REPORT DOCUMENTATION PAGE	Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	14.Jan.03	THESIS

4. TITLE AND SUBTITLE
MODEL PREDICTIVE CONTROL FOR TEMINAL AREA ENERGY MANAGEMENT AND APPROACH AND LANDING OF A REUSABLE LAUNCH VEHICLE

5. FUNDING NUMBERS

6. AUTHOR(S)
1ST LT LEPOME ROBERT C II

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

8. PERFORMING ORGANIZATION REPORT NUMBER
CI02-808

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
THE DEPARTMENT OF THE AIR FORCE
AFIT/CIA, BLDG 125
2950 P STREET
WPAFB OH 45433

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION AVAILABILITY STATEMENT
Unlimited distribution
In Accordance With AFI 35-205/AFIT Sup 1

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

14. SUBJECT TERMS	15. NUMBER OF PAGES	16. PRICE CODE
	236	

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT

Model Predictive Control for Terminal Area Energy Management and Approach and Landing for a Reusable Launch Vehicle

by

Robert C. LePome II

B.S. Astronautical Engineering
B.S. Mathematics
United States Air Force Academy, 2000

SUBMITTED TO THE DEPARTMENT OF AERONAUTICS AND ASTRONAUTICS IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE IN AERONAUTICS AND ASTRONAUTICS
AT THE
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

JUNE 2002

Signature of Author Robert C. LePome II
Department of Aeronautics and Astronautics
May 10, 2002

Certified by
Piero A. Miotto
The Charles Stark Draper Laboratory, Inc.
Technical Supervisor

Certified by
Wallace E. Vander Velde
Professor of Aeronautics and Astronautics
Thesis Advisor

Accepted by
Wallace E. Vander Velde
Professor of Aeronautics and Astronautics
Chair, Committee on Graduate Students

Model Predictive Control for Terminal Area Energy Management and Approach and Landing for a Reusable Launch Vehicle

by

Robert C. LePome II

Submitted to the Department of Aeronautics and Astronautics on
May 10, 2002, in partial fulfillment of the requirements for the
Degree of Master of Science in Aeronautics and Astronautics

Abstract

The space industry plans to develop new reusable launch vehicles. The new vehicles will need advanced, new guidance and control systems. Since 1996 Draper Laboratory has been developing the next generation guidance and control for reusable launch vehicles in which guidance and control is integrated into one correlated system.

Draper's research of integrated guidance and control originated with a single loop multivariable control scheme using time-invariant linear quadratic regulator theory. The research has since evolved into the use of model predictive control theory. The main focus of this thesis is the theory and design of model predictive control for entry of aerospace vehicles. The goal is to develop design criteria and guidelines explaining how to select the model predictive control parameters: prediction horizon, simulation rates, and weighting matrices. A secondary goal is to tightly couple an onboard trajectory generation algorithm with the model predictive controller to improve tracking performance and robustness.

Favorable tracking is achieved through two model predictive control architectures, which are discussed. The first architecture has an inner loop stability augmentation system with model predictive control used as an outer loop. The second architecture replaces the inner and outer loops with a single model predictive controller. The two architectures demonstrate the flexibility of model predictive control to adapt to new vehicles; the model predictive control may be used to augment an existing inner loop or may be used as a stand-alone controller. The design focuses primarily on the architecture without a stability augmentation system.

Technical Supervisor: Piero A Miotto
Title: Technical Supervisor, The Charles Stark Draper Laboratory, Inc.

Thesis Advisor: Wallace E. Vander Velde
Title: Professor of Aeronautics and Astronautics

[This page intentionally left blank]

THE VIEWS EXPRESSED IN THIS ARTICLE ARE THOSE OF THE AUTHOR AND DO NOT REFLECT THE OFFICIAL POLICY OR POSITION OF THE UNITED STATES, DEPARTMENT OF DEFENSE, OR THE U.S. GOVERNMENT

Table of Contents

[This page intentionally left blank]

List of Figures

List of Tables

[This page intentionally left blank]

List of Symbols

A,B,C,D ..... State space matrices

B_d ..... Input matrix for unmeasured disturbances

B_u ..... Input matrix for control

B_v ..... Input matrix for measured disturbances

BW ..... Bandwidth

b ..... Wingspan or body reference frame

c ..... Mean aerodynamic chord

D_d ..... Feed-forward output matrix for unmeasured disturbances

D_v ..... Feed-forward output matrix for measured disturbances

d ..... Unmeasured disturbance

H_d ..... Unmeasured prediction matrix

H_v ..... Measured prediction matrix

h ..... Altitude state or local horizon reference frame

I_P ..... Column matrix of identity matrices

i ..... Inertial reference frame

J ..... Cost function

J_M ..... Matrix simplifying the control

K_1 ..... Lower triangular matrix of identity matrices

K_d ..... Unmeasured disturbance gain matrix

K_{du} ..... MPC gain matrix

K_r ..... State reference gain matrix

K_T ..... Control reference gain matrix

K_u ..... Previous control gain matrix

K_v ..... Measured disturbance gain matrix

K_x ..... State gain matrix

M ..... Control horizon

nd ..... Number of unmeasured disturbances

nu ..... Number of inputs

nv ..... Number of measured disturbances

nx ..... Number of states

ny ..... Number of outputs

P ..... Roll rate state or prediction horizon

Q ..... Pitch rate state

$\bar{q}$ ..... Dynamic pressure

$\Re$ ..... Set of real numbers (superscripts indicate real numbers in matrices of the given dimension)

R ..... Yaw rate state

r ..... Reference vector projected in time

S_u ..... Control prediction matrix

S_{u1} ..... Previous control prediction matrix

S_x ..... State prediction matrix

s ..... Planform area or stability reference frame

t ..... Time

u ..... Control vector

u_T ..... Target control vector

V ..... Velocity state

v ..... Measured disturbance or velocity reference frame

W_u ..... Control weighting matrix

W_Δ ..... Change in control weighting matrix

W_y ..... State weighting matrix

x ..... Downrange state or state vector

y ..... Crossrange state or predicted output

z ..... MPC control set

α ..... Angle of attack state

β ..... Sideslip angle state

χ ..... Heading angle state

δ_a ..... Aileron position

δ_e ..... Elevon position

δ_r ..... Rudder position

δ_{sb} ..... Speed brake position

Φ_M ..... Phase margin

γ ..... Flight path angle state

μ ..... Bank angle state

θ ..... Pitch angle state

τ ..... Time delay

ω_p ..... Frequency at which the phase margin is found

* ..... Indicates an optimal parameter

$\in$ ..... Slack variable for constrained simulation or indicates an element of a set

[This page intentionally left blank]

List of Abbreviations

A/L..... Approach and Landing

CSDL..... Charles Stark Draper Laboratory

EOM..... Equations of Motion

G&C ..... Guidance and Control

GN&C..... Guidance, Navigation, and Control

HAC..... Heading Alignment Cone

IG&C ..... Integrated Guidance and Control

L/D ..... Lift to Drag Ratio

LPV ..... Linear Parameter Varying

LQR..... Linear Quadratic Regulator

LTI..... Linear Time Invariant

MIMO ..... Multiple Input Multiple Output

MPC ..... Model Predictive Control

MPC_ALL..... Model Predictive Control Architecture without Inner Loop SAS

MPC_SAS ..... Model Predictive Control Architecture with Inner Loop SAS

NASA ..... National Aeronautics and Space Administration

NGGC ..... Next Generation Guidance and Control

OSC ..... Orbital Sciences Corporation

RLV..... Reusable Launch Vehicle

SAS..... Stability Augmentation System

SISO..... Single Input Single Output

TAEM ..... Terminal Area Energy Management

[This page intentionally left blank]

Chapter 1

Introduction

The aerospace industry and the National Aeronautics & Space Administration (NASA) plan to develop new reusable launch vehicles in the future. The Lockheed Martin X-33 and the Orbital Sciences X-34 are two examples of endeavors in the recent past. The new vehicles will require new guidance and control (G&C) systems to operate them. Previous guidance and control systems have limited flexibility. They use predefined trajectories for both boost and entry segments of flight. Predefined trajectories are mission specific and require extensive preflight design for every flight. Since the trajectories are predefined, the vehicle is vulnerable to changes in flight conditions after the design. High winds, slight mission changes, fuel consumption variations, and atmospheric temperature changes are just a few examples of possible changes in flight conditions.

The guidance and control systems from previous vehicles such as the space shuttle were designed separately. The result is a poorly correlated response where the guidance and control react to each other's contribution instead of working in a fully collaborated effort. Careful design is needed to avoid poor vehicle response. Finally, in the event of a subsystem failure requiring an abort in the launch or entry flight phase, predefined trajectories tend to limit the available abort options [Ref. 1].

Current design objectives are to make access to space a routine event, to achieve airplane-like operations, and to reduce the cost of space flight. Future G&C systems must reduce the mission specific labor such as reducing the number of I-loads needed for each launch. This reduction is needed to standardize launch and entry procedures to meet the above objectives. Mission reliability must be increased, and the required flight support for routine launches must be minimized [Ref. 2].

In response to the design objectives, the Charles Stark Draper Laboratory (CSDL) is developing guidance and control approaches for boost, entry, and landing of reusable launch vehicles (RLV). Draper's next generation guidance and control (NGGC) research focuses on three key fields: autonomous abort technology, onboard trajectory generation, and integrated guidance and control (IG&C). Onboard operations

significantly reduce preflight design and mission support. It increases reliability because it makes space launch and entry more routine with fewer I-loads and calls for a G&C law that is not mission specific. An integrated guidance and control design using modern control in past research has shown great promise for creating an onboard design. In addition, onboard trajectory generation and onboard flight control is needed to make the abort technology a reality. As flight conditions change in flight as prescribed by an abort scenario, onboard trajectory generation allows the control system to recalculate optimal control inputs for a new and more accurate trajectory. Recalculating the inputs is critical because the abort obviously outdates the precalculated control inputs as the state of the vehicle might vary quite significantly from a nominal flight. In a launch scenario, an abort may save the vehicle and payload from destruction allowing for another launch attempt. In an entry scenario, an abort may require a landing on an alternate runway within the vehicle's range. The abort capability could then save the vehicle, thereby increasing mission reliability. A fixed actuator may cause an abort on entry. A correlated onboard G&C system allowing reconfigurability is needed in such an example.

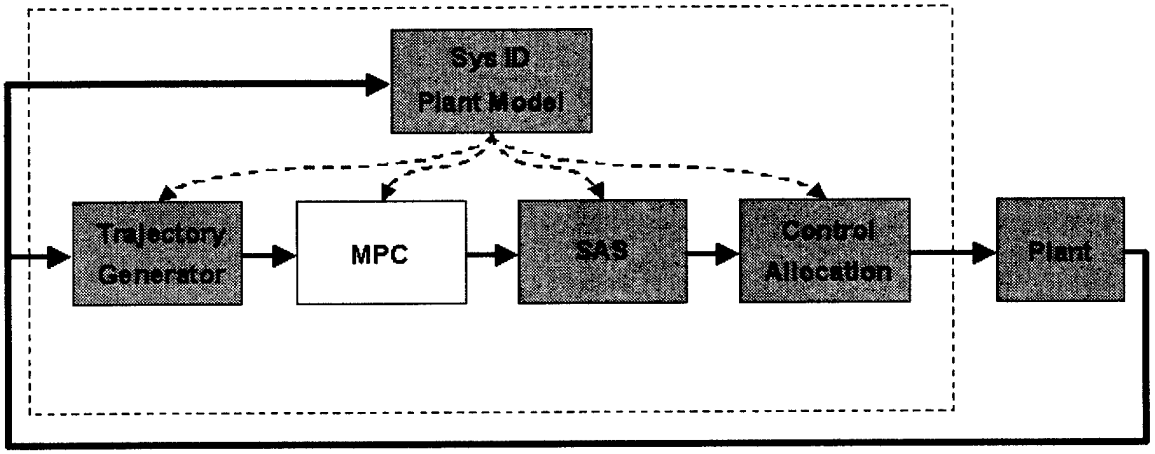

Fig. 1 Next Generation Guidance and Control Goal

Fig. 1 shows a block diagram of Draper's long term NGGC goal. The trajectory generation has undergone multiple design iterations originating from a longitudinal dynamics only design for approach and landing (A/L). Lateral dynamics and the subsonic portion of the terminal area energy management (TAEM) corridor were added next. Current trajectory generation research is developing optimal trajectories

throughout the entire TAEM corridor for the supersonic, subsonic, and A/L flight regimes.

The control allocation inputs moment commands from a controller and vehicle state information from sensors. It then issues thruster commands for high altitude flight. As the vehicle descends through the atmosphere, the air density increases and the aerosurfaces begin to influence the vehicle's control. During the transition from high altitude to low altitude flight a blend of thruster and aerosurface commands is issued by the control allocation. For low altitude flight, the aerosurfaces are used solely for control. The control allocation can further be used when reconfiguration is required.

The MPC and Stability Augmentation System (SAS) blocks describe the flight control. Integrated guidance and control research has been worked on in parallel with trajectory generation and control allocation. Chomel [1998] developed a longitudinal system for the approach and landing flight segment of the X-34 [Ref. 3]. The research concluded that modern control could be used for RLVs to improve landing performance and to simplify the gain design. Research in 1999 then followed again in an MIT master's thesis, expanding the flight envelope. Lateral dynamics and state integrators on key trajectory states were added to a linear quadratic regulator (LQR) controller. Supersonic and subsonic TAEM flight augmented the approach and landing phase of the trajectory. Again the X-34 was the chosen testbed. The research showed that a relatively simple LQR flight controller provided comparable performance to the X-34 classical algorithm used as the baseline case [Ref. 1]. Both research conclusions for the flight control have motivated continued research in Draper's NGGC ultimate goal.

The focus of this thesis is the design of Model Predictive Controllers for RLVs with correlated guidance and control systems. Model Predictive Control uses an internal plant model to predict future state outputs over a given time horizon. The controller tracks a trajectory over the time horizon by generating control inputs that minimize the error between the projected state outputs and a reference signal.

Two controller architectures are considered. The first architecture has an inner loop LQR SAS with MPC used as an outer loop (shown in Fig. 1). The second architecture replaces the inner and outer loops with a single MPC controller. Both architectures simplify the problem and take advantage of prior research by removing the control allocation from the problem. The flight control issues aerosurface commands directly to the plant. The trajectory is assumed to be a given from a guidance subsystem.

1.1 Previous MPC Research

Some of the earliest applications of MPC have been in chemical engineering, dating as far back as the 1970s [Ref. 4]. The plants were well known and constant for systems with relatively slow dynamics. In 1994, Berlin and Frank [Ref. 5] used MPC to control a 3-tank system applying multiple input multiple output (MIMO) control with two inputs and two outputs. Berlin and Frank achieved good tracking performance while considering multiple disturbances. With recent advances in computing power, MPC has been able to be applied to aerospace vehicles. MPC can now be applied to systems with nonlinear dynamics. Hauser and Jadbabaie have successfully applied MPC to a thrust vectored flying wing for forward flight at the California Institute of Technology [Ref. 6]. Additional applications of MPC have been made recently to the aerospace industry. Shearer applied MPC to controlling an F-16 fighter aircraft model. A linearized system was used to approximate the aircraft's nonlinear dynamics with favorable results [Ref. 7]. Because of increased computing rates of today's computer's, MPC research is gaining popularity in the aerospace industry.

1.2 Optimal Control Problem Definition

The optimal control problem presented in this thesis is of the following form:

Find an admissible optimal control (u^*) that causes the system

$$\dot{x}(t) = a(x(t), u(t), t) \tag{1}$$

to follow an admissible optimal trajectory (x^*) that minimizes the performance function

$$J = h(x(t_f), t_f) + \int_{t_0}^{t_f} g(x(t), u(t), t)dt \tag{2}$$

where $a, h, \text{and } g$ are scalar functions. t_f is final time, t_0 is initial time, and t is time.

A few restrictions, in general, accompany the definition described by equations (1) and (2). First, an optimal control may not exist for a given system, particularly when constraints are introduced. It may be impossible to find an admissible control value that

follows an admissible trajectory. Second, if an admissible optimal control exists, a unique solution is not guaranteed. Consider the simplified example of a sine wave, $f(x)=\sin(x)$. $x=3\pi/2$ and $x=7\pi/2$ give a minimum value of –1 so multiple control values lead to a single minimum cost. Such a situation may occur when the performance parameter has multiple global minimums. Last, the optimal control approach always calculates the global minimum of the cost, not a local minimum. Global minimum means that all admissible control values that generate admissible trajectories result in a higher (or equal if the solution is not unique) cost than the optimal admissible control. Mathematically the global minimum is written

$$\begin{aligned} J^* &\triangleq h(x^*(t_f),t_f)+\int_{t_0}^{t_f} g(x^*(t),u^*(t),t)\,dt \\ &\le h(x(t_f),t_f)+\int_{t_0}^{t_f} g(x(t),u(t),t)\,dt \quad \forall\, u\in U \text{ that makes } x\in X \end{aligned} \tag{3}$$

Where $h,$ and g are scalar functions. t_f is final time, t_0 is initial time, and t is time. x and u are the state and control vectors, respectively. The asterisk designates the optimal values.

The system used in the optimal control scheme may be classified by linearity and if it varies with time.

	Nonlinear	**Linear**
Time-variant	$\dot{x}(t)=a(x(t),u(t),t)$	$\dot{x}(t)=A(t)x(t)+B(t)u(t)$
Time-invariant	$\dot{x}(t)=a(x(t),u(t))$	$\dot{x}(t)=Ax(t)+Bu(t)$

Tab. 1 Optimal Control System Classification

A(t) and B(t) are matrices of size $nx \times nx$ and $nx \times nu$, respectively with time-varying elements. A and B are constant matrices of size $nx \times nx$ and $nx \times nu$ [Ref. 8]. nx is the number of states and nu is the number of control inputs.

The MPC optimal control problem solved in this thesis is linear, and time-invariant. The continuous form $\dot{x}(t) = Ax(t) + Bu(t)$ is converted to the discrete form $\dot{x}(k+1) = Ax(k) + Bu(k)$ and is used to minimize the performance parameter in matrix form

$$\begin{aligned} J = & \min_{\underline{z}(k) \in R^{(M^* nu) \times 1}} \left[\underline{u}(k) - \underline{u}_T(k)\right]^T W_u \left[\underline{u}(k) - \underline{u}_T(k)\right] \\ & + \underline{\Delta u}^T(k) W_\Delta \underline{\Delta u}(k) + \left[\underline{y}(k) - \underline{r}(k)\right]^T W_y \left[\underline{y}(k) - \underline{r}(k)\right] \end{aligned} \quad (4)$$

This cost function places a penalty on deviations of the control input from the reference control input, changes in control from the previous control value, and state tracking errors. The cost function is more closely examined in section 2.1.3.1.

1.3 Thesis Objective

The focus of this research is to develop an integrated guidance and control algorithm and procedures for designing the algorithm using model predictive control for the entry of RLVs in the subsonic TAEM region and A/L. Lateral and longitudinal dynamics are controlled in a multivariable controller. The Orbital Sciences X-34 technology demonstrator is the testbed for the research, however, the principles discussed may be applied to a variety of vehicles. To maintain generality and flexibility, the research stresses the concepts and design criteria needed to design an MPC controller for any aerospace vehicle. In addition, the intent of the research is to investigate the flexibility of MPC to be used to augment an existing inner loop or to take the role of the entire controller. The advantages of MPC are briefly identified and exploited to demonstrate constraint handling, show an aptitude for reconfigurability, and to obtain favorable tracking performance.

1.4 Thesis Overview

Chapter 1 presents an introduction to the thesis. It identifies the need for developing the next generation guidance and control. Then it describes Draper Laboratory's plan for creating NGGC and shows how this thesis contributes to the overall goal. Chapter 2 provides the reader with an introduction to the concepts and a section hi-lighting some of the potential benefits and problems associated with MPC. Additionally, the mathematical background of MPC is discussed for the constrained and unconstrained cases. Chapter 3 narrows the focus of the thesis to the design of the MPC applied to the specific problem of controlling the X-34. The X-34 vehicle is described and some preliminary design considerations are described such as state and architecture selection. Chapter 4 shows key design criteria one should follow when designing an MPC controller. It uses a simplified example using only the vehicle's longitudinal dynamics. In Chapter 5 the defined design criteria are applied to the full nonlinear simulation with both longitudinal and lateral dynamics. Chapter 6 includes the results of using the designed controller in simulations for two trajectories. Finally, Chapter 7 draws conclusions based on the research and makes suggestions for future research.

[This page intentionally left blank]

Chapter 2

Model Predictive Control Theory

Model Predictive Control theory is a model base optimal control technique. Every MPC system has the same general architecture composed of an internal plant model and an optimizer. Fig. 2 is a block diagram showing this structure. The MPC controller has an internal model of the plant dynamics, which it uses to predict future outputs of the actual plant. The MPC's optimizer differences the predicted outputs with a reference trajectory. It takes that error signal, the past input to the actual plant, and any constraints imposed on the system and calculates a set of optimal, future inputs for the actual plant according to a defined cost function. The set of future control inputs are the inputs that will drive the output to the desired reference set points. A variety of cost functions may be used; however, the system described by equation (4) is used in this research.

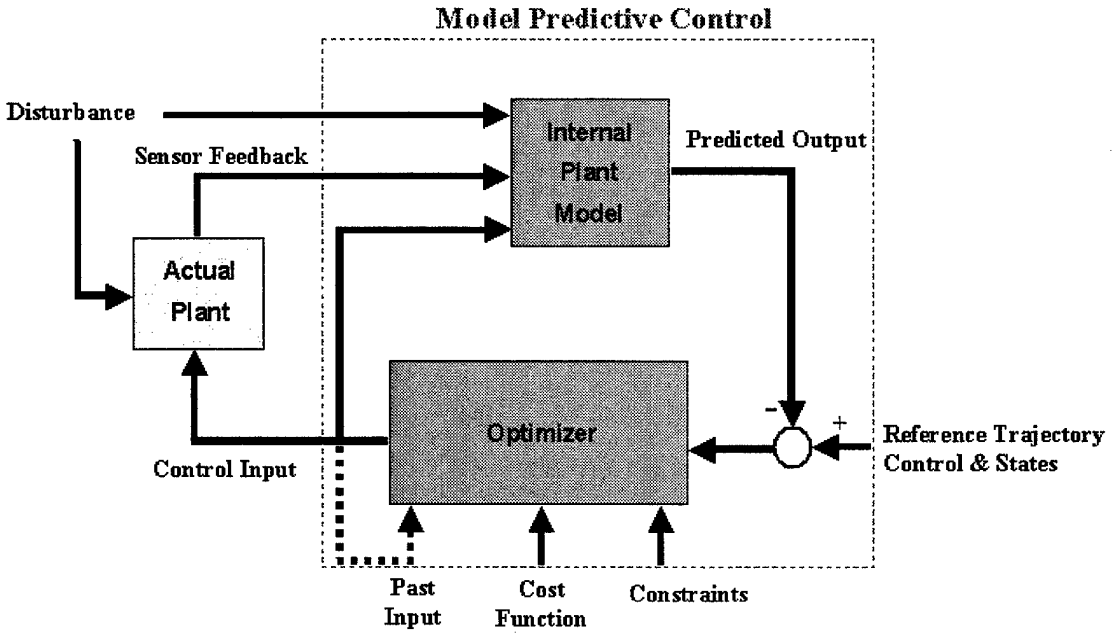

Fig. 2 General MPC Architecture

Fig. 3 and Fig. 4 illustrate MPC theory in greater detail through a plot of the predicted output and the future inputs. The vertical axis represents the current time with the

shaded and unshaded sections signifying the past and future time, respectively. The horizontal dashed line in Fig. 3 shows the desired reference output projected P steps into the future from time t to time t+P. The open circles represent the P predicted outputs Δt apart. The horizontal dashed lines in Fig. 4 represent hard constraints that the input values may not exceed. The output constraints in Fig. 3 and the input reference in Fig. 4 are omitted from the figures [Ref. 9].

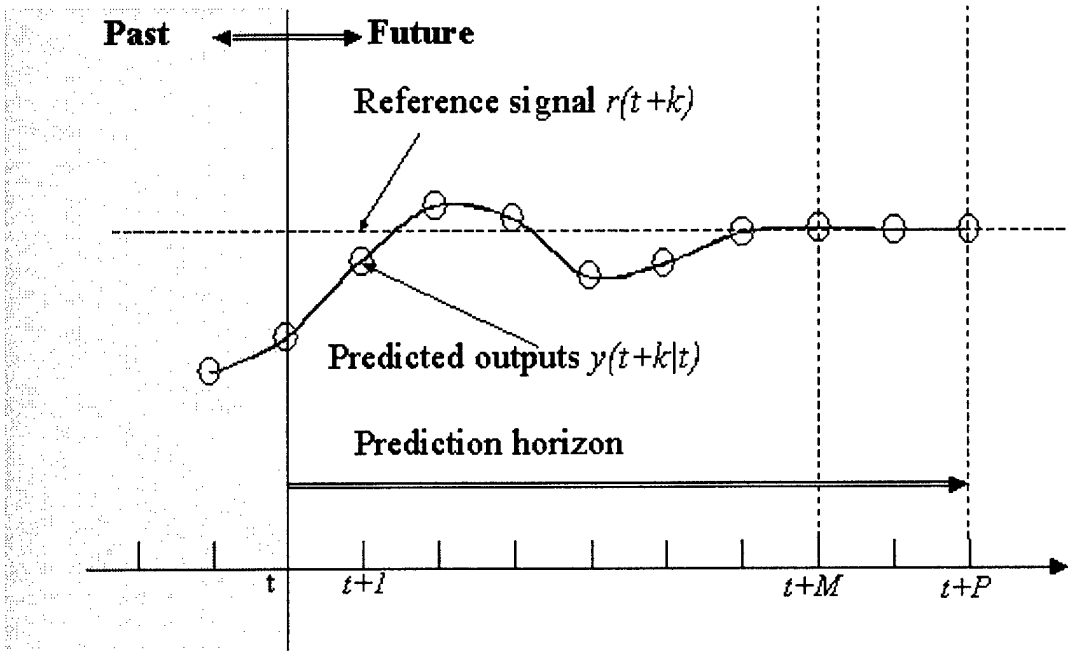

Fig. 3 MPC Output Problem Definition

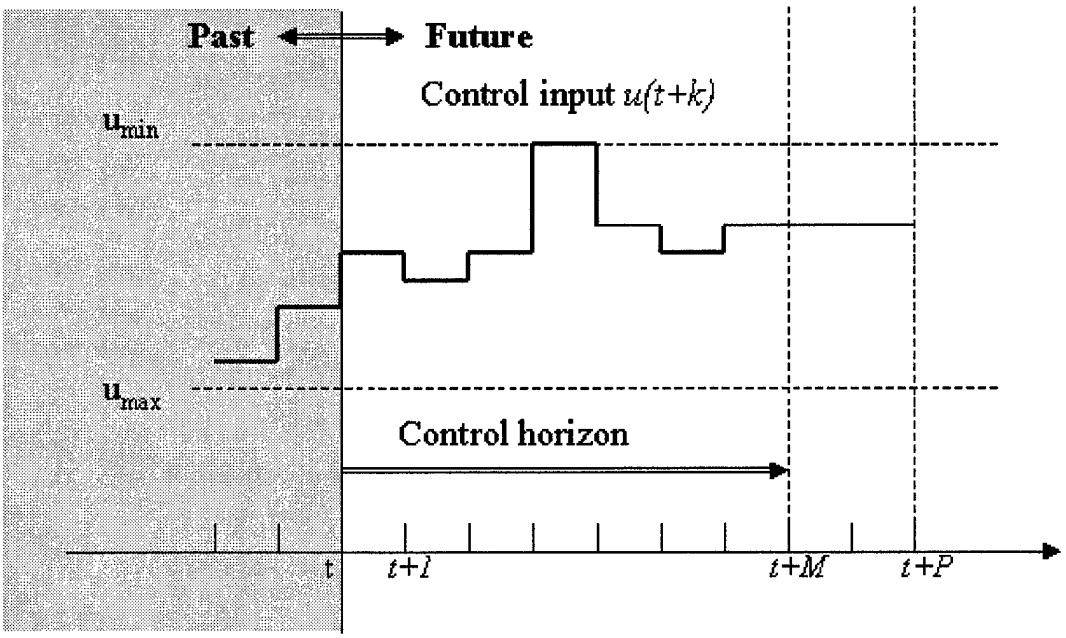

Fig. 4 MPC Input Problem Definition

The following sections describe the MPC structure more explicitly. The discussion starts with an explanation of the prediction concept followed by a description of how the internal plant model and the optimizer make predictions and find optimal control inputs. Finally, potential benefits and problems with using MPC are assessed.

2.1 MPC Design Components

2.1.1 Prediction & Control Horizon Description

The *Prediction Horizon (P)* is the number of predictions in the future the MPC uses to gain understanding of the effects of the control inputs on the system. Each prediction is equally spaced in time as determined by the designer. The *Control Horizon (M)* is the number of free movements for the control inputs calculated in the future. A free movement is the ability of the control to assume a new value. The respective control inputs are held constant for a length of time of Δt until the next sampling prediction. The control horizon may take on two forms, an integer or a vector. An integer value indicates either the number of free control movements in the prediction horizon or the number of base functions used to describe the control. The context of the problem

dictates which interpretation the designer is using. Using base functions calls for the control to be described by a linear combination of predefined functions. The next section gives a more rigorous definition and example of base functions. Both interpretations of the integer control horizon value represent the number of degrees of freedom available to describe the control over the horizon. When the control horizon is an integer, it must be less than or equal to the prediction horizon. If both horizons are equal, this indicates that the control is allowed to take on a different value at every prediction calculation. If, on the other hand, $M < P$, then the first M control inputs are allowed to vary, and subsequent control values are held equal to the last input for the remaining prediction horizon. Fig. 4 shows an example of an integer value for the control horizon with $M < P$. Note how the control input is allowed to vary for the first M moves and is held constant thereafter. A plant with a time delay gives reason to selecting a control horizon less than the prediction horizon as it requires additional prediction samples to see the effects of the M control values.

When the control horizon is represented by a vector, the elements of the vector indicate which control inputs must be equal. The elements must also sum to P. For example, $P = 9$ and $M = [2\ 4\ 3]$ means that the first two control values must be equal. The next four values are equal to each other but may be different from the previous two values. Similarly, the last three moves must be equal, but not necessarily the same as the previous two or four values. This strategy is called blocking and is demonstrated in Fig. 5.

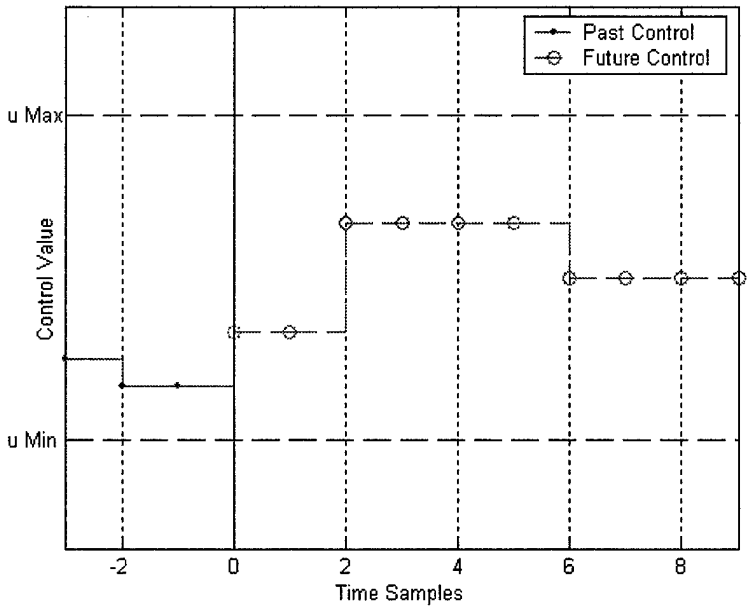

Fig. 5 Control Horizon with Blocking

Blocking may be advantageous when otherwise M << P because the final control value would be used for the majority of the prediction. For situations when M is an integer and M << P, each control value acts for one time sample in the optimization except for the final value, acting for P – M time samples. In this case, it may be beneficial to use blocking to distribute the length of time each control input is acting [Ref. 9].

For controllers that run at a faster rate than the prediction rate, the MPC controller calculates a set of M control inputs for its prediction, but only applies the first control value at the current time. It then discards the remaining calculated control values and calculates an entirely new set in the next iteration. In the next iteration, MPC has the benefit of seeing the effects of applying the first control value and the benefit of looking one time sample farther into the future and than it did in the previous iteration. On the other hand, for controllers that run at a slower rate than the prediction rate, multiple control inputs may be used. For example, if MPC is used as an outer loop to augment an existing inner loop, the outer loop may run at 2 Hz but predictions and hence control values may be made at a rate of 10 Hz. Instead of the controller using the first control value for 0.5 seconds and discarding the rest, the controller could use the first 5 control values implementing them at 0.1-second intervals and discard the remaining control values.

Although MPC discards most calculated control inputs, it must optimize over the entire prediction horizon because the current control value applied will likely depend on set point changes in the future. If a constraint is imposed affecting the future, the MPC must adjust the applied control at the current time to meet that constraint. In addition, for non-minimum phase plants (known for their characteristic initial inverse response), the MPC needs to know the short and long term effects of a given control input, so optimizing over the full prediction horizon is critical. Similarly, plants with time delays require additional predictions to fully understand the impact of the applied control [Ref. 9].

2.1.2 Internal Model

The internal plant model is essential to model predictive control because it is the mechanism used to estimate the future output. The MPC uses the internal model to simulate how the actual plant will react to the MPC generated input signal. The input signal is calculated based on the internal plant dynamics, the vehicle's current state, previous control value, and the future state and control reference values. By knowing how the actual plant reacts, the MPC can then select the optimal control value that will minimize errors throughout the entire prediction horizon. Compared to other optimal controllers such as LQR, MPC accepts a greater error in tracking at the current time, if by the end of the prediction horizon the overall error is reduced as a result of accepting the early error.

The internal model may be linear or nonlinear, time varying or time-invariant. However, the more the internal model accurately represents the actual model the more accurate the predictions and hence the control will be. For nonlinear plants in aerospace applications, a nonlinear internal model may provide better accuracy particularly when performing maneuvers exercising the nonlinear dynamics. This greater accuracy, however, comes at the cost of additional computation time. Full state or partial state feedback may be employed. If partial state feedback is used, a state estimator is required. Clearly if the state estimator and/or the plant model is very inaccurate, the MPC will issue poor control commands resulting in poor performance. Robustness to plant uncertainty may be a design and implementation issue when perfect plant knowledge is not assumed.

2.1.3 Optimizer

The optimizer calculates the optimal control inputs for the actual plant by minimizing the cost function. When the system is unconstrained a closed form mathematical solution is derived. However, when constraints are present, the optimizer numerically finds the inputs minimizing the constrained cost function.

2.1.3.1 MPC Optimizer Formulation & Cost Function

The purpose of this section is to present and discuss a standard formulation of the MPC control problem. Consider the discrete linear time-invariant (LTI) plant with state and output equations:

$$x(k+1) = Ax(k) + B_u u(k) + B_v v(k) + B_d d(k) \quad (5)$$

$$y(k) = Cx(k) + D_v v(k) + D_d d(k) \quad (6)$$

where $x(k) \in \Re^{nx\times 1}$ is a vector of states, $u(k) \in \Re^{nu\times 1}$ is a vector of inputs, $y(k) \in \Re^{ny\times 1}$ is a vector of outputs, $v(k) \in \Re^{nv\times 1}$ is a vector of measured disturbances, and $d(k) \in \Re^{nd\times 1}$ is a vector of unmeasured disturbances.

nx is the number of states, nu is the number of control inputs, ny is the number of outputs, nv is the number of measured disturbances, and nd is the number of unmeasured disturbances. A is the state matrix, and B_u, B_v, and B_d are matrices describing the effects of u(k), v(k), and d(k) respectively. C is the output matrix, and D_v, and D_d are feed-forward matrices to the output for the measured and unmeasured disturbances respectively.

The model predictive control is based on the solution of the following optimization problem:

$$J=\sum_{i=0}^{P-1}\left[u(k+i)-u_T(k+i)\right]^T w_u(i)\left[u(k+i)-u_T(k+i)\right]+\sum_{i=0}^{M-1}\Delta u(k+i)^T w_\Delta(i)\Delta u(k+i)$$
$$+\sum_{i=0}^{P-1}\left[y(k+i)-r(k+i)\right]^T w_y(i)\left[y(k+i)-r(k+i)\right] \qquad (7)$$

where $u(k+i)$ is the predicted control. $u_T(k+i)$ is the projected target reference control value. $y(k+i)$ is the predicted output. $r(k+i)$ is the projected reference output, and $w_u(i)$, $w_\Delta(i)$, and $w_y(i)$ are weighting variables for each term. P and M are the prediction and control horizons.

Equation (7) may be solved with and without constraints. The optimization parameter $\underline{z}(k)$ is related to the variation of the input variables through the following relation:

$$\underline{\Delta u}(k)=J_M\,\underline{z}(k) \qquad \underline{\Delta u}(k)=\underline{u}(k)-\underline{u}(k-1)$$
$$J_M\in\Re^{(P*nu)\cdot(M*nu)} \qquad (8)$$

where the underscore notation signifies a vector of vectors. J_M is a matrix used to impose additional constraints on the optimum $\underline{\Delta u}(k)$. J_M may be an identity matrix $(P=M)$, used in blocking $(P\neq M)$, or in implementing base functions $(P\neq M)$; the decision lies with the designer. When J_M is an identity matrix there are no additional constraints on the optimum $\underline{\Delta u}(k)$ and the control is allowed to vary freely throughout the control horizon. The following is a description of how J_M is used for blocking. Consider again the example with P = 9 and M = [2 4 3]. With the first two control steps constant, the next four steps constant, and the final three steps constant $(u(k)=u(k+1),u(k+2)=u(k+3)=u(k+4)=u(k+5),u(k+6)=u(k+7)=u(k+8))$, J_M would then take on the form:

$$J_M = \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & I \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{9}$$

The third form J_M may take is when base functions are used. Base functions are specific control profiles throughout the prediction horizon. They may be ramp, sinusoidal, exponential functions, or other types of functions. The function type is a design choice. The $\underline{\Delta u}(k)$ applied to the plant is a linear combination of the base functions. $\underline{\Delta u}(k)$ is expressed mathematically as follows:

$$\underline{\Delta u}(k) = z_1 f_1(k) + z_2 f_2(k) + \ldots z_P f_P(k) \tag{10}$$

where the $f(k)$ are the base functions. Consider the following example. Let P = 9 and let there be five ramp base functions as shown in Fig. 6 for a SISO system.

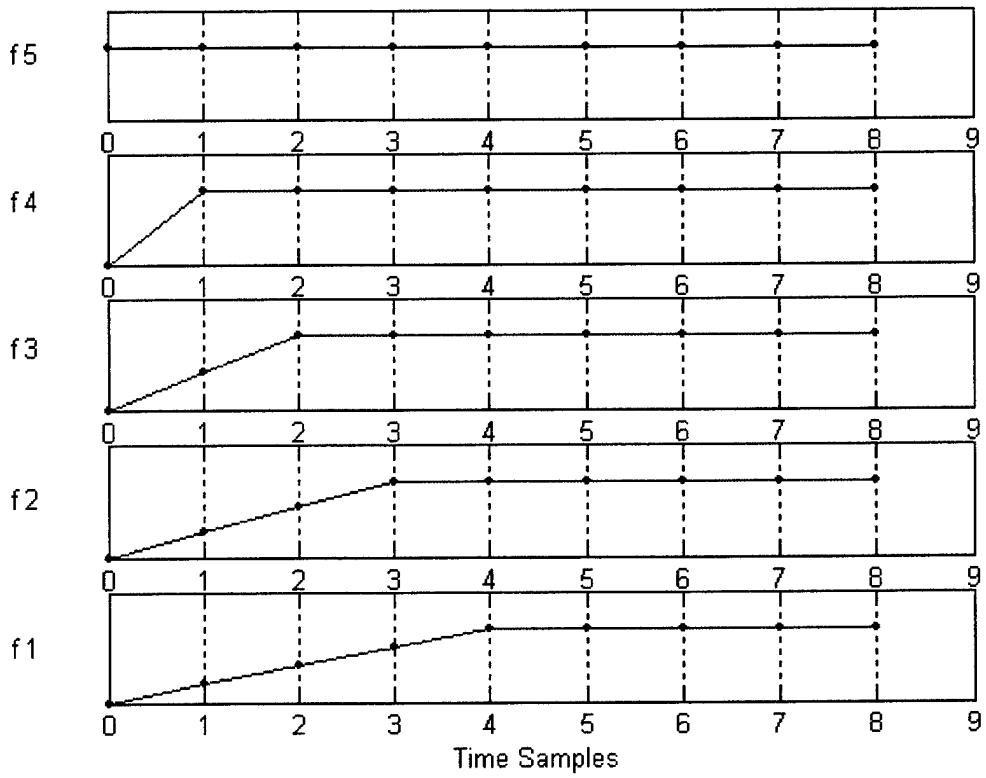

Fig. 6 Example Set of 5 Ramp Base Functions

The control vector $\underline{\Delta u}(k)$ results from the matrix multiplication $\underline{\Delta u}(k) = J_M \underline{z}(k)$ with the newly defined J_M.

$$J_M = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0.5 & 0.33 & 0.25 \\ 1 & 1 & 1 & 0.66 & 0.5 \\ 1 & 1 & 1 & 1 & 0.75 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \quad (11)$$

Base functions are used to reduce the number of optimizing variables. Fewer optimizing variables allows for quicker computation as an optimizer only needs to find values for a

few variables. The use of base functions should not significantly change the solution. It should only reduce computation time. In the next two sections, it is shown that the unconstrained solution has a closed form solution, but the constrained solution must use an optimizer to calculate the optimal values. Since the unconstrained case has a closed form, the benefit of base functions is smaller matrices making fewer calculations. Base functions and blocking may not be used at the same time because their influence is based on how J_M is constructed.

In a matrix form the cost function can now be rewritten as:

$$\begin{aligned} J = & \min_{\underline{z}(k)\in R^{(NP^*nu)\times 1}} \left[\underline{u}(k)-\underline{u}_T(k)\right]^T W_u \left[\underline{u}(k)-\underline{u}_T(k)\right] \\ & +\underline{z}^T(k) J_M^T W_\Delta J_M \underline{z}(k)+\left[\underline{y}(k)-\underline{r}(k)\right]^T W_y \left[\underline{y}(k)-\underline{r}(k)\right] \end{aligned} \qquad (12)$$

where $\underline{u}(k)\in\Re^{(P^*nu)\times 1}$, $\underline{u}_T(k)\in\Re^{(P^*nu)\times 1}$, $\underline{y}(k)\in\Re^{(P^*ny)\times 1}$, $\underline{r}(k)\in\Re^{(P^*ny)\times 1}$

$W_u\in\Re^{(P^*nu)\times(P^*nu)}$, $W_\Delta\in\Re^{(P^*nu)\times(P^*nu)}$, $W_y\in\Re^{(P^*ny)\times(P^*ny)}$

The first term of equation (12) penalizes deviations of the control input variable from the reference control. The second term penalizes the changes in the control input. The final term penalizes the tracking error (deviations of the state output from the reference trajectory). The cost function described in equation (12) is used throughout this thesis, however, MPC is not restricted to this cost function. For example, Berlin and Frank [Ref. 5] use a cost function penalizing the state tracking error and the absolute control effort. Furthermore, Heise and Maciejowski [Ref. 10] penalize the state tracking error and the changes in control input. The cost function in (12) is especially applicable to aerospace engineering as onboard trajectory generators are able to issue reference signals not only for the output of the states, but also for the control inputs. The selected cost function takes advantage of both reference signals.

Next, the predicted control $\underline{u}(k)$ may be expressed in the following manner:

$$\underline{u}(k) = I_p u(k-1) + K_1 \underline{\Delta u}(k) \qquad (13)$$

$$I_1 = Identity(nu \times nu) \qquad I_P = \begin{bmatrix} I_1 \\ I_1 \\ \vdots \\ I_1 \end{bmatrix} \in \Re^{(P*nu)\times nu} \qquad (14)$$

$$K_1 = \begin{bmatrix} I_1 & 0 & \cdots & 0 \\ I_1 & I_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ I_1 & I_1 & \cdots & I_1 \end{bmatrix} \in \Re^{(P*nu)\times(P*nu)} \qquad (15)$$

Expressing $\underline{u}(k)$ in this manner lends itself to being expressed according to the equations in (8) to form $\underline{z}(k)$. These conversions are necessary so that the cost function may be solved for a single variable, $\underline{z}(k)$.

2.1.3.2 Unconstrained Closed Form Solution

The goal of this derivation is to find the $\underline{z}(k)$ that minimizes the cost function described in (12). To do this, all of the terms in the cost function must first be defined as a function of $\underline{z}(k)$. Then the derivative of the cost function is set to 0, and $\underline{z}(k)$ is found explicitly. Because the cost function is quadratic, a unique solution is guaranteed.

Let $\Delta u(k+i\,|\,k)$ and $y(k+i\,|\,k)$ be the change in control input and the output predictions obtained by iterating the model i times in the future from the current state k. Define $\underline{\Delta u}(k)$, $\underline{y}(k)$, $\underline{u}(k)$, and $\underline{u_T}(k)$

$$\underline{\Delta u}(k) = \begin{bmatrix} \Delta u(k\,|\,k) \\ \Delta u(k+1\,|\,k) \\ \vdots \\ \Delta u(k+P-1\,|\,k) \end{bmatrix} \in \Re^{(P*nu)\times 1} \qquad \underline{y}(k) = \begin{bmatrix} y(k+1\,|\,k) \\ y(k+2\,|\,k) \\ \vdots \\ y(k+P\,|\,k) \end{bmatrix} \in \Re^{(P*ny)\times 1} \qquad (16)$$

$$\underline{u}(k)=\begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+P-1) \end{bmatrix} \in \Re^{(P*nu)\times 1} \qquad \underline{u_T}(k)=\begin{bmatrix} u_T(k) \\ u_T(k+1) \\ \vdots \\ u_T(k+P-1) \end{bmatrix} \in \Re^{(P*nu)\times 1} \qquad (17)$$

Similarly define $\underline{v}(k)$ and $\underline{d}(k)$:

$$\underline{v}(k)=\begin{bmatrix} v(k) \\ v(k+1) \\ \vdots \\ v(k+P) \end{bmatrix} \in \Re^{(P*nu+nv)\times 1} \qquad \underline{d}(k)=\begin{bmatrix} d(k) \\ d(k+1) \\ \vdots \\ d(k+P) \end{bmatrix} \in \Re^{(P*nu+nd)\times 1} \qquad (18)$$

When equation (5) is substituted into equation (6) for $y(k+1|k)$ to $y(k+i|k)$, the prediction at time k can then be expressed in the following form:

$$y(k+i|k)=C\left[A^i x(k)+\sum_{h=0}^{i-1}A^{i-1-h}B_u\left[u(k-1)+\sum_{j=0}^{h}\Delta u(k+j)+B_v v(k+h)+B_d d(k+h)\right]\right] \qquad (19)$$
$$+D_v v(k+i)+D_d d(k+i)$$

In the simplified vector form equation (19) becomes:

$$\underline{y}(k)=S_x x(k)+S_{u1}u(k-1)+S_u\underline{\Delta u}(k)+H_v\underline{v}(k)+H_d\underline{d}(k) \qquad (20)$$

where the terms are defined as

$$\underline{y}(k)=\begin{bmatrix} y(k+1|k) \\ y(k+2|k) \\ \vdots \\ y(k+P|k) \end{bmatrix} \in \Re^{(P*ny)\times 1} \qquad S_x=\begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^P \end{bmatrix} \in \Re^{(P*ny)\times nx} \qquad (21)$$

$$S_{u1} = \begin{bmatrix} CB_u \\ CB_u + CAB_u \\ \vdots \\ \sum_{h=0}^{P-1} CA^h B_u \end{bmatrix} \in \Re^{(P^* ny) \times nu} \qquad (22)$$

$$S_u = \begin{bmatrix} CB_u & 0 & 0 & 0 \\ CB_u + CAB_u & CB_u & 0 & 0 \\ \vdots & \vdots & \ddots & 0 \\ \sum_{h=0}^{P-1} CA^h B_u & \sum_{h=0}^{P-2} CA^h B_u & \cdots & CB_u \end{bmatrix} \in \Re^{(P^* ny) \times (P^* nu)} \qquad (23)$$

$$H_v = \begin{bmatrix} CB_v & D_v & 0 & \cdots & 0 \\ CAB_v & CB_v & D_v & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ CA^{P-1}B_v & CA^{P-2}B_v & CA^{P-3}B_v & \cdots & D_v \end{bmatrix} \in \Re^{(P^* ny) \times (P^* nu + nv)} \qquad (24)$$

$$H_d = \begin{bmatrix} CB_d & D_d & 0 & \cdots & 0 \\ CAB_d & CB_d & D_d & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ CA^{P-1}B_d & CA^{P-2}B_d & CA^{P-3}B_d & \cdots & D_d \end{bmatrix} \in \Re^{(P^* ny) \times (P^* nu + nd)} \qquad (25)$$

Now that $\underline{y}(k)$ is explicitly defined it may be substituted into (12) and rewritten as a function of $\underline{z}(k)$. For simplicity the cost function is partitioned into three terms and evaluated one term at a time. The derivative of each term is taken with respect to $\underline{z}(k)$.

The first term is written as a function of $\underline{z}(k)$.

$$\begin{aligned} J_1 &= \left[\underline{u} - \underline{u}_T\right]^T W_u \left[\underline{u} - \underline{u}_T\right] = \\ &= \left[I_p u(k-1) + K_1 \underline{\Delta u} - \underline{u}_T\right]^T W_u \left[I_p u(k-1) + K_1 \underline{\Delta u} - \underline{u}_T\right] \\ &= \left[I_p u(k-1) + K_1 J_M \underline{z} - \underline{u}_T\right]^T W_u \left[I_p u(k-1) + K_1 J_M \underline{z} - \underline{u}_T\right] \end{aligned} \qquad (26)$$

Now the derivative of the first term is found to be

$$\frac{\partial J_1}{\partial \underline{z}} = 2\left[I_p u(k-1) - \underline{u}_T\right]^T W_u K_1 J_M + 2\underline{z}^T J_M^T K_1^T W_u K_1 J_M \qquad (27)$$

The second term of the cost function is already expressed as a function of $\underline{z}(k)$.

$$J_2 = \underline{z}^T J_M^T W_\Delta J_M \underline{z} \qquad (28)$$

The derivative of the second term is

$$\frac{\partial J_2}{\partial \underline{z}} = 2\underline{z}^T J_M^T W_\Delta J_M \qquad (29)$$

Substitute the vector form of $\underline{y}(k)$ into the third term.

$$\begin{aligned} J_3 &= \left[\underline{y}-\underline{r}\right]^T W_y \left[\underline{y}-\underline{r}\right] \\ &= \left[S_x x + S_{u1} u(k-1) + S_u \underline{\Delta u} + H_v \underline{v} + H_d \underline{d} - \underline{r}\right]^T W_y \left[S_x x + S_{u1} u(k-1) + S_u \underline{\Delta u} + H_v \underline{v} + H_d \underline{d} - \underline{r}\right] \end{aligned} \qquad (30)$$

For terms without $\underline{\Delta u}$, define the sum as F since they can be treated as constants when taking the derivative.

$$F = \left[S_x x + S_{u1} u(k-1) + H_v \underline{v} + H_d \underline{d} - \underline{r}\right] \qquad (31)$$

The substitution of F simplifies J_3.

$$J_3 = \left[F + S_u \underline{\Delta u}\right]^T W_y \left[F + S_u \underline{\Delta u}\right] \qquad (32)$$

Next, taking the derivative gives

$$\frac{\partial J_3}{\partial z} = 2F^T W_y S_u J_M + 2\underline{z}^T J_M^T S_u^T W_y S_u J_M \qquad (33)$$

Sum the terms and set equal to zero to get the minimum cost.

$$\frac{\partial J}{\partial \underline{z}} = \frac{\partial J_1}{\partial \underline{z}} + \frac{\partial J_2}{\partial \underline{z}} + \frac{\partial J_3}{\partial \underline{z}} = 0 \qquad (34)$$

$\frac{\partial J}{\partial \underline{z}}$ written out expands to

$$\begin{aligned}\left[I_p u(k-1) - \underline{u}_T\right]^T W_u K_1 J_M + \underline{z}^{*T} J_M^T K_1^T W_u K_1 J_M + \underline{z}^{*T} J_M^T W_\Delta J_M \\ + F^T W_y S_u J_M + \underline{z}^{*T} J_M^T S_u^T W_y S_u J_M = 0\end{aligned} \qquad (35)$$

$\underline{z}^*$ represents the optimal solution to the cost function.

$$\begin{aligned}\underline{z}^{*T}\left(J_M^T K_1^T W_u K_1 J_M + J_M^T W_\Delta J_M + J_M^T S_u^T W_y S_u J_M\right) = \\ - F^T W_y S_u J_M - \left[I_p u(k-1) - \underline{u}_T\right]^T W_u K_1 J_M\end{aligned} \qquad (36)$$

Solve for $\underline{z}^*$

$$\underline{z}^* = -K_{du}^{-1}\left[F^T W_y S_u J_M + \left(I_p u(k-1) - \underline{u}_T\right)^T W_u K_1 J_M\right]^T \qquad (37)$$

With K_{du}^{-1} defined to be

$$K_{du} = \left(J_M^T K_1^T W_u K_1 J_M + J_M^T W_\Delta J_M + J_M^T S_u^T W_y S_u J_M\right) \qquad (38)$$

It is now convenient to replace the constants described by F

$$\underline{z}^* = -K_{du}^{-1}\left[\left(S_x x + S_{u1} u(k-1) + H_v \underline{v} + H_d \underline{d} - \underline{r}\right)^T W_y S_u J_M + \left(I_p u(k-1) - \underline{u}_T\right)^T W_u K_1 J_M\right]^T \qquad (39)$$

Group like terms in $\underline{z}^*$

$$\underline{z}^* = -K_{du}^{-1}\left[\begin{array}{l} -\underline{r}^T W_y S_u J_M + \left(H_v \underline{v} + H_d \underline{d}\right)^T W_y S_u J_M \\ \quad + u^T(k-1)\left(I_p^T W_u K_1 J_M + S_{u1}^T W_y S_u J_M\right) - \underline{u}_T^T W_u K_1 J_M + x^T S_x^T W_y S_u J_M \end{array}\right]^T \quad (40)$$

$\underline{z}^*$ can be written more concisely by defining the following matrices K_r, K_v, K_d, K_u, K_T, and K_x

$$K_r = \left[-W_y S_u J_M\right] \in \Re^{(P*ny)\times(M*nu)} \quad (41)$$

$$K_v = \left[H_v^T W_y S_u J_M\right] \in \Re^{(P*nu+nv)\times(M*nu)} \quad (42)$$

$$K_d = \left[H_d^T W_y S_u J_M\right] \in \Re^{(P*nu+nd)\times(M*nu)} \quad (43)$$

$$K_u = \left[I_p^T W_u K_1 J_M + S_{u1}^T W_y S_u J_M\right] \in \Re^{nu\times(M*nu)} \quad (44)$$

$$K_T = \left[-W_u K_1 J_M\right] \in \Re^{(P*nu)\times(M*nu)} \quad (45)$$

$$K_x = \left[S_x^T W_y S_u J_M\right] \in \Re^{nx\times(M*nu)} \quad (46)$$

The final optimal control is found to be

$$\underline{z}^*(k) = -K_{du}^{-1}\left[\begin{array}{l} \underline{r}^T(k) K_r + \underline{v}^T(k) K_v + \underline{d}^T(k) K_d \\ \quad + u^T(k-1) K_u + \underline{u}_T^T(k) K_T + x^T(k) K_x \end{array}\right]^T \quad (47)$$

In this derivation the optimal control is found directly. However, the unconstrained system may be described as a gain matrix multiplied by a state error signal. The optimal control is a linear feedback law with a feed forward signal of reference information.

2.1.3.3 Constrained Solution

The constrained solution solves the same problem as the unconstrained problem subject to the following constraints:

$$
\begin{aligned}
& u_i^{\min} \le u(k+i|k) \le u_i^{\max} \\
& \Delta u_i^{\min} \le \Delta u(k+i|k) \le \Delta u_i^{\max} \qquad i = 0,\ldots,P\text{-}1 \\
& -\epsilon + y_i^{\min} \le y(k+i+1|k) \le y_i^{\max} + \epsilon \qquad (48) \\
& \Delta u(k+j|k) = 0, \qquad j = M,\ldots,P \\
& \epsilon \ge 0
\end{aligned}
$$

where $u_i^{\min}$, $u_i^{\max}$, $\Delta u_i^{\min}$, $\Delta u_i^{\max}$, $y_i^{\min}$, and $y_i^{\max}$ are the design constraints selected by the designer. ϵ is a slack variable, also a design choice. Hard constraints are imposed on the control input and on the change in control input. Hard constraints mean that the MPC will not violate the constraints under any circumstance. Soft constraints are placed on the state output and may be violated, but at a high penalty cost. The penalty is the product of the amount the constraint was violated and the slack variable ϵ. ϵ is typically a large value so that MPC would rather sacrifice error in another state before violating soft constraints. When constraints are imposed on the system, no closed form solution exists so an optimizer must be used. The slack variable is required to insure the optimizer converges. Without the slack variable, the optimizer may not converge when the problem is overly constrained, the plant model doesn't match the actual model, or from round off error. Failure to converge in flight would likely result in loss of the vehicle. As a precaution for isolated points, if convergence is impossible, the previous applied control value is used. The research for this thesis used the Matlab optimizing routine DANTZGMP.m, however, any reliable optimizer may be used in a constrained MPC problem [Ref. 9].

2.2 Potential Benefits and Problems of MPC

2.2.1 Potential Benefits

Model Predictive Control like any other controller has potential benefits and problems associated with it. The benefits of using an MPC controller are discussed first. A useful

advantage of MPC is its prediction feature. MPC optimizes over the prediction horizon instead of an optimization of a current time. For this reason the MPC allows errors at the current time if allowing the errors enables the MPC to reduce future errors significantly. The MPC then uses anticipative action to its advantage. For example, if an MPC controller predicts a step change in the reference signal, it will begin changing the control input to the plant to accommodate for that step before it actually receives the command. This anticipation is useful as it can avoid large errors in overshoot by starting early and moving less aggressively. In addition, the anticipative behavior of MPC allows it to effectively meet constraints. If a constraint is imposed on the control input, the MPC will see the constraint in the prediction horizon and move preemptively to meet that constraint. Because MPC allows constraints, the designer is free to use the soft constraints on the output to tailor the output to specific values. Hard constraints may also be used to avoid saturating the control system.

Often a constraint on the input or output of a system will make it impossible to achieve the absolute minimum of the cost function. The controller must then find the minimum of the cost subject to the constraint. In such a situation it is advantageous to operate as close to the constraint as possible. Using anticipation, the MPC is capable of operating nominally very close to constraints because it can predict if it is going to violate a constraint. If it predicts it will violate a constraint, it takes corrective action to insure the constraint is met.

Another advantage to MPC is that in a closed loop system, stability is guaranteed even if the open loop plant is unstable and constraints are imposed. System stability is further addressed in Chapter 4. Additional information on stability can also be found in (Heise and Maciejowski) [Ref. 10] and (Bemporad and Morari) [Ref. 11]. Heise and Maciejowski use a state space formulation and a piecewise linear time-invariant control law with state and input constraints and a perfect model. The MPC uses an internal model control framework for defining sufficient conditions guaranteeing stability for stable and unstable plants under all constraints considered. Bemporad and Morari conduct a survey of a variety of stability formulations proposed in recent MPC literature. According to the survey, the specific method used to show stability must be chosen carefully to insure that the control system being analyzed meets all of the method's assumptions. As a consequence, ad hoc tuning of the MPC from a comprehensive set

of simulations over a variety of operating conditions is found to be a technique that may be applied to all MPC control laws.

MPC can control a variety of systems ranging from ones with simple dynamics to others with complex nonlinear dynamics, including unstable plants and systems with long delay times or of non-minimum phase. The SISO case may be easily extended to MIMO [Ref. 4]. It is an established and proven control algorithm in industry. Its concepts are well known and have been used by process plant engineers and by chemical engineers for years. However, MPC is relatively new to the aerospace industry.

MPC may compensate for measurable disturbances through feed forward control [Ref. 4]. Unmeasured disturbances may also be applied to the MPC model.

MPC controllers have multiple uses for a given problem. It can be used to augment an existing inner loop to create an MPC outer loop for the states with slow dynamics, or it may be used as the sole controller without an inner loop. These two architectures are demonstrated in the following chapter. MPC's flexibility expands its utility.

2.2.2 Potential Problems

With any controller there exists some disadvantages. The look ahead feature of MPC comes at the cost of greater complexity when compared to other modern controllers such as LQR. MPC requires the addition of the prediction routine to the control law. Organizing the large amount of data MPC requires for predictions is often a challenge. The predictions for the unconstrained case are organized using matrices, but each prediction is created using matrices, so the notation describing matrices within matrices may be confusing at times. The matrices easily become very large and the associated computation time increases significantly.

Another disadvantage is that the reference output in the future must be known. This information is not always readily available, so MPC may only be applied to certain problems where that information is known.

The MPC uses the internal plant model to make predictions about the output of the real plant. With that information it selects the control inputs that minimize the deviation from reference according to the cost function. This dependence on the internal plant model requires the model to closely match the actual plant. Robustness to model uncertainty is an issue.

Finally, one of the problems with MPC lies in the selection of the various parameters. This is where MPC application can become more of an art form than a science. Selecting the prediction horizon, control horizon, simulation rates, applying the proper constraints, and populating the weighting matrices are difficult tasks. As the number of states and control inputs increase, populating the weighing matrices becomes increasingly more difficult. The parameters cannot be calculated directly in a closed form solution, however, there do exist procedures a designer may perform to select adequate horizon lengths, weighting values, and other selection parameters. Chapter 4 provides a simplified MPC example to illustrate these procedures to help future designers choose the appropriate parameters. These procedures are applied in Chapter 5 to select the parameters for the full MPC simulation controlling the X-34.

[This page intentionally left blank]

Chapter 3

Architecture Description

The purpose of this chapter is to describe how MPC is applied to control the X-34 to fly a commanded trajectory from the guidance subsystem. The chapter begins with a vehicle description followed by a declaration of the selected states. Sample trajectories are introduced next. The chapter concludes with a discussion of the architectures used and an explanation of the linearization process.

3.1 X-34 Description

The X-34 was chosen as the testbed for research for four main reasons. First, the X-34 is representative of typical low lift to drag ratio (L/D) reusable launch vehicles. In addition, it has similar characteristics to the Space Shuttle. The Space Shuttle is also a low L/D vehicle and has the same control surfaces as the X-34. Third, Draper has closely worked with the Orbital Sciences Corporation (OSC), the developers of the X-34. From this close interaction much information about the X-34 has been shared, making the X-34 a logical choice for Draper's research. Finally, the X-34 was selected in order to take advantage of prior IG&C research [Ref. 1] [Ref. 3] [Ref. 12].

The X-34 is a technology demonstrator that is launched from the belly of an L-1011 aircraft, much like the Pegasus launch vehicle. The X-34 is secured underneath an L-1011 aircraft initially. The X-34 is released from the L-1011 at an altitude of about 30,000 feet. Following release, the X-34 ignites a kerosene and liquid oxygen engine with 60,000 pounds of thrust sending the vehicle to an altitude of 250,000 ft and to speeds approaching mach 8. The vehicle then reenters the atmosphere and lands on a runway [Ref. 1] [Ref. 3] [Ref. 12]. The vehicle carries no crewmembers and is, therefore, totally autonomous with respect to its guidance, navigation, and control (GN&C) operation.

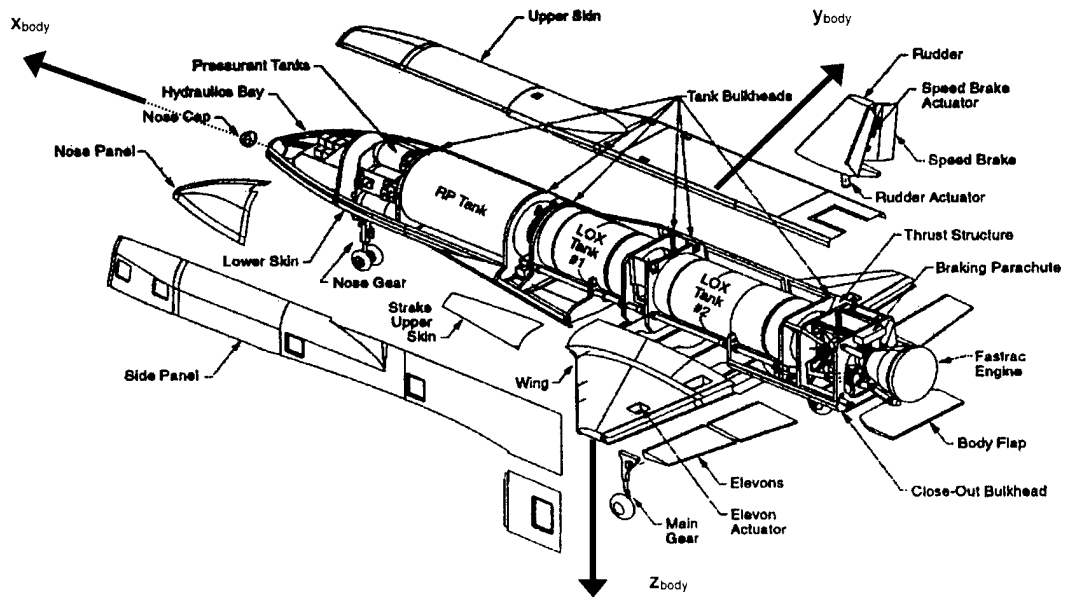

Fig. 7 Expanded View of the X-34

Fig. 7 and Tab. 2 show an expanded view of the X-34 and a summary of its physical characteristics. The landing weight is just greater than one-third the launch weight. The vehicle burns nearly 30,000 lbm of fuel causing this change in weight and causing the mass properties of the vehicle to change significantly. Upon entry, the center of gravity in the X-34 has moved well aft of the center of pressure. Normally such a configuration is undesirable as it causes the vehicle to be statically unstable. High performance fighter aircraft are designed in this way to increase maneuverability. In the case of the X-34, this instability results from a trade-off between launch and landing weight and stability requirements.

Length	58.3 ft
Wing Span, b	27.67 ft
Mean Aerodynamic Chord, c	14.54 ft
Planform Area, S	357.5 ft^2
Approximate Launch Weight	46,500 lbm
Approximate Landing Weight	18,000 lbm

Tab. 2 Physical Characteristics of the X-34

While the X-34 is unstable on entry, it can be stabilized in flight using its actuators in a closed loop control law. The X-34 has four types of aerosurfaces: an elevon, rudder, speed brake, and a body flap. The body flap is used to trim the vehicle and is omitted from the flight control in this research. The elevons function as both elevators and ailerons. When the elevons are deflected in unison, they act as elevators, but a differential deflection achieves the aileron control.

Tab. 3 shows the four aero controls available, the range of deflection, and approximate bandwidths for the actuators [Ref. 12].

Control	Symbol	Range of Motion (deg)	Actuator Bandwidth (Hz)
Elevon	δ_e	-34.2 to +15.8	8
Aileron	δ_a	-20 to +20	8
Rudder	δ_r	-25 to +25	6
Speed brake	δ_{sb}	0 to +103	0.5

Tab. 3 Control Variables and Actuator Characteristics

The sign convention for the actuators is defined in a right-handed body frame. The x direction is out of the nose of the vehicle. The y direction points off the right wing and

the z direction points down to complete the coordinate system. These directions are shown in Fig. 7. For the directional description, assume the observer is positioned as the pilot in the nose of the vehicle. A positive δ_e signifies downward motion of both elevons causing the vehicle's nose to pitch down. A positive δ_a corresponds to downward motion of the right elevon and upward motion of the left elevon generating a negative rolling rotation to the left. A positive δ_r means the trailing edge of the rudder moves left towards the left wing forcing the vehicle nose left and a negative yaw motion. Strong coupling between the aileron and rudder exist [Ref. 12]. The X-34 is a bank to turn vehicle. Aileron motion causes both a rolling and yawing moment. Similarly, rudder action generates rolling and yawing moments. The rudder alone has little capability over heading changes without incurring significant sideslip angles. As a result, the vehicle must bank and use some rudder input to complete a coordinated turn. The yaw rate is slow because the vehicle must first perform a roll maneuver [Ref. 1]. The speed brake has minimal drag when completely closed at 0° and maximum drag when δ_{sb} is 103°. The brake is used for controlling velocity only.

3.2 State Selection

The vehicle equations of motion (EOM) for a rigid body employ both longitudinal and lateral dynamics. The EOM for a vehicle with six degrees of freedom have been derived in previous research and may be found in [Ref. 1]. The following are a few assumptions made when deriving the EOMs:

- The vehicle may be treated as a rigid body
- The vehicle has a plane of symmetry
- Earth is an inertial reference frame
- The vehicle's mass properties are constant

The first two assumptions are valid as the X-34 is a mostly rigid vehicle, and it has symmetry about the x_{body}, z_{body} plane. This research is limited to the low altitude subsonic portion of TAEM and A/L, so assuming the Earth is an inertial reference frame is a valid approximation. Finally, for the entry portion of flight in the subsonic range, it is valid to assume the X-34's mass properties are constant. For the boost phase this would not be a valid assumption as the propellant mass is significant and reduces as it

burns. For subsonic entry, the vehicle does not use any propellant and relies solely on its aerosurfaces for control [Ref. 13].

Twelve quantities are required to accurately and completely represent the vehicle at any point in space. The twelve states are composed of three position states, three velocity states, three attitude states, and three angular rotation states. Flight path states are used because they allow for simplified integration for the guidance and control functions [Ref. 12].

State Description	Symbol	Units	Type
Downrange Position	x	Ft	Position
Crossrange Position	y	Ft	Position
Altitude	h	Ft	Position
Inertial (Ground-Relative) Speed	V	Ft/s	Velocity
Flight Path Angle	γ	Deg	Velocity
Heading Angle	χ	Deg	Velocity
Bank Angle about Velocity Vector	μ	Deg	Attitude
Angle of Attack	α	Deg	Attitude
Sideslip Angle	β	Deg	Attitude
Body Roll Rate	P	Rad/s	Rotational Rate
Body Pitch Rate	Q	Rad/s	Rotational Rate
Body Yaw Rate	R	Rad/s	Rotational Rate

Tab. 4 Description of State Variables

Tab. 4 shows the twelve selected states, their represented symbol, the units, and type. Six right-handed reference frames are used to explain the twelve state variables.

Inertial reference frame (i): an inertial frame centered at the beginning of the runway with the x-axis along the runway's centerline. The y-axis is perpendicular to the runway's centerline. The z-axis points into the ground, completing the right hand system.

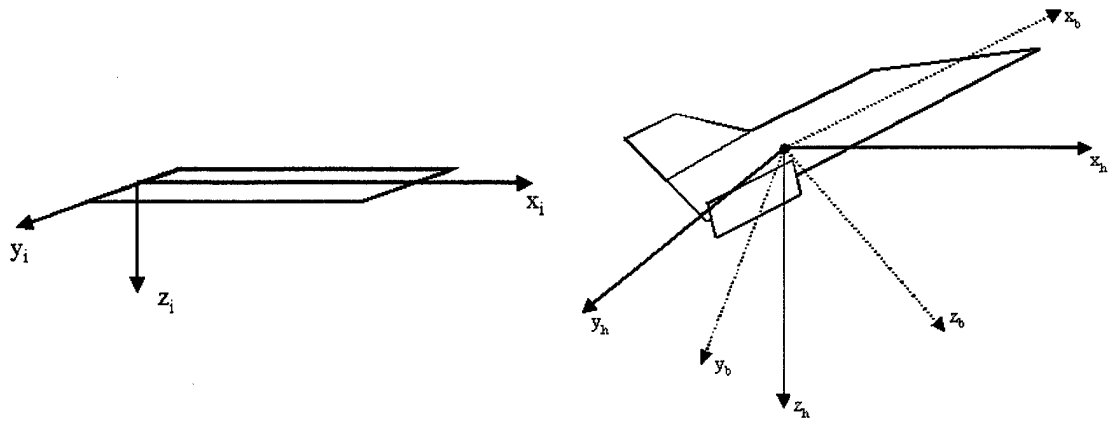

Fig. 8 Inertial, Local Horizon, and Body Frames

Local horizon frame (h): a frame identical to the inertial reference frame but centered at the vehicle's center of mass.

Body frame (b): centered at the vehicle's center of mass with the x-axis out of the vehicle's nose, the y-axis out of the right wing, and the z-axis out of the bottom of the vehicle. This frame is often referred to as the nose, right wing, down frame.

Fig. 8 illustrates the inertial, local horizon, and body frames.

Velocity frame (v): centered at the vehicle's center of mass with the x-axis along the inertial velocity vector. The y-axis is off to the right side in the local horizon xy plane and the z-axis completes the system.

Stability frame (s): This frame is centered at the vehicle's center of mass. The x-axis is along the projection of the inertial velocity vector on the xz plane of the body frame. The y-axis is in line with the body y-axis. Finally, the z-axis is in the same direction as the z-axis in the velocity frame.

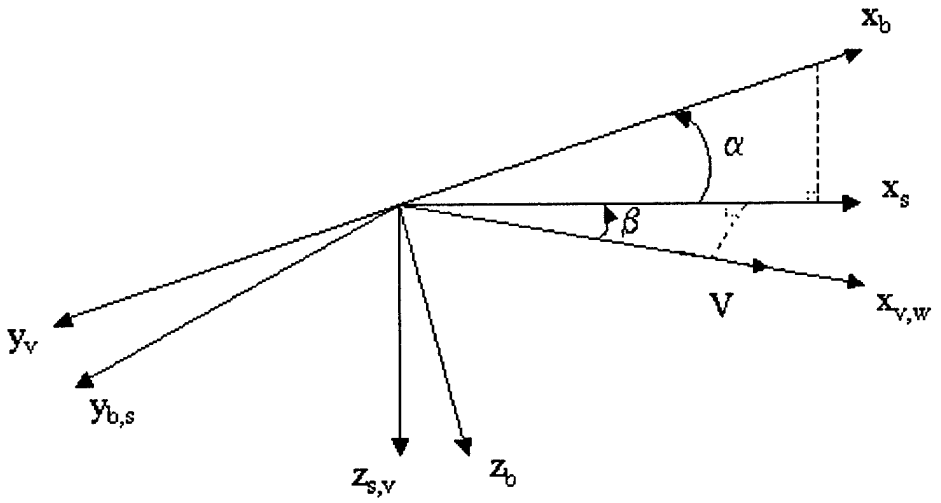

Fig. 9 Velocity, Stability, and Body Frames

Wind frame (w): The origin is at the vehicle's center of gravity. The x-axis coincides with the inertial velocity vector. The y and z-axes point in the same direction as the y and z-axes of the velocity frame except the axes are rotated through the vehicle's bank angle μ.

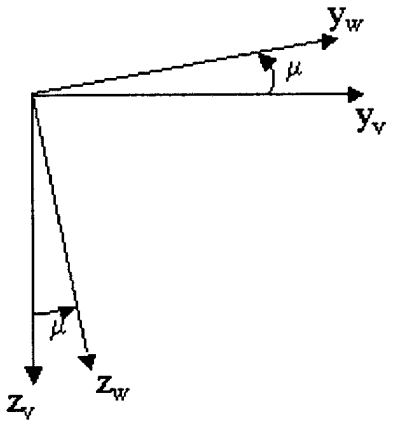

Fig. 10 Velocity and Wind Frames

Fig. 9 and Fig. 10 show the velocity, stability, and wind frames. In Fig. 10 the x-axis for both velocity and wind frames is positive into the page [Ref. 1].

Sign conventions are assigned to each of the 12 states. Downrange position is measured positive along the longitudinal axis of the runway starting at the beginning of the runway. Initially the vehicle starts with a negative downrange value. Crossrange

position is measured perpendicular to the runway's longitudinal axis with the left direction as the positive direction. Altitude is measured perpendicular from the plane of the runway with above ground measured as positive. Inertial speed is the magnitude of the total ground-relative velocity and thus always positive. The terms inertial speed and velocity are used interchangeably in this thesis when referring to the state. Flight path angle is the angle from the xy plane of the local horizon frame to the vehicle's velocity vector. The xy plane of the local horizon frame is 0. A negative flight path angle occurs when the vehicle's velocity is pitching downward towards the ground as show in Fig. 11.

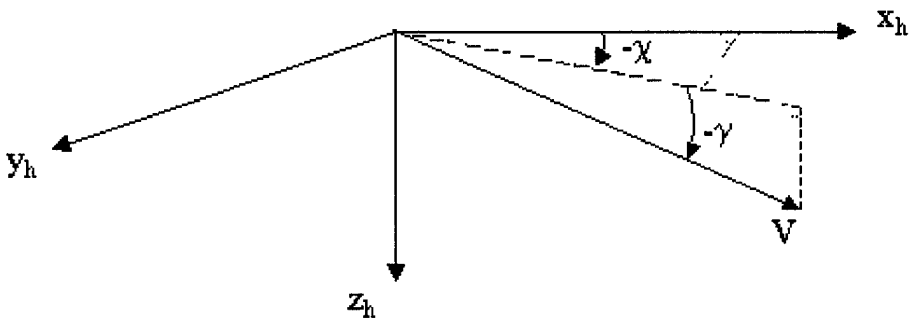

Fig. 11 Chi and Gamma Definition

The heading angle is the angle from the x-axis of the local horizon plane to the projection of the inertial velocity vector on the local horizon xy plane. A negative heading angle is shown in Fig. 11. Bank angle is the angle from the y-axis of the velocity frame to the y-axis of the wind frame. Bank angle is the roll angle of the vehicle with positive meaning a roll to the left as seen in Fig. 10. A positive angle of attack is the angle from the x-axis of the stability frame to the x-axis of the body frame as shown in Fig. 9. Sideslip angle is from the inertial velocity vector to the x-axis of the stability frame. A positive sideslip angle is presented in Fig. 9. Roll, pitch, and yaw rates are self-explanatory with positive defined as a left roll, a nose up pitch, and a left yaw motion respectively.

3.3 Flight Phases

The trajectories used for this research are all developed by code from the guidance division at Draper Laboratories. Each trajectory is restricted to the subsonic portion of

TAEM and the approach and landing phases of flight. Each trajectory is composed of as many as five of the following phases: acquisition, wings level flight (pre heading alignment), heading alignment, wings level flight (post heading alignment), and approach and landing. Fig. 12 shows the five flight phases in a typical trajectory [Ref. 14].

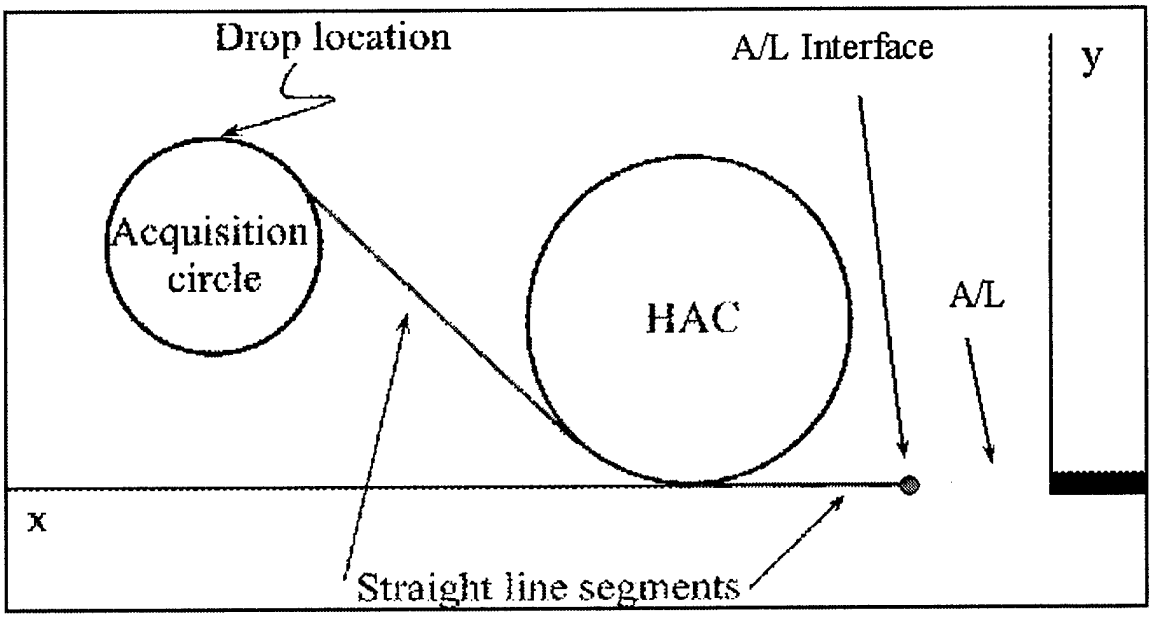

Fig. 12 Flight Phases

The A/L flight phase is exactly the same for each trajectory beginning at the A/L interface. The A/L interface is a point directly uprange of the runway, aligned with the centerline, and at 10,000 feet of altitude. The remaining four phases are then allowed to vary to give generality to the trajectories. The trajectory generation solves a two-point boundary value problem with the starting point as the vehicle drop location and the ending point as the approach and landing interface. A constraint on the boundary value problem requires the vehicle to have certain energy properties at A/L to insure it makes a safe landing on the runway. Each trajectory is geometrically based using straight lines, a cone, and a circle in its creation. The first phase is the acquisition phase starting from when the vehicle is released from the L-1011 aircraft. The vehicle is assumed to start with the wings level followed by an immediate bank. While in the bank, it flies along the circumference of the acquisition circle. A predetermined maximum allowable normal acceleration dictates the radius of curvature of the acquisition circle. The purpose of this phase is to change the vehicle's initial heading until it aligns with a

tangent point on the heading alignment cone (HAC). The HAC is an imaginary inverted cone placed tangent to the runway centerline and uprange of the runway. The first of two wings level flight phases is a straight line connecting the acquisition circle to the HAC at tangent points on each geometric shape. There is no banking occurring during this phase resulting in straight flight. The next phase is the heading alignment phase in which the vehicle performs its main banking maneuver. The X-34 follows the perimeter of the HAC in a near constant bank until the vehicle's heading is aligned with the runway's centerline. Following the heading alignment phase is the second wings level flight phase. It is a straight flight to the A/L intercept. Once at the A/L intercept, the approach and landing phase guides the vehicle to touch down on the runway.

Since the acquisition circle is only needed to change the vehicle's heading to point to the tangent point on the HAC, a subset of trajectories exists where the initial vehicle heading is in line with the HAC and no acquisition circle phase exists.

The two level flight phases and the A/L phase exercise longitudinal dynamics primarily, while the acquisition and heading alignment phases exercise both dynamics with an emphasis on the lateral dynamics. The vehicle has different flight characteristics resulting from varying dynamic pressures throughout each flight phase. These differences should be considered when selecting the weighting matrices, $W_u, W_\Delta, \text{and } W_y$. Many options may be considered when deriving the weighting matrices, but two methods stand out as likely choices. The first is to find weightings for each flight point, schedule the weighting matrices, and linearly interpolate between flight points. The second is to select separate weighting matrices for each flight phase that are held constant throughout the phase. Abrupt changes in the weighting matrices may create undesirable transient errors. When transients are observed, the weightings must be blended when applied. MPC allows for some natural blending, as P weighting matrices must be defined for the output prediction. For the weighting scheme based on flight phases, new weighting matrices should be introduced as follows assuming equal prediction and loop rates: Before transitioning from one flight phase to another, the MPC sees the same weightings throughout the entire prediction. When the transition point is within the prediction horizon of the MPC, the MPC sees the old weighting matrix for the entire horizon except for the last prediction sample. The last prediction sample is the new weighting matrix for the next phase. During the next iteration, the MPC sees one fewer old matrix weighting in the prediction and one more new weighting matrix.

The process continues until the MPC only sees the new weighting matrix. In this way the weighting matrices are blended naturally between flight phases.

3.4 Trajectories

The trajectories generated by the guidance system are composed of the flight phases described in 3.3. Each trajectory has approximately 180 flight points for about 3 minutes of flight. The points during A/L are given more frequently than in the subsonic portion of TAEM at altitudes higher than A/L. More frequent measurements are deemed desirable during the landing portion for more accurate results. During the trajectory generation process a vehicle energy level is fixed and the drop location is varied. The trajectories used range in their aggressiveness from fairly benign trajectories to very aggressive trajectories. A benign trajectory starts the vehicle in the middle of the flight envelope. An example of a benign trajectory is one that has little to no initial crossrange offset from the runway centerline and the initial heading value does not deviate from a heading in line with the HAC by more than a few degrees. Aggressive trajectories have starting points near the edges of the flight envelope. Specific definition of the flight envelope is beyond the scope of this thesis, but suffice it to say that an aggressive trajectory is one that meets one or more of the following conditions:

1) The initial vehicle position is far from the runway making the vehicle low on energy and forcing it to assume an extended max glide trajectory to make it to the A/L interface.

2) The initial position is very close to the runway and the vehicle is high on energy requiring an extended steep glide slope and a fully open speed brake.

3) The initial heading is more than 15 degrees from the tangent line to the HAC. In this case the vehicle has initial conditions in a hard bank.

4) The initial crossrange position is far from the runway centerline requiring an aggressive banking maneuver.

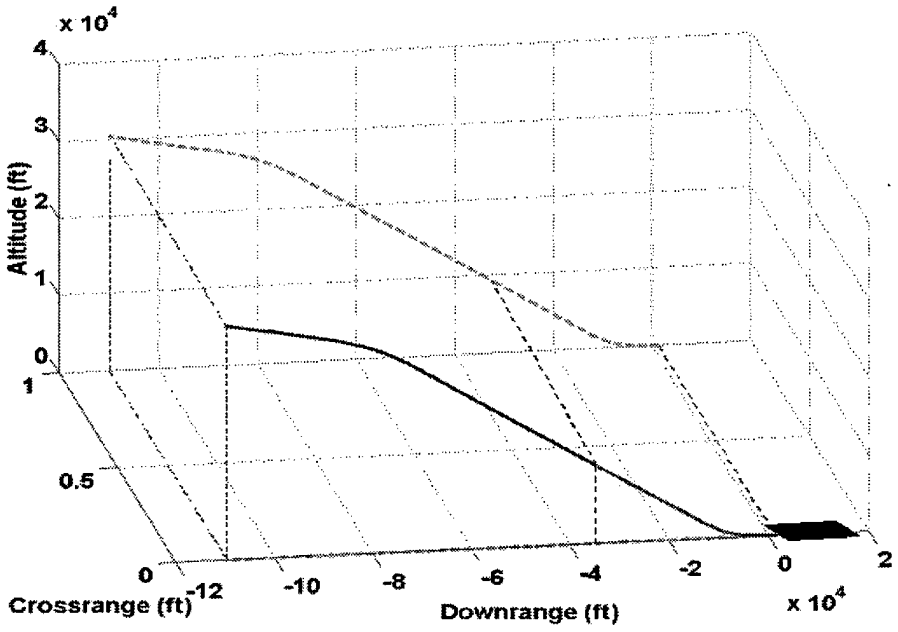

Fig. 13 Straight Trajectory

Fig. 13 is an example the most benign trajectory possible. The vehicle starts with a heading and position in line with the runway's centerline. Because the heading is aligned immediately, there is no acquisition turn phase and since the vehicle's position has 0 crossrange, there is no heading alignment phase. The result is just one straight, wings level flight section and the straight approach and landing phase. This trajectory is useful in isolating the longitudinal dynamics to help weight the longitudinal states in the weighting matrix.

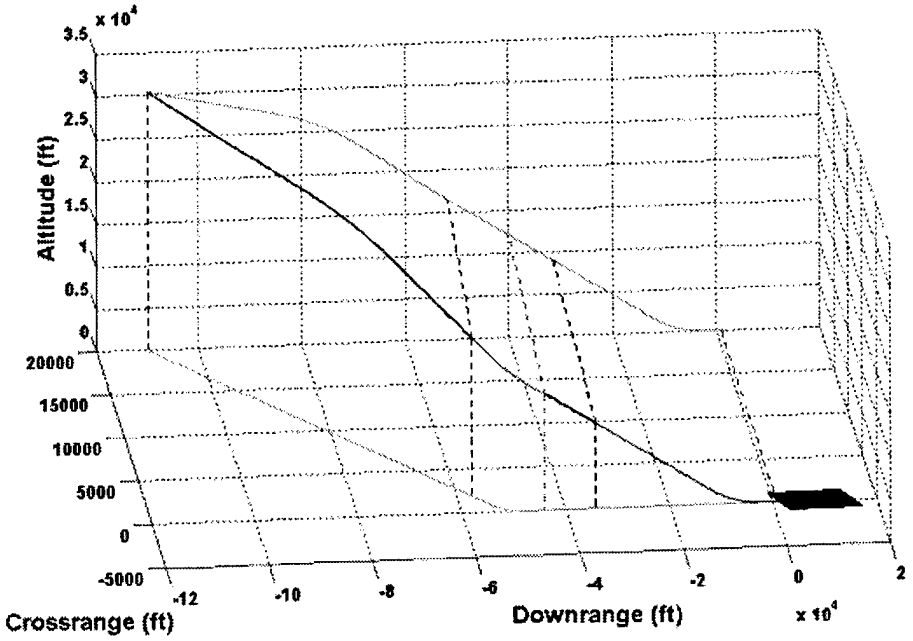

Fig. 14 Trajectory with a Single Banking Maneuver

A more complex, but still benign trajectory is the trajectory shown in Fig. 14. This trajectory starts with a heading towards the tangent point on the HAC, so there is no acquisition turn. It is however, offset from the runway's centerline by 20,000 feet. This trajectory emphasizes the longitudinal dynamics during the straight flight portions and introduces the lateral dynamics during the heading alignment phase. This trajectory is useful when weighting the lateral states. For a complete design many trajectories should be considered. Fig. 13 and Fig. 14 show two of those trajectories useful when isolating particular dynamics.

3.5 Architecture Selection

When developing a controller, the architecture used in implementing it is important to solving the given problem. For this reason, many architectures were evaluated for this research and narrowed to two specific architectures described below. Both architectures assume no disturbances exist. The purpose of this research is not necessarily to prove that one architecture is better than another, though some comparisons are made to show advantages and disadvantages a designer should be

aware of when designing an MPC controller. The two architectures are used to show that MPC can be applied in multiple ways to achieve favorable results.

3.5.1 MPC with Inner Loop SAS

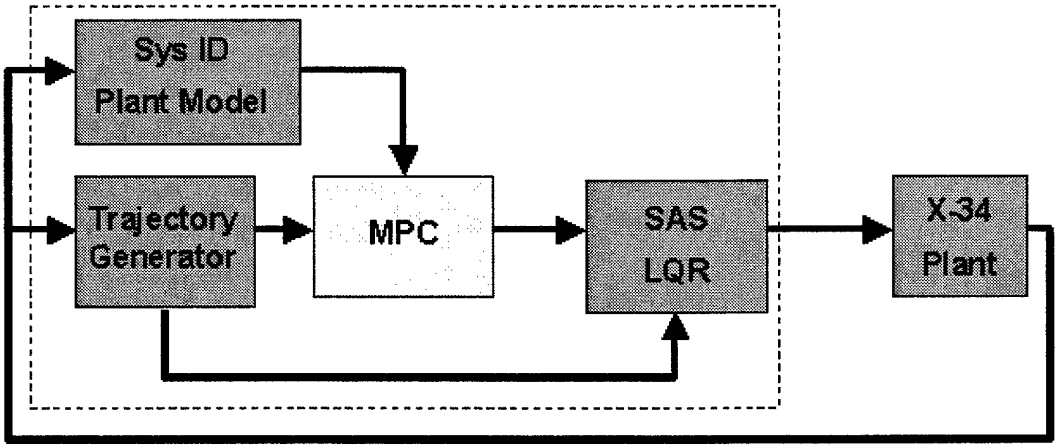

Fig. 15 MPC Architecture with Inner Loop SAS

Fig. 15 shows the first of two architectures. It is referred to as the MPC_SAS architecture throughout the remainder of the thesis. In this architecture the MPC controller is used as an outer loop to augment an existing inner loop stability augmentation system. The plant used is a full nonlinear plant describing the X-34's dynamics. The internal plant model is an LTI approximation of the actual nonlinear plant. While a nonlinear internal plant suggests a more accurate model, the LTI system allows for a simplified design and a state space representation. Furthermore, the increased accuracy of a nonlinear internal plant comes at the expense of increased computational time. The internal plant is approximated at each flight point throughout the trajectory and then linearly interpolated between flight points over a span of no more than 2.5 seconds of flight. The trajectory generator is given from the guidance subsystem. Combining the frequent LTI samples and linear interpolation between flight points, an accurate internal plant model is achieved. Since it is unlikely a perfect model could be constructed onboard the vehicle, some deviation between the internal plant and the actual plant in the research is advantageous because it shows favorable results may be obtained in light of a slight plant mismatch. The MPC block in this architecture controls the altitude, inertial speed, and crossrange trajectory states. These three states were selected because they are slow dynamic states allowing the MPC to run at a

slower rate than the SAS. Furthermore, these states give MPC partial information about both longitudinal and lateral dynamics to give a more coordinated response.

The MPC block outputs flight path, heading, and speed brake commands. The only aerosurface commanded by the MPC is the speed brake because the velocity is the only state influencing the speed brake position. The flight path and heading commands enter the LQR block where the nine remaining states are controlled. The inner loop SAS is an LQR servo, however, in general, many controllers could be used for inner loop stability. The LQR was selected to take advantage of previous research and because LQR is similar to MPC in that they are both MIMO optimal controllers that optimize similar cost functions. The LQR uses all twelve states to calculate gains, but applies gains only to the nine states the LQR is responsible for controlling. By incorporating all twelve states in the gain calculation, the LQR is able to take the dynamics of the three outer loop states into consideration when making the gains for the nine controlled states. The LQR multiplies the gains by the difference in the command and current value of the states to generate aerosurface commands for the elevon, aileron, and rudder.

Full state feedback is assumed throughout this architecture. If this assumption were not made, a Kalman estimator would be needed to accurately estimate the missing states.

This architecture is useful because often a vehicle already has an inner loop control system and only needs an outer loop for certain states. MPC is flexible enough to act as just an outer loop. Since MPC is computationally intensive, the MPC_SAS architecture allows the MPC to be applied to specific states that will benefit from MPC control and leave out states where MPC is not worth the computational effort. A disadvantage of the MPC_SAS structure is that the MPC only predicts the future outputs of the states it is given, so this structure forces the MPC to act without full information. However, by reserving the MPC for only states with slow dynamics, the lack of information rarely is a problem. Finally, anticipative action is only seen in the states controlled by the MPC.

3.5.2 MPC without Inner Loop SAS

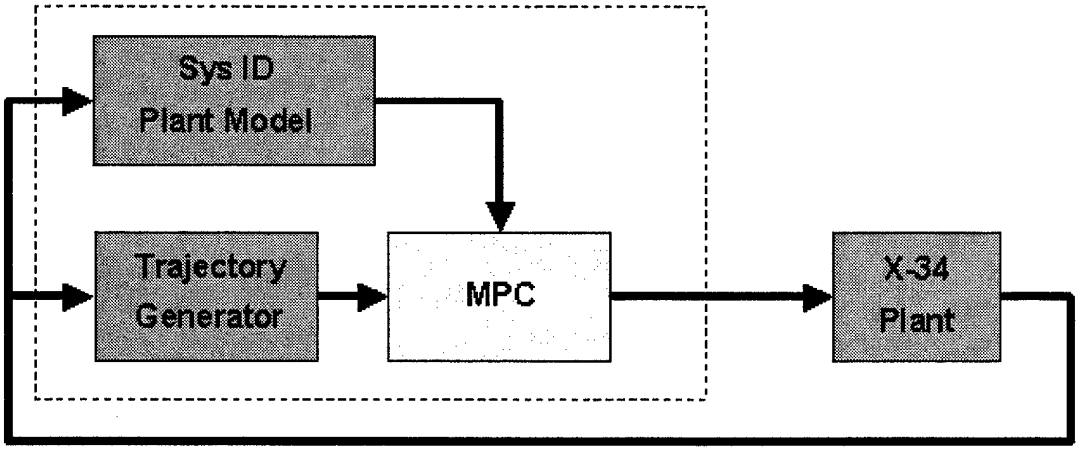

Fig. 16 MPC Architecture without Inner Loop SAS

Fig. 16 exhibits the second architecture evaluated. This architecture will be referred to as MPC_ALL in the future. The plant, internal plant, and trajectory generator are the same in both architectures. MPC_ALL is unique because it does not have any type of inner loop. Full state feedback is assumed with all twelve states inputted into the MPC controller. The MPC then controls all twelve states and generates all four aerosurface commands for the plant.

This architecture may be used when the vehicle does not have any portion of the control system developed. One advantage with the MPC_ALL architecture is that MPC has knowledge of all of the states, so it can develop a fully coordinated optimal solution. In addition, some anticipative action may be observed in all of the states. MPC_ALL comes with disadvantages as well as benefits. This architecture has fast and slow dynamics included, so the MPC must be run at a higher rate than seen with the outer loop of the MPC_SAS architecture. This increased rate combined with the added states makes this architecture very computationally intensive. A bittersweet feature of MPC_ALL is that the controller design is simplified in one sense and more complex in another. It is a simpler design because no inner loop needs to be designed and integration issues between inner and outer loops is nonexistent. It is more complex, however, because the designer must now develop a weighting matrix for all of the states. A careful weighting design is required because the entire controller relies on the single loop. With the MPC_SAS architecture, the inner loop has most of the states, making accurate weightings in the MPC less critical.

3.6 Trajectory Linearization

In section 3.5 it is stated that the plant used is a full nonlinear plant representing the X-34's dynamics, but the internal plant is an LTI state space model. It is necessary to explain the linearization process used to make the transformation. The derivation starts from the nonlinear plant. The vehicle dynamics are described by nonlinear equations of motion derived in [Ref. 1] and represented as a function of the states and control inputs below

$$\dot{x} = F(x,u) \tag{49}$$

The states and the control may be expressed as the sum of their nominal values and an incremental deviation from the nominal condition. The notation x_0 and u_0 is used to represent the nominal condition for the vehicle states and control.

$$\begin{aligned} x(t) &= x_0 + \delta x(t) \;\Rightarrow\; \delta x(t) = x(t) - x_0 \\ u(t) &= u_0 + \delta u(t) \;\Rightarrow\; \delta u(t) = u(t) - u_0 \end{aligned} \tag{50}$$

Combining the above equations gives the following result:

$$\dot{x} = F\left[\left(x_0 + \delta x(t)\right), \left(u_0 + \delta u(t)\right)\right] \tag{51}$$

$\dot{x}$ may now be described by an expanded Taylor series centered on the nominal values. A total of n functions with n states and w control inputs are needed in the expansion. For generality, the i^{th} function is shown below.

$$
\begin{aligned}
f_i[(x_0+\delta x(t)),(u_0+\delta u(t))] = f_i(x_0,u_0) &+ \left.\frac{\partial f_i}{\partial x_1}\right|_{x_0}\delta x_1 \\
&+ \left.\frac{\partial f_i}{\partial x_2}\right|_{x_0}\delta x_2 \cdots + \left.\frac{\partial f_i}{\partial x_n}\right|_{x_0}\delta x_n \\
&+ \left.\frac{\partial f_i}{\partial u_1}\right|_{u_0}\delta u_1 + \left.\frac{\partial f_i}{\partial u_2}\right|_{u_0}\delta u_2 \cdots + \left.\frac{\partial f_i}{\partial u_w}\right|_{u_0}\delta u_w \\
&+ \text{Higher Order Terms}
\end{aligned}
\tag{52}
$$

The notation $\left.\frac{\partial f_i}{\partial x_j}\right|_{x_0}$ represents the derivative of the i[th] function with respect to the j[th] state evaluated at the nominal conditions.

It is assumed that the deviations δx_i are small, making higher order terms very small and, therefore, negligible. The Taylor series expression may now be written in a more compact matrix form below to include all n functions:

$$
\dot{x} = f(x_0,u_0) + \left.\frac{\partial f}{\partial x}\right|_{x_0}\delta x + \left.\frac{\partial f}{\partial u}\right|_{u_0}\delta u \tag{53}
$$

where

$$
\frac{\partial f}{\partial x} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \qquad \frac{\partial f}{\partial u} = \begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \frac{\partial f_1}{\partial u_2} & \cdots & \frac{\partial f_1}{\partial u_w} \\ \frac{\partial f_2}{\partial u_1} & \frac{\partial f_2}{\partial u_2} & \cdots & \frac{\partial f_2}{\partial u_w} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial u_1} & \frac{\partial f_n}{\partial u_2} & \cdots & \frac{\partial f_n}{\partial u_w} \end{bmatrix} \tag{54}
$$

are constant matrices evaluated at specific points x_0 and u_0 and called the Jacobian matrices. $f(x_0,u_0)$ is a vector of the nonlinear equations evaluated at the nominal condition and may be written as $\dot{x}_0$ since it is just a specific evaluation of equation (49). In many derivations of a linearized model as in [Ref. 15], x_0 and u_0 are defined as

equilibrium points such that $\dot{x}_0 = 0$. This is valid for linearizing about a flight point in simpler models. MPC requires a model linearized about the trajectory over the prediction horizon. The nominal condition may not be defined as $\dot{x}_0 = 0$ because the states selected for this research include velocity and position states prohibiting such a simplification. $\dot{x}_0 = 0$ means that at the nominal conditions, the derivative of the states is constant. It is possible for a nonzero velocity to be constant allowing $\dot{V}_0 = 0$, but it is impossible to have a constant nonzero velocity and a constant position at the same time. An aerospace vehicle often has a high velocity causing a significant change in position in as little as a few seconds of time, so the $\dot{x}_0$ may not be neglected. For the research conducted, it is assumed that the vehicle's velocity terms are constant over the prediction horizon making the higher order derivatives 0. This is a valid assumption for flight segments without rapid, aggressive maneuvers.

To further simplify the notation, let A and B represent the Jacobians for the states and control inputs respectively giving:

$$\dot{x} = \dot{x}_0 + A\delta x + B\delta u \qquad (55)$$

δx and δu may be rewritten as a difference and the A and B matrices may be distributed

$$\dot{x} = \dot{x}_0 + A(x - x_0) + B(u - u_0) \;\Rightarrow\; \dot{x} = \dot{x}_0 + Ax - Ax_0 + Bu - Bu_0 \qquad (56)$$

Because the nominal state and control vectors are specified, it is useful to collect the constant terms in a single vector h_0.

$$h_0 = \dot{x}_0 - Ax_0 - Bu_0 \qquad (57)$$

Finally to include the constant terms and to maintain a state space formulation it is necessary to augment the plant matrices as shown in equation (58).

$$\underbrace{\begin{bmatrix} \dot{x} \\ \dot{h}_0 \end{bmatrix}}_{\dot{x}_{new}} = \underbrace{\begin{bmatrix} A & I \\ 0 & 0 \end{bmatrix}}_{A_{new}} \underbrace{\begin{bmatrix} x \\ h_0 \end{bmatrix}}_{x_{new}} + \underbrace{\begin{bmatrix} B \\ 0 \end{bmatrix}}_{B_{new}} u \qquad (58)$$

$\dot{x}_{new}$ and x_{new} are now of size 2n x 1. A_{new} is of size 2n x 2n and B_{new} is of size 2n x w. Only one augmentation is necessary because the higher order terms are assumed constant over the prediction horizon. The internal model without the augmentation is incorrect and results in a poor representation of the actual plant. It follows that the predictions of such a model are incorrect. To better illustrate this fact, Fig. 17 shows the propagated altitude reference and the predicted future outputs with and without the plant augmentation for the vehicle at flight point 140 of the single bank trajectory. Flight point 140 is selected as it is representative of a typical flight point in a wings level state. The flight point is before the banking maneuver at an altitude of 24,000 feet and about 67 seconds into the 3-minute flight. This flight point is used in examples throughout the remainder of the thesis starting with the next chapter. A 3-second prediction is shown. For longer horizons, the curve representing the plant without the augmentation diverges significantly from the actual altitude profile. The prediction with the augmentation is a very close approximation to the actual reference giving an accurate linearization about the trajectory.

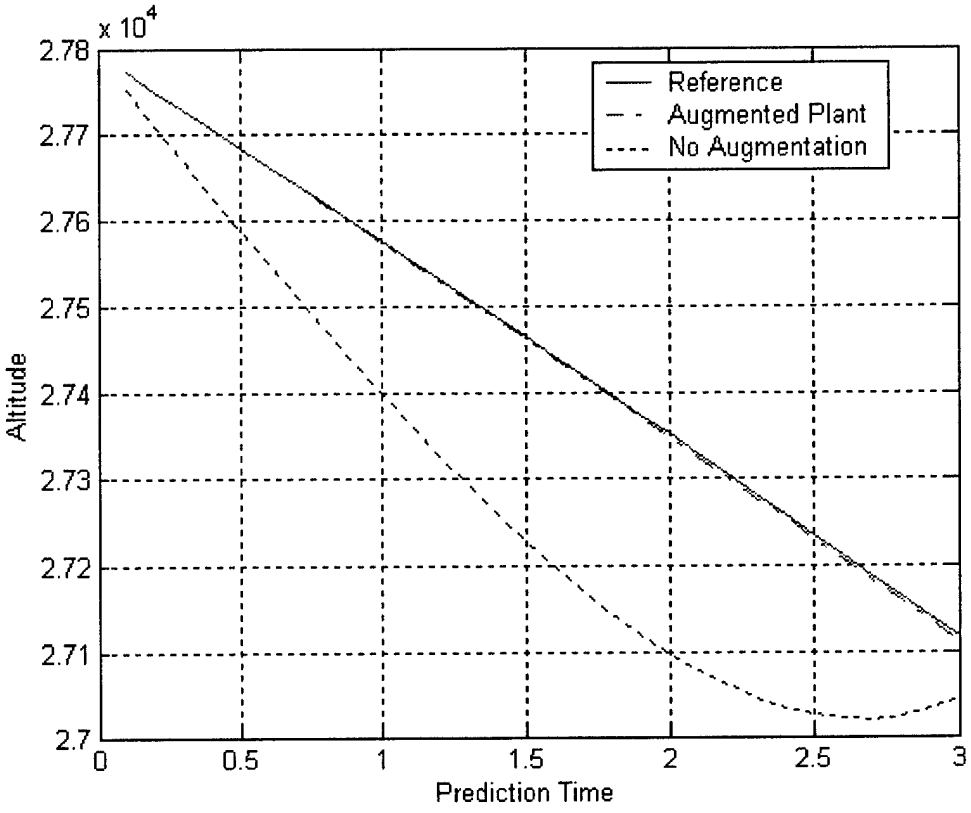

Fig. 17 Altitude Prediction With and Without Plant Augmentation

Chapter 4

MPC Design Guidelines

Model Predictive Control has many design variables that must be chosen. These parameters include: prediction and control horizons, appropriate inner and outer loop rates, and relative state and control weightings. Selecting these parameters is nontrivial. At this stage, the controller design becomes more of an art than a science. No one has yet defined equations to directly calculate these values, and little work has been done to develop rules for selecting them. The goal of the next two chapters is to provide some insight to properly selecting the above parameters. Guidelines, procedures, and suggestions are provided to aid MPC designers to select the parameters in a methodical manner. Many of the parameters are mapped to classical control properties.

It is now helpful to make a temporary digression from the full MPC design to a simplified example. This example is valuable for defining some design criteria and guidelines to help properly design an MPC controller. Only the longitudinal dynamics are considered for a simplified design. For this example, a benign trajectory is selected and a state space LTI model is calculated for all twelve states at each flight point. The state vector is reduced to only four longitudinal states and extracted from the newly created state space model. The reduced state vector is defined as

$$x = \begin{bmatrix} V \\ \alpha \\ Q \\ \theta \end{bmatrix} \tag{59}$$

V is velocity, α is the angle of attack, Q is the pitch rate, and θ is pitch angle. α and θ are used instead of α and γ.

During the linearization process, the $\dot{x}_0 = 0$ approximation is made causing the plant augmentation step discussed in section 3.6 to be unnecessary. It is reasonable to assume that none of these states change significantly over the prediction horizon

because the prediction horizon is typically on the order of a few seconds. The system being solved reduces to

$$\dot{x} = A\delta x + B\delta u \qquad (60)$$

The linearized four state model is used for the actual plant and for the internal plant during simulation to eliminate any error from a plant model mismatch. The system architecture is modeled after the MPC_ALL architecture shown in Fig. 16 except θ is the only state controlled by the MPC, leaving the remaining states uncontrolled. The elevon is subsequently the only control input as the remaining aerosurfaces do not significantly influence the θ profile. By having one state and one control input, the weighting matrices for the states, control, and change in control are reduced to scalar values.

For this example, the single bank trajectory shown in Fig. 14 has been selected because it is a typical, benign trajectory. The trajectory has lateral dynamics included in the banking portion of flight, however, throughout this example only the longitudinal dynamics are extracted. It is important to note, that there is some coupling between the longitudinal and lateral dynamics that will be seen in the following subsections. An example of the coupling is the evident correlated pitch command during a banking maneuver. It is well known that when an aerospace vehicle such as the X-34 banks, the nose of the vehicle drops naturally unless there is an added pitch command to keep the nose up.

4.1 Prediction Horizon Guidelines

The prediction horizon is an important parameter to select correctly because it represents the length of time the MPC will predict into the future. If the prediction horizon is too small, MPC will not have adequate knowledge of the plant and the response will not track the command well or may go unstable. If the prediction horizon is too long, the computational time becomes too great. For the full simulation where the internal linear plant approximates the actual nonlinear plant, it is found that as the prediction horizon gets longer, the linear model loses validity resulting in poor performance. For the longitudinal example, the internal and actual plants are identical, so the only penalty of a long prediction horizon is computational effort.

Selecting the prediction horizon is independent of the weighting matrices and not significantly influenced by the other parameters. The procedure developed to determine the appropriate prediction horizon uses this independence and makes the P horizon a parameter that should be found first. This procedure is not the only way to find the prediction horizon, but it is one that has been found to work effectively. It is applied to this specific example and then it is summarized at the end of this section.

From the previous chapter, flight point 140 was selected as a typical flight point in the single bank trajectory. This flight point is used for this example as well. The continuous LTI model is described by the state space model in (61).

$$A=\begin{bmatrix} -0.223 & -0.1596 & 0 & -24.0882 \\ -0.0079 & -0.5927 & 57.2958 & 0.4898 \\ -0.0001 & 0.0081 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad B=\begin{bmatrix} -0.1234 \\ -0.2106 \\ -0.0700 \\ 0 \end{bmatrix}$$

$$C=\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad D=\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad (61)$$

The matrices A, B, C, and D are then converted to the discrete form and used to formulate the prediction matrices S_x, S_u, S_{u1} etc. described in the unconstrained closed form solution section. Next, the weights on u and Δu are set to 0.5 and 1.0, respectively. Since the P horizon is independent of u and Δu, it is not important what they are fixed to, just that they are fixed values. A target bandwidth is selected next corresponding to the desired level of performance required of the closed loop system. For this example, a bandwidth of 1.0 rad/s is selected. This bandwidth is reasonable for such a vehicle in subsonic TAEM and may be increased for a faster response. A range of possible P horizons is selected for evaluation. The selected horizon ranges from 0.1 seconds to 7 seconds in 0.1-second increments. A 0.1-second horizon is very short for most systems. Likewise, 7 seconds is very long. The best horizon from this range is selected using an iterative process.

A single input single output (SISO) closed loop transfer function is created from the θ reference value to the final θ output. The closed loop system is shown in Fig. 18.

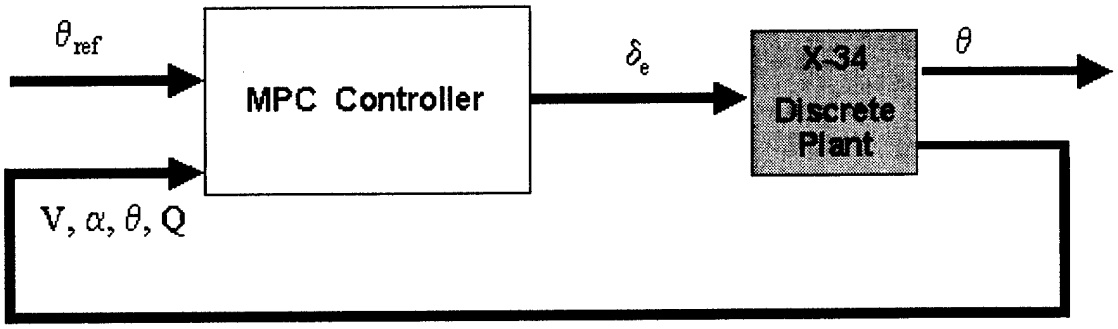

Fig. 18 Longitudinal Example Architecture

The MPC controller for the unconstrained system has a closed form solution. The mathematical background section calculates the control directly from the prediction matrices, reference signals, current state vector, and previous control vector. Alternatively, the new control value could be described as a single gain matrix multiplied by an error signal, making it a linear system. The same linearized state space plant used as the internal model is also used as the actual plant in the above figure. The plant is representative of the current flight point being evaluated. It is assumed that the plant and the θ input reference value are constant throughout the prediction horizon. Because the closed loop system shown in Fig. 18 only has linear components, the desired transfer function may be calculated analytically. The transfer function can also be found by assembling the diagram as a Simulink model and calling the Matlab function "dlinmod.m". A Bode plot of the newly derived transfer function is created. The system bandwidth is found as the frequency where the magnitude crosses the –3 db level. Next, the Matlab search function "FMINBD.m" varies the θ state weighting until the system achieves the desired bandwidth. The θ weighting is stored, and the procedure is repeated for each P horizon in the selected range.

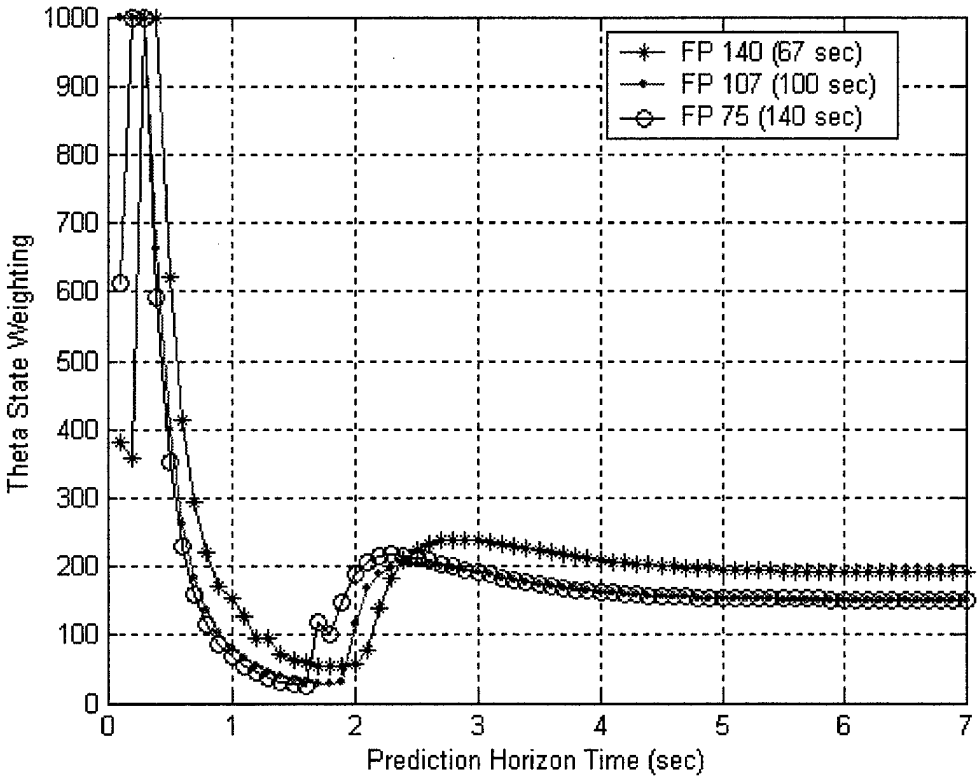

Fig. 19 Theta State Weighting for Various P Horizons and Flight Points

Fig. 19 shows a plot of the θ weights that gives the closed loop MPC system a bandwidth of 1.0 rad/s as a function of the prediction horizon. The optimizer converges to a θ weighting for every P horizon tested. In addition, when the weights change very little from one horizon to another, the set of weights as a whole is said to converge to a solution. Flight points 107 and 75 have been added and are shown in the figure. The flight points 107 and 75 were selected because they are typical flight points in the trajectory but have different dynamic pressures than flight point 140. Flight point 107 is 100 seconds into the flight at about 15,600 feet above the ground. It represents the vehicle in a banked state at about 29 degrees of bank in flight phase 3. Flight point 75 is farther down the trajectory at 140 seconds into the flight with an altitude of 6,300 feet. Flight point 75 is in the approach and landing flight phase. The three flight points are shown to demonstrate that not every flight point converges to the final weighting at exactly the same P horizon, so all flight points must be evaluated. The flight point requiring the longest P horizon is the flight point that dictates the appropriate horizon length. It is not important or expected that the state weights be the same value between

flight points because different flight points have different dynamics. It is, however, necessary and sufficient that the state weighting converges for a single flight point.

Obtaining the correct θ weighting is critical because there is a direct correlation between the state weighting and the control gains that are applied to generate the optimal control. Fig. 20 shows the gain contribution from the Kr and Ku gain matrices applied to obtain the optimal control as a function of horizon length. In the same way that the weightings converged to a final solution as the horizon is lengthened, the gains converge to their optimal values at the same horizon length. From Fig. 19 and Fig. 20 a prediction horizon of 4 seconds is selected.

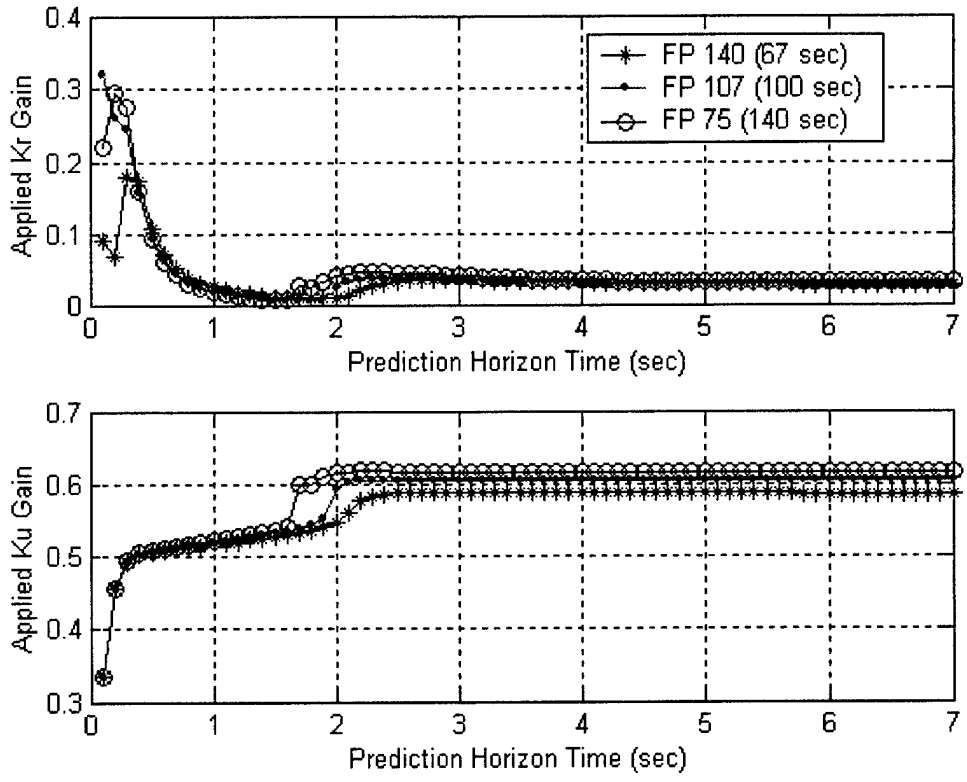

Fig. 20 Control Gains for Various P Horizons and Flight Points

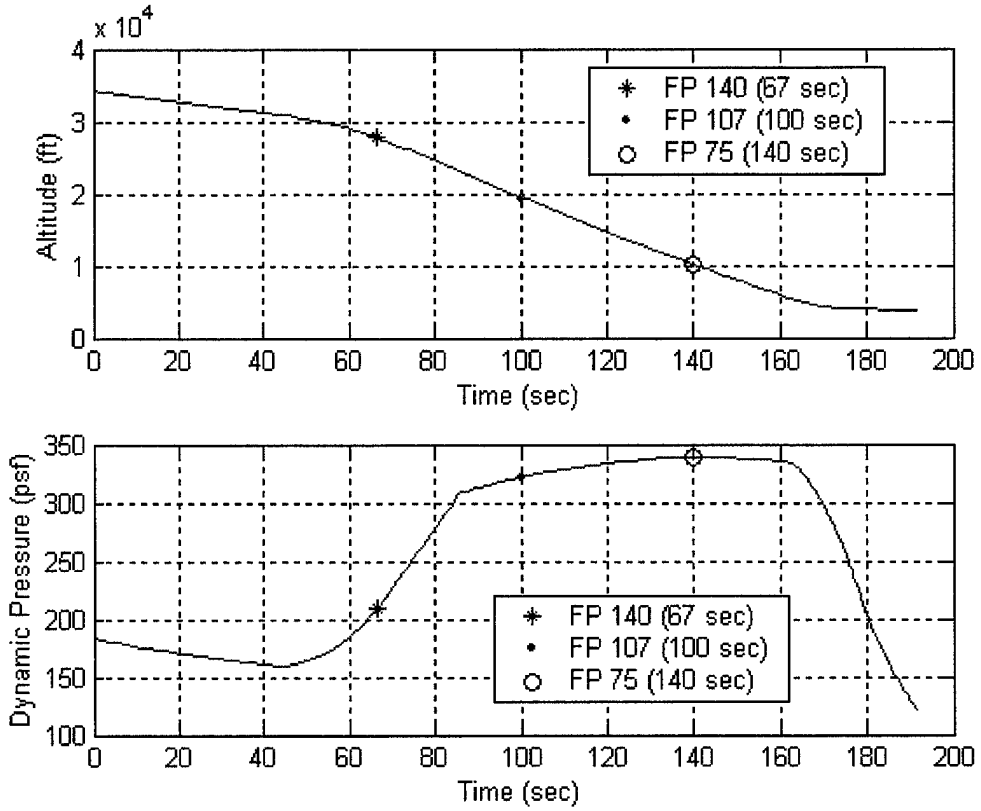

Fig. 21 Altitude and Dynamic Pressure Flight Profiles

Graphing the altitude and dynamic pressure in Fig. 21 provides a greater description of the selected flight points. The figures also show that for a given target bandwidth, a higher dynamic pressure contributes to a slightly faster convergence allowing for a shorter P horizon as seen in Fig. 19. The flight points with the lowest dynamic pressures should be examined when making a final P horizon selection. However, since the flight point 140 has a very different dynamic pressure than flight points 107 and 75 but similar P horizon lengths, it is noted that the dynamic pressure only makes a small contribution to the P horizon selection.

Finally, Fig. 19 and Fig. 21 suggest a correlation between the dynamic pressure and the optimal weighting for a given target bandwidth. Flight points 107 and 75 have similar dynamic pressures and converge to similar state weightings. Flight point 140 has a lower dynamic pressure, but converges to a higher state weighting in order to achieve the 1.0 rad/s target bandwidth. The effect of changing dynamic pressure is to change the plant dynamics, which then require a different weight. The convergence to a

particular weighting between flight points is not needed when selecting the P horizon, but it provides insight to the weighting strategy discussed in the weighting matrix section.

Before advancing to the next section a final argument is made for selecting 4 seconds for the prediction horizon.

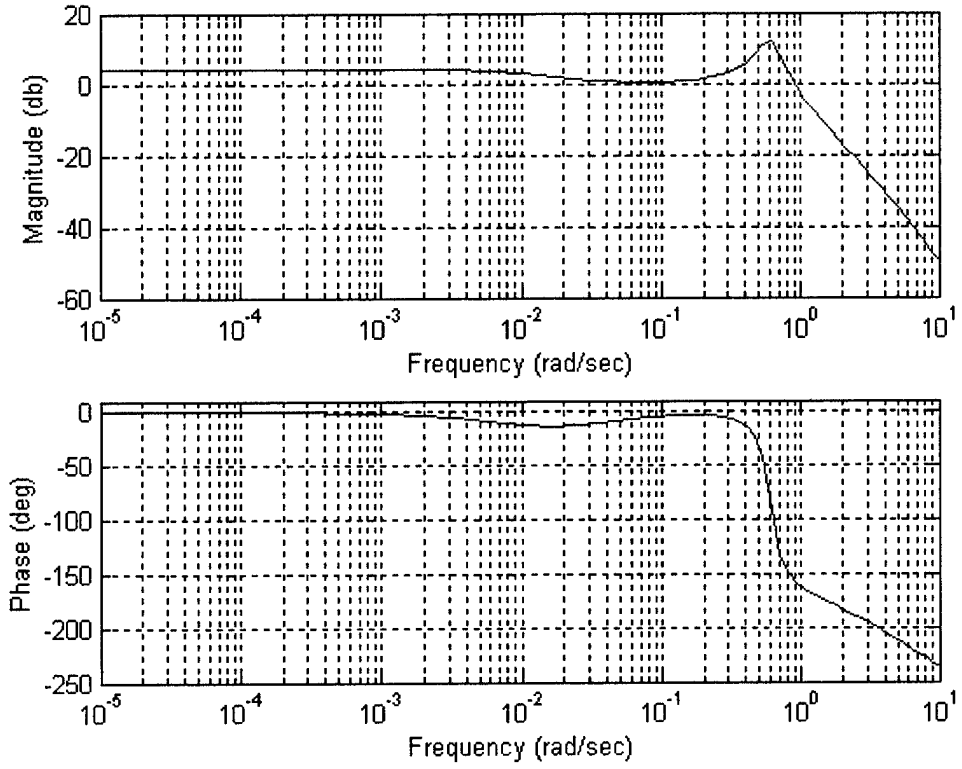

Fig. 22 Bode Plot for Flight Point 140 with P horizon = 1.4 sec

Fig. 22 shows a Bode plot of flight point 140 for the closed loop system from Fig. 18 with a prediction horizon of 1.4 seconds and the associated θ weighting of approximately 73. A P horizon of 1.4 seconds is too short for this system. The low frequency magnitude has drifted significantly from a desired system gain of 1 (0 db).

Fig. 23 also shows a Bode plot of flight point 140 for the same closed loop system, but with a prediction horizon of 4 seconds and θ weighting of 215. The low frequency magnitude is very close to the desired system gain of 1 (0 db). The curves are smoother and more consistent. Plotting the Bode plots for increasing P horizon show the low frequency system magnitude approaches 1 (0 db). A 4-second horizon allows

the internal plant to see a greater segment of the actual plant's dynamics than a 1.4-second horizon. This in turn generates a higher θ weighting, producing a higher feedback gain.

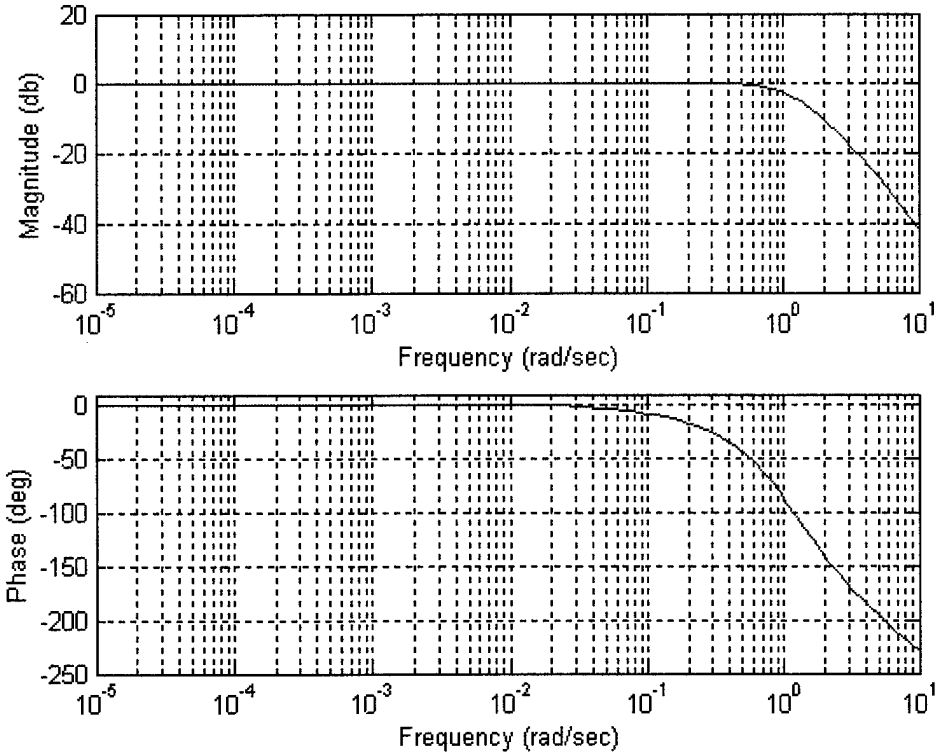

Fig. 23 Bode Plot for Flight Point 140 with P horizon = 4 sec

It is useful to evaluate additional flight points. Fig. 24 shows a prediction horizon of 1.4 seconds for flight points 107 and 75 with derived optimal θ weightings of 51.6 and 42, respectively. Both flight points have nonzero db magnitudes at low frequencies. Finally, with a 4 second prediction horizon, the weightings for flight points 107 and 75 are 220.9 and 226, respectively. The 4-second P horizon allows the target bandwidth to be achieved as shown in Fig. 25.

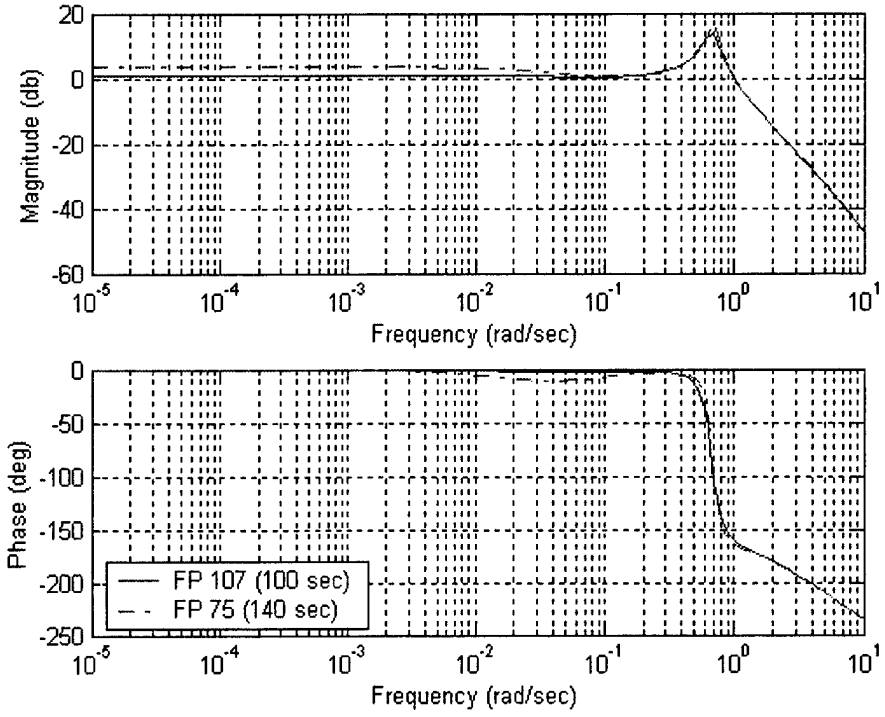

Fig. 24 Bode Plot for Flight Point 107 & 75 with P horizon = 1.4 sec

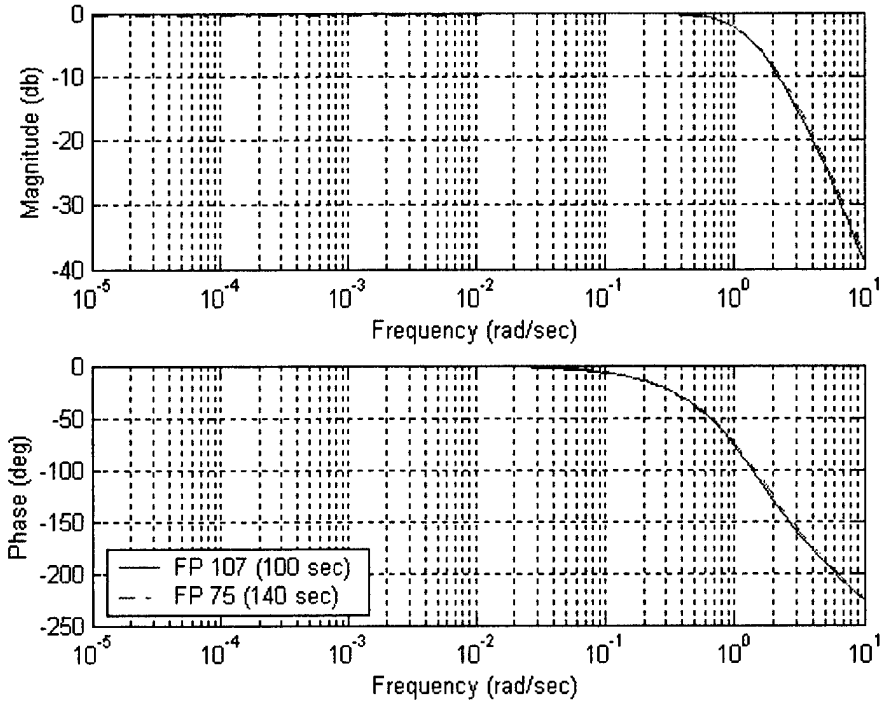

Fig. 25 Bode Plot for Flight Point 107 & 75 with P horizon = 4 sec

The P horizon selection algorithm is summarized as follows:

1. Start with a linearized continuous state space model.
2. Convert to a discrete model and use the new model to calculate the prediction matrices for the unconstrained closed form solution according to the mathematical derivation in section 2.1.3.2.
3. Assign fixed values to the u and Δu weighting matrices.
4. Define a desired bandwidth that equates to the required performance.
5. Use a search program that changes the state weighting value until the closed loop MPC controller achieves the desired bandwidth.
6. Loop through steps 2 through 5 for various P horizons.
7. Plot the state weightings and select a prediction horizon corresponding to the point when the weightings converge sufficiently.

4.2 Guidelines for Simulation Rates

The MPC, X-34 simulation has 4 significant rates: the prediction rate, inner loop rate, outer loop rate, and the simulation rate. The prediction rate is the frequency at which the MPC controller predicts the future output within the prediction horizon. The prediction rate is the time step used in solving the discrete differential equation routine, $\dot{x}(k+1) = Ax(k) + Bu(k)$. The inner and outer loop rates are the rates determining how frequently the inner and outer loop controllers generate new input commands for the plant. The longitudinal example has the structure of MPC_ALL where no inner loop exists, so only one loop or controller rate exists. The simulation rate is the rate at which the overall simulation is computed. This is to say the rate that the plant is updated and generates state values that are fed back to the controller. If the controller rate is slower than the simulation rate, the controller ignores the intermediate data. When the controller rate is equal to the simulation rate, all of the updated outputs are used.

The X-34 is a dynamically unstable vehicle on entry requiring high prediction, simulation, and loop rates to force stability. The high rates come at the cost of increased computational time. Selecting the rates is a trade off between performance and

calculation time. A simulation rate of 50 Hz and a loop rate of 10 Hz are found to yield an adequate balance between performance capability and computational time.

The prediction rate is closely related to the horizon length. While the horizon length is a function of the slow dynamics, the prediction rate is a function of the fast dynamics. A high prediction rate will capture the fast and slow dynamics, but a low rate will only capture the slow dynamics. For instance, a change in altitude is accomplished gradually and can be captured with a prediction rate as low as 2 Hz. A state representing faster dynamics, such as θ, may experience step commands requiring a quick response. A faster response stipulates a high prediction rate.

The prediction rate is found after the prediction horizon length is determined and is limited by acceptable computational time.

$$\frac{\text{Horizon Length}}{\Delta t} = \#\ \text{of Predictions} \qquad (62)$$

As the time between predictions decreases for a given horizon length, the number of predictions required grows quickly corresponding to increased simulation run time. Base functions can help to reduce the system complexity allowing for a faster prediction rate, but the base functions should not be depended on to make drastic reductions in simulation time. A prediction rate of 10 Hz is subsequently selected necessitating 40 predictions each time the MPC algorithm is called.

The prediction rate and loop rate must be determined prior to proceeding to the weighting matrices because the weighting matrices are dependent on the chosen rates.

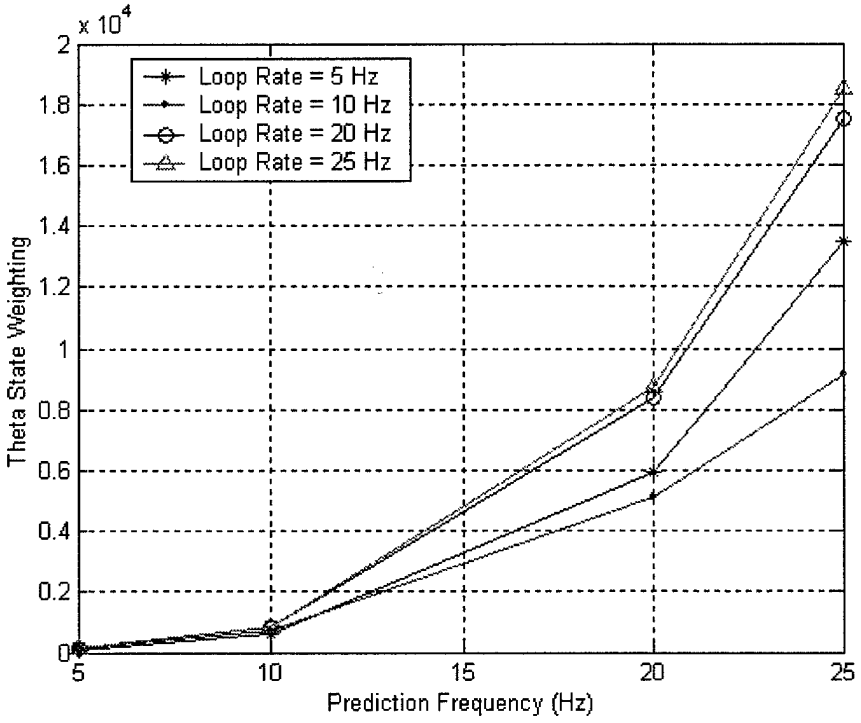

Fig. 26 State Weighting Variation With Simulation Rates

Fig. 26 plots the optimal state weightings for θ to achieve a target bandwidth for a given flight point. A significant variation in derived weightings is observed as the loop and prediction rates change. The weightings change more significantly by changing the predication rate, than by changing the loop rate.

The system control gains decrease with increasing prediction frequency causing the state weightings to increase to maintain the same level of performance. Should the prediction and loop rates be changed after selecting the weighting matrices, the system response may change significantly.

The increase in the optimal state weighting to maintain a desired bandwidth is a natural occurrence when the prediction rate is increased. The control in the unconstrained case is calculated using a closed form solution and reduces to a product of a gain matrix times the difference between the reference and feedback. The individual gains for each prediction must decrease when the frequency is increased because the gain matrix is calculated from the A, B, and C, matrices. Since the prediction rate is increased, the A and B matrices are more finely discretized, making their individual elements smaller. The error is assumed to be nearly constant over the prediction horizon for all prediction

rates considered. Each gain/error product now has a reduced value. MPC calculates optimal control values over the entire prediction horizon, but only applies the control at the first time step and recalculates a new complete set of control commands on the next iteration. Increasing the prediction rate, then decreases the applied control value. To increase the control to its value prior to increasing the prediction rate then requires an increased state weighting.

4.3 Guidelines for MPC Weighting Matrices

Once the prediction horizon and simulation rates have been determined, it is useful to find an appropriate profile for the weighting values in the trajectory. The procedure for finding the weighting values is similar to the procedure used for finding the prediction horizon. The most significant difference is the number of flight points used to find the weighting values. In the P horizon algorithm a single flight point is used primarily and there is a loop for changing P horizon lengths. In the weighting determination, a single prediction horizon length is selected and a loop is used for changing the flight points.

For the state weighting matrix, start with the linearized model for a selected flight point in the trajectory, convert to a discrete model, and calculate the prediction matrices for the flight point using the P horizon of 4 seconds already found. The values for u and Δu remain 0.5 and 1.0 as they were when finding the horizon. Next, a desired bandwidth is selected. One could set it at a single value as it was previously. However, since the entire trajectory is considered, it makes sense to lightly correlate the bandwidth with the dynamic pressure. The bandwidth and the dynamic pressure should be proportional. This coupling is introduced to help reduce the variation in the state weighting between flight points. Achieving a high bandwidth typically requires a higher state weighting, but a higher dynamic pressure allows for lower state weightings because the vehicle has a quicker response and more control authority. The proper relationship used between the bandwidth and the dynamic pressure is found through iteration. For this example, equation (63) relates the desired bandwidth to the dynamic pressure at the i^{th} flight point.

$$BW_i = 0.8 + \overline{q_i} * 0.001 \qquad (63)$$

This relationship grants a high bandwidth for higher dynamic pressures and a low bandwidth for lower dynamic pressures, effectively reducing the variation in state weighting values throughout the flight. The bandwidth varies no more than 0.2 rad/s between the minimum and maximum bandwidths, preventing drastic changes in the desired performance. The same θ state weighting search is done using the closed loop Simulink model of Fig. 18 as described in the P horizon selection procedure. The weighting is stored and the procedure is repeated for each flight point.

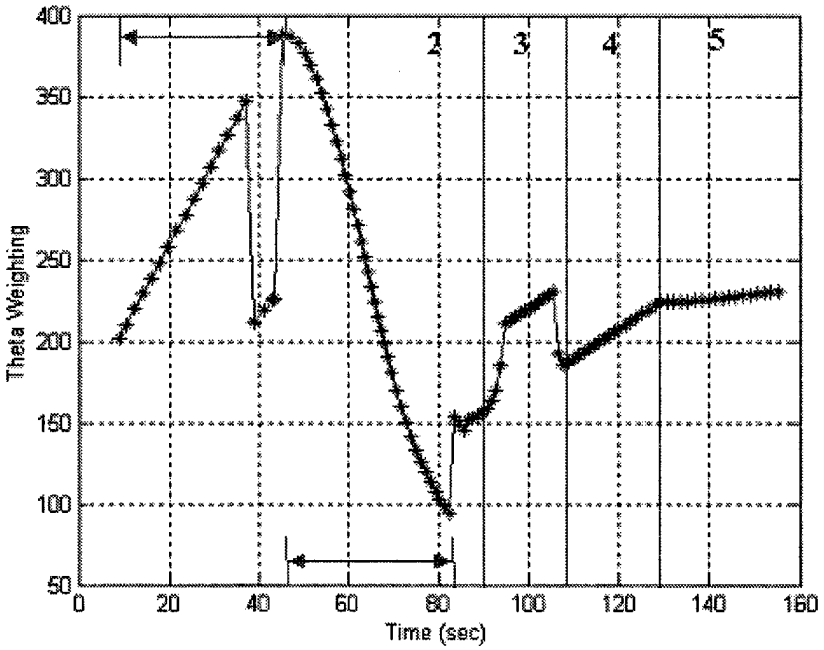

Fig. 27 State Weighting Profile

Fig. 27 shows the resulting state weighting flight profile for a section of flight where the asterisks represent the theta weighting values at the specific flight points. Starting from the left side of the graph, the first arrowed section corresponds to increasing weighting values caused partly by a decreasing dynamic pressure profile. This concept was seen in Fig. 19 and Fig. 21 in the previous section. The 3 flight points near 40 seconds of flight are slight inconsistencies in the optimal weightings possibly caused by step changes in commanded pitch rates. The second arrowed section of flight represents a steep increase in dynamic pressure and a subsequent decreasing weighting value. The

numbers and vertical lines designate the different flight phases for the trajectory. Phase 2 represents the level flight prior to the heading alignment phase. The step change in the optimal weightings of phase 3 corresponds to the banking portion of the trajectory. It is here that the longitudinal and lateral dynamic coupling is observed. The vehicle is banking and the vehicle's nose naturally drops. To counteract the dropping nose, an increased pitch command is required, corresponding to heightened θ weights. Section 4 is the post heading alignment section of level flight and section 5 is the approach and landing portion. The clear segmentation in Fig. 27 suggests that either the state weighting values should be scheduled with a look up table or more simply assigned constant weighting values for each flight phase. Single values for each flight phase lead to fewer weighting transitions in flight. Furthermore, experience shows that having exact optimal weightings at all times is not necessary for acceptable performance.

Once an initial weighting profile is found such as the one in Fig. 27, the u weight and desired bandwidth may be changed to help fine tune the system. It is not necessary in this example to change the Δu weighting as there are only three weightings in total and the weighting values in the cost function are relative, leaving only two degrees of freedom. Therefore, any fine-tuning in this example is confined to changing u and the desired bandwidth.

The θ state weighting algorithm is summarized as follows:

1. Start with a linearized state space model for a flight point.
2. Convert to a discrete model and use the new model to calculate the prediction matrices for the unconstrained closed form solution according to the mathematical derivation in section 2.1.3.2.
3. Assign fixed values to the u and Δu weighting matrices.
4. Define a desired bandwidth that equates to the required performance.
5. Use a search program that changes the state weighting value until the closed loop MPC controller achieves the desired bandwidth.
6. Loop through steps 2 through 5 for various flight points.

4.4 Base Function Guidelines

With the prediction horizon, simulation rates, and the weighting matrices chosen, base functions may be applied. Base function selection is less important than finding the P horizon and weighting matrices and in practice is no more than a selection of how best to employ the J_M matrix introduced in section 2.1.3.1. The P horizon and weighting matrix selection must be done correctly to achieve the required performance. In addition, poor selection of P and the weighting matrices may lead to instabilities or steady state errors. The weighting matrices, in particular, have far more design choices and combinations than the base functions. The different combinations may lead to a wide variation in performance, so proper selection of the weighting matrices is more complicated. Poor selection of the base functions, on the other hand, is rare and can be easily avoided in the design process by using the maximum degrees of freedom for the control. Changing the number or type of base functions should have little effect on the system performance and thus should only be used to reduce computational complexity. Base functions provide some benefit in the unconstrained problem, but it is limited because a closed form solution is used. The savings in time are realized when calculating inverses of the reduced matrices. In addition, a few matrix operations are saved when J_M is being multiplied. A greater computational savings is seen when base functions are applied to a constrained simulation. The constrained simulation uses the optimizer to search for the minimum cost. By using base functions, the number of free variables is reduced, allowing the optimizer to converge to a value more quickly.

To select the appropriate number of base functions, a comparison is made between the absolute cost using the base functions and the absolute cost without the base functions. Flight point 140 will be used in this example. For each flight point, start with the continuous linearized state space model. The A, B, C, and D matrices are converted to discrete and used to find the prediction matrices for a prediction horizon of 4 seconds. The weighting matrices do not change from those selected in 4.3. A θ error of 1 degree is introduced to the system equating to a 1 degree difference in the θ reference and the current θ state value. The target control and current control values are held error free. Next, the optimal $\underline{z}^*$ is calculated from the closed form solution. The cost function is described by equations (26), (28), and (30). It is assembled and written below with the disturbance terms omitted.

$$
\begin{aligned}
J = &\left[I_p u(k-1) + K_1 J_M \underline{z} - \underline{u}_T\right]^T W_u \left[I_p u(k-1) + K_1 J_M \underline{z} - \underline{u}_T\right] \\
&+ \underline{z}^T J_M^T W_\Delta J_M \underline{z} \\
&+ \left[S_x x + S_{u1} u(k-1) + S_u \underline{\Delta u} - \underline{r}\right]^T W_y \left[S_x x + S_{u1} u(k-1) + S_u \underline{\Delta u} - \underline{r}\right]
\end{aligned}
\tag{64}
$$

A loop is used to calculate the cost while varying the number of base functions from 1 to P. A P horizon of 4 seconds is used at a prediction rate of 10 Hz so 40 predictions are made. The first 5 base functions used are described in Fig. 6 with the pattern continuing for subsequent base functions. The base functions are applied by changing the J_M matrix from a 40 x 1 vector for one base function to a 40 x 40 matrix for 40 base functions as shown in (65).

$$
J_M = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \text{ for 1 base function} \qquad J_M = \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ 1 & 1 & .5 & .333 & \cdots & .026 \\ 1 & 1 & 1 & .666 & \cdots & .051 \\ 1 & 1 & 1 & 1 & \cdots & .077 \\ 1 & 1 & 1 & 1 & \cdots & .103 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & 1 & \cdots & 1 \end{bmatrix} \text{ for 40 base functions} \tag{65}
$$

The cost values for each number of base functions are compared to the cost when using $J_M = I$, the equivalent of not using base functions.

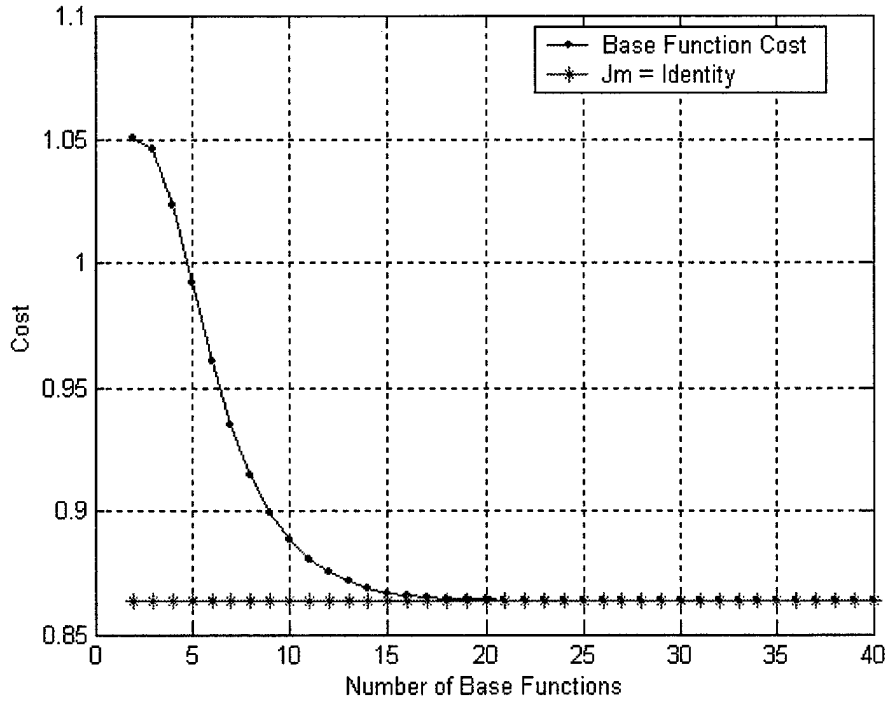

Fig. 28 Base Function Cost for 1-Degree θ Reference Error

When the maximum degrees of freedom are available for solving the cost function, the minimum cost is obtained. The curve in Fig. 28 represents the cost for various base functions ranging from 2 to 40 functions. The cost for only 1 base function was so high it could not be placed on the plot. Only using 1 base function over constrains the problem and should not be considered an option. The following metric is used to normalize the error with using base functions to help the designer determine the appropriate number of base functions.

$$\%\,\text{Cost Error} = \frac{(J_B - J_I)}{J_I} * 100 \qquad (66)$$

where J_B is the cost associated with a specific number of base functions and J_I is the cost when base functions are not used.

Tab. 5 displays the error for using a variety of base functions.

Number of Base Functions	% Cost Error
2	21.66
5	14.87
10	2.88
15	0.35
20	0.03
30	0.01
40	0

Tab. 5 Base Function Errors

15 base functions are selected. It is conservative because little performance loss would likely be seen with as few as 10 base functions. For a thorough selection, the cost and errors should be calculated for all flight points in the trajectory.

Note that there is no error when 40 base functions are used. While J_M used as an identity and J_M used as P base functions are two different matrices, the cost is mathematically equivalent because they are both basis matrices with P degrees of freedom.

The base function selection algorithm is summarized as follows:

1. Start with a linearized state space model for a flight point.
2. Convert to a discrete model and use the new model to calculate the prediction matrices for the unconstrained closed form solution according to the mathematical derivation in section 2.1.3.2.
3. Assign fixed values to the state, u, and Δu weighting matrices.
4. Introduce an error between the state reference and current value.
5. Calculate the optimal solution $\underline{z}^*$ and the associated cost.
6. Loop through steps 2 through 5 for various J_M configurations.

7. Calculate the percent error related to using each number of base functions.

4.5 System Results

The general design of the simplified longitudinal example is complete. It is now necessary to evaluate the controller's performance and stability. First, it is shown how the system at flight point 140 responded to a 1-degree step change in the reference theta command at 5 seconds. Fig. 29 shows how the system responded to the step change with a prediction horizon of 4 seconds and a target bandwidth of 1.5 rad/sec. The target bandwidth was changed slightly to achieve a faster and more accurate response while fine-tuning the system. The optimal state weightings also changed, but followed directly from the bandwidth change as dictated by the procedure in section 4.3. This is the only design choice to change from the design described in the previous sections. The system is still operating with a simulation rate of 10 Hz, using 15 base functions, and has u and Δu weightings of 0.5 and 1.0.

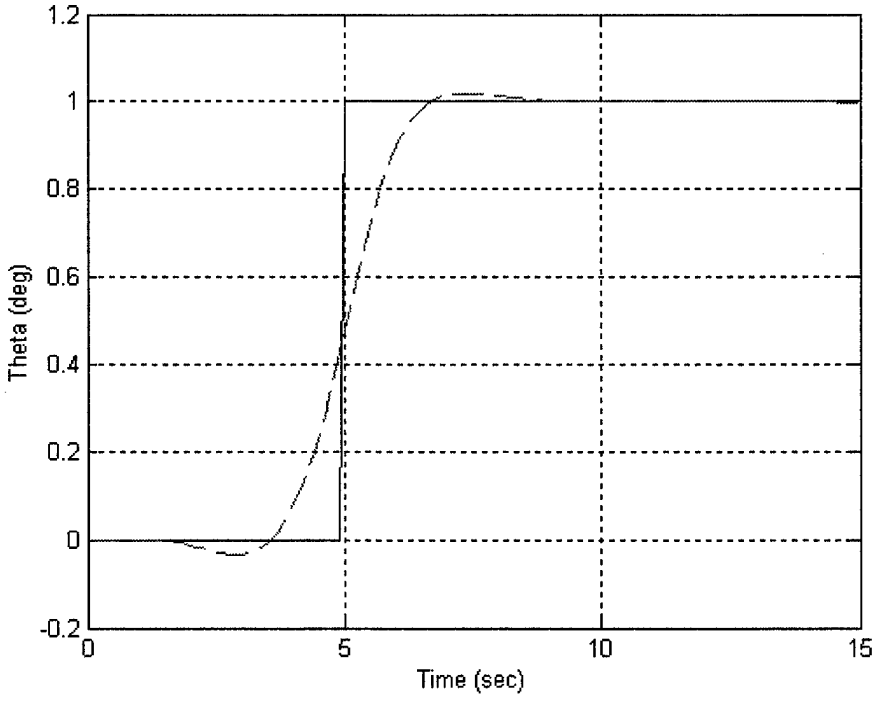

Fig. 29 System Response to a θ Step Input

The solid line is the command and the dashed line is the response. The system begins responding at a time of 1 second when it first sees the step change at the end of its 4-second prediction horizon. The movement at 1 second is subtle and then becomes more aggressive as more of the step input is seen in the prediction horizon. In addition, because the controller knows the plant dynamics, MPC knows how quickly the system can respond to the new value, and thus, knows the appropriate time to move more aggressively. From Fig. 29 it can be seen that the system is nonminimum phase by its initial dip in the negative direction.

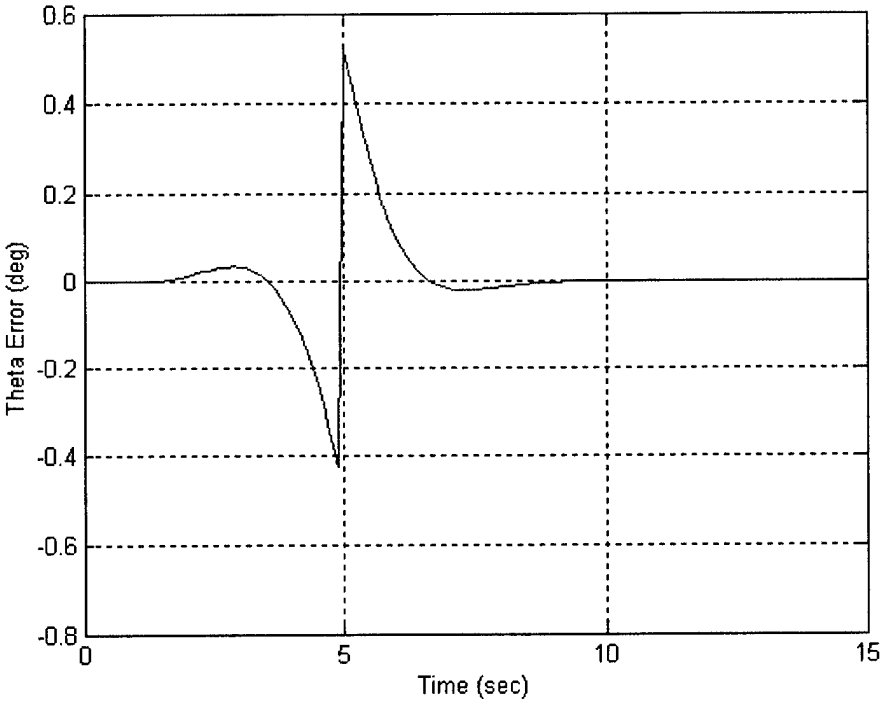

Fig. 30 System Tracking Error to a θ Step Input

The above error plot shows MPC's anticipative behavior and the acceptance of error before the command to reduce the peak error. The proactive movement is especially advantageous for reducing the peak errors for nonminimum phase dynamics. Without the anticipation, the controller would have to respond after receiving the command. The nonminimum phase nature would then push the response in the negative direction giving an absolute peak error greater than unity at the beginning of the maneuver. Using the anticipation to its advantage, the MPC accepts some error prior to the step and keeps the peak error to about 0.5 degrees.

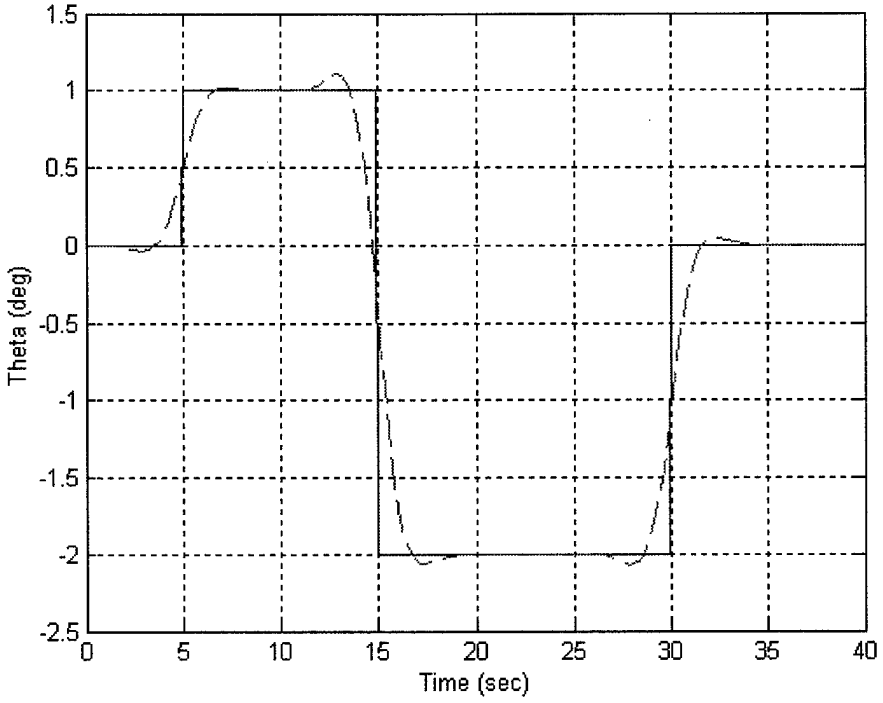

Fig. 31 System Response to a Theta Doublet

Fig. 31 shows the response when a commanded doublet is input to the system. The anticipation and the nonminimum phase performance can be seen in the doublet as with the step input. The nonminimum phase undershoots and final overshoots are proportional to the severity of the instantaneous command change and are a function of the bandwidth. The overshoot deviations at 15 and 30 seconds are 3 and 2 times the deviations seen at 5 seconds.

In addition to responding well to changes in the reference, the MPC successfully stabilized the unstable plant. The continuous plant poles and the closed loop poles are plotted in Fig. 32. The closed loop system has an additional pole caused by a unit delay block in the MPC block of the closed loop Simulink diagram. A unit delay is needed to obtain the previous control value $u(k-1)$. The previous control is required to solve the cost function for the optimal Δu.

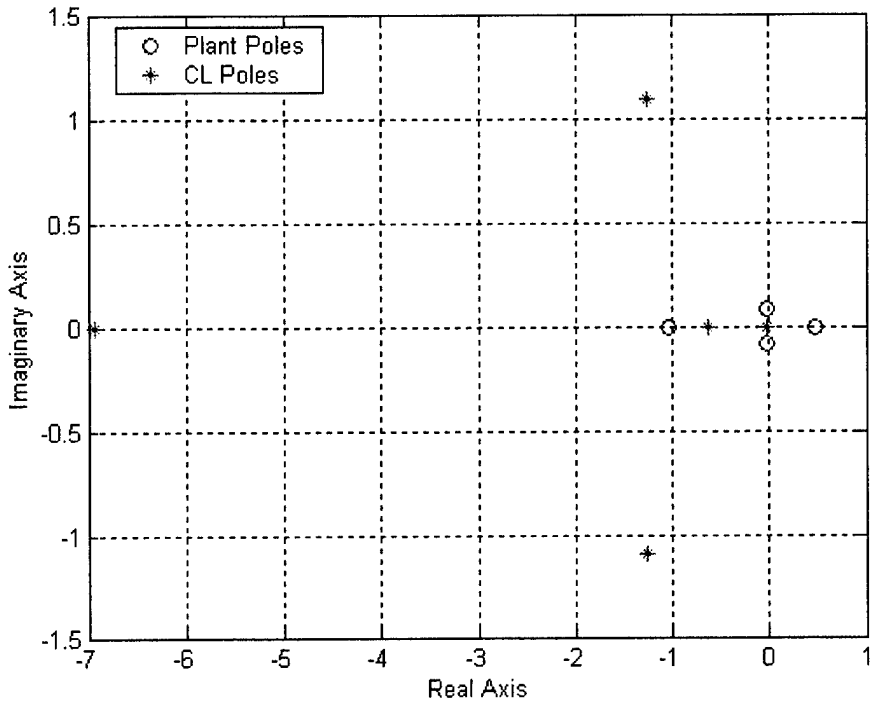

Fig. 32 Plant and Closed Loop Pole Location for Flight Point 140

Fig. 33 shows the closed loop pole locations for the entire trajectory. For every flight point, the MPC stabilizes the plant forcing all of the poles into the left hand plane. It also shows the distinct general locations of the short period and remaining poles.

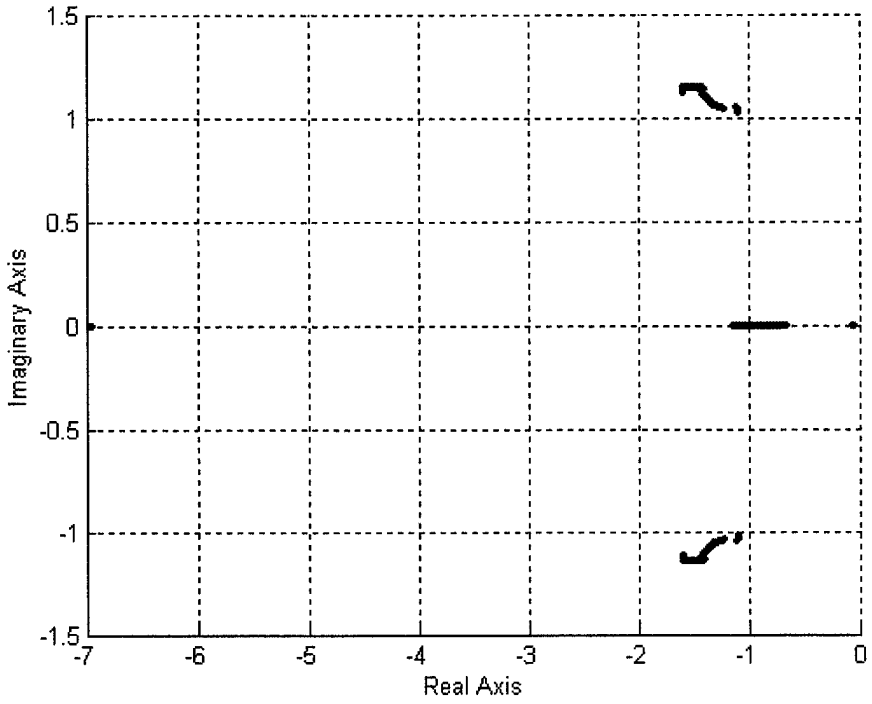

Fig. 33 Closed Loop Pole Location for the Full Trajectory

It is expected that for varying dynamics and a desired bandwidth relating to changing dynamic pressure that the closed loop poles will move through various locations in the left hand plane. Fig. 34 shows a zoomed in view of the closed loop short period poles for the full trajectory. Not only do the poles move significantly, but also there are slight discontinuities in the pole locations throughout the flight. The discontinuities arise from the minimizing nature of the cost function. The poles are placed for each flight point wherever the cost is minimized without regard to the previous pole location. For most points, the pole location follows a trend from one point to another. However, discontinuities may appear when the dynamics change between flight points and when the weightings change between flight phases.

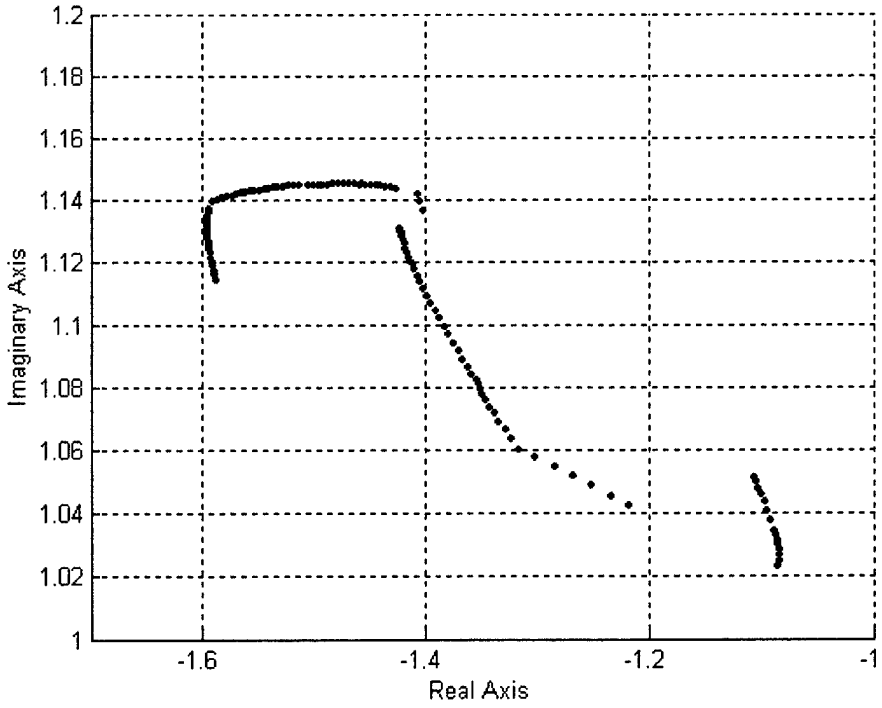

Fig. 34 Short Period Pole Location for Full Trajectory

Fig. 35 shows a further zoomed view of the short period pole location for a 30-flight point segment of the trajectory ending at flight point 140. It shows a consistent trend in the pole location for this flight segment. The pole location for this section of flight starts with the point closest to the origin and gradually moves radially away from the origin.

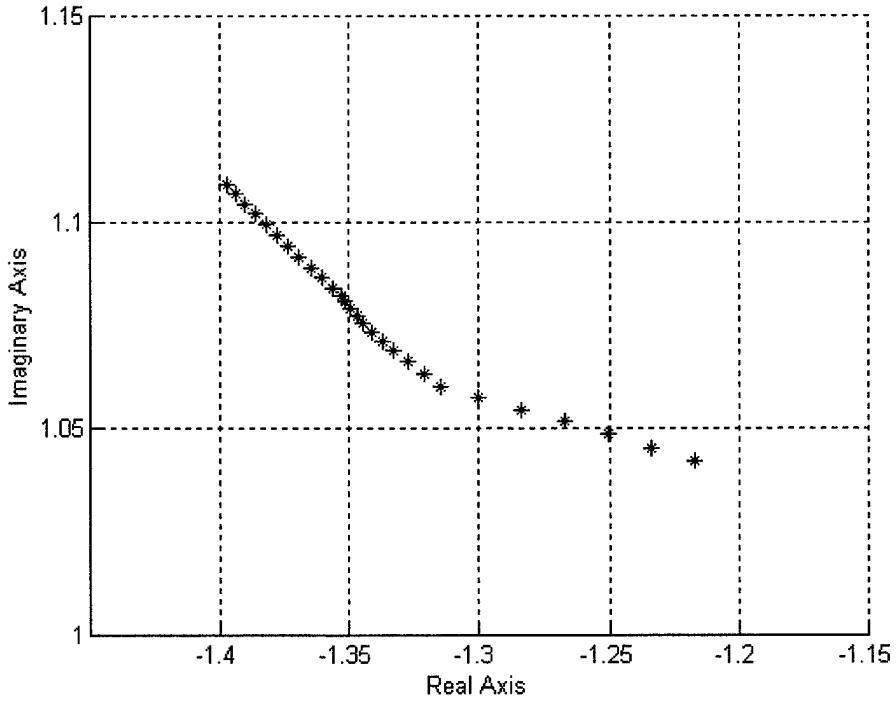

Fig. 35 Short Period Pole Location for 30 Flight Points

It is also desirable to find the gain and phase margin associated with the closed loop system for flight point 140. The loop is first broken at the plant input to get the open loop transfer function (Fig. 36). The Matlab command "MARGIN.m" is then used to find the gain and phase margins and the frequencies when the magnitude is 0 db and the phase is + or – 180 degrees.

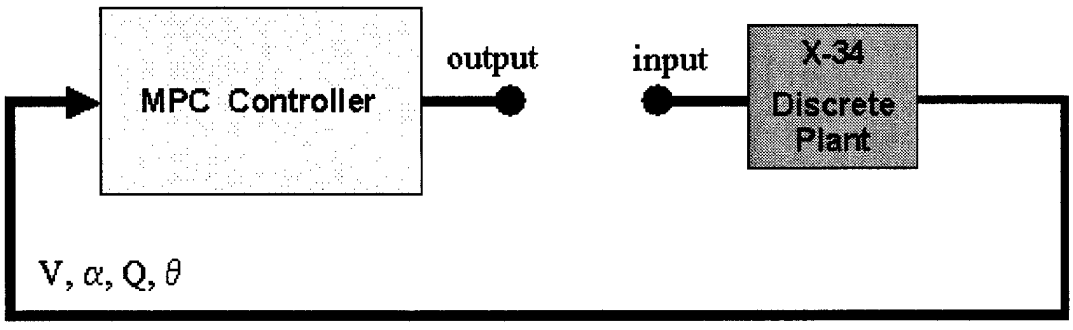

Fig. 36 Longitudinal Example Architecture Open Loop

The gain margin is found to be 0.19 (14.42 db) at a frequency of 0.49 rad/sec. The phase margin is 48.61 degrees at a frequency of 2.24 rad/sec. These margins may be

verified in many ways. The Nyquist method is presented first, followed by directly applying the gain and phase margin to the system to bring the system poles to the verge of instability.

A Nyquist plot is shown in Fig. 37 for the open loop system and shown zoomed in on the origin in Fig. 38.

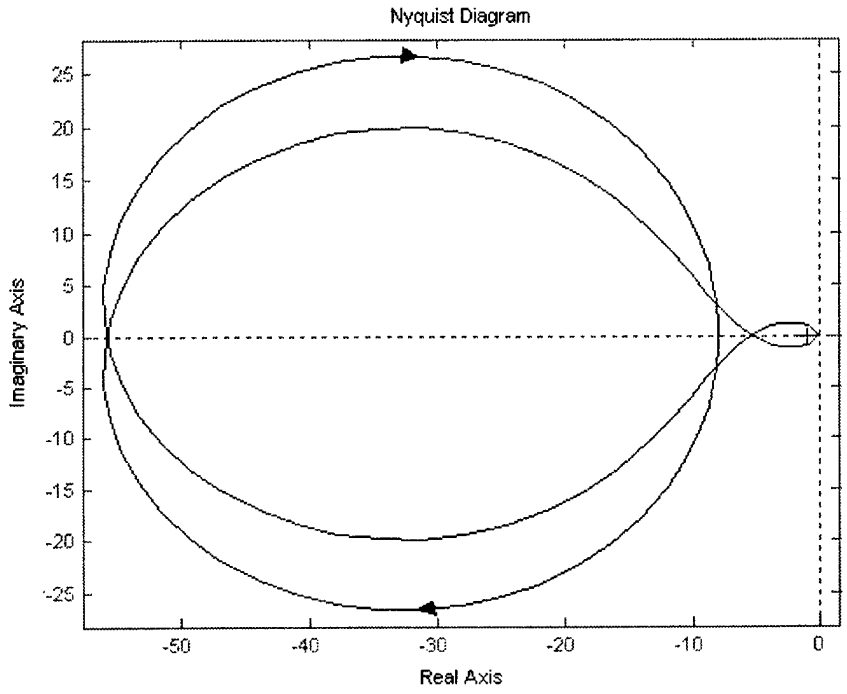

Fig. 37 Nyquist Plot of Open Loop System

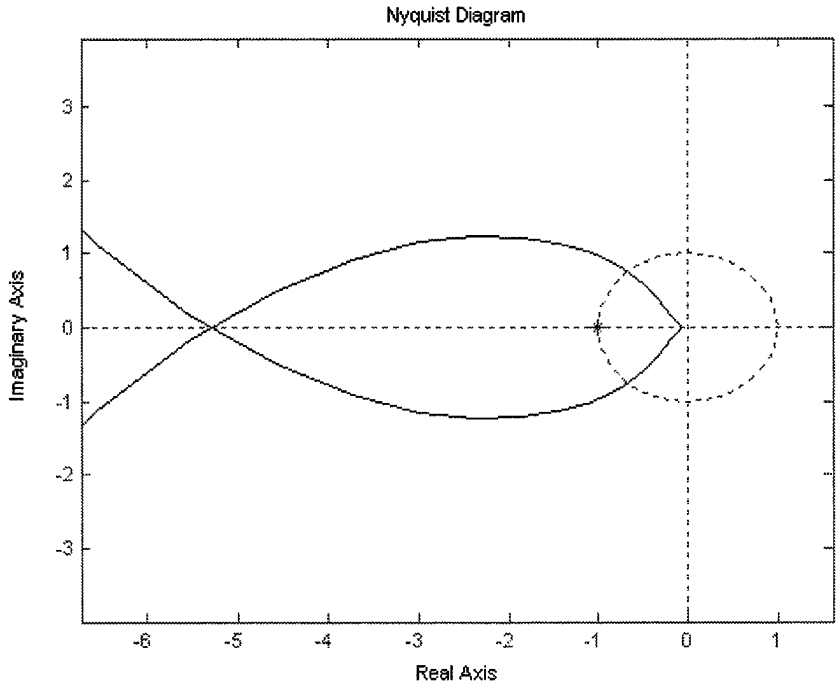

Fig. 38 Zoomed Nyquist Plot of Open Loop System

A circle of radius 1 is shown as a dotted circle. From the Nyquist plot the gain and phase margins can be verified. The plot crosses the real axis at –5.29. The gain margin is then 1/-5.29 or 0.19 (14.42 db) as stated above. Furthermore, the plot intersects the unit circle at approximately (-0.66, 0.75i) giving a phase margin of about 48.59 degrees.

The gain and phase margins are now input as shown by Fig. 39 into the model where the loop was broken. The gain is applied by multiplying by a simple gain block. The phase is applied through a second order Padé approximation of the time delay. A Padé approximation does not change the system's gain, but does impose a time delay. The second order Padé approximation loses validity quickly beyond frequencies of 10 rad/sec. Because the phase margin is found at 2.24 rad/sec, the second order Padé is sufficiently accurate [Ref. 16]. The second order Padé is described in the Laplace domain by:

$$e^{-\tau s} = \frac{1 - p_1(\tau s) + p_2(\tau s)^2}{1 + p_1(\tau s) + p_2(\tau s)^2} \qquad (67)$$

where τ is the time delay found through the relationship

$$\Phi_M = \omega_p \tau \qquad (68)$$

Φ_M is the phase margin and ω_p is the frequency where the phase margin is found.

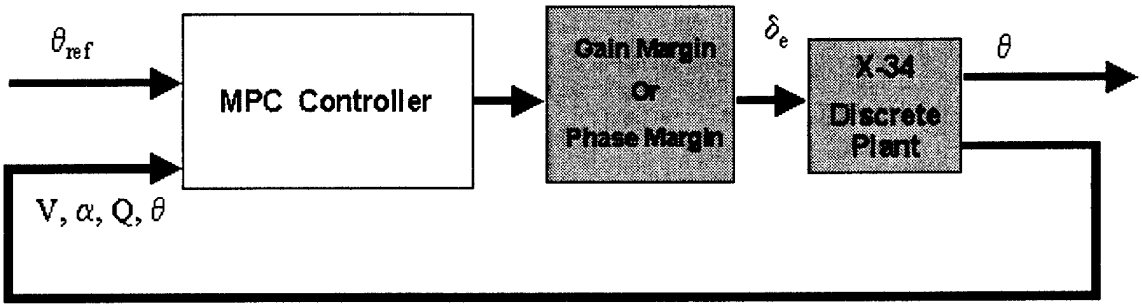

Fig. 39 Longitudinal Example Architecture With Gain or Phase Margin

When the gain or phase margin is introduced to the system, at least one pole is forced to the imaginary axis as that is the threshold for stability. Any increase in the gain or phase margin past that point would then drive the system unstable. The first graph in Fig. 40 plots the closed loop poles and the closed loop poles with the gain margin applied. The second graph plots the closed loop poles and the closed loop poles with the phase margin applied using the Padé approximation. The Padé introduces two additional fast poles to the system. In both plots the closed loop complex conjugate pair of poles at (-1.26 +-1.09i) are forced to the imaginary axis.

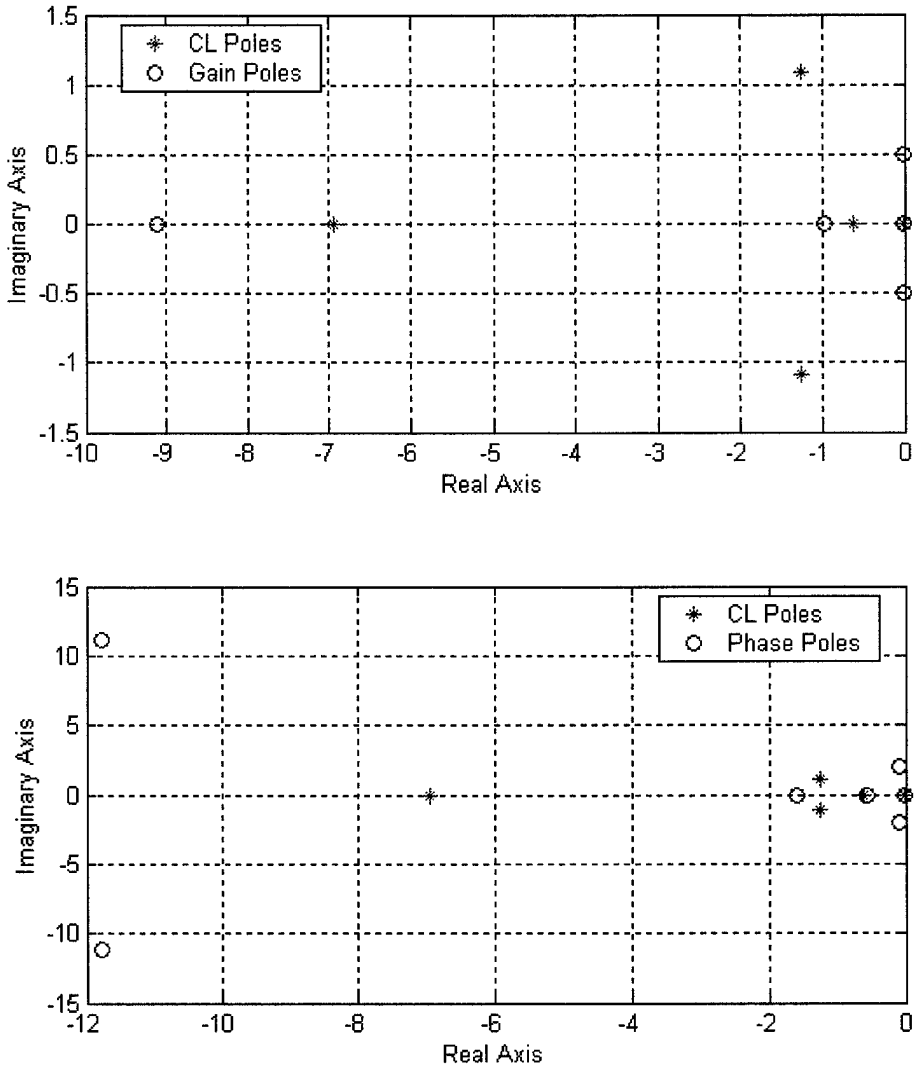

Fig. 40 Closed Loop Pole Location with Gain or Phase Margin Applied

This section concludes with a graph showing the gain and phase margins for the flight points for the trajectory. Some step changes are seen at the same places as seen in the optimal state weightings, however, the margins are never less than 10 db and 44 degrees.

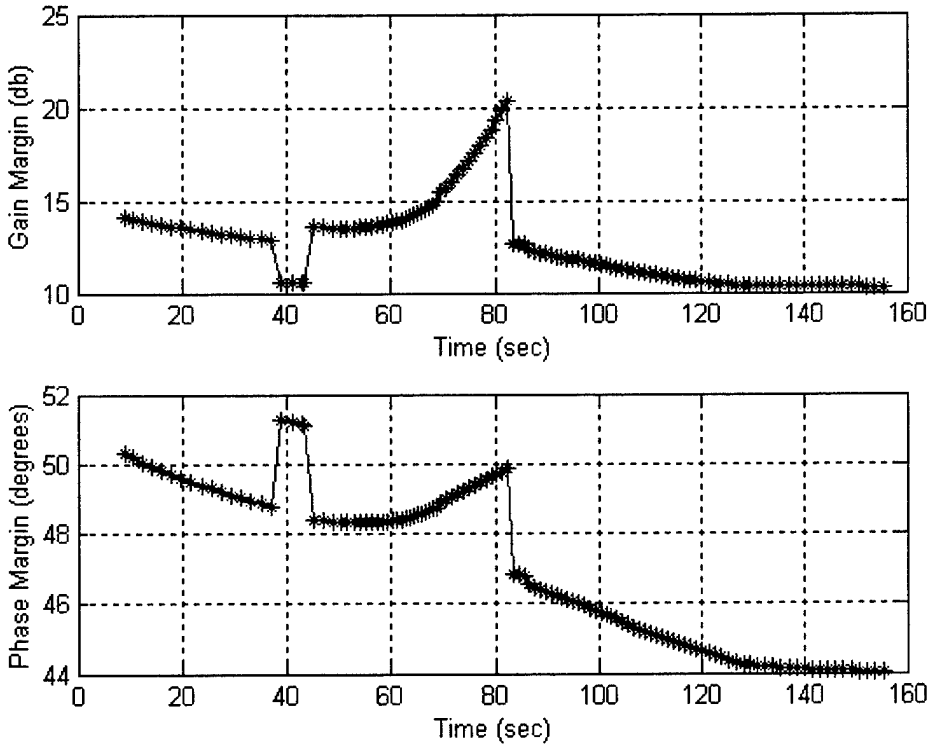

Fig. 41 Gain and Phase Margins for Various Flight Points

Chapter 5

MPC Application to the X-34

The longitudinal example aids in designing the full simulation in two ways. First, it provides procedures a designer may use to select the prediction horizon length, simulation rates, state and control weightings, and the appropriate number of base functions. These procedures have been found to work for the longitudinal case considered, but are not guaranteed to work for every system. The concepts discussed may be applied to other systems, but some variations may be in order. For example, when obtaining the prediction horizon, the longitudinal case defined a target bandwidth and searched for a θ weighting that would give the desired bandwidth. The system then looped through the P horizon lengths until the weightings converged. The designer may find that bandwidth is not the appropriate parameter to optimize to in every situation. For some applications it may be better to define desirable sections in the left hand plane for the closed loop poles to be placed and then search for the W_y matrix weighting that achieves this goal. The procedures presented merely provide one avenue a designer may pursue when selecting the various parameters.

The second function the longitudinal example performs is to give some insight to expected values for the design parameters. For example, the longitudinal example showed converging weighting values between 3 and 4 seconds for the P horizon. One would expect the full 12 state MPC_ALL simulation to require a horizon length of the same order. The weighting values for the full simulation, however, may vary quite significantly from the example as all of the states are controlled and all 4 control surfaces are used.

Because the longitudinal example lacks a SAS, it follows the architecture of MPC_ALL. The following sections apply the design criteria from the longitudinal example to the MPC_ALL architecture. The design criteria may be applied to the MPC_SAS architecture as well, but it is not discussed here for brevity.

5.1 Prediction Horizon Selection

The prediction horizon for the MPC_ALL simulation is selected to be 3 seconds. This value is taken from knowledge of the longitudinal example and from analyzing multiple runs using the full nonlinear plant dynamics. The P horizon must be long enough to capture both slow and fast dynamics of the system. The longitudinal example has a plant with dynamics from the states V, α, Q, and θ. The states from the longitudinal example are representative of both fast and slow dynamics. Fig. 19 shows a convergence of the fast dynamics at about 1.5 seconds. It also shows a convergence in the slow dynamics between 3 and 4 seconds. The slow dynamics of the longitudinal channel are on the same order as the slow dynamics of the lateral channel so, it is reasonable to conclude that the slow lateral dynamics are captured in the 3 second P horizon. The following figures provide additional support for the selection of 3 seconds for the P horizon.

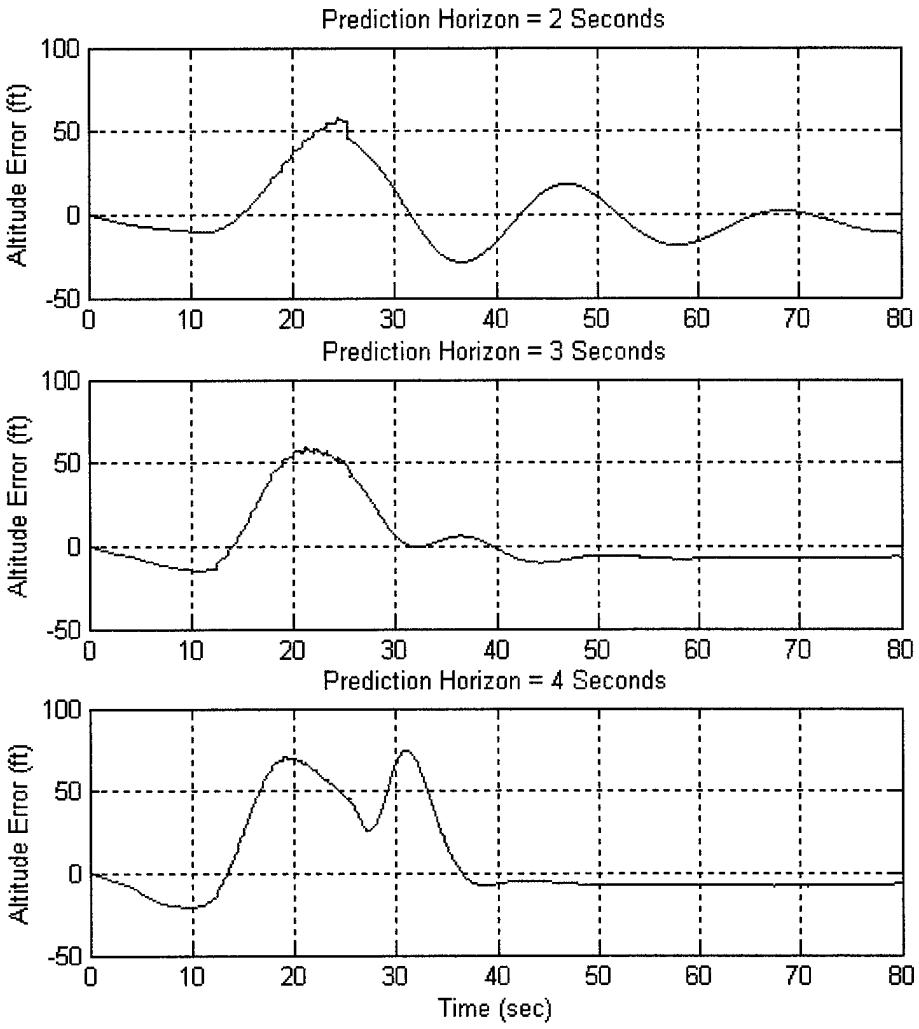

Fig. 42 MPC_ALL Altitude Error Plots for Varying P Horizons

Fig. 42 shows three plots of the error from tracking the altitude for the single bank trajectory. The figures do not show the full trajectory. Instead, they are zoomed in on a portion of flight just before the bank, including the bank, and continuing to the approach and landing phase where the final errors are corrected. The three plots differ only by the length of the prediction horizon. The three prediction horizons selected are 2, 3, and 4 seconds. The damped oscillation in the plot for 2 seconds shows that the horizon length is too short. The plots for the 3 and 4 second horizons show sufficient information is obtained as the error induced from the bank converges to an error of less than 7 feet, which is left to be eliminated in the remaining portion of the A/L flight phase.

While too short of a P horizon leads to poor performance, it is also undesirable to select a P horizon longer than necessary. The computation time to arrive at the optimal control values increases with increasing P horizon length because the matrices quickly grow in size. The internal plant model uses a linearized approximation of the actual plant. In section 3.6 the linearization process is discussed and it is assumed that $\dot{x}_0$ is constant over the horizon. As the length of the horizon increases, the constant $\dot{x}_0$ assumption loses validity. Fig. 43 shows the same portion of flight as Fig. 42, but for the tracking of bank angle. During the immediate post bank transition segment of flight, it is seen that the constant $\dot{x}_0$ assumption loses validity and some oscillation is observed before the error is eliminated. The banking overshoot increases with increasing horizon length. The 4-second horizon leads to an overshoot of nearly 60 degrees. Such an overshoot is unacceptable and may be reduced by simply reducing the horizon length. Therefore, a 3-second prediction horizon is selected.

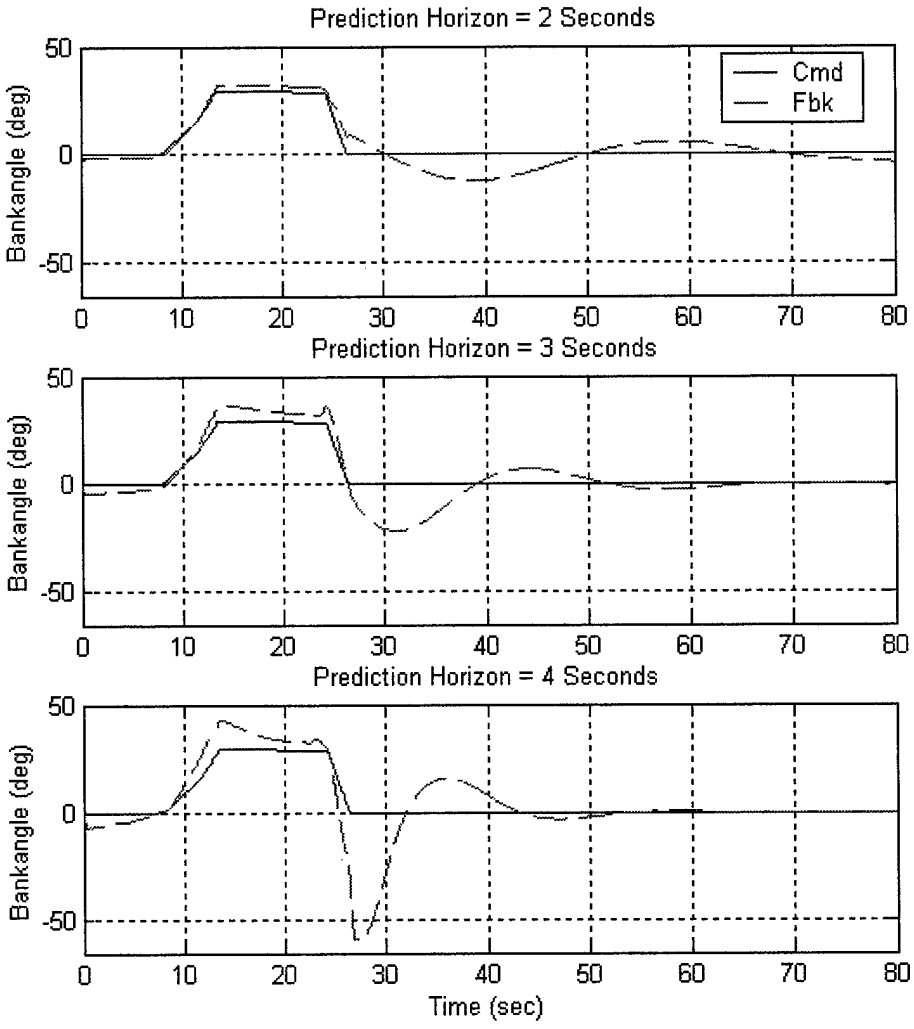

Fig. 43 MPC_ALL Bank Angle Tracking for Varying P Horizons

5.2 Simulation Rates

The longitudinal example introduced 4 significant rates: simulation rate, prediction rate, inner loop rate, and the outer loop rate. The construction of the simulation requires the simulation rate to be equal to or faster than the loop rate. To allow a greater flexibility in the selection of the loop rate, a simulation rate of 50 Hz is selected. Additionally, a simulation rate higher than the loop and prediction rates has a negligible penalty on computation time and nearly no change in the system performance. When the

simulation is run at a higher rate than the loop and prediction frequencies, the simulation collects feedback measurements more frequently, but only applies different control values at the loop rate. Between the control updates, a constant control is applied and smoothed to the simulation rate in the data collection.

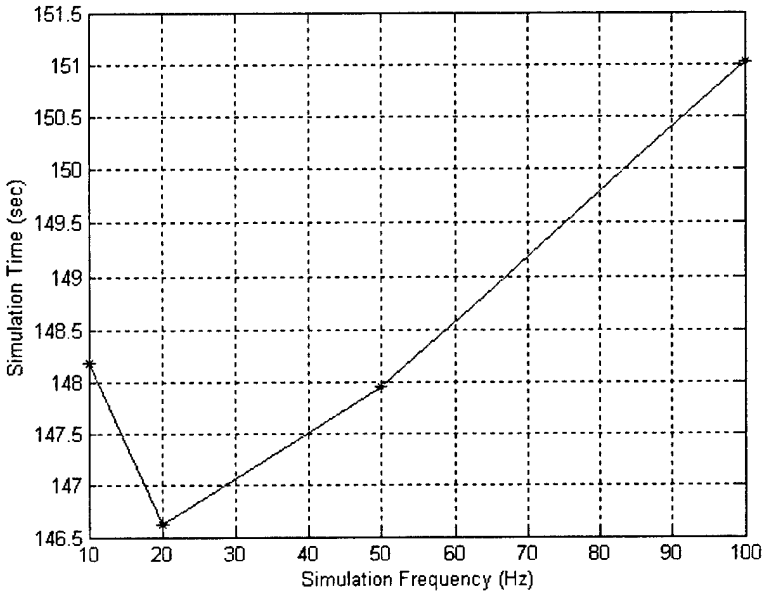

Fig. 44 Computational Time for Varying Simulation Rates

Fig. 44 shows the computational time required to simulate 20 seconds of flight while varying the simulation rate and holding the loop and prediction rates constant at 10 Hz. Base functions were not used in generating this data. The variation in simulation time is attributed to slight changes in the computer's efficiency from run to run.

With the simulation rate fixed, the loop rates may be selected. The MPC_SAS has separate inner and outer loop rates dictating the frequency that the LQR and MPC controllers issue new commands. Alternatively, the MPC_ALL architecture does not have a separate inner loop SAS. For MPC_ALL, only one loop rate exists. Procedures for selecting both inner and outer loop rates for the MPC_SAS are omitted from discussion as the design focus is on MPC_ALL. However, a trade off analysis similar to that presented for the MPC_ALL may be applied to MPC_SAS for finding these values.

The loop and prediction rates are found simultaneously. From the longitudinal example it was shown that both rates affect the weighting matrices. Because the system

performance is directly tied to the weighting matrices, the loop and prediction rates must be selected before the weightings are derived.

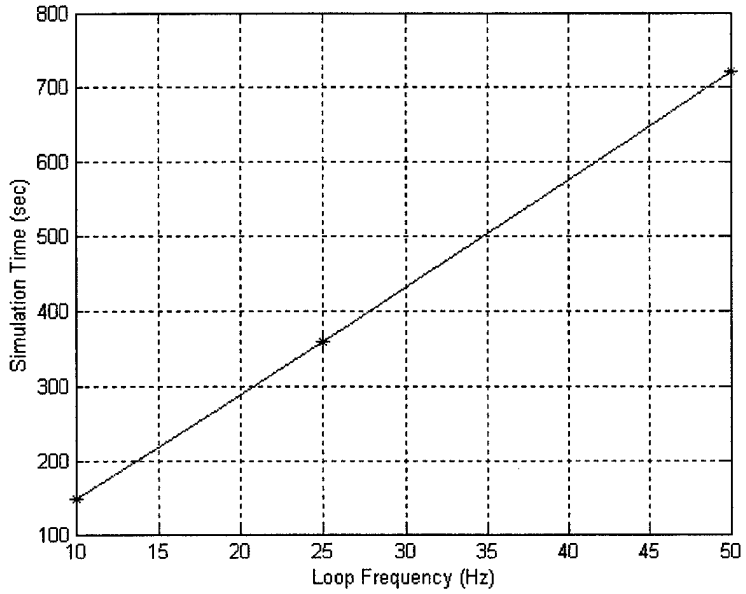

Fig. 45 Computational Time for Varying Loop Rates

The simulation run time for 20 seconds of flight is plotted in Fig. 45 for the loop frequencies, 10 Hz, 25 Hz and 50 Hz. The prediction and simulation rates are set to 10 and 50 Hz, respectively. The figure shows nearly a linear increase in computing time with increasing loop frequency. Loop rates of 2 and 5 Hz were too slow and the vehicle went unstable within the 20-second interval. The simulation for 2 Hz failed at about 5.5 seconds of simulated flight with a computation time of about 11 seconds. Similarly, the 5 Hz simulation failed at 8.5 seconds with a required run time of about 33.4 seconds. Had these two simulations completed the 20 seconds of flight, the final calculation times would have roughly fit into the above linear depiction. A linear dependence on computation time with loop frequency is a reasonable outcome as the MPC calculation is the time consuming portion of the simulation. If the MPC calculation is called twice as frequently, it is logical that it would take approximately twice as long to complete a given flight.

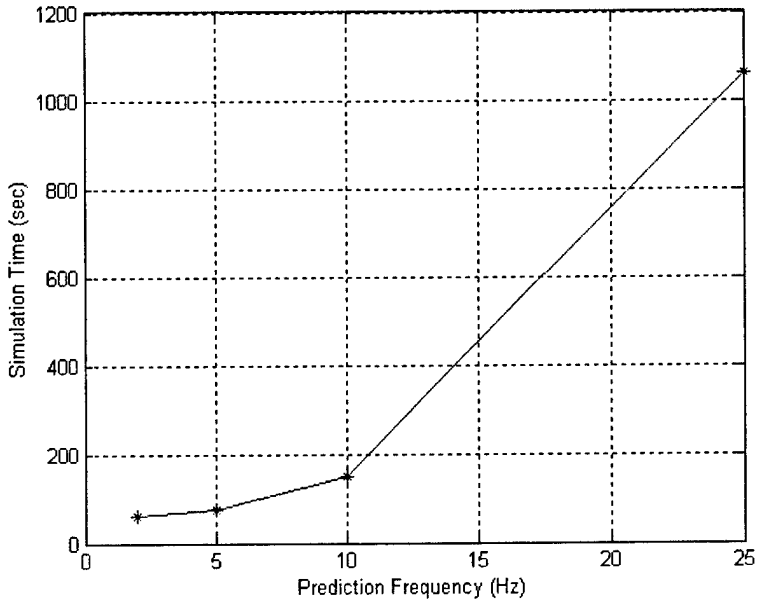

Fig. 46 Computational Time for Varying Prediction Rates

Fig. 46 shows the computational time for 20 seconds of flight while varying the prediction rate. The loop and simulation rates are set to 10 Hz and 50 Hz, respectively. The computational time increases very rapidly as the prediction rate increases. This rapid increase is caused by the increasing size of the prediction matrices. Finding matrix inverses are known to be very computationally intensive as the matrices grow in size. Subsequently, the simulation run times greatly increase. The data shown in the above two figures was generated without using base functions.

Because loop rates of 2 Hz and 5 Hz lead to instabilities, a loop rate of 10 Hz is the minimum frequency that should be chosen. 25 Hz, however, leads to long computational times. The computational times become too great for prediction rates greater than 10 Hz. Considering Fig. 45 and Fig. 46, the prediction and loop rates are both selected to be 10 Hz.

5.3 MPC Weighting Matrices

Finding the appropriate weighting matrices is a challenging process. Some trial and error is required, making it a time consuming task. Of the parameters discussed, this is the least intuitive to select. The W_y matrix penalizes the deviation of the vehicle's state

from the state reference setpoint. W_u increases the system cost for departures in the aerosurface position from the control reference setpoint. Finally, the W_Δ matrix penalizes changes in the current control value from the value at the previous time step. Each weighting matrix is a diagonal matrix with one nonnegative entry per state or control input. A weighting of 0 in the state matrix means there is no penalty for that particular state deviating from its reference; MPC ignores control of that state. A nonzero weight tells MPC to track the corresponding state. In general, the greater the weighting, the more closely MPC tries to track the state. Similarly for the control matrix, a 0 weighting places no penalty on the aerosurface's position, allowing it to move freely, and a positive weighting corresponds to tracking the reference control input. As the weighting on the control surface approaches infinity, MPC forces the aerosurface to the commanded trim condition. An infinite weighting reduces the system robustness, as the vehicle cannot move the aerosurface to zero out a state error from a disturbance. Positive values in the change in control weighting matrix penalize rapid or sharp changes in the control signal.

In the longitudinal example, the weightings were found quickly as the weighting matrices were reduced to scalar values and the control weights were held constant allowing for only one unknown state weighting. For the MPC_ALL simulation, such simplification is more difficult as all 12 states are tracked by 4 control surfaces and 4 control input rates resulting in selecting 20 unknown weights. A multi-stage process should be taken when obtaining the weights for such a complex problem. For the purposes of this research a four-step procedure has been followed. First, Bryson's method is used to obtain initial weightings. Trial and error is then performed to achieve a flyable model. Next, a pole placement procedure is employed. Finally, additional trial and error is used to fine-tune the system.

The first step in populating the weighting matrices stems from an LQR technique called Bryson's method. An LQR servo controller is an optimal control technique solving a cost function similar to the cost function used for MPC. The LQR system starts with an LTI state space model and minimizes the following cost function:

$$J = \int_0^\infty \left[x^T(t) Q x(t) + u^T(t) R u(t) \right] dt \qquad (69)$$

where x and u are the state and control vectors, Q and R are the state and control weighting matrices. Q and R are symmetric and $Q \geq 0$ and $R > 0$ [Ref. 15]. Bryson's method is known to be a good initial guess at formulating diagonal weighting matrices for LQR. Because MPC and LQR have similar cost functions, Bryson's method is applicable for MPC. The method requires the control designer to select the maximum error permissible in each state and control input. The weighting is then found as the square of the reciprocal of the max error [Ref. 1].

$$Q_{ii} = \left(\frac{1}{x_{i\max}} \right)^2 \qquad R_{ii} = \left(\frac{1}{u_{i\max}} \right)^2 \qquad (70)$$

Bryson's method is demonstrated for the MPC_ALL nonlinear simulation as shown in Tab. 6. The maximum error column is a design choice. There is no guarantee that the system will not violate the selected maximum error bounds as they simply give the initial weighting ratio between the states.

State	Units	Max Error	W_y Weighting
h	Ft	50	4.0×10^{-4}
V	Ft/s	15	0.0044
α	Deg	0.25	16
γ	Deg	0.25	16
Q	Rad/s	0.0175	3,265
X	Ft	200	2.5×10^{-5}
Y	Ft	100	1.0×10^{-4}
χ	Deg	0.5	4
μ	Deg	0.5	4
β	Deg	3	0.1111
P	Rad/s	0.0175	3,265
R	Rad/s	0.0175	3,265
Control	**Units**	**Max Error**	**W Weighting**
δ_e	Deg	1	1
δ_{sb}	Deg	3	0.1111
δ_a	Deg	5	0.04
δ_r	Deg	5	0.04
Δ Control	**Units**	**Max Error**	**W Weighting**
δ_e	Deg	0	0
δ_{sb}	Deg	1	1
δ_a	Deg	1	1
δ_r	Deg	1	1

Tab. 6 Bryson's Method State & Control Weightings

Weightings may be found for every flight point and then scheduled as discussed in section 3.3, however, this requires finding weighting matrices for every flight point and developing a smooth scheduling method. This may certainly be done, but the added complexity is not warranted for the preliminary weighting matrix design. Instead,

separate weighting matrices are defined using Bryson's method for each flight phase. The weights are blended naturally as discussed in 3.3 to reduce or avoid undesirable transients while changing flight phases.

While using Bryson's method may give favorable results and a stable system, it is not guaranteed to do so. It must be stressed that Bryson's method is merely a starting point for finding the weighting matrices, and not the final solution. In this application, the weights as stated thus far lead to instabilities during flight. It is now necessary to adjust the weightings through trial and error. The goal of the trial and error process is to find weights that lead to stable flight. Some trial and error may be used to achieve improved performance, but favorable performance is secondary to obtaining stable flight. The next stage in the weighting process addresses achieving the desired performance. The following guidelines are useful when weighting the matrices during the trial and error phase:

- The weightings between the states and controls are relative, so one weighting may be set to a given value such as unity, reducing the selection process by one value.

- Changing the weightings may lead to unpredictable results because of the states' and controls' relative nature. For example, changing the altitude weighting may adversely affect the weighting ratio between the angle of attack (α) and flight path angle (γ) states. One weighting change should be made at a time when tuning the system.

- Experience has shown that for this problem changing the W_Δ has little effect on the solution to the cost function. Select moderate values for W_Δ and concentrate on the remaining weighting matrices.

- Fast inner-loop dynamics such as α, and the rotational rates tend to require high weightings. States with slower dynamics such as altitude, downrange, and crossrange typically have low weightings.

- The weighting matrices become less important when hard and soft constraints are imposed, since the system will accept any cost to avoid violating hard constraints. Additionally, the system has a very high penalty for violating soft constraints, making the weighting matrices less influential when exceeding soft constraints [Ref. 9].

After some trial and error, new weights are found that lead to stable flight. The next step is to use these new weighting matrices for pole placement. A procedure is now

followed with a similar concept to that of the longitudinal example except the system now has 12 controlled states instead of 1. The longitudinal example called for a linearized model of a SISO closed loop system from the state input reference to the state output. A bandwidth representing the desired performance was then selected and a search program was used to modify the weighting value until the desired bandwidth was achieved.

However, in the MPC_ALL architecture, all of the states are controlled in a MIMO system. For the purposes of obtaining the weighting matrices, a closed loop transfer function may be obtained from the input reference of a single state to the output of that state as shown in Fig. 47 for α.

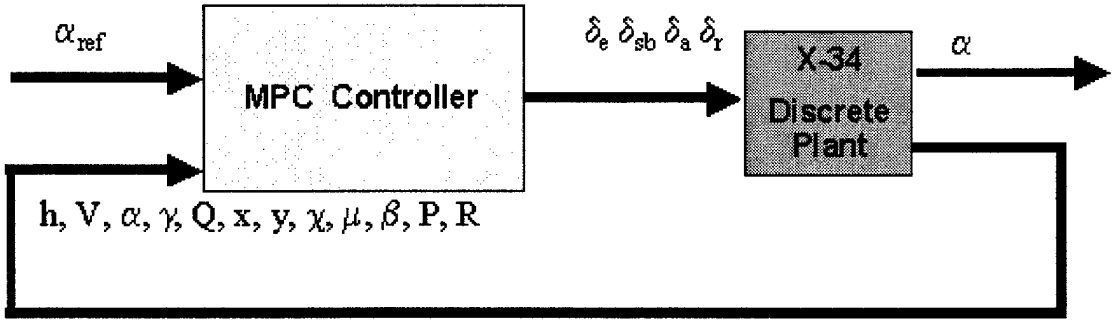

Fig. 47 MPC_ALL Linearized Architecture

Instead of selecting a bandwidth, the desired closed loop pole location is selected to represent the desired tracking performance. A search routine may be used to vary the weighting matrices until the pole location is achieved, or the weights may be found without a search routine by plotting the closed loop poles for various weights as demonstrated in this section. The weights are then selected corresponding to the desired pole locations.

The following example places the poles for the short period mode in the longitudinal channel for the full MPC_ALL nonlinear simulation flying the "straight" trajectory from section 3.4. This trajectory has no lateral maneuvers allowing for a careful weighting of the longitudinal states. The α and Q state weights are modified to place the phugoid poles. In this research, a close tracking of α is desirable, so the poles are placed such that a close tracking is achieved.

The pole placement routine only gives insight to the selected states and the associated poles, so a thorough weighting selection requires the procedure to be applied carefully to subsequent states. Since the state and control weightings are relative, this process should only be used on sets of states with little to no coupling between them. For example, a longitudinal state set of α and Q and a lateral state set of μ and P are mostly decoupled sets and are good candidates for this procedure. Even when the procedure is used on decoupled sets of states, the designer should be mindful of the relative nature of the weightings.

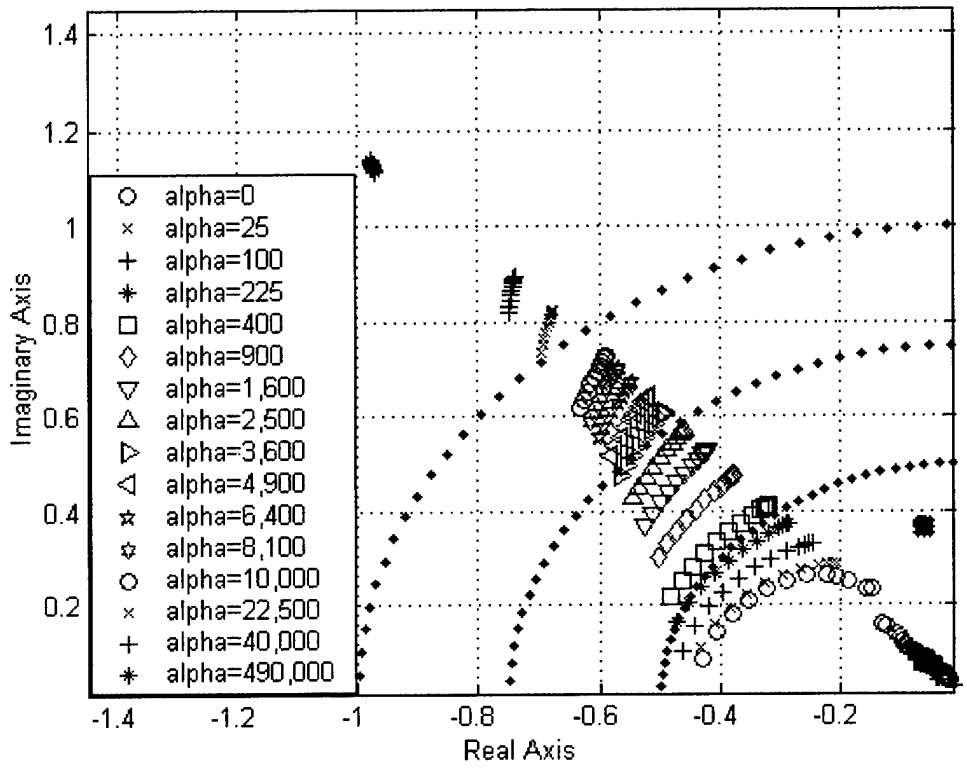

Fig. 48 Q and α Pole Placement

$$Q = \begin{bmatrix} 0 & 900 & 10{,}000 & 22{,}500 & 40{,}000 & 90{,}000 & 160{,}000 & \dots \\ & 250{,}000 & 360{,}000 & 490{,}000 & 640{,}000 & \dots & & \\ & 810{,}000 & 1{,}000{,}000 & & & & & \end{bmatrix}$$

$$\alpha = \begin{bmatrix} 0 & 25 & 100 & 225 & 400 & 900 & 1{,}600 & 2{,}500 & 3{,}600 & \dots \\ & 4{,}900 & 6{,}400 & 8{,}100 & 10{,}000 & \dots & & & & \\ & 22{,}500 & 40{,}000 & 490{,}000 & & & & & & \end{bmatrix} \qquad (71)$$

Fig. 48 shows the positive conjugate of the closed loop short period pole for a range of Q and α weightings. As the α weighting increases, the pole location extends radially from the origin. As the Q weighting increases, the pole moves diagonally downward creating a region of possible pole locations. The desired pole location for a close tracking of α is roughly in the middle of the region at –0.465 + 0.565i. This location corresponds to an α weighting of 2,500 and a Q weighting of 40,000. These values are significant departures from the α = 16 and Q = 3,265 weightings obtained from Bryson's method. It is important to validate the new weightings to insure proper selection.

The closed loop system with the new weightings is linearized discretely. A Bode plot of the closed loop model in Fig. 49 shows the effect of a nonminimum phase zero at the lower frequencies reducing the db magnitude. Between the 0.08 and 0.4 rad/s frequencies the magnitude levels at approximately 0 db corresponding to a desirable tracking performance. The system bandwidth is 0.525 rad/s at –3 db.

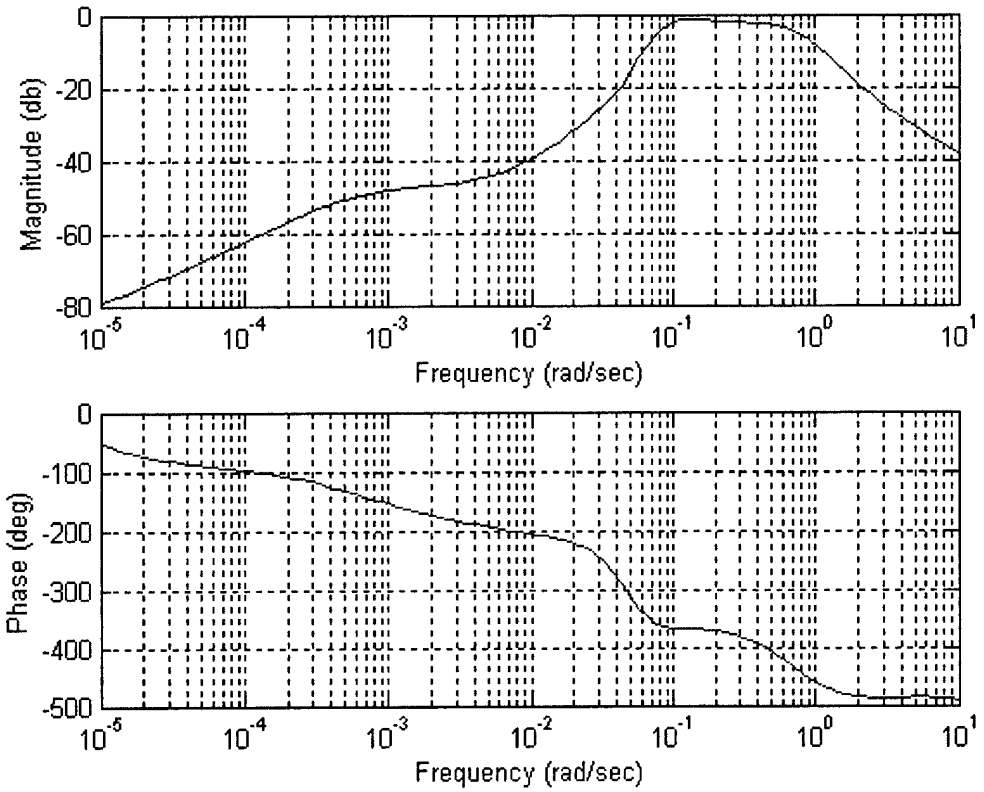

Fig. 49 Bode Plot for Q = 40,000 and α = 2,500

The Bode plot suggests a good tracking performance on the α state. It is necessary to validate that the simulation has adequate tracking with the new weightings. Because these weightings are higher than those obtained from the Bryson's method and trial and error, the MPC is expected to track the α closer. The following series of plots shows the improved tracking performance with increasing α and Q weightings for a naturally commanded step change in the α command near the flight point 140. Tab. 7 shows the weights for the figures.

Figure	α wt	Q wt	Method
Fig. 50	16	2,500	Bryson's & Trial and Error
Fig. 51	400	2,500	Intermediate Weighting
Fig. 52	2,500	40,000	Pole Placement

Tab. 7 α and Q Weightings

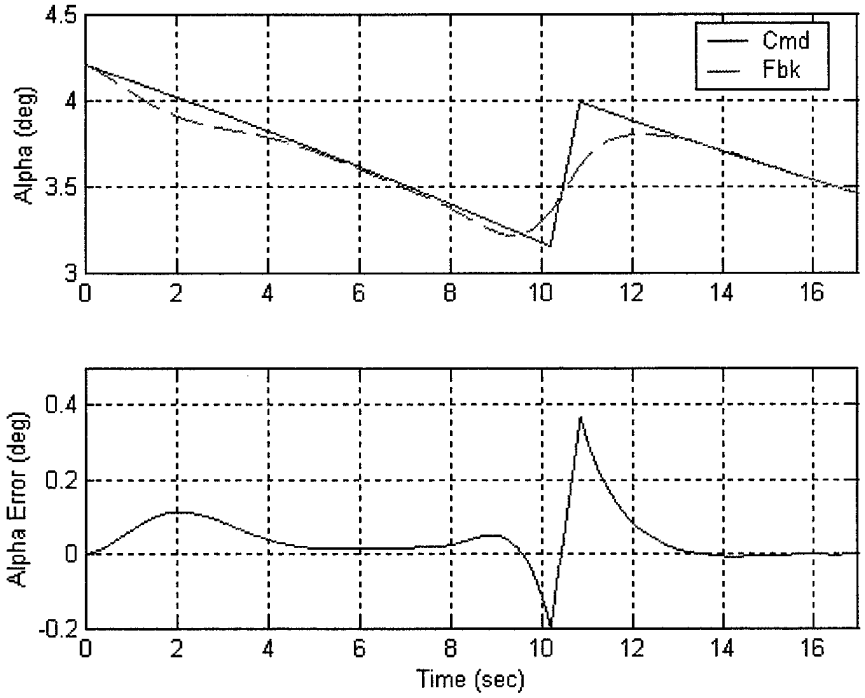

Fig. 50 α Tracking for α = 16 Q = 2,500

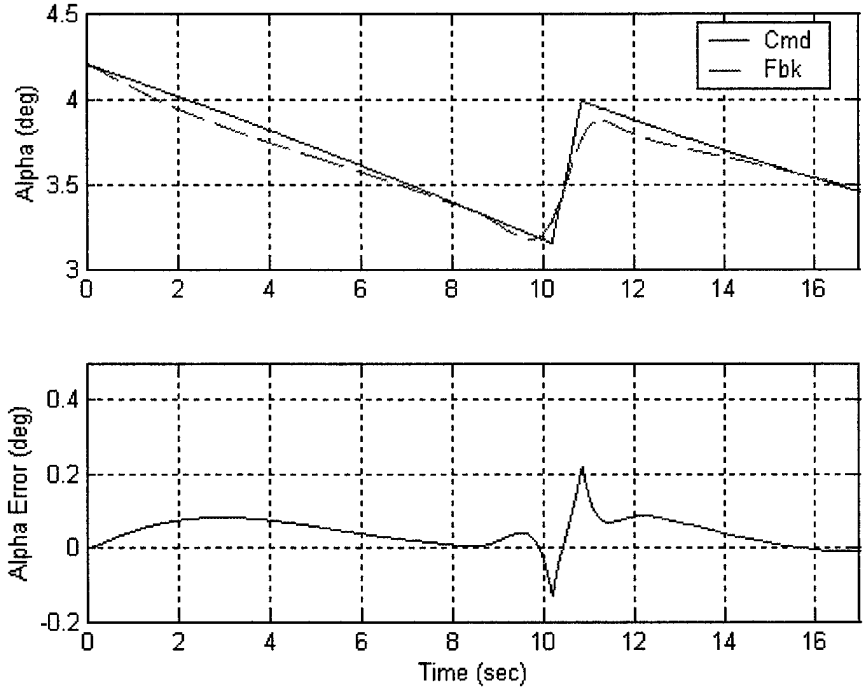

Fig. 51 α Tracking for α = 400 Q = 2,500

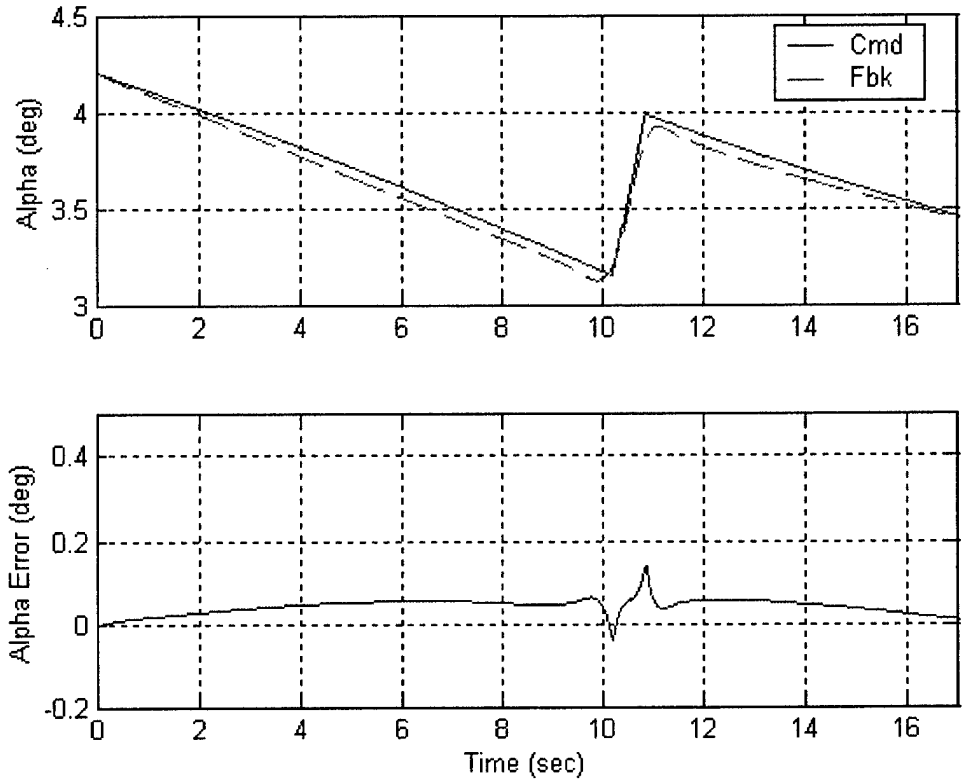

Fig. 52 α Tracking for α = 2,500 Q = 40,000

When a lower weighting is placed on α, MPC tries to reduce errors by acting well before receiving the command change. The increase in α leads to reduced errors and hence improved tracking. In each figure, the prediction horizon is 3 seconds. However, as the α weighting increases, the anticipative nature of MPC is not as pronounced. With increasing weighting, the bandwidth increases and a faster response is observed. Given the faster response, movement at first indication of the step change would lead to a greater error than delaying action until closer to receiving the command. Hence, less preemptive action is taken when MPC knows the change may be made quickly. In this fashion, the MPC may be thought of as a smart controller, by acting sooner to reduce peak errors in controlled states with little emphasis on tracking and by acting later for controlled states with great emphasis on tracking.

Pole placement weightings may be scheduled, should that be found desirable. The following describes the development of such a schedule. In section 4.5 it was found that the poles of the short period move significantly and have discontinuities throughout the

trajectory. The desired specific pole placement may be difficult to achieve for some points due to these discontinuities. Such discontinuities present challenges when trying to schedule the weightings.

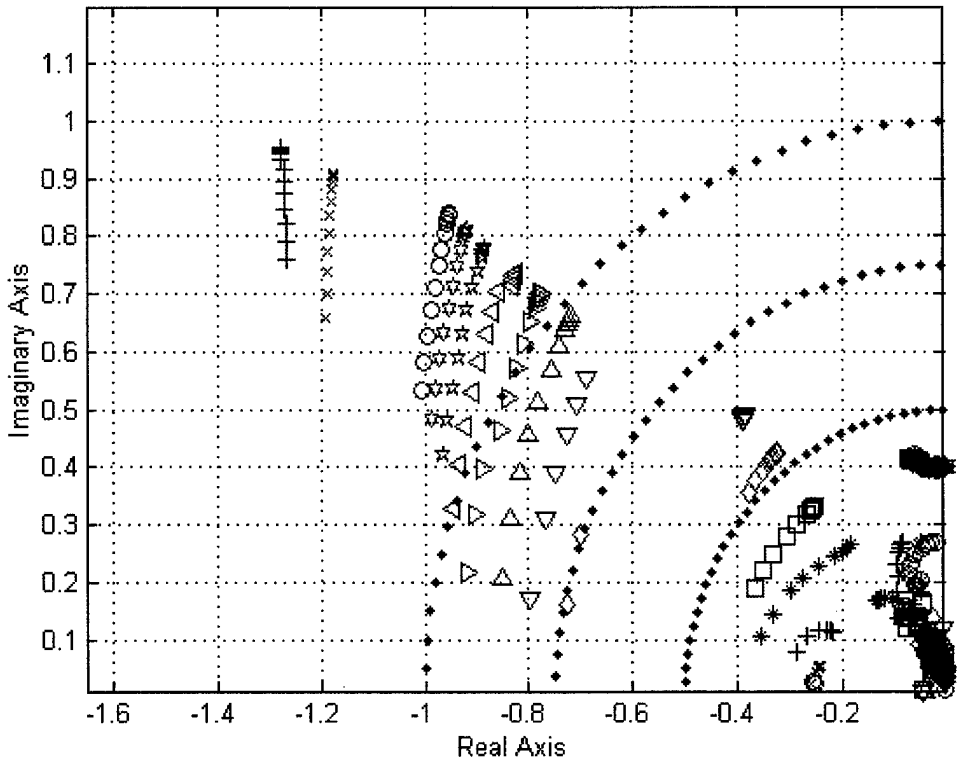

Fig. 53 Pole Location for Flight Point 76

Fig. 53 shows a discontinuity in the pole placement for flight point 76. The same set of α and Q weightings are plotted as in Fig. 48. In Fig. 48, the poles fan out fairly uniformly and the voids at the higher weighting values only exist because not enough α and Q weightings have been selected to fill the space. In contrast, Fig. 53 illustrates a noticeable void between –0.4 and –0.7 on the real axis as an area where poles may not be placed easily. As the weightings increase, the cost function finds a new minimum by changing the pole location quite abruptly. The cluster of up-side-down triangles near –0.4 + 0.5i jump significantly with higher Q weightings. Similarly, a dislocation is observed from the same cluster to the right-side-up triangles with only a small increase in α. The desired pole location of –0.465 + 0.565i may not be met for this flight point during the pole location discontinuity. As a result, a new pole location is selected at

–0.75 + 0.6i. This new pole location is a significant and unavoidable departure from –0.465 + 0.565i, but maintains continuity in the α and Q weights.

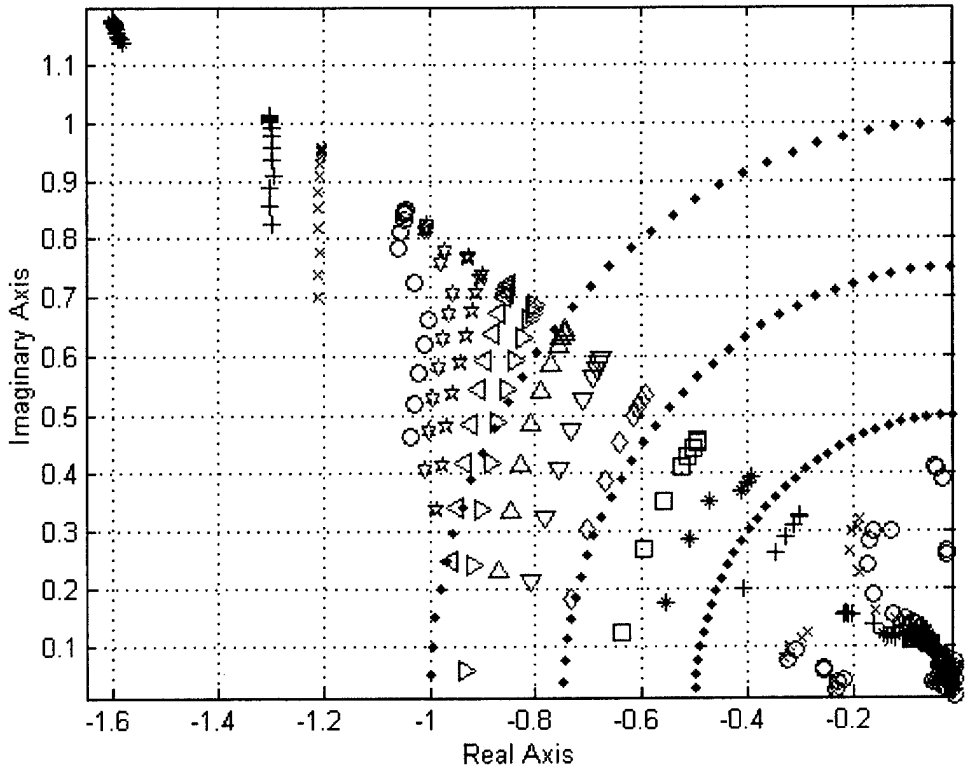

Fig. 54 Pole Location for Flight Point 70

A few seconds further down the trajectory at flight point 70, continuity is restored to the region mapped by the α and Q weights as seen in Fig. 54. The discontinuity in flight has led to a new region of possible pole locations. Because the new area is still in the left hand plane, the discontinuity disrupts a smooth scheduling scheme for a constant pole location and pushes the poles farther left, but does not cause instabilities. Selecting a different desired pole location when a discontinuity arises allows for a smoother weighting schedule to be developed.

Fig. 55 shows the α and Q weightings for each flight point in the straight trajectory found through the pole placement method. The schedule has been artificially smoothed at the discontinuity for flight point 76 by selecting the new pole location. A weighting schedule such as this may be used when preferred over weighting by flight segments.

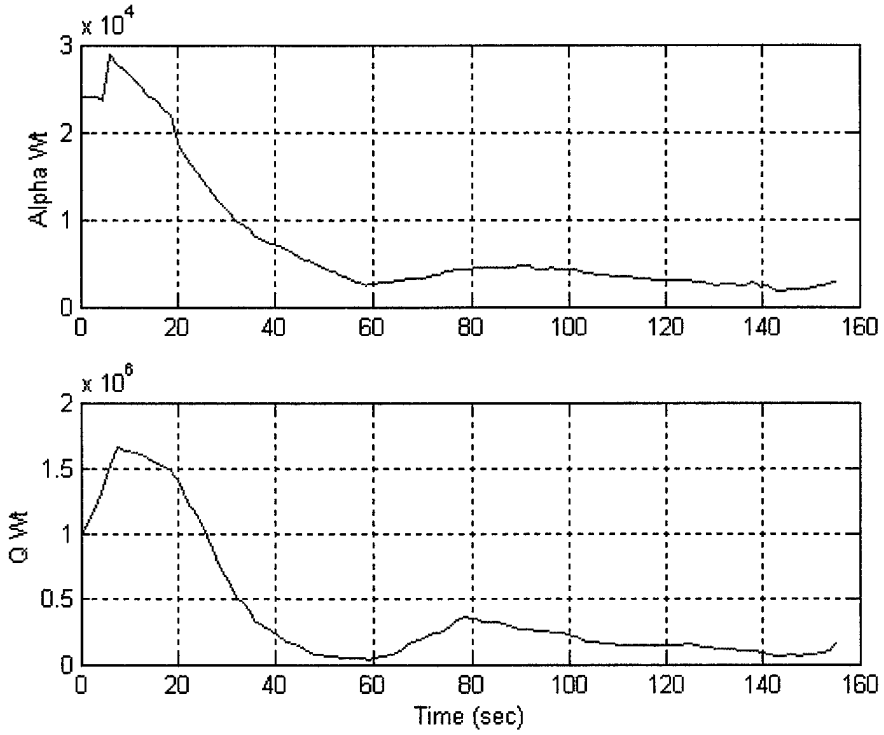

Fig. 55 α and Q Weighting Schedule for Pole Placement

Once the pole placement is accomplished, the final phase of trial and error is conducted. The trial and error is necessary to insure no relative weightings were unexpectedly altered. If the pole placement is too aggressive, the actuators may reach saturation. A period of trial and error with small changes in the state and control weightings helps to eliminate saturation. In addition, trial and error allows for some increase in performance for states not addressed through the pole placement procedure. Time constraints prohibited selecting the weightings for the lateral channel through a pole placement procedure. Instead, they were selected using the trial and error guidelines. During the final trial and error phase, the weightings on α and Q had to be increased slightly in response to the weighting of the lateral channel. This slight increase was necessary to maintain a high level of α tracking and to maintain the approximate short period pole location.

Weighting by scheduling has been discussed for the benefit of future designs. A flight phase weighting scheme is adopted for this specific problem. The pole placement strategy, however, is still used when finding the α and Q weightings for each phase.

The final weightings after trial and error are summarized in Tab. 8 for MPC_ALL and in Tab. 9 and Tab. 10 for MPC_SAS.

State	Phase 1	Phase 2	Phase 3	Phase 4	Phase 5
h	0.25	0.36	0.64	1.21	1.21
V	0.01	0.04	0.09	0.16	0.25
α	3,600	4,900	9,025	8,100	6,400
γ	1	1	1	1	1
Q	67,600	42,025	84,100	57,600	38,025
x	1	1	1	1	1
y	0.25	0.49	0.01	0.25	0.25
χ	9	64	49	36	36
μ	1	9	9	4	4
β	400	400	400	400	400
P	2,500	900	62,500	1,600	900
R	2,500	900	62,500	1,600	900
Control	**Phase 1**	**Phase 2**	**Phase 3**	**Phase 4**	**Phase 5**
δ_e	64	49	49	36	25
δ_{sb}	64	25	25	25	25
δ_a	64	1	1	9	9
δ_r	64	1	1	9	9
Δ Control	**Phase 1**	**Phase 2**	**Phase 3**	**Phase 4**	**Phase 5**
δ_e	1	1	1	1	1
δ_{sb}	1	1	1	1	1
δ_a	1	1	1	1	1
δ_r	1	1	1	1	1

Tab. 8 Final Weighting Matrices for MPC_ALL

State	Phase 1	Phase 2	Phase 3	Phase 4	Phase 5
LQR h	0.001	0.004	0.04	1	1
LQR V	0.1	0.5	1	1	1.5
LQR α	25	25	25	15	10
LQR γ	10	10	10	4	4
LQR Q	32.8281	32.8281	32.8281	32.8281	32.8281
LQR x	0.001	0.001	0.001	0.001	0.001
LQR y	0.001	0.01	0.01	0.01	0.01
LQR χ	0.1	1	5	2	2
LQR μ	1	0.1	10	10	10
LQR β	6	10	30	20	12
LQR P	32.8281	32.8281	32.8281	32.8281	32.8281
LQR R	32.8281	32.8281	32.8281	65.6561	65.6561
Control	**Phase 1**	**Phase 2**	**Phase 3**	**Phase 4**	**Phase 5**
δ_e	0.1	0.25	0.5	0.25	0.25
δ_{sb}	5	0.1	0.1	0.03	0.03
δ_a	0.5	0.7	2	3.77	3.77
δ_r	0.8	0.5	0.7	0.77	0.77

Tab. 9 Final LQR Weighting Matrices for MPC_SAS

Altitude, velocity, and crossrange weights are given for the MPC_SAS architecture because the LQR develops a set of gains for all 12 states using the complete plant, not just the portion of the plant representing the 9 inner loop states. However, only the gains for the inner loop states are used as the MPC actively controls the altitude, velocity, and crossrange in the outer loop.

State	Phase 1	Phase 2	Phase 3	Phase 4	Phase 5
MPC h	0.0025	0.01	0.0225	0.0625	0.16
MPC V	0.81	1	1.21	1.44	1.69
MPC y	0.81	1	1.1025	1.44	1.96
MPC Cmd	**Phase 1**	**Phase 2**	**Phase 3**	**Phase 4**	**Phase 5**
MPC γ	81	100	100	81	64
MPC δ_{sb}	2.56	4	4	3.24	2.89
MPC χ	400	400	400	400	400
Δ MPC Cmd	**Phase 1**	**Phase 2**	**Phase 3**	**Phase 4**	**Phase 5**
MPC γ	1	1	1	1	1
MPC δ_{sb}	1	1	1	1	1
MPC χ	1	1	1	1	1

Tab. 10 Final MPC Weighting Matrices for MPC_SAS

5.4 Base Function Selection

The procedure used to obtain the proper number of base functions discussed in the longitudinal example may be applied to the full simulation without change. The horizon is 3 seconds and the prediction rate used is 10 Hz giving 30 possible base functions. Fig. 56 shows the cost associated with a 1-degree error between the α reference and current α state value for flight point 140. The costs from using only one or two base functions have been omitted from the plots because their costs were too high to be placed on the graph.

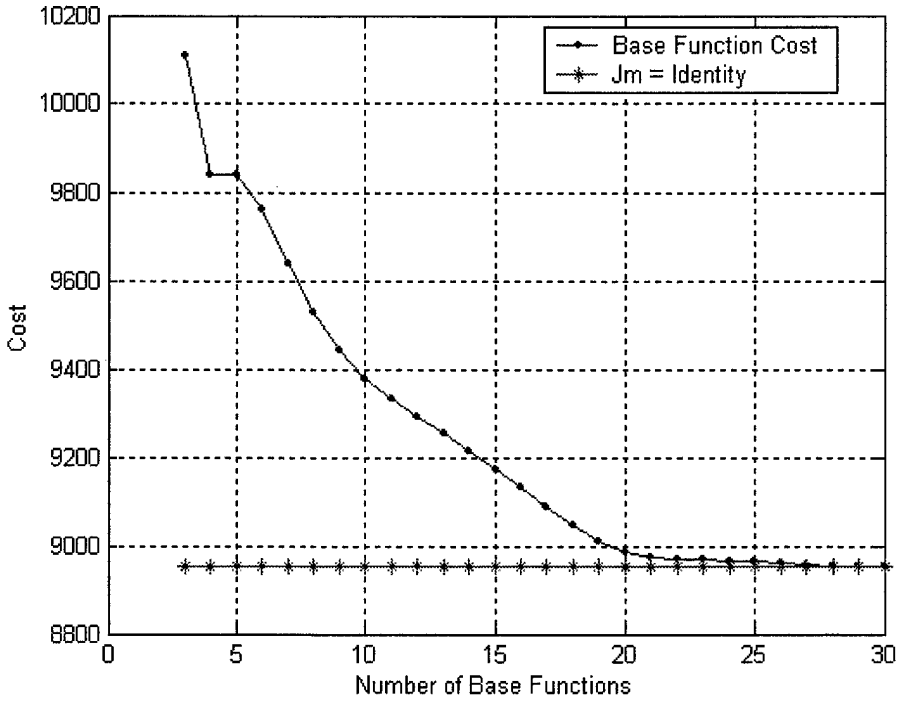

Fig. 56 Base Function Cost for 1-Degree α Reference Error

The cost is an effective parameter to select the proper number of base functions. Another parameter that may be used is the initial control position. Because an error is introduced to the system, the MPC moves the aerosurfaces from their nominal locations. For a system with a relatively high weighting on α arrived at through the pole placement in the previous section, a significant deviation from trim is expected in the elevon. The nominal trim position for the elevon is –6.70 degrees. The same procedure is followed for evaluating the control as was used for the cost. Fig. 57 shows the elevon position for all 30 base functions denoted by the dots, and the position when not using base functions is shown as the starred line. The elevon position varies considerably for fewer than 10 base functions before converging.

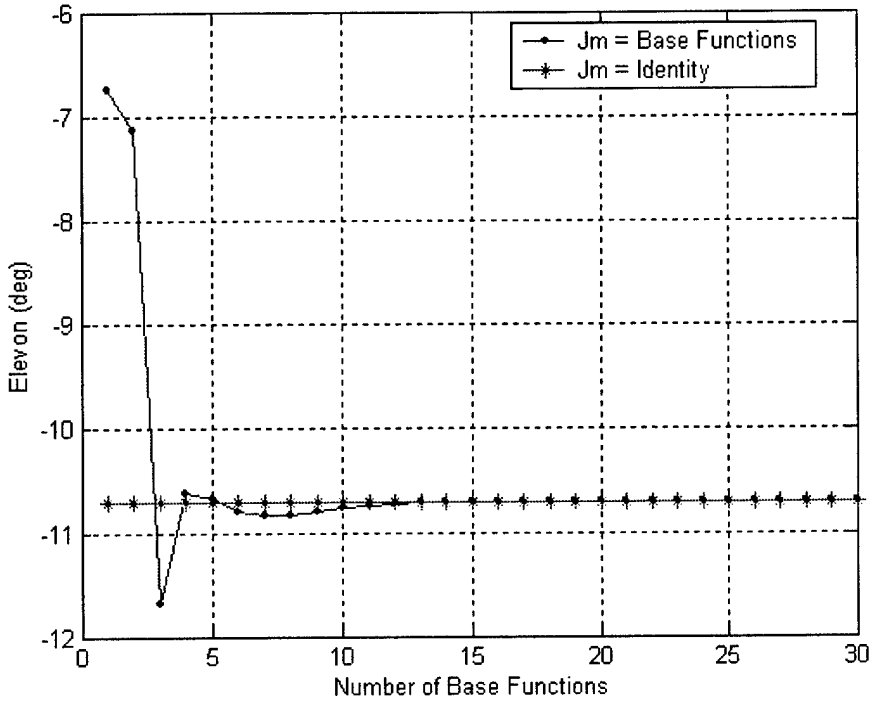

Fig. 57 Elevon Position for 1-Degree α Reference Error

The elevon is the aerosurface with control over eliminating the error in α, however, some mild coupling exists in the dynamics and the remaining control surfaces move slightly from their nominal values. For flight point 140, 0 degrees is the nominal position for the speed brake, aileron, and rudder. Convergence is obtained for the three surfaces when 10 base functions are applied.

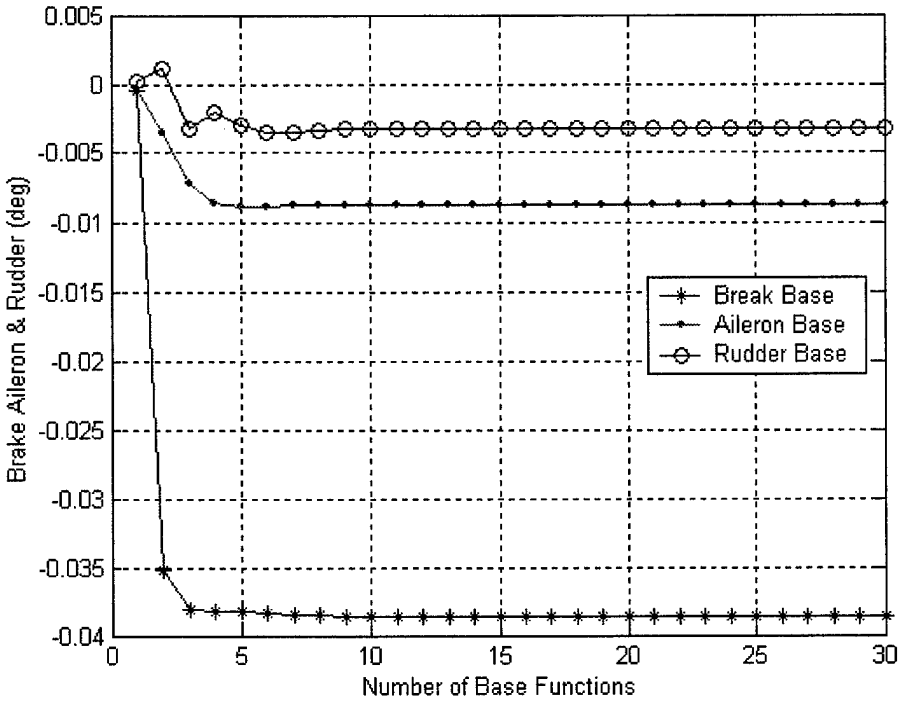

Fig. 58 Brake Aileron & Rudder Position for 1-Degree α Reference Error

While the cost is the parameter to be minimized, examining the control is a very accurate tool for selecting the number of base functions to use. If the control input for a selected number of base functions is nearly the same as for setting the J_M matrix to the identity, then the output will almost be the same. Ideally, the cost and the control should be evaluated when selecting the number of base functions to use. Tab. 11 shows the percent error in the cost and control incurred by using base functions. Equation 66 was used to calculate the error.

Number of Base Functions	% Cost Error	% Control Error
3	12.888	9.119
5	9.926	0.306
10	4.787	0.539
15	2.506	0.014
20	0.383	0.002
25	0.131	7.69×10^{-5}
30	0	0

Tab. 11 Base Function Errors for Cost & Control

The cost with 10 base functions is reasonable but the reduction to less than 3 percent error with 15 base functions is desirable. Furthermore, oscillation is observed in the control error between 5 and 10 base functions, thereby narrowing the selection to 10-15 functions. By 15 base functions the control has converged and significantly reduced in error. Based on the cost and control error, 15 base functions are selected. The base functions selected are the ramp functions described in section 4.4.

With the base functions selected, it is necessary to evaluate the output performance of the simulation. Fig. 59 shows the command and output response of using 15 base functions and using an identity for J_M. The same α step and weightings are used as in Fig. 52. The matrix weightings are those obtained right after the pole placement and before the final trial and error phase. The output using base functions is indistinguishable from the output not using base functions. The error shown is the difference in output between using and not using base functions.

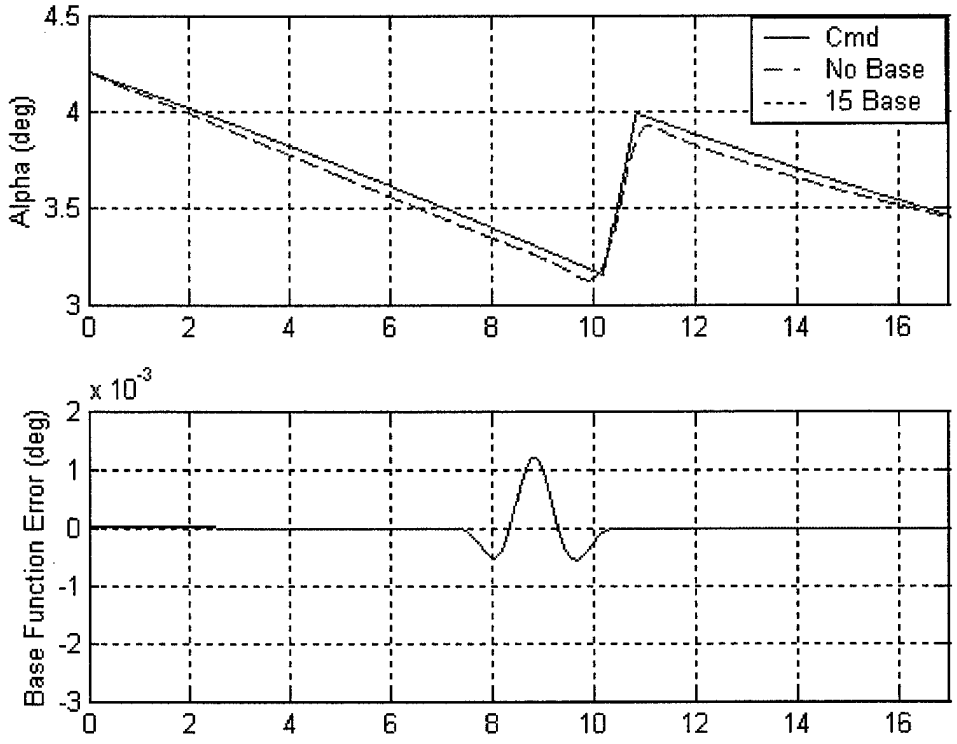

Fig. 59 α Tracking 15 Base Functions (Unconstrained)

Fig. 59 shows the α response for the unconstrained case. The same plot is shown in Fig. 60 but for the constrained solution. The constraints were imposed on the altitude, but set sufficiently far away that the output would never come in contact with them. The constraints were only applied to trigger the flag in the simulation to use the optimizer for the constrained solution. The constrained simulation also shows an indistinguishable output when using 15 base functions. In either the unconstrained or constrained simulations, the error is not greater than 0.003 degrees, a negligible difference.

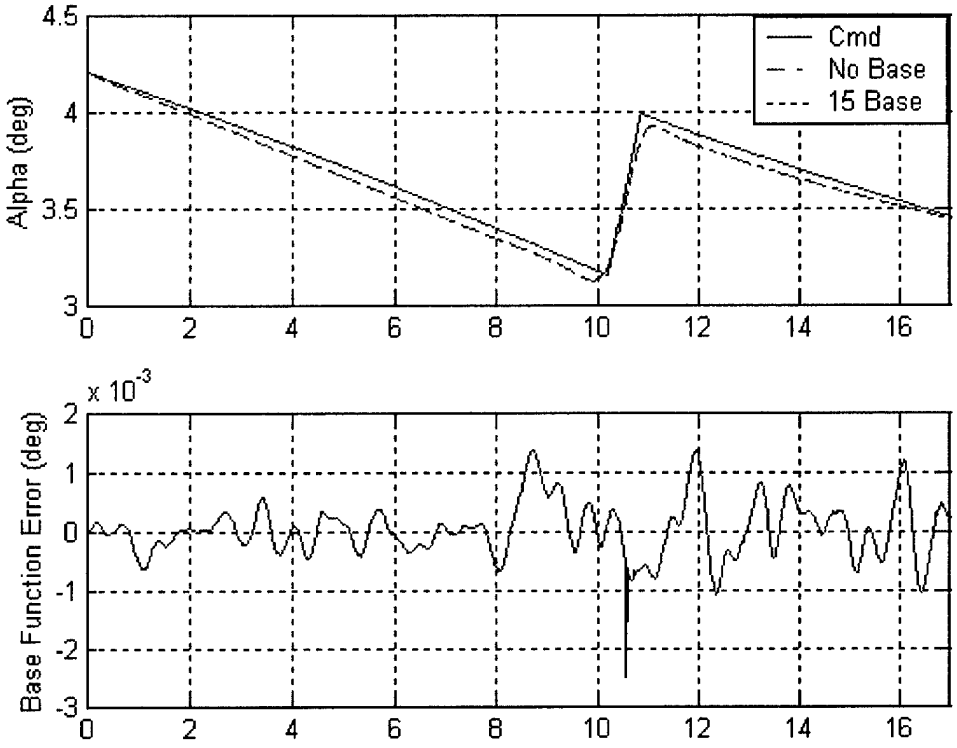

Fig. 60 α Tracking 15 Base Functions (Constrained)

In section 4.4 it was claimed that base functions are used to reduce the simulation computational time and that base functions are most beneficial when used on the constrained simulation. Tab. 12 shows the actual time required to simulate 20 seconds of flight in Fig. 59 and Fig. 60. The two figures are zoomed in on the first 17 seconds to show greater detail around the step. Base functions are effective in reducing the computation time without a significant loss in performance.

	Time No Base Fun (seconds)	Time 15 Base Fun (seconds)	% Reduced Time
Unconstrained	197.92	142.03	28
Constrained	266.42	173.73	35

Tab. 12 Computational Time Using Base Functions

Chapter 6

Results & Performance

This chapter shows the system performance of the MPC_ALL and MPC_SAS architectures. The two trajectories flown are the "straight" and "single bank" trajectories described in 3.4. The response to flying three aggressive trajectories is presented in the appendix. The first aggressive trajectory represents a situation where the vehicle is initially far from the runway and low on energy. The second portrays the vehicle close to the runway and high on energy initially. The final trajectory presents a case when the vehicle initially has a high crossrange value far away from the trajectory and is low on energy.

Each trajectory flies from its starting point at about 30,000 feet to approximately 2,000 feet above ground (the runway altitude is at 3,840.5 feet). The landing procedure begins when the vehicle is at about 2,000 feet above ground. During the landing, the vehicle's landing gear is deployed and the flight dynamics of the vehicle change significantly. The MPC has not been designed to account for these changing dynamics, so this portion is not shown in the results.

The selected design parameters from the previous sections are summarized in Tab. 13, and the weighting matrices are described in Tab. 8, Tab. 9, and Tab. 10 at the end of section 5.3.

	MPC_ALL	MPC_SAS
P Horizon	3 sec	3 sec
Prediction Rate	10 Hz	10 Hz
Simulation Rate	50 Hz	50 Hz
Outer Loop Rate	10 Hz	5 Hz
Inner Loop Rate	N/A	50 Hz
Base Functions	15	15

Tab. 13 Design Parameter Summary

For all figures showing results in this chapter the solid line represents the reference and the dashed line represents the vehicle's response unless otherwise stated.

6.1 Command Versus Actual Plots

The MPC_ALL architecture is presented first, flying the "straight" trajectory, followed by the "single bank" trajectory. The MPC_SAS is then shown flying the "straight" and "single bank" trajectories. Each trajectory is plotted in a set of five figures.

The results from flying the MPC_ALL "straight" trajectory are presented in Fig. 61 to Fig. 65. The only commanded maneuvers lie within the longitudinal channel. Close tracking is observed in all states with the controller eliminating most errors encountered. α is tracked particularly closely as designed in the matrix weighting section. Two step changes in the α reference at 35 seconds and 60 seconds are arrived through moderate movements in the elevon. Only minute movements are seen in the aileron and rudder caused by mild coupling between the longitudinal and lateral states. The lateral dynamics, however, are not actively exercised, leaving the lateral states following closely with no significant deviations from the nominal conditions.

Because the vehicle starts aligned with the runway, there is only one change in weighting matrices at about 120 seconds into the flight. A smooth transition between

weighting matrices takes place as the vehicle begins the approach and landing flight phase.

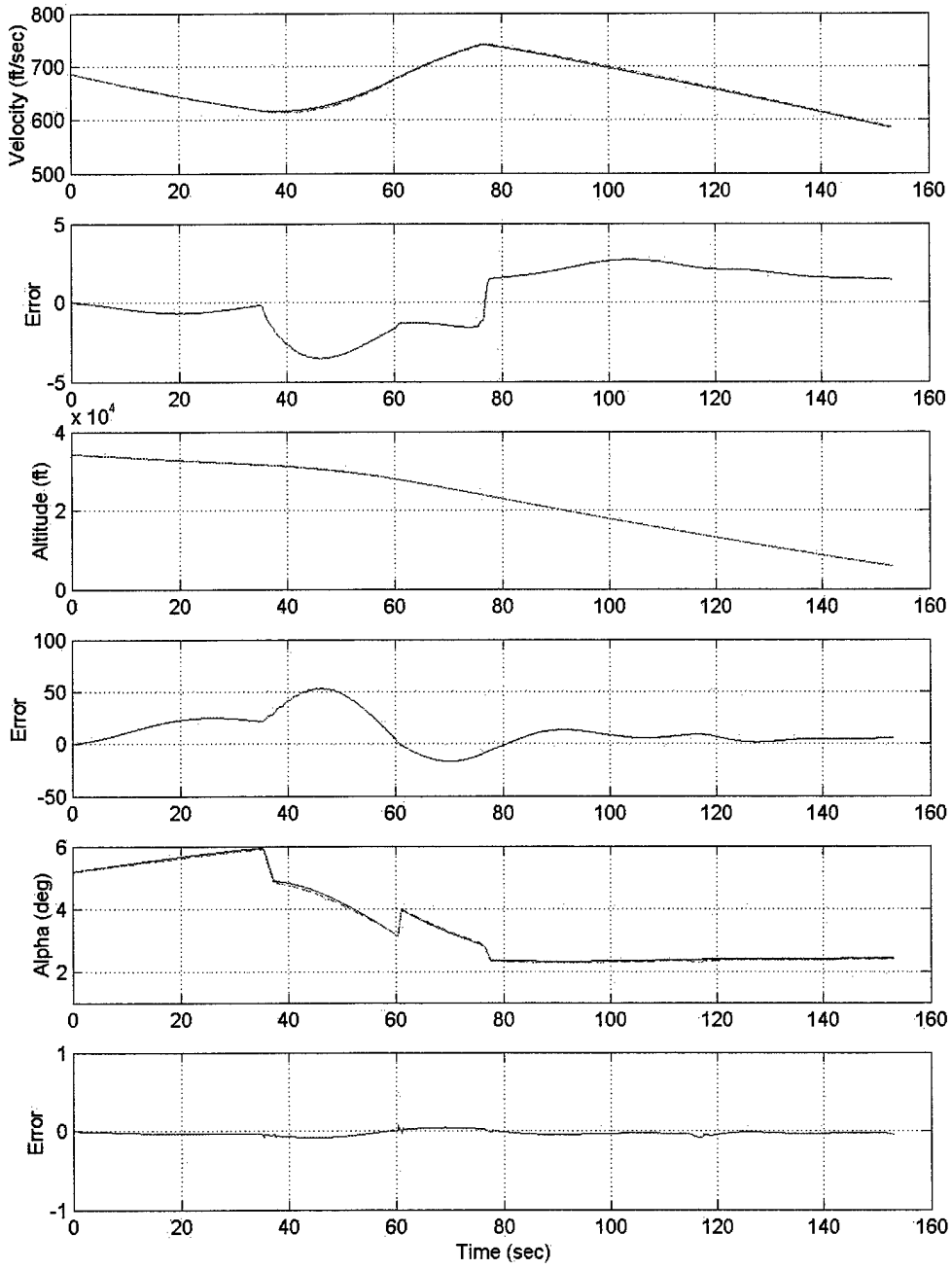

Fig. 61 MPC_ALL Straight (1 of 5)

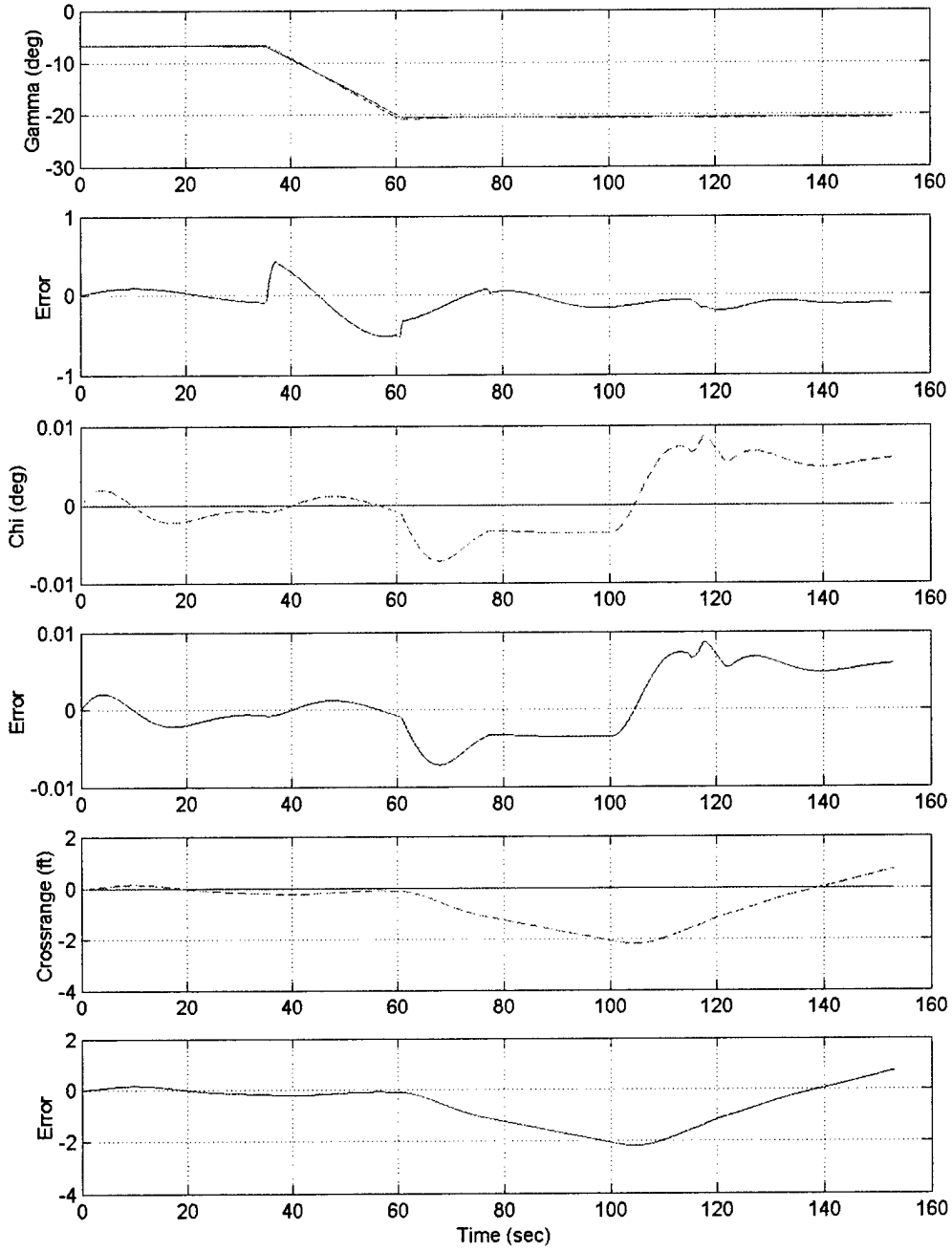

Fig. 62 MPC_ALL Straight (2 of 5)

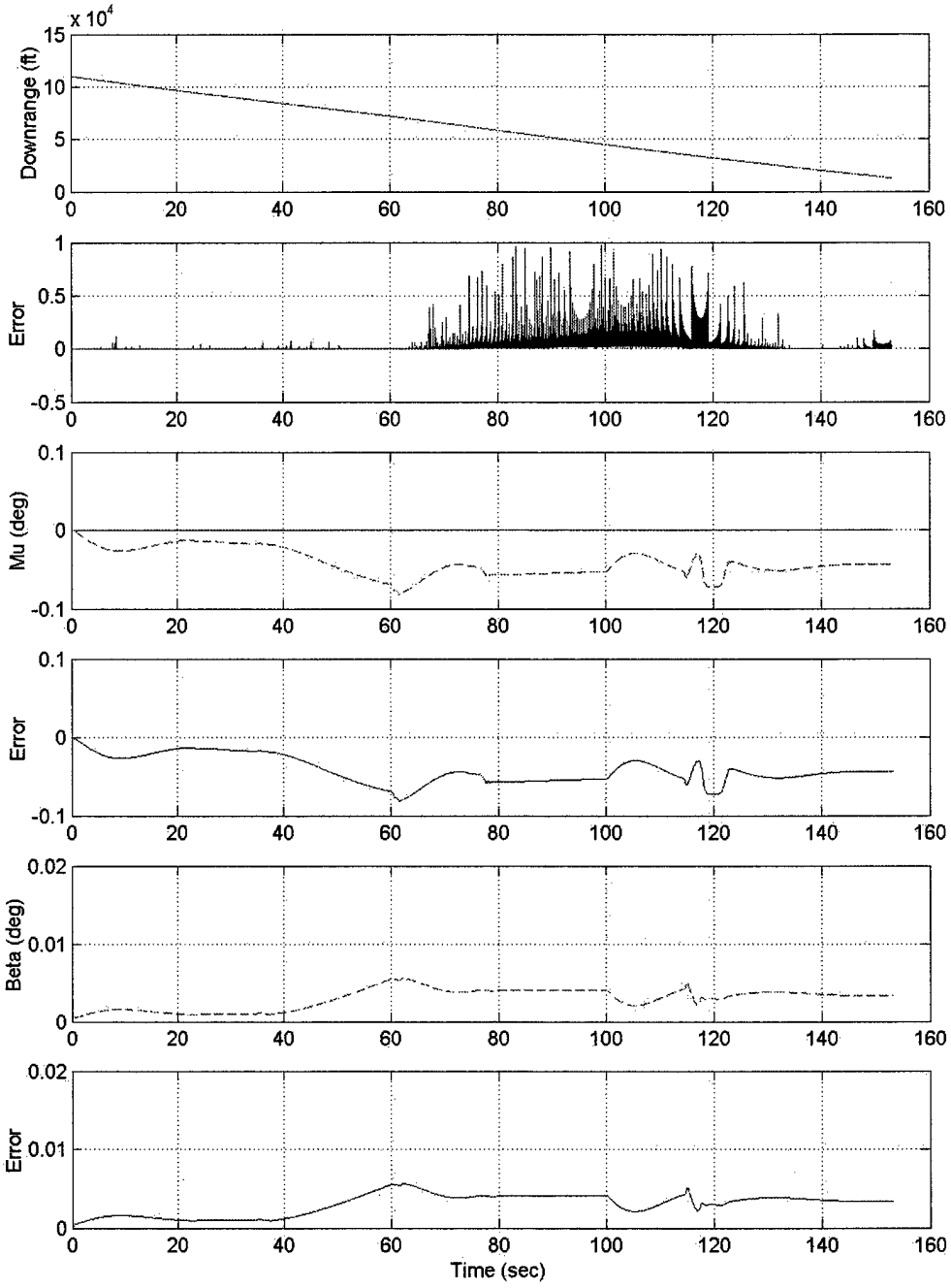

Fig. 63 MPC_ALL Straight (3 of 5)

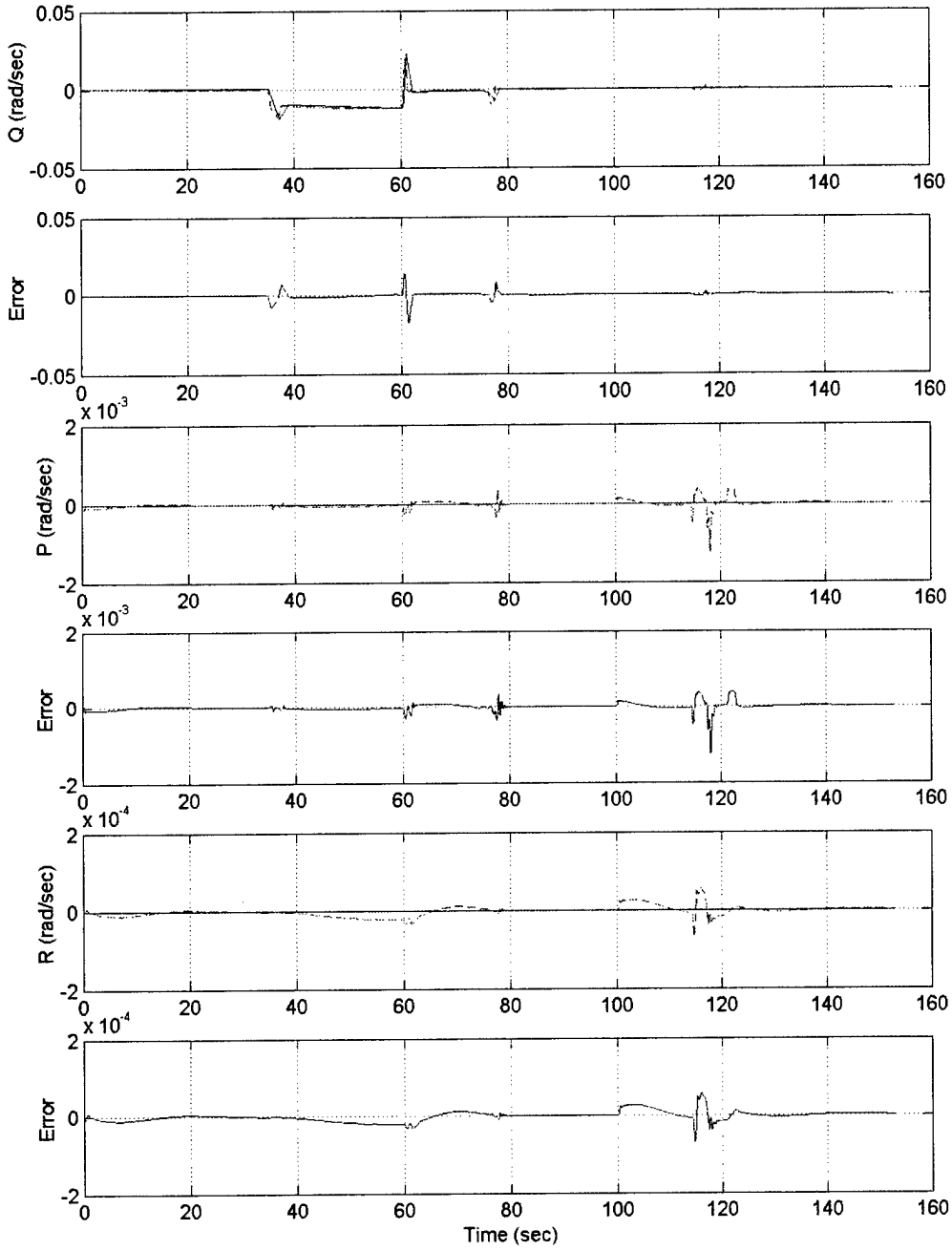

Fig. 64 MPC_ALL Straight (4 of 5)

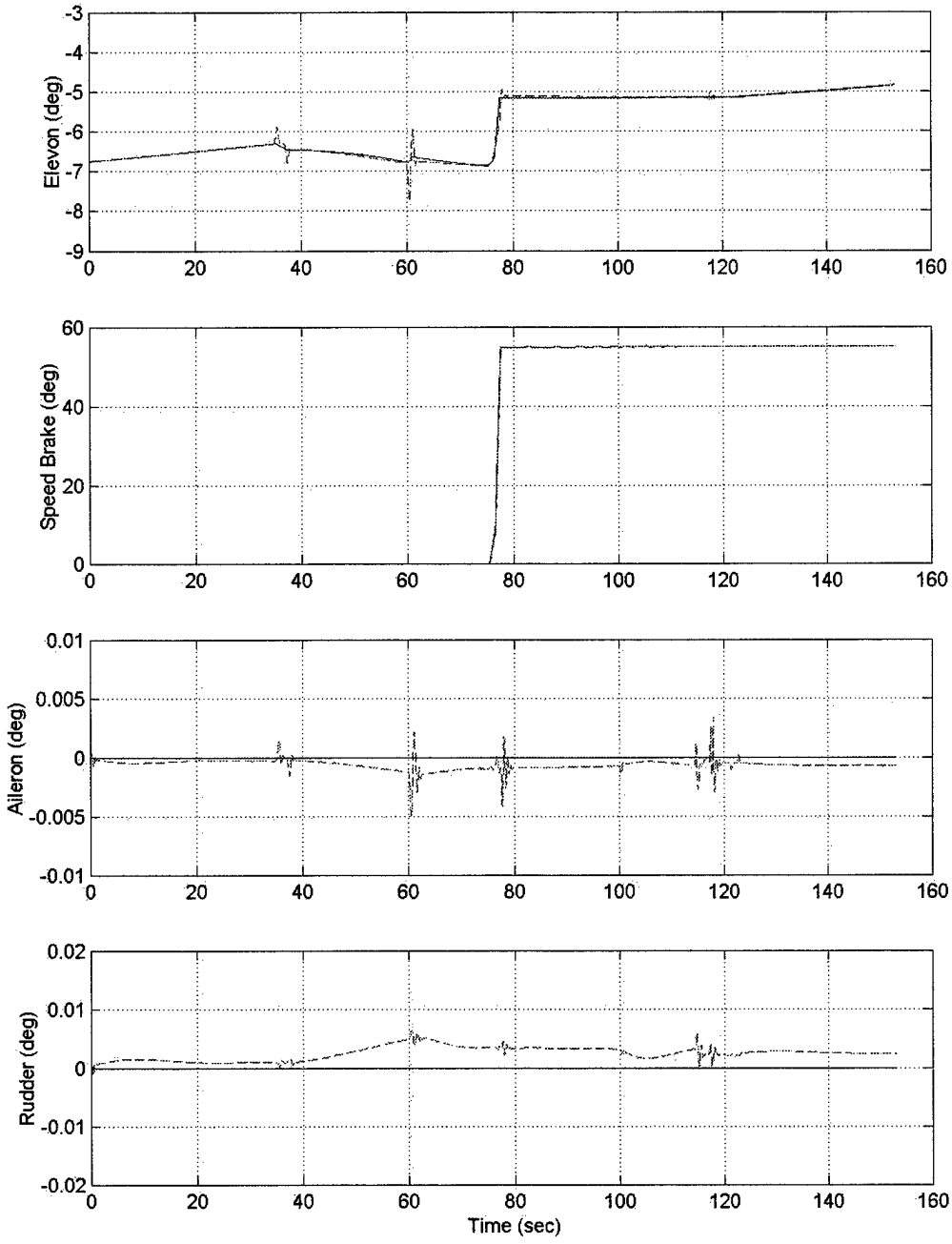

Fig. 65 MPC_ALL Straight (5 of 5)

Close tracking of the longitudinal states is maintained in the "single bank" trajectory. The effective coupling between the longitudinal and the added lateral dynamics is seen in α as the small increase in the command starting at about 85 seconds and ending near 100 seconds. This increase in the command corrects for the vehicle's natural tendency to pitch down during the bank. A small error is seen in α immediately following the bank as it arrives at the final value commanded for landing.

While the longitudinal channel is tracked closely, there is damped oscillation in the lateral channel following the banking maneuver. The overshoot in the bank angle and in the heading causes the vehicle to cross the runway centerline 4 times before becoming aligned. The loose tracking may suggest additional design is required for the lateral state weightings. The low loop rate is a contributor to the observed oscillation. A higher loop rate is required to accurately track all of the vehicle states closely. This higher rate, however, was unobtainable in this research due to the increased computational time required. Such a high computational time is unfeasible. Finally, a small portion of the error in the lateral channel may be caused by conflicts between the lateral states. A conflict in the states during a banking maneuver is difficult to visualize. Consider the following simplified example. If the vehicle exits the heading alignment cone with a 0 degree heading angle and a 0 degree bank angle but offset in the crossrange direction, it will be flying on a path parallel with the runway centerline. In order to eliminate the crossrange error, the vehicle must change its heading, and a heading change requires a banking maneuver. Because the vehicle is following the 0 degree commanded heading and bank angles, the vehicle must introduce an error in those two states in order to eliminate the crossrange error. Applying a relatively high weighting on the crossrange state helps to make the vehicle remove the crossrange error, but too high of a weighting can lead to too much banking overshoot during the banking portion of flight. A well-coordinated set of commands between the heading angle, bank angle, and crossrange states is necessary to help reduce conflicts in the states.

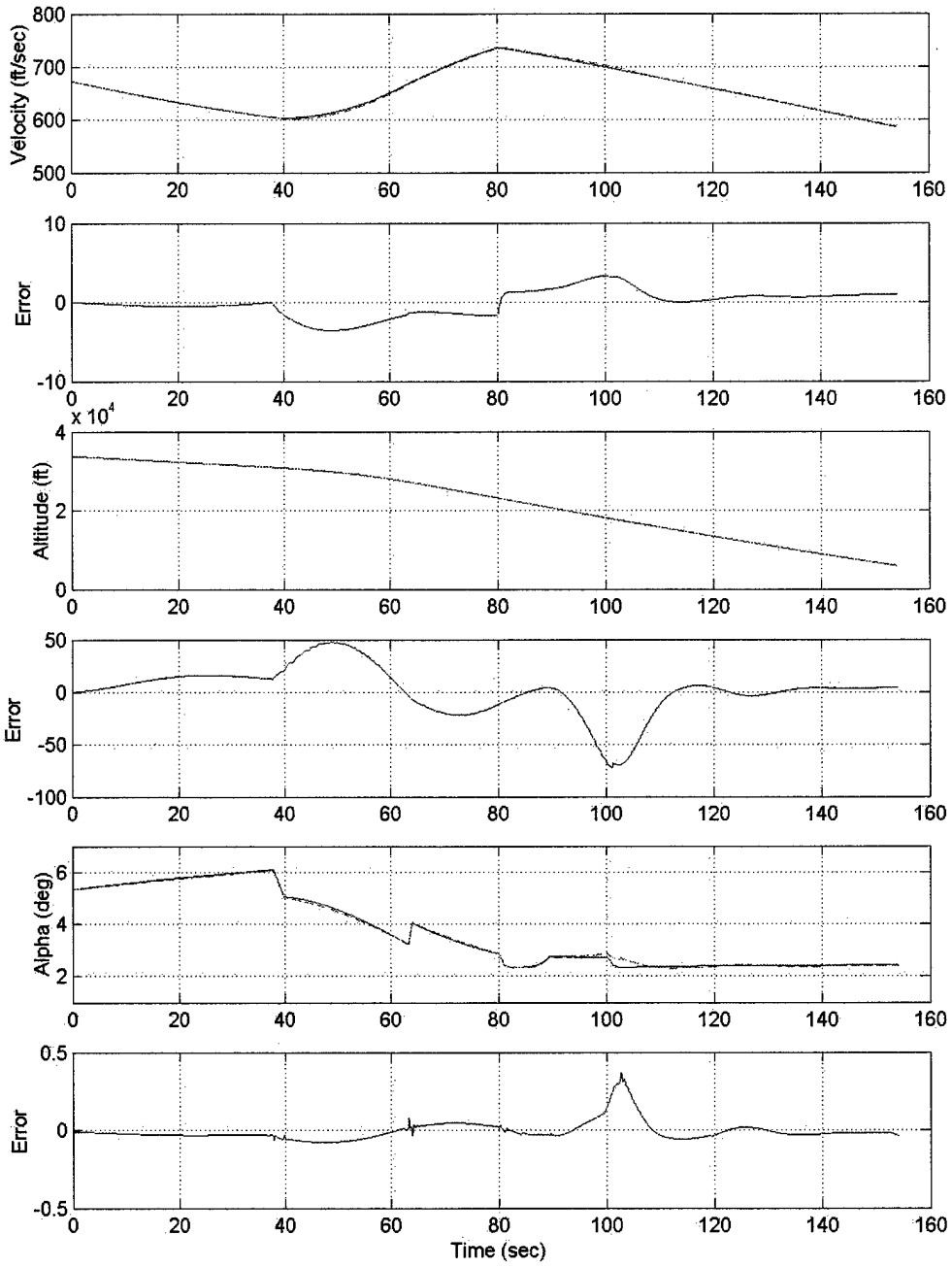

Fig. 66 MPC_ALL Single Bank (1 of 5)

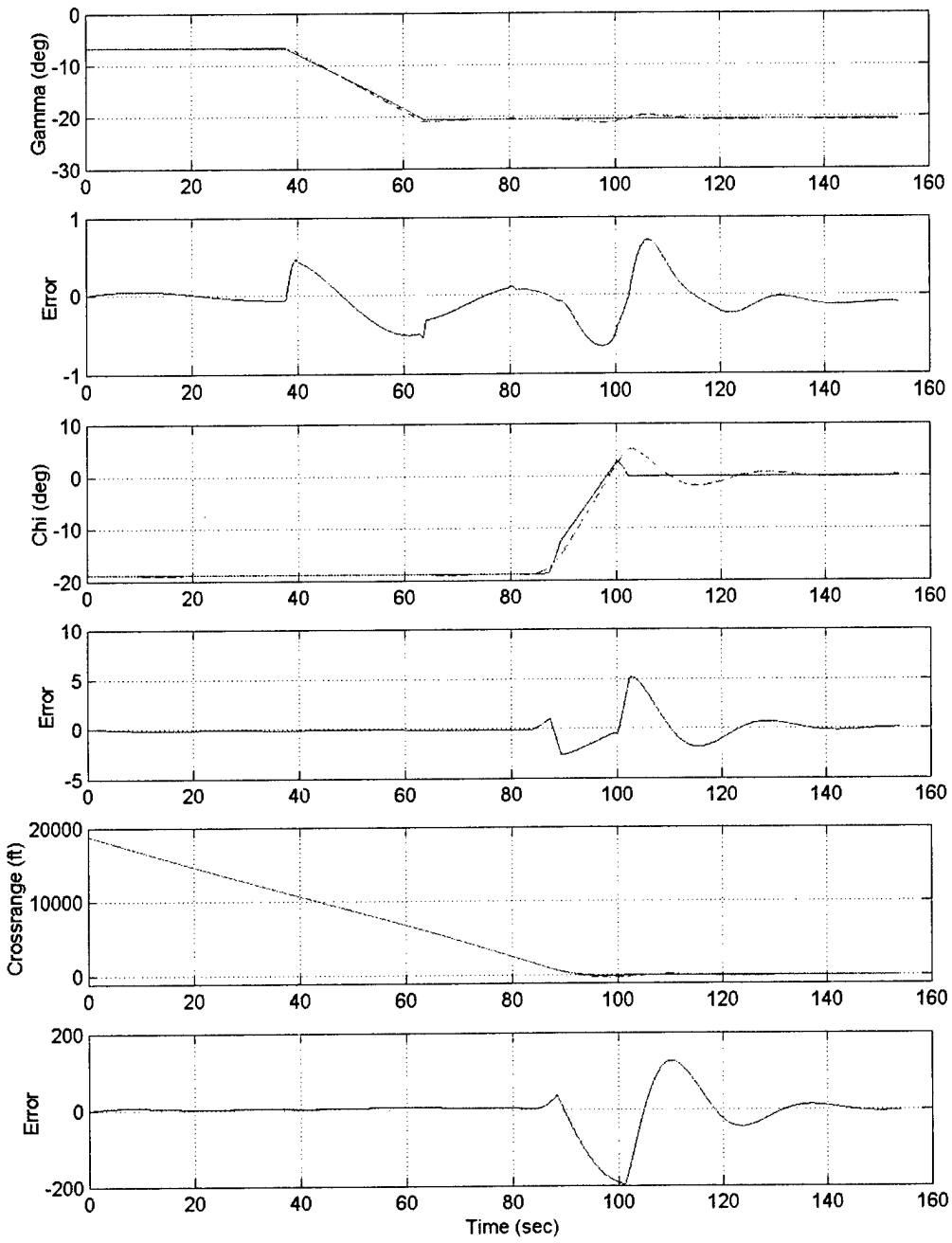

Fig. 67 MPC_ALL Single Bank (2 of 5)

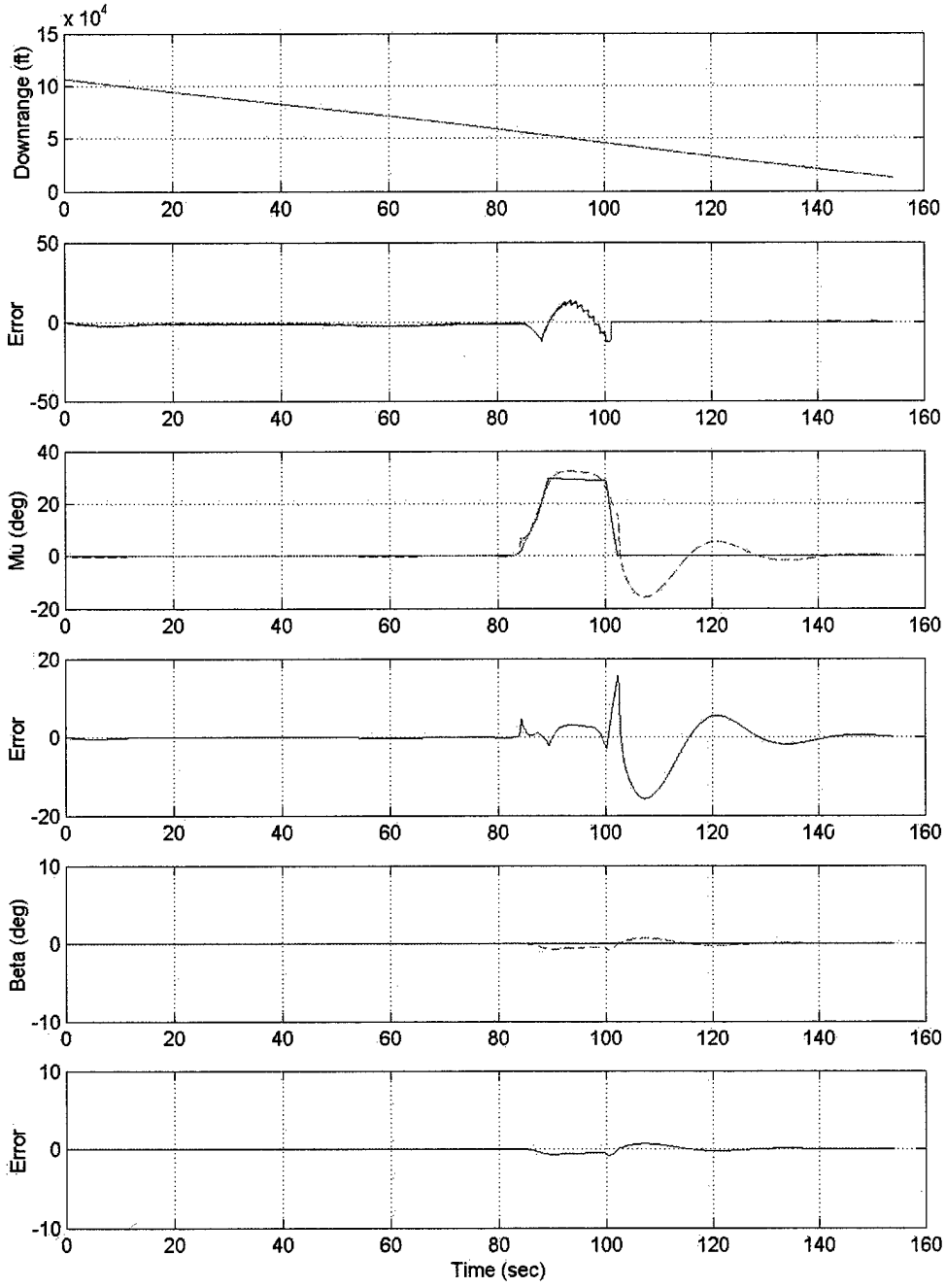

Fig. 68 MPC_ALL Single Bank (3 of 5)

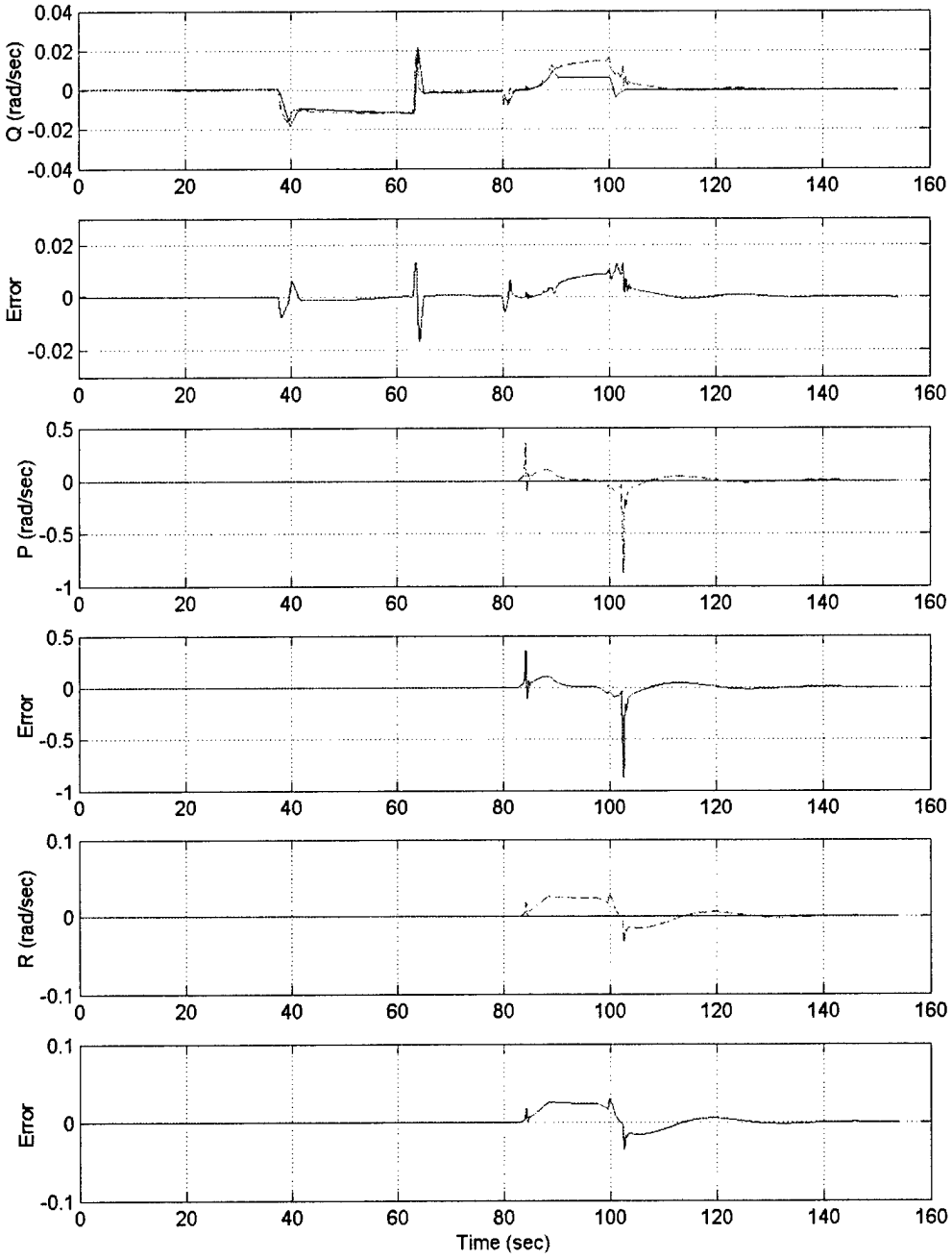

Fig. 69 MPC_ALL Single Bank (4 of 5)

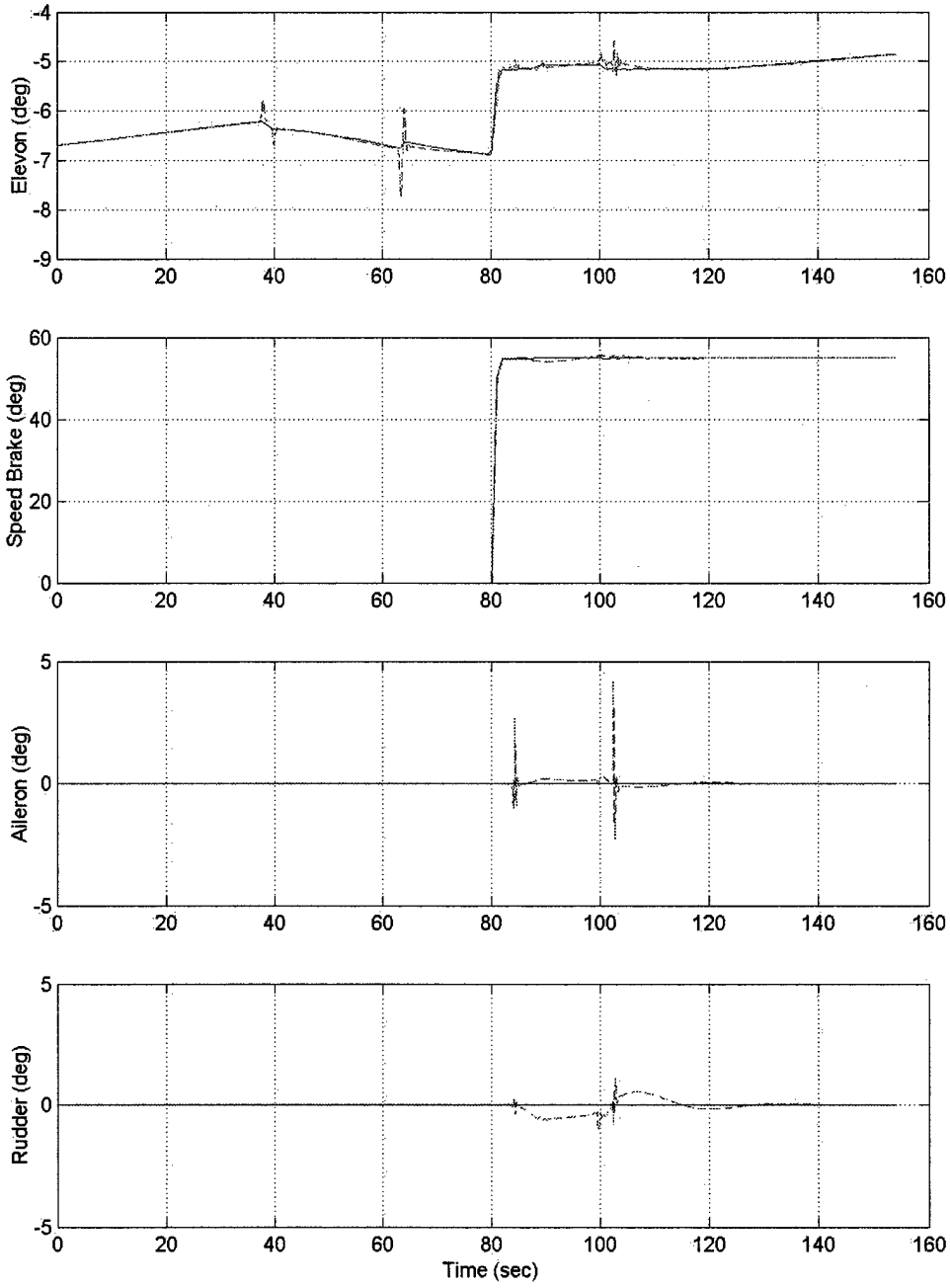

Fig. 70 MPC_ALL Single Bank (5 of 5)

The performance seen in the "straight" trajectory using the MPC_SAS architecture is very similar to that of the MPC_ALL architecture. They both adequately remove errors in the flight. The MPC_SAS has the higher inner loop rate allowing it to closely regulate the lateral inner loop states. Specifically, the heading angle and the crossrange errors are held very close to 0. The change in weighting matrices at about 120 seconds is clearly visible in the elevon causing a small disturbance in the α and γ states. MPC_SAS does not handle the transition particularly well in part because the natural smoothing technique is not as gradual for the MPC_SAS as it is for the MPC_ALL. This stems from the lower outer loop rate of the MPC_SAS. In MPC_ALL, the prediction rate and loop rate are both 10 Hz making the new weighting matrix enter the prediction horizon with one 0.1 time step at a time. The MPC_SAS, however, has an outer loop rate of 5 Hz and a prediction rate of 10 Hz. The prediction then sees the new weighting matrix change in two 0.1 time step blocks each time the MPC is called. The slightly choppier introduction of the new weighting matrices is only one cause of the disturbance in the elevon position as the change in MPC weightings between the flight phases is small. The integration between the inner and outer loops adds to the disturbances at flight phase change points. Additional smoothing is needed to create a more accurate model, however, the advantages and disadvantages of MPC_SAS may still be evaluated.

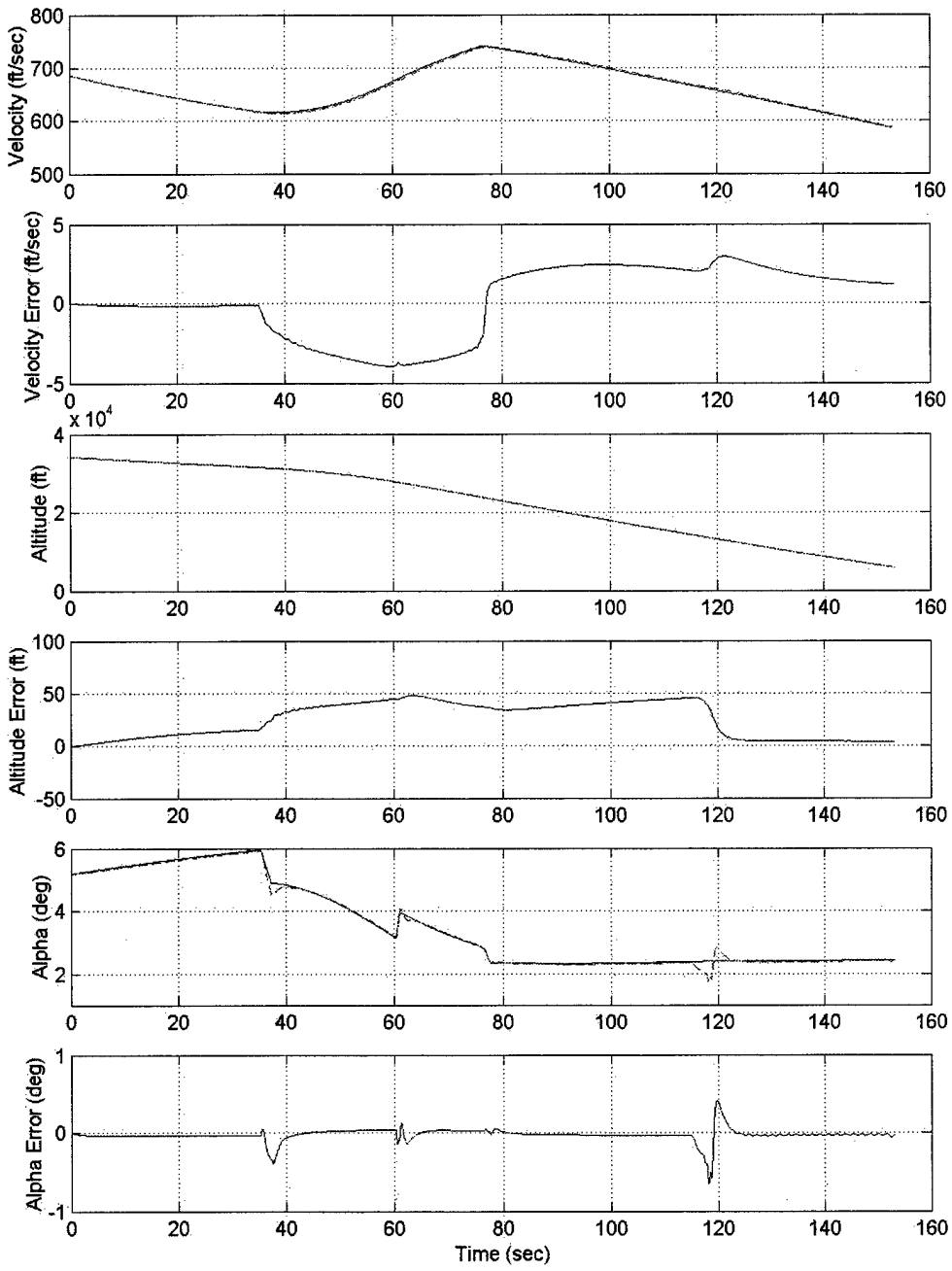

Fig. 71 MPC_SAS Straight (1 of 5)

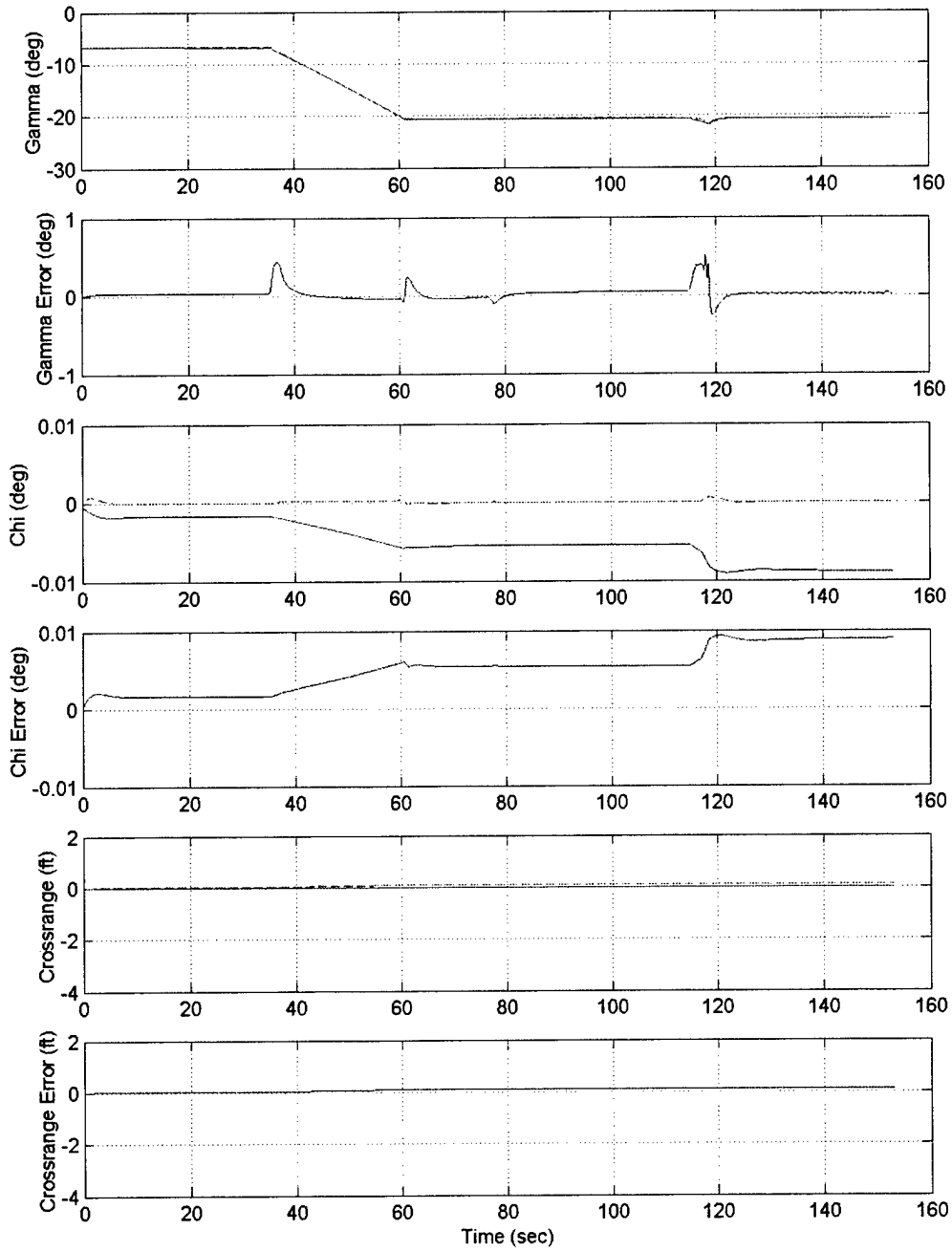

Fig. 72 MPC_SAS Straight (2 of 5)

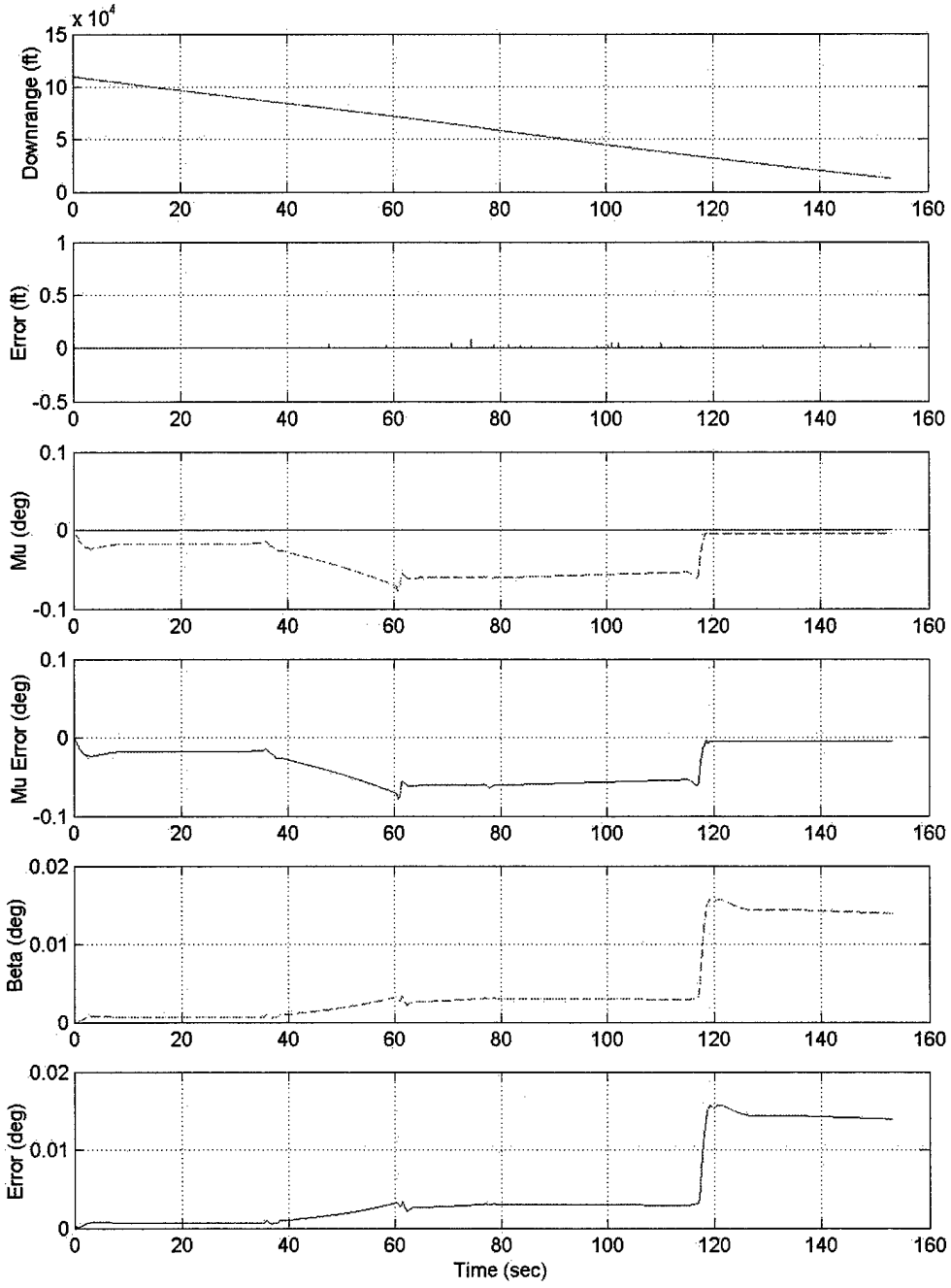

Fig. 73 MPC_SAS Straight (3 of 5)

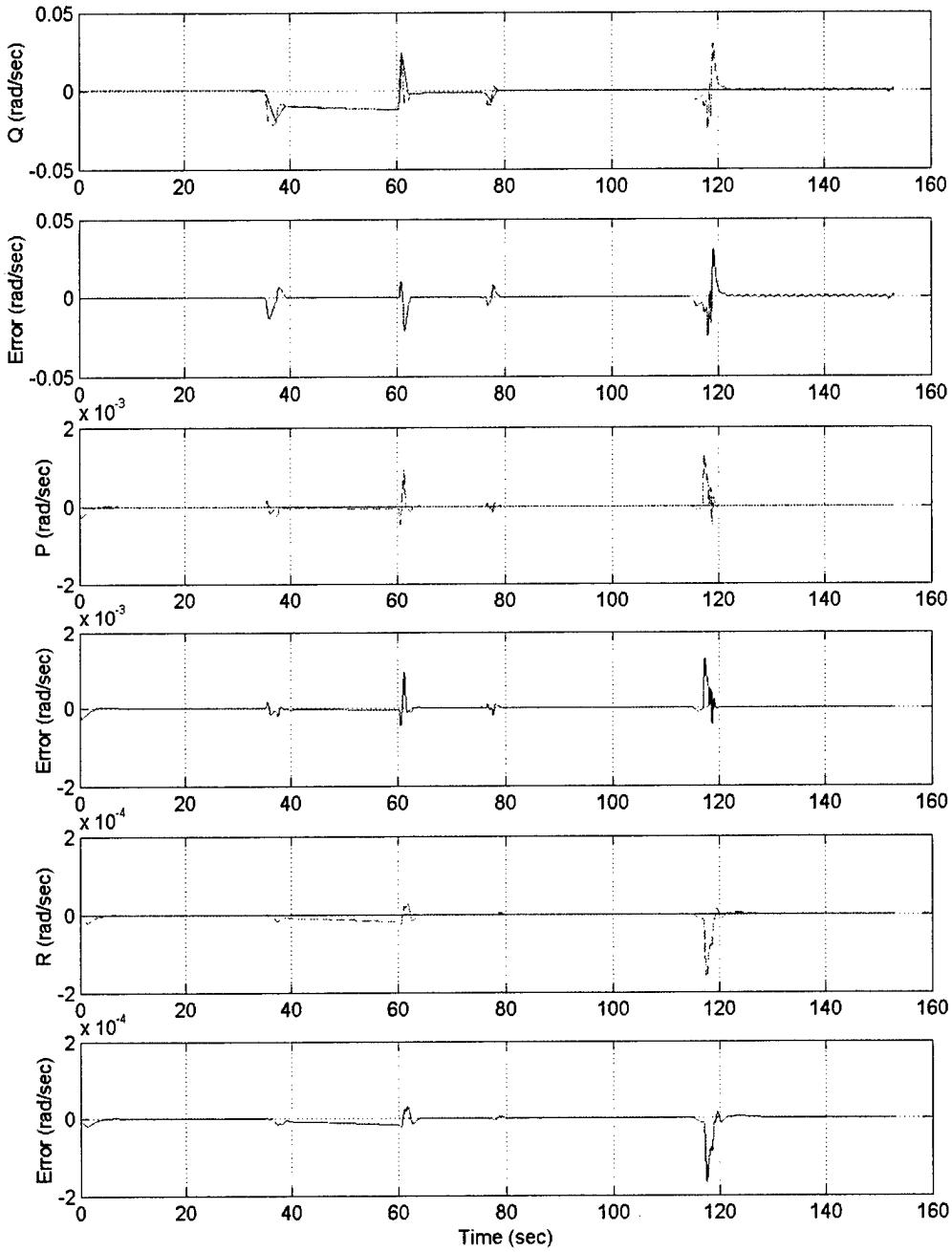

Fig. 74 MPC_SAS Straight (4 of 5)

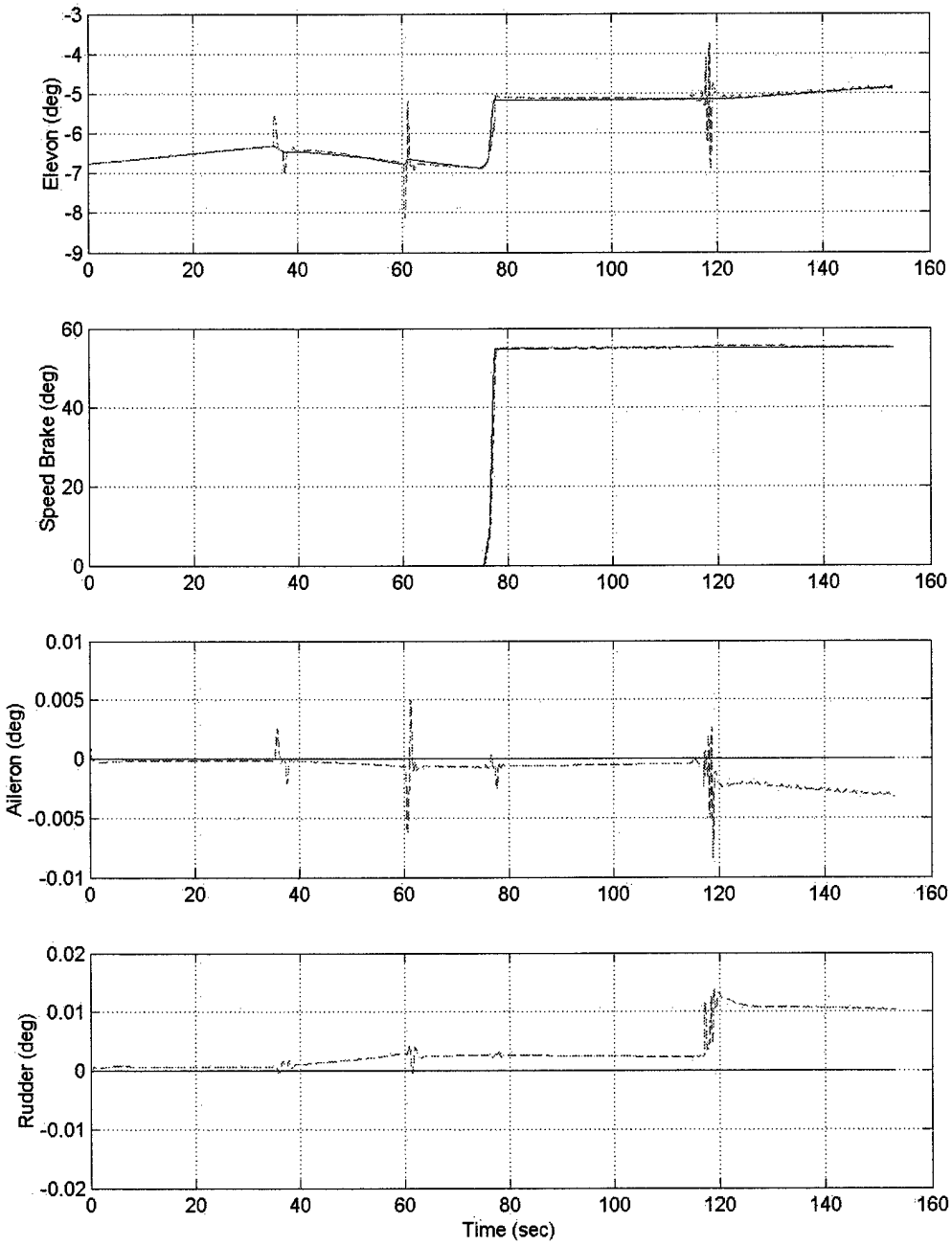

Fig. 75 MPC_SAS Straight (5 of 5)

The MPC_SAS successfully and quickly drives all of the state errors to 0 throughout the "single bank" flight. All of the longitudinal states follow the commands closely with only minor overshoots in step changes in α. Of particular importance, the MPC_SAS has a high inner loop rate controlling the χ and μ states giving a very close tracking of those states. By the nature of the MPC_SAS architecture, conflicts between states are reduced. The MPC controls the outer loop states h, V, and y by generating γ, brake, and χ commands. If the vehicle were to exit the HAC flying parallel to the runway as previously considered, the error in y causes MPC to change the χ command slightly to eliminate the error. An error in χ is not necessarily incurred when making the heading change since the command is altered, thus reducing conflicts between y and χ. The brake being a surface command goes directly to the plant. The γ and χ commands serve as the reference setpoints for the inner loop. Consequently, the γ and χ command signals plotted in Fig. 72 and Fig. 78 originate from the output of the MPC. To better illustrate, Fig. 76 shows the MPC generated command and the feed forward reference signal for γ and χ from the guidance system. The figure also shows the vehicle's final response. The MPC generates a command similar to the guidance command, but makes the appropriate variations to minimize the total error. The γ command is not significantly changed from the γ reference signal from guidance. The χ command does change noticeably from the guidance command. The vehicle then tracks the guidance command very closely with only a small overshoot in heading. The MPC commands were plotted in the MPC_SAS plots for the "straight" trajectory, though less noticeable. The γ and χ error plots shown in Fig. 78 represent the difference between the MPC generated command and the vehicle's actual response. The remaining error plots represent the difference between the guidance generated commands and the vehicle's actual response.

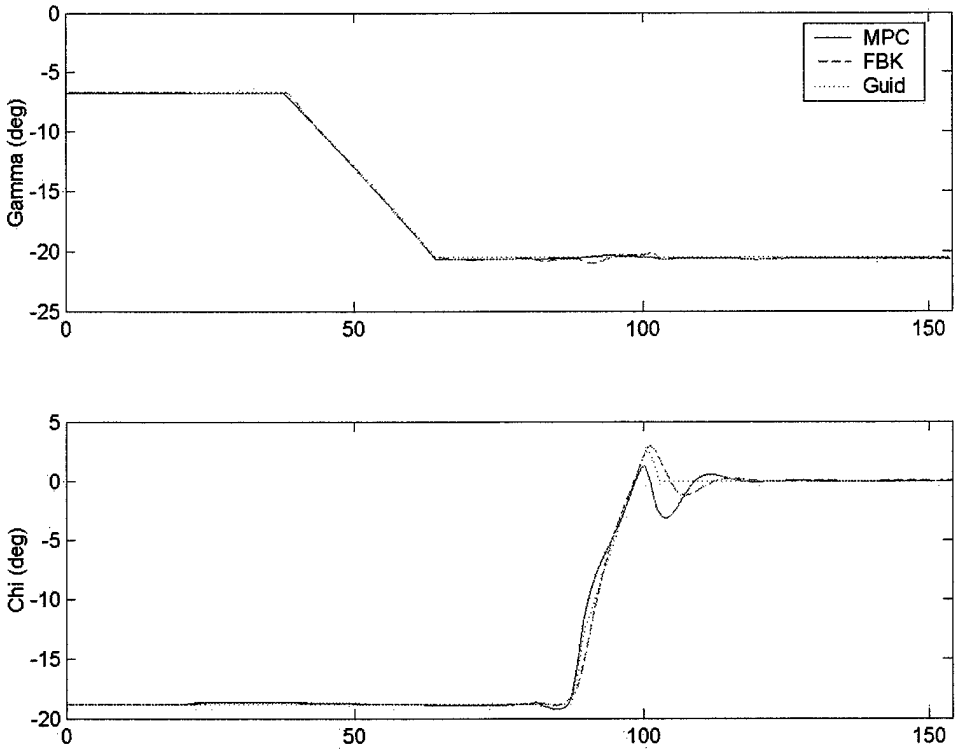

Fig. 76 MPC and Guidance Commands

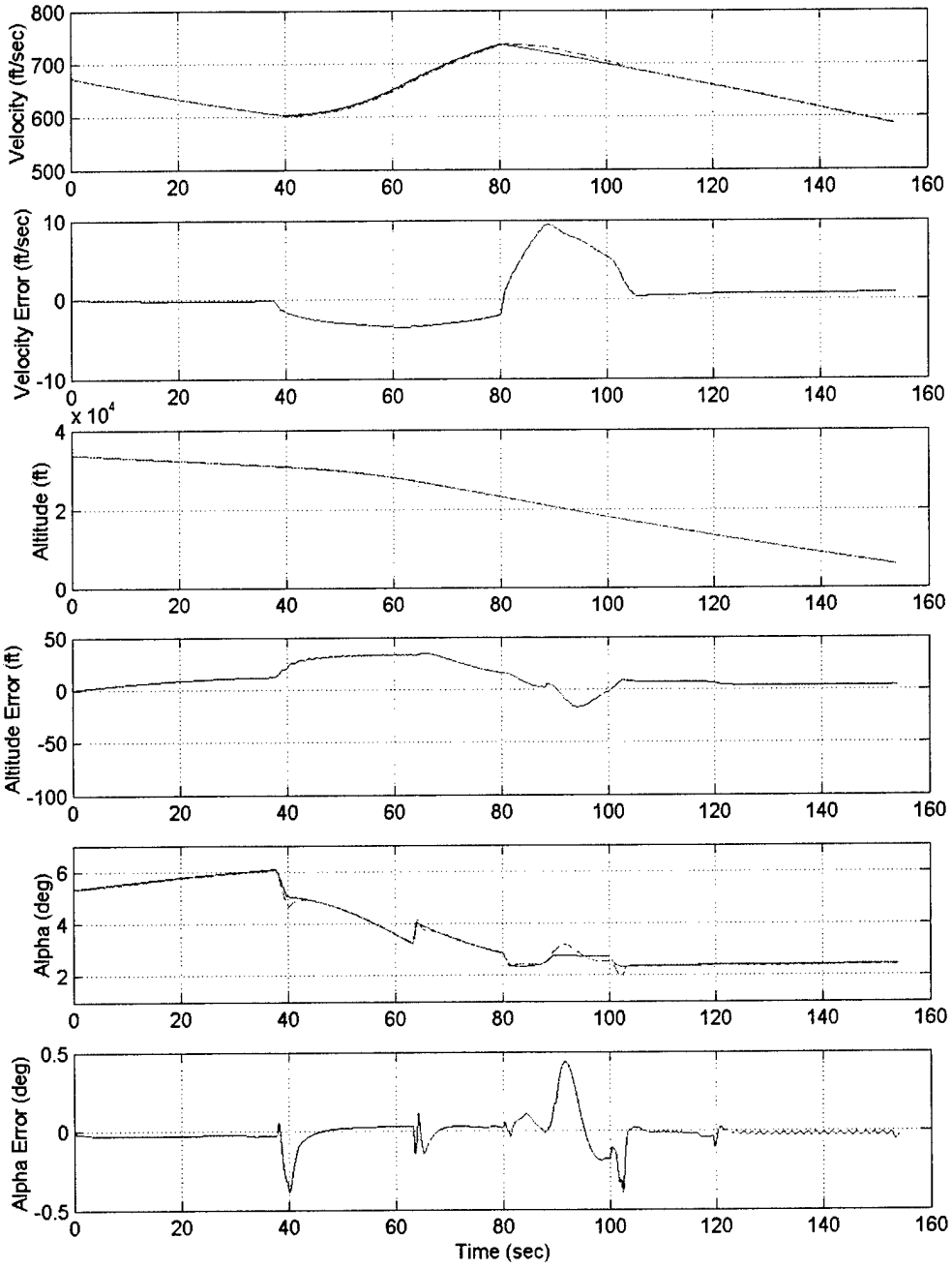

Fig. 77 MPC_SAS Single Bank (1 of 5)

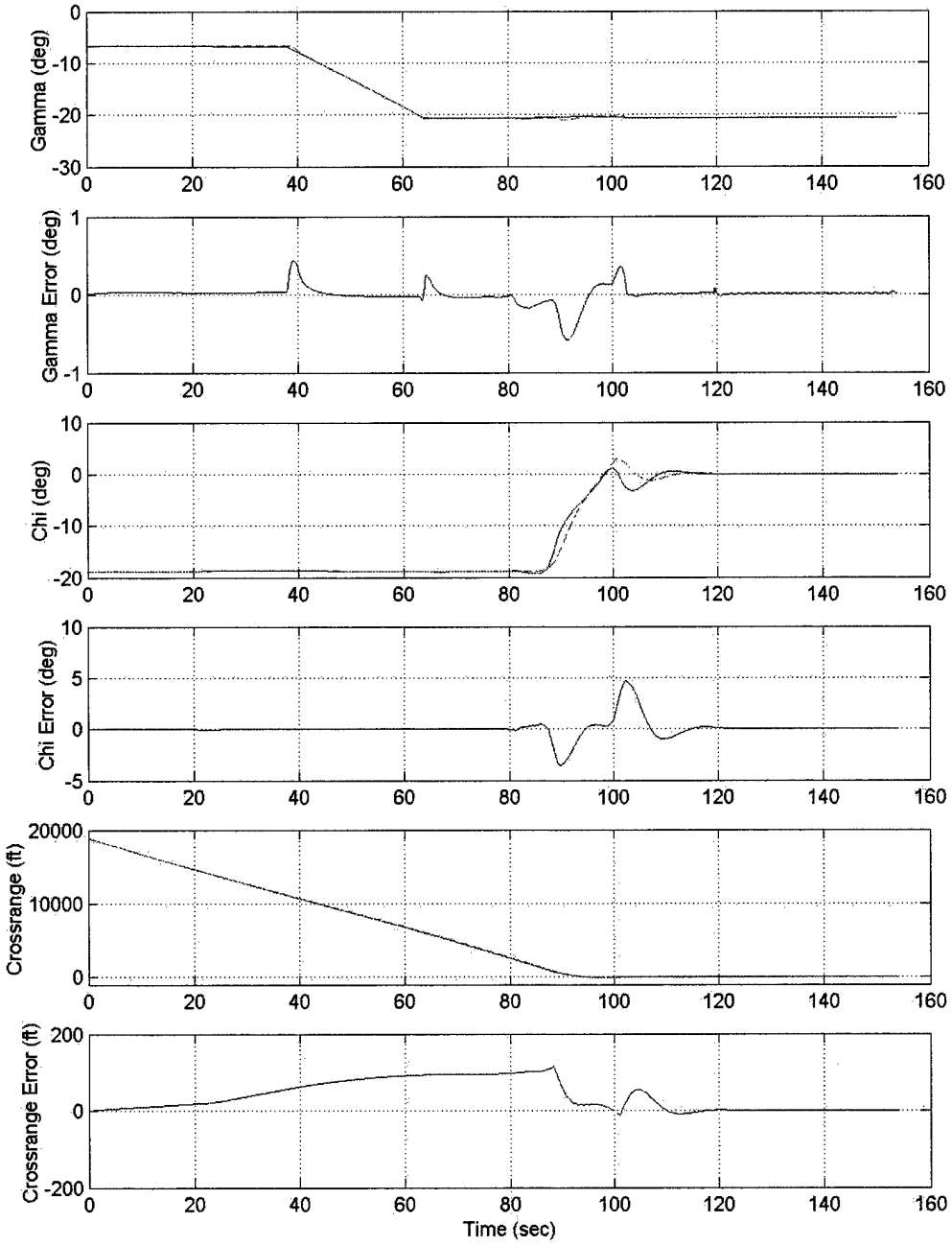

Fig. 78 MPC_SAS Single Bank (2 of 5)

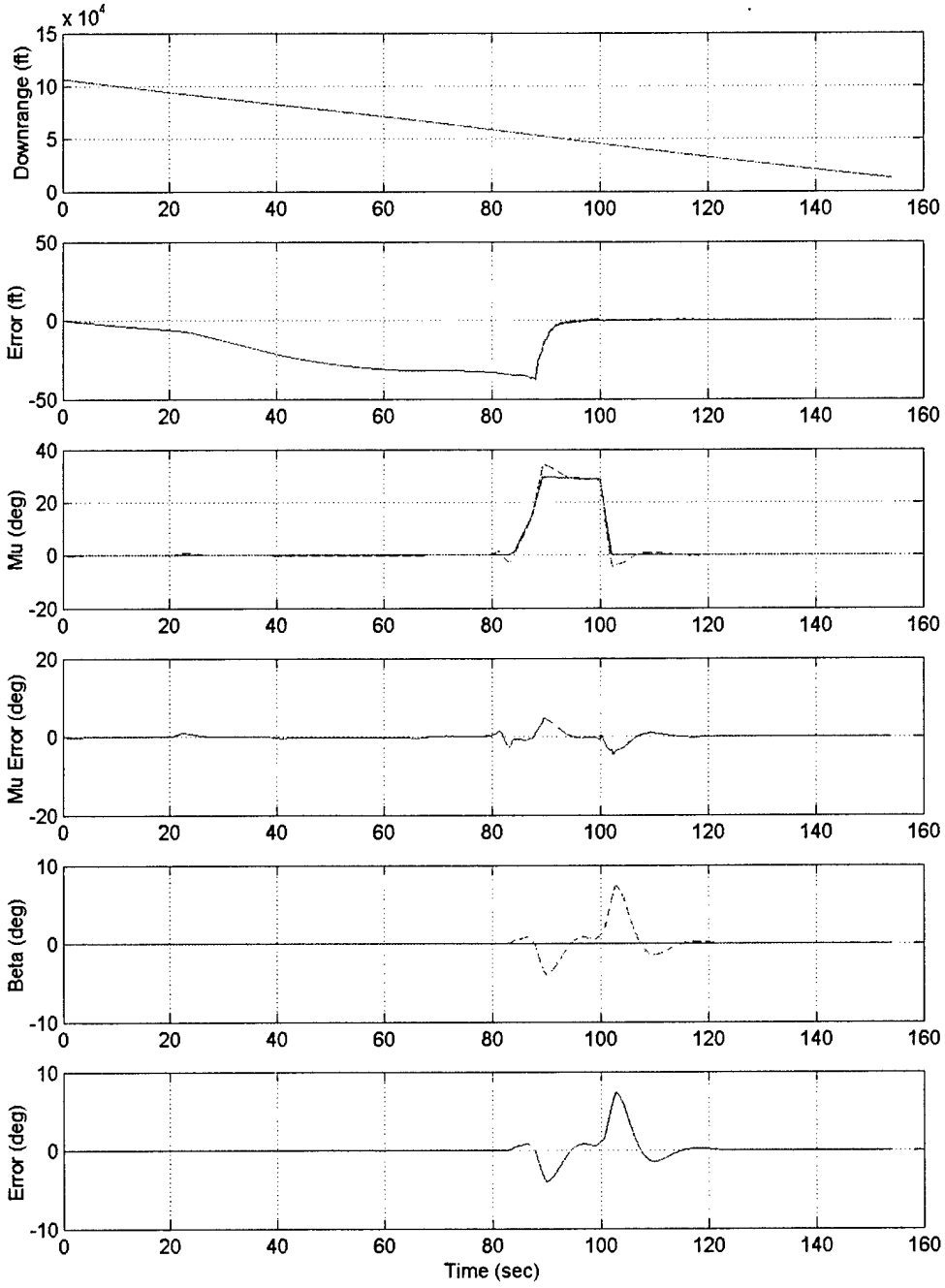

Fig. 79 MPC_SAS Single Bank (3 of 5)

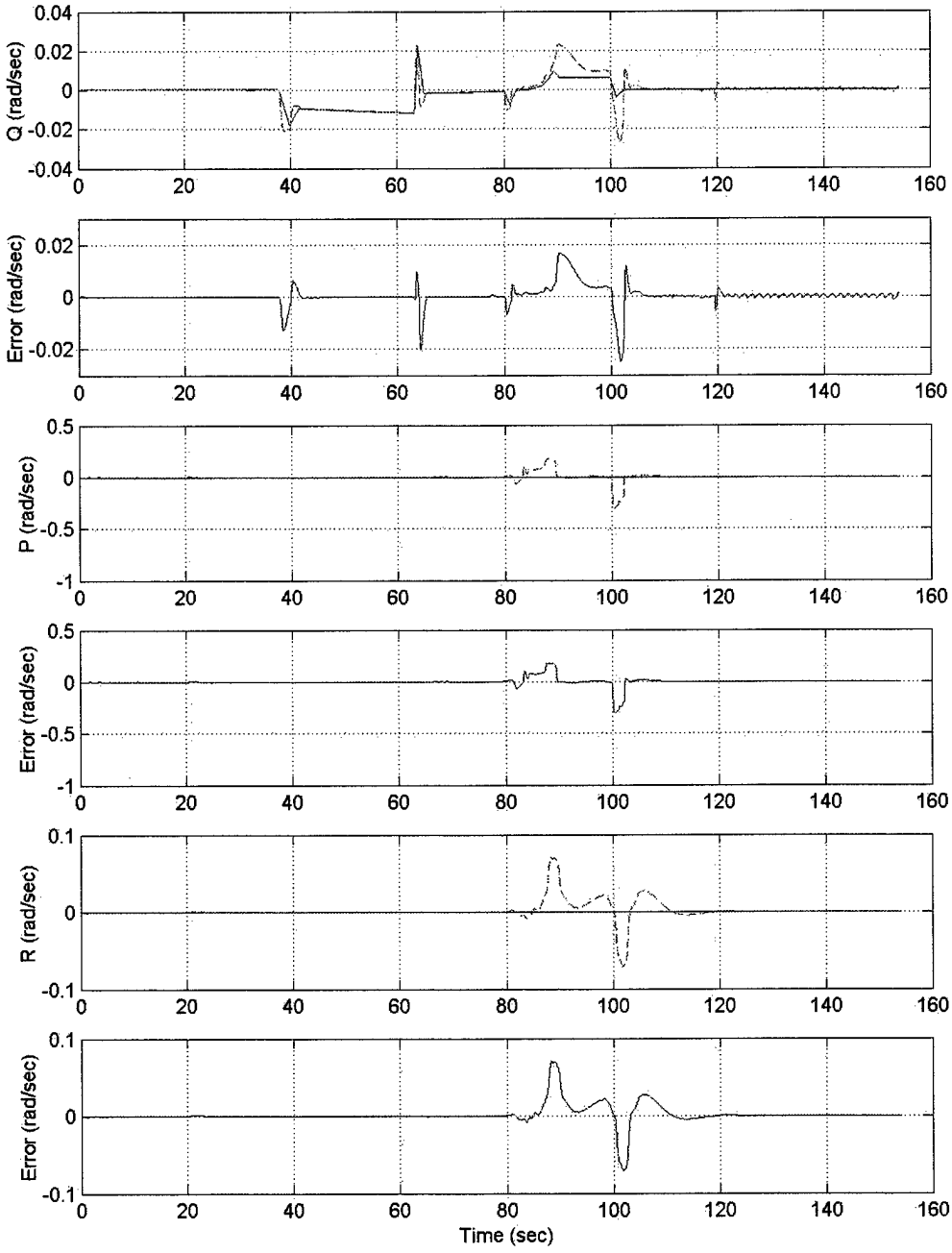

Fig. 80 MPC_SAS Single Bank (4 of 5)

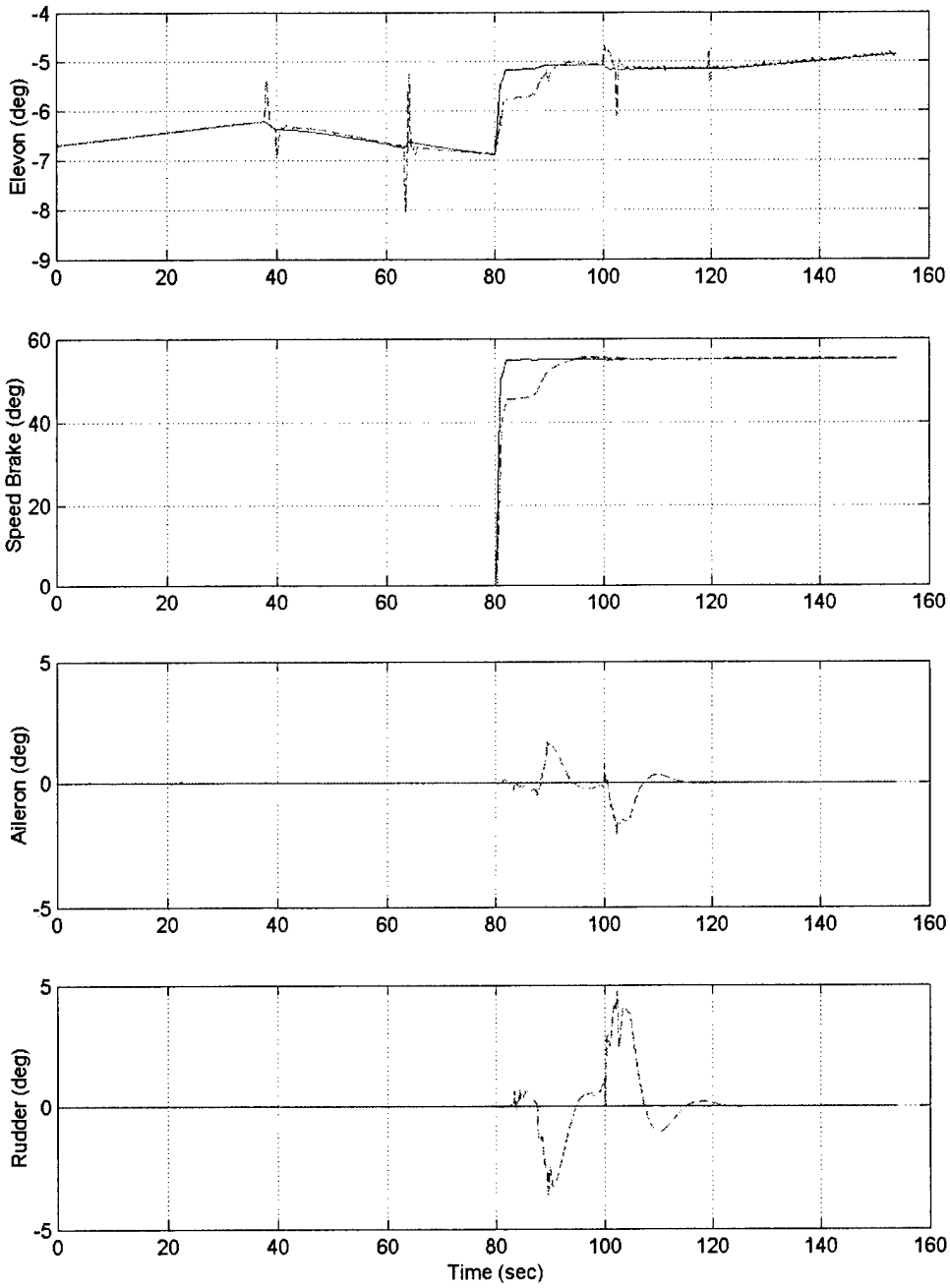

Fig. 81 MPC_SAS Single Bank (5 of 5)

6.2 Constraint Handling

Two examples of constraint handling are shown in this section. The first is an example of how constraints may be applied to the control input to avoid actuator saturation or simply to reduce the peak levels of actuator deflection. The second example shows how MPC performs when encountering a fixed actuator.

Input constraints are hard constraints such that the MPC will do anything to avoid violating the constraints. This differs from the output soft constraints. MPC may break soft constraints, but limits such action as a very high penalty is incurred. Input constraints may be applied to insure the actuators do not saturate. It is also used to prevent large spikes in control action. For example, the MPC_ALL architecture has been weighted to give a close tracking on α. It indeed tracks the command very closely, but at the expense of moderate deviations from the elevon trim condition. For the step changes in α encountered thus far, the elevon did not come near saturation, however, more aggressive maneuvers in other trajectories may yield actuator saturation. The step change in α near the flight point 140 in the "straight" trajectory is revisited. The first 3 graphs in Fig. 82 show a zoomed view of the vehicle's response to the α step, the error, and the elevon movement using MPC_ALL. Constraints are then applied to the elevon prohibiting its movement below –7.5 degrees and above –6.25 degrees. The vehicle's response, error, and the elevon movement are shown in the final 3 plots of Fig. 82.

The constrained flight tracks α very closely and differs from the unconstrained flight by a slight overshoot. The elevon begins its movement near 9.5 seconds with a slight nonminimum phase response. It meets the input constraint slightly faster than the unconstrained case. Since a minimum elevon deflection of about –7.8 degrees minimized the cost function in the unconstrained simulation, it follows that in the constrained flight, the elevon will move as close as possible to the –7.8 position. The constraints prohibit such a deflection, so the elevon moves to –7.5 degrees and holds the position. The elevon completes its movement by meeting, but not exceeding the upper constraint before returning to the trim condition. In this fashion, constraints have been applied to limit the peak deviation of the actuator from trim while still tracking α well. The actuator did not move as far, but held its peak deviation for a longer time to achieve the shown performance.

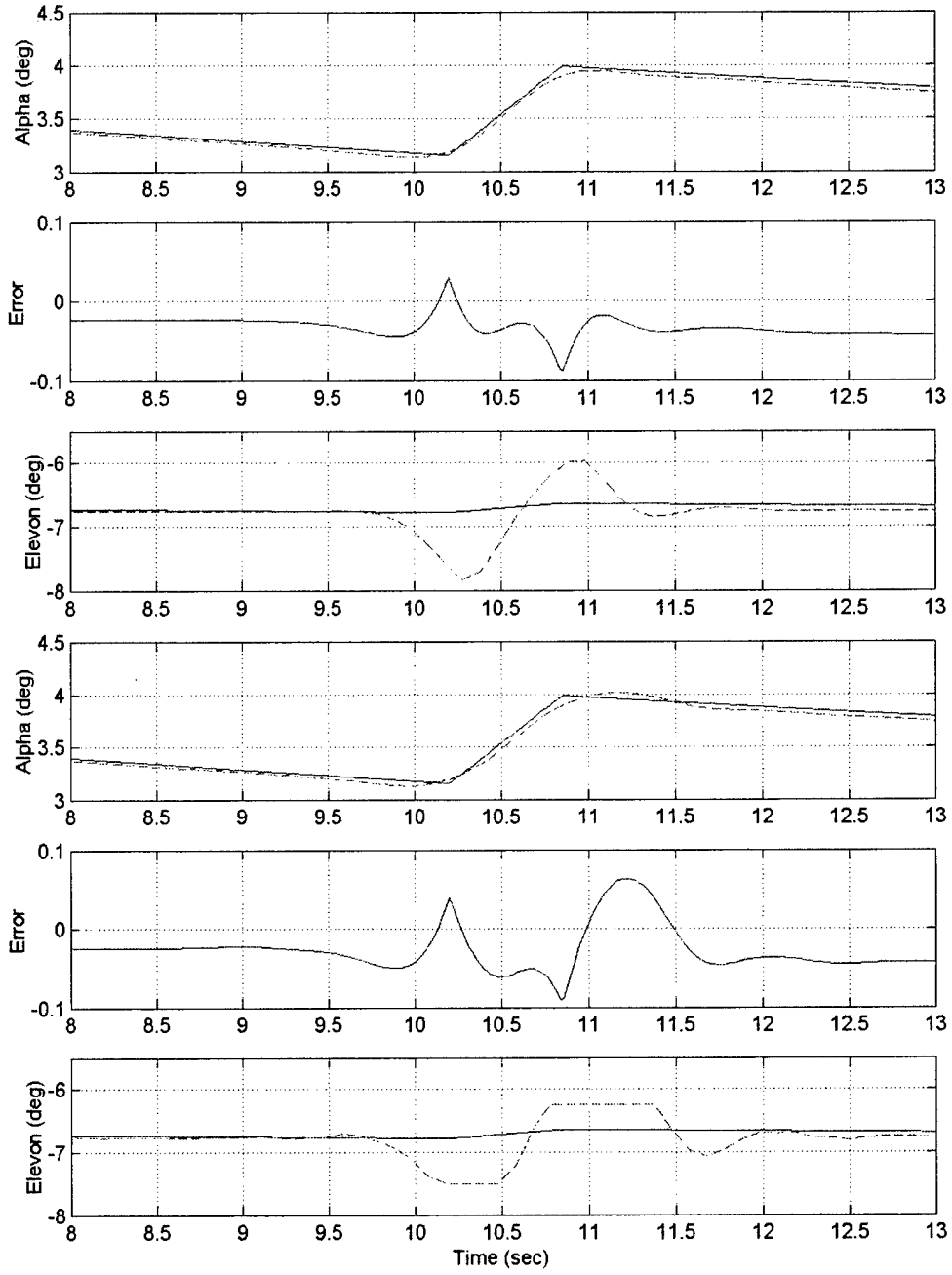

Fig. 82 MPC_ALL Response to a Constrained Input

The second use of constraints simulates a failure in the rudder. The "single bank" trajectory is flown for both architectures holding the rudder fixed at 0 degrees. Because the MPC_ALL generates surface commands, the rudder may be fixed by imposing upper and lower rudder constraints of +-0.001 degrees. The MPC_SAS controller creates γ, speed brake, and χ commands, so the rudder may not be fixed using input constraints. Instead, the LQR rudder weighting is set very high to prevent its movement. In both architectures the control system is aware of the failure. For the MPC_ALL simulation, the MPC controller has information that the rudder is fixed through the use of the hard input constraints that cannot be violated. In contrast, the MPC controller itself does not have any information indicating that the rudder is fixed in the MPC_SAS architecture because the rudder is artificially constrained through the high weighting in the LQR. Consequently, only the inner loop LQR portion of the control system has knowledge of the failure.

MPC_ALL and MPC_SAS are able to fly the trajectory and to reduce the state errors almost to 0 by the end of the flight. A few changes are observed in MPC_ALL compared to the unconstrained flight. The most significant differences are seen in the aerosurfaces. The aileron of the constrained flight experiences low amplitude, high frequency oscillation and a decreased peak value. The oscillation in the aileron increases the settling time slightly for y, χ, and μ. The elevon also experiences low amplitude, high frequency oscillation at the beginning of the banking flight phase and continuing for the remainder of the flight. The elevon's oscillation creates a very fine oscillation in α that is only observable in the error plot.

The MPC_SAS controller shows little change in the tracking performance of the longitudinal states compared to the unconstrained flight. The most substantial difference is seen in the lateral channel where the bank angle overshoot is greater than even the MPC_ALL simulation. The sideslip angle is significantly reduced in magnitude, but slightly oscillatory. The aileron reduces in peak deviation, but develops oscillation. Very low amplitude oscillation emerges in the elevon in the final 30 seconds of flight.

System performance is not excellent in the fixed actuator simulation, but the vehicle is able to complete the heading alignment and make it to the final landing phase ending with minimal state errors.

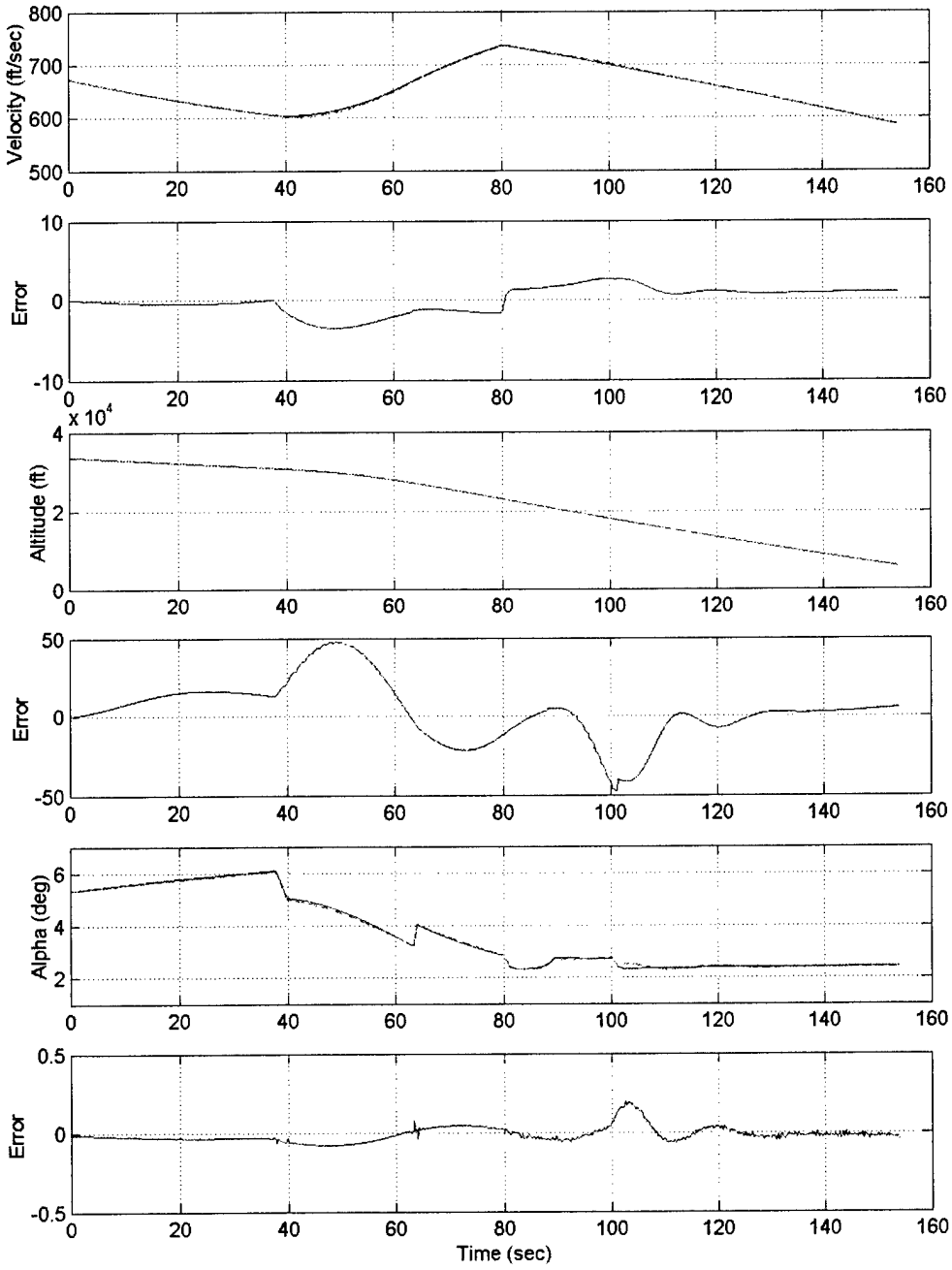

Fig. 83 MPC_ALL Fixed Rudder (1 of 5)

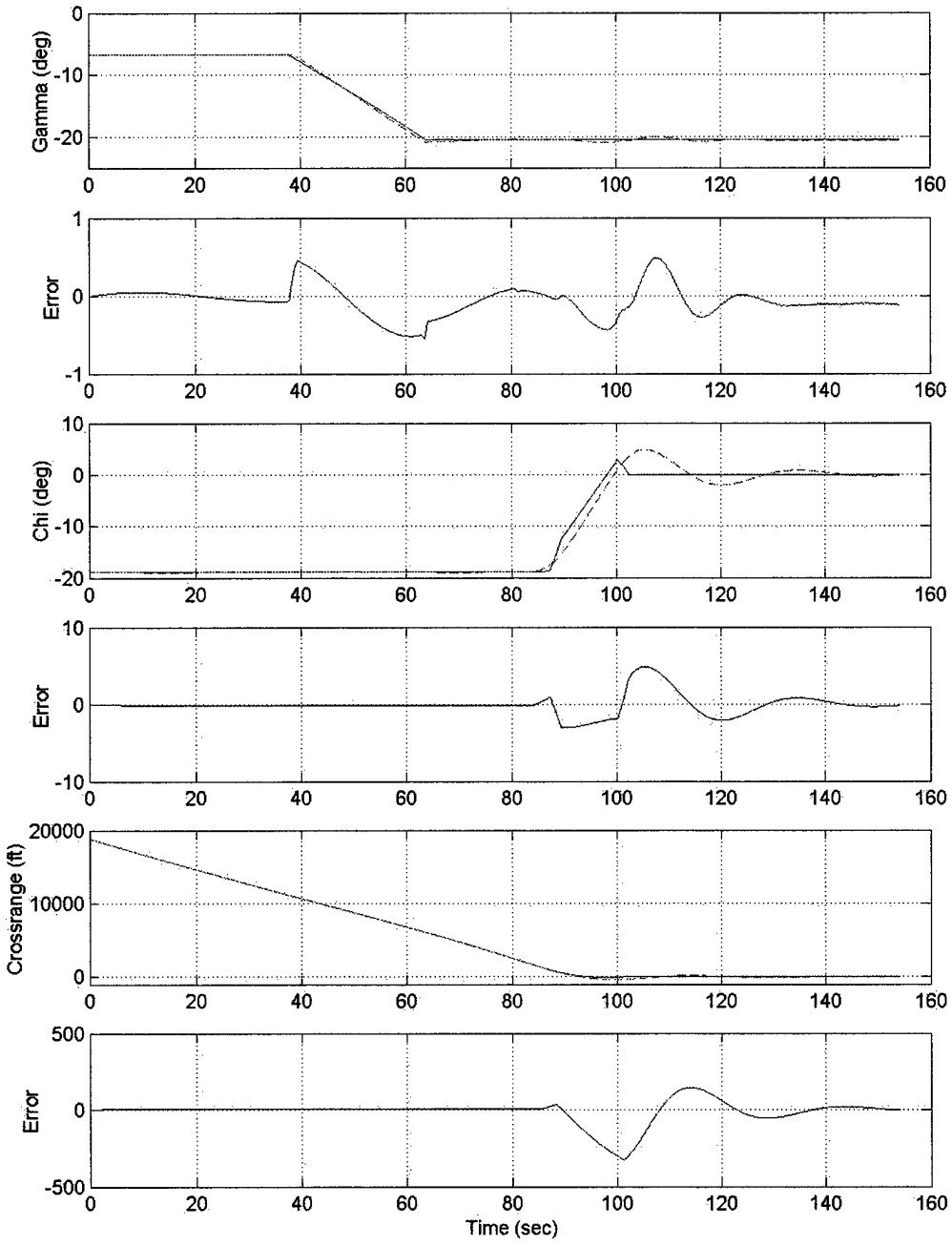

Fig. 84 MPC_ALL Fixed Rudder (2 of 5)

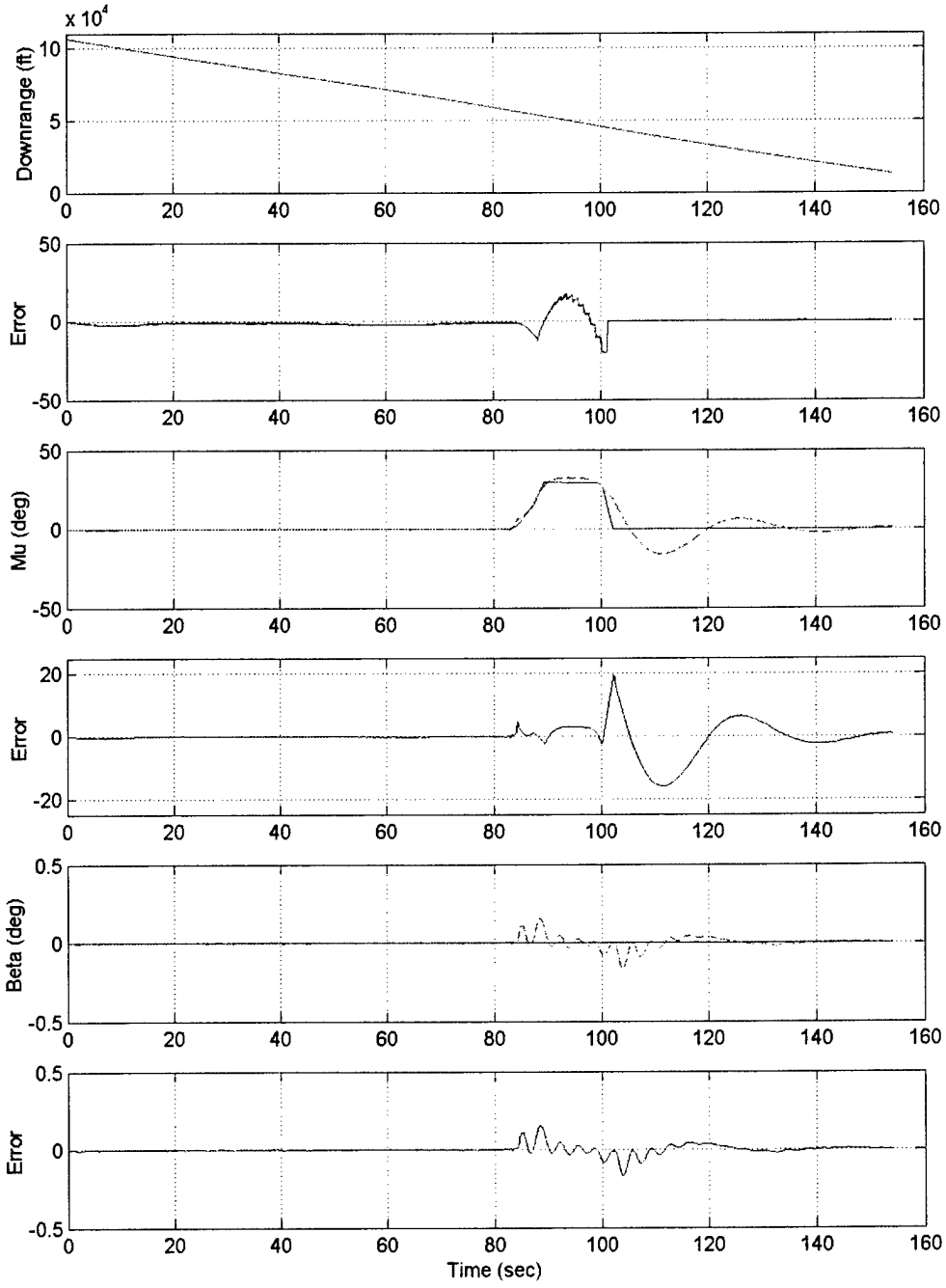

Fig. 85 MPC_ALL Fixed Rudder (3 of 5)

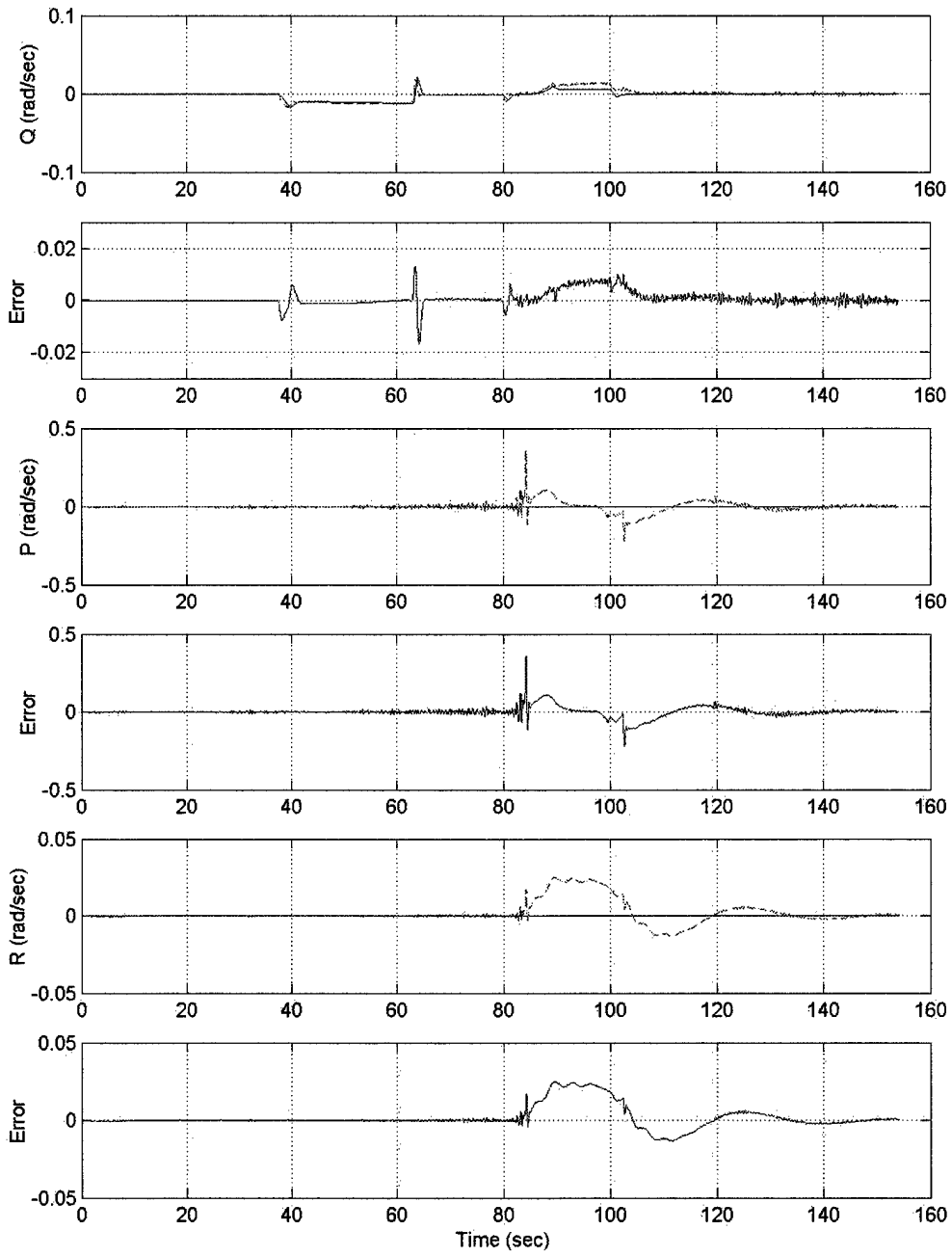

Fig. 86 MPC_ALL Fixed Rudder (4 of 5)

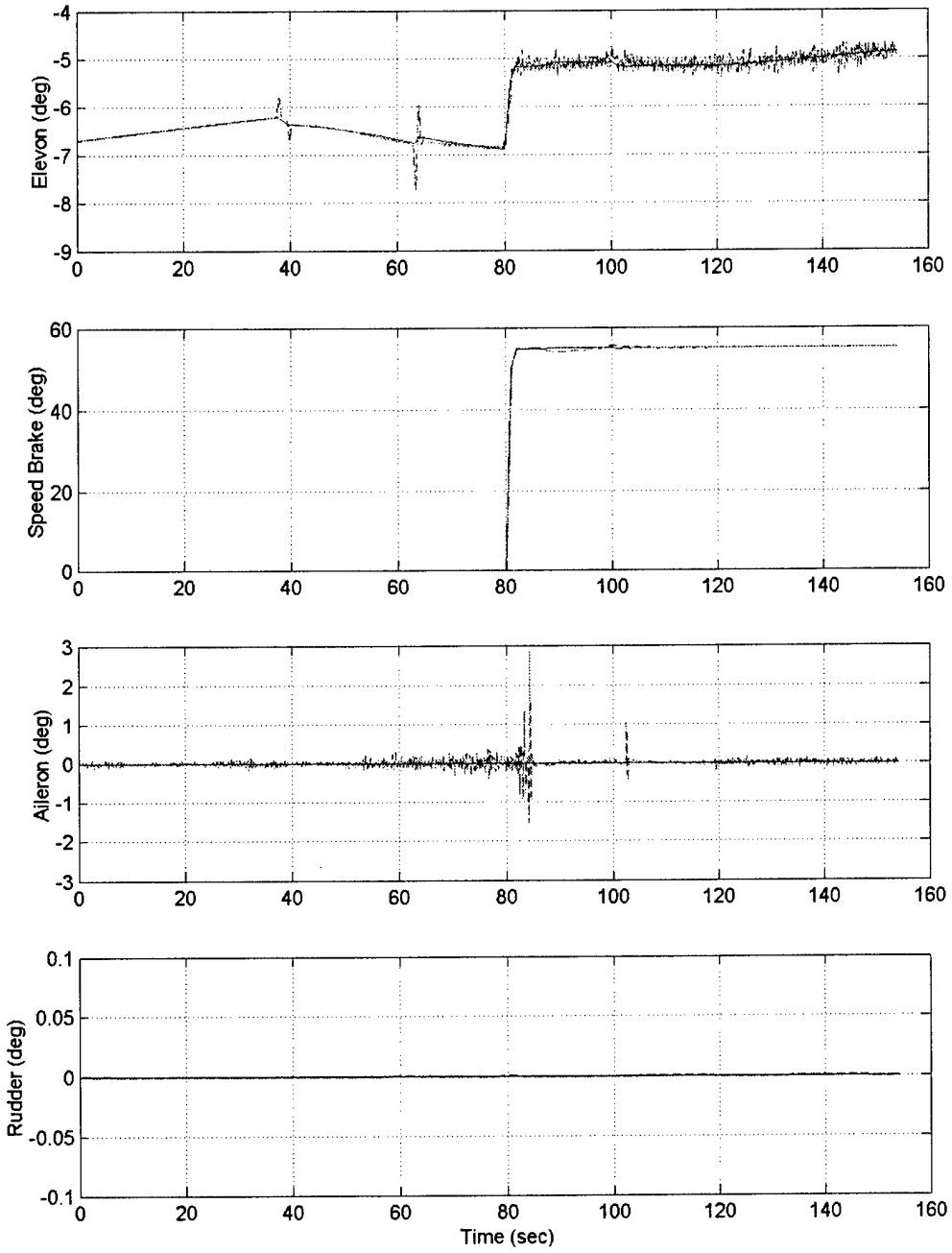

Fig. 87 MPC_ALL Fixed Rudder (5 of 5)

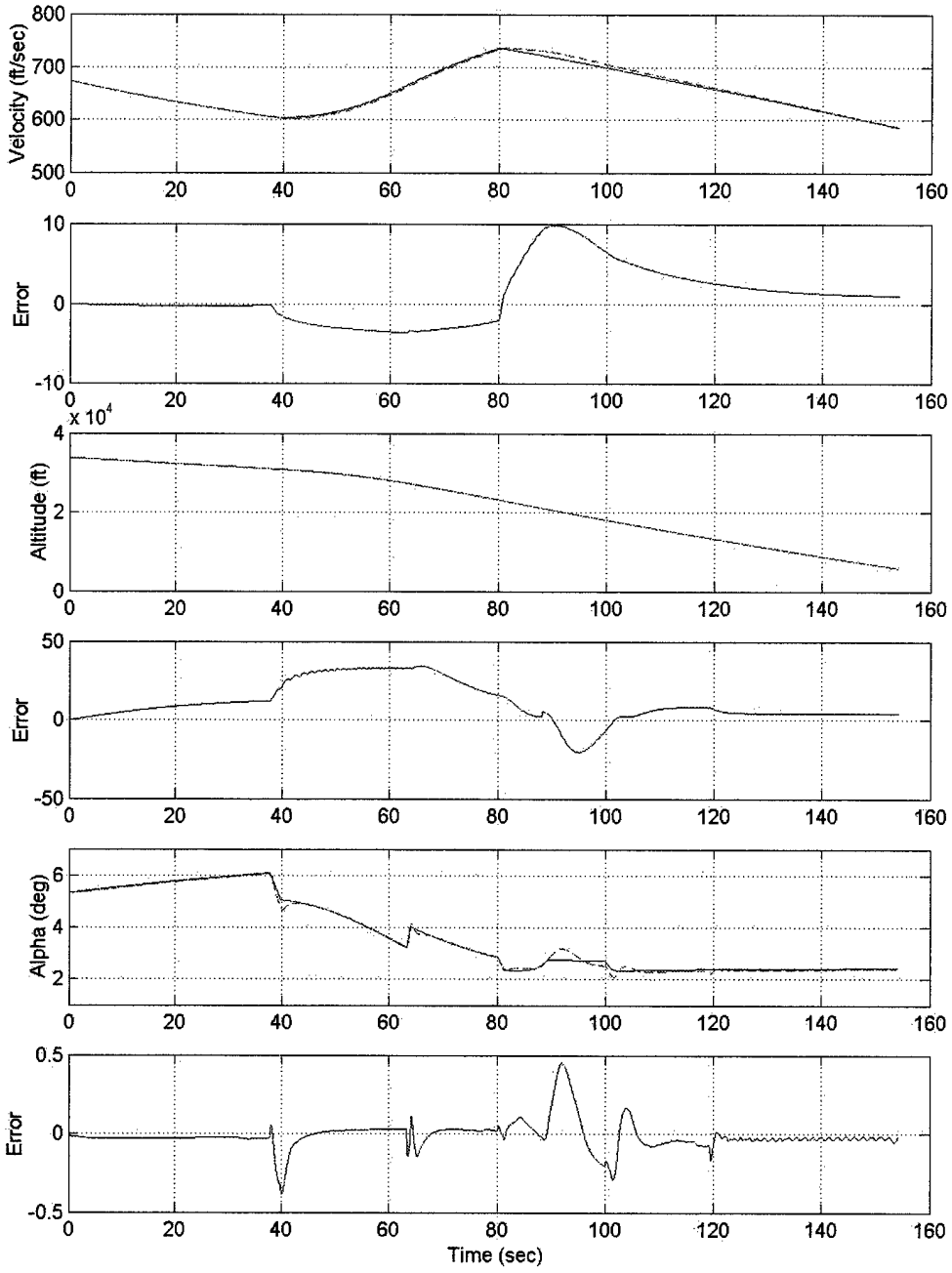

Fig. 88 MPC_SAS Fixed Rudder (1 of 5)

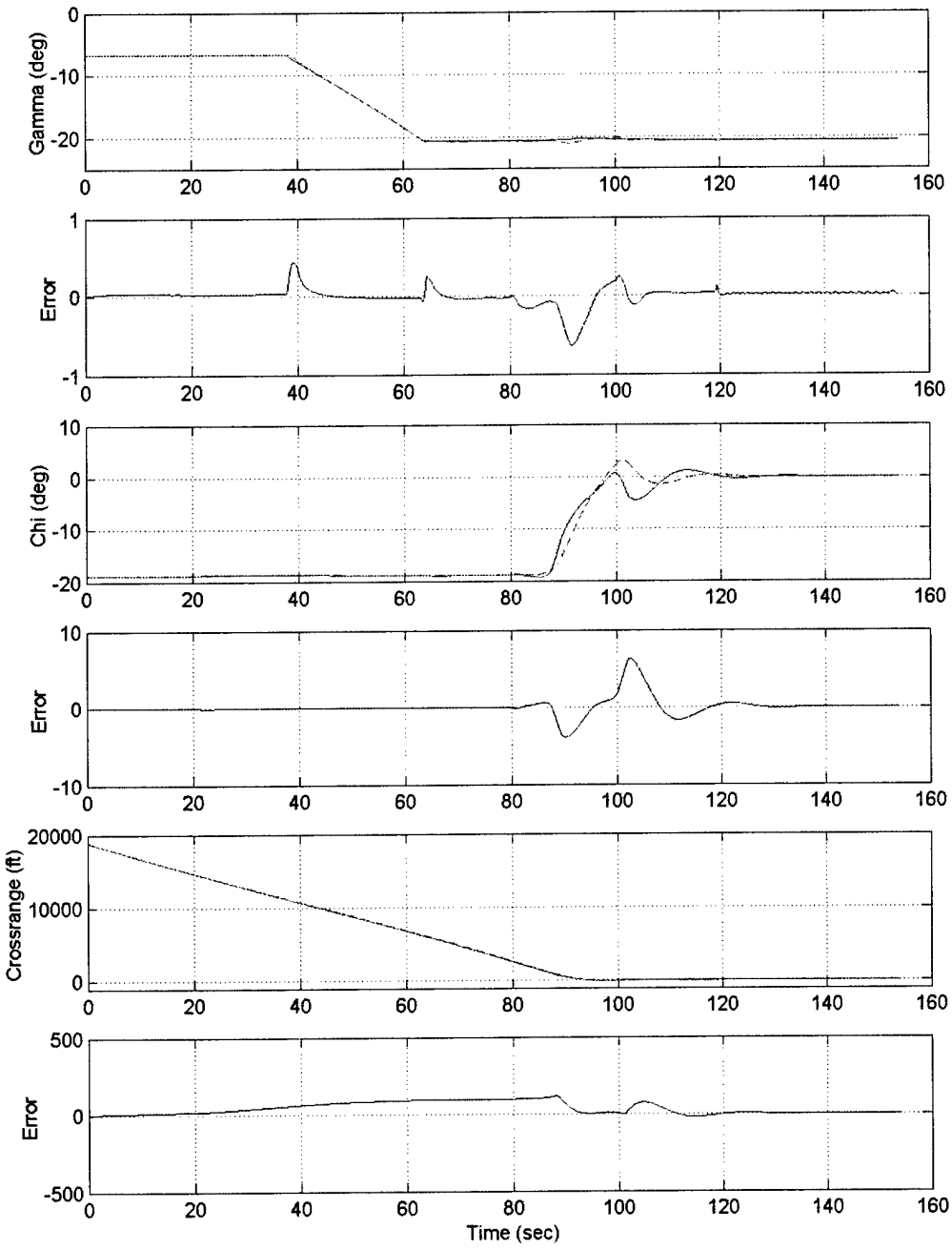

Fig. 89 MPC_SAS Fixed Rudder (2 of 5)

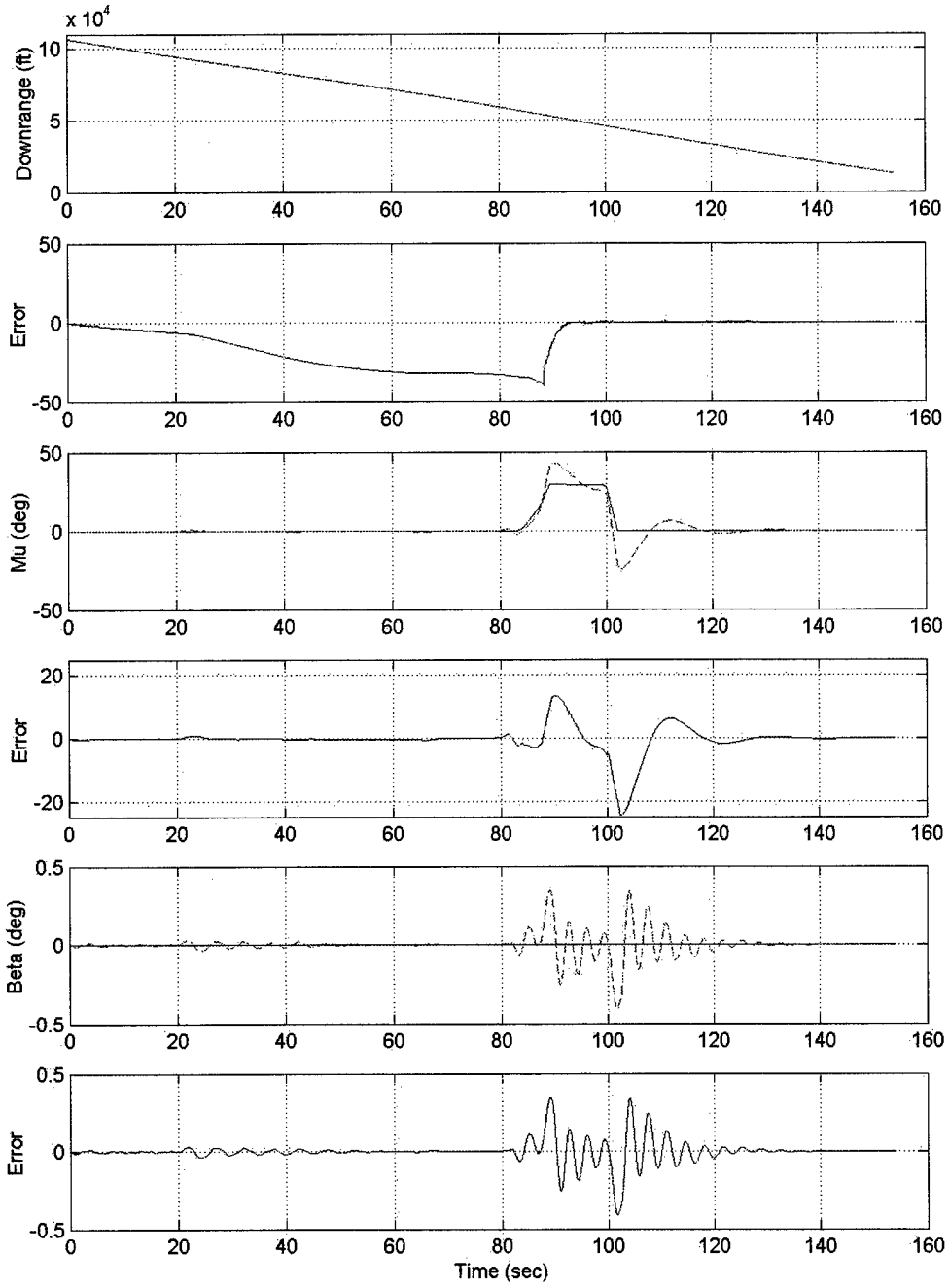

Fig. 90 MPC_SAS Fixed Rudder (3 of 5)

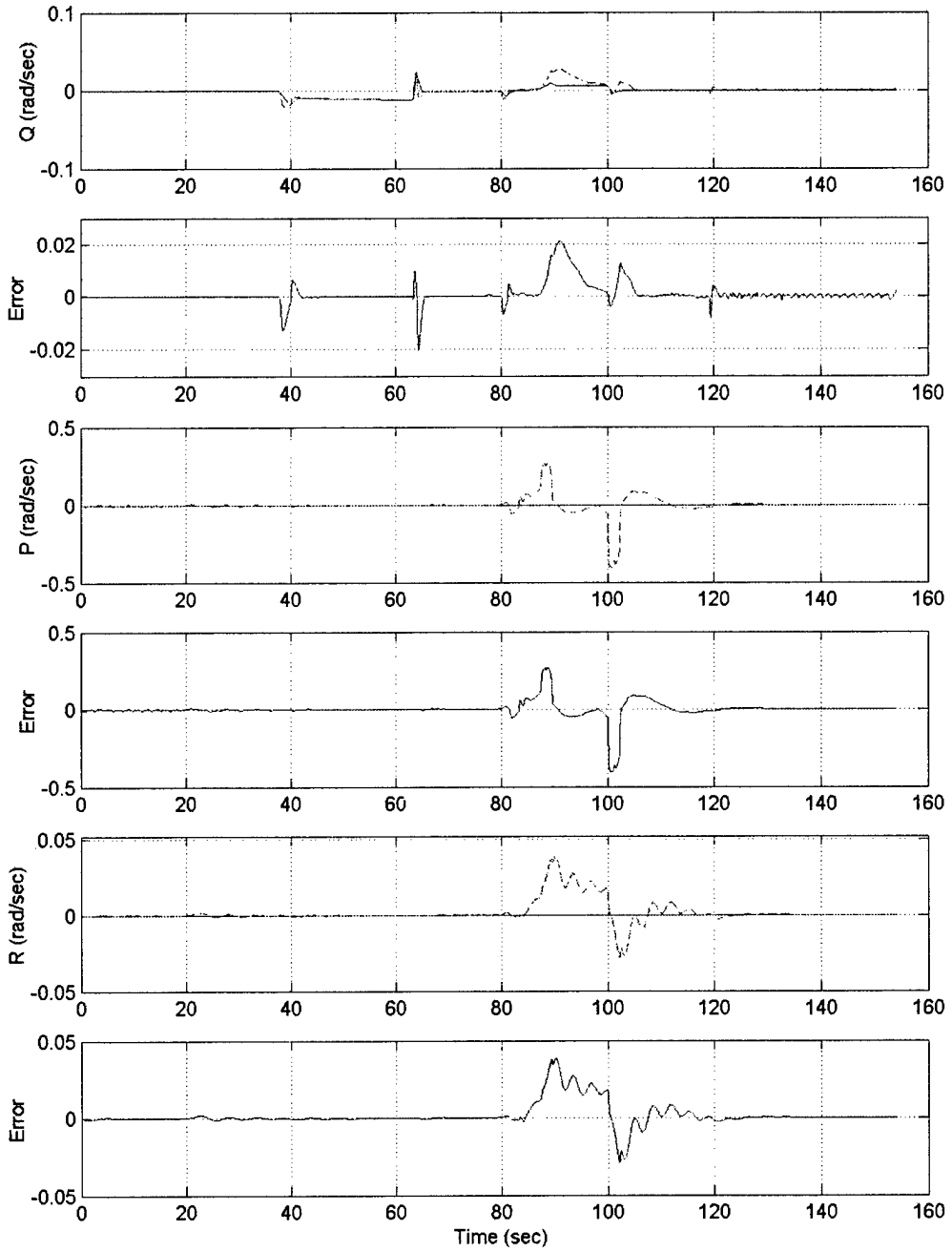

Fig. 91 MPC_SAS Fixed Rudder (4 of 5)

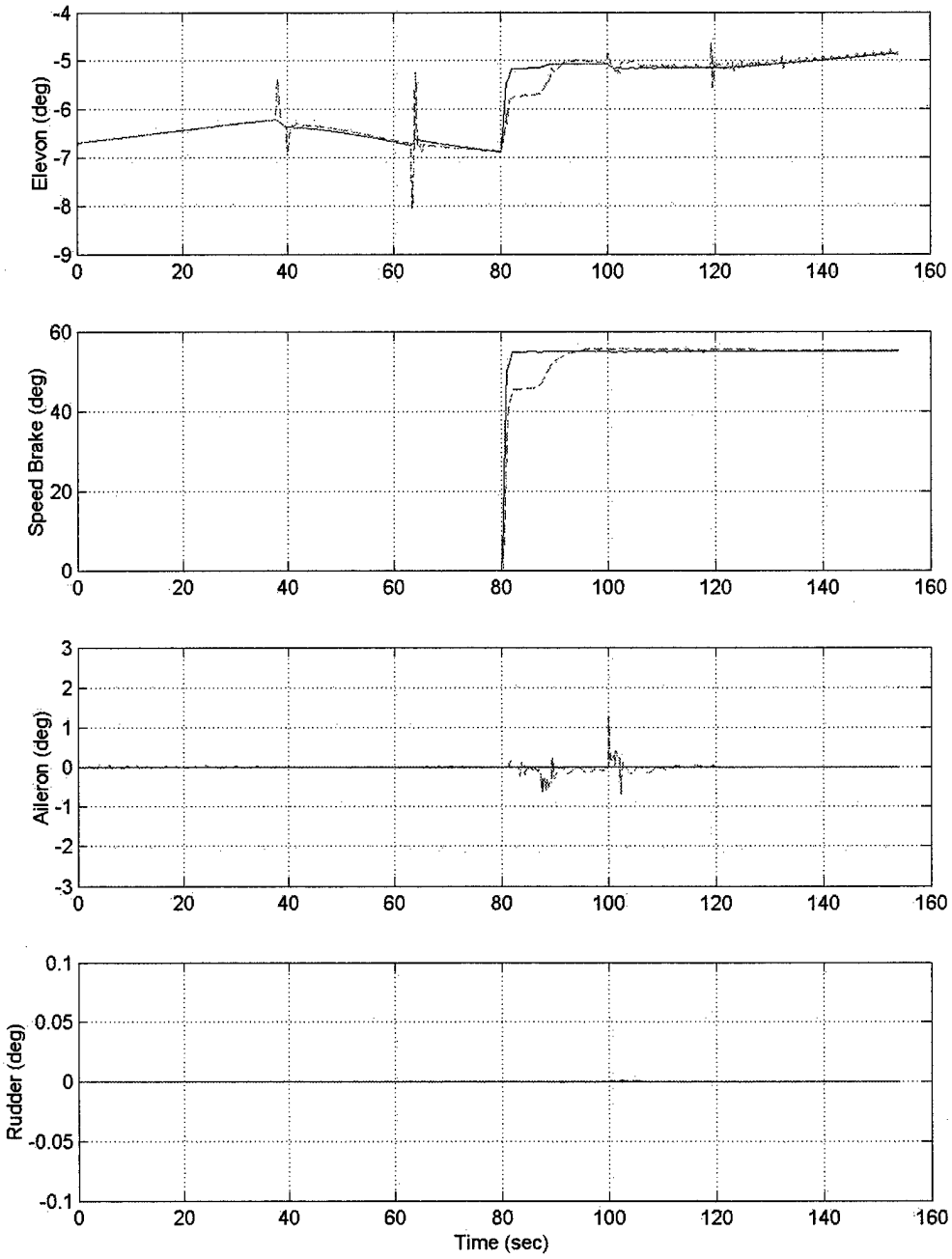

Fig. 92 MPC_SAS Fixed Rudder (5 of 5)

6.3 Response to an Imperfect Internal Model

To simulate the system's response to an imperfect model two imperfections are considered. The first is a situation where the vehicle has a 20% greater pitching moment coefficient than represented in the internal model. Similarly, the second imperfection is a 20% decrease in the moment coefficient than represented in the internal model. The goal of these tests is to evaluate how well the MPC adjusts to a mismatch between the internal model and actual plant. The MPC_ALL and MPC_SAS are evaluated flying the "straight" trajectory.

The MPC_ALL flying with a 20% greater pitching moment coefficient is presented first and shows nearly identical performance in the lateral channel to the nominal case as expected. The only noticeable difference is a slight increase in oscillation in the lateral rates, aileron, and rudder during the transition between weighting matrices near 120 seconds of flight. In the longitudinal channel, the velocity tracking is comparable to the nominal case. γ is not as closely tracked and the transition in weighting matrices is noticeable. The most prominent effect of the model mismatch is the bias in α raising the response above the command and the bias lowering the elevon. The model is not expecting such a strong pitching moment and the vehicle's nose subsequently rises. The elevon assumes a lower value attempting to lower the vehicle's angle of attack so that closer tracking is achieved. The nearly constant biases in angle of attack and the elevon result from the weighting matrices. The vehicle is unable to satisfy both the α tracking and the elevon trim position, so the MPC finds the medium that minimizes the total cost. The high weighting on α causes a close tracking, keeping the average error below about 0.25 degrees. The elevon then holds a near constant error of about 1 degree. Reduced tracking precision in altitude results from the constant α and elevon biases. The vehicle remains about 40 feet low going into the final landing portion of flight.

The MPC_ALL flight with a 20% decrease in the pitching moment coefficient follows suit with a negative bias on α and a positive bias in the elevon. The altitude error leaves the vehicle slightly higher than the nominal flight. The performance trends seen in the MPC_ALL flights continue with the MPC_SAS with the exception of the altitude error. The MPC_ALL held a nonzero altitude error, but the MPC_SAS was able to completely eliminate the altitude errors by the end of the flight. MPC_SAS experienced a significant

disruption when changing weighting matrices particularly in the final case with the 20% decrease in the moment coefficient showing that a more detailed smoothing scheme is needed to transition between weighting matrices.

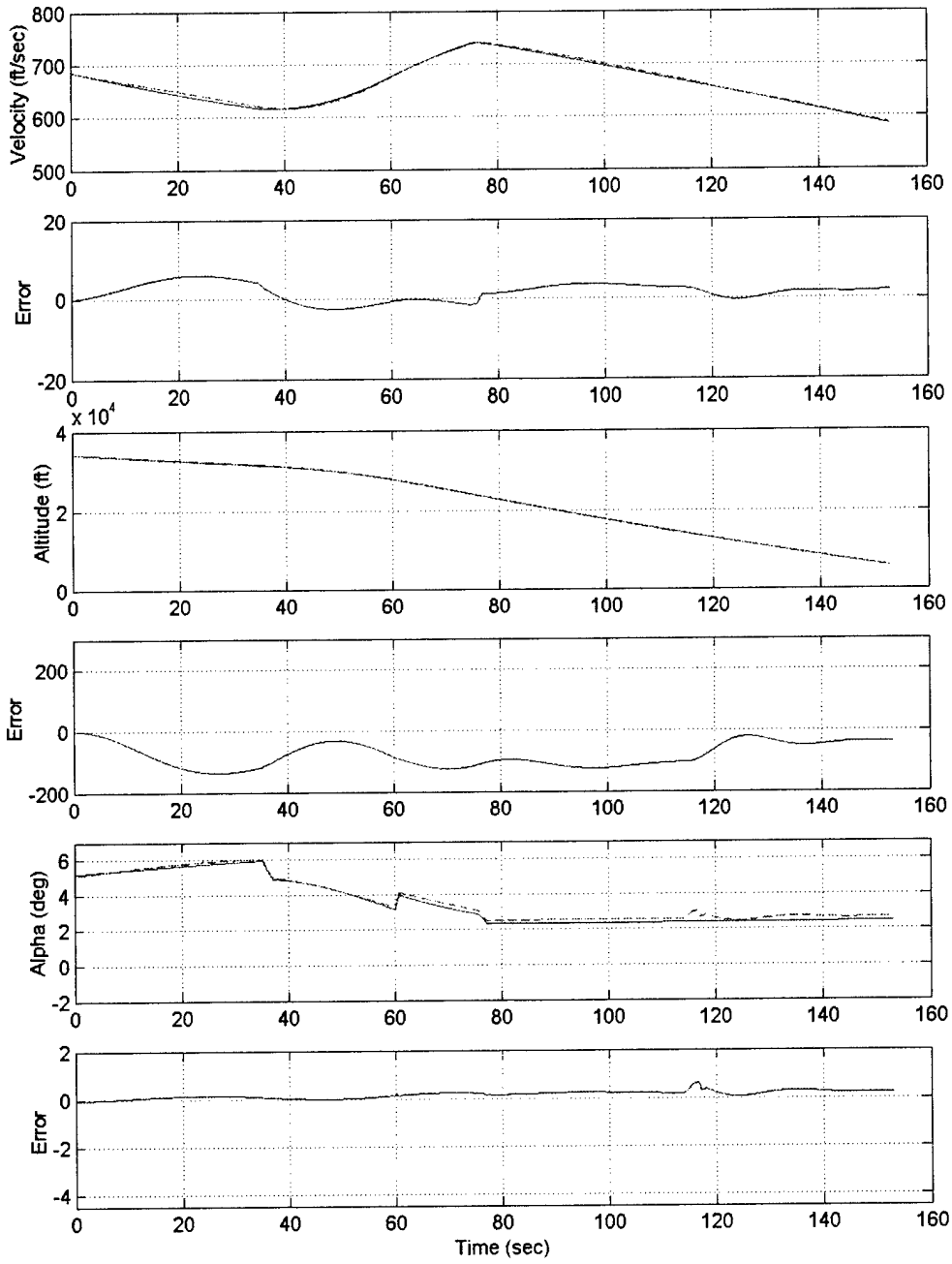

Fig. 93 MPC_ALL 20% Pitching Moment Coefficient Increase (1 of 5)

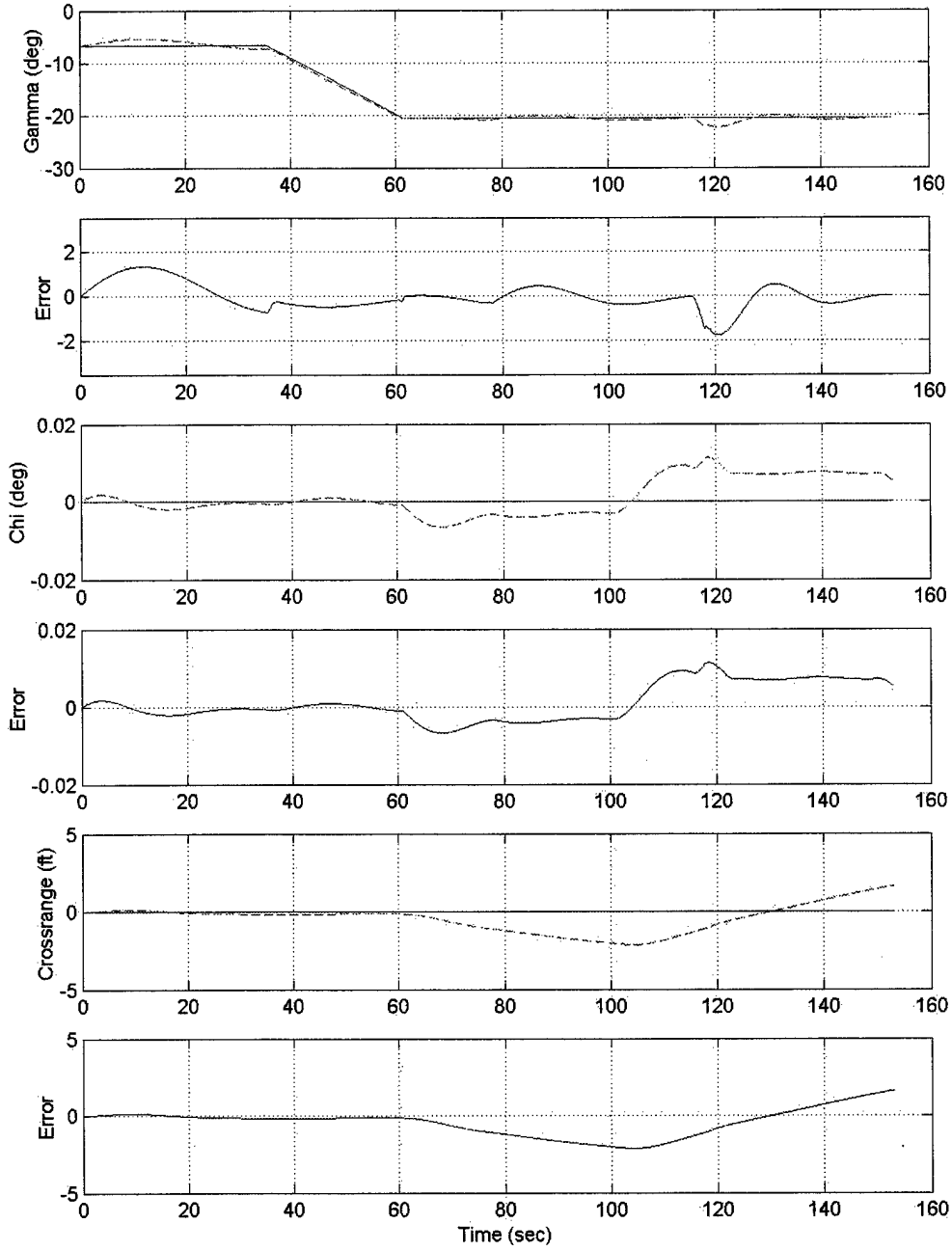

Fig. 94 MPC_ALL 20% Pitching Moment Coefficient Increase (2 of 5)

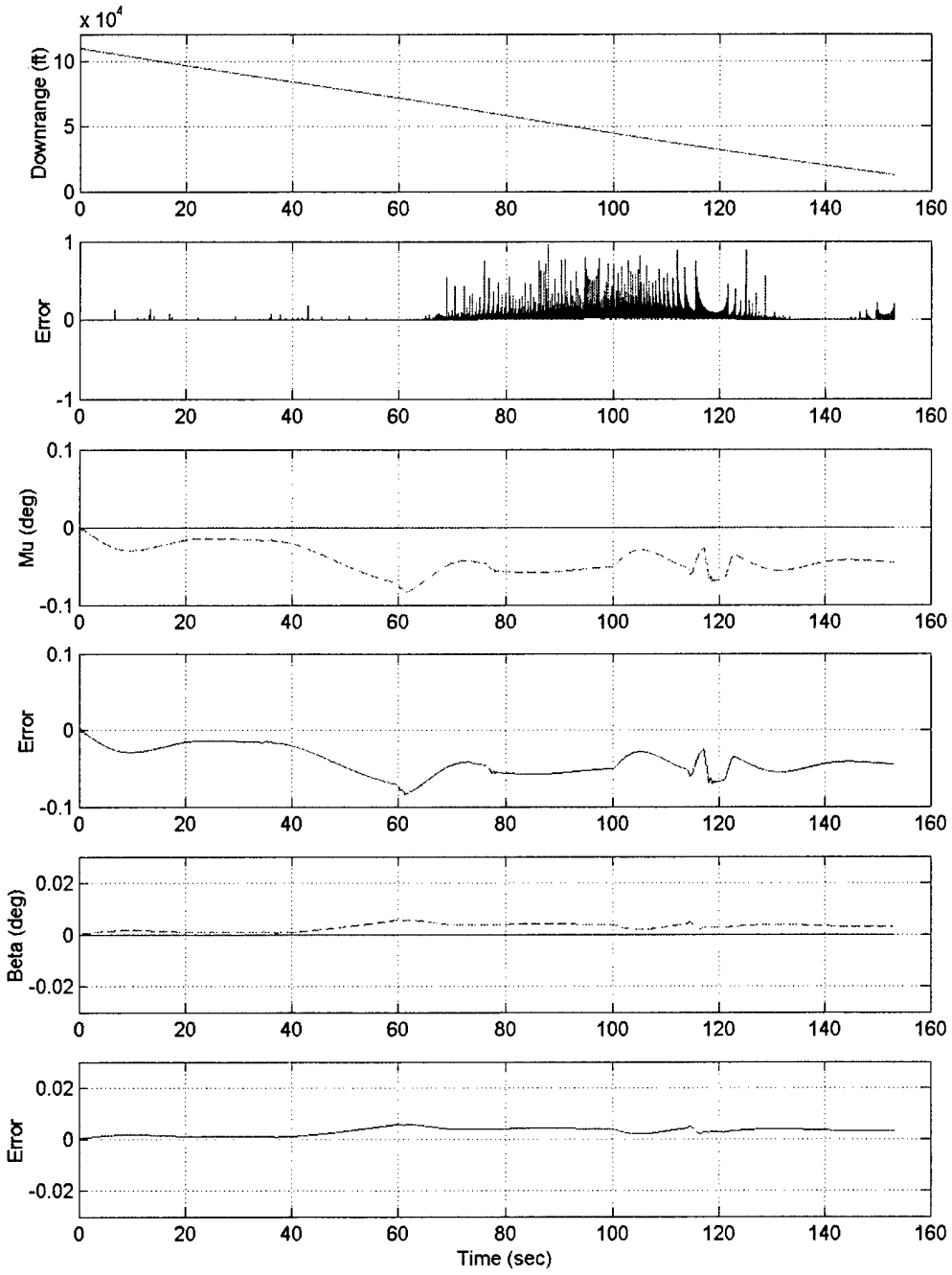

Fig. 95 MPC_ALL 20% Pitching Moment Coefficient Increase (3 of 5)

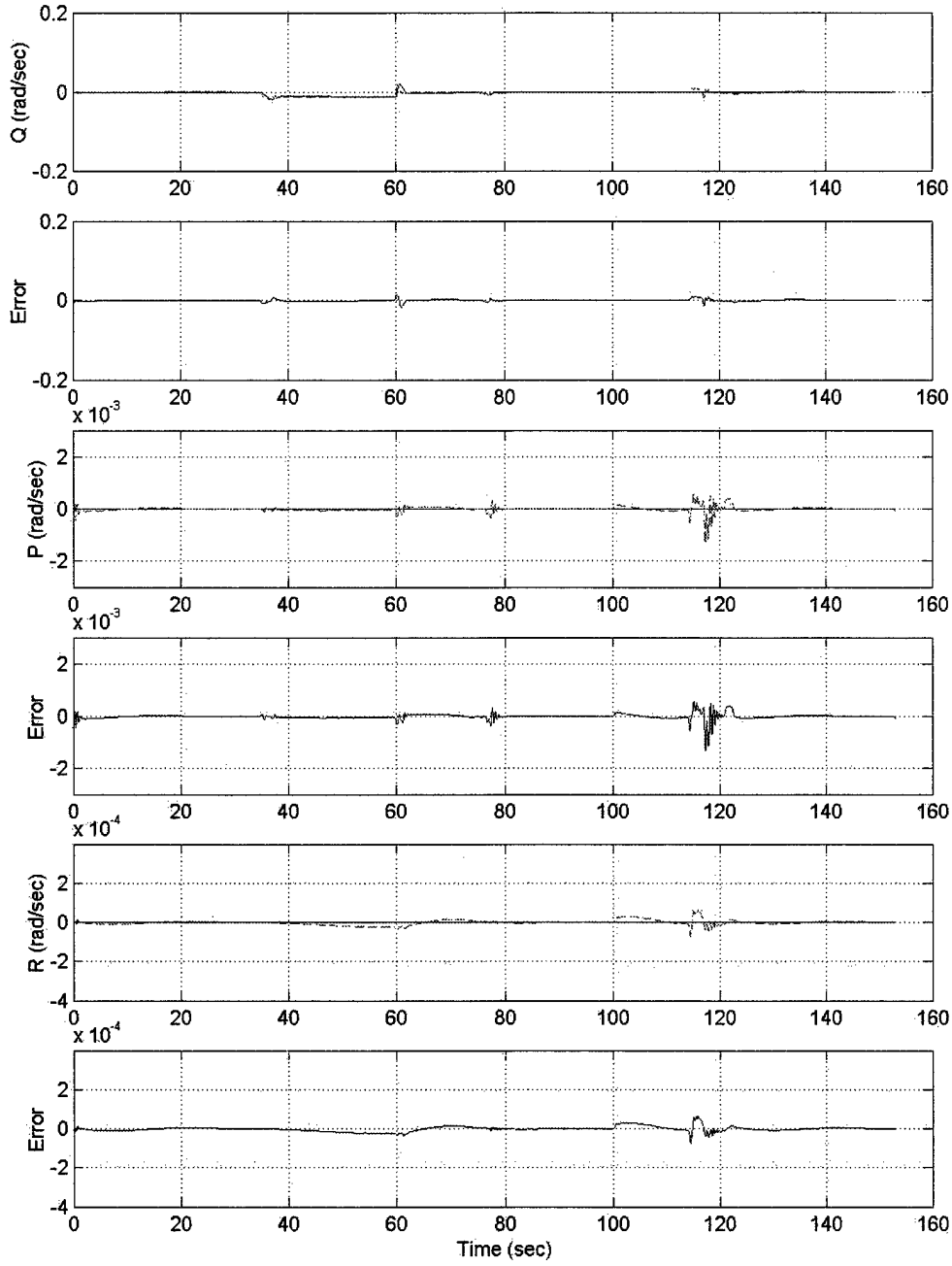

Fig. 96 MPC_ALL 20% Pitching Moment Coefficient Increase (4 of 5)

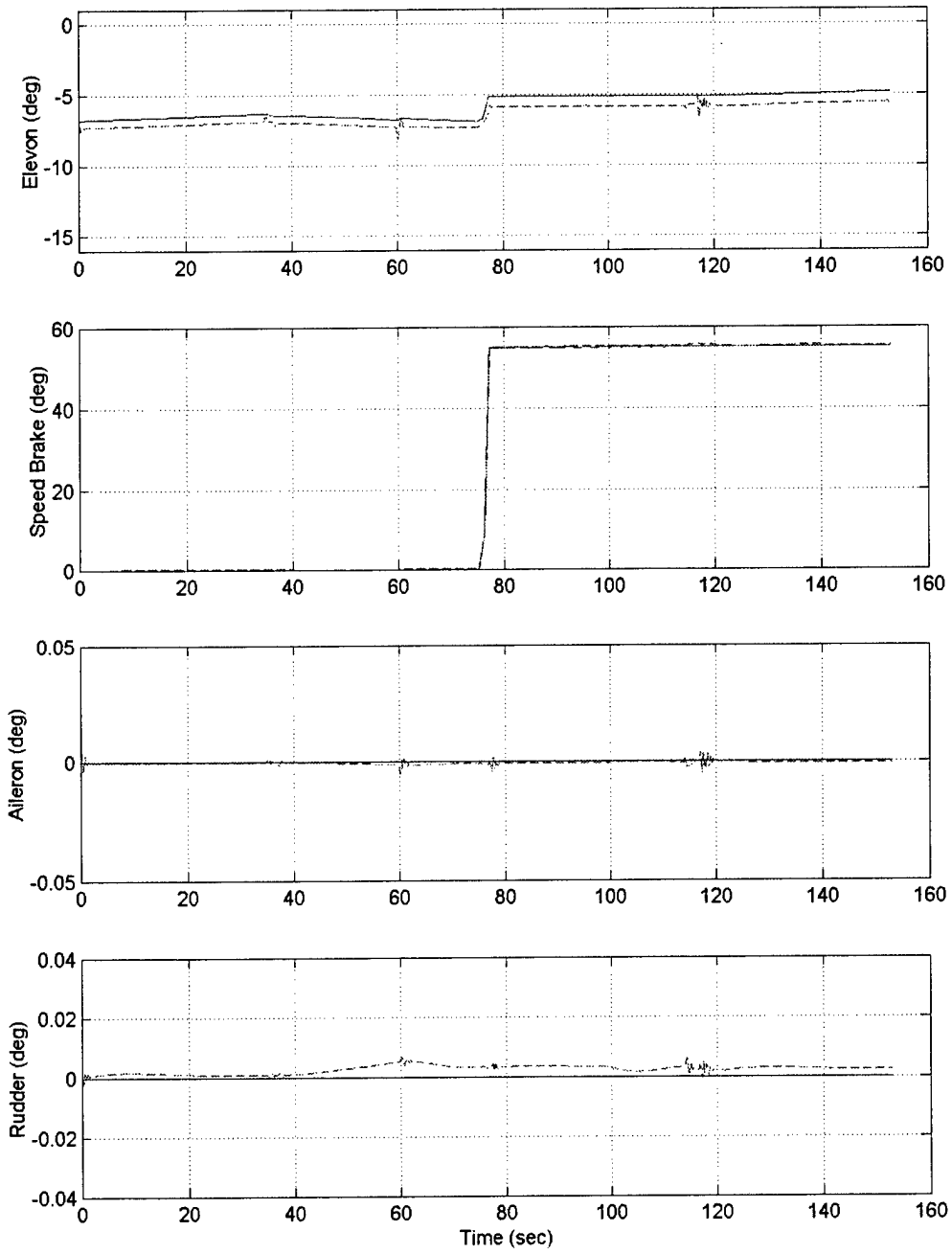

Fig. 97 MPC_ALL 20% Pitching Moment Coefficient Increase (5 of 5)

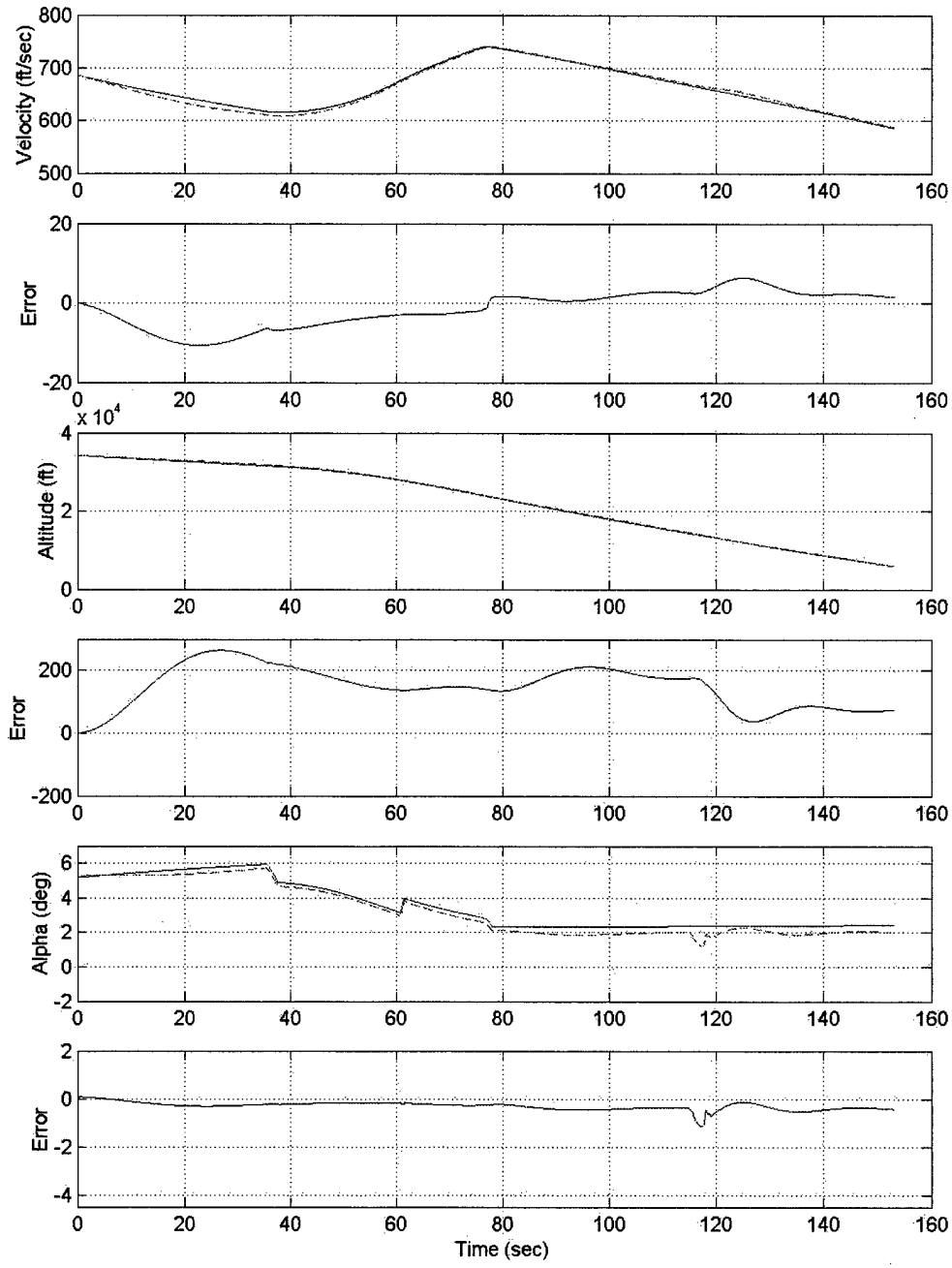

Fig. 98 MPC_ALL 20% Pitching Moment Coefficient Decrease (1 of 5)

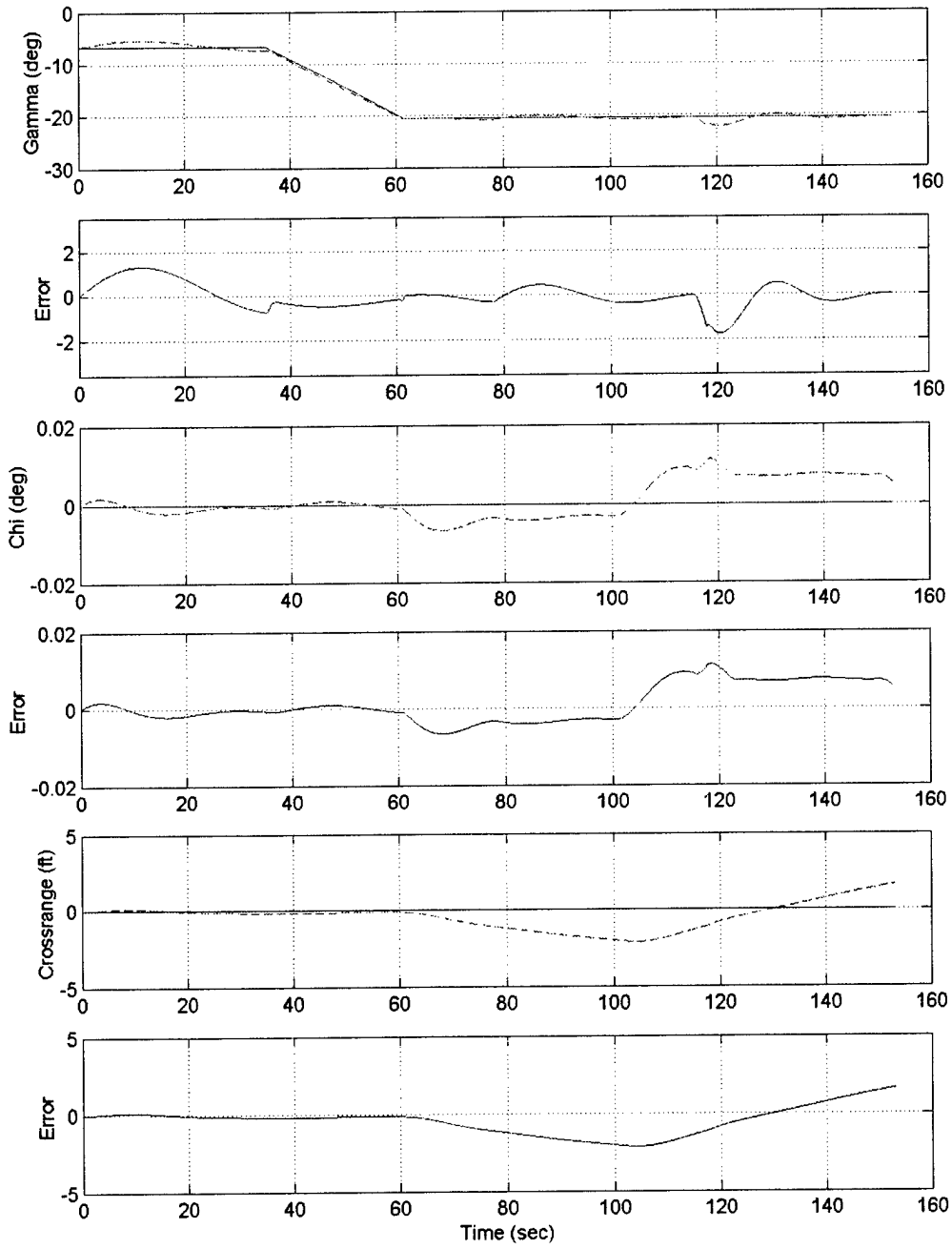

Fig. 99 MPC_ALL 20% Pitching Moment Coefficient Decrease (2 of 5)

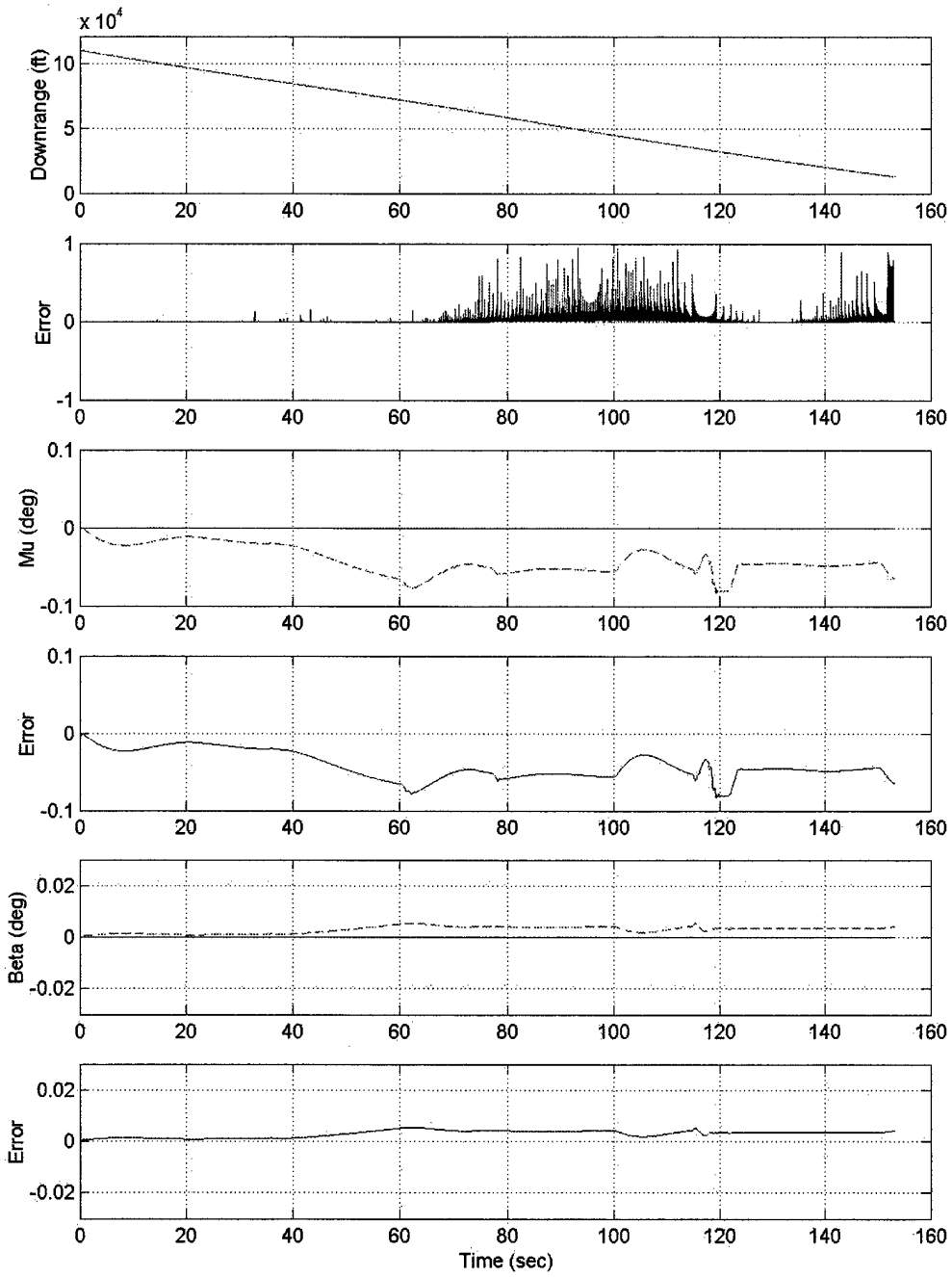

Fig. 100 MPC_ALL 20% Pitching Moment Coefficient Decrease (3 of 5)

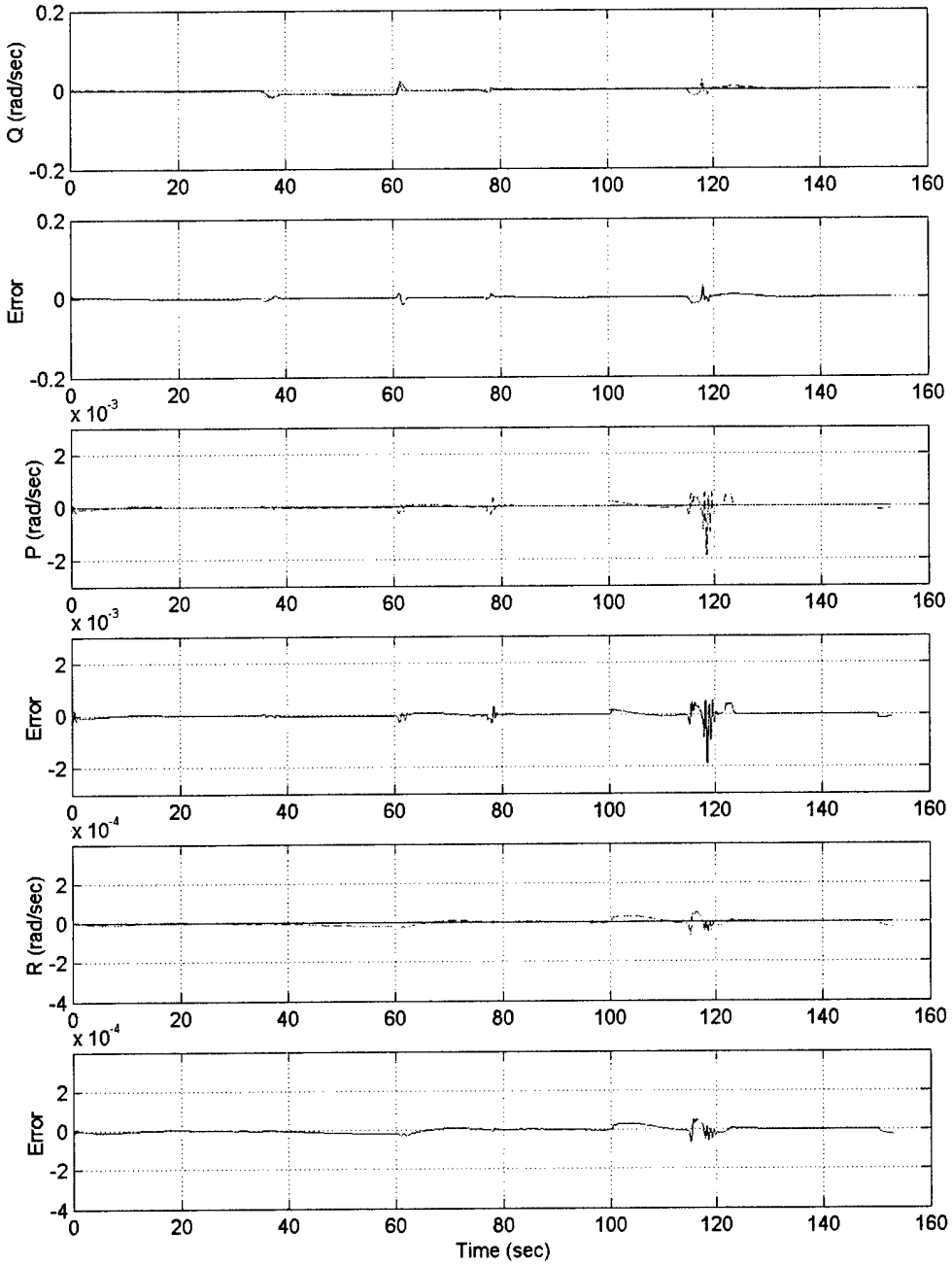

Fig. 101 MPC_ALL 20% Pitching Moment Coefficient Decrease (4 of 5)

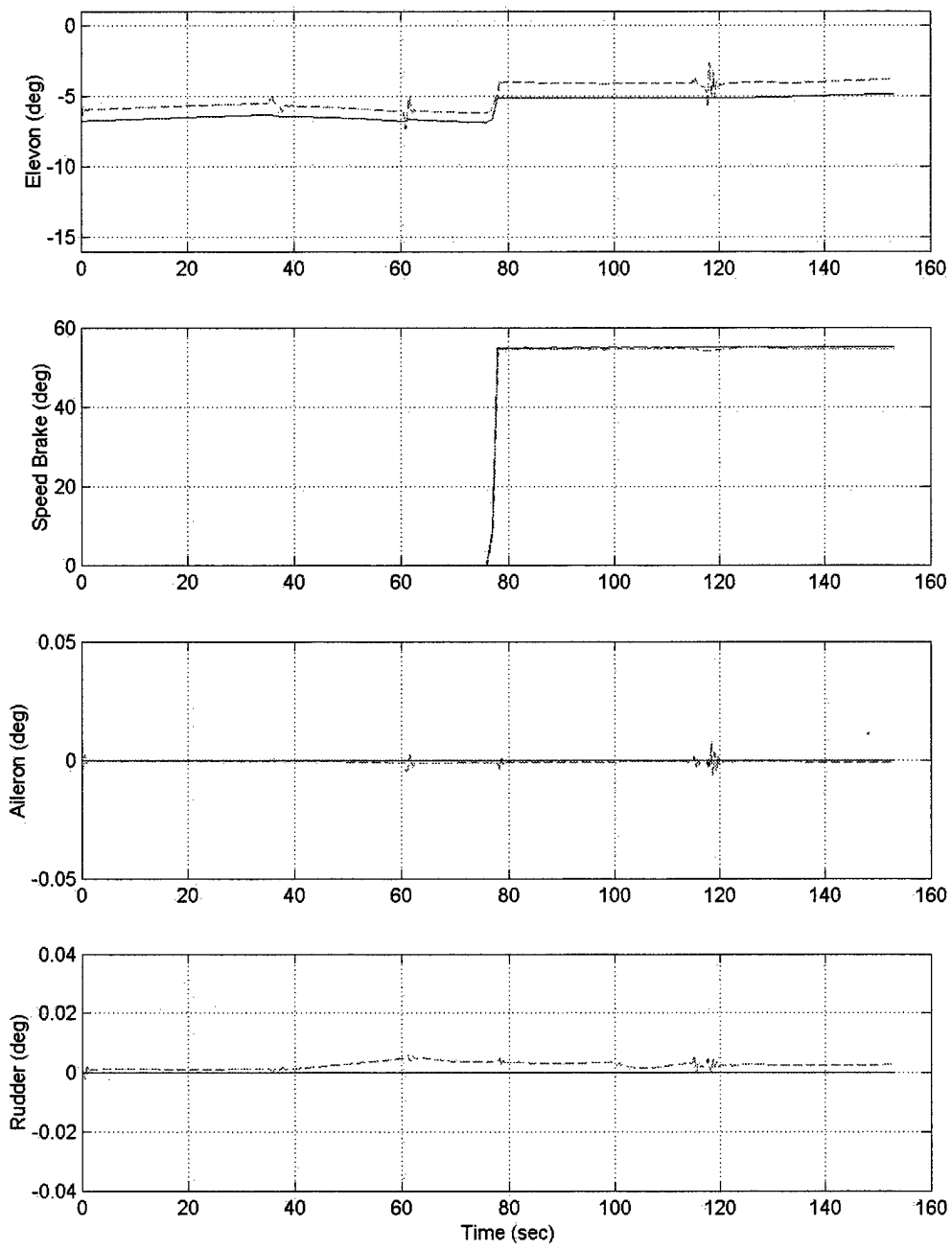

Fig. 102 MPC_ALL 20% Pitching Moment Coefficient Decrease (5 of 5)

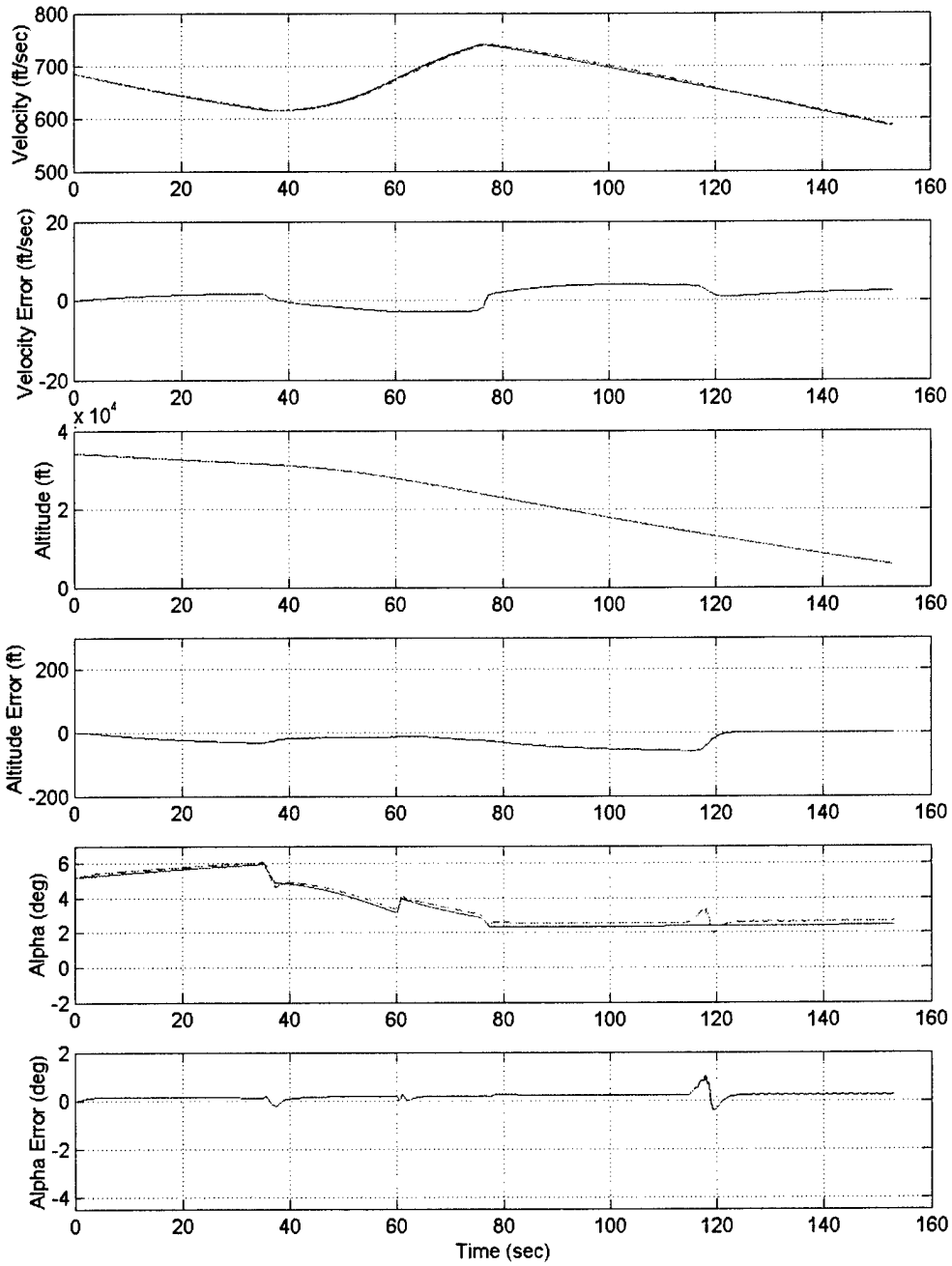

Fig. 103 MPC_SAS 20% Pitching Moment Coefficient Increase (1 of 5)

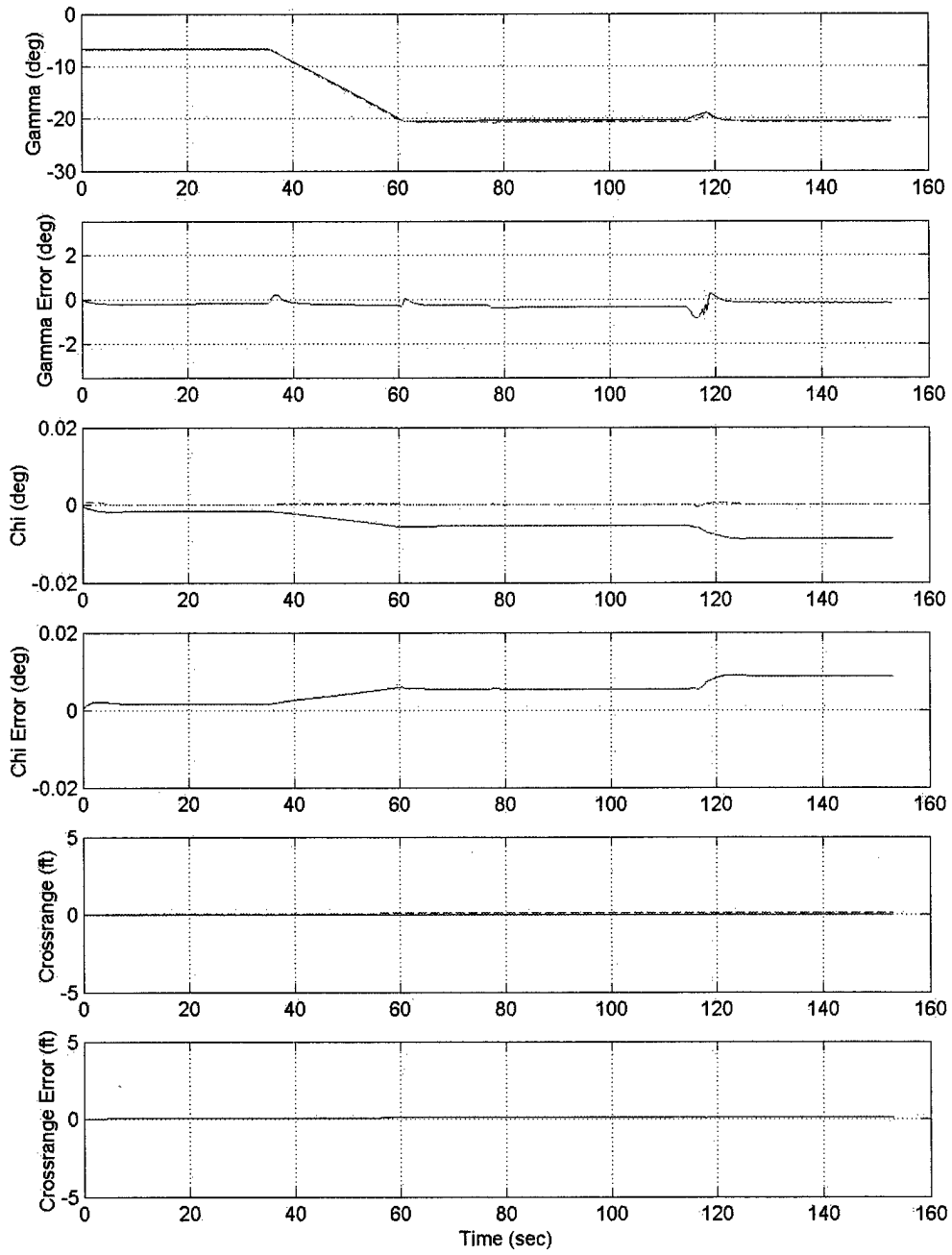

Fig. 104 MPC_SAS 20% Pitching Moment Coefficient Increase (2 of 5)

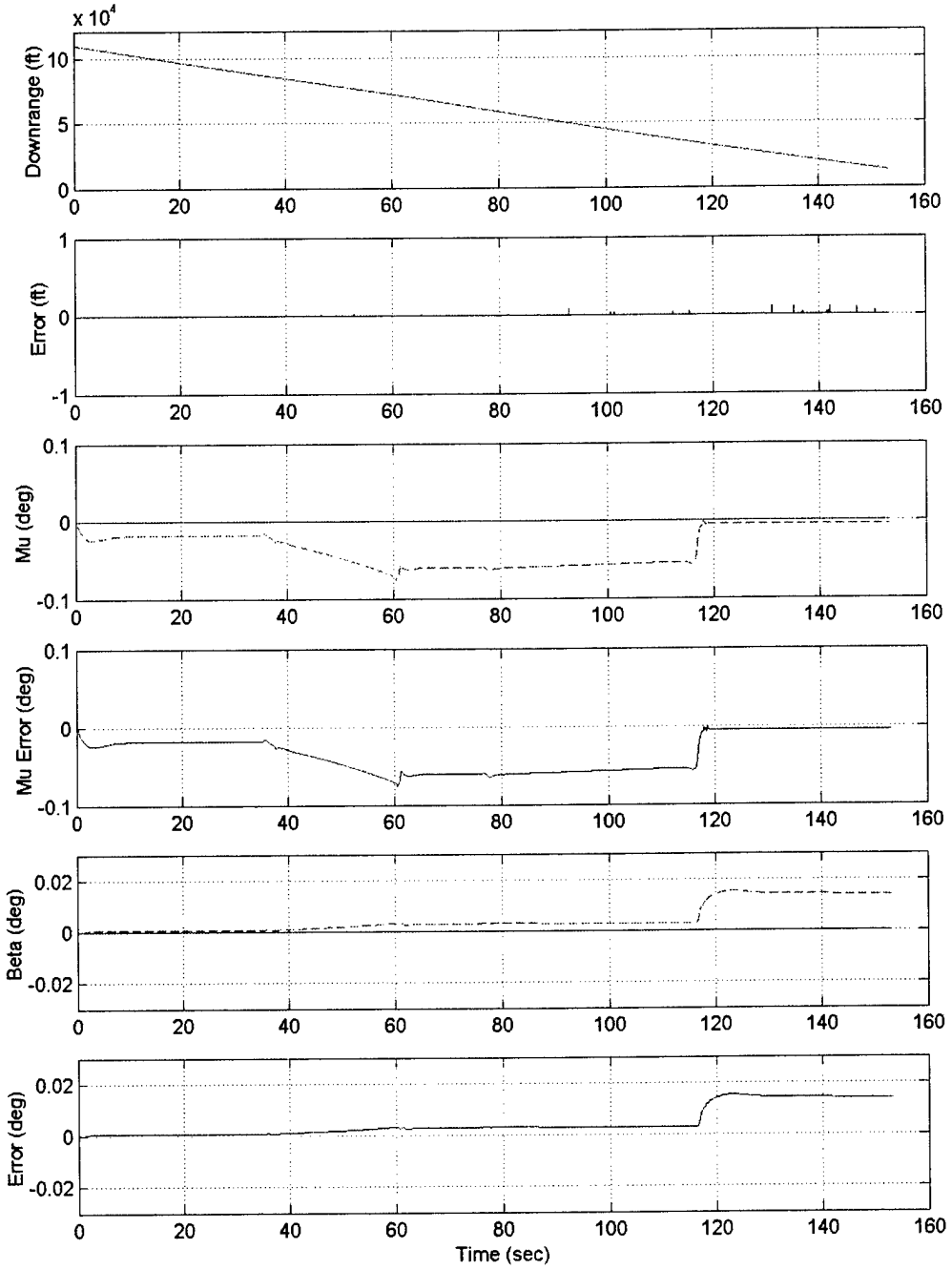

Fig. 105 MPC_SAS 20% Pitching Moment Coefficient Increase (3 of 5)

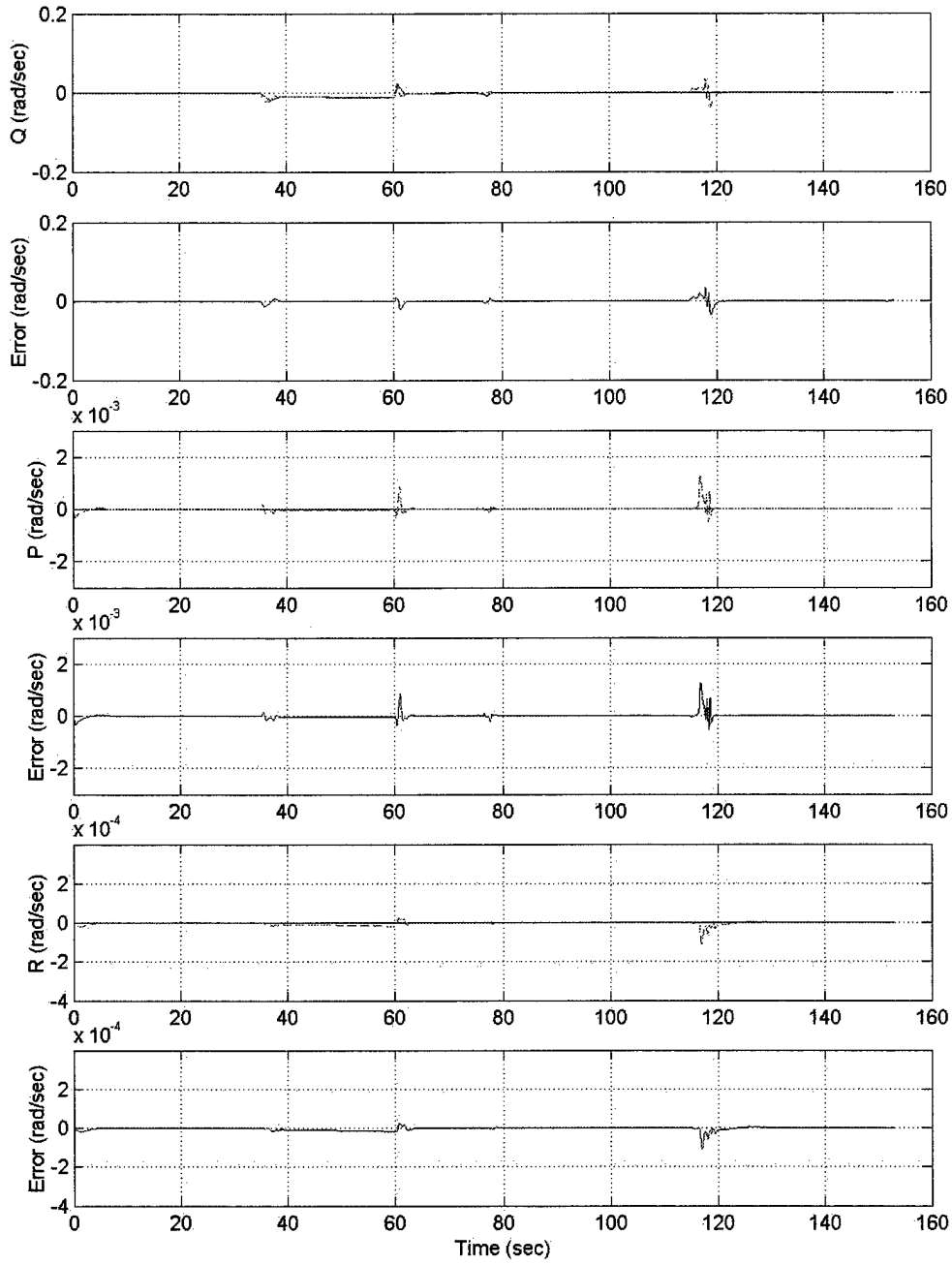

Fig. 106 MPC_SAS 20% Pitching Moment Coefficient Increase (4 of 5)

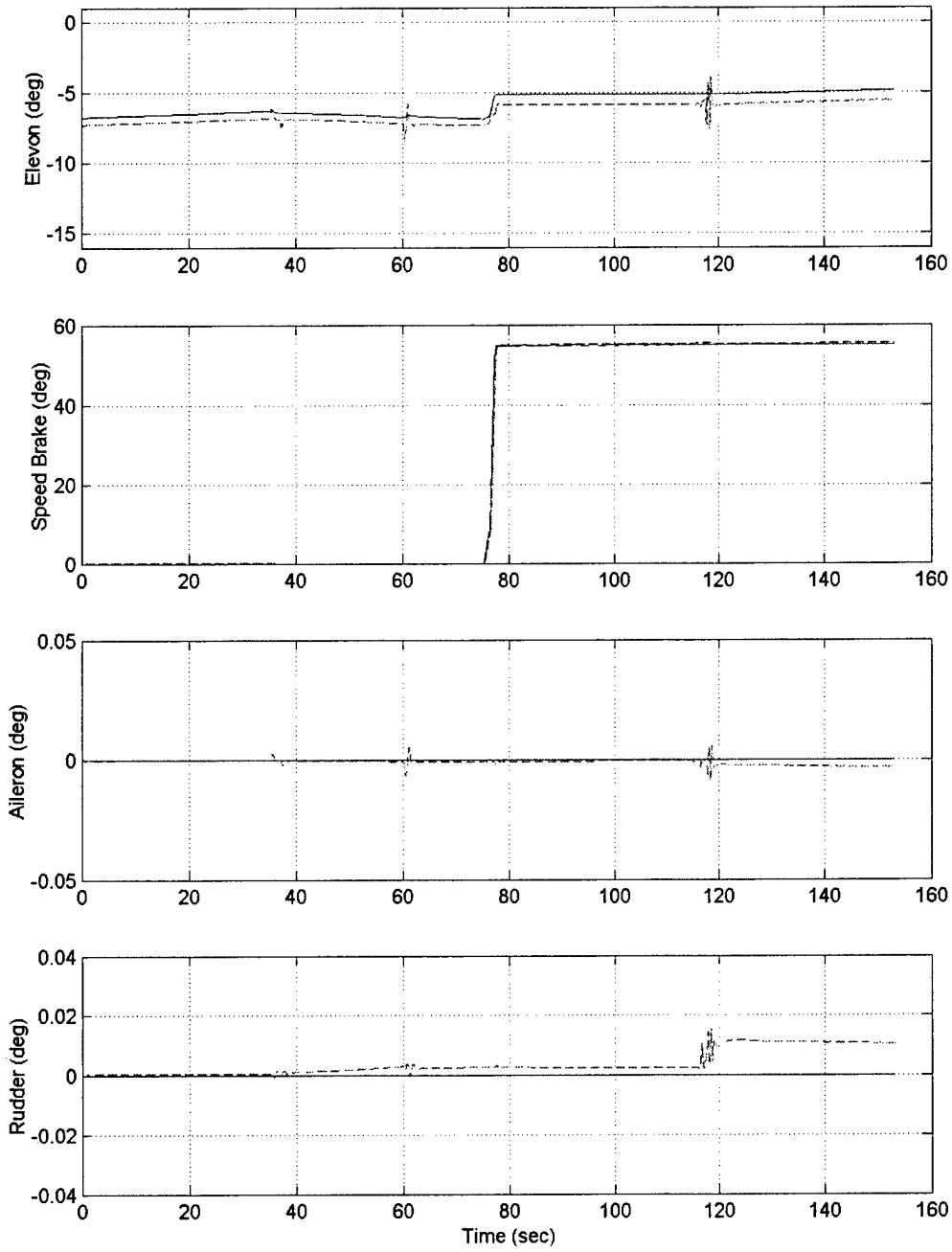

Fig. 107 MPC_SAS 20% Pitching Moment Coefficient Increase (5 of 5)

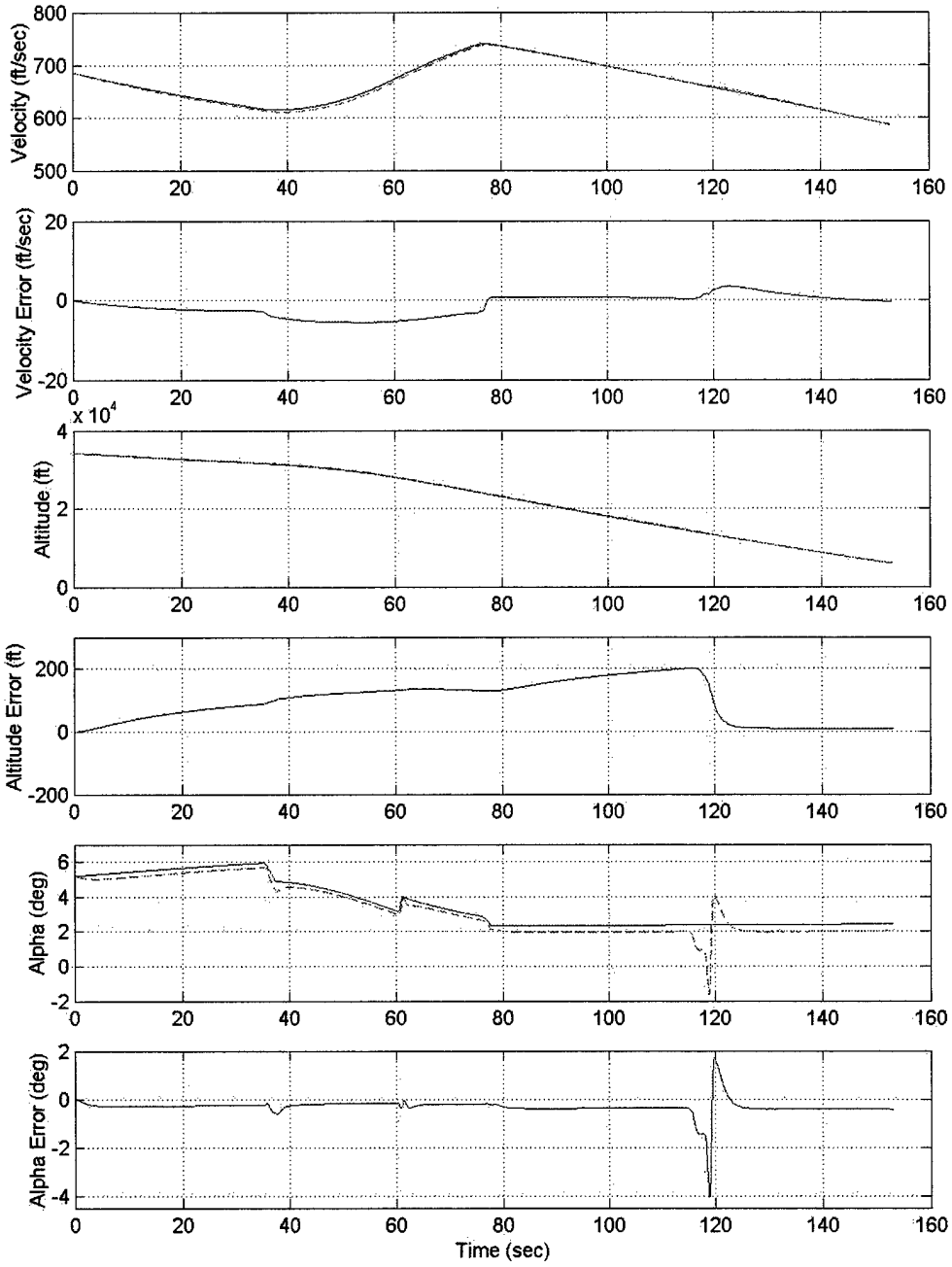

Fig. 108 MPC_SAS 20% Pitching Moment Coefficient Decrease (1 of 5)

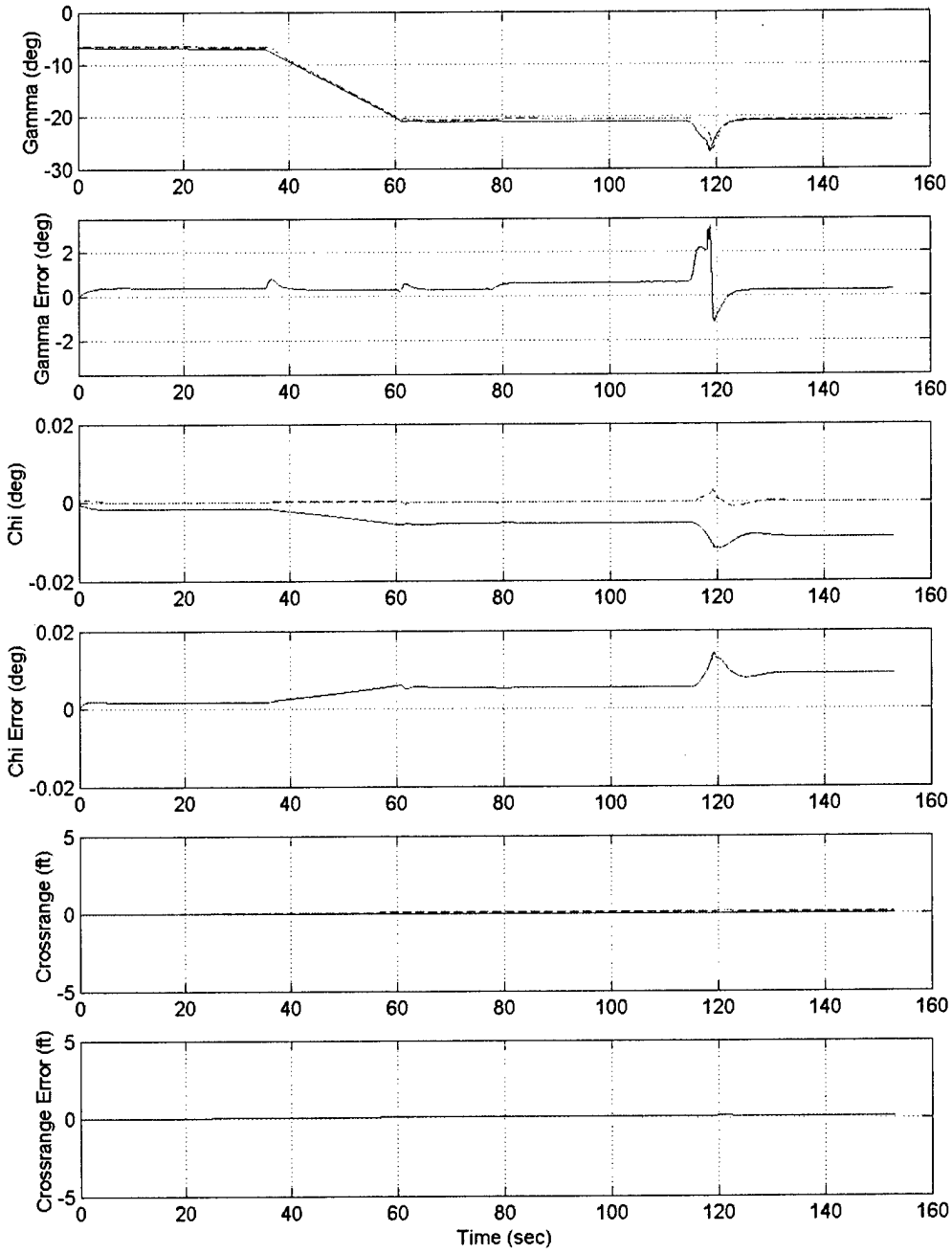

Fig. 109 MPC_SAS 20% Pitching Moment Coefficient Decrease (2 of 5)

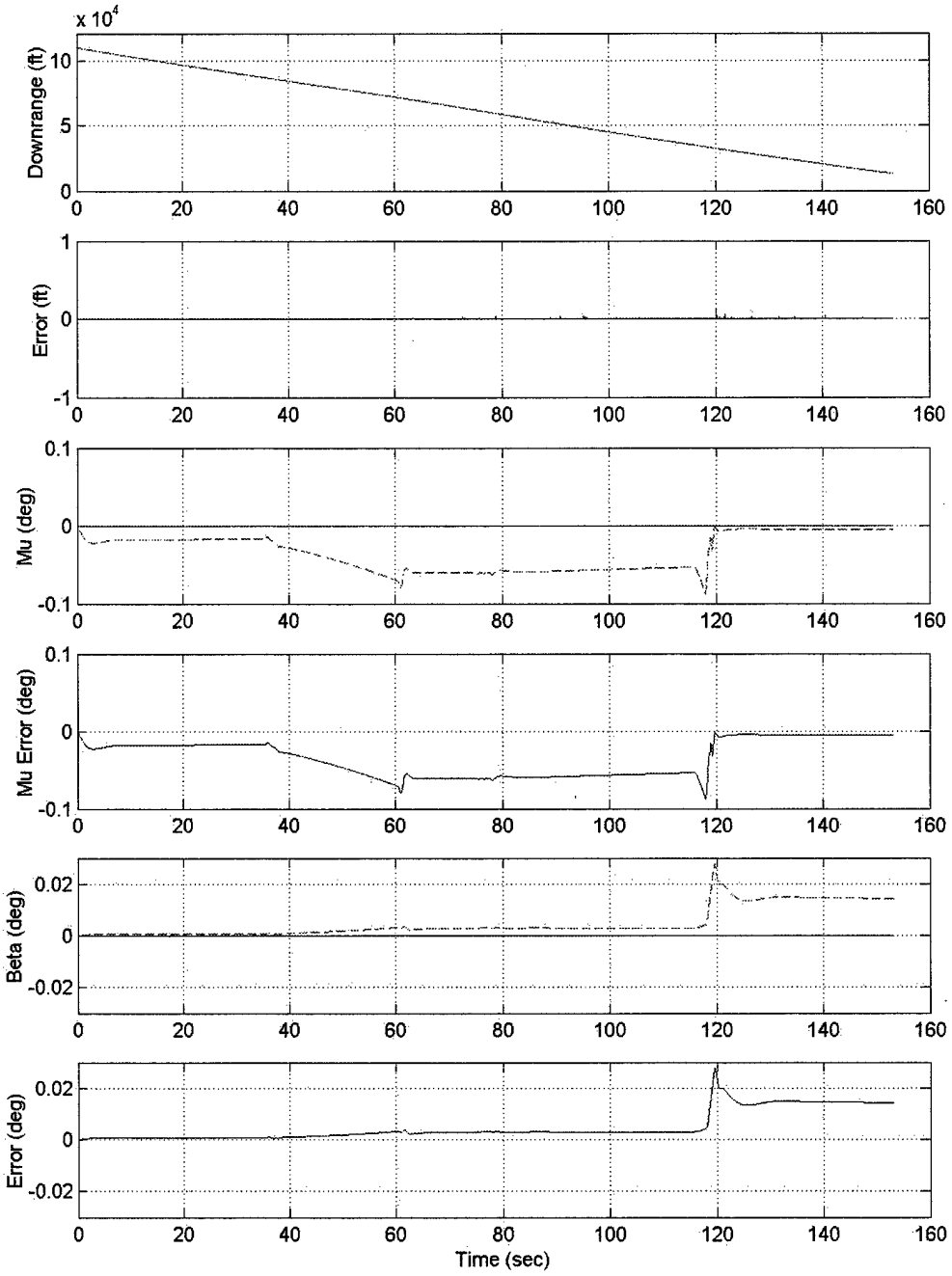

Fig. 110 MPC_SAS 20% Pitching Moment Coefficient Decrease (3 of 5)

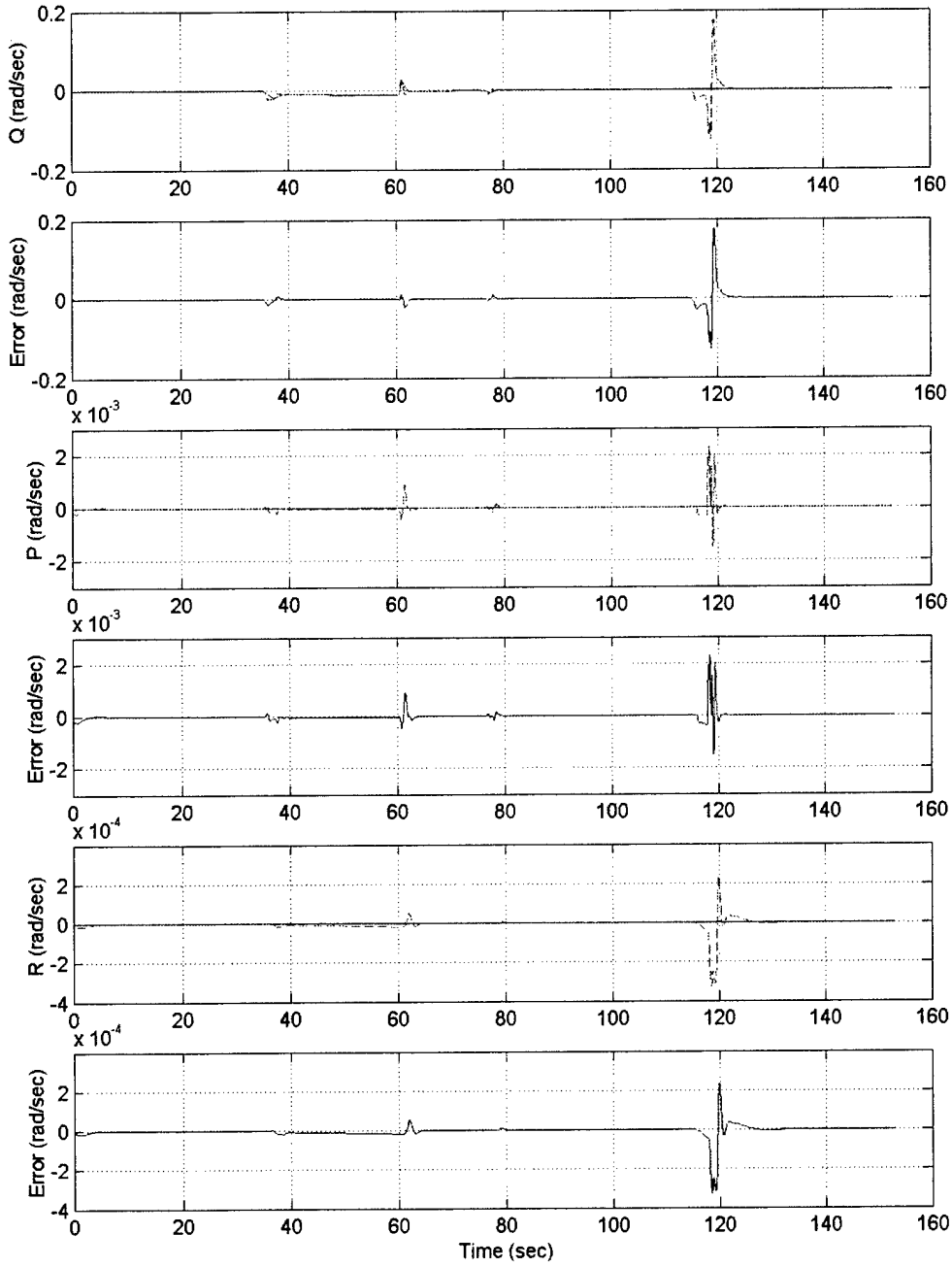

Fig. 111 MPC_SAS 20% Pitching Moment Coefficient Decrease (4 of 5)

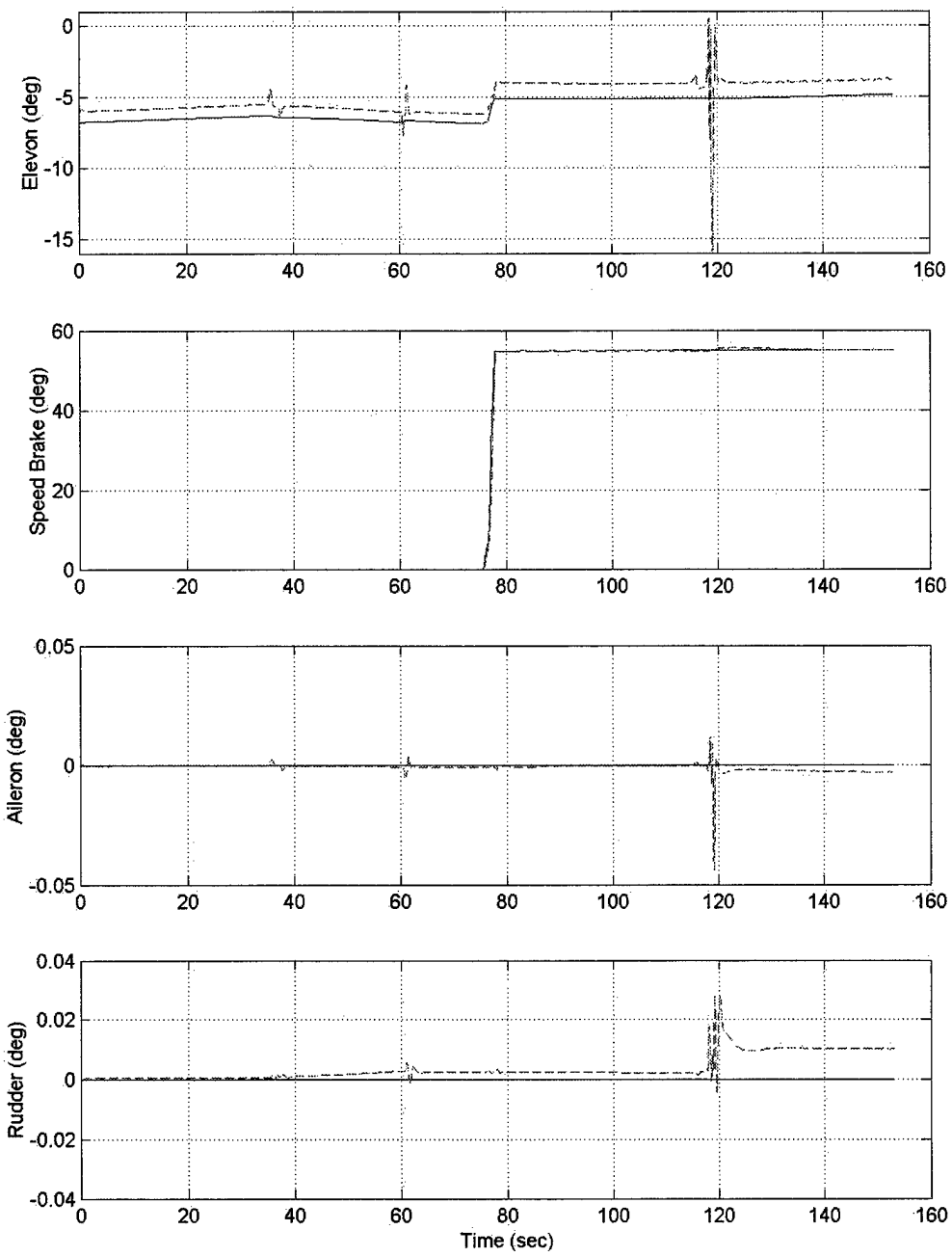

Fig. 112 MPC_SAS 20% Pitching Moment Coefficient Decrease (5 of 5)

[This page intentionally left blank]

Chapter 7

Conclusions

7.1 Conclusions

This research endeavored to meet three primary research objectives. The first and primary objective was to develop a set of design criteria to aid future model predictive control designers in selecting the prediction horizon, simulation rates, weighting matrices, and base functions. The two lesser objectives were to apply the newly found design criteria to the 12 state, nonlinear X-34 technology demonstrator model and to show flexible application of model predictive control to two design architectures.

Procedures and guidelines have been developed to help select each specific design parameter. Populating the weighting matrices proved to be the most challenging parameter to select because of the sensitive nature of the weighting relationships between the states and control. Furthermore, controlling all states with four actuators led to the selection of 20 weights. Pole placement, however, proved to aid significantly in the weighting selection, giving the selection order and structure. In addition to defining design guidelines, it was discovered that a prescribed order of design parameter selection is required. The prediction horizon is selected first, followed by the simulation rates. The loop rates and prediction rate then dictate the weighting selections. Finally, simplifying base functions are used to reduce computational complexity.

Model predictive control was successfully applied to control a nonlinear simulation of the X-34 in the subsonic portion of the terminal area energy management corridor and approach and landing. The application to two different design architectures showed that model predictive control could be used as the system's sole controller to track the desired states, or it could be used to track a subset of states in conjunction with an inner loop stability augmentation system. The flexibility of model predictive control allows it to be applied to a wide range of future aerospace vehicles.

Model predictive control applied to aerospace vehicles provides many potential benefits, which include its anticipative behavior, reducing peak errors, and its constraint handling ability. Model predictive control theory also exhibits disadvantages. It is computationally intensive, and requires an accurate internal model to represent the actual plant. The computation proved to be a substantial burden. A 12 state augmented model creates very large matrices, making simulation runtimes lengthy. The predictive controller currently designed is not refined or tested enough to be flown onboard the X-34. In addition, its computational intensity makes it unfeasible for use as an onboard, real time controller. Base functions prove helpful to meeting the goal of onboard flight, but do not reduce the calculations enough. As faster computers develop, their increased computing speeds make onboard flight a very realistic prospect in the near future.

In all simulated flights, the controller quickly and efficiently eliminated all errors throughout the longitudinal channel. MPC_ALL and MPC_SAS both tracked the longitudinal states very closely. The pole placement approach helped to insure the favorable performance in MPC_ALL. The high inner loop rate of the MPC_SAS coupled with a coordinated effort between the altitude and velocity states in the outer loop and the angle of attack and pitch rate in the inner loop yielded few longitudinal errors. The lack of a pole placement strategy for the lateral channel and the low simulation rate in MPC_ALL showed room for improved lateral tracking. Considering the unstable dynamics of the X-34, a loop rate of 10 Hz for MPC_ALL is a minimum. The loop rate should be increased to improve the lateral performance.

The controllers could be improved to reduce the peak errors obtained and to remove the errors more quickly in all trajectories. However, it must be noted that the errors were eliminated by the end of each flight, including the aggressive trajectories.

7.2 Recommendations for Future Work

Although model predictive control has been studied and applied in industry for 30 years, it remains a new field of study in aerospace applications. There is plenty of room for growth in future research. The recommendations presented here, however, remain within the context of the ongoing research at the Charles Stark Draper Laboratory.

First, specific to controlling the X-34, reevaluation of the weighting matrices is recommended. Benefits of pole placement were seen in the longitudinal channel. It is

expected that the Dutch roll motion will be eliminated and closer tracking in the lateral states will be achieved with a careful pole placement strategy similar to that used for the longitudinal states. The weighting matrices will certainly need to be repopulated should a higher loop rate be chosen, a necessary action to increase system performance.

Second, the current simulation is designed around a trajectory. The long-term vision develops a model in which the simulation is designed for a flyable space instead of trajectory based. A weighting scheme scheduling the states and control weights according to altitude and mach number is suggested for such an open design.

Third, the current internal plant uses a linear approximation of the nonlinear plant at each flight point and linearly interpolates between flight points when necessary. A transition to a full nonlinear internal plant is desirable. An intermediate step using linear parameter varying theory is encouraged to help make the transition. The linear parameter varying strategy effectively schedules the A, B, C, and D matrices describing the linear system according to a chosen parameter such as dynamic pressure or altitude. Linear parameter varying techniques provide for increased flexibility.

Finally, this research only introduced the use of constraints to improve performance. Utilizing the constraint handling capability of model predictive control should be further explored and understood. Additional simulations where the controller is unaware of a fixed aerosurface should be analyzed. These additional simulations would give greater insight to how predictive control may be used to further research in reconfigurable control.

[This page intentionally left blank]

Appendix

The results chapter showed the performance of MPC_ALL and MPC_SAS flying the "straight" and "single bank" trajectories. As stated before, those two trajectories are representative of benign trajectories. The maneuvers are not especially demanding and do not fully challenge the capability of MPC. Multiple trajectories are required to completely evaluate the MPC design. The following three aggressive trajectories are examples of more difficult flights. They each represent trajectories at the edge of the flyable space. Common elements between the aggressive trajectories are that they each initially start with the vehicle heading to the designated tangent point on the HAC and all of the trajectories start at an altitude of approximately 30,000 feet. The first of the aggressive trajectories is a "long" trajectory as shown in Fig. 113. It initially starts aligned with the runway centerline and does not have any banking maneuvers. This trajectory qualifies as aggressive because it starts the vehicle very far from the runway. This simulates a flight where the vehicle is low on energy for the distance it must travel to arrive at the runway. The vehicle assumes an α profile giving the vehicle a maximum glide characteristic allowing it to fly as far as possible. When the vehicle arrives at the nominal point before the A/L section, it begins an increased dive as in the "straight" trajectory.

The second aggressive trajectory is "short" (Fig. 114). The vehicle is again initially aligned with the runway. In this situation, however, the vehicle has too much energy and is too close to the runway. It must perform a steep dive to correct for the altitude and open the speed brake to bleed off the excess energy.

The final trajectory referred to as "high crossrange". Shown in Fig. 115, it is a trajectory that starts the vehicle far from the runway and has a very large initial crossrange value. The X-34 is offset so far from the runway that it must fly almost perpendicular to the runway centerline. The vehicle subsequently has an extended banking phase.

The simulations run the three trajectories under the same conditions as flown for the "straight" and "single bank" trajectories in 6.1. The full nonlinear actual plant is used. The internal plant is an LTI approximation of the nonlinear plant, and the simulation flies the unconstrained solution using the selected design parameters in Tab. 13 and the weightings from Tab. 8, Tab. 9, and Tab. 10.

The MPC_ALL architecture is flown first, followed by the MPC_SAS. Because these trajectories represent aggressive flights, large peak errors are expected and observed throughout the flight, but all of the flights drive the errors towards 0. The errors initially start at 0 because the guidance system has purposefully generated a trajectory starting at the vehicle's initial position. These aggressive trajectories help demonstrate the boundaries of the flyable space. They also can be used to simulate an abort scenario where the guidance subsystem has generated a new trajectory on the fly to guide the vehicle safely to a runway. Such an onboard trajectory generation is one of Draper's goals for next generation guidance and control. The "long" trajectory transitions weighting matrices at 209.8 seconds. The "short" trajectory transitions its weighting matrices at 70.5. The "high crossrange" trajectory switches weighting matrices from the first wings level phase to the banking phase at 82.6 seconds. It switches to the second wings level phase at 162.3 seconds and makes its final switch to the approach and landing phase at 178.9 seconds.

The results from MPC_SAS flying the "high crossrange" trajectory are not shown. The simulation is sensitive to large excursions from the reference setpoints, and the vehicle experienced large errors that were not recoverable. As such, the simulation failed when attempting to fly this trajectory.

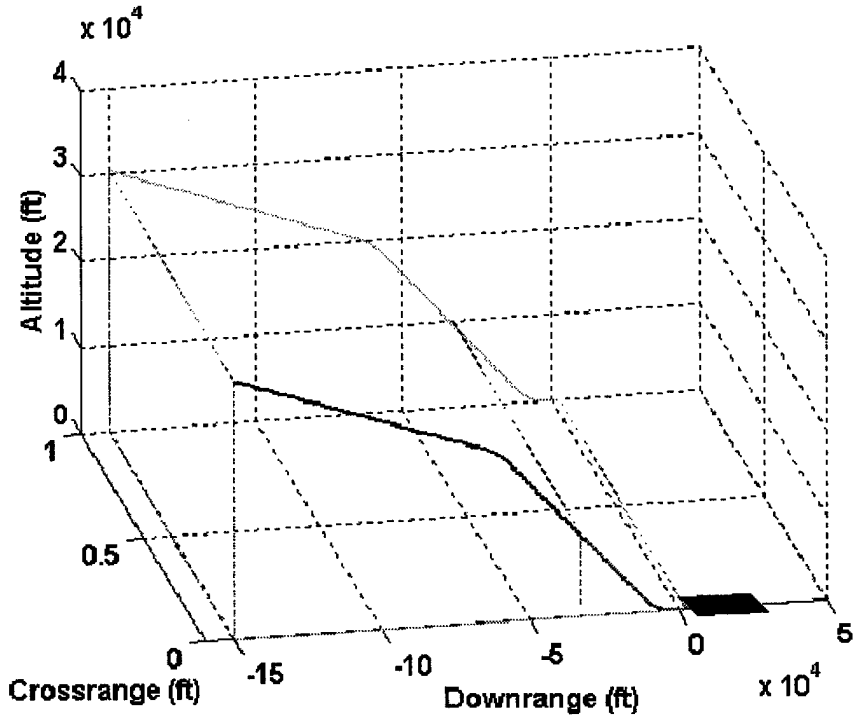

Fig. 113 Long Range Trajectory

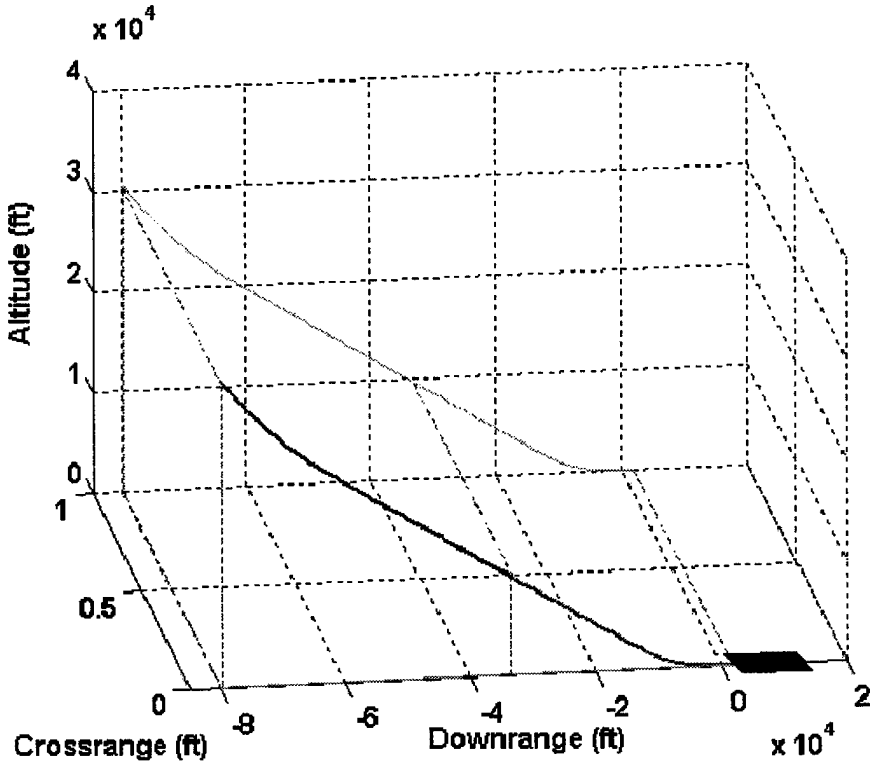

Fig. 114 Short Range Trajectory

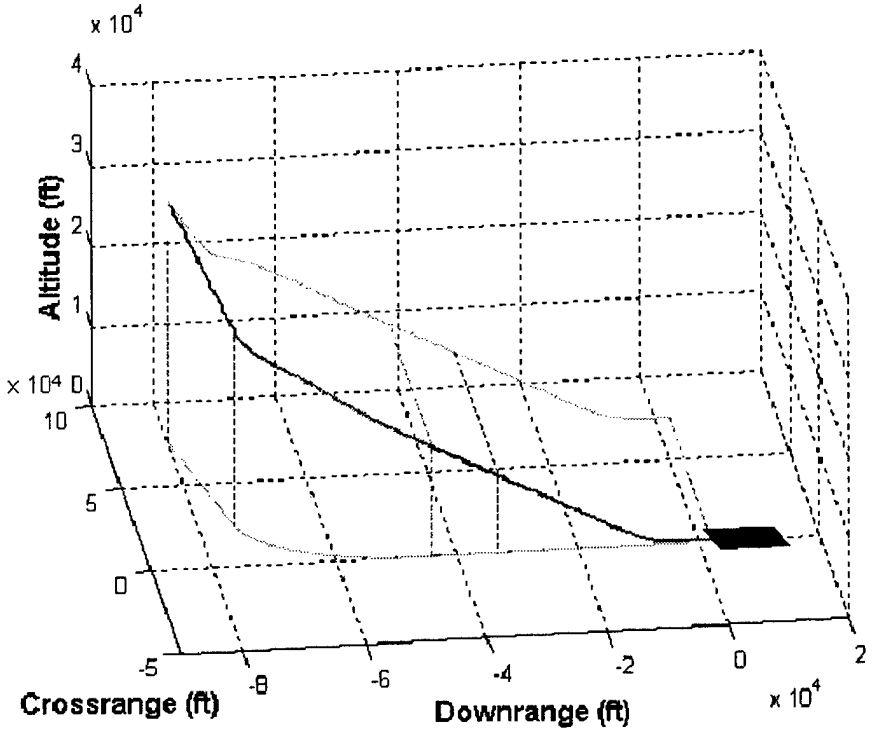

Fig. 115 High Crossrange Trajectory

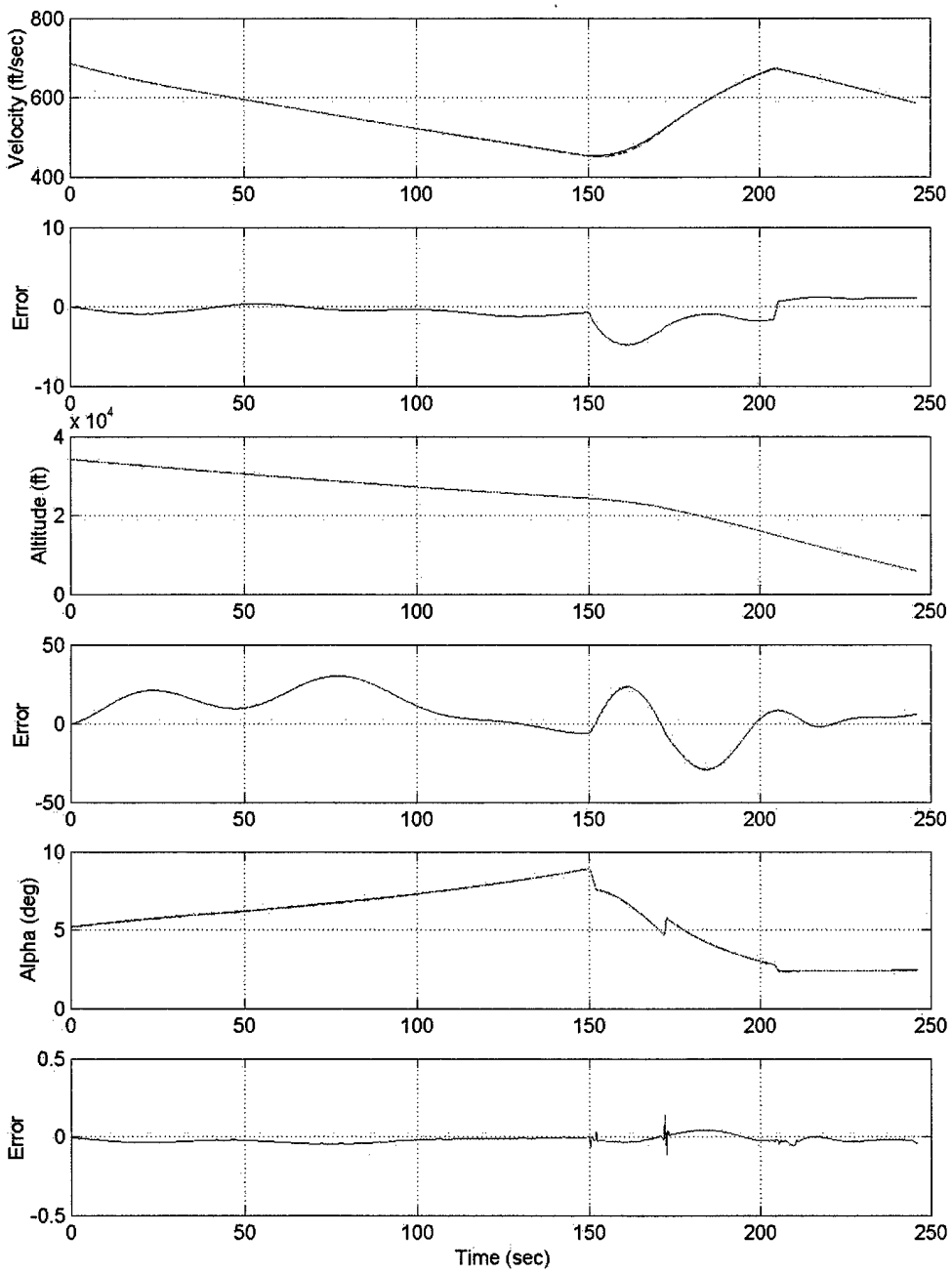

Fig. 116 MPC_ALL Long Range (1 of 5)

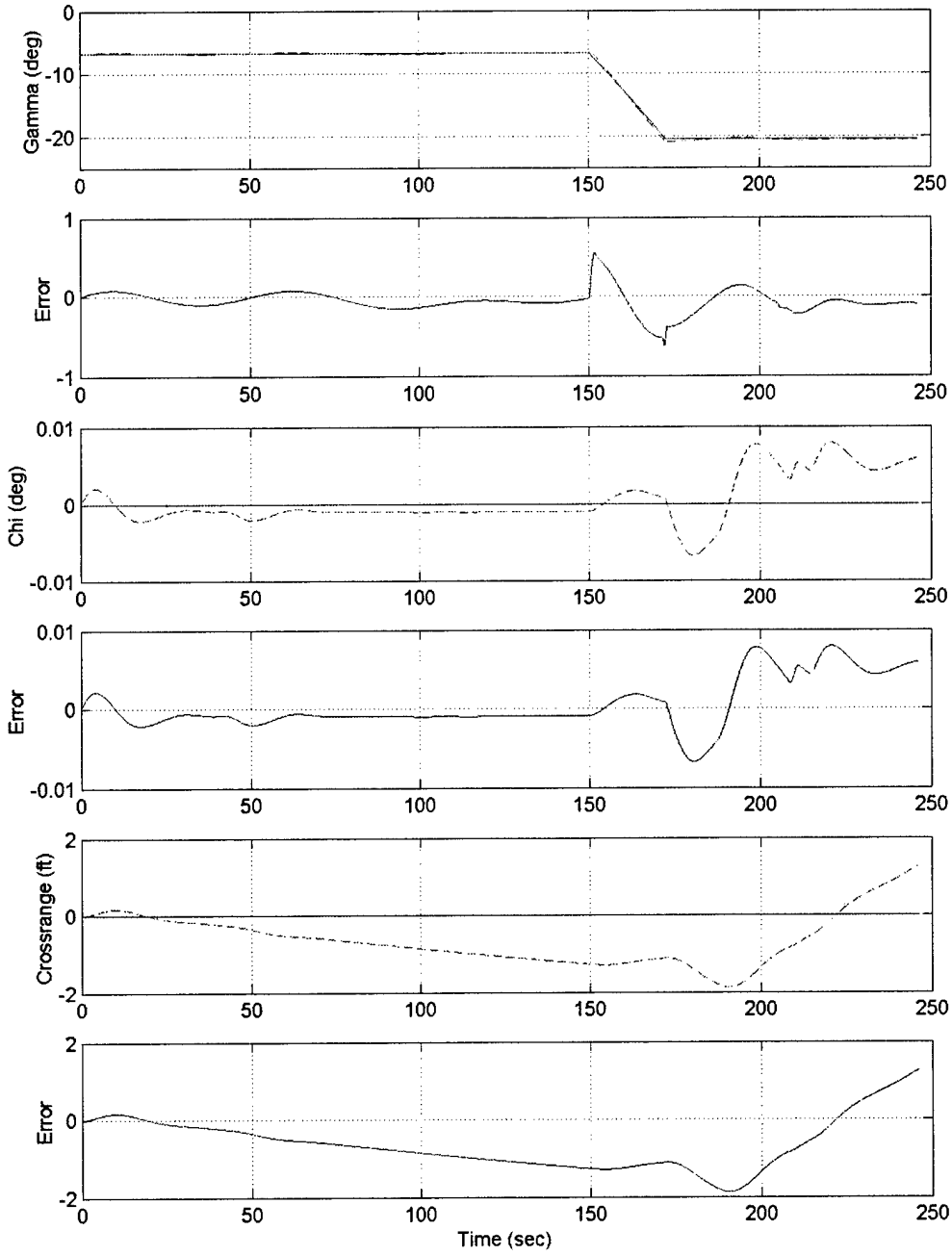

Fig. 117 MPC_ALL Long Range (2 of 5)

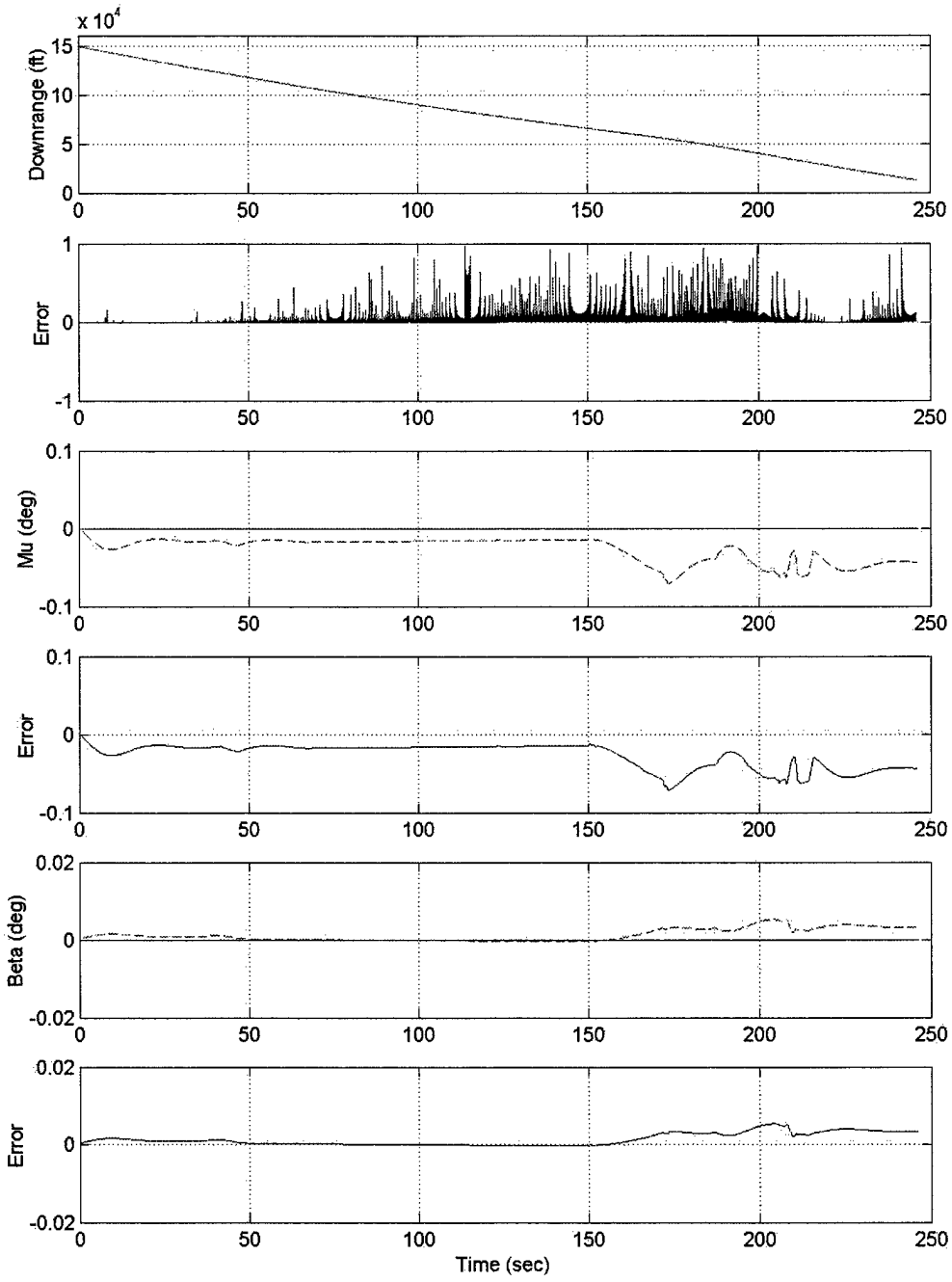

Fig. 118 MPC_ALL Long Range (3 of 5)

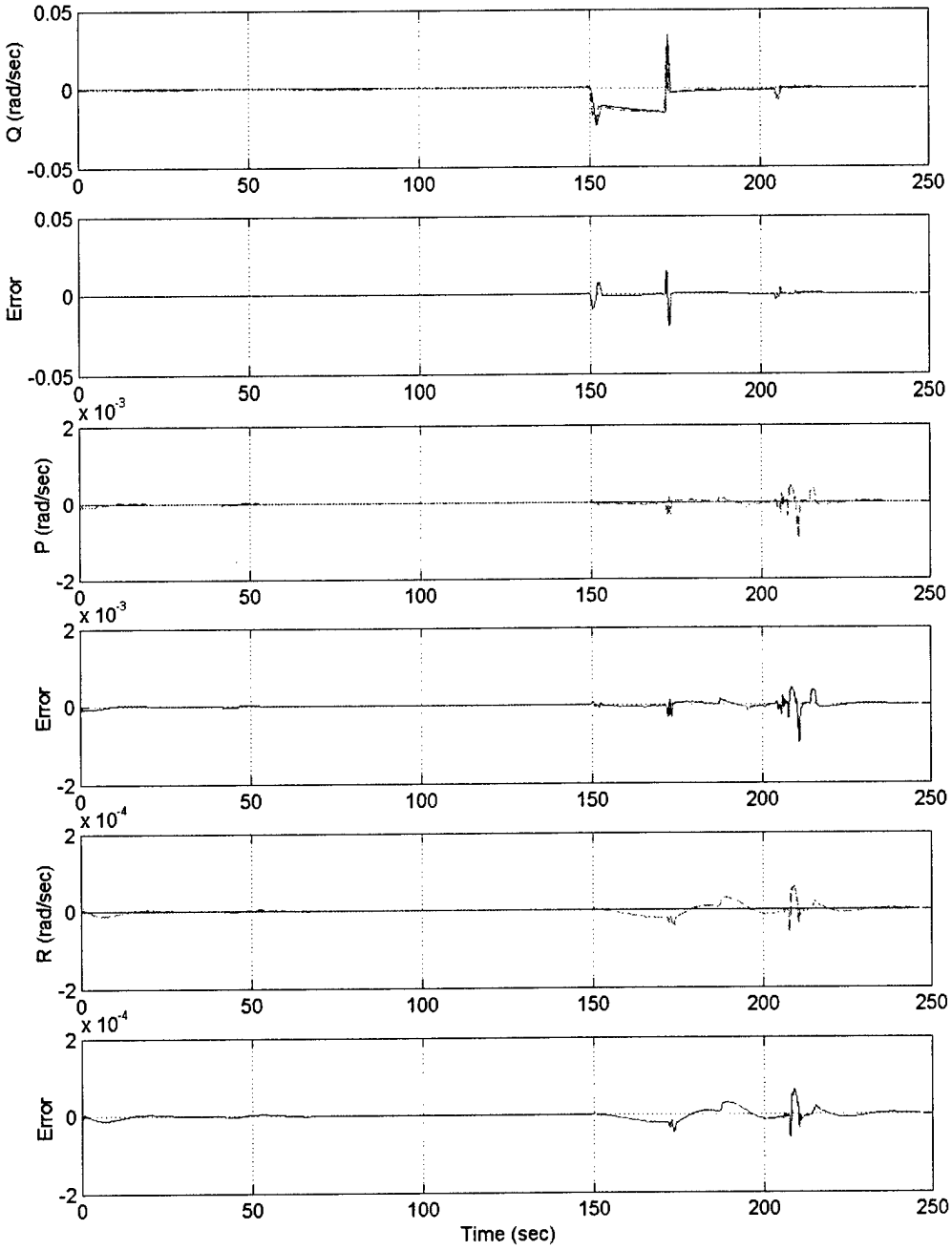

Fig. 119 MPC_ALL Long Range (4 of 5)

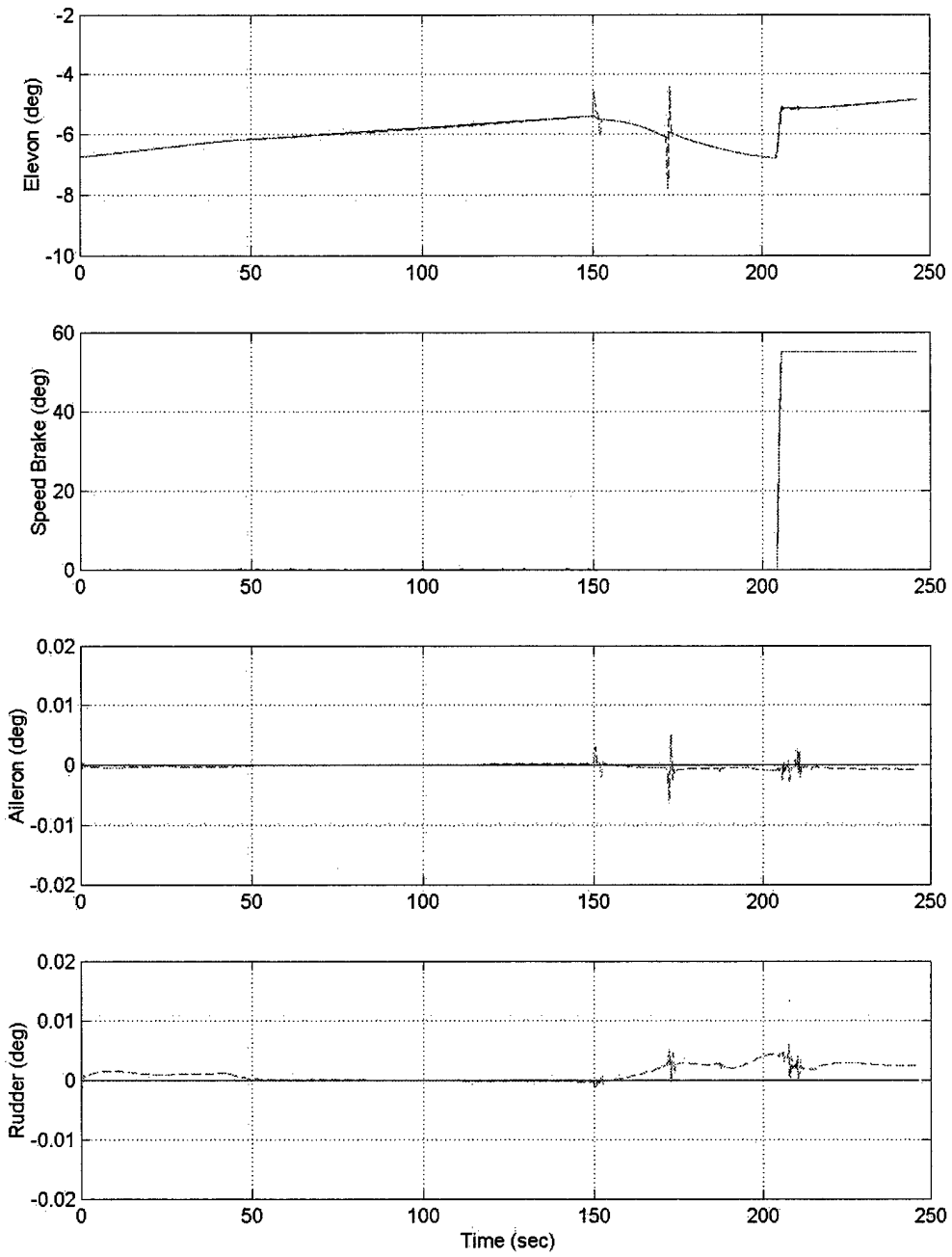

Fig. 120 MPC_ALL Long Range (5 of 5)

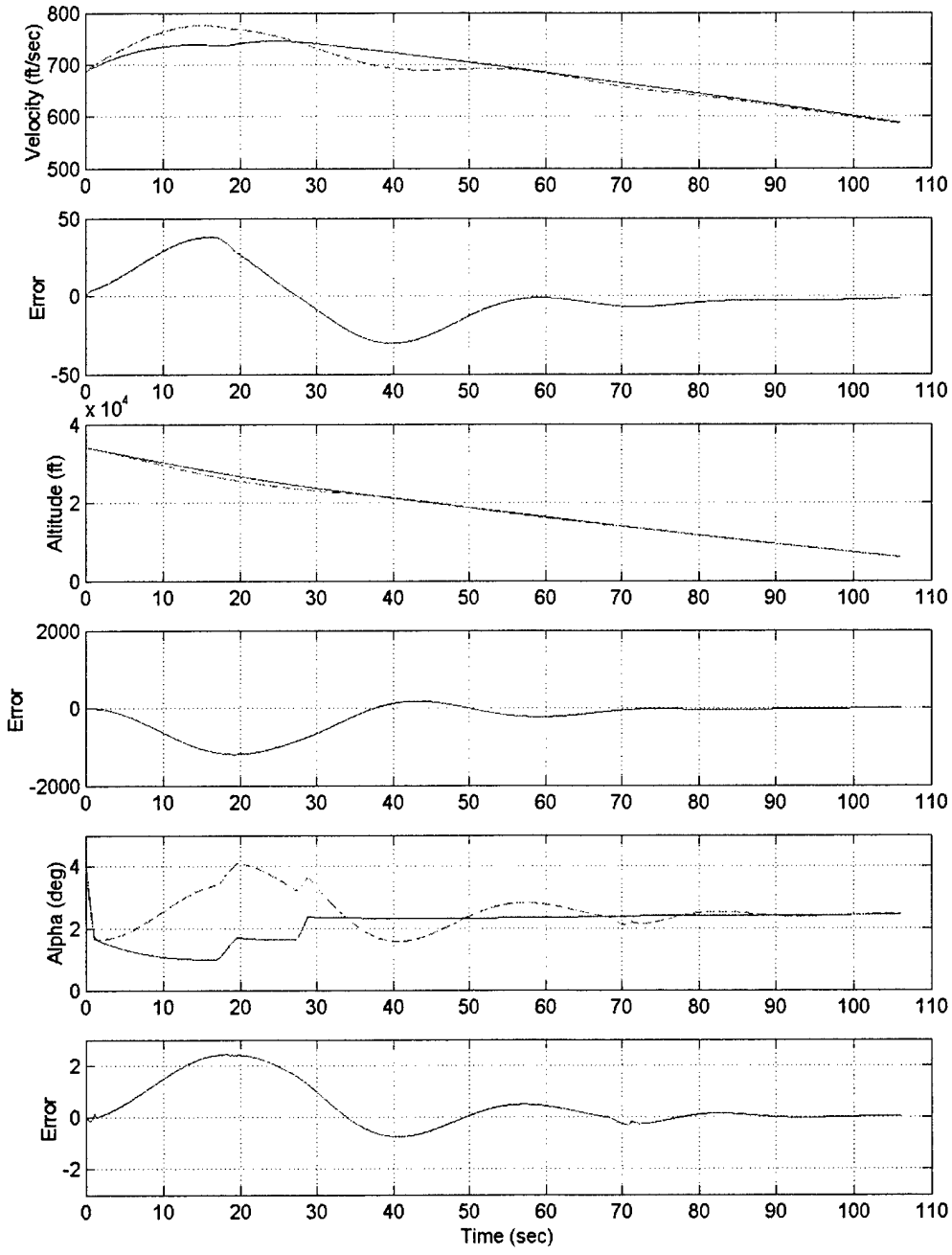

Fig. 121 MPC_ALL Short Range (1 of 5)

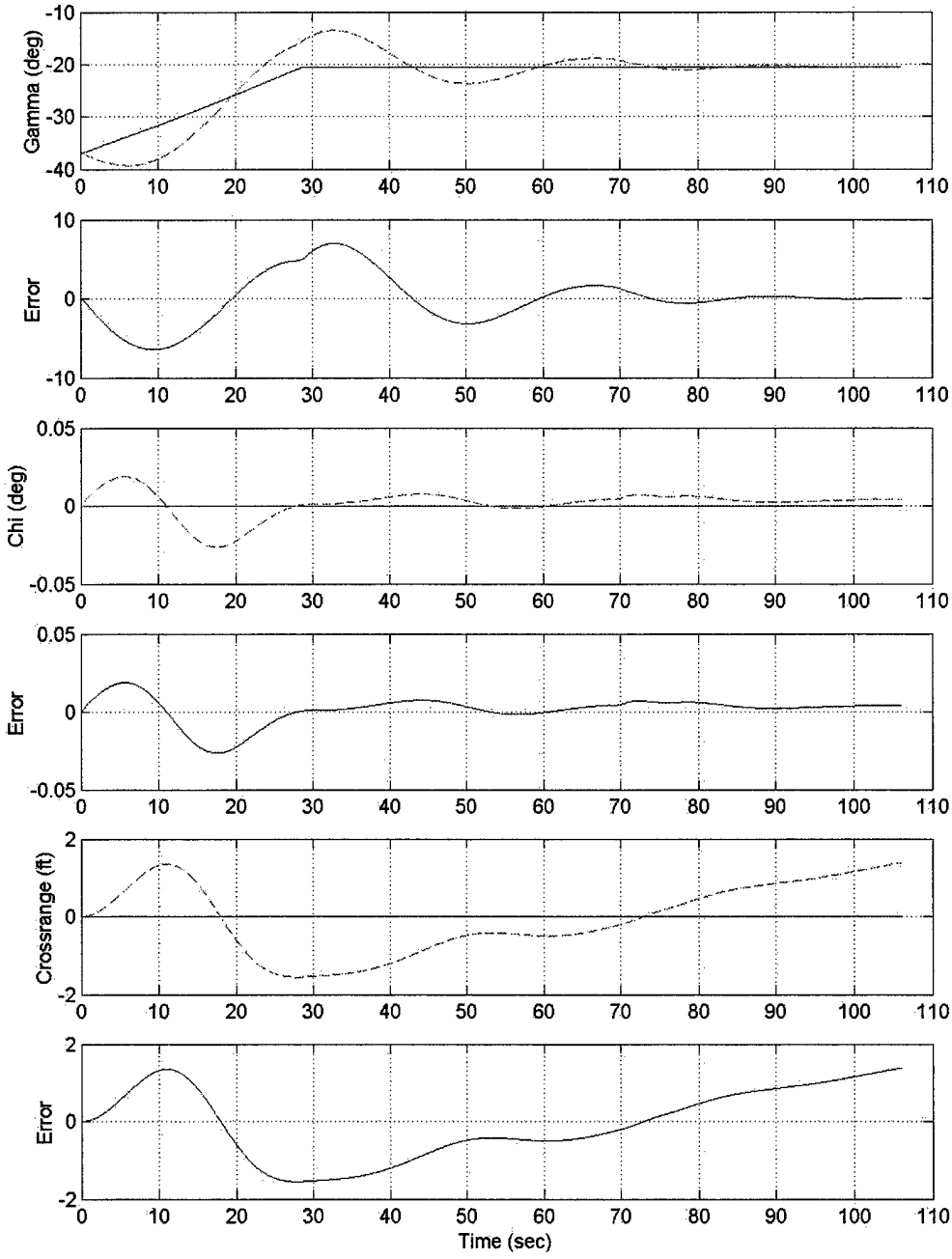

Fig. 122 MPC_ALL Short Range (2 of 5)

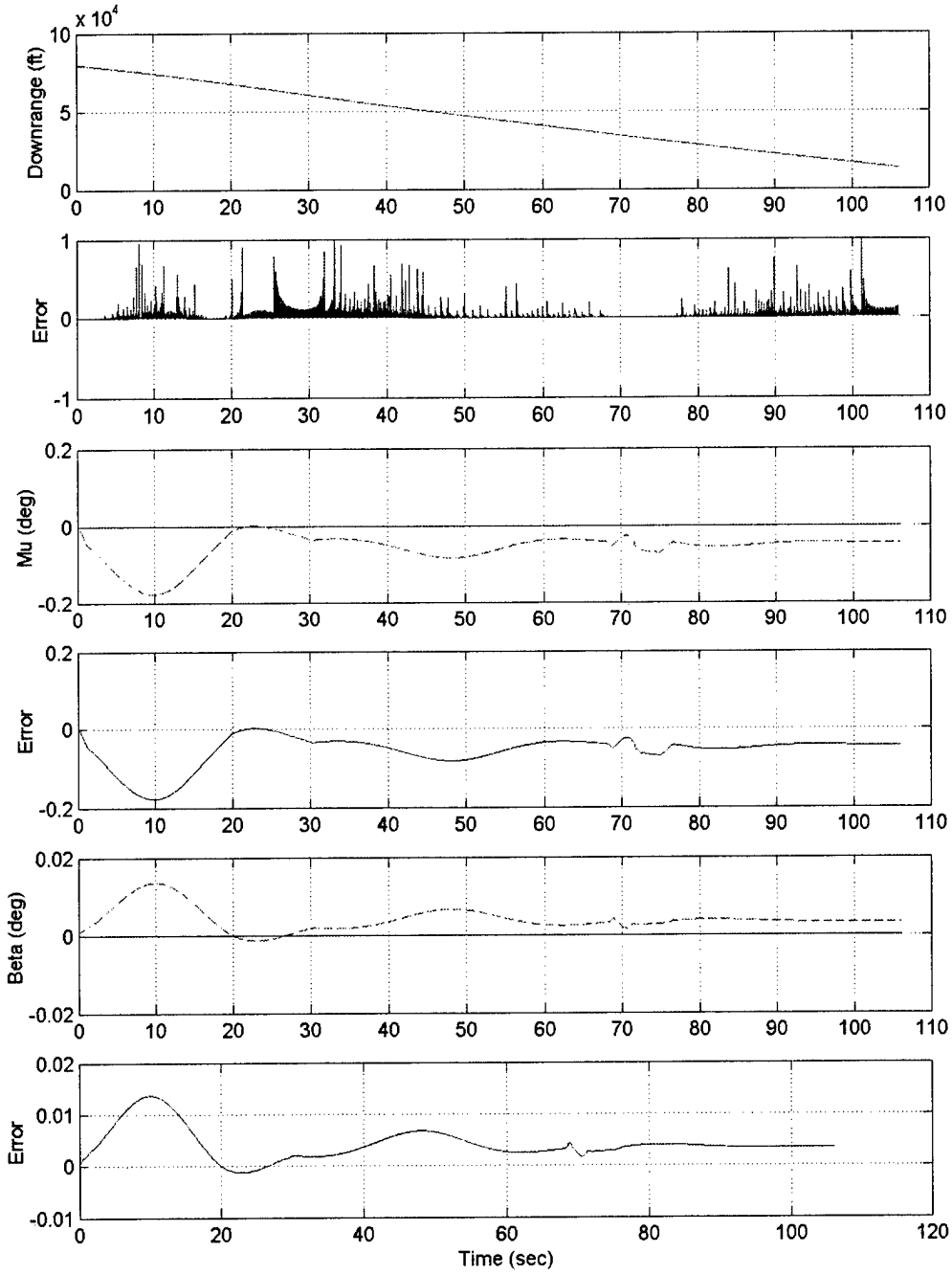

Fig. 123 MPC_ALL Short Range (3 of 5)

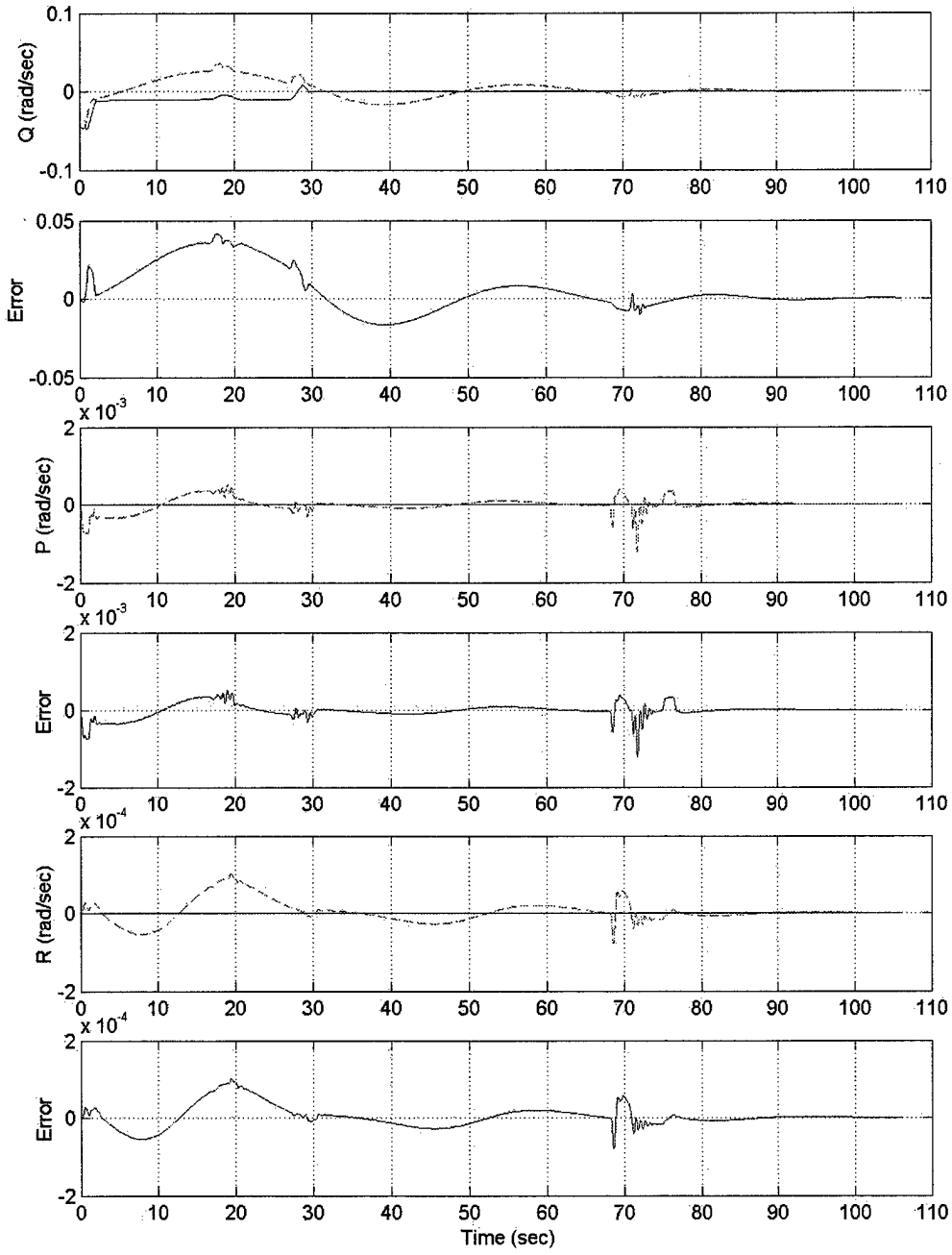

Fig. 124 MPC_ALL Short Range (4 of 5)

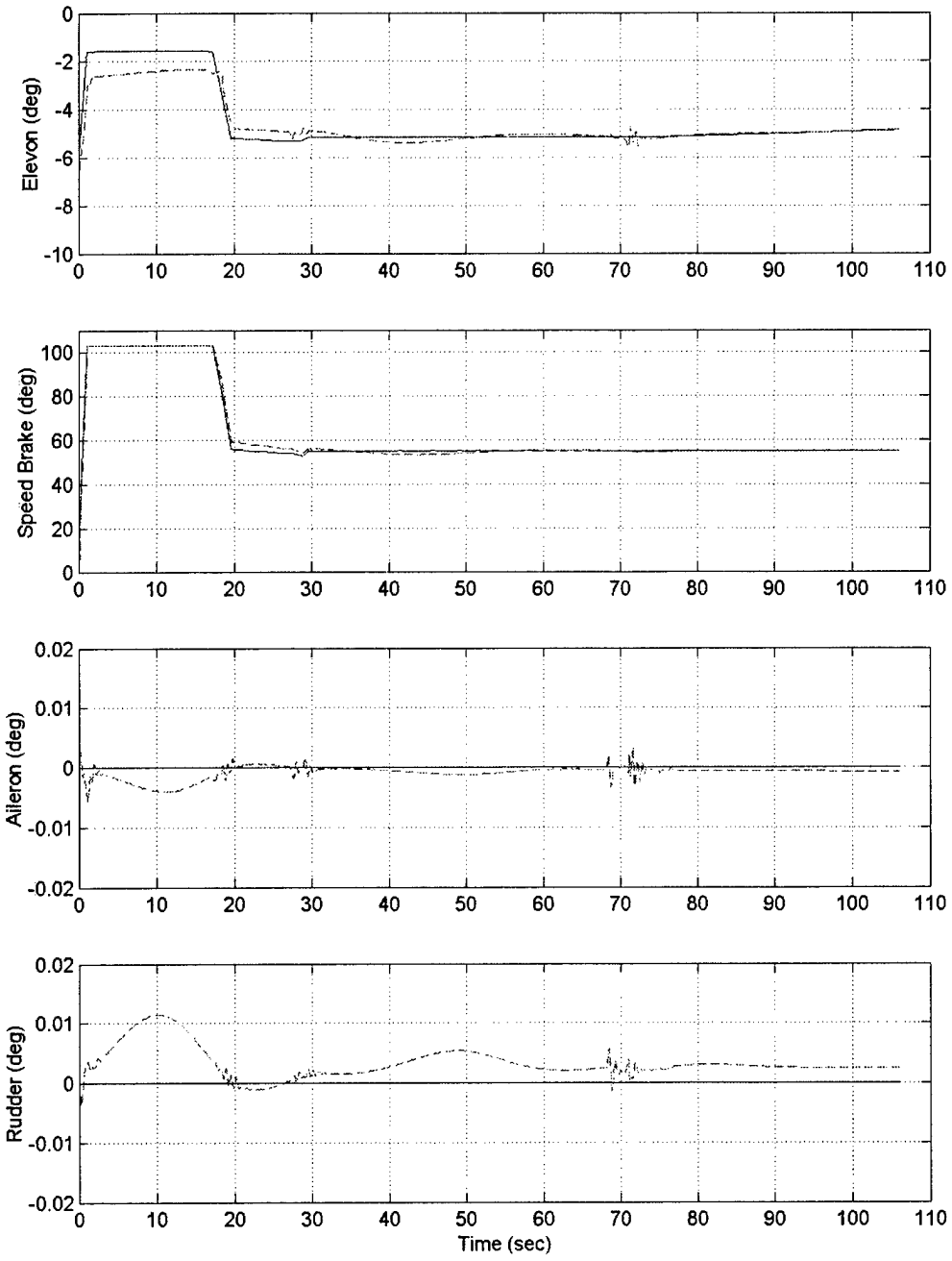

Fig. 125 MPC_ALL Short Range (5 of 5)

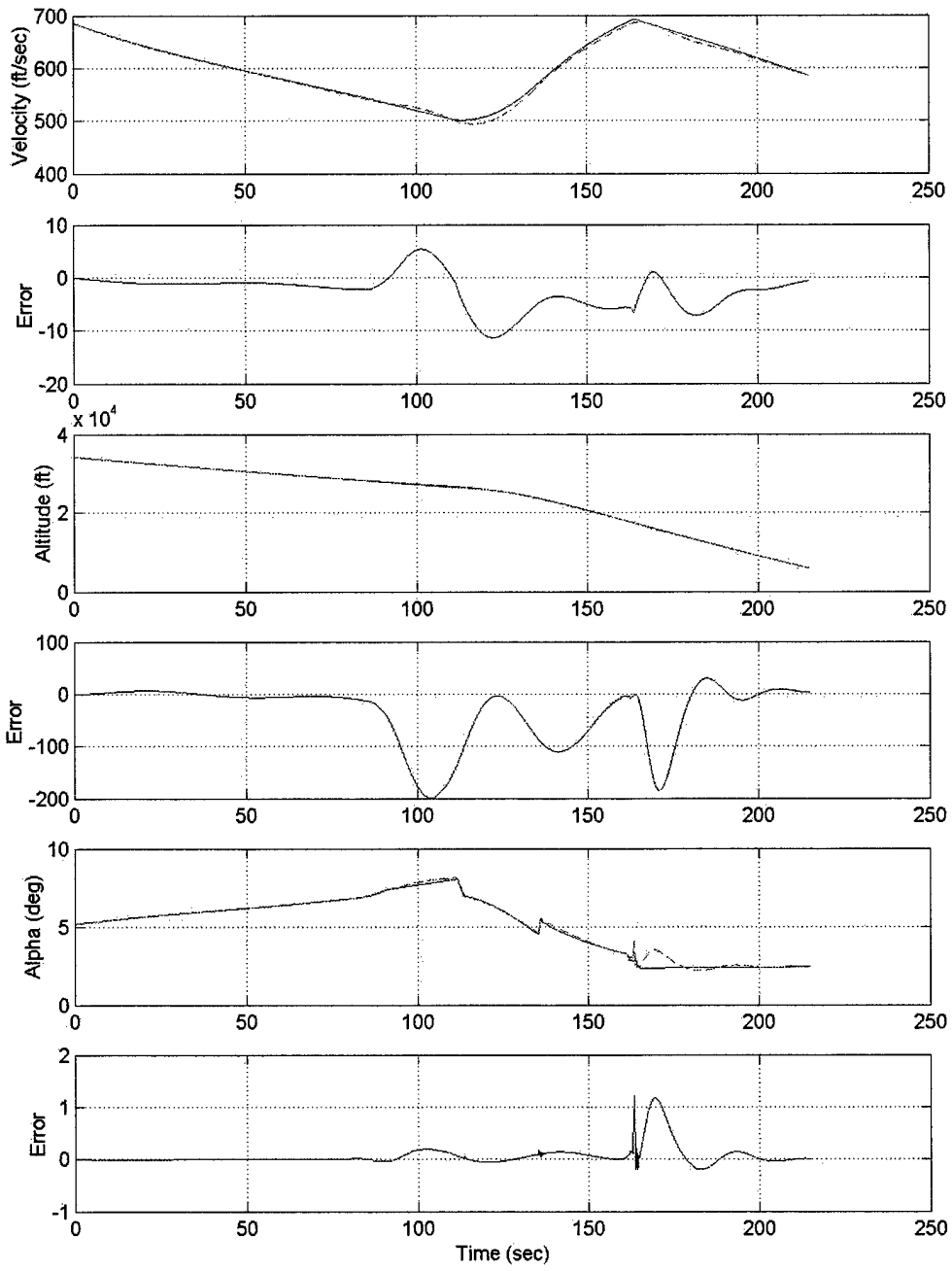

Fig. 126 MPC_ALL High Crossrange (1 of 5)

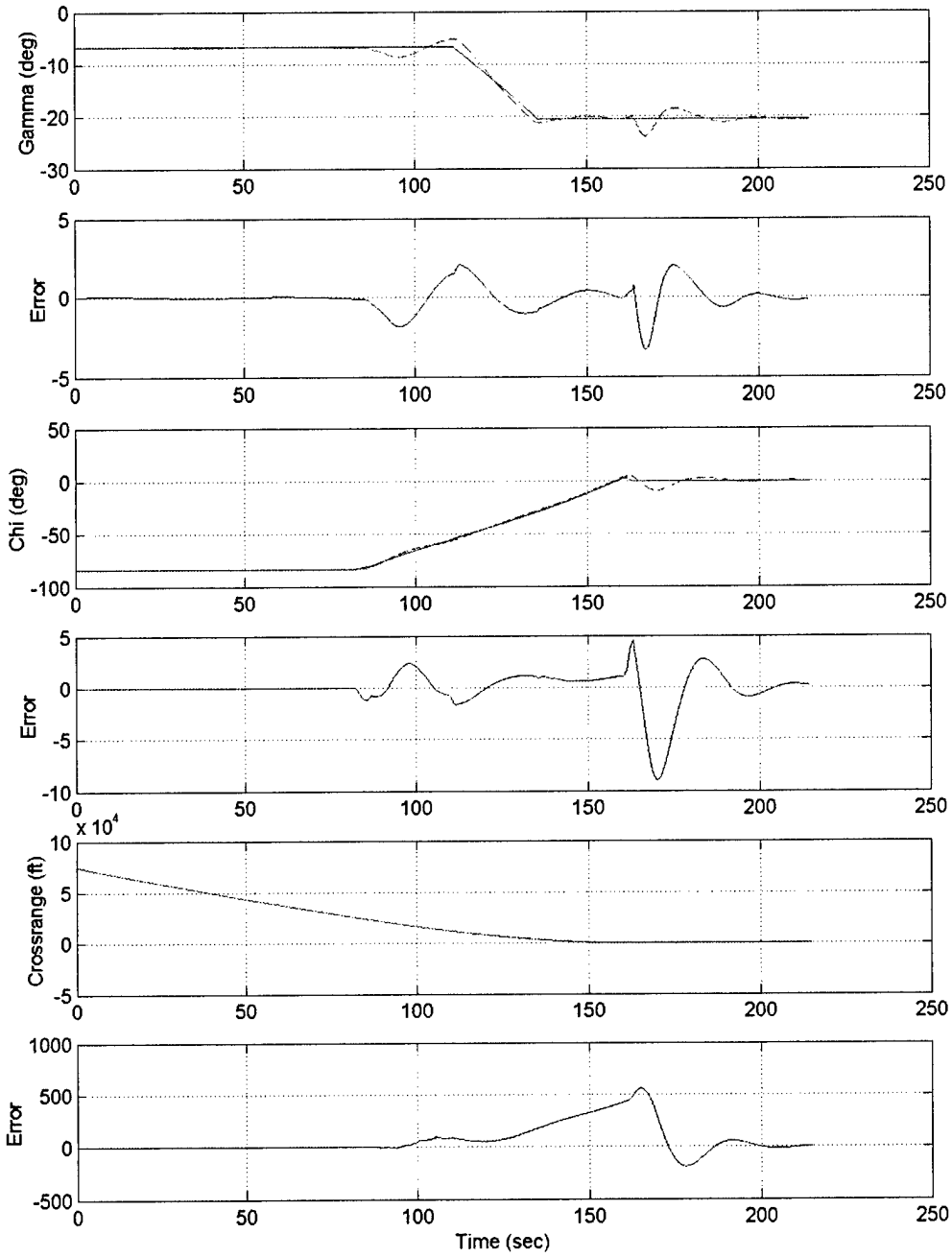

Fig. 127 MPC_ALL High Crossrange (2 of 5)

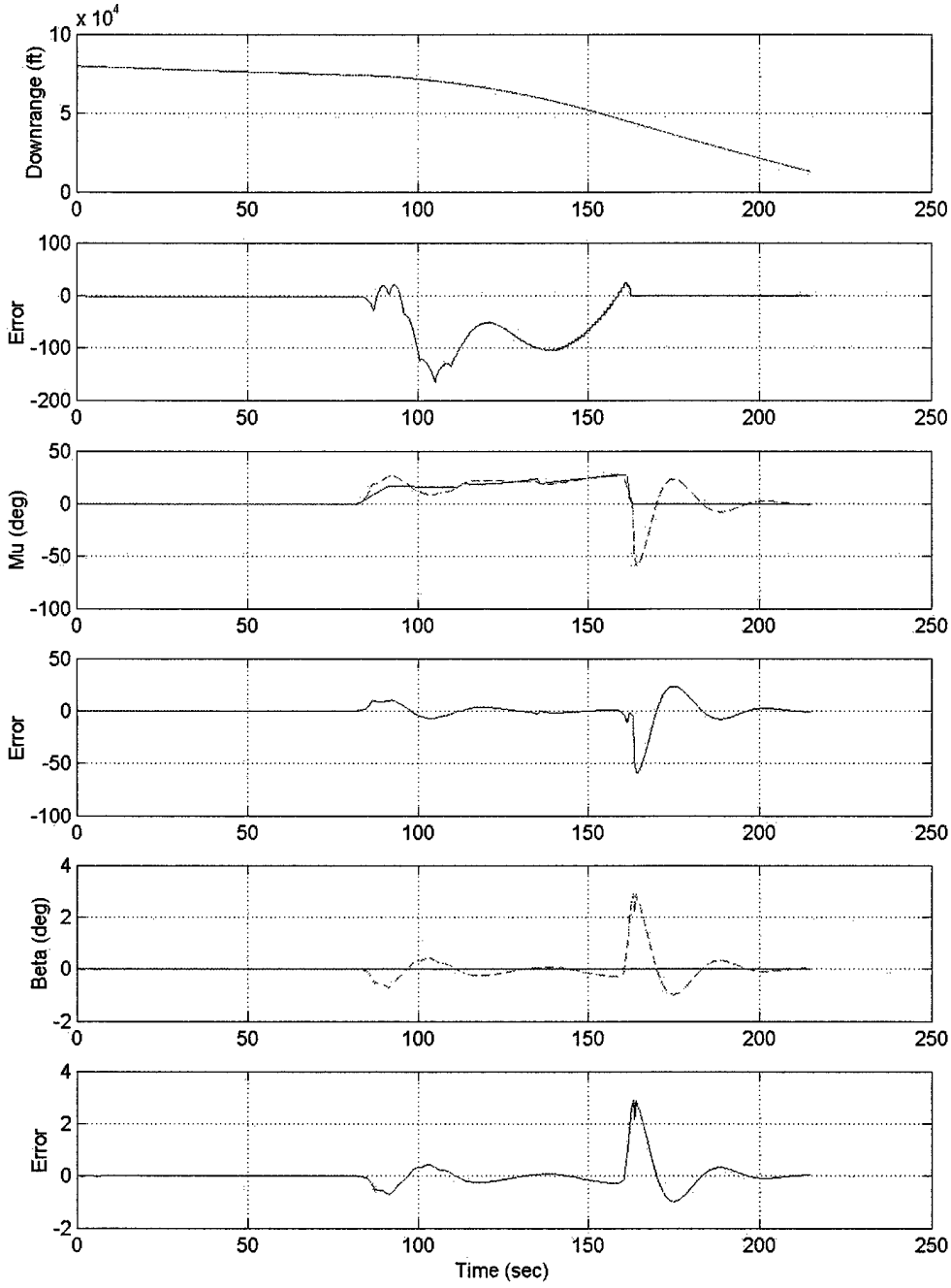

Fig. 128 MPC_ALL High Crossrange (3 of 5)

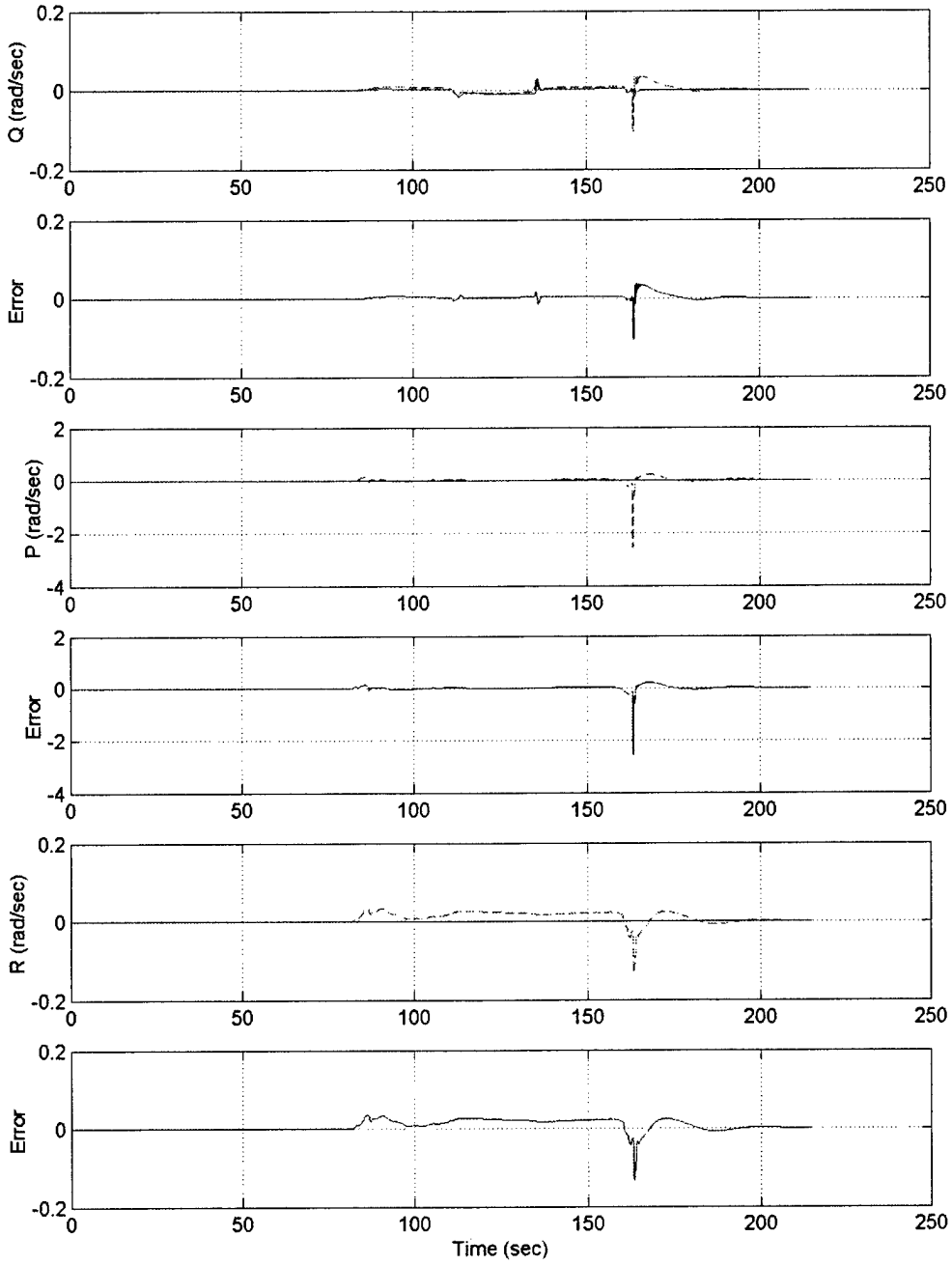

Fig. 129 MPC_ALL High Crossrange (4 of 5)

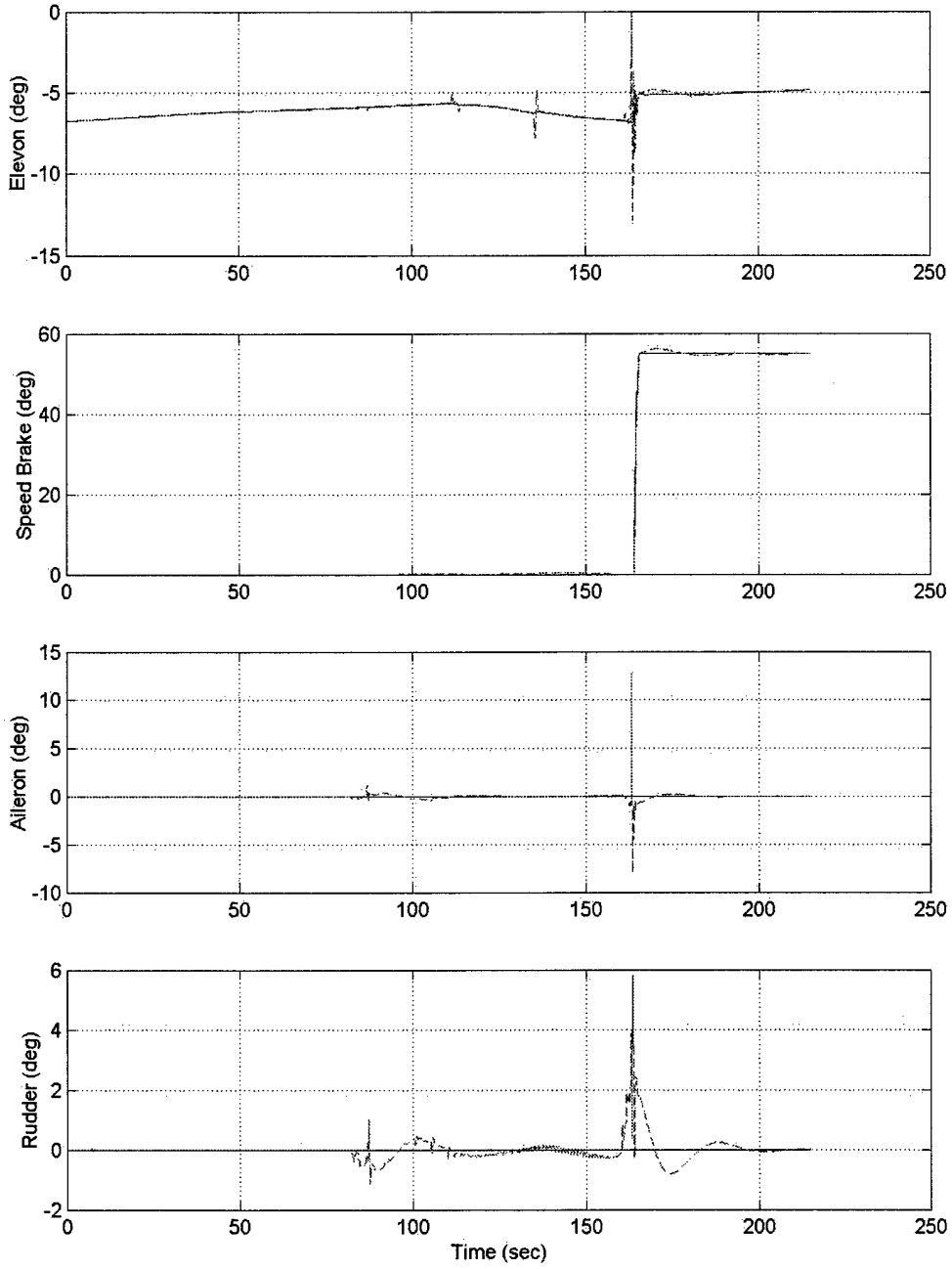

Fig. 130 MPC_ALL High Crossrange (5 of 5)

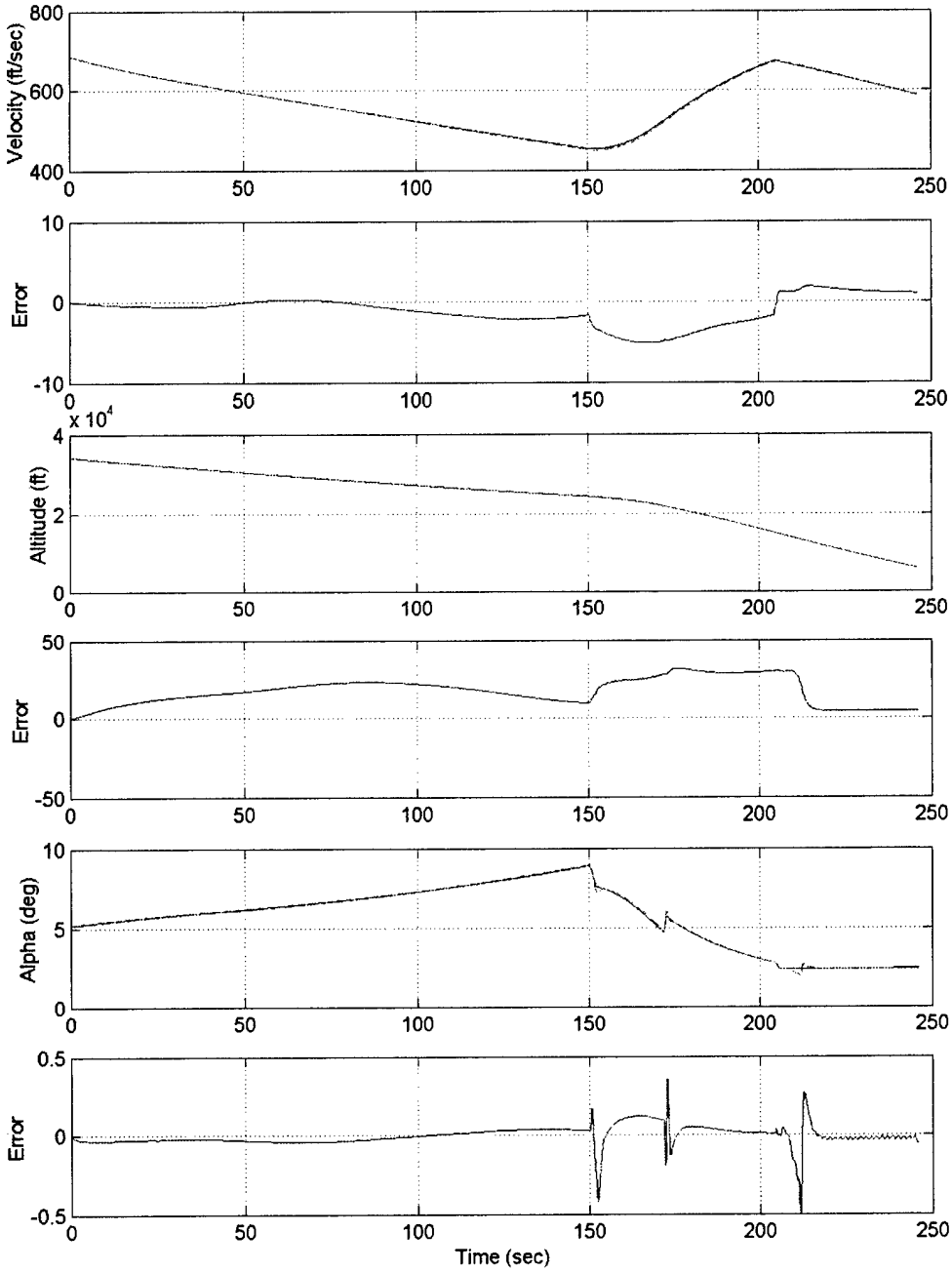

Fig. 131 MPC_SAS Long Range (1 of 5)

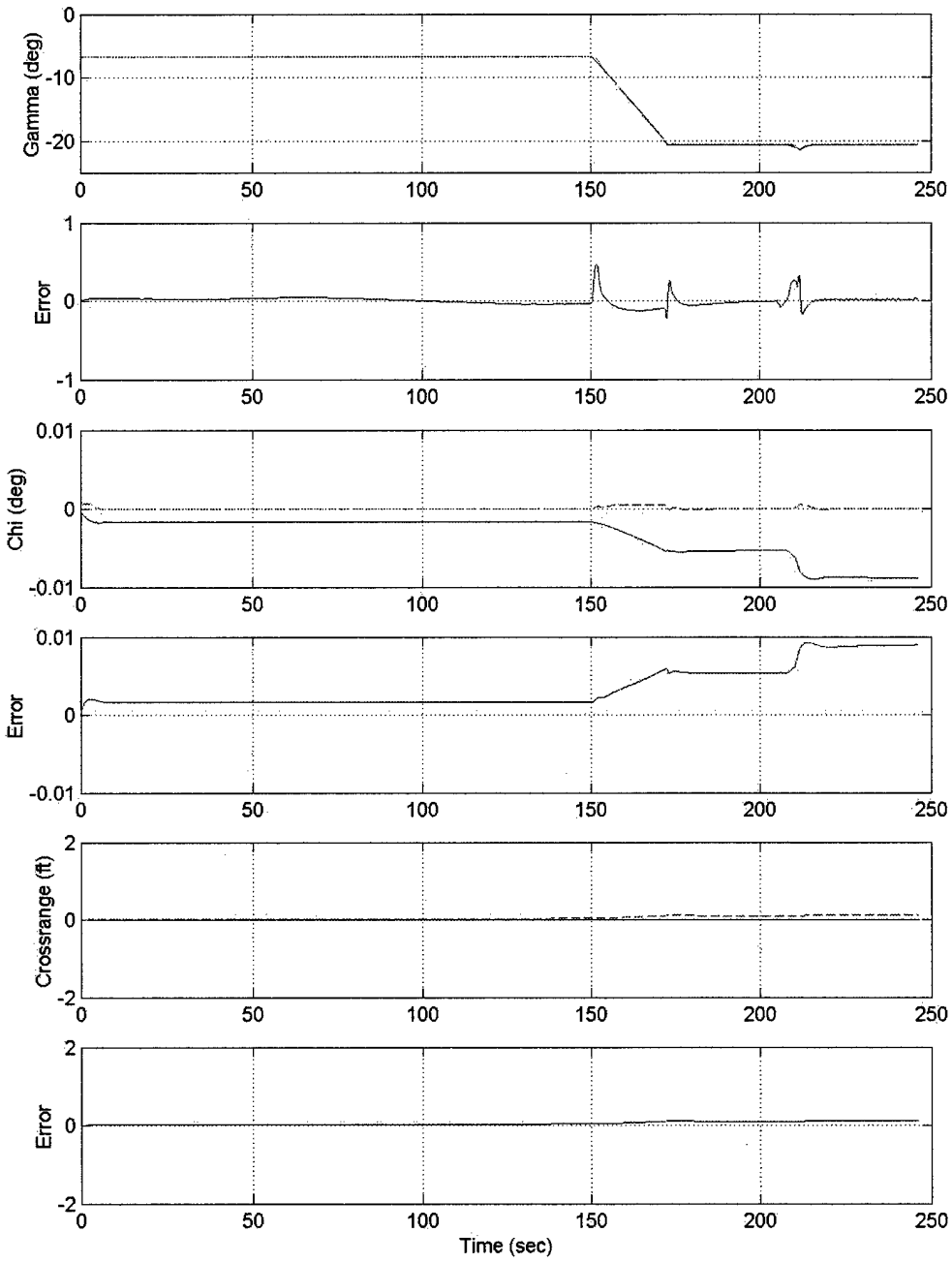

Fig. 132 MPC_SAS Long Range (2 of 5)

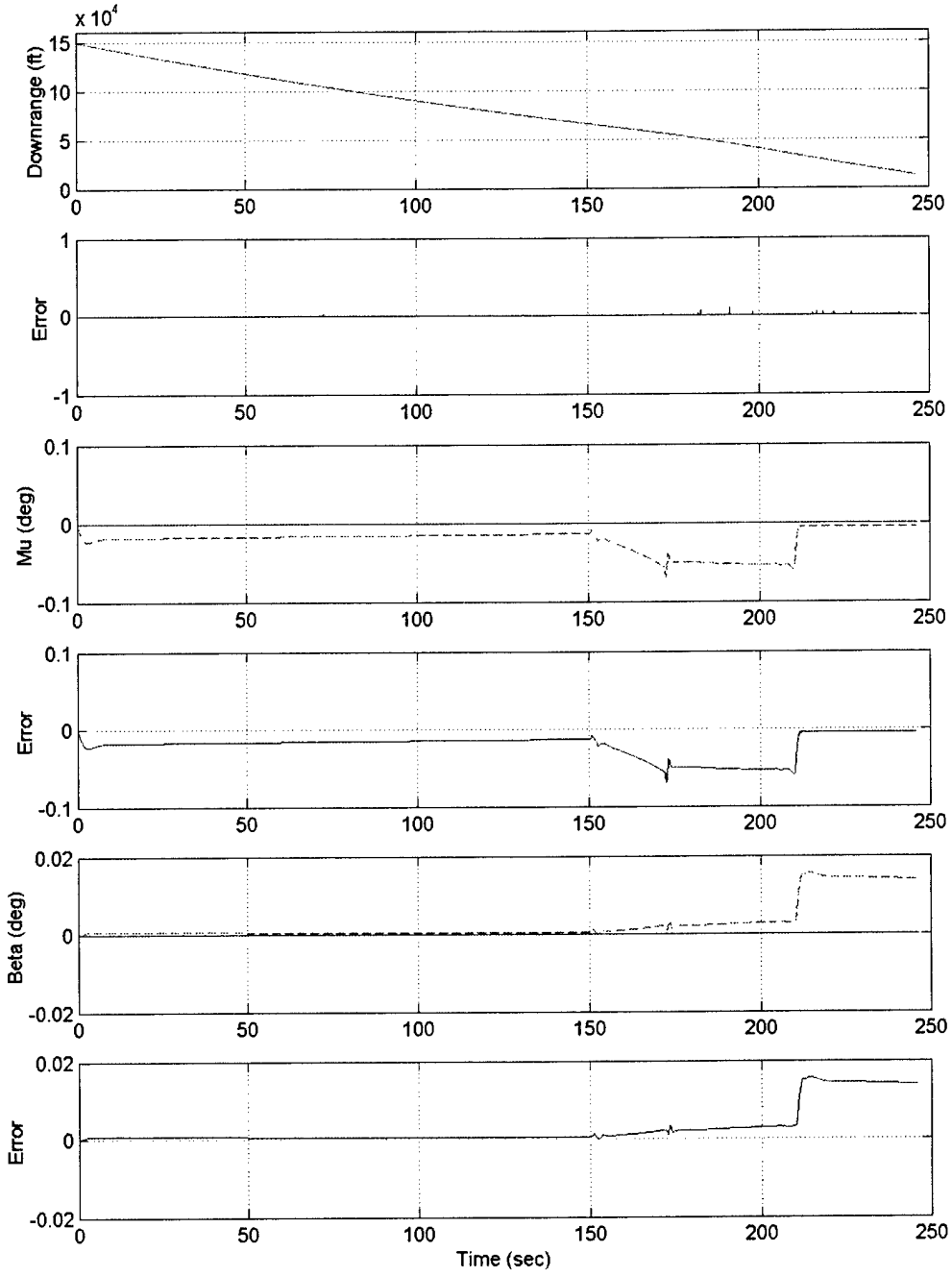

Fig. 133 MPC_SAS Long Range (3 of 5)

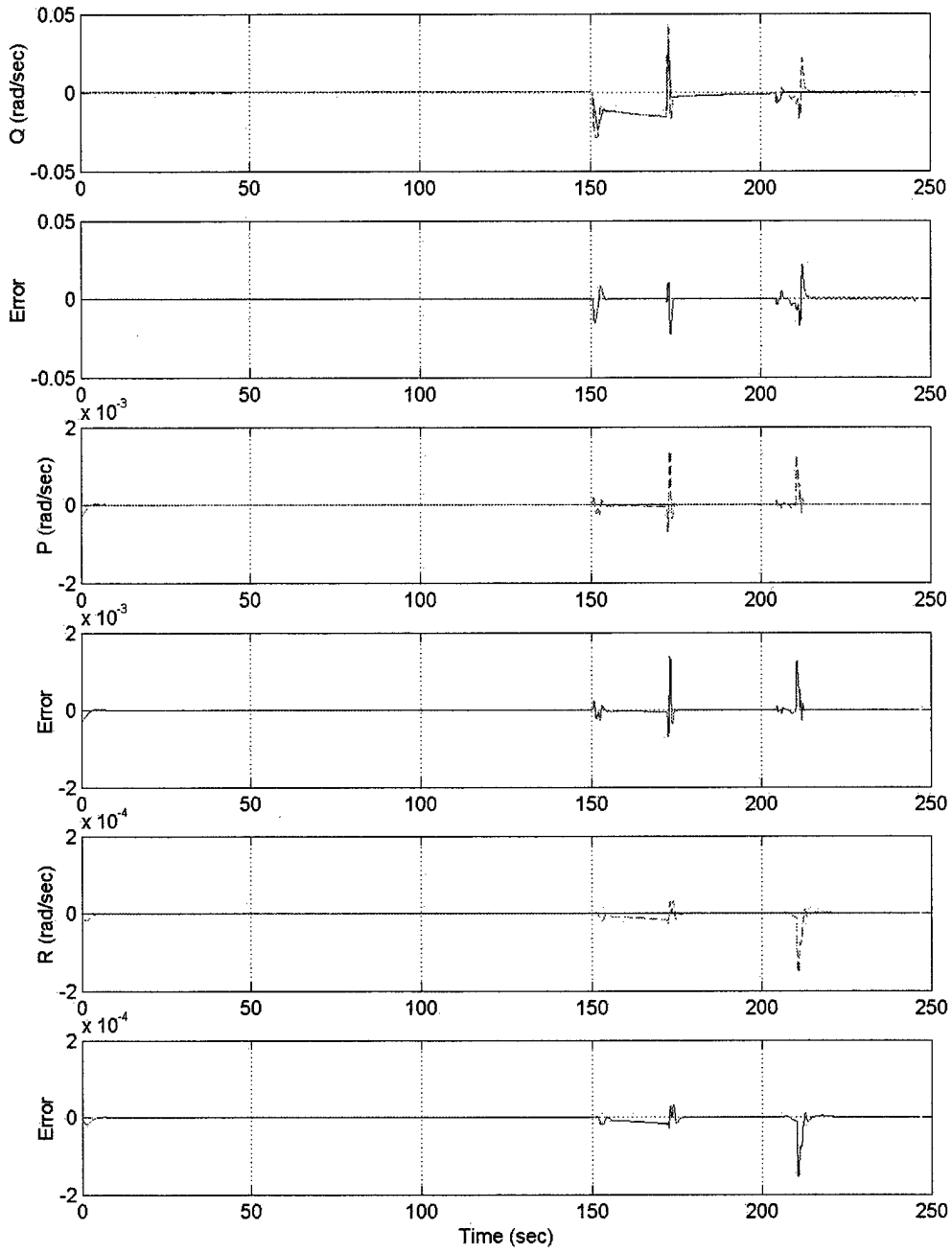

Fig. 134 MPC_SAS Long Range (4 of 5)

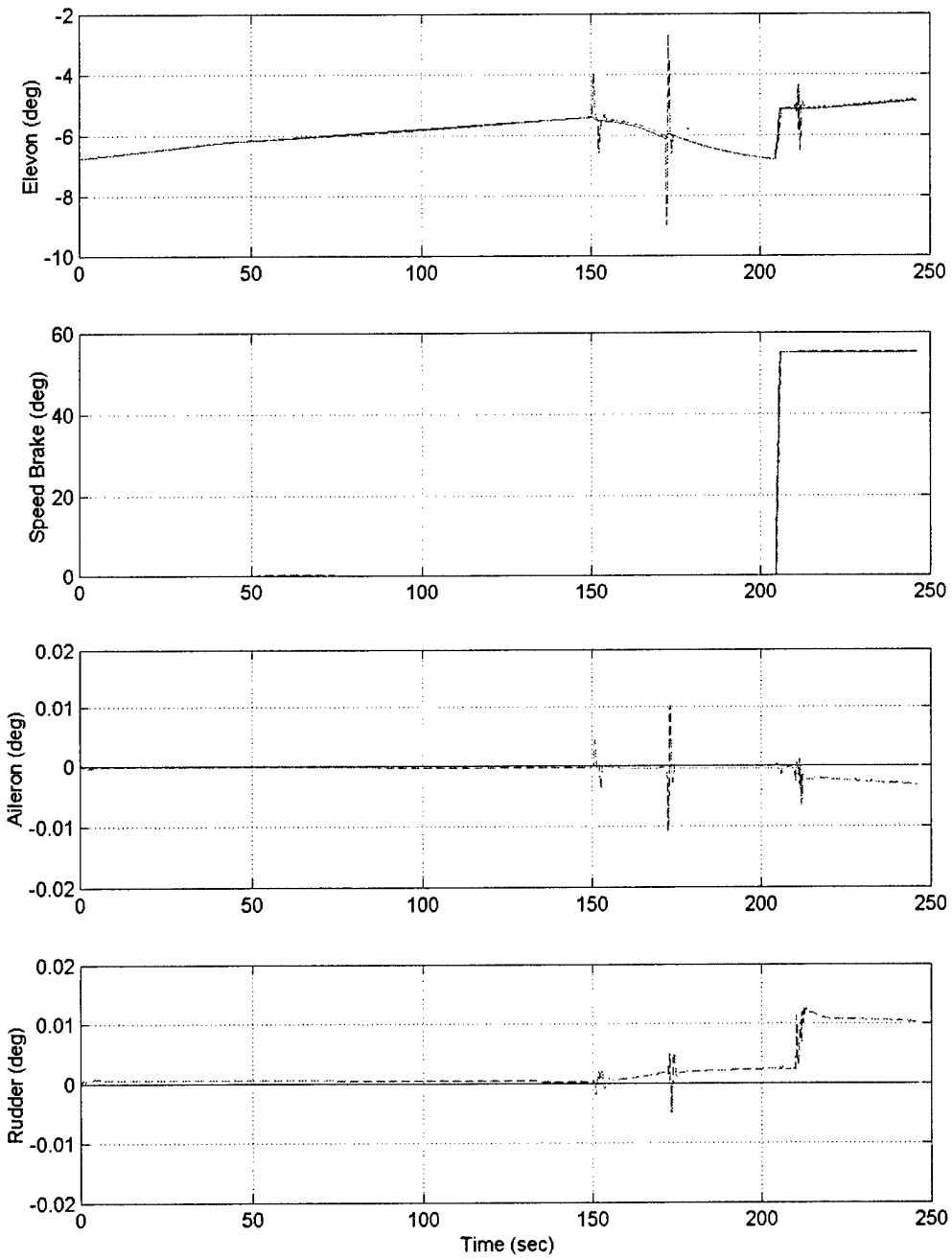

Fig. 135 MPC_SAS Long Range (5 of 5)

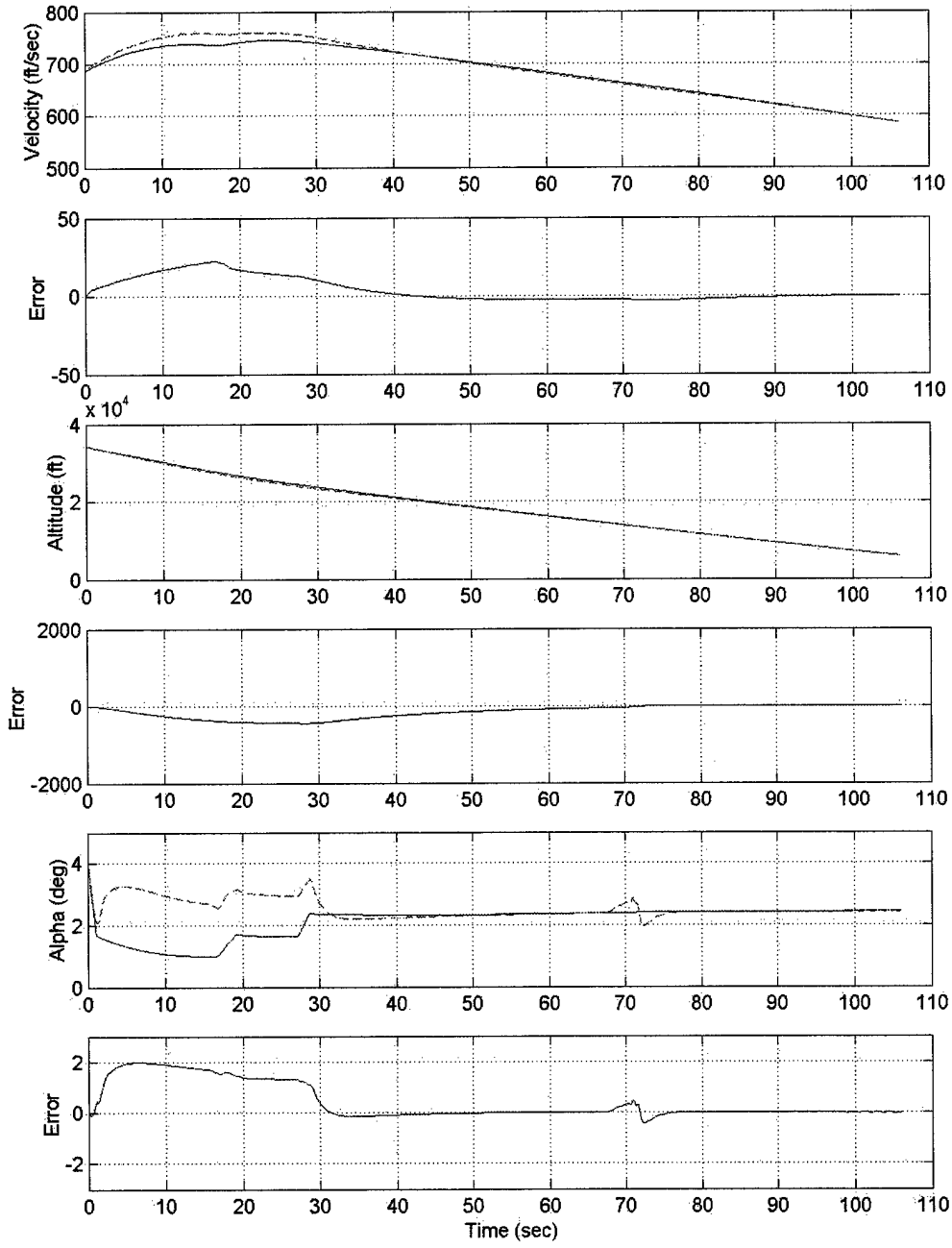

Fig. 136 MPC_SAS Short Range (1 of 5)

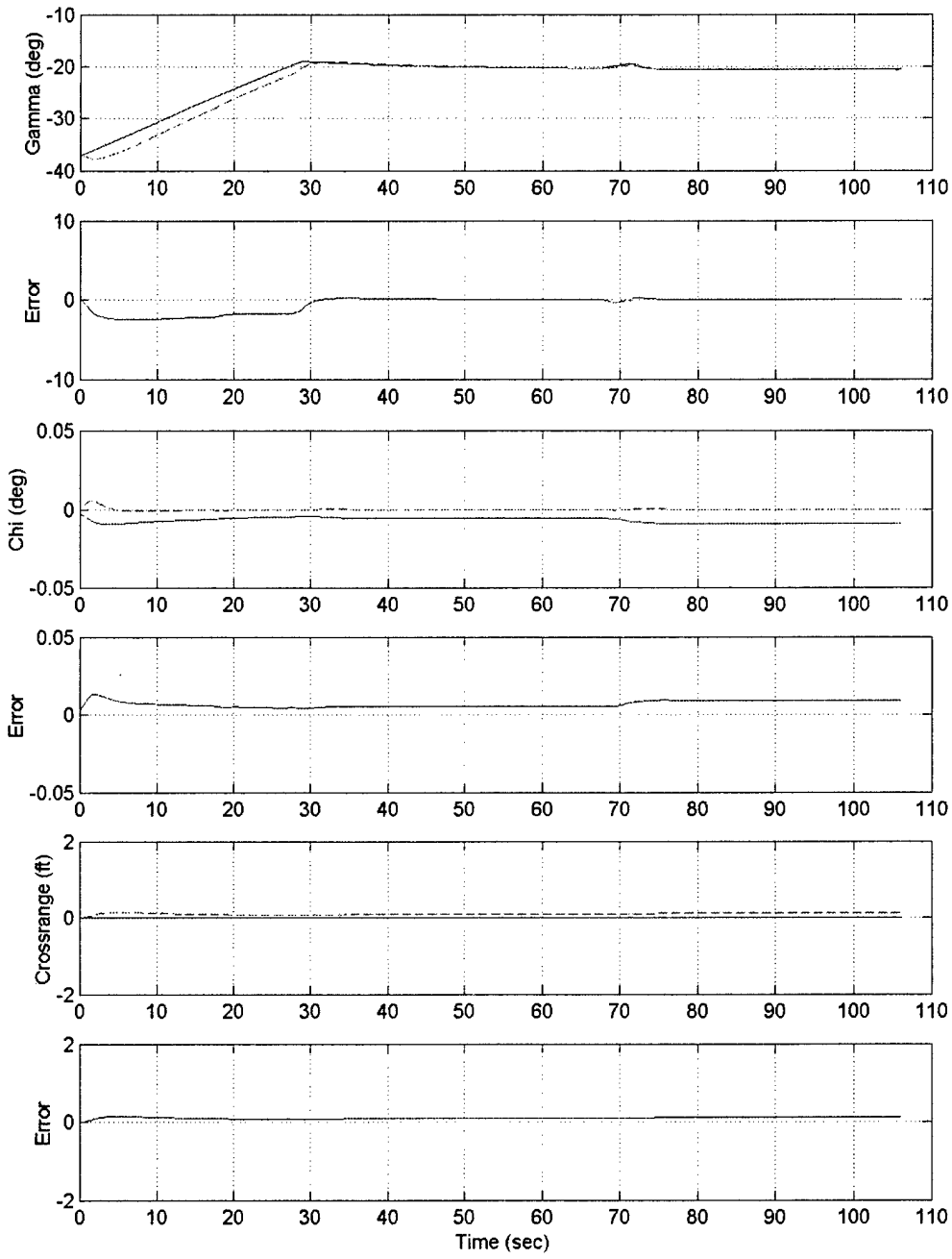

Fig. 137 MPC_SAS Short Range (2 of 5)

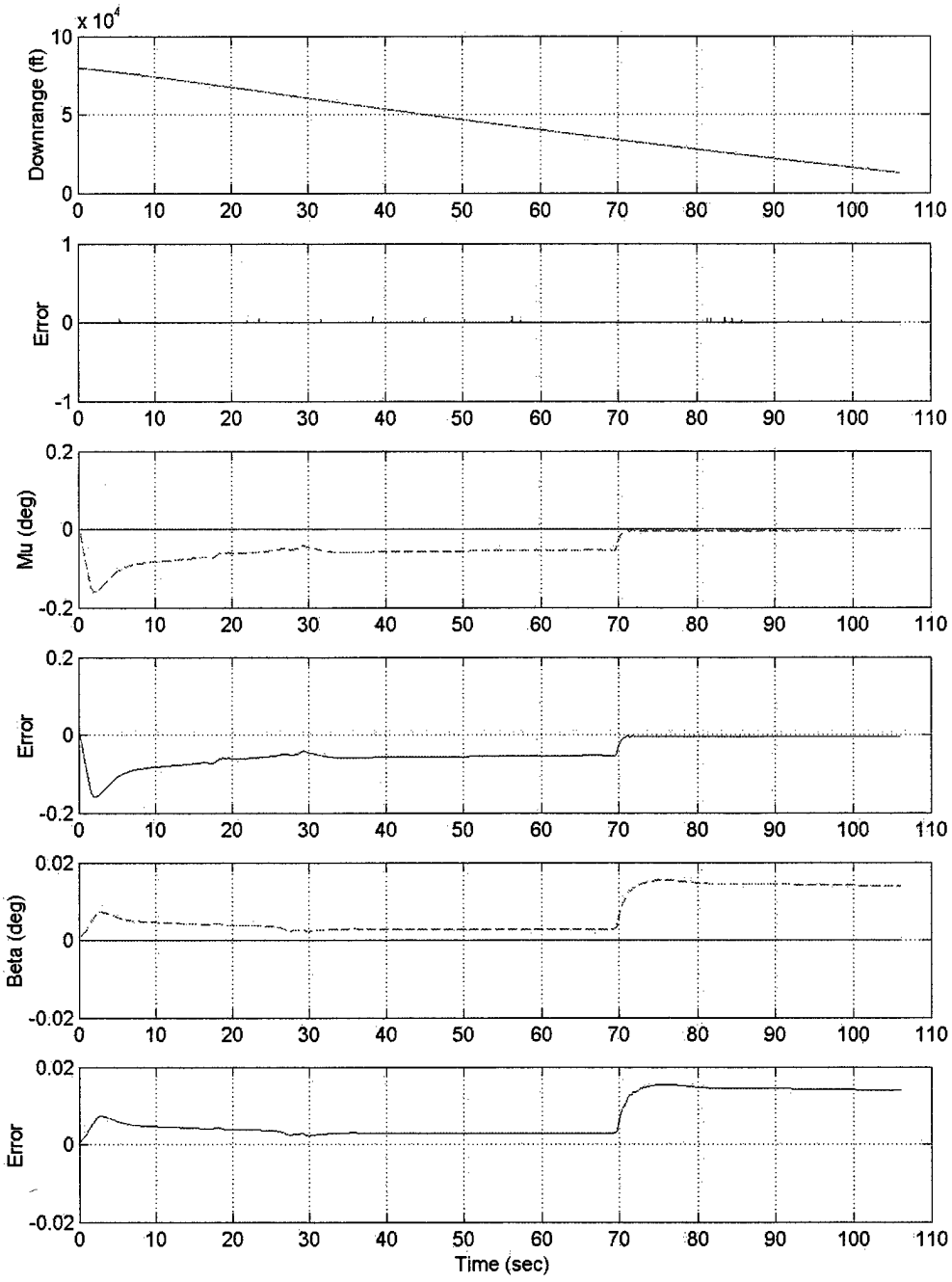

Fig. 138 MPC_SAS Short Range (3 of 5)

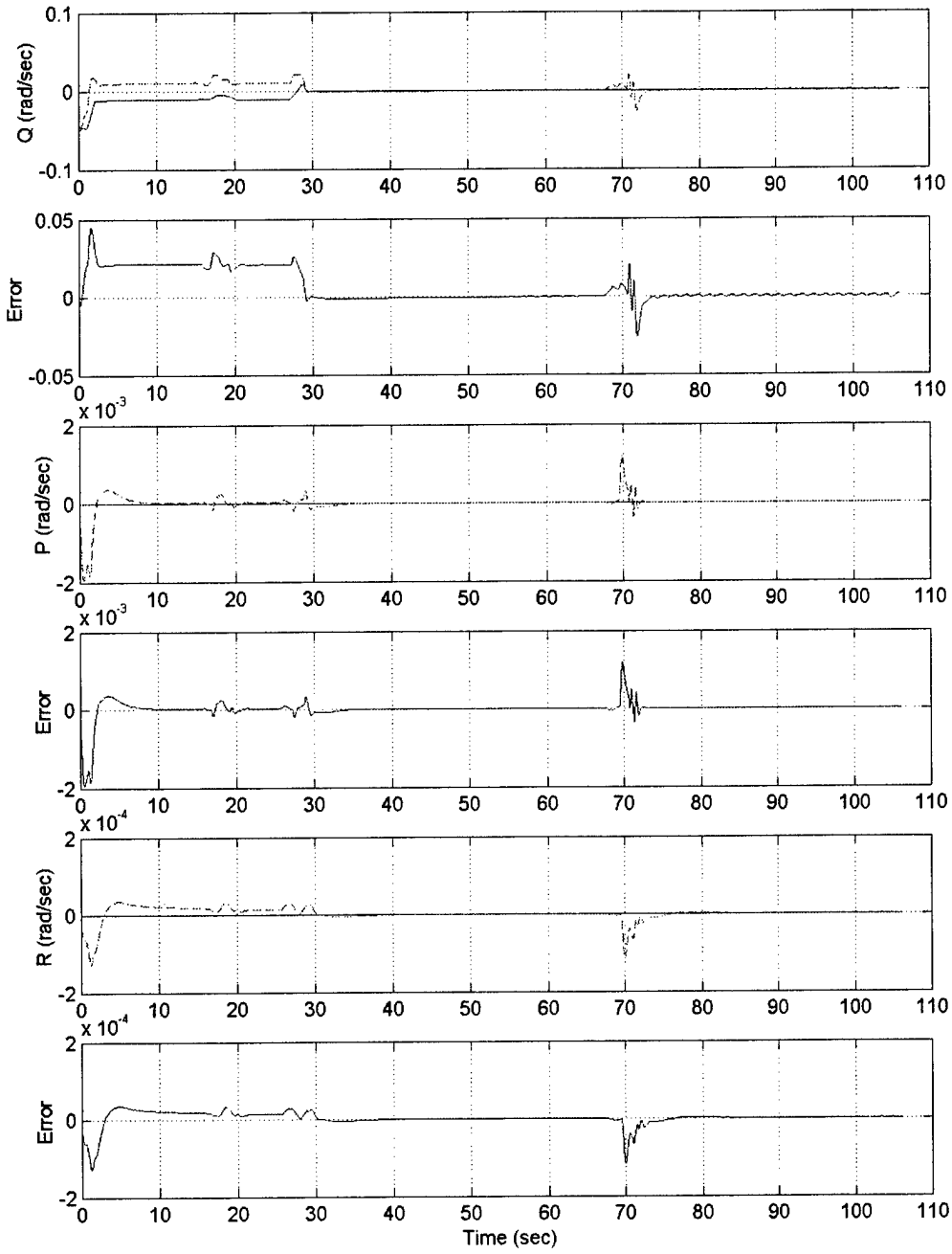

Fig. 139 MPC_SAS Short Range (4 of 5)

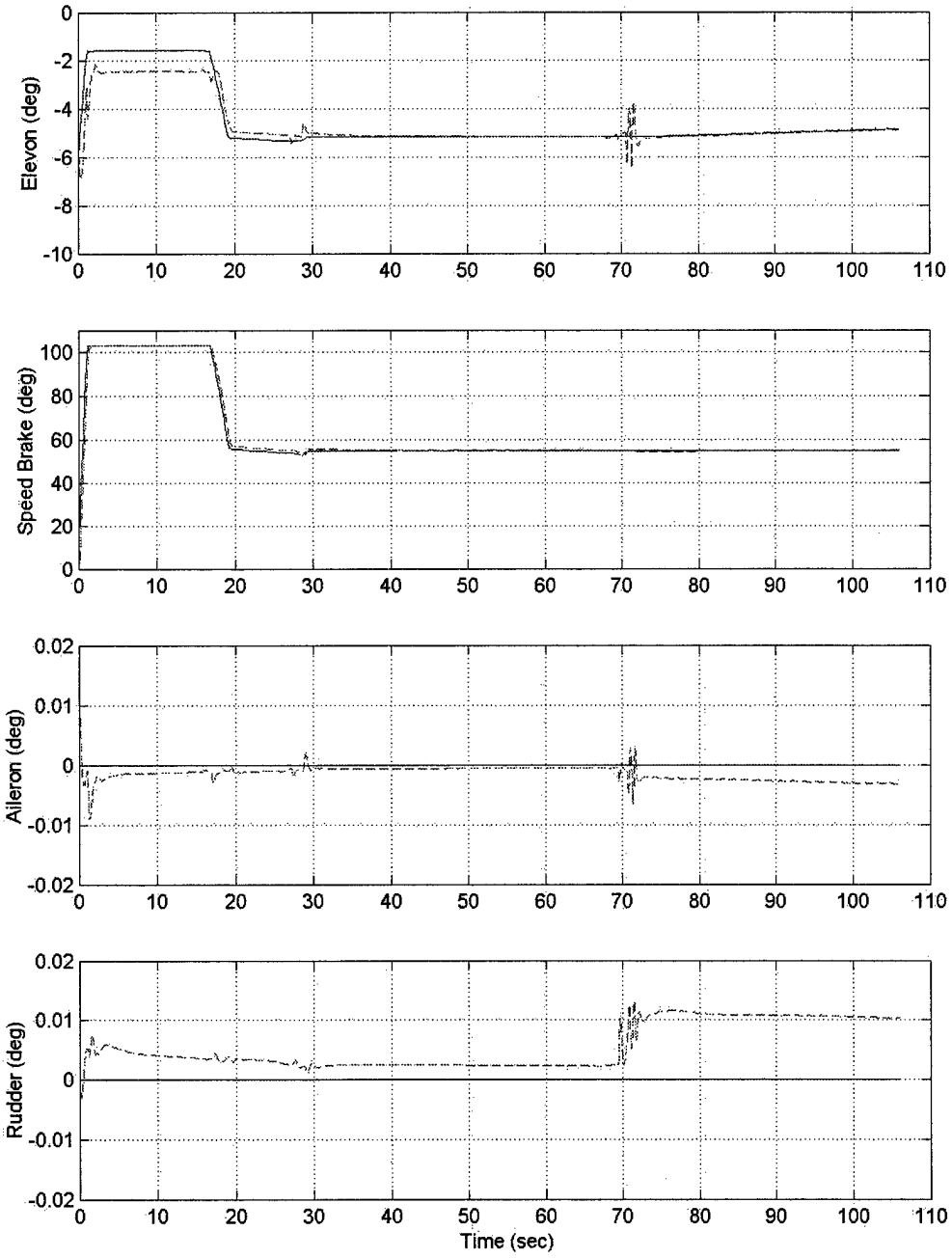

Fig. 140 MPC_SAS Short Range (5 of 5)

[This page intentionally left blank]

References

Ref. 1 Tracy, Chisholm C., Integrated Entry Guidance and Control for Autonomous Reusable Launch Vehicles. Masters Thesis, Department of Aeronautics and Astronautics, MIT, September 1999.

Ref. 2 Space Launch Initiative News web site, http://www.slinews.com. May 2002.

Ref. 3 Chomel, C., Design of a Robust Integrated Guidance and Control Algorithm for the Landing of an Autonomous Reusable Launch Vehicle. Masters Thesis, Department of Aeronautics and Astronautics, MIT, June 1998.

Ref. 4 Camacho, Eduardo F. and Carlos Bordons, Model Predictive Control. Springer-Verlag. London, England 1999.

Ref. 5 Berlin, F. and P.M. Frank, "Design and Realization of a MIMO Predictive Controller for a 3-Tank System" *Advances in Model-Based Predictive Control*, Ed. Dave W. Clarke, Oxford University Press Inc. June 1994.

Ref. 6 Hauser, John. and Ali Jadbabaie, "Aggressive Maneuvering of a Thrust Vectored Flying Wing: A Receding Horizon Approach" hauser@cds.caltech.edu alij@cds.caltech.edu CDC00-REG1946 2000. Research supported by AFOSR and DARPA

Ref. 7 Shearer, Christopher M., Constrained Model Predictive Control of a Nonlinear Aerospace System. Masters Thesis, Department of Aeronautic Engineering, Air Force Institute of Technology, December 1997.

Ref. 8 Kirk, Donald E., Optimal Control Theory: An Introduction. Prentice-Hall Inc. Englewood Cliffs, New Jersey 1970.

Ref. 9 Bemporad, A., M. Morari and N. L. Ricker, The MPC Simulink Library: User's Guide Version 1. 2000. http://control.ethz.ch/~bemporad/ bemporad@aut.ee.ethz.ch

Ref. 10 S.A. Heise and J.M. Maciejowski, "Stability of Constrained MBPC Using an Internal Model Control Structure" *Advances in Model-Based Predictive Control*, Ed. Dave W. Clarke, Oxford University Press Inc. June 1994.

Ref. 11 Bemporad, Alberto and Manfred Morari, "Robust Model Predictive Control: A Survey ". Automatic Control Laboratory, Swiss Federal Institute of Technology (ETH). Physikstrasse 3, CH-8092 Zurich, Switzerland. bemporad@aut.ee.ethz.ch morari@aut.ee.ethz.ch http://control.ethz.ch/

Ref. 12 Shertzer, Richard H., Control Allocation for the Next Generation of Entry Vehicles. Masters Thesis, Department of Aeronautics and Astronautics, MIT, June 2001.

Ref. 13 Banda, Siva S., Application of Multivariable Control Theory to Aircraft Control Laws. WL-TR-96-3099. Wright Laboratory Air Force Material Command Wright-Patterson Air Force Base, OH 45433-7562, May 1996.

Ref. 14 Girerd, A. R., Onboard Trajectory Generation for the Unpowered Landing of Autonomous Reusable Launch Vehicles. Masters Thesis, Department of Aeronautics and Astronautics, MIT, June 2001.

Ref. 15 Belanger, Pierre R., Control Engineering: A Modern Approach. Saunders College Publishing. Forth Worth, Texas 1995.

Ref. 16 Miotto, Piero., Fixed Structure Methods for Flight Control Analysis and Automated Gain Scheduling. Doctoral Thesis, Department of Aeronautics and Astronautics, MIT, June 1997.