Linearized Rigid-Body Static and Dynamic Stability of an Aircraft With a Bio-Inspired Rotating Empennage

Austin J. Kohler
Utah State University

Follow this and additional works at: https://digitalcommons.usu.edu/etd

Part of the Mechanical Engineering Commons

Recommended Citation
https://digitalcommons.usu.edu/etd/8675

This Thesis is brought to you for free and open access by the Graduate Studies at DigitalCommons@USU. It has been accepted for inclusion in All Graduate Theses and Dissertations by an authorized administrator of DigitalCommons@USU. For more information, please contact digitalcommons@usu.edu.
LINEARIZED RIGID-BODY STATIC AND DYNAMIC STABILITY OF AN AIRCRAFT WITH A BIO-INSPIRED ROTATING EMPENNAGE

by

Austin J. Kohler

A thesis submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

in

Mechanical Engineering

Approved:

Douglas F. Hunsaker, Ph.D.
Major Professor

James J. Joo, Ph.D.
Committee Member

Stephen A. Whitmore, Ph.D.
Committee Member

D. Richard Cutler, Ph.D.
Vice Provost of Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah
2022
ABSTRACT

Linearized Rigid-Body Static and Dynamic Stability of an Aircraft with a Bio-Inspired Rotating Empennage

by

Austin J. Kohler, Master of Science
Utah State University, 2022

Major Professor: Douglas F. Hunsaker, Ph.D.
Department: Mechanical and Aerospace Engineering

This work explores the effects of a Bio-Inspired Rotating Empennage (BIRE) on the static and dynamic stability and handling qualities of a fighter aircraft. The BIRE-modified aircraft does not have a vertical tail, and is instead capable of rotating the horizontal tail about the body x-axis for maneuvering. The dynamic characteristics of the BIRE-modified aircraft are compared to a baseline unmodified aircraft, similar to the F16, with a traditional vertical tail in the linear aerodynamic range below stall. Linearized aerodynamic models for each aircraft, based on previous work, are used alongside a set of coupled dynamic equations of motion for asymmetric aircraft, derived in this work, to estimate the dynamic response of each aircraft to disturbances from steady level and banked trim conditions.

The static stability analysis suggests that modifying the baseline with a BIRE decreases the aircraft’s static pitch, roll and yaw stability. The dynamic stability analysis suggests that modifying the baseline aircraft with a BIRE; 1) slightly decreases the aircraft’s short period damping and slightly increases the aircraft’s short period frequency, 2) decreases the aircraft’s phugoid damping and slightly increases the aircraft’s phugoid frequency, 3) slightly increases the aircraft’s roll damping, 4) decreases the aircraft’s spiral damping for steady level flight and increases the aircraft’s spiral damping sensitivity to center of gravity location when banked, and 5) produces a non-traditional dutch roll mode. The handling quality analysis suggests that modifying the baseline aircraft with a BIRE has a negligible
effect on the aircraft’s short period, phugoid, roll, and spiral handling quality levels, but decreases the aircraft’s dutch roll handling quality levels.
PUBLIC ABSTRACT

Linearized Rigid-Body Static and Dynamic Stability of an Aircraft with a Bio-Inspired Rotating Empennage

Austin J. Kohler

The United States Air Force (USAF) will likely seek to remove the vertical tail of next-generation fighter aircraft. This work seeks to characterize the static and dynamic stability and handling qualities of a vertical-tailless aircraft concept that would satisfy the USAF’s goal. This concept aircraft, one modified with a Bio-Inspired Rotating Empennage (BIRE), does not have a vertical tail, and is instead capable of rotating the horizontal tail about the fuselage axis for maneuvering. The dynamic characteristics of the BIRE-modified aircraft are compared to a baseline unmodified aircraft, similar to the F16, with a traditional vertical tail. Linearized aerodynamic models for each aircraft, based on previous work, are used alongside a set of coupled dynamic equations of motion for asymmetric aircraft, derived in this work, to estimate the dynamic response of each aircraft to disturbances from steady level and banked trim conditions.

The static stability analysis suggests that modifying the baseline with a BIRE decreases the aircraft’s static pitch, roll and yaw stability. The dynamic stability analysis suggests that modifying the baseline aircraft with a BIRE; 1) slightly decreases the aircraft’s short period damping and slightly increases the aircraft’s short period frequency, 2) decreases the aircraft’s phugoid damping and slightly increases the aircraft’s phugoid frequency, 3) slightly increases the aircraft’s roll damping, 4) decreases the aircraft’s spiral damping for steady level flight and increases the aircraft’s spiral damping sensitivity to center of gravity location when banked, and 5) produces a non-traditional dutch roll mode. The handling quality analysis suggests that modifying the baseline aircraft with a BIRE decreases only the aircraft’s dutch roll handling quality levels.
Dedicated to Sarah, whose support will forever be invaluable.
ACKNOWLEDGMENTS

I want to foremost thank my mentor and major professor, Dr. Hunsaker, for his guidance and inspiring example. I would also like to thank Christian Bolander and Ben Moulton for their support and contributions to this work.

This work was funded by the Air Force Office of Scientific Research (AFOSR) Lab Task 20RQCOR028 with Dr. Les Lee as the program manager. The content of this thesis been cleared for public release, Case Number: AFRL-2022-5640.

Austin Kohler
CONTENTS

<table>
<thead>
<tr>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT .......................... iii</td>
</tr>
<tr>
<td>PUBLIC ABSTRACT .................... v</td>
</tr>
<tr>
<td>ACKNOWLEDGMENTS ................... vii</td>
</tr>
<tr>
<td>LIST OF TABLES ..................... x</td>
</tr>
<tr>
<td>LIST OF FIGURES ................... xi</td>
</tr>
<tr>
<td>NOMENCLATURE ...................... xii</td>
</tr>
<tr>
<td>1 INTRODUCTION ........................ 1</td>
</tr>
<tr>
<td>1.1 Motivation for Current Work ........ 1</td>
</tr>
<tr>
<td>1.1.1 Importance of Aircraft Stability to Handling Qualities ... 2</td>
</tr>
<tr>
<td>1.1.2 Stability Concerns of the BIRE Concept ... 2</td>
</tr>
<tr>
<td>1.2 Objectives ....................... 3</td>
</tr>
<tr>
<td>2 AIRCRAFT MODEL .................... 4</td>
</tr>
<tr>
<td>2.1 Source of Aircraft Model ........... 4</td>
</tr>
<tr>
<td>2.2 Aircraft Geometric Properties ....... 4</td>
</tr>
<tr>
<td>2.3 Aircraft Mass Properties .......... 5</td>
</tr>
<tr>
<td>2.4 Aerodynamic Model ............... 6</td>
</tr>
<tr>
<td>3 STABILITY THEORY .................. 9</td>
</tr>
<tr>
<td>3.1 Static Stability ................... 9</td>
</tr>
<tr>
<td>3.2 Dynamic Stability ................ 9</td>
</tr>
<tr>
<td>3.2.1 6-DOF Dynamic Equations ........ 10</td>
</tr>
<tr>
<td>3.2.2 Linearized Dynamic Equations .... 11</td>
</tr>
<tr>
<td>3.2.3 Coupled Dimensional Eigensystem . . 32</td>
</tr>
<tr>
<td>3.2.4 Dynamic Modes .................. 33</td>
</tr>
<tr>
<td>3.3 Handling Qualities ................ 35</td>
</tr>
<tr>
<td>3.3.1 Short Period Mode Requirements ... 36</td>
</tr>
<tr>
<td>3.3.2 Phugoid Mode Requirements ........ 37</td>
</tr>
<tr>
<td>3.3.3 Roll Mode Requirements .......... 38</td>
</tr>
<tr>
<td>3.3.4 Spiral Mode Requirements ........ 38</td>
</tr>
<tr>
<td>3.3.5 Dutch-Roll Mode Requirements .... 38</td>
</tr>
<tr>
<td>4 ANALYSIS METHODOLOGY ................ 40</td>
</tr>
<tr>
<td>4.1 Stability Analysis Methodology Outline ... 40</td>
</tr>
<tr>
<td>4.2 Static Stability Analysis Methodology ... 40</td>
</tr>
<tr>
<td>4.3 Dynamic Stability Analysis Methodology ... 41</td>
</tr>
</tbody>
</table>
LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Lifting surface geometric properties that are relevant to the stability analysis presented in other sections. Adapted from table by Hunsaker et al. [1].</td>
</tr>
<tr>
<td>2.2</td>
<td>Control surface deflection limits of the baseline and BIRE aircraft. Used with permission [1].</td>
</tr>
<tr>
<td>2.3</td>
<td>Mass properties of the baseline and BIRE aircraft. Used with permission [1].</td>
</tr>
<tr>
<td>3.1</td>
<td>U.S. Military handling quality levels adapted from Table 10.2.1 in Phillips [2]</td>
</tr>
<tr>
<td>3.2</td>
<td>U.S. Military flight phase categories adapted from Table 10.2.2 in Phillips [2]</td>
</tr>
<tr>
<td>3.3</td>
<td>U.S. Military aircraft classifications adapted from Table 10.2.3 in Phillips [2]</td>
</tr>
<tr>
<td>3.4</td>
<td>Short period CAP requirements for all aircraft classifications. Adapted from Eq. 10.3.8 in Phillips [3].</td>
</tr>
<tr>
<td>3.5</td>
<td>Short period damping requirements for all aircraft classifications. Adapted from Table 10.3.5 in Phillips [3].</td>
</tr>
<tr>
<td>3.6</td>
<td>Phugoid damping requirements for all flight phases and aircraft classifications. Adapted from Table 10.3.2 in Phillips [3].</td>
</tr>
<tr>
<td>3.7</td>
<td>Roll mode time constant requirements. Adapted from Table 10.3.3 in Phillips [3].</td>
</tr>
<tr>
<td>3.8</td>
<td>Spiral mode doubling time requirements. Adapted from Table 10.3.4 in Phillips [3].</td>
</tr>
<tr>
<td>3.9</td>
<td>Dutch-roll mode frequency and damping requirements. Adapted from Table 10.3.6 in Phillips [3].</td>
</tr>
<tr>
<td>4.1</td>
<td>Flight conditions considered for analysis. Used with permission [1].</td>
</tr>
<tr>
<td>A.1</td>
<td>Baseline F-16 aerodynamic model force derivative coefficients.</td>
</tr>
<tr>
<td>A.2</td>
<td>Baseline F-16 aerodynamic model moment derivative coefficients.</td>
</tr>
<tr>
<td>A.3</td>
<td>BIRE force derivative coefficient sinusoidal fit parameters used in Equation 2.9.</td>
</tr>
<tr>
<td>A.4</td>
<td>BIRE moment derivative coefficient sinusoidal fit parameters used in Equation 2.9.</td>
</tr>
</tbody>
</table>
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>The red kite rotates its tail for presumed lateral control during a banking turn. The BIRE fighter concept seeks to emulate the kite’s rotating control system. Used with permission [1].</td>
<td>1</td>
</tr>
<tr>
<td>2.1</td>
<td>The properties of both aircraft are defined relative to this body-fixed coordinate system. The coordinate system origin coincides with the center of gravity. Used with permission [1].</td>
<td>5</td>
</tr>
<tr>
<td>5.1</td>
<td>Pitch stability derivative, $C_{m,\alpha}$, as a function of BIRE rotation angle, $\delta_B$.</td>
<td>49</td>
</tr>
<tr>
<td>5.2</td>
<td>Roll stability derivative, $C_{l,\beta}$, as a function of BIRE rotation angle, $\delta_B$.</td>
<td>50</td>
</tr>
<tr>
<td>5.3</td>
<td>Yaw stability derivative, $C_{n,\beta}$, as a function of BIRE rotation angle, $\delta_B$.</td>
<td>51</td>
</tr>
<tr>
<td>5.4</td>
<td>The short period mode eigenvalues as functions of bank angle and CG location. CG location ranges from +1 ft. forward of the nominal to +0 forward of the nominal location, with the largest marker corresponding to +1 ft. forward and the smallest marker corresponding to +0 ft. forward. The four labeled points indicate at what CG location the aircraft’s short period split into two real eigenvalues.</td>
<td>52</td>
</tr>
<tr>
<td>5.5</td>
<td>The phugoid mode eigenvalues as functions of bank angle and CG location. CG location ranges from +1 ft. forward of the nominal to +0 forward of the nominal location, with the largest marker corresponding to +1 ft. forward and the smallest marker corresponding to +0 ft. forward.</td>
<td>54</td>
</tr>
<tr>
<td>5.6</td>
<td>The roll mode damping rates as functions of bank angle and CG location.</td>
<td>56</td>
</tr>
<tr>
<td>5.7</td>
<td>The spiral mode damping rates as functions of bank angle and CG location.</td>
<td>57</td>
</tr>
<tr>
<td>5.8</td>
<td>The dutch roll mode eigenvalues as functions of bank angle and CG location. CG location ranges from +1 ft. forward of the nominal to +0 forward of the nominal location, with the largest marker corresponding to +1 ft. forward and the smallest marker corresponding to +0 ft. forward.</td>
<td>59</td>
</tr>
<tr>
<td>5.9</td>
<td>Handling quality levels of each aircraft for each dynamic mode and bank angle. The BIRE aircraft is able to match the handling quality levels of the Baseline’s phugoid, roll, and spiral modes.</td>
<td>61</td>
</tr>
</tbody>
</table>
### NOMENCLATURE

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AFRL</td>
<td>Air-Force Research Lab</td>
</tr>
<tr>
<td>AoA</td>
<td>Angle of Attack</td>
</tr>
<tr>
<td>BIRE</td>
<td>Bio-Inspired Rotating Empennage</td>
</tr>
<tr>
<td>CG</td>
<td>Center of Gravity</td>
</tr>
<tr>
<td>DOF</td>
<td>Degree of Freedom</td>
</tr>
<tr>
<td>$b$</td>
<td>Span</td>
</tr>
<tr>
<td>$b_w$</td>
<td>Main wing span</td>
</tr>
<tr>
<td>$C_D$</td>
<td>Drag coefficient</td>
</tr>
<tr>
<td>$C_{D_0}$</td>
<td>Drag coefficient when $\alpha, \beta, \bar{p}, \bar{q}, \bar{r}, \delta_c, \delta_a, \delta_r = 0$</td>
</tr>
<tr>
<td>$C_{D,L}$</td>
<td>Derivative of drag coefficient, $C_D$, with respect to pseudo-lift coefficient, $C_{L_1}$</td>
</tr>
<tr>
<td>$C_{D,L^2}$</td>
<td>Derivative of drag coefficient, $C_D$, with respect to pseudo-lift coefficient squared, $C_{L_1}^2$</td>
</tr>
<tr>
<td>$C_{D,\bar{p}}$</td>
<td>Derivative of drag coefficient, $C_D$, with respect to roll rate, $\bar{p}$</td>
</tr>
<tr>
<td>$C_{D,S\bar{p}}$</td>
<td>Derivative of drag roll derivative, $C_{D,\bar{p}}$, with respect to pseudo-sideslip coefficient, $C_{S_1}$</td>
</tr>
<tr>
<td>$C_{D,\bar{q}}$</td>
<td>Derivative of drag coefficient, $C_D$, with respect to pitch rate, $\bar{q}$</td>
</tr>
<tr>
<td>$C_{D,L\bar{q}}$</td>
<td>Derivative of drag pitch derivative, $C_{D,\bar{q}}$, with respect to pseudo-lift coefficient, $C_{L_1}$</td>
</tr>
<tr>
<td>$C_{D,L^2\bar{q}}$</td>
<td>Derivative of drag pitch derivative, $C_{D,\bar{q}}$, with respect to pseudo-lift coefficient squared, $C_{L_1}^2$</td>
</tr>
<tr>
<td>$C_{D,\bar{r}}$</td>
<td>Derivative of drag coefficient, $C_D$, with respect to yaw rate, $\bar{r}$</td>
</tr>
<tr>
<td>$C_{D,S\bar{r}}$</td>
<td>Derivative of drag yaw derivative, $C_{D,\bar{r}}$, with respect to pseudo-sideslip coefficient, $C_{S_1}$</td>
</tr>
<tr>
<td>$C_{D,S}$</td>
<td>Derivative of drag coefficient, $C_D$, with respect to pseudo-sideslip coefficient, $C_{S_1}$</td>
</tr>
<tr>
<td>$C_{D,S^2}$</td>
<td>Derivative of drag coefficient, $C_D$, with respect to pseudo-sideslip coefficient squared, $C_{S_1}^2$</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>$C_{D,\beta}$</td>
<td>Derivative of drag coefficient, $C_D$, with respect to sideslip angle, $\beta$</td>
</tr>
<tr>
<td>$C_{D,\delta_a}$</td>
<td>Derivative of drag coefficient, $C_D$, with respect to aileron deflection, $\delta_a$</td>
</tr>
<tr>
<td>$C_{D,S\delta_a}$</td>
<td>Derivative of drag aileron derivative, $C_{D,\delta_a}$, with respect to pseudo-sideslip coefficient, $C_{S_1}$</td>
</tr>
<tr>
<td>$C_{D,\delta_e}$</td>
<td>Derivative of drag coefficient, $C_D$, with respect to elevator deflection, $\delta_e$</td>
</tr>
<tr>
<td>$C_{D,L\delta_e}$</td>
<td>Derivative of drag elevator derivative, $C_{D,\delta_e}$, with respect to pseudo-lift coefficient, $C_{L_1}$</td>
</tr>
<tr>
<td>$C_{D,\delta_e^2}$</td>
<td>Derivative of drag coefficient, $C_D$, with respect to elevator deflection squared, $\delta_e^2$</td>
</tr>
<tr>
<td>$C_{D,\delta_r}$</td>
<td>Derivative of drag coefficient, $C_D$, with respect to rudder deflection, $\delta_r$</td>
</tr>
<tr>
<td>$C_{D,S\delta_r}$</td>
<td>Derivative of drag rudder derivative, $C_{D,\delta_r}$, with respect to pseudo-sideslip coefficient, $C_{S_1}$</td>
</tr>
<tr>
<td>$C_L$</td>
<td>Lift coefficient</td>
</tr>
<tr>
<td>$C_{L,w}$</td>
<td>Lift coefficient of the main wing</td>
</tr>
<tr>
<td>$C_{L,h}$</td>
<td>Lift coefficient of the horizontal tail</td>
</tr>
<tr>
<td>$C_{L,0}$</td>
<td>Lift coefficient when $\alpha, \beta, \bar{p}, \bar{q}, \bar{r}, \delta_e, \delta_a, \delta_r = 0$</td>
</tr>
<tr>
<td>$C_{L,1}$</td>
<td>Pseudo lift coefficient</td>
</tr>
<tr>
<td>$C_{L,\bar{p}}$</td>
<td>Derivative of lift coefficient, $C_L$, with respect to roll rate, $\bar{p}$</td>
</tr>
<tr>
<td>$C_{L,\bar{q}}$</td>
<td>Derivative of lift coefficient, $C_L$, with respect to pitch rate, $\bar{q}$</td>
</tr>
<tr>
<td>$C_{L,\bar{r}}$</td>
<td>Derivative of lift coefficient, $C_L$, with respect to yaw rate, $\bar{r}$</td>
</tr>
<tr>
<td>$C_{L,\alpha}$</td>
<td>Derivative of lift coefficient, $C_L$, with respect to angle of attack, $\alpha$</td>
</tr>
<tr>
<td>$C_{L,\beta}$</td>
<td>Derivative of lift coefficient, $C_L$, with respect to sideslip angle, $\beta$</td>
</tr>
<tr>
<td>$C_{L,\delta_a}$</td>
<td>Derivative of lift coefficient, $C_L$, with respect to aileron deflection, $\delta_a$</td>
</tr>
<tr>
<td>$C_{L,\delta_e}$</td>
<td>Derivative of lift coefficient, $C_L$, with respect to elevator deflection, $\delta_e$</td>
</tr>
<tr>
<td>$C_\ell$</td>
<td>Rolling moment coefficient</td>
</tr>
<tr>
<td>$C_{\ell,0}$</td>
<td>Rolling moment coefficient when $\alpha, \beta, \bar{p}, \bar{q}, \bar{r}, \delta_e, \delta_a, \delta_r = 0$</td>
</tr>
</tbody>
</table>
\( C_{\ell, \bar{p}} \) Derivative of rolling moment coefficient, \( C_{\ell} \), with respect to roll rate, \( \bar{p} \)  
\( C_{\ell, \bar{q}} \) Derivative of rolling moment coefficient, \( C_{\ell} \), with respect to pitch rate, \( \bar{q} \)  
\( C_{\ell, \bar{r}} \) Derivative of rolling moment coefficient, \( C_{\ell} \), with respect to yaw rate, \( \bar{r} \)  
\( C_{\ell, L_{L}} \) Derivative of rolling moment rudder derivative, \( C_{\ell, L} \), with respect to pseudo-lift coefficient, \( C_{L_{L}} \)  
\( C_{\ell, \alpha} \) Derivative of rolling moment coefficient, \( C_{\ell} \), with respect to angle of attack, \( \alpha \)  
\( C_{\ell, \beta} \) Derivative of rolling moment coefficient, \( C_{\ell} \), with respect to sideslip angle, \( \beta \)  
\( C_{\ell, \delta_{a}} \) Derivative of rolling moment coefficient, \( C_{\ell} \), with respect to aileron deflection, \( \delta_{a} \)  
\( C_{\ell, \delta_{e}} \) Derivative of rolling moment coefficient, \( C_{\ell} \), with respect to elevator deflection, \( \delta_{e} \)  
\( C_{\ell, \delta_{r}} \) Derivative of rolling moment coefficient, \( C_{\ell} \), with respect to rudder deflection, \( \delta_{r} \)  
\( C_{m} \) Pitching moment coefficient  
\( C_{m_{0}} \) Pitching moment coefficient when \( \alpha, \beta, \bar{p}, \bar{q}, \bar{r}, \delta_{e}, \delta_{a}, \delta_{r} = 0 \)  
\( C_{m, \bar{p}} \) Derivative of pitching moment coefficient, \( C_{m} \), with respect to roll rate, \( \bar{p} \)  
\( C_{m, \bar{q}} \) Derivative of pitching moment coefficient, \( C_{m} \), with respect to pitch rate, \( \bar{q} \)  
\( C_{m, \bar{r}} \) Derivative of pitching moment coefficient, \( C_{m} \), with respect to yaw rate, \( \bar{r} \)  
\( C_{m, \alpha} \) Derivative of pitching moment coefficient, \( C_{m} \), with respect to angle of attack, \( \alpha \)  
\( C_{m, \beta} \) Derivative of pitching moment coefficient, \( C_{m} \), with respect to sideslip angle, \( \beta \)  
\( C_{m, \delta_{a}} \) Derivative of pitching moment coefficient, \( C_{m} \), with respect to aileron deflection, \( \delta_{a} \)  
\( C_{m, \delta_{e}} \) Derivative of pitching moment coefficient, \( C_{m} \), with respect to elevator deflection, \( \delta_{e} \)  
\( C_{n} \) Yawing moment coefficient  
\( C_{n_{0}} \) Yawing moment coefficient when \( \alpha, \beta, \bar{p}, \bar{q}, \bar{r}, \delta_{e}, \delta_{a}, \delta_{r} = 0 \)
$C_{n,\dot{p}}$ Derivative of yawing moment coefficient, $C_n$, with respect to roll rate, $\dot{p}$

$C_{n,L\dot{p}}$ Derivative of yawing moment roll derivative, $C_{n,\dot{p}}$, with respect to pseudo-lift coefficient, $C_{L_1}$

$C_{n,\dot{q}}$ Derivative of yawing moment coefficient, $C_n$, with respect to pitch rate, $\dot{q}$

$C_{n,\dot{r}}$ Derivative of yawing moment coefficient, $C_n$, with respect to yaw rate, $\dot{r}$

$C_{n,\alpha}$ Derivative of yawing moment coefficient, $C_n$, with respect to angle of attack, $\alpha$

$C_{n,\beta}$ Derivative of yawing moment coefficient, $C_n$, with respect to sideslip angle, $\beta$

$C_{n,\delta_e}$ Derivative of yawing moment coefficient, $C_n$, with respect to aileron deflection, $\delta_a$

$C_{n,\delta_a}$ Derivative of yawing moment coefficient, $C_n$, with respect to elevator deflection, $\delta_e$

$C_{n,L\delta_a}$ Derivative of yawing moment aileron derivative, $C_{n,\delta_a}$, with respect to pseudo-lift coefficient, $C_{L_1}$

$C_{n,\delta_r}$ Derivative of yawing moment coefficient, $C_n$, with respect to rudder deflection, $\delta_r$

$C_S$ Sideslip coefficient

$C_{S_0}$ Sideslip coefficient when $\alpha, \beta, \dot{p}, \dot{q}, \dot{r}, \delta_e, \delta_a, \delta_r = 0$

$C_{S_1}$ Pseudo sideslip coefficient

$C_{S,\dot{p}}$ Derivative of sideslip coefficient, $C_S$, with respect to roll rate, $\dot{p}$

$C_{S,L\dot{p}}$ Derivative of sideslip rolling rate derivative, $C_{S,\dot{p}}$, with respect to pseudo-lift coefficient, $C_{L_1}$

$C_{S,\dot{q}}$ Derivative of sideslip coefficient, $C_S$, with respect to pitch rate, $\dot{q}$

$C_{S,\dot{r}}$ Derivative of sideslip coefficient, $C_S$, with respect to yaw rate, $\dot{r}$

$C_{S,\alpha}$ Derivative of sideslip coefficient, $C_S$, with respect to angle of attack, $\alpha$

$C_{S,\beta}$ Derivative of sideslip coefficient, $C_S$, with respect to sideslip angle, $\beta$
Derivative of sideslip coefficient, $C_S$, with respect to aileron deflection, $\delta_a$

Derivative of sideslip coefficient, $C_S$, with respect to elevator deflection, $\delta_e$

Derivative of sideslip coefficient, $C_S$, with respect to rudder deflection, $\delta_r$

Cosine of bank angle, $\phi$

Cosine of azimuth angle, $\psi$

Cosine of elevation angle, $\theta$

BIRE coefficient; function of BIRE rotation angle, $\delta_B$

Mean chord

Main wing mean aerodynamic chord

Aerodynamic forces in the $x_b, y_b, z_b$ directions, respectively

Derivatives of the aerodynamic forces with respect to roll rate, $p$, in the $x_b, y_b, z_b$ directions, respectively

Derivatives of the aerodynamic forces with respect to pitch rate, $q$, in the $x_b, y_b, z_b$ directions, respectively

Derivatives of the aerodynamic forces with respect to yaw rate, $r$, in the $x_b, y_b, z_b$ directions, respectively

Derivatives of the aerodynamic forces with respect to forward velocity, $u$, in the $x_b, y_b, z_b$ directions, respectively

Derivatives of the aerodynamic forces with respect to forward acceleration, $\dot{u}$, in the $x_b, y_b, z_b$ directions, respectively

Derivatives of the aerodynamic forces with respect to side velocity, $v$, in the $x_b, y_b, z_b$ directions, respectively

Derivatives of the aerodynamic forces with respect to side acceleration, $\dot{v}$, in the $x_b, y_b, z_b$ directions, respectively

Derivatives of the aerodynamic forces with respect to vertical velocity, $w$, in the $x_b, y_b, z_b$ directions, respectively

Derivatives of the aerodynamic forces with respect to vertical acceleration, $\dot{w}$, in the $x_b, y_b, z_b$ directions, respectively

Derivatives of the aerodynamic forces with respect to aileron deflection, $\delta_a$, in the $x_b, y_b, z_b$ directions, respectively

Derivatives of the aerodynamic forces with respect to elevator deflection, $\delta_e$, in the $x_b, y_b, z_b$ directions, respectively
$F_{x_b,\delta_r}, F_{y_b,\delta_r}, F_{z_b,\delta_r}$ Derivatives of the aerodynamic forces with respect to rudder deflection, $\delta_r$, in the $x_b, y_b, z_b$ directions, respectively

$g$ Acceleration due to gravity

$h_x, h_y, h_z$ Aircraft engine momentum

$I_{xx}, I_{yy}, I_{zz}$ Aircraft moments of inertia

$I_{xy}, I_{xz}, I_{yz}$ Aircraft products of inertia

$L$ Dimensional lift force

$L_{\alpha}$ Derivative of lift force, $L$, with respect to angle of attack, $\alpha$

$l_{act}$ Distance between the main wing tip and horizontal tail tip in the $x_b$ direction

$M_{x_b, M_{y_b}, M_{z_b}}$ Aerodynamic moments about the $x_b, y_b, z_b$ axes, respectively

$M_{x_b,p, M_{y_b,p}, M_{z_b,p}}$ Derivatives of the aerodynamic moments with respect to roll rate, $p$, about the $x_b, y_b, z_b$ axes, respectively

$M_{x_b,q,M_{y_b,q}, M_{z_b,q}}$ Derivatives of the aerodynamic moments with respect to pitch rate, $q$, about the $x_b, y_b, z_b$ axes, respectively

$M_{x_b,r, M_{y_b,r}, M_{z_b,r}}$ Derivatives of the aerodynamic moments with respect to yaw rate, $r$, about the $x_b, y_b, z_b$ axes, respectively

$M_{x_b,u, M_{y_b,u}, M_{z_b,u}}$ Derivatives of the aerodynamic moments with respect to forward velocity, $u$, about the $x_b, y_b, z_b$ axes, respectively

$M_{x_b,\dot{u}, M_{y_b,\dot{u}}, M_{z_b,\dot{u}}}$ Derivatives of the aerodynamic moments with respect to forward acceleration, $\dot{u}$, about the $x_b, y_b, z_b$ axes, respectively

$M_{x_b,v, M_{y_b,v}, M_{z_b,v}}$ Derivatives of the aerodynamic moments with respect to side velocity, $v$, about the $x_b, y_b, z_b$ axes, respectively

$M_{x_b,\dot{v}, M_{y_b,\dot{v}}, M_{z_b,\dot{v}}}$ Derivatives of the aerodynamic moments with respect to side acceleration, $\dot{v}$, about the $x_b, y_b, z_b$ axes, respectively

$M_{x_b,w, M_{y_b,w}, M_{z_b,w}}$ Derivatives of the aerodynamic moments with respect to vertical velocity, $w$, about the $x_b, y_b, z_b$ axes, respectively

$M_{x_b,\dot{w}, M_{y_b,\dot{w}}, M_{z_b,\dot{w}}}$ Derivatives of the aerodynamic moments with respect to vertical acceleration, $\dot{w}$, about the $x_b, y_b, z_b$ axes, respectively

$M_{x_b,\delta_a, M_{y_b,\delta_a}, M_{z_b,\delta_a}}$ Derivatives of the aerodynamic moments with respect to aileron deflection, $\delta_a$, about the $x_b, y_b, z_b$ axes, respectively

$M_{x_b,\delta_e, M_{y_b,\delta_e}, M_{z_b,\delta_e}}$ Derivatives of the aerodynamic moments with respect to elevator deflection, $\delta_e$, about the $x_b, y_b, z_b$ axes, respectively

$M_{x_b,\delta_r, M_{y_b,\delta_r}, M_{z_b,\delta_r}}$ Derivatives of the aerodynamic moments with respect to rudder deflection, $\delta_r$, about the $x_b, y_b, z_b$ axes, respectively
Load factor

Roll rate

Roll acceleration

Pitch rate

Pitch acceleration

Yaw rate

Yaw acceleration

Planform area

Planform area of the main wing

Planform area of the horizontal tail

Sine of bank angle, \( \phi \)

Sine of azimuth angle, \( \psi \)

Sine of elevation angle, \( \theta \)

Tangent of bank angle, \( \phi \)

Tangent of azimuth angle, \( \psi \)

Tangent of elevation angle, \( \theta \)

Time

Translational velocities in the \( x_b, y_b, z_b \) directions, respectively

Translational accelerations in the \( x_b, y_b, z_b \) directions, respectively

Freestream velocity

Wind velocities in the \( x_f, y_f, z_f \) directions, respectively

Aircraft weight

Body-fixed \( x, y, z \) cartesian coordinates

\( x_b \) position of the horizontal tail

\( x_b \) position of the main wing’s aft-most tip

Translational velocities in the \( x_b, y_b, z_b \) directions, respectively

Earth-fixed \( x, y, z \) cartesian coordinates

Translational velocities in the \( x_f, y_f, z_f \) directions, respectively

Wind axes \( x, y, z \) cartesian coordinates
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\chi$</td>
<td>Eigenvector</td>
</tr>
<tr>
<td>$\chi_p, \chi_q, \chi_r$</td>
<td>Eigenvector components corresponding to changes in rotation rates, $p, q, r$, respectively</td>
</tr>
<tr>
<td>$\chi_u, \chi_v, \chi_w$</td>
<td>Eigenvector components corresponding to changes in translational velocities, $u, v, w$, respectively</td>
</tr>
<tr>
<td>$\chi_{xf}, \chi_{yf}, \chi_{zf}$</td>
<td>Eigenvector components corresponding to changes in positions, $x_f, y_f, z_f$, respectively</td>
</tr>
<tr>
<td>$\chi_\phi, \chi_\theta, \chi_\psi$</td>
<td>Eigenvector components corresponding to changes in orientation angles, $\phi, \theta, \psi$, respectively</td>
</tr>
<tr>
<td>$\Delta$</td>
<td>Prefix denoting change in the variable that follows</td>
</tr>
<tr>
<td>$o$</td>
<td>Subscript denoting variable’s value at equilibrium state</td>
</tr>
<tr>
<td>$\delta_a$</td>
<td>Aileron deflection angle</td>
</tr>
<tr>
<td>$\delta_B$</td>
<td>BIRE rotation angle</td>
</tr>
<tr>
<td>$\delta_d$</td>
<td>Differential stabilator deflection angle</td>
</tr>
<tr>
<td>$\delta_e$</td>
<td>Symmetric stabilator deflection angle</td>
</tr>
<tr>
<td>$\delta_e^B$</td>
<td>Symmetric BIRE stabilator deflection angle</td>
</tr>
<tr>
<td>$\delta_r$</td>
<td>Rudder deflection angle</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>Eigenvalue</td>
</tr>
<tr>
<td>$\lambda_1, \lambda_2$</td>
<td>First and second complex conjugate eigenvalues, respectively</td>
</tr>
<tr>
<td>$\omega_d$</td>
<td>Dimensional damped frequency</td>
</tr>
<tr>
<td>$\omega_n$</td>
<td>Dimensional undamped natural frequency</td>
</tr>
<tr>
<td>$\phi$</td>
<td>Bank angle</td>
</tr>
<tr>
<td>$\dot{\phi}$</td>
<td>Bank rate</td>
</tr>
<tr>
<td>$\psi$</td>
<td>Azimuth angle</td>
</tr>
<tr>
<td>$\dot{\psi}$</td>
<td>Azimuth rate</td>
</tr>
<tr>
<td>$\sigma$</td>
<td>Dimensional damping rate</td>
</tr>
<tr>
<td>$\theta$</td>
<td>Elevation angle</td>
</tr>
<tr>
<td>$\dot{\theta}$</td>
<td>Elevation rate</td>
</tr>
<tr>
<td>$\zeta$</td>
<td>Dimensional damping ratio</td>
</tr>
</tbody>
</table>
1.1 Motivation for Current Work

As indicated by tailless concepts released by Northrop Grumman, Lockheed-Martin, and Boeing [4], next generation tactical aircraft will likely seek to remove the vertical stabilizer to reduce drag [5] and increase performance [6]. In nature, birds are able to maneuver with tails lacking a vertical stabilizer. Pennycuick [7] proposed that some birds actively control their flight by adjusting the rotation of their tails. Other authors also believe birds are able to control their yaw [8], [9] and roll [10], [11] using these tail rotations. Figure 1.1 shows a red kite rotating its tail during flight alongside a fighter aircraft fitted with a Bio-Inspired Rotating Empennage (BIRE) that seeks to emulate the bird’s rotating tail control system. The static trim and mechanical actuation of the BIRE have already been studied and are detailed in companion works by Hunsaker et al. [1] and Myszka [12], respectively. This work seeks to expand upon this companion work by quantifying the stability characteristics of the BIRE.

Fig. 1.1: The red kite rotates its tail for presumed lateral control during a banking turn. The BIRE fighter concept seeks to emulate the kite’s rotating control system. Used with permission [1].
1.1.1 Importance of Aircraft Stability to Handling Qualities

The pilot’s ability to maneuver an aircraft during flight is closely related to its stability characteristics. Rigid-body aircraft stability has been well understood for many years and is presented in largely the same way by various authors [14], [15], [16]. An aircraft’s stability is typically separated into two types: static and dynamic. An aircraft is statically stable if it returns to the original equilibrium state when subject to a small change in orientation [17]. An aircraft is dynamically stable if its dynamic modes dampen out over time [18]. Most aircraft designs aim to be statically and dynamically stable. However, some aircraft are purposefully designed to be unstable in certain axes for maneuverability and use active control systems to make the aircraft feel stable to the pilot. This is the case for many fighter aircraft.

Although handling qualities are largely based on pilot opinion, aviation agencies have developed criteria that aim to objectively connect the pilot’s perceived handling qualities to stability parameters [2]. The Cooper-Harper rating scale [19] is a binary decision flowchart commonly used to quantify pilot opinions of an aircraft’s handling qualities. However, this system has a 10-point scale, which is usually excessive, so this work will follow Phillips’s [2] suggestion and use rating criteria presented in U.S. military specifications [20], [21] instead. These military specifications condense the Cooper-Harper rating scale to four handling levels, each directly linked to the time-dependant parameters of a specific dynamic oscillatory mode. Level 1 handling qualities are generally required under normal flight conditions [2]. For actively controlled fighter aircraft, dynamic modes that do not naturally attain level 1 handling qualities require an active control system to attain equivalent level 1 handling qualities for the pilot.

1.1.2 Stability Concerns of the BIRE Concept

In traditional aircraft, the vertical stabilizer and rudder provide lateral stability [22] and contribute to lateral control [23]. One of the major concerns with implementing the BIRE is achieving acceptable stability and handling qualities without a vertical tail. Other designs, similar in principle to the BIRE, have suggested it is possible to achieve acceptable
stability and handling qualities by rotating the control surfaces. Roetman et al. [24] analyzed a variable-dihedral horizontal tail and found it was able to achieve level 1 handling qualities under various flight conditions. Bras et al. [25] explored 2-, 3-, and 4-degree-of-freedom (DOF) rotating empennage control systems and found that the 2-DOF design, consisting of a rotary tail with fixed dihedral and linked elevators, provided acceptable static and dynamic stability. The BIRE concept expands upon Bras et al.’s [25] 2-DOF design by adding an extra DOF via independently actuated elevons.

1.2 Objectives

This work aims to quantify the stability and handling qualities of a BIRE-modified aircraft in comparison to an unmodified baseline aircraft. The objectives of this work are as follows:

1. Identify how modifying a fighter aircraft with the BIRE affects the static stability, dynamic stability, and handling qualities of that aircraft

2. Quantify these effects for an example fighter aircraft

The following sections outline how this work accomplishes these objectives. Chapters 2 – 3 lay forth background information necessary to accomplish the objectives. Chapter 4 discusses the methodology this work uses to accomplish the objectives. Chapter 5 details the results of accomplishing the objectives.
2.1 Source of Aircraft Model

This work will utilize the same aircraft models developed in the companion work by Hunsaker et al. [1]. The companion work chose to use a single-engine tactical fighter similar to the F-16 Fighting Falcon as the baseline aircraft. Hunsaker et al. [1] detail complete aircraft properties and full aerodynamic models for the baseline and BIRE aircraft. They also explain how this model was created, what publicly-available F-16 data it is based on, and how the model was validated. This section includes the properties and aerodynamic model parameters expected to be relevant to this work.

2.2 Aircraft Geometric Properties

With the exception of the empennage, the baseline and BIRE aircraft are identical. The mass and aerodynamic properties of each aircraft are measured about the same body-fixed coordinate system, which is shown in Figure 2.1. The body-fixed coordinate system’s origin is located at the aircraft’s center of gravity. The CG is located at 35% of the mean aerodynamic chord of the main wing, $\bar{c}_w$, measured from the leading edge of the wing root and projected onto the $x_b - z_b$ plane such that the $y_b$ coordinate of the CG is zero.

Table 2.1 details the geometric lifting surface properties relevant to the stability analysis presented in later sections. Other geometric properties, such as specific dimensions, sweep angles, and airfoil shapes are outlined in Hunsaker et al. [1].

Table 2.2 describes the maximum deflections of each control surface. Note that the BIRE has 3 DOF: 1) Symmetric BIRE, $\delta_c^B$, in which both the stabilators rotate symmetrically, 2) Differential tail, $\delta_d$, in which the stabilators rotate antisymmetrically, and 3) BIRE rotation, $\delta_B$, in which the whole BIRE assembly rotates about the body x-axis.
Fig. 2.1: The properties of both aircraft are defined relative to this body-fixed coordinate system. The coordinate system origin coincides with the center of gravity. Used with permission [1].

Table 2.1: Lifting surface geometric properties that are relevant to the stability analysis presented in other sections. Adapted from table by Hunsaker et al. [1].

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Main Wing</th>
<th>Horizontal Tail</th>
<th>Vertical Tail</th>
<th>BIRE Tail</th>
</tr>
</thead>
<tbody>
<tr>
<td>Planform Area, $S$, [ft$^2$]</td>
<td>300</td>
<td>63.7</td>
<td>54.7</td>
<td>63.7</td>
</tr>
<tr>
<td>Span, $b$, [ft]</td>
<td>30</td>
<td>9.39</td>
<td>8.41</td>
<td>9.39</td>
</tr>
<tr>
<td>Mean Chord, $c$, [ft]</td>
<td>11.32</td>
<td>9.6</td>
<td>8.18</td>
<td>9.6</td>
</tr>
</tbody>
</table>

Table 2.2: Control surface deflection limits of the baseline and BIRE aircraft. Used with permission [1].

<table>
<thead>
<tr>
<th>Control Surface</th>
<th>Symbol</th>
<th>Deflection Limits</th>
<th>Aircraft</th>
<th>Lifting Surface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Symmetric Stabilator</td>
<td>$\delta_e$</td>
<td>$\pm 25^\circ$</td>
<td>Baseline</td>
<td>Horizontal Tail</td>
</tr>
<tr>
<td>Symmetric BIRE</td>
<td>$\delta^B_e$</td>
<td>$\pm 25^\circ$</td>
<td>BIRE</td>
<td>BIRE Tail</td>
</tr>
<tr>
<td>Aileron</td>
<td>$\delta_a$</td>
<td>$\pm 21.5^\circ$</td>
<td>Both</td>
<td>Main Wing</td>
</tr>
<tr>
<td>Differential Tail</td>
<td>$\delta_d$</td>
<td>$\pm 5.375^\circ$</td>
<td>Both</td>
<td>Horizontal &amp; BIRE Tail</td>
</tr>
<tr>
<td>Rudder</td>
<td>$\delta_r$</td>
<td>$\pm 30^\circ$</td>
<td>Baseline</td>
<td>Vertical Tail</td>
</tr>
<tr>
<td>BIRE Rotation</td>
<td>$\delta_B$</td>
<td>$\pm 180^\circ$</td>
<td>BIRE</td>
<td>BIRE Tail</td>
</tr>
</tbody>
</table>

2.3 Aircraft Mass Properties

This study uses the same weight, inertia, and engine momentum estimations as the companion work [1]. These estimates are based on data from Stevens and Lewis [26].
Table 2.3 shows how each aircraft’s physical properties differ due to the differences in empennage configuration. How these empennage differences contribute to each aircraft’s mass properties are discussed in the companion work [1].

Table 2.3: Mass properties of the baseline and BIRE aircraft. Used with permission [1].

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Baseline Aircraft</th>
<th>BIRE Design</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weight, ( W ), [lbs.]</td>
<td>20,500</td>
<td>21,000</td>
</tr>
<tr>
<td>Inertia, ( I_{xx} ), [slug-ft^2]</td>
<td>9,496</td>
<td>9,280</td>
</tr>
<tr>
<td>Inertia, ( I_{yy} ), [slug-ft^2]</td>
<td>55,814</td>
<td>58,288 - 161 \cos(2\delta_B)</td>
</tr>
<tr>
<td>Inertia, ( I_{zz} ), [slug-ft^2]</td>
<td>63,100</td>
<td>65,606 + 161 \cos(2\delta_B)</td>
</tr>
<tr>
<td>Inertia, ( I_{xy} ), [slug-ft^2]</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Inertia, ( I_{xz} ), [slug-ft^2]</td>
<td>982</td>
<td>-5</td>
</tr>
<tr>
<td>Inertia, ( I_{yz} ), [slug-ft^2]</td>
<td>0</td>
<td>-161</td>
</tr>
<tr>
<td>Engine Momentum, ( h_x ), [slug-ft^2/\text{s}]</td>
<td>160</td>
<td>160</td>
</tr>
<tr>
<td>Engine Momentum, ( h_y ), [slug-ft^2/\text{s}]</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Engine Momentum, ( h_z ), [slug-ft^2/\text{s}]</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

2.4 Aerodynamic Model

The aerodynamic model for the baseline and BIRE-modified aircraft below stall were studied extensively by Hunsaker et al. [1]. The same aerodynamic model is used for this work.

For traditional aircraft below stall, the nondimensional lift, \( C_L \), sideslip, \( C_S \), drag, \( C_D \), rolling moment, \( C_l \), pitching moment, \( C_m \), and yawing moment, \( C_n \), coefficients can be approximated as

\[
C_L = C_{L_0} + C_{L_\alpha} \alpha + C_{L_\delta e} \delta_e + C_{L_\delta a} \delta_a + C_{L_\delta r} \delta_r
\]

\[
C_S = C_{S_\beta} \beta + (C_{S_\delta a} C_{L_1} + C_{S_\delta l}) \delta_l + C_{S_\delta r} \delta_r + C_{S_\delta a} \delta_a + C_{S_\delta e} \delta_e
\]

\[
C_D = C_{D_0} + C_{D_\alpha} C_{L_1} + C_{D_\delta l} C_{L_1}^2 + C_{D_\delta \sigma} C_{S_1}^2

+ C_{D_\delta a} C_{S_1} \delta_a + (C_{D_\delta a} C_{L_1} + C_{D_\delta l}) \delta_l + C_{D_\delta r} \delta_r
\]

\[
+ C_{D_\delta a} C_{S_1} \delta_a + (C_{D_\delta l} C_{L_1} + C_{D_\delta e}) \delta_e + C_{D_\delta l} C_{S_1} \delta_l + C_{D_\delta r} C_{S_1} \delta_r
\]
\[
C_\ell = C_{\ell,\beta} + C_{\ell,\beta} + (C_{\ell,Lr}C_{L_1} + C_{\ell,p})r + C_{\ell,\delta_a} \delta_a + C_{\ell,\delta_r} \delta_r \quad (2.4)
\]

\[
C_m = C_{m_0} + C_{m,\alpha} \alpha + C_{m,\eta} \eta + C_{m,\delta_e} \delta_e
\]

\[
C_n = C_{n,\beta} + (C_{n,Lp}C_{L_1} + C_{n,p})\bar{p} + C_{n,\tau} \tau + (C_{n,Ld_a}C_{L_1} + C_{n,\delta_a}) \delta_a + C_{n,\delta_r} \delta_r \quad (2.6)
\]

where \(C_{L_0}\) and \(C_{m_0}\) denote the coefficient value at zero angle of attack, \(\alpha\), zero sideslip angle, \(\beta\), zero roll, pitch, yaw rates, \(p, \eta, \tau\), and zero elevator, aileron, rudder deflections, \(\delta_e, \delta_a, \delta_r\). \(C_{D_0}\) denotes the drag coefficient at zero lift, \(C_L\). The coefficients \(C_{a,b}, C_{a,b}^2\), and \(C_{a,b}\) denote the derivative \(C_{a,b}\) as a function of pseudo-lift, \(C_{L_1}\), pseudo-lift squared, \(C_{L_1}^2\) and pseudo-sideslip, \(C_{S_1}\), coefficients respectively. The pseudo-lift and pseudo-sideslip coefficients provide brevity to the above equations and are defined

\[
C_{L_1} \equiv C_{L_0} + C_{L,\alpha} \alpha
\]

\[
C_{S_1} \equiv C_{S_0} + C_{S,\beta} \beta
\]

This symmetric aerodynamic model is used to describe the aerodynamic forces on the baseline aircraft. Values for these coefficients are included in Appendix A.

For the BIRE aircraft, each coefficient derivative is a function of bire rotation, \(\delta_B\). Each BIRE coefficient derivative can be modeled as a sinusoidal function of the form

\[
\hat{C} = A \sin (\omega \delta_B + \phi) + z
\]

where \(A\) is amplitude, \(\omega\) is frequency, \(\phi\) is phase, and \(z\) is offset. The BIRE coefficient modelling process is detailed in the companion work [1].

After inputting a BIRE angle to each coefficient derivative fit, the resulting coefficient derivatives can be used to find the total lift, sideslip, drag, rolling moment, pitching moment, and yawing moment coefficients

\[
\hat{C}_L = \hat{C}_{L_1} + \hat{C}_{L,\beta} + \hat{C}_{L,\beta} + \hat{C}_{L,\beta} + \hat{C}_{L,\beta} + \hat{C}_{L,\beta} + \hat{C}_{L,\beta} \delta_a + \hat{C}_{L,\delta_e} \delta_e
\]

\[ \text{(2.10)} \]
\[
\hat{C}_S = \hat{C}_{S_0} + \hat{C}_{S,\alpha} + \hat{C}_{S,\beta} + \left(\hat{C}_{S,Lp}\hat{C}_{L_1} + \hat{C}_{S,p}\right)\hat{p} + \hat{C}_{S,\bar{q}}\hat{q} + \hat{C}_{S,\tau}\hat{\tau} + \hat{C}_{S,\delta}\delta_a + \hat{C}_{S,\delta}\delta_e \tag{2.11}
\]

\[
\hat{C}_D = \hat{C}_{D_0} + \hat{C}_{D,L}\hat{C}_{L_1} + \hat{C}_{D,L^2}\hat{C}_{L_1} + \hat{C}_{D,S}\hat{C}_{S_1} + \hat{C}_{D,S^2}\hat{C}_{S_1}^2
+ \left(\hat{C}_{D,S}\hat{C}_{S_1} + \hat{C}_{D,p}\right)\hat{p} + \left(\hat{C}_{D,L^2}\hat{C}_{L_1} + \hat{C}_{D,L}\hat{C}_{L_1} + \hat{C}_{D,\bar{q}}\right)\hat{q} + \left(\hat{C}_{D,\tau}\hat{C}_{S_1} + \hat{C}_{D,\tau}\right)\hat{\tau}
+ \left(\hat{C}_{D,S}\delta_a\hat{C}_{S_1} + \hat{C}_{D,\delta_a}\right)\delta_a + \left(\hat{C}_{D,L}\delta_a\hat{C}_{L_1} + \hat{C}_{D,\delta_a}\right)\delta_e + C_{D,\delta\bar{e}}\delta_e^2 \tag{2.12}
\]

\[
\hat{C}_\ell = \hat{C}_{\ell_0} + \hat{C}_{\ell,\alpha} + \hat{C}_{\ell,\beta} + \hat{C}_{\ell,p}\hat{p} + \hat{C}_{\ell,\bar{q}}\hat{q} + \left(\hat{C}_{\ell,Lp}\hat{C}_{L_1} + \hat{C}_{\ell,p}\right)\hat{p} + \hat{C}_{\ell,\bar{q}}\hat{q} + \hat{C}_{\ell,\tau}\hat{\tau} + \hat{C}_{\ell,\delta}\delta_a + \hat{C}_{\ell,\delta_e}\delta_e \tag{2.13}
\]

\[
\hat{C}_m = \hat{C}_{m_0} + \hat{C}_{m,\alpha} + \hat{C}_{m,\beta} + \hat{C}_{m,p}\hat{p} + \hat{C}_{m,\bar{q}}\hat{q} + \hat{C}_{m,\tau}\hat{\tau} + \hat{C}_{m,\delta}\delta_a + \hat{C}_{m,\delta_e}\delta_e \tag{2.14}
\]

\[
\hat{C}_n = \hat{C}_{n_0} + \hat{C}_{n,\alpha} + \hat{C}_{n,\beta} + \left(\hat{C}_{n,Lp}\hat{C}_{L_1} + \hat{C}_{n,p}\right)\hat{p} + \hat{C}_{n,\bar{q}}\hat{q} + \hat{C}_{n,\tau}\hat{\tau} + \left(\hat{C}_{n,L}\delta_a\hat{C}_{L_1} + \hat{C}_{n,\delta}\right)\delta_a + \hat{C}_{n,\delta_e}\delta_e \tag{2.15}
\]

where a hat signifies the coefficient is a function of $\delta_B$. This asymmetric aerodynamic model is used to describe the aerodynamic forces on the BIRE aircraft. Values for these BIRE coefficients are included in Appendix A.
CHAPTER 3
STABILITY THEORY

3.1 Static Stability

For an aircraft to naturally maintain a trimmed flight condition it needs to be statically stable. When subject to small disturbances, a statically stable aircraft will return to its trim state without any additional input from the pilot. There are three primary derivative coefficients that determine an aircraft’s static stability in the three rotation axes. For an aircraft to be statically stable in pitch

\[ \frac{\delta C_m}{\delta \alpha} \equiv C_{m,\alpha} < 0 \] (3.1)

For an aircraft to be statically stable in roll

\[ \frac{\delta C_\ell}{\delta \beta} \equiv C_{\ell,\beta} < 0 \] (3.2)

For an aircraft to be statically stable in yaw

\[ \frac{\delta C_n}{\delta \beta} \equiv C_{n,\beta} > 0 \] (3.3)

3.2 Dynamic Stability

Analyzing an aircraft’s static stability is an important first step in understanding how an aircraft responds to disturbances from a trim flight condition. However, understanding the dynamic response of an aircraft to disturbances is vital when determining the pilot’s actual experience with an aircraft. The dynamic response of an aircraft can be categorized based on its rigid body DOFs into five traditional aircraft dynamic modes. An aircraft is described as being dynamically stable if all of these dynamic modes are convergent and
dampen out with time.

The eigensystem necessary for analyzing the dynamic stability of an aircraft is based on a set of linearized equations of motion. Phillips [29] derives such a set of coupled, linearized, 6-DOF equations of motion for a rigid aircraft. This derivation accounts for the coupled nature of turning flight and some of the coupled terms present in asymmetric aircraft, such as products of inertia. However, Phillips’s derivation makes some assumptions, such as cancelling out some asymmetric force and moment derivative terms, that are not true for a truly asymmetric aircraft such as a BIRE-modified aircraft. For this reason, a sizeable portion of this work has been focused on correctly deriving the general, coupled, linearized dynamic equations of motion for a rigid, asymmetric aircraft. This derivation is presented in the following Sections 3.2.1 - 3.2.2.

### 3.2.1 6-DOF Dynamic Equations

For a rigid-body-approximated aircraft, there are three translational and three rotational DOFs. The general non-linear dynamic equations of motion for an aircraft are

\[
\begin{aligned}
\begin{bmatrix}
\dot{u} \\
\dot{v} \\
\dot{w}
\end{bmatrix}
&= \frac{g}{W}
\begin{bmatrix}
F_{xb} \\
F_{yb} \\
F_{zb}
\end{bmatrix}
+ g
\begin{bmatrix}
-S_\theta \\
S_\phi C_\theta \\
C_\phi C_\theta
\end{bmatrix}
+ \begin{bmatrix}
rv - qw \\
pw - ru \\
qu - pv
\end{bmatrix}
\end{aligned}
\]  
(3.4)

\[
\begin{aligned}
\begin{bmatrix}
\dot{p} \\
\dot{q} \\
\dot{r}
\end{bmatrix}
&= 
\begin{bmatrix}
I_{xb} & -I_{xyb} & -I_{zb} \\
-I_{xyb} & I_{yb} & -I_{zb} \\
-I_{zb} & -I_{yb} & I_{zb}
\end{bmatrix}^{-1}
\begin{bmatrix}
M_{xb} \\
M_{yb} \\
M_{zb}
\end{bmatrix}
+ \begin{bmatrix}
0 & -h_{zb} & h_{yb} \\
h_{zb} & 0 & -h_{xb} \\
-h_{yb} & h_{xb} & 0
\end{bmatrix}
\begin{bmatrix}
p \\
q \\
r
\end{bmatrix}

&+ \begin{bmatrix}
(I_{yb} - I_{zb})qr + I_{yb}(q^2 - r^2) + I_{xb}pq - I_{xyb}pr \\
(I_{zb} - I_{xb})pr + I_{xb}(r^2 - p^2) + I_{yb}qr - I_{xyb}pq \\
(I_{xb} - I_{yb})pq + I_{yb}(p^2 - q^2) + I_{zb}pr - I_{xb}qr
\end{bmatrix}
\end{aligned}
\]  
(3.5)
\[
\begin{align*}
\begin{cases}
\dot{x}_f &= \begin{bmatrix}
C_{\theta}C_{\psi} & S_{\phi}S_{\theta}C_{\psi} - C_{\phi}S_{\psi} & C_{\phi}S_{\theta}C_{\psi} + S_{\phi}S_{\psi}
\end{bmatrix}
\begin{bmatrix}
u \\
v \\
w
\end{bmatrix} + \begin{bmatrix}
V_{wx_f} \\
V_{wy_f}
\end{bmatrix} \\
\dot{y}_f &= \begin{bmatrix}
C_{\theta}S_{\psi} & S_{\phi}S_{\theta}S_{\psi} - C_{\phi}C_{\psi} & C_{\phi}S_{\theta}S_{\psi} + S_{\phi}C_{\psi}
\end{bmatrix}
\begin{bmatrix}
u \\
v \\
w
\end{bmatrix} + \begin{bmatrix}
V_{wx_f} \\
V_{wy_f}
\end{bmatrix} \\
\dot{z}_f &= \begin{bmatrix}
-S_{\theta} & S_{\phi}C_{\theta} & C_{\phi}C_{\theta}
\end{bmatrix}
\begin{bmatrix}
u \\
v \\
w
\end{bmatrix} + \begin{bmatrix}
V_{wx_f} \\
V_{wy_f}
\end{bmatrix}
\end{cases}
\end{align*}
\]

(3.6)

These coupled equations of motion take into account gyroscopic effects as well as aircraft asymmetry. These asymmetries lead to a system in which the longitudinal and lateral dynamics are coupled. The incorporation of the BIRE results in an asymmetric aircraft. Therefore, the typically-applied symmetric approximation must be discarded and the coupled system must be used for BIRE stability analysis.

### 3.2.2 Linearized Dynamic Equations

The 6-DOF coupled dynamic equations are highly nonlinear and do not have closed form solutions. However, closed form solutions are necessary to obtain the properties of the five dynamic modes via linear eigensystem analysis techniques. For this reason, the 6-DOF dynamic equations must be linearized using an analytical tool called small-disturbance theory [30].

In small-disturbance theory, the equations of motion are constrained to small deviations from an equilibrium flight condition. Following Phillips’s [29] derivation of the coupled dimensional eigensystem, turning flight will be used as the equilibrium flight condition in this work as it naturally involves coupling between longitudinal and lateral dynamics and captures other common flight conditions such as steady level flight.

Applying small disturbance theory to the 6-DOF dynamic equations, each variable becomes a sum of the variable’s value at the equilibrium condition, denoted by the subscript “\(o\)”, and the variable’s deviation from the equilibrium state, denoted by the prefix “\(\Delta\)”.

\[
\begin{align*}
\begin{cases}
\dot{\phi} &= \begin{bmatrix}
1 & S_{\phi}S_{\theta}/C_{\theta} & C_{\phi}S_{\theta}/C_{\theta}
\end{bmatrix}
\begin{bmatrix}
p \\
q \\
r
\end{bmatrix} \\
\dot{\theta} &= \begin{bmatrix}
0 & C_{\phi} & -S_{\phi}
\end{bmatrix}
\begin{bmatrix}
p \\
q \\
r
\end{bmatrix} \\
\dot{\psi} &= \begin{bmatrix}
0 & S_{\phi}/C_{\theta} & C_{\phi}/C_{\theta}
\end{bmatrix}
\begin{bmatrix}
p \\
q \\
r
\end{bmatrix}
\end{cases}
\end{align*}
\]

(3.7)
These variable sums can be expressed

\[
\begin{align*}
    u &= u_o + \Delta u \\
    v &= v_o + \Delta v \\
    w &= w_o + \Delta w \\
    p &= p_o + \Delta p \\
    q &= q_o + \Delta q \\
    r &= r_o + \Delta r \\
    x_f &= x_o + \Delta x_f \\
    y_f &= y_o + \Delta y_f \\
    z_f &= z_o + \Delta z_f \\
    \phi &= \phi_o + \Delta \phi \\
    \theta &= \theta_o + \Delta \theta \\
    \psi &= \psi_o + \Delta \psi \\
    F_{x_b} &= F_{x_b,0} + \Delta F_{x_b} \\
    F_{y_b} &= F_{y_b,0} + \Delta F_{y_b} \\
    F_{z_b} &= F_{z_b,0} + \Delta F_{z_b} \\
    W_{x_b} &= W_{x_b,0} + \Delta W_{x_b} \\
    W_{y_b} &= W_{y_b,0} + \Delta W_{y_b} \\
    W_{z_b} &= W_{z_b,0} + \Delta W_{z_b} \\
    M_{x_b} &= M_{x_b,0} + \Delta M_{x_b} \\
    M_{y_b} &= M_{y_b,0} + \Delta M_{y_b} \\
    M_{z_b} &= M_{z_b,0} + \Delta M_{z_b} \\
    \delta_a &= \delta_{a,0} + \Delta \delta_a \\
    \delta_e &= \delta_{e,0} + \Delta \delta_e \\
    \delta_r &= \delta_{r,0} + \Delta \delta_r
\end{align*}
\] (3.8)

The equilibrium components of these variable sums can be simplified by applying a number of assumptions about the equilibrium flight condition. Using turning flight as the equilibrium flight condition, there exists an equilibrium turning rate, \( \Omega \), that is aligned with the \( z_f \) axis and is a function of airspeed and bank angle. This equilibrium turning rate can be expressed

\[
\Omega = \frac{g \tan \psi_o}{V_o}
\] (3.9)

The alignment of the body-fixed coordinate system relative to the earth-fixed coordinate system is arbitrary. For ease of calculation this work will align the aircraft such that the translational velocity vector, with magnitude \( V_o \), is aligned with the \( x_b \) axis. Applying this assumption, the equilibrium velocities become

\[
\begin{align*}
    \begin{bmatrix}
        u \\
        v \\
        w
    \end{bmatrix}
    &=
    \begin{bmatrix}
        V_o \\
        0 \\
        0
    \end{bmatrix}
\end{align*}
\] (3.10)
The equilibrium angular rates can be simplified by assuming the only earth-fixed rotation rate is the turning rate, $\Omega$. Converting these assumptions to body-fixed coordinates, the equilibrium angular rates become

\[
\begin{bmatrix}
  p \\
  q \\
  r
\end{bmatrix}_o = \Omega \begin{bmatrix}
  -S_{\theta_o} \\
  S_{\phi_o} C_{\theta_o} \\
  C_{\phi_o} C_{\theta_o}
\end{bmatrix}
\] (3.11)

The equilibrium aerodynamic forces can be simplified by assuming constant equilibrium airspeed and zero sideforce. Applying these assumptions and rearranging the weight terms, Eq. 3.4 becomes

\[
\begin{bmatrix}
  0 \\
  0 \\
  0
\end{bmatrix} = \begin{bmatrix}
  F_{xb} \\
  0 \\
  F_{zb}
\end{bmatrix} + W \begin{bmatrix}
  -S_{\theta} \\
  S_{\phi_o} C_{\theta_o} \\
  C_{\phi_o} C_{\theta_o}
\end{bmatrix} + W \left\{ \begin{bmatrix}
  r v - q w \\
  p w - r u \\
  qu - p v
\end{bmatrix} \right\}
\] (3.12)

Eq. 3.12 can be simplified further by substituting the general variables with the equilibrium variables, including the equilibrium velocities and angular rates expressed in Eqs. 3.10 and 3.11. Applying these substitutions gives

\[
\begin{bmatrix}
  0 \\
  0 \\
  0
\end{bmatrix} = \begin{bmatrix}
  F_{xb,o} \\
  0 \\
  F_{zb,o}
\end{bmatrix} + W \begin{bmatrix}
  -S_{\theta_o} \\
  S_{\phi_o} C_{\theta_o} \\
  C_{\phi_o} C_{\theta_o}
\end{bmatrix} + \frac{W \Omega V_o}{g} \begin{bmatrix}
  0 \\
  -C_{\phi_o} C_{\theta_o} \\
  S_{\phi_o} C_{\theta_o}
\end{bmatrix}
\] (3.13)

Eq. 3.13 can be simplified further using the definition of the equilibrium turning rate expressed in Eq. 3.9. Solving this simplified version of Eq. 3.13 for the equilibrium aerodynamic forces required to maintain steady turning flight gives

\[
\begin{bmatrix}
  F_{xb} \\
  F_{yb} \\
  F_{zb}
\end{bmatrix}_{o} = W \begin{bmatrix}
  S_{\theta_o} \\
  0 \\
  -C_{\phi_o} C_{\theta_o}
\end{bmatrix}
\] (3.14)
The equilibrium Euler angles can be simplified by substituting the general angular rates in Eq. 3.7 with the equilibrium angular rates expressed in Eq. 3.11. Applying these substitutions, Eq. 3.7 becomes

\[
\begin{bmatrix}
\dot{\phi} \\
\dot{\theta} \\
\dot{\psi}
\end{bmatrix}
= \begin{bmatrix}
0 \\
0 \\
\Omega
\end{bmatrix}
\tag{3.15}
\]

Eq. 3.15 can then be integrated to solve for the equilibrium Euler angles. Using zero heading at time \( t = 0 \) as the initial condition for the integration, the time-dependant equilibrium Euler angles can be expressed

\[
\begin{bmatrix}
\phi \\
\theta \\
\psi
\end{bmatrix}
= \begin{bmatrix}
\phi_o \\
\theta_o \\
\Omega t
\end{bmatrix}
\tag{3.16}
\]

The equilibrium earth-fixed position can be simplified by integrating Eq. 3.6 after substituting the general velocities with the equilibrium velocities expressed in Eq. 3.10. Applying this substitution and integration, the equilibrium position can be expressed

\[
\begin{bmatrix}
x_f \\
y_f \\
z_f
\end{bmatrix}
= \begin{bmatrix}
V_{wx_f} t + V_o C_{\theta_o} S_{\Omega t}/\Omega \\
V_{wy_f} t - V_o C_{\theta_o} C_{\Omega t}/\Omega \\
(V_{wz_f} - V_o S_{\theta_o}) t
\end{bmatrix}
\tag{3.17}
\]

Substituting the equilibrium variables in Eq. 3.8 with the simplified ones expressed in Eqs. 3.10 – 3.17, the simplified small-disturbance variables become

\[
\begin{align*}
\dot{u} &= V_o + \Delta u \\
\dot{v} &= \Delta v \\
\dot{w} &= \Delta w \\
p &= -\frac{g T_{\phi_o} S_{\theta_o}}{V_o} + \Delta p \\
q &= \frac{g T_{\phi_o} S_{\phi_o} C_{\theta_o}}{V_o} + \Delta q \\
r &= \frac{g S_{\phi_o} C_{\theta_o}}{V_o} + \Delta r \\
xf &= V_{wx_f} t + \frac{V_o^2 C_{\theta_o} S_{\Omega t}}{g T_{\phi_o}} + \Delta xf \\
yf &= V_{wy_f} t - \frac{V_o^2 C_{\theta_o} C_{\Omega t}}{g T_{\phi_o}} + \Delta yf \\
zf &= (V_{wz_f} - V_o S_{\theta_o}) t + \Delta zf
\end{align*}
\]
\[ \phi = \phi_o + \Delta \phi \quad \theta = \theta_o + \Delta \theta \quad \psi = \frac{gT_{\phi_o}t}{V_o} + \Delta \psi \]  

(3.18)

\[ F_{xb} = W S_{\theta_o} + \Delta F_{xb} \quad F_{yb} = \Delta F_{yb} \quad F_{z_b} = \frac{-WC_{\theta_o}}{C_{\phi_o}} + \Delta F_{z_b} \]

\[ M_{xb} = M_{xbo} + \Delta M_{xb} \quad M_{yb} = M_{ybo} + \Delta M_{yb} \quad M_{z_b} = M_{zbo} + \Delta M_{z_b} \]

\[ \delta_a = \delta_{ao} + \Delta \delta_a \quad \delta_e = \delta_{eo} + \Delta \delta_e \quad \delta_r = \delta_{ro} + \Delta \delta_r \]

and the simplified small-deviation variable derivatives relevant to Eqs. 3.4 – 3.7 become

\[ \dot{u} = \Delta \dot{u} \quad \dot{v} = \Delta \dot{v} \quad \dot{w} = \Delta \dot{w} \]

\[ \dot{p} = \Delta \dot{p} \quad \dot{q} = \Delta \dot{q} \quad \dot{r} = \Delta \dot{r} \]  

(3.19)

\[ \dot{x}_f = V_w x_f + \frac{T_{\psi_o} V_o C_{\theta_o} C_{\Omega t}}{T_{\phi_o}} + \Delta \dot{x}_f \quad \dot{y}_f = V_w y_f + \frac{T_{\psi_o} V_o C_{\theta_o} S_{\Omega t}}{T_{\phi_o}} + \Delta \dot{y}_f \quad \dot{z}_f = (V_w z_f - V_o S_{\theta_o}) + \Delta \dot{z}_f \]

\[ \dot{\phi} = \Delta \dot{\phi} \quad \dot{\theta} = \Delta \dot{\theta} \quad \dot{\psi} = \frac{gT_{\phi_o}}{V_o} + \Delta \dot{\psi} \]

Applying the sine and cosine compound angle identities and assuming angle deviations are small enough to apply small-angle approximations, the sines and cosines of the simplified small-disturbance angles become

\[ S_{\phi} \approx S_{\phi_o} + C_{\phi_o} \Delta \phi \quad C_{\phi} \approx C_{\phi_o} - S_{\phi_o} \Delta \phi \]

\[ S_{\theta} \approx S_{\theta_o} + C_{\theta_o} \Delta \theta \quad C_{\theta} \approx C_{\theta_o} - S_{\theta_o} \Delta \theta \]  

(3.20)

\[ S_{\psi} \approx S_{\Omega t} + C_{\Omega t} \Delta \psi \quad C_{\psi} \approx C_{\Omega t} - S_{\Omega t} \Delta \psi \]

Before substituting the variables in the coupled equations of motion with the simplified small-disturbance variables, the changes in aerodynamic forces and moments, \( \Delta F \) and \( \Delta M \), can also be simplified using small-disturbance theory. The changes in aerodynamic forces as functions of small-disturbance variables can be expressed
The changes in aerodynamic moments as functions of small-disturbance variables can be expressed

\[
\begin{align*}
\Delta M_{x_b} &= \begin{bmatrix}
\frac{\partial M_{x_b}}{\partial u} & \frac{\partial M_{x_b}}{\partial v} & \frac{\partial M_{x_b}}{\partial w} \\
\frac{\partial M_{y_b}}{\partial u} & \frac{\partial M_{y_b}}{\partial v} & \frac{\partial M_{y_b}}{\partial w} \\
\frac{\partial M_{z_b}}{\partial u} & \frac{\partial M_{z_b}}{\partial v} & \frac{\partial M_{z_b}}{\partial w}
\end{bmatrix}
\begin{bmatrix}
\Delta u \\
\Delta v \\
\Delta w
\end{bmatrix} + \begin{bmatrix}
\frac{\partial M_{x_b}}{\partial p} & \frac{\partial M_{x_b}}{\partial q} & \frac{\partial M_{x_b}}{\partial r} \\
\frac{\partial M_{y_b}}{\partial p} & \frac{\partial M_{y_b}}{\partial q} & \frac{\partial M_{y_b}}{\partial r} \\
\frac{\partial M_{z_b}}{\partial p} & \frac{\partial M_{z_b}}{\partial q} & \frac{\partial M_{z_b}}{\partial r}
\end{bmatrix}
\begin{bmatrix}
\Delta p \\
\Delta q \\
\Delta r
\end{bmatrix}
\end{align*}
\] (3.22)

The aerodynamic forces and moments are dependant on translational accelerations, \(\dot{u}, \dot{v}, \dot{w}\), because it takes time for the vorticity of the main wing to travel to the tail and affect its aerodynamics. However, sideslip has little effect on vorticity, so the sideslip acceleration, \(\dot{\nu}\), can be assumed to be negligible. Assuming sideslip acceleration is negligible, the force and moment derivatives with respect to sideslip acceleration become

\[
\frac{\partial F_{x_b}}{\partial \dot{v}} = \frac{\partial F_{y_b}}{\partial \dot{v}} = \frac{\partial F_{z_b}}{\partial \dot{v}} = \frac{\partial M_{x_b}}{\partial \dot{v}} = \frac{\partial M_{y_b}}{\partial \dot{v}} = \frac{\partial M_{z_b}}{\partial \dot{v}} = 0
\] (3.23)

Additionally, many of the remaining force and moment derivatives with respect to translational acceleration are typically negligible. Two terms, \(F_{z_b,\dot{w}}\) and \(M_{y_b,\dot{w}}\), are typically non-negligible and can be approximated using the aircraft geometry and translational ac-
closures. Following Phillips [31]'s derivation, the remaining translational accelerations can be approximated as

\[ F_{x_b,\dot{u}} \cong F_{z_b,\dot{w}} \cong M_{y_b,\dot{v}} \cong F_{x_b,\dot{w}} \cong 0 \]

\[ F_{z_b,\dot{w}} \cong -\rho S_w S_h l_{wt} \frac{\partial C_{L_w}}{\partial \alpha} \frac{\partial C_{L_b}}{\partial \alpha} \]

\[ M_{y_b,\dot{v}} \cong \rho S_w S_h l_{wt} x_{bh} \frac{\partial C_{L_w}}{\partial \alpha} \frac{\partial C_{L_b}}{\partial \alpha} \]

\[ l_{wt} = \begin{cases} 1.1(x_{but} - x_{bh}) & x_{but} > x_{bh} \\ 0 & x_{bh} > x_{but} \end{cases} \]

Applying these derivative assumptions to the changes in aerodynamic forces and moments gives

\[
\begin{align*}
\Delta F_{x_b} & = \left[ \frac{\partial F_{x_b}}{\partial u} \quad \frac{\partial F_{x_b}}{\partial v} \quad \frac{\partial F_{x_b}}{\partial w} \right] \{ \Delta u \} + \left[ \frac{\partial F_{x_b}}{\partial q} \quad \frac{\partial F_{x_b}}{\partial r} \right] \{ \Delta p \} \\
\Delta F_{y_b} & = \left[ \frac{\partial F_{y_b}}{\partial u} \quad \frac{\partial F_{y_b}}{\partial v} \quad \frac{\partial F_{y_b}}{\partial w} \right] \{ \Delta v \} + \left[ \frac{\partial F_{y_b}}{\partial q} \quad \frac{\partial F_{y_b}}{\partial r} \right] \{ \Delta q \} \\
\Delta F_{z_b} & = \left[ \frac{\partial F_{z_b}}{\partial u} \quad \frac{\partial F_{z_b}}{\partial v} \quad \frac{\partial F_{z_b}}{\partial w} \right] \{ \Delta w \} + \left[ \frac{\partial F_{z_b}}{\partial q} \quad \frac{\partial F_{z_b}}{\partial r} \right] \{ \Delta r \}
\end{align*}
\]

and

\[
\begin{align*}
\Delta M_{x_b} & = \left[ \frac{\partial M_{x_b}}{\partial u} \quad \frac{\partial M_{x_b}}{\partial v} \quad \frac{\partial M_{x_b}}{\partial w} \right] \{ \Delta u \} + \left[ \frac{\partial M_{x_b}}{\partial q} \quad \frac{\partial M_{x_b}}{\partial r} \right] \{ \Delta p \} \\
\Delta M_{y_b} & = \left[ \frac{\partial M_{y_b}}{\partial u} \quad \frac{\partial M_{y_b}}{\partial v} \quad \frac{\partial M_{y_b}}{\partial w} \right] \{ \Delta v \} + \left[ \frac{\partial M_{y_b}}{\partial q} \quad \frac{\partial M_{y_b}}{\partial r} \right] \{ \Delta q \} \\
\Delta M_{z_b} & = \left[ \frac{\partial M_{z_b}}{\partial u} \quad \frac{\partial M_{z_b}}{\partial v} \quad \frac{\partial M_{z_b}}{\partial w} \right] \{ \Delta w \} + \left[ \frac{\partial M_{z_b}}{\partial q} \quad \frac{\partial M_{z_b}}{\partial r} \right] \{ \Delta r \}
\end{align*}
\]
The simplified small-disturbance variables can now be substituted for the general variables in Eqs. 3.4 - 3.7 to create the linearized coupled equations of motion. With these substitutions, Eqs. 3.4 - 3.7 become

\[
\begin{align*}
\Delta \dot{u} &= \frac{g}{W} \left\{ \begin{array}{c}
WS_{\theta_o} + \Delta F_{xb} \\
\Delta F_{yb} \\
-\frac{WC_{\theta_o}}{C_{\phi_o}} + \Delta F_{zb}
\end{array} \right\} + g \left\{ \begin{array}{c}
-(S_{\theta_o} + C_{\theta_o} \Delta \theta) \\
(S_{\phi_o} + C_{\phi_o} \Delta \phi)(C_{\theta_o} - S_{\theta_o} \Delta \theta) \\
(C_{\phi_o} - S_{\phi_o} \Delta \phi)(C_{\theta_o} - S_{\theta_o} \Delta \theta)
\end{array} \right\} \\
\Delta \dot{v} &= \left\{ \begin{array}{c}
\frac{gS_{\phi_o}C_{\theta_o}}{V_o} + \Delta r \\
\frac{gT_{\phi_o}S_{\theta_o}}{V_o} + \Delta p \\
\frac{gT_{\phi_o}C_{\theta_o}}{V_o} + \Delta q
\end{array} \right\} \Delta v - \left\{ \begin{array}{c}
\frac{gS_{\phi_o}C_{\theta_o}}{V_o} + \Delta r \\
\frac{gT_{\phi_o}S_{\theta_o}}{V_o} + \Delta p \\
\frac{gT_{\phi_o}C_{\theta_o}}{V_o} + \Delta q
\end{array} \right\}(V_o + \Delta u)
\end{align*}
\] (3.27)
\[
\begin{align*}
\begin{bmatrix}
I_{xxb} & -I_{xyb} & -I_{xz} \\
-I_{xyb} & I_{yyb} & -I_{yz} \\
-I_{xz} & -I_{yz} & I_{zz}
\end{bmatrix}
\begin{bmatrix}
\Delta \dot{p} \\
\Delta \dot{q} \\
\Delta \dot{r}
\end{bmatrix}
&= 
\begin{bmatrix}
M_{x,0} + \Delta M_{x} \\
M_{y,0} + \Delta M_{y} \\
M_{z,0} + \Delta M_{z}
\end{bmatrix}
+ 
\begin{bmatrix}
0 & -h_{zb} & h_{yb} \\
h_{zb} & 0 & -h_{xb} \\
-h_{yb} & h_{xb} & 0
\end{bmatrix}
\begin{bmatrix}
\frac{-gT_{\thetao} S_{\thetao} C_{\thetao}}{V_o} + \Delta p \\
\frac{qT_{\thetao} S_{\thetao} C_{\thetao}}{V_o} + \Delta q \\
\frac{gS_{\thetao} C_{\thetao}}{V_o} + \Delta r
\end{bmatrix} \\
+ 
\begin{bmatrix}
I_{yyb} - I_{zz} \left( \frac{gT_{\thetao} S_{\thetao} C_{\thetao}}{V_o} + \Delta q \right) \left( \frac{gS_{\thetao} C_{\thetao}}{V_o} + \Delta r \right) \\
I_{zz} - I_{xxb} \left( -\frac{gT_{\thetao} S_{\thetao}}{V_o} + \Delta p \right) \left( \frac{gS_{\thetao} C_{\thetao}}{V_o} + \Delta r \right) \\
I_{xxb} - I_{yyb} \left( -\frac{gT_{\thetao} S_{\thetao}}{V_o} + \Delta p \right) \left( \frac{gT_{\thetao} S_{\thetao} C_{\thetao}}{V_o} + \Delta q \right) \\
\end{bmatrix}
+ 
\begin{bmatrix}
I_{zz} \left( \frac{gS_{\thetao} C_{\thetao}}{V_o} + \Delta r \right)^2 - \left( \frac{gS_{\thetao} C_{\thetao}}{V_o} + \Delta r \right)^2 \\
I_{xxb} \left( \frac{gS_{\thetao} C_{\thetao}}{V_o} + \Delta r \right)^2 - \left( \frac{gT_{\thetao} S_{\thetao}}{V_o} + \Delta p \right)^2 \\
I_{yyb} \left( \frac{-gT_{\thetao} S_{\thetao}}{V_o} + \Delta p \right)^2 - \left( \frac{gT_{\thetao} S_{\thetao} C_{\thetao}}{V_o} + \Delta q \right)^2 \\
\end{bmatrix} \\
\end{align*}
\tag{3.28}
\]

\[
\begin{align*}
\begin{bmatrix}
V_{wxf} & T_{\thetao} V_o C_{\thetao} S_{\thetao} C_{\Omega} \overline{T_{\thetao}} & + \Delta \dot{x}_f \\
V_{yf} & T_{\thetao} V_o C_{\thetao} S_{\thetao} C_{\Omega} \overline{T_{\thetao}} & + \Delta \dot{y}_f \\
V_{zf} - V_o S_{\thetao} & + \Delta \dot{z}_f
\end{bmatrix}
&= 
\begin{bmatrix}
A_{xu} & A_{xv} & A_{xw} \\
A_{yu} & A_{yv} & A_{yw} \\
A_{zu} & A_{zw} & A_{zw}
\end{bmatrix}
\begin{bmatrix}
V_o + \Delta u \\
\Delta v \\
\Delta w
\end{bmatrix}
+ 
\begin{bmatrix}
V_{wxf} \\
V_{wyf} \\
V_{wzf}
\end{bmatrix} \\
\end{align*}
\tag{3.29}
\]

\[A_{xu} = (C_{\thetao} - S_{\thetao} \Delta \theta)(C_{\Omega} - S_{\Omega} \Delta \psi)\]

\[A_{xv} = (S_{\phio} + C_{\thetao} \Delta \phi)(S_{\thetao} + C_{\thetao} \Delta \theta)(C_{\Omega} - S_{\Omega} \Delta \psi) - (C_{\phio} - S_{\phio} \Delta \phi)(S_{\Omega} + C_{\Omega} \Delta \psi)\]

\[A_{xw} = (C_{\phio} - S_{\phio} \Delta \phi)(S_{\thetao} + C_{\thetao} \Delta \theta)(C_{\Omega} - S_{\Omega} \Delta \psi) + (S_{\phio} + C_{\phio} \Delta \phi)(S_{\Omega} + C_{\Omega} \Delta \psi)\]

\[A_{yu} = (C_{\thetao} - S_{\thetao} \Delta \theta)(S_{\Omega} + C_{\Omega} \Delta \psi)\]
\[ A_{yy} = (S_{\phi_0} + C_{\phi_0} \Delta \phi)(S_{\theta_0} + C_{\theta_0} \Delta \theta)(S_{\Omega t} + C_{\Omega t} \Delta \psi) - (C_{\phi_0} - S_{\phi_0} \Delta \phi)(C_{\Omega t} - S_{\Omega t} \Delta \psi) \]

\[ A_{yw} = (C_{\phi_0} - S_{\phi_0} \Delta \phi)(S_{\theta_0} + C_{\theta_0} \Delta \theta)(S_{\Omega t} + C_{\Omega t} \Delta \psi) + (S_{\phi_0} + C_{\phi_0} \Delta \phi)(C_{\Omega t} - S_{\Omega t} \Delta \psi) \]

\[ A_{zw} = -(S_{\theta_0} + C_{\theta_0} \Delta \theta) \]

\[ A_{zx} = (S_{\phi_0} + C_{\phi_0} \Delta \phi)(C_{\theta_0} - S_{\theta_0} \Delta \theta) \]

\[ A_{xz} = (C_{\phi_0} - S_{\phi_0} \Delta \phi)(C_{\theta_0} - S_{\theta_0} \Delta \theta) \]

\[
\begin{align*}
\Delta \dot{\phi} &= \frac{1}{(C_{\phi_0} - S_{\phi_0} \Delta \phi)} \left( \frac{(S_{\phi_0} + C_{\phi_0} \Delta \phi)(S_{\theta_0} + C_{\theta_0} \Delta \theta)}{C_{\phi_0} - S_{\phi_0} \Delta \phi} \right) + \frac{gT_{\phi_0} S_{\phi_0} C_{\theta_0}}{V_o} + \Delta_p \\
\Delta \dot{\theta} &= \left( \frac{(C_{\phi_0} - S_{\phi_0} \Delta \phi)}{C_{\phi_0} - S_{\phi_0} \Delta \phi} \right) + \left( \frac{(S_{\phi_0} + C_{\phi_0} \Delta \phi)}{C_{\phi_0} - S_{\phi_0} \Delta \phi} \right) \left( \frac{gT_{\phi_0} S_{\phi_0} C_{\theta_0}}{V_o} + \Delta q \right) \\
gT_{\phi_0} \frac{S_{\phi_0}}{V_o} + \Delta \psi &= \left( \frac{(S_{\phi_0} + C_{\phi_0} \Delta \phi)}{C_{\phi_0} - S_{\phi_0} \Delta \phi} \right) + \left( \frac{(C_{\phi_0} - S_{\phi_0} \Delta \phi)}{C_{\phi_0} - S_{\phi_0} \Delta \phi} \right) \left( \frac{gS_{\phi_0} C_{\theta_0}}{V_o} + \Delta r \right)
\end{align*}
\]

Expanding each of the twelve equations of motion, applying the \( \Delta F \) and \( \Delta M \) substitutions, and assuming the product of deviations is negligible gives

\[ \Delta \ddot{u} = \frac{g}{W} F_{x_0, u} \Delta u + \frac{g}{W} F_{x_0, v} \Delta v + \frac{g}{W} F_{x_0, w} \Delta w + \frac{g}{W} F_{x_0, p} \Delta p + \frac{g}{W} F_{x_0, q} \Delta q + \frac{g}{W} F_{x_0, r} \Delta r \\
+ \frac{g}{W} F_{x_0, \delta_0} \Delta \delta_0 + \frac{g}{W} F_{x_0, \delta_e} \Delta \delta_e + \frac{g}{W} F_{x_0, \delta_r} \Delta \delta_r - gC_{\theta_0} \Delta \theta + \frac{gS_{\phi_0} C_{\theta_0}}{V_o} \Delta v - \frac{gT_{\phi_0} S_{\phi_0} C_{\theta_0}}{V_o} \Delta w \]

\[ \Delta \ddot{v} = \frac{g}{W} F_{y_0, u} \Delta u + \frac{g}{W} F_{y_0, v} \Delta v + \frac{g}{W} F_{y_0, w} \Delta w + \frac{g}{W} F_{y_0, p} \Delta p + \frac{g}{W} F_{y_0, q} \Delta q + \frac{g}{W} F_{y_0, r} \Delta r \\
+ \frac{g}{W} F_{y_0, \delta_0} \Delta \delta_0 + \frac{g}{W} F_{y_0, \delta_e} \Delta \delta_e + \frac{g}{W} F_{y_0, \delta_r} \Delta \delta_r - gS_{\phi_0} S_{\theta_0} \Delta \theta + gC_{\phi_0} C_{\theta_0} \Delta \phi \]

\[ - \frac{gT_{\phi_0} S_{\phi_0}}{V_o} \Delta w - \frac{gS_{\phi_0} C_{\theta_0}}{V_o} \Delta u - V_o \Delta r \]
\[
\Delta \dot{w} = -\frac{gC_{\theta o}}{C_{\theta o}} + \frac{g}{W} F_{z_b,u} \Delta u + \frac{g}{W} F_{z_b,v} \Delta v + \frac{g}{W} F_{z_b,w} \Delta w + \frac{g}{W} F_{z_b,p} \Delta p + \frac{g}{W} F_{z_b,q} \Delta q + \frac{g}{W} F_{z_b,r} \Delta r \\
+ \frac{g}{W} F_{z_b,u} \Delta w + \frac{g}{W} F_{z_b,v} \Delta \delta a + \frac{g}{W} F_{z_b,w} \Delta \delta e + \frac{g}{W} F_{z_b,p} \Delta \delta r + gC_{\phi o} C_{\theta o} - gC_{\phi o} S_{\theta o} \Delta \theta \\
- gS_{\phi o} C_{\theta o} \Delta \phi + gT_{\phi o} S_{\phi o} C_{\theta o} + \frac{g T_{\phi o} S_{\phi o} C_{\theta o}}{V_o} \Delta u + V_o \Delta q + \frac{g T_{\phi o} S_{\theta o}}{V_o} \Delta v
\]

\[
I_{xxb} \Delta \dot{p} - I_{xyb} \Delta \dot{q} - I_{xxb} \Delta \dot{r} = M_{xx,0} + M_{xb,u} \Delta u + M_{xb,v} \Delta v + M_{xb,w} \Delta w + M_{xb,p} \Delta p + M_{xb,q} \Delta q \\
+ M_{xx,0} \Delta r + M_{xx,\delta a} \Delta \delta a + M_{xx,\delta e} \Delta \delta e + M_{xx,\delta r} \Delta \delta r \\
- h_{xx} \frac{g T_{\phi o} S_{\phi o} C_{\theta o}}{V_o} - h_{xx} \Delta q + h_{y_b} \frac{g S_{\phi o} C_{\theta o}}{V_o} + h_{y_b} \Delta r \\
+ (I_{yyb} - I_{zzb}) \frac{g^2}{V_o} T_{\phi o} S_{\phi o} C_{\theta o}^2 + (I_{yyb} - I_{zzb}) \frac{g T_{\phi o} S_{\phi o} C_{\theta o}}{V_o} \Delta r \\
+ (I_{yyb} - I_{zzb}) \frac{g S_{\phi o} C_{\theta o}}{V_o} \Delta q + I_{yyb} \frac{g^2}{V_o} (T_{\phi o} S_{\phi o} C_{\theta o}^2 - S_{\phi o} C_{\theta o}^2) \\
+ 2 I_{yyb} \frac{g T_{\phi o} S_{\phi o} C_{\theta o}}{V_o} \Delta q - 2 I_{yyb} \frac{g S_{\phi o} C_{\theta o}}{V_o} \Delta r \\
- I_{xxb} \frac{g^2}{V_o} T_{\phi o} S_{\phi o} S_{\theta o} C_{\theta o} - I_{xxb} \frac{g T_{\phi o} S_{\theta o}}{V_o} \Delta q + I_{xxb} \frac{g T_{\phi o} S_{\phi o} C_{\theta o}}{V_o} \Delta p \\
+ I_{xyb} \frac{g^2}{V_o} T_{\phi o} S_{\phi o} S_{\theta o} C_{\theta o} + I_{xyb} \frac{g T_{\phi o} S_{\phi o} C_{\theta o}}{V_o} \Delta r - I_{xyb} \frac{g S_{\phi o} C_{\theta o}}{V_o} \Delta p
\]

\[
-I_{xyb} \Delta \dot{p} + I_{yxb} \Delta \dot{q} - I_{yxb} \Delta \dot{r} = M_{yb,0} + M_{yb,u} \Delta u + M_{yb,v} \Delta v + M_{yb,w} \Delta w + M_{yb,p} \Delta p + M_{yb,q} \Delta q \\
+ M_{yb,0} \Delta r + M_{yb,\delta a} \Delta \delta a + M_{yb,\delta e} \Delta \delta e + M_{yb,\delta r} \Delta \delta r \\
- h_{yb} \frac{g T_{\phi o} S_{\theta o}}{V_o} + h_{yb} \Delta p - h_{yb} \frac{g S_{\phi o} C_{\theta o}}{V_o} - h_{yb} \Delta r \\
- (I_{zzb} - I_{xxb}) \frac{g^2}{V_o} T_{\phi o} S_{\phi o} S_{\theta o} C_{\theta o} - (I_{zzb} - I_{xxb}) \frac{g T_{\phi o} S_{\theta o}}{V_o} \Delta r \\
+ (I_{zzb} - I_{xxb}) \frac{g S_{\phi o} C_{\theta o}}{V_o} \Delta p + I_{zzb} \frac{g^2}{V_o} (S_{\phi o} C_{\theta o}^2 - T_{\phi o} S_{\theta o}^2) \\
+ 2 I_{zzb} \frac{g S_{\phi o} C_{\theta o}}{V_o} \Delta r + 2 I_{zzb} \frac{g T_{\phi o} S_{\phi o} C_{\theta o}}{V_o} \Delta p \\
+ I_{yxb} \frac{g^2}{V_o} T_{\phi o} S_{\phi o} C_{\theta o}^2 + I_{yxb} \frac{g T_{\phi o} S_{\phi o} C_{\theta o}}{V_o} \Delta r + I_{yxb} \frac{g S_{\phi o} C_{\theta o}}{V_o} \Delta q \\
+ I_{yxb} \frac{g^2}{V_o} T_{\phi o} S_{\phi o} S_{\theta o} C_{\theta o} + I_{yxb} \frac{g T_{\phi o} S_{\phi o} C_{\theta o}}{V_o} \Delta q - I_{yxb} \frac{g T_{\phi o} S_{\phi o} C_{\theta o}}{V_o} \Delta p
\]
\[-I_{xx} \Delta \dot{p} - I_{yy} \Delta \dot{q} + I_{xz} \Delta \dot{r} = M_{x,0} + M_{z,b,0} \Delta u + M_{z,b,v} \Delta v + M_{z,b,w} \Delta w + M_{z,b,p} \Delta p \]
\[+ M_{x,q} \Delta q + M_{z,b,r} \Delta r + M_{z,b,\delta_a} \Delta \delta_a + M_{z,b,\delta_e} \Delta \delta_e + M_{z,b,\delta_r} \Delta \delta_r \]
\[+ h_{yx} \frac{gT_{\phi_0}S_{\theta_0}}{V_o} - h_{xy} \Delta p + h_{zx} \frac{gT_{\phi_0}S_{\phi_0}C_{\theta_0}}{V_o} + h_{xz} \Delta q \]
\[-(I_{xx} - I_{yy}) \frac{g^2}{V_o} T_{\phi_0}^2 S_{\phi_0} S_{\theta_0} C_{\theta_0} - (I_{xx} - I_{yy}) \frac{gT_{\phi_0}S_{\theta_0}}{V_o} \Delta q \]
\[+ (I_{xx} - I_{yy}) \frac{gT_{\phi_0}S_{\phi_0}C_{\theta_0}}{V_o} \Delta p + I_{yx} \frac{g^2}{V_o} (T_{\phi_0}^2 S_{\phi_0}^2 - T_{\phi_0}^2 S_{\phi_0}^2 C_{\theta_0}^2) \]
\[-2I_{xy} \frac{gT_{\phi_0}S_{\theta_0}}{V_o} \Delta p - 2I_{xy} \frac{gT_{\phi_0}S_{\phi_0}C_{\theta_0}}{V_o} \Delta q \]
\[-I_{yz} \frac{g^2}{V_o} T_{\phi_0} S_{\phi_0} S_{\theta_0} C_{\theta_0} - I_{yz} \frac{gT_{\phi_0} S_{\theta_0}}{V_o} \Delta r + I_{yz} \frac{gS_{\phi_0} C_{\theta_0}}{V_o} \Delta p \]
\[-I_{xz} \frac{g^2}{V_o} C_{\phi_0} S_{\theta_0} C_{\theta_0} - I_{xz} \frac{gT_{\phi_0} S_{\phi_0} C_{\theta_0}}{V_o} \Delta r - I_{xz} \frac{gS_{\phi_0} C_{\theta_0}}{V_o} \Delta q \]

\[\Delta \dot{x}_f = C_{\theta_0} C_{\Omega t} V_o + C_{\theta_0} C_{\Omega t} \Delta u - C_{\phi_0} S_{\Omega t} V_o \Delta \psi - S_{\theta_0} C_{\Omega t} V_o \Delta \theta \]
\[+ (S_{\phi_0} S_{\theta_0} C_{\Omega t} - C_{\phi_0} S_{\Omega t}) \Delta v + (C_{\phi_0} S_{\theta_0} C_{\Omega t} + S_{\phi_0} S_{\Omega t}) \Delta w - \frac{V_o C_{\theta_0} T_{\phi_0} S_{\Omega t}}{T_{\phi_0}} \]

\[\Delta \dot{y}_f = C_{\theta_0} S_{\Omega t} V_o + C_{\theta_0} S_{\Omega t} \Delta u + C_{\phi_0} C_{\Omega t} V_o \Delta \psi - S_{\theta_0} S_{\Omega t} V_o \Delta \theta \]
\[+ (S_{\phi_0} S_{\theta_0} S_{\Omega t} - C_{\phi_0} C_{\Omega t}) \Delta v + (C_{\phi_0} S_{\theta_0} S_{\Omega t} + S_{\phi_0} C_{\Omega t}) \Delta w - \frac{V_o C_{\theta_0} T_{\phi_0} S_{\Omega t}}{T_{\phi_0}} \]

\[\Delta \dot{z}_f = -S_{\theta_0} V_o - S_{\theta_0} \Delta u - C_{\theta_0} V_o \Delta \theta + S_{\phi_0} C_{\theta_0} \Delta v + C_{\phi_0} C_{\theta_0} \Delta w + V_o S_{\theta_0} \]

\[\Delta \phi = -\frac{gT_{\phi_0} S_{\theta_0}}{V_o} + \Delta p + \frac{gT_{\phi_0} S_{\phi_0}^2 C_{\theta_0} T_{\theta_0}}{V_o} + \frac{gT_{\phi_0} S_{\phi_0}^2 C_{\theta_0} (T_{\phi_0}^2 + 1)}{V_o} \Delta \theta \]
\[+ \frac{gS_{\phi_0} S_{\theta_0}}{V_o} \Delta \phi + S_{\phi_0} T_{\theta_0} \Delta q + \frac{gS_{\phi_0} C_{\phi_0} S_{\theta_0}}{V_o} \Delta \phi \]
\[+ \frac{gS_{\phi_0} C_{\phi_0} C_{\theta_0} (T_{\phi_0}^2 + 1)}{V_o} \Delta \theta - \frac{gS_{\phi_0} S_{\phi_0} C_{\theta_0}}{V_o} \Delta \phi + C_{\phi_0} T_{\theta_0} \Delta r \]
\[ \Delta \dot{\theta} = \frac{g S_{\phi_o}^2 C_{\theta_o}}{V_o} + C_{\phi_o} \Delta q - \frac{g T_{\phi_o} S_{\phi_o}^2 C_{\theta_o}}{V_o} \Delta \phi - \frac{g S_{\phi_o}^2 C_{\theta_o}}{V_o} - S_{\phi_o} \Delta r - \frac{g S_{\phi_o} C_{\phi_o} C_{\theta_o}}{V_o} \Delta \phi \]

\[ \Delta \dot{\psi} = \frac{g T_{\phi_o} S_{\phi_o}^2}{V_o} + \frac{g T_{\phi_o} S_{\phi_o}^2 T_{\theta_o}}{V_o} \Delta \theta + \frac{g S_{\phi_o}^2}{V_o} \Delta \phi + \frac{S_{\phi_o}}{C_{\theta_o}} \Delta q \]

\[ + \frac{g S_{\phi_o} C_{\phi_o}}{V_o} + \frac{g S_{\phi_o} C_{\phi_o} T_{\theta_o}}{V_o} \Delta \theta - \frac{g S_{\phi_o}^2}{V_o} \Delta \phi + \frac{C_{\phi_o}}{C_{\theta_o}} \Delta r - \frac{g T_{\phi_o}}{V_o} \]

where the nonlinear sine and cosine fractions in the \( \Delta \dot{\phi} \) and \( \Delta \dot{\psi} \) equations were simplified by multiplying each fraction by a whole such that each denominator becomes a product of squares and the product of deviations cancels out. An example of this is

\[
\frac{(S_{\phi_o} + C_{\phi_o} \Delta \phi)(S_{\theta_o} + C_{\theta_o} \Delta \theta)}{(C_{\theta_o} - S_{\theta_o} \Delta \theta)} = \frac{(S_{\phi_o} + C_{\phi_o} \Delta \phi)(S_{\theta_o} + C_{\theta_o} \Delta \theta)}{(C_{\theta_o} - S_{\theta_o} \Delta \theta)} \frac{(C_{\theta_o} + S_{\theta_o} \Delta \theta)}{(C_{\theta_o} + S_{\theta_o} \Delta \theta)} \frac{(S_{\phi_o} + C_{\phi_o} \Delta \phi)(S_{\theta_o} + C_{\theta_o} \Delta \theta)}{(C_{\theta_o} - S_{\theta_o} \Delta \theta)}
\]

\[
\frac{(S_{\phi_o} + C_{\phi_o} \Delta \phi)(S_{\theta_o} + C_{\theta_o} \Delta \theta)}{(C_{\theta_o} - S_{\theta_o} \Delta \theta)} = S_{\phi_o} T_{\theta_o} + S_{\phi_o} (T_{\theta_o}^2 + 1) \Delta \theta + C_{\phi_o} T_{\theta_o} \Delta \phi
\]

Simplifying Eq. 3.31 by combining each deviation’s coefficients gives

\[
\left( \frac{W}{g} \right) \Delta \dot{u} = \left( F_{x,b,u} \right) \Delta u + \left( F_{x,b,v} + \frac{WS_{\phi_o} C_{\theta_o}}{V_o} \right) \Delta v + \left( F_{x,b,w} - \frac{WT_{\phi_o} S_{\phi_o} C_{\theta_o}}{V_o} \right) \Delta w
\]

\[ + \left( F_{x,b,p} \right) \Delta p + \left( F_{x,b,q} \right) \Delta q + \left( F_{x,b,r} \right) \Delta r
\]

\[ + \left( 0 \right) \Delta x_f + \left( 0 \right) \Delta y_f + \left( 0 \right) \Delta z_f
\]

\[ + \left( 0 \right) \Delta \phi + \left( - WC_{\theta_o} \right) \Delta \theta + \left( 0 \right) \Delta \psi
\]

\[ + \left( F_{x,b,\delta_u} \right) \Delta \delta_u + \left( F_{x,b,\delta_v} \right) \Delta \delta_v + \left( F_{x,b,\delta_r} \right) \Delta \delta_r
\]

\[ + \left( 0 \right) \]
\[
\begin{aligned}
\left( \frac{W}{g} \right) \Delta \dot{v} &= \left( F_{y_b,v} - \frac{WS_\phi C_{\theta_o}}{V_o} \right) \Delta u + \left( F_{y_b,w} - \frac{WT_\phi S_{\theta_o}}{V_o} \right) \Delta w \\
& \quad + (F_{y_b,p}) \Delta p + (F_{y_b,q}) \Delta q + \left( F_{y_b,r} - \frac{W}{g} V_o \right) \Delta r \\
& \quad + (0) \Delta x_f + (0) \Delta y_f + (0) \Delta z_f \\
& \quad + (WC_{\theta_o} C_{\phi_o}) \Delta \phi + (-WS_\phi S_{\theta_o}) \Delta \theta + (0) \Delta \psi \\
& \quad + (F_{y_b,\delta_a}) \Delta \delta_a + (F_{y_b,\delta_e}) \Delta \delta_e + (F_{y_b,\delta_r}) \Delta \delta_r \\
& \quad + (0)
\end{aligned}
\]

\[
\begin{aligned}
\left( \frac{W}{g} - F_{z_b,\dot{w}} \right) \Delta \dot{w} &= \left( F_{z_b,u} + \frac{WT_\phi S_{\phi_o} C_{\theta_o}}{V_o} \right) \Delta u + \left( F_{z_b,v} + \frac{WT_\phi S_{\theta_o}}{V_o} \right) \Delta v + (F_{z_b,w}) \Delta w \\
& \quad + (F_{z_b,p}) \Delta p + \left( F_{z_b,q} + \frac{W}{g} V_o \right) \Delta q + (F_{z_b,r}) \Delta r \\
& \quad + (0) \Delta x_f + (0) \Delta y_f + (0) \Delta z_f \\
& \quad + (-WC_{\theta_o} S_{\phi_o}) \Delta \phi + (-WC_{\phi_o} S_{\theta_o}) \Delta \theta + (0) \Delta \psi \\
& \quad + (F_{z_b,\delta_a}) \Delta \delta_a + (F_{z_b,\delta_e}) \Delta \delta_e + (F_{z_b,\delta_r}) \Delta \delta_r \\
& \quad + (0)
\end{aligned}
\]
\[I_{xxb} \Delta p - I_{xyb} \Delta q - I_{xbz} \Delta r = (M_{xb,u}) \Delta u + (M_{xb,v}) \Delta v + (M_{xb,w}) \Delta w + \]
\[+ \left( M_{xb,p} + I_{xb} \frac{gT_{\phi o} S_{\phi o} C_{\theta o}}{V_o} - I_{xyb} \frac{gS_{\theta o} C_{\phi o}}{V_o} \right) \Delta p \]
\[+ \left( M_{xb,q} - h_{zb} + (I_{ygb} - I_{zzb}) \frac{gS_{\phi o} C_{\theta o}}{V_o} + 2I_{yzb} \frac{gT_{\phi o} S_{\phi o} C_{\theta o}}{V_o} \right) \Delta q \]
\[+ \left( M_{xb,r} + h_{yb} + (I_{ygb} - I_{zzb}) \frac{gT_{\phi o} S_{\phi o} C_{\theta o}}{V_o} - 2I_{yzb} \frac{gS_{\phi o} C_{\theta o}}{V_o} \right) \Delta r \]
\[+ (0) \Delta x_f + (0) \Delta y_f + (0) \Delta z_f \]
\[+ (0) \Delta \phi + (0) \Delta \theta + (0) \Delta \psi \]
\[+ (M_{xb,\delta_a}) \Delta \delta_a + (M_{xb,\delta_e}) \Delta \delta_e + (M_{xb,\delta_r}) \Delta \delta_r \]
\[+ \left( M_{xb,\delta_o} - h_{zb} \frac{gT_{\phi o} S_{\phi o} C_{\theta o}}{V_o} + h_{yb} \frac{gS_{\phi o} C_{\theta o}}{V_o} \right. \]
\[+ (I_{ygb} - I_{zzb}) \frac{g^2}{V_o^2} T_{\phi o} S_{\phi o}^2 C_{\theta o} C_{\phi o} + I_{yzb} \frac{g^2}{V_o^2} \left( T_{\phi o}^2 S_{\phi o}^2 C_{\phi o}^2 - S_{\phi o}^2 C_{\theta o}^2 \right) \]
\[- I_{xbz} \frac{g^2}{V_o^2} T_{\phi o}^2 S_{\phi o} S_{\theta o} C_{\phi o} + I_{xyb} \frac{g^2}{V_o^2} T_{\phi o} S_{\phi o} S_{\theta o} C_{\phi o} \right) \]
\[-I_{xyb} \Delta \dot{p} + I_{yhb} \Delta \dot{q} - I_{yzb} \Delta \dot{r} - M_{yb,\omega} \Delta \dot{w} = (M_{yb,u}) \Delta u + (M_{yb,v}) \Delta v + (M_{yb,w}) \Delta w \]

\[+ \left( M_{yb,p} + h_{zb} + (I_{zzb} - I_{xxb}) \frac{gS_{\phi_0}C_{\theta_0}}{V_0} + 2I_{xb} \frac{gT_{\phi_0}S_{\theta_0}}{V_0} \right) \Delta p \]

\[+ \left( M_{yb,q} + I_{xyh} \frac{gS_{\phi_0}C_{\theta_0}}{V_0} + I_{yzb} \frac{gT_{\phi_0}S_{\theta_0}}{V_0} \right) \Delta q \]

\[+ \left( M_{yb,r} - h_{xb} - (I_{zzb} - I_{xxb}) \frac{gT_{\phi_0}S_{\theta_0}}{V_0} + 2I_{xb} \frac{gS_{\phi_0}C_{\theta_0}}{V_0} \right) \Delta r \]

\[+ (0) \Delta x_f + (0) \Delta y_f + (0) \Delta z_f \]

\[+ (0) \Delta \phi + (0) \Delta \theta + (0) \Delta \psi \]

\[+ (M_{yb,\delta_a}) \Delta \delta_a + (M_{yb,\delta_e}) \Delta \delta_e + (M_{yb,\delta_r}) \Delta \delta_r \]

\[+ \left( M_{yb,o} - h_{zb} \frac{gT_{\phi_0}S_{\theta_0}}{V_0} - h_{xb} \frac{gS_{\phi_0}C_{\theta_0}}{V_0} \right) \]

\[-(I_{zzb} - I_{xxb}) \frac{g^2}{V_0^2} T_{\phi_0}S_{\phi_0}S_{\theta_0}C_{\theta_0} + I_{xb} \frac{g^2}{V_0^2} (S_{\phi_0}^2 C_{\theta_0}^2 - T_{\phi_0}^2 S_{\theta_0}^2) \]

\[+ I_{xyb} \frac{g^2}{V_0^2} T_{\phi_0}S_{\phi_0}^2 C_{\theta_0}^2 + I_{yzb} \frac{g^2}{V_0^2} T_{\phi_0}^2 S_{\phi_0}S_{\theta_0}C_{\theta_0} \]
\[ -I_{xz_b} \Delta \dot{p} - I_{yz_b} \Delta \dot{q} + I_{zz_b} \Delta \dot{r} = (M_{z_b,u}) \Delta u + (M_{z_b,v}) \Delta v + (M_{z_b,w}) \Delta w \]

\[ + \left( M_{z_b,p} - h_{yb} + (I_{xx_b} - I_{yy_b}) \frac{g T_{\phi_0} S_{\phi_0} C_{\phi_0}}{V_o} - 2 I_{yx_b} \frac{g T_{\phi_0} S_{\phi_0} C_{\phi_0}}{V_o} \right) \Delta p \]

\[ + \left( M_{z_b,q} + h_{xb} - (I_{xx_b} - I_{yy_b}) \frac{g T_{\phi_0} S_{\phi_0} C_{\phi_0}}{V_o} - 2 I_{yx_b} \frac{g T_{\phi_0} S_{\phi_0} C_{\phi_0}}{V_o} \right) \Delta q \]

\[ + \left( M_{z_b,r} - I_{yz_b} \frac{g T_{\phi_0} S_{\phi_0}}{V_o} - I_{xz_b} \frac{g T_{\phi_0} S_{\phi_0} C_{\phi_0}}{V_o} \right) \Delta r \]

\[ + (0) \Delta x_f + (0) \Delta y_f + (0) \Delta z_f \]

\[ + (0) \Delta \phi + (0) \Delta \theta + (0) \Delta \psi \]

\[ + (M_{z_b,\delta_a}) \Delta \delta_a + (M_{z_b,\delta_e}) \Delta \delta_e + (M_{z_b,\delta_r}) \Delta \delta_r \]

\[ + \left( M_{z_b,o} + h_{yb} \frac{g T_{\phi_0} S_{\phi_0}}{V_o} + h_{xb} \frac{g T_{\phi_0} S_{\phi_0} C_{\phi_0}}{V_o} \right) \]

\[ - (I_{xx_b} - I_{yy_b}) \frac{g^2 T_{\phi_0}^2 S_{\phi_0}^2 S_{\theta_o} C_{\theta_o}}{V_o^2} + I_{yx_b} \frac{g^2 T_{\phi_0}^2 S_{\phi_0} C_{\phi_0}^2}{V_o} \]

\[ - I_{yz_b} \frac{g^2 T_{\phi_0}^2 S_{\phi_0} S_{\theta_o} C_{\theta_o} - I_{xz_b} \frac{g^2 T_{\phi_0}^2 S_{\phi_0}^2 C_{\phi_0}^2}{V_o}}{V_o} \]

\[ (3.33) \]

\[ \Delta \dot{x}_f = (C_{\theta_o} C_{\Omega_t}) \Delta u + (S_{\phi_o} S_{\theta_o} C_{\Omega_t} - C_{\phi_o} S_{\Omega_t}) \Delta v + (C_{\phi_o} S_{\theta_o} C_{\Omega_t} + S_{\phi_o} S_{\Omega_t}) \Delta w \]

\[ + (0) \Delta p + (0) \Delta q + (0) \Delta r \]

\[ + (0) \Delta x_f + (0) \Delta y_f + (0) \Delta z_f \]

\[ + (0) \Delta \phi + (-S_{\theta_o} C_{\Omega_t} V_o) \Delta \theta + (-C_{\theta_o} S_{\Omega_t} V_o) \Delta \psi \]

\[ + (0) \Delta \delta_a + (0) \Delta \delta_e + (0) \Delta \delta_r \]

\[ + \left( C_{\theta_o} C_{\Omega_t} V_o - \frac{T_{\psi_0} V_o C_{\theta_o} C_{\Omega_t}}{T_{\phi_0}} \right) \]
\[ \Delta y_f = (C_{\theta_o} S_{\Omega}) \Delta u + (S_{\phi_o} S_{\theta_o} S_{\Omega} - C_{\phi_o} C_{\Omega}) \Delta v + (C_{\phi_o} S_{\theta_o} S_{\Omega} + S_{\phi_o} C_{\Omega}) \Delta w \\
+ (0) \Delta p + (0) \Delta q + (0) \Delta r \\
+ (0) \Delta x_f + (0) \Delta y_f + (0) \Delta z_f \\
+ (0) \Delta \phi + (-S_{\theta_o} S_{\Omega} V_o) \Delta \theta + (C_{\theta_o} C_{\Omega} V_o) \Delta \psi \\
+ (0) \Delta \delta_a + (0) \Delta \delta_e + (0) \Delta \delta_r \\
+ \left( C_{\theta_o} S_{\Omega} V_o - \frac{T_{\psi_o} V_o C_{\theta_o} S_{\Omega}}{T_{\phi_o}} \right) \]

\[ \Delta \dot{z}_f = (-S_{\theta_o}) \Delta u + (S_{\phi_o} C_{\theta_o}) \Delta v + (C_{\phi_o} C_{\theta_o}) \Delta w \\
+ (0) \Delta p + (0) \Delta q + (0) \Delta r \\
+ (0) \Delta x_f + (0) \Delta y_f + (0) \Delta z_f \\
+ (0) \Delta \phi + (-C_{\theta_o} V_o) \Delta \theta + (0) \Delta \psi \\
+ (0) \Delta \delta_a + (0) \Delta \delta_e + (0) \Delta \delta_r \\
+ (0) \]

\[ \Delta \dot{\phi} = (0) \Delta u + (0) \Delta v + (0) \Delta w \\
+ (1) \Delta p + (S_{\phi_o} T_{\theta_o}) \Delta q + (C_{\phi_o} T_{\theta_o}) \Delta r \\
+ (0) \Delta x_f + (0) \Delta y_f + (0) \Delta z_f \\
+ (0) \Delta \phi + \left( \frac{g T_{\phi_o}}{V_o C_{\theta_o}} \right) \Delta \theta + (0) \Delta \psi \\
+ (0) \Delta \delta_a + (0) \Delta \delta_e + (0) \Delta \delta_r \\
+ (0) \]
\[ \Delta \dot{\theta} = (0) \Delta u + (0) \Delta v + (0) \Delta w \]
\[ + (0) \Delta p + (C_{\phi \psi}) \Delta q + (-S_{\phi \psi}) \Delta r \]
\[ + (0) \Delta x_f + (0) \Delta y_f + (0) \Delta z_f \]
\[ + \left( \frac{-gT_{\phi \psi}C_{\theta \phi}}{V_o} \right) \Delta \phi + (0) \Delta \theta + (0) \Delta \psi \]
\[ + (0) \Delta \delta_a + (0) \Delta \delta_e + (0) \Delta \delta_r \]
\[ + (0) \]

\[ \Delta \dot{\psi} = (0) \Delta u + (0) \Delta v + (0) \Delta w \]
\[ + (0) \Delta p + \left( \frac{S_{\phi \psi}}{C_{\theta \psi}} \right) \Delta q + \left( \frac{C_{\phi \psi}}{C_{\theta \psi}} \right) \Delta r \]
\[ + (0) \Delta x_f + (0) \Delta y_f + (0) \Delta z_f \]
\[ + \left( \frac{gT_{\phi \psi}T_{\theta \psi}}{V_o} \right) \Delta \phi + (0) \Delta \theta + (0) \Delta \psi \]
\[ + (0) \Delta \delta_a + (0) \Delta \delta_e + (0) \Delta \delta_r \]
\[ + (0) \]

By arranging Eq. 3.33 into matrix form, the coupled linearized dynamic equations of motion for an asymmetric aircraft can be expressed.
where

\[
[B]_{11} = \begin{bmatrix}
\frac{W}{g} & 0 & 0 \\
0 & \frac{W}{g} & 0 \\
0 & 0 & \frac{W}{g} - F_{zw,\dot{w}}
\end{bmatrix}, \quad [B]_{21} = \begin{bmatrix}
0 & 0 & 0 \\
0 & 0 & -M_{yz,\dot{w}} \\
0 & 0 & 0
\end{bmatrix}
\]

\[
[I] = \begin{bmatrix}
I_{xxb} & -I_{xyb} & -I_{xzb} \\
-I_{xyb} & I_{yyb} & -I_{yzb} \\
-I_{xzb} & -I_{yzb} & I_{zzb}
\end{bmatrix}, \quad [i] = \begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix}, \quad [n] = \begin{bmatrix}
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0
\end{bmatrix}
\]

\[
[A]_{11} = \begin{bmatrix}
F_{x_u} & F_{x_v} + \frac{WS_{\phi_o}C_{\theta_o}}{V_0} & F_{x_w} - \frac{WT_{\phi_o}S_{\phi_o}C_{\theta_o}}{V_0} \\
F_{y_u} - \frac{WS_{\phi_o}C_{\theta_o}}{V_0} & F_{y_v} & F_{y_w} - \frac{WT_{\phi_o}S_{\phi_o}C_{\theta_o}}{V_0} \\
F_{z_u} + \frac{WT_{\phi_o}S_{\phi_o}C_{\theta_o}}{V_0} & F_{z_v} + \frac{WT_{\phi_o}S_{\phi_o}C_{\theta_o}}{V_0} & F_{z_w}
\end{bmatrix}
\]
\[
\begin{bmatrix}
F_{x_b, p} & F_{x_b, q} & F_{x_b, r} \\
F_{y_b, p} & F_{y_b, q} & F_{y_b, r} - \frac{w}{g} V_o \\
F_{z_b, p} & F_{z_b, q} + \frac{w}{g} V_o & F_{z_b, r}
\end{bmatrix}
\]
\[
\begin{bmatrix}
0 & -W C_{\phi o} & 0 \\
W C_{\phi o} C_{\theta o} & -W S_{\phi o} S_{\theta o} & 0 \\
-W S_{\phi o} C_{\theta o} & -W C_{\phi o} S_{\theta o} & 0
\end{bmatrix}
\]
\[
\begin{bmatrix}
M_{x_b, u} & M_{x_b, v} & M_{x_b, w} \\
M_{y_b, u} & M_{y_b, v} & M_{y_b, w} \\
M_{z_b, u} & M_{z_b, v} & M_{z_b, w}
\end{bmatrix}
\]
\[
\begin{bmatrix}
A_{M_{xp}} & A_{M_{xq}} & A_{M_{xr}} \\
A_{M_{yp}} & A_{M_{yq}} & A_{M_{yr}} \\
A_{M_{zp}} & A_{M_{zq}} & A_{M_{zr}}
\end{bmatrix}
\]
\[
A_{M_{xp}} = M_{x_b, p} + I_{x_3} \frac{g T_{\phi o} S_{\phi o} C_{\theta o}}{V_o} - I_{x_3} \frac{g S_{\phi o} C_{\theta o}}{V_o}
\]
\[
A_{M_{xq}} = M_{x_b, q} + I_{x_3} \frac{g T_{\phi o} S_{\phi o} C_{\theta o}}{V_o} + 2 I_{y_3} \frac{g T_{\phi o} S_{\phi o} C_{\theta o}}{V_o} - I_{x_3} \frac{g T_{\phi o} S_{\phi o}}{V_o}
\]
\[
A_{M_{xr}} = M_{x_b, r} + I_{x_3} \frac{g T_{\phi o} S_{\phi o} C_{\theta o}}{V_o} - 2 I_{y_3} \frac{g T_{\phi o} S_{\phi o} C_{\theta o}}{V_o} + I_{y_3} \frac{g T_{\phi o} S_{\phi o}}{V_o}
\]
\[
A_{M_{yp}} = M_{y_b, p} + I_{x_3} \frac{g T_{\phi o} S_{\phi o} C_{\theta o}}{V_o} + 2 I_{y_3} \frac{g T_{\phi o} S_{\phi o} C_{\theta o}}{V_o} - I_{y_3} \frac{g T_{\phi o} S_{\phi o}}{V_o}
\]
\[
A_{M_{yq}} = M_{y_b, q} + I_{x_3} \frac{g T_{\phi o} S_{\phi o} C_{\theta o}}{V_o} + I_{y_3} \frac{g T_{\phi o} S_{\phi o}}{V_o}
\]
\[
A_{M_{yr}} = M_{y_b, r} - I_{x_3} \frac{g T_{\phi o} S_{\phi o} C_{\theta o}}{V_o} + 2 I_{y_3} \frac{g T_{\phi o} S_{\phi o} C_{\theta o}}{V_o} + I_{y_3} \frac{g T_{\phi o} S_{\phi o}}{V_o}
\]
\[
A_{M_{zp}} = M_{z_b, p} - I_{x_3} \frac{g T_{\phi o} S_{\phi o} C_{\theta o}}{V_o} - 2 I_{y_3} \frac{g T_{\phi o} S_{\phi o} C_{\theta o}}{V_o} + I_{y_3} \frac{g T_{\phi o} S_{\phi o}}{V_o}
\]
\[
A_{M_{zq}} = M_{z_b, q} + I_{x_3} \frac{g T_{\phi o} S_{\phi o} C_{\theta o}}{V_o} - I_{x_3} \frac{g T_{\phi o} S_{\phi o}}{V_o}
\]
\[
\begin{bmatrix}
C_{\theta o} C_{\phi t} & S_{\phi o} S_{\theta o} C_{\phi t} - C_{\phi o} S_{\phi t} & C_{\phi o} S_{\theta o} C_{\phi t} + S_{\phi o} S_{\phi t} \\
C_{\phi o} S_{\phi t} & S_{\phi o} S_{\theta o} S_{\phi t} - C_{\phi o} S_{\phi t} & C_{\phi o} S_{\theta o} S_{\phi t} + S_{\phi o} C_{\phi t} \\
-S_{\phi o} & S_{\phi o} C_{\theta o} & C_{\phi o} C_{\theta o}
\end{bmatrix}
\]
\[
\begin{bmatrix}
0 & -S_{\phi o} C_{\phi t} V_o & -C_{\phi o} S_{\phi t} V_o \\
0 & -S_{\phi o} S_{\phi t} V_o & C_{\phi o} C_{\phi t} V_o \\
0 & -C_{\phi o} V_o & 0
\end{bmatrix}
\]
\[
\begin{bmatrix}
1 & S_{\phi o} T_{\phi o} & C_{\phi o} T_{\theta o} \\
0 & C_{\phi o} & -S_{\phi o}
\end{bmatrix}
\]
\[
[A]_{44} = \begin{bmatrix}
0 & \frac{g}{V_o} \frac{T}{\phi_o} C_{\theta_o} & 0 \\
-\frac{g}{V_o} T_{\phi_o} C_{\theta_o} & 0 & 0 \\
0 & \frac{g}{V_o} T_{\phi_o} T_{\theta_o} & 0
\end{bmatrix}
\]

\[
[C]_1 = \begin{bmatrix}
F_{x_b,\delta_a} & F_{x_b,\delta_e} & F_{x_b,\delta_r} \\
F_{y_b,\delta_a} & F_{y_b,\delta_e} & F_{y_b,\delta_r} \\
F_{z_b,\delta_a} & F_{z_b,\delta_e} & F_{z_b,\delta_r}
\end{bmatrix}
\quad
[C]_2 = \begin{bmatrix}
M_{x_b,\delta_a} & M_{x_b,\delta_e} & M_{x_b,\delta_r} \\
M_{y_b,\delta_a} & M_{y_b,\delta_e} & M_{y_b,\delta_r} \\
M_{z_b,\delta_a} & M_{z_b,\delta_e} & M_{z_b,\delta_r}
\end{bmatrix}
\]

### 3.2.3 Coupled Dimensional Eigensystem

For any linear system, the general eigenproblem can be expressed

\[
([A] - \lambda [B])\{\chi\} = 0 \quad (3.35)
\]

where all \(\lambda\)'s and corresponding \(\chi\)'s that satisfy the eigenproblem are the eigenvalues and corresponding eigenvectors of the eigensystem, respectively. The dimensional damping rate, \(\sigma\), frequency, \(\omega_d\), damping ratio, \(\zeta\), and undamped natural frequency, \(\omega_n\), are

\[
\sigma = -\text{real}(\lambda) \quad (3.36)
\]

\[
\omega_d = |\text{imag}(\lambda)| \quad (3.37)
\]

\[
\zeta = \frac{\lambda_1 + \lambda_2}{2\sqrt{\lambda_1 \lambda_2}} \quad (3.38)
\]

\[
\omega_n = \sqrt{\lambda_1 \lambda_2} \quad (3.39)
\]

In addition to these fundamental mode properties, there are two other mode properties that can be derived from the damping rate. The 99% Damping Time is defined as the time it takes a convergent oscillatory mode to damp out 99% its starting amplitude and can be expressed [32]

\[
\text{99\% Damping Time} = \frac{-\ln(0.01)}{\sigma} \quad (3.40)
\]
The *Doubling Time* is defined as the time it takes a divergent mode to double its starting amplitude and can be expressed [3]

\[
\text{Doubling Time} = -\frac{\ln(2)}{\sigma}
\] (3.41)

Applying the linearized dynamic equations of motion to the general eigenproblem, the coupled dimensional aircraft dynamics eigensystem can be expressed

\[
\begin{pmatrix}
[A]_{11} & [A]_{12} & [n] & [A]_{14} \\
[A]_{21} & [A]_{22} & [n] & [n] \\
[A]_{31} & [n] & [n] & [A]_{34} \\
[n] & [A]_{42} & [n] & [A]_{44}
\end{pmatrix}
- \lambda
\begin{pmatrix}
[B]_{11} & [n] & [n] & [n] \\
[B]_{21} & [I] & [n] & [n] \\
[n] & [n] & [n] & [i]
\end{pmatrix}
\begin{pmatrix}
\chi_u \\
\chi_v \\
\chi_w \\
\chi_p \\
\chi_q \\
\chi_r \\
\chi_x_f \\
\chi_y_f \\
\chi_z_f \\
\chi_\phi \\
\chi_\theta \\
\chi_\psi
\end{pmatrix}
= \begin{pmatrix}
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0
\end{pmatrix}
\] (3.42)

### 3.2.4 Dynamic Modes

Solving the coupled dimensional eigensystem results in 12 eigenvalues and 12 corresponding eigenvectors. For each eigenvector, the relative magnitudes of each component can be used to determine what mode that eigenvector corresponds to. The corresponding eigenvalues can then be used in Eqs. 3.36 – 3.39 to determine the properties of that mode.

Six eigenvalues typically correspond to longitudinal dynamics while the other six typically correspond to lateral dynamics. Of the six longitudinal eigenvalues, a pair corresponds to the short period mode, a pair corresponds to the phugoid mode, and the remaining two
identically-zero values correspond to longitudinal rigid-body displacement [33]. Of the six lateral eigenvalues, the two distinct real values correspond to the non-oscillatory roll and spiral modes, the complex pair corresponds to the Dutch roll mode, and the remaining two identically-zero values correspond to lateral rigid-body displacement [32].

**Short Period Mode**

The short-period mode involves an exchange of potential energy and rotational kinetic energy. This exchange results in high-frequency oscillations in AoA and vertical translation [3]. The short-period mode typically lasts for a few seconds and is more damped than the phugoid mode.

**Phugoid Mode**

The phugoid mode involves an exchange of potential energy and translational kinetic energy. This results in a slow oscillation in AoA and vertical translation [3]. The phugoid mode typically lasts several minutes and is lightly damped.

**Roll Mode**

The roll mode involves the roll damping due to the wingspan of the main wing. This results in an aircraft quickly approaching a steady roll rate when the ailerons are deflected [3]. The roll mode is typically very heavily damped and only lasts for fractions of a second for large wingspans.

**Spiral Mode**

The spiral mode involves the relative magnitudes of the aircraft’s roll and yaw stability and manifests in an aircraft slowly changing heading. The spiral mode typically slowly converges or diverges.
Dutch Roll Mode

Dutch roll involves out of phase oscillations in roll, yaw, and sideslip. Dutch roll is typically described by pilots as the aircraft "fishtailing." and typically has a period on the order of 2 to 10 seconds [3].

3.3 Handling Qualities

Military Specification MIL-F-8785C [20] uses the properties of each dynamic mode to categorize how well a pilot would be able to respond to an excitement of that mode. MIL-F-8785C [20] describes four handling quality levels that correspond to a pilot’s rating of the aircraft’s handling characteristics for a mode [2]. MIL-F-8785C [20] also describes three flight phase categories and four aircraft classifications to further refine under what conditions a mode has a specific handling quality level. The handling quality levels, flight phase categories, and aircraft classifications are summarized in Tables 3.1 - 3.3, respectively.

Table 3.1: U.S. Military handling quality levels adapted from Table 10.2.1 in Phillips [2]

<table>
<thead>
<tr>
<th>Level</th>
<th>Rating</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Satisfactory</td>
<td>Handling qualities clearly adequate for mission. Minimal pilot workload required and no degradation in mission effectiveness exists.</td>
</tr>
<tr>
<td>2</td>
<td>Acceptable</td>
<td>Handling qualities adequate to complete mission, but some pilot workload is required and/or mission effectiveness is lessened.</td>
</tr>
<tr>
<td>3</td>
<td>Controllable</td>
<td>Handling qualities adequate for safety, but excessive pilot workload is required and/or mission effectiveness is inadequate.</td>
</tr>
<tr>
<td>4</td>
<td>Uncontrollable</td>
<td>Unofficial level in which handling qualities are inadequate overall.</td>
</tr>
</tbody>
</table>
Table 3.2: U.S. Military flight phase categories adapted from Table 10.2.2 in Phillips [2]

<table>
<thead>
<tr>
<th>Flight Phase</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Category A</td>
<td>Nonterminal rapid or precise maneuvering, such as combat, reconnaissance, and close-formation flying.</td>
</tr>
<tr>
<td>Category B</td>
<td>Nonterminal gradual or imprecise maneuvering, such as climb, cruise, and descent.</td>
</tr>
<tr>
<td>Category C</td>
<td>Terminal gradual or accurate maneuvering, such as takeoff, approach, and landing.</td>
</tr>
</tbody>
</table>

Table 3.3: U.S. Military aircraft classifications adapted from Table 10.2.3 in Phillips [2]

<table>
<thead>
<tr>
<th>Classification</th>
<th>Aircraft Size and Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class I</td>
<td>Small, light aircraft such as light utility, primary trainer, or light observation.</td>
</tr>
<tr>
<td>Class II</td>
<td>Medium-weight, low-maneuverability aircraft such as reconnaissance, tactical bomber, and heavy attack.</td>
</tr>
<tr>
<td>Class III</td>
<td>Large, heavy, low-maneuverability aircraft such as heavy transport, heavy bomber, and airborne command.</td>
</tr>
<tr>
<td>Class IV</td>
<td>High-maneuverability aircraft such as fighter-interceptor, attack, and tactical reconnaissance.</td>
</tr>
</tbody>
</table>

3.3.1 Short Period Mode Requirements

As humans are most sensitive to accelerations, the normal acceleration due to the short period is most important to a pilot’s opinion of an aircraft’s short period dynamic response. This normal acceleration is related to how the load factor, \( n \), changes with AoA, \( \alpha \). This derivative, called acceleration sensitivity, can be expressed

\[
acceleration\ sensitivity = \frac{\partial n}{\partial \alpha} = \frac{C_{L,\alpha}}{C_W} = \frac{\rho V^2 w S_W C_{L,\alpha}}{2W} \quad (3.43)
\]

When the acceleration sensitivity is above a threshold value (3.5 for category A, 0.0 for category B, and 5.0 for category C), the short period frequency and acceleration sensitivity are proportional and can be combined into a single correlation parameter called the control anticipation parameter (CAP) [3]. CAP can be expressed
\[ CAP = \frac{\frac{\omega_{nSP}}{\partial n}}{\partial \alpha} = \frac{2W(\omega_{nSP})^2}{\rho V_\infty^2 S_w C_{L,\alpha}} \] (3.44)

where \( \omega_{nSP} \) is the short period natural frequency. Assuming adequate short period damping, Table 3.4 describes the short period CAP requirements.

Table 3.4: Short period CAP requirements for all aircraft classifications. Adapted from Eq. 10.3.8 in Phillips [3].

<table>
<thead>
<tr>
<th>Level</th>
<th>Category A ([s^{-2}])</th>
<th>Category B ([s^{-2}])</th>
<th>Category C ([s^{-2}])</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.280 &lt; CAP &lt; 3.60</td>
<td>0.085 &lt; CAP &lt; 3.60</td>
<td>0.150 &lt; CAP &lt; 3.60</td>
</tr>
<tr>
<td>2</td>
<td>0.150 &lt; CAP &lt; 10.0</td>
<td>0.038 &lt; CAP &lt; 10.0</td>
<td>0.096 &lt; CAP &lt; 10.0</td>
</tr>
</tbody>
</table>

In addition to CAP, damping plays a roll in pilot opinion of an aircraft’s short period handling qualities. Table 3.5 describes the short period damping requirements for all aircraft classifications, assuming adequate short period CAP requirements. The requirements displayed in Tables 3.4 and 3.5 must both be met for an aircraft to attain the specified short period handling quality level.

Table 3.5: Short period damping requirements for all aircraft classifications. Adapted from Table 10.3.5 in Phillips [3].

<table>
<thead>
<tr>
<th>Level</th>
<th>Category A, C</th>
<th>Category B</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.35 &lt; (\zeta) &lt; 1.30</td>
<td>0.30 &lt; (\zeta) &lt; 2.00</td>
</tr>
<tr>
<td>2</td>
<td>0.25 &lt; (\zeta) &lt; 2.00</td>
<td>0.20 &lt; (\zeta) &lt; 2.00</td>
</tr>
<tr>
<td>3</td>
<td>0.15 &lt; (\zeta)</td>
<td>0.15 &lt; (\zeta)</td>
</tr>
</tbody>
</table>

### 3.3.2 Phugoid Mode Requirements

Due to the phugoid mode having a relatively long period compared to pilot response time, pilot opinion of an aircraft’s phugoid mode is only based on phugoid damping. [3]. Table 3.6 describes the phugoid damping requirements for all flight phases and aircraft classifications.
Table 3.6: Phugoid damping requirements for all flight phases and aircraft classifications. Adapted from Table 10.3.2 in Phillips [3].

<table>
<thead>
<tr>
<th>Level</th>
<th>Damping Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( \zeta &gt; 0.04 )</td>
</tr>
<tr>
<td>2</td>
<td>( \zeta &gt; 0.00 )</td>
</tr>
<tr>
<td>3</td>
<td>Doubling Time &gt; 55 s</td>
</tr>
</tbody>
</table>

3.3.3 Roll Mode Requirements

The roll mode is heavily overdamped and has a short time constant. To the pilot, the time constant is typically short enough that aileron inputs appear to directly control roll rate rather than roll acceleration. Pilot opinion of an aircraft’s roll response is therefore only based on the roll mode time constant, \(1/\sigma\). Table 3.7 describes the roll mode time constant requirements.

Table 3.7: Roll mode time constant requirements. Adapted from Table 10.3.3 in Phillips [3].

<table>
<thead>
<tr>
<th>Flight Phase</th>
<th>Aircraft Class</th>
<th>Level 1 [s]</th>
<th>Level 2 [s]</th>
<th>Level 3 [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>A, C I, IV</td>
<td></td>
<td>1/(\sigma) &lt; 1.0</td>
<td>1/(\sigma) &lt; 1.4</td>
<td>1/(\sigma) &lt; 10.</td>
</tr>
<tr>
<td>A, C II, III</td>
<td></td>
<td>1/(\sigma) &lt; 1.4</td>
<td>1/(\sigma) &lt; 3.0</td>
<td>1/(\sigma) &lt; 10.</td>
</tr>
<tr>
<td>B All</td>
<td></td>
<td>1/(\sigma) &lt; 1.4</td>
<td>1/(\sigma) &lt; 3.0</td>
<td>1/(\sigma) &lt; 10.</td>
</tr>
</tbody>
</table>

3.3.4 Spiral Mode Requirements

The spiral mode is either slowly convergent or slowly divergent. Due to the spiral mode having a relatively long period compared to pilot response time, pilot opinion of an aircraft’s spiral mode is only based on the doubling time remaining high enough that the pilot can correct divergence with minimal effort [3]. Table 3.8 describes the spiral mode doubling time requirements.

3.3.5 Dutch-Roll Mode Requirements

As dutch roll can have a quarter-period close to pilot response time, it can be difficult for a pilot to control an aircraft that has light dutch-roll damping and a short period. Therefore,
Table 3.8: Spiral mode doubling time requirements. Adapted from Table 10.3.4 in Phillips [3].

<table>
<thead>
<tr>
<th>Flight Phase</th>
<th>Aircraft Class</th>
<th>Level 1 [s]</th>
<th>Level 2 [s]</th>
<th>Level 3 [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>A I, IV</td>
<td>Doubling Time &gt; 12</td>
<td>Doubling Time &gt; 12</td>
<td>Doubling Time &gt; 4</td>
<td></td>
</tr>
<tr>
<td>A II, III</td>
<td>Doubling Time &gt; 20</td>
<td>Doubling Time &gt; 12</td>
<td>Doubling Time &gt; 4</td>
<td></td>
</tr>
<tr>
<td>B, C All</td>
<td>Doubling Time &gt; 20</td>
<td>Doubling Time &gt; 12</td>
<td>Doubling Time &gt; 4</td>
<td></td>
</tr>
</tbody>
</table>

the dutch-roll damping ratio, undamped natural frequency, and product of damping ratio and undamped natural frequency are all important to pilot opinion. Table 3.9 describes the dutch-roll frequency and damping requirements for a more specialized subset of the four aircraft classifications.

Table 3.9: Dutch-roll mode frequency and damping requirements. Adapted from Table 10.3.6 in Phillips [3].

<table>
<thead>
<tr>
<th>Level</th>
<th>Flight Phase</th>
<th>Aircraft Class</th>
<th>ζ Req.</th>
<th>ζωn</th>
<th>ωn [rad/s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A I</td>
<td>IV-CO, IV-GA</td>
<td>&gt; 0.40</td>
<td>ωn</td>
<td>&gt; 1.0</td>
</tr>
<tr>
<td>1</td>
<td>A I</td>
<td>I, IV-other</td>
<td>&gt; 0.19</td>
<td>ωn</td>
<td>&gt; 1.0</td>
</tr>
<tr>
<td>1</td>
<td>A II, III</td>
<td>&gt; 0.19</td>
<td>ωn</td>
<td>&gt; 0.4</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>B All</td>
<td>&gt; 0.08</td>
<td>ωn</td>
<td>&gt; 0.4</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>C I, II-C, IV</td>
<td>&gt; 0.08</td>
<td>ωn</td>
<td>&gt; 0.4</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>C II-L, III</td>
<td>&gt; 0.08</td>
<td>ωn</td>
<td>&gt; 0.4</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>All</td>
<td>&gt; 0.02</td>
<td>ωn</td>
<td>&gt; 0.4</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>All</td>
<td>&gt; 0.00</td>
<td>–</td>
<td>–</td>
<td></td>
</tr>
</tbody>
</table>

CO – combat  
GA – ground attack  
C – carrier-based  
L – land based
4.1 Stability Analysis Methodology Outline

The process of identifying and quantifying the stability and handling qualities of the baseline and BIRE aircraft requires three steps:

1. Evaluate the static stability of each aircraft over a range of BIRE rotation angles
2. Compute the linearized dynamic stability of each aircraft over a range of flight conditions
3. Quantify the handling qualities of each aircraft using the dynamic stability results

Once these three steps are completed, the resulting data can be used to compare the stability and handling qualities of each aircraft and draw conclusions about the effects of the BIRE on an aircraft’s stability and handling. The following subsections outline the methodology behind each step.

4.2 Static Stability Analysis Methodology

Compared to the other steps, analyzing the static stability of an aircraft is a simple process. Despite it’s simplicity, however, analyzing an aircraft’s static stability is an important first step in understanding how an aircraft responds to disturbances from a trim flight condition. As outlined in Section 3.1, the static stability of each aircraft can be determined using Eq. 3.1 – 3.3. This analysis will result in three plots in which the value of each stability derivative for each aircraft can be compared over a range of BIRE rotations.
4.3 Dynamic Stability Analysis Methodology

The dynamic stability analysis process can be summarized by the following steps:

1. Choose a flight condition to analyze the stability about
2. Determine all values necessary to solve the coupled dynamic eigenproblem found in Eq. 3.42
3. Solve the coupled eigenproblem numerically to determine the eigensystem’s eigenvectors and eigenvalues
4. Match each eigenvalue to the dynamic mode it represents using the eigenvalue’s corresponding eigenvector
5. Calculate the properties of each dynamic mode
6. Compare the mode properties of each aircraft to draw conclusions on the effects of the BIRE

The following subsections detail each step of this analysis process.

4.3.1 Step 1: Choose Flight Condition

Due to many factors, an aircraft’s dynamic stability can vary depending on which flight condition the aircraft is at and what maneuver, or trim condition, it is flying. For this reason this work intends to analyze the stability of each aircraft at multiple flight conditions and maneuvers.

Table 4.1 describes the flight conditions used for the analysis performed in companion efforts on the BIRE project. The selection reasoning and process of these flight conditions are discussed in Hunsaker et al. [1]. This work focuses on flight condition C2 only, but analyzes the stability of the aircraft at this flight condition for two different maneuvers: 1) steady level flight and 2) steady coordinated turn. This combination of flight condition and maneuvers provides enough results to begin to understand the general trends in the stability design space below stall.
Table 4.1: Flight conditions considered for analysis. Used with permission [1].

<table>
<thead>
<tr>
<th>Label</th>
<th>Description</th>
<th>Altitude [ft]</th>
<th>Velocity [ft/s]</th>
<th>Mach Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>Take-off and Approach</td>
<td>1,000</td>
<td>222</td>
<td>0.2</td>
</tr>
<tr>
<td>T2</td>
<td>Power-On Departure Stall</td>
<td>15,000</td>
<td>201</td>
<td>0.19</td>
</tr>
<tr>
<td>C1</td>
<td>Turbulent Penetration Speed</td>
<td>1,000</td>
<td>890</td>
<td>0.8</td>
</tr>
<tr>
<td>C2</td>
<td>Air Combat Maneuver</td>
<td>15,000</td>
<td>634</td>
<td>0.6</td>
</tr>
<tr>
<td>C3</td>
<td>Maximum Sustained Load Factor</td>
<td>30,000</td>
<td>796</td>
<td>0.8</td>
</tr>
</tbody>
</table>

T – Take-off and Landing Conditions  
C – Cruise Conditions

Note: Transformation of Wind Axes

Before discussing how to determine the values necessary to solving the eigenproblem, it is important to define the coordinate system that is used throughout the dynamic stability analysis. As outlined in Chapter 1, both static and dynamic stability heavily rely on force and moment derivatives. For static stability, the nondimensional derivative coefficients defined in the aerodynamic model can be used directly. However, the dynamic stability analysis requires dimensional derivatives that are relative to a different coordinate system than the aerodynamic model. As shown in Figure 2.1, the aerodynamic model was developed relative to the body-fixed coordinate system. However, as mentioned in Section 3.2.2, the coupled dimensional eigensystem used for dynamic stability analysis was developed about a coordinate system in which the aircraft is aligned with the direction of flight. This coordinate system will be referred to as the wind axes. The transformation of a vector from the body fixed coordinate system to the wind axes is [34]

\[
\begin{pmatrix}
x_w \\
y_w \\
z_w
\end{pmatrix} =
\begin{bmatrix}
\cos \alpha \cos \beta & \sin \beta & \sin \alpha \cos \beta \\
-\cos \alpha \sin \beta & \cos \beta & -\sin \alpha \sin \beta \\
-\sin \alpha & 0 & \cos \alpha
\end{bmatrix}
\begin{pmatrix}
x_b \\
y_b \\
z_b
\end{pmatrix}
\]

(4.1)

The aircraft properties and dimensional force and moment derivatives necessary for the dynamic stability analysis will be transformed from the body fixed coordinate system to the wind axes using Eq. 4.1 as part of the following Step 2.
4.3.2 Step 2: Determine Necessary Values

Referring to Eq. 3.34, there are three primary types of variables used to fill out the eigenproblem matrices: 1) aircraft properties, 2) equilibrium state properties, and 3) dimensional force and moment derivatives. The aircraft properties can be determined by translating the body-fixed values defined in the aircraft model (see Section 2.1) to the wind axes using Eq. 4.1. The equilibrium state angles and freestream velocity are simply determined based on what maneuver and flight condition the aircraft is flying. The dimensional force and moment derivatives require more effort to determine.

When Phillips [29] derives the coupled dynamic eigensystem, he nondimensionalizes the linearized equations before arranging them into the form of the general eigenproblem. This nondimensionalization aims to allow the direct use of the nondimensional aerodynamic coefficients that constitute an aircraft’s aerodynamic model. However, there are two reasons this nondimensionalization is unnecessary for this work’s analysis. First, the author has previously implemented the aerodynamic model in a flight simulator as part of the overall BIRE effort, and this simulator makes the direct computation of the dimensional force and moment derivatives simple. Second, the nondimensional results from solving the eigenproblem must be redimensionalized to convert the mode properties to a useful form. For these reasons, this work calculates the dimensional force and moment derivatives numerically, utilizing a computational implementation of the aerodynamic model and equations of motion.

The computational implementation of the aircraft model was developed as part of a BIRE flight simulator. This simulator is able to calculate the dimensional forces and moments on the baseline or BIRE aircraft for any user-prescribed trim condition. The simulator contains the full rigid body dynamic models for each aircraft and is able to trim the aircraft to fly at each flight condition and maneuver discussed in Step 1. Using this tool and numerical finite differencing, the dimensional force and moment derivatives used in the eigensystem can be determined by:
1. Trimming the aircraft to the desired equilibrium flight condition and maneuver

2. Calculating the body-fixed aerodynamic forces and moments on the aircraft at this equilibrium state

3. Changing a derivative state variable, such as AoA or control surface deflection, and recalculate the body-fixed aerodynamic forces and moments at this new state

4. Using finite differencing and the results from part 3 to determine the force and moment derivatives with respect to the state variable changed in part 3

5. Repeating parts 3-4 for each derivative state variable utilized in the coupled eigensystem

6. Transforming each force and moment derivative from the body-fixed coordinate system to the wind axes using Eq. 4.1

**4.3.3 Step 3: Solve Coupled Eigenproblem**

Once the necessary values have been determined they can be used to solve the coupled eigenproblem. This work uses a numerical C++ linear algebra package named Eigen [35] to solve the 12 by 12 eigensystem for 12 eigenvalues and 12 corresponding eigenvectors. These eigenvalues and eigenvectors can then be used to determine the dynamic mode properties.

**4.3.4 Step 4: Match Eigenvalues to Dynamic Modes**

As discussed in Section 3.2.4, each of the 12 eigenvalue-eigenvector pairings correspond to a dynamic mode. Of these 12, there are typically six longitudinal eigenvalues and six lateral eigenvalues. For the typical six longitudinal eigenvalues, a pair corresponds to the short period mode, a pair corresponds to the phugoid mode, and the remaining two identically-zero values correspond to longitudinal rigid-body displacement [33]. For the typical six lateral eigenvalues, the two distinct real values correspond to the non-oscillatory roll and spiral modes, the complex pair corresponds to the Dutch-roll mode, and the remaining two identically-zero values correspond to lateral rigid-body displacement [32].
One can determine which mode an eigenvalue corresponds to by analyzing the magnitude of each component of the eigenvalue’s corresponding eigenvector. As shown in Eq. 3.42, each eigenvector component represents the change in a state variable. Additionally, these 12 state variables can be organized into four groups based on what type of DOF they represent:

- $x_f$, $y_f$, and $z_f$ represent changes in aircraft position
- $u$, $v$, and $w$ represent changes in aircraft velocity
- $\phi$, $\theta$, and $\psi$ represent changes in aircraft orientation
- $p$, $q$, and $r$ represent changes in aircraft rotation rate

The motion an eigenvector represents can be determined by analyzing the magnitude of a state variable relative to the other state variables in its group. The state variable(s) with the largest magnitude relative to its group corresponds to the primary motion direction excited by its eigenvector’s dynamic mode. One can then match this primary motion direction with the primary motion direction expected of a dynamic mode to determine which dynamic mode the eigenvector represents. It is important to note that as discussed in Section 3.2.4, oscillatory modes have a corresponding complex pair of eigenvalues rather than a single corresponding eigenvalue, so these complex pairs should be matched to their dynamic mode together. The following subsections outline which state variables are expected to be dominant, or have the largest relative magnitude in their group, for each dynamic mode.

For non-traditional aircraft, like the BIRE aircraft, it may prove difficult to match an eigenvector to its dynamic mode based solely on relative magnitudes. If this is the case, the mode properties calculated using the eigenvalue and Eq.s 3.36 – 3.39 can be used in conjunction with the eigenvector component relative magnitudes to get a better picture of the motion represented by an eigenvector.
**Short Period Mode**

The short-period mode involves high-frequency oscillations in AoA and vertical translation [3]. The short period therefore has a dominant $\chi_w$ and $\chi_{zf}$ [33].

**Phugoid Mode**

The phugoid mode involves slow oscillations in AoA, vertical translation, and forward velocity because unlike the short-period, the aircraft remains aligned with the oscillating direction of flight. The phugoid therefore has a dominant $\chi_u$ and $\chi_{zf}$. As the short period and phugoid are both oscillations in vertical translation and AoA, it is sometimes also necessary to use the eigenvalue magnitude to differentiate between the two. The phugoid mode typically has a smaller eigenvalue magnitude than the short period mode [33].

**Roll Mode**

The roll mode involves the aircraft quickly approaching a steady roll rate when the ailerons are deflected. The roll mode therefore has a dominant $\chi_p$ and $\chi_{\phi}$. The roll mode is also heavily damped and therefore typically has a very short 99% damping time on the order of less than a second.

**Spiral Mode**

The spiral mode involves an aircraft slowly changing heading. The spiral mode therefore has a dominant $\chi_{yf}$ and $\chi_{\psi}$. The spiral mode is also typically slowly convergent or slowly divergent so its 99% damping time (if convergent) or doubling time (if divergent) should be on the order of several minutes [32].

**Dutch-Roll Mode**

Dutch-roll involves out of phase oscillations in roll, yaw, and sideslip. The Dutch-roll mode therefore has a dominant $\chi_{\phi}$, $\chi_{\psi}$, and $\chi_{\psi}$. The magnitude of $\chi_{\phi}$ is typically less than the magnitude of $\chi_{\psi}$ because aircraft typically yaw more than roll when experiencing the Dutch-roll mode [32].
Rigid-body Translation Modes

The four identically-zero eigenvalues correspond to the trivial rigid-body translation modes. Two identically-zero eigenvalues represent longitudinal motion and the other two eigenvalues represent lateral motion [32; 33].

4.3.5 Step 5: Calculate Mode Properties

Once each eigenvector has been matched to the dynamic mode it represents, the mode properties of that eigenvector’s mode can be calculated using that eigenvector’s corresponding eigenvalue and Eq.s 3.36 – 3.39. In addition to these fundamental mode properties, the 99% damping time defined in Eq. 3.40 or the doubling time defined in Eq. 3.41 can also be useful derived mode properties.

4.3.6 Step 6: Draw Conclusions from Comparisons

Once the mode properties of each aircraft are determined for a given flight condition, these mode properties can be compared to identify how modifying an aircraft with the BIRE affects its dynamic stability. Plots and tables detailing the mode properties of each aircraft can also be made to succinctly quantify each comparison. Trends in these comparison results can be observed and conclusions can be discussed based on these trends.

4.4 Handling Qualities Analysis Methodology

This work uses the mode properties resulting from the dynamic stability analysis and Tables 3.4 – 3.9 to define the handling quality level of each aircraft for each dynamic mode, each flight condition, and each maneuver. This handling level data can then be presented in plots and tables to show how modifying an aircraft with the BIRE affects its handling qualities. The trends expected to be of most interest to the author are which modes and under what flight condition-maneuver combinations does the BIRE aircraft’s handling level match that of the baseline aircraft. This information should prove insightful in the current BIRE development effort.
CHAPTER 5
RESULTS

5.1 Static Stability Results

The following subsections explore each of the three static stability derivatives. For each
derivative, the static stability properties about its respective rotation axis are presented and
discussed.

5.1.1 Static Pitch Stability

Figure 5.1 displays the pitch stability derivative of the BIRE aircraft as a function of
BIRE rotation angle, $\delta_B$. The baseline aircraft’s pitch stability derivative is also shown
on the plot as a constant value line. As explained in equation 3.1, an aircraft is statically
stable in pitch if the pitch stability derivative is negative. Figure 5.1 shows that both
aircraft are statically unstable in pitch for all $\delta_B$. This is expected, as fighter aircraft are
often statically unstable in pitch for increased maneuverability. Comparing the two aircraft,
Figure 5.1 shows that both aircraft have similar pitch stability at $\delta_B = 0^\circ$. As the BIRE is
rotated, however, the BIRE pitch stability decreases, up to a maximum $C_{m,\alpha}$ at $\delta_B = \pm 90^\circ$.
This is expected because the projected planform area of the BIRE stabilators that resists
pitching motion is comparable to that of the baseline at $\delta_B = 0$, but decreases as the BIRE
is rotated. The intersection points on Figure 5.1 corresponding to each aircraft having an
identical $C_{m,\alpha}$ occurs at $\delta_B = \pm 6^\circ$. This suggests that modifying the baseline aircraft with
a BIRE makes the aircraft statically less stable in pitch for BIRE rotations outside the
range $-6^\circ < \delta_B < 6^\circ$. 
5.1.2 Static Roll Stability

Figure 5.9 displays the roll stability derivative of the BIRE aircraft as a function of BIRE rotation angle, \( \delta_B \). The baseline aircraft’s roll stability derivative is also shown on the plot as a constant value line. As explained in equation 3.2, an aircraft is statically stable in roll if the roll stability derivative is negative. Comparing the two aircraft, Figure 5.9 shows that both aircraft are statically stable in roll. It also shows that the BIRE aircraft has a smaller roll stability derivative magnitude than the baseline. This suggests that modifying the baseline aircraft with a BIRE makes the aircraft less statically stable in roll, but not enough to make it wholly unstable. This is consistent with what is expected of an aircraft with no vertical tail because vertical surfaces provide resistance to rolling motion.

5.1.3 Static Yaw Stability

Figure 5.3 displays the yaw stability derivative of the BIRE aircraft as a function of BIRE rotation angle, \( \delta_B \). The baseline aircraft’s yaw stability derivative is also shown on the plot as a constant value line. As explained in equation 3.3, an aircraft is statically
Fig. 5.2: Roll stability derivative, $C_{l,\beta}$, as a function of BIRE rotation angle, $\delta_B$.

stable in yaw if the yaw stability derivative is positive. Comparing the two aircraft, Figure 5.3 shows that the baseline aircraft is statically stable in yaw while the BIRE aircraft is statically stable in yaw outside the range of $-14^\circ < \delta_B < 14^\circ$. Preliminary trim results suggest the BIRE rotation angle is typically within this range. Outside this range, the static yaw stability derivative can have a larger or smaller magnitude than the baseline aircraft, depending on $\delta_B$. This suggests that modifying the baseline aircraft with a BIRE makes the aircraft less statically stable, even unstable, in yaw for typical values of $\delta_B$. 
Fig. 5.3: Yaw stability derivative, $C_{n,\beta}$, as a function of BIRE rotation angle, $\delta_B$.

5.2 Dynamic Stability Results

When the CG is at its nominal location, each aircraft exhibits non-traditional dynamic modes. This is expected because the baseline is based on a fighter aircraft with a low aspect ratio, highly swept wing. For this reason, each dynamic mode is studied as a function of CG location.

From testing, it was found that shifting the CG forward one foot eliminates the instabilities inherent in the baseline aircraft and produces dynamic modes that line up with those of traditional aircraft. Therefore, each dynamic mode is studied as a function of CG location ranging from one foot forward of the nominal CG location to at the nominal CG location. A step size within this range of 0.2 ft was selected to provide sufficient resolution to understand the stability trends resulting from shifting the CG location.

In addition to CG location, the dynamic modes of each aircraft are studied as functions of bank angle. The two most common flight maneuvers for typical aircraft are steady level flight and steady coordinated turning. Both of these flight maneuvers can be expressed using a bank angle. Thus, by studying the dynamic modes of each aircraft as functions of
bank angle, a more complete understanding of each aircraft’s stability can be achieved. A bank angle of $0^\circ$ corresponds to steady level flight. A bank angle of $60^\circ$ was selected to represent a moderate coordinated turn. Thus, each dynamic mode is studied at bank angles of $0^\circ$ and $60^\circ$.

Combining the CG location and bank angle ranges with the two different aircraft provides a wide variety of comparisons and trends to be determined from the results. The following subsections explore each dynamic mode individually. For each mode, the stability properties of that mode are presented and discussed.

### 5.2.1 Short Period Mode Results

Figure 5.4 displays the short period eigenvalues for the baseline and BIRE aircraft over a range of CG locations and bank angles.

![Fig. 5.4: The short period mode eigenvalues as functions of bank angle and CG location. CG location ranges from +1 ft. forward of the nominal to +0 forward of the nominal location, with the largest marker corresponding to +1 ft. forward and the smallest marker corresponding to +0 ft. forward. The four labeled points indicate at what CG location the aircraft’s short period split into two real eigenvalues.](image-url)
With the CG location shifted 1 foot forward and at a bank angle of 0°, the baseline aircraft’s short period is convergent with a 99% damping time of 5.699 seconds. As the CG location forward shift is decreased from +1 ft to +0 ft, the baseline’s short period frequency decreases until the oscillatory complex conjugate eigenvalues split into two real eigenvalues at a CG location forward shift of +0.4064 ft., as indicated by the labeled split point on Figure 5.4.

With the CG location shifted 1 foot forward and at a bank angle of 60°, the baseline aircraft’s short period is convergent with a 99% damping time of 5.859 seconds. As the CG location forward shift is decreased from +1 ft to +0 ft, the baseline’s short period frequency decreases until the oscillatory complex conjugate eigenvalues split into two real eigenvalues at a CG location forward shift of +0.4250 ft., as indicated by the labeled split point on Figure 5.4.

With the CG location shifted 1 foot forward and at a bank angle of 0°, the BIRE aircraft’s short period is convergent with a 99% damping time of 5.787 seconds. As the CG location forward shift is decreased from +1 ft to +0 ft, the BIRE aircraft’s short period frequency decreases until the oscillatory complex conjugate eigenvalues split into two real eigenvalues at a CG location forward shift of +0.3885 ft., as indicated by the labeled split point on Figure 5.4.

With the CG location shifted 1 foot forward and at a bank angle of 60°, the BIRE aircraft’s short period is convergent with a 99% damping time of 5.954 seconds. As the CG location forward shift is decreased from +1 ft to +0 ft, the BIRE aircraft’s short period frequency decreases until the oscillatory complex conjugate eigenvalues split into two real eigenvalues at a CG location forward shift of +0.4049 ft., as indicated by the labeled split point on Figure 5.4.

Comparing the baseline and BIRE aircraft’s short period properties, both aircraft exhibit similar trends overall. Both aircraft’s short period oscillatory complex conjugate eigenvalues split into two real eigenvalues at a CG forward location shift ranging from +0.3885 to +0.4250 ft. Comparing the baseline and BIRE’s real eigenvalue split point, the BIRE’s
short period eigenvalues split at a lower CG forward location shift than those of the baseline. These trends are consistent for both bank angles. These results suggest that modifying the baseline fighter aircraft with a BIRE slightly decreases the aircraft’s short period damping and slightly increases the aircraft’s short period frequency. Further research is required to understand what the split non-traditional modes represent physically.

### 5.2.2 Phugoid Mode Results

Figure 5.5 displays the phugoid mode eigenvalues for the baseline and BIRE aircraft over a range of CG locations and bank angles.

![Phugoid Mode Eigenvalues](image)

**Fig. 5.5:** The phugoid mode eigenvalues as functions of bank angle and CG location. CG location ranges from +1 ft. forward of the nominal to +0 forward of the nominal location, with the largest marker corresponding to +1 ft. forward and the smallest marker corresponding to +0 ft. forward.

With the CG location shifted 1 foot forward and at a bank angle of 0°, the baseline aircraft’s phugoid mode is convergent with a 99% damping time of 2,673 seconds. As the CG location forward shift is decreased from +1 ft to +0 ft, the damping and frequency oscillate within the convergent region.
With the CG location shifted 1 foot forward and at a bank angle of 60°, the baseline aircraft’s phugoid mode is divergent with a doubling time of 109.9 seconds. As the CG location forward shift is decreased from +1 ft to +0 ft, the frequency remains relatively constant while the damping oscillates within the divergent region.

With the CG location shifted 1 foot forward and at a bank angle of 0°, the BIRE aircraft’s phugoid mode is convergent with a 99% damping time of 3,083 seconds. As the CG location forward shift is decreased from +1 ft to +0 ft, the damping and frequency oscillate within the convergent region.

With the CG location shifted 1 foot forward and at a bank angle of 60°, the BIRE aircraft’s phugoid mode is divergent with a doubling time of 70.32 seconds. As the CG location forward shift is decreased from +1 ft to +0 ft, the frequency remains relatively constant while the damping oscillates within the divergent region.

Comparing the baseline and BIRE aircraft’s phugoid mode properties, the baseline’s phugoid mode generally has more damping and a slightly lower frequency than the BIRE aircraft’s phugoid mode. Both aircraft’s phugoid damping and frequency appear to oscillate as the CG location forward shift changes. These trends are consistent for both bank angles. These results suggest that modifying a fighter aircraft with a BIRE decreases the aircraft’s phugoid mode damping and slightly increases the aircraft’s phugoid mode frequency. Further research is required to understand the oscillatory nature of the baseline’s phugoid mode properties.

5.2.3 Roll Mode Results

Figure 5.6 displays the roll mode damping rates for the baseline and BIRE aircraft over a range of CG locations and bank angles.

With the CG location shifted 1 foot forward and at a bank angle of 0°, the baseline aircraft’s roll mode is convergent with a 99% damping time of 1.911 seconds. As the CG location forward shift is decreased from +1 ft to +0 ft, the damping rate increases roughly linearly from 2.410 1/s to 2.417 1/s.
With the CG location shifted 1 foot forward and at a bank angle of 60°, the baseline aircraft’s roll mode is convergent with a 99% damping time of 1.901 seconds. As the CG location forward shift is decreased from +1 ft to +0 ft, the damping rate increases from 2.423 1/s to 2.434 1/s.

With the CG location shifted 1 foot forward and at a bank angle of 0°, the BIRE aircraft’s roll mode is convergent with a 99% damping time of 1.904 seconds. As the CG location forward shift is decreased from +1 ft to +0 ft, the damping rate slightly increases roughly linearly from 2.418 1/s to 2.423 1/s.

With the CG location shifted 1 foot forward and at a bank angle of 60°, the BIRE aircraft’s roll mode is convergent with a 99% damping time of 1.894 seconds. As the CG location forward shift is decreased from +1 ft to +0 ft, the damping rate increases from 2.431 1/s to 2.443 1/s.

Comparing the baseline and BIRE aircraft’s roll mode properties, the baseline’s roll mode has slightly less damping than the BIRE aircraft. This trend is consistent for both bank angles. This suggests that modifying a fighter aircraft with a BIRE very slightly
increases the roll mode damping of the aircraft. These results do not vary significantly with changes in CG location.

5.2.4 Spiral Mode Results

Figure 5.7 displays the spiral mode damping rates for the baseline and BIRE aircraft over a range of CG locations and bank angles.

Fig. 5.7: The spiral mode damping rates as functions of bank angle and CG location.

With the CG location shifted 1 foot forward and at a bank angle of 0°, the baseline aircraft’s spiral mode is divergent with a doubling time of 93.61 seconds. As the CG location forward shift is decreased from +1 ft to +0 ft, the damping rate slightly increases roughly linearly from -0.0074 1/s to -0.0062 1/s.

With the CG location shifted 1 foot forward and at a bank angle of 60°, the baseline aircraft’s spiral mode is divergent with a doubling time of 1,130 seconds. As the CG location forward shift is decreased from +1 ft to +0 ft, the damping rate oscillates between convergence and divergence with maximum damping rate of 0.0055 1/s and a minimum
damping rate of -0.0049 1/s. The jump between the maximum and minimum damping rates occurs between CG location forward shifts of +0.2 ft and +0.4 ft.

With the CG location shifted 1 foot forward and at a bank angle of 0°, the BIRE aircraft’s spiral mode is divergent with a doubling time of 65.53 seconds. As the CG location forward shift is decreased from +1 ft to +0 ft, the damping rate slightly increases roughly linearly from -0.0106 1/s to -0.0096 1/s.

With the CG location shifted 1 foot forward and at a bank angle of 60°, the BIRE aircraft’s spiral mode is convergent with a 99% damping time of 4,502 seconds. As the CG location forward shift is decreased from +1 ft to +0 ft, the damping rate oscillates between convergence and divergence with maximum damping rate of 0.0174 1/s and a minimum damping rate of -0.0099 1/s. The jump between the maximum and minimum damping rates occurs between CG location forward shifts of +0.2 ft and +0.4 ft.

Comparing the baseline and BIRE aircraft’s spiral mode properties, the baseline’s spiral mode damping rate is less affected by bank angle than the BIRE’s spiral mode damping rate. For steady level flight, the baseline’s spiral mode is more damped than the BIRE aircraft’s spiral mode. When banked at 60°, both aircraft exhibit oscillations in spiral mode damping rate, with a large positive jump in damping rate occurring between CG location forward shifts of +0.2 ft and +0.4 ft. The damping rate jump and oscillations are both more extreme in magnitude for the BIRE aircraft than the baseline aircraft. Additionally, the spiral mode damping rate increases for both aircraft as the aircraft is banked. This suggests that modifying the baseline aircraft with a BIRE decreases the spiral mode damping for steady level flight and increases the spiral mode damping rate sensitivity to CG location when banked. Further research is required to fully understand the spiral mode damping rate oscillations and jumping behavior exhibited by both aircraft while banked.

5.2.5 Dutch Roll Mode Results

Figure 5.8 displays the dutch roll eigenvalues for the baseline and BIRE aircraft over a range of CG locations and bank angles.
Fig. 5.8: The dutch roll mode eigenvalues as functions of bank angle and CG location. CG location ranges from +1 ft. forward of the nominal to +0 forward of the nominal location, with the largest marker corresponding to +1 ft. forward and the smallest marker corresponding to +0 ft. forward.

With the CG location shifted 1 foot forward and at a bank angle of 0°, the baseline aircraft’s dutch roll is convergent with a 99% damping time of 32.49 seconds. As the CG location forward shift is decreased from +1 ft to +0 ft, the dutch roll frequency and damping both slightly decrease.

With the CG location shifted 1 foot forward and at a bank angle of 60°, the baseline aircraft’s dutch roll is convergent with a 99% damping time of 44.96 seconds. As the CG location forward shift is decreased from +1 ft to +0 ft, the dutch roll frequency and damping both slightly decrease.

With the CG location shifted 1 foot forward and at a bank angle of 0°, the BIRE aircraft’s dutch roll eigenvalues are split into two non-traditional real eigenvalues. As the CG location forward shift is decreased from +1 ft to +0 ft, the magnitudes of the real eigenvalues generally increase. However, further research is required to understand what these non-traditional dutch roll modes physically represent.

With the CG location shifted 1 foot forward and at a bank angle of 60°, the BIRE aircraft’s dutch roll eigenvalues are split into two non-traditional real eigenvalues. As the
CG location forward shift is decreased from +1 ft to +0 ft, the magnitudes of the real
eigenvalues generally increase. However, further research is required to understand what
these non-traditional dutch roll modes physically represent.

Comparing the baseline and BIRE aircraft’s dutch roll properties is not possible because
the BIRE aircraft’s dutch roll eigenvalues are split into two non-traditional real eigenvalues.
As stated previously, further research is required to understand what these non-traditional
dutch roll modes physically represent.

5.3 Handling Quality Results

The dynamic stability mode properties presented in the previous section can be used to
understand how modifying the baseline aircraft with a BIRE affects the aircraft’s handling
qualities. However, as shown in the dynamic stability section, at the nominal CG location
some of the baseline and BIRE aircraft dynamic modes do not line up with those of tra-
ditional aircraft. Because the military handling qualities were developed with traditional
aircraft dynamic modes in mind, the handling qualities of the baseline and BIRE aircraft
cannot be quantified with the CG at its nominal location. For this reason, the handling
qualities of each aircraft are examined with the CG shifted one foot forward.

Shifting the CG forward means that the examined handling qualities are not indicative
of those of the actual baseline or BIRE aircraft. However, the handling qualities examined
with a forward-shifted CG can still provide valuable insight into how modifying the baseline
aircraft with a BIRE affects its handling qualities. Therefore, the following handling quality
results are meant only to compare the baseline and BIRE aircraft and not to quantify the
actual handling qualities of these aircraft.

Figure 3.3 compares the handling quality level of each aircraft for each dynamic mode.
Note that flight condition C2 falls in flight phase Category A and both aircraft are military
Class IV. As shown in figure 3.3, the BIRE aircraft maintains the same handling quality
level as the baseline for the short period, phugoid, roll, and spiral modes. However, the
BIRE aircraft’s dutch roll mode has reduced handling quality levels compared to the baseline
aircraft. The baseline aircraft’s dutch roll handling level is reduced from level 2 to level
Fig. 5.9: Handling quality levels of each aircraft for each dynamic mode and bank angle. The BIRE aircraft is able to match the handling quality levels of the Baseline’s phugoid, roll, and spiral modes.

4 with the addition of the BIRE. Note that level 4 is assigned to the BIRE aircraft’s non-traditional dutch roll mode not because the mode properties are outside acceptable handling quality ranges, but rather because the aircraft exhibits non-traditional dutch roll eigenvalues. As stated in Section 5.2.5, further research is required to understand these modes physically and map them to an appropriate handling quality level. Overall, the handling quality trends indicate that modifying the baseline aircraft with a BIRE reduces its dutch roll handling qualities. The BIRE aircraft’s dutch roll mode should therefore be focused on specifically when designing an active controller for a BIRE-modified aircraft.
CHAPTER 6
CONCLUSIONS

6.1 Objectives Summary

This work aims to quantify the stability and handling qualities of a BIRE-modified aircraft in comparison to an unmodified baseline aircraft. The objectives of this work are as follows:

1. Identify how modifying a fighter aircraft with the BIRE affects the static stability, dynamic stability, and handling qualities of that aircraft
2. Quantify these effects for an example fighter aircraft

To accomplish these objectives, this work:

- Established an aircraft model (see Chapter 2)
- Derived the coupled, asymmetric, linearized, dynamic equations of motion necessary for analyzing the stability of an asymmetric aircraft (see Chapter 3)
- Created an eigensystem solver to analyze the dynamic stability of a baseline and BIRE aircraft as functions of CG location and bank angle using the aforementioned coupled equations of motion (see Chapter 4)
- Quantified the stability and handling qualities of the Baseline and BIRE aircraft for different CG locations and bank angles using the aforementioned eigensystem solver, thus satisfying objective 2 (see Chapter 5)
- Explored the trends present in the aforementioned stability and handling quality results to better understand the stability effects of modifying the baseline aircraft with a BIRE, thus satisfying objective 1 (see Chapter 5)

The results that satisfy objectives 1 and 2 are summarized in the following section.
6.2 Results Summary

The static stability analysis suggests that modifying the baseline aircraft with a BIRE:

- Decreases the aircraft’s static pitch stability outside the range of $-6^\circ < \delta_B < 6^\circ$
- Decreases the aircraft’s static roll stability, but not enough to make the aircraft wholly unstable in roll
- Decreases the aircraft’s static yaw stability to unstable levels within the commonly occurring range of $-14^\circ < \delta_B < 14^\circ$

The dynamic stability analysis suggests that modifying the baseline aircraft with a BIRE:

- Slightly decreases the aircraft’s short period damping and slightly increases the aircraft’s short period frequency
- Decreases the aircraft’s phugoid damping and slightly increases the aircraft’s phugoid frequency
- Very slightly increases the aircraft’s roll damping
- Decreases the spiral mode damping for steady level flight and increases the spiral mode damping rate sensitivity to CG location when banked
- Produces a non-traditional dutch roll mode

The handling quality analysis suggests that modifying the baseline aircraft with a BIRE:

- Has negligible effect on the aircraft’s short period, phugoid, roll, and spiral mode handling quality levels
- Decreases the aircraft’s dutch roll handling quality levels

6.3 Future Work

As mentioned in previous sections, the following trends and results require further research to understand fully:
• The physical representation of the non-traditional real eigenvalues of both aircraft’s short period mode

• The physical representation of the non-traditional real eigenvalues of the BIRE aircraft’s dutch roll mode

• The oscillatory nature of both aircraft’s phugoid mode properties

• The spiral mode damping rate oscillations and jumping behavior exhibited by both aircraft while banked

• How the trends observed in this work change when considering a wider range of operating conditions
REFERENCES


APPENDICES
APPENDIX A
AERODYNAMIC MODEL COEFFICIENT VALUES

Table A.1: Baseline F-16 aerodynamic model force derivative coefficients.

<table>
<thead>
<tr>
<th>Linearized Coefficient</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{L,0}$</td>
<td>0.0456</td>
</tr>
<tr>
<td>$C_{L,\alpha}$</td>
<td>3.5791</td>
</tr>
<tr>
<td>$C_{L,\vartheta}$</td>
<td>2.3916</td>
</tr>
<tr>
<td>$C_{L,\delta_e}$</td>
<td>0.5652</td>
</tr>
<tr>
<td>$C_{S,\beta}$</td>
<td>-0.9008</td>
</tr>
<tr>
<td>$C_{S,\varpi}$</td>
<td>-0.0153</td>
</tr>
<tr>
<td>$C_{S,LP}$</td>
<td>0.3318</td>
</tr>
<tr>
<td>$C_{S,\pi}$</td>
<td>0.4357</td>
</tr>
<tr>
<td>$C_{S,\delta_a}$</td>
<td>0.0656</td>
</tr>
<tr>
<td>$C_{S,\delta_r}$</td>
<td>0.1698</td>
</tr>
<tr>
<td>$C_{D,0}$</td>
<td>0.0218</td>
</tr>
<tr>
<td>$C_{D,L}$</td>
<td>-0.0036</td>
</tr>
<tr>
<td>$C_{D,L^2}$</td>
<td>0.1834</td>
</tr>
<tr>
<td>$C_{D,S^2}$</td>
<td>0.6081</td>
</tr>
<tr>
<td>$C_{D,SP}$</td>
<td>0.0768</td>
</tr>
<tr>
<td>$C_{D,\vartheta}$</td>
<td>0.0368</td>
</tr>
<tr>
<td>$C_{D,LP}$</td>
<td>0.7750</td>
</tr>
<tr>
<td>$C_{D,L^2\vartheta}$</td>
<td>-0.1844</td>
</tr>
<tr>
<td>$C_{D,SP}$</td>
<td>-0.7239</td>
</tr>
<tr>
<td>$C_{D,\delta_e}$</td>
<td>-0.0032</td>
</tr>
<tr>
<td>$C_{D,L\delta_e}$</td>
<td>0.1775</td>
</tr>
<tr>
<td>$C_{D,\delta_e^2}$</td>
<td>0.2854</td>
</tr>
<tr>
<td>$C_{D,S\delta_a}$</td>
<td>0.1118</td>
</tr>
<tr>
<td>$C_{D,S\delta_r}$</td>
<td>0.1604</td>
</tr>
</tbody>
</table>
Table A.2: Baseline F-16 aerodynamic model moment derivative coefficients.

<table>
<thead>
<tr>
<th>Linearized Coefficient</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{\ell,\beta}$</td>
<td>-0.0787</td>
</tr>
<tr>
<td>$C_{\ell,p}$</td>
<td>-0.3182</td>
</tr>
<tr>
<td>$C_{\ell,\alpha}$</td>
<td>0.0469</td>
</tr>
<tr>
<td>$C_{\ell,\eta}$</td>
<td>0.1067</td>
</tr>
<tr>
<td>$C_{\ell,\delta_a}$</td>
<td>-0.0741</td>
</tr>
<tr>
<td>$C_{\ell,\delta_r}$</td>
<td>0.0257</td>
</tr>
<tr>
<td>$C_{m_0}$</td>
<td>-0.0077</td>
</tr>
<tr>
<td>$C_{m,\alpha}$</td>
<td>0.1375</td>
</tr>
<tr>
<td>$C_{m,\eta}$</td>
<td>-4.8503</td>
</tr>
<tr>
<td>$C_{m,\delta_e}$</td>
<td>-0.5881</td>
</tr>
<tr>
<td>$C_{n,\beta}$</td>
<td>0.2426</td>
</tr>
<tr>
<td>$C_{n,p}$</td>
<td>0.0131</td>
</tr>
<tr>
<td>$C_{n,\delta_e}$</td>
<td>-0.1005</td>
</tr>
<tr>
<td>$C_{n,\alpha}$</td>
<td>-0.1787</td>
</tr>
<tr>
<td>$C_{n,\delta_a}$</td>
<td>-0.0276</td>
</tr>
<tr>
<td>$C_{n,\delta}$</td>
<td>-0.0077</td>
</tr>
<tr>
<td>$C_{n,\delta_r}$</td>
<td>-0.0877</td>
</tr>
</tbody>
</table>
Table A.3: BIRE force derivative coefficient sinusoidal fit parameters used in Equation 2.9.

<table>
<thead>
<tr>
<th>Linearized Coefficient</th>
<th>$A$</th>
<th>$\omega$</th>
<th>$\phi$</th>
<th>$z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\hat{C}_{L_0}$</td>
<td>-0.0144</td>
<td>2</td>
<td>1.5708</td>
<td>0.0621</td>
</tr>
<tr>
<td>$\hat{C}_{L,\alpha}$</td>
<td>0.1091</td>
<td>2</td>
<td>1.5708</td>
<td>3.5469</td>
</tr>
<tr>
<td>$\hat{C}_{L,\beta}$</td>
<td>-0.7215</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$\hat{C}_{L,\varphi}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$\hat{C}_{L,\delta_a}$</td>
<td>2.0261</td>
<td>2</td>
<td>1.5708</td>
<td>1.5469</td>
</tr>
<tr>
<td>$\hat{C}_{L,\delta_e}$</td>
<td>0.6797</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$\hat{C}_{L,\delta_2}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$\hat{C}_{L,\delta_c}$</td>
<td>0.7646</td>
<td>1</td>
<td>1.5708</td>
<td>-0.1822</td>
</tr>
<tr>
<td>$\hat{C}_{S_0}$</td>
<td>-0.0105</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$\hat{C}_{S,\alpha}$</td>
<td>0.1833</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$\hat{C}_{S,\beta}$</td>
<td>0.6805</td>
<td>2</td>
<td>1.5708</td>
<td>-0.8492</td>
</tr>
<tr>
<td>$\hat{C}_{S,\varphi}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$\hat{C}_{S,\delta_a}$</td>
<td>0.7352</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$\hat{C}_{S,\delta_e}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0209</td>
</tr>
<tr>
<td>$\hat{C}_{S,\delta_2}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-0.0028</td>
</tr>
<tr>
<td>$\hat{C}_{S,\delta_c}$</td>
<td>0.3882</td>
<td>2</td>
<td>1.5708</td>
<td>0.3700</td>
</tr>
<tr>
<td>$\hat{C}_{S,\delta_2}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-0.1146</td>
</tr>
<tr>
<td>$\hat{C}_{S,\delta_a}$</td>
<td>0.0492</td>
<td>2</td>
<td>1.5708</td>
<td>-0.0380</td>
</tr>
<tr>
<td>$\hat{C}_{S,\delta_e}$</td>
<td>-0.0061</td>
<td>1</td>
<td>1.5708</td>
<td>0.0015</td>
</tr>
<tr>
<td>$\hat{C}_{S,\delta_2}$</td>
<td>0.1830</td>
<td>1</td>
<td>1.5708</td>
<td>0</td>
</tr>
<tr>
<td>$\hat{C}_{L,\delta_2}$</td>
<td>-0.0950</td>
<td>1</td>
<td>1.5708</td>
<td>0.4244</td>
</tr>
</tbody>
</table>
Table A.4: BIRE moment derivative coefficient sinusoidal fit parameters used in Equation 2.9.

<table>
<thead>
<tr>
<th>Linearized Coefficient</th>
<th>( A )</th>
<th>( \omega )</th>
<th>( \phi )</th>
<th>( z )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \hat{C}_{\ell_0} )</td>
<td>0.0002</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( \hat{C}<em>{\ell</em>\alpha} )</td>
<td>-0.0023</td>
<td>4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( \hat{C}<em>{\ell</em>\beta} )</td>
<td>0.0017</td>
<td>2</td>
<td>1.5708</td>
<td>-0.0283</td>
</tr>
<tr>
<td>( \hat{C}<em>{\ell</em>\pi} )</td>
<td>0.0040</td>
<td>2</td>
<td>1.5708</td>
<td>-0.3069</td>
</tr>
<tr>
<td>( \hat{C}<em>{\ell</em>\tau} )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( \hat{C}<em>{\ell</em>\kappa} )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0062</td>
</tr>
<tr>
<td>( \hat{C}<em>{\ell</em>\delta_\alpha} )</td>
<td>0.0140</td>
<td>2</td>
<td>1.5708</td>
<td>-0.1065</td>
</tr>
<tr>
<td>( \hat{C}<em>{\ell</em>\delta_\kappa} )</td>
<td>0.0017</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( \hat{C}_{m_0} )</td>
<td>0.0164</td>
<td>2</td>
<td>1.5708</td>
<td>-0.0218</td>
</tr>
<tr>
<td>( \hat{C}<em>{m</em>\alpha} )</td>
<td>-0.1381</td>
<td>2</td>
<td>1.5708</td>
<td>0.2720</td>
</tr>
<tr>
<td>( \hat{C}<em>{m</em>\beta} )</td>
<td>0.8299</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( \hat{C}<em>{m</em>\kappa} )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( \hat{C}<em>{m</em>\pi} )</td>
<td>-2.3551</td>
<td>2</td>
<td>1.5708</td>
<td>-2.5457</td>
</tr>
<tr>
<td>( \hat{C}<em>{m</em>\tau} )</td>
<td>-0.7666</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( \hat{C}<em>{m</em>\delta_\alpha} )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( \hat{C}<em>{m</em>\delta_\kappa} )</td>
<td>-0.9115</td>
<td>1</td>
<td>1.5708</td>
<td>0.2914</td>
</tr>
<tr>
<td>( \hat{C}_{n_0} )</td>
<td>0.0005</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( \hat{C}<em>{n</em>\alpha} )</td>
<td>-0.0929</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( \hat{C}<em>{n</em>\beta} )</td>
<td>-0.3176</td>
<td>2</td>
<td>1.5708</td>
<td>0.2804</td>
</tr>
<tr>
<td>( \hat{C}<em>{n</em>\pi} )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0010</td>
</tr>
<tr>
<td>( \hat{C}<em>{n</em>\tau} )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-0.0689</td>
</tr>
<tr>
<td>( \hat{C}<em>{n</em>\kappa} )</td>
<td>-0.9204</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( \hat{C}<em>{n</em>\delta_\alpha} )</td>
<td>0.2893</td>
<td>2</td>
<td>1.5708</td>
<td>-0.2789</td>
</tr>
<tr>
<td>( \hat{C}<em>{n</em>\delta_\kappa} )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0131</td>
</tr>
<tr>
<td>( \hat{C}<em>{n</em>\kappa_\delta_\alpha} )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0411</td>
</tr>
<tr>
<td>( \hat{C}<em>{n</em>\delta_\kappa} )</td>
<td>-0.3527</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
APPENDIX B
EXAMPLE CODE

The following code provides example C++ implementations of the eigensystem solving process components outlined in Chapter 4. The following function and variable definitions are part of an "Aircraft" class that handles the physical state calculations of a flight simulator. As this flight simulator is built into Unreal Engine 4 [36], some data types, built in engine classes, and macros are Unreal Engine specific. Only the functions relevant to the stability analysis are provided here, so some member functions and variables might be only declared, but not defined here.

Sections taken from Aircraft.h:

```cpp
#pragma once

#include "CoreMinimal.h"
#include "GameFramework/Pawn.h"
#include "Eigen/Dense"
#include "Aircraft.generated.h"

UCLASS()
class FLIGHTSIMUE4_API AAircraft : public APawn
{
    GENERATED_BODY()

    static constexpr double Pi = 3.1415926535897932384626433832795;
    static constexpr double RE_ft = 20888146.3254593;

    public:
        // Sets default values for this pawn's properties
        AAircraft();

        // Called every frame
        virtual void Tick(float DeltaTime) override;

        // Called to bind functionality to input
        virtual void SetupPlayerInputComponent(class UInputComponent* PlayerInputComponent) override;

        // Used to switch between F16 and BIRE aircraft at runtime
        UPROPERTY(EditAnywhere, BlueprintReadWrite)
        bool bIsBireAircraft = false;

        // Used to clamp control surface deflections in blueprints
```
```cpp
32  UPROPERTY(EditAnywhere, BlueprintReadWrite)
33  float daMax = 1.0f;
34
35  UPROPERTY(EditAnywhere, BlueprintReadWrite)
36  float deMax = 1.0f;
37
38  UPROPERTY(EditAnywhere, BlueprintReadWrite)
39  float drMax = 1.0f;
40
41 protected:
42    // Called when the game starts or when spawned
43    virtual void BeginPlay() override;
44
45    // BEGIN FlightSim Functions
46    /** Initializes variables and calculates initial state based on JSON file
47       * @param ConfigFileName = .JSON file path
48       */
49    UFUNCTION(BlueprintCallable, Category = "FlightSim")
50    void InitializeAircraftFromJSON(FString ConfigFileName);
51
52    /** Calculates atmospheric properties (SI units) based in an input geometric altitude, H (m)
53       * @param H = Altitude (m)
54       * @return [Z, T, p, rho, a] = [geopotential altitude (m), temperature (K), pressure (N/m^2), density (kg/m^3), speed of sound (m/s)]
55       */
56    TArray<double> CalculateStdAtmProperties_SI(double H);
57
58    /** Wrapper for CalculateStdAtmProperties_SI that calculates atmospheric properties (english units) based in an input geometric altitude, H (ft)
59       * @param H = Altitude (ft)
60       * @return [Z, T, p, rho, a] = [geopotential altitude (ft), temperature (R), pressure (lbf/ft^2), density (slugs/ft^3), speed of sound (ft/s)]
61       * @see CalculateStdAtmProperties_SI()
62       */
63    TArray<double> CalculateStdAtmProperties_English(double H);
64
65    /** Uses aircraft equations of motion to calculate the change in aircraft states
66       * IntegrateStates_RK4() uses this function to integrate AircraftStates forward in time
67       * @param t = current time (s)
68       * @param States = current AircraftStates member var
69       * @param Controls = current AircraftControls member var
70       * @return Change in state formatted like AircraftStates member var
71       * @see IntegrateStates_RK4()
72       */
73    Eigen::Matrix<double, 13, 1> CalculateStatesChange(double t, Eigen::Matrix<double, 13, 1> States, Eigen::Vector4d Controls);
74
75    /** Integrates AircraftStates forward in time using the Runge-Kutta 4 integration method
76       * @param t0 = current time (s)
77       * @param States = current AircraftStates member var to be passed into CalculateStatesChange()
78       * @param Controls = current AircraftControls member var to be passed into CalculateStatesChange()
79       * @param dt = delta time, or time interval of integration (s)
80       * @return New AircraftStates var after integration
81       * @see CalculateStatesChange()
82    */
```
/* 
    Eigen::Matrix<double, 13, 1> IntegrateStates_RK4(double t0, Eigen::Matrix<double, 13, 1> States, Eigen::
    Vector4d Controls, double dt);

    /** Calculates pseudo aerodynamic forces and moments given the aircraft's current states/controls
     * @param States = current AircraftStates member var
     * @param Controls = current AircraftControls member var
     * @return [Fx, Fy, Fz, Mx, My, Mz] = pseudo aerodynamic forces [Fx, Fy, Fz] and moments [Mx, My, Mz] in
     * body fixed frame (x-forward, y-right, z-down)
     */
    TArray<double> CalculateAeroForces(Eigen::Matrix<double, 13, 1> States, Eigen::Vector4d Controls);

    /** Calculates residual used in the newton’s method step of InitializeFromTrim()
     * @param G = trim solution [alpha, beta, da, de, dr, tau] = [angle of attack (rad), sideslip angle (rad),
     * [AircraftControls]]
     * @param pqr = rotation rates [p, q, r] (rad/s)
     * @return Size-6 TArray of residuals based on modified force and moment equations, arranged like [RFx,
     * RFy, RFz, RMx, RMy, RMz]
     * @see InitializeFromTrim()
     */
    Eigen::Matrix<double, 6, 1> CalculateResidual(Eigen::Matrix<double, 6, 1> G, Eigen::Vector3d pqr);

    /** Calculates initial states/controls based on trim settings in JSON config file and Jacobian trim
     * algorithm
     * Used in InitializeAircraftFromJSON()
     * @see InitializeAircraftFromJSON()
     */
    void InitializeFromTrimJacobian();

    /** Calculates initial states/controls based on trim settings in JSON config file and fixed-point
     * iteration trim algorithm
     * Used in InitializeAircraftFromJSON()
     * @see InitializeAircraftFromJSON()
     */
    void InitializeFromTrimFixedPoint();

    /** Calculates initial states/controls based on state settings in JSON config file
     * Used in InitializeAircraftFromJSON()
     * @see InitializeAircraftFromJSON()
     */
    void InitializeFromState();

    /** Gets AircraftStates as floats for use in Blueprints
     * Converts u, v, w and xf, yf, zf to cm for use in UE
     * @return AircraftStates = (u, v, w, p, q, r, xf, yf, zf, e0, ex, ey, ez)
     */
    UFUNCTION(BlueprintCallable, Category = "FlightSim")
    void GetAircraftStatesUE(float &u, float &v, float &w, float &p, float &q, float &r, float &xf,
                            float &yf, float &zf, float &e0, float &ex, float &ey, float &ez);

    /** Gets AircraftControls as floats for use in Blueprints
     * @return AircraftControls = (da, de, dr, tau)
     */
    UFUNCTION(BlueprintCallable, Category = "FlightSim")
    void GetAircraftControls(float &da, float &de, float &dr, float &tau);
/** Sets AircraftControls from floats for use in Blueprints
 * @param AircraftControls = (da, de, dr, tau)
 */
UFUNCTION(BlueprintCallable, Category = "FlightSim")
void SetAircraftControls(float da, float de, float dr, float tau);

/** Sets which aircraft is being used for dynamics from bool for use in Blueprints
 * @param IsBireAircraft = true for BIRE, false for F16
 */
UFUNCTION(BlueprintCallable, Category = "FlightSim")
void SetIsBireAircraft(bool IsBireAircraft);

/** Gets latitude and longitude [deg] as floats for use in Blueprints */
UFUNCTION(BlueprintCallable, Category = "FlightSim")
void GetLatitudeLongitudeDeg(float &Latitude, float &Longitude);

/** Gets ground-fixed orientation eular angles [deg] as floats for use in Blueprints */
UFUNCTION(BlueprintCallable, Category = "FlightSim")
void GetEulerAngles(float &BankAngle, float &ElevationAngle, float &AzimuthAngle);

/** Gets ground-fixed velocity vector components as floats for use in Blueprints */
UFUNCTION(BlueprintCallable, Category = "FlightSim")
void GetGlobalVelocityVector(float &u, float &v, float &w);

/** Gets mach number as float for use in Blueprints */
UFUNCTION(BlueprintCallable, Category = "FlightSim")
void GetMachNumber(float &MachNumber);

/** Gets load factor (G-force) magnitude as float for use in Blueprints */
/* UFUNCTION(BlueprintCallable, Category = "FlightSim")
void GetLoadFactorMagnitude(float &LoadFactorMagnitude);*/

/** Integrate aircraft states forward in time
 * Should be called each tick
 * @param DeltaTime = DeltaTime from Tick() function
 */
UFUNCTION(BlueprintCallable, Category = "FlightSim")
void TickAircraftStates(float DeltaTime);

/** Integrate aircraft latitude and longitude forward in time
 * Should be called each tick
 * @param dx, dy, dz = change in xf, yf, zf after a single tick's worth of integration
 * @param H1 = altitude before states integration
 */
void TickLatitudeLongitude(float dx, float dy, float dz, float H1);

/** Checks if aircraft is stalled
 * @return True if aircraft is stalled
 */
UFUNCTION(BlueprintCallable, Category = "FlightSim")
bool IsAircraftStalled();

/** Calculates a sine-fit coefficient for BIRE aircraft
 * @param BireCoeffArray = array of sine wave parameters used to calculate specified BIRE aero coefficient
 */
* @param dB = BIRE deflection [rad]
* @return BIRE coefficient value
*/
double FitBIRECoefficient(TArray<double> BireCoeffArray, double dB);

/** Calculates CL,CS,CD,Cl,Cm,Cn for BIRE aircraft using BCL0, BCLA, etc coefficients
* @param alpha = angle of attack [rad]
* @param beta = sideslip angle [rad]
* @param pbar, qbar, rbar = rotation rates [rad/s]
* @param da = aileron deflection [rad]
* @param de = elevator deflection [rad]
* @param dB = BIRE deflection [rad]
* @return Array of BIRE aero coefficients, [CL,CS,CD,Cl,Cm,Cn]
*/
TArray<double> CalculateBIRECoefficients(double alpha, double beta, double pbar, double qbar, double rbar
, double da, double de, double dB);

/** Calculates the compressibility-corrected version of the input aerodynamic coefficient, Coeff, using a
modified Prandtl–Glauert correction
* @param Coeff = aero coefficient to be corrected for compressibility effects
* @param Lambda = half-chord sweep angle of lifting surface related to Coeff
* @param AspectRatio = aspect ratio of lifting surface related to Coeff
* @param MachNum = current mach number
* @return Compressibility-corrected version of Coeff for use in CalculateAeroForces()
*/
double CalculateCompressibilityCorrection(double Coeff, double Lambda, double AspectRatio, double MachNum);

/** Calculates earth-fixed gust velocity and acceleration using randomized damped sinusoidal model
* @param t = current simulation time
* @return <Vgx, Vgy, Vgz, Vgxdot, Vgydot, Vgzdot> where Vg = gust velocity, Vgdot = gust acceleration
*/
void UpdateGustState(double t);

/** Gets GustVelocity as floats for use in Blueprints
* @param GustVelocity = (Vgx, Vgy, Vgz)
*/
UFUNCTION(BlueprintCallable, Category = "FlightSim")
void GetGustVelocity(float & Vgx, float & Vgy, float & Vgz);

/** Gets the calibrated airspeed for use in Blueprints
* @param CAS = Calibrated Airspeed [ft/s]
*/
UFUNCTION(BlueprintCallable, Category = "FlightSim")
void GetCalibratedAirspeed(float & CAS);

/** Logs the input state variable. */
void LogStates(Eigen::Matrix<double, 13, 1> state, FString name);

/** Updates current BIRE inertia values, Ixx to Izz, based on current BIRE angle */
void UpdateBIREInertia();
// END FlightSim Functions
// BEGIN Quaternion Functions

/** Multiplies quaternions qA and qB together */
TArray<double> QuatMult(TArray<double> qA, TArray<double> qB);

/** Converts Euler = [roll, pitch, yaw] (rad) to a quaternion */
TArray<double> EulerToQuat(TArray<double> Euler);

/** Converts quaternion Quat to euler angles = [roll, pitch, yaw] (rad) */
TArray<double> QuatToEuler(TArray<double> Quat);

/** Converts size-3 body fixed vector, BodyFixed, to earth fixed vector using orientation quaternion, Quat */
TArray<double> BodyToEarthFixed(TArray<double> BodyFixed, TArray<double> Quat);

/** Converts size-3 earth fixed vector, EarthFixed, to body fixed vector using orientation quaternion, Quat */
TArray<double> EarthToBodyFixed(TArray<double> EarthFixed, TArray<double> Quat);

/** Normalizes quaternion, Quat, to have magnitude 1 */
TArray<double> QuatNormalize(TArray<double> Quat);

// END Quaternion Functions

// BEGIN Stability Functions

/** Calculates stability mode properties and saves them to file.
* For each eigenvalue, the following mode properties are saved to file (if applicable to mode):
* eigenvector
damping rate, sigma
frequency, omega_d
damping ratio, zeta
undamped natural frequency, omega_n
99% damping time, timeTo99Damp
dothing time, timeToDouble
*/
UFUNCTION(BlueprintCallable, Category = "FlightSim")
void OutputStabilityModeProperties();

/** Solves the stability eigensystem and returns the EigenSolver class that holds the eigenvalues and eigenvectors */
Eigen::EigenSolver<Eigen::Matrix<double, 12, 12>> SolveStabilityEigensystem();

/** Converts a body frame vector to a wind frame vector using the aircraft's current alpha and beta */
Eigen::Vector3<double> BodyToWindFrame(Eigen::Vector3<double> bodyVector);

/** Converts a wind frame vector to a body frame vector using the aircraft's current alpha and beta */
Eigen::Vector3<double> WindToBodyFrame(Eigen::Vector3<double> windVector);
/** Uses finite differencing to calculate dimensional force and moment coefficients necessary to solve
stability eigensystem
* @param windDeltas = array of state deltas in wind frame used to approximate the force and moment
derivatives. windDeltas = [du, dv, dw, dp, dq, dr]
*/
void CalculateForceMomentDerivatives(TArray<double> windDeltas);

// END Stability Functions

// BEGIN Landing Functions
/** Calculates total landing gear forces and moments using a spring-damper system for each of 3 landing
gear
* @param dzFront = front landing gear spring compression (ft) based on blueprint line trace
* @param dzRight and dzLeft are the same as dzFront, but for the right and left landing gear, respectively
* @param dt = time step [sec] for rate calculations
* @param CollisionNormal = normal vector of ground plane from line trace
*/
UFUNCTION (BlueprintCallable, Category = "FlightSim")
void UpdateLandingForces(float dzFront, float dzRight, float dzLeft, float dt, FVector CollisionNormal);

// END Landing Functions

private:
// BEGIN Json Variables
// Json Variables = Aerodynamic variables prescribed in configuration .json file (ConfigFileName from
// InitializeAircraftFromJSON())
// See InitializeAircraftFromJSON() for engineering abbreviation definitions
bool bConstantDensityAtmosphere;
double windMagn;
double windDir;
Eigen::Vector3d windVect;
// Mass/geometry properties
double Sw;
double bw;
double W;
double IxxInit;
double IyyInit;
double IzzInit;
double IxyInit;
double IxzInit;
double IyzInit;
double dIBIRE;
double hx;
double hy;
double hz;
Eigen::Vector3d CGShift;
// Thrust properties
Eigen::Vector3d ThrustLoc;
Eigen::Vector3d ThrustDir;
double ThrustT0;
double ThrustT1;
double ThrustT2;
double ThrustA;
double CDpbar;
double CDqbar;
double CDLqbar;
double CDLqbar;
double CDLqbar;
double CDrbar;
double CDSrbar;
double CDde;
double CDLde;
double CDde2;
double CDda;
double CDSdr;
double Clb;
double Clpbar;
double Clrbar;
double ClLrbar;
double Clda;
double Cldr;
double Cm0;
double Cma;
double Cmqbar;
double Cmde;
double Cnb;
double Cnpbar;
double CnLpbar;
double Cnrbar;
double Cnda;
double CnLda;
double Cndr;

// BIRE coefficient arrays = [A, w, phi, z, delta, multiplier] for each BIRE aero coefficient
// i.e. BCL0 = [BIRE CLO's amplitude (A), BIRE CLO's frequency (w), etc.]
TArray<double> BCL0;
TArray<double> BCLa;
TArray<double> BCLb;
TArray<double> BCSpbar;
TArray<double> BCSLpbar;
TArray<double> BCS0;
TArray<double> BCSa;
TArray<double> BCSb;
TArray<double> BCSpbar;
TArray<double> BCSLpbar;
TArray<double> BCS0;
TArray<double> BCSa;
TArray<double> BCSb;
TArray<double> BCS0;
TArray<double> BCSa;
TArray<double> BCSb;
TArray<double> BCDS;
TArray<double> BCDS2;
TArray<double> BCDS2pbar;
TArray<double> BCDSpbar;
TArray<double> BCDqbar;
TArray<double> BCDLqbar;
TArray<double> BCDrbar;
TArray<double> BCDSrbar;
TArray<double> BCDda;
TArray<double> BCDSda;
TArray<double> BCDde;
TArray<double> BCDLde;
TArray<double> BCDde2;
TArray<double> BC10;
TArray<double> BC1a;
TArray<double> BC1b;
TArray<double> BC1pbar;
TArray<double> BC1qbar;
TArray<double> BC1rbar;
TArray<double> BC1rLbar;
TArray<double> BClda;
TArray<double> BCde;

TArray<double> BCm0;
TArray<double> BCma;
TArray<double> BCmb;
TArray<double> BCmpbar;
TArray<double> BCmqbar;
TArray<double> BCmrbar;
TArray<double> BCMda;
TArray<double> BCMde;

TArray<double> BCn0;
TArray<double> BCna;
TArray<double> BCnb;
TArray<double> BCnpbar;
TArray<double> BCnLpbar;
TArray<double> BCnqbar;
TArray<double> BCnrbar;
TArray<double> BCDnda;
TArray<double> BCDnLda;
TArray<double> BCnde;

// END Json Variables

// BEGIN Precomputed Variables
// Precomputed Variables = Global variables precomputed in InitializeAircraftFromJSON() for computation efficiency
// cw = Mean wing chord
double cw;
// rho0 = Initial air density
double rho0;
// WInv = 1 / W
double WInv;
// gInit = g @ altInit
double gInit;
// Wg = W/g
double Wg;
// Ixx to Izz = Current inertia values
double Ixx;
double Iyy;
double Izz;
double Ixy;
double Ixz;
double Iyz;
// IInv = 3x3 Matrix inverse of aircraft inertia tensor (constructed from Ixx, Ixy, Iyy, etc.)
Eigen::Matrix3d IInv;
// hArray = 3x3 Angular momentum tensor (constructed from hx, hy, hz)
Eigen::Matrix3d hArray;
// Whether elevation angle is supplied in JSON config
bool bElevProvided;
// Whether bank angle is supplied in JSON config
bool bBankProvided;
// a/c = sin()/cos(), b = beta, phi = bank angle, theta = elevation angle, gamma = climb angle (rad)
double sbInit;
double cbInit;
double sphiInit;
double cphiInit;
double sthetaInit;
double cthetaInit;
double sgammaInit;
double cgammaInit;
// END Precomputed Variables

// BEGIN Stability Variables
// Fxyz_uvw = velocity force derivatives = [[Fxu, Fxv, Fxw],
// [Fyu, Fyv, Fyw],
// [Fxu, Fxv, Fxw]]
Eigen::Matrix3d Fxyz_uvw;
// Mxyz_uvw = velocity moment derivatives = [[Mxu, Mxv, Mxw],
// [Myu, Myv, Myw],
// [Mxu, Mxv, Mxw]]
Eigen::Matrix3d Mxyz_uvw;
// Fxyz_pqr = rate force derivatives = [[Fxp, Fxq, Fxr],
// [Fyp, Fyq, Fyr],
// [Fzp, Fzq, Fzr]]
Eigen::Matrix3d Fxyz_pqr;
// Mxyz_pqr = rate moment derivatives = [[Mxp, Mxq, Mxr],
// [Myp, Myq, Myr],
// [Mzp, Mzq, Mzr]]
Eigen::Matrix3d Mxyz_pqr;
double Fz_wdot;
double My_wdot;
// END Stability Variables

// BEGIN Runtime Variables
// Runtime Variables = Global variables storing aircraft states/conditions during runtime
// AircraftStates = [u, v, w, p, q, r, x̘f, y̘f, z̘f, e0, ex, ey, ez]
// u, v, w = x, y, z velocity components in body fixed coordinate frame (x-forward, y-right, z-down) (ft/s)
// p, q, r = rotation rates about x, y, z axis (body fixed frame) (rad/s)
// x̘f, y̘f, z̘f = global position (body fixed frame) (ft)
// e0, ex, ey, ez = quaternion orientation
Eigen::Matrix<double, 13, 1> AircraftStates;
// AircraftControls = [da, de, dr, tau]
// da = aileron deflection (rad)
// de = elevator deflection (rad)
// dr = rudder deflection (rad)
// tau = throttle setting (0 < tau < 1)
Eigen::Vector4d AircraftControls;
// Latitude/Longitude variables (spherical-earth approximation)
// These are updated during runtime but are initialized from JSON
double latitude;
double longitude;
// Landing forces and moments
// LandingForces = [Fx, Fy, Fz, Mx, My, Mz] in body fixed coordinates (lbf, lbf-ft)
TArray<double> LandingForces;
// Current and previous gear deflections are used to calculate rate for landing gear damper force
Eigen::Vector3d CurrentGearDeflections;
Eigen::Vector3d PreviousGearDeflections;
// Gust model global state variables
Eigen::Vector3d GustVelocity;
Eigen::Vector3d GustAmplitude;
Eigen::Vector3d GustOmega;
Eigen::Vector3d GustLambda;
Eigen::Vector3d GustDelay;
Eigen::Vector3d GustPreviousStartTimes;
Eigen::Vector3d GustStartTimes;
Eigen::Vector3d GustAcceleration;
// END Runtime Variables

Sections taken from Aircraft.cpp:

```cpp
doctor AAircraft::OutputStabilityModeProperties() {
    // Disable physics simulation and retrim aircraft
    AAircraft::SetActorTickEnabled(false);
    AAircraft::InitializeAircraftFromJSON("F16_MUx_Adjusted.json");
    // Approximate and save force and moment derivatives
    double du = 1.0;
    double dv = 0.5 * du;
    double dw = du;
    double dp = 0.06;
    double dq = 0.5 * dp;
    double dr = 0.5 * dp;
    TArray<double> deltas = TArray<double>({ du, dv, dw, dp, dq, dr });
    CalculateForceMomentDerivatives(deltas);
```
// Solve eigensystem and save eigenvalues to file
Eigen::EigenSolver<Eigen::Matrix<double, 12, 12>> stabilityEigensystemSolution = 
SolveStabilityEigensystem();

// For each eigenvalue, save the corresponding mode properties
FString aircraftName = "_F16";
if (bIsBireAircraft)
{
    aircraftName = "_BIRE";
}
FString bankCGStr = FString::Printf(TEXT("b%li_cg%li"), int(180./Pi*bankInit), int(100.0 * CGShift(0)));
FString outputCaseFilename = "modeProp " + aircraftName + bankCGStr;
FString outputPathFString = FPaths::ProjectContentDir() + "StabilityOutput/" + outputCaseFilename + ".txt";
UE_LOG(LogTemp, Warning, TEXT("%s"), *outputPathFString)
const char * outputPathCharPtr = StringCast<ANSICHAR>(*outputPathFString).Get();
std::ofstream modeOutputFile;
modeOutputFile.open(outputPathCharPtr);
modeOutputFile << "Eigensystem Solution Mode Properties\n";
modeOutputFile << StringCast<ANSICHAR>(*outputCaseFilename).Get();

for (int i = 0; i < 12; i++)
{
    modeOutputFile << "\n\n---------------------------------------------------------------------";
    std::complex<double> eigenvalue = stabilityEigensystemSolution.eigenvalues()[i];
    modeOutputFile << "\nEigenvalue [1/s] = " << eigenvalue;
}

// Determine which state is dominant for each group: uvw, pqr, xyz, phithetapsi
Eigen::VectorXd eigenvector = stabilityEigensystemSolution.eigenvectors().col(i);
TArray<int> dominantIndexArray({0, 3, 6, 9});
for (int j = 0; j < 4; j++)
{
    int index1 = j * 3;
    double evecReal1 = abs(eigenvector[index1].real());
    double evecReal2 = abs(eigenvector[index1 + 1].real());
    double evecReal3 = abs(eigenvector[index1 + 2].real());
    if ((evecReal1 > evecReal2) && (evecReal1 > evecReal3))
    {
        dominantIndexArray[j] = index1;
    }
    else if ((evecReal2 > evecReal1) && (evecReal2 > evecReal3))
    {
        dominantIndexArray[j] = index1 + 1;
    }
    else
    {
        dominantIndexArray[j] = index1 + 2;
    }
}
const char* stateNameArray[12] = {"u", "v", "w", "p", "q", "r", "z", "y", "g", "$\phi$", "$\theta$", "$\psi$"};
modeOutputFile << "\nDominant eigenvector states = ";
for (int j = 0; j < 4; j++)
{  
  modeOutputFile << stateNameArray[dominantIndexArray[j]] << ",";
}

modeOutputFile << \n"Eigenvector = \n" << eigenvector;

double sigma = -eigenvalue.real();
modeOutputFile << \n"Damping rate, sigma [1/s] = " << sigma;

if (eigenvalue.imag() != 0.0)
{
  // Oscillatory properties for oscillatory modes (eigenvalue = complex conjugate)
  double omega_d = abs(eigenvalue.imag());
  modeOutputFile << \n"Damped natural frequency, omega_d [rad/s] = " << omega_d;
  double period = 2. * Pi / omega_d;
  modeOutputFile << \n"Damped period [s] = " << period;

  std::complex<double> lambda1 = eigenvalue;
  std::complex<double> lambda2 = std::complex<double>(eigenvalue.real(), -1.*eigenvalue.imag());
  double omega_n = std::real(sqrt(lambda1 * lambda2));
  modeOutputFile << \n"Undamped natural frequency, omega_n [rad/s] = " << omega_n;

  double zeta = std::real(-0.5 * (lambda1 + lambda2) / (sqrt(lambda1 * lambda2)));
  modeOutputFile << \n"Damping ratio, zeta = " << zeta;
}

if (eigenvalue.real() > 0.)
{
  // Doubling time for divergent modes
  double doublingTime = -log(2.) / sigma;
  modeOutputFile << \n"Doubling time [s] = " << doublingTime;
}

if (eigenvalue.real() < 0.)
{
  // 99% damping time for convergent modes
  double dampingTime99 = -log(0.01) / sigma;
  modeOutputFile << \n"99% Damping Time [s] = " << dampingTime99;
}

modeOutputFile.close();
UE_LOG(LogTemp, Warning, TEXT("Stability output finished."));
}

Eigen::EigenSolver<Eigen::Matrix<double, 12, 12>> AAircraft::SolveStabilityEigensystem()
{
  // Precompute repeated values
  // Sine, Cosine, Tangent of trim phi and theta

  // Precompute values
  // ...
TArray<double> EulerAngles0 = QuatToEuler(TArray<double>( AircraftStates(9), AircraftStates(10), AircraftStates(11), AircraftStates(12 )));

UE_LOG(LogTemp, Warning, TEXT(" Euler angles (phi, theta, psi) = %f\t%f\t%f"), float(EulerAngles0[0]), float(EulerAngles0[1]), float(EulerAngles0[2]));

double phi0 = EulerAngles0[0];
double Sphi = sin(phi0);
double Cphi = cos(phi0);
double Tphi = Sphi / Cphi;
double theta0 = EulerAngles0[1];
double Sttheta = sin(theta0);
double Ctheta = cos(theta0);
double Ttheta = Sttheta / Ctheta;

// Force and moment derivatives

// dForce / dVelocity

double Fxu = Fxyz_uvw(0, 0);
double Fxv = Fxyz_uvw(0, 1);
double Fxw = Fxyz_uvw(0, 2);
double Fyu = Fxyz_uvw(1, 0);
double Fyv = Fxyz_uvw(1, 1);
double Fyw = Fxyz_uvw(1, 2);
double Fzu = Fxyz_uvw(2, 0);
double Fzv = Fxyz_uvw(2, 1);
double Fzw = Fxyz_uvw(2, 2);

UE_LOG(LogTemp, Warning, TEXT("F, uvw = %f\t%f\t%f"), float(Fxu), float(Fxv), float(Fxw));
UE_LOG(LogTemp, Warning, TEXT(" %f\t%f\t%f"), float(Fyu), float(Fyv), float(Fyw));
UE_LOG(LogTemp, Warning, TEXT(" %f\t%f\t%f"), float(Fzu), float(Fzv), float(Fzw));

// dMoment / dVelocity

double Mxu = Mxyz_uvw(0, 0);
double Mxv = Mxyz_uvw(0, 1);
double Mxw = Mxyz_uvw(0, 2);
double Myu = Mxyz_uvw(1, 0);
double Myv = Mxyz_uvw(1, 1);
double Myw = Mxyz_uvw(1, 2);
double Mzu = Mxyz_uvw(2, 0);
double Mzv = Mxyz_uvw(2, 1);
double Mzw = Mxyz_uvw(2, 2);

UE_LOG(LogTemp, Warning, TEXT("M, uvw = %f\t%f\t%f"), float(Mxu), float(Mxv), float(Mxw));
UE_LOG(LogTemp, Warning, TEXT(" %f\t%f\t%f"), float(Myu), float(Myv), float(Myw));
UE_LOG(LogTemp, Warning, TEXT(" %f\t%f\t%f"), float(Mzu), float(Mzv), float(Mzw));

// dForce / dRotationRate

double Fxp = Fxyz_pqr(0, 0);
double Fxq = Fxyz_pqr(0, 1);
double Fxr = Fxyz_pqr(0, 2);
double Fyp = Fxyz_pqr(1, 0);
double Fyq = Fxyz_pqr(1, 1);
double Fyr = Fxyz_pqr(1, 2);
double Fzp = Fxyz_pqr(2, 0);
double Fzq = Fxyz_pqr(2, 1);
double Fzr = Fxyz_pqr(2, 2);

UE_LOG(LogTemp, Warning, TEXT("F, pqr = %f\t%f\t%f"), float(Fxp), float(Fxq), float(Fxr));
UE_LOG(LogTemp, Warning, TEXT(" %f\t%f\t%f"), float(Fyp), float(Fyq), float(Fyr));
UE_LOG(LogTemp, Warning, TEXT(" %f\t%f\t%f"), float(Fzp), float(Fzq), float(Fzr));

// dMoment / dRotationRate

double Map = Mxyz_pqr(0, 0);
double Mapq = Mxyz_pqr(0, 1);

double Mxp = Mxyz_pqr(0, 1, 0);
double Myq = Mxyz_pqr(1, 1, 0);
double Myr = Mxyz_pqr(1, 2, 0);
double Mzp = Mxyz_pqr(2, 0, 0);
double Mzq = Mxyz_pqr(2, 1, 0);
double Mzr = Mxyz_pqr(2, 2, 0);
UE_LOG(LogTemp, Warning, TEXT(" \%f \%f \%f"), float(Mxp), float(Myq), float(Myr));
UE_LOG(LogTemp, Warning, TEXT(" \%f \%f \%f"), float(Myp), float(Myq), float(Myr));
UE_LOG(LogTemp, Warning, TEXT(" \%f \%f \%f"), float(Mzp), float(Mzq), float(Mzr));

// Initial velocity

double V0 = VInit;
double WV0 = W / V0;
double gV0 = gInit / V0;

// Moment rate derivative components

double AMxp = Mxp + gV0*(Ixz*Tphi*Sphi*Ctheta - Ixy*Sphi*Ctheta);
double AMxq = Mxq - hz + gV0*((Iyy-Izz)*Sphi*Ctheta + 2.*Iyz*Tphi*Sphi*Ctheta - Ixz*Tphi*Stheta);
double AMxr = Mxr + hy + gV0*((Iyy-Izz)*Tphi*Sphi*Ctheta - 2.*Iyz*Sphi*Ctheta + Ixy*Tphi*Stheta);
double AMyp = Myp + hz + gV0*((Izz-Ixx)*Sphi*Ctheta + 2.*Ixz*Tphi*Stheta - Iyz*Tphi*Sphi*Ctheta);
double AMyq = Myq + gV0*(Ixy*Sphi*Ctheta + Iyz*Tphi*Stheta);
double AMyr = Myr - hx + gV0*(-(Izz-Ixx)*Tphi*Stheta + 2.*Ixz*Sphi*Ctheta - Ixy*Tphi*Sphi*Ctheta);
double AMzp = Mzp - hy + gV0*(-(Ixx-Iyy)*Tphi*Stheta + Iyz*Sphi*Ctheta);
double AMzq = Mzq + hx + gV0*(-(Ixx-Iyy)*Tphi*Stheta - 2.*Ixy*Sphi*Ctheta + Ixz*Sphi*Ctheta);
double AMzr = Mzr + gV0*(-Iyz*Tphi*Stheta - Ixz*Sphi*Ctheta);

// Equilibrium turning rate, Omega

double psi0 = 0.;
Omega = gV0*tan(psi0);
double t = 0;
OmegaT = Omega * t;
SOt = sin(OmegaT);
COt = cos(OmegaT);

// Populate A and B matrices using linearized dimensional coupled eigensystem

Eigen::Matrix<double, 12, 12> AMatrix;

UE_LOG(LogTemp, Warning, TEXT("\nA:"));
for (int i = 0; i < 12; i++)
for (int j = 0; j < 12; j++)
UE_LOG(LogTemp, Warning, TEXT("%f	%f	%f	%f	%f	%f	%f	%f	%f	%f	%f"), float(AMatrix(i, 0)), float(AMatrix(i, 1)), float(AMatrix(i, 2)), float(AMatrix(i, 3)), float(AMatrix(i, 4)), float(AMatrix(i, 5)), float(AMatrix(i, 6)), float(AMatrix(i, 7)), float(AMatrix(i, 8)), float(AMatrix(i, 9)), float(AMatrix(i, 10)), float(AMatrix(i, 11)));}

Eigen::Matrix<double, 12, 12> BMatrix{
{Wg, 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.},
{0., Wg, 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.},
{0., 0., (Wg - Fz_wdot), 0., 0., 0., 0., 0., 0., 0., 0., 0.},
{0., 0., 0., Ixx, -Ixy, -Ixz, 0., 0., 0., 0., 0., 0.},
{0., 0., -My_wdot, -Ixy, Iyy, -Iyz, 0. , 0., 0., 0., 0., 0.},
{0., 0., 0., -Ixz, -Iyz, Izz, 0. , 0. , 0., 0., 0., 0.},
{0. , 0. , 0. , 0. , 0. , 0. , 1, 0. , 0. , 0. , 0. , 0.},
{0. , 0. , 0. , 0. , 0. , 0. , 0. , 1. , 0. , 0. , 0. , 0.},
{0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 1. , 0. , 0. , 0.},
{0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 1. , 0. , 0.},
{0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 1. , 0.},
{0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 1.}};

UE_LOG(LogTemp, Warning, TEXT("B:"));
for (int i = 0; i < 12; i++) {
UE_LOG(LogTemp, Warning, TEXT("%f	%f	%f	%f	%f	%f	%f	%f	%f	%f	%f"), float(BMatrix(i, 0)), float(BMatrix(i, 1)), float(BMatrix(i, 2)), float(BMatrix(i, 3)), float(BMatrix(i, 4)), float(BMatrix(i, 5)), float(BMatrix(i, 6)), float(BMatrix(i, 7)), float(BMatrix(i, 8)), float(BMatrix(i, 9)), float(BMatrix(i, 10)), float(BMatrix(i, 11)));}

// Solve Bx = A for x = B^(-1)A = BinvAMatrix
Eigen::Matrix<double, 12, 12> BinvAMatrix = BMatrix.colPivHouseholderQr().solve(AMatrix);

// Save the EigenSolver class that contains the eigenvalues and eigenvectors of BinvAMatrix
Eigen::EigenSolver<Eigen::Matrix<double, 12, 12>> eigensystemSolution(BinvAMatrix);

return eigensystemSolution;

void AAircraft::CalculateForceMomentDerivatives(TArray<double> windDeltas)
{
// Create array of deltaState's to be added to the equilibrium state in the central difference approximation loop below
// Create vector for each wind delta
Eigen::Vector3<double> windDu({ windDeltas[0], 0., 0. });
Eigen::Vector3<double> windDv({ 0., windDeltas[1], 0. });
Eigen::Vector3<double> windDw({ 0., 0., windDeltas[2] });
Eigen::Vector3<double> windDp({ windDeltas[3], 0., 0. });
Eigen::Vector3<double> windDq({ 0., windDeltas[4], 0. });
Eigen::Vector3<double> windDr({ 0., 0., windDeltas[5] });
// Convert each wind delta to the body frame
Eigen::Vector3<double> bodyDu = WindToBodyFrame(windDu);
Eigen::Vector3<double> bodyDv = WindToBodyFrame(windDv);
Eigen::Vector3<double> bodyDw = WindToBodyFrame(windDw);
Eigen::Vector3<double> bodyDp = WindToBodyFrame(windDp);
Eigen::Vector3<double> bodyDq = WindToBodyFrame(windDq);
Eigen::Vector3<double> bodyDr = WindToBodyFrame(windDr);

// Populate matrix with deltaState rows based on body deltas
Eigen::Matrix<double, 6, 13> deltaStates({
  {bodyDu(0), bodyDu(1), bodyDu(2), 0., 0., 0.,
   0., 0., 0., 0., 0., 0., 0., 0.},
  {bodyDv(0), bodyDv(1), bodyDv(2), 0., 0., 0.,
   0., 0., 0., 0., 0., 0., 0., 0.},
  {bodyDw(0), bodyDw(1), bodyDw(2), 0., 0., 0.,
   0., 0., 0., 0., 0., 0., 0., 0.},
  {0., 0., 0., bodyDp(0), bodyDp(1), bodyDp(2),
   0., 0., 0., 0., 0., 0., 0., 0.},
  {0., 0., 0., bodyDq(0), bodyDq(1), bodyDq(2),
   0., 0., 0., 0., 0., 0., 0., 0.},
  {0., 0., 0., bodyDr(0), bodyDr(1), bodyDr(2),
   0., 0., 0., 0., 0., 0., 0., 0.}
});

// Save equilibrium trimmed states
Eigen::Matrix<double, 13, 1> equilibriumStates = AircraftStates;

// Loop through perturbations
// NOTE: i = (0, 1, 2, 3, 4, 5) => (du, dv, dw, dp, dq, dr) in wind frame
for (int i = 0; i < 6; i++)
{
  LogStates(equilibriumStates, FString(TEXT("--- Aircraft States")));
  Eigen::Matrix<double, 1, 13> deltaState = deltaStates.row(i);
  LogStates(deltaState, FString(TEXT("deltaState")));

  // Calculate forces and moments at each perturbation
  Eigen::Matrix<double, 13, 1> positivePerturbedStates = equilibriumStates;
  positivePerturbedStates += deltaState;
  LogStates(positivePerturbedStates, FString(TEXT("states_i+1")));
  TArray<double> perturbedAeroForces_iPlus1 = CalculateAeroForces(positivePerturbedStates, AircraftControls);
  positivePerturbedStates += deltaState;
  LogStates(positivePerturbedStates, FString(TEXT("states_i+2")));
  TArray<double> perturbedAeroForces_iPlus2 = CalculateAeroForces(positivePerturbedStates, AircraftControls);

  Eigen::Matrix<double, 13, 1> negativePerturbedStates = equilibriumStates;
  negativePerturbedStates -= deltaState;
  LogStates(negativePerturbedStates, FString(TEXT("states_i-1")));
  TArray<double> perturbedAeroForces_iMinus1 = CalculateAeroForces(negativePerturbedStates, AircraftControls);
  negativePerturbedStates -= deltaState;
  LogStates(negativePerturbedStates, FString(TEXT("states_i-2")));
  TArray<double> perturbedAeroForces_iMinus2 = CalculateAeroForces(negativePerturbedStates, AircraftControls);

  // Convert force TArray into eigen arrays so we can approximate derivatives of each force and moment simultaneously using coefficient-wise operations
  Eigen::Array<double, 6, 1, 1> forceMatrix_iPlus1 {
    perturbedAeroForces_iPlus1[0], perturbedAeroForces_iPlus1[1], perturbedAeroForces_iPlus1[2],
  }
perturbedAeroForces_iPlus1[3], perturbedAeroForces_iPlus1[4], perturbedAeroForces_iPlus1[5];

Eigen::Array<double, 6, 1> forceMatrix_iPlus2 {
perturbedAeroForces_iPlus2[0], perturbedAeroForces_iPlus2[1], perturbedAeroForces_iPlus2[2],
perturbedAeroForces_iPlus2[3], perturbedAeroForces_iPlus2[4], perturbedAeroForces_iPlus2[5];
};

Eigen::Array<double, 6, 1> forceMatrix_iMinus1 {
perturbedAeroForces_iMinus1[0], perturbedAeroForces_iMinus1[1], perturbedAeroForces_iMinus1[2],
perturbedAeroForces_iMinus1[3], perturbedAeroForces_iMinus1[4], perturbedAeroForces_iMinus1[5];
};

Eigen::Array<double, 6, 1> forceMatrix_iMinus2 {
perturbedAeroForces_iMinus2[0], perturbedAeroForces_iMinus2[1], perturbedAeroForces_iMinus2[2],
perturbedAeroForces_iMinus2[3], perturbedAeroForces_iMinus2[4], perturbedAeroForces_iMinus2[5];
};

UE_LOG(LogTemp, Warning, TEXT("h = %f"), windDeltas[i]);

UE_LOG(LogTemp, Warning, TEXT(" forceMatrix_iPlus1 = {%f	 %f	 %f	 %f	 %f	 %f}"),
forceMatrix_iPlus1(4), forceMatrix_iPlus1(5));

UE_LOG(LogTemp, Warning, TEXT(" forceMatrix_iPlus2 = {%f	 %f	 %f	 %f	 %f	 %f}"),
forceMatrix_iPlus2(4), forceMatrix_iPlus2(5));

UE_LOG(LogTemp, Warning, TEXT(" forceMatrix_iMinus1 = {%f	 %f	 %f	 %f	 %f	 %f}"),
forceMatrix_iMinus1(0), forceMatrix_iMinus1(1), forceMatrix_iMinus1(2), forceMatrix_iMinus1(3),
forceMatrix_iMinus1(4), forceMatrix_iMinus1(5));

UE_LOG(LogTemp, Warning, TEXT(" forceMatrix_iMinus2 = {%f	 %f	 %f	 %f	 %f	 %f}"),
forceMatrix_iMinus2(0), forceMatrix_iMinus2(1), forceMatrix_iMinus2(2), forceMatrix_iMinus2(3),
forceMatrix_iMinus2(4), forceMatrix_iMinus2(5));

Eigen::Array<double, 6, 1> forceDerivativeMatrix = (-forceMatrix_iPlus2 + 8 * forceMatrix_iPlus1 - 8 * forceMatrix_iMinus1 + forceMatrix_iMinus2) * (1.0 / (12.0 * windDeltas[i]));

UE_LOG(LogTemp, Warning, TEXT("forceDerivativeMatrix = {%f	 %f	 %f	 %f	 %f	 %f}"),
forceDerivativeMatrix(0), forceDerivativeMatrix(1), forceDerivativeMatrix(2),
forceDerivativeMatrix(3), forceDerivativeMatrix(4), forceDerivativeMatrix(5));

// Approximate derivatives using central difference with 2nd order taylor series term (error of O(h^4))

UELOG(LogTemp, Warning, TEXT(" NOTE: Higher order central difference equation: f'(x_i) = (-f(x_{i+2}) + 8*f(x_{i+1}) - 8*f(x_{i-1}) + f(x_{i-2})) / (12*h)"));

// Save forces and moments in derivative member variables
if (i < 3)
{
  // For i < 3, deltas = du, dv, dw
  Fxyz_uvw(0, i) = forceDerivativeMatrix(0);
  Fxyz_uvw(1, i) = forceDerivativeMatrix(1);
  Fxyz_uvw(2, i) = forceDerivativeMatrix(2);
  Mxyz_uvw(0, i) = forceDerivativeMatrix(3);
  Mxyz_uvw(1, i) = forceDerivativeMatrix(4);
}
Mxyz_uvw(2, i) = forceDerivativeMatrix(5);

else {

    UE_LOG(LogTemp, Warning, TEXT("Count for 2nd loop."));

    // For 3 <= i < 6, deltas = dp, dq, dr and 0 <= j < 3
    int j = i - 3;
    Fxyz_pqr(0, j) = forceDerivativeMatrix(0);
    Fxyz_pqr(1, j) = forceDerivativeMatrix(1);
    Fxyz_pqr(2, j) = forceDerivativeMatrix(2);
    Mxyz_pqr(0, j) = forceDerivativeMatrix(3);
    Mxyz_pqr(1, j) = forceDerivativeMatrix(4);
    Mxyz_pqr(2, j) = forceDerivativeMatrix(5);
}

// Approximate Fz_wdot and My_wdot using 7.6.66 (Phillips p. 785)
   double rho = CalculateStdAtmProperties_English(- AircraftStates(8)) [3];
   double Sh = 63.675;
   double xbst = -7.358;
   double xbbh = -13.13;
   double lwt = 1.1 * (xbst - xbbh);
   double CLaw = 3.3775691217788646;
   double CLah = 1.3657050471586294;
   if (bIsBireAircraft) {

        // Override F16 CLah with BIRE CLah
        double bireAngle = AircraftControls(2);
        CLah = 1.3858047943592773 * abs(cos(bireAngle));
    }

   Fz_wdot = -(rho * Sw * Sh * lwt * CLaw * CLah) / (PI * bw * bw);
   My_wdot = (rho * Sw * Sh * lwt * xbbh * CLaw * CLah) / (PI * bw * bw);
}