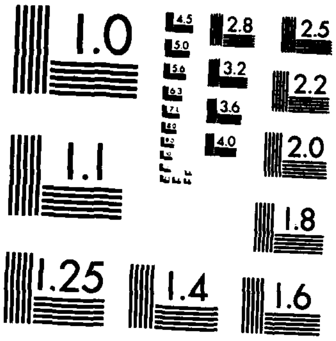
AD-A164 108 AN LQG UP-AND AWAY FLIGHT CONTROL DESIGN FOR THE STOL 1/3
F-15 AIRCRAFT(U) AIR FORCE INST OF TECH
WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING
UNCLASSIFIED R A HOUSTON DEC 85 AFIT/GE/ENG/85D-21 F/G 1/4 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD-A164 108

①

DTIC
ELECTE
FEB 1 3 1986
S D
D

AN LQG UP-AND-AWAY FLIGHT CONTROL
DESIGN FOR THE STOL F-15 AIRCRAFT

THESIS

Robert A. Houston
Second Lieutenant, USAF

AFIT/GE/ENG/85D-21

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

DTIC FILE COPY

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

86 2 11 125

86 2 11 125

①

AFIT/GE/ENG/85D-21

DTIC
S ELECTE D
FEB 1 3 1986
D

AN LQG UP-AND-AWAY FLIGHT CONTROL DESIGN FOR THE STOL F-15 AIRCRAFT

THESIS

Robert A. Houston
Second Lieutenant, USAF

AFIT/GE/ENG/85D-21

Approved for public release; distribution unlimited

AFIT/GE/ENG/85D-21

AN LQG UP-AND-AWAY FLIGHT CONTROL DESIGN FOR THE STOL F-15 AIRCRAFT

THESIS

Presented to the Faculty of the 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 Electrical Engineering

Robert A. Houston, B.S.E.E.
Second Lieutenant, USAF

December 1985

Approved for public release; distribution unlimited

Accesion For
NTIS CRA&I ☑
DTIC TAB ☐
Unannounced ☐
Justification ....
By ....
Distribution /
Availability Codes

Dist	Avail and / or Special
A-1	

QUALITY INSPECTED 3

Acknowledgements

Throughout the course of this thesis effort, many people have assisted in its completion. I would like to take this opportunity to express my gratitude to those who have contributed to the culmination of the research I have conducted during my graduate education.

First, I would like to express my gratitude to Captains Kevin Sheehan, Greg Gross, Bruce Acker, and Lieutenant Bruce Clough for the synergistic effect of our concurrent study of controlling the STOL F-15 aircraft. Our morning "coffee sessions" proved to be an invaluable forum for discussing problems and reflecting on possible approaches to solve those problems. Another person who has had a profound impact on my thesis effort is my advisor, Dr. Peter S. Maybeck. I have found Dr. Maybeck to be an "educator" in the truest sense of the word. His interest, patience, and guidance have been inspirational. Finally, I wish to express my deepest gratitude to my wife Olga for her constant support during the years I have spent pursuing my education.

— Robert Houston

Table of Contents

Page

Acknowledgements ii

List of Figures v

List of Tables ix

Abstract x

I. Introduction 1

- 1.1 Background 1
- 1.2 Problem 3
- 1.3 Scope 4
- 1.4 Assumptions 4
- 1.5 Sequence of Presentation 6

II. LQG Theoretical Development 7

- 2.1 Introduction 7
- 2.2 Synthesis of LQG Regulators 8
- 2.3 Synthesis of PI Controllers via LQG Methods 17
- 2.4 Model Following Controllers 27
- 2.5 Command Generator Tracker Synthesis Techniques 36
- 2.6 Summary 46

III. Modeling Considerations for the STOL F-15 . . . 48

- 3.1 Introduction 48
- 3.2 Modeling the STOL F-15 Aircraft . . . 50
- 3.3 Models for CGT Design and Performance Evaluation 54
- 3.4 The STOL F-15 Flight Envelope 60
- 3.5 Summary 63

IV. Robustness Enhancement Techniques 65

- 4.1 Introduction 65
- 4.2 Implicit Model Following 65
- 4.3 Loop Transfer Recovery 68
- 4.4 Summary 71

Page

V. Experimental Methods and Results 73

5.1 Introduction 73
5.2 Detailed Portrayal of Design Model . . 73
5.3 Truth Model Specification 76
5.4 Explicit Model Derivation 82
5.5 Implicit Model Derivation 85
5.6 Pitch Pointing Controller Design . . . 89
5.7 Robustness Analysis 108
5.8 Summary 120

VI. Conclusions and Recommendations 121

6.1 Introduction 121
6.2 Conclusions 121
6.3 Recommendations for Further Study . . 123

Appendix A: STOLCAT Program Listing 127

Appendix B: STOL F-15 Aerodynamic Data 152

Appendix C: ODEF15 168

Appendix D: Derivation of Nonlinear Thrust/ Nozzle Model 192

Appendix E: Basic Kalman Filtering Theory 197

Appendix F: Plotted Data for Section 5.7 199

Appendix G: Notes on LTR Technique for Application to LQG/PI/KF System 223

Bibliography 226

Vita 229

List of Figures

Figure Page

2.1 Position Form PI Control Law Based on Control Differences 26

2.2 Open-Loop Command Generator Tracker 40

2.3 A Block Diagram of the CGT/PI/KF Controller 45

3.1 STOL F-15 Flight Envelope 62

5.1 Eleven-State Truth Model 78

5.2 Comparison of Second Order and Third Order Actuator Frequency Responses 79

5.3 Nine-State Truth Model 81

5.4 Ideal Explicit Model Aircraft Responses . . . 86

5.5 Aircraft Response Without (Upper) and With (Lower) Actuators/Poorly Chosen Weighting Matrices 96

5.6 Aircraft Response Without (Upper) and With (Lower) Actuators/Well Chosen Weighting Matrices 97

5.7 Pitch Rate Channel Poor Implicit Model Weighting (Lower)/Well Chosen Implicit Model Weightings 98

5.8 Basic Aircraft Response 102

5.9 Aircraft Response with Actuators 103

5.10 Aircraft Response with Actuators, Position Limits, and Kalman Filter in the Loop . . . 106

5.11 Aircraft Response with Actuators, Position Limits, and Kalman Filter with the Loop/ Control Based on $\hat{\underline{x}}(t_i^-)$ 108

D.1 Diagram of Nozzle Deflection 192

Figure Page

F.1 Aircraft Response with Actuators, Position Limits, and Kalman Filter in the Loop/ Throttle Absolute Value Function 200

F.2 Aircraft Response with Actuators, Position Limits, and Kalman Filter in the Loop/ Throttle--A Linear Function of Stabilator Deflection 201

F.3 Aircraft Response with Actuators, Position Limits, and Kalman Filter in the Loop/ Control Based on $\hat{\underline{x}}(t_i^-)$/10 Percent Increase in $\underline{F}$ Matrix 202

F.4 Aircraft Response with Actuators, Position Limits, and Kalman Filter in the Loop/ Control Based on $\hat{\underline{x}}(t_i^-)$/25 Percent Increase in $\underline{F}$ Matrix 203

F.5 Aircraft Response with Actuators, Position Limits, and Kalman Filter in the Loop/Control Based on $\hat{\underline{x}}(t_i^-)$/100 Percent Increase in $\underline{F}$ Matrix 204

F.6 Aircraft Response with Actuators, Position Limits, and Kalman Filter in the Loop/ Control Based on $\hat{\underline{x}}(t_i^-)$/200 Percent Increase in $\underline{F}$ Matrix 205

F.7 Aircraft Response with Actuators/Free Floating Canard Failure 206

F.8 Aircraft Response with Actuators/Anti-Windup Compensation 207

F.9 Aircraft Response with Actuators and Kalman Filter in the Loop/Anti-Windup Compensation . 208

F.10 Aircraft Response with Actuators and Kalman Filter in the Loop/Control Based on $\underline{x}(t_i^-)$/ Anti-Windup Compensation 209

F.11 Aircraft Response with Actuators, Position Limits, and Kalman Filter in the Loop/ Control Based on $\hat{\underline{x}}(t_i^-)$/40 Fold Increase in Measurement Noise 210

Figure Page

F.12 Aircraft Response with Actuators, Position Limits, and Kalman Filter in the Loop/ Anti-Windup Compensation/40 Fold Increase in Measurement Noise 211

F.13 Aircraft Response with Actuators and Kalman Filter in the Loop/Control Based on $\hat{\underline{x}}(t_i^-)$/ Anti-Windup Compensation/40 Fold Increase in Measurement Noise 212

F.14 Aircraft Response with Actuators, Position Limits, and Kalman Filter in the Loop/ Control Based on $\hat{\underline{x}}(t_i^-)$/High Level Noise (q = 0.8) in Pitch Rate and Angle of Attack Channels 213

F.15 Aircraft Response with Actuators, Position Limits, and Kalman Filter in the Loop/ Control Based on $\hat{\underline{x}}(t_i^-)$/High Level Noise (q = 0.8) in Pitch Rate and Angle of Attack Channels/LTR Tuning with Loop Broken at the Input 214

F.16 Aircraft Response with Actuators, Position Limits, and Kalman Filter in the Loop/ Anti-Windup Compensation/C-Tuned 215

F.17. Aircraft Response with Actuators, Position Limits, and Kalman Filter in the Loop/ Control Based on $\hat{\underline{x}}(t_i^-)$/Anti-Windup Compensation/C-Tuned 216

F.18 Aircraft Response with Actuators, Position Limits, Rate Limits, and Kalman Filter in the Loop/Control Based on $\hat{\underline{x}}(t_i^-)$/C-Tuned/ Anti-Windup Compensation/200 Percent Increase in $\underline{F}$ Matrix 217

F.19 Aircraft Response with Actuators, Position Limits, Anti-Windup Compensation/Control Based on $\hat{\underline{x}}(t_i^-)$/40 Fold Increase in Measurement Noise/C-Tuned 218

F.20 Aircraft Response with Actuators, Position Limits, Anti-Windup Compensation/Control Based on $\hat{\underline{x}}(t_i^-)$/C-Tuned/16 Percent Increase in M_{δ_C} 219

Figure | Page

F.21 Aircraft Response with Actuators, Position Limits, Anti-Windup Compensation/Control Based on $\hat{\underline{x}}(t_i^-)$/C-Tuned/16 Percent Increase in M_{δ_S} 220

F.22. Aircraft Response with Actuators, Position Limits, Anti-Windup Compensation/Control Based on $\hat{\underline{x}}(t_i^-)$/C-Tuned/40 Percent Decrease in M_{δ_S} 221

F.23 Aircraft Response with Actuators, Position Limits, Anti-Windup Compensation/Control Based on $\hat{\underline{x}}(t_i^-)$/C-Tuned/100 Percent Decrease in M_{δ_C} 222

List of Tables

Table Page

5.1 Effect of C-Tuning Filter 118

AFIT/GE/ENG/85D-21

Abstract

A robust controller for the STOL F-15 aircraft is developed using the LQG/LTR (linear system model, quadratic cost, gaussian models of uncertainty used for controller synthesis, with loop transfer recovery techniques of tuning the filter in the loop for control robustness enhancement) methods. Full state feedback controllers are synthesized using CGT/PI (Command Generator Tracking feedforward compensator to provide direct incorporation of flying qualities into the design process, with proportional plus integral feedback control) synthesis, using implicit model following techniques to improve full state robustness characteristics. Finally, a Kalman filter is used to replace the unrealistic assumption of full state availability with estimated states, using a LTR scheme to recover as much full state robustness characteristics as possible.

AN LQG UP-AND-AWAY FLIGHT CONTROL DESIGN
FOR THE STOL F-15 AIRCRAFT

I. Introduction

1.1 Background

The initial approach used to design aircraft control systems involved a mixture of classical single-input/single-output (SISO) control design techniques and engineering insight. In the past this has been effective, providing that the designer had sufficient intuition and was willing to carry the design process through enough single loop iterations to meet specifications. However, as airframes became more complex and control surfaces more numerous, this method became unwieldy, creating a need for multiple-input/multiple-output (MIMO) controller synthesis techniques. One school of thought emerging from this need is the extension of classical techniques to the MIMO case, thereby taking advantage of frequency domain theory developed by Bode, Nyquist, and Hurwitz (24). A second approach to MIMO controller design involves time domain techniques, which, assuming that the plant can be modeled as linear, can be readily coupled with linear system theory to yield a procedure which can be implemented on digital computers. Although both of the previously mentioned MIMO design philosophies have adamant supporters, they are really two sides of the same coin, so neither should be abandoned for the other.

A design method which incorporates both the ease of implementation associated with time domain techniques and the invaluable stability information obtained from frequency domain analysis is the Linear-Quadratic-Gaussian (LQG) (11; 12) approach coupled with Loop Transfer Recovery (LTR) (5; 9; 13; 22). A more specific formulation using LQG synthesis methods, which is especially attractive for flight control applications, is Command Generator Tracking (CGT) combined with a regulator (R) designed via LQG methods; this is often denoted as a CGT/R design (12:151-166). This method yields a controller capable of incorporating handling qualities directly into the design process, while rejecting specified disturbance inputs. A more useful formulation of the CGT technique would include a proportional plus integral (PI) controller in a closed loop law incorporating a CGT precompensator in order to achieve type 1 characteristics; this is designated as a CGT/PI controller. However, as will be discussed later, problems have been encountered in the design of the full CGT/PI controller when a filter is embedded in the control loop and LTR techniques are applied to achieve a robust controller. Therefore, in some applications, CGT/R and CGT/PI designs might both be pursued by means of LQG/LTR techniques.

1.2 Problem

The objective in this study is to design a robust tracker controller for the STOL F-15 aircraft using digital Command Generator Tracking to incorporate specified aircraft handling qualities into the design process. This controller will be designed via LQ methods using implicit model following to enhance robustness (16). Kalman filtering techniques will be used to replace full state feedback with state estimates. The robust tracker controller will be designed to maintain aircraft stability (and as desirable performance as possible) in the face of large parameter variations such as control actuator failure, mismodeled actuator dynamics, and actuator saturations. This study will also address the robustness degradation due to reducing the order of the aircraft model and also due to operating the aircraft at a different point in its operational envelope than that used for controller design. The feasibility of using the LTR tuning method developed by Doyle and Stein (5; 22) in a CGT/PI controller will also be investigated.

It is not an objective of this thesis to engage in a debate over the adequacy of existing MIMO controller synthesis techniques; however, several methods of addressing the same STOL F-15 problem will be carried out concurrently with this study (1; 20), in the hope that a comparison of the obtained results will prove useful in constructing logical approaches to similar problems.

1.3 Scope

This thesis effort will encompass the LQG/LTR design and analysis of a CGT/PI/KF longitudinal flight control system for the STOL F-15 aircraft. Specifically, the added aerodynamic surfaces of the STOL version of the F-15 are exploited to accomplish a pitch pointing maneuver at 4 points within the aircraft's combat envelope. Further, a complete robustness enhancement/analysis of the controller designed at the "nominal" flight condition of Mach 0.9 at 20,000 feet will be carried out. Such topics as plant variations, control derivative variations, noise corruption of plant states, throttle gain scheduling, control surface failures, measurement noise corruption, and impact of LTR tuning on a PI controller design will be addressed. A nonlinear analysis of the controller design, specifically admitting both position and rate limit saturations of actuators, will be carried out using Monte Carlo analysis software designed as a part of this thesis effort.

1.4 Assumptions

The physical description of an aircraft's dynamics can be represented by a set of nonlinear differential equations. Although these equations present a very accurate portrayal of the true aircraft, they fail to be useful in designing control systems due to the tremendous computational loading incurred for a relatively small improvement in performance compared to controllers based on linearized aircraft equations of motion. Therefore, for the purposes of the designs accomplished in this study, perturbation equations, linearized about specific trim conditions, will be assumed adequate for the design models representing the STOL F-15 aircraft's equations of motion. A further assumption which will be made is that all noise corruptions in both the system and measurement models will be adequately modeled as white and Gaussian. Although a true white noise would contain infinite power, the assumption of white noise physically implies that colored, i.e. time correlated, noises affecting the system will appear white over the bandpass of interest in the aircraft dynamics (11). The assumption of Gaussianess can be justified by invoking the central limit theorem of probability theory (11; 18). The above mentioned assumptions are considered major in that they are a significant influence on the entire thesis. Other assumptions made throughout the remainder of this study are of less global impact; therefore, these will be presented only as needed for specific development. Although wind buffet rejection in the CGT controller is not addressed, this could be accomplished, if desired, using the same methodology.

1.5 Sequence of Presentation

The material contained in this thesis is presented in such a manner as first to build a theoretical framework and then to use the developed control synthesis techniques in a specific application. Chapter II presents CGT/R and CGT/PI controller forms and discusses the advantages and disadvantages of each. Chapter III introduces the equations of motion and the model for the STOL F-15 aircraft. Chapter IV is concerned with stability and robustness enhancement of the controllers discussed in Chapter II. Chapter V is the culmination of the preceding three chapters; in this chapter controller designs are carried out and evaluated using existing and designed computer aided design software (7; 16; 17). Finally, Chapter VI presents conclusions of the research conducted and recommendations for further study.

II. LQG Theoretical Development

2.1 Introduction

This chapter is intended to serve as an introduction to the theoretical foundation upon which the controller design of Chapter V is based. It is assumed that the reader has had an upper division undergraduate or a first year graduate level course in classical control theory (4) and has a knowledge of modern optimal control which includes basic LQG techniques and Kalman filtering (11; 13). Any further material considered substantive to a full understanding of Chapter V will be presented herein.

The remaining sections contained in this chapter develop four interrelated concepts. First, a derivation of LQG regulators will be presented. Shortcomings of this form of controller will be discussed as a motivation for the subsequent section on proportional plus integral controllers designed via LQG methods. Section 2.4 is intended to give the reader a conceptual introduction to model following design techniques. Thus, it yields a coherent transition into Section 2.5 which introduces Command Generator Tracking theory and its closed loop application with both regulator and PI controller forms.

As a final statement before embarking on the theoretical sections of this chapter, the reader should be aware that the following derivations, although functionally complete, are not intended to be mathematically rigorous. Those who desire more theoretical detail are directed to References 11 and 12. However, this thesis is directed toward engineering application, and consequently derivations will be intended for the practicing engineer.

2.2 Synthesis of LQG Regulators

This section presents regulator design via LQG synthesis techniques. The following derivation is taken primarily from Reference 13 with exceptions as noted.

Consider the linear, discrete time, vector valued, stochastic difference equation

$$\underline{x}(t_{i+1}) = \underline{\Phi}(t_{i+1},t_i)\underline{x}(t_i) + \underline{B}_d(t_i)\underline{u}(t_i) + \underline{G}_d(t_i)\underline{w}_d(t_i) \tag{2-1}$$

where

$\underline{x}$ is an n-dimensional state vector,

$\underline{\Phi}$ is an nxn state transition matrix,

$\underline{u}$ is an r-dimensional control deterministic control input,

$\underline{B}$ is an nxr input matrix, and

$\underline{w}_d$ is a discrete-time Gaussian noise sequence completely characterized by

$$E\{\underline{w}_d(t_i)\} = \underline{0} \tag{2-2}$$

and

$$E\{\underline{w}_d(t_i)\underline{w}_d^T(t_j)\} = \underline{Q}_d\,\delta_{ij} \tag{2-3}$$

where $\underline{Q}_d$ is the covariance of the noise sequence, and δ_{ij} is the Kronecker delta function.

Although a discrete system of the above form could arise naturally, in flight control applications, and many other applications as well, this model would be an equivalent discrete-time representation for an underlying continuous system (11:133,192). A stochastic differential equation which could be used to describe such an underlying continuous system would be

$$\dot{\underline{x}}(t) = \underline{F}(t)\underline{x}(t) + \underline{B}(t)\underline{u}(t) + \underline{G}(t)\underline{w}(t) \tag{2-4}$$

where w(t) is zero-mean white Gaussian noise with a covariance

$$E\{\underline{w}(t)\underline{w}^T(t+\tau)\} = \underline{Q}\delta(\tau) \tag{2-5}$$

where $\delta(\tau)$ is the Dirac delta function.

The discrete-time input matrix $\underline{B}_d$ and the noise strength $\underline{Q}_d$ of Equation (2-1) can be derived from the underlying continuous system model parameters by the relations (11):

$$\underline{B}_d(t_i) = \int_{t_i}^{t_{i+1}} \underline{\Phi}(t_{i+1},\tau)\underline{B}(\tau)\,d\tau \tag{2-6}$$

and

$$\underline{Q}_d(t_i) = \int_{t_i}^{t_{i+1}} \underline{\Phi}(t_{i+1},\tau)\underline{G}(\tau)\underline{Q}(\tau)\underline{G}^T(\tau)\underline{\Phi}^T(t_{i+1},\tau)\,d\tau \tag{2-7}$$

respectively.

For the LQG regulator it is desired that certain linear combinations of the state variables, designated control variables, be regulated to zero. In the case of flight controller design, linearized models are used, and the perturbation control variables are linear combinations of perturbation states; regulating these perturbation variables to zero is equivalent to driving the aircraft back to the specific trim condition used for the basis of the linearized equations (6:154,165). These variables are represented by the p-dimensional vector $\underline{y}_c$ defined by

$$\underline{y}_c(t) = \underline{C}(t)\underline{x}(t) \tag{2-8}$$

where $\underline{C}$ is a pxn output matrix. A direct transmission term ($\underline{D}(t)\underline{u}(t)$) in equation (2-8) would create cross-coupling between the states and the control inputs in the quadratic cost function; however, no generality is lost due to the omission of this term, as will be shown subsequently.

As an initial attempt to produce an optimal control function $\underline{u}(t)$, consider minimization of a cost function of the form

$$J = E\left\{\sum_{i=0}^{N} \frac{1}{2}(\underline{y}_c^T(t_i)\underline{Y}(t_i)\underline{y}_c(t_i) + \underline{u}^T(t_i)\underline{U}(t_i)\underline{u}(t_i)) + \frac{1}{2}(\underline{y}_c^T(t_{N+1})\underline{Y}_f\underline{y}_c(t_{N+1}))\right\} \tag{2-9}$$

where $\underline{Y}(t_i)$ and $\underline{Y}_f$ are pxp symmetric weighting matrices on the control variables at time t_i and the final time t_f, respectively. These matrices are both chosen to be positive definite under the assumption that there should exist a cost associated with any control variable deviation from zero, or deviation from trim conditions for aircraft perturbation variables. $\underline{U}(t_i)$ is an rxr symmetric positive definite weighting matrix on the control inputs; in this case the weighting is chosen positive definite in order to avoid a controller which attempts to expend infinite control energy.

The magnitudes of $\underline{Y}$ and $\underline{U}$ are chosen relative to one another. If $\underline{Y}$ is chosen larger, tighter control of the state trajectories is achieved. If $\underline{U}$ is chosen larger, less control "energy" can be expended.

The cost minimizing control function $\underline{u}^*(t_i)$ can be shown to be a linear function of the system states

$$\underline{u}^*(t_i) = -\underline{G}_c^*(t_i)\underline{x}(t_i) \qquad (2\text{-}10)$$

Note that Equation (2-10) makes the unrealistic assumption that all of the system state variables will be accessible. A more practical assumption would incorporate a measurement model consisting of incomplete, noise corrupted measurements as follows (11:203,225):

$$\underline{z}(t_i) = \underline{H}(t_i)\underline{x}(t_i) + \underline{v}(t_i) \qquad (2\text{-}11)$$

In the above equation $\underline{v}(t_i)$ is a zero-mean Gaussian m-dimensional noise sequence with covariance $\underline{R}(t_i)$ and $\underline{H}$ is an mxn measurement matrix. It is assumed that $\underline{v}$ is independent of the discrete dynamics model noise $\underline{w}_d(t_i)$ of Equation (2-1) (11:203,225).

In light of Equation (2-11), the optimal control function described by Equation (2-10) would seem to be little more than a mathematical abstraction, void of any practical engineering application; however, this is not the case. Under the LQG assumptions, certainty equivalence may be invoked (12:24,45); thus, the optimal control feedback weighting matrix, $\underline{G}_c^*$, for a nondeterministic system with only partial, noise-corrupted state measurements available, can be derived under the assumption of a deterministic system model with access to all states. Once derived, this optimal gain may be cascaded with a Kalman filter in order to produce an optimal stochastic control of the following form:

$$\underline{u}^*(t_i) = -\underline{G}_c^*(t_i)\hat{\underline{x}}(t_i^+) \tag{2-12}$$

The optimal gain function, $\underline{u}^*$, can be obtained by solving for the optimal gain $\underline{G}_c^*(t_i)$ as a function of the backward Riccati recursion difference equation for the matrix $\underline{K}_c$:

$$\underline{G}_c^*(t_i) = (\underline{U}(t_i) + \underline{B}_d^T(t_i)\underline{K}_c(t_{i+1})\underline{B}_d(t_i))^{-1}$$

$$\cdot\ (\underline{B}_d^T(t_i)\underline{K}_c(t_{i+1})\underline{\Phi}(t_{i+1},t_i)) \qquad (2\text{-}13)$$

where

$$\underline{K}_c(t_i) = \underline{X}(t_i) + \underline{\Phi}^T(t_{i+1},t_i)\underline{K}_c(t_{i+1})$$

$$\cdot\ (\underline{\Phi}(t_{i+1},t_i) - \underline{B}_d(t_i)\underline{G}_c^*(t_i)) \qquad (2\text{-}14)$$

(Note that $\underline{K}_c$ has been constructed in order to provide tractability in the solution for the optimal gain matrix and bears no relation to the Kalman filtering gain $\underline{K}$.)

Since Equation (2-14) is a backward running equation, the solution must be generated from a terminal condition defined as

$$\underline{K}_c(t_{N+1}) = \underline{X}_f = \underline{C}^T(t_f)\underline{Y}_f\underline{C}(t_f) \qquad (2\text{-}15)$$

This completes the formulation of the simple LQG regulator; however, an important problem which has not yet been addressed is the need to exert effective sample data control over a continuous system, not only at each sample time, but between sample times as well.

Consider a flight control application in which a digital controller is implemented to maintain stable flight in a statically unstable flight mode. If the sampling rate of the controller is less than approximately five times the Nyquist rate, the aircraft could go unstable between sample times. For flight controller design, the initial and terminal transients of an LQG controller for a time-invariant system subjected to stationary noises, and designed using constant weighting matrices, can be considered negligible in comparison to the length of time that the aircraft is in steady state conditions. Consequently, the controller gains $\underline{G}_c^*$ matrices (as well as any Kalman filter gains which might be associated with the system) become constants.

An appropriate quadratic cost function which incorporates a continuous weighting of the system states and control inputs is

$$J_c = E\left\{\frac{1}{2}\underline{x}^T(t_{N+1})\underline{X}_f\underline{x}(t_{N+1}) + \sum_{i=0}^{N}\left(\int_{t_i}^{t_{i+1}} \frac{1}{2}(\underline{x}^T(t)\underline{W}_{xx}(t)\underline{x}(t) + \underline{u}^T(t)\underline{W}_{uu}(t)\underline{u}(t) + 2\underline{x}^T(t)\underline{W}_{xu}\underline{u}(t))dt\right)\right\} \qquad (2\text{-}16)$$

where $\underline{W}_{xx}(t)$ is positive semidefinite, so some states may not have a cost associated with them if desired, and $\underline{W}_{uu}(t)$ is positive definite for all t for the same reason as discussed for $\underline{U}(t_i)$. Upon dividing this function into n+1 control intervals over (t_o, t_{N+1}), the cost function becomes

$$J_c = E\left\{\frac{1}{2}\underline{x}^T(t_{N+1})\underline{X}_f\underline{x}(t_{N+1}) + \sum_{i=0}^{N}\frac{1}{2}[\underline{x}^T(t_i)\underline{X}(t_i)\underline{x}(t_i) + \underline{u}^T(t_i)\underline{U}(t_i)\underline{u}(t_i) + 2\underline{x}^T(t_i)\underline{S}(t_i)\underline{u}(t_i)]\right\} \qquad (2\text{-}17)$$

where

$$\underline{X}(t_i) = \int_{t_i}^{t_{i+1}} \underline{\Phi}^T(t,t_i)\underline{W}_{xx}(t)\underline{\Phi}(t,t_i)dt \tag{2-18}$$

$$\underline{U}(t_i) = \int_{t_i}^{t_{i+1}} [\overline{\underline{B}}^T(t,t_i)\underline{W}_{xx}(t)\overline{\underline{B}}(t,t_i) + \underline{W}_{uu}(t)$$

$$+ \overline{\underline{B}}^T(t,t_i)\underline{W}_{xu}(t) + \underline{W}_{xu}^T\overline{\underline{B}}(t,t_i)]dt \tag{2-19}$$

$$\underline{S}(t_i) = \int_{t_i}^{t_{i+1}} [\underline{\Phi}^T(t,t_i)\underline{W}_{xx}(t)\overline{\underline{B}}(t,t_i)+\underline{\Phi}^T(t,t_i)\underline{W}_{xu}(t)]dt \tag{2-20}$$

and

$$\overline{\underline{B}}(t,t_i) = \int_{t_i}^{t_i} \underline{\Phi}(t,\tau)\underline{B}(\tau)d\tau \tag{2-21}$$

Thus the desire to exert control between sample times has generated cross terms in the discrete quadratic cost function. Heuristically, Equation (2-20) can be interpreted as an indication of how coupling between the states and the control inputs can arise in the cost function. The second term on the right side of Equation (2-20) is a function of $\underline{W}_{xu}$, which arises from natural coupling between $\underline{x}$ and $\underline{u}$ in the continuous-time setting, such as the desire to control a deflection rate of a control surface, yielding a desire to put a quadratic weighting on the derivative of the system state vector $\underline{x}(t)$. The first term on the right side of Equation (2-20) is a function of $\underline{W}_{xx}$; this indicates that a cross coupling in the discrete cost function can be generated by the desire to exert control between the sample times even if no natural coupling exists in the continuous-time system.

The previous discussion motivates the need for cross coupling terms in the discrete quadratic cost function associated with the LQG regulator. Since this cross coupling exists, there is no need to extend Equation (2-8) to include a direct transmission term as this would not modify the Equation (2-16) (see page 81, Reference 12, for a more detailed treatment of cross coupling as a consequence of a direct feedthrough of the control input).

The optimal gain matrix Equations (2-13) and (2-14) can be modified to account for the addition of cross coupling terms between $\underline{x}$ and $\underline{u}$ in the following manner:

$$\underline{G}_c^*(t_i) = (\underline{U}(t_i) + \underline{B}_d^T(t_i)\underline{K}_c(t_{i+1})\underline{B}_d(t_i))^{-1} \cdot (\underline{B}_d(t_i)^T\underline{K}_c(t_{i+1})\underline{\Phi}(t_{i+1},t_i) + \underline{S}^T(t_i)) \quad (2\text{-}22)$$

where $\underline{K}_c$ is now defined as the solution to:

$$\underline{K}_c(t_i) = \underline{X}(t_i) + \underline{\Phi}^T(t_{i+1},t_i)\underline{K}_c(t_{i+1})\underline{\Phi}(t_{i+1},t_i) - (\underline{B}_d^T(t_i)\underline{K}_c(t_{i+1})\underline{\Phi}(t_{i+1},t_i) + \underline{S}^T(t_i))^T\underline{G}_c^*(t_i) \quad (2\text{-}23)$$

The LQG stochastic regulator described in this section has been shown to possess desirable qualities such as providing control between sample times and allowing full state feedback to be replaced with Kalman filter estimates. However, it suffers from the drawback of displaying type zero characteristics (4:199-201). The desire to compensate for modeling errors, reject constant unmodeled disturbances, and achieve zero steady state tracking error, motivates a design which displays type 1 behavior. Such a controller is the proportional plus integral design developed in Section 2.3.

2.3 Synthesis of PI Controllers via LQG Methods

As stated in the previous section, the desire to obtain a controller which exhibits type 1 characteristics motivates investigation of proportional plus integral forms. In this section the PI controller of classical control theory is derived using modern methods under LQG assumptions. The application of LQG synthesis techniques to the design of PI controllers allows for systematic extension from SISO to MIMO systems (particularly for the proper evaluation of cross-coupling gains in a MIMO PI controller), and therefore, greater flexibility in flight control design problems. The LQG design of PI controllers is further motivated by other important factors such as iterative design capability, ease of off-diagonal weighting between integral and proportional channels, stability of LQ designs, and robustness enhancement through both LTR and implicit model following techniques. Again, unless stated otherwise the development of this section is taken from Reference 12.

Before embarking on the mathematical derivation of the PI controller, it is in order to discuss some of the motivations for implementing this particular type of controller structure. Consider a controller driven by an error signal, which in turn generates a system control input. It is desirable to structure this controller in such a manner as to maintain control of the state trajectories of the system even in the event that the input to the controller itself is driven to zero (13); e.g., if the tracking error is zero, under a particular set of equilibrium conditions, one still wants the controller to produce the control necessary to keep the system at that desirable equilibrium condition. The PI structure is able to accomplish this task due to the "integral action" which is not inherent in simple regulator schemes (13). A second advantage gained by PI forms over simple regulator structures is the ability to reject constant unmodeled disturbances, thereby improving steady state performance of the system (13). A conclusion that can be drawn from the foregoing discussion is that, while not absolutely necessary for flight control applications, the PI controller structure can be a significant improvement over simple regulator forms.

Consider a nominal control $\underline{u}_o$ needed to hold a deterministic time invariant system of the form

$$\underline{x}(t_{i+1}) = \underline{\Phi x}(t_i) + \underline{B}_d\underline{u}(t_i) \tag{2-24}$$

in an equilibrium condition. The nominal control could represent the control necessary to maintain an aircraft in a specified maneuver or trim condition.

From the above definition of $\underline{u}_o$, the nominal state trajectory $\underline{x}_o$ can be written as

$$\underline{x}_o = \underline{\Phi x}_o + \underline{B}_d\underline{u}_o \tag{2-25}$$

and a desired set of control variables are defined by

$$\underline{y}_d = \underline{Cx}_o + \underline{D}_y\underline{u}_o \tag{2-26}$$

(Note that the direct feedthrough matrix $\underline{D}_y$ is explicitly included here. Because the derivation of the PI controller involves more than just a simple LQ design, this term cannot be removed from Equation (2-26) by embedding it in cost function cross terms as was done in the derivation of simple regulator forms.)

In matrix form, Equations (2-25) and (2-26) become

$$\begin{bmatrix} \underline{\Phi} - \underline{I} & \underline{B}_d \\ \underline{C} & \underline{D}_y \end{bmatrix} \begin{bmatrix} \underline{x}_o \\ \underline{u}_o \end{bmatrix} = \begin{bmatrix} \underline{0} \\ \underline{y}_o \end{bmatrix} \tag{2-27}$$

This matrix equation can be used to solve the $\underline{x}_o$ and $\underline{u}_o$ by

$$\begin{bmatrix} \underline{x}_o \\ \underline{u}_o \end{bmatrix} = \begin{bmatrix} \underline{\Phi} - \underline{I} & \underline{B}_d \\ \underline{C} & \underline{D}_y \end{bmatrix}^{-1} \begin{bmatrix} \underline{0} \\ \underline{y}_d \end{bmatrix} \tag{2-28}$$

where the augmented matrix inverse can be partitioned as:

$$\begin{bmatrix} \underline{\Phi} - \underline{I} & \underline{B}_d \\ \underline{C} & \underline{D}_y \end{bmatrix}^{-1} = \begin{bmatrix} \underline{\Pi}_{11} & \underline{\Pi}_{12} \\ \underline{\Pi}_{21} & \underline{\Pi}_{22} \end{bmatrix} \tag{2-29}$$

For cases where the matrix on the left side of Equation (2-29) is not square, pseudoinverse techniques may be used to yield a solution in some cases (12:124; 22:142); however, for the purposes of this development it is assumed that the number of controls is equal to the number of controlled outputs and that the matrix in question is in fact invertible, so pseudoinverse forms will not be needed. With the nominal state trajectory and control defined by Equation (2-28), it is now possible to define the following set of perturbation variables:

$$\underline{\delta x}(t_i) = \underline{x}(t_i) - \underline{x}_o = \underline{x}(t_i) - \underline{\Pi}_{12}\underline{y}_d \tag{2-30}$$

$$\underline{\delta u}(t_i) = \underline{u}(t_i) - \underline{u}_o = \underline{u}(t_i) = \underline{\Pi}_{22}\underline{y}_d \tag{2-31}$$

and

$$\underline{\delta y}_c(t_i) = \underline{y}_c(t_i) - \underline{y}_d \tag{2-32}$$

As discussed in Section 2.2, it is desired to regulate these perturbation variables to zero; however, it is further desired that this be accomplished with a controller which incorporates integral action into its design. The derivation to follow will develop such a controller based upon pseudorates (13).

The difference in the control perturbation state over, $\underline{\delta u}$, one sample time can be expressed, using Equation (2-31), as

$$\underline{\delta u}(t_{i+1}) - \underline{\delta u}(t_i) = (\underline{u}(t_{i+1}) - \underline{u}_o) - (\underline{u}(t_i) - \underline{u}_o) \tag{2-33}$$

Therefore, $\underline{\delta u}(t_{i+1})$ may be expressed as

$$\underline{\delta u}(t_{i+1}) = \underline{\delta u}(t_i) + (\underline{u}(t_{i+1}) - \underline{u}(t_i)) \tag{2-34}$$

where the second term on the right side of Equation (3-34) can be interpreted as an Euler or tangent line integration (13:68,81) as shown in Equation (2-35).

$$\underline{\Delta u}(t_i) = (\underline{u}(t_{i+1}) - \underline{u}(t_i)) \cong \Delta t\dot{\underline{u}}(t_i) \tag{2-35}$$

From Equations (2-34) and (2-35) $\underline{\delta u}$ may be defined as a new state variable and augmented to Equation (2-24) to form the new state space model:

$$\begin{bmatrix} \underline{\delta x}(t_{i+1}) \\ \underline{\delta u}(t_{i+1}) \end{bmatrix} = \begin{bmatrix} \underline{\Phi} & \underline{B}_d \\ \underline{0} & \underline{I} \end{bmatrix} \begin{bmatrix} \underline{\delta x}(t_i) \\ \underline{\delta u}(t_i) \end{bmatrix} + \begin{bmatrix} \underline{0} \\ \underline{I} \end{bmatrix} \underline{\delta u}(t_i) \qquad (2\text{-}36)$$

where the pseudorate $\underline{\delta u}$ now forms the control input to the augmented system model.

The quadratic cost function which is minimized in order to yield an optimal deterministic control for the system of Equation (2-36) is

$$J = \sum_{i=-1}^{N} \left\{ \begin{bmatrix} \underline{\delta x}(t_i) \\ \underline{\delta u}(t_i) \\ \underline{\Delta u}(t_i) \end{bmatrix}^T \begin{bmatrix} \underline{X}_{11} & \underline{X}_{12} & \underline{S}_1 \\ \underline{X}_{12}^T & \underline{X}_{22} & \underline{S}_2 \\ \underline{S}_1^T & \underline{S}_2^T & \underline{U} \end{bmatrix} \begin{bmatrix} \underline{\delta x}(t_i) \\ \underline{\delta u}(t_i) \\ \underline{\Delta u}(t_i) \end{bmatrix} \right\}$$

$$+ \frac{1}{2} \begin{bmatrix} \underline{\delta x}(t_{N+1}) \\ \underline{\delta u}(t_{N+1}) \end{bmatrix}^T \begin{bmatrix} \underline{X}_{f11} & \underline{0} \\ \underline{0} & \underline{0} \end{bmatrix} \begin{bmatrix} \underline{\delta x}(t_{N+1}) \\ \underline{\delta u}(t_{N+1}) \end{bmatrix} \qquad (2\text{-}37)$$

Note that the lower limit on the summation in Equation (2-37) starts at -1 instead of 0. This change in the usual quadratic cost notation is motivated by the need to control the initial condition of the system in order to achieve good initial transient responses through placing a weight on $\underline{u}(t_{-1})$ (12:142). The $\underline{X}_{11}$ term in Equation (2-37) provides a cost weighting on the state trajectory deviation from nominal and $\underline{X}_{22}$ weights control magnitudes, both of which could have been accomplished by the regulator formulation of the previous section. However, the $\underline{U}$ term is a weighting on control differences, which under the approximation of Equation (2-35) can be considered a weighting on control rates. The cross terms $\underline{X}_{12}$, $\underline{S}_1$, and $\underline{S}_2$ can arise through natural coupling or through discretization of a continuous cost function as discussed in the simple regulator case of Section 2.2.

Solving for the optimal control function based upon Equation (2-37), and restricting attention to cost functions which allow the terminal time to approach infinity; yields a constant gain feedback control law of the form

$$\underline{\Delta u}(t_i) = -(\underline{G}^*_{c1} \mid \underline{G}^*_{c2}) \begin{bmatrix} \underline{\delta x}(t_i) \\ \underline{\delta u}(t_i) \end{bmatrix} \qquad (2\text{-}38)$$

where the gains $\underline{G}^*_{c1}$ and $\underline{G}^*_{c2}$ can be solved for in the same manner presented in Section 2.2.

Although Equation (2-38) represents an optimal perturbation regulator with a capability to regulate both control magnitudes and control rates, it does not achieve the desired objective of attaining type 1 system response since it lacks integral action. This deficiency will now be addressed.

The desired form of the PI controller based upon Equations (2-35) and (2-38) should be an optimal control signal constructed in terms of the system perturbation states of the following form:

$$\underline{\delta u}^*(t_{i+1}) = \underline{\delta u}^*(t_i) - \underline{K}_x(\underline{\delta x}(t_{i+1}) - \underline{\delta x}(t_i))$$

$$- \underline{K}_z(\underline{C}\,\underline{\delta x}(t_i) + \underline{D}_y\,\underline{\delta u}(t_i)) \qquad (2\text{-}39)$$

where $\underline{K}_x$ and $\underline{K}_z$ are constant gain matrices. The top partition of Equation (2-36) can be written as

$$\underline{\delta x}(t_{i+1}) = \underline{\delta x}(t_i) + (\underline{\Phi}-\underline{I})\underline{\delta x}(t_i) + \underline{B}_d\underline{\delta u}(t_i) \qquad (2\text{-}40)$$

Inserting Equation (2-40) into (2-39) and writing in matrix form yields

$$\underline{\delta u}^*(t_{i+1}) - \underline{\delta u}^*(t_i) = -(\underline{K}_x\underline{K}_z)\begin{bmatrix}\underline{\Phi}-\text{I} & \underline{B}_d \\ \underline{C} & \underline{D}_y\end{bmatrix}\begin{bmatrix}\underline{\delta x}(t_i) \\ \underline{\delta u}(t_i)\end{bmatrix} \qquad (2\text{-}41)$$

By setting the right sides of Equations (2-39) and (2-41) equal, the gains $\underline{K}_x$ and $\underline{K}_z$ may be evaluated as

$$\underline{K}_x = \underline{G}^*_{c1}\,\underline{\Pi}_{11} + \underline{G}^*_{c2}\,\underline{\Pi}_{21} \qquad (2\text{-}42)$$

and

$$\underline{K}_z = \underline{G}^*_{c1}\,\underline{\Pi}_{12} + \underline{G}^*_{c2}\,\underline{\Pi}_{22} \qquad (2\text{-}43)$$

Thus, once $\underline{G}^*_c$ is established, $\underline{K}_x$ and $\underline{K}_z$ may be computed.

Under the assumption that $\underline{y}_d$ is piecewise constant and varies slowly enough to allow any system transients to damp out sufficiently between changes, then the PI control law becomes

$$\underline{u}^*(t_i) = \underline{u}^*(t_{i-1}) - \underline{K}_x(\underline{x}(t_i) - \underline{x}(t_{i-1})) + \underline{K}_z(\underline{y}_d(t_i) - \underline{y}_c(t_{i-1})) \quad (2\text{-}44)$$

A diagram of this PI controller form is shown in Figure 2.1 on the next page.

This represents the final form of the pseudorate PI controller derived using LQG methods. This controller achieves type 1 characteristics; therefore, it will be able to track the desired nominal trajectories with zero mean error despite imperfect models or constant unknown disturbances such as steady cross winds.

At the beginning of this section, two assumptions were made which will now be justified. First, the system was assumed to be linear and time invariant. This assumption may seem restrictive. However, one of the motivations of linearizing aircraft equations of motion about a nominal flight condition is to produce a time invariant system model (6) for ease of controller synthesis; therefore, for the purposes of aircraft control system design, this is considered valid. Second, the original system in Equation (2-24) was unrealistically assumed to be deterministic with full state accessibility. However, under the LQG

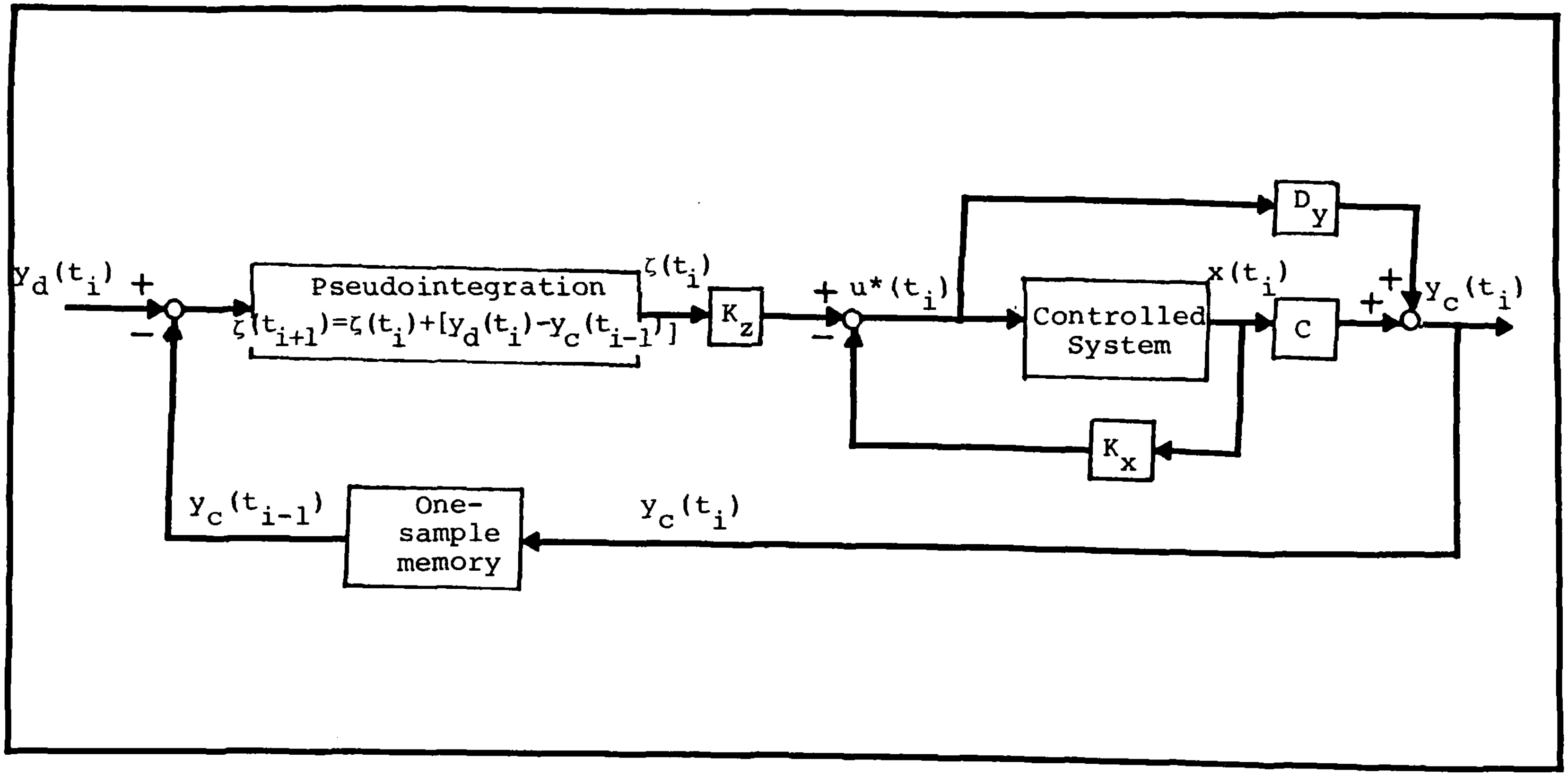


Fig. 2.1. Position Form PI Control Law Based on Control Differences

assumptions, certainty equivalence may be invoked if only partial, noise-corrupted measurements are available, as in the case of the regulator design of the previous section, and a Kalman filter may be introduced into the loop to provide estimates of the system states.

2.4 Model Following Controllers

Up to this point in the development of the theory to be used in Chapter V, the controllers presented have been intended primarily for the purpose of regulating a set of control variables to zero or some other specified trim condition. Another objective of the final controller to be implemented in this study is the ability for forcing the controlled variables to behave like those of a predetermined model (7:10-13). In terms of classical control theory, this method can be equated to designing to meet specifications such as damping ratio, peak overshoot, rise time, etc. (4), or in the context of flight control this would include the incorporation of handling qualities (6:490-526) specifically into the design process. A technique which accomplishes this task is aptly referred to as "model following," and can be divided into two distinct categories, both of which will be presented in this section.

Before proceeding with the development of the two forms of model following controllers, the reader is advised to keep in mind that this section is intended as a conceptual introduction to model following controllers, meant primarily for those who may not be familiar with the basic theory. The CGT type of model following controller implemented in Chapter V is not derived until Section 2.5, in order to allow the reader to become familiar with the general concepts of model following before being exposed to the more detailed and application-oriented form presented in the following section. The material which follows is taken from References 16 and 7 unless stated otherwise.

The first class of model following controllers is known as the implicit model following type. This is due to the fact that, for this class of controllers, the system dynamics of the model are not incorporated explicitly into the on-line controller; rather, the model is embedded into the definition of the cost function and thus affects the manner in which the controller gains are evaluated. In the second class, the model system dynamics are simulated by the controller, with the difference between the modeled output and the system output being incorporated into a cost function. Thus, these controllers are said to exhibit explicit model following characteristics. Implicit model following controllers will be presented first, followed by a derivation of explicit types.

Consider a continuous-time formulation of Equation (2-24):

$$\underline{\dot{x}}(t) = \underline{F}\underline{x}(t) + \underline{B}\underline{u}(t) \qquad (2\text{-}45)$$

where justification of a deterministic time invariant model for flight control purposes was presented at the end of Section 2.3. Also assume an optimal feedback control law of the following form:

$$\underline{u}^*(t_i) = -\underline{G}_c^*(t_i)\underline{x}(t_i) \qquad (2\text{-}46)$$

The above simple regulator type control law is being used in order to provide insights into implicit model following; the same concepts could be embedded into a PI formulation as well. Note that $\underline{u}(t)$ is equal to $\underline{u}(t_i)$ for $t_i \leq t < t_{i-1}$ with a zero order hold used for digital-to-analog interfacing. An optimal control function can be achieved based on Equation (2-46) by minimizing the continuous-time cost function

$$J_I = \frac{1}{2}\int_0^\infty [\underline{x}^T(t)\underline{X}\underline{x}(t) + \underline{u}^T(t)\underline{U}\underline{u}(t)]dt \qquad (2\text{-}47)$$

where Equation (2-47) exerts a quadratic weight on control energy and state trajectory deviations from zero. A discrete-time version of Equation (2-47) can be derived based upon the same procedure used to discretize Equation (2-16) (11:203,225). Note that the infinite upper integration limit in Equation (2-47) will allow for constant gain controllers since terminal transients can be neglected.

Now define a set of controlled variables

$$\underline{y}(t) = \underline{Cx}(t) \tag{2-48}$$

It is desired that the trajectories of these controlled variables track a set of model variables denoted by the differential equation

$$\dot{\underline{y}}_m(t) = \underline{F}_m \underline{y}_m(t) \tag{2-49}$$

where it is required that both $\underline{y}_m(t)$ and $\underline{y}(t)$ be p-dimensional vectors.

In order to provide a weighting on controlled variable deviation from the model characteristics described by Equation (2-49), a quadratic cost is established that will penalize $\underline{y}(t)$ deviations from the desired characteristics of:

$$\dot{\underline{y}}(t) = \underline{F}_m \underline{y}(t) \tag{2-50}$$

Thus, by implicitly incorporating the modeled system into the cost function by placing a quadratic penalty on the difference between the actual $\dot{\underline{y}}$ and the right side of Equation (2-50), and also adding a cost on the control "energy" $\frac{1}{2} \sum_{i=1}^{r} u_i^2(t)$, the quadratic cost function becomes

$$\mathrm{J}_{\mathrm{I}} = \frac{1}{2} \int_{0}^{\infty} \Big\{ \underline{\dot{y}}(t) - \underline{F}_{m}\underline{y}(t))^{T} \, \underline{Y}_{I}(\underline{\dot{y}}(t) - \underline{F}_{m}\underline{y}(t)) + \underline{u}^{T}(t)\underline{U}\underline{u}(t) \Big\} dt \qquad (2\text{-}51)$$

By invoking Equation (2-48), Equation (2-51) becomes

$$\mathrm{J}_{\mathrm{I}} = \frac{1}{2} \int_{0}^{\infty} \underline{x}^{T}(t)\underline{X}_{I}\underline{x}(t) + 2\underline{u}^{T}(t)\underline{S}_{I}\underline{x}(t) + \underline{u}^{T}(t)\underline{U}_{I}\underline{u}(t)\, dt \qquad (2\text{-}52)$$

where

$$\underline{X}_{I} = (\underline{CF} - \underline{F}_{m}\underline{C})^{T}\underline{Y}_{I}(\underline{CF} - \underline{F}_{m}\underline{C}) \qquad (2\text{-}53)$$

$$\underline{S}_{I} = \underline{B}^{T}\underline{C}^{T}\underline{Y}_{I}(\underline{CF} - \underline{F}_{m}\underline{C}) \qquad (2\text{-}54)$$

and

$$\underline{U}_{I} = \underline{U} + \underline{B}^{T}\underline{C}^{T}\underline{Y}_{I}\underline{CB} \qquad (2\text{-}55)$$

The cross term $\underline{S}_{I}$ arises due to the need to track derivatives of the system outputs, i.e. output rates of Equation (2-50). Once the cost function of Equation (2-52) has been defined, an equivalent discrete form may be derived to yield a sample data control law rather than a continuous type (12:74-76). Controllers for this sample data system can be derived via methods presented in either Section 2.2 or Section 2.3 to yield an optimal control function $\underline{u}^{*}(t_{i})$.

A second approach to model following design is explicit model following. In order to present this approach, the same basic system of Equation (2-45) will be employed. It is worth noting that although the same model equation is used in the derivation of both the implicit and explicit model following controllers, the $\underline{F}_m$ is not the same for both cases.

Consider a system modeling the desired characteristics of the plant

$$\dot{\underline{x}}_m = \underline{F}_m \underline{x}_m \tag{2-56}$$

where time arguments have been dropped for notational tractability.

Now define a new state equation based on the augmentation of the plant states and the model states

$$\dot{\underline{x}}' = \frac{d}{dt}\begin{bmatrix} \underline{x} \\ \underline{x}_m \end{bmatrix} = \begin{bmatrix} \underline{F} & \underline{0} \\ \underline{0} & \underline{F}_m \end{bmatrix}\begin{bmatrix} \underline{x} \\ \underline{x}_m \end{bmatrix} + \begin{bmatrix} \underline{B} \\ \underline{0} \end{bmatrix}\underline{u} \tag{2-57}$$

Thus Equation (2-57) may be represented by

$$\dot{\underline{x}}' = \underline{F}'\underline{x}' + \underline{B}'\underline{u} \tag{2-58}$$

The output vector for the augmented system is defined by

$$\underline{y}' = (\underline{y}_c - \underline{y}_m) = \underline{C}\underline{x} - \underline{C}_m\underline{x}_m = [\underline{C} \mid \underline{C}_m]\begin{bmatrix} \underline{x} \\ \hdashline \underline{x}_m \end{bmatrix} = \underline{C}'\underline{x}' \tag{2-59}$$

where $\underline{C}'$ is chosen to accomplish the desired differencing between modeled state variables and plant controlled variables. Therefore, a standard quadratic performance index may be defined by

$$\underline{J}_E = \int_0^{\infty} \frac{1}{2}(\underline{y}'^T \underline{Y}_E \underline{y} + \underline{u}^T \underline{R}\,\underline{u})dt \qquad (2\text{-}60)$$

The optimal control input for the augmented system as generated by solving the backward Riccati difference equation associated with Equation (2-60) is

$$\underline{u}^* = -[\underline{G}^*_{c1}\ \underline{G}^*_{c2}]\ \underline{x}' \qquad (2\text{-}61)$$

From Equation (2-61) it can be seen that the optimal control input to the system is comprised of the system states fed back through a gain matrix $\underline{G}^*_{c1}$, plus model states fed forward through the set of gains $\underline{G}^*_{c2}$. The feedback through $\underline{G}^*_{c1}$ should provide for tight tracking (7:10-13) which yields a system that is fast enough to follow the responses of the modeled system as commanded through the feedforward controller structure. It can be shown that the feedback compensator is independent of the structure of the explicitly modeled system. Thus, a simple regulator or a PI controller may be designed independently of the explicit model characteristics. As will be shown in the subsequent section, this feedback may be structured in such a manner as to exploit implicit model following to enhance controller robustness characteristics and explicit model following for handling qualities.

Either of the previously discussed model following techniques would prove to be adequate under the idealized assumptions of perfect models which were uncorrupted by disturbances. However, in actual application each type exhibits certain advantages and disadvantages relative to the other.

Implicit model following bases its control on a weighting of rate deviations, as evidenced by Equation (2-51). Therefore, this type of controller achieves a better tracking of the transient response of the model system than explicit model following controllers do. Also, since the implicit formulation attempts to match system and model pole placement, this method exhibits better rejection of unmodeled zero mean disturbances than explicit types. Specifically, since it affects the feedback path rather than a feedforward path, implicit model following may be used to improve the robustness of the controlled system.

As previously stated, the explicit model following technique must embed a model of the system dynamics within the controller. The impact of this embedded model is twofold. First, a more complex controller is needed since the model states are augmented to the system states. This entails higher computational loading. Second, a time lag may be introduced into the system, possibly forcing the designer to settle for a suboptimal control law to offset this controller-induced delay. Since the actual difference between the system and model outputs are weighted, no specific pole placement is achieved as in the implicit scheme; thus, the explicit model controller often must maintain higher feedback gains in order to track modeled transient responses. However, this direct comparison of system and model outputs also produce desirable qualities such as improved steady state performance and reduced sensitivity to parameter variations (7:10-13; 16:1-8).

A primary area of importance to be considered when weighing the relative advantages and disadvantages of model following controllers is robustness enhancement (see Chapter IV). The implicit model following controller can have an effect on system robustness since it incorporates a model system into the cost function to be minimized and thereby affects the feedback gains. Therefore, if the form of the implicit model is chosen properly, it can improve the overall robustness of the system (7:10-13). Alternatively, the explicit scheme allows the model to affect only the feedforward path of the control system. As a result of this, the controller based solely upon an explicit model could improve handling qualities; however, it would not be able to improve system robustness characteristics (14). In view of the apparent differences between the objectives to be achieved by implementing an implicit or explicit controller, it should be emphasized that the implicit and explicit models are not one in the same.

In the past the implicit approach has been more widely used due mainly to its ease of implementation (7:10-13). However, an approach which will be introduced in the following section of this chapter will propose using both implicit and explicit forms in one controller in an attempt to achieve the desirable characteristics of each.

2.5 Command Generator Tracker Synthesis Techniques

This section introduces the controller design technique known as command generator tracking (CGT). This formulation requires a system to track commanded inputs with desirable response characteristics, while simultaneously rejecting disturbances. Thus, the CGT controller is capable of forcing the system state variables of interest to maintain desired trajectories (12:151,166). In terms of the language introduced in the previous section, the CGT scheme relies on explicit model following techniques; however, as will be discussed later in this section, implicit model following may also be incorporated into CGT/regulator or CGT/PI designs (16; 7:10-13; 14). The CGT technique is particularly attractive for aircraft flight controller design because it allows handling qualities to be incorporated directly into the design process.

In order to develop the CGT technique, consider the linear time invariant system model

$$\underline{x}(t_{i+1}) = \underline{\Phi}\underline{x}(t_i) + \underline{B}_d\underline{u}(t_i) + \underline{E}\underline{x}_n(t_i) + \underline{w}_d(t_i) \qquad (2\text{-}62)$$

$$\underline{y}(t_i) = \underline{C}\underline{x}(t_i) + \underline{D}_y\underline{u}(t_i) + \underline{E}_y\underline{n}_d(t_i) \qquad (2\text{-}63)$$

where $\underline{w}_d(t_i)$ is a zero-mean white Gaussian noise sequence, and $\underline{n}_d(t_i)$ is a time correlated noise sequence modeled by

$$\underline{n}_d(t_{i+1}) = \underline{\Phi}_n\underline{n}_d(t_i) + \underline{B}_{dn}\underline{n}_{cmd}(t_i) + \underline{G}_{dn}\underline{w}_{dn}(t_i) \qquad (2\text{-}64)$$

It is desired that the output of Equation (2-63) emulate the output of a command generator model

$$\underline{x}_m(t_{i+1}) = \underline{\Phi}_m\underline{x}_m(t_i) + \underline{B}_{dm}\underline{u}_m \qquad (2\text{-}65)$$

$$\underline{y}_m(t_i) = \underline{C}_m\underline{x}_m(t_i) + \underline{D}_m\underline{u}_m \qquad (2\text{-}66)$$

where $\underline{y}_m(t_i)$ and $\underline{y}(t_i)$ must both be of dimension p. For flight control applications, the input $\underline{u}_m$ to the command generator model can be considered piecewise constant over a time interval of interest since it is assumed to vary slowly in comparison to the sampling rate of the digital control system.

In order to achieve tracking of the modeled system, the CGT must drive the error

$$\underline{e}(t_i) = \underline{y}_c(t_i) - \underline{y}_m(t_i) \qquad (2\text{-}67)$$

to zero. In terms of the previously established models, Equation (2-67) can be expressed as

$$\underline{e}(t_i) = (\underline{C}\ \underline{D}_y\ \underline{E}_y) \begin{bmatrix} \underline{x}(t_i) \\ \underline{u}(t_i) \\ \underline{n}_d(t_i) \end{bmatrix} - [\underline{C}_m\ \underline{D}_m] \begin{bmatrix} \underline{x}_m(t_i) \\ \underline{u}_m \end{bmatrix} \qquad (2\text{-}68)$$

For convenience in construction of the final CGT law, an ideal trajectory can be defined as the plant state trajectory which will drive Equation (2-68) to zero for all time and satisfy

$$\underline{x}_I(t_{i+1}) = \underline{\Phi}\,\underline{x}_I(t_i) + \underline{B}_d\underline{u}_I(t_i) + \underline{E}_x\underline{n}_d(t_i) \qquad (2\text{-}69)$$

The stated desire to drive the error defined in Equation (2-67) to zero, coupled with the form of Equations (2-68) and (2-69) can be combined with a third condition:

$$\begin{bmatrix} \underline{x}_I(t_i) \\ \underline{u}_I(t_i) \end{bmatrix} = \begin{bmatrix} \underline{A}_{11} & \underline{A}_{12} & \underline{A}_{13} \\ \underline{A}_{21} & \underline{A}_{22} & \underline{A}_{23} \end{bmatrix} \begin{bmatrix} \underline{x}_m(t_i) \\ \underline{u}_m(t_i) \\ \underline{n}_d(t_i) \end{bmatrix} \qquad (2\text{-}70)$$

Recalling Equation (2-29),

$$\begin{bmatrix} \underline{\Phi} - \underline{I} & \underline{B}_d \\ \underline{C} & \underline{D}_y \end{bmatrix}^{-1} = \begin{bmatrix} \underline{\Pi}_{11} & \underline{\Pi}_{12} \\ \underline{\Pi}_{13} & \underline{\Pi}_{14} \end{bmatrix} \qquad (2\text{-}71)$$

the solution to the CGT problem formulation can be shown (12:155) to be of the form of Equation (2-70) with:

$$\underline{A}_{11} = \underline{\Pi}_{11}\,\underline{A}_{11}\,(\underline{\Phi}_{m} - \underline{I}) + \underline{\Pi}_{12}\,\underline{C}_{m} \tag{2-72}$$

$$\underline{A}_{12} = \underline{\Pi}_{11}\,\underline{A}_{11}\underline{B}_{dm} + \underline{\Pi}_{12}\,\underline{D}_{m} \tag{2-73}$$

$$\underline{A}_{13} = \underline{\Pi}_{11}\,\underline{A}_{13}\,(\underline{\Phi}_{n} - \underline{I}) - \underline{\Pi}_{11}\,\underline{E}_{x} - \underline{\Pi}_{12}\,\underline{E}_{y} \tag{2-74}$$

$$\underline{A}_{21} = \underline{\Pi}_{21}\,\underline{A}_{11}\,(\underline{\Phi}_{m} - \underline{I}) + \underline{\Pi}_{22}\,\underline{C}_{m} \tag{2-75}$$

$$\underline{A}_{22} = \underline{\Pi}_{21}\,\underline{A}_{11}\underline{B}_{dm} + \underline{\Pi}_{22}\,\underline{D}_{m} \tag{2-76}$$

$$\underline{A}_{23} = \underline{\Pi}_{21}\,\underline{A}_{13}\,(\underline{\Phi}_{n} - \underline{I}) - \underline{\Pi}_{21}\,\underline{E}_{x} - \underline{\Pi}_{22}\,\underline{E}_{y} \tag{2-77}$$

Upon solving Equations (2-72) through (2-77), the control input can be generated as the lower partition of Equation (2-70):

$$\underline{u}_{I}(t_{i}) = \underline{A}_{21}\,\underline{x}_{m}(t_{i}) + \underline{A}_{22}\,\underline{u}_{m}(t_{i}) + \underline{A}_{23}\,\underline{n}_{d}(t_{i}) \tag{2-78}$$

The block diagram of the control law represented by Equation (2-77) is shown in Figure 2.2 on the next page.

As evidenced by Figure 2.2, this formulation of the CGT law is an open loop form; therefore, this controller will not compensate for uncertainties in the system model. The need to compensate for these inevitable uncertainties which arise in flight control applications motivates the construction of a CGT law which incorporates feedback into

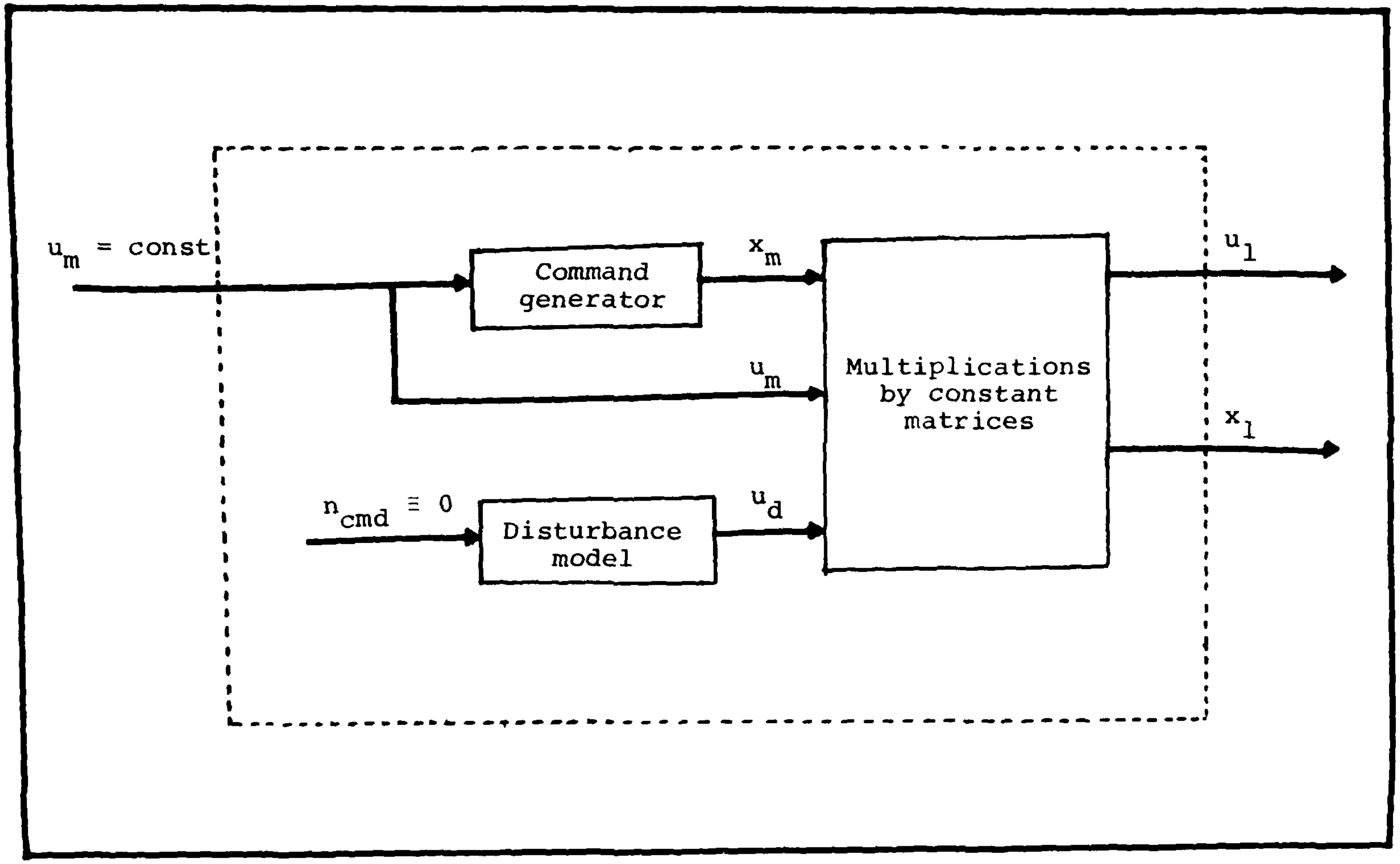


Fig. 2.2. Open-Loop Command Generator Tracker

the final design. A second improvement over the simple open loop configuration CGT would be the use of a closed loop PI law, instead of a simple regulator law, thus yielding a command generator tracking/proportional plus integral or (CGT/PI), controller. The advantage of a CGT/PI controller is that it enables the system to achieve type 1 characteristics: (1) it can force the mean steady state error between the actual plant and the command generator model to zero, and (2) it can reject modeled and unmodeled disturbance inputs.

The CGT/PI control law that is used for the purpose this thesis provides type 1 system characteristics and has the desirable quality of being derived via LQG synthesis methods. The final form of this controller can be shown to be of the following form (16:2-15):

$$\begin{aligned}
\underline{u}(t_i) &= \underline{u}(t_{i-1}) - \underline{K}_x(\underline{x}(t_i) - \underline{x}(t_{i-1})) \\
&+ \underline{K}_z \left\{ [\underline{C}_m \ \underline{D}_m] \begin{bmatrix} \underline{x}_m(t_{i-1}) \\ \underline{u}_m(t_i) \end{bmatrix} - [\underline{C} \ \underline{D}_y] \begin{bmatrix} \underline{x}(t_{i-1}) \\ \underline{u}(t_{i-1}) \end{bmatrix} \right\} \\
&+ \underline{K}_{xm}(\underline{x}_m(t_i) - \underline{x}_m(t_{i-1})) \\
&+ \underline{K}_{xu}(\underline{u}_m(t_i) - \underline{u}_m(t_{i-1})) \\
&+ \underline{K}_{xn}(\underline{n}_d(t_i) - \underline{n}_d(t_{i-1})) \qquad (2\text{-}79)
\end{aligned}$$

(The use of $\underline{u}_m$ at t_i instead of t_{i-1} accounts for an inconsistency in the definition of the desired state trajectory when a step change to $\underline{u}$ is applied (12:161-162).) In terms of Section 2.3 the gains associated with Equation (2-79) are defined as:

$$\underline{K}_x = \underline{G}^*_{c1}\underline{\Pi}_{11} + \underline{G}^*_{c2}\underline{\Pi}_{21} \tag{2-80}$$

$$\underline{K}_z = \underline{G}^*_{c1}\underline{\Pi}_{12} + \underline{G}^*_{c2}\underline{\Pi}_{22} \tag{2-81}$$

$$\underline{K}_{xm} = \underline{K}_x\underline{A}_{11} + \underline{A}_{21} \tag{2-82}$$

$$\underline{K}_{xu} = \underline{K}_x\underline{A}_{12} + \underline{A}_{22} \tag{2-83}$$

$$\underline{Kx}_n = \underline{K}_x\underline{A}_{13} + \underline{A}_{23} \tag{2-84}$$

Up to this point the assumption of full state feedback has been allowed, due to certainty equivalence. In order to account for the more accurate physical case of incomplete, noise-corrupted state availability, a Kalman filter can be introduced into the design process. This final innovation will produce what is termed the Command Generator Tracking/Proportional plus Integral/Kalman Filter (CGT/PI/KF) controller. The type of Kalman filter to be employed in the CGT/PI/KF design is a standard steady state, constant-gain filter designed for time invariant system models with stationary noises, ignoring the initial gain transients. For flight control applications a constant-gain filter is acceptable due to the relatively short transient period experienced in relation to the steady state operation of the aircraft.

The Kalman filter will produce a state estimate $\hat{\underline{x}}(t_i^+)$; however, in cases where unmodeled physical delays within the structure of the plant or controller are present, a control law may preferably be established based upon $\hat{\underline{x}}(t_i^-)$ (12:161). Basing the control input on measurements up to, but not including, the measurement that becomes available at time t_i can remove computational delay time; however, the resulting control law is less precise than a law based on $\hat{\underline{x}}(t_i^+)$. The modeled noise states $\underline{n}_d(t_i)$ may also be replaced with Kalman filter estimates, but it must be remembered that the plant state estimates and the noise state estimates do not decompose into the independent filters (12:166). Considering the above, the filter model will simply consist of the dynamics model of the plant augmented with the dynamics model of the time correlated noise corruption (16). This augmented filter configuration has a measurement model of the form

$$\underline{z}(t_i) = [\underline{H} \quad \underline{H}_n] \begin{bmatrix} \underline{x}(t_i) \\ \underline{n}_d(t_i) \end{bmatrix} + \underline{v}(t_i) \qquad (2\text{-}85)$$

where $\underline{v}(t_i)$ is characterized as a white zero-mean Gaussian noise sequence with a covariance $\underline{R}$.

The previous discussion developed the fundamental design equations needed to implement the CGT/PI/KF controller. Although this form of the CGT law is desirable in that it incorporates a PI controller to achieve type 1 performance characteristics, it has the drawback of being difficult to robustify (see Chapter IV for a discussion of robustness enhancement). Therefore, a slight digression is in order so that a less elaborate form of the CGT law may be presented. This particular form of the CGT controller replaces the PI feedback channel with a simple regulator and may be used if problems arise as a result of applying specific robustness enhancement techniques.

The Command Generator Tracker/Regulator (CGT/R) is developed fully in Reference 12 where it is shown to be of the following form:

$$\begin{aligned}\underline{u}(t_i) &= -\underline{G}_c^* \, \underline{x}(t_i) + (\underline{A}_{21} + \underline{G}_c^* \underline{A}_{11}) \, \underline{x}_m(t_i) \\ &\quad + (\underline{A}_{22} + \underline{G}_c^* \underline{A}_{12}) \, \underline{u}_m(t_i) + (\underline{A}_{23} + \underline{G}_c^* \underline{A}_{13}) \, \underline{n}_d(t_i)\end{aligned} \tag{2-86}$$

As in the CGT/PI case, the states of the plant and modeled noise corruptions may be estimated by a Kalman filter to yield a CGT/R/KF design. A block diagram of the CGT/PI/KF form is shown in Figure 2.3 on the next page.

The last topic to be discussed in this section relates to the model following techniques introduced in Section 2.4. As previously stated, there exist two subcategories of model following controllers: implicit and explicit; however, these two forms are not mutually

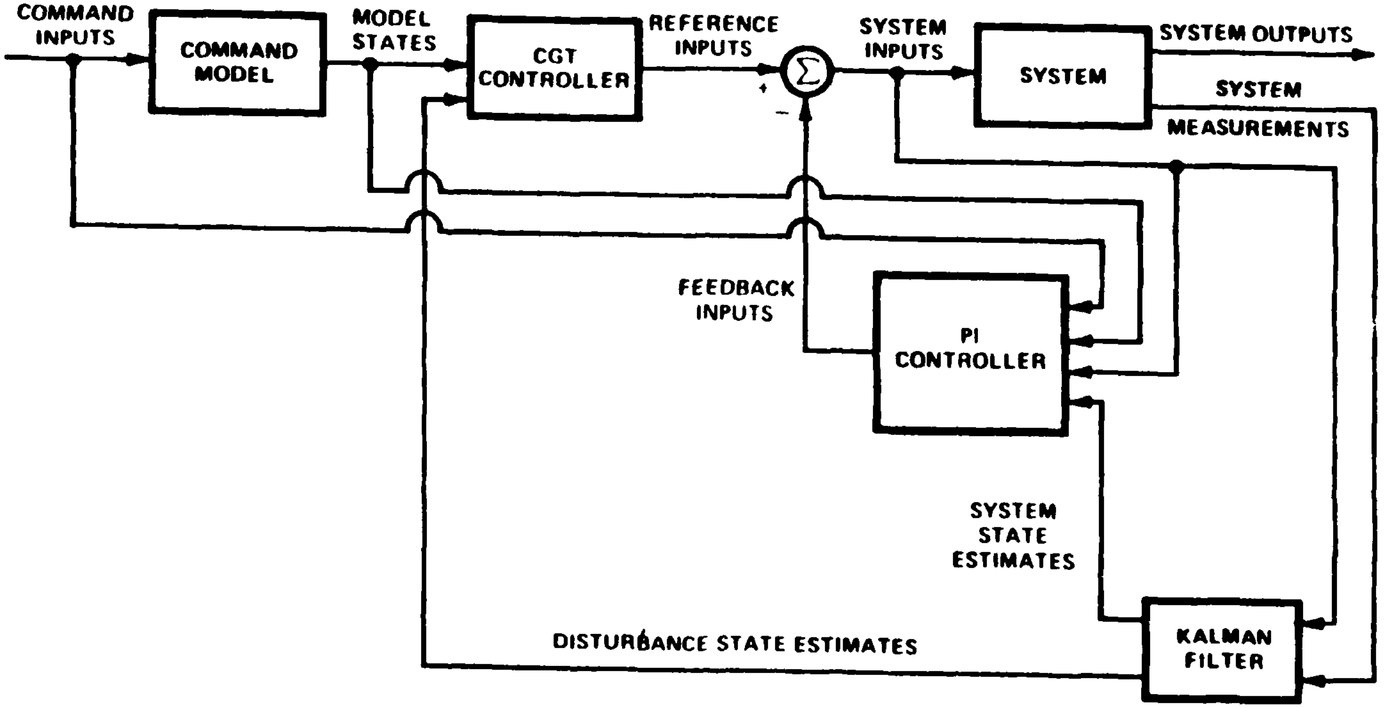


Fig. 2.3. A Block Diagram of the CGT/PI/KF Controller

exclusive. In this thesis the work developed by Miller (16) will be implemented to derive a controller which exploits both implicit and explicit model following through use of a quadratic cost function of the form

$$J_c = \frac{1}{2} \int_0^\infty ((\dot{\underline{y}} - \underline{F}_m \underline{y})^T \underline{Y}_I (\dot{\underline{y}} - \underline{F}_m \underline{y}) + (\underline{y} - \underline{y}_m)^T \underline{Y}_E (\underline{y} - \underline{y}_m) + \underline{u}^T \underline{U} \underline{u}) dt \qquad (2\text{-}87)$$

By using a cost function of the form of Equation (2-87), both the handling qualities and robustness characteristics may be improved by use of explicit and implicit model following, respectively.

Although some standard initialization steps can be used to make a first estimate of $\underline{Y}_I$ and $\underline{Y}_E$, these weightings are usually developed through an iterative trial and error method. Chapter V will provide some insights into this process.

2.6 Summary

This chapter has served to generate the basic design tools used in the aircraft controller design of Chapter V. First, the fundamental LQG regulator was introduced and its uses and shortcomings were assessed. Next, the PI controller of classical control theory was developed using powerful modern LQG methods. Sections 2.4 and 2.5 presented both general model following theory and more specific CGT/PI and CGT/R control forms. Certainty equivalence then allowed definition of CGT/PI/KF and CGT/R/KF laws. The CGT/PI/KF controller design laws will be applied to a fighter aircraft combat mode controller in Chapter V using interactive computer-aided design and evaluation packages (7; 16; 17).

III. Modeling Considerations for the STOL F-15

3.1 Introduction

This chapter will introduce the models upon which the controller designs of Chapter V are based. Also, some of the unique features of the short takeoff and landing (STOL) F-15 will be discussed. In the interest of clarity and brevity, such data as aerodynamic derivatives and aircraft state models within the flight envelope will not all be presented at this time; instead, these data will be included in Appendix B. The design software used in this thesis (7; 16), will be introduced inasmuch as it is needed to justify specific model development. Those readers who desire a more detailed discussion of the software are directed to References 7 and 16.

The STOL F-15 is a modified version of the F-15 currently being used by the United States Air Force as an air superiority fighter aircraft. The major modifications to the standard F-15 which produce the STOL capability are the introduction of canards, rotating vanes, and two dimensional thrust vectoring nozzles. Of these three modifications, the vanes and thrust nozzles are unconventional control surfaces and bear future explanation. The rotating vanes consist of four louvered panels located on the upper and lower surfaces of the thrust vectoring attachment located at the rear of the engines. The two dimensional nozzles are located at the rear of the same thrust vectoring assembly. This thesis effort will be limited to a controller design for the "up and away" characteristics of the STOL F-15. Since the rotating vanes do not provide a significant amount of control within the flight envelope being studied, they will be considered closed at all times. Therefore, the vanes are completely eliminated from the control input vector.

The second section of this chapter will introduce some of the basic aerodynamic modeling considerations relevant to the control of the STOL F-15 aircraft. Section 3.3 discusses some of the models other than the basic aircraft model which are motivated by the CGT design method and the computer aided design (CAD) package used to implement this design method. The flight envelope of the STOL F-15 is the topic of Section 3.4. The nominal design point is specified and other points in the flight envelope are discussed as well, with the goal of establishing a "nominal" point to be used for controller designs and several points on the edge of the envelope to be used to evaluate the robustness of these designs.

The final section of this chapter provides a summary of the topics presented and highlights those areas of particular relevance to the subsequent chapters.

3.2 Modeling the STOL F-15 Aircraft

This section will discuss the construction of the linearized model used to represent the STOL F-15 aircraft. This presentation will assume that the reader is familiar with atmospheric flight dynamics; those who require a detailed presentation of the fundamental principles of aircraft modeling and flight control are directed to References 6 and 7. The basic aircraft model is derived from the general nonlinear equations of motion which describe the dynamics of the STOL F-15. These equations can be greatly simplified by assuming that: 1) the earth is an inertial plane as opposed to a non-inertial spheroid. This "flat earth" assumption can be considered reasonable at flight conditions which do not exceed Mach 3 (6), which is well below all velocities being considered in this thesis; 2) the atmosphere is at rest; 3) all elastic effects will be considered negligible, yielding a "rigid body" model with no elastic freedom. The rejection of wind buffeting could be incorporated into the controller synthesis methodology (12:151; 14); however, this issue will not be addressed herein. The ability of the controller to reject such disturbances without explicitly modeling them will be assessed, however. The force and moment equations which result from the above simplifying assumptions may be linearized about specific flight conditions, using small disturbance theory (6:154), resulting in a linearized model for the STOL F-15. By invoking the assumptions concerning aircraft symmetry, absence of gyroscopic effects, and neglecting aerodynamic cross coupling terms (6:161), the lateral and longitudinal modes of the STOL F-15 may be completely decoupled and treated separately for the purposes of controller design. It is this set of decoupled, linearized equations of motion placed in matrix form which will be developed for the longitudinal mode of flight.

The aerodynamic data at various operating points within the flight envelope of the STOL F-15 were provided by McDonnell Aircraft Engineering (McAir). These data consisted of aircraft parameters and longitudinal and lateral non-dimensionalized body axis force coefficients. Also, moments of inertia were provided; however, these were provided in dimensionalized body axis form. In order to convert the data provided by McAir into dimensionalized body axis form, an existing axis conversion program written by Mr. Finley Barfield (3) was modified by Capt. Greg Mandt and Lt. Bruce Clough, and re-named STOLCAT. STOLCAT incorporates the non-dimensionalized body axis force coefficients associated with the canards, two dimensional thrust nozzles, and rotating vanes, along with the conventional force coefficients to produce a set of dimensionalized body axis coefficients. The STOLCAT software also forms both the lateral and longitudinal state space three-degree-of-freedom equations of motion of the following form:

$$\underline{\dot{x}}(t) = \underline{Fx} + \underline{Bu}(t) \qquad (3\text{-}1)$$

where

$\underline{x}$ = aircraft state vector

$\underline{F}$ = aircraft dynamics fundamental matrix

$\underline{B}$ = control derivative matrix

$\underline{u}$ = control vector

At Mach 0.9 at 20,000 feet of altitude, Equation (3-1) is:

$$\begin{bmatrix} \dot{u} \\ \dot{q} \\ \dot{\alpha} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} -.018 & -20.023 & 27.9 & -32.19 \\ -.23\text{X}10^{-3} & -1.999 & 10.81 & 0 \\ -.34\text{X}10^{-6} & .9997 & -1.49511 & -7.4\text{X}10^{-3} \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} u \\ q \\ \alpha \\ \theta \end{bmatrix}$$

$$+ \begin{bmatrix} 1.5 & -9.9 & 45 \\ 11.7 & -19.69 & 0 \\ -.045 & -.19 & 0 \\ 0 & 0 & 0 \end{bmatrix} \underline{u}(t) \qquad (3\text{-}1a)$$

A complete FORTRAN program listing of STOLCAT, along with the aerodynamic data for the STOL-F-15, are included in Appendices A and B of this thesis.

In the longitudinal mode the control vector consists of the canard input δ_C, the stabilator input δ_S, and the throttle input δ_T. Although the rotating vanes and thrust vectoring nozzles are available, they are not used in the designs considered in this thesis. The vanes are neglected due to their limited usefulness at the altitude and airspeeds that are being considered. The nozzles are useful for "up-and-away" flight; however, the simultaneous control of thrust and nozzle deflection introduces a severe nonlinearity into the system dynamics which is beyond the scope of this thesis (see Appendix C for a more complete discussion of this nonlinear effect). The longitudinal control vector is shown below:

$$\underline{u}(t) = \begin{bmatrix} \delta_C(t) \\ \delta_S(t) \\ \delta_T(t) \end{bmatrix} \qquad (3\text{-}2)$$

Although a basic state model representing the aircraft dynamics can be produced by the methods discussed in this section, this model alone will not be sufficient to accomplish the controller designs required of this thesis effort. The use of CGT techniques and the desire to accomplish performance evaluations of the final designs require that other dynamics models be constructed. The following section will discuss these other dynamics models, which will be used along with the basic aircraft design model to affect the final designs of Chapter V.

3.3 Models for CGT Design and Performance Evaluation

Although the previous section developed the linearized decoupled equations of motion for the STOL F-15, these equations alone will not be sufficient to complete the designs required of this thesis. This section will introduce the reader to the models other than the basic aircraft model which will be used in Chapter V. The models presented in this chapter will be somewhat general in nature, thus saving the detailed descriptions of the models particular to the STOL F-15 for Chapter V and Appendix B.

The underlying motivation for the following models is theoretical in nature (see Chapter II); however, their specific form is dictated by the CAD package CGTPIF (7; 16). A discussion of the models from a software user's point of view will be presented at this time in order to give the reader a qualitative understanding of the models as they pertain to the use of CGTPIF. Therefore, the goal of this section is to ground the reader in the uses of the various models presented herein before the actual CGT designs are presented.

Four different models will be described in this section. The first is the design model, followed by the command model or explicit model, the truth model, and finally, the implicit model.

The design model used by CGTPIF is the primary model used in the design process. It is derived from the basic aircraft model Equation (3-1), augmented with noise inputs. The specific form of the design model which is required by CGTPIF is as follows (7):

$$\underline{\dot{x}}(t) = \underline{Fx}(t) + \underline{Bu}(t) + \underline{E}_x\underline{n} + \underline{Gw}(t) \tag{3-3}$$

$$\underline{\dot{n}}(t) = \underline{F}_n\underline{n}(t) + \underline{G}_n\underline{w}_n(t) \tag{3-4}$$

$$\underline{y}(t) = \underline{Cx}(t) + \underline{D}_y\underline{u}(t) \tag{3-5}$$

$$\underline{z}(t_i) = \underline{Hx}(t_i) + \underline{v}(t_i) \tag{3-6}$$

where $\underline{y}$ represents the controlled outputs and $\underline{z}$ is the noise-corrupted measurement vector. The $\underline{n}$ term denotes a time correlated noise disturbance input into the system which is to be rejected by the controller; however, for the purposes of the work accomplished in this thesis, this input will be removed from the system. The noise inputs of the system are characterized by

$$E\{\underline{w}(t)\underline{w}^T(t+\tau)\} = \underline{Q}\delta(\tau) \tag{3-7}$$

$$E\{\underline{w}_n(t)\underline{w}_n^T(t+\tau)\} = \underline{Q}_n\delta(\tau) \tag{3-8}$$

and the measurement noise of the system is described by

$$E\{\underline{v}(t_i)\underline{v}^T(t_j)\} = \underline{R}\delta_{ij} \qquad (3\text{-}9)$$

The dimensionalities which must be established before entering the design model in the CGTPIF software are

n = number of system states
r = number of system inputs
p = number of system outputs
m = number of system measurements
w = number of independent system noises

The command model is an explicit model (see Sections 2.4 and 2.5) used to describe the desired behavior of the system. Embedded in this model are the specifications that the actual system must meet. Typically for flight control applications this model will be used to command a second order response for the output variables, thus allowing incorporation of specified handling qualities or "feel" of the aircraft as it responds to the pilot's control inputs. A second possible flight control application for the command model would be to command an optimal evasive maneuver for an aircraft operating in a combat situation. The specific form of this model as dictated by CGTPIF is as follows:

$$\dot{\underline{x}}_E(t) = \underline{F}_E\underline{x}_E(t) + \underline{B}_E\underline{u}_E(t) \qquad (3\text{-}10)$$

$$\underline{y}_E = \underline{C}_E\underline{x}_E(t) + \underline{D}_E\underline{u}_E(t) \tag{3-11}$$

where the subscript E denotes Explicit model, and dimensionalities for the command model are

n_E = number of command model states
r_E = number of command model inputs
p_E = number of command model outputs

CGTPIF further requires that the number of command model states be less than or equal to the number of design model states (software constraint; theoretical extensions are possible), and that the number of system outputs be equal to the number of command model outputs (logical theoretical constraint, since system outputs are supposed to track the command model outputs).

CGTPIF (7; 16) not only provides for the design of Command Generator Tracker/Proportional plus Integral/Kalman Filter (CGT/PI/KF) and Command Generator Tracker/Regulator/Kalman Filter (CGT/R/KF) controllers, but provides for performance analysis of the resulting designs. In order to accomplish this analysis, a linear model must be created which represents, as closely as possible, the actual dynamics of the system. This "truth model" will represent the same system as the design model; however, it will typically be of a higher dimensionality and complexity than the design model since it will include higher order sensor and actuator dynamics as well as wind buffeting states possibly ignored in the CGT design process. The form of the truth model required by CGTPIF is found in Reference 7 to be:

$$\dot{\underline{x}}_T(t) = \underline{F}_T\underline{x}_T(t) + \underline{B}_T\underline{u}_T(t) + \underline{G}_T\underline{w}_T(t) \tag{3-12}$$

$$z_T(t_i) = \underline{H}_T\underline{x}_T(t_i) + \underline{v}_T(t_i) \tag{3-13}$$

$$\underline{x}(t) = \underline{T}_{DT}\underline{x}_T(t) \tag{3-14}$$

$$\underline{n}(t) = \underline{T}_{NT}\underline{x}_T(t) \tag{3-15}$$

with associated noise statistics

$$E\{\underline{w}_T(t)\underline{w}_T^T(t+\tau)\} = \underline{Q}_T\delta(\tau) \tag{3-16}$$

$$E\{\underline{v}_T(t_i)\underline{v}_T^T(t_j)\} = \underline{R}_T\,\delta_{ij} \tag{3-17}$$

where

n_T = number of truth model states

r_T = number of truth model inputs

m_T = number of truth model measurements

w_T = number of independent noises in the truth model

In Equations (3-14) and (3-15), $\underline{T}_{DT}$ and $\underline{T}_{NT}$ are matrices which transform the truth model state and disturbance vectors into vectors corresponding to the design model state and disturbance vectors respectively. In order for CGTPIF to accomplish an analysis of the designed system, obviously r_T must be equal to r, and m_T must equal m.

The last model to be introduced in this section is the implicit model. This model is not as easily understood on an intuitive level as the previous three (see Section 2.4); however, it may be thought of as a means of providing a pole placement technique to improve the robustness characteristics of the overall control system (see Section 4.2). The implicit model is so named because it is embedded into the process of choosing the weights of an LQ controller and does not, unlike the explicit model, appear explicitly in the final controller implementation. A more detailed discussion of how to design the implicit model to achieve robustness is included in Chapter IV. As with the previous three models, the specific form of the implicit model is a function of CGTPIF requirements (16), and is given by:

$$\dot{\underline{y}}_I(t) = \underline{F}_I\underline{y}_I(t) \tag{3-18}$$

where

n_I = number of implicit model states
r_I = number of implicit model inputs
p_I = number of implicit model outputs

It cannot be overemphasized that, although implicit and explicit models have the same basic structure and share the same input routine in CGTPIF, they are not the same model, they can and should serve different purposes, and they need not be related in any way.

This section is intended to give the reader a qualitative understanding of the models used in this thesis. It is basically a "bridge" from the theoretical modeling developments in Chapter II to the complete and detailed models presented in Chapter V. Therefore, this section does not stand alone as a complete development of models relevant to this thesis; however, coupled with the developments in Chapters II and V, the reader should transition from a theoretical understanding to a qualitative "feel" to a complete understanding of the specific CGTPIF-oriented models as they pertain to the STOL F-15 aircraft.

3.4 The STOL F-15 Flight Envelope

As stated in the previous section, CGTPIF allows for a performance analysis of completed controllers by use of a truth model to represent the "real world" characteristics of the system being controlled. To be more specific, it allows analysis of the full-state feedback controller and a separate analysis of the Kalman Filter, but it does not provide for analysis of the total controller as a cascade of these two components. Another CAD package called PERFEVAL is available which allows for a performance analysis of a Kalman Filter based CGT/PI controller (17). This software capability is exploited not only to evaluate the controller designs at the nominal design conditions of Mach 0.9 at 20,000 feet altitude, but also to evaluate the robustness of the controller throughout the combat operation range of the STOL F-15, and to evaluate control system robustness in the face of variations in stability derivatives, surface failures, mismodeled actuator dynamics, etc. See Figure 3.1 for a depiction of the flight conditions being considered in this thesis. The motivation for designing a robust system with the capability to operate over a large portion of the STOL F-15's flight envelope is two-fold. First, a control system which does not require extensive gain scheduling or real time parameter estimation for adjustment to changes in operating conditions greatly reduces the complexity of the control system. This reduction in controller complexity not only reduces the computational loading of the on-board flight computer system, but reduces the physical space requirements and weight requirements of the controller as well. Second, a system which can at least maintain stability in the face of parameter variations could be a first step in a reconfigurable control system (i.e. a robust law to hold the aircraft in the air while reconfiguration is accomplished), capable of greatly enhancing the survivability of the STOL F-15 in the face of combat damage.

In order to construct the truth models used to represent the variations in the dynamics of the STOL F-15 as the aircraft progresses through its flight envelope, it is necessary to: a) vary the entries uniformly in the truth

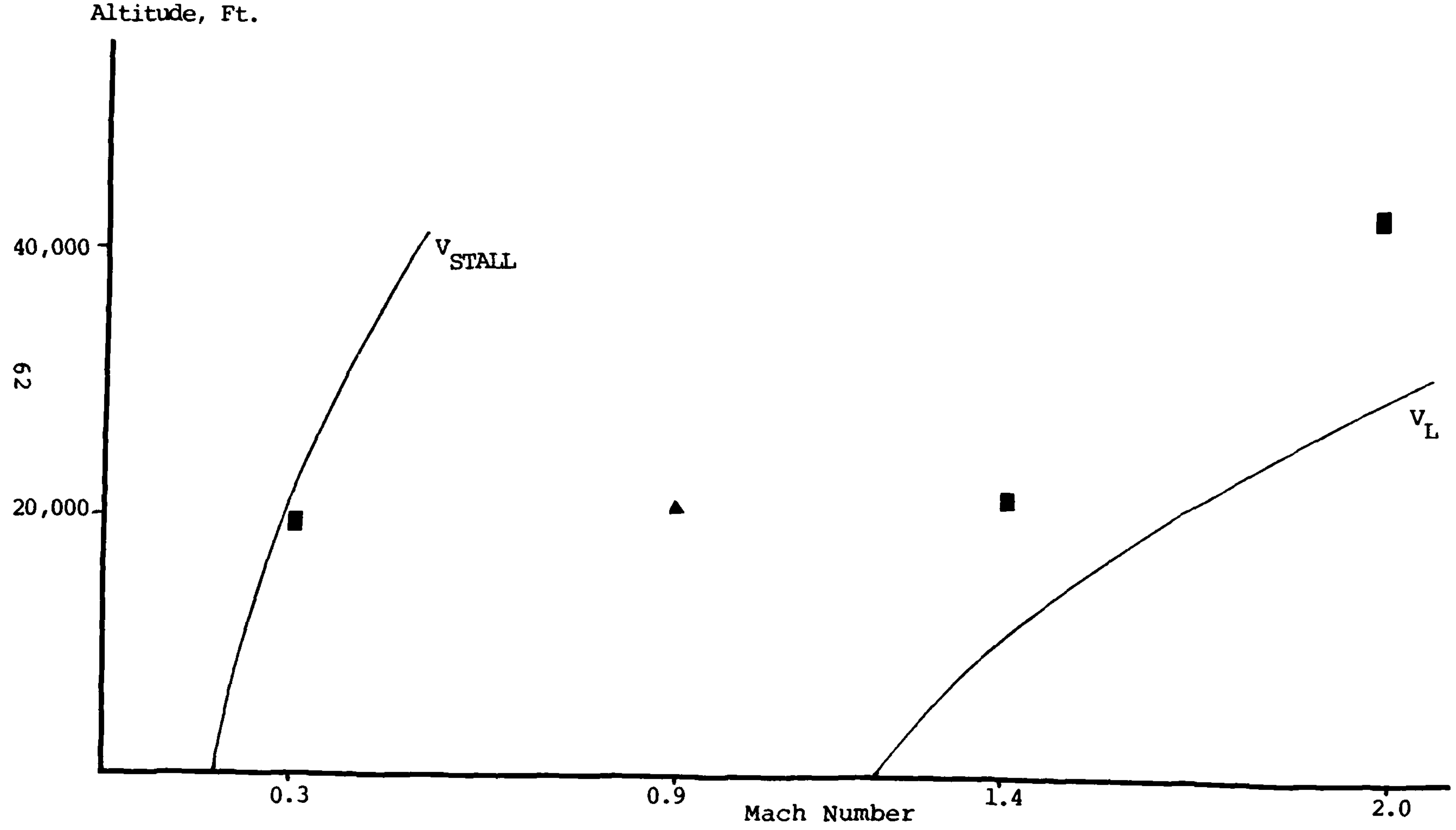


Fig. 3.1. STOL F-15 Flight Envelope

model which represent the aircraft's dynamics by a set percentage according to what degree the flight condition is to be varied; b) use data, linearized at flight conditions which lie on or close to the boundaries of the flight envelope which the controller is to operate within, to derive new equations of motion; c) inject noise into the truth model states which are likely to be mismodeled or subject to variations; or d) re-derive the aircraft equations to include possible failures such as partial or total loss of actuators. All of these techniques are discussed in Chapter V.

The STOL F-15's flight envelope is depicted in Figure 3.1. The conditions which are used to construct off-design conditions are shown as boxes, while the nominal design condition, which is chosen as representative of the "standard" flight condition for entering into air-to-air combat (22), is depicted as a triangle.

3.5 Summary

This chapter has introduced several important concepts related to modeling the STOL F-15 aircraft, with the goal of using these models to construct CGT/PI and CGT/R control laws. In Section 3.2 the basic assumptions made in order to produce time invariant, linearized, decoupled equations of motion are discussed. Section 3.3 introduced the form of the dynamics equations which are motivated by the CGTPIF (7; 16) CAD package which is used to generate the controllers of Chapter V. Also, a software package called PERFEVAL (17) is introduced in order to analyze the Kalman-filter-based controllers designed using CGTPIF (see Appendix D). The purpose of Section 3.4 was to justify the points in the flight envelope used as variant operating conditions for analyzing the robustness of the control systems generated during the course of this thesis effort. It is important for the reader to realize that this chapter is not intended as a comprehensive explanation of modeling the STOL F-15 aircraft, but rather a transition to provide familiarization with the types of models and terminology which are used extensively in Chapter V.

IV. Robustness Enhancement Techniques

4.1 Introduction

The concepts and methods of robustness enhancement techniques used in this thesis are discussed in this chapter. Two specific types of robustness techniques are addressed. First, implicit model following techniques are presented. Implicit model following controllers were initially presented in Section 3.4, but in this section the implicit model following technique will briefly be re-examined in the light of its impact on closed-loop full-state feedback system robustness. Secondly, the Loop Transfer Recovery (LTR) technique will be developed and shown to provide asymptotic full-state feedback system robustness characteristics to Kalman-filter-based controllers.

4.2 Implicit Model Following

The derivation of the implicit model following technique is contained in Section 3.4, so this section will not be oriented towards mathematical rigor. Instead, the objective of this section is to introduce the reader to some of the qualitative aspects of using the implicit model controller as a robustness enhancement technique without mathematical proof.

Recall from Section 3.4 the implicit model following controller is based on an implicit model of the form

$$\dot{\underline{y}}_I(t) = \underline{F}_I \underline{y}_I(t) \tag{4-1}$$

with a cost function constructed by weighting the following difference:

$$\underline{e}(t) = \dot{\underline{y}}(t) - \underline{F}_I \underline{y}(t) \tag{4-2}$$

along with a quadratic penalty on control values, where $\underline{y}$ is the output vector associated with the plant. Thus the objective is to force the plant to adopt dynamics as described by the implicit model $\underline{F}_I$ matrix.

The question which needs to be asked is, how should the implicit model be chosen in order to improve the overall robustness characteristics of the system? In the case of the explicit model, which has no effect on robustness, it is apparent that the model should produce outputs which follow a desirable trajectory for the actual plant outputs to emulate. However, by examining Equation (4-2), it can be seen that the implicit model, unlike the explicit model, is used in its complete dynamical form in the cost function definition. Therefore, the state matrix $\underline{F}_I$ is used along with the output variables of the actual system to define a cost function for developing the feedback control via LQ methods. The $\underline{F}_I$ matrix can be thought of as embedding the desired characteristic equation properties of the controlled variables into the definition of the cost function. Therefore, the implicit model provides the ability to carry out classical "pole placement" in a MIMO design using powerful LQ synthesis techniques.

The preceding discussion defines a classical control theory parallel to implicit model following; however, it does not specifically address the robustness issue. The pole placement concept can be exploited to this end. Consider an aircraft controller design based on a reduced-order model of a system, with order reduction carried out by ignoring actuator dynamics considered to be beyond the bandwidth of the controller. Using LQ synthesis techniques, a regulator or PI controller (see Sections 2.2 and 2.3) can be designed based upon this reduced order model. If, upon completion of the controller design, it is found that the desired bandwidth constraints are violated, the unmodeled actuator dynamics may be excited, resulting in system instability; i.e. the controller is not "robust" in the face of unmodeled higher order dynamics. The inclusion of an implicit model in the cost function used to derive the previously discussed controller provides the designer with a means to place a quadratic penalty on the deviation of Equation (4-2) from zero, thus penalizing deviations of the output variables from behavior which is dictated by the characteristic equation defined by $\underline{F}_I$ in Equation (4-2). Therefore, $\underline{F}_I$ may be chosen to "place" the poles of the controller, providing a means to reduce the system bandwidth and robustify the controller against ignored higher order dynamics. In terms of the classical root locus, the designer should place the desired poles of the system far enough from the unmodeled poles that "unexpected" movement of these unmodeled poles will have the least possible effect on the stability of the overall system (14). It has also been shown that if the eigenvectors of the desired system model are nearly orthogonal, the robustness characteristics of the closed system are improved (14).

4.3 Loop Transfer Recovery

Recall that by invoking certainty equivalence, the LQ controller design can be carried out under the assumption of full state availability. Then, once the final design is complete, the full state feedback may be replaced with a Kalman filter. Intrinsic to the LQ full-state feedback controllers are certain guaranteed minimal stability robustness properties at design conditions (13; 22). Further, as discussed in the previous section, robustness of the full state system to unmodeled plant dynamics can be accomplished using implicit modeling techniques. However, this overall robustness achieved in the full state design is found to be degraded when the Kalman filter is incorporated into the controller (5; 22). In this section a method known as Loop Transfer Recovery (LTR) will be introduced, which is able to accomplish asymptotic recovery of the full state robustness characteristics. Unless otherwise stated, the following development is taken from Reference 14; however, for a more detailed discussion the reader is directed to References 5, 9, and 22.

The LTR method is a means of "tuning" the Kalman filter used to provide state estimates in order to improve the robustness characteristics of the overall closed loop system. The LTR technique cannot improve the robustness of the controller beyond that of the full state feedback system; however, it can provide robustness up to and asymptotically including that of the original system. The objective of the LTR technique is to tune the Kalman filter in such a manner that the return difference function (9; 22) of the filter-based controller becomes asymptotically equal to the return difference function of the full-state-feedback-system. Recall the dynamical system equation

$$\dot{\underline{x}}(t) = \underline{F}\underline{x}(t) + \underline{B}\underline{u}(t) + \underline{G}\underline{w}(t) \qquad (4\text{-}3)$$

For a continuous-time, minimum phase system the LTR tuning is based on the following equation:

$$\underline{Q}_{LTR}(q) = \underline{Q}_{o} + q^{2}\,\underline{B}\,\underline{V}\,\underline{B}^{T} \qquad (4\text{-}4)$$

where $\underline{Q}_o$ is the strength of the dynamics noise in the filter before LTR tuning is applied, and $\underline{V}$ is any positive definite matrix (commonly chosen to be the identity matrix $\underline{I}$). The robustness of the LTR-tuned filter/controller approaches that of the full-state-feedback-system as the scalar q approaches infinity. The physical interpretation of Equation (4-4) is a process of injecting noise into the input channels of the system, i.e. additional white noise is added to the system model at the same points of entry as used by the control inputs $\underline{u}(t)$.

The discrete-time formulation of the LTR technique is an extension of the continuous-time case made by any one of the following methods: 1) completing the entire controller design in the continuous-time domain, including the LTR tuning, and discretizing the resulting design; 2) performing the LTR tuning on the continuous-time system, obtaining the equivalent discrete time system model, and proceeding to apply LQG design techniques to the discrete time system; or 3) obtaining the discrete-time model of the system, carry out the LQG design, and then inject white noise into the entry points of $\underline{u}(t)$. The white noise which is injected into the system can be replaced by a time correlated noise if robustness enhancement is required over a particular frequency range (9); however, this technique will not be pursued in this thesis. Also, it has been shown that a dual LTR tuning technique exists that recovers the full-state-feedback robustness characteristics of the system by adjusting the weighting matrices of the LQ regulator (21).

However, for the purposes of this thesis, attention will be limited to the technique introduced in Reference 22.

Although the LTR tuning has been shown to provide more enhanced robustness as the scalar q is increased, it also allows more noise to be passed through the system. This results in degraded performance at design conditions as compared to the non-LTR tuned controller. Therefore, engineering judgment must be used to determine the proper q which provides the desired balance between system robustness and controller performance at design conditions.

4.4 Summary

In this chapter, two robustness enhancement techniques were introduced. First, implicit model following was presented. This scheme was shown to provide robustness improvement in the face of unmodeled system dynamics by allowing the designer to invoke pole placement, system bandwidth rolloff, and eigenvector orthogonalization techniques directly in an LQG-synthesized controller. Secondly, the LTR technique was shown to be a tuning method which allows the full state robustness characteristics to be recovered asymptotically when a filter is introduced into the loop to provide state estimates.

The preceding development is meant to provide engineering insight, not detailed mathematics. The reader who desires a more detailed treatment of the robustness enhancement techniques which are applicable to LQG designs is directed to References 5, 9, 13, 16, 21, and 22.

V. Experimental Methods and Results

5.1 Introduction

An analysis of the design work accomplished in the course of this thesis effort, along with a presentation of the results obtained, will be the topic of this chapter. The next four sections of this chapter derive the models for the longitudinal mode dynamics of the STOL F-15 (see Chapter III), followed by a section on pitch pointing controller design. Section 5.7 will present the methods used to test control system robustness, both with and without a Kalman filter embedded in the control system. Finally, Section 5.8 summarizes the preceding sections.

The reader is assumed to have read Chapters III and IV, and at least "scanned" Chapter II (especially Sections 2.3-2.5) before embarking on this chapter. This preparation is necessary since much of the material discussed in these previous chapters will be referenced in this chapter without further explanation.

5.2 Detailed Portrayal of Design Model

As discussed in Chapter III, the linearized longitudinal equations of motion for the STOL F-15 are of the following form:

$$\underline{\dot{x}}(t) = \underline{Fx}(t) + \underline{Bu}(t) \tag{5-1}$$

where the state vector $\underline{x}(t)$ consists of velocity $u(t)$, pitch rate $q(t)$, angle of attack $\alpha(t)$, and pitch angle $\theta(t)$. Thus Equation (5-1) becomes

$$\begin{bmatrix} \dot{u}(t) \\ \dot{q}(t) \\ \dot{\alpha}(t) \\ \dot{\theta}(t) \end{bmatrix} = \begin{bmatrix} F_{11} & F_{12} & F_{13} & F_{14} \\ F_{21} & F_{22} & F_{23} & F_{24} \\ F_{31} & F_{32} & F_{33} & F_{34} \\ F_{41} & F_{42} & F_{43} & F_{44} \end{bmatrix} \begin{bmatrix} u(t) \\ q(t) \\ \alpha(t) \\ \theta(t) \end{bmatrix} + \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ B_{31} & B_{32} & B_{33} \\ B_{41} & B_{42} & B_{43} \end{bmatrix} \begin{bmatrix} \delta_C(t) \\ \delta_S(t) \\ \delta_T(t) \end{bmatrix} \tag{5-2}$$

where δ_C, δ_S, and δ_T are the inputs which drive the canard, stabilator, and throttle, respectively. The numerical value of the coefficients in Equation (5-1) for a STOL F-15 at a velocity of mach 0.9 and an altitude of 20,000 feet are computed by STOLCAT using aerodynamic data provided by McAir to be

$$\begin{bmatrix} \dot{u}(t) \\ \dot{q}(t) \\ \dot{\alpha}(t) \\ \dot{\theta}(t) \end{bmatrix} = \begin{bmatrix} -0.18 & -20 & 27.9 & -32.19 \\ -.23\text{x}10^{-3} & -1.99 & 10.81 & 0 \\ -.35\text{x}10^{-6} & .999 & -1.45 & -.73\text{x}10^{-3} \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} u(t) \\ q(t) \\ \alpha(t) \\ \theta(t) \end{bmatrix}$$

$$+ \begin{bmatrix} 1.43 & -9.96 & 45 \\ 11.72 & -19.96 & 0 \\ -.045 & -.19 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_C(t) \\ \delta_S(t) \\ \delta_T(t) \end{bmatrix} \tag{5-3}$$

For numerical values of the $\underline{F}$ and $\underline{B}$ matrices in Equation (5-1) at flight conditions other than that of Equation (5-3), see Appendix B.

For reasons which will be discussed in Section 5.4, the output variables were chosen to be

$$\underline{Y}(t) = \underline{Cx}(t) = \begin{bmatrix} \gamma(t) \\ \theta(t) \\ q(t) \end{bmatrix} \qquad (5\text{-}4)$$

where $\gamma(t)$ is flight path angle. Using the approximation

$$\theta \cong \alpha + \gamma \qquad (5\text{-}5)$$

the $\underline{C}$ matrix of Equation (5-4) is determined to be

$$\underline{C} = \begin{bmatrix} 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix} \qquad (5\text{-}6)$$

(Note that during the course of the research conducted in conjunction with this thesis, a nonlinear dynamical equation of motion for the STOL F-15 was derived which allowed for the simultaneous control of both thrust and nozzle deflection. See Appendix D for a discussion of this model and problems encountered in its implementation for controller design.)

All controllers presented in this chapter are designed based on this four-state design model in order to keep controller complexity to a minimum. However, for performance analysis of these designs, a more complex "truth model" will need to be derived. This truth model is the subject of the following section in this chapter.

5.3 Truth Model Specification

The truth model derivation begins with the four-state aircraft model presented in Section 5.2. In order to provide a more complete aircraft model, actuator dynamics states were augmented to these original system states. The dynamics associated with the canard and the stabilator actuators were given by McAir to be of the following form:

$$\frac{\delta_S(s)}{e_{\delta_S}(s)} = \frac{\delta_C(s)}{e_{\delta_C}(s)} = \frac{\delta(s)}{e_{\delta}(s)} = \frac{30.62(272.7)^2}{(s+30.62)(s^2+277.2s+74474)} \qquad (5\text{-}7)$$

where e_δ is the commanded value of δ. The associated state space representation is:

$$\begin{bmatrix} \dot{\delta}(t) \\ \ddot{\delta}(t) \\ \dddot{\delta}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -2.3\text{x}10^{-6} & -8.3\text{x}10^{-5} & -307.8 \end{bmatrix} \begin{bmatrix} \delta(t) \\ \dot{\delta}(t) \\ \ddot{\delta}(t) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 2.3\text{x}10^{6} \end{bmatrix} e_\delta(t) \qquad (5\text{-}8)$$

Also, first order actuator dynamics for the throttle dynamics were approximated using the first order lag response

$$\frac{\delta_T(t)}{e_{\delta_T}(s)} = \frac{20}{s + 20} \qquad (5\text{-}9)$$

or, in state space form

$$\dot{\delta}_T(t) = \frac{-1}{20}\,\delta(t) + 20\,e_{\delta_T}(t) \qquad (5\text{-}10)$$

With these states augmented to the original system states, the resulting truth model is shown in Figure 5.1 on the next page.

In order to accomplish a reduction in the complexity of this truth model to lessen the computational burden with minimal impact on truth model adequacy, the third order actuator dynamics associated with the canard and stabilator are replaced with second order approximations (10) of the form:

$$\frac{\delta_S(s)}{e_{\delta_S}(s)} = \frac{\delta_C(s)}{e_{\delta_C}(s)} = \frac{\delta(s)}{e_{\delta}(s)} = \frac{8356.2}{s^2+303.52s + 8356.2} \qquad (5\text{-}11)$$

When represented in state space form Equation (5-11) becomes:

$$\begin{bmatrix} \dot{\delta}(t) \\ \ddot{\delta}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -8356.2 & -303.52 \end{bmatrix} \begin{bmatrix} \delta(t) \\ \dot{\delta}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 8356.2 \end{bmatrix} e_{\delta}(t) \qquad (5\text{-}12)$$

As can be seen from Figure 5.2, this is an accurate approximation over the bandwidth of interest in the system. The

$$\begin{bmatrix} \dot{u} \\ \dot{q} \\ \dot{\alpha} \\ \dot{\theta} \\ \dot{\delta}_C \\ \ddot{\delta}_C \\ \dddot{\delta}_C \\ \dot{\delta}_S \\ \ddot{\delta}_S \\ \dddot{\delta}_S \\ \dot{\delta}_T \end{bmatrix} = \left[\begin{array}{cccc|ccc|ccc|c} F_{11} & F_{12} & F_{13} & F_{14} & B_{11} & 0 & 0 & B_{12} & 0 & 0 & B_{13} \\ F_{21} & F_{22} & F_{23} & F_{24} & B_{21} & 0 & 0 & B_{22} & 0 & 0 & B_{23} \\ F_{31} & F_{32} & F_{33} & F_{34} & B_{31} & 0 & 0 & B_{32} & 0 & 0 & B_{33} \\ F_{41} & F_{42} & F_{43} & F_{44} & B_{41} & 0 & 0 & B_{42} & 0 & 0 & B_{43} \\ \hline 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -2.3\text{x}10^6 & -8.3\text{x}10^5 & -307.8 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2.3\text{x}10^6 & -8.3\text{x}10^5 & -307.8 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -20 \end{array}\right] \begin{bmatrix} u \\ q \\ \alpha \\ \theta \\ \delta_C \\ \dot{\delta}_C \\ \ddot{\delta}_C \\ \delta_S \\ \dot{\delta}_S \\ \ddot{\delta}_S \\ \delta_T \end{bmatrix}$$

$$+ \left[\begin{array}{ccc} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ 0 & 0 & 0 \\ 2.3\text{x}10^6 & 0 & 0 \\ \hline 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 2.3\text{x}10^6 & 0 \\ \hline 0 & 0 & 20 \end{array}\right] \begin{bmatrix} e_{\delta_C} \\ e_{\delta_S} \\ e_{\delta_T} \end{bmatrix}$$

Fig. 5.1. Eleven-State Truth Model

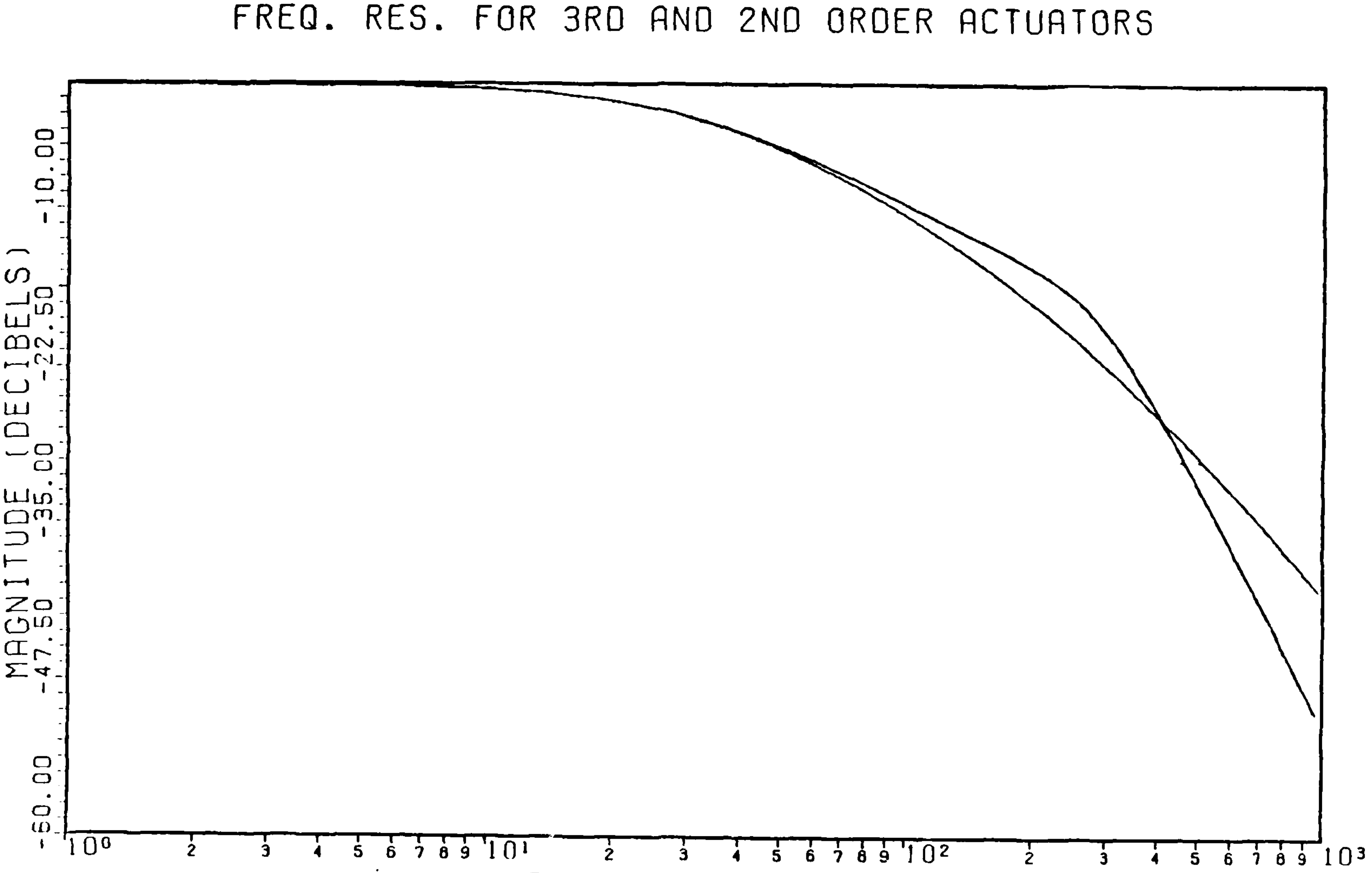


Fig. 5.2. Comparison of Second Order and Third Order Actuator Frequency Responses

truth model which results from the reduced order actuator dynamics approximation is shown in Figure 5.3 on the next page.

In order to yield the same output variables as those found in Equation (5-4), the $\underline{C}$ matrix for the truth model of Figure 5.3 is

$$\underline{C} = \left[\begin{array}{cccc|cc|cc|c} 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \tag{5-13}$$

Also, the $\underline{T}_{NT}$ matrix discussed in Chapter III, which is used to relate the states of the truth model to the states of the design model for CGTPIF performance evaluation (7; 16), is

$$\underline{T}_{NT} = \left[\begin{array}{cccc|cc|cc|c} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \tag{5-14}$$

As discussed in Chapter III, it will be this truth model which is used to represent the "real world" characteristics of the STOL F-15 in both linear and nonlinear computer analysis of the pitch pointing controller presented in Section 5.7 of this chapter. The nonlinearities of the latter analysis have to do with magnitude and rate saturations of the actuators.

$$\begin{bmatrix} \dot{u} \\ \dot{q} \\ \dot{\alpha} \\ \dot{\theta} \\ \dot{\delta}_C \\ \ddot{\delta}_C \\ \dot{\delta}_S \\ \ddot{\delta}_S \\ \dot{\delta}_T \end{bmatrix} = \left[\begin{array}{cccc|cc|cc|c} F_{11} & F_{12} & F_{13} & F_{14} & B_{11} & 0 & B_{21} & 0 & B_{13} \\ F_{21} & F_{22} & F_{23} & F_{24} & B_{21} & 0 & B_{22} & 0 & B_{23} \\ F_{31} & F_{32} & F_{33} & F_{34} & B_{31} & 0 & B_{23} & 0 & B_{33} \\ F_{41} & F_{42} & F_{43} & F_{44} & B_{41} & 0 & B_{24} & 0 & B_{34} \\ \hline 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -2356.2 & -303.5 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -2356.2 & -303.5 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -20 \end{array}\right] \begin{bmatrix} u \\ q \\ \alpha \\ \theta \\ \delta_C \\ \dot{\delta}_C \\ \delta_S \\ \dot{\delta}_S \\ \delta_T \end{bmatrix}$$

$$+ \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 8356.2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 8356.2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} e_{\delta_C} \\ e_{\delta_S} \\ e_{\delta_T} \end{bmatrix}$$

Fig. 5.3. Nine-State Truth Model

This ends the derivation of the aerodynamic data based aircraft models. It is important for the reader to realize that the design and truth models are functions of the airframe configuration of the STOL F-15 and the actuator dynamics associated with the servos which drive its control surfaces; however, the models which are presented in the following two sections are not directly tied to the physical properties of the aircraft itself. Instead, they are derived by the designer based on the desired performance characteristics of the controller.

5.4 Explicit Model Derivation

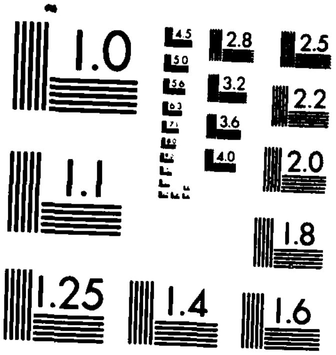
Unlike the two previously presented models, the explicit model is not based on provided aerodynamic data. Instead, it is completely determined by the control system designer based, in this case, on the desired aircraft handling qualities. The first step which was taken in deriving this model was to define the system outputs needed in order to accomplish the pitch pointing maneuver. Since pitch pointing entails decoupling pitch angle and flight path, these two angles are obvious candidates for output variables. As stated in Chapter III, the design software (7; 16) requires that the number of outputs be equal to the number of inputs, which in this case is 3. Therefore, one output variable remains to be chosen. Based on previous work in this area, the third output variable was chosen to be pitch rate, since this has been shown to increase controller stability (7; 16).

The next step that was taken after the output variables were defined, was to establish the desired trajectories for these variables to follow. It was to this end that the explicit model was designed. For a pitch pointing maneuver, the flight path requires no states in the explicit model, since this output variable is to be commanded to zero for all time. This can be simply accomplished by "zeroing out" the row in the explicit model output matrix which corresponds to the flight path variable. The desired trajectory for the pitch angle was expressed using a second order explicit model dynamics of the following form:

$$\begin{bmatrix} \dot{x}(t) \\ \ddot{x}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega_N & -\zeta\omega_N \end{bmatrix} \begin{bmatrix} x(t) \\ \dot{x}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \omega_N \end{bmatrix} K\delta_{CMD}(t) \qquad (5\text{-}15)$$

where $\delta_{CMD}(t)$ is the commanded pitch angle step change (0.035 radians or 2 degrees for the designs analyzed in this study) and $x(t)$ is the desirable model-achieved pitch angle. Note that ζ and ω_N were chosen to be 0.5 and 3 rad/sec respectively, in order to provide the type damping and natural frequency response desired by fighter aircraft pilots (8). In the first attempt to form the explicit model for pitch rate, a first order model was used; however, when excited by a step input (the only type of command input available in CGTPIF), the first order model commanded a non-zero steady state output to the system. Although the dynamics of the system eventually drove the actual pitch rate to an extremely small value over the 6 second time period examined for the aircraft responses in the combat mode of operation, they were not exactly zero. This small steady state error in pitch rate resulted in a constant rate of change in the control surface deflection and overall long-term system instability. In an attempt to alleviate this problem, a second order model was introduced for pitch rate. This model is of the same basic form as Equation (5-15). However, instead of taking $\dot{x}$ as the ideal trajectory to be tracked, x was used since this value would have a steady state value of exactly zero. Through an iterative process, the entry in the $\underline{B}$ matrix which corresponds to the input to the pitch rate model was adjusted to achieve the best tracking of the command model outputs by the outputs of the actual system. It was observed that usually the best results were obtained when the value of the entry in the $\underline{B}$ matrix for a second order system was about 10-15 percent larger than the systems natural damping frequency, ω_N. The damping of the second order pitch rate model was chosen to be 0.707 in order to minimize settling time and, as discovered by empirical observation, the natural frequency which achieved best overall tracking by the actual system was the same as the natural frequency of

AD-A164 188 AN LQG UP-AND AWAY FLIGHT CONTROL DESIGN FOR THE STOL 2/3
F-15 AIRCRAFT(U) AIR FORCE INST OF TECH
WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING
UNCLASSIFIED R A HOUSTON DEC 85 AFIT/GE/ENG/85D-21 F/G 1/4 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

the pitch angle model, i.e., 3 rad/sec. The final form of the explicit model based on the above derivation is:

$$\begin{bmatrix} \dot{x}_1(t) \\ \ddot{x}_1(t) \\ \hline \dot{x}_2(t) \\ \ddot{x}_2(t) \end{bmatrix} = \left[\begin{array}{cc|cc} 0 & 1 & 0 & 0 \\ -9 & -4.32 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 \\ 0 & 0 & -9 & -3 \end{array}\right] \begin{bmatrix} x_1(t) \\ \dot{x}_1(t) \\ \hline x_2(t) \\ \dot{x}_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 10 \\ \hline 0 \\ 9 \end{bmatrix} \delta_{CMD}(t) \tag{5-16}$$

with the model output matrix

$$\underline{y}_m(t) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ \dot{x}_1 \\ \hline x_2 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} \gamma_{IDEAL}(t) \\ \theta_{IDEAL}(t) \\ q_{IDEAL}(t) \end{bmatrix} \tag{5-17}$$

Figure 5.4, next page, is a plot of the time histories of these "ideal" responses.

5.5 Implicit Model Derivation

Like the explicit model discussed in the previous section, the implicit model is also completely dictated by the designer based on the specifications to be met by the final control design. For the purposes of the design accomplished in this thesis effort, the implicit model was constructed to provide system robustness in the face of unmodeled high frequency dynamics of the system. This robustness enhancement is basically a high frequency "roll off" effect (see Section 4.2).

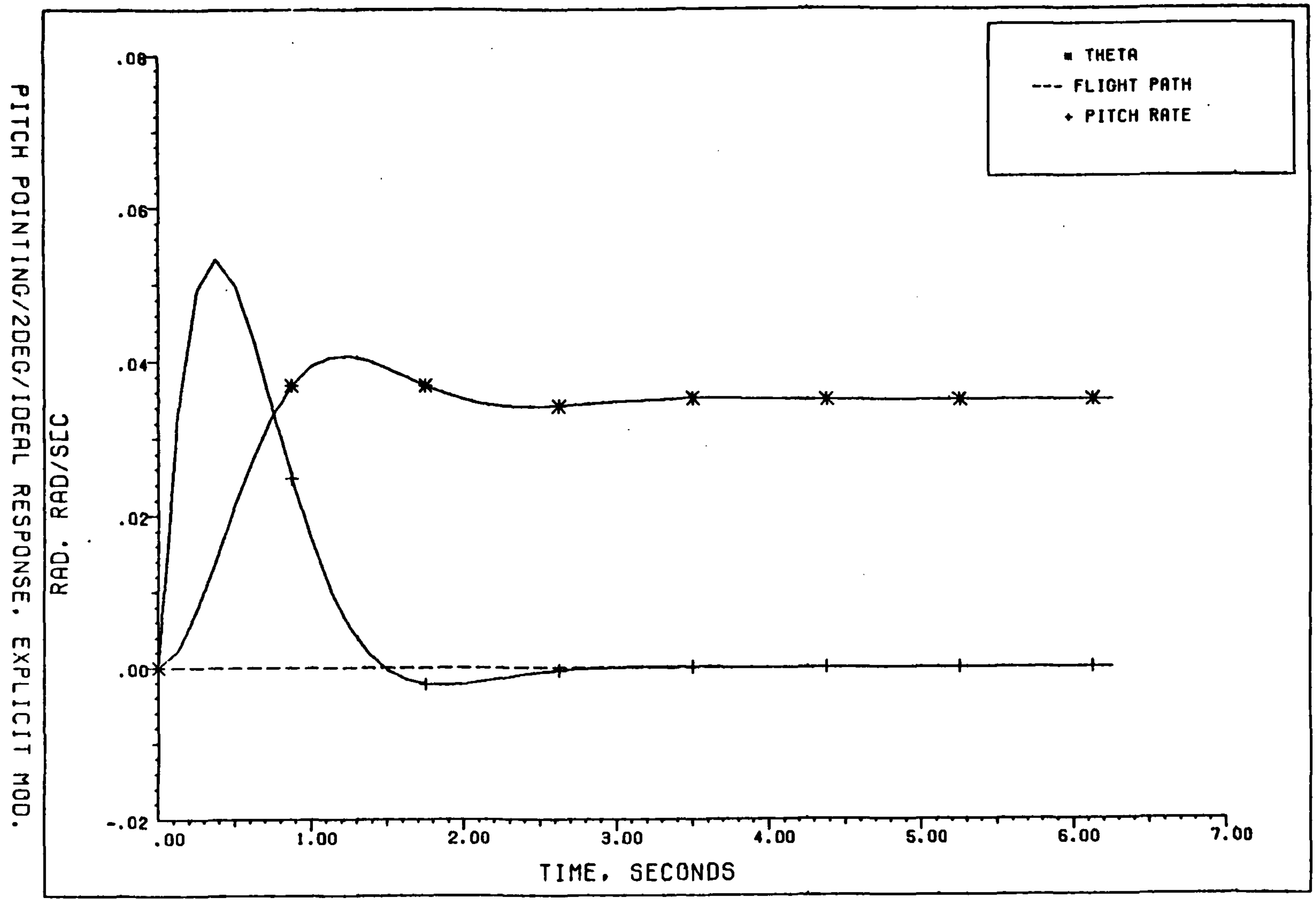


Fig. 5.4. Ideal Explicit Model Aircraft Responses

At the points in the envelope which were considered in this study (see Section 3.4), aside from Mach 0.3 at 20,000 ft as will be discussed shortly, the implicit model was used as a means to embed bandwidth reduction into the LQ design process. For all of these flight conditions, the implicit model used was of the following form:

$$\dot{\underline{x}}_I(t) = \begin{bmatrix} -0.1 & 0 & 0 \\ 0 & -0.1 & 0 \\ 0 & 0 & -0.1 \end{bmatrix} \underline{x}_I(t) \qquad (5\text{-}18)$$

As the magnitudes of the negative numbers along the diagonal of Equation (5-18) were reduced, the bandwidth of the system was also reduced; however, if these entries were made too small, the pitch rate displayed an oscillation. This problem can be explained by recalling that the implicit model can be thought of as a pole placement technique. Thus as the entries in the $\underline{F}_I$ matrix become smaller they drive the poles of the system towards the imaginary axis in the s-plane and induce neutral system stability.

Based on the above, it can be seen that the implicit model is desired to reduce the system bandwidth. While the ability to limit the bandwidth showed to be a useful application for the implicit model, it also showed some limited ability; in fact, to speed up the system when working at the mach 0.3 @ 20,000 ft. flight condition.

However, this "band-extension" characteristic showed only flight change in the overall response of the system at this point in the envelope. This is attributed to the fact that, while the implicit model can artificially slow down (i.e. reduce the bandwidth) of a fast system, it cannot speed up what would otherwise be a sluggish system without lowering the weights on the control amplitudes and rates. At best, it was found that the implicit model was useful to "push" the slower flight condition to the limits of its maximum bandwidth capability. This was accomplished by entering large negative entries along the major diagonal of the $\underline{F}_I$ matrix in the implicit model. For the controller designed at Mach 0.3 at 20,000 ft, it was found that the implicit model which achieved the fastest settling to an initial condition (where an initial condition is defined as steady state value for the state lasting from time equal to $-\infty$ to C^- and set to zero at time 0^+) without introducing instability was

$$\underline{\dot{x}}_I(t) = \begin{bmatrix} -24 & 0 & 0 \\ 0 & -24 & 0 \\ 0 & 0 & -24 \end{bmatrix} \underline{x}_I(t) \qquad (5\text{-}19)$$

where the off-diagonal terms are zero for reasons discussed in Section 4.2.

Based on the preceding discussion, it can be stated that the implicit model is extremely useful as a bandwidth reduction technique and somewhat useful as a "band extension" technique. However, for both applications care must be taken to limit the magnitude of the entries in the $\underline{F}$ matrix so as not to induce system instability.

5.6 Pitch Pointing Controller Design

Through use of the advanced control surface architecture used on the STOL F-15 (see Chapter III), certain maneuvers are possible which cannot be performed on a conventional F-15 aircraft. In the longitudinal mode, these maneuvers consist of pitch pointing and vertical translation, the former of which will be the subject of the control design presented in this section. The characteristic which separates these "enhanced" maneuvers from conventional maneuvers is the ability to use control surfaces to produce lift while driving moments to zero, i.e., to generate "direct lift" (15:69, 72).

Pitch pointing consists of pointing the aircraft's nose up or down while maintaining a fixed flight path. This maneuver is especially useful for air-to-air gunnery since gun sight errors may be easily nulled without changing the trajectory of the aircraft. In terms of angles used to describe the longitudinal orientation of the aircraft, pitch pointing consists of commanding the flight path angle to zero while simultaneously commanding a desired pitch angle. A second maneuver which exploits the direct lift capabilities of the STOL F-15 is vertical translation; however, this mode of flight is not addressed in this thesis.

The first step which was taken in the design of the pitch pointing controller was to determine the weighting matrices to be used in the definition of the cost function used to derive the PI control law. The initial attempt used the strategy that the inverse of the square of the maximum variation allowable in a particular variable being weighted would serve as the weighting on that variable (12). This approximation is considered reasonable due to the fact that, if all variables weighted in this manner reach maximum allowable values simultaneously, they will contribute equivalent amounts to the cost, so that the controller will expend equal amounts of effort on all channels. This weighting scheme is depicted in Equations (5-20) to (5-22) below:

$$\underline{W}_{\underline{u}_m}(i,i) = \frac{1}{[\text{Maximum Control Surface Deflection}]^2} \tag{5-20}$$

for the weights on the control magnitudes, as in Equation (2-19),

$$\underline{W}_{\underline{u}_R}(i,i) = \frac{1}{[\text{Maximum Control Surface Rate}]^2} \tag{5-21}$$

for the weights on the control rates, as in Equation (2-37), and

$$\underline{W}_y(i,i) = \frac{1}{[\text{Maximum Allowable Deviation in Output}]^2} \tag{5-22}$$

for the weights on the output magnitudes, as in Equation (2-18), where $\underline{W}(i,i)$ is the (i,i) element of the diagonal weighting matrix. From the information provided by McAir (20), the following rate and deflection limits were obtained for the canard and stabilator:

$$\text{Canard Position Limits} = -35°;\ +15° \tag{5-23}$$

$$\text{Canard Rate Limit} = 23°/\text{sec} \tag{5-24}$$

$$\text{Stabilator Position Limits} = -29°;\ +15° \tag{5-25}$$

$$\text{Stabilator Rate Limit} = 46°/\text{sec} \tag{5-26}$$

Based on Equations (5-23) to (5-26), the magnitude weightings for the input vector $\underline{u}(t)$, (recall that the entries of $\underline{u}$ consist of δ_C, δ_S, and δ_T) were determined to be

$$\underline{W}_{u_m} = \begin{bmatrix} 14.59 & 0 & 0 \\ 0 & 14.59 & 0 \\ 0 & 0 & 14.59 \end{bmatrix} \tag{5-27}$$

in Equation (2-19), and the rate weights were determined to be

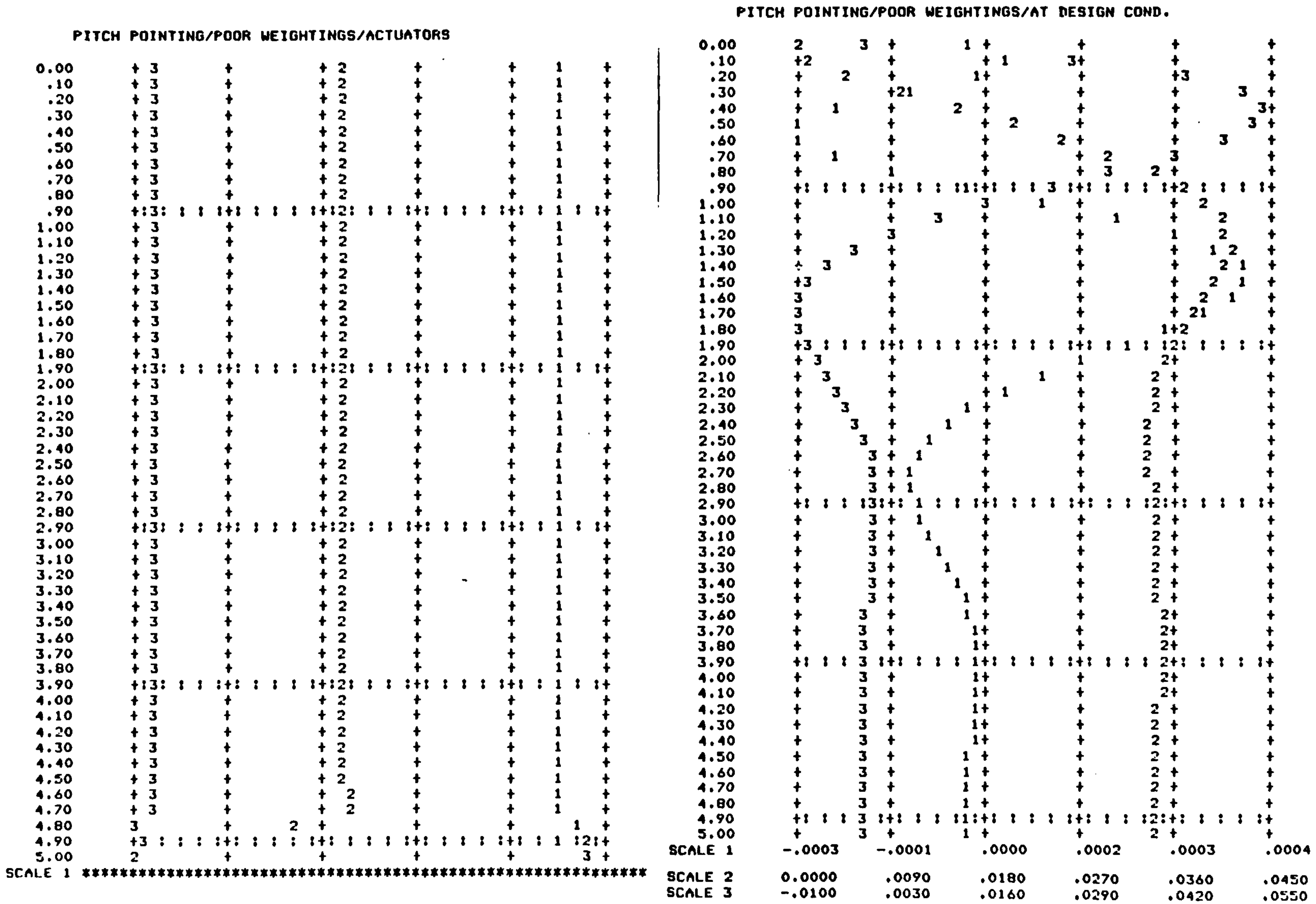
$$\underline{W}_{u_R} = \begin{bmatrix} 6.21 & 0 & 0 \\ 0 & 1.55 & 0 \\ 0 & 0 & 1.55 \end{bmatrix} \qquad (5\text{-}28)$$

For instance, the 1,1 element of W_{u_m} is $1/[.262 \text{ rad}]^2$ since this was the smaller of the upper and lower limits. Note that in Equations (5-27) and (5-28), the value for weightings on throttle input magnitude and throttle input rate were assumed to be approximately the same value as the other weightings in these weighting matrices. This assumption was made based simply on the lack of any quantitative information of the physical limitations of the throttle response for the STOL F-15 and bears further investigation.

In order to make a "first cut" derivation of the weighting matrix on the system outputs, y, a 10.47 milliradian (0.6 degrees) pitch point was considered. Assuming deviation of no more than 10 percent yielded a maximum allowable deviation of 1.047 milliradians (0.06 degrees). In order to obtain a weighting on flight path, it was assumed that the flight path should be allowed ten times less deviation than pitch angle, i.e., 0.1047 milliradians (0.006 degrees). Similarly, the pitch rate weight was initially set based on the weighting of pitch angle (since these states are linearly related through a derivative). Based on these assumptions the output variable weighting matrix becomes (recalling that the output variables are γ, θ, and q, respectively):

$$\underline{W}_y = \begin{bmatrix} 9.16\text{x}10^6 & 0 & 0 \\ 0 & 9.16\text{x}10^5 & 0 \\ 0 & 0 & 9.16\text{x}10^5 \end{bmatrix} \qquad (5\text{-}29)$$

The initial designs were based on the above weighting matrices and the explicit model of Equation (5-16). These controllers showed responses almost identical to those of the ideal responses when tested using a truth model which was identical to the design model for evaluation purposes; see Figure 5.5 next page (compare the upper portion of Figure 5.5 to the ideal response of Figure 5.4). (Admittedly, this is a questionable practice at best. However, for the purposes of this study all designs were initially tested against the four-state design model for the purpose of establishing the desired characteristics to embed in the explicit model to achieve desirable handling qualities. By no means were these evaluations meant to yield any stability information about the system.) However, when the truth model varied from the design model, the system designed on the basis of the above weightings was unstable, both with and without implicit model following (see the lower portion of Figure 5.5). The desire to achieve tight control over the output variables, as evidenced by Equation (5-29), proved


PITCH POINTING/POOR WEIGHTINGS/ACTUATORS

PITCH POINTING/POOR WEIGHTINGS/AT DESIGN COND.

Fig. 5.5. Aircraft Response Without (Upper) and With (Lower) Actuators/ Poorly Chosen Weighting Matrices ; 1 = flight path; 2 = pitch angle; 3 = pitch rate

to have disastrous effects on system robustness. It was found that the solution to this problem was to abandon the "inverse-maximum deviation-squared" solution to the weighting matrices in favor of a more intuitive approach.

The observation was made that, if the values of the weighting matrix determinants differed from one another appreciably (by approximately more than two orders of magnitude), the controller designed based on these weights would be unstable in the face of linear second-order actuator dynamics in the truth model. Therefore, the diagonal entries of all three weighting matrices were set equal to 0.1 and then were adjusted iteratively until the desired responses were achieved with a nine-state truth model. The thought progression for determining the weights which needed to be changed was as follows:

1. Starting from the "equal-weighting" condition of 0.1 along the diagonal of each weighting matrix, the system response was analyzed against the four-state truth model (again, this was not a test of system robustness, but rather, a test of acceptable system response in general). If a rate or position limit was violated in the actuators, then the weight associated with that rate or magnitude was increased in order to exert more restraint over that variable, and thus reduce its expenditure of control energy. Interestingly, the weights associated with the output variables needed no adjustment at this time since the

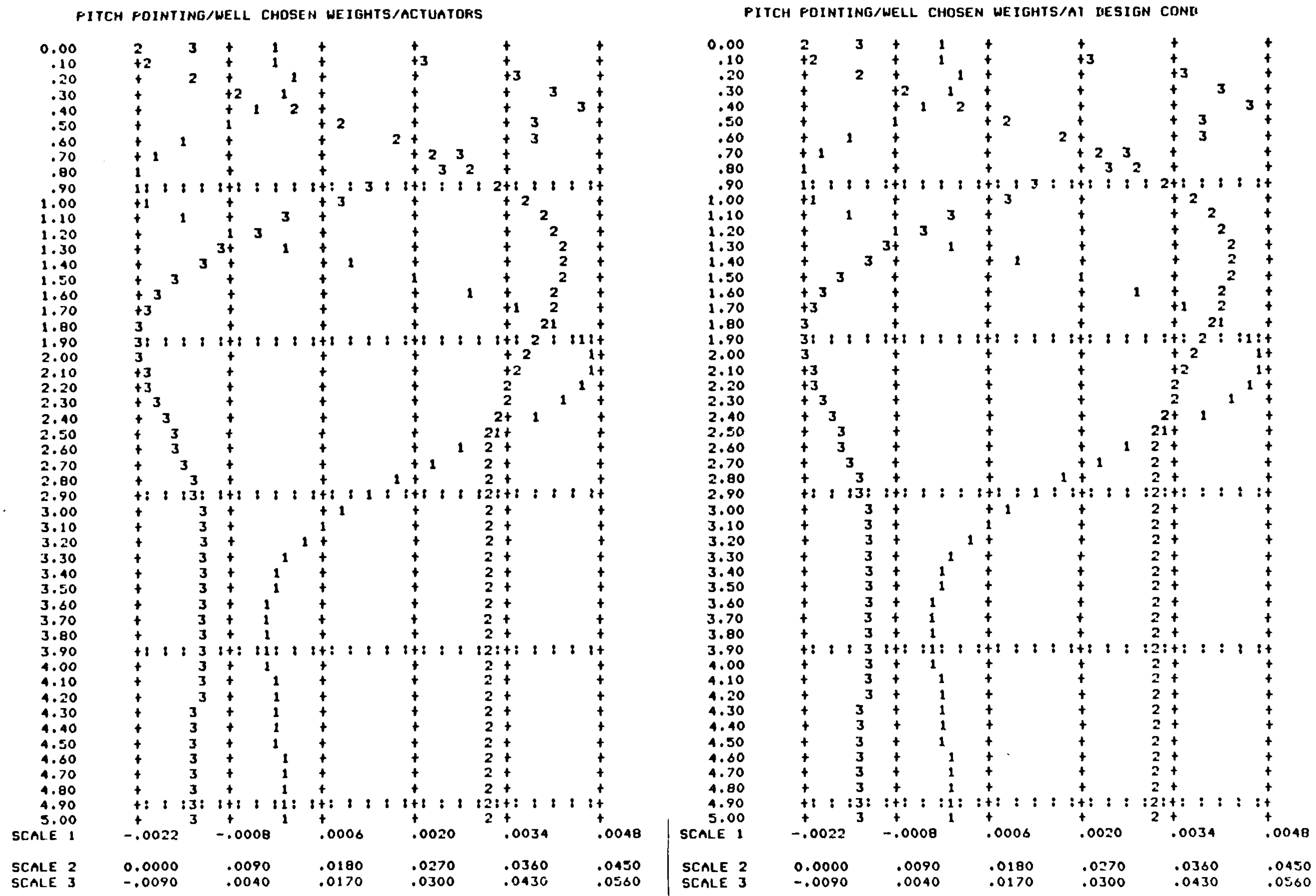


Fig. 5.6. Aircraft Response Without (Upper) and With (Lower) Actuators/ Well Chosen Weighting Matrices ; 1 = flight path; 2 = pitch angle; 3 = pitch rate

trajectories of the output variables were close to the ideal case depicted in Figure 5.4 (compare Figure 5.4 and the upper section of Figure 5.6).

2. At this point, the controller was tested against the 9-state linear truth model of Figure 5.3 and the system was found to be unstable. In order to stabilize the system, the weights on the output variables were reduced to allow for less "tight control" of these variables under "off-design" conditions. It was found that this procedure did indeed stabilize the system in the face of actuator dynamics; see the lower portion of Figure 5.6. It was also observed that the pitch rate channel was by far the best indicator of system stability; i.e., when the system was near instability, oscillations would appear in the pitch rate. Some degree of this oscillatory behavior continued on the pitch rate channel despite attempts to change the weighting of the controlled variables; see the lower section of Figure 5.7 on the next page.

3. Finally, the implicit model of Equation (5-18) was introduced into the controller design. The weightings on magnitude and rate variations were determined using the same type of procedure used in step 1 to be:

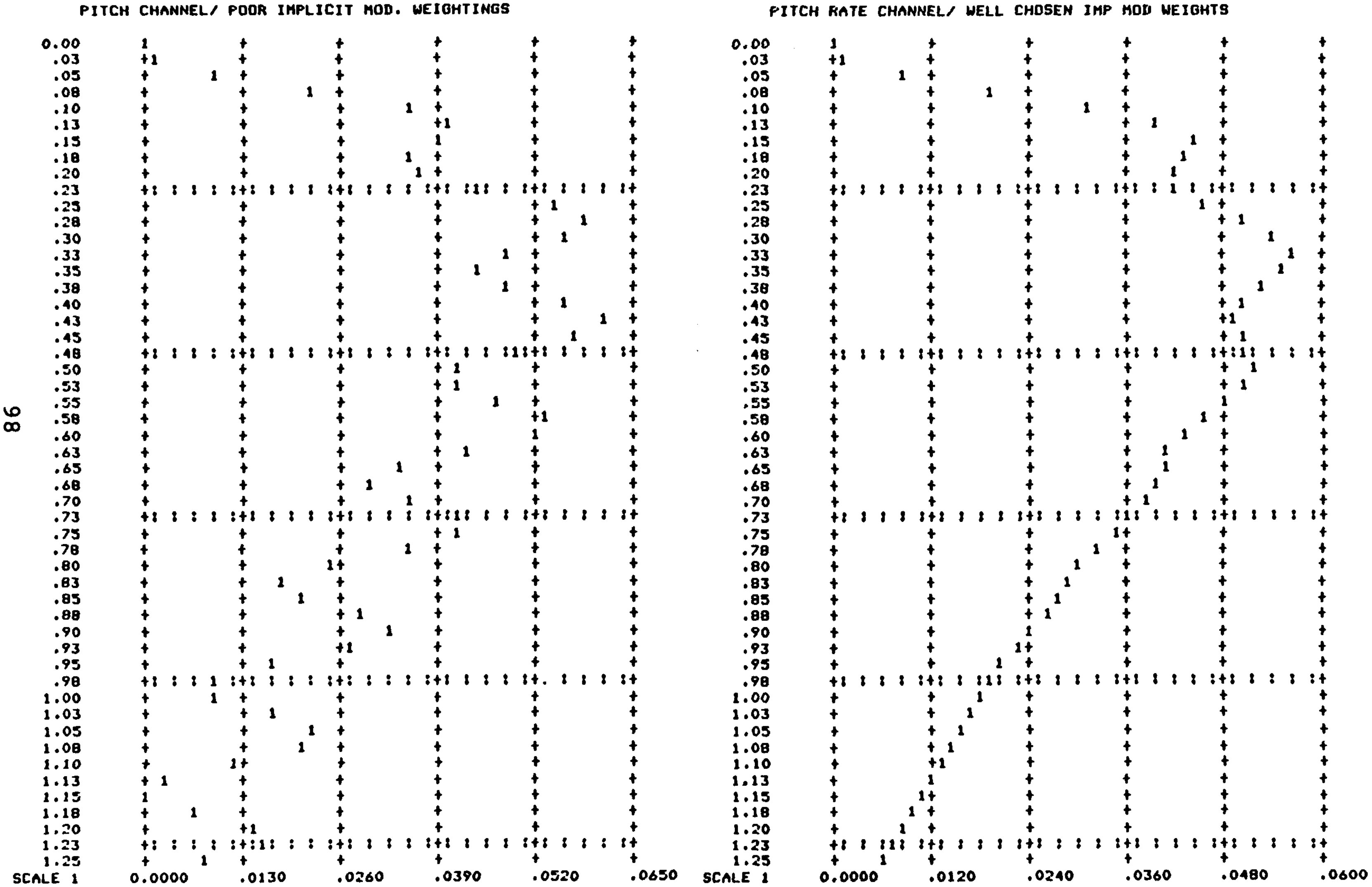


Fig. 5.7. Pitch Rate Channel Poor Implicit Model Weighting (Lower)/Well Chosen Implicit Model Weightings (Upper)

$$\underline{W}_{u_{mI}} = \begin{bmatrix} .02 & 0 & 0 \\ 0 & .02 & 0 \\ 0 & 0 & .02 \end{bmatrix} \tag{5-30}$$

$$\underline{W}_{u_{RI}} = \begin{bmatrix} .002 & 0 & 0 \\ 0 & .002 & 0 \\ 0 & 0 & .002 \end{bmatrix} \tag{5-31}$$

respectively. The criterion for acceptable performance was reduced ringing in the pitch rate channel when tested against the nine-state truth model, as shown in the upper section of Figure 5.7.

Based on steps 1 and 2, the weighting matrices of Equations (5-27) to (5-29) were adjusted to be (see Figure 5.6 and the upper portion of Figure 5.7 for the aircraft responses using these weightings):

$$\underline{W}_{u_{m}} = \begin{bmatrix} .02 & 0 & 0 \\ 0 & .02 & 0 \\ 0 & 0 & .02 \end{bmatrix} \tag{5-32}$$

$$\underline{W}_{u_{R}} = \begin{bmatrix} .008 & 0 & 0 \\ 0 & .008 & 0 \\ 0 & 0 & .008 \end{bmatrix} \tag{5-33}$$

and

$$\underline{W}_y = \begin{bmatrix} .08 & 0 & 0 \\ 0 & .05 & 0 \\ 0 & 0 & .01 \end{bmatrix} \tag{5-34}$$

The above process was carried out for the four flight conditions presented in Section 3.4, and stabilized controllers were obtained at these flight conditions when tested against a nine-state truth model. All designs were achieved using the same weighting matrices, explicit model, and implicit model (except for the implicit model used at Mach 0.3 at 20,000 ft., as discussed in Section 5.5). Therefore, the results presented for this thesis will be those at the "nominal" flight condition of Mach 0.9 at 20,000 ft., since this is representative of both the results and design methodologies used at all flight conditions.

The final controller was designed based on the previously described weighting matrices and the implicit and explicit models of Equations (5-16) and (5-18). The resulting gain matrices (see Section 3.5, especially Equation (2-79)) were generated using CGTPIF (7; 16):

$$\underline{K}_x = \begin{bmatrix} .4319\text{x}10^{-5} & 5.643 & -235.8 & .3534 \\ .7038\text{x}10^{-5} & .4257 & -140.5 & .1653 \\ .806 & .3077 & -23.84 & .4177\text{x}10^{-1} \end{bmatrix} \tag{5-35}$$

$$\underline{K}_z = \begin{bmatrix} 9.066 & -8.844 & 5.981 \\ 5.325 & -5.665 & 3.234 \\ 1.375 & -.7268 & .979 \end{bmatrix} \tag{5-36}$$

$$\underline{K}_{xm} = \begin{bmatrix} -1.219 & 11.54 & -244.9 & -6.25 \\ -.2815 & 4.048 & -145.4 & -3.715 \\ -.7221\text{x}10^{-1} & 1.352 & -24.52 & -.6312 \end{bmatrix} \tag{5-37}$$

$$\underline{K}_{xu} = \begin{bmatrix} .6428 \\ -.1105 \\ .8416\text{x}10^{-2} \end{bmatrix} \tag{5-38}$$

The responses to a 0.035 radian (2 degrees) pitch point command for a controller based on Equation (2-79) with the gain matrices of Equations (5-35) to (5-38) are shown for both a four-state truth model, Figure 5.8, and a nine-state truth model, Figure 5.9, on the following pages. Note that in both cases the aircraft responses are close to those of Figure 5.4. However, for the nine-state case a slight ringing occurs in the pitch rate channel; this is due to the instabilities introduced by including the actuators in the truth model while maintaining a four-state design model. This ringing appears pronounced due to the common scaling of the output variables. However, this is actually an acceptable response as evidenced by the pitch angle and flight path channels; i.e., the pitch

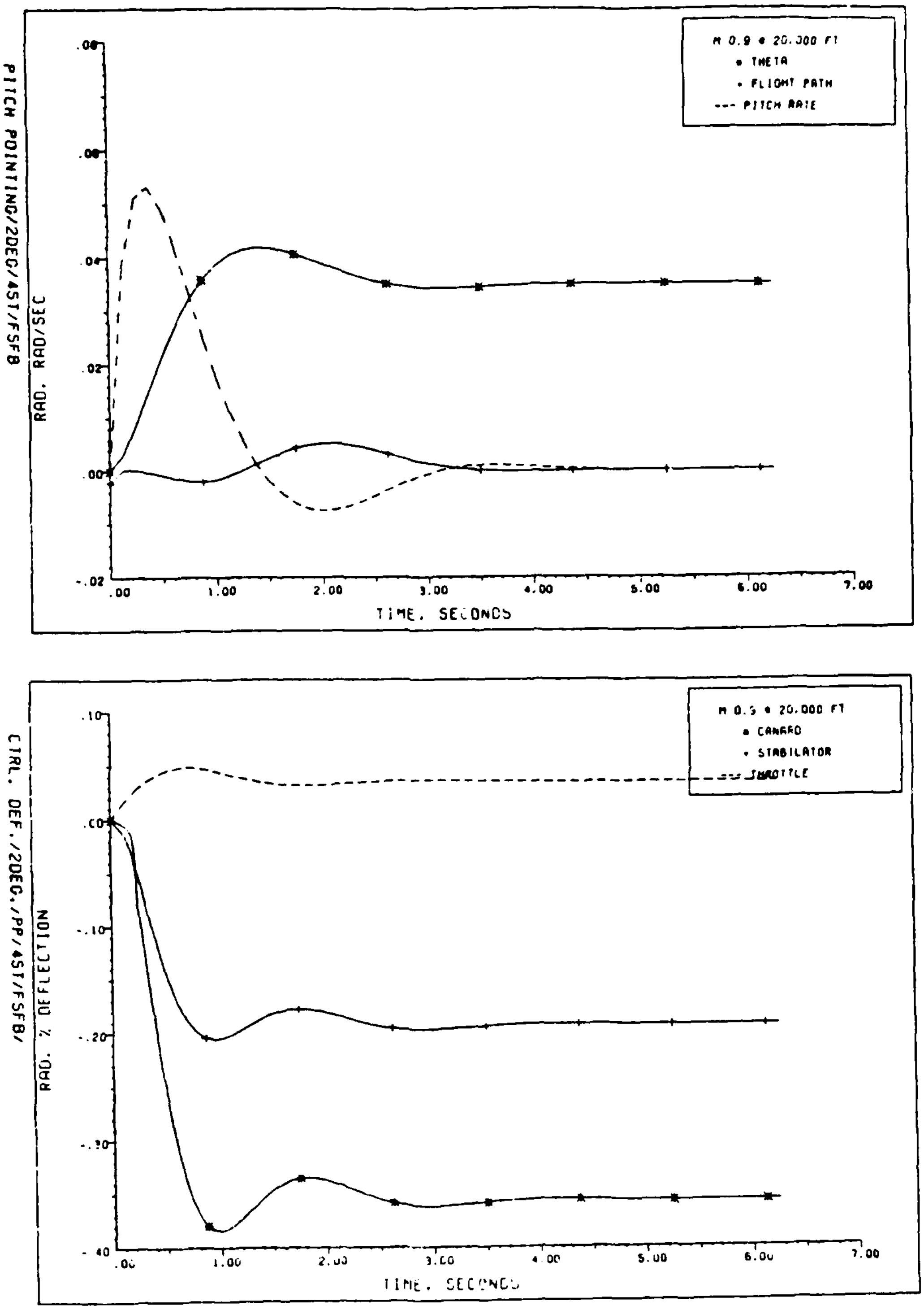


Fig. 5.8. Basic Aircraft Response
a. Aircraft responses; b. Control deflections

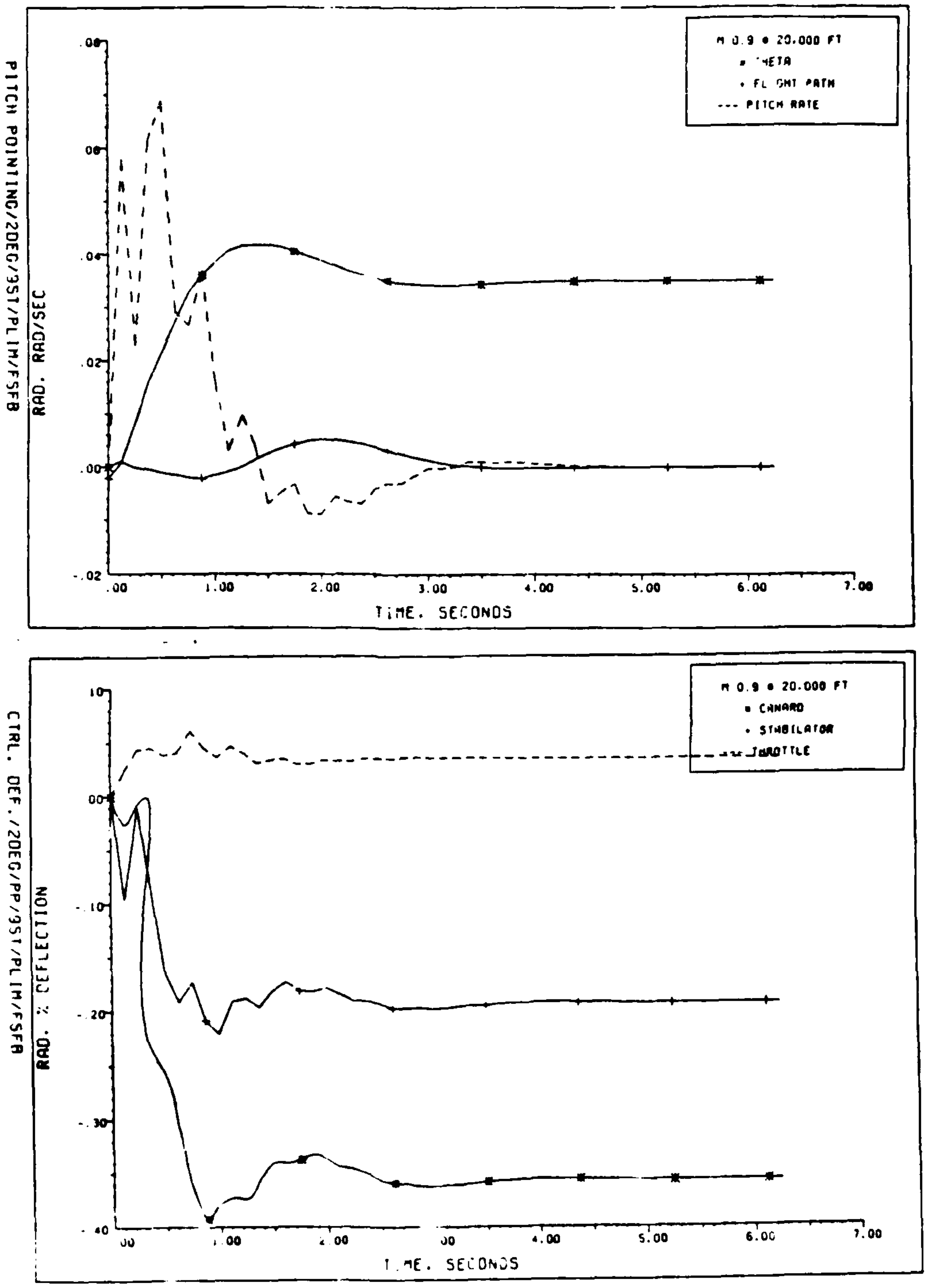


Fig. 5.9. Aircraft Response with Actuators
a. Aircraft responses; b. Control deflections

pointing maneuver itself is accomplished well despite the time lags introduced by the actuator dynamics.

Once the full-state controller design was complete, a constant-gain Kalman filter was separately designed to replace the assumption of full state availability with state estimates based upon noise-corrupted partial state availability. The measurements were modeled to be of the following form:

$$\underline{z}(t_i) = \underline{H}\,\underline{x}(t_i) + \underline{v}(t_i) \tag{5-39}$$

where

$$\underline{H} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{5-40}$$

The covariance of the measurement noise, $\underline{v}(t_i)$, was taken from Reference 7 to be

$$\underline{R} = \begin{bmatrix} .476\text{x}10^{-5} & 0 & 0 \\ 0 & .122\text{x}10^{-4} & 0 \\ 0 & 0 & .322\text{x}10^{-4} \end{bmatrix} \tag{5-41}$$

In this development, a term $\underline{G}w(t)$ was added to the dynamics state equation, with w(t) being zero-mean white Gaussian noise, independent of $\underline{v}(t_i)$, and of strength Q where

$$\underline{G} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \quad ; \quad Q = .001 \tag{5.42}$$

This form of $\underline{G}w(t)$ is motivated as a "first cut" at introducing the effects of wind buffeting on the aircraft (8). Using CGTPIF and PERFEVAL (7; 16; 17), the Kalman filter gain matrix was determined to be

$$K = \begin{bmatrix} -1.213 & .1626 & .4552\text{x}10^{-1} \\ .203 & .2704 & .1635 \\ .1173 & .2032 & .1024 \\ .69\text{x}10^{-1} & 4579\text{x}10^{-1} & .3001\text{x}10^{-1} \end{bmatrix} \tag{5-43}$$

(for a more detailed discussion of basic Kalman Filtering theory the reader is directed to Appendix E and Reference 11). The pitch pointing maneuver with the Kalman filter in the loop is shown in Figure 5.10 on the next page. In order to account for unmodeled time lags in the system, the controller was also tested using the suboptimal control law based on $\hat{\underline{x}}(t_i^-)$. This form of the control law bases the control input to the system on the Kalman filter's best estimate of the system states before the actual measurement is incorporated; this allows $\underline{u}(t_i)$ to be computed before time t_i to remove computational delay time of

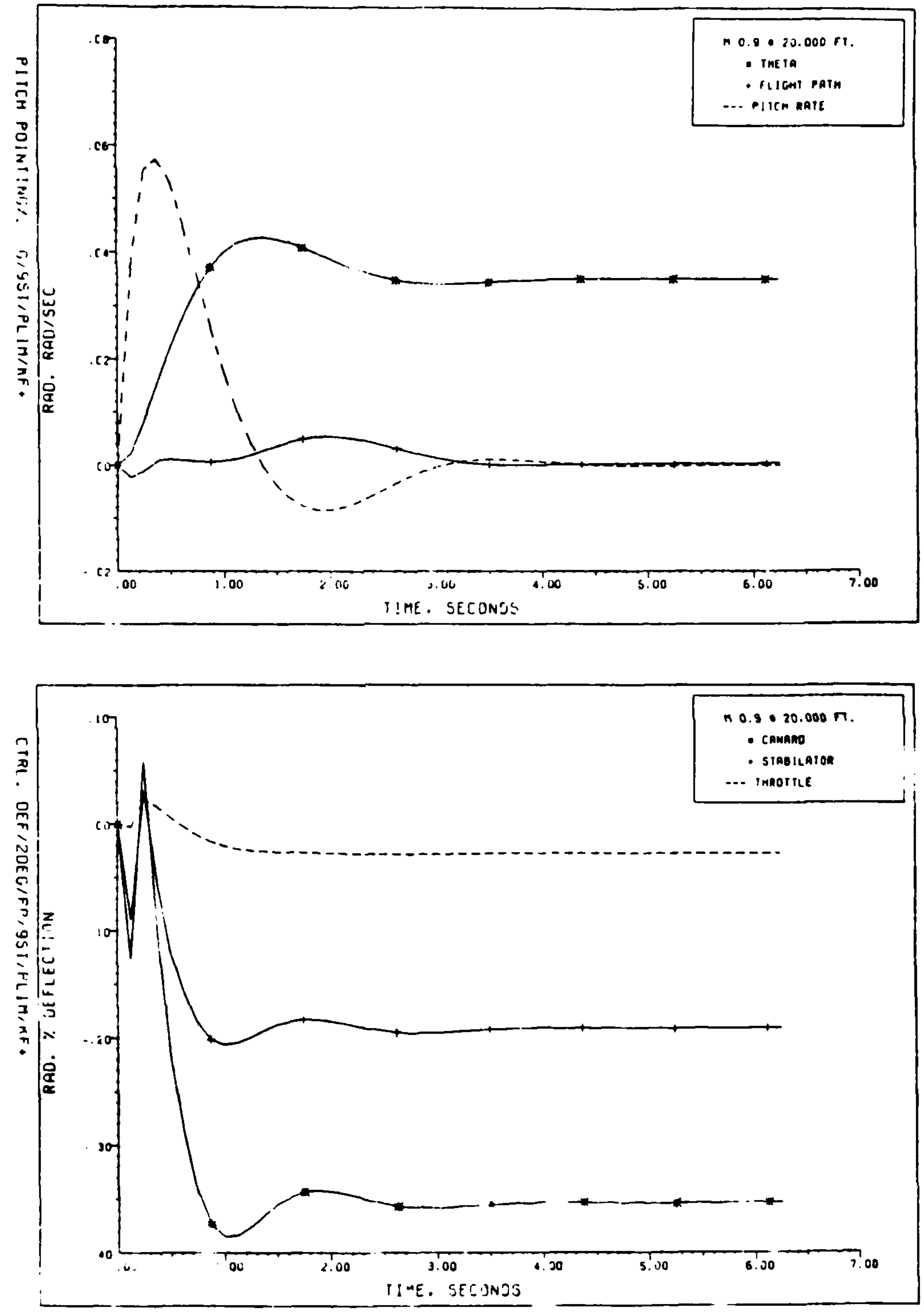


Fig. 5.10. Aircraft Response with Actuators, Position Limits, and Kalman Filter in the Loop

a. Aircraft responses; b. Control deflections

computing $\hat{\underline{x}}(t_i^+)$ and then the control $\underline{u}(t_i)$ based on this state estimate. This response is shown in Figure 5.11. Note that the responses both with and without the time delay in the system are essentially the same. Therefore, the impact of using this suboptimal control law is minimal. Also it is apparent that the Kalman filter does not degrade the ability of the controller to achieve the pitch pointing maneuver within a 3.5 second time interval.

The pitch pointing controller design presented in this section has been shown to display desirable characteristics when tested against a nine-state truth model, both with and without a Kalman filter embedded in the control law. It will be these system responses which will serve as a baseline for the robustness analysis discussed in the following section.

5.7 Robustness Analysis

The pitch pointing controller presented in the previous section was shown to be stable when evaluated against a nine-state truth model. In this section the controller evaluation will be extended using software generated specifically to include parameter variation and nonlinear effects such as control surface rate and position limits (see Appendix C). Before presenting the results of this robustness analysis, it is in order to state that the parameter variation technique which is employed in this

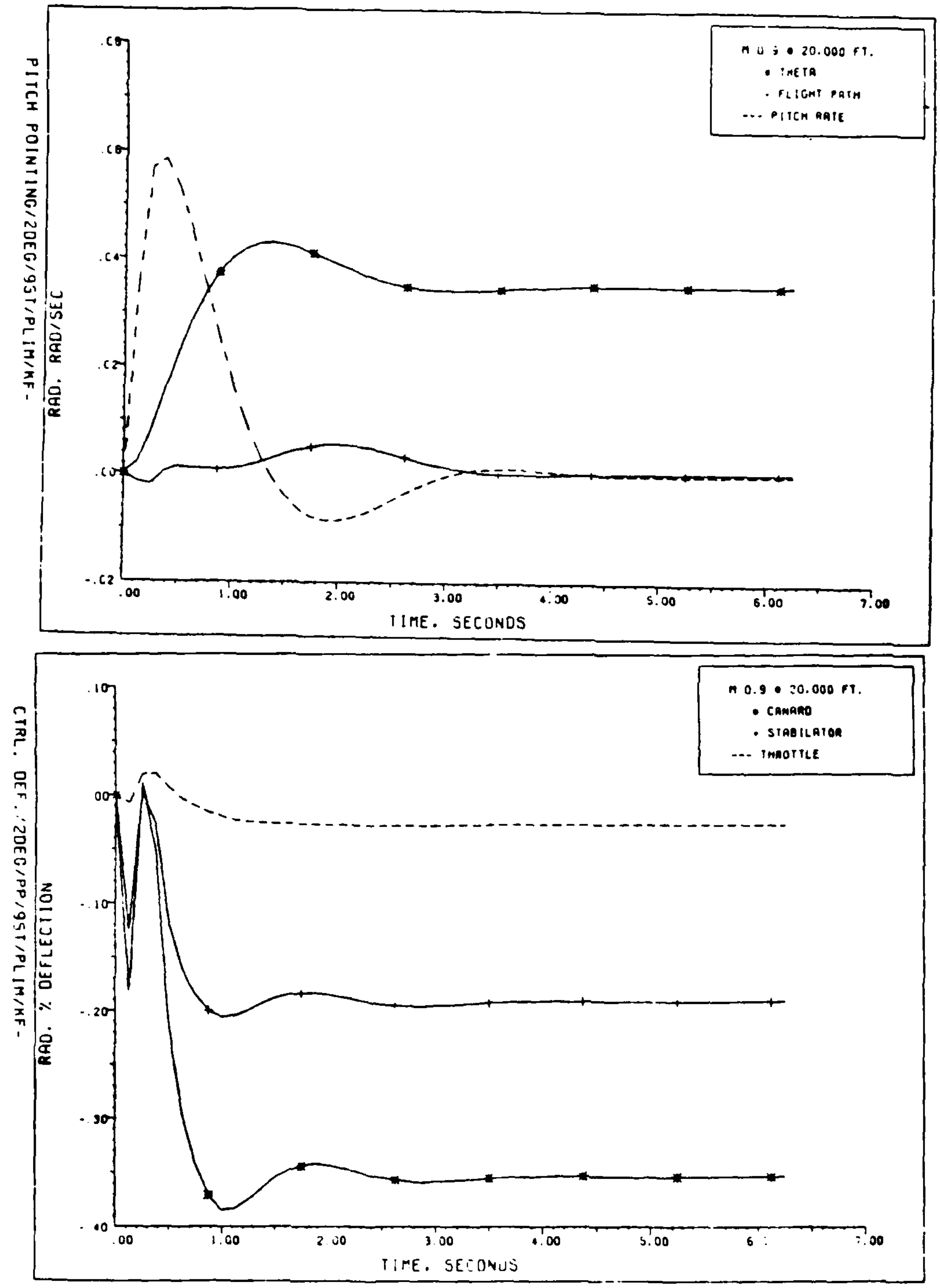


Fig. 5.11. Aircraft Response with Actuators, Position Limits, and Kalman Filter with the Loop/Control Based on $\underline{\hat{x}}(t_i^-)$

a. Aircraft responses; b. Control deflections

study is a somewhat limited tool for establishing system robustness. A more precise analytical method (5) would include structured singular value analysis to provide complete information about the system's robustness characteristics; however, this approach is beyond the scope of this thesis. The plotted data corresponding this robustness analysis are contained in Appendix F of this thesis. Also, it should be noted that, for the majority of the robustness analyses presented in this section, actuator rate limits are not included. This is due to the fact that severe instabilities were induced by limiting the rates to those specified in the previous section. However, at the time of this writing, there exists conflicting information about the actual rate limitations of these surfaces. In an attempt to make a "worst case" analysis, the lowest rate limits available are used for the robustness analysis conducted in this thesis.

The first nonlinear effect which was introduced into the system was a "sign swapping" routine to account for a problem which was identified with the linear model (1; 20) (see Appendix C for a discussion of the nonlinear analysis program ODEF15). The problem which existed was that, as the control surfaces passed through zero angle of attack relative to the aircraft, the sign of the coefficient in the $\underline{B}$ matrix which corresponded to the drag induced by the control surface failed to change sign. Effectively, this corresponded to a system which could increase thrust by increasing the deflection of the control surfaces. Of course this is a completely unrealistic situation, so the sign swapping routine is included in all of the following analyses without further discussion. Note however that, since velocity is not specifically controlled in this design, airspeed is not held constant in steady state. Therefore, throttle deflection may actually drop in a maneuver which requires an increase in thrust in order to maintain airspeed (however, this drop is not due to the linear analysis problem of experiencing "negative drag" since the sign swapping routine was being employed while this problem was identified). This phenomenon is not a problem over the six-second period determined to be essentially the longest period that the pilot would fly "hands-off" in the flight envelope considered in this study, since the airspeed drops less than 1 percent over this time period. In fact it is common practice to eliminate velocity completely from the dynamical equations of motion using a short period approximation (6), thereby completely ignoring the controller impact on velocity.

The correct way to address this problem is obviously to place a quadratic weight on velocity and exploit the properties of the PI control structure (see Chapter II) to assure constant steady state airspeed. However, due to the limitations of the software (7), in order to augment velocity to the system outputs, an independent control must be augmented to the input vector. An attempt was made to introduce a new control input by incorporating the thrust vectoring nozzles; however, nonlinearities associated with this problem were severe and the approach was abandoned (see Appendix D for a derivation of this nonlinear model).

Another attempt to provide control of velocity was made by introducing velocity into the existing output vector. First pitch rate control was removed and replaced with velocity control. This approach robbed the system of a significant amount of its robustness characteristics. In fact, without controlling pitch rate the system could not even be stabilized against in the face of second-order actuator dynamics, thus validating the original assumption that pitch rate control would enhance system robustness. The second approach attempted to drive the linear combination of velocity and flight path angle to zero. This resulted in a system which could drive the combination to zero; however, both velocity and flight path angle displayed steady state error (as expected).

Since throttle was available as a system input, two ad-hoc techniques were derived to demonstrate that the throttle input could be used to maintain airspeed if needed in a four-input/four-output system. First an absolute value function was placed on the control input to the throttle:

$$e'_{\delta_T}(t) = \left| e_{\delta_T}(t) \right| \tag{5-44}$$

and $e'_{\delta_T}(t)$ then replaced $e_{\delta_T}(t)$ in Equation (5-9). This approach maintained airspeed for a two-degree pitch pointing maneuver; however, it was not practical for varying commanded pitch angles. In order to address this problem, the following form of throttle control was implemented:

$$e''_{\delta_T}(t) = -0.2\ e_{\delta_C}(t) \tag{5-45}$$

while the throttle command was made a linear function of stabilator deflection. The system responses using these ad-hoc techniques are shown in Figures F.1 and F.2 in Appendix F.

The first check of system robustness was to evaluate how well the system could withstand variations in the $\underline{F}$ matrix of Equation (2-4). In order to establish this, all entries were varied by a specific amount in the following form:

$$\dot{\underline{x}}(t) = (\underline{F}+\underline{\Delta F})\underline{x}(t) + \underline{B}u(t) \tag{5-46}$$

The system was found to be able to display acceptable response with the variation $\underline{\Delta F}$ being up to 100 percent of the matrix $\underline{F}$. This insensitivity to variation in $\underline{F}$ is logical since the original aircraft is unstable at this flight condition, thereby requiring that the controller suppress the actual dynamics of the system. Plots of 10 percent, 25 percent, 100 percent, and 200 percent increase in $\underline{F}$ in Appendix F (F.3 to F.6) show the trend of reduction in direct lift capability, creating flight path angle/pitch angle coupling as $\underline{\Delta F}$ increases. Also note that at 200 percent variation in $\underline{F}$, Figure F.6, the pitch rate channel is shown to be the leading indicator of system instability.

Another attempt to create a realistic change in aircraft characteristics was accomplished by failing the canard. This was a "free-floating" failure in which the canard was fixed at zero angle of attack in relation to the relative velocity of the aircraft. This failure resulted in coupling between flight path and pitch angle, which is to be expected since a failure of the canard removes all direct lift capabilities. See Figure F.7 for the full-state feedback aircraft response in the face of a full canard failure. In this plot it is shown that stability is maintained with the canard failure; however, the ability to achieve direct-lift is lost, therefore the pitch pointing maneuver can not be accomplished.

In order to address a common problem known as windup, which can affect the robustness of PI type controllers, an anti-windup compensator was added to the system. Readers who desire a derivation of the anti-windup compensator presented in this study are directed to Reference 16. Windup occurs when a PI controller encounters saturation limit. While the proportional channel will react immediately to compensate for the large errors which occur as a satiration is encountered, the integral channel will build up to a large control output. This buildup will continue until a sign change occurs in the input to the integral channel, even after the error in the system has been driven back to a small value. This "out of phase" compensation to saturations can induce system oscillation and instability. The anti-windup compensator reduces the effect of this phenomenon by placing limits on the value of the command inputs to the control surfaces of the aircraft. This precludes sending control signals which can cause these surfaces to be driven into saturation, thereby eliminating the possibility of inducing windup in the system. In some cases it was found that anti-windup compensation, when used without imposing saturation limits on the control surfaces in the truth model, could cause instability because it limited the control available to the system. However, for most cases the anti-windup compensator showed improved response. Comparing Figure F.8 to Figure 5.9 shows that the anti-windup compensator has reduced the oscillation in the pitch rate channel for the full-state feedback system with actuator limits. Figures F.9 and F.10 show the aircraft responses with a Kalman filter in the loop and with a Kalman filter with control based on $\hat{\underline{x}}(t_i^-)$, respectively. Notice that these responses are identical to Figures 5.10 and 5.11, indicating that for these controllers the saturation limits are never reached for a two-degree pitch pointing maneuver.

The sensors used on the STOL F-15 can be modeled using second-order dynamics (20). However, since these measurement dynamics were not included in the truth model, the measurement noise was increased as a "first cut" at evaluating the systems robustness in the face of unmodeled sensor dynamics and other uncertainties. Figure F.11 in Appendix F shows the effect of increasing the measurement noise to a level 40 times higher than that given in Equation (5-41) in the truth model without changing the measurement noise level used to design the Kalman filter. While the system maintained stability with this increased measurement noise, it can be seen from Figure F.11 that this condition causes increased workload on the control surfaces. An attempt was made to limit this overworking of the control surfaces by using anti-windup control; however, as evidenced in Figures F.12 and F.13, this did not remove the flutter in the canard and stabilator.

At this point the LTR tuning technique was introduced in an attempt to enhance system robustness with a Kalman filter in the loop (see Section 4.3). Despite exhaustive attempts to apply this technique, it was found that for all but extremely small values of q in Equation (4-4), the result was system instability. The system was finally stabilized with a q value of .00316. The analysis technique used in this thesis is severely limited in its regard to quantify the increase in system robustness using the LTR technique. First of all, just how much of the full state robustness was lost by introduction of a Kalman filter? If the original Kalman filter-based controller was already near the robustness levels of the full state system, the application of an LTR technique might be a waste of time. On the other hand, if serious robustness degradation has been introduced by the loss of full state availability, there is no way to measure how much has been gained in the way of system robustness with the LTR technique, i.e., when is the point of diminishing returns reached as q is increased further and further? Without the ability to apply structured singular value analysis, the LTR technique generated more questions than answers. However, attempts were made to analyze the impact of applying the LTR technique despite these shortcomings in the analytical approach available for this research.

In order to establish a standard by which to judge the LTR-tuned controller, white Gaussian noise was injected into the pitch rate and angle of attack channels in the truth model without changing the original controller design, in order to produce visible instability in the system. This process noise was made large enough ($Q = .08$) to have a marked impact on the system, as shown in Figure F.14. Once this "high-noise" baseline was established, the LTR tuning technique was applied to the control system in an effort to identify any improvement (i.e. less oscillatory behavior) in the face of high noise injection in the truth model. From the response of the system with LTR tuning, Figure F.15 in Appendix F, it can be seen that some reduction in oscillatory behavior of the system is achieved. However, this result is in no way intended to provide conclusive information of the application of the LTR technique; to the contrary, it would be less than technically correct to draw any conclusions from the application of LTR tuning on the basis of this admittedly limited analysis of the technique.

As a final attempt to try a tuning approach to increase the system robustness with the Kalman filter in the loop, an ad-hoc method was attempted based on the following:

$$\underline{Q}(q) = \underline{Q}_o + q^2\underline{C}^T\underline{C} \qquad (5\text{-}47)$$

where all the variables are the same as those presented in Section 4.3 for LTR tuning except for $\underline{C}$ which is the output matrix of Equation (2-8). Based on the observation that pitch rate is the most sensitive channel to parameter variation, the "$\underline{C}$-tuned" system was tested using variations in the $\underline{B}$ matrix of Equation (2-4). The stability derivatives M_{δ_C} and M_{δ_S} were independently varied, both with and without $\underline{C}$-tuning. The results shown in Table 5.1

TABLE 5.1

EFFECT OF C-TUNING FILTER

	With C Tuning		Without C Tuning	
	M_{δ_C}	M_{δ_S}	M_{δ_C}	M_{δ_S}
Percent Increase in Stability Derivative	16%	16%	7%	10%
Percent Decrease in Stability Derivative	100%	40%	100%	7%

indicate that a substantial increase in system robustness is achieved. Plotted responses of the C-tuned system are given in Appendix F, Figures F.16 through F.23. In Figure F.16 the full-state feedback system response with actuators is shown. It can be seen from this figure that the pitch rate channel exhibits no oscillation, as compared to Figures 5.9 and F.8, indicating increased system stability. However, the settling time is increased from 4.5 seconds to over 6 seconds. This type of tradeoff between robustness and response characteristics is indicative of LTR tuning scheme's characteristics (22). In Figure F.17 the aircraft response is shown with rate limits included in the truth model. Without C-tuning, imposing rate limits on the actuators drove the system into instability even in the full-state feedback system without actuators; however, when C-tuning was employed, the system was stable with not only rate limits but with actuators, position limits, a Kalman filter in the loop, and a time lag simultaneously introduced in the truth model as well (see Figure F.17). While stability is maintained, an effective steady-state error is incurred over the 6-second time period, as shown in the figure. Comparing Figure F.18 to Figure F.6 shows that, with C-tuning the system is stabilized in the face of a 200 percent additive increase in the F matrix (as compared to 100 percent for the non-C-tuned system of Figure F.5). Figure F.19 shows the C-tuned system response in the face of a 40-fold increase in measurement noise in the truth model. Comparing this response to that of Figures F.11, F.12, and F.13 shows that the control surface flutter is suppressed in the C-tuned system. The aircraft responses for variations in the B matrix stability derivatives, M_{δ_C} and M_{δ_S} (see Table 5.1) are shown in Figures F.19 through F.23. It should be noted that the C-tuning technique is not being presented as a theoretically correct method to regain system robustness for a CGT/PI/KF controller. Quite the contrary, this method is presented simply as an ad-hoc method which happens to have provided excellent results. To make any statements about this method for general application would require a complete mathematical proof validating that the return difference functions (at some specified loop breaking point of physical significance) for the controller both with and without the Kalman filter in the loop asymptotically become equal as q is increased in Equation (5-47). However, time did not allow a full investigation of the validity of this approach. Appendix G provides a preliminary explanation for the results obtained using C-tuning and some comments on the application of LTR tuning for loop breaking at points other than the point at which the input enters the system.

5.8 Summary

This section has presented the pitch pointing controller designed as a part of this thesis effort. The thought process used to arrive at the models and weighting matrices was introduced along with their final form. The robustness of the final controller was analyzed in the face of parameter variations, control surface saturations, canard failure, and highly noise-corrupted measurements. Also, LTR tuning was investigated as a means to recover the robustness of the system lost by introducing a Kalman filter into the controller to provide state estimates. Finally, an ad-hoc method for tuning the filter was introduced along with the encouraging results obtained using this method.

VI. Conclusions and Recommendations

6.1 Introduction

This section is intended to tie together the ideas and design approaches presented in the course of this thesis, at a more general and qualitative level than previously used. This is not a collection of unrelated ideas, but rather, an "overall view" of the research conducted in the course of preparing this document. Hopefully, the reader will be able to obtain an understanding of the nature of the design methodologies which have culminated in the controllers presented herein and the implications of these approaches for future flight control designs.

6.2 Conclusions

The most important result which has been demonstrated in research conducted in the writing of this thesis is the complete viability of the CGT/PI/KF design method for MIMO advanced fighter aircraft controller design. Not only was this approach shown to provide a systematic and intuitive design approach, but it was also shown to result in a controller which easily embedded pilot handling qualities and overall system robustness directly into the design process. Implicit model following was used as a method to limit the system bandwidth to guard against exciting ignored higher-order dynamics and inducing instability. This allowed us to design a system based on a relatively low-order model (four states) which displayed excellent robustness characteristics in the face of unmodeled actuator dynamics, control surface rate and position limits, actuator failures, and extreme variations in both the F and B matrices of the system. Explicit modeling was shown to be an effective means to embed handling qualities into the design process with a small increase in overall controller complexity. Finally, the nature of the iterative design process associated with the LQG-based PI structure allowed for useful insights to be developed readily within the design iterations themselves in order to achieve final specifications.

Another result which is worth including in this discussion is the questions which were raised pertaining to the LTR tuning of a CGT/PI/KF controller structure. While not rigorously proven by any means, the analysis of the LTR approach seemed to indicate that, in this particular case, the implementation of a CGT/PI/KF controller may not be consistent with the application of the original Doyle and Stein method of recovering loop transmission characteristics with the loop broken at the input to the system. A complete analysis of the frequency domain characteristics for the loop broken at various points in the system is appropriate to establish the applicability of the LTR method.

At the onset of this thesis effort, many goals were set to be accomplished. Of these initial goals, most were met or addressed at least to a limited degree. However, even in light of the work accomplished, the tremendous capability of the CGT/PI/KF design method was not fully exploited in the course of this research and bears further investigation.

6.3 Recommendations for Further Study

Although some valuable ground has been gained in the course of this thesis work, much more remains to be accomplished in the design and analysis of a CGT/PI/KF flight controller for the STOL F-15. The major areas which need to be addressed are presented in this section. This is not an "all inclusive" list of the possible ramifications of this thesis. However, it is a list of "possibilities" which warrant being addressed in the future.

1. One of the major drawbacks to the controller designs which were able to be accomplished in this research was the lack of integrated software. In order to use the CGT/PI/KF design method to its fullest extent, interactive software must be designed which allows the designer the flexibility not only to achieve the basic design, but also to test the design using actuator and sensor noise, time lags, Kalman filters in the loop, servo saturations and other designer-specified nonlinearities, and LTR tuning

simply by specifying a scalar q value and where the controller loop is to be broken. Also, as will be discussed below, the ability to accomplish structured singular value analysis and frequency domain analysis is a "must" for the proposed software as a means to establish the relative robustness of controllers in a direct manner. This type of integrated software package, were it available, would allow the designer the flexibility to design a system iteratively while testing it against increasingly accurate portrayals of the "real World" environment in which it is to function. Therefore, it is proposed that this type of CAD package either be designed or purchased for further studies in this area.

2. A problem which arose in the course of this thesis was the simultaneous control of both velocity and thrust vectoring nozzles. As shown in Appendix E, this results in a harshly nonlinear system which could not be adequately controlled using the linear constant-gain approaches adopted in this thesis and parallel efforts (1:20). Either a practical linearization should be applied to this system so that linear techniques may be implemented, or, possibly, more advanced techniques could be applied to this problem using parameter estimation or extended Kalman filtering techniques.

3. For further studies in the control of the STOL F-15, the controller designs should be tested in the face

of wind gusts modeled as outputs of a linear time invariant system driven by white Gaussian noise. Additional states to represent turbulence should be incorporated into the truth model and also, perhaps, in the filter/controller in an attempt to test controller robustness in turbulence.

4. A logical progression from the work accomplished in this thesis would be to extend the control designs to multiple longitudinal maneuvers and to investigate lateral mode controllers as well. A "single controller" approach could be analyzed in an attempt to eliminate the need to adjust controllers to a specific maneuver while at a fixed flight condition. This could produce a control system which is able to perform several maneuvers and relieves the pilot from "switching modes" in order to achieve different maneuvers.

5. An extremely useful tool to analyze the final controllers based on the above suggestions would be to derive dynamical equations of motion for the STOL F-15 which included asymmetrical surface failures. This would allow the impact of mode coupling on the system robustness to be evaluated in a maneuver such as a coordinated turn.

6. An interesting application of the CGT/PI/KF controller could be to attempt a more complicated "maneuver." For instance, instead of a coordinated turn or a pitch pointing, an entire evasive maneuver or automatic

mid-air collision avoidance scenario could be implemented using the aircraft's radar as a system measurement.

7. Last, but certainly not least, it is strongly suggested that for any further application of a CGT/PI/KF controller synthesis, the applicability of LTR tuning be rigorously verified and coupled with the use of structured singular value analysis and frequency domain analysis in order to gain more insight into the entire robustness issue.

Appendix A: STOLCAT Program Listing

Introduction

The following computer code listing is the STOLCAT program used to convert the aerodynamic data provided by McDonnell Air Co. into lateral and longitudinal state space three-degree-of-freedom equations of motion (see Section 3.2). It is written in FORTRAN V and is hosted on a Control Data Corporation Cyber mainframe computer. This program is completely interactive and provides prompts to specify the required input and the units for that input. It should be noted that no means is provided to store entered data in a permanent file structure. Therefore, care should be exercised to avoid making errors when entering data to avoid having to re-enter the program to enter all data from the beginning.

A.2 STOLCAT PROGRAM LISTING

```
      program stolcat
      real alpha,q,s,c,b,u,dtheta,w,bixx,biyy,bizz,
     1bixz,dalpha,dpr,vt,
     2cza,czq,czu,czd1,czd2,czd3,czd4,czd5,czd6,czd7,czd8,
     3cxa,cxq,cxu,cxd1,cxd2,cxd3,cxd4,cxd5,cxd6,cxd7,cxd8,
     4cma,cmq,cmu,cmd1,cmd2,cmd3,cmd4,cmd5,cmd6,cmd7,cmd8,
     5z1,za,zh,zq,zu,zd1,zd2,zd3,zd4,zd5,zd6,zd7,zd8,
     6xa,xh,xq,xu,xd1,xd2,xd3,xd4,xd5,xd6,xd7,xd8,
     7m1,ma,mh,mq,mu,md1,md2,md3,md4,md5,md6,md7,md8
      real cnb,cyb,clb,l,n
      dimension amat(4,4),bmat(4,8)
      dimension dirmat(5,5),dirbmat(5,9)
      character*3 key, key1, data1, data2, data3, run
      character*1 stab1,stab2
      data q /48.1/,s /608./, c /15.94/, b /42.7/, u /201./
      data dtheta /11.8030/, dalpha /11.8030/,w /33576.14/
      data bixx /23644./,biyy /181847./,bizz /199674./,bixz /-3086./
      data cza /-7.84976e-2/, cxa /1.5095276e-3/, cma /9.574118e-3/
      data czq /0./, cxq /0./, cmq /-.16951603/
      data czu /-1.06551597/, cxu /-6.1932e-3/, cmu /6.394289e-2/
      data czh /-1.676463e-4/, cxh /6.662777e-4/, cmh /1.76622e-4/
      data czd1 /-2.63634e-3/, cxd1 /-1.552420e-3/, cmd1 /5.57696e-3/
      data czd2 /-8.31511e-3/, cxd2 /-2.749671e-4/, cmd2 /-1.02066e-2/
      data czd3 /-5.59102e-3/, cxd3 /1.157373e-3/, cmd3 /8.52107e-4/
      data czd4 /-4.50843e-3/, cxd4 /9.4211093e-4/, cmd4 /-2.11118e-3/
      data czd5 /1.896349e-3/, cxd5 /-3.120989e-3/, cmd5 /2.55459e-3/
      data czd6 /-7.422954e-4/, cxd6 /-3.595656e-3/, cmd6 /-1.30123e-3/

      data czd7 /1.896349e-3/, cxd7 /-3.120989e-3/, cmd7 /2.55459e-3/
      data czd8 /-7.422954e-4/, cxd8 /-3.595658e-3/, cmd8 /-1.30123e-3/

      data clb /-2.973933e-3/, cnb /-5.5065055e-4/, cyb /-1.637941e-2/
      data clp /-5.740524e-3/, cnp /-2.3099719e-3/, cyp / 0.000000000/
      data clr / 3.902348e-3/, cnr /-9.6998151e-3/, cyr / 0.000000000/
      data cld1/1.0017e-4/, cnd1/-1.3256e-3/, cyd1/3.0606e-3/
      data cld2/-1.14939e-4/, cnd2/5.1323e-4/, cyd2/1.3139e-3/
      data cld3/8.5104e-4/, cnd3/4.4837e-4/, cyd3/-1.0622e-3/
      data cld4/7.5284e-4/, cnd4/7.6138e-5/, cyd4/-1.5235e-4/
      data cld5/6.9959e-4/, cnd5/0.00/, cyd5/0.00/
      data cld6/9.6816e-5/, cnd6/1.5934e-4/, cyd6/0.0/
      data cld7/-3.7897e-5/, cnd7/1.8357e-4/, cyd7/0.0/
      data cld8/-9.6816e-5/, cnd8/-1.5934e-4/, cyd8/0.0/
      data cld9/3.7837e-5/, cnd9/-1.8357e-4/, cyd9/0.0/
      dpr = 57.2957795
      write(*,5)
```

```
5     format(1x,'****************************************************')

      write(*,10)
10    format(1x,'*** stability derivative transformation program  ***')

      write(*,20)
20    format(1x,'****************************************************')

      write(*,100)
100   format(1x,'enter body axis (non-dimensionalized) coefficients ')
      write(*,101)
101   format(1x,'for transformation to dimensionalized body axis')
      write(*,102)
102   format(1x,'and to generate state and input matrices.')
      write(*,41)
41    format(1x,'note:  all coefficients are requested when computing')

103   continue
      write(*,30)
30    format(1x,'****************************************************')

      write(*,106)
106   format(1x,'to transform only longitudinal data - type long')
      write(*,107)
107   format(1x,'to transform only lateral-directional data - type
lat')
      write(*,108)
108   format(1x,'to transform both long and lat-dir data - type both')
      write(*,111)
111   format(1x,'Keyword =    ')
      read(*,109) Key
109   format(a3)
       if(Key .eq. 'lat') go to 104
       if(Key .eq. 'lon') go to 104
       if(Key .eq. 'bot') go to 104
       if(Key .eq. 'gam') go to 536
      go to 103
104   continue
      write(*,500)
500   format(1x,'****************************************************')

      write(*,510)
510   format(1x,'q  (dynamic pressure - lbs/ft**2) = ')
      read(*,*) q
      write(*,520)
520   format(1x,'s  (wing reference area - ft**2) = ')
      read(*,*) s
      write(*,530)
530   format(1x,'c  (wing mean aerodynamic cord - ft) = ')
      read(*,*) c
      write(*,540)
540   format(1x,'b  (wing span - ft) = ')
      read(*,*) b
      write(*,550)
550   format(1x,'vt (trim velocity - ft/sec) = ')
      read (*,*) u
      vt=u
      write(*,560)
560   format(1x,'theta (pitch angle - degs) = ')
      read(*,*) dtheta
      write(*,570)
570   format(1x,'w  (weight - lbs) = ')
      read(*,*) w
      write(*,575)
575   format(1x,'inertias must be input in body axis.')
      write(*,580)
580   format(1x,'ixx  (slug-ft**2) = ')
      read(*,*) bixx
      write(*,585)
585   format(1x,'iyy  (slug-ft**2) = ')
      read(*,*) biyy
      write(*,590)
590   format(1x,'izz  (slug-ft**2) = ')
      read(*,*) bizz
      write(*,595)
595   format(1x,'ixz  (slug-ft**2) = ')
      read(*,*) bixz
596   continue
      write(*,597)
597   format(1x,'*******************************************************')

      write(*,610)
610   format(16x,'aircraft parameters')
      write(*,615) q
615   format(1x,'q  (dynamic pressure - lbs/ft**2) = ',g13.6)
      write(*,620) s
620   format(1x,'s  (wing reference area - ft**2) = ',g13.6)
      write(*,625) c
625   format(1x,'c  (wing mean aerodynamic cord - ft) = ',g13.6)
      write(*,630) b
630   format(1x,'b  (wing span - ft) = ',g13.6)
      write(*,635) u
635   format(1x,'vt (trim velocity - ft/sec) = ',g13.6)
      write(*,640) dtheta
640   format(1x,'theta = ',g13.6)
      write(*,645) w
645   format(1x,'w  (weight - lbs) = ',g13.6)
      write(*,650) bixx
650   format(1x,'ixx  (slug-ft**2) = ',g13.6)
      write(*,655) biyy
```

```
655   format( 1x,'iyy  (slug-ft**2) = ',g13.6)
      write(*,660) bizz
660   format( 1x,'izz  (slug-ft**2) = ',g13.6)
      write(*,665) bixz
665   format( 1x,'ixz  (slug-ft**2) = ',g13.6)
      write(*,670)
670   format( 1x,'*****************************************************')

600   continue
      write(*,675)
675   format( 1x,'is the entered data correct ?  (yes/no) ')
      read(*,680) data3
680   format( a3)
      write(*,685)
685   format( 1x,'*****************************************************')

      if(data3 .eq. 'no ') go to 104
      if(data3 .eq. 'yes') go to 686
      go to 600
686   continue
      write(*,105)
105   format( 1x,'alpha (deg) = ')
      read(*,*) dalpha
      theta = dtheta/dpr
      alpha = dalpha/dpr
      if(key .eq. 'lat')go to 446
      if(key .eq. 'gam')go to 97
      write(*,110)
110   format ( 1x,'cza = ')
      read(*,*) cza
      write(*,120)
120   format( 1x,'cxa = ')
      read(*,*) cxa
      write(*,130)
130   format( 1x,'cma = ')
      read(*,*) cma
      write(*,140)
140   format( 1x,'czq = ')
      read(*,*) czq
      write(*,150)
150   format( 1x,'cxq = ')
      read(*,*) cxq
      write(*,160)
160   format( 1x,'cmq = ')
      read(*,*) cmq
      write(*,170)
170   format( 1x,'czu = ')
      read(*,*) czu
      write(*,130)
180   format( 1x,'cxu = ')
```

```
      read(*,*) cxu
      write(*,190)
190   format( 1x,'cmu = ')
      read(*,*) cmu
      write(*,191)
191   format( 1x,'czh = ')
      read(*,*) czh
      write(*,192)
192   format( 1x,'cxh = ')
      read(*,*) cxh
      write(*,193)
193   format( 1x,'cmh = ')
      read(*,*) cmh
      write(*,200)
200   format( 1x,'czd1 = ')
      read(*,*) czd1
      write(*,202)
202   format( 1x,'cxd1 = ')
      read(*,*) cxd1
      write(*,204)
204   format( 1x,'cmd1 = ')
      read(*,*) cmd1
      write(*,206)
206   format( 1x,'czd2 = ')
      read(*,*) czd2
      write(*,208)
208   format( 1x,'cxd2 = ')
      read(*,*) cxd2
      write(*,210)
210   format( 1x,'cmd2 = ')
      read(*,*) cmd2
      write(*,212)
212   format( 1x,'czd3 = ')
      read(*,*) czd3
      write(*,214)
214   format( 1x,'cxd3 = ')
      read(*,*) cxd3
      write(*,216)
216   format( 1x,'cmd3 = ')
      read(*,*) cmd3
      write(*,218)
218   format( 1x,'czd4 = ')
      read(*,*) czd4
      write(*,45)
45    format( 1x,'cxd4 = ')
      read(*,*) cxd4
      write(*,50)
50    format( 1x,'cmd4 = ')
      read(*,*) cmd4
      write(*,55)
```

```
55    format( 1x ,'czd5 = ')
      read(*,*) czd5
      write(*,60)
60    format( 1x ,'cxd5 = ')
      read(*,*) cxd5
      write(*,65)
65    format( 1x ,'cmd5 = ')
      read(*,*) cmd5
      write(*,70)
70    format( 1x ,'czd6 = ')
      read(*,*) czd6
      write(*,75)
75    format( 1x ,'cxd6 = ')
      read(*,*) cxd6
      write(*,80)
80    format( 1x ,'cmd6 = ')
      read(*,*) cmd6
      write(*,85)
85    format( 1x ,'czd7')
      read(*,*) czd7
      write(*,88)
88    format( 1x ,'cxd7')
      read(*,*) cxd7
      write(*,90)
90    format( 1x ,'cmd7 = ')
      read(*,*) cmd7
      write(*,92)
92    format( 1x ,'czd8 = ')
      read(*,*) czd8
      write(*,94)
94    format( 1x ,'cxd8 = ')
      read(*,*) cxd8
      write(*,96)
96    format( 1x ,'cmd8 = ')
      read(*,*) cmd8
97    continue
      write(*,225)
225   format( 1x ,'******************************************************')

      write(*,230) dalpha
230   format( 15x ,'alpha =',g13.6)
      write(*,345)
345   format(6x,'longitudinal non-dim body axis coefficients(1/deg)')

      cal = cos(alpha)
      sal = sin(alpha)
      cossq = cal**2
      sinsq = sal**2
      cossin = cal*sal
      cth = cos(theta)
      sth = sin(theta)

      write(*,360) cza,cma,cxa
360   format(3x,'cza = ',g13.6,8x,'cma = ',g13.6,5x,'cxa = ',g13.6)
      write(*,390) czq,cmq,cxq
390   format(3x,'czq = ',g13.6,8x,'cmq = ',g13.6,5x,'cxq = ',g13.6)
      write(*,400) czh,cmh,cxh
400   format(3x,'czh = ',g13.6,8x,'cmh = ',g13.6,5x,'cxh = ',g13.6)
      write(*,410) czu,cmu,cxu
410   format(3x,'czu = ',g13.6,8x,'cmu = ',g13.6,5x,'cxu = ',g13.6)
      write(*,370) czd1,cmd1,cxd1
370   format(2x,'czd1 = ',g13.6,7x,'cmd1 = ',g13.6,4x,'cxd1 = ',g13.6)
      write(*,380) czd2,cmd2,cxd2
380   format(2x,'czd2 = ',g13.6,7x,'cmd2 = ',g13.6,4x,'cxd2 = ',g13.6)
      write(*,381) czd3,cmd3,cxd3
381   format(2x,'czd3 = ',g13.6,7x,'cmd3 = ',g13.6,4x,'cxd3 = ',g13.6)
      write(*,382) czd4,cmd4,cxd4
382   forma.(2x,'czd4 = ',g13.6,7x,'cmd4 = ',g13.6,4x,'cxd4 = ',g13.6)
      write(*,383) czd5,cmd5,cxd5
383   format(2x,'czd5 = ',g13.6,7x,'cmd5 = ',g13.6,4x,'cxd5 = ',g13.6)
      write(*,384) czd6,cmd6,cxd6
384   format(2x,'czd6 = ',g13.6,7x,'cmd6 = ',g13.6,4x,'cxd6 = ',g13.6)
      write(*,385) czd7,cmd7,cxd7
385   format(2x,'czd7 = ',g13.6,7x,'cmd7 = ',g13.6,4x,'cxd7 = ',g13.6)
      write(*,386) czd8,cmd8,cxd8
386   format(2x,'czd8 = ',g13.6,7x,'cmd8 = ',g13.6,4x,'cxd8 = ',g13.6)
      write(*,310)
310   format(1x,'**************************************************')

315   continue
      write(*,320)
320   format(1x,'is the entered data correct ?   (yes/no)')
      read(*,330) data1
330   format(a3)
      if(data1 .eq. 'no ') go to 686
      if(data1 .eq. 'yes') go to 340
      go to 315
340   continue
       write(*,420)
420   format(1x,'**************************************************')

      z1 = (q*s*32.2)/w
      a = c/(2.0*u)
      theta = dtheta/dpr

      za = z1*cza*dpr
      zh = (z1/u)*czh
      zq = z1*a*czq*dpr
      zu = 2.*(z1/u)*czu
      zd1 = z1*czd1*dpr
      zd2 = z1*czd2*dpr
      zd3 = z1*czd3*dpr
      zd4 = z1*czd4*dpr
      zd5 = z1*czd5*dpr
      zd6 = z1*czd6*dpr
      zd7 = z1*czd7*dpr
      zd8 = z1*czd8*dpr

      xa  = z1*cxa*dpr
      xh  = (z1/u)*cxh
      xq  = z1*a*cxq*dpr
      xu  = 2.*(z1/u)*cxu
      xd1 = z1*cxd1*dpr
      xd2 = z1*cxd2*dpr
      xd3 = z1*cxd3*dpr
      xd4 = z1*cxd4*dpr
      xd5 = z1*cxd5*dpr
      xd6 = z1*cxd6*dpr
      xd7 = z1*cxd7*dpr
      xd8 = z1*cxd8*dpr

      m1  = (q*s*c)/biyy

      ma  = m1*cma*dpr
      mh  = (m1/u)*cmh
      mq  = m1*a*cmq*dpr
      mu  = 2.*(m1/u)*cmu
      md1 = m1*cmd1*dpr
      md2 = m1*cmd2*dpr
      md3 = m1*cmd3*dpr
      md4 = m1*cmd4*dpr
      md5 = m1*cmd5*dpr
      md6 = m1*cmd6*dpr
      md7 = m1*cmd7*dpr
      md8 = m1*cmd8*dpr

      write(*,700)
700   format (5x,'longitudinal axis dimensional derivatives')
      write(*,705)
705   format (15x,'body axis (1/rad)')
      write(*,710) za,ma,xa
710   format(4x,'za = ',g13.6,9x,'ma = ',g13.6,6x,'xa = ',g13.6)
      write(*,720) zq,mq,xq
720   format(4x,'zq = ',g13.6,9x,'mq = ',g13.6,6x,'xq = ',g13.6)
      write(*,730) zh,mh,xh
730   format(4x,'zh = ',g13.6,9x,'mh = ',g13.6,6x,'xh = ',g13.6)
      write(*,740) zu,mu,xu
740   format(4x,'zu = ',g13.6,9x,'mu = ',g13.6,6x,'xu = ',g13.6)
       write(*,750) zd1,md1,xd1
750   format(3x,'zd1 = ',g13.6,8x,'md1 = ',g13.6,5x,'xd1 = ',g13.6)
```

```
      write(*,760) zd2,md2,xd2
760   format(3x,'zd2 = ',g13.6,8x,'md2 = ',g13.6,5x,'xd2 = ',g13.6)
      write(*,770) zd3,md3,xd3
770   format(3x,'zd3 = ',g13.6,8x,'md3 = ',g13.6,5x,'xd3 = ',g13.6)
      write(*,780) zd4,md4,xd4
780   format(3x,'zd4 = ',g13.6,8x,'md4 = ',g13.6,5x,'xd4 = ',g13.6)
      write(*,790) zd5,md5,xd5
790   format(3x,'zd5 = ',g13.6,8x,'md5 = ',g13.6,5x,'xd5 = ',g13.6)
      write(*,800) zd6,md6,xd6
800   format(3x,'zd6 = ',g13.6,8x,'md6 = ',g13.6,5x,'xd6 = ',g13.6)
      write(*,810) zd7,md7,xd7
810   format(3x,'zd7 = ',g13.6,8x,'md7 = ',g13.6,5x,'xd7 = ',g13.6)
      write(*,820) zd3,md8,xd8
820   format(3x,'zd8 = ',g13.6,8x,'md8 = ',g13.6,5x,'xd8 = ',g13.6)
      write(*,830)
830   format(1x,'******************************************************')


     development of state matricies

     development of the plant matrix - a

      vt=u
      amat(1,1) = xu
      amat(1,2) = -vt*sal
      amat(1,3) = xa
      amat(1,4) = -32.2*cth
      amat(2,1) = mu
      amat(2,2) = mq
      amat(2,3) = ma
      amat(2,4) = 0.0
      amat(3,1) = zu/vt
      amat(3,2) = cal
      amat(3,3) = za/vt
      amat(3,4) = -32.2*sth/vt
      amat(4,1) = 0.0
      amat(4,2) = 1.0
      amat(4,3) = 0.0
      amat(4,4) = 0.0



      write(*,*)
      write(*,850)
850   format('1',5x,'longitudnal state matrix(body axis)')
      write(*,*)
      write(*,842)
842   format('0',2x,'for state1=u,state2=q,state3=alpha,state4=theta')
      write(*,*)
      do 855 i=1,4
      write(*,860) (amat(i,j),j=1,4)
855   continue
860   format('0',2x,4(g13.6,4x))
      write(*,*)

     now we'll get the input matrix - b

      bmat(1,1) = xd1
      bmat(1,2) = xd2
      bmat(1,3) = xd3
      bmat(1,4) = xd4
      bmat(1,5) = xd5
      bmat(1,6) = xd6
      bmat(1,7) = xd7
      bmat(1,8) = xd8
      bmat(2,1) = md1
      bmat(2,2) = md2
      bmat(2,3) = md3
      bmat(2,4) = md4
      bmat(2,5) = md5
      bmat(2,6) = md6
      bmat(2,7) = md7
      bmat(2,8) = md8
      bmat(3,1) = zd1/vt
      bmat(3,2) = zd2/vt
      bmat(3,3) = zd3/vt
      bmat(3,4) = zd4/vt
      bmat(3,5) = zd5/vt
      bmat(3,6) = zd6/vt
      bmat(3,7) = zd7/vt
      bmat(3,8) = zd8/vt
      do 865 i=1,8
      bmat(4,i) = 0.0
865   continue

     print out the long input matrix

      write(*,*)
      write(*,870)
870   format( '0',5x,'longitudnal input matrix')
      write(*,*)
      write(*,868)
868   format(2x,'for del1=canard,del2=stab,del3=tef,del4=dr aileron')
      write(*,869)
869   format(2x,'    del5=rt rv, del6=rb rv, del7=lt rv, del8=lb rv')
      write(*,*)
      write(*,*)
      write(*,871)
871   format('0',5x,'row1',11x,'row2',11x,'row3',11x,'row4')
      write(*,*)
     do 872 i=1,8
     write(*,880) (bmat(j,i),j=1,4)
872  continue
     write(*,*)
875  continue
     write(*,873)
873  format(1x,' do you want stab axis data for long?(y/n)')
     read(*,874) stabl
874  format(a1)
     if ( stabl .eq. 'y' ) go to 877
     if ( stabl .eq. 'n' ) go to 857
     go to 875
877  continue

************************************************************
*(((((((    *
*(     convert body axis data to stability axis(    *
*(       (for check with mcair data)((    *
*(((((((    *
*(((((((    *
************************************************************

      smu =( mu*cal + (ma/u)*sal*cal)
      smh =( ( smu / mu ) * mh )
      sma =( ma * cossq - mu * u * sal )
      smq = mq
      smd1 = md1
      smd2 = md2
      smd3 = md3
      smd4 = md4
      smd5 = md5
      smd6 = md6
      smd7 = md7
      smd8 = md8

      sxu=xu*cossq+(za/u)*sinsq*cal+((xa/u)*cal+zu)*sal*cal
      sxh = (sxu/xu)*xh
      sxa = xa*cal**3 -u*zu*sinsq - (u*xu - za*cal)*cal*sal
      sxq =( xq*cal + zq*sal )
      sxd1 = (xd1*cal + zd1*sal)
      sxd2 =( xd2*cal + zd2*sal)
      sxd3 = (xd3*cal + zd3*sal)
      sxd4 = (xd4*cal + zd4*sal)
      sxd5 = (xd5*cal + zd5*sal)
      sxd6 = (xd6*cal + zd6*sal)
      sxd7 = (xd7*cal + zd7*sal)
      sxd8 = (xd8*cal + zd8*sal)

      szu=zu*cossq-(xa/u)*sinsq*cal-(xu-(za/u)*cal)*sal*cal
      szh = (szu/zu) * zh
```

```
       sza=za*cal**3 + u*xu*sinsq - (u*zu + xa*cal)*cal*sal
       szq = (zq*cal - xq*sal)
       szd1 = (zd1*cal - xd1*sal)
       szd2 = (zd2*cal - xd2*sal)
       szd3 = (zd3*cal - xd3*sal)
       szd4 = (zd4*cal - xd4*sal)
       szd5 = (zd5*cal - xd5*sal)
       szd6 = (zd6*cal - xd6*sal)
       szd7 = (zd7*cal - xd7*sal)
       szd8 = (zd8*cal - xd8*sal)
       write(*,701)
701   format ('0',5x,'longitudinal axis dimensional derivatives')
      write(*,702)
702   format (15x,' stability axis (1/rad) ')
      write(*,711) sza,sma,sxa
711   format(4x,'za = ',g13.6,9x,'ma = ',g13.6,6x,'xa = ',g13.6)
      write(*,721) szq,smq,sxq
721   format(4x,'zq = ',g13.6,9x,'mq = ',g13.6,6x,'xq = ',g13.6)
      write(*,731) szh,smh,sxh
731   format(4x,'zh = ',g13.6,9x,'mh = ',g13.6,6x,'xh = ',g13.6)
      write(*,741) szu,smu,sxu
741   format(4x,'zu = ',g13.6,9x,'mu = ',g13.6,6x,'xu = ',g13.6)
      write(*,751) szd1,smd1,sxd1
751   format(3x,'zd1 = ',g13.6,8x,'md1 = ',g13.6,5x,'xd1 = ',g13.6)
      write(*,761) szd2,smd2,sxd2
761   format(3x,'zd2 = ',g13.6,8x,'md2 = ',g13.6,5x,'xd2 = ',g13.6)
      write(*,771) szd3,smd3,sxd3
771   format(3x,'zd3 = ',g13.6,8x,'md3 = ',g13.6,5x,'xd3 = ',g13.6)
      write(*,781) szd4,smd4,sxd4
781   format(3x,'zd4 = ',g13.6,8x,'md4 = ',g13.6,5x,'xd4 = ',g13.6)
      write(*,791) szd5,smd5,sxd5
791   format(3x,'zd5 = ',g13.6,8x,'md5 = ',g13.6,5x,'xd5 = ',g13.6)
      write(*,800) szd6,smd6,sxd6
801   format(3x,'zd6 = ',g13.6,8x,'md6 = ',g13.6,5x,'xd6 = ',g13.6)
      write(*,811) szd7,smd7,sxd7
811   format(3x,'zd7 = ',g13.6,8x,'md7 = ',g13.6,5x,'xd7 = ',g13.6)
      write(*,820) szd8,smd8,sxd8
821   format(3x,'zd8 = ',g13.6,8x,'md8 = ',g13.6,5x,'xd8 = ',g13.6)
      write(*,830)
880   format(2x,4(g13.6,2x))

       amat(1,1) = sxu
       amat(1,2) = 0.0
       amat(1,3) = sxa
       amat(1,4) = -32.2*cth
       amat(2,1) = smu
       amat(2,2) = smq
       amat(2,3) = sma
       amat(2,4) = 0.0
       amat(3,1) = szu/u
      amat(3,2) = 1.0
      amat(3,3) = sza/u
      amat(3,4) = -32.2*sth/u
      amat(4,1) = 0.0
      amat(4,2) = 1.0
      amat(4,3) = 0.0
      amat(4,4) = 0.0
    write(*,851)
851  format('0',5x,'longitudnal state matrix (stab axis)')
    write(*,*)
    write(*,842)
    write(*,*)
    do 856 i=1,4
    write(*,860) (amat(i,j),j=1,4)
856  continue
857  continue
     if (Key .eq. 'bot' ) go to 446
     if (Key .eq. 'gam' ) go to 465
421  continue
     write(*,430)
430  format(1x,'is another program run desired ?  (yes/no)')
     read(*,440) run
440  format(a3)
     write(*,445)
445  format(1x,'***************************************************')

     if(run .eq. 'no ') go to 450
     if(run .eq. 'yes') go to 103
     go to 421
446  continue

    this is where the lateral directional starts

     write(*,1110)
1110 format(1x,'clb (1/deg) = ')
     read(*,*) clb
     write(*,1120)
1120 format(1x,'cnb (1/deg) = ')
     read(*,*) cnb
     write(*,1130)
1130 format(1x,'cyb (1/deg) = ')
     read(*,*) cyb
     write(*,1140)
1140 format(1x,'clp (1/deg) = ')
     read(*,*) clp
     write(*,1150)
1150 format(1x,'cnp (1/deg) = ')
     read(*,*) cnp
     write(*,1160)
1160 format(1x,'cyp (1/deg) = ')
```

```
      read(*,*) cyp
      write(*,1170)
 1170 format(1x,'clr (1/deg) = ')
      read(*,*) clr
      write(*,1180)
 1180 format(1x,'cnr (1/deg) = ')
      read(*,*) cnr
      write(*,1190)
 1190 format(1x,'cyr (1/deg) = ')
      read(*,*) cyr
      write(*,1200)
 1200 format(1x,'cld1 (1/deg) = ')
      read(*,*) cld1
      write(*,1210)
 1210 format(1x,'cnd1 (1/deg) = ')
      read(*,*) cnd1
      write(*,1220)
 1220 format(1x,'cyd1 (1/deg) = ')
      read(*,*) cyd1
      write(*,1230)
 1230 format(1x,'cld2 (1/deg) = ')
      read(*,*) cld2
      write(*,1240)
 1240 format(1x,'cnd2 (1/deg) = ')
      read(*,*) cnd2
      write(*,1250)
 1250 format(1x,'cyd2 (1/deg) = ')
      read(*,*) cyd2
      write(*,1260)
 1260 format(1x,'cld3 (1/deg) = ')
      read(*,*) cld3
      write(*,1270)
 1270 format(1x,'cnd3 (1/deg) = ')
      read(*,*) cnd3
      write(*,1280)
 1280 format(1x,'cyd3 (1/deg) = ')
      read(*,*) cyd3
      write(*,1290)
 1290 format(1x,'cld4 (1/deg) = ')
      read(*,*) cld4
      write(*,1330)
 1330 format(1x,'cnd5 (1/deg) = ')
      read(*,*) cnd5
      write(*,1340)
 1340 format(1x,'cyd5 (1/deg) = ')
      read(*,*) cyd5
      write(*,1350)
 1350 format(1x,'cld6 (1/deg) = ')
      read(*,*) cld6
      write(*,1360)
 1360 format(1x,'cnd6 (1/deg) = ')
      read(*,*) cnd6
      write(*,1370)
 1370 format(1x,'cyd6 (1/deg) = ')
      read(*,*) cyd6
      write(*,1380)
 1380 format(1x,'cld7 (1/deg) = ')
      read(*,*) cld7
      write(*,1390)
 1390 format(1x,'cnd7 (1/deg) = ')
      read(*,*) cnd7
      write(*,1400)
 1400 format(1x,'cyd7 (1/deg) = ')
      read(*,*) cyd7
      write(*,1410)
 1410 format(1x,'cld8 (1/deg) = ')
      read(*,*) cld8
      write(*,1420)
 1420 format(1x,'cnd8 (1/deg) = ')
      read(*,*) cnd8
      write(*,1430)
 1430 format(1x,'cyd8 (1/deg) = ')
      read(*,*) cyd8
      write(*,1440)
 1440 format(1x,'cld9 (1/deg) = ')
      read(*,*) cld9
      write(*,1450)
 1450 format(1x,'cnd9 (1/deg) = ')
      read(*,*) cnd9
      write(*,1460)
 1460 format(1x,'cyd9 (1/deg) = ')
      read(*,*) cyd9
 1465 continue
      write(*,1470)
 1470 format('1',8x,'lat-dir body axis coefficients')
      if(key .eq. 'lon') go to 1490
      if(key .eq. 'bot') go to 1490
      write(*,1480) dalpha
 1480 format(15x,'alpha = ',g13.6)
 1490 continue
```

```
      write(*,1500) clb,cnb,cyb
1500  format(3x,'clb = ',g13.6,8x,'cnb = ',g13.6,5x,'cyb = ',g13.6)
      write(*,1510) clp,cnp,cyp
1510  format(3x,'clp = ',g13.6,8x,'cnp = ',g13.6,5x,'cyp = ',g13.6)
      write(*,1520) clr,cnr,cyr
1520  format(3x,'clr = ',g13.6,8x,'cnr = ',g13.6,5x,'cyr = ',g13.6)
      write(*,1530) cld1,cnd1,cyd1
1530  format(2x,'cld1 = ',g13.6,7x,'cnd1 = ',g13.6,4x,'cyd1 = ',g13.6)
      write(*,1540) cld2,cnd2,cyd2
1540  format(2x,'cld2 = ',g13.6,7x,'cnd2 = ',g13.6,4x,'cyd2 = ',g13.6)
      write(*,1550) cld3,cnd3,cyd3
1550  format(2x,'cld3 = ',g13.6,7x,'cnd3 = ',g13.6,4x,'cyd3 = ',g13.6)
      write(*,1560) cld4,cnd4,cyd4
1560  format(2x,'cld4 = ',g13.6,7x,'cnd4 = ',g13.6,4x,'cyd4 = ',g13.6)
      write(*,1570) cld5,cnd5,cyd5
1570  format(2x,'cld5 = ',g13.6,7x,'cnd5 = ',g13.6,4x,'cyd5 = ',g13.6)
      write(*,1580) cld6,cnd6,cyd6
1580  format(2x,'cld6 = ',g13.6,7x,'cnd6 = ',g13.6,4x,'cyd6 = ',g13.6)
      write(*,1590) cld7,cnd7,cyd7
1590  format(2x,'cld7 = ',g13.6,7x,'cnd7 = ',g13.6,4x,'cyd7 = ',g13.6)
      write(*,1600) cld8,cnd8,cyd8
1600  format(2x,'cld8 = ',g13.6,7x,'cnd8 = ',g13.6,4x,'cyd8 = ',g13.6)
      write(*,1610) cld9,cnd9,cyd9
1610  format(2x,'cld9 = ',g13.6,7x,'cnd9 = ',g13.6,4x,'cyd9 = ',g13.6)
      write(*,*)
      write(*,1620)
1620  format(1x,'***************************************************')

1625  continue
      write(*,1630)
1630  format(1x,'is the entered data correct ?   (yes/no)')
      read(*,1640) data2
1640  format(a3)
      if ( data2 .eq. 'no') go to 446
      if ( data2 .eq. 'yes' ) go to 1645
      go to 1625
1645  continue
      write(*,1646)
1646  format(1x,'do you want stab axis data for lat-dir? (y/n)')
      read(*,1647) stab2
1647  format(a1)
      if ( stab2 .eq. 'n') go to 1901
      if ( stab2 .eq. 'y') go to 1648
      go to 1645
1648  continue
      bsalph=-alpha
      csa=cos(bsalph)
      ssa=sin(bsalph)
      cs=csa*csa
      ss=ssa*ssa
```

```
     sclp=clp*cs + cnr*ss - (clr + cnp)*csa*ssa
     sclr=clr*cs - cnp*ss + (clp - cnr)*csa*ssa
     sclb=clb*csa - cnb*ssa
     scld1=cld1*csa - cnd1*ssa
     scld2=cld2*csa - cnd2*ssa
     scld3=cld3*csa - cnd3*ssa
     scld4=cld4*csa - cnd4*ssa
     scld5=cld5*csa - cnd5*ssa
     scld6=cld6*csa - cnd6*ssa
     scld7=cld7*csa - cnd7*ssa
     scld8=cld8*csa - cnd8*ssa
     scld9=cld9*csa - cnd9*ssa

     scnp=cnp*cs - clr*ss + (clp - cnr)*csa*ssa
     scnr=cnr*cs + clp*ss + (clr + cnp)*csa*ssa
     scnb=cnb*csa + clb*ssa
     scnd1=cnd1*csa + cld1*ssa
     scnd2=cnd2*csa + cld2*ssa
     scnd3=cnd3*csa + cld3*ssa
     scnd4=cnd4*csa + cld4*ssa
     scnd5=cnd5*csa + cld5*ssa
     scnd6=cnd6*csa + cld6*ssa
     scnd7=cnd7*csa + cld7*ssa
     scnd8=cnd8*csa + cld8*ssa
     scnd9=cnd9*csa + cld9*ssa

     scyp=cyp*csa - cyr*ssa
     scyr=cyr*csa + cyp*ssa
     scyb=cyb
     write(*,1471)
1471 format(8x,'lat-dir stab axis coefficients')
     write(*,1501) sclb,scnb,scyb
1501 format(3x,'clb = ',g13.6,8x,'cnb = ',g13.6,5x,'cyb = ',g13.6)
     write(*,1511) sclp,scnp,scyp
1511 format(3x,'clp = ',g13.6,8x,'cnp = ',g13.6,5x,'cyp = ',g13.6)
     write(*,1521) sclr,scnr,scyr
1521 format(3x,'clr = ',g13.6,8x,'cnr = ',g13.6,5x,'cyr = ',g13.6)
     write(*,1531) scld1,scnd1,cyd1
1531 format(2x,'cld1 = ',g13.6,7x,'cnd1 = ',g13.6,4x,'cyd1 = ',g13.6)
     write(*,1541) scld2,scnd2,cyd2
1541 format(2x,'cld2 = ',g13.6,7x,'cnd2 = ',g13.6,4x,'cyd2 = ',g13.6)
     write(*,1551) scld3,scnd3,cyd3
1551 format(2x,'cld3 = ',g13.6,7x,'cnd3 = ',g13.6,4x,'cyd3 = ',g13.6)
     write(*,1561) scld4,scnd4,cyd4
1561 format(2x,'cld4 = ',g13.6,7x,'cnd4 = ',g13.6,4x,'cyd4 = ',g13.6)
     write(*,1571) scld5,scnd5,cyd5
1571 format(2x,'cld5 = ',g13.6,7x,'cnd5 = ',g13.6,4x,'cyd5 = ',g13.6)
     write(*,1581) scld6,scnd6,cyd6
1581 format(2x,'cld6 = ',g13.6,7x,'cnd6 = ',g13.6,4x,'cyd6 = ',g13.6)
```

```
      write(*,1591) scld7,scnd7,cyd7
 1591 format(2x,'cld7 = ',g13.6,7x,'cnd7 = ',g13.6,4x,'cyd7 = ',g13.6)
      write(*,1601) scld8,scnd8,cyd8
 1601 format(2x,'cld8 = ',g13.6,7x,'cnd8 = ',g13.6,4x,'cyd8 = ',g13.6)
      write(*,1611) scld9,scnd9,cyd9
 1611 format(2x,'cld9 = ',g13.6,7x,'cnd9 = ',g13.6,4x,'cyd9 = ',g13.6)
      write(*,*)

      sixx=bixx*cossq + bizz*sinsq - bixz*sin(2*alpha)
      siyy=biyy
      sizz=bizz*cossq + bixx*sinsq + bixz*sin(2*alpha)
      sixz=bixz*cos(2*alpha) + .5*(bixx - bizz)*sin(2*alpha)

      sn  = dpr*(q*s*b)/sizz
      sl  = dpr*(q*s*b)/sixx
      sb  = b/(2.0*u)
      sy  = dpr*(q*s*32.2)/w
      snb  = sn*scnb
      snp  = sn*sb*scnp
      snr  = sn*sb*scnr
      snd1  = sn*scnd1
      snd2  = sn*scnd2
      snd3  = sn*scnd3
      snd4  = sn*scnd4
      snd5  = sn*scnd5
      snd6  = sn*scnd6
      snd7  = sn*scnd7
      snd8  = sn*scnd8
      snd9  = sn*scnd9

      slb  = sl*sclb
      slp  = sl*sb*sclp
      slr  = sl*sb*sclr
      sld1  = sl*scld1
      sld2  = sl*scld2
      sld3  = sl*scld3
      sld4  = sl*scld4
      sld5  = sl*scld5
      sld6  = sl*scld6
      sld7  = sl*scld7
      sld8  = sl*scld8
      sld9  = sl*scld9

      syb  = sy*scyb
      syr  = sy*sb*scyr
      syp  = sy*sb*scyp
      syd1  = sy* cyd1
      syd2  = sy* cyd2
      syd3  = sy* cyd3
      syd4  = sy* cyd4
```

```
     syd5 = sy* cyd5
     syd6 = sy* cyd6
     syd7 = sy* cyd7
     syd8 = sy* cyd8
     syd9 = sy* cyd9

     write(*,1661)
1661 format(5x,'lat-dir stab axis dimensional derivatives(1/rad)')
     write(*,1671) snb,slb,syb
1671 format(4x,'nb = ',g13.6,9x,'lb = ',g13.6,5x,'yb = ',g13.6)
     write(*,1681) snp,slp,syp
1681 format(4x,'np = ',g13.6,9x,'lp = ',g13.6,5x,'yp = ',g13.6)
     write(*,1691) snr,slr,syr
1691 format(4x,'nr = ',g13.6,9x,'lr = ',g13.6,5x,'yr = ',g13.6)
     write(*,1701) snd1,sld1,syd1
1701 format(3x,'nd1 = ',g13.6,8x,'ld1 = ',g13.6,4x,'yd1 = ',g13.6)
     write(*,1711) snd2,sld2,syd2
1711 format(3x,'nd2 = ',g13.6,8x,'ld2 = ',g13.6,4x,'yd2 = ',g13.6)
     write(*,1721) snd3,sld3,syd3
1721 format(3x,'nd3 = ',g13.6,8x,'ld3 = ',g13.6,4x,'yd3 = ',g13.6)
     write(*,1731) snd4,sld4,syd4
1731 format(3x,'nd4 = ',g13.6,8x,'ld4 = ',g13.6,4x,'yd4 = ',g13.6)
     write(*,1741) snd5,sld5,syd5
1741 format(3x,'nd5 = ',g13.6,8x,'ld5 = ',g13.6,4x,'yd5 = ',g13.6)
     write(*,1751) snd6,sld6,syd6
1751 format(3x,'nd6 = ',g13.6,8x,'ld6 = ',g13.6,4x,'yd6 = ',g13.6)
     write(*,1761) snd7,sld7,syd7
1761 format(3x,'nd7 = ',g13.6,8x,'ld7 = ',g13.6,4x,'yd7 = ',g13.6)
     write(*,1771) snd8,sld8,syd8
1771 format(3x,'nd8 = ',g13.6,8x,'ld8 = ',g13.6,4x,'yd8 = ',g13.6)
     write(*,1781) snd9,sld9,syd9
1781 format(3x,'nd9 = ',g13.6,8x,'ld9 = ',g13.6,4x,'yd9 = ',g13.6)

     write(*,1650)
1650 format(1x,'****************************************************')

1801 continue
     n = dpr*(q*s*b)/bizz
     l = dpr*(q*s*b)/bixx
     bb = b/(2.0*u)
     y = dpr*(q*s*32.2)/w
     bnb = n*cnb
     bnp = n*bb*cnp
     bnr = n*bb*cnr
     bnd1 = n*cnd1
     bnd2 = n*cnd2
     bnd3 = n*cnd3
     bnd4 = n*cnd4
     bnd5 = n*cnd5
     bnd6 = n*cnd6
```

```
      bnd7 = n*cnd7
      bnd8 = n*cnd8
      bnd9 = n*cnd9

      blb  = l*clb
      blp  = l*bb*clp
      blr  = l*bb*clr
      bld1 = l*cld1
      bld2 = l*cld2
      bld3 = l*cld3
      bld4 = l*cld4
      bld5 = l*cld5
      bld6 = l*cld6
      bld7 = l*cld7
      bld8 = l*cld8
      bld9 = l*cld9

      byb  = y*cyb
      byr  = y*bb*cyr
      byp  = y*bb*cyp
      byd1 = y*cyd1
      byd2 = y*cyd2
      byd3 = y*cyd3
      byd4 = y*cyd4
      byd5 = y*cyd5
      byd6 = y*cyd6
      byd7 = y*cyd7
      byd8 = y*cyd8
      byd9 = y*cyd9

      write(*,1660)
 1660 format(5x,'lat-dir body axis dimensional derivatives(1/rad)')
      write(*,1670) bnb,blb,byb
 1670 format(4x,'nb = ',g13.6,9x,'lb = ',g13.6,5x,'yb = ',g13.6)
      write(*,1680) bnp,blp,byp
 1680 format(4x,'np = ',g13.6,9x,'lp = ',g13.6,5x,'yp = ',g13.6)
      write(*,1690) bnr,blr,byr
 1690 format(4x,'nr = ',g13.6,9x,'lr = ',g13.6,5x,'yr = ',g13.6)
      write(*,1700) bnd1,bld1,byd1
 1700 format(3x,'nd1 = ',g13.6,8x,'ld1 = ',g13.6,4x,'yd1 = ',g13.6)
      write(*,1710) bnd2,bld2,byd2
 1710 format(3x,'nd2 = ',g13.6,8x,'ld2 = ',g13.6,4x,'yd2 = ',g13.6)
      write(*,1720) bnd3,bld3,byd3
 1720 format(3x,'nd3 = ',g13.6,8x,'ld3 = ',g13.6,4x,'yd3 = ',g13.6)
      write(*,1730) bnd4,bld4,byd4
 1730 format(3x,'nd4 = ',g13.6,8x,'ld4 = ',g13.6,4x,'yd4 = ',g13.6)
      write(*,1740) bnd5,bld5,byd5
 1740 format(3x,'nd5 = ',g13.6,8x,'ld5 = ',g13.6,4x,'yd5 = ',g13.6)
      write(*,1750) bnd6,bld6,byd6
 1750 format(3x,'nd5 = ',g13.6,8x,'ld6 = ',g13.6,4x,'yd6 = ',g13.6)
```

```
      write(*,1760) bnd7,bld7,byd7
1760  format(3x,'nd7 = ',g13.6,8x,'ld7 = ',g13.6,4x,'yd7 = ',g13.6)
      write(*,1770) bnd8,bld8,byd8
1770  format(3x,'nd8 = ',g13.6,8x,'ld8 = ',g13.6,4x,'yd8 = ',g13.6)
      write(*,1780) bnd9,bld9,byd9
1780  format(3x,'nd9 = ',g13.6,8x,'ld9 = ',g13.6,4x,'yd9 = ',g13.6)
      write(*,1790)
1790  format( 1x,'***************************************************')

      write(*,1800)
1800  format( 1x,'***************************************************')



     conversion of data into state space form


      d = 1.0 - ((bixz*bixz)/(bixx*bizz))
      r1 = bixz/bizz
      r2 = bixz/bixx

      pbnb = (bnb + r1*blb)/d
      pbnp = (bnp + r1*blp)/d
      pbnr = (bnr + r1*blr)/d
      pbnd1 = (bnd1 + r1*bld1)/d
      pbnd2 = (bnd2 + r1*bld2)/d
      pbnd3 = (bnd3 + r1*bld3)/d
      pbnd4 = (bnd4 + r1*bld4)/d
      pbnd5 = (bnd5 + r1*bld5)/d
      pbnd6 = (bnd6 + r1*bld6)/d
      pbnd7 = (bnd7 + r1*bld7)/d
      pbnd8 = (bnd8 + r1*bld8)/d
      pbnd9 = (bnd9 + r1*bld9)/d

      pblb = (blb + r2*bnb)/d
      pblp = (blp + r2*bnp)/d
      pblr = (blr + r2*bnr)/d
      pbld1 = (bld1 + r2*bnd1)/d
      pbld2 = (bld2 + r2*bnd2)/d
      pbld3 = (bld3 + r2*bnd3)/d
      pbld4 = (bld4 + r2*bnd4)/d
      pbld5 = (bld5 + r2*bnd5)/d
      pbld6 = (bld6 + r2*bnd6)/d
      pbld7 = (bld7 + r2*bnd7)/d
      pbld8 = (bld8 + r2*bnd8)/d
      pbld9 = (bld9 + r2*bnd9)/d

      pbyb = byb/u
      pbyp = sal
      pbyr = -cal
      pbyphi = 32.2*cth/u
      pbyd1 = byd1/u
      pbyd2 = byd2/u
      pbyd3 = byd3/u
      pbyd4 = byd4/u
      pbyd5 = byd5/u
      pbyd6 = byd6/u
      pbyd7 = byd7/u
      pbyd8 = byd8/u
      pbyd9 = byd9/u

( lateral directional state matrix


      do 1805 i=1,5
       do 1806 j=1,5
1806   dirmat(i,j)=0.0
1805  continue
      dirmat(1,3)=1.0
      dirmat(2,1)=pbyphi
      dirmat(2,2)=pbyb
      dirmat(2,3)=pbyp
      dirmat(2,4)=pbyr
      dirmat(2,5)=32.2*sth/u
      dirmat(3,2)=pblb
      dirmat(3,3)=pblp
      dirmat(3,4)=pblr
      dirmat(4,2)=pbnb
      dirmat(4,3)=pbnp
      dirmat(4,4)=pbnr
      dirmat(5,4)=1.0


       output the state matrix


      write(*,830)
      write(*,1810)
1810  format('1',2x,'lateral directional state matrix')
      write(*,1820)
1820  format('0',5x,'states = phi,beta,p,r,psi')
      write(*,*)
      write(*,1825) (dirmat(1,i),i=1,5)
      write(*,1825) (dirmat(2,i),i=1,5)
      write(*,1825) (dirmat(3,i),i=1,5)
      write(*,1825) (dirmat(4,i),i=1,5)
      write(*,1825) (dirmat(5,i),i=1,5)
      write(*,*)
1825  format('0',2x,5(g11.4,4x) )
```

```
       lateral directional input matrix

      do 1830 i=1,9
      dirbmat(1,i)=0.0
      dirbmat(5,i)=0.0
 1830 continue
      dirbmat(2,1)=pbyd1
      dirbmat(2,2)=pbyd2
      dirbmat(2,3)=pbyd3
      dirbmat(2,4)=pbyd4
      dirbmat(2,5)=pbyd5
      dirbmat(2,6)=pbyd6
      dirbmat(2,7)=pbyd7
      dirbmat(2,8)=pbyd8
      dirbmat(2,9)=pbyd9
      dirbmat(3,1)=pbnd1
      dirbmat(3,2)=pbnd2
      dirbmat(3,3)=pbnd3
      dirbmat(3,4)=pbnd4
      dirbmat(3,5)=pbnd5
      dirbmat(3,6)=pbnd6
      dirbmat(3,7)=pbnd7
      dirbmat(3,8)=pbnd8
      dirbmat(3,9)=pbnd9
      dirbmat(4,1)=pbld1
      dirbmat(4,2)=pbld2
      dirbmat(4,3)=pbld3
      dirbmat(4,4)=pbld4
      dirbmat(4,5)=pbld5
      dirbmat(4,6)=pbld6
      dirbmat(4,7)=pbld7
      dirbmat(4,8)=pbld8
      dirbmat(4,9)=pbld9

      print out the input matrix

      write(*,1850)
 1850 format('0',2x,'lateral directional input matrix')
      write(*,1860)
 1860 format('0',4x,'for inputs: del1=rudder,del2=diff can')
      write(*,1870)
 1870 format( 6x,'del3=diff stab, del4=diff ail, del5=diff tef')
      write(*,1880)
 1880 format(6x,'del6 to 9 are reverser vane ports')
      write(*,1890)
 1890 format('0',5x,'row1',11x,'row2',11x,'row3',11x,'row4',11x,'row5')

      do 1900 i=1,9
      write(*,1825) (dirbmat(j,i),j=1,5)
```

```
1900 continue
     go to 421
450  continue
     end
```

APPENDIX B STOL F-15 AERODYNAMIC DATA

B.1 LONGITUDINAL AND LATERAL DATA: MACH 0.3 AT 20,000 FEET

```
 *****************************************************
                 aircraft parameters
 q  (dynamic pressure - lbs/ft**2) =   61.3430
 s  (wing reference area - ft**2) =   608.000
 c  (wing mean aerodynamic cord - ft) =   15.9400
 b  (wing span - ft) =   42.7000
 vt (trim velocity - ft/sec) =   311.178
 theta =   12.2435
 w  (weight - lbs) =   377394.
 ixx  (slug-ft**2) =   25938.0
 iyy  (slug-ft**2) =   185287.
 izz  (slug-ft**2) =   206359.
 ixz  (slug-ft**2) =  -2543.00
 *****************************************************
               alpha =  12.2435
      longitudinal non-dim body axis coefficients(1/deg)
  cza =  -.794180e-01         cma =   .426440e-02     cxa =
-.197155e-02
  czq =  0.                   cmq =  -.156080         cxq =  0.
  czh =   .636483e-04         cmh =  -.278182e-04     cxh =
.959477e-03
  czu =  -.128400e-01         cmu =  -.561187e-02     cxu =  -.193560
 czd1 =  -.308825e-02        cmd1 =   .649500e-02    cxd1 =
-.831577e-03
 czd2 =  -.109619e-01        cmd2 =  -.114632e-01    cxd2 =
-.146560e-02
 czd3 =  -.342080e-02        cmd3 =  -.365130e-02    cxd3 =
.592350e-04
 czd4 =  -.342080e-02        cmd4 =  -.365180e-02    cxd4 =
.592340e-04
 czd5 =  0.                  cmd5 =  0.              cxd5 =  0.
 czd6 =  0.                  cmd6 =  0.              cxd6 =  0.
 czd7 =  0.                  cmd7 =  0.              cxd7 =  0.
 czd8 =  0.                  cmd8 =  0.              cxd8 =  0.
*****************************************************
     longitudinal axis dimensional derivatives
               body axis (1/rad)
   za =  -14.4801             ma =   .783958          xa =  -.359468
   zq =  0.                   mq =  -.734905          xq =  0.
   zh =   .650889e-06         mh =  -.286835e-06      xh =
.981134e-05
   zu =  -.262612e-03         mu =  -.115729e-03      xu =
-.395882e-02
  zd1 =  -.563073            md1 =   1.19403         xd1 =  -.151619
  zd2 =  -1.99865               md2 =  -2.10737               xd2 =  -.267219
  zd3 =  -.623705               md3 =  -.671339               xd3 =
.108002e-01
  zd4 =  -.623705               md4 =  -.671339               xd4 =
.108000e-01
  zd5 =  0.                     md5 =  0.                     xd5 =  0.
  zd6 =  0.                     md6 =  0.                     xd6 =  0.
  zd7 =  0.                     md7 =  0.                     xd7 =  0.
  zd8 =  0.                     md8 =  0.                     xd8 =  0.
****************************************************

     longitudinal state matrix(body axis)

  for state1=u,state2=q,state3=alpha,state4=theta

   -.395882e-02     -65.9905          -.353468          -31.4676
   -.115729e-03     -.734905           .783958          0.
   -.843930e-06      .977255          -.465331e-01      -.219442e-01
   0.                1.00000          0.                0.


             longitudinal input matrix

 for del1=canard,del2=stab,del3=tef,del4=dr aileron
     del5=rt rv, del6=rb rv, del7=lt rv, del8=lb rv


     row1            row2             row3             row4

  -.151619         1.19403        -.180949e-02   0.
  -.267219        -2.10737        -.642287e-02   0.
   .108002e-01    -.671339        -.200434e-02   0.
   .108000e-01    -.671339        -.200434e-02   0.
  0.              0.              0.             0.
  0.              0.              0.             0.
  0.              0.              0.             0.
  0.              0.              0.             0.

     longitudinal axis dimensional derivatives
               stability axis (1/rad)
   za =  -13.4800               ma =   .756338              xa =  -3.00916
   zq =  0.                     mq =  -.734905              xq =  0.
   zh =   .218207e-04           mh =   .101375e-05          xh =
.151542e-04
   zu =  -.880393e-02           mu =   .403016e-03          xu =
-.611427e-02
  zd1 =  -.518112               md1 =   1.19403             xd1 =  -.267580
  zd2 =  -1.89653               md2 =  -2.10737             xd2 =  -.684989
  zd3 =  -.611810               md3 =  -.671339             xd3 =  -.121713
  zd4 =  -.611810               md4 =  -.671339             xd4 =  -.121713
  zd5 =  0.                   md5 =  0.                 xd5 =  0.
  zd6 =  0.                   md6 =  0.                 xd6 =  0.
  zd7 =  0.                   md7 =  0.                 xd7 =  0.
  zd8 =  0.                   md8 =  0.                 xd8 =  0.
************************************************
       lat-dir body axis coefficients
  clb =  -.254330e-02         cnb =  -.483366e-03       cyb =
-.167190e-01
  clp =  -.534782e-02         cnp =  -.232644e-02       cyp =  0.
  clr =   .382830e-02         cnr =  -.893797e-02       cyr =  0.
 cld1 =   .742100e-04        cnd1 =  -.144010e-02      cyd1 =
.317730e-02
 cld2 =  -.185490e-04        cnd2 =   .565260e-03      cyd2 =
.131870e-02
 cld3 =   .885870e-03        cnd3 =   .368420e-03      cyd3 =
-.101060e-02
 cld4 =   .683140e-03        cnd4 =   .638620e-04      cyd4 =
-.845490e-04
 cld5 =   .711050e-03        cnd5 =  0.                cyd5 =  0.
 cld6 =   .184410e-03        cnd6 =  0.                cyd6 =  0.
 cld7 =  -.184410e-03        cnd7 =  0.                cyd7 =  0.
 cld8 =  0.                  cnd8 =  0.                cyd8 =  0.
 cld9 =  0.                  cnd9 =  0.                cyd9 =  0.

************************************************
      lat-dir stab axis coefficients
  clb =  -.258796e-02         cnb =   .663776e-04       cyb =
-.167190e-01
  clp =  -.519803e-02         cnp =  -.313802e-02       cyp =  0.
  clr =   .301672e-02         cnr =  -.908776e-02       cyr =  0.
 cld1 =  -.232875e-03        cnd1 =  -.142308e-02      cyd1 =
.317730e-02
 cld2 =   .101746e-03        cnd2 =   .556337e-03      cyd2 =
.131970e-02
 cld3 =   .943851e-03        cnd3 =   .172177e-03      cyd3 =
-.101060e-02
 cld4 =   .681145e-03        cnd4 =  -.824618e-04      cyd4 =
-.845490e-04
 cld5 =   .694877e-03        cnd5 =  -.150790e-03      cyd5 =  0.
 cld6 =   .180216e-03        cnd6 =  -.391072e-04      cyd6 =  0.
 cld7 =  -.180216e-03        cnd7 =   .391072e-04      cyd7 =  0.
 cld8 =  0.                  cnd8 =  0.                cyd8 =  0.
 cld9 =  0.                  cnd9 =  0.                cyd9 =  0.

    lat-dir stab axis dimensional derivatives(1/rad)
   nb =   .309923e-01          lb =  -6.72659           yb =  -3.04833
   np =  -.396268e-01          lp =  -.926963           yp =  0.
   nr =  -.283521              lr =   .537975           yr =  0.
  nd1 =  -.658510             ld1 =  -.605287          yd1 =   .579309
  nd2 =   .257436             ld2 =   .264456          yd2 =   .240435
  nd3 =   .796721e-01        ld3 =   2.45325         yd3 =  -.184260
  nd4 =  -.381580e-01        ld4 =   1.77042         yd4 =
-.154156e-01
  nd5 =  -.697758e-01        ld5 =   1.80612         yd5 =  0.
  nd6 =  -.180963e-01        ld6 =   .468414         yd6 =  0.
  nd7 =   .180963e-01        ld7 =  -.468414         yd7 =  0.
  nd8 =  0.                  ld8 =  0.               yd8 =  0.
  nd9 =  0.                  ld9 =  0.               yd9 =  0.
***********************************************************
    lat-dir body axis dimensional derivatives( 1/rad)
   nb =  -.213733             lb =  -8.94706          yb =  -3.04833
   np =  -.705792e-01         lp =  -1.29077          yp =  0.
   nr =  -.271159             lr =   .924012          yr =  0.
  nd1 =  -.636778            ld1 =   .261063         yd1 =   .579309
  nd2 =   .249945            ld2 =  -.652534e-01     yd2 =   .240435
  nd3 =   .162907            ld3 =   3.11640         yd3 =  -.184260
  nd4 =   .282383e-01        ld4 =   2.40321         yd4 =
-.154156e-01
  nd5 =  0.                  ld5 =   2.50140         yd5 =  0.
  nd6 =  0.                  ld6 =   .648735         yd6 =  0.
  nd7 =  0.                  ld7 =  -.648735         yd7 =  0.
  nd8 =  0.                  ld8 =  0.               yd8 =  0.
  nd9 =  0.                  ld9 =  0.               yd9 =  0.
***********************************************************
  lateral directional state matrix
     states = phi,beta,p,r,psi

   0.               0.                 1.000           0.                0.
    .1011           -.9796e-02         .2121           -.9773
.2194e-01
   0.               -8.937            -1.285            .9517            0.
   0.               -.1036            -.5474e-01       -.2829            0.
   0.               0.                0.                1.000            0.

  lateral directional input matrix
    for inputs: del1=rudder,del2=diff can
     del3=diff stab, del4=diff ail, del5=diff tef
     del6 to 9 are reverser vane ports
     row1               row2               row3              row4               row5
   0.                 .1862e-02        -.6408             .3239            0.
   0.                 .7727e-03         .2511            -.8987e-01        0.
   0.               -.5921e-03          .1247             3.104            0.
   0.               -.4954e-04        -.1373e-02          2.403            0.
   0.               0.                -.3086e-01          2.504            0.
   0.               0.                -.8004e-02          .6495            0.
   0.               0.                 .8004e-02         -.6495            0.
   0.               0.                0.                0.                 0.
   0.               0.                0.                0.                 0.
```

B.2 LONGITUDINAL AND LATERAL DATA: MACH 0.9 AT 20,000 FEET

```
*****************************************************
                aircraft parameters
 q  (dynamic pressure - lbs/ft**2) =   551.440
 s  (wing reference area - ft**2) =   608.000
 c  (wing mean aerodynamic cord - ft) =   15.9399
 b  (wing span - ft) =   42.7000
 vt (trim velocity - ft/sec) =   933.530
 theta =   1.22896
 w  (weight - lbs) =   37794.2
 ixx  (slug-ft**2) =   25938.0
 iyy  (slug-ft**2) =   185237.
 izz  (slug-ft**2) =   206359.
 ixz  (slug-ft**2) =  -2543.00
 ****************************************************
                alpha =  1.22897
      longitudinal non-dim body axis coefficients( 1/deg)
   cza =  -.852800e-01          cma =   .654210e-02     cxa =
.170230e-02
   czq =  0.                    cmq =  -.141700         cxq =  0.
   czh =   .787600e-05          cmh =  -.566610e-04     cxh =
.434250e-03
   czu =  -.529610e-03          cmu =  -.381020e-02     cxu =
-.292000e-01
  czd1 =  -.258140e-02         cmd1 =   .709040e-02    cxd1 =
.872020e-04
  czd2 =  -.106980e-01         cmd2 =  -.119130e-01    cxd2 =
-.608630e-03
  czd3 =  -.643700e-03         cmd3 =  -.102230e-02    cxd3 =
-.103160e-04
  czd4 =  -.643700e-03         cmd4 =  -.102230e-02    cxd4 =
-.103160e-04
  czd5 =  0.                   cmd5 =  0.              cxd5 =  0.
  czd6 =  0.                   cmd6 =  0.              cxd6 =  0.
  czd7 =  0.                   cmd7 =  0.              cxd7 =  0.
  czd8 =  0.                   cmd8 =  0.              cxd8 =  0.
 ****************************************************
     longitudinal axis dimensional derivatives
                body axis ( 1/rad)
    za =  -1395.73              ma =   10.8143          xa =   27.8638
    zq =  0.                    mq =  -1.99977          xq =  0.
    zh =   .240996e-05          mh =  -.175112e-05      xh =
.132875e-03
    zu =  -.324109e-03          mu =  -.235510e-03      xu =
-.178697e-01
   zd1 =  -42.2484             md1 =   11.7207         xd1 =   1.42719
   zd2 =  -175.088             md2 =  -19.6926         xd2 =  -9.96113
   zd3 =  -10.5351             md3 =  -1.68990         xd3 =  -.168837
```

```
    zd4 =  -10.5351            md4 =  -1.68990          xd4 =  -.168837
    zd5 =  0.                  md5 =  0.                xd5 =  0.
    zd6 =  0.                  md6 =  0.                xd6 =  0.
    zd7 =  0.                  md7 =  0.                xd7 =  0.
    zd8 =  0.                  md8 =  0.                xd8 =  0.
**************************************************

     longitudinal state matrix(body axis)
  for state1=u,state2=q,state3=alpha,state4=theta

   -.178697e-01     -20.0223          27.8688          -32.1926
   -.235510e-03     -1.99377          10.8143          0.
   -.347186e-06      .999770         -1.49511          -.739792e-03
   0.                1.00000         0.                0.


     longitudinal  input matrix

  for del1=canard,del2=stab,del3=tef,del4=dr aileron
      del5=rt rv, del6=rb rv, del7=lt rv, del8=lb rv


     row1            row2            row3            row4

    1.42719         11.7207         -.452566e-01   0.
   -9.96113        -19.6926         -.187555       0.
   -.168837        -1.68990         -.112852e-01   0.
   -.168837        -1.68990         -.112852e-01   0.
   0.              0.               0.             0.
   0.              0.               0.             0.
   0.              0.               0.             0.
   0.              0.               0.             0.

     longitudinal axis dimensional derivatives
                stability axis (1/rad)
     za =  -1335.37            ma =   10.8141          xa =  -1.71436
     zq =  0.                  mq =  -1.99977          xq =  0.
     zh =   .237392e-03        mh =   .962678e-07      xh =
 .133220e-03
     zu =  -.320068e-01        mu =   .129472e-04      xu =
 -.179160e-01
    zd1 =  -42.2693            md1 =   11.7207         xd1 =   .520720
    zd2 =  -174.835            md2 =  -19.6926         xd2 =  -13.7141
    zd3 =  -10.5291            md3 =  -1.68990         xd3 =  -.394754
    zd4 =  -10.5291            md4 =  -1.68990         xd4 =  -.394754
    zd5 =  0.                  md5 =  0.               xd5 =  0.
    zd6 =  0.                  md6 =  0.               xd6 =  0.
    zd7 =  0.                  md7 =  0.               xd7 =  0.
    zd8 =  0.                  md8 =  0.               xd8 =  0.
**************************************************
```

```
       lat-dir body axis coefficients
   clb =  -.122900e-02        cnb =   .237600e-02     cyb =
-.210860e-01
   clp =  -.512830e-02        cnp =  -.203200e-03     cyp =  0.
   clr =   .107820e-02        cnr =  -.865630e-02     cyr =  0.
  cld1 =   .381080e-04       cnd1 =  -.124030e-02    cyd1 =
.284130e-02
  cld2 =  -.107000e-03       cnd2 =   .435700e-03    cyd2 =
.552720e-03
  cld3 =   .788040e-03       cnd3 =   .502250e-03    cyd3 =
-.127680e-02
  cld4 =   .342770e-03       cnd4 =   .672630e-04    cyd4 =
-.145770e-03
  cld5 =   .662000e-03       cnd5 =  0.              cyd5 =  0.
  cld6 =   .910410e-04       cnd6 =  0.              cyd6 =  0.
  cld7 =  -.910410e-04       cnd7 =  0.              cyd7 =  0.
  cld8 =  0.                 cnd8 =  0.              cyd8 =  0.
  cld9 =  0.                 cnd9 =  0.              cyd9 =  0.

 ***************************************************
       lat-dir stab axis coefficients
   clb =  -.117776e-02        cnb =   .240181e-02     cyb =
-.210860e-01
   clp =  -.511116e-02        cnp =  -.279253e-03     cyp =  0.
   clr =   .100215e-02        cnr =  -.867344e-02     cyr =  0.
  cld1 =   .114974e-04       cnd1 =  -.124083e-02    cyd1 =
.284130e-02
  cld2 =  -.976305e-04       cnd2 =   .437895e-03    cyd2 =
.552720e-03
  cld3 =   .798631e-03       cnd3 =   .485233e-03    cyd3 =
-.127680e-02
  cld4 =   .344134e-03       cnd4 =   .598958e-04    cyd4 =
-.145770e-03
  cld5 =   .661848e-03       cnd5 =  -.141985e-04    cyd5 =  0.
  cld6 =   .910201e-04       cnd6 =  -.195264e-05    cyd6 =  0.
  cld7 =  -.910201e-04       cnd7 =   .195264e-05    cyd7 =  0.
  cld8 =  0.                 cnd8 =  0.              cyd8 =  0.
  cld9 =  0.                 cnd9 =  0.              cyd9 =  0.

    lat-dir stab axis dimensional derivatives(1/rad)
    nb =   9.55592              lb =  -36.9716         yb =  -345.104
    np =  -.254093e-01          lp =  -3.66945         yp =  0.
    nr =  -.789213              lr =   .719470         yr =  0.
   nd1 =  -4.93681             ld1 =   .360920        yd1 =   46.5021
   nd2 =   1.74222             ld2 =  -3.06477        yd2 =   9.04608
   nd3 =   1.93056             ld3 =   25.0702        yd3 =  -20.8967
   nd4 =   .238303             ld4 =   10.8029        yd4 =  -2.38574
   nd5 =  -.564307e-01         ld5 =   20.7764        yd5 =  0.
   nd6 =  -.776883e-02         ld6 =   2.85726        yd6 =  0.
   nd7 =   .776883e-02         ld7 =  -2.85726        yd7 =  0.
   nd8 =  0.                     ld8 =  0.                 yd8 =  0.
   nd9 =  0.                     ld9 =  0.                 yd9 =  0.
****************************************************
     lat-dir body axis dimensional derivatives( 1/rad)
    nb  =   9.44442              lb  =  -38.8658           yb  =  -345.104
    np  =  -.184724e-01          lp  =  -3.70902           yp  =  0.
    nr  =  -.786920              lr  =    .779803          yr  =  0.
   nd1  =  -4.93010              ld1 =   1.20512           yd1 =   46.5021
   nd2  =    1.73187             ld2 =  -3.38376           yd2 =   9.04608
   nd3  =    1.99641             ld3 =   24.9209           yd3 =  -20.8967
   nd4  =    .267365             ld4 =   10.8397           yd4 =  -2.38574
   nd5 =  0.                     ld5 =   20.9350           yd5 =  0.
   nd6 =  0.                     ld6 =   2.87907           yd6 =  0.
   nd7 =  0.                     ld7 =  -2.87907           yd7 =  0.
   nd8 =  0.                     ld8 =  0.                 yd8 =  0.
   nd9 =  0.                     ld9 =  0.                 yd9 =  0.
****************************************************
  lateral directional state matrix
     states = phi,beta,p,r,psi

   0.              0.                1.000           0.              0.
    .3448e-01     -.3697              .2145e-01     -.9998
.7399e-03
   0.             -39.84            -3.712            .8580          0.
   0.               9.935             .2727e-01     -.7975           0.
   0.              0.               0.               1.000           0.

  lateral directional input matrix
    for inputs: del1=rudder,del2=diff can
      del3=diff stab, del4=diff ail, del5=diff tef
      del6 to 9 are reverser vane ports
     row1             row2             row3             row4             row5
   0.               .4981e-01       -4.951            1.691          0.
   0.               .9690e-02        1.776           -3.559          0.
   0.             -.2238e-01         1.691            24.76          0.
   0.             -.2556e-02          .1339           10.83          0.
   0.              0.              -.2533             20.96          0.
   0.              0.              -.3552e-01         2.883          0.
   0.              0.                .3552e-01       -2.883          0.
   0.              0.               0.               0.              0.
   0.              0.               0.               0.              0.
```

B.3 LONGITUDINAL AND LATERAL DATA: MACH 1.4 AT 20,000 FEET

```
*****************************************************
               aircraft parameters
q  (dynamic pressure - lbs/ft**2) =   1335.91
s  (wing reference area - ft**2) =   608.000
c  (wing mean aerodynamic cord - ft) =   15.9399
b  (wing span - ft) =   42.7000
vt (trim velocity - ft/sec) =   1452.66
theta =  -.182000
w  (weight - lbs) =   37794.2
ixx  (slug-ft**2) =   25938.0
iyy  (slug-ft**2) =   185287.
izz  (slug-ft**2) =   206359.
ixz  (slug-ft**2) =  -2543.00
*****************************************************
*****************************************************
              alpha = -.182000
         longitudinal non-dim body axis coefficients(1/deg)
  cza =  -.654580e-01        cma =  -.134490e-01      cxa =
.108250e-02
  czq =  0.                  cmq =  -.256400          cxq =  0.
  czh =   .330080e-03        cmh =   .308330e-03      cxh =
.424060e-03
  czu =  -.142680e-01        cmu =   .133280e-01      cxu =
-.183300e-01
 czd1 =  -.800000e-03       cmd1 =   .331630e-02     cxd1 =
.267760e-04
 czd2 =  -.692730e-02       cmd2 =  -.836780e-02     cxd2 =
-.536760e-03
 czd3 =  -.889950e-03       cmd3 =  -.186670e-02     cxd3 =
-.337950e-04
 czd4 =  -.889960e-03       cmd4 =  -.186670e-02     cxd4 =
-.337950e-04
 czd5 =  0.                 cmd5 =  0.               cxd5 =  0.
 czd6 =  0.                 cmd6 =  0.               cxd6 =  0.
 czd7 =  0.                 cmd7 =  0.               cxd7 =  0.
 czd8 =  0.                 cmd8 =  0.               cxd8 =  0.
*****************************************************
    longitudinal axis dimensional derivatives
              body axis (1/rad)
   za =  -2595.36            ma =  -53.8436           xa =   42.9203
   zq =  0.                  mq =  -5.63188           xq =  0.
   zh =   .157242e-03        mh =   .148311e-04       xh =
.202011e-03
   zu =  -.135938e-01        mu =   .128219e-02       xu =
-.174639e-01
  zd1 =  -31.7194           md1 =   13.2769          xd1 =   1.06165
  zd2 =  -274.662           md2 =  -33.5003          xd2 =  -21.2821
```

```
    zd3 =  -35.2858                    md3 =   -7.47341            xd3 =   -1.33995
    zd4 =  -35.2862                    md4 =   -7.47341            xd4 =   -1.33995
    zd5 =  0.                          md5 =  0.                   xd5 =  0.
    zd6 =  0.                          md6 =  0.                   xd6 =  0.
    zd7 =  0.                          md7 =  0.                   xd7 =  0.
    zd8 =  0.                          md8 =  0.                   xd8 =  0.
**************************************************

        longitudnal state matrix(body axis)

    for state1=u,state2=q,state3=alpha,state4=theta

     -.174639e-01      4.61437          42.9203          -32.1998
      .128219e-02     -5.63188         -53.8436          0.
     -.935787e-05      .999995         -1.78663           .704109e-04
     0.                1.00000         0.                0.


        longitudnal input matrix

   for del1=canard,del2=stab,del3=tef,del4=dr aileron
        del5=rt rv, del6=rb rv, del7=lt rv, del8=lb rv


        row1            row2             row3              row4

     1.06165          13.2769         -.218354e-01   0.
    -21.2821         -33.5008         -.189075       0.
    -1.33995         -7.47341         -.242905e-01   0.
    -1.33995         -7.47341         -.242908e-01   0.
    0.               0.               0.             0.
    0.               0.               0.             0.
    0.               0.               0.             0.
    0.               0.               0.             0.

        longitudinal axis dimensional derivatives
                     stability axis (1/rad)
     za =  -2595.25                    ma =  -53.8371              xa =   51.0833
     zq =  0.                          mq =  -5.63188              xq =  0.
     zh =    .922398e-04               mh =    .161929e-04         xh =
.202804e-03
     zu =  -.797430e-02                mu =    .139992e-02         xu =
-.175324e-01
    zd1 =  -31.7158                   md1 =   13.2769             xd1 =   1.16240
    zd2 =  -274.728                   md2 =  -33.5008             xd2 =  -20.4095
    zd3 =  -35.2899                   md3 =  -7.47341             xd3 =  -1.22785
    zd4 =  -35.2903                   md4 =  -7.47341             xd4 =  -1.22785
    zd5 =  0.                         md5 =  0.                   xd5 =  0.
    zd6 =  0.                         md6 =  0.                   xd6 =  0.
    zd7 =  0.                         md7 =  0.                   xd7 =  0.
```

```
  zd8 =  0.                    md8 =  0.                xd8 =  0.
***************************************************
        lat-dir body axis coefficients
  clb  =  -.647440e-03         cnb  =   .946930e-03     cyb  =
-.165410e-01
  clp  =  -.551230e-02         cnp  =   .156840e-04     cyp  =  0.
  clr  =   .910580e-03         cnr  =  -.153450e-01     cyr  =  0.
 cld1 =   .292080e-04        cnd1 =  -.364720e-03    cyd1 =
.747910e-03
 cld2 =  -.562730e-03        cnd2 =   .435740e-04    cyd2 =
.365130e-03
 cld3 =   .666430e-03        cnd3 =  5.37130         cyd3 =
-.452300e-03
 cld4 =   .455370e-04        cnd4 =   .429460e-05    cyd4 =
-.181700e-04
 cld5 =   .675220e-03        cnd5 =  0.              cyd5 =  0.
 cld6 =   .346180e-04        cnd6 =  -.695530e-06    cyd6 =  0.
 cld7 =  -.346170e-04        cnd7 =   .695530e-06    cyd7 =  0.
 cld8 =  0.                  cnd8 =  0.              cyd8 =  0.
 cld9 =  0.                  cnd9 =  0.              cyd9 =  0.

***************************************************
       lat-dir stab axis coefficients
  clb  =  -.650445e-03         cnb  =   .944869e-03     cyb  =
-.165410e-01
  clp  =  -.551534e-02         cnp  =   .469080e-04     cyp  =  0.
  clr  =   .941804e-03         cnr  =  -.153420e-01     cyr  =  0.
 cld1 =   .303664e-04        cnd1 =  -.364625e-03    cyd1 =
.747910e-03
 cld2 =  -.562866e-03        cnd2 =   .417863e-04    cyd2 =
.365130e-03
 cld3 =  -.163955e-01        cnd3 =  5.37128         cyd3 =
-.452300e-03
 cld4 =   .455231e-04        cnd4 =   .443923e-05    cyd4 =
-.181700e-04
 cld5 =   .675217e-03        cnd5 =   .214483e-05    cyd5 =  0.
 cld6 =   .346200e-04        cnd6 =  -.535563e-06    cyd6 =  0.
 cld7 =  -.346190e-04        cnd7 =   .535566e-06    cyd7 =  0.
 cld8 =  0.                  cnd8 =  0.              cyd8 =  0.
 cld9 =  0.                  cnd9 =  0.              cyd9 =  0.

    lat-dir stab axis dimensional derivatives(1/rad)
   nb  =   9.09807             lb  =  -49.8592          yb  =  -655.838
   np  =   .663833e-02         lp  =  -6.21357          yp  =  0.
   nr  =  -2.17116             lr  =   1.06103          yr  =  0.
  nd1 =  -3.51095            ld1 =   2.32771         yd1 =   29.6541
  nd2 =   .402357            ld2 =  -43.1459         yd2 =   14.4771
  nd3 =   51719.6            ld3 =  -1256.78         yd3 =  -17.9333
  nd4 =   .427450e-01        ld4 =   3.48953         yd4 =  -.720426
  nd5 =   .206524e-01        ld5 =   51.7531         yd5 =  0.
  nd6 =  -.563834e-02         ld6 =   2.65376         yd6 =  0.
  nd7 =   .563837e-02         ld7 =  -2.65369         yd7 =  0.
  nd8 =  0.                   ld8 =  0.               yd8 =  0.
  nd9 =  0.                   ld9 =  0.               yd9 =  0.
******************************************************
    lat-dir body axis dimensional derivatives( 1/rad)
   nb  =   9.11855             lb  =  -49.6014         yb  =  -655.838
   np  =   .221972e-02         lp  =  -6.20671         yp  =  0.
   nr  =  -2.17174             lr  =   1.02529         yr  =  0.
  nd1 =  -3.51210             ld1 =   2.23767         yd1 =   29.6541
  nd2 =   .419600             ld2 =  -43.1117         yd2 =   14.4771
  nd3 =   51723.4             ld3 =   51.0563         yd3 =  -17.9333
  nd4 =   .413552e-01         ld4 =   3.48866         yd4 =  -.720426
  nd5 =  0.                   ld5 =   51.7297         yd5 =  0.
  nd6 =  -.669767e-02         ld6 =   2.65214         yd6 =  0.
  nd7 =   .669767e-02         ld7 =  -2.65207         yd7 =  0.
  nd8 =  0.                   ld8 =  0.               yd8 =  0.
  nd9 =  0.                   ld9 =  0.               yd9 =  0.
******************************************************
  lateral directional state matrix
     states = phi,beta,p,r,psi

   0.              0.               1.000          0.              0.
    .2217e-01      -.4515          -.3176e-02      -1.000
-.7041e-04
   0.              -50.56          -6.214           1.240          0.
   0.               9.742           .7880e-01      -2.187          0.
   0.              0.              0.               1.000          0.

  lateral directional input matrix
    for inputs: del1=rudder,del2=diff can
     del3=diff stab, del4=diff ail, del5=diff tef
     del6 to 9 are reverser vane ports
     row1            row2            row3           row4            row5
   0.              .2041e-01      -3.544           2.585          0.
   0.              .9966e-02       .9520          -43.21          0.
   0.             -.1235e-01       .5179e+05      -5026.          0.
   0.             -.4359e-03      -.1638e-02       3.439          0.
   0.              0.             -.6382           51.79          0.
   0.              0.             -.3943e-01       2.656          0.
   0.              0.              .3943e-01      -2.656          0.
   0.              0.              0.              0.             0.
   0.              0.              0.              0.             0.
```

B.4 LONGITUDINAL AND LATERAL DATA: MACH 2.0 AT 40,000 FEET

```
******************************************************
                  aircraft parameters
q  (dynamic pressure - lbs/ft**2) =   1104.44
s  (wing reference area - ft**2) =   608.000
c  (wing mean aerodynamic cord - ft) =   15.9400
b  (wing span - ft) =   42.7000
vt (trim velocity - ft/sec) =   1942.00
theta =   .151500
w  (weight - lbs) =   37794.2
ixx  (slug-ft**2) =   25638.0
iyy  (slug-ft**2) =   185287.
izz  (slug-ft**2) =   206359.
ixz  (slug-ft**2) =  -2543.00
*****************************************************
                  alpha =  .151500
         longitudinal non-dim body axis coefficients(1/deg)
   cza =  -.499000e-01          cma =  -.745840e-02     cxa =
-.973700e-04
   czq =  0.                    cmq =  -.654300         cxq =  0.
   czh =  -.735710e-03          cmh =  -.118360e-02     cxh =
.903860e-03
   czu =   .204110e-01          cmu =  -.328360e-01     cxu =  -2.50760
  czd1 =  -.125090e-02         cmd1 =   .214210e-02    cxd1 =
-.585440e-04
  czd2 =  -.404930e-02         cmd2 =  -6.38570        cxd2 =
-.464650e-03
  czd3 =  -.113030e-02         cmd3 =  -.254300e-02    cxd3 =
-.231340e-05
  czd4 =  -.113090e-02         cmd4 =  -.254300e-02    cxd4 =
-.231340e-05
  czd5 =  0.                   cmd5 =  0.              cxd5 =  0.
  czd6 =  0.                   cmd6 =  0.              cxd6 =  0.
  czd7 =  0.                   cmd7 =  0.              cxd7 =  0.
  czd8 =  0.                   cmd8 =  0.              cxd8 =  0.
 *****************************************************
    longitudinal axis dimensional derivatives
               body axis (1/rad)
   za =  -1635.69              ma =  -24.6864           xa =  -3.19172
   zq =  0.                    mq =  -8.88787           xq =  0.
   zh =  -.216738e-03          mh =  -.352083e-04       xh =
.266274e-03
   zu =   .120260e-01          mu =  -.195353e-02       xu =  -1.47746
  zd1 =  -41.0036             md1 =   7.09009         xd1 =  -1.91903
  zd2 =  -132.733             md2 =  -21135.9         xd2 =  -15.2309
  zd3 =  -37.0701             md3 =  -8.41701         xd3 =
-.758316e-01
  zd4 =  -37.0701             md4 =  -8.41701         xd4 =
-.758316e-01
  zd5 =  0.                      md5 =  0.                      xd5 =  0.
  zd6 =  0.                      md6 =  0.                      xd6 =  0.
  zd7 =  0.                      md7 =  0.                      xd7 =  0.
  zd8 =  0.                      md8 =  0.                      xd8 =  0.
***********************************************

     longitudnal  state  matrix(body  axis)

  for state1=u,state2=q,state3=alpha,state4=theta

   -1.47746          -5.13438          -3.19172          -32.1939
   -.195353e-02      -8.88787          -24.6864          0.
    .619260e-05       .999397          -.842269          -.438426e-04
   0.                 1.00000          0.                0.


     longitudnal  input  matrix

 for del1=canard,del2=stab,del3=tef,del4=dr aileron
     del5=rt rv, del6=rb rv, del7=lt rv, del8=lb rv


     row1             row2              row3              row4

  -1.91303          7.09009          -.211141e-01    0.
  -15.2309         -21135.9          -.683487e-01    0.
  -.758316e-01     -8.41701          -.190886e-01    0.
  -.758316e-01     -8.41701          -.190886e-01    0.
  0.               0.                0.              0.
  0.               0.                0.              0.
  0.               0.                0.              0.
  0.               0.                0.              0.

     longitudinal  axis  dimensional  derivatives
                     stability  axis  (1/rad)
   za  =  -1635.74                ma  =  -24.6762            xa  =
.698505e-01
   zq  =  0.                      mq  =  -8.88787            xq  =  0.
   zh  =  -.247006e-03            mh  =  -.358139e-04        xh  =
.266268e-03
   zu  =   .137055e-01            mu  =  -.198714e-02        xu  =  -1.47743
  zd1  =  -40.9984               md1  =   7.09003           xd1  =  -2.02744
  zd2  =  -132.692               md2  =  -21135.9           xd2  =  -15.5818
  zd3  =  -37.0698               md3  =  -8.41701           xd3  =  -.173851
  zd4  =  -37.0698               md4  =  -8.41701           xd4  =  -.173851
  zd5  =  0.                     md5  =  0.                 xd5  =  0.
  zd6  =  0.                     md6  =  0.                 xd6  =  0.
  zd7  =  0.                     md7  =  0.                 xd7  =  0.
  zd8  =  0.                     md8  =  0.                 xd8  =  0.
```

```
*****************************************************
        lat-dir body axis coefficients
  clb =  -.760580e-03         cnb =   .162300e-03     cyb =
-.144930e-01
  clp =  -.555180e-02         cnp =  -.249150e-05     cyp =  0.
  clr =   .675400e-03         cnr =  -.375400e-01     cyr =  0.
 cld1 =   .339610e-04        cnd1 =  -.275500e-03    cyd1 =
.498310e-03
 cld2 =  -.434310e-03        cnd2 =   .158240e-03    cyd2 =
.829290e-03
 cld3 =   .460600e-03        cnd3 =   .247690e-04    cyd3 =  -2.93180
 cld4 =   .104500e-03        cnd4 =   .562860e-05    cyd4 =
-.770730e-04
 cld5 =   .671730e-03        cnd5 =  0.              cyd5 =  0.
 cld6 =   .376750e-04        cnd6 =  0.              cyd6 =  0.
 cld7 =  -.376750e-04        cnd7 =  0.              cyd7 =  0.
 cld8 =  0.                  cnd8 =  0.              cyd8 =  0.
 cld9 =  0.                  cnd9 =  0.              cyd9 =  0.

*****************************************************
       lat-dir stab axis coefficients
  clb =  -.760148e-03         cnb =   .164311e-03     cyb =
-.144930e-01
  clp =  -.555024e-02         cnp =  -.870782e-04     cyp =  0.
  clr =   .590813e-03         cnr =  -.375416e-01     cyr =  0.
 cld1 =   .332324e-04        cnd1 =  -.275589e-03    cyd1 =
.498310e-03
 cld2 =  -.433890e-03        cnd2 =   .159388e-03    cyd2 =
.829290e-03
 cld3 =   .460664e-03        cnd3 =   .235510e-04    cyd3 =  -2.93180
 cld4 =   .104515e-03        cnd4 =   .535226e-05    cyd4 =
-.770730e-04
 cld5 =   .671728e-03        cnd5 =  -.177617e-05    cyd5 =  0.
 cld6 =   .376749e-04        cnd6 =  -.996191e-07    cyd6 =  0.
 cld7 =  -.376749e-04        cnd7 =   .996191e-07    cyd7 =  0.
 cld8 =  0.                  cnd8 =  0.              cyd8 =  0.
 cld9 =  0.                  cnd9 =  0.              cyd9 =  0.

    lat-dir stab axis dimensional derivatives(1/rad)
   nb =   1.30819            lb =  -48.1185          yb =  -475.070
   np =  -.762187e-02        lp =  -3.86255          yp =  0.
   nr =  -3.28593            lr =   .411162          yr =  0.
  nd1 =  -2.19415           ld1 =   2.10366         yd1 =   16.3343
  nd2 =   1.26899           ld2 =  -27.4659         yd2 =   27.1835
  nd3 =   .137505           ld3 =   29.1607         yd3 =  -96102.3
  nd4 =   .425129e-01       ld4 =   6.61592         yd4 =  -2.52640
  nd5 =  -.141413e-01       ld5 =   42.5213         yd5 =  0.
  nd6 =  -.793134e-03       ld6 =   2.38487         yd6 =  0.
  nd7 =   .793134e-03       ld7 =  -2.38487         yd7 =  0.
  nd8 =  0.                 ld8 =  0.               yd8 =  0.
  nd9 =  0.                 ld9 =  0.                   yd9 =  0.
**************************************************
    lat-dir body axis dimensional derivatives( 1/rad)
   nb =    1.29209            lb =  -48.1731              yb =  -475.070
   np =  -.218063e-03         lp =  -3.86583              yp =  0.
   nr =  -3.28561             lr =    .470294             yr =  0.
  nd1 =  -2.19328            ld1 =   2.15100             yd1 =   16.3343
  nd2 =   1.25976            ld2 =  -27.5080             yd2 =   27.1835
  nd3 =   .197188            ld3 =   29.1732             yd3 =  -96102.3
  nd4 =   .448098e-01        ld4 =   6.61875             yd4 =  -2.52640
  nd5 =  0.                  ld5 =   42.5456             yd5 =  0.
  nd6 =  0.                  ld6 =   2.38623             yd6 =  0.
  nd7 =  0.                  ld7 =  -2.38623             yd7 =  0.
  nd8 =  0.                  ld8 =  0.                   yd8 =  0.
  nd9 =  0.                  ld9 =  0.                   yd9 =  0.
**************************************************
  lateral directional state matrix
     states = phi,beta,p,r,psi

   0.              0.                1.000        0.              0.
    .1658e-01      -.2446            .2644e-02    -1.000
.4384e-04
   0.              -48.36           -3.870         .7934          0.
   0.               1.888            .4748e-01    -3.295          0.
   0.              0.               0.             1.000          0.

  lateral directional input matrix
    for inputs: del1=rudder,del2=diff can
     del3=diff stab, del4=diff ail, del5=diff tef
     del6 to 9 are reverser vane ports
     row1             row2             row3             row4             row5
   0.               .8411e-02       -2.222           2.369           0.
   0.               .1400e-01        1.601          -27.66           0.
   0.              -49.49           -.1625           29.19           0.
   0.              -.1301e-02       -.3680e-01       6.622           0.
   0.              0.               -.5249           42.60           0.
   0.              0.               -.2944e-01       2.383           0.
   0.              0.                .2944e-01      -2.389           0.
   0.              0.               0.              0.               0.
   0.              0.               0.              0.               0.
**************************************************
```

Appendix C: ODEF15

In order to analyze the CGT/PI/KF controller designs used in the course of this thesis, ODEF15 was written. ODEF15 is written in FORTRAN V and is hosted on the CDC Cyber mainframe computer. It is derived from expanding an analysis package known as ODEACT written by Maj W. Miller (16). The software is completely interactive, providing the user with prompts for each input. Also, a file structure is available which allows the user both to read and to write data to local files. The code is specific to the STOL F-15 only in the actuator dynamics and rate/position limits, which are "hardwired" into the software. However, by eliminating the entries of the actuator dynamics and saturation limits from the external subroutine FSTOL, any CGT/PI/KF-controlled system could be analyzed. Files can be formed for Kalman filter-based systems as well as full state feedback systems. The user must declare the library routines IMSL and Ode (International Mathematical and Statistical Library, and Ordinary Differential Equations solver) before running ODEF15 as these libraries are referenced in the main program. For the Kalman filter-based systems, the user is asked how many iterations are desired for the Monte Carlo analysis. It is advised that the user limit the number of iterations to 7 or less to avoid long computational times (10 run averages can take in excess of 10 minutes).

After data has been entered and the analysis has been performed, the user is provided with a menu to choose output data to be plotted to the terminal using the PLOTLP plotting routine from Reference 7. Finally, plot files are created and written to user-specified output files. These plot files can be routed to the CALCOMP plotter using the PLOTM routine available through the Super Procfile available on the NOS operating system (2).

```
     PROGRAM ODEF15
C
C**********************************************************************
C
C SIMULATION PROGRAM TO TEST A PI  OR CGT/PI, WITH/WITHOUT KALMAN
C FILTER, BASED ON A 4-STATE MODEL OF THE STOL F-15.  OPTIONS
C INCLUDE MODIFICATION OF DYNAMICS MATRIX, USE OF 2-STATE
C ACTUATOR MODELS, APPLICATION OF RATE/POSITION LIMITS ON ACTUATORS,
C AND EMPLOYMENT OF ANTI-WINDUP COMPENSATION.  USER SUPPLIES DYNAMICS
C MATRIX, OUTPUT MATRIX, CGT COMMAND MODEL, CONTROLLER GAINS AND
C KALMAN FILTER MATRICES.
C
C DATE OF LAST REVISION:  01 NOV 85
C LIBRARIES USED:  ODE,IMSL5
C
C**********************************************************************
C
      REAL WORK(352),X(12),DX(12),OUT(51,4),T,TOUT,TSAMP,PLTVEC(260)
      REAL XTEMP(12)
      REAL AWORK(4,4),BWORK(4,4),CWORK(4,4),AM(4,4),BM(4,4),Z(3)
      REAL IPHIX(4,4)
      REAL RELERR,ABSERR,DSIM,OUT1(51,5),UOUT2(51,4),UOUT3(51,5),Y(4)
      REAL V(3),EVTMP(2)
      INTEGER NR,MMM,JJJ
      DOUBLE PRECISION DSEED
      INTEGER I,J,K,IFLAG,JFLAG,JCFLAG,NFLAG,IWORK(5),IDSIM
      INTEGER MMFLAG,IIFLAG
      COMMON/MATRIX/UO(2),A(12,12),C(4,12),KX(4,12),KZ(4,4),KXM(4,4),
     1 KXU(4,4),PHI(4,4),PHINT(4,4),CM(4,4),B(3,3),EV(2),KFLAG,MM
      COMMON/CONTRL/UNEW(4),UOLD(4),UCMD(4),UCOLD(4),XOLD(12),
     1 XMOLD(4),XM(4),MFLAG,EVA(2),
     1 H(3,4),PHIX(4,4),BD(4,3),QD(4,4),R(3,3),XHP(4),XHM(4),K(4,3)
C
      EXTERNAL F1,F3,F4,FSTOL
      CHARACTER ANSW*1,TITLE*50,DATA*6,SAVE*6,PLOT*6
C
C
C**********************************************************************
C
C INPUT SECTION.  DATA MAY BE READ IN FROM AN 'OLD' FILE, AND SAVED
C TO ANY OTHER FILE.  ONLY ONE SET OF DATA PER FILE NAME.  PLOTS ARE
C AUTOMATICALLY SAVED IN A 'PLOT FILE'.
C
C**********************************************************************
C
 20     PRINT*,' INCORPORATE KALMAN FILTER? Y/N:'
        READ(*,'(A)') ANSW
        IF (ANSW.NE.'Y'.AND.ANSW.NE.'N') GO TO 20
        IF (ANSW.EQ.'Y') IIFLAG=1
        IF (ANSW.EQ.'N') IIFLAG=0
 902    PRINT*,'DATA TO BE READ FROM FILE? Y/N: '
      READ(*,'(A)')ANSW
      IF(ANSW.NE.'Y'.AND.ANSW.NE.'N') GO TO 902
      IF(ANSW.EQ.'N') GO TO 30
         JCFLAG=0
```

```
            PRINT*,'ENTER NAME OF DATA FILE: '
            READ(*,'(A)')DATA
            OPEN(2,FILE=DATA,STATUS='OLD',FORM='UNFORMATTED',ERR=902)
            READ(2)((A(I,J),I=1,12),J=1,12)
          READ(2)((B(I,J),I=1,3),J=1,3)
            READ(2)((C(I,J),I=1,4),J=1,12)
            READ(2)((KX(I,J),I=1,4),J=1,12)
            READ(2)((KZ(I,J),I=1,4),J=1,4)
            READ(2)((KXU(I,J),I=1,4),J=1,4)
            READ(2)((KXM(I,J),I=1,4),J=1,4)
            READ(2)((AM(I,J),I=1,4),J=1,4)
            READ(2)((BM(I,J),I=1,4),J=1,4)
            READ(2)((CM(I,J),I=1,4),J=1,4)
          IF(IIFLAG.EQ.0) GO TO 901
          READ(2)((PHIX(I,J),I=1,4),J=1,4)
          READ(2)((BD(I,J),I=1,4),J=1,3)
          READ(2)((QD(I,J),I=1,4),J=1,4)
          READ(2)(
(H(I,J),I=1,3),J=1,4)
          READ(2)((K(I,J),I=1,4),J=1,3)
          READ(2)((R(I,J),I=1,3),J=1,3)
      901 CONTINUE
            REWIND(2)
            CLOSE(2)
          GO TO 140
      C
      C FOR KEYBOARD INPUT, ONLY NON-ZERO MATRIX ELEMENTS ARE REQUIRED.
      C NO NON-ZERO ENTRIES SHOULD BE MADE FOR COLUMNS 5,6,7,9,10 OR 11
      C OF A OR KX, BUT NO PROTECTION PROVIDED AGAINST DOING SO.
      C
      30  DO 50 I=1,12
            DO 50 J=1,12
              A(I,J)=0.0
      50  CONTINUE
          DO 42 I=1,3
           DO 42 J=1,3
               R(I,J)=0.0
            B(I,J)=0.0
      42  CONTINUE
          DO 64 I=1,12
            DO 60 J=1,4
              KX(J,I)=0.0
      60    CONTINUE
            DO 64 L=1,4
              C(L,I)=0.0
      64  CONTINUE
          DO 66 I=1,4
            DO 66 J=1,4
                 PHIX(I,J)=0.0
              KZ(I,J)=0.0
              KXU(I,J)=0.0
              KXM(I,J)=0.0
      66  CONTINUE
          DO 70 I=1,4
            DO 70 J=1,4
              AM(I,J)=0
```

```
                  BM(I,J)=0
                  CM(I,J)=0
                  QD(I,J)=0.0
      70    C
ONTINUE
            DO 903 I=1,4
               DO 903 J=1,3
                  H(J,I)=0.0
                  K(I,J)=0.0
                  BD(I,J)=0.0
      903   CONTINUE
            JFLAG=0
      72    PRINT*,'ENTER DYNAMICS MATRIX: '
            CALL EDIT(A,12,12)
            IF(JFLAG.NE.0) GO TO 140
     74     PRINT*,'ENTER CONTROL MATRIX: '
            CALL EDIT(B,3,3)
            IF(JFLAG.NE.0) GO TO 140
      76    PRINT*,'ENTER OUTPUT MATRIX: '
            CALL EDIT(C,4,12)
            IF(JFLAG.NE.0) GO TO 140
      78    PRINT*,'ENTER KX MATRIX: '
            CALL EDIT(KX,4,12)
            IF(JFLAG.NE.0) GO TO 140
      80    PRINT*,'ENTER KZ MATRIX: '
            CALL EDIT(KZ,4,4)
            IF(JFLAG.NE.0) GO TO 140
      82    PRINT*,'ENTER KXM MATRIX: '
            CALL EDIT(KXM,4,4)
            IF(JFLAG.NE.0) GO TO 140
      84    PRINT*,'ENTER KXU MATRIX: '
            CALL EDIT(KXU,4,4)
            IF(JFLAG.NE.0) GO TO 140
      86    PRINT*,'ENTER MODEL DYNAMICS MATRIX: '
            JCFLAG=0
            CALL EDIT(AM,4,4)
            IF(JFLAG.NE.0) GO TO 140
      88    PRINT*,'ENTER MODEL CONTROL MATRIX: '
            JCFLAG=0
            CALL EDIT(BM,4,4)
            IF(JFLAG.NE.0) GO TO 140
      90    PRINT*,'ENTER MODEL OUTPUT MATRIX: '
            CALL EDIT(CM,4,4)
            IF(JFLAG.NE.0) GO TO 140
            IF(IIFLAG.EQ.0) GO TO 140
      904   PRINT*,'ENTER STATE TRANSITION MATRIX:'
            CALL EDIT(PHIX,4,4)
            IF(JFLAG.NE.0) GO TO 140
      905   PRINT*,'ENTER DISCRETE TIME INPUT MATRIX'
            CALL EDIT(BD,4,3)
            IF(JFLAG.NE.0) GO TO 140
      906   PRINT*,'ENTER DISCRETE TIME COVARIANCE MATRIX:'
            CALL EDIT(QD,4,4)
            IF(JFLAG.NE.0) GO TO 140
      907    PRINT*,'ENTER MEASUREMENT MATRIX:'
            CALL EDIT(H,3,4)
      IF(JFLAG.NE.0) GO TO 140
  908  PRINT*,'ENTER KALMAN FILTER GAINS:'
      CALL EDIT(K,4,3)
      IF(JFLAG.NE.0) GO TO 140
909   PRINT*,'ENTER MEASUREMENT NOISE COV. MATRIX'
      CALL EDIT(R,3,3)
      IF(JFLAG.NE.0) GO TO 140
 140  PRINT*,'ANY CHANGES TO MATRICES? Y/N: '
      READ(*,'(A)')ANSW
      IF(ANSW.NE.'Y'.AND.ANSW.NE.'N') GO TO 140
 142  IF(ANSW.EQ.'Y') THEN
         PRINT*,'1=A     2=C    3=KX    4=KZ    5=KXM    6=KXU'
         PRINT*,'7=AM    8=BM    9=CM  10=B '
      PRINT*,'11=PHIX   12=BD   13=QD   14=H   15=K  16=R'
      PRINT*,' ENTER CHOICE:'
         READ*,JFLAG
         GO TO (72,76,78,80,82,84,86,88,90,74,904,905,
     1 906,907,908,909) JFLAG
      ELSE
         JFLAG=0
      END IF
 150  PRINT*,'WRITE DATA TO OUTPUT FILE? Y/N: '
      READ(*,'(A)')ANSW
      IF(ANSW.NE.'Y'.AND.ANSW.NE.'N') GO TO 150
      IF(ANSW.EQ.'Y') THEN
         PRINT*,'ENTER NAME OF OUTPUT FILE: '
         READ(*,'(A)')SAVE
         OPEN(3,FILE=SAVE,FORM='UNFORMATTED',ERR=80)
         WRITE(3)((A(I,J),I=1,12),J=1,12)
      WRITE(3)((B(I,J),I=1,3),J=1,3)
         WRITE(3)((C(I,J),I=1,4),J=1,12)
         WRITE(3)((KX(I,J),I=1,4),J=1,12)
         WRITE(3)((KZ(I,J),I=1,4),J=1,4)
         WRITE(3)((KXU(I,J),I=1,4),J=1,4)
         WRITE(3)((KXM(I,J),I=1,4),J=1,4)
         WRITE(3)((AM(I,J),I=1,4),J=1,4)
         WRITE(3)((BM(I,J),I=1,4),J=1,4)
         WRITE(3)((CM(I,J),I=1,4),J=1,4)
      IF(IIFLAG.EQ.0) GO TO 910
         WRITE(3)((PHIX(I,J),I=1,4),J=1,4)
         WRITE(3)((BD(I,J),I=1,4),J=1,3)
         WRITE(3)((QD(I,J),I=1,4),J=1,4)
      WRITE(3)((H(I,J),I=1,3),J=1,4)
         WRITE(3)((K(I,J),I=1,4),J=1,3)
         WRITE(3)((R(I,J),I=1,3),J=1,3)
  910 CONTINUE
         ENDFILE(3)
         REWIND(3)
         CLOSE(3)
      END IF
C
C***********************************************************************
C
C NOW SET UP CONDITIONS FOR CALLING ODE.  ALL INITIAL CONDITIONS
C ARE ZERO UNLESS CHANGED BY USER INPUT.
```

```
C
C**********************************************************************
C
      IF(JCFLAG.EQ.0) THEN
         PRINT*,'ENTER SAMPLING TIME: '
         READ*,TSAMP
         CALL DSCRT(AM,4,TSAMP,PHI,PHINT,30,AWORK,BWORK,CWORK)
         CALL MATML(PHINT,BM,AWORK,4,4,4)
         CALL COPYMT(AWORK,PHINT,4,4)
      PRINT*,'    '
         JCFLAG=1
      END IF
      PRINT*,'ENTER CANARD TRIM ANGLE OF ATTACK'
      READ*,EV(1)
      PRINT*,'ENTER STABILATOR TRIM ANGLE OF ATTACK IN RADIANS '
      READ*,EV(2)
      EVTMP(1)=EV(1)
      EVTMP(2)=EV(2)
 154  PRINT*,' RATE/POSITION LIMITS? Y/N: '
      READ(*,'(A)')ANSW
      IF(ANSW.NE.'Y'.AND.ANSW.NE.'N') GO TO 154
      IF(ANSW.EQ.'Y') MFLAG=1
      IF(ANSW.EQ.'N') MFLAG=0
 156  PRINT*,' ANTI-WINDUP COMPENSATION? Y/N: '
      READ(*,'(A)')ANSW
      IF(ANSW.NE.'Y'.AND.ANSW.NE.'N') GO TO 156
      IF(ANSW.EQ.'Y') NFLAG=1
      IF(ANSW.EQ.'N') NFLAG=0
 222    PRINT*,'EMPLOY ACTUATOR DYNAMICS? Y/N: '
       READ(*,'(A)')ANSW
       IF(ANSW.NE.'Y'.AND.ANSW.NE.'N')GO TO 222
       IF(ANSW.EQ.'Y')MM=0
       IF(ANSW.EQ.'N')MM=1
 158  PRINT*,'ENTER DESIRED RESPONSE DURATION: '
      READ*,DSIM
      IF(DSIM.LT.0.1) GO TO 158
      IDSIM=INT(DSIM/(50.0*TSAMP)+.99)
 160  DO 170 I=1,12
         X(I)=0.0
         XOLD(I)=0.0
 170  CONTINUE
      EVA(1)=0.0
      EVA(2)=0.0
      DO 172 I=1,4
         UOLD(I)=0.0
          Y(I)=0.0
      XHP(I)=0.0
      XHM(I)=0.0
         UNEW(I)=0.0
 172  CONTINUE
      DO 175 I=1,4
         UCMD(I)=0.0
         UCOLD(I)=0.0
         XM(I)=0.0
         XMOLD(I)=0.0
      Y(I)=0.0
 175  CONTINUE
 180  PRINT*,'ENTER I AND X(I); 0,0 TO TERMINATE: '
 190  READ*,IIII,EL
      IF(IIII.LE.12.AND.IIII.GE.1) THEN
         X(IIII)=EL
         GO TO 190
      ELSE IF(IIII.EQ.0) THEN
         GO TO 200
      ELSE
         PRINT*,'SUBSCRIPT OUT OF RANGE'
         GO TO 180
      END IF
 200   PRINT*,' SELECT COMMAND INPUT & STEP MAGNITUDE:'
       READ*,IK,ELL
       IF(IK.LE.3.AND.IK.GE.1)THEN
       UCMD(IK)=ELL
       ELSE
       PRINT*,' SUBSCRIPT OUT OF RANGE'
       GO TO 200
       END IF
      T=0.0
      TOUT=0.0
      IFLAG=-1
      RELERR=1.E-08
      ABSERR=1.E-07
      U0(1)=0.0
      U0(2)=0.0
      IF(IIFLAG.EQ.1)THEN
      PRINT*,'ENTER SOURCE OF OUTPUT MEASUREMENTS'
      PRINT*,'1=SYSTEM STATES   2=OUTPUT VARIABLES'
      READ*,NN
      IF(NN.EQ.1)THEN
 913  PRINT*,'ENTER STATES TO BE MEASURED (3)'
      READ*,L,M,N
      IF((L.GT.12).OR.(L.LT.1)) GO TO 913
      IF((M.GT.12).OR.(M.LT.1)) GO TO 913
      IF((N.GT.12).OR.(M.LT.1)) GO TO 913
      ELSE
 914  PRINT*,'ENTER OUTPUT VARIABLES TO BE MEASURED (3)'
      READ*,L,M,N
      IF((L.GT.3).OR.(L.LT.1)) GO TO 914
      IF((M.GT.3).OR.(M.LT.1)) GO TO 914
      IF((N.GT.3).OR.(N.LT.1)) GO TO 914
      END IF
      END IF
      XITER=1.
      IF(IIFLAG.EQ.0) GO TO 920
 951  PRINT*,'ENTER # OF ITERATIONS FOR FILTER AVERAGE'
      READ*,XITER
      NR=3
      PRINT*,'SELECT SEED VALUE FOR RANDOM # GENERATION'
      READ*,DSEED
 937  PRINT*,'CONTROL BASED ON XHAT+ OR XHAT-?'
      PRINT*,'XHAT+=1      XHAT-=2'
      READ*,JMFLAG
      IF((JMFLAG.GT.2).OR.(JMFLAG.LT.1))GO TO 937
920   CONTINUE
      DO 750 II=1,51
      OUT(II,1)=0.
      OUT1(II,1)=0.
      UOUT2(II,1)=0.
      UOUT3(II,1)=0.
      OUT(II,2)=0.
      OUT(II,3)=0.
      OUT(II,4)=0.
      OUT1(II,2)=0.
      OUT1(II,3)=0.
      OUT1(II,4)=0.
      OUT1(II,5)=0.
      UOUT2(II,2)=0.
      UOUT2(II,3)=0.
      UOUT2(II,4)=0.
      UOUT3(II,2)=0.
      UOUT3(II,3)=0.
      UOUT3(II,4)=0.
      UOUT3(II,5)=0.
750   CONTINUE
      IF(IIFLAG.EQ.0) GO TO 1040
      DO 780 IJK=1,XITER
      T=0.0
      TOUT=0.0
      IFLAG=-1
      RELERR=1.E-08
      ABSERR=1.E-07
      UO(1)=0.0
      UO(2)=0.0
      DO 1010 I=1,4
      XM(I)=0.0
      UOLD(I)=0.0
      UNEW(I)=0.0
      UCMD(I)=0.
      UCOLD(I)=0.
      XMOLD(I)=0.
      Y(I)=0.
      XHP(I)=0.
        XHM(I)=0.
1010  CONTINUE
      DO 1020 I=1,12
      XTEMP(I)=0.
      X(I)=0.
      XOLD(I)=0.
1020  CONTINUE
      X(IIII)=EL
      UCMD(IK)=ELL
      EVA(1)=0.0
      EVA(2)=0.0
      EV(1)=EVTMP(1)
      EV(2)=EVTMP(2)
1040  CONTINUE
```

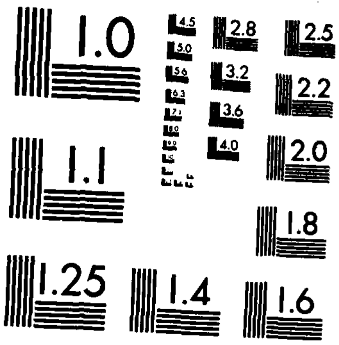
```
      DO 300 I=IDSIM+1,51*IDSIM+1
      CALL ERXSET(300,0)
      CALL MATML(C,X,Y,4,12,1)
         IF(IDSIM.NE.1.AND.MOD(I,IDSIM).EQ.1) THEN
            J=INT(I/IDSIM)
           OUT(J,1)=TOUT+OUT(J,1)
           OUT(J,2)=Y(1)+OUT(J,2)
           OUT(J,3)=Y(2)+OUT(J,3)
           OUT(J,4)=Y(3)+OUT(J,4)
C
          OUT1(J,1)=TOUT+OUT1(J,1)
          OUT1(J,2)=UNEW(1)+OUT1(J,2)
          OUT1(J,3)=UNEW(2)+OUT1(J,3)
          OUT1(J,4)=UNEW(3)+OUT1(J,4)
          OUT1(J,5)=UNEW(4)+OUT1(J,5)
C
          UOUT2(J,1)=TOUT+UOUT2(J,1)
          UOUT2(J,2)=X(5)+UOUT2(J,2)
          UOUT2(J,3)=X(7)+UOUT2(J,3)
          UOUT2(J,4)=X(9)+UOUT2(J,4)
C
C
          UOUT3(J,1)=TOUT+UOUT3(J,1)
          UOUT3(J,2)=X(1)+UOUT3(J,2)
          UOUT3(J,3)=X(2)+UOUT3(J,3)
          UOUT3(J,4)=X(3)+UOUT3(J,4)
          UOUT3(J,5)=X(4)+UOUT3(J,5)
         ELSE IF(IDSIM.EQ.1) THEN
            J=I-IDSIM
            OUT(J,1)=TOUT+OUT(J,1)
            OUT(J,2)=Y(1)+OUT(J,2)
            OUT(J,3)=Y(2)+OUT(J,3)
            OUT(J,4)=Y(3)+OUT(J,4)
C
            OUT1(J,1)=TOUT+OUT1(J,1)
            OUT1(J,2)=UNEW(1)+OUT1(J,2)
            OUT1(J,3)=UNEW(2)+OUT1(J,3)
            OUT1(J,4)=UNEW(3)+OUT1(J,4)
            OUT1(J,5)=UNEW(4)+OUT1(J,5)
C
            UOUT2(J,1)=TOUT+UOUT2(J,1)
            UOUT2(J,2)=X(5)+UOUT2(J,2)
            UOUT2(J,3)=X(7)+UOUT2(J,3)
            UOUT2(J,4)=X(9)+UOUT2(J,4)
C
            UOUT3(J,1)=TOUT+UOUT3(J,1)
            UOUT3(J,2)=X(1)+UOUT3(J,2)
            UOUT3(J,3)=X(2)+UOUT3(J,3)
            UOUT3(J,4)=X(3)+UOUT3(J,4)
            UOUT3(J,5)=X(4)+UOUT3(J,5)
         END IF
         TOUT=TOUT+TSAMP
      IF(IIFLAG.EQ.1)THEN
      CALL GGNML(DSEED,NR,V)
      V(1)=R(1,1)*V(1)
      V(2)=R(2,2)*V(2)
      V(3)=R(3,3)*V(3)
      IF(NN.EQ.1)THEN
      Z(1)=X(L)+V(1)
      Z(2)=X(M)+V(2)
      Z(3)=X(N)+V(3)
      ELSE
      Z(1)=Y(L)+V(1)
      Z(2)=Y(M)+V(2)
      Z(3)=Y(N)+V(3)
      END IF
      CALL KFILT(Z)
      IF(JMFLAG.EQ.1) THEN
      DO 915 J=1,4
      XTEMP(J)=XHP(J)
      DO 915 KK=5,12
      XTEMP(KK)=0.0
  915 CONTINUE
      ELSE
      DO 980 J=1,4
      XTEMP(J)=XHM(J)
      DO 980 KK=5,12
      XTEMP(KK)=0.0
980   CONTINUE
      END IF
         CALL GCSTAR(XTEMP,NFLAG)
         DO 250 J=1,12
            XOLD(J)=XTEMP(J)
 250     CONTINUE
         DO 260 J=1,4
            UOLD(J)=UNEW(J)
 260     CONTINUE
         DO 262 J=1,4
            UCOLD(J)=UCMD(J)
            XMOLD(J)=XM(J)
 262     CONTINUE
      ELSE
      CALL GCSTAR(X,NFLAG)
      DO 1060 JM=1,12
      XOLD(JI )=X(JM)
1060  CONTINUE
      DO 1070 JM=1,4
      XMOLD(JM)=XM(JM)
      UCOLD(JM)=UCMD(JM)
      UOLD(JM)=UNEW(JM)
1070  CONTINUE
      END IF
            CALL ODE(FSTOL,12,X,T,TOUT,RELERR,ABSERR,IFLAG,WORK,IWORK)
         T=TOUT
         IF(IFLAG.NE.2) THEN
            PRINT'(" IFLAG = ",I2)',IFLAG
         ELSE
            IFLAG=-2
         END IF
 300  CONTINUE
```

```
780   CONTINUE
      IF(IIFLAG.EQ.0)GO TO 1310
      DO 755 I=1,51
      OUT(I,1)=OUT(I,1)/XITER
      OUT(I,2)=OUT(I,2)/XITER
      OUT(I,3)=OUT(I,3)/XITER
      OUT(I,4)=OUT(I,4)/XITER
      OUT1(I,1)=OUT1(I,1)/XITER
      OUT1(I,2)=OUT1(I,2)/XITER
      OUT1(I,3)=OUT1(I,3)/XITER
      OUT1(I,4)=OUT1(I,4)/XITER
      OUT1(I,5)=OUT1(I,5)/XITER
      UOUT2(I,1)=UOUT2(I,1)/XITER
      UOUT2(I,2)=UOUT2(I,2)/XITER
      UOUT2(I,3)=UOUT2(I,3)/XITER
      UOUT2(I,4)=UOUT2(I,4)/XITER
      UOUT3(I,1)=UOUT3(I,1)/XITER
      UOUT3(I,2)=UOUT3(I,2)/XITER
      UOUT3(I,3)=UOUT3(I,3)/XITER
      UOUT3(I,4)=UOUT3(I,4)/XITER
      UOUT3(I,5)=UOUT3(I,5)/XITER
755   CONTINUE
1310  CONTINUE
999   PRINT*,' 1=OUTPUT VARIABLES  2=CONTROL INPUTS '
      PRINT*,' 3=CONTROL DEFLECTIONS  4=SYSTEM STATES'
      READ*, MMFLAG
      GO TO (1000,1100,1200,1300) MMFLAG
1000  CALL SETPLT(OUT,51,5,PLTVEC)
      PRINT*,' +---------------ENTER TITLE FOR PLOT------------+'
      PRINT*
      READ(*,'(A)')TITLE
      CALL PLOTLP(PLTVEC,51,3,-1,1,0,TITLE)
      GO TO 1400
1100  CALL SETPLT(OUT1,51,5,PLTVEC)
      PRINT*,' +---------------ENTER TITLE FOR PLOT------------+'
      PRINT*
      READ(*,'(A)')TITLE
      CALL PLOTLP(PLTVEC,51,4,-1,1,0,TITLE)
      GO TO 1400
1200  CALL SETPLT(UOUT2,51,5,PLTVEC)
      PRINT*,' +---------------ENTER TITLE FOR PLOT------------+'
      PRINT*
      READ(*,'(A)')TITLE
      CALL PLOTLP(PLTVEC,51,3,-1,1,0,TITLE)
      GO TO 1400
1300  CALL SETPLT(UOUT3,51,5,PLTVEC)
      PRINT*,' +---------------ENTER TITLE FOR PLOT------------+'
      PRINT*
      READ(*,'(A)')TITLE
      CALL PLOTLP(PLTVEC,51,4,-1,1,0,TITLE)
1400  PRINT*,' MORE OUTPUT PLOTS?'
      READ(*,'(A)')ANSW
      IF(ANSW.NE.'Y'.AND.ANSW.NE.'N') GO TO 1500
      IF(ANSW.EQ.'Y') GO TO 999
1500  CONTINUE
```

```
 3214 PRINT*,'INPUT NAME FOR CALCOMP PLOT OF OUTPUT'
      READ(*,'(A)')PLOT
      OPEN(5,FILE=PLOT,STATUS='NEW',FORM='FORMATTED',ERR=3214)
      WRITE(5,FMT='(4E20.5)')((OUT(J,I),I=1,4),J=1,51)
      ENDFILE(5)
      REWIND(5)
      CLOSE(5)
 3215  PRINT*,'INPUT NAME FOR CALCOMP PLOT OF CTRL. DEF.'
       READ(*,'(A)')PLOT
       OPEN(6,FILE=PLOT,STATUS='NEW',FORM='FORMATTED',ERR=3215)
       WRITE(6,FMT='(4E20.5)')((UOUT2(J,I),I=1,4),J=1,51)
       ENDFILE(6)
       REWIND(6)
       CLOSE(6)
 525  PRINT*,'CHANGE MATRICES? Y/N: '
      READ(*,'(A)')ANSW
      IF(ANSW.NE.'Y'.AND.ANSW.NE.'N') GO TO 525
      IF(ANSW.EQ.'Y') GO TO 142
 530  PRINT*,'MORE RUNS WITH NEW MODEL? Y/N: '
      READ(*,'(A)')ANSW
      IF(ANSW.NE.'Y'.AND.ANSW.NE.'N') GO TO 530
      IF(ANSW.EQ.'Y') GO TO 20
      END
C
C END PROGRAM ODEF15 -----------------------------------------
C
C
       SUBROUTINE FSTOL(T,X,DX)
C
C
C
C***********************************************************************
C
C  THIS IS A SET OF FIRST ORDER ORDINARY DIFFERENTIAL EQUATIONS THAT
C  DEFINE THE THE DYNAMICS OF THE STOL F-15 AIRCRAFT.
C  ACTUATOR DYNAMICS ARE INCLUDED AS ENTERED IN THE 12 X 12
C  A MATRIX WHICH HAS BEEN ENTERED AT THE ONSET OF THE PROGRAM
C  IT IS ASSUMED THAT SECOND ORDER ACTUATORS ARE ASSOCIATED WITH
C  THE STABILATOR AND CANARD, AND FIRST ORDER WITH THE NOZZLE.
C  NOTE THAT A NON-LINEARITY IS INTRODUCED INTO THE MODEL BY THE
C  CONTROL OF BOTH THRUST AND NOZZLE DEFLECTION.
C
C***********************************************************************
C
C
      REAL T,X(12),DX(12),BNL(3,3)
      COMMON/MATRIX/UO(2),A(12,12),C(4,12),KX(4.12),KZ(4,4),KXM(4,4)
     1,KXU(4,4),PHI(4,4),PHINT(4,4),IM(4,4),B(3,3),EV(2),KFLAG,MM

      COMMON/CONTRL/UNEW(4),UOLD(4),UCMD(4),UCOLD(4)
     1,XOLD(12),XMOLD(4),XM(4),MFLAG,EVA(2)
C
C
      DO 444 I=1,3
        DO 444 J=1,3
```

AD-A164 108   AN LQG UP-AND AWAY FLIGHT CONTROL DESIGN FOR THE STOL   3/3
F-15 AIRCRAFT(U) AIR FORCE INST OF TECH
WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING
UNCLASSIFIED   R A HOUSTON DEC 85 AFIT/GE/ENG/85D-21   F/G 1/4   NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

```
          BNL(I,J)=0.
 444    CONTINUE
C  SET THE SIGN TO ACCOUNT FOR CONTROL SURFACES PASSING
C  THROUGH A ZERO ANGLE OF ATTACK RELATIVE TO THE A/C
C
C
      EVA(1)=EV(1)+X(3)
      EVA(2)=EV(2)+X(3)
      DO 5000 II=1,3
        BNL(II,3)=B(II,3)
          DO 5000 JJ=1,2
            BNL(II,JJ)=B(II,JJ)
      IF(EVA(JJ).GE.0)THEN
        IF((UO(JJ)+EVA(JJ)).LT.0) THEN
          BNL(1,JJ)=-B(1,JJ)
        ENDIF
      ELSE
        IF((UO(JJ)+EVA(JJ)).GT.0) THEN
           BNL(1,JJ)=-B(1,JJ)
      ENDIF
      ENDIF
5000     CONTINUE
      IF(MM.EQ.1)THEN
      DX(1)=A(1,1)*X(1)+A(1,2)*X(2)+A(1,3)*X(3)+A(1,4)*X(4)
     1+BNL(1,1)*UNEW(1)+BNL(1,2)*UNEW(2)+BNL(1,3)*UNEW(3)
      DX(2)=A(2,1)*X(1)+A(2,2)*X(2)+A(2,3)*X(3)+A(2,4)*X(4)
     1+BNL(2,1)*UNEW(1)+BNL(2,2)*UNEW(2)
      DX(3)=A(3,1)*X(1)+A(3,2)*X(2)+A(3,3)*X(3)+A(3,4)*X(4)
     1+BNL(3,1)*UNEW(1)+BNL(3,2)*UNEW(2)
       DX(4)=X(2)
      DX(5)=0.
      DX(6)=0.
      DX(7)=0.
      DX(8)=0.
      DX(9)=0.
      DX(10)=0.
      DX(11)=0.
      DX(12)=0.
       UO(1)=UNEW(1)
       UO(2)=UNEW(2)
      X(5)=UNEW(1)
       X(7)=UNEW(2)
      X(9)=UNEW(3)
C
      ELSE
        DX(1)=A(1,1)*X(1)+A(1,2)*X(2)+A(1,3)*X(3)+A(1,4)*X(4)
     1+BNL(1,1)*X(5)+BNL(1,2)*X(7)+BNL(1,3)*X(9)
        DX(2)=A(2,1)*X(1)+A(2,2)*X(2)+A(2,3)*X(3)+A(2,4)*X(4)
     1+BNL(2,1)*X(5)+BNL(2,2)*X(7)
        DX(3)=A(3,1)*X(1)+A(3,2)*X(2)+A(3,3)*X(3)+A(3,4)*X(4)
     1+BNL(3,1)*X(5)+BNL(3,2)*X(7)
        DX(4)=X(2)
        DX(5)=X(6)
        DX(6)=-8356.*X(5)-303.*X(6)+8356.*UNEW(1)
        DX(7)=X(8)
        DX(8)=-8356.*X(7)-303.*X(8)+8356.*UNEW(2)
        DX(9)=-20.*X(9)+20.*UNEW(3)
        DX(10)=0.
        DX(11)=0.
        DX(12)=0.
       UO(1)=X(5)
       UO(2)=X(7)
      END IF
C
      IF(MFLAG.EQ.1)THEN
         IF(X(5).GE..262.AND.DX(5).GT.0.0)DX(5)=0.0
         IF(X(5).LE.-.611.AND.DX(5).LT.0.0)DX(5)=0.0
         IF(X(6).GE..401.AND.DX(6).GT.0.0)DX(6)=0.0
         IF(X(6).LE.-.401.AND.DX(6).LT.0.0)DX(6)=0.0
         IF(X(7).GE..262.AND.DX(7).GT.0.0)DX(7)=0.0
         IF(X(7).LE.-.506.AND.DX(7).LT.0.0)DX(7)=0.0
      IF(X(8).GE..803.AND.DX(8).GT.0.0)DX(8)=0.0
         IF(X(8).LE.-.803.AND.DX(8).LT.0.0)DX(8)=0.0
      END IF
       RETURN
       END
C
C END PUBFROGRAM FSTOL-----------------------------------------
C
C
      SUBROUTINE GCSTAR(X,NFLAG)
C
C*********************************************************************
C
C SUBROUTINE TO CALCULATE THE CONTROLS AT EACH SAMPLE TIME.
C ANTI-WINDUP COMPENSATED IF NFLAG=1.
C
C*********************************************************************
C
      REAL X(12),DEL(12),DEL2(12)
      INTEGER NFLAG
      COMMON/MATRIX/UO(2),A(12,12),C(4,12),KX(4,12),KZ(4,4),KXM(4,4),
     1 KXU(4,4),PHI(4,4),PHINT(4,4),CM(4,4),B(3,3),EV(2),KFLAG,MM
      COMMON/CONTRL/UNEW(4),UOLD(4),UCMD(4),UCOLD(4),XOLD(12),
     1 XMOLD(4),XM(4),MFLAG,EVA(2)
      CALL MATML(PHI,XMOLD,XM,4,4,1)
      CALL MATML(PHINT,UCMD,DEL,4,4,1)
      CALL MATAD(XM,DEL,XM,4,1)
      CALL MATSB(X,XOLD,DEL,12,1)
      CALL MATML(KX,DEL,DEL2,4,12,1)
      CALL MATSB(UOLD,DEL2,UNEW,4,1)
      CALL MATSB(XM,XMOLD,DEL,4,1)
      CALL MATML(KXM,DEL,DEL2,4,4,1)
      CALL MATAD(UNEW,DEL2,UNEW,4,1)
      CALL MATSB(UCMD,UCOLD,DEL,4,1)
      CALL MATML(KXU,DEL,DEL2,4,4,1)
      CALL MATAD(UNEW,DEL2,UNEW,4,1)
      CALL MATML(CM,XMOLD,DEL,4,4,1)
      CALL MATML(C,XOLD,DEL2,4,12,1)
      CALL MATSB(DEL,DEL2,DEL,4,1)
      CALL MATML(KZ,DEL,DEL2,4,4,1)
      CALL MATAD(UNEW,DEL2,UNEW,4,1)
      IF(NFLAG.EQ.1) THEN
      IF(UNEW(1).GT..49-.87*X(5))UNEW(1)=(.49-.87*X(5))*1.
      IF(UNEW(1).LT.-1.14-.87*X(5))UNEW(1)=(-1.14-.87*X(5))*1.
      IF(UNEW(2).GT..49-.87*X(7))UNEW(2)=.49-.87*X(7)
      IF(UNEW(2).LT.-.96-.87*X(7))UNEW(2)=-.96-.87*X(7)
      IF(UNEW(1).GT..706+X(5))UNEW(1)=.706+X(5)
      IF(UNEW(1).LT.-.706+X(5))UNEW(1)=-.706+X(5)
      IF(UNEW(2).GT.1.522+X(7))UNEW(2)=1.522+X(7)
      IF(UNEW(2).LT.-1.522+X(7))UNEW(2)=-1.522+X(7)
      END IF
      RETURN
      END
C
C END SUBROUTINE GCSTAR ------------------------------------------------
C
C
      SUBROUTINE RPOUT(A,M,N)
C
C**********************************************************************
C
C THIS ROUTINE PRINTS OUT A REAL MATRIX A
C
C***********************************************************************
C
      REAL A(M,N)
      INTEGER I,J,N,M
      DO 200 I=1,M
         PRINT'(" ",5(E11.4,3X))',(A(I,J),J=1,N)
         PRINT*
 200  CONTINUE
      END
C
C END SUBROUTINE RPOUT -------------------------------------------------
C
C
      SUBROUTINE SETPLT(A,N,M,X)
C
C**********************************************************************
C
C THIS ROUTINE CONVERTS A REAL MATRIX OF DIMENSION N BY M INTO A
C VECTOR THAT IS COMPATIBLE WITH R.M. FLOYD'S PRINTER PLOTTING
C ROUTINE, PLOTLP.  THE INPUT MATRIX IS A.
C N= ROW DIMENSION OF A, THE NUMBER OF POINTS TO BE PLOTTED
C M= COLUMN DIMENSION OF A, THE NUMBER OF FUNCTIONS TO BE PLOTTED +1
C X= THE PLOTTING VECTOR, DIMENSION N*M
C
C**********************************************************************
C
      REAL A(N,M),X(N*M)
      INTEGER N,M,I,J
      DO 100 J=1,M
         DO 100 I=1,N
            X(I+(J-1)*N)=A(I,J)
 100  CONTINUE
      END
C
C END SUBROUTINE SETPLT -----------------------------------------------
C
C
      SUBROUTINE PLOTLP(A,N,M,IPSC,ISCL,LPTERM,TITLE)
C
C**********************************************************************
C
C THIS ROUTINE WAS ADAPTED FROM R.M. FLOYD'S THESIS TO PRODUCE
C PRINTER PLOTS OF COMPUTED RESULTS.
C A= VECTOR OF DATA, CONVERTED FROM MATRIX FORM BY SUBROUTINE SETPLT
C N= NUMBER OF POINTS (INDEPENDENT VARIABLE) TO BE PLOTTED
C M= NUMBER OF FUNCTIONS (DEPENDENT VARIABLES) TO BE PLOTTED
C IPSC = -1-->ALL VARIABLES SCALED TOGETHER (1 PLOT)
C      =  0-->SCALED TOGETHER AND SEPARATELY (2 PLOTS)
C      = +1-->SCALED SEPARATELY (1 PLOT)
C ISCL =  0-->PLOT OVER EXACT RANGE OF VARIABLE
C        +1-->PLOT WITH EVEN SCALING
C LPTERM = 0-->PLOT 50 CHARACTERS WIDE
C         +1-->PLOT 100 CHARACTERS WIDE
C TITLE = MAX OF 50 CHARACTERS, TYPE CHARACTER
C
C**********************************************************************
C
      REAL YSCAL(6),YMIN(6),YPR(11),RISPAC,RMIN,RMAX,YL,YH,XPR,A(*)
      REAL SCAL
      INTEGER IBLNK(6),IPSC,ISCL,LPTERM,IPAPER,ISPAC,IPRTI,ISC,J,IC,IX
      INTEGER IL,JP,ITEMP,M1,M2,M,N,ICO,I
      CHARACTER TITLE*50
      CHARACTER*1 BLANK,PLUS,COLON,GRID,SYMBOL(6),OUT(101)
      DATA BLANK,PLUS,COLON,SYMBOL(1),SYMBOL(2)/' ','+',':','1','2'/
      DATA SYMBOL(3),SYMBOL(4),SYMBOL(5),SYMBOL(6)/'3','4','5','6'/
      IPAPER=5*(1+LPTERM)
      ISPAC=10*IPAPER
      RISPAC=REAL(ISPAC)
      ISPAC=ISPAC+1
      IPRTI=IPAPER+1
      RMIN=A(N+1)
      RMAX=RMIN
 25   DO 41 ISC=1,M
         M1=ISC*N+1
         YL=A(M1)
         YH=YL
         M2=N*(ISC+1)
         DO 40 J=M1,M2
            IF(A(J).LT.YL)THEN
               YL=A(J)
            END IF
            IF(A(J).GT.YH)THEN
               YH=A(J)
            END IF
 40      CONTINUE
         IF(YL.LT.RMIN)THEN
```

```
            RMIN=YL
         END IF
         IF(YH.GT.RMAX)THEN
            RMAX=YH
         END IF
         IF(IPSC.GE.0)THEN
            CALL VARSCL(YL,YH,YSCAL(ISC),RISPAC,ISCL)
         END IF
         YMIN(ISC)=YL
  41  CONTINUE
      IF(IPSC.LE.0) THEN
         CALL VARSCL(RMIN,RMAX,SCAL,RISPAC,ISCL)
      END IF
      IC=2-IABS(IPSC)
      DO 42 IX=1,ISPAC
         OUT(IX)=BLANK
  42  CONTINUE
      DO 100,ICO=1,IC
         PRINT'("1",11X,A50)',TITLE
         WRITE(4,'(11X,A50)')TITLE
         WRITE(4,'(A1)')BLANK
         PRINT*
         DO 60 I=1,N
            XPR=A(I)
            IF(MOD(I,10).EQ.0)THEN
               GRID=COLON
            ELSE
               GRID=BLANK
            END IF
            DO 44 IX=2,ISPAC,2
               OUT(IX)=GRID
  44        CONTINUE
            DO 46 IX=1,ISPAC,10
               OUT(IX)=PLUS
  46        CONTINUE
            DO 55 J=1,M
               IL=I+J*N
               IF(IPSC.EQ.-1)THEN
                 JP=INT((A(IL)-RMIN)/SCAL)+1
               ELSE IF(IPSC.EQ.0)THEN
                  IPSCT=IPSC+ICO
                  IF(IPSCT.EQ.2)THEN
                     JP=INT((A(IL)-YMIN(J))/YSCAL(J))+1
                  ELSE
                     JP=INT((A(IL)-RMIN)/SCAL)+1
                  END IF
               ELSE
                     JP=INT((A(IL)-YMIN(J))/YSCAL(J))+1
               END IF
  50           OUT(JP)=SYMBOL(J)
               IBLNK(J)=JP
  55        CONTINUE
            PRINT'(" ",F11.4,6X,101A1)',XPR,(OUT(IX),IX=1,ISPAC)
            WRITE(4,'(F11.4,6X,101A1)')XPR,(OUT(IX),IX=1,ISPAC)
            DO 59 J=1,M
```

```
               ITEMP=IBLNK(J)
               OUT(ITEMP)=BLANK
 59         CONTINUE
 60      CONTINUE
         IF(IPSC.NE.1)THEN
            IF(IPSCT.NE.2)THEN
               YPR(1)=RMIN
               DO 70 I=1,IPAPER
                  YPR(I+1)=YPR(I)+10.*SCAL
 70            CONTINUE
               PRINT'("0     SCALE    ",11E10.3)',(YPR(I),I=1,IPRTI)
               WRITE(4,'(A1)')BLANK
               WRITE(4,'("     SCALE    ",11E10.3)')(YPR(I),I=1,IPRTI)
               WRITE(4,'(A1)')BLANK
               WRITE(4,'(A1)')BLANK
            END IF
         END IF
         IF(IPSC.EQ.1.OR.IPSCT.EQ.2)THEN
            DO 76 ISC=1,M
               YPR(1)=YMIN(ISC)
               DO 74 I=1,IPAPER
                  YPR(I+1)=YPR(I)+10.*YSCAL(ISC)
 74            CONTINUE
               PRINT'("0     SCALE ",A1,1X,11E10.3)',SYMBOL(ISC),(YPR(IX
)
     1,IX=1,IPRTI)
               WRITE(4,'(A1)')BLANK
               WRITE(4,'("     SCALE ",A1,1X,11E10.3)')SYMBOL(ISC),
     1(YPR(IX),IX=1,IPRTI)
 76         CONTINUE
         END IF
         DO 90 ISC=1,56-N
            WRITE(4,'(A1)')BLANK
 90      CONTINUE
 100  CONTINUE
      PRINT'("1")'
      END
C
C END SUBROUTINE PLOTLP ----------------------------------------------
C
C
      SUBROUTINE VARSCL(XMIN,XMAX,SCALE,RSPACE,ISCL)
C
C*********************************************************************
C
C THIS IS A SCALING ROUTINE THAT SUPPORTS PLOTLP
C ADAPTED FROM R.M. FLOYD'S THESIS
C
C*********************************************************************
C
      REAL XMIN,XMAX,SCALE,RSPACE,EXP,XMINT,XMAXT
      INTEGER ISCL,ISCAL
      IF(XMAX.EQ.XMIN)THEN
         XMIN=.9*XMIN-10.
      END IF
      SCALE=XMAX-XMIN
      IF(ISCL.NE.0)THEN
         EXP=INT(100.+LOG10(SCALE))-100.
         FACTOR=10.**(1.-EXP)
         XMINT=XMIN*FACTOR
         XMAXT=XMAX*FACTOR
         IF(XMAXT.GE.0.)THEN
            XMAXT=XMAXT+.9
         END IF
         IF(XMINT.LE.0.)THEN
            XMINT=XMINT-.9
         END IF
         XMINT=AINT(XMINT)
         ISCAL=XMAXT-XMINT
         IF(MOD(ISCAL,5).NE.0)THEN
            ISCAL=ISCAL+5-MOD(ISCAL,5)
         END IF
         FACTOR=10.**(EXP-1.)
         XMIN=XMINT*FACTOR
         SCALE=FACTOR*REAL(ISCAL)
      END IF
      SCALE=SCALE/RSPACE
      END
C
C END SUBROUTINE VARSCL ----------------------------------------------
C
C
      SUBROUTINE EDIT(EDMAT,M,N)
C
C*********************************************************************
C
C THIS ROUTINE ALLOWS THE USER TO EDIT AN M BY N MATRIX EDMAT
C
C*********************************************************************
C
      REAL EL,EDMAT(M,N)
      INTEGER M,N,I,J
      CHARACTER ANSW*1
 10   PRINT*,'LIST CURRENT VALUES? Y/N: '
      READ(*,'(A)')ANSW
      IF(ANSW.NE.'Y'.AND.ANSW.NE.'N') GO TO 10
      IF(ANSW.EQ.'Y') CALL RFOUT(EDMAT,M,N)
      PRINT*,'ENTER 0,0,0 <CR> WHEN ALL CHANGES HAVE BEEN MADE'
 100  PRINT*,'ENTER ROW #, COLUMN #, AND MATRIX ELEMENT: '
 110  READ*,I,J,EL
      IF(I.GT.0.AND.I.LE.M.AND.J.GT.0.AND.J.LE.N)THEN
         EDMAT(I,J)=EL
         GO TO 110
      ELSE IF(I.EQ.0.AND.J.EQ.0)THEN
 150  PRINT*,'LIST MODIFIED MATRIX? Y/N: '
      READ(*,'(A)')ANSW
      IF(ANSW.NE.'Y'.AND.ANSW.NE.'N') GO TO 150
      IF(ANSW.EQ.'Y') CALL RFOUT(EDMAT,M,N)
 200     PRINT*,'ANY MORE CHANGES TO THIS MATRIX? Y/N: '
         READ(*,'(A)')ANSW
         IF(ANSW.NE.'Y'.AND.ANSW.NE.'N')GO TO 200
         IF(ANSW.EQ.'Y')GO TO 100
         IF(ANSW.EQ.'N')RETURN
      ELSE
         PRINT*,'SUBSCRIPT OUT OF RANGE'
         GO TO 100
      END IF
      END
C
C END SUBROUTINE EDIT-------------------------------------------------
C
C
      SUBROUTINE MATML(A,B,C,L,M,N)
C
C********************************************************************
C
C THIS ROUTINE WILL MULTIPLY TWO REAL MATRICES
C A=AN L BY M MATRIX
C B=AN M BY N MATRIX
C C=THE L BY N PRODUCT OF A AND B
C NOTE: ACTUAL ARGUMENT C MUST DIFFER FROM A AND B
C
C********************************************************************
C
      REAL A(L,M),B(M,N),C(L,N)
      INTEGER I,J,K,L,M,N
      DO 100 I=1,L
         DO 100 J=1,N
            C(I,J)=0.0
 100  CONTINUE
      DO 200 I=1,L
         DO 200 J=1,N
            DO 200 K=1,M
               C(I,J)=C(I,J)+A(I,K)*B(K,J)
 200  CONTINUE
      END
C
C END SUBROUTINE MATML ---------------------------------------------
C
C
      SUBROUTINE MATAD(A,B,C,L,M)
C
C********************************************************************
C
C THIS ROUTINE ADDS TWO REAL MATRICES OF DIMENSION L BY M
C A AND B ARE THE INPUTS, C IS THE SUM
C
C*********************************************************************
C
      REAL A(L,M),B(L,M),C(L,M)
      INTEGER I,J,L,M
      DO 100 I=1,L
         DO 100 J=1,M
            C(I,J)=A(I,J)+B(I,J)
 100  CONTINUE
      END
```

```
C
C END SUBROUTINE MATAD -------------------------------------------
C
C
      SUBROUTINE MATSB(A,B,C,L,M)
C
C**********************************************************************
C
C THIS ROUTINE SUBTRACTS REAL MATRIX B FROM REAL MATRIX A
C DIFFERENCE IS RETURNED IN REAL MATRIX C.
C ALL THREE MATRICES ARE OF DIMENSION L BY M
C
C**********************************************************************
C
      REAL A(L,M),B(L,M),C(L,M)
      INTEGER I,J,L,M
      DO 100 I=1,L
         DO 100 J=1,M
            C(I,J)=A(I,J)-B(I,J)
 100  CONTINUE
      END
C
C END SUBROUTINE MATSB -------------------------------------------
C
C
      SUBROUTINE SMUL(A,B,C,L,M)
C
C**********************************************************************
C
C THIS ROUTINE MULTIPLIES A REAL MATRIX BY A REAL SCALAR
C A= THE SCALAR
C B= THE MATRIX
C C= THE PRODUCT
C B AND C ARE OF DIMENSION L BY M
C
C**********************************************************************
C
      REAL A,B(L,M),C(L,M)
      INTEGER I,J,L,M
      DO 100 I=1,L
         DO 100 J=1,M
            C(I,J)=A*B(I,J)
 100  CONTINUE
      END
C
C END SUBROUTINE SMUL --------------------------------------------
C
C
      SUBROUTINE COPYMT(A,B,N,M)
C
C**********************************************************************
C
C THIS ROUTINE COPIES A REAL MATRIX A INTO REAL MATRIX B.
C BOTH MATRICES ARE OF DIMENSION N BY M.
C
```

```
C*********************************************************************
C
      REAL A(N,M),B(N,M)
      INTEGER I,J,N,M
      DO 100 I=1,N
         DO 100 J=1,M
            B(I,J)=A(I,J)
 100  CONTINUE
      END
C
C END SUBROUTINE COPYMT ----------------------------------------------
C
C
      SUBROUTINE DSCRT(A,N,TSAMP,PHI,PHINT,M,TP,TIDENT,CWORK)
C
C**********************************************************************
C
C THIS ROUTINE APPROXIMATES THE STATE TRANSITION MATRIX AND ITS
C INTEGRAL FOR A TIME INVARIANT LINEAR SYSTEM AS A MATRIX EXPONENTIAL
C OVER A SMALL SAMPLE PERIOD.  RESULTS RETURNED IN REAL MATRICES.
C A= SYSTEM DYNAMICS MATRIX, TYPE REAL
C N= STATE DIMENSION
C TSAMP= SAMPLING PERIOD
C PHI= STATE TRANSITION MATRIX, TYPE REAL
C PHINT= APPROXIMATE INTEGRAL OF PHI, TYPE REAL
C M= NUMBER OF TERMS USED IN EXPONENTIAL EXPANSION
C TP, TIDENT AND CWORK ARE DUMMY ARRAYS
C
C*********************************************************************
C
      REAL A(N,N),PHINT(N,N),PHI(N,N),TIDENT(N,N),TP(N,N)
      REAL CWORK(N,N)
      REAL TSAMP,RIJ
      INTEGER I,J,M,N
      DO 200 I=1,N
         DO 100 J=1,N
            TIDENT(I,J)=0.0
 100     CONTINUE
         TIDENT(I,I)=1.0
 200  CONTINUE
      CALL SMUL(TSAMP,TIDENT,PHINT,N,N)
      CALL COPYMT(PHINT,TP,N,N)
      CALL SMUL(TSAMP,A,PHI,N,N)
      DO 300 I=1,M
         CALL MATML(TP,PHI,CWORK,N,N,N)
         CALL COPYMT(CWORK,TP,N,N)
         RIJ=1.0/REAL(I+1)
         CALL SMUL(RIJ,TP,TP,N,N)
         CALL MATAD(PHINT,TP,PHINT,N,N)
 300  CONTINUE
      CALL MATML(A,PHINT,TP,N,N,N)
      CALL MATAD(TIDENT,TP,PHI,N,N)
C
C END SUBROUTINE DSCRT ----------------------------------------------
C
```

```
      END
       SUBROUTINE KFILT(Z)
C
C******************************************************************
C
C SUBROUTINE TO INCORPORATE THE KALMAN FILTER INTO THE LOOP FOR
C NON-LINEAR PERFORMANCE ANALYSIS.
C
C****************************************************************
C
      COMMON/CONTRL/UNEW(4),UOLD(4),UCMD(4),UCOLD(4),XOLD(12),
     1 XMOLD(4),XM(4),MFLAG,EVA(2),
     1 H(3,4),PHIX(4,4),BD(4,3),QD(4,4),R(3,3),XHP(4),XHM(4),K(4,3)
      REAL AWORK(4,1),BWORK(4,1),CWORK(3,1),DWORK(3,1),EWORK(4,1)
      REAL Z(3)
      CALL MATML(PHIX,XHP,AWORK,4,4,1)
      CALL MATML(BD,UNEW,BWORK,4,3,1)
      CALL MATAD(AWORK,BWORK,XHM,4,1)
      CALL MATML(H,XHM,CWORK,3,4,1)
      CALL MATSB(Z,CWRK,DWORK,3,1)
      CALL MATML(K,DWORK,EWORK,4,3,1)
      CALL MATAD(XHM,EWORK,XHP,4,1)
C
C END SUBROUTINE KFILT----------------------------------------------
C
      RETURN
      END
```

Appendix D: Derivation of Nonlinear Thrust/Nozzle Model

The following is a derivation of the STOL F-15 aircraft model using both thrust (throttle) and nozzle deflection as inputs to the system. As will be seen in this derivation, the simultaneous control of these two quantities introduces a nonlinearity into the system model. Efforts were made to design a constant-gain controller for this system; however, instability proved to be a severe problem which could not be overcome. To allow the use of both throttle and thrust vectoring nozzles in flight requires that either a proper linearization of this model be derived or that extensions to the constant-gain CGT/PI/KF be introduced to compensate for this nonlinearity.

For Figure D.1, the X direction force, Z direction force, and longitudinal moment equations become:

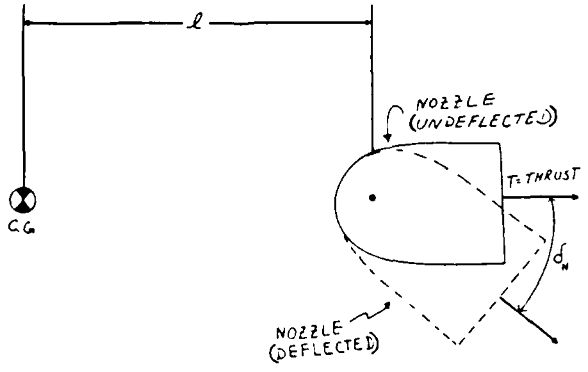


Fig. D.1. Diagram of Nozzle Deflection

$$f_{T_X} = \frac{T}{m} \cos \delta_N \simeq \frac{T}{m} \qquad \text{(D-1)}$$

$$f_{T_Z} = T \sin \delta_N \simeq T\delta_N \qquad \text{(D-2)}$$

$$M_T = \ell f_{T_Z} \cong \ell T \delta_N \qquad \text{(D-3)}$$

respectively, where m is the aircraft mass. Also note that small angle approximations are made in Equations (D-1) and (D-2).

Incorporating (D-1) and (D-2) into the standard perturbation equations of motion (in dimensional derivative form) (19) yields:

$$\dot{u} = -g\theta \cos \theta_o + X_u u + X_\alpha \alpha + X_{\delta_S} \delta_S + \frac{T}{m} + X_{\delta_C} \delta_C \qquad \text{(D-4)}$$

$$\dot{w} - u_o q = -g\theta \sin \theta_o + Z_u u + Z_\alpha \alpha + Z_{\dot{\alpha}} \dot{\alpha} + Z_q q + Z_{\delta_C} \delta_C + Z_{\delta_S} \delta_S + \frac{T}{m} \delta_N \qquad \text{(D-5)}$$

$$\dot{q} = M_u u + M_{T_u} u + M_\alpha \alpha + M_{\dot{\alpha}} \dot{\alpha} + M_q q + M_{\delta_S} \delta_S + \frac{\ell T}{I_{yy}} \delta_N \qquad \text{(D-6)}$$

Now, taking the Laplace transform of (D-4) to (D-6), again invoking small angle approximations, and using the approximation that the acceleration in the z direction is equal to the forward velocity times the derivative of the angle of attack:

$$\dot{w} \cong u_o\dot{\alpha} \tag{D-7}$$

Equations (D-4) to (D-6) become:

$$su = -g\theta + X_u u + X_{T_u} u + X_\alpha \alpha + X_{\delta_S} \delta_S + X_{\delta_C} \delta_C + \frac{T}{m} \tag{D-8}$$

$$s\alpha u_o - u_o q = -g\theta\theta_o + Z_u u + Z_\alpha \alpha + sZ_{\dot{\alpha}}\dot{\alpha} + Z_q q + Z_{\delta_S}\delta_S + Z_{\delta_C}\delta_C + \frac{T\delta_N}{m} \tag{D-9}$$

$$sq = M_u u + M_{T_u} u + M_\alpha \alpha + M_{T_\alpha} \alpha + sM_{\dot{\alpha}}\dot{\alpha} + M_q q + M_{\delta_S}\delta_S + M_{\delta_C}\delta_C + \frac{\ell_T \delta_N}{I_{yy}} \tag{D-10}$$

Upon rearranging (D-9), it becomes:

$$s\alpha = \frac{u_o q}{(u_o - Z_{\dot{\alpha}})} - \frac{g\theta\theta_o}{(u_o - Z_{\dot{\alpha}})} + \frac{Z_u u}{(u_o - Z_{\dot{\alpha}})} + \frac{Z_\alpha \alpha}{(u_o - Z_{\dot{\alpha}})} + \frac{Z_q q}{(u_o - Z_{\dot{\alpha}})} + \frac{Z_{\delta_C}\delta_C}{(u_o - Z_{\dot{\alpha}})} + \frac{Z_{\delta_S}\delta_S}{(u_o - Z_{\dot{\alpha}})} + \frac{T\delta_N}{m(u_o - Z_{\dot{\alpha}})} \tag{D-11}$$

In state space form, the original aircraft equations of motion are of the form:

$$\begin{bmatrix} \dot{u} \\ \dot{q} \\ \dot{\alpha} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ A_{31} & A_{32} & A_{33} & A_{34} \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} u \\ q \\ \alpha \\ \theta \end{bmatrix} + \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ B_{31} & B_{32} & B_{33} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_C \\ \delta_S \\ \delta_N \end{bmatrix} \qquad \text{(D-12)}$$

Using the following first-order lag model for throttle dynamics (20)

$$\frac{I}{\delta_T} = \frac{K_T}{s + \frac{1}{\tau_T}} \qquad \text{(D-13)}$$

and Equations (D-8), (D-10), and (D-11), the nonlinear model which incorporates both thrust and nozzle inputs, is formed as:

$$\left[\begin{array}{c} \dot{u} \\ \dot{q} \\ \dot{\alpha} \\ \dot{\theta} \\ \hline T \end{array}\right] = \left[\begin{array}{cccc|c} A_{11} & A_{12} & A_{13} & A_{14} & \frac{1}{m} \\ A_{21} & A_{22} & A_{23} & A_{24} & 0 \\ A_{31} & A_{32} & A_{33} & A_{34} & 0 \\ 0 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & \frac{-1}{\tau_T} \end{array}\right] \left[\begin{array}{c} u \\ q \\ \alpha \\ \theta \\ \hline T \end{array}\right]$$

$$+ \begin{bmatrix} B_{11} & B_{12} & B_{13} & 0 \\ B_{12} & B_{22} & [B_{23} + \frac{T}{m(u_o - Z_{\dot{\alpha}})}] & 0 \\ B_{31} & B_{32} & [B_{33} + \frac{\ell T}{I_{yy}}] & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_C \\ \delta_S \\ \delta_N \\ \delta_T \end{bmatrix} \tag{D-14}$$

Note that (D-14) has been written in a form to make the linear terms and nonlinear terms obvious.

Appendix E: Basic Kalman Filtering Theory

The following is a presentation of steady-state constant-gain Kalman filter theory. This is intended to acquaint the reader with the simple form of filter which was used in the controller presented in Chapter V. For a complete treatment of this topic, see Reference 11.

The Kalman filter is an optimal recursive algorithm which is used to produce estimates of system states based on partial, noise-corrupted measurements of the form:

$$\underline{z}(t_i) = \underline{H}\,\underline{x}(t_i) + \underline{v}(t_i) \tag{E-1}$$

where $\underline{H}$ is the system measurement matrix and $\underline{v}$ is a zero-mean white Gaussian noise with associated covariance

$$E\{\underline{v}(t_i)\underline{v}^T(t_j)\} = R\,\delta_{ij} \tag{E-2}$$

The mean and covariance of the states after a measurement are defined conditionally via Bayes rule (11:18) based on the systems measurement history, $Z(t_{i-1})$, to be

$$\hat{\underline{x}}(t_{i-1}^+) = E\{\underline{x}(t_{i-1}) | Z(t_{i-1}) = Z_{i-1}\} \tag{E-3}$$

$$P(t_{i-1}^+) = E\{[\underline{x}(t_{i-1}) - \hat{\underline{x}}(t_{i-1}^+)][\underline{x}(t_{i-1}) - \hat{\underline{x}}(t_{i-1}^+)]^T | Z(t_{i-1}) = Z_{i-1}\} \tag{E-4}$$

Once the update of the state estimate and covariance is accomplished, these quantities are propagated to the next sample time to attain the conditional mean and covariance at that time, before measurement update, defined as

$$\underline{\hat{x}}(t_i^-) = E\{\underline{x}(t_i)|Z(t_{i-1}) = Z_{i-1}\} \tag{E-5}$$

$$P(t_i^-) = E\{[x(t_i)-\hat{x}(t_i^-)][\underline{x}(t_i)-\hat{x}(t_i^-)]^T|Z(t_{i-1}) = Z_{i-1}\} \tag{E-6}$$

Based on the above equations, the Kalman filter equations are shown to be (11):

$$\underline{\hat{x}}(t_i^-) = \underline{\Phi}\underline{\hat{x}}(t_{i-1}^+) + \underline{B}_d\,\underline{u}(t_{i-1}) \tag{E-7}$$

$$\underline{P}(t_i^-) = \underline{\Phi}P(t_{i-1}^+)\Phi^T + \underline{G}_d\,Q_d\,G_d^+ \tag{E-8}$$

$$\underline{K}(t_i^-) = P(t_i^-)H^T[H\,P(t_i^-)H^T+R]^{-1} \tag{E-9}$$

$$\underline{\hat{x}}(t_i^+) = \underline{\hat{x}}(t_i^-) + \underline{K}(t_i)[\underline{z}_i-H\,\underline{\hat{x}}(t_i^-)] \tag{E-10}$$

$$\underline{P}(t_i^+) = \underline{P}(t_i^-) - \underline{K}(t_i)\;H\,P(t_i^-) \tag{E-11}$$

It is this form of the Kalman filter which is used to replace the assumption of full state availability with estimates produced as in Equation (E-10) for the designs presented in Chapter V.

Appendix F: Plotted Data for Section 5.7

Introduction

The following plots correspond to the robustness analysis conducted in the course of this thesis (see Section 5.7). Each plot contains the time histories of the pitch angle, flight path, and pitch rate channels along with throttle, canard, and stabilator deflections over a 6-second period.

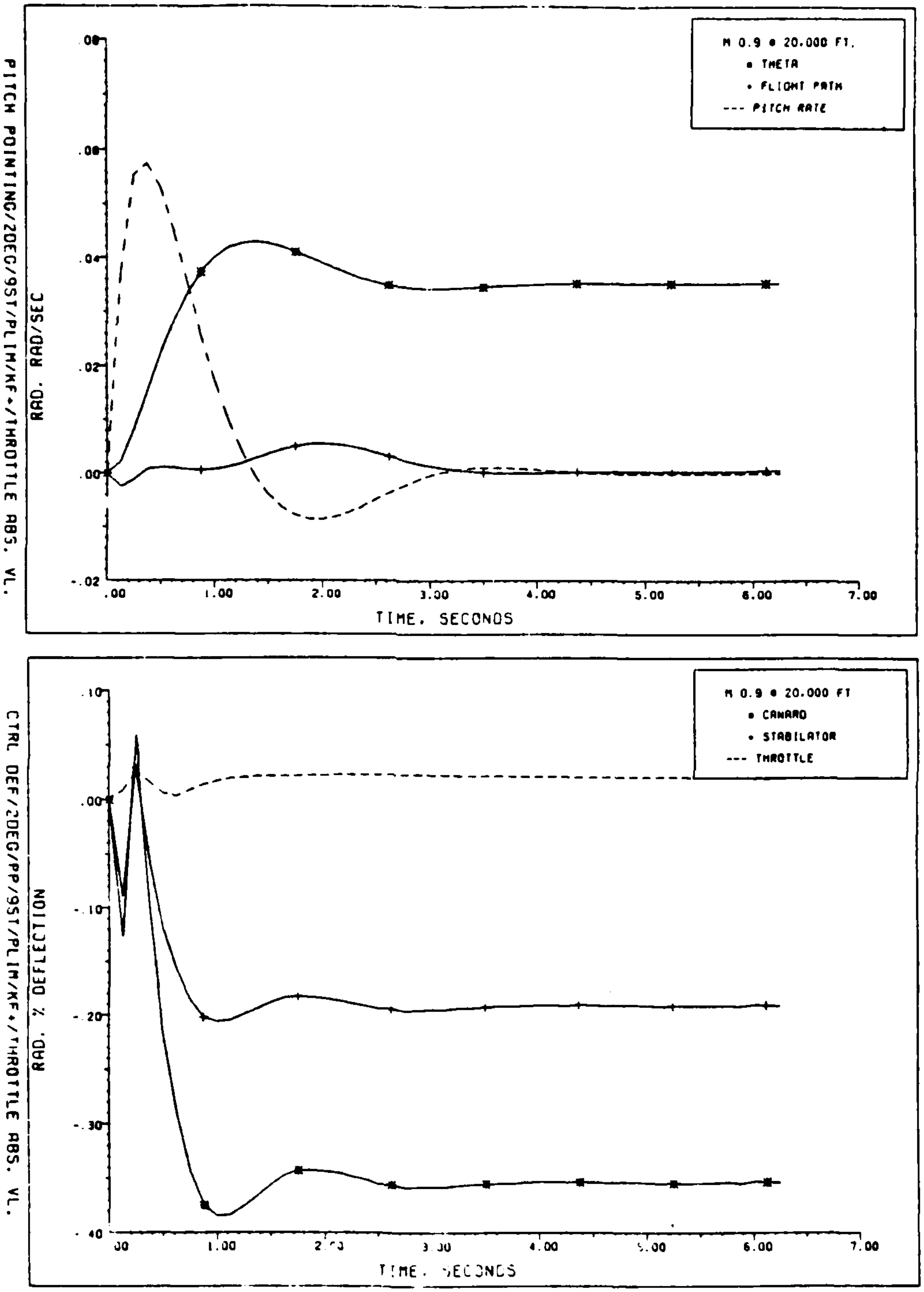


Fig. F.1. Aircraft Response with Actuators, Position Limits, and Kalman Filter in the Loop/Throttle Absolute Value Function
a. Aircraft responses; b. Control deflections

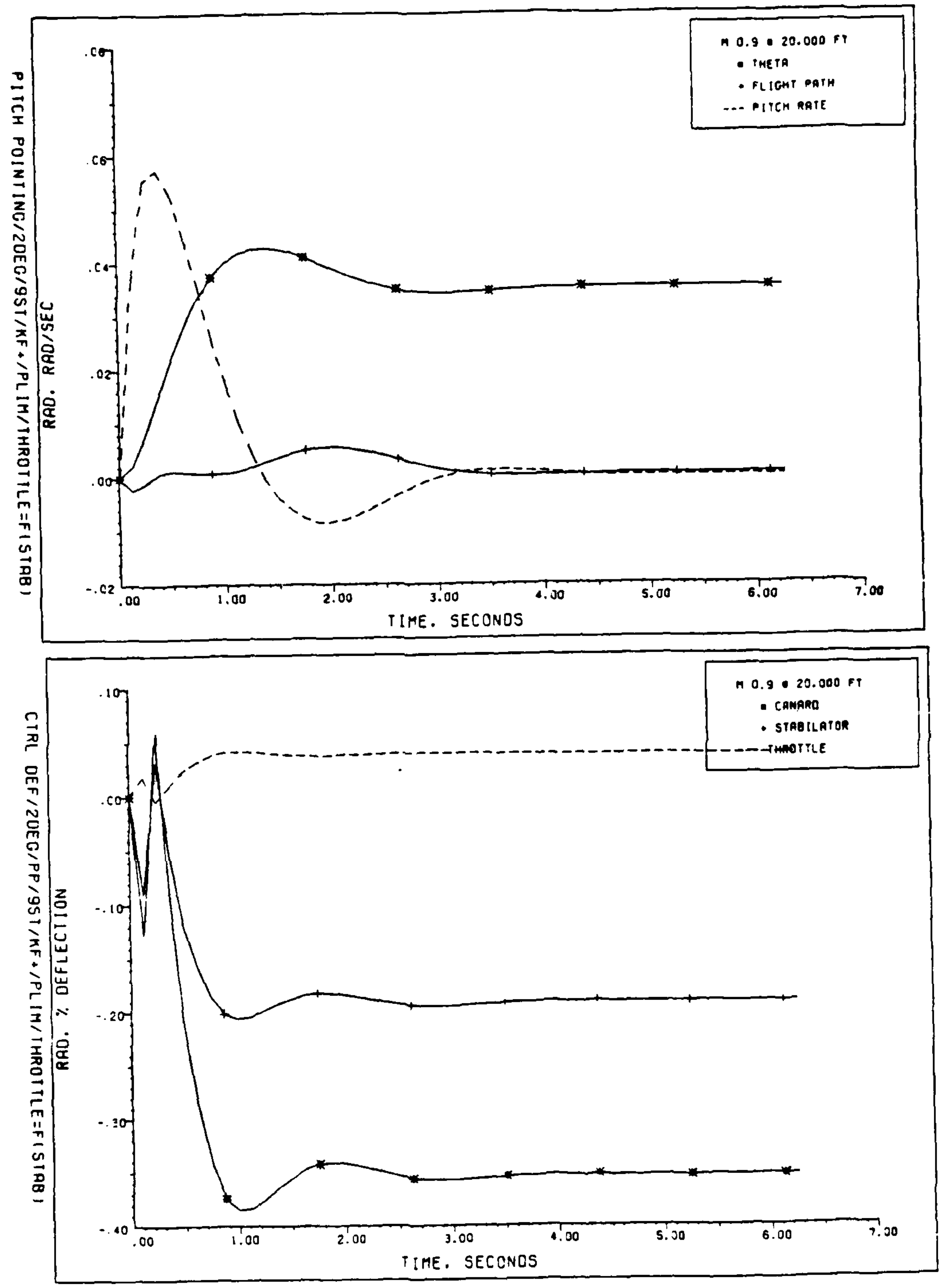


Fig. F.2. Aircraft Response with Actuators, Position Limits, and Kalman Filter in the Loop/Throttle--A Linear Function of Stabilator Deflection
a. Aircraft responses; b. Control deflections

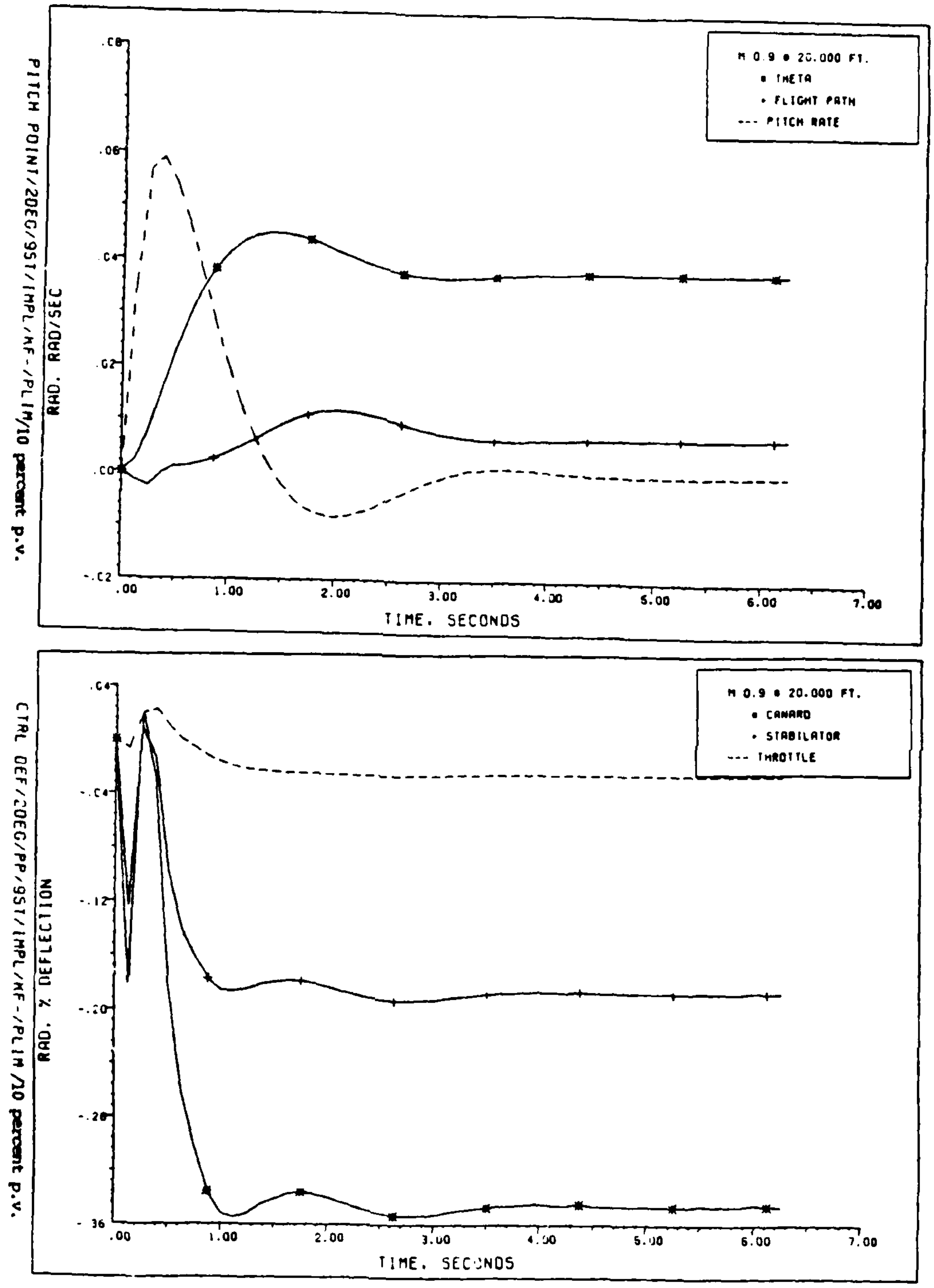


Fig. F.3. Aircraft Response with Actuators, Position Limits, and Kalman Filter in the Loop/Control Based on $\hat{\underline{x}}(t_i^-)$/10 Percent Increase in $\underline{F}$ Matrix
a. Aircraft responses; b. Control deflections

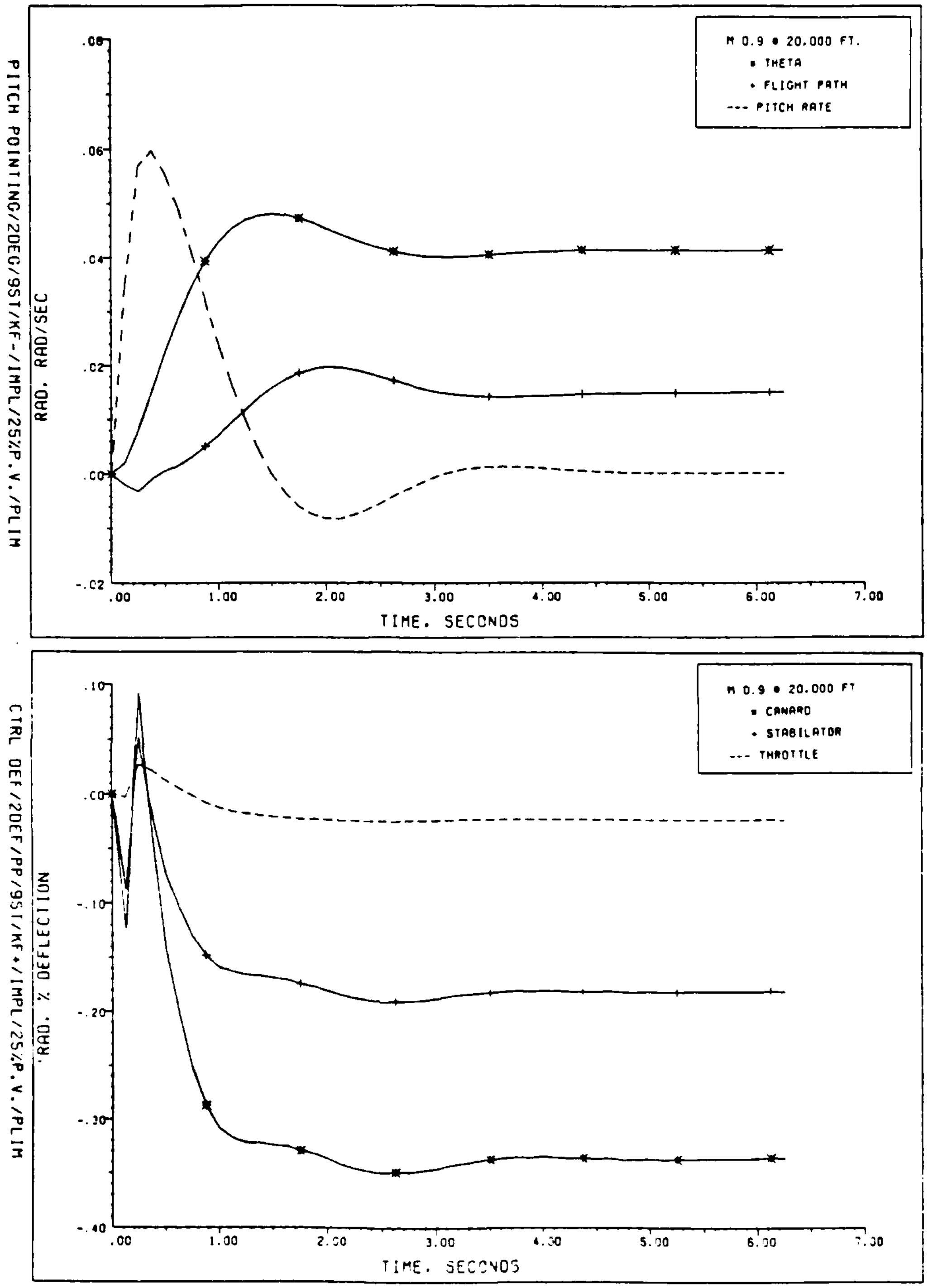


Fig. F.4. Aircraft response with Actuators, Position Limits, and Kalman Filter in the Loop/Control Based on $\hat{\underline{x}}(t_i^-)$/ 25 Percent Increase in $\underline{F}$ Matrix
a. Aircraft responses; b. Control deflections

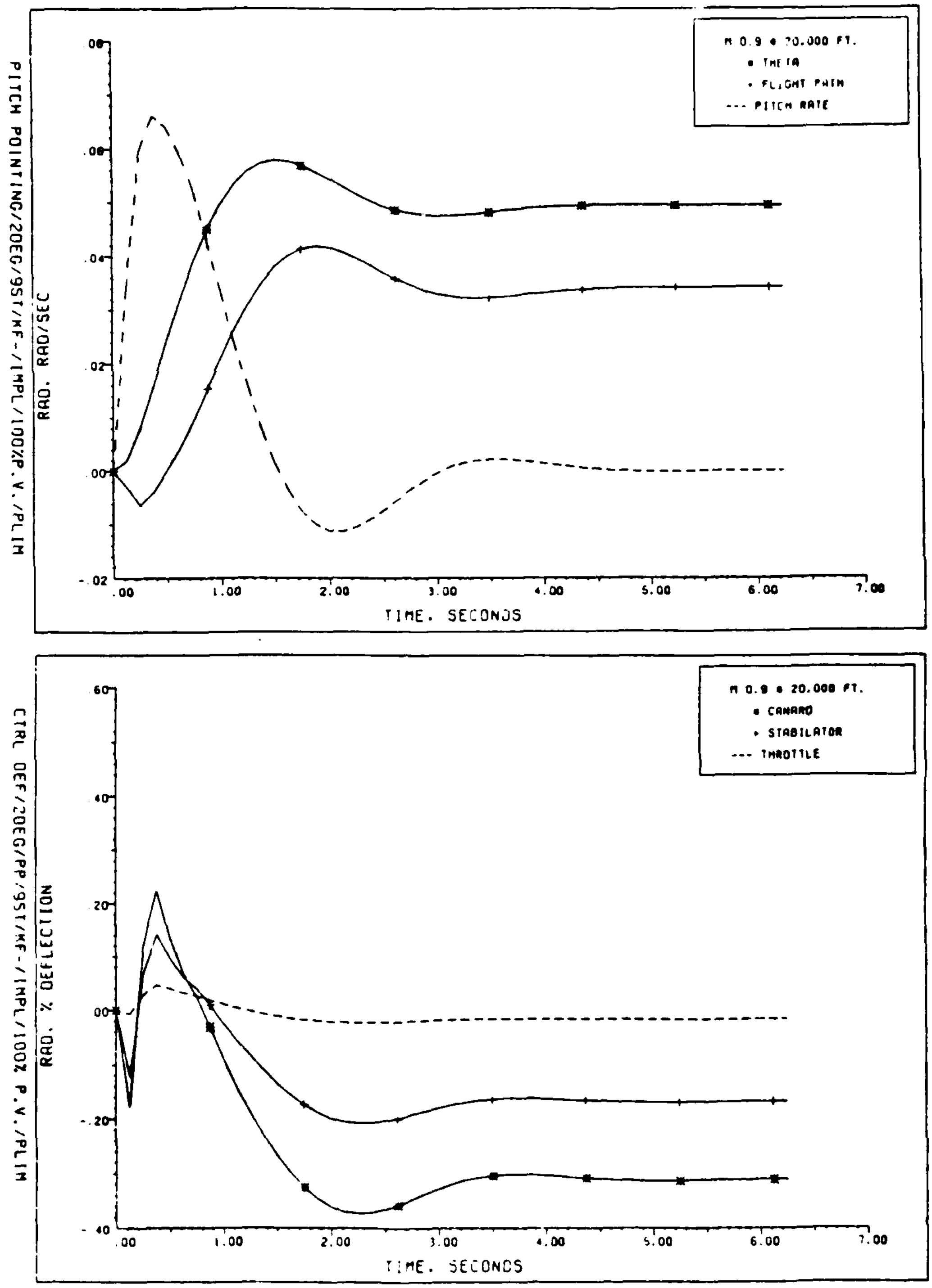


Fig. F.5. Aircraft Response with Actuators, Position Limits, and Kalman Filter in the Loop/Control Based on $\hat{x}(t_i^-)$/100 Percent Increase in $\underline{F}$ Matrix
$\underline{a}$. Aircraft responses; b. Control deflections

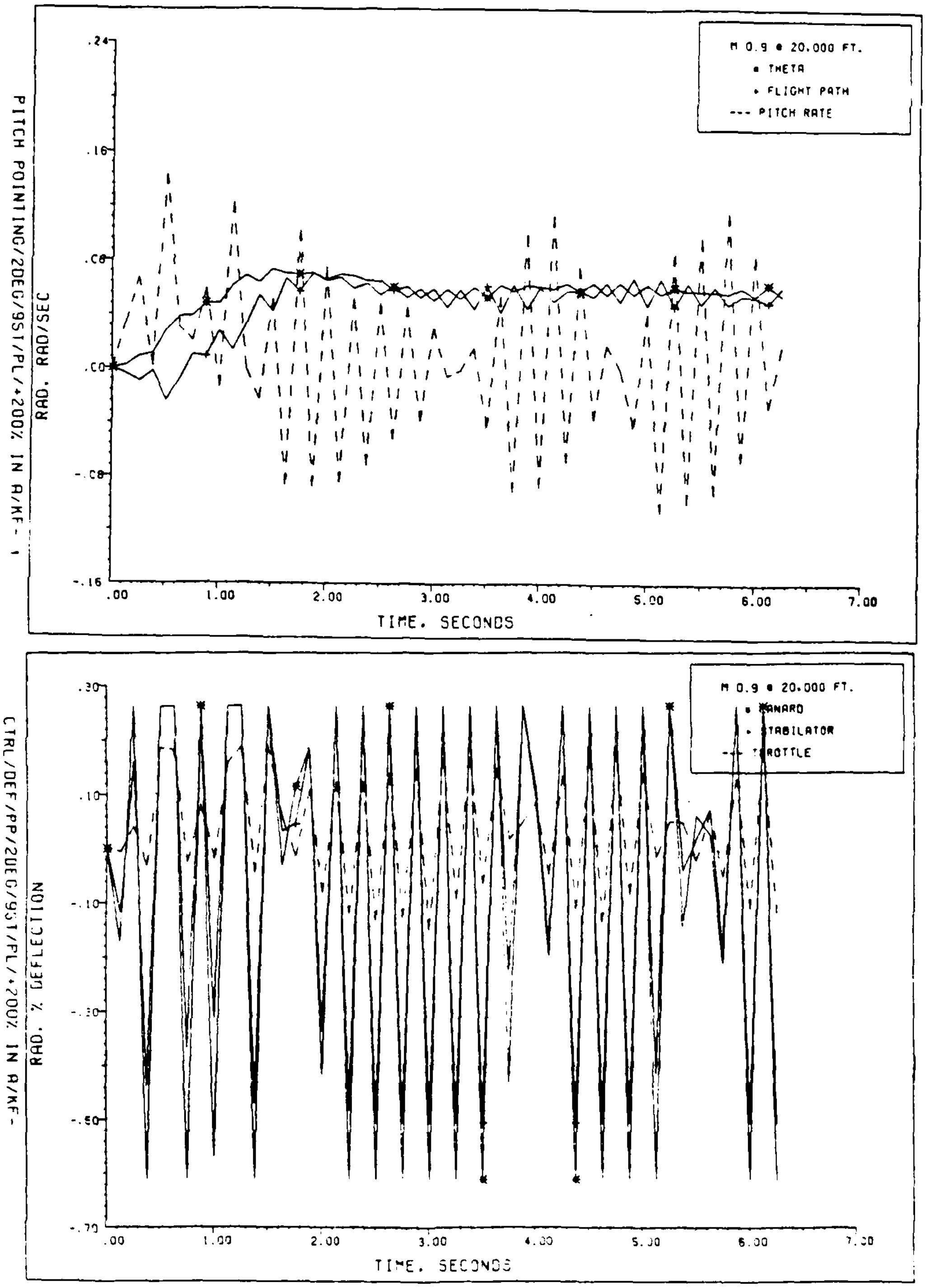


Fig. F.6. Aircraft Response with Actuators, Position Limits, and Kalman Filter in the Loop/Control Based on $\hat{\underline{x}}(t_i^-)$/ 200 Percent increase in $\underline{F}$ Matrix
a. Aircraft responses; b. Control deflections

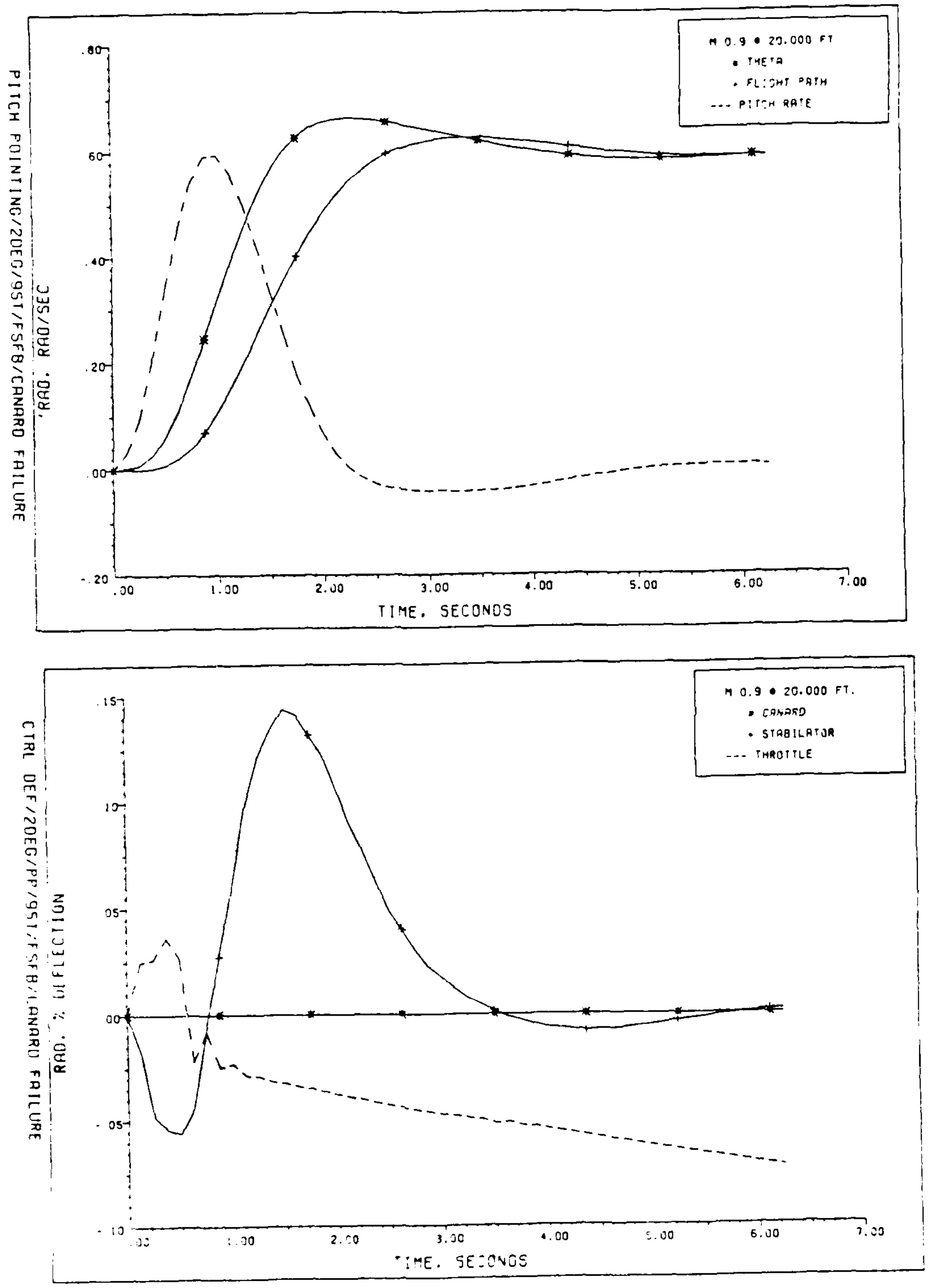


Fig. F.7. Aircraft Response with Actuators/Free Floating Canard Failure
a. Aircraft responses; b. Control deflections

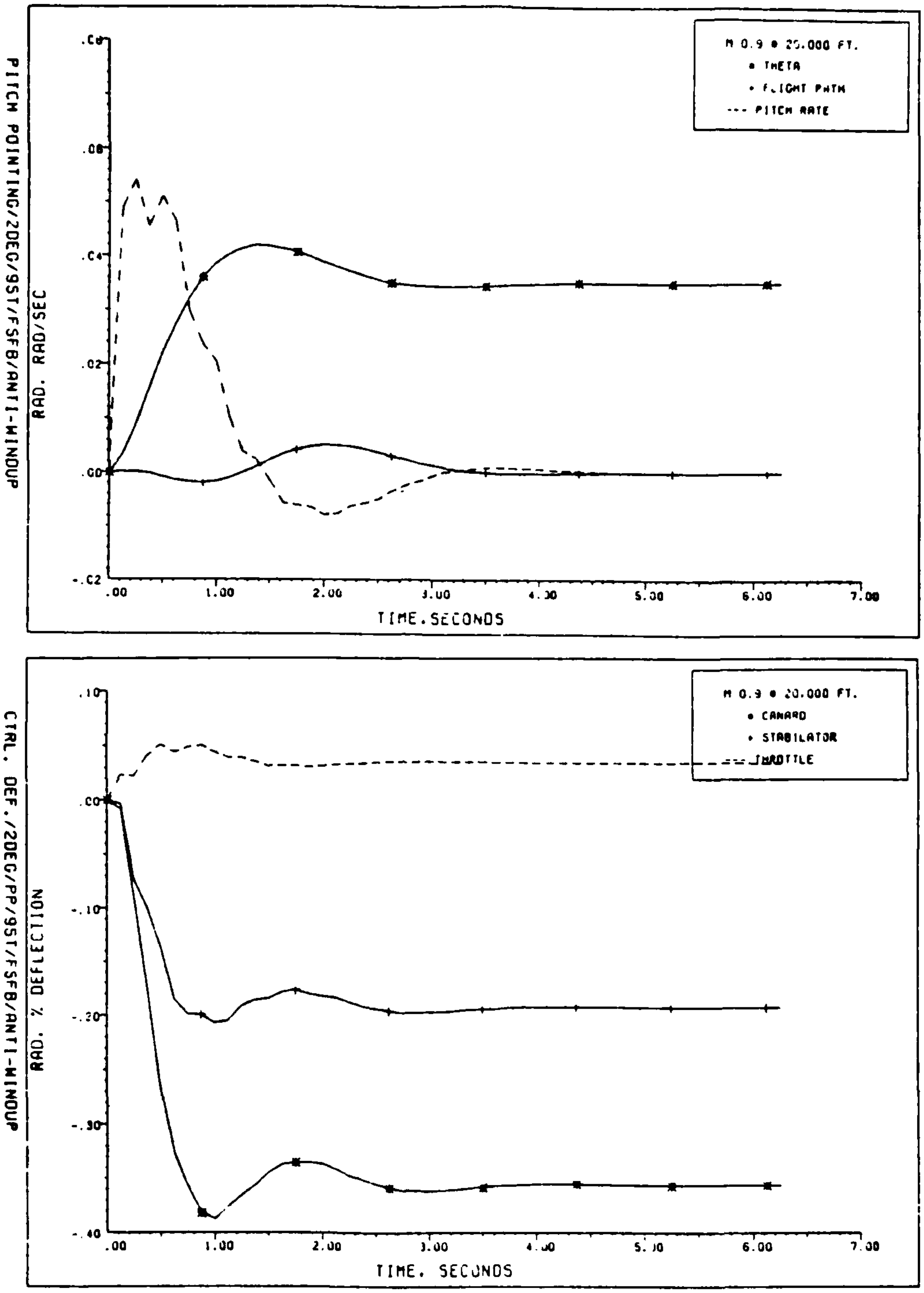


Fig. F.8. Aircraft Response with Actuators/
Anti-Windup Compensation
a. Aircraft responses; b. Control deflections

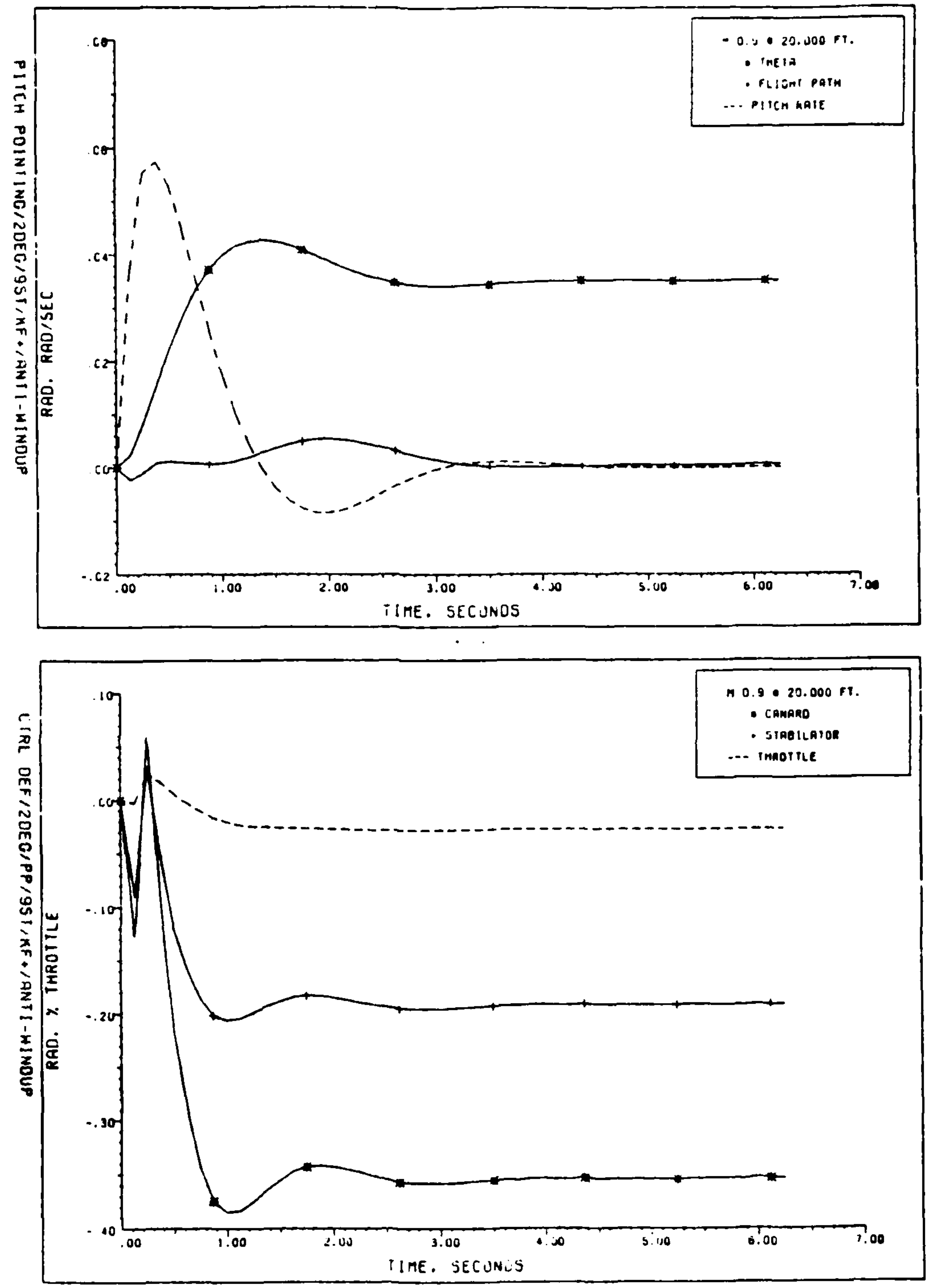


Fig. F.9. Aircraft Response with Actuators and Kalman Filter in the Loop/Anti-Windup Compensation
a. Aircraft responses; b. Control deflections

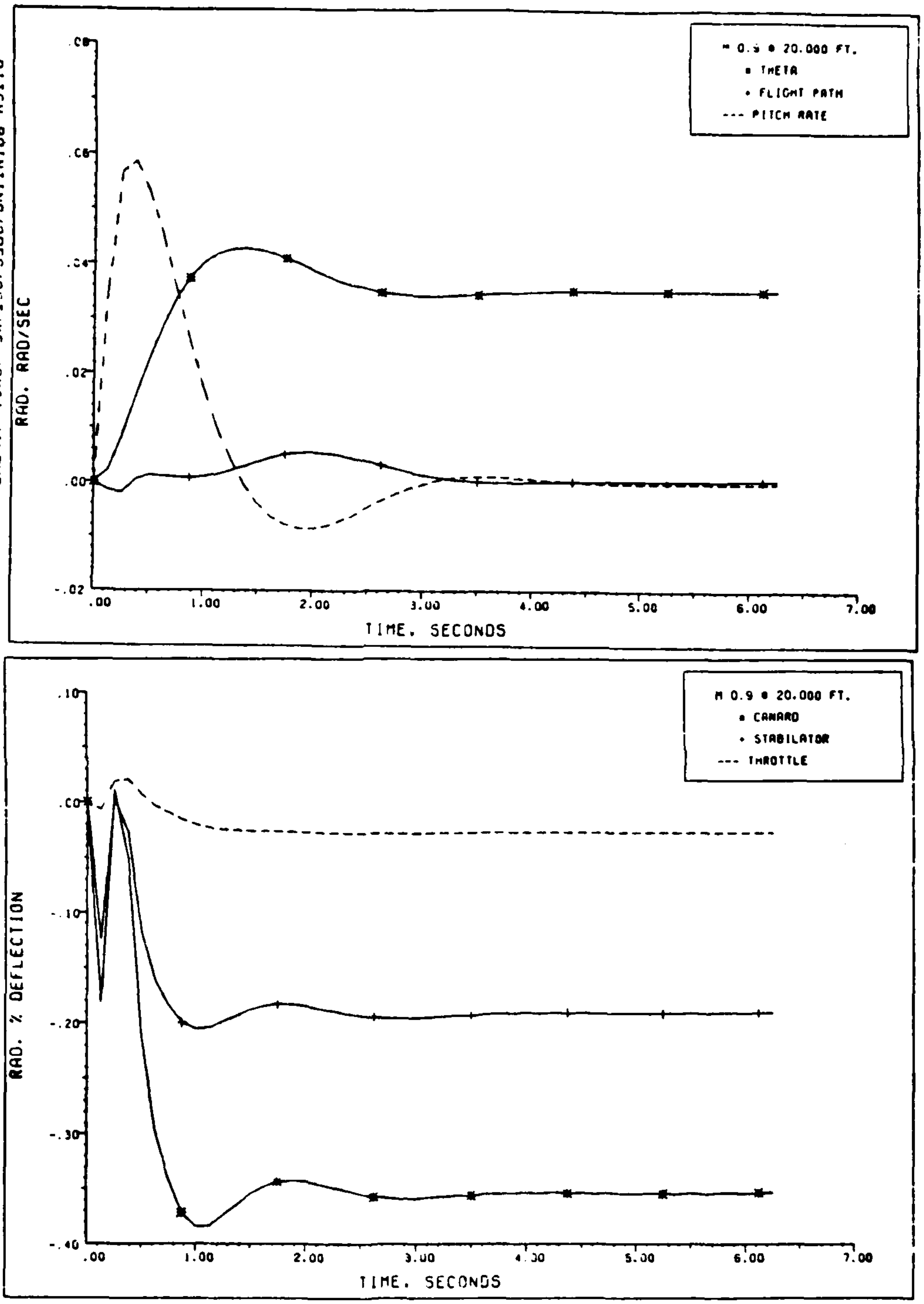


Fig. F.10. Aircraft Response with Actuators and Kalman Filter in the Loop/Control Based on $\hat{\underline{x}}(t_i^-)$/ Anti-Windup Compensation

a. Aircraft responses; b. Control deflections

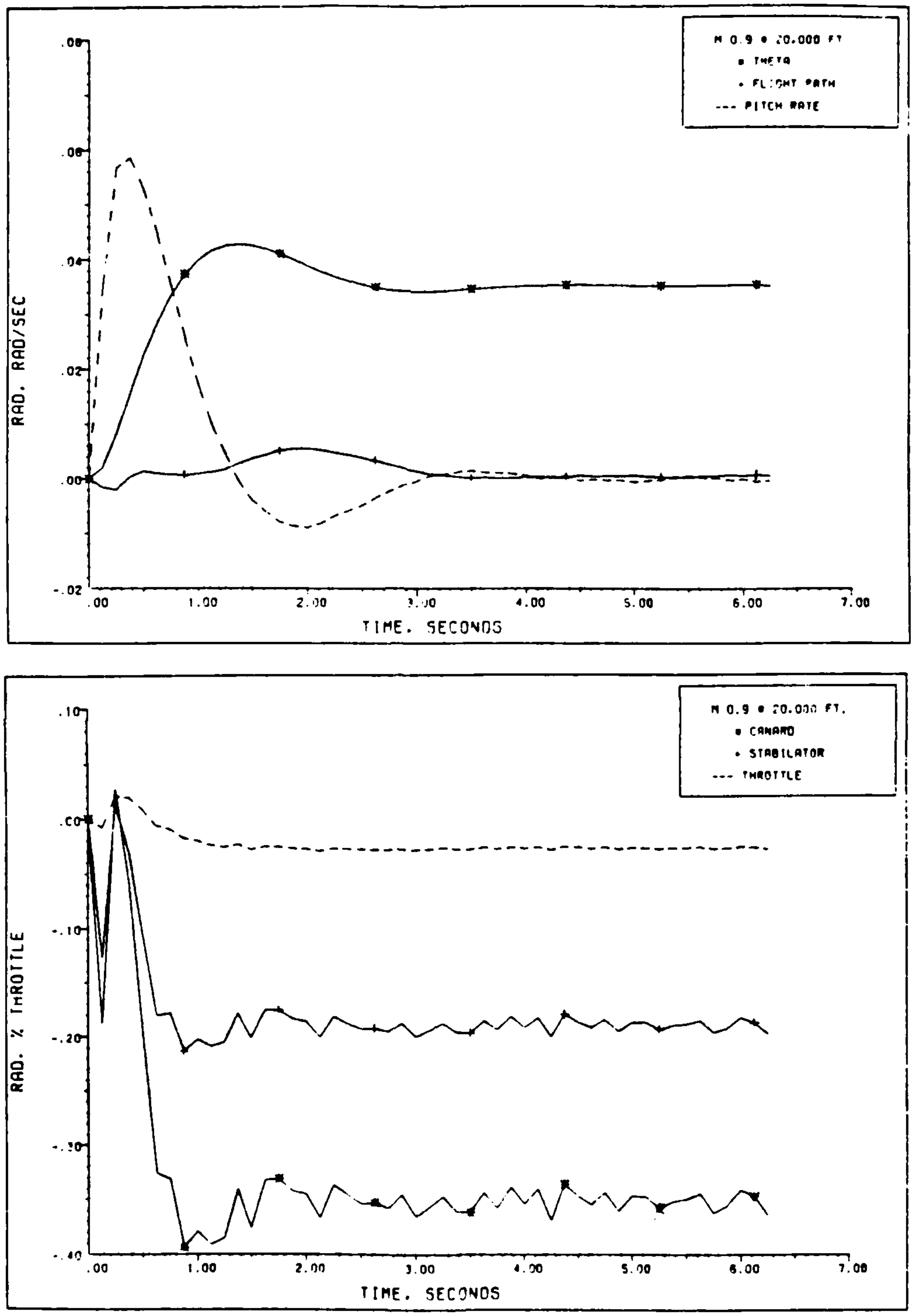


Fig. F.11. Aircraft Response with Actuators, Position Limits, and Kalman Filter in the Loop/Control Based on $\hat{\underline{x}}(t_i^-)$/ 40 Fold Increase in Measurement Noise
a. Aircraft responses; b. Control deflections

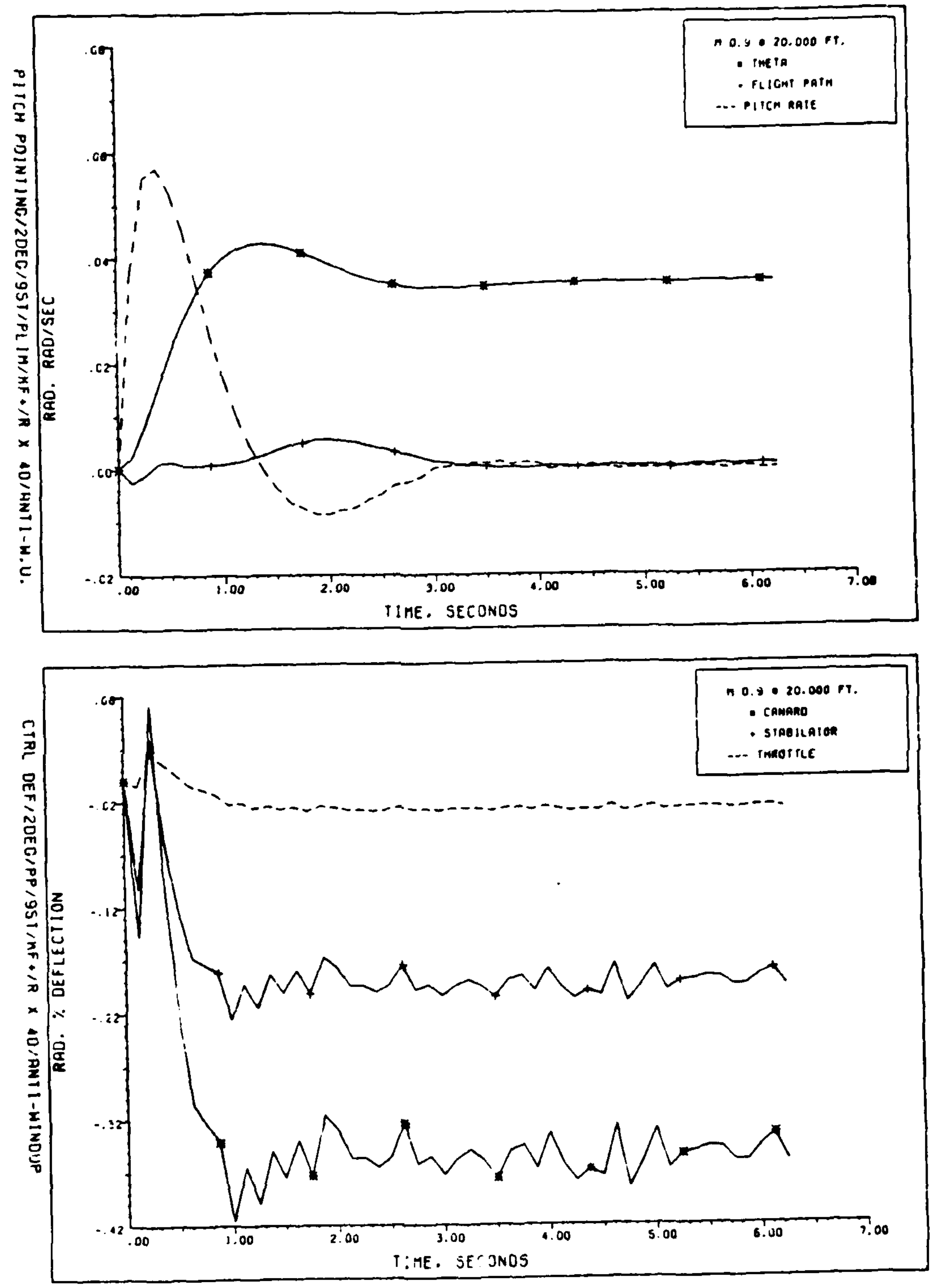


Fig. F.12. Aircraft Response with Actuators, Position Limits, and Kalman Filter in the Loop/Anti-Windup Compensation/40 Fold Increase in Measurement Noise
a. Aircraft responses; b. Control deflections

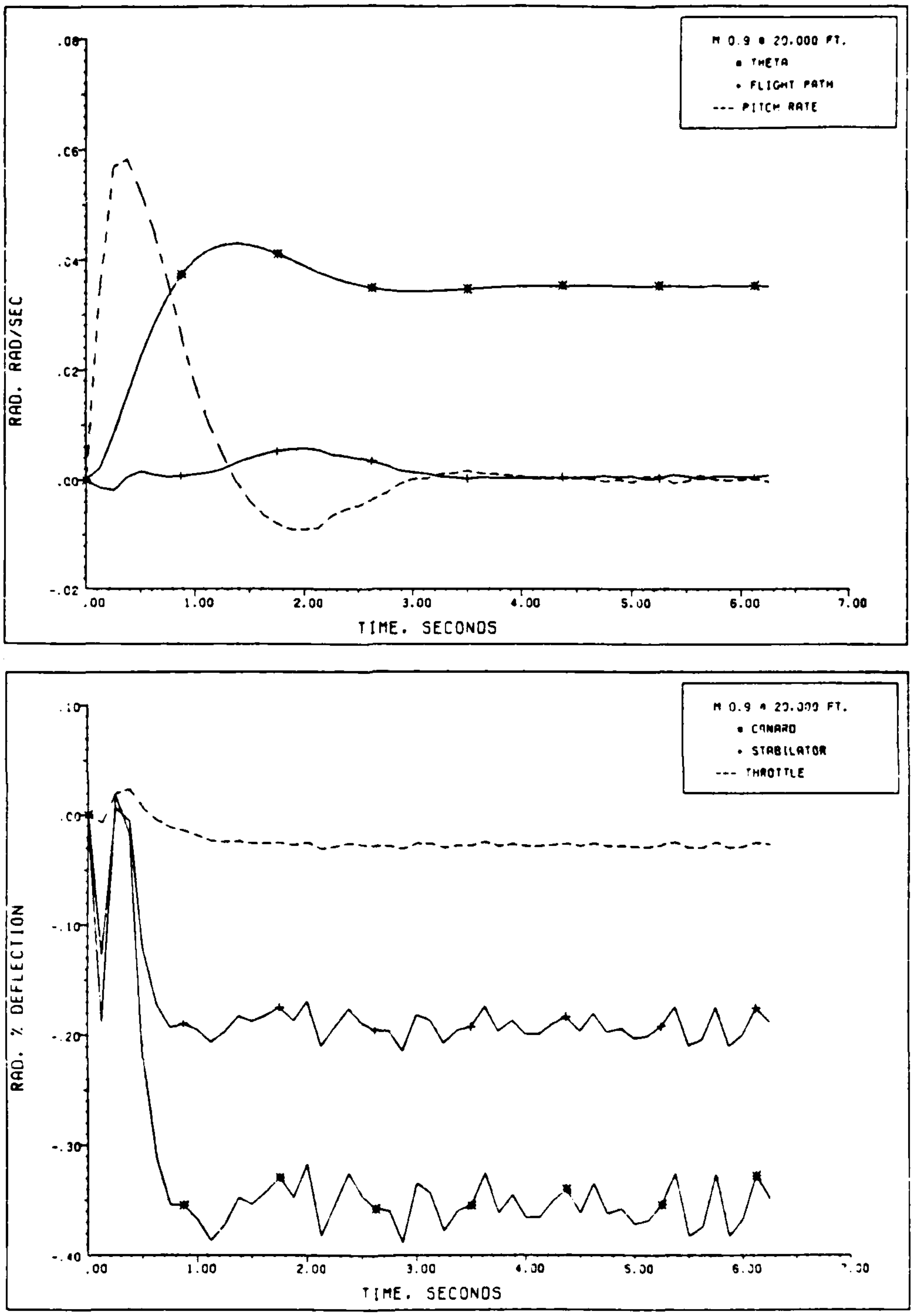


Fig. F.13. Aircraft Response with Actuators and Kalman Filter in the Loop/Control Based on $\underline{x}(t_i^-)$/ Anti-Windup Compensation/40 Fold Increase in Measurement Noise
a. Aircraft responses; b. Control deflections

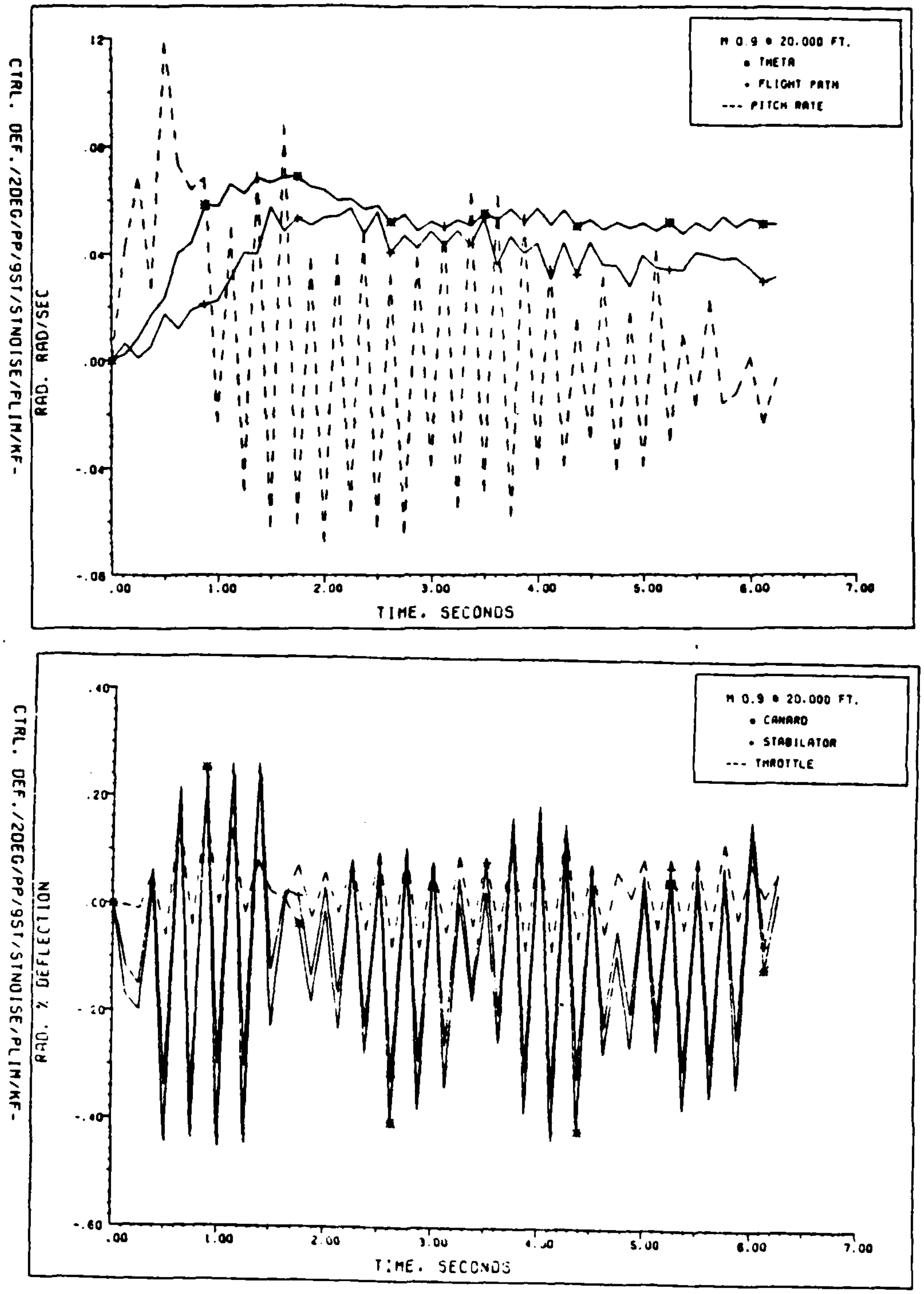


Fig. F.14. Aircraft Response with Actuators, Position Limits, and Kalman Filter in the Loop/Control Based on $\hat{\underline{x}}(t_i^-)$/ High Level Noise (q=0.8) in Pitch Rate and Angle of Attack Channels
a. Aircraft responses; b. Control deflections

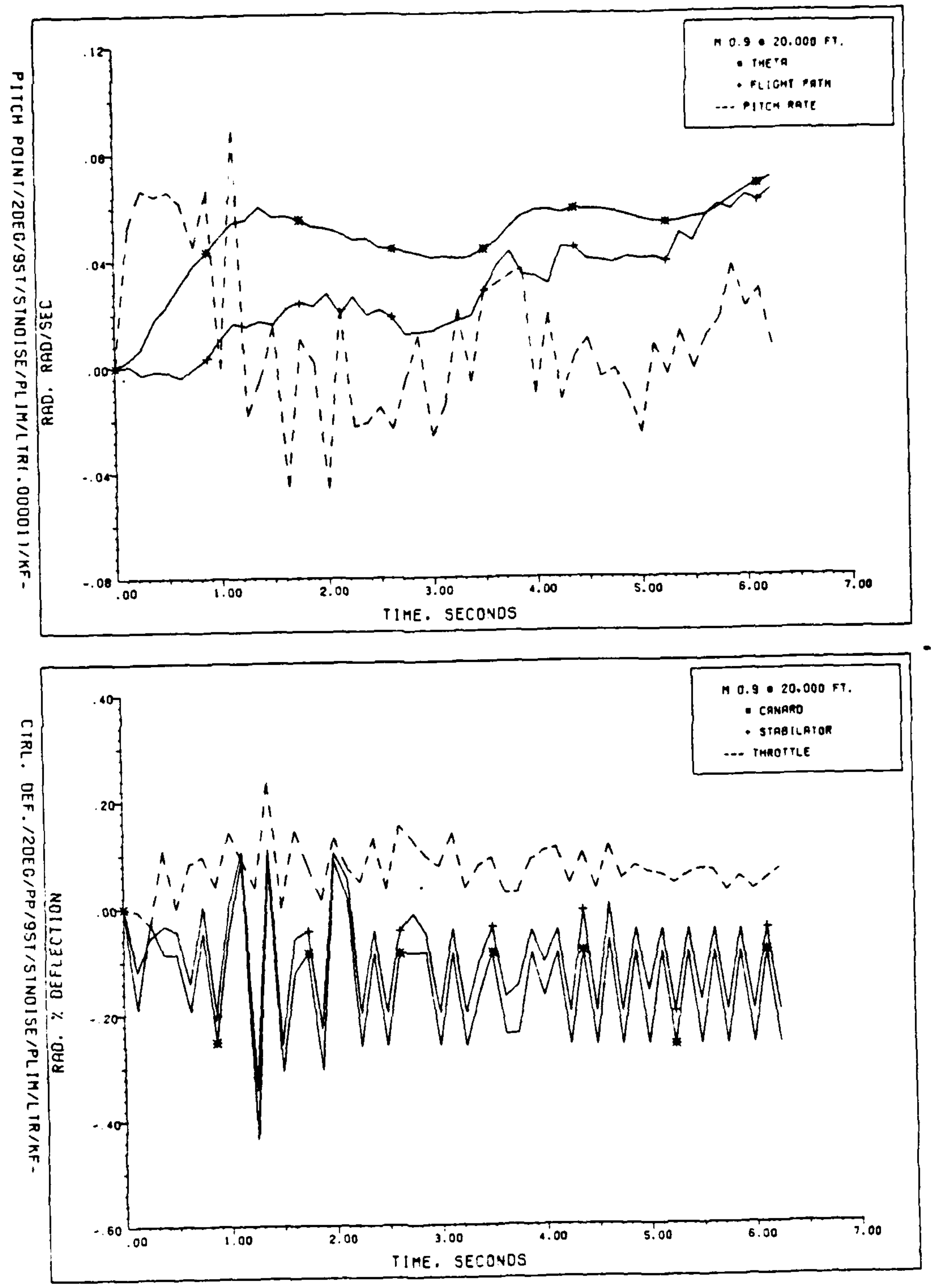


Fig. F.15. Aircraft Response with Actuators, Position Limits, and Kalman Filter in the Loop/Control Based on $\hat{\underline{x}}(t_i^-)$/ High Level Noise (q=0.8) in Pitch Rate and Angle of Attack Channels/LTR Tuning with Loop Broken at the Input
a. Aircraft responses; b. Control deflections

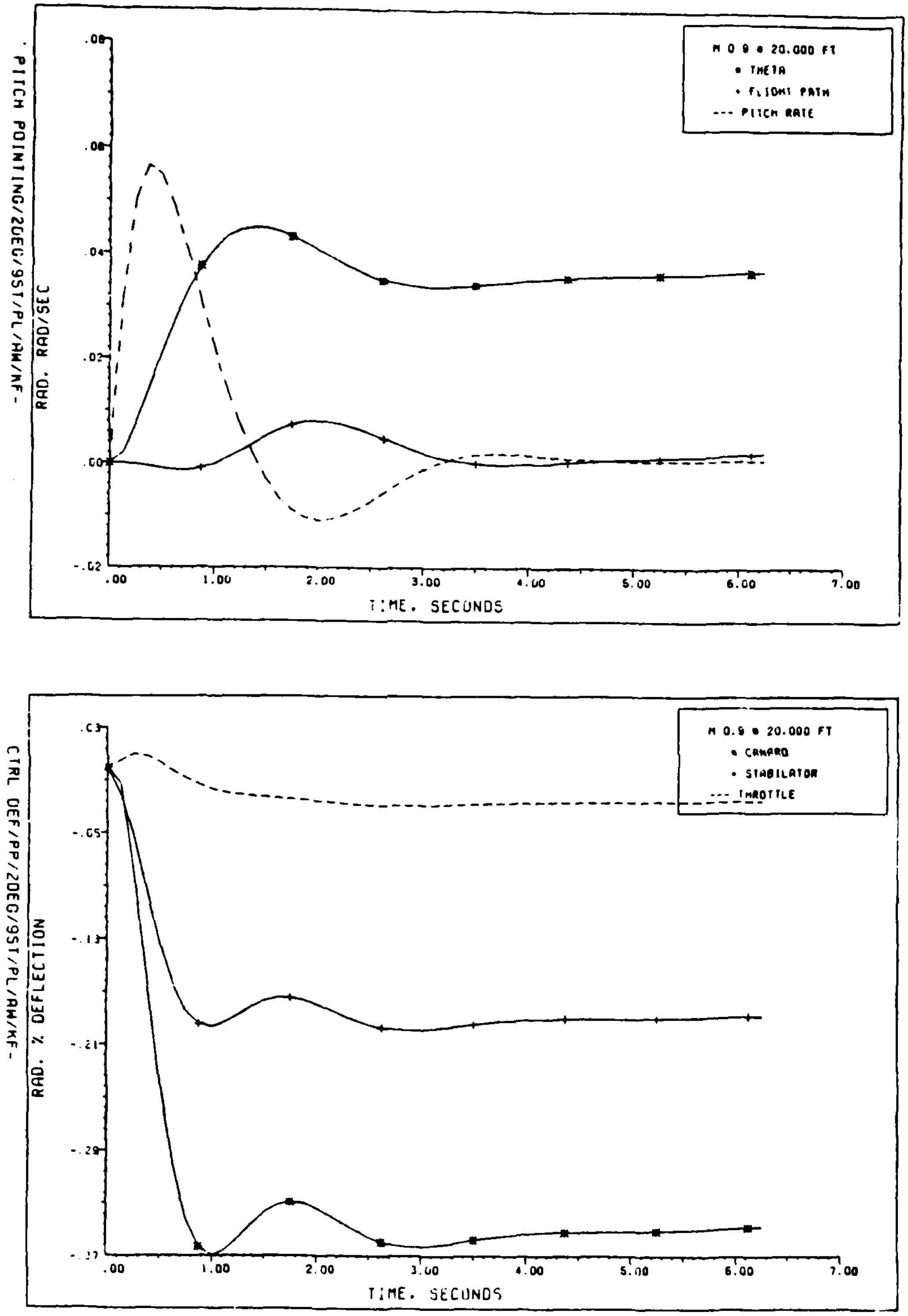


Fig. F.16. Aircraft Response with Actuators, Position Limits, and Kalman Filter in the Loop/Anti-Windup Compensation/C-Tuned

a. Aircraft responses; b. Control deflections

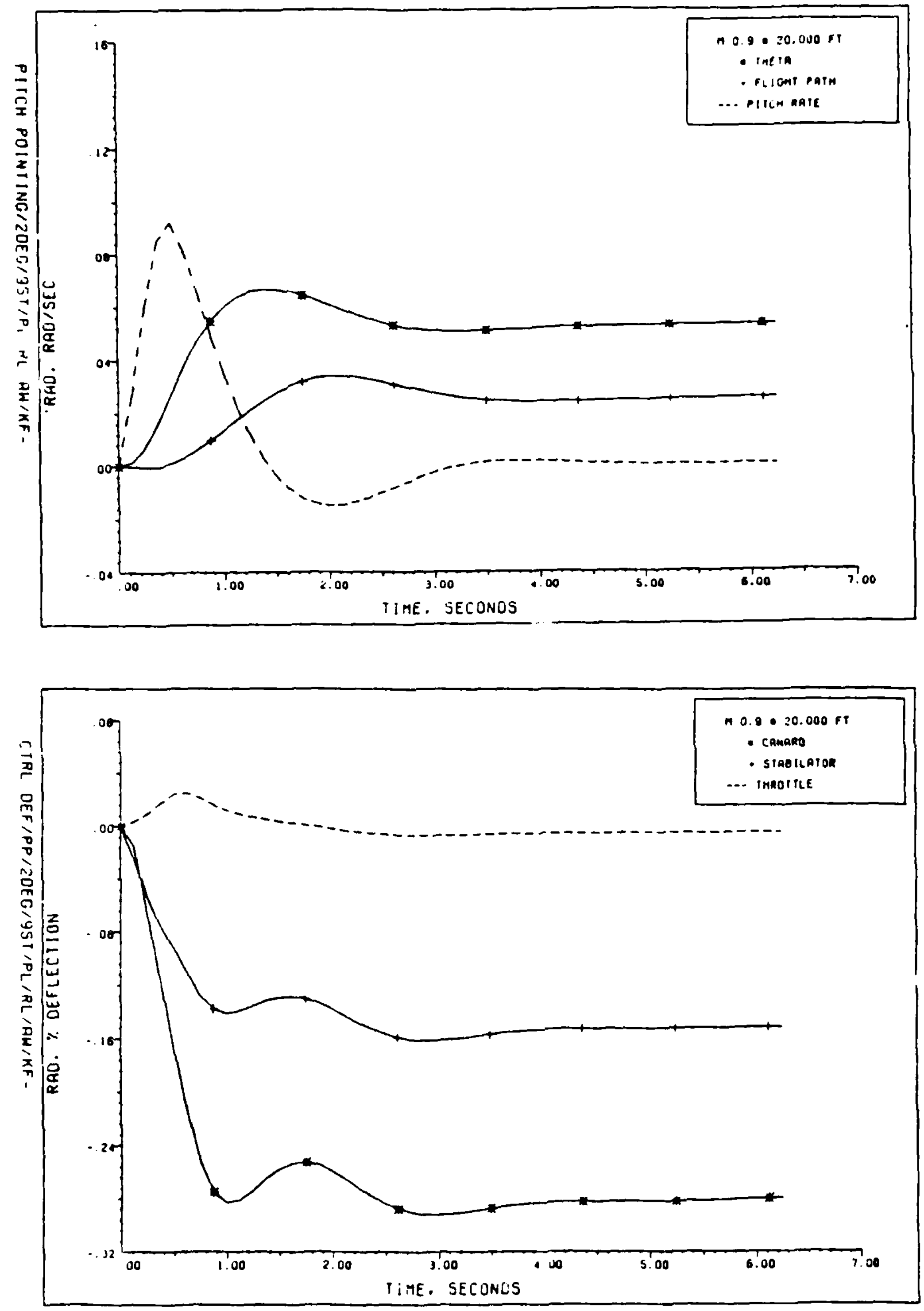


Fig. F.17. Aircraft Response with Actuators, Position Limits, and Kalman Filter in the Loop/Control Based on $\underline{\hat{x}}(t_i^-)$ / Anti-Windup Compensation/C-Tuned
a. Aircraft responses; b. Control deflections

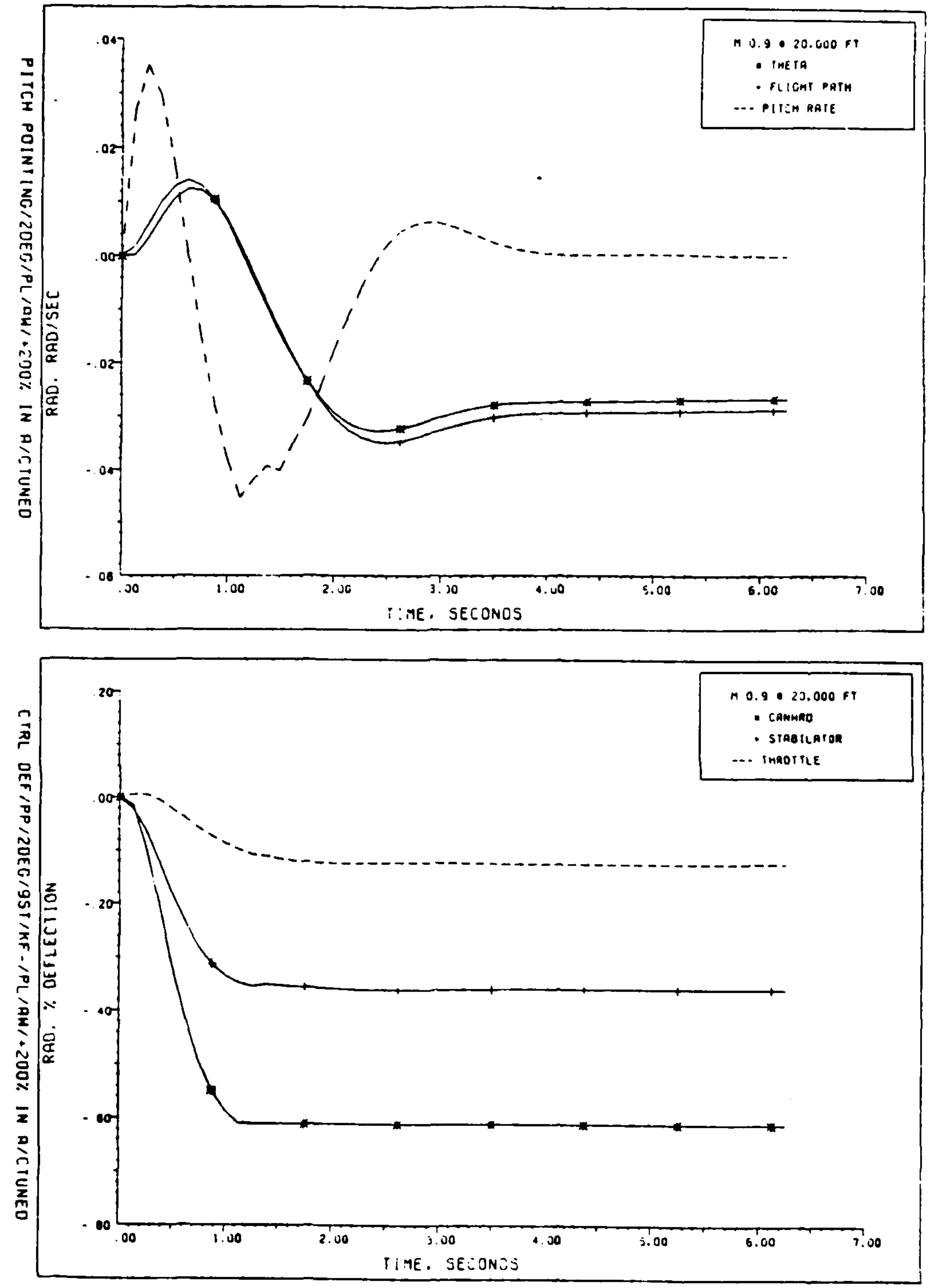


Fig. F.18. Aircraft Response with Actuators, Position Limits, Rate Limits, and Kalman Filter in the Loop/ Control Based on $\hat{\underline{x}}(t_i^-)$/C-Tuned/Anti-Windup Compensation/200 Percent Increase in $\underline{F}$ Matrix
a. Aircraft responses; b. Control deflections

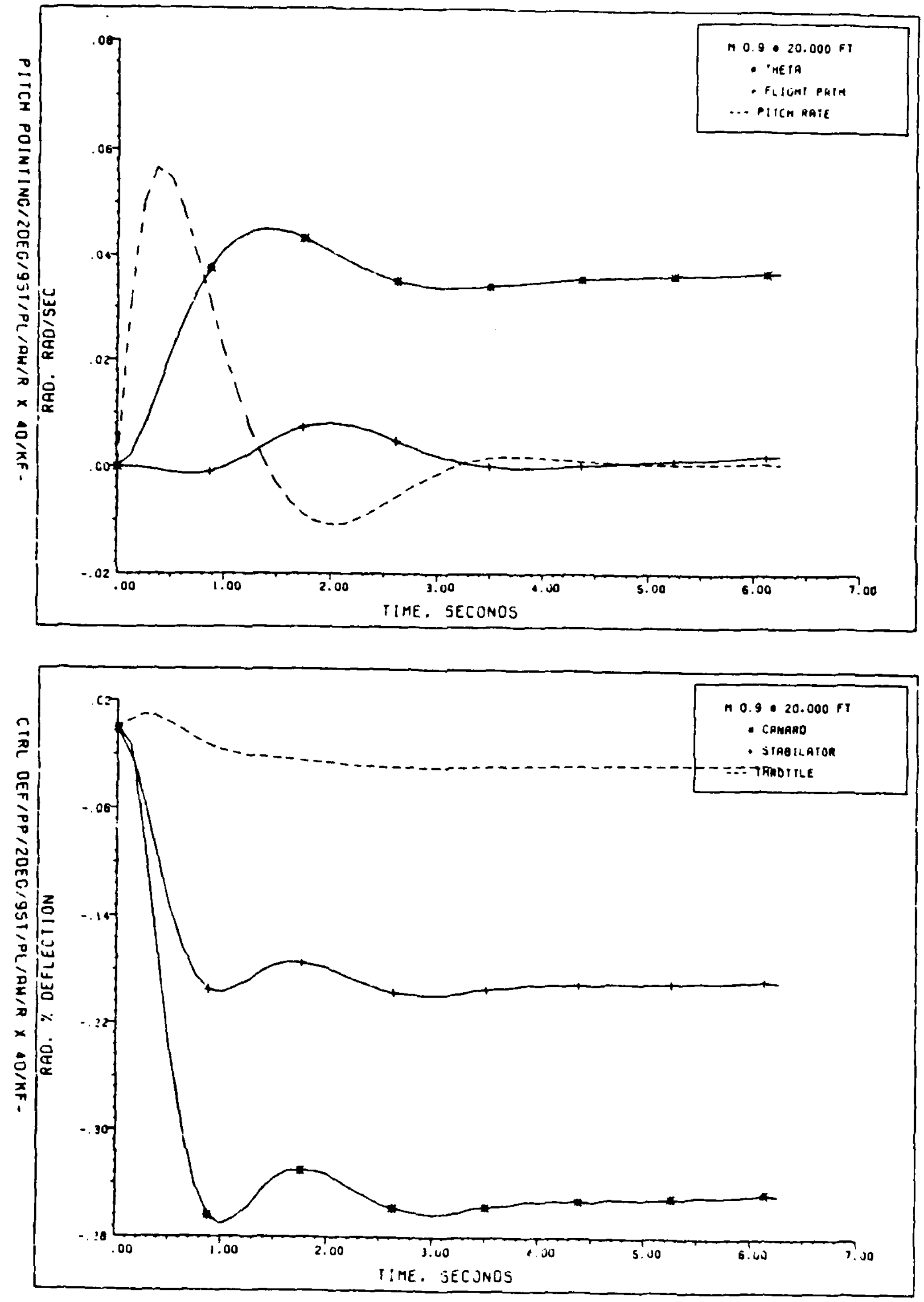


Fig. F.19. Aircraft Response with Actuators, Position Limits, Anti-Windup Compensation/Control Based on $\hat{\underline{x}}(t_i^-)$/40 Fold Increase in Measurement Noise/ C-Tuned
a. Aircraft responses; b. Control deflections

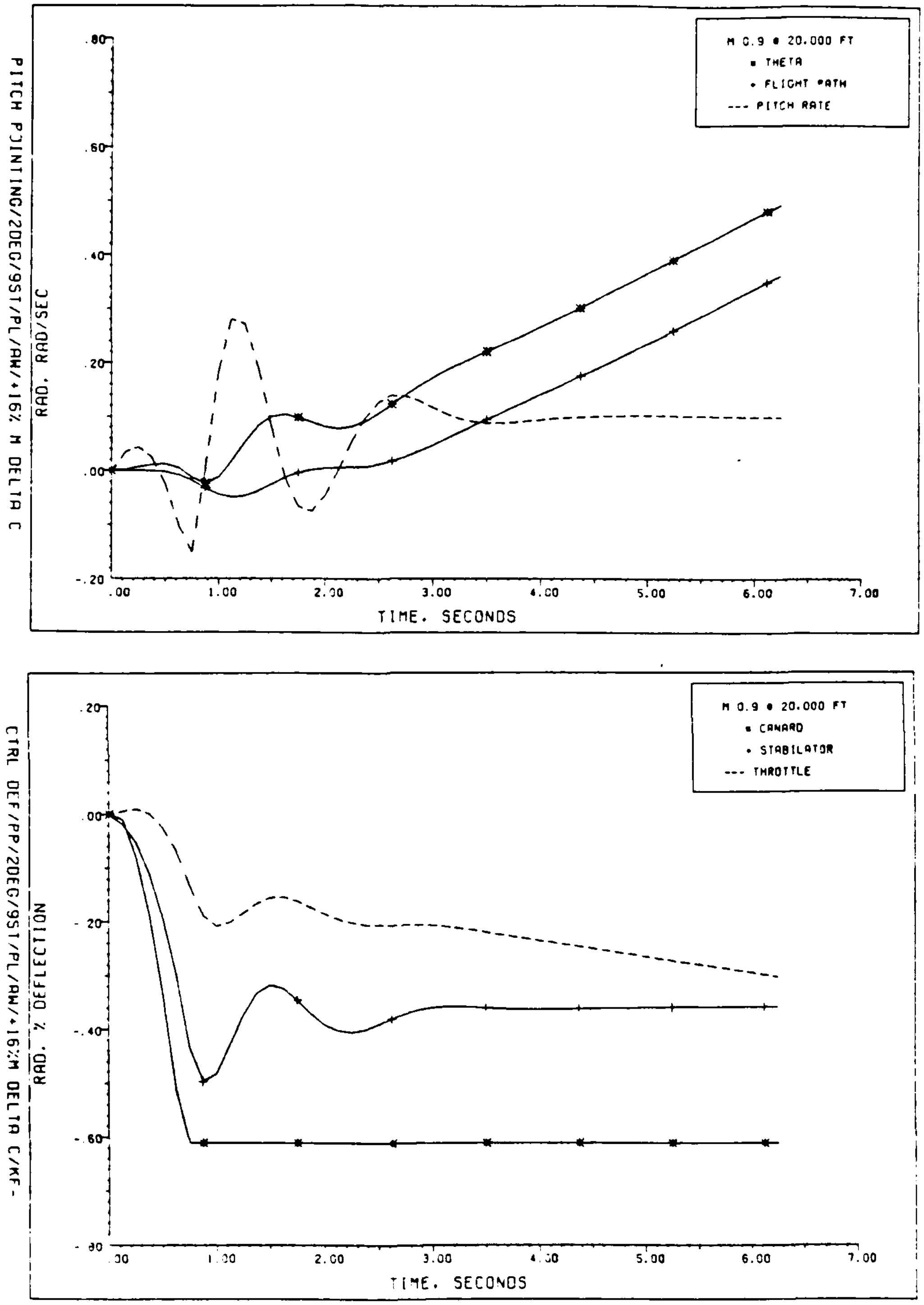


Fig. F.20. Aircraft Response with Actuators, Position Limits, Anti-Windup Compensation/Control Based on $\hat{\underline{x}}(t_i^-)$/ C-Tuned/16 Percent Increase in M_{δ_C}

a. Aircraft responses; b. Control deflections

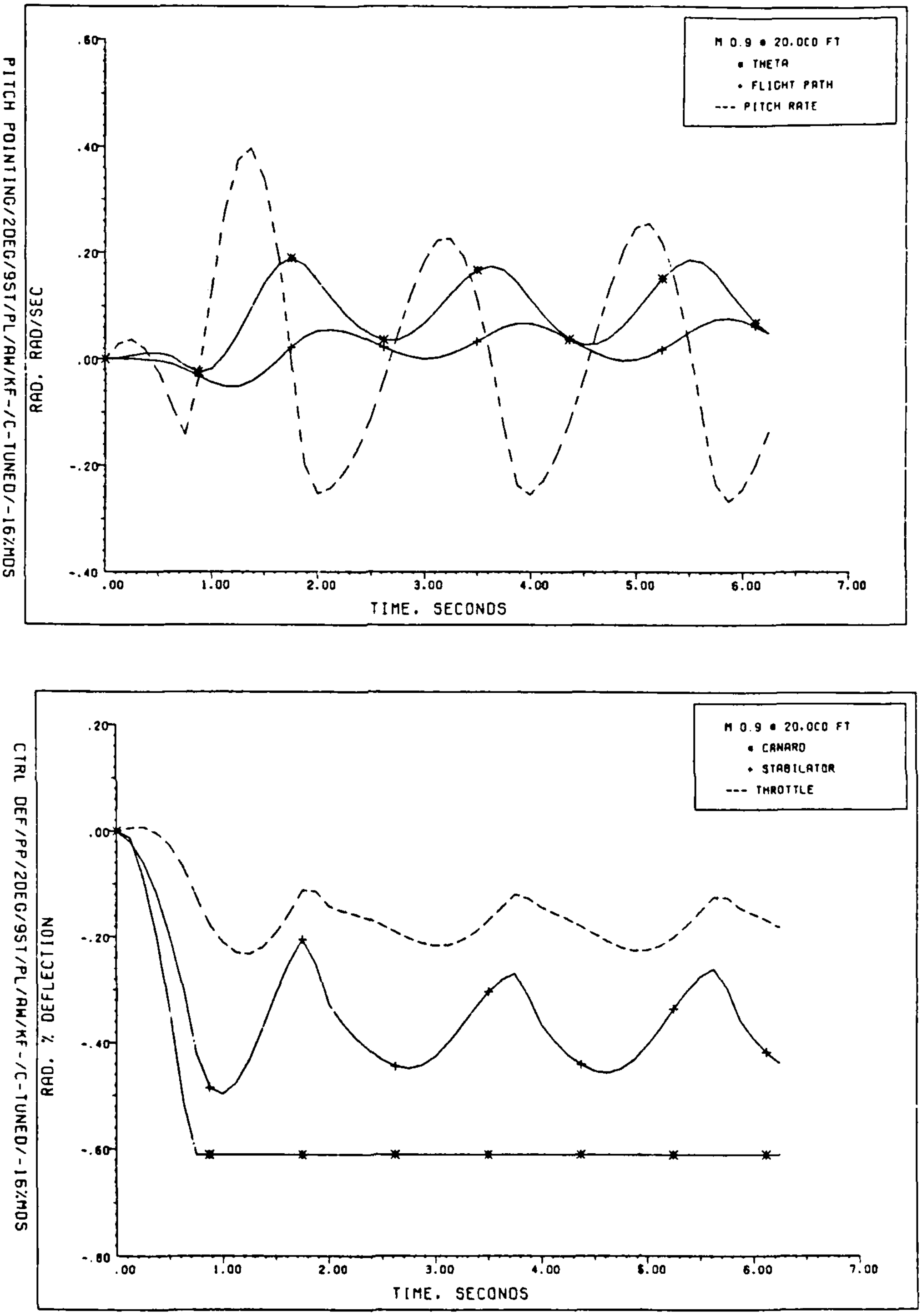


Fig. F.21. Aircraft Response with Actuators, Position Limits, Anti-Windup Compensation/Control Based on $\hat{\underline{x}}(t_i^-)$/ C-Tuned/16 Percent Increase in M_{δ_S}

a. Aircraft responses; b. Control deflections

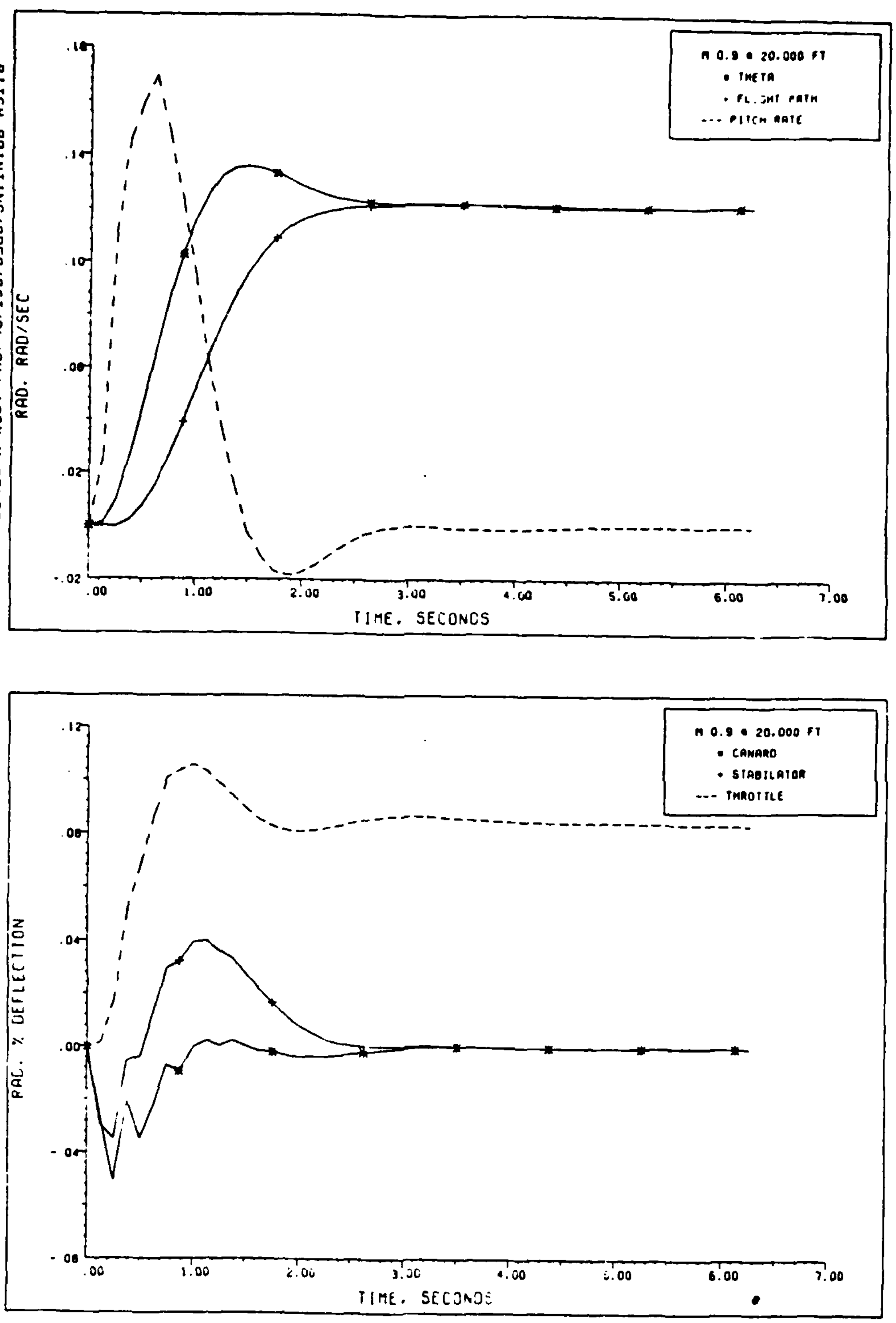


Fig. F.22. Aircraft Response with Actuators, Position Limits, Anti-Windup Compensation/Control Based on $\hat{\underline{x}}(t_i^-)$/ C-Tuned/40 Percent Decrease in M_{δ_S}

a. Aircraft responses; b. Control deflections

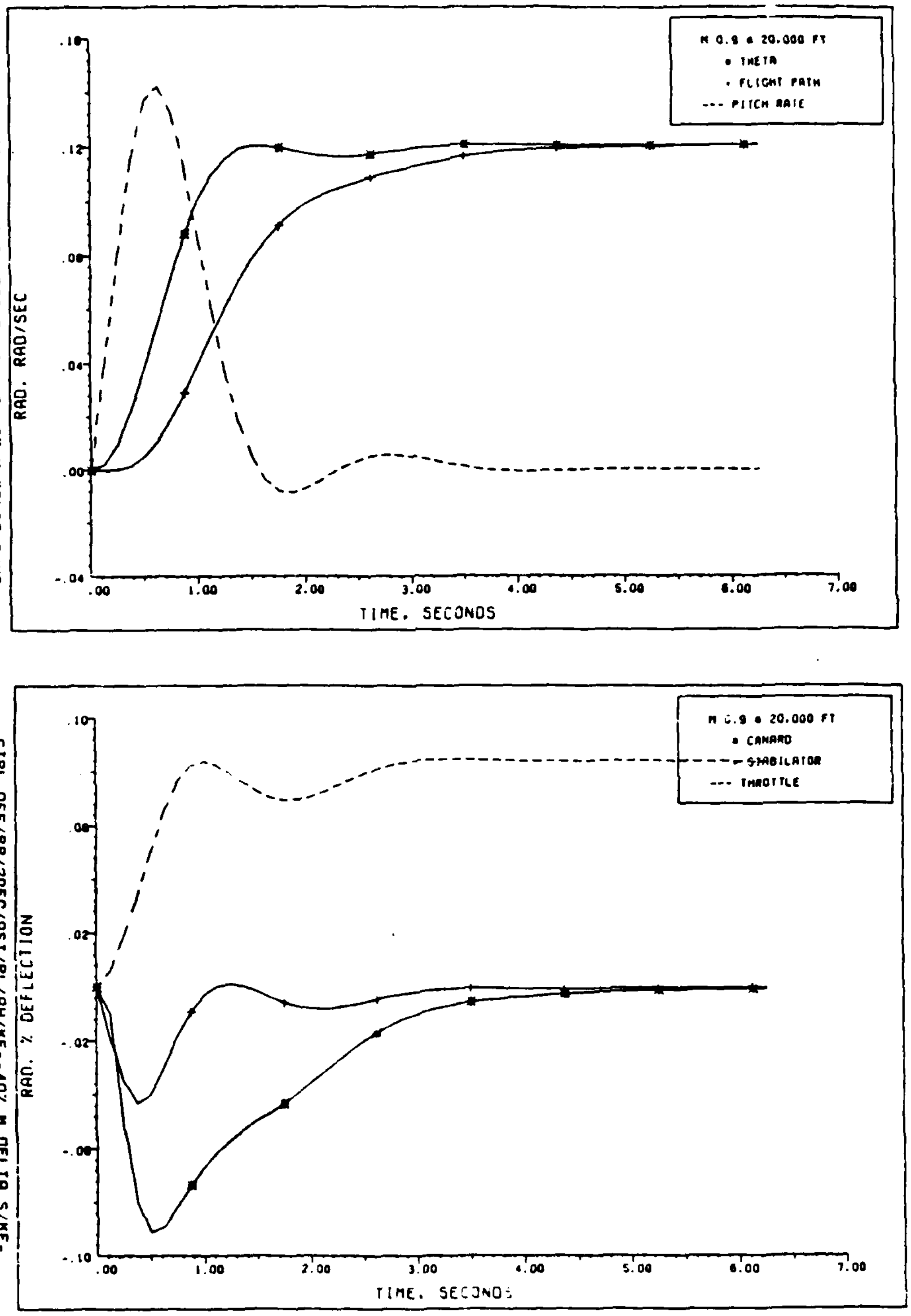


Fig. F.23. Aircraft Response with Actuators, Position Limits, Anti-Windup Compensation/Control Based on $\hat{\underline{x}}(t_i^-)$/ C-Tuned/100 Percent Decrease in M_{δ_C}

a. Aircraft responses; b. Control deflections

Appendix G: Notes on LTR Technique for Application to LQG/PI/KF System

In this appendix, some ideas which have been generated concerning LTR tuning and its application to the designs used in this thesis will be presented. No definite conclusions will be presented in this appendix; however, some suggestions which may be worth pursuing will be introduced.

At the onset of using the LTR technique in this thesis effort, Equation (4-4) was applied in a rather blind fashion. When results showed no improvement in the system robustness, the assumption was made that, perhaps, the LTR tuning technique was not applicable to the LQG/PI/KF controller structure which was implemented in the designs presented in Chapter V. Following the apparent failure of the LTR technique, the ad-hoc C-tuning method was attempted and system robustness was greatly enhanced. After a more detailed investigation of the LTR method (G1:G2), three conclusions have been reached concerning the above-mentioned results:

1. The fact that LTR tuning based on Equation (4-4) did not improve system robustness simply implies that the system robustness characteristics for the loop broken at the input are not desirable in the full state case.

Therefore, as q was increased, the filter-based system asymptotically approached what could have been poor robustness characteristics in the non-filter-based system.

2. The favorable results obtained using C-tuning is completely reasonable under existing LTR theory. As stated in Reference G.1, robustness enhancement can be obtained at any arbitrary break in the controller loop by injecting white noise into that point in the system. Therefore, the C-tuning roughly corresponds to LTR tuning applied with the system broken at the output. Based on the preceding, an explanation for the success of the LTR method applied at the output instead of the input of the system indicates that the full state controller displays better robustness characteristics in the variables of interest to us in this application than the system broken at the input, and it is these loop transmission characteristics that are recovered.

3. The above assumptions should be investigated using computer analysis of the frequency response of the system broken at both the input and output to verify that indeed the injection of white noise into the outputs of the design model is the proper form of the LTR tuning technique to be used in this particular flight control design. Once the proper point to regain system loop transmission robustness is established (perhaps not even the input or the output points, but some other point in the loop), the LTR

technique can be applied and structured singular value analysis could be used to establish the amount of recovery which is achieved as q is increased.

Based on the fact that what is referred to as C-tuning is really LTR tuning, it has been shown that, in fact, LTR tuning is applicable to a CGT/PI/KF control structure. The problem which remains to be solved is the forementioned task of determining the proper point in the loop to break the loop, or at least to show that the loop transmission robustness characteristics of the system broken at the output do indeed display favorable frequency response characteristics (i.e., bandwidth, crossover frequency slope, etc.).

Bibliography

1. Doyle, S. C. and G. Stein. "Robustness with Observers." IEEE Transactions on Automatic Control, Vol. AC-24, No. 4, August 1979, pp. 373-377.
2. Wall, J. Honeywell Systems and Research Center, Minneapolis MN. Personal interview, 25 November 1985.

Bibliography

1. Acker, B. H. Multivariable Output Control Law Design for the STOL/F-15 in Landing Configuration. MS thesis, GE/EE/85D-1, School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1985.

2. ASD Computer Center. CDC NOS User's Guide (Revision B). Wright-Patterson AFB OH: Requirements and Plans Branch, Aeronautical Systems Division, 1983.

3. Barfield, F. A. Multivariable Control Laws for the AFTI/F-16. MS thesis. Air Force Institute of Technology (AU), Wright-Patterson AFB OH, July 1983.

4. D'Azzo, J. J. and C. H. Houpis. Linear Control Systems Analysis and Design (Second Edition). New York: McGraw-Hill Book Company, 1981.

5. Doyle, J. C. "Multivariable Design Techniques Based on Singular Value of Generalizations of Classical Control Theory," Multivariable Control Theory Analysis and Design. Honeywell Systems & Research Center, Minneapolis, 3.1-3.14.

6. Etkin, B. Dynamics of Atmospheric Flight. New York: John Wiley & Sons, 1972.

7. Floyd, R. M. Design of Advanced Digital Flight Control Systems Via Command Generator Tracker (CGT) Synthesis Methods, Volumes 1 and 2. MS thesis. Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1980.

8. Hoh, R. H. et al. Proposed Mil Standards and Handbook--Flying Qualities of Air Vehicles, Volume 2. Report No. AFWAI-TR-82-3081, Vol. 2. Washington: Government Printing Office, 1982.

9. Howey, J. M. Robust Flight Controllers. MS thesis. Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1980.

10. Larimer, S. J. An Interactive Computer-Aided Design Program for Digital and Continuous System Analysis and Synthesis (TOTAL). MS thesis. Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1978.

11. Maybeck, P. S. Stochastic Models, Estimation, and Control, Volume 1. New York: Academic Press, 1979.

12. Maybeck, P. S. Stochastic Models, Estimation, and Control, Volume 3. New York: Academic Press, 1982.

13. Maybeck, P. S., R. M. Floyd and A. Moseley. Synthesis and Performance Evaluation Tools for CGT/PI Advanced Digital Flight Control Systems, IEEE 1983 National Aerospace and Electronics Conference (NAECON 1983), Dayton, 1259-1266 (May 1983).

14. Maybeck, P. S., W. Miller and J. M. Howey. Robustness Enhancement for LQG Digital Flight Controller Design, IEEE 1984 National Aerospace and Electronics Conference (NAECON 1984), Dayton, 1-14 (May 1984).

15. McMonagle, D. "Pilot Report: AFTI/F-16," Air Force Magazine, 68: 68-73 (April 1985).

16. Miller, W. Robust Multivariable Controller Design Via Implicit Model-Following Methods. MS thesis. Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1983.

17. Moseley, A. Design of Advanced Digital Flight Control Systems Via Command Generator Tracker (CGT) Techniques, Volumes 1 and 2. MS thesis. Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1982.

18. Papoulis, A. Probability, Random Variables, and Stochastic Processes (Second Edition). New York: McGraw-Hill, 1984.

19. Roskam, J. Flight Dynamics. Kansas: Roskam Aviation and Engineering, 1972.

20. Sheehan, K. A. Multivariable Control Law Design for Enhanced Air Combat Maneuvering: F-15/STOL Derivative Fighter. MS thesis, GE/EE/85D-38. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1985.

21. Sivan, Raphael and H. Kwakernaak. Linear Optimal Control Systems. New York: Wiley Interscience, 1972.

22. Stein, G. Multivariable Control Theory Analysis and Design. Honeywell Systems & Research Center, Minneapolis MN, pp. 5.1-5.9.

23. Strang, G. *Linear Algebra and Its Applications*. New York: Academic Press, 1980.

24. Vincent, J. H. "Direct Incorporation of Flying Qualities Criteria into Multivariable Flight Control Design," *Proceedings of the AIAA Guidance and Control Conference*, Seattle, 1-14 (August 20-22, 1983).

VITA

Robert Houston was born on 24 February 1961 in Dallas, Texas, He graduated from high school in Somerset, Texas in May 1979, and enlisted in the United States Air Force in July 1979. He attended Basic Military Training School and Security Specialist training at Lackland Air Force Base, Texas. Upon graduation from technical training he attended Air Base Ground Defense combat school at Camp Bullis, Texas. From December 1979 to June 1980, he served as a close-in security sentry guarding B-52, KC-135, and EC-135 alert aircraft at Ellsworth Air Force Base, South Dakota. In June 1980 after attaining the rank of Airman First Class, he was selected for Airman Scholarship and Commissioning program and enrolled in the Reserve Officers Training Corps at the University of Texas at Austin. Upon graduation with a degree in Electrical Engineering in May 1984, he was commissioned a second lieutenant in the United States Air Force Reserve. He entered the School of Engineering of the Air Force Institute of Technology in May 1984 and has pursued the Guidance and Control curriculum of the Electrical Engineering Department.

Permanent address: P. O. Box 6
Von Ormy, Texas 78073

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION: UNCLASSIFIED

1b. RESTRICTIVE MARKINGS:

2a. SECURITY CLASSIFICATION AUTHORITY:

2b. DECLASSIFICATION/DOWNGRADING SCHEDULE:

3. DISTRIBUTION/AVAILABILITY OF REPORT: Approved for public release; distribution unlimited

4. PERFORMING ORGANIZATION REPORT NUMBER(S): AFIT/GE/ENG/85D-21

5. MONITORING ORGANIZATION REPORT NUMBER(S):

6a. NAME OF PERFORMING ORGANIZATION: School of Engineering

6b. OFFICE SYMBOL *(If applicable)*: AFIT/ENG

7a. NAME OF MONITORING ORGANIZATION:

6c. ADDRESS *(City, State and ZIP Code)*:
Air Force Institute of Technology
Wright-Patterson AFB, OH 45433

7b. ADDRESS *(City, State and ZIP Code)*:

8a. NAME OF FUNDING/SPONSORING ORGANIZATION: Flight Dynamics Lab

8b. OFFICE SYMBOL *(If applicable)*: AFWAL/FIGL

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER:

8c. ADDRESS *(City, State and ZIP Code)*: Wright-Patterson AFB, OH 45433

10. SOURCE OF FUNDING NOS.

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO.

11. TITLE *(Include Security Classification)*: See Block 19

12. PERSONAL AUTHOR(S): Robert A. Houston, B.S.E.E., 2Lt, USAF

13a. TYPE OF REPORT: MS Thesis

13b. TIME COVERED: FROM ________ TO ________

14. DATE OF REPORT *(Yr., Mo., Day)*: 1985 December

15. PAGE COUNT: 242

16. SUPPLEMENTARY NOTATION:

17. COSATI CODES

FIELD	GROUP	SUB. GR.
01	03	

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*: Optimal Control, Model-Following Control, Robustness, Proportional-Integral Control, STOL Aircraft Control, Flight Control Systems

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

Title: An LQG Up-and-Away Flight Control Design
for the STOL F-15 Aircraft

Thesis Chairman: Dr. Peter S. Maybeck

Approved for public release: IAW AFR 190-17.
LYNN E. WOLAVER 16 JAN 86
Dean for Research and Professional Development
Air Force Institute of Technology ~~([illegible])~~
Wright-Patterson AFB OH 45433

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT: UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS ☐

21. ABSTRACT SECURITY CLASSIFICATION: UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL: Peter S. Maybeck, B.S., Ph.D.

22b. TELEPHONE NUMBER *(Include Area Code)*: 513-255-3576

22c. OFFICE SYMBOL: AFIT/ENG

DD FORM 1473, 83 APR EDITION OF 1 JAN 73 IS OBSOLETE.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

A robust controller for the STOL F-15 aircraft is developed using the LQG/LTR (linear system model, quadratic cost, gaussian models of uncertainty used for controller synthesis, with loop transfer recovery techniques of tuning the filter in the loop for control robustness enhancement) methods. Full state feedback controllers are synthesized using CGT/PI (Command Generator Tracking feedforward compensator to provide direct incorporation of flying qualities into the design process, with proportional plus integral feedback control) synthesis, using implicit model following techniques to improve full state robustness characteristics. Finally, a Kalman filter is used to replace the unrealistic assumption of full state availability with estimated states, using a LTR scheme to recover as much full state robustness characteristics as possible. (THESIS, [illegible]

END
FILMED

3-86

DTIC