
I.three = wn(5);
```

B. Input File: *input_4dof_cessna182.m*

```
%subroutine to define input values for 4DOF model
function input4dof_gen_aviation

%% Input Variables to specify what to run and how long
%kiss = 0, use full model, kiss = 1, use simple model

I.kiss = 1;
I.fftstate = 5; % state to do fft on
I.sim_run = 1; % to run, 1, or not to run, 0
I.tf = 300;         %end time
I.t0 = 0;           %start time
I.wing = 'b';        % 'a' will map wing root behavior
I.timesteps = 500*I.tf; %number of time steps (when using 'runge' for ode)
I.case = 1;
%which model to run, 1-OL (both), 2-Linear (OL/CL), 3-Nolinear (OL/CL)
I.ts = 10;       % Time to start disturbance

%% Input Trim Velocity and Instantaneous Disturbance
I.u0 = 244.5;     %ft/s, velocity to trim at %% DNE speed is 300
% I.u0 = 228;      % use I.u0 = 228 to get LCO (with k = 1e5 or less)
I.w0 = 0;

%Velocities used to verify Linear & Nonlinear are the same for small
%disturbances:
% I.ui = .0001*I.u0;
% I.wi = -.00002*I.u0;

%Regular Velocities:
% I.ui = .01*I.u0;
% I.wi = -.02*I.u0;

I.ui = 0;
I.wi = 0;

%% Gust Velocity

I.dm = 0.5;              % Time length for gust to reach maximum velocity

%Velocities used to verify Linear & Nonlinear are the same for small
%disturbances:
% I.Uhg = .0001*I.u0;
% I.Uvg = -.00008*I.u0;

%Regular Velocities:
I.Uhg = 0%0.15*I.u0;
I.Uvg = -.03*I.u0;
% I.Uhg = 0;
% I.Uvg = 0;

%input physical constants
I.rho = 2.0495e-3;                  % slug/ft^3 air
I.g = 32.2;                         % gravity ft/s^2


%% Input System Parameters
I.W = 2650;                         %(lb) (from Roskam pg. 480)
I.m = I.W/I.g;                      %(slug) total mass of system
```

```
I.mf = 0.85*I.m;                        %(slug) total mass of fuselage
I.mw = I.m - I.mf;                      %(slug) total mass of wing
I.lf = 28;                              %(ft) length of fuselage


%% Wing
%wing is attached at elastic axis
I.k0 = 1e5;            %lbf_ft/rad  wing root stiffness


I.alfr = 0;                        %root angle of attack
I.span = 36;                       %(ft) wingspan (from JAWA)
I.c = 4.9;                         %(ft) chord (from Roskam pg. 480)
I.b = I.c/2;                       %(ft) halfchord
I.S = 174;                         %wing area, (ft^2) (from Roskam pg. 480)
I.AR = (I.span^2)/I.S;             %aspect ratio
I.e = 0.075*I.c;                   %distance btw aerodynamic center and hinge
I.d = -0.042002*I.c*cos(I.alfr);
I.dc = 0.0078*I.c;                 %distance btw cgw and hinge
I.k = I.k0;
I.k_1 = I.k;
I.oe = 0.75;                       %Oswald efficiency
I.R = 1/(pi*I.oe*I.AR);


%% Tail
I.spant = 10.49;                   %(ft) tail span 14.4
I.St = 22.96;                      %(ft^2) tail area (from JAWA)
I.Sf = 15;                         %(ft^2)  flaperon area
I.Se = 16.61;                      %(ft^2)  elevator area (from JAWA)
I.ct = I.St/I.spant;               %(ft) tail chord
I.bt = I.ct/2;                     %(ft) tail half chord
I.ARt = (I.spant^2)/I.St;          %tail aspect ratio
I.Rt = 1/(pi*I.oe*I.ARt);
I.f = 23.6;                        %distance btw cg of fuselage and tail pivot
I.lt = I.f + I.d;                  %(ft) length from tail attachment to cg_sys


%% Moment of Inertia
I.Iy = 1346;             % aircraft mass MOI (measured about CG), slug-ft^2
I.Iw = I.Iy*0.15;               %wing MOI wrt its cg
I.If = I.Iy - I.Iw - (I.mf*I.mw/I.m)*(I.dc^2 + I.d^2 ...
    + 2*I.d*I.dc*cos(I.alfr));        %MOI of fuselage about its cg


%% Wing Frequency
I.om_wing =  sqrt(I.k/(I.Iw + (I.mf*I.mw/I.m)*(I.dc^2))); %natural frequency of wing about support


%% Lift/Drag Coefficients
I.Clalfw = 4.3;
I.Cl0 = 0.307;          %(from Roskam pg. 483)
I.Cl0w = I.Cl0;
I.Cd0 = 0.027;          %(from Roskam pg. 483)
I.Cd0w = 0.027;         %(from Roskam pg. 483)
I.Cdalf = 0.121;        %(from Roskam pg. 483)
I.Cdalfw = I.Cdalf;
I.Cdalft = 0;

I.Clalft = 1.87;        % Assumption (to better match Roskam values)
I.Cldelta = 0.43;       %(from Roskam pg. 483)
I.Cddelta = 0;
I.Clt0 = 0;             % Assumption
I.Cdt0 = 0.0355;        % Assumption (to better match Roskam values)
I.Cdu = 0;
I.Clu = 0;


I.Clq = 3.9;            %(from Roskam pg. 483)
I.Cmq = -12.4;          %(from Roskam pg. 483)
I.Cm_de = -1.122;       %(from Roskam pg. 483)
```

```
I.deda = (2*I.Clalfw)/(pi*I.AR);
I.eps0 = (2*I.Cl0)/(pi*I.AR);

%% Control Surface Effectiveness Parameters
I.vt = (I.lt*(I.St))/(I.c*I.S);
I.tau = -I.Cm_de/(I.vt*1*I.Clalft);          % approx from Nelson pg 66

I.tau_f = 0.25; % approximation (see Nelson pg. 64)

save fdinputs
```

C.   Initial Guess Function for Trim Solver: *initguess.m*

```
function x0 = initguess(u, w, I)

K0 = 6.875678389548968e-006;        % /ft
h0 = 5000;           % ft
rho0 = 0.00238;      % lb-s^2/ft^4
rho = rho0*(1-K0*h0)^(4.25);         % slug/ft^3 air

Vh = sqrt((u^2) + (w^2));
% Calculates Velocity Scalar from x and y components

Q = 0.5*rho*(Vh^2);
Qt = Q;

eta = ((Qt*I.St)/(Q*I.S));
CLde = I.Cldelta;

CLdf = I.Clalfw*I.tau_f*(I.Sf/I.S);

Cmac = -0.045;
Mac = Q*I.S*I.c*Cmac;
Mact = 0;

%Building Matrix for Linearized Equations of Motion at TRIM
%(of the format H*x = P)
H(1,:) = [0
    -Q*I.S*(I.Cdalfw + eta*I.Cdalft*(1 - I.deda))
    0
    1];          % From translation equation #1

H(2,:) = [-Q*I.S*CLdf
    -Q*I.S*(I.Clalfw + eta*I.Clalft*(1 - I.deda))
    -Q*I.S*CLde
    0];          % From translation equation #2

H(3,:) = [(I.e - (I.mw/I.m)*I.dc + (I.mf/I.m)*I.d)*CLdf*Q*I.S
    (Q*I.S*(I.Clalfw*(I.e - (I.mw/I.m)*I.dc + (I.mf/I.m)*I.d) +...
    I.Cd0*(I.e - (I.mw/I.m)*I.dc) - eta*I.Clalft*(1 - I.deda)*...
    ((I.mw/I.m)*(I.dc + I.d) + I.f) - eta*I.Cdt0*(I.mw/I.m)*...
    I.dc*(1 - I.deda)))
    -Q*I.S*CLde*((I.mw/I.m)*(I.dc + I.d) + I.f)
    (I.mw/I.m)*I.dc*I.alfr];           % From System Moment equation

H(4,:) = [(I.e - (I.mw/I.m)*I.dc)*CLdf*Q*I.S
    (Q*I.S*((I.Clalfw + I.Cd0)*(I.e - (I.mw/I.m)*I.dc) - ...
    eta*I.Clalft*(1 - I.deda)*(I.mw/I.m)*I.dc - eta*(I.mw/I.m)...
    *I.dc*I.Cdt0*(1 - I.deda)) - I.k)
    -Q*I.S*CLde*(I.mw/I.m)*I.dc
    (I.mw/I.m)*I.dc*I.alfr];             % From Wing Moment equation
```

```
P = -[-Q*I.S*(I.Cd0 + eta*I.Cdalft*(I.alfr + I.eps0) + eta*I.Cdt0)
    I.m*I.g - Q*I.S*(I.Cl0 - eta*I.Clalft*(I.alfr + I.eps0))
    (Q*I.S*(I.Cl0*(I.e - (I.mw/I.m)*I.dc + (I.mf/I.m)*I.d) + ...
    eta*I.Clalft*(I.alfr + I.eps0)*((I.mw/I.m)*(I.dc + I.d) + I.f) ...
    + eta*I.Cdt0*(I.alfr + I.eps0)) + Mac + Mact)
    (Q*I.S*(I.Cl0*(I.e - (I.mw/I.m)*I.dc) + eta*I.Clalft*(I.mw/I.m)...
    *I.dc*(I.alfr + I.eps0) + eta*(I.mw/I.m)*I.dc*I.Cdt0*(I.alfr +...
    I.eps0)) + I.k*I.alfr) + Mac];

initial = inv(H)*P;
df0 = initial(1);
phi0 = initial(2);
del0 = initial(3);
T0 = initial(4);

% Set Initial guess for controls and states
x0 = [u
    w
    w/u
    0
    phi0
    0
    del0
    df0
    T0]';
```

D. Trim Solver Function: *trim4dof.m*

```
function J = trimfourdof(x)

global I o

%% Bring in initial guess states
u = x(1);
w = x(2);
theta = x(3);
theta_d = x(4);
phi = x(5);
if I.k > 1e8
    phi = I.alfr;        % to avoid machine zero error
end
phi_d = x(6);
h = x(3) - atan2(x(2),x(1));
del = x(7);
df = x(8);
T = x(9);

%% Solve for Lift and Drag from initial guess states

liftdrag

%% Expression for x_dot

EOM

%% Output
Weight = diag([1 1 1 1 1 1]); % state weighting matrix
J = x_dot'*Weight*x_dot;
```

E. Nonlinear Constraints for Trim Solver: *nonlincon.m*

```
function [c,ceq] = nonlincon(x)

global I o

u = x(1);
w = x(2);
theta = x(3);
theta_d = x(4);
phi = x(5);
phi_d = x(6);
h = x(3) - atan2(x(2),x(1));
del = x(7);
df = x(8);
T = x(9);

Is1 = I.Iw + I.If + (I.mf*I.mw/I.m)*(I.dc^2 + I.d^2 + 2*I.d*I.dc*cos(phi));
Is2 = I.Iw + (I.mf*I.mw/I.m)*(I.dc^2 + I.d*I.dc*cos(phi));

Iw1 = I.Iw + ((I.mf/I.m)^2)*I.mw*(I.dc^2 + I.d^2 + 2*I.d*I.dc*cos(phi));
Iw2 = I.Iw + ((I.mf/I.m)^2)*I.mw*(I.dc^2 + I.d*I.dc*cos(phi));

liftdrag

Cmac = -0.045;
Mac = Q*I.S*I.c*Cmac;
Mact = 0;

c = [x(2) - 15
     -x(2) - 15
     x(7) - (28*pi/180)
     -x(7) - (21*pi/180)
     x(8) - (22*pi/180)
     -x(8) - (15*pi/180)];

ceq = [x(1) - I.u0
    (L*sin(theta - gam_w) + Lt*sin(theta - gam_t) - D*cos(theta - gam_w)...
    - Dt*cos(theta - gam_t) + T - I.m*I.g*sin(theta))
    (-L*cos(theta - gam_w) - Lt*cos(theta - gam_t) - Dt*sin(theta - gam_t)...
    - D*sin(theta - gam_w) + I.m*I.g*cos(theta))
    ((I.mf*I.mw/I.m)*I.d*I.dc*(2*theta_d*phi_d + phi_d^2)*sin(phi) + ...
    L*((I.e - (I.mw/I.m)*I.dc)*cos(alpha_w) + (I.mf/I.m)*I.d*cos(theta - gam_w))...
    + D*((I.e - (I.mw/I.m)*I.dc)*sin(alpha_w) + (I.mf/I.m)*I.d*sin(theta - gam_w))...
    - Lt*(((I.mw/I.m)*I.d + I.f)*cos(theta - gam_t) + (I.mw/I.m)*I.dc*cos(alpha_t))...
    - Dt*(((I.mw/I.m)*I.d + I.f)*sin(theta - gam_t) + (I.mw/I.m)*I.dc*sin(alpha_t))...
    + T*(I.mw/I.m)*I.dc*sin(phi) + Mac)
    (-I.k*(phi - I.alfr) - (I.mw*I.mf/I.m)*I.d*I.dc*(theta_d^2)*sin(phi) + ...
    L*(I.e - (I.mw/I.m)*I.dc)*cos(alpha_w)...
    + D*(I.e - (I.mw/I.m)*I.dc)*sin(alpha_w) - Lt*(I.mw/I.m)*I.dc*cos(alpha_t)...
    - Dt*(I.mw/I.m)*I.dc*sin(alpha_t) + T*(I.mw/I.m)*I.dc*sin(phi) + Mac)
    x(3) - atan2(x(2),x(1))
    x(4)
    x(6)];
```

F. Aerodynamics File: *liftdrag.m*

```
%% Angles

%update physical constants
K0 = 6.8757e-006;        % /ft
h0 = 5000;          % ft
% h
```

```
h_t = h + h0;
rho0 = 0.00238;      % lb-s^2/ft^4
rho = rho0*(1-K0*h_t)^(4.25);     %slug/ft^3 air

% Wing velocity, determined using velocity at system CM and included
% induced velocity due to rotation of Wing CM about system CM
xw_d = u - (I.mf/I.m)*I.dc*(theta_d + phi_d)*sin(phi);
yw_d = w - (I.mf/I.m)*(I.d*theta_d + I.dc*(theta_d + phi_d)*cos(phi));

gam_w = theta - atan2(yw_d,xw_d);

% Tail velocity, determined using velocity at system CM and included
% induced velocity due to rotation of Tail CM about system CM
xt_d = u + (I.mw/I.m)*I.dc*(theta_d + phi_d)*sin(phi);
yt_d = w + (I.d + I.f)*theta_d + (I.mw/I.m)*(I.dc*(theta_d + ...
    phi_d)*cos(phi));

gam_t = theta - atan2(yt_d,xt_d);

% Angle of Attack for Wing and Tail
alpha_w = theta + phi - gam_w;

alpha_t = alpha_w*(1 - I.deda) - I.alfr - I.eps0;

% for the system
gam = theta - atan2(w,u);
V = sqrt(u^2 + w^2);

gam_1 = [gam_w
    gam_t
    gam];

for n = 1:length(gam_1)
    if abs(gam_1(n)) <= 1e-7          % avoids machine zero error
        gam_1(n) = 0;
    else
        gam_1(n) = gam_1(n);
    end
end

gam_w = gam_1(1);
gam_t = gam_1(2);
gam = gam_1(3);
%% Lift/Drag

% for the wing
Vw = sqrt(xw_d^2 + yw_d^2) ;  %Calculates Velocity Scalar from x and y
Q = 0.5*rho*(Vw^2);           %Calculates Dynamic Pressure

% for the tail
Vt = sqrt(xt_d^2 + yt_d^2);
Qt = 0.5*rho*(Vt^2);

eta = ((Qt*I.St)/(Q*I.S));
CLde = I.Cldelta;
CLdf = I.Clalfw*I.tau_f*(I.Sf/I.S);

CLw = I.Cl0 + I.Clalfw*alpha_w + CLdf*df;
CLt = I.Clt0 + I.Clalft*(alpha_t) + CLde*del + I.Clq*theta_d;

CDw = I.Cd0 + I.Cdalfw*alpha_w;
CDt = I.Cdt0 + I.Cdalft*alpha_t;

L = CLw*Q*I.S;
```

```
D = Q*I.S*CDw;

Lt = CLt*Qt*I.St;
Dt = Qt*I.St*CDt;

if o == 1
    Lvg = Q*I.S*I.Clalfw*(V_vg/Vw);
    Lhg = (rho/2)*(2*V_hg*Vw + V_hg^2)*I.S*CLw;
else
    Lvg = 0;
    Lhg = 0;
end
```

G.   Equations of Motion File: *EOM.m*

```
%% M*x_dot = F_nl

% Moment of inertia quantities:
Is1 = I.Iw + I.If + (I.mf*I.mw/I.m)*(I.dc^2 + I.d^2 + 2*I.d*I.dc*cos(phi));
Is2 = I.Iw + (I.mf*I.mw/I.m)*(I.dc^2 + I.d*I.dc*cos(phi));

Iw1 = I.Iw + (I.mf*I.mw/I.m)*(I.dc^2 + I.d*I.dc*cos(phi));
Iw2 = I.Iw + (I.mf*I.mw/I.m)*(I.dc^2);

% Mass Matrix
M = [I.m 0 I.m*w 0 0 0
    0 I.m (-I.m*u) 0 0 0
    0 0 1 0 0 0
    0 0 0 Is1 0 Is2
    0 0 0 0 1 0
    0 0 0 Iw1 0 Iw2];

Cmac = -0.045;
Mac = Q*I.S*I.c*Cmac;
Mact = 0;

% Nonlinear Forces

F_nl(1) = (L + Lhg + Lvg)*sin(theta - gam_w) + Lt*sin(theta - gam_t) ...
    - D*cos(theta - gam_w)- Dt*cos(theta - gam_t) + T - I.m*I.g*sin(theta);
F_nl(2) = -(L + Lhg + Lvg)*cos(theta - gam_w) - Lt*cos(theta - gam_t) ...
    - Dt*sin(theta - gam_t) - D*sin(theta - gam_w) + I.m*I.g*cos(theta);
F_nl(3) = theta_d;
F_nl(4) = (I.mf*I.mw/I.m)*I.d*I.dc*(2*theta_d*phi_d + phi_d^2)*sin(phi) ...
    + (L + Lvg)*((I.e - (I.mw/I.m)*I.dc)*cos(alpha_w) ...
    + (I.mf/I.m)*I.d*cos(theta - gam_w)) + D*((I.e - ...
    (I.mw/I.m)*I.dc)*sin(alpha_w) + (I.mf/I.m)*I.d*sin(theta - gam_w))...
    - Lt*(((I.mw/I.m)*I.d + I.f)*cos(theta - gam_t) + ...
    (I.mw/I.m)*I.dc*cos(alpha_t)) - Dt*(((I.mw/I.m)*I.d + ...
    I.f)*sin(theta - gam_t) + (I.mw/I.m)*I.dc*sin(alpha_t))...
    + T*(I.mw/I.m)*I.dc*sin(phi)+ Lhg*(I.mf/I.m)*(I.d*cos(theta - gam_w)...
    + I.dc*cos(alpha_w)) + Mac;
F_nl(5) = phi_d;
F_nl(6) = - (I.mw*I.mf/I.m)*I.d*I.dc*(theta_d^2)*sin(phi) ...
    + (L + Lvg)*(I.e - (I.mw/I.m)*I.dc)*cos(alpha_w)-I.k*(phi - I.alfr) ...
    + D*(I.e - (I.mw/I.m)*I.dc)*sin(alpha_w) ...
    - Lt*(I.mw/I.m)*I.dc*cos(alpha_t) - Dt*(I.mw/I.m)*I.dc*sin(alpha_t)...
    + T*(I.mw/I.m)*I.dc*sin(phi) + Mac+ Lhg*(I.mf/I.m)*I.dc*cos(alpha_w);

x_dot = inv(M)*F_nl';
```

H.   Linerization Function: *linize.m*

```
function [A,B] = linize(xeq,I,o)

% States
u = xeq(1);
w = xeq(2);
theta = xeq(3);
theta_d = xeq(4);
phi = xeq(5);
phi_d = xeq(6);
h = xeq(3) - atan2(xeq(2),xeq(1));

% Controls
df = xeq(8);
del = xeq(7);
T = xeq(9);

% Bring in Lift/Drag components
%update physical constants
K0 = 6.8757e-006;        % /ft
h0 = 5000;           % ft
% h
h_t = h + h0;
rho0 = 0.00238;      % lb-s^2/ft^4
rho = rho0*(1-K0*h_t)^(4.25);   %slug/ft^3 air


% Wing velocity, determined using velocity at system CM and included
% induced velocity due to rotation of Wing CM about system CM
xw_d = u - (I.mf/I.m)*I.dc*(theta_d + phi_d)*sin(phi);
yw_d = w - (I.mf/I.m)*(I.d*theta_d + I.dc*(theta_d + phi_d)*cos(phi));

gam_w = theta - atan2(yw_d,xw_d);

% Tail velocity, determined using velocity at system CM and included
% induced velocity due to rotation of Tail CM about system CM
xt_d = u + (I.mw/I.m)*I.dc*(theta_d + phi_d)*sin(phi);
yt_d = w + (I.d + I.f)*theta_d + (I.mw/I.m)*(I.dc*(theta_d + ...
    phi_d)*cos(phi));

gam_t = theta - atan2(yt_d,xt_d);

% Angle of Attack for Wing and Tail
alpha_w = theta + phi - gam_w;

alpha_t = alpha_w*(1 - I.deda) - I.alfr - I.eps0;

% for the system
gam = theta - atan2(w,u);
V = sqrt(u^2 + w^2);

gam_1 = [gam_w
    gam_t
    gam];

for n = 1:length(gam_1)
    if abs(gam_1(n)) <= 1e-4
        gam_1(n) = 0;
    else
        gam_1(n) = gam_1(n);
    end
end
```

```
gam_w = gam_1(1);
gam_t = gam_1(2);
gam = gam_1(3);
%% Lift/Drag

% for the wing
Vw = sqrt(xw_d^2 + yw_d^2);    % Calculates Velocity Scalar from x and y
Q = 0.5*rho*(Vw^2);            % Calculates Dynamic Pressure

% for the tail
Vt = sqrt(xt_d^2 + yt_d^2);
Qt = 0.5*rho*(Vt^2);

%% Moment of Inertia Terms
Is1 = I.Iw + I.If + (I.mf*I.mw/I.m)*(I.dc^2 + I.d^2 + 2*I.d*I.dc*cos(phi));
Iw1 = I.Iw + (I.mf*I.mw/I.m)*(I.d^2 + I.d*I.dc*cos(phi));

Is2 = I.Iw + (I.mf*I.mw/I.m)*(I.dc^2 + I.d*I.dc*cos(phi));
Iw2 = I.Iw + (I.mf*I.mw/I.m)*I.dc^2;


%% Lift/Drag/Moment Coefficients
eta = ((Qt*I.St)/(Q*I.S));

CLde = I.Cldelta;
CLdf = I.Clalfw*I.tau_f*(I.Sf/I.S);

CLw_1 = I.Cl0 + I.Clalfw*alpha_w;
CLt_1 = (I.Clt0 + I.Clalft*(alpha_t) +  I.Clq*eta*theta_d);
CDw_1 = I.Cd0 + I.Cdalfw*alpha_w;
CDt_1 = I.Cdt0 + I.Cdalft*alpha_t;

CLw_u = 0;
CLt_u = 0;
CDw_u = I.R*(CLw_u^2);
CDt_u = I.Rt*(CLt_u^2);

CLw_a = I.Clalfw;
CLt_a = I.Clalft;
CDw_a = I.Cdalfw;
CDt_a = I.Cdalft;

CDwt_u = CDw_u + eta*CDt_u;
CDwt_a = CDw_a + eta*CDt_a*(1 - I.deda);
CLwt_u = CLw_u + eta*CLt_u;
CLwt_a = CLw_a + eta*CLt_a*(1 - I.deda);

CLwf_a = CLwt_a;
CL1 = CLw_1 + eta*CLt_1;
CD1 = CDw_1 + eta*CDt_1;

%% Force Terms
Xu = -((Q*I.S)/(I.m*u))*(2*CD1 + CDwt_u);
Xw = -((Q*I.S)/(I.m*u))*(CDwt_a - CL1);
Xtheta = 0;
Xq = 0;
Xphi = 0;
Xphid = 0;

Zu = -(Q*I.S)/(I.m*u)*(2*CL1 + CLwt_u);
Zw = -(Q*I.S)/(I.m*u)*(CLwt_a + CD1);
Ztheta = 0;
Zq = -Q*I.S*I.c*I.Clq/(2*I.m*u);
Zphi = 0;
```

```
Zphid = 0;

%% Moment Terms
%for the system
Mu = 0;
Cma = (CLwf_a*(I.e - (I.mw/I.m)*(I.dc) + (I.mf/I.m)*I.d) + ...
    I.Cd0*(I.e - (I.mw/I.m)*I.dc)...
    - eta*I.Clalft*((I.mw/I.m)*(I.dc + I.d) + I.f)*(1 - I.deda))/I.c;
Mw = (Q*I.S*I.c)/(u)*Cma;

Mw_d = 0;
% Mw_d = 0 is an assumption, since the value is determined from flight
%testing, it is normal to ignore in calculations (but remember
%to state it as an assumption!)

Mphid = 0;
Cmq = -2.0*CLt_a*(eta)*((-I.f - (I.mw/I.m)*(I.d + I.dc*cos(phi)))^2 ...
    + ((I.mw/I.m)*I.dc*sin(phi))^2)/(I.c^2);
Mq = Cmq*(I.c/(2*u))*((Q*I.S*I.c));

%for the wing
a = sqrt((I.e*cos(phi) - (I.mw/I.m)*I.dc*cos(phi) + (I.mf/I.m)*I.d)^2 ...
    + (I.e*sin(phi) - (I.mw/I.m)*I.dc*sin(phi))^2);
b = sqrt((-I.f - (I.mw/I.m)*(I.d + I.dc*cos(phi)))^2 ...
    + ((I.mw/I.m)*I.dc*sin(phi))^2);

Cphia = (a*I.Clalfw + I.Cd0*(I.e - (I.mw/I.m)*I.dc))/(I.c);

Pu = 0;
Pw = (Q*I.S*I.c)/(u)*Cphia;
Pu = 0;
Pw_d = 0;        %% since I am making p=r=0 for all time
Pphid = 0;
Cpq = -.2*CLt_a*(eta)*((-I.f - I.d - I.dc*cos(phi))^2 ...
    + (I.dc*sin(phi))^2)/(I.c^2);
Pq = Cpq*(I.c/(2*u))*((Q*I.S*I.c));

Xphi = 0;
Xphid = 0;

Zphi = 0;
Zphid = 0;
Mphid = 0;
Pphid = 0;

%% M*x_dot = K*x....A = inv(M)*K

M = [1 0 0 0 0 0
    0 1 0 0 0 0
    0 0 1 0 0 0
    0 0 0 Is1 0 Is2
    0 0 0 0 1 0
    0 0 0 Iw1 0 Iw2]

K = [Xu Xw (- I.g) -w Xphi (Xphid)
    Zu Zw Xtheta u Zphi Zphid
    0 0 0 1 0 0
    Mu Mw 0 Mq 0 Mphid
    0 0 0 0 0 1
    Pu Pw 0 Pq (-I.k) Pphid]

%% Controls Contributions
Z_del_u = -(Q*I.S/I.m)*I.Cldelta;
X_del_u = 0;
```

```
Cmdel = -(I.Cldelta*(eta)*sqrt((I.f + (I.mw/I.m)*(I.d ...
    + I.dc*cos(phi)))^2 + ((I.mw/I.m)*I.dc*sin(phi) )^2))/I.c;
M_del = (Q*I.S*I.c)/(Is1)*Cmdel;

Cpdel = -(I.Cldelta*(eta)*(I.mw/I.m)*I.dc*cos(alpha_w))/I.c;
P_del = (Q*I.S*I.c)/(Iw2)*Cpdel;


CLdf = I.Clalfw*I.tau_f*(I.Sf/I.S);

Z_df_u = -(Q*I.S/I.m)*CLdf;
X_df_u = 0;

Cpdf = -CLdf*((I.e - I.dc)/I.c);
P_df = (Q*I.S*I.c)/(Iw2)*Cpdf;

Cmdf = -CLdf*(sqrt(((I.mf/I.m)*I.d + (I.e -(I.mw/I.m)*I.dc)*cos(phi))^2 ...
     + (-(I.e + (I.mw/I.m))*I.dc*sin(phi))^2)/I.c);
M_df = (Q*I.S*I.c)/(Is1)*Cmdf;

%% Final Output
B = [X_del_u X_df_u
    Z_del_u Z_df_u
    0 0
    M_del M_df
    0 0
    P_del P_df];
% Remember...no controls on should be on kinematic angles (theta & phi)

A = inv(M)*K
```

I.   ODE File with Linear Controller: *4dof.m*

```
%function to integrate history of 4 dof longitudinal model
function [x_dot] = dof4(t,x)

global  rootsA xeq preper I o E K phi_big gamma_big index yk xk od xk_f lk
%% Bring in initial guess states
u = x(1);
w = x(2);
theta = x(3);
theta_d = x(4);
phi = x(5);
phi_d = x(6);
h = x(3) - atan2(x(2),x(1));
del = xeq(7);
df = xeq(8);
T = xeq(9);

index = index + 1;
%% Gust Velocity

V_vg = 0;
V_hg = 0;
if t == 10
    if od == 2
    load yk_f
    yk = yk_f;
    end
end

if t <= (10 + 2*I.dm)
```

```
    if t >= 10
        o = 1;
        V_vg = (I.Uvg/2)*(1 - cos(pi*(t- 10)/I.dm));
        V_hg = (I.Uhg/2)*(1 - cos(pi*(t- 10)/I.dm));
    end
else
    V_vg = 0;
    V_hg = 0;
    o = 0;
end
dist = zeros(12,1);
dist1 = zeros(6,1);
%% Get new controls (linear), uses same gains, etc as linear model
if od == 2
    D = [1 0
        0 1];
    C1=[eye(14,14)
        zeros(2,14)];
    D1=[zeros(14,2)
        D];
    ym = [0 0]';

    dist = zeros(12,1);

    if t <= (10 + 2*I.dm)
        if t >= 10

        liftdragDISCRETE

        V_vg = (I.Uvg/2)*(1 - cos(pi*(t- 10)/I.dm));
        V_hg = (I.Uhg/2)*(1 - cos(pi*(t- 10)/I.dm));

        CLdf = I.Clalfw*I.tau_f*(I.Sf/I.S);

        L_vg = (rho/2)*V_vg*Vw*I.S*I.Clalfw;
        L_hg = (rho/2)*(2*V_hg*Vw + V_hg^2)*I.S*(I.Cl0 + ...
            I.Clalfw*alpha_w + CLdf*xeq(8));

        V_bar = (sqrt(xeq(1)^2 + xeq(2)^2)); %velocity magnitude from u & w
        dist(1,1) = ((L_vg + L_hg)*sin(theta - gam_w))/V_bar;
        dist(2,1) = -(L_vg + L_hg)*cos(theta - gam_w))/V_bar;
        dist(3,1) = 0;
        dist(4,1) = (L_vg*((I.e - (I.mw/I.m)*I.dc) + (I.mf/I.m)*I.d)+ ...
            L_hg*(I.mf/I.m)*(I.d*cos(theta - gam_w) ...
            + I.dc*cos(alpha_w)))/V_bar;
        dist(5,1) = 0;
        dist(6,1) = (L_vg*(I.e - (I.mw/I.m)*I.dc)+ ...
            L_hg*(I.mf/I.m)*I.dc*cos(alpha_w))/V_bar;

        dist(1:6,1) = inv(M)*dist(1:6,1);
        end
    else
        dist = zeros(12,1);
    end

    if t == 0
        u_0 = [0 0]';

        x_0 = [zeros(12,1)];

        xk = x_0;
        lk = u_0;
    else
        lk = E*ym - K*yk;
```

```
        end

        xk1(:,index + 1) = phi_big*xk + gamma_big*lk + dist;

        xk = xk1(:,index + 1);
        xk_f = xk;

        if t < 10
            save xk_f
        end
        l(:,index) = lk;

        y(:,index) = C1*[xk;lk]+ D1*lk;

        yk = y(1:12,index);

        del = xeq(7) + lk(1);
        df = xeq(8) + lk(2);

        if del > ((7/45)*pi)
            del = ((7/45)*pi);
        elseif del < -((7/60)*pi)
            del = -((7/60)*pi);
        end

        if df > ((11/90)*pi)
            df = ((11/90)*pi);
        elseif df < -((1/12)*pi)
            df = -((1/12)*pi);
        end
end


%% Define angles and determine Lift and Drag
u = x(1);
w = x(2);
theta = x(3);
theta_d = x(4);
phi = x(5);
phi_d = x(6);
h = x(3) - atan2(x(2),x(1));

liftdrag         % Calls liftdrag function


%% Expression for x_dot

EOM              % Calls EOM function

h = xeq(3) - atan2(xeq(2),xeq(1));
if h <= 1e-7          % Avoids machine zero error
    h = 0;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%set a timer
%calculate percent complete
if t == 0
    preper = 0;
end
if 100*t/I.tf >= preper + 1
    clc
    preper = floor(100*t/I.tf);
    disp('equillibrium state')
```

```
        disp(xeq(1:6))
        disp('equilibrium altitude')
        disp(h)
        disp('equilibrium control')
        disp(xeq(7:8))
        disp('equilibrium thrust')
        disp(xeq(9))
        disp('eigenvalues of state jacobian')
        disp(rootsA)
        disp('natural frequency of wing')
        disp(I.om_wing)
        disp('percent complete')
        disp(preper)
end
```

J.   ODE File with Nonlinear Controller: *4dof_nl.m*

```
%function to integrate history of 4 dof longitudinal model
function [x_dot] = dof4_nl(t,x)

global  rootsA xeq preper I od theta_dd phi_dd o theta_d df del
%% Bring in initial guess states

u = x(1);
w = x(2);
theta = x(3);
theta_d = x(4);
phi = x(5);
phi_d = x(6);
h = x(3) - atan2(x(2),x(1));
del = xeq(7);
df = xeq(8);
T = xeq(9);


%% Gust Velocity

V_vg = 0;
V_hg = 0;
if t <= (I.ts + 2*I.dm)
    if t >= I.ts
        o = 1;
        V_vg = (I.Uvg/2)*(1 - cos(pi*(t- I.ts)/I.dm));
        V_hg = (I.Uhg/2)*(1 - cos(pi*(t- I.ts)/I.dm));
    end
else
    V_vg = 0;
    V_hg = 0;
    o = 0;
end
%% Define Angles and Determine Lift and Drag

liftdrag                 % Calls liftdrag function

Cmac = -0.045;
Mac = Q*I.S*I.c*Cmac;
%% Get new controls (nonlinear)

if od == 2
    if t == 0
        theta_dd = 0;   %initialize as zero, for trim initial condition
        phi_dd = 0;
    else
    end
```

```matlab
        P = [0.005*I.om_wing 0
             0 0.3*I.om_wing];       %% Gains for one-minus-cosine input
%  P = [1.55*I.om_wing 0
%      0 2.25*I.om_wing];            %% Gains for sinusoidal input

%%Using theta_d and phi_d as controlled states
    g1 = ((I.mw*I.mf)/I.m)*(I.d*I.dc*(2*theta_d*phi_d + phi_d^2))...
        + Q*I.S*(I.Cl0 + I.Clalfw*alpha_w)*((I.e - (I.mw/I.m)*I.dc)...
        *cos(alpha_w) + (I.mf/I.m)*I.d*cos(theta - gam_w)) ...
        + Q*I.S*(I.Cd0+ I.Cdalfw*alpha_w)*((I.e - (I.mw/I.m)*I.dc)*...
        sin(alpha_w) + (I.mf/I.m)*I.d*sin(theta - gam_w)) - ...
        Qt*I.St*(I.Clt0 + I.Clalft*(alpha_t) + I.Clq*theta_d)*...
        ((I.mw/I.m)*I.dc*cos(alpha_t)+ ((I.mw/I.m)*I.d + I.f)*...
        cos(theta - gam_t))- Qt*I.St*(I.Cdt0 + I.Cdalft*(alpha_t))*...
        ((I.mw/I.m)*I.dc*sin(alpha_t) + ((I.mw/I.m)*I.d + I.f)*...
        sin(theta - gam_t)) + T*(I.mw/I.m)*I.dc*sin(phi) + Mac;
    h_df1 = Q*I.S*CLdf*((I.e - (I.mw/I.m)*I.dc)*cos(alpha_w) + ...
        (I.mf/I.m)*I.d*cos(theta - gam_w));
    h_de1 = -Qt*I.St*CLde*((I.mw/I.m)*I.dc*cos(alpha_t) + ...
        ((I.mw/I.m)*I.d + I.f)*cos(theta - gam_t));

    g2 = - ((I.mw*I.mf)/I.m)*I.d*I.dc*(theta_d^2)*sin(phi)...
        - I.k*(phi - I.alfr) + Q*I.S*(I.Cl0 + I.Clalfw*alpha_w)*...
        (I.e - (I.mw/I.m)*I.dc)*cos(alpha_w) + Q*I.S*(I.Cd0 + ...
        I.Cdalfw*alpha_w)*(I.e - (I.mw/I.m)*I.dc)*sin(alpha_w)...
        - Qt*I.St*(I.Clt0 + I.Clalft*(alpha_t) + I.Clq*theta_d)*...
        (I.mw/I.m)*I.dc*cos(alpha_t) - Qt*I.St*(I.Cdt0 + ...
        I.Cdalft*(alpha_t))*(I.mw/I.m)*I.dc*sin(alpha_t)...
        + T*(I.mw/I.m)*I.dc*sin(phi) + Mac;

    h_df2 = Q*I.S*CLdf*(I.e - (I.mw/I.m)*I.dc)*cos(alpha_w);
    h_de2 = -Qt*I.St*CLde*(I.mw/I.m)*I.dc*cos(alpha_t);

    Is1 = I.Iw + I.If + (I.mf*I.mw/I.m)*(I.dc^2 + I.d^2 + ...
        2*I.d*I.dc*cos(phi));
    Is2 = I.Iw + (I.mf*I.mw/I.m)*(I.dc^2 + I.d*I.dc*cos(phi));

    Iw1 = I.Iw + (I.mf*I.mw/I.m)*(I.dc^2 + I.d*I.dc*cos(phi));
    Iw2 = I.Iw + (I.mf*I.mw/I.m)*(I.dc^2);

    Ibig = [Is1 Is2; Iw1 Iw2];

    H = -inv(Ibig)*[h_df1 h_de1; h_df2 h_de2];
    G = inv(Ibig)*[g1; g2];

    df_de(1,:) =  inv(H)*(P*([theta_d;phi_d] + ...
        [theta - xeq(3); phi - xeq(5)])+ G);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


    df = df_de(1,1);
    del = df_de(1,2);

    %%  Make sure the control deflections stay within the operation
    %%  boundary
    if del > ((7/45)*pi)
        del = ((7/45)*pi);
    elseif del < -((7/60)*pi)
        del = -((7/60)*pi);
    end

    if df > ((11/90)*pi)
```

```
        df = ((11/90)*pi);
    elseif df < -((1/12)*pi)
        df = -((1/12)*pi);
    end

    df_de2(1,:) = [df;del];
else
end
CLw = I.Cl0 + I.Clalfw*alpha_w + CLdf*df;
CLt = I.Clt0 + I.Clalft*(alpha_t) + CLde*del + I.Clq*theta_d;


CDw = I.Cd0 + I.Cdalfw*alpha_w;
CDt = I.Cdt0 + I.Cdalft*alpha_t;

L = CLw*Q*I.S;
D = Q*I.S*CDw;

Lt = CLt*Qt*I.St;
Dt = Qt*I.St*CDt;
%% Expression for x_dot

EOM                 % Calls EOM function

theta_d = x_dot(3);
theta_dd = x_dot(4);
phi_dd = x_dot(6);


h = xeq(3) - atan2(xeq(2),xeq(1));
if h <= 1e-7        % avoids machine zero error
    h = 0;
end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%set a timer
%calculate percent complete
if t == 0
    preper = 0;
end
if 100*t/I.tf >= preper + 1
    clc
    preper = floor(100*t/I.tf);
    disp('equillibrium state')
    disp(xeq(1:6))
    disp('equilibrium altitude')
    disp(h)
    disp('equilibrium control')
    disp(xeq(7:8))
    disp('equilibrium thrust')
    disp(xeq(9))
    disp('eigenvalues of state jacobian')
    disp(rootsA)
    disp('natural frequency of wing')
    disp(I.om_wing)
    disp('percent complete')
    disp(preper)
end
```

K. Open-loop Simulation File: *openloop.m*

```
%% Linear Controller
h_1 = 0.002;
T_1 = 0.002;
```

```
t_f = I.tf;

wn_1 = 10;
zeta_1 = 0.6;

A_del = [0 1
    -wn_1^2 -2*wn_1*zeta_1];
A_T = [0 1
    -wn_1^2 -2*wn_1*zeta_1];
A_dT = [A_del zeros(2,2)
    zeros(2,2) A_T];

A_new = [A B(:,1) zeros(6,1) B(:,2) zeros(6,1)
    zeros(4,6) A_dT];
B_new = [zeros(6,2)
    0 0
    100 0
    0 0
    0 100];

H = [1 0 0 0 0 0 0 0 0 0
    0 0 1 0 0 0 0 0 0 0];

H1 = [1 0 0 0 0 0 0 0 0 0
    0 0 1 0 0 0 0 0 0 0];

D = zeros(2,2);

cont = [B_new A_new*B_new (A_new^2)*B_new (A_new^3)*B_new ...
    (A_new^4)*B_new (A_new^5)*B_new];
disp('Controllability Rank:')
cont_r = rank(cont)

Obs = [H; H*A_new; H*(A_new^2); H*(A_new^3); H*(A_new^4); H*(A_new^5)];
disp('Observability Rank:')
obs_r = rank(Obs)

Abig = [A_new zeros(10,2)
    H1 zeros(2,2)];
Bbig = [B_new
    zeros(2,2)];

[phi_big,gamma_big] = c2d(Abig,Bbig,h_1);
[phi,gamma] = c2d(A_new,B_new,h_1);

[phi_s,gamma_s] = c2d(A,B,h_1);  %% for original system
[phi_3,gamma_3] = c2d(A(1:4,1:4),B(1:4,:),h_1);

[X12, X22] = QPMCALC((phi - eye(10)), gamma, H, D)

phi = [];
Q1 = [10 1 1 5e2 1 1e7];
Qp = [1 1e4 1 1e2];
R = [1 1e6];

Q2 = diag([Q1 R Qp]);
S = diag([1 1]);
N = zeros(12,2);
[K,Qd,Rd,Nd,s,e] = LQRDJV(Abig,Bbig,Q2,S,N,T_1);


K1 = K(:,1:10);
K2 = K(:,11:12);
E = K2*X22 + K1*X12;
```

```
nframes = t_f/h_1;

D = [1 0
     0 1];
C1=[eye(14,14)
     zeros(2,14)];
D1=[zeros(14,2)
     D];


u_0 = [0 0]';

x_0 = [zeros(12,1)] + [I.ui;I.wi;zeros(10,1)];

xks(1:6,1) = x_0(1:6);
xks3(1:4,1) = x_0(1:4);
count = 0;
xk1_s = xks;
xk1_s3 = xks3;
xk = x_0;
xk1 = xk;
xk2 = xk;
lk = u_0;
ym = [0 0]';

time(1) = 0;
dist = zeros(12,1);
dist1 = zeros(6,1);

nspace = linspace(0,I.tf,I.timesteps);
for i = 1:nframes

    lk = u_0;
    lk1(:,i) = lk;


    if time(i) == I.ts
        xk = xk + [I.ui;I.wi;zeros(10,1)];
        xk2 = xk2 + [I.ui;I.wi;zeros(10,1)];
    end

    if time(i) <= (I.ts + 2*I.dm)
        if time(i) >= I.ts
            o = 1;
            index = index + 1;
            V_vg1(index) = (I.Uvg/2)*(1 - cos(pi*(time(i)-I.ts)/I.dm));
            V_hg1(index) = (I.Uhg/2)*(1 - cos(pi*(time(i)-I.ts)/I.dm));
            V_vg = V_vg1(index);
            V_hg = V_hg1(index);
        end
        liftdragDISCRETE

        CLdf = I.Clalfw*I.tau_f*(I.Sf/I.S);

        L_vg = (rho/2)*V_vg*Vw*I.S*I.Clalfw;
        L_hg = (rho/2)*(2*V_hg*Vw + V_hg^2)*I.S*(I.Cl0 + ...
            I.Clalfw*alpha_w + CLdf*xeq(8));

        V_bar = (sqrt(xeq(1)^2 + xeq(2)^2)); %velocity magnitude from u & w
        dist(1,1) = ((L_vg + L_hg)*sin(theta - gam_w))/V_bar;
        dist(2,1) = -(L_vg + L_hg)*cos(theta - gam_w)/V_bar;
        dist(3,1) = 0;
        dist(4,1) = (L_vg*((I.e - (I.mw/I.m)*I.dc) + (I.mf/I.m)*I.d)+ ...
            L_hg*(I.mf/I.m)*(I.d*cos(theta - gam_w) ...
```

```
        + I.dc*cos(alpha_w)))/V_bar;
    dist(5,1) = 0;
    dist(6,1) = (L_vg*(I.e - (I.mw/I.m)*I.dc)+ ...
        L_hg*(I.mf/I.m)*I.dc*cos(alpha_w))/V_bar;

    dist(1:6,1) = inv(M)*dist(1:6,1);
    dist1 = dist(1:6,1);
    if time == 0
        dist = zeros(12,1);
    end

    l(:,i) = lk;

    xk1(:,i + 1) = phi_big*xk + gamma_big*lk + dist;
    xk12(:,i + 1) = phi_big*xk2 + dist;

    xk1_s(:,i + 1) = phi_s*xks(1:6) + dist1;%

    xk = xk1(:,i + 1);
    xk2 = xk12(:,i + 1);

    xks = xk1_s(:,i + 1);
else
    dist = zeros(12,1);
    dist1 = zeros(6,1);

    Vg1(i) = 0;

    l(:,i) = lk;

    xk1(:,i + 1) = phi_big*xk + gamma_big*lk + dist;
    xk12(:,i + 1) = phi_big*xk2 + dist;

    xk1_s(:,i + 1) = phi_s*xks(1:6) + dist1;%

    xk = xk1(:,i + 1);

    xk2 = xk12(:,i + 1);

    xks = xk1_s(:,i + 1);
end

y(:,i) = C1*[xk;lk]+ D1*lk;
y2(:,i) = C1*[xk2;lk];
y_s(:,i) = xk1_s(:,i + 1);
yk = y(1:12,i);
%        pause
count = count + h_1;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%set a timer
%calculate percent complete
if time(i) == 0
    preper2 = 0;
end
if 100*time(i)/I.tf >= preper2 + 1
    clc
    preper2 = floor(100*time(i)/I.tf);
    disp('equillibrium state')
    disp(xeq(1:6))
    disp('equilibrium control')
    disp(xeq(7:8))
    disp('equilibrium thrust')
    disp(xeq(9))
```

```
            disp('eigenvalues of state jacobian')
            disp(rootsA)
            disp('natural frequency of wing')
            disp(I.om_wing)
            disp('percent complete')
            disp(preper2)
        end
        time(i + 1) = i*h_1;
end
%% Integrate time histories
od = 1;
x0_ode = [xeq(1)
    xeq(2)
    xeq(3)
    xeq(4)
    xeq(5)
    xeq(6)];
del_eq = [xeq(7)
    xeq(8)
    xeq(9)];

tspan = [0:0.005:(I.ts - 0.01)];
[t1,x1] = ode45('dof4_nl', tspan,x0_ode);
x0_ode2(:,1) = x1(end,:)' + [I.ui;I.wi;0;0;0;0];%

tspan2 = [I.ts:0.005:I.tf];
[t2,x2] = ode45('dof4_nl', tspan2,x0_ode2);

%%Use code below instead of ode45 when you need a constant step size,
%%solution takes significantly longer to run:
% tspan = [0,(I.ts - 0.01)];
% [t1,x1] = runge('dof4_nl', tspan, I.timesteps,x0_ode);
% x0_ode2(:,1) = x1(end,:)' + [I.ui;I.wi;0;0;0;0];
%
% tspan2 = [I.ts,I.tf];
% [t2,x2] = runge('dof4_nl', tspan2, I.timesteps,x0_ode2);

t = [t1;t2];
x = [x1;x2];

x_f(:,1) = sqrt(x(:,1).^2 + x(:,2).^2);
x_f(:,2) = x(:,3) - atan2(x(:,2),x(:,1));
h = [];
h(:,1) = x(:,3) - atan2(x(:,2),x(:,1));

for i = 1:length(h)
    if abs(h(i,1)) <= 1e-4
        h(i,1) = 0;
    else
        h(i,1) = h(i,1);
    end
end
toc
if od == 1
    x1_ode = x;
    t1_ode = t;
    x_f1_ode = x_f;
    h1_ode = h;
else
    x2_ode = x;       %% controlled case
    t2_ode = t;
    x_f2_ode = x_f;
    h2_ode = h;
end
```

```
%% Plots
figure (8)
subplot(2,1,1)
plot(time(1:300000),y2(1,:) + xeq(1),t1_ode,x1_ode(:,1)),...
    ylabel('u (ft/s)'),xlabel('time (s)'),...
    legend('Linear Open-Loop Response','Nonlinear Open-Loop Response')
subplot(2,1,2)
plot(time(1:300000),y2(2,:) + xeq(2),t1_ode,x1_ode(:,2)),...
    ylabel('w (ft/s)'),xlabel('time (s)')


figure (9)
subplot(2,1,1)
plot(time(1:300000),(y2(3,:) + xeq(3))*(180/pi),t1_ode,x1_ode(:,3)*...
    (180/pi)),ylabel('theta (deg)'),xlabel('time (s)'),...
    legend('Linear Open-Loop Response','Nonlinear Open-Loop Response')
subplot(2,1,2)
plot(time(1:300000),(y2(5,:) + xeq(5))*(180/pi),t1_ode,x1_ode(:,5)*...
    (180/pi)),ylabel('phi (deg)'),xlabel('time (s)')
```

L.   Closed-loop Simulation File with Linear Controller: *linearcontrol.m*

```
%% Linear Controller
h_1 = 0.002;
T_1 = 0.002;
t_f = I.tf;

wn_1 = 10;
zeta_1 = 0.6;

A_del = [0 1
    -wn_1^2 -2*wn_1*zeta_1];
A_T = [0 1
    -wn_1^2 -2*wn_1*zeta_1];
A_dT = [A_del zeros(2,2)
    zeros(2,2) A_T];

A_new = [A B(:,1) zeros(6,1) B(:,2) zeros(6,1)
    zeros(4,6) A_dT];
B_new = [zeros(6,2)
    0 0
    100 0
    0 0
    0 100];

H = [1 0 0 0 0 0 0 0 0 0
    0 0 1 0 0 0 0 0 0 0];

H1 = [1 0 0 0 0 0 0 0 0 0
    0 0 1 0 0 0 0 0 0 0];

D = zeros(2,2);

cont = [B_new A_new*B_new (A_new^2)*B_new (A_new^3)*B_new ...
    (A_new^4)*B_new (A_new^5)*B_new];
disp('Controllability Rank:')
cont_r = rank(cont)

Obs = [H; H*A_new; H*(A_new^2); H*(A_new^3); H*(A_new^4); H*(A_new^5)];
disp('Observability Rank:')
obs_r = rank(Obs)
```

```
Abig = [A_new zeros(10,2)
    H1 zeros(2,2)];
Bbig = [B_new
    zeros(2,2)];

[phi_big,gamma_big] = c2d(Abig,Bbig,h_1);
[phi,gamma] = c2d(A_new,B_new,h_1);

[phi_s,gamma_s] = c2d(A,B,h_1);  %% for original system
[phi_3,gamma_3] = c2d(A(1:4,1:4),B(1:4,:),h_1);

[X12, X22] = QPMCALC((phi - eye(10)), gamma, H, D)

phi = [];
Q1 = [10 1e2 1 1e2 1 1e3];
R = [5e4 1 1 1e5];
Qp = [1 1];

Q2 = diag([Q1 R Qp]);
S = diag([1 1]);
N = zeros(12,2);
[K,Qd,Rd,Nd,s,e] = LQRDJV(Abig,Bbig,Q2,S,N,T_1);


K1 = K(:,1:10);
K2 = K(:,11:12);
E = X22 + K1*X12;

nframes = t_f/h_1;

D = [1 0
    0 1];
C1=[eye(14,14)
    zeros(2,14)];
D1=[zeros(14,2)
    D];

u_0 = [0 0]';

x_0 = [zeros(12,1)];

xks(1:6,1) = x_0(1:6);
xks3(1:4,1) = x_0(1:4);
count = 0;
xk1_s = xks;
xk1_s3 = xks3;
xk = x_0;
xk1 = xk;
xk2 = xk;
lk = u_0;
ym = [0 0]';

time(1) = 0;
dist = zeros(12,1);
dist1 = zeros(6,1);

nspace = linspace(0,I.tf,I.timesteps);
for i = 1:nframes
    if abs(count-T_1) < 1e-8
        lk = E*ym - K*yk;
        %          lk = u_0;
        lk1(:,i) = lk;
        count = 0;
    end
```

```
    if time(i) == I.ts             % for instantaneous disturbance
        xk = xk + [I.ui;I.wi;zeros(10,1)]
        xk2 = xk2 + [I.ui;I.wi;zeros(10,1)]
    end
dist = zeros(12,1);
        dist1 = zeros(6,1);

        V_vg = 0;
        V_hg = 0;

if time(i) <= (I.ts + 2*I.dm)
    if time(i) >= I.ts
        o = 1;
        index = index + 1;
        V_vg1(index) = (I.Uvg/2)*(1 - cos(pi*(time(i)-I.ts)/I.dm));
        V_hg1(index) = (I.Uhg/2)*(1 - cos(pi*(time(i)-I.ts)/I.dm));
        V_vg = V_vg1(index);
        V_hg = V_hg1(index);
    end
        liftdragDISCRETE

    CLdf = I.Clalfw*I.tau_f*(I.Sf/I.S);

    L_vg = (rho/2)*V_vg*Vw*I.S*I.Clalfw;
    L_hg = (rho/2)*(2*V_hg*Vw + V_hg^2)*I.S*(I.Cl0 + ...
        I.Clalfw*alpha_w + CLdf*xeq(8));

    V_bar = (sqrt(xeq(1)^2 + xeq(2)^2)); %velocity magnitude from u & w
    dist(1,1) = ((L_vg + L_hg)*sin(theta - gam_w))/V_bar;
    dist(2,1) = -(L_vg + L_hg)*cos(theta - gam_w)/V_bar;
    dist(3,1) = 0;
    dist(4,1) = (L_vg*((I.e - (I.mw/I.m)*I.dc) + (I.mf/I.m)*I.d)+ ...
        L_hg*(I.mf/I.m)*(I.d*cos(theta - gam_w) ...
        + I.dc*cos(alpha_w)))/V_bar;
    dist(5,1) = 0;
    dist(6,1) = (L_vg*(I.e - (I.mw/I.m)*I.dc)+ ...
        L_hg*(I.mf/I.m)*I.dc*cos(alpha_w))/V_bar;

    dist(1:6,1) = inv(M)*dist(1:6,1);
    dist1 = dist(1:6,1);

        if time == 0
        dist = zeros(12,1);
    end

    l(:,i) = lk;

    xk1(:,i + 1) = phi_big*xk + gamma_big*lk + dist;
    xk12(:,i + 1) = phi_big*xk2 + dist;

    xk1_s(:,i + 1) = phi_s*xks(1:6) + dist1;%

    xk = xk1(:,i + 1);
    xk2 = xk12(:,i + 1);

    xks = xk1_s(:,i + 1);
else
    dist = zeros(12,1);
    dist1 = zeros(6,1);

    Vg1(i) = 0;

    l(:,i) = lk;
```

```
        xk1(:,i + 1) = phi_big*xk + gamma_big*lk + dist;
        xk12(:,i + 1) = phi_big*xk2 + dist;

        xk1_s(:,i + 1) = phi_s*xks(1:6) + dist1;%

        xk = xk1(:,i + 1);

        xk2 = xk12(:,i + 1);

        xks = xk1_s(:,i + 1);
    end

    y(:,i) = C1*[xk;lk]+ D1*lk;
    y2(:,i) = C1*[xk2;lk];
    y_s(:,i) = xk1_s(:,i + 1);
    yk = y(1:12,i);

    count = count + h_1;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%set a timer
%calculate percent complete
if time(i) == 0
    preper2 = 0;
end
if 100*time(i)/I.tf >= preper2 + 1
    clc
    preper2 = floor(100*time(i)/I.tf);
    disp('equillibrium state')
    disp(xeq(1:6))
    disp('equilibrium altitude')
    disp(h)
    disp('equilibrium control')
    disp(xeq(7:8))
    disp('equilibrium thrust')
    disp(xeq(9))
    disp('eigenvalues of state jacobian')
    disp(rootsA)
    disp('natural frequency of wing')
    disp(I.om_wing)
    disp('percent complete')
    disp(preper2)
end
    time(i + 1) = i*h_1;
end

%% Integrate time histories
for od = 1:2
    index = 0;
    x0_ode = [xeq(1)
        xeq(2)
        xeq(3)
        xeq(4)
        xeq(5)
        xeq(6)];
    del_eq = [xeq(7)
        xeq(8)
        xeq(9)];

    tspan = [0:0.005:9.99];
    %
    [t1,x1] = ode45('dof4', tspan,x0_ode);

    x0_ode2(:,1) = x1(end,:)' + [I.ui;I.wi;0;0;0;0];
```

```
    tspan2 = [10:0.005:I.tf];

    [t2,x2] = ode45('dof4', tspan2,x0_ode2);

%%Use code below instead of ode45 when you need a constant step size,
%%solution takes significantly longer to run:
%     tspan = [0,9.99];
%     [t1,x1] = runge('dof4', tspan, I.timesteps,x0_ode);
%     x0_ode2(:,1) = x1(end,:)' + [I.ui;I.wi;0;0;0;0];
%     tspan2 = [10,I.tf];
%     [t2,x2] = runge('dof4', tspan2, I.timesteps,x0_ode2);

    t = [t1;t2];
    x = [x1;x2];

    toc
    if od == 1
        x1_ode = x;
        t1_ode = t;
%        x_f1_ode = x_f;
%        h1_ode = h;
    else
        x2_ode = x;      %% controlled case
        t2_ode = t;
%        x_f2_ode = x_f;
%        h2_ode = h;
    end
end
%% Plots
figure (1)
subplot(2,1,1)
plot(time(1:150000),y2(1,:) + xeq(1),time(1:150000),y(1,:) + xeq(1)),...
    ylabel('u (ft/s)'),xlabel('time (s)'),...
    legend('Linear Open-Loop Response','PIF-NZSP-CRW Controller')
subplot(2,1,2)
plot(time(1:150000),y2(2,:) + xeq(2),time(1:150000),y(2,:) + ...
    xeq(2)),ylabel('w (ft/s)'),xlabel('time (s)')

figure (2)
subplot(2,1,1)
plot(time(1:150000),(y2(3,:) + xeq(3))*(180/pi),time(1:150000),(y(3,:) ...
    + xeq(3))*(180/pi)),ylabel('theta (deg)'),xlabel('time (s)'),...
    legend('Linear Open-Loop Response','PIF-NZSP-CRW Controller')
subplot(2,1,2)
plot(time(1:150000),(y2(5,:) + xeq(5))*(180/pi),time(1:150000),(y(5,:)...
    + xeq(5))*(180/pi)),ylabel('phi (deg)'),xlabel('time (s)')

figure (3)
subplot(2,1,1)
plot(time(1:150000),(y2(7,:) + xeq(7))*(180/pi),time(1:150000),(y(7,:) + ...
    xeq(7))*(180/pi)),ylabel('delta_e (deg)'),xlabel('time (s)'),...
    legend('Linear Open-Loop Response','PIF-NZSP-CRW Controller')
subplot(2,1,2)
plot(time(1:150000),y2(8,:),time(1:150000),y(8,:)),...
    ylabel('deltae_d (deg/s)'),xlabel('time (s)')

figure (4)
subplot(2,1,1)
plot(time(1:150000),(y2(9,:) + xeq(8))*(180/pi),time(1:150000),(y(9,:)...
    + xeq(8))*(180/pi)),ylabel('delta_f (deg)'),xlabel('time (s)'),...
    legend('Linear Open-Loop Response','PIF-NZSP-CRW Controller')
subplot(2,1,2)
plot(time(1:150000),(y2(10,:))*(180/pi),time(1:150000),(y(10,:))*(180/pi))...
```

```
,ylabel('deltaf_d (deg/s)'),xlabel('time (s)')

figure (5)
subplot(2,1,1)
plot(t1_ode,x1_ode(:,1),t2_ode,x2_ode(:,1)),ylabel('u (ft/s)'),...
    xlabel('time (s)'),legend('Nonlinear Open-Loop Response',...
    'PIF-NZSP-CRW Controller')
subplot(2,1,2)
plot(t1_ode,x1_ode(:,2),t2_ode,x2_ode(:,2)),ylabel('w (ft/s)'),...
    xlabel('time (s)')

figure (6)
subplot(2,1,1)
plot(t1_ode,x1_ode(:,3)*(180/pi),t2_ode,x2_ode(:,3)*(180/pi)),...
    ylabel('theta (deg)'),xlabel('time (s)'),...
    legend('Nonlinear Open-Loop Response','PIF-NZSP-CRW Controller')
subplot(2,1,2)
plot(t1_ode,x1_ode(:,5)*(180/pi),t2_ode,x2_ode(:,5)*(180/pi)),...
    ylabel('phi (deg)'),xlabel('time (s)')
```

M.   Closed-loop Simulation File with Nonlinear Controller: *nonlinearcontrol.m*

```
%% Integrate time histories
for od = 1:2
x0_ode = [xeq(1)
    xeq(2)
    xeq(3)
    xeq(4)
    xeq(5)
    xeq(6)];
del_eq = [xeq(7)
    xeq(8)
    xeq(9)];

%%Use code below instead of ode45 when you need a constant step size,
%%solution takes significantly longer to run:
% tspan = [0,(I.ts-0.01)];
% [t1,x1] = runge('dof4_nl', tspan, I.timesteps,x0_ode);
%
% x0_ode2(:,1) = x1(end,:)' + [I.ui;I.wi;0;0;0;0];
% tspan2 = [I.ts,I.tf];
% [t2,x2] = runge('dof4_nl', tspan2, I.timesteps,x0_ode2);


tspan = [0:0.005:(I.ts-0.01)];
[t1,x1] = ode45('dof4_nl', tspan,x0_ode);

x0_ode2(:,1) = x1(end,:)' + [I.ui;I.wi;0;0;0;0];
tspan2 = [I.ts:0.005:I.tf];
[t2,x2] = ode45('dof4_nl', tspan2,x0_ode2);


t = [t1;t2];
x = [x1;x2];

toc
if od == 1
    x1_ode = x;
    t1_ode = t;
else
    x2_ode = x;      %% controlled case
    t2_ode = t;
end
```

```
end
%% Plots

figure (1)
subplot(2,1,1)
plot(t1_ode,x1_ode(:,1),t2_ode,x2_ode(:,1))...
    ,ylabel('u (ft/s)'),xlabel('time (s)'),...
    legend('Nonlinear Open-Loop Response','Lyapunov Controller')
subplot(2,1,2)
plot(t1_ode,x1_ode(:,2),t2_ode,x2_ode(:,2),t,repmat(xeq(2),1,length(t)))...
    ,ylabel('w (ft/s)'),xlabel('time (s)')

figure(2)
subplot(2,1,1)
plot(t1_ode,x1_ode(:,3)*180/pi,t2_ode,x2_ode(:,3)*180/pi),...
    ylabel('theta (deg)'),xlabel('time (s)'),...
    legend('Nonlinear Open-Loop Response','Lyapunov Controller')
subplot(2,1,2)
plot(t1_ode,x1_ode(:,5)*180/pi,t2_ode,x2_ode(:,5)*180/pi,t,repmat(xeq(5)...
    ,1,length(t))*180/pi),ylabel('phi (deg)'),xlabel('time (s)')
```

N. Discrete LQR Function (written by Dr. Valasek): *LQRDJV.m*

```
function [k,Qd,Rd,Nd,s,e] = lqrdjv(a,b,q,r,nn,Ts)
%%%%%function [k,s,e,Qd,Rd,Nd] = lqrdjv(a,b,q,r,nn,Ts)
%LQRDJV  Discrete linear quadratic regulator design from continuous
%        cost function.
%
%   ***********************************************************************
%
%            =====>  modified by J. Valasek, 5 Dec 94   <=====
%
%            This routine now sends back the Q^, R^, and M^ matrices.
%
%            NOTE: the order of the passed-back arguments is NOT the
%                  same as the original MATLAB version.
%
%   ***********************************************************************
%
% [K,S,E] = LQRD(A,B,Q,R,Ts) calculates the optimal feedback gain
% matrix K such that the discrete feedback law  u[n] = -K x[n]
% minimizes a discrete cost function equivalent to the continuous
% cost function
% J = Integral {x'Qx + u'Ru} dt
%                                              .
% subject to the continuous constraint equation:  x = Ax + Bu
%
% Also returned is S, the discrete Riccati equation solution, and
% the closed loop eigenvalues E = EIG(Ad-Bd*K).
%
% The gain matrix is determined by discretizing the continuous plant
% (A,B,C,D) and continuous weighting matrices (Q,R) using the sample
% time Ts and the zero order hold approximation. The gain matrix is
% then calculated using DLQR.
%
% [K,S,E] = LQRD(A,B,Q,R,N,Ts) includes the cross-term N that
% relates u to x in the cost function.
% J = Integral {x'Qx + u'Ru + 2*x'Nu}
%
% See also: C2D, LQED, DLQR, and LQR.

% Clay M. Thompson 7-16-90
% Copyright (c) 1986-93 by the MathWorks, Inc.
```

```
% Reference: This routine is based on the routine JDEQUIV.M by Franklin,
% Powell and Workman and is described on pp. 439-441 of "Digital Control
% of Dynamic Systems".

error(nargchk(5,6,nargin));
error(abcdchk(a,b));
[nx,na] = size(a);
[nb,nu] = size(b);

[nq,mq] = size(q);
if (nx ~= nq) | (nx ~= mq), error('A and Q must be the same size.'); end
[nr,mr] = size(r);
if (mr ~= nr) | (nu ~= mr), error('B and R must be consistent.'); end

if nargin==5,
  Ts = nn;
  nn = zeros(nb,nu);
else
  [nnn,mn] = size(nn);
  if (nnn ~= nx) | (mn ~= nu), error('N must be consistent with Q and R.'); end
end

% Check if q is positive semi-definite and symmetric
if any(eig(q) < -eps) | (norm(q'-q,1)/norm(q,1) > eps)
  disp('Warning: Q is not symmetric and positive semi-definite');
end
% Check if r is positive definite and symmetric
if any(eig(r) <= -eps) | (norm(r'-r,1)/norm(r,1) > eps)
  disp('Warning: R is not symmetric and positive definite');
end

% Discretize the state-space system.
[ad,bd] = c2d(a,b,Ts);

% --- Determine discrete equivalent of continuous cost function ---
n = nx+nu;
Za = zeros(nx); Zb = zeros(nx,nu); Zu = zeros(nu);
M = [ -a' Zb   q   nn
      -b' Zu  nn'  r
      Za  Zb   a   b
      Zb' Zu  Zb' Zu];
phi = expm(M*Ts);
phi12 = phi(1:n,n+1:2*n);
phi22 = phi(n+1:2*n,n+1:2*n);
QQ = phi22'*phi12;
QQ = (QQ+QQ')/2; % Make sure QQ is symmetric
Qd = QQ(1:nx,1:nx) ;
Rd = QQ(nx+1:n,nx+1:n) ;
Nd = QQ(1:nx,nx+1:n) ;

% Design the gain matrix using the discrete plant and discrete cost function
[k,s,e] = dlqr(ad,bd,Qd,Rd,Nd);
```

O.   Quadpartition Matrix Function (written by Dr. Valasek): *LQRDJV.m*

```
% qpmcalc.m
% This M-file assembles the Quad Partition Matrix (QPM) and returns
% the sub-matrices X12 and X22 from passed A, B, C, and D.
%
% ===================================================================
%
% written by:   J. Valasek
```

```
%                 WMU Aircraft Design and Control Laboratory
%                 19 January 1996
%
%  ====================================================================
%       DIGITAL FLIGHT CONTROL SYSTEMS: Analysis and Design
%                                by
%                          David R. Downing
%                          John Valasek
%
%    The University of Kansas - Division of Continuing Education
%
%                          1 September 2003
%  ====================================================================
function [X12, X22] = QPMCALC(A, B, C, D) ;

%
% .. determine dimenstions of passed matrices and vectors
%

[rowa, cola] = size(A) ;

[rowb, colb] = size(B) ;

[rowc, colc] = size(C) ;

[rowd, cold] = size(D) ;

%
% .. quad partition matrix and its inverse
%

qpm  = [A, B ; C, D]   ;

qpmi = inv(qpm)        ;

%
% .. break out the X12 and X22 quadrants
%

X12 = qpmi(1:rowa, cola+1:cola+colb)              ;

X22 = qpmi(rowa+1:rowa+rowc, cola+1:cola+colb) ;
```

P.   Constant Step Size ODE Solver (written by Dr. Hurtado): *runge.m*

```
function [time,y]=runge(fname,tspan,npt,yi)
% function [time,y]=runge(fname,tspan,npt,yi)
% fixed time step 4th order runge kutta
% fname : ode name
% tspan : [t0 tf]
% npt : number of points in discretization
% yi initial conditions
tf=tspan(2);
nptp1=npt+1; dt=(tspan(2)-tspan(1))/npt;
c=yi; n=length(yi);

y=zeros(nptp1,n); time=zeros(nptp1,1);

whos n fname t c dt
t=tspan(1); y(1,:)=yi'; time(1)=t;
for i=2:nptp1,
 [c]=rk4(fname,n,t,c,dt);
 y(i,:)=c';
```

```
  t=t+dt;
  time(i)=t;
end
```

Q.   4th Order Runga-Kutta Solver (written by Dr. Hurtado): *rk4.m*

```
function [x]=rk4(fname,n,t,x,dt)
% function [x]=rk4(fname,n,t,x,dt)
% 4th order runge kutta
% fixed time step

%[x,lmda]=rk4(fname,n,t,x,dt,u)
%eval(['[f2,lmda]=' fname '(t2,y2,u);'])

y1=zeros(n,1);  y2=y1;  y3=y1;  y4=y1;

for i=1:n, y1(i)=x(i); end
t1=t;
eval(['[f1]=' fname '(t1,y1);'])

for i=1:n, y2(i)=0.5*dt*f1(i)+y1(i); end
t2=t1+0.5*dt;
eval(['[f2]=' fname '(t2,y2);'])

for i=1:n, y3(i)=0.5*dt*f2(i)+y1(i); end
eval(['[f3]=' fname '(t2,y3);'])

for i=1:n, y4(i)=dt*f3(i)+y1(i); end
t4=t1+dt;
eval(['[f4]=' fname '(t4,y4);'])

g=dt*(f1+f4+2.0*(f2+f3))/6.0;
[nr,nc] = size(g); if nc>nr, g=g'; end
x=x+g;
```

R.   FFT Function: *myfft.m*

```
%my fft will use matlab's fft to find frequency content of time domain
%signal and put into more useful form
%outputs frequencies and amplitudes
function [freq,Amp] = myfft(t,x,I);
% function [f,Amp] = myfft(t);

x = x(:,I.fftstate);
x = x - mean(x);

L = length(x);          %length of signal vector/matrix
Fs = 1/(t(2) - t(1))              %sample frequency
NFFT = 2^nextpow2(L)

freq = Fs/2*linspace(0,1,NFFT/2);        %frequencies
Amp = fft(x,NFFT)/L;          %amplitudes
Amp = 2*abs(Amp(1:NFFT/2,:));

figure(18)
loglog( repmat(I.one,1,length(Amp)), Amp, repmat(I.two,1,...
    length(Amp)), Amp, repmat(I.three,1,length(Amp)), Amp, freq*2*pi,Amp)...
    ,legend('Wn_1','Wn_2','Wn_3')
ylabel('Amplitude'),xlabel('Frequency (rad/s)')
```

S. Aerodynamic File for Linear Controller for Nonlinear Response: *liftdragDIS-CRETE.m*

```
%% Angles
if i == 1
    u = xeq(1);
    w = xeq(2);
    theta = 0;
    theta_d = 0;
    phi = 0;
    phi_d = 0;
else
    u = xeq(1) + xk(1);
    w = xeq(2) + xk(2);
    theta = xk(3);
    theta_d = xk(4);
    phi = xk(5);
    phi_d = xk(6);
end

h = theta - atan2(w,u);
%update physical constants
K0 = 6.8757e-006;        % /ft
h0 = 5000;           % ft
h_t = h + h0;
rho0 = 0.00238;      % lb-s^2/ft^4
rho = rho0*(1-K0*h_t)^(4.25);    %slug/ft^3 air


% Wing velocity, determined using velocity at system CM and included
% induced velocity due to rotation of Wing CM about system CM
xw_d = u - (I.mf/I.m)*I.dc*(theta_d + phi_d)*sin(phi);
yw_d = w - (I.mf/I.m)*(I.d*theta_d + I.dc*(theta_d + phi_d)*cos(phi));

gam_w = theta - atan2(yw_d,xw_d);

% Tail velocity, determined using velocity at system CM and included
% induced velocity due to rotation of Tail CM about system CM
xt_d = u + (I.mw/I.m)*I.dc*(theta_d + phi_d)*sin(phi);
yt_d = w + (I.d + I.f)*theta_d + (I.mw/I.m)*(I.dc*(theta_d + ...
    phi_d)*cos(phi));

gam_t = theta - atan2(yt_d,xt_d);

% Angle of Attack for Wing and Tail
alpha_w = theta + phi - gam_w;

alpha_t = alpha_w*(1 - I.deda) - I.alfr - I.eps0;

% for the system
gam = theta - atan2(w,u);
V = sqrt(u^2 + w^2);

gam_1 = [gam_w
    gam_t
    gam];

for n = 1:length(gam_1)
    if abs(gam_1(n)) <= 1e-4
        gam_1(n) = 0;
    else
        gam_1(n) = gam_1(n);
    end
```

```
end

gam_w = gam_1(1);
gam_t = gam_1(2);
gam = gam_1(3);
%% Lift/Drag

% for the wing
Vw = sqrt(xw_d^2 + yw_d^2);   % Calculates Velocity Scalar from x and y
Q = 0.5*rho*(Vw^2);           % Calculates Dynamic Pressure

% for the tail
Vt = sqrt(xt_d^2 + yt_d^2);
Qt = 0.5*rho*(Vt^2);

eta = ((Qt*I.St)/(Q*I.S));
CLde = I.Cldelta;
CLdf = I.Clalfw*I.tau_f*(I.Sf/I.S);

% Moment of Inertias
Is1 = I.Iw + I.If + (I.mf*I.mw/I.m)*(I.dc^2 + I.d^2 + 2*I.d*I.dc*cos(phi));
Is2 = I.Iw + (I.mf*I.mw/I.m)*(I.dc^2 + I.d*I.dc*cos(phi));

Iw1 = I.Iw + ((I.mf/I.m)^2)*I.mw*(I.dc^2 + I.d*I.dc*cos(phi));
Iw2 = I.Iw + ((I.mf/I.m)^2)*I.mw*(I.dc^2);

% Mass Matrix
M = [I.m 0 I.m*w 0 0 0
     0 I.m (-I.m*u) 0 0 0
     0 0 1 0 0 0
     0 0 0 Is1 0 Is2
     0 0 0 0 1 0
     0 0 0 Iw1 0 Iw2];
```

VITA

Amy Marie Lucas received her Bachelors of Science degree in May of 2007 from Lipscomb University, double majoring in Engineering Mechanics and Spanish. She entered the Aerospace Engineering Masters program at Texas A&M University in September 2007 and began research in Aeroelasticity under the advisement of Dr. Thomas Strganac. She graduated with her Masters degree in Aerospace Engineering in August 2009. She was awarded the Graduate Research Fellowship from NASA and, through the fellowship, collaborated with the Aeroelasticity group at NASA Langley for her research. She plans to pursue a career in the Aerospace Industry focusing on the area of Dynamics and Control.

Mrs. Lucas may be reached at lucasam28@gmail.com or by contacting Dr. Thomas W. Strganac, Department of Aerospace Engineering, Texas A&M University, TAMU 3141, College Station, TX 77843.