

NONLINEAR FLIGHT CONTROL USING
ADAPTIVE CRITIC BASED NEURAL NETWORKS

by

SERGIO ESTEBAN RONCERO

A THESIS

Presented to the Faculty of the Graduate School of the

UNIVERSITY OF MISSOURI-ROLLA

In Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCES

in

AEROSPACE ENGINEERING

2002

Dr. S.N. Balakrishnan, Advisor

Dr. W. Eversman

Dr. L. Acar

Dr. C. A. Kluever

ABSTRACT

In this thesis an adaptive critic based neural network is developed to obtain near-optimal control laws for a robust nonlinear flight control system. The adaptive critic approach consists of two neural networks. The first network, called the critic, captures the mapping between the states of a dynamical system and the co-states that arise in an optimal control problem. The second network, called the action network, maps the states of a system to the control. Ultimately the purpose of the nonlinear flight control system developed in this work is to pave the way for an adaptive reconfigurable nonlinear controller that would make aviation a safe way of transportation even in the presence of control failures and/or damaged aerodynamic surfaces.

In order to show the effectiveness of the nonlinear adaptive critic neurocontroller used in this thesis, the author first shows the nonlinear properties using a nonlinear aircraft model to increase its stall envelope, then he proceeds to conduct a sensitivity study of the neurocontroller to unmodeled uncertainties to show the inherent robustness properties of the neurocontroller. Finally the author implements the neurocontroller design in a nonlinear tracker control system for a simplified version of the previous model.

ACKNOWLEDGMENTS

This study was supported in part by a grant from National Science Foundation ECS-9976588 and the NASA-Ames Research Center. Without the help of the sponsors this research would have not been able to be conducted and because of this I gratefully thank for the opportunity given by them.

I would like to thanks Dr. S. N. Balakrishnan for his guidance and support through this research and most important for being above everything a great friend and keeping encouraging me through tough times. I would also like to thank Dr. Levent Acar, Dr. Walter Eversman, and Dr. Craig A. Kluever for serving on my graduate committee. I would also like to thank George P. Wright Jr. for his revisions.

I would like to express special thanks and gratitude to my parents Jerónimo Esteban Delgado and María del Sagrario Roncero Domínguez, and my sisters Ruth & Judith Esteban Roncero, and all the friends and family for their continuous support and encouragement in my pursuit of my goals through the years and the distance.

TABLE OF CONTENTS

	Page
ABSTRACT.....	iii
ACKNOWLEDGMENTS	iv
LIST OF ILLUSTRATIONS.....	viii
LIST OF TABLES.....	xvi
SECTION	
1. INTRODUCTION.....	1
1.1. MOTIVATION – RECONFIGURABLE FLIGHT CONTROL.....	1
1.2. NEUROBIOLOGY.....	3
1.2.1. Origins of Neurobiology	4
1.2.2. Modern Neurobiology	6
1.3. ARTIFICIAL NEURAL NETWORKS.....	10
1.3.1. History of Artificial Neural Networks.....	11
1.3.2. What is an Artificial Neural Network?.....	15
1.3.3. Classification of Neural Networks	17
1.3.4. What Can and Cannot be Done with Artificial Neural Networks?	18
1.3.5. Other Approaches?.....	19
2. OPTIMAL CONTROL	23
2.1. OPTIMAL CONTROL THEORY.....	23
2.1.1. Definition of Optimal Control	23
2.1.2. Pontryagin Minimum Principle: Hamiltonian Formulation.....	24
2.1.3. Dynamic Programming	25
2.1.4. The Optimal Tracking Problem	27
2.1.5. A Special Case - Linear Quadratic Regulator.....	28
2.2. NONLINEAR OPTIMAL CONTROL BACKGROUND	30
2.2.1. Classical Control	30
2.2.2. Nonlinear Optimal Control Methods.....	32
2.2.3. Nonlinear Control and Neural Networks.....	33
3. ADAPTIVE CRITIC NEURAL NETWORK DESIGN.....	42

3.1. NEUROCONTROLLER DEVELOPMENT.....	42
3.2. TRAINING SYNTHESIS OF THE NEUROCONTROLLER.....	47
3.2.1. Training Synthesis of the Action Neurocontroller	47
3.2.2. Training Synthesis of the Critic Neural Network.....	48
4. PROBLEM DEFINITION	51
4.1. AIRCRAFT MODEL.....	51
4.1.1. Reference Coordinate System	51
4.1.2. Non-linear Six-Degrees-Of-Freedom Model	52
4.1.3. Airplane Controls	58
4.1.4. Perturbed State Equations of Motion Model.....	59
4.1.5. Longitudinal Linearized Model.....	64
4.1.6. Phugoid and Short Period Approximations.....	67
4.2. NONLINEAR HIGH ANGLE OF ATTACK PROBLEM	69
4.2.1. Problem Formulation.....	69
4.2.2. Optimal Control Formulation.....	75
4.2.3. Approach to Solving the Nonlinear High Angle of Attack Problem	82
4.2.4. Testing the Bench Mark Controller.....	82
4.2.5. Neural Network Analysis	103
4.2.5.1 Neural network architecture.....	103
4.2.5.2 Neural network initial training approach	106
4.2.5.3 Trained neural networks limits	112
4.2.6. Analysis of Numerical Results	118
4.2.7. Sensitivity Study of the Neurocontroller to Unmodeled Uncertainties	138
4.2.8. Analysis of Results to Unmodeled Uncertainties.....	139
4.2.8.1 Analysis of the sensitivity study to varying time lag.....	140
4.2.8.2 Analysis of the sensitivity study to varying elevator effectiveness.....	152
4.3. NON-LINEAR TRACKING PROBLEM.....	161
4.3.1. Problem Formulation.....	161
4.3.2. Optimal Tracking Control Formulation	162
4.3.3. Calculation of the Steady State Values	165

4.3.4. Neural Network Architecture	177
4.3.5. Neural Network Training Approach.....	178
4.3.6. Training Synthesis of the Neurocontroller	180
4.3.6.1 Training synthesis of the action neurocontroller.....	180
4.3.6.2 Training synthesis of the critic neurocontroller.	181
4.3.7. Analysis of Numerical Results	182
5. CONCLUSIONS	193
BIBLIOGRAPHY	196
VITA	204

LIST OF ILLUSTRATIONS

Figure	Page
1.1. Typical Interaction Between Neurons	7
1.2. Sodium-Potassium Permeability	8
1.3. Detail of the Membrane Potential Reversal.....	9
1.4. Detail of the Neural Synaptic Cleft	9
1.5. An artificial Neuron or Perceptron.....	16
3.1. General Structure of One Layer Neural Network.....	42
3.2. Three Layer Neural Network Architecture.....	43
3.3. Screen Shot for the Command Window of the MATLAB [®] Release 12.....	44
3.4. Training Synthesis of the Action Neural Network Controller	48
3.5. Training Synthesis of the Critic Neural Network Controller	49
4.1. Definition of the Earth-Fixed and Body-Fixed Coordinate Systems.	52
4.2. Rotation Over a Heading Angle of Ψ About Z_1 ^[79]	54
4.3. Rotation Over a Pitch Angle of Θ About Y_2 ^[79]	55
4.4. Rotation Over a Bank Angle of Φ About X_3 ^[79]	55
4.5. Definition of the Stability Axis System for Zero Sideslip ^[79]	57
4.6. Definition of the Stability Axis System for the Case of Nonzero Sideslip Angle.	58
4.7. General Definition of the Airplane Control Surfaces ^[79]	59
4.8. F-8 Crusader in Flight ^[79]	70
4.9. F-8 Aircraft Dynamic Coordinate System ^[79]	70
4.10. States Results ($\alpha_{(0)}=23^\circ$, $\theta_{(0)}=0^\circ$, $q_{(0)}=0^\circ/\text{sec}$).....	83

4.11.	Altitude, Control and Cost Results ($\alpha_{(0)} = 23^\circ$, $\theta_{(0)} = 0^\circ$, $q_{(0)} = 0^\circ/\text{sec}$).....	84
4.12.	States Results ($\alpha_{(0)} = 23.5^\circ$, $\theta_{(0)} = 0^\circ$, $q_{(0)} = 0^\circ/\text{sec}$).....	86
4.13.	Altitude, Control and Cost Results ($\alpha_{(0)} = 23.5^\circ$, $\theta_{(0)} = 0^\circ$, $q_{(0)} = 0^\circ/\text{sec}$).....	87
4.14.	States Results ($\alpha_{(0)} = 25.73^\circ$, $\theta_{(0)} = 0^\circ$, $q_{(0)} = 0^\circ/\text{sec}$).....	87
4.15.	Altitude, Control and Cost Results ($\alpha_{(0)} = 25.73^\circ$, $\theta_{(0)} = 0^\circ$, $q_{(0)} = 0^\circ/\text{sec}$).....	88
4.16.	States Results ($\alpha_{(0)} = 25.99^\circ$, $\theta_{(0)} = 0^\circ$, $q_{(0)} = 0^\circ/\text{sec}$).....	89
4.17.	Altitude, Control and Cost Results ($\alpha_{(0)} = 25.99^\circ$, $\theta_{(0)} = 0^\circ$, $q_{(0)} = 0^\circ/\text{sec}$).....	89
4.18.	States Results ($\alpha_{(0)} = 27.09^\circ$, $\theta_{(0)} = 0^\circ$, $q_{(0)} = 0^\circ/\text{sec}$).....	90
4.19.	Altitude, Control and Cost Results ($\alpha_{(0)} = 27.09^\circ$, $\theta_{(0)} = 0^\circ$, $q_{(0)} = 0^\circ/\text{sec}$).....	90
4.20.	Maximum Allowable Initial Angle of Attack for Different Initial Pitch Angles and Pitch Rates for the LQR Controller.....	93
4.21.	Maximum Allowable Initial Angle of Attack for Different Initial Pitch Angles and Pitch Rates for the 2 nd -Order Controller.....	93
4.22.	Maximum Allowable Initial Angle of Attack for Different Initial Pitch Angles and Pitch Rates for the 3 rd -Order Controller.....	94
4.23.	Maximum Allowable Initial Angle of Attack for Different Initial Pitch Angles and Pitch Rates Comparison of the 3 Controller.....	94
4.24.	Angle of Attack Time History for the 3 rd -Order Controller ($\theta_{(0)} = \pm 20^\circ$ and $q_{(0)} = 20^\circ/\text{sec}$).....	95
4.25.	Pitch Angle Time History for the 3 rd -Order Controller ($\theta_{(0)} = \pm 20^\circ$ and $q_{(0)} = -20^\circ/\text{sec}$).....	96
4.26.	Pitch Rate Time History for the 3 rd -Order Controller ($\theta_{(0)} = \pm 20^\circ$ and $q_{(0)} = -20^\circ/\text{sec}$).....	96
4.27.	Flight Path Angle Time History for the 3 rd -Order Controller ($\theta_{(0)} = \pm 20^\circ$ and $q_{(0)} = -20^\circ/\text{sec}$).....	97
4.28.	Drop in Altitude Time History for the 3 rd -Order Controller ($\theta_{(0)} = \pm 20^\circ$ and $q_{(0)} = -20^\circ/\text{sec}$).....	97

4.29.	Angle of Attack Time History for the 3 rd -Order Controller ($\theta_{(0)}=\pm 20^\circ$ and $q_{(0)}=0$ °/sec)	98
4.30.	Pitch Angle Time History for the 3 rd -Order Controller ($\theta_{(0)}=\pm 20^\circ$ and $q_{(0)}=0$ °/sec)	99
4.31.	Pitch Rate Time History for the 3 rd -Order Controller ($\theta_{(0)}=\pm 20^\circ$ and $q_{(0)}=0$ °/sec)	99
4.32.	Flight Path Angle Time History for the 3 rd -Order Controller ($\theta_{(0)}=\pm 20^\circ$ and $q_{(0)}=0$ °/sec)	100
4.33.	Drop in Altitude Time History for the 3 rd -Order Controller ($\theta_{(0)}=\pm 20^\circ$ and $q_{(0)}=0$ °/sec)	100
4.34.	Angle of Attack Time History for the 3 rd -Order Controller ($\theta_{(0)}=\pm 20^\circ$ and $q_{(0)}=20$ °/sec)	101
4.35.	Pitch Angle Time History for the 3 rd -Order Controller ($\theta_{(0)}=\pm 20^\circ$ and $q_{(0)}=20$ °/sec)	101
4.36.	Pitch Rate Time History for the 3 rd -Order Controller ($\theta_{(0)}=\pm 20^\circ$ and $q_{(0)}=20$ °/sec)	102
4.37.	Flight Path Angle Time History for the 3 rd -Order Controller ($\theta_{(0)}=\pm 20^\circ$ and $q_{(0)}=20$ °/sec)	102
4.38.	Drop in Altitude Time History for the 3 rd -Order Controller ($\theta_{(0)}=\pm 20^\circ$ and $q_{(0)}=20$ °/sec)	103
4.39.	Expansion of the Action NN.	105
4.40.	Expanded View of the 1 st Hidden Layer for the Action NN.	106
4.41.	Initialization of the Action Neural Network.	107
4.42.	Initialization of the Critic Neural Network.	108
4.43.	Demonstration of Pitch Angle Steady State Error.	109
4.44.	Fixed Pitch Angle Steady State Error.	110
4.45.	Action NN with Modified Training Target.	112
4.46.	Feedback Neurocontroller Implementation.	112

4.47.	Comparison of the Maximum Allowable Initial Angle of Attack for the 3 rd -Order and the Dual-NN Controller.....	113
4.48.	Angle of Attack Time History for the Dual-NN Controller ($\theta_{(0)}=\pm 20^\circ$ and $q_{(0)}=0^\circ/\text{sec}$).....	116
4.49.	Pitch Angle Time History for the Dual-NN Controller ($\theta_{(0)}=\pm 20^\circ$ and $q_{(0)}=0^\circ/\text{sec}$).....	116
4.50.	Pitch Rate Time History for the Dual-NN Controller ($\theta_{(0)}=\pm 20^\circ$ and $q_{(0)}=0^\circ/\text{sec}$).....	117
4.51.	Flight Path Angle Time History for the Dual-NN Controller ($\theta_{(0)}=\pm 20^\circ$ and $q_{(0)}=0^\circ/\text{sec}$).....	117
4.52.	Drop in Altitude Time History for the Dual-NN Controller ($\theta_{(0)}=\pm 20^\circ$ and $q_{(0)}=0^\circ/\text{sec}$).....	118
4.53.	States Results ($\alpha_{(0)}=25.73^\circ$, $\theta_{(0)}=0^\circ$, $q_{(0)}=0^\circ/\text{sec}$).....	121
4.54.	Altitude, Control and Cost Results ($\alpha_{(0)}=25.73^\circ$, $\theta_{(0)}=0^\circ$, $q_{(0)}=0^\circ/\text{sec}$).....	121
4.55.	States Results ($\alpha_{(0)}=25.99^\circ$, $\theta_{(0)}=0^\circ$, $q_{(0)}=0^\circ/\text{sec}$).....	122
4.56.	Altitude, Control and Cost Results ($\alpha_{(0)}=25.99^\circ$, $\theta_{(0)}=0^\circ$, $q_{(0)}=0^\circ/\text{sec}$).....	122
4.57.	States Results ($\alpha_{(0)}=27.09^\circ$, $\theta_{(0)}=0^\circ$, $q_{(0)}=0^\circ/\text{sec}$).....	123
4.58.	Altitude, Control and Cost Results ($\alpha_{(0)}=27.09^\circ$, $\theta_{(0)}=0^\circ$, $q_{(0)}=0^\circ/\text{sec}$).....	123
4.59.	States Results ($\alpha_{(0)}=30^\circ$, $\theta_{(0)}=0^\circ$, $q_{(0)}=0^\circ/\text{sec}$).....	125
4.60.	Altitude, Control and Cost Results ($\alpha_{(0)}=30^\circ$, $\theta_{(0)}=0^\circ$ and $q_{(0)}=0^\circ/\text{sec}$).....	125
4.61.	States Results ($\alpha_{(0)}=30^\circ$, $\theta_{(0)}=20^\circ$, $q_{(0)}=5^\circ/\text{sec}$).....	127
4.62.	Altitude, Control and Cost Results ($\alpha_{(0)}=30^\circ$, $\theta_{(0)}=20^\circ$, $q_{(0)}=5^\circ/\text{sec}$).....	127
4.63.	States Results ($\theta_{(0)}=\pm 20^\circ$, $\alpha_{(0)}=30^\circ$, $q_{(0)}=5^\circ/\text{sec}$).....	130
4.64.	Altitude, Control and Cost Results ($\theta_{(0)}=\pm 20^\circ$, $\alpha_{(0)}=30^\circ$, $q_{(0)}=5^\circ/\text{sec}$).....	130
4.65.	States Results ($q_{(0)}=\pm 20^\circ$, $\alpha_{(0)}=30^\circ$, $\theta_{(0)}=20^\circ/\text{sec}$).....	131
4.66.	Altitude, Control and Cost Results ($q_{(0)}=\pm 20^\circ$, $\alpha_{(0)}=30^\circ$, $\theta_{(0)}=5^\circ/\text{sec}$).....	131
4.67.	States Results ($\alpha_{(0)}=35^\circ$, $\theta_{(0)}=20^\circ$, $q_{(0)}=5^\circ/\text{sec}$).....	132

4.68.	Altitude, Control and Cost Results ($\alpha_{(0)}=35^\circ$, $\theta_{(0)}=20^\circ$, $q_{(0)}=5^\circ/\text{sec}$).....	132
4.69.	States Results ($\alpha_{(0)}=22.9-35^\circ$, $\theta_{(0)}=10^\circ$, $q_{(0)}=5^\circ/\text{sec}$).	134
4.70.	Altitude, Control and Cost Results ($\alpha_{(0)}=22.9-35^\circ$, $\theta_{(0)}=10^\circ$, $q_{(0)}=5^\circ/\text{sec}$).	134
4.71.	States Results ($\theta_{(0)}=\pm 20^\circ$, $\alpha_{(0)}=35^\circ$, $q_{(0)}=5^\circ/\text{sec}$).	136
4.72.	Altitude, Control and Cost Results ($\theta_{(0)}=\pm 20^\circ$, $\alpha_{(0)}=35^\circ$, $q_{(0)}=5^\circ/\text{sec}$).	136
4.73.	States Results ($\theta_{(0)}=\pm 20^\circ$, $\alpha_{(0)}=36.5^\circ$, $q_{(0)}=5^\circ/\text{sec}$).	137
4.74.	Altitude, Control and Cost Results ($\theta_{(0)}=\pm 20^\circ$, $\alpha_{(0)}=36.5^\circ$, $q_{(0)}=5^\circ/\text{sec}$).	137
4.75.	States Results ($\tau=0.01$, $\alpha(0)=30^\circ$, $\theta(0)=20^\circ$, $q(0)=5^\circ/\text{sec}$).	140
4.76.	Altitude, Control and Cost Results ($\tau=0.01$, $\alpha(0)=30^\circ$, $\theta(0)=20^\circ$, $q(0)=5^\circ/\text{sec}$)...	141
4.77.	States Results ($\tau=0.05$, $\alpha(0)=30^\circ$, $\theta(0)=20^\circ$, $q(0)=5^\circ/\text{sec}$).	142
4.78.	Altitude, Control and Cost Results ($\tau=0.05$, $\alpha(0)=30^\circ$, $\theta(0)=20^\circ$, $q(0)=5^\circ/\text{sec}$)...	143
4.79.	States Results ($\tau=0.1$, $\alpha(0)=30^\circ$, $\theta(0)=20^\circ$, $q(0)=5^\circ/\text{sec}$).	143
4.80.	Altitude, Control and Cost Results ($\tau=0.1$, $\alpha(0)=30^\circ$, $\theta(0)=20^\circ$, $q(0)=5^\circ/\text{sec}$)....	144
4.81.	States Results ($\tau=0.2$, $\alpha(0)=30^\circ$, $\theta(0)=20^\circ$, $q(0)=5^\circ/\text{sec}$).	144
4.82.	Altitude, Control and Cost Results ($\tau=0.2$, $\alpha(0)=30^\circ$, $\theta(0)=20^\circ$, $q(0)=5^\circ/\text{sec}$)....	145
4.83.	States Results ($\tau=0.4$, $\alpha(0)=30^\circ$, $\theta(0)=20^\circ$, $q(0)=5^\circ/\text{sec}$).	147
4.84.	Altitude, Control and Cost Results ($\tau=0.4$, $\alpha(0)=30^\circ$, $\theta(0)=20^\circ$, $q(0)=5^\circ/\text{sec}$)....	147
4.85.	States Results ($\tau=0.1$, $\alpha(0)=35^\circ$, $\theta(0)=20^\circ$, $q(0)=5^\circ/\text{sec}$).	148
4.86.	Altitude, Control and Cost Results ($\tau=0.1$, $\alpha(0)=35^\circ$, $\theta(0)=20^\circ$, $q(0)=5^\circ/\text{sec}$)....	148
4.87.	States Results ($\tau=0.2$, $\alpha(0)=35^\circ$, $\theta(0)=20^\circ$, $q(0)=5^\circ/\text{sec}$).	149
4.88.	Altitude, Control and Cost Results ($\tau=0.2$, $\alpha(0)=35^\circ$, $\theta(0)=20^\circ$, $q(0)=5^\circ/\text{sec}$)....	149
4.89.	Maximum Allowable Time-lag Constant ($\alpha(0)=30^\circ$, $\theta(0)=\pm 20^\circ$, $q(0)=\pm 20^\circ/\text{sec}$)	151

4.90.	States Results ($\theta(0)=\pm 20^\circ$, $\alpha(0)=30^\circ$, $q(0)=5^\circ/\text{sec}$).	151
4.91.	Altitude, Control and Cost Results ($\theta(0)=\pm 20^\circ$, $\alpha(0)=30^\circ$, $q(0)=5^\circ/\text{sec}$).	152
4.92.	States Results ($D_{un}=-0.10$, $\alpha(0)=30^\circ$, $\theta(0)=20^\circ$, $q(0)=5^\circ/\text{sec}$).	153
4.93.	Altitude, Control and Cost Results ($D_{un}=-0.10$, $\alpha(0)=30^\circ$, $\theta(0)=20^\circ$, $q(0)=5^\circ/\text{sec}$)	154
4.94.	States Results ($D_{un}=-0.25$, $\alpha(0)=30^\circ$, $\theta(0)=20^\circ$, $q(0)=5^\circ/\text{sec}$).	154
4.95.	Altitude, Control and Cost Results ($D_{un}=-0.25$, $\alpha(0)=30^\circ$, $\theta(0)=20^\circ$, $q(0)=5^\circ/\text{sec}$)	155
4.96.	States Results ($D_{un}=-0.50$, $\alpha(0)=30^\circ$, $\theta(0)=20^\circ$, $q(0)=5^\circ/\text{sec}$).	155
4.97.	Altitude, Control and Cost Results ($D_{un}=-0.50$, $\alpha(0)=30^\circ$, $\theta(0)=20^\circ$, $q(0)=5^\circ/\text{sec}$)	156
4.98.	States Results ($D_{un}=-0.75$, $\alpha(0)=30^\circ$, $\theta(0)=20^\circ$, $q(0)=5^\circ/\text{sec}$).	157
4.99.	Altitude, Control and Cost Results ($D_{un}=-0.75$, $\alpha(0)=30^\circ$, $\theta(0)=20^\circ$, $q(0)=5^\circ/\text{sec}$.)	157
4.100.	States Results ($D_{un}=-0.50$, $\alpha(0)=35^\circ$, $\theta(0)=20^\circ$, $q(0)=5^\circ/\text{sec}$).	158
4.101.	Altitude, Control and Cost Results ($D_{un}=-0.50$, $\alpha(0)=35^\circ$, $\theta(0)=20^\circ$, $q(0)=5^\circ/\text{sec}$)	158
4.102.	States Results ($D_{un}=0.25$, $\alpha(0)=30^\circ$, $\theta(0)=20^\circ$, $q(0)=5^\circ/\text{sec}$).	159
4.103.	Altitude, Control and Cost Results ($D_{un}=0.25$, $\alpha(0)=30^\circ$, $\theta(0)=20^\circ$, $q(0)=5^\circ/\text{sec}$)	160
4.104.	States Results ($D_{un}=0.50$, $\alpha(0)=30^\circ$, $\theta(0)=20^\circ$, $q(0)=5^\circ/\text{sec}$).	160
4.105.	Altitude, Control and Cost Results ($D_{un}=0.50$, $\alpha(0)=30^\circ$, $\theta(0)=20^\circ$, $q(0)=5^\circ/\text{sec}$)	161
4.106.	Real and Imaginary Parts of the Roots for Eq. (208), $q_D=\pm 15^\circ/\text{sec}$.	167
4.107.	Real and Imaginary Parts of the Roots for Eq. (208) $q_D=-11.5$ to $+7.3^\circ/\text{sec}$.	167
4.108.	1 st Root of the solution to Eq. (208).	168

4.109. 2 nd Root of the solution to Eq. (208).....	168
4.110. 3 rd Root of the solution to Eq. (208).....	169
4.111. Satisfaction of the Cubic Polynomial in Eq. (208).....	169
4.112. Satisfaction of the Left-hand Side of Eqs. (205) and (206).....	170
4.113. Model to Check the Correctness of the 3 Roots.....	171
4.114. States and Errors, LQR Simulation for the 1 st Root, $q_D=5$ °/sec	174
4.115. Total Elevator Control and $\bar{\delta}_e$, LQR Simulation for the 1 st Root, $q_D=5$ °/sec	175
4.116. States and Errors, LQR Simulation for the 2 nd Root, $q_D=5$ °/sec.....	175
4.117. Total Elevator Control and $\bar{\delta}_e$, LQR Simulation for the 2 nd Root, $q_D=5$ °/sec.....	176
4.118. States and Errors, LQR Simulation for the 3 rd Root, $q_D=5$ °/sec.....	176
4.119. Total Elevator Control and $\bar{\delta}_e$, LQR Simulation for the 3 rd Root, $q_D=5$ °/sec	177
4.120. Schematic of the Input-Output Relation for the Action Neural Network	178
4.121. Schematic of the Input-Output Relation for the Critic Neural Network	178
4.122. Initialization of the Action Neural Network.....	179
4.123. Initialization of the Critic Neural Network.	180
4.124. Implementation of the Neural Network Tracking Feedback Controller.	182
4.125. States and Errors Simulation ($q_D=-2.5$ °/sec).....	184
4.126. Total Tail Rotation Control and $\bar{\delta}_e$ Simulation ($q_D=-2.5$ °/sec).....	184
4.127. Cost Time History ($q_D=-2.5$ °/sec).....	185
4.128. States and Errors Simulation ($q_D=2.5$ °/sec).....	185
4.129. Total Tail Rotation Control and $\bar{\delta}_e$ Simulation ($q_D=2.5$ °/sec).....	186
4.130. Cost Time History ($q_D=2.5$ °/sec).....	186

4.131. States and Errors Simulation ($q_D=-11$ °/sec).....	187
4.132. Total Tail Rotation Control and $\bar{\delta}_e$ Simulation for ($q_D=-11$ °/sec).	187
4.133. Cost Time History ($q_D=-11$ °/sec).....	188
4.134. States and Errors Simulation ($q_D=7$ °/sec).....	188
4.135. Total Tail Rotation Control and $\bar{\delta}_e$ Simulation ($q_D=7$ °/sec).	189
4.136. Cost Time History ($q_D=7$ °/sec).....	189
4.137. States and Errors Simulation ($q_D=-5^\circ/\text{sec}$, $q_D=5^\circ/\text{sec}$).....	190
4.138. Total Tail Rotation Control and $\bar{\delta}_e$ Simulation ($q_D=-5^\circ/\text{sec}$, $q_D=5^\circ/\text{sec}$).	190
4.139. Cost Time History ($q_D=-5^\circ/\text{sec}$, $q_D=5^\circ/\text{sec}$).....	191
4.140. States and Errors Simulation ($q_D=3^\circ/\text{sec}$, $q_D=-2.5^\circ/\text{sec}$, $q_D=5^\circ/\text{sec}$).	191
4.141. Total Tail Rotation Control and $\bar{\delta}_e$ Simulation ($q_D=3^\circ/\text{sec}$, $q_D=-2.5^\circ/\text{sec}$, $q_D=5^\circ/\text{sec}$)	192
4.142. Cost Time History for Variable ($q_D=3^\circ/\text{sec}$, $q_D=-2.5^\circ/\text{sec}$, $q_D=5^\circ/\text{sec}$).....	192

LIST OF TABLES

Table		Page
4.1.	Associated Symbols for the States, Perturbed States and Steady States	60
4.2.	Maximum Allowable Initial Angle of Attack for Different Initial Pitch Angles and Pitch Rates for the LQR Controller.....	91
4.3.	Maximum Allowable Initial Angle of Attack for Different Initial Pitch Angles and Pitch Rates for the 2 nd Order Controller.....	92
4.4.	Maximum Allowable Initial Angle of Attack for Different Initial Pitch Angles and Pitch Rates for 3 rd Order Controller.	92
4.5.	Maximum Allowable Initial Angle of Attack for Different Initial Pitch Angles and Pitch Rates for the Dual-NN Controller.	114
4.6.	Maximum Allowable Time-lag Constants for Different Initial Pitch Angles and Pitch Rates for the Dual-NN Controller.	150
4.7.	Associated Desired Values for a Comanded Pitch Rate of a $q_D = 5^\circ/\text{sec}$	172

1. INTRODUCTION

1.1. MOTIVATION – RECONFIGURABLE FLIGHT CONTROL

It is a proven fact that aviation transportation is the most secure media of transportation existing today. Statistically speaking, the risk of being involved in a commercial accident, with multiple fatalities, is approximately one in three million [1]. In other words, you would have to fly everyday for more than 8200 years to accumulate three million flights.

Just in the United States (U.S.) alone, the number of U.S. highway deaths in a typical six-month period — about 21,000 — roughly equals to all commercial jet fatalities worldwide since the dawn of jet aviation four decades ago. In fact, fewer people have died in commercial airplane accidents in America over the past 60 years than are killed in U.S. auto accidents in a typical three-month period.

Despite these encouraging statistics, civil aviation transportation continues to be considered by many people as a dangerous media of transportation, making people afraid of flying. Many aviation industries and partners keep trying to transform the image of aviation into a safer transport media by working together with the world-wide aviation community to reduce the accidents by continuously introducing improvements that will guarantee the safety of the passengers.

The National Aeronautics and Space Administration (NASA) and the National Transportation Safety Board (NTSB) have in recent years launched a series of initiatives among them the NASA Aviation Safety Program, which is part of the “*Three Pillars for Success*” [2], that has as a goal to reduce the aircraft accident rate by a factor of five within 10 years, and by a factor of 10 within 25 years [3].

Initiatives like the ones taken by NASA and NTSB have encouraged a new brand of engineers that are trying to introduce new technologies that have long been prove to be a good candidate to solve many of the problems that have caused aviation accidents. Many of the reported commercial accidents are related to the loss or malfunctioning of actuator surfaces, such physical damage of control or aerodynamic surfaces, control saturation, or stuck control surfaces.

When one of these malfunctions are present, due to the natural ability of the human brain to solve extremely complex problems, if a pilot is given enough time to detect the malfunction, he or she would be able to determine the controls available, adapt to the existing model, and have a chance to control the damaged aircraft. But in reality, when a malfunction with the potential of causing a crash occurs, the pilots have limited amount of time to react, and generally they have no knowledge of the nature of the failures. This makes it almost impossible to determine the amplitude of the failures and react on time according to those malfunctions.

Researchers are hoping that neural networks (NN), and/or similar technologies that possess natural abilities to solve extremely complex problems, adapt and learn from unknown situations, will pave the way to solving many of the problems presented by the different aviation safety boards to try to reduce the rate of accidents.

It is important to note that the events here described as malfunctions or accidents are extremely rare, but despite the low percentage rate of accidents, it is the duty of the aviation industry to find ways to reduce that rate of accidents to a minimum. A major breakthrough in aviation control laws would be being able to design a Reconfigurable Flight Control (RFC) system that would adapt in real-time to compensate for in-flight aircraft damage and system failures, with the important feature that the RFC would not have a priori knowledge of the nature of the failures or damages.

This task seems difficult if not impossible a priori, but in recent years, a great deal of research has been dedicated to solve this problem. RFC, in the form of a neurodynamic programming system, has been suggested by Werbos [4] as the best possible candidate for trying to “minimize the probability of a crash after and aircraft has experienced unforeseen, unpredictable damage so severe that no controller can absolutely guarantee stability” [4]. According to Werbos, another type of controller could not meet the necessary reconfigurability requirements, and despite some good results that have been shown by reconfigurable models using methods other than neurocontrollers, they have been done so by taking unreasonable assumptions that would not be realistic in real life situations. The purpose of this thesis is not to debate whether neurocontrollers are the appropriate choice for RFC, but to show that the Adaptive Critic Neural Network (ACNN) design described in this work shows very promising results towards the

common goal of solving the RFC problem. As it will be seen in section 1.3.5, several alternatives to neurocontrollers have also been suggested and have shown promising results.

NASA and the National Science Foundation (NSF) have been indicating their interest in the past decade to provide the means and resources that will allow the possibility of designing a feasible neurocontroller to increase the possibilities of survivability of an aircraft in the case of malfunctions and/or damages to an airplane. RFC really took off as a major research investment after an initial \$4 million contract from NASA-Ames to McDonnell-Douglas in St Louis. The contract was managed by Charles Jorgensen and the funding for that project was made possible by breakthrough results by Whites and Urnes, described in Chapter 7 of [5].

The Neuro Engineering Laboratory at NASA Ames Research Center is developing flight control software that uses NN to compensate for failures resulting from control surface damage or system malfunctions [6]. A NN is defined by the Neuro Engineering Laboratory at NASA Ames as a piece of software that works through pattern recognition, and thus is able to "learn" from sensory input, resulting in systems that can adapt to changes in their external circumstances. A more detailed definition of what is a neural network, and what it can do, will be done in sections 1.2.2, 1.2.3, and 1.2.4. NN have been the subject of theoretical and applied research since the early 1960's. They have been applied to a wide range of problems: from investment analysis to the control of modern equipment. The Information Technology Base Program (ITBP) has funded intelligent flight control research since 1997 as part of its *Intelligent System, Controls, and Operations Project* [6]. One of the fundamental objectives of the ITBP is to understand neuron processing in the human brain in order to produce computing architectures that make use of new biologically inspired capabilities. Section 1.2 will try to provide a background for the understanding of how neuron processing works in the human brain, and how that understanding has motivated many scientists.

1.2. NEUROBIOLOGY.

The author has taken the path of neurocontroller to try to solve some of the challenges that involve making aviation a safer place despite the existence of many other

controller architectures that are considered as more conventional controllers for today's aviation, such PID controllers, gain scheduling, feedback linearization, sliding mode to name few. It is imperative that before the author proceeds into describing in more detail the different neurocontrollers used today, to take the time to introduce the origins of neurobiology, and define some concepts of modern neurobiology, which have been an inspiration to many modern NN architectures, and especially to the author.

1.2.1. Origins of Neurobiology. Santiago Ramón y Cajal was one of the most outstanding neuroscientists of all time. He has been considered the father of modern neurobiology for his contributions and revolutionary views on the structure and functioning of the human brain, which are still used in today's modern neurobiology studies. Dr. Ramón y Cajal was born in Petilla de Aragón, a small village in the North of Spain. He studied medicine in the Faculty of Medicine in Zaragoza [7].

Dr. Ramón y Cajal received numerous prizes, honorary degrees and distinctions, among the most important being the Nobel Prize for Physiology or Medicine in 1906. To describe the work of Dr. Ramón y Cajal is a rather difficult task, and beyond the scope of this thesis because, unlike other great scientists, he is not known for only one discovery, but for his many and important contributions to our knowledge of the organization of the nervous system. For references in Dr. Ramón y Cajal's work, the reader can refer to some of his published work [8].

At the break of the 19th century little was known about the functioning or the structure of the human brain, although it was believed to be the center of the human thinking. It was also wrongly believed that human thinking was achieved by a continuous organ that spanned the entire human body. Dr. Ramón y Cajal disagreed with that view, and believed that human thinking was formed by a series of interconnected organs, but at the time he did not have the means of proving his theories.

Around 1885 Dr. Ramón y Cajal's studies turned around 180 degrees. At the age of 35 he was first exposed to a new visualization technique that had been around for almost 14 years introduced by Camillo Golgi, an Italian scientist that was born in July 1843 in Corteno, a village in the mountains near Brescia in northern Italy [9]. This new visualization technique consisted in a new impregnation technique of the cells, which was incredibly far more advanced than some of the existing staining techniques of the time.

This impregnation technique was the fruit of a work that started in 1872, when due to financial problems, Golgi had to interrupt his academic commitment, and accept the post of Chief Medical Officer at the Hospital of Chronically Ill (*Pio Luogo degli Incurabili*) in Abbiategrasso, a village near Pavia and Milan. In the seclusion of this hospital, he transformed a little kitchen into a rudimentary laboratory, and continued his search for a new staining technique for the nervous tissue. In 1873 he published a short note in the *Gazzetta Medica Italiana*, in which he described that he could observe the elements of the nervous tissue "studying metallic impregnations... after a long series of attempts" [10]. This was the discovery of the black reaction, also known as *reazione nera*. The *reazione nera* was based on the property of nervous tissue hardening in potassium bichromate and impregnated with silver nitrate. This revolutionary staining technique, which is still in use today, was named after him, *Golgi staining* or *Golgi impregnation*. The revolution of the process consisted in the fact that only a limited number of neurons were impregnated at random once the nervous cells were exposed with the silver nitrate. The reason for the random impregnation is still a mystery today. This new impregnation technique permitted for the first time a clear visualization of a nerve cell body with all its processes in its entirety [11].

This was what Dr. Ramón y Cajal needed to prove his theories, and by 1891, was receiving wide support from studies pursued in other laboratories that were using Golgi's new staining technique. This new visualization technique provided Dr. Ramón y Cajal with the means to prove that the architecture of the neural system was formed by a huge amount of individuals rather than one single organ. At the time, the best way to get his discoveries to other scientists was to draw by hand the images that he observed in the rudimentary microscopes of the time. These drawings were shown in symposiums more or less like we do today, with the only difference that today we use computers and transparencies.

The transcendence and importance of Dr. Ramón y Cajal's work towards the learning of how the human brain works has granted him the importance, and the privilege that his original glass microscope slides were the first historically important scientific artifacts to be flown in space. A series of these slides and some of the drawings made by Dr. Ramón y Cajal were stowed aboard Columbia in one of the NeuroLab Missions. The

slides and drawings included every part of the nervous system to be studied on the Neurolab missions [12].

Through initiatives of NASA and other international space agencies the Neurolab Mission, which specialized in the study of cerebral functions, began on April 17 1998 with the launching of the space shuttle Columbia, flight STS90, from Kennedy Space Center, Cape Canaveral, Florida. Studies were centered on the analysis of micro-gravity effects on the nervous system. Two *Instituto Ramón y Cajal* laboratories, conducted by Javier DeFelipe and Luis Miguel García-Segura, participated in the scientific project.

In honor of Dr. Ramón y Cajal, NASA included a sample of his scientific legacy in the Neurolab mission, thus recognizing the many and important contributions of the illustrious Nobel Prize in Medicine and Physiology to neuroscience. Twelve histological preparations and nine original scientific drawings, belonging to the *Dr. Ramón y Cajal Legacy* [13], traveled through space. Once the space flight ended, the histological preparations and scientific drawings, temporarily lent to NASA, were exhibited to the general public at the NASA Museum for one year [13]. Dr. Ramón y Cajal's observations and theories remain strongly valid at present, and have encouraged a vast amount of scientist and researchers that have seen in artificial neurobiology, the door to be able to make this world a better and safer place.

1.2.2. Modern Neurobiology. The questions that intrigues scientists, engineers, and doctors all over the world, is: How does the human brain really work? What is in the human brain that makes it so fascinating? What can and cannot be accomplished by the human brain? The answer to these questions, and many more that have intrigued scientists, are not as clear as we would like, and although scientists have been able to locate the areas of the brain that get activated when we conduct everyday tasks like hearing, seeing, speaking, or creating, its complexity leaves lots of room for speculation and imagination. The human brain is an organ that has more than 100 billion cells all working in unison, in a complex connected network. Dr. Ramón y Cajal's intuitive description of the neural net was very accurate according to studies that have been conducted in recent years. The neurons that form these nets, which Dr. Ramón y Cajal called the *butterflies of the soul* [14], have been described as polarized cells that receive signals on highly branched extensions of their bodies, called dendrites, and send

information along unbranched extensions, called axons. One of the questions that intrigued Dr. Ramón y Cajal was to find out how those *butterflies of the soul* flapped their wings and were able to communicate with each other. A more detailed knowledge of how the neurons work, and interact with each other, has been produced in recent years.

When a neuron has been excited, it conveys information to other neurons by generating impulses known as action-potentials. These signals propagate like waves down the length of the cell's single axon and are converted to chemical signals at the synapses, which are the contact point between neurons [14]. Figure 1.1 shows a drawing of a typical interaction between neurons.

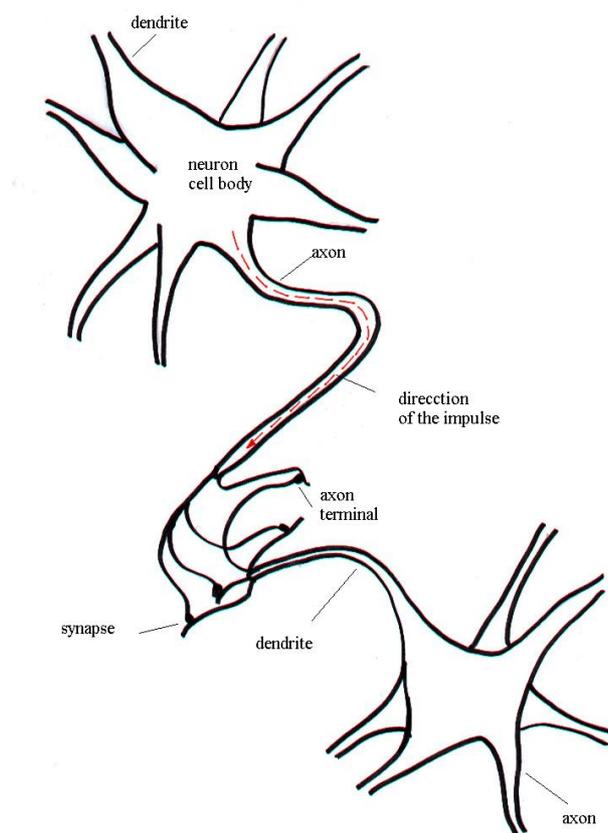


Figure 1.1. Typical Interaction Between Neurons.

When a neuron is at rest, its external membrane maintains an electric potential difference of about -70 millivolts. This means that the inner surface is negative relative to the outer surface. At rest the membrane is more permeable to potassium ions than to sodium ions, as indicated by the length of the arrows in Figure 1.2.

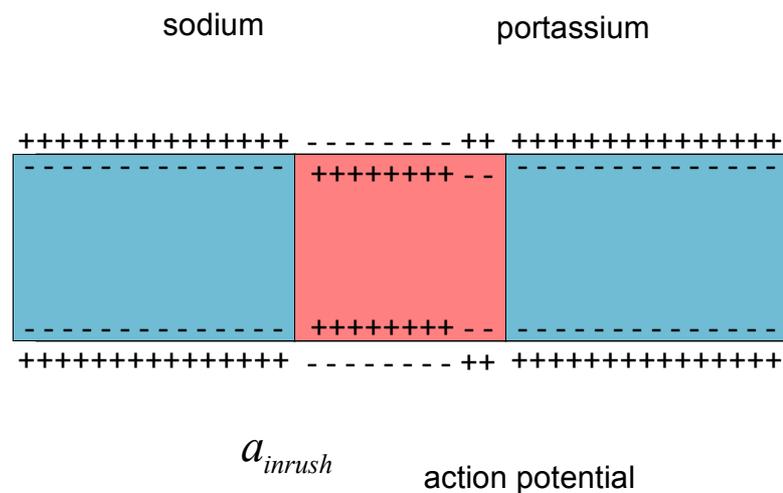


Figure 1.2. Sodium-Potassium Permeability.

When the cell is stimulated by another neuron, the permeability to the sodium increases to an inrush of positive charges, which is defined as a_{inrush} in Figures 1.2 and 1.3. This inrush triggers an impulse or momentary reversal of the membrane potential, as seen in $a_{reversal}$ in Figure 1.3. This impulse is initiated at the junction of the cell body and the axon and is conducted away from the cell body towards the next neuron.

When the impulse reaches the axon terminals of the pre-synaptic cleft of the neuron, it induces the release of neurotransmitter molecules. These neurotransmitters diffuse across a narrow cleft and bind to receptors in the postsynaptic membrane. The binding of the neurotransmitters to the receptors of the postsynaptic membrane leads to

the opening of ion channels and often, in turn, to the generation of action potentials in the postsynaptic neuron [14]. This can be seen in Figure 1.4.

The synaptic cleft mentioned in the previous paragraph is the small gap that exists between the neurons. Dr. Ramón y Cajal felt there was a gap, but he had no idea how small it was. With the advance in electronic microscopes, the gap has been measured to be around 20-30 nm, that is nanometers, or millionth of a millimeter, which probably explains why Dr. Ramón y Cajal was not able to see it with a regular light microscope [15].

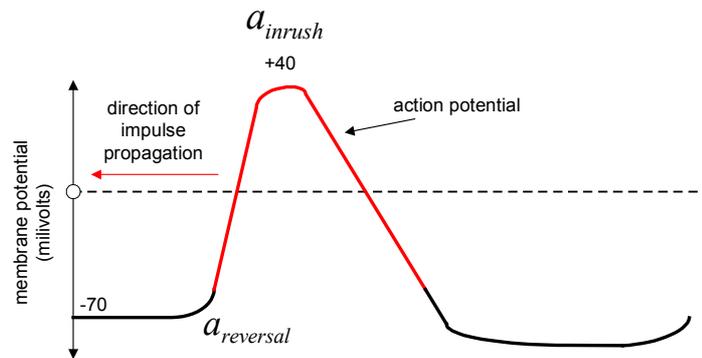


Figure 1.3. Detail of the Membrane Potential Reversal.

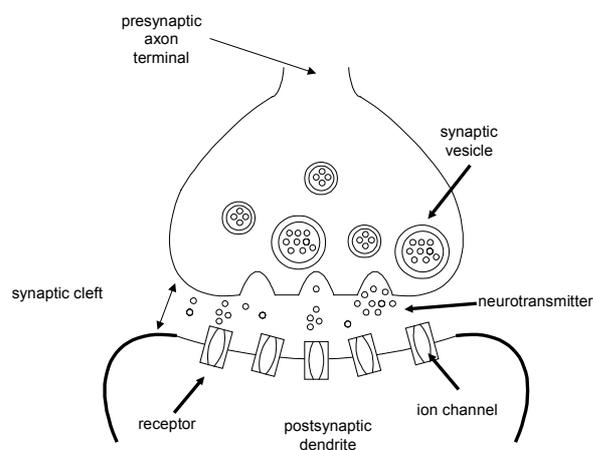


Figure 1.4. Detail of the Neural Synaptic Cleft.

1.3. ARTIFICIAL NEURAL NETWORKS.

The complexity of the human brain has long fascinated human kind, not only the last few decades. During the last century, the idea of trying to use the knowledge obtained in human neurobiology to solve the most complex, difficult and challenging problems gained wide acceptance. With more than 100 billion neurons all working at the same time, the human brain forms an incredible complex biological neural net that is able to solve extremely complicated tasks in tenths of a second. Some of these tasks, which seem so easy to perform to the eyes of any kid, seem impossible to solve or mimic even with the advance of today's technology:

- A baseball outfielder being able to track and catch a baseball despite not having any empirical knowledge about the speed or the trajectory of the ball.
- A tennis player being able to return a service that comes at speeds in excess of 100 miles per hour and hit it with the precision required to place the ball at will.
- A person that catches almost instantaneously a glass of water that has been tipped over the table.
- Remembering a list of telephone numbers, and associate them to a particular face or place.
- Maintaining balance with one leg, and if we think about it, even maintaining balance with two legs!
- Learning to read or write, and speak more than one language with fluidity.
- Painting, composing songs, and many more everyday tasks that we take for granted.

Do we realize the job that our brain does when performing these and many other tasks? Some of these tasks seem so simple because they belong to events that we perform or see being performed everyday, but we have to remember that when we are little kids, all these tasks seem almost impossible to accomplish for us. To be able to excel in those tasks, lots of training is required, and what some people might call natural abilities or skills when we observe the grade of perfection that they have achieved

performing some of the more complex tasks, the truth is that person has a better predisposition to learn. The readiness of the human brain to learn makes people stand out and seem different in all areas of our society. These learning capabilities are what have inspired the branch of artificial intelligence (AI).

But not all tasks that the human brain performs everyday require extensive training and skills. Tasks that are crucial to us to survive are regulated by the involuntary region of the brain. Breathing, pumping blood, regulating temperature in the body, generating energy, sensing, vision, smelling, touch, taste, and many more vital tasks, are regulated and controlled by the brain at the same time, and again we seem to take them for granted. Only when we try to model these tasks into everyday engineering problems, such as controlling chemical process, room temperature control, robots actuators, and many more, we then realize how complicated these tasks are.

The quest for systems that would be able to emulate some of these tasks that seem so simple for so many of nature's creations, has been the holy grail for many scientists and engineers over the years, and the path to that success has always been linked to the understanding of the functioning of the human brain. Scientists have tried since the early 1940's to use artificial neurons like formulations to try to solve complex problems. The following section tries to describe some of the relevant historic events that define the origins of artificial NN.

1.3.1. History of Artificial Neural Networks. During the second half of the 20th century the connection between the brain and digital computers was made and generated a great deal of interest. This led to the creation of the first models on how the human brain could work. In 1943, neurophysiologist Warren McCulloch and mathematician Walter Pitts wrote a paper on how neurons might work by modeling a simple neural network using electrical circuits [16].

McCulloch and Pitts showed that networks of artificial neurons could, in principle, compute any arithmetic or logical function [17]. Their work is often acknowledged as the origin of the neural network field. The main feature of their neuron model was that the weighted sum of input signals was compared to a threshold to determine the neuron output. When the weighted sum of input signals was greater than or equal to the threshold, the output was set to zero.

McCulloch and Pitts were followed in 1949 by Donald Hebb [18], whose work pointed out the fact that neural pathways are strengthened each time they are used, a concept fundamentally essential to the ways in which humans learn. Hebb argued that if two nerves fire at the same time, the connection between them is reinforced. Hebb proposed a possible mechanism for synaptic modifications of the brain, which has been used since then to train artificial NN [19].

As computers became more advanced in the 1950's, it was finally possible to simulate hypothetical NN. Nathaniel Rochester made the first step towards these simulations from the IBM research laboratories. Unfortunately for him, the first attempt to do so failed [16]. The first practical application of artificial NN did not come until the late 1950s, with the invention of the perceptron network and associated learning rule by Frank Rosenblatt [20]. Rosenblatt and his colleagues built a perceptron network and demonstrated its ability to perform pattern recognition. This early success generated a great deal of interest in neural network research. For a more detailed description of what a perceptron is refer to section 1.3.2 in this thesis. Unfortunately, it was later shown that the basic perceptron network design could solve only a limited class of problems. Minsky and Papert [21] demonstrated that due to the limitation of the perceptron, this design was not able to implement certain elementary functions, but this problem was solved later on in the 1980's with the introduction of multiplayer perceptron networks and advancements in learning rules.

In 1959, Bernard Widrow and his graduate student Marcian Hoff, of Stanford, developed NN models called the "ADALINE" (ADaptive LInear NEuron) network. The ADALINE network had the limitation that could only solve linearly separable problems [22]. The ADALINE network was developed to recognize binary patterns so that if it was reading streaming bits from a phone line, it could predict the next bit. ADALINE was the first real world application of a neural network using an adaptive filter that eliminated echoes on phone lines. In 1962, Widrow and Hoff developed a learning procedure called the Least Mean square (LMS), which minimized the mean square error and therefore tried to move the decision boundaries as far from the training pattern as possible [19].

Despite their initial success, both Rosenblatt's and Widrow's networks suffered from the limitations described by Minsky and Papert [21]. At that point many researchers, influenced by the ideas proposed by Minsky and Papert, believed that the research in NN had reached a dead end and this, along with the arrival of the digital computers, helped to put a stop on the field of NN for an entire decade. In addition to Minsky and Papert's paper, many people in the field were using a learning function that was fundamentally flawed because it was not differentiable across the entire line. As a result, research and funding went down drastically. Traditional von Neumann architectures took over the computing scene, and the neural research was left behind. John Louis von Neumann was the synthesizer and promoter of the stored program concept, whose logical design of the Institute for Advanced Studies (IAS) became the prototype of most of its successors. The von Neumann Architecture corresponds to the basic structural computer model with memory units, processing units, control units and input/output (I/O) units. Ironically, John von Neumann himself suggested the imitation of neural functions by using telegraph relays or vacuum tubes [23].

Research in NN emerged again during the 1970s with scientists that believed in NN. Some of the most important scientists that emerged during the 1970s were Kohonen, Anderson and Grossberg. In 1972, Kohonen [24] and Anderson [25] developed a similar network architectures, independently of one another, which could act as memories. The Kohonen rule allowed the weights of the neuron to learn an input vector that was suitable for recognition applications [26]. Grossberg's work was geared towards modeling specific functions of mind and brain by using nonlinear mathematics [27]. Grossberg was highly influenced by the interdisciplinary work done in brain function by Helmholtz, Maxwell and Mach and this was shown in the diversity of his research which could not be placed in one research area, but in several. Grossberg acknowledged that in order to solve the intelligence problem with NN it was necessary to equally approach the areas of mathematics, psychology and neurophysiology [19].

In 1982, John Hopfield of Caltech, again renewed interest in the field of NN. Hopfield presented a paper to the National Academy of Sciences, using the approach of creating more useful neural network machines by using bidirectional connection lines [29]. Previously, the connections between neurons were only one way. Hopfield used

statistical mechanics to explain the operation of certain classes of recurrent networks, which could be used as an associative memory. Hopfield's papers were identified as a close analogy between his neural network and the *Ising* memory model of magnetic materials, which was used in statistical physics. This attracted many physicists' attention towards the field of NN [19]. Hopfield's views of NN emphasized their use for practical problems instead of trying to solve very complex problems. It was a clear statement that in order to run, first it was needed to crawl. Some of those applications were described in several of his papers [28] and [29] such as analog-to-digital conversion and optimization problems such as the salesman problem.

That same year, Reilly and Cooper used a "Hybrid network" with multiple layers, each one using a different problem-solving strategy. Also in 1982, there was a joint US-Japan conference on Cooperative/Competitive NN in which Japan announced a new Fifth Generation computing effort, which involved AI on NN, which generated worry among US researchers that felt the US could be left behind in the field of NN. The importance in announcing Fifth generation computing efforts can be understood by defining the different generations of computing. The first generation computing effort used switches and wires, second generation used transistors, third generation used solid-state technology like integrated circuits and higher level programming languages, and the fourth generation consisted in code generators. As a result, there was more funding and thus more research in the field [16].

A second revolution emerged in the middle 1980s, with the introduction of backpropagation algorithms for training multilayered perceptron networks and hence overcoming the inherent limitations of single layer perceptrons that Minsky and Papert described in the 1960s. The first description of an algorithm to train multilayer networks was contained in the dissertation of Ph D. Candidate Paul Werbos [30]. The thesis presented a generic algorithm in the context of general networks and was not that extended among the neural network community. It wasn't until the mid 1980s, when backpropagation was rediscovered independently and widely publicized by David Rumelhart, Geoffrey Hinton and Ronald Williams [31], David Parker [32], and Yann Le Cun [33]. Backpropagation represented a generalization of the learning rules introduced by Widrow and Hoff by sharing some of the concepts with LMS, such that both were the

approximate steepest descent algorithms with the performance index being the mean square error. Backpropagation learning rules signified a breakthrough in neural network training in the way in which the derivatives for the update rules were calculated. For a single layer linear network the error was an explicit linear function of the networks weights, and its derivatives with respect to the weights, which in return could easily be computed. In multilayer networks, weights and the errors are more complex and in order to calculate the derivatives, the chain rule of calculus is required [19].

From this point on, the history of NN has been engrossing their arcs with thousands of papers in the area of NN. What seemed to be a lost end research in the 1960's, has become a widely popular and mature technology that has found its place in many areas of research, including medicine, engineering, learning and many others.

1.3.2. What is an Artificial Neural Network? Neural networks are part of a group of intelligence technologies for data analysis that differ from other classical analysis techniques by learning about a subject from the data provided to the networks rather than being programmed by the user in a traditional sense. NN gather their knowledge by detecting the patterns and relationships in the sets of data, learning from relationships and adapting to change [34]. Through the years, biological models of the brain were developed and inspired scientists to try to solve complex problems. The introduction of the perceptron by Rosenblatt in 1959 opened the door for trying to mimic biological neurons.

The perceptron is a mathematical model of a biological neuron. While in actual biological neurons, the dendrite receives electrical signals from the axons of other neurons. In the perceptron, these electrical signals are represented as numerical values. At the synapses between the dendrite and axons, electrical signals are modulated in various amounts. As described in section 1.1.3, the synapses lead to the generation of action-potentials in the postsynaptic neuron, which in turn, is the ultimate cause of the decision-making process. This is also modeled in the perceptron by multiplying each input value by a value called the weight. An actual biological neuron fires an output signal only when the total strength of the input signals exceeds a certain threshold. This is modeled in artificial neurons by calculating the weighted sum of the inputs to represent the total strength of the input signals, and applying an activation or transfer function on

the sum to determine its output. As in biological NN, perceptrons are interconnected with each other such that the output of each perceptron is fed to other perceptrons. Figure 1.5 shows a sample of an artificial neuron in which the connections to the left of the weights represent the inputs that might come from the connections of other perceptrons. After the different inputs, or signals are summed, an activation or transfer function is used to map the inputs to satisfy the specifications of the problem being solved.

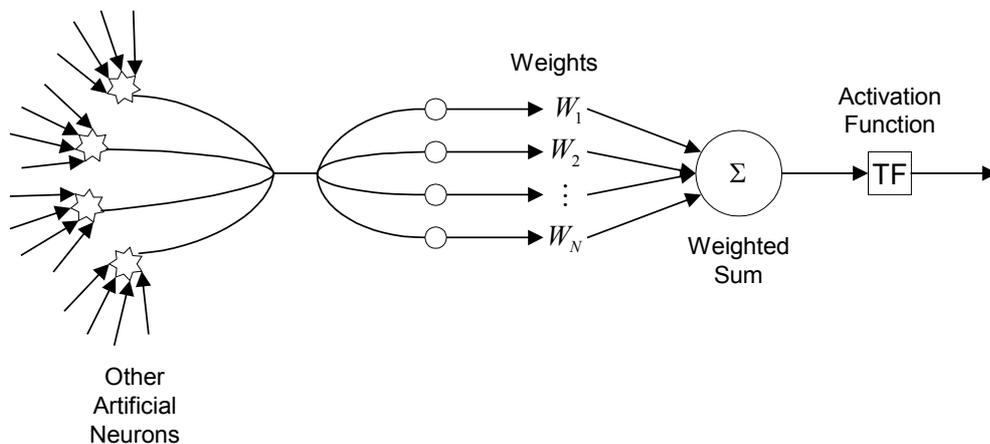


Figure 1.5. An artificial Neuron or Perceptron.

The choice of transfer functions is left to the user. The transfer functions, also called activation functions, can be linear or nonlinear, continuous or non-continuous functions depending on the nature of the problem to be solved. Some of the transfer functions used are hard limit, linear, saturation, sigmoid, hyperbolic tangent sigmoid, or competitive functions. The reader can refer to [19] for further details about transfer functions. Section 3.1 will explain in more detail the type of nonlinear continuous transfer function chosen for the NN design used in this thesis.

1.3.3. Classification of Neural Networks. In recent years, many papers have been published on NN controllers. According to Werbos [35], these fall under five classes, and he describes each one as:

- Supervised Control: in which a neural net learns the mapping from sensor inputs to desired actions by adapting to a training set of examples of what it should have done.
- Direct Inverse Control: in which a neural net learns the inverse dynamics of a system so that it can make the system follow a desired trajectory.
- Neural Adaptive Control: linear mappings are replaced by neural nets resulting in greater robustness and ability to handle nonlinearities.
- Back Propagation Through Time: adapts an optimal controller essentially by solving a calculus of variations problem.
- Adaptive Critic Methods: approximate the Bellman equation of dynamic programming.

Adaptive critic designs include a large family of methods, which attempt to approximate dynamic programming in the general case. The dynamic programming approach for solving the optimal control problem will be discussed in more detail in section 2.1.3. A version of the adaptive critic approach, first introduced by Balakrishnan and Viegas [36], is used in this thesis and uses two sets of NN, one named the Critic Neural Network (CNN), which captures the relationship between states and co-states, or Lagrange multipliers at the node points. The other set of networks, named the Action Neural Network (ANN), captures the relationship between states and control. After successful synthesis of both networks, the results yield an optimal control in a feedback form. A more detailed description on the synthesis of Adaptive Critic Neural Network (ACNN) methodology presented in this study is described in section 3. In this study, the ACNN is developed to obtain a near optimal controller to extend the stall regime for an aircraft.

1.3.4. What Can and Cannot be Done with Artificial Neural Networks? NN have the capability to solve extremely complex problems due to their ability to form highly complex architectures. These capabilities have not been fully utilized in real world applications, but can be seen in every day applications that include voice, image or pattern recognition.

NN have gained a lot of attention in the field of control in the last ten years. In many fields, including aerospace, engineering systems whose response we want to control are becoming more and more complex. With dwindling funds for development of new aircraft, missile, space systems etc., activities have increased in trying to use current existing systems in more stressed environments. A basic requirement for good performance or even stable performance in such situations is the development of proper control systems. Traditional control designs, based on classical control theory, have prove to be inadequate in such situations. Although a huge amount of time and studies over the past twenty years have been devoted to linear system based robust control, there are few actual systems with robust controllers obtained from norm-based theories. Nonlinearities of many dynamic systems have generated interest in the studies beyond the linearized notions in their development. This sets the stage for the field of NN, which are basically nonlinear mappings, some references are listed [35], [36], [37], and [38].

In 1988, the Defense Advanced Research Projects Agency (DARPA) presented the *DARPA Neural Network Study*, in which a series of possible applications for NN were listed. Areas included, aerospace, automotive, banking, defense, electronics, entertainment, financial, insurance, manufacturing, medical, oil and gas, robotics, speech, securities, telecommunications and transportation [39]. The wide range of possible applications for NN shows that the limit of how far NN can go only depends in the imagination of the scientists.

Many scientists, skeptics and believers, have raised the question of how far can NN go? What are they good for and what are their limitations? Can they ever model the human consciousness and emotions as we have seen in so many science fiction movies? If we ever reach the potential to do so, how far should we let NN go? There are those scientists that think that with NN we will be able to have a better understanding of how the human brain works and thus both medical field and scientific fields will gain from

each other by trying to reproduce AI with computers. On the other hand, there are also those that think human thought will never be simulated with NN or any other computational tool. In his book *Shadows of the Mind*, Roger Penrose [40] suggests that deep problems in AI, physics, and the philosophy of mind are closely connected, also suggested by Grossberg [26]. Penrose presents a detailed argument, using Gödel's theorem, for the conclusion that human thought cannot, and will never be simulated by any computation [40].

Feelings like pain, vision, touch, taste and smell, have all been physically linked to different parts of the brain and scientists have been able to describe, more or less, how they work. But not even scientists can get a grasp of how love, fear, happiness or morals, to name a few, are generated in the brain. Scientists have been able to locate the parts of the brain that regulate the release of chemicals that control some of those emotions but raises another question: Do we limit the explanation of the feelings of a person by associating them to mere chemical and hormone releases in certain parts of the brain? Or is there much more to it? The issue of AI is a very complex and sensitive one since there is much more to it than if scientists can ever model something with such complexities. Lots of moral issues will be raised if the technology one day reaches the point in which science fiction might not be a mere illusion, and robots may have dreams like the late Isaac Asimov wrote in his *Robot Dreams*.

Again, this moral issue is not the purpose of this thesis, but the author felt it necessary to raise couple of these issues, to make the reader aware that the ultimate purpose of this work is not to just to show how well a certain NN performs relative to some conventional methods, but show that in order to create the ultimate RFC system, we need to focus our attention to the ultimate RFC machine, human beings, who are able to acquire large amounts of data and make decisions weighing all possible scenarios.

1.3.5. Other Approaches? NN are really powerful mapping tools that can be trained and thus learn by being fed with data related to the problem. It has been proven that with the correct number of layer and artificial neurons, any nonlinear function can be modeled with artificial NN and be able to solve extremely complex problems which open the door to use tools that until now were not viable due to the restrictive computational requirements. But NN also have problems and limitations when trying to

achieve intelligent control. NN are extremely efficient tools when it comes to solving complicated problems, and have shown robustness for a large trained envelope, but what happens when the problem diverges completely from the envelope at which they were trained, or if the problem changes completely?

It will be shown in section 2.2.2, that some non-classical optimal control techniques offer the possibility of designing controllers for a wider range of conditions by using a series of linear controllers that are parameterized in several operating points by one or more variables. These systems are then linearized at the chosen points and linear feedback controllers are designed and implemented at each point. A problem might be encountered if the parameterized variables used to determine which operating point the system is located at falls out of the range at which it was trained, or the problem is changed completely. Similar problems will be encountered for any type of controllers that do not possess reconfigurability properties. Humans possess the ability to make intelligent decisions under stress situations, even when they do not have all the necessary information or when the environment changes completely. NN ability to solve extremely complex problems could benefit from some of the computational approaches that try to mimic the way in which humans think under those stress situations.

One of those approaches is fuzzy logic. Fuzzy logic was first introduced in 1965 by professor Lotfi Zadeh, in the shape of defining the basis for fuzzy computer chips [41]. Despite their name, fuzzy logic is not a vague technology. Fuzzy logic is a type of logic that recognizes more than simple true and false values, such as computers do, and provides a very precise approach for dealing with uncertainty, which grows out of the complexity of human behavior [42].

Traditional or classical logic tries to categorize information into binary patterns which are the basis of the computer language, while fuzzy logic looks at the partially true and partially false situations which make up 99.9% of human reasoning in everyday life. When analyzing human reasoning we observe that many of the choices we make are based on incomplete information that is not clearly defined as black or white. Simplifying problems when not all information is available, when only part of the information is available, or simply when we are not able to handle all the information, generates some of the best human decisions. By reducing the complexity of the decision-

making by relating similar information thus reducing the complexity of the problem we are able to react faster and more efficient in everyday situations. When doing this, instead of trying to generate artificial mathematical models of the real world, fuzzy logic is able to extremely simplify some complex problems and provide solutions that although might not have the accuracy of some of the optimal solutions obtained with classical control theory or NN based controllers, they provide a solution that is cheaper, simpler and far more effective than other systems when it comes to real life implementations.

By integrating fuzzy logic with the power of NN, we would be able to gain the consciousness of making fast and optimal decisions while being able to adjust to situations that are not well defined and might cause a computer not to understand. Fuzzy logic and NN have been successfully integrated for uses as diverse as automotive engineering, applicant screening for jobs, the control of a crane and the monitoring of glaucoma [42].

Mohammad Jamshidi [43] describes some of these innovative approaches, which combine fuzzy logic with other technologies, in his book *Applications of Fuzzy Logic*. In this book, the world's leading fuzzy logic experts present detailed coverage of new applications that deliver exceptionally high levels of autonomy and adaptability. These high machine intelligence quotient systems combine fuzzy logic with related techniques such as NN and genetic algorithms, leading to significantly improved performance in a wide variety of control systems.

Another interesting approach taken by many researchers focus their attention on natural selection, which was first introduced by Charles Darwin in his book *On the Origin of Species by Means of Natural Selection* published in 1859. Darwin described that nature favors the strongest and the fittest species to survive. The fittest will move on and reproduce while the weak die out. This natural selection is extrapolated to processes like Evolutionary Strategy (ES) and Genetic Algorithms (GA), which were first introduced by Rechenberg in 1973 [44], and Holland in 1975 [45], respectively.

In 1963 two students at the Technical University of Berlin (TUB) met and were soon to collaborate on experiments, which used the wind tunnel at the Institute of Flow Engineering. During the search for the optimal shapes of bodies in a flow, which was then a matter of laborious intuitive experimentation, the idea was conceived of

proceeding strategically. However, attempts with the coordinate and simple gradient strategies were unsuccessful. Then one of the students, Ingo Rechenberg hit upon the idea of trying random changes in the parameters defining the shape, following the example of natural mutation. The ES strategy was born [46]. ES was invented to solve technical optimization problems (TOPs) like constructing an optimal flashing nozzle. Usually no closed form analytical objective function is available for TOPs and hence, no applicable optimization method exists, but the engineer's intuition.

GA's are models of machine learning, which derives its behavior from a metaphor of the processes of evolution as seen in nature. This is done by the creation within a machine of a population of individuals represented by artificial chromosomes similar in concept to the base-4 chromosomes that we see in our own DNA. The individuals in the population then go through a process of evolution following Darwin's maxim that the fittest will move on and reproduce while the weakest dies out [46].

These different approaches, and many others that try to take advantage of the knowledge gained by Mother Nature by many different means such as evolution (GA), mutation (ES), the complexity of human behavior (fuzzy logic), and the power and complexity of human thinking (NN), paves the path for researchers to try to achieve the difficult task of obtaining RFC.

2. OPTIMAL CONTROL

2.1. OPTIMAL CONTROL THEORY

The development of the modern control theory, the state-space method, and the calculus of variations make it possible to introduce optimal control theory. Optimal control is the unique method that can deal with a nonlinear system without any approximation. It has no limitation either for linear or nonlinear systems, and either for single input-output (I/O) or multiple I/O systems. It has been used to solve many complex nonlinear, multivariable problems in a variety of industrial settings, particularly in aerospace applications. The application of optimal control theory needs to clearly define the problem and identify the performance measures, which will be introduced in the following sections. It needs to be noted that since a computer will be used to implement the control in its feedback form, it is necessary to use the discrete-time formulation, which will be used in the following sections.

2.1.1. Definition of Optimal Control. The definition of optimal control is to find an admissible control, u_i^* , which causes a system to follow an admissible trajectory, x_i^* , that minimizes the discrete-time performance measure or cost function J [47]. Where the i subscript represents the control at time i of the discrete-time model.

$$J = \sum_{i=0}^{N-1} L_i(x_i, u_i) \quad (1)$$

where $L_i(\bullet)$ is a scalar, also known as the utility function. When N approaches infinity, ∞ , this is equivalent to an infinite-time regulator problem. The class of problems solved in this thesis is defined by the discrete-time system model given as

$$x_{i+1} = f_i(x_i, u_i) \quad (2)$$

where $f_i(\bullet)$ can be either, a linear or nonlinear combination of the states that describe the dynamics of the system, and has dimensions of $nx1$; the states, x_i , have dimensions $nx1$ and the control, u_i , has dimensions of $mx1$. Two approaches are

generally used to obtain a solution to the optimal control problem. One is the Pontryagin's minimum principle, also known as Hamiltonian formulation, the other dynamic programming. The following sections will introduce both methods.

2.1.2. Pontryagin Minimum Principle: Hamiltonian Formulation. In Hamiltonian formulation, the performance index described in Eq. (1) is optimized treating the state equations as constraints thus defining the discrete-time Hamiltonian, H_i , as

$$H_i(x_i, \lambda_{i+1}, u_i) = L_i(x_i, u_i) + \lambda_{i+1}^T f_i(x_i, u_i) \quad (3)$$

where H_i is a scalar, λ_{i+1} is the co-state at $i+1$ with dimensions $nx1$. The discrete-time necessary conditions for optimality, or Euler-Lagrange equations, are obtained by applying the calculus of variation to the Hamiltonian with respect to the states, co-state and control respectively yielding

$$\lambda_i = \left(\frac{\partial H_i}{\partial x_i} \right)^T = \left(\frac{\partial f_i}{\partial x_i} \right)^T \lambda_{i+1} + \left(\frac{\partial L_i}{\partial x_i} \right)^T \quad (4)$$

$$\left(\frac{\partial H_i}{\partial \lambda_{i+1}} \right)^T = f_i = x_{i+1} \quad (5)$$

$$\frac{\partial H_i}{\partial u_i} = \left(\frac{\partial f_i}{\partial u_i} \right)^T \lambda_{i+1} + \left(\frac{\partial L_i}{\partial u_i} \right)^T = 0 \quad (6)$$

Equations (4), (5) and (6) become the optimality Euler-Lagrange equations. A quadratic utility cost function is chosen such that

$$L_i = \frac{1}{2} x_i^T Q x_i + \frac{1}{2} u_i^T R u_i \quad (7)$$

where Q is the weighted state matrix and R is the weighted control matrix with dimensions $n \times n$ and $m \times m$ respectively. Substituting Eq. (7) into Eq. (3) the Hamiltonian can be expressed as

$$H_i(x_i, \lambda_{i+1}, u_i) = \frac{1}{2} x_i^T Q x_i + \frac{1}{2} u_i^T R u_i + \lambda_{i+1}^T f_i(x_i, u_i) \quad (8)$$

And after substituting Eq. (8) into the general optimality equations (4), (5), and (6), the necessary conditions for optimality become

$$\lambda_i = \left[\frac{\partial f_i}{\partial x_i} \right]^T \lambda_{i+1} + Q x_i \quad (9)$$

$$x_{i+1} = f_i(x_i, u_i) \quad (10)$$

$$u_i^* = -R^{-1} \left[\frac{\partial f_i}{\partial u_i} \right]^T \lambda_{i+1} \quad (11)$$

where u_i^* is the discrete-time optimal feedback control law, which is obtained by simultaneously solving Eqs. (9) and (10) while minimizing the new defined cost function

$$J = \frac{1}{2} \sum_{i=0}^{N-1} x_i^T Q x_i + u_i^T R u_i \quad (12)$$

2.1.3. Dynamic Programming. Dynamic programming is the other approach that can be used to obtain the solution to the optimal control problem, and as described in reference [4], the only exact and efficient method possible, in the general case for solving stochastic dynamic optimization problems. Dynamic programming provides a computational technique to apply the principle of optimality to a sequence of decisions that define an optimal control policy. The principle of optimality is described by Bellman [48] as follows:

“An optimal policy has the property that no matter what the previous decisions (i.e., controls) have been, the remaining decisions must constitute an optimal policy with regard to the resulting from those previous decisions.”

The Hamilton-Jacobi-Bellman (HJB) equation, which is a general mathematical description of the optimality conditions, can also be used as a direct representation of the principle of optimality. The Hamilton-Jacobi-Bellman equation for a discrete-time system is given by

$$J_i^*(x_i) = \min_{u_i} \left(L_i(x_i, u_i) + \left\langle J_{i+1}^* \left(\frac{x_{i+1}}{1+d} \right) \right\rangle \right) \quad (13)$$

where $J_i^*(x_i)$ is assumed to be the minimum cost associated with going from time i to the final time, and is a scalar. The optimal value is indicated by $*$. This application calls for a minimization over time, thus the “min” in Eq. (13). $L_i(x_i, u_i)$ is a user-supplied utility function [23], which is the cost of going from time i to time $i+1$; d is a user-supplied discount factor, generally assumed to be zero; “ $\langle \rangle$ ” denotes the expectation value; u_i should be the value which maximizes the right hand side of equation (13). Finally J_{i+1}^* is assumed to be the minimum cost associated with going from time $i+1$ to the final time. For conventional tracking problems, the utility function is generally chosen as the tracking error at time i such that $(x_i - r_i)^2$, where x_i is the actual state, and r_i is the desired value for the tracking state. In such case, dynamic programming gives the recipe for how to minimize the tracking error over all future time [23]. If both sides of equation (13) are partially differentiable with respect to the state i , and define

$$\lambda_i = \left(\frac{\partial J_i}{\partial x_i} \right)^T \quad (14)$$

where λ_i and x_i have dimensions of $nx1$, then

$$\lambda_i^T = \frac{\partial L_i(x_i, u_i)}{\partial x_i} + \frac{\partial L_i(x_i, u_i)}{\partial u_i} \frac{\partial u_i}{\partial x_i} + \left\langle \lambda_{i+1}^T \frac{\partial x_{i+1}}{\partial x_i} \right\rangle + \left\langle \lambda_{i+1}^T \frac{\partial x_{i+1}}{\partial u_i} \frac{\partial u_i}{\partial x_i} \right\rangle \quad (15)$$

From Eq. (15), it can be seen that if λ_{i+1} , $L_i(x_i, u_i)$, and the system model derivatives are known, then λ_i can be found. Similarly, Eq. (14) needs to be solved backward in time starting from the final time and proceeding to the initial time. Next, the optimality equation is defined as

$$\frac{\partial J_i}{\partial u_i} = 0 = \frac{\partial L_i(x_i, u_i)}{\partial u_i} + \lambda_{i+1}^T \frac{\partial x_{i+1}}{\partial u_i} \quad (16)$$

Dynamic programming uses these equations to aid in solving either an infinite horizon policy or to determine the control policy for a finite horizon problem.

2.1.4. The Optimal Tracking Problem. For the class of problems defined in this thesis, infinite horizon problems, when minimizing the cost function defined in Eq. (1), given by the utility function defined by Eq. (7), the necessary conditions for optimality are satisfied by solving backwards Eq. (9) and using that result in the feedback control law, described in Eq. (11). This formulation is only valid when we want to drive the perturbed states of the nominal system to zero, which is assumed to be the desired states. When the desired states are different than zero, then the utility function needs to be modified to minimize the tracking error of the state as

$$L_i = \frac{1}{2} [x_i - r]^T Q [x_i - r] + \frac{1}{2} [u_i - u_D]^T R [u_i - u_D] \quad (17)$$

where x_i is the state, r the desired value of the tracked states, u_i is the actual control, and u_D is the steady state control necessary to achieve the r . If the redefined cost function is substituted into Eq. (8), and then into the necessary conditions for optimality, Eqs. (9)-(11), this produces a new set of necessary conditions that need to be solved to achieve the optimal tracking control problem defined by

$$\lambda_i = \left[\frac{\partial f_i}{\partial x_i} \right]^T \lambda_{i+1} + Q[x_i - r_i] \quad (18)$$

$$u_i = -R^{-1} \left[\frac{\partial f_i}{\partial u_i} \right]^T \lambda_{i+1} + \bar{u} \quad (19)$$

From inspecting Eqs. (18) and (19), it can be observed that in order to solve the optimal tracking problem the desired tracking states, r , and steady state control, \bar{u} , are required. This represents a challenge, as well as an inconvenience, for this formulation since it implies that in order to solve the optimal control law for tracking a desired state, it is necessary to know a priori the steady state control that drives the system to the desired steady state. This approach will be tested in section 4.3 with a class of problems that allow the optimal tracking approach.

2.1.5. A Special Case - Linear Quadratic Regulator. A special case of the optimal control problem is that in which the discrete-time system model expressed in Eq. (2) can be expressed in the discrete-time state-space model such as:

$$x_{i+1} = Ax_i + Bu_i \quad (20)$$

where A represents the state matrix and B the control matrix. In these types of problems the approach followed is to calculate an optimal feedback gain matrix K such that the state-feedback law is provided by $u_i = -Kx_i$ and minimizes a quadratic cost function of the form

$$J(u) = \sum_{i=1}^{\infty} (x_i^T Q x_i + u_i^T R u_i + 2x_i^T N u_i) \quad (21)$$

Note that the utility function inside of the summation of Eq. (21) includes an extra term compared to the one defined in Eq. (12). In the same manner as the weighted state matrix, Q , and the weighted control matrix, R , N represents a matrix that include the

weighted cost for the cross terms between the states and the control. In the general case this matrix is zero. The approach for solving the optimal control problem for this class of problems consist of solving the associated steady state Riccati equation:

$$A^T S + SA - (SB + N)R^{-1}(B^T S + N^T) + Q = 0 \quad (22)$$

and assume a state-feedback gain K of the form

$$K = R^{-1}(B^T S + N^T) \quad (23)$$

This is known as the discrete-time Linear Quadratic Regulator (LQR) problem and the feedback control law takes the form of

$$u_i = -Kx_i = -R^{-1}(B^T S + N^T)x_i \quad (24)$$

where S represents the solution for the associated steady state Riccati equation. The conditions to guarantee a solution for the associated steady state Riccati equation are:

- The pair (A, B) is stabilizable.
- $R > 0$ and $Q - NR^{-1}N^T \geq 0$.
- $Q - NR^{-1}N^T, A - BR^{-1}N^T$ has no unobservable mode on the imaginary axis.

Although a nonlinear system needs to be linearized about a operating point in order to be able to use the discrete-time LQR approach, the insight knowledge provided by the discrete-time LQR solution, was of invaluable help during the analysis process for both, comparison and initialization of the NN formulation as will be seen in the following sections. Several mathematical optimization techniques are available to solve for the S that satisfies Eq. (22), and thus providing with the feedback control law, of the form described in Eq. (24). This thesis uses a built in MATLAB function called “dlqr.m,”

which in return calls a function called “dare.m” that solves the discrete-time algebraic Riccati equation by using algorithms by Arnold and Laub [49]. Throughout the remainder of this thesis the discrete-time LQR will be referred as LQR for simplicity.

2.2. NONLINEAR OPTIMAL CONTROL BACKGROUND

This section provides a background to the different control theory approaches to the nonlinear problems, from classical control methods to non-classical methods and ending with some of the latest approaches in NN, which will serve as an introduction to section 3 where the NN approach used in this thesis will be discussed in detail.

2.2.1. Classical Control. Many approaches have been used over the years to solve the problem of automated control. Classic control techniques try to obtain feedback control laws by conducting comprehensive analysis of the system model. Some of these classic control techniques include *design via root locus techniques*, *frequency response techniques* and *state space techniques* to name a few. Nise gives a detailed definition of all the classical controllers described above in his reference book *Control Systems Engineering* [50]. The author will try to recap some of the definitions contained in [50] in the next paragraphs.

In the *design via root locus*, the designer is able to choose the proper loop gain to meet a transient response specification by graphically analyzing both the transient and the stability information provided by the root locus. Since the transient response is dictated by the poles at a point in the root locus, this technique is limited to the transient responses and the steady-state error represented by points along where the root locus are available. In order to improve these limitations, cascade compensators are introduced in the form of ideal integral, or proportional-integral (PI) controller, ideal derivative or proportional derivative (PD) controller, proportional-plus-integral-plus-derivative (PID) controller, lag compensators, lead compensators, lag-lead compensators and feedback compensation.

Steady-state design compensators are implemented via PI controllers or lag compensators. PI controllers add a pole at the origin, thereby increasing the system type. Lag compensators, usually implemented with passive networks, do not place the pole at the origin but near it. Both methods add a zero very close to the pole in order not to affect the transient response.

The transient response design compensators are implemented through PD controllers or lead compensators. PD controllers add a zero to compensate the transient response, while lead compensators add a pole along with the zero. Lead compensators are usually passive networks. We can correct both transient response and steady-state error with a PID or lag-lead compensator. Both of these are simply combinations of the previously described compensators.

Feedback compensation can also be used to improve the transient response, where the compensator is placed in the feedback path. The feedback gain is used to change the compensator zero or the system's open-loop poles, giving the designer a wide choice of various root loci. The system gain is then varied to move along the selected root locus to the design point. An advantage of feedback compensation is the ability to design a fast response into a subsystem independently of the system's total response.

Other classical approach is the *design via frequency response*. This approach follows the same lines of root locus via gain adjustment with the difference that the tools used do not require a computer. Instead Bode plots and Nyquist diagrams are used along side each other to provide stability and transient information about the system that is used to design a desirable controller [50]. Nyquist criterion is used to determine if a system is stable by looking at the magnitude of the frequency response. Increasing the phase margin reduces the percent of overshoot of the response, decreasing the bandwidth increases the speed of the response, and the steady-state error is improved by increasing the low-frequency magnitude responses.

Another classical method is the *state-space design*, in which the desired system's pole locations are specified and then a controller consisting of state-variable feedback gains is designed to meet these requirements. Controller design consists of feeding back the state variables to the input of the system through specified gains that were found by matching the coefficients of the system's characteristic polynomial with the coefficients of the desired characteristic polynomial. If the state variables are not available, an observer is designed to emulate the plant and provide estimated state variables that will be used to obtain the gains.

Today systems operate in wider regimes than those in which they were originally designed and therefore the controllers need to be much more robust to be able to operate

beyond the design envelope. Classic control techniques lack the robustness that is necessary to approach the extreme situations that define problems like the RFC problem that has motivated this study. These types of beyond-the-envelope systems use less classic approaches to solve the highly nonlinear problems.

2.2.2. Nonlinear Optimal Control Methods. This section describes some of the tools used in nonlinear control. Khalil [51] lists in his *Nonlinear Systems* textbook various common tools used for nonlinear design, such as *linearization*, *integral control*, *gain scheduling*, *feedback linearization*, *sliding mode control*, *Lyapunov redesign*, *backstepping*, *passivity-based control*, and *high gain observers*. A brief description of each one of the methods without getting into the details of the formulation is provided by [51] and summarized below.

In the *design via linearization*, the controller is guaranteed to work over the neighborhood of the single operating point that was used for the linearization. This limitation is extended to a wider range of operating points with the *gain scheduling* method, by parameterizing several operating points by one or more variables. The system is then linearized at the chosen points, and linear feedback controllers are designed and implemented at each point. This creates a series of linear controllers that are activated by monitoring the scheduling variables and hence being able to operate at different points of the envelope. This is one of the most commonly used design tools in the aviation industry today, due to the simplicity of the design and its capability to work at different operating points.

The *integral control* approach ensures asymptotic regulation under all parameters that do not destroy the stability of the closed-loop. The integral action is introduced by integrating the regulation error between the measured and desired states. By regulating the integrated error to be zero at equilibrium, the feedback controller creates an asymptotically stable equilibrium point.

Feedback linearization is one of the most widely used methods when trying to control nonlinear systems by taking a different perspective to linearization of the systems. The idea behind the feedback linearization problem consists in the stabilization of the nonlinear state equation into a controllable linear state equation by introducing terms in the controller to reduce or cancel the nonlinearities. Feedback linearization can be

divided in full-state linearization, where the state equation is completely linearized, and input-output linearization, where the input-output map is linearized, and the state equation may be only partially linearized.

In the *sliding mode control* approach, trajectories are forced to reach a sliding manifold in finite time and to stay on the manifold for all future time. Motion on the manifold is independent of matching uncertainties. By using a lower order model, the sliding manifold is designed to achieve the control objective. The *Lyapunov redesign* uses a Lyapunov-like function of a nominal system to design an additional control component that makes the design robust to large matched uncertainties. Both the *sliding mode control* and the *Lyapunov redesign* produce discontinuous controllers, which could suffer from chattering in the presence of delays or unmodeled high-frequency dynamics.

Backstepping is a recursive procedure that interlaces the choice of Lyapunov function with the *design feedback control*. It breaks a design problem for the full system into a sequence of design problems for low order subsystems, using this extra flexibility between the lower order and scalar subsystems to solve stabilization, tracking and robust control problems under less restrictive conditions.

Passivity based controllers exploit passivity of the open-loop system in the design of feedback control by damping injection. High-gain observers consider the fact that state feedback might not be available in many practical problems and extends previous control techniques to output feedback.

These are some of the most important nonlinear methods that are available in the academic literature, but many other methods have emerged through the years by merging the best parts and pieces of the above methods described with the power of NN, fuzzy logic and GA, yielding very powerful methods that are able to solve some of the more complex nonlinear problems. Some of these methods will be introduced along with a mention of authors for reference to the reader in section 2.2.3.

2.2.3. Nonlinear Control and Neural Networks. As mentioned in section 1.3.4, NN have gained a lot of attention in the field of control over the last ten years. Optimal control formulations often lead to two point boundary value problems [47]. For this reason, except for a very special class of problems, like LQR problems, it is quite difficult to solve for the controller in state feedback form. Moreover for nonlinear

problems, the solution depends on the initial conditions. In real-life problems, however, it is difficult to predict the initial conditions a priori. Hence, it is necessary to obtain control functions that apply to an entire range of initial conditions to retain the feedback nature of the solution. The method of dynamic programming handles this problem by producing a family of optimal paths, or what is known as the *field of extremals* [47]. One great drawback of the dynamic programming approach, however, is that it requires a prohibitive amount of computation and storage in producing this entire *field of extremals* [47]. NN provides a solution to the problem of covering the entire *field of extremals*. This section intends to introduce some of the work done in the area of NN towards solving the highly demanded nonlinear control problems.

NN have been used extensively in the control of lumped parameter systems, which includes control of nonlinear plants. Various studies have realized neural network assisted controllers based on feedback linearization, dynamic inversion, reinforcement learning etc., in many fields like robotics, flight vehicles, chemical processes, motors, automobiles etc. A survey paper [52] is cited for reference. It should be noted, however, that so far almost all of the neural network applications on control systems reported in literature have been confined to lumped parameter systems. The work conducted in this thesis presents the powerful adaptive-critic methodology for a systematic synthesis of optimal neuro-controllers for distributed parameter systems, in a much desirable state feedback form.

The neural network approach presented in this paper has many resemblances to the Adaptive Critic Neural Network (ACNN) method presented in Balakrishnan and Biega [53] and Balakrishnan and Saini [54]. Balakrishnan and Viegga focus on the use of the ACNN architecture to obtain an optimal neurocontroller based in the dual network architecture formed by an action neural network (ANN) and a critic neural network (CNN). The ANN maps the states of a system to the control, while the second network, the CNN, captures the mapping between the states of a dynamical system and the co-states that arise in an optimal control problem. The equations that satisfy the optimality of the problem are solved with the help of NN. This makes it possible to synthesize the closed loop controllers for this complex process. It also allows the philosophy of dynamic programming to be carried out without the need for near impossible

computation and storage requirements. Another advantage of this neural network approach include the fact that no *a priori* assumptions about the form of the feedback control are needed; *i.e.*, one need not assume the control expressed in any particular form. The consequence of this off-line computational method is that the resulting control is available to be used as on-line state-feedback control for an entire envelope of initial conditions. A more detailed explanation of the adaptive and critic methods used in this thesis will be described in section 3.

Balakrishnan and Saini [54] use the ACNN architecture to design a controller for auto landing an aircraft. Balakrishnan and Han [55] extend the ACNN formulation to solve a terminal constraint optimal control problem using an expanded form of the ACNN architecture where the optimization goal is for a trajectory in minimum time to reach a set of final state constraints. The approach taken for the terminal constrain problem is to reformulate the state and optimal control equations to change the independent variable to that of one of the former states, generating a fixed final condition with respect to the independent variable. This sets a hard constraint on the Hamiltonian equations so that the final conditions are met exactly though the one-dimensional state equation which is no longer invariant to the independent variable. Thus a series of ACNN pairs are used in sequence along the trajectory to account for the variance.

Plummer [56] touches a family of terminal control problems in which he extends one of the most popular training algorithms for feed forward NN, backpropagation-through-time, to address the limitation that the feedforward NN algorithms have when dealing with the family of problems in which the cost function includes the elapsed trajectory-time. He approaches these limitations by reforming the controller design as a constraint optimization problem defined over the entire *field of extremals* for which the set of trajectory-times is incorporated into the cost which correspond to standard backpropagation-through-time with the addition of certain transversality conditions. The new gradient algorithm based on these conditions, called time-optimal backpropagation-through-time, is tested on two benchmark minimum-time control problems.

Jagannathan and Lewis [57] introduced a family of novel multilayer discrete-time neural-net controllers for the control of a class of multi-input multi-output dynamical

systems. The neural net controller includes modified delta rule weight tuning and exhibits an on-line learning instead of an off-line, so that control is immediate with no explicit learning phase needed. The structure of the neural network controller is derived using a filtered error/passivity approach in which the linearity in the parameters is not required and certainty equivalence is not used, hence overcoming several limitations of standard adaptive control. The stability analysis of the neurocontroller is done using the Lyapunov's direct method to guarantee the performance and the stability of the weight tuning algorithms of the neural nets. They make use of the passivity based controller properties described above despite the original system having not passivity properties, by using the neurocontroller to make the closed-loop system passive. This allows that the additional unknown bounded disturbances do not destroy the stability and tracking performance of the system.

Calise and Kim [58] demonstrated the power of the neural network within the realm of nonlinear control systems, with specific focus on aircraft control. The strength of their design lays in the implementation of feedback linearization along with NN as an alternative to gain scheduling, which simplifies the problem of designing complex flight control system for high-performance fighter aircrafts. Their design consists of a command and stability augmentation control system based on the feedback linearization, that uses an off-line trained network to invert the nonlinearities, while an online trained neural network is used to compensate for imperfections in the inversion and changes to the original dynamics and/or failures in the controls surfaces. A stable weight adjustment rule for the weights of the on-line neural network is also presented using a Lyapunov-like function.

Calise's effort to demonstrate the power of merging nonlinear control theory with the NN ability to model nonlinearities has yielded an extensive series of papers for a wide range of problems, the aerospace realm being the one with the most contributions. From helicopters to reusable launch vehicles, Calise and many more other authors have dedicated an incredible amount of work and resources to design neurocontrollers that would be able to approach the reconfigurable in an innovative and efficient approach. Some of these novel works are introduced below.

Calise, Pei and Prasad [59] show, in an actual flight system of an unmanned helicopter, the potential benefits of neural network direct adaptive control by designing an outer loop trajectory-tracking controller. Calise, Johnson and Rysdyk [60], use the X-33 Reusable Launch Vehicle technology demonstrator model to demonstrate a version of Calise's neurocontroller. The specific adaptive control method, called Pseudo-Control Hedging, is based in the concept of modifying a reference model to prevent an adaptation law from adapting to saturation of the vehicle input characteristic such as actuator position limits, actuator position rate limits and linear input dynamics. The same methodology is applied by Calise and Johnson [61] to a type of failures that led to a reduction in total control authority of the X-33 model. They accomplish this by preventing the outer loop dynamics to adapt to the inner-loop dynamics while operating at the control limits. Calise, Lee, and Sharma [62] [63] show the approach taken to the RFC problem using a model of a tailless fighter aircraft configured with multiple and redundant control actuation devices, which is later tested in both a piloted simulation and in flight test on the X-36 aircraft.

Haley and Soloway [64] propose a Neural Generalized Predictive Control (NGPC) algorithm capable of real-time control law reconfiguration, model adaptation, and the ability to identify failures in control effectiveness by using an innovative user define cost function that can be associated to either the aircraft outputs or to the control inputs. The NGPC algorithm operates in two modes, prediction mode, in which uses the aircraft model to predict the aircraft's response, and control mode in which the control input that minimized the user specified function is passed to the aircraft as actuator position commands which then produce the desired aircraft response. When failure simulations are introduced, such as frozen elevator, the NGPC algorithm learns the changed dynamics and reconfigures to use alternative controllers, like symmetric ailerons to stabilize the aircraft.

Bull, Kaneshige, and Totah [65] introduce an innovative generic neural flight control and autopilots system to provide adaptive flight control, without requiring extensive gain-scheduling or explicit system identification. The autopilot system is applied to a wide range of vehicle systems and is formed by a generic autopilot, a neural flight controller and a mode control panel, and a flight director. The generic guidance

system performs automatic gain-scheduling using frequency separation, based upon the neural flight control system's specified reference model. The neural flight control architecture is based on the augmented model inversion controller developed by Calise and Rysdyk [66], which is a direct adaptive tracking controller that integrates feedback linearization theory with both pre-trained and on-line learning NN. Pre-trained NN provide estimates of the aerodynamics stability and control characteristics required for model inversion. The on-line learning NN are used to generate command augmentation signals to compensate for the errors in the estimates and from the model inversion. The online learning NN also provide additional potential for adapting to changes in aircraft dynamics due to damage or failure. The mode control panel is the pilots' interface with the generic autopilot, and the flight directors, provide guidance commands to the pilot through the graphical display of pitch and bank errors. The generic neural flight control is tested in the NASA Advanced Concepts Flight Simulator (ACFS) [67].

Reference [68] describes an intelligent fault tolerant flight control system that blends aerodynamic and propulsion actuation for safe flight operation in the presence of actuators failures. Fault tolerance is obtained by a nonlinear adaptive control strategy based on on-line learning NN and actuator relocation scheme. The adaptive control block incorporates a recently developed technique for adaptation in the presence of actuator saturation, rate limits and failure. The proposed integrated aerodynamic/propulsion flight control system is evaluated in a nonlinear flight simulation.

Kim and Lee [69] propose a nonlinear flight control system using back-stepping and a NN controller that is tested in a non-linear six-degree-of-freedom simulation for an F-16 aircraft. The back-stepping controller is used to stabilize all state variables simultaneously without separating the fast dynamics from the slow dynamics, while the adaptive NN controller is used to compensate for the effect of the aerodynamic modeling errors, by assuming that the aerodynamic coefficients include uncertainty. The Lyapunov stability theorem is used to demonstrate that the tracking errors and weights of NN exponentially converge to a compact set under mild assumptions on the aerodynamic uncertainties and nonlinearities.

Ferrari and Stengel [70] take the approach of designing a nonlinear control system that takes advantage of priori knowledge and experience gained from linear controllers,

while capitalizing in the broader capabilities of adaptive, nonlinear control theory and computational NN. The importance of this novel approach lies in the fact that the gradients of the nonlinear control law represents the gain matrices of the equivalent locally linearized controllers by using a set of hypersurfaces expressed as NN that represent satisfactory linear controllers designed over the plant's operating range.

Other areas like fuzzy logic and GA have also shown good results in the areas of mobile robot navigation, and for reference, only a few papers are described in this section. The author had the privilege to recently attend the 15th Triennial World Congress of the International Federation of Automatic Control (IFAC) that was held in July of 2002 in Barcelona, Spain, and came across the work done by some researchers in the fields of fuzzy logic and GA that were successfully applied in actual physical systems. Despite the fact that these fuzzy logic and GA models did not seek to produce optimal solutions to the problems they were applied, they were implemented in real models, and were able, in general, to deal with the uncertainties that were present in the environments they operated.

Hartana and Sasiadek [71] present a sensor fusion for dead-reckoning mobile robot navigation. Odometry and sonar measurement signals are fused together using extended Kalman filter (EKF) and Adaptive Fuzzy Logic System (AFLS). Two different methods are used to adapt EKF, the first uses two exponential data weighting functions to estimate the process and white noise covariance, while the second method only uses the white noise covariance. The paper shows that the fused signal of odometry and sonar measurements along with the EKF and the AFLS is more accurate than any of the original signals considered separately, and the enhanced, more accurate signal, is used to successfully guide and navigate the robot.

Green and Sasiadek [72] show the comparison results for tracking of a square trajectory by a two-link flexible robot manipulator, using as comparison an inverse dynamics control (IDC) and fuzzy logic control (FLC). A repetitive control technique is used to train a robot on the premise that it must execute periodic motions so that its performance improves after each iteration. The results show that while the repetitive learning inverse dynamics control (RLIDC) achieves no improvement in tracking, repetitive learning fuzzy logic control (RLFLC) achieves greater precision where cyclic

tracking enables the fuzzy inference system to self-adapt and further reduce tracking errors.

Cuesta and Ollero [73] demonstrate the properties of a fuzzy reactive navigation, which is applied to a nonholonomic mobile robot ROMEO-3R. Despite the fact that reactive navigation approaches usually present worse dynamic performance, since oscillations are very common in these techniques, they are an emerging alternative to (or in combination) planned schemes. The influence of the own reactive navigation parameters (such as sensor range, computation time, control gains,...) in the stability of the system is analyzed, and stability conditions in terms of these parameters are also introduced.

Martinez-de Dios and Ollero [74] present a study for a general infrared vision system to be used in robotic applications in natural outdoor environments. In these applications the robustness of the vision system and the automatic settings of the infrared cameras are very important issues. A piecewise linear model of the infrared camera is used for the design and development of a fuzzy control method by applying visual feedback techniques. The vision system included a new fuzzy-multiresolution threshold computation method, which considered knowledge of the application and information on the illumination conditions to select an appropriate threshold for the segmentation of the object of interest and described some experiments for surveillance.

Anibal Ollero is Professor at the *Ingeniería de Sistemas y Automática Department*, University of Seville, Spain. He has participated and has led more than 40 research and development projects on robotics, intelligent control, computer vision and artificial intelligence, funded by Spanish agencies, regional agencies, the European Community (ESPRIT, Telematics Applications, Environment and Climate y CRAFT-BRITE) and several industries. Some of these projects can be seen at his World-Wide-Web home page [75], and range from perception techniques for detection and tracking using infrared and visual cameras, distributed environmental disaster information and control systems, systems for estimation of the biomass of specimens in fish farms using computer vision and others techniques, forest fire monitoring using perception systems, automatic steering of large vehicles, intelligent control systems for mobile robots for greenhouse servicing, systems for automatic functional testing of airplanes and

functional, automation tests in the aeronautic industry, or research and development for planetary Rovers to name few.

Armingol, de la Escalera, Mata and Salichs [76] describe a vision-based landmark learning and recognition system for use in mobile robot navigation tasks. The system uses GA for both learning and recognition processes. The system is able to learn new landmarks with very little human intervention. The recognition system can read text inside landmarks, when present. This learning ability is tested with two very different landmarks that have been successfully used for indoor topological robot navigation. In addition, some new landmarks are learned, that will be tested for indoor-outdoor navigation in future works.

3. ADAPTIVE CRITIC NEURAL NETWORK DESIGN

This section provides a more detailed description and the training synthesis approach of the NN architecture used in this thesis.

3.1. NEUROCONTROLLER DEVELOPMENT.

The ACNN is a feed forward backpropagation architecture consisting of two separate networks, an Action Neural Network (ANN) and a Critic Neural Network (CNN). Each network has its own independent characteristics but at the same time their intrinsic relationship is a key point for obtaining the near optimal control law for a given system. The general structure of one layer neural network is shown in Figure 3.1.

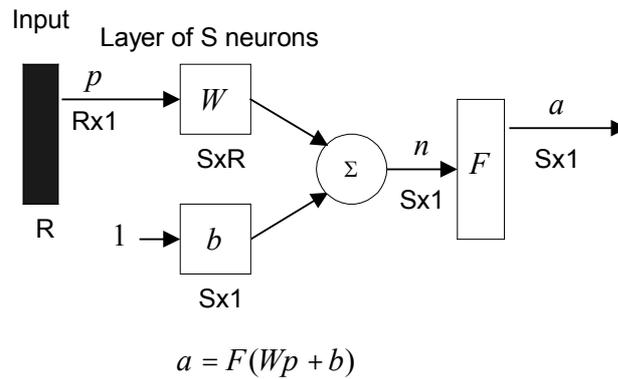


Figure 3.1. General Structure of One Layer Neural Network.

where p is the input to the NN, a is the output of each NN, W is the weight associated with the given layer, b is the bias associated with the weight-input combination, n represents the summation of the weight and the bias before being transformed by the activation function, and F is activation function or transfer function associated with the layer.

For this thesis, it has been chosen that both, the ANN and CNN architectures, consist of three layer NN. The three layer NN consists of two hidden layers with hyperbolic tangent sigmoid transfer function (HTS-TF), and an output layer with a linear transfer function. Considering the schematics defined in Figure 3.1, where a represent the output of the NN, the output of the HTS-TF is defined as

$$a = \frac{e^n - e^{-n}}{e^n + e^{-n}} \quad (25)$$

Figure 3.2 shows a schematic of the 3-layer NN architecture used in the thesis.

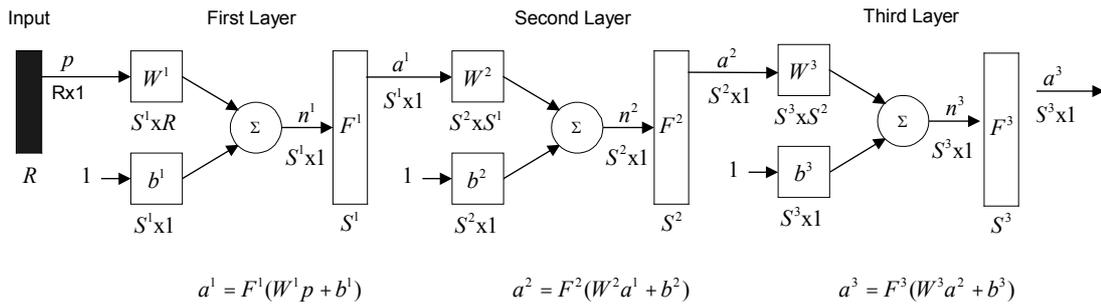


Figure 3.2. Three Layer Neural Network Architecture.

where the output for both ANN and CNN networks is defined as

$$a^3 = W^3 F^2(W^2 F^1(W^1 p + b^1) + b^2) + b^3 \quad (26)$$

It is really important to make sure that the dimensions of the corresponding weight matrices, for each of the layers, be consistent with the dimensions of the inputs and the desired outputs. The dimensions for both the ANN and CNN architecture are defined by the notation N_{p, S^1, S^2, S^3} , where the first subscript represent the dimension of the

inputs, P , the number of neurons in each one of the hidden layers layer is defined by S^1 and S^2 , and the dimension of the output is defined by S^3 . For example, a NN architecture of the form $N_{3,6,6,1}$, corresponds to 3 inputs, 6 neurons for the first and second hidden layers and 1 neuron corresponding to the single output.

The NN training procedures are implemented in the software package MATLAB[®] version 5.3.0 Release 11 during the first part of the analysis and finalized in version 6.0.0.88 Release 12 [77]. MATLAB[®] is a software package for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation that provides a convenient interface to built-in state-of-the-art subroutine libraries, and incorporates a high-level programming language. Figure 3.3 shows a screen shot of the command window interface for MATLAB[®] version 6.0.0.88 Release 12.

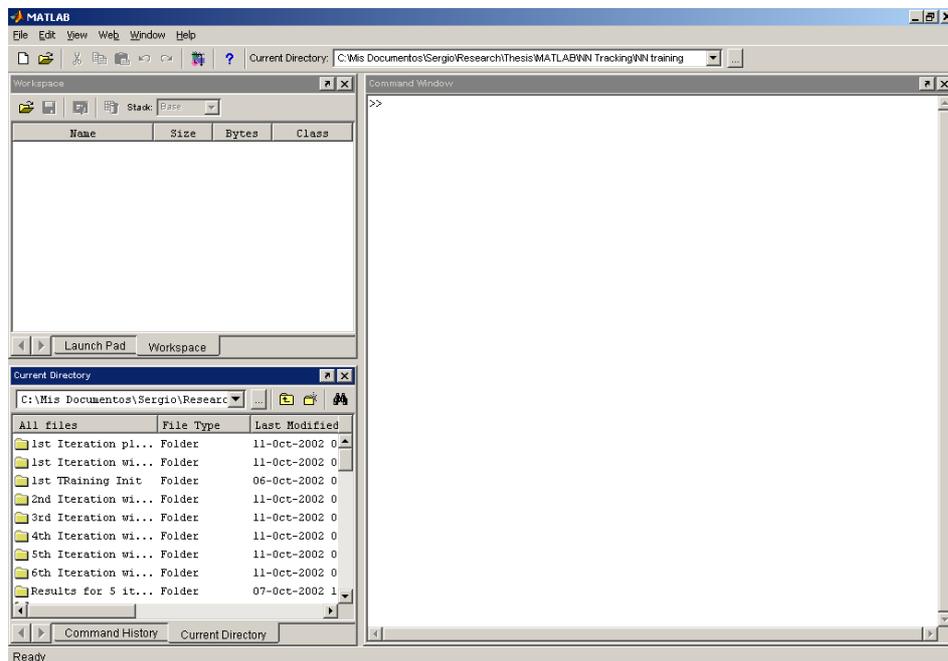


Figure 3.3. Screen Shot for the Command Window of the MATLAB[®] Release 12.

The different subroutine libraries, called M-files, are grouped in toolboxes, which provide different analytical tools for different applications. The Neural Network Toolbox is the one used to train the NN architectures. The feed-forward backpropagation network is created using the M-file “*newff.m*” which is defined as.

$$\text{net} = \text{newff}(\mathbf{P}_R, [S_1 \ S_2 \dots \ S_{N_i}], \{\text{TF}_1 \ \text{TF}_2 \dots \ \text{TF}_{N_i}\}, \mathbf{B}_{\text{TF}}, \mathbf{B}_{\text{LF}}, \mathbf{P}_F) \quad (27)$$

The M-file receives the information of the architecture of the NN in the following manner:

- \mathbf{P}_R is a $R \times 2$ matrix that defines the minimum and maximum values for R input elements.
- S_i determines the number of neurons for each one of the N_i layers.
- TF_i - Transfer function of i th layer.
- \mathbf{B}_{TF} defines the backpropagation network training function.
- \mathbf{B}_{LF} defines the backpropagation weight/bias learning function.
- \mathbf{P}_F defines the performance function used.

As mentioned above, the HTS-TF is chosen for the hidden layers with a linear transfer function for the output layer. At the beginning of the training procedure, several backpropagation training algorithms were considered:

- Gradient descent backpropagation.
- Gradient descent with momentum backpropagation.
- Gradient descent with adaptive linear backpropagation.
- Gradient descent with momentum and adaptive linear backpropagation.
- Levenberg-Marquardt backpropagation.

The Levenberg-Marquardt backpropagation algorithm being the one that produced the best results and being the fastest backpropagation algorithm although with

the limitation that it requires lots of computing memory. A better description of this training algorithm is provided in the MATLAB[®] Neural Network Toolbox. [77].

Levenberg-Marquardt algorithm was designed to approach second-order training speed without having to compute the Hessian matrix. When the performance function has the form of a sum of squares (as is typical in training feed-forward networks), then the Hessian matrix can be approximated as:

$$H = J^T J \quad (28)$$

and the gradient can be computed as

$$g = J^T e \quad (29)$$

where J is the Jacobian matrix that contains first derivatives of the network errors with respect to the weights and biases, and e is a vector of network errors. The Jacobian matrix can be computed through a standard backpropagation technique that is much less complex than computing the Hessian matrix. The Levenberg-Marquardt algorithm uses this approximation to the Hessian matrix in the following Newton-like update:

$$X_{k+1} = X_k - [J^T J + \mu I]^{-1} J^T e \quad (30)$$

When the scalar μ is zero, this is just Newton's method, using the approximate Hessian matrix. When μ is large, this becomes gradient descent with a small step size. Newton's method is faster and more accurate near an error minimum, so the aim is to shift towards Newton's method as quickly as possible. Thus, μ is decreased after each successful step (reduction in performance function) and is increased only when a tentative step would increase the performance function. In

this way, the performance function will always be reduced at each iteration of the algorithm.

This algorithm is used to update the weights during the training process in which data corresponding to the range of interest of the states is feed to the NN. The training algorithm has the property that can train any network as long as its weight, net input, and transfer functions have derivative functions. This is ensured since the activation function used in this work, Eq. (25), maps the data introduced into the vector n into a smooth and continuous function with hyperbolic tangent profile that smoothly squeezes the data in the ± 1 range.

3.2. TRAINING SYNTHESIS OF THE NEUROCONTROLLER.

Both networks, the ANN and the CNN, are originally initialized with a stable solution, which will be described in more detail in section 4.2.3.3. The inputs for the training of both NN, consist of randomized sets of data that enclose the entire domain of interest. Target training outputs for both ANN and CNN are obtained using Eqs. (9) and (11) for the optimal control problem and Eqs. (18) and (19) for the tracking formulation. For simplicity and to avoid confusion in the notation, when noting the equations used to obtain the training target values in the steps described bellow, the formulation for the optimal problem in section 2.1.2 will be used. The reader can replace the equations by the corresponding optimal tracking problem described in section 2.1.4 for the training synthesis of the NN for the optimal tracking problem.

3.2.1. Training Synthesis of the Action Neurocontroller. The steps followed for the training of the Action NN are:

1. The initial CNN is *assumed* to be optimal.
2. The initial output u_i , is obtained by feeding random values of the states x_i , to the ANN.
3. The discrete-time nonlinear equations of motion, Eq. (2), are propagated forward to obtain x_{i+1} , using the states x_i and the output u_i of the ANN.

4. The Critic NN is feed the output form step-3, x_{i+1} , to calculate the Lagrange multiplier, λ_{i+1} , associated with x_{i+1} .
5. The ANN is then trained using x_i as input and the optimal control, u_i^* , as target, which is calculated using the output from step-4, λ_{i+1} , and Eq. (11).

Steps 1 through 5 are repeated until the desired level of accuracy for the ANN is achieved. The level of accuracy is reached when the norm of the errors between the current ANN outputs and the previous trained outputs is below a pre-established error ε . Figure 3.4 shows a schematic of the steps described above for the synthesis of the Action NN.

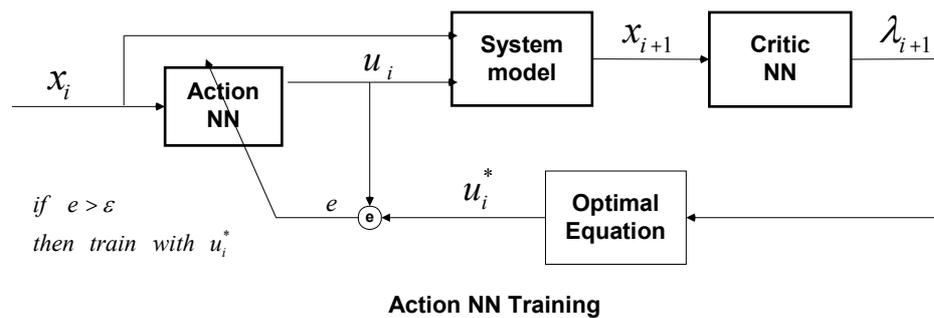


Figure 3.4. Training Synthesis of the Action Neural Network Controller.

3.2.2. Training Synthesis of the Critic Neural Network. The steps followed for the training of the Critic NN are:

1. The ANN is *assumed* to be optimal.
2. The initial output u_i , is obtained by feeding random values of the states x_i , to the ANN.

3. The discrete-time nonlinear equations of motion, Eq. (2), are propagated to get x_{i+1} . The Critic NN is feed this result to calculate Lagrange multiplier, λ_{i+1} , associated with x_{i+1} .
4. The discrete-time nonlinear equations of motion, Eq. (2), and the correspondent co-state differential equations, Eq. (9), are simultaneously solved backwards in time to obtain the target λ_i^* .
5. The CNN is then trained using x_i as input and λ_i^* from step-4 as target.

Steps 1 through 5 are repeated until the desired level of accuracy for the CNN is achieved. The level of accuracy is reached when the norm of the errors between the current CNN outputs and the previous trained outputs is below a certain pre-established error ε . Figure 3.5 shows a schematic of the steps described above for the synthesis of the Critic NN.

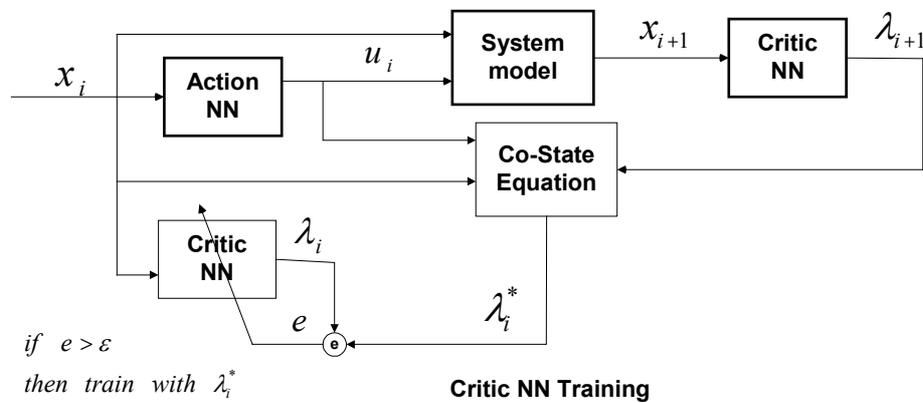


Figure 3.5. Training Synthesis of the Critic Neural Network Controller.

Step 5 in the CNN training marks the end of one training cycle for both the ANN and CNN. Training cycles are continued until there is no acceptable change in the

outputs of the ANN and CNN relative to the previous training cycle. Again the level of accuracy required to terminate the training cycles is reached when the 2-norm of the errors between the current NN outputs and the previous cycle outputs is below a pre-established error ε . At this point the output u_i^* of the Action NN is considered to be optimal. Note that the assumption done in step-1, for the synthesis of both ANN and CNN training, is necessary in order to achieve the desired convergence of the NN. Although for the first iteration, the NN are initialized with a stable solution, after the first training synthesis iteration, each NN that is being trained, i.e. ANN training synthesis, is suboptimal relative to the NN assumed to be optimal in step 1, i.e. the ANN is suboptimal relative to the CNN during the ANN training synthesis.

4. PROBLEM DEFINITION

4.1. AIRCRAFT MODEL.

4.1.1. Reference Coordinate System. In order to better understand the model that will be used in the formulation for the nonlinear neurocontroller, which will be introduced in section 4.2, it is necessary to define the equations that govern the motion of a rigid airplane. Roskam's [78] formulation and definitions will be used, throughout the remainder of this section, to define the airplane's equations of motion and the reference system where these equations are valid. Figure 4.1 shows the two systems used to define the equations that govern the motion of a rigid airplane, the Earth-fixed system and the airplane body-fixed system. The Earth-fixed system is denoted by $X'Y'Z'$, which will be considered the inertial reference frame in which the Newton's laws of motion are valid. This model reference neglects rotational velocity of the Earth. The airplane body fixed system is defined by XYZ .

The XYZ airplane body fixed system is fixed relative to the airplane, where the positive X axis is along the fuselage, the positive Y axis is along the starboard (right) wing, and the positive Z axis is directed downward, perpendicular to the XY plane as shown by the directions of the arrows in Figure 4.1. The origin is located at the geometric center of gravity. The translational motion of the airplane is given by the velocity components of the velocity: forward velocity (U), *side-slip velocity* (V), and downward velocity (W) which are directed along the X , Y , and Z directions respectively. The free stream velocity, V_∞ , represents the vector sum resultant of the velocity component, U , V , and W . The rotational motion is given by the angular velocity components: roll rate (P), pitch rate (Q), and yaw rate (R), about the X , Y , and Z axes respectively. These rotational velocities are due to the moments about the airplane body-fixed system: roll moment (L), pitch moment (M), and yaw moment (N) about the X , Y , and Z axes, respectively. Section 4.1.2 will define the non-linear six-degrees-of-freedom (6-DOF) equations of motion for the general case.

Figure 4.1 also shows that the airplane is assumed to consist of continuum mass elements, dm , that are kept track by a series of vectors, \vec{r}' , which connect the origin $X'Y'Z'$ with each mass element. Each mass element is subject to the acceleration of

gravity, \vec{g} , which is assumed to be oriented along the positive Z' -axis of the Earth-fixed coordinate system, thus assuming that the Earth is flat. This creates a gravitational force acting in each element mass equal to $\rho_A \vec{g} dv = \vec{g} dm$, where ρ_A represents the local mass density of the airplane and dv is an airplane volume element. The elements that are located in the surface of the airplane are also subject to combined aerodynamic and thrust forces per unit area denoted by \vec{F} . These aerodynamic and thrust-combined forces will be expanded in the next section.

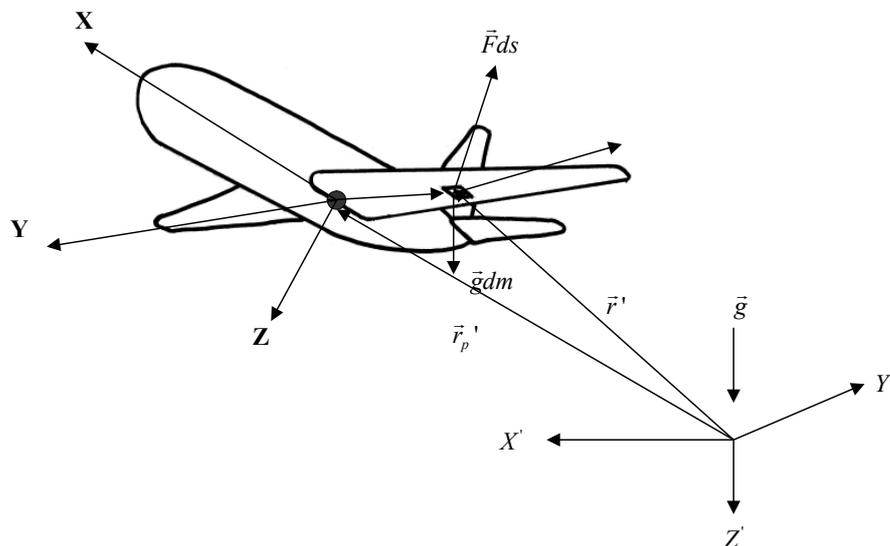


Figure 4.1. Definition of the Earth-Fixed and Body-Fixed Coordinate Systems.

4.1.2. Non-linear Six-Degrees-Of-Freedom Model. By applying the Newton's second law to Figure 4.1, such as the linear and angular momentum are equal to the externally applied forces and moments respectively, the results are the creation of the vector-integral form of the equations of motion for the linear and angular momentum defined as Eqs. (31) and (32) respectively:

$$\frac{d}{dt} \int_v \rho_A \frac{d\vec{r}'}{dt} dv = \int_v \rho_A \vec{g} dv + \int_S \vec{F} ds \quad (31)$$

$$\frac{d}{dt} \int_v \vec{r}' \times \rho_A \frac{d\vec{r}'}{dt} dv = \int_v \vec{r}' \times \rho_A \vec{g} dv + \int_S \vec{r}' \times \vec{F} ds \quad (32)$$

The left hand side of Eqs. (31) and (32) represent the linear and angular momentum respectively while the right hand side represents the applied forces and moments respectively. After expanding these equations, see reference [78] for more details, the results yield the force and moment equations in the airplane body-fixed axis system XYZ:

$$m(\dot{U} - VR + WQ) = -mg' \sin \Theta + F_{A_x} + F_{T_x} \quad (33)$$

$$m(\dot{V} + UR - WP) = mg \sin \Phi \cos \Theta + F_{A_y} + F_{T_y} \quad (34)$$

$$m(\dot{W} - UQ + VP) = mg \cos \Phi \cos \Theta + F_{A_z} + F_{T_z} \quad (35)$$

$$I_{xx} \dot{P} - I_{xz} \dot{R} - I_{xz} PQ + (I_{zz} - I_{yy})RQ = L_A + L_T \quad (36)$$

$$I_{yy} \dot{Q} + (I_{xx} - I_{zz})PR + I_{xz}(P^2 - R^2) = M_A + M_T \quad (37)$$

$$I_{zz} \dot{R} - I_{xz} \dot{P} + (I_{yy} - I_{xx})PQ + I_{xz}QR = N_A + N_T \quad (38)$$

where I_{xx} , I_{yy} , I_{zz} and I_{xz} represent the moments of inertia, and m is the mass. Note that in the right hand side of Eq. (33)-(38), the applied forces and moments have different subscripts depending if they correspond to the aerodynamic or thrust components respectively, i.e. F_{A_x}, F_{T_x} , while it can also be seen that the aerodynamic and thrust forces in Eqs. (33)-(35) also present a subscript that describes in which axes the force is exerted, i.e. F_{A_x}, F_{T_x} for the forces acting along the X-axis in Eq. (33). The orientation of the airplane relative to the Earth-fixed coordinate system X'Y'Z', is obtained by introducing three sequential rotations over the Euler angles: heading angle (Ψ), the pitch attitude angle (Θ), and the bank or roll angle (Φ).

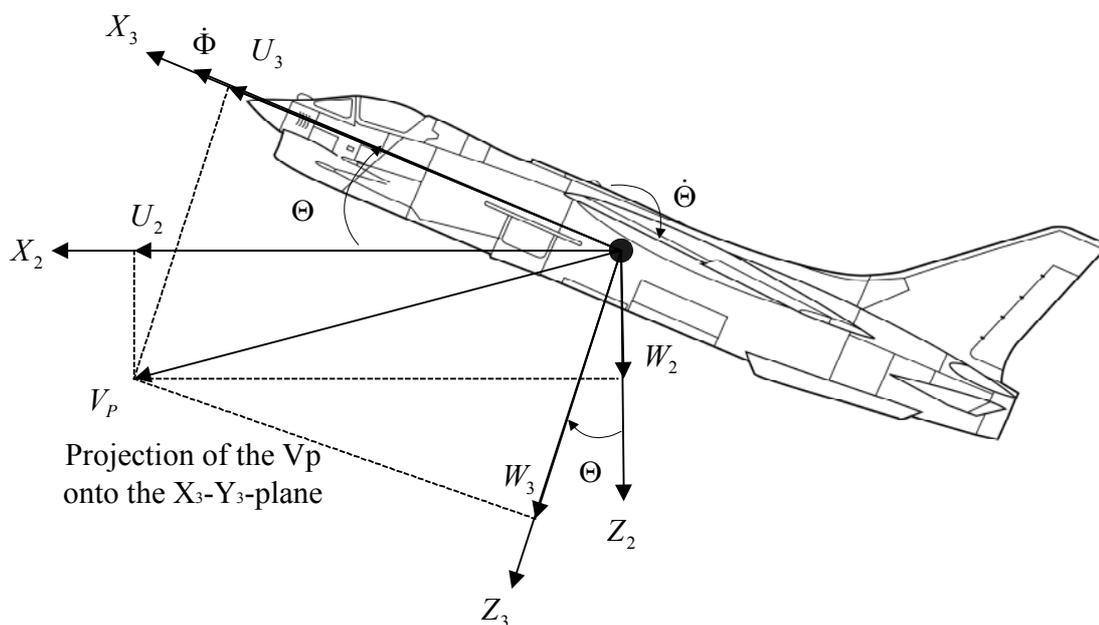


Figure 4.3. Rotation Over a Pitch Angle of Θ About Y_2 ^[79].

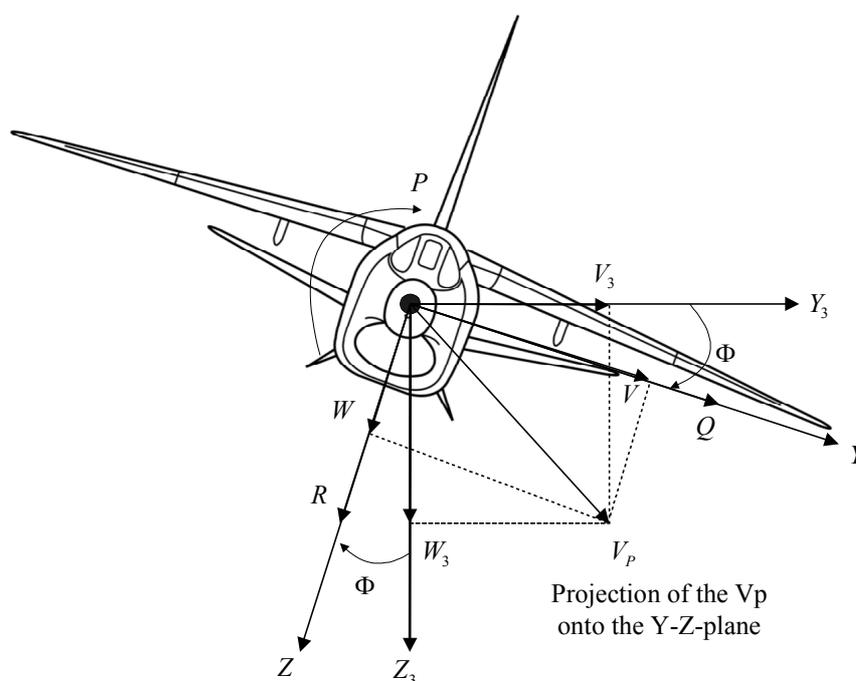


Figure 4.4. Rotation Over a Bank Angle of Φ About X_3 ^[79].

In return, these rotations generate the body-fixed axis kinematic equations [78]:

$$\dot{\Phi} = P + \tan \Theta (Q \sin \Phi + R \cos \Phi) \quad (39)$$

$$\dot{\Theta} = Q \cos \Phi - R \sin \Phi \quad (40)$$

$$\dot{\Psi} = \frac{Q \sin \Phi + R \cos \Phi}{\cos \Theta} \quad (41)$$

The navigation equations that determine the location of the aircraft at any given time are given by [80]

$$\dot{p}_N = U \cos \Theta \cos \Psi + V(-\cos \Phi \sin \Psi + \sin \Phi \sin \Theta \cos \Psi) + W(\sin \Phi \sin \Psi + \cos \Phi \sin \Theta \cos \Psi) \quad (42)$$

$$\dot{p}_E = U \cos \Theta \sin \Psi + V(\cos \Phi \cos \Psi + \sin \Phi \sin \Theta \sin \Psi) + W(-\sin \Phi \cos \Psi + \cos \Phi \sin \Theta \sin \Psi) \quad (43)$$

$$\dot{h} = U \sin \Theta - V \sin \Phi \cos \Theta - W \cos \Phi \cos \Theta \quad (44)$$

where \dot{p}_N , \dot{p}_E and \dot{h} represent the north, east, and vertical components of the aircraft velocity in the locally level geographic frame on the surface of the Earth. One more change of coordinate system is necessary to be able to reduce the complexity of the mathematical models for the aerodynamic forces and moments and the thrust forces and moments. This new coordinate system is called the stability axis system, defined by $X_S Y_S Z_S$, and introduces the aerodynamic angle of attack angle, α , and the aerodynamic sideslip angle, β . Figure 4.5 shows the stability axis system defined for an airplane in a steady state, wing level, straight line flight for zero initial sideslip angle, where α_1 is the steady state angle-of-attack, γ_1 is the steady state flight-path-angle, and θ_1 is the steady state pitch-attitude-angle.

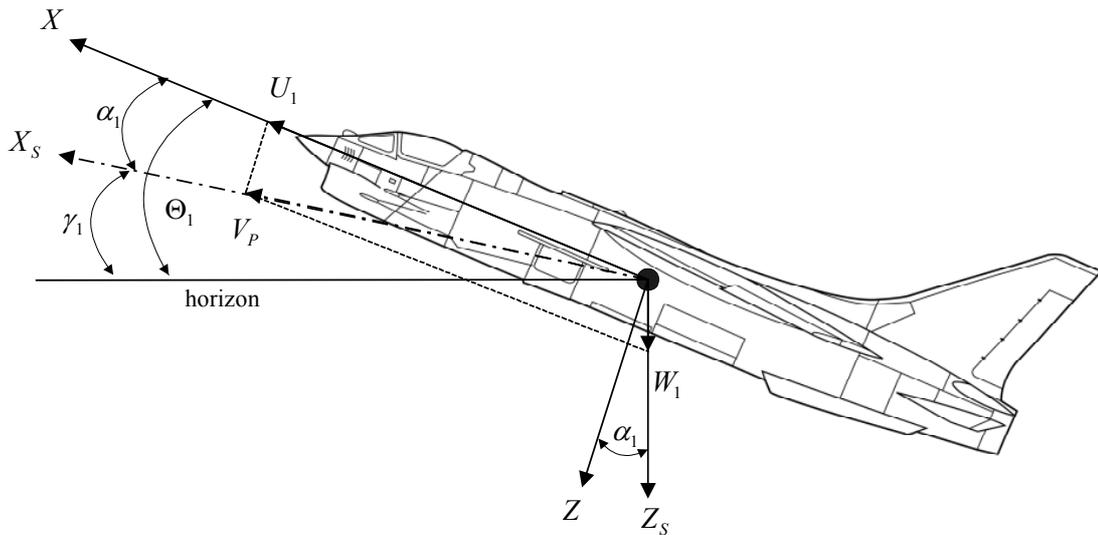


Figure 4.5. Definition of the Stability Axis System for Zero Sideslip^[79].

For the zero sideslip angle case, the flight path angle can be expressed as in Eq. (63)

$$\Theta_1 = \gamma_1 + \alpha_1 \quad (45)$$

Figure 4.6 shows the stability axis system defined for the case in which the sideslip angle is not zero, where β_l is the steady state sideslip-angle, and V_P is the true airspeed velocity. The aerodynamic angle of attack and sideslip angles are related to the velocity components through

$$\beta_1 = \arcsin \frac{V_1}{V_{P_1}} \approx \frac{V_1}{U_1} \quad (46)$$

$$\alpha_1 = \arcsin \frac{W_1}{V_{P_1} \cos \beta} \approx \frac{W_1}{U_1} \quad (47)$$

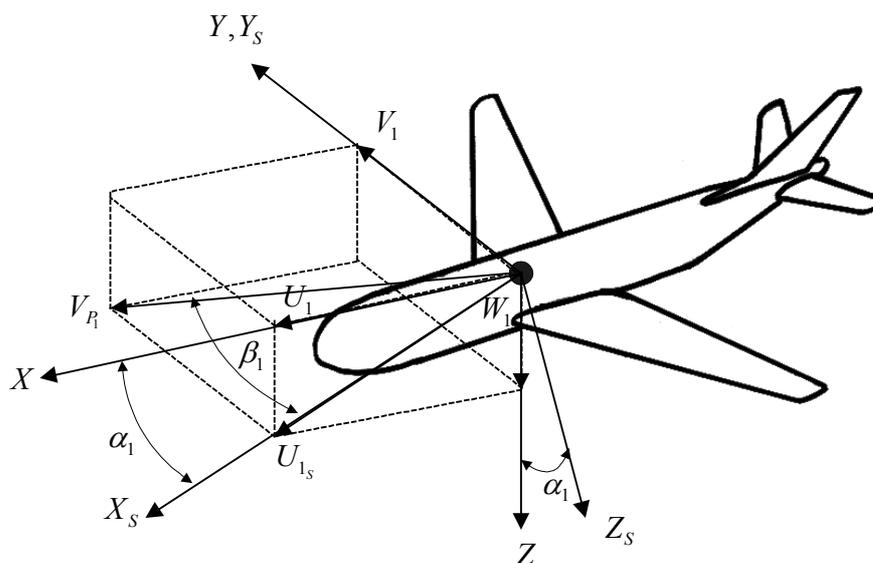


Figure 4.6. Definition of the Stability Axis System for the Case of Nonzero Sideslip Angle.

4.1.3. Airplane Controls. There are three basic types of controls on an airplane that are designed to change and control the moments about the XYZ axis. These control surfaces are the ailerons, the elevators, and the rudder, which can be deflected at the command of the pilot, or the control system, trying to change the moments about the XYZ system. The ailerons are generally mounted at the trailing edge of the wing, near the wingtips, while the elevators are generally located on the horizontal tail with the variation that in some modern aircraft the complete horizontal stabilizer is rotated instead of having an elevator. The rudder is located at the trailing edge on the vertical tail. Control in an airplane is achieved by deflection of the control surfaces such that can effectively change the aerodynamic forces on the surface, i.e. if there is a downward deflection of the elevator, this will increase the lift of horizontal tail, which will in return create a pitch down moment about the center of gravity of the airplane. The opposite happens when the deflection is upwards for the case of the wing or the tail. The increase or decrease in aerodynamic lift is beyond the scope of this thesis and some references can be look at [80], [81], [82] and [83] to name few ones. Figure 4.7 shows a description of

the three control surfaces available for the F-8 Crusader used in this thesis, and it can be seen that horizontal tail that rotates completely.

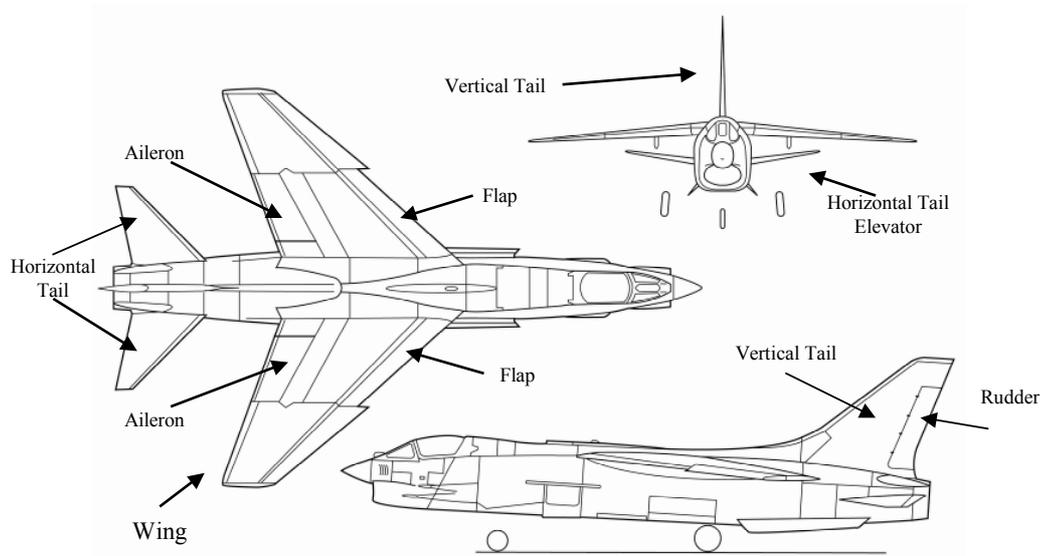


Figure 4.7. General Definition of the Airplane Control Surfaces ^[79].

4.1.4. Perturbed State Equations of Motion Model. In order to simplify the highly nonlinear equations of motion defined in Eqs. (33)-(41), two special flight conditions, are considered in more detail:

- Steady state flight condition.
- Perturbed state flight condition.

Only the second flight condition will be discussed in this thesis, yielding the equations that form the basis for the airplane model that will be used this study. Although the perturbed state equations of motion yield a linearized model, this study will use a modification of such model that is highly nonlinear in the angle of attack and will be further discussed in section 4.2.1. Roskam [78] defines the perturbed state flight as:

“A perturbed state flight condition is defined as one for which ALL motion variables are defined relative to a known steady state flight condition“

For that case, according to Roskam [78], the substitutions are applied to all motion variables, forces and moments in the original Eqs. of motion (33)-(41). For example, the forward velocity state, U , uses the substitution $U = U_1 + u$, where the subscript in U_1 defines the steady state flight condition, and the lower case variable, u , defines the perturbed state flight condition. Similar substitutions are conducted for the rest of the states. Table 4.1 shows the associated substitutions between the states and the perturbed and steady state flight conditions.

Table 4.1. Associated Symbols for the States, Perturbed States and Steady States.

States Variables	State Variable Symbol	Perturbed State Variable Symbol	Steady State Variable Symbol
Forward Velocity	U	u	U_1
Side-slip Velocity	V	v	V_1
Downward Velocity	W	w	W_1
Roll Rate	P	p	P_1
Pitch Rate	Q	q	Q_1
Yaw Rate	R	r	R_1
Roll Angle	Φ	ϕ	Φ_1
Pitch Attitude Angle	Θ	θ	Θ_1
Heading Angle	Ψ	ψ	Ψ_1

The new perturbation equations of motion are defined in Eqs. (48)-(56)

$$m(\dot{u} - (V_1 + v)(R_1 + r) + (W_1 + w)(Q_1 + q)) = -mg \sin(\Theta_1 + \theta) + F_{A_{y1}} + f_{A_x} + F_{T_{y1}} + f_{T_x} \quad (48)$$

$$m(\dot{v} + (U_1 + u)(R_1 + r) - (W_1 + w)(P_1 + p)) = mg \sin(\Phi_1 + \phi) \cos(\Theta_1 + \theta) + F_{A_{y1}} + f_{A_y} + F_{T_{y1}} + f_{T_y} \quad (49)$$

$$m(\dot{w} - (U_1 + u)(Q_1 + q) + (V_1 + v)(P_1 + p)) = mg \cos(\Phi_1 + \phi) \cos(\Theta_1 + \theta) + F_{A_1} + f_{A_z} + F_{T_1} + f_{T_z} \quad (50)$$

$$I_{xx} \dot{p} - I_{xz} \dot{r} - I_{xz} (P_1 + p)(Q_1 + q) + (I_{zz} - I_{yy})(R_1 + r)(Q_1 + q) = L_{A_1} + l_A + L_{T_1} + l_T \quad (51)$$

$$I_{yy} \dot{q} + (I_{xx} - I_{zz})(P_1 + p)(R_1 + r) + I_{xz} [(P_1 + p)^2 - (R_1 + r)^2] = M_{A_1} + m_A + M_{T_1} + m_T \quad (52)$$

$$I_{zz} \dot{r} - I_{xz} \dot{p} + (I_{yy} - I_{xx})(P_1 + p)(Q_1 + q) + I_{xz} (Q_1 + q)(R_1 + r) = N_{A_1} + n_A + N_{T_1} + n_T \quad (53)$$

$$P_1 + p = (\dot{\Phi}_1 + \dot{\phi}) - (\dot{\Psi}_1 + \dot{\psi}) \sin(\Theta_1 + \theta) \quad (54)$$

$$Q_1 + q = (\dot{\Theta}_1 + \dot{\theta}) \cos(\Phi_1 + \phi) + (\dot{\Psi}_1 + \dot{\psi}) \cos(\Theta_1 + \theta) \sin(\Phi_1 + \phi) \quad (55)$$

$$R_1 + r = (\dot{\Psi}_1 + \dot{\psi}) \cos(\Theta_1 + \theta) \cos(\Phi_1 + \phi) - (\dot{\Theta}_1 + \dot{\theta}) \sin(\Phi_1 + \phi) \quad (56)$$

After some trigonometric manipulations and approximations, which include some restrictions to the allowable magnitude of the motion perturbations, see reference [78] for further analysis, Eqs (48)-(56) are simplified by eliminating the small perturbations and neglecting the nonlinear terms compared with the linear terms, thus yielding

$$m(\dot{u} - V_1 r - R_1 v + W_1 q + Q_1 w) = -mg\theta \cos \Theta_1 + f_{A_x} + f_{T_x} \quad (57)$$

$$m(\dot{v} + U_1 r + R_1 u - W_1 p - P_1 w) = -mg\theta \sin \Phi_1 \sin \Theta_1 + mg\phi \cos \Phi_1 \cos \Theta_1 + f_{A_y} + f_{T_y} \quad (58)$$

$$m(\dot{w} - U_1 q - Q_1 u + V_1 p + P_1 v) = -mg\theta \cos \Phi_1 \sin \Theta_1 - mg\phi \sin \Phi_1 \cos \Theta_1 + f_{A_z} + f_{T_z} \quad (59)$$

$$I_{xx} \dot{p} - I_{xz} \dot{r} - I_{xz} (P_1 q + Q_1 p) + (I_{zz} - I_{yy})(R_1 q + Q_1 r) = l_A + l_T \quad (60)$$

$$I_{yy} \dot{q} + (I_{xx} - I_{zz})(P_1 r + R_1 p) + I_{xz} (2P_1 p - 2R_1 r) = m_A + m_T \quad (61)$$

$$I_{zz} \dot{r} - I_{xz} \dot{p} + (I_{yy} - I_{xx})(P_1 q + Q_1 p) + I_{xz}(Q_1 r + R_1 q) = n_A + n_T \quad (62)$$

$$p = \dot{\phi} - \dot{\Psi}_1 \theta \cos \Theta_1 - \dot{\psi} \sin \Theta_1 \quad (63)$$

$$q = -\dot{\Theta}_1 \phi \sin \Phi_1 + \dot{\theta} \cos \Phi_1 + \dot{\Psi}_1 \phi \cos \Theta_1 \cos \Phi_1 - \dot{\Psi}_1 \theta \sin \Theta_1 \sin \Phi_1 + \dot{\psi} \cos \Theta_1 \sin \Phi_1 \quad (64)$$

$$r = -\dot{\Psi}_1 \phi \cos \Theta_1 \sin \Phi_1 - \dot{\Psi}_1 \theta \sin \Theta_1 \cos \Phi_1 + \dot{\psi} \cos \Theta_1 \cos \Phi_1 - \dot{\Theta}_1 \phi \cos \Phi_1 - \dot{\theta} \sin \Phi_1 \quad (65)$$

which form the nine perturbed equations of motion relative to a very general steady state in which all motion variables are allowed to have non-zero steady state values. By considering that the majority of airplane dynamic stability problems are concerned with perturbed motions relative to a wings level, steady state, straight line flight conditions with a relative small flight path angle, the following conditions hold and allow a further simplification of the perturbed equations of motion:

- No initial steady state side velocities exists; $V_1=0$.
- No initial steady bank angle exists; $\Phi_1=0$.
- No initial angular velocities exists; $P_1=Q_1=R_1=\dot{\Psi}_1 = \dot{\Theta}_1 = \dot{\Phi}_1=0$

thus yielding a simplified perturbed state equations:

$$m(\dot{u} - W_1 q) = -mg\theta \cos \Theta_1 + f_{A_x} + f_{T_x} \quad (66)$$

$$m(\dot{v} + U_1 r - W_1 p) = mg\phi \cos \Theta_1 + f_{A_y} + f_{T_y} \quad (67)$$

$$m(\dot{w} - U_1 q) = -mg\theta \sin \Theta_1 + f_{A_z} + f_{T_z} \quad (68)$$

$$I_{xx} \dot{p} - I_{xz} \dot{r} = l_A + l_T \quad (69)$$

$$I_{yy} \dot{q} = m_A + m_T \quad (70)$$

$$I_{zz} \dot{r} - I_{xz} \dot{p} = n_A + n_T \quad (71)$$

$$p = \dot{\phi} - \dot{\psi} \sin \Theta_1 \quad (72)$$

$$q = \dot{\theta} \quad (73)$$

$$r = \dot{\psi} \cos \Theta_1 \quad (74)$$

The perturbed state equations of motion need to be augmented using the perturbed aerodynamic forces and moments, and the perturbed thrust forces and moments. The perturbed aerodynamic forces and moments, and the perturbed thrust forces and moments are expressed in state space form [78] as:

$$\begin{bmatrix} \frac{f_{A_x}}{q_1 S} \\ \frac{f_{A_z}}{q_1 S} \\ \frac{m_A}{q_1 \bar{c} S} \end{bmatrix} = \begin{bmatrix} -(C_{D_u} + 2C_{L_1}) & (-C_{D_\alpha} + C_{L_1}) & -C_{D_\alpha} & -C_{D_q} & -C_{D_{\delta e}} \\ -(C_{L_u} + 2C_{L_1}) & (-C_{L_\alpha} - C_{D_1}) & -C_{L_\alpha} & -C_{L_q} & -C_{L_{\delta e}} \\ (C_{m_u} + 2C_{m_1}) & C_{m_\alpha} & C_{m_\alpha} & C_{m_q} & C_{m_{\delta e}} \end{bmatrix} \begin{bmatrix} \frac{u}{U_1} \\ \alpha \\ \frac{\dot{\alpha} \bar{c}}{2U_1} \\ \frac{q \bar{c}}{2U_1} \\ \delta_e \end{bmatrix} \quad (75)$$

$$\begin{bmatrix} \frac{f_{A_y}}{q_1 S} \\ \frac{l_A}{q_1 \bar{c} S} \\ \frac{m_A}{q_1 \bar{c} S} \end{bmatrix} = \begin{bmatrix} C_{y_\beta} & C_{y_\beta} & C_{y_p} & C_{y_r} & C_{y_{\delta a}} & C_{y_{\delta r}} \\ C_{l_\beta} & C_{l_\beta} & C_{l_p} & C_{l_r} & C_{l_{\delta a}} & C_{l_{\delta r}} \\ C_{n_\beta} & C_{n_\beta} & C_{n_p} & C_{n_r} & C_{n_{\delta a}} & C_{n_{\delta r}} \end{bmatrix} \begin{bmatrix} \beta \\ \dot{\beta} b \\ \frac{pb}{2U_1} \\ \frac{rb}{2U_1} \\ \delta_a \\ \delta_r \end{bmatrix} \quad (76)$$

$$\begin{bmatrix} \frac{f_{A_x}}{q_1 S} \\ \frac{f_{A_z}}{q_1 S} \\ \frac{m_A}{q_1 \bar{c} S} \end{bmatrix} = \begin{bmatrix} (C_{T_{x_u}} + 2C_{T_{x_1}}) & 0 \\ 0 & 0 \\ (C_{M_{x_u}} + 2C_{M_{x_1}}) & C_{m_{T\alpha}} \end{bmatrix} \begin{bmatrix} \frac{u}{U_1} \\ \alpha \end{bmatrix} \quad (77)$$

$$\begin{bmatrix} \frac{f_{T_y}}{q_1 S} \\ \frac{l_T}{q_1 S} \\ \frac{n_T}{q_1 \bar{c} S} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ C_{n_{T\beta}} \end{bmatrix} \beta \quad (78)$$

where q_l is the steady state dynamic pressure defined by $q_l = \frac{1}{2} \rho U_1^2$, S is the wing area, b is the span of the wing, and \bar{c} is the mean aerodynamic chord of the wing. The coefficients in Eqs. (75)-(78) represent stability and control dimensionless coefficients that capture the relationship of the partial derivatives of the forces and moments with respect to the state vectors. A more detailed description of these dimensionless coefficients is provided in reference [78].

4.1.5. Longitudinal Linearized Model. In order to simplify the stability and control analysis problem for airplanes, the perturbed equations of motion are generally decoupled into longitudinal and lateral-directional modes. Only the first one will be developed in this section, since the work described here only looks at the longitudinal dynamics for the model airplane that will be described in section 4.2. For the longitudinal model Eqs. (66), (68), (68) and (73) are selected. The term W_l , which corresponds to the steady state velocity along the Z-axis, is by definition equal to zero since the stability axis system is selected as the coordinate system. By making the substitutions $q = \dot{\theta}$ and $w = U_1 \alpha$ in Eqs. (66), (68), (70) and (73), these equations can now be expressed in terms the perturbed forward velocity, u , the perturbed aerodynamic angle-of-attack, α , the perturbed pitch attitude angle, θ , and the perturbed pitch rate, q . In order to simplify the state space notation, after substituting Eqs. (75) and (77) into the

perturbed longitudinal equations of motion, the dimensional stability derivatives are introduced [78]:

$$X_U = -\frac{\bar{q}S(C_{D_u} + 2C_{D_1})}{mU_1} \quad (79)$$

$$X_{T_U} = \frac{\bar{q}S(C_{T_{x_u}} + 2C_{T_{x_1}})}{mU_1} \quad (80)$$

$$X_\alpha = -\frac{\bar{q}S(C_{D_\alpha} + C_{L_1})}{mU_1} \quad (81)$$

$$X_{\delta_e} = -\frac{\bar{q}SC_{D_{\delta_e}}}{m} \quad (82)$$

$$Z_U = -\frac{\bar{q}S(C_{L_u} + 2C_{L_1})}{mU_1} \quad (83)$$

$$Z_\alpha = -\frac{\bar{q}S(C_{L_\alpha} + C_{L_1})}{m} \quad (84)$$

$$Z_\alpha = -\frac{\bar{q}S\bar{c}C_{L_\alpha}}{2mU_1} \quad (85)$$

$$Z_q = -\frac{\bar{q}S\bar{c}C_{L_q}}{2mU_1} \quad (86)$$

$$M_U = \frac{\bar{q}S\bar{c}(C_{M_u} + 2C_{M_1})}{I_{yy}U_1} \quad (87)$$

$$M_{T_U} = \frac{\bar{q}S\bar{c}(C_{M_{T_u}} + 2C_{M_{T_1}})}{I_{yy}U_1} \quad (88)$$

$$M_\alpha = \frac{\bar{q}S\bar{c}C_{M_\alpha}}{I_{yy}} \quad (89)$$

$$M_{T_\alpha} = \frac{\bar{q}S\bar{c}C_{M_{T_\alpha}}}{I_{yy}} \quad (90)$$

$$M_{\alpha} = \frac{\bar{q}S\bar{c}^{-2} C_{M_{\alpha}}}{2I_{yy}U_1} \quad (91)$$

$$M_q = \frac{\bar{q}S\bar{c}^{-2} C_{M_q}}{2I_{yy}U_1} \quad (92)$$

$$M_{\delta_e} = \frac{\bar{q}S\bar{c}C_{M_{\delta_e}}}{I_{yy}} \quad (93)$$

The perturbed longitudinal equations with dimensional stability derivatives become:

$$\dot{u} = -g\theta \cos \theta_1 + X_u u + X_{Tu} u + X_{\alpha} \alpha + X_{\delta_{ele}} \delta_{ele} \quad (94)$$

$$U_1 \dot{\alpha} - U_1 \dot{\theta} = -g\theta \sin \theta_1 + Z_u u + Z_{\alpha} \alpha + Z_{\dot{\alpha}} \dot{\alpha} + Z_q \dot{\theta} + Z_{\delta_{ele}} \delta_{ele} \quad (95)$$

$$\ddot{\theta} = M_u u + M_{Tu} u + M_{\alpha} \alpha + M_{T\alpha} \alpha + M_{\dot{\alpha}} \dot{\alpha} + M_q \dot{\theta} + M_{\delta_{ele}} \delta_{ele} \quad (96)$$

$$\dot{\theta} = q \quad (97)$$

The state-space matrix model for the longitudinal mode can be expressed as

$$\dot{X}_{lon} = E_{lon}^{-1} A_{lon} X_{lon} + E_{lon}^{-1} B_{lon} U_{lon} \quad (98)$$

where the state and the control vectors are defined respectively as

$$X_{lon} = [u \quad \alpha \quad q \quad \theta]^T \quad (99)$$

$$U_{lon} = [\delta_e] \quad (100)$$

where δ_e is the elevator deflection and the matrices in Eq. (98) are defined by

$$E_{lon} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & U_1 - Z_\alpha & 0 & 0 \\ 0 & -M_\alpha & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (101)$$

$$A_{lon} = \begin{bmatrix} (X_U + X_{TU}) & X_\alpha & X_q & -g \cos \theta_1 \\ Z_U & Z_\alpha & (Z_q + U_1) & -g \sin \theta_1 \\ (M_U + M_{TU}) & (M_\alpha + M_{T\alpha}) & M_q & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (102)$$

$$B_{lon} = \begin{bmatrix} X_{\delta e} \\ Z_{\delta e} \\ M_{\delta e} \\ 0 \end{bmatrix} \quad (103)$$

After substituting Eqs. (101), (102) and (103) into Eq. (98), the longitudinal differential equations in state space become:

$$\begin{bmatrix} \dot{u} \\ \dot{\alpha} \\ \dot{q} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} (X_U + X_{TU}) & X_\alpha & X_q & -g \cos \theta_1 \\ \frac{Z_U}{u_1 - Z_{\dot{\alpha}}} & \frac{Z_\alpha}{u_1 - Z_{\dot{\alpha}}} & (Z_q + U_1) & -g \sin \theta_1 \\ M_\alpha Z_U + (M_U + M_{TU}) & M_\alpha Z_\alpha + (M_\alpha + M_{T\alpha}) & M_\alpha (Z_q + U_1) + M_q & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} u \\ \alpha \\ q \\ \theta \end{bmatrix} + \begin{bmatrix} X_{\delta e} \\ Z_{\delta e} \\ M_\alpha Z_{\delta e} + M_{\delta e} \\ 0 \end{bmatrix} \delta_{ele} \quad (104)$$

4.1.6. Phugoid and Short Period Approximations. The analysis of the dynamic response of the longitudinal state-space model can be conducted via modal or eigenvector analysis, which shows that the longitudinal motion can be differentiated in two modes according to the nature or the behavior of the system: short period mode and Phugoid, or long-period mode. Each one of the two modes describes a completely different characteristic of the behavior of the aircraft being modeled, and it is crucial for the aerospace engineer to consider the existence of both modes.

According to Nelson [84] “We can think of the long-period or Phugoid mode as a gradual interchange of potential and kinetic energy about the equilibrium attitude and airspeed.” The long-period mode is characterized by changes in pitch attitude, altitude and velocity at a nearly constant angle of attack. An approximation to the Phugoid mode can be obtained by neglecting the pitch moment equation and assuming that the change in angle of attack is zero.

$$\dot{u} = -g\theta \cos \theta_1 + X_u u + X_{T_u} u + X_{\delta_{ele}} \delta_{ele} \quad (105)$$

$$-U_1 \dot{\theta} = -g\theta \sin \theta_1 + Z_u u + Z_{\delta_{ele}} \delta_{ele} \quad (106)$$

being the state-space approximation for the long-period mode:

$$\begin{bmatrix} \dot{u} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} (X_u + X_{T_u}) & -g \cos \theta_1 \\ -\frac{Z_u}{U_1} & \frac{g \sin \theta_1}{U_1} \end{bmatrix} \begin{bmatrix} u \\ \theta \end{bmatrix} + \begin{bmatrix} X_{\delta_{ele}} \\ -\frac{Z_{\delta_{ele}}}{U_1} \end{bmatrix} \delta_{ele} \quad (107)$$

Out of the two modes, the long-period mode is the one that occurs so slowly, that the pilot can easily counteract and eliminate the disturbance by small control movements. The second of the modes, the short period, is the more important of the two when considering the stability and control behavior of the aircraft. If an airplane has a high frequency and a heavily damped short period, then the airplane will respond rapidly to an elevator input without any undesirable overshoot, yet to the contrary the short-period mode is lightly damped or has a relatively low frequency, the airplane will be difficult to control and in some case may even be dangerous to fly [84]. The approximation for the short-period can be made by assuming that the change in forward velocity is zero and dropping the X-force Eq. (94) yielding

$$\dot{\alpha} = \frac{Z_{\alpha}}{(U_1 - Z_{\dot{\alpha}})} \alpha + \frac{Z_q + U_1}{U_1 - Z_{\dot{\alpha}}} q + \frac{Z_{\delta_{ele}}}{U_1 - Z_{\dot{\alpha}}} \delta_{ele} \quad (108)$$

$$\dot{q} = \left(M_\alpha + M_{T\alpha} + \frac{M_{\dot{\alpha}} Z_\alpha}{(U_1 - Z_{\dot{\alpha}})} \right) \alpha + \left(M_q + \frac{M_{\dot{\alpha}} (Z_q + U_1)}{U_1 - Z_{\dot{\alpha}}} \right) q + \left(M_{\delta_{ele}} + \frac{M_{\dot{\alpha}} Z_{\delta_{ele}}}{U_1 - Z_{\dot{\alpha}}} \right) \delta_{ele} \quad (109)$$

The state-space approximation for the short-period mode is defined as

$$\begin{bmatrix} \dot{\alpha} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} \frac{Z_\alpha}{(U_1 - Z_{\dot{\alpha}})} & \frac{Z_q + U_1}{U_1 - Z_{\dot{\alpha}}} \\ \left(M_\alpha + M_{T\alpha} + \frac{M_{\dot{\alpha}} Z_\alpha}{(U_1 - Z_{\dot{\alpha}})} \right) & \left(M_q + \frac{M_{\dot{\alpha}} (Z_q + U_1)}{U_1 - Z_{\dot{\alpha}}} \right) \end{bmatrix} \begin{bmatrix} \alpha \\ q \end{bmatrix} + \begin{bmatrix} Z_{\delta_{ele}} \\ U_1 - Z_{\dot{\alpha}} \\ M_{\delta_{ele}} + \frac{M_{\dot{\alpha}} Z_{\delta_{ele}}}{U_1 - Z_{\dot{\alpha}}} \end{bmatrix} \delta_{ele} \quad (110)$$

Defining the Short period approximation for this thesis is important as it will be used in section 4.3 when defining the model to be used for the tracking controller.

4.2. NONLINEAR HIGH ANGLE OF ATTACK PROBLEM

This section introduces and describes in detail the nonlinear aircraft model that will be used to develop an optimal NN controller that will try to extend the range of angle of attack beyond the stall regions at which the airplane can operate. The model is extracted from a paper written by Garrard and Jordan [85], and uses a model of a F-8 Crusader fighter aircraft. Figure 4.8 shows a picture of the F-8 Crusader in flight taken from the public Dryden Flight Research Center Graphics Collection library [79]. As mentioned earlier, for this type of aircraft, the horizontal stabilizer rotates completely as if the entire tail was an elevator. The control deflection will be referred as tail rotation throughout the remainder of this thesis.

4.2.1. Problem Formulation. The nonlinear model used in this thesis is obtained from Garrard and Jordan [85]. The model described in this section is in the continuous-time domain, and will be discretized prior to implementation in the computer. Figure 4.9 shows the coordinate system used in the formulation. From the coordinate system described in Figure 4.9, the lift component is separated into its wing and tail components [86] yielding the basic decoupled longitudinal equations of motion:

$$m(\dot{u} + w\dot{\theta}) = -mg \sin \theta + L_w \sin \alpha + L_t \sin \alpha_t \quad (111)$$

$$m(\dot{w} - U_1\dot{\theta}) = mg \cos \theta - L_w \cos \alpha - L_t \cos \alpha_t \quad (112)$$

$$I_{yy}\ddot{\theta} = M_w + lL_w \cos \alpha - l_t L_t \cos \alpha_t - c\dot{\theta} \quad (113)$$



Figure 4.8. F-8 Crusader in Flight^[79].

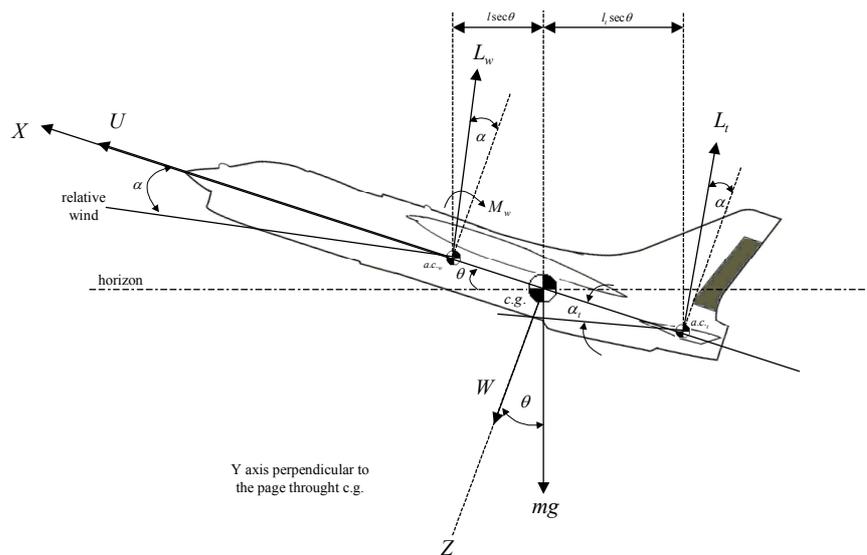


Figure 4.9. F-8 Aircraft Dynamic Coordinate System^[79].

where L_w and L_t are the wing and tail lift respectively, M_w is the wing moment about the center of gravity, l is the distance between the wing aerodynamic center and the aircraft center of gravity, l_t is the distance between the tail aerodynamic center and the aircraft center of gravity, and $c\dot{\theta}$ is the damping moment. Note that this model considers the drag contribution small compared with the lift and weight and therefore is neglected the analysis. The L_w and L_t terms are defined as

$$L_w = C_{L_w} \bar{q} S \quad (114)$$

$$L_t = C_{L_t} \bar{q} S_t \quad (115)$$

where C_{L_w} and C_{L_t} represent the lift coefficient for the wing and tail respectively, \bar{q} is the dynamic pressure, S is the wing area and S_t is the tail area. Garrard and Jordan [85] show that for large angles of attack, cubic approximations of the lift coefficient are more accurate in predicting the lift capabilities of an aerodynamic surface, i.e. wing or tail, thus they approximate both the wing and tail lift coefficients with cubic approximations which are defined in Eqs. (116) and (117)

$$C_{L_w} = C_{L_w}^0 + C_{L_w}^1 \alpha - C_{L_w}^2 \alpha^3 \quad (116)$$

$$C_{L_t} = C_{L_t}^0 + C_{L_t}^1 \alpha_t - C_{L_t}^2 \alpha_t^3 + a_e \delta_e \quad (117)$$

where $C_{L_w}^0, C_{L_w}^1, C_{L_w}^2, C_{L_t}^0, C_{L_t}^1, C_{L_t}^2$ are constants peculiar to the individual aircraft, δ_e represents the tail rotation, and a_e represents the variation in tail lift coefficient with respect to the tail rotation, $\partial C_{L_t} / \partial \delta_e$. Recalling that the entire tail of the F-8 Crusader rotates, then the tail angle of attack can be redefined as

$$\alpha_t = \alpha - \varepsilon + \delta_e \quad (118)$$

where ε is the downwash angle. The downwash angle is a consequence of the trailing vortices created by the wing tips, which can effectively change the direction of

the flow that the tail sees, i.e. $\alpha_t \neq \alpha_w$. When the tail is within the wing wake, this effect cannot be ignored, and this is the case for the F-8 Crusader. The downwash can be expressed as:

$$\varepsilon = \varepsilon_0 + a_\varepsilon \alpha \quad (119)$$

where ε_0 is a constant that represent the downwash angle at zero angle of attack and a_ε represents the rate of change in downwash with respect to the angle of attack, $\partial\varepsilon/\partial\alpha$. Garrard and Jordan [85] introduce the following approximations:

$$w = U_1 \tan \alpha \approx U_1 \left(\alpha + \frac{\alpha^3}{3} \right) \quad (120)$$

$$\dot{w} = \dot{U}_1 \tan \alpha + U_1 \dot{\alpha} \sec^2 \alpha \approx U_1 \dot{\alpha} \sec^2 \alpha \quad (121)$$

A nominal flight condition of $\theta_0 = 0$ is assumed so that $\theta = \Delta\theta$ and the following trigonometric assumptions are made:

$$\cos \theta \approx 1 - \frac{\theta^2}{2} \quad (122)$$

$$\cos \alpha \approx 1 - \frac{\alpha^2}{2} \quad (123)$$

$$\sin \theta \approx \theta - \frac{\theta^3}{6} \quad (124)$$

$$\sin \alpha \approx \alpha - \frac{\alpha^3}{6} \quad (125)$$

Substituting Eqs. (120)-(125) into Eqs. (111)-(113) yield the nonlinear equations of motion:

$$\dot{U} = -U \left(\alpha + \frac{\alpha^3}{3} \right) \dot{\theta} - g \left(\theta - \frac{\theta^3}{6} \right) + \left(C_{L_w}^0 + C_{L_w}^1 \alpha - C_{L_w}^2 \frac{\alpha^3}{6} \right) \frac{\bar{q}}{m} S + \left(\left(C_{L_t}^0 + C_{L_t}^1 ((1-a_\varepsilon)\alpha - \varepsilon_0 + \delta_e) - C_{L_t}^2 ((1-a_\varepsilon)\alpha - \varepsilon_0 + \delta_e)^3 + a_e \delta_e \right) \left((1-a_\varepsilon)\alpha - \varepsilon_0 + \delta_e - \frac{((1-a_\varepsilon)\alpha - \varepsilon_0 + \delta_e)^3}{6} \right) \right) \frac{\bar{q}}{m} S_t \quad (126)$$

$$\dot{\alpha} = \dot{\theta} (1 - \alpha^2) + \frac{g}{U} \left(1 - \frac{\theta^2}{2} - \alpha^2 \right) - \left(C_{L_w}^0 + C_{L_w}^1 \alpha - C_{L_w}^2 \alpha^3 - C_{L_w}^0 \frac{3\alpha^2}{2} - C_{L_w}^1 \frac{3\alpha^3}{2} \right) \frac{\bar{q}S}{mU} - \left(\left(C_{L_t}^0 + C_{L_t}^1 ((1-a_\varepsilon)\alpha - \varepsilon_0 + \delta_e) - C_{L_t}^2 ((1-a_\varepsilon)\alpha - \varepsilon_0 + \delta_e)^3 + a_e \delta_e \right) \left(1 - \frac{((1-a_\varepsilon)\alpha - \varepsilon_0 + \delta_e)^2}{2} \right) \right) \frac{\bar{q}S_t}{mU} \quad (127)$$

$$\ddot{\theta} = M_w I_{yy}^{-1} + \left(C_{L_w}^0 + C_{L_w}^1 \alpha - C_{L_w}^2 \alpha^3 \right) \left(1 - \frac{\alpha^2}{2} \right) \bar{q} S I_{yy}^{-1} - \left(\left(C_{L_t}^0 + C_{L_t}^1 ((1-a_\varepsilon)\alpha - \varepsilon_0 + \delta_e) - C_{L_t}^2 ((1-a_\varepsilon)\alpha - \varepsilon_0 + \delta_e)^3 + a_e \delta_e \right) \left(1 - \frac{((1-a_\varepsilon)\alpha - \varepsilon_0 + \delta_e)^2}{2} \right) \bar{q} S_t I_{yy}^{-1} \right) - c \dot{\theta} I_{yy}^{-1} \quad (128)$$

Considering disturbances of an F-8 Crusader in level, unaccelerated flight at Mach=0.85, and an altitude of 30,000 ft, substituting the data provided in [85] for the F-8 Crusader, introducing the trim conditions [85], and ignoring Eq. (126) the equations, become:

$$\dot{\alpha} = \dot{\theta} - \alpha^2 \dot{\theta} - 0.088\alpha \dot{\theta} - 0.877\alpha + 0.47\alpha^2 + 3.846\alpha^3 + 0.01618\delta_e \alpha + 0.00367\delta_e^2 - 0.215\delta_e + 0.28\delta_e \alpha^2 + 0.47\delta_e^2 \alpha + 0.63\delta_e^3 - 0.019\theta^2 \quad (129)$$

$$\ddot{\theta} = -0.396\dot{\theta} - 4.208\alpha - 0.47\alpha^2 - 3.564\alpha^3 - 0.7916\delta_e \alpha - 20.967\delta_e + 6.265\delta_e \alpha^2 + 46\delta_e^2 + 61.4\delta_e^3 \quad (130)$$

For the nonlinear equations (129) and (130), the terms involving $\delta_e^n, n = 2, 3, 4, \dots$ and $\alpha^n \delta_e^m, n, m = 1, 2, 3, \dots$ are eliminated since these terms are small. The resulting continuous-time nonlinear equations of motion can be expressed in the form

$$\dot{x} = Ax + \phi(x) + b\delta_e \quad (131)$$

where x describes the states of the system,

$$x = [\alpha \quad \theta \quad q]^T \quad (132)$$

The A matrix represents the linearized plant of the system:

$$A = \begin{bmatrix} -0.877 & 0 & 1 \\ 0 & 0 & 1 \\ -4.208 & 0 & -0.396 \end{bmatrix} \quad (133)$$

and b represents the control matrix of the system

$$b = \begin{bmatrix} -0.215 \\ 0 \\ -20.967 \end{bmatrix} \quad (134)$$

The nonlinear aerodynamics due to high angles of attack are embedded in the extra term $\phi(x)$

$$\phi(x) = \begin{bmatrix} -\alpha^2 q - 0.088\alpha q - 0.019\theta^2 + 0.47\alpha^2 + 3.846\alpha^3 \\ 0 \\ -0.47\alpha^2 - 3.564\alpha^3 \end{bmatrix} \quad (135)$$

In order to be consistent with the discrete-time formulation, the nonlinear equations of motion described in Eq. (131) are discretized so they can be implemented in the computer.

4.2.2. Optimal Control Formulation. The control problem consist is minimize the quadratic performance index given by

$$J = \frac{1}{2} \sum_0^{\infty} [x_i^T Q x_i + R \delta_{e(i)}^2] \quad (136)$$

The state and control weight matrices used in this thesis are defined by [85]

$$Q = \begin{bmatrix} 0.25 & 0 & 0 \\ 0 & 0.25 & 0 \\ 0 & 0 & 0.25 \end{bmatrix} \quad (137)$$

and the controller weight is $R=1$. In order to analyze the implications of the state and control weighting used in this thesis, the state weight matrix is assumed to be of the form described in Eqs. (138) and (139)

$$Q = \begin{bmatrix} \frac{1}{\alpha_{MAX}^2} & 0 & 0 \\ 0 & \frac{1}{\theta_{MAX}^2} & 0 \\ 0 & 0 & \frac{1}{q_{MAX}^2} \end{bmatrix} \quad (138)$$

where α_{MAX} , θ_{MAX} , and q_{MAX} define the maximum allowable states by the cost function. Using the magnitudes described in Eq. (137), the maximum allowable states are $\alpha_{MAX} = \theta_{MAX} = 2rad \approx 114.59^\circ$, and $q_{MAX} = 2rad/sec \approx 114.59^\circ/sec$, which seem to be unrealistically high allowable states, unless you realize that the purpose of this controller is to extend the range of angle of attack beyond the stall regions, and then after reaching those extreme angles of attack conditions, the controller needs to be able safely recover to the zero-steady-state. If the weight cost function penalizes the higher values of the states, then the controller will not be able to extend the stall region beyond the

penalized values. Analyzing the physical dynamics of the airplane shows that the angle of attack, pitch angle and pitch rate are coupled in such a way that if only the pitch angle and pitch rate were penalized, the initial angle of attack would not be extended, thus it was required to have a cost function that would not be penalized the extreme values. Similarly, if the control weight matrix is considered to be of the form

$$R = \begin{bmatrix} 1 \\ \delta_{eMAX}^2 \end{bmatrix} \quad (139)$$

where δ_{eMAX} defines the maximum allowable control by the cost function, then $\delta_{eMAX} = 1rad \approx 57.29^\circ$ which exceeds the maximum allowable control $\pm 25^\circ$. Again, the same logic described above holds for explaining the fact that the weight matrix does not penalize the controller for being above the maximum allowable control in greater extent. The physical limits on the controller are enforced by logic in the integration routines such that only a maximum tail rotation of $\pm 25^\circ$ is allowed, and similarly a maximum tail rotation rate of $\pm 60^\circ/sec$, is also enforced [85].

Garrard and Jordan propose three different controllers to extend the stall regime. These controllers will be used as a benchmark against the NN controller developed in this thesis. The first model represents the LQR solution of the linear model of Eq. (131). The linear model used is defined by removing the nonlinear dynamics such that Eq. (131) becomes

$$\begin{bmatrix} \dot{\alpha} \\ \dot{\theta} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} -0.877 & 0 & 1 \\ 0 & 0 & 1 \\ -4.208 & 0 & -0.396 \end{bmatrix} \begin{bmatrix} \alpha \\ \theta \\ q \end{bmatrix} + \begin{bmatrix} -0.215 \\ 0 \\ -20.967 \end{bmatrix} \delta_e \quad (140)$$

Solving the associated steady state Riccati equation yields the gain matrix

$$K = [0.052559 \quad -0.5 \quad -0.521044] \quad (141)$$

which is expressed in the discrete-time feedback LQR linear control as

$$\delta_{e1,(i)} = -Kx_i = -0.053\alpha_i + 0.5\theta_i + 0.521q_i \quad (142)$$

Garrard and Jordan [85] propose a second and third order controllers to help reduce the loss of attitude when the angle of attack exceeds the stall limit by using singular perturbations to solve the Hamiltonian-Jacobi partial differential equation [85] defined in Eq. (143)

$$\frac{\partial V^T}{\partial x} Ax + \frac{\partial V^T}{\partial x} \phi - \frac{1}{4} \frac{\partial V^T}{\partial x} b R^{-1} b^T \frac{\partial V^T}{\partial x} + x^T Q x = 0 \quad (143)$$

with $V(0) = 0$. As demonstrated by Lee and Markus [87], the unique optimal feedback control using this approach is provided by

$$\delta_e = -\frac{1}{2} R^{-1} b^T \frac{\partial V}{\partial x} \quad (144)$$

Due to the complexity involved in analytically solving Eq. (143), perturbational procedures are used to obtain approximate solutions, [88], [89], [90], [91] and [92]. The optimal solution can be presented in series format as

$$V(x) = \sum_{n=0}^{\infty} V_n(x) \quad (145)$$

If the nonlinear dynamics is defined

$$\phi(x) = \sum_{n=1}^N f_{n+1}(x) \quad (146)$$

$$\phi(x) = \sum_{n=1}^N f_{n+1}(x) = \begin{bmatrix} 0.47\alpha^2 - 0.088\alpha q - 0.019\theta^2 \\ 0 \\ -0.47\alpha^2 \end{bmatrix} + \begin{bmatrix} 3.846\alpha^3 - \alpha^2 q \\ 0 \\ -3.564\alpha^3 \end{bmatrix} \quad (147)$$

The V_n 's terms are given by the following equations:

$$\frac{\partial V_0^T}{\partial x} Ax - \frac{1}{4} \frac{\partial V_0^T}{\partial x} b r^{-1} b^T \frac{\partial V_0^T}{\partial x} + x^T Q x = 0 \quad (148)$$

$$\frac{\partial V_1^T}{\partial x} Ax - \frac{1}{4} \frac{\partial V_1^T}{\partial x} b R^{-1} b^T \frac{\partial V_0^T}{\partial x} - \frac{1}{4} \frac{\partial V_0^T}{\partial x} b R^{-1} b^T \frac{\partial V_1^T}{\partial x} + \frac{\partial V_0^T}{\partial x} f_2 = 0 \quad (149)$$

$$\frac{\partial V_n^T}{\partial x} Ax - \frac{1}{4} \frac{\partial V_n^T}{\partial x} b R^{-1} b^T \frac{\partial V_0^T}{\partial x} - \frac{1}{4} \frac{\partial V_0^T}{\partial x} b R^{-1} b^T \frac{\partial V_n^T}{\partial x} + \frac{\partial V_0^T}{\partial x} f_{n+1} + \quad (150)$$

$$\sum_{K=1}^{n-1} \frac{\partial V_K^T}{\partial x} f_{n+1-K} - \frac{1}{4} \sum_{K=1}^{n-1} \frac{\partial V_K^T}{\partial x} b R^{-1} b^T \frac{\partial V_{n-K}^T}{\partial x} = 0$$

The resulting optimal control is

$$\delta_e = -R^{-1} b^T \sum_{n=0}^{\infty} \frac{\partial V_n}{\partial x} \quad (151)$$

Determination for the nonlinear controller via perturbation methods is really laborious complex with simple algebra involved but tedious. The solution for the first equation of Eq. (145) is

$$V_0 = x^T P x \quad (152)$$

where P is the solution to the steady state Riccati equation, Eq. (22).

$$\frac{\partial V_0}{\partial x} = P x \quad (153)$$

and substituting Eq. (153) into Eq. (151)

$$\delta_e = -R^{-1}b^T Px \quad (154)$$

with

$$P = \begin{bmatrix} 0.16090086046144 & -0.08882707457570 & -0.00415667734054 \\ -0.08882707457570 & 0.35915318511485 & 0.02475784905012 \\ -0.00415667734054 & 0.02475784905012 & 0.02489329375966 \end{bmatrix}$$

yielding the same results as described previously:

$$\delta_{e1} = -0.0526\alpha + 0.5\theta + 0.521q \quad (142)$$

Determination of the rest of the terms in Eq. (145), ie. V_n , $n=1,2,3,\dots$ is as follows

Assume, where $x_1=\alpha$, $x_2=\theta$ and $x_3=q$

$$V_n = \sum_{k=0}^{n+2-k} \sum_{j=0}^{n+2} a_{n+2-j-k,j,k}^n x_1^{n+2-j-k} x_2^j x_3^k \quad (155)$$

Calculate $\frac{\partial V_n}{\partial x}$.

Substitutes $\frac{\partial V_n}{\partial x}$ into Eq. (150).

Set the sum of coefficients of like terms equal to zero.

Solve the resulting simultaneous algebraic equations for $a_{n+2-j-k,j,k}^n$

After V_n is obtained, then $\frac{\partial V_n}{\partial x}$ can be calculated and substituted into Eq. (151) to

obtain the $\delta_{e,n+1}$

For the second order control, ten unknown coefficients must be found, where four coefficients are almost zero, thus yielding the resulting expression for V_1

$$V_1 = 0.058x_1^3 - 0.077x_1^2x_2 - 0.002x_1^2x_3 + 0.045x_1x_2^2 - 0.015x_2^3 - 0.003x_1x_2x_3 \quad (156)$$

the resulting expression for $\frac{\partial V_1}{\partial x}$ is

$$\frac{\partial V_1}{\partial x} = \begin{bmatrix} 0.174x_1^2 - 0.154x_1x_2 - 0.004x_1x_3 + 0.045x_2^2 - 0.003x_2x_3 \\ -0.077x_2 + 0.09x_1x_2 - 0.045x_2^2 - 0.003x_1x_3 \\ -0.002x_1^2 - 0.003x_1x_2 \end{bmatrix} \quad (157)$$

and substituting Eq. (157) into Eq. (151) yielding

$$\delta_{e,2} = 0.04x_1^2 - 0.048x_1x_2 + 0.0004x_1x_3 + 0.005x_2^2 - 0.0003x_2x_3 \quad (158)$$

where only the first two terms are significant. When expressing the non-linear control up to the 2nd Order terms in the discrete-time form yields:

$$\delta_{e2(i)} = -0.0526\alpha_i + 0.5\theta_i + 0.521q_i + 0.04\alpha_i^2 - 0.048\alpha_i\theta_i \quad (159)$$

Using the similar methodology the nonlinear discrete-time controller up to the third order terms the 3rd-Order controller is

$$\delta_{e3(i)} = -0.0526\alpha_i + 0.5\theta_i + 0.521q_i + 0.04\alpha_i^2 - 0.048\alpha_i\theta_i + 0.374\alpha_i^3 - 0.312\alpha_i^2\theta_i \quad (160)$$

Equations (142), (159) and (160) yield the controllers proposed by Garrard and Jordan [82] which will be used as bench marks against the neurocontroller developed in this work.

This thesis uses the NN based architectures and Hamiltonian formulation, described previously in section 2.1.2, to develop near optimal feedback control laws for the aircraft problem outlined in Section 4.2.1. The Hamiltonian for this optimal control problem, defined previously in Eq. (8), is redefined to include the system model described in Eq. (131)

$$H_i = \frac{1}{2} \left(x_i^T Q x_i + R \delta_{e_i}^2 \right) + \lambda_{i+1}^T f_i(x_i, u_i) \quad (161)$$

Expanding the right hand side of Eq. (4) and using Eq. (161), the discrete-time co-state or Lagrange multipliers differential equations become.

$$\lambda_{1(i)} = \frac{\partial H_i}{\partial \alpha_i} \quad (162)$$

$$\lambda_{2(i)} = \frac{\partial H_i}{\partial \theta_i} \quad (163)$$

$$\lambda_{3(i)} = \frac{\partial H_i}{\partial q_i} \quad (164)$$

Note that the discrete-time sub-index is set in parenthesis to avoid confusion where other sub-indexes are present. The necessary condition for optimal control [47] is defined by Eq. (6)

$$\frac{\partial H_i}{\partial u_i} = 0 \quad (6)$$

which leads to the optimal control equation

$$\delta_{e(i)}^* = -R^{-1} b^T \lambda_{i+1} \quad (165)$$

4.2.3. Approach to Solving the Nonlinear High Angle of Attack Problem. At the flight conditions described in section 4.2.2, of Mach=0.85 and at 30000 feet, the F-8 stalls when the angle of attack is above 23.5° , which means that the wings start losing lift and thus the airplane cannot sustain flight. Therefore the stall angle of attack of 23.5° becomes the reference point of the training procedure. A series of ordered steps were followed in order to have an organized training procedure. The steps followed are outlined below:

- Simulation of the results from Garrard and Jordan to obtain a benchmark for the work.
- Determine the training range conditions.
- Train the neurocontroller to model the linear controller (discrete-time LQR solution) to control the linear model described in Eq. (140), below the stall conditions (i.e. $\alpha < 23.5^\circ$).
- Train the neurocontroller to model the linear controller (discrete-time LQR solution) to control the nonlinear model described in Eq. (131) controller below the stall conditions.
- Train the neurocontroller to control the nonlinear model below the stall conditions using the Action and Critic NN training procedures described in section 2.3.2.
- Train the neurocontroller to control the nonlinear model above the stall conditions using the Action and Critic NN training procedures described in section 2.3.2.

4.2.4. Testing the Bench Mark Controller. In order to determine the benchmark limits for Garrard and Jordan's controllers, a fixed-step Runge-Kutta 4th order integration routine was written in the MATLAB[®] interface. Several initial angles of attack, $\alpha_{(0)}$, were tested for the controller laws defined in Eqs. (142), (145), and (146), while the initial pitch angle and pitch rate were kept at zero to test the limits of the angle of attack. The time integration step-size used for this simulations was $\Delta t=0.01$ seconds. Figures 4.10-4.14 show the simulation results that test the limits of all three benchmark

controllers for varying initial angles of attack, and keeping at zero the initial pitch angle and pitch rate.

Figure 4.10 shows the simulation results for an initial angle of attack that lays just below the stall region, initial angle of attack of $\alpha_{(0)}=23^\circ$ for zero pitch angle of $\theta_{(0)}=0^\circ$, and pitch rate of $q_{(0)}=0^\circ/\text{sec}$. The figure is divided in 4 subplots, angle of attack and pitch angle in the top left and top right subplots respectively, while the bottom subplots depict the pitch rate and the flight path angle, respectively. The units for the states are degrees ($^\circ$) for the angle of attack, pitch angle and flight path angle, while the pitch rate has units of degrees-per-second ($^\circ/\text{sec}$).

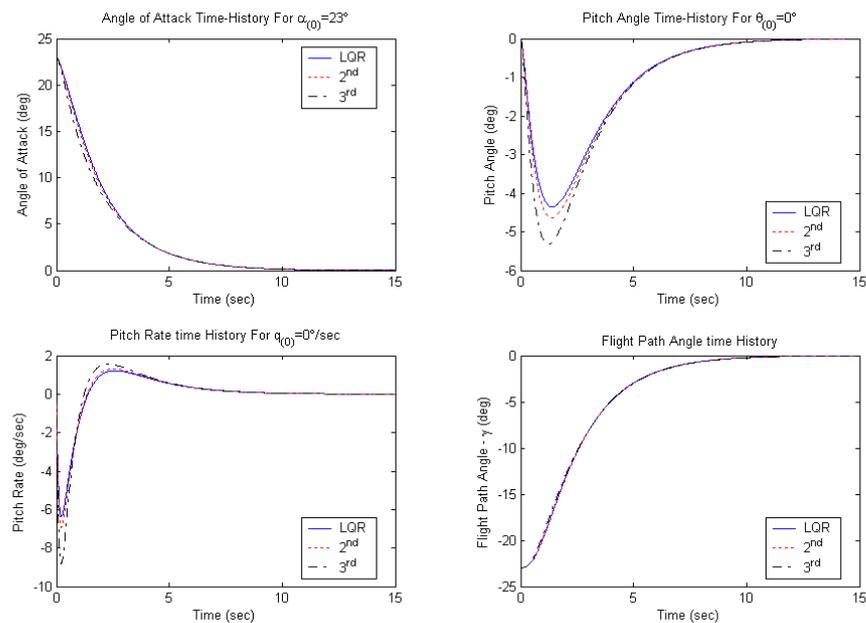


Figure 4.10. States Results ($\alpha_{(0)}=23^\circ$, $\theta_{(0)}=0^\circ$, $q_{(0)}=0^\circ/\text{sec}$).

Figure 4.11 is also divided in 4 subplots and shows the simulation results for the correspondent altitude loss, altitude loss-rate, tail rotation and cost for the above conditions. The altitude loss has units of feet (ft), the altitude loss-rate has units of feet-

per-second (ft/sec), and the tail rotation has units of degrees. Throughout the remainder of this section, the figures describing different initial conditions will hold the same structure and units. The three controllers are compared in the plots, where the solid blue line corresponds to the LQR controller, the dotted red line and dashed black line correspond to the 2nd-Order and 3rd-Order controller proposed by Garrard [85], respectively.

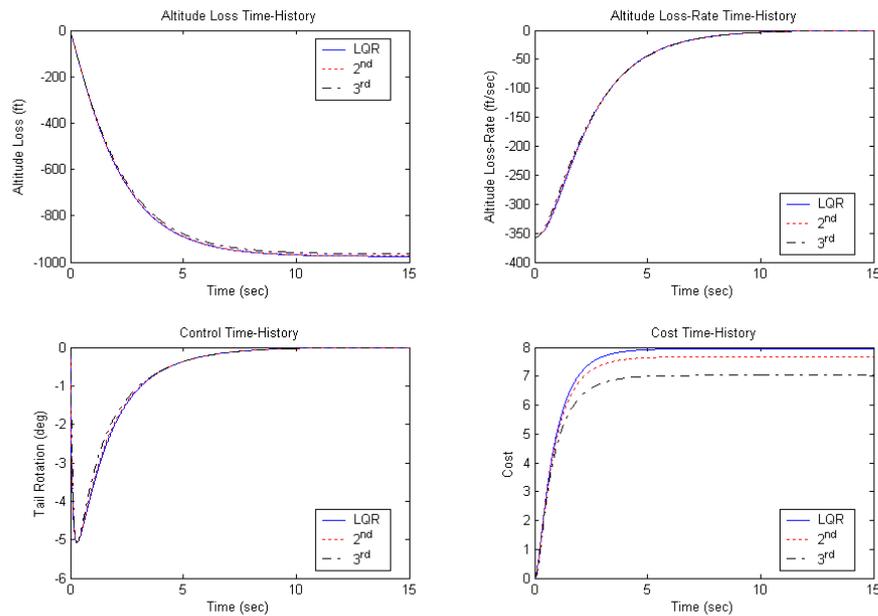


Figure 4.11. Altitude, Control and Cost Results ($\alpha_{(0)}=23^\circ$, $\theta_{(0)}=0^\circ$, $q_{(0)}=0^\circ/\text{sec}$).

In order to aid in the analysis of the results, and to provide the reader with a better understanding of the behavior of the airplane throughout the recovery maneuver from the stall region, the flight path angle and the drop in altitude are introduced in this section, and will be used in the remainder of the thesis. The flight path angle is introduced by using Eq. (45), which was previously introduced in section 4.1.2., to obtain the flight path angle as a linear function of the angle of attack time history and the pitch angle time history such that:

$$\gamma_i = \theta_i - \alpha_i \quad (45)$$

Remember that this is possible only for the zero-sideslip angle case, and since only the longitudinal plane is considered in this study, the assumption is a valid one. The drop in altitude is introduced into the analysis by simplifying Eq. (49) for the longitudinal case such that $V=\Phi=0$, thus yielding Eq. (166), and then substituting Eq. (120) into Eq. (166), yielding Eq. (167), which becomes a simplified differential equation for the altitude for the longitudinal plane of the model described in section 4.2.

$$\dot{h} = U_1 \sin \theta - W \cos \theta \quad (166)$$

$$\dot{h} = U_1 \sin \theta - U_1 \tan \alpha \cos \theta \quad (167)$$

Equation (167) provides an analogous differential equation to the nonlinear problem defined in section 4.2, which can provide an insight of the behavior of the airplane in addition to the flight path angle. Figures 4.12 and 4.13 show the simulation results that, according to the literature [85], represent the initial conditions at which the F-8 Crusader stalls, which correspond to $\alpha_{(0)} = 23.5^\circ$, $\theta_{(0)} = 0^\circ$, and $q_{(0)} = 0^\circ/\text{sec}$. At these initial conditions, all three proposed controllers are able to recover from the stall condition, with the only difference that the 3rd-Order controller has a slightly better performance than the 2nd-Order, which has also a slightly better performance than the LQR controller. This can better be observed when analyzing and comparing the drop in altitude and the cost comparison for all three controllers.

Figures 4.14 and 4.15 show the limiting case for which the LQR controller effectively recovers the airplane from stall, which corresponds to $\alpha_{(0)} = 25.73^\circ$, $\theta_{(0)} = 0^\circ$, and $q_{(0)} = 0^\circ/\text{sec}$. The difference in performance is more accentuated for these initial conditions, and it can be clearly seen in Figures 4.14 and 4.15, that the 3rd-Order controller has better performance than the 2nd-Order, which has also a better performance than the LQR controller. Beyond this initial angle of attack, the LQR controller cannot effectively recover from stall. The initial angle of attack is slowly increased to test the limits for the other two nonlinear controllers, while the LQR controller is disregarded in

the analysis. The flight path angle time-history, located in the lower right corner of Figure 4.14, indicates the loss in altitude of the aircraft for all controllers. The loss in altitude becomes dictated by the negative sign in the flight path angle. The loss in altitude is also reflected in the altitude loss time-history and the altitude loss-rate time history described in Figure 4.15. The 3rd-Order controller provides the smallest loss in altitude relative to the other two controllers since it achieves the zero flight path angle faster than any of the controllers proposed by Garrard and Jordan [85]. It needs to be noted that a negative flight path angle does not imply a dive-nose attitude of the aircraft, since the attitude is determined by the angle of attack and the pitch angle. The flight path angle determines the angle of the path of the aircraft relative to the Earth-Fixed coordinate system, which in return, if the angle is negative describes a descending path or loss in altitude, or an ascending path if positive. Figure 4.15 also shows the control effort of the tail rotation, for all three controllers.

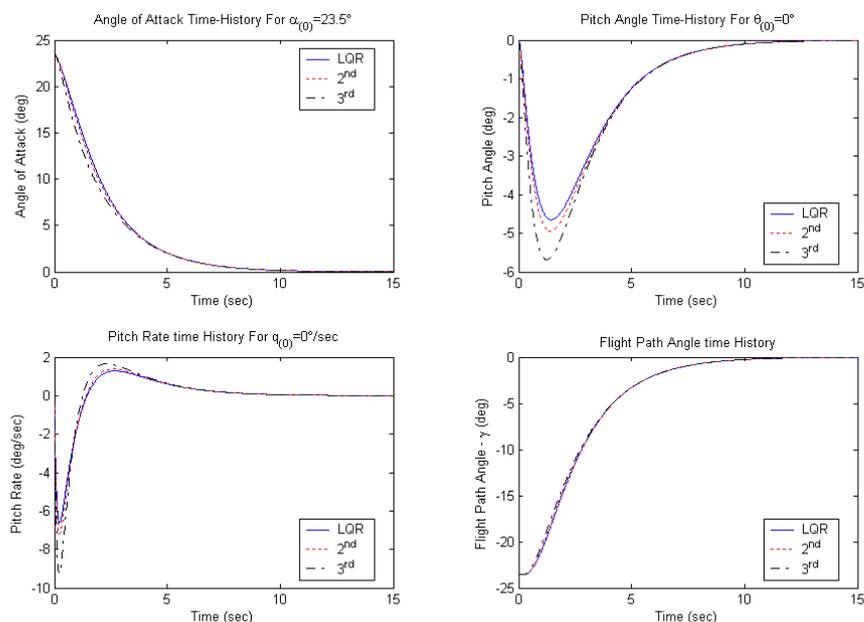


Figure 4.12. States Results ($\alpha_{(0)} = 23.5^\circ$, $\theta_{(0)} = 0^\circ$, $q_{(0)} = 0^\circ/\text{sec}$).

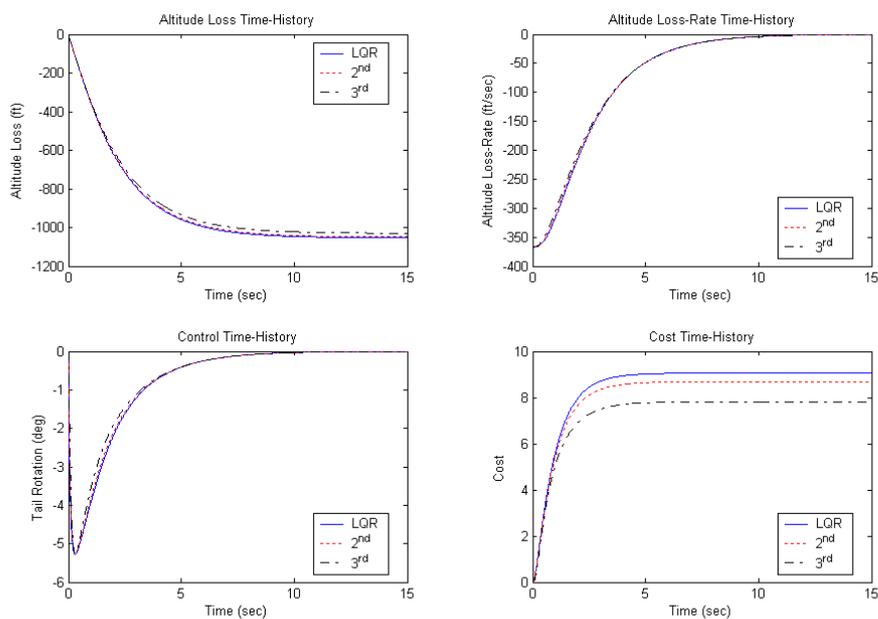


Figure 4.13. Altitude, Control and Cost Results ($\alpha_{(0)} = 23.5^\circ$, $\theta_{(0)} = 0^\circ$, $q_{(0)} = 0^\circ/\text{sec}$).

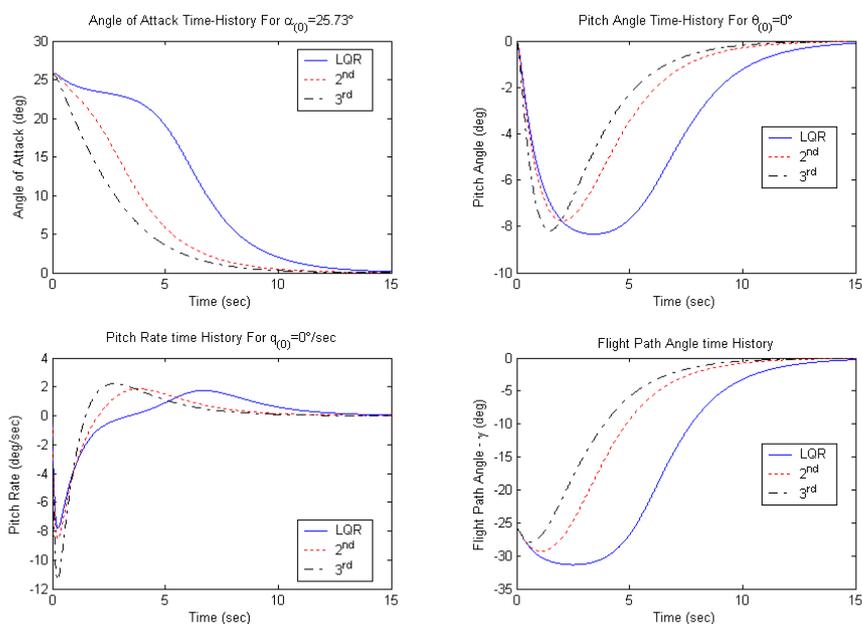


Figure 4.14. States Results ($\alpha_{(0)} = 25.73^\circ$, $\theta_{(0)} = 0^\circ$, $q_{(0)} = 0^\circ/\text{sec}$).

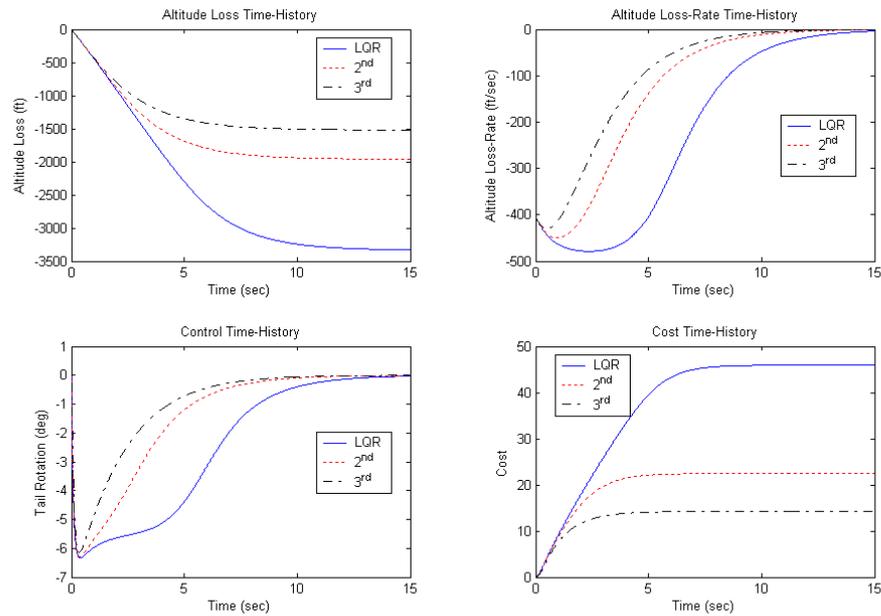


Figure 4.15. Altitude, Control and Cost Results ($\alpha_{(0)}=25.73^\circ$, $\theta_{(0)}=0^\circ$, $q_{(0)}=0^\circ/\text{sec}$).

Figures 4.16 and 4.17 show the limiting case for which the 2nd-Order controller effectively recovers the airplane from stall, which corresponds to an initial angle of attack of $\alpha_{(0)}=25.99^\circ$, $\theta_{(0)}=0^\circ$, and $q_{(0)}=0^\circ/\text{sec}$. Beyond this angle the controller cannot effectively recover from stall. It is noted that the second order controller offers only a small improvement in extending the stall-recovery-region relative to the LQR controller. Despite that both the 2nd-Order and the 3rd-Order reach steady state conditions almost at the same time, the differences in performance are evident in the simulations described in Figures 4.16 and 4.17. While the 3rd-Order controller has a loss in altitude of approximately 1600 ft, the 2nd-Order controller drops almost 4000 ft. The difference is more drastic when comparing the cost as seen in Figure 4.17. Figures 4.18 and 4.19 show the limiting case for which the 3rd-Order controller effectively recovers the airplane from stall, which corresponds to an initial conditions of $\alpha_{(0)}=27.09^\circ$, $\theta_{(0)}=0^\circ$, and $q_{(0)}=0^\circ/\text{sec}$. Beyond these initial conditions, none of the controllers proposed by Garrard and Jordan [85] can effectively recover from stall.

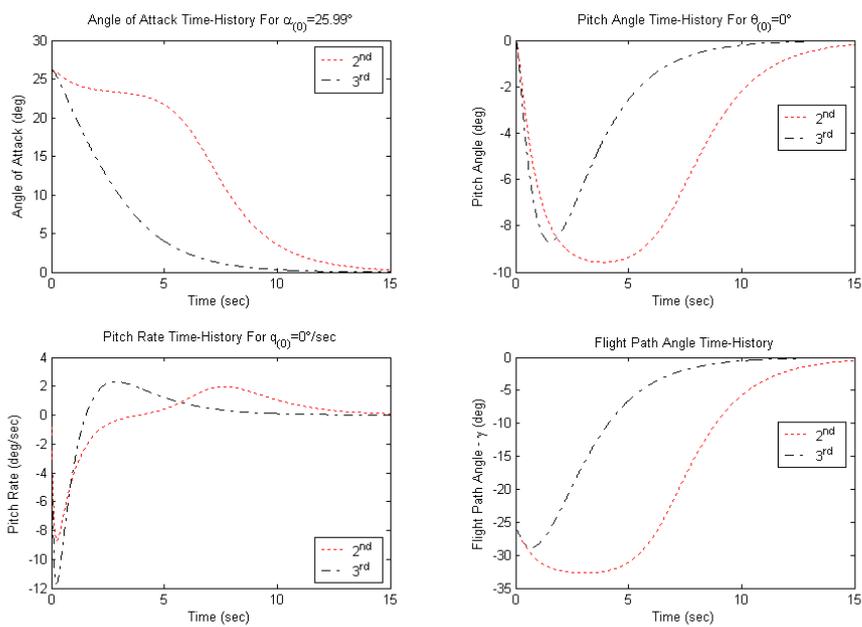


Figure 4.16. States Results ($\alpha_{(0)} = 25.99^\circ$, $\theta_{(0)} = 0^\circ$, $q_{(0)} = 0^\circ/\text{sec}$).

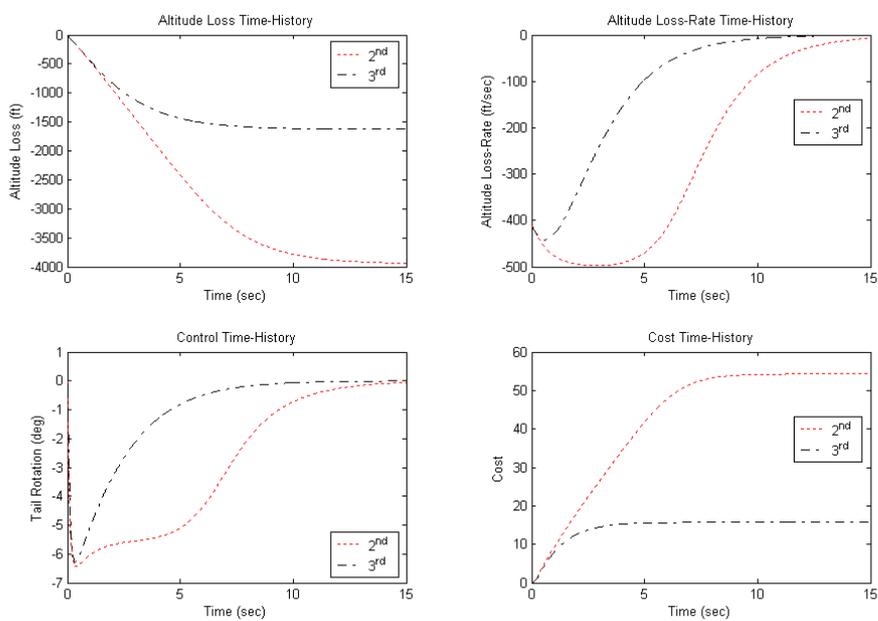


Figure 4.17. Altitude, Control and Cost Results ($\alpha_{(0)} = 25.99^\circ$, $\theta_{(0)} = 0^\circ$, $q_{(0)} = 0^\circ/\text{sec}$).

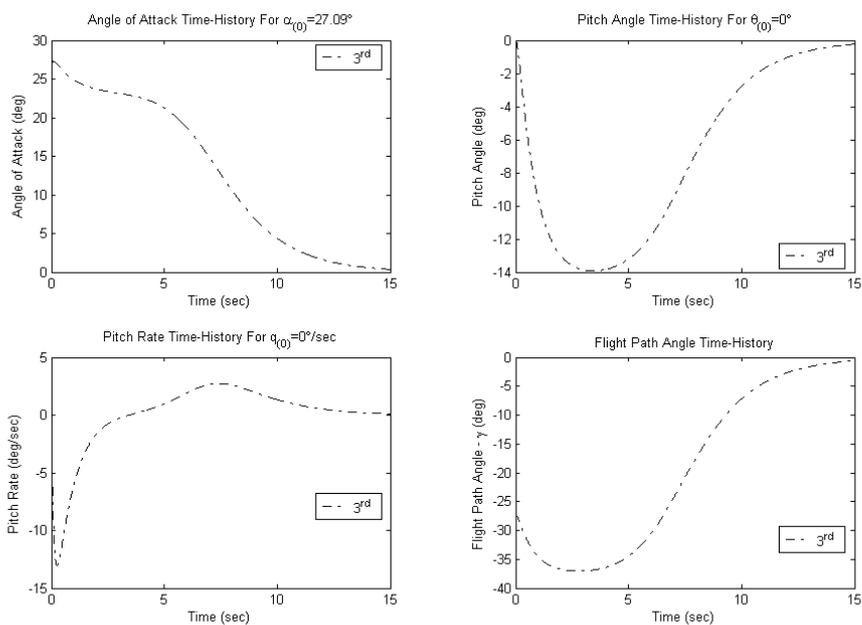


Figure 4.18. States Results ($\alpha_{(0)} = 27.09^\circ$, $\theta_{(0)} = 0^\circ$, $q_{(0)} = 0^\circ/\text{sec}$).

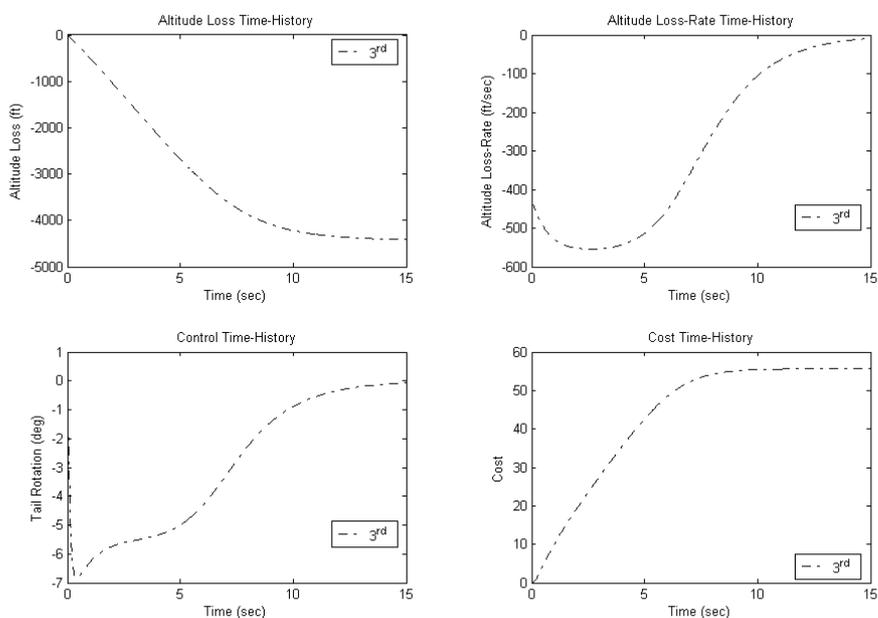


Figure 4.19. Altitude, Control and Cost Results ($\alpha_{(0)} = 27.09^\circ$, $\theta_{(0)} = 0^\circ$, $q_{(0)} = 0^\circ/\text{sec}$).

The limitation in the initial pitch angle and pitch rate to $\theta_{(0)} = 0^\circ$ and $q_{(0)} = 0^\circ/\text{sec}$ respectively, considerably reduces the analysis of the three controllers, therefore a more detailed analysis of the maximum allowable initial angles of attack for the three proposed controllers is conducted by varying the initial pitch angle by $\theta_{(0)} = \pm 20^\circ$, and the initial pitch rate by $q_{(0)} = \pm 20^\circ/\text{sec}$. Tables 4.2, 4.3 and 4.4 provide the maximum allowable initial angle of attack for different initial pitch angles and pitch rates for the three proposed controllers by Garrard and Jordan [85]. The units displayed in Tables 4.2-4.4 are degrees for the angle of attack, pitch angle and tail rotation, while the pitch rate has units of degrees-per-second.

Figures 4.20, 4.21, and 4.22 show the results of the the tabulated tables. Figure 4.23 shows how all three controllers compared with each other as a function of variable initial pitch angle and pitch rate. The top part of Figure 4.23 provides a better understanding of how the 3rd-Order controller, top surface, provides the maximum extension in the initial angles of attack for variable initial pitch angle and pitch rate, while the 2nd-Order controller, second from the top, generates only a slight improvement in the extension of the maximum initial angle of attack, relative to the LQR controller.

Table 4.2. Maximum Allowable Initial Angle of Attack for Different Initial Pitch Angles and Pitch Rates for the LQR Controller.

		Initial Pitch Rate $q_{(0)}$								
		-20°/sec	-15°/sec	-10°/sec	-5°/sec	0°/sec	5°/sec	10°/sec	15°/sec	20°/sec
Initial Pitch Angle $\theta_{(0)}$	-20°	19.37°	19.26°	19.14°	19.02°	18.89°	18.77°	18.65°	18.51°	18.38°
	-15°	21.56°	21.43°	21.31°	21.17°	21.02°	20.9°	20.76°	20.62°	20.47°
	-10°	23.41°	23.27°	23.13°	22.99°	22.84°	22.7°	22.55°	22.4°	22.25°
	-5°	24.97°	24.83°	24.69°	24.55°	24.39°	24.24°	24.09°	23.94°	23.79°
	0°	26.32°	26.18°	26.03°	25.88°	25.73°	25.58°	25.43°	25.27°	25.11°
	5°	27.49°	27.34°	27.2°	27.05°	26.9°	26.75°	26.59°	26.44°	26.28°
	10°	28.51°	28.37°	28.22°	28.07°	27.92°	27.77°	27.62°	27.47°	27.31°
	15°	29.42°	29.28°	29.13°	28.98°	28.84°	28.69°	28.54°	28.38°	28.23°
	20°	30.23°	30.09°	29.95°	29.8°	29.65°	29.51°	29.36°	29.21°	29.06°

Table 4.3. Maximum Allowable Initial Angle of Attack for Different Initial Pitch Angles and Pitch Rates for the 2nd Order Controller.

		Initial Pitch Rate $q(0)$								
		-20°/sec	-15°/sec	-10°/sec	-5°/sec	0°/sec	5°/sec	10°/sec	15°/sec	20°/sec
Initial Pitch Angle $\theta(0)$	-20°	19.97°	19.83°	19.69°	19.56°	19.43°	19.29°	19.15°	19.01°	18.87°
	-15°	22.04°	21.9°	21.76°	21.63°	21.48°	21.34°	21.2°	21.06°	20.92°
	-10°	23.81°	23.66°	23.51°	23.37°	23.2°	23.05°	22.9°	22.75°	22.6°
	-5°	25.3°	25.15°	25°	24.86°	24.72°	24.57°	24.42°	24.27°	24.12°
	0°	26.61°	26.45°	26.29°	26.14°	25.99°	25.83°	25.67°	25.51°	25.35°
	5°	27.72°	27.56°	27.4°	27.25°	27.1°	26.94°	26.78°	26.62°	26.46°
	10°	28.7°	28.54°	28.38°	28.23°	28.08°	27.92°	27.76°	27.6°	27.44°
	15°	29.55°	29.4°	29.25°	29.11°	28.96°	28.81°	28.66°	28.51°	28.36°
	20°	30.33°	30.18°	30.03°	29.89°	29.74°	29.59°	29.44°	29.29°	29.14°

Table 4.4. Maximum Allowable Initial Angle of Attack for Different Initial Pitch Angles and Pitch Rates for 3rd Order Controller.

		Initial Pitch Rate $q(0)$								
		-20°/sec	-15°/sec	-10°/sec	-5°/sec	0°/sec	5°/sec	10°/sec	15°/sec	20°/sec
Initial Pitch Angle $\theta(0)$	-20°	21.77°	21.65°	21.52°	21.39°	21.26°	21.12°	20.98°	20.85°	20.71°
	-15°	23.66°	23.52°	23.38°	23.24°	23.1°	22.96°	22.81°	22.66°	22.51°
	-10°	25.24°	25.09°	24.95°	24.8°	24.65°	24.5°	24.35°	24.2°	24.04°
	-5°	26.57°	26.42°	26.27°	26.12°	25.97°	25.81°	25.66°	25.5°	25.34°
	0°	27.7°	27.55°	27.4°	27.24°	27.09°	26.93°	26.78°	26.62°	26.46°
	5°	28.67°	28.52°	28.37°	28.21°	28.06°	27.9°	27.74°	27.58°	27.42°
	10°	29.51°	29.36°	29.21°	29.06°	28.9°	28.75°	28.59°	28.43°	28.27°
	15°	30.25°	30.1°	29.95°	29.8°	29.65°	29.5°	29.34°	29.18°	29.02°
	20°	30.91°	30.76°	30.61°	30.46°	30.31°	30.16°	30°	29.85°	29.69°

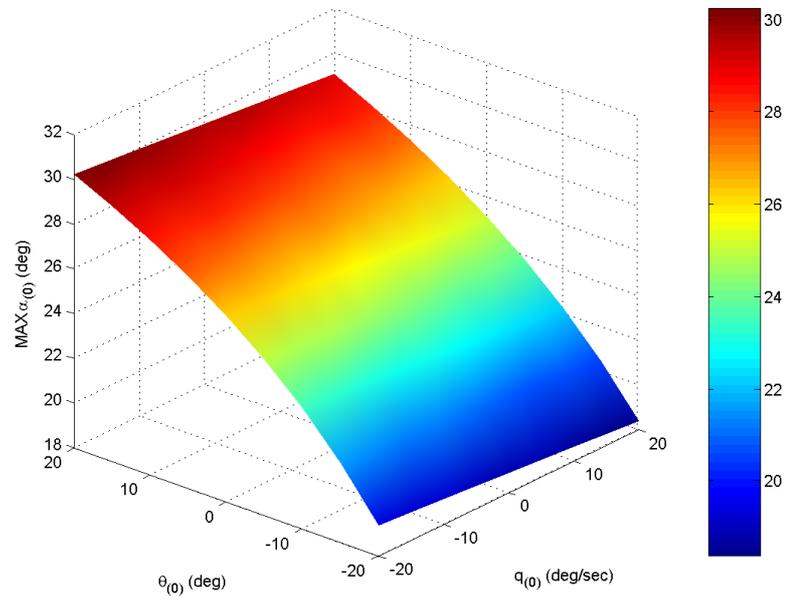


Figure 4.20. Maximum Allowable Initial Angle of Attack for Different Initial Pitch Angles and Pitch Rates for the LQR Controller.

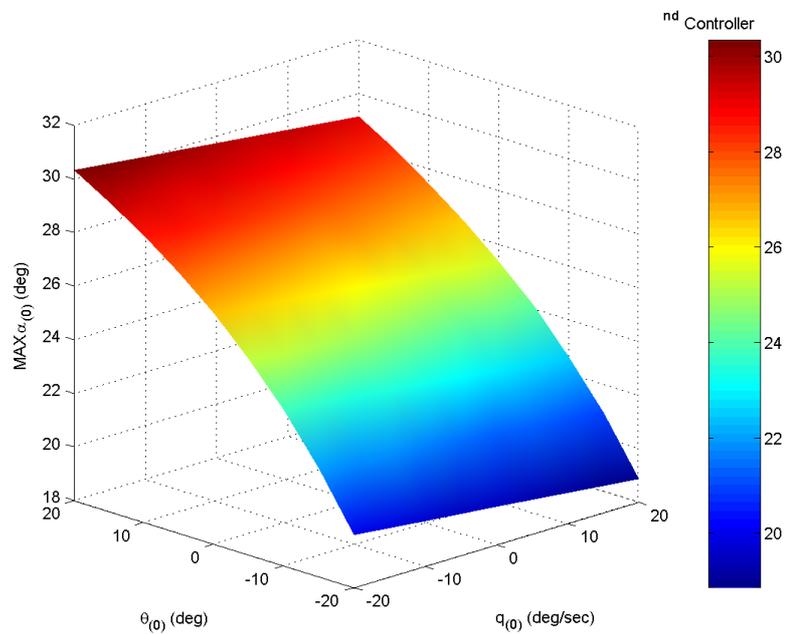


Figure 4.21. Maximum Allowable Initial Angle of Attack for Different Initial Pitch Angles and Pitch Rates for the 2nd-Order Controller.

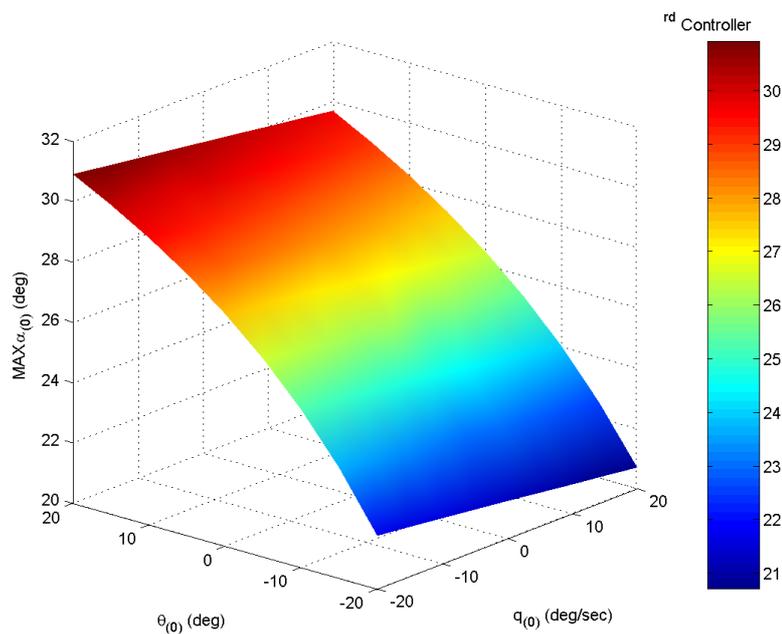


Figure 4.22. Maximum Allowable Initial Angle of Attack for Different Initial Pitch Angles and Pitch Rates for the 3rd-Order Controller.

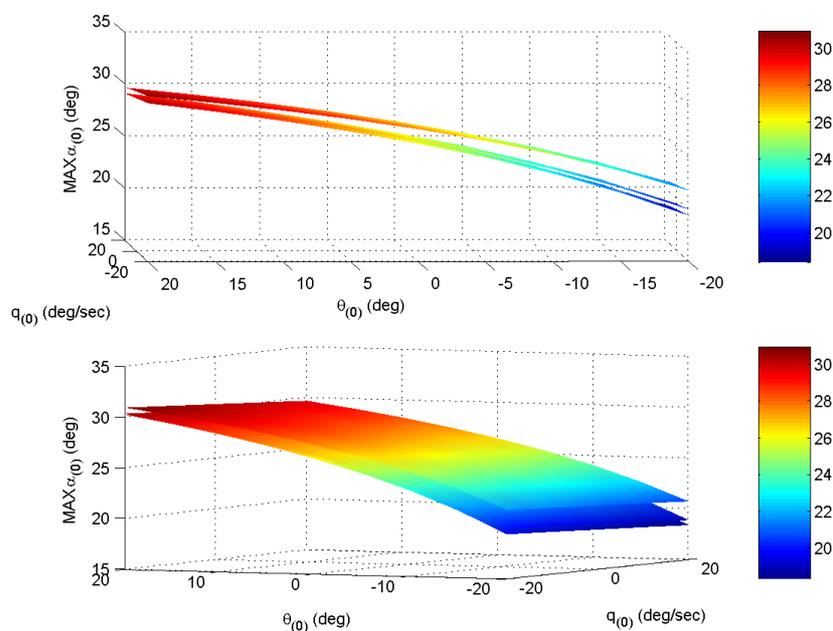


Figure 4.23. Maximum Allowable Initial Angle of Attack for Different Initial Pitch Angles and Pitch Rates Comparison of the 3 Controller.

Simulations for the maximum initial allowable angles of attack for the three proposed controllers, listed in Tables 4.2-4.4, were conducted to observe the behavior of the states. Only few simulations for the 3rd order controller will be shown and discussed here to simplify the discussion. These simulations will be shown in Figures 4.24 through 4.38. Figures 4.24 through 4.28 show the time history simulations for the angle of attack, pitch angle, pitch rate, flight path angle and drop in altitude respectively, corresponding to the maximum initial values corresponding to the first column in Table 4.4, that is the maximum initial angles of attack at which the 3rd-Order controller can safely recover the airplane from the stall region, which corresponds to the variable pitch angles, $\theta_{(0)}=\pm 20^\circ$ and for a pitch rate of $q_{(0)}=-20^\circ/\text{sec}$.

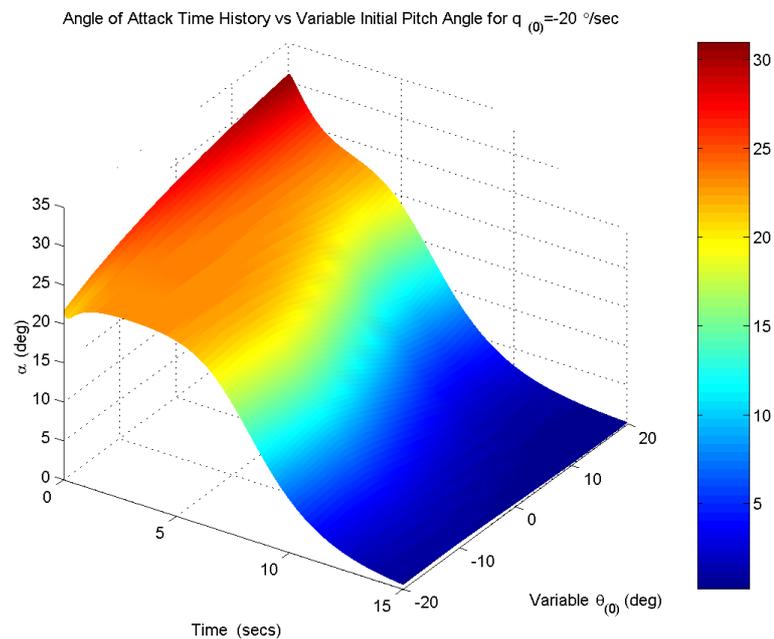


Figure 4.24. Angle of Attack Time History for the 3rd-Order Controller ($\theta_{(0)}=\pm 20^\circ$ and $q_{(0)} = -20^\circ/\text{sec}$).

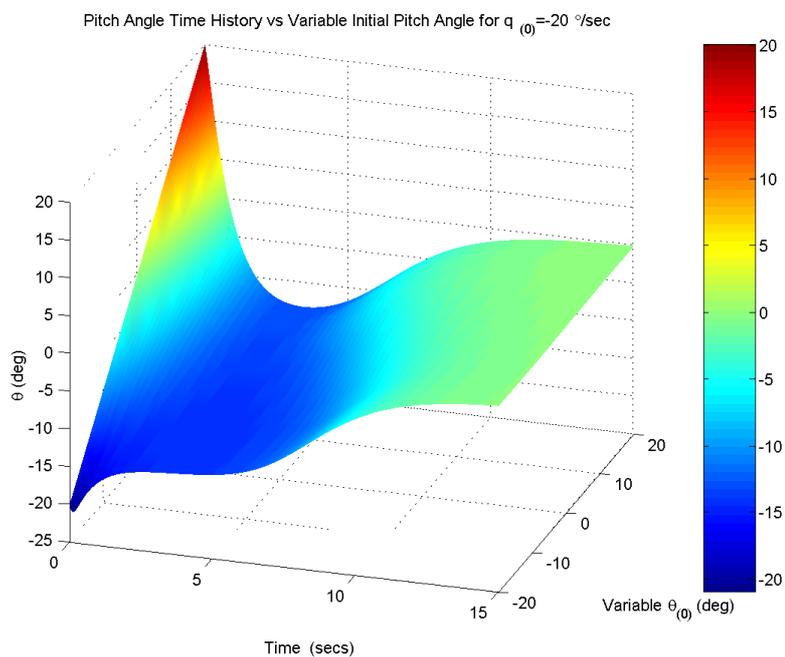


Figure 4.25. Pitch Angle Time History for the 3rd-Order Controller ($\theta_{(0)} = \pm 20^\circ$ and $q_{(0)} = -20^\circ/\text{sec}$).

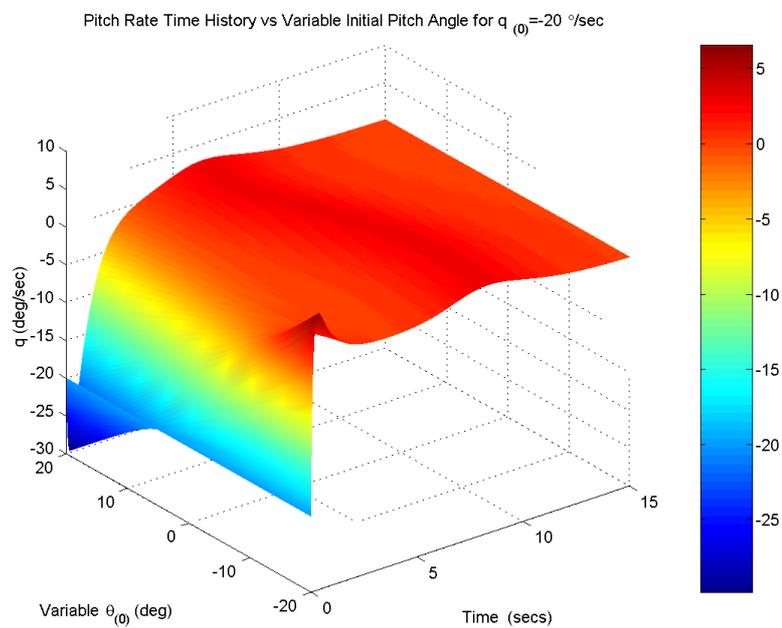


Figure 4.26. Pitch Rate Time History for the 3rd-Order Controller ($\theta_{(0)} = \pm 20^\circ$ and $q_{(0)} = -20^\circ/\text{sec}$).

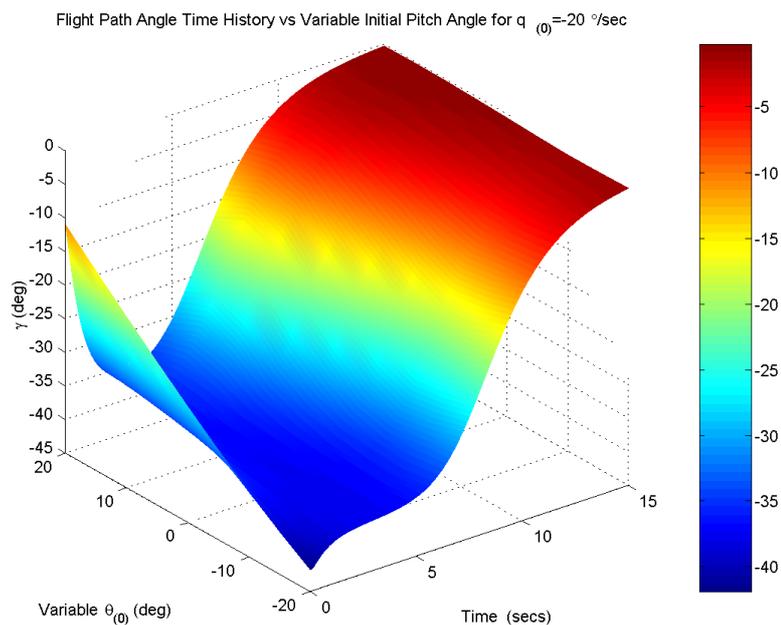


Figure 4.27. Flight Path Angle Time History for the 3rd-Order Controller ($\theta_{(0)} = \pm 20^\circ$ and $q_{(0)} = -20^\circ/\text{sec}$).

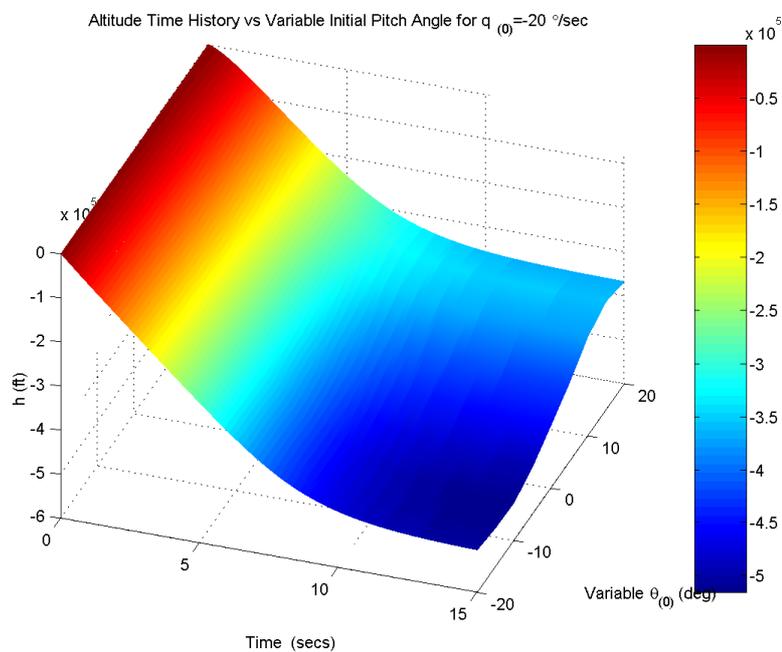


Figure 4.28. Drop in Altitude Time History for the 3rd-Order Controller ($\theta_{(0)} = \pm 20^\circ$ and $q_{(0)} = -20^\circ/\text{sec}$).

Figures 4.29 through 4.33 show the time history simulations for the angle of attack, pitch angle, pitch rate, flight path angle and drop in altitude respectively, corresponding to the maximum initial values corresponding to the fifth column in Table 4.4, that is the maximum initial angles of attack at which the 3rd-Order controller can safely recover the airplane from the stall region, which corresponds to the variable pitch angles, $\theta_{(0)} = \pm 20^\circ$ and for a pitch rate of $q_{(0)} = 0^\circ/\text{sec}$.

Figures 4.34 through 4.38 show the time history simulations for the angle of attack, pitch angle, pitch rate, flight path angle and drop in altitude respectively, corresponding to the maximum initial values corresponding to the ninth column in Table 4.4, that is the maximum initial angles of attack at which the 3rd-Order controller can safely recover the airplane from the stall region, which corresponds to the variable pitch angles, $\theta_{(0)} = \pm 20^\circ$ and for a pitch rate of $q_{(0)} = 20^\circ/\text{sec}$.

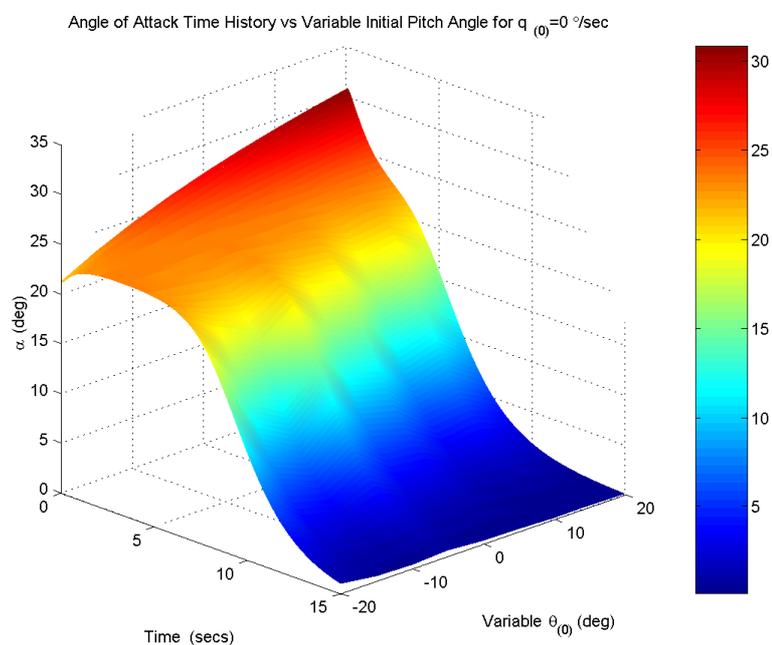


Figure 4.29. Angle of Attack Time History for the 3rd-Order Controller ($\theta_{(0)} = \pm 20^\circ$ and $q_{(0)} = 0^\circ/\text{sec}$).

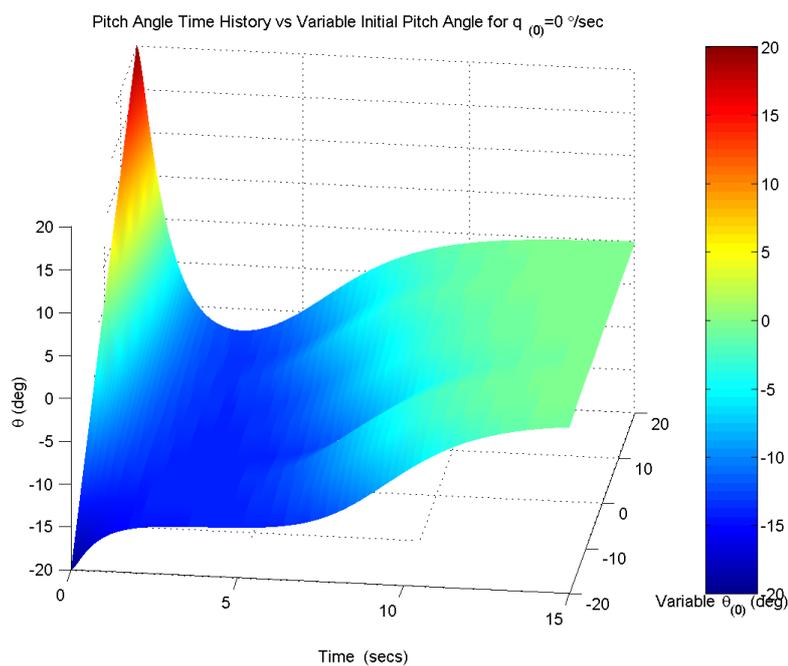


Figure 4.30. Pitch Angle Time History for the 3rd-Order Controller ($\theta_{(0)}=\pm 20^\circ$ and $q_{(0)}=0^\circ/\text{sec}$).

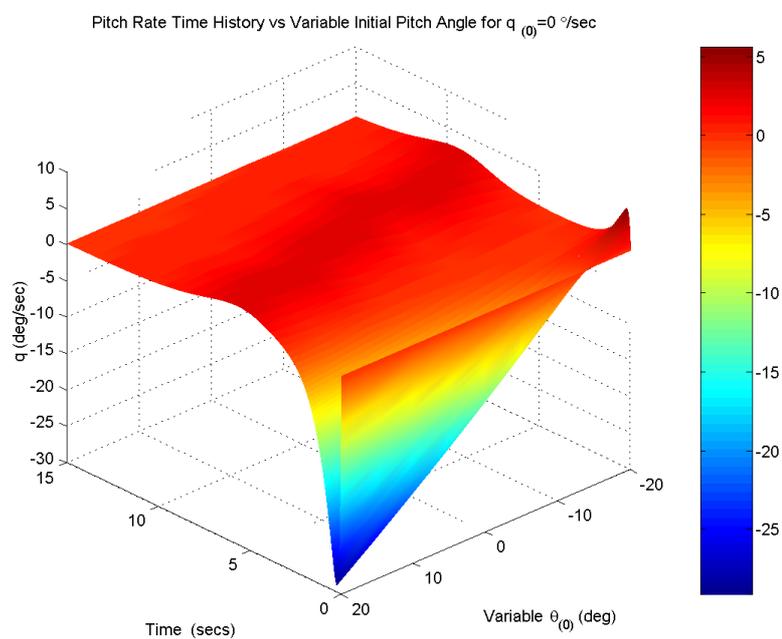


Figure 4.31. Pitch Rate Time History for the 3rd-Order Controller ($\theta_{(0)}=\pm 20^\circ$ and $q_{(0)}=0^\circ/\text{sec}$).

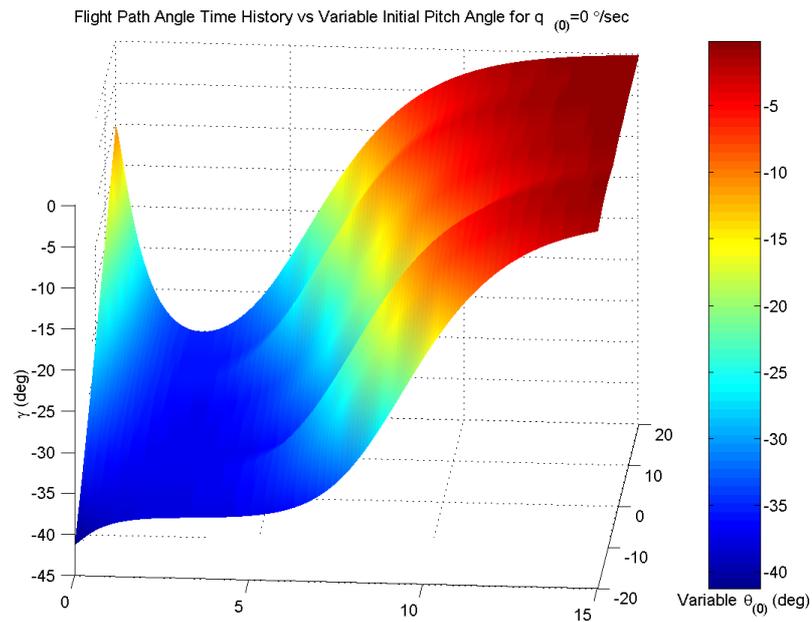


Figure 4.32. Flight Path Angle Time History for the 3rd-Order Controller ($\theta_{(0)}=\pm 20^\circ$ and $q_{(0)}=0^\circ/\text{sec}$).

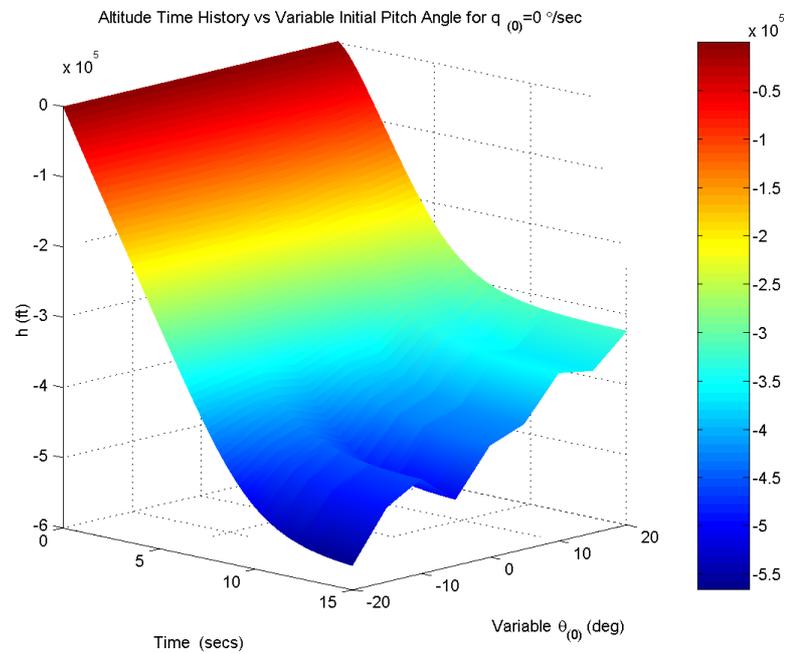


Figure 4.33. Drop in Altitude Time History for the 3rd-Order Controller ($\theta_{(0)}=\pm 20^\circ$ and $q_{(0)}=0^\circ/\text{sec}$).

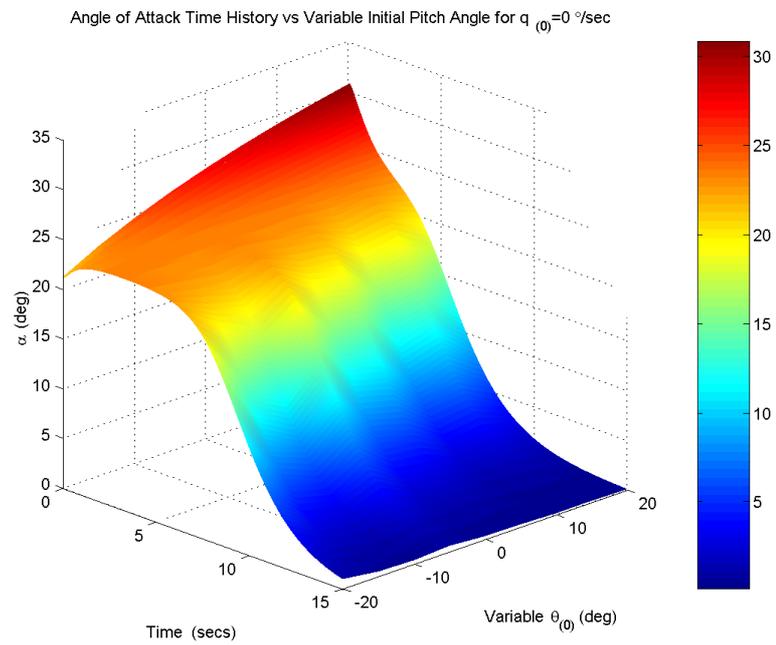


Figure 4.34. Angle of Attack Time History for the 3rd-Order Controller ($\theta_{(0)}=\pm 20^\circ$ and $q_{(0)}=20^\circ/\text{sec}$).

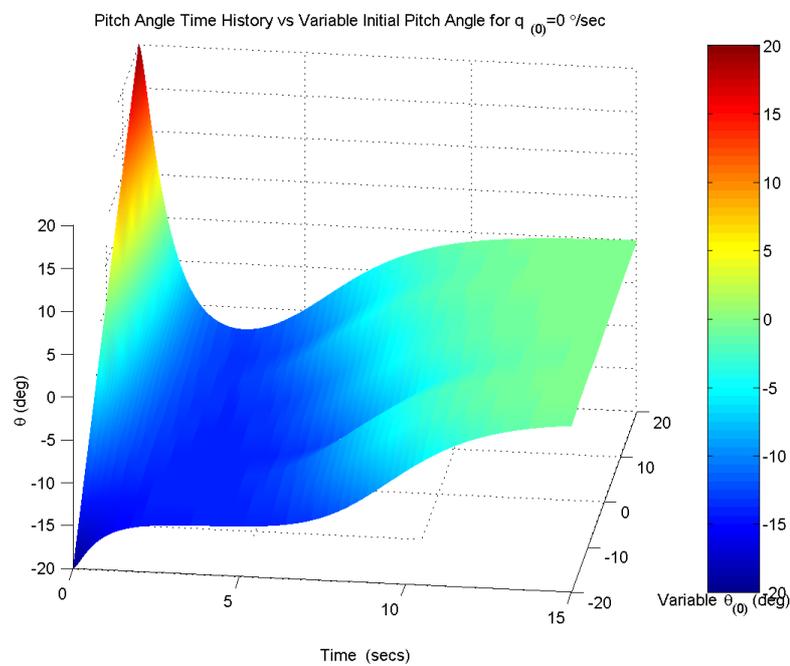


Figure 4.35. Pitch Angle Time History for the 3rd-Order Controller ($\theta_{(0)}=\pm 20^\circ$ and $q_{(0)}=20^\circ/\text{sec}$).

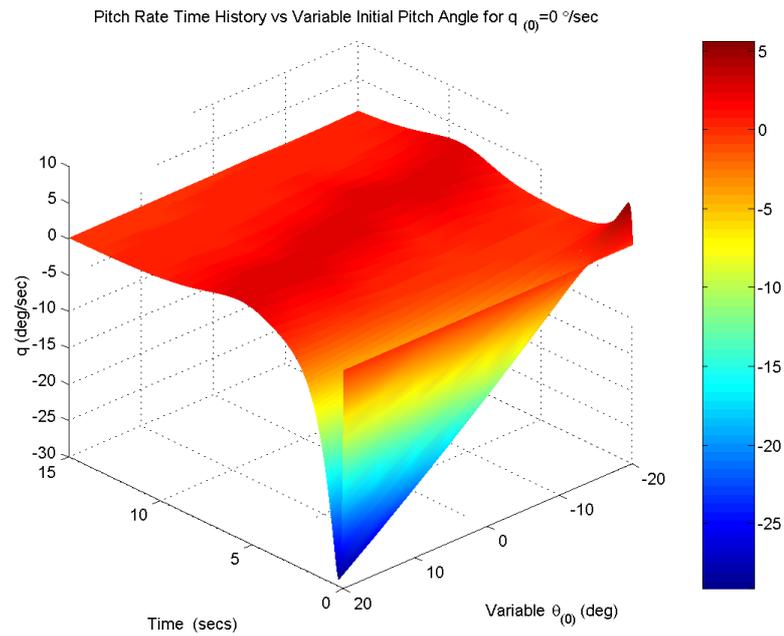


Figure 4.36. Pitch Rate Time History for the 3rd-Order Controller ($\theta_{(0)}=\pm 20^\circ$ and $q_{(0)}=20^\circ/\text{sec}$).

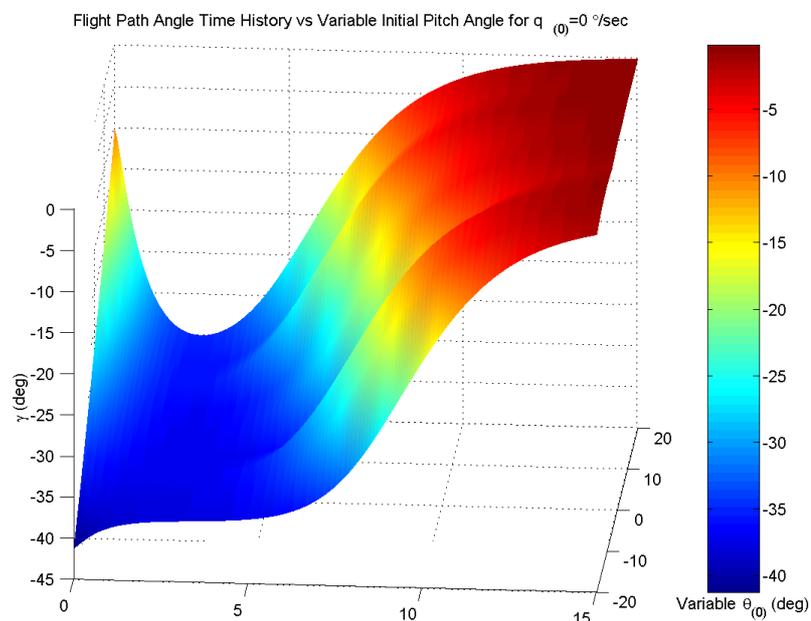


Figure 4.37. Flight Path Angle Time History for the 3rd-Order Controller ($\theta_{(0)}=\pm 20^\circ$ and $q_{(0)}=20^\circ/\text{sec}$).

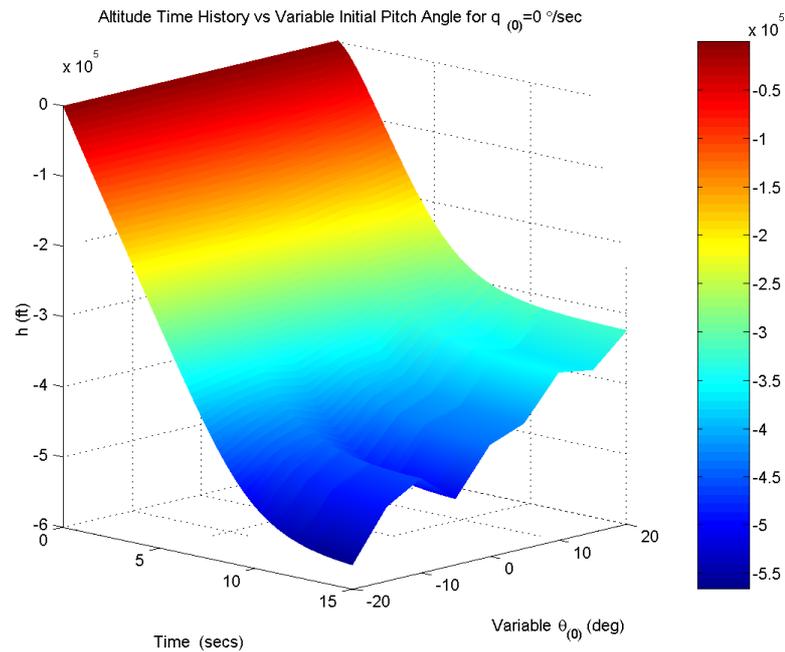


Figure 4.38. Drop in Altitude Time History for the 3rd-Order Controller ($\theta_{(0)}=\pm 20^\circ$ and $q_{(0)}=20^\circ/\text{sec}$).

4.2.5. Neural Network Analysis. This section defines the NN architecture for the nonlinear problem, the NN initial training approach, and the limits for the trained Dual-NN.

4.2.5.1 Neural network architecture. The neurocontroller development approach, described in section 3.1, the ACNN formulation used in this study is a feed forward backpropagation architecture, consisting of two hidden layers with hyperbolic tangent sigmoid transfer function, and an output layer with a linear transfer function. The ANN architecture for the synthesis of the nonlinear tracker problem is defined by $N_{3,4,4,1}$ where the first subscript represents the three inputs to the NN, angle of attack, pitch angle and pitch rate, the second and third subscripts represent the number of neurons, four in each of the hidden layers, and the last subscript represents the single output of the Action, the control or tail rotation. Similarly, the architecture of the CNN is defined by $N_{3,6,6,3}$ where the first subscript represents the three inputs to the NN, again angle of attack, pitch angle and pitch rate, the second and third subscripts represent the number of neurons in

the hidden layers and the last subscript represents the three outputs for the Critic, the co-state or Lagrange multipliers associated to the three states.

The complexity in mapping the co-states for the CNN makes it necessary to have a more complex NN architecture, thus the difference in the number of neurons between the Action and the Critic NN. It needs to be noted that the number of neurons chosen for this work is not the only possibility, and not the optimal number of neurons either, but it was chosen trying to minimize the computational time required for the training process. Despite that, in general for nonlinear mappings, the more the neurons in the hidden layers the better the approximation that can be made, the amount of computational complexity and time required to train grows exponentially with the number of neurons per layer. It also needs to be noted that if the numbers of neurons per layer exceeds the maximum number of points required to map the nonlinear function, then aliasing might produce errors in the mapping. In order to have a feeling of how much more complex can the computations become, the reader can look at Eqs. (25) and (26), described in section 3.1, which describe the transfer function and the output for both the Action and Critic NN. For example, the ANN output is defined as:

$$a_{1x1} = W_{1x4}^3 f^2(W_{4x4}^2 f^1(W_{4x3}^1 p_{3x1} + b_{4x1}^1) + b_{4x1}^2) + b_{1x1}^3 \quad (168)$$

where p_{3x1} is the input vector, corresponding to the three states, and the output corresponding to the tail rotation is defined by a_{1x1} . The subscripts in the weights and biases denote their dimensions. Figure 4.39 describes the NN architectures for the ANN in more detail. Note that the first and second layers have 4 neurons each, denoted by circles, while the output layer has only one neuron. Each neuron is interconnected with each other, therefore as the number of neurons are increased the connections increase as well. To better understand this complexity, Figure 4.40 describes in more detail the structure of each one of the neurons, represented by circles in Figure 4.39, by expanding the 1st hidden layer.

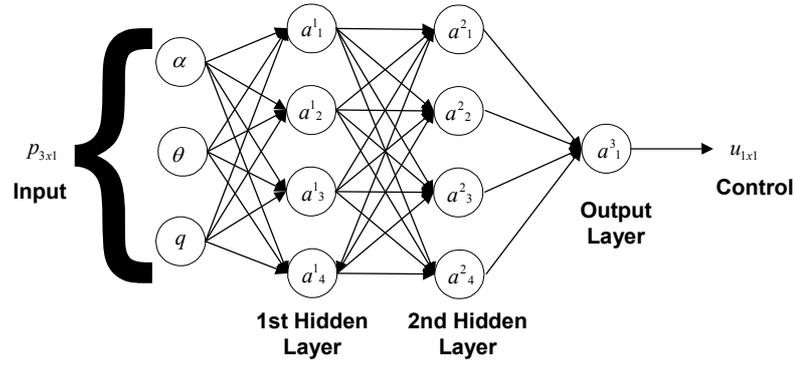


Figure 4.39. Expansion of the Action NN.

The weight and bias vector for this 1st hidden layer is defined bellow to aid understanding the complexity of the model

$$W^1 = \begin{bmatrix} W_{11}^1 & W_{12}^1 & W_{13}^1 & W_{14}^1 \\ W_{21}^1 & W_{22}^1 & W_{23}^1 & W_{24}^1 \\ W_{31}^1 & W_{32}^1 & W_{33}^1 & W_{34}^1 \end{bmatrix} \text{ and } b^1 = [b_1^1 \quad b_2^1 \quad b_3^1 \quad b_4^1]^T$$

Analyzing Figure 4.40, it can be observed that the output from the first neuron of the first hidden layer is defined by the activation function described in Eq.(25)

$$a_1^1 = \frac{e^{n_1^1} - e^{-n_1^1}}{e^{n_1^1} + e^{-n_1^1}} \quad (25)$$

where n_1^1 is the input to the activation function which is defined as:

$$n_1^1 = W_{11}^1 \alpha + W_{12}^1 \theta + W_{13}^1 q + b_1^1 \quad (169)$$

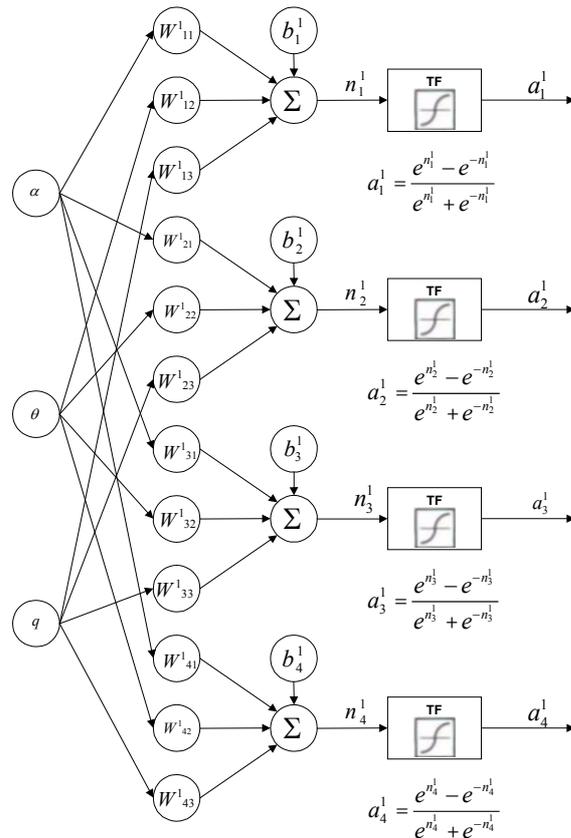


Figure 4.40. Expanded View of the 1st Hidden Layer for the Action NN.

4.2.5.2 Neural network initial training approach. After careful analysis of the results from the system simulation it was determined that the training ranges would start from -5° to 30° for the angle of attack, $\pm 20^\circ$ for the pitch attitude angle, and $\pm 25^\circ/\text{sec}$ for the pitch rate. The first phase of the training procedure for the ANN and the CNN was to initialize both networks. In order to assure the convergence of the training procedure, instead of using a random initialization, it was decided to train both NN to map a percentage of the optimal linear solution for the linear model described in Eq. (164). By initializing the NN using targets as a percentage of the optimal solution, it was assured that the initial NN would be in the direction of the optimal linear problem, thus being stable. At the same time, by not providing the NN with the exact optimal solution, it was assured that the NN was not biased with the LQR solution. Once the mapping

initialization was achieved, the linear model was substituted by the nonlinear model described in Eq. (131). The initial percentage used for the mapping initialization was 80 so that the Action NN network would map a control of the form

$$\delta_{e,NN,i} = -\Delta Kx_i \quad (170)$$

where Δ represents the percentage of the direction, for this training procedure being $\Delta = 0.80$. Figure 4.41 shows the schematics for the initialization of the ANN. The inputs are the three states and the output was the control associated to the three states.

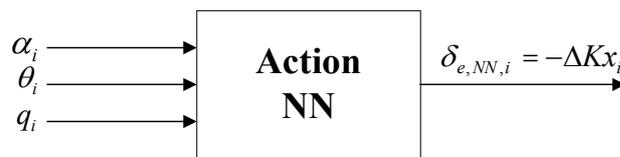


Figure 4.41. Initialization of the Action Neural Network.

The Critic was initialized in the same manner using the solution to the steady state Riccati equation, where the optimal linear co-state history is defined by:

$$\lambda_i = Sx_i \quad (171)$$

where S is the solution to the steady state Riccati Eq. (22). The Critic NN network was initialized such that it would map the Lagrange multipliers as a percentage of the optimal linear solution as it is shown in Figure 4.42.

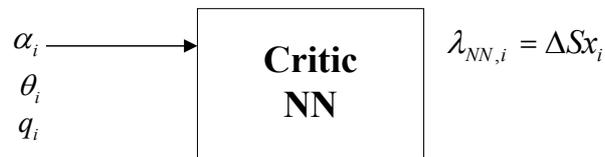


Figure 4.42. Initialization of the Critic Neural Network.

The sets of training inputs were three vectors of 2000 points each. Each one of the three vectors had equally distributed points over the entire range of the respective state, that is for example for the vector corresponding to the angle off attack, 2000 equally distributed points over the range of -5° to $+30^\circ$ were generated. Once the three vectors were generated, they were randomized before being feed as the inputs to the training of the Action and Critic NN to increase their training capabilities.

After the initialization was finished the training procedure followed the training steps outlined in sections 3.2.1 and 3.2.2 for the Action NN and Critic NN respectively. In order to avoid that the Action and Critic NN would not be able to map the characteristics at one region or another of the training range, a telescoping training procedure was conducted, that is, the training procedure was started around the origin of the angle of attack, pitch angle and pitch rate, and after each successful training iteration, the training range was slowly increased so that the controllers would be able to map the origin area. For example, the initial training range set for the angle of attach started from $\pm 5^\circ$, and after each successful training iteration in which there was no significant change between the outputs of the Action and Critic NN, the which the NN were considered to be suboptimal for those ranges, then the ranges were progressively extended until reaching the desired training ranges. It is important to note that as the range in the angle of attack was increased, more points were dedicated to the area surrounding the origin such that the telescoping training was more efficient.

Despite this telescoping training, when the NN feedback controllers were tested in simulations it was observed that despite the NN was able to considerably extend the stall region, there seem to be a small steady state error in the pitch angle such that the

controller would not completely drive the pitch angle to zero, being that steady state error of about 1 degree. This can be seen the top-right figure in Figure 4.43.

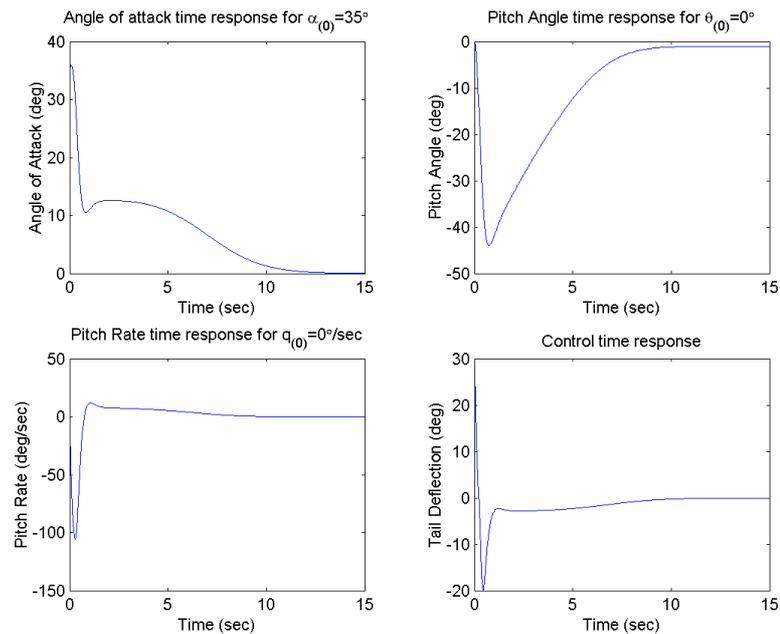


Figure 4.43. Demonstration of Pitch Angle Steady State Error.

Although the steady-state error in pitch angle is really small and the controller, is able to effectively bring the airplane to zero angle of attack and zero pitch rate, it is inadmissible to accept that steady-state error. A non-zero steady state value of the pitch angle means that the flight path angle is not, and according to Eq. (45), which in return means, that the airplane would not be able to hold altitude, and would loose or increase altitude, depending if the flight path angle was negative or positive respectively.

After careful study and analysis of the NN, it was determined that the problem was caused by the fact that while extending the range of initial angle of attack, pitch angle and pitch rate, despite the telescoping training procedure, and that more points were allocated to the center region as the range for the training states were increased, the

resulting NN lacked the necessary mapping properties near the origin, which would assure a zero flight path angle steady state. This had a quick and easy fix, which considering the enormous amount of training and simulation time, was favored instead of doing all the training again increasing the numbers of neurons in the NN to avoid this problem.

The fix consisted of training a second NN around the origin for all three states and when the angle of attack was near the origin, then the feedback controller would switch from the 1st training ANN to the 2nd ANN, thus using a Dual-NN architecture. The value of angle of attack at which the feedback controller would be activated was determined to be 2°, which shows again that the error was minimal, but again was also inadmissible to allow the pitch angle to go to a steady state error despite of how small it might have been. Figure 4.44 shows the results for the Dual-NN feedback controller.

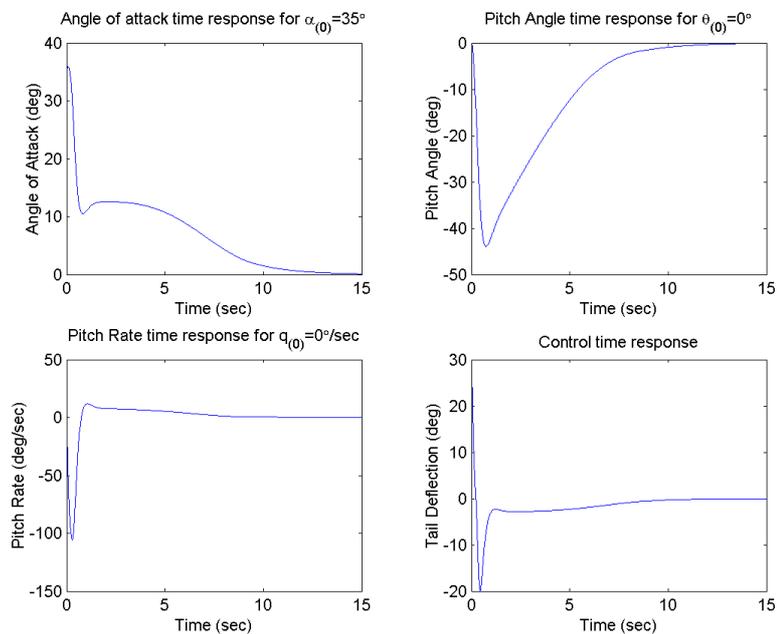


Figure 4.44. Fixed Pitch Angle Steady State Error

In order to improve the overall training process, during the training of the Action and Critic NN, the new target inputs, $\delta_{e,NEW}^*$ and λ_{NEW}^* respectively, were not used in their totality to adjust the new targets. In order to avoid that the new target vectors used to train the NN differ considerably from the previous training network, and thus avoiding a huge change in the gradient of the direction towards the optimal solution, learning rates are introduced so that only a percentage of the new target vector is used to provide the correct direction to the optimal solution. These learning rates are modeled as such that the new target directions are provided by:

$$\delta_{e,NEW}^* = (1 - \gamma_1)\delta_{e,OLD} + \gamma_1\delta_e^* \quad (172)$$

$$\lambda_{NEW}^* = (1 - \gamma_2)\lambda_{OLD} + \gamma_2\lambda^* \quad (173)$$

where $\delta_{e,OLD}^*$ and λ_{OLD}^* represent the old targets inputs for the Action and Critic NN respectively, where γ_1 and γ_2 are the learning rates for the Action and Critic NN respectively that determine the percentage of the optimal target that is given to the next training iteration. They have values of $\gamma_1 < 1$ and $\gamma_2 < 1$, which does not necessarily mean that $\gamma_1 = \gamma_2$ since the Action and the Critic will need different learning rates in order to avoid arriving to a local minima. The values for the learning rates were varied during the training procedure depending the convergence results of the training iterations. Figure 4.45 is a modification of Figure 3.3 previously described in section 3.2, which includes the learning rates for the training targets.

As described in section 4.2.3, Steps 1 through 5 for the Action NN and Critic NN are repeated until the desired level of accuracy for the NNs is achieved and there is no acceptable change in the outputs of the trained NN after a cycle. For this problem the level of accuracy is reached when the 2-norm of the errors between the current outputs and the previous trained outputs is below $e < \varepsilon = 0.008$. Once the training was considered finished, the ANN was tested as a feedback controller of the form described in Figure 4.46.

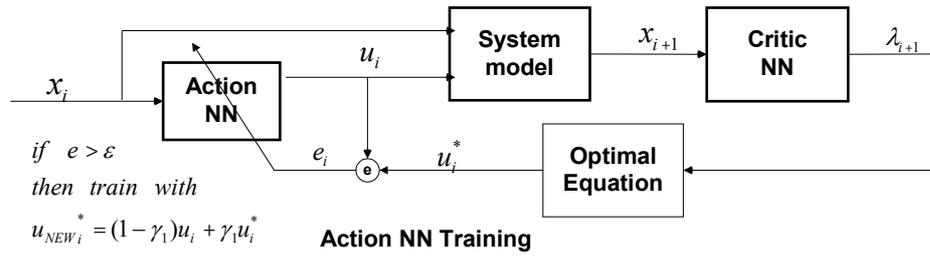


Figure 4.45. Action NN with Modified Training Target.

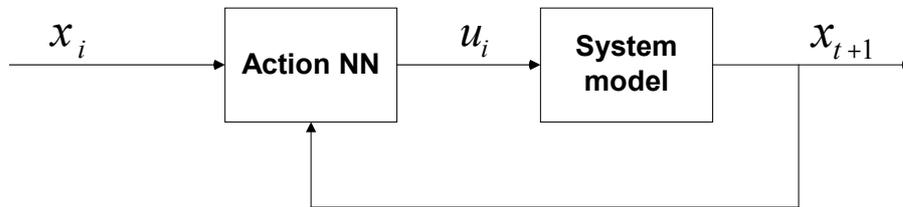


Figure 4.46. Feedback Neurocontroller Implementation.

4.2.5.3 Trained neural networks limits. Once the Dual-NN was trained, in order to compare its performance with the proposed controllers by Garrard and Jordan [85], an analysis similar to that conducted in section 4.2.4 to test the limits of the maximum allowable initial angles of attack for the three controllers, is conducted for the Dual-NN. Table 4.5 shows the maximum initial angles of attack at which the Dual-NN can successfully recover from the stall region for varying initial pitch angle by $\theta_{(0)} = \pm 20^\circ$, and varying initial pitch rate by $q_{(0)} = \pm 20^\circ/\text{sec}$.

When compared with the 3rd-Order, which generated the best performance out of the three controllers proposed by Garrard and Jordan [85], the Dual-NN shows apriori that the values of maximum initial angle of attack are much larger than any of the values shown in Table 4.4. Figure 4.47 shows the comparison between the maximum allowable initial angles of attack for the 3rd-Order controller and the Dual-NN. It can be seen that the Dual-NN extends considerably the maximum allowable initial angle of attack for all

the ranges, but in order to determine the feasibility of the Dual-NN, simulations need to be made to determine magnitude of the states to determine if the performance of the Dual-NN is admissible.

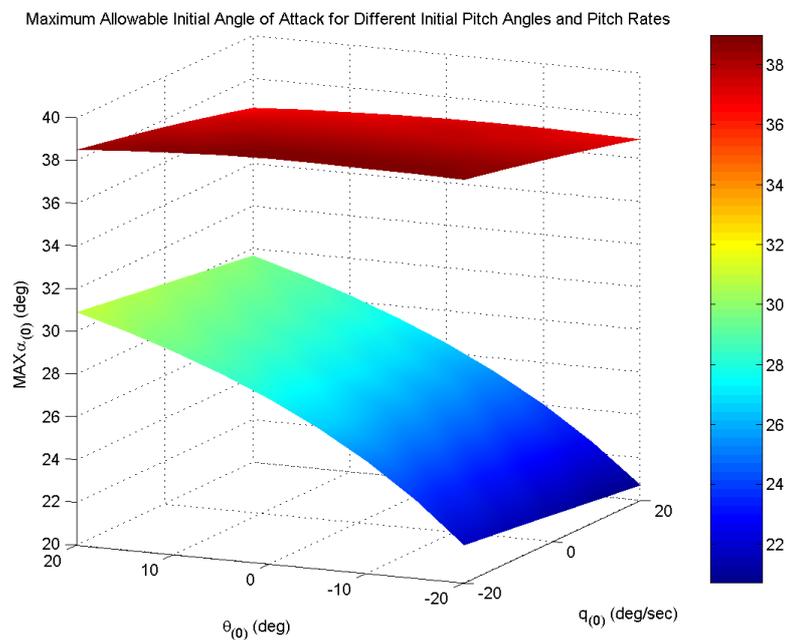


Figure 4.47. Comparison of the Maximum Allowable Initial Angle of Attack for the 3rd-Order and the Dual-NN Controller.

Figures 4.48 through 4.52 correspond to the time history simulations for the angle of attack, pitch angle, pitch rate, flight path angle and drop in altitude respectively, corresponding to the maximum initial values corresponding to the fifth column in Table 4.5, that is the maximum initial angle of attack at which the Dual-NN controller can safely recover the airplane from the stall region, which corresponds to the variable pitch angles, $\theta_{(0)} = \pm 20^\circ$ and for a pitch rate of $q_{(0)} = 0^\circ/\text{sec}$.

Table 4.5. Maximum Allowable Initial Angle of Attack for Different Initial Pitch Angles and Pitch Rates for the Dual-NN Controller.

		Initial Pitch Rate $q_{(0)}$								
		-20°/sec	-15°/sec	-10°/sec	-5°/sec	0°/sec	5°/sec	10°/sec	15°/sec	20°/sec
Initial Pitch Angle $\theta_{(0)}$	-20°	38.86°	38.64°	38.42°	38.18°	37.94°	37.69°	37.43°	37.17°	36.89°
	-15°	38.91°	38.69°	38.47°	38.23°	37.99°	37.74°	37.48°	37.21°	36.93°
	-10°	38.93°	38.72°	38.51°	38.28°	38.03°	37.78°	37.53°	37.25°	36.98°
	-5°	38.96°	38.75°	38.54°	38.31°	38.06°	37.81°	37.54°	37.27°	36.99°
	0°	38.96°	38.74°	38.52°	38.29°	38.01°	37.79°	37.53°	37.25°	36.97°
	5°	38.91°	38.69°	38.47°	38.23°	37.99°	37.74°	37.48°	37.21°	36.92°
	10°	38.8°	38.6°	38.38°	38.15°	37.91°	37.66°	37.4°	37.13°	36.83°
	15°	38.67°	38.46°	38.25°	38.03°	37.79°	37.54°	37.29°	37.02°	36.74°
	20°	38.5°	38.3°	38.09°	37.87°	37.64°	37.4°	37.14°	36.88°	36.6°

Despite that the Dual-NN controller is able to recover the airplane from the stall region for all the extreme initial conditions, above described, when looking at the physical implications of the dynamics of the simulations, it can be seen that large values are encountered in the angle of attack, pitch angle, pitch rate and flight path angle during the first few seconds of the recovery maneuver. Figure 4.48 shows that the Dual-NN brings down the initial values of angle of attack, listed in the 5th column of Table 4.5, to a change of about 80° in approximately a second, creating an overshoot corresponding to values in the angle of attack of $\alpha_{(0)} \approx -40^\circ$. The Dual-NN brings back the angle of attack to values of $\alpha_{(0)} \approx 11^\circ$ in the next second of the recovery maneuver, and from it brings the airplane smoothly to the steady state of zero angle of attack. The same trends of overshoot are observed in the magnitudes of pitch angle and pitch rate, which show values in excess of $\theta_{(0)} \approx -140^\circ$ for the pitch angle and $q_{(0)} \approx -300^\circ/\text{sec}$ for the pitch rate during the first second of the recovery maneuver and relaxes considerably in the next second of the maneuver to values near the desired steady state. The magnitudes of responses for the maximum allowable initial angle of attack for different initial pitch angles and pitch rates for the Dual-NN Controller need to be analyzed in more detail taking into consideration that:

- The extreme changes in angle of attack, pitch angle and pitch rate, and consequently the drop in altitude, are only encountered during the first few seconds of the recovery maneuver.
- The model used in this thesis uses a simplified longitudinal model of the dynamics of the aircraft with nonlinearities in the angle of attack, which does not fully describe the dynamics of the 6-DOF aircraft model.
- The airplane used for this thesis corresponds to a highly maneuverable fighter which can be safely assumed to be able to sustain extreme conditions.
- The model used in this thesis does not include dynamics for the structural integrity of the aircraft, which is beyond the scope of this thesis.

Recalling the analysis done in section 4.2.2, regarding the magnitude of the states and control weighting matrices, Eqs. (138) and (139), in which it was determined that if the weight cost function penalized the states and the control to lower values, i.e. Q and r used in this thesis, the controller would not have been able to extend the stall region beyond the penalized values. Despite that the Dual-NN controller is only doing what it has been asked to do, which is extending the stall-regime without penalizing the effort done by the control or the airplane, and as mentioned above the model used in this work is a simplified one that does not include the full nonlinear behavior of the dynamics, neither include the dynamics for the structural integrity of the aircraft, as an engineer, the author needs to take the results described above with a grain of salt. The physical implications of the magnitudes achieved by the state for the extreme limiting conditions at which the Dual-NN can recover the airplane from the stall region need to be taken into consideration.

For that reason more relaxed initial conditions that allow for more reasonable magnitudes for the state, will be used for the Dual-NN simulations described in the next section. As it will be seen in the next section, despite the relaxation of the initial conditions, the Dual-NN controller synthesized in the thesis, outperforms any of the three proposed controllers by Garrard and Jordan [85], and considerably extends the stall region.

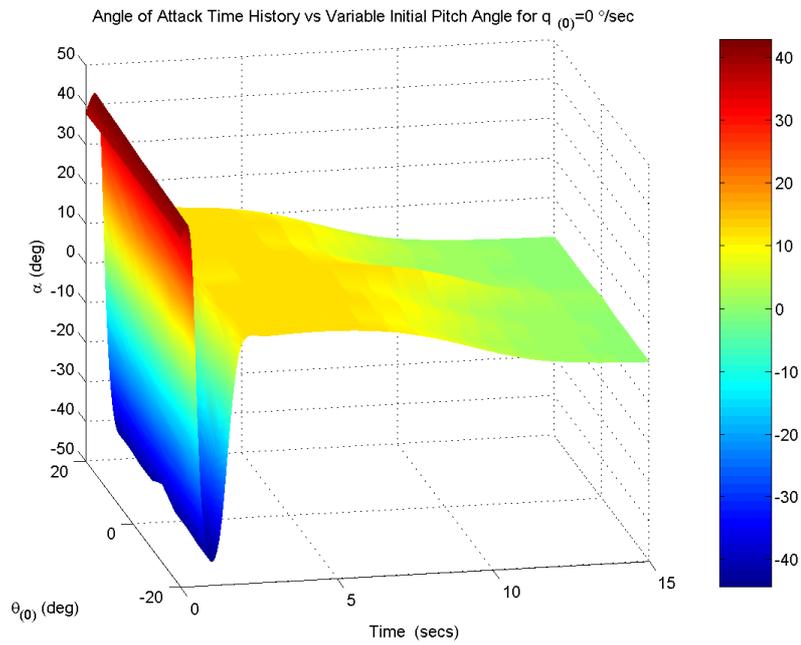


Figure 4.48. Angle of Attack Time History for the Dual-NN Controller ($\theta_{(0)}=\pm 20^\circ$ and $q_{(0)}=0^\circ/\text{sec}$).

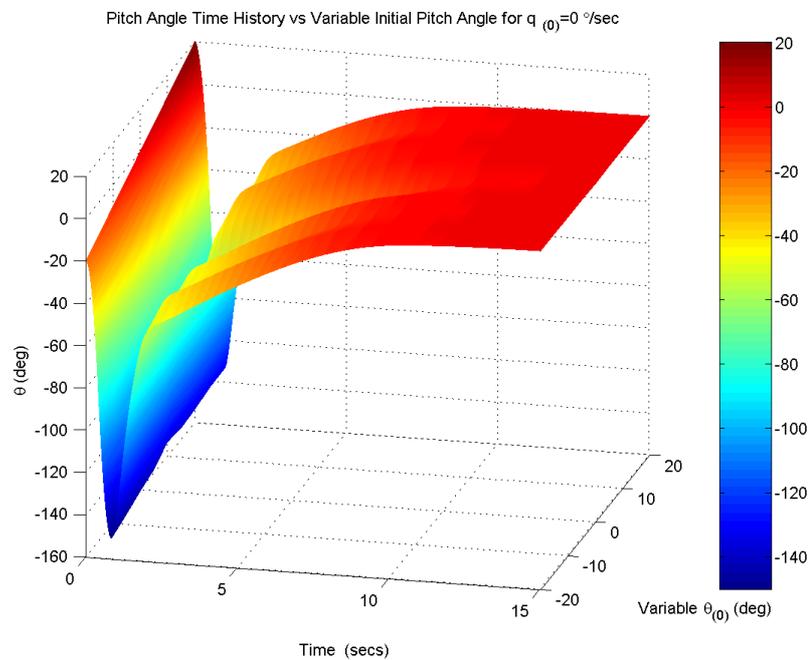


Figure 4.49. Pitch Angle Time History for the Dual-NN Controller ($\theta_{(0)}=\pm 20^\circ$ and $q_{(0)}=0^\circ/\text{sec}$).

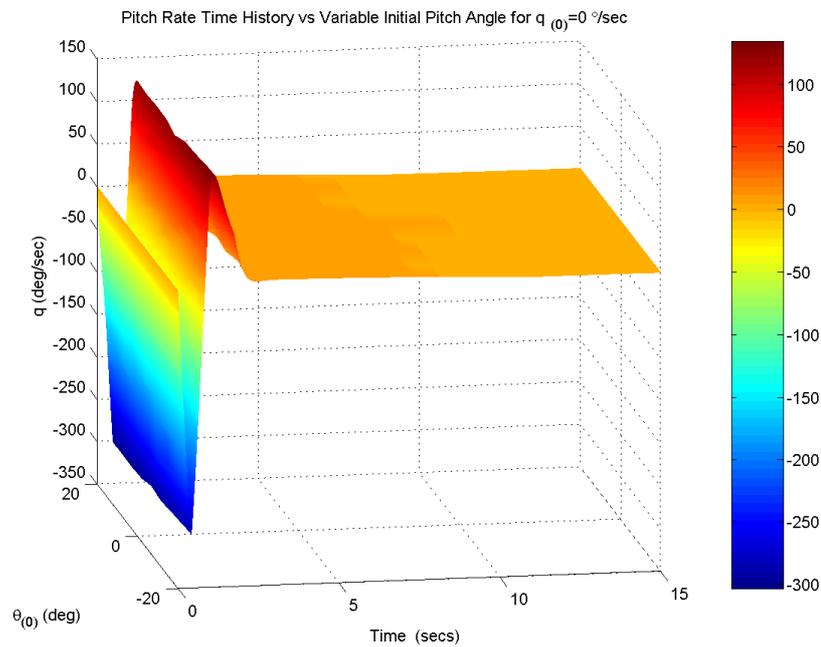


Figure 4.50. Pitch Rate Time History for the Dual-NN Controller ($\theta_{(0)}=\pm 20^\circ$ and $q_{(0)}=0^\circ/\text{sec}$).

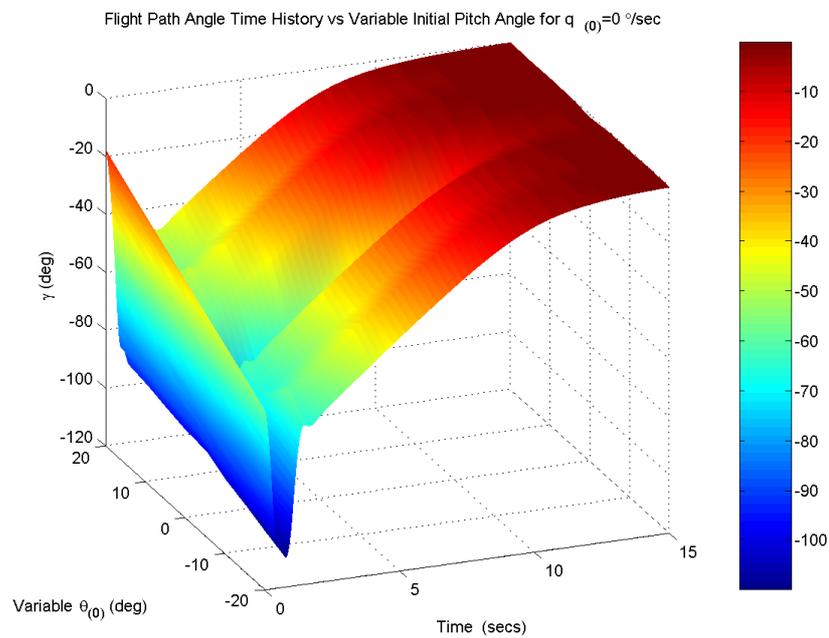


Figure 4.51. Flight Path Angle Time History for the Dual-NN Controller ($\theta_{(0)}=\pm 20^\circ$ and $q_{(0)}=0^\circ/\text{sec}$).

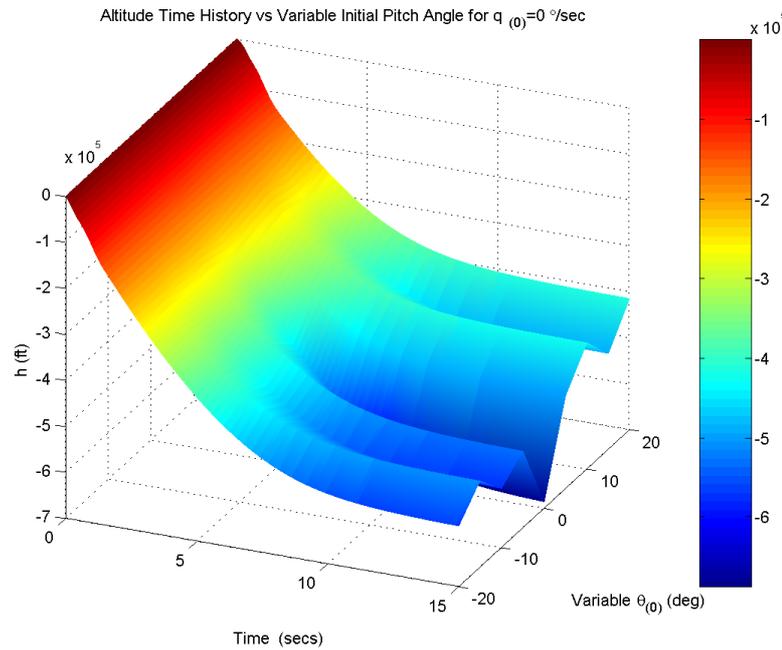


Figure 4.52. Drop in Altitude Time History for the Dual-NN Controller ($\theta_{(0)}=\pm 20^\circ$ and $q_{(0)}=0^\circ/\text{sec}$).

4.2.6. Analysis of Numerical Results. This section describes the simulations conducted to test the performance of the aircraft of the three different controllers derived by Garrard and Jordan [85] (δ_{e1} , δ_{e2} and δ_{e3} , Eqs. (142), (145) and (146) respectively) against the Dual-NN solution. As previously described, the initial flight conditions are Mach=0.85 and 30,000 feet (9000 m). It needs to be noted that, as mentioned in section 4.2.2, a limiter is placed in the simulations such that the tail rotation would not exceed a maximum rotation of $\pm 25^\circ$, and a tail rotation rate of $\pm 60^\circ/\text{sec}$. This is achieved by introducing logic in the integration routine that checks that the commanded tail rotation, generated by the controllers, does not exceed the $\pm 25^\circ$ allowable tail rotation. If the commanded tail rotation exceeds that value, then the logic in the integration routine limits the controller as such that

$$\text{if } |\delta_{e(i+1)}| > 25^\circ \text{ then } \delta_{e(i+1)} = \pm 25^\circ \quad (174)$$

The \pm in Eq. (174) indicates the correct sign of the tail rotation. Secondly, to check that the tail rotation rate is limited to $\pm 60^\circ/\text{sec}$, another logic routine is introduced as such that it checks the previously commanded control, i.e. $\delta_{e(i)}$, and compares it with the current commanded control, i.e. $\delta_{e(i+1)}$ and makes sure that the new commanded control does not exceed the limit tail rotation rate. Since the integration time-step used is $\Delta t = 0.01$, then the maximum tail rotation rate at each integration time-step is set to $0.6^\circ/\text{step}$. The rotation rate at each time step is calculated using Eq. (174). If the calculated rate at each step exceeds the maximum allowable tail rotation rate per step, then the new control is calculated using the previous control as indicated in Eq. (176)

$$\delta_{e,rate(i)} = (\delta_{e(i+1)} - \delta_{e(i)}) \Delta t \quad (175)$$

$$\delta_{e(i+1)} = \delta_{e(i)} \pm \delta_{e,rate_{MAX}} \quad (176)$$

The \pm indicates the correct sign of the tail rotation rate according to the magnitude of the previous rate. It needs to be noted that at $t(0) = 0$, the initial tail rotation is assumed to be $\delta_e(0) = 0^\circ$, therefore both the maximum tail rotation, and maximum tail rotation-rate are enforced for the initial commanded controller at $t(1) = \Delta t$, avoiding that way an unreasonable instantaneously initial tail rotation command.

For simplicity, the performance analysis comparison of the three proposed controllers by Garrard and Jordan [85], and the Dual-NN controller, will only be conducted with the limiting initial angles of attack that correspond to initial pitch angle and pitch rates of $\theta_{(0)} = 0^\circ$ and $q_{(0)} = 0^\circ/\text{sec}$, for the three compared controllers. Although these initial conditions seem a little bit unrealistic, due to the physical implication of zero pitch angle and pitch rate during a sudden increase in the angle of attack, these initial conditions are only used as a benchmark for comparison of the Dual-NN controller against the proposed controllers. This simplification reduces the lengthy amount of plots to be displayed in this thesis, and allows centering the focus of the analysis in the performance of the Dual-NN controller. After the Dual-NN is compared with the three controllers, different initial conditions are used to determine the limits and the

performance of the Dual-NN. The limiting conditions for the LQR, the 2nd-Order and the 3rd-Order controller, are $\alpha_{(0)}=25.73^\circ$, $\alpha_{(0)}=25.99^\circ$, and $\alpha_{(0)}=27.09^\circ$ respectively as seen in Tables 4.2, 4.3 and 4.4 in section 4.2.4.

The comparisons of the NN with the limiting cases of the compared controllers are displayed in Figures 4.53 through 4.58. Figures 4.53 and 4.54 show the time response for initial conditions of $\alpha_{(0)}=25.73^\circ$, $\theta_{(0)}=0^\circ$, and $q_{(0)}=0^\circ/\text{sec}$. Figures 4.55 and 4.56 show the time response for initial conditions of $\alpha_{(0)}=25.99^\circ$, $\theta_{(0)}=0^\circ$, and $q_{(0)}=0^\circ/\text{sec}$. Figures 4.57 and 4.58 show the time response for initial conditions of $\alpha_{(0)}=27.09^\circ$, $\theta_{(0)}=0^\circ$, and $q_{(0)}=0^\circ/\text{sec}$. It can be seen that the Dual-NN controller reaches equilibrium faster than any of the three compared controllers, and in addition has the lowest loss in altitude, the lowest cost and reaches the steady state of zero flight path angle faster than any of the three proposed controllers by Garrard and Jordan [85].

The flight path angle time history, depicted in bottom right hand corner of Figures 4.53, 4.55 and 4.57, and the altitude time history, depicted in the top left corner of Figures 4.54, 4.56 and 4.58, show that the Dual-NN outperforms any of the other three compared controllers by reaching the steady state faster, and having the smaller altitude loss. When comparing the flight path angle and the altitude loss-rate time-histories, similarities can be seen in their profiles. Flight path angle provides a direct relationship for the altitude loss rate, which in return indicates a loss in altitude. This is the main reason why it was considered inadmissible that a controller would create a negative steady state error in the pitch angle, no matter how small the steady state error was. As described previously, a non-zero steady state error implies a non-zero altitude loss rate, which in return implies a continuous change in altitude, and for the case described in Figure 4.19, where there existed a negative steady state error, would have implied a continuous loss in altitude despite that the aircraft recovered from the stall condition.

The tail rotation time-history and the cost associated to each control are depicted in the lower left corner and lower right corner respectively of Figures 4.54, 4.56 and 4.58, and it can be seen that the Dual-NN controller has the smallest control effort and cost associated.

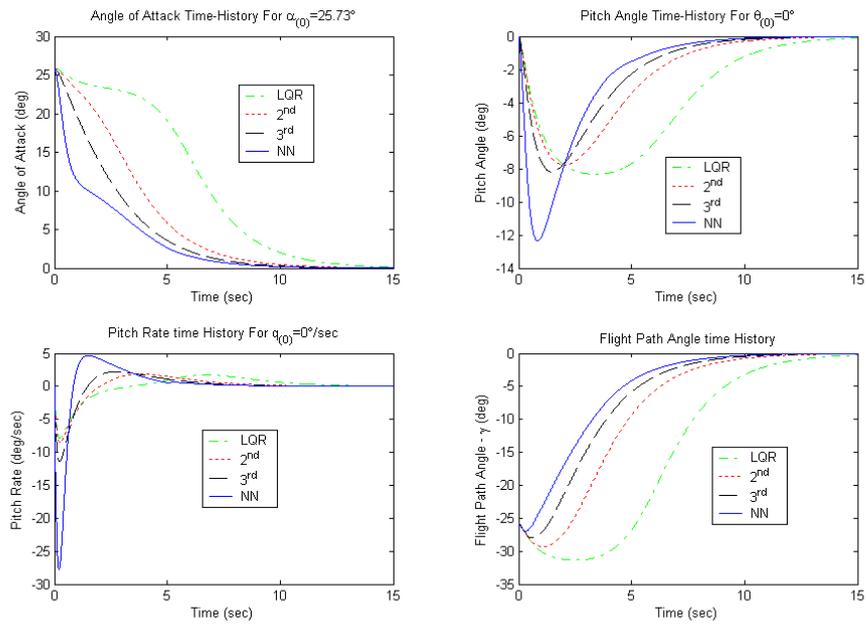


Figure 4.53. States Results ($\alpha_{(0)}=25.73^\circ, \theta_{(0)}=0^\circ, q_{(0)}=0^\circ/\text{sec}$).

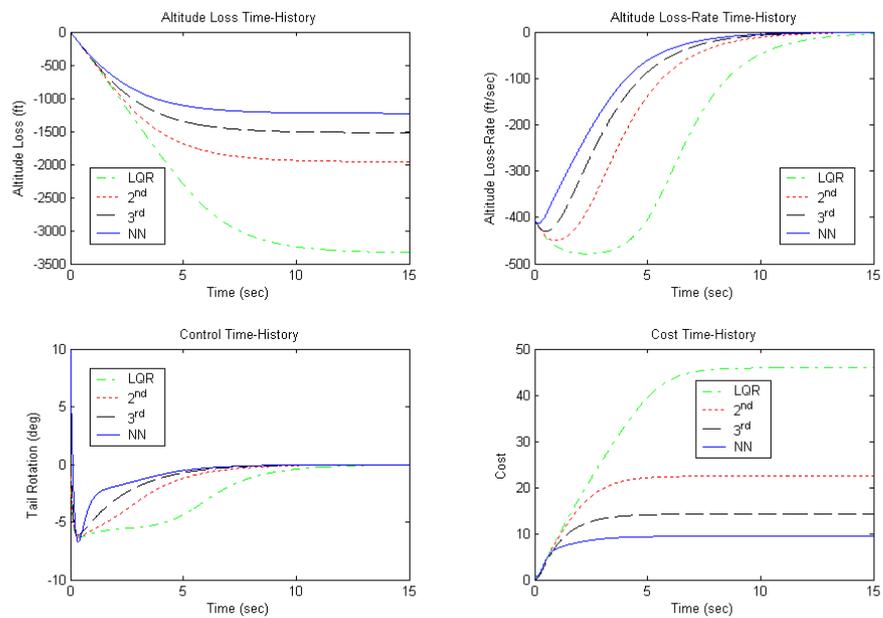


Figure 4.54. Altitude, Control and Cost Results ($\alpha_{(0)}=25.73^\circ, \theta_{(0)}=0^\circ, q_{(0)}=0^\circ/\text{sec}$).

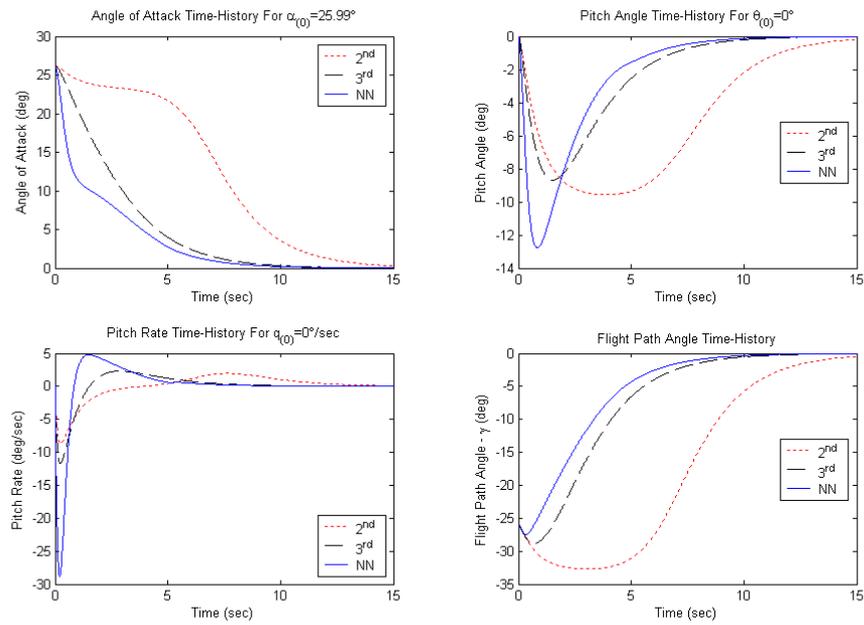


Figure 4.55. States Results ($\alpha_{(0)}=25.99^\circ$, $\theta_{(0)}=0^\circ$, $q_{(0)}=0^\circ/\text{sec}$).

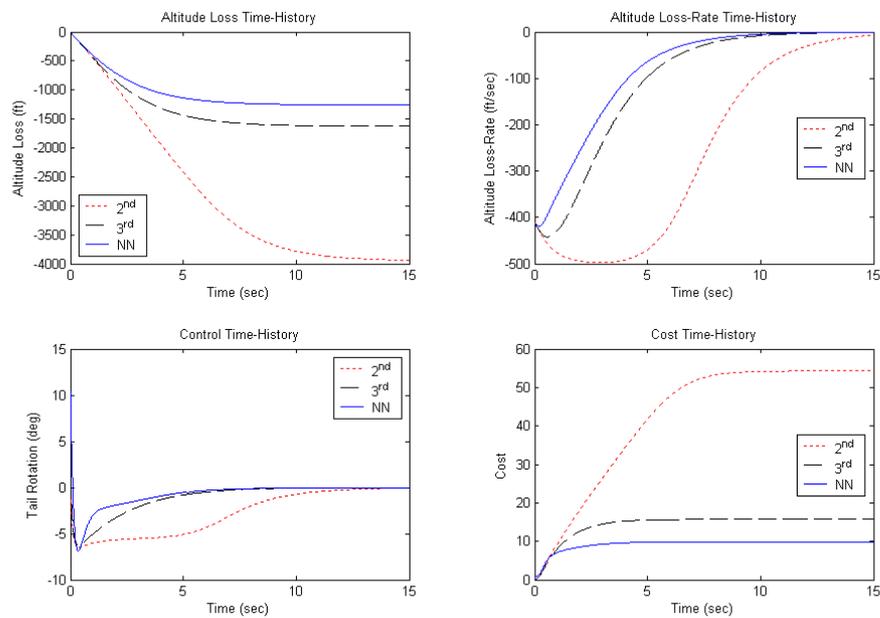


Figure 4.56. Altitude, Control and Cost Results ($\alpha_{(0)}=25.99^\circ$, $\theta_{(0)}=0^\circ$, $q_{(0)}=0^\circ/\text{sec}$).

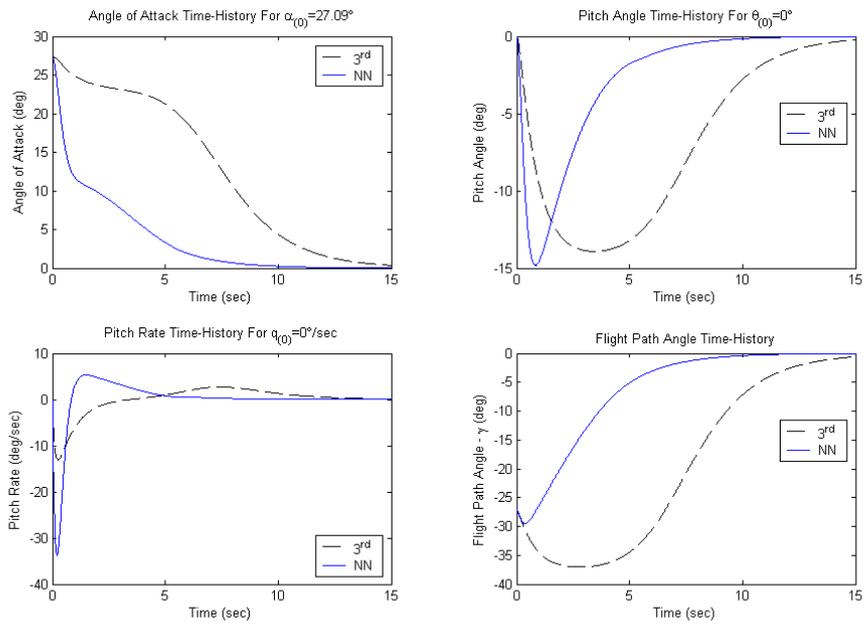


Figure 4.57. States Results ($\alpha_{(0)}=27.09^\circ$, $\theta_{(0)}=0^\circ$, $q_{(0)}=0^\circ/\text{sec}$).

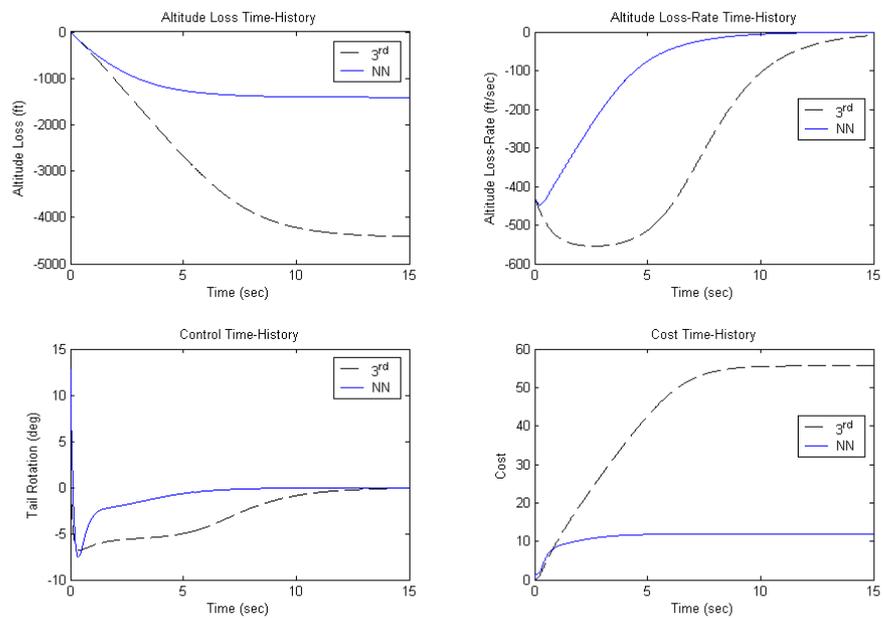


Figure 4.58. Altitude, Control and Cost Results ($\alpha_{(0)}=27.09^\circ$, $\theta_{(0)}=0^\circ$, $q_{(0)}=0^\circ/\text{sec}$).

It needs to be noted that in order to bring the initial angle of attack much faster than any of the three other controllers, the Dual-NN controller allows the pitch angle and pitch rate to increase more than the other controllers during the first seconds of maneuver. This can also be seen in the cost during the first seconds where the cost associated to the Dual-NN is greater during those few seconds, but the general performance of the Dual-NN controller is far superior than the rest of the proposed controllers. This faster response reflects in the fact that the Dual-NN controller regains level flight faster than the other three controllers. This higher value of pitch angle and pitch rate during the first two seconds of the recovery maneuver is the common feature for the dual-NN controller as the initial angle of attack is increased.

As described in section 4.2.5.3, since the magnitude of the state and control weighting matrices used in this thesis allow for these high values of pitch angle and pitch rate, which in turn, are required in order to allow the control to extend the stall region, these high values in the pitch angle and pitch rate will be considered reasonable as long as they are not sustained for long periods of time, which might be considered a hazard for the structural integrity of even an agile fighter, and as long as the pitch angle does not create extreme instantaneous loss in altitude which can also be seen as a hazard for the structural integrity of the airplane.

After this initial angle of attack of 27.09° , none of the three controllers presented by Garrard and Jordan [85] can recover from stall. In order to test the limits of the NN control, it was decided to further increase the initial value of angle of attack to $\alpha_{(0)}=30^\circ$, while keeping $\theta_{(0)}=0^\circ$ and $q_{(0)}=0^\circ/\text{sec}$. Figures 4.59 and 4.60 show the time histories for the states, altitude loss control and cost for the Dual-NN controller.

Figure 4.59 shows that the Dual-NN is able to bring down the initial angle of attack of $\alpha_{(0)}=30^\circ$, to a little more than 10° , in little more than one second. In order to achieve this fast response, the controller brings the initial pitch angle to approximately -22° , and the initial pitch rate to approximately $-50^\circ/\text{sec}$ in the half second of the recovery maneuver. After that, the Dual-NN brings the high values of pitch angle and pitch rate back to the desired steady state range by the end of the first second, allowing only those extreme conditions for a brief period of time.

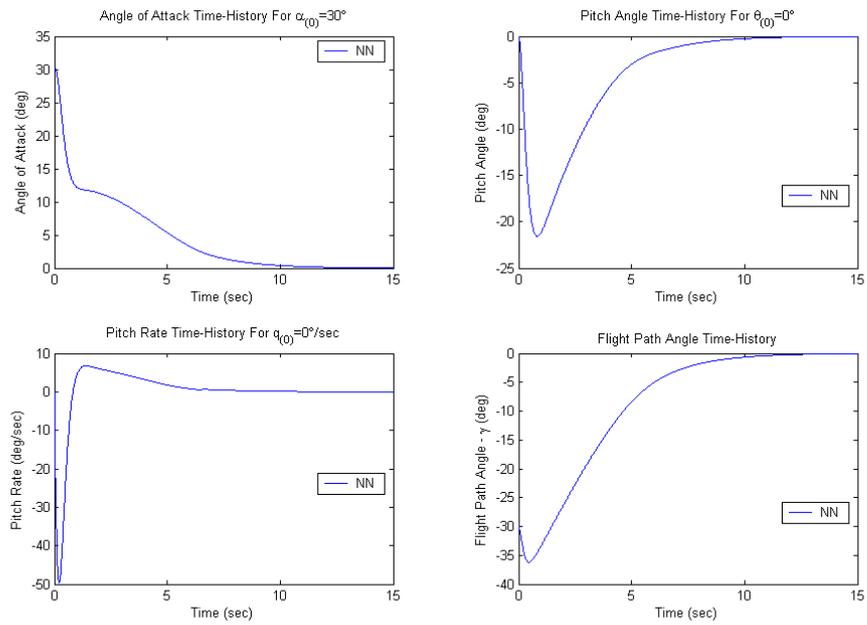


Figure 4.59. States Results ($\alpha_{(0)}=30^\circ$, $\theta_{(0)}=0^\circ$, $q_{(0)}=0^\circ/\text{sec}$).

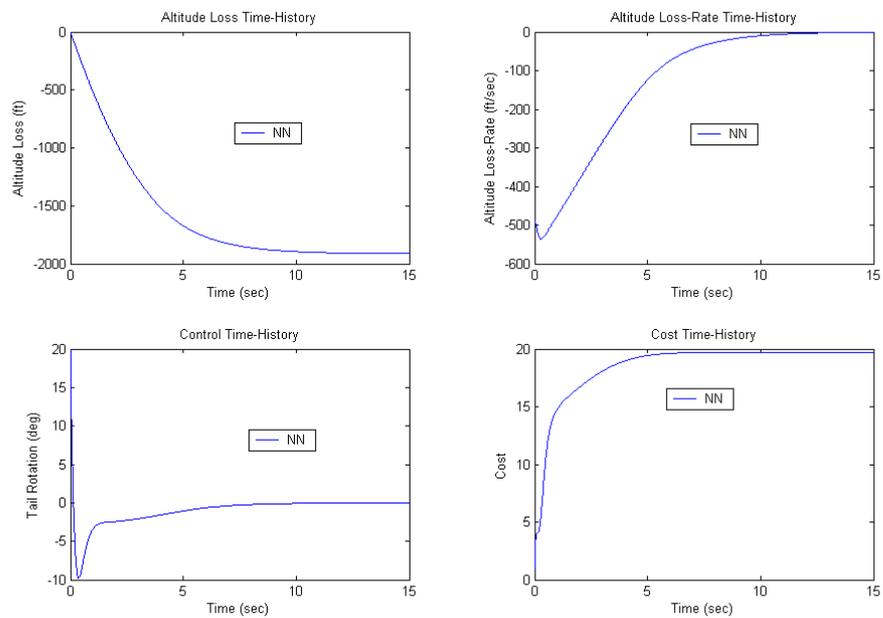


Figure 4.60. Altitude, Control and Cost Results ($\alpha_{(0)}=30^\circ$, $\theta_{(0)}=0^\circ$ and $q_{(0)}=0^\circ/\text{sec}$).

It is really important to note that the excessive pitch angle and pitch rate during the first instants of the recovery maneuver are extremely sensitive to the choice of initial conditions of pitch angle and pitch rate. As mentioned previously, a change in the angle of attack with initial conditions of zero pitch angle and pitch rate, implies a sudden, almost instantaneous change in the initial angle of attack, more like an impulse, which is really an unrealistic behavior for an airplane encountering a wind gust of such magnitude. Recalling Figure 4.5 in section 4.1.2, it can be seen that the angle of attack represents the angle of the airplane relative to its velocity vector, while the pitch angle represents the angle of the airplane relative to the horizon. A positive angle of attack with zero initial pitch angle represents a condition in which the flight path angle is equal to the angle of attack, but with opposite sign, i.e. negative. This implies a high negative altitude loss-rate. Although a negative flight path angle and a drop in altitude are expected as the angle of attack increases beyond the stall region, due to the loss in lift, if the maneuver is done progressively, meaning that the pilot slowly increases the angle of attack beyond the stall angle, allowing the airplane's flight path angle to be negative but small, then the pitch angle will be, according to Eq. (45), equal to the angle of attack plus the small negative flight path angle thus being a more reasonable initial pitch angle than that of zero. This only proves that the Dual-NN controller is showing incredible control response for such extreme initial conditions as tested above.

In order to analyze the case in which a more relaxed maneuver is used to extend the initial stall regime, more conservative initial conditions are used in the simulations. Figures 4.61 and 4.62 show the simulations results for initial conditions for the angle of attack of $\alpha_{(0)} = 30^\circ$, pitch angle of $\theta_{(0)} = 20^\circ$ and a pitch rate $q_{(0)} = 5^\circ/\text{sec}$. When comparing these results with the previous results described in Figures 4.59 and 4.60, it can be observed that the NN performs much better for this combination of initial conditions, than those described in Figures 4.59 and 4.60, i.e. $\theta_{(0)} = 0^\circ$ and $q_{(0)} = 0^\circ/\text{sec}$. This can be better observed in the altitude loss for both cases, as seen in the top left corner of Figures 4.60 and 4.62, where for the more extreme initial condition the loss in altitude during the recovery maneuver is approximately -1900 ft., while for the more relaxed initial condition, the loss in altitude is approximately only -750 ft. Figure 4.62 also shows that the tail rotation control effort is still within the limits.

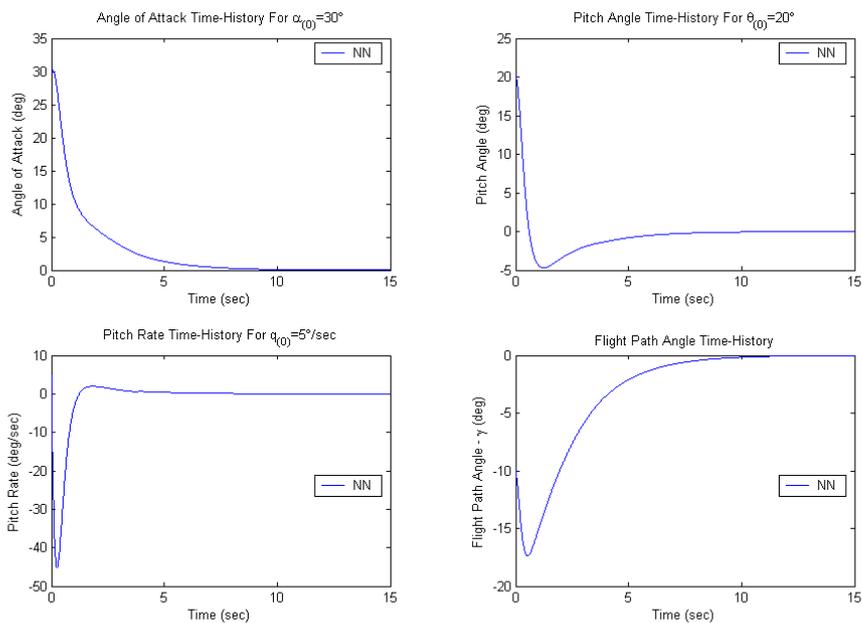


Figure 4.61. States Results ($\alpha_{(0)}=30^\circ, \theta_{(0)}=20^\circ, q_{(0)}=5^\circ/\text{sec}$).

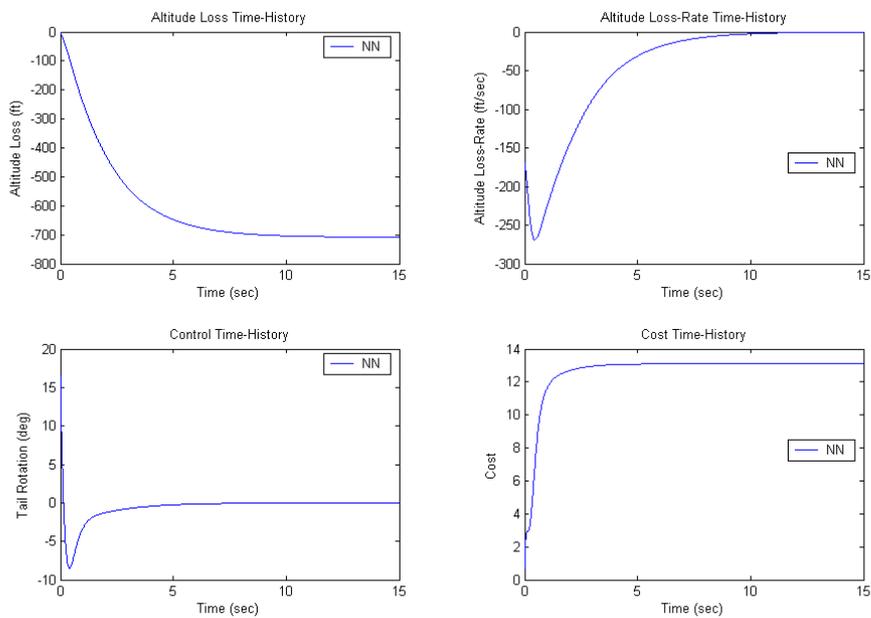


Figure 4.62. Altitude, Control and Cost Results ($\alpha_{(0)}=30^\circ, \theta_{(0)}=20^\circ, q_{(0)}=5^\circ/\text{sec}$).

In order to have a better picture of the influence that different initial conditions have in the angle of attack recovery, a study is conducted by varying the initial conditions for the simulations for a range of $\theta_{(0)} = \pm 20^\circ$ and $q_{(0)} = \pm 20^\circ$.

Figure 4.63 shows the time-history for the angle of attack, pitch angle, pitch rate, and flight path angle for different initial conditions of the pitch angle of $\theta_{(0)} = \pm 20^\circ$, while maintaining the initial conditions of $\alpha_{(0)} = 30^\circ$ and $q_{(0)} = 5^\circ/\text{sec}$ constant. It can be seen that as the initial pitch angle moves towards the negative side of the parametric range, a decrease in performance is observed in all graphs. This is expected as the initial flight path angle becomes more negative as the initial pitch angle moves towards the negative region of the range. These create a set of more extreme initial conditions that demand in return a higher pitch angle and pitch rate in the initial instants to bring down the initial angle of attack.

Figure 4.64 shows the time-history for the altitude loss, altitude loss-rate, the associated cost and the tail rotation control for different initial conditions of the pitch angle of $\theta_{(0)} = \pm 25^\circ$, while maintaining the initial conditions of $\alpha_{(0)} = 30^\circ$ and $q_{(0)} = 5^\circ/\text{sec}$ constant. Again, the decrease in performance as the initial pitch angle conditions move towards the negative region of the range is evident in the loss in altitude and the cost associated. The tail rotation time history shows that for the more extreme initial conditions, there is an initial saturation of the commanded tail rotation, indicated by achieving the maximum limit of $+25^\circ$. Despite that initial saturation, the Dual-NN controller is able to bring down the initial angle of attack of $\alpha_{(0)} = 30^\circ$ to little more than 11° in less than 1 second. After that first second, and depending on the associated pitch angle and pitch rate initial conditions, the angle of attack is brought down to the steady state of zero with a faster or slower response.

Figure 4.65 shows the time-history for the angle of attack, pitch angle, pitch rate, and flight path angle for different initial conditions of the pitch rate from $q_{(0)} = \pm 20^\circ/\text{sec}$, while maintaining the initial conditions of $\alpha_{(0)} = 30^\circ$ and $\theta_{(0)} = 20^\circ$ constant. It can be seen that as the initial pitch rate is varied, the change in performance varies too little, and the angle of attack, pitch angle, pitch rate, and flight path angle time-histories show little variation.

Figure 4.66 shows the time-history for the altitude loss, altitude loss-rate, associated cost and the tail rotation control for different initial conditions of the pitch rate from $q_{(0)} = \pm 20^\circ/\text{sec}$, while maintaining the initial conditions of $\alpha_{(0)} = 30^\circ$ and $\theta_{(0)} = 20^\circ$ constant. The same trends are observed for the altitude loss and altitude loss rate, with the only apparent difference observed in the associated cost and the initial commanded tail rotation, which requires a higher performance of the Dual-NN controller for the higher initial pitch rates since it needs to turn that high positive initial pitch rate down to a negative value in order to bring the initial angle of attack down to a steady state value of zero.

The initial angle of attack is increased even further to test the limits of the Dual-NN controller. Figure 4.67 shows the angle of attack, pitch angle, pitch rate and flight path angle results for a more conservative initial conditions with initial angle of attack of $\alpha_{(0)} = 35^\circ$, pitch angle of $\theta_{(0)} = 20^\circ$ and a pitch rate $q_{(0)} = 5^\circ/\text{sec}$. It can be observed that the Dual-NN is still able to recover from the stall condition of $\alpha_{(0)} = 35^\circ$, which represents an extension to the stall regime of 49%. It also needs to be noted that this initial angle of attack creates a response of pitch rate in excess of $-90^\circ/\text{sec}$ during the first half second of the maneuver, which recovers to smaller values of pitch rate by the end of the first second. The flight path angle history shows the loss of altitude during the maneuver, which is accentuated during the first seconds of the maneuver in which the negative pitch angle and the negative pitch rate of the aircraft induce the change in the flight path of the aircraft.

Figure 4.68 shows the altitude loss, altitude loss-rate, tail rotation control, and cost time-histories for the above initial conditions. When comparing Figure 4.68 with Figure 4.66, which corresponds to a smaller initial angle of attack of $\alpha_{(0)} = 30^\circ$, it is observed the general decrease in performance. While the altitude loss is approximately 700 ft for $\alpha_{(0)} = 30^\circ$, the altitude loss for the $\alpha_{(0)} = 35^\circ$ case is more than 3 times higher, approximately 2500 ft. The same trends are seen when comparing the tail rotation effort, and the associated cost.

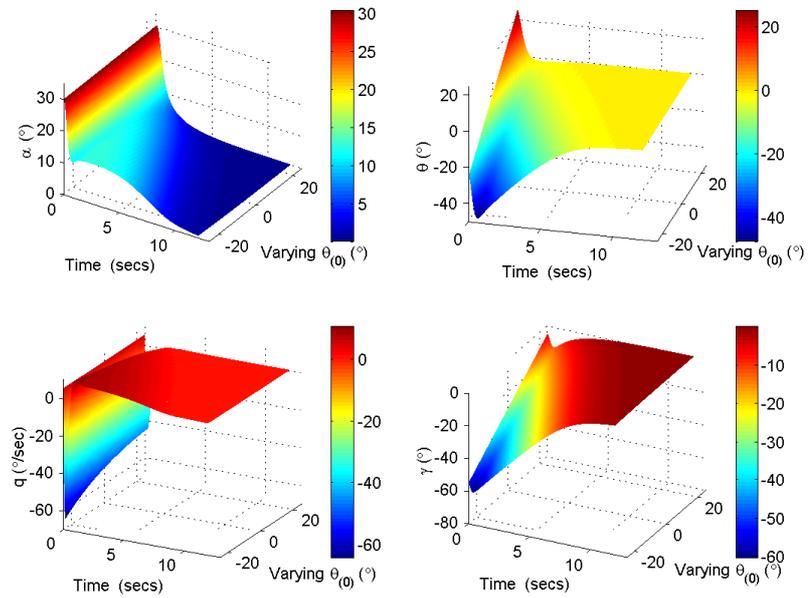


Figure 4.63. States Results ($\theta_{(0)}=\pm 20^{\circ}$, $\alpha_{(0)}=30^{\circ}$, $q_{(0)}=5^{\circ}/\text{sec}$).

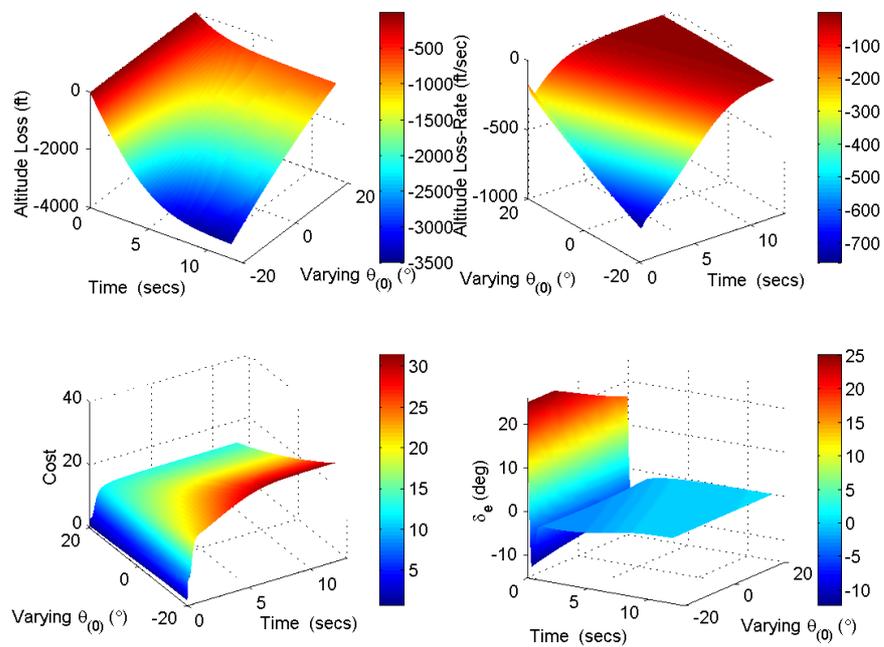


Figure 4.64. Altitude, Control and Cost Results ($\theta_{(0)}=\pm 20^{\circ}$, $\alpha_{(0)}=30^{\circ}$, $q_{(0)}=5^{\circ}/\text{sec}$).

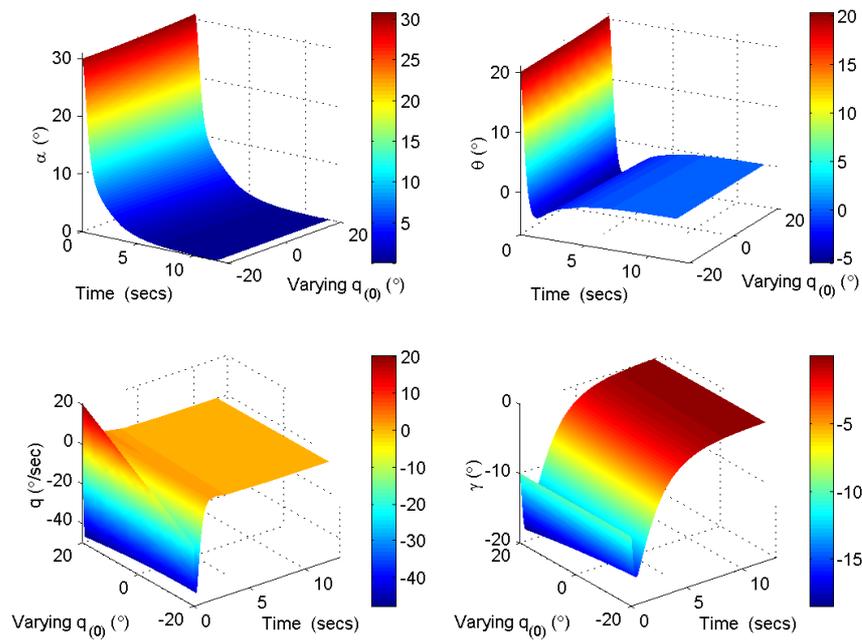


Figure 4.65. States Results ($q_{(0)} = \pm 20^\circ$, $\alpha_{(0)} = 30^\circ$, $\theta_{(0)} = 20^\circ/\text{sec}$).

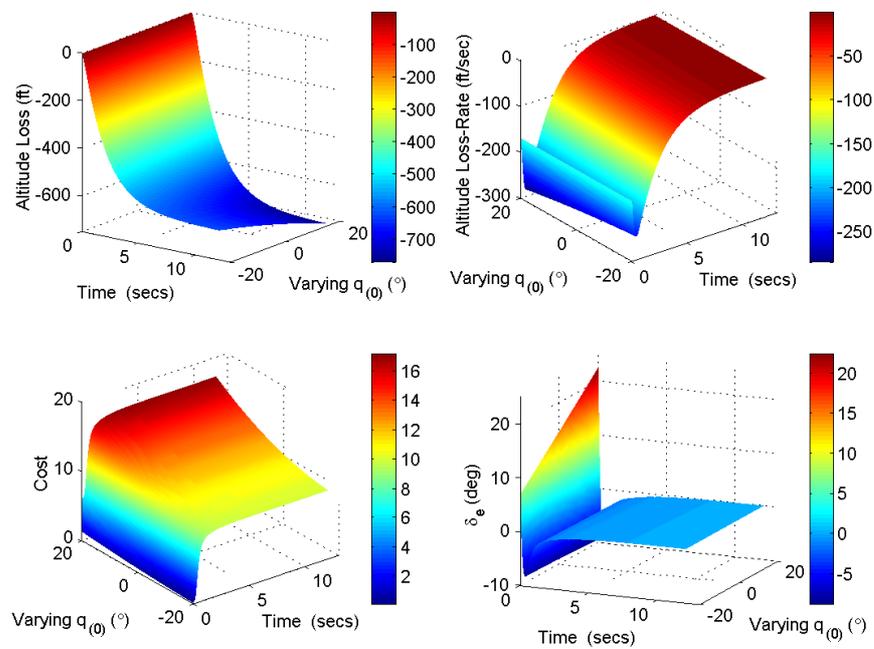


Figure 4.66. Altitude, Control and Cost Results ($q_{(0)} = \pm 20^\circ$, $\alpha_{(0)} = 30^\circ$, $\theta_{(0)} = 5^\circ/\text{sec}$).

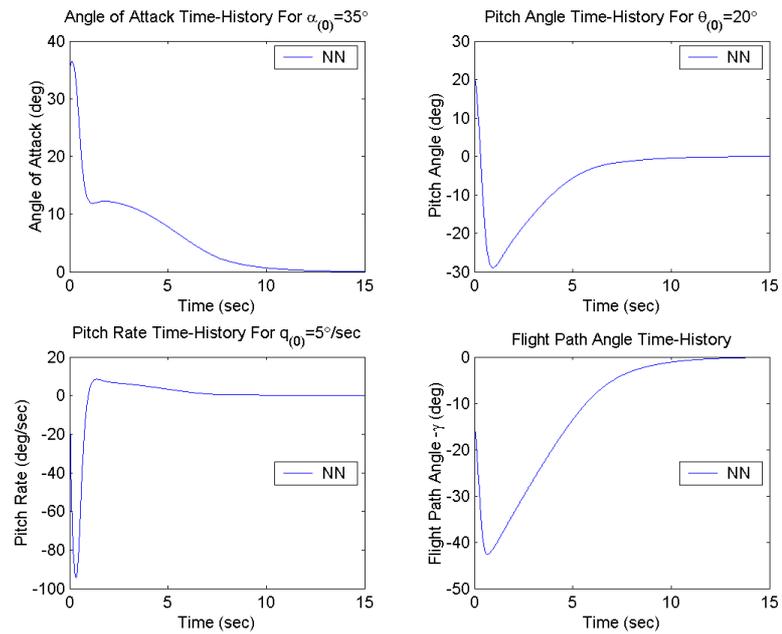


Figure 4.67. States Results ($\alpha_{(0)}=35^\circ$, $\theta_{(0)}=20^\circ$, $q_{(0)}=5^\circ/\text{sec}$).

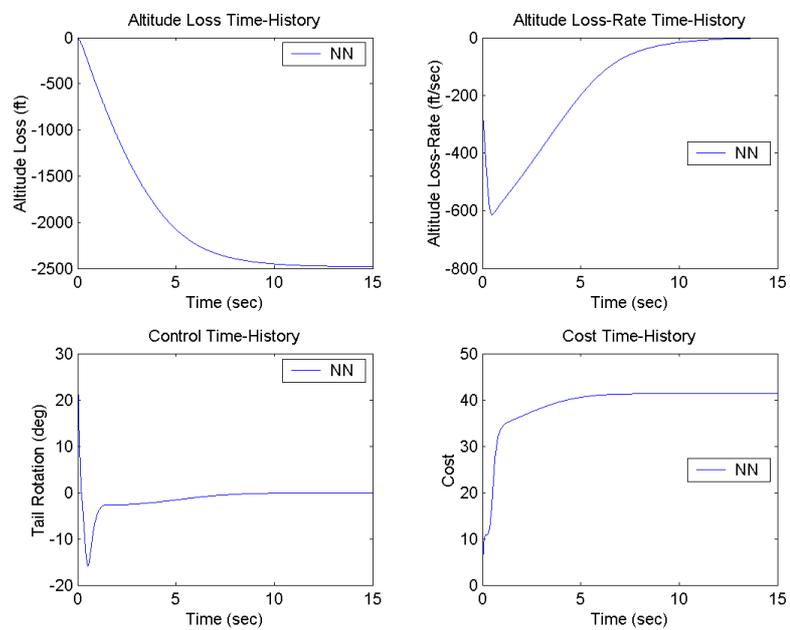


Figure 4.68. Altitude, Control and Cost Results ($\alpha_{(0)}=35^\circ$, $\theta_{(0)}=20^\circ$, $q_{(0)}=5^\circ/\text{sec}$).

In order to observe the variation of initial angle of attack, Figures 4.69 and 4.70 show the time response simulations for different initial angles of attack, below the stall condition, $\alpha_{(0)} = 22.9^\circ$, to well beyond the stall condition, $\alpha_{(0)} = 35^\circ$. Figure 4.69 shows the time-history responses for the angle of attack, pitch angle, pitch rate, and flight path angle. Figure 4.69 show the time-history responses for the altitude loss, altitude loss-rate, cost and tail rotation control respectively. The initial conditions of $\theta_{(0)} = 20^\circ$ and $q_{(0)} = 5^\circ/\text{sec}$ are maintained constant while the initial angle of attack is varied. It is observed that as the initial angle of attack is increased the initial responses for the pitch angle and pitch rate move towards more extreme regions with maximum values of approximately $\theta = -30^\circ$ and approximately $q = -90^\circ/\text{sec}$ within the first seconds for the maximum analyzed initial angle of attack. The pitch rate response quickly recovers after the first second, while the pitch angle recovery is a little bit more gradual. This behavior is also depicted in the response of the flight path angle, that shows a great change in the flight path of the aircraft in excess of $\gamma = -40^\circ$ for the maximum initial angle of attack.

Figure 4.70 shows the altitude loss, altitude loss-rate, cost and tail rotation for the initial conditions described above. The altitude loss portion of figure 4.70 shows the drastic change in altitude loss as the initial angle of attack is increased. This decrease in overall performance is even greater when you compare the cost time-history as the angle of attack is increased. Figure 4.70 also shows that the saturation limits for the initial tail rotation need to be enforced for the conditions in which the initial angle of attack is above $\alpha_{(0)} = 32^\circ$. This saturation of the tail rotation is only encountered in the initial commanded control.

In order to extend the analysis to observe the the influence of the variation of pitch angle in the higher initial angle of attack of $\alpha_{(0)} = 35^\circ$, Figures 4.71 and 4.72 show the time-history for different initial conditions of the pitch angle of $\theta_{(0)} = \pm 20^\circ$, while maintaininig the initial conditions of $\alpha_{(0)} = 35^\circ$ and $q_{(0)} = 5^\circ/\text{sec}$ constant. Only the variation in pitch angle is shown as previosly in Figures 4.65 and 4.66, the variation of pitch angle has a greater effect that the variation in pitch rate. As the initial pitch angle moves away towards the negative side of the parametric range, a decrease in performance is observed in all graphs.

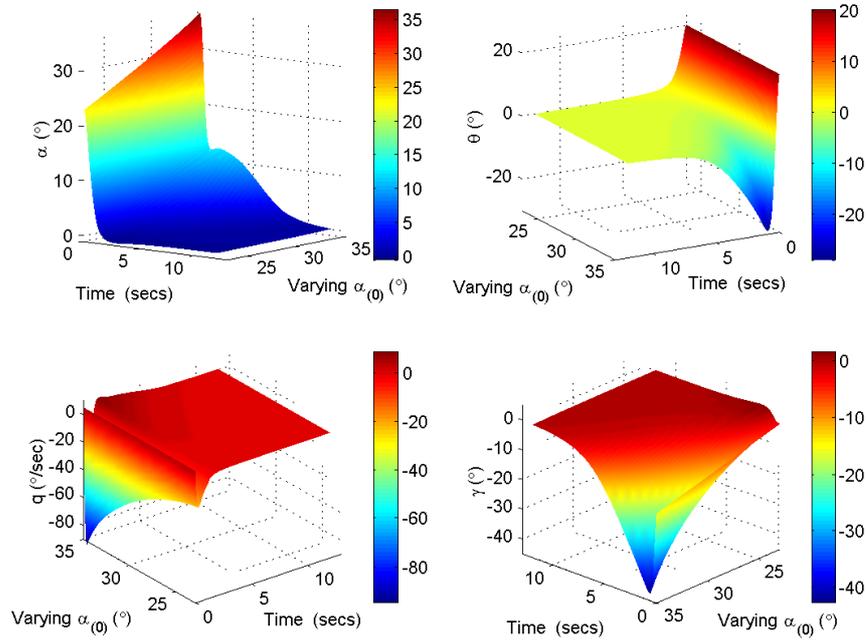


Figure 4.69. States Results ($\alpha_{(0)}=22.9-35^{\circ}$, $\theta_{(0)}=10^{\circ}$, $q_{(0)}=5^{\circ}/\text{sec}$).

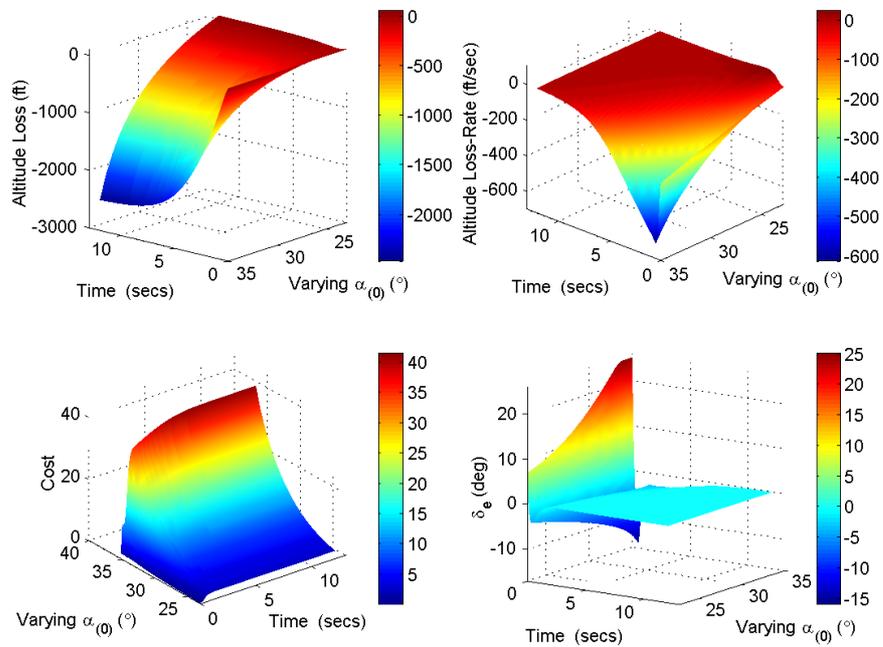


Figure 4.70. Altitude, Control and Cost Results ($\alpha_{(0)}=22.9-35^{\circ}$, $\theta_{(0)}=10^{\circ}$, $q_{(0)}=5^{\circ}/\text{sec}$).

Figure 4.71 shows the fast recovery of the initial angle of attack during the first second of the recovery, and also shows that as the initial pitch angle is varied towards the more negative region of the range, the airplane experiences pitch angles during the first seconds of the recovery maneuver that range from approximately $\theta_{(0)} = -20^\circ$ to $\theta_{(0)} = -70^\circ$. The same trends are shown in the pitch rate time-histories with maximum values of pitch rate of approximately $q_{(0)} = -80^\circ/\text{sec}$ to $q_{(0)} = -125^\circ/\text{sec}$, during the first half second of the recovery maneuver.

Figure 4.72 shows the drastic change in altitude loss and the associated cost as the initial pitch angle is varied. Figure 4.72 also shows the tail rotation control and it can be seen that the saturation limits of $\delta_{e,\text{MAX}} = +25^\circ$ need to be enforced at the initial tail rotation command for all different variations in pitch angle. The $\delta_{e,\text{MAX}} = -25^\circ$ saturation limit is also encountered at 0.5 seconds of the recovery maneuver of angle of attack for extreme negative portion of the initial pitch angle. Despite the enforcement of the maximum tail rotation, the control is able to bring the aircraft back to a steady state condition of zero angle of attack, pitch angle, pitch rate and flight path angle after a little bit more than 10 seconds.

Beyond this initial angle of attack, $\alpha_{(0)} = 35^\circ$, despite the fact that the Dual-NN controller can still recover to larger values of initial angle of attack as seen in Table 4.5 of section 4.2.5.3, it is considered that the values of pitch angle, pitch rate, flight path angle and loss in altitude are not reasonable, despite that the controller is still able to bring the airplane back from the stall region. Figures 4.73 and 4.74 show the time-history simulations for varying $\theta_{(0)} = \pm 20^\circ$ with $\alpha_{(0)} = 36.5^\circ$ and $q_{(0)} = 5^\circ/\text{sec}$. Despite the fast recovery of the initial angle of attack as seen in Figure 4.73, it encounters an overshoot by reaching negative values of angle of attack. The flight path angle reaches values of almost -90° , while the pitch rate encounters values of $-200^\circ/\text{sec}$ for the most negative values of initial pitch angle.

Figure 4.74 shows the tail rotation control history, being clear that the Dual-NN is commanding more than what the tail can actually physically allow since saturation limits are observed in the first seconds of the recovery. The cost time-history shows that the overall cost increases by 100% relative to the simulations described for an initial angle of attack of $\alpha_{(0)} = 35^\circ$ described in Figure 4.72.

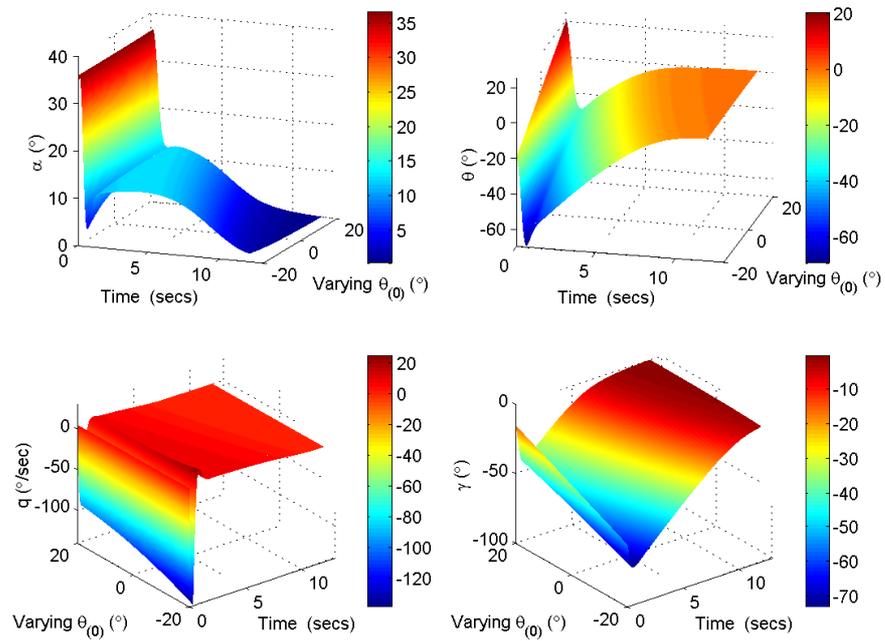


Figure 4.71. States Results ($\theta_{(0)}=\pm 20^\circ$, $\alpha_{(0)}=35^\circ$, $q_{(0)}=5^\circ/\text{sec}$).

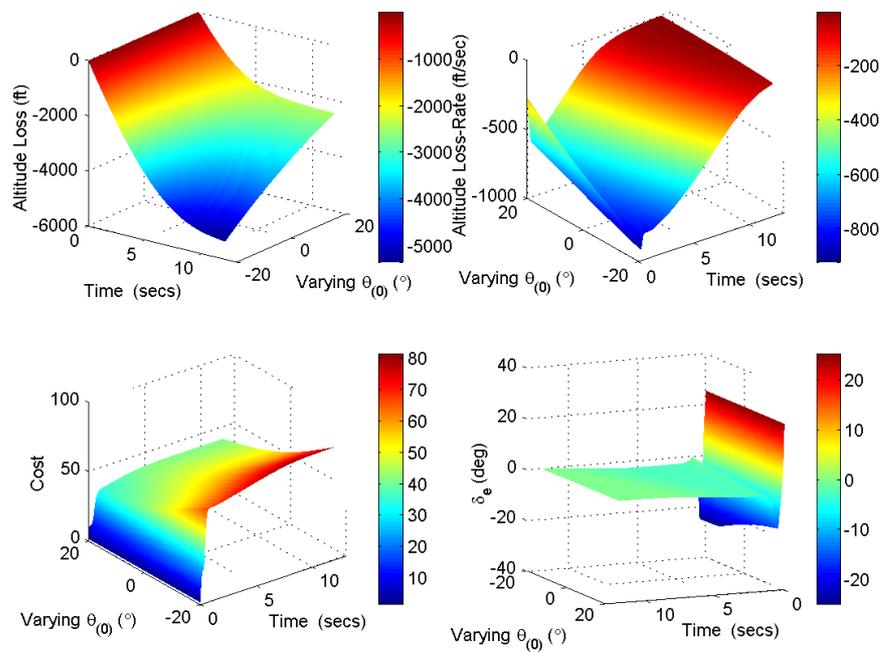


Figure 4.72. Altitude, Control and Cost Results ($\theta_{(0)}=\pm 20^\circ$, $\alpha_{(0)}=35^\circ$, $q_{(0)}=5^\circ/\text{sec}$).

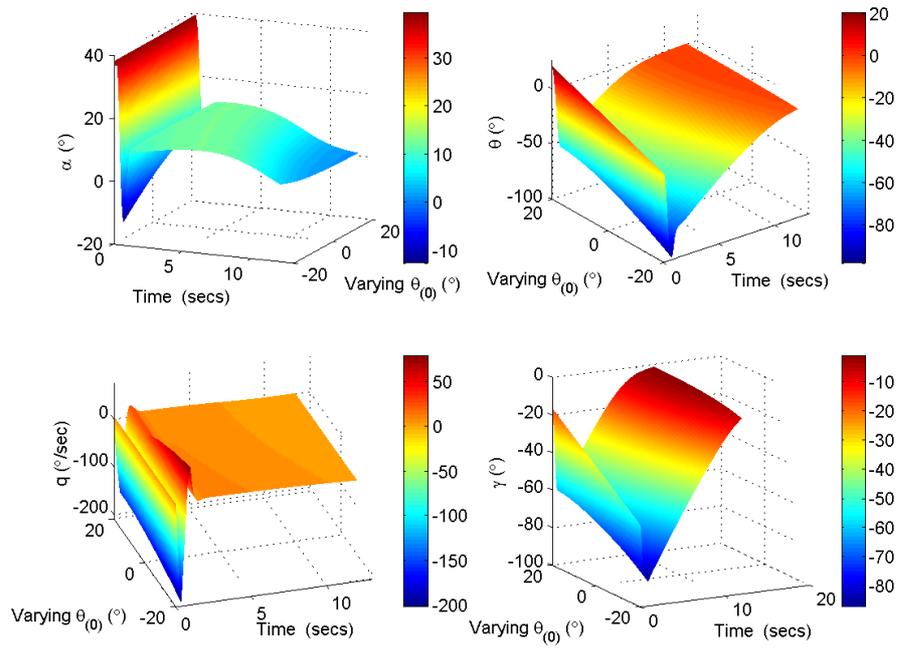


Figure 4.73. States Results ($\theta_{(0)}=\pm 20^{\circ}$, $\alpha_{(0)}=36.5^{\circ}$, $q_{(0)}=5^{\circ}/\text{sec}$).

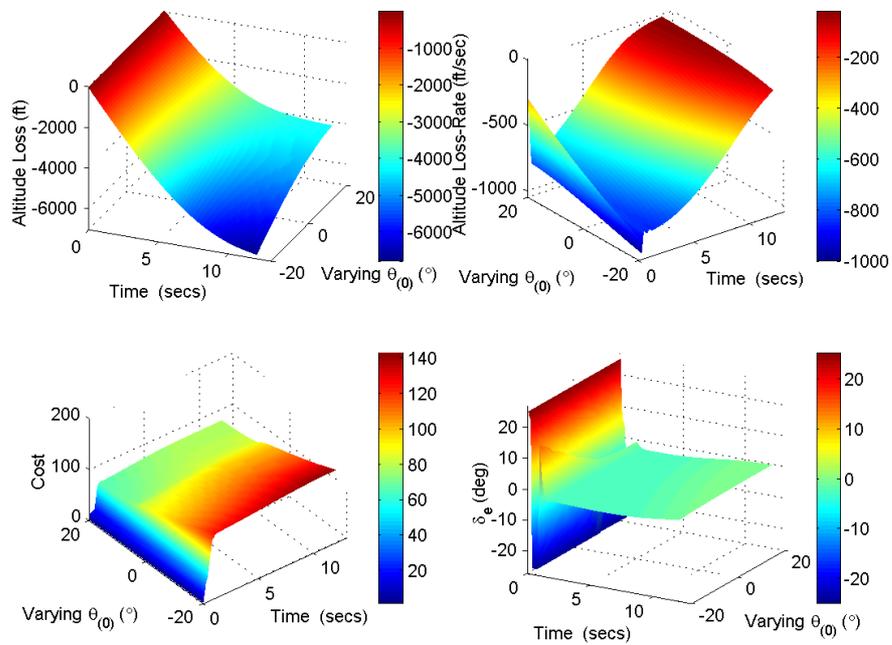


Figure 4.74. Altitude, Control and Cost Results ($\theta_{(0)}=\pm 20^{\circ}$, $\alpha_{(0)}=36.5^{\circ}$, $q_{(0)}=5^{\circ}/\text{sec}$).

4.2.7. Sensitivity Study of the Neurocontroller to Unmodeled Uncertainties.

The formulation presented in this section introduces unmodel uncertainties in the control derivative matrix to study the robustness of the designed neurocontroller to model/input uncertainties. Two types of unmodeled uncertainties (UU) are considered:

- Time lag between commanded and actual input control.
- Inaccurate control effectiveness derivatives.

The time lag between commanded and actual input control is modeled with a first order time-lag defined as

$$\delta_{e(i+1)} = \frac{1}{\tau} (\delta_{e(i)} - \delta_{e,NN(i)}) \quad (177)$$

where the actual control us defined by,

$$\delta_{e(i)} = \delta_{e,NN(i)} + e^{\frac{-\Delta t}{\tau}} (\delta_{e(i-1)} - \delta_{e,NN(i)}) \quad (178)$$

$\delta_{e(i)}$ represents the tail rotation, $\delta_{e,NN(i)}$ is the ANN commanded control, $\delta_{e(i-1)}$ is the tail rotation in the previous step time, and τ is the tail-time constant. In this study τ is varied from 0.4 to 0.02. This type of UU relate to those generated by the physical limitations of actuating a control surface. Unless electrical actuators are used, the hydraulic actuators, commonly used in the control surfaces of current aircraft, tend to generate a time lag between the commanded control deflection by the pilot or the autopilot system, and the actual control seen by the control surfaces. The study of this type of uncertainties is important since time constants are difficult to estimate accurately.

The second type of uncertainties addressed in this study are inaccurate control effectiveness derivatives. These uncertainties can be encountered in two situations:

- Inaccurate prediction of the tail control surfaces derivatives, $C_{L_{\delta e}}$ and $C_{M_{\delta e}}$.

- Loss of a percentage of the control surface.

In this thesis, this uncertainty is modeled by introducing a disturbance in the control matrix such that,

$$x_{i+1} = Ax_i + \phi(x_i) + b(1 + D_{un})\delta_{e(i)} \quad (179)$$

D_{un} represents the percentage of decrease or increase in tail effectiveness, for example a value of $D_{un} = -0.50$ represents a loss of 50% in the tail effectiveness. In this study D_{un} is varied from -0.50 to 0.50 . When $D_{un} < 0$, it can be associated to either a loss in effective area of the control surface, i.e. hit by a missile which destroys a percentage of the effective tail surface area, or to an error in the estimation of the tail effectiveness derivatives as such that their real effectiveness is less than predicted. When $D_{un} > 0$, this can be associated to a an error in the estimation of the tail effectiveness derivatives meaning that their real effectiveness is greater than predicted, which also needs to be checked to assure that the airplane does not become unstable from over-control.

4.2.8. Analysis of Results to Unmodeled Uncertainties. In order to analyze the inherent robustness to UU, several cases were studied varying the tail time constant τ , the tail effectiveness D_{un} and the initial angle of attack $\alpha(0)$. All studied cases have initial conditions of pitch angle of $\theta(0) = 20^\circ$ and pitch rate of $q(0) = 5^\circ/\text{sec}$ to limit the number of studied cases. The simulations are compared with the Dual-NN controller simulations described in section 4.2.6, which will be referred in this section as the desired controllers since it is free of UU. Needs to be noted that the Dual-NN controller with unmodel uncertainties (Dual-NN-UU) has no knowledge of the existence of these uncertainties. The robustness analysis is divided in two parts:

- Sensitivity study to varying τ and $\alpha(0)$, which will be referred as UU₁ throughout the remainder of the analysis.
- Sensitivity study to varying only D_{un} and $\alpha(0)$, which will be referred as UU₂ throughout the remainder of the analysis.

4.2.8.1 Analysis of the sensitivity study to varying time lag. Figures 4.75 and 4.76 show the time-history simulations for a time lag with unmodeled uncertainties of $\tau=0.01$ and initial angle of attack of $\alpha(0)=30^\circ$. Similar to the analysis conducted in previous sections, Figure 4.75 and 4.76 are divided in four subplots each. The angle of attack and pitch angle are located in the top left and top right corners respectively of Figure 4.75, while the pitch rate and the flight path angle are located in the bottom left and right corners respectively of the same figure. The altitude loss and altitude loss-rate are located in the top left and right corners respectively of Figure 4.76, while the associated cost and the flight path angle are located in the bottom left and right corners respectively of the same figure. The units for the plots are the same as used in previous analysis, with the angle of attack, pitch angle, flight path angle and tail rotation having units of degrees; the pitch rate having units of degrees-per-second; the altitude loss having units of feet; the altitude loss-rate having units of feet-per-second, and the cost being dimensionless.

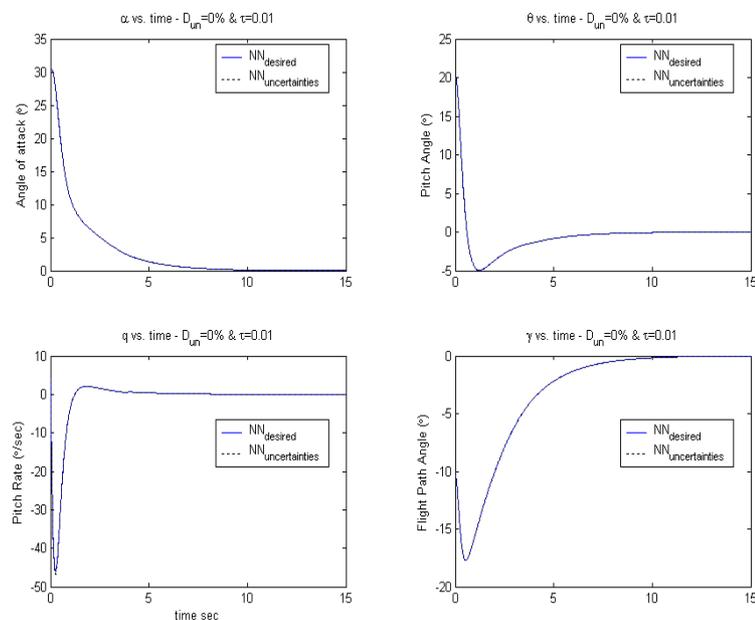


Figure 4.75. States Results ($\tau=0.01$, $\alpha(0)=30^\circ$, $\theta(0)=20^\circ$, $q(0)=5^\circ/\text{sec}$).

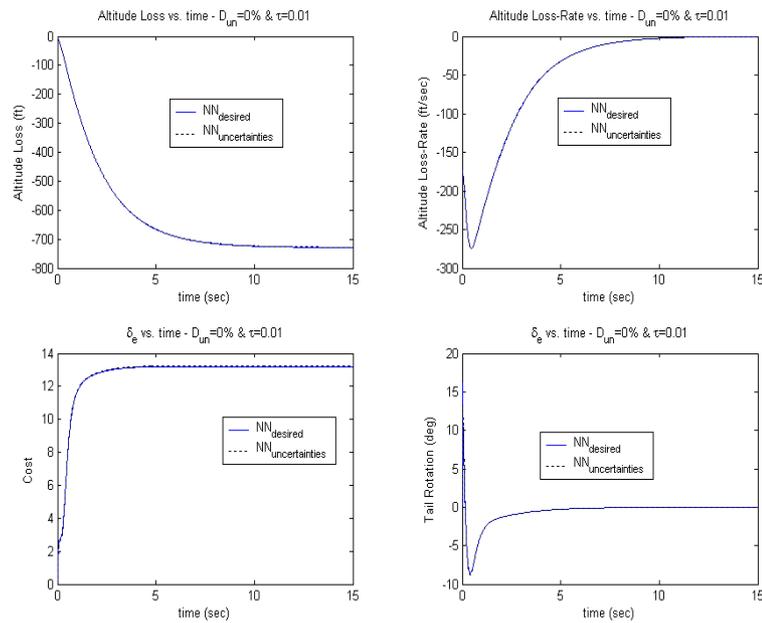


Figure 4.76. Altitude, Control and Cost Results ($\tau=0.01$, $\alpha(0)=30^\circ$, $\theta(0)=20^\circ$, $q(0)=5^\circ/\text{sec}$).

Figures 4.77 and 4.78 show the simulations for an increase in the time lag of $\tau=0.05$. The simulations show that the time lag slightly affects the control, as seen by its slower response in the bottom right hand corner of Figure 4.78. This effect can also be observed in the simulations for the pitch angle, pitch rate and flight path angle. The altitude loss difference between both controllers is almost negligible, but the difference in states is clearly present when obtaining the associated cost as seen in the bottom left corner of Figure 4.78.

The time lag is increased to $\tau=0.1$, and the results are displayed in Figures 4.79 and 4.80. The simulations show that the effects of the time lag in the states are greater than those described in the previous figures. The time lag in the control can now be seen very distinctively in the bottom right hand corner of Figure 4.80. The first few seconds of the simulations for the angle of attack, pitch angle, pitch rate and flight path angle, show the more accentuated difference between the desired Dual-NN controller and the Dual-NN-UU₁, as seen in Figure 4.79. The overall decrease in performance is observed

by the increase of associated cost of the controller with model uncertainties as seen in the bottom left corner of Figure 4.80.

Results for further increase in the time lag constant of $\tau=0.2$ are displayed in Figures 4.81 and 4.82. The larger time lag constant introduced in the simulation, clearly reflects that the commanded tail rotation is having problems trying to keep up with actual states of the aircraft during the initial portion of the recovery maneuver. Fluctuations of the states can be seen during the first 4 seconds of the recovery maneuver, after which the states normalize bringing the airplane to a zero-steady state. The higher loss in altitude is evident in the top left portion of Figure 4.82, while the associated cost shows the reduction in performance as the time lag constant is increased. Despite the decrease in performance, the Dual-NN-UU₁ is able to successfully bring the airplane to zero steady state conditions about the same time that the desired Dual-NN. These incredible results demonstrate the inherent robustness properties of the neurocontroller used in this thesis to the type of UU₁ described above.

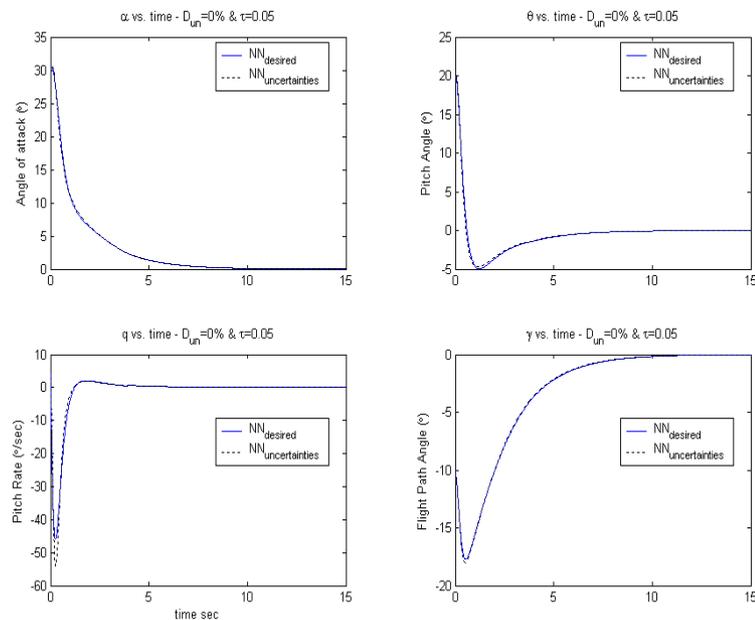


Figure 4.77. States Results ($\tau=0.05$, $\alpha(0)=30^\circ$, $\theta(0)=20^\circ$, $q(0)=5^\circ/\text{sec}$).

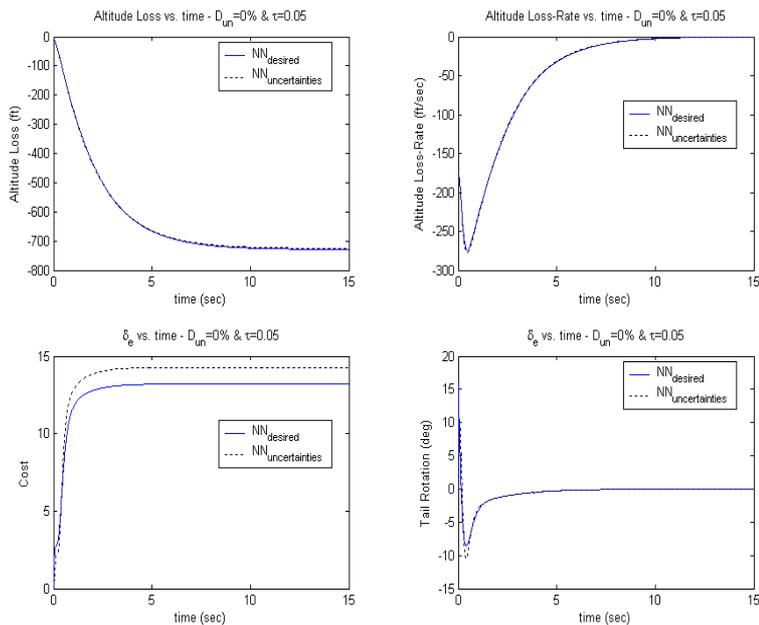


Figure 4.78. Altitude, Control and Cost Results ($\tau=0.05$, $\alpha(0)=30^\circ$, $\theta(0)=20^\circ$, $q(0)=5^\circ/\text{sec}$).

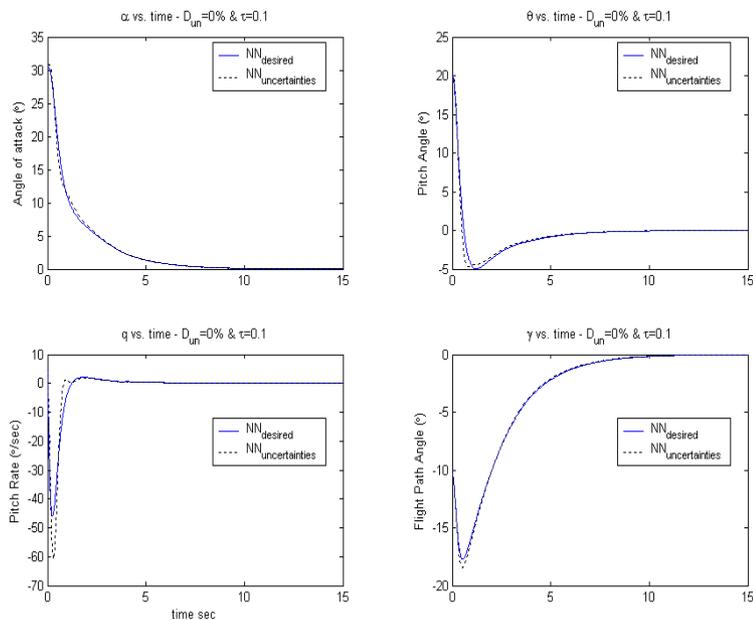


Figure 4.79. States Results ($\tau=0.1$, $\alpha(0)=30^\circ$, $\theta(0)=20^\circ$, $q(0)=5^\circ/\text{sec}$).

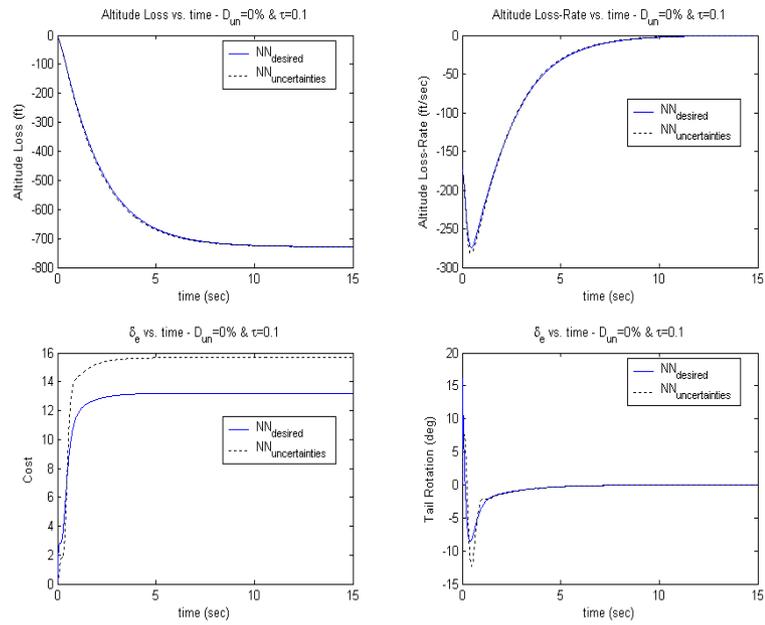


Figure 4.80. Altitude, Control and Cost Results ($\tau=0.1$, $\alpha(0)=30^\circ$, $\theta(0)=20^\circ$, $q(0)=5^\circ/\text{sec}$).

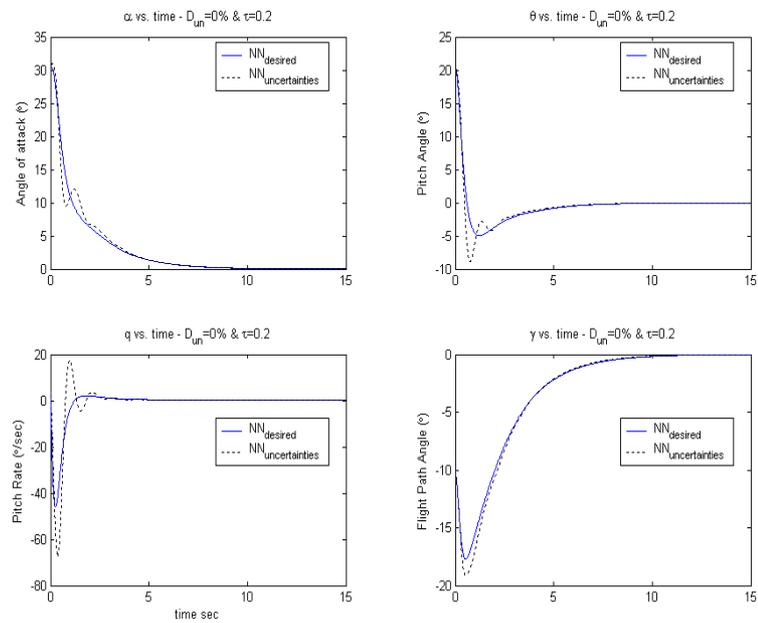


Figure 4.81. States Results ($\tau=0.2$, $\alpha(0)=30^\circ$, $\theta(0)=20^\circ$, $q(0)=5^\circ/\text{sec}$).

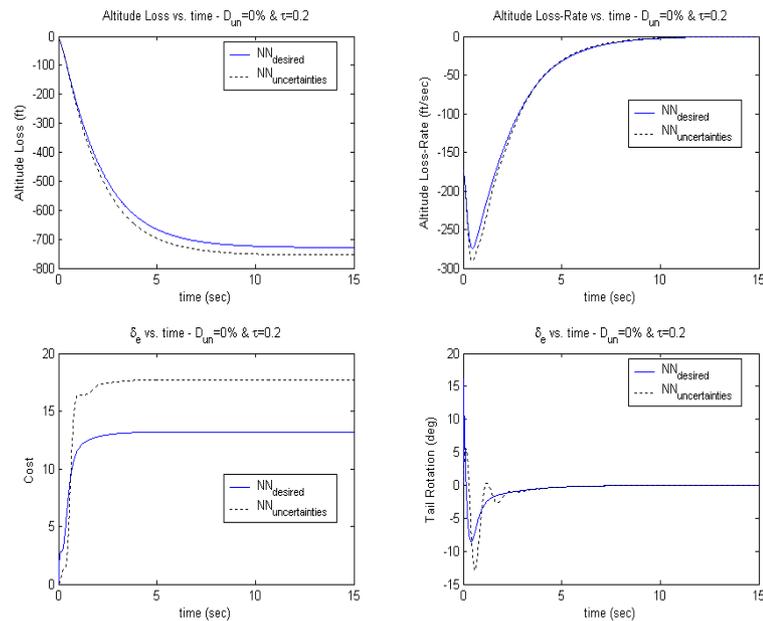


Figure 4.82. Altitude, Control and Cost Results ($\tau=0.2$, $\alpha(0)=30^\circ$, $\theta(0)=20^\circ$, $q(0)=5^\circ/\text{sec}$).

Figures 4.83 and 4.84 show the results for the extreme case of time lag uncertainties with a time constant of $\tau=0.4$, which although it is an extremely large time constant for a normal actuation system, the Dual-NN-UU₁ is able to adapt to those unmodeled uncertainties and recover from the initial stall conditions. The fluctuations of the states are even greater with this time constant as seen in Figure 4.83. The drop in altitude is even more accentuated, as seen in the top left portion of Figure 4.84, and the control time-history shows the fluctuation of the tail rotation trying to keep up with the states. Despite the fact that the Dual-NN-UU₁ brings the airplane to a zero-steady-state at approximately the same time as the desired Dual-NN, the oscillations encountered in the angle of attack, pitch angle and pitch rate during the first 7 seconds of the recovery maneuver, are considered to be a bit unrealistic when looking at the physical dynamics of an airplane. Again, despite the fact that the model used here neglects some dynamics, see section 4.2.1 for more detail, in order to focus in the nonlinearities in the high angle of attack, and considering that such simplifications are necessary in order to be able to have a reasonable model that would show the nonlinear capabilities of the neucontroller here

described, the author considers that despite that the Dual-NN shows an incredible performance, the results need to be taken with a grain of salt, and only those results that show reasonable airplane behavior, despite that the airplane model used in this work does not include these higher order dynamics. For the reasons mentioned above, Figure 4.83 and 4.84 do not represent acceptable results, but are presented to show the reader the limits of the Dual-NN to UU_1 .

The initial angle of attack is increased to $\alpha(0)=35^\circ$ and the time lag constant is lowered to $\tau=0.1$ in order to analyze the effects of the time lag as the initial angle of attack is increased. Figures 4.85 and 4.86 show the time-history simulations for the new conditions. As the initial angle of attack is increased to $\alpha(0)=35^\circ$, the time lag constant of $\tau=0.1$ affects in greater measure the response of the Dual-NN controller and the states as is seen when comparing simulations in Figures 4.85 and 4.86 with the simulations described previously for Figure 4.79 and 4.80. Figure 4.85 shows an overshoot of the angle of attack, pitch angle and pitch rate responses relative to the desired Dual-NN control solution during the first second of the recovery maneuver. The larger loss in altitude is observed on the top left corner of Figure 4.86, while the greater effort in tail rotation control relative to the desired Dual-NN controller is observed in the bottom right corner of the same figure. Despite the overall decrease in performance, the Dual-NN- UU_1 is still able to recover the airplane from the stall region.

Figures 4.87 and 4.88 show the state and control responses for an initial angle of attack of $\alpha(0)=35^\circ$, and a time constant of $\tau=0.2$. The extreme time lag constant and the high initial angle of attack make the controller to be extremely slow, and to show oscillations like those seen in Figure 4.83 and 4.84. Again, even though the Dual-NN- UU_1 is able to recover the airplane to zero-steady-state conditions the dynamics encountered are considered unacceptable. At this point the analysis of the robustness to time lag UU_1 is considered finished. Next section describes the analysis conducted to elevator effectiveness unmodeled uncertainties (UU_2).

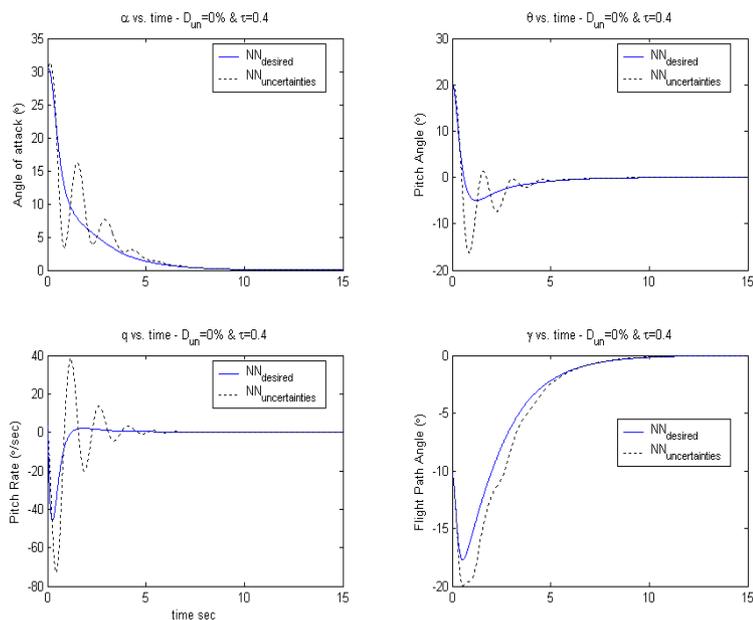


Figure 4.83. States Results ($\tau=0.4$, $\alpha(0) = 30^\circ$, $\theta(0) = 20^\circ$, $q(0) = 5^\circ/\text{sec}$).

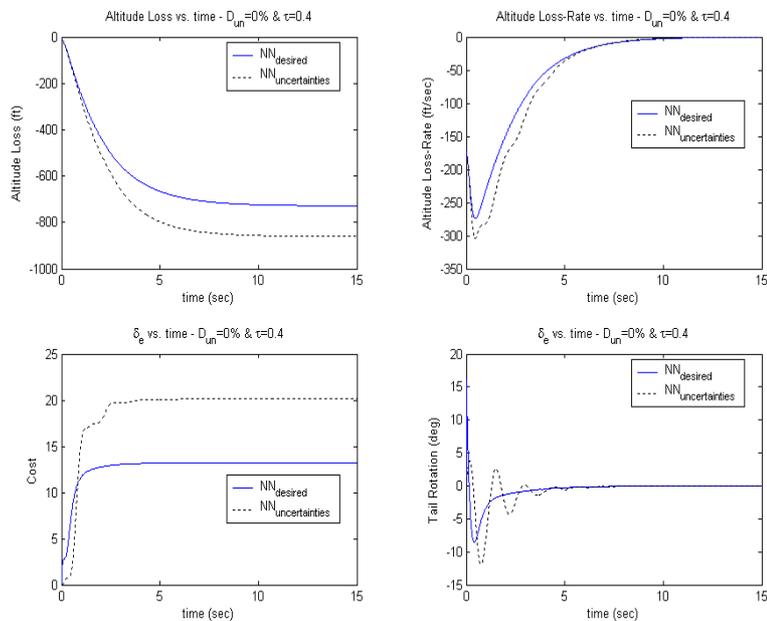


Figure 4.84. Altitude, Control and Cost Results ($\tau=0.4$, $\alpha(0) = 30^\circ$, $\theta(0) = 20^\circ$, $q(0) = 5^\circ/\text{sec}$.)

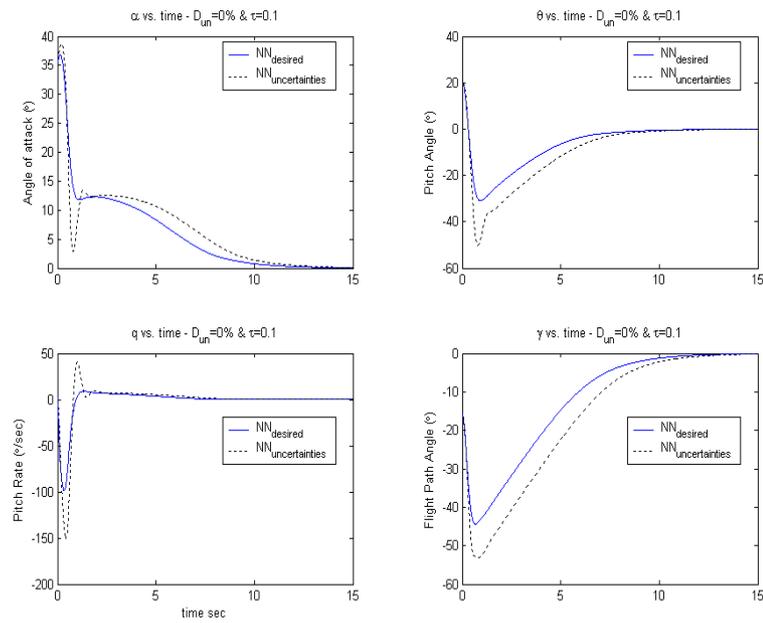


Figure 4.85. States Results ($\tau=0.1$, $\alpha(0)=35^\circ$, $\theta(0)=20^\circ$, $q(0)=5^\circ/\text{sec}$).

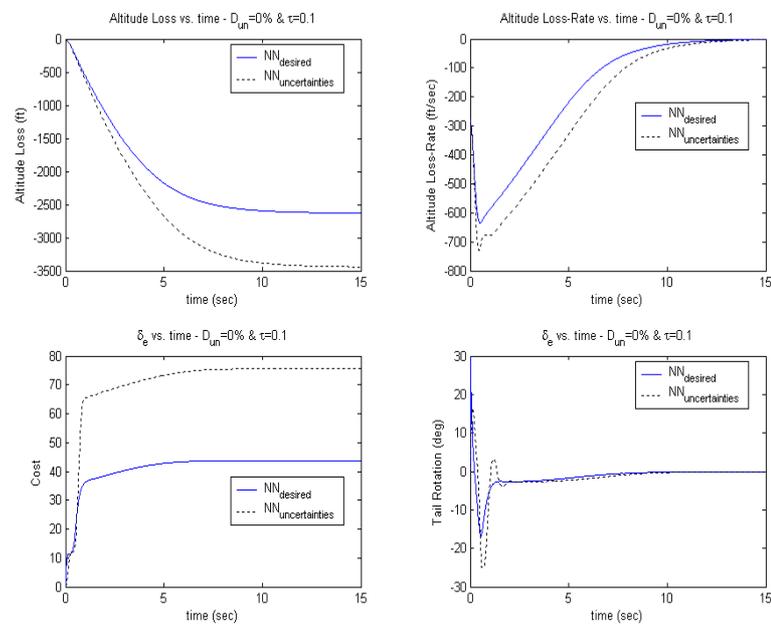


Figure 4.86. Altitude, Control and Cost Results ($\tau=0.1$, $\alpha(0)=35^\circ$, $\theta(0)=20^\circ$, $q(0)=5^\circ/\text{sec}$).

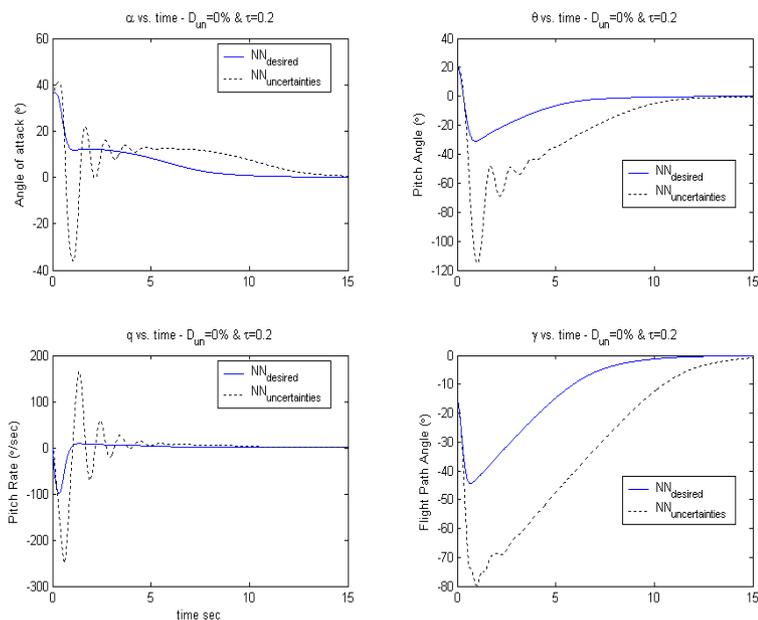


Figure 4.87. States Results ($\tau=0.2$, $\alpha(0) = 35^\circ$, $\theta(0) = 20^\circ$, $q(0) = 5^\circ/\text{sec}$).

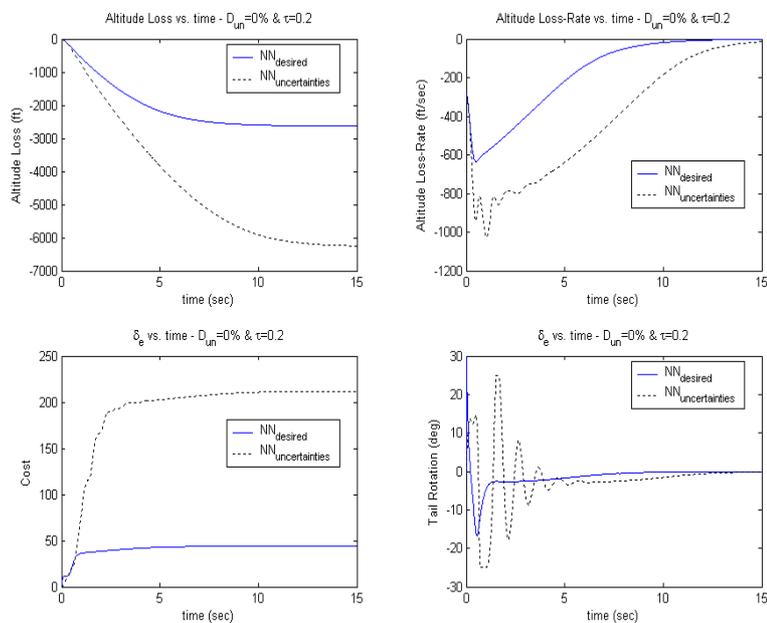


Figure 4.88. Altitude, Control and Cost Results ($\tau=0.2$, $\alpha(0) = 35^\circ$, $\theta(0) = 20^\circ$, $q(0) = 5^\circ/\text{sec}$).

In order to show the limits of the sensitivity study of the time lag unmodeled uncertainties, a parametric study was conducted for a set of initial angle of attack and varying the initial pitch angle and pitch rate, $\theta(0) = \pm 20^\circ$ and $q(0) = \pm 20^\circ/\text{sec}$ respectively. The maximum values of time lags allowed for each of the initial conditions before the simulations blow up are shown in Table 4.6.

Table 4.6. Maximum Allowable Time-lag Constants for Different Initial Pitch Angles and Pitch Rates for the Dual-NN Controller.

		Initial Pitch Rate $q(0)$								
		-20°/sec	-15°/sec	-10°/sec	-5°/sec	0°/sec	5°/sec	10°/sec	15°/sec	20°/sec
Initial Pitch Angle $\Theta(0)$	-20°	0.43	0.42	0.41	0.4	0.4	0.39	0.37	0.36	0.35
	-15°	0.47	0.46	0.45	0.44	0.44	0.43	0.41	0.39	0.39
	-10°	0.52	0.51	0.5	0.48	0.48	0.47	0.45	0.44	0.43
	-5°	0.6	0.58	0.56	0.53	0.53	0.52	0.49	0.48	0.46
	0°	0.69	0.66	0.63	0.62	0.59	0.58	0.57	0.54	0.5
	5°	0.79	0.78	0.74	0.71	0.67	0.65	0.61	0.58	0.57
	10°	0.96	0.91	0.87	0.84	0.79	0.73	0.72	0.67	0.62
	15°	1.22	1.15	1.06	0.99	0.96	0.89	0.8	0.74	0.67
	20°	1.66	1.68	1.5	1.25	1.22	1.1	0.97	0.87	0.78

Table 4.6 shows that the allowable time-lag constant increases as pitch rate decreases, and as pitch angle increases. The values range from 0.35 to 1.66, which represents an extreme time constant. Figure 4.89 shows a representation of the values tabulated above. It needs to be remember that the maximum allowable time-constants only represent the experimental results to test the limits of the sensitivity study. As mentioned earlier, beyond a maximum time-lag constant, the simulations show that the despite the Dual-NN brings the airplane back from the stall region, the magnitude and behavior of some encountered during the simulations need to be taken with a grain of salt due to the physical limitations of the model here used. Figures 4.90 and 4.91 show the simulations results for the states, altitude, control and cost time histories for the maximum allowable conditions above described.

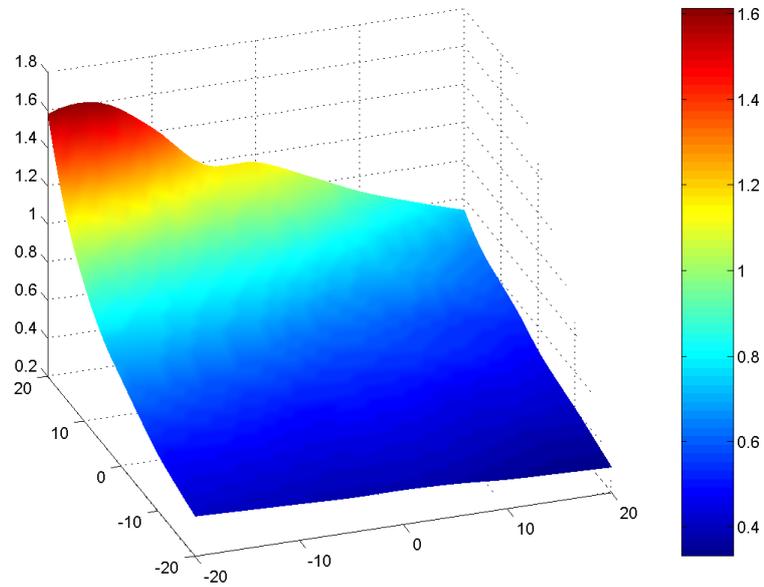


Figure 4.89. Maximum Allowable Time-lag Constant ($\alpha(0)=30^\circ$, $\theta(0)=\pm 20^\circ$, $q(0)=\pm 20^\circ/\text{sec}$).

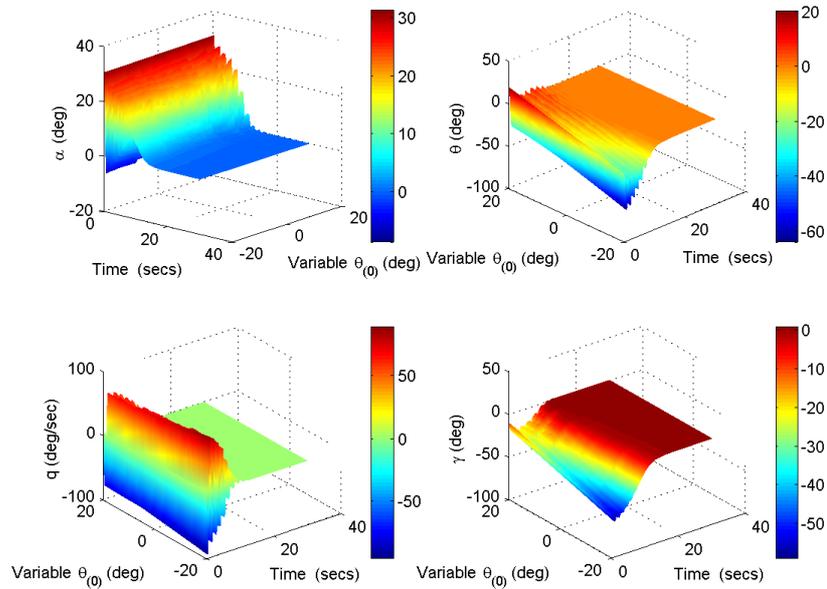


Figure 4.90. States Results ($\theta(0)=\pm 20^\circ$, $\alpha(0)=30^\circ$, $q(0)=5^\circ/\text{sec}$).

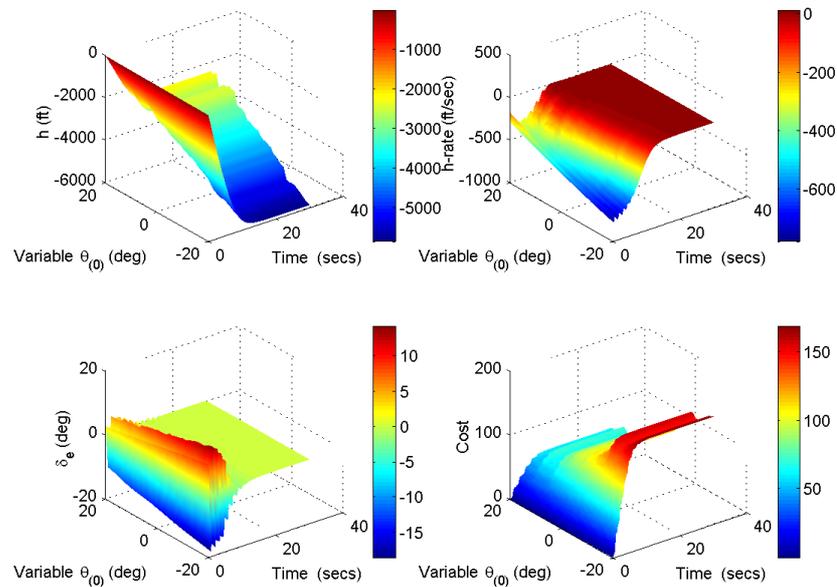


Figure 4.91. Altitude, Control and Cost Results ($\theta_{(0)}=\pm 20^\circ$, $\alpha_{(0)}=30^\circ$, $q_{(0)}=5^\circ/\text{sec}$).

4.2.8.2 Analysis of the sensitivity study to varying elevator effectiveness.

After the first part of the analysis of the inherent robustness of the Dual-NN controller is conducted, the analysis to tail effectiveness robustness is introduced by considering disturbances in the control matrix as described in Eq. (179). The variations in the percentage of decrease in tail effectiveness studied in this section vary from $D_{un}=\pm 50\%$ of the predicted tail effectiveness. During this portion of the study the time constant is assumed to be zero, and the initial pitch angle and pitch rate are kept constant at $\theta(0)=20^\circ$ and $q(0)=5^\circ/\text{sec}$ respectively.

Figures 4.92 and 4.93 show the simulations for a case of tail effectiveness UU_2 of $D_{un}=-0.10$, i.e. 10% decrease in tail effectiveness, and $\alpha(0)=30^\circ$. The simulation results show that a decrease in tail effectiveness of 10% is handled by the Dual-NN- UU_2 with only a small deviation from the desired Dual-NN trajectory. The major difference is observed in a larger loss of altitude by the Dual-NN- UU_2 of approximately 30 ft relative to the desired Dual-NN. Again, it needs to be emphasized that the Dual-NN- UU_2 has no

knowledge of the type, the magnitude, or even the existence of the disturbances in the model.

Figures 4.94 and 4.95 show the simulation results when the tail effectiveness is decreased by 25%. The differences between the desired Dual-NN and the Dual-NN-UU₂ are more apparent for the UU₂. The overall decrease in performance can be seen in both Figures 4.94 and 4.95, as the Dual-NN-UU₂ requires a bigger control effort in order to bring the airplane back to zero-steady-state. Figures 4.96 and 4.97 show simulations with a decrease in the tail effectiveness of 50%. The Dual-NN-UU₂ is still able to recover from the stall region despite the uncertainties generate a slower recovery to the zero-steady-state, as seen in Figure 4.96, and a larger drop in altitude as seen in the top portion of Figure 4.197. The cost increases considerably relative to the desired Dual-NN, which is expected, as the Dual-NN-UU₂ needs to generate a greater effort during the first instants of the recovery maneuver to bring the airplane back form the stall region.

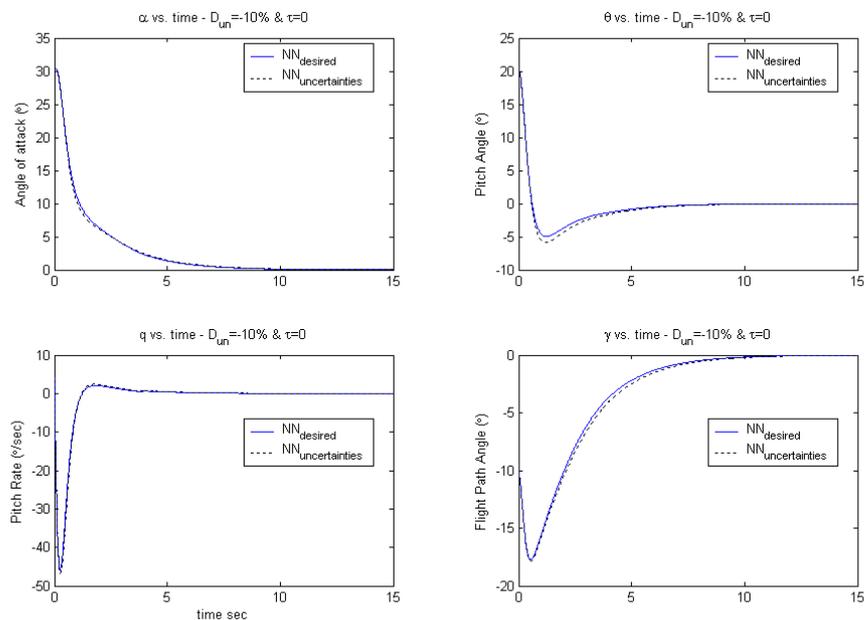


Figure 4.92. States Results ($D_{un} = -0.10$, $\alpha(0) = 30^\circ$, $\theta(0) = 20^\circ$, $q(0) = 5^\circ/\text{sec}$).

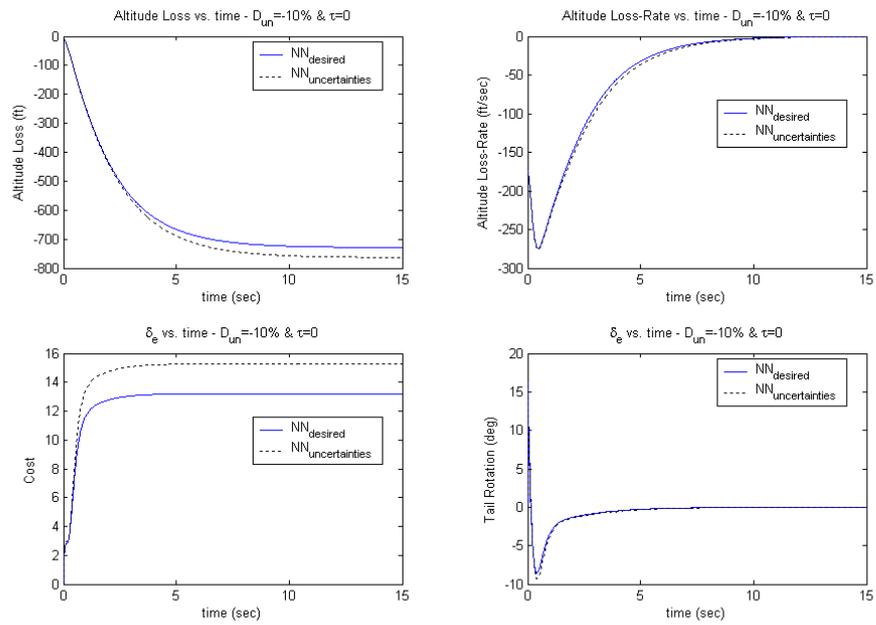


Figure 4.93. Altitude, Control and Cost Results ($D_{un} = -0.10$, $\alpha(0) = 30^\circ$, $\theta(0) = 20^\circ$, $q(0) = 5^\circ/\text{sec}$).

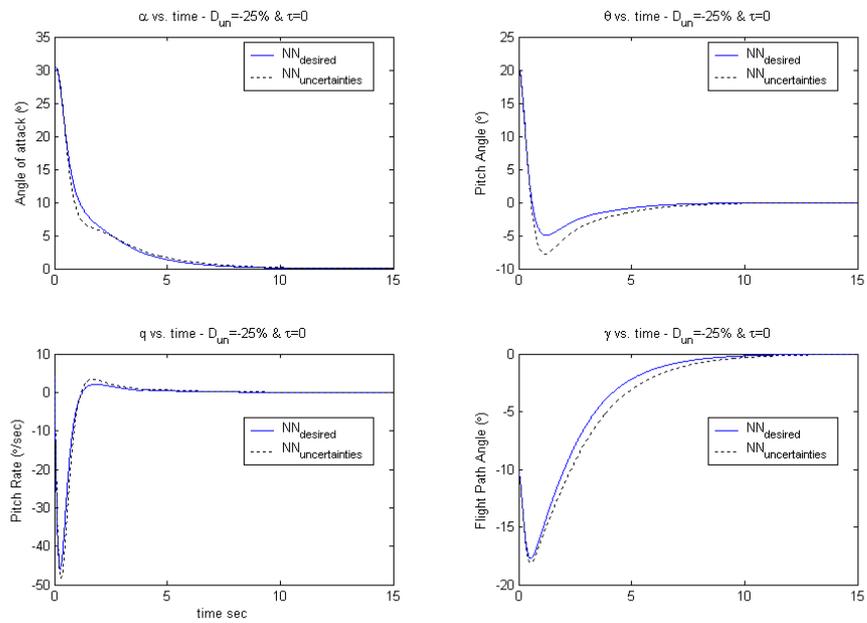


Figure 4.94. States Results ($D_{un} = -0.25$, $\alpha(0) = 30^\circ$, $\theta(0) = 20^\circ$, $q(0) = 5^\circ/\text{sec}$).

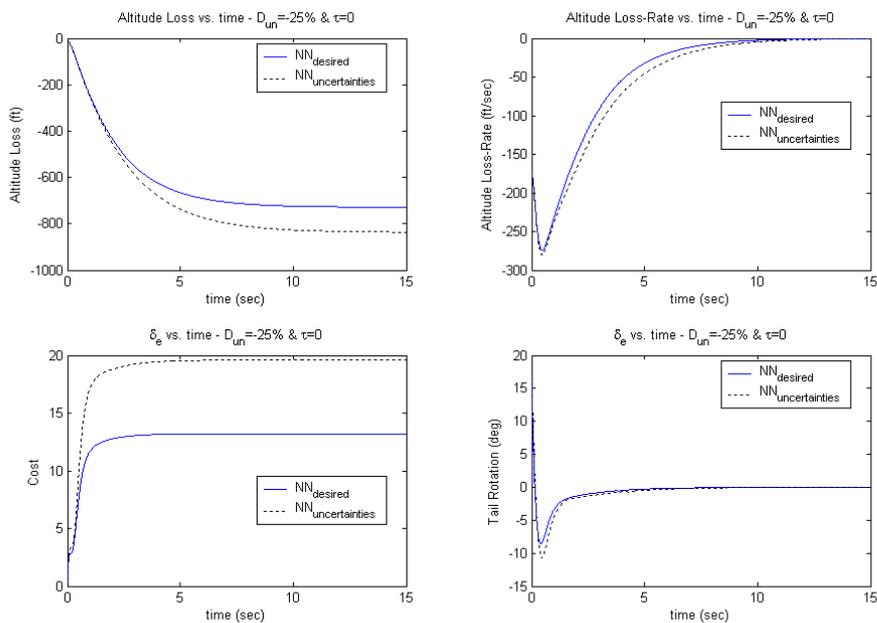


Figure 4.95. Altitude, Control and Cost Results ($D_{un} = -0.25$, $\alpha(0) = 30^\circ$, $\theta(0) = 20^\circ$, $q(0) = 5^\circ/\text{sec}$).

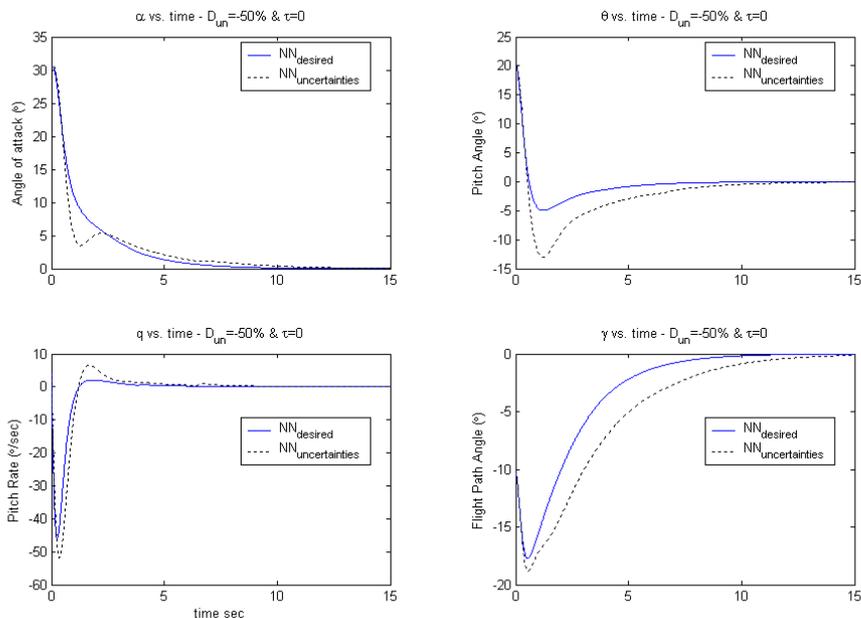


Figure 4.96. States Results ($D_{un} = -0.50$, $\alpha(0) = 30^\circ$, $\theta(0) = 20^\circ$, $q(0) = 5^\circ/\text{sec}$).

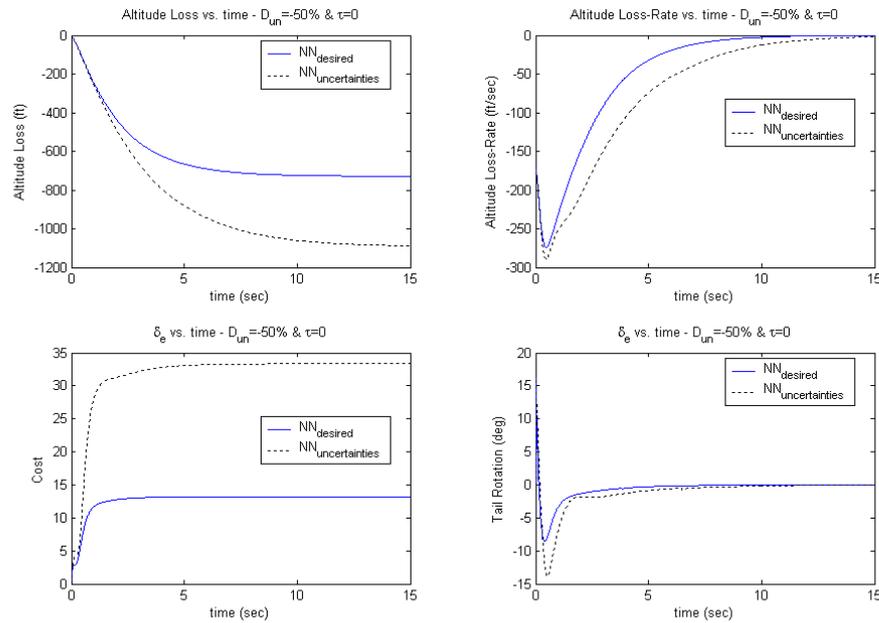


Figure 4.97. Altitude, Control and Cost Results ($D_{un}=-0.50$, $\alpha(0)=30^\circ$, $\theta(0)=20^\circ$, $q(0)=5^\circ/\text{sec}$).

Figures 4.98 and 4.99 show an extreme case of decrease in elevator effectiveness which is beyond the limits considered in this study. For the simulations shown in Figures 4.97 and 4.99, the tail effectiveness is decreased by 75%, $D_{un}=-0.75$. This analysis is conducted to show that despite the fact that Dual-NN-UU₂ is able to recover the airplane from the extreme UU₂ conditions, if the decrease in elevator effectiveness is associated to a loss in effective control area, these results might seem unrealistic in real life, and therefore unacceptable in this thesis. Again this is due to the fact that the Dual-NN-UU₂ shows an incredible inherent robustness to the type of UU shown in this thesis, but for the model described in section 4.2.1. Figures 4.100 and 4.101 show the simulation results for an increase of the initial angle of attack of $\alpha(0)=35^\circ$, and a tail effectiveness decreased of 50%. Again the oscillations encountered in the first instants, despite the Dual-NN-UU₂ being able to bring the airplane back to zero-steady-states, are unacceptable.

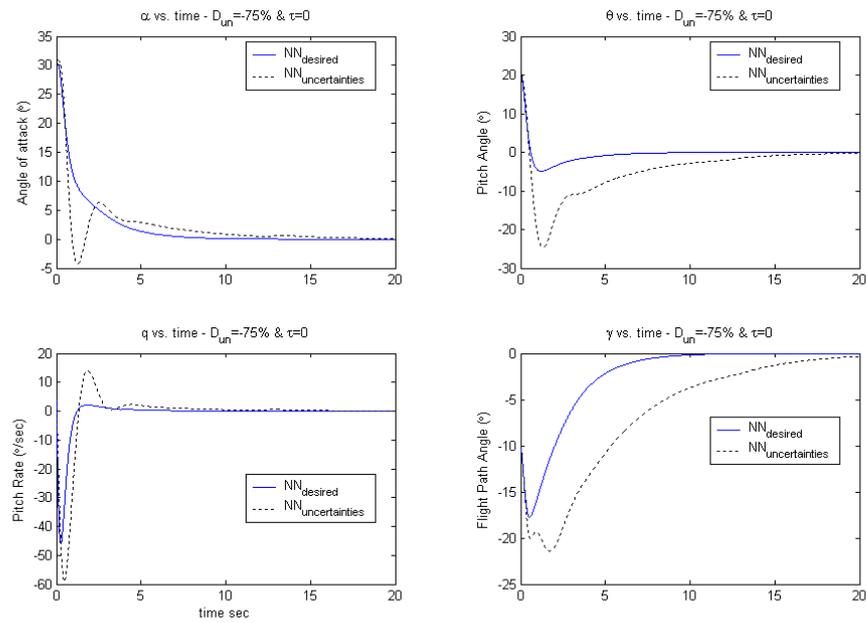


Figure 4.98. States Results ($D_{un} = -0.75$, $\alpha(0) = 30^\circ$, $\theta(0) = 20^\circ$, $q(0) = 5^\circ/\text{sec}$).

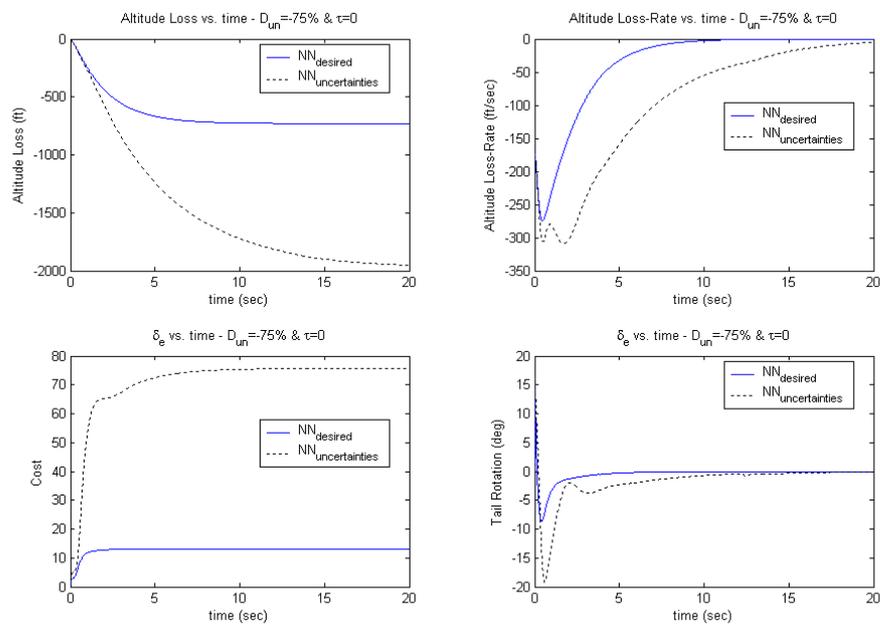


Figure 4.99. Altitude, Control and Cost Results ($D_{un} = -0.75$, $\alpha(0) = 30^\circ$, $\theta(0) = 20^\circ$, $q(0) = 5^\circ/\text{sec}$.)

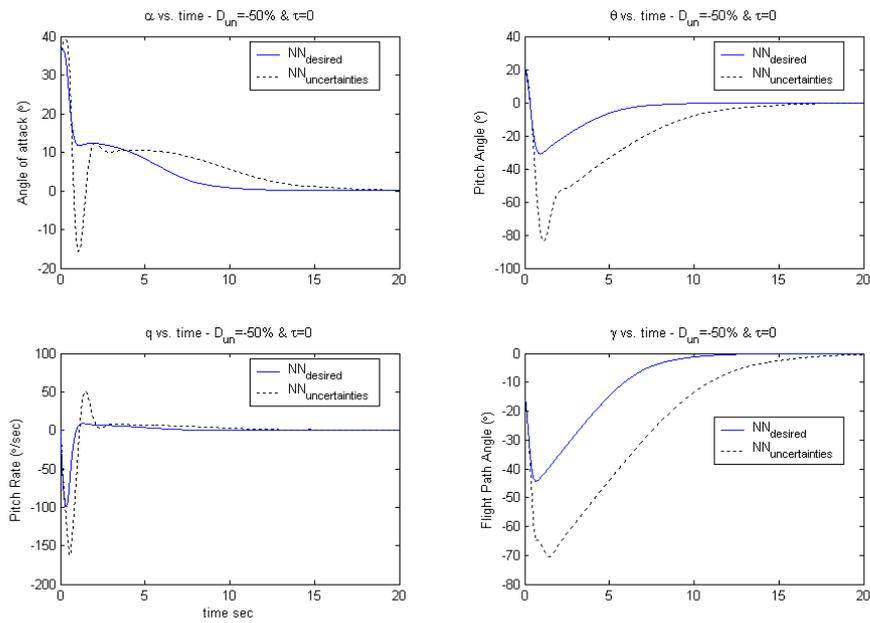


Figure 4.100. States Results ($D_{un} = -0.50$, $\alpha(0) = 35^\circ$, $\theta(0) = 20^\circ$, $q(0) = 5^\circ/\text{sec}$).

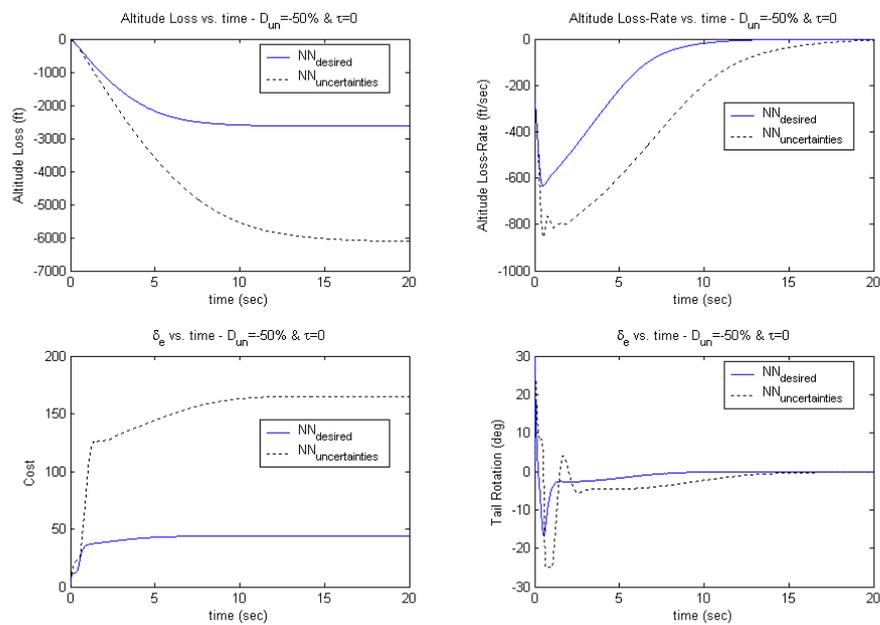


Figure 4.101. Altitude, Control and Cost Results ($D_{un} = -0.50$, $\alpha(0) = 35^\circ$, $\theta(0) = 20^\circ$, $q(0) = 5^\circ/\text{sec}$).

In order to check for the case in which the tail effectiveness has been underestimated and the uncertainties encountered imply an increase in the control authority, several cases of positive tail effectiveness coefficient are simulated. Figures 4.102 and 4.103 and Figures 4.104 and 4.105 show the simulations for an increase in the tail effectiveness of $D_{un}=0.25$ and $D_{un}=0.50$ respectively. Figures 4.102 and 4.104 show the states time histories and it can be seen that the states reach the steady-state of zero faster than the desired NN control. Figures 4.103 and 4.105 show that the controller requires less control effort in order to achieve the faster zero-steady-state condition.

The performance is also improved for the case with the increase in the tail effectiveness, which was expected but was necessary to check to make sure that an over control authority, caused by a wrong model, would not create an unstable configuration.

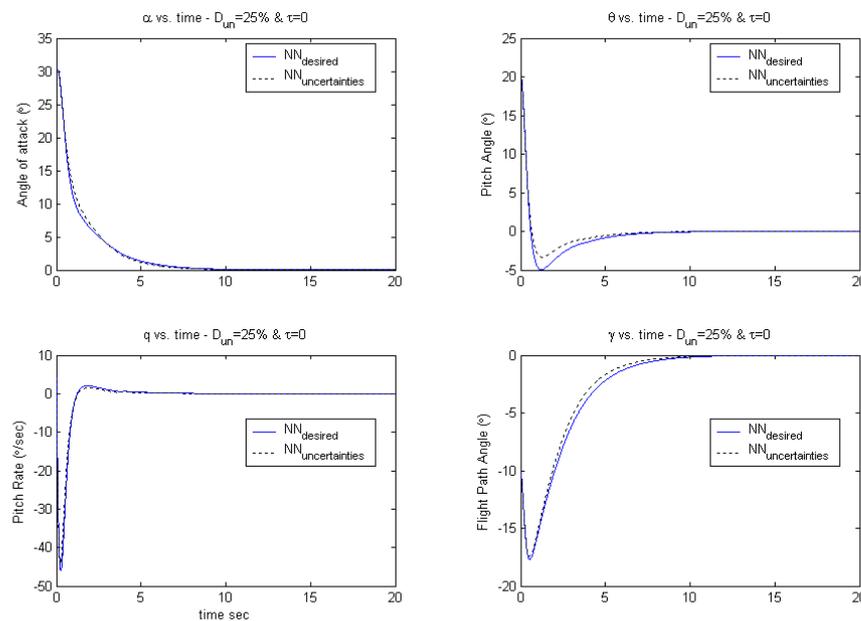


Figure 4.102. States Results ($D_{un}=0.25$, $\alpha(0)=30^\circ$, $\theta(0)=20^\circ$, $q(0)=5^\circ/\text{sec}$).

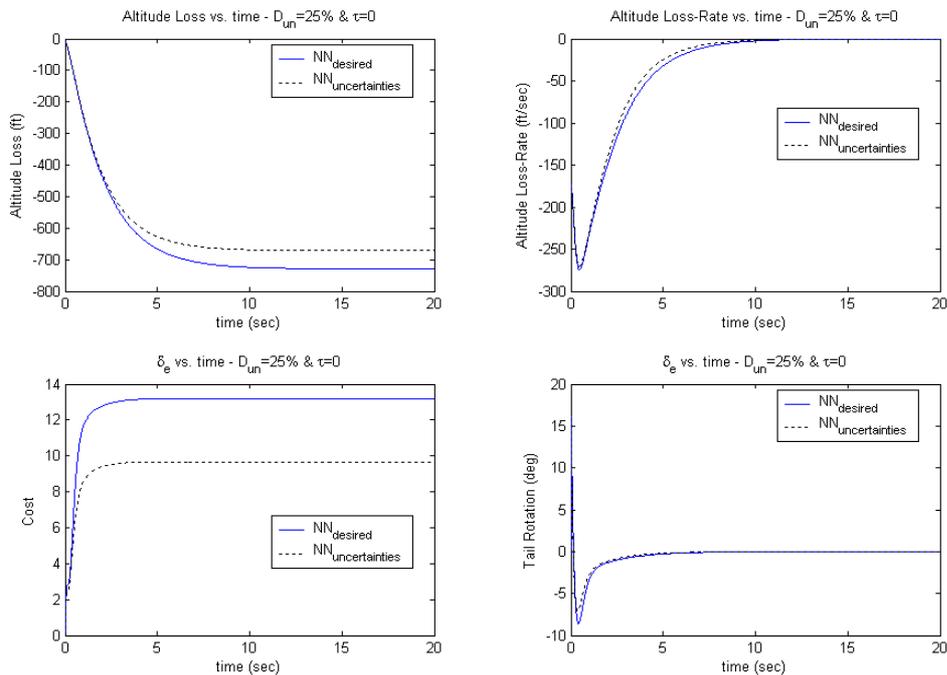


Figure 4.103. Altitude, Control and Cost Results ($D_{un}=0.25$ $\alpha(0)=30^\circ$, $\theta(0)=20^\circ$, $q(0)=5^\circ/\text{sec}$).

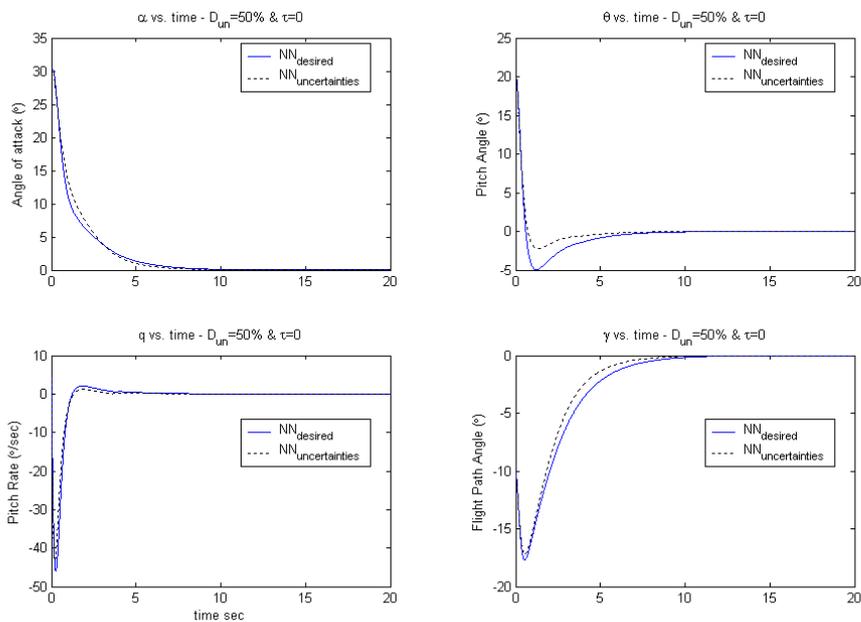


Figure 4.104. States Results ($D_{un}=0.50$, $\alpha(0)=30^\circ$, $\theta(0)=20^\circ$, $q(0)=5^\circ/\text{sec}$).

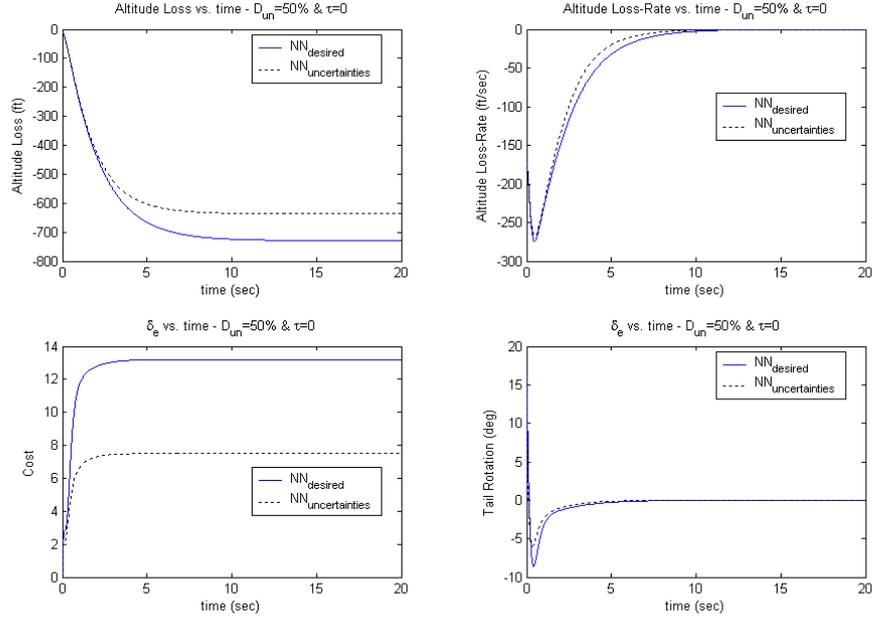


Figure 4.105. Altitude, Control and Cost Results
($D_{un}=0.50$, $\alpha(0)=30^\circ$, $\theta(0)=20^\circ$, $q(0)=5^\circ/\text{sec}$).

4.3. NON-LINEAR TRACKING PROBLEM.

This section extends the ACNN controller formulation to a tracking problem. The formulation used is based in a simplified model of the F-8 Crusader described in section 4.2.1 with the same initial flight conditions of level, unaccelerated flight at Mach=0.85, and an altitude of 30,000 ft.

4.3.1. Problem Formulation. The tracking problem described in this section, tries to track a commanded pitch rate, q_D using only the simplified short period dynamics, see section 4.1.5.1 for more details. The simplified nonlinear dynamics become:

$$\dot{\alpha} = -0.877\alpha + q - \alpha^2 q - 0.088\alpha q + 0.47\alpha^2 + 3.846\alpha^3 - 0.215\delta_e \quad (180)$$

$$\dot{q} = -4.208\alpha - 0.396q - 0.47\alpha^2 - 3.564\alpha^3 - 20.967\delta_e \quad (181)$$

The new state space model for the short period approximation can be represented by Eq. (182)

$$\dot{x}_1 = f(x_{SP}, \delta_e) = A_1 x_{SP} + \phi_1(x_{SP}) + b_1 \delta_e \quad (182)$$

where

$$x_{SP} = [\alpha \quad q]^T \quad (183)$$

$$A_1 = \begin{bmatrix} -0.877 & 1 \\ -4.208 & -0.396 \end{bmatrix} \quad (184)$$

$$b_1 = \begin{bmatrix} -0.215 \\ -20.967 \end{bmatrix} \quad (185)$$

$$\phi_1(x_{SP}) = \begin{bmatrix} -\alpha^2 q - 0.088\alpha q + 0.47\alpha^2 + 3.846\alpha^3 \\ -0.47\alpha^2 - 3.564\alpha^3 \end{bmatrix} \quad (186)$$

where x_1 is the simplified state vector for the simplified short period mode, A_1 represents the simplified plant matrix, b_1 represents the simplified control weight matrix and $\phi_1(x_1)$ represent the simplified nonlinear aerodynamics due to high angles of attack. Again it needs to be remember that the continuous-time equations defined above are discretized being consistent with the optimal control formulation.

4.3.2. Optimal Tracking Control Formulation. The optimal tracking control formulation as described in detail section 2.1.4, tries to find an admissible discrete-time control u_i^* , which causes the system $x_{i+1} = f(x_i, u_i)$ to follow an admissible trajectory r_i^* that minimizes the cost function defined in Eq. (1)

$$J = \sum_{i=0}^{N-1} L_i(x_i, u_i) \quad (1)$$

For the tracking problem the performance index, defined in Eq. (17) is substituted into Eq. (1) yielding the cost function to be minimized:

$$J = \frac{1}{2} \sum_0^{\infty} \left([x_i - r]^T Q_1 [x_i - r] + [u_i - u_D]^T R_1 [u_i - u_D] \right) \quad (187)$$

where for the problem defined in this section $x = x_{SP} = [\alpha \quad q]^T$, $r = [\alpha_D \quad q_D]^T$, $u = \delta_e$ and $u_D = \delta_{e,D}$. The weighting state and control matrices are defined similarly to those defined in Eqs. (138) and (139).

$$Q_1 = \begin{bmatrix} \frac{1}{\alpha_{\max}^2} & 0 \\ 0 & \frac{1}{q_{\max}^2} \end{bmatrix} \quad (188)$$

$$R_1 = \begin{bmatrix} \frac{1}{\delta_{\max}^2} \end{bmatrix} \quad (189)$$

In order to simplify the analysis the error dynamics state is introduced such that

$$e_x = [e_\alpha \quad e_q]^T \quad (190)$$

$$e_\alpha = \alpha - \alpha_d \quad (191)$$

$$e_q = q - q_d \quad (192)$$

$$\bar{\delta}_e = \delta_e - \delta_{e,D} \quad (193)$$

Substituting Eqs. (191)-(193) into Eqs. (180) and (181) yields the error dynamics:

$$\begin{aligned} \dot{e}_\alpha = & -0.877(e_\alpha + \alpha_D) + (e_q + q_D) - (e_\alpha + \alpha_D)^2(e_q + q_D) \\ & - 0.088(e_\alpha + \alpha_D)(e_q + q_D) + 0.47(e_\alpha + \alpha_D)^2 \\ & + 3.846(e_\alpha + \alpha_D)^3 - 0.215(\bar{\delta}_e + \delta_{e,D}) \end{aligned} \quad (194)$$

$$\begin{aligned} \dot{e}_q = & -4.208(e_\alpha + \alpha_D) - 0.396(e_q + q_D) - 0.47(e_\alpha + \alpha_D)^2 \\ & - 3.564(e_\alpha + \alpha_D)^3 - 20.967(\bar{\delta}_e + \delta_{e,D}) \end{aligned} \quad (195)$$

These continuous-time differential equations are discretized in order to be consistent with the discrete-time optimal tracking control formulation. The discrete-time cost function becomes

$$J = \frac{1}{2} \sum_0^{\infty} \left(e_{x(i)}^T Q_1 e_{x(i)} + \bar{\delta}_{e(i)}^T R_1 \bar{\delta}_{e(i)} \right) \quad (196)$$

The discrete-time Hamiltonian is defined as

$$H_i = \frac{1}{2} e_{x(i)}^T Q e_{x(i)} + \frac{1}{2} \bar{\delta}_{e(i)}^T R \bar{\delta}_{e(i)} + \lambda_{i+1}^T f_i(e_{x(i)}, \bar{\delta}_{e(i)}) \quad (197)$$

The discrete-time Euler-Lagrange necessary conditions for optimality define the state, co-state and the optimal control equation as

$$\lambda_i = \left(\frac{\partial H_i}{\partial e_{x(i)}} \right)^T \quad (198)$$

$$\frac{\partial H_i}{\partial \bar{\delta}_{e(i)}} = 0 \quad (199)$$

$$e_{x(i+1)} = \left(\frac{\partial H_i}{\partial \lambda_{i+1}} \right)^T \quad (200)$$

The discrete-time differential equations of the co-states of the tracking system are defined in Eqs. (201) and (202)

$$\lambda_{1(i)} = \frac{\partial H_i}{\partial e_{a(i)}} \quad (201)$$

$$\lambda_{2(i)} = \frac{\partial H_i}{\partial e_{q(i)}} \quad (202)$$

The necessary condition for optimal control, as defined in Eq. (199) leads to

$$\bar{\delta}_{e(i)} = \delta_{e(i)} - \delta_{e,D} = -R_1^{-1} b_1^T \lambda_{i+1} \quad (203)$$

and the discrete-time optimal control law becomes

$$\delta_{e(i)}^* = -R_1^{-1} b_1^T \lambda_{i+1} + \delta_{e,D} \quad (204)$$

Note that as mention earlier in section 2.1.4, analyzing the error dynamics described in Eqs. (194) and (195) and the co-state differential Eqs. (201) and (202), in order to solve the optimal tracking problem, the desired tracking states and steady state control are required , i.e. α_D , q_D and $\delta_{e,D}$. Obtaining the commanded pitch rate is not a problem since for this study q_D becomes the commanded state by the pilot/user, but obtaining the associated steady state angle of attack, α_D , and the steady state tail rotation, $\delta_{e,D}$, is a great challenge. Section 4.3.3 presents the method used to solve for the associated steady state angle of attack and tail rotation.

4.3.3. Calculation of the Steady State Values. In order to obtain the associated steady state values for the angle of attack and tail rotation the short period dynamics at the instant in which these conditions are attained are studied in more detail. At the exact instant when the desired commanded pitch rate is reached and maintained, the pitch rate derivative is zero, and the angle of attack and tail rotation are assumed to have zero rate, since they retain the steady state values of α_D and $\delta_{e,D}$. Considering this, the left hand side of Eqs. (180) and (181) are equal to zero yielding a set of two equations with two unknowns, α_D and u_D

$$0 = -0.877\alpha_D + q_D - \alpha_D^2 q_D - 0.088\alpha_D q_D + 0.47\alpha_D^2 + 3.846\alpha_D^3 - 0.215\delta_{e,D} \quad (205)$$

$$0 = -4.208\alpha_D - 0.396q_D - 0.47\alpha_D^2 - 3.564\alpha_D^3 - 20.967\delta_{e,D} \quad (206)$$

Since the desired pitch rate, q_D , is known, then α_D and $\delta_{e,D}$ need to be calculated by simultaneously solving Eqs (205) and (206). Equation (206) is solved in terms of $\delta_{e,D}$, yielding

$$\delta_{e,D} = -\frac{3.564}{20.967}\alpha_D^3 - \frac{0.47}{20.967}\alpha_D^2 - \frac{4.208}{20.967}\alpha_D - \frac{0.396}{20.967}q_D \quad (207)$$

Substituting Eq. (207) into Eq. (205) yields:

$$\begin{aligned} & \left(3.846 + \frac{0.76626}{20.967}\right)\alpha_D^3 + \left(0.47 + \frac{0.10105}{20.967} - q_D\right)\alpha_D^2 + \\ & \left(\frac{0.90472}{20.967} - 0.877 - 0.088q_D\right)\alpha_D + \left(1 + \frac{0.08514}{20.967}\right)q_D = 0 \end{aligned} \quad (208)$$

Equation (208) is of the form of a cubic polynomial with three roots that satisfy the equation for a given q_D . Using the numerical solution routine “roots.m” from MATLAB[®], the three roots for different q_D are obtained and analyzed to determine which one of the three roots is the correct one.

Figure 4.106 shows roots of Eq. (208) varying the commanded pitch rate from $q_D = \pm 15$ °/sec. It is observed that when the commanded pitch rate, q_D , is greater than 7.37496233 °/sec or smaller than -11.61497518 °/sec, the roots of the cubic polynomial present imaginary parts for the second and third root. Figure 4.105 shows the imaginary and real parts for the three roots after narrowing the limits of the q_D to those described above. This limits the effective range of commanded pitch rate for this tracking controller from -11.5 °/sec to 7.3 °/sec, since having an imaginary steady state angle of attack and tail rotation control makes no physical sense, thus limiting the effective envelope at which the tracking controller can work for the short period approximation.

After narrowing the range of q_D the results for the desired angle of attack, α_D , and its correspondent desired tail after solving Eq. (208) are displayed in Figures 4.108, 4.109 and 4.110 respectively for the three different roots.

Figure 4.111 represents the error generated by the numerical routine that solves for the roots of the cubic polynomial in Eq. (208), and Figure 4.112 describes the error satisfying the left hand side of Eqs. (205) and (206) for the obtained desired values.

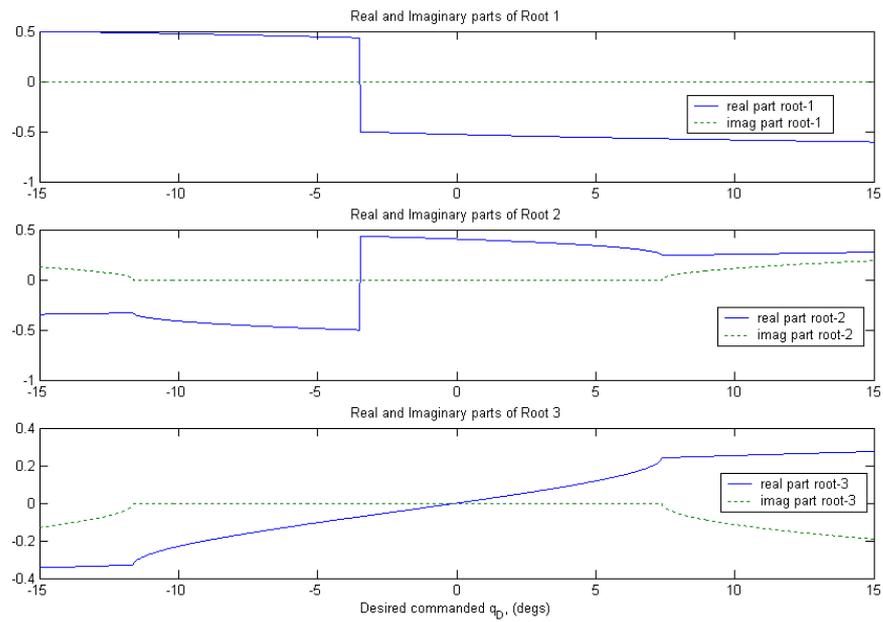


Figure 4.106. Real and Imaginary Parts of the Roots for Eq. (208), $q_D = \pm 15^\circ/\text{sec}$.

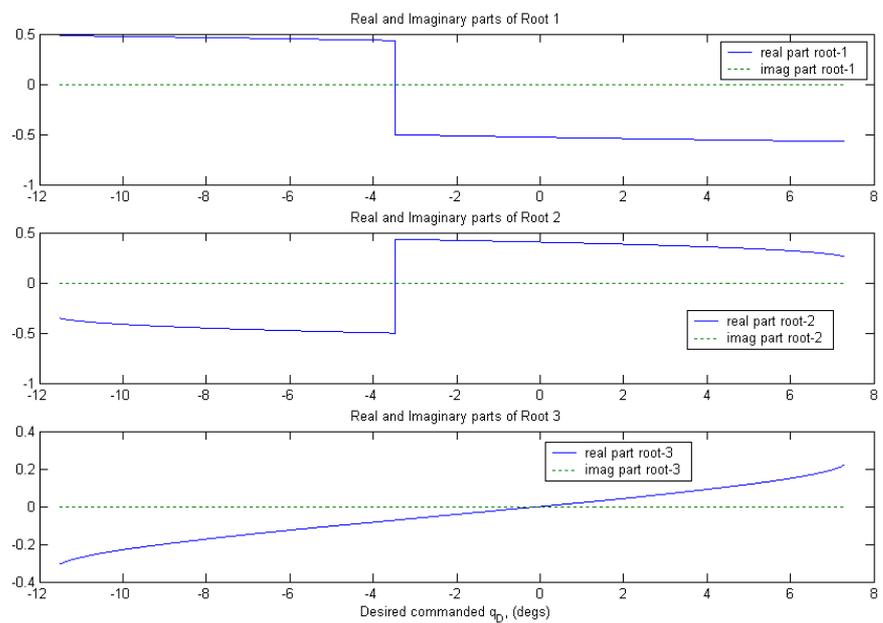


Figure 4.107. Real and Imaginary Parts of the Roots for Eq. (208) $q_D = -11.5$ to $+7.3$ $^\circ/\text{sec}$.

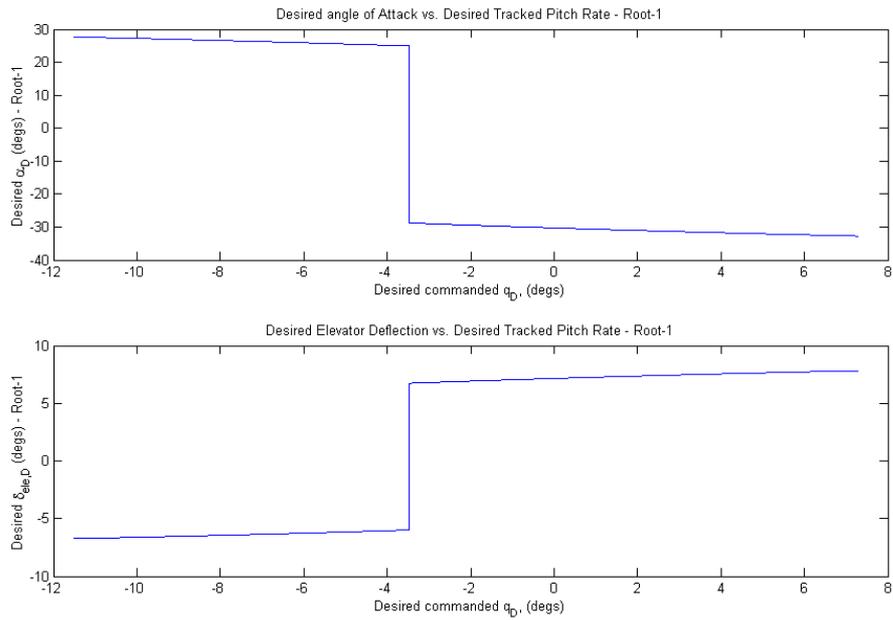


Figure 4.108. 1st Root of the solution to Eq. (208).

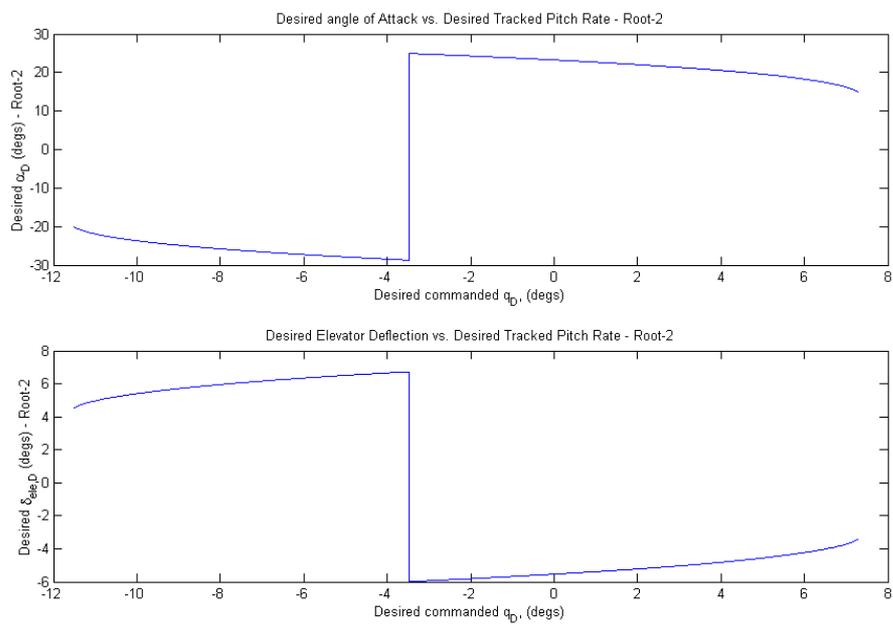


Figure 4.109. 2nd Root of the solution to Eq. (208).

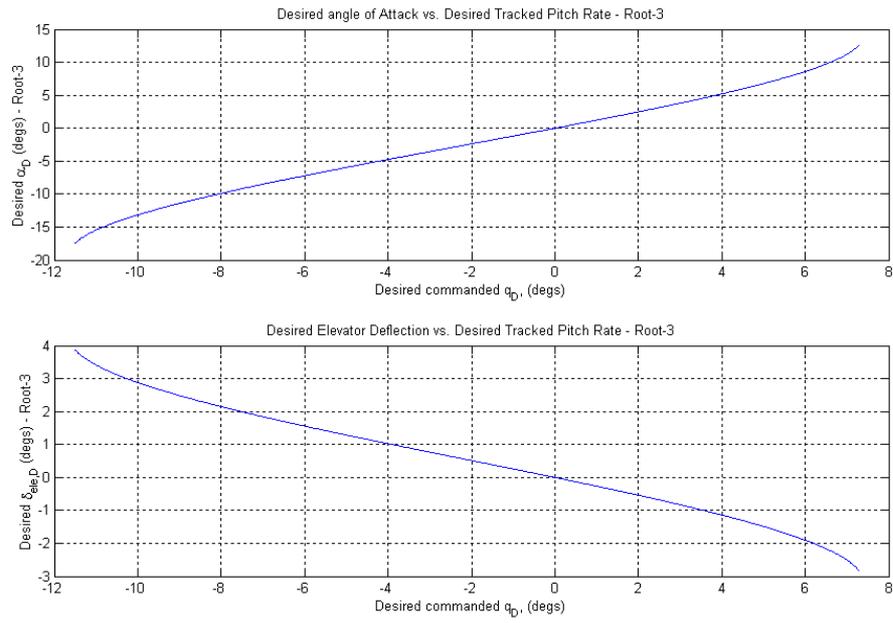


Figure 4.110. 3rd Root of the solution to Eq. (208).

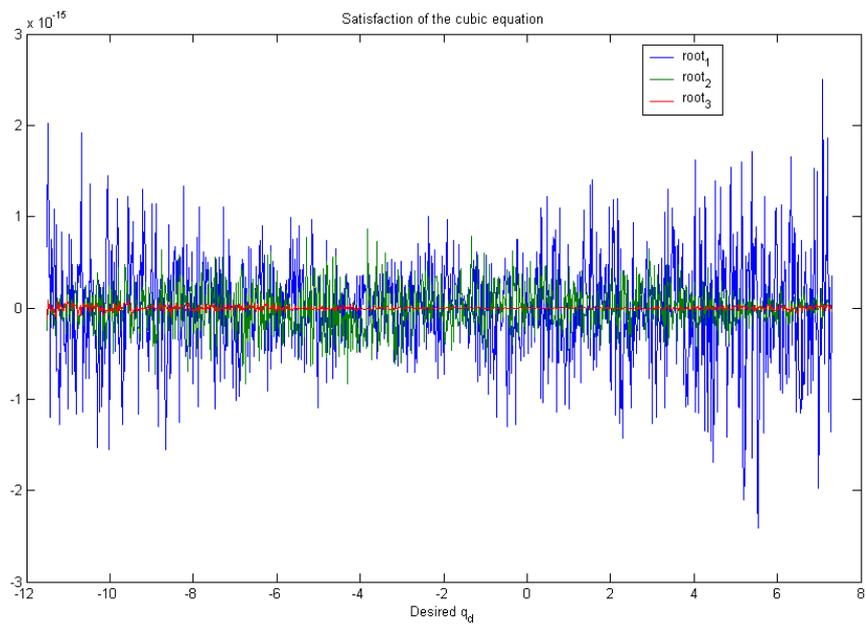


Figure 4.111. Satisfaction of the Cubic Polynomial in Eq. (208).

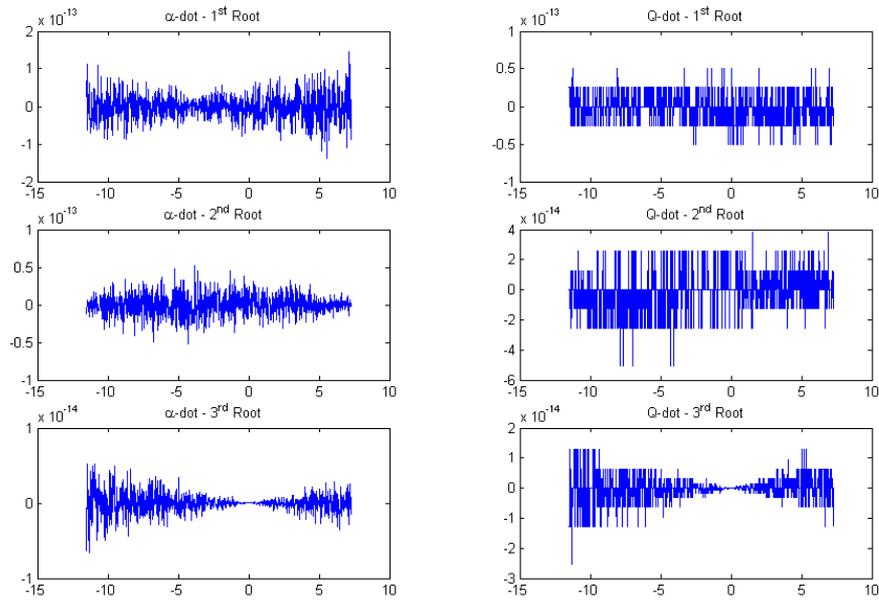


Figure 4.112. Satisfaction of the Left-hand Side of Eqs. (205) and (206).

In order to determine which root is to be used in the neurocontroller, an extensive analysis is done by first analyzing the physical implications of each one of the roots, and secondly, implementation of the three possible roots with a LQR version of the problem to determine their effectiveness. The discrete-time LQR version of the simplified short period dynamics, neglects the nonlinear terms in Eq. (182) such the model used is

$$x_{i+1} = A_1 x_i + b_1 \delta_{e(i)} \quad (209)$$

where the discrete-time LQR controller solution is defined as

$$\delta_{e(i)} = \bar{\delta}_{e(i)} + \delta_{e,D} = -R_1^{-1} B_1^T S e_{x(i)} + \delta_{e,D} \quad (210)$$

where S is the solution to the associated Riccati equation defined in Eq (22). Figure 4.113 shows a schematic of the model used to check the correctness of each one of the 3 roots.

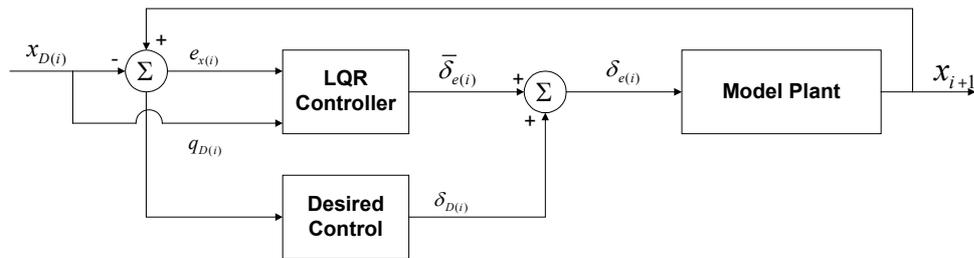


Figure 4.113. Model to Check the Correctness of the 3 Roots.

Analyzing the physical implications of the desired angles and their associated tail rotations for the three possible roots, it can be seen in Figures 4.108 and 4.109, which correspond to the solutions for the 1st and 2nd roots, that there exists a discontinuity for both roots at $q_D = -3.4832$ °/sec. This discontinuity divides the range of desired pitch rate in two sub-ranges, from $q_D = -11.5$ °/sec to $q_D = -3.4832$ °/sec for the first range, and from $q_D = -3.4832$ °/sec to $q_D = 7.3$ °/sec for the second range.

This discontinuity is evident when looking at the results provided by the first two roots. For the 1st root, as the desired pitch rate is increased from the negative value of -11.5 °/sec to the limiting value of $q_D = -3.4832$, the range in the associated desired angle of attack varies from $\alpha_D = 27.6889^\circ$ to $\alpha_D = 24.9393^\circ$, then the associated desired angle of attack jumps to approximately $\alpha_D = -28.7237^\circ$, which corresponds to a jump of more than 53° . For the second range of desired pitch rate, corresponding to desired pitch rates from $q_D = -3.4832$ °/sec to $q_D = 7.3$ °/sec, the range of the associated desired angle of attack goes from $\alpha_D = -28.7237^\circ$ to $\alpha_D = -32.7364^\circ$.

Similarly, for the 2nd root, as the desired pitch rate is increased from the negative value of -11.5 °/sec to the limiting value of $q_D = -3.4832$, the range in the associated

desired angle of attack varies from $\alpha_D=-20.219^\circ$ to $\alpha_D=-28.7141^\circ$, then the associated desired angle of attack jumps to approximately $\alpha_D=24.9314^\circ$, which corresponds to a jump of more than 53° . For the second range of desired pitch rate, corresponding to desired pitch rates from $q_D=-3.4832^\circ/\text{sec}$ to $q_D=7.3^\circ/\text{sec}$, the range of the associated desired angle of attack go from $\alpha_D=24.9314^\circ$ to $\alpha_D=-14.9269^\circ$. These discontinuities in the desired angle of attack in the 1st and 2nd roots affect the desired tail rotation since they are related by Eq. (207).

On the other hand, when comparing the previous results with those from the 3rd root, it is observed that the associated desired angle of attack goes continuously from $\alpha_D=-17.439^\circ$ to $\alpha_D=12.6827^\circ$ for the range desired pitch rate from $q_D=-11.5^\circ/\text{sec}$ to $q_D=7.3^\circ/\text{sec}$. In addition it is also observed that for a $q_D=0^\circ/\text{sec}$, the associated desired angle of attack is $\alpha_D=0^\circ$ and consequently $\delta_{e,D}=0^\circ$ which is what would you expect in order to maintain the trim conditions for zero desired pitch rate. This is not the case for the 1st and 2nd roots, where for a $q_D=0^\circ$ a desired $\alpha_D=-30.2863^\circ$ and $\alpha_D=23.2793^\circ$ are generated respectively, which does not make any sense.

Therefore, it can be concluded that the 1st and 2nd roots are bad candidates. In order to make sure that the above reasoning is correct, a series of simulations are conducted for a desired pitch rate of $q_D=5^\circ/\text{sec}$ using the LQR controller as baseline to determine the effectiveness of the three roots. The initial conditions assumed are $\alpha(0)=0$ and $q(0)=0^\circ/\text{sec}$, such as that before commanding a desired pitch rate, the airplane has first achieved zero-steady-state initial conditions. Table 4.7 shows the associated values of desired angle of attack and tail rotation for the three different roots.

Table 4.7. Associated Desired Values for a Commanded Pitch Rate of a $q_D=5^\circ/\text{sec}$.

	Commanded Pitch Rate ($^\circ/\text{sec}$)	Associated Angle of Attack ($^\circ$)	Associated Tail Rotation ($^\circ$)
1 st Root	$q_D=5^\circ/\text{sec}$	$\alpha_D=-32.0513^\circ$	$\delta_{e,D}=7.6411^\circ$
2 nd Root	$q_D=5^\circ/\text{sec}$	$\alpha_D=19.5618^\circ$	$\delta_{e,D}=-4.5577^\circ$
3 rd Root	$q_D=5^\circ/\text{sec}$	$\alpha_D=6.7702^\circ$	$\delta_{e,D}=-1.4872^\circ$

Figure 4.114 shows the time-histories simulations of the states and errors associated with the 1st root. The figure is divided in 4 subplots, where angle of attack and pitch rate are shown in the top left and right corners respectively. Each one of the time histories are compared with the desired value denoted by a dashed back line. The bottom left and right corners depict the error of the angle of attack and the pitch rate relative o the desired values. The units for the angle of attack and its associated error are degrees, while the units for the pitch rate and its associated error are degrees-per-second. From table 4.6 it can be seen that the associated desired values from the 1st root solution for a $q_D=5^\circ/\text{sec}$ are $\alpha_D=-32.0513^\circ$ and $\delta_{e,D}=7.6411^\circ$. It can be seen in the top portion of Figure 4.114, the angle of attack and the pitch rate are driven to steady states values that do not correspond to desired pitch rate of $q_D=5^\circ/\text{sec}$. This is expected according to the analysis previously conducted, in which the 1st root was considered not a good candidate. Furthermore, the steady state at which states are taken, $\alpha_D=1.9638^\circ$ and $q_D=1.5968^\circ/\text{sec}$, correspond to the expected values for solving Eq. (208) with the 3rd root for a $q_D=1.5968^\circ/\text{sec}$. Figure 4.115 shows the time history for the total tail rotation control, δ_e , and the control generated from the LQR, $\bar{\delta}_e$, and it can be seen that again, neither the tail rotation control goes to the desired elevator deflection associated with the 1st root of $\delta_{e,D}=7.6411^\circ$, nor the control from the LQR goes to zero. Again, the steady state for the total tail rotation, $\delta_{e,D}=-0.4262^\circ$, corresponds to the solution of Eq. (207) with the 3rd root for a $q_D=1.5968^\circ/\text{sec}$.

Figure 4.116 shows the time-histories simulations of the states and errors associated with the 2nd root. From table 4.6 it can be seen that the associated desired values from the 2nd root solution for a $q_D=5^\circ/\text{sec}$ are $\alpha_D=19.5618^\circ$ and $\delta_{e,D}=-4.5577^\circ$. It can be seen in the top portion of Figure 4.116, the angle of attack and the pitch rate are driven to steady states values that do not correspond to desired pitch rate of $q_D=5^\circ/\text{sec}$. This is expected according to the analysis previously conducted, in which the 1st root was considered not a good candidate. Furthermore, the steady state at which the states are taken, $\alpha_D=8.8183^\circ$ and $q_D=6.101^\circ/\text{sec}$, correspond to the expected values for solving Eq. (208) with the 3rd root for a $q_D=6.101^\circ/\text{sec}$. Figure 4.117 shows the time history for the total tail rotation control, δ_e , and the control generated from the LQR, $\bar{\delta}_e$, and it can be seen that again, neither the tail rotation control goes to the desired elevator deflection

associated with the 1st root of $\delta_{e,D} = -4.5577^\circ$, nor the control from the LQR goes to zero. Again, the steady state for the total tail rotation, $\delta_{e,D} = -1.9509^\circ$, corresponds to the solution of Eq. (210) with the 3rd root for a $q_D = 1.5968^\circ/\text{sec}$.

Figures 4.118 and 4.120 shows the time-histories simulations of the states, errors and controls, associated with the 3rd root. From table 4.6 it can be seen that the associated desired values from the 3rd root solution for a $q_D = 5^\circ/\text{sec}$ are $\alpha_D = 6.7702^\circ$ and $\delta_{e,D} = -1.4872^\circ$. Figure 4.119 shows that both the pitch rate and the angle of attack reach the desired values associated of $\alpha_D = 6.7702^\circ$ and $q_D = 5^\circ/\text{sec}$. Figure 4.119 shows that the 3rd root takes the tail rotation to the associated desired $\delta_{ele,D} = -1.4872^\circ$ and the error control, $\bar{\delta}_e$, goes to zero as an indication that the errors in the angle of attack and pitch rate are zero as seen in the bottom of Figure 4.118.

The simulation results, in addition to the analysis performed previously about the physical implications of the magnitude and shape of the roots relative to the envelope of desired pitch rates, demonstrates that the 3rd root is the correct choice to be used in generating the NN controller, which will be explained in more detail in the next section.

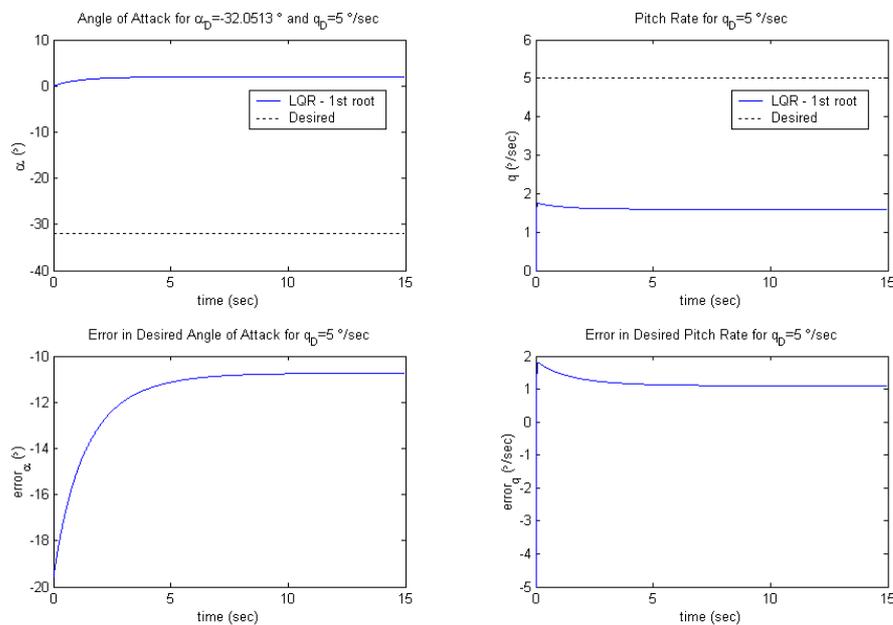


Figure 4.114 States and Errors, LQR Simulation for the 1st Root, $q_D = 5^\circ/\text{sec}$.

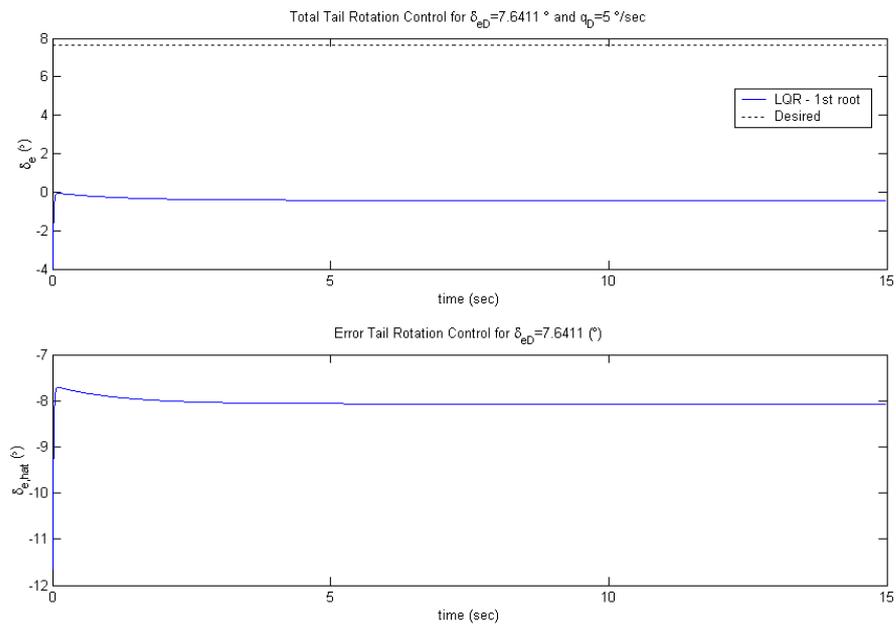


Figure 4.115. Total Elevator Control and $\bar{\delta}_e$, LQR Simulation for the 1st Root, $q_D=5^\circ/\text{sec}$.

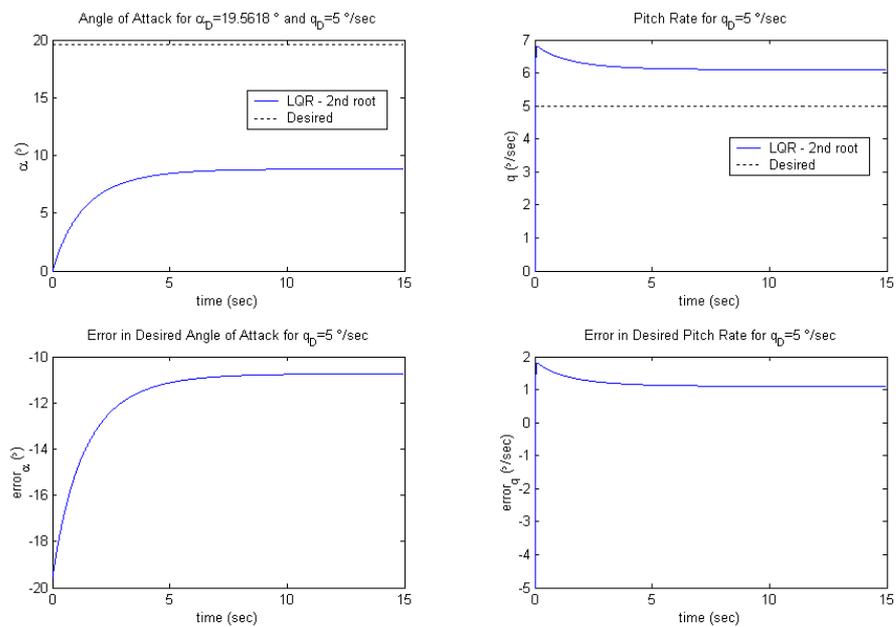


Figure. 4.116 States and Errors, LQR Simulation for the 2nd Root, $q_D=5^\circ/\text{sec}$.

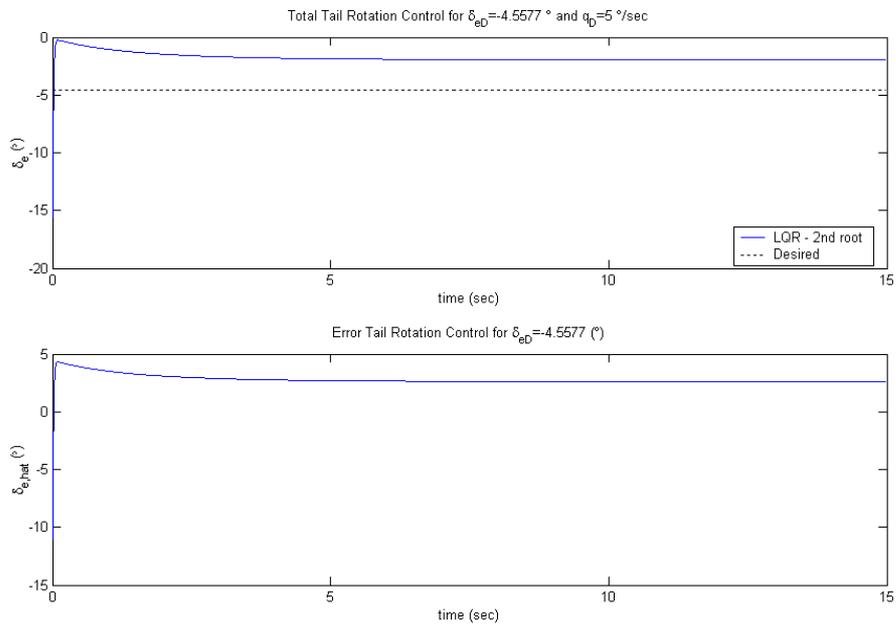


Figure 4.117. Total Elevator Control and $\bar{\delta}_e$, LQR Simulation for the 2nd Root, $q_D=5^\circ/\text{sec}$.

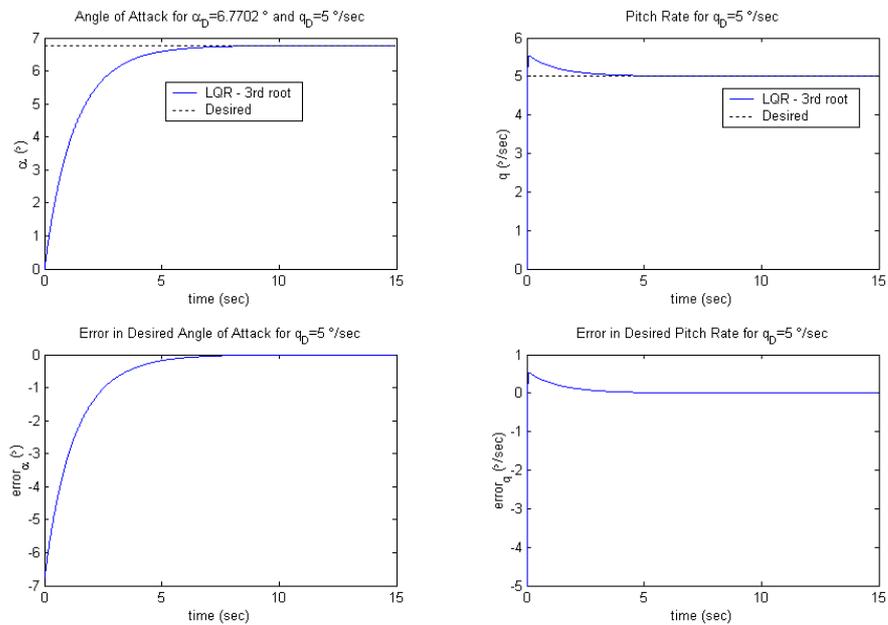


Figure 4.118. States and Errors, LQR Simulation for the 3rd Root, $q_D=5^\circ/\text{sec}$.

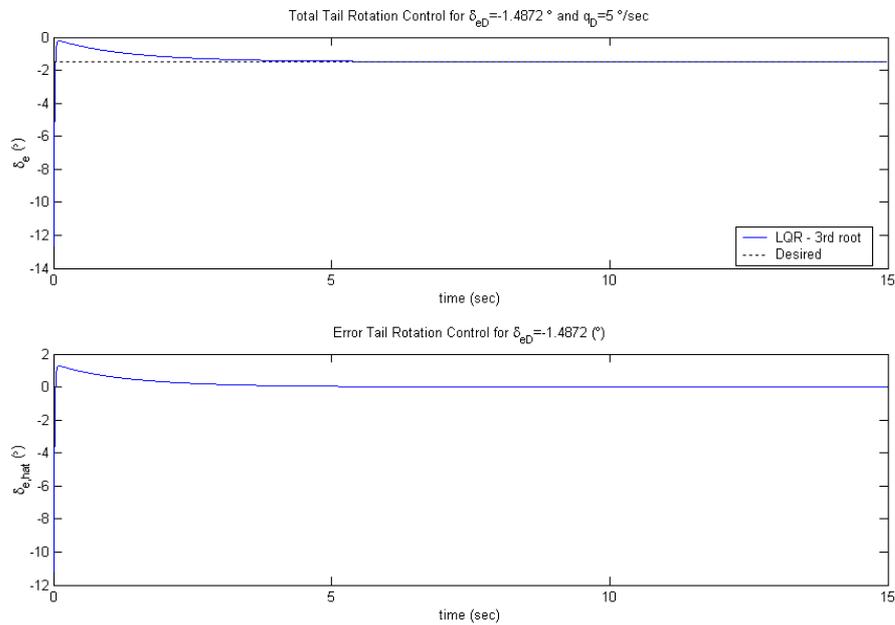


Figure 4.119. Total Elevator Control and $\bar{\delta}_e$, LQR Simulation for the 3rd Root, $q_D=5$ %/sec.

4.3.4. Neural Network Architecture. Recalling the neurocontroller development approach, described in section 3.3.1, and the NN architecture described in section 4.2.5.1, the Adaptive Critic NN is a feed forward back-propagation architecture, consisting of two hidden layers with hyperbolic tangent sigmoid transfer function, and an output layer with a linear transfer function.

The ANN architecture is defined $N_{3,4,4,1}$, i.e. 3 neurons corresponding to the three error state inputs, P, 6 neurons for the first and second hidden layers, and 1 neuron for the single control output, the error control, \bar{u} . The input to the NN is defined by

$$P = [e_\alpha \quad e_\theta \quad q_D]^T \quad (211)$$

where the errors are calculated at each time step using Eqs. (191) and (192). The CNN architecture is defined by $N_{3,6,6,2}$, i.e. 3 neurons corresponding to the three inputs, P , 6 neurons for the first and second hidden layers, and two co-states as outputs. Again, the activation function used in the two-hidden layers is the Hyperbolic Tangent Sigmoid transfer function, defined in Eq. (25). The training algorithm used to backpropagate the errors of the weight is the Levenberg-Marquardt training algorithm, and the training was implemented in MATLAB[®] using the Neural Network Toolbox.

Again the complexity and time-to-train increases with the number of neurons in the Action and Critic NN, so it was tried to train with the least amount of neurons that would still output good training results without compromising the time-to-train. Figures 4.120 and 4.121 show the diagrams that represent the schematics of the inputs to both the Action and the Critic NN, and their correspondent outputs.

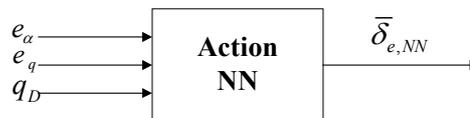


Figure 4.120. Schematic of the Input-Output Relation for the Action Neural Network

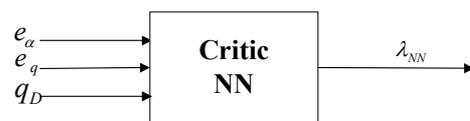


Figure 4.121. Schematic of the Input-Output Relation for the Critic Neural Network

4.3.5. Neural Network Training Approach. Following the same procedure as in section 4.2.5.2, the training ranges were determined after careful analysis of the results from the system simulation, and the results obtained in the nonlinear high angle of attack NN simulations and the robustness study conducted in the previous sections. As seen in

Figures 4.120 and 4.121, the inputs to the NN are in the form of the errors of the actual angle of attack and pitch rate to the desired values, and the commanded pitch rate, q_D . The range for the commanded pitch rate is set to slightly below the limits determined in section 4.3.3, $q_D = -11^\circ/\text{sec}$ to $q_D = 7^\circ/\text{sec}$.

The same telescoping strategy as described in section 3.2.2.3 was used for training both NN, slowly increasing q_D and the errors in the desired angle of attack and the desired pitch rate from the origin and expanding it to the set limits. For both the e_α and e_q the initial telescoping was started at $\pm 10^\circ$ and $10^\circ/\text{sec}$ respectively.

Analogous to the initial training procedure described in section 4.2.5.2, the first phase of the training procedure for the ANN and the CNN was to initialize both networks. In order to assure the convergence of the training procedure, instead of using a random initialization, it was decided to train both NN to map a percentage of the optimal linear solution for the linear model described in Eq. (203). By training the NN using as targets a percentage of the optimal solution it was assured that the initial NN would be in the direction of the optimal linear problem, thus being stable, but without providing the NN with the exact optimal solution which might bias the training procedure. Once the mapping initialization was achieved, the linear model was substituted by the nonlinear model described in Eq. (200). The initial percentage used for the mapping initialization was 80 so that the Action NN network would map a control of the form

$$\bar{\delta}_{e(i)} = -\Delta K e_{x(i)} \quad (212)$$

where Δ represents the percentage of the direction, for this training procedure being $\Delta = 0.80$. Figure 4.122 shows the schematics of the ANN such that the inputs would be the two errors and the desired pitch rate.

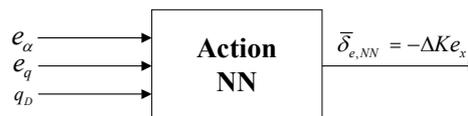


Figure 4.122. Initialization of the Action Neural Network.

The Critic was initialized in the same manner using the solution to Riccati equation, where the optimal linear co-state history is defined by:

$$\lambda_{(i)} = S e_{x(i)} \quad (213)$$

where S is the solution to the Riccati Eq. (40). The Critic NN network was initialized such that it would map the Lagrange multipliers as a percentage of the optimal linear solution as it is shown in Figure 4.123.

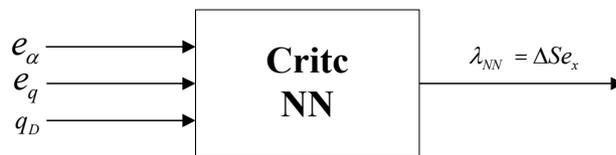


Figure 4.123. Initialization of the Critic Neural Network.

4.3.6. Training Synthesis of the Neurocontroller. As described in section 4.2.3, Steps 1 through 5 for the Action NN and Critic NN are repeated until the desired level of accuracy for the NNs is achieved and there is no acceptable change in the outputs of the trained NN after a cycle. For this problem the level of accuracy is reached when the 2-norm of the errors between the current outputs and the previous trained outputs is below $e < \varepsilon = 0.008$. Once the training was considered finished the ANN was tested as a feedback controller of the form described in Figure 4.124.

4.3.6.1 Training synthesis of the action neurocontroller.

1. The initial CNN is *assumed* to be optimal.
2. The initial ANN output, $\bar{\delta}_{e(i)}$, is obtained by feeding random values of the states $e_{x(i)}$ and their associated commanded pitch rate, q_D , to the ANN.

3. The discrete-time nonlinear differential equations of the error dynamics, Eqs. (194) and (195), are used to integrate forward to obtain $e_{x(i+1)}$, and the associated q_D using the states $e_{x(i)}$ and the output $\bar{\delta}_{e(i)}$ of the ANN.
4. The Critic NN is feed the output form step 3, $e_{x(i+1)}$ and the associated q_D , to calculate the Lagrange multiplier, $\lambda_{NN(i+1)}$, associated with $e_{x(i+1)}$.
5. The ANN is then trained using $e_{x(i)}$ and the associated q_D as input and the optimal control, and $\bar{\delta}_{e(i)}^*$ from Eq. (203), as target: $\bar{\delta}_{e(i)}^* = -R_1^{-1}b_1^T \lambda_{i+1}$

Steps 1 through 5 are repeated until the desired level of accuracy for the ANN is achieved. For the nonlinear tracking problem the level of accuracy is reached when the 2-norm of the errors between the current ANN outputs and the previous trained outputs are below 0.001.

4.3.6.2 Training synthesis of the critic neurocontroller.

1. The ANN is *assumed* to be optimal.
2. The initial output $\bar{\delta}_{e(i)}$, is obtained by feeding random values of the states $e_{x(i)}$ and the associated q_D , to the ANN.
3. The discrete-time nonlinear differential equations of the error dynamics, Eqs. (194) and (195), are used to integrate forward to obtain $e_{x(i+1)}$, and the associated q_D using the states $e_{x(i)}$ and the output $\bar{\delta}_{e(i)}$ of the ANN.
4. The Critic NN is feed $e_{x(i+1)}$ and their associated q_D , to calculate $\lambda_{NN(i+1)}$.
5. The discrete-time nonlinear differential equations for the error dynamics ,Eqs. (194) and (195) and the correspondent co-state error dynamic differential equations, Eqs.(201) and (202), are integrated backwards in time to obtain $\lambda_{(i)}^*$.
6. The CNN is then trained using $e_{x(i)}$ and the associated q_D as inputs and $\lambda_{(i)}^*$ from step 4 as target.

Steps 1 through 6 are repeated until the desired level of accuracy for the CNN is achieved. For the nonlinear tracking problem the level of accuracy is reached when the 2-norm of the errors between the current CNN outputs and the previous trained outputs are below an established error of 0.001.

After the NN has been trained it is implemented in the form of a feedback controller as described in Figure 4.124.

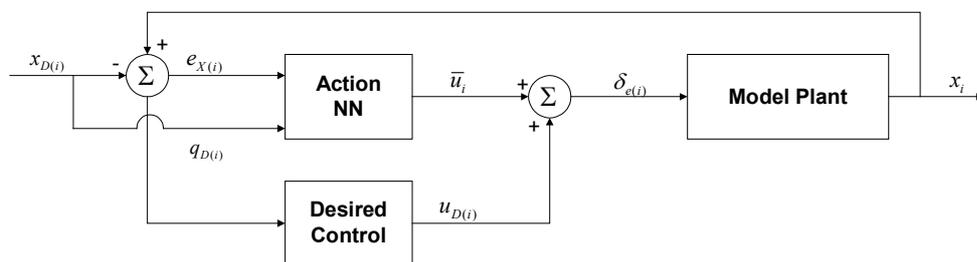


Figure 4.124. Implementation of the Neural Network Tracking Feedback Controller.

4.3.7. Analysis of Numerical Results. The simulation results are divided in two parts. In the first part, single commanded desired pitch rates are tested within the range of available commanded pitch rates, and in the second part of the analysis, multiple commanded pitch rates are introduced.

Figures 4.125 and 4.126 show the time-histories simulations for a commanded pitch rate of $q_D = -2.5^\circ/\text{sec}$. Figure 4.125 is divided in 4 subplots, where angle of attack and pitch rate are shown in the top left and right corners, respectively. Each one of the time histories are compared with the desired value denoted by a dashed back line. The bottom left and right corners depict the error of the angle of attack and the pitch rate relative to the desired values. The units for the angle of attack and its associated error are degrees, while the units for the pitch rate and its associated error are degrees-per-second.

Figure 4.126 is divided in two subplots, being the error control from the NN $\bar{\delta}_e$ located in the top portion, and the total tail rotation at the bottom, both with units of

degrees. It is seen that the NN controller successfully tracked the desired pitch rate, and that the associated desired angle of attack and total tail rotation also reach the desired steady states. The NN is compared with the LQR controller, and it is observed that the NN performs equally to the LQR controller. This is the general trend that will be seen through the remainder of the tracking analysis, and is due to the fact that the nonlinearities described in section 4.3.1 only come into importance significance when the operating ranges of angle of attack are above the stall conditions. Despite that the controller always operates assuming that the airplane has always reached a zero-steady-state angle of attack and pitch rate, before commanding a desired pitch rate, the results described in this section show the good performance of the tracking system. Figure 4.127 shows the cost comparison between the NN and the LQR solution and it is seen that the Dual-NN offers a slightly lower cost.

Figures 4.128, 4.129, and 4.130 show the simulation results for a commanded pitch rate of $q_D=2.5^\circ/\text{sec}$. Figures 4.131, 4.132, and 4.133 show the simulation results for $q_D=-11^\circ/\text{sec}$. Figures 4.134, 4.135, and 4.136 show the simulation results for $q_D=7^\circ/\text{sec}$. All states simulations, Figures 4.128, 4.131 and 4.134 show that the NN controller is able to track the desired pitch rate with a slightly better performance relative to the LQR as seen in the cost time history, Figures 4.130, 4.133 and 4.136 respectively. The better performance can also be seen on the control time history shown in Figures 4.129, 4.132 and 4.135. The Dual-NN controller shows a smoother commanded tail rotation than the LQR controller.

After the simulations for several single commanded pitch rates, the tracking formulation is extended to multiple commanded pitch rates in the same simulation. Two combinations are analyzed, the first starting with a commanded pitch rate of $q_D=-5^\circ/\text{sec}$ for 7.5 seconds and then changing to $q_D=5^\circ/\text{sec}$. The second combination uses three different commanded pitch rates, being the first 7.5 seconds $q_D=3^\circ/\text{sec}$, the second set of 7.5 seconds $q_D=-2.5^\circ/\text{sec}$, and $q_D=5^\circ/\text{sec}$ for the last 7.5 seconds. The results for the two commanded pitch rates are shown in Figures 4.137, 4.138 and 4.139, while the simulations for the three commanded pitch rates are shown in Figures 4.139, 4.140 and 4.141. Again it can be seen that the NN has a slightly better performance relative to the LQR controller.

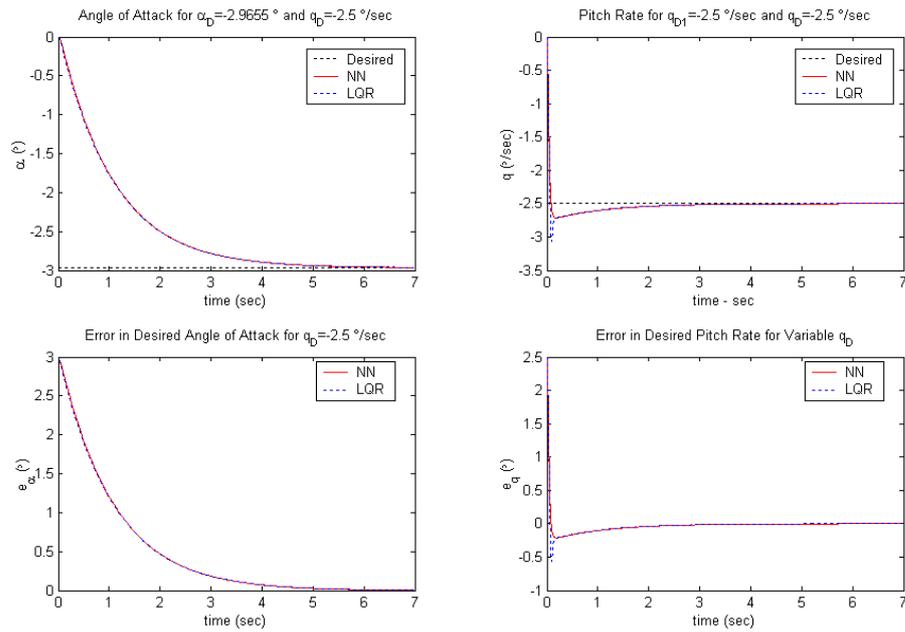


Figure 4.125. States and Errors Simulation ($q_D = -2.5^\circ/\text{sec}$).

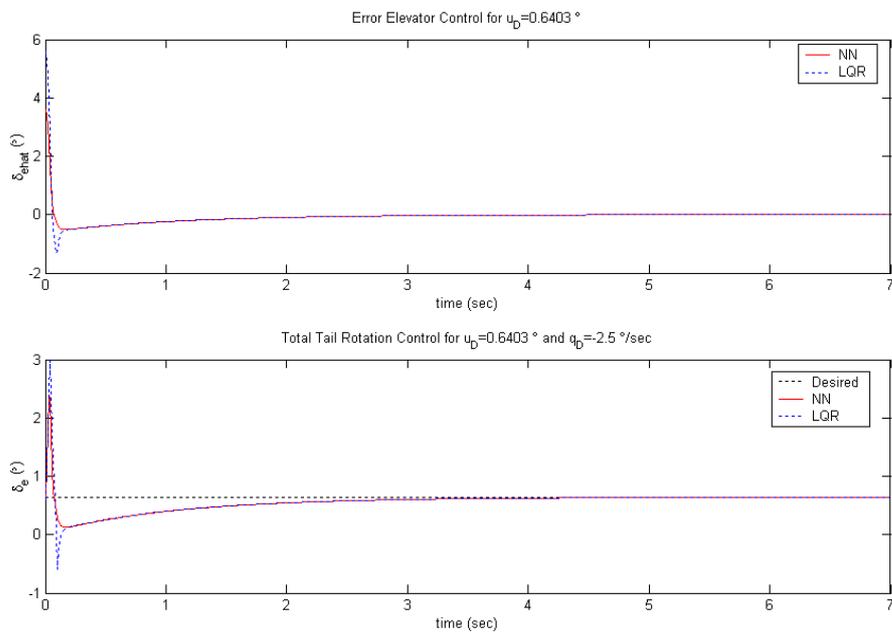


Figure 4.126. Total Tail Rotation Control and $\bar{\delta}_e$ Simulation ($q_D = -2.5^\circ/\text{sec}$).

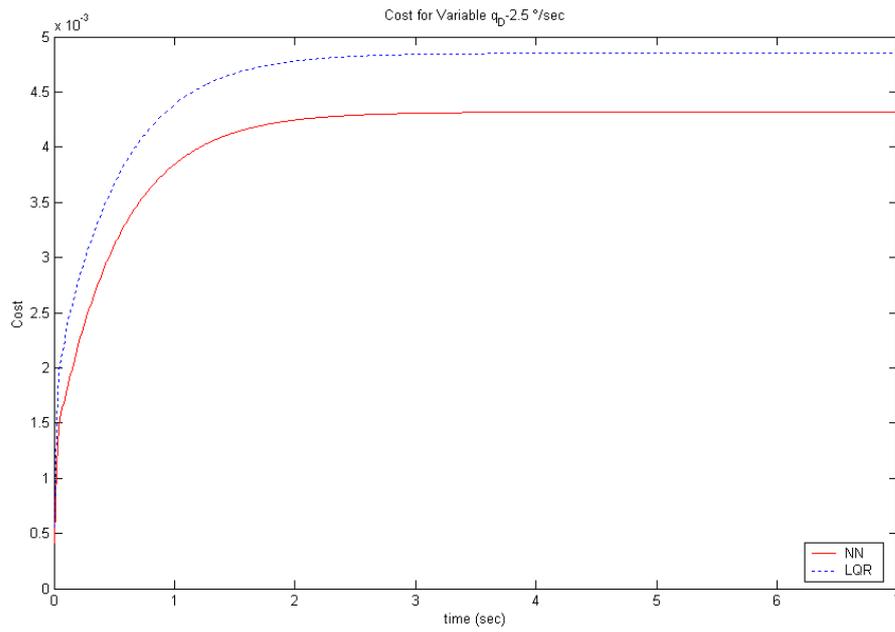


Figure 4.127. Cost Time History ($q_D = -2.5 \text{ }^\circ/\text{sec}$).

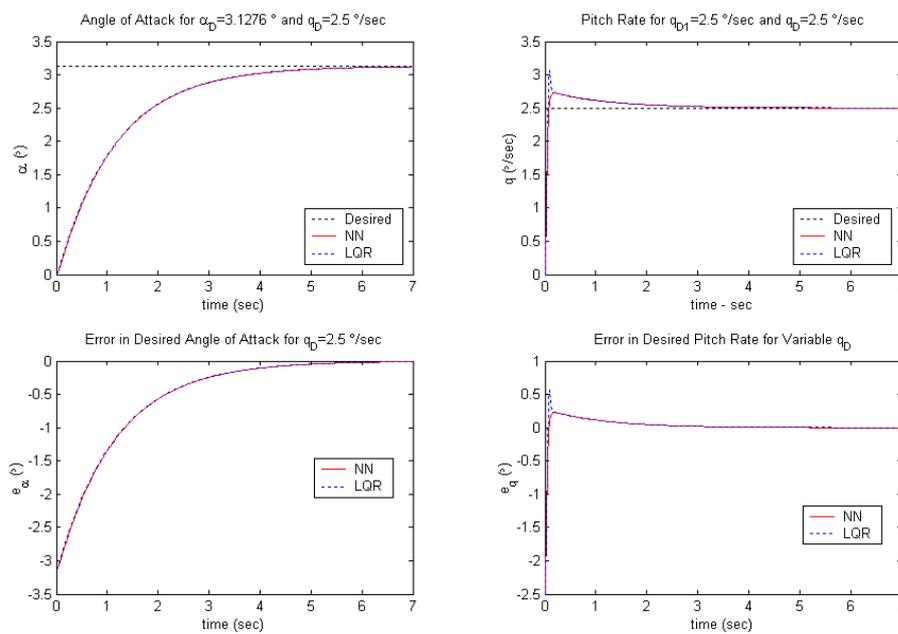


Figure 4.128. States and Errors Simulation ($q_D = 2.5 \text{ }^\circ/\text{sec}$).

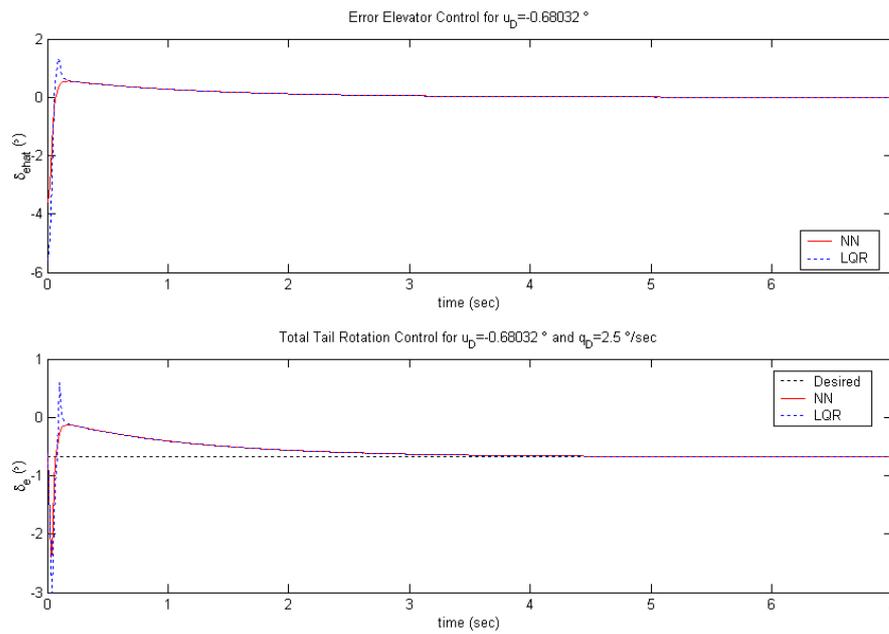


Figure 4.129. Total Tail Rotation Control and $\bar{\delta}_e$ Simulation ($q_D=2.5^\circ/\text{sec}$).

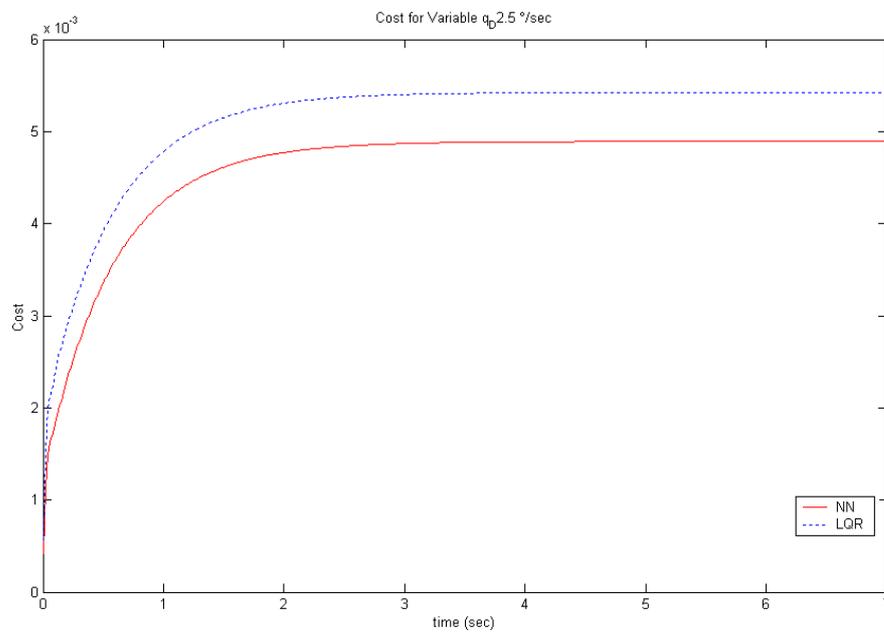


Figure 4.130. Cost Time History ($q_D=2.5^\circ/\text{sec}$).

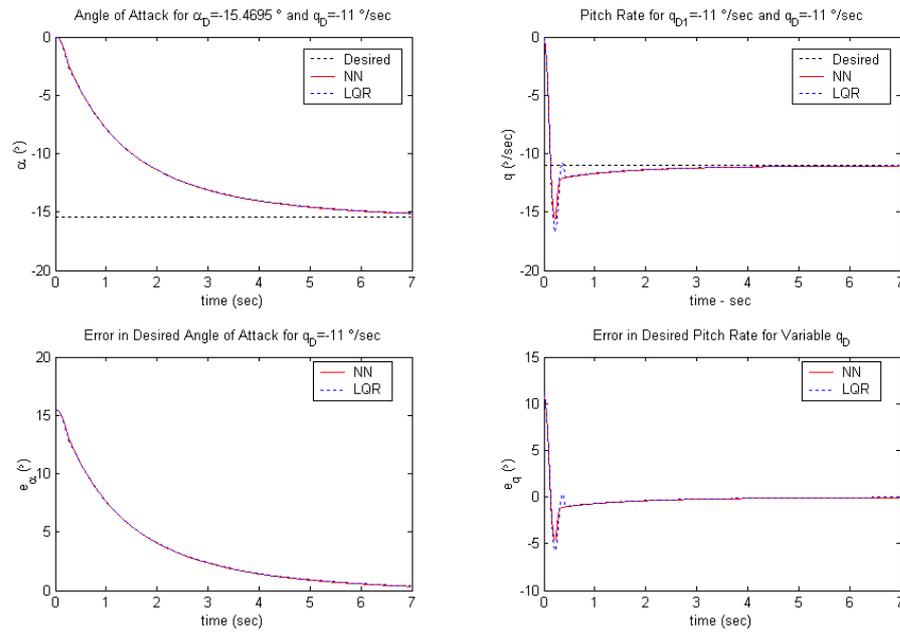


Figure 4.131. States and Errors Simulation ($q_D = -11$ °/sec).

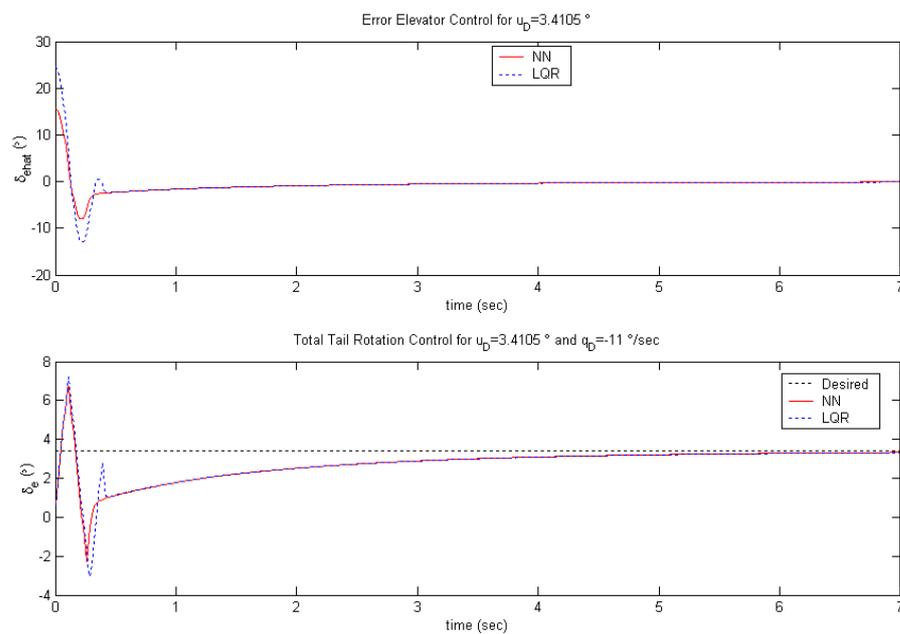


Figure 4.132. Total Tail Rotation Control and $\bar{\delta}_e$ Simulation for ($q_D = -11$ °/sec).

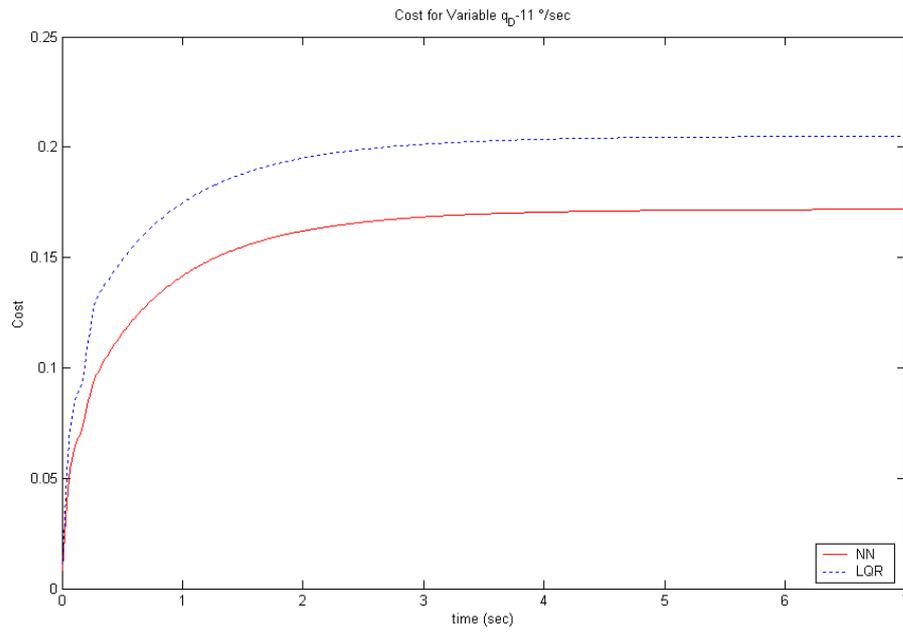


Figure 4.133. Cost Time History ($q_D = -11 \text{ %/sec}$).

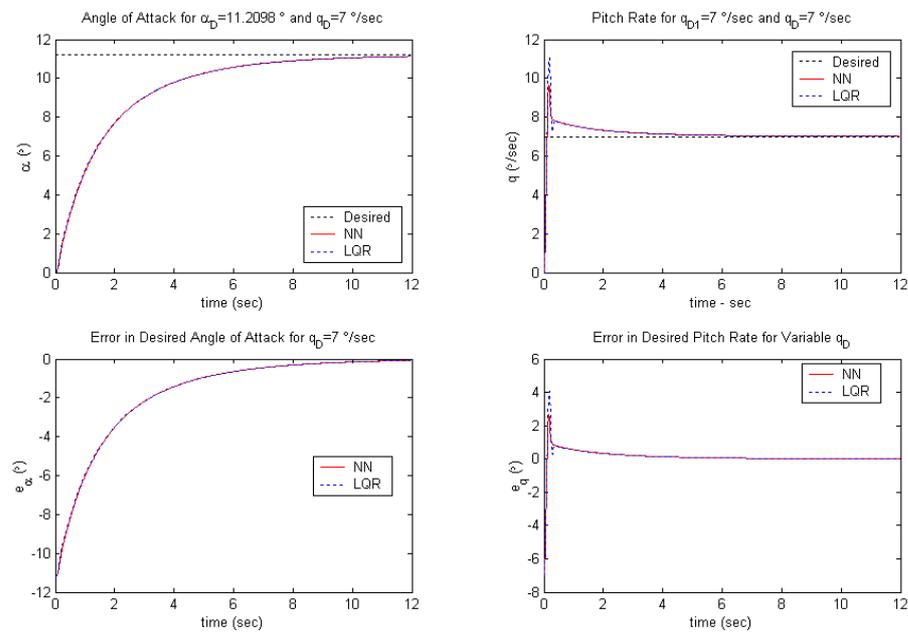


Figure 4.134. States and Errors Simulation ($q_D = 7 \text{ %/sec}$).

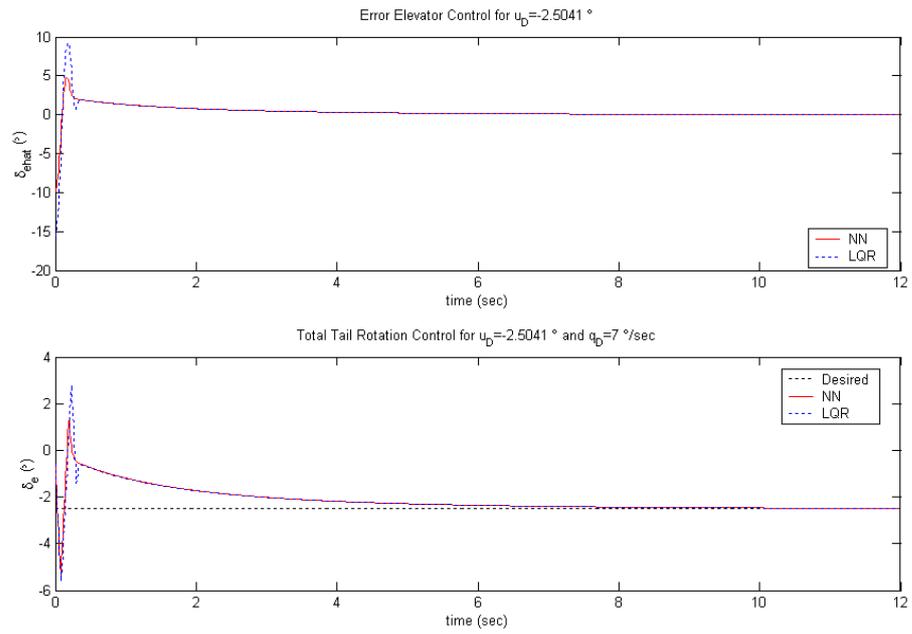


Figure 4.135. Total Tail Rotation Control and $\bar{\delta}_e$ Simulation ($q_D = 7^\circ/\text{sec}$).

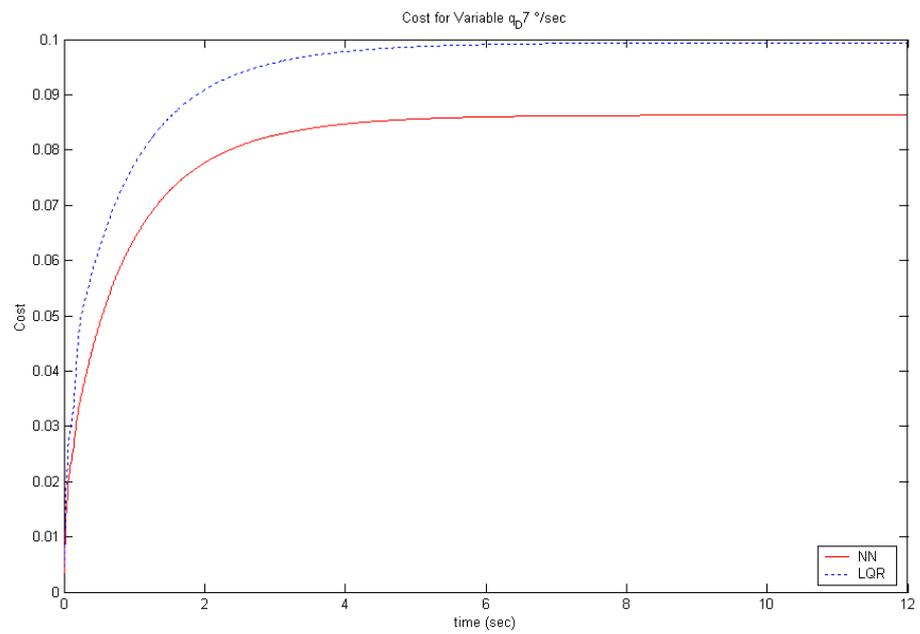


Figure 4.136. Cost Time History ($q_D = 7^\circ/\text{sec}$).

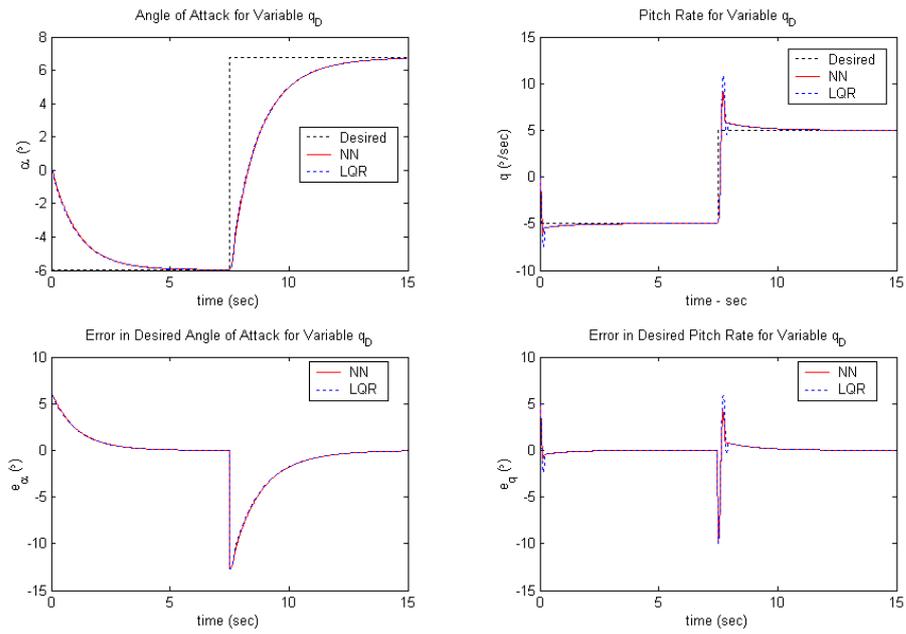


Figure 4.137. States and Errors Simulation ($q_D=-5^\circ/\text{sec}$, $q_D=5^\circ/\text{sec}$).

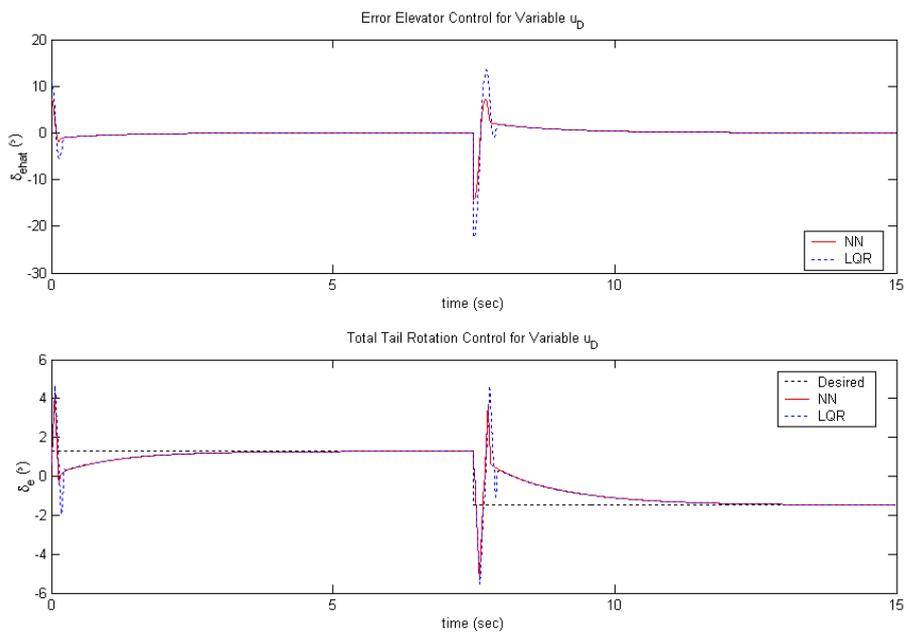


Figure 4.138. Total Tail Rotation Control and $\bar{\delta}_e$ Simulation ($q_D=-5^\circ/\text{sec}$, $q_D=5^\circ/\text{sec}$).

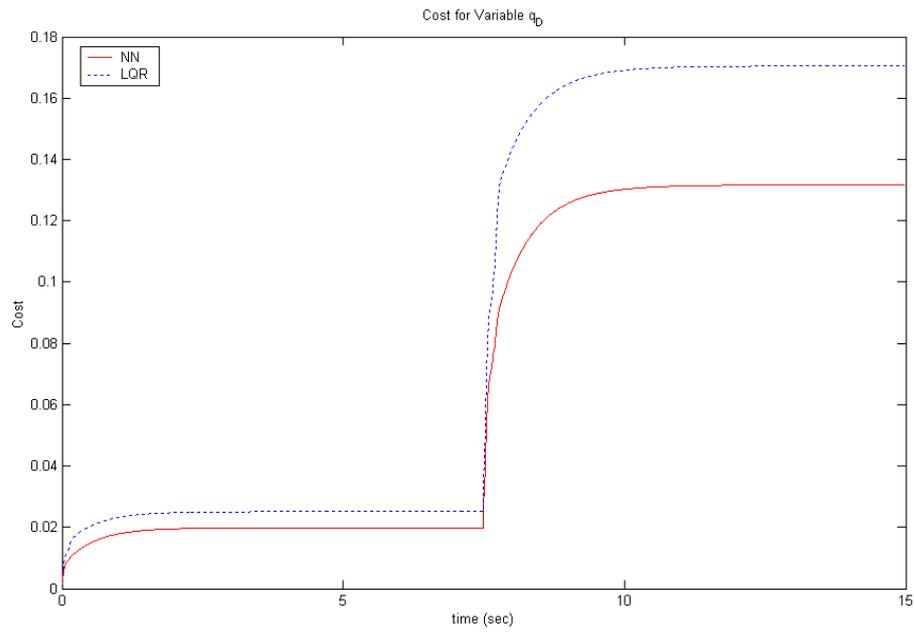


Figure 4.139. Cost Time History ($q_D=-5^\circ/\text{sec}$, $q_D=5^\circ/\text{sec}$).

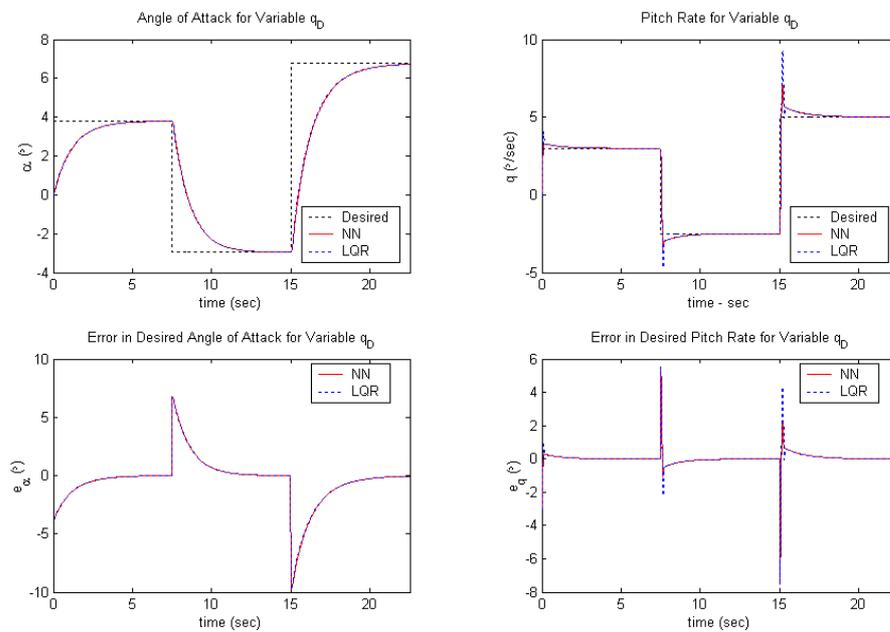


Figure 4.140. States and Errors Simulation ($q_D=3^\circ/\text{sec}$, $q_D=-2.5^\circ/\text{sec}$, $q_D=5^\circ/\text{sec}$).

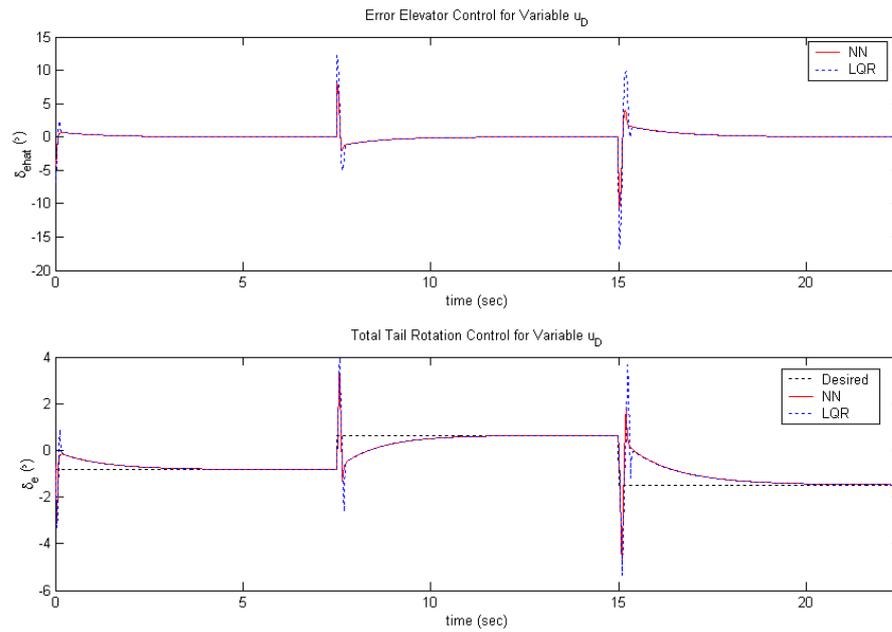


Figure 4.141. Total Tail Rotation Control and $\bar{\delta}_e$ Simulation ($q_D=3^\circ/\text{sec}$, $q_D=-2.5^\circ/\text{sec}$, $q_D=5^\circ/\text{sec}$).

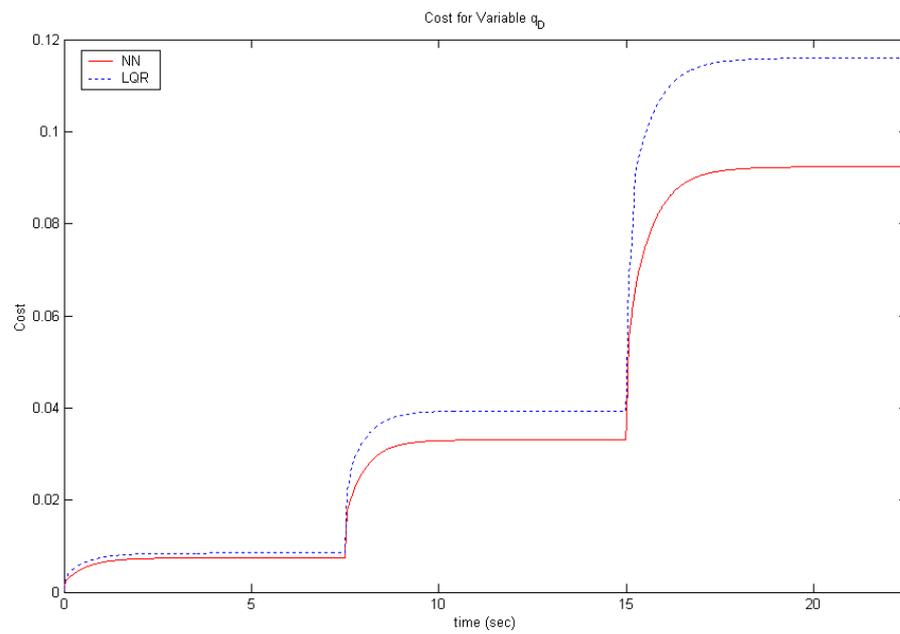


Figure 4.142. Cost Time History for Variable ($q_D=3^\circ/\text{sec}$, $q_D=-2.5^\circ/\text{sec}$, $q_D=5^\circ/\text{sec}$).

5. CONCLUSIONS

A Dual-NN solution for nonlinear optimal control problems has been presented. As seen in the analysis results described in section 4.2.6, the neurocontroller extends the stable region of operations for the aircraft model considered considerably. Further it has been shown in section 4.2.7 that the Dual-NN controller possesses inherent robustness characteristics that allow an aircraft to operate with high levels of the unmodeled uncertainties associated with the time lag and the tail effectiveness.

The equations that satisfy the optimality of the problem are solved with the help of NN. This makes it possible to synthesize the closed loop controllers for this complex process. It allows the philosophy of dynamic programming to be carried out without the need for near impossible computation and storage requirements. Another advantage of this Dual-NN approach includes the fact that no *a priori* assumptions about the form of the feedback control are needed; *i.e.*, one need not assume the control expressed in any particular form. The consequence of this off-line computational method is that the resulting control is available to be used as on-line state-feedback control for an entire envelope of initial conditions.

Despite the fact that the model used here neglects some dynamics, see section 4.2.1 for more detail, in order to focus the study in the nonlinearities in the high angle of attack, and considering that such simplifications are necessary in order to be able to have a reasonable model that would show the nonlinear capabilities of the neurcontroller here described, the author considers that the Dual-NN shows an incredible performance. When looking at the physical implications of some of the results, the reader needs to realize that the model here described, despite the high nonlinearities in the angle of attack, is a simplified longitudinal model of the full 6-DOF, which can not fully represent the dynamics of a real aircraft.

When looking at the unmodeled uncertainties introduced in the system, it is not fair to say that the results described in this thesis are unreasonable because a real system aircraft most likely will not be able to show the dynamics described in some of the plots. This would be equivalent to comparing apples with oranges since the evident differences

between the model used in this thesis and the 6-DOF model described in section 4.1.2. It is only fair to say that the Dual-NN architecture presented in this thesis performs really good for the model here provided for training.

Also a NN solution for nonlinear optimal tracking control problems has been presented in section 4.3. The neurocontroller approach is able to successfully track single and multiple commanded pitch rates within the range of available desired commands. The NN tracking controller performs identically to the LQR controller, providing no apparent improvement to the linearized controller. This is caused by the fact that during the tracking phase, the angles of attack encountered are not sufficiently high to make the nonlinear terms in Eqs. (166) and (167) play a role in the dynamics of the problem. The optimal tracking formulation described and used in this thesis is found to be limiting to the cases in which the desired states and control values associated with the commanded state are able to be found. This limits the type of problems that can be used since not always can the dynamics be solved for all the variables.

Future work on this NN architecture would include extending the NN capabilities to a more complex model, 6-DOF, with the ultimate goal of demonstrating the NN capabilities into a real aircraft system.

There has been a great deal of controversy among researchers, both in the field and not in the field of NN, about the validity and capabilities of NN to be able to imitate artificial intelligence (AI) in the future. Some claim that AI will never be reached since there is much more to AI in humans than billions of neurons connected between each other learning at every impulse generated between each other, and even if the technology allows in the future the co-existence of multilayered nets of artificial neurons, the essence of human thinking, which has yet to be discovered, will lack of the essence that makes nature so complex, so simple, and so perfect.

Again, the morality issue is not the purpose of this thesis, but the author felt necessary to raise a couple of these issues, to make the reader aware that the ultimate purpose of this work is not to just show how well or bad certain NN work relative to some conventional methods, but to show that in order to create the ultimate RFC system, we need to focus our attention to the ultimate RFC machine, human beings, who are able to acquire large amounts of data and make decisions weighting all possible scenarios.

The truth is that science today has only been able to disclose a tinny percentage of how the human brain works. Which regions of the brain are used for speech, which are used for thinking, which parts of the brain control emotions, etc... but the intent of this paper is not to demonstrate the feasibility or not of artificial neural nets to emulate AI, but to use the knowledge gain in AI up to date to solve problems that today's scientific technology and thinking cannot and will not solve unless a different approach is used. The NN approach here discussed only represents a small stone in the road to achieving RFC. As seen in section 2.2.3 many engineers have been able to successfully implement neurocontrollers that possess some reconfigurability capabilities, but the author of this thesis believes that researchers in the area of NN should take more into consideration Grossberg's interdisciplinary approach of solving complex problems by including the areas of mathematics, psychology and neurophysiology into their problem-solution tactics, instead of limiting the solution to try to solve problems mathematically. Some of today's more complex nonlinear controllers that have tried to emulate the inherent abilities of the human brain have forgotten that the ability of humans to solve complex problems does not completely rely on the mathematical approach of the problem.

But again these are only the thoughts of the author.

BIBLIOGRAPHY

1. The Boeing Company, "How safe is air travel?" *Boeing Comercial Airplanes – Jetliner Safety Information Site*. 15 February 2002. 20 July 2002
<<http://www.boeing.com/commercial/safety/flash.html>>.
2. NASA. "Aeronautics & Space Transportation Technology: Three Pillars for Success." *Aerospace Technology Enterprise*. March 1997. 20 Oct. 2002
<<http://www.aerospace.nasa.gov/library/3pillars2.pdf>>.
3. NASA. "Increase Safety: Making a Safe Aviation System even Safer." *Aerospace Technology Enterprise*. 6 August 2002. 20 September 2002
<<http://www.aerospace.nasa.gov/goals/safety.htm>>
4. Werbos, P.J., "Reconfigurable Flight Control Via Neurodynamic Programming and Universally Stable Adaptive Control," *Proceedings of the 2001 American Control Conference (AACC)*, Arlington, VA, June 2001.
5. Sofge, D. and White, D., "Applied learning: optimal control for manufacturing," in *Handbook of Intelligent Control Neural, Fuzzy, and Adaptive Approaches* (D. White and D. Sofge, eds.), New York, pp. 259-281, Van Nostrand Reinhold, 1992.
6. NeuroEngineering Laboratory at NASA Ames Research Center. "Landing Safely with a Broken Airplane." *NASA Aerospace Technology News*, 22 May 2001. 20 July 2002 <http://www.aero-space.nasa.gov/curevent/news/vol2_iss3/landing.htm>.
7. DeFelipe, Javier, "Cajal." *MIT Encyclopedia of the Cognitive Sciences*, Cambridge, MA: MIT Press, 1998.
8. DeFelipe, J. and E.G. Jones. *Cajal's degeneration and regeneration of the nervous system*. New York: Oxford University Press, 1991.
9. Bentivoglio, Marina. "Life and discoveries of Santiago Ramón y Cajal." *The Official Web Site of The Nobel Foundation*. 20 April 1998. 20 July 2002
<<http://www.nobel.se/medicine/articles/cajal/index.html>>.
10. Bentivoglio, Marina. "Life and discoveries of Camillo Golgi." *The Official Web Site of The Nobel Foundation*. 20 April 1998. 20 July 2002
<<http://www.nobel.se/medicine/articles/golgi/index.html>>.
11. The Nobel Foundation. "The black reaction - la reazione near." *The Official Web Site of The Nobel Foundation*. 16 June 2000. 20 July 2002
<<http://www.nobel.se/medicine/articles/golgi/black.html>>.

12. Walton, Kerry. "How did Cajal Hop a Ride Aboard Neurolab?" *NASA Neurolab Web: Santiago Ramon y Cajal*. 4 June 1998. 25 July 2002
<<http://neurolab.jsc.nasa.gov/onboard.htm>>.
13. Consejo Superior de Investigaciones Cientificas - Instituto Cajal. "The Cajal Legacy." *Correction of the Cajal Institute Paper*. 25 July 2002
<<http://www.cajal.csic.es/memoria/LegadoCajal98-99/Legadoi.htm>>.
14. Fischbach, Gerald D., "Mind and Brain a Scientific American Special Report," Scientific American Inc, 1994.
15. Athabasca University. "Synaptic Transmission." *Advanced Biological Psychology Tutorials*. 2 March 2002. 25 July 2002
<<http://psych.athabascau.ca/html/Psych402/Biotutorials/11/part1.html>>.
16. Roberts, E., Hennessy, J., Clabaugh, C., Myszewski, D., and Pang, J. "The Intellectual Excitement of Computer Science." *Neural Networks - Sophomore College Students Projects 2000*. 25 July 2002 <<http://cse.stanford.edu/classes/sophomore-college/projects-00/neural-networks/Biology/index.html>>.
17. McCulloch, W., and Pitts, W., "A Logical Calculus of the Ideas Immanent in Nervous Activity," *Bulletin of Mathematical Biophysics*, vol. 5, pp. 115-133, 1943. (Chapters 1,4,18)
18. Hebb, D.O., "The organization of Behavior," New York: Wiley, 1949. (Chapters 1,7, 13)
19. Hagan, M. T., Demuth, H. B. and Beale, M., "Neural Network Design," PWS Publishing Company, 1999. (Chapters 7, 10, 13, 15, 18)
20. Rosenblatt, F., "The perceptron: A Probabilistic Model for Information Storage and Organization in the Brain." *Psychological Review*, vol. 65, pp. 386-408, 1958. (Chapters 1,4)
21. Minsky, M. and Papert, S., "Perceptrons," Cambridge, MA: MIT Press, 1969. (Chapters 1,4)
22. Widrow, B. and Hoff, M.E., "Adaptive Switching Circuits," *1960 IRE WESCON Convention Record*, New York: IRE Part 4, pp. 96-104, 1960. (Chapters 1,10)
23. Von Neumann, J. and Morgenstern, O., "The Theory of Games and Economic Behavior," Princeton NJ: Princeton U. Press, 1953.
24. Kohonen, T., "Correlation, Matrix Memories," *IEEE Transactions on Computers*, vol. 21, pp. 353-359, 1972. (Chapters 1,13,18)

25. Anderson, J.A., "A Simple Neural Network Generating an Interactive Memory," *Mathematical biosciences*, vol. 14, pp. 197-220, 1972. (Chapter 1,13,18)
26. Grossberg, S., "Nonlinear Difference – Differential Equations in Prediction and Learning Theory," *Proceedings of the National Academy of Sciences*, vol. 60, pp. 758-765, 1968. (Chapter 13)
27. Hopfield, J.J., "Neural Networks and Physical Systems with Emergent Collective Computational Properties," *Proceedings of the National Academy of Sciences*. vol. 79, pp. 2554-2558, 1982. (Chapters 1, 18)
28. Tank, D. W., and Hopfield, J.J., "Simple Neural Optimization Networks: an A/D Converter, Signal Decision Circuit and a Linear Programming Circuit," *IEEE Transaction on Circuits and Systems*, vol. 33, no. 5, pp. 533-541, 1986. (Chapter 16)
29. Hopfield, J.J. and Tank, D. W., "Neural Computation of Decisions in Optimization Problems," *Biological Cybernetics*, vol. 52, pp 141-154, 1985. (Chapter 18)
30. Werbos, P.J., "Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences." *Ph.D Dissertation, Harvard University, Cambridge, MA, 1974.* also published as *The Roots of Backpropagation*, New York: John Wiley & Sons, 1994. (Chapter 1)
31. Rumelhart, D.E., Hinton, G.E., and Williams, R.J., "Learning Representations by Back-propagating Errors," *Nature*, vol. 323, pp. 533-536, 1986. (Chapter 11)
32. Parker, D.B., "Learning-logic: Casting the Cortex of the Human Brain Silicon," *Technical Report TR-47, Center for Computational Research Economics and Management Science, Cambridge, MA: MIT Press, 1985.* (Chapter 11)
33. Cun, Y. Le, "Une procedure d'apprentissage pour reseau a seuil assymetrique," *Cognitiva*, vol. 85, pp. 599-604, 1985. (Chapter 11)
34. Neosciences. "What are Neural Networks?" *Neural Networks – Introducing Neural Networks*. 13 March 2001. 25 July 2002
<http://www.neosciences.com/Technologies/nn_intro.htm>.
35. Werbos, P.W., "Neurocontrol and Supervised Learning: An Overview and Evaluation," *Handbook of Intelligent Control*, Van Nostrand Reinhold, 1992.
36. Balakrishnan, S. N. and Biega, V. "Adaptive Critic Based Neural Networks for Aircraft Optimal Control." *Journal of Guidance, Control and Dynamics*, vol. 19, no. 4, pp. 893-898, July-August 1996.

37. Balakrishnan, S.N. and Saini, G., "Adaptive Critic Based Neurocontroller for Autolanding of Aircraft with Varying Glideslopes," *Proceedings of IEEE International Conference on Neural Networks*, vol.4, pp. 2288-2291, 1997.
38. Anderson, C.W., "Learning to Control an Inverted Pendulum using Neural Networks," *IEEE Control Systems Magazine*, vol. 9, pp. 33-37, April 1998.
39. DARPA, "Neural Network Study," Lexington, MA: MIT Lincoln Laboratory. 1988. (Chapter 1)
40. Penrose, R., "Shadows of the Mind," *A Search for the Missing Science of Consciousness*, Oxford University Press, May 1996.
41. Zadeh, L. A., "Fuzzy Sets," *Information and Control*, 8(3): 338-53, June 1965.
42. Azerbaijan International, "What is Fuzzy Logic?" Winter 1994. 20 September 2002 <http://www.azer.com/aiweb/categories/magazine/24_folder/24_articles/24_fuzzywhat.html>.
43. Boverie, S., Jamshidi, M., Titli, A., and Zadeh, L. A., "Applications of Fuzzy Logic: Towards High Machine Intelligence Quotient Systems," Prentice Hall, January 1997.
44. Rechenberg, I., "Evolutionsstrategie," *Frommann-Holzboog*, Stuttgart, Germany, 1973.
45. Holland, J., "Adaptation in Natural and Artificial Systems," Cambridge, MA: MIT Press, 1975.
46. Carnegie Mellon University Artificial Intelligence Repository, July 8, 1999. 20 September 2002 <<http://www-2.cs.cmu.edu/afs/cs.cmu.edu/project/ai-repository/ai/html/air.html>>.
47. Bryson, A. E. and Ho, Y., "Applied Optimal Control," Hemisphere Publishing Co., pp.128-211. 1975.
48. Bellman, R.E., "Linear Equations and Quadratic Criteria," vol. 1, 1967.
49. Arnold III, W.F. and Laub, A.J., "Generalized Eigenproblem Algorithms and Software for Algebraic Riccati Equations," *Proc. IEEE*, 72, 1746-1754, 1984.
50. Nise, N. S., "Control Systems Engineering," Addison-Wesley Publishing Company, Second Edition, 1995.
51. Khalil, H. K., "Nonlinear Systems," Prentice Hall, New Jersey, 2002.

52. Hunt, K. J., "Neural Networks for Control Systems – A Survey," *Automatica*, vol. 28, no.6, pp.1083-1112, 1992.
53. Balakrishnan, S. N., and Biega, V., "Adaptive Critic Based Neural Networks for Aircraft Optimal Control," *Journal of Guidance, Control, and Dynamics*, vol. 19, no. 4, pp. 893-898, 1996.
54. Balakrishnan, S.N. and Saini, G., "Adaptive Critic based Neurocontroller for Autolanding of Aircraft with Varying Glideslopes," *Proceedings of IEEE International Conference on Neural Networks*. vol.4, pp. 2288-2291, 1997.
55. Balakrishnan, S. N. and Han, D. C., "Adaptive Critic Based Neural Networks for Agile Missile Control," *1998 AIAA Guidance, Navigation and Control Conference*, Boston, MA, vol. 3, pp. 1803-1812, 1998.
56. Plumer, E. S., "Optimal Control of Terminal Processes Using Neural Networks," *IEEE Transactions on Neural Networks*, vol. 7, no. 2, pp. 408-418, 1996.
57. Lewis, F. L. and Jagannathan, S., "Multilayer Discrete-Time Neural-Net Controller with Guaranteed Performance," *IEEE Transactions On Neural Networks*, vol. 7, no. 1, January 1996.
58. Kim, B. S., and Calise, A. J., "Nonlinear Flight Control Using Neural Networks," *Journal of Guidance, Control, and Dynamics*, vol. 20, no. 1, pp. 26-33, 1997.
59. Calise, A.J., Corban, J.E., Pei Y. and Prasad, J.V.R., "Adaptive Nonlinear Controller Synthesis and Flight Test Evaluation on an Unmanned Helicopter," *Proceedings to the IEEE International Conference on Control Applications*, 1999.
60. Calise , A. J., Johnson, E. N., Rysdyk, H. and Rysdyk, R., "Feedback linearization with Neural Network Augmentation Applied to X-33 Altitude Control," *Proceeding of the AIAA Guidance Navigation and Control Conference*, Denver, August 2000.
61. Calise, A. J., and Johnson, E. N., "Reusable Launch Vehicle Adaptive Guidance and Control Using Neural Networks," *Proceedings of the AIAA Guidance Navigation and Control Conference*, Montreal, August 2001.
62. Calise, A.J., Lee, S. and Sharma, M., "Direct Adaptive Reconfigurable Control of a Tailless Fighter Aircraft," *Proceedings of the AIAA Guidance Navigation and Control Conference*, Boston, August 1998.
63. Calise, A.J., Lee, S. and Sharma, M., "Development of a Reconfigurable Flight Control Law for the X-36 Tailless Fighter Aircraft." *Proceedings of the AIAA Guidance Navigation and Control Conference*, Denver, August 2000.

64. Haley, P. and Soloway, D., "Aircraft Reconfiguration Using Neural Generalized Predictive Control," *Proceedings of the American Control Conference, (AACC)*, Arlington, VA, pp. 2294-2929, June 2001.
65. Bull, J., Kaneshige, J. and Totah, J., "Generic Neural Flight Control and Autopilot System," *Proceedings of the AIAA Guidance, Navigation and Control Conference*, Denver, AIAA-2000-4281, August 2000.
66. Calise, A. J. and Rysdyk, R. T., "Fault Tolerant Flight Control via Adaptive Neural Network Augmentation," *Proceedings of the AIAA Guidance, Navigation and Control Conference*, Boston, AIAA-1998-4483, August 1998.
67. Blake, M., "The NASA Advanced Concepts Flight Simulator: A Unique Transport Aircraft Research Environment," Reston, VA, AIAA 96-3518, 1996.
68. Idan, M., Johnson, M., Calise, A. J. and Kaneshige, J., "Intelligent Aerodynamic/ Propulsion Flight Control For Safety: A Nonlinear Adaptive Approach," *Proceedings of the American Control Conference (AACC)*, Arlington, VA, June 2001.
69. Lee, T., Kim, Y., "Nonlinear Adaptive Flight Control Using Backstepping and Neural Networks Controller," *Journal of Guidance, Control, and Dynamics*, vol. 24, no. 4, pp. 675-682, July-August 2001.
70. Ferrari, S. and Stengel, R. F., "Classical/Neural Synthesis of Nonlinear Control Systems," *Proceedings of the AIAA Guidance, Navigation and Control Conference*, Denver, CO, AIAA-2000-4552, August 14-17 2000.
71. Hartana, P. and Sasiadek, J. Z., "Adaptive Fuzzy Logic System for Sensor Fusion in Dead-Reckoning Mobile Robot Navigation," *Proceedings of the 15th Triennial World Congress of the IFAC*, Barcelona, Spain July 21-26 2002.
72. A. Green, J. and Sasiadek, J. Z., "Inverse Dynamics and Fuzzy Repetitive Learning Flexible Robot Control," *Proceedings of the 15th Triennial World Congress of the IFAC*, Barcelona, Spain July 21-26 2002.
73. Cuesta, F. and Ollero, A. "Stability Analysis of Fuzzy Reactive Navigation," *Proceedings of the 15th Triennial World Congress of the IFAC*, Barcelona, Spain July 21-26 2002.
74. Martinez-de Dios, J.R. and Ollero, A., "An Infrared Vision System for Field Robotics Applications," *Proceedings of the 15th Triennial World Congress of the IFAC*, Barcelona, Spain July 21-26 2002.
75. Ollero, A. "Anibal Ollero's Personal Pages." 2002. 20 October 2002
<<http://www.esi2.us.es/~aollero/>>.

76. Armingol, J.M., de la Escalera, A., Mata, M. and Salichs, M.A., "Learning Visual Landmarks for Mobile Robot Navigation," *Proceedings of the 15th Triennial World Congress of the IFAC*, Barcelona, Spain, July 21-26 2002.
77. MATLAB, The Mathworks Inc., 2000.
78. Roskam, J., "Airplane Flight Dynamics and Automatic Flight Controls – Part I," DARCorporation, Kansas, 1995.
79. NASA, Dryden Flight Research Center F-8 Graphics Collection, 3 May 2002. 8 November 2002 <http://www.dfrc.nasa.gov/gallery/graphics/F-8/Small/F-8_3view.gif>.
80. Lewis, F. L. and Stevens, B. L., "Aircraft Control and Simulation," John Wiley & Sons, Inc., New York, 1992.
81. Anderson Jr., J. D., "Introduction to Flight," McGraw-Hill Inc., New York, 1989.
82. Anderson Jr., J. D., "Fundamentals of Aerodynamics," McGraw-Hill Inc., New York, 1991.
83. Bertin, J. J., Smith, M. L., "Aerodynamics for Engineers," Prentice Hall, New Jersey, 1998.
84. Nelson, R. C., "Flight Stability and Automatic control," McGraw-Hill, Boston, 1998.
85. Garrard, W. L. and Jordan, M. "Design of Nonlinear Automatic Flight Control Systems," *Automatica*, vol. 13, pp 497-505, 1977.
86. Etkin, B., "Dynamics of Atmospheric Flight," John Wiley, New York, 1972.
87. Lee, E.B. and Markus, L., "Foundations of Optimal Control Theory," John Wiley, New York, 1967
88. Al'Brekht, E. G., "On the Optimal Stabilization of Nonlinear Systems," *PM M-J, Appl. Math. Mech.* 25, pp. 1254-1266, 1961.
89. Clark, L.G. , Garrard, W. L. and McClamroch, N. H., "An Approach to Sub-optimal Feedback Control of Nonlinear System," *Int. J. Control* 7, 290-296, 1967.
90. Lukes, D.L., "Optimal Regulation of nonlinear Dynamical Systems," *SIAM J. control* 7, pp. 75-100, 1969.
91. Garrard, W.L., "Suboptimal Feedback Control for Nonlinear Systems," *Automatica* 8, pp. 219-221, 1972.

92. Clark, L.G. and Garrard, W.L., "On the Synthesis of Suboptimal, Inertia-wheel Altitude Control Systems," *Automatica* 5, pp. 781-789, 1969.

VITA

Sergio Esteban Roncero was born in Barcelona, Spain, on May 31st 1974. He received his Bachelor of Science in Aerospace Engineering from the University of Missouri –Rolla, Rolla, Missouri, United States of America in May 1999, graduating with *Magna Cum Laude* and with Honors. Following his graduation, he pursued graduate studies at the University of Missouri – Rolla in August 1999. He received his Masters of Science Degree in Aerospace Engineering in December 2002.

During his time at the University of Missouri – Rolla, he played varsity tennis, being named Most Valuable Player (MVP) for the tennis team both 1996-1997 and 1997-1998 academic years. He was awarded with the Sportsmanship Award for the year 1997-1998 by coaches and players of the Mid-America Intercollegiate Athletics Association (MIAA) Tennis Division and earned medal in the MIAA final conference after winning 3rd place in the 1st number player's draw in April 1998. He was named to the MIAA's All Academic team for the 1997-1998 school year.

He was president of Sigma Gamma Tau National Honor Society of Aerospace Engineering from January 1999 through May 1999. He was awarded by the School of Engineering Honors Committee with the best Honors Project for the academic year 98-99, funded by the NASA Missouri Space Grant Consortium. He was awarded with the Aerospace Engineering Distinguished Student in recognition and appreciation of outstanding performance and contribution to the department by the faculty in May 1999. In April 2000 he received the second place on the graduate division of the 2000 American Institute of Aeronautics and Astronautics (AIAA) Region V student paper conference for the paper and presentation of "Static and Dynamic Analysis of an Unconventional Plane: Flying Wing." He was vice-President of the UMR Advanced Aero-Vehicle Group (AAVG) from August 1999 through May 2001.

Sergio Esteban Roncero was inducted into Sigma Gamma Tau National Honor Society of Aerospace Engineering in January of 1998. He has been a member of AIAA since August 1998, member of the Order of the Engineer since May 1999, member of the USA National Society of Professional Engineers (NSPE) since August 1999.