

12-2019

Autonomous Autorotation of a Tile-Rotor Aircraft Using Model Predictive Control

Elias Wilson

Follow this and additional works at: <https://commons.erau.edu/edt>

 Part of the [Aerospace Engineering Commons](#)

This Thesis - Open Access is brought to you for free and open access by Scholarly Commons. It has been accepted for inclusion in Dissertations and Theses by an authorized administrator of Scholarly Commons. For more information, please contact commons@erau.edu.

AUTONOMOUS AUTOROTATION OF A TILT-ROTOR AIRCRAFT USING
MODEL PREDICTIVE CONTROL

A Thesis

Submitted to the Faculty of
Embry-Riddle Aeronautical University

by

Elias Wilson

In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Aerospace Engineering

December 2019
Embry-Riddle Aeronautical University
Daytona Beach, Florida

AUTONOMOUS AUTOROTATION OF A TILT-ROTOR AIRCRAFT USING
MODEL PREDICTIVE CONTROL

by

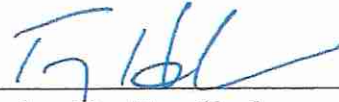
Elias Wilson

This Thesis was prepared under the direction of the candidate's Committee Chair, Dr. Richard Prazenica, Department of Aerospace Engineering, and has been approved by the members of the Thesis Committee. It was submitted to the Office of the Senior Vice President for Academic Affairs and Provost, and was accepted in partial fulfillment of the requirements for the Degree of Master of Science in Aerospace Engineering.

THESIS COMMITTEE



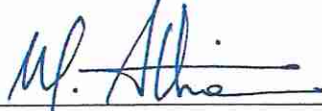
Chairman, Dr. Richard Prazenica



Member, Dr. Troy Henderson



Member, Dr. Morad Nazari



Graduate Program Coordinator,
Dr. Magdy Attia

12/3/2019

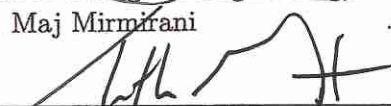
Date



Dean of College of Engineering,
Dr. Maj Mirmirani

12/3/19

Date



Associate Provost of Academic Support,
Dr. Christopher Grant

12/3/19

Date

ACKNOWLEDGEMENTS

The Aerospace Engineering department at Embry-Riddle has changed my life. Starting here, I learned the basics of physics and mathematics, which largely lead to my success in upper level engineering course. As I approached my bachelors degree, I realized academic work was what I wanted to do. Professors like Dr. Ricklick and Dr. Anderson showed me interesting and different sides of engineering that led to where I am now.

I need to thank my advisor Dr. Prazenica for letting me roam free and really figure out how to learn on my own. Over the past year of work I have grown more than the first 17 years of school and that is due to his inspiring persona. Whenever there was doubt and I needed help, he was always there to assist with strong academic reasoning and a load of sarcasm. I am looking forward to the following years as I continue my education.

The other committee members, Dr. Henderson and Dr. Nazari, have not only helped make my document stronger but also have helped make me a better student. After undergraduate physics courses, I felt invincible in the world of kinematics and kinetics. Little did I know that Dr. Henderson would open my eyes to a whole world of advanced dynamics and confusion. Dr. Nazari's curiosity and genuine desire of knowledge has inspired me to strive for a deep understanding of topics rather than a "good enough" overview. Where some may suffice with giving a quick judgment, Dr. Nazari will take the time to analyze and think about a problem, no matter how small or irrelevant.

My close acquaintances Ben, Nick, and Aryslan have pushed me to become a better a student as well. Their work, dedication, and sheer brain power is an absolute testament to student success. From the common gossip about poorly designed homework problems with Ben to the long studies- during work- about why people procrastinate with Nick, I always had something to learn. Now, working

next to Aryslan, I always have someone to talk to about the cool abstract ideas I read about and I have a great source of things that I can barely understand.

TABLE OF CONTENTS

	Page
LIST OF CONTENTS	iii
LIST OF FIGURES	v
ABBREVIATIONS	vi
SYMBOLS	vii
ABSTRACT	12
1. Introduction	13
1.1 XV-15 Overview	14
1.2 Autorotation and Landing	15
1.3 Objectives	19
2. Current Technology	21
3. Tilt Rotor Simulation	26
3.1 Vehicle Axes Systems	26
3.2 Rigid Body Dynamics	27
3.2.1 Rigid Body Motion	27
3.2.2 Hub Motion	31
3.3 Rotor System	32
3.3.1 Blade Dynamics	33
3.3.2 Blade Motion	33
3.3.3 Blade Aerodynamics	35
3.3.4 Rotor Forces and Moments	38
3.3.5 Flapping Equation	40
3.3.6 Azimuth Dynamics	41
3.3.7 Inflow Model	41
3.4 Control Blending	45
4. Controller Design	47
4.1 Optimal Control Techniques	47
4.2 Reduced Order Model	48
4.2.1 Rotor System	48
4.2.1.1 Multi-Blade Coordinates	48
4.2.1.2 Multi-Blade Flapping Equations	49
4.2.1.3 Wind Axes Transformations	56
4.2.1.4 Forces and Moments	57
4.2.1.5 Inflow	59
4.3 Optimization	60

4.3.1	Cost Function	60
4.3.2	Prediction	62
4.3.3	Constraints	63
4.3.4	Recurrent Neural Network	66
4.3.5	Optimization Stability	67
5.	Model Comparison	70
5.1	Collective Shift	70
5.2	Control Inputs with Constant Rotor Speed	71
5.2.1	Longitudinal Doublet	71
5.2.2	Lateral Doublet	73
5.2.3	Rudder Doublet	75
5.2.4	Collective Doublet	77
5.3	Variable Rotor Speed	79
5.3.1	Uncontrolled	79
5.3.2	Longitudinal Doublet	81
5.3.3	Lateral Doublet	83
5.3.4	Rudder Doublet	85
5.3.5	Collective Doublet	87
6.	Controller Simulation	90
6.1	Forward Flight Constant Rotor Speed	90
6.2	Autorotation Entrance	93
6.3	Rotor Speed Constraints	96
7.	Conclusions and Future Work	100
	REFERENCES	103
A.	APPENDICES	107
A.1	Reduced Order Model Jacobian	107
A.1.1	Dynamics Function	107
A.1.2	Hub Velocities and Sideslip	111
A.1.3	Wind axis transformations	113
A.1.4	Thrust Coefficient	113
A.1.5	Torque Coefficient	115
A.1.6	Inflow	116
A.1.7	Flapping - Hub Velocities	119
A.1.8	Flapping - Rotor Speed	122
A.1.9	Hub Forces and Moments	128
A.1.10	Dimensional Derivatives - Rotor Speed	129
A.1.11	Controls	130
A.2	Positive Semidefiniteness of A^TQA	130

LIST OF FIGURES

Figure	Page
1.1 Bell XV-15 experimental tilt rotor aircraft (Curry, 2002)	15
1.2 Blade aerodynamics during autorotation (Leishman, 2006).	16
1.3 Axial flight power requirements normalized by hovering conditions (Leishman, 2006).	17
1.4 Generic forward flight power requirements for 13000lb aircraft with 2 - 25ft rotors and a flat plate drag area of $9ft^2$ (Leishman, 2006).	18
1.5 Power out maneuvers for generic helicopter (Koyovis, 2017).	19
3.1 3-2-1 Rotation from <i>Earth</i> frame to <i>body</i> frame (Pamadi, 2004).	30
3.2 (a) shows the sectioning of a blade for blade element theory and (b) shows the aerodynamics of that blade section (Leishman, 2000).	35
5.1 Thrust produced by a single rotor in the high fidelity simulation (SIM) and reduced order model (ROM) for varying collective inputs.	71
5.2 Longitudinal states comparison of reduced order model (ROM) and high fidelity simulation (SIM) given a 5° longitudinal doublet and an initial x_b velocity of $10\frac{ft}{s}$	72
5.3 Flapping states comparison of reduced order model (ROM) and high fidelity simulation (SIM) given a 5° longitudinal doublet and an initial x_b velocity of $10\frac{ft}{s}$	73
5.4 Longitudinal states comparison of reduced order model (ROM) and high fidelity simulation (SIM) given a 5° lateral doublet and an initial x_b velocity of $10\frac{ft}{s}$	74
5.5 Lateral states comparison of reduced order model (ROM) and high fidelity simulation (SIM) given a 5° lateral doublet and an initial x_b velocity of $10\frac{ft}{s}$	74
5.6 Flapping states comparison of reduced order model (ROM) and high fidelity simulation (SIM) given a 5° lateral doublet and an initial x_b velocity of $10\frac{ft}{s}$	75
5.7 Longitudinal states comparison of reduced order model (ROM) and high fidelity simulation (SIM) given a 5° rudder doublet and an initial v_b velocity of $10\frac{ft}{s}$	76

Figure	Page
5.8 Lateral states comparison of reduced order model (ROM) and high fidelity simulation (SIM) given a 5° rudder doublet and an initial v_b velocity of $10\frac{ft}{s}$	76
5.9 Flapping states comparison of reduced order model (ROM) and high fidelity simulation (SIM) given a 5° rudder doublet and an initial v_b velocity of $10\frac{ft}{s}$	77
5.10 Longitudinal states comparison of reduced order model (ROM) and high fidelity simulation (SIM) given a 5° collective doublet and an initial u_b and w_b velocity of $10\frac{ft}{s}$	78
5.11 Flapping states comparison of reduced order model (ROM) and high fidelity simulation (SIM) given a 5° collective doublet and an initial u_b and w_b velocity of $10\frac{ft}{s}$	78
5.12 Rotor speed comparison of reduced order model (ROM) and high fidelity simulation (SIM) given an initial u_b and w_b velocity of $80\frac{ft}{s}$ and $40\frac{ft}{s}$, respectively.	79
5.13 Longitudinal states comparison of reduced order model (ROM) and high fidelity simulation (SIM) with variable rotor speed given an initial u_b and w_b velocity of $80\frac{ft}{s}$ and $40\frac{ft}{s}$, respectively.	80
5.14 Flapping states comparison of reduced order model (ROM) and high fidelity simulation (SIM) with variable rotor speed given an initial u_b and w_b velocity of $80\frac{ft}{s}$ and $40\frac{ft}{s}$, respectively.	80
5.15 Rotor speed comparison of reduced order model (ROM) and high fidelity simulation (SIM) given a 5° longitudinal doublet and an initial u_b and w_b velocity of $80\frac{ft}{s}$ and $40\frac{ft}{s}$, respectively.	81
5.16 Longitudinal states comparison of reduced order model (ROM) and high fidelity simulation (SIM) with variable rotor speed given a 5° longitudinal doublet and an initial u_b and w_b velocity of $80\frac{ft}{s}$ and $40\frac{ft}{s}$, respectively.	82
5.17 Flapping states comparison of reduced order model (ROM) and high fidelity simulation (SIM) with variable rotor speed given a 5° longitudinal doublet and an initial u_b and w_b velocity of $80\frac{ft}{s}$ and $40\frac{ft}{s}$, respectively.	82

Figure	Page
5.18 Longitudinal states comparison of reduced order model (ROM) and high fidelity simulation (SIM) with variable rotor speed given a 5° lateral doublet and an initial u_b and w_b velocity of $80\frac{ft}{s}$ and $40\frac{ft}{s}$, respectively. .	83
5.19 Lateral states comparison of reduced order model (ROM) and high fidelity simulation (SIM) with variable rotor speed given a 5° lateral doublet and an initial u_b and w_b velocity of $80\frac{ft}{s}$ and $40\frac{ft}{s}$, respectively. .	84
5.20 Flapping states comparison of reduced order model (ROM) and high fidelity simulation (SIM) with variable rotor speed given a 5° lateral doublet and an initial u_b and w_b velocity of $80\frac{ft}{s}$ and $40\frac{ft}{s}$, respectively. .	84
5.21 Rotor speed comparison of reduced order model (ROM) and high fidelity simulation (SIM) given a 5° lateral doublet and an initial u_b and w_b velocity of $80\frac{ft}{s}$ and $40\frac{ft}{s}$, respectively.	85
5.22 Rotor speed comparison of reduced order model (ROM) and high fidelity simulation (SIM) given a 5° rudder doublet and an initial u_b and w_b velocity of $80\frac{ft}{s}$ and $40\frac{ft}{s}$, respectively.	85
5.23 Longitudinal states comparison of reduced order model (ROM) and high fidelity simulation (SIM) with variable rotor speed given a 5° rudder doublet and an initial u_b and w_b velocity of $80\frac{ft}{s}$ and $40\frac{ft}{s}$, respectively. .	86
5.24 Lateral states comparison of reduced order model (ROM) and high fidelity simulation (SIM) with variable rotor speed given a 5° rudder doublet and an initial u_b and w_b velocity of $80\frac{ft}{s}$ and $40\frac{ft}{s}$, respectively. .	86
5.25 Flapping states comparison of reduced order model (ROM) and high fidelity simulation (SIM) with variable rotor speed given a 5° rudder doublet and an initial u_b and w_b velocity of $80\frac{ft}{s}$ and $40\frac{ft}{s}$, respectively. .	87
5.26 Longitudinal states comparison of reduced order model (ROM) and high fidelity simulation (SIM) with variable rotor speed given a 5° collective doublet and an initial u_b and w_b velocity of $80\frac{ft}{s}$ and $40\frac{ft}{s}$, respectively. .	88
5.27 Flapping states comparison of reduced order model (ROM) and high fidelity simulation (SIM) with variable rotor speed given a 5° collective doublet and an initial u_b and w_b velocity of $80\frac{ft}{s}$ and $40\frac{ft}{s}$, respectively. .	88
5.28 Rotor speed comparison of reduced order model (ROM) and high fidelity simulation (SIM) given a 5° collective doublet and an initial u_b and w_b velocity of $80\frac{ft}{s}$ and $40\frac{ft}{s}$, respectively.	89

Figure	Page
6.1 Controlled longitudinal states with constant rotor speed in forward flight.	91
6.2 Controlled lateral states with constant rotor speed in forward flight. . . .	92
6.3 Controller inputs with constant rotor speed in forward flight.	92
6.4 Controlled longitudinal states with constant rotor speed in forward flight and initial conditions away from the reference.	93
6.5 Controller inputs with constant rotor speed in forward flight and initial conditions away from the reference.	93
6.6 Controlled longitudinal states of a longitudinally restricted simulation and a full spatial 6-DOF simulation.	94
6.7 Controlled lateral states of a longitudinally restricted simulation and a full spatial 6-DOF simulation.	95
6.8 Rotor speed of a longitudinally restricted simulation and a full spatial 6-DOF simulation.	95
6.9 Controller inputs of a longitudinally restricted simulation and a full spatial 6-DOF simulation.	96
6.10 Longitudinal states of the vehicle starting near the low rotor speed constraint boundary.	97
6.11 Rotor speed starting near the low rotor speed constraint boundary. . . .	97
6.12 Control inputs starting near the low rotor speed constraint boundary. . .	98
6.13 Longitudinal states of vehicle starting near the upper rotor speed constraint boundary.	98
6.14 Rotor speed starting near the upper boundary.	99
6.15 Control inputs near the upper rotor speed constraint boundary.	99

ABBREVIATIONS

<i>AEI</i>	All engines inoperative
<i>AR</i>	Autorotate
<i>BEMT</i>	Blade element momentum theory
<i>BET</i>	Blade element theory
<i>BLC</i>	Blade loading coefficient
<i>CG</i>	Center of gravity
<i>DDP</i>	Differential dynamic programming
<i>DOF</i>	Degree of freedom
<i>FAA</i>	Federal Aviation Administration
<i>GD</i>	Gradient descent
<i>H – V</i>	Height-velocity
<i>KKT</i>	Karush-Khun-Tucker
<i>LQR</i>	Linear quadratic regulator
<i>MPC</i>	Model predictive control
<i>NLDI</i>	Nonlinear dynamic inversion
<i>ODE</i>	Ordinary differential equation
<i>OEI</i>	One engines inoperative
<i>PD</i>	Positive definite
<i>PSD</i>	Positive semidefinite
<i>RNN</i>	Recurrent neural network
<i>ROM</i>	Reduced order model
<i>TR</i>	Tilt Rotor
<i>VTOL</i>	Vertical take off and landing

SYMBOLS

α	Angle of attack
$\bar{\lambda}_0, \bar{\lambda}_s, \bar{\lambda}_c$	Static normalized uniform, longitudinal, and lateral inflows
$\bar{\lambda}_m$	Static normalized total uniform inflow
\bar{c}	Average blade chord
\bar{C}_d	Average blade parasitic drag coefficient
\bar{r}_b	Blade center of mass
β	Blade flapping
$\beta_0, \beta_c, \beta_s$	Coning, longitudinal flapping, and lateral flapping
χ	Wake skew angle
\ddot{x}	Second time derivative of x
$\Delta\sigma$	Change in the Lagrangian multipliers
Δd	Local drag component
Δl	Local lift component
Δr	Width of blade section
ΔU	Change in the control matrix
$\dot{\omega}_{yc}$	Angular acceleration flapping contribution
\dot{x}	Time derivative of x
γ	Lock number
$\hat{\theta}_{0l}, \hat{\theta}_{cl}, \hat{\theta}_{sl}$	Collective, longitudinal, and lateral inputs to left rotor
$\hat{\theta}_{0r}, \hat{\theta}_{cr}, \hat{\theta}_{sr}$	Collective, longitudinal, and lateral inputs to right rotor
$\hat{\theta}_0, \hat{\theta}_c, \hat{\theta}_s$	Collective, longitudinal, and lateral inputs to rotor
$\hat{\theta}_{tw}$	Blade pitch due to twist
\hat{I}_{xz}	Auxiliary term for finding \dot{p}
\hat{L}	Inflow dynamic gains matrix in <i>hub</i> axis
$\lambda_0, \lambda_c, \lambda_s$	Normalized inflow components
λ_m	Total resultant normal normalized inflow
\mathcal{L}	Optimization Lagrangian
μ_w	Normalized velocity in wind axis

μ_h, ν_h, ξ_h	Normalized hub velocities
ν_0, ν_c, ν_s	Dimensional normal, longitudinal, and lateral inflow components
Ω	Rotor speed
Ω_0	Nominal rotor speed
$\omega_{xc}, \omega_{x\beta}, \omega_{zc}, \omega_{z\beta}$	Angular velocity flapping contributions
ϕ	Flow angle
ϕ_b, θ_b, Ψ_b	<i>body</i> axis roll, pitch, and yaw
Ψ_h	Rotor azimuth
Ψ_w	Rotor sideslip angle
ρ	Air density
τ	Generic value
θ	Blade pitch
$\theta_0, \theta_c, \theta_s, \theta_{rud}$	Collective, longitudinal, lateral, and rudder inputs
\tilde{L}	Inflow dynamic gains matrix
$\underline{\alpha}$	<i>body</i> axis angular acceleration vector
$\underline{\beta}$	Multi-blade coordinate flapping
$\underline{\beta}_w$	Multi-blade coordinate flapping in <i>wind</i> axis
$\underline{\lambda}_w$	Quasi-static flapping <i>body</i> angular rates and acceptations
$\underline{\lambda}_w$	Quasi-static flapping normalized inflow in <i>wind</i> axis
$\underline{\omega}_1, \underline{\omega}_2, \underline{\omega}_3$	Auxiliary vectors for finding <i>blade</i> axis angular velocities
$\underline{\omega}_b$	<i>body</i> axis angular rate vector
$\underline{\omega}_w$	Quasi-static flapping angular velocity components
$\underline{\sigma}$	Lagrangian multipliers
$\underline{\theta}_w$	Quasi-static flapping control inputs in <i>wind</i> axis
$\underline{\vartheta}_{bl}$	<i>blade</i> axis local velocity vector
$\underline{\vartheta}_b$	<i>body</i> axis velocity vector
$\underline{\vartheta}_h$	<i>hub</i> axis velocity vector
\underline{a}_h	<i>hub</i> axis acceleration vector
\underline{e}	State error
\underline{F}_h	<i>hub</i> axis forces

\underline{M}_b	<i>body</i> axis moments
\underline{M}_h	<i>hub</i> axis moments
\underline{R}_{bl_i}	Location of blade section in <i>blade</i> axis
\underline{R}_h	Location of hub in <i>body</i> axis
\underline{u}	Control inputs
\underline{x}	States
\underline{x}^*	Reference states
Υ	Learning rate parameter
ϱ	Dynamic pressure
A	Rotor area
a_0	Linear variation of lift to angle of attack
$A_{\beta\theta}, A_{\beta\lambda}, A_{\beta\omega}$	Quasi-static flapping matrix coefficients
$a_{hblzc}, a_{hblz\beta}$	<i>hub</i> axis acceleration flapping contributions
C	Constraint matrix
c	Chord of blade section
C_D	Drag coefficient
C_L	Lift coefficient
C_J	Generic rotor moment coefficient
C_l	Rolling moment coefficient
C_{M0}, D_{M0}, H_{M0}	Multi-blade coordinate flapping matrix coefficients
C_M	Pitching moment coefficient
C_Q	Rotor torque coefficient
C_T	Thrust coefficient
C_{x_h}, C_{y_h}	Force coefficient in <i>hub</i> axis
C_{x_w}	Longitudinal force coefficient in <i>wind</i> axis
D_{θ_0}	Differential collective
$F_{x_{bl}}, F_{y_{bl}}, F_{z_{bl}}$	<i>blade</i> axis force components
$F_{x_b}, F_{y_b}, F_{z_b}$	<i>body</i> axis force components
$F_{x_h}, F_{y_h}, F_{z_h}$	<i>hub</i> axis force components
i, j, k	Indexing values

I_B	Blade inertia
I_{xx}	Rolling moment of inertia
I_{xz}	Pitch-yaw product of inertia
I_{yy}	Pitching moment inertia
I_{zz}	Yawing moment inertia
J	Generic rotor moment
K_β	Hub spring stiffness
L	Cost function
M	Blade mass
m	vehicle mass
$M_{x_{bl}}, M_{y_{bl}}, M_{z_{bl}}$	<i>blade</i> axis moment components
$M_{x_b}, M_{y_b}, M_{z_b}$	<i>body</i> axis moment components
$M_{x_h}, M_{y_h}, M_{z_h}$	<i>hub</i> axis moment components
N_{bs}	Number of blade sections
N_B	Number of blades
N_{cl}	Description of rotor rotation direction
N_s	Number of discrete time steps within prediction horizon
p, q, r	<i>body</i> axis angular rates
p_w, q_w	Rolling and pitching rates in the <i>wind</i> axis
Q	State weighting matrix
R	Controls weighting matrix
S	Terminal state weighting matrix
s	Rotor solidity
S_β	Stiffness number
T	Rotor thrust
t_s	Discrete time step
T_b^E	Transformation from <i>body</i> to <i>Earth</i> axes
T_b^h	Transformation from <i>body</i> to <i>hub</i> axes
T_E^b	Transformation from <i>Earth</i> to <i>body</i> axes
T_h^{bl}	Transformation from <i>hub</i> to <i>blade</i> axes

U	Control matrix
$U*$	Reshaped controls matrix
u_{bl}, v_{bl}, w_{bl}	<i>blade</i> axis velocity components
u_b, v_b, w_b	<i>body</i> axis velocity components
u_h, v_h, w_h	<i>hub</i> axis velocity components
U_{max}, U_{min}	Maximum and minimum control limits
U_P	Local normal velocity
U_T	Local tangential velocity
V	Mass flow parameter
V_m	Harmonic rotor inflow
V_t	Total rotor inflow
V_T	Blade tip velocity
X	State matrix
x, y, z	Flat Earth location
x^+	Activation function applied to x
x_{bl}, y_{bl}, z_{bl}	<i>blade</i> axis components
x_b, y_b, z_b	<i>body</i> axis components
x_h, y_h, z_h	<i>body</i> axis components
λ_β^2	Flap frequency
x'	Derivative of x with respect to azimuth angle
x''	Second derivative of x with respect to azimuth angle
${}^\beta T_w^h$	Transformation for flapping from <i>wind</i> to <i>hub</i> axes
${}^\Delta T_w^h$	Transformation for inflow from <i>wind</i> to <i>hub</i> axes
${}^\omega T_w^h$	Transformation for quasi-static flapping angular rates and accelerations from <i>wind</i> to <i>hub</i> axes
${}^\theta T_w^h$	Transformation for quasi-static flapping controls from <i>wind</i> to <i>hub</i> axes

ABSTRACT

Tilt rotor vehicles are governed by FAA laws also used for conventional helicopters, which require autorotational maneuvering and landing given a total power failure. With low inertia rotors and high disk loading of tilt rotor vehicles, this already difficult task becomes significantly more challenging. In this work, a model predictive controller is developed to autonomously maneuver and land a tilt rotor given complete power loss. A high fidelity model of a tilt rotor vehicle is created and used to simulate the vehicle dynamics and response to control inputs. A reduced order dynamic model is used within a model predictive control algorithm to predict vehicle states on a receding horizon and optimize the control inputs. Constraint and cost functions are designed to promote reliable nonlinear optimization using a recurrent neural network. Simulation results show that the controller works in both normal operation states and in power-off autorotation.

1. Introduction

Tilt rotor (TR) vehicle designs have been explored since the middle of the 20th century with a goal of increasing the mission radius of vertical take-off and landing (VTOL) capable vehicles. Tilt rotor vehicles provide the unique characteristic of blending VTOL capability with efficient wing borne flight. The cruising aerodynamic efficiency of a fixed wing is far superior to that of a rotor disk. While TR designs combine useful aspects of helicopters and planes, they are susceptible to issues involving high disk loading and low rotor inertia.

Separating one large rotor into two smaller rotors significantly decreases the overall rotor disk area and, in turn, increases the disk loading. Higher disk loading increases power demands across all ranges of rotor borne flight. Inertia depends on the square of length, so shorter blades inherently have lower inertia, and low rotor inertia hinders the amount of energy that can be stored in the rotor system.

The FAA classifies large TR aircraft as large helicopters and requires that they can autorotate (AR) and land under total power loss (Grant, 2016). Autorotation is a special operating point of a windmilling rotor disk where zero net power is required for steady state operation. Gravitational potential energy is converted into mechanical energy within the rotor system, providing the power needed to produce thrust and control forces. The rate of descent required for AR is quite high for any helicopter. A TR with high disk loading and large power demands then requires an even higher rate of descent. Kinetic energy is stored in the rotor during AR and used for control and flaring. Tilt rotor aircraft have low rotor inertia leading to low stored kinetic energy and rapid changes in rotational speed while maneuvering. Autorotation of a TR aircraft, while necessary, is difficult and can be aided by autonomous control.

Model predictive control (MPC) computes the control input history that minimizes a mission-based cost function over a finite time horizon. A portion of

this control history is implemented after which a new optimization problem is solved; therefore, the process continues over a sliding time horizon until the mission is complete. The vehicle state history evolves based on the control input history through a nonlinear dynamic model. The cost function incorporates objectives such as tracking a desired reference trajectory and minimizing control effort. In addition, state and control inequality constraints can be enforced in the MPC implementation, which in the case of AR can include limits on the rotational speed of the rotors and swash plate control limits. MPC can also be implemented as a trajectory planning algorithm, which can provide and re-plan reference trajectories during the AR phase of flight.

1.1 XV-15 Overview

The Bell XV-15 started development in the early 1970's as a refined test bed for TR operations after its predecessor - the Bell XV-3. The idea for the XV-3 was first introduced as military leaders began realizing the consequences of conventional helicopter aerodynamics which reduced mission range effectiveness. The XV-3 was the first convertible aircraft design heavily endorsed by any military and subsequently faced significant issues. The XV-3 experienced rotor dynamic instability during hover and transition but, with some perseverance, designers were able to overcome the challenges and prove that the TR design was useful (Maisel et al., 2000). Taking the lessons from the XV-3, Bell introduced designs for the XV-15 which became largely successful and can be seen in modern day TR designs including the Bell-Boeing V-22 and Leonardo AW609.

The XV-15 had two 25ft diameter gimballed rotors. A gimballed rotor includes blades that are stiffly connected at the hub and move as a together as a disk. The rigid hub connection helps alleviate flapping instabilities and unwanted forces experienced by the XV-3 (Edenborough et al., 1972). Deriving equations of motion and connecting the aerodynamics for all three rotor blades proves difficult, so an

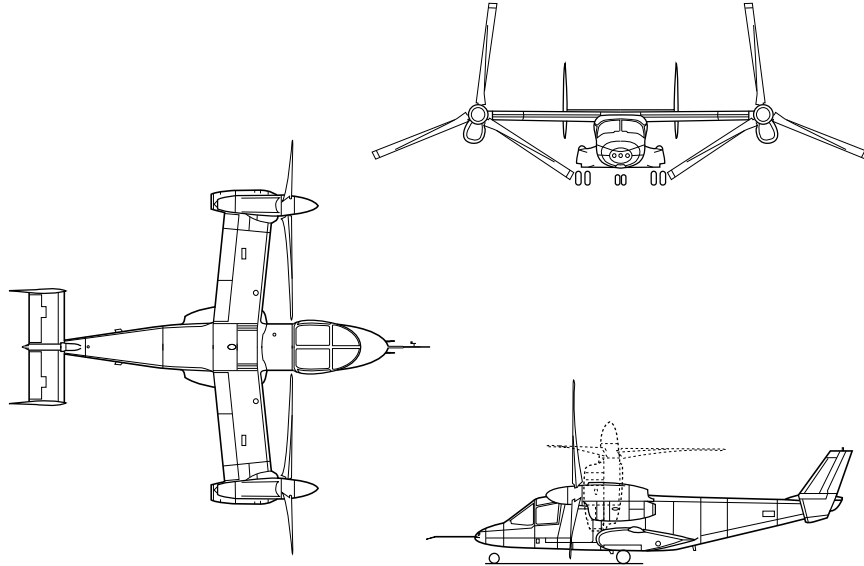


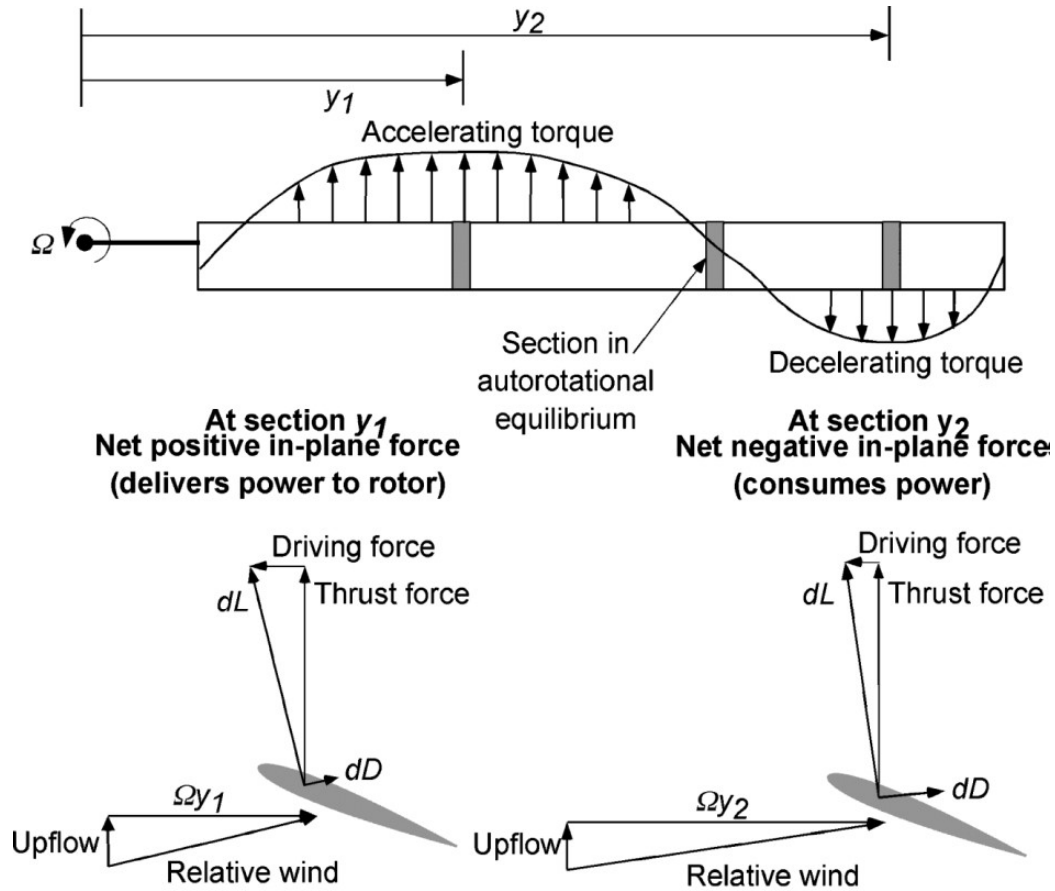
Figure 1.1. Bell XV-15 experimental tilt rotor aircraft (Curry, 2002)

individual blade model with low hub spring stiffness is used giving the system a flap frequency close to one and low moment transmission through the hub (McVicar, 1993).

1.2 Autorotation and Landing

Autorotation is a specific operating point in rotorcraft descending flight where the upward flow of air through the rotor disk provides power to spin the rotor and produce thrust. Disengaged from the engine, the rotor transmits no moments parallel to the rotor shaft. Rotor control inputs must be carefully applied as they not only maneuver the vehicle but also affect the rotor speed. If the rotor speed increases too much, the rotor structure (hub, blades, etc...) will catastrophically fail, and if the rotor speed falls too low, the blades will stall, stop producing thrust, and not provide any chance of recovery. Typical helicopter rotor systems can operate from 0.8 - 1.2 times the normal operating rotor speed (Leishman, 2006).

Upward flow through the rotor disk combines with the local blade velocities generating power consuming and power producing sections. Shown in Figure 1.2, at the inner portion of the disk, where local blade velocities are low, high angles of attack cause the total aerodynamic force to point in the direction of rotation and



NOTE: Angles exaggerated for clarity

Figure 1.2. Blade aerodynamics during autorotation (Leishman, 2006).

produce power within the rotor system. The outer portion of the disk acts as the rotor normally would: consuming power and creating thrust. When controlling a helicopter in AR, a pilot is balancing the areas of power consumption and production within the rotor disk.

A rotor system in vertical AR is naturally stable when separated from the vehicle. Figure 1.3 shows the power required for axial flight of a generic helicopter. The required power curve's intersection with the horizontal axis is the point of ideal AR, and the slope of the curve shows the state's stability. The required power curve is based on momentum theory and experimental results. Positive climb velocities and high descent velocities well define the rotor wake so momentum theory can be used to find the inflow conditions. Between -2 and $0 V_c/v_h$, where

V_c is the climb velocity and v_h is the induced inflow at hover, momentum theory fails as the wake structure is not well defined. Competing forces of downward induced flow and upward incoming flow produce highly unsteady and non-uniform aerodynamics.

The positive slope at a zero power requirement means a small increase in climb velocity causes an increase in power consumption and, therefore a subsequent decrease in climb velocity. Conversely, a decrease in climb velocity causes an increase in power production and thrust. Combining the rotor with slower vehicle dynamics in forward flight complicates the rotor stability significantly, but the same general ideas hold true.

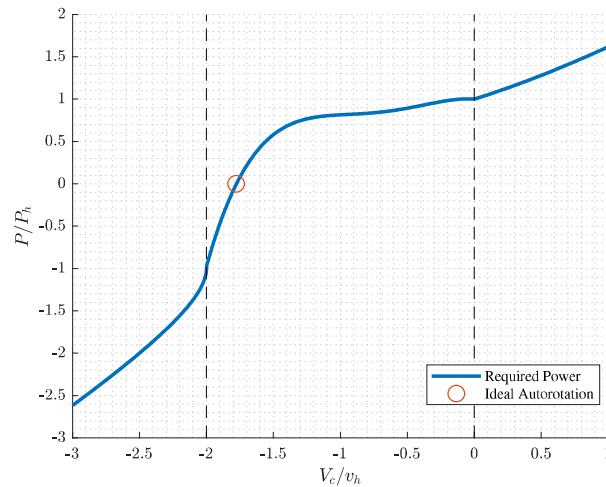


Figure 1.3. Axial flight power requirements normalized by hovering conditions (Leishman, 2006).

Rotorcraft exhibit aerodynamics similar to fixed wing aircraft. Induced rotor power requirements initially decrease with forward speed as the rotor inflow decreases. Parasitic airframe drag increases quickly at high speeds from a cubic speed dependency along with a quadratic power increase from the parasitic drag of the blades themselves. Combining the induced and parasitic powers yields the curve shown in Figure 1.4. Since AR relies on power conversion from gravitational potential energy to mechanical energy which can do some work on a

fluid, the slowest descent rates occur when the power requirements for flight are lowest. Autorotation must occur with some forward speed so descent rate can be minimized and the time in the air maximized.

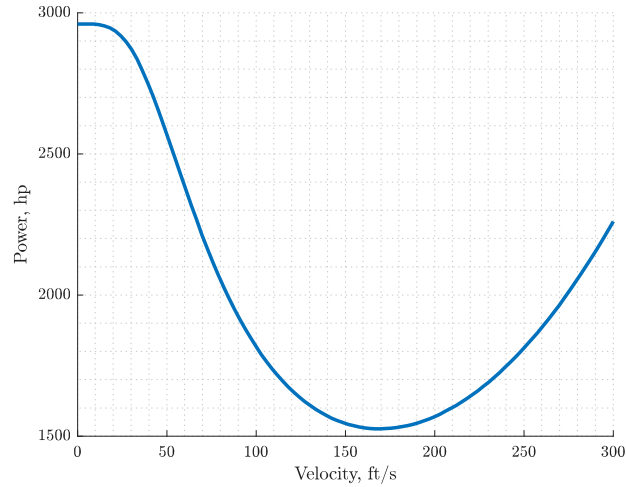


Figure 1.4. Generic forward flight power requirements for 13000lb aircraft with 2 - 25ft rotors and a flat plate drag area of $9ft^2$ (Leishman, 2006).

Autorotation refers to the steady state descent where total power requirements are equal to zero and is simply a piece of the entire power-off landing maneuver. Figure 1.5 shows the entire procedure. At condition 1, the helicopter has just experienced a power failure and must enter AR. Entering AR requires lowering blade pitch, so rotor speed is preserved, and increasing or decreasing forward speed to either the point of minimum descent rate or maximum range. Once in AR, condition 2, light maneuvering can be performed while maintaining rotor speed. The most demanding portion of landing are the last three steps. The flare, condition 3, requires the vehicle to pitch up to arrest forward velocity, which will in turn increase the rotor speed significantly. Increasing the blade pitch and leveling the vehicle at condition 4 then uses the stored kinetic within the rotor so a soft landing with low forward speed can be performed at condition 5.

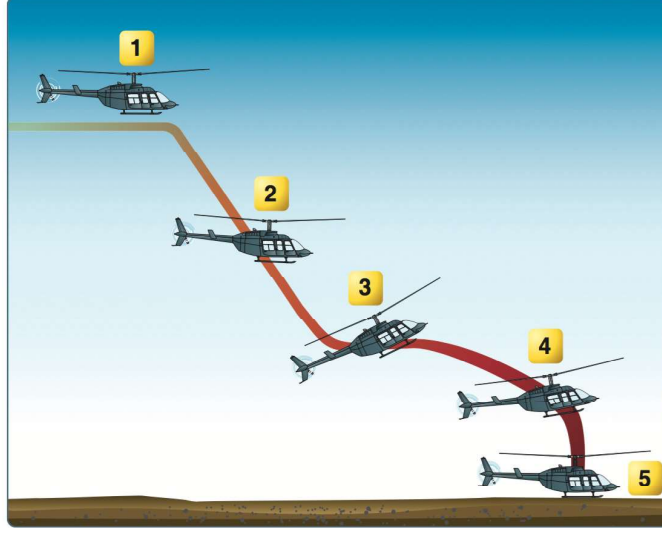


Figure 1.5. Power out maneuvers for generic helicopter (Koyovis, 2017).

1.3 Objectives

The objective of this thesis is to develop a MPC algorithm which can bring a TR into a steady autorotational descent. In doing so, a high fidelity simulation is built to represent the vehicle begin controlled. and a reduced order model (ROM) is created to be used for prediction within the controller. A recurrent neural network (RNN) is used to solve the constrained nonlinear optimization problem contained within the MPC setup.

First, the current technology and history of AR control is explored with an emphasis on the importance of optimal control. In chapter 3, the high fidelity simulation is explained including: rigid body motion, blade mechanics, aerodynamics, and inflow dynamics. The axes systems used within the model are explained along with the rotations used to transport vectors between them. Chapter 4 explains the controller design. First, the simplifying assumptions used to create the ROM are introduced and the limitations explained. Then, the optimal control problem is posed including the cost and constraint functions. A mathematical derivation of the cost and constraint function gradients is given along with the implementation within the RNN. Finally, convergence of the optimization

algorithm is proven for the given prediction model. Chapter 5 shows a comparison the two models along with controlled simulations. The comparison is done with constant and variable rotor speed. Controlled simulations are shown with the vehicle flying in a nominal flight condition as well as AR. Lastly, the constraints are tested by starting simulations near constraint boundaries and placing reference trajectories beyond those boundaries.

2. Current Technology

Optimal control was first applied to autorotation (AR) by Johnson (1977) where he determined longitudinal power-off landing trajectories given varying initial conditions. A nonlinear longitudinal model was derived and used with a gradient descent method to minimize the landing velocity while finding optimal control inputs based on altitude. Johnson proved that the optimal AR trajectory from hover is a vertical descent. The method was compared to Bell Helicopter flight test data of a OH-58A with a High Energy Rotor System and showed significant correlation (Dooley & Yeary, 1979).

Lee extended Johnson's work by adding inequality constraints to the optimization problem (Lee, 1985; Lee, 1990). Using slack variables, the inequality constraints placed on rotor thrust and sink rate were transformed to equality constraints. A Sequential Gradient Restoration Algorithm was used to find trajectories of power-off landings from hover and forward flight, and then compared to the OH-58A flight test data.

Using a 2-D point mass model of a UH-60A helicopter, optimal trajectories were found for a variety of takeoff and landing scenarios with one engine inoperative (OEI) (Zhao, Sharma, et al., 1995; Zhao, Jhemi, et al., 1996). Specifying constraints on rotor thrust, speed, and power consumption, the helicopter model successfully performed emergency maneuvers including: rejected takeoff, continued takeoff, continued landing, and bailed landing. The work was then extended the work to tilt rotor (TR) vehicles. Short OEI runway takeoffs of the XV-15 were explored where path constraints were placed on pitch, rotor speed, altitude, thrust, longitudinal stick, and nacelle angle (Carlson & Zhao, 2002; Carlson & Zhao, 2003). Sensitive XV-15 city center operations with OEI were posed as nonlinear optimization problems focused on minimizing the heliport size (Carlson & Zhao, 2004). Maximum safe gross weight of the vehicle was found to be

determined by OEI vertical takeoff.

Tilt rotor operations were further explored by Carlson (1999) where he examined OEI and all engines inoperative (AEI) with the XV-15. The low inertia rotor systems require that the XV-15 be able to operate with lower rotor speeds than a conventional helicopter. Keeping the rotor speed above 78%, the nominal value, requires that the vehicle begin power-off landing maneuvers with air speeds above 50 knots. Relaxing the rotor speed constraint to 60% allows the vehicle to land from hover at 850ft altitude. Carlson showed that landings that start on the edge of the height-velocity (H-V) curve occur with significant forward and vertical speed.

Bibik and Narkiewicz (2012) continued work on OEI and AEI situations in conventional utility helicopters by including a more comprehensive model. Previously, only longitudinal motion was considered, which is a reasonable assumption if the vehicle does not need to maneuver. A discrete time adaptive optimal controller was used, which not only used a standard quadratic cost function to find optimal trajectories but also allowed the performance indices and constraints to be changed real time, depending on flight conditions. During the flare and landing phases, rotor speed limits can be relaxed as the vehicle is close enough to the ground to withstand an impact and the rotor speed needs to drastically vary as the vehicle transfers potential and kinetic energy to mechanical work.

A real time trajectory optimization algorithm was developed by Aponso et al. (2005). Focused on making the system usable by a remote human pilot, the optimal control outputs of the algorithm were collective and vehicle pitch. An outer loop controller or pilot could then use the collective and longitudinal cyclic to follow the trajectory. Power-off maneuvers could be initiated within the unsafe region of the H-V curve if the optimal control inputs were used. The work focused on

an unmanned helicopter with the goal of saving valuable packages and sensors on board.

Taamallah et al. (2017) used the concept of differential flatness to reduce a typically large optimization problem to a simple algebraic one. By excluding the traditional helicopter control inputs and instead finding total aerodynamic forces and moments, the dynamics problem becomes flat, which allows Fliess’s ideas of differential flatness to be used (Fliess, 1990). Simplifying the model to exclude high order rotor dynamics is valid assuming the controls stay within a certain bandwidth. A robust control algorithm was then used for trajectory tracking.

Using the work of Aponso (2005), Tierny and Langelaan (2010) developed a method of computing a safe set of vehicle states for flare entrance. If the vehicle enters the flare maneuver within the safe set, a safe trajectory to touchdown is guaranteed to exist. The states included within the set are the spatial positions from the landing point, airspeed, rotor speed, and descent rate. Separating the entire landing maneuver into different sections could allow different algorithms, each optimized for their own portion, to feed into each other. The AR controller could find optimal trajectories such that its end point lies within the safe flare set.

Bang and Lee (2008) used trained radial basis functions to control AR of a simplified point mass model of a OH-58A helicopter based on Johnson’s work. Only longitudinal motion is considered in the model and rotor mechanics are simplified to disk actuation. These simplifications helped training with a neural network and a reinforcement learning algorithm described by Watkins (1989). After 9000 epochs, the controller was performing AR maneuvers with an 80% success rate. Around 19000 RBFs were trained within a large action and state space, leading to high computational load.

Abbeel et al. (2008) simulated and implemented an autonomous AR controller for a small remote controlled helicopter. A nonlinear form of the linear quadratic

regulator, differential dynamic programming (DDP), was used to follow a specified trajectory which was an optimum of several sub-optimal trajectories. A human pilot performed several AR maneuvers in which the vehicle states were recorded and then used to predict the optimal trajectory for AR. DDP allows the control optimization problem to be reduced from minimizing a $Nm \times Nm$ matrix to a $m \times m$ matrix N times, where N denotes the length of the control horizon and m is the number of control dimensions (Tassa et al., 2014). The controller successfully landed a RC helicopter several times using an AR maneuver. However, being constrained to a single optimal trajectory, the system is not robust to large disturbances or varying initial conditions.

Dalamagkidis et al. (2010) used model predictive control (MPC) combined with a recurrent neural network (RNN) to control a small unmanned helicopter in a vertical AR. A vertical AR was selected as it provides the lowest chance of the vehicle hitting another object. The primary goal of the controller was to reduce probability of damaging outside objects with vehicle safety being the secondary goal. The RNN decreased computation cost of the nonlinear optimization problem required for MPC by removing the need to calculate the second derivative of the system's Hamiltonian. An ordinary differential equation (ODE) whose equilibrium point is the solution to the optimization problem is modelled and solved by the RNN. Xia et al. (2008) proved that the RNN will globally converge for constrained nonlinear optimization problems if the Hessian of the function is postive semi-definite.

Yomchinda et al. (2012) used nonlinear dynamic inversion (NLDI) to provide stability to a helicopter in AR. Inner loop NLDI sets the roll and pitch axes responses to Attitude Command/Attitude Hold, and the yaw axis to Rate Command/Attitude Hold which allows an outer loop position controller to make simple attitude and rate commands. The outer loop control works to keep the

helicopter in coordinated flight as it maneuvers through the given trajectories. A human pilot could replace the outer loop position control, with the NLDI controller still performing attitude control.

An optimal control algorithm was used by Yomchinda (2013) to generate trajectories for the different phases of the AR maneuver including entrance, autorotation with coordinated maneuvers, flaring, and landing. Using a simplified helicopter model, assuming the rotor tip path plane remains in the same orientation with respect to the fuselage, optimal trajectories were computed off-line using a standard quadratic cost function. Using airspeed, interpolation was performed between the stored entrance trajectories. Maneuvers were optimized using curves described by Dubins (1957 & 1961). Assuming a mass is moving at a constant rate and must pass through two points on a plane, Dubins defines a set of curves that will give the shortest path between the points. Yomchinda used these curves to create a 2-D planar trajectory, which was then extended to 3-D by including descent rates.

3. Tilt Rotor Simulation

The XV-15 is simulated using a nonlinear dynamics model described by McVicar (1993) utilizing individual rigid blade motion and a dynamic inflow model. Rigid body motion is tracked in a flat earth reference frame.

3.1 Vehicle Axes Systems

The TR vehicle utilizes 9 frames of reference for simulation: 1 centered at the vehicle CG, 1 aligned with each of the 2 rotor hubs, and 1 attached to each of the 6 rotor blades. The set of axes located at the vehicle CG is used for rigid body motion and is denoted as the *body* axes (subscript b). Moving from the *body* axes to *hub* axes (subscript h) requires a rotation around y_b by the tilt angle γ . At $\gamma = 0$ the nacelle is straight up, in hover mode; while at $\gamma = 90^\circ$ the nacelle is pointing ahead, in forward flight mode. For this work, γ is fixed at 0 reducing Equation (3.1) to identity and aligning the *hub* and *body* axes.

$$T_b^h = \begin{bmatrix} \cos \gamma & 0 & -\sin \gamma \\ 0 & 1 & 0 \\ \sin \gamma & 0 & \cos \gamma \end{bmatrix} \quad (3.1)$$

The rotor blades rotate around z_h by the azimuth angle Ψ_h and flap in the *blade* axes (subscript bl), around y_{bl} by the angle β . The *blade* axes system is fixed to the blade at the hub with x_{bl} pointing along the span of the blade, z_{bl} pointing in the direction of z_h , and y_{bl} pointing either into or away from the wind, depending on the direction of rotation. A 0 azimuth angle aligns the x_{bl} axis along the $-x_h$ axis with positive rotation Ω occurring around the $-z_h$ axis: counter-clockwise when viewed from above.

$$T_h^{bl} = \begin{bmatrix} \cos \beta & 0 & -\sin \beta \\ 0 & 1 & 0 \\ \sin \beta & 0 & \cos \beta \end{bmatrix} \begin{bmatrix} -\cos \Psi_h & \sin \Psi_h & 0 \\ -\sin \Psi_h & -\cos \Psi_h & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.2)$$

Assuming a small flapping angle ($\sin \beta \approx \beta$ and $\cos \beta \approx 1$), Equation (3.2) can be reduced to Equation (3.3).

$$T_h^{bl} = \begin{bmatrix} -\cos \Psi_h & \sin \Psi_h & -\beta \\ -\sin \Psi_h & -\cos \Psi_h & 0 \\ -\beta \cos \Psi_h & \beta \sin \Psi_h & 1 \end{bmatrix} \quad (3.3)$$

Converting from the *hub* or *blade* axis back to the *body* requires transposing the respective rotation matrix or transposing multiple matrices and reversing the multiplication.

3.2 Rigid Body Dynamics

Applications of rigid body motion are used across wide ranges of mechanical engineering and design. From spacecraft and aircraft design to automobile and robotics, any project relating to the motion of body through space will use the basic principles of rigid body motion. Baruh (1999) explains the fundamental concepts of mechanics put in place by scientists like Euler. Pamadi (2004) explains rigid body dynamics related directly to aircraft stability.

3.2.1 Rigid Body Motion

Being relatively close to the Earth, aircraft are subject to an approximately uniform constant gravity field that causes acceleration towards the ground. Taking the effects of gravity into account, the rate of change of the *body* velocities are given by:

$$\dot{u}_b = -(w_b q - v_b r) + \frac{1}{m} F_{x_b} - g \sin \theta_b \quad (3.4)$$

$$\dot{v}_b = -(u_b r - w_b p) + \frac{1}{m} F_{y_b} + g \cos \theta_b \sin \phi_b \quad (3.5)$$

$$\dot{w}_b = -(v_b p - u_b q) + \frac{1}{m} F_{z_b} - g \cos \theta_b \cos \phi_b \quad (3.6)$$

where u_b , v_b , and w_b are the x_b , y_b , and z_b velocities respectively, p , q , and r are the *body* axes angular velocities, F_{x_b} , F_{y_b} , and F_{z_b} are the total forces acting in each respective axis, g is the acceleration due to gravity, and ϕ_b and θ_b are the flat earth roll and pitch Euler angles. Navigating through 3-D space requires the aircraft to rotate about the three *body* axes. Maneuvers begin by applying moments to the body, causing changes in angular rates given by:

$$\dot{p} = I_{xx}^{-1} [(I_{yy} - I_{zz})qr + I_{xz}(\dot{r} + pq) + M_{x_b}] \quad (3.7)$$

$$\dot{q} = I_{yy}^{-1} [(I_{zz} - I_{xx})rp + I_{xz}(r^2 - p^2) + M_{y_b}] \quad (3.8)$$

$$\dot{r} = I_{zz}^{-1} [(I_{xx} - I_{yy})pq + I_{xz}(\dot{p} - qr) + M_{z_b}] \quad (3.9)$$

where I_{xx} , I_{yy} , and I_{zz} are the moments of inertia around the *body* axes and I_{xz} is the product of inertia caused by asymmetry within the vehicle structure. M_{x_b} , M_{y_b} , and M_{z_b} are the total moments acting around each *body* axis. Equation (3.7) relies on Equation (3.9), so finding the solution requires substituting one equation into the other. Rewriting Equation (3.7) using Equation (3.9) yields:

$$\dot{p} = \hat{I}_{xz}^{-1} \left[(I_{yy} - I_{zz})qr + M_{x_b} + \frac{I_{xz}}{I_{zz}} \left(\frac{I_{zz}(I_{xx} - I_{yy}) - 1}{I_{zz}} pq - I_{xz}qr + M_{z_b} \right) \right] \quad (3.10)$$

where \hat{I}_{xz} is given by:

$$\hat{I}_{xz} = I_{xx} \left(1 - \frac{I_{xz}^2}{I_{xx}I_{zz}} \right) \quad (3.11)$$

which allows integration of the dynamic system.

The standard aircraft Euler angles are a set of rotations known as a 3-2-1 sequence since the transformation from the reference Earth frame to the *body* frame requires a rotation around the 3 (yaw) axis, then the 2 (pitch) axis, and, finally, the 1 (roll) axis (Baruh, 1999). In total, there are 12 possible combinations of rotations that can be used to describe axes transformations. With 3 rotations, any

general transformation from one axes system to another can be performed but in each set, a singularity will exist for some given rotation. For the 3-2-1 sequence, the singularity occurs when the 2nd angle (pitch) goes to 90° : experienced by the tangent and secant terms in Equation (3.18) and Equation (3.20). This is not an issue for non-aerobatic aircraft as they do not fly straight up away from the ground. In the sense of MPC, some issues could arise if a candidate control input causes large motions over the prediction horizon. The 3-2-1 rotation, from the *Earth* to *body* axes, is given by the following matrix operation:

$$T_E^b = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi_b & \sin \phi_b \\ 0 & -\sin \phi_b & \cos \phi_b \end{bmatrix} \begin{bmatrix} \cos \theta_b & 0 & -\sin \theta_b \\ 0 & 1 & 0 \\ \sin \theta_b & 0 & \cos \theta_b \end{bmatrix} \begin{bmatrix} \cos \Psi_b & \sin \Psi_b & 0 \\ -\sin \Psi_b & \cos \Psi_b & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.12)$$

with ϕ_b , θ_b , and Ψ_b representing the roll, pitch and yaw respectively. The transformation from the *body* axes to the *Earth* axes is obtained by transposing the product of the three transformations:

$$T_b^E = \begin{bmatrix} \cos \theta_b \cos \Psi_b & \sin \theta_b \sin \phi_b \cos \Psi_b - \sin \Psi_b \cos \phi_b & \sin \theta_b \cos \phi_b \cos \Psi_b + \sin \Psi_b \sin \phi_b \\ \cos \theta_b \sin \Psi_b & \sin \Psi_b \sin \theta_b \sin \phi_b + \cos \Psi_b \cos \phi_b & \sin \Psi_b \sin \theta_b \cos \phi_b - \cos \Psi_b \sin \phi_b \\ -\sin \theta_b & \sin \phi_b \cos \theta_b & \cos \phi_b \cos \theta_b \end{bmatrix} \quad (3.13)$$

Planning or following a trajectory in 3-D space requires tracking of the vehicle position through time using Equation (3.13). Transforming the *body* velocities to the *Earth* frame, the positions of the vehicle can be tracked with:

$$\begin{aligned} \dot{x} = & u_b \cos \theta_b \cos \Psi_b + v_b (\cos \Psi_b \sin \phi_b \sin \theta_b - \sin \Psi_b \cos \phi_b) \\ & + w_b (\sin \Psi_b \sin \phi_b + \cos \Psi_b \cos \phi_b \sin \theta_b) \end{aligned} \quad (3.14)$$

$$\begin{aligned} \dot{y} = & u_b \sin \Psi_b \cos \theta_b + v_b (\cos \Psi_b \cos \phi_b + \sin \Psi_b \sin \phi_b \sin \theta_b) \\ & + w_b (\sin \Psi_b \cos \phi_b \sin \theta_b - \cos \Psi_b \sin \theta_b) \end{aligned} \quad (3.15)$$

$$\dot{z} = -u_b \sin \theta_b + v_b \sin \phi_b \cos \theta_b + w_b \cos \phi_b \cos \theta_b \quad (3.16)$$

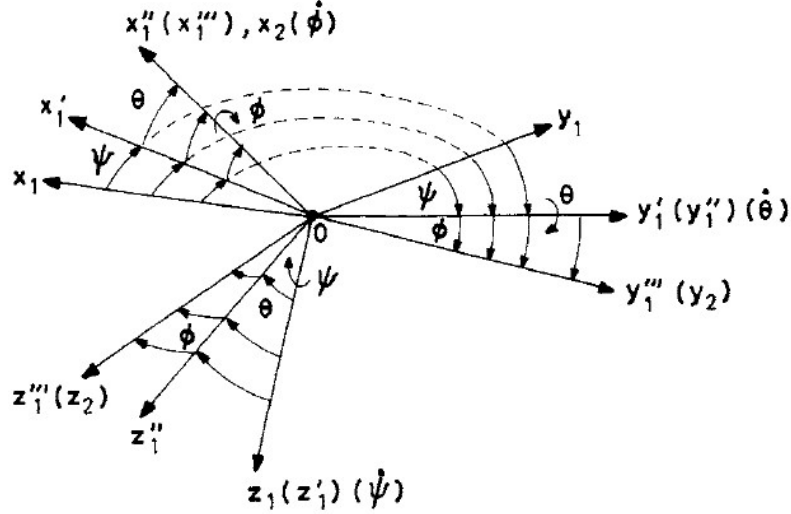


Figure 3.1. 3-2-1 Rotation from *Earth* frame to *body* frame (Pamadi, 2004).

In level flight, $\phi_b = \theta_b = 0$, roll rate and pitch rate directly coordinate to changes in roll and pitch, but given large values of roll and pitch, the *body* angular rates do not correlate directly to changes in the respective Euler angles (i.e. if $\phi_b = 90^\circ$, yaw rate causes change in pitch). Transforming *body* angular velocities to *Earth* angular velocities requires a more graphic analysis than standard vector transformations performed with Equation (3.13). Figure 3.1 shows the transformation steps from the *Earth* ($0x_1y_1z_1$) to *body* frame ($0x_1'''y_1'''z_1'''$). ϕ_b follows the x_b axis, meaning p directly tracks into $\dot{\phi}_b$, while θ_b and Ψ_b align with varying transformation steps. θ_b is a rotation around a new axis (y_1'') that has been rotated from the *body* frame by ϕ_b . Transforming $\dot{\theta}_b$ to back to the *body* frame simply requires undoing the ϕ_b rotation around x_1''' . Secondly, Ψ_b occurs around an axis that has been rotated by ϕ_b and θ_b ; transforming $\dot{\Psi}_b$ back to the *body* frame requires a rotation around the y_1'' axis by θ_b , then a rotation around x_1''' by ϕ_b . Etkin (1972) helps thoroughly explain the derivation. These sets of transformations gives the *body* angular velocities as a function of the rates of change the Euler

angles:

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin \theta_b \\ 0 & \cos \phi_b & \sin \phi_b \cos \theta_b \\ 0 & -\sin \phi_b & \cos \phi_b \cos \theta_b \end{bmatrix} \begin{bmatrix} \dot{\phi}_b \\ \dot{\theta}_b \\ \dot{\Psi}_b \end{bmatrix} \quad (3.17)$$

The transformation given by Equation (3.17) is not a proper orthogonal rotation matrix like Equation (3.13), so the inversion requires standard matrix operations instead of transposition. Euler angles can be tracked from the *body* angular rates with:

$$\dot{\phi}_b = p + q \sin \phi_b \tan \theta_b + r \cos \phi_b \tan \theta_b \quad (3.18)$$

$$\dot{\theta}_b = q \cos \phi_b - r \sin \phi_b \quad (3.19)$$

$$\dot{\Psi}_b = q \sin \phi_b \sec \theta_b + r \cos \phi_b \sec \theta_b \quad (3.20)$$

3.2.2 Hub Motion

The rotor hubs are subject to more complex motion as their rotation around the CG creates additional velocities that affect the aerodynamics of the rotor system. The transport theorem can be used to find the velocity of the hub (Baruh, 1999).

$$\underline{\vartheta}_h = \underline{\vartheta}_b + \underline{\omega}_b \times \underline{R}_h \quad (3.21)$$

where $\underline{\vartheta}_h = [u_h \ v_h \ w_h]^T$, $\underline{\vartheta}_b = [u_b \ v_b \ w_b]^T$, $\underline{\omega}_b = [p \ q \ r]^T$, and $\underline{R}_h = [x_h \ y_h \ z_h]^T$. \underline{R}_h is the relative distance between the hub and CG in the *body* axes. It is important to note that $\underline{\vartheta}_b$ must be transformed to the *hub* axes using Equation (3.1) prior to being used in Equation (3.21), but with the tilt angle at 0, the *hub* and *body* axis are aligned so no rotation needs to be performed. The linear hub velocities are given by the following the equations:

$$u_h = u_b - ry_h + qz_h \quad (3.22)$$

$$v_h = v_b + rx_h - pz_h \quad (3.23)$$

$$w_h = u_b - qx_h + py_h \quad (3.24)$$

Applying the transport theorem again to the hub velocity, while acknowledging the velocity and acceleration of the hub in the *body* axes is zero (i.e. the structure of the body is rigid), the acceleration of the hub can be written as:

$$\underline{a}_h = \underline{a}_b + \underline{\omega}_b \times (\underline{\omega}_b \times \underline{R}_h) + \underline{\alpha}_b \times \underline{R}_h \quad (3.25)$$

where $\underline{a}_h = [\dot{u}_h \ \dot{v}_h \ \dot{w}_h]^T$, $\underline{a}_b = [\dot{u}_b \ \dot{v}_b \ \dot{w}_b]^T$, and $\underline{\alpha}_b = [\dot{p} \ \dot{q} \ \dot{r}]^T$. Again, \underline{a}_b must be converted to the *hub* axes with Equation (3.1) before using Equation (3.25), but the *body* and *hub* axes are aligned so no transformation needs to be performed. The linear accelerations of the hub can then be written as:

$$\dot{u}_h = \dot{u}_b + (q^2 + r^2) x_h + (pq - \dot{r}) y_h - (rp + \dot{q}) z_h \quad (3.26)$$

$$\dot{v}_h = \dot{v}_b - (p^2 + r^2) y_h - (pq + \dot{r}) x_h - (rq + \dot{p}) z_h \quad (3.27)$$

$$\dot{w}_h = \dot{w}_b + (p^2 + q^2) x_z + (\dot{q} - rp) x_h + (rq + \dot{p}) z_h \quad (3.28)$$

3.3 Rotor System

The XV-15 has 3-bladed gimballed rotors, which will be modeled with individual rotor blades that have a low hub spring stiffness. Gimballed rotors do not transfer moments through the hub except the torque created by the rotor spinning. The low moment transfer forces the natural frequency of the rotor to one per revolution. If there is some moment transfer, as in a rigidly mounted rotor, the

dynamics of the blades themselves can be greatly affected. Padfield describes the flap frequency of a rotor (Padfield, 2018):

$$\lambda_\beta^2 = 1 + \frac{K_\beta}{I_B \Omega^2} \quad (3.29)$$

where λ_β is the flap frequency, K_β is hub spring stiffness, and I_B is the inertia of the blade. In any rotor, centrifugal forces dominate and drive the flapping frequency to one. Exciting the rotor at its natural frequency of one per revolution then causes changes that occur 90° out of phase. High hub spring stiffness will cause the natural frequency to change and the phase lag to shift. Gimballed rotors have flapping frequencies close to one so 90° phase lag should be expected with control inputs and external disturbances.

3.3.1 Blade Dynamics

The blades are rigid and free to flap relatively uncoupled from each other. Some secondary effects occur as the forces and moments caused by the blades cause changes in the inflow, which affects all of the blades. The two rotor system inflows are assumed to not affect each other.

3.3.2 Blade Motion

The blades rotate around z_h (the azimuth, Ψ_h) at the rate Ω and flap (β) around y_{bl} at the rate of $\dot{\beta}$. Both the azimuth and flapping dynamics are modelled by second order equations forced by aerodynamic and inertial terms. Aerodynamic forces and inertia are products of the blade velocities, but the angular velocity of the blades must be determined first. A single blade's angular velocity is comprised of three components (McVicar, 1993).

$$\underline{\omega}_{bl} = \underline{\omega}_1 + \underline{\omega}_2 + \underline{\omega}_3 \quad (3.30)$$

where $\underline{\omega}_1$ is the component from the vehicle's angular rates, $\underline{\omega}_2$ is the component from Ω , and $\underline{\omega}_3$ is the component from $\dot{\beta}$. Converting the vehicles angular rates into

the *blade* axes with Equation (3.2) yields:

$$\underline{\omega}_1 = \begin{bmatrix} -p \cos \Psi_h + q \sin \Psi_h - r\beta \\ -p \sin \Psi_h - q \cos \Psi_h \\ -p\beta \cos \Psi_h + q\beta \sin \Psi_h + r \end{bmatrix} \quad (3.31)$$

Converting the angular velocity around the hub to the *blade* axes yields:

$$\underline{\omega}_2 = \begin{bmatrix} \beta\Omega \\ 0 \\ -\Omega \end{bmatrix} \quad (3.32)$$

Finally, the flapping angular velocity occurs in the *blade* axis, so its contribution can be written as:

$$\underline{\omega}_3 = \begin{bmatrix} 0 \\ \dot{\beta} \\ 0 \end{bmatrix} \quad (3.33)$$

The velocity of a blade section (*i*) can then be found with the transport theorem:

$$\underline{v}_{bl_i} = \underline{v}_h + \underline{\omega}_{bl} \times \underline{R}_{bl_i} \quad (3.34)$$

where $\underline{v}_{bl_i} = [u_{bl_i} \ v_{bl_i} \ w_{bl_i}]^T$ and $\underline{R}_{bl_i} = [x_{bl_i} \ 0 \ 0]^T$. The hub velocity must be converted to the *blade* axes with Equation (3.3) before using Equation (3.34). The local velocity of the blade sections can be written as:

$$u_{bl_i} = -u_h \cos \Psi_h + v_h \sin \Psi_h - \beta w_h \quad (3.35)$$

$$v_{bl_i} = (x_{bl_i} q \beta - u_h) \sin \Psi_h - (x_{bl_i} p \beta - v_h) \cos \Psi_h + x_{bl_i} (r - \Omega) \quad (3.36)$$

$$w_{bl_i} = (x_{bl_i} q - u_h \beta) \cos \Psi_h + (v_h \beta + x_{bl_i} p) \sin \Psi_h + w_h + x_{bl_i} \dot{\beta} \quad (3.37)$$

3.3.3 Blade Aerodynamics

Splitting the blade along the spanwise direction into N_{bs} sections allows the varying aerodynamics along the blade to be analyzed in a method called blade element theory (BET), shown in Figure 3.2. Combining BET with momentum theory, for calculating the inflow, creates blade element momentum theory (BEMT). Generally, BEMT is used to analyze the performance of rotors in steady state flight conditions but, in this case, the inflow model, while still based on momentum theory, is a dynamic model that allows the inflow states to vary with time. Using the inflow model, hub velocities, and blade angular velocity, the aerodynamics of the blade can be modelled.

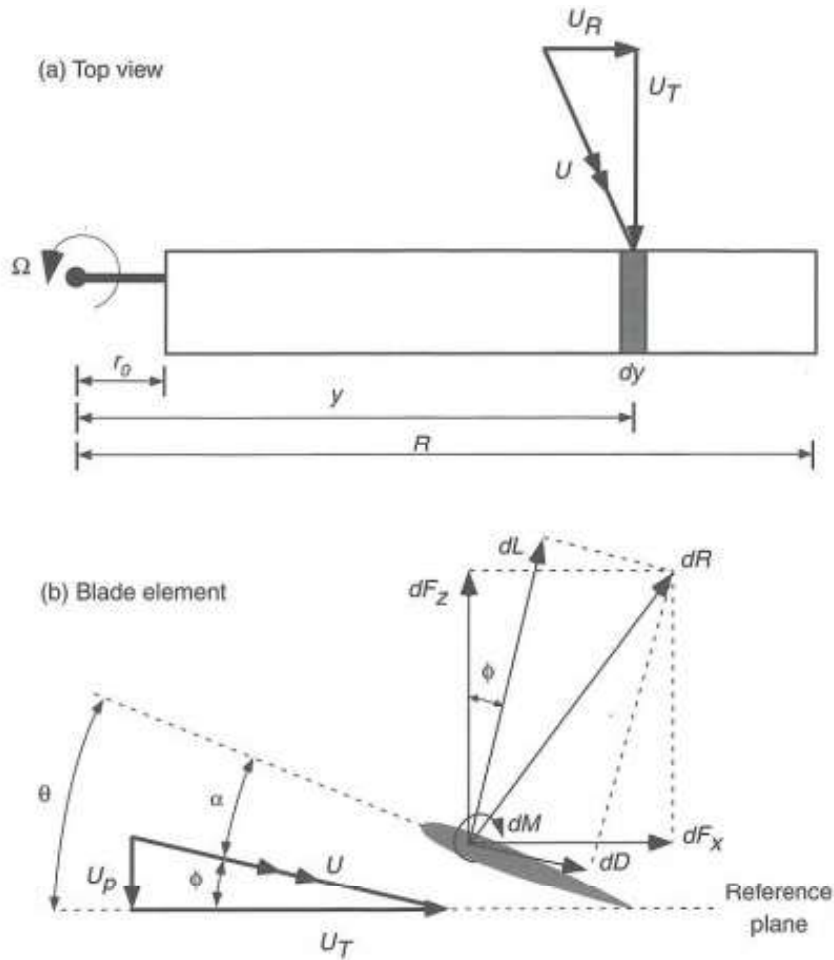


Figure 3.2. (a) shows the sectioning of a blade for blade element theory and (b) shows the aerodynamics of that blade section (Leishman, 2000).

The rotor rotation around z_h causes a large variation in velocity along the span of the blade. Near the root, the tangential velocity of the blade elements (v_{bl_i} or U_T in Figure 3.2) is low, which results in large local angles of attack and low production of thrust. The tangential velocity of the blade near the tip is highest, causing lower angles of attack and high production of thrust. Evaluating the aerodynamics at each blade section describes how a decrease in angle of attack with a large increase in velocity results in an increase in thrust. The local lift and drag created by each blade section are given by:

$$\Delta l = \varrho c C_L(\alpha) \Delta r \quad (3.38)$$

$$\Delta d = \varrho c C_D(\alpha) \Delta r \quad (3.39)$$

where Δl is the local lift, Δd is the local drag, ϱ is the dynamic pressure, c is the section chord, Δr is the section width, C_L is the lift coefficient, C_D is the drag coefficient, and α is the angle of attack. Introducing the variable N_{cl} , described by McVicar, allows the aerodynamic forces to be resolved correctly for the two rotors spinning in opposite directions (McVicar, 1993).

$$N_{cl} = \begin{cases} 1, & \text{if rotating anti-clockwise when viewed from above} \\ -1, & \text{if rotating clockwise when viewed from above} \end{cases} \quad (3.40)$$

A blade section's local tangential flow is then described by:

$$U_T = -N_{cl} v_{bl_i} \quad (3.41)$$

The perpendicular flow depends on both the local blade velocity and the inflow:

$$U_P = w_{bl_i} - \nu_0 - \frac{x_{bl_i}}{R} (\nu_c \cos \Psi_h + \nu_s \sin \Psi_h) \quad (3.42)$$

where ν_0 , ν_c , and ν_s are the normal, longitudinal, and lateral inflow components, respectively and R is the blade radius. The local dynamic pressure is then found

with:

$$\varrho = \frac{1}{2} \rho (U_T^2 + U_P^2) \quad (3.43)$$

The two flow components combine and create the flow angle:

$$\phi = \tan^{-1} \frac{U_T}{U_P} \quad (3.44)$$

Angle of attack is a combination of the applied blade pitch (θ) and the flow angle:

$$\alpha = \theta + \phi \quad (3.45)$$

The blade pitch is given by:

$$\theta = \hat{\theta}_0 + \hat{\theta}_c \cos \Psi_h + \hat{\theta}_s \sin \Psi_h + \hat{\theta}_{tw} \quad (3.46)$$

where $\hat{\theta}_0$, $\hat{\theta}_c$, and $\hat{\theta}_s$ are the collective, longitudinal cyclic, and lateral cyclic inputs, respectively, and $\hat{\theta}_{tw}$ is the local blade pitch due to twist. The total blade forces in *blade* axes can then be resolved using the flow angle with lift and drag components:

$$F_{x_{bl}} = 0 \quad (3.47)$$

$$F_{y_{bl}} = N_{cl} \sum_{i=1}^{N_{bs}} [\Delta d_i \cos \phi_i - \Delta l_i \sin \phi_i] \quad (3.48)$$

$$F_{z_{bl}} = - \sum_{i=1}^{N_{bs}} [\Delta d_i \sin \phi_i + \Delta l_i \cos \phi_i] \quad (3.49)$$

where $F_{x_{bl}}$, $F_{y_{bl}}$, and $F_{z_{bl}}$ are the total aerodynamic forces in the blade axes. Only 2-D aerodynamics are considered in the model, which allows consideration for stall and Reynold's number effects but does include any spanwise flow effects. Increased aerodynamic fidelity is outside the scope of this model as it increases the inflow calculation complexity greatly. The aerodynamic moments acting on the blades are also important, and again, the moment acting around the x_{bl} axis will be ignored.

The pitching moment created by the blades is important to rotor system design as the pitch links and control actuators must be able to both handle and overcome the forces created. However, for the sake of this study, the pitching moment effects on the vehicle and controls will be neglected. The aerodynamic moments are then given by:

$$M_{x_{bl}} = 0 \quad (3.50)$$

$$M_{y_{bl}} = \sum_{i=1}^{N_{bs}} [x_{bl_i} (\Delta d_i \sin \phi_i + \Delta l_i \cos \phi_i)] \quad (3.51)$$

$$M_{z_{bl}} = N_{cl} \sum_{i=1}^{N_{bs}} [x_{bl_i} (\Delta d_i \cos \phi_i - \Delta l_i \sin \phi_i)] \quad (3.52)$$

3.3.4 Rotor Forces and Moments

The total rotor forces acting on the hub are resolved using the individual blade forces and the transpose of Equation (3.3):

$$F_{x_h} = \sum_{j=1}^{N_b} [-F_{y_{blj}} \sin \Psi_{hj} - F_{z_{blj}} \cos \Psi_{hj}] \quad (3.53)$$

$$F_{z_h} = \sum_{j=1}^{N_b} [-F_{y_{blj}} \cos \Psi_{hj} + F_{z_{blj}} \sin \Psi_{hj}] \quad (3.54)$$

$$F_{z_h} = \sum_{j=1}^{N_b} [F_{z_{blj}}] \quad (3.55)$$

where j is a summation variable representing each blade. The aerodynamic blade forces are directly transferred to the hub but the moments are not. The torque required to spin the rotor, $M_{z_{bl}}$, is normally transferred to a transmission in the hub through a shaft; however, while in AR, a clutch is opened allowing the rotors to spin freely. In a conventional helicopter, disconnecting the main rotor not only

allows recovery in total power failure but also provides a means of handling the loss of tail rotor effectiveness, as the main job of the tail rotor is to counter the large moment created by the main rotor. Tilt rotor vehicles do not have tail rotors because the counter rotation of the two large rotors creates a zero net torque. The only moment transfer to the hub is through the hub spring stiffness:

$$M_{x_h} = -K_\beta \sum_{j=1}^{N_b} [\beta_j \sin \Psi_{hj}] \quad (3.56)$$

$$M_{y_h} = -K_\beta \sum_{j=1}^{N_b} [\beta_j \cos \Psi_{hj}] \quad (3.57)$$

$$M_{z_h} = 0 \quad (3.58)$$

The total moments acting in the *body* axes are a combination of the spring stiffness moments and the hub forces acting at the position \underline{R}_h away from the hub. Using a cross product, the total moments caused by the hub forces and moments in the *body* axes can be found as:

$$\underline{M}_b = \underline{M}_h + \underline{R}_h \times \underline{F}_h \quad (3.59)$$

where $\underline{M}_b = [M_{x_b} \ M_{y_b} \ M_{z_b}]^T$, $\underline{M}_h = [M_{x_h} \ M_{y_h} \ M_{z_h}]^T$, and $\underline{F}_h = [F_{x_h} \ F_{y_h} \ F_{z_h}]^T$. The individual *body* axes moments can be written as:

$$M_{x_b} = M_{x_h} + F_{z_h} y_h - F_{y_h} z_h \quad (3.60)$$

$$M_{y_b} = M_{y_h} - F_{z_h} x_h + F_{x_h} z_h \quad (3.61)$$

$$M_{z_b} = F_{y_h} x_h - F_{x_h} y_h \quad (3.62)$$

3.3.5 Flapping Equation

Evaluating the rotor inertia and motion through space, McVicar (1993) derives and thoroughly details a general nonlinear flapping equation for the blades:

$$\begin{aligned} \ddot{\beta} - \omega_{x\beta}\omega_{z\beta}\beta^2 - \left(\omega_{z\beta}\omega_{xc} + \omega_{zc}\omega_{x\beta} + a_{hblz\beta} - \frac{K_\beta}{I_B} \right) \beta \\ = (\omega_{zc}\omega_{xc} + a_{hblzc} - \dot{\omega}_{yc}) + \frac{M_{ybl}}{I_B} \end{aligned} \quad (3.63)$$

where the coefficients are described as:

$$a_{hblzc} = \frac{M\bar{r}_B}{I_B}\dot{w}_h$$

$$a_{hblz\beta} = \frac{M\bar{r}_B}{I_B}(-\dot{u}_h \cos \Psi_h + \dot{v}_h \sin \Psi_h)$$

$$\omega_{xc} = -p \cos \Psi_h + q \sin \Psi_h$$

$$\omega_{x\beta} = -r + \Omega$$

$$\omega_{zc} = r - \Omega$$

$$\omega_{z\beta} = -p \cos \Psi_h + q \sin \Psi_h$$

$$\dot{\omega}_{yc} = -\dot{p} \sin \Psi_h - p\Omega \cos \Psi_h - \dot{q} \cos \Psi_h + \Omega q \sin \Psi_h$$

The terms a_{hblzc} and $a_{hblz\beta}$ model the linear acceleration of the hub and its effect on the blades. The hub acceleration in the z_h axis proportionally correlates into flapping acceleration as the blade mass (M) and center of mass (\bar{r}_B) interact with the blades moment of inertia. The hub's acceleration in the x_h and y_h axes act like spring terms and can be stabilizing or destabilizing depending on the direction of acceleration and azimuth position. If the blade is flapped some amount and the

x_{bl} axis is opposing the direction of acceleration, then the acceleration forces β to 0 and vice versa. The angular velocity terms, ω_{xc} , $\omega_{x\beta}$, ω_{zc} , and $\omega_{z\beta}$, represent the blade's angular velocity with a subscript c denoting constant terms and subscript β denoting terms multiplied by β . The final term $\dot{\omega}_{yc}$ maps the vehicle angular accelerations into blade flapping acceleration.

3.3.6 Azimuth Dynamics

Without lead and lag hinges at the hub, the blade dynamics around the azimuth are straightforward. Assuming the hub rotating assembly has negligible inertia, the azimuth acceleration is given by:

$$\dot{\Omega} = \frac{1}{N_B I_B} \sum_{j=1}^{N_B} [M_{z_{blj}}] \quad (3.64)$$

Rotors spinning clockwise will have negative values of Ω , and with the addition of the N_c term, the moment term, $M_{z_{bl}}$, will correctly track the sign change.

3.3.7 Inflow Model

The Peters-HaQuang dynamic inflow model is used as it provides accurate results with low computation cost (Peters & HaQuang, 1988). The rotor inflow is modeled with a first order system described by Equation (3.69), which operates in the wind axis of the rotor system. Using non-dimensional quantities helps simplify the equations. Velocities have been non-dimensionalized by the rotor tip speed V_T :

$$\mu_h = \frac{u_h}{V_T}, \quad \nu_h = \frac{v_h}{V_T}, \quad \xi_h = \frac{w_h}{V_T} \quad (3.65)$$

The rotor induced inflows ν_0 , ν_s , and ν_c are non-dimensionalized similarly:

$$\lambda_0 = \frac{\nu_0}{V_T}, \quad \lambda_s = \frac{\nu_s}{V_T}, \quad \lambda_c = \frac{\nu_c}{V_T} \quad (3.66)$$

The rotor thrust T is non-dimensionalized by:

$$C_T = \frac{T}{\rho A V_T^2} \quad (3.67)$$

A generic rotor moment J is non-dimensionalized by:

$$C_J = \frac{J}{\rho A V_T^2 R} \quad (3.68)$$

where ρ is the air density, A is the area of the rotor disk, and R is the radius of the rotor. The Peters-HaQuang dynamic inflow model is then given by:

$$\frac{1}{\Omega} M_d \begin{bmatrix} \dot{\lambda}_0 \\ \dot{\lambda}_s \\ \dot{\lambda}_c \end{bmatrix} + \hat{L}^{-1} \begin{bmatrix} \lambda_0 \\ \lambda_s \\ \lambda_c \end{bmatrix} = \begin{bmatrix} C_T \\ C_l \\ -C_M \end{bmatrix}_{aero} \quad (3.69)$$

λ_0 is the uniform normal flow velocity across the disk. λ_s and λ_c are harmonic components which affect blade flapping and the overall aerodynamic system through the dynamic gain matrix \hat{L}^{-1} , which can be found with Equation (3.70). V is the mass flow parameter and must be handled correctly for various operating states. M_d is the apparent mass matrix of the air within the rotor system and allows the acceleration of the air to be modeled. The forcing term comprised of the rotor thrust, rolling, and pitching coefficients (C_T , C_l , and C_M respectively) operate in the *hub* axes and must only include the forces and moments acting on the air. The rotor's pitching moment around the CG of the aircraft will be much different that the pitching moment on the air within the rotor system.

$$\hat{L}^{-1} = V T_w^{hT} \tilde{L}^{-1} T_w^h \quad (3.70)$$

where ${}^\Delta T_w^h$ is rotation to the *wind* axes given by Equation (3.71) and \tilde{L} is the dynamic gain matrix operating in the *wind* axes given by Equation (3.73).

$${}^\Delta T_w^h = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \Psi_w & \sin \Psi_w \\ 0 & -\sin \Psi_w & \cos \Psi_w \end{bmatrix} \quad (3.71)$$

where Ψ_w is the *hub* axis side-slip angle given by:

$$\Psi_w = \tan^{-1} \frac{\nu_h}{\mu_h} \quad (3.72)$$

The *wind* axis dynamic gain matrix is given by:

$$\tilde{L} = \begin{bmatrix} \frac{1}{2} & 0 & -\frac{15\pi}{64} \sqrt{\frac{1-\sin\chi}{1+\sin\chi}} \\ 0 & \frac{4}{1+\sin\chi} & 0 \\ \frac{15\pi}{64} \sqrt{\frac{1-\sin\chi}{1+\sin\chi}} & 0 & \frac{4\sin\chi}{1+\sin\chi} \end{bmatrix} \quad (3.73)$$

where χ is the wake skew angle given by:

$$\chi = \tan^{-1} \frac{|\lambda_m - \xi_h|}{\sqrt{\mu_h^2 + \nu_h^2}} \quad (3.74)$$

λ_m is given by Equation (3.78). Early inflow models like Glauert's assume the air accelerates instantaneously and do not produce accurate transient effects for simulation and control design (Wheatley, 1935). The Peters-HaQuang model utilizes the apparent mass matrix to simulate the acceleration of the air. This acceleration phase causes a spike in rotor thrust as the inflow has not yet decreased the local angle of attack on the blades. The M_{d11} element was taken to be $\frac{8}{3\pi}$, which produces results closer to flight test data instead of $\frac{128}{75\pi}$ (Chen, 1989).

$$M_d = \begin{bmatrix} \frac{8}{3\pi} & 0 & 0 \\ 0 & \frac{16}{45\pi} & 0 \\ 0 & 0 & \frac{16}{45\pi} \end{bmatrix} \quad (3.75)$$

The mass flow parameter V is given by:

$$V = \begin{bmatrix} V_t & 0 & 0 \\ 0 & V_m & 0 \\ 0 & 0 & V_m \end{bmatrix} \quad (3.76)$$

where V_t is the total flow through the rotor disk and V_m relates the normal and harmonic components. V_t is given by:

$$V_t = \sqrt{\mu_h^2 + \nu_h^2 + (\lambda_m - \xi_h)^2} \quad (3.77)$$

where λ_m is the total resultant normal inflow through the rotor disk, which produces thrust given by:

$$\lambda_m = \frac{1}{2} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}^T T_w^h{}^T L^{-1} T_w^h \begin{bmatrix} \lambda_0 \\ \lambda_s \\ \lambda_c \end{bmatrix} \quad (3.78)$$

The dynamic inflow model is generally used for conventional operating states where the flow through the disk passes in the z_h direction. In the windmill-brake state, the flow passes through the rotor in the $-z_h$ direction and requires a modification to the mass flow parameter. Murakami and Houston derived a correction shown in Equation (3.79) (Murakami & Houston, 2009).

$$V_m = \frac{\mu_h^2 + \nu_h^2 + (\lambda_m - \xi_h)^2 + \lambda_m |\lambda_m - \xi_h|}{\sqrt{\mu_h^2 + \nu_h^2 + (\lambda_m - \xi_h)^2}} \quad (3.79)$$

In the static case, which can be used for simple models, the inflow components are given by:

$$\bar{\lambda}_m = \bar{\lambda}_0 = \frac{C_T}{2V_T} \quad (3.80)$$

$$\bar{\lambda}_s = \frac{15\pi}{32} \sqrt{\frac{1 - \sin \chi}{1 + \sin \chi}} \sin \beta_h \bar{\lambda}_m \quad (3.81)$$

$$\bar{\lambda}_c = \frac{15\pi}{32} \sqrt{\frac{1 - \sin \chi}{1 + \sin \chi}} \cos \beta_h \bar{\lambda}_m \quad (3.82)$$

Solving for $\bar{\lambda}_m$ and $\bar{\lambda}_0$ requires an iterative process because C_T is a function of the inflow along with V_T . Using a fixed point iteration with the following scheme is

robust enough for most flight conditions.

$$\tilde{\lambda}_{k+1} = \frac{C_T(\tilde{\lambda}_k)}{2V_T(\tilde{\lambda}_k)} \quad (3.83)$$

where $\tilde{\lambda}_k$ is the iterative guess at some step k with the initial guess $\tilde{\lambda}_0$ given by the hover induced velocity:

$$\tilde{\lambda}_0 = \sqrt{\frac{\bar{C}_T}{2}} \quad (3.84)$$

Here \bar{C}_T is assumed to not be a function of the inflow and rather a static value that is thrust accounting for half of the vehicle weight, non-dimensionalized by Equation (3.67). Using Equation (3.84) to start the simulation and then feeding the converged solution of $\tilde{\lambda}$ to the next step of the simulation will increase the robustness of the solution scheme.

3.4 Control Blending

The control inputs used for the controller are the same as a pilot would have for simplicity. Each rotor has 3 controls: collective, longitudinal cyclic, and lateral cyclic, and there are 4 controls available to the pilot: collective (θ_0), longitudinal stick (θ_c), lateral stick (θ_s), and rudder (θ_{rud}). The control inputs are mapped to the rotors by:

$$\hat{\theta}_{0l} = \frac{1}{2} (\theta_0 + D_{\theta_0} \theta_s) \quad (3.85)$$

$$\hat{\theta}_{cl} = \frac{1}{2} (1 - D_{\theta_0}) \theta_s \quad (3.86)$$

$$\hat{\theta}_{sl} = -\frac{1}{2} (\theta_c - \theta_{rud}) \quad (3.87)$$

$$\hat{\theta}_{0r} = \frac{1}{2} (\theta_0 - D_{\theta_0} \theta_s) \quad (3.88)$$

$$\hat{\theta}_{cr} = -\frac{1}{2} (1 - D_{\theta_0}) \theta_s \quad (3.89)$$

$$\hat{\theta}_{sr} = \frac{1}{2} (\theta_c + \theta_{rud}) \quad (3.90)$$

where the subscripts l and r are added to denote the left and right rotor, respectively, and D_{θ_0} is used to linearly vary the use of differential collective or combined cyclic for roll control. If D_{θ_0} is set to 1, then only differential collective (DC) will be used, and if D_{θ_0} is set to 0, only combined cyclic (CC) will be used. During AR, the use of collective causes large changes in rotor speed, but using DC creates large rolling moments. Smaller rolling moments are created by the use of CC, but fast maneuvers in the lateral-directional plane will not be necessary for landing.

4. Controller Design

Controlling a plant along a specified trajectory is the overlying goal for any controls problem. Different control theories have been developed over the past century, each with their own benefits and pitfalls. Optimal control offers a unique characteristic in that its purpose is to minimize a cost using the inputs to the system. Depending on the plant and situation, the cost can weigh system states and inputs differently yielding varying ranges of controller performance.

4.1 Optimal Control Techniques

On an infinite horizon, standard optimal control techniques like the Linear Quadratic Regulator entail solving the algebraic Riccati equation to provide the optimal control input for a given reference trajectory. However, infinite horizon techniques lack robustness when the plant is subject to disturbances and unmodeled dynamics. Work has been done to extend the LQR's capabilities by adding disturbance rejection and applying it in a sub-optimal way to nonlinear systems. Gao et al. (2017) added a disturbance estimation component to the LQR by assuming the disturbance becomes constant in the infinite horizon. Wernli and Cook (1975) applied LQR to a nonlinear system in a sub-optimal manner by linearizing the system at instances in time and applying optimal linear inputs. The sub-optimal implementation relies on the plant being stable within a region around an operating point. While these methods provide a means of improving the robustness of the LQR, their operation on a infinite horizon enforces undesirable assumptions.

Model predictive control is another form of optimal control which that can be applied to both linear and nonlinear systems. Model predictive control predicts the model trajectory over a finite horizon, optimizes the input over that period of time, implements some portion of the control input, and repeats the process. Increasing the speed of the algorithm can be accomplished by implementing larger portions of

the computed control input or by assuming the control input can be expressed in terms of a set of basis functions and solving for the basic coefficients.

4.2 Reduced Order Model

Model predictive control relies on predicting the future states of the vehicle for control input optimization. The reduced order model (ROM) used for prediction must be simple enough to run quickly over the finite horizon time as the optimization requires iterating and performing a gradient descent within the solution space, but the model must also retain key dynamics so the states are accurately predicted. The reduced order TR model relies on quasi-steady disk flapping equations and the Peters & HaQuang inflow model, which allows the simulation time step to increase one order of magnitude over the individual blade model simulation.

4.2.1 Rotor System

Padfield (2018) derives a set of quasi-steady flapping equations that can be used for vehicle simulation given the dynamics of the overall vehicle are significantly slower than the rotor itself. The XV-15, with two large masses at the ends of its wings and a large tail, has high rolling and yawing inertia along with large amounts of aerodynamic damping. This yields an effective low pass filter that is mostly unaffected by the quick rotor dynamics.

4.2.1.1 Multi-Blade Coordinates

Resolving the individual flapping blades to a multi-blade coordinate disk simplifies the modeling task. In the case of a 3-bladed rotor, the individual blades create a 3 degree of freedom (DOF) system to model the flapping motion of each blade. Individual blade flapping can be resolved into a 3 DOF disk: coning (β_0), longitudinal tilt (β_c), and lateral tilt (β_s) (Padfield, 2018). The transformation to

multi-blade coordinates is performed as follows:

$$\underline{\beta} = \begin{bmatrix} \beta_0 \\ \beta_c \\ \beta_s \end{bmatrix} = \begin{bmatrix} \frac{1}{N_b} \sum_{j=1}^{N_b} [\beta_j] \\ \frac{2}{N_b} \sum_{j=1}^{N_b} [\beta_j \cos \Psi_{hj}] \\ \frac{2}{N_b} \sum_{j=1}^{N_b} [\beta_j \sin \Psi_{hj}] \end{bmatrix} \quad (4.1)$$

4.2.1.2 Multi-Blade Flapping Equations

Padfield derives the multi-blade coordinate flapping equation assuming steady 2-D aerodynamics with linear lift and quadratic drag curves (Padfield, 2018).

Reversed flow effects are ignored and the inflow is assumed to be uniform in the spanwise direction, meaning the blades have a hyperbolic twist profile. However, in the derivation, a linear twist profile is used which further simplifies the results. The largest difference between hyperbolic and linear twist profiles occurs at the root of the blade where aerodynamic forces are low. The blade pitch is written as:

$$\tilde{\theta} = \tilde{\theta}_0 + \tilde{r}_b \tilde{\theta}_{tw} \quad (4.2)$$

where $\tilde{\theta}$ is the continuous blade pitch profile, $\tilde{\theta}_0$ is the total collective pitch, \tilde{r}_b is the nondimensionalized blade position, and $\tilde{\theta}_{tw}$ is the total blade twist from root to tip. The total collective pitch is a combination of inputted collective and the existing blade twist:

$$\tilde{\theta}_0 = \theta_{root} + \hat{\theta}_0 \quad (4.3)$$

where θ_{root} is the blade pitch at the root. The harmonic inputs $\tilde{\theta}_s$ and $\tilde{\theta}_c$ are equal to the simulation inputs $\hat{\theta}_s$ and $\hat{\theta}_c$ but a distinction is made for clarity.

These assumptions allows analytical expressions for the rotor aerodynamics to be derived and yield a linear blade flapping equation given by:

$$\underline{\beta}_w'' + C_{M0} \underline{\beta}_w' + D_{M0} \underline{\beta}_w = H_{M0} \quad (4.4)$$

where $\underline{\beta}_w''$ is the second derivative of the blade flapping with respect to the rotor

azimuth angle in the *wind* axes. The matrix coefficients are given by:

$$C_{M0} = \frac{\gamma}{8} \begin{bmatrix} 1 & 0 & \frac{2}{3}\mu_w \\ 0 & 1 & \frac{16}{\gamma} \\ \frac{4}{3}\mu_w & -\frac{16}{\gamma} & 1 \end{bmatrix}$$

$$D_{M0} = \frac{\gamma}{8} \begin{bmatrix} \frac{8\lambda_\beta^2}{\gamma} & 0 & 0 \\ \frac{4}{3}\mu_w & S_\beta & 1 + \frac{\mu_w^2}{2} \\ 0 & \frac{\mu_w^2}{2} - 1 & S_\beta \end{bmatrix}$$

$$H_{M0} = \frac{\gamma}{8} \begin{bmatrix} \tilde{\theta}_0 (1 + \mu_w) + 4\tilde{\theta}_{tw} \left(\frac{1}{5} + \frac{\mu_w^2}{6} \right) + \frac{4}{3}\mu_w \tilde{\theta}_{sw} + \frac{4}{3} (\xi_h - \lambda_0) + \frac{2}{3}\mu_w (\bar{p}_w - \lambda_{sw}) \\ \frac{16}{\gamma} (\bar{p}_w + \frac{\bar{q}_w'}{2}) + \tilde{\theta}_{cw} \left(1 + \frac{\mu_w^2}{2} \right) + (\bar{q}_w - \lambda_{cw}) \\ -\frac{16}{\gamma} (\bar{q}_w - \frac{\bar{p}_w'}{2}) + \frac{8}{3}\mu_w \tilde{\theta}_0 + 2\mu_w \tilde{\theta}_{tw} + \tilde{\theta}_{sw} (1 + \frac{3}{2}\mu_w^2) + 2\mu_w (\xi_h - \lambda_0) + (\bar{p}_w - \lambda_{sw}) \end{bmatrix}$$

The lock number (γ) of a rotor blade describes the relation between aerodynamic and inertial forces and is found with:

$$\gamma = \frac{\rho \bar{c} a_0 R^4}{I_B} \quad (4.5)$$

where \bar{c} is the average chord along the blade and a_0 is the blade's linear variation of lift with angle of attack. Higher lock numbers infer the blade will experience larger motions when perturbed. The stiffness number (S_β) explains the hub, spring, and blade interaction:

$$S_\beta = \frac{8 (\lambda_\beta^2 - 1)}{\gamma} \quad (4.6)$$

Higher stiffness numbers imply that the equivalent spring forces take larger effect on the blade's motion and tend to drive the phase lag below 90° . The AugustaWestland Lynx operates with a rigidly mounted rotor boasting a high stiffness number that pushes the phase lag to around 80° (Padfield, 2018).

The wind axes advance ratio (μ_w) is the magnitude of the *hub* axes velocity normalized by the rotor tip speed, or written in terms of the *hub* advance ratios:

$$\mu_w = \sqrt{\mu_h^2 + \nu_h^2} \quad (4.7)$$

Deriving rotor dynamics with respect to rotor azimuth angle offers simple periodic solutions of the motion. The aerodynamics of a rotor blade in forward flight vary significantly with respect to their azimuth position, and generally, rotor systems are governed to operate at a constant rotational speed, making it an organic normalization term. For simulation, derivatives with respect to azimuth can be converted to time derivatives with the chain rule:

$$\frac{d\underline{\beta}}{dt} = \frac{d\Psi_h}{dt} \frac{d\underline{\beta}}{d\Psi_h} \quad (4.8)$$

with familiar terms:

$$\dot{\underline{\beta}} = \Omega \underline{\beta}' \quad (4.9)$$

Similarly, second derivatives can be computed as:

$$\ddot{\underline{\beta}} = \Omega^2 \underline{\beta}'' \quad (4.10)$$

Transforming the individual rotor blades to a disk enables some simplifications. The rotor dynamics operate at a significantly higher frequency than the total vehicle, which leads to quasi-static rotor flapping (Padfield, 2018). Setting $\underline{\beta}'' = \underline{\beta}' = 0$ turns Equation (4.4) into a linear algebraic flapping equation. Since the flapping is considered as a disk, $\dot{\underline{\beta}}$ does not correlate to a single blade flapping up and down as it moves around the azimuth, but instead describes the average motion of all the blades. The aerodynamics of each blade, while assumed to be steady, linear, and 2-D, are still well modeled with respect to azimuth angle. The quasi-static flapping equation is written as:

$$\underline{\beta}_w = A_{\beta\theta}\underline{\theta}_w + A_{\beta\lambda}\underline{\lambda}_w + A_{\beta\omega}\underline{\omega}_w \quad (4.11)$$

where the *wind* axes inflow vector ($\underline{\lambda}_w$), control vector ($\underline{\theta}_w$), and angular velocity

vector $(\underline{\omega}_w)$ are formed as:

$$\lambda_w = \begin{bmatrix} \xi_h - \lambda_0 \\ \lambda_{sw} \\ \lambda_{cw} \end{bmatrix} \quad \underline{\theta}_w = \begin{bmatrix} \tilde{\theta}_0 \\ \tilde{\theta}_{tw} \\ \tilde{\theta}_{sw} \\ \tilde{\theta}_{cw} \end{bmatrix} \quad \underline{\omega}_w = \begin{bmatrix} \bar{p}'_w \\ \bar{q}'_w \\ \bar{p}_w \\ \bar{q}_w \end{bmatrix} \quad (4.12)$$

with an overhead bar denoting normalization with respect to rotor speed. A common term (η_β) appears throughout the matrix coefficients and is given as a function of the stiffness number:

$$\eta_\beta = -\frac{1}{1 + S_\beta^2} \quad (4.13)$$

The control vector matrix coefficient is given as:

$$A_{\beta\theta} = \frac{\gamma}{8\lambda_\beta^2} \begin{bmatrix} a_{11} & a_{12} & a_{13} & 0 \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{bmatrix} \quad (4.14)$$

$$a_{11} = 1 + \mu_w^2$$

$$a_{12} = \frac{4}{5} + \frac{2}{3}\mu_w^2$$

$$a_{13} = \frac{4}{3}\mu_w$$

$$a_{21} = \eta_\beta \frac{4}{3}\mu_w \left(S_\beta (1 + \mu_w^2) + \frac{16\lambda_\beta^2}{\gamma} \left(1 + \frac{\mu_w^2}{2} \right) \right)$$

$$a_{22} = \eta_\beta 2\mu_w \left(\frac{8\lambda_\beta^2}{\gamma} \left(1 + \frac{\mu_w^2}{2} \right) + \frac{8}{15} S_\beta \left(1 + \frac{5}{2}\mu_w^2 \right) \right)$$

$$a_{23} = \eta_\beta \left(\frac{8\lambda_\beta^2}{\gamma} (1 + 2\mu_w^2) + \left(\frac{4}{3}\mu_w \right)^2 S_\beta \right)$$

$$a_{24} = -N_{cl}\eta_\beta S_\beta \frac{8\lambda_\beta^2}{\gamma} \left(1 + \frac{\mu_w^2}{2}\right)$$

$$a_{31} = N_{cl}\eta_\beta \frac{4}{3}\mu_w \left(1 + \frac{\mu_w^2}{2} - S_\beta \frac{16\lambda_\beta^2}{\gamma}\right)$$

$$a_{32} = N_{cl}\eta_\beta 2\mu_w \left(\frac{8}{15} \left(1 + \frac{\mu_w^2}{2}\right) - S_\beta \frac{8\lambda_\beta^2}{\gamma}\right)$$

$$a_{33} = N_{cl}\eta_\beta \left(\left(\frac{4}{3}\mu_w\right)^2 - S_\beta \frac{8\lambda_\beta^2}{\gamma} \left(1 + \frac{3}{2}\mu_w^2\right)\right)$$

$$a_{34} = -\eta_\beta \frac{8\lambda_\beta^2}{\gamma} \left(1 - \frac{\mu_w^4}{2}\right)$$

The inflow matrix coefficient is given by:

$$A_{\beta\lambda} = \frac{\gamma}{8\lambda_\beta^2} \begin{bmatrix} b_{11} & b_{12} & 0 \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} \quad (4.15)$$

$$b_{11} = \frac{4}{3}$$

$$b_{12} = -\frac{2}{3}\mu_w$$

$$b_{21} = \eta_\beta \mu_w \left(\left(\frac{4}{3}\right)^2 S_\beta + \frac{16\lambda_\beta^2}{\gamma} \left(1 + \frac{\mu_w^2}{2}\right)\right)$$

$$b_{22} = -N_{cl}\eta_\beta \left(\frac{8\lambda_\beta^2}{\gamma} \left(1 + \frac{\mu_w^2}{2}\right) + \frac{S_\beta}{2} \left(\frac{4}{3}\mu_w\right)^2\right)$$

$$b_{23} = \eta_\beta \frac{8\lambda_\beta^2}{\gamma} S_\beta$$

$$b_{31} = N_{cl}\eta_\beta\mu_w \left(\left(\frac{4}{3} \right)^2 \left(1 - \frac{\mu_w^2}{2} \right) - S_\beta \frac{16\lambda_\beta^2}{\gamma} \right)$$

$$b_{32} = \eta_\beta \left(\frac{8\lambda_\beta^2}{\gamma} S_\beta - \frac{1}{2} \left(\frac{4}{3} \mu_w \right)^2 \right)$$

$$b_{33} = N_{cl}\eta_\beta \frac{8\lambda_\beta^2}{\gamma} \left(1 - \frac{\mu_w^2}{2} \right)$$

Lastly, the angular velocity matrix coefficient is:

$$A_{\beta\omega} = \frac{\gamma}{8\lambda_\beta^2} \begin{bmatrix} 0 & 0 & c_{13} & 0 \\ c_{21} & c_{22} & c_{23} & c_{24} \\ c_{31} & c_{32} & c_{33} & c_{34} \end{bmatrix} \quad (4.16)$$

$$c_{13} = \frac{2}{3}\mu_w$$

$$c_{21} = N_{cl}\eta_\beta \left(\frac{8\lambda_\beta}{\gamma} \right)^2 \left(1 + \frac{\mu_w^2}{2} \right)$$

$$c_{22} = -\eta_\beta \left(\frac{8\lambda_\beta}{\gamma} \right)^2 S_\beta$$

$$c_{23} = N_{cl}\eta_\beta \left(\frac{8\lambda_\beta^2}{\gamma} \left(1 + \frac{\mu_w^2}{2} - \frac{16S_\beta}{\gamma} \right) + \frac{S_\beta}{2} \left(\frac{4}{3} \mu_w \right)^2 \right)$$

$$c_{24} = -\eta_\beta \frac{8\lambda_\beta^2}{\gamma} \left(S_\beta + \frac{16}{\gamma} \left(1 + \frac{\mu_w^2}{2} \right) \right)$$

$$c_{31} = -\eta_\beta S_\beta \left(\frac{8\lambda_\beta}{\gamma} \right)^2$$

$$c_{32} = N_{cl}\eta_\beta \left(\frac{8\lambda_\beta}{\gamma} \right)^2 \left(\frac{\mu_w^2}{2} - 1 \right)$$

$$c_{33} = \eta_\beta \frac{8\lambda_\beta^2}{\gamma} \left(\frac{16}{\gamma} \left(\frac{\mu_w^2}{2} - 1 \right) - S_\beta \right) + \frac{1}{2} \left(\frac{4}{3} \mu_w \right)^2$$

$$c_{34} = N_{cl} \eta_\beta \frac{8\lambda_\beta^2}{\gamma} \left(\frac{16S_\beta}{\gamma} + \frac{\mu_w^2}{2} - 1 \right)$$

where N_{cl} , defined in Equation (3.40), adjusts for the direction of rotation.

The quasi-static flapping equations allow the reduced order model to run at time steps an order of magnitude higher than the fully nonlinear simulation. Changing the flapping dynamics description from the second order equation (3.63) to the quasi-static linear flapping equation (4.11) reduces the number of states within the model by 12 and removes high frequency blade motion.

Assuming that the higher order dynamics have decayed, the individual blade model is dominated by one per revolution motion of three blades, each offset by 120° , which resolves the forces to a three per revolution cycle. One revolution of the XV-15 rotor system takes about 0.1 seconds, which produces forces in a periodic fashion at 30Hz. Therefore, the simulation time steps need to be at least 60Hz, accounting for Nyquist sampling frequency (Ayanoglu, 1997). Realistically, higher order blade dynamics will appear in simulation, up to three per revolution of each blade, so the sampling frequency is increased to 180Hz.

The reduced order rotor model, while offering benefits in computation cost, does not correctly model coupling between yaw and roll motions. Derived for a conventional helicopter, the effect of yaw rate on flapping is neglected as the rotor spins much faster than the fuselage (or, at least, hopefully it does). A tilt-rotor's rotor system experiences large motion from the *body* axes angular rates. With a positive yaw rate, the left rotor hub has a positive u_h velocity and the right hub has a negative u_h velocity of equal magnitude. Equation (4.11) only considers these two values, so the effect of yaw on the two rotor systems is equivalent but opposite.

In the individual blade rotor system, the yaw rate is fed directly into the blade

angular motion. A blade on the left rotor, when sweeping around the outside of the rotor, will see a substantial increase in velocity from a positive yaw rate because it is far away from the vehicle's CG. When the same blade sweeps around the inside of the disk, yaw rate has a lesser negative effect on the blade velocity since the distance to the CG has decreased. The opposite is true for the right rotor. A blade sweeping around the outside of the right rotor disk will see a large decrease in velocity and a small increase once it has reached the inside of the disk. Overall, the effect causes the left rotor to produce more thrust since its blades have a higher average velocity than the right side. This causes a positive rolling moment.

4.2.1.3 Wind Axes Transformations

The multi-blade coordinate flapping equations operate within the *wind* axes. The x_w axis points into the incoming flow at an angle of Ψ_w from the x_h axis. Acting as a standard right-handed coordinate system, the y_w axis remains perpendicular to the incoming air and the z_w axis is aligned with z_h . Flapping angles in the *wind* axes can be transformed to the *hub* axes using:

$$\underline{\beta}T_w^h = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \Psi_w & \sin \Psi_w \\ 0 & -\sin \Psi_w & \cos \Psi_w \end{bmatrix} \quad (4.17)$$

The rotor coning angle (β_0) is constant between the *hub* and *wind* axes. The rotation matrices $\underline{\beta}T_w^h$ and $\underline{\lambda}T_h^w$ are equal. The flapping angle β_c correlates to a rotation around the y_h axis and β_s around the x_h , while the inflow vector follows the standard convention of x_h then y_h with λ_s and λ_c , respectively. A distinction is maintained for clarity and later analysis. The control vector (θ) can be transferred

to the *wind* axes using:

$${}^{\theta}T_h^w = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \cos \Psi_w & -\sin \Psi_w \\ 0 & 0 & \sin \Psi_w & \cos \Psi_w \end{bmatrix} \quad (4.18)$$

The inflow vector is transformed to the *wind* axes using Equation (3.71) and the angular velocity vector is transformed using similar techniques:

$${}^{\omega}T_h^w = \begin{bmatrix} \cos \Psi_w & -\sin \Psi_w & 0 & 0 \\ \sin \Psi_w & \cos \Psi_w & 0 & 0 \\ 0 & 0 & \cos \Psi_w & -\sin \Psi_w \\ 0 & 0 & \sin \Psi_w & \cos \Psi_w \end{bmatrix} \quad (4.19)$$

4.2.1.4 Forces and Moments

The total rotor force acts nearly perpendicular to the disk, which tilts to create control forces and moments (Padfield, 2018). Using the same assumptions put in place for the multi-blade flapping equations, the total rotor force can be found to the zeroth harmonic as:

$$C_T = \frac{a_0 s}{2} \left[\hat{\theta}_0 \left(\frac{1}{3} + \frac{\mu_w^2}{2} \right) + \frac{\mu_w}{2} \left(\hat{\theta}_{sw} + \frac{\bar{p}_w}{2} \right) + \left(\frac{\xi_h - \lambda_0}{2} \right) + \frac{1}{4} (1 + \mu_w^2) \hat{\theta}_{tw} \right] \quad (4.20)$$

where s denotes the rotor solidity. Solidity is the ratio of the total blade area to the total disk area. Typical rotor systems have solidity ratios from 0.05 - 0.1 (Leishman, 2006). Increasing the solidity decreases the average blade angle of attack for a given thrust but also increases the parasitic drag. Solidity can be approximated with:

$$s = \frac{N_b \bar{c}}{\pi R} \quad (4.21)$$

Using the *hub* axes flapping angles, the x_h and y_h force coefficients can be written as:

$$C_{x_h} = C_T \beta_c \quad (4.22)$$

$$C_{y_h} = -C_T \beta_s \quad (4.23)$$

Moments passing through the rotor hub are a sole product of the hub spring stiffness and can be computed as follows:

$$M_{x_h} = -\frac{N_b}{2} K_\beta \beta_s \quad (4.24)$$

$$M_{y_h} = -\frac{N_b}{2} K_\beta \beta_c \quad (4.25)$$

The total rotor moments acting in the *body* axes can be computed with (3.59).

The rotor torque must accurately predict the control and state effects on the rotor speed. Any inaccuracies within the azimuth dynamics of the reduced order model could lead to catastrophic failures during AR. Padfield derives the rotor torque coefficient to be (Padfield, 2018):

$$C_Q = -C_T (\xi_h - \lambda_0) + \mu_w C_{x_w} + \frac{\bar{C}_d s}{8} (1 + 3\mu_w^2) \quad (4.26)$$

where C_Q is the rotor torque coefficient, C_{x_w} is the x_w force coefficient, and \bar{C}_d is the average coefficient of drag. Padfield derives Equation (4.26) with the coefficient of drag as a quadratic function with respect to C_T , helping account for induced drag, but \bar{C}_d is kept as a constant value. The inflow term $(\xi_h - \lambda_0)$ is negative in normal working states and shows that increasing thrust increases the rotor torque. That is:

$$\frac{\partial C_Q}{\partial C_T} = -(\xi_h - \lambda_0) \quad (4.27)$$

This relationship fails when the flow direction reverses as $(\xi_h - \lambda_0)$ becomes positive. An increase in C_T should always cause an increase in rotor torque, regardless of the inflow condition. Padfield assumes a uniform inflow distribution and linear blade aerodynamics which, in the case of AR, is not completely accurate. Essentially, the magnitude of the inflow term is most important, so modifying Equation (4.28) to

$$\frac{\partial C_Q}{\partial C_T} = \text{sig}(\xi_h - \lambda_0) (\xi_h - \lambda_0) \quad (4.28)$$

keeps the derivative positive and haven been proven to provide accurate torque predictions for the controller. The sigmoid function ($\text{sig}(\cdot)$) is scaled and shifted to produced values between $(-1, 1)$ and is defined as:

$$\text{sig}(\tau) = \frac{2}{1 + e^{-10000\tau}} - 1 \quad (4.29)$$

where τ is a generic value. The sigmoid function essentially acts as a sign function but is differentiable at zero as required by the optimization algorithm. Rotor forces and moments in the *hub* or *wind* axes are easily found using their respective flapping angles:

$$C_{x_w} = C_T \beta_{cw} \quad (4.30)$$

4.2.1.5 Inflow

The Peters and HaQuang inflow model is used within the ROM. Using the correction described in Equation (3.79) creates an issue in that again the derivative of the absolute value function is not defined at 0. Rewriting the absolute value term in the harmonic inflow equation as

$$|\lambda_m - \xi_h| = \text{sign}(\lambda_m - \xi_h) (\lambda_m - \xi_h) \quad (4.31)$$

allows an approximation using Equation (4.29). The harmonic inflow equation used within the ROM is then:

$$V_m = \frac{\mu_h^2 + \nu_h^2 + (\lambda_m - \xi_h)^2 + \lambda_m \operatorname{sig}(\lambda_m - \xi_h)(\lambda_m - \xi_h)}{\sqrt{\mu_h^2 + \nu_h^2 + (\lambda_m - \xi_h)^2}} \quad (4.32)$$

4.3 Optimization

Predicting the vehicle states over a finite horizon of discrete time steps (N_s) allows a nonlinear optimization algorithm to minimize a cost function using the control inputs. Vehicle states are desired to follow a reference trajectory or end at a specific terminal state; including the vehicle's dynamics within the optimization allows the controller to find a realizable trajectory. The problem is posed as:

$$\min L(X(U), U) \quad \text{s.t.} \quad C(X(U), U) \leq 0 \quad (4.33)$$

where X and U are time histories of the states and controls, respectively, over the receding horizon. The cost function (L) can penalize states individually by adding additional cost to some and/or completely neglecting others. Increasing the cost of a specific state creates a soft constraint and emphasizes its importance. Hard constraints (C) can be used within the optimization as either equality or inequality functions that describe surfaces or firm boundaries within the solution space where the optimal solution must exist. If the optimal solution falls on a constraint surface or boundary that specific constraint is known as being active. The constraint is inactive if the optimal solution does not fall on the surface or boundary.

4.3.1 Cost Function

A quadratic cost function is radially unbounded, leading to its gradient pointing in the direction of the origin. This implies that gradient descent (GD) will then force solutions to a minimum. In the case of reference tracking, the objective is to force the tracking error to 0. The tracking error at some time step (\underline{e}_k) can be

defined as:

$$\underline{e}_k = \underline{x}_k - \underline{x}_k^* \quad (4.34)$$

where $\underline{x}_k \in \mathbb{R}^n$ is the current state vector and $\underline{x}_k^* \in \mathbb{R}^n$ is the current reference state.

The quadratic cost function associated with the error can then be written as the sum of a terminal and running cost:

$$L = \frac{1}{2} \underline{e}_{N_s}^T S \underline{e}_{N_s} + \frac{1}{2} \sum_{k=1}^{N_s-1} [\underline{e}_k^T Q \underline{e}_k + \underline{u}_k^T R \underline{u}_k] \quad (4.35)$$

where $S \in \mathbb{R}^{n \times n}$ and $Q \in \mathbb{R}^{n \times n}$ are symmetric positive semidefinite (PSD), and $R \in \mathbb{R}^{m \times m}$ is a symmetric positive definite (PD) matrix, where m represents the number of control inputs. Performing GD relies on the first derivative of the cost function with respect to the matrix of control inputs. Analyzing the derivative at each time step and assembling those into a matrix will allow the optimization algorithm to find a solution. Taking the derivative of Equation (4.35) with respect to a single time step of control inputs (\underline{u}_i) yields:

$$\frac{\partial L}{\partial \underline{u}_i} = \underline{e}_{N_s}^T S \frac{\partial \underline{e}_{N_s}}{\partial \underline{u}_i} + \sum_{k=1}^{N_s-1} \left[\underline{e}_k^T Q \frac{\partial \underline{e}_k}{\partial \underline{u}_i} + \underline{u}_k^T R \frac{\partial \underline{u}_k}{\partial \underline{u}_i} \right] \quad (4.36)$$

where $i = 1, 2, \dots, N_s - 1$ and denotes a specific time unconnected to the summing variable k . The reference trajectory does not depend on the control or vehicle states so its derivative can be equated as:

$$\frac{\partial \underline{e}_k}{\partial \underline{u}_i} = \frac{\partial \underline{x}_k}{\partial \underline{u}_i} \quad (4.37)$$

Equation (4.36) then simplifies to:

$$\frac{\partial L}{\partial \underline{u}_i} = \underline{e}_{N_s}^T S \frac{\partial \underline{x}_{N_s}}{\partial \underline{u}_i} + \sum_{k=1}^{N_s-1} \left[\underline{e}_k^T Q \frac{\partial \underline{x}_k}{\partial \underline{u}_i} + \underline{u}_k^T R \frac{\partial \underline{u}_k}{\partial \underline{u}_i} \right] \quad (4.38)$$

The derivatives can then be assembled into a $m \times N_s - 1$ dimensional matrix:

$$\frac{\partial L}{\partial \underline{U}} = \left[\frac{\partial L}{\partial \underline{u}_1}^T \quad \frac{\partial L}{\partial \underline{u}_2}^T \quad \dots \quad \frac{\partial L}{\partial \underline{u}_{N_s-1}}^T \right] \quad (4.39)$$

with U representing the time history matrix of control inputs:

$$U = [\underline{u}_1 \ \underline{u}_2 \ \cdots \ \underline{u}_{N_s-1}] \quad (4.40)$$

4.3.2 Prediction

Prediction of the vehicle states are included within the optimization through their dependence on the control inputs. The nonlinear vehicle dynamics can be described by the function:

$$\dot{\underline{x}} = f(\underline{x}, \underline{u}) \quad (4.41)$$

Discretizing Equation (4.41) describes a following time step as:

$$\underline{x}_{k+1} = \underline{x}_k + t_s f(\underline{x}_k, \underline{u}_k) \quad (4.42)$$

where t_s is the discrete time step. Taking the derivative of Equation (4.42) with respect to \underline{u}_i gives:

$$\frac{\partial \underline{x}_{k+1}}{\partial \underline{u}_i} = \frac{\partial \underline{x}_k}{\partial \underline{u}_i} + t_s \frac{\partial f}{\partial \underline{u}_i}(\underline{x}_k, \underline{u}_k) \quad (4.43)$$

Expanding with the chain and product rules:

$$\frac{\partial \underline{x}_{k+1}}{\partial \underline{u}_i} = \frac{\partial \underline{x}_k}{\partial \underline{u}_i} + t_s \frac{\partial f}{\partial \underline{x}_k}(\underline{x}_k, \underline{u}_k) \frac{\partial \underline{x}_k}{\partial \underline{u}_i} + t_s \frac{\partial f}{\partial \underline{u}_k}(\underline{x}_k, \underline{u}_k) \frac{\partial \underline{u}_k}{\partial \underline{u}_i} \quad (4.44)$$

Grouping terms, the derivative can be written as:

$$\frac{\partial \underline{x}_{k+1}}{\partial \underline{u}_i} = \left[I + t_s \frac{\partial f}{\partial \underline{x}_k}(\underline{x}_k, \underline{u}_k) \right] \frac{\partial \underline{x}_k}{\partial \underline{u}_i} + t_s \frac{\partial f}{\partial \underline{u}_k}(\underline{x}_k, \underline{u}_k) \frac{\partial \underline{u}_k}{\partial \underline{u}_i} \quad (4.45)$$

Dalamagkidis (2009) describes logic that correctly applies Equation (4.45) to the optimization problem. First, control inputs cannot affect the current or past states:

$$\frac{\partial \underline{x}_k}{\partial \underline{u}_i} = 0 \ \forall \ (\underline{x}_k, \underline{u}_i) : i \geq k \quad (4.46)$$

Second, control inputs do not depend on other control inputs:

$$\frac{\partial \underline{u}_k}{\partial \underline{u}_i} = \begin{cases} I, & i = k \\ 0, & \text{otherwise} \end{cases} \quad (4.47)$$

Using Eqs. (4.46) and (4.47), the proceeding states dependency on the current control input is given by:

$$\frac{\partial \underline{x}_{k+1}}{\partial \underline{u}_i} = \begin{cases} 0, & k < i \\ t_s \frac{\partial f}{\partial \underline{u}_k}(\underline{x}_k, \underline{u}_k), & k = i \\ \left[I + t_s \frac{\partial f}{\partial \underline{x}_k}(\underline{x}_k, \underline{u}_k) \right] \frac{\partial \underline{x}_k}{\partial \underline{u}_i}, & \text{otherwise} \end{cases} \quad (4.48)$$

4.3.3 Constraints

Constraining the optimization problem can capture hard physical or safety limits of the system and ensure they are not exceeded. Introducing the constraints as inequalities, shown in Equation (4.33), confines the solution space accordingly. Constraints on thrust and rotor speed will keep the blades from stalling and rotor speed from drifting. If the collective is increased significantly, local angles of attack along most of the blade can increase past critical points, causing a loss of thrust. The average angle of attack along the blade spans can be related to the blade loading coefficient (BLC). Conventional rotor blades will begin to stall once the BLC reaches $0.12 - 0.14$ (Leishman, 2006). The XV-15 has small rotor disks, high solidity, highly twisted blades, and can reach BLC values of 0.18 before stall greatly effects the thrust (Felker et al., 1985). The BLC can be easily computed within the controller as C_T is being found at every time step for model prediction, and keeping it below 0.18 will deter stalling:

$$\frac{C_T}{s} \leq 0.18 \quad (4.49)$$

where $\frac{C_T}{s}$ is the BLC. Normally, rotor speed is regulated to a constant value during helicopter operation. Tilt-rotors are unique in that the rotor speed has two nominal values - one for helicopter mode and one for airplane mode. In any case, rotor speed is a vital part of proper operation. The centrifugal forces caused by rotation balance with the aerodynamic forces on the blades and essentially act as a spring, ensuring the stability of the rotor system. The local blade velocities caused by rotation far exceed the velocities caused by forward flight. Therefore as the rotor slows down, the local angles of attack will increase at a constant thrust level. The rotor speed decreasing too much will lead to blade stall. The rotor hub and blades are designed to handle the centrifugal stresses caused by normal operation. If the rotor speed increases too much, the hub or blades can catastrophically fail. The rotor speed must be kept within 20% of its nominal value:

$$0.8\Omega_0 \leq \Omega \leq 1.2\Omega_0 \quad (4.50)$$

where Ω_0 is the nominal rotor speed. The upper limit set on C_T and lower limit set on Ω correlate to the same problem - blade stall. Dalamagkidis removed the lower limit on rotor speed and instead relied on a thrust coefficient constraint. Doing so allows the vehicle to lower its rotor speed below the safe value if it is generating zero thrust. Keeping the lower rotor speed constraint increases the size of the constraint matrix and therefore the derivative, which also increases the computational cost. Fortunately, the upper and lower rotor speed limits are related to each other and calculation of the derivative only needs to occur for one of them.

Arranging the constraints in a vector yields:

$$\begin{aligned}
C(X, U) = & [C_{Tr}(\underline{x}_1, \underline{u}_1) - 0.18s \quad \cdots \quad C_{Tr}(\underline{x}_{N_s}, \underline{u}_{N_s}) - 0.18s \\
& \Omega_r(\underline{x}_1, \underline{u}_1) - 1.2\Omega_0 \quad \cdots \quad \Omega_r(\underline{x}_{N_s}, \underline{u}_{N_s}) - 1.2\Omega_0 \\
& -\Omega_r(\underline{x}_1, \underline{u}_1) + 0.8\Omega_0 \quad \cdots \quad -\Omega_r(\underline{x}_{N_s}, \underline{u}_{N_s}) + 0.8\Omega_0 \\
& C_{Tl}(\underline{x}_1, \underline{u}_1) - 0.18s \quad \cdots \quad C_{Tl}(\underline{x}_{N_s}, \underline{u}_{N_s}) - 0.18s \\
& \Omega_l(\underline{x}_1, \underline{u}_1) - 1.2\Omega_0 \quad \cdots \quad \Omega_l(\underline{x}_{N_s}, \underline{u}_{N_s}) - 1.2\Omega_0 \\
& -\Omega_l(\underline{x}_1, \underline{u}_1) + 0.8\Omega_0 \quad \cdots \quad -\Omega_l(\underline{x}_{N_s}, \underline{u}_{N_s}) + 0.8\Omega_0]^T \quad (4.51)
\end{aligned}$$

with $C \in \mathbb{R}^{6N_s}$ and subscripts r and l denoting the right and left rotors, respectively. Vectorizing the constraints in Equation (4.51), rather than using a matrix similar to U , yields a lower rank gradient and eases implementation. The derivative of C with respect to U results in a tensor of dimension $cN_s \times N_s - 1 \times m$, where c is the number of constraints. Reshaping U into a vector can further reduce the gradient's rank.

$$U^* = [\theta_{0_1} \quad \cdots \quad \theta_{0_{N_s-1}} \quad \theta_{c_1} \quad \cdots \quad \theta_{c_{N_s-1}} \quad \theta_{s_1} \quad \cdots \quad \theta_{s_{N_s-1}} \quad \theta_{rud_1} \quad \cdots \quad \theta_{rud_{N_s-1}}]^T \quad (4.52)$$

Taking the gradient of Equation (4.51) then yields, in block matrix form:

$$\frac{\partial C}{\partial U^*}^T = \begin{bmatrix} \frac{\partial C_{Tr_{1 \dots N_s}}}{\partial \theta_{0_1 \dots N_s-1}} & \frac{\partial \Omega_{r_{1 \dots N_s}}}{\partial \theta_{0_1 \dots N_s-1}} & \frac{-\partial \Omega_{r_{1 \dots N_s}}}{\partial \theta_{0_1 \dots N_s-1}} & \frac{\partial C_{Tl_{1 \dots N_s}}}{\partial \theta_{0_1 \dots N_s-1}} & \frac{\partial \Omega_{l_{1 \dots N_s}}}{\partial \theta_{0_1 \dots N_s-1}} & \frac{-\partial \Omega_{l_{1 \dots N_s}}}{\partial \theta_{0_1 \dots N_s-1}} \\ \frac{\partial C_{Tr_{1 \dots N_s}}}{\partial \theta_{c_1 \dots N_s-1}} & \frac{\partial \Omega_{r_{1 \dots N_s}}}{\partial \theta_{c_1 \dots N_s-1}} & \frac{-\partial \Omega_{r_{1 \dots N_s}}}{\partial \theta_{c_1 \dots N_s-1}} & \frac{\partial C_{Tl_{1 \dots N_s}}}{\partial \theta_{c_1 \dots N_s-1}} & \frac{\partial \Omega_{l_{1 \dots N_s}}}{\partial \theta_{c_1 \dots N_s-1}} & \frac{-\partial \Omega_{l_{1 \dots N_s}}}{\partial \theta_{c_1 \dots N_s-1}} \\ \frac{\partial C_{Tr_{1 \dots N_s}}}{\partial \theta_{s_1 \dots N_s-1}} & \frac{\partial \Omega_{r_{1 \dots N_s}}}{\partial \theta_{s_1 \dots N_s-1}} & \frac{-\partial \Omega_{r_{1 \dots N_s}}}{\partial \theta_{s_1 \dots N_s-1}} & \frac{\partial C_{Tl_{1 \dots N_s}}}{\partial \theta_{s_1 \dots N_s-1}} & \frac{\partial \Omega_{l_{1 \dots N_s}}}{\partial \theta_{s_1 \dots N_s-1}} & \frac{-\partial \Omega_{l_{1 \dots N_s}}}{\partial \theta_{s_1 \dots N_s-1}} \\ \frac{\partial C_{Tr_{1 \dots N_s}}}{\partial \theta_{rud_1 \dots N_s-1}} & \frac{\partial \Omega_{r_{1 \dots N_s}}}{\partial \theta_{rud_1 \dots N_s-1}} & \frac{-\partial \Omega_{r_{1 \dots N_s}}}{\partial \theta_{rud_1 \dots N_s-1}} & \frac{\partial C_{Tl_{1 \dots N_s}}}{\partial \theta_{rud_1 \dots N_s-1}} & \frac{\partial \Omega_{l_{1 \dots N_s}}}{\partial \theta_{rud_1 \dots N_s-1}} & \frac{-\partial \Omega_{l_{1 \dots N_s}}}{\partial \theta_{rud_1 \dots N_s-1}} \end{bmatrix} \quad (4.53)$$

where the derivative of the (1, 1) element can be expanded, similarly to Equation (4.45), as:

$$\frac{\partial C_{Tr}(\underline{x}_k, \underline{u}_k)}{\partial \theta_{0_i}} = \frac{\partial C_{Tr}(\underline{x}_k, \underline{u}_k)}{\partial \underline{x}_k} \frac{\partial \underline{x}_k}{\partial \theta_{0_i}} + \frac{\partial C_{Tr}(\underline{x}_k, \underline{u}_k)}{\partial \underline{u}_k} \frac{\partial \underline{u}_k}{\partial \theta_{0_i}} \quad (4.54)$$

which yields a triangular matrix since the states and controls cannot affect previous constraints. Using the process described by Equation (4.54) with the logic from Eqs. (4.47) and (4.48), each block element of Equation (4.53) can be evaluated and assembled.

The swash plate controls have actuation limits, posing the final constraint, which can be conveniently handled by the activation function used within the recurrent neural network optimization algorithm. Efficiently implementing a saturation can be accomplished with:

$$(\tau)^+ = \min(U_{max}, \max(U_{min}, \tau)) \quad (4.55)$$

where τ is a generic value and U_{max} and U_{min} are the maximum and minimum control limits, respectively.

4.3.4 Recurrent Neural Network

Xia has developed a body of work regarding neural dynamic approaches to nonlinear optimization (Xia, Leung, et al., 2002; Xia & Wang, 2004; Xia & Wang, 2005; Xia, Feng, & Kamel, 2007). A RNN which rapidly solves convex nonlinear optimization problems subject to nonlinear constraints was developed (Xia, Feng, & Wang, 2008):

$$\begin{bmatrix} \Delta U \\ \Delta \underline{\sigma} \end{bmatrix} = \Upsilon \begin{bmatrix} -U + \left(U - \frac{\partial L}{\partial U} - \left[\frac{\partial C}{\partial U^*}^T \underline{\sigma} \right]^* \right)^+ \\ -\underline{\sigma} + (\underline{\sigma} + C(U))^+ \end{bmatrix} \quad (4.56)$$

where $\underline{\sigma} \in \mathbb{R}^{cN_s}$ is a vector of Lagrange multipliers, $\Upsilon > 0$ is the learning rate parameter, and a reshaping of τ is denoted by τ^* . The set of ODE's modeled in Equation (4.56) come from applying the projection theorem applied to a Karush-Khun-Tucker (KKT) point (Kinderlehrer & Stampacchia, 1980). The set of projection equations cast the KKT conditions into two equations, which can then be turned into ODE's with their steady state solutions being the optimal solution

of the problem. The activation function, as described by the projection theorem, is:

$$(\tau)^+ = \max(0, \tau) \quad (4.57)$$

which keeps the Lagrangian multipliers and controls from diverging. However, a saturation function as given by Equation (4.55) can be used in place of the maximum function, within the first line of Equation (4.56), to keep large control inputs from entering the prediction system. Using the maximum function alone does not set any limit to the magnitude of control inputs and can lead to simulation failure within the inflow model. The important action of the activation function is ensuring that its output is PSD. Shifting the control inputs up with the minimum value before use in Equation (4.55), limiting the floor to 0, iterating with the RNN, then shifting the control inputs back down with the minimum control value allows the system to converge while still producing negative control outputs:

$$(\tau)^+ = \min(2U_{max}, \max(0, \tau + U_{min})) \quad (4.58)$$

where $U_{max} = -U_{min}$ and $U_{min} < 0_{m \times N_s}$. The activation function used within the second line of Equation (4.56) is still taken to be Equation (4.57) as the Lagrangian multipliers are defined as positive and not bound within the same saturation limit.

4.3.5 Optimization Stability

The RNN used to find optimal control inputs has a relaxed stability criteria compared to other neural dynamic and conventional approaches. The Hessian of the Lagrangian must be PSD, that is:

$$\frac{\partial^2 \mathcal{L}}{\partial \underline{u}_i^2} \geq 0 \quad \forall i = 1, 2, \dots, N_s - 1 \quad (4.59)$$

where \mathcal{L} is the Lagrangian defined as:

$$\mathcal{L} = L(X, U) + \underline{\sigma}^T C(X, U) \quad (4.60)$$

The Hessian of the Lagrangian can then be written as:

$$\frac{\partial^2 \mathcal{L}}{\partial \underline{u}_i^2} = \frac{\partial^2 L}{\partial \underline{u}_i^2}(X, U) + \underline{a}^T \frac{\partial^2 C}{\partial \underline{u}_i^2}(X, U) \quad (4.61)$$

The first term of Equation (4.61) can be evaluated by differentiating

Equation (4.38) once more:

$$\frac{\partial^2 L}{\partial \underline{u}_i^2} = \frac{\partial \underline{x}_{N_s}}{\partial \underline{u}_i}^T S \frac{\partial \underline{x}_{N_s}}{\partial \underline{u}_i} + \underline{e}_{N_s}^T S \frac{\partial^2 \underline{x}_{N_s}}{\partial \underline{u}_i^2} + \sum_{k=1}^{N_s-1} \left[\frac{\partial \underline{x}_k}{\partial \underline{u}_i}^T Q \frac{\partial \underline{x}_k}{\partial \underline{u}_i} + \underline{e}_k^T Q \frac{\partial^2 \underline{x}_k}{\partial \underline{u}_i^2} \right] \quad (4.62)$$

where, again, S and Q are both symmetric and PSD. The quadratic terms are guaranteed to be PSD (shown in Appendix B) and the remaining terms can be proven to equal 0. Differentiating Equation (4.43) again yields:

$$\frac{\partial^2 \underline{x}_{k+1}}{\partial \underline{u}_i^2} = \frac{\partial^2 \underline{x}_k}{\partial \underline{u}_i^2} + t_s \frac{\partial^2 f}{\partial \underline{u}_i^2}(\underline{x}_k, \underline{u}_k) \quad (4.63)$$

which describes the second order effects of the control inputs throughout the horizon. The ROM used within this work is a control affine system:

$$f(\underline{x}_k, \underline{u}_k) = h(\underline{x}_k) + g(\underline{x}_k)\underline{u}_k \quad (4.64)$$

which shows that the inputs do not have any second order effects on the states.

Evaluating the second derivative of Equation (4.64) with respect to some control input u_i yields:

$$\frac{\partial^2 f}{\partial \underline{u}_i^2}(\underline{x}_k, \underline{u}_k) = 0 \quad \forall i, k = 1, 2, \dots, N_s - 1 \quad (4.65)$$

Using this result along with the fact that the control inputs have no effect on the initial state - that is:

$$\frac{\partial^2 \underline{x}_1}{\partial \underline{u}_i^2} = 0 \quad \forall i = 1, 2, \dots, N_s - 1 \quad (4.66)$$

Equation (4.62) can be reduced to:

$$\frac{\partial^2 L}{\partial \underline{u}_i^2} = \frac{\partial \underline{x}_{N_s}}{\partial \underline{u}_i}^T S \frac{\partial \underline{x}_{N_s}}{\partial \underline{u}_i} + \sum_{k=1}^{N_s-1} \left[\frac{\partial \underline{x}_k}{\partial \underline{u}_i}^T Q \frac{\partial \underline{x}_k}{\partial \underline{u}_i} \right] \quad (4.67)$$

which is guaranteed to be PSD. Second, since the constraint function depends solely on the states and control inputs, there are no second order effects on the constraints from the controls:

$$\frac{\partial^2 C}{\partial \underline{u}_i^2} = 0 \quad \forall \quad i = 1, 2, \dots, N_s - 1 \quad (4.68)$$

Therefore,

$$\frac{\partial^2 \mathcal{L}}{\partial \underline{u}_i^2} \geq 0 \quad \forall \quad i = 1, 2, \dots, N_s - 1 \quad (4.69)$$

and the RNN will globally converge to an optimal solution of Equation (4.33).

5. Model Comparison

The ROM should accurately describe the vehicle states over the prediction horizon for effective controller implementation. Model predictive control offers some robustness to unmodelled dynamics compared to other optimal control techniques, but large prediction errors can still lead to ineffective and poor controller performance. Given equal initial conditions, the ROM is compared to the high fidelity simulation (SIM), subject to varying control inputs. In each case where a control input is applied, the input comes in the form of a 5° doublet. The doublet has a period of 2 seconds and is applied at a simulation time of 1 second:

$$\underline{u}_j = \begin{cases} \underline{u}_{j0}, & t < 1 \\ \underline{u}_{j0} + 5^\circ, & 1 \leq t < 2 \\ \underline{u}_{j0} - 5^\circ, & 2 \leq t < 3 \\ \underline{u}_{j0}, & t \geq 3 \end{cases} \quad (5.1)$$

where $j = 1, 2, 3, 4$ signifies which control is being affected and \underline{u}_{j0} is the nominal value of \underline{u}_j .

5.1 Collective Shift

The ROM and SIM exhibit similar control power and flapping dynamics but have an offset in the coning angle and collective input. This is likely related to the underlying assumptions made with the linearized flapping equations. The flow angle along the blade span is assumed to be small enough that a local portion of thrust (ΔT) is given as:

$$\Delta T = \Delta l + \Delta d\phi \quad (5.2)$$

Given the large twist of tilt rotor blades, this assumption, while helpful for deriving simple analytical equations, is not accurate for the entire span of the blade. The ROM takes significantly higher levels of collective input to reach the same amount

of thrust. Figure 5.1 shows the static thrust from a single rotor the two models at varying collective inputs. Fitting lines to the two curves and equating thrust gives a relation between the ROM and SIM:

$$\hat{\theta}_{0SIM} = 0.9603\hat{\theta}_{0ROM} - 0.1865 \quad (5.3)$$

showing that the two curves are nearly parallel. The steady offset will not affect the controller performance if this relation is used to transform the calculated control inputs before use in the SIM.

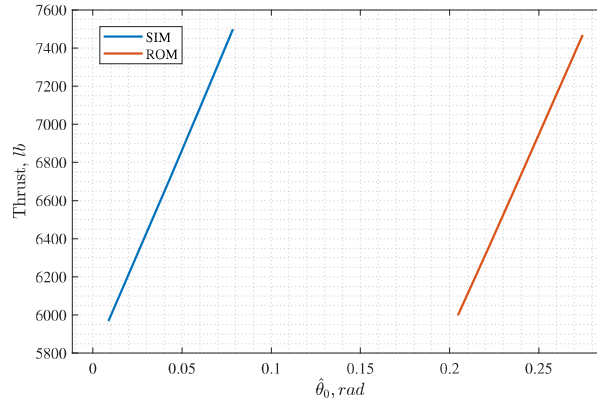


Figure 5.1. Thrust produced by a single rotor in the high fidelity simulation (SIM) and reduced order model (ROM) for varying collective inputs.

5.2 Control Inputs with Constant Rotor Speed

The two models are compared with a constant rotor speed, which allows the control effectiveness and natural vehicle response of both models to be easily equated. Longitudinal states are tracked accurately given longitudinal stick and collective inputs. Roll rate and lateral stick inputs are well predicted given small yaw rates. Rudder inputs, while predicted moderately well, lead to large roll error due to an unmodelled coupling between yaw rate and rolling moment.

5.2.1 Longitudinal Doublet

Figures 5.2 and 5.3 show the longitudinal and flapping states of the ROM and SIM. Applying a longitudinal doublet with an initial forward speed shows a good correlation in pitch rate between the two models. Some errors in u_b and

w_b propagate into the system as the simulation time continues to increase. The longitudinal states track closely for 3 seconds after the doublet is ended, which provides sufficient prediction time for the MPC algorithm. The effects of the control input and vehicle states on blade flapping are accurately predicted. Some steady state error in the coning angle (β_0) exists but does not affect the vehicle dynamics. The control input causes symmetric lateral flapping; therefore, the lateral-direction vehicle states are decoupled from longitudinal inputs.

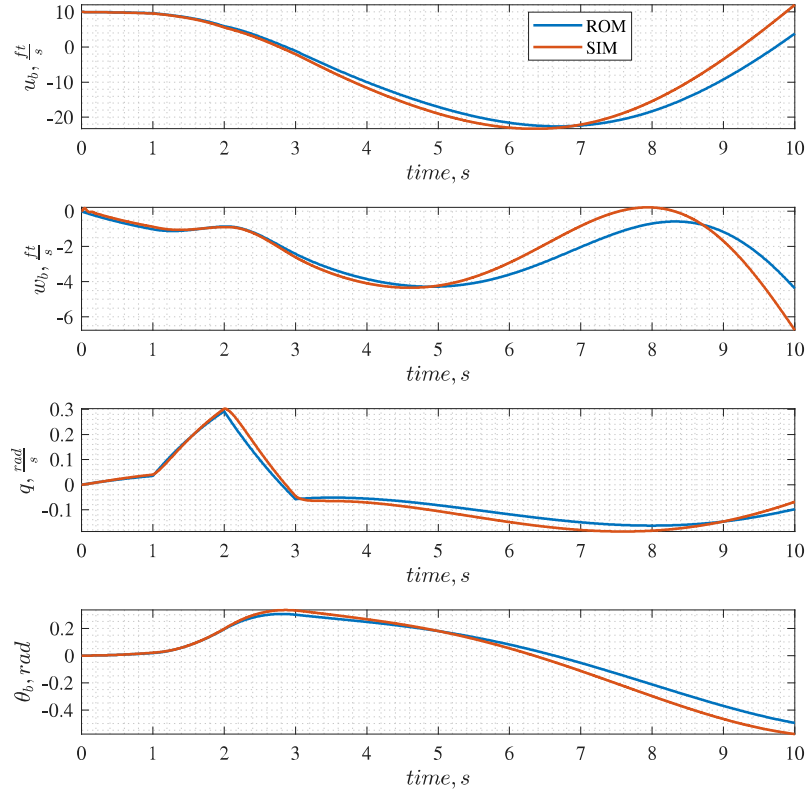


Figure 5.2. Longitudinal states comparison of reduced order model (ROM) and high fidelity simulation (SIM) given a 5° longitudinal doublet and an initial x_b velocity of $10 \frac{ft}{s}$.

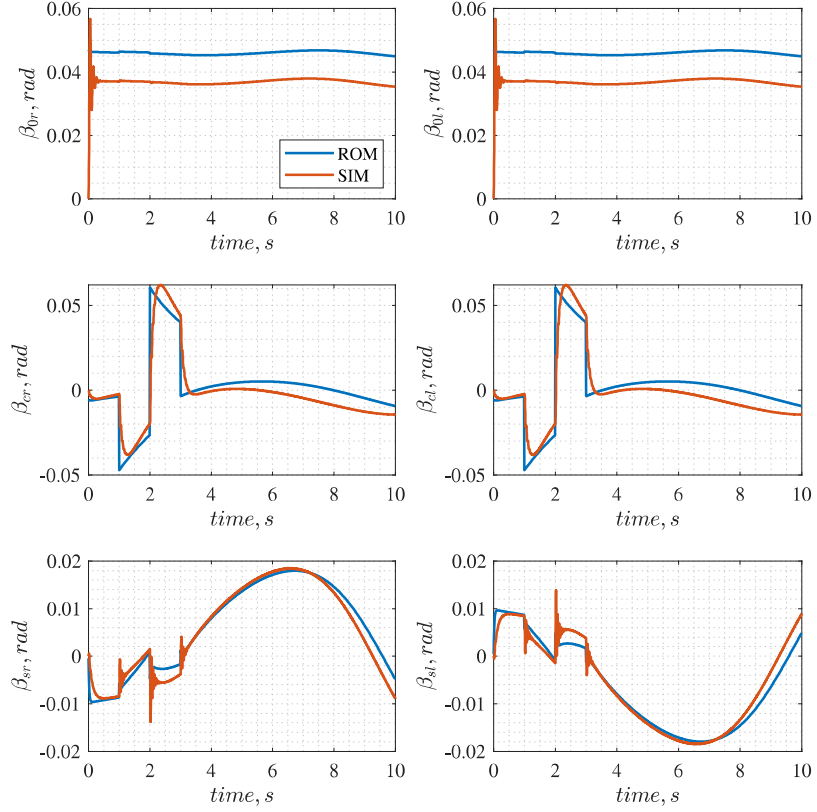


Figure 5.3. Flapping states comparison of reduced order model (ROM) and high fidelity simulation (SIM) given a 5° longitudinal doublet and an initial x_b velocity of $10 \frac{ft}{s}$.

5.2.2 Lateral Doublet

Figures 5.4, 5.5, and 5.6 show the longitudinal, lateral-directional, and flapping states of the vehicle given an initial u_b velocity and a lateral stick doublet. The direct effect on roll rate is accurately predicted and most vehicle states track well with each other. The yaw rate experiences some tracking error that is paralleled by the error in longitudinal flapping. The lateral input correlates to combined lateral cyclic instead of differential collective, which is shown by the large response seen by β_s and small response in β_0 .

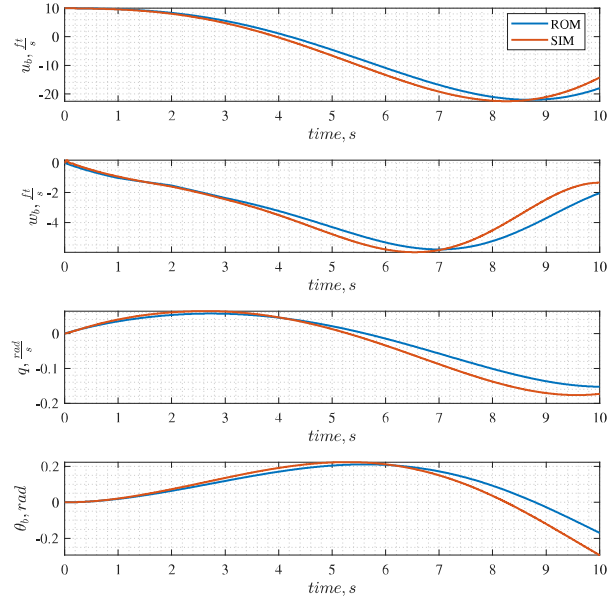


Figure 5.4. Longitudinal states comparison of reduced order model (ROM) and high fidelity simulation (SIM) given a 5° lateral doublet and an initial x_b velocity of $10 \frac{ft}{s}$.

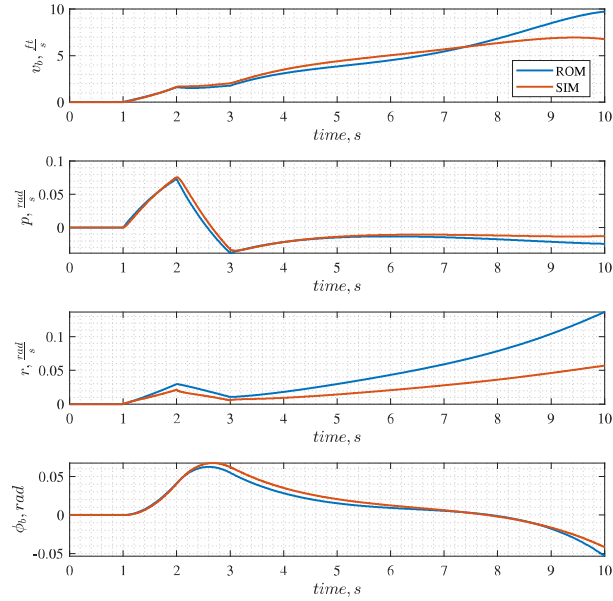


Figure 5.5. Lateral states comparison of reduced order model (ROM) and high fidelity simulation (SIM) given a 5° lateral doublet and an initial x_b velocity of $10 \frac{ft}{s}$.

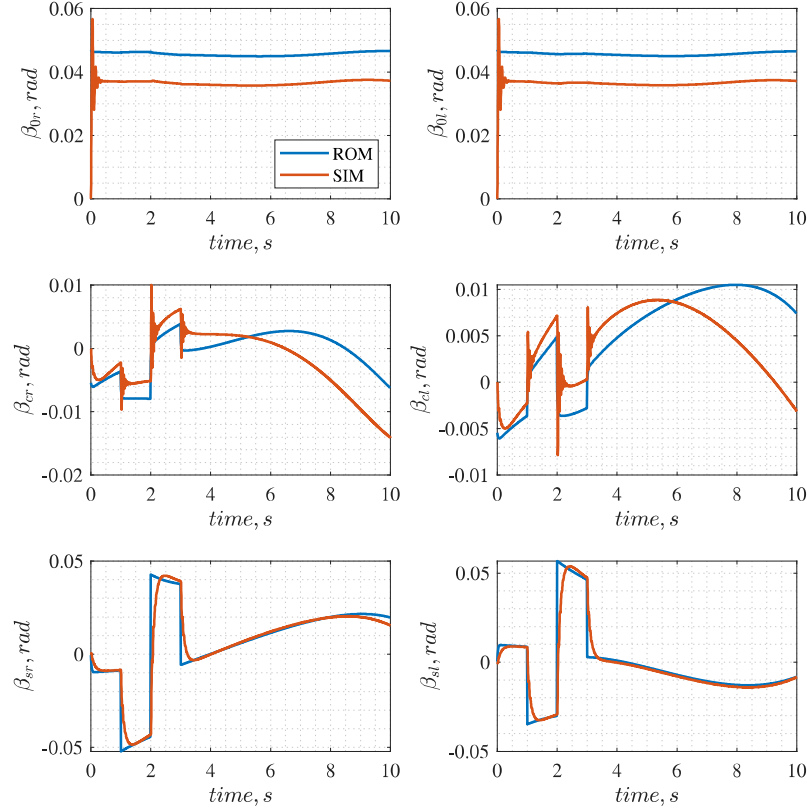


Figure 5.6. Flapping states comparison of reduced order model (ROM) and high fidelity simulation (SIM) given a 5° lateral doublet and an initial x_b velocity of $10 \frac{ft}{s}$.

5.2.3 Rudder Doublet

Figures 5.7, 5.8, and 5.9 show the longitudinal, lateral-directional, and flapping states given an initial v_b velocity and a rudder doublet. The direct yaw effect of the rudder is slightly over-predicted by the ROM and a significant steady state error exists after the doublet. The roll rate is not predicted well as described in Section 4.2.1.2. There is a coupling between yaw rate and the rolling moment, which is not captured by the ROM. The natural vehicle response to the initial conditions tracks well except for the yaw and roll rates where significant error builds within the first second of simulation.

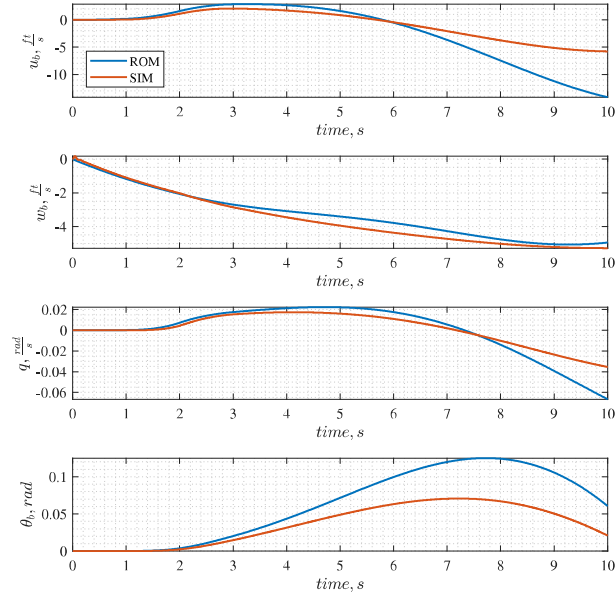


Figure 5.7. Longitudinal states comparison of reduced order model (ROM) and high fidelity simulation (SIM) given a 5° rudder doublet and an initial v_b velocity of $10 \frac{ft}{s}$.

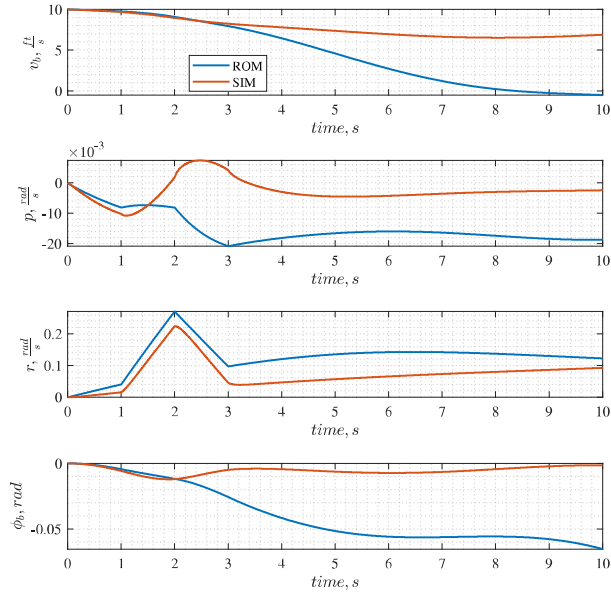


Figure 5.8. Lateral states comparison of reduced order model (ROM) and high fidelity simulation (SIM) given a 5° rudder doublet and an initial v_b velocity of $10 \frac{ft}{s}$.

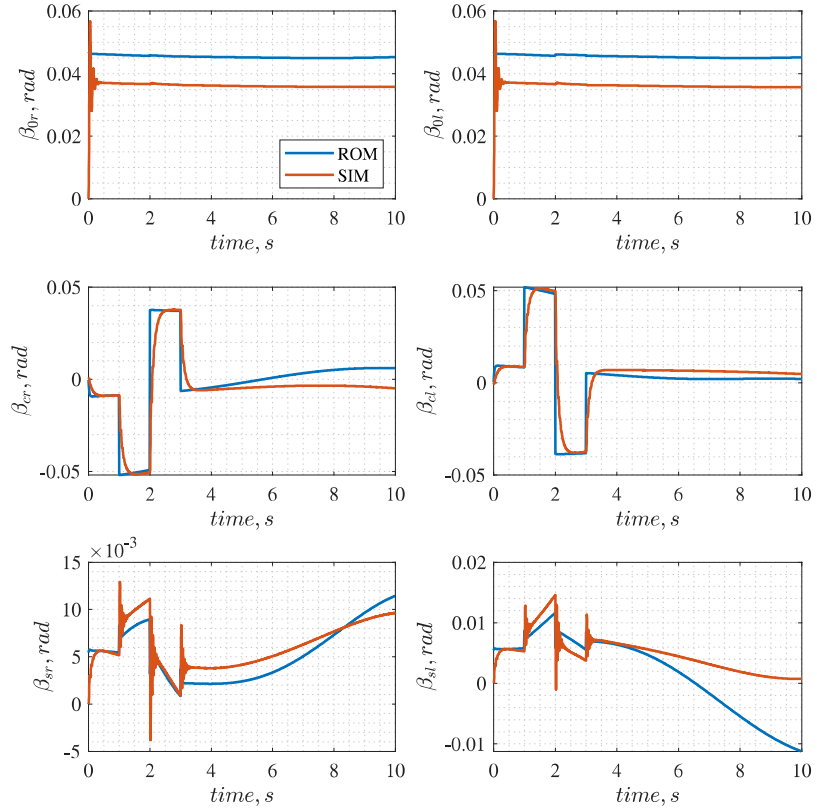


Figure 5.9. Flapping states comparison of reduced order model (ROM) and high fidelity simulation (SIM) given a 5° rudder doublet and an initial v_b velocity of $10 \frac{ft}{s}$.

5.2.4 Collective Doublet

Figures 5.10 and 5.11 show the longitudinal and flapping states given an initial u_b and w_b velocity and a 5° collective doublet. The effect of the collective input on the sink rate is well predicted along with the vehicle states after this point. The collective input does not excite any lateral-directional motion since the lateral flapping states are symmetric. The transient coning angles are significantly under-predicted but the steady difference is correct.

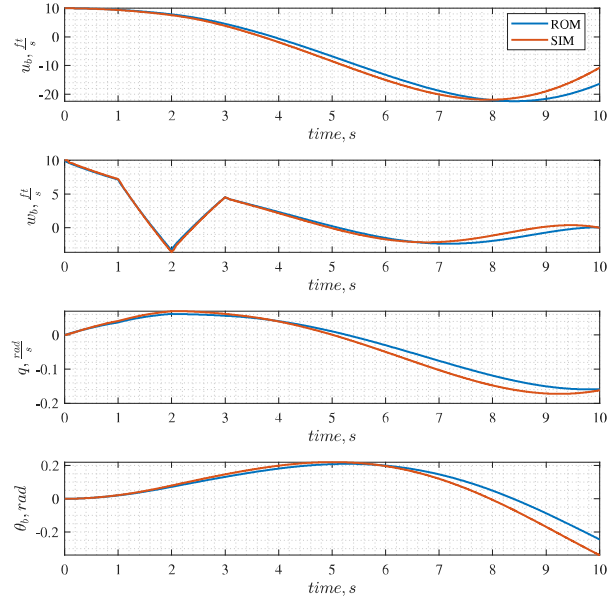


Figure 5.10. Longitudinal states comparison of reduced order model (ROM) and high fidelity simulation (SIM) given a 5° collective doublet and an initial u_b and w_b velocity of $10 \frac{ft}{s}$.

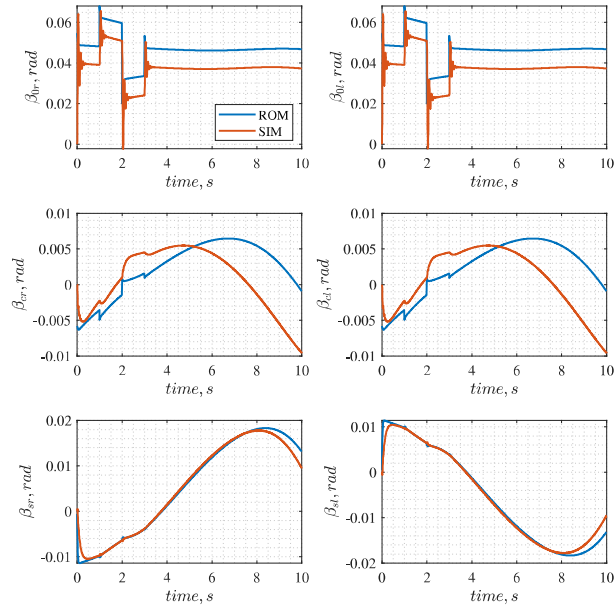


Figure 5.11. Flapping states comparison of reduced order model (ROM) and high fidelity simulation (SIM) given a 5° collective doublet and an initial u_b and w_b velocity of $10 \frac{ft}{s}$.

5.3 Variable Rotor Speed

Letting the rotor speed vary adds significant nonlinearity to rotorcraft simulation. Rotor speed is responsible for keeping the rotors blades extended outward while subject to aerodynamic forces. Angular velocity around the z_h axis gives the local blade sections sufficient velocity to produce thrust. Varying rotor speed changes not only the flapping dynamics in a nonlinear manner but also the blade aerodynamics. The ROM does not capture nonlinear aerodynamics but does still effectively predict blade flapping and vehicle states with variable rotor speed.

5.3.1 Uncontrolled

Figures 5.13, 5.14, and 5.12 show the longitudinal states, flapping states, and rotor speed given a large initial u_b and w_b velocity with a varying rotor speed. The vehicle states change drastically over the short simulation. The initial forward speed causes a large pitching moment and pitch rate, which results in θ_b exceeding 60° . The ROM slightly under-predicts the pitch, which partially leads to the difference in u_b and w_b . The rotors act as parachutes and tend to force the z_h axis into the oncoming wind, which is shown by the longitudinal flapping. The coning angle decreases as u_b goes to zero, then increases once again as u_b becomes increasingly negative, which is predicted accurately by the ROM.

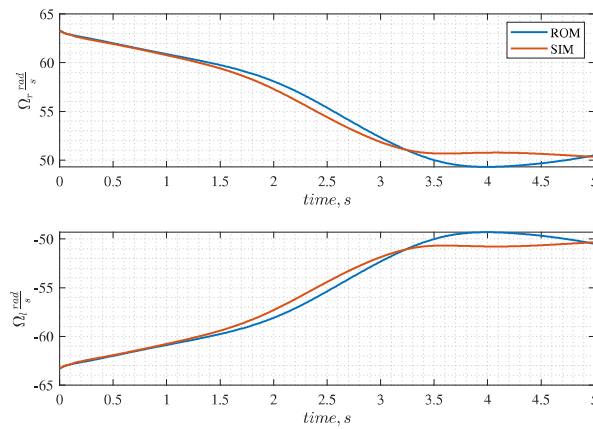


Figure 5.12. Rotor speed comparison of reduced order model (ROM) and high fidelity simulation (SIM) given an initial u_b and w_b velocity of $80 \frac{ft}{s}$ and $40 \frac{ft}{s}$, respectively.

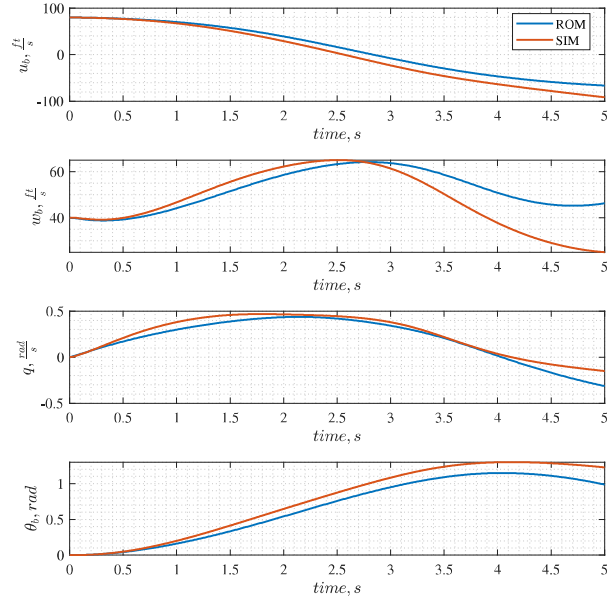


Figure 5.13. Longitudinal states comparison of reduced order model (ROM) and high fidelity simulation (SIM) with variable rotor speed given an initial u_b and w_b velocity of $80 \frac{ft}{s}$ and $40 \frac{ft}{s}$, respectively.

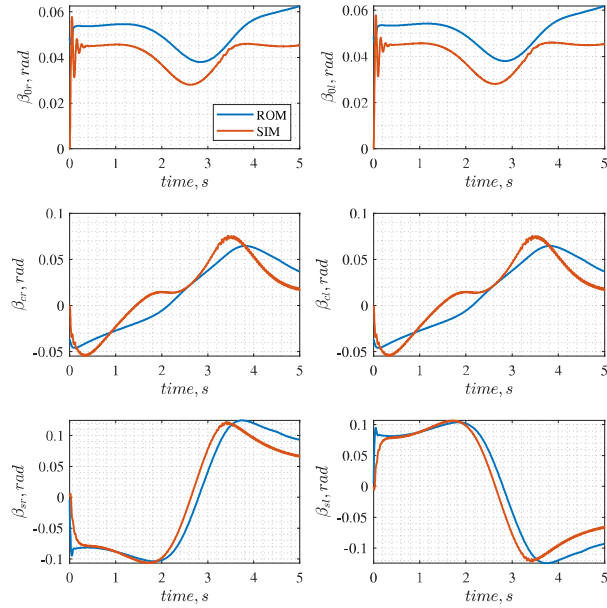


Figure 5.14. Flapping states comparison of reduced order model (ROM) and high fidelity simulation (SIM) with variable rotor speed given an initial u_b and w_b velocity of $80 \frac{ft}{s}$ and $40 \frac{ft}{s}$, respectively.

5.3.2 Longitudinal Doublet

Figures 5.16, 5.17, and 5.15 show the longitudinal states, flapping states, and rotor speeds given a longitudinal input doublet and initial u_b and w_b velocity with variable rotor speed. The initial positive portion of the doublet does not significantly increase the pitch rate, which is already increasing quickly due to the large forward speed, and the two models show similar effects. The negative portion of the doublet begins shortly before u_b crosses zero, which produces a similar effect to the positive portion in that the control and natural vehicle response are working in unison to negatively pitch the vehicle. The magnitude of the maximum pitch is largely unaffected by the control input compared to the uncontrolled case. Longitudinal blade flapping is roughly predicted. The highly dynamic system is likely exciting unmodeled nonlinearities in the SIM.

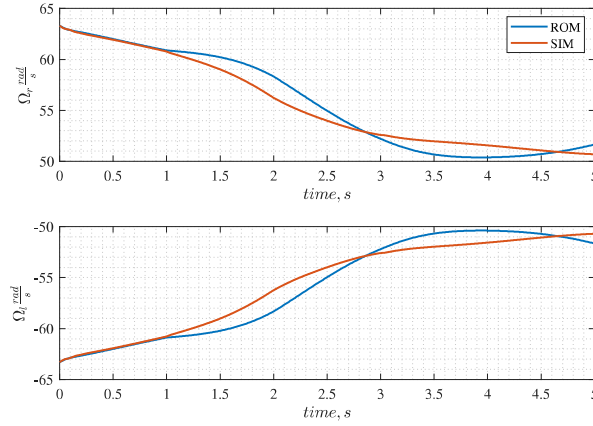


Figure 5.15. Rotor speed comparison of reduced order model (ROM) and high fidelity simulation (SIM) given a 5° longitudinal doublet and an initial u_b and w_b velocity of $80 \frac{ft}{s}$ and $40 \frac{ft}{s}$, respectively.

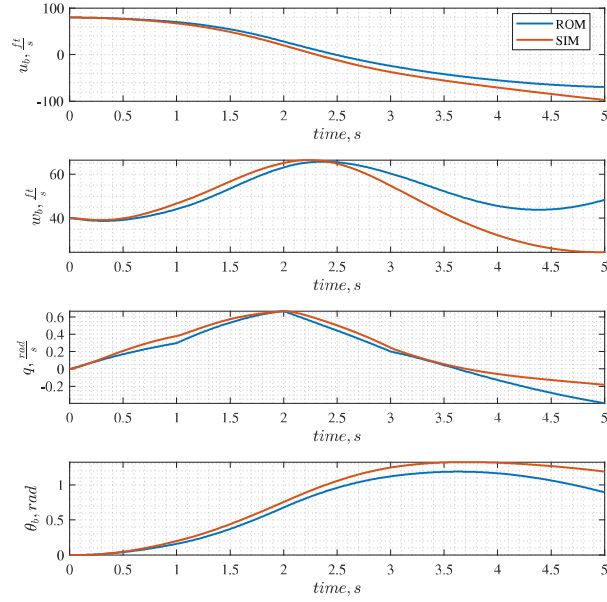


Figure 5.16. Longitudinal states comparison of reduced order model (ROM) and high fidelity simulation (SIM) with variable rotor speed given a 5° longitudinal doublet and an initial u_b and w_b velocity of $80 \frac{ft}{s}$ and $40 \frac{ft}{s}$, respectively.

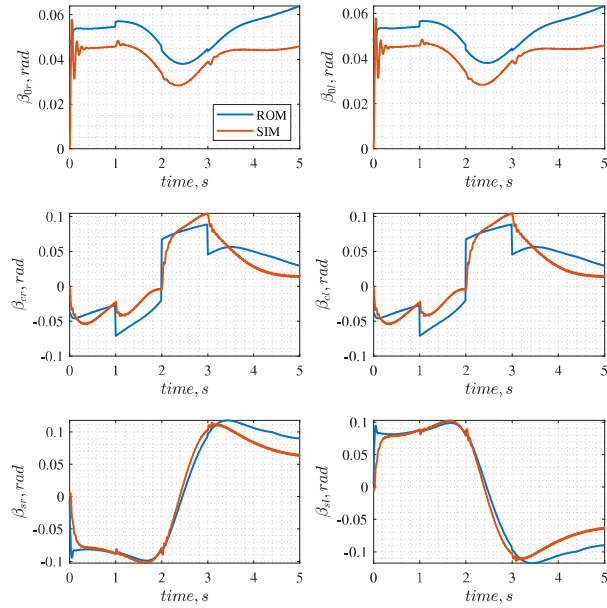


Figure 5.17. Flapping states comparison of reduced order model (ROM) and high fidelity simulation (SIM) with variable rotor speed given a 5° longitudinal doublet and an initial u_b and w_b velocity of $80 \frac{ft}{s}$ and $40 \frac{ft}{s}$, respectively.

5.3.3 Lateral Doublet

Figures 5.18, 5.19, 5.20, and 5.21 show the longitudinal states, lateral states, flapping states, and rotor speeds given a lateral input doublet and initial u_b and w_b velocity with variable rotor speed. The doublet is applied through combined cyclic and its effect is not well predicted by the ROM. Roll rate is significantly under predicted causing large discrepancies in both the longitudinal and lateral-directional states. Blade flapping is not not predicted well either which is also caused by the difference in roll rate. Flapping induced by roll rate come from the z_h position and mainly affects the lateral plane.

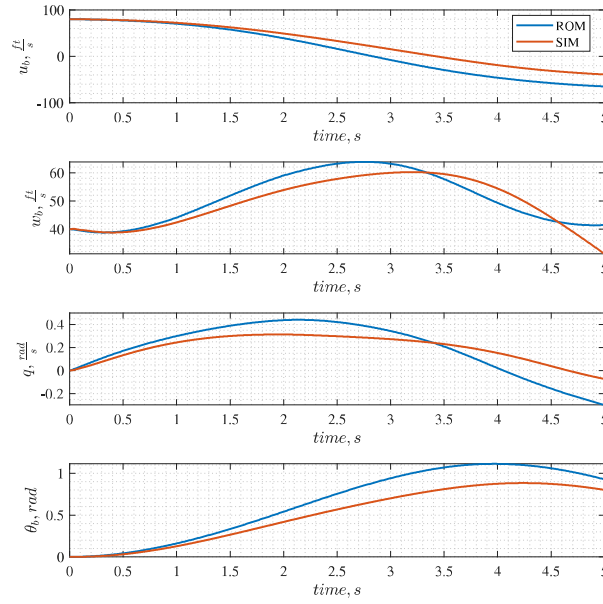


Figure 5.18. Longitudinal states comparison of reduced order model (ROM) and high fidelity simulation (SIM) with variable rotor speed given a 5° lateral doublet and an initial u_b and w_b velocity of $80 \frac{ft}{s}$ and $40 \frac{ft}{s}$, respectively.

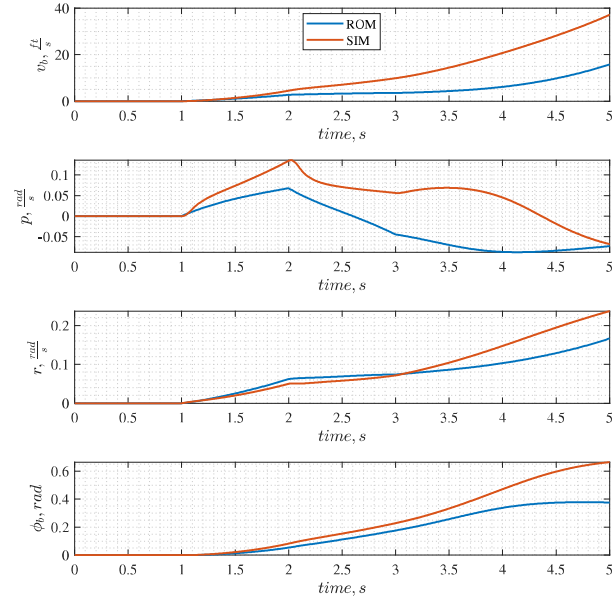


Figure 5.19. Lateral states comparison of reduced order model (ROM) and high fidelity simulation (SIM) with variable rotor speed given a 5° lateral doublet and an initial u_b and w_b velocity of $80 \frac{ft}{s}$ and $40 \frac{ft}{s}$, respectively.

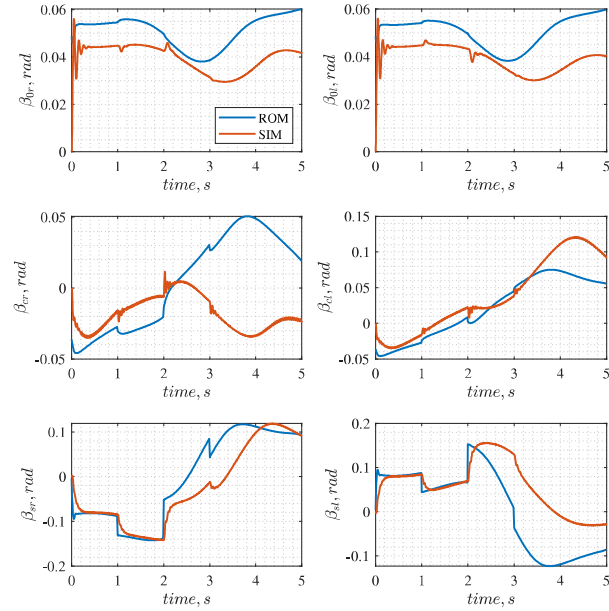


Figure 5.20. Flapping states comparison of reduced order model (ROM) and high fidelity simulation (SIM) with variable rotor speed given a 5° lateral doublet and an initial u_b and w_b velocity of $80 \frac{ft}{s}$ and $40 \frac{ft}{s}$, respectively.

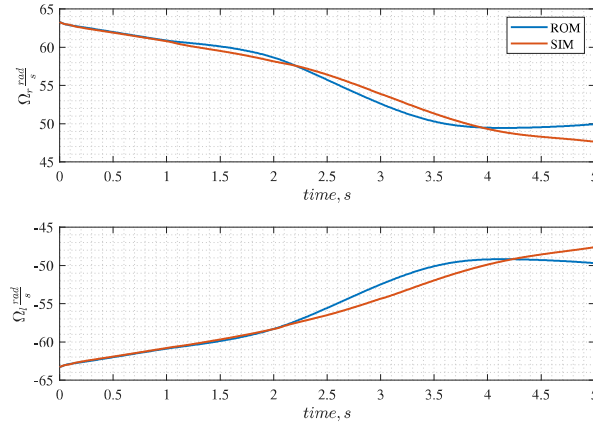


Figure 5.21. Rotor speed comparison of reduced order model (ROM) and high fidelity simulation (SIM) given a 5° lateral doublet and an initial u_b and w_b velocity of $80 \frac{ft}{s}$ and $40 \frac{ft}{s}$, respectively.

5.3.4 Rudder Doublet

Figures 5.23, 5.24, 5.25, and 5.22 show the longitudinal states, lateral states, flapping states, and rotor speeds given a rudder input doublet and initial u_b and w_b velocity with variable rotor speed. The coupling between yaw rate and rolling moment is again evident and neither roll rate nor roll angle are well tracked. The sink rate and lateral velocity are under predicted following the control input likely due to the error in the roll angle. Yaw rate itself is estimated fairly well along with pitch and pitch rate.

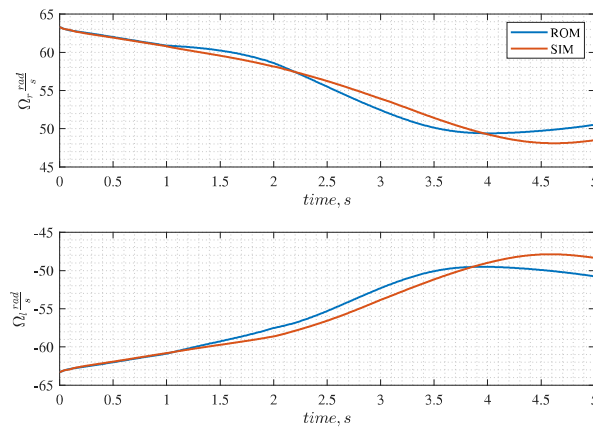


Figure 5.22. Rotor speed comparison of reduced order model (ROM) and high fidelity simulation (SIM) given a 5° rudder doublet and an initial u_b and w_b velocity of $80 \frac{ft}{s}$ and $40 \frac{ft}{s}$, respectively.

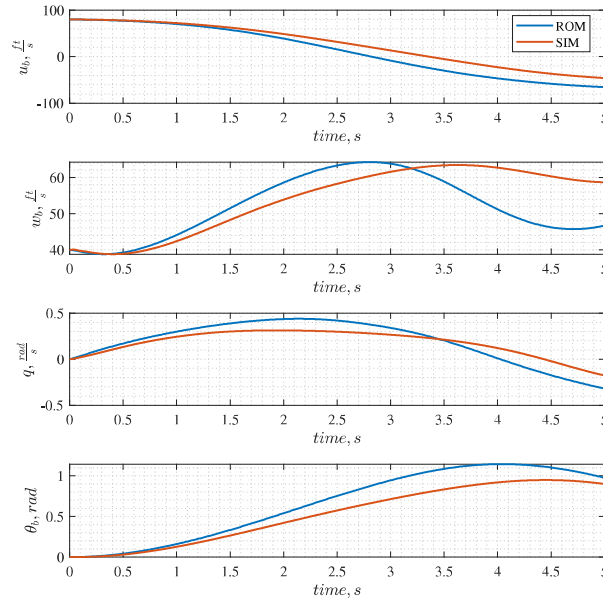


Figure 5.23. Longitudinal states comparison of reduced order model (ROM) and high fidelity simulation (SIM) with variable rotor speed given a 5° rudder doublet and an initial u_b and w_b velocity of $80\frac{ft}{s}$ and $40\frac{ft}{s}$, respectively.

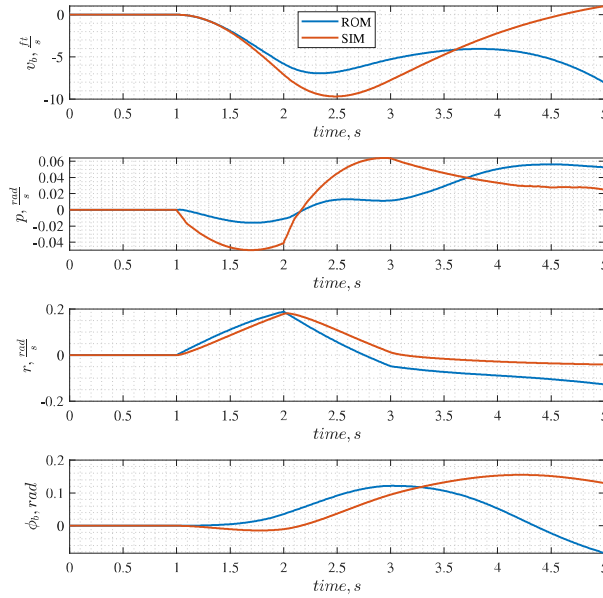


Figure 5.24. Lateral states comparison of reduced order model (ROM) and high fidelity simulation (SIM) with variable rotor speed given a 5° rudder doublet and an initial u_b and w_b velocity of $80\frac{ft}{s}$ and $40\frac{ft}{s}$, respectively.

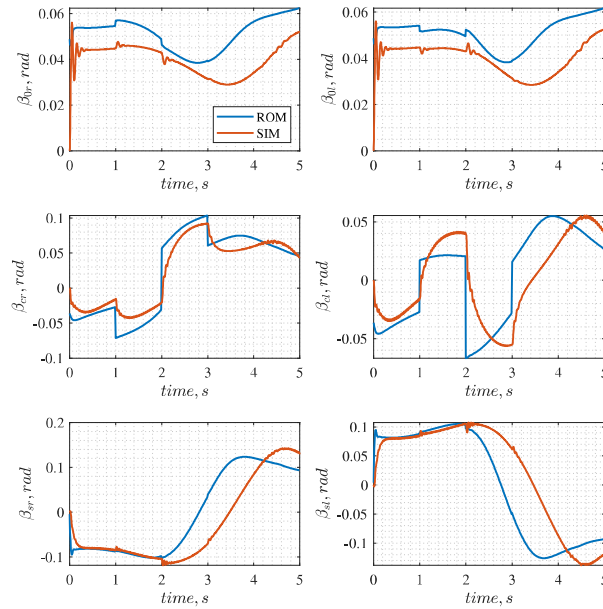


Figure 5.25. Flapping states comparison of reduced order model (ROM) and high fidelity simulation (SIM) with variable rotor speed given a 5° rudder doublet and an initial u_b and w_b velocity of $80 \frac{ft}{s}$ and $40 \frac{ft}{s}$, respectively.

5.3.5 Collective Doublet

Figures 5.26, 5.27, and 5.28 show the longitudinal states, flapping states, and rotor speeds given a collective input doublet and initial u_b and w_b velocity with variable rotor speed. Collective input effects must be accurately modelled, especially by the rotor speed, as it has a large effect on the aerodynamic state of the rotors. The ROM shows poor correlation in w_b after the doublet is applied but this could be in part due to the combined errors of the other states. Interestingly, both the ROM and SIM show that the increased collective only marginally decreases the rate of change of w_b , which is again likely caused by the vehicle tending to force the z_h axis into the wind. Most importantly, the ROM very accurately predicts the rate of energy removal from the rotor during the initial positive portion of the doublet.

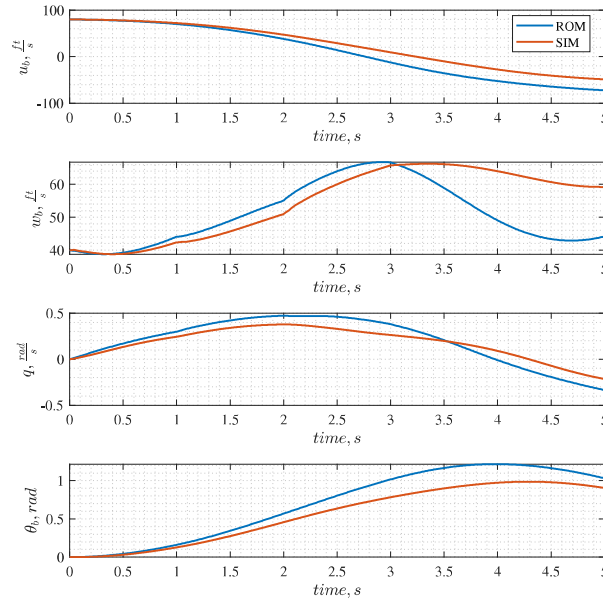


Figure 5.26. Longitudinal states comparison of reduced order model (ROM) and high fidelity simulation (SIM) with variable rotor speed given a 5° collective doublet and an initial u_b and w_b velocity of $80 \frac{ft}{s}$ and $40 \frac{ft}{s}$, respectively.

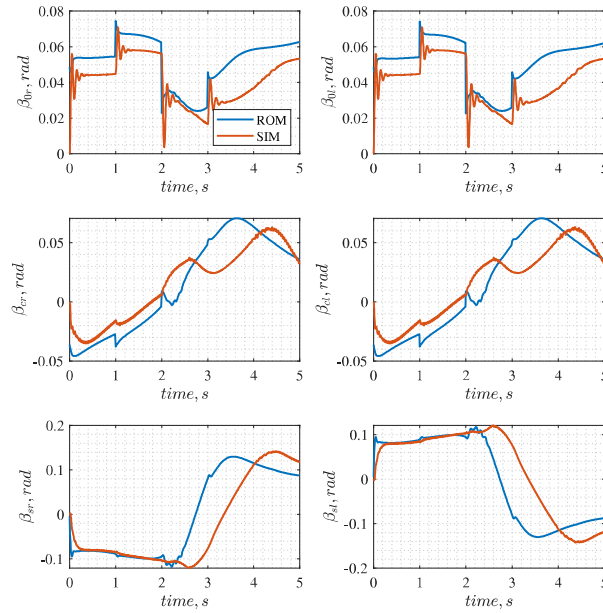


Figure 5.27. Flapping states comparison of reduced order model (ROM) and high fidelity simulation (SIM) with variable rotor speed given a 5° collective doublet and an initial u_b and w_b velocity of $80 \frac{ft}{s}$ and $40 \frac{ft}{s}$, respectively.

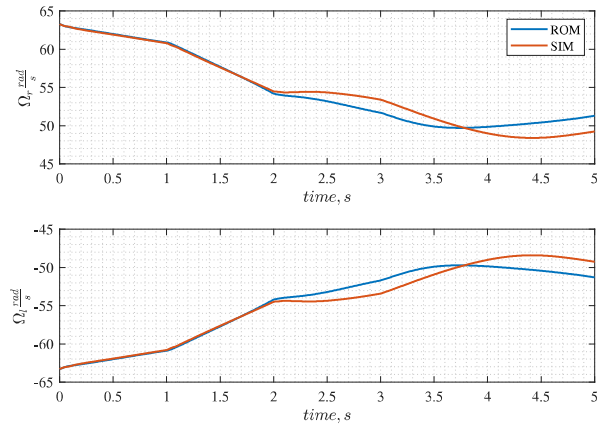


Figure 5.28. Rotor speed comparison of reduced order model (ROM) and high fidelity simulation (SIM) given a 5° collective doublet and an initial u_b and w_b velocity of $80 \frac{ft}{s}$ and $40 \frac{ft}{s}$, respectively.

6. Controller Simulation

The controller has been applied to nominal flight as well as power-off maneuvers at 83.3Hz. A high frequency was used to help improve performance in the presence of unmodeled dynamics. The prediction horizon was taken to be 1.2s ($N_s = 100$ and $t_s = 0.012$). In standard forward flight, a prediction horizon of 1s was determined to provide sufficient length that the controller could reach a relatively steady state value without large oscillations. A shorter horizon, while significantly less costly to compute, does not perform well enough to control the SIM. The weighting matrices Q and R were varied to analyze their effects and test the controller performance. The states have not been normalized for use inside of the controller, so using different magnitudes within the Q matrix will help ensure states are weighted evenly.

The controller was applied to forward flight starting at the reference state but not statically trimmed and away from the reference state. A comparison is made between the performance of the controller in all axes and just the longitudinal axis during AR entrance. Finally, the rotor speed constraints are tested by starting the simulation near a boundary and putting the reference trajectory beyond it.

6.1 Forward Flight Constant Rotor Speed

Flight controllers are used largely to make vehicles maintain constant speeds and attitudes for long periods of time in cruise. Figures 6.1, 6.2, and 6.3 show the vehicle states and control inputs with an initial u_b velocity. It attempts to regulate all angular rates to zero, which works well in the longitudinal axis but begins to fail in the lateral-directional axis. The values of p , r , and ϕ_b are sufficiently small that the controller is working but needs further refinement. The performance suggests the lateral-directional states are not being well predicted, which is reinforced by the model comparisons presented in Chapter 5. Restricting the dynamics to the longitudinal axis and only solving for optimal values of collective and longitudinal

stick significantly improves the controller performance. Figures 6.4 and 6.5 show the longitudinal states and control inputs given an initial condition farther away from the reference. The controller is able to take the vehicle from a dynamic initial position to the reference forward flight state. The values of pitch and pitch rate have not fully settled to zero within the simulation time but show a converging trend. The control inputs show the effect of the saturation function within the RNN. The control limits are met and but not exceeded while the system attempts to arrest transient motion. Ripples in collective are produced, reflected in w_b , likely due to an inconsistency between the ROM and SIM. The ROM is likely over predicting the effective control power of the collective causing the value to shift up and down.

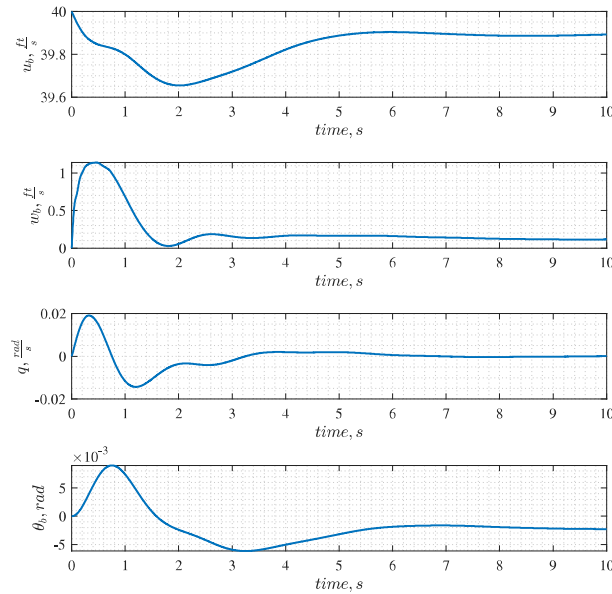


Figure 6.1. Controlled longitudinal states with constant rotor speed in forward flight.

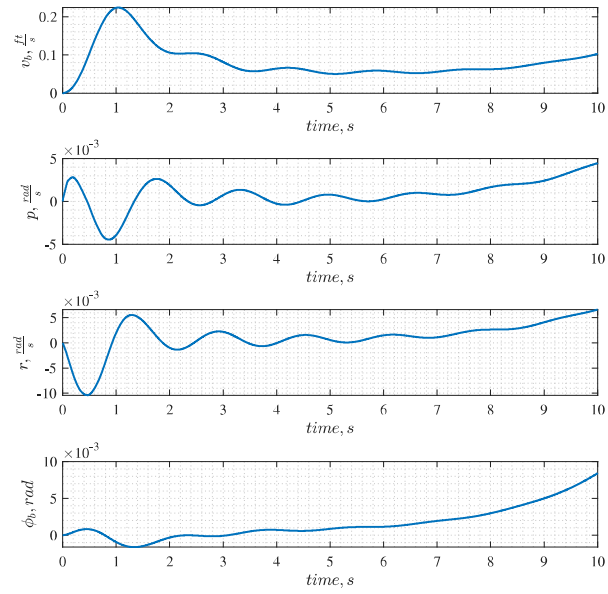


Figure 6.2. Controlled lateral states with constant rotor speed in forward flight.

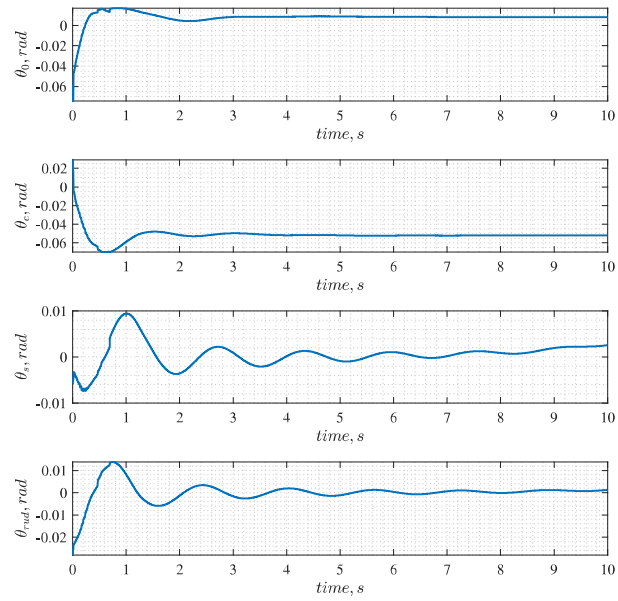


Figure 6.3. Controller inputs with constant rotor speed in forward flight.

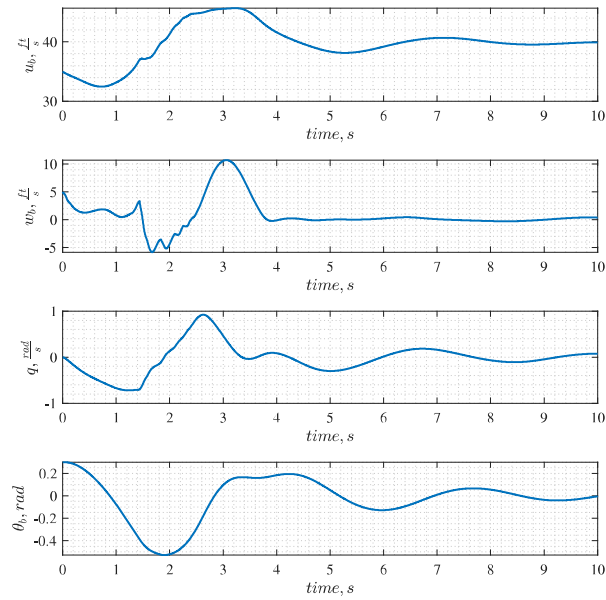


Figure 6.4. Controlled longitudinal states with constant rotor speed in forward flight and initial conditions away from the reference.

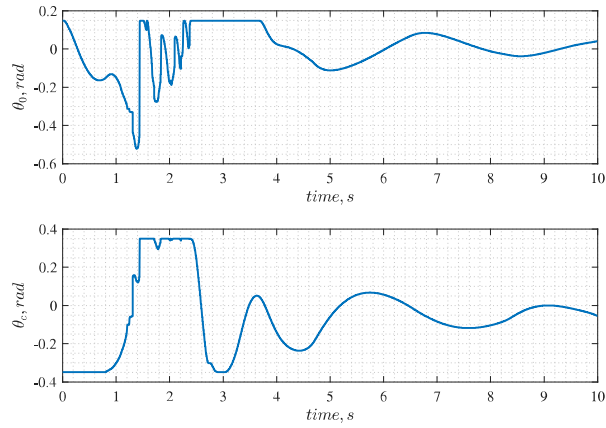


Figure 6.5. Controller inputs with constant rotor speed in forward flight and initial conditions away from the reference.

6.2 Autorotation Entrance

Upon engine failure, the controller needs to regulate the states to a steady AR. The rotor speed must be forced to a constant value near the nominal operating point, the *body* angular rates should be regulated to zero, and the *body* velocities forced to the reference values. Figures 6.6 - 6.9 show two simulations of the controller attempting to enter a steady AR: one restricted to longitudinal motion

and one including lateral-directional motion. The weighting matrix is set such that w_b is not weighted but u_b is, so the controller will find a sink rate at the given forward speed which results in a proper energy balance between the rotors and air flow.

The longitudinal states are well controlled when the dynamics are restricted but not when lateral-directional states are allowed to propagate. Large weights are put on the angular rates and Euler angles attempting to keep the body level and controlled. The roll rate is regulated to zero well but yaw rate is steadily increasing. The poor performance is again most likely due to inaccuracies within the prediction model. Lowering the weights on the angular rates and Euler angles, relative to the velocities, does not improve the controller performance.

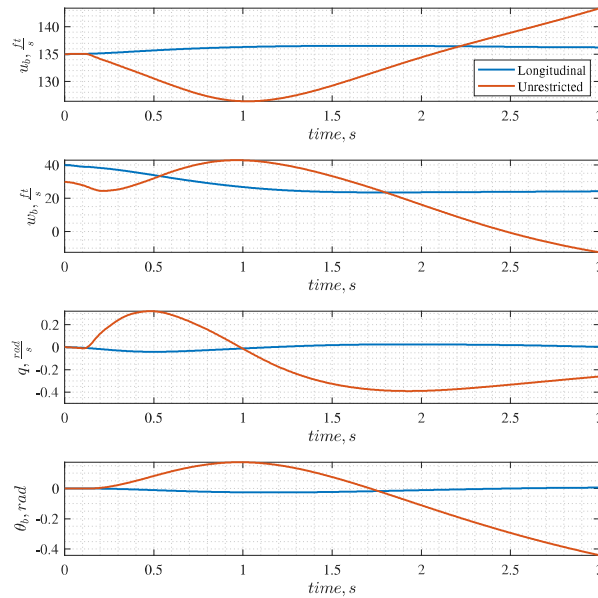


Figure 6.6. Controlled longitudinal states of a longitudinally restricted simulation and a full spatial 6-DOF simulation.

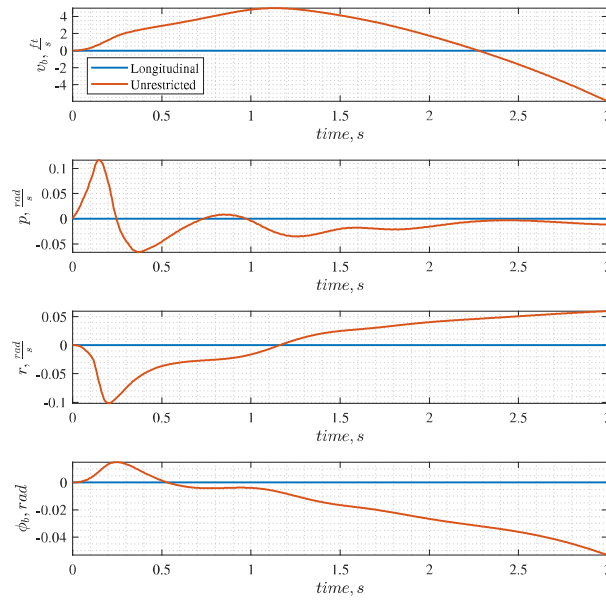


Figure 6.7. Controlled lateral states of a longitudinally restricted simulation and a full spatial 6-DOF simulation.

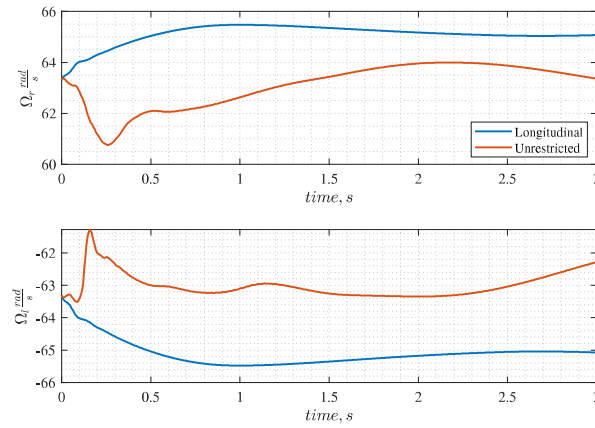


Figure 6.8. Rotor speed of a longitudinally restricted simulation and a full spatial 6-DOF simulation.

In both simulations, the rotor speed is well regulated. The large control inputs at the beginning of the unrestricted simulation result in the decrease in rotor speed. The energy contained within the rotor is transferred to the body as roll and yaw rates. The small change in rotor speed shows the amount of energy that is stored within the two rotor systems. During flare and landing, this energy is used to

control the vehicle. Lateral stick and rudder inputs remain zero for the longitudinal simulation.

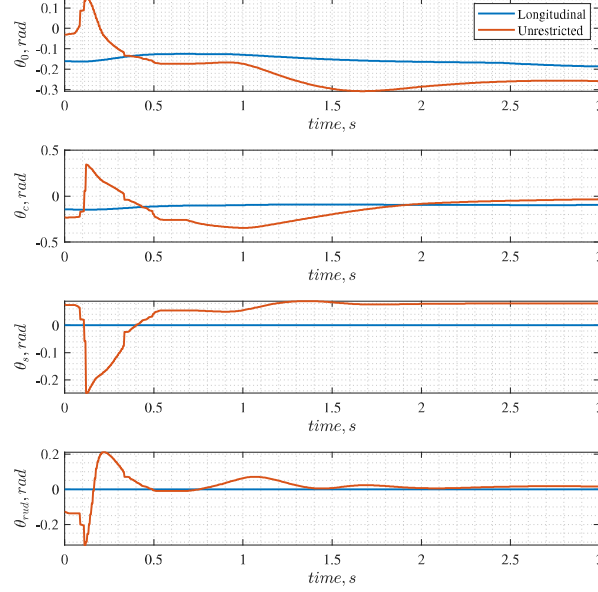


Figure 6.9. Controller inputs of a longitudinally restricted simulation and a full spatial 6-DOF simulation.

6.3 Rotor Speed Constraints

During AR entrance the constraints are not active. To test the rotor speed constraints, the simulation is started near the constraint boundaries with the reference trajectory placed beyond the boundary. Figures 6.10 - 6.12 show a longitudinal simulation with low initial sink rate and rotor speed. Initiating the rotor speed at $0.85\Omega_0$ and weighting w_b causes the low rotor speed constraint to become active. The controller wants to maintain the vehicle's initial condition, besides the low rotor speed, but the constraint violations results in w_b increasing significantly by pitching the vehicle up and using the forward speed to pass air through the rotor disks.

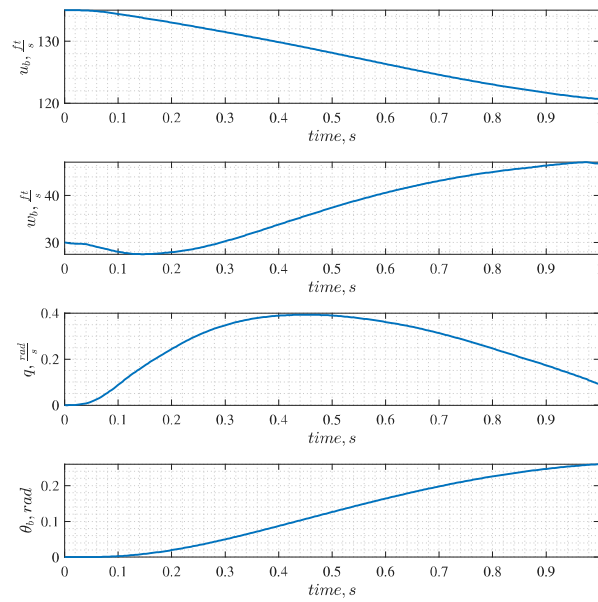


Figure 6.10. Longitudinal states of the vehicle starting near the low rotor speed constraint boundary.

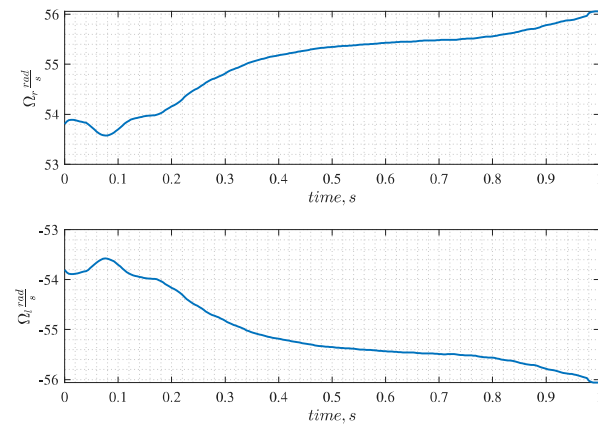


Figure 6.11. Rotor speed starting near the low rotor speed constraint boundary.

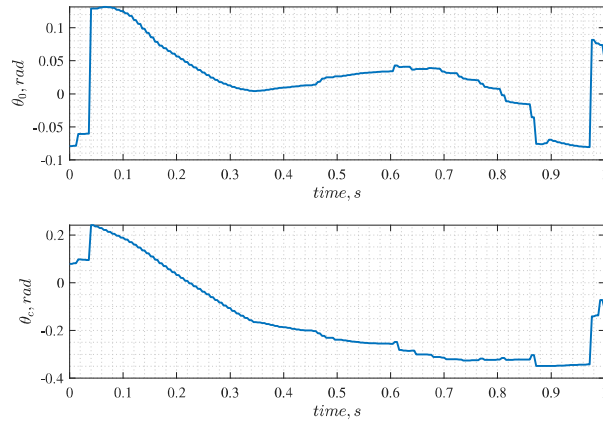


Figure 6.12. Control inputs starting near the low rotor speed constraint boundary.

Starting the simulation with a high w_b velocity and $1.15\Omega_0$ forces the upper rotor speed constraint to become active, shown in Figures 6.13-6.15. The controller maintains the vehicle's attitude and forward speed but significantly decreases w_b by putting in large amounts of collective, showing that the modification to Equation (4.28) works.

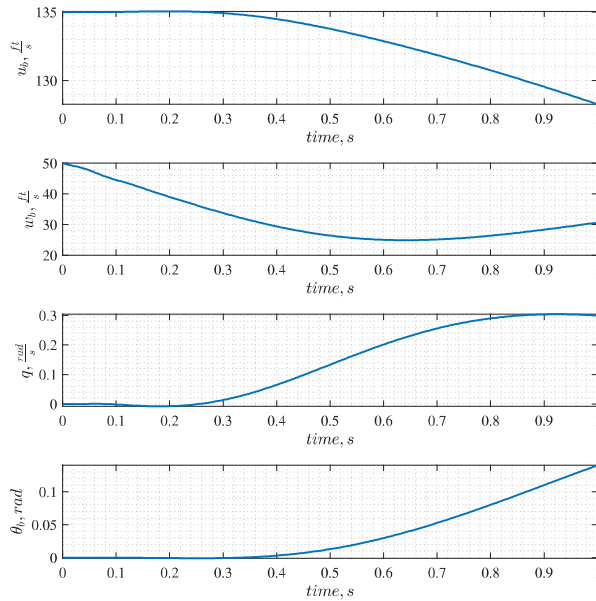


Figure 6.13. Longitudinal states of vehicle starting near the upper rotor speed constraint boundary.

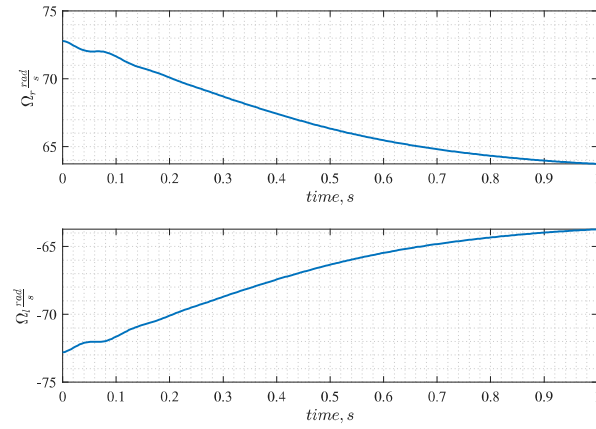


Figure 6.14. Rotor speed starting near the upper boundary.

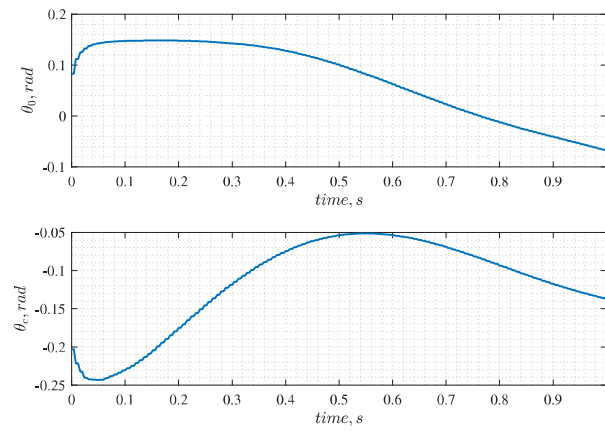


Figure 6.15. Control inputs near the upper rotor speed constraint boundary.

7. Conclusions and Future Work

A model predictive controller was used to control a tilt rotor craft in autorotation and forward flight. A reduced order model was developed utilizing quasi-static flapping equations, which increased the minimum time step for accurate simulation. Using the reduced order model to predict the vehicle states, a recurrent neural network was used to find the optimal control inputs given a reference trajectory. Constraints are placed on rotor speed and thrust, which ensure the vehicle will safely maneuver.

Based on linearized aerodynamics and neglecting nonuniform flow around the rotor disk, the quasi-static flapping equations accurately predict blade flapping in nominal operating conditions. During autorotation, when strong nonlinearities are present, the predictions begin to degrade but work well overall. A strong coupling between yaw rate and rolling moment exists within the high fidelity simulation due to the opposite directions of rotor rotation and geometry of the tilt rotor. The reduced order model does not accurately predict this coupling since the rotors are treated as disks are only concerned with the total *wind* axis velocity at the hub. No correction was added to the reduced order model to accommodate for the moment, but a simple linear coupling could capture the desired effect. The linearizing assumptions also lead to an error when the sink rate exceeds the induced inflow. A sign changing coefficient was added to the rotor torque derivative to ensure the rotor torque always increases with increasing thrust. Simulation results show that the controller accurately predicts violation of the upper rotor speed constraint, given a large sink rate, and implements the correct controls.

The number of discrete time steps increases computational cost significantly and usable performance required the prediction horizon to be at least 1 second. The inflow dynamics required the time step of the reduced order model to be at most 0.015 seconds where the high fidelity simulation, with individual

blade flapping, needed a time step smaller than 0.005 seconds. Using a static approximation of the inflow within the reduced order model could lead to significant increases in the time step. Solving for the static inflow requires a numerical nonlinear equation solver like Newton-Raphson iteration. The stability of the method at varying vehicle states would need to be thoroughly analyzed.

A recurrent neural network was used to solve the nonlinear optimization problem. The algorithm provides a unique capability of solving convex problems and a class of non-convex problems. The control affine system used within the prediction along with the quadratic cost function lead to the recurrent neural network globally converging to an optimal solution of the problem. The mathematical proof, while important for stability, does not guarantee the problem will be solved well. Implementing the optimization algorithm required the state and control weighting matrices, learning rate, constraint function, and convergence criteria all be correctly scaled. Numerical errors can occur if any one piece is incorrectly used. Alternatively, the convergence time of the controller is also significantly impacted. For instance, if the state weighting matrix is scaled low and the learning rate is low, the controller will behave in a stable way but convergence will take significantly longer. Conversely, if the state weighting matrix contains large values and the learning rate parameter is large, unstable oscillations in convergence can occur as the control inputs jump back and forth over the optimal value. Similarly, if the cost function is weighted significantly higher than the constraint function, convergence will likely occur when optimally of the cost function is satisfied with no regard to the constraint function.

A reference trajectory is used to compute control inputs. Often, optimal control is used to create a reference trajectory for another controller to follow. In this work, the reference trajectory is simply a time history of the controller goals, which lets it weight and compute each state and control input along the

horizon. In the controller simulations, the reference trajectory is a constant velocity in each axis, zero angular rates, zero Euler angles, and the nominal rotor speed. Inflow states are unweighted. A simple trajectory algorithm can be created that uses kinematics to plan a path to a specified landing spot, which the controller could then follow by computing the necessary control inputs. These control inputs can be used directly to control the vehicle or can be passed with the computed history of states to another controller that can better handle unmodeled dynamics and disturbances. Given the controller's poor performance in the lateral-directional planes and the decoupling of longitudinal inputs, the model prediction algorithm could be reduced to the longitudinal states where it can predict optimal inputs for collective and longitudinal stick, which largely control the rotor speed, and the lateral-directional control could be performed with a model reference adaptive controller, for example. Doing so would significantly increase the speed of the model predictive control algorithm and allow it to handle constraints while letting the adaptive controller deal with the highly nonlinear dynamics of lateral-directional motion.

REFERENCES

- Abbeel, P., Coates, A., Hunter, T., & Ng, A. (2008). *Autonomous autorotation of an rc helicopter*. Stanford University.
- Aponso, B., Bachelder, E., & Lee, D. (2005). Automated autorotation for unmanned rotorcraft recovery. AHS International Specialists' Meeting on Unmanned Rotorcraft.
- Ayanoglu, E. (1997). Data transmission when the sampling frequency exceeds the nyquist rate. *IEEE Communications Letters*, 1(6), 157–159. doi:10.1109/4234.649925
- Baruh, H. (1999). *Analytical dynamics*. McGraw-Hill.
- Bibik, P. & Narkiewicz, J. (2012). Helicopter optimal control after power failure using comprehensive dynamic model. *Journal of Guidance, Control, and Dynamics*, 35(4). doi:10.2514/1.51371
- Carlson, E. B. (1999). *Optimal tiltrotor aircraft operations during power failure* (Doctoral dissertation, University of Minnesota).
- Carlson, E. & Zhao, Y. (2002). Optimal short takeoff of tiltrotor aircraft in one engine failure. *Journal of Aircraft*, 39(2).
- Carlson, E. & Zhao, Y. (2003). Prediction of tiltrotor height-velocity diagrams using optimal control theory. *Journal of Aircraft*, 40(5).
- Carlson, E. & Zhao, Y. (2004). Optimal city-center takeoff operation of tiltrotor aircraft in one engine failure. *Journal of Aerospace Engineering*, 17(1). doi:10.1061/(ASCE)0893-1321(2004)17:1(26)
- Chen, R. (1989). A survey of nonuniform inflow models for rotorcraft flight dynamics and control applications. *15th European Rotorcraft Forum*. doi:19900006622
- Curry, M. (2002). Xv-15 tilt rotor. NASA Dryden XV-15 Tilt Rotor Graphics Collection.
- Dalamagkidis, K. (2009). *Autonomous vertical autorotation for unmanned helicopters* (Doctoral dissertation, University of South Florida).
- Dalamagkidis, K., Valavanis, K., & Piegl, L. (2010). Nonlinear model predictive control with neural network optimization for autonomous autorotation of small unmanned helicopters. *IEEE Transactions on Control Systems Technology*, 19(4), 818–831. doi:10.1109/TCST.2010.2054092

- Dooley, L. & Yearly, R. (1979). *Flight test evaluation of the high inertia rotor system* (tech. rep. No. ADA071648). Bell Helicopter Textron.
- Dubins, L. (1957). On curves of minimal length with a constraint on average curvature, and with a prescribed initial and terminal positions and tangents. *American Journal of Mathematics*, 79(3), 497–516.
- Dubins, L. (1961). On plane curves with curvature. *Pacific Journal of Mathematics*, 11(2), 471–481.
- Edenborough, H., Gaffey, T., & Weiberg, J. (1972). Analyses and tests confirm design of proprotor aircraft. *American Institute of Aeronautics and Astronautics*, (72-803).
- Etkin, B. (1972). *Dynamics of atmospheric flight* (3rd ed.). John Wiley & Sons.
- Felker, F., Betzina, M., & Signor, D. (1985). *Performance and loads data from a hover test of a full-scale xv-15 rotor* (tech. rep. No. 886833). NASA.
- Fliess, M. (1990). Generalized controller canonical forms for linear and nonlinear dynamics. *IEEE Transactions on Automatic Control*, 35(9).
- Gao, B., Hong, J., Yu, S., & Chen, H. (2017). Linear-quadratic output regulator with disturbance rejection: Application to vehicle launch control. IEEE American Control Conference. Seattle, WA.
- Grant, L. T. (2016). Certification of transport category rotorcraft. *Federal Aviation Administration*.
- Johnson, W. (1977). *Helicopter optimal descent and landing after power loss* (tech. rep. No. 19770019123). NASA.
- Kinderlehrer, D. & Stampacchia, G. (1980). *An introduction to variational inequalities and their applications*. Academic Press.
- Koyovis. (2017). Why is *vertical* autorotation in a helicopter not recommended? Aviation StackExchange. Retrieved from <https://aviation.stackexchange.com/questions/42969/why-is-vertical-autorotation-in-a-helicopter-not-recommended/42970>
- Lee, A. (1985). *Optimal landing of a helicopter in autorotation* (Doctoral dissertation, Stanford University).
- Lee, A. (1990). Optimal autorotational descent of a helicopter with control and state inequality constraints. *Journal of Guidance, Control, and Dynamics*, 13(5), 922–924. doi:10.2514/3.25420

- Lee, D. J. & Bang, H. (2008). Autonomous autorotation of an unmanned helicopter using a reinforcement learning algorithm. In *Aiaa guidance, navigation, and control*.
- Leishman, J. G. (2000). *Principles of helicopter aerodynamics* (1st ed.). Cambridge University Press.
- Leishman, J. G. (2006). *Principles of helicopter aerodynamics* (2nd ed.). Cambridge University Press.
- Maisel, M. D., Giulianetti, D. J., & Dugan, D. C. (2000). The history of the xv-15 tilt rotor research aircraft from concept to flight. In *Monographs in aerospace history #17*. The NASA History Series.
- McVicar, J. S. G. (1993). *A generic tilt-rotor simulation model with parallel implementation* (Doctoral dissertation, University of Glasgow).
- Murakami, Y. & Houston, S. S. (2009). Correction of the definition of mass flow parameter in dynamic inflow modelling. *Proceedings of the Institution of Mechanical Engineers Part G: Journal of Aerospace Engineering*, (223), 1037–1040. doi:10.1243/09544100JAERO604
- Padfield, G. (2018). *Helicopter flight dynamics* (3rd ed.). John Wiley & Sons.
- Pamadi, B. N. (2004). *Performance, stability, dynamics, and control of airplanes* (2nd ed.). AIAA Education Series.
- Peters, D. A. & HaQuang, N. (1988). Dynamic inflow for practical applications. *Journal of the American Helicopter Society*, 33. doi:10.4050/JAHS.33.64
- Taamallah, S., Bombois, X., & den Hof, P. V. (2017). Trajectory planning and trajectory tracking for a small-scale helicopter in autorotation. *Control Engineering Practice*, 58, 88–106.
- Tassa, Y., Mansard, N., & Todorov, E. (2014). Control limited differential dynamic programming. International conference on robotics and automation.
- Tierney, S. & Langelaan, J. (2010). Autorotation path planning using backwards reachable set and optimal control. Ahs forum.
- Watkins, C. (1989). *Learning from delayed rewards* (Doctoral dissertation, King's College).
- Wernli, A. & Cook, G. (1975). Suboptimal control for the nonlinear quadratic regulator problem. *Automatica*, 11, 75–84.

- Wheatley, J. B. (1935). An aerodynamic analysis of the autogiro rotor with a comparison between calculated and experimental results. *National Advisory Committee for Aeronautics*, (487). doi:19930091561
- Xia, Y., Feng, G., & Kamel, M. (2007). Development and analysis of a neural dynamical approach to nonlinear programming problems. *IEEE Transactions on Automatic Control*, 52(11), 2154–2159.
- Xia, Y., Feng, G., & Wang, J. (2008). A novel recurrent neural network for solving nonlinear optimization problems with inequality constraints. *IEEE Transactions on Neural Networks*, 19(8). doi:10.1109/TNN.2008.2000273
- Xia, Y., Leung, H., & Wang, J. (2002). A projection neural network and its application to constrained optimization problems. *IEEE Transactions on Circuits and Systems - I: Fundamental Theory and Applications*, 49(4), 447–458.
- Xia, Y. & Wang, J. (2004). A recurrent neural network for nonlinear convex optimization subject to nonlinear inequality constraints. *IEEE Transactions on Circuits and Systems - I: Regular Papers*, 51(7), 1385–1394.
- Xia, Y. & Wang, J. (2005). A recurrent neural network for solving nonlinear convex problems subject to linear constraints. *IEEE Transactions on Neural Networks*, 16(2), 379–386.
- Yomchinda, T. (2013). *Real-time path planning and autonomous control for helicopter autorotation* (Doctoral dissertation, Pennsylvania State University).
- Yomchinda, T., Horn, J. F., & Langelaan, J. W. (2012). Autonomous control and path planning for autorotaiton of unammed helicopters. In *68th american helicopter society international annual forum* (Vol. 4, pp. 2941–2955).
- Zhao, Y., Jhemi, A., & Chen, R. (1996). Optimal vertical takeoff and landing helicopter operation in one engine failure. *Journal of Aircraft*, 33(2).
- Zhao, Y., Sharma, V., Chen, R., Hindson, W., & Aiken, E. (1995). *Optimal category-a helicopter flight trajectories for operation from a clear heliport* (tech. rep. No. 20020011642). NASA.

A. APPENDICES

A.1 Reduced Order Model Jacobian

The recurrent neural network uses the systems gradient at each time step to find the optimal control. Calculating the derivative is costly as the state space is large. Deriving the underlying equations using the product and chain rules allows values to be stored and used multiple times for different derivatives, which helps speed the calculation.

A.1.1 Dynamics Function

The derivative of Eq. (4.41) must be evaluated to determine the effect of each control within the prediction horizon on the cost and constraint functions. The linear dynamics matrix $\frac{\partial f}{\partial \underline{x}}(\underline{x}, \underline{u})$ will be denoted as A and is given by:

$$A = \begin{bmatrix} \frac{\partial \dot{u}_b}{\partial \underline{x}} & \frac{\partial \dot{v}_b}{\partial \underline{x}} & \frac{\partial \dot{w}_b}{\partial \underline{x}} & \frac{\partial \dot{p}}{\partial \underline{x}} & \frac{\partial \dot{q}}{\partial \underline{x}} & \frac{\partial \dot{r}}{\partial \underline{x}} & \frac{\partial \dot{\phi}_b}{\partial \underline{x}} & \frac{\partial \dot{\theta}_b}{\partial \underline{x}} & \frac{\partial \dot{\Psi}_b}{\partial \underline{x}} & \frac{\partial \dot{x}_b}{\partial \underline{x}} & \frac{\partial \dot{y}_b}{\partial \underline{x}} & \frac{\partial \dot{z}_b}{\partial \underline{x}} \\ \frac{\partial \dot{\Omega}_r}{\partial \underline{x}} & \frac{\partial \dot{\Omega}_l}{\partial \underline{x}} & \frac{\partial \dot{\lambda}_{0r}}{\partial \underline{x}} & \frac{\partial \dot{\lambda}_{sr}}{\partial \underline{x}} & \frac{\partial \dot{\lambda}_{cr}}{\partial \underline{x}} & \frac{\partial \dot{\lambda}_{0l}}{\partial \underline{x}} & \frac{\partial \dot{\lambda}_{sl}}{\partial \underline{x}} & \frac{\partial \dot{\lambda}_{cl}}{\partial \underline{x}} \end{bmatrix}^T \quad (\text{A.1})$$

where each entry of A is a row vector and the state vector \underline{x} is given by :

$$\underline{x} = [u_b \ v_b \ w_b \ p \ q \ r \ \phi_b \ \theta_b \ \Psi_b \ x_b \ y_b \ z_b \ \Omega_r \ \Omega_l \ \lambda_{0r} \ \lambda_{sr} \ \lambda_{cr} \ \lambda_{0l} \ \lambda_{sl} \ \lambda_{cl}]^T. \quad (\text{A.2})$$

The rows of A are then found, assuming air density is constant with respect to altitude, in Eqs. (A.3) - (A.21):

$$\frac{\partial \dot{u}}{\partial \underline{x}} = \frac{1}{m} \begin{bmatrix} \frac{\partial F_x}{\partial u} & mr + \frac{\partial F_x}{\partial v} & -mq + \frac{\partial F_x}{\partial w} & \frac{\partial F_x}{\partial p} & -mw + \frac{\partial F_x}{\partial q} & mv + \frac{\partial F_x}{\partial r} \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{\partial F_x}{\partial \Omega_r} & \frac{\partial F_x}{\partial \Omega_l} & \frac{\partial F_x}{\partial \lambda_{0r}} & \frac{\partial F_x}{\partial \lambda_{sr}} & \frac{\partial F_x}{\partial \lambda_{cr}} & \frac{\partial F_x}{\partial \lambda_{0l}} & \frac{\partial F_x}{\partial \lambda_{sl}} & \frac{\partial F_x}{\partial \lambda_{cl}} \end{bmatrix}^T \quad (\text{A.3})$$

$$\begin{aligned} \frac{\partial \dot{\Psi}_b}{\partial \underline{x}} = & \begin{bmatrix} 0 & 0 & 0 & 0 & \sin \phi_b \sec \theta_b & \cos \phi_b \sec \theta_b & q \cos \phi_b \sec \theta_b - r \sin \phi_b \sec \theta_b \\ q \sin \phi_b \tan \theta_b \sec \theta_b + r \cos \phi_b \tan \theta_b \sec \theta_b & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \end{aligned} \quad (\text{A.10})$$

$$\begin{aligned} \frac{\partial \dot{x}}{\partial \underline{x}} = & \begin{bmatrix} \cos \theta_b \cos \Psi_b & \cos \Psi_b \sin \phi_b \sin \theta_b - \sin \Psi_b \cos \phi_b \\ \sin \Psi_b \sin \psi_b + \cos \Psi_b \cos \phi_b \sin \theta_b & 0 & 0 & 0 \\ v (\cos \Psi_b \cos \phi_b \sin \theta_b + \sin \Psi_b \sin \phi_b) + w (\sin \Psi_b \cos \phi_b - \cos \Psi_b \sin \phi_b \sin \theta_b) \\ -u \sin \theta_b \cos \Psi_b + v \cos \Psi_b \sin \phi_b \cos \theta_b + w \cos \Psi_b \cos \phi_b \cos \theta_b \\ -u_b \cos \theta_b \sin \Psi_b - v_b (\sin \Psi_b \sin \phi_b \sin \theta_b + \cos \Psi_b \cos \phi_b) \\ + w_b (\cos \Psi_b \sin \phi_b - \sin \Psi_b \cos \phi_b \sin \theta_b) \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \end{aligned} \quad (\text{A.11})$$

$$\begin{aligned} \frac{\partial \dot{y}}{\partial \underline{x}} = & \begin{bmatrix} \sin \Psi_b \cos \theta_b & \cos \Psi_b \cos \phi_b + \sin \Psi_b \sin \phi_b \sin \theta_b \\ \sin \Psi_b \cos \phi_b \sin \theta_b - \cos \Psi_b \sin \phi_b & 0 & 0 & 0 \\ v_b (\sin \Psi_b \cos \phi_b \sin \theta_b - \cos \Psi_b \sin \phi_b) - w_b (\sin \Psi_b \sin \phi_b \sin \theta_b + \cos \Psi_b \cos \phi_b) \\ -u_b \sin \theta_b \cos \Psi_b + v_b \sin \Psi_b \sin \phi_b \cos \theta_b + w \sin \Psi_b \cos \phi_b \cos \theta_b \\ u_b \cos \Psi_b \cos \theta_b + v_b (\cos \Psi_b \sin \phi_b \sin \theta_b - \sin \Psi_b \cos \phi_b) \\ + w_b (\cos \Psi_b \cos \phi_b \sin \theta_b + \sin \Psi_b \sin \phi_b) \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \end{aligned} \quad (\text{A.12})$$

$$\frac{\partial \dot{z}}{\partial \underline{x}} = \begin{bmatrix} -\sin \theta_b & \sin \phi_b \cos \theta_b & \cos \phi_b \cos \theta_b & 0 & 0 & 0 & v_b \cos \phi_b \cos \theta_b - w_b \sin \phi_b \cos \theta_b \\ -u_b \cos \theta_b + v_b \sin \phi_b \sin \theta_b - w \cos \phi_b \sin \theta_b & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \quad (\text{A.13})$$

$$\frac{\partial \dot{\Omega}_r}{\partial \underline{x}} = \frac{\rho A V_{Tr}^2 R}{3 I_B} \begin{bmatrix} \frac{\partial C_{Qr}}{\partial u_b} & \frac{\partial C_{Qr}}{\partial v_b} & \frac{\partial C_{Qr}}{\partial w_b} & \frac{\partial C_{Qr}}{\partial p} & \frac{\partial C_{Qr}}{\partial q} & \frac{\partial C_{Qr}}{\partial r} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{\partial C_{Qr}}{\partial \Omega_r} & 0 & \frac{\partial C_{Qr}}{\partial \lambda_{0r}} & \frac{\partial C_{Qr}}{\partial \lambda_{sr}} & \frac{\partial C_{Qr}}{\partial \lambda_{cr}} & 0 & 0 & 0 \end{bmatrix}^T \quad (\text{A.14})$$

$$\frac{\partial \dot{\Omega}_l}{\partial \underline{x}} = \frac{\rho A V_{Tl}^2 R}{3 I_B} \begin{bmatrix} \frac{\partial C_{Ql}}{\partial u_b} & \frac{\partial C_{Ql}}{\partial v_b} & \frac{\partial C_{Ql}}{\partial w_b} & \frac{\partial C_{Ql}}{\partial p} & \frac{\partial C_{Ql}}{\partial q} & \frac{\partial C_{Ql}}{\partial r} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{\partial C_{Ql}}{\partial \Omega_l} & 0 & 0 & 0 & \frac{\partial C_{Ql}}{\partial \lambda_{0l}} & \frac{\partial C_{Ql}}{\partial \lambda_{sl}} & \frac{\partial C_{Ql}}{\partial \lambda_{cl}} \end{bmatrix}^T \quad (\text{A.15})$$

$$\frac{\partial \dot{\lambda}_{0r}}{\partial \underline{x}} = \begin{bmatrix} \frac{\partial \dot{\lambda}_{0r}}{\partial u_b} & \frac{\partial \dot{\lambda}_{0r}}{\partial v_b} & \frac{\partial \dot{\lambda}_{0r}}{\partial w_b} & \frac{\partial \dot{\lambda}_{0r}}{\partial p} & \frac{\partial \dot{\lambda}_{0r}}{\partial q} & \frac{\partial \dot{\lambda}_{0r}}{\partial r} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{\partial \dot{\lambda}_{0r}}{\partial \Omega_r} & 0 & \frac{\partial \dot{\lambda}_{0r}}{\partial \lambda_{0r}} & \frac{\partial \dot{\lambda}_{0r}}{\partial \lambda_{sr}} & \frac{\partial \dot{\lambda}_{0r}}{\partial \lambda_{cr}} & 0 & 0 & 0 \end{bmatrix}^T \quad (\text{A.16})$$

$$\frac{\partial \dot{\lambda}_{sr}}{\partial \underline{x}} = \begin{bmatrix} \frac{\partial \dot{\lambda}_{sr}}{\partial u_b} & \frac{\partial \dot{\lambda}_{sr}}{\partial v_b} & \frac{\partial \dot{\lambda}_{sr}}{\partial w_b} & \frac{\partial \dot{\lambda}_{sr}}{\partial p} & \frac{\partial \dot{\lambda}_{sr}}{\partial q} & \frac{\partial \dot{\lambda}_{sr}}{\partial r} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{\partial \dot{\lambda}_{sr}}{\partial \Omega_r} & 0 & \frac{\partial \dot{\lambda}_{sr}}{\partial \lambda_{0r}} & \frac{\partial \dot{\lambda}_{sr}}{\partial \lambda_{sr}} & \frac{\partial \dot{\lambda}_{sr}}{\partial \lambda_{cr}} & 0 & 0 & 0 \end{bmatrix}^T \quad (\text{A.17})$$

$$\frac{\partial \dot{\lambda}_{cr}}{\partial \underline{x}} = \begin{bmatrix} \frac{\partial \dot{\lambda}_{cr}}{\partial u_b} & \frac{\partial \dot{\lambda}_{cr}}{\partial v_b} & \frac{\partial \dot{\lambda}_{cr}}{\partial w_b} & \frac{\partial \dot{\lambda}_{cr}}{\partial p} & \frac{\partial \dot{\lambda}_{cr}}{\partial q} & \frac{\partial \dot{\lambda}_{cr}}{\partial r} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{\partial \dot{\lambda}_{cr}}{\partial \Omega_r} & 0 & \frac{\partial \dot{\lambda}_{cr}}{\partial \lambda_{0r}} & \frac{\partial \dot{\lambda}_{cr}}{\partial \lambda_{sr}} & \frac{\partial \dot{\lambda}_{cr}}{\partial \lambda_{cr}} & 0 & 0 & 0 \end{bmatrix}^T \quad (\text{A.18})$$

$$\frac{\partial \dot{\lambda}_{ol}}{\partial \underline{x}} = \begin{bmatrix} \frac{\partial \dot{\lambda}_{ol}}{\partial u_b} & \frac{\partial \dot{\lambda}_{ol}}{\partial v_b} & \frac{\partial \dot{\lambda}_{ol}}{\partial w_b} & \frac{\partial \dot{\lambda}_{ol}}{\partial p} & \frac{\partial \dot{\lambda}_{ol}}{\partial q} & \frac{\partial \dot{\lambda}_{ol}}{\partial r} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{\partial \dot{\lambda}_{ol}}{\partial \Omega_l} & 0 & 0 & 0 & \frac{\partial \dot{\lambda}_{ol}}{\partial \lambda_{0l}} & \frac{\partial \dot{\lambda}_{ol}}{\partial \lambda_{sl}} & \frac{\partial \dot{\lambda}_{ol}}{\partial \lambda_{cl}} \end{bmatrix}^T \quad (\text{A.19})$$

$$\frac{\partial \dot{\lambda}_{sl}}{\partial \underline{x}} = \begin{bmatrix} \frac{\partial \dot{\lambda}_{sl}}{\partial u_b} & \frac{\partial \dot{\lambda}_{sl}}{\partial v_b} & \frac{\partial \dot{\lambda}_{sl}}{\partial w_b} & \frac{\partial \dot{\lambda}_{sl}}{\partial p} & \frac{\partial \dot{\lambda}_{sl}}{\partial q} & \frac{\partial \dot{\lambda}_{sl}}{\partial r} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{\partial \dot{\lambda}_{sl}}{\partial \Omega_l} & 0 & 0 & 0 & \frac{\partial \dot{\lambda}_{sl}}{\partial \lambda_{0l}} & \frac{\partial \dot{\lambda}_{sl}}{\partial \lambda_{sl}} & \frac{\partial \dot{\lambda}_{sl}}{\partial \lambda_{cl}} \end{bmatrix}^T \quad (\text{A.20})$$

$$\frac{\partial \dot{\lambda}_{cl}}{\partial \underline{x}} = \begin{bmatrix} \frac{\partial \dot{\lambda}_{cl}}{\partial u_b} & \frac{\partial \dot{\lambda}_{cl}}{\partial v_b} & \frac{\partial \dot{\lambda}_{cl}}{\partial w_b} & \frac{\partial \dot{\lambda}_{cl}}{\partial p} & \frac{\partial \dot{\lambda}_{cl}}{\partial q} & \frac{\partial \dot{\lambda}_{cl}}{\partial r} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{\partial \dot{\lambda}_{cl}}{\partial \Omega_l} & 0 & 0 & 0 & \frac{\partial \dot{\lambda}_{cl}}{\partial \lambda_{0l}} & \frac{\partial \dot{\lambda}_{cl}}{\partial \lambda_{sl}} & \frac{\partial \dot{\lambda}_{cl}}{\partial \lambda_{cl}} \end{bmatrix}^T \quad (\text{A.21})$$

A.1.2 Hub Velocities and Sideslip

Rotor dynamics are driven largely by the aerodynamic states at the rotor hubs due to their velocities. The hub velocities depend on not only the vehicle's *body* axis velocities but also the angular rates. Since, the *hub* axes systems are aligned

with the *body* axes, their relation is straightforward:

$$\frac{\partial u_h}{\partial u_b} = \frac{\partial v_h}{\partial v_b} = \frac{\partial w_h}{\partial w_b} = 1 \quad (\text{A.22})$$

The angular rates affect *hub* velocities through \underline{R}_h yielding:

$$\frac{\partial \underline{v}_h}{\partial \underline{\omega}_b} = \begin{bmatrix} 0 & z_h & -y_h \\ -z_h & 0 & x_h \\ y_h & -x_h & 0 \end{bmatrix} \quad (\text{A.23})$$

Quasi-static flapping equations are cast in the rotor system *wind* axes leading to cleaner derivatives when evaluating the advance ratio μ_w . The effect of *hub* axis velocities on μ_w are given by:

$$\frac{\partial \mu_w}{\partial u_h} = \frac{u_h}{V_T \sqrt{u_h^2 + v_h^2}} \quad (\text{A.24})$$

$$\frac{\partial \mu_w}{\partial v_h} = \frac{v_h}{V_T \sqrt{u_h^2 + v_h^2}} \quad (\text{A.25})$$

While the total in plane velocity has the largest effect on longitudinal and lateral flapping, the normal flow through the rotor disk produces thrust and w_h impacts that through:

$$\frac{\partial \xi_h}{\partial w_h} = \frac{1}{V_T} \quad (\text{A.26})$$

Lastly, transforming the quasi-static flapping equations into the *hub* axes requires the rotor sideslip angle. Hub velocities affect the sideslip with:

$$\frac{\partial \Psi_w}{\partial u_h} = \frac{-v_h}{u_h^2 + v_h^2} \quad (\text{A.27})$$

$$\frac{\partial \Psi_w}{\partial v_h} = \frac{u_h}{u_h^2 + v_h^2} \quad (\text{A.28})$$

Rotor hub velocity derivatives are undefined if the *hub* axes have no longitudinal or lateral velocity. This can occur if the vehicle stops moving or rotates such that the *hub* velocities induced by *body* angular rates are equal and opposite to the *body* velocities. During normal flight and autorotational descent these issues will not appear, but during landing, when the vehicle is moving slowly issues could appear.

A.1.3 Wind axis transformations

Transformation to and from the *hub* and *wind* axes are determined by the linear *hub* velocities. The rotor sideslip (Ψ_w) dependencies on hub velocities are given by Eqs. (A.27) and (A.28), which can be used with following equations to find the derivatives of the transformations:

$$\frac{\partial^\theta T_h^w}{\partial \Psi_w} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -\sin \Psi_w & \cos \Psi_w \\ 0 & 0 & -\cos \Psi_w & -\sin \Psi_w \end{bmatrix} \quad (\text{A.29})$$

$$\frac{\partial^\lambda T_h^w}{\partial \Psi_w} = \frac{\partial^\beta T_w^h}{\partial \Psi_w} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -\sin \Psi_w & \cos \Psi_w \\ 0 & -\cos \Psi_w & -\sin \Psi_w \end{bmatrix} \quad (\text{A.30})$$

$$\frac{\partial^\omega T_h^w}{\partial \Psi_w} = \begin{bmatrix} -\sin \Psi_w & \cos \Psi_w & 0 & 0 \\ -\cos \Psi_w & -\sin \Psi_w & 0 & 0 \\ 0 & 0 & -\sin \Psi_w & \cos \Psi_w \\ 0 & 0 & -\cos \Psi_w & -\sin \Psi_w \end{bmatrix} \quad (\text{A.31})$$

A.1.4 Thrust Coefficient

Rotor thrust depends on the in-plane velocities as well as the normal velocities. The rotor must induce some flow through itself to create thrust but increases in

the inflow cause decreases in thrust as the average angle of attack on the blade is decreased. Increasing the downward hub velocity increases the thrust from the opposite effect. The derivatives of the rotor thrust are given by:

$$\frac{\partial C_T}{\partial \mu_w} = \frac{a_0 s}{2} \left(\hat{\theta}_0 \mu_w + \frac{1}{2} \left(\hat{\theta}_{sw} + \frac{\bar{p}_w}{2} \right) + \frac{\mu_w}{2} \hat{\theta}_{tw} \right) \quad (\text{A.32})$$

$$\frac{\partial C_T}{\partial \hat{\theta}_0} = \frac{a_0 s}{2} \left(\frac{1}{3} + \frac{\mu_w^2}{2} \right) \quad (\text{A.33})$$

$$\frac{\partial C_T}{\partial \hat{\theta}_{sw}} = \frac{a_0 s \mu_w}{4} \quad (\text{A.34})$$

$$\frac{\partial C_T}{\partial \bar{p}_w} = \frac{a_0 s \mu_w}{8} \quad (\text{A.35})$$

$$\frac{\partial C_T}{\partial \xi_h} = \frac{a_0 s \mu_w}{4} \quad (\text{A.36})$$

$$\frac{\partial C_T}{\partial \lambda_0} = -\frac{a_0 s \mu_w}{4} \quad (\text{A.37})$$

Using the chain rule and product rule, combinations of the thrust derivatives above and other independent derivatives will yield the relation between rotor thrust and the vehicle states:

$$\frac{\partial C_T}{\partial u_h} = \frac{\partial C_T}{\partial \mu_w} \frac{\partial \mu_w}{\partial u_h} + \frac{\partial C_T}{\partial \hat{\theta}_{sw}} \frac{\partial \hat{\theta}_{sw}}{\partial u_h} + \frac{\partial C_T}{\partial \bar{p}_w} \frac{\partial \bar{p}_w}{\partial u_h} \quad (\text{A.38})$$

$$\frac{\partial C_T}{\partial v_h} = \frac{\partial C_T}{\partial \mu_w} \frac{\partial \mu_w}{\partial v_h} + \frac{\partial C_T}{\partial \hat{\theta}_{sw}} \frac{\partial \hat{\theta}_{sw}}{\partial v_h} + \frac{\partial C_T}{\partial \bar{p}_w} \frac{\partial \bar{p}_w}{\partial v_h} \quad (\text{A.39})$$

$$\frac{\partial C_T}{\partial w_h} = \frac{\partial C_T}{\partial \xi_h} \frac{\partial \xi_h}{\partial w_h} \quad (\text{A.40})$$

$$\frac{\partial C_T}{\partial p} = \frac{\partial C_T}{\partial v_h} \frac{\partial v_h}{\partial p} + \frac{\partial C_T}{\partial w_h} \frac{\partial w_h}{\partial p} + \frac{\partial C_T}{\partial \bar{p}_w} \frac{\partial \bar{p}_w}{\partial p} \quad (\text{A.41})$$

$$\frac{\partial C_T}{\partial q} = \frac{\partial C_T}{\partial u_h} \frac{\partial u_h}{\partial q} + \frac{\partial C_T}{\partial w_h} \frac{\partial w_h}{\partial q} + \frac{\partial C_T}{\partial \bar{p}_w} \frac{\partial \bar{p}_w}{\partial q} \quad (\text{A.42})$$

$$\frac{\partial C_T}{\partial r} = \frac{\partial C_T}{\partial u_h} \frac{\partial u_h}{\partial r} + \frac{\partial C_T}{\partial v_h} \frac{\partial v_h}{\partial r} \quad (\text{A.43})$$

$$\frac{\partial C_T}{\partial \Omega} = \frac{\partial C_T}{\partial \mu_w} \frac{\partial \mu_w}{\partial \Omega} + \frac{\partial C_T}{\partial \bar{p}_w} \frac{\partial \bar{p}_w}{\partial \Omega} + \frac{\partial C_T}{\partial \xi_h} \frac{\partial \xi_h}{\partial \Omega} + \frac{\partial C_T}{\partial \lambda_0} \frac{\partial \lambda_0}{\partial \Omega} \quad (\text{A.44})$$

$$\frac{\partial C_T}{\partial \lambda_s} = \frac{\partial C_T}{\partial \lambda_c} = 0 \quad (\text{A.45})$$

A.1.5 Torque Coefficient

The rotor torque is handled similarly to the thrust. Taking the derivative of (4.26) in terms of its own variables, then using the chain rule and product rule, will give the derivative of the rotor torque in terms of the vehicle's states. Section 4.2.1.4 explains the modification to Eq.(A.47). The derivatives of (4.26) with respect to its own variables are given as:

$$\frac{\partial C_Q}{\partial \mu_w} = C_{xw} + \frac{3}{4} \bar{C}_d s \mu_w \quad (\text{A.46})$$

$$\frac{\partial C_Q}{\partial C_T} = \text{sig}(\xi_h - \lambda_0) (\xi_h - \lambda_0) \quad (\text{A.47})$$

$$\frac{\partial C_Q}{\partial \xi_h} = -C_T \quad (\text{A.48})$$

$$\frac{\partial C_Q}{\partial \lambda_0} = C_T \quad (\text{A.49})$$

$$\frac{\partial C_Q}{\partial C_{xw}} = \mu_w \quad (\text{A.50})$$

A.1.6 Inflow

The rotor inflow depends on the *hub* velocities and forces as well at the inflow states themselves. Velocity effects can be found with:

$$\frac{\partial \dot{\underline{\lambda}}}{\partial \mu_w} = M_d^{-1} \left(\frac{\partial \underline{F}_{aero}}{\partial \mu_w} - \frac{\partial \hat{L}^{-1}}{\partial \mu_w} \underline{\lambda} \right) \quad (\text{A.51})$$

where $\underline{F}_{aero} = [C_T \quad -C_l \quad C_M]^T$ and $\underline{\lambda} = [\lambda_0 \quad \lambda_s \quad \lambda_c]^T$. The normal inflow affects the inflow dynamics through:

$$\frac{\partial \dot{\underline{\lambda}}}{\partial \lambda_0} = M_d^{-1} \left(\frac{\partial \underline{F}_{aero}}{\partial \lambda_0} - \frac{\partial \hat{L}^{-1}}{\partial \lambda_0} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \right) \quad (\text{A.52})$$

The lateral inflow affects the inflow dynamics through:

$$\frac{\partial \dot{\underline{\lambda}}}{\partial \lambda_s} = M_d^{-1} \left(\frac{\partial \underline{F}_{aero}}{\partial \lambda_s} - \hat{L}^{-1} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \right) \quad (\text{A.53})$$

Longitudinal effects can be found similarly. Vehicle states that do not directly appear within the velocities or inflow can affect the inflow dynamics through the forcing term. The forcing term's effect is given by:

$$\frac{\partial \dot{\underline{\lambda}}}{\partial \underline{F}_{aero}} = M_d^{-1} \quad (\text{A.54})$$

The inflow dynamics are derived in the *wind* axes, then transformed to the *hub* axes. The transformation of the dynamics gain matrix depends on Ψ_w while the matrix itself depends on the wake skew angle (χ). The mass flow parameter (V) depends directly on *hub* velocities and the normal inflow. The dynamics gain

matrix derivative can be taken as:

$$\frac{\partial \hat{L}^{-1}}{\partial \mu_w} = \frac{\partial V}{\partial \mu_w} {}^\lambda T_w^h {}^T \tilde{L}^{-1} {}^\lambda T_w^h + V \frac{\partial {}^\lambda T_w^h {}^T}{\partial \mu_w} \tilde{L}^{-1} {}^\lambda T_w^h + V {}^\lambda T_w^h {}^T \frac{\partial \tilde{L}^{-1}}{\partial \mu_w} {}^\lambda T_w^h + V {}^\lambda T_w^h {}^T \tilde{L}^{-1} \frac{\partial {}^\lambda T_w^h}{\partial \mu_w} \quad (\text{A.55})$$

$$\frac{\partial \hat{L}^{-1}}{\partial \lambda_0} = \frac{\partial V}{\partial \lambda_0} {}^\lambda T_w^h {}^T \tilde{L}^{-1} {}^\lambda T_w^h + V {}^\lambda T_w^h {}^T \frac{\partial \tilde{L}^{-1}}{\partial \lambda_0} {}^\lambda T_w^h \quad (\text{A.56})$$

$$\frac{\partial \hat{L}^{-1}}{\partial \xi_h} = \frac{\partial V}{\partial \xi_h} {}^\lambda T_w^h {}^T \tilde{L}^{-1} {}^\lambda T_w^h + V {}^\lambda T_w^h {}^T \frac{\partial \tilde{L}^{-1}}{\partial \xi_h} {}^\lambda T_w^h \quad (\text{A.57})$$

The derivative of the mass flow parameter is taken as:

$$\frac{\partial V}{\partial \mu_w} = \begin{bmatrix} \frac{\partial V_t}{\partial \mu_w} & 0 & 0 \\ 0 & \frac{\partial V_m}{\partial \mu_w} & 0 \\ 0 & 0 & \frac{\partial V_m}{\partial \mu_w} \end{bmatrix} \quad (\text{A.58})$$

An equivalent process is used for ξ_h and λ_0 . The total inflow (V_t) derivatives are given by:

$$\frac{\partial V_t}{\partial \mu_w} = \frac{\mu_w}{V_t} \quad (\text{A.59})$$

$$\frac{\partial V_t}{\partial \lambda_0} = \frac{\lambda_0 - \xi_h}{V_t} \quad (\text{A.60})$$

$$\frac{\partial V_t}{\partial \xi_h} = -\frac{\lambda_0 - \xi_h}{V_t} \quad (\text{A.61})$$

The harmonic inflow component can be written as:

$$V_m = \frac{V_J}{V_t} \quad (\text{A.62})$$

which eases the differentiation process. The derivative is then taken as:

$$\frac{\partial V_m}{\partial \mu_w} = \frac{\frac{\partial V_J}{\partial \mu_w} V_t - V_J \frac{\partial V_t}{\partial \mu_w}}{V_t^2} \quad (\text{A.63})$$

where the derivatives of the auxiliary term (V_J) are taken as:

$$\frac{\partial V_J}{\partial \mu_w} = 2\mu_w \quad (\text{A.64})$$

$$\frac{\partial V_J}{\partial \lambda_0} = |\lambda_0 - \xi_h| + \lambda_0 \operatorname{sign}(\lambda_0 - \xi_h) + 2\lambda_0 - 2\xi_h \quad (\text{A.65})$$

$$\frac{\partial V_J}{\partial \xi_h} = -\lambda_0 \operatorname{sign}(\lambda_0 - \xi_h) - 2\lambda_0 + 2\xi_h \quad (\text{A.66})$$

The dynamics gain matrix is inverted for use in the inflow model. The derivative of the inverse can be expanded as:

$$\tilde{L}\tilde{L}^{-1} = I \quad (\text{A.67})$$

$$\frac{\partial \tilde{L}}{\partial \mu_w} \tilde{L}^{-1} + \tilde{L} \frac{\partial \tilde{L}^{-1}}{\partial \mu_w} = 0 \quad (\text{A.68})$$

$$\frac{\partial \tilde{L}^{-1}}{\partial \mu_w} = -\tilde{L}^{-1} \frac{\partial \tilde{L}}{\partial \mu_w} \tilde{L}^{-1} \quad (\text{A.69})$$

Then the derivative of \tilde{L} itself can be taken with respect to the wake skew angle as:

$$\frac{\partial \tilde{L}}{\partial \chi} = \begin{bmatrix} 0 & 0 & \frac{\partial L_{13}}{\partial \chi} \\ 0 & \frac{\partial L_{22}}{\partial \chi} & 0 \\ \frac{\partial L_{31}}{\partial \chi} & 0 & \frac{\partial L_{33}}{\partial \chi} \end{bmatrix} \quad (\text{A.70})$$

The components are related to each other with:

$$\frac{\partial L_{13}}{\partial \chi} = -\frac{\partial L_{31}}{\partial \chi} = \frac{15\pi}{64} \frac{\cos \chi}{\sqrt{1 - \sin \chi} (1 + \sin \chi)^{3/2}} \quad (\text{A.71})$$

$$\frac{\partial L_{22}}{\partial u_h} = -\frac{\partial L_{33}}{\partial u_h} = -\frac{4 \cos \chi}{(1 + \sin \chi)^2} \quad (\text{A.72})$$

Finally, the wake skew angle depends on the velocities and inflow as:

$$\frac{\partial \chi}{\partial \mu_w} = -\frac{|\lambda_0 - \xi_h|}{V_t^2} \quad (\text{A.73})$$

$$\frac{\partial \chi}{\partial \lambda_0} = \frac{\mu_w \text{sign}(\lambda_0 - \xi_w)}{V_t^2} \quad (\text{A.74})$$

$$\frac{\partial \chi}{\partial \xi_h} = -\frac{\mu_w \text{sign}(\lambda_0 - \xi_w)}{V_t^2} \quad (\text{A.75})$$

A.1.7 Flapping - Hub Velocities

Quasi-static flapping depends heavily on the rotor advance ratio. The derivative of flapping in the *wind* axes is taken as:

$$\begin{aligned} \frac{\partial \beta_w}{\partial \mu_w} = & \frac{\partial A_{\beta\theta}}{\partial \mu_w} T_h^w \underline{\theta}_h + A_{\beta\theta} \frac{\partial T_h^w}{\partial \mu_w} \underline{\theta}_h + \frac{\partial A_{\beta\lambda}}{\partial \mu_w} T_h^w \underline{\lambda}_h \\ & + A_{\beta\lambda} \frac{\partial T_h^w}{\partial \mu_w} \underline{\lambda}_h + \frac{\partial A_{\beta\omega}}{\partial \mu_w} T_h^w \underline{\omega}_h + A_{\beta\omega} \frac{\partial T_h^w}{\partial \mu_w} \underline{\omega}_h \end{aligned} \quad (\text{A.76})$$

Transforming the *wind* axes derivative to the *hub* axes is done using:

$$\frac{\partial \underline{\beta}_h}{\partial \mu_w} = \frac{\partial T_w^h}{\partial \mu_w} \underline{\beta}_w + T_w^h \frac{\partial \underline{\beta}_w}{\partial \mu_w} \quad (\text{A.77})$$

The derivative of the control matrix coefficient are then found as:

$$\frac{\partial A_{\beta\theta}}{\partial \mu_w} = \frac{\gamma}{8\lambda_\beta^2} \begin{bmatrix} a_{11}^* & a_{12}^* & a_{13}^* & 0 \\ a_{21}^* & a_{22}^* & a_{23}^* & a_{24}^* \\ a_{31}^* & a_{32}^* & a_{33}^* & a_{34}^* \end{bmatrix} \quad (\text{A.78})$$

where τ^* denotes the derivative of τ with respect to μ_w .

$$a_{11}^* = 2\mu_w$$

$$a_{12}^* = \frac{4}{3}\mu_w$$

$$a_{13}^* = \frac{4}{3}$$

$$a_{21}^* = \eta_\beta \frac{4}{3} \left(S_\beta (1 + \mu_w^2) + \frac{16\lambda_\beta^2}{\gamma} \left(1 + \frac{\mu_w^2}{2} \right) \right) + \eta_\beta \frac{4}{3} \mu_w \left(2S_\beta \mu_w + \frac{16\lambda_\beta^2}{\gamma} \mu_w \right)$$

$$a_{22}^* = 2\eta_\beta \left(\frac{8\lambda_\beta^2}{\gamma} \left(1 + \frac{\mu_w^2}{2} \right) + \frac{8}{15} S_\beta \left(1 + \frac{5}{2} \mu_w^2 \right) \right) + 2\eta_\beta \mu_w \left(\frac{8\lambda_\beta^2}{\gamma} \mu_w + \frac{8}{3} S_\beta \mu_w \right)$$

$$a_{23}^* = \eta_\beta \left(\frac{32\lambda_\beta^2}{\gamma} \mu_w + \frac{32}{9} \mu_w S_\beta \right)$$

$$a_{24}^* = -\eta_\beta S_\beta \frac{8\lambda_\beta^2}{\gamma} \mu_w$$

$$a_{31}^* = \eta_\beta \frac{4}{3} \left(1 + \frac{\mu_w^2}{2} - 2S_\beta \frac{8\lambda_\beta^2}{\gamma} \right) + \eta_\beta \frac{4}{3} \mu_w^2$$

$$a_{32}^* = \eta_\beta 2 \left(\frac{8}{15} \left(1 + \frac{\mu_w^2}{3} \right) - S_\beta \frac{8\lambda_\beta^2}{\gamma} \right) + \frac{32}{45} \eta_\beta \mu_w^2$$

$$a_{33}^* = \eta_\beta \mu_w \left(\frac{32}{9} - S_\beta \frac{24\lambda_\beta^2}{\gamma} \right)$$

$$a_{34}^* = \eta_\beta \frac{16\lambda_\beta^2}{\gamma} \mu_w^3$$

Similarly, the derivative of the inflow matrix coefficient is given as:

$$\frac{\partial A_{\beta\lambda}}{\partial \mu_w} = \frac{\gamma}{8\lambda_\beta^2} \begin{bmatrix} 0 & b_{12}^* & 0 \\ b_{21}^* & b_{22}^* & 0 \\ b_{31}^* & b_{32}^* & b_{33}^* \end{bmatrix} \quad (\text{A.79})$$

$$b_{12}^* = -\frac{2}{3}$$

$$b_{21}^* = \eta_\beta \left(\left(\frac{4}{3} \right)^2 S_\beta + \frac{16\lambda_\beta^2}{\gamma} \left(1 + \frac{\mu_w^2}{2} \right) \right) + \eta_\beta \frac{16\lambda_\beta^2}{\gamma} \mu_w^2$$

$$b_{22}^* = -\eta_\beta \left(\frac{8\lambda_\beta^2}{\gamma} + \frac{16}{9} S_\beta \right) \mu_w$$

$$b_{31}^* = \eta_\beta \left(\left(\frac{4}{3} \right)^2 \left(1 - \frac{\mu_w^2}{2} \right) - S_\beta \frac{16\lambda_\beta^2}{\gamma} \right) - \frac{4}{3} \eta_\beta \mu_w^2$$

$$b_{32}^* = -\left(\frac{4}{3} \right)^2 \eta_\beta \mu_w$$

$$b_{33}^* = -\frac{8\lambda_\beta^2}{\gamma}\eta_\beta\mu_w$$

Lastly, the angular rate matrix coefficient's derivative is:

$$\frac{\partial A_{\beta\omega}}{\partial \mu_w} = \frac{\gamma}{8\lambda_\beta^2} \begin{bmatrix} 0 & 0 & c_{13}^* & 0 \\ c_{21}^* & 0 & c_{23}^* & c_{24}^* \\ 0 & c_{32}^* & c_{33}^* & c_{34}^* \end{bmatrix} \quad (\text{A.80})$$

$$c_{13}^* = \frac{2}{3}$$

$$c_{21}^* = \eta_\beta \left(\frac{8\lambda_\beta}{\gamma} \right)^2 \mu_w$$

$$c_{23}^* = \eta_\beta \left(\frac{8\lambda_\beta^2}{\gamma} + \frac{16}{9}S_\beta \right) \mu_w$$

$$c_{24}^* = -\eta_\beta \frac{128\lambda_\beta^2}{\gamma^2} \mu_w$$

$$c_{32}^* = \eta_\beta \left(\frac{8\lambda_\beta}{\gamma} \right)^2 \mu_w$$

$$c_{33}^* = \left(\eta_\beta \frac{128\lambda_\beta^2}{\gamma^2} + \frac{16}{9} \right) \mu_w$$

$$c_{34}^* = \eta_\beta \frac{\lambda_\beta^2}{\gamma} \mu_w$$

The normal velocity does not affect μ_w but does effect in the inflow vector, so the derivative of flapping with respect to w_h is found as:

$$\frac{\partial \underline{\beta}_w}{\partial w_h} = A_{\beta\lambda} {}^\lambda T_h^w \begin{bmatrix} \frac{\partial \xi_h}{\partial w_h} \\ 0 \\ 0 \end{bmatrix} \quad (\text{A.81})$$

Transforming to the *hub* axes requires simply using the transformation matrix:

$$\frac{\partial \underline{\beta}_h}{\partial w_h} = {}^\beta T_h^w \frac{\partial \underline{\beta}_w}{\partial w_h} \quad (\text{A.82})$$

A.1.8 Flapping - Rotor Speed

Blade flapping also depends highly on rotor speed as it causes the balance of inertial and aerodynamic forces. The derivative of *wind* axis blade flapping is given as:

$$\frac{\partial \underline{\beta}_w}{\partial \Omega} = \frac{\partial A_{\beta\theta}}{\partial \Omega} T_h^w \underline{\theta} + \frac{\partial A_{\beta\lambda}}{\partial \Omega} T_h^w \underline{\lambda} + A_{\beta\lambda} T_h^w \frac{\partial \underline{\lambda}}{\partial \Omega} + \frac{\partial A_{\beta\omega}}{\partial \Omega} T_h^w \underline{\omega} + A_{\beta\omega} T_h^w \frac{\partial \underline{\omega}}{\partial \Omega} \quad (\text{A.83})$$

Transforming to the *hub* axes:

$$\frac{\partial \underline{\beta}_h}{\partial \Omega} = {}^\beta T_w^h \frac{\partial \underline{\beta}_w}{\partial \Omega} \quad (\text{A.84})$$

Flapping frequency depends on rotor speed as:

$$\lambda_\beta^{2*} = -2 \frac{K_\beta}{I_B \Omega^3} \quad (\text{A.85})$$

where τ^* now denotes the derivative of τ with respect to Ω . The stiffness number derivative is then found as:

$$S_\beta^* = \frac{8}{\gamma} \lambda_\beta^{2*} \quad (\text{A.86})$$

Lastly, the auxiliary term (η_β):

$$\eta_\beta^* = \frac{2S_\beta S_\beta^*}{(1 + S_\beta^2)^2} \quad (\text{A.87})$$

The coefficient of the flapping matrices depend on Ω . Defining the controls flapping matrix as:

$$A_{\beta\theta} = \frac{\gamma}{8\lambda_\beta^2} M_\theta \quad (\text{A.88})$$

allows the derivative to be found:

$$A_{\beta\theta}^* = \frac{\gamma}{8\lambda_\beta^2} \left(M_\theta^* - \frac{M_\theta}{\lambda_\beta^2} \lambda_\beta^{2*} \right) \quad (\text{A.89})$$

The derivative of the auxiliary flapping matrix is then taken as:

$$M_\theta^* = \begin{bmatrix} a_{11}^* & a_{12}^* & a_{13}^* & 0 \\ a_{21}^* & a_{22}^* & a_{23}^* & a_{24}^* \\ a_{31}^* & a_{32}^* & a_{33}^* & a_{34}^* \end{bmatrix} \quad (\text{A.90})$$

$$a_{11}^* = 2\mu_w\mu_w^*$$

$$a_{12}^* = \frac{4}{3}\mu_w\mu_w^*$$

$$a_{13}^* = \frac{4}{3}\mu_w^*$$

$$Z_1 = S_\beta (1 + \mu_w^2) + \frac{16\lambda_\beta^2}{\gamma} \left(1 + \frac{\mu_w^2}{2} \right)$$

$$Z_1^* = S_\beta^* (1 + \mu_w^2) + \frac{16}{\gamma} \lambda_\beta^{2*} \left(1 + \frac{\mu_w^2}{2} \right) + \mu_w\mu_w^* \left(2S_\beta + \frac{16\lambda_\beta^2}{\gamma} \right)$$

$$d_{21}^* = \eta_\beta^* \frac{4}{3} \mu_w Z_1 + \eta_\beta \frac{4}{3} (\mu_w^* Z_1 + \mu_w Z_1^*)$$

$$Z_2 = \frac{8\lambda_\beta^2}{\gamma} \left(1 + \frac{\mu_w^2}{2} \right) + \frac{8}{15} S_\beta \left(1 + \frac{5}{2} \mu_w^2 \right)$$

$$Z_2^* = \frac{8}{\gamma} \lambda_\beta^{2*} \left(1 + \frac{\mu_w^2}{2} \right) + \frac{8}{15} S_\beta^* \left(1 + \frac{5}{2} \mu_w^2 \right) + \mu_w\mu_w^* \left(\frac{8\lambda_\beta^2}{\gamma} + \frac{8}{3} S_\beta \right)$$

$$a_{22}^* = 2\eta_\beta^* \mu_w Z_2 + 2\eta_\beta (\mu_w^* Z_2 + \mu_w Z_2^*)$$

$$Z_3 = \frac{8\lambda_\beta^2}{\gamma} (1 + 2\mu_w^2) + \left(\frac{4}{3}\mu_w\right)^2 S_\beta$$

$$Z_3^* = \frac{8}{\gamma} \lambda_\beta^{2*} (1 + 2\mu_w^2) + \mu_w \mu_w^* \left(\frac{32\lambda_\beta^2}{\gamma} + \frac{32}{9} S_\beta \right) + \left(\frac{4}{3}\mu_w\right)^2 S_\beta^*$$

$$a_{23}^* = \eta_\beta^* Z_3 + \eta_\beta Z_3^*$$

$$Z_4 = \frac{8\lambda_\beta^2}{\gamma} \left(1 + \frac{\mu_w^2}{2}\right)$$

$$Z_4^* = \frac{8}{\gamma} \lambda_\beta^{2*} \left(1 + \frac{\mu_w^2}{2}\right) + \frac{8\lambda_\beta^2}{\gamma} \mu_w \mu_w^*$$

$$a_{24}^* = -\eta_\beta^* S_\beta Z_4 - \eta_\beta (S_\beta^* Z_4 + S_\beta Z_4^*)$$

$$Z_5 = 1 + \frac{\mu_w^2}{2} - 2S_\beta \frac{8\lambda_\beta^2}{\gamma}$$

$$Z_5^* = \mu_w \mu_w^* - 2S_\beta^* \frac{8\lambda_\beta^2}{\gamma} - 2S_\beta \frac{8}{\gamma} \lambda_\beta^{2*}$$

$$a_{31}^* = \eta_\beta^* \frac{4}{3} \mu_w Z_5 + \eta_\beta \frac{4}{3} (\mu_w^* Z_5 + \mu_w Z_5^*)$$

$$Z_6 = \frac{8}{15} \left(1 + \frac{\mu_w^2}{3}\right) - S_\beta \frac{8\lambda_\beta^2}{\gamma}$$

$$Z_6^* = \frac{16}{45} \mu_w \mu_w^* - S_\beta^* \frac{8\lambda_\beta^2}{\gamma} - S_\beta \frac{8}{\gamma} \lambda_\beta^{2*}$$

$$a_{32}^* = 2\eta_\beta^* \mu_w Z_6 + 2\eta_\beta (\mu_w^* Z_6 + \mu_w Z_6^*)$$

$$Z_7 = \frac{8\lambda_\beta^2}{\gamma} \left(1 + \frac{3}{2}\mu_w^2\right)$$

$$Z_7^* = \frac{8}{\gamma} \lambda_\beta^{2*} \left(1 + \frac{3}{2}\mu_w^2\right) + \frac{24\lambda_\beta^2}{\gamma} \mu_w \mu_w^*$$

$$Z_8 = \left(\frac{4}{3}\mu_w\right)^2 - S_\beta Z_7$$

$$Z_8^* = \frac{32}{9} \mu_w \mu_w^* - S_\beta^* Z_7 - S_\beta Z_7^*$$

$$a_{33}^* = \eta_\beta^* Z_8 + \eta_\beta Z_8^*$$

$$\begin{aligned}
Z_9 &= \frac{8\lambda_\beta^2}{\gamma} \left(1 - \frac{\mu_w^4}{2} \right) \\
Z_9^* &= \frac{8}{\gamma} \lambda_\beta^{2*} \left(1 - \frac{\mu_w^4}{2} \right) - \frac{16\lambda_\beta^2}{\gamma} \mu_w^3 \mu_w^* \\
a_{34}^* &= -\eta_\beta Z_9 - \eta_\beta Z_9^*
\end{aligned}$$

The inflow flapping matrix is defined as:

$$A_{\beta\lambda} = \frac{\gamma}{8\lambda_\beta^2} M_\lambda \quad (\text{A.91})$$

leading to the similar derivative:

$$\frac{\partial A_{\beta\lambda}}{\partial \Omega} = \frac{\gamma}{8\lambda_\beta^2} \left(M_\lambda^* - \frac{M_\lambda}{\lambda_\beta^2} \lambda_\beta^{2*} \right) \quad (\text{A.92})$$

The derivative of the auxiliary inflow matrix is:

$$M_\lambda = \begin{bmatrix} 0 & b_{12}^* & 0 \\ b_{21}^* & b_{22}^* & b_{23}^* \\ b_{31}^* & b_{32}^* & b_{33}^* \end{bmatrix} \quad (\text{A.93})$$

$$\begin{aligned}
b_{12}^* &= -\frac{2}{3} \mu_w^* \\
Z_{10} &= 16 \left(\frac{S_\beta}{9} + \frac{\lambda_\beta^2}{\gamma} \left(1 + \frac{\mu_w^2}{2} \right) \right) \\
Z_{10}^* &= 16 \left(\frac{S_\beta^*}{9} + \frac{\lambda_\beta^{2*}}{\gamma} \left(1 + \frac{\mu_w^2}{2} \right) + \frac{\lambda_\beta^2}{\gamma} \mu_w \mu_w^* \right) \\
b_{21}^* &= \eta_\beta^* \mu_w Z_{10} + \eta_\beta (\mu_w^* Z_{10} + \mu_w Z_{10}^*) \\
Z_{11} &= Z_4 + \frac{S_\beta}{2} \left(\frac{4}{3} \mu_w \right)^2 \\
Z_{11}^* &= Z_4^* + 16 \mu_w \left(\frac{S_\beta^*}{18} \mu_w + \frac{S_\beta}{9} \mu_w^* \right)
\end{aligned}$$

$$\begin{aligned}
b_{22}^* &= -\eta_\beta^* Z_{11} - \eta_\beta Z_{11}^* \\
b_{23}^* &= \eta_\beta^* \frac{8\lambda_\beta^2}{\gamma} S_\beta + \frac{8}{\gamma} \eta_\beta (\lambda_\beta^{2*} S_\beta + \lambda_\beta^2 S_\beta^*) \\
Z_{12} &= \left(\frac{4}{3}\right)^2 \left(1 - \frac{\mu_w^2}{2}\right) - S_\beta \frac{16\lambda_\beta^2}{\gamma} \\
Z_{12}^* &= -16 \left(\frac{\mu_w}{9} \mu_w^* + S_\beta^* \frac{\lambda_\beta^2}{\gamma} + S_\beta \frac{\lambda_\beta^{2*}}{\gamma} \right) \\
b_{31}^* &= \eta_\beta^* \mu_w Z_{12} + \eta_\beta (\mu_w^* Z_{12} + \mu_w Z_{12}^*) \\
Z_{13} &= \frac{8\lambda_\beta^2}{\gamma} S_\beta - \frac{1}{2} \left(\frac{4}{3} \mu_w\right)^2 \\
Z_{13}^* &= \frac{8}{\gamma} \lambda_\beta^* S_\beta + \frac{8\lambda_\beta^2}{\gamma} S_\beta^* - \frac{16}{9} \mu_w \mu_w^* \\
b_{32}^* &= \eta_\beta^* Z_{13} + \eta_\beta Z_{13}^* \\
Z_{14} &= \frac{8\lambda_\beta^2}{\gamma} \left(1 - \frac{\mu_w^2}{2}\right) \\
Z_{14}^* &= \frac{8}{\gamma} \lambda_\beta^{2*} \left(1 - \frac{\mu_w^2}{2}\right) - \frac{8\lambda_\beta^2}{\gamma} \mu_w \mu_w^* \\
b_{33}^* &= \eta_\beta^* Z_{14} + \eta_\beta Z_{14}^*
\end{aligned}$$

The angular rate flapping matrix is defined as:

$$A_{\beta\omega} = \frac{\gamma}{8\lambda_\beta^2} M_\omega \quad (\text{A.94})$$

giving:

$$A_{\beta\omega}^* = \frac{\gamma}{8\lambda_\beta^2} \left(M_\omega^* - \frac{M_\omega}{\lambda_\beta^2} \lambda_\beta^{2*} \right) \quad (\text{A.95})$$

The derivative of the auxiliary angular rate matrix is:

$$M_\omega = \begin{bmatrix} 0 & 0 & c_{13}^* & 0 \\ c_{21}^* & c_{22}^* & c_{23}^* & c_{24}^* \\ c_{31}^* & c_{32}^* & c_{33}^* & c_{34}^* \end{bmatrix} \quad (\text{A.96})$$

$$c_{13}^* = \frac{2}{3}\mu_w^*$$

$$c_{21}^* = \eta_\beta^* Z_4 + \eta_\beta Z_4^*$$

$$c_{22}^* = -\eta_\beta \left(\frac{8\lambda_\beta}{\gamma} \right)^2 S_\beta - \eta_\beta \left(\left(\frac{8}{\gamma} \right)^2 \lambda_\beta^{2*} S_\beta + \left(\frac{8\lambda_\beta}{\gamma} \right)^2 S_\beta^* \right)$$

$$Z_{15} = \frac{8\lambda_\beta^2}{\gamma} \left(1 + \frac{\mu_w^2}{2} - \frac{16S_\beta}{\gamma} \right) + \frac{S_\beta}{2} \left(\frac{4}{3}\mu_w \right)^2$$

$$Z_{15}^* = \frac{8}{\gamma} \lambda_\beta^{2*} \left(1 + \frac{\mu_w^2}{2} - \frac{16S_\beta}{\gamma} \right) + \frac{8\lambda_\beta^2}{\gamma} \left(\mu_w \mu_w^* - \frac{16}{\gamma} S_\beta^* \right) + \frac{1}{2} S_\beta^* \left(\frac{4}{3}\mu_w \right)^2 + \frac{16}{9} S_\beta \mu_w \mu_w^*$$

$$c_{23}^* = \eta_\beta^* Z_{15} + \eta_\beta Z_{15}^*$$

$$Z_{16} = S_\beta + \frac{16}{\gamma} \left(1 + \frac{\mu_w^2}{2} \right)$$

$$Z_{16}^* = S_\beta^* + \frac{16}{\gamma} \mu_w \mu_w^*$$

$$c_{24}^* = -\eta_\beta^* \frac{8\lambda_\beta^2}{\gamma} Z_{16} - \eta_\beta \left(\frac{8}{\gamma} \lambda_\beta^{2*} Z_{16} + \frac{8\lambda_\beta^2}{\gamma} Z_{16}^* \right)$$

$$c_{31}^* = -\eta_\beta^* S_\beta \left(\frac{8\lambda_\beta}{\gamma} \right)^2 - \eta_\beta \left(S_\beta^* \left(\frac{8\lambda_\beta}{\gamma} \right)^2 + S_\beta \left(\frac{8}{\gamma} \right)^2 \lambda_\beta^{2*} \right)$$

$$c_{32}^* = \eta_\beta^* \left(\frac{8\lambda_\beta}{\gamma} \right)^2 \left(\frac{\mu_w^2}{2} - 1 \right) + \eta_\beta \left(\left(\frac{8}{\gamma} \right)^2 \lambda_\beta^{2*} \left(\frac{\mu_w^2}{2} - 1 \right) + \left(\frac{8\lambda_\beta}{\gamma} \right)^2 \mu_w \mu_w^* \right)$$

$$Z_{17} = \frac{8\lambda_\beta^2}{\gamma} \left(\frac{16}{\gamma} \left(\frac{\mu_w^2}{2} - 1 \right) - S_\beta \right)$$

$$Z_{17}^* = \frac{8}{\gamma} \lambda_\beta^{2*} \left(\frac{16}{\gamma} \left(\frac{\mu_w^2}{2} - 1 \right) - S_\beta \right) + \frac{8\lambda_\beta^2}{\gamma} \left(\frac{16}{\gamma} \mu_w \mu_w^* - S_\beta^* \right)$$

$$\begin{aligned}
c_{33}^* &= \eta_\beta^* Z_{17} + \frac{16}{9} \mu_w \mu_w^* + \eta_\beta Z_{17}^* \\
Z_{18} &= \frac{8\lambda_\beta^2}{\gamma} \left(\frac{16S_\beta}{\gamma} + \frac{\mu_w^2}{2} - 1 \right) \\
Z_{18}^* &= \frac{8}{\gamma} \lambda_\beta^{2*} \left(\frac{16S_\beta}{\gamma} + \frac{\mu_w^2}{2} - 1 \right) + \frac{8\lambda_\beta^2}{\gamma} \left(\frac{16}{\gamma} S_\beta^* + \mu_w \mu_w^* \right) \\
c_{34}^* &= \eta_\beta^* Z_{18} + \eta_\beta Z_{18}^*
\end{aligned}$$

A.1.9 Hub Forces and Moments

Rotor forces and moments depend on both the amount of thrust being produced by the rotor and the flapping states of the rotor. Defining the rotor force coefficient as:

$$\underline{C}_{F_h} = \begin{bmatrix} C_{xh} \\ C_{yh} \\ C_{zh} \end{bmatrix} \quad (\text{A.97})$$

then leads to introduction of thrust and flapping:

$$\underline{C}_{F_h} = \begin{bmatrix} C_T \beta_c \\ -C_T \beta_s \\ -C_T \end{bmatrix} \quad (\text{A.98})$$

Taking the derivative with respect to C_T :

$$\frac{\partial \underline{C}_{F_h}}{\partial C_T} = \begin{bmatrix} \beta_c \\ -\beta_s \\ -1 \end{bmatrix} \quad (\text{A.99})$$

then the derivative with respect to $\underline{\beta}_h$

$$\frac{\partial \underline{C}_{F_h}}{\partial \underline{\beta}} = \begin{bmatrix} 0 & C_T & 0 \\ 0 & 0 & -C_T \\ 0 & 0 & 0 \end{bmatrix} \quad (\text{A.100})$$

Using the familiar process involving the product rule and chain rule, the force derivatives with respect to the vehicle states can be assembled. The rotor moment derivative solely depends on blade flapping. Defining a moment vector as:

$$\underline{M}_h = \begin{bmatrix} M_{x_h} \\ M_{y_h} \end{bmatrix} \quad (\text{A.101})$$

yields a derivative with respect to blade flapping of:

$$\frac{\partial \underline{M}_h}{\partial \underline{\beta}_h} = -\frac{N_b}{2} K_\beta \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad (\text{A.102})$$

A.1.10 Dimensional Derivatives - Rotor Speed

The nondimensionalizing terms are functions of rotor speed and must also be taken into account. A generic rotor force K can be written as:

$$K = \rho A V_T^2 R C_K \quad (\text{A.103})$$

leading to the derivative:

$$\frac{\partial K}{\partial \Omega} = \rho A V_T \left(V_T \frac{\partial C_K}{\partial \Omega} + 2 R C_K \right) \quad (\text{A.104})$$

Secondly, a generic rotor moment J written as:

$$J = \rho A V_T^2 R C_J \quad (\text{A.105})$$

can be differentiated as:

$$\frac{\partial J}{\partial \Omega} = \rho A R V_T \left(V_T \frac{\partial C_J}{\partial \Omega} + 2 R C_J \right) \quad (\text{A.106})$$

A.1.11 Controls

Control inputs are mapped to the individual rotors to create desired forces and moments in an efficient way. The right rotor controls depend on the inputs as:

$$\frac{\partial \underline{\theta}_r}{\partial \underline{u}} = \frac{1}{2} \begin{bmatrix} 1 & 0 & -D_{\theta_0} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & -(1 - D_{\theta_0}) & 0 \end{bmatrix} \quad (\text{A.107})$$

The left rotor's control derivative is given by:

$$\frac{\partial \underline{\theta}_l}{\partial \underline{u}} = \frac{1}{2} \begin{bmatrix} 1 & 0 & D_{\theta_0} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & -(1 - D_{\theta_0}) & 0 \end{bmatrix} \quad (\text{A.108})$$

A.2 Positive Semidefiniteness of $A^T Q A$

Let $Q \in \mathbb{R}^{n \times n}$ be a positive semidefinite (PSD) symmetric matrix. There exists an eigenvalue decomposition of Q in the form of:

$$Q = P \Lambda P^T \quad (\text{A.109})$$

where $P \in \mathbb{R}^{n \times n}$ is composed of orthogonal eigenvectors and $\Lambda \in \mathbb{R}^{n \times n}$ is a diagonal matrix of eigenvalues. Since Q is PSD, its eigenvalues are non-negative and the square root of Λ can be taken:

$$Q = \left(P \Lambda^{\frac{1}{2}} \right) \left(\Lambda^{\frac{1}{2}} P^T \right) \quad (\text{A.110})$$

Let $y \in \mathbb{R}$ be a function of $\underline{x} \in \mathbb{R}^n$ and be defined as:

$$y = \underline{x}^T A^T Q A \underline{x} \quad (\text{A.111})$$

where $A \in \mathbb{R}^{n \times n}$ is of generic value. Equation (A.111) can be rewritten as:

$$y = \underline{x}^T A^T \left(P \Lambda^{\frac{1}{2}} \right) \left(\Lambda^{\frac{1}{2}} P^T \right) A \underline{x} \quad (\text{A.112})$$

Let $\underline{z} \in \mathbb{R}^n$ be defined as:

$$\underline{z} = \left(\Lambda^{\frac{1}{2}} P^T \right) A \underline{x} \quad (\text{A.113})$$

If \underline{x} is chosen such that it is a singular value of A or any eigenvalues of Q are zero then \underline{z} may equal zero but it will remain non-negative. Equation (A.112) can be rewritten as:

$$y = \underline{z}^T \underline{z} \quad (\text{A.114})$$

showing that:

$$y \geq 0 \quad \forall \quad \underline{z} \in \mathbb{R}^n \quad (\text{A.115})$$

and proving that:

$$A^T Q A \geq 0 \quad (\text{A.116})$$