

Air Force Institute of Technology

AFIT Scholar

Theses and Dissertations

Student Graduate Works

12-1993

Exploring QDES as a Tool for Determining Limits of Achievable Performance in Aircraft Design

Mark F. Reidinger

Follow this and additional works at: <https://scholar.afit.edu/etd>



Part of the [Aerospace Engineering Commons](#), and the [Controls and Control Theory Commons](#)

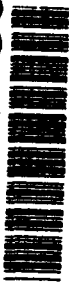
Recommended Citation

Reidinger, Mark F., "Exploring QDES as a Tool for Determining Limits of Achievable Performance in Aircraft Design" (1993). *Theses and Dissertations*. 6631.

<https://scholar.afit.edu/etd/6631>

This Thesis is brought to you for free and open access by the Student Graduate Works at AFIT Scholar. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of AFIT Scholar. For more information, please contact AFIT.ENWL.Repository@us.af.mil.

AD-A275 359



EXPLORING QDES AS A TOOL FOR DETERMINING
LIMITS OF ACHIEVABLE PERFORMANCE
IN AIRCRAFT DESIGN

THESIS

Mark F. Reidinger

Major, USAF

AFIT/GAE/ENY/93D-23

Approved for public release; distribution unlimited

DTIC
ELECTE
FEB 03 1994
S E D

15186 93-30432



**EXPLORING QDES AS A TOOL FOR DETERMINING LIMITS
OF ACHIEVABLE PERFORMANCE
IN AIRCRAFT DESIGN**

THESIS

Presented to the Faculty of the Graduate School of Engineering

of the Air Force Institute of Technology

Air University

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

Mark F. Reidinger, B.S.

Major, USAF

December, 1993

DTIC QUALITY INSPECTED 5

Accession For	
NTIS	CRA&I <input checked="" type="checkbox"/>
DTIC	TAB <input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution /	
Availability Codes	
Dist	Avail and / or Special
A-1	

Approved for public release; distribution unlimited

Acknowledgments

I would like to thank all the members of my section, GAE-93D, for making AFIT a somewhat enjoyable experience and for making my job as their section leader so easy.

I wish to thank all of the AFIT staff, especially Dr. Ridgely for his help with this thesis and for his excellent instruction in innumerable control classes.

I would also like to thank Mr. Russ Milleron of AFIT Computer Resources who helped me set up QDES and led me through the intricacies of the Sun operating system. Also many thanks to Captain Kevin Boyum to answering my countless questions on Microsoft Word, with which this thesis was written.

Last, but not least, I want to thank my wife Susan for 18 months of seeing me only at mealtimes and occasionally on the weekends. Her enthusiasm and encouragement helped an old guy like me go back to school after 13 years and survive.

Mark F. Reidinger

Table of Contents

	Page
Acknowledgments	ii
Table of Contents	iii
List of Figures	ix
List of Tables.....	xii
List of Symbols.....	xiii
Abstract.....	xvi
I. Introduction	1-1
1.1 Background	1-1
1.2 Objective.....	1-2
1.3 Assumptions	1-2
1.4 Scope.....	1-3
1.5 Approach	1-4
1.6 Overview	1-5
II. Development of Theory.....	2-1
2.1 Background	2-1
2.2 Standard Feedback Diagram.....	2-1
2.3 Transfer Matrix Characteristics	2-3
2.4 Coprime Factorization.....	2-4
2.5 Parameterization of All Stabilizing Compensators.....	2-5
2.6 Closed Loop Transfer Matrices	2-6
2.7 QDES Theory of Operation.....	2-7
2.8 Summary.....	2-8
III. Measures of Performance	3-1
3.1 Introduction	3-1

3.2	Aircraft Performance.....	3-1
3.3	Time Response Characteristics	3-2
3.4	Stability Margins	3-4
3.5	Summary.....	3-7
IV.	Classical Design	4-1
4.1	Introduction	4-1
4.2	Axis System and Sign Convention	4-1
4.3	Aircraft Model	4-2
4.4	Inner Loop Pitch SAS Theory	4-3
4.5	Inner Loop Pitch SAS for the A-4D	4-5
4.6	Summary.....	4-10
V.	QDES Design for The A-4D.....	5-1
5.1	Model Development.....	5-1
5.3	Plant Discretization	5-4
5.4	Generating the Impulse Response Matrices.....	5-4
5.5	Writing the QDES Source File	5-8
5.5.1	The Preamble.....	5-8
5.5.2	The Declarations Section	5-9
5.5.3	The Objective Section.....	5-9
5.5.4	The Constraints Section.....	5-10
5.6	Running QDES	5-11
5.7	Forming the QDES Controller and Closed Loop System	5-11
5.8	A-4D QDES Design Results.....	5-12
5.8.1	A-4D QDES Design, 1 g Response.....	5-12
5.8.2	A-4D QDES Design, 3 g Response.....	5-15
5.8.3	A-4D QDES Design, 4.5 g Response.....	5-18

5.8.4	A-4D QDES Design, 4.6 g Response	5-20
5.9	A-4D Short Period Approximation	5-22
5.9.1	A-4D Short Period QDES Design, 1 g Response	5-23
5.9.2	A-4D Short Period QDES Design, 3 g Response	5-25
5.9.3	A-4D Short Period QDES Design, 4.6 g Response	5-27
5.9.4	A-4D Short Period QDES Design, 4.7 g Response	5-29
5.10	Summary	5-31
VI.	QDES Design for the F-16	6-1
6.1	Introduction	6-1
6.2	Differences From the A-4D Problem	6-1
6.3	F-16 QDES Design Results	6-2
6.3.1	F-16 QDES Design, 1 g Response	6-2
6.3.2	F-16 QDES Design, 9.8 g Response	6-5
6.3.3	F-16 QDES Design, 10.0 and 10.2 g Responses	6-7
6.3.4	F-16 QDES Design, 10.4 g Response	6-10
6.4	F-16 Short Period Approximation	6-11
6.4.1	F-16 Short Period QDES Design, 1 g Response	6-12
6.4.2	F-16 Short Period QDES Design, 10.4 g Response	6-14
6.4.3	F-16 Short Period QDES Design, 10.6 g Response	6-16
6.5	Summary	6-17
VII.	Step-By-Step Procedure	7-1
7.1	Introduction	7-1
7.2	QDES Installation and Test	7-1
7.3	Model Source	7-2
7.4	Using the "nzprp.m" File	7-2
7.4.1	Entering Plant Data	7-2

7.4.2	Discretizing the Plant.....	7-4
7.4.3	Forming the P-K Style Matrices.....	7-4
7.4.4	Forming an Observer Based Controller	7-4
7.4.5	Calling "obc2ttt.m"	7-5
7.5	Writing the QDES Source File	7-6
7.5.1	The Preamble.....	7-6
7.5.2	The Declarations Section	7-6
7.5.3	The Constraints Section.....	7-7
7.6	Running QDES	7-10
7.7	Outputting the Results.....	7-11
7.8	Summary.....	7-12
VIII.	Conclusions and Recommendations.....	8-1
8.1	Introduction	8-1
8.2	Usefulness of QDES	8-1
8.2.1	Time Response	8-1
8.2.2	Frequency Response	8-2
8.3	Other Approaches	8-3
8.3.1	Longer Run Times	8-3
8.3.2	Different Nominal Plant	8-4
8.3.3	Output Margins	8-4
8.4	Conclusion.....	8-5
Appendix A.	A-4D State Space Models and Design Information.....	A-1
A.1	A-4D Longitudinal Model at Flight Condition 5	A-1
A.1.1	Original Plant Model.....	A-1
A.1.2	A-4D Plant with Actuator Dynamics.....	A-2
A.1.3	A-4D Model in Desired Units	A-3

A.2	A-4D QDES Design Information.....	A-4
A.2.1	A-4D 1 g Command	A-4
A.2.2	A-4D 3 g Command	A-5
A.2.3	A-4D 4.5 g Command	A-5
A.2.4	A-4D 4.6 g Command	A-6
A.3	A-4D Short Period Approximation Flight Condition 5.....	A-6
A.4	A-4D Short Period Approximation QDES Design Information	A-6
A.4.1	A-4D Short Period 1 g Command.....	A-7
A.4.2	A-4D Short Period 3 g Command.....	A-7
A.4.3	A-4D Short Period 4.6 g Command.....	A-7
A.4.4	A-4D Short Period 4.7 g Command.....	A-8
Appendix B.	F-16 State Space Models and Design Information	B-1
B.1	F-16 Full Longitudinal Model.....	B-1
B.2	F-16 QDES Design Information	B-2
B.2.1	F-16 1 g Command.....	B-2
B.2.2	F-16 9.8 g Command.....	B-3
B.2.3	F-16 10.0 g Command.....	B-3
B.2.4	F-16 10.2 g Command.....	B-4
B.2.5	F-16 10.4 g Command.....	B-4
B.3	F-16 Short Period Approximation	B-5
B.4	F-16 Short Period Approximation QDES Design Information.....	B-5
B.4.1	F-16 Short Period 1 g Command	B-6
B.4.2	F-16 Short Period 10.4 g Command.....	B-6
B.4.3	F-16 Short Period 10.6 g Command.....	B-6
Appendix C.	MATLAB Script Files	C-1
C.1	The "nzprp.m" File.....	C-1

C.2	The QDES Source File.....	C-5
C.3	The "nzpst.n" File.....	C-7
C.4	The "visnz.m" File.....	C-12
C.5	The "surf.m" File.....	C-19
	Bibliography.....	BIB-1
	Vita.....	VITA-1

List of Figures

Figure	Page
2.1. Standard Feedback Diagram	2-2
3.1. Unit Step Response	3-4
3.2. Polar Plot with M Circle Radius.....	3-6
4.1. Longitudinal Axis System	4-2
4.2. Elevator Sign Convention	4-2
4.3. A-4D N_z to N_{zcmd} Static Gain Feedback	4-4
4.4. Inner Loop Pitch SAS.....	4-4
4.5. A-4D q to δ_{ecmd} Root Locus.....	4-5
4.6. A-4D Outer Loop N_z to δ_{ecmd} Root Locus.....	4-6
4.7. A-4D Inner Loop Pitch SAS Unit Step Response.....	4-7
4.8. A-4D Elevator Deflection For 1 g Step Response	4-8
4.9. A-4D Forward Velocity and AOA Change For 1 g Step Response	4-8
4.10. Polar Plot of A-4D With Inner Loop Pitch SAS	4-9
5.1. Block Diagram for A-4D QDES Design.....	5-3
5.2. A-4D QDES Design, 1 g Response.....	5-13
5.3. A-4D Elevator and AOA 1 g Responses	5-14
5.4. A-4D 1 g Input Sensitivity and CLBW.....	5-15
5.5. A-4D QDES Design, 3 g Response.....	5-16
5.6. A-4D Elevator and AOA 3 g Responses	5-17
5.7. A-4D 3 g Input Sensitivity and CLBW.....	5-17
5.8. A-4D QDES Design, 4.5 g Response.....	5-18
5.9. A-4D Elevator and AOA 4.5 g Responses.....	5-19
5.10. A-4D 4.5 g Input Sensitivity and CLBW.....	5-19
5.11. A-4D QDES Design 4.6 g Response.....	5-20

5.12. A-4D Elevator and AOA 4.6 g Responses.....	5-21
5.13. A-4D 4.6 g Input Sensitivity and CLBW	5-21
5.14. A-4D Short Period QDES Design, 1 g Response	5-24
5.15. A-4D Short Period Elevator and AOA 1 g Responses	5-24
5.16. A-4D Short Period 1 g Input Sensitivity and CLBW	5-25
5.17. A-4D Short Period QDES Design, 3 g Response	5-26
5.18. A-4D Short Period Elevator and AOA 3 g Responses	5-26
5.19. A-4D Short Period 3 g Input Sensitivity and CLBW	5-27
5.20. A-4D Short Period QDES Design, 4.6 g Response.....	5-28
5.21. A-4D Short Period Elevator and AOA 4.6 g Responses	5-28
5.22. A-4D Short Period 4.6 g Input Sensitivity and CLBW	5-29
5.23. A-4D Short Period QDES Design, 4.7 g Response.....	5-30
5.24. A-4D Short Period Elevator and AOA 4.7 g Responses	5-30
5.25. A-4D Short Period 4.7 g Input Sensitivity and CLBW	5-31
6.1. F-16 QDES Design, 1 g Response	6-4
6.2. F-16 Elevator and AOA 1 g Responses	6-4
6.3. F-16 1 g Input Sensitivity and CLBW	6-5
6.4. F-16 QDES Design, 9.8 g Response	6-6
6.5. F-16 Elevator and AOA 9.8 g Responses	6-7
6.6. F-16 QDES Design, 10.0 g Response	6-8
6.7. F-16 QDES Design, 10.2 g Response	6-8
6.8. F-16 Elevator and AOA 10.0 g Responses	6-9
6.9. F-16 Elevator and AOA 10.2 g Responses	6-9
6.10. F-16 QDES Design, 10.4 g Response.....	6-10
6.11. F-16 Elevator and AOA 10.4 g Responses	6-11
6.12. F-16 Short Period QDES Design, 1 g Response.....	6-13

6.13. F-16 Short Period Elevator and AOA 1 g Responses.....	6-13
6.14. F-16 Short Period 1 g Input Sensitivity and CLBW	6-14
6.15. F-16 Short Period QDES Design, 10.4 g Response	6-15
6.16. F-16 Short Period Elevator and AOA 10.4 g Responses.....	6-15
6.17. F-16 Short Period QDES Design, 10.6 g Response	6-16
6.18. F-16 Short Period Elevator and AOA 10.6 g Responses.....	6-17

List of Tables

Table	Pages
4.1. A-4D Flight Condition 5 Parameters	4-3

List of Symbols

Symbol	Page
g	Acceleration of Gravity..... 1-1
q	Aircraft Pitch Rate..... 1-3
N_z	Normal Acceleration of Aircraft c.g. 1-3
AOA	Angle of Attack..... 1-3
c.g.	Center of Gravity..... 1-4
SAS	Stability Augmentation System 1-4
N_{zcmd}	Normal Acceleration Command..... 1-4
P	System Plant..... 2-1
$K(s)$	Controller or Compensator 2-1
w	Exogenous Inputs..... 2-1
z	Regulated Outputs..... 2-1
θ	Aircraft Pitch Angle..... 2-1
y	Sensed Outputs 2-2
u	Actuator Input..... 2-2
δ_{ecmd}	Elevator Deflection Command..... 2-2
RHP	Right Half s-Plane 2-3
LHP	Left Half s-Plane..... 2-3
H_{zw}	Closed Loop System Transfer Matrix 2-5
MIMO	Multi-Input Multi-Output 2-8
FIR	Finite Impulse Response 2-8
M_p	Overshoot 3-2
U_s	Undershoot 3-2
T_r	Rise Time..... 3-2
T_{sl}	Settling Time..... 3-2

LQR	Linear Quadratic Regulator	2-5
LQE	Linear Quadratic Estimator.....	2-5
e_{cmd}	Command Error	3-3
e_{ss}	Steady State Error	3-3
$L(s)$	Open Loop Transfer Function.....	3-5
S	Sensitivity.....	3-5
M_{rad}	M Circle Radius.....	3-5
SISO	Single-Input Single-Output.....	3-5
$\ \cdot \ _{\infty}$	Infinity Norm.....	3-5
GM_u	Upper Gain Margin.....	3-6
GM_l	Lower Gain Margin	3-6
PM	Phase Margin.....	3-6
x_s	Stability Axis System x-Axis.. ..	4-1
z_s	Stability Axis System z-Axis	4-1
y_s	Stability Axis System y-Axis	4-1
γ_0	Equilibrium Flight Path Angle.....	4-1
x_v	Vehicle Carried Axis System x-Axis	4-1
α_{trim}	Trim Angle of Attack	4-1
α	Angle of Attack.....	4-2
γ	Flight Path Angle.....	4-2
δ_e	Elevator Deflection.....	4-2
h	Altitude	4-3
M	Mach.....	4-3
V_0	Forward Velocity at Equilibrium Flight Condition.....	4-3
ζ_p	Phugoid Damping Ratio.....	4-3
ζ_{sp}	Short Period Damping Ratio.....	4-3

PI	Proportional Plus Integral	4-3
K_q	Inner Loop Gain	4-5
K_{inz}	PI Controller Zero Location	4-5
K_{nz}	PI Controller Gain	4-6
v	Forward Velocity	4-7
sec	Second(s)	4-7
deg	Degree(s)	4-8
ft	Feet	4-8
rad	Radians	4-9
dB	Decibels	4-9
U_{in}	Loop Input For Stability Margins	5-1
U_{out}	Loop Output For Stability Margins	5-2
Hz	Hertz	5-4
T_s	Sample Time	5-5
zoh	Zero Order Hold	5-5
K	State Feedback Gain Matrix	5-5
L	Kalman Filter Gain Matrix	5-5
CLBW	Closed Loop Bandwidth	5-14

Abstract

This thesis explores the usefulness of a computer aided control design software package called QDES in determining the limits of achievable aircraft performance as it relates to controller capability. Modern aircraft, particularly fighters, are being designed to be statically unstable to enhance their maneuverability and performance. It is possible that the aircraft, although physically capable of a certain level of performance due to its engine/airframe combination, may be uncontrollable up to this level. This study sought to develop a methodology to use QDES to make a preliminary analysis of an aircraft design to determine if there exists a controller that will enable this design to meet its performance goals, and if not, to determine what the limits of achievable performance are.

EXPLORING QDES AS A TOOL FOR DETERMINING LIMITS OF ACHIEVABLE PERFORMANCE IN AIRCRAFT DESIGN

I. Introduction

1.1 Background

Modern aircraft, particularly military fighters, are capable of some very impressive levels of performance. The engine and airframe technology has produced aircraft that seem to be limited only by the ability of humans to fly them. Another limitation can arise in the area of controller design. In other words, a given aircraft may have the physical capability to pull nine g's or roll at 720 degrees per second, but the question to be asked is, can a controller be found that will enable it to perform at this level while maintaining desired stability margins and response characteristics such as rise time, settling time, overshoot, etc.?

A computer program called QDES, written by Dr. Stephen Boyd of Stanford University, may help to answer this question. This program uses the theory of parameterization of all stabilizing controllers in the parameter Q along with the fact that many closed loop system performance requirements may be cast as convex constraints on the closed loop transfer function H_{zw} . QDES uses a convex problem solver to search for a control law in the parameter Q which satisfies the convex constraints. If one cannot be

found, QDES returns a message to this effect. This thesis explores the possibility of using QDES to answer the crucial question posed in the previous paragraph.

1.2 Objective

The objectives of this thesis are two-fold. The first objective is to explore the feasibility of using QDES to determine the limits of achievable performance of an aircraft as it relates to controller design. The second objective is to derive a simplified process whereby QDES, along with several associated MATLAB script files, may be used by someone not intimately familiar with aircraft control to determine the limits of achievable performance of a proposed aircraft design. All that would be required are the state space representations of the linearized aircraft equations of motion for a given flight condition.

1.3 Assumptions

The following assumptions were made concerning this thesis:

- The aircraft responses to control surface deflections are accurately modeled by a given set of linearized equations of motion for a specified flight condition. Per Etkin [9], this representation makes the following assumptions:
 - The aircraft is a rigid body whose mass is unchanging with time.
 - The earth is considered flat.
 - Altitude is constant and thus atmospheric density and the acceleration of gravity are constant.
 - The earth's rotation is neglected.
 - Longitudinal equations are in the wind axis and the lateral are in the body axis.

These equation sets are not coupled.

- The aircraft has a plane of symmetry and the momenta of spinning rotors is neglected.
- The trimmed angle of attack is known for the specified flight condition.
- The aircraft surface deflection and surface deflection rate limits as well as the acceleration and angle of attack limits are known.
- All transfer functions and transfer function matrices in this thesis are real, rational, and linear time invariant. This is a requirement for the use of QDES.
- Aircraft pitch rate (q) and normal acceleration (N_z) can be measured perfectly and are available for feedback. Thus, sensor noise will not be considered in the system model.
- The aircraft is considered to be flying in still air and not subject to atmospheric disturbances. Thus, process noise will not be considered in the system model.
- All feedback is considered positive. This is consistent with most of the post modern control theory literature.

1.4 Scope

The scope of this thesis will be limited to the following tasks:

- Set up QDES on the AFIT computers and ensure its proper operation.
- Use QDES to explore the limits of performance for two different fighter aircraft. This will be limited to determining maximum normal acceleration (N_z) for the longitudinal mode. These limits will be subject to constraints on the time response of the closed loop system as well as constraints on control usage, system stability margins, and angle of attack (AOA) limitations.
- Develop a step-by-step procedure for use of QDES to implement the process described in the previous paragraph given the assumptions in Section (1.3). This

will include the writing of any necessary MATLAB script files to assist the non-controls oriented engineer in easily carrying out this procedure.

- Evaluate the value and limitations of using the procedure as developed.

1.5 Approach

The following approach will be taken to accomplish the tasks outlined in section (1.4):

- Take the state space representation of the linearized equations of motion for the A-4D aircraft at Mach 0.6, 15,000 ft and design an inner loop pitch stability augmentation system (inner loop pitch SAS) controller using root locus design techniques for normal acceleration at the c.g. (N_z) to normal acceleration command (N_{zcmd}).
- Use QDES to design a controller for the same problem, N_z to N_{zcmd} and compare it with the classical design. This will build familiarity with the QDES design process and provide a cross check of its results.
- Use QDES to explore the limits of performance (maximum N_z) subject to constraints as listed in Section (1.4).
- Take the state space representation of the linearized equations of motion for the longitudinal mode of the F-16 (a more challenging aircraft to control) at Mach 0.6, sea level, and use QDES to explore the limits of performance as described for the A-4D.
- Take the results of the above steps and write the step by step procedure described in Section (1.4).

1.6 Overview

This thesis is organized into eight chapters and three appendices. Chapter II gives a brief explanation of the parameterization of all stabilizing compensators and how QDES uses that theory to design closed loop systems. Chapter III defines and explains the measures of performance to be used in this thesis. Chapter IV applies the Inner Loop Pitch SAS classical control method to the A-4D N_z to N_{zcmd} problem to provide a baseline for evaluations of QDES designs. Chapter V walks through the QDES design process by using it to explore the N_z limits of performance for the A-4D. Chapter VI shows the results of the same process applied to the F-16. Chapter VII provides a step-by-step process for the non-controls engineer to apply QDES in evaluating whether a proposed design can meet its desired performance level from a control standpoint. It also presents the lessons learned in applying QDES to the limits of performance problem. Chapter VIII provides an evaluation of QDES' usefulness for this task, and some caveats and cautions should one attempt to do so.

II. Development of Theory

2.1 Background

QDES uses the theory of the parameterization of all stabilizing compensators. In his thesis, DeLaney [6] presents a very rigorous review of this theory and its relation to QDES. Rather than present a duplication of that effort, a somewhat condensed summary of the information required to understand the processes presented in this thesis will be provided.

2.2 Standard Feedback Diagram

The starting point for any discussion of control theory is the system representation or block diagram. Figure 2.1 is the standard feedback or "P-K" diagram. The following definitions from Boyd [5] describe the components and input/output signals of the closed loop system.

P: Plant. The system to be controlled. In this thesis, the aircraft.

K(s): Controller or Compensator. The algorithm used by the control processor to generate the actuator commands based on the sensor and command signals received.

w: Exogenous Inputs. All input signals to the model except the actuator input. For an aircraft one of these would be normal acceleration command (N_{zcmd}).

z: Regulated Outputs. Every output signal from the model. These do not have to be measurable. An example would be pitch angle (θ).

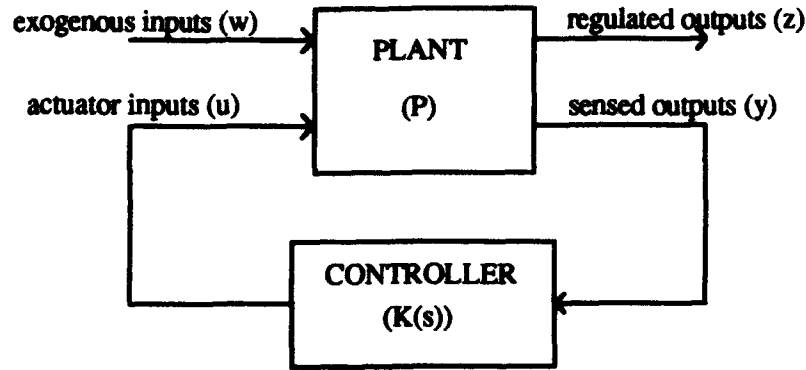


Figure 2.1. Standard Feedback Diagram

y: Sensed Outputs. Output signals that are accessible to the controller. These must be measurable. For the aircraft longitudinal axis, q is commonly used.

u: Actuator Inputs. The inputs to the model that can be manipulated by the controller. For most aircraft in the longitudinal mode, this is elevator deflection commanded (δ_{ecmd}).

The state space representation of the plant transfer matrix P is

$$P = \left[\begin{array}{c|cc} A & B_w & B_u \\ \hline C_z & D_{zw} & D_{zu} \\ C_y & D_{yw} & D_{yu} \end{array} \right] \quad (2.1)$$

The individual matrices within P correspond to those used in the state-variable differential equations

$$\dot{x} = Ax + B_w w + B_u u \quad (2.2)$$

$$z = C_z x + D_{zw} w + D_{zu} u \quad (2.3)$$

$$y = C_y x + D_{yw} w + D_{yu} u \quad (2.4)$$

which describe the state-space representation of the plant P . A thorough understanding of this plant description is necessary to set up the control problem in QDES.

Alternatively, the plant matrix can be partitioned as

$$P = \left[\begin{array}{c|c} P_{zw} & P_{zu} \\ \hline P_{yw} & P_{yu} \end{array} \right] \quad (2.5)$$

corresponding to

$$z = P_{zw}w + P_{zu}u \quad (2.6)$$

$$y = P_{yw}w + P_{yu}u \quad (2.7)$$

where P_{zw} is the transfer matrix from w to z , etc.

Finally, the P matrix can be partitioned as

$$P = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array} \right] \quad (2.8)$$

corresponding to the state-variable differential equations

$$\dot{x} = Ax + Bu \quad (2.9)$$

$$y = Cx + Du \quad (2.10)$$

These various representations of the plant transfer matrix will be referenced throughout this thesis.

2.3 Transfer Matrix Characteristics

A plant transfer matrix as described in equation (2.8) is said to be *controllable* if (A,B) is *stabilizable* and (C,A) is *detectable*. To say (A,B) is stabilizable means that all right half s -plane (RHP) eigenvalues of A (unstable modes) can be moved to the left half s -plane (LHP) using state feedback. These modes are said to be *controllable*. Characterizing (C,A) as detectable means that the eigenvalues of A which are *unobservable* (can't be seen and therefore can't be moved) are in the open LHP (stable). To control a system it *must* be stabilizable and detectable.

A transfer function or transfer function matrix is an element of RH_{∞} ($\in RH_{\infty}$) if it is proper, real, rational, and has no poles in the RHP or on the imaginary axis. A real, rational transfer function can be written as a ratio of finite polynomials in s with real coefficients. A transfer function is proper if the degree of the denominator is greater than or equal to the degree of the numerator [7].

2.4 Coprime Factorization

The concept of coprime factorization is central to the theory of the parameterization of all stabilizing compensators. The following derivation and formulas are from Francis [10] as presented by Ridgely [15].

Two matrices $D, N \in RH_\infty$ are right coprime if they have the same number of columns and there exist matrices (not unique) $\tilde{X}, \tilde{Y} \in RH_\infty$ such that

$$\tilde{X}D - \tilde{Y}N = I \quad (\text{Bezout Identity}) \quad (2.11)$$

Two matrices $\tilde{D}, \tilde{N} \in RH_\infty$ are left coprime if they have the same number of rows and there exist matrices (not unique) $X, Y \in RH_\infty$ such that

$$\tilde{D}X - \tilde{N}Y = I \quad (2.12)$$

X, Y, \tilde{X} , and \tilde{Y} are known as *Bezout Factors*.

Ridgely [15] shows that any finite dimensional, linear time-invariant transfer matrix $G(s)$ has a doubly coprime factorization

$$G(s) = ND^{-1} = \tilde{D}^{-1}\tilde{N} \quad (2.13)$$

where

$$\begin{bmatrix} \tilde{X} & -\tilde{Y} \\ -\tilde{N} & \tilde{D} \end{bmatrix} \begin{bmatrix} D & Y \\ N & X \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} \quad (2.14)$$

which is known as a *Generalized Bezout Identity*. Note that the (1,1) and (2,2) blocks of equation (2.14) are equations (2.11) and (2.12) respectively. This representation ensures that $G(s)$ has no RHP pole zero cancellations guaranteeing (A,B) is stabilizable and (C,A) is detectable as described in Section (2.3).

To find the eight matrices on the left side of equation (2.14), start again with equation (2.8) with (A,B) stabilizable and (C,A) detectable. Choose matrices F and H such that

$$A_F \equiv A + BF \quad (2.15)$$

$$A_H \equiv A + HC \quad (2.16)$$

are stable. F can be a full state feedback gain matrix found using Linear Quadratic Regulator (LQR) theory and H an estimator gain matrix found by Linear Quadratic Estimator (LQE) theory. Also define

$$C_F \equiv C + DF \quad (2.17)$$

$$B_H \equiv B + HD \quad (2.18)$$

With equations (2.15) - (2.18), the eight matrices from equation (2.14) are given by

$$N = \left[\begin{array}{c|c} A_F & B \\ \hline C_F & D \end{array} \right] \quad \tilde{N} = \left[\begin{array}{c|c} A_H & B_H \\ \hline C & D \end{array} \right] \quad (2.19)$$

$$D = \left[\begin{array}{c|c} A_F & B \\ \hline F & I \end{array} \right] \quad \tilde{D} = \left[\begin{array}{c|c} A_H & H \\ \hline C & I \end{array} \right] \quad (2.20)$$

$$X = \left[\begin{array}{c|c} A_F & -H \\ \hline C_F & I \end{array} \right] \quad \tilde{X} = \left[\begin{array}{c|c} A_H & -B_H \\ \hline F & I \end{array} \right] \quad (2.21)$$

$$Y = \left[\begin{array}{c|c} A_F & -H \\ \hline F & 0 \end{array} \right] \quad \tilde{Y} = \left[\begin{array}{c|c} A_H & -H \\ \hline F & 0 \end{array} \right] \quad (2.22)$$

2.5 Parameterization of All Stabilizing Compensators

Consider again the plant transfer matrix partitioned as in equation (2.5). With this representation for P used in the block diagram in figure (2.1), the closed loop transfer matrix is

$$H_{zw} = P_{zw} + P_{zu}K[I - P_{yu}K]^{-1}P_{yw} \quad (2.23)$$

Francis [10] proves that K stabilizes P if and only if it stabilizes P_{yu} . Furthermore, through block diagram manipulation, it can be shown that only P_{yu} "sees" the feedback matrix K . Thus, P_{yu} determines the stabilizability of the closed loop system while the other transfer matrices (P_{zw} , P_{zu} , and P_{yw}) determine its performance.

Let

$$P_{yu}(s) = ND^{-1} = \tilde{D}^{-1}\tilde{N} \quad (2.24)$$

The eight RH_∞ matrices of this doubly-coprime factorization of P_{yu} can be found by the methods described in Section (2.4). The family of all real rational controllers ($K(s)$) that stabilize P_{yu} and therefore P is parameterized by

$$K(s) = [Y(s) - D(s)Q(s)][X(s) - N(s)Q(s)]^{-1} \quad (\text{right coprime factorization}) \quad (2.25)$$

$$= [\tilde{X}(s) - Q(s)\tilde{N}(s)]^{-1} [\tilde{Y}(s) - Q(s)\tilde{D}(s)] \quad (\text{left coprime factorization}) \quad (2.26)$$

where $Q(s) \in RH_\infty$. This is known as the *Youla Parameterization* [15]. If $Q(s) = 0$, $K(s)$ is called the *Central* or *Nominal* controller. If matrices F and H were found using LQR and LQE theory as in Section (2.4), the central or nominal controller is known as an *Observer Based* controller [10], [15]. This concept is key to setting up the control problem for QDES.

2.6 Closed Loop Transfer Matrices

Starting with the doubly-coprime factorization of P_{yu}

$$P_{yu} = ND^{-1} = \tilde{D}^{-1}\tilde{N} \quad (2.27)$$

$$\begin{bmatrix} \tilde{X} & -\tilde{Y} \\ -\tilde{N} & \tilde{D} \end{bmatrix} \begin{bmatrix} D & Y \\ N & X \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} \quad (2.28)$$

and using the formulas for $K(s)$ shown in equations (2.25) and (2.26), Francis [10] proves that the closed loop transfer matrix H_{zw} can be written as

$$H_{zw} = T_1 - T_2QT_3 \quad (2.29)$$

where

$$T_1 = P_{zw} + P_{zu}DYP_{yw} \quad (2.30)$$

$$T_2 = P_{zu}D \quad (2.31)$$

$$T_3 = \tilde{D}P_{yw} \quad (2.32)$$

Francis also shows that the state space representation of T_1 , T_2 , and T_3 can be written as

$$T_1 = \left[\begin{array}{cc|c} A_F & -B_u F & B_w \\ 0 & A_H & B_w + H D_{yw} \\ \hline C_z + D_{zu} & -D_{zu} F & D_{zw} \end{array} \right] \quad (2.33)$$

$$T_2 = \left[\begin{array}{c|c} A_F & B_u \\ \hline C_z + D_{zu} F & D_{zu} \end{array} \right] \quad (2.34)$$

$$T_3 = \left[\begin{array}{c|c} A_H & B_w + H D_{yw} \\ \hline C_y & D_{yw} \end{array} \right] \quad (2.35)$$

where the various matrices are as defined in equations (2.1) and (2.15) - (2.18). Notice that T_1 , T_2 , and T_3 can be formed having only the plant information as represented in equation (2.1) and its coprime factorization as described in section (2.4).

2.7 QDES Theory of Operation

QDES seeks to design the parameter $Q(s)$ such that the closed loop system H_{zw} from equation (2.29) meets a set of closed loop constraints or requirements which are *convex* in Q . Boyd [3] defines this concept as follows: "if H_{zw} is a closed loop response which meets our performance requirements, and \tilde{H}_{zw} is any other closed loop response which meets our requirements, then the closed loop response $(H_{zw} + \tilde{H}_{zw})/2$ meets our requirements." If the above condition holds, the requirements are convex.

QDES is a C compiler which translates control constraints such as step response characteristics or transfer function values into constraints on Q and then uses a numerical convex problem solver to find the values of Q enabling H_{zw} to meet these constraints [3]. If QDES cannot find the required values of Q , this indicates that no controller of that order can be designed to meet them. If QDES does find the required values of Q , the controller can be found via equations (2.25) and (2.26). Because QDES uses numerical techniques, it requires the continuous plant model to be discretized. Furthermore, QDES

uses the impulse responses of the matrices T_1 , T_2 , and T_3 . Use of these impulse response matrices allows the problem to be finite dimensional and the resulting controller (usually of very high order) to be implemented with feedback around a multi-input, multi-output (MIMO) finite impulse response (FIR) filter as described in [3]. A thorough discussion of the actual mechanics of using QDES follows in Chapter VII.

2.8 *Summary*

This chapter discussed the control theory behind the operation of QDES. Starting with the standard feedback diagram, it defined the input and output vector signals and the various representations of the transfer matrices that relate them to each other. With these matrices, coprime factorization was introduced and was used to demonstrate the concept of the parameterization of all stabilizing compensators. Given this parameterization, the closed loop system H_{zw} was represented in terms of the stable closed loop maps T_1 , T_2 , T_3 and the parameter Q . The concept of convex constraints on H_{zw} was then introduced and the method whereby QDES uses a numerical convex problem solver to arrive at the required values of Q was briefly discussed.

III. Measures of Performance

3.1 Introduction

As the purpose of this thesis is to explore the usefulness of QDES in determining the limits of achievable aircraft performance, it is necessary to specify and define the measures of performance to be used. For this thesis, these measures will be categorized under aircraft performance, time response characteristics, and stability margins.

3.2 Aircraft Performance

Aircraft performance can be characterized by various measures such as speed, range, payload capacity, ordnance delivery accuracy, maneuverability, etc. From a controllability standpoint, maneuverability is the measure most directly related to the aircraft control system. In the longitudinal axis, to which this thesis is limited, maximum normal acceleration (N_z) and pitch rate (q) are the two most closely tied to the aircraft's control system. Maximum attainable angle of attack (AOA) is another longitudinal measure of performance, but it is more a function of aerodynamic design rather than the control system capability, unless the *control* of the aircraft *at* high AOA is the area to be examined. This challenging aircraft control problem is beyond the scope of this thesis. The change in forward velocity is also an important measure of performance, but it is more a function of aerodynamic design, wing loading, and thrust availability than control system capability.

This thesis uses N_z as the aircraft measure of performance. This is the most logical in that all aircraft are structurally g-limited. Furthermore, normal g capability relates

directly to two maneuverability measures important to air-to-air combat and air-to-ground weapons delivery. The first is turn radius which is a function of true airspeed and N_z . The lower the true airspeed and the higher the normal g , the smaller the turn radius and the *tighter* the turn. The smallest turn radius occurs at the well known *corner velocity*, the lowest airspeed where the aircraft can pull the maximum g allowed by structural limits without exceeding AOA limitations. Directly related to the turn radius is the nose tracking rate, how many degrees per second the nose of the aircraft moves across the horizon. Again, the maximum nose tracking rate occurs when the aircraft is pulling maximum g at the lowest possible speed.

While maximum N_z relates to the magnitude of the aircraft's performance, one also needs to examine the quality of that level of performance.

3.3 Time Response Characteristics

One method of evaluating the quality of a closed loop system's response, and thus the control law used, is measurement of its unit step response characteristics. QDES allows easy constraint or optimization of these characteristics. Furthermore, simulation of the closed loop system response to a unit step input provides quick and easy evaluation of them.

The following characteristics and their definitions will be used in this thesis. In all cases, the command value is normally 1.0, the unit step input.

Overshoot (M_p): The maximum amount by which the step response exceeds the command value, usually expressed as a percentage of the command value.

Undershoot (U_s): The maximum negative value (assuming a positive step command) attained by the step response, usually expressed as a percentage of the command value.

Rise Time (T_r): The earliest time after which the step response exceeds 0.8 of the command value [5].

Settling Time (T_{sl}): The earliest time after which the step response is always a certain percentage of the command value [5]. Note that this is slightly different from the normally accepted definition which uses the *final, or steady state value* rather than the command value. The reason for this is that in using QDES, one seeks to constrain T_{sl} before knowing what the final value will be. Therefore, it is assumed to be the command value and the constraints are specified accordingly. This will become clear when the actual mechanics of using QDES are discussed in Chapter V.

Command Error (e_{cmd}): The difference between the command value and the value of the step response at the end of the time period of interest. It is expressed as a percentage of the command value. This measure is chosen over the more common steady state error (e_{ss}) because in this thesis, the aircraft response will be examined over a short time period. The steady state error is defined from the final value of the response once all the transients have died out. In the case of the Phugoid oscillation, this could be hundreds of seconds after the commanded input. Thus, this measure of performance would have little meaning for the responses to be examined.

Figure 3.1 illustrates a typical unit step response.

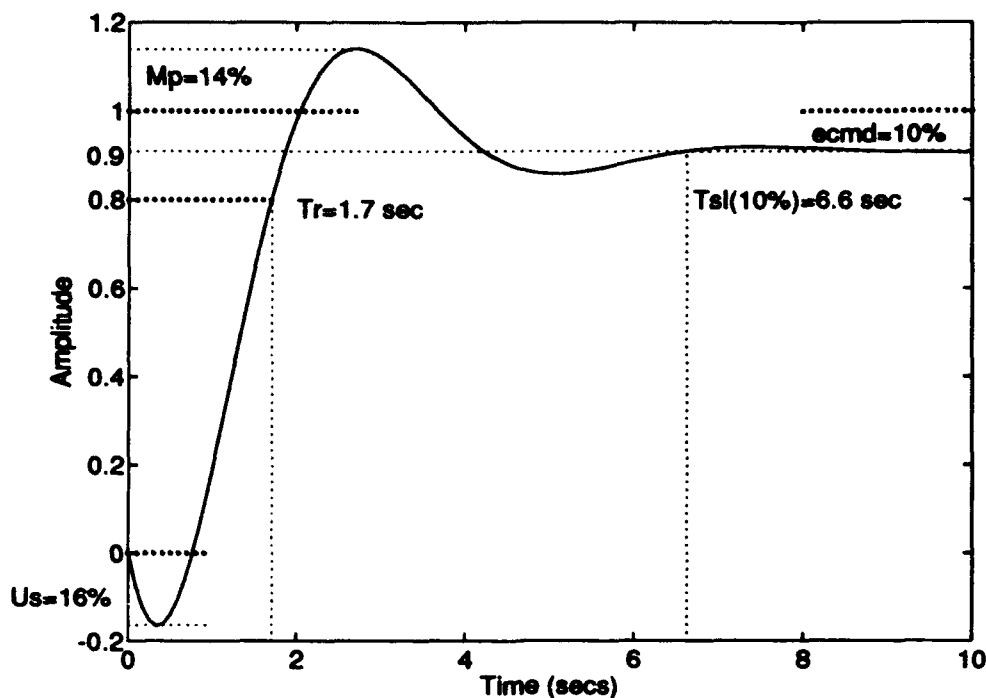


Figure 3.1. Unit Step Response

It is not difficult to appreciate how these unit step response characteristics relate directly to the quality of the aircraft's response. For instance, when a pilot desires to pull five g's, he or she wants the aircraft to respond fairly quickly (T_r), not overshoot the mark excessively (M_p), get to the commanded value quickly (T_{sl}), and stay there (e_{cmd}). He or she also doesn't want the aircraft to initially go in the wrong direction (U_s), or if this is unavoidable, the pilot would at least like this nonminimum phase response to be minimized.

3.4 Stability Margins

In designing a control system for an aircraft (or any plant), the control engineer is forced to work with a mathematical model which is an approximation of the actual system. The model contains approximation errors and perhaps unmodeled modes or dynamics.

The accumulation of these errors is known as the plant uncertainty. The control engineer seeks to design a control system that allows the closed loop system to retain its stability and performance characteristics in the face of a certain level of uncertainty. Measures of the system's resistance to being destabilized by this uncertainty are known as the stability margins. Classically, these are the *gain* and *phase margins*, that amount of gain or phase change that results in a change in the encirclement of the critical point (-1 for negative feedback) in the well known Nyquist Stability Criteria.

A slightly different representation of stability margins, known as *vector margins* can also be calculated from the polar plot of the open loop or *loop transfer function* $L(s)=PK$. The point of closest approach of the polar plot to the critical point is known as the *M circle* radius. This is illustrated in figure 3.2. Note that the critical point is +1 since all feedback is considered positive. Franklin [11] demonstrates that the minimum distance from any point on the polar plot to the critical point (the M circle radius), is the reciprocal of the magnitude (for SISO) or infinity norm (for MIMO) of the sensitivity. Since all feedback is considered positive, the sensitivity is defined as

$$S = [I - PK]^{-1} \quad (3.1)$$

and thus the M circle radius is

$$M_{\text{rad}} = \frac{1}{\|S\|_{\infty}} \quad (3.2)$$

where $\|S\|_{\infty}$, the infinity norm of the sensitivity, is the supremum over all frequency of the maximum singular value of $L(s)$.

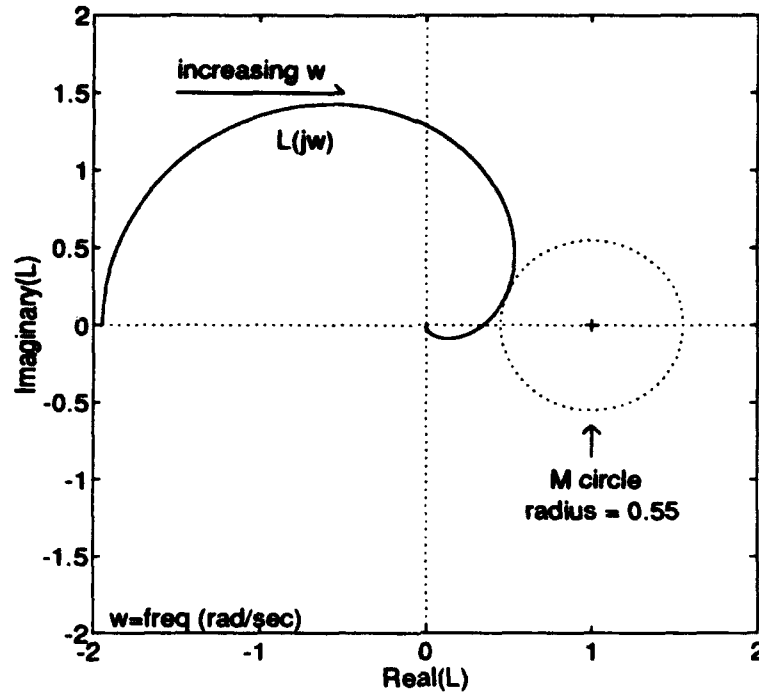


Figure 3.2. Polar Plot with M Circle Radius

These vector margins can be expressed as

$$GM_u = -20 \log \left(1 - \frac{1}{\|S\|_\infty} \right) \quad (\text{upper gain margin (dB)}) \quad (3.3)$$

$$GM_l = -20 \log \left(1 + \frac{1}{\|S\|_\infty} \right) \quad (\text{lower gain margin (dB)}) \quad (3.4)$$

$$PM = \pm 2 \arcsin \left(\frac{1}{2\|S\|_\infty} \right) \quad (\text{phase margin (deg)}) \quad (3.5)$$

The advantage of vector margins over classical margins is that they take into account simultaneous perturbations in both gain and phase, whereas the latter do not. Thus classical margins may give an overly optimistic prediction of the stability of the system in the face of plant uncertainty. Therefore, vector margins are always more conservative.

Because $\|S\|_{\infty}$ can never be less than one, the vector margins described in equations (3.3), (3.4), and (3.5) have a limited range of values which are:

$$0.0 < GM_u \leq \infty$$

$$-6.0 \leq GM_l < 0.0$$

$$0.0 < PM \leq 60.0$$

This thesis will use vector margins as a measure of performance and a constraint for the closed loop systems to be developed by QDES. QDES allows easy constraint of these margins by limiting the maximum value a specified SISO transfer function may take on. If this transfer function is the input or output sensitivity, then the M circle radius and thus the vector margins can be constrained as desired. The actual mechanics of this process are discussed in section Chapter VII.

3.5 *Summary*

This chapter discussed the measures of performance to be evaluated while using QDES to explore the limits of achievable performance for aircraft. These measures were divided into the categories of aircraft performance, time response characteristics, and stability margins. A brief explanation of vector margins was also included.

IV. Classical Design

4.1 Introduction

As stated in Section (1.5), a control design for N_z to N_{zcmd} for a fighter aircraft was developed using classical control methods. The purpose was two fold. First, the results would show what level of performance in terms of time response characteristics and stability margins could be achieved manually. Secondly, the classical design would provide a consistency check for the QDES results and verify that the program was being used properly.

4.2 Axes System and Sign Convention

Before discussing the aircraft model to be used, it is necessary to specify the axis system and sign convention used in the model. The linearized equations of motion for the longitudinal axis are usually done in the stability axis system. In this system, x_s is aligned with the aircraft velocity vector (assuming no sideslip), z_s is down, orthogonal to x_s and in the aircraft plane of symmetry, and y_s is directed out the right wing, orthogonal to x_s and z_s . The equilibrium flight path angle (γ_0) is measured between the local horizon (x_v in the vehicle carried axis) and x_s . The trim angle of attack (α_{trim}) is measured from x_s to some specified fixed reference on the body, usually either the wing cord line or the zero lift line. These angles and axes are illustrated in figure (4.1).

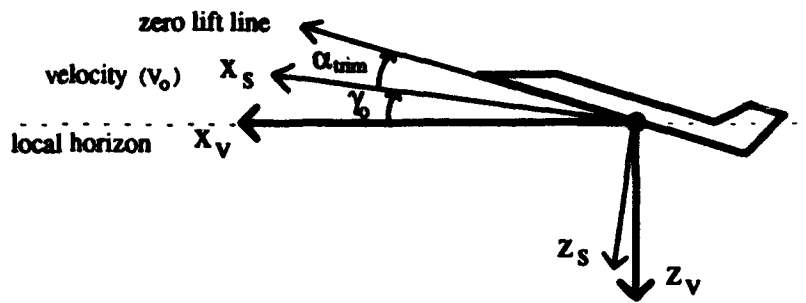


Figure 4.1. Longitudinal Axis System

The aircraft pitch angle (θ), angle of attack (α), flight path angle (γ), and pitch rate (q) are defined as positive in the nose up direction. The elevator deflection (δ_e) is positive trailing edge down as shown in figure (4.2).



Figure 4.2. Elevator Sign Convention

A positive elevator deflection then results in a positive normal acceleration (N_z) which is nose down. This is the normally accepted convention. However, for this thesis, positive N_z will be taken as nose up and the aircraft C matrix will be modified to reflect this. From a performance perspective this makes more sense as most designers and flyers are interested in how many positive g's as seen in the cockpit (nose up) the aircraft can *pull*. The ability of an aircraft to *pushover* more than two or three g's is not of great operational importance.

4.3 Aircraft Model

The aircraft longitudinal model used in this and the QDES designs is derived from the linearized longitudinal equations of motion at a specified flight condition with its attendant equilibrium values. This linearization process is thoroughly developed in both Etkin [9] and McRuer [13]. The assumptions and simplifications associated with it were

listed in Section (1.3). The data chosen for this design is the A-4D Skyhawk at flight condition 5 as listed in McRuer [13]. Flight condition 5 parameters are in table (4.1).

Altitude (h) (ft)	Mach (M)	Equilibrium Velocity (V_0) (ft/sec)	Trim AOA (α_{trim}) (deg)	Eq Flt Path Angle (γ_c) (deg)
15,000	0.6	634	3.4	0.0

Table 4.1. A-4D Flight Condition 5 Parameters

The state space representation for the longitudinal axis is shown in Appendix (A), section (A.1).

4.4 Inner Loop Pitch SAS Theory

Control of an aircraft longitudinal axis which exhibits the classical *Short Period* and *Phugoid* modes (as does the A-4D) is complicated by the very low Phugoid damping ratio (ζ_p). Furthermore, the uncompensated system does not have a pole at the origin of the s-plane and thus may exhibit an unacceptable steady state error. Figure (4.3) shows the A-4D N_z response using a static gain feedback of N_z to a unit step command, N_{zcmd} . Note the poor damping and aforementioned steady state error. The static gain feedback cannot correct the damping problem because, although it drives the Phugoid poles to their zeros, decreasing their influence and increasing ζ_p , it does so at the expense of Short Period damping (ζ_{sp}). Also, because it adds no pole at the origin, e_{ss} is quite poor (100%). A proportional plus integral (PI) controller corrects the e_{ss} problem, but suffers from the same limitations in terms of damping. Thus, a method known as Inner Loop Pitch Stability Augmentation System (SAS) was developed. This method uses an inner loop static gain to feed back pitch rate (q) around the plant to increase ζ_{sp} . A PI controller is then placed in the outer loop ahead of the plant to assure zero e_{ss} and to draw the locus

further into the LHP. A gain is then selected which results in an acceptable ζ_{sp} and ζ_p .

Figure (4.4) shows the block diagram of this arrangement. Application of this process to the A-4D follows in the next section.

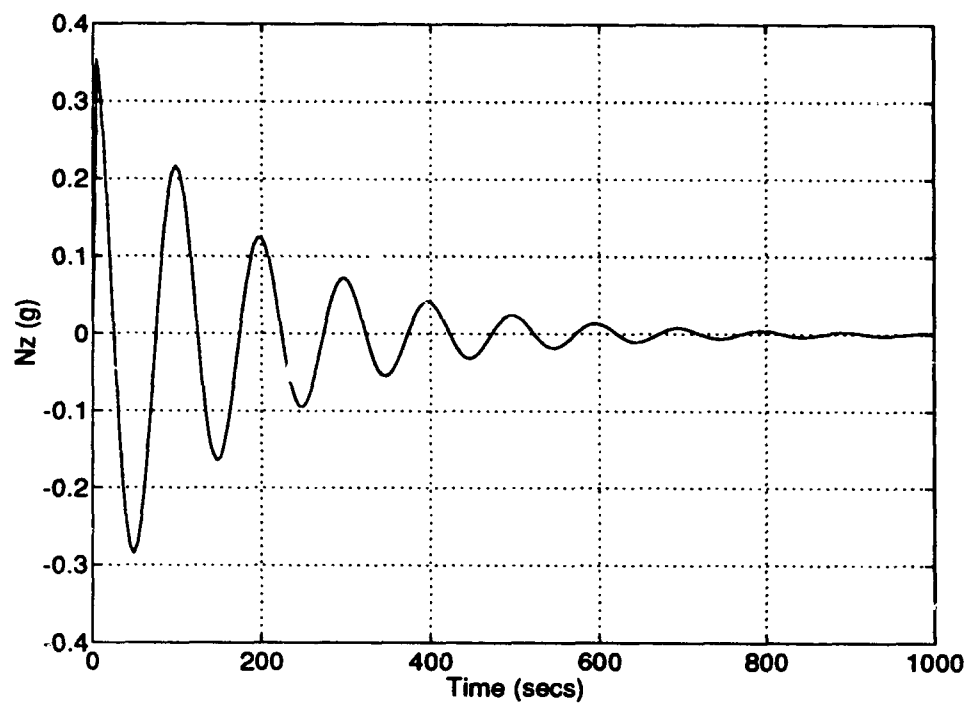


Figure 4.3. A-4D N_z to N_{zcmd} Static Gain Feedback

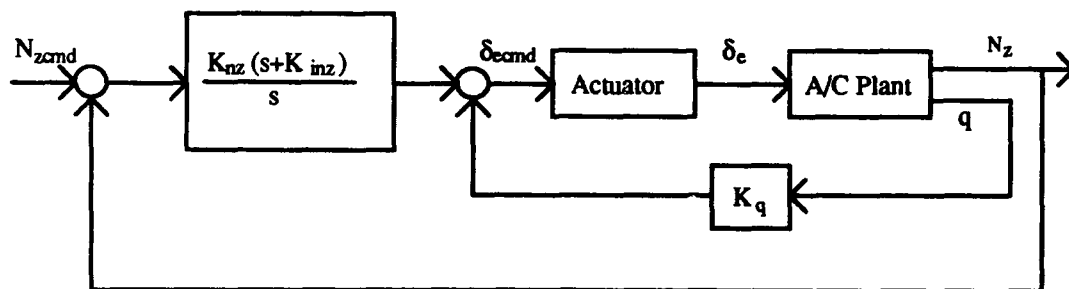


Figure 4.4. Inner Loop Pitch SAS

4.5 Inner Loop Pitch SAS for the A-4D

To choose the inner loop gain, start with the root locus of $\frac{q}{\delta_{\text{cmd}}}$ as shown in figure (4.5).

The open loop poles of the longitudinal axis are:

$$\begin{aligned} & -1.12012 \pm 3.50943j \\ & -0.005512 \pm 0.06324j \\ & -10.0 \end{aligned}$$

Examination of the root locus shows that the best ζ_{sp} that can be achieved is about 0.65. This occurs at an inner loop gain (K_q) of -1.883. The loop is closed resulting in the following inner loop closed loop poles:

$$\begin{aligned} & -4.0001 \pm 4.74989j \\ & -.00528 \pm 0.05758j \\ & -4.24067 \end{aligned}$$

which now become the open loop poles of the outer loop.

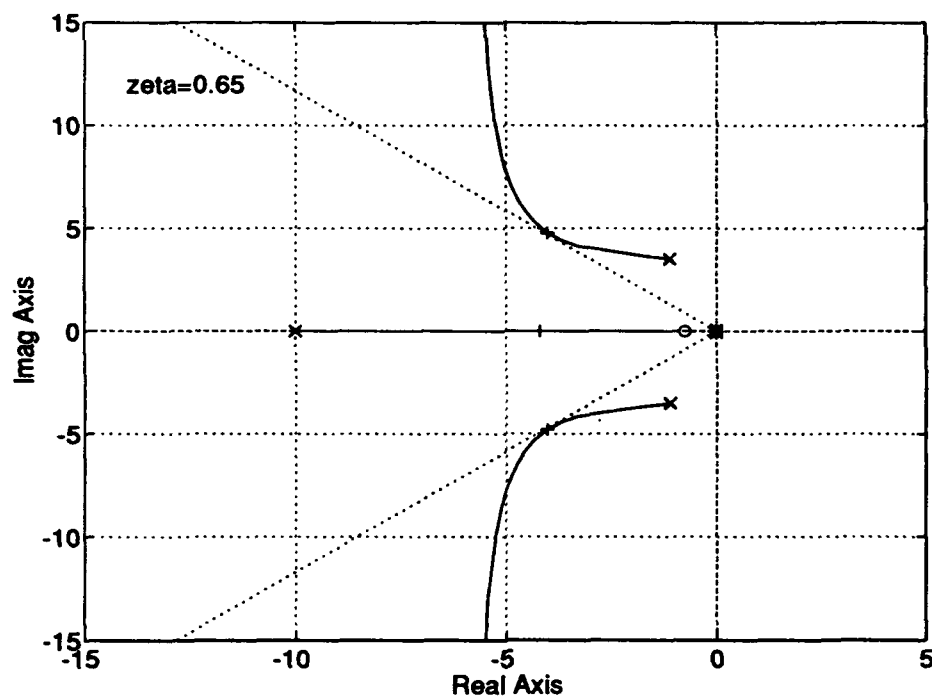


Figure 4.5. A-4D q to δ_{cmd} Root Locus

Next, the outer loop root locus of $\frac{N_z}{\delta_{ecmd}}$ is examined to determine the best location for the

zero inserted by the PI controller via the parameter K_{inz} . Placing the zero at -8 results in the root locus shown in figure (4.6). This placement pulls the locus further into the LHP. Selection of the outer loop gain (K_{nz}) of 0.7 results in the best combination of ζ_{sp} and ζ_p .

The closed loop poles are:

$$-3.84535 \pm 3.42199j$$

$$-2.16893 \pm 2.45225j$$

$$-0.007114$$

$$-0.0000003$$

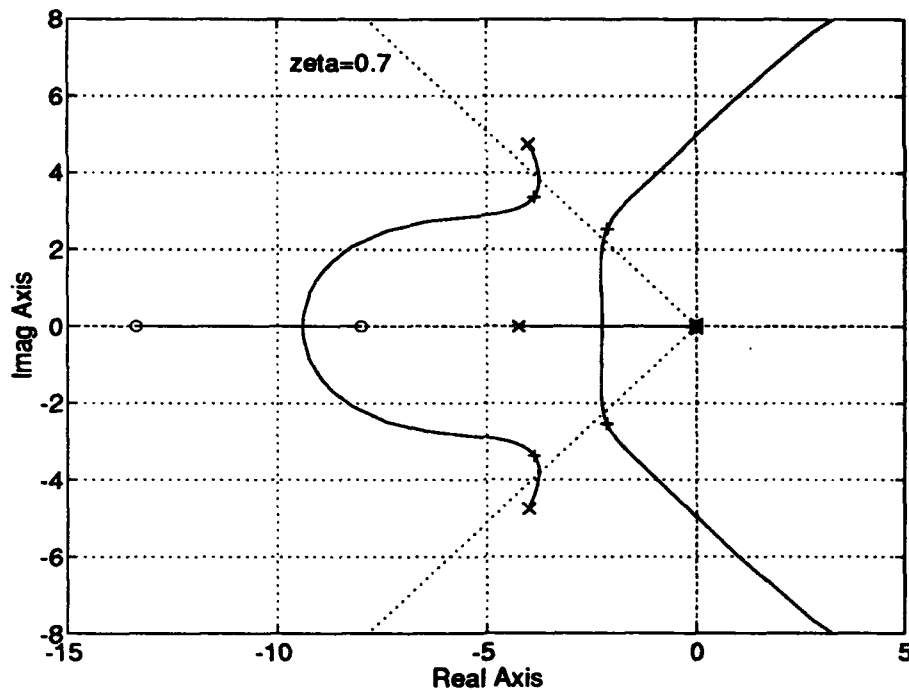


Figure 4.6. A-4D Outer Loop N_z to δ_{ecmd} Root Locus

The loop is then closed and the system response $\frac{N_z}{N_{zcmd}}$ to a unit step input is examined.

This time response with its measures of performance are shown in figure (4.7). The elevator deflection (δ_e) is shown in figure (4.8) while the change in forward velocity (v) and angle of attack (α) are shown in figure (4.9).

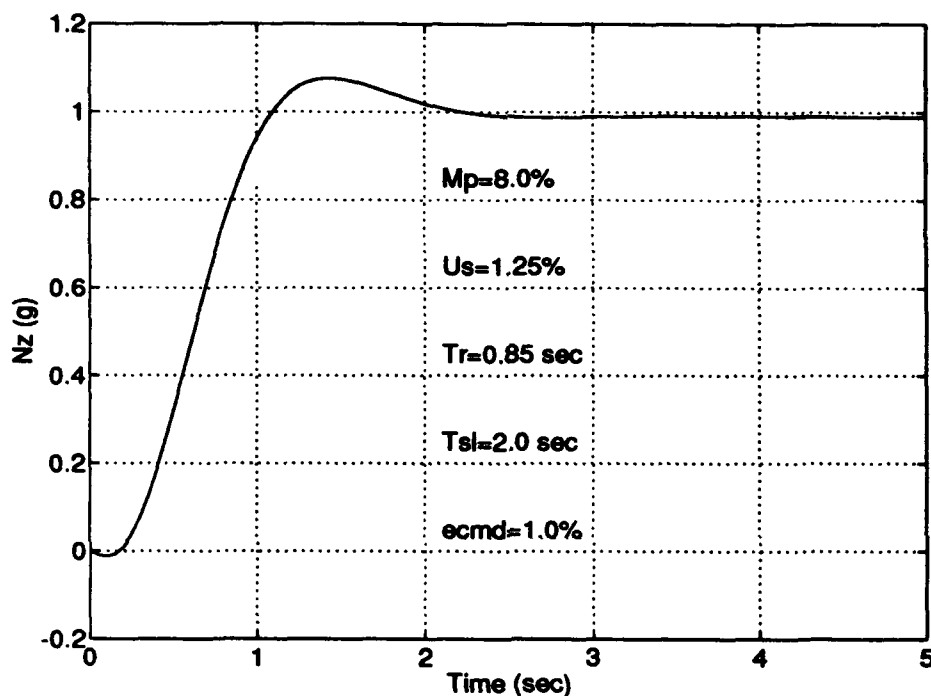


Figure 4.7. A-4D Inner Loop Pitch SAS Unit Step Response

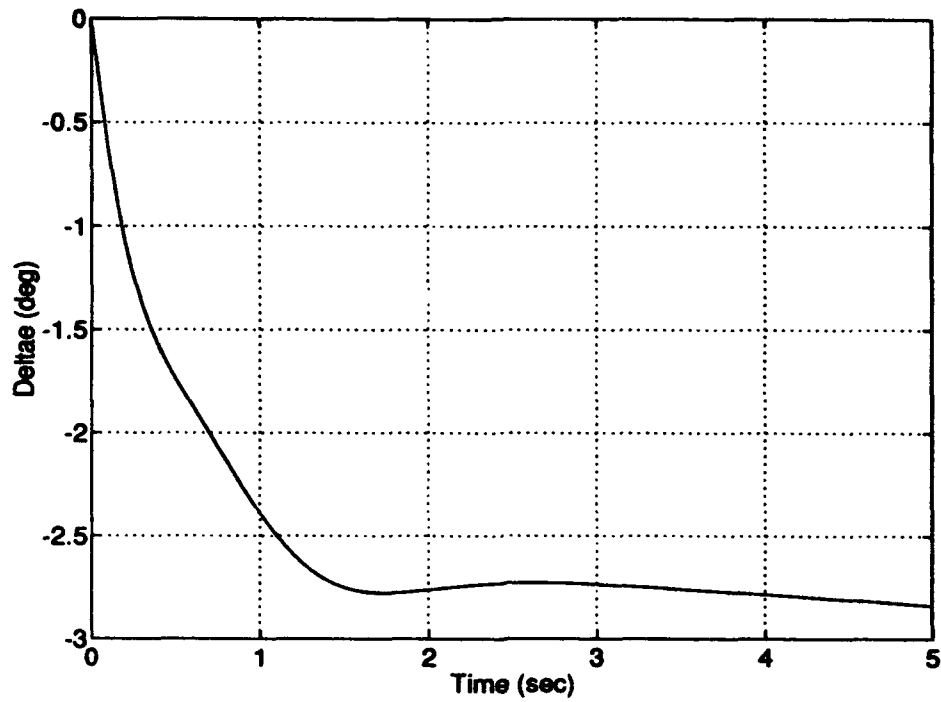


Figure 4.8. A-4D Elevator Deflection For 1 g Step Response

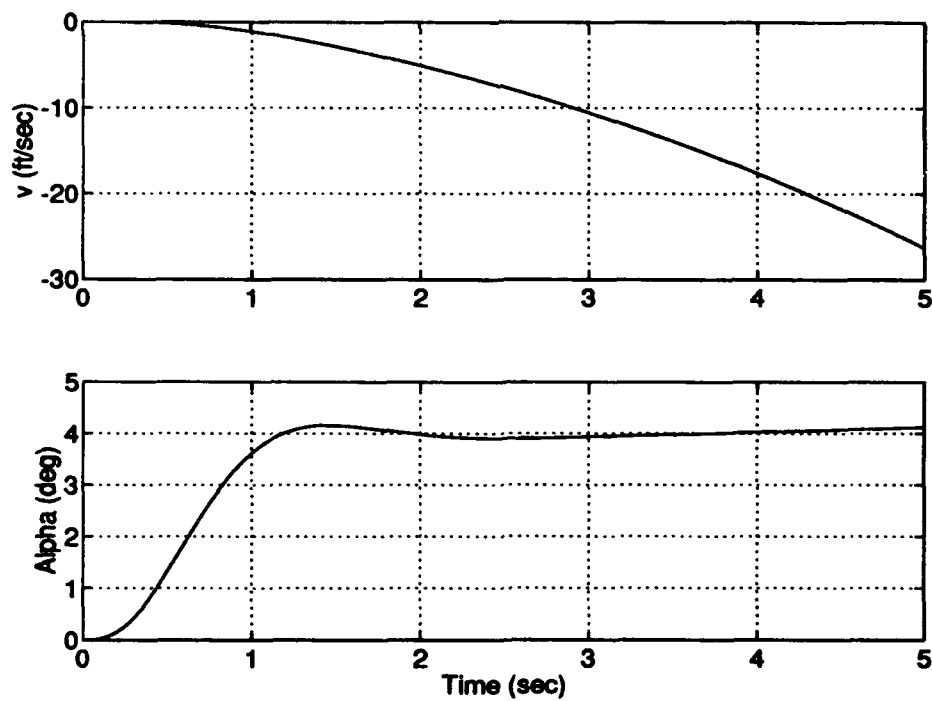


Figure 4.9. A-4D Forward Velocity and AOA Change For 1 g Step Response

It is important to remember that these are *perturbation* quantities away from the equilibrium values as shown in table (4.1). Thus, since the aircraft is in steady level 1 g flight, the actual N_z the pilot feels is 2 g's. Likewise, for the angle of attack it is necessary to add the α_{trim} value to that shown in figure (4.9) to determine what α the aircraft actually experiences. The same holds for forward velocity v change and its equilibrium value (V_0).

Figure (4.10) shows the polar plot of L for the A-4D with the inner loop pitch SAS controller. The M circle radius of 0.62 results in the following vector margins for this design:

$$GM_u = 8.4 \text{ dB}$$

$$GM_l = -4.2 \text{ dB}$$

$$PM = \pm 36 \text{ deg}$$

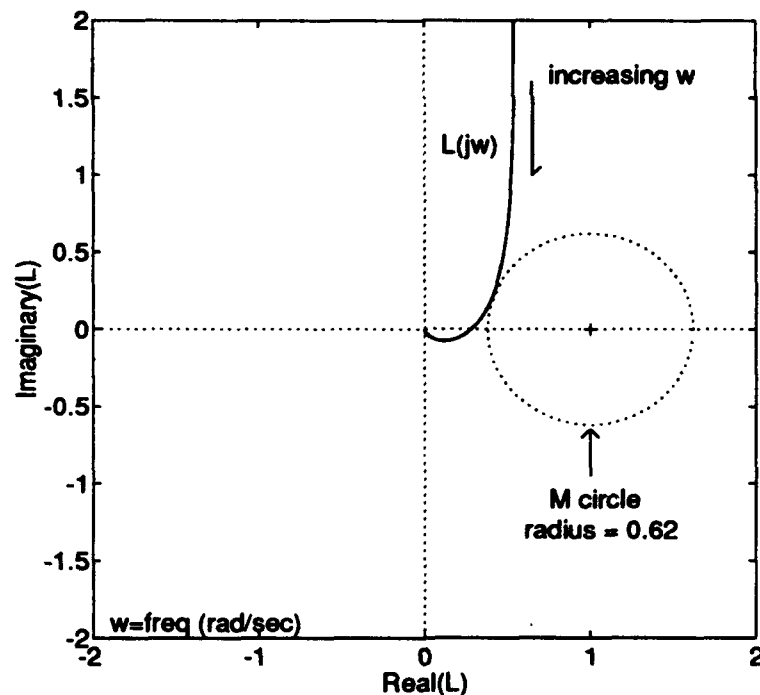


Figure 4.10. Polar Plot of A-4D With Inner Loop Pitch SAS

4.6 Summary

This chapter introduced the aircraft model for the longitudinal axis and defined the axis system and sign convention used. The inner loop pitch SAS theory was briefly explained and then applied to the A-4D aircraft N_z to N_{zcmd} problem. The results shown provide a useful comparison with the results obtained with QDES for the same problem.

V. QDES Design for The A-4D

5.1 Introduction

This chapter explores using QDES to solve the N_z to N_{zcmd} control problem for the A-4D as was done classically in Chapter IV. With the classical design, the process was performed and the controller developed resulting in certain measures of performance which were evaluated after the fact. The ability to go back to say, achieve better stability margins or a faster rise time, is theoretically possible, but would be more of a trial and error process. With QDES the process is reversed. Starting with the same plant model, one can specify several characteristics of the desired closed loop system and then let QDES attempt to develop the controller that will deliver them. This chapter will walk through that process, ultimately using QDES to evaluate the limits of performance for the A-4D as defined in Chapter III. Specific details of using QDES are reserved for the step-by-step procedure outlined in Chapter VII.

5.2 Model Development

The first step in using QDES is developing the model. Through this process, the various input and output vectors to the system as described in Section (2.2) are identified. This makes the necessary transfer functions available for QDES to use as it seeks to find a control law to satisfy the specified constraints. For the A-4D example, the vectors are:

- Exogenous Inputs (w):
 - w1: Normal Acceleration Command (N_{zcmd})
 - w2: Loop Input for Stability Margin Constraint/M Measurement (U_{in})
- Regulated Outputs (z):
 - z1: Angle of Attack (α)

z2: Normal Acceleration of the Aircraft c.g. (N_z)

z3: Pitch Rate (q)

z4: Elevator Deflection (δ_e)

z5: Elevator Deflection Command (δ_{ecmd}) and Loop Output for Margins (U_{out})

- **Sensed Outputs (y):**

y1: Pitch Rate (q)

y2: Normal Acceleration of the Aircraft c.g. (N_z)

y3: Normal Acceleration Command (N_{zcmd})

- **Actuator Inputs (u):**

u1: Elevator Deflection Command (δ_{ecmd})

The rationale for these choices is as follows. The transfer function between U_{out} and U_{in} can be shown to be the sensitivity (S) as shown in equation (3.1). Thus, making this available to QDES allows constraint and observation of S , and therefore the stability margins as described in Section (3.4). Outputting δ_e and α allows observation and constraint of these values which is important for determining the limits of performance; i.e., when the stall α or maximum value of δ_e is reached, the maximum attainable N_z is determined. N_z and q were chosen for feedback as rate and acceleration sensors tend to provide better information than pure angular measurements. N_{zcmd} is also made available to the controller based on DeLaney's [6] experiences with QDES, and also because it simply makes sense to provide this known signal to the controller. The choice of N_{zcmd} as an input and N_z as an output is obvious.

Figure (5.1) shows the block diagram representation of the aircraft plant and controller with these input and output vectors. Note that this diagram represents the same relationship between the input and output vectors and the plant components as the standard feedback diagram of figure (2.1). However, this representation makes construction of the nine matrices used in equations (2.2), (2.3), and (2.4) much more

straightforward. In figure (5.1), G is the A-4D state space without the actuator as shown in Appendix A, Section (A.1.1), and G_a is the actuator state space, shown in Section (A.1.2). K_1 and K_2 together represent the single $K(s)$ in figure (2.1).

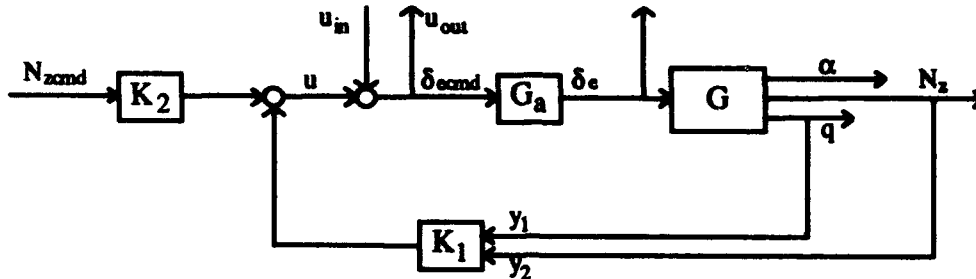


Figure 5.1. Block Diagram for A-4D QDES Design

Thus the state space equations of the plant for this model are:

$$\begin{bmatrix} \dot{u} \\ \dot{\alpha} \\ \dot{\theta} \\ \dot{q} \\ \dot{\delta_e} \end{bmatrix} = \begin{bmatrix} & & & & \\ & A_g & & & \\ 0 & 0 & 0 & 0 & -10 \end{bmatrix} \begin{bmatrix} u \\ \alpha \\ \theta \\ q \\ \delta_e \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 10 \end{bmatrix} \begin{bmatrix} N_{zcmd} \\ U_{in} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 10 \end{bmatrix} [\delta_{ecmd}] \quad (5.1)$$

$\begin{matrix} & & & & \\ & & & & \\ & & & & \\ & & & & \\ & & & & \end{matrix} \begin{matrix} A \\ B_w \\ B_u \end{matrix}$

$$\begin{bmatrix} \alpha \\ N_z \\ q \\ \delta_e \\ \delta_{ecmd} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ & C_{nz} & & & \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u \\ \alpha \\ \theta \\ q \\ \delta_e \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} N_{zcmd} \\ U_{in} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} [\delta_{ecmd}] \quad (5.2)$$

$\begin{matrix} & & & & \\ & & & & \\ & & & & \\ & & & & \\ & & & & \end{matrix} \begin{matrix} C_z \\ D_{zw} \\ D_{zu} \end{matrix}$

$$\begin{bmatrix} q \\ N_z \\ N_{zcmd} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ & C_{nz} & & & \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u \\ \alpha \\ \theta \\ q \\ \delta_e \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} N_{zcmd} \\ U_{in} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} [\delta_{ecmd}] \quad (5.3)$$

$\begin{matrix} & & & & \\ & & & & \\ & & & & \end{matrix} \begin{matrix} C_y \\ D_{yw} \\ D_{yu} \end{matrix}$

In equations (5.2) and (5.3), C_{nz} and D_{nz} refer to the "nz" row of the C and D matrices of G in figure (5.1).

5.3 Plant Discretization

As described in Section (2.7), the plant to be controlled by QDES must be discretized. For this thesis, this was easily accomplished by use to the "c2dm" command in MATLAB. The only decision that needs to be made is the discretization rate. This is driven by the highest frequency mode in the plant. In Chapter IV, the A-4D plant poles were identified. The corresponding frequencies are:

.06 rad/sec (.0095 Hz)

3.7 rad/sec (.5888 Hz)

10 rad/sec (1.592 Hz)

The highest frequency comes from the actuator pole. Houpis [12] recommends sampling at least twice as fast as the highest frequency (Shannon's Sampling Theorem). However, it has been suggested that sampling at five times the highest frequency is actually better [14]. Thus, for the A-4D, a sampling rate of 8 Hz was used. Certainly, a higher sampling rate can always be used, but there is little utility in doing so as is discussed in Chapter VII. The plant that was discretized is the A-4D as listed in Appendix (A), Section (A.1.3), which includes the actuator dynamics, and has been scaled to the proper units with N_z defined as positive up.

5.4 Generating the Impulse Response Matrices

As described in Section (2.7), QDES starts with the impulse responses of the three matrices (T1, T2, and T3) which describe the closed loop plant with a nominal stabilizing controller as represented in equation (2.29). To generate these for the A-4D example, a MATLAB script file "nzprp.m" was written as shown in Appendix (C), Section (C.1).

The file is heavily commented, identifying each step of the process with instructions as to what information is to be inserted for a generic aircraft plant model. The following is a walk through this file with explanations and rationale for the parameters and processes used.

The original plant model as listed in Section (A.1.1) is entered. The actuator dynamics are augmented to the plant and N_z is changed to the positive up convention resulting in the matrices in Section (A.1.2). Finally, the units are converted resulting in the state space shown in Section (A.1.3).

The scale factor is used to change the value of N_{zcmd} . Since QDES constrains the time domain characteristics in terms of the unit step response, the system needs to be *told* that a unit step input is really another value, say 4 g's. By dividing the system C matrix by the scale factor, QDES designs a closed loop system as if the commanded value was the scale factor times the unit step. The various constraints such as α and δ_e are properly interpreted while the internal states reflect their actual values for the non-unity input.

The plant is discretized using the "c2dm" command as previously discussed. Note that the command asks for the sample time (T_s) which is the inverse of the sample rate. Also, the type of discretization must be specified. In this case, the Zero Order Hold ('zoh') was used.

The nine matrices as developed in Section (5.2) are entered. Notice that wherever an entry is not either a one or a zero, the data from the discretized plant is inserted. Also, the matrices must be labeled as listed so as to be properly read by the MATLAB script file "obc2ttt.m".

As previously discussed, a nominal stabilizing controller must be designed to form the nominal closed loop plant. There are several ways to do this. The method employed here involves choosing a full state feedback gain matrix (K) using the MATLAB "dlqr" command and the Kalman filter gain matrix (L) using the command "dlqe". These

matrices are then combined with the plant information entered in the nine matrices using a parameterization similar to that listed in equations (2.33), (2.34), and (2.35) to form the nominal closed loop system. To use "dlqr", two weighting matrices must be chosen. The first is the state weighting matrix $Q1$. The method employed is the same as used by DeLaney [6] whereby the states are weighted by the inverse of the square of the maximum value the state is allowed to take on. For example, the A-4D's elevator is considered to be limited to ± 25 deg. Therefore, the δ_e state is weighted by 0.0016. The state weighting matrix for this example is:

$$Q1 = \begin{bmatrix} 0.00044 & 0 & 0 & 0 & 0 \\ 0 & 0.0016 & 0 & 0 & 0 \\ 0 & 0 & 0.00111 & 0 & 0 \\ 0 & 0 & 0 & 0.00111 & 0 \\ 0 & 0 & 0 & 0 & 0.0016 \end{bmatrix}$$

Note that this matrix must be positive semidefinite meaning that its eigenvalues are ≥ 0 . The other matrix to be chosen is the control weight. This value is chosen somewhat arbitrarily, and in the absence of any concrete guidance, DeLaney's number of 0.1 was used. Thus, the control weighting matrix is:

$$R1 = [0.1]$$

This matrix must be positive definite (eigenvalues > 0) to prevent the use of infinite control power. To use "dlqe" more weightings must be chosen. The first is G which determines what states could be affected by process noise. For an aircraft, this might be a wind gust. A value of unity is put on every state that could be affected by such a gust, and a zero on each that could not. G for this design is:

$$G = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

The other two weights to be specified are Q2 and R2. These represent the expected value of the state and measurement noises respectively. Again, using DeLaney's numbers:

$$Q2 = [0.1]$$

$$R2 = \begin{bmatrix} 0.001 & 0 & 0 \\ 0 & 0.001 & 0 \\ 0 & 0 & 0.001 \end{bmatrix}$$

The choice of weightings is somewhat arbitrary. In theory, as long as the nominal system is stable, its actual performance characteristics are not that important. QDES should arrive at the same closed loop system (if it can find a feasible solution at all), although it may take more or less time depending on how close the nominal system is to the desired system. As will be seen, this was *not* necessarily the case.

Once the weightings are chosen to derive K and L, two other pieces of information need to be made available to "obc2ttt.m" before it can be called. The first is the *source* file name. This name must match the QDES source file name and identifies the impulse response matrices (_t1, _t2, _t3) plus the "_ttt.mat" file with the current problem. For this example, A4D7NZ was used. Also, the number of samples must be specified. This is the plant discretization rate (Hz) times the time interval (sec) over which you desire QDES to solve the problem. The time period chosen for the design was five seconds. The rationale for this choice is that the purpose is to examine the maximum g attainable in addition to whether a controller can be designed to work at this level. Thus, compliance with the time and margin constraints is as important as finding when the δ_e and α limits are reached. Additionally, since control of thrust is not included in the model, examination of sustained

g capability is not really possible. Lastly, sustaining even moderate g for an extended period of time would result in a change in pitch angle (θ) outside that valid for the small angle approximation used in the linearization process alluded to in Section (1.3). Thus, for this example, a rate of 8 Hz for 5 sec yields 40 samples.

Finally, the MATLAB script file "obc2ttt.m" is called. This file actually calls the function "obc2ttt_fn.m". Note that to use "obc2ttt.m", the plant must be strictly proper, that is pydu must equal zero. Given the state space representations used in this thesis with the actuator dynamics augmented, this will always be the case. These files are supplied with the QDES tape as described in Erickson [8] as are the others necessary to carry out the process described in this thesis. The specifics of what files are needed and how to use them is described in Chapter VII. When the file "a4d7nzprp.m" was run, the A4D7NZ_t1, A4D7NZ_t2, A4D7NZ_t3, and A4D7NZ_ttt.mat files were output for later use by QDES.

5.5 *Writing the QDES Source File*

Once the impulse response matrices have been generated, the QDES source file must be written to specify the problem so QDES may attempt a solution. The source file is written in "C" and close attention to the examples in the QDES User Commands [4] is a must for avoiding syntax errors. The source file is divided into four parts which will be discussed separately. The source file for this A-4D example is shown in Appendix (C), Section (C.2).

5.5.1 The Preamble In the preamble, the exogenous inputs, regulated outputs, and sensed outputs as listed in Section (5.2) are defined by a symbol of the user's choice. They are also numbered, and this numbering must reflect the order of the variables listed in Section (5.2) and as used in equations (5.1), (5.2), and (5.3).

5.5.2 The Declarations Section This section tells QDES how many of each of the variables listed in the preamble as well as the number of actuators are present in the problem. Also input is the number of samples from the "nzprp.m" file and the number of FIR taps. The number of taps determines the number of terms in the finite difference equations which are the discrete equivalents of the continuous state-variable differential equation as shown in Chapter II. More taps equals more information that QDES has available to solve the control problem. There are restrictions to the number of taps used however. Boyd [4] recommends limiting the product of the number of actuators, number of sensed outputs, and the number of taps to less than 200 due to limitations of the convex problem solver used. For the problem set up as used in this thesis, that number is 66. Furthermore, DeLaney [6] states that the number of taps should be strictly less than the number of samples, or unconstrained dynamics may appear in the Q portion of the compensator. In a few attempts where this restriction was violated, the author did not experience this phenomenon. Lastly, the size of the QDES controller is equal to the size of the nominal controller plus the number of taps. Thus, if the size of the controller required is one of the elements of the design investigation, this value could be adjusted accordingly. More discussion of the choice of number of taps follows in Chapter VII. In this example, 35 taps were used. The last entry in this section are the names of the impulse response matrices that were created with "nzprp.m" using "obc2ttt.m".

5.5.3 The Objective Section This section tells QDES to minimize one of several available relationships between the inputs and outputs, or a time response characteristic as described in [4]. Use of objective functions was tried, but not used for the final results of this thesis for several reasons. First, based on his experience with QDES, DeLaney recommended not using them due to the time it takes for the program to run. The author found this to be a problem as well. Secondly, as the purpose here is to evaluate the

maximum N_z attainable by the aircraft given a set of constraints, there was no real need to attempt to minimize any of these constraints, and thus using an objective function only served to slow the process down.

5.5.4 The Constraints Section

The constraint section allows the user to impose limits on the closed loop system to be designed. These take the form of time response characteristics and various input/output relationships, again as described in [4]. For this thesis, three different time response constraints were used along with a constraint on the magnitude of the transfer function representing the sensitivity. The complete list of the constraints used is found in the source file in Section (C.2). A thorough discussion of the mechanics of writing the constraint section is found in Chapter VII. For purposes of this chapter, the time response constraints for this example were:

$$T_r < 1.5 \text{ sec}$$

$$M_p < 10 \%$$

$$U_s < 5 \%$$

$$T_{sl} < 2.5 \text{ sec (3 \%)}$$

$$e_{cmd} < 3 \%$$

These constraints form an envelope within which the time response is to remain. This is seen on the time response graphs that follow. These were chosen based on the results from the classical design, but relaxed slightly especially for e_{cmd} . Experience with QDES showed that making e_{cmd} needlessly restrictive often resulted in no solution. The elevator deflection was constrained to ± 25 deg and angle of attack as follows: $-5 \text{ deg} < \alpha < 18 \text{ deg}$. Note these are not necessary the actual A-4D limits but are representative of aircraft of this type. Also, it must be remembered that the values derived from the linearized model and thus from evaluation of QDES results are *perturbation* values. Thus, if the aircraft has an angle of attack limit of 20 deg and the flight condition α_{trim} is 1.5 deg, the

upper constraint would be given to QDES as 18.5 deg. Lastly, the vector margins [-4,7] dB and [± 35] deg were imposed by constraining $\|S\|_{\infty} < 1.6628$. Actually, if the sensitivity constraint is met, the vector gain margins will be better than those shown. This possibly confusing relationship is fully explained in Chapter VII. The choice of margin constraints was again driven by the results of the classical design and the fact that the lowest acceptable phase margin is generally considered to be 30 deg [11].

5.6 *Running QDES*

Once the impulse response matrices are developed and the QDES source file is written, the actual running of QDES is a simple matter. The specific mechanics of doing so is reserved for Chapter VII. When the QDES run is complete, QDES forms the "sourcefile.q" and "sourcefile.out" files which are subsequently used to form the QDES controller. If QDES cannot find a controller that meets the constraints, it will generate a message to that affect and output "sourcefile.last.q" and "sourcefile.out" files. The ".out" file can be examined to see what constraints QDES could not meet. Run times for the examples in this chapter were on the order of one minute or less, which works well with the iterative process necessary to determine the N_z limits of performance. Other combinations of numbers of samples, number of taps, and sample rates which were run to explore QDES' capability resulted in significantly longer run times.

5.7 *Forming the QDES Controller and Closed Loop System*

The QDES controller and closed loop system can be easily formed via the MATLAB script files "qobc2comp.m" and "gethzw.m" supplied with the QDES tape and described in [8]. To employ these, a MATLAB script file "nzpst.m" was written which is found in Appendix (C), Section (C.3). Again this file is heavily commented and explained

in detail in Chapter VII. It does require entering some of the same parameters as those entered in the "nzprp.m" file. The file also simulates the response of the QDES closed loop system (H_{zw}) to a unit step and outputs graphs of the N_z time response, the δ_e and α time histories, the Bode magnitude plot of the sensitivity, and the Bode magnitude of N_z to N_{zcmd} closed loop system.

5.8 A-4D QDES Design Results

This section examines the results of using QDES to determine the N_z limits of performance subject to the constraints listed in Section (5.4.4).

5.8.1 A-4D QDES Design, 1 g Response Figure (5.2) shows the time response of the A-4D N_z to N_{zcmd} for 1 g. Notice that this is not the picture of an ideal step response, but that it does remain within the time response constraint envelope. QDES only guarantees that the closed loop system meets the constraints, not that the controller developed produces a *nice* response. This is also consistent with the admonitions of the program's author, Dr. Boyd, to the affect that QDES was never intended to be used to *design* controllers. Also, notice at the five second point, the response is just outside the constraint envelope. This is due to the wording of the constraint which requires the response to meet the constraints up to the $n_sample-1$ point.

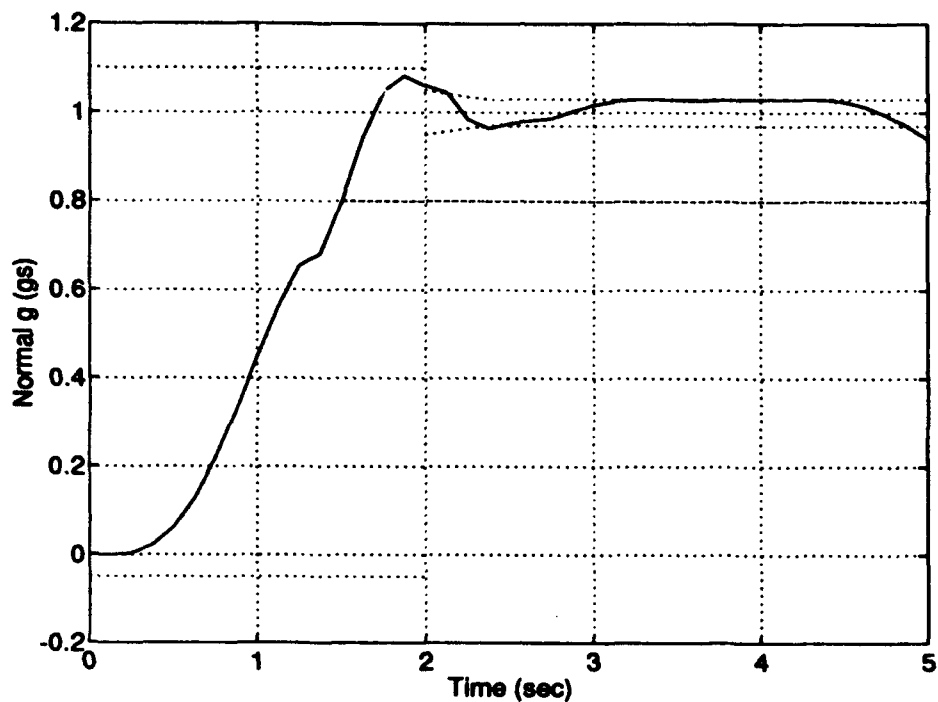


Figure 5.2. A-4D QDES Design, 1 g Response

Figure (5.3) shows the time histories of the elevator deflection and the angle of attack. From the graphs, one can estimate that the g response will probably be α -limited at about 4.5 g's when the α response hits its limits of 18 deg. QDES will verify whether a controller can be designed at this level and if any changes in the response can be made to achieve a higher level of performance.

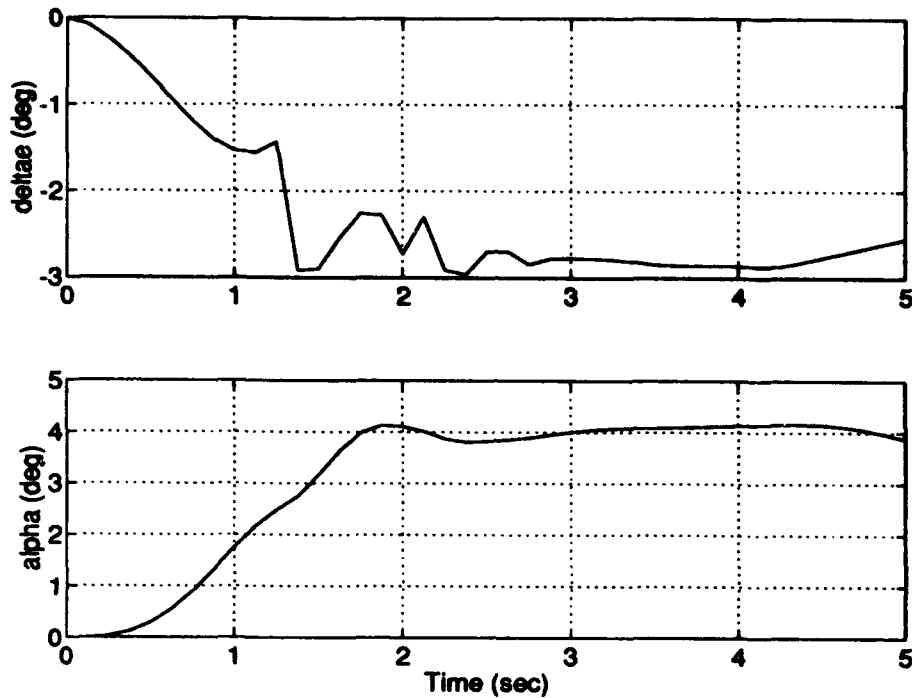


Figure 5.3. A-4D Elevator and AOA 1 g Responses

The top half of figure (5.4) shows the Bode magnitude of the input sensitivity in absolute magnitude, not in dB. The maximum value attained on this plot is $\|S\|_{\infty}$ from which the following vector margins for this design were calculated:

$$GM_u = -2.6 \text{ dB}$$

$$GM_l = +3.8 \text{ dB}$$

$$PM = \pm 20 \text{ deg}$$

Notice that the maximum value of the input sensitivity is greater than the value (1.6628) specified in the constraint section of the QDES source file even though QDES indicated that it had met all the constraints by returning a solution. Chapter VIII addresses this phenomenon and a theory as to why it occurs. The bottom half of figure (5.4) shows the Bode magnitude (dB) of the closed loop N_z to N_{zcmd} transfer function. This is included to examine the closed loop bandwidth (CLBW). When dealing with a discrete system, Anderson [1] suggests that one needs to look at the CLBW of the final design to ensure

that the ratio of the sampling rate to the CLBW is between about 4 and 20. For a sampling rate of 8 Hz (50 rad/sec), the CLBW would have to exceed approximately 12 rad/sec before the lower end of this range was neared. As seen in the plot, the CLBW is approximately 2.5 rad/sec (-3dB down point). Thus, the sample rate of 8 Hz is still valid.

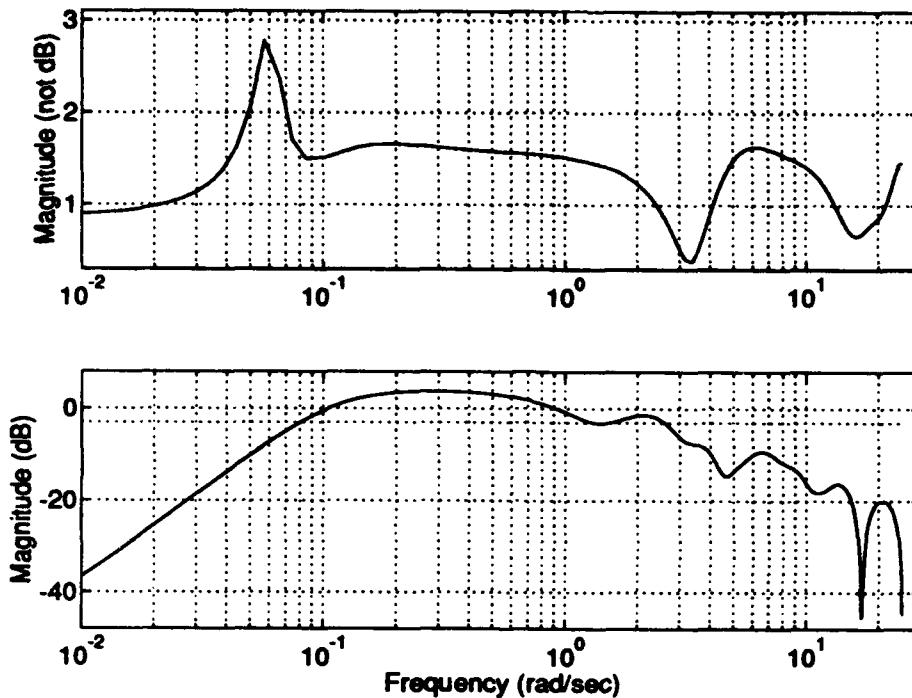


Figure 5.4. A-4D 1 g Input Sensitivity and CLBW

5.8.2 A-4D QDES Design, 3 g Response Figures (5.5), (5.6), and (5.7) show the same results for the 3 g response of the QDES design. The only marked difference between these results and those for 1 g occurs with the input sensitivity. In this case it is constrained to be less than 1.6628 resulting in the following vector margins:

$$GM_u = -4.3 \text{ dB}$$

$$GM_l = +8.7 \text{ dB}$$

$$PM = \pm 37 \text{ deg}$$

Again, Chapter VIII will examine this occurrence.

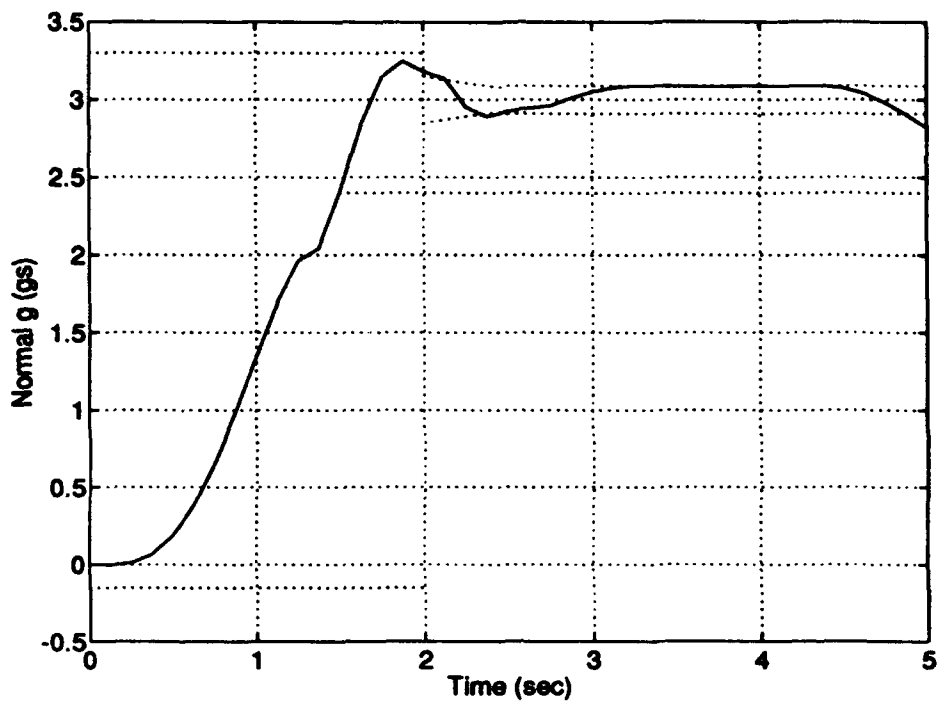


Figure 5.5. A-4D QDES Design, 3 g Response

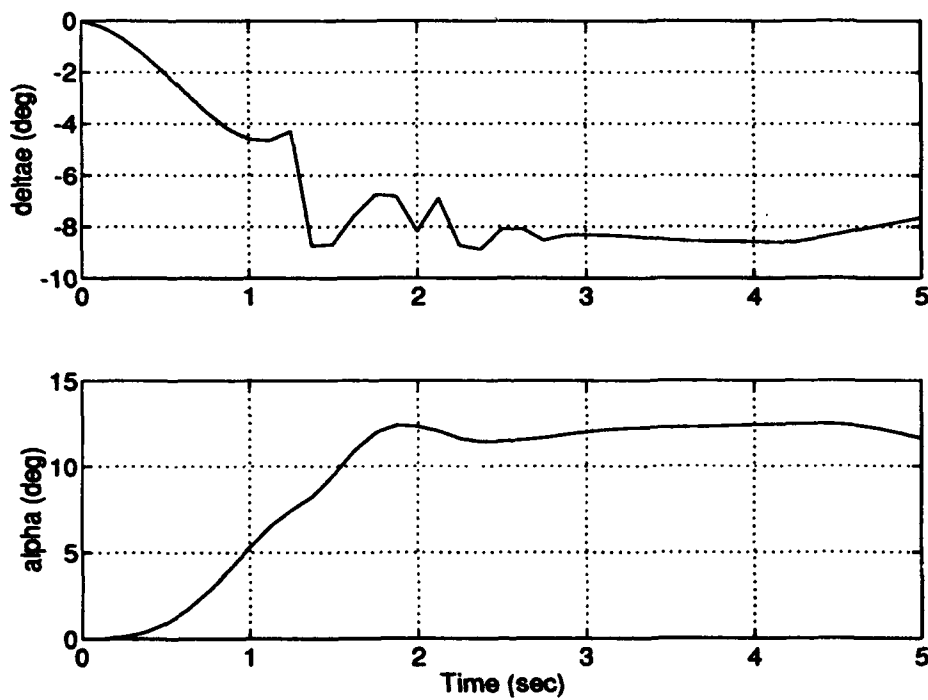


Figure 5.6. A-4D Elevator and AOA 3 g Responses

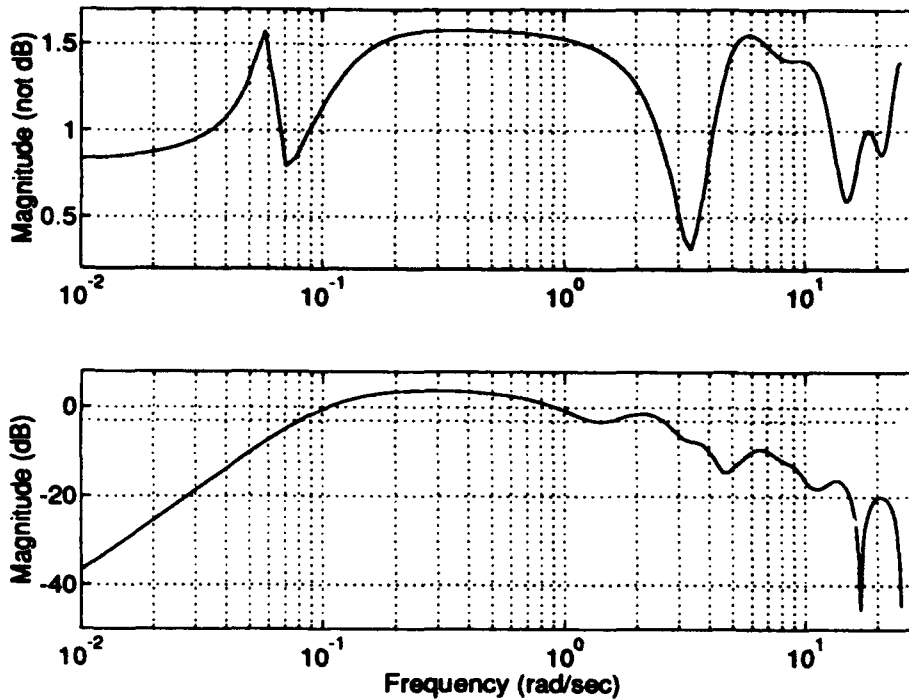


Figure 5.7. A-4D 3 g Input Sensitivity and CLBW

5.8.3 A-4D QDES Design, 4.5 g Response Figures (5.8), (5.9), and (5.10) show the results of the 4.5 g QDES design. Notice an expansion of the α plot shows it is right against the 18 deg limit. One might expect that the maximum N_z attainable for these constraints has been reached. According to QDES, this is the case as 4.6 g's was tried and QDES could not return a solution. The stability margins for this design are:

$$GM_u = -4.1 \text{ dB}$$

$$GM_1 = +8.1 \text{ dB}$$

$$PM = \pm 35 \text{ deg}$$

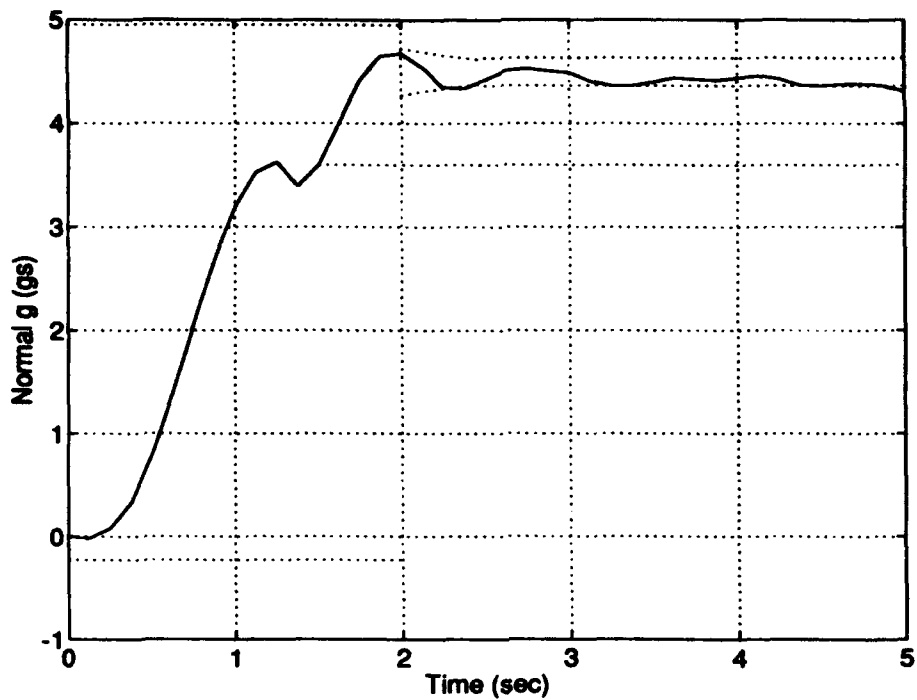


Figure 5.8. A-4D QDES Design, 4.5 g Response

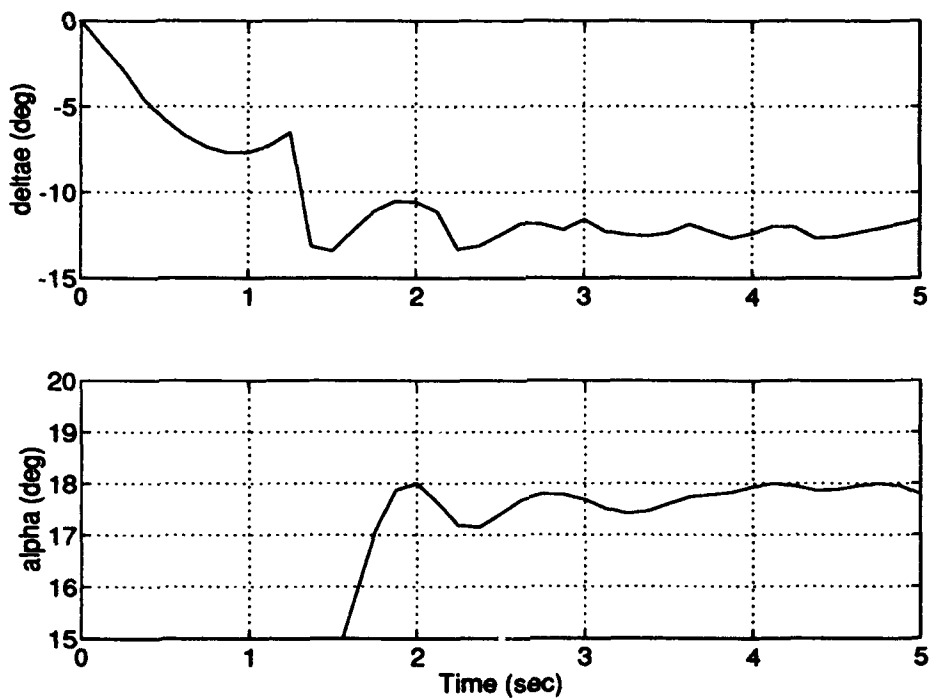


Figure 5.9. A-4D Elevator and AOA 4.5 g Responses

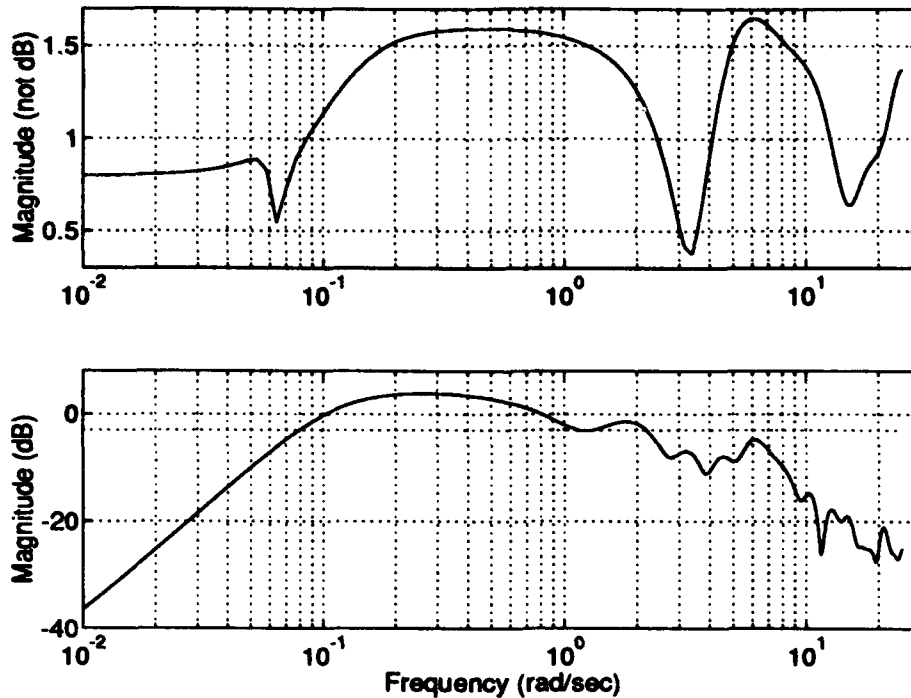


Figure 5.10. A-4D 4.5 g Input Sensitivity and CLBW

5.8.4 A-4D QDES Design, 4.6 g Response In an attempt to explore how QDES would modify its closed loop system to changes in the constraints, T_r and T_{sl} were relaxed by 0.5 sec and a 4.6 g problem was tried. Figures (5.11), (5.12), and (5.13) show the results. Note how the system displays some nonminimum phase activity and has reduced M_p such that α does not spike above 18 degs initially thus allowing a 4.6 g response. The gain and phase margins are identical to the 4.5 g example.

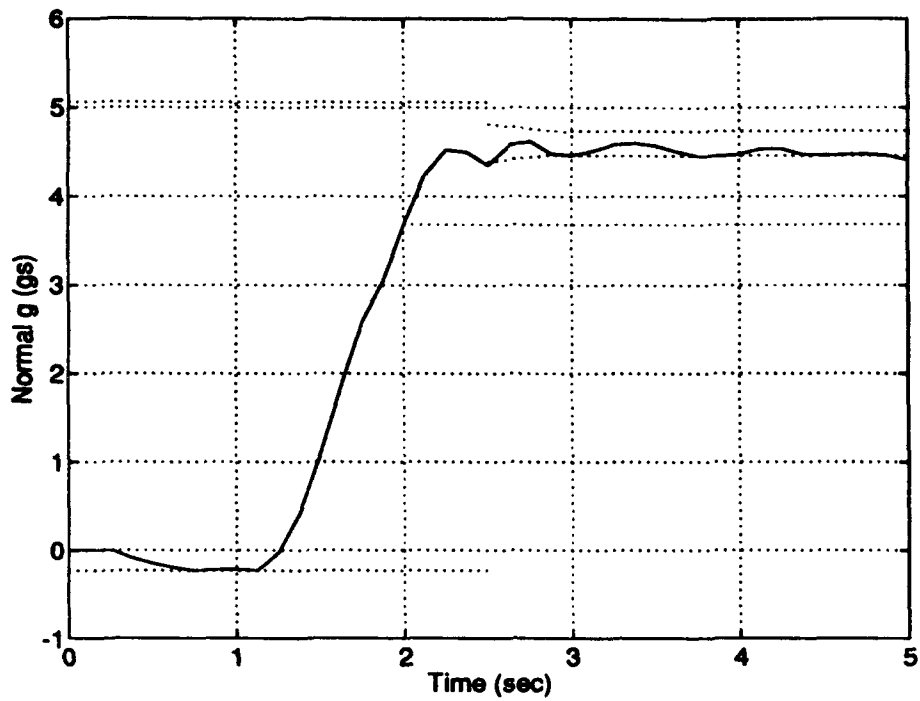


Figure 5.11. A-4D QDES Design 4.6 g Response

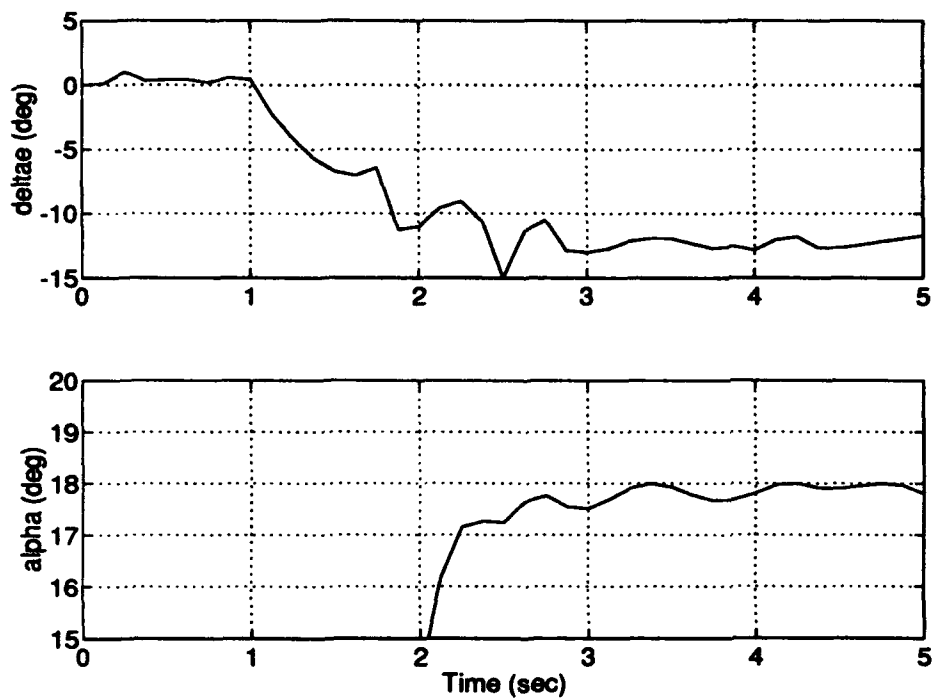


Figure 5.12. A-4D Elevator and AOA 4.6 g Responses

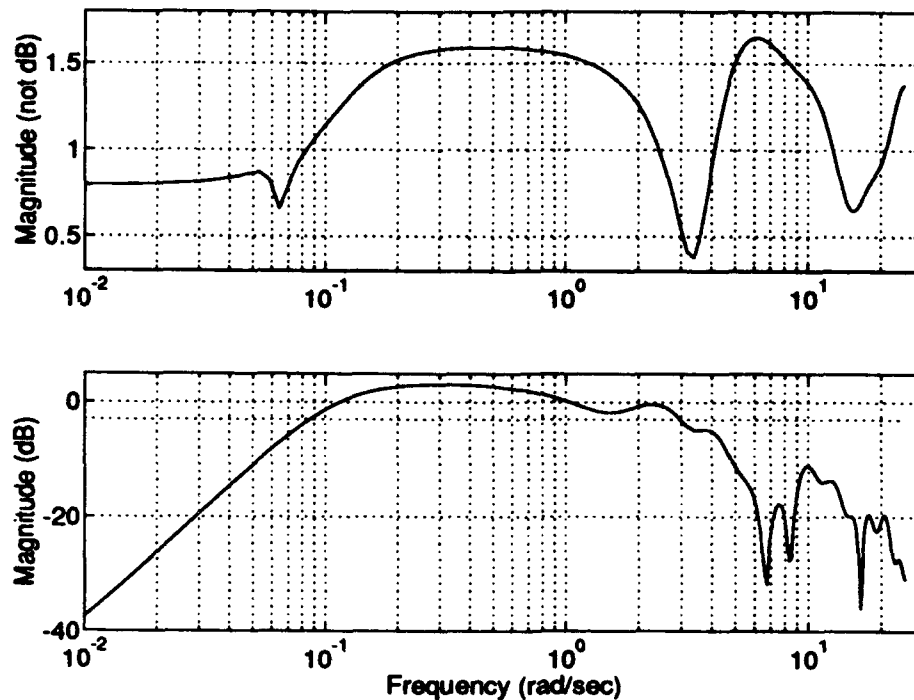


Figure 5.13. A-4D 4.6 g Input Sensitivity and CLBW

5.9 A-4D Short Period Approximation

Since the time period over which the response of the aircraft is being examined is relatively short, it is of interest to see what QDES will do using the Short Period approximation. For the Short Period approximation, the Phugoid mode is removed since its response is manifested at relatively long time periods after the application of the input. During the Phugoid part of the response, the forward velocity (v) changes very little. Therefore, to form the Short Period approximation, the v row and column are removed from the A matrix and the v row is removed from the B matrix in the A-4D model in Section (A.1.1). When this is done, the pitch rate (θ) column of the A matrix is all zeros, and the state space realization is non-minimal; i.e., the θ state is not needed. Therefore it is removed as well. The resulting state space realization with the actuator dynamics and

scaled to the proper units is listed in Appendix (A), Section (A.3). The state-variable differential equations for the A-4D Short Period approximation are:

$$\begin{bmatrix} \dot{\alpha} \\ \dot{q} \\ \dot{\delta}_e \end{bmatrix} = \underbrace{\begin{bmatrix} A_g & & B_g \\ & 0 & 0 \\ 0 & 0 & -10 \end{bmatrix}}_A \underbrace{\begin{bmatrix} \alpha \\ q \\ \delta_e \end{bmatrix}}_{B_w} + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 10 \end{bmatrix}}_{B_u} \underbrace{\begin{bmatrix} N_{zcmd} \\ U_{in} \end{bmatrix}}_{B_u} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 10 \end{bmatrix}}_{B_u} [\delta_{ecmd}] \quad (5.4)$$

$$\begin{bmatrix} \alpha \\ N_z \\ q \\ \delta_e \\ \delta_{ecmd} \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ C_{nz} & & D_{nz} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}}_{C_z} \underbrace{\begin{bmatrix} \alpha \\ q \\ \delta_e \end{bmatrix}}_{D_{zw}} + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}}_{D_{zw}} \underbrace{\begin{bmatrix} N_{zcmd} \\ U_{in} \end{bmatrix}}_{D_{zu}} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}}_{D_{zu}} [\delta_{ecmd}] \quad (5.5)$$

$$\begin{bmatrix} q \\ N_z \\ N_{zcmd} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 & 0 \\ C_{nz} & & D_{nz} \\ 0 & 0 & 0 \end{bmatrix}}_{C_y} \underbrace{\begin{bmatrix} \alpha \\ q \\ \delta_e \end{bmatrix}}_{D_{yw}} + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix}}_{D_{yw}} \underbrace{\begin{bmatrix} N_{zcmd} \\ U_{in} \end{bmatrix}}_{D_{yu}} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}}_{D_{yu}} [\delta_{ecmd}] \quad (5.6)$$

All processes and procedures for using QDES are the same as presented for the full A-4D model except that the Short Period plant was discretized at 20 Hz and 50 FIR taps were used. This provided a smoother response than that seen in the full A-4D model. Also, QDES was constrained to provide zero command error. This constraint was tried with the full model, but it did not improve the response as it did with the Short Period approximation and often it caused QDES not to return a solution.

5.9.1 A-4D Short Period QDES Design, 1 g Response The 1 g response for the A-4D Short Period QDES design is shown in figures (5.14), (5.15), and (5.16). The differences from the full model include a smoother response that more closely and steadily tracks commanded value and the sensitivity which lacks the low frequency spike shown in the full model. This is due to the absence of the low frequency Phugoid pole. The vector margins are:

$GM_u = -4.1 \text{ dB}$
 $GM_l = +8.3 \text{ dB}$
 $PM = \pm 36 \text{ deg}$

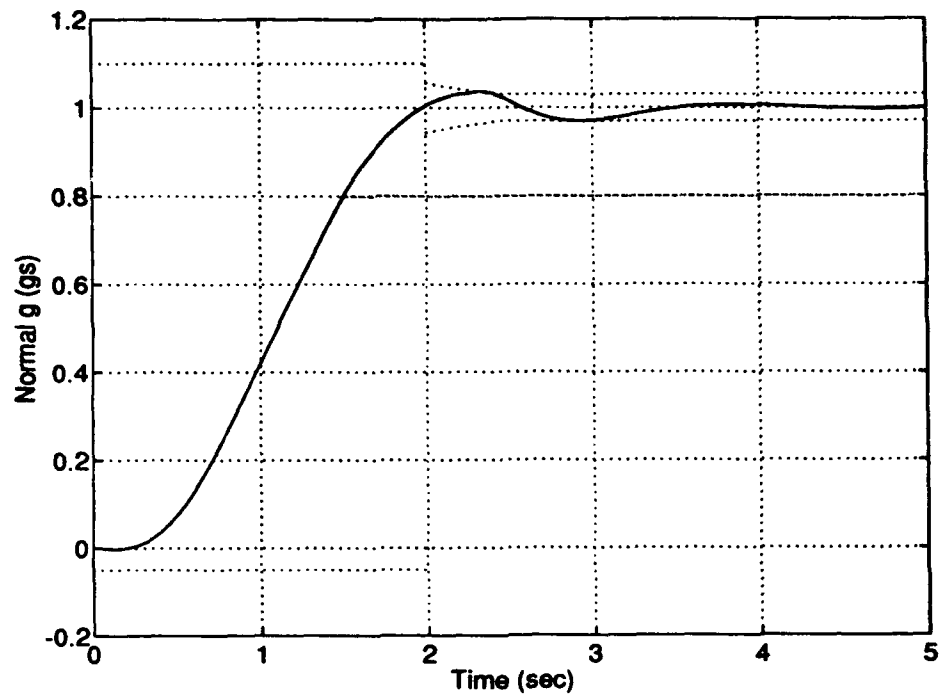


Figure 5.14. A-4D Short Period QDES Design, 1 g Response

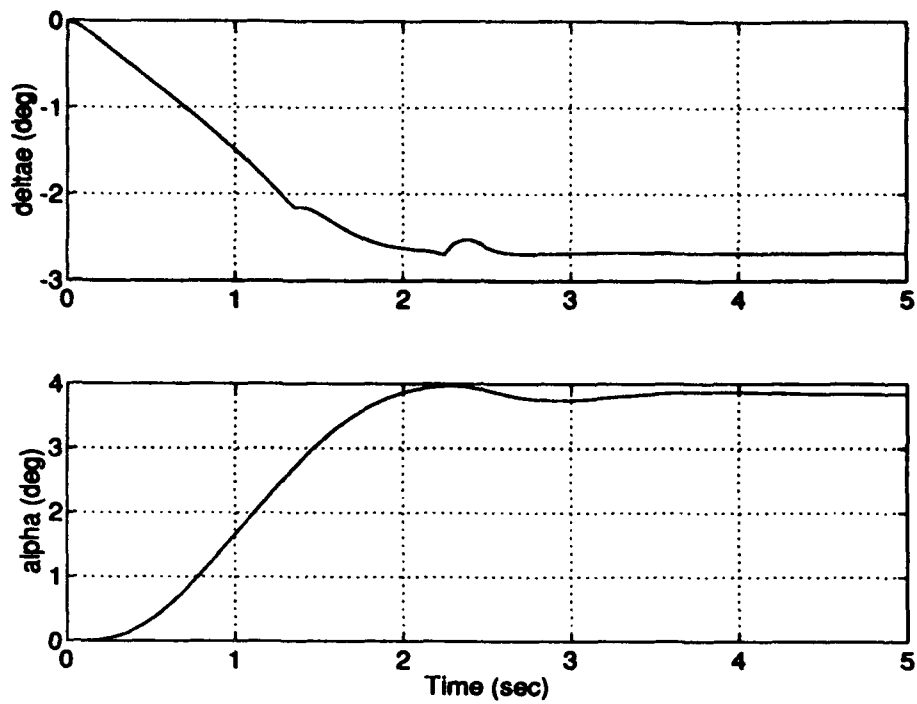


Figure 5.15. A-4D Short Period Elevator and AOA 1 g Responses

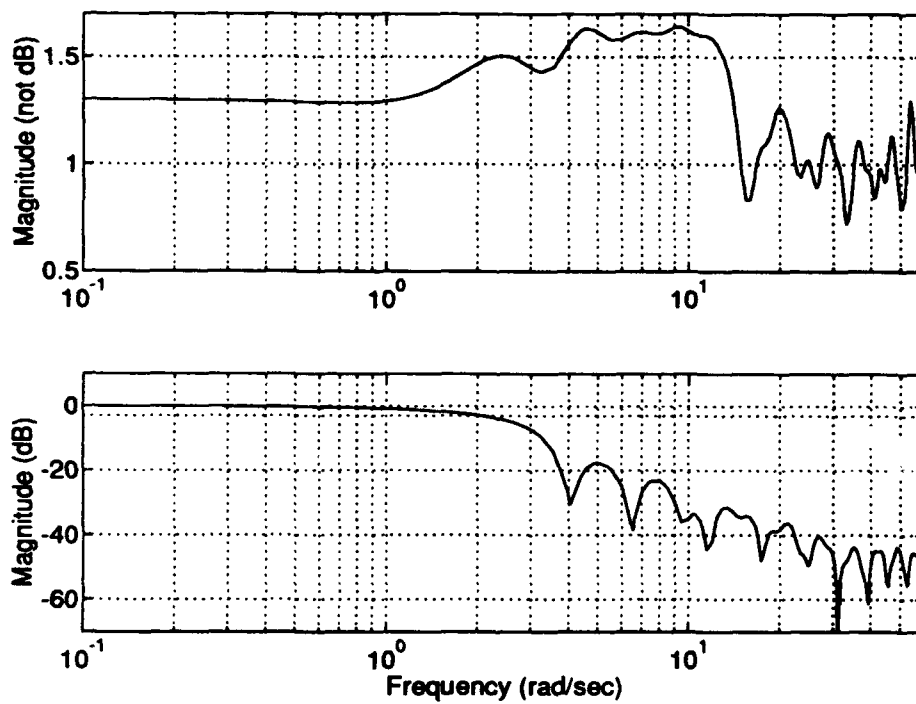


Figure 5.16. A-4D Short Period 1 g Input Sensitivity and CLBW

5.9.2 A-4D Short Period QDES Design, 3 g Response Figures (5.17), (5.18), and (5.19) show the A-4D Short Period QDES design for a 3 g input. The vector margins are:

$$GM_u = -4.0 \text{ dB}$$

$$GM_l = +7.8 \text{ dB}$$

$$PM = \pm 35 \text{ deg}$$

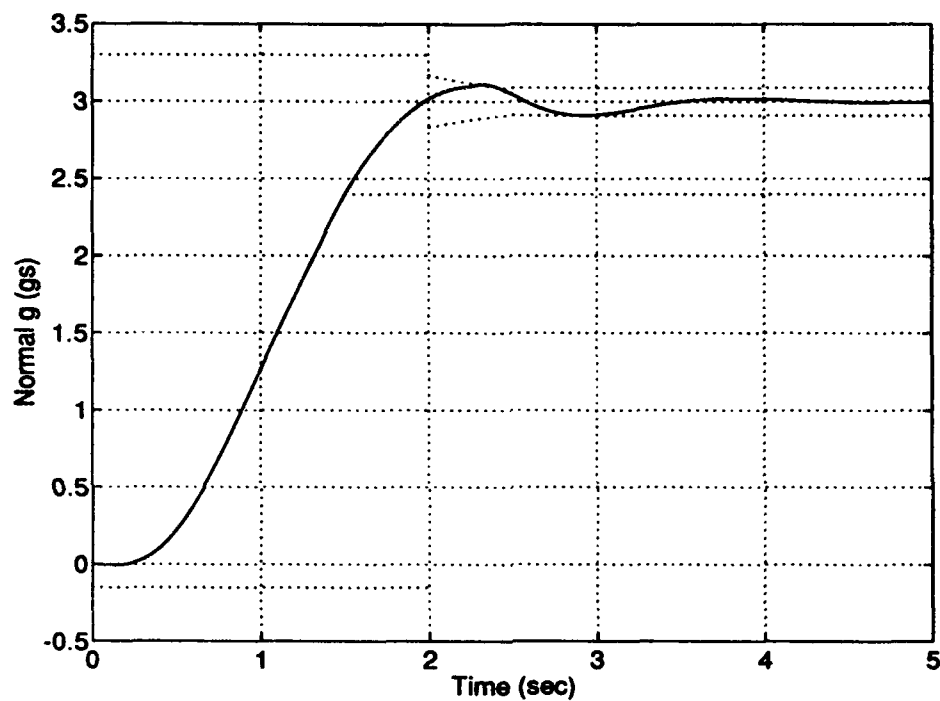


Figure 5.17. A-4D Short Period QDES Design, 3 g Response

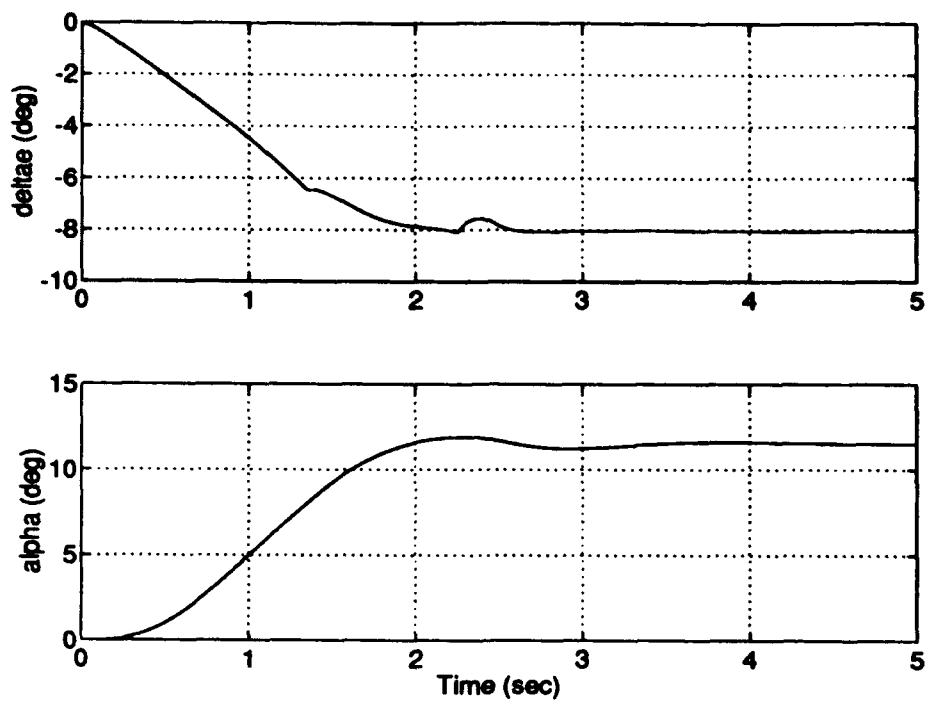


Figure 5.18. A-4D Short Period Elevator and AOA 3 g Responses

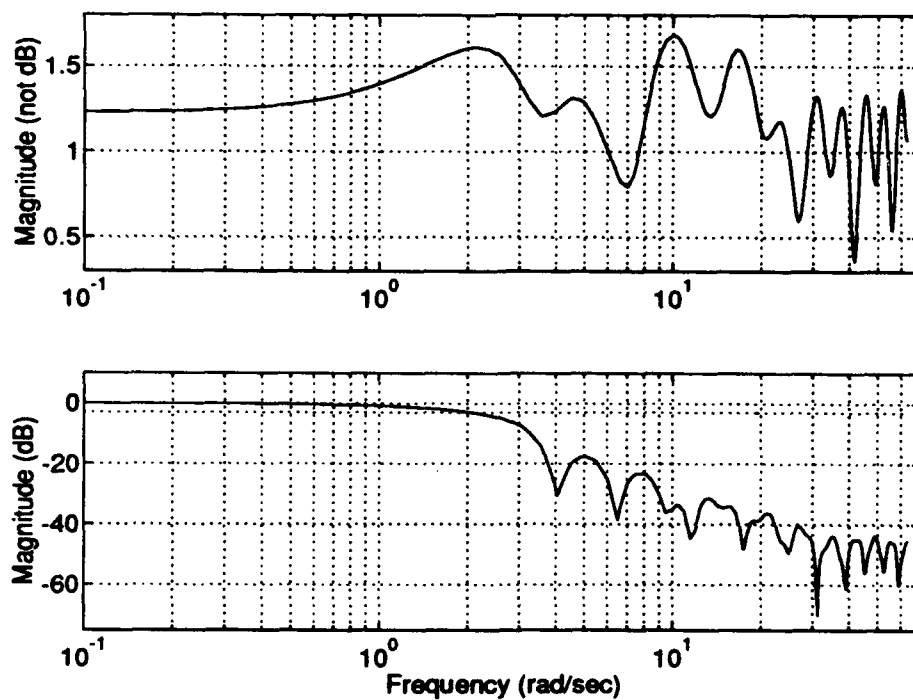


Figure 5.19. A-4D Short Period 3 g Input Sensitivity and CLBW

5.9.3 A-4D Short Period QDES Design, 4.6 g Response Figures (5.20), (5.21), and (5.22) show the A-4D Short Period QDES design for a 4.6 g command. Notice again that the response appears to be very close to the α limit of 18 deg. In fact, it is the limit for these constraints, as 4.7 g's was tried and QDES could not provide a solution. The vector margins are:

$$GM_u = -4.1 \text{ dB}$$

$$GM_l = +8.2 \text{ dB}$$

$$PM = \pm 35 \text{ deg}$$

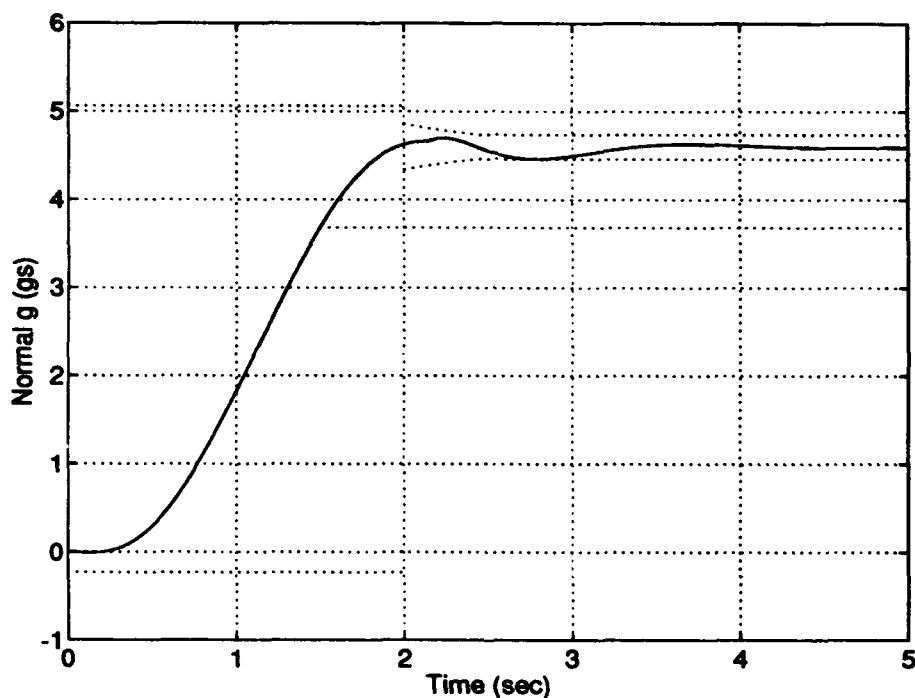


Figure 5.20. A-4D Short Period QDES Design, 4.6 g Response

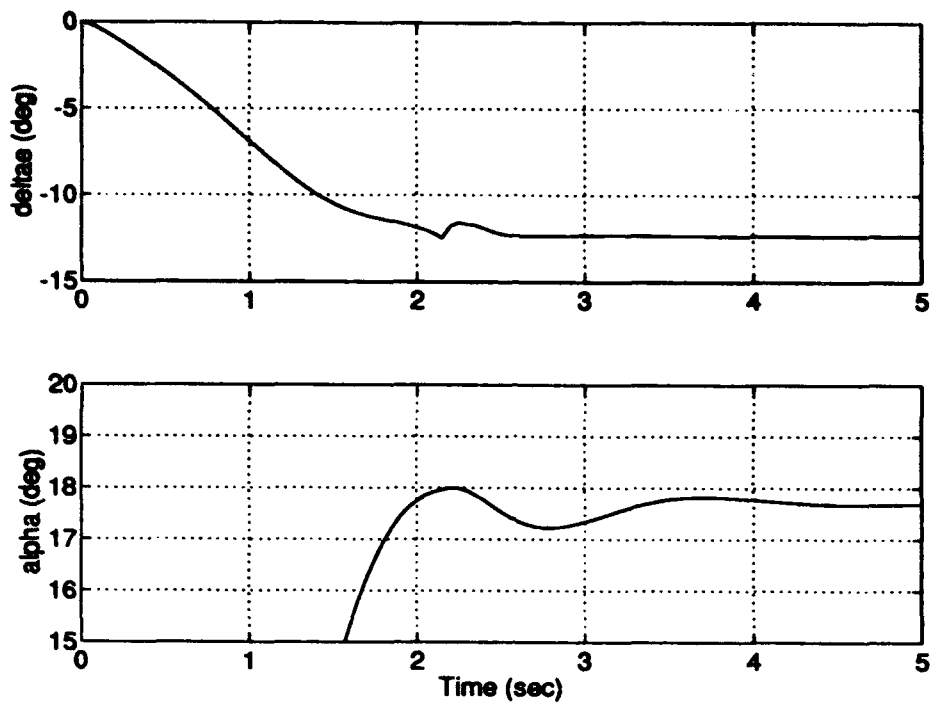


Figure 5.21. A-4D Short Period Elevator and AOA 4.6 g Responses

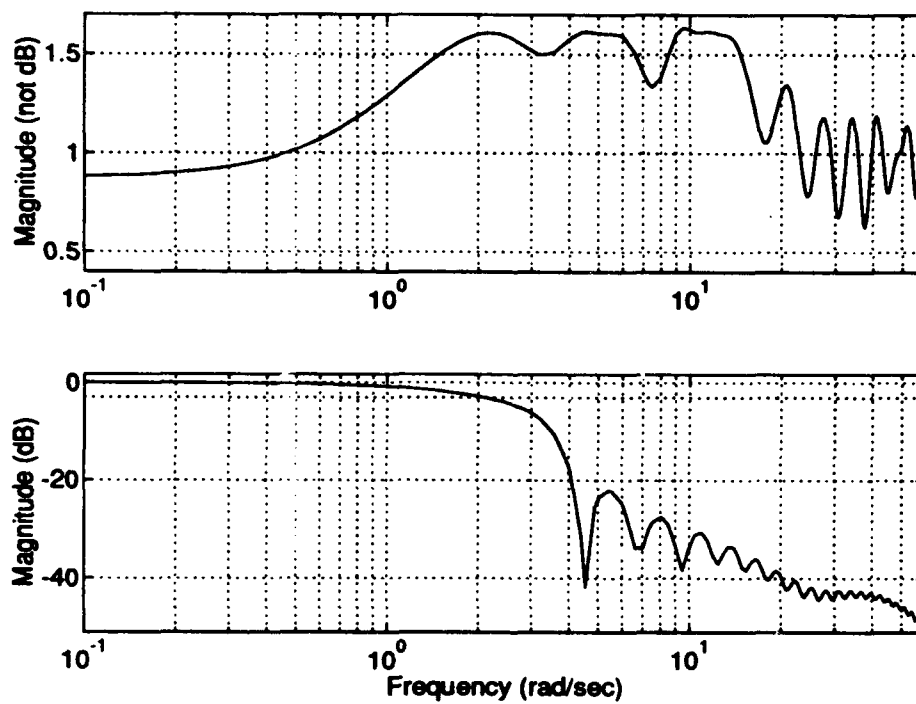


Figure 5.22. A-4D Short Period 4.6 g Input Sensitivity and CLBW

5.9.4 A-4D Short Period QDES Design, 4.7 g Response As was done for the full model A-4D case, the T_r and T_{s1} constraints were relaxed by 0.5 sec and a QDES solution was again sought. As no solution was found, the constraints were relaxed another 0.5 seconds with the same result. Finally, the $e_{cmd} = 0$ constraint was removed and a solution found which is shown in figures (5.23), (5.24), and (5.25). Notice how the N_z response hugs the lower 3 % error envelope and how QDES has reduced M_p to meet the constraint. The vector margins for this design are:

$$GM_u = -4.1 \text{ dB}$$

$$GM_1 = +8.2 \text{ dB}$$

$$PM = \pm 35 \text{ deg}$$

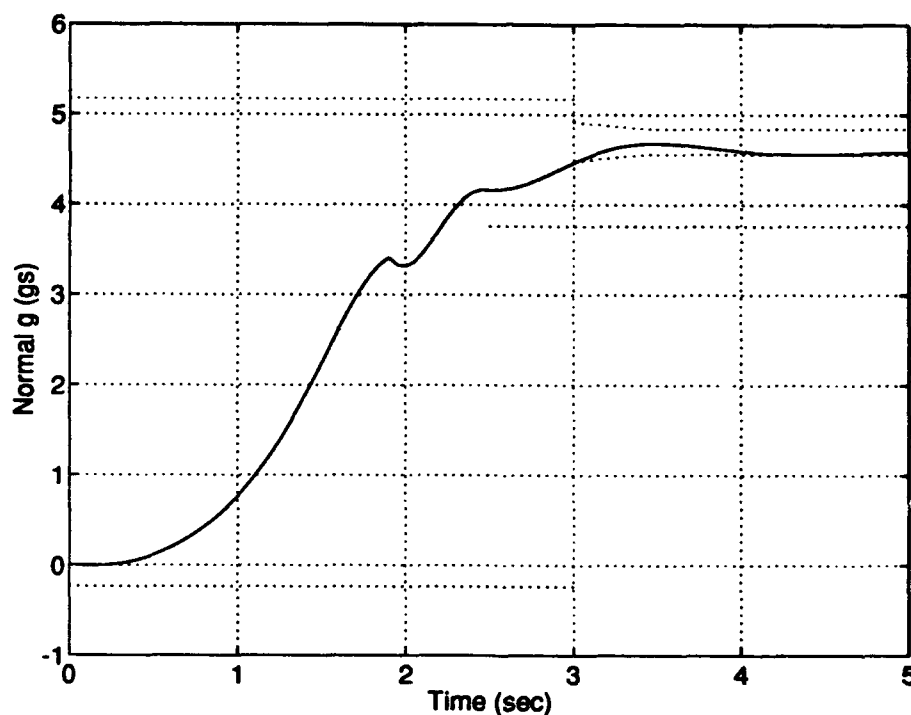


Figure 5.23. A-4D Short Period QDES Design, 4.7 g Response

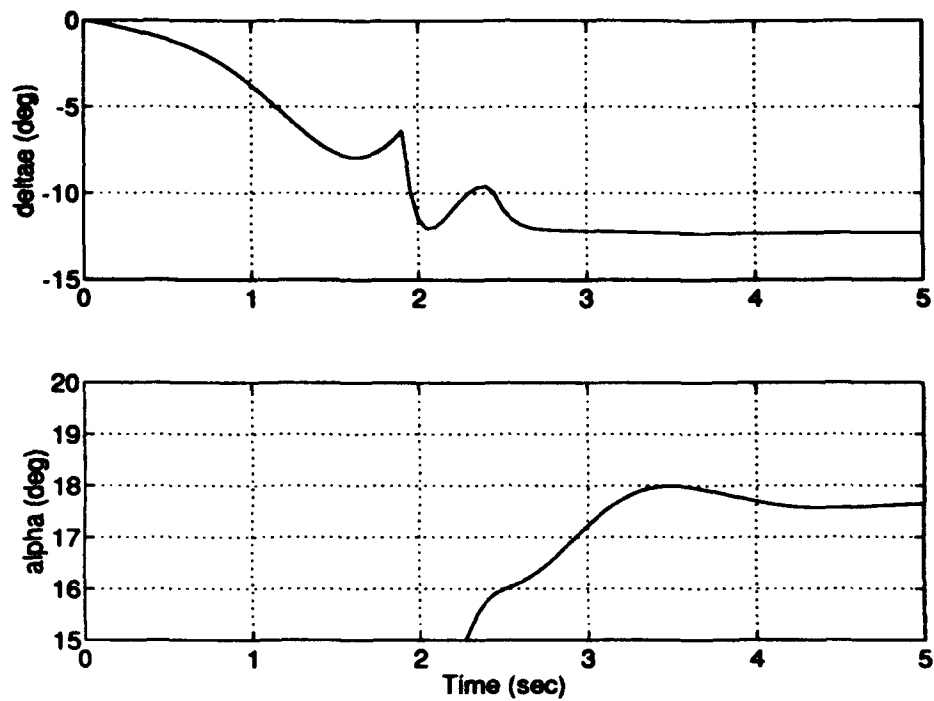


Figure 5.24. A-4D Short Period Elevator and AOA 4.7 g Responses

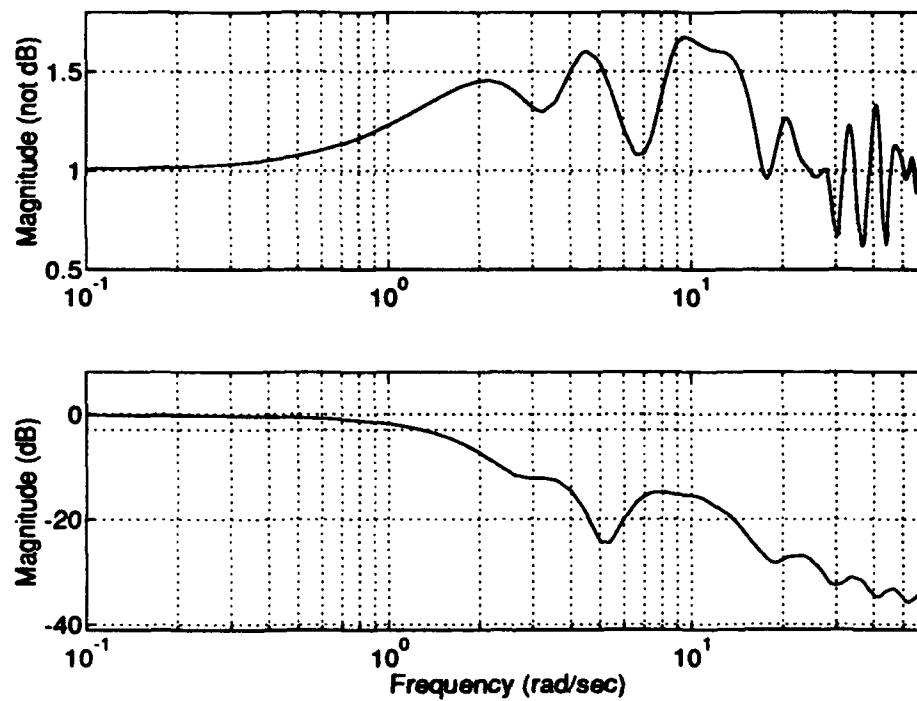


Figure 5.25. A-4D Short Period 4.7 g Input Sensitivity and CLBW

5.10 Summary

This chapter demonstrated the use of QDES in evaluating the limits of performance for N_z for the A-4D aircraft. It stepped through the process of model development and implementation with QDES through all the associated MATLAB script files needed to prepare the problem for QDES and examine its solution. A more detailed step-by-step procedure is provided in Chapter VII.

The results were fairly straightforward with a few interesting results. In the case of the full model, the time response, although not as smooth as in the classical design, performed as well in terms of the time domain measures of performance except for the command error. This appears to be caused by the Phugoid mode not being sufficiently controlled by QDES. This is also apparent in the lack of constraint of the sensitivity and thus the vector margins for the 1 g case. However, for the higher g runs, the full model had margins within the specified constraints. The Short Period approximation, having gotten rid of the Phugoid mode, performed quite well for all runs and even managed a slightly higher g level before hitting the α limit. Chapter VIII will examine the sensitivity result more closely.

For a relatively easy-to-control aircraft like the A-4D, the value of using QDES may not be quite overwhelmingly apparent. However, the ability to re-design the system to meet ever varying constraints for the evaluation of limits of performance lends quite a bit of flexibility to the design and evaluation process. Countless "what ifs" can be examined in a relatively easy manner compared to re-designing and tweaking a classically designed controller.

VI. QDES Design for the F-16

6.1 Introduction

This chapter explores using QDES to solve the N_z to N_{zcmd} problem for the F-16 as was done for the A-4D in Chapter V. Except where noted, the problem set up and constraints are the same as those used for the A-4D. The F-16 longitudinal model at sea level and Mach 0.6 has the following poles:

-20.0
-4.3542
+1.9055
-0.0086±.0708i

Note that the RHP poles makes this aircraft statically unstable and therefore much more difficult to control. It was of interest to see what QDES' capabilities were with this model as opposed to the A-4D which is statically stable. Appendix (B) contains the state space representation which is used in the problem set up as depicted in equations (5.1), (5.2), and (5.3).

6.2 Differences From the A-4D Problem

The highest frequency pole for the F-16 is the actuator at 20 rad/sec. Note this is twice as fast as that for the A-4D, therefore the minimum sample rate must also be twice as fast, or 16 Hz. When run for 5 sec this equates to twice the number of sample times which does not merely double the size of the problem. In fact, it doubles the number of *each* of the various constraints over their particular time span. In the A-4D example, the job sizes were approximately 6 megabytes. With the F-16 using 50 taps the job size was almost 85 megabytes, which caused the computations to slow to a virtual standstill. In

fact, above approximately 45 megabytes, the computer entered a mode whereby it had to wait for sufficient disk space to become available before performing computations. DeLaney recommended never allowing the problem to exceed 100 megabytes when not using any objective functions. However, to get reasonable run times (30 minutes or less) which would allow a designer to quickly evaluate his or her preliminary design, it was found the job size needed to be restricted to approximately 43 megabytes. To accomplish this, either the number of samples or the number of taps needed to be reduced. Through much trial and error, it was found that reducing the number of taps excessively precluded a solution where one was known to exist from previous trials. Thus, since the sample rate needed to remain at its minimum value of 16 Hz, the run time was reduced to reduce the number of samples. This resulted in the F-16 problems being run at 16 Hz for 3.5 sec with 45 FIR Taps. This resulted in job sizes of around 42 megabytes which for the F-16 full model ran for 5 to 15 minutes. Since this investigation is concerned with finding the maximum attainable N_z within the time response and margin constraints and not maximum sustained g, this change does not impact the value of the experiment.

The other difference from the A-4D example had to do with the constraint of the sensitivity. With the F-16, QDES would not return a solution if S was constrained below a value of 25. Thus all runs were made with this very liberal constraint in place.

6.3 F-16 QDES Design Results

The results from the F-16 design runs are presented in the same manner as for the A-4D. Significant differences will be noted as they are presented with a summary at the end of the chapter.

6.3.1 F-16 QDES Design, 1 g Response Figures (6.1), (6.2), and (6.3) show the results of the QDES design for a 1 g command in the same format as that used for the A-

4D. Note the N_z response is similar to the A-4D as are the δ_e and α responses, although the F-16 at this flight condition uses less elevator and shows less increase in AOA. Imposing the same constraints as on the A-4D, one expects to be able to attain a higher g level than the A-4D. In fact, it appears that the F-16 will also be α -limited, but at approximately 9.8 g's. Examination of the sensitivity plot shows the same behavior as the A-4D at 1g. There is a spike at the Phugoid frequency of 0.07 rad/sec which exceeds the constraint value even though QDES claimed to provide a solution that met all constraints. The rest of the plot above the Phugoid spike is constrained below 25 as requested. This spike at the Phugoid frequency results in predictably poor margins of:

$$GM_u = -0.24 \text{ dB}$$

$$GM_1 = +0.25 \text{ dB}$$

$$PM = 1.6 \text{ deg}$$

Note also the CLBW is approximately 3.5 rad/sec. For a sample rate of 16 Hz, the CLBW would need to be about 25 rad/sec before it would violate the guidelines discussed in Section (5.8).

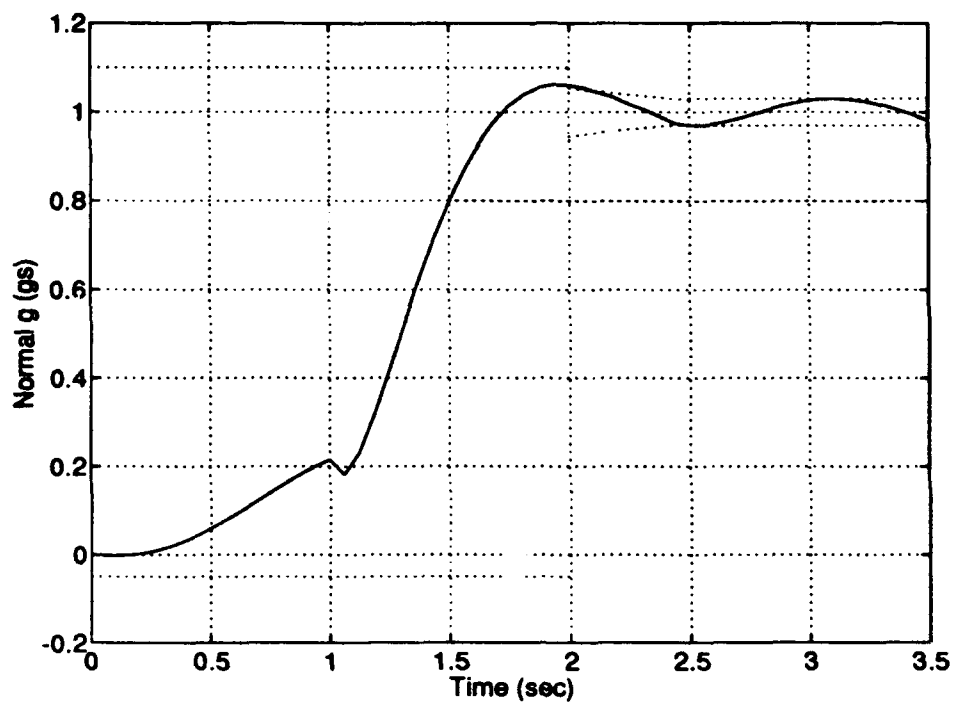


Figure 6.1. F-16 QDES Design, 1 g Response

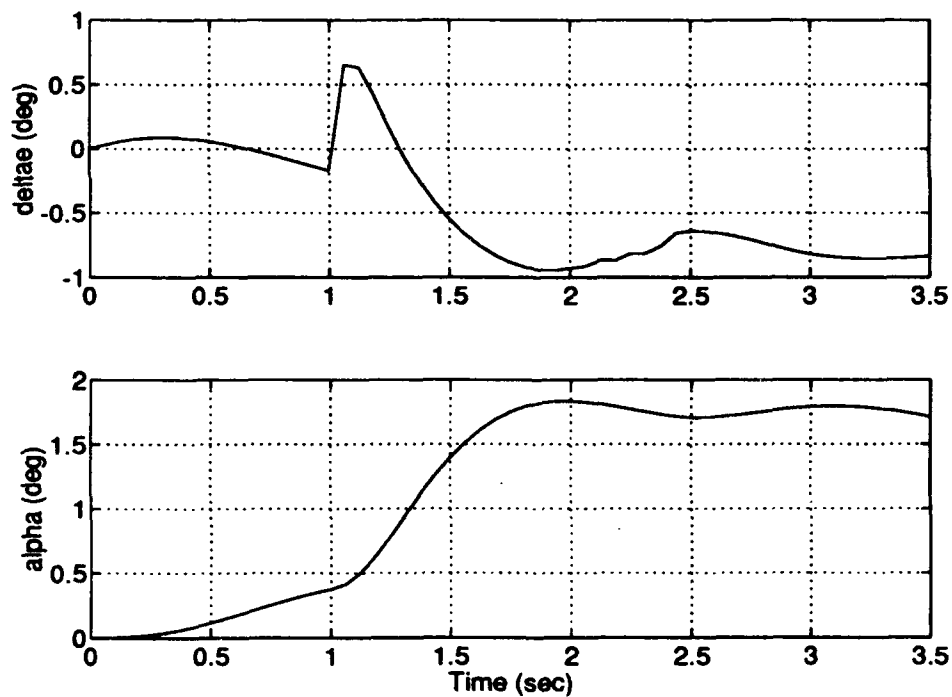


Figure 6.2. F-16 Elevator and AOA 1 g Responses

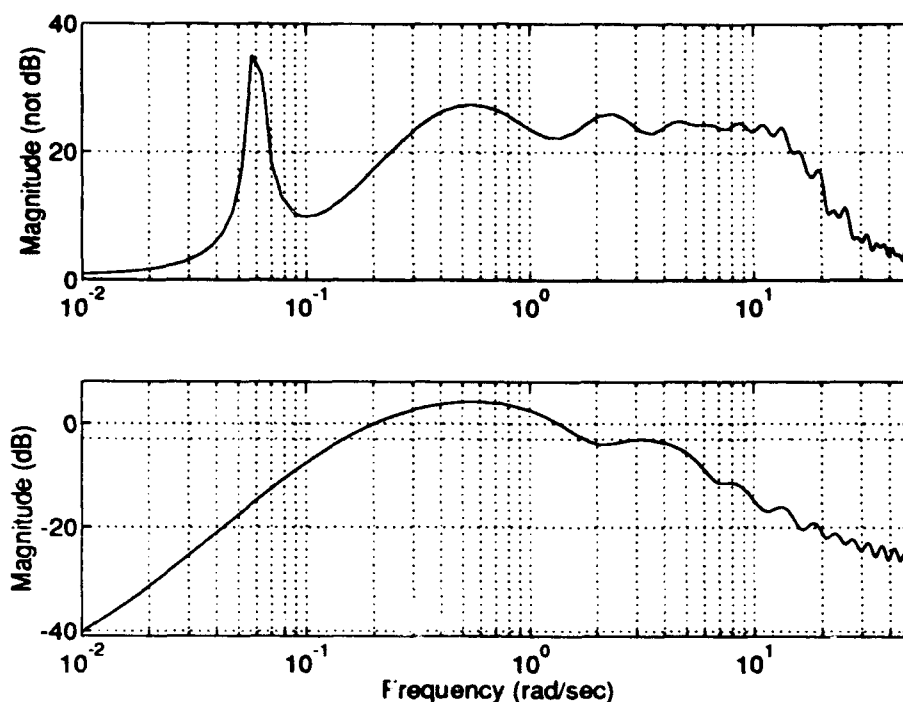


Figure 6.3. F-16 1 g Input Sensitivity and CLBW

6.3 *F-16 QDES Design, 9.8 g Response* Figures (6.4) and (6.5) show the results of the QDES design for a 9.8 g command. This value was chosen as it appeared to be near the maximum N_z . The plots of input sensitivity and CLBW are not shown for the rest of the F-16 examples as they do not change appreciably as the g level is increased. There is always a spike of approximately magnitude 45 at the Phugoid frequency, and the plot is constrained below 25 after that spike. The CLBW varies slightly between 1.5 and 4 rad/sec but never approaches the 25 rad/sec limit previously discussed. Notice that the AOA is at 18 deg, so there may not be potential for a higher g level. Also, although the elevator deflection is well within limits, the deflection rate needs to be examined. For the F-16, the maximum elevator deflection rate is 60 deg/sec. The "burble" in the N_z response at 1 sec is caused by displacement of the elevator in the wrong direction (positive) at a rate exceeding 100 deg/sec. If this rate were required to achieve the desired g level in the

nose up direction, this value would be of concern. However, since the aircraft is going the wrong way at this point, the designer does not *want* the elevator to be able to respond that quickly and so the fact that it cannot is not of concern. In the time period between 1 and 2 seconds where the elevator is deflecting to cause the desired level of performance, the deflection rate is approximately 29 deg/sec which is well within limits.

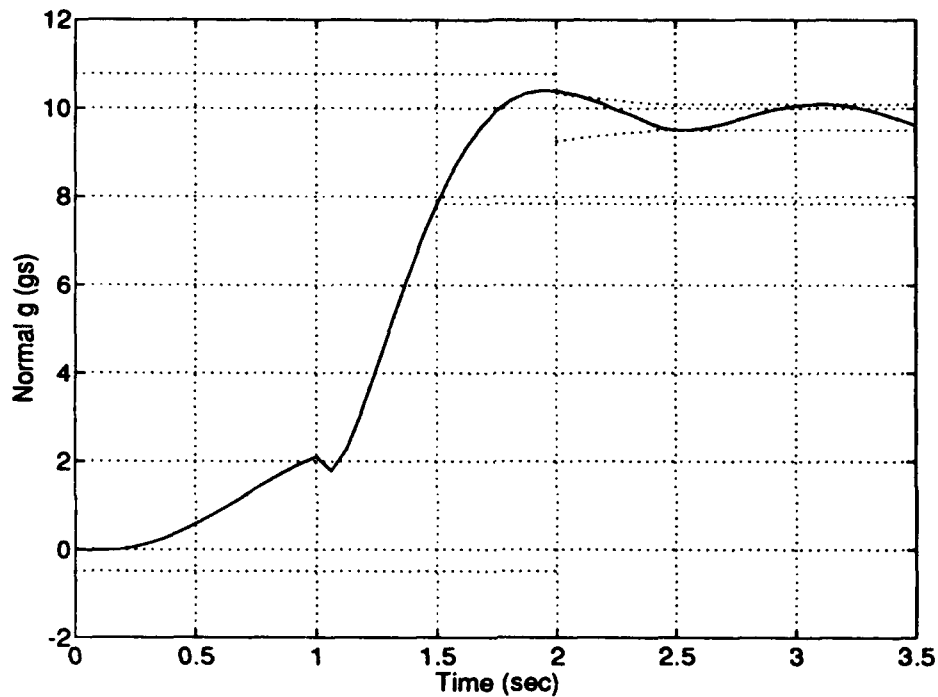


Figure 6.4. F-16 QDES Design, 9.8 g Response

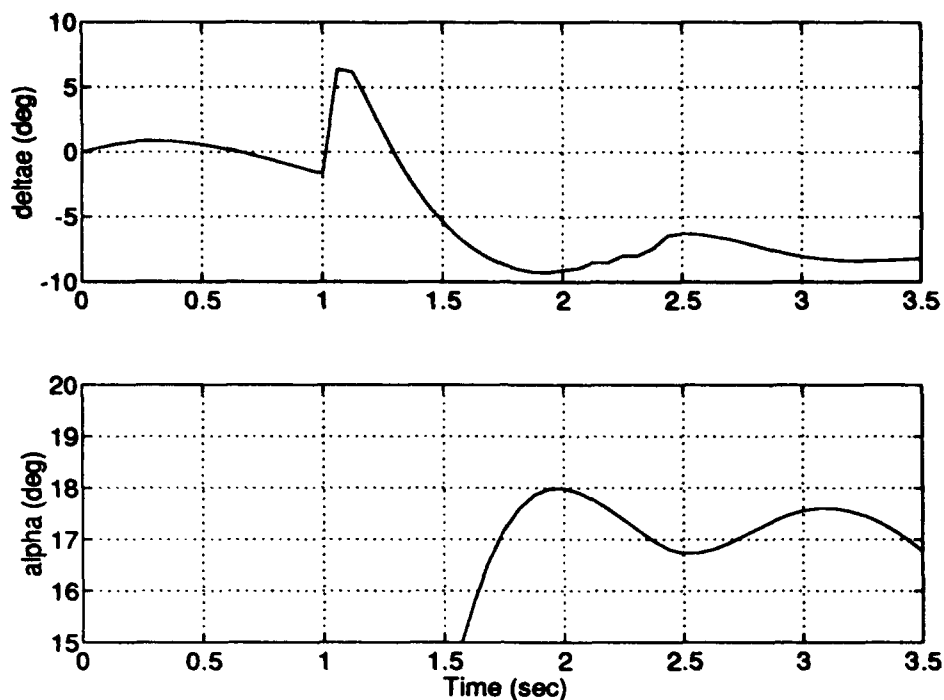


Figure 6.5. F-16 Elevator and AOA 9.8 g Responses

6.3.3 F-16 QDES Design, 10.0 and 10.2 g Responses This section compares the responses for the QDES designs for 10.0 and 10.2 g's. Comparing figures (6.6) and (6.7), notice how QDES has eliminated the overshoot from the 10.0 g response in order to keep the AOA within limits allowing 10.2 g's to be achieved. This is also seen when comparing the AOA responses in figures (6.8) and (6.9). Note that for 10.0 g's, the AOA limit is hit at 1.8 sec corresponding to when the N_z response hits its maximum overshoot. For the 10.2 g case, the 18 AOA limit occurs at about 3 sec when the response is within the 3 % e_{cmd} envelope.

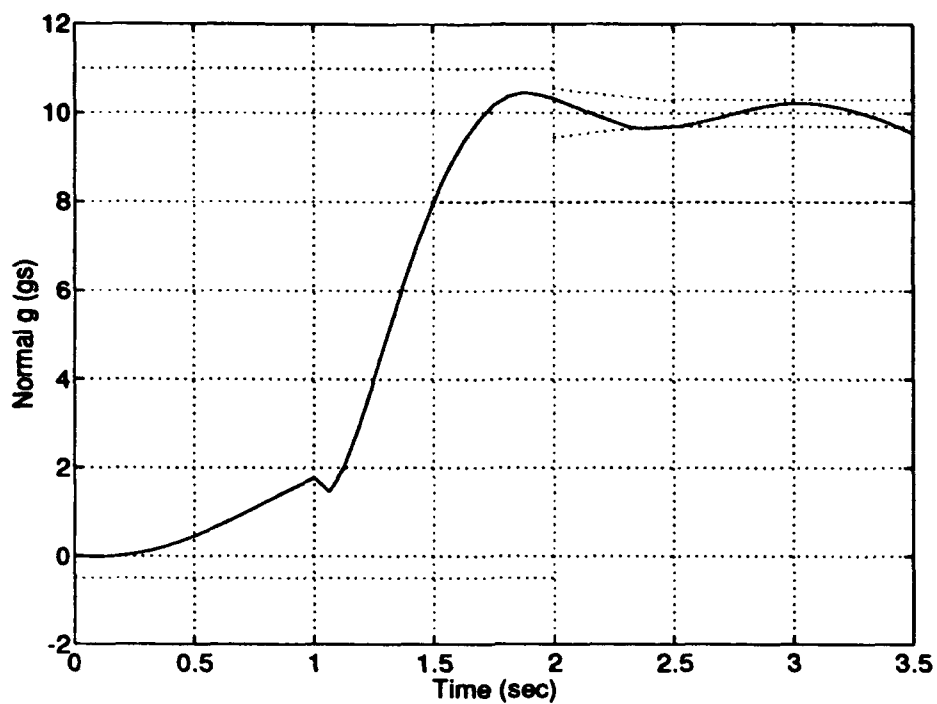


Figure 6.6. F-16 QDES Design, 10.0 g Response

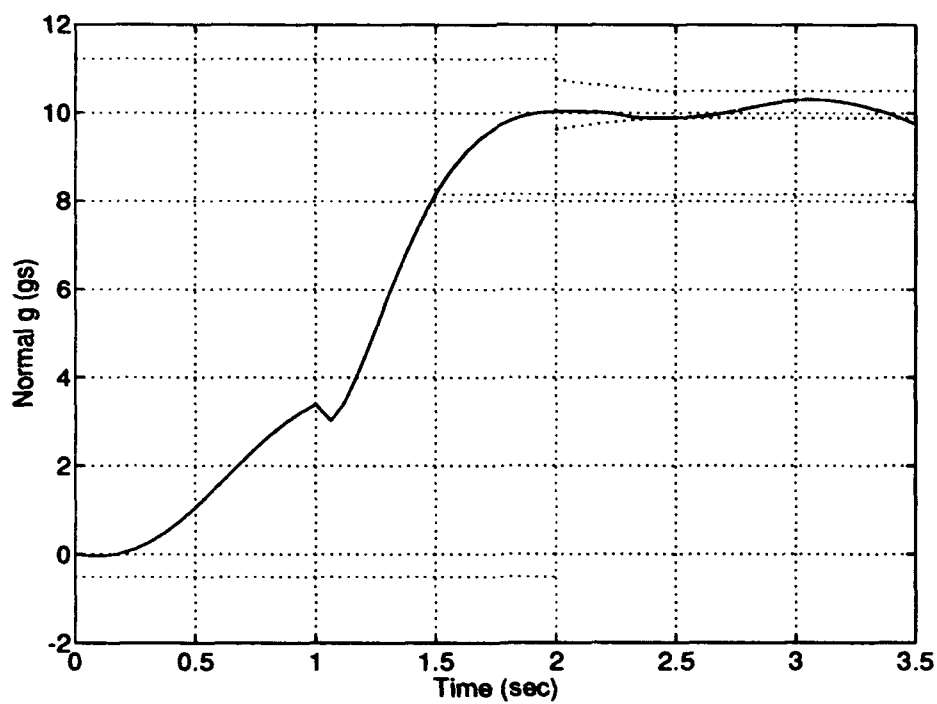


Figure 6.7. F-16 QDES Design, 10.2 g Response

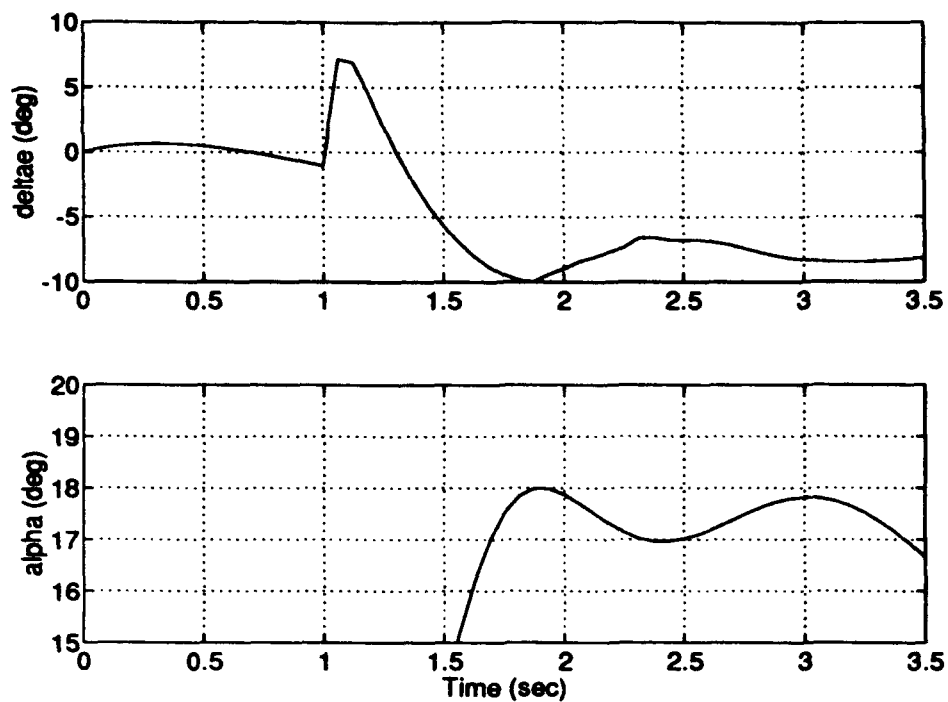


Figure 6.8. F-16 Elevator and AOA 10.0 g Responses

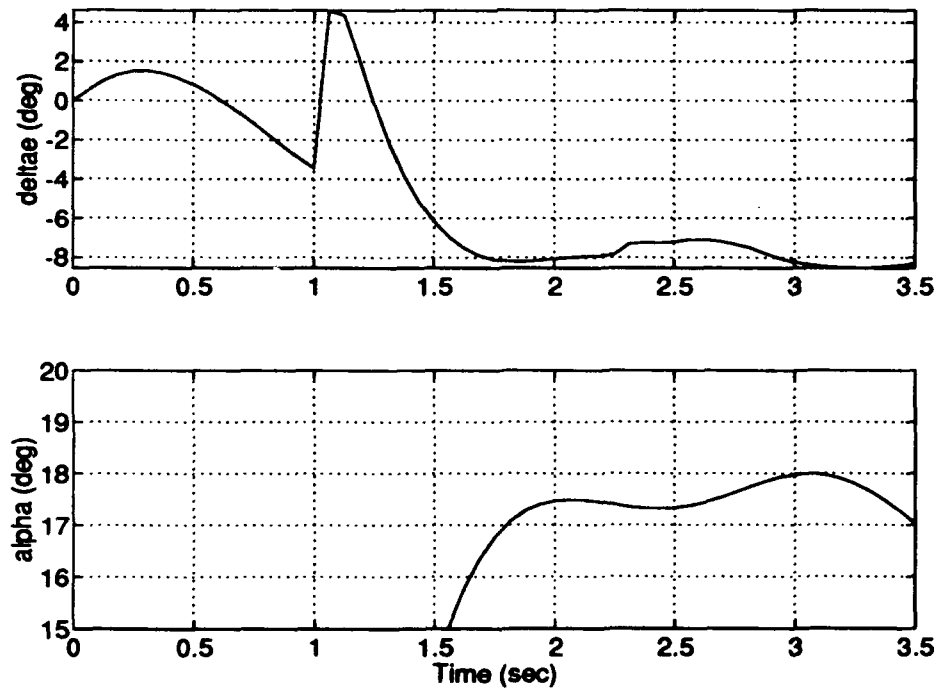


Figure 6.9. F-16 Elevator and AOA 10.2 g Responses

6.3.4 F-16 QDES Design, 10.4 g Response Finally, the limits of performance for the given constraints are reached at 10.4 g's. Notice how the N_z response has flattened out. Also, the α plot stays right at the 18 deg limit after reaching it. The maximum elevator deflection rate (excluding the "burble") is 32 deg/sec. Figures (6.10) and (6.11) show these results.

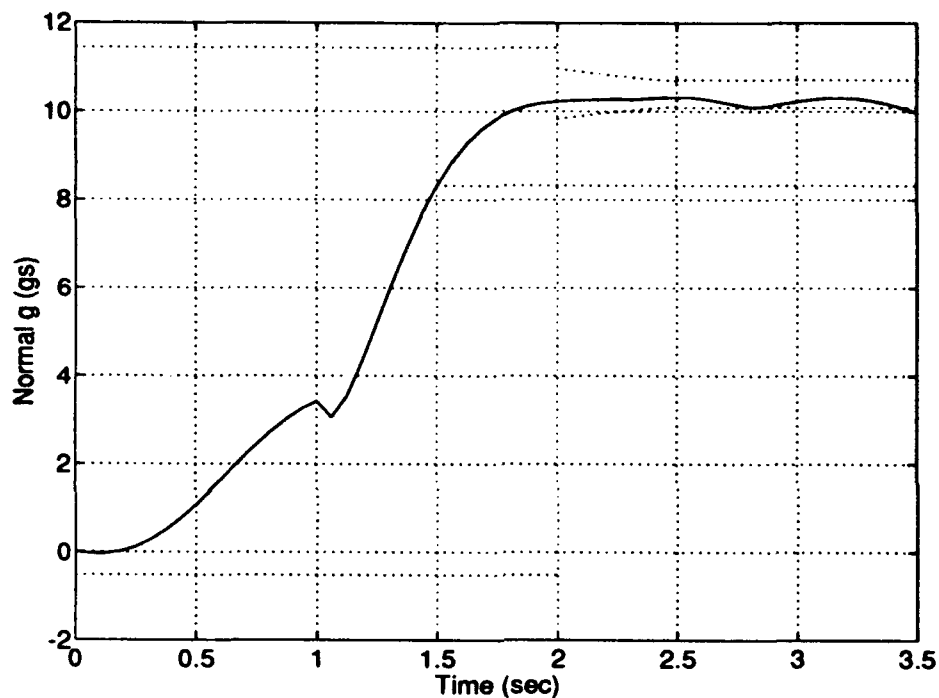


Figure 6.10. F-16 QDES Design, 10.4 g Response

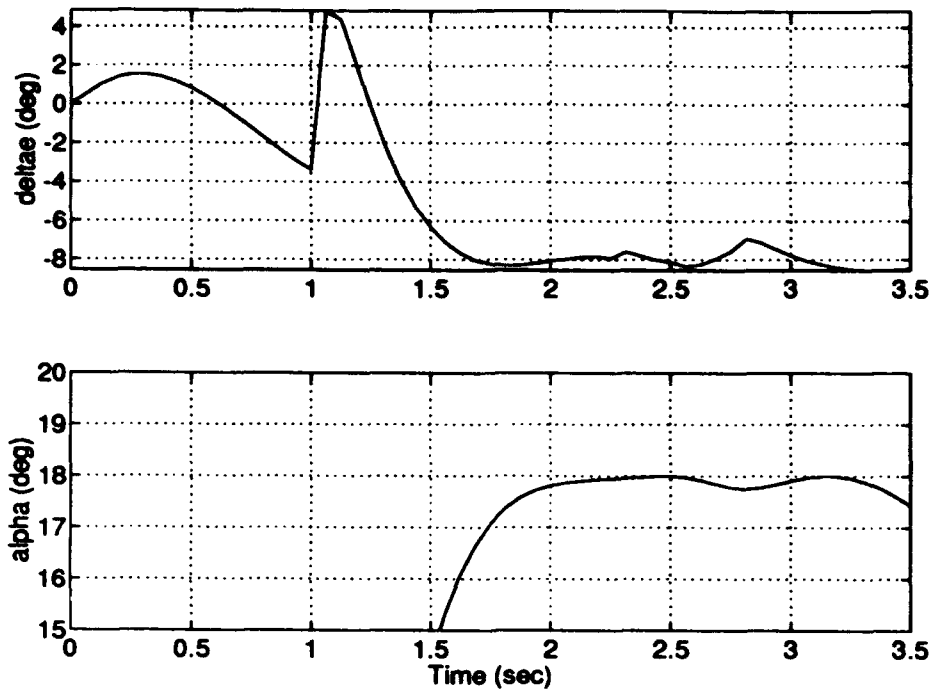


Figure 6.11. F-16 Elevator and AOA 10.4 g Responses

6.4 F-16 Short Period Approximation

As in the A-4D example, QDES was tried on the Short Period approximation for the F-16. Initially, only the forward velocity (v) state was removed as the θ column of the A matrix did not become all zeros. However, this left a low frequency RHP pole at .0015 which prevented S from being constrained below approximately 45. Therefore, both the v and θ state were removed leaving the following poles:

-20.00
 -4.350
 $+1.905$

However, this did not completely clear up the sensitivity constraint problem as it did with the A-4D. QDES was not able to return a solution with the sensitivity constrained below 20 which resulted in the following margins for all runs:

$$GM_u = -0.4 \text{ dB}$$

$$GM_1 = +0.4 \text{ dB}$$

$$PM = 2.8 \text{ deg}$$

Constraining e_{cmd} to zero was also tried as it was successful in the A-4D Short Period case. However, for the F-16, this constraint made the response more oscillatory than without it. Therefore it was not used.

6.4.1 F-16 Short Period QDES Design, 1 g Response The F-16 Short Period QDES design response to a 1 g command is shown in figures (6.12), (6.13), and (6.14). Notice the N_z response is smoother than the full model. The elevator initially deflects the wrong way, resulting in some nonminimum phase behavior. The sensitivity plot does not have a low frequency spike due to the removal of the Phugoid mode; however, it cannot be constrained below approximately 20. The closed loop bandwidth is approximately 1.6 rad/sec, again well below the 25 rad/sec limit discussed in Section (6.3.1). The sensitivity and CLBW plots are not shown for subsequent Short Period runs as they are virtually identical in each case.

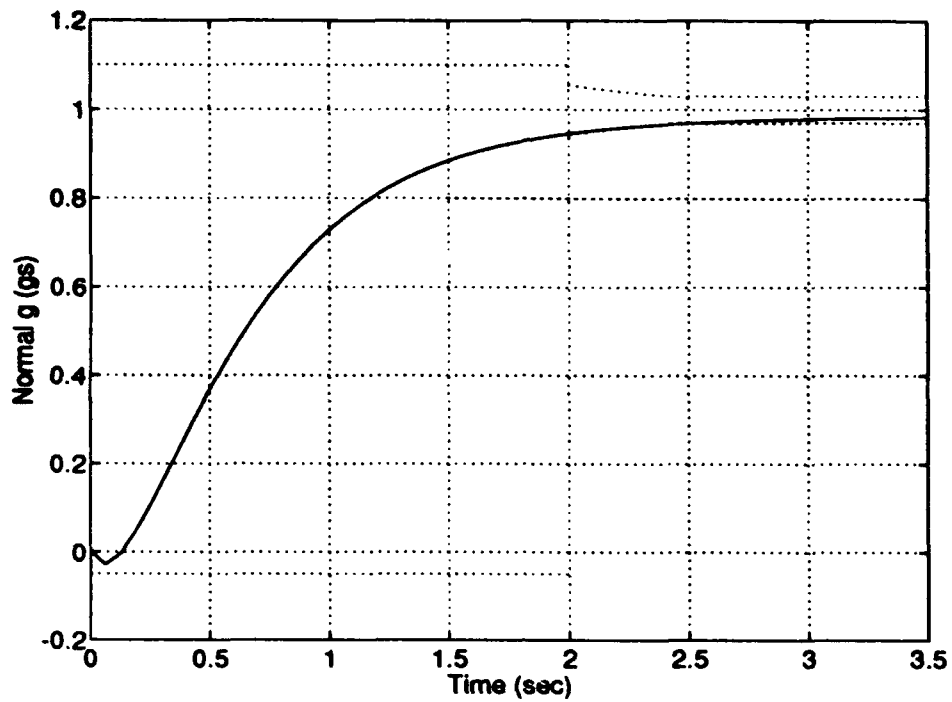


Figure 6.12. F-16 Short Period QDES Design, 1 g Response

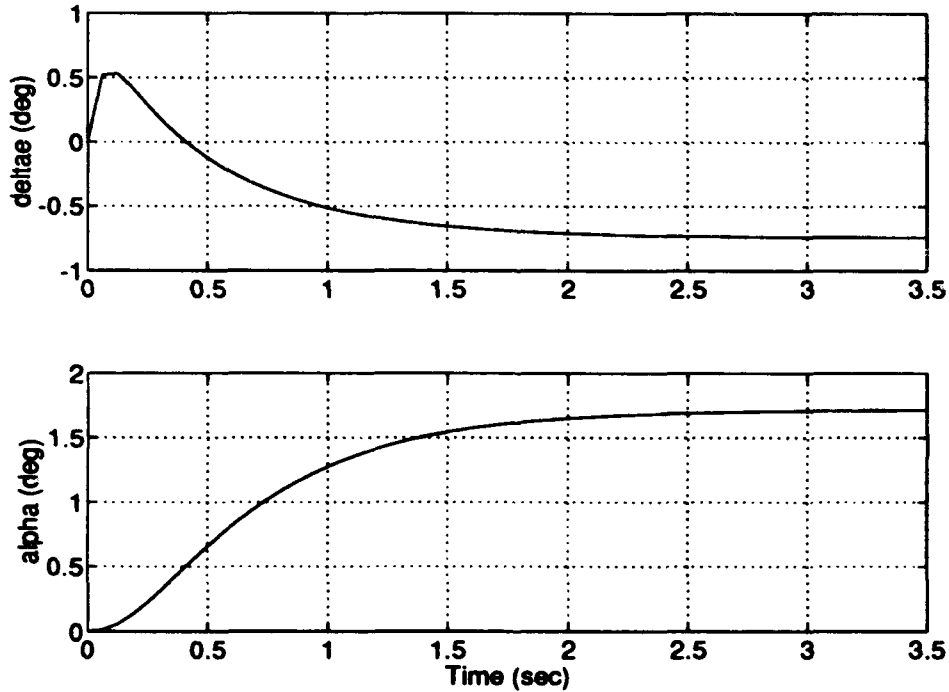


Figure 6.13. F-16 Short Period Elevator and AOA 1 g Responses

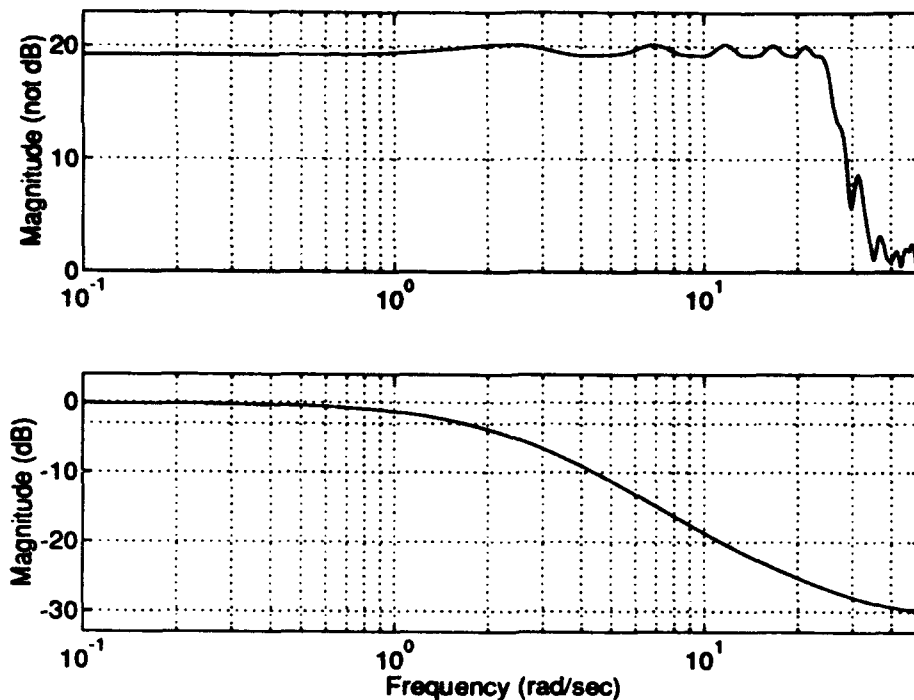


Figure 6.14. F-16 Short Period 1 g Input Sensitivity and CLBW

6.4.2 F-16 Short Period QDES Design, 10.4 g Response Since the F-16 full model maximum N_z was 10.4 g, the Short Period approximation was next attempted at this level. The responses are shown in figures (6.15) and (6.16). Notice that due to the smoothness of the response, the AOA has not yet hit the 18 deg limit, even though the aircraft is delivering a 10.4 g response. Also, the maximum elevator deflection rate (excluding the nonminimum phase behavior) is only 19 deg/sec contrasted with 32 deg/sec for the full model. The fact that the elevator does not have to compensate for a "burble" at approximately 1 sec, as did the full model, results in less control usage.

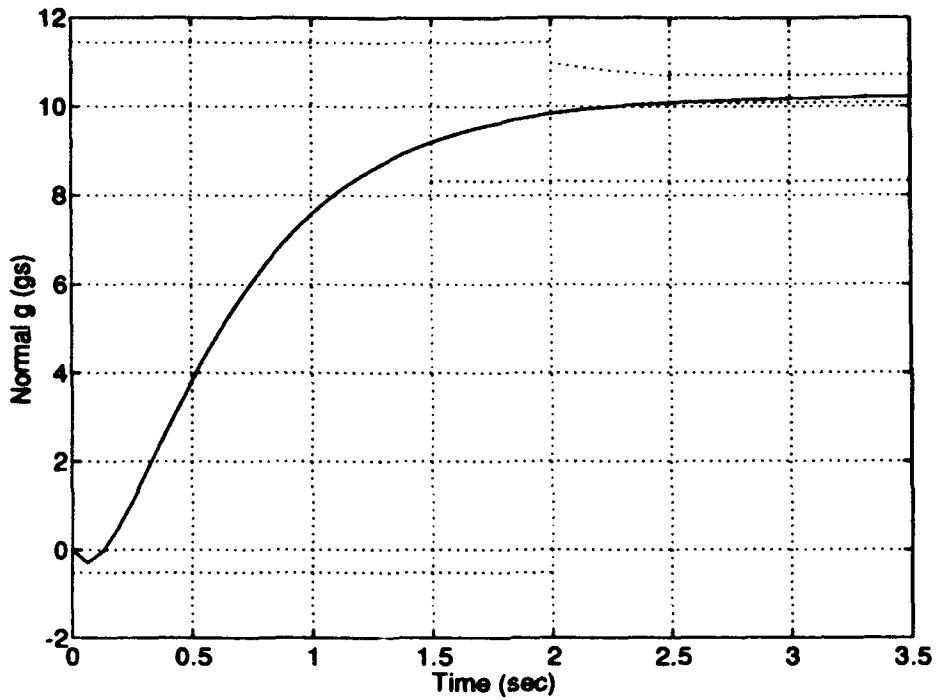


Figure 6.15. F-16 Short Period QDES Design, 10.4 g Response

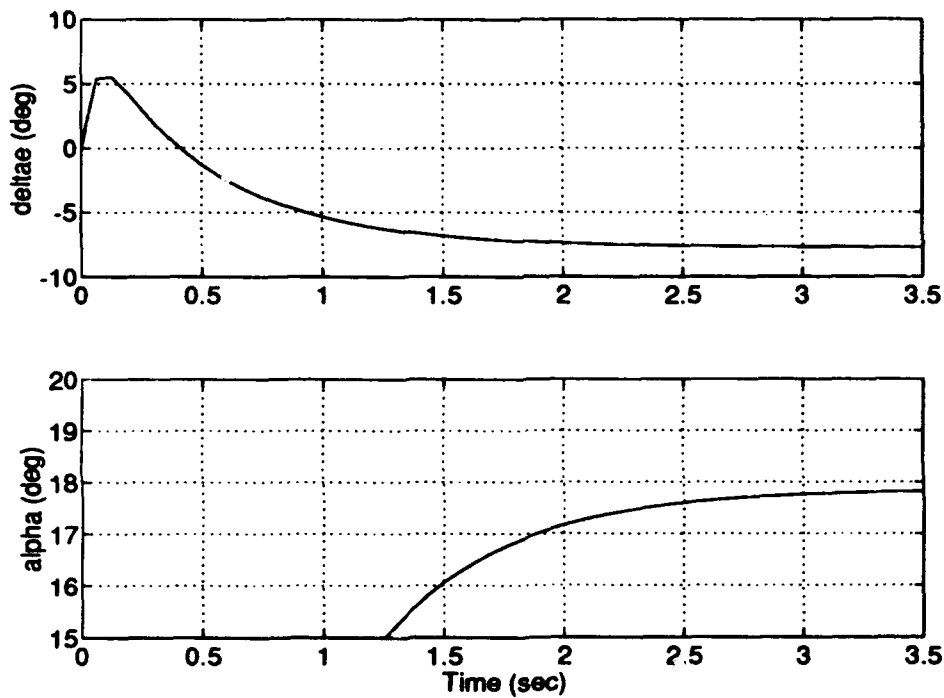


Figure 6.16. F-16 Short Period Elevator and AOA 10.4 g Responses

6.4.3 F-16 Short Period QDES Design, 10.6 g Response

The limit of performance for the F-16 Short Period QDES design was 10.6 g's. The responses are shown in figures (6.17) and (6.18). Notice that the N_z response hugs the lower 3 % e_{cmd} boundary, so this design may not actually be delivering a higher level of g than the full model's value of 10.4. From an operational standpoint, addition of 0.2 g's has little significance. However, from a strictly analytical point of view, given the *identical* constraints that *both* models were subjected to, this process determined that the Short Period was capable of a higher level of performance. Of significance is once again the lower elevator deflection rate (20 deg/sec) required by the Short Period design to achieve the same performance level.

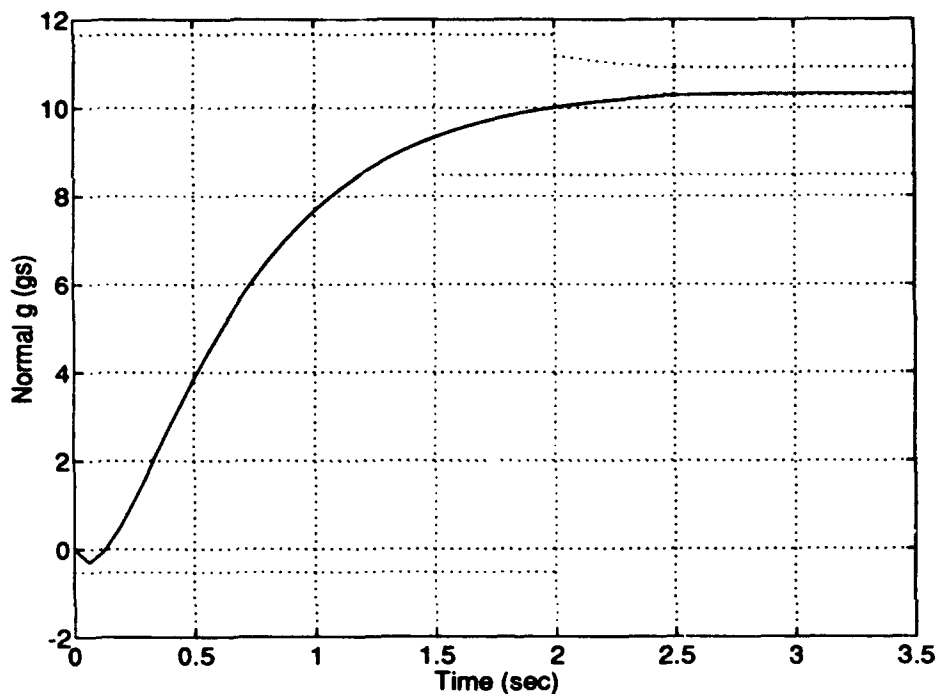


Figure 6.17. F-16 Short Period QDES Design, 10.6 g Response

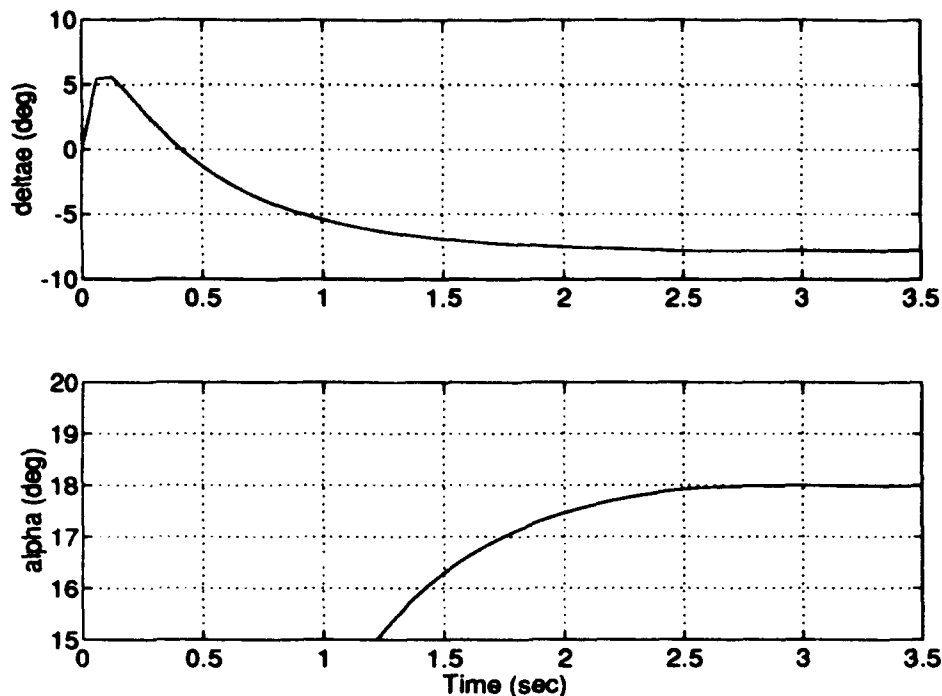


Figure 6.18. F-16 Short Period Elevator and AOA 10.6 g Responses

6.5 Summary

This chapter explored using QDES to determine the limits of performance for the F-16 as was done in Chapter V for the A-4D. This was of particular interest in that the F-16 is a much more difficult aircraft to control and does not lend itself so easily to a classical design as did the A-4D. The ability of QDES to quickly design a controller for varying performance demands proves its worth in a situation like this. However, the increased sampling rate required for the F-16 did increase the problem size and slow the process down considerably compared to the A-4D example.

The F-16 full model responses were generally smoother than the A-4D. This may be due to a higher number of FIR taps being used (45 vs. 35) as allowed by a higher number of sample times. As the limiting value of N_z was reached, it appears QDES was better

able to manipulate the time response of the F-16, in particular the overshoot, thus keeping AOA from hitting the limiting value of 18.

The Short Period approximation responses were generally smoother than their full model counterparts. Both achieved a slightly higher level of performance, due to the less oscillatory nature of their responses. With fewer transients in N_z , and thus the α responses, the average amplitude of the AOA response was higher when the upper limit of its oscillation touched up against the 18 deg limit. This allowed slightly more elevator deflection and consequently slightly more g.

The constraint of the sensitivity and thus control of the stability margins was not successful for the F-16. In the full model example, the spike in S at the Phugoid frequency remained through all the runs, unlike the A-4D example where it disappeared for any response over 1 g. The F-16 Short Period, although lacking the low frequency spike, could not be constrained below 20. This resulted in unacceptably low stability margins.

The author is aware that the F-16 is not intended to be flown at greater than 9 g's.

VII. Step-By-Step Procedure

7.1 Introduction

This chapter provides a detailed step-by-step procedure for using QDES to evaluate aircraft limits of performance as was done in Chapters V and VI. The intent is to provide sufficient detail and explanation to enable the non-controls engineer to do so. Although the explanation will use the evaluation of maximum N_z , this can easily be adapted to any of the longitudinal parameters. It could also be adapted to the lateral-directional problem if one is sufficiently familiar with that area. To avoid needless repetition, this chapter should be used in conjunction with this thesis and its appendices as well as the documentation provided with the QDES program.

7.2 QDES Installation and Test

Install QDES on the computer system per the instructions in the QDES user's guide supplied with the tape. It will be necessary to copy all the MATLAB script files to be used into the user's home directory or sub-directory, wherever the work is to be performed. This implies that the user must have MATLAB available. The files supplied with the tape require MATLAB version 2.00 or higher. The files in Appendix C were written for version 4.0. The author found it useful to make paper copies of all the README files supplied with the individual scripts and programs. The "Two Mass" design example should be run to check the correct operation of the system. Barratt [2] offers two options for this. The use of "qdes_set.m" *must* be used to check the ability of the system to generate the impulse response matrices. Primarily this will check if the file "mat2qdes" is installed in the proper path. If not, a message will be generated. This is explained in the README file in the "src" sub-directory. If the other "Two Mass" option

is chosen, "mat2qdes" will not be called and QDES will use impulse response matrices already generated and supplied with the program.

Once QDES' correct operation has been verified, the following MATLAB files must be copied into the directory where the work will be performed. Note there are many other script files supplied with the QDES program tape; however, for the process presented here, these are the only ones needed. For more information, see [8]. These files are: fir2ss.m, gethzw.m, loadq.m, midimpulse.m, obc2ttt.m, obc2ttt_fn.m, qobc2comp.m. It may be useful to make paper copies of these files for easy reference should a problem occur.

7.3 *Model Source*

The A, B, C, and D matrices in these examples are derived from the linearized equations of motion for the aircraft at a given equilibrium flight condition. These matrices can be formed from the stability derivatives. Formulas for the estimation of these derivatives and construction of the matrices can be found in Etkin [9] and other flight dynamics textbooks. It is critical that the units and the order of the states are known.

7.4 *Using the "nzprp.m" File*

As briefly described in Chapter V, a MATLAB file similar to the "nzprp.m" shown in Appendix (C), Section (C.1) should be written. This section will walk through this file in detail, explaining how the various entries are arrived at and the rationale behind them based on the author's and others experiences using QDES.

7.4.1 *Entering Plant Data* The plant A, B, C, and D matrices are entered. To avoid confusion, it is recommended that the order of the states match those used in this

thesis. Next, the actuator dynamics are augmented. Appendix (A), Section (A.1.1) and (A.1.2) show how the matrices change as the state space representation of the actuator is augmented to the system. The script file does this automatically. Normally, the state-variable model matrices are constructed such that N_z is positive nose downward. This is easily changed to the positive nose up convention used in this thesis by multiplying the "nz" (last) row of the C matrix by -1 . If the units are not as desired, they can be easily converted. Form the diagonal T matrix to scale the states. The number on the diagonal is what must be multiplied to the state to convert from the present units to the desired units. In this example, the original states are in non-dimensional velocity and radians. To convert to ft/sec and deg, form the matrix:

$$T = \begin{bmatrix} 634 & 0 & 0 & 0 & 0 \\ 0 & 57.3 & 0 & 0 & 0 \\ 0 & 0 & 57.3 & 0 & 0 \\ 0 & 0 & 0 & 57.3 & 0 \\ 0 & 0 & 0 & 0 & 57.3 \end{bmatrix}$$

where the equilibrium velocity (V_0) is 634 ft/sec and 1 rad is 57.3 deg. To convert the input (δ_{cmd}) from rad to deg, form:

$$S = [57.3]$$

Finally, to convert the outputs from non-dimensional velocity, rad, and ft/sec, form:

$$W = \begin{bmatrix} 634 & 0 & 0 & 0 & 0 \\ 0 & 57.3 & 0 & 0 & 0 \\ 0 & 0 & 57.3 & 0 & 0 \\ 0 & 0 & 0 & 57.3 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{32.2} \end{bmatrix}$$

where 1g is 32.2 ft/sec. Once these matrices are formed, the script file multiplies them to the original plant matrices as shown in the file to convert them to the new units.

As mentioned in Chapter V, the scale factor is entered to tell QDES what a "unit step" input really is.

7.4.2 Discretizing the Plant Chapter V provided some guidelines to selection of the sample rate (Srate in the file) to be entered. The author recommends using the discretization rate based on five times the frequency of the fastest system pole (usually the actuator). Use of a lower rate risks not capturing all the plant dynamics. In some cases, use of a higher rate may result in a smoother response as in the case of the A-4D Short Period example, so there is room for experimentation. The author found that increasing sampling rate tended to raise the minimum value below which the sensitivity could be constrained. Also, increasing the sampling rate greatly increases the size of the problem with its attendant slowing of the process as was experienced with the F-16 in Chapter VI.

7.4.3 Forming the P-K Style Matrices The formation of the nine matrices needed by "obc2ttt.m" is straightforward as developed in Chapter V. These can be used exactly as depicted in the file *unless* it is desired to change the number or order of the various inputs or outputs (w, z, y, u). If so, a diagram such as figure (5.1) must be drawn and the matrices constructed.

7.4.4 Forming an Observer Based Controller As explained in Chapter V, QDES must be given a stable nominal plant with a nominal controller. In this process, this is accomplished by forming an Observer Based Controller using the MATLAB "dlqr" and "dlqe" commands to generate the state feedback gain matrix (K) and the Kalman filter gain matrix (L). It is recommended that the weights listed Chapter V and as entered in the example file be used. The author experimented with other weights and with using the

MATLAB "place" command as done in the "Two Mass" example. Both cases produced worse results. In theory, as long as the nominal system is stable, it shouldn't matter to QDES what it is. The resulting solution should be the same. This was certainly not the case.

7.4.5 Calling "obc2ttt.m" Two other pieces of information need to be entered before calling "obc2ttt.m" to form the impulse response matrices. The first is the source file name. This name is used to tie the QDES source file, the impulse response matrices, and the output files together. Whatever is entered here will be tacked on to several files associated with the problem being addressed by the QDES source file. The other piece of information is the number of samples (*n_sample*). This, along with the sample rate, determines over what time span QDES will search for a solution. It is the sample rate in Hz times the time over which one wishes QDES to work. As noted in Chapter VI, the number of samples relates directly to the size of the problem in terms of computer memory. Therefore, this number should be kept as small as possible. With a fixed sample rate, this means the time span should be kept as small as possible. As discussed in Section (5.4), for the N_z problem there is little point in running the problem for a long period of time. Three to five seconds should be reasonable in most cases, unless the aircraft has an *unusually* slow response. Assuming there are no errors, "obc2ttt.m" writes the following files to the current directory: A4D7NZ_t1, A4D7NZ_t2, A4D7NZ_t3, and A4D7NZ_ttt.mat Note these use the source file name from the example script. The first three are the impulse response matrices, and the last contains the plant information and nominal controller for use by QDES and for forming the closed loop plant once QDES arrives at a solution. The "nzprp.m", as written, also displays the following message on the screen:

When writing your Qdes source file, be sure that

`n_sample=56`

so that it matches with the impulse response matrices you just created

For this set up your sample rate was: 16 Hz Sample time was: 0.0625 sec

Also, be sure that your "nzpst.m" file uses the Scale Factor

`Sfnz= 1.0`

7.5 Writing the QDES Source File

This section walks through the writing of the QDES source file. The purpose and use of this file was discussed in Chapter V. This section will detail the mechanics of putting it together. Appendix (C), Section (C.2) shows the source file used for the A-4D example and will be referenced. This file, written in "C", must be given the same name as that entered under "source =" in the "nzprp.m" script file.

7.5.1 The Preamble In the preamble, the exogenous inputs (w), regulated outputs (z), and sensed outputs (y) are defined and ordered or numbered. This order *must* match that used when constructing the matrices in "nzprp.m". When writing the constraints, these signals may be referenced either by their name as defined or their number. In this example, numbers were used exclusively.

7.5.2 The Declarations Section In the declarations, the number of each of these signals listed in the preamble must be listed, as well as the number of actuators. The number of samples from the "nzprp.m" file is entered as well as the number of FIR taps to be used. As mentioned in Chapter V, the number of taps is constrained. The author found a good rule of thumb was to make the number of taps five less than the number of samples unless this would cause the number of taps times number of actuators times number of sensed outputs to exceed 200, or if it made the problem too large in terms of computer memory. The rationale for using as many taps as possible is that the goal is to see if a

controller can be built to meet the constraints. If more taps are used, QDES has more information with which to work and can better answer that question. In most cases, using a higher number of taps also resulted in a smoother response. Finally, the impulse response matrices must be defined by name. The example source file can be used as a model and should be followed exactly to ensure proper syntax.

7.5.3 The Constraints Section This is the most difficult section to write. The example source file can be used as a model unless some other form of constraint is desired. If so, [4] should be consulted. The first constraint shown is the input margin. In English, this statement says that the user desires the magnitude over all frequency of the output [5] (δ_{cmd}) to the input [2] (U_{in}) to be less than or equal to 1.6628 which corresponds to the value of the sensitivity necessary to give a vector phase margin of 35 deg. The values corresponding to the desired vector margins can be calculated from the following formulas:

$$\|S\|_{\infty} = \frac{1}{1 - 10^{-(\text{DUGM} \times 0.05)}} \quad (7.1)$$

$$\|S\|_{\infty} = \frac{1}{10^{-(\text{DLGM} \times 0.05)} - 1} \quad (7.2)$$

$$\|S\|_{\infty} = \frac{1}{2 \sin(.5 \times \text{DPM})} \quad (7.3)$$

where:

DUGM = Desired Upper Gain Margin (dB)

DLGM = Desired Lower Gain Margin (dB)

DPM = Desired Phase Margin (deg)

Note that only one value is entered (the lowest) although there are three separate constraints. Thus, whatever constraint gives the lowest value drives the $\|S\|_{\infty}$. If achieved, the other two margins will be better than desired. On the other hand, if QDES misses on constraining $\|S\|_{\infty}$, this may not effect two of the margins, depending on how

much it missed by. When using this constraint function, QDES warns the user that a 0.34 dB error may result. This equates to a factor of 1.0399. Thus, if a certain value of $|S|_{\infty}$ is absolutely required, this value should be multiplied by 1.0399 and the difference between this and the desired value should be subtracted from the desired value and entered in the constraint. This assumes QDES will error on the high side which would give lower margins than desired if not compensated for. Constraining the maximum elevator deflection and angle of attack are done via the "u_step[][](t)" function. Again, the output is listed first, then the input. Note that the time (t) is discrete time steps, not actual clock time. The elevator constraint in the source file says that for a unit step input to [1] (N_{zcmd}) the output [4] (δ_e) should stay within the values listed. The same idea holds for the angle of attack constraints. Note that α_{trim} needs to be taken into account when setting the numbers in the constraint as noted in Section (5.5.4). The overshoot and undershoot constraints are fairly straightforward. To draw the remaining time domain constraint curves and lines shown in response plots in Chapters V and VI, various exponential relationships are required. To accomplish this, a MATLAB script file "visnz.m" was written. This file is found in Appendix (C), Section (C.4). By following the comments, one can enter the desired time domain constraints and the program will figure all the exponential relationships and draw a sample envelope for examination. By varying the inputs, one can tailor the envelope to virtually any shape desired. Most importantly, the script file also writes out the constraints as they should appear in the QDES source file. An example of the output from "visnz.m" for the A-4D runs is shown.

This is how your Qdes Source File constraint statements should look to draw the equivalent constraint envelope for Qdes and constrain the vector margins of your selected SISO transfer function.

```
***** FOR NORMAL ACCELERATION (Nz) *****
```

```

/* Put Constraints On "      " Vector Margins */

for w=0.0 to %pi step 0.02 do {
mag_H[out][in](1,w)<=1.663;
}

/* Put Constraints On Overshoot */

overshoot[out][in]<= 0.1;

/* Put Constrains On Undershoot */

undershoot[out][in]<=0.05;

/* Put Constraints On Settling Time */

for t=16 to 20 do
1-0.8392^t<=u_step[out][in](t)<=1+0.8392^t;

/* Request Zero Command Error*/

Re_H[out][in](1,0) = 1;

      AND/OR

/* Put Constraints On Command Error */

for t=20 to n_sample-1 do
0.97+0*(t-20)<=u_step[out][in](t)<=1.03-0*(t-20);

/* Put Constraint On Rise Time */

for t=12 to n_sample-1 do
0.8<=u_step[out][in](t);

/* Put In Upper Constraint Curve (if desired) */

for t=0 to 4.8 do
u_step[out][in](t)<=(0.1596*t)^3+0.2;

for t=4.8 to 12 do
u_step[out][in](t)<=0.45-(0.7663-0.1064*(t-4.8))^3+0.65;

/* Put In Lower Constraint Curve (if desired) */

for t=4 to 7.2 do
(0.2303*(t-4))^3<=u_step[out][in](t);

for t=7.2 to 12 do
0.4-(0.7368-0.1535*(t-7.2))^3+0.4<=u_step[out][in](t);

```

In a windows environment, it is then a simple matter of cutting and pasting the appropriate constraint to the QDES source file. The only other thing that must be done is to insert the name or number of the output and input of interest in the brackets where it currently says [out] [in] and double check the syntax for proper brackets as shown in the example source file.

7.6 *Running QDES*

To run QDES, type "qdes A4D7NZ" on the command line and hit return. If QDES finds any syntax or other errors it will stop and provide a message. These are fairly clear, identifying what line of the source file contains the error. Other times it may identify a mismatch between the number of samples entered in the source file and the number actually in the impulse response matrices. Again, these are fairly obvious. If there are no errors, QDES returns the following:

```
qdes version 1.5 informal distribution
copyright 1987 Stanford University
got all coefficients
Warning: Reference to Mag_H may incur +/-0.34 dB error.
329 control constraints
0 objective terms
generating feasibility program
```

Calls to LSOPTN

```
-----
print level = 10
feasibility tolerance = 1e-06
crash tolerance = 0.0001
infinite bound size = 1e+10
print level = 0
solving feasibility program
problem type FP
```

If QDES finds a solution, the following will be returned:

```
found feasible point in 62 iterations
writing F162NZ.q F162NZ.out
```

If no solution is found QDES returns the following:

```
no feasible point found in 131 iterations
sum of infeasibilities = 0.0459192
writing F162NZ.last.q F162NZ.out
```

When the sum of the infeasibilities is quite small (less than 0.1), adding a few more taps (if possible) may result in a solution. The ".out" file may be examined to see what constraints QDES could not meet. For multiple runs, if the sample rate, number of samples, or scale factor have not changed, "nzprp.m" does not have to be run again. Constraints in QDES can be changed and more runs attempted. If the sample rate changes however, "visnz.m" will have to be run with the new value as all the calculations for the constraint curves are based on this. When first starting a QDES problem, it is recommended that the constraints be added in progressively. Otherwise, if no solution is found, it may be difficult to ascertain which constraint is causing the difficulty. The e_{cmd} constraint should be entered first to get QDES to form a step response. The other time constraints can then be added starting from the end of the response (T_{sl}) and working forward (T_r). The margin constraint should be entered after all the time constraints have been explored as it tends to lengthen the run times. Elevator and AOA constraints should be added only when the limits of performance are to be explored, as they greatly increase the problem size and slow the computations. Use of the zero command error constraint should be reserved until all other objectives are met. It may improve the response as it did for the A-4D Short Period, but more often than not, it may result in no solution being returned.

7.7 *Outputting the Results*

The MATLAB script file "nzpst.m" in Appendix (C), Section (C.3), takes the QDES solution, forms the closed loop system, and outputs the information for each run as seen in Chapters V and VI. The QDES source file name must be entered to enable

"qobc2comp.m" to form the QDES controller and "gethzw.m" to form the closed loop system. The sample rate, run time (sec), and scale factor are entered and, of course, must match those used to set up the problem. The data from "visnz.m" should also be entered to enable the file to draw the same constraint envelope as that imposed on QDES. In the section of the file where the N_z time response is plotted, each plot command for the various time constraint curves is on a separate line and labeled. These may be individually commented out as desired (using %) to plot only those constraints which were actually imposed on QDES. To evaluate elevator deflection rates, another MATLAB script file "surf.m" was written and is found in Appendix (C), Section (C.5). The file contains simple instructions for its use. When used, it returns the surface deflection rate over the time frame designated.

7.8 *Summary*

This chapter developed a simplified, step-by-step approach to using QDES to evaluate aircraft N_z limits of performance as accomplished in Chapters V and VI. Although not completely self-contained, when used with the body of this thesis and its appendices, it provides most of the nuts-and-bolts information needed by a non-controls type engineer to perform this type of analysis.

VIII. Conclusions and Recommendations

8.1 Introduction

This chapter presents an evaluation of using QDES to explore the limits of performance in aircraft design as was accomplished in this thesis. It also offers possible explanations as to why some aspects of the process did not work as well as anticipated as well as other approaches tried. Finally, it offers a suggestion for future research in this area.

8.2 Usefulness of QDES

QDES' method of operation is quite ingenious. The ability to enter a number of constraints and then let the computer see if it is possible to build a controller to meet them lends itself perfectly to the limits of performance problem. Evaluation of QDES' usefulness for this problem can be divided into two areas, time response and frequency response.

8.2.1 Time Response Examining the time response portion of the problem, that is, the aircraft N_z , α , and δ_e responses to a step input, QDES did quite well. Although the N_z responses were not always picture perfect, they remained within the constraint envelope. Furthermore, QDES was easily able to manipulate them by reducing the overshoot or oscillation of the response to accommodate higher N_{zcmd} and still remain within the AOA constraint. This capability allowed quick determination of the maximum achievable performance of the aircraft at the specified flight condition. The results from this type evaluation will indicate whether or not the aircraft can be controlled up to the limits of performance; however, the actual values of N_z , α , or δ_e should be viewed with

caution. This is because, even at moderate g levels over a short time period, the angles which were assumed small in the equations of motion linearization process are no longer small. This is a limitation of the modeling of the problem and occurs whenever the linearized equations are used regardless of the controller design methodology.

8.2.2 Frequency Response The success of QDES in constraining the input sensitivity and thus evaluating the achievable vector stability margins was mixed. For the A-4D full model example, the sensitivity had a high amplitude spike at the Phugoid frequency that could not be constrained at 1 g even though QDES said it had found a feasible solution. The same phenomena occurred in the F-16 full model example. The most logical explanation for this is that QDES did not get enough plant data to see and control the Phugoid mode. Although the sampling rate used was more than sufficient to capture the low frequency Phugoid (it was based on the much faster actuator pole), the sampling time was not a sufficient fraction of the relatively long (100 sec) Phugoid period. Thus, the impulse response matrices, from which QDES gets its information, did not contain enough samples for QDES to see the Phugoid. However, this does not explain why the sensitivity was constrained *above* 1 g for the A-4D example but *not* for the F-16. The author feels this has to do with the information given to QDES. QDES does not use the uncompensated plant for its design as is done classically. Rather, it takes the complete closed loop system with a nominal controller already in place. It is felt that differences in this nominal plant can cause different results from QDES in the area of the input sensitivity. For example, Appendix (A), Section (A.2) contains the state feedback gain matrix (K), the Kalman filter gain matrix (L), and the discrete poles of the nominal system for each A-4D full model run. Notice that although K remains the same for all runs, L and the discrete poles do change. The most significant changes in the discrete poles occur between 1 and 3 g 's when the sensitivity goes from being unconstrained to constrained.

From 3 g's on up, the values are similar, as are the frequency response results. The F-16 full model Appendix (B), Section (B.2) did not exhibit exactly the same behavior. Only the first discrete pole changed significantly from 1 to 9.8 g's and then remained essentially unchanged for the rest of the runs. All the other discrete poles remained virtually unchanged through all the runs. Interestingly, the low frequency Phugoid poles are *not* the ones changed when going from 1 g to higher g levels in either the A-4D or F-16 case. One would expect that for a change in the nominal system poles to improve the QDES results, this change would be a movement of the Phugoid poles to a higher frequency. Turning to the Short Period examples, the A-4D Short Period performed well in the frequency response area with no unconstrained low frequency spikes. This is what one would expect, having removed the low frequency Phugoid mode. However, the F-16 Short Period did not perform as well. Although no low frequency spike was observed, the sensitivity across the entire frequency range would not be constrained below approximately 20 regardless of the g level. Examination of the change in the discrete poles of the nominal system from 1 to higher g levels show the same behavior as the F-16 full model. The first pole changes significantly from 1 g to the higher levels, but is unchanged after that. The remainder of the poles are unchanged throughout all the runs. None of these discrete poles are low frequency Phugoid type.

8.3 *Other Approaches*

In an attempt to see if some of these frequency problems could be side-stepped, several different methods were tried with mixed results.

8.3.1 *Longer Run Times* In an attempt to capture the elusive Phugoid mode, some examples with longer run times were tried. However, to get a solution, the sampling rate had to be reduced to well below the minimum required to capture the fastest

pole. Even then, in only one case with the A-4D sampled at 3 Hz for 20 seconds was the sensitivity constrained. This combination of sample rate and run time is incompatible with the limits of performance procedure developed in this thesis. Aside from possibly not capturing all the aircraft plant dynamics, the use of a long run time would result in no solutions because α and δ_e would increase rapidly as forward velocity decreased in the pitch up. The limits would soon be exceeded with no QDES solution returned.

8.3.2 Different Nominal Plant Since the A-4D full model responded favorably to a change in the nominal plant offered to QDES, an attempt was made to use the MATLAB "place" command to generate the K and L matrices as was done in the "Two Mass" example [2]. The intent was to move the Phugoid poles so as to increase their frequency. Several different placements were tried with results that were considerably worse in terms of the frequency response as compared to the weighing used in the examples shown in this thesis.

8.3.3 Output Margins An alternative to constraining and observing the input sensitivity is to do so at the output of the plant. This requires constraining the individual SISO transfer functions

$$\begin{array}{c} \frac{N_z}{N_{zcmd}} \\ \frac{\alpha}{N_{zcmd}} \\ \frac{q}{N_{zcmd}} \end{array}$$

This was attempted on the F-16 full model and Short Period approximations. The F-16 full model output sensitivities behaved in the same manner as the input sensitivity. The N_z sensitivity was flat at unity, while the α and q responses had magnitude spikes at the Phugoid frequency which would not be constrained at either 1 g or higher levels. The

Short Period output sensitivities were constrained below 2 in all cases. Thus, in the F-16 Short Period case, observing the margins at the output resulted in a markedly different and better result than observing the input margins.

8.4 Conclusion

The limits of performance methodology presented in thesis could be used with some cautions. The N_z performance results, subject to the α and δ_e constraints, are reliable with the caveats offered in Section (8.2.1) concerning the linearization process. However, the results from the vector margin evaluation are too inconsistent. Other design processes have produced controllers for the F-16 with phase margins near 60 deg. An area for future research might be investigation of the relationship between the nominal controller offered to QDES and the frequency results returned. To be truly useful for aircraft limits of performance evaluation, QDES needs to be modified. More speed, plus continuous time domain and MIMO independent stability margins capability would go a long way in overcoming the difficulties encountered in this thesis.

Appendix A. A-4D State Space Models and Design Information

This appendix contains the continuous state space model for the A-4D at flight condition 5. This model was used for the classical design of Chapter IV and is the starting point for the QDES design of Chapter V. It also contains the state feedback gain matrix (K), the Kalman filter gain matrix (L), and the discrete poles of the nominal closed loop system provided to QDES for each design run. The matrices and vectors are as represented in the state variable differential equations

$$\dot{x} = Ax + Bu$$

$$y = Cx + Du$$

A.1 A-4D Longitudinal Model at Flight Condition 5

A.1.1 Original Plant Model The state, input, and output vectors are as follows:

$$x = \begin{bmatrix} v \\ \alpha \\ \theta \\ q \end{bmatrix} \quad u = [\delta_e] \quad y = \begin{bmatrix} v \\ \alpha \\ \theta \\ q \\ N_z \end{bmatrix}$$

where

v is forward velocity (*normalized to V_0*).

α is angle of attack (*rad*).

θ is pitch angle (*rad*).

q is pitch rate (*rad/sec*).

N_z is normal acceleration at the aircraft center of gravity (*ft/sec²*).

δ_e is elevator deflection (*rad*).

The matrices associated with this model are as follows:

$$A = \begin{bmatrix} -0.01266 & -0.00588 & -0.05075 & 0 \\ -0.10100 & -0.81670 & 0 & 0.99840 \\ 0 & 0 & 0 & 1 \\ -0.21798 & -12.4304 & 0 & -1.42190 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ -0.08963 \\ 0 \\ -19.3880 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -64.0340 & -517.788 & 0 & -1.02708 \end{bmatrix} \quad D = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -56.8254 \end{bmatrix}$$

A.1.2 A-4D Plant with Actuator Dynamics Inserting the elevator actuator dynamics

$$\frac{\delta_e}{\delta_{ecmd}} = \frac{10}{s+10}$$

represented as

$$A = [-10] \quad B = [10] \quad C = [1] \quad D = [0]$$

and changing N_z to positive nose up yields

$$A = \begin{bmatrix} -0.01266 & -0.00588 & -0.05075 & 0 & 0 \\ -0.10100 & -0.81670 & 0 & 0.99840 & -0.08963 \\ 0 & 0 & 0 & 1 & 0 \\ -0.21798 & -12.4304 & 0 & -1.42190 & -19.3880 \\ 0 & 0 & 0 & 0 & -10 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 10 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 64.0340 & 517.788 & 0 & 1.02708 & 56.8254 \end{bmatrix} \quad D = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

The state, input, and output vectors are now

$$x = \begin{bmatrix} v \\ \alpha \\ \theta \\ q \\ \delta_e \end{bmatrix} \quad u = [\delta_{ecmd}] \quad y = \begin{bmatrix} v \\ \alpha \\ \theta \\ q \\ N_z \end{bmatrix}$$

where

v is forward velocity (*normalized to V_o*).

α is angle of attack (*rad*).

θ is pitch angle (*rad*).

q is pitch rate (*rad/sec*).

N_z is normal acceleration at the aircraft center of gravity (*ft/sec²*).

δ_e is elevator deflection (*rad*).

δ_{ecmd} is elevator deflection commanded (*rad*).

A.1.3 A-4D Model in Desired Units Converting forward velocity to ft/sec, angles and angular rates to degrees, and N_z to g's yields

$$A = \begin{bmatrix} -0.0126600 & -0.0650596 & -0.5615270 & 0 & 0 \\ -0.0091282 & -0.8167000 & 0 & 0.9984000 & -0.0896300 \\ 0 & 0 & 0 & 1 & 0 \\ -0.0197007 & -12.430400 & 0 & -1.4219000 & -19.388000 \\ 0 & 0 & 0 & 0 & -10 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 10 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0.0031366 & 0.2806347 & 0 & 0.0005566 & 0.0307986 \end{bmatrix} \quad D = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

with the same state, input, and output vectors and the following units

v is forward velocity (*ft/sec*).

α is angle of attack (*deg*).

θ is pitch angle (*deg*).

q is pitch rate (*deg/sec*).

N_z is normal acceleration at the aircraft center of gravity (*g's*).

δ_e is elevator deflection (*deg*).

$\delta_{e\text{cmd}}$ is elevator deflection commanded (*deg*).

A.2 A-4D QDES Design Information

This section contains the feedback gain matrices, Kalman filter gain matrices, and nominal system discrete poles for the various QDES design runs on the A-4D.

A.2.1 A-4D 1 g Command

$$K = [0.01311 \quad 0.19535 \quad -.01405 \quad -0.03883 \quad 0.08394]$$

$$L = \begin{bmatrix} 0.14333 & 2.80394 & 0 \\ 0.19384 & 2.63160 & 0 \\ 0.12734 & 2.84941 & 0 \\ 0.97931 & 0.05539 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Discrete poles:

-0.45308
0.74824 ± 0.34392i
0.30119
0.29055
0.99892 ± 0.00747i
0.98908 ± 0.00901i
0.28650

A.2.2 A-4D 3 g Command

$$K = [0.01311 \quad 0.19535 \quad -.01405 \quad -0.03883 \quad 0.08394]$$

$$L = \begin{bmatrix} -0.00074 & 6.28365 & 0 \\ 0.07557 & 5.78362 & 0 \\ -0.00205 & 6.27067 & 0 \\ 0.99503 & 0.00725 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Discrete poles:

-0.15254
 0.74824 ± 0.34392i
 0.18425
 0.29055
 0.99914 ± 0.00769i
 0.98908 ± 0.00901i
 0.28650

A.2.3 A-4D 4.5 g Command

$$K = [0.01311 \quad 0.19535 \quad -0.01405 \quad -0.03883 \quad 0.08394]$$

$$L = \begin{bmatrix} -0.10271 & 6.88706 & 0 \\ -0.01630 & 6.32552 & 0 \\ -0.10177 & 6.85886 & 0 \\ 0.99634 & -0.00096 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Discrete poles:

0.08924 ± 0.22067i
 0.74824 ± 0.34392i
 0.99925 ± 0.00786i
 0.29055
 0.98908 ± 0.00901i
 0.28650

A.2.4 A-4D 4.6 g Command The values for the 4.6 g command are virtually identical to the 4.5 g values.

A.3 A-4D Short Period Approximation Flight Condition 5

The state, input, and output vectors are as follows:

$$x = \begin{bmatrix} \alpha \\ q \\ \delta_e \end{bmatrix} \quad u = [\delta_e] \quad y = \begin{bmatrix} \alpha \\ q \\ N_z \end{bmatrix}$$

where

α is angle of attack (*deg*).

q is pitch rate (*deg/sec*).

N_z is normal acceleration at the aircraft center of gravity (*g's*).

δ_e is elevator deflection (*deg*).

The matrices associated with this model are as follows:

$$A = \begin{bmatrix} -0.8167000 & 0.9984000 & -0.0896300 \\ -12.430400 & -1.4219000 & -19.388000 \\ 0 & 0 & -10 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 0 \\ 10 \end{bmatrix}$$
$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0.2806347 & 0.0005566 & 0.030796 \end{bmatrix} \quad D = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

A.4 A-4D Short Period Approximation QDES Design Information

This section contains the feedback gain matrices, Kalman filter gain matrices, and nominal system discrete poles for the various QDES design runs on the A-4D Short Period approximation.

A.4.1 A-4D Short Period 1 g Command

$$K = [0.06590 \quad -0.03256 \quad 0.07101]$$

$$L = \begin{bmatrix} 0.39196 & 2.05267 & 0 \\ 0.96155 & 0.11053 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Discrete poles:

-0.34372
 0.91255 ± 0.15801i
 0.60934
 0.66816
 0.60653

A.4.2 A-4D Short Period 3 g Command

$$K = [0.06590 \quad -0.03256 \quad 0.07101]$$

$$L = \begin{bmatrix} 0.29054 & 5.25594 & 0 \\ 0.99024 & 0.02736 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Discrete poles:

-0.27149
 0.91255 ± 0.15801i
 0.60934
 0.65167
 0.60653

A.4.3 A-4D Short Period 4.6 g Command

$$K = [0.06590 \quad -0.03256 \quad 0.07101]$$

$$L = \begin{bmatrix} 0.16411 & 6.74475 & 0 \\ 0.99265 & 0.01013 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Discrete poles:

-0.17346
0.91255±0.15801i
0.63142
0.60934
0.60653

A.4.4 A-4D Short Period 4.7 g Command

$$K = [0.06590 \quad -0.03256 \quad 0.07101]$$

$$L = \begin{bmatrix} 0.15607 & 6.80685 & 0 \\ 0.99274 & 0.00944 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Discrete Poles:

-0.16705
0.91255±0.15801i
0.60934
0.62997
0.60653

Appendix B. F-16 State Space Models and Design Information

This appendix contains the continuous state space model for the F-16 at sea level and Mach 0.6 as used in Chapter VI. It also contains the state feedback gain matrix (K), the Kalman filter gain matrix (L), and the discrete poles of the nominal closed loop system provided to QDES for each design run. The state, input, and output vectors are the same as used for the A-4D in Appendix (A). Initial units are slightly different as noted.

B.1 F-16 Full Longitudinal Model

The matrices associated with the original plant without actuator dynamics are:

$$A = \begin{bmatrix} -0.01485 & 37.3820 & -32.2000 & -17.9400 \\ -0.00008 & -1.49100 & -0.00130 & 0.00600 \\ 0 & 0 & 0 & 1 \\ -0.00036 & 9.75300 & 0.00029 & -0.96000 \end{bmatrix} \quad B = \begin{bmatrix} -0.00214 \\ 0.18800 \\ 0 \\ 19.0400 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -0.00166 & -31.0239 & -0.027049 & -0.08323 \end{bmatrix} \quad D = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 3.9118 \end{bmatrix}$$

with units:

v (ft/sec)

α (rad)

θ (rad)

q (rad/sec)

N_z (g's)

δ_e (rad)

Adding the actuator dynamics:

$$\frac{\delta_e}{\delta_{ecmd}} = \frac{20}{s+20}$$

as in Appendix (A), converting to the same units as used for the A-4D, and changing N_z to positive nose up yields

$$\begin{aligned} A &= \begin{bmatrix} -0.01485 & 0.65239 & -0.56195 & -0.31308 & -0.00004 \\ -0.00458 & -1.49100 & -0.00130 & 0.99600 & 0.18800 \\ 0 & 0 & 0 & 1 & 0 \\ -0.02063 & 9.75300 & 0.00029 & -0.96000 & 19.0400 \\ 0 & 0 & 0 & 0 & 20 \end{bmatrix} & B &= \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 20 \end{bmatrix} \\ C &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0.001660 & 0.514429 & 0.000472 & 0.001452 & -0.068268 \end{bmatrix} & D &= \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \end{aligned}$$

B.2 F-16 QDES Design Information

This section contains the feedback gain matrices, Kalman filter gain matrices, and nominal system discrete poles for the various QDES design runs for the F-16.

B.2.1 F-16 1 g Command

$$K = [-0.01648 \quad 0.58854 \quad 0.11327 \quad 0.24277 \quad 0.21629]$$

$$L = \begin{bmatrix} 0.67653 & 0.57277 & 0 \\ 0.71583 & 0.50054 & 0 \\ 0.72843 & 0.47892 & 0 \\ 0.78094 & 0.39017 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Discrete Poles:

0.41699±0.42016i
 0.28605
 0.74142
 0.88567
 0.98139
 0.99972±0.00373i
 0.99192
 0.28650

B.2.2 F-16 9.8 g Command

$$K = [-0.01648 \quad 0.58854 \quad 0.11327 \quad 0.24277 \quad 0.21629]$$

$$L = \begin{bmatrix} 0.97855 & 0.08234 & 0 \\ 0.97913 & 0.07223 & 0 \\ 0.98051 & 0.06887 & 0 \\ 0.98735 & 0.05445 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Discrete Poles:

0.44808±0.54110i
 0.28605
 0.74142
 0.88567
 0.98139
 0.99980±0.00377i
 0.99192
 0.28650

B.2.3 F-16 10.0 g Command

$$K = [-0.01648 \quad 0.58854 \quad 0.11327 \quad 0.24277 \quad 0.21629]$$

$$L = \begin{bmatrix} 0.97873 & 0.08071 & 0 \\ 0.97929 & 0.07079 & 0 \\ 0.98066 & 0.06750 & 0 \\ 0.98746 & 0.05337 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Discrete Poles:

0.44810 ± 0.54116i
 0.28605
 0.74142
 0.88567
 0.98139
 0.99980 ± 0.00377i
 0.99192
 0.28650

B.2.4 F-16 10.2 g Command

$$K = [-0.01648 \quad 0.58854 \quad 0.11327 \quad 0.24277 \quad 0.21629]$$

$$L = \begin{bmatrix} 0.97890 & 0.07913 & 0 \\ 0.97943 & 0.06942 & 0 \\ 0.98079 & 0.06619 & 0 \\ 0.98757 & 0.05233 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Discrete Poles:

0.44812 ± 0.54121i
 0.28605
 0.74142
 0.88567
 0.98139
 0.99980 ± 0.00377i
 0.99192
 0.28650

B.2.5 F-16 10.4 g Command

$$K = [-0.01648 \quad 0.58854 \quad 0.11327 \quad 0.24277 \quad 0.21629]$$

$$L = \begin{bmatrix} 0.97906 & 0.07762 & 0 \\ 0.97957 & 0.06809 & 0 \\ 0.98093 & 0.06493 & 0 \\ 0.98768 & 0.05133 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Discrete Poles:

$$0.44814 \pm 0.54127i$$

$$0.28605$$

$$0.74142$$

$$0.88567$$

$$0.98139$$

$$0.99980 \pm 0.00377i$$

$$0.99192$$

$$0.28650$$

B.3 F-16 Short Period Approximation

The state, input, and output vectors with their associated quantities and units are the same as for the A-4D as listed in Section (A.3). The matrices associated with the model are as follows:

$$A = \begin{bmatrix} -1.49100 & 0.99600 & 0.18800 \\ 9.75300 & -0.9600 & 19.040 \\ 0 & 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 0 \\ 20 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0.54143 & 0.00145 & -0.06827 \end{bmatrix} \quad D = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

B.4 F-16 Short Period Approximation QDES Design Information

This section contains the feedback gain matrices, Kalman filter gain matrices, and nominal system discrete poles for the various QDES design runs on the F-16 Short Period approximation.

B.4.1 F-16 Short Period 1 g Command

$$K = [0.59479 \quad 0.21936 \quad 0.19813]$$

$$L = \begin{bmatrix} 0.71714 & 0.49985 & 0 \\ 0.78214 & 0.38947 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$0.41735 \pm 0.42185i$$

$$0.88681$$

$$0.28605$$

$$0.74049$$

$$0.28650$$

B.4.2 F-16 Short Period 10.4 g Command

$$K = [0.59479 \quad 0.21936 \quad 0.19813]$$

$$L = \begin{bmatrix} 0.97958 & 0.06786 & 0 \\ 0.98770 & 0.05113 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$0.44840 \pm 0.54225i$$

$$0.88681$$

$$0.28605$$

$$0.74049$$

$$0.28650$$

B.4.3 F-16 Short Period 10.6 g Command

$$K = [0.59479 \quad 0.21936 \quad 0.19813]$$

$$L = \begin{bmatrix} 0.97971 & 0.06659 & 0 \\ 0.98780 & 0.05017 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

0.44841 ± 0.54229i

0.88681

0.28605

0.74049

0.28650

Appendix C. MATLAB Script Files

This appendix contains representative copies of the MATLAB script files written by the author and used for the analysis in this thesis. They are recommended as models for use along with the procedures outlined in Chapters V and VII.

C.1 The "nzprp.m" File

```
% Filename: "Qdes sourcefile name + nzprp.m" (for normal acceleration prep)

% This script file takes the aircraft plant data for the longitudinal axis
% and forms the Impulse Response Matrices (_t1, _t2, _t3) needed by Qdes as
% well as the _ttt.mat matrix needed to form the Qdes controller. To use
% this script, you need the aircraft longitudinal state space plant matrices
% and the actuator dynamics. You will also need to have the nine system
% matrices derived from the standard P-K form of the aircraft and controller
% set up for NORMAL ACCELERATION (NZ) command with your desired Exogenous
% Inputs (w), Regulated Outputs (z), Sensed Outputs (y), and Actuator Inputs
% (u).

% Comments in ALL CAPS indicate where you are to enter data.

% Enter plant in rad, normalized Velocity, and ft/sec^2 (Nz at cg)

a=[-.01266,-.00588,-.05075,0;
   -.101,-.8167,0,.9984;
   0,0,0,1;
   -.21798,-12.4304,0,-1.4219];

b=[0;
   -.08963;
   0;
   -19.388];

c=[1,0,0,0;
   0,1,0,0;
   0,0,1,0;
   0,0,0,1;
   -64.034,-517.788,0,-1.02708];

d=[0;
   0;
   0;
   0;
   -56.8254];

% The States Are: ubar (normalized forward velocity)
```

```

%           alpha (angle of attack)
%           theta (pitch angle)
%           q      (pitch rate)
%

% Outputs are: ubar
%           alpha
%           theta
%           q
%           nzog (normal g at the center of gravity)

% Insert actuator dynamics

ap=[a,b;zeros(1,4),-10];
bp=[zeros(4,1);10];
cp=[eye(4),zeros(4,1);c(5,:),d(5,:)];
dp=zeros(5,1);

%                               NOTE: All subsequent
% The States are now: ubar      calculations in this file
%           alpha      as well as the nzsst.m file
%           theta      that produces the results
%           q          assume these states in this order.
%           deltae (elevator deflection)

% Outputs are: ubar
%           alpha
%           theta
%           q
%           nzog

% Change sign of the last row of cp (nzog) to make positive 'g' up.
cp(5,:)=-cp(5,:);

% Convert to deg, ftsSec, and g's.

T=diag([634,57.3,57.3,57.3,57.3]); % State Scaling

S=[57.3]; % Input Scaling

W=diag([634,57.3,57.3,57.3,(1/32.2)]); % Output Scaling

apd=T*ap*inv(T);
bpd=T*bp*inv(S);
cpd=W*cp*inv(T);
dpd=W*dp*inv(S);

% ENTER THE DESIRED SCALING FACTOR. This allows you to analyse the system
% for other than a unit step input. If you put in put in 1, the normal
% acceleration command input is 1 g. If you put in 10, the normal acceleration
% command is 10 g's, etc. Note that all the constraints are still written

```

```
% as if you are using a unit step, but the output of Nz will be the response
% to a 10 g command and all the states will reflect the desired 10 g command.
% Note that you MUST also enter the same scaling factor in the "nzpst.m" file
% which produces the plots of the system response.
```

```
Sfnz=1;
cpds=(1/Sfnz)*cpd;
```

```
% ENTER THE PLANT DISCRETIZATION RATE IN HZ.
```

```
Srate=8;
```

```
Ts=1/Srate;
[ad,bd,cd,dd]=c2dm(apd,bpd,cpds,dpd,Ts,'zoh');
```

```
% ENTER THE NINE QDES STYLE MATRICES. Note: The matrices must be labeled (pa,
% pbw, pbu, pcz, pcy, pdzw, pdyw, pdzu, pdyu) for use in the script file
% "obc2ttt".
```

```
pa=ad;
pbw=[zeros(4,2);0,10]; % input matrix for exogenous inputs (w)
pbu=bd; % input matrix for actuator inputs (u)
pcz=[0,1,0,0,0;
     cd(5,:);
     0,0,0,1,0;
     0,0,0,0,1;
     zeros(1,5)]; % output matrix for regulated outputs (z)
pcy=[0,0,0,1,0;
     cd(5,:);
     zeros(1,5)]; % output matrix for sensed outputs (y)
pdzw=[zeros(4,2);
      0,1]; % d matrix from w to z;
pdzu=[zeros(4,1);1]; % d matrix from u to z
pdyw=[0,0;
      0,0;
      1,0]; % d matrix from w to y
pdyu=[0;0;0]; % d matrix from u to y; zero; plant is s.p.
```

```
% Form an observer based nominal controller.
```

```
% Since pydu=0, the plant is strictly proper. Can use "obc2ttt". Therefore
% need "k" the State Feedback Matrix and "l" the Observer Feedback Matrix.
```

```
% Find "k" using "dlqr"
```

```
% State weighting based on formula  $(1/X_i^2)$  where  $X_i$  is max value the state
% is allowed to take on.
```

```
Q1=diag([.000044,.0016,.00111,.00111,.0016]); % State weighting
R1=.1; % Control weighting
```

```

[k,S,E]=dlqr(ad,bd,Q1,R1);

% Find "l" Using "dlqe"

G=[1;1;1;0]; % Put "1" on every state that could be affected by a plant
               % disturbance (wind gust) and zero on others (actuator state).

Q2=.1;
R2=.001*eye(size(pdyu,1));

l=dlqe(ad,G,pcy,Q2,R2);

% Form the closed loop A matrix with the nominal controller and check
% its poles.

ACL=[pa,-pbu*k;l*pcy,pa-l*pcy-pbu*k];

% Use "obc2ttt" to get the T1, T2, and T3 Impulse Response Matrices

% ENTER THE NAME OF THE QDES SOURCE FILE HERE
%
source = 'A4D7NZ';

% ENTER THE NUMBER OF SAMPLES This is the plant discretization rate (Srate)
% from above time the number of seconds you wish to analyse the system.
% (ex: 20 hz times 10 seconds = 200. n_sample = 200)

n_sample =40 ;

obc2ttt;

disp(' ');
disp('*****');
disp(' ');
disp('When writing your Qdes source file, be sure that');
disp(' ');
Z1='          n_sample=';
Z2=[Z1 num2str(n_sample)];
disp(Z2);
disp(' ');
disp('so that it matches with the impulse response matrices you just created');
disp(' ');
Z8=' Hz ';
Z9=' sec';
Z5='For this set up your sample rate was: ';
Z7='Sample time was: ';
Z6=[Z5 num2str(Srate) Z8 Z7 num2str(Ts) Z9];
disp(Z6);
disp(' ');
disp(' ');
disp('Also, be sure that your "nzipst.m" file uses the Scale Factor');
disp(' ');

```

```

Z3='          Sfnz=';
Z4=[Z3 num2str(Sfnz)];
disp(Z4);
disp(' ');
disp(' ');
disp('*****');
end;

```

C.2 The QDES Source File

```

/* Source File For A4D7NZ  Filename: "A4D7NZ" */

/* This Is The PREAMBLE */

/* exogenous inputs (w) */

#define NZ_COM          1 /* reference input (Nz command) */
#define U_IN            2 /* loop input for u */

/* regulated outputs (z) */

#define ALF_OUT          1 /* angle of attack (alpha) */
#define NZ_OUT           2 /* normal accel (Nz) */
#define Q_OUT            3 /* pitch rate (q) */
#define EL_OUT           4 /* elevator deflection (deltae) */
#define U_OUT            5 /* elevator deflection commanded (deltaecomm) */

/* sensed outputs (y) */

#define Q_OUT_MEAS       1 /* measured q for feedback */
#define NZ_OUT_MEAS      2 /* measured Nz for feedback */
#define NZ_REF           3 /* reference output (Nz command) */

/* This Is The DECLARATIONS Section */

/* Note: n_act*n_sens*n_tap must be < 200 */

declarations {
    n_exog = 2;
    n_reg = 5;
    n_sens = 3;
    n_act = 1;
    n_sample = 40;
    n_tap = 35;

    t1_coeffs = {
#include "A4D7NZ_t1"
    };

```

```

        t2_coeffs = {
#include "A4D7NZ_t2"
        };

        t3_coeffs = {
#include "A4D7NZ_t3"
        };
}

/* This is the CONSTRAINTS Section */

constraints {

/* Put Constraints On U_OUT/U_IN Vector Margins */

for w=0.0 to %pi step 0.02 do {
mag_H[5][2](1,w)<=1.6628;
}

/* Put Constraints On Elevator Deflection */

for t=0 to n_sample-1 do
-25.0<=u_step[4][1]<=25.0;

/* Put Constraints on Angle of Attack */

for t=0 to n_sample-1 do
-5.0<=u_step[1][1]<=18.0;

/* Put Constraints On Overshoot */

overshoot[2][1]<= 0.1;

/* Put Constrains On Undershoot */

undershoot[2][1]<=0.05;

/* Put Constraints On Settling Time */

for t=16 to 20 do
1-0.8392^t<=u_step[2][1](t)<=1+0.8392^t;

/* Put Constraints On Steady State Error */

for t=20 to n_sample-1 do
0.97+0*(t-20)<=u_step[2][1](t)<=1.03-0*(t-20);

/* Put Constraint On Rise Time */

for t=12 to n_sample-1 do
0.8<=u_step[2][1](t);
}

```

C3 The "nzpst.n" File

```
% Filename: "Qdes source file name + nzpst.m" (for normal acceleration post)

% This file is run after successfully running the Qdes source file and getting
% the Q coefficients. (files .out and .q)

% This file takes the Qdes formed Q coefficients, forms the Qdes controller,
% forms the closed loop system, and simulates the system response to a
% step input of normal acceleration. It outputs a plot of the normal
% acceleration time response with its constraint envelop, angle of attack and
% elevator deflection time histories, the input sensitivity bode magnitude plot,
% the closed loop bode magnitude plot, and the input vector gain and phase
% margins.

% Note that if you used a scale factor in the "nzprp.m" file, the plot of normal
% acceleration will be scaled to represent the response to your desired
% commanded normal acceleration. The time histories will show the actual
% values of the states (angle of attack and elevator deflection).

% You will need the inputs you made to "visnz.m" so that this file can
% draw the constraint envelope you imposed on the system on the output plot
% of normal acceleration.

% Remarks in ALL CAPS indicate where you are to enter data.

% Use "qobc2comp" to get Q coefficients from ".q" and form the controller
% Note: "qobc2comp" uses "_ttt.mat" Generated in "nzprp.m" with "obc2ttt"
% Therefore it must be available in the directory you are working in.

% ENTER THE NAME OF THE QDES SOURCE FILE HERE
%
[ka,kb,kc,kd]=qobc2comp('A4D7NZ');

% Use "gethzw" to form the closed loop system (Hzw)

% ENTER THE NAME OF THE QDES SOURCE FILE HERE
%
[ha,hb,hc,hd]=gethzw('A4D7NZ',ka,kb,kc,kd);

% Simulate the closed loop system response to a unit step input

% ENTER PLANT DISCRETIZATION RATE (IN HZ) AND TIME YOU WANT THE SIMULATION
% TO RUN FOR. These MUST match the values you have been using all along in
% "visnz.m", "nzprp.m", and the Qdes source file.

Srate=8;
Tmax=5;

t=0:(1/Srate):Tmax;
```



```

% ENTER THE SCALE FACTOR USED IN "nzprp.m"

Sfnz=1
hcs=[hc(1,:);Sfnz*hc(2,:);hc(3,:);hc(4,:);hc(5,:)];

% Create the input vector (w for the closed loop system Hzw)

% ENTER INPUTS FOR THE SIMULATION. NOTE THAT THE ORDER OF THE
% INPUTS MUST MATCH THAT USED IN THE PREAMBLE OF THE QDES SOURCE FILE.

u1=ones(length(t),1); % Reference (commanded normal acceleration) (always 1)
u2=zeros(length(t),1); % U_IN (always zero)

u=[u1,u2];

[Y,X]=dlsim(ha,hb,hcs,hd,u);

% ENTER DATA FROM "visnz.m" FOR THE NORMAL ACCELERATION CONSTRAINT ENVELOPE

Trnz=1.5;
Mpnz=0.1;
Usnz=-.05;
Tsnz=2.5;
pernz=3;
Essnz=3;
Tsnzst=2.0;
Ynz=.2;
Tmponz=1.5;
Tyznz=0.5;
Tynznz=1.5;
Ytynznz=.80;

%%%%%%%%%% calculations for normal acceleration envelope plot %%%%%%%%%%%

t1nz=0;t2nz=0;t3nz=0;t4nz=0;t5nz=0;t6nz=0;t7nz=0;t8nz=0;mpnz=0;trnz=0;
tsnzup=0;tsnzlo=0;essnzup=0;essnzlo=0;hi13nz=0;hi23nz=0;lo13nz=0;lo23nz=0;
usnz=0;

Stime=1/Srate;

Mpnz=Mpnz+1;
t1nz=0:Stime:Tsnzst; % Peak Overshoot
for i=1:length(t1nz),
    mpnz(i)=Mpnz;
end;
mpnz=mpnz*Sfnz;

for i=1:length(t1nz), % Undershoot
    usnz(i)=Usnz;
end;
usnz=usnz*Sfnz;

```

```

t2nz=Tmz:Stime:Tmax;           % Rise Time
for i=1:length(t2nz),
    trnz(i)=0.8;
end;
trnz=trnz*Sfnz;

t3nz=Tsnzst:Stime:Tsnz;         % Settling Time
Tsnzeb=(.01*pernz)^(1/(Tsnz*Srate));
for i=1:length(t3nz),
    tsnzup(i)=1+Tsnzeb^(i+Tsnzst*Srate);
    tsnzlo(i)=1-Tsnzeb^(i+Tsnzst*Srate);
end;
tsnzup=tsnzup*Sfnz;
tsnzlo=tsnzlo*Sfnz;

if Essnz==0,                    % Steady State Error When Set To Zero
    t4nz=0:Stime:Tmax;          % Continues Curve From Settling Time
    for i=Tsnz*Srate:(Tmax*Srate)+1,
        essnzup(i)=1+Tsnzeb^(i);
        essnzlo(i)=1-Tsnzeb^(i);
    end;
else
    t4nz=Tsnz:Stime:Tmax;       % Steady State Error When Not Zero
                                % Draws Line From Y Value @ Tsq To Essq @ Tmax

    essnzdcz=((1+pernz*.01)-(1+Essnz*.01))/(Tmax*Srate-Tsnz*Srate);
    for i=1:(Tmax*Srate-Tsnz*Srate)+1,
        essnzup(i)=(1+pernz*.01)-essnzdcz*(i);
        essnzlo(i)=(1-pernz*.01)+essnzdcz*(i);
    end;
end;
essnzup=essnzup*Sfnz;
essnzlo=essnzlo*Sfnz;

hil1nz=(.5*(Mpnz-Ynz))^(1/3);
hil2nz=hil1nz/(.4*Tmpnz*Srate);
t5nz=0:Stime:(.4*Tmpnz);
for i=1:(.4*Tmpnz*Srate)+1,
    hil3nz(i)=(hil2nz*(i-1))^3+Ynz;
end;
hil3nz=hil3nz*Sfnz;

hi21nz=(Mpnz-.5*(Mpnz-Ynz))-Ynz)^(1/3);
tdifhinz=(Tmpnz*Srate)-(.4*Tmpnz*Srate);
hi22nz=hi21nz/tdifhinz;
t6nz=.4*Tmpnz:Stime:Tmpnz;
for i=1:tdifhinz+1,
    hi23nz(i)=((Mpnz-.5*(Mpnz-Ynz))-Ynz)-((hi21nz-hi22nz*(i-1))^3) ...
        +(Mpnz-.5*(Mpnz-Ynz));
end;
hi23nz=hi23nz*Sfnz;

```

```

lo11nz=(.5*Ytynznz)^(1/3);
tdiflo1nz=.4*((Tynznz-Tyznz)*Srate);
lo12nz=lo11nz/tdiflo1nz;
t7nz=Tyznz:Stime:(Tyznz+.4*(Tynznz-Tyznz));
for i=1:tdiflo1nz+1,
    lo13nz(i)=(lo12nz*(i-1))^3;
end;
lo13nz=lo13nz*Sfnz;

lo21nz=(.5*(Ytynznz))^(1/3);
tdiflo2nz=.6*((Tynznz-Tyznz)*Srate);
lo22nz=lo21nz/tdiflo2nz;
t8nz=.4*(Tynznz-Tyznz)+Tyznz:Stime:Tynznz;
for i=1:tdiflo2nz+1,
    lo23nz(i)=.5*(Ytynznz)-(lo21nz-lo22nz*(i-1))^3+.5*Ytynznz;
end;
lo23nz=lo23nz*Sfnz;

%%%%%%%%%%%% plot normal g time response and constraint envelope %%%%%%%%%%

figure(1);

subplot(111);
plot(t,Y(:,2)); % plots normal g time response
%title('A4D NZ_OUT/NZ_COM');
ylabel('Normal g (gs)');
xlabel('Time (sec)');
grid;
hold;
plot(t1nz,mpnz,':'); % plots peak overshoot line
plot(t1nz,usnz,':'); % plots undershoot line
plot(t2nz,tmnz,':'); % plots rise time line
plot(t3nz,tsnzup,':'); % plots upper settling time curve
plot(t3nz,tsnzlo,':'); % plots lower settling time curve
plot(t4nz,essnzup,':'); % plots upper steady state error curve
plot(t4nz,essnzlo,':'); % plots lower steady state errorcurve
%plot(t5nz,hi13nz,':'); % plots 1st half of upper constraint curve
%plot(t6nz,hi23nz,':'); % plots 2nd half of upper constraint curve
%plot(t7nz,lo13nz,':'); % plots 1st half of lower constraint curve
%plot(t8nz,lo23nz,':'); % plots 2nd half of lower constraint curve
hold;

t1nz=0;t2nz=0;t3nz=0;t4nz=0;t5nz=0;t6nz=0;t7nz=0;t8nz=0;mpnz=0;tmnz=0;
tsnzup=0;tsnzlo=0;essnzup=0;essnzlo=0;hi13nz=0;hi23nz=0;lo13nz=0;lo23nz=0;
usnz=0;

%%%%%%%%%%%% plot angle of attack and elevator deflection, %%%%%%%%%%

figure(2);

subplot(211);
plot(t,Y(:,4));

```

```

%title('Elevator Deflection to Normal G Command');
ylabel('deltae (deg)');
grid;

subplot(212);
plot(t,Y(:,1))
%title('Angle of Attack to Normal G Command');
ylabel('alpha (deg)');
xlabel('Time (sec)');
grid;

%%% figure M-Circle radius and gain and phase margins for U_OUT/U_IN %%%%

[M,P,W]=dcode(ha,hb,hc,hd,Stime,2);
m_crcl=1/(max(M(:,5)));
upgmdb=-20*log10(1-m_crcl)
lwgmdb=-20*log10(1+m_crcl)
prdeg=57.3*2*asin(m_crcl/2)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% plot input sensitivity of U_OUT/U_IN %%%%%%%%%%

MdB=20*log10(M);

figure(3);

subplot(211);
semilogx(W,M(:,5));
ylabel('Magnitude (not dB)');
grid;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% plot complimentary sensitivity %%%%%%%%%%
tdb=0;

[M2,P2,W2]=dcode(ha,hb,hc,hd,Stime,1);
M2dB=20*log10(M2);

subplot(212);
semilogx(W2,M2dB(:,2));
grid;
hold;
for i=1:length(W2),
    tdb(i)=-3;
end;
semilogx(W2,tdb,':');
%title('Complimentary Sensitivity NZ_OUT/NZ_COM');
ylabel('Magnitude (dB)');
hold;
xlabel('Frequency (rad/sec)');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% output the size of the Qdes controller %%%%%%%%%%

ord=size(ka,1);

```

```

X1=num2str(ord);
X2='The Order Of The Qdes Controller is ';
Z1=[X2 X1];
disp(' ');
disp(' ');
disp('*****');
disp(' ');
disp(Z1);
disp(' ');
disp('*****');
end;

```

C.4 The "visz.m" File

```

% This file assists in developing the unit step response constraint envelope
% for the aircraft normal acceleration (Nz)
% Filename: "visnz.m".

% In addition to drawing the constraint envelope, the program generates the
% Qdes Source File statements (in proper format and syntax) to force Qdes to
% impose the exact same constraint envelope as that developed by this program.
% Also the program will generate the Qdes Source File statements that impose
% vector margins of your choice on any desired SISO transfer function of the
% system.

% The program requires that you enter the desired Rise Time (Tr), Settling Time
% (Tsl) and desired percent (per), Overshoot (Mp), Undershoot (Us), and
% Steady State Error (Ess) for the time response of Nz.

% Additional constraint barriers can be constructed that do not correspond to
% any particular step response figure of merit but are useful in shaping the
% response. These are described prior to the section where you are required
% to input the specifications.

% The program also requires you to enter the discretization parameters that you
% used when setting up the control problem for Qdes. This involves the
% Sample Rate (in Hz) and the time period you asked Qdes to work over (in sec).
% This turns out to be "n_sample" times the sample time which is the inverse
% of the Sample Rate (ie. 1/20 Hz = .05 secs).

% Comments in ALL CAPS indicate where you are to enter data.

% ENTER THE DESIRED VECTOR MARGINS FOR THE SISO TRANSFER FUNCTION YOU WISH TO
% CONSTRAIN. LOWER GAIN MARGIN MUST BE NEGATIVE.

Dugmz=7; % Desired Upper Gain Margin (dB) Range: (0.0 < Dugmz < +inf)
Dlgmz=-4; % Desired Lower Gain Margin (dB) Range: (-6.0 < Dlgmz < 0.0)
Dpmz=35; % Desired Phase Margin (deg) Range: (0.0 < Dpmz < 60)

% ENTER TIME PERIOD (IN SECS) FOR WHICH YOU ASKED QDES TO WORK

```

```

Tmax=5;

% ENTER SAMPLE RATE (IN HZ) AT WHICH YOU DICRITIZED THE PLANT

Srate=8;

Stime=1/Srate;

% ENTER THE FIVE UNIT STEP FIGURES OF MERIT YOU WISH QDES TO MEET

Trnz=1.5;      % rise time

Mpnz=0.1;      % overshoot

Usnz=-0.05;    % undershoot (enter negative number)

Tsnz=2.5;      % settling time
pernz=3;       % percent value at the settling time

Essnz=3;       % steady state error (must be <= pernz)

% The Settling Time constraint envelope is an exponential curve that starts
% at a time of your choosing, decays exponentially while it passes through
% the proper percentage above and below 1 at the requested Ts, and achieves
% the requested Ess at Tmax. (This is why Ess must be <= pernz).

% ENTER THE TIME YOU WISH THE SETTLING TIME CONSTRAINT TO BEGIN

Tsnzst=2;

% You may refine the upper constraint envelope with an exponential curve which
% starts at a point of your choosing on the y axis at time zero, and meets the
% overshoot line at a time of your choosing.

% ENTER Y-VALUE AT TIME ZERO WHERE YOU WANT THE UPPER CONSTRAINT LINE TO START

Ynz=.2;

% ENTER THE TIME AT WHICH YOU WANT THE UPPER CONSTRAINT LINE TO TOUCH THE
% OVERSHOOT LINE

Tmpernz=1.5;

% You may refine the lower constraint envelope with an exponential curve that
% starts at Y=0 at a time of your choosing and ends at a Y value and time
% of your choosing.

% ENTER THE TIME WHERE YOU WANT THE LOWER CONSTRAINT ENVELOPE CURVE TO START

Tyznz=0.5;

```

% ENTER THE POINT (TIME AND Y VALUE) WHERE YOU WANT THE LOWER CONSTRAINT
 % ENVELOPE CURVE TO END (normally at 0.8 and Tr).

Tynznz=1.5;
 Ytynznz=.80;

calculations for normal g #####
 t1nz=0;t2nz=0;t3nz=0;t4nz=0;t5nz=0;t6nz=0;t7nz=0;t8nz=0;mpnz=0;trnz=0;tsnzup=0;
 tsnzlo=0;essnzup=0;essnzlo=0;hi13nz=0;hi23nz=0;lo13nz=0;lo23nz=0;usnz=0;

Sens1nz=1/(1-10^(-Dugmnz/20));
 Sens2nz=1/(10^(-Dlgnz/20)-1);
 Sens3nz=1/(2*sin((Dpmnz/57.2958)/2));
 Sens4nz=[Sens1nz,Sens2nz,Sens3nz];
 Sensnz=min(Sens4nz);

Mpnz=Mpnz+1;
 t1nz=0:Stime:Tsnzst; % Peak Overshoot
 for i=1:length(t1nz),
 mpnz(i)=Mpnz;
 end;

for i=1:length(t1nz), % Undershoot
 usnz(i)=Usnz;
 end;

t2nz=Trnz:Stime:Tmax; % Rise Time
 for i=1:length(t2nz),
 trnz(i)=0.8;
 end;

t3nz=Tsnzst:Stime:Tsnz; % Settling Time
 Tsnzeb=(.01*pernz)^(1/(Tsnz*Srate));
 for i=1:length(t3nz),
 tsnzup(i)=1+Tsnzeb^(i+Tsnzst*Srate);
 tsnzlo(i)=1-Tsnzeb^(i+Tsnzst*Srate);
 end;

if Essnz==0, % Steady State Error When Set To Zero
 t4nz=0:Stime:Tmax; % Continues Curve From Settling Time
 for i=Tsnz*Srate:(Tmax*Srate)+1,
 essnzup(i)=1+Tsnzeb^(i);
 essnzlo(i)=1-Tsnzeb^(i);
 end;
else
 t4nz=Tsnz:Stime:Tmax; % Steady State Error When Not Zero
 % Draws Line From Y Value @ Tsq To Essq @ Tmax

essnzdcz=((1+pernz*.01)-(1+Essnz*.01))/(Tmax*Srate-Tsnz*Srate);
 for i=1:(Tmax*Srate-Tsnz*Srate)+1,
 essnzup(i)=(1+pernz*.01)-essnzdcz*(i);
 essnzlo(i)=(1-pernz*.01)+essnzdcz*(i);

```

end;
end;

hillnz=(.5*(Mpnz-Ynz))^(1/3);
hil2nz=hillnz/(.4*Tmpnz*Srate);
t5nz=0:Stime:(.4*Tmpnz);
for i=1:(.4*Tmpnz*Srate)+1,
    hil3nz(i)=(hil2nz*(i-1))^3+Ynz;
end;

hi2lnz=(Mpnz-.5*(Mpnz-Ynz))-Ynz)^(1/3);
tdifhinz=(Tmpnz*Srate)-(.4*Tmpnz*Srate);
hi22nz=hi2lnz/tdifhinz;
t6nz=.4*Tmpnz:Stime:Tmpnz;
for i=1:tdifhinz+1,
    hi23nz(i)=((Mpnz-.5*(Mpnz-Ynz))-Ynz)-((hi2lnz-hi22nz*(i-1))^3) ...
        +(Mpnz-.5*(Mpnz-Ynz));
end;

lol1nz=(.5*Ytynznz)^(1/3);
tdiflolnz=.4*((Tynznz-Tynznz)*Srate);
lol2nz=lol1nz/tdiflolnz;
t7nz=Tynznz:Stime:(Tynznz+.4*(Tynznz-Tynznz));
for i=1:tdiflolnz+1,
    lol3nz(i)=(lol2nz*(i-1))^3;
end;

lo2lnz=(.5*(Ytynznz))^(1/3);
tdiflo2nz=.6*((Tynznz-Tynznz)*Srate);
lo22nz=lo2lnz/tdiflo2nz;
t8nz=.4*(Tynznz-Tynznz)+Tynznz:Stime:Tynznz;
for i=1:tdiflo2nz+1,
    lo23nz(i)=.5*(Ytynznz)-(lo2lnz-lo22nz*(i-1))^3+.5*Ytynznz;
end;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% plot normal g constraint envelope %%%%%%%%%%
figure(1);
subplot(111);
plot(t1nz,mpnz,':');          % plots peak overshoot line
hold;
plot(t1nz,usnz,':');          % plots undershoot line
plot(t2nz,tmnz,':');          % plots rise time line
plot(t3nz,tsnzup,':');        % plots upper settling time curve
plot(t3nz,tsnzlo,':');        % plots lower settling time curve
plot(t4nz,essnzup,':');        % plots upper steady state error curve
plot(t4nz,essnzlo,':');        % plots lower steady state error curve
plot(t5nz,hil3nz,':');        % plots 1st half of upper constraint curve
plot(t6nz,hi23nz,':');        % plots 2nd half of upper constraint curve
plot(t7nz,lol3nz,':');        % plots 1st half of lower constraint curve
plot(t8nz,lo23nz,':');        % plots 2nd half of lower constraint curve
hold;
grid;

```



```

axis([0,Tmax,Usnz-.1,Mpnz+.2]);
xlabel('time (secs)');
ylabel('normal g (Nz)');
title('Normal G (Nz) Unit Step Response Constraint Envelope');

t1nz=0;t2nz=0;t3nz=0;t4nz=0;t5nz=0;t6nz=0;t7nz=0;t8nz=0;mpnz=0;trnz=0;tsnzup=0;
tsnzlo=0;essnzup=0;essnzlo=0;hi13nz=0;hi23nz=0;lo13nz=0;lo23nz=0;usnz=0;

##### generate Qdes source file constraint statements #####

A1='overshoot[out][in]<= ';
A2='for t=';
A3=' to ';
A4=' do';
A5='n_sample-1 do';
A6='<=u_step[out][in](t)<=';
A7='^t';
A8='1+';
A9='1-';
A10='+';
A11='*';
A12='(';
A13=')';
A15='t';
A16='-';

A19='u_step[out][in](t)<=';
A20='^3';
A25='<=u_step[out][in](t);';
A31='for w=0.0 to %pi step 0.02 do {';
A32='mag_H[out][in](1,w)<=';
A33=')';
A34='undershoot[out][in]<=';

S1='';

disp(' ');
disp(' ');
disp('*****');
disp('*****');
disp(' ');
disp('This is how your Qdes Source File constraint statements should');
disp(' ');
disp('look to draw the equivalent constraint envelope for Qdes and constrain');
disp(' ');
disp('the vector margins of your selected SISO transfer function.');
```

for normal acceleration

```

A14n=num2str(essnzdcz);
A17n=num2str(1-pernz*.01);
A18n=num2str(1+pernz*.01);
A21n=num2str((Mpnz-.5*(Mpnz-Ynz))-Ynz);
A22n=num2str(Mpnz-.5*(Mpnz-Ynz));
A23n=num2str(hi21nz);
A24n=num2str(hi22nz);
A26n=num2str(Ytynznz*.5);
A27n=num2str(lo21nz);
A28n=num2str(lo22nz);
A29n=num2str(Srate*(Tyznz+.4*(Tyznznz-Tyznz)));
A30n=num2str(.4*Tmpnz*Srate);

T1n=num2str(Trnz*Srate);
```

```

T2n=num2str(Tsnz*Srate);
T3n=num2str(Tsnzst*Srate);
T4n=num2str(Tmpnz*Srate);
T5n=num2str(Tyznz*Srate);
T6n=num2str(Tynznz*Srate);

B1n=[A1 num2str(Mpnz-1) S1];
B2n=[A2 T3n A3 T2n A4];
B3n=[A9 num2str(Tsnzeb) A7 A6 A8 num2str(Tsnzeb) A7 S1];
B4n=[A2 T2n A3 A5];
B5n=[A17n A10 A14n A11 A12 A15 A16 T2n A13 A6 A18n A16 A14n A11 A12 A15 A16 ...
      T2n A13 S1];
B6n=[A2 T1n A3 A5];
B7n=[A2 num2str(0) A3 A30n A4];
B8n=[A19 A12 num2str(hil2nz) A11 A15 A13 A20 A10 num2str(Ynz) S1];
B9n=[A2 A30n A3 T4n A4];
B10n=[A19 A21n A16 A12 A23n A16 A24n A11 A12 A15 A16 A30n A13 A13 A20 A10 ...
      A22n S1];
B11n=[A2 T5n A3 A29n A4];
B12n=[A12 num2str(1ol2nz) A11 A12 A15 A16 T5n A13 A13 A20 A25];
B13n=[A2 A29n A3 T6n A4];
B14n=[A26n A16 A12 A27n A16 A28n A11 A12 A15 A16 A29n A13 A13 A20 A10 A26n A25];
B15n=[A32 num2str(Sensnz) S1];
B16n=[A34 num2str(-Usnz) S1];

disp('***** FOR NORMAL ACCELERATION (Nz) *****');
disp(' ');
disp('/* Put Constraints On " " Vector Margins */');
disp(' ');
disp(A31);
disp(B15n);
disp(A33);
disp(' ');
disp('/* Put Constraints On Overshoot */');
disp(' ');
disp(B1n);
disp(' ');
disp('/* Put Constrains On Undershoot */');
disp(' ');
disp(B16n);
disp(' ');
disp('/* Put Constraints On Settling Time */');
disp(' ');
disp(B2n);
disp(B3n);
disp(' ');
disp('/* Request Zero Steady State Error*/');
disp(' ');
disp('Re_H[out][in](1,0) = 1;');
disp(' ');
disp(' OR');
disp(' ');

```

```

disp('/* Put Constraints On Steady State Error (if not zero) */');
disp(' ');
disp(B4n);
disp(B5n);
disp(' ');
disp('/* Put Constraint On Rise Time */');
disp(' ');
disp(B6n);
disp('0.8<=u_step[out][in](t);');
disp(' ');
disp('/* Put In Upper Constraint Curve (if desired) */');
disp(' ');
disp(B7n);
disp(B8n);
disp(' ');
disp(B9n);
disp(B10n);
disp(' ');
disp('/* Put In Lower Constraint Curve (if desired) */');
disp(' ');
disp(B11n);
disp(B12n);
disp(' ');
disp(B13n);
disp(B14n);
disp(' ');
disp(' ');
disp('*****');
disp(' ');
disp(' ');
Mpq=Mpq-1;
Mpnz=Mpnz-1;
disp('Here are the numbers to put into the "nzpst.m"');
disp('files to draw the time same time response curves you just created');
disp(' ');
disp('***** FOR NORMAL ACCELERATION (Nz) *****');
disp(' ');
Trnz
Mpnz
Usnz
Tsnz
pernz
Essnz
Tsnzst
Ynz
Tmznz
Tyznz
Tynznz
Ytynznz
disp('*****');
disp('*****');

```

end;

C.5 The "surf.m" File

```
% This program calculates the control surface deflection rate from the time
% history of the control surface

% To use, adjust plot axes so that you can ID the steepest slope of the
% control surface deflection time history. Use the cursor to pull off the
% magnitude and time of the two end points of the segment with the largest
% slope. The program will take the absolute value of the difference of the
% magnitudes and divide it by the time difference between the two to get the
% control surface deflection rate.

[t1,m1]=ginput(1);
[t2,m2]=ginput(1);

mdiff=abs(m2-m1);
tdiff=abs(t2-t1);

max_deflection_rate=mdiff/tdiff

% Set variables back to zero

t1=0; t2=0; m1=0; m2=0; mdiff=0; tdiff=0; max_deflection_rate=0;

end;
```

Bibliography

1. Anderson, Brian D.O., and John B. More. *Optimal Control Linear Quadratic Methods*. Prentice-Hall, Inc. 1990.
2. Barratt, Craig. "Two Mass Q Design Example" Stanford University, July 1987.
3. Boyd, Stephen, et al. "A New CAD Method and Associated Architectures for Linear Controllers," *IEEE Control Systems Magazine* (1989).
4. Boyd, Stephen P. *QDES User Commands* (302 Edition). Stanford University, 30 May 1987.
5. Boyd, Stephen P. and Craig H. Barratt. *Linear Control Design Limits Of Performance*. Prentice-Hall, Inc. 1991.
6. DeLaney, Joseph P. *Determining the Limits of Achievable Performance of an Air-to-Air Missile Using QDES*. MS thesis, Air Force Institute of Technology, 1992.
7. Doyle, John, D., Bruce A. Francis, and Allen R. Tannenbaum. *Feedback Control Theory*. Macmillan Publishing Co. 1992.
8. Erickson, Mark. *More Matlab Scripts for use with QDES*. Stanford University, 1989.
9. Etkin, Bernard. *Dynamics of Atmospheric Flight*. John Wiley & Sons, Inc., 1972.
10. Francis, Bruce A., *A Course In H_{∞} Control Theory*, Vol 88 in Lecture Notes in Control and Information Sciences, Springer-Verlag, Berlin, 1987.
11. Franklin, Gene F., J. David Powell, and Abbas Emami-Naeini. *Feedback Control of Dynamic Systems*. Addison-Wesley, 1991
12. Houpis, Constantine H. and Gary B. Lamont. *Digital Control Systems, Theory, Hardware, Software*. McGraw-Hill, Inc., 1992.
13. McRuer, Duane, Irving Ashkenas, and Dunstan Graham. *Aircraft Dynamics and Automatic Control*. Princeton University Press, 1973.
14. Pachter, Meir. Professor, Department of Electrical and Computer Engineering, Air Force Institute of Technology (AETC), Wright-Patterson AFB, OH. Personal Conversations. October 1993.

15. Ridgely, D. Brett. Lectures presented and personal notes taken during MECH 623, Advanced Functional Optimization. School of Engineering, Air Force Institute of Technology (AETC), Wright-Patterson AFB, OH, March 1993 - May 1993.

Form Approved
OMB No. 0704-0188

[illegible]

NSN 7540-01-280-5500