Development of An Intelligent Flight Propulsion Control System

Final Report
October 1, 1999

September 1, 1998 - August 31, 1999

Research supported by the NASA Ames Research Center
NASA Grant No. NAG 2-1174

Dr. A.J. Calise	Principal Investigator
Dr. R.T. Rysdyk	Investigator
B.K. Leonhardt	Research Assistant
J.J. Totah	NASA Grant Monitor
J. Kaneshige	NASA Grant Monitor

School of Aerospace Engineering
Georgia Institute of Technology
Atlanta, GA, 30332-0150

Contents

List of Figures

Summary

The initial design and demonstration of an Intelligent Flight Propulsion and Control System (IFPCS) is documented. The design is based on the implementation of a nonlinear adaptive flight control architecture. This initial design of the IFPCS enhances flight safety by using propulsion sources to provide redundancy in flight control. The IFPCS enhances the conventional gain scheduled approach in significant ways;

- The IFPCS provides a back up flight control system that results in consistent responses over a wide range of unanticipated failures.
- The IFPCS is applicable to a variety of aircraft models without redesign and,
- significantly reduces the laborious research & design necessary in a gain scheduled approach.

The control augmentation is detailed within an approximate *Input-Output Linearization* setting. The availability of propulsion only provides two control inputs, symmetric and differential thrust.

Earlier Propulsion Control Augmentation (PCA) work performed by NASA provided for a trajectory controller with pilot command input of glidepath and heading. This work is aimed at demonstrating the flexibility of the IFPCS in providing consistency in flying qualities under a variety of failure scenarios. This report documents the initial design phase where propulsion only is used. Results confirm that the engine dynamics and associated hard nonlinearities result in poor handling qualities at best. However, as demonstrated in simulation, the IFPCS is capable of results similar to the gain scheduled designs of the NASA PCA work. The IFPCS design uses crude estimates of aircraft behaviour.

The adaptive control architecture demonstrates robust stability and provides robust performance. In this work, robust stability means that all states, errors, and adaptive parameters remain bounded under a wide class of uncertainties and input and output disturbances. Robust performance is measured in the quality of the tracking. The results demonstrate the flexibility of the IFPCS architecture and the ability to provide robust performance under a broad range of uncertainty. Robust stability is proved using Lyapunov like analysis.

Future development of the IFPCS will include integration of conventional control surfaces with the use of propulsion augmentation, and utilization of available lift and drag devices, to demonstrate adaptive control capability under a greater variety of failure scenarios. Further work will specifically address the effects of actuator saturation.

1 Introduction

The objective of this work is the design and demonstration of an intelligent model inversion controller architecture that provides control augmentation using propulsive forces only. The controller does not require extensive tuning and is applicable to a variety of aircraft without redesign. It provides backup flight control and maintains consistent response over a wide range of uncertainties and disturances, including unanticipated failures.

This report describes the lateral and longitudinal applications. The applications aim to provide redundancy to the conventional control surfaces. The possibilities of using propulsive control to provide reasonable handling are explored. The application of propulsive control is limited by throttle saturations in both rate and position. Furthermore, the bandwidth of the augmentation is governed by the engine dynamics.

One of the objectives is to implement the controller architecture on the "Stone-Soup-Simulator" (SSS) as used by NASA Ames. This provides the proper environment for integration onto the Advanced Flight Control Simulator (AFCS) used at Ames. The code associated with the SSS in its current version does not provide a convenient environment for development and analyses. Therefore, a similar nonlinear dynamic model representative of a civilian transport aircraft was used for the analyses in this report. The results of integration into the SSS code are included as a seperate chapter.

1.1 The NASA PCA program

Following the DC-10 accident at Sioux City, Iowa in 1989, the NTSB recommended;

> Encourage research and development of backup flight control systems for newly certified wide-body airplanes that utilize an alternate source of motive power separate from that source used for the for the conventional control system [1].

In general there is no satisfactory method onboard the aircraft for effectively controlling the flight path with a disabled primary flight control system. Manual throttle control of the engines is extremely difficult due to the dynamic response of the aircraft in this situation. NASA Dryden Flight Research Center (DFRC) investigated the use of propulsion for emergency flight control in a variety of aircraft. These investigations demonstrated that throttle control of engines alone can be used to augment or replace the aircraft primary control system to safely land the aircraft. DFRC used specific control laws for bank angle and flightpath angle in the flight control computer, driving the engines in response to pilot input commands. An example as applied to a Boeing 747-400 is presented in Fig. 1.1, taken from Ref. [2]. The gains in Fig. 1.1 are scheduled with respect to

- velocity,
- altitude,
- c.g. location,
- operating mode:

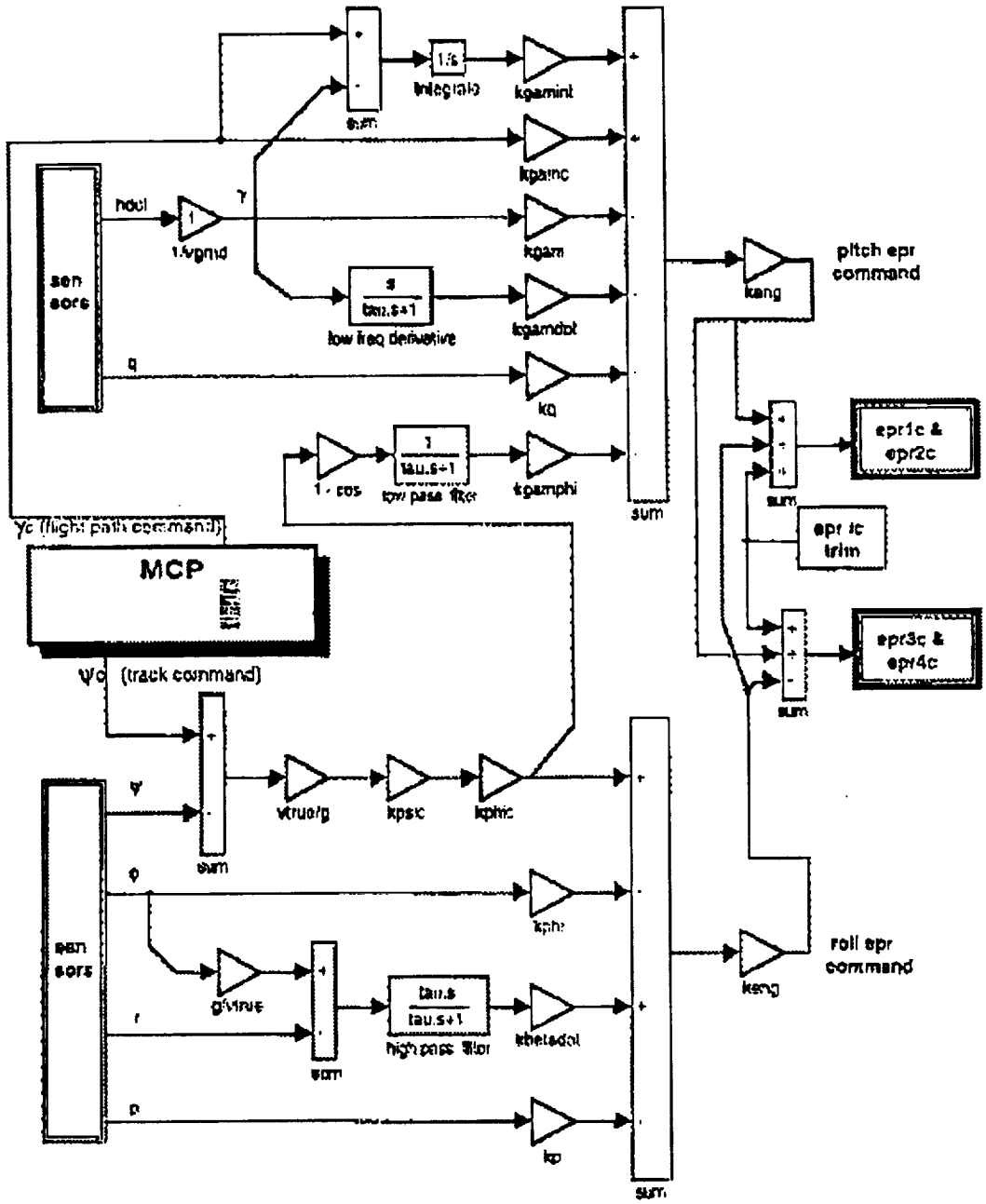

Figure 1.1: PCA conventional control law block diagram. Gains are scheduled with: velocity, altitude, c.g. location, operating mode, configuration, nature of failure, location of operational engines.

- unusual atitude recovery,
- cruise flight,
- ILS-approach,

- configuration:
  - flaps,
  - landing gear,

- nature of failure:
  - jammed vs.
  - floating control surfaces

  and,

- number and location of engines available for control.

The experience of the designers, as well as access to accurate simulation models, allowed for good gain scheduled designs based on classic linear control techniques. However, gain scheduling with respect to the nature of failures significantly limits the safety that may be provided by the controller structure. It requires not only that the operator knows which failure condition is current, but also anticipation of the particular failure.

> **The applicability of a back–up flight control system will be greatly increased if it can provide desirable responses over a wide range of unanticipated failures.**

In flight failures may include partial loss off lifting surfaces, or asymmetricaly jammed control surfaces. Furthermore, the design benefits from avoiding the laborious research necessary for a gain scheduling design approach. It can relieve the reliance on accurate simulation models, which are not always available.

1.2 Nonlinear Adaptive Propulsion Control

The work presented in this report aims at the implementation of a nonlinear adaptive flight control architecture that provides a flight safety enhancement by using the propulsive sources as redundant control. The nonlinear adaptive architecture improves upon the conventional gain scheduled approach in design of a PCA, by allowing for a large variety of in flight failures with a minimum of gain scheduling.

A rigorous theoretical development is provided. The theory combines recent developments in *approximate linearization* and *nonlinear adaptive control*. The development is based on a proof using Lyapunov like stability analysis. The proof is not included in this report but can be found in Ref [3].

1.3 Transition of Results

For syntheses, we used simulation involving a linear model of the MD-11, as well as a nonlinear model of a generic airliner in MatLab-environment. The linear representation of the MD-11 is from Ref. [4]. The nonlinear aircraft model is based on the European *Research Civil Aircraft Model* (RCAM) [5]. This aircraft model represents an initiative designed to promote the use of robust flight control design techniques in the European aeronautical industry. The RCAM includes actuator and engine nonlinearities, as well as nonlinear aerodynamics. The aircraft weights 220,000-331,000 lbs. It has two engines underneath its main wings, together capable of providing a thrust to weight ratio of approximately 0.35. The engines have no inclination angle, and are about $6\,ft$ below the cg, and $26\,ft$ outside the plane of symmetry. The aircraft is configured for landing. With these characteristics, the RCAM offered a convenient basis from which to investigate various aspects of the PCA application.

The results obtained within the Matlab environment were subsequently adapted and implemented within the portable version of the miniACFS-code of NASA Ames. This portable version is referred to as *Stone Soup Simulator* (SSS). Due to the limited access to information pertaining to the SSS, it was not practical to obtain reliable stability & control derivatives of the aircraft represented in the code. The nature of the controller presented in this report allows for good performance, under mild assumptions, and using a crude inversion model. However, prudent design practice suggests the use of the most accurate data available. Initially, a Jacobian linearization of the RCAM was used for model inversion control design for this code. A better approximation was later obtained by utilizing Boeing-747 stability & control derivatives [6] within the linear model inversion.

It is expected that the controller architecture will be further evaluated in piloted simulated flight at NASA Ames research facilities in the second phase of this project.

2 Input Output Linearization with Neural Network Augmentation

2.1 Introduction

The controller architecture proposed for this part of the research is based on *Feedback Linearization* (FBL). FBL has attracted a lot of attention in the last decades. Fundamental contributions have been provided in several text-books, including [7, 8, 9]. This method is fundamentally different from conventional linearization, as it is an exact state transformation and uses state feedback, rather than local linearization of the dynamics. FBL algebraically transforms the nonlinear dynamic system into a linear one. Linear control techniques can then be applied to the transformed system.

In exact FBL, the outputs are differentiated with respect to time (or the independent variable) until the first explicit dependence on the control. The number of required derivatives, r, for a given output variable is its *relative degree*. However, sometimes the first appearance of the control is weak or negligable for reasonable values of the control variable. The FBL theory can still be used in the form of *Approximate Feedback Linearization*, formalized by Hauser [10], and extended to non-control affine systems in Ref [11].

Approximate FBL combines the advantages of a conventional linearized approximate model of the aircraft dynamics, within the FBL paradigm of the full nonlinear system. Using an approximate model, rather than exact algebraic expressions results in an inversion error. We augment the approximate FBL with an adaptive 'learning-while-controlling' neural network (NN) to compensate for this error.

2.2 Approximate Feedback Linearization and Bounded Tracking

Consider the aircraft dynamics represented in the general form

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}) \tag{2.1}$$

$$\mathbf{y} = h(\mathbf{x}, \mathbf{u}) \tag{2.2}$$

We wish to design a tracking controller for a smooth and bounded trajectory with bounded derivatives of sufficient order. Suppose that in the i^{th} element of the output the components of $\frac{\partial h_i}{\partial \mathbf{u}}$ are all small. We reorganize those elements of the output function as

$$y_i = \phi_{i,0}(\mathbf{x}) + \varepsilon\psi_{i,0}(\mathbf{x}, \mathbf{u}) \tag{2.3}$$

so that $\varepsilon\psi_{i,0} << \phi_{i,0}$. With this representation, the output may be differentiated successively until the control effect is significant. Each output y_i is differentiated r_i times. Starting with

Eqn. (2.3),

$$\begin{array}{lcll} \dot{y}_i & = & \phi_{i,1}(\mathbf{x}) & + \ \varepsilon\psi_{i,1}(\mathbf{x},\mathbf{u},\dot{\mathbf{u}}) \\ \vdots & = & \vdots & + \ \vdots \\ y_i^{(r_i-1)} & = & \phi_{i,r_i-1}(\mathbf{x}) & + \ \varepsilon\psi_{i,r_i-1}(\mathbf{x},\mathbf{u},\dot{\mathbf{u}},\cdots) \\ y_i^{(r_i)} & = & \phi_{i,r_i}(\mathbf{x},\mathbf{u}) & + \ \varepsilon\psi_{i,r_i}(\mathbf{x},\mathbf{u},\dot{\mathbf{u}},\cdots) \end{array} \tag{2.4}$$

We say that the effect of the control is significant when $\frac{\partial\phi}{\partial\mathbf{u}} = \mathcal{O}(\phi)$. At each step, the terms that are significant are included in ϕ and those that are not are included in $\varepsilon\psi$. This means that $\psi_{i,j}$ includes,

- the time derivative of $\psi_{i,j-1}$,
- small terms from the time derivative of $\phi_{i,j-1}$.

Definition 2.1 *A system of the form*

$$\begin{array}{rcl} \dot{\mathbf{x}} & = & f(\mathbf{x},\mathbf{u}) \ \ \mathbf{x} \in \mathbb{R}^n, \ \mathbf{u}, \ \mathbf{y} \in \mathbb{R}^m \\ \mathbf{y} & = & \Phi_0(\mathbf{x}) + \varepsilon\Psi_0(\mathbf{y},\mathbf{u}) \end{array} \tag{2.5}$$

is said to have a well-defined approximate local vector relative degree $[r_1, r_2, \cdots, r_m]^T$ *in the neighbourhood* $(\Omega_x \times \Omega_u)$ *of a point* $(\mathbf{x}_o, \ \mathbf{u}_o)$ *if the Jacobian* $\frac{\partial\Phi}{\partial\mathbf{u}}$ *is nonsingular* $\forall\mathbf{x} \in \Omega_x$ *and* $\forall\mathbf{u} \in \Omega_u$, *where*

$$\Phi(\mathbf{x},\mathbf{u}) = [\ \phi_{1,r_1} \ \cdots \ \phi_{m,r_m}\ ]^T \tag{2.6}$$

and, where r_i *is the least order time derivative of* y_i *on which the effect of control is significant.*

Theorem 2.1 *Approximate input-ouput linearizability*

For a system of the form (2.5) that has a well-defined approximate relative degree $[r_1, r_2, \cdots, r_m]^T$, *there exists a state-dependent control transformation to a pseudo-control*

$$\boldsymbol{\nu}^* = [\ \nu_1^* \ \nu_2^* \ \cdots \ \nu_m^*\ ]^T = \Phi(\mathbf{x},\mathbf{u}) \tag{2.7}$$

that transforms the system into m decoupled approximately linear systems

$$\begin{array}{rcl} y_1^{(r_1)} & = & v_1^* + \varepsilon\psi_{1,r_1}(\mathbf{x},\ \mathbf{u},\ \dot{\mathbf{u}},\cdots) \\ \vdots & = & \vdots \\ y_m^{(r_m)} & = & v_m^* + \varepsilon\psi_{m,r_m}(\mathbf{x},\ \mathbf{u},\ \dot{\mathbf{u}},\cdots) \end{array} \tag{2.8}$$

A proof is presented in [11].

Consider a system of the form (2.5) that is approximately linearizable. The total approximate relative degree is defined as

$$\bar{r} \ = \ \sum_{i=1}^{m} r_i \tag{2.9}$$

If $\bar{r} = n$, then the closed loop can be interpreted as approximately linearized from input-to-state by transformation to state-variables $\boldsymbol{\xi}$, where

$$\xi_i = [\phi_{i,0}(\mathbf{x}) \cdots \phi_{i,r_i-1}(\mathbf{x})]^T \tag{2.10}$$

$$\boldsymbol{\xi} = \left[\xi_1^T \; \xi_2^T \cdots \xi_m^T\right]^T \tag{2.11}$$

If $\bar{r} < n$, then the nonlinear system (2.5) can be transformed into a so-called *normal form*. The normal form of the system may be represented as

$$\dot{\xi}_i = [\xi_{i,2} \cdots \xi_{i,r_i} \;\; \phi_i'(\boldsymbol{\xi}, \boldsymbol{\mu}, \mathbf{u}) + \epsilon\psi_i'(\boldsymbol{\xi}, \boldsymbol{\mu}, \mathbf{u}, \dot{\mathbf{u}}, \cdots)]^T \tag{2.12}$$

$$\dot{\boldsymbol{\mu}} = \Theta(\boldsymbol{\xi}, \boldsymbol{\mu}) \tag{2.13}$$

With $\bar{r} < n$, there are internal dynamics, which must be examined for stability. In general, these internal dynamics may depend nonlinearly on the normal outputs, $\boldsymbol{\xi}$. Therefore, we consider the so-called *zero-dynamics*, which are the internal dynamics when the output $\mathbf{y}$, and all its time derivatives are maintained at zero by the control $\mathbf{u}$, which implies $\boldsymbol{\xi} \equiv 0$. Thus the control $\mathbf{u}$ is such that the zero–dynamics are formally defined by

$$\dot{\boldsymbol{\mu}} = \Theta(0, \boldsymbol{\mu}) \tag{2.14}$$

Remark 2.1 *It is not necessary to obtain the explicit expression for $\boldsymbol{\mu}$. We may also analyse the internal dynamics by completing the state transformation Eqn (2.12), with $n - \bar{r}$ vector fields, and using the expression for $\mathbf{u}(\mathbf{x})$ that provides for $\boldsymbol{\xi} \equiv 0$.*

Linearized Model Inversion Design

For convenience define $\boldsymbol{\xi}_r$,

$$\boldsymbol{\xi}_r = \left[y_1^{(r_1)} \cdots y_m^{(r_m)}\right]^T \tag{2.15}$$

The approximate linearizing transformation of definition 2.1 and theorem 2.1 may be summarized as

$$\dot{\boldsymbol{\xi}}_r = \boldsymbol{\nu}^* + \epsilon\Psi(\mathbf{x}, \mathbf{u}) \tag{2.16}$$

$$\boldsymbol{\nu}^* = \Phi(\mathbf{x}, \mathbf{u}) \tag{2.17}$$

where, similarly to Eqn 2.6,

$$\Psi(\mathbf{x}, \mathbf{u}) = [\psi_{1,r_1} \cdots \psi_{m,r_m}]^T \tag{2.18}$$

If $\Phi(\mathbf{x}, \mathbf{u})$ is invertible with respect to the control, we may express the original control input in terms of the pseudo control

$$\mathbf{u} = \Phi^{-1}(\mathbf{x}, \boldsymbol{\nu}) \tag{2.19}$$

Considering the nonlinear nature that we wish to allow for the plant, the inverse of $\Phi(\mathbf{x}, \mathbf{u})$ may be particularly complex are impractical to obtain. If the condition of Theorem 2.1 is

satisfied, then the Jacobian $\Phi_{0u} = \frac{\partial \Phi}{\partial u}\Big|_{x_0,u_0}$ is nonsingular for a set $(\mathbf{x}_0, \mathbf{u}_0) \in (\Omega_x \times \Omega_u)$, and a linearized form of 2.19 may be constructed as

$$\mathbf{u} = \mathbf{u}_0 + \Phi_{0u}^{-1}\{\nu - \Phi_0 - \Phi_{0x}(\mathbf{x} - \mathbf{x}_0)\} \tag{2.20}$$

where $\mathbf{u}_0$ and $\mathbf{x}_0$ are trim values of the plant, $\Phi_0 = \Phi(\mathbf{x}_0, \mathbf{u}_0)$, and Φ_{0x} is the Jacobian with respect to the state at $(\mathbf{x}_0, \mathbf{u}_0)$. Using Eqn 2.20 for the control law implementation will result in an inversion error. If we define this inversion error as

$$\Delta(\mathbf{x}_0, \mathbf{u}_0, \mathbf{x}, \nu) = \Phi(\mathbf{x}, \mathbf{u}) - \{\Phi_0 + \Phi_{0x}(\mathbf{x} - \mathbf{x}_0) + \Phi_{0u}(\mathbf{u} - \mathbf{u}_0)\} \tag{2.21}$$

Remark 2.2 *Note that Eqn (2.20) may in fact be implemented with approximations for the trim values, the Jacobians, and the nominal operating point*

$$\mathbf{u} = \hat{\mathbf{u}}_0 + \hat{\Phi}_{0u}^{-1}\left\{\nu - \hat{\Phi}_0 - \hat{\Phi}_{0x}(\mathbf{x} - \hat{\mathbf{x}}_0)\right\} \tag{2.22}$$

The approximations should be designed 'reasonably' accurate to avoid unnecessary control effort. Specifically, the sign of the elements of the Jacobian should be known,

$$sgn(\frac{\partial \hat{\phi}_0}{\partial u}|_{ij}) = sgn(\frac{\partial \phi_0}{\partial u}|_{ij}) \tag{2.23}$$

Using Eqn (2.21), Eqn (2.17) can be expressed as

$$\nu = \Phi(\mathbf{x}, \mathbf{u}) - \Delta(\mathbf{x}_0, \mathbf{u}_0, \mathbf{x}, \nu) \tag{2.24}$$

and the closed loop as

$$\dot{\xi}_r = \nu + \Delta_\varepsilon(\mathbf{x}_0, \mathbf{u}_0, \mathbf{x}, \nu) \tag{2.25}$$

where

$$\Delta_\varepsilon(\mathbf{x}_0, \mathbf{u}_0, \mathbf{x}, \nu) = \Delta(\mathbf{x}_0, \mathbf{u}_0, \mathbf{x}, \nu) + \varepsilon\Psi(\mathbf{x}, \mathbf{u}) \tag{2.26}$$

2.3 Online Neural Network Augmentation

In the first phase of this work we use a separate neural network (NN) for each channel. Thus, the NNs have a single output. The outline provided here is stated for NNs with multiple outputs, allowing a single NN for multiple channels. In future work we will consider the extension of the application to include a single NN for multiple channels [3].

We design the pseudo control signal as

$$\nu = \nu_{pd} - \nu_{ad} - \nu_r \tag{2.27}$$

where ν_{ad} is an adaptive contribution from an online approximator to be detailed next, and ν_r is a robustifying signal. Construct $\nu_{pd} = [\nu_{pd_1} \cdots \nu_{pd_m}]^T$, where if $i = 1, \cdots, m$

$$\nu_{pd_i} = y_{C_i}^{r_i} + \sum_{\rho=1}^{r_i-1} k_{i\rho}\tilde{y}^{(\rho)} \tag{2.28}$$

where the coefficients $k_{i\rho}$ chosen such that the error dynamics $y_i^{r_i} = \nu_{pd_i}$ are exponentially stable. Define $A_i \in \mathbb{R}^{(r_i-1)\times(r_i-1)}$ and $\mathbf{b}_i \in \mathbb{R}^{(r_i-1)\times(1)}$ as

$$A_i = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ \vdots & & & \ddots & & \vdots \\ k_{i1} & k_{i2} & & \cdots & & k_{r_i-1} \end{bmatrix} \qquad \mathbf{b}_i = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}^T \tag{2.29}$$

Define $\tilde{y}^{(\rho)} = \tilde{y}_c^{(\rho)} - \tilde{y}^{(\rho)}$, and also define

$$\mathbf{e}_i = \left[\tilde{y}\ \dot{\tilde{y}}\ \cdots\ \tilde{y}^{(r_i-1)}\right]^T \tag{2.30}$$

With these definitions, the elements of Eqn (2.25), which are

$$y^{(r_i)} = \nu_i + \Delta_{\varepsilon i} \tag{2.31}$$

may be represented in state space canonical form as

$$\dot{\mathbf{e}}_i = A_i\mathbf{e}_i + \mathbf{b}_i\left(\nu_{ad} + \nu_r - \Delta_{\varepsilon i}\right) \tag{2.32}$$

2.4 Linear Neural Network Structure

This section derives a network adaptation law that guarantees boundedness of the tracking error and of the network weights. Consider a linear-in-the-parameters NN approximation of a possibly nonlinear inversion error

$$W^{*'}\beta = \tilde{f} - \varepsilon \tag{2.33}$$

where $0 \le \|\varepsilon\| \le \bar{\varepsilon}$. The vector ε is referred to as the NN *reconstruction error*, or *residual error*. W^* is an $n_1 \times n_2$–matrix of constant 'ideal' parameter values that, in some sense, minimize $\|\varepsilon\|$. These values are not necessarily unique. β is an n_1–vector consisting of basis functions. Here n_2 gives the number of NN outputs.

For a linear-in-the-parameters network

$$\nu_{AD} = \hat{W}'\beta(\bar{\mathbf{x}}, \nu) \tag{2.34}$$

where $\hat{W}$ is an $n_1 \times n_2$–estimation of W^*, and $\bar{\mathbf{x}}$ is made up of selected elements of the state vector. The selection of the elements of $\bar{\mathbf{x}}$ is done through careful assessment of the inversion error [12]. Since the input to the NN includes the pseudo control signal, ν , which in turn is a function of the output of the NN, ν_{AD} , a fixed point problem occurs. ν is input to the NN through a *squashing* function. This insures that at least one fixed point solution exists.

The error dynamics can be written with $n_2 = 1$ as

$$\dot{\mathbf{e}} = A\mathbf{e} + \mathbf{b}(\nu_{AD} - \tilde{f}) \tag{2.35}$$

with

$$A = \begin{bmatrix} 0 & 1 \\ -K_P & -K_D \end{bmatrix} \tag{2.36}$$

and $\mathbf{e} = [\int \tilde{q}\ \tilde{q}\ ]'$ and $\mathbf{b} = [0\ 1]'$. Define the weight error as $\widetilde{W} = \hat{W} - W^*$. Rewrite (2.35) as

$$\dot{\mathbf{e}} = A\mathbf{e} + \mathbf{b}\widetilde{W}'\boldsymbol{\beta} + \mathbf{b}(W^*\boldsymbol{\beta} - \tilde{f}) \tag{2.37}$$

The update law of the NN weights is designed as

$$\dot{\widetilde{W}} = \dot{\hat{W}} = -\Gamma\zeta\boldsymbol{\beta} \tag{2.38}$$

$\Gamma = \Gamma' > 0$ and where ζ is a scalar filtered error term,

$$\zeta = \mathbf{e}'P\mathbf{b} \tag{2.39}$$

with

$$P = \begin{bmatrix} \frac{K_D}{K_P} + \frac{1}{2K_D} & \frac{1}{2K_P} \\ \frac{1}{2K_P} & \frac{1+K_P}{2K_PK_D} \end{bmatrix} = \begin{bmatrix} P_{11} & P_{12} \\ P_{12} & P_{22} \end{bmatrix} \tag{2.40}$$

Remark 2.3 *The implementation as presented provides for ACAH in the longitudinal channel, i.e. a design with second order tracking error dynamics. This design may be generalized where*

$$\mathbf{e} = [\tilde{x}^{(n)}\ \tilde{x}^{(n-1)}\ \cdots\ \tilde{x}\ ]' \tag{2.41}$$

and $b = [0 \cdots 0\,1]'$ a size-n vector, and P an $n \times n$-matrix that solves $A_n'P + PA_n - 2I_n = 0$, with A_n in canonical form similar to (2.36) but expanded to order n.

Remark 2.4 *Notice that the closed loop system defined by Eqns (2.37), (2.38), and (2.39) may also be generalized to $n_2 \geq 1$. This would be the case if one NN is used in a controller architecture that integrates the three control channels. In that case we can, for instance, organize*

$$\mathbf{e} = [\tilde{\mathbf{x}}'\dot{\tilde{\mathbf{x}}}']' \tag{2.42}$$

where $\dot{\tilde{\mathbf{x}}}$ and $\tilde{\mathbf{x}}$ are size n_2-vectors. A will then be designed a $2n_2 \times 2n_2$ block matrix,

$$A = \begin{bmatrix} 0 & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \tag{2.43}$$

with $A_{12} = I_{n_2}$ and

$$\begin{aligned} A_{21} &= -diag\{K_{P_1}, K_{P_2}, \cdots, K_{P_{n2}}\},\ and \\ A_{22} &= -diag\{K_{D_1}, K_{D_2}, \cdots, K_{D_{n2}}\} \end{aligned}$$

where the index corresponds to the elements of $\tilde{\mathbf{x}}$. Furthermore, $b = \begin{bmatrix} 0 & I_{n2} \end{bmatrix}'$ and

$$\dot{\widetilde{W}} = \dot{\hat{W}} = -\Gamma\boldsymbol{\beta}E' \tag{2.44}$$

where $E = [\zeta_1\ \zeta_2 \cdots \zeta_{n2}]'$.

Neural Network Input Design

The NN can consist of any linearly parameterized feedforward structure that is capable of approximately reconstructing the inversion error. Ref [13] uses Radial Basis Functions (RBFs) because these functions are universal approximators even when the network is linearly parameterized. However, it is well know that RBFs are very poor at interpolation between their design centers, and a large number of such basis functions are needed for networks with multi- dimensional input vectors. In Ref [14], RBFs were used to capture variations in Mach number, because in the trans-sonic region, these variations are difficult to represent by polynomial functions. In the current implementation, a single-layer sigma-pi network is used. The inputs to the network consist of the state variables, the pseudo control and a bias term. Fig 2.1 shows a general depiction of a sigma-pi network. The values represent the weights associated with a nested kronecker product of input signal categories, and therefore they are (binary) constants. The values are the variable network weights.

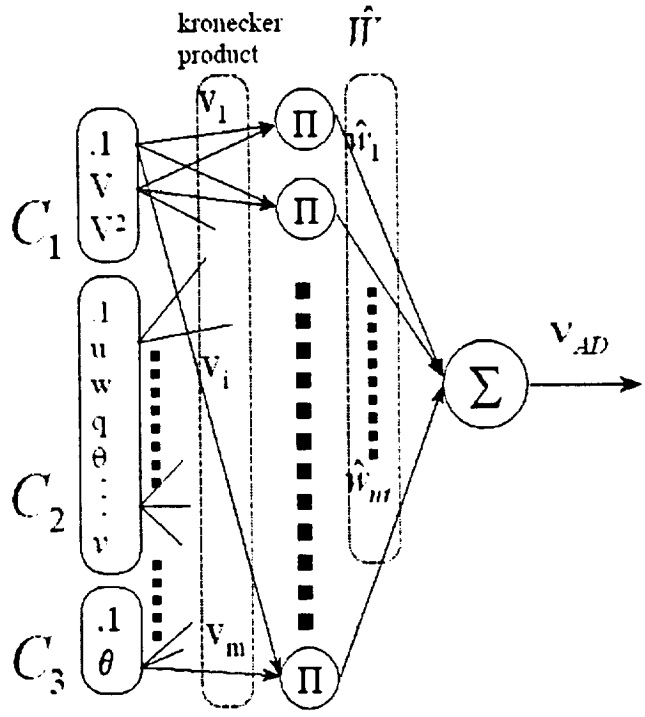

Figure 2.1: Neural network structure, designed for longitudinal channel.

The input/output map of the NN for the longitudinal channel is represented as

$$\nu_{AD} = W'\beta(\bar{x}, \upsilon_{\dot{q}}, \upsilon_{\dot{p}}) \tag{2.45}$$

where $\upsilon_{\dot{q}}$, and $\upsilon_{\dot{p}}$ are respectively the pitch channel and roll channel elements of the pseudo-control υ. The vector of basis functions, β is akin to the *regressor vector* in adaptive control texts. The basis functions are made up from a sufficiently rich set of functions so that the inversion error can be accurately reconstructed at the network output. The basis functions are constructed by grouping normalized inputs into three categories. The first category is

used to model the inversion error effects due to changes in airspeed, since the stability and control derivatives are strongly dependent on dynamic pressure

$$C_1 : \left\{ b_1, V_T, V_T^2 \right\} \tag{2.46}$$

Where b_i indicates a bias term. In allowing the plant to be nonlinear and uncertain in the control as well as in the states, the inversion error is a function of both state variables, x_i, and pseudo control, υ. These, and a bias, are therefore contained in the second category

$$C_2 : \{ b_2, x_i, x_j, \cdots, \upsilon \} \tag{2.47}$$

The third category is used to approximate higher order effects. These may e.g. be due to the transformation between the body frame and the inertial frame

$$C_3 : \{ b_3, x_i \} \tag{2.48}$$

Finally, the vector of basis functions is composed of combinations of the elements of C_1, C_2, and C_3 by means of the kronecker product

$$\beta = \text{kron}\left(\text{kron}(C_1, C_2), C_3\right) \tag{2.49}$$

where,

$$\text{kron}(x, y) = [\, x_1 y_1 \, x_1 y_2 \cdots x_m y_n \,]' \tag{2.50}$$

3 Controller Design

This section describes the approximately linearizing controller design for rate command systems in pitch and roll, as well as a yaw rate command system as alternative to roll rate command. It is assumed that the only available control consists of thrust from the outboard engines.

3.1 Architecture

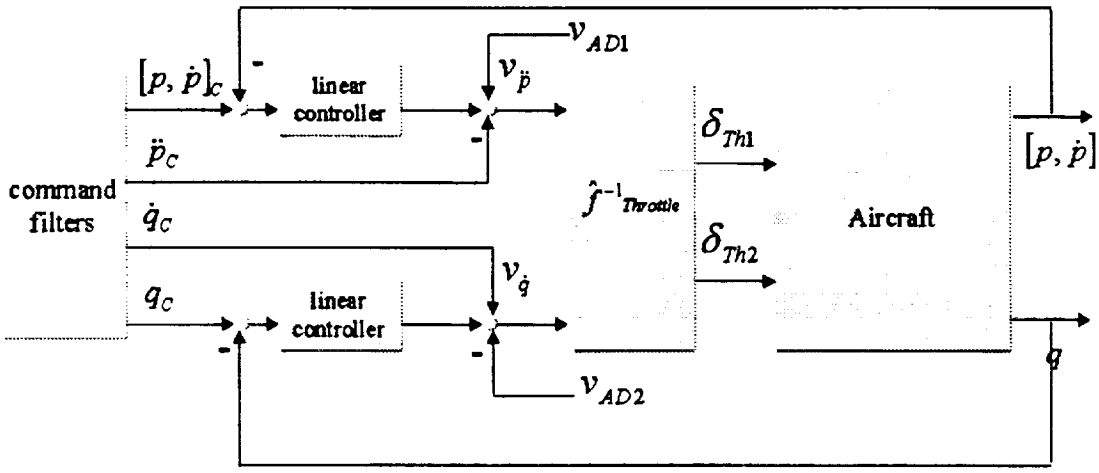

Figure 3.1: Architecture for angular rate commands using propulsive control.

Figure 3.1 displays the architecture for the propulsive-control-only setup for rate-command-attitude-hold (RCAH) in pitch and roll. The inversion with respect to the throttles is based on the symmetric and asymmetric (or differential) application of thrust. These create respectively a moment about the body pitch and yaw axes. The roll rate is therefore indirectly controlled by the application of asymmetric thrust.

Let the aircaft state be

$$\mathbf{x} = [p\, q\, r\, u\, v\, w]^T \tag{3.1}$$

We may represent the rigid body aircraft dynamics as

$$\dot{\mathbf{x}} = [f_1(\mathbf{x})\, f_2(\mathbf{x})\, f_3(\mathbf{x})\, f_4(\mathbf{x})\, f_5(\mathbf{x})\, f_6(\mathbf{x})]^T \tag{3.2}$$

With the following inertia terms,

$$c_3 = \frac{I_z}{I_x I_z - I_{xz}^2},\ c_4 = \frac{I_{xz}}{I_x I_z - I_{xz}^2},\ c_7 = \frac{1}{I_y},\ c_9 = \frac{I_x}{I_x I_z - I_{xz}^2} \tag{3.3}$$

let Γ, Λ, and, Ω represent the aerodynamic moments, constructed as

$$\Gamma = c_3 L + c_4 N \tag{3.4}$$
$$\Lambda = c_7 M \tag{3.5}$$
$$\Omega = c_4 L + c_9 N \tag{3.6}$$

where L, M, and N represent the usual moments about the body fixed axes. Then

$$\begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \end{bmatrix} = \begin{bmatrix} (c_1 r + c_2 p) q & + & \Gamma \\ c_5 p r - c_6 (p^2 - r^2) & + & \Lambda \\ (c_8 p - c_2 r) q & + & \Omega \\ F_x/m & & \\ F_y/m & & \\ F_z/m & & \end{bmatrix} + \begin{bmatrix} 0 & c_4 \\ c_7 & 0 \\ 0 & c_9 \\ 1/(z_e m) & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} M_E \\ N_E \end{bmatrix} \tag{3.7}$$

where M_E and N_E represent respectively the moments due to symmetric and asymmetric thrust. We choose pitch and roll-rate as outputs for input-output linearization using symmetric and asymmetric thrust as input.

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} p \\ q \end{bmatrix} \tag{3.8}$$

The effect of asymmetric thrust on roll rate is mostly a secondary effect. To obtain significant control of roll-rate using asymmetric thrust, we neglect the effect of the inertial coupling c_4 in Eqn 3.7, and therefore consider the second derivative of y_1.

$$\ddot{y}_1 = \frac{\partial f_1}{\partial p}\dot{p} + \frac{\partial f_1}{\partial q}\dot{q} + \frac{\partial f_1}{\partial r}\dot{r} + \Gamma_v \dot{v} \tag{3.9}$$

The effect of thrust appears in the expression for $\ddot{y}_1$ as

$$\ddot{y}_1 \sim \Gamma_r c_9 N_E + (c_1 c_9 + c_2 c_4)\, q\, N_E + (c_1 r + c_2 p)\, M_E \tag{3.10}$$

In the pitch channel, M_E has a significant effect on the first derivative

$$\dot{y}_2 = c_5 p r - c_6 (p^2 - r^2) + \Lambda + c7\, M_E \tag{3.11}$$

For the following analyses we write the system in *normal form* [7]. Therefore define $\boldsymbol{\xi} = [p\ \dot{p}\ q]^T$ and let

$$\boldsymbol{\xi}_r = \begin{bmatrix} \dot{p} \\ q \end{bmatrix} \tag{3.12}$$

Then, the input-output transmission may be presented as

$$\dot{\boldsymbol{\xi}}_r = \mathcal{F}(\mathbf{x}) + \mathcal{G}(\mathbf{x}) \begin{bmatrix} M_E \\ N_E \end{bmatrix} \tag{3.13}$$

Let "$\hat{}$" represent an approximation, then

$$\dot{\boldsymbol{\xi}}_r \approx \hat{\mathcal{F}}(\mathbf{x}) + \hat{\mathcal{G}}(\mathbf{x}) \begin{bmatrix} M_E \\ N_E \end{bmatrix} \tag{3.14}$$

The feedback linearizing transformation is

$$\dot{\boldsymbol{\xi}}_r = \begin{bmatrix} \nu_{\dot{p}} \\ \nu_{\dot{q}} \end{bmatrix} \tag{3.15}$$

$$\begin{bmatrix} \nu_{\dot{p}} \\ \nu_{\dot{q}} \end{bmatrix} = \hat{\mathcal{F}}(\mathbf{x}) + \hat{\mathcal{G}}(\mathbf{x}) \begin{bmatrix} M_E \\ N_E \end{bmatrix} \tag{3.16}$$

M_E and N_E may be obtained in terms of the pseudo control by inversion of 3.16.

$$\begin{bmatrix} M_E \\ N_E \end{bmatrix} = \hat{\mathcal{G}}^{-1}(\mathbf{x}) \left\{ \begin{bmatrix} \nu_{\dot{p}} \\ \nu_{\dot{q}} \end{bmatrix} - \hat{\mathcal{F}}(\mathbf{x}) \right\} \tag{3.17}$$

The inverse of $\hat{\mathcal{G}}$ is

$$\hat{\mathcal{G}}^{-1} = \begin{bmatrix} 0 & \frac{1}{c_7} \\ \frac{c_7}{-\det(\hat{\mathcal{G}})} & \frac{(c_1 r + c_2 p) c_7}{-\det(\hat{\mathcal{G}})} \end{bmatrix} \tag{3.18}$$

where $\det(\hat{\mathcal{G}})$

$$\det(\hat{\mathcal{G}}) = -\left((c_1 q + \Gamma_r) c_9 + (c_2 q + \Gamma_p) c_4\right) c_7 \tag{3.19}$$

Matrix $\hat{\mathcal{G}}$ becomes singular for an unlikely pitch rate,

$$q = -\frac{c_9 \Gamma_r + c_4 \Gamma_p}{c_1 c_9 + c_2 c_4} \geq 0.5 \,\text{rad/sec} \tag{3.20}$$

In order to analyze the remaining dynamics when $\boldsymbol{\xi}$ is controlled, we consider the decoupled linearized modes in the lateral and longitudinal channels. If we choose the design based on these modes, $\hat{\mathcal{G}}$ reduces to

$$\hat{\mathcal{G}}^{-1} = \begin{bmatrix} 0 & \frac{1}{c_7} \\ \frac{1}{\Gamma_r c_9 + \Gamma_p c_4} & 0 \end{bmatrix} \tag{3.21}$$

The symmetric and asymmetric throttle deflections, resp. $s\delta_{th}$, and $a\delta_{th}$ are obtained as

$$\begin{bmatrix} s\delta_{th} \\ a\delta_{th} \end{bmatrix} = \hat{\mathcal{G}}'^{-1} \left\{ \begin{bmatrix} \nu_{\dot{p}} \\ \nu_{\dot{q}} \end{bmatrix} - \hat{\mathcal{F}}(\mathbf{x}) \right\} \tag{3.22}$$

where

$$\hat{\mathcal{G}}'^{-1} = \begin{bmatrix} 0 & \frac{1}{2\Lambda_{\delta_{th}}} \\ \frac{1}{2\Gamma_r \Omega_{\delta_{th}} + 2\Gamma_p \Gamma_{\delta_{th}}} & 0 \end{bmatrix} \tag{3.23}$$

The closed loop stability analyses are contained in section 4.

3.2 Construction of the throttle positions

From the signals $a\delta_{th}$ and $s\delta_{th}$ the throttle positions can be constructed as

$$\delta_{th1} = \frac{1}{2}(s\delta_{th} + a\delta_{th}) \tag{3.24}$$

$$\delta_{th2} = \frac{1}{2}(s\delta_{th} - a\delta_{th}) \tag{3.25}$$

3.3 Construction of the Pseudo Signals

The transformation 3.15–3.17 allows the design of desired pseudo signals. The pseudo signal for Rate-Command-Attitude-Hold in the pitch channel can be constructed as

$$\nu_{\dot{q}} = \dot{q}_c + \alpha_1 \tilde{q} + \alpha_0 \int_0^t \tilde{q}\, dt \tag{3.26}$$

where $\tilde{q} = q_c - q$, and where q_c, $\dot{q}_c$ are obtained using a 1^{st} order command filter 3.1. In the roll channel, we require a 2^{nd} order command filter to generate p_c, $\dot{p}_c$, and $\ddot{p}_c$

$$\nu_{\dot{p}} = \ddot{p}_c + \alpha_2 \dot{\tilde{p}} + \alpha_1 \tilde{p} + \alpha_0 \int_0^t \tilde{p}\, dt \tag{3.27}$$

4 Stability Analyses

4.1 Introduction

Most air-transport aircraft are designed with attitude and speed stability, but may display roll divergence. In the longitudinal direction, an aircraft is trimmed for a certain speed. Given a constant aircraft configuration, each combination of control surface positions will result in a particular speed. Speed stability implies that, given an external disturbance, the aircraft will eventually return to this unique speed.

The trim point, as well as the stability characteristics, may differ between the cases of 'frozen' or 'jammed' control surfaces and 'free floating' surfaces. In working with the SSS version of the miniACFS, there is no provision for simulating free floating control surfaces. Therefore, we used the trim positions provided by the SSS code and 'failure of the control surfaces' simply implies that the control surface positions remain frozen from the time of the failure.

This chapter provides the analyses of the PCA control laws and the associated zero dynamics for the different channels. We apply input-output linearization techniques to provide rate command attitude hold (RCAH) control, and investigate its feasibility using propulsion only.

The stability analyses are performed based on a Jacobian linearized representation of the aircraft in a landing configuration. This approach will provide for a straightforward means to analyze the zero dynamics.

4.2 Roll channel Augmentation

4.2.1 Linear analysis of zero dynamics

The Jacobian linearized form of the lateral dynamics, without the engine dynamics, may be represented as

$$\begin{bmatrix} \dot{p} \\ \dot{r} \\ \dot{v} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \Gamma_p & \Gamma_r & \Gamma_v & \Gamma_\phi \\ \Omega_p & \Omega_r & \Omega_v & \Omega_\phi \\ Y_p & Y_r & Y_v & Y_\phi \\ 1 & \tan\theta_0 & 0 & 0 \end{bmatrix} \begin{bmatrix} p \\ r \\ v \\ \phi \end{bmatrix} + \begin{bmatrix} 2\Gamma_{\delta_{th}} \\ 2\Omega_{\delta_{th}} \\ 2Y_{\delta_{th}} \\ 0 \end{bmatrix} a\delta_{th} \tag{4.1}$$

$$\doteq A\mathbf{x} + \mathbf{b}\, a\delta_{th} \tag{4.2}$$

The effect of asymmetric control input in roll is $\Gamma_{\delta_{th}}/\Omega_{\delta_{th}} << 1$. Following the definitions in Eqns 2.11, and 2.12, we obtain for the lateral channels

$$\boldsymbol{\xi} = [p\ \dot{p}]^T \tag{4.3}$$

The zero dynamics may be analyzed by using $\boldsymbol{\xi} \equiv 0$ and following Remark 2.1. By neglecting the effect of $\Gamma_{\delta_{th}}$ and $Y_{\delta_{th}}$ we may approximately represent the zero dynamics by

$$\begin{bmatrix} \dot{v} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} Y_v - \frac{\Gamma_v Y_r}{\Gamma_r} & Y_\phi - \frac{\Gamma_\phi Y_r}{\Gamma_r} \\ -\tan\theta_0 \frac{\Gamma_v}{\Gamma_r} & -\tan\theta_0 \frac{\Gamma_\phi}{\Gamma_r} \end{bmatrix} \begin{bmatrix} \dot{v} \\ \dot{\phi} \end{bmatrix} \tag{4.4}$$

Example 4.1 For the MD-11 as presented in Ref [4], Eqn (4.4) has eigenvalues at -1.415 and .005. The slow RHP pole is due to the roll divergence mode, and reflects the fact that, given a roll rate command system, the pilot provides the integration of this mode when tracking a desired bank angle. The effect of neglecting $\Gamma_{\delta_{th}}$ and $Y_{\delta_{th}}$, is in the order of magnitude of $1.0e-3$.

4.2.2 Analysis of output tracking dynamics

If we use a linear controller design for the pseudo control $\nu_{\ddot{p}}$

$$\nu_{\ddot{p}} = \ddot{p}_c + \alpha_2(\dot{p}_c - \dot{p}) + \alpha_1(p_c - p) + \alpha_0(\phi_c - \phi)$$

Then if we control $\ddot{p}_c = \dot{p}_c = p_c = \phi_c = 0$ we obtain

$$\begin{aligned} \nu_{\ddot{p}} &= -(\alpha_2\dot{p} + \alpha_1 p + \alpha_0\phi) \\ &= -[\alpha_2\Gamma_p + \alpha_1 \;\; \alpha_2\Gamma_r \;\; \alpha_2\Gamma_v \;\; \alpha_0 \;\; 0]\,\mathbf{x}_{lat} \end{aligned} \tag{4.5}$$

where $\mathbf{x}_{lat} = [p\ r\ v\ \phi\ \psi]'$. We may check the closed loop modes by examining the eigenvalues and corresponding eigenvectors. Choose the roll rate as the output of interest,

$$C = [1\,0\,0\,0\,0] \tag{4.6}$$

Then CA represents the 1^{st} row of A, and

$$\dot{p} = CA\mathbf{x} \Rightarrow \tag{4.7}$$

$$\ddot{p} = CA\dot{\mathbf{x}} \tag{4.8}$$

$$= CA\{A\mathbf{x} + \mathbf{b}\,a\delta_{th}\} \tag{4.9}$$

The linearizing transformation is represented by

$$\ddot{p} = \nu_{\ddot{p}} \tag{4.10}$$

$$\nu_{\ddot{p}} = CA\{A\mathbf{x} + \mathbf{b}\,a\delta_{th}\} \tag{4.11}$$

The first term $A\mathbf{x}$ will serve to cancel the natural modes of the aircraft. For various reasons, we may not want to use the complete linear model of the aircraft in the inversion. For example, in case of non-minimum phase zeros in the system, an inverting controller would be unstable. Also, we may choose not to cancel beneficial cross-coupling in the system. Therefore, allow for approximate linearization [15, 10], expressed as

$$\nu_{\ddot{p}} = CA\left\{\hat{A}\mathbf{x} + \hat{\mathbf{b}}\,a\delta_{th}\right\} \tag{4.12}$$

The inverse control transformation Eqn (4.12) is

$$a\delta_{th} = (CA\hat{\mathbf{b}})^{-1}\left\{\nu_{\ddot{p}} - CA\hat{A}\mathbf{x}\right\} \tag{4.13}$$

$$\doteq -\mathbf{k}_1\mathbf{x} \tag{4.14}$$

In terms of estimates of aerodynamic derivatives, this becomes

$$\begin{aligned} N_E &= \frac{1}{\Gamma_r c_9}\{\nu_{\dot{p}} - (\Gamma_p\Gamma_p + \Gamma_r\Omega_p + \Gamma_v Y_p/m)\,p \\ &\quad - (\Gamma_p\Gamma_r + \Gamma_r\Omega_r + \Gamma_v Y_r/m)\,r \\ &\quad - (\Gamma_p\Gamma_v + \Gamma_r\Omega_v + \Gamma_v Y_v/m)\,v\} \\ &\doteq \frac{1}{\Gamma_r c_9}\{\nu_{\dot{p}} - \Gamma_1 p - \Gamma_2 r - \Gamma_3 v\} \end{aligned} \tag{4.15}$$

The lateral modes in the closed loop are then represented by

$$A_c = A - \mathbf{b}\,\mathbf{k}_1 \tag{4.16}$$

We may analyze the closed loop dynamics about the equilibrium by checking if A_c is Hurwitz.

Example 4.2 For the MD–11, with $\zeta = 0.8$, $\omega_n = 0.5$ and $\alpha_0 = 0$, $\alpha_1 = \omega_n^2$, and $\alpha_2 = 2\zeta\omega_n$, applying 'exact' inversion by using $\hat{A} = A$ in Eqn 4.13, we obtain eigenvalues of A_c as

$$-1.645 \quad -0.379 \pm 0.273i \quad 0.0019 \tag{4.17}$$

The unstable pole is due to the slightly non-minimum phase zero of the open loop aircraft. To avoid the singularity this causes in model inversion control architecture we can now use an approximate inversion by using $\hat{A}$ in Eqn (4.13), where $\hat{A} = A|_{\Gamma_r=0}$. The resulting closed loop then gives poles located at

$$-2.094 \quad -0.130 \pm 0.209i \quad -0.548 \tag{4.18}$$

4.2.3 Effect of engine dynamics

The preceding results do not consider the engine dynamics. The engines are modeled as first order systems, with a time constant τ_{eng}. Let ay_{th} be the asymmetric output of the approximate engine model in terms of throttle position,

$$a\dot{x}_{th} = -\widehat{K}_{eng}\,ax_{th} + \widehat{K}_{eng}a\delta_{thc} \tag{4.19}$$

$$ay_{th} = ax_{th} \tag{4.20}$$

where $K_{eng} = \tau_{eng}^{-1}$. Define $\mathbf{x}' = [\,\mathbf{x}\ ax_{th}\,]^T$. A linear representation of the closed loop then may be constructed as

$$\dot{\mathbf{x}}' = \begin{bmatrix} \mathrm{A} & \mathbf{b} \\ 0\cdots 0 & -K_{eng} \end{bmatrix}\mathbf{x}' + \begin{bmatrix} 0\cdots 0 & K_{eng} \end{bmatrix}^T a\delta_{thc} \tag{4.21}$$

The closed loop is then represented by

$$A_c' = \begin{bmatrix} \mathrm{A} & \mathbf{b} \\ 0\cdots 0 & -K_{eng} \end{bmatrix} - \begin{bmatrix} 0\cdots 0 & K_{eng} \end{bmatrix}^T \begin{bmatrix} \mathbf{k}_1 & 0 \end{bmatrix} \tag{4.22}$$

The resulting eigenvectors & values show the effect of the engine dynamics on the feedback linearization.

$$A_c' = \begin{bmatrix} \Gamma_p & \Gamma_r & \Gamma_v & 0 & 0 & 2\Gamma_{\delta_{th}} \\ \Omega_p & \Omega_r & \Omega_v & 0 & 0 & 2\Omega_{\delta_{th}} \\ \frac{Y_p}{m} & \frac{Y_r}{m} & \frac{Y_v}{m} & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ -\frac{K_{eng}}{\Gamma_r c_9}\beta_1 & -\frac{K_{eng}}{\Gamma_r c_9}\beta_2 & -\frac{K_{eng}}{\Gamma_r c_9}\beta_3 & -\frac{K_{eng}}{\Gamma_r c_9}\alpha_0 & 0 & -K_{eng} \end{bmatrix}$$

where

$$\begin{aligned} \beta_1 &= \Gamma_1 + \alpha_1 + \alpha_2\Gamma_p \\ \beta_2 &= \Gamma_2 + \alpha_2\Gamma_r \\ \beta_3 &= \Gamma_3 + \alpha_2\Gamma_v \end{aligned}$$

with $\Gamma_{1,2,3}$ defined in Eqn 4.15.

Example 4.3 For the MD-11 [4], with $\zeta = 0.8$, $\omega_n = 0.5$ and $\alpha_0 = 0$, $\alpha_1 = \omega_n^2$, and $\alpha_2 = 2\zeta\omega_n$, and using the approximate inversion $\hat{A}_{lat} = A_{lat}|_{\Gamma_r=0}$ in Eqn (4.13), we obtain eigenvalues of A_c' as

$$-1.049 \pm 1.828i \quad -0.112 \pm 0.195i \quad -0.616 \tag{4.23}$$

We may analyze the effect of the engine dynamics in a simple root locus representation, Fig 4.1. This will provide a straightforward tool to aid in choice of the gains by setting the ratio of K_P to K_D, thus providing a root locus by varying K_D.

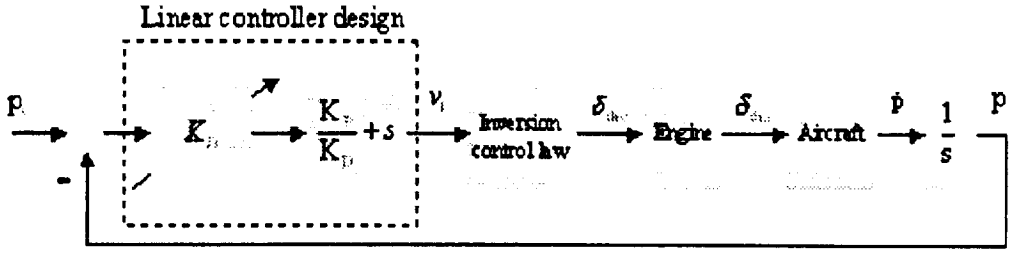

Figure 4.1: Roll channel setup for design of the tracking error dynamics.

4.3 Longitudinal channel

The longitudinal modes, without engine dynamics, are approximated by

$$\begin{bmatrix} \dot{q} \\ \dot{u} \\ \dot{w} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \Lambda_q & \Lambda_u & \Lambda_w & 0 \\ X_q & X_u & X_w & 0 \\ Z_q & Z_u & Z_w & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} q \\ u \\ w \\ \theta \end{bmatrix} + \begin{bmatrix} 2\,\Lambda_{\delta_{th}} \\ 2X_{\delta_{th}} \\ 2Z_{\delta_{th}} \\ 0 \end{bmatrix} s\delta_{th} \tag{4.24}$$

$$\doteq A\mathbf{x} + \mathbf{b}\, s\delta_{th} \tag{4.25}$$

where $s\delta_{th}$ represents *symmetric* throttle deflection, i.e. the sum effect of both throttle deflections, where $\Delta\delta_{th1} = \Delta\delta_{th2}$, and $2\,\Lambda_{\delta_{th}}\, s\delta_{th} = c_7 M_E$ with M_E the moment about the body Y–axis due to engine thrust.

Let $A_{1,:}$ represent the first row of A, and b_1 the first element of $\mathbf{b}$. The feedback linearizing transformation is

$$\dot{q} = \nu_{\dot{q}} \tag{4.26}$$

$$\nu_{\dot{q}} = A_{1,:}\mathbf{x} + b_1 s\delta_{th} \tag{4.27}$$

Eqn 4.27 is inverted to obtain the control-law

$$s\delta_{th} = \frac{1}{b_1}\{\nu_{\dot{q}} - A_{1,:}\mathbf{x}\} \tag{4.28}$$

In terms of stability derivatives and pseudo control this is

$$s\delta_{th} = \frac{1}{2\,\Lambda_{\delta_{th}}}\{\nu_{\dot{q}} - \Lambda_q q - \Lambda_u u - \Lambda_w w\} \tag{4.29}$$

We now have the freedom to design the signal $\nu_{\dot{q}}$ to provide desired closed loop responses. In a model reference setup we design

$$\nu_{\dot{q}} = \dot{q}_c + \alpha_1(q_c - q) + \alpha_0(\theta_c - \theta) \tag{4.30}$$

Analysis of zero–dynamics

For the longitudinal channel, with a rate command system $\xi = q$. Based on remark 2.1, with $\boldsymbol{\xi} \equiv 0 \Rightarrow \dot{q} = q = \theta = 0$, from 4.25 obtain

$$s\delta_{th} = -\frac{\Lambda_u}{2\,\Lambda_{\delta_{th}}}u - \frac{\Lambda_w}{2\,\Lambda_{\delta_{th}}}w \tag{4.31}$$

The zero dynamics may be represented by

$$\dot{\boldsymbol{\mu}} = \begin{bmatrix} X_u - \frac{X_{\delta_{th}}\Lambda_u}{\Lambda_{\delta_{th}}} & X_w - \frac{X_{\delta_{th}}\Lambda_w}{\Lambda_{\delta_{th}}} \\ Z_u - \frac{Z_{\delta_{th}}\Lambda_u}{\Lambda_{\delta_{th}}} & Z_w - \frac{Z_{\delta_{th}}\Lambda_w}{\Lambda_{\delta_{th}}} \end{bmatrix} \boldsymbol{\mu} \tag{4.32}$$

4.3.1 Analysis of output tracking dynamics

To analyze the tracking dynamics we start by considering stabilization only, $\dot{q}_c = q_c = \theta_c = 0$, then

$$\begin{aligned} \nu_{\dot{q}} &= -\alpha_1 q - \alpha_0 \theta && (4.33) \\ &\doteq -\mathbf{k}\mathbf{x} && (4.34) \end{aligned}$$

Therefore, the symmetric thrust control law becomes

$$\begin{aligned} s\delta_{th} &= \frac{1}{b_1}\{-\mathbf{k} - A_{1,:}\}\,\mathbf{x} && (4.35) \\ &\doteq -\mathbf{k}_1\mathbf{x} && (4.36) \end{aligned}$$

The longitudinal modes in closed loop are then represented by

$$A_c = A - \mathbf{b}\,\mathbf{k}_1 \qquad (4.37)$$

4.3.2 Effect of engine dynamics

Let sy_{th} be the output of the approximate engine model,

$$\begin{aligned} s\dot{x}_{th} &= -K_{eng}\, sx_{th} + K_{eng} s\delta_{thc} && (4.38) \\ sy_{th} &= sx_{th} && (4.39) \end{aligned}$$

and define $\mathbf{x}' = [\mathbf{x}\ sx_{th}]^T$. Following steps similar to section 4.2.3,

$$A_c' = \begin{bmatrix} \Lambda_q & \Lambda_u & \Lambda_w & 0 & 2\Lambda_{\delta_{th}} \\ X_q & X_u & X_w & 0 & 2X_{\delta_{th}} \\ Z_q & Z_u & Z_w & 0 & 2Z_{\delta_{th}} \\ 1 & 0 & 0 & 0 & 0 \\ -\frac{K_{eng}}{2\Lambda_{\delta_{th}}}(\alpha_1 + \Lambda_q) & -\frac{K_{eng}}{2\Lambda_{\delta_{th}}}\Lambda_u & -\frac{K_{eng}}{2\Lambda_{\delta_{th}}}\Lambda_w & -\frac{K_{eng}}{2\Lambda_{\delta_{th}}}\alpha_0 & -K_{eng} \end{bmatrix} \qquad (4.40)$$

4.4 Augmentation of Yaw Channel

As an alternative to providing the pilot with handling qualities in the roll channel, it may be easier to provide an autopilot function in terms of heading selection. This is equivalent to either a yaw-rate command or heading command, depending on the type of augmentation that is desired. The lateral modes are approximated by the linear representation 4.2. where $a\delta_{th}$ represents *asymmetric* or *differential* throttle deflection, i.e. the effect of both throttle deflections.

Let $A_{2,:}$ represent the second row of A, and b_2 the second element of $\mathbf{b}$. A feedback linearizing transformation is

$$\begin{aligned} \dot{r} &= \nu_{\dot{r}} && (4.41) \\ \nu_{\dot{r}} &= A_{2,:}\mathbf{x} + b_2 a\delta_{th} && (4.42) \end{aligned}$$

Eqn 4.42 is inverted to obtain the control-law

$$a\delta_{th} = \frac{1}{2\Omega_{\delta_{th}}}\{\nu_{\dot{r}} - A_{2,:}\mathbf{x}\} \tag{4.43}$$

In terms of stability derivatives and pseudo control this is

$$a\delta_{th} = \frac{1}{2\Omega_{\delta_{th}}}\{\nu_{\dot{r}} - \Omega_p p - \Omega_r r - \Omega_v v\} \tag{4.44}$$

We now have the freedom to design the signal $\nu_{\dot{r}}$ to provide desired closed loop responses. In a model reference setup we design

$$\nu_{\dot{r}} = \dot{r}_c + \alpha_1(r_c - r) + \alpha_0(\psi_c - \psi) \tag{4.45}$$

4.4.1 Analysis of zero–dynamics

This analysis is based on linearized expressions 4.2. Neglecting the effects of $\Gamma_{\delta_{th}}$ and Y_{dth}, it can be shown that the equations for $\dot{p}$ and $\dot{v}$ are in so–called *normal form*, Eqn2.13. Therefore, choose

$$\boldsymbol{\eta} = [p\ v]^T \tag{4.46}$$

From an analysis similar to section 4.2, with $\boldsymbol{\xi} \equiv 0 \Rightarrow r \equiv 0$, obtain

$$a\delta_{th} = -\frac{\Omega_p}{2\,\Omega_{\delta_{th}}}p - \frac{\Omega_v}{2\,\Omega_{\delta_{th}}}v \tag{4.47}$$

The zero dynamics are represented by

$$\dot{\boldsymbol{\eta}} = \begin{bmatrix} \Gamma_p - \frac{\Gamma_{\delta_{th}}}{\Omega_{\delta_{th}}}\Omega_p & \Gamma_v - \frac{\Gamma_{\delta_{th}}}{\Omega_{\delta_{th}}}\Omega_v \\ Y_p - \frac{Y_{\delta_{th}}}{\Omega_{\delta_{th}}}\Omega_p & Y_v - \frac{Y_{\delta_{th}}}{\Omega_{\delta_{th}}}\Omega_v \end{bmatrix} \boldsymbol{\eta} \tag{4.48}$$

4.4.2 Closed loop tracking dynamics

To analyze the zero-dynamics we start by considering stabilization only, $\dot{r}_c = r_c = \psi_c = 0$, then

$$\nu_{\dot{r}} = -\alpha_1 r - \alpha_0 \psi \tag{4.49}$$

$$\doteq -\mathbf{k}\mathbf{x} \tag{4.50}$$

Therefore, the differential thrust control law becomes

$$a\delta_{th} = \frac{1}{2\Omega_{\delta_{th}}}\{-\mathbf{k} - A_{2,:}\}\,\mathbf{x} \tag{4.51}$$

$$\doteq -\mathbf{k}_1\mathbf{x} \tag{4.52}$$

The lateral modes in the closed loop are then represented by

$$A_c = A - \mathbf{b}\,\mathbf{k}_1 \tag{4.53}$$

Example 4.4 The closed loop matrix A_c for the MD–11 has the following eigenvalues

$$-0.780 \; 0.008 \pm 0.366i \quad -0.400 \pm 0.301i \tag{4.54}$$

4.4.3 Effect of Engine Dynamics

Similar to section 4.2, the effect of the engine dynamics on the closed loop stabilization may be captured as

$$A_c' = \begin{bmatrix} \Gamma_p & \Gamma_r & \Gamma_v & 0 & 0 & 2\,\Gamma_{\delta_{th}} \\ \Omega_p & \Omega_r & \Omega_v & 0 & 0 & 2\,\Omega_{\delta_{th}} \\ Y_p & Y_r & Y_v & 0 & 0 & 2\,Y_{\delta_{th}} \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ -\frac{K_{eng}\Omega_p}{2\Omega_{\delta_{th}}} & -\frac{K_{eng}}{2}\left(\frac{\alpha_1}{\Omega_{\delta_{th}}} + \frac{\Omega_r}{\Omega_{\delta_{th}}}\right) & -\frac{K_{eng}\Omega_v}{2\Omega_{\delta_{th}}} & 0 & -\frac{K_{eng}\alpha_0}{2\Omega_{\delta_{th}}} & -K_{eng} \end{bmatrix} \tag{4.55}$$

For the choice of controller gains we may look at a simple root locus analysis where we select the ratio K_p/K_d and vary K_d. See the representation in Fig 4.1. The roll channel specifically is presented in section 5.4, Fig. 5.3.

Table 5.1 *Pole Locations in Roll Channel*

	open loop	*original MD-11 PCA*	*FBL,* $k_d = 1.0$ $k_p/k_d = 2.0$ $k_i/k_d = 1.0$
dutch roll	*-0.08±0.69i*	*-0.291±0.38i*	
spiral & rolling mode	*0.026, -0.80*	*-0.13±0.09i*	
zero dynamics			
engine	*-2.00*	*-2.15*	*-2.15*
controller		*-0.35*	
FBL poles			*-1.84±0.39i*
			-0.55±0.62i

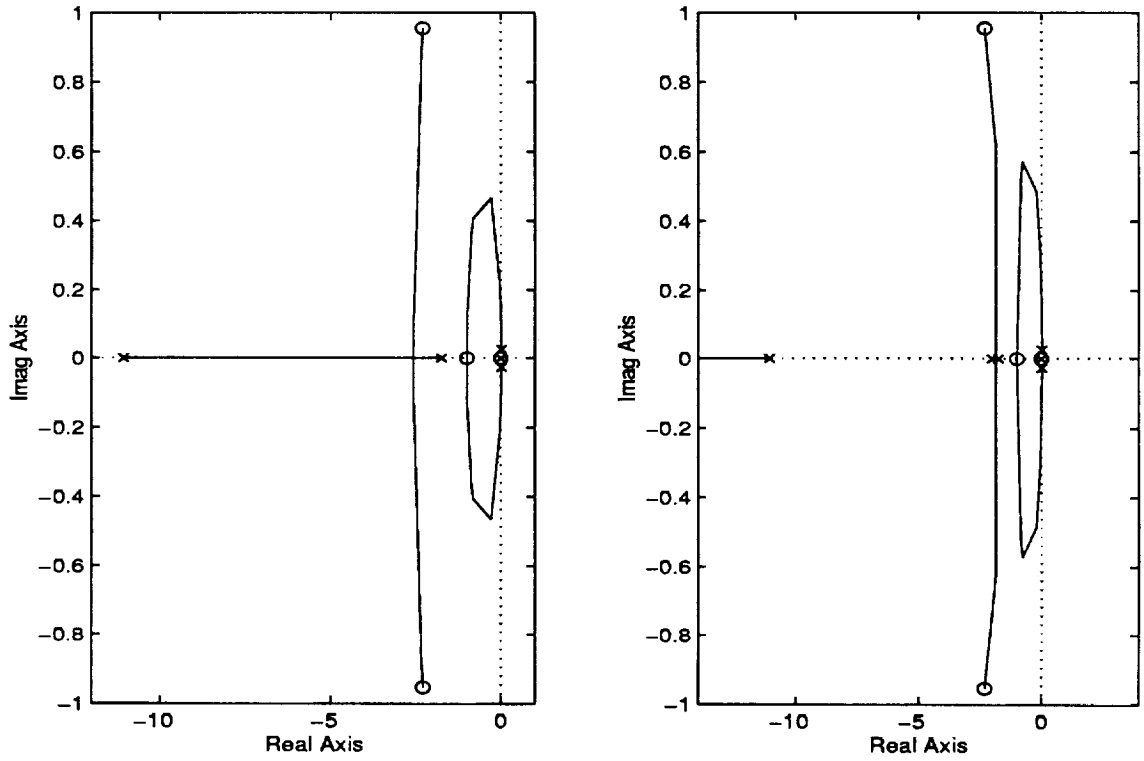

Figure 5.1: Roll channel root locus, with (left) and without (right) engine dynamics.

5.3 Pitch rate augmentation

The zero dynamics are represented by Eqn (4.32), which for the MD–11, consist of two real modes with poles at

$$-0.1463 \text{ and } -0.4581 \tag{5.3}$$

The zero dynamics are cancelled, and are therefore 'not visible', in the feedback linearizing control design. The loop dynamics of the linear controller and plant can be approximated

as

$$k_d \frac{(s + k_p/k_d)}{(s^2 + k_d s + k_p)} \tag{5.4}$$

This representation can now be used in a root locus, which will also reveal the effect of the engine dynamics. Based on a given ratio k_p/k_d, the gain k_d can be used as the varying gain for analysis. Fig 5.2 shows the closed loop results for the nominal MD–11. The left plot shows the effect of the feedback linearization, i.e. the open loop modes are cancelled, and in its place is a double integrator pole at the origin. The zero dynamics show up as cancelled poles. The controller zero at $s = -k_p/k_d$ draws these integrator poles well into the left half plane. The location of the closed loop poles is determined by k_d. However, the useful choices of k_d are limited by the engine dynamics. The effect that the engine dynamics have on the possible pole locations is indicated in the right plot. Table 5.2 gives the pole locations for the original PCA design, and also some examples of pole locations with the current feedback linearizing (FBL) setup.

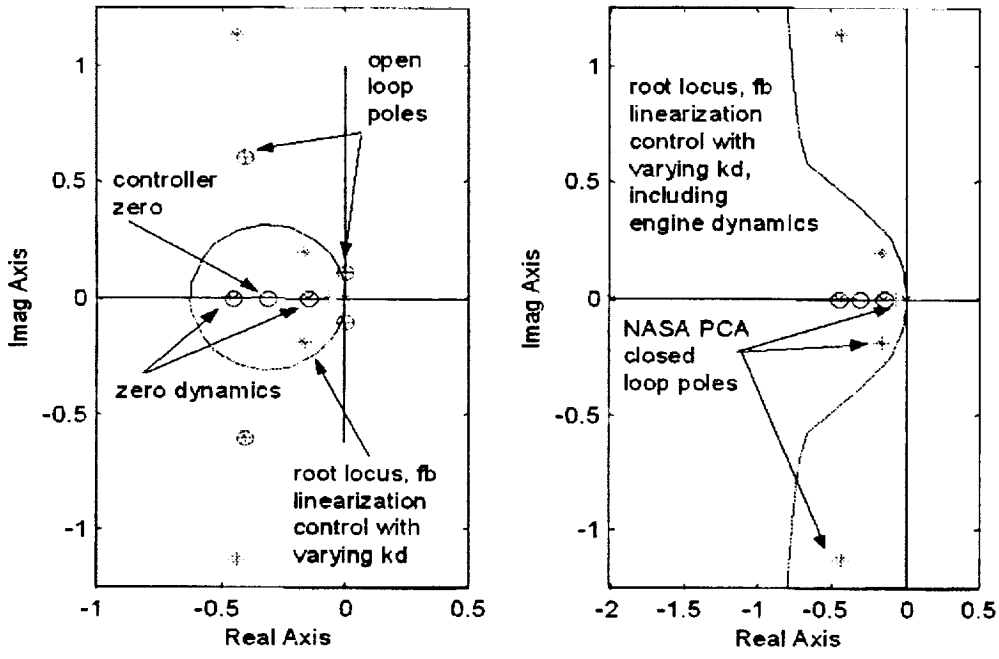

Figure 5.2: Root loci for the longitudinal MD–11 model, without (left) and with (right) the engine dynamics.

Table 5.2 *Pole Locations in Longitudinal Channel*

	open loop	*original MD-11 PCA*	*FBL,* $k_p = 0.36$ $k_d = 0.72$	*FBL,* $k_p = 0.26$ $k_d = 0.83$	*FBL,* $k_p = 0.50$ $k_d = 1.20$
phugoid	*-0.01±0.11i*	*-0.17±0.19i*	*-0.69±0.32i*	*-0.66± 0.58i*	*-0.47±1.09i*
short pd.	*-0.41±0.60i*	*-0.44±1.13i*			
zero			*-0.45*	*-0.45*	*-0.45*
dynamics			*-0.15*	*-0.15*	*-0.15*
engine	*-2.00*	*-2.15*	*-0.63*	*-0.68*	*-1.07*

5.4 Yaw channel augmentation

For the MD–11 [4], the zero dynamics when considering a yaw rate command system are given by Eqn (4.48), which has eigenvalues

$$-0.507 \quad -0.289 \tag{5.5}$$

In design of the controller, the effect of the engine dynamics, Eqn 4.55 ,on the tracking performance may be captured in a simple root locus analysis as indicated in Fig 5.3. We set the ratio $\alpha_0/\alpha_1 = K_p/K_d$ and we vary K_d. Based on the results we choose $\alpha_0/\alpha_1 = K_p/K_d = 0.1$. In Fig 5.3, K_d is varried from $0 \rightarrow 0.3$. The better response is obtained from $K_d = 0.25$ and thus $K_p = 0.025$. The results are poles located at

$$-0.184 \pm 0.175i, \quad -0.233, \quad -0.223 \tag{5.6}$$

This provides a response to a commanded heading with $\zeta = 0.72$, $\omega_n = 0.25$ and $T_{5\%} \approx 16\,sec.$

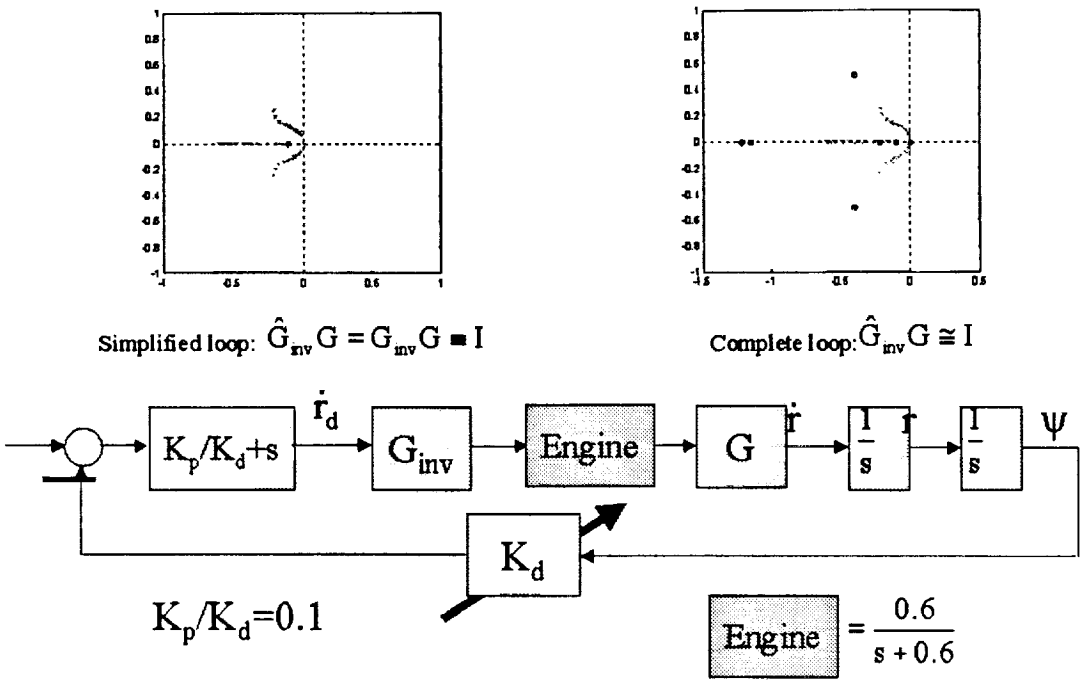

Figure 5.3: Yaw channel, simple root locus analysis of the effect of engine dynamics.

6 Stone Soup Simulator Implementation

6.1 Dynamic Inversion for Control Law Design

In section 3.1 we chose pitch rate/symmetric thrust and roll rate/asymmetric thrust for input/output linearization with *relative degrees* of, $r = 1$ and $r = 2$ respectively. To implement these results using dynamic inversion, we linearize the aircraft dynamics, design controllers for the symmetric and asymmetric control signals then transform them to individual engine commands.

6.1.1 Linearized Aircraft Model

We can expand the rigid body equations of motion using the velocity vector, $\mathbf{V} = [u \quad v \quad w]^T$, the angular rate vector, $\omega = [p \quad q \quad r]^T$, and the attitude vector, $\mathbf{\Phi} = [\phi \quad \theta \quad \psi]^T$, to make up our state vector in the body reference frame [16]

$$\mathbf{x} = [u \quad v \quad w \quad p \quad q \quad r \quad \phi \quad \theta \quad \psi]^T \tag{6.1}$$

The external force and torque vectors in body axis frame are defined

$$\mathbf{F} = [X \quad Y \quad Z]^T \tag{6.2}$$

$$\mathbf{T} = [L \quad M \quad N]^T \tag{6.3}$$

When necessary, the subscript E will be used to identify propulsive (engine) terms. Using the following simplifying nomenclature that utilizes some of the McFarland inertia constants (c_i) given in Appendix A.

$$\Gamma = c_3 L + c_4 N \tag{6.4}$$
$$\Lambda = c_7 M \tag{6.5}$$
$$\Omega = c_4 L + c_9 N \tag{6.6}$$

The force equations are

$$\dot{u} = rv - qw - g_0' sin\theta + (X + X_E)/m \tag{6.7}$$
$$\dot{v} = pw - ru + g_0' sin\phi cos\theta + Y/m \tag{6.8}$$
$$\dot{w} = qu - pv + g_0' cos\phi cos\theta + Z/m \tag{6.9}$$

where g_0' is the gravity constant. The moment equations are

$$\dot{p} = c_1 rq + c_2 pq + \Gamma + c_3 L_E + c_4 N_E \tag{6.10}$$
$$\dot{q} = c_5 pr - c_6 (p^2 - r^2) + \Lambda + c_7 M_E \tag{6.11}$$
$$\dot{r} = c_8 pq - c_2 rq + \Omega + c_4 L_E + c_9 N_E \tag{6.12}$$

The kinematic equations are

$$\dot{\phi} = p + (qsin\phi + rcos\phi)tan\theta \tag{6.13}$$
$$\dot{\theta} = qcos\phi - rsin\phi \tag{6.14}$$
$$\dot{\psi} = (qsin\phi + rcos\phi)/cos\theta \tag{6.15}$$

The input signals are the symmetric (M_E) and asymmetric (N_E) external torques due to propulsion. Noting that the symmetric torque component will also contribute a longitudinal force component, we can replace X_E with M_E/z_e where z_e is the engine vertical offset. For this class of aircraft, we will further assume a plane of symmetry about the body x,z plane. The full nonlinear system of equations are

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \\ \dot{p} \\ \dot{q} \\ \dot{r} \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} rv - qw - g_0' sin\theta & + & \bar{X} \\ pw - ru + g_0' sin\phi cos\theta & + & \bar{Y} \\ qu - pv + g_0' cos\phi cos\theta & + & \bar{Z} \\ c_1 rq + c_2 pq & + & \Gamma \\ c_5 pr - c_6 (p^2 - r^2) & + & \Lambda \\ c_8 pq - c_2 rq & + & \Omega \\ p + tan\theta(qsin\phi + rcos\phi) & & \\ qcos\phi - rsin\phi & & \\ (qsin\phi + rcos\phi)/cos\theta & & \end{bmatrix} + \begin{bmatrix} 1/z_e m & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & c_4 \\ c_7 & 0 \\ 0 & c_9 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} M_E \\ N_E \end{bmatrix} \tag{6.16}$$

The bar notation indicates normalization by mass of the vehicle (i.e. $\bar{X} = X/m$). Using a first order Taylor's series to linearize about a nominal operating point

$$\mathbf{x_0} = [u_0 \quad v_0 \quad w_0 \quad p_0 \quad q_0 \quad r_0 \quad \phi_0 \quad \theta_0 \quad \psi_0]^T \tag{6.17}$$

yields the familiar linearized equations of motion

$$\begin{aligned}
\Delta\dot{u} &\approx \bar{X}_u\Delta u + (\bar{X}_v + r_0)\Delta v + (\bar{X}_w - q_0)\Delta w \\
&+ \bar{X}_p\Delta p + (\bar{X}_q - w_0)\Delta q + (\bar{X}_r + v_0)\Delta r \\
&- g_0' cos\theta_0\Delta\phi + \frac{\Delta M_E}{z_e m} \qquad (6.18) \\
\Delta\dot{v} &\approx (\bar{Y}_u - r_0)\Delta u + \bar{Y}_v\Delta v + (\bar{Y}_w + p_0)\Delta w \\
&+ (\bar{Y}_p + w_0)\Delta p + \bar{Y}_q\Delta q + (\bar{Y}_r - u_0)\Delta r \\
&+ g_0' cos\phi_0 cos\theta_0\Delta\phi - g_0' sin\phi_0 sin\theta_0\Delta\theta \qquad (6.19) \\
\Delta\dot{w} &\approx \frac{(\bar{Z}_u + q_0)}{1-\bar{Z}_{\dot{w}}}\Delta u + \frac{(\bar{Z}_v - p_0)}{1-\bar{Z}_{\dot{w}}}\Delta v + \frac{\bar{Z}_w}{1-\bar{Z}_{\dot{w}}}\Delta w \\
&+ \frac{(\bar{Z}_p - v_0)}{1-\bar{Z}_{\dot{w}}}\Delta p + \frac{(\bar{Z}_q + u_0)}{1-\bar{Z}_{\dot{w}}}\Delta q + \frac{\bar{Z}_r}{1-\bar{Z}_{\dot{w}}}\Delta r \\
&- \frac{g_0' sin\phi_0 cos\theta_0}{1-\bar{Z}_{\dot{w}}}\Delta\phi - \frac{g_0' cos\phi_0 sin\theta_0}{1-\bar{Z}_{\dot{w}}}\Delta\theta \qquad (6.20) \\
\Delta\dot{p} &\approx \Gamma_u\Delta u + \Gamma_v\Delta v + \Gamma_w\Delta w \\
&+ \Gamma_p\Delta p + \Gamma_q\Delta q + \Gamma_r\Delta r + c_4\Delta N_E \qquad (6.21) \\
\Delta\dot{q} &\approx h_1\Delta u + h_2\Delta v + h_3\Delta w + h_4\Delta p + h_5\Delta q + h_6\Delta r \\
&+ h_7\Delta\phi + h_8\Delta\theta + c_7\Delta M_E \qquad (6.22) \\
\Delta\dot{r} &\approx \Omega_u\Delta u + \Omega_v\Delta v + \Omega_w\Delta w \\
&+ \Omega_p\Delta p + \Omega_q\Delta q + \Omega_r\Delta r + c_9\Delta N_E \qquad (6.23)
\end{aligned}$$

where the subscripted notation indicates partial derivatives (i.e. $\frac{\partial M}{\partial z} = M_z$) and the coefficients h_i are

$$\begin{aligned}
h_1 &= \Lambda_u + \Lambda_{\dot{w}}\frac{\bar{Z}_u + q_0}{1-\bar{Z}_{\dot{w}}} \\
h_2 &= \Lambda_v + \Lambda_{\dot{w}}\frac{\bar{Z}_v - p_0}{1-\bar{Z}_{\dot{w}}} \\
h_3 &= \Lambda_w + \Lambda_{\dot{w}}\frac{\bar{Z}_w}{1-\bar{Z}_{\dot{w}}} \\
h_4 &= \Lambda_p + \Lambda_{\dot{w}}\frac{\bar{Z}_p - v_0}{1-\bar{Z}_{\dot{w}}} - c_5 r_0 - 2c_6 p_0 \\
h_5 &= \Lambda_q + \Lambda_{\dot{w}}\frac{\bar{Z}_q + u_0}{1-\bar{Z}_{\dot{w}}} \\
h_6 &= \Lambda_r + \Lambda_{\dot{w}}\frac{\bar{Z}_r}{1-\bar{Z}_{\dot{w}}} + c_5 p_0 - 2c_6 r_0 \\
h_7 &= -\Lambda_{\dot{w}}\frac{g_0' sin\phi_0 cos\theta_0}{1-\bar{Z}_{\dot{w}}} \\
h_8 &= -\Lambda_{\dot{w}}\frac{g_0' cos\phi_0 sin\theta_0}{1-\bar{Z}_{\dot{w}}}
\end{aligned}$$

6.1.2 Symmetric Control

Equations (6.18) and (6.22) show that a symmetric thrust input will result in changes in longitudinal angular and translational acceleration. The dominant effect will is a change in pitch attitude followed by the secondary effect of a change in longituinal velocities. The focus of this section is on the control of pitch attitude and since we are limited to one symmetric control effector, it is reserved for pitch control exclusively. Therefore, we chose to ignore the effect of symmetric thrust control on aircraft speed. Note, however, that speed stability as described in section 4 is inherent to most air-transport aircraft. The linearized pitching moment equation, (6.22), confirms that pitch rate/symmetrical thrust is of *realtive degree*, $r = 1$, since a symmetric thrust change will result in a significant change in the pitch rate. Using this equation, the application of Theorem 2.1 is straignt forward. The symmetric control transformation is

$$\Delta\dot{q} = \nu_{\dot{q}} \tag{6.24}$$

where

$$\begin{aligned} \nu_{\dot{q}} &= h_1\Delta u + h_2\Delta v + h_3\Delta w + h_4\Delta p + h_5\Delta q + h_6\Delta r \\ &+ h_7\Delta\phi + h_8\Delta\theta + c_7\Delta M_E \end{aligned} \tag{6.25}$$

The partial feedback linearizing symmetric control law becomes

$$\begin{aligned} \Delta M_E &= \frac{1}{c_7}[\nu_{\dot{q}} - h_1\Delta u - h_2\Delta v - h_3\Delta w - h_4\Delta p - h_5\Delta q - h_6\Delta r \\ &- h_7\Delta\phi - h_8\Delta\theta] \end{aligned} \tag{6.26}$$

This control law allows use to create the desired response through appropriate construction of the psuedo control $\nu_{\dot{q}}$.

6.1.3 Asymmetric Control

Equations (6.21) and (6.23) show that an asymmetric thrust input will result in changes in roll rate and yaw rate proportional to the values of c_4 and c_9 repectively. Because of the magnitude of c_4, trying to implement equation (6.21) would result in a high gain controller leading to control saturation issues. For input/output linearization control of roll rate, we differentiate (6.21) again. The second derivative of roll rate is significantly effected by asymmetrical control through c_9 and confirms that roll rate/asymmetrical thrust is of approximate *relative degree*, $r = 2$. Thus, differentiating equation (6.21) again to obtain the second derivative of roll rate

$$\ddot{p} = \Gamma_u\Delta\dot{u} + \Gamma_v\Delta\dot{v} + \Gamma_w\Delta\dot{w} + \Gamma_p\Delta\dot{p} + \Gamma_q\Delta\dot{q} + \Gamma_r\Delta\dot{r} \tag{6.27}$$

Substituting the results from the linearized equations of motion into the above equation, we get the linear approximation for the derivative of the rolling moment equation

$$\begin{aligned} \ddot{p} &\approx k_1\Delta u + k_2\Delta v + k_3\Delta w + k_4\Delta p + k_5\Delta q + k_6\Delta r + k_7\Delta\phi + k_8\Delta\theta \\ &+ (\frac{\Gamma_u}{z_e m} + c_7\Gamma_q)\Delta M_E + (c_4\Gamma_p + c_9\Gamma_r)\Delta N_E \end{aligned} \tag{6.28}$$

where the k_i coefficients are

$$
\begin{aligned}
k_1 &= \Gamma_u \bar{X}_u + \Gamma_v(\bar{Y}_u - r_0) + \Gamma_w \frac{\bar{Z}_u + q_0}{1 - \bar{Z}_{\dot{w}}} + \Gamma_p \Gamma_u + \Gamma_q h_1 + \Gamma_r \Omega_u \\
k_2 &= \Gamma_u(\bar{X}_v + r_0) + \Gamma_v \bar{Y}_v + \Gamma_w \frac{\bar{Z}_v - p_0}{1 - \bar{Z}_{\dot{w}}} + \Gamma_p \Gamma_v + \Gamma_q h_2 + \Gamma_r \Omega_v \\
k_3 &= \Gamma_u(\bar{X}_w - q_0) + \Gamma_v(\bar{Y}_w + p_0) + \Gamma_w \frac{\bar{Z}_w}{1 - \bar{Z}_{\dot{w}}} + \Gamma_p \Gamma_w + \Gamma_q h_3 + \Gamma_r \Omega_w \\
k_4 &= \Gamma_u \bar{X}_p + \Gamma_v(\bar{Y}_p + w_0) + \Gamma_w \frac{\bar{Z}_p - v_0}{1 - \bar{Z}_{\dot{w}}} \\
&+ \Gamma_p(\Gamma_p + c_2 q_0) + \Gamma_q h_4 + \Gamma_r(\Omega_p + c_8 q_0) \\
k_5 &= \Gamma_u(\bar{X}_q - w_0) + \Gamma_v \bar{Y}_q + \Gamma_w \frac{\bar{Z}_q + u_0}{1 - \bar{Z}_{\dot{w}}} \\
&+ \Gamma_p(\Gamma_q + c_1 r_0 + c_2 p_0) + \Gamma_q h_5 + \Gamma_r(\Omega_q + c_8 p_0 - c_2 r_0) \\
k_6 &= \Gamma_u(\bar{X}_r + v_0) + \Gamma_v(\bar{Y}_r - u_0) + \Gamma_w \frac{\bar{Z}_r}{1 - \bar{Z}_{\dot{w}}} \\
&+ \Gamma_p(\Gamma_r + c_1 q_0) + \Gamma_q h_6 + \Gamma_r(\Omega_r - c_2 q_0) \\
k_7 &= \Gamma_v g_0' cos\phi_0 cos\theta_0 - \Gamma_w g_0' \frac{sin\phi_0 cos\theta_0}{1 - \bar{Z}_{\dot{w}}} + \Gamma_q h_7 \\
k_8 &= -\Gamma_u g_0' sin\theta_0 - \Gamma_v g_0' sin\phi_0 sin\theta_0 - \Gamma_w \frac{g_0' cos\phi_0 sin\theta_0}{1 - \bar{Z}_{\dot{w}}} + \Gamma_q cos\phi_0 h_8 \\
k_9 &= \frac{\Gamma_u}{z_e m} + c_7 \Gamma_q \\
k_{10} &= (c_4 \Gamma_p + c_9 \Gamma_r
\end{aligned}
$$

Again, applying Theorem 2.1, the asymmetric control transformation is

$$\ddot{p} = \nu_{\ddot{p}} \tag{6.29}$$

where

$$
\begin{aligned}
\nu_{\ddot{p}} &= k_1 \Delta u + k_2 \Delta v + k_3 \Delta w + k_4 \Delta p + k_5 \Delta q + k_6 \Delta r + k_7 \Delta \phi + k_8 \Delta \theta \\
&+ k_9 \Delta M_E + k_{10} \Delta N_E
\end{aligned} \tag{6.30}
$$

The partial feedback linearizing asymmetric control law becomes

$$
\begin{aligned}
\Delta N_E \approx \frac{1}{k_{10}} [&\nu_{\ddot{p}} - k_1 \Delta u - k_2 \Delta v - k_3 \Delta w - k_4 \Delta p - k_5 \Delta q - k_6 \Delta r \\
&- k_7 \Delta \phi - k_8 \Delta \theta - k_9 \Delta M_E]
\end{aligned} \tag{6.31}
$$

6.1.4 Engine Control Transformation

The required change in thrust from the nominal operating point of each engine can be found from

$$\Delta \delta_1 = \frac{1}{2}\left(\frac{\Delta M_E}{z_e} + \frac{\Delta N_E}{y_e}\right) \tag{6.32}$$

$$\Delta\delta_2 = \frac{1}{2}(\frac{\Delta M_E}{z_e} - \frac{\Delta N_E}{y_e}) \tag{6.33}$$

Where z_e and y_e are the distance of each engine below the nominal C.G. and away from the plane of symmetry repectively. Finally, the throttle position for each engine is

$$\delta_1 = \delta_{1(trim)} + \Delta\delta_1 \tag{6.34}$$
$$\delta_2 = \delta_{2(trim)} + \Delta\delta_2 \tag{6.35}$$

The equations developed here can be simplified further by the appropriate choice of the equilibrium condition and dropping insignificant terms due to aircraft configuration.

6.2 Implementation

Background on the Stone Soup Simulator

The Stone Soup Simulator (SSS) originated from the Advanced Concepts Flight Simulator (ACFS) located at the NASA Ames Crew-Vehicle Systems Research Facility (CVSRF) [17]. The ACFS is a full-mission flight simulator representative of a generic commercial transport aircraft employing many advanced flight systems as well as features existing in the newest aircraft being built today. A miniature version of the ACFS (miniACFS) was developed at the NASA Ames Aviation Operations Branch. This desktop flight simulator runs on one or more SGI workstations and uses graphical displays to provide "soft" interfaces in place of hardware controls. The miniACFS provides the flight deck simulation capability for the Part-Task Laboratory and the Airspace Operations Laboratory. The Stone Soup Simulator (SSS) represents a version of the miniACFS in which proprietary portions of the software have been replaced to enable limited distribution. The "Stone Soup" name was chosen in acknowledgement of the contributions made by the research community which made the existence of this tool possible.

6.2.1 Model Inversion Assumptions

Because we did not have access to the aircraft characteristics represented by the SSS, we used a crude approximation in the aero model for control design. Most notably, the stability derivitives used for the dynamic inversion came from the Boeing 747-100 in straight and level flight (M=0.2, Sea Level) [18]. This information is included as Appendix B. To keep a consistent set of data, the B747 mass properties were also used. Typically, we choose the equilibrium condition such that all of the angular velocities, the roll angle and sideward component of linear velocity are zero. The coefficients for (6.26) simplify to

$$h_1 = \Lambda_u + \Lambda_{\dot{w}}\frac{\bar{Z}_u}{1-\bar{Z}_{\dot{w}}}$$
$$h_2 = 0$$
$$h_3 = \Lambda_w + \Lambda_{\dot{w}}\frac{\bar{Z}_w}{1-\bar{Z}_{\dot{w}}}$$

$$
\begin{aligned}
h_4 &= -\Lambda_{\dot{w}} \frac{v_0}{1-\bar{Z}_{\dot{w}}} \\
h_5 &= \Lambda_q + \Lambda_{\dot{w}} \frac{\bar{Z}_q + u_0}{1-\bar{Z}_{\dot{w}}} \\
h_6 &= 0 \\
h_7 &= 0 \\
h_8 &= -\Lambda_{\dot{w}} \frac{g_0' sin\theta_0}{1-\bar{Z}_{\dot{w}}}
\end{aligned}
$$

Then (6.26) becomes

$$
\Delta M_E \approx \frac{1}{c_7}[\nu_{\dot{q}} - h_1\Delta u - h_3\Delta w - h_4\Delta p - h_5\Delta q - h_8\Delta\theta] \tag{6.36}
$$

The coefficients for (6.31) simplify to

$$
\begin{aligned}
k_1 &= 0 \\
k_2 &= \Gamma_v\bar{Y}_v + \Gamma_p\Gamma_v + \Gamma_r\Omega_v \\
k_3 &= 0 \\
k_4 &= \Gamma_v w_0 + \Gamma_p\Gamma_p + \Gamma_r\Omega_p \\
k_5 &= 0 \\
k_6 &= -\Gamma_v u_0 + \Gamma_p\Gamma_r + \Gamma_r\Omega_r \\
k_7 &= \Gamma_v g_0' cos\theta_0 \\
k_8 &= 0 \\
k_9 &= 0 \\
k_{10} &= c_4\Gamma_p + c_9\Gamma_r
\end{aligned}
$$

and equation (6.31) becomes

$$
\Delta N_E \approx \frac{1}{k_{10}}[\nu_{\dot{p}} - k_2\Delta v - k_4\Delta p - k_6\Delta r - k_7\Delta\phi] \tag{6.37}
$$

While using the above equations in the SSS, it became apparent that more accurate information is necessary. However, since that information is currently unavailable, we chose to *adjust* the control signals. In particular, since the effectiveness of the propulsive control was underestimated, we used scaling to keep their values from saturating when using dynamic inversion without the neural network. The resultant scale factors were $\frac{1}{5}$ and $\frac{1}{125}$ for the symmetric and asymmetric controls respectively. The command filters, inversion model and neural networks were implemented as described in figure 3.1. For this implementation, we assumed that all aircraft states are available for feedback including pitch acceleration ($\dot{p}$). However, note the latter was not necessary since the pitch acceleration state could have been recreated by use of a lead-lag compensator.

6.2.2 Control Architecture

Since the SSS aircraft has only the two engines available for propulsion control, only two degrees of freedom can be controlled at one time. As stated earlier, speed stability is a

inherent in civil aircraft and thus we allow this effect to dictate the speed of the aircraft and keep the two control effectors for attitude controls (pitch and roll). The control architecture implemented in this study was separated into two parts: the fast inner-loops and the slower outer-loops (auto-pilots) as shown in Figure 6.1.

The inner loop controllers consist of a command filter, a tracking error dynamics com-

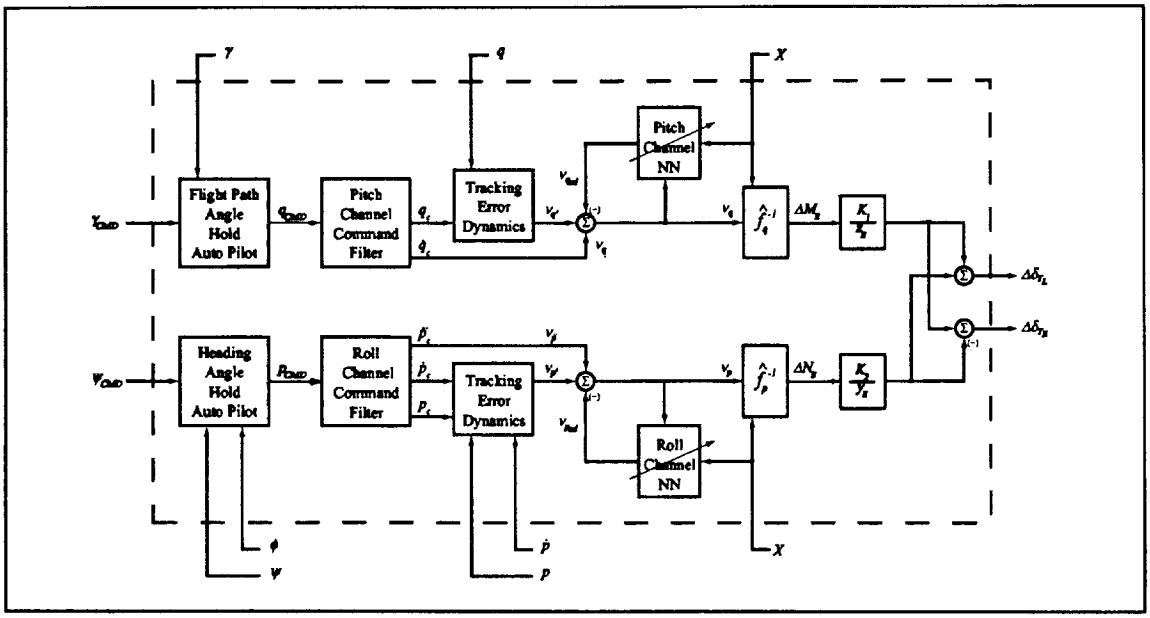

Figure 6.1: Overall Control Archetecture.

pensator, a linear in the parameters neural network and a dynamic model inversion. The command filters are used to prescribe the desired handling qualities to the variables required for control, while the tracking error compensator applies these same desired qualities to the error signals. The dynamic inversion model then transforms the psuedocontrol to the actual control. Since the inversion model is an approximation, the neural network is used to reconstruct and cancel model invesion errors. To make resonable comparisons with the existing NASA PCA controllers already included in the SSS, simple outer-loop autopilots were created for flight path angle and azimuthal (heading) angle. These autopilots have much slower time constants than the inner-loops since their primary objective is serve as simple pilot models.

Pitch Channel

The pitch channel uses a first order command filter followed by a proportional-derivative (PD) linear tracking error dynamics compensator. To match the dynamics between these, the same natural frequencies and damping ratios are used. Because the command filter is only first order, $\tau = 2\zeta\omega_n$ was used to approximate the relationship of the real (first order) and complex conjugate (second order) poles. The linear in the parameters neural network was implemented as described in Reference [3] to recontrust the model inversion errors. The

neural networks employ the following scaled input signals:

$$\begin{aligned} \bar{V} &= (V - 300)/500 \\ \bar{H} &= (H - 30000)/40000 \\ \bar{\nu}_p &= \frac{1 - e^{-\nu_p}}{1 + e^{-\nu_p}} \\ \bar{\nu}_q &= \frac{1 - e^{-\nu_q}}{1 + e^{-\nu_q}} \end{aligned} \tag{6.38}$$

where $V = \sqrt{u^2 + v^2 + w^2}$ and H is the altitude. In addition to the above signals, the NNs will require some of the states (ϕ, θ, p, q) and some of the command filter signals $(q_c, p_c, \dot{p}_c)$. The input vectors, described in Reference [3], for the pitch channel are constructed from a 0.1 bias and the above signals as follows

$$\begin{aligned} \mathbf{C}_1 &= [0.1 \quad \bar{V} \quad \bar{V}^2 \quad \bar{H}]^T \\ \mathbf{C}_2 &= [0.1 \quad \bar{\nu}_q \quad \theta \quad \phi \quad \dot{q} \quad q_c]^T \\ \mathbf{C}_3 &= [0.1 \quad \alpha \quad \beta]^T \end{aligned} \tag{6.39}$$

The outer-loop for the pitch channel is a simple flight path angle hold autopilot consisting of a simple integrator. Since we are exciting the phugoid mode to control pitch, the assumption that $\dot{\alpha}$ is negligible is warrented. The flying qualities of this controller is controlled by specifying τ_q. The command filter, tracking error compensator and flight path angle hold auto pilot are shown in Figure 6.2. Control saturation is avoided by limiting the error signal, $\bar{e}_\gamma = \gamma_{CMD} - \gamma$, by the equivalent error produced with a maximum pitch rate,

$$\bar{e}_\gamma = q_{limit}\tau_q \tag{6.40}$$

Roll Channel

The roll channel differs from the pitch channel in a number of ways. The primary deviation is due to the relative degree of the control in this channel. As stated earlier, the relative degree of the roll control is two and thus the psuedo control is related to $\ddot{p}$. In order to provide this information, the command filter needs to be at least second order. However, to soften acceleration spikes, a third order command filter with a fast pole at $s = -\alpha\zeta\omega_n$ is used. The associated transfer function is

$$\frac{p_c}{p_{CMD}(s)} = \frac{\alpha\zeta\omega_n^3}{(s + \alpha\zeta\omega_n)(s^2 + 2\zeta\omega_n s + \omega_n^2)} \tag{6.41}$$

The command filter is followed by a proportional-integral (PI) linear tracking error dynamics compensator. Again, to match the dynamics between these, the same natural frequencies and damping ratios are used in the PI compensator and the dominant poles of the command filter. Using the scaled inputs descibed for the NN in the pitch channel and the appropriate

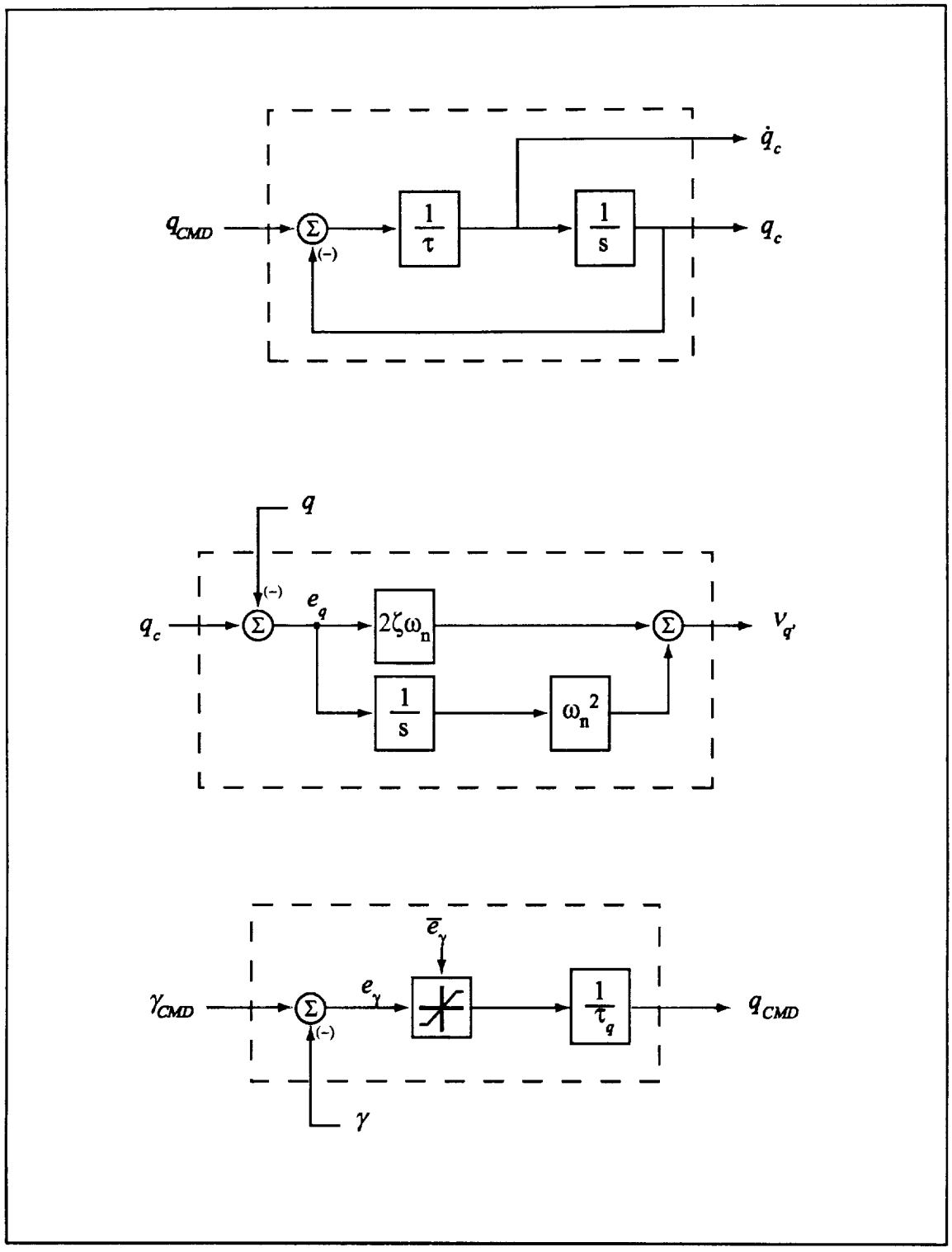

Figure 6.2: Pitch channel a) Command filter, b) Tracking error compensator, c) Flight path angle hold autopilot.

sigmoidal function, the input vectors for the roll channel NN are constructed as follows

$$\begin{aligned} \mathbf{C}_1 &= [0.1 \quad \bar{V} \quad \bar{V}^2 \quad \bar{H}]^T \\ \mathbf{C}_2 &= [0.1 \quad \bar{\nu}_p \quad \phi \quad \dot{\phi} \quad q_c \quad \dot{q}_c]^T \\ \mathbf{C}_3 &= [0.1 \quad \alpha \quad \beta]^T \end{aligned} \tag{6.42}$$

Note that C_1 and C_3 for the pitch and roll channels are identical. The outer-loop for the roll channel is an azimuthal angle (heading) hold autopilot consisting of a cascade of two simple integrators on ψ and ϕ. The gain $\frac{V_H}{g}$ is required to model this channel as two integrators, where $V_H = Vcos\theta$ is the horizontal plane aircraft velocity. Figure 6.3 shows the block diagrams of the roll channel command filter, tracking error compensator and the heading hold autopilot. The flying qualities of heading hold autopilot is straight forward. First, consider the simplified representation of this loop with the aircraft replaced by two integrators as shown in Figure 6.4. The transfer function is

$$\frac{\Psi(s)}{\Psi_{CMD}(s)} = \frac{K_\psi K_\phi}{s^2 + K_\phi s + K_\psi K_\phi} \tag{6.43}$$

with $2\zeta\omega_n = K_\phi$ and $\omega^2 = K_\psi K_\phi$. This allows us to specify the flying qualities in terms of the desired natural frequency and damping ratios. Again, this loop, by design, is made slower than the inner loop. Control saturation was avoided by limiting the error signal, $\bar{e}_\psi = \psi_{CMD} - \psi$, by an equivalent error for a maximum roll angle or a maximum roll rate propagated through the appropriate gains,

$$\bar{e}_\psi = min(\phi_{limit}, \frac{p_{limit}}{K_\phi} + \phi)\frac{g}{K_\psi V_H} \tag{6.44}$$

6.2.3 Simulation Results

The SSS was initiated in the nominal aircraft configuration and at the design operating condition (V_T=180kts, H_{abs}=2000ft, δ_{flaps}=0°, slats retracted, CG=0.2875$\bar{c}_{MAC}$) and allowed to reach a steady state. The conventional flight control surfaces were failed and, immediately following, the Intelligent Flight Control System (IFCS) was engaged. A pulsed angular rate doublet was then sent to the command filters which generated the desired command signals based on the predetermined flying qulaities. Note that separate runs were performed for the roll rate and pitch rate doublets to show the rate command attitude hold responses of the inner-loops without the outer-loop controllers.

Figure 6.5 shows the aircraft response using the model inversion only and also with neural network adaptive augmentation for a pitch rate doublet command. The command filter frequency and damping ratio were chosen to be 0.5 rads/sec and 0.8 respectively. The magnitude of the pitch rate command (about 0.25 deg/s) was selected to represent a nominal transport aircraft maneuvering rate while the pulse width keeps the aircraft from departing too far from the initial conditions. Figure 6.6 shows similar responses for the roll rate doublet command. Here, the command filter frequency and damping ratio were chosen to

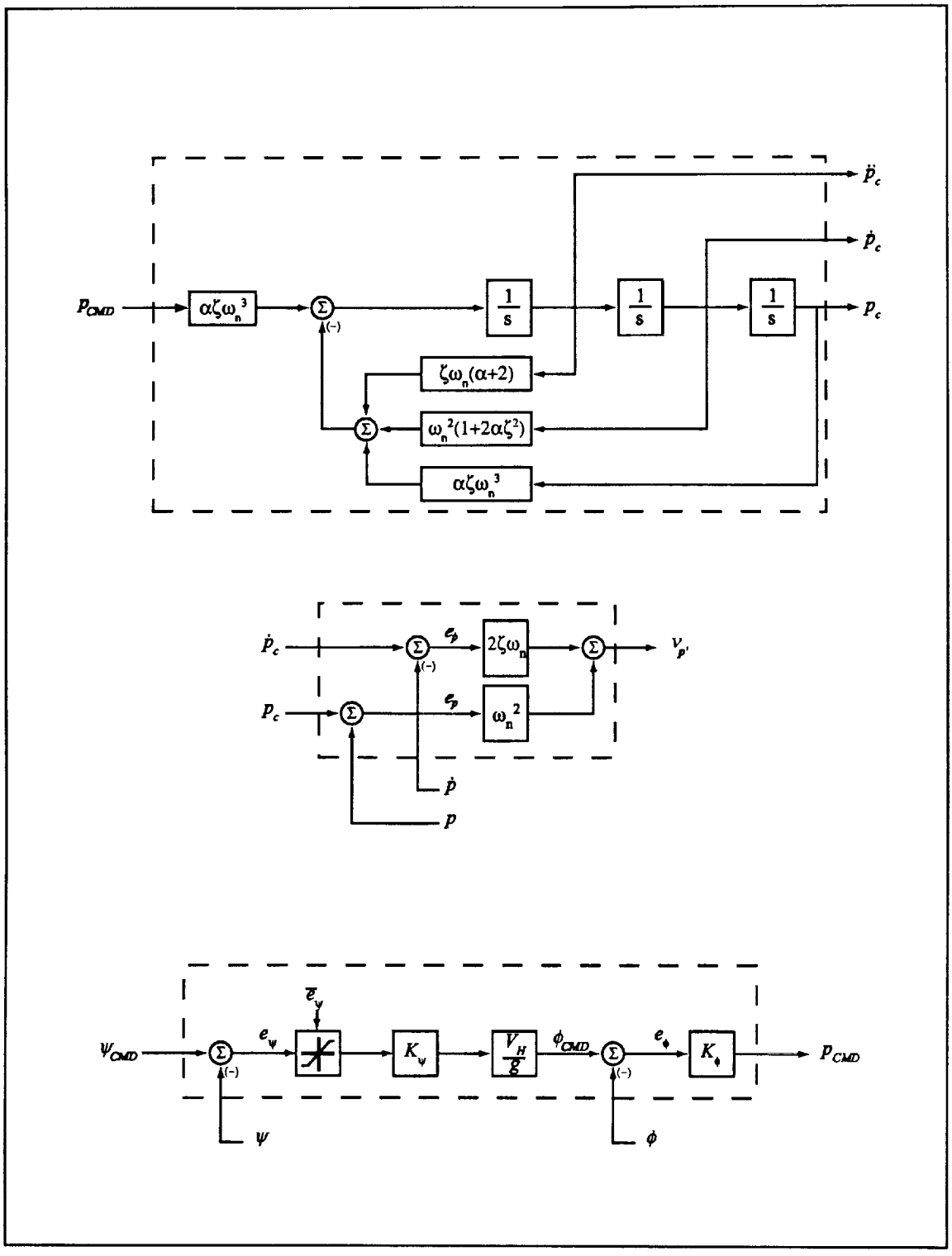

Figure 6.3: Roll channel a) Command filter, b) Tracking error compensator, c) Flight path angle hold autopilot.

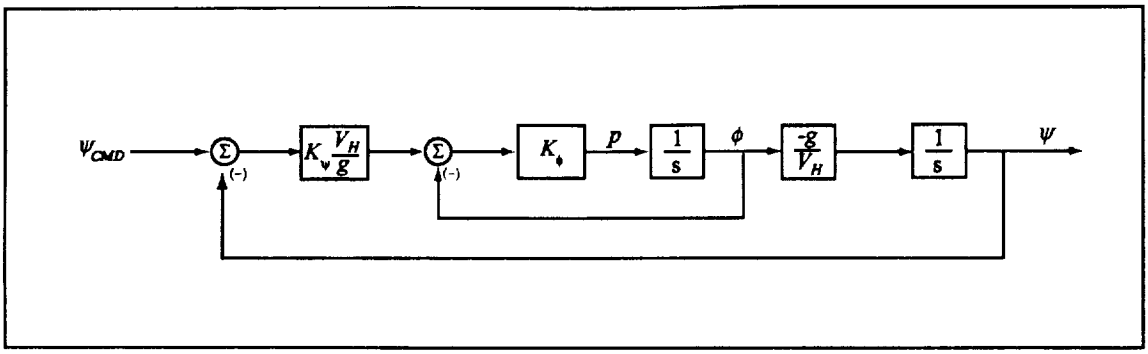

Figure 6.4: Simplified roll channel autopilot loop.

be 1.5 rads/sec and 0.8 respectively. The magnitude of the roll rate command is about 0.60 deg/s and is far below the nominal rate for a transport aircraft. This, in part, is due to the reduced roll control authority of the asymmetric engine inputs. Again, the pulse width was selected to keep the aircraft in a "proper" attitude. These plots show that the neural network is compensating for the dynamic inversion error due to the crude approximation used.

To show the stability and versatility of this control architecture, runs were made with an agressive simultaneous pitch rate and roll rate doublet for the design point as wall as "off design" points. The first case was to alter the aircraft configuration by lowering the flaps to $\delta_F = 20^o$ and extending the slates to $\delta_S = 20^o$ at the design flight condition. The second case is the nominal aircraft configuration at the cruise condition (V_T=285kts, H_{abs}=30000ft). Figure 6.7 shows the baseline aircraft response to this manuever at the nominal design point. Figure 6.8 shows the response for the aircraft configured with the flaps and slats extended at the same flight condition. Figure 6.9 shows the baseline aircraft response for a cruise condition (30,000ft altitude, Mach 0.7). Note that the magnitudes of the doublets were halved to keep the aircraft in a "proper" attitude during this high speed manuever. The oscillations exibited in the roll rate response is due to unmodeled nonlinearities in the engine dynamics, in particular, rate saturation. Figure 6.10 is the response to a changing aircraft configuration (i.e. $\delta_F = 0^o to 20^o$) while the inner-loops regulate zero pitch and roll rates compared to the dynamic model inversion only case. All these figures indicate that the neural network will compensate for aircraft configuration and flight condition changes across the flight envelope relative to the design point for the model.

To compare responses to the SSS NASA PCA, the outer-loops were engaged to hold aircraft flight path angle and heading during a flap deployment. The result is shown in Figure 6.11. The saturated control lead to integrator wind up that caused the large overshoot in pitch rate at approximately 30 seconds. Figures 6.12 and 6.13 show the vehicle responses to step flight path angle and heading commands respectively for the nominal design point also compared to the NASA PCA. Lastly, Figures 6.14 and 6.15 show the same for the up and away (cruise) condition. These figures show that the neural network architecture and the associated outer-loops provided very similar handling qualities in flight path angle and heading responses as

the NASA's SSS PCA for both fight regimes. However, the pitch rate and roll rate responses were quite different. The NASA PCA displayed more oscillations in the pitch rate and larger variances in roll rate to achieve similar results in flight path angle and heading respectively. It should be noted that the rate limits (like the inner-loop responses) and bank angle limit were reduced to accommodate the high speed case.

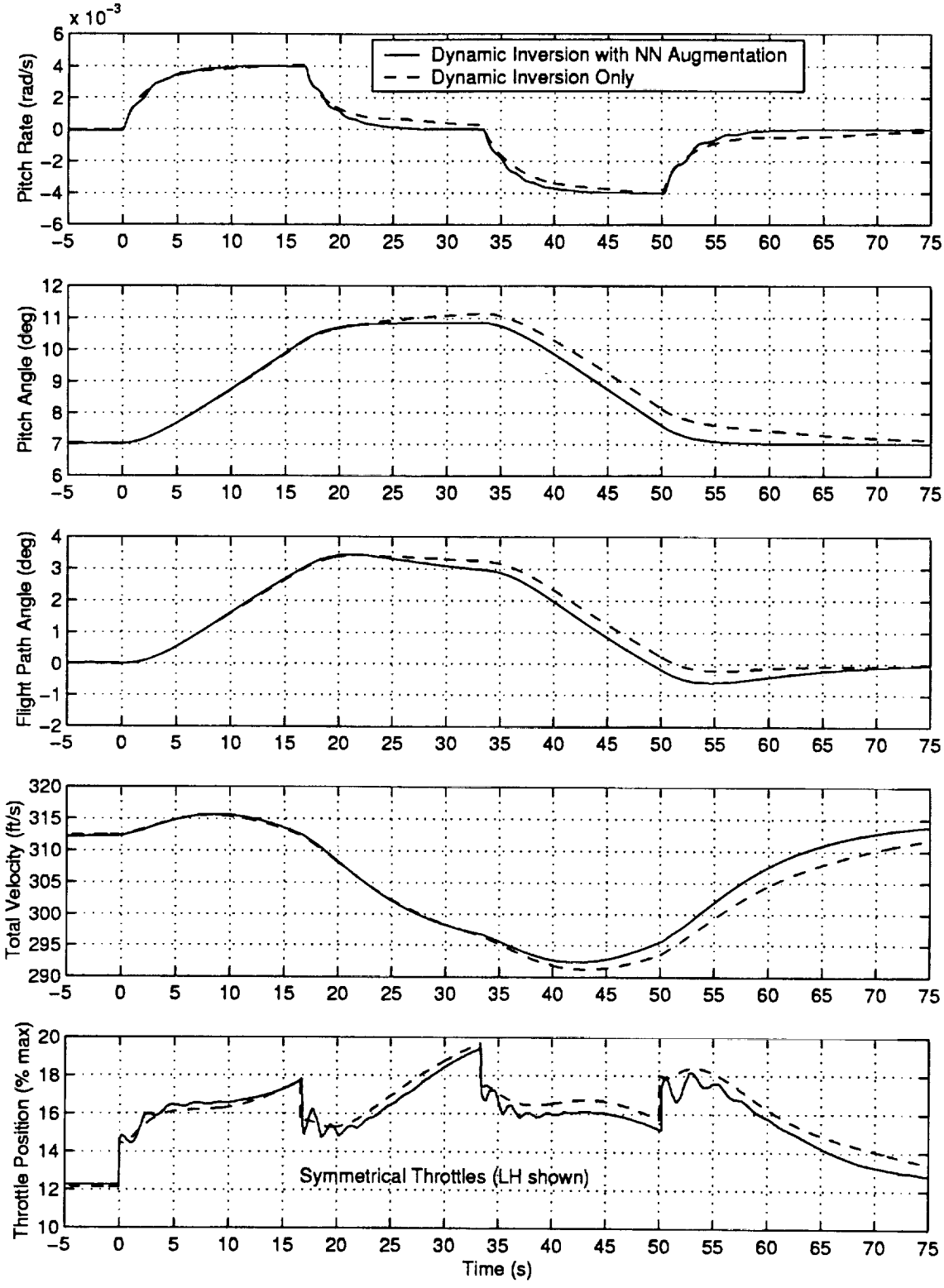

Figure 6.5: Inner-Loop Rate Command Attitude Hold Response in Pitch Channel (V=180kts, H=2,000ft, δ_F=0deg).

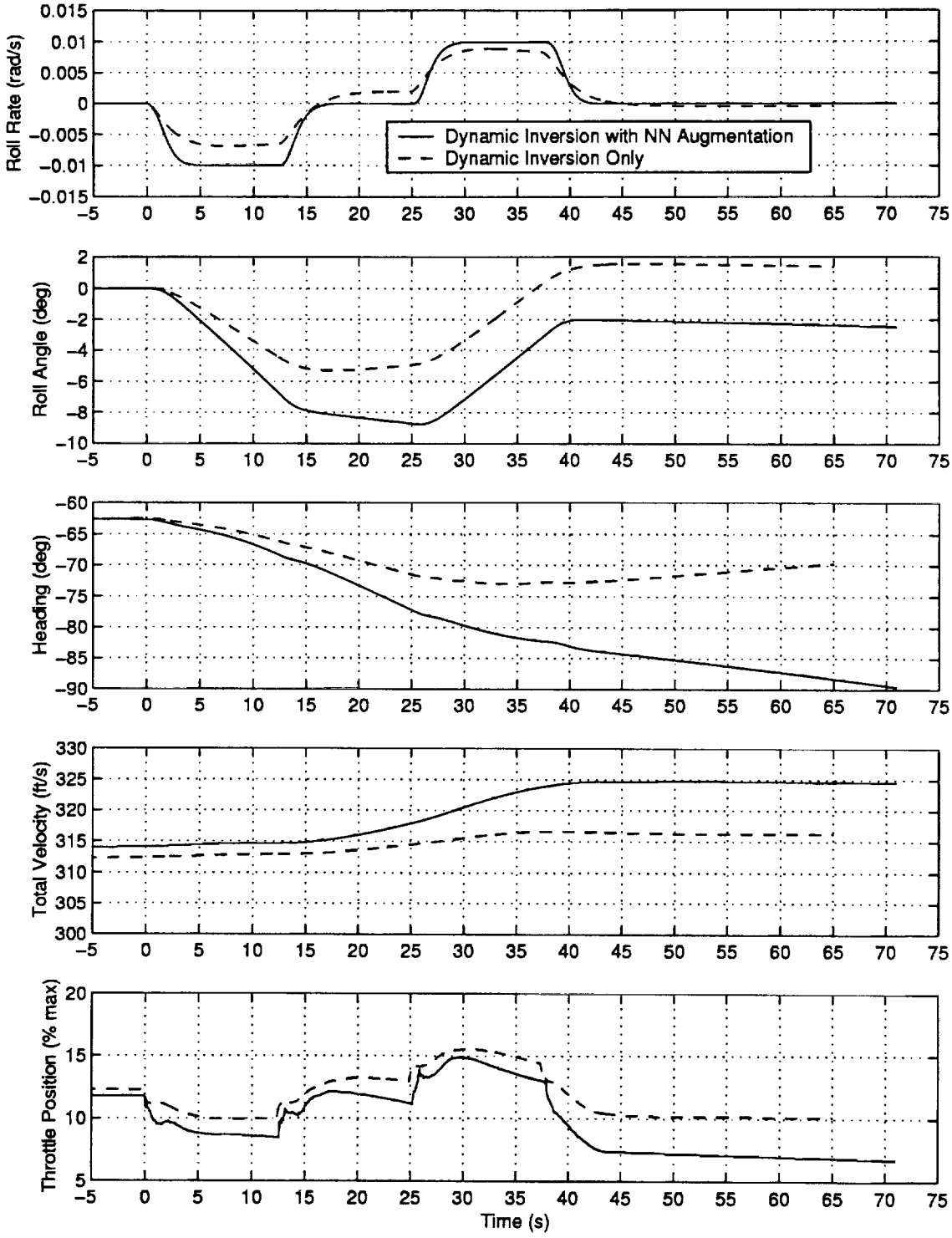

Figure 6.6: Inner-Loop Rate Command Attitude Hold Response in Roll Channel (V=180kts, H=2,000ft, δ_F=0deg).

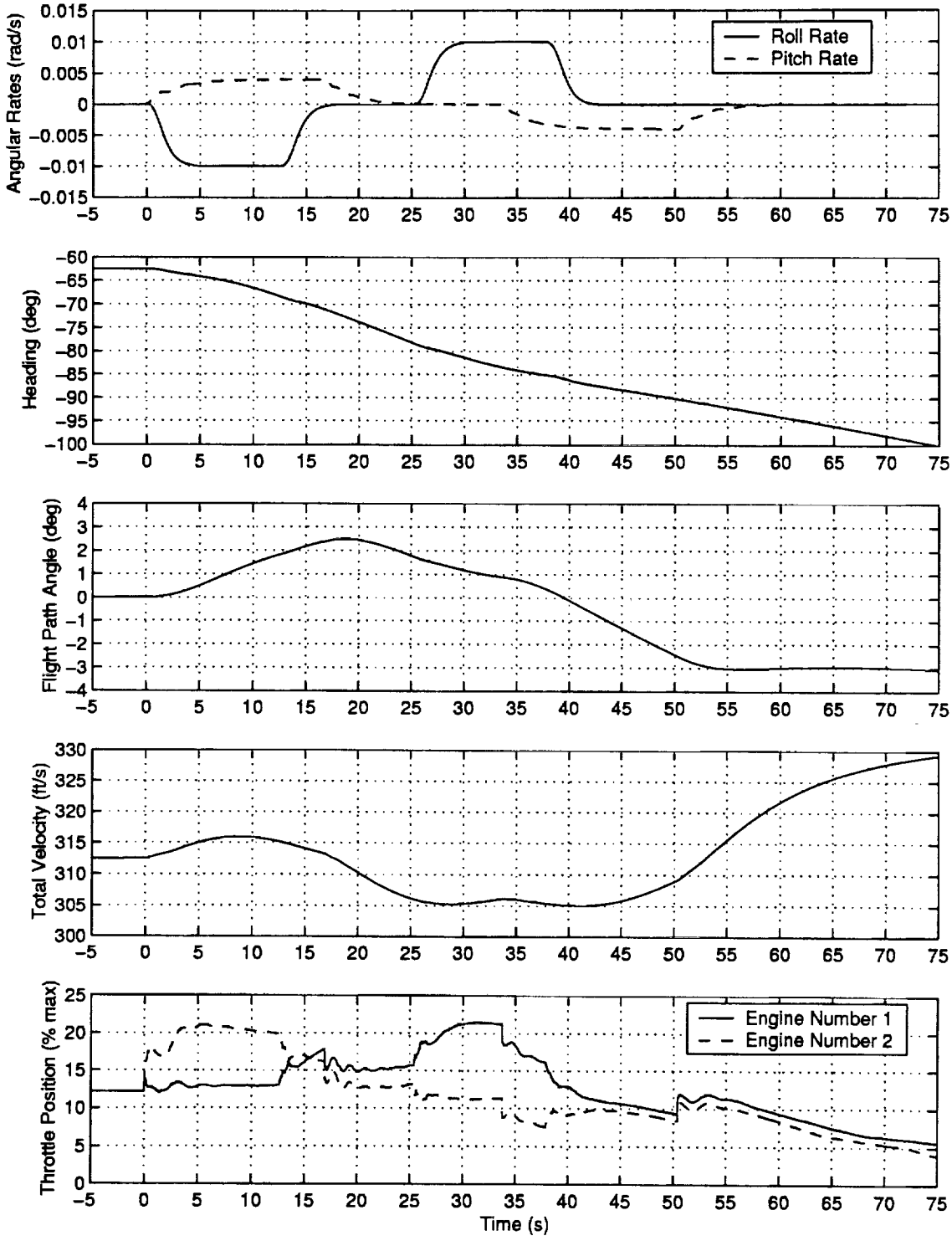

Figure 6.7: Simultaneous Roll Rate and Pitch Rate Doublets (V=180kts, H=2,000ft, δ_F=0deg).

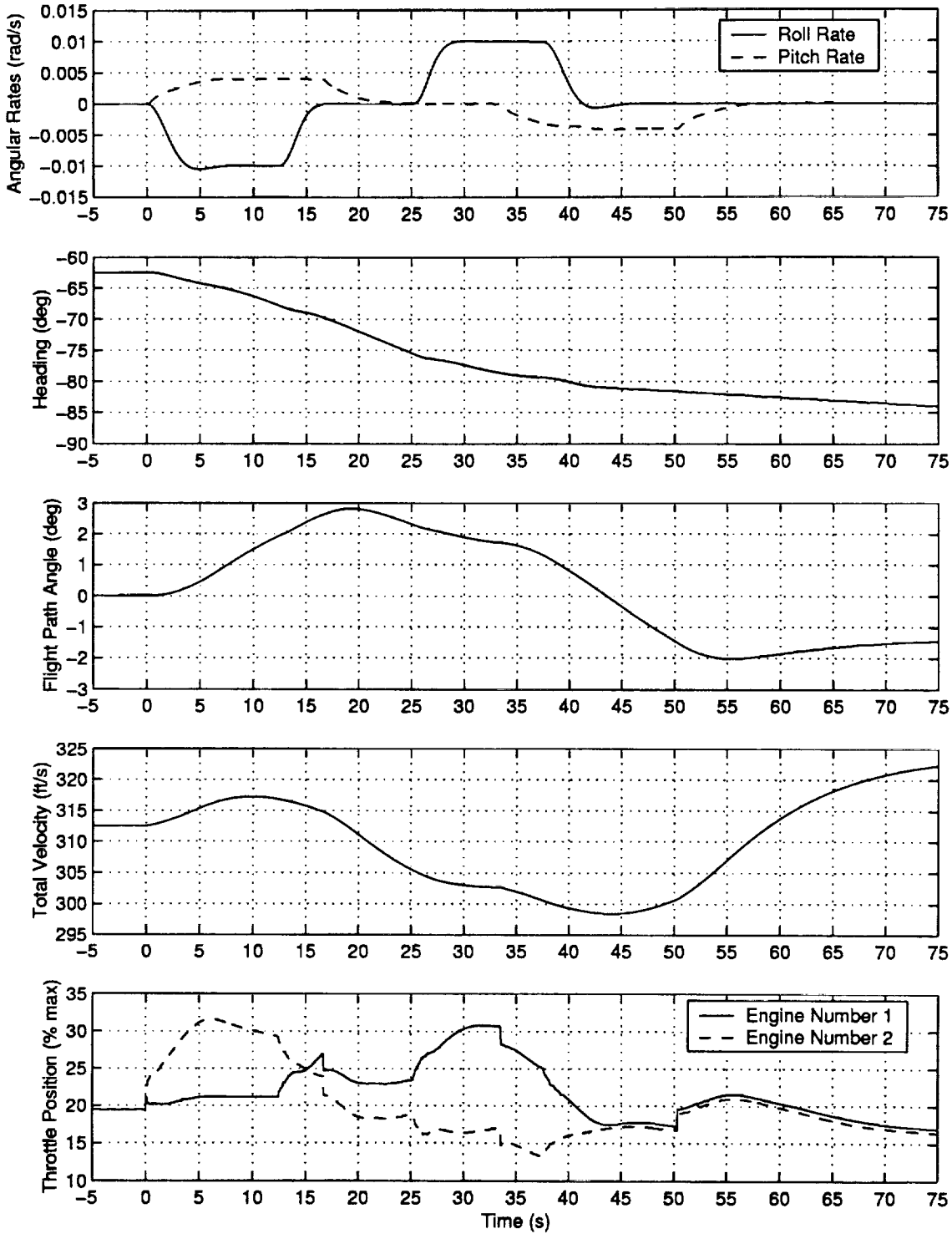

Figure 6.8: Simultaneous Roll Rate and Pitch Rate Doublets (V=180kts, H=2,000ft, δ_F=20deg).

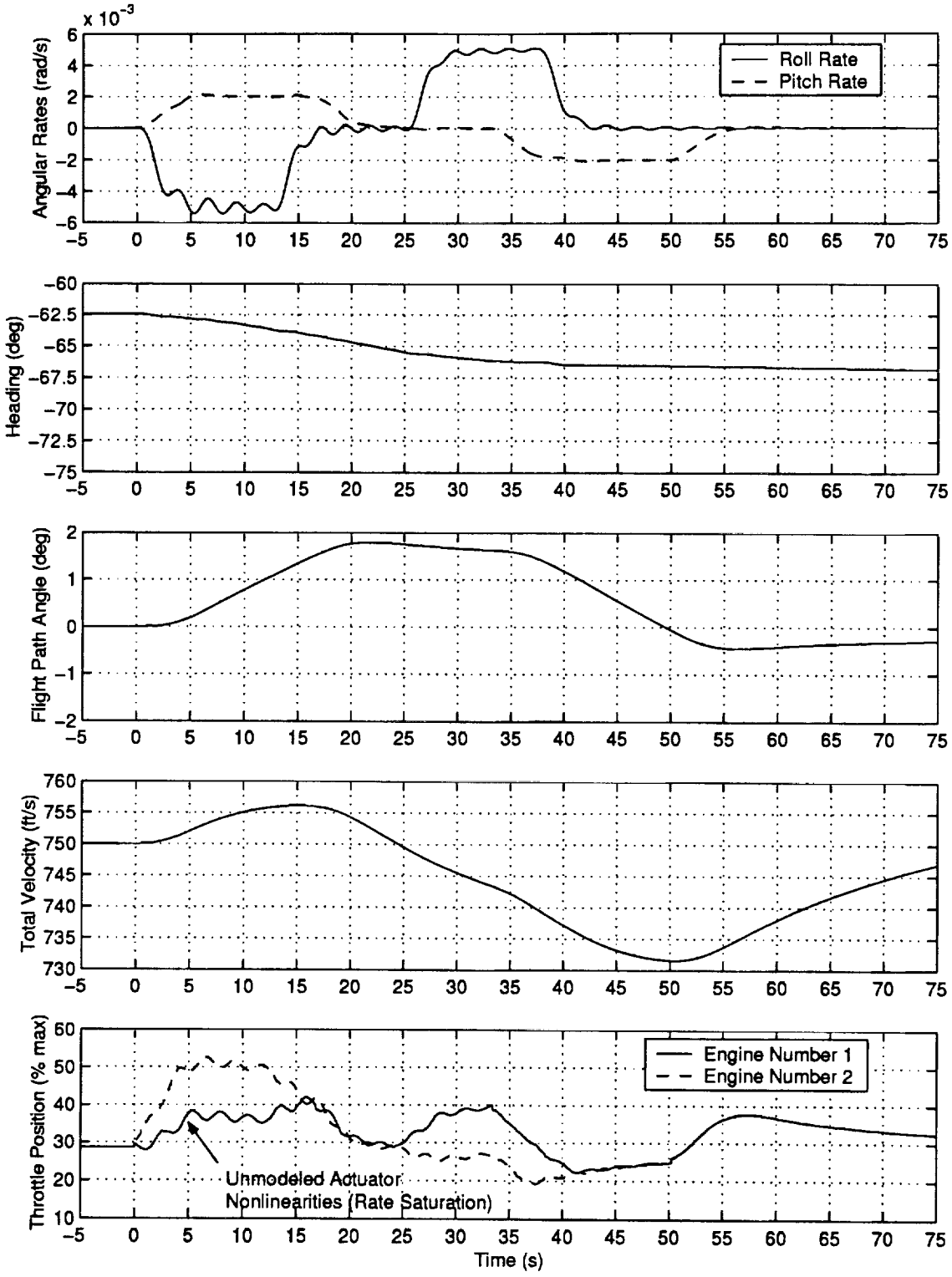

Figure 6.9: Simultaneous Roll Rate and Pitch Rate Doublets (V=285kts, H=30,000ft, δ_F=0deg).

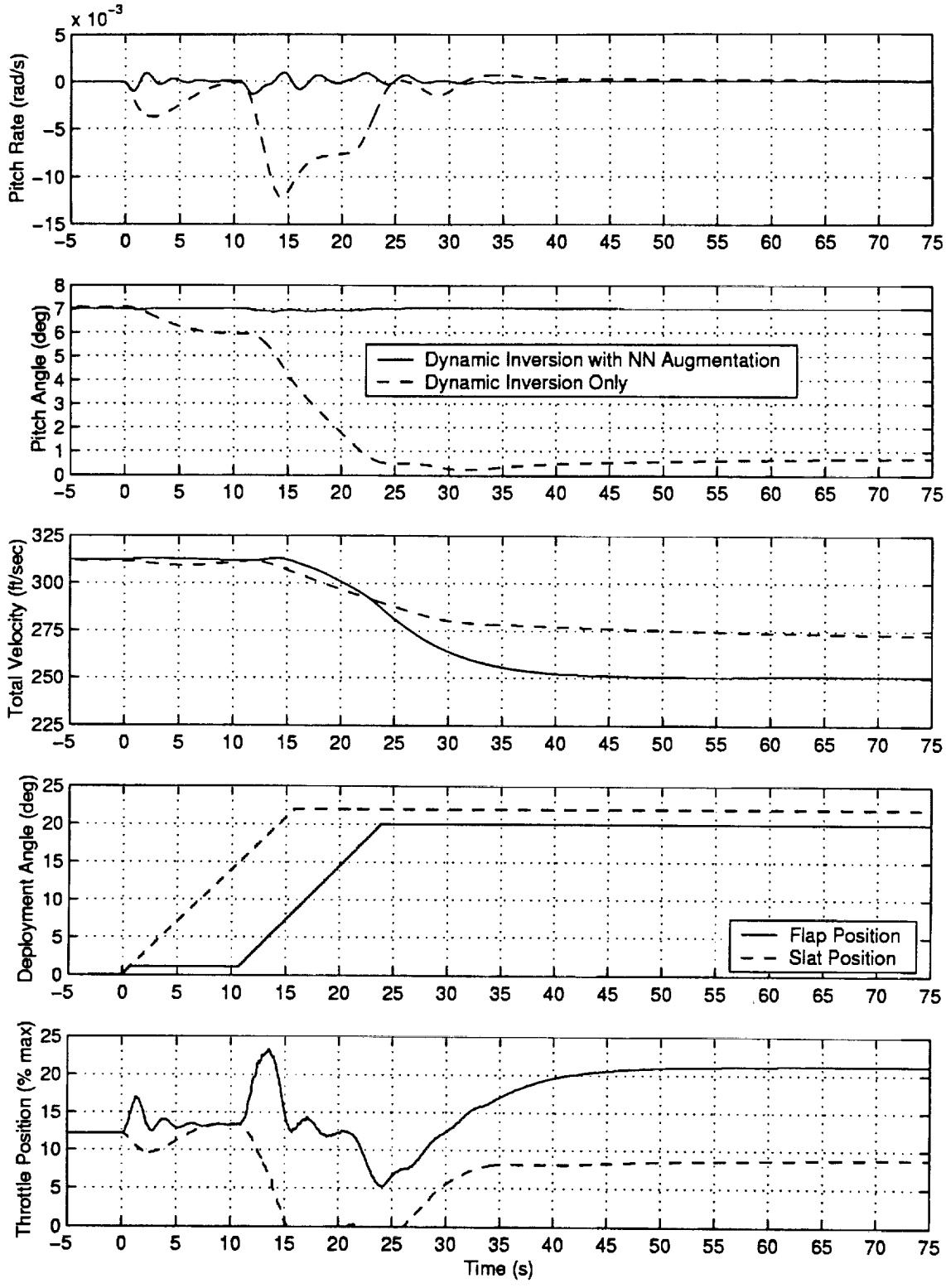

Figure 6.10: Flap Deployment with Pitch Rate Regulation (V=180kts, H=2,000ft, δ_F=0 to 20deg).

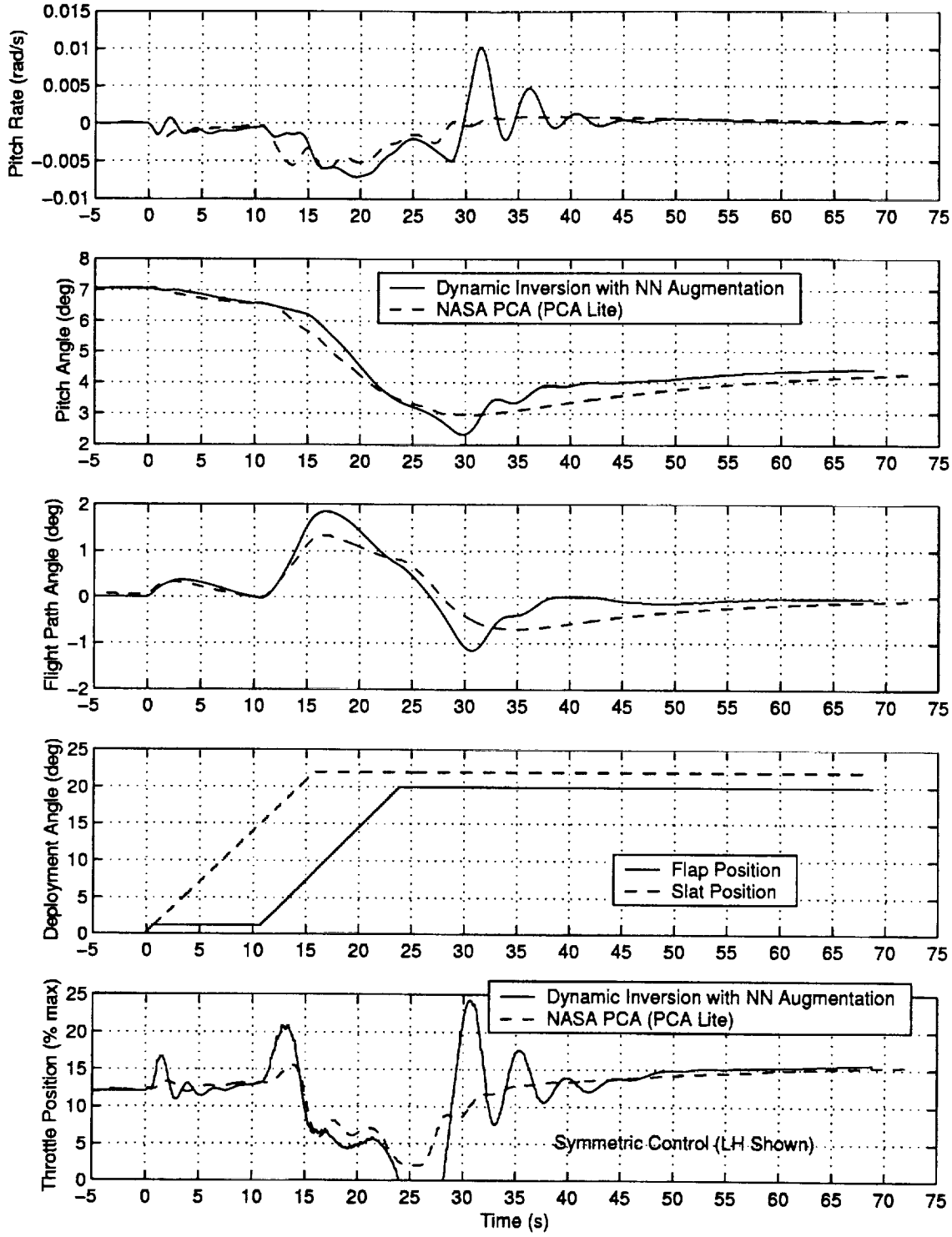

Figure 6.11: Flap Deployment with Flight Path Angle Regulation (V=180kts, H=2,000ft, δ_F=0 to 20deg).

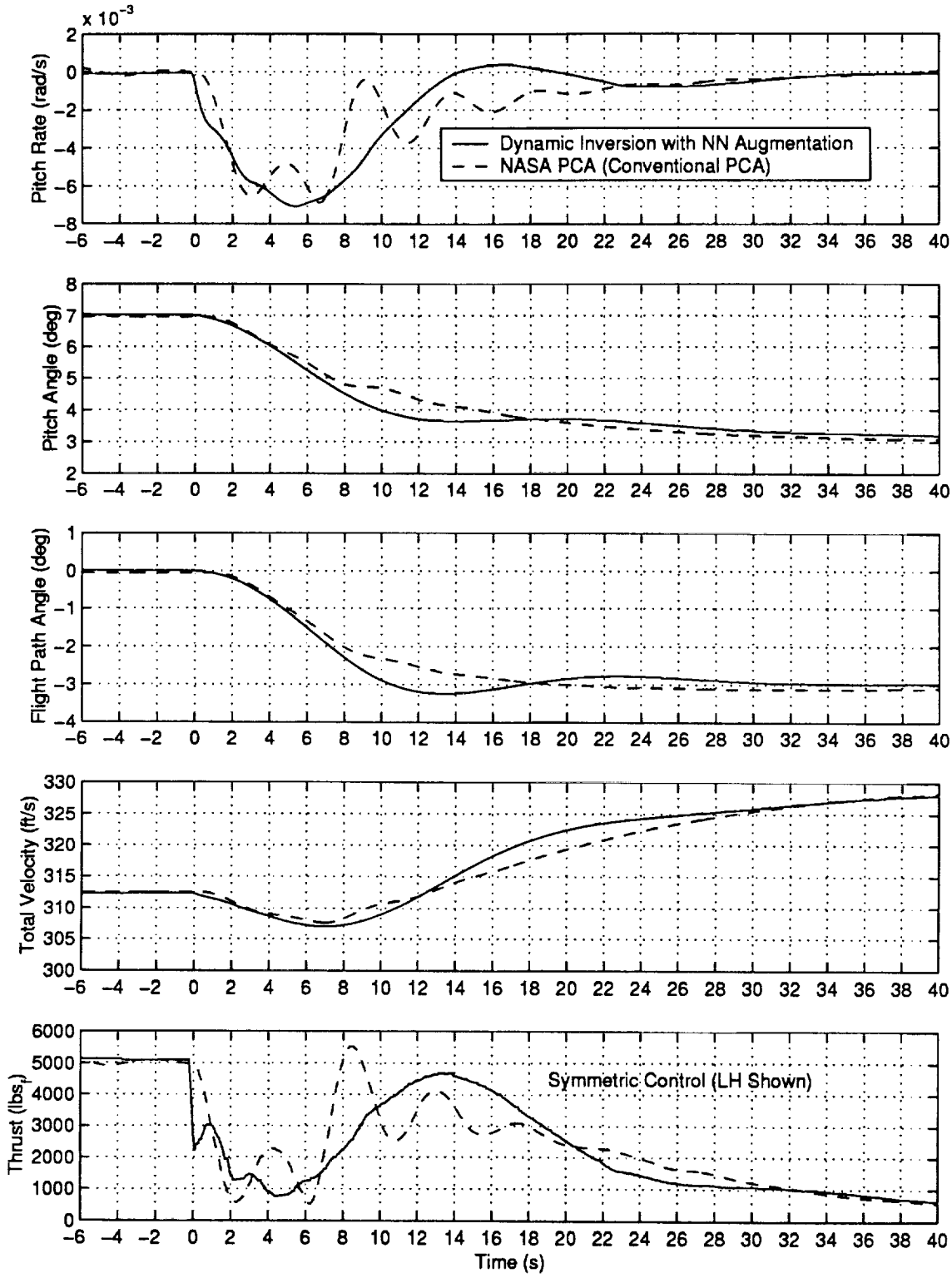

Figure 6.12: Outer-Loop Flight Path Angle Step Response (V=180kts, H=2,000ft, δ_F=0deg).

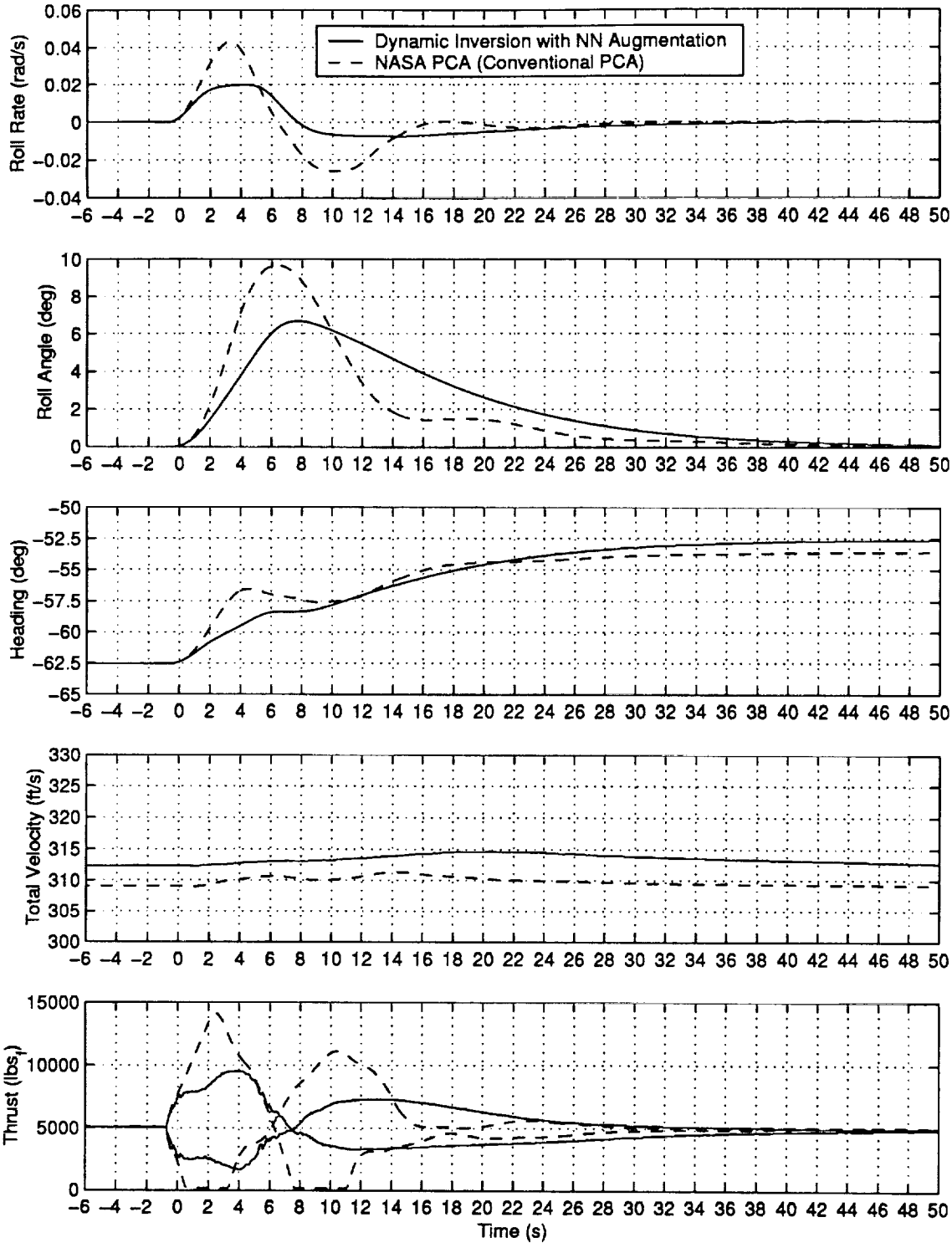

Figure 6.13: Outer-Loop Azimuthal (Heading) Angle Step Response (V=180kts, H=2,000ft, δ_F=0deg).

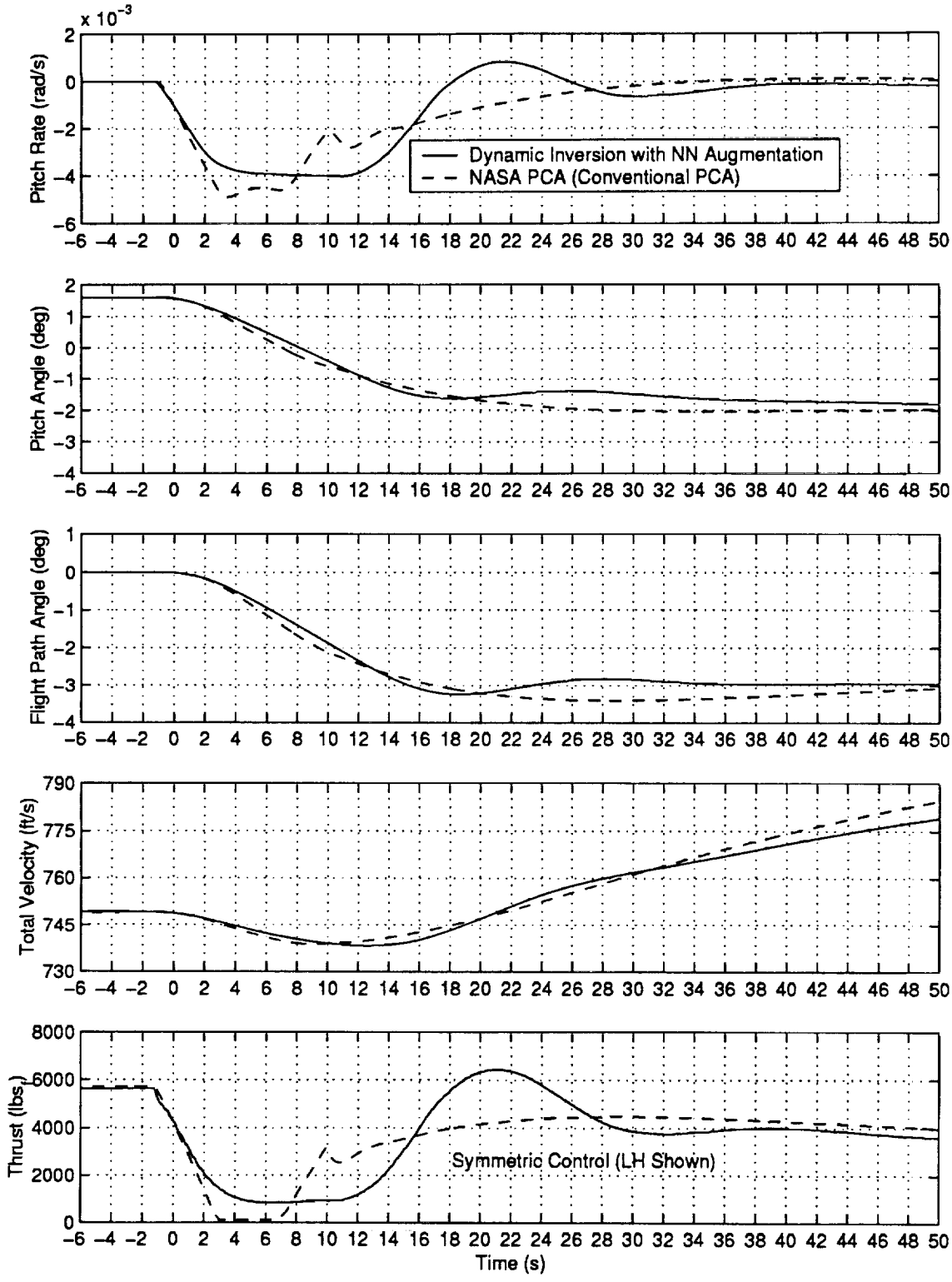

Figure 6.14: Outer-Loop Flight Path Angle Step Response (V=285kts, H=30,000ft, δ_F=0deg).

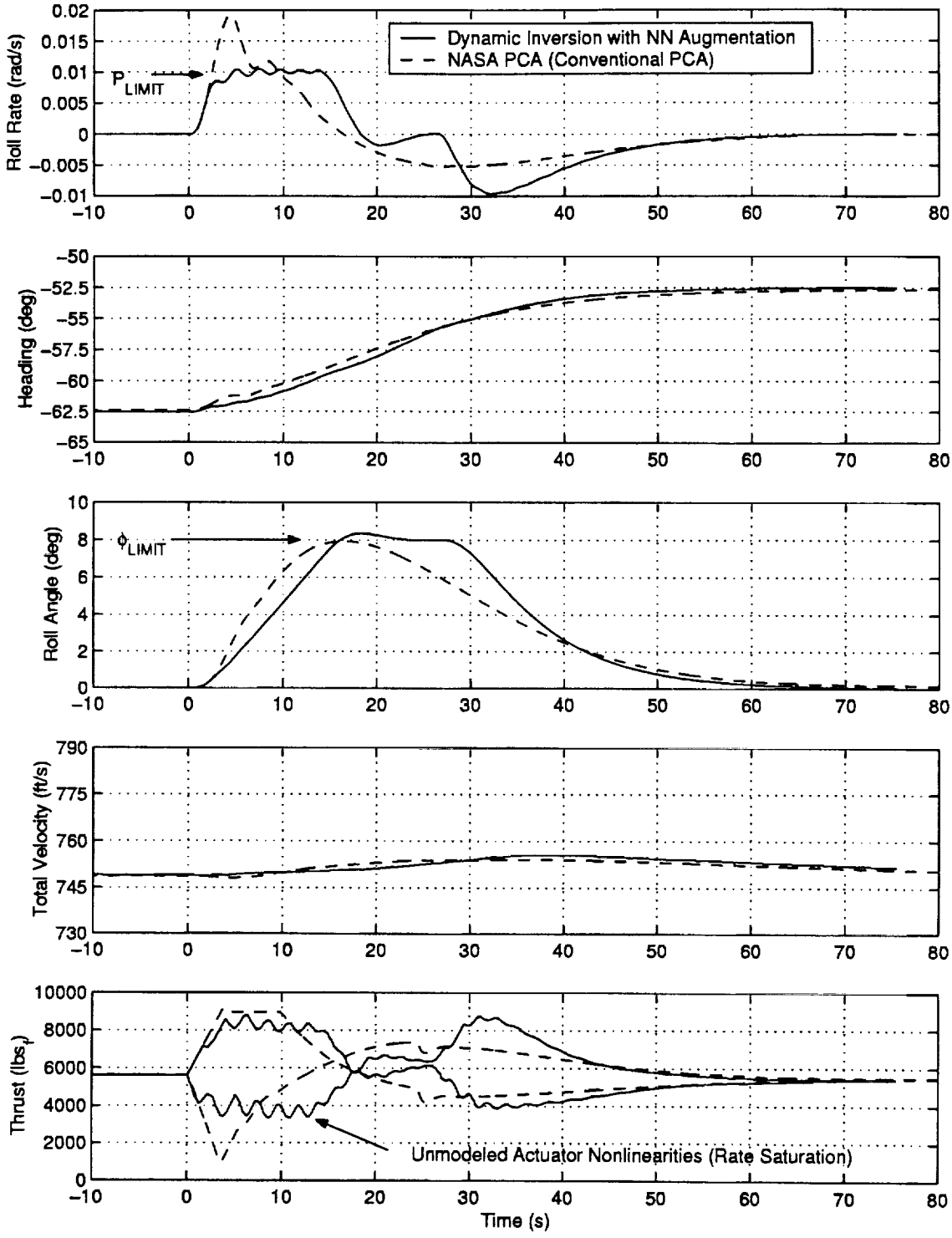

Figure 6.15: Outer-Loop Azimuthal (Heading) Angle Step Response (V=285kts, H=30,000ft, δ_F=0deg).

7 Recommended Further Developments

This section provides a brief overview of what we envision as the major aspects of further investigation.

Model accuracy and data availability

The implementation of the controller architecture was hampered by limited availability of data on the SSS aircraft model, as well as by limited access to the known data. The crude knowledge of the model still allowed for a good demonstration of the flexibility of the controller architecture, demonstrating the fact that it does not rely on detailed aerodynamic and performance data as required in more classic gain scheduled approaches. However, prudence in the design of flight controllers, as well as the desire for clean implementations, suggests that more accurate knowledge of the SSS aircraft model should be used in further development.

Single-Hidden-Layer NN implementation

The current demonstration was based on the controller architecture augmented with a linear in the parameters NN. Linear and non-linear *single-hidden-layer* (SHL) NNs fit within the same controller framework. The SHL-NNs are more powerful in their approximation properties, and are therefore preferred for highly nonlinear mapping tasks. In nominal operation, it is hard to demonstrate their advantage over linear NNs. However, the cases we wish to consider include the possibility of asymmetric failures, and a variety of possible control effectors. Therefore, the nonlinearities may become significant, and the use of SHL-NN is preferable.

Tolerance to unanticipated failures

As more detailed knowledge of the SSS aircraft model becomes available, we wish to evolve the controller into a fully integrated IFPCS. This controller will enhance the flight safety in case of unanticipated critical failures by utilizing a variety of conventional aerodynamic surfaces and configuration controls. For example, hydraulic surfaces, speed brakes, electromechanically driven spoilers, flaps and or slats deployment. The controller architecture as presented, with access to a variety of these effectors, can provide tolerance to unanticipated failures. The implementation of this work will require access to the various dynamic models of the configuration surfaces within the SSS code.

Taking advantage of the maximum available propulsion performance

The results in section 5 include oscillatory behaviour which may be caused by interaction of engine dynamics and the inverting controller. This demonstrates an inherent drawback of feedback linearizing controllers. An illustration of this effect is given in Fig 7. The oscillatory interference would usually be avoided by further bandwidth reduction of the controller. In the

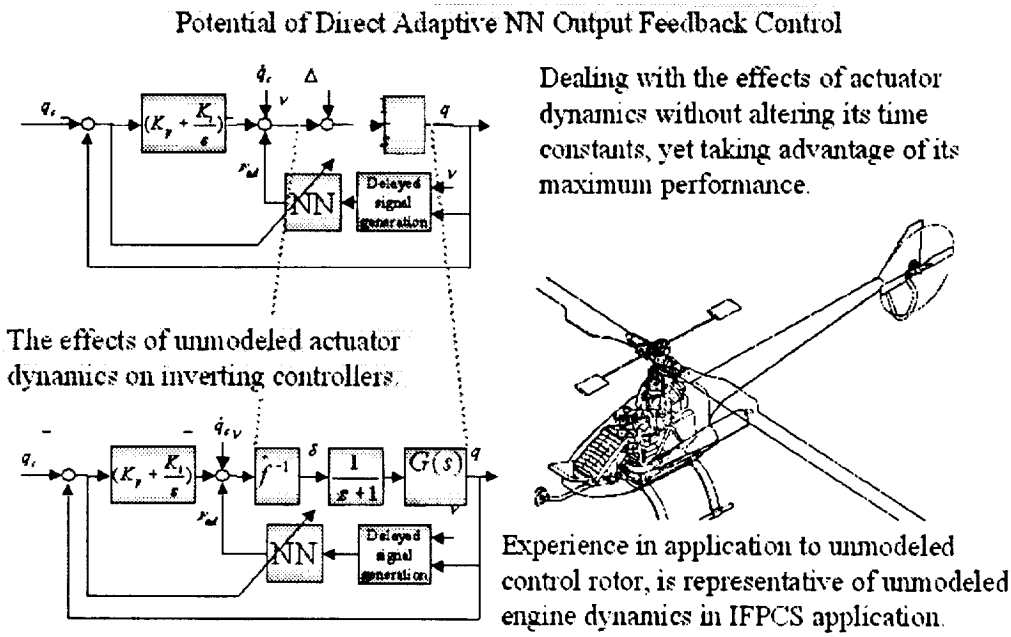

Figure 7.1: Illustration of the effect of unmodeled actuator dynamics. The effect of the unmodelled *control rotor* in the high bandwidth control of the R–50 autonomous helicopter, is similar to the effect of unmodeled engine dynamics for the IFPCS.

IFPCS this is undesirable because of the performance requirements for safety. Furthermore, one would like to take advantage of the maximum available propulsion performance, without attempting to alter its inherent dynamics.

Preliminary results, based on a recent development in *Direct Output Feedback Control* Ref [19, 20, 21], demonstrate the ability to provide robustness with respect to unmodeled actuator dynamics and delays. The IFPCS is an ideal platform for application of this recent development.

Dealing with actuator saturation

Stability and robustness with respect to actuator saturations are subjects of many recent investigations, e.g. Ref [22]. A controller structure similar to that of section 3 is possible, using a form of predictive control Ref [23], and allowing for both rate and position saturations Ref [24]. Rather than a pure inverting controller, this would provide for an optimal controller. The optimal controller performance is guaranteed in the face of actuator saturations. When not saturated, the controller signal is very similar to that of the inverting controller, with the additional effect of control weighting. The optimal controller may be augmented with an adaptive NN in a similar fashion as presented in this report. This would lead to an adaptive IFPCS with some robustness with respect to actuator saturations.

8 Conclusion

This document reports on the first phase of the development of an Intelligent Flight Propulsion Control System (IFPCS). The approximate Feedback Linearizing (FBL) controller of previous work of the authors was integrated with the *approximate Input/Output Linearization* formalism from recent literature. A partial FBL controller, based on a crude approximation of the actual aircraft, is developed. Adaptive Neural Networks (ANNs) are introduced to compensate for the inversion error caused by cascading the linear inversion model with the actual aircraft. The resulting controller design is used to drive the throttle inputs of both wing mounted engines to achieve desired pitch rate and roll rate tracking. The actual aircraft is represented by the NASA *Stone Soup Simulator* representing a generic midsized transport aircraft.

Results indicate that, despite using only crude knowledge of the aircraft, performance similar to extensively gain scheduled design is achievable. The addition of ANNs provides robustness with respect to the uncertainties and unmodeled nonlinearities in the aircraft. Therefore, the nonlinear adaptive design enhances the conventional controller in significant ways. Specifically with respect to,

- consistency of response over a variety of failures,
- applicability to different aircraft without redesign,
- reduction in extensive gain scheduling research & design.

Immediate further work will include a demonstration of propulsion only control in some unanticipated (including asymmetric) failures.

The architecture, as presented in this report, provides a powerful bases for an IFPCS as envisioned by the NTSB & NASA administrators. Specifically;

- the approximate I/O linearizing setup can be extended to output feedback designs,
- the application can be enhanced by utilization of all available primary and secondary control surfaces, with straight forward adjustments for different bandwidth capabilities and control authority limits.

A continuation of the research & development will include:

- Improvement of the inversion model parameters when more data becomes available.
- Implementation and demonstration with the *single-hidden-layer* neural network.
- Development of a fully integrated IFPCS, which will make use of spoilers, flaps and slats, to investigate to what extend handling qualities may be maintained under various unanticipated failure scenarios.
- Use of recent developments in *direct output feedback*, which is not limited by the need for bandwidth separation with engine dynamics, and thus can take maximum advantage of available propulsion.
- The augmentation of the existing controller architecture to allow for actuator and engine rate and position saturations.

A McFarland Inertia Coefficients

$$
\begin{aligned}
\kappa &= I_{xx}I_{zz} - I_{xz}^2 \\
c_1 &= ((I_{yy} - I_{zz})I_{zz} - I_{xz}^2)/\kappa \\
c_2 &= ((I_{xx} - I_{yy} - I_{zz})I_{xz})/\kappa \\
c_3 &= I_{zz}/\kappa \\
c_4 &= I_{xz}/\kappa \\
c_5 &= (I_{zz} - I_{xx})/I_{yy} \\
c_6 &= I_{xz}/I_{yy} \\
c_7 &= 1/I_{yy} \\
c_8 &= I_{xx}(I_{xx} - I_{yy} + I_{xz}^2)/\kappa \\
c_9 &= I_{xx}/\kappa
\end{aligned}
$$

B Data for the Boeing B747-400

	$X(lb)$	$Y(lb)$	$Z(lb)$	$\bar{L}(ft*lb)$	$M(ft*lb)$	$N(ft*lb)$
$u(ft/s)$	-366.1	0	-3538	0	3779	0
$v(ft/s)$	0	-1559	0	-86120	0	39750
$w(ft/s)$	2137	0	-8969	0	-57170	0
$\dot{w}(ft/s^2)$	0	0	585.1	0	-7946	0
$p(rad/s)$	0	0	0	$-1.370*10^7$	0	$-6.688e6$
$q(rad/s)$	0	0	$-1.090e5$	0	$-1.153e7$	0
$r(rad/s)$	0	0	0	4.832e6	0	$-1.014e7$

$$
\begin{aligned}
Weight &= 5.640*10^5 slug*ft^2 \\
I_{xx} &= 1.42*10^7 slug*ft^2 \\
I_{yy} &= 3.23*10^7 slug*ft^2 \\
I_{zz} &= 4.54*10^7 slug*ft^2 \\
I_{xz} &= 8.70*10^5 slug*ft^2
\end{aligned}
$$

Bibliography

[1] National Transportation Safety Board. United airlines flight 232. Aircraft Accident Report PB90-910406, NTSB/AAR-90/06, McDonnell-Douglas DC–10, Sioux Gateway Airport, NTSB, Sioux City, Iowa, July 1989.

[2] J.Bull et al. Piloted simulation tests of propulsion control as backup to loss of primary flight controls for a b747-400 jet transport. Technical memorandum 112191, NASA, April 1997.

[3] R.T.Rysdyk. *Adaptive Nonlinear Flight Control.* PhD thesis, Georgia Institute of Technology, School of Aerospace Engineering, Atlanta, GA, November 1998. PhD-thesis.

[4] J.Burken and etal. Flight test of a propulion based emergency control system on the md–11 airplane with emphasis on the lateral axis. TM 4746, NASA, July 1996.

[5] J.F.Magni, S.Bennani, and J.Terlouw (Eds). *Robust Flight Control*, volume 224 of *Lecture Notes in Control and Information Sciences.* Springer, 1997. GARTEUR.

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

[7] A. Isidori. "Nonlinear Control Systems". Springer Verlag, Berlin, 1989.

[8] J.E.Slotine and W.Li. "Applied Nonlinear Control". Prentice Hall, 1991.

[9] H. Nijmeijer and A.J. van der Schaft. *Nonlinear Dynamical Control Systems.* Springer-Verlag, New York, NY, 1990.

[10] J.Hauser. Nonlinear control via approximate input–output linearization: The ball and beam example. *IEEE, Transactions on Automatic Control*, 37(3), 1992.

[11] S. Bharadwaj, A. Rao, and K. Mease. Entry trajectory tracking law via feedback linearization. *Journal of Guidance,Control and Dynamics*, 21(5):726–732, 1998.

[12] J.Leitner, A.J.Calise, and J.V.R.Prasad. Analysis of adaptive neural networks for helicopter flight controls. In *AIAA Journal of Guidance, Control, and Dynamics*, volume 20-5, pages 972–979, Sept.-Oct. 1997.

[13] R.M.Sanner and J.E.Slotine. Gaussian networks for direct adaptive control. In *IEEE Transactions on Neural Networks*, volume 3-6, pages 837–863, 1992.

[14] B.S.Kim and A.J.Calise. Nonlinear flight control using neural networks. In *AIAA Journal of Guidance, Control, and Dynamics*, volume 20-1, pages 26–33, 1997.

[15] J.Hauser. Nonlinear control via uniform system approximation. *Systems & Control Letters*, 17:145–154, 1991.

[16] B.L. Stevens and F.L. Lewis. *Aircraft Control and Simulation.* John Wiley & Sons, New York, 1992.

[17] J.Kaneshige. Stone soup simulator web site.

[18] B.Etkin and L.D. Reid. *Dynamics of Flight:Stability and Control, 3rd Edition.* John Wiley & Sons, New York, 1996.

[19] N.Hovakimyan, R.T.Rysdyk, and A.J.Calise. Dynamic neural networks for output feedback control. *Submitted to the International Journal on Robust Nonlinear Control*, 1999. Presented at the Conference on Decision and Control, 1999 Arizona, USA.

[20] Naira Hovakimyan, Hungu Lee, and Anthony Calise. On approximate nn realization of an unknown dynamic system from its input-output history. *submitted to The International Journal of Robust and Nonlinear Control*, 1999. to be presented at the 2000 ACC.

[21] Anthony Calise, Naira Hovakimyan, and Hungu Lee. Adaptive output feedback control of nonlinear systems using neural networks. *submitted to The International Journal of Robust and Nonlinear Control*, 1999. to be presented at the 2000 ACC.

[22] special issue on control limits. *International Journal of Control*, July 1999.

[23] S.N.Singh, M.Steinberg, and R.D.DiGirolamo. Nonlinear predicitive control of feedback linearizable systems and flight control system design. *Journal of Guidance, Control, and Dynamics*, 18(5):1023–1028, September-October 1995.

[24] Ping Lu. Tracking control of nonlinear systems with bounded controls and control rates. *Autmatica*, 33(6):199–1202, 1997.