INSTITUTE

FOR

AEROSPACE STUDIES

UNIVERSITY OF TORONTO

A STUDY OF WASHOUT FILTERS FOR A SIMULATOR MOTION BASE

by

Zhi-Qiang Liu

December, 1983

UTIAS Technical Note No. 246
CN ISSN 0082-5263

A STUDY OF WASHOUT FILTERS
FOR A SIMULATOR MOTION BASE

by

Zhi-Qiang Liu

Submitted March, 1983

December, 1983

UTIAS Technical Note No. 246
CN ISSN 0082-5263

ACKNOWLEDGEMENTS

The author would like to express his gratitude to his thesis supervisor, Professor L.D.Reid, for his inspiration and encouragement, for his guidance and willing availability for consulation, and for his constructive criticisms and timely discussions. The assistance he received from the staff and the students at UTIAS is very much appreciated.

Financial support for this work was provided through a China National Graduate Scholarship.

ABSTRACT

The conventional linear washout filter and coordinated adaptive washout filter for a six-degree-of-freedom flight simulator are surveyed. A nonlinear optimal washout filter based on nonlinear regulator and optimal control theories has been synthesized. The proposed nonlinear optimal washout filter is capable of producing the drive signal according to the magnitudes of inputs while it minimizes the given performance criterion. For each channel* four different cases are tested using computer simulation. Comparisons are made with the results obtained from a linear washout filter and an adaptive washout filter. The observation is that the nonlinear optimal and adaptive washout filters are superior to the linear washout filters in some aspects. Recommendations for future work and improvement are also included.

---

*Throughout this study the term 'channel' refers to the longitudinal, or the lateral, or the vertical simulator travel direction in which the control signals are applied.

TABLE OF CONTENTS

Chapter | Page

Chapter | | Page

LIST OF SYMBOLS

a_{cx}, a_{cy}, a_{cz}	Acceleration of the simulated aircraft in body axis,(m/sec)
$\underline{A}$, $\underline{A}_0$, $\underline{A}_1$, $\underline{A}_2$, $\underline{A}_3$	System matrices
$\underline{B}$, $\underline{B}_1$, $\underline{B}_2$, $\underline{B}_3$	Input matrix
$\underline{C}$	Output transfer function
$\underline{F}$, $\underline{\tilde{F}}$	Feedback matrices
$\vec{A}_c$ or $\underline{A}_c$	Translational forces measured at the centroid location
f_{cx}, f_{cy}	Longitudinal and lateral accelerations at the centroid location, in body-axis,(m/sec)
f_{cz}	Vertical acceleration at the centroid location,in body-axis,(m/sec)
f_{ix}, f_{iy}, f_{iz}	Translational acceleration commands prior to translational washout filter
$f(\underline{x}, t)$	A function of $\underline{x}$ and t
$\underline{f}_c$	Specific force vector of the simulated aircraft, in body-fixed frame
$\underline{\varepsilon}_1$	Specific force error vector
$G(s)$	Transfer function: Laplace transformation
J, J_1, J_2, J_3	Cost functions (performance indeces)
$\underline{\nabla J}$	Gradient of the cost function J
$\underline{L}_{c_i}$, $\underline{\hat{L}}_{c_i}$, $\underline{\tilde{L}}_{c_i}$	Rotation matrices
p'_c, q'_c	Angular tilt rates, in body-fixed frame,(rad/sec)
$\underline{Q}$, $\underline{R}$	Weighting matrices
$\underline{R}_T$	Angular rates transformation matrix
R_x, R_y, R_z	The centroid location with respect to the centre of gravity in body-fixed frame
$\vec{r}_{c_i}$ or $\underline{r}_{c_i}$	The centroid location vector relative to the inertial frame
$\underline{\hat{\ddot{r}}}$	Acceleration vector of the simulator in the inertial frame

$\underline{U}$, $\underline{U}_1$, $\underline{U}_2$	Control vectors
U_3	Control input used in the vertical optimal filter
$\underline{U}_L$	Linear control vector
$\underline{U}_{NL}$	Nonlinear control vector
$V(\underline{x})$	Lyapunov function in terms of $\underline{x}$
$\underline{\nabla_x V(\underline{x})}$	Gradient of Lyapunov function with respect to the elements of
$\underline{W}$, $\underline{W}_1$, $\underline{W}_2$, $\underline{W}_3$	Disturbance input vectors
$\hat{x}$, $\hat{y}$, $\hat{z}$	Commanded translational positions after compensation
x_i, y_i, z_i	The inertial frame translational position commands
$\hat{\hat{x}}_c$, $\hat{\hat{y}}_c$, $\hat{\hat{z}}_c$	The elements of $\underline{\hat{\hat{r}}_c}$ (m/sec)
x_p, y_p, z_p	Coordinates of pilot's seat with respect to the centre of gravity in body-fixed frame (m)
x_{pc}, y_{pc}, z_{pc}	Coordinates of the centroid location with respect to pilot's seat, in body-fixed frame
$\underline{y}$	Output vector
$\hat{\phi}$, $\hat{\theta}$, $\hat{\psi}$	Commanded angles after compensation,(rad)
$\hat{\phi}_c$, $\hat{\theta}_c$, $\hat{\psi}_c$	Euler angles of cockpit of simulator,(rad)
ϕ_c, θ_c, ψ_c	Euler angles of simulated aircraft,(rad)
$\underline{\beta}_c$, $\hat{\beta}_c$	The angular vectors, when the angles are very small (<< 1 rad)
$\underline{\dot{\beta}}_c$	Angular rate vector in body-fixed frame of the simulated aircraft
$\underline{\hat{\dot{\beta}}}_c$	Angular rate vector in cockpit-fixed frame of the simulator
$\vec{\omega}_c$ or $\underline{\omega}_c$	Rotation rate vector of body-fixed frame relative to inertial frame
$\underline{\hat{\omega}}$	Rotation rate vector in cockpit-fixed frame of the simulator relative to the

	inertial frame
$\underline{\varepsilon}_2$	Rotation rate error vector
$\underline{\Gamma}$, $\underline{\Gamma}_1$, $\underline{\Gamma}_2$, $\underline{\Gamma}_3$	Disturbance transfer matrices
$<\underline{\cdot}, \underline{\cdot}>$	Denotes inner product of two vectors
$(\underset{\rightarrow}{\ })$	Denotes a vector
$(\underline{\ })^T$	Denotes transpose of a matrix or a column vector
$(\underline{\ })$	Denotes a matrix or a column vector
$(\hat{\ })$	Denotes variable that is in the simulator cockpit
ζ	Damping factor
ω_n	Natural freqency
$\sigma(\underline{\cdot})$	Denotes spectrum or eigenvalues of a square matrix

CHAPTER I

INTRODUCTION

The advent of fixed-base flight simulators has provided both researchers and trainees with low-cost safe devices in which the pilot can visualize the simulated flight manoeuvre by means of on board CRT and other instruments located around the pilot. Lacking in motion, this kind of simulator seldom provides high fidelity (this is not the case for space craft simulators). To meet the need for high quality mimicing of the flight situation, motion-base simulators came into being. With the aid of modern techniques in computer science, control theory, and video graphics, motion base simulators can provide more realism and greater authenticity, at the same time reducing the inconsistency in flight results between simulated flight and real flight. It is desired that the simulator cockpit is commanded to move about in accordance with the states that the real aircraft possesses. Unfortunately, simulation as the definition implies, is not duplication, it can reproduce the real world only approximately. Flight simulation is the art of imitating real flight, it uses the mechanisms of illusion and deception to achieve certain purposes.

Normally, there are two basic constraints in simulation. The first is that man-made models, differential equations representing the dynamics of simulated aircraft, for instance, can only approximate the real situation to a certain degree at the very best. Theoretically the mathematical model for the object studied can be built to be as accurate as possible, but in doing so the object should be well understood, which, for a complicated system, is often not feasible in practice. The second and often the most fatal constraint is the physical limitations in the artificial environment. For instance, the flight simulator which can have six degrees of freedom is mounted in a mechanical structure with limited manoeuvre capability. In each degree of freedom the motion system can not exceed physical limits on position, velocity and acceleration. An example of such limitations is summarized in Table 1.1.

It has been of long standing interest to find a way out of the dilemma in dealing with flight simulation. Researchers have made painstaking efforts to construct motion cue generating circuitry as well as to establish useful theories, such as linear washout filter [1], optimal washout filter [2], quasi-optimal washout filter [3], and adaptive washout filter [4] among which the linear washout filter is classical and fundamental.

As mentioned earlier, the flight simulator itself is nothing more than a device used to provide (or to "deceive") pilot with the "feeling" of real flight. It is the pilot's perception that is of major concern in flight simulation. Therefore in order to prevent the cockpit from hitting the limits of the motion base, a logical way is to modify the commanded variables [1]. Research related to human motion perception organs has been going on for years [5,6,7]. But today many questions still remain unsolved. However, empirical knowledge combined with theoretical and practical considerations lead to the assumption that a pilot can "sense" the same quantities as can be measured by three linear and three rotational accelerometers mounted along three perpendicular axes [1], or simply, that only the accelerations can be "sensed"by the pilot. Consequently, a specific force is defined, and specific force cues are studied in this report.

In this study, the conventional linear washout filter and adaptive washout filter are briefly surveyed, a computer program to simulate these washout filters is developed, and the time responses to different inputs are plotted for later comparison. The major part of this research is to synthezise a nonlinear optimal washout filter based on nonlinear regulator and optimal control theories. The detailed theoretical background and development are given, then the case study and comparisons are carried out. The computer subroutines used for solving the optimal problem are provided in an appendix.

CHAPTER II

CONCEPTUAL ASPECTS OF FLIGHT SIMULATION

Broadly speaking, the term "flight simulator" refers to any device, for example a wind tunnel, that imitates the flight environment. However here it is commonly considered as a class of devices used for both research and training in the investigation of man (pilot) and flight vehicle (cockpit plus motion base). The emphasis may vary from man to machine but always with the integration of both.

Conceptually, a piloted flight simulator consists of, in varying degrees, the following components:

1. A cockpit which can be moved about via commands issued to servo drive systems.
2. Airplane control devices (e.g. stick, rudder pedals etc.) located in the cockpit.
3. A real-time computer (not necessarily on board) which takes input signals from the controls and solves aircraft equations of motion to determine its states (e.g. positions, velocities, attitudes, and angular velocities).
4. Assorted aircraft instruments and all other visual indicators which might be installed on a real aircraft to provide a measure of the aircraft's states (determined by the computer) to the pilot.

The instruments and visual displays can be commanded to act in accordance with the computed aircraft states. Ideally, the cab would also be commanded to move about in accordance with the aircraft states, but it is impossible to do this in practice because of the constraints in the mechanical structure. Usually a motion base can move only a few feet in any direction with limited velocities and accelerations, similar limitations also exist in angular rotations and rotational rates.

Due to the physical limitations of the motion base, some modification of the computed motion commands is necessary before they are used to control the cockpit motion, otherwise, the motion base would be driven into its limits and hence give totally erroneous motion cues to the pilot. A conceptual block diagram of a flight simulator is given in Figure 2.1.

The object of washout filter research is to investigate ways of using computed motion variables to obtain signals representing simulator motions compatible with the limitations of the motion base. In general, the movement of the motion base is inconsistant with the pilot's instruments and other visual displays. However it is observed that human motion sensing system is also limited and selective, that is, specifically he/she may be more sensitive to some motion cues than others. In practice, acceleration or force is considered to have the most pronounced impact on the human perception system. Based on this observation, the signal modification scheme should involve producing an allowable motion which gives the pilot the best motion cues possible.

2.1 Translational Motion Sensing---specific force

As it is observed that human perception organs are biased to force impact, therefore it is useful to define the specific force for the later development of washout filters.

Specific force is defined to be the difference between inertial acceleration and gravitation [8]. Three appropriately mounted linear accelerometers measure the specific force vector (three components).

Since position and constant velocity are not sensed by human perception organs, initial conditions on these quantities may be selected to satisfy simulator constraints. For example, to good approximation, constant velocity motion may be simulated by a cockpit at rest on the ground.

2.2 Rotational Motion Sensing

Although both rotational rate and acceleration are sensed by the pilot we can consider rotational rate as a primary quantity in our mathematical development. That is, if rotational rates are the same in the motion generator as they are in the aircraft then the rotational accelerations will also be the same.

CHAPTER III

REFERENCE FRAMES, ROTATION MATRICES, AND CENTROID TRANSFORMATION

As mentioned earlier, translational accelerations and rotation rates are considered important inputs to human perception organs. Therefore we may use appropriate quatities from the washout circuitry to eventually serve as the input to the motion drive systems. The main interests are summarized here:

1) The three components of specific force acting on the simulated aircraft.
2) The three components of rotational rate acting on the pilot at the cockpit location in the simulated aircraft.

3.1 Reference frames

Since the simulator cockpit is supposed to move like a real aircraft, it is convenient to define a cockpit-fixed reference frame F_c, usually referred to as the body-fixed reference frame in the simulator. Throughout this study, F_c will be a cockpit-fixed reference frame whose origin is at centroid of the motion platform and whose x-axis is parallel to the cockpit reference line. The z-axis is normally downwards in the plane of symmetry and the y-axis orients according to the right hand rule, the detailed convention of cockpit-fixed reference frame is similar to that of body-fixed reference frame [9], (and see Figure 3.1).

Another commonly used reference frame is the inertial reference frame. Throughout this report the inertial reference frame is denoted by F_i. It is assumed that the earth's rotation is negligible, therefore we adopt a local tangent plane as an inertial reference, we also assume that gravitation acts along the direction Z_i of Figuire 3.1 and has a constant magnitude. These assumptions are reasonable for all flight simulators.

For consistency and clarity, throughout this report the following conventions are adopted. The lower-cases c and i when used as subscripts indicate that variables are defined in cockpit and inertial reference frames respectively. To denote variables sensed by a pilot in the cockpit of the simulator, the symbol ^ is used, say, $\hat{f}$ is a variable sensed in the simulator cockpit.

Conventionally, we make use of the notations given in reference [9] to establish the following definitions, geometric relationships, and matrices which will be employed in the development of the equations of washout filters.

3.2 The Rotation Matrix and the Rotation rate Matrix

3.2.1. Rotation Matrices ($\underline{L}_{c_i}$ and $\underline{L}_{i_c}$)

$$\underline{L}_{c_i} = \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \sin\phi\cos\theta \\ \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi & \cos\phi\cos\theta \end{bmatrix} \tag{3.1}$$

where $\underline{L}_{c_i}$ denotes the transformation matrix from $\mathcal{F}_i$ to $\mathcal{F}_c$; ψ, θ, and ϕ are the Euler angles defined in reference [9].

It is known that $\underline{L}_{c_i}$ is an orthogonal matrix and the following relation exists between $\underline{L}_{c_i}$ and $\underline{L}_{i_c}$:

$$\underline{L}_{c_i}^{-1} = \underline{L}_{c_i}^{T} = \underline{L}_{i_c} \tag{3.2}$$

The detailed description of equations (3.1) and (3.2) is available in reference [9].

3.2.2. The Rotation Rate Transformation Matrix $\underline{R}_T$ Relating to $\mathcal{F}_i$ and $\mathcal{F}_c$

$$\underline{R}_T = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \sin\phi\cos\theta \\ 0 & -\sin\phi & \cos\psi\cos\theta \end{bmatrix} \tag{3.3}$$

The inverse of $\underline{R}_T$ is

$$\underline{R}_T^{-1} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta \end{bmatrix} \tag{3.4}$$

3.2.3. The Centroid Transformation

A useful reference point in the simulator is the centroid of the upper frame of the motion base. The location of the centroid with respect to the centre of gravity is defined as (see Figure 3.2)

$$R_x = x_p + x_{pc}, \quad R_y = y_p + y_{pc}, \quad R_z = z_p + z_{pc} \qquad (3.5)$$

which are in the body-fixed frame, where x_p, y_p and z_p locate the pilot's seat with respect to the centre of gravity of the simulated aircraft. x_{pc}, y_{pc} and z_{pc} locate the centroid with respect to pilot's seat. A reference to Figure 3.2 may be helpful in understanding these variables. According to [9], once the centriod location is determined, the translational acceleration of the centroid is given by the following equation

$$\underset{\to}{A}_c \triangleq \underline{L}_{c_i}\, \ddot{\underset{\to}{r}}_{c_i} = \ddot{\underset{\to}{r}}_{c_i} + \ddot{\underset{\to}{R}}_c + \underset{\to}{\omega}_c \times \dot{\underset{\to}{r}}_{cc} + \underset{\to}{\omega}_c \times \dot{\underset{\to}{r}}_{cc} + \dot{\underset{\to}{\omega}}_c \times \underset{\to}{R}_c$$
$$+ 2\underset{\to}{\omega}_c \times \dot{\underset{\to}{R}}_c + \underset{\to}{\omega}_c \times \underset{\to}{\omega}_c \times \underset{\to}{R}_c \qquad (3.6)$$

where $\underset{\to}{r}_{c_i}$, $\underset{\to}{r}_{cc}$, $\underset{\to}{\omega}_c$, and $\underset{\to}{R}_c$ are the vectors shown in Figure 3.2.

Usually, once the configuration of the motion base is made, $\underset{\to}{R}_c$ is a constant vector. Therefore

$$\dot{\underset{\to}{R}}_c = \ddot{\underset{\to}{R}}_c = \underset{\to}{0}$$

equation (3.6) becomes

$$\underset{\to}{A}_c = \ddot{\underset{\to}{r}}_{cc} + \underset{\to}{\omega}_c \times \ddot{\underset{\to}{r}}_{cc} + \dot{\underset{\to}{\omega}}_c \times \underset{\to}{R}_c + \underset{\to}{\omega}_c \times \underset{\to}{\omega}_c \times \underset{\to}{R}_c \qquad (3.7)$$

CHAPTER IV

LINEAR WASHOUT FILTER

Traditionally, washout filters were derived empirically. Among many methods, the most fundamental ones are

a) Scaling;
b) residual tilting (coordinating);
c) linear filtering.

To meet the performance requirements, combinations of these techniques are often necessary. The essential part of washout circuitry is a high-pass filter used to exclude undesired low frequency signals from the motion base input. A high-pass filter is always used in linear washout circuitry, because low frequencies or constant inputs would require a large motion base excursion [1] which might lead the motion base to hit the simulator's travel limits.

In 1970, Conard and Schmidt proposed a coordinated linear washout filter [1]. As the name implies, in this method they coordinate the translational channels and rotational channels to simulate partially steady state specific forces (see Figure 4.1 for the function block diagram). Hence a better representation of the specific forces may be produced in principle.

The detailed derivation of a linear washout filter is given in reference [1].

It is observed that an effective washout filter for the acceleration input should have at least a transfer function of third order. For illustrative purposes, a typical second order high pass is given as follows [1]

$$G(s) = \frac{ks^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{4.1}$$

suppose $\hat{\ddot{r}}_c(s)$ [†], $\ddot{r}_c(s)$ are the simulator cockpit and the simulated aircraft accelerations respectively, then

$$\hat{\ddot{r}}_c(s) = \frac{ks^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}\,\ddot{r}_c(s) \tag{4.2}$$

[†] In general, $\hat{\ddot{r}}_c$ is only the high frequency part of the total simulator motion base acceleration.

Let the initial conditions be $\dot{r}_c(o) = 0$, $r_c(o) = 0$, and suppose $\ddot{r}_c$ is a step input, then time responses are given in Figures 6.4∿6.7, the values used for ζ and ω_n are given in Table 1 of Appendix C. Investigation of these figures shows that this kind of filter is capable of "washing out" the specific force inputs. However, because of the linearity of the washout filter, all motion cues are washed out at the same time regardless of the difference in magnitudes of inputs. Therefore the linear washout filter often unnecessarily reduces the capability of the motion base, which in turn reduces the fidelity of the simulator.

CHAPTER V

COORDINATED ADAPTIVE WASHOUT FILTER

Following the same idea of coordination of translation and rotation to generate more accurate longitudinal and lateral force cues, R.V.Parrish et al conceived the coordinated adaptive washout filter in 1974 [4].

The design philosophy for these filters is to present as much of the force cues as possible within the constraints of the motion base. Theoretically, the coordinated adaptive washout filter is based on the theory of parameter optimization.

The detailed development of the adaptive washout filter was carried out by R.V.Parrsh et al [4]. For completeness, some major aspects of this development and theoretical background will be introduced briefly in the following sections. The results of a computer simulation will also be given later.

5.1 Parameter Optimization

Optimization is one of the most important problems in control system engineering. One aspect of optimization is the selection of system parameters in such a manner that the performance of the system is as close to optimum as possible, based on a given criterion for optimality. For example it may be desired to minimize cost or energy consumption or to maximize profit, productivity, or distance of travel etc..

In the following sections, the mathematical development of the parameter optimization using continuous steepest descent is presented.

5.1.1 Mathematical Description of Dynamic Systems

Dynamic systems are described by means of differential equations. Any system of order n can be represented by n first-order equations. Without loss of generality the dynamic system can be expressed by the state equation:

$$\dot{\underline{x}} = \underline{f}(\underline{x}, t, \underline{\alpha}, \underline{u}) \tag{5.1}$$

where $\dot{\underline{x}} \triangleq [\dot{x}_1, \dot{x}_2, \ldots, \dot{x}_n]^T$, $\underline{x} \triangleq [x_1, x_2, \ldots, x_n]^T$, $\underline{\alpha} \triangleq [\alpha_1, \alpha_2, \ldots, \alpha_n]^T$ $\underline{u} \triangleq [u_1, u_2, \ldots, u_m]^T$ and α_i represent the adjustable parameters, and $\underline{u}$ the input.

The initial condition is given by

$$\underline{x}(o) = \underline{x}_o$$

Description of dynamic systems in terms of their states is consistent with modern control system theory and provides for a compact interpretation of the behavior of multiparameter systems.

For each set of parameter values, say, $\underline{\alpha}^{(1)}$, or $\underline{\alpha}^{(2)}$ the system behavior will be described by means of a solution given by $\underline{x}(\underline{\alpha}^{(1)}, t)$ or $\underline{x}(\underline{\alpha}^{(2)}, t)$.

In solving optimization problems, a performance criterion function relating to the parameters, the input, and the states of the system is always needed. For simplicity we denote the criterion function as

$$J = J(\underline{x}, \underline{\alpha}, \underline{u})$$

Usually, for an optimization problem, it is desired that by selecting $\underline{\alpha}$ or $\underline{u}$ or both, that

$$J \rightarrow \min J(\underline{x}, \underline{\alpha}, \underline{u}) \quad \text{or} \quad \max J(\underline{x}, \underline{\alpha}, \underline{u})$$

The configuration of criterion function varies in different problems.

5.1.2 Optimization by Continuous Steepest Descent [10]

In system engineering, optimization is categorized into two main problems; static optimization which ignores the dynamic characteristics of the system and dynamic optimization.

First we consider the problem of static optimization. A typical static system is a set of algebraic equations with a number of adjustable parameters. It may be stated in the matrix form

$$\underline{A}\,\underline{x} = \underline{b}$$

where $\underline{x}$ and $\underline{b}$ are n-dimentional vectors, $\underline{A}$ is an nxn matrix. A criterion function depending on the particular values of the parameters is denoted as follows

$$J = J(\underline{\alpha}) \tag{5.2}$$

where $\underline{A}$ is $\underline{A}(\underline{\alpha})$ and $J = J[\underline{x}(\underline{\alpha}), \underline{\alpha}] = J(\underline{\alpha})$.

It is desired to derive a method of adjusting the parameters such that starting from an arbitrary initial point $\underline{\alpha}_o$, the parameters will move toward the values which minimize J. It is known that the path of the steepest descent is the path which is normal to the contour lines in the parameter space which represent constant values of the criterion function. Consequently, it can be seen intuitively that the parameters should be adjusted such that their rate of change with respect to time will be tangental to the gradient vector in this same space. If each component of the changing parameter vector, i.e. each component of $\dot{\underline{\alpha}}$ is colinear with the corresponding component of the gradient vector, then the adjustment will in fact be along the path of steepest descent (see Figure 5.1).

To mathematically verify the above statement, we formulate the rate of change of a criterion function with respect to time as follows

$$\frac{dJ}{dt} = \frac{\partial J}{\partial \alpha_1}\frac{d\alpha_1}{dt} + \frac{\partial J}{\partial \alpha_2}\frac{d\alpha_2}{dt} + \ldots + \frac{\partial J}{\partial \alpha_n}\frac{d\alpha_n}{dt} \qquad (5.3)$$

or in vector form, equation (5.3) can be written as follows

$$\frac{dJ}{dt} = \langle \underline{\nabla J}, \dot{\underline{\alpha}} \rangle \qquad (5.4)$$

where

$$\underline{\nabla J} = \begin{bmatrix} \frac{\partial J}{\partial \alpha_1} \\ \frac{\partial J}{\partial \alpha_2} \\ \vdots \\ \frac{\partial J}{\partial \alpha_n} \end{bmatrix}, \qquad \dot{\underline{\alpha}} = \begin{bmatrix} \dot{\alpha}_1 \\ \dot{\alpha}_2 \\ \vdots \\ \dot{\alpha}_n \end{bmatrix}$$

The rate of change of J with respect to time is the inner product of the two vectors $\underline{\nabla J}$ and $\dot{\underline{\alpha}}$.

Now we wish to maximize dJ/dt. Clearly from equation (5.4), we know that maximizing dJ/dt is equivalant to maximizing the inner product. This occurs when the two vectors $\underline{\nabla J}$ and $\dot{\underline{\alpha}}$ are parallel or, in other words, when corresponding components of the two vectors are proportional to one another. That is

$$\frac{\dot{\alpha}_i}{\dot{\alpha}_j} = \frac{\partial J/\partial \alpha_i}{\partial J/\partial \alpha_j} \qquad (5.5)$$

or in vector form

$$\underline{\dot{\alpha}} = K\underline{\nabla J} \qquad (5.6)$$

where K is a coefficient. If $K > 0$, then equation (5.6) represents an ascent path; if $K < 0$ it represents a descent path, and is referred to as continuous steepest descent(CSD).

5.1.3 Extension of CSD to Dynamic Systems

The previous section has been concerned with the problem of static parameter optimization. Most engineering problems, however, are concerned with the optimization of dynamic systems described by differential equations, consequently the method outlined in the preceding section cannot be applied directly to dynamic systems.

For the sake of simplicity, we consider a model reference adaptive (MRA) control system depicted in Figure 5.2, where $y_p(\underline{\alpha},t)$ is the output of the dynamic system, $\underline{\alpha}$ is the adjustable parameter vector which can be adjusted continuously to make $y_p(\underline{\alpha}, t)$ as close to the output of the reference model as possible. However, $y_p(\underline{\alpha}, t)$ is not an instantaneous function of due to the characteristics of the dynamic system, rather, it depends on the present state and history of both the system and the parameters. Consequently, the fundamental assumption made in Section 5.1.2, namely, that J is an algebraic function of the parameters is now violated. In dynamic systems J depends on the entire time history of the parameters.

In order to make use of the steepest descent method, we have to make some modifications in the development. One way to circumvent the problem is to fix $\underline{\alpha}$ with respect to time during computation of the gradient. Another way of modification which is more extensively used in adaptive control problems to attain parameter optimization is to assume that the rate of adjustment of parameters is slow compared to the basic time constraints of the system itself. This is the so-called "approximate gradient method".

5.2 Model Reference Adaptive Control (MRAC)---a review

In contrast to conventional control theroy, adaptive control refers to the control of partially known systems [11]. For many years there has been an increasing interest in adaptive control which can be attributed to the fact that there is invariably some uncertainty in the dynamic characteristics of most practical systems.

For this class of system, the tools of conventional control theory, even when used efficiently in the design of controllers, are inadequate in achieving satisfactory performance in the entire range over which the characteristics of the system may vary. Hence some type of monitoring of the system's behavior followed by the adjustment of the control input, i.e. feedback, is needed and is referred to as adaptive control. It is possible to monitor different system characteristics and take different control actions, and hence there is a large class of nonlinear feedback systems which can be referred to as adaptive control systems.

Since adaptive control systems are nonlinear feedback systems, there is the distinct possibility that such systems can become unstable. Even though there has been interest in this area for over twenty years, due to the lack of a well developed stability theory for such systems, the application of adaptive control to practical systems has not been attempted on a large scale, until recently. Most applications and research have been made in control of aircraft and spacecraft which indicates that adaptive control theory may be especially suitable for flight vehicle control system design.

Among many theories proposed, the model reference adaptive control has been widely applied. In this investigation, we will use it to solve the motion base control problem.

5.2.1 The General Statement of the Problem

The input and output of a linear time-invariant plant with unknown parameters are $\underline{\alpha}(\cdot)$ and $y_p(\cdot)$ respectively (see Figure 5.2). A linear time-invariant reference model and a reference input $r(\cdot)$ are specified which result in a model output $y_m(\cdot)$. From all available on-line data it is desired to determine the control input such that the error $(y_p - y_m)$ tends to zero.

Our interest now is to determine the information needed to solve the problem and generate a model for realizing the controller. The parameterization of the control object, the structure of the controller and the manner in which the controller parameters have to be adjusted to achieve stable control are all found to be important.

5.2.2 The Structure of the Controller (direct control)

A controlled plant p is completely represented by the input-output pair $\{u(\cdot), \ y_p(\cdot)\}$ and can be modelled by a transfer function

$$G_p(s) = \frac{K_p W_p(s)}{R_p(s)} \tag{5.7}$$

where $W_p(s)$ and $R_p(s)$ are polynomials of degrees $m(\leq n-1)$ and respectively. A stable reference model is represented by the input-output pair$\{r(\cdot), \ y_m(\cdot)\}$ and has a transfer function

$$G_m(s) = \frac{K_m W_m(s)}{R_m(s)} \tag{5.8}$$

The error between plant and model outputs is defined as

$$e(t) \triangleq y_p(t) - y_m(t) \tag{5.9}$$

The problem is to determine the control input u(t), so that

$$\lim_{t\to\infty} e(t) = 0 \tag{5.10}$$

Assume that the transfer function $G_p(s)$ for the plant has n poles, then as Narendra asserted in [12], $G_p(s)$ has a maximum of 2n unknown parameters which are coefficients of $K_pW_p(s)$ and $R_p(s)$, therefore the controller structure must have adequate freedom so that by adjusting the control parameters the transfer function of the plant together with the controller can match that of any specified model.

For direct control, the configuration shown in Figure 5.3 has evolved as the basic one for the controller. The input u(t) and the output $y_p(t)$ of the plant are correspondingly fed into the two filters of identical form, whose state vector $\underline{V}_1(t)$ and $\underline{V}_2(t)$ are of dimension (n-1). Together with r(t) and the output $y_p(t)$ they constitute the 2n signals whose linear combination yields the desired input u(t). If $\underline{\gamma}(t)$ is a control parameter vector with 2n elements, then

$$u(t) = \underline{\gamma}^T(t)\underline{W}(t)$$

where

$$\underline{\gamma}^T(t) = [\gamma_1(t), \gamma_0(t), \ldots, \gamma_{2n}(t)]$$

$$\underline{W}^T(t) = [r(t), \underline{V}_1^T(t), y_p(t), \underline{V}_2^T(t)]$$

It is shown in reference [12] that there exists a constant vector $\underline{\gamma}^*$ of dimension 2n such that when $\underline{\gamma}(t) \equiv \underline{\gamma}^*$, the transfer function of the plant will match that of the model. Hence, it only remains to show how $\underline{\gamma}(t)$ is to be adjusted so that

$$\lim_{t\to\infty} \underline{\gamma}(t) = \underline{\gamma}^* \tag{5.11}$$

5.2.3 Modification of the Control Structure

The adaptive control structure in the previous section is based on the idea that by adjusting parameters, the system output error will eventually vanish. In the control of a flight simulator, the model output and the controlled system (i.e. the simulator) output can never be matched because of the special characteristics of the system. Therefore some modifications should be made.

A.P.Sage has suggested a configuration [13]. Instead of directly using the system error as a criterion, he defined a cost function $J(\underline{e})$ related to the system errors. Then he minimizes the cost function by forming the gradient vector for $J(\underline{e})$, and adjusts system parameters, possiblly by a linear programming procedure (approximate steepest descent, for instance), until the gradient becomes zero. Before the model reference adaptive system is in full adaptation to the model, the gradient will not be zero and is defined as the error quantity

$$EQ = \frac{\partial J}{\partial \underline{p}} \qquad (5.12)$$

where $\underline{p}^T = [p_1, p_2, \ldots, p_m]$ is a parameter vector.

The steepest descent procedure is implemented as introduced in Section 5.1.

5.3 The Adaptive Washout Filter [4]

Based on the theory and the discussion in Sections 5.1 and 5.2, the proposed adaptive washout filter is illustrated in Figure 5.4. It is clear that this adaptive filter is a model reference adaptive control system which has a structure similar to the one shown in Figure 5.2, and uses the input generated from the dynamic equations of the simulated aircraft as a reference. The output of the controlled system is compared to the aircraft equations of motion. After the comparison, the adaptive parameters are adjusted according to the motion base environment, at the same time minimizing the cost function by using an approximate steepest descent method.

In this proposed adaptive washout filter, the cost function J is defined for each channel in the form of

$$J = \frac{1}{2}(f_m - f_s)^2 + \frac{W}{2}(\dot{\alpha}_m - \dot{\alpha}_s)^2 + \frac{b}{2}x_s^2 + \frac{c}{2}\dot{x}_s^2 \qquad (5.13)$$

where
- f_m ----the acceleration of the reference model;
- $\dot{\alpha}_m$ ----the angular velocity of the reference model;
- f_s ----the acceleration of the simulator;
- $\dot{\alpha}_s$ ----the angular velocity of the simulator;
- x_s ----the position away from the neutral point;
- $\dot{x}_s$ ----the translational velocity of the simulator,

which are all in inertial frame.

At present, we assume that the hydraulic system of the simulator has only proportional action.

The control law is defined as follows

$$\ddot{x}_s = p_{s,1}\, f_m - d\dot{x}_s - e x_s \tag{5.14a}$$

$$\dot{\alpha}_s = p_{s,2}\, f_m + p_{s,3}\, \dot{\alpha}_m \tag{5.14b}$$

where $p_{s,j}$ $(j = 1,2,3)$ are adjustable parameters, f_m, $\dot{\alpha}_m$ are reference model inputs, $\ddot{x}_s$, $\dot{x}_s$, x_s, α_s are states of the simulator, and d and e are pre-determined constant coefficients.

Applying steepest descent procedure yields

$$\dot{p}_{s,j} = -K \frac{\partial J}{\partial p_{s,j}} \qquad j = 1,\ 2,\ 3 \tag{5.15}$$

from equation (5.13) we have (where the present case is such that $f_s = \ddot{x}_s$)

$$\frac{\partial J}{\partial p_{s,j}} = (f_m - \ddot{x}_s)\left(\frac{\partial f_m}{\partial p_{s,j}} - \frac{\partial \dddot{x}_s}{\partial p_{s,j}}\right) + W(\dot{\alpha}_m - \dot{\alpha}_s)\left(\frac{\partial \dot{\alpha}_m}{\partial p_{s,j}} - \frac{\partial \dot{\alpha}_s}{\partial p_{s,j}}\right)$$
$$+ b x_s \frac{\partial x_s}{\partial p_{s,j}} + c\dot{x}_s \frac{\partial \dot{x}_s}{\partial p_{s,j}} \tag{5.16}$$

Substituting equation (5.16) in equation (5.15) we get

$$\dot{p}_{s,j} = -K\left\{(f_m - \ddot{x}_s)\left(\frac{\partial f_m}{\partial p_{s,j}} - \frac{\partial \dddot{x}_s}{\partial p_{s,j}}\right) + W(\dot{\alpha}_m - \dot{\alpha}_s)\left(\frac{\partial \dot{\alpha}_m}{\partial p_{s,j}} - \frac{\partial \dot{\alpha}_s}{\partial p_{s,j}}\right)\right.$$
$$\left. + b x_s \frac{\partial x_s}{\partial p_{s,j}} + c\dot{x}_s \frac{\partial \dot{x}_s}{\partial p_{s,j}}\right\} \tag{5.17}$$

The state sensitivity equations are obtained by assuming that the parameters $p_{s,j}$ are independent, and that derivatives are continuous in the adjustable parameters and time. For example, if $\underline{x}=\underline{x}(\underline{p},t)$, where $\underline{p}$ is a parameter vector, t time; $\underline{p}$, $i\varepsilon\bar{n}$, are $\bar{i}nd\bar{e}p\bar{e}ndent$, and $\underline{x}$ has continuous derivatives with $\bar{r}esp\bar{e}ct$ to $\underline{p}$ and t, therefore $\bar{w}e$ have[14]

$$\frac{\partial}{\partial p_i}\left(\frac{\partial^2 x}{\partial t^2}\right) = \frac{d^2}{dt^2}\left(\frac{\partial x}{\partial p_i}\right) = \frac{d}{dt}\left(\frac{\partial \dot{x}}{\partial p_i}\right)$$

From equation (5.14), we get

$$\frac{d}{dt}\left(\frac{\partial x_s}{\partial p_{s,j}}\right) = \frac{\partial p_{s,1}}{\partial p_{s,j}} f_m + p_{s,1}\frac{\partial f_m}{\partial p_{s,j}} - d\frac{\partial \dot{x}_s}{\partial p_{s,j}} - e\frac{\partial x_s}{\partial p_{s,j}} \tag{5.18}$$

$$\frac{d}{dt}\left(\frac{\partial \alpha_s}{\partial p_{s,j}}\right) = \frac{\partial p_{s,2}}{\partial p_{s,j}} f_m + p_{s,2}\frac{\partial f_m}{\partial p_{s,j}} + \frac{\partial p_{s,3}}{\partial p_{s,j}}\dot{\alpha}_m + p_{s,3}\frac{\partial \dot{\alpha}_m}{\partial p_{s,j}} \tag{5.19}$$

Note that the assumption that the $p_{s,j}$ are independent was used here. Therefore

$$\frac{\partial p_{s,i}}{\partial p_{s,j}} = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases}$$

Thus from equation (5.18) and equation (5.19), we have

$$\frac{d}{dt}\left(\frac{\partial \dot{x}_s}{\partial p_{s,1}}\right) = f_m - d\frac{\partial \dot{x}_s}{\partial p_{s,1}} - e\frac{\partial x_s}{\partial p_{s,1}} \tag{5.20}$$

$$\frac{d}{dt}\left(\frac{\partial x_s}{\partial p_{s,2}}\right) = p_{s,1}\frac{\partial f_m}{\partial p_{s,2}} - d\frac{\partial \dot{x}_s}{\partial p_{s,2}} - e\frac{\partial x_s}{\partial p_{s,2}} \tag{5.21}$$

$$\frac{d}{dt}\left(\frac{\partial x_s}{\partial p_{s,3}}\right) = p_{s,1}\frac{\partial f_m}{\partial p_{s,3}} - d\frac{\partial \dot{x}_s}{\partial p_{s,3}} - e\frac{\partial x_s}{\partial p_{s,3}} \tag{5.22}$$

$$\frac{d}{dt}\left(\frac{\partial \dot{\alpha}_s}{\partial p_{s,2}}\right) = f_m + p_{s,2}\frac{\partial f_m}{\partial p_{s,2}} \tag{5.23}$$

$$\frac{d}{dt}\left(\frac{\partial \dot{\alpha}_s}{\partial p_{s,3}}\right) = p_{s,2}\frac{\partial f_m}{\partial p_{s,3}} + \dot{\alpha}_m \tag{5.24}$$

From simultaneous integration of the equations (5.17)~(5.23) and the corresponding equations of Appendix A in real time, we get the adaptive parameters $p_{s,j}$ (j=1,2,3) used in the control law. Selection of the values of the constants [W, b, c, d, e, k,{$p_{s,j}$(0),j = 1, 2, 3}] must be based on the constraints of the motion base and the flight environment, as well as the desired emphasis of washout (i.e. to represent specific force, rotational rate, or some combination of both).

The detailed equations for all three channels are documented in Appendix A.

A computer program was made to implement this control system. The time responses to different inputs are given in

Figures 6.8∿6.19, the parameters used in the computation are given in Table 2 of Appendix C. The discussion of the results is deferred to Chapter 6 to allow a comparison with the results of the nonlinear optimal washout filter.

CHAPTER VI

NONLINEAR OPTIMAL WASHOUT FILTER

In developing washout filters for a flight simulator motion base, one of the key objectives is to allow as large force cues as possible and at the same time keep the motion base within its limits. This, hopefully, will provide the pilot with good fidelity. Trying to obtain better motion and force cues is a perplexing problem which has been with the engineers involved in this field for many years. Efforts have been resulted in little improvement so far.

To attack this problem it seems logical that the optimal control theory is one of the most promising methods. Recently, several researchers have developed optimal washout filters [2,3]. As pointed out by J.Sandor and D.Williamson [15], to achieve the desired control for this kind of system certain states should be penalized more heavily. Conventionally this can be done by choosing appropriate weighting of the states in the performance index. But unfortunately, this approach may often lead to "ill conditioned" linear feedback gain which can sometimes destabilize the system. Further study has revealed that the desired process should be highly non-linear. Relying on the application of linear control theory will not help very much to solve the problem.

In the current context, application of nonlinear optimal control theory implies construction of a nonlinear control input for a system which may not necessarily be a nonlinear system. In the following sections, for simplicity, we assume that the controlled system is linear. The detailed development is described below. Examples for the three response channels are given.

6.1 Theoretical Development

It is observed that practical problems of feedback control frequently involve specifications which cannot be met by purely linear designs. For example, soft-saturate type constraints are often imposed on certain state variables such as velocities and accelerations.

For completeness, the following definitions are given for readers.

Definition 1.(square integrable function) [26]

A function $f \varepsilon R$ is said to be square integrable if

$$f \cdot |f| \; \varepsilon \; \mathcal{L}^1(0, \infty; \; R^{\Omega})$$

where

$$\mathcal{L}^1(0, \infty; R^{\Omega}) \triangleq \left\{ g \,\middle|\, 0 < \int_{-\infty}^{+\infty} g\,dm < +\infty, \text{ for all } m \in R^{\Omega} \right\}$$

The set of all square integrable functions ε R is denoted by $\mathcal{L}^2(0, \infty; R^{\Omega})$, where R^{Ω} is the set of all real numbers and includes $-\infty$ and $+\infty$.

Loosely speaking, if a function f is squared, and the integral satisfies the following relation

$$\infty < \int_{-\infty}^{+\infty} f^2 dm < +\infty$$

we then call f a "square integrable function". □

To design an asymptotically stabilizing nonlinear feedback law such that trajectories of the system are optimal in some sense, P.J.Moylan et al [16] established the following definition which will be helpful in the development.

Definition 2. (Return Difference Condition,R.D.C) [16]

Consider the controllable linear system

$$\dot{\underline{x}} = \underline{f}(\underline{x}, t) + \underline{B}\,\underline{u}(t) \tag{6.1}$$

with $\underline{x}(0) = \underline{x}_o$, $\underline{x}(t) \in R^n$, and $\underline{u}(t) \in R^m$.

A function $F: R^n \to R^m$ is said to satisfy the Return Difference Condition (R.D.C.), if

$$\int_0^{\infty} \|\underline{u}(t) + \underline{F}[\underline{x}(t)]\|^2 dt \geq \int_0^{\infty} \|\underline{u}(t)\|^2 dt \tag{6.2}$$

for all $u \in \mathcal{L}^2(0, \infty; R^m)$ generating a trajectory $\underline{x}(\cdot)$ of equation (6.1) with $\underline{x}(0) = \underline{x}_o = \underline{0}$ and $\lim_{t\to\infty} \underline{x}(t) = \underline{0}$ where $\mathcal{L}^2$ is a set of square integrable functions. □

To interpret equation (6.2), we may consider it as implying that a feedback law of $-\underline{F}[\underline{x}(t)]$ constitutes a negative feedback, with $\underline{u}_e$ denoting an external control applied to the system (6.1). $\underline{u}_e$ can be expressed as follows

$$\underline{u}_e = \underline{u}_L + \underline{F}(\underline{x}) \tag{6.3}$$

where $\underline{u}_L$ denotes a linear control input. This control structure is shown in Figure 6.1.

The importance of the R.D.C. is shown in the following theorem [16] which we adopt here without proof.

Theorem 1.

For the system (6.1), the asymptotically stable control law

$$\underline{u}(t) = -\underline{F}[\underline{x}(t)] \tag{6.4}$$

is optimal for problem of minimizing, subject to the boundary condition $\underline{x}(\infty) = \underline{0}$, a performance index of the form

$$J = \int_0^\infty [m(\underline{x}) + \underline{u}^T\underline{u}]dt \tag{6.5}$$

with $m(\underline{x})$ nonnegative for all $\underline{x}$, if and only if $\underline{F}(\underline{x})$ satisfies the R.D.C.. □

One of Lyapunov's theorems is very important in the design of the present nonlinear washout filter. We introduce it here, the proof of the theorem is quite lengthy. Interested readers may consult reference [17].

Theorem 2.(a theorem of Lyapunov)

For a system

$$\dot{\underline{x}} = \underline{A}\,\underline{x} + \underline{B}\,\underline{u}$$

Let $R(\lambda_i)$ denote the real part of the eigenvalues of the system matrix $\underline{A}$, if $R(\lambda_i) < 0$, for all $i \in \underline{n}$, and $\xi(\underline{x})$ is a definite form of even degree of m, then we define V by

$$\sum_{j=1}^{n} (a_{j1}x_1 + a_{j2}x_2 + \ldots + a_{jn}x_n) \frac{\partial V}{\partial x_j} = -\xi(\underline{x}) \tag{6.6}$$

where a_{jk}, $j\varepsilon\underline{n}$, $k\varepsilon\underline{n}$ are elements of $\underline{A}$.

The form V of the degree m defined above is also definite and of sign contrary to ξ. Especially, if $\xi > 0$ then $dV/dt < 0$, this implies that V is a Lyapunov function. □

With these theorems, now we consider the system

$$\dot{\underline{x}} = \underline{A}\,\underline{x} + \underline{B}\,\underline{u} + \underline{\Gamma}\,\underline{W}$$
$$\underline{y} = \underline{C}\,\underline{x} + \underline{D}\,\underline{u} \tag{6.7}$$

where

$$\underline{A} \in R^{n\times n}, \quad \underline{B} \in R^{n\times m}, \quad \underline{C} \in R^{q\times n}, \quad \underline{D} \in R^{q\times m}$$

$$\underline{x} \in R^{n}, \quad \underline{y} \in R^{q}, \quad \text{and} \quad \underline{u} \in R^{m}$$

It is convenient to assume that the system (6.7) is completely controllable and observable, i.e.

$$\text{Rank}[\underline{B} \mid \underline{A}\,\underline{B} \mid \ldots \mid \underline{A}^n\underline{B}] = n$$

and

$$\text{Rank}[\underline{C}^T \mid \underline{A}^T\underline{C}^T \mid (\underline{A}^T)^2\underline{C}^T \mid \ldots \mid (\underline{A}^T)^n\underline{C}^T] = n$$

From lemma 1 [15] and using the definition and the theorems given above, we have the following corollary.

Corollary

Find $\underline{F}$ such that $\underline{A}_o = \underline{A} - \underline{B}\,\underline{F}$ and $\sigma(\underline{A}_o) \in C^-$, where $\sigma(\cdot)$ denotes the spectrum or eigenvalues of a matrix, C^- denotes the left half of the complex plane.

Consider the nonlinear function

$$\underline{k}(\underline{x}) = -\underline{R}^{-1}\underline{B}^T\underline{\nabla_x}V \tag{6.8}$$

where $\underline{R} > 0$, $\underline{R} = \text{diag}[r_{11}, r_{22}, \ldots r_{mm}]$; V is the solution of the following partial differential equation

$$\langle\underline{\nabla_x}V, \underline{A}_o\underline{x}\rangle = -\xi(\underline{x}) \tag{6.9}$$

for some nonnegative definite homogeneous form $\xi(\underline{x})$ of even degree, and $\underline{\nabla_x}V \triangleq \partial V/\partial\underline{x}$. Then $\underline{k}(\underline{x})$ satisties the R.D.C.. For

$$\dot{\underline{x}} = \underline{A}_o\underline{x} - \underline{B}\,\underline{u}_{NL} \tag{6.10}$$

where $\underline{u}_{NL} \triangleq -\underline{k}(\underline{x})$, the solution $\underline{x}(\cdot)$ is asymptotically stable and $\underline{u}_{NL}$ minimizes the performance index

$$J = \int_0^\infty \left[\frac{1}{2}(\underline{x}^T\underline{Q}\,\underline{x} + \underline{u}^T\underline{R}\,\underline{u}) + \xi(\underline{x})\right] dt \tag{6.11}$$

Proof:

By the Lyapunov theorem (given as theorem 2.), and the relation (6.9), we know that V is a Lyapunov function, and for the system (6.10), we assume that

$$\dot{V} = \left(\frac{\partial V}{\partial \underline{x}}\right)^T \frac{\partial x}{\partial t} = \left(\frac{\partial V}{\partial \underline{x}}\right)^T \dot{\underline{x}} = \left(\frac{\partial V}{\partial \underline{x}}\right)^T (\underline{A_o x} - \underline{B}\ \underline{u}_{NL})$$

$$= -\xi - [(\underline{\nabla_x V})^T \underline{B}]\underline{R}^{-1}[\underline{B}^T \underline{\nabla_x V}]$$

$$= -[\xi + \underline{W}^T \underline{R}^{-1} \underline{W}]$$

where $\underline{W} \triangleq \underline{B}^T \underline{\nabla_x V}$.

Since it is assumed that $\xi(\underline{x}) \geqq 0$, $\underline{R} > 0$ and $R=\mathrm{diag}[r_{11}, r_{22}, \ldots, r_{mm}]$, therefore $\underline{W}^T \underline{R}^{-1} \underline{W} > 0$, this implies that

$$\dot{V} < 0$$

which confirms that the solution for equation (6.10) is aymptotically stable.

Now we need to prove that with this $\underline{k}(\underline{x})$, the R.D.C. is satisfied. We construct a functional as follows

$$\int_0^\infty \underline{v}^T \underline{v}\, dt \leqq \int_0^\infty [\underline{v}^T \underline{v} + 2\xi(\underline{x}) + \underline{W}^T \underline{R}^{-1} \underline{W}]dt$$

$$= \int_0^\infty (\underline{v}^T \underline{v} - 2 \langle \underline{\nabla_x V}, \underline{A_o x}\rangle + \underline{W}^T \underline{R}^{-1} \underline{W})dt \qquad (6.12)$$

along the trajectories

$$\dot{\underline{x}} = \underline{A_o x} - [\underline{B}\ \underline{R}^{-1} \underline{B}^T \underline{\nabla_x V} + \underline{B}\ \underline{R}^{-1} \underline{v}] \qquad (6.13)$$

This form is valid, because of the controllability of the system.

The following relation is then verified

$$\langle \underline{\nabla_x V}, \underline{A_o x}\rangle = \langle \underline{\nabla_x V}, \dot{\underline{x}}\rangle + \underline{W}^T \underline{R}^{-1} \underline{W} + \langle \underline{\nabla_x V}, \underline{B}\ \underline{R}^{-1} \underline{v}\rangle$$

or

$$\langle \underline{\nabla_x V}, \underline{A_o x}\rangle = \frac{dV}{dt} + \underline{W}\ \underline{R}^{-1} \underline{W} + \langle \underline{\nabla_x V}, \underline{B}\ \underline{R}^{-1} \underline{v}\rangle \qquad (6.14)$$

Upon subtituting equation (6.14) into equation (6.12), we get

$$\int_0^\infty \underline{v}^T\underline{v}\, dt \leq \int_0^\infty (\underline{v}^T\underline{v} - 2\langle \underline{\nabla_x V}, \underline{B}\,\underline{R}^{-1}\underline{v}\rangle - \underline{W}^T\underline{R}^{-1}\underline{W})dt$$

$$- 2[V(t_\infty) - V(t_o)]$$

Note that $V(t_\infty) = V(t_o) = 0$, therefore

$$\int_0^\infty \underline{v}^T\underline{v}\, dt \leq \int_0^\infty (\underline{v}^T\underline{v} - 2\langle \underline{\nabla_x V}, \underline{B}\,\underline{R}^{-1}\underline{v}\rangle - \underline{W}^T\underline{R}^{-1}\underline{W})dt \tag{6.15}$$

and

$$\int_0^\infty \|\underline{v} + \underline{k}(\underline{x})\|^2 dt = \int_0^\infty (\underline{v} - \underline{R}^{-1}\underline{B}^T\underline{\nabla_x V})^T(\underline{v} - \underline{R}^{-1}\underline{B}^T\underline{\nabla_x V})dt$$

$$= \int_0^\infty (\underline{v}^T\underline{v} - 2\underline{v}^T\underline{R}^{-1}\underline{B}^T\underline{\nabla_x V} + \underline{\nabla_x V}^T\underline{B}(\underline{R}^{-1})^T\underline{R}^{-1}\underline{B}^T\underline{\nabla_x V})dt \tag{6.16}$$

But $\underline{R}$ is a diagonal, mxm matrix, so is $\underline{R}^{-1}$, that is $(\underline{R}^{-1})^T = \underline{R}^{-1}$, and note that $\underline{W} = \underline{B}^T\underline{\nabla_x V}$. Therefore equation (6.16) becomes

$$\int_0^\infty \|\underline{v} + \underline{k}(\underline{x})\|^2 dt = \int_0^\infty [\underline{v}^T\underline{v} - 2\langle \underline{\nabla_x V}, \underline{B}\,\underline{R}^{-1}\underline{v}\rangle + \underline{W}^T(\underline{R}^{-1})^2\underline{W}]dt \tag{6.17}$$

Since $\underline{R}>0$, we have $\underline{R}^{-1} > 0$, and $\underline{W}^T(\underline{R}^{-1})^2\underline{W} > 0$. From equations (6.15) and (6.17) the following is true

$$\int_0^\infty \underline{v}^T\underline{v} \leq \int_0^\infty [\underline{v}^T\underline{v} - 2\langle \underline{\nabla_x V}, \underline{B}\,\underline{R}^{-1}\underline{v}\rangle + \underline{W}^T(\underline{R}^{-1})^2\underline{W}]dt$$

$$= \int_0^\infty \|\underline{v} + \underline{k}(\underline{x})\|^2 dt$$

By theorem 1 and the conclusion of [16], the results are extended to the case where $\underline{u}^T\underline{u}$ is replaced by $\frac{1}{2}\underline{u}^T\underline{R}\underline{u}$ in equation (6.5).

Since

$$m(\underline{x}) = \xi(\underline{x}) + \frac{1}{2}\,\underline{x}^T\underline{Q}\,\underline{x} > 0$$

therefore the control law

$$\underline{u}_{NL} = \underline{R}^{-1}\underline{B}^T\underline{\nabla_x V}$$

minimizes the performance index J. □

As done in reference [15], we also make use of the notation of $\underline{x}^{[j]}$. For detailed explanation of $\underline{x}^{[j]}$ please see Appendix B. The lemma below is helpful in designing the controller.

Lemma [6]

Consider the partial differential equation

$$\langle \underline{A}\,\underline{x}, \underline{\nabla_x V}\rangle = -\xi(\underline{x}) \tag{6.18}$$

where $\underline{A}$ is a stable matrix†, $\xi(\cdot)$ is a homogeneous function having the form

$$\xi(\underline{x}) = \sum_{j=2}^{m} \langle \underline{x}^{[j]}, \underline{Q}_j \underline{x}^{[j]}\rangle$$

for some choice of matrices $\underline{Q}_j$. Then there exists a solution

$$V(\underline{x}) = \sum_{j=2}^{m} \langle \underline{x}^{[j]}, \underline{p}_j \underline{x}^{[j]}\rangle$$

where $\underline{p}_j$ is a solution of the linear equation

$$\underline{A}^T_{[j]}\underline{p}_j + \underline{p}_j \underline{A}_{[j]} = -\underline{Q}_j \qquad \text{for } j = 1, 2, \ldots, n$$

where the definition of $\underline{A}_{[j]}$ is given in reference [15].

If $\underline{A}$ is strictly stable, then the $\underline{p}_j$ are unique, which in turn implies that $V(\underline{x})$ is unique. Furthermore, if $\xi(\underline{x})$ is nonnegative definite, so too is $V(\underline{x})$. □

6.2 Controller Design Procedures

With the results obtained in Section 6.1, we establish the following procedures for the design of nonlinear optimal washout filter.

Given the system

$$\dot{\underline{x}} = \underline{A}\,\underline{x} + \underline{B}\,\underline{u} + \underline{\Gamma}\,\underline{W}$$

$$\underline{y} = \underline{c}\,\underline{x} + \underline{D}\,\underline{u}$$

with $(\underline{A}, \underline{B})$ -controllable and $(\underline{C}, \underline{A})$ -observable, $\underline{W}$ is the disturbance vector.

---

† A nxn matrix is said to be stable, if $\sigma(A) \in C^-$. For a controllable system this assumption is always valid.

Step 1:

Check $\underline{A}$, see if it has the spectrum $\sigma(\underline{A}) \varepsilon C^{-}$, if not, construct a feedback matrix $\underline{F}$, such that

$$\sigma(\underline{A} - \underline{B}\ \underline{F}) = \{\lambda_1, \lambda_2, \ldots, \lambda_n\} \varepsilon C^{-}$$

where λ_i are the desired eigenvalues of $\underline{A}-\underline{B}\ \underline{F}$.

Now the system has the form

$$\underline{\dot{x}} = (\underline{A} - \underline{B}\ \underline{F})\underline{x} + \underline{B}\ \underline{u} + \underline{\Gamma}\ \underline{W}$$

Since the system is completely controllable, such exists [18].

Step 2:

For optimal control, we need to solve the following algeraic Riccati equation

$$\underline{A}_o^T \underline{P} + \underline{P}\ \underline{A}_o - \underline{P}\ \underline{B}\ \underline{R}^{-1}\underline{B}^T\underline{P} + \underline{Q} = \underline{0}$$

where $\underline{A}_o \triangleq \underline{A}-\underline{B}\ \underline{F}$, and $\underline{F}$ is given in step 1.

Step 3:

Find $\underline{\tilde{F}}$ of the form

$$\underline{\tilde{F}} = -\underline{R}^{-1}\underline{B}^T\underline{P}$$

Step 4:

Reconstruct the system matrix with the new $\underline{\tilde{F}}$:

$$\underline{\tilde{A}}_o = \underline{A}_o - \underline{B}\ \underline{\tilde{F}}$$

Step 5:

Construct a cost function according to the corollary

$$J = \int_0^\infty \left[\frac{1}{2} (\underline{x}^T\underline{Q}\ \underline{x} + \underline{u}^T\underline{R}\ \underline{u}) + \xi(\underline{x}) \right] dt$$

where $\underline{Q}$, $\underline{R}$, and $\xi(\underline{x})$ are determined accordingly.

Step 6:

According to the given system and $\xi(\underline{x})$, define a Lyapunov function with its coefficients to be determined later.

Step 7:
Solve for the coefficients of $V(\underline{x})$ from the equation

$$<(\underline{A}_o - \underline{B}\,\underline{\tilde{F}})\underline{x},\ \underline{\nabla_x V}> = -\xi(\underline{x})$$

Step 8:
Construct the control

$$\underline{u} = \underline{u}_L + \underline{u}_{NL} = -\underline{R}^{-1}\underline{B}^T\underline{p}\,\underline{x} - \underline{R}^{-1}\underline{B}^T\underline{\nabla_x V}$$

Step 9:
Rewrite the closed loop system as

$$\dot{\underline{x}} = (\underline{A}_o - \underline{B}\,\underline{\tilde{F}})\underline{x} - \underline{B}\,\underline{R}^{-1}\underline{B}^T\underline{\nabla_x V} + \underline{\Gamma}\,\underline{W}$$

$$\underline{y} = \underline{C}\,\underline{x} + \underline{D}\,\underline{u}$$

Solving for $\underline{x}$, we finally get the controlled trajetories.

To intuitively illustrate the design procedures, a flow chart is depicted in Figure 6.2.

6.3 Formulation of the Washout filter

In the two preceding sections we introduced the theoretical development and the optimal control system.

We will use all the results obtained to formulate the nonlinear optimal filter in this section.

6.3.1 Motion Cue Generation

From the work done by Schmindt and Conard, we know that the specific force vector and angular velocity in the cab frame of the simulated aircraft can be represented as follows

$$\underline{f}_c = \underline{L}_{c_i}(\underline{\ddot{r}}_{c_i} - \underline{g}) = \underline{L}_{c_i}\underline{\ddot{r}}_{c_i} - \underline{L}_{c_i}\underline{g} = \underline{A}_c - \underline{L}_{c_i}\underline{g} \qquad (6.19)$$

$$\underline{\omega}_c = \underline{R}_T\dot{\underline{\beta}}_c \qquad (6.20)$$

where

$\dot{\underline{\beta}}_c^T = [\dot{\phi}_c\ \dot{\theta}_c\ \dot{\psi}_c]$, $\underline{g}^T = [0\ 0\ g]$, and $g = 9.81\ m/sec^2$

The specific force vector and angular velocity in the cab frame of the simulator have the forms similar to equations (6.19) and (6.20). For completeness, the we give the expression as follows,

$$\hat{\underline{f}}_c = \hat{\underline{L}}_{c_i}(\hat{\ddot{\underline{r}}}_{c_i} - \underline{g}) \tag{6.21}$$

$$\hat{\underline{\omega}}_c = \hat{\underline{R}}_T \, \hat{\dot{\underline{\beta}}}_c \tag{6.22}$$

As mentioned before, it is desired that $\hat{\underline{f}}_c$ and $\hat{\underline{\omega}}_c$ are generated as close to $\underline{f}_c$ and $\underline{\omega}_c$ respectively as possible. But due to the physical constraints of the simulator motion base, we know that it is not practical to have them identical. Therefore we will choose

$$\hat{\underline{f}}_c = \underline{f}_c + \underline{\varepsilon}_1 \tag{6.23}$$

$$\hat{\underline{\omega}}_c = \underline{\omega}_c + \underline{\varepsilon}_2 \tag{6.24}$$

and constrain the $\underline{\varepsilon}_1$ and $\underline{\varepsilon}_2$ such that motion base excursion is limited by defining the cost function with the form

$$J = \int_0^\infty \frac{1}{2} (\underline{x}^T \underline{Q}\, \underline{x} + \underline{u}^T \underline{R}\, \underline{u}) + \xi(\underline{x}) \quad dt \tag{6.25}$$

where

$$\underline{u} \triangleq \begin{bmatrix} \underline{\varepsilon}_1 \\ \underline{\varepsilon}_2 \end{bmatrix}, \quad \underline{x} \triangleq \begin{bmatrix} \hat{\underline{r}}_{c_i} \\ \hat{\underline{v}}_{c_i} \\ \hat{\underline{\beta}}_{c_i} \end{bmatrix}, \quad \hat{\dot{\underline{r}}}_{c_i} \triangleq \hat{\underline{v}}_{c_i}$$

Upon substituting equations (6.19),(6.20),(6.21),and (6.22)into equations (6.23) and (6.24), we,have

$$\hat{\dot{\underline{r}}}_{c_i} = \hat{\underline{L}}_{c_i}^{-1}(\underline{L}_{c_i}\, \ddot{\underline{r}}_{c_i} + \underline{\varepsilon}_1) - \hat{\underline{L}}_{c_i}^{-1}\, \underline{L}_{c_i}\, \underline{g} + \underline{g} \tag{6.26}$$

$$\hat{\dot{\underline{\beta}}}_{c_i} = \hat{\underline{R}}_T^{-1}(\underline{\omega}_c + \underline{\varepsilon}_2) \tag{6.27}$$

and from equation (3.6), equation (6.26) becomes

$$\hat{\ddot{\underline{r}}}_{c_i} = \hat{\underline{L}}_{c_i}^{-1}(\underline{A}_c + \underline{\varepsilon}_1) - [\tilde{\underline{L}}_{c_i} - \underline{I}]\underline{g} \qquad (6.26a)$$

where $\hat{\underline{L}}_{c_i}$ is obtained from equation (3.1) by replacing ϕ, θ, ψ with $\hat{\phi}_c$, $\hat{\theta}_c$, $\hat{\psi}_c$ respectively, and $\tilde{\underline{L}}_{c_i} \triangleq \hat{\underline{L}}_{c_i}^{-1} \underline{L}_{c_i}$

$\hat{\underline{f}}_c \triangleq [\hat{f}_{cx}\ \hat{f}_{cy}\ \hat{f}_{cz}]^T$, $\underline{A}_c = [a_{cx}, a_{cy}, a_{cz}]^T$, and $\underline{\omega}_c = [P_c, Q_c, R_c]^T$

6.3.2 Linearization

It is clear that equations (6.26) and (6.27) are nonlinear and time variable. For simplicity, we linearize (6.26) and (6.27) about the equilibrium states

$$\hat{\underline{r}}_{c_i}(e) = \underline{0}, \quad \hat{\dot{\underline{r}}}_{c_i}(e) = \underline{0}, \quad \hat{\theta}_c(e) = \hat{\phi}_c(e) = \hat{\psi}_c(e) = 0$$

and $\underline{A}_c$, $\underline{\varepsilon}_1$, $\underline{\varepsilon}_2$, $\underline{\omega}_c$, and $\hat{\underline{R}}_T^{-1}$ are also taken to be linearized about the equilibrium point.

Then

$$\tilde{\underline{L}}_{c_i} = \begin{bmatrix} 1 & \psi_c - \hat{\psi}_c & -(\theta_c - \hat{\theta}_c) \\ -(\psi_c - \hat{\psi}_c) & 1 & \phi_c - \hat{\phi}_c \\ \theta_c - \hat{\theta}_c & -(\phi_c - \hat{\phi}_c) & 1 \end{bmatrix} \qquad (6.28)$$

The equations (6.26) and (6.27) then become

$$\hat{\ddot{\underline{r}}}_{c_i} = \underline{A}_c + \underline{\varepsilon}_1 - \tilde{\underline{g}}(\underline{\beta}_c - \hat{\underline{\beta}}_c) \qquad (6.29)$$

where

$$\tilde{\underline{g}} = \begin{bmatrix} 0 & -g & 0 \\ g & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

and

$$\hat{\dot{\underline{\beta}}}_c = \underline{\omega}_c + \underline{\varepsilon}_2 \qquad (6.30)$$

6.3.3 State Space Representation

It is convenient to treat the problem in state space. Here we define

$$\hat{\dot{\underline{r}}}_{c_i} \triangleq \hat{\underline{v}}_{c_i}$$

Then we construct the state vector

$$\underline{x} = \begin{bmatrix} \hat{\underline{r}}_{c_i} \\ \hat{\underline{v}}_{c_i} \\ \hat{\underline{\beta}}_c \end{bmatrix}$$

Therefore state equation for the linearized system is

$$\dot{\underline{x}} = \underline{A}\,\underline{x} + \underline{B}\,\underline{u} + \underline{\Gamma}\,\underline{W}$$

where

$$\underline{A} \in R^{9\times 9}, \quad \underline{B}, \underline{\Gamma} \in R^{9\times 6}; \quad \underline{u}, \underline{W} \in R^{6\times 1}$$

with

$$\underline{A} \triangleq \begin{bmatrix} \underline{0} & \underline{I} & \underline{0} \\ \underline{0} & \underline{0} & \underline{\tilde{g}} \\ \underline{0} & \underline{0} & \underline{0} \end{bmatrix}, \qquad \underline{B} \triangleq \begin{bmatrix} \underline{0} & \underline{0} \\ \underline{I} & \underline{0} \\ \underline{0} & \underline{I} \end{bmatrix} \triangleq \underline{\Gamma}$$

$$\underline{u} = \begin{bmatrix} \underline{\varepsilon}_1 \\ \underline{\varepsilon}_2 \end{bmatrix}, \qquad \underline{W} \triangleq \begin{bmatrix} \underline{A}_c + \underline{\tilde{g}}\,\underline{\beta}_c \\ \underline{\omega}_c \end{bmatrix}, \qquad \underline{\varepsilon}_1 = \begin{bmatrix} f_{x_e} \\ f_{y_e} \\ f_{z_e} \end{bmatrix}, \qquad \underline{\varepsilon}_2 = \begin{bmatrix} \omega_{y_e} \\ \omega_{x_e} \\ \omega_{z_e} \end{bmatrix}$$

or explicitly expressed as follows

$$\begin{bmatrix} \dot{\hat{x}}_c \\ \dot{\hat{y}}_c \\ \dot{\hat{z}}_c \\ \dot{\hat{v}}_x \\ \dot{\hat{v}}_y \\ \dot{\hat{v}}_z \\ \dot{\hat{\phi}}_c \\ \dot{\hat{\theta}}_c \\ \dot{\hat{\psi}}_c \end{bmatrix} = \begin{bmatrix} 0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&-g&0 \\ 0&0&0&0&0&0&g&0&0 \\ 0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0 \end{bmatrix} \begin{bmatrix} \hat{x}_c \\ \hat{y}_c \\ \hat{z}_c \\ \hat{v}_x \\ \hat{v}_y \\ \hat{v}_z \\ \hat{\phi}_c \\ \hat{\theta}_c \\ \hat{\psi}_c \end{bmatrix} + \begin{bmatrix} 0&0&0&0&0&0 \\ 0&0&0&0&0&0 \\ 0&0&0&0&0&0 \\ 1&0&0&0&0&0 \\ 0&1&0&0&0&0 \\ 0&0&1&0&0&0 \\ 0&0&0&1&0&0 \\ 0&0&0&0&1&0 \\ 0&0&0&0&0&1 \end{bmatrix} \begin{bmatrix} f_{xe} \\ f_{ye} \\ f_{ze} \\ \omega_{ye} \\ \omega_{xe} \\ \omega_{ze} \end{bmatrix}$$

Contd...

$$
+\begin{bmatrix} 0&0&0&0&0&0\\ 0&0&0&0&0&0\\ 0&0&0&0&0&0\\ 1&0&0&0&0&0\\ 0&1&0&0&0&0\\ 0&0&1&0&0&0\\ 0&0&0&1&0&0\\ 0&0&0&0&1&0\\ 0&0&0&0&0&1 \end{bmatrix}\begin{bmatrix} a_{cx}+g\theta_c\\ a_{cy}-g\phi_c\\ a_{cz}\\ Q_c\\ P_c\\ R_c \end{bmatrix} \tag{6.31}
$$

which can be further partitioned into three subsystems each representing a channel.

1) Longitudinal Subsystem

$$
\begin{bmatrix} \dot{\hat{x}}_c\\ \dot{\hat{v}}_x\\ \dot{\hat{\theta}}_c \end{bmatrix} = \begin{bmatrix} 0&1&0\\ 0&0&-g\\ 0&0&0 \end{bmatrix}\begin{bmatrix} \hat{x}_c\\ \hat{v}_x\\ \hat{\theta}_c \end{bmatrix} + \begin{bmatrix} 0&0\\ 1&0\\ 0&1 \end{bmatrix}\begin{bmatrix} f_{xe}\\ \omega_{xe} \end{bmatrix} + \begin{bmatrix} 0&0\\ 1&0\\ 0&1 \end{bmatrix}\begin{bmatrix} a_{cx}+g\phi_c\\ Q_c \end{bmatrix}
$$

or

$$
\dot{\underline{x}}_1 = \underline{A}_1\underline{x}_1 + \underline{B}_1\underline{u}_1 + \underline{\Gamma}_1\underline{W}_1 \tag{6.32}
$$

2) Lateral subsystem

$$
\begin{bmatrix} \dot{\hat{y}}_c\\ \dot{\hat{v}}_y\\ \dot{\hat{\phi}}_c \end{bmatrix} = \begin{bmatrix} 0&1&0\\ 0&0&g\\ 0&0&0 \end{bmatrix}\begin{bmatrix} \hat{y}_c\\ \hat{v}_y\\ \hat{\phi}_c \end{bmatrix} + \begin{bmatrix} 0&0\\ 1&0\\ 0&1 \end{bmatrix}\begin{bmatrix} f_{ye}\\ \omega_{ye} \end{bmatrix} + \begin{bmatrix} 0&0\\ 1&0\\ 0&1 \end{bmatrix}\begin{bmatrix} a_{cy}-g\phi_c\\ P_c \end{bmatrix}
$$

or

$$
\dot{\underline{x}}_2 = \underline{A}_2\underline{x}_2 + \underline{B}_2\underline{u}_2 + \underline{\Gamma}_2\underline{W}_2 \tag{6.33}
$$

3) Vertical subsystem

$$
\begin{bmatrix} \dot{\hat{z}}_c\\ \dot{\hat{v}}_z\\ \dot{\hat{\psi}}_c \end{bmatrix} = \begin{bmatrix} 0&1&0\\ 0&0&0\\ 0&0&0 \end{bmatrix}\begin{bmatrix} \hat{z}_c\\ \hat{v}_z\\ \hat{\phi}_c \end{bmatrix} + \begin{bmatrix} 0&0\\ 1&0\\ 0&1 \end{bmatrix}\begin{bmatrix} f_{ze}\\ \omega_{ze} \end{bmatrix} + \begin{bmatrix} 0&0\\ 1&0\\ 0&1 \end{bmatrix}\begin{bmatrix} a_{cz}\\ R_c \end{bmatrix}
$$

or

$$\dot{\underline{x}}_3 = \underline{A}_3\underline{x}_3 + \underline{B}_3\underline{u}_3 + \underline{\Gamma}_3\underline{W}_3 \tag{6.34}$$

A quick inspection shows that all these three subsystems are controllable.

6.3.4 The Optimal Washout Filter

In Sections 6.3.2 and 6.3.3 we derived the system state space equations. All these equations were linearized and decoupled, which allows us to construct the nonlinear optimal washout filters channel-wise. Later we will see that decoupling is very important in applying this approach.

1) Longitudinal Washout Filter

For the system given by equation (6.29) we construct the following cost function

$$J = \int_0^\infty \left[\frac{1}{2} (\underline{x}_1^T \underline{Q}\, \underline{x}_1 + \underline{u}_1^T \underline{R}\, \underline{u}_1) + \xi(\underline{x}_1) \right] dt \tag{6.35}$$

which has the same form as (6.11), where

$$\underline{R} = \text{diag}\left[\frac{1}{R_{11}^2}, \frac{1}{R_{22}^2} \right]$$

$$\underline{Q} = \text{diag}\left[\frac{a_{11}}{x_L^2}, \frac{a_{22}}{V_L^2}, \frac{a_{33}}{\theta_L^2} \right]. \qquad \{R_{ii},\ i = 1, 2\},\ \{a_{jj},\ j = 1, 2, 3\}$$

x_L, V_L and θ_L are determined by the designer.

Let

$$\xi(\underline{x}_1) = \frac{a_o}{2} \left[\left(\frac{\hat{x}_c}{x_L} \right)^4 + \left(\frac{\hat{v}_x}{v_L} \right)^4 + \left(\frac{\hat{\theta}_c}{\theta_L} \right)^4 \right] \tag{6.36}$$

Looking at $\underline{A}_1$, we see that

$$\sigma(\underline{A}_1) = \{0, 0, 0\} \notin C^-$$

i.e. the system is at the critical point. In order to solve the Riccati equation we need to make it stable. It is convenient to assume that the desired eigenvalues are

$$\sigma(\underline{A}_o) = \{-1, -1, -1\}$$

Using the method introduced by Wonham [18], for this simple system, a short calculation yields (a computer program is available for this manipulation and is listed in Appendix D),

$$\underline{F} = \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \end{bmatrix}$$

Then the new system matrix becomes

$$\underline{A}_o = \underline{A}_1 - \underline{B}_1\underline{F}$$

Now the algebraic Riccati equation

$$\underline{A}_o^T \underline{P} + \underline{P}\,\underline{A}_o - \underline{P}\,\underline{B}\,\underline{R}^{-1}\underline{B}^T\underline{P} + \underline{Q} = \underline{0}$$

can be solve by a modified Riccati subroutine which gives $\underline{P}$ and $\underline{\tilde{F}}$ in return (see Appendix D), with

$$\underline{\tilde{F}} = -\underline{R}^{-1}\,\underline{B}_1^T\,\underline{P}$$

and

$$\sigma(\underline{A}_o - \underline{B}_1\,\underline{\tilde{F}}) \in C^-$$

Using the Lyapunov function form of equation (6.15)

$$V(\underline{x}) = a_1\hat{x}_c^{\,4} + a_2\hat{x}_c^{\,3}\hat{\theta}_c + a_3\hat{x}_c^{\,2}\hat{\theta}_c\hat{v}_x + a_4\hat{x}_c\hat{\theta}_c^{\,2}v_x + a_5\hat{\theta}_c^{\,2}\hat{v}_x^{\,2}$$

$$+ a_6\hat{\theta}_c^{\,3}\hat{v}_x + a_7\hat{\theta}_c^{\,4} + a_8\hat{\theta}_c^{\,3}\hat{x}_c + a_9\hat{\theta}_c^{\,2}\hat{x}_c^{\,2} + a_{10}\hat{\theta}_c\hat{x}_c\hat{v}_x^{\,2}$$

$$+ a_{11}\hat{x}_c^{\,3}\hat{v}_x + a_{12}\hat{x}_c^{\,2}\hat{v}_x^{\,2} + a_{13}\hat{x}_c\hat{v}_x^{\,3} + a_{14}\hat{\theta}_c\hat{v}_x^{\,3} + a_{15}\hat{v}_x^{\,4} \qquad (6.37)$$

The coefficients a_i $(i = 1,2,3,\ldots,15)$ are to be determined, that is, we have to solve 15 equations! For this simple third order system the problem is not severe, but for systems with the order of five, for instance, there will be seventy unknown coefficients, to determine these coefficients uniquely, seventy equations have to be solved! (The number of equations to solved can be determined by the formula

$$N = \frac{(n+3)!}{(n-1)!\,4!}$$

where n is the order of a given system.) Even for a computer, this is an awesome number. This explains why decouplability of a system is important in using this method, which is a significant disadvantage.

Now we use the equation

$$<(\underline{A}_o - \underline{B}_1 \underline{\tilde{F}})\underline{x}_1 , \underline{\nabla_x V}> = -\xi(\underline{x})$$

by equating the coefficients of terms of same order on both sides, we get fifteen equations which are solved for the a_i .

The controlled system is then reconstructed having a form like the one given in Step 9 of Section 6.2.

2) Lateral and vertical washout filters

Following exactly the same way we synthesize nonlinear optimal washout filters for lateral and vertical channels.

6.4 Computational Considerations

The proposed nonlinear optimal washout filter is expected to be implemented by a real-time mini-computer. As usual, one of the main concerns in real-time digital computer control is the computational feasibility. The algorithms should be so compact that they can be implemented using a very short sampling period [19], and the configurations of the algorithms should not require too large an amount of memory. These problems have existed with modern control practice for many years. Generally speaking, most modern control methodologies depend on the digital computer, some of them, for instance the Kalman filter, were even tailored to be implemented on the computer. Due to the intricacies of control systems, algorithms developed from these theories are often not feasible in practice, and special treatment or modification [21] is often needed. Fortunately, the system studied in this chapter is, at least for the time being, simplified, linear, and time invariant, therefore the Riccati equations are algebraic and can be solved off-line and have constant solutions throughout the control process.

To illustate the problem, the discrete counterpart to the continuous systems is as follows

$$\underline{x}(k+1) = \underline{A}\ \underline{x}(k) + \underline{B}\ \underline{u}(k) + \underline{\Gamma}\ \underline{W}(k) \tag{6.38}$$

where $\underline{x}(k)$, $\underline{u}(k)$, and $\underline{W}(k)$ denote the values sampled at the time t_k .

The cost function is

$$J = \sum_{j=0}^{N} \left\{ \frac{1}{2} [\underline{x}^T(j)\underline{Q}\ \underline{x}(j) + \underline{u}^T(j)\underline{R}\ \underline{u}(j)] + \xi[\underline{x}(j)] \right\} \tag{6.39}$$

where $\underline{Q} \geq 0$, $\underline{R} > 0$ and $\xi[\underline{x}(j)]$ are the same as in the previous sections.

Since the system is supposed to be linear and time invariant, the method described in Section 6.2 can be directly applied to this case to solve for the coefficients in $V(\underline{x})$. In this way $V(\underline{x})$ and $\underline{\nabla_x V}$ can be formulated beforehand. With the system and the cost function given in equations (6.38) and (6.39), the following control input for the discrete system results,

$$\underline{u}(k) = \underline{u}_L(k) + \underline{u}_{NL}(k) = -\underline{R}^{-1}\,\underline{B}^T\,\underline{P}\,\underline{x}(k) - \underline{R}^{-1}\,\underline{B}^T\,\underline{\nabla_x V(k)} \qquad (6.40)$$

This is eventually used to drive the motion base. For illustrative purposes, the general scheme for the control system is depicted in Figure 6.3.

It is clear that when this control system is implemented in real-time, the main task for the computer is to manipulate some multiplications and additions which are not considered to be a heavy burden, and therefore should not introduce significant delay into the simulation.

As the simulator is a complex system which involves several parts to be controlled by the computer, consideration of computational aspects for the simulator is a topic open for further studies.

6.5 Tests and Discussion

6.5.1 Selection of Weighting Matrices $\underline{R}$ and $\underline{Q}$

It is evident that the solution of the Riccati equation is closely dependent on the weighting matrices $\underline{R}$ and $\underline{Q}$ in the cost function. Despite years of theoretical research and steadily growing lore of applications, so little is known about the relationships between the weighting matrices and specific criteria (the cost functions) that the designer must invariantly resort to trial and error iterations.

To solve this problem, serious researchers have devised various intuitive ways to "select quadratic weights". These range from the simple diagonal inverse-square weighting approach of Bryson [20] , to local quadratic equivalence methods [21], and various versions of model-following [22,23,24] among which Bryson's method is considered most general and popular. In the investigation of the proposed design approach, Bryson's method was adopted to choose the weighting matrices because the system is simple and well defined.

In each channel four different weighting matrices were chosen. The parameters selected are summarized in Table 3 of Appendix C.

The performance characteristics of the nonlinear optimal washout filter are discussed in the next section along with the linear and adaptive washout filters.

6.5.2 Properties of the Nonlinear Optimal Washout Filter

To explore the differences among the three types of washout filter in terms of system time responses to step inputs, the responses of the filters are plotted in Figures 6.4~6.39. For clarity the inputs used in different cases are given in Appendix C. The parameters selected for the linear washout filter and the adaptive washout filter are also summarized in Appendix C.

Due to the linearity of the washout filter, Figures 6.4~6.7 show that the time responses for each case always vanish (i.e. become zero) at the same time. This means that the controlled system will be driven to move for the same time duration regardless of the magnitude of input. As described at the beginning of this chapter, this phenomenon is due to the intrinsic properties of linear systems, and is considered inefficient, as it often makes the already poor performance of the simulator even poorer.

The responses of the adaptive washout filter to different inputs are shown in Figures 6.8~6.19, which reveal the nonlinearity of the control system. In this case, the responses to different magnitudes of inputs no longer vanish at the same point, but owing to the large number of parameters required, it is very hard to obtain the desired responses.

In Figure 6.40 the responses of $\hat{\theta}_c$ and $\hat{\dot{\theta}}_c$ to step inputs $f_x = 0.6$ m/sec^2, $f_y = 0.5$ m/sec^2 , and $\dot{\theta}_c = 0.16$ rad/sec are depicted. It is clear that for $\hat{\dot{\theta}}_c$ the adaptive washout filter responds as an exponential function. $\hat{\dot{\theta}}_c$ becomes zero and $\hat{\theta}_c$ reaches its steady state of 0.16 rad in about 6 seconds. The adaptive control law has a tilt coordination feature (see Appendix A), that is, the rotational channel is coordinated with the translational acceleration to simulate steady state specific force. To explore this feature the response of $\hat{\theta}_c$ to $f_{ix} = 0.3g$ alone is given in Figure 6.41 which is obtained by setting the parameters $\eta_2 = 0.2$ and $\eta_3 = 0$ (see Appendix A for the corresponding equations). In this figure we can see that $\hat{\theta}_c$ is increasing with time though very slowly. From Figures 6.40~6.41, we find that the tilt angles are obtained by coordination of both force (or acceleration) and rotation cues in the adaptive washout filter.

Looking at the responses for the nonlinear optimal washout filter in Figures 6.20~6.31, we find an expected, interesting feature of the control system. Unlike the linear washout filters, the response durations are dependent on the magnitude of input. When the input is one g (9.18m /sec^2) the durations of the initial positive responses in all the three channels are longer than that for the three g input. Remarkably, the relative negative overshoots (see Appendix B for the definition) are much less than with both the linear and adaptive washout filters; for instance, for a step acceleration input of 3g the relative negative overshoot of the linear washout filter in Figure 6.4 is 0.22, for the adaptive filter in Figure 6.8 is o.34, and for the nonlinear optimal washout filter in Figure 6.20 is 0.17. The negative overshoot often causes confusion to the pilot, therefore the effort to reduce the negative motion cues to below the threshold of the pilot's perception is an important design specification for the simulator. For comparison, the relative negative overshoots for all the cases are summarized in Table 6.1.

Finally, it is interesting to compare the excursion responses of the linear and adaptive washout filters. In order to investigate the characteristics of the two filters in terms of excursion, for each filter four different sets of parameters are selected, and two step acceleration inputs with different magnitudes are fed into the filter for each set of parameters. It is observed from Figures 6.36~6.39 that for the adaptive washout filter the excursion profiles in response to the two different acceleration inputs are fairly close, while in Figures 6.32~6.35 for the linear washout filter the excursions are strongly related to the magnitudes of the inputs, and in the steady state the excursions are proportional to the inputs. It is also found that in Figures 6.32~6.39 the excursion responses of the linear washout filter for all the cases, except case 4, are much lower than that of the adaptive washout filter. As mentioned earlier, the response of the linear washout filter is always proportional to the input. If a step input with a fairly large magnitude is used then the excursion will exceed the limits of the motion base no matter what parameters have been chosen. But for the adaptive washout filter, if a set of parameters is carefully selected the excursion profiles can be controlled within the given limits (see Figure 6.39). This reveals that, owing to the strong adaptation characteristics, the adaptive washout filter may be used to control the motion base to remain well within the travel limits, and by proper selection of parameters better performance will be achieved.

CHAPTER VII

CONCLUSIONS AND RECOMMENDATIONS

A nonlinear optimal washout filter has been synthesized using techniques based on the nonlinear regulator and optimal control theories and tested on the computer. This proposed washout filter was superior to the conventional linear washout filter in that it provided different control signals for the system according to the input magnitudes such that the motion of the simulator was optimized by minimizing a given performance criterion.

It is also observed that the adaptive washout filter has very strong adaptation capability. It automatically changes the gain according to the input such that for different input levels, it can keep the excursion of the motion base fairly close. This indicates that by proper selection of the parameters, we may be able to control the simulator to achieve excellent performance.

In making use of the nonlinear optimal washout filter, the control system must be decoupled to avoid the generation of an enormous number of algebraic equations which are to be solved for the coefficients used in synthesizing the control system. As decoupling is a widely adopted technique in studying the behavior of flight vehicles (in normal performance), it will have no significant effects on many simulation applications.

The overall study indicates that the nonlinear optimal and the adaptive washout filters may be considered as the preferred options in generating washout filters.

It is recommended that for future research in this area, the following suggestions be considered

1) A human pilot model and gust model should be used to obtain a linear system which describes the stochastic properties of the desired specific forces and angular velocities.
2) As mentioned in 1), the input provided by the pilot can be highly random. In dealing with this sort of control problem, the multistage adaptive control theory [25] holds substantial prospects.
3) The dynamics of the hydraulic system should be included in the controlled system equations, which may result in the following nonlinear state equations

$$\dot{\underline{x}} = f(\underline{x}, t) + \underline{B}(t)\underline{u} + \underline{\Gamma}(t)\underline{W}$$

The optimal control for this system would be more difficult to implement on a real-time computer. The development of techniques to handle this case is a topic for further research.

REFERENCES

1. Schmidt, S.F. and Conard, B.,

   Motion Drive Signals for Piloted Flight Simulators

   NASA CR-1601, 1970

2. Kurosak, M.,

   Optimal Washout Filter for Control of a Motion Base Simulator

   Proc. Seventh Triennal World Congress of FFAC

3. Friedland, B.; King, C.K.; and Hotton, M.F.,

   Quasi-Optimum Design of a Six-Degree-of-Freedom Moving Base Simulator Control System

   NASA CR-2312, Otc. 1973

4. Parrish, R.V.; Dieudonne, J.E.; Martin, D.J., Jr.; and Bowles, R.L.,

   Coordinated Adaptive Filters for Motion Simulators

   Proc. of the 1973 Summer Computer Simulation Conference, Simulation Councils, Inc., C. 1973, pp.295-300

5. Young, L.R.; and Oman, C.M.,

   Model for Vestibular Adaptation to Horizontal Rotation

   Fourth Symp. on the Role of the Vestibular Organ in Space Exploration, NASA SP-187, 1970

6. Metry, J.L.,

   The Vestibular System and Human Space Orientation

   NASA CR-628, 1966

7. Peter, R.A.,

   Dynamics of the Vetibular System and Their Relation to

Motion Perception, Spatial Disorientation and Illusions

NASA CR-1309, 1969

8. Broxmeter, C.,

Inertial Navigation Systems

Electronic Sciences Series, McGraw-Hill Book Comp., 1964

9. Etkin, B.,

Dynamics of Atmospheric Flight

John Wiley Sons, Inc., 1972

10. Wong, Yong-Chu,

An Introduction to Design of Optimal Control Systems (in Chinese)

Academic Press, China, 1980

11. Landau, I.D.,

A Survey of Model Reference Adaptive Techniques----Theory and application

Automatica, Vol.10, pp.353-379

12. Narendra, K.S. and Valavani, L.S.,

Stable Adaptive Controller Design---Direct Control

IEEE Trans.on Automatic Control, Vol.AC-23, No.4, Aug.1978

13. Sage, A.P.,

Optimum Systems Control

Prentice-Hall, Inc., 1968

14. Coddington, E.A. and Levinson, N.,

Theory of Ordinary Differential Equations

McGraw-Hill Book Co., Inc., 1955

15. Sandor, J. and Williamson, D.,

Design of Nonlinear Regulators for Linear Plants

IEEE Trans.on Automatic Control, Vol.AC-22, No.1, Feb.1977

16. Moylan,P.J. and Anderson,B.D.O.,

<u>Nonlinear Regulator Theory and an Inverse Optimal Control Problem</u>

IEEE Trans.on Automatic Control,Vol.AC-18,No.5,Oct.1973

17. Cesari,L.,

<u>Asymptotic Behavior and Stability Problems in Ordinary Differential Equations</u>

Springer-Verlag, 1959

18. Wonham,W.M.,

<u>Linear Multivariable Control: A Geometric Approach</u>

Springer-Verlag,1979

19. Takeshi,T.,

<u>Improved Direct Digital Control Algorithm for Microprossesor Implementation</u>

IEEE Trans.on Automatic Control,Vol.AC-27,No.2,Apr.1982

20. Bryson,A.E.,and Ho,Y.C.,

<u>Applied Optimal Control</u>

Ginn and Company, 1969

21. Skelton,G.B.,

<u>Launch Booster Gust Alleviation</u>

3rd Annu. AIAA Meeting,Boston,MA,Nov.1966,paper 66-969

22. Tyler,J.S.Jr.,

<u>The Characteristics of Model Following Systems as Synthesized by Optimal Control</u>

IEEE Trans. on Automatic Control,Vol. AC-9, 1966

23. Sten,G. and Henke,A.H.,

<u>A Design Procedure and Handling Quality Criterion for Lateral-Directional Flight Control Systems</u>

May 1971, AFFDL-TR-70-152

24. Kreindler,E. and Rothchild, D.,
Model Following in Linear-Quadratic Optimization
AIAA J., Vol. 14,pp.835-842, July 1976

25. Dunn,H.J. and Montgomery, R.C.,
A Moving Window Parameter Adaptive Control System for the the F8-DFBW Aircraft
IEEE Trans.on Automatic Control,Vol.AC-22,No.5,Oct.1977

26. Jacobs,K.,
Measure and Integral
Academic Press,1978

APPENDIX A

EQUATIONS FOR THE ADAPTIVE WASHOUT FILTER [4]

1. Body to Inertial Transformation

(a) Specific force

In the adaptive washout filter design the behaviour of the system is studied in the inertial reference frame. We denote specific force in aircraft body axes F_B components as follows:

$$\underline{f}_B = \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} \doteq \begin{bmatrix} f_x \\ f_y \\ -g \end{bmatrix} \qquad (A.1)$$

where we assume that $f_z \doteq -g$ as a simplifying approximation (i.e., all Euler angles are small and the inertial acceleration $a_{zc} << g$).

Let the desired specific force in simulator body axes F_c components be

$$\hat{\underline{f}}_c = \begin{bmatrix} \hat{f}_x \\ \hat{f}_y \\ \hat{f}_z \end{bmatrix} \qquad (A.2)$$

and the following relation holds for an ideal simulator,

$$\hat{\underline{f}}_c = \underline{f}_B \qquad (A.3)$$

To transform $\hat{\underline{f}}_c$ to the inertial frame, we have

$$\hat{\underline{f}}_i = \hat{\underline{L}}_{ic}\, \hat{\underline{f}}_c = \hat{\underline{L}}_{ic}\, \underline{f}_B \qquad (A.4)$$

For small Euler angles, equation (A.4) becomes

$$\hat{\underline{f}}_i = \begin{bmatrix} f_{ix} \\ f_{iy} \\ f_{iz} \end{bmatrix} = \begin{bmatrix} f_x - \hat{\psi} f_y - \hat{\theta} g \\ \hat{\psi} f_x + f_y + \hat{\phi} g \\ -\hat{\theta} f_x + \hat{\phi} f_y - g \end{bmatrix} \qquad (A.5)$$

Remark:

If the inertial acceleration of the simulator in F_i is $[\hat{\ddot{x}}_c, \hat{\ddot{y}}_c\ \hat{\ddot{z}}_c]^T$, then, by the definition of specific force, the actual simulator specific force in F_i is

$$\underline{\hat{f}}_i = \begin{vmatrix} \hat{\ddot{x}}_c \\ \hat{\ddot{y}}_c \\ \hat{\ddot{z}}_c - g \end{vmatrix} \quad \text{(A.6)}$$

From equations (A.5) and (A.6), it is clear that for an ideal simulator

$$\hat{\ddot{x}}_c = f_{ix}, \quad \hat{\ddot{y}}_c = f_{iy}$$

(b) The partial derivatives

From equation (A.4) the exact expression for f_{ix} and f_{iy} are given as follows:

$$f_{ix} = f_x \cos\hat{\theta}_c \cos\hat{\psi}_c + f_y(\sin\hat{\phi}_c \sin\hat{\theta}_c \cos\hat{\psi}_c - \cos\hat{\phi}_c \sin\hat{\psi}_c)$$

$$+ f_z(\cos\hat{\phi}_c \sin\hat{\theta}_c \cos\hat{\psi}_c + \sin\hat{\phi}_c \sin\hat{\psi}_c) \quad \text{(A.7)}$$

$$f_{iy} = f_x \cos\hat{\theta}_c \sin\hat{\psi}_c + f_y(\sin\hat{\phi}_c \sin\hat{\theta}_c \sin\hat{\psi}_c + \cos\hat{\phi}_c \cos\hat{\psi}_c)$$

$$+ f_z(\cos\hat{\phi}_c \sin\hat{\theta}_c \sin\hat{\psi}_c - \sin\hat{\phi}_c \cos\hat{\psi}_c) \quad \text{(A.8)}$$

Therefore from equations (A.7) and (A.8) we have the following partial derivatives:

$$\frac{\partial f_{ix}}{\partial \hat{\theta}_c} = -f_x \sin\hat{\theta}_c \cos\hat{\psi}_c + f_y \sin\hat{\phi}_c \cos\hat{\theta}_c \cos\hat{\psi}_c \quad \text{(A.9)}$$

$$\frac{\partial f_{iy}}{\partial \hat{\phi}_c} = f_y(\cos\hat{\phi}_c \sin\hat{\theta}_c \sin\hat{\psi}_c - \sin\hat{\phi}_c \cos\hat{\psi}_c)$$

$$+ f_z(-\sin\hat{\phi}_c \sin\hat{\theta}_c \sin\hat{\psi}_c - \cos\hat{\phi}_c \cos\hat{\psi}_c) \quad \text{(A.10)}$$

For small Euler angles and $f_z \doteq -g$, equations (A.9) and (A.10) become

$$\frac{\partial f_{ix}}{\partial \hat{\theta}_c} = -f_x + f_y \hat{\phi}_c - g \quad \text{(A.11)}$$

$$\frac{\partial f_{iy}}{\partial \hat{\phi}_c} = -\hat{\phi}_c f_y + g \tag{A.12}$$

2. Longitudinal Filter (variables are in inertial frame F_i)

(a) The cost function

$$J_x = \frac{1}{2} (f_{ix} - \hat{\ddot{x}}_c)^2 + \frac{\omega_x}{2} (\dot{\theta}_c - \hat{\dot{\theta}}_c)^2 + \frac{b_x}{2} \hat{x}_c + \frac{c_x}{2} \hat{\dot{x}}_c$$

(b) The control laws

$$\hat{\ddot{x}}_c = \eta_1 f_{ix} - d_x \hat{\dot{x}}_c - e_x \hat{x}_c$$

$$\hat{\dot{\theta}}_c = \eta_2 f_{ix} + \eta_3 \dot{\theta}_c$$

(c) Steepest descent for the adaptive parameters

$$\dot{\eta}_1 = -k_x \frac{\partial J_x}{\partial \eta_1}$$

$$\dot{\eta}_2 = -k_x \frac{\partial J_y}{\partial \eta_2}$$

$$\dot{\eta}_3 = -k_x \frac{\partial J_z}{\partial \eta_3}$$

(d) State sensitivity equations

$$\frac{\partial \hat{\ddot{x}}_c}{\partial \eta_1} = f_{ix} - dx \frac{\partial \hat{\dot{x}}_c}{\partial \eta_1} - e_x \frac{\partial \hat{x}_c}{\partial \eta_1}$$

$$\frac{\partial \hat{\ddot{x}}_c}{\partial \eta_2} = \eta_1 \frac{\partial f_{ix}}{\partial \eta_2} - dx \frac{\partial \hat{\dot{x}}_c}{\partial \eta_2} - e_x \frac{\partial \hat{x}_c}{\partial \eta_2}$$

$$\frac{\partial \hat{\ddot{x}}_c}{\partial \eta_3} = \eta_1 \frac{\partial f_{ix}}{\partial \eta_3} - dx \frac{\partial \hat{\dot{x}}_c}{\partial \eta_3} - e_x \frac{\partial \hat{x}_c}{\partial \eta_3}$$

with

$$\frac{\partial f_{ix}}{\partial \eta_2} = \frac{\partial f_{ix}}{\partial \hat{\theta}_c} \frac{\partial \hat{\theta}_c}{\partial \eta_2}$$

$$\frac{\partial f_{ix}}{\partial \eta_3} = \frac{\partial f_{ix}}{\partial \hat{\theta}_c} \frac{\partial \hat{\theta}_c}{\partial \eta_3}$$

and

$$\frac{\partial\hat{\dot{\theta}}_c}{\partial\eta_2} = f_{ix} + \eta_2 \frac{\partial f_{ix}}{\partial\hat{\theta}_c} \frac{\partial\hat{\theta}_c}{\partial\eta_2}$$

$$\frac{\partial\hat{\dot{\theta}}_c}{\partial\eta_3} = \eta_2 \frac{\partial f_{ix}}{\partial\eta_3} + \dot{\theta}_c = \eta_2 \frac{\partial f_{ix}}{\partial\hat{\theta}_c} \frac{\partial\hat{\theta}_c}{\partial\eta_3} + \dot{\theta}_c$$

From the cost function J_2, we have

$$\frac{\partial J_x}{\partial\eta_1} = (\hat{\ddot{x}}_c - f_{ix}) \frac{\partial\hat{\ddot{x}}_c}{\partial\eta_1} + b_x\hat{x}_c \frac{\partial\hat{x}_c}{\partial\eta_1} + c_x\hat{\dot{x}}_c \frac{\partial\hat{\dot{x}}_c}{\partial\eta_1}$$

$$\frac{\partial J_x}{\partial\eta_2} = (f_{ix} - \hat{\ddot{x}}_c) \left(\frac{\partial f_{ix}}{\partial\hat{\theta}_c} \frac{\partial\hat{\theta}_c}{\partial\eta_2} - \frac{\partial\hat{\ddot{x}}_c}{\partial\eta_2} \right) - \omega_x(\dot{\theta}_c - \hat{\dot{\theta}}_c) \frac{\partial\hat{\dot{\theta}}_c}{\partial\eta_2}$$
$$+ b_x\hat{x}_c \frac{\partial\hat{x}_c}{\partial\eta_2} + c_x\hat{\dot{x}}_c \frac{\partial\hat{\dot{x}}_c}{\partial\eta_2}$$

$$\frac{\partial J_x}{\partial\eta_3} = (f_{ix} - \hat{\ddot{x}}_c) \left(\frac{\partial f_{ix}}{\partial\hat{\theta}_c} \frac{\partial\hat{\theta}_c}{\partial\eta_3} - \frac{\partial\hat{\ddot{x}}_c}{\partial\eta_3} \right) - \omega_x(\dot{\theta}_c - \hat{\dot{\theta}}_c) \frac{\partial\hat{\dot{\theta}}_c}{\partial\eta_3}$$
$$+ b_x\hat{x}_c \frac{\partial\hat{x}_c}{\partial\eta_3} + c_x\hat{\dot{x}}_c \frac{\partial\hat{\dot{x}}_c}{\partial\eta_3}$$

3. Lateral Filter

(a) Cost function

$$J_y = \frac{1}{2} (f_{iy} - \hat{\ddot{y}}_c)^2 + \frac{W_y}{2} (\dot{\phi}_c - \hat{\dot{\phi}}_c)^2 + \frac{b_y}{2} \hat{y}_c^{\,2} + \frac{c_y}{2} \hat{\dot{y}}_c^{\,2}$$

(b) Control laws

$$\hat{\ddot{y}}_c = \xi_1 f_{iy} - d_y\hat{\dot{y}}_c - e_y\hat{y}_c$$

$$\hat{\dot{\phi}}_c = -\xi_2 f_{iy} + \xi_3\dot{\phi}_c$$

(c) Steepest descent for the adaptive parameters

$$\dot{\xi}_1 = -k_y \frac{\partial J_y}{\partial\xi_1}$$

$$\dot{\xi}_2 = -k_y \frac{\partial J_y}{\partial \xi_2}$$

$$\dot{\xi}_3 = -k_y \frac{\partial J_y}{\partial \xi_3}$$

(d) State sensitivity equations

$$\frac{\partial \hat{\ddot{y}}_c}{\partial \xi_1} = f_{iy} - d_y \frac{\partial \hat{\dot{y}}_c}{\partial \xi_1} - e_y \frac{\partial \hat{y}_c}{\partial \xi_1}$$

$$\frac{\partial \hat{\ddot{y}}_c}{\partial \xi_2} = \xi_1 \frac{\partial f_{iy}}{\partial \xi_2} - d_y \frac{\partial \hat{\dot{y}}_c}{\partial \xi_2} - e_y \frac{\partial \hat{y}_c}{\partial \xi_2}$$

$$\frac{\partial \hat{\ddot{y}}_c}{\partial \xi_3} = \xi_1 \frac{\partial f_{iy}}{\partial \xi_3} - d_y \frac{\partial \hat{\dot{y}}_c}{\partial \xi_3} - e_y \frac{\partial \hat{y}_c}{\partial \xi_3}$$

with

$$\frac{\partial f_{iy}}{\partial \xi_2} = \frac{\partial f_{iy}}{\partial \hat{\phi}_c} \frac{\partial \hat{\phi}_c}{\partial \xi_2}$$

$$\frac{\partial f_{iy}}{\partial \xi_3} = \frac{\partial f_{iy}}{\partial \hat{\phi}_c} \frac{\partial \hat{\phi}_c}{\partial \xi_3}$$

and

$$\frac{\partial \hat{\dot{\phi}}_c}{\partial \xi_2} = -f_{iy} - \xi_2 \frac{\partial f_{iy}}{\partial \hat{\phi}_c} \frac{\partial \hat{\phi}_c}{\partial \xi_2}$$

$$\frac{\partial \hat{\dot{\phi}}_c}{\partial \xi_3} = -\xi_2 \frac{\partial f_{iy}}{\partial \xi_3} + \dot{\phi}_c$$

From the cost function J_y, we have

$$\frac{\partial J_y}{\partial \xi_1} = (\hat{\ddot{y}}_c - f_{iy}) \frac{\partial \hat{\ddot{y}}_c}{\partial \xi_1} + b_y \hat{y}_c \frac{\partial \hat{y}}{\partial \xi_1} + c_y \hat{\dot{y}} \frac{\partial \hat{\dot{y}}_c}{\partial \xi_1}$$

$$\frac{\partial J_y}{\partial \xi_2} = (f_{iy} - \hat{\ddot{y}}_c) \left(\frac{\partial f_{iy}}{\partial \xi_2} - \frac{\partial \hat{\ddot{y}}_c}{\partial \xi_2} \right) - w_y (\dot{\phi}_c - \hat{\dot{\phi}}_c) \frac{\partial \hat{\dot{\phi}}_c}{\partial \xi_2} + b_y \hat{y}_c \frac{\partial \hat{y}_c}{\partial \xi_2}$$

$$+ c_y \hat{\dot{y}}_c \frac{\partial \hat{\dot{y}}_c}{\partial \xi_2}$$

$$\frac{\partial J_y}{\partial \xi_3} = (f_{iy} - \hat{\ddot{y}}_c)\left(\frac{\partial f_{iy}}{\partial \xi_3} - \frac{\partial \hat{\ddot{y}}_c}{\partial \xi_3}\right) - w_y(\dot{\phi}_c - \hat{\dot{\phi}}_c)\,\frac{\partial \hat{\dot{\phi}}_c}{\partial \xi_3} + b_y \hat{y}_c \frac{\partial \hat{y}_c}{\partial \xi_3}$$

$$+ c_y \hat{\dot{y}}_c \frac{\partial \hat{\dot{y}}}{\partial \xi_3}$$

4. Vertical Filter

(a) Cost function

$$J_z = \frac{1}{2}(f_{iz} - \hat{\ddot{z}}_c)^2 + \frac{b_z}{2}\hat{z}_c^{\,2} + \frac{c_z}{2}\hat{\dot{z}}_c^{\,2}$$

(b) The control law

$$\hat{\ddot{z}}_c = \xi f_{iz} - d_z \hat{\dot{z}}_c - e_z \hat{z}_c$$

(c) Steepest descent

$$\dot{\xi} = -k_z \frac{\partial J_z}{\partial \xi}$$

(d) State sensitivity equations

$$\frac{\partial \hat{\ddot{z}}_c}{\partial \xi} = f_{iz} - d_z \frac{\partial \hat{\dot{z}}_c}{\partial \xi} - e_z \frac{\partial \hat{z}_c}{\partial \xi}$$

$$\frac{\partial J_z}{\partial \xi} = (\hat{\ddot{z}}_c - f_{iz})\frac{\partial \hat{\ddot{z}}_c}{\partial \xi} + b_z \hat{z}_c \frac{\partial \hat{z}_c}{\partial \xi} + c_z \hat{\dot{z}}_c \frac{\partial \hat{\dot{z}}_c}{\partial \xi}$$

APPENDIX B

1. Definition of $\underline{x}^{[p]}$

$\underline{x}^{[p]}$ is a $N_n^p = \binom{n+p-1}{n}$ dimensional vector with elements of the form

$$\alpha \prod_{i=1}^{n} x_i^{p_i}$$

where p_i are the non-negative integers such that

$$\sum_{i=1}^{n} p_i = p \qquad \text{and} \qquad \alpha = \sqrt{\binom{p}{p_1}\binom{p-p_1}{p_2}\cdots\binom{p-p_1-\cdots-p_{n-1}}{p_n}}$$

or explicitly,

$$\alpha^2 = \frac{p!}{(p-p_1)!p_1!}\cdot\frac{(p-p_1)!}{(p-p_1-p_2)!p_2!}\cdots\frac{(p-p_1-\cdots-p_{n-2})!}{(p-p_1-p_2-\cdots-p_{n-1})!p_{n-1}!}$$

$$\cdot\frac{(p-p_1-\cdots-p_{n-1})!}{0!\ p_n!} = \frac{p!}{p_1!p_2!p_3!\cdots p_n!}$$

It is shown in reference [15] that

$$\|\underline{x}^{[p]}\| = \|\underline{x}\|^p$$

For illustration, we use the following examples:

Example 1

Let n = 2, p = 2, we have the following possible combinations:

$p_{11} = 1, \quad p_{21} = 1$

$p_{12} = 0, \quad p_{22} = 2$

$p_{13} = 2, \quad p_{23} = 0$

Therefore, from the definition above, it yields

$$\underline{x}^{[2]} = \begin{bmatrix} \alpha_1 \prod_{i=1}^{n} x_i^{p_{i1}} \\ \alpha_2 \prod_{i=1}^{n} x_i^{p_{i2}} \\ \alpha_3 \prod_{i=1}^{n} x_i^{p_{i3}} \end{bmatrix} = \begin{bmatrix} \sqrt{2}\, x_1 x_2 \\ x_2^{\,2} \\ x_1^{\,2} \end{bmatrix}$$

where

$$\alpha_1 = \sqrt{\binom{2}{1}} = \sqrt{\frac{2!}{(2-1)!1!}} = \sqrt{2}$$

$$\alpha_2 = \sqrt{\binom{2}{0}} = 1$$

$$\alpha_3 = \sqrt{\binom{2}{2}} = 1$$

Example 2

Let n = 3, p = 2, since

$$\sum_{j=1}^{3} p_{ij} = p = 2$$

the possible combinations are

$p_{11} = 1,\quad p_{21} = 1,\quad p_{31} = 0$

$p_{12} = 1,\quad p_{22} = 0,\quad p_{32} = 1$

$p_{13} = 0,\quad p_{23} = 1,\quad p_{33} = 1$

$p_{14} = 0,\quad p_{24} = 0,\quad p_{34} = 2$

$p_{15} = 0,\quad p_{25} = 2,\quad p_{35} = 0$

$p_{16} = 2,\quad p_{26} = 0,\quad p_{36} = 0$

Therefore

$$\underline{x}^{[2]} = \begin{bmatrix} \alpha_1 \prod_{i=1}^{3} x_i^{p_{i1}} \\ \alpha_2 \prod_{i=1}^{3} x_i^{p_{i2}} \\ \alpha_3 \prod_{i=1}^{3} x_i^{p_{i3}} \\ \alpha_4 \prod_{i=1}^{3} x_i^{p_{i4}} \\ \alpha_5 \prod_{i=1}^{3} x_i^{p_{i5}} \\ \alpha_6 \prod_{i=1}^{3} x_i^{p_{i6}} \end{bmatrix} = \begin{bmatrix} \sqrt{2}\, x_1 x_2 \\ \sqrt{2}\, x_1 x_3 \\ x_2 x_3 \\ x_3^2 \\ x_2^2 \\ x_1^2 \end{bmatrix}$$

where

$$\alpha_1 = \sqrt{\frac{2!}{(2-1)!1!} \cdot \frac{(2-1)!}{(2-2)!1!}} = \sqrt{2}$$

$$\alpha_2 = \sqrt{\frac{2!}{(2-1)!1!} \quad \frac{(2-1)!}{(2-1-0)!}} = \sqrt{2}$$

$$\alpha_3 = \alpha_4 = \alpha_5 = \alpha_6 = 1$$

2. Pole Assignment - construct the feedback stabilizing matrix $\underline{F}$.

Theorem of pole assignment [18]

For a system

$$\dot{\underline{x}} = \underline{A}\,\underline{x} + \underline{B}\,\underline{u} + \underline{\Gamma}\,\underline{W}$$

The pair $(\underline{A}, \underline{B})$ is controllable if and only if for every symmetric set of n complex numbers, there exists a map $\underline{F}$: $H \rightarrow U$ such that

$$\sigma(\underline{A} + \underline{BF}) = \Lambda$$

where H and U denote the state space and control space respectively.

3. The Relative Negative Overshoot

Definition

The relative negative overshoot is defined as follows:

$$S_i = \frac{b_s}{a_i}$$

where S_i is the relative negative overshoot value, b_s is the value measured at the first peak of negative overshoot with its sign, a_i is the initial value of the response curve.

APPENDIX C THE PARAMETERS CHOSEN FOR THE TESTS

TABLE 1. THE PARAMETERS FOR THE LINEAR WASHOUT FILTER

In each case the following step inputs are used

$\ddot{x} = 1g, 3g, \quad (g = 9.81\ m/sec^2)$

parameters \ case NO.	1	2	3	4
ζ	0.7	0.5	0.3	1.1
ω_n (rad/sec)	1.42	2.0	3.3	0.91

TABLE 2. THE PARAMETERS FOR THE ADAPTIVE WASHOUT FILTER

In all the three filters the following step inputs are used for each case

$f_x = f_y = 1g$ and $f_x = f_y = 3g$

a) The longitudinal filter

$\dot{\theta}_c = 0.13$ rad/sec

parameters \ case NO.	1	2	3	4
ω_x (m^2/rad^2sec^2)	0.0093	0.0063	0.0122	0.065
b_x (per sec^4)	0.01	0.005	0.02	0.07
c_x (per sec)	0.2	0.15	0.4	0.54
d_x (rad/sec)	0.707	0.323	1.2	1.7
e_x (rad/sec^2)	0.25	0.16	0.65	0.85
k_x (sec^3/m^2)	0.323	0.22	0.6	0.8
$\eta_1(0)$	0.02	0.015	0.06	0.08
$\eta_2(0)$	0.0058	0.005	0.007	0.01
$\eta_3(0)$	0.5	0.38	0.7	0.9

b) The lateral filter $\dot{\phi}_c = 0.2$ rad/sec

case NO. / parameters	1	2	3	4
W_y (m^2/rad^2sec^2)	0.0085	0.0063	0.01	0.015
b_y (per sec^4)	0.01	0.007	0.015	0.02
c_y (per sec)	2.0	1.64	2.3	3.4
d_y (rad/sec)	1.273	1.0	1.4	2.3
e_y (rad/sec^2)	0.81	0.52	1.0	1.5
k_y (sec^3/m^2)	0.517	0.37	0.72	1.2
$\xi_1(0)$	0.05	0.035	0.1	0.13
$\xi_2(0)$	0.02	0.02	0.05	0.83
$\xi_3(0)$	1.5	1.25	2.0	2.4

c) The vertical filter $\dot{\psi}_c = 0.3$ rad/sec

case NO. / parameters	1	2	3	4
b_ψ (per sec^4)	0.1	0.07	0.06	0.13
e_ψ (rad/sec^2)	0.3	0.28	0.24	0.34
k_ψ (sec/rad^2)	100.0	95.0	85.0	120.0
b_z (per sec^4)	0.1	0.06	0.07	0.11
c_z (per sec^2)	0.1	0.08	0.09	0.12
d_z (rad/sec)	1.2727	1.0	1.62	1.4
e_z (rad/sec^2)	0.81	0.62	0.7	0.92
k_z (sec^3/m^2)	0.517	0.27	0.83	0.95
ξ	0.05	0.025	0.09	0.12

TABLE 3. THE NONLINEAR OPTIMAL WASHOUT FILTER

a) The longitudinal filter

Step input: $a_{cx} = 1g, 3g$ and $\dot{\theta}_c = 0.2$ rad/sec

parameters \ case NO.	1	2	3	4
R_{11} (m/sec^2)	0.1	0.45	0.45	0.45
R_{22} (rad/sec)	0.2	0.1	0.1	0.1
x_L (m)	0.1	0.91	0.91	0.91
v_L (m/sec)	0.2	0.61	0.61	0.61
θ_L (rad)	0.1	0.44	0.44	0.44
a_{11}	0.1	0.05	0.15	0.1
a_{22}	0.1	0.05	0.15	0.1
a_{33}	0.1	0.05	0.15	0.1
a_o	0.2	1.2	1.2	0.8

b) The lateral filter

Step input: $a_{cy} = 1g, 3g$ and $\dot{\phi}_c = 0.2$ rad/sec

parameters \ case NO.	1	2	3	4
R_{11} (m/sec^2)	0.2	0.2	0.2	0.2
R_{22} (rad/sec)	0.05	0.05	0.05	0.05
y_L (m)	0.91	0.91	0.91	0.91
v_L (m/sec)	4.8	4.8	4.8	4.8
ϕ_L (rad)	0.44	0.44	0.44	0.44
b_{11}	0.1	0.05	0.15	0.1
b_{22}	0.1	0.05	0.15	0.1
b_{33}	0.1	0.05	0.15	0.1
b_o	1.2	1.2	1.2	0.8

c) The vertical filter

Step input: $a_{cz} = 1g, 3g$

parameters \ case NO.	1	2	3	4
R_{11} (m/sec^2)	1.57	1.962	1.57	1.57
c_{11}	0.1	0.05	0.05	0.1
c_{22}	0.1	0.05	0.05	0.1
c_o	0.9	0.6	0.9	0.9
v_L (m/sec)	0.61	0.61	0.61	0.72
z_L (m)	0.991	0.991	0.991	0.72

APPENDIX D

A LIST OF PROGRAMS USED

```
                C    THIS PROGRAM SIMULATES THE NONLINEAR OPTIMAL LONGITUDINAL
                C       WASHOUT FILTER. THIS PROGRAM WORKED SCCESSFULLY THE FIRST
                C       TIME ON THE MTM ON OCT.2Ø,1982,PM 2:1Ø.
                $HOLL
      ØØØØØØI           IMPLICIT REAL*8(A-H,O-Z)
      ØØØØØ6I           DIMENSION A(3,3),B(3,2) ,Q(3,3),R(2,2),P(3,3),F(2,3),W(3,3),
                       *WA(3,3),WP(3,3),WQ(3,3),A1(15),X(3),XD(3),WK2(45Ø),XO1(3),
                       /TT(6Ø2),XO(3),XDD1(6Ø2),XDD2(6Ø2),WORKA(6)
      ØØØØØ6I           COMMON/DOT1/F,PT,QT,RT,G
      ØØØØØ6I           COMMON/DOT2/A1,DLX,DLV,DLT,FXA,DTHTA,T1
      ØØØØØ6I           COMMON/DOT3/GW2,QW2
      ØØØØØ6I           EXTERNAL SYS, AUX2
      ØØØØØ6I           CALL PLOTS(1,Ø,Ø)
                C
                C    READ IN THE SYSTEM MATRICES AND THE STABILIZING FEEDBACK MATRIX.
                C
      ØØ499CI           OPEN(UNIT=Ø5,FILE='LONDAT.DTA')
      ØØ49E8I           OPEN(UNIT=Ø6,FILE='CON:')
      ØØ4A34I           WRITE(6,444)
      ØØ4A5ØI           DO 1 I = 1,3
      ØØ4A58I         1 READ(5,*) (A(I,J),J = 1,3)
      ØØ4AECI           DO 4 I =  1,3
      ØØ4AF4I         4 READ(5,*) (B(I,K),K = 1,2)
      ØØ4B84I           DO 3 II = 1,2
      ØØ4B8CI         3 READ(5,*) (F(II,JK),JK= 1,3)
      ØØ4C2ØI           WRITE(6,1ØØ)
      ØØ4C3CI           WRITE(6,1Ø1)
      ØØ4C58I           WRITE(6,1Ø2) ((A(I,J),J=1,3),I=1,3)
      ØØ4CF8I           WRITE(6,11Ø)
      ØØ4D14I           WRITE(6,1Ø3)
      ØØ4D3ØI           WRITE(6,1Ø4) ((B(I,J),J=1,2),I=1,3)
      ØØ4DCCI           WRITE(6,11Ø)
      ØØ4DE8I           WRITE(6,111)
      ØØ4EØ4I           WRITE(6,1Ø2) ((F(I,J),J=1,3),I=1,2)
                C***********************************************************************
      ØØ4EA4I           DO 15 JN = 1,4
                C***********************************************************************
      ØØ4EACI           IP = Ø
      ØØ4EB4I           IS =2
      ØØ4EBCI           T = 1.D-4
      ØØ4EC8I           NN = 3
      ØØ4EDØI           MM= 2
      ØØ4ED8I           N2 = 15
      ØØ4EEØI           G = 9.8Ø62
                C       EPS = 1.ØE-6
      ØØ4EECI           NSIG = 5
      ØØ4EF4I           ITMAX = 12Ø
      ØØ4EFEI           M = 6ØØ
      ØØ4FØ8I           WRITE(6,123) M
      ØØ4F28I           H = .Ø2
      ØØ4F34I           AX=6.Ø
      ØØ4F4ØI           AY=4.Ø
                C***********************************************************************
      ØØ4F4CI           READ(5,*) DETL,DETN,XL,VL,THTL,QW,GW
      ØØ4F8ØI           WRITE(6,1ØØ)
      ØØ4F9CI           WRITE(6,1Ø9)
      ØØ4FB8I           WRITE(6,1Ø8) DETL,DETN,XL,VL,THTL,QW,GW
```

```
004FF0I          PT = DETN/(2.*XL**4)
005012I          QT = DETN/(2.*VL**4)
005034I          RT = DETN/(2.*THTL**4)
005056I          DLX = DETL/(XL*XL)
005070I          DLV = DETL/(VL*VL)
00508AI          DLT = DETL/(THTL*THTL)
0050A4I          GW2 = GW*GW
0050B6I          QW2 = QW*QW
0050C8I          Q(1,1) = DLX
0050D4I          Q(1,2) = .0
0050E0I          Q(1,3) = .0
0050ECI          Q(2,1) = .0
0050F8I          Q(2,2) = DLV
005104I          Q(2,3) = .0
005110I          Q(3,1) = .0
00511CI          Q(3,2) = .0
005128I          Q(3,3) = DLT
005134I          R(1,1) = 1./GW2
005146I          R(1,2) = .0
005152I          R(2,1) = .0
00515EI          R(2,2) = 1./QW2
005170I          DO 2 II = 1,15
005178I        2 A1(II) = .0
0051A2I          WRITE(6,110)
0051BCI          WRITE(6,105)
0051D8I          WRITE(6,102) ((Q(I,J),J = 1,3),I=1,3)
005274I          WRITE(6,110)
005290I          WRITE(6,106)
0052ACI          WRITE(6,104) ((R(I,J),J=1,2),I=1,2)
005348I          WRITE(6,110)
005364I          CALL RICATI(A,B,Q,R,P,F,3,2,NN,MM,IP,IS,T,W,WA,WP,WQ)
0053CCI          WRITE(6,107)
0053E8I          WRITE(6,102) ((F(I,J),J = 1,3),I = 1,2)
005488I          WRITE(6,110)
0054A4I          CALL ZSPOW(AUX2, NSIG,N2,ITMAX,PAR,A1,FNORM1,WK2,IER2)
0054DCI          WRITE(6,112)
0054F8I          WRITE(6,113)
005514I          WRITE(6,120) (A1(JI),JI = 1,7)
005586I          WRITE(6,121)
0055A0I          WRITE(6,122) (A1(JI), JI = 8,15)
005612I          WRITE(6,100)
00562CI          XA = 1.0
005638I          YA = 5.5
005644I          IF(JN.GT.2.AND.MOD(JN,2).NE.0) GO TO 21
005688I          IF(MOD(JN,2).EQ.0) GO TO 13
0056BCI          GO TO 20
0056C2I       21 CONTINUE
0056C2I          XA = 9.0
0056CEI          YA = 5.0
0056DAI          GO TO 20
0056E0I       13 CONTINUE
0056E0I          XA = 0.
0056ECI          YA = -5.
0056F8I       20 CONTINUE
0056F8I          CALL PLOT(XA,YA,-3)
005720I          CALL REGION
            C
```

```
          C   PUT THE EXPLANATION IN THE GRAPH
          C
ØØ5728I         CALL SYMBOL(1.25,3.68,Ø.12,'NON-LINEAR OPTIMAL WASHOUT FILTER',Ø.,
               /+33)
ØØ57BCI         CALL SYMBOL(2.Ø,3.42,Ø.12,'(LONGITUDINAL)',Ø.,+14)
ØØ5838I         CALL SYMBOL(1.56,3.25,Ø.Ø7,2,Ø.,-1)
ØØ58A4I         CALL SYMBOL(1.62,3.19,Ø.1,45,Ø.,-1)
ØØ591ØI         CALL NUMBER(999.,999.,Ø.1,3.Ø,Ø.,+1)
ØØ598ØI         CALL SYMBOL(999.,999.,Ø.1,'G (INPUT)',Ø.,+9)
ØØ59FCI         CALL SYMBOL(1.56,3.Ø,Ø.Ø7,5,Ø.,-1)
ØØ5A68I         CALL SYMBOL(1.62,2.94,Ø.1,45,Ø.,-1)
ØØ5AD4I         CALL NUMBER(999.,999.,Ø.1,1.Ø,Ø.,+1)
ØØ5B44I         CALL SYMBOL(999.,999.,Ø.1,'G (INPUT)',Ø.,+9)
ØØ5BCØI         CALL SYMBOL(4.Ø,3.25,Ø.1,'PARAMETERS',Ø.,+1Ø)
ØØ5C3CI         CALL SYMBOL(4.2,3.Ø8,Ø.1,'DETL= ',Ø.,+6)
ØØ5CBØI         CALL NUMBER(999.,999.,Ø.1,DETL,Ø.,+3)
ØØ5D14I         CALL SYMBOL(4.2,2.91,Ø.1,'DETN= ',Ø.,+6)
ØØ5D88I         CALL NUMBER(999.,999.,Ø.1,DETN,Ø.,+3)
ØØ5DECI         CALL SYMBOL(4.2,2.74,Ø.1,'XL= ',Ø.,+4)
ØØ5E6ØI         CALL NUMBER(999.,999.,Ø.1,XL,Ø.,+3)
ØØ5EC4I         CALL SYMBOL(4.2,2.57,Ø.1,'VL= ',Ø.,+4)
ØØ5F38I         CALL NUMBER(999.,999.,Ø.1,VL,Ø.,+3)
ØØ5F9CI         CALL SYMBOL(4.2,2.4,Ø.1,'THTL= ',Ø.,+6)
ØØ6Ø1ØI         CALL NUMBER(999.,999.,Ø.1,THTL,Ø.,+3)
ØØ6Ø74I         CALL SYMBOL(4.2,2.23,Ø.1,'QW= ',Ø.,+4)
ØØ6ØE8I         CALL NUMBER(999.,999.,Ø.1,QW,Ø.,+3)
ØØ614CI         CALL SYMBOL(4.2,2.Ø6,Ø.1,'GW= ',Ø.,+4)
ØØ61CØI         CALL NUMBER(999.,999.,Ø.1,GW,Ø.,+3)
          C
ØØ6224I         DO 1Ø J = 1,2
ØØ622CI         X(1) = .Ø
ØØ6238I         X(2) = .Ø
ØØ6244I         X(3) = .Ø
ØØ625ØI         T1 = .Ø
ØØ625CI         READ(5,*) FXA,DTHTA
ØØ627CI         FXA = 9.81*FXA                                               M/SS
ØØ628EI         WRITE(6,116) FXA,DTHTA
ØØ62BØI         WRITE(6,11Ø)
ØØ62CCI         WRITE(6,117)
ØØ62E8I         DO 11 JP = 1,M
ØØ62FCI         CALL RKM(3,H,T1,X,XO,XO1,XD,SYS)
ØØ6338I         TT(JP) = T1
ØØ634CI         IF(J.EQ.2) GO TO 22
ØØ6362I         XDD1(JP) = XD(2)
ØØ6374I         GO TO 12
ØØ637AI      22 XDD2(JP) = XD(2)
ØØ638CI      12 IF(MOD(JP,2Ø).EQ.Ø) WRITE(6,118) T1,XD(1),XD(2),XD(3),X(1),X(3)
ØØ63ECI      11 CONTINUE
ØØ64Ø4I         WRITE(6,1ØØ)
ØØ642ØI      1Ø CONTINUE
ØØ6436I         CALL SCALE(TT,6.,M,2)
ØØ646CI         CALL SCALE(XDD1,4.,M,2)
ØØ64A4I         CALL SCALE(XDD2,4.,M,2)
ØØ64DCI         WRITE(6,777) XDD1(M+1),XDD2(M+1),XDD1(M+2),XDD2(M+2)
ØØ656ØI         WRITE(6,1ØØ)
          C
          C   SELECT THE COMMON SCALE FACTORS
```

```
                C
 ØØ657CI              WORKA(1) = XDD1(M+1)
 ØØ659ØI              WORKA(2) = XDD2(M+1)
 ØØ65A2I              WORKA(3) = XDD1(M+1)+XDD1(M+2)*AY
 ØØ65CCI              WORKA(4) = XDD2(M+1)+XDD2(M+2)*AY
 ØØ65F2I              CALL SCALE(WORKA,4.,4,2)
 ØØ663ØI              XDD1(M+1) = WORKA(5)
 ØØ6644I              XDD2(M+1) = WORKA(5)
 ØØ6656I              XDD1(M+2) = WORKA(6)
 ØØ666AI              XDD2(M+2) = WORKA(6)
                C
                C    PRINT OUT THE SCALE FACTORS
                C
 ØØ667CI              WRITE(6,555) M,XDD1(M+1) ,XDD2(M+1),XDD1(M+2),XDD2(M+2)
 ØØ67Ø4I              WRITE(6,666) JP,WORKA(5),WORKA(6)
 ØØ672CI              WRITE(6,1ØØ)
                C
                C    CALLES FOR PLOTTING
                C
 ØØ6748I              CALL AXIS(Ø.,Ø.,'TIME(SEC.)',-1Ø,AX,Ø.,TT(M+1),TT(M+2))
 ØØ67ECI              CALL AXIS(Ø.,Ø.,'X ACCEL. (M/SEC.**2)',+2Ø,AY,9Ø.,
                     /XDD1(M+1),XDD1(M+2))
 ØØ6898I              CALL LINE(TT,XDD1,M,2,8Ø,2)
 ØØ68DCI              CALL LINE(TT,XDD2,M,2,8Ø,5)
 ØØ692ØI           15 CONTINUE
 ØØ6936I              CALL PLOT(Ø.,Ø.,+999)
 ØØ6978I         1ØØ FORMAT(1X,6Ø(1H*))
 ØØ6988I         1Ø1 FORMAT(1X,6(1H*),2X,'MATRIX A',2X,6(1H*))
 ØØ69ACI         1Ø2 FORMAT(3(1X,F8.4,2X))
 ØØ69BEI         1Ø3 FORMAT(1X,6(1H*),2X,'MATRIX B',2X,6(1H*))
 ØØ69E2I         1Ø4 FORMAT(2(1X,F8.4,2X))
 ØØ69F4I         1Ø5 FORMAT(1X,6(1H*),2X,'MATRIX Q',2X,6(1H*))
 ØØ6A18I         1Ø6 FORMAT(1X,6(1H*),2X,'MATRIX R',2X,6(1H*))
 ØØ6A3CI         1Ø7 FORMAT(1X,6(1H*),2X,'MATRIX F',2X,6(1H*))
 ØØ6A6ØI         1Ø8 FORMAT(1X,7(F7.4,2X))
 ØØ6A72I         1Ø9 FORMAT(3X,'DETL',5X,'DETN',5X,'XL',8X,'VL',6X,'THTL',5X,'QW',7X,'G
                     /W')
 ØØ6AAAI         11Ø FORMAT(1X,28(1H*))
 ØØ6ABAI         111 FORMAT(1X,6(1H*),2X,'MATRIX FØ',2X,6(1H*))
 ØØ6AEØI         112 FORMAT(1X,5(1H*),2X,'THE COEFFICIENTS',2X,5(1H*))
 ØØ6BØCI         113 FORMAT(6X,'A1',12X,'A2',12X,'A3',12X,'A4',12X,'A5',12X,'A6',12X,'A
                     /7')
 ØØ6B3EI         116 FORMAT(6X,'FXA = ',F7.4,2X,'DTHTA = ',F7.4)
 ØØ6B62I         117 FORMAT(4X,'T',1ØX,'XD1',11X,'XD2',11X,'XD3',1ØX,'X1',12X,'X3')
 ØØ6B9ØI         118 FORMAT(1X, F6.3,2X,5(E12.5,2X))
 ØØ6BA8I         12Ø FORMAT(1X,7(E12.5 ,2X))
 ØØ6BBCI         121 FORMAT(6X,'A8',12X,'A9',12X,'A1Ø',11X,'A11',11X,'A12',11X,
                     /'A13',11X,'A14',11X,'A15')
 ØØ6BFAI         122 FORMAT(1X,8(E12.5,2X))
 ØØ6CØEI         123 FORMAT(1X,'THE ITERATIONS ARE',1X,I4)
 ØØ6C3ØI         444 FORMAT(8(1H*),2X,'THE RESULTS OF THE LONGITUDINAL WASHOUT
                     /FILTER',2X,8(1H*))
 ØØ6C8ØI         555 FORMAT(1X,'M=',I4,2X,'XDD1WL=',F7.4,2X,'XDD2WL=',F7.4,2X,
                     /'XDD1WU=',F7.4,2X,'XDD2WU=',F7.4)
 ØØ6CC8I         666 FORMAT(1X,'JP=',I4,2X,'WORKA5=',F7.4,2X,'WORKA6=',F7.4)
 ØØ6CF6I         777 FORMAT(1X,'XDD1L=',F7.4,2X,'XDD2L=',F7.4,2X,'XDD1U=',F7.4,
                     /2X,'XDD2U=',F7.4)
```

```
FORTRAN-VIID RØ5-ØØ.ØØ
FORTRAN VIID: LICENSED RESTRICTED RIGHTS AS STATED IN LICENSE CL-ØØ13         ***, SEE DOCUMENTATION PACKAGE, Ø4-1Ø1M99.     17/Ø2/83  16:58:18  PAGE   5/    5
229  ØØ6D32I          STOP
23Ø  ØØ6D3AI          END

NO  ERRORS:F7D  RØ5-ØØ.ØØ  MAINPROG     .MAIN      17/Ø2/83  16:58:42  TABLE SPACE:    9 KB
STATEMENT BUFFER:   2Ø LINES/1321 BYTES     STACK SPACE: 157 WORDS
SINGLE PRECISION FLOATING PT SUPPORT REQUIRED FOR EXECUTION
DOUBLE PRECISION FLOATING PT SUPPORT REQUIRED FOR EXECUTION
```

D - 6

```
FORTRAN-VIID RØ5-ØØ.ØØ
FORTRAN VIID: LICENSED RESTRICTED RIGHTS AS STATED IN LICENSE CL-ØØ13         ***, SEE DOCUMENTATION PACKAGE, Ø4-1Ø1M99.     17/Ø2/83  16:58:18  PAGE   1/    6
  1  ØØØØØØI          SUBROUTINE REGION
  2  ØØØØØ4I          XR = 6.Ø
  3  ØØØØ1ØI          YR = 4.Ø
  4  ØØØØ1CI          CALL NEWPEN(3)
  5  ØØØØ3CI          CALL PLOT(XR,Ø.,2)
  6  ØØØØ7ØI          CALL PLOT(XR,YR,2)
  7  ØØØØ98I          CALL PLOT(Ø.,YR,2)
  8  ØØØØCCI          CALL PLOT(Ø.,Ø.,2)
  9  ØØØ1ØCI          CALL NEWPEN(1)
 1Ø  ØØØ12CI          RETURN
 11  ØØØ132I          END

NO  ERRORS:F7D  RØ5-ØØ.ØØ  SUBROUTINE  REGION   17/Ø2/83  16:58:43  TABLE SPACE:    1 KB
STATEMENT BUFFER:   2Ø LINES/1321 BYTES     STACK SPACE:  52 WORDS
SINGLE PRECISION FLOATING PT SUPPORT REQUIRED FOR EXECUTION
```

```
FORTRAN-VIID RØ5-ØØ.ØØ                                                                    17/Ø2/83  16:58:18  PAGE   1/        7
FORTRAN VIID: LICENSED RESTRICTED RIGHTS AS STATED IN LICENSE CL-ØØ13        ***, SEE DOCUMENTATION PACKAGE, Ø4-1Ø1M99.
 ØØØØØØI          SUBROUTINE AUX2(A1,K,PAR)
 ØØØØØ4I          IMPLICIT REAL*8(A-H,O-Z)
 ØØØØØ4I          DIMENSION A1(15),PAR(1),F(2,3),WK2(45Ø),FT(15)
 ØØØØØ4I          COMMON/DOT1/F,PT,QT,RT,G
          C       GO TO (1,2,3,4,5,6,7,8,9,1Ø,11,12,13,14,15),K
 ØØØØØ4I          FT(1)= F(1,1)*A1(11)+F(2,1)*A1(2)-PT
          C       RETURN
 ØØØEC8I          FT(2) = A1(13)-4.*F(1,2)*A1(15)-F(2,2)*A1(14)+RT
          C       RETURN
 ØØØFØ6I          FT(3) = (G+F(1,3))*A1(6)+4.*F(2,3)*A1(7)-QT
          C       RETURN
 ØØØF3EI          FT(4)= 4.*A1(1)-2.*F(1,1)*A1(12)-F(1,2)*A1(11)-F(2,1)*A1(3)-F(2,2)
                 /*A1(2)
          C       RETURN
 ØØØFAØI          FT(5)= 3.*A1(2)-2.*F(1,1)*A1(1Ø)-(F(1,2)+F(2,3))*A1(3)-2.*(G+F(1,3
                 /))*A1(12)-2.*F(2,1)*A1(4)-2.*F(2,2)*A1(9)
          C       RETURN
 ØØ1Ø32I          FT(6)= 2.*A1(3)-3.*F(1,1)*A1(14)-2.*F(1,2)*A1(1Ø)-3.*(G+F(1,3))*A1
                 /(13)-2.*F(2,1)*A1(5)-2.*F(2,2)*A1(4)-F(2,3)*A1(1Ø)
          C       RETURN
 ØØ1ØD6I          FT(7)= A1(4)-(2.*F(1,2)+2.*F(2,3))*A1(5)-3.*(G+F(1,3))*A1(14)-3.*F
                 /(2,2)*A1(6)
          C       RETURN
 ØØ114ØI          FT(8)= A1(8)-(F(1,2)+3.*F(2,3))*A1(6)-2.*(G+F(1,3))*A1(5)-4.*F(2,2
                 /)*A1(7)
          C       RETURN
 ØØ11A2I          FT(9)= 2.*A1(9)-2.*F(1,1)*A1(5)-(F(1,2)+2.*F(2,3))*A1(4)-2.*(G+F(1
                 /,3))*A1(1Ø)-3.*F(2,1)*A1(6)-3.*F(2,2)*A1(8)
          C       RETURN
 ØØ123AI          FT(1Ø)=A1(1Ø)-(3.*F(1,2)+F(2,3))*A1(14)-4.*(G+F(1,3))*A1(15)-2.*F(
                 /2,2)*A1(5)
          C       RETURN
 ØØ129CI          FT(11)=3.*A1(11)-3.*F(1,1)*A1(13)-2.*F(1,2)*A1(12)-F(2,1)*A1(1Ø)-F
                 /(2,2)*A1(3)
          C       RETURN
 ØØ13Ø4I          FT(12)=2.*A1(12)-4.*F(1,1)*A1(15)-3.*F(1,2)*A1(13)-F(2,1)*A1(14)-F
                 /(2,2)*A1(1Ø)
          C       RETURN
 ØØ136CI          FT(13)=F(1,1)*A1(3)+(G+F(1,3))*A1(11)+2.*F(2,1)*A1(9)+F(2,3)*A1(2)
          C       RETURN
 ØØ13C2I          FT(14)=F(1,1)*A1(4)+(G+F(1,3))*A1(3)+3.*F(2,1)*A1(8)+2.*F(2,3)*A1(
                 /9)
          C       RETURN
 ØØ141EI          FT(15)=F(1,1)*A1(6)+(G+F(1,3))*A1(4)+4.*F(2,1)*A1(7)+3.*F(2,3)*A1(
                 /8)
 ØØ147AI          RETURN
 ØØ148ØI          END
WARNING #   9 ***************************************************************************
          >>> VARIABLE NOT INITIALIZED IN PROGRAM                                          <<<

          WK2


NO  ERRORS:F7D  RØ5-ØØ.ØØ  SUBROUTINE  AUX2       17/Ø2/83  16:58:51  TABLE SPACE:   2 KB
STATEMENT BUFFER:   2Ø LINES/1321 BYTES    STACK SPACE: 154 WORDS
DOUBLE PRECISION FLOATING PT SUPPORT REQUIRED FOR EXECUTION
```

```
FORTRAN-VIID RØ5-ØØ.ØØ                                                              17/Ø2/83  16:58:18  PAGE   1/      8
FORTRAN VIID: LICENSED RESTRICTED RIGHTS AS STATED IN LICENSE CL-ØØ13        ***, SEE DOCUMENTATION PACKAGE, Ø4-1Ø1M99.
ØØØØØØI           SUBROUTINE RKM(N,H,X,Y,YO,YO1,YP,SYS)
ØØØØØ4I           IMPLICIT REAL*8(A-H,O-Z)
ØØØØØ4I           DIMENSION Y(N),YO(N),YO1(N),YP(N),GI(5),A1(15),F(2,3)
ØØØØØAI           COMMON/DOT1/F,PT,QT,RT,G
ØØØØØAI           COMMON/DOT2/A1,DLX,DLV,DLT,FXA,DTHTA,T1
ØØØØØAI           COMMON/DOT3/GW2,QW2
ØØØØØAI           GI(1) = .5
ØØØØ66I           GI(2) = .5
ØØØØ7ØI           GI(3) = 1.
ØØØØ7AI           GI(4) = 1.
ØØØØ84I           GI(5) = .5
ØØØØ8EI           XO = X
ØØØØ9AI           DO 1 I = 1,N
ØØØØAEI           YO(I) = Y(I)
ØØØØD6I         1 YO1(I) = Y(I)
ØØØ116I           DO 2 J = 1,4
ØØØ11EI           CALL SYS(N,X,YO1,YP)
ØØØ168I           X = XO+GI(J)*H
ØØØ186I           DO 2 I = 1,N
ØØØ19AI           YO1(I) = YO(I)+GI(J)*H*YP(I)
ØØØ1E8I         2 Y(I) = Y(I)+GI(J+1)*H*YP(I)/3.Ø
ØØØ26CI           RETURN
ØØØ272I           END

NO  ERRORS:F7D  RØ5-ØØ.ØØ  SUBROUTINE  RKM       17/Ø2/83  16:58:52  TABLE SPACE:    2 KB
STATEMENT BUFFER:    2Ø LINES/1321 BYTES    STACK SPACE: 158 WORDS
DOUBLE PRECISION FLOATING PT SUPPORT REQUIRED FOR EXECUTION
```

```
FORTRAN-VIID RØ5-ØØ.ØØ                                                              17/Ø2/83  16:58:18  PAGE   1/      9
FORTRAN VIID: LICENSED RESTRICTED RIGHTS AS STATED IN LICENSE CL-ØØ13        ***, SEE DOCUMENTATION PACKAGE, Ø4-1Ø1M99.
ØØØØØØI           SUBROUTINE SYS(N,T,X,XD)
ØØØØØ4I           IMPLICIT REAL*8(A-H,O-Z)
ØØØØØ4I           DIMENSION X(3),XD(3),F(2,3),A1(15)
ØØØØØ4I           COMMON/DOT1/F,PT,QT,RT,G
ØØØØØ4I           COMMON/DOT2/A1,DLX,DLV,DLT,FXA,DTHTA,T1
ØØØØØ4I           COMMON/DOT3/GW2,QW2
ØØØØØ4I           DXLV = A1(3)*X(1)*X(1)*X(3)+A1(4)*X(1)*X(3)*X(3)+2.*A1(5)*X(3)*X(3
                 /)*X(2)+A1(6)*X(3)*X(3)*X(3)+2.*A1(1Ø)*X(1)*X(2)*X(3)+A1(11)*X(1)*X
                 /(1)*X(1)+2.*A1(12)*X(1)*X(1)*X(2)+3.*A1(13)*X(1)*X(2)*X(2)+3.*A1(1
                 /4)*X(3)*X(2)*X(2)+4.*A1(15)*X(2)*X(2)*X(2)
ØØØ1B8I           DXLT = A1(2)*X(1)*X(1)*X(1)+A1(3)*X(1)*X(1)*X(2)+2.*A1(4)*X(1)*X(2
                 /)*X(3)+2.*A1(5)*X(2)*X(2)*X(3)+3.*A1(6)*X(2)*X(3)*X(3)+4.*A1(7)*X(
                 /3)*X(3)*X(3)+3.*A1(8)*X(1)*X(3)*X(3)+2.*A1(9)*X(1)*X(1)*X(3)+A1(1Ø
                 /)*X(1)*X(2)*X(2)+A1(14)*X(2)*X(2)*X(2)
ØØØ35CI           XD(1) = X(2)
ØØØ37ØI           XD(2) = -F(1,1)*X(1)-F(1,2)*X(2)-(F(1,3)+G)*X(3)-DXLV*GW2+FXA
                 /+G*DTHTA*T1
ØØØ3E4I           XD(3) = -F(2,1)*X(1)-F(2,2)*X(2)-F(2,3)*X(3)-DXLT*QW2+DTHTA
ØØØ43EI           RETURN
ØØØ444I           END

NO  ERRORS:F7D  RØ5-ØØ.ØØ  SUBROUTINE  SYS       17/Ø2/83  16:58:54  TABLE SPACE:    2 KB
STATEMENT BUFFER:    2Ø LINES/1321 BYTES    STACK SPACE: 146 WORDS
DOUBLE PRECISION FLOATING PT SUPPORT REQUIRED FOR EXECUTION
```

```
ØØØØØØI            SUBROUTINE RICATI(A,B,Q,R,P,F,N,M,NN,MM,IP,IS,T,W,WA,WP,WQ)              ØØØ416ØØ
             C     SUBROUTINE RICATI(A,B,Q,R,P,F,N;M,NN,MM,IP,IS,T,W,WA,WP,WQ)              ØØØ4146Ø
             C                                                                              ØØØ4147Ø
             C   SOLVES     Ø = A'P + PA + Q - PB(R.INV.)B'P                                ØØØ4148Ø
             C                                                                              ØØØ4149Ø
             C   (A-B*F) HAS THE DESIRED SPECTRUM UPON RETURN                               ØØØ415ØØ
             C   CALLS SUBROUTINES MAT AND LYAPUN                                           ØØØ4151Ø
             C   IP IS PRINT CONTROL   = Ø,1,2           2 GIVES FULL PRINTOUT              ØØØ4152Ø
             C   IS :  = 1 (A) IS ASSUMED STABLE , = 2 (A-BF) IS ASSUMED STABLE             ØØØ4153Ø
             C   IS IS UNCHANGED FOR SUCCESS , SET = Ø FOR NON-CONVERGENCE                  ØØØ4154Ø
             C         SET NEGATIVE FOR NON-CONVERGENCE IN S/R LYAPUN (INSTABILITY)         ØØØ4155Ø
             C      WORKING STORAGE :  (W) , (WA) , (WP) , (WQ)                             ØØØ4156Ø
             C       DIMENSIN OF F CHANGED FROM  F(NN,N) TO F(M,N). DATE:3Ø/Ø8/1973         ØØØ4157Ø
             C   STABILITY IS IN THE SENSE THAT THE SYSTEM IS ASYMPTOTICALLY STABLE.        ØØØ41571
             C                                                                              ØØØ4158Ø
             C                                                                              ØØØ4159Ø
ØØØØØ4I            IMPLICIT REAL*8(A-H,O-Z)                                                 ØØØ4161Ø
ØØØØØ4I            DIMENSION A(NN,N),B(NN,M),P(NN,N),Q(NN,N),R(MM,M),F(M,N)                 ØØØ4163Ø
ØØØØØAI            DIMENSION W(NN,N),WA(NN,N),WP(NN,N),WQ(NN,N)                             ØØØ4164Ø
ØØØØØAI            ABS(X)=DABS(X)                                                           ØØØ4162Ø
ØØØØE4I            LIT=3Ø                                                                   ØØØ4165Ø
ØØØØEEI            ERR=1.E-8                                                                ØØØ4166Ø
ØØØØFAI            ZERO=1.E-11                                                              ØØØ4167Ø
ØØØ1Ø6I            JS=Ø                                                                     ØØØ4168Ø
ØØØ1ØEI            TRO=1.E+5Ø                                                               ØØØ4169Ø
ØØØ11AI            IF(IP.GE.1) WRITE(6,9Ø1)                                                 ØØØ417ØØ
ØØØ144I            IF(IP.LE.1) GO TO 19                                                     ØØØ4171Ø
ØØØ15AI            WRITE(6,9Ø2)                                                             ØØØ4172Ø
ØØØ174I            DO 13 I=1,N                                                              ØØØ4173Ø
ØØØ188I         13 WRITE(6,91Ø)(A(I,J),J=1,N)                                               ØØØ4174Ø
ØØØ238I            WRITE(6,911)                                                             ØØØ4175Ø
ØØØ254I            DO  14 I=1,N                                                             ØØØ4176Ø
ØØØ268I         14 WRITE(6,91Ø)(B(I,J),J=1,M)                                               ØØØ4177Ø
ØØØ318I            WRITE(6,911)                                                             ØØØ4178Ø
ØØØ334I            DO 15  I=1,N                                                             ØØØ4179Ø
ØØØ348I         15 WRITE(6,91Ø)(Q(I,J),J=1,N)                                               ØØØ418ØØ
ØØØ3F8I            WRITE(6,911)                                                             ØØØ4181Ø
ØØØ414I            DO 16 I=1,M                                                              ØØØ4182Ø
ØØØ428I         16 WRITE(6,91Ø)(R(I,J),J=1,M)                                               ØØØ4183Ø
ØØØ4D8I            WRITE(6,911)                                                             ØØØ4184Ø
             C                 INVERT (R)                                                   ØØØ4185Ø
ØØØ4F4I         19 DO 2Ø I=1,M                                                              ØØØ4186Ø
ØØØ5Ø8I            DO 2Ø J=1,M                                                              ØØØ4187Ø
ØØØ51CI         2Ø P(I,J)=R(I,J)                                                            ØØØ4188Ø
ØØØ594I            CALL MAT(P,P,M,M,NN,ZERO,DET,K)                                          ØØØ4189Ø
ØØØ5ECI            IF(IS.LT.2) GO TO 99                                                     ØØØ419ØØ
             C       GIVEN (A - B*F) STABLE , FIND PØ BY SOLVING                            ØØØ4191Ø
             C           (A - BF)'P + P(A - BF) + (F'RF + Q) = Ø                            ØØØ4192Ø
ØØØ6Ø2I            WRITE(6,912)                                                             ØØØ4193Ø
ØØØ61CI            IF(IP.LE.1) GO TO 14Ø                                                    ØØØ4194Ø
ØØØ632I            DO 135 I=1,M                                                             ØØØ4195Ø
ØØØ646I        135 WRITE(6,91Ø)(F(I,J),J=1,N)                                               ØØØ4196Ø
ØØØ6F4I            WRITE(6,911)                                                             ØØØ4197Ø
ØØØ71ØI        14Ø DO 6Ø I=1,N                                                              ØØØ4198Ø
ØØØ724I            DO 6Ø J=1,M                                                              ØØØ4199Ø
ØØØ738I            S=Ø.                                                                     ØØØ42ØØØ
ØØØ744I            DO 5Ø K=1,M                                                              ØØØ42Ø1Ø
```

```
ØØØ758I      5Ø S=S+F(K,I)*R(K,J)                                    ØØØ42Ø2Ø
ØØØ7C4I      6Ø W(I,J)=S                                             ØØØ42Ø3Ø
ØØØ81EI         DO 8Ø I=1,N                                          ØØØ42Ø4Ø
ØØØ832I         DO 8Ø J=1,N                                          ØØØ42Ø5Ø
ØØØ846I         S=Ø.                                                 ØØØ42Ø6Ø
ØØØ852I         SS=Ø.                                                ØØØ42Ø7Ø
ØØØ85EI         DO 7Ø K=1,M                                          ØØØ42Ø8Ø
ØØØ872I         S=S+B(I,K)*F(K,J)                                    ØØØ42Ø9Ø
ØØØ8C6I      7Ø SS=SS+W(I,K)*F(K,J)                                  ØØØ421ØØ
ØØØ932I         WQ(I,J)=Q(I,J)+SS                                    ØØØ4211Ø
ØØØ98ØI      8Ø WA(I,J)=A(I,J)-S                                     ØØØ4212Ø
ØØØ9FEI         CALL LYAPUN(WA,WQ,WP,W,N,NN,T)                       ØØØ4213Ø
ØØØA74I      99 CONTINUE                                             ØØØ4214Ø
          C                 F = (R.INV)(B')       P = (B(R.INV)B')   ØØØ4215Ø
ØØØA74I         DO 24Ø I=1,M                                         ØØØ4216Ø
ØØØA88I         DO 24Ø J=1,N                                         ØØØ4217Ø
ØØØA9CI         S=Ø.                                                 ØØØ4218Ø
ØØØAA8I         DO 23Ø K=1,M                                         ØØØ4219Ø
ØØØABCI     23Ø S=S+P(I,K)*B(J,K)                                    ØØØ422ØØ
ØØØB28I     24Ø F(I,J)=S                                             ØØØ4221Ø
ØØØB82I         DO 26Ø I=1,N                                         ØØØ4222Ø
ØØØB96I         DO 26Ø J=1,N                                         ØØØ4223Ø
ØØØBAAI         IF(IS.LT.2) WP(I,J)=Ø.                               ØØØ4224Ø
ØØØBE4I         S=Ø.                                                 ØØØ4225Ø
ØØØBFØI         DO 25Ø K=1,M                                         ØØØ4226Ø
ØØØCØ4I     25Ø S=S+B(I,K)*F(K,J)                                    ØØØ4227Ø
ØØØC7ØI     26Ø P(I,J)=S                                             ØØØ4228Ø
          C                                                          ØØØ4229Ø
ØØØCCAI     3ØØ DO 7ØØ IT=1,LIT                                      ØØØ423ØØ
          C                 W = (B(R.INV)B'P)                        ØØØ4231Ø
ØØØCDEI         DO 42Ø I=1,N                                         ØØØ4232Ø
ØØØCF2I         DO 42Ø J=1,N                                         ØØØ4233Ø
ØØØDØ6I         S=Ø.                                                 ØØØ4234Ø
ØØØD12I         DO 41Ø K=1,N                                         ØØØ4235Ø
ØØØD26I     41Ø S=S+P(I,K)*WP(K,J)                                   ØØØ4236Ø
ØØØD92I     42Ø W(I,J)=S                                             ØØØ4237Ø
          C                 WA = (A-B(R.INV)B'P)      WQ = (Q + PB(R.INV)B'P)   ØØØ4238Ø
ØØØDECI         DO 46Ø I=1,N                                         ØØØ4239Ø
ØØØEØØI         DO 46Ø J=1,N                                         ØØØ424ØØ
ØØØE14I         S=Ø.                                                 ØØØ4241Ø
ØØØE2ØI         DO 43Ø K=1,N                                         ØØØ4242Ø
ØØØE34I     43Ø S=S+WP(I,K)*W(K,J)                                   ØØØ4243Ø
ØØØEAØI         WQ(I,J)=S+Q(I,J)                                     ØØØ4244Ø
ØØØEEEI     46Ø WA(I,J)=A(I,J)-W(I,J)                                ØØØ4245Ø
          C                 SOLVE Ø = (WA)'(WP) + (WP)(WA) + (WQ)    ØØØ4246Ø
ØØØF8AI         CALL LYAPUN(WA,WQ,WP,W,N,NN,T)                       ØØØ4247Ø
ØØ1ØØØI         TRN=Ø.                                               ØØØ4248Ø
ØØ1ØØCI         DO 51Ø I=1,N                                         ØØØ4249Ø
ØØ1Ø2ØI     51Ø TRN=TRN+WP(I,I)                                      ØØØ425ØØ
          C                 TEST CONVERGENCE BY TRACE(P)             ØØØ4251Ø
ØØ1Ø68I         IF(IP.EQ.1) WRITE(6,9Ø5) IT,TRN                      ØØØ4252Ø
ØØ1Ø9CI         CRIT=ABS(TRN-TRO)/TRO                                ØØØ4253Ø
ØØ1ØCEI         IF(CRIT.LE.ERR) GO TO 515                            ØØØ4254Ø
ØØ1ØE6I         IF(IP.LE.1) GO TO 61Ø                                ØØØ4255Ø
ØØ1ØFCI     515 IF(IP.NE.1) WRITE(6,9Ø5) IT,TRN                      ØØØ4256Ø
ØØ113ØI         DO 52Ø I=1,N                                         ØØØ4257Ø
ØØ1144I     52Ø WRITE(6,91Ø)(WP(I,J),J=1,N)                          ØØØ4258Ø
```

```
ØØ11F4I    61Ø IF(CRIT.LE.ERR) GO TO 8ØØ                                    ØØØ4259Ø
ØØ12ØCI        IF(TRN.GE.1.E+6Ø) GO TO 1Ø1Ø                                 ØØØ426ØØ
ØØ1224I        TRO=TRN                                                      ØØØ4261Ø
ØØ123ØI    7ØØ CONTINUE                                                     ØØØ4262Ø
ØØ1248I        IS=Ø                                                         ØØØ4263Ø
ØØ125ØI        WRITE(6,9Ø7)      IT                                         ØØØ4264Ø
ØØ127ØI    8ØØ DO 82Ø I=1,M                                                 ØØØ4265Ø
ØØ1284I        DO 82Ø J=1,N                                                 ØØØ4266Ø
ØØ1298I        S=Ø.                                                         ØØØ4267Ø
ØØ12A4I        DO 81Ø K=1,N                                                 ØØØ4268Ø
ØØ12B8I    81Ø S=S+F(I,K)*WP(K,J)                                           ØØØ4269Ø
ØØ1324I    82Ø W(I,J)=S                                                     ØØØ427ØØ
ØØ137EI        DO 84Ø I=1,N                                                 ØØØ4271Ø
ØØ1392I        DO 83Ø J=1,N                                                 ØØØ4272Ø
ØØ13A6I    83Ø P(I,J)=WP(I,J)                                               ØØØ4273Ø
ØØ14Ø6I        DO 84Ø J=1,M                                                 ØØØ4274Ø
ØØ141AI    84Ø F(J,I)=W(J,I)                                                ØØØ4275Ø
ØØ1492I        IF(IP.LE.1) GO TO 777                                        ØØØ4276Ø
ØØ14A8I        WRITE(6,911)                                                 ØØØ4277Ø
ØØ14C4I        DO 85Ø I=1,M                                                 ØØØ4278Ø
ØØ14D8I    85Ø WRITE(6,91Ø)(F(I,J),J=1,N)                                   ØØØ4279Ø
ØØ1588I        GO TO 777                                                    ØØØ428ØØ
ØØ158EI   1Ø1Ø IS=-IS                                                       ØØØ4281Ø
ØØ159CI        WRITE(6,9Ø8) IT                                              ØØØ4282Ø
ØØ15BCI    777 IF(IP.GE.1) WRITE(6,9Ø6) IT,TRN                              ØØØ4283Ø
ØØ15FØI        RETURN                                                       ØØØ4284Ø
ØØ15F6I    9Ø1 FORMAT(/,1X,12Ø(1H*),//,2ØX,17HSUBROUTINE RICATI  //)        ØØØ4285Ø
ØØ162ØI    9Ø2 FORMAT(3ØX,38HRICCATI PROBLEM MATRICES A / B / Q / R  ,/)   ØØØ4286Ø
ØØ1654I    9Ø5 FORMAT(/,1ØX,16HRICATI ITERATION,I4,1ØX,4HCOST,1PE2Ø.6,/)   ØØØ4287Ø
ØØ1682I    9Ø6 FORMAT(//,2ØX,22HEXIT FROM RICATI AFTER ,I3,12H ITERATIONS ,1ØX, ØØØ4288Ø
             1   6HCOST =,1PE2Ø.6,/,1X,12Ø(1H*),//)                         ØØØ4289Ø
ØØ16D2I    9Ø7 FORMAT(///,2ØX,2ØHNO CONVERGENCE AFTER,I4,12H ITERATIONS  ,//) ØØØ429ØØ
ØØ17Ø8I    9Ø8 FORMAT(///,2ØX,18HSYSTEM UNSTABLE AT ,I4,13H-TH ITERATION  ,//) ØØØ4 1
ØØ173CI   91Ø  FORMAT(5X,1P1ØE12.3)
ØØ174EI    911 FORMAT(/,1ØX,2Ø(1H-),/)                                      ØØØ4293Ø
ØØ176ØI    912 FORMAT(/,1ØX,26HSTABILIZATION IS INDICATED        ,/)       ØØØ4294Ø
ØØ1788I        END                                                          ØØØ4295Ø
WARNING # 3Ø1 ************************************************************************
      >>> UNREFERENCED LABEL                                               <<<

          3ØØ

NO  ERRORS:F7D  RØ5-ØØ.ØØ  SUBROUTINE  RICATI    17/Ø2/83  16:59:Ø8  TABLE SPACE:   6 KB
STATEMENT BUFFER:   2Ø LINES/1321 BYTES    STACK SPACE: 186 WORDS
DOUBLE PRECISION FLOATING PT SUPPORT REQUIRED FOR EXECUTION
```

```
ØØØØØØI         SUBROUTINE MAT (B,A,M,N,ND,ZERO,D,IRANK)                            ØØØ3115Ø
        C          CALCULATES  RANK, DETERMINANT AND INVERSE  OF MATRIX.              ØØØ3116Ø
        C          METHOD IS DOUBLE PIVOTTED GAUSSIAN ELIMINATION.                    ØØØ3117Ø
        C       B IS A M BY N MATRIX STORED IN ND BY N MATRIX IN CALLING ROUTINE.    ØØØ3118Ø
        C       IRANK IS RETURNED AS RANK(B)         A IS USED AS WKG/STOR.          ØØØ3119Ø
        C       D IS RETURNED AS DET(B)                                               ØØØ312ØØ
        C       IF M=N A IS RETURNED AS (B.INVERSE)                                   ØØØ3121Ø
        C       ZERO IS USED TO TEST RELATIVE PIVOT SIZE.                             ØØØ3122Ø
        C       IF S/R USED BY RECURSIVE ROUTINE SET IPRNT=Ø TO SUPPRESS PRINTING.ØØØ3123Ø
        C        REVERSE THE FOLLOWING 2 CARDS TO OBTAIN  THE REAL*4 VERSION.        ØØØ3124Ø
ØØØØØ4I         IMPLICIT REAL*8(A-H,O-Z)                                            ØØØ3125Ø
ØØØØØ4I         DIMENSION A(ND,N)        ,B(ND,N)                                   ØØØ3127Ø
ØØØØØAI         DIMENSION ISWCH (1ØØ),JSWCH(1ØØ)                                    ØØØ3128Ø
ØØØØØAI         ABS(Q)=DABS(Q)                                                      ØØØ3126Ø
ØØØ3CCI         NSW=1ØØ                                                             ØØØ3129Ø
ØØØ3D6I         IPRNT=1                                                             ØØØ313ØØ
ØØØ3DEI         IPRNT=Ø                                                             ØØØ3131Ø
ØØØ3E6I         IRANK=Ø                                                             ØØØ3132Ø
ØØØ3EEI         D=1.                                                                ØØØ3133Ø
ØØØ3FAI         IF(M.NE.N) D=Ø.                                                     ØØØ3134Ø
ØØØ418I         DO 1Ø I1=1,M                                                        ØØØ3135Ø
ØØØ42CI         DO 1Ø I2=1,N                                                        ØØØ3136Ø
ØØØ44ØI      1Ø A(I1,I2)=B(I1,I2)                                                   ØØØ3137Ø
ØØØ4B8I         DO 1 I=1,NSW                                                        ØØØ3138Ø
ØØØ4CCI         ISWCH(I)=Ø                                                          ØØØ3139Ø
ØØØ4DAI       1 JSWCH(I)=Ø                                                          ØØØ314ØØ
ØØØ5ØØI         MM=M                                                                ØØØ3141Ø
ØØØ5ØCI         IF(N.LT.M)    MM=N                                                  ØØØ3142Ø
ØØØ52AI         DO 2 I=1,MM                                                         ØØØ3143Ø
ØØØ53EI         AMAX=Ø.Ø                                                            ØØØ3144Ø
ØØØ54AI         DO 3 I1=1,M                                                         ØØØ3145Ø
ØØØ55EI         DO 4 I2=1,N                                                         ØØØ3146Ø
ØØØ572I         IF(ISWCH(I1).NE.Ø.OR.JSWCH(I2).NE.Ø) GO TO  4                       ØØØ3147Ø
ØØØ5A4I         IF (ABS(A(I1,I2)).LE.AMAX) GO TO 4                                  ØØØ3148Ø
ØØØ5F8I         IPIVOT=I1                                                           ØØØ3149Ø
ØØØ6Ø4I         JPIVOT=I2                                                           ØØØ315ØØ
ØØØ61ØI         ABIG=A(I1,I2)                                                       ØØØ3151Ø
ØØØ63AI         AMAX=ABS(ABIG)                                                      ØØØ3152Ø
ØØØ654I       4 CONTINUE                                                            ØØØ3153Ø
ØØØ66CI       3 CONTINUE                                                            ØØØ3154Ø
ØØØ684I         D=D*ABIG                                                            ØØØ3155Ø
ØØØ696I         IF(I.NE.1) GO TO 33                                                 ØØØ3156Ø
ØØØ6ACI         BMAX=AMAX                                                           ØØØ3157Ø
ØØØ6B8I         IF((AMAX.LE.ZERO).AND.(M.EQ.N)) GO TO 999                           ØØØ3158Ø
ØØØ6E2I         IF(AMAX.LE.ZERO) GO TO 99                                           ØØØ3159Ø
ØØØ6FAI         GO TO 34                                                            ØØØ316ØØ
ØØØ7ØØI      33 CMAX=AMAX/BMAX                                                      ØØØ3161Ø
ØØØ712I         IF((CMAX.LE.ZERO).AND.(M.EQ.N)) GO TO 999                           ØØØ3162Ø
ØØØ73CI         IF(CMAX.LE.ZERO) GO TO 99                                           ØØØ3163Ø
ØØØ754I      34 IRANK=IRANK+1                                                       ØØØ3164Ø
ØØØ762I         APIVOT=1.Ø/A(IPIVOT,JPIVOT)                                         ØØØ3165Ø
ØØØ792I         A(IPIVOT,JPIVOT)=APIVOT                                             ØØØ3166Ø
ØØØ7BCI         ISWCH(IPIVOT)=JPIVOT                                                ØØØ3167Ø
ØØØ7CEI         JSWCH(JPIVOT)=IPIVOT                                                ØØØ3168Ø
ØØØ7EØI         DO 5 I1=1,M                                                         ØØØ3169Ø
ØØØ7F4I         DO 6 I2=1,N                                                         ØØØ317ØØ
ØØØ8Ø8I         IF(I1.EQ.IPIVOT.OR.I2.EQ.JPIVOT)  GO TO 6                           ØØØ3171Ø
```

```
ØØØ832I          A(I1,I2)=A(I1,I2)-A(IPIVOT,I2)*A(I1,JPIVOT)*APIVOT         ØØØ3172Ø
ØØØ8CAI        6 CONTINUE                                                   ØØØ3173Ø
ØØØ8E2I        5 CONTINUE                                                   ØØØ3174Ø
ØØØ8FAI          DO 17Ø I1=1,M                                              ØØØ3175Ø
ØØØ9ØEI          IF(I1.EQ.IPIVOT) GO TO 17Ø                                 ØØØ3176Ø
ØØØ926I          A(I1,JPIVOT)=A(I1,JPIVOT)*APIVOT                           ØØØ3177Ø
ØØØ974I      17Ø CONTINUE                                                   ØØØ3178Ø
ØØØ98CI          DO 175 J1=1,N                                              ØØØ3179Ø
ØØØ9AØI          IF(J1.EQ.JPIVOT) GO TO 175                                 ØØØ318ØØ
ØØØ9B8I          A(IPIVOT,J1)=-A(IPIVOT,J1)*APIVOT                          ØØØ3181Ø
ØØØAØEI      175 CONTINUE                                                   ØØØ3182Ø
ØØØA26I        2 CONTINUE                                                   ØØØ3183Ø
ØØØA3EI       99 IF(IPRNT.EQ.1) WRITE(6,9Ø1)M,N,IRANK,BMAX,AMAX,D           ØØØ3184Ø
ØØØA8ØI          IF(M.NE.N) GO TO 771                                       ØØØ3185Ø
ØØØA98I          DO 77 I1=1,M                                               ØØØ3186Ø
ØØØAACI       11 IF(ISWCH(I1).EQ.I1) GO TO 77                               ØØØ3187Ø
ØØØACAI          K=ISWCH(I1)                                                ØØØ3188Ø
ØØØADCI          DO 88 J=1,M                                                ØØØ3189Ø
ØØØAFØI          TEMP=A(I1,J)                                               ØØØ319ØØ
ØØØB1AI          A(I1,J)=A(K,J)                                             ØØØ3191Ø
ØØØB62I          A(K,J)=TEMP                                                ØØØ3192Ø
ØØØB8CI       88 CONTINUE                                                   ØØØ3193Ø
ØØØBA4I          ITEMP=ISWCH(I1)                                            ØØØ3194Ø
ØØØBB6I          ISWCH(I1)=ISWCH(K)                                         ØØØ3195Ø
ØØØBCEI          ISWCH(K)=ITEMP                                             ØØØ3196Ø
ØØØBEØI          GO TO 11                                                   ØØØ3197Ø
ØØØBE4I       77 CONTINUE                                                   ØØØ3198Ø
ØØØBFCI          DO 55  I=1,M                                               ØØØ3199Ø
ØØØC1ØI       12 IF (JSWCH(I).EQ.I)  GO TO 55                               ØØØ32ØØØ
ØØØC2EI          K=JSWCH(I)                                                 ØØØ32Ø1Ø
ØØØC4ØI          DO 44 J=1,M                                                ØØØ32Ø2Ø
ØØØC54I          TEMP=A(J,I)                                                ØØØ32Ø3Ø
ØØØC7EI          A(J,I)=A(J,K)                                              ØØØ32Ø4Ø
ØØØCC6I          A(J,K)=TEMP                                                ØØØ32Ø5Ø
ØØØCFØI       44 CONTINUE                                                   ØØØ32Ø6Ø
ØØØDØ8I          ITEMP=JSWCH(I)                                             ØØØ32Ø7Ø
ØØØD1AI          JSWCH(I)=JSWCH(K)                                          ØØØ32Ø8Ø
ØØØD32I          JSWCH(K)=ITEMP                                             ØØØ32Ø9Ø
ØØØD44I          GO TO 12                                                   ØØØ321ØØ
ØØØD48I       55 CONTINUE                                                   ØØØ3211Ø
ØØØD6ØI      771 RETURN                                                     ØØØ3212Ø
ØØØD66I      999 D=Ø.                                                       ØØØ3213Ø
ØØØD72I          WRITE(6,2ØØ) M,N,IRANK,BMAX,AMAX                           ØØØ3214Ø
ØØØDAØI      9Ø1 FORMAT(5X,7HRANK OF,I3,3H BY,I3,1ØH MATRIX IS,I3,5X,       ØØØ3215Ø
                1 24HFIRST AND LAST PIVOTS --,1PE1Ø.3,1PE14.3,5X,14HDETERMINANT --, ØØØ3216Ø
                2 1PE13.3)                                                  ØØØ3217Ø
ØØØEØAI      2ØØ FORMAT(5X,7HRANK OF,I3,3H BY,I3,1ØH MATRIX IS,I3,5X,       ØØØ3218Ø
                1 24HFIRST AND LAST PIVOTS --,1PE1Ø.3,1PE14.3,5X,15H ** SINGULAR **)ØØØ3219Ø
ØØØE7ØI          RETURN                                                     ØØØ322ØØ
ØØØE76I          END                                                        ØØØ3221Ø
```

```
NO  ERRORS:F7D  RØ5-ØØ.ØØ  SUBROUTINE  MAT       17/Ø2/83  16:59:18  TABLE SPACE:   3 KB
STATEMENT BUFFER:   2Ø LINES/1321 BYTES    STACK SPACE: 183 WORDS
DOUBLE PRECISION FLOATING PT SUPPORT REQUIRED FOR EXECUTION
```

```
ØØØØØØI           SUBROUTINE LYAPUN (A,C,Q,WS1,N,NDIM,T)                              ØØØ3Ø16Ø
             C                                                                         ØØØ3Ø17Ø
             C       SOLVES THE LYAPUNOV EQUATION :  A'Q + QA + C = Ø                  ØØØ3Ø18Ø
             C                                                                         ØØØ3Ø19Ø
             C       CALLS S/R MAT FOR MATRIX INVERSION                                ØØØ3Ø2ØØ
             C       TYPICAL STEP-SIZE = 1.E-4, 1.E-7 FOR WIDE SPREAD OF EIGENVALUESØØØ3Ø21Ø
             C                                                                         ØØØ3Ø22Ø
ØØØØØ4I           IMPLICIT REAL*8(A-H,O-Z)                                             ØØØ3Ø23Ø
ØØØØØ4I           DIMENSION A(NDIM,N), Q(NDIM,N),C(NDIM,N),WS1(NDIM,N)                 ØØØ3Ø28Ø
             C-1                                                    )MUS(DNR=)MUS(ØØØ3Ø26Ø
             C-1                                                     )X(ELBD=)X(BØØØ3Ø27Ø
ØØØØØAI           DOUBLE PRECISION SUM                                                 ØØØ3Ø29Ø
ØØØØØAI           LOGICAL FLAG                                                         ØØØ3Ø3ØØ
ØØØØØAI           DUB(X)=X                                                             ØØØ3Ø24Ø
ØØØØACI           RD(SUM)=SUM                                                          ØØØ3Ø25Ø
ØØØ114I           FLAG=.FALSE.                                                         ØØØ3Ø31Ø
ØØØ11CI           MAXIT=4Ø                                                             ØØØ3Ø32Ø
ØØØ126I           ZERO=1.E-12                                                          ØØØ3Ø33Ø
ØØØ132I           ERR=1.E-12                                                           ØØØ3Ø34Ø
ØØØ13EI           KK=-1                                                                ØØØ3Ø35Ø
ØØØ146I         5 DO 1Ø I=1,N                                                          ØØØ3Ø36Ø
ØØØ15AI           DO 1Ø J=1,N                                                          ØØØ3Ø37Ø
ØØØ16EI        1Ø WS1(I,J)=Ø.Ø                                                         ØØØ3Ø38Ø
ØØØ1C8I           DO 11 I=1,N                                                          ØØØ3Ø39Ø
ØØØ1DCI        11 WS1(I,I)=1.Ø                                                         ØØØ3Ø4ØØ
ØØØ21EI           T1=Ø.5*T                                                             ØØØ3Ø41Ø
ØØØ23ØI           T2=T1*T/6.Ø                                                          ØØØ3Ø42Ø
ØØØ248I           DO 2Ø I=1,N                                                          ØØØ3Ø43Ø
ØØØ25CI           DO 2Ø K=1,N                                                          ØØØ3Ø44Ø
ØØØ27ØI           SUM=Ø.ØDØ                                                            ØØØ3Ø45Ø
ØØØ27CI           DO 15 J=1,N                                                          ØØØ3Ø46Ø
ØØØ29ØI        15 SUM=SUM+T2*A(I,J)*A(J,K)                                             ØØØ3Ø47Ø
ØØØ3Ø2I        2Ø WS1(I,K)=WS1(I,K)+RD(SUM)                                            ØØØ3Ø48Ø
ØØØ38EI           DO 21 I=1,N                                                          ØØØ3Ø49Ø
ØØØ3A2I           DO 21 J=1,N                                                          ØØØ3Ø5ØØ
ØØØ3B6I        21 A(I,J)=RD(DUB(T1)*DUB(A(I,J)))                                       ØØØ3Ø51Ø
ØØØ488I           DO 24 I=1,N                                                          ØØØ3Ø52Ø
ØØØ49CI           DO 24 J=1,N                                                          ØØØ3Ø53Ø
ØØØ4BØI           Q(I,J)=WS1(I,J)-A(I,J)                                               ØØØ3Ø54Ø
ØØØ51CI        24 WS1(I,J)=WS1(I,J)+A(I,J)                                             ØØØ3Ø55Ø
ØØØ5B8I           CALL MAT(Q,Q,N,N,NDIM,ZERO,DET,IRANK)                                ØØØ3Ø56Ø
ØØØ61ØI           DO 28 I=1,N                                                          ØØØ3Ø57Ø
ØØØ624I           DO 28 J=1,N                                                          ØØØ3Ø58Ø
ØØØ638I           SUM=Ø.                                                               ØØØ3Ø59Ø
ØØØ644I           DO 26 K=1,N                                                          ØØØ3Ø6ØØ
ØØØ658I        26 SUM=SUM + Q(I,K)*WS1(K,J)                                            ØØØ3Ø61Ø
ØØØ6C4I        28 A(I,J)=SUM                                                           ØØØ3Ø62Ø
ØØØ71EI           DO 3Ø I=1,N                                                          ØØØ3Ø63Ø
ØØØ732I           DO 3Ø J=I,N                                                          ØØØ3Ø64Ø
ØØØ74AI           C(I,J)=RD(DUB(T)*DUB(C(I,J))/3.)                                     ØØØ3Ø65Ø
ØØØ7F4I           Q(I,J)=C(I,J)                                                        ØØØ3Ø66Ø
ØØØ83CI        3Ø Q(J,I)=Q(I,J)                                                        ØØØ3Ø67Ø
ØØØ8B4I        4Ø KK=KK+1                                                              ØØØ3Ø68Ø
ØØØ8C2I           IF (KK.EQ.Ø) GO TO 8Ø                                                ØØØ3Ø69Ø
ØØØ8D8I           DO 65 I=1,N                                                          ØØØ3Ø7ØØ
ØØØ8ECI           DO 65 K=1,N                                                          ØØØ3Ø71Ø
ØØØ9ØØI           SUM=Ø.ØDØ                                                            ØØØ3Ø72Ø
```

```
 0000908CI          DO 60 J=1,N                                                 00030730
 0000920I       60  SUM=SUM+A(I,J)*A(J,K)                                       00030740
 000098CI       65  WS1(I,K)=SUM                                                00030750
 00009E6I           DO 70 I=1,N                                                 00030760
 00009FAI           DO 70 J=1,N                                                 00030770
 0000A0EI       70  A(I,J)=WS1(I,J)                                             00030780
 0000A86I       80  DO 95 I=1,N                                                 00030790
 0000A9AI           DO 95 K=1,N                                                 00030800
 0000AAEI           SUM=0.0D0                                                   00030810
 0000ABAI           DO 90 J=1,N                                                 00030820
 0000ACEI       90  SUM=SUM+Q(I,J)*A(J,K)                                       00030830
 0000B3AI       95  WS1(I,K)=SUM                                                00030840
 0000B94I           DO 110 I=1,N                                                00030850
 0000BA8I           DO 110 K=I,N                                                00030860
 0000BC0I           SUM=0.0D0                                                   00030870
 0000BCCI           DO 100 J=1,N                                                00030880
 0000BE0I      100  SUM=SUM+A(J,I)*WS1(J,K)                                     00030890
 0000C4CI           IF(DABS(SUM).GT.ERR) FLAG=.FALSE.                           00030900
 0000C78I      737  IF (KK.GT.0) GO TO 105                                      00030910
 0000C8EI           Q(I,K)=2.0*(Q(I,K)+2.0*SUM)                                 00030920
 0000CE8I           GO TO 110                                                   00030930
 0000CEEI      105  Q(I,K)=Q(I,K)+SUM                                           00030940
 0000D3CI      110  CONTINUE                                                    00030950
 0000D6CI           IF (FLAG) GO TO 130                                         00030960
 0000D7EI           IF(KK.GT.MAXIT) GO TO 130                                   00030970
 0000D96I           DO 120 I=1,N                                                00030980
 0000DAAI           DO 120 J=I,N                                                00030990
 0000DC2I      120  Q(J,I)=Q(I,J)                                               00031000
 0000E3AI           FLAG=.TRUE.                                                 00031010
 0000E42I           GO TO 40                                                    00031020
 0000E46I      130  DO 140 I=1,N                                                00031030
 0000E5AI           DO 140 J=I,N                                                00031040
 0000E72I           Q(I,J)=Q(I,J)-C(I,J)                                        00031050
 0000EDEI      140  Q(J,I)=Q(I,J)                                               00031060
 0000F56I           IF(KK.LE.MAXIT) GO TO 77                                    00031070
 0000F6EI      250  WRITE(6,9300)                                               00031080
 0000F88I       77  WRITE(6,8765) KK                                            00031090
 0000FA8I           RETURN                                                      00031100
 0000FAEI     8765  FORMAT(  43X,22HEXIT FROM LYAPUN AFTER  ,I3, 11H ITERATIONS  , /) 00031110
 0000FE0I     9300  FORMAT(/,10X,5(1H*),26H NO CONVERGENCE IN LYAPUN ,5(1H*),/) 00031120
 0001014I           END                                                         00031130
WARNING # 301 ****************************************************************************
          >>> UNREFERENCED LABEL                                              <<<

          5      737      250

NO  ERRORS:F7D  R05-00.00  SUBROUTINE  LYAPUN    17/02/83  16:59:31  TABLE SPACE:   4 KB
STATEMENT BUFFER:   20 LINES/1321 BYTES    STACK SPACE: 214 WORDS
DOUBLE PRECISION FLOATING PT SUPPORT REQUIRED FOR EXECUTION
```

```
ØØØØØØI          SUBROUTINE SETPOL(A,B,F,POLZ,N,M,ND,ZERO,JGUD,NSET,R,U,V)
            C    POLE-ASSIGNMENT FOR PAIR(A,B) ...
            C      RESTRICTION: ASSIGNMENT IS ALONG REAL AXIS ONLY.
            C                   CONTROLLABLE MODES ARE SET TO (POLZ)
            C                   NSET POLES ARE ASSIGNED, RIGHTMOST FIRST
            C                   NSET  .LT. Ø PRODUCES DETAILED PRINTOUT
            C                   JGUD SET = 1 IF ASSIGNMENT IS SATISFACTORY
            C    WORKING STORAGE IS R, U, V
ØØØØØ4I          DIMENSION A(ND,N),B(ND,M),F(ND,N),POLZ(N)
ØØØØØAI          DIMENSION VR(2Ø),VI(2Ø),IANA(2Ø),VB(2Ø),FM(2Ø)
ØØØØØAI          DIMENSION R(ND,N),V(ND,N),U(ND,N)
ØØØØØAI          DOUBLE PRECISION VB,S
ØØØØØAI          MSET=IABS(NSET)
ØØØ256I          WRITE(6,9Ø1) MSET,N
ØØØ278I          IF(NSET.GT.Ø) GO TO 5
ØØØ28EI          WRITE(6,9Ø2)
ØØØ2A8I          DO 3 I=1,N
ØØØ2BCI        3 WRITE(6,922) (A(I,J),J=1,N)
ØØØ36CI          WRITE(6,966)
ØØØ388I          DO 4 I=1,N
ØØØ39CI        4 WRITE(6,922) (B(I,J),J=1,M)
ØØØ44CI        5 RYTMAX=POLZ(1)
ØØØ45CI          DO 6 I=1,MSET
ØØØ47ØI          IF(POLZ(I).GT.RYTMAX) RYTMAX=POLZ(I)
ØØØ4AAI        6 CONTINUE
ØØØ4C2I          DO 1Ø I=1,N
ØØØ4D6I          DO 9 J=1,M
ØØØ4EAI        9 F(J,I)=Ø.
ØØØ52CI          DO 1Ø J=1,N
ØØØ54ØI       1Ø R(J,I)=A(I,J)
            C --- SET POLES 1 MODE AT A TIME ---
ØØØ5B8I      1Ø1 DO 5ØØ MO=1,MSET
ØØØ5CCI          WRITE(6,966)
            C --- GET EIGENVALUES AND LEFT EIGENVECTORS OF (A+B*F) ---
ØØØ5E8I          T1=24.
            C-2
ØØØ5F4I          CALL EIGENP(N,ND,R,T1,VR,VI,V,U,IANA)
            C-2
ØØØ66ØI          IF(NSET.GT.Ø) GO TO 1Ø7
ØØØ676I          WRITE(6,9Ø8)
ØØØ69ØI          DO 1Ø5 J=1,N
ØØØ6A4I          IF(IANA(J).EQ.Ø) WRITE(6,9Ø9)
ØØØ6D4I      1Ø5 WRITE(6,91Ø) J,VR(J),VI(J),IANA(J)
ØØØ754I      1Ø7 WRITE(6,9Ø5) (POLZ(I), I=1,MSET)
ØØØ7DCI          WRITE(6,9Ø3)(VR(I),I=1,N)
            C      --- FIND RIGHTMOST POLE ---
ØØØ85CI          MM=1
ØØØ864I          RYT=VR(1)
ØØØ86EI          DO 11Ø I=2,N
ØØØ882I          IF(VR(I).LT.RYT) GO TO 11Ø
ØØØ8AØI          MM=I
ØØØ8ACI          RYT=VR(MM)
ØØØ8BEI      11Ø CONTINUE
ØØØ8D6I          IF(RYT.LT.RYTMAX) GO TO 311
ØØØ8EEI          WRITE(6,93ØØ) MM,POLZ(MO)
            C  --- CALCULATE FEEDBACK GAINS, CHOOSE THE LOWEST ---
ØØØ92CI          DO 21Ø J=1,M
```

```
ØØØ94ØI           VB(J)=Ø.
ØØØ952I           DO 21Ø I=1,N
ØØØ966I      21Ø  VB(J)=VB(J)+V(I,MM)*B(I,J)
ØØØAØ2I           JJ=1
ØØØAØAI           VDB=Ø.
ØØØA16I           DO 22Ø J=1,M
ØØØA2AI           IF(DABS(VB(J)).LT.VDB) GO TO 22Ø
ØØØA72I           JJ=J
ØØØA7EI           VDB=DABS(VB(JJ))
ØØØABAI      22Ø  CONTINUE
ØØØAD2I           IF(NSET.LT.Ø.) WRITE(6,917) (VB(J), J=1,M)
ØØØB64I           IF (VDB.LT.ZERO) GO TO 3Ø1
           C  --- MODE MM IS CONTROLLABLE ---
ØØØB7CI           FHAT=(POLZ(MO)-VR(MM))/VB(JJ)
ØØØBC6I           DO 246 I=1,N
ØØØBDAI           FM(I)=FHAT*V(I,MM)
ØØØC1ØI      246  F(JJ,I)=F(JJ,I)+FM(I)
ØØØC7CI           IF(NSET.LT.Ø) WRITE(6,919) (FM(I),I=1,N)
ØØØDØCI           GO TO 45Ø
ØØØD12I      3Ø1  WRITE(6,92ØØ) VR(MM)
ØØØD44I           GO TO 45Ø
ØØØD4AI      311  WRITE(6,94ØØ)
ØØØD64I           GO TO 45Ø
           C  --- GET NEW (A+B*F)TRANSPOSE ---
ØØØD6AI      45Ø  DO 455 I=1,N
ØØØD7EI           DO 455 J=1,N
ØØØD92I           S=Ø.DØ
ØØØD9EI           DO 453 K=1,M
ØØØDB2I      453  S=S+B(I,K)*F(K,J)
ØØØE2AI           V(I,J)=S+A(I,J)
ØØØE9ØI      455  R(J,I)=V(I,J)
ØØØFØ8I           IF(RYT.LT.RYTMAX) GO TO 6ØØ
ØØØF2ØI      5ØØ  CONTINUE
           C  --- CHECK POLE ASSGINMENT ---
ØØØF38I      6ØØ  WRITE(6,912)
ØØØF54I           DO 512 I=1,M
ØØØF68I      512  WRITE(6,922) (F(I,J),J=1,N)
ØØ1Ø18I           IF(NSET.GT.Ø) GO TO 514
ØØ1Ø2EI           WRITE(6,913)
ØØ1Ø48I           DO 513 I=1,N
ØØ1Ø5CI      513  WRITE(6,922) (V(I,J),J=1,N)
ØØ11ØCI      514  JGUD=1
ØØ1114I           IF(RYT.LT.RYTMAX) GO TO 625
ØØ112CI           WRITE(6,966)
ØØ1148I           T1=24.
           C-2
ØØ1154I           CALL EIGENP(N,ND,R,T1,VR,VI,V,U,IANA)
ØØ11CØI           WRITE(6,9Ø5) (POLZ(I),I=1,MSET)
ØØ1248I           WRITE(6,9Ø3) (VR(I),I=1,N)
ØØ12C8I           WRITE(6,9Ø4) (VI(I),I=1,N)
ØØ1348I           IF(MSET.LT.N) GO TO 625
ØØ136ØI           DO 61Ø I=1,N
ØØ1374I           IF(VR(I).GT.RYTMAX*.9) GO TO 62Ø
ØØ139AI      61Ø  CONTINUE
ØØ13B2I      625  WRITE(6,967)
ØØ13CCI           RETURN
ØØ13D2I      62Ø  JGUD=Ø
```

```
ØØ13DAI         GO TO 625
             C
ØØ13DEI       9Ø1 FORMAT(//,1X,125('*'),/2ØX,'SETPOL :',1ØX,'POLE-
                 *ASSIGNMENT OF',' CONTROLLABLE MODES ',//
                 *,3ØX,'SET',I3,' MODES OF',I3,'-TH ORDER SYSTEM',//)
ØØ1468I       9Ø2 FORMAT(2ØX,'SYSTEM MATRICES :A AND B',/)
ØØ148EI       9Ø3 FORMAT(2X,'REAL PART OF SPECTRUM :',1ØF9.3)
ØØ14B6I       9Ø4 FORMAT(2X,'IMAG PART OF SPECTRUM :',1ØF9.3)
ØØ14DEI       9Ø5 FORMAT(/,5X,'DESIRED POLES :',5X,1ØF9.3)
ØØ15Ø2I       9Ø8 FORMAT(1ØX,'EIGENVALUES : REAL AND IMAG. PARTS :')
ØØ1532I       9Ø9 FORMAT(6ØX,1Ø('*'),'REAL PART NOT FOUND')
ØØ1558I       91Ø FORMAT(I5,5X,1P2E2Ø.8,I1Ø)
ØØ156EI       912 FORMAT(/,2ØX,'FEEDBACK MATRIX F',/)
ØØ158EI       913 FORMAT(//,2ØX,'CLOSED-LOOP MATRIX (A+B*F)',/)
ØØ15B8I       917 FORMAT(2X,'B-TILDA ROW :',1P1ØE11.3,/15X,1P1ØE11.3)
ØØ15E4I       919 FORMAT(5X,'FEEDBACK :',1P1ØE11.3,/,15X,1P1ØE11.3)
ØØ16ØEI       922 FORMAT(5X,1P1ØE11.3)
ØØ162ØI       966 FORMAT(//,1X)
ØØ162CI       967 FORMAT(//,2ØX,'EXIT FROM SETPOL',/,1X,125('*'),//)
ØØ1656I      92ØØ FORMAT(4ØX,2Ø('='),'MODE UNCONTROLLABLE :',F2Ø.4)
ØØ168ØI      93ØØ FORMAT(/,5X,'ASSIGNING MODE',I3,'TO',F1Ø.5,'------')
ØØ16ACI      94ØØ FORMAT(/,2ØX,'ALL POLES ARE LEFTWARD OF RIGHTMOST
                 *DESIRED POLE',/)
ØØ16FAI         END
WARNING # 3Ø1 ************************************************************************
          >>> UNREFERENCED LABEL                                                   <<<

              1Ø1

NO  ERRORS:F7D  RØ5-ØØ.ØØ  SUBROUTINE  SETPOL   25/Ø4/83  12:Ø3:4Ø  TABLE SPACE:   6 KB
STATEMENT BUFFER:   2Ø LINES/1321 BYTES     STACK SPACE: 186 WORDS
SINGLE PRECISION FLOATING PT SUPPORT REQUIRED FOR EXECUTION
DOUBLE PRECISION FLOATING PT SUPPORT REQUIRED FOR EXECUTION
```

TABLE 1.1 PERFORMANCE LIMITS [1]

Degree of freedom	Performance limits		
	Position	Velocity	Acceleration
Horizontal x	Forward 1.245 m Aft 1.219 m	±0.610m/sec	±0.6g
Lateral y	Left 1.219 m Right 1.219 m	±0.610m/sec	±0.6g
Vertical z	Up 0.991 m Down 0.762 m	±0.610m/sec	±0.6g
Yaw ψ	$\pm 32^{\circ}$	±15 /sec	±50 /sec
Pitch θ	$+30^{\circ}$ -20°	±15 /sec	±50 /sec
Roll ϕ	$\pm 22^{\circ}$	±15 /sec	±50 /sec

TABLE 6.1 Least relative negative overshoot of $\hat{\ddot{x}}_c, \hat{\ddot{y}}_c, \hat{\ddot{z}}_c$

1) The linear washout filter

case NO.	1	2	3	4
	-0.22	-0.32	-0.46	-0.12

2) The adaptive washout filter

case NO. / channel	1		2		3		4	
	a	b	a	b	a	b	a	b
Longitudinal	-0.347	-0.53	-0.39	-0.64	-0.33	-0.46	-0.28	-0.36
Lateral	-0.6	-0.82	-0.46	-0.63	-0.35	-0.62	-0.27	-0.45
Vertical	-0.31	-0.45	-0.37	-0.31	-0.23	-0.25	-0.29	-0.28

3) The nonlinear optimal washout filter

case NO. / channel	1		2		3		4	
	a	b	a	b	a	b	a	b
Longitudinal	-0.16	0.0	-0.02	-0.03	0.0	0.0	0.0	0.0
Lateral	-0.17	-0.2	-0.15	-0.17	-0.15	-0.18	-0.17	-0.2
Vertical	-0.12	-0.2	-0.11	-0.2	-0.1	-0.19	-0.12	-0.17

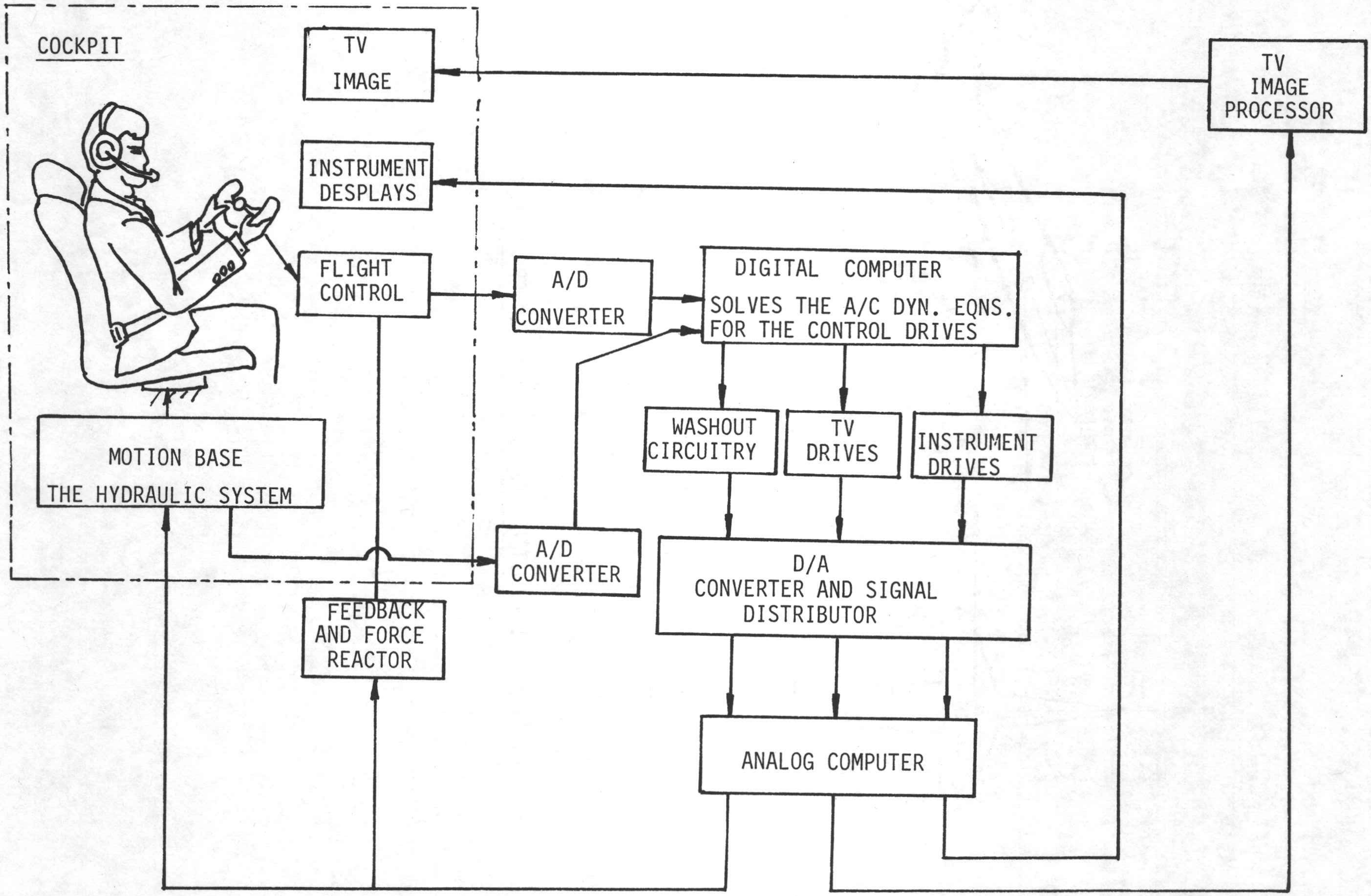


FIGURE 2.1 AN ILLUSTRATIVE CONFIGURATION OF A SIMULATOR

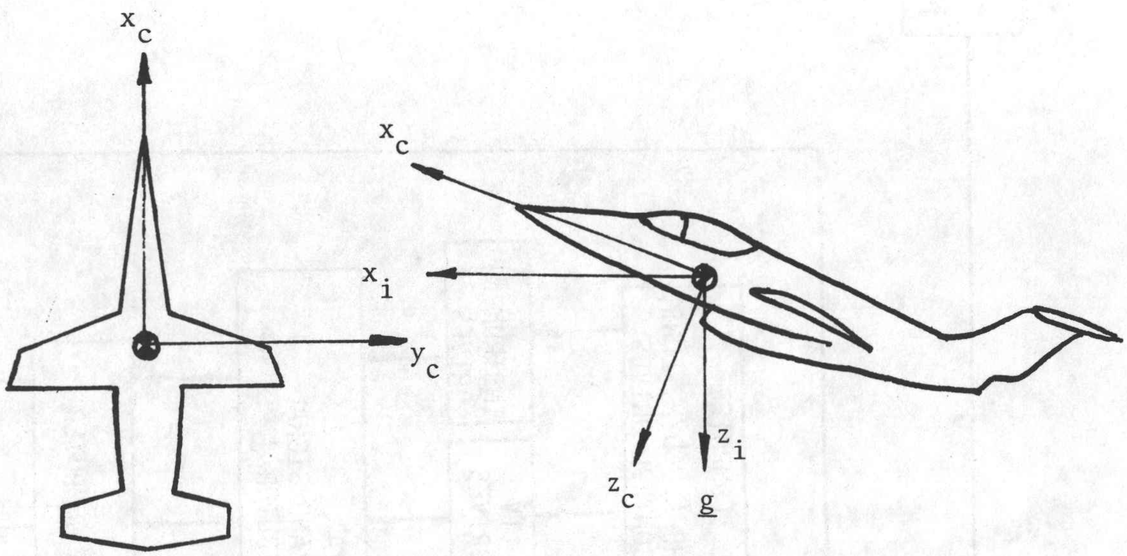
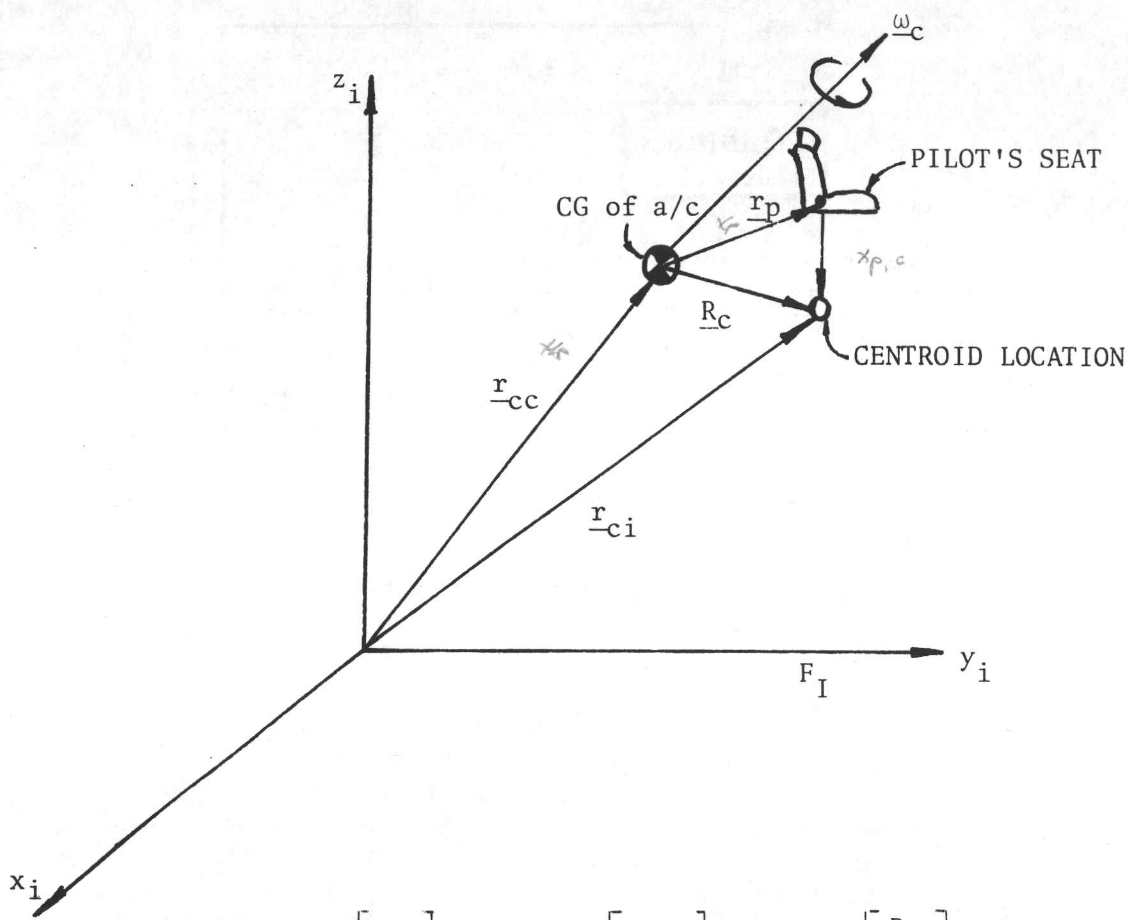


Figure 3.1 DEFINITION OF COCKPIT-FIXED FRAME F_c



where $\underline{r}_p = \begin{bmatrix} x_p \\ y_p \\ z_p \end{bmatrix}$, $\underline{r}_{pc} = \begin{bmatrix} x_{pc} \\ y_{pc} \\ z_{pc} \end{bmatrix}$, $\underline{R}_c = \begin{bmatrix} R_x \\ R_y \\ R_z \end{bmatrix}$

in body-fixed frame.

FIGURE 3.2 CENTROID TRANSFORMATION

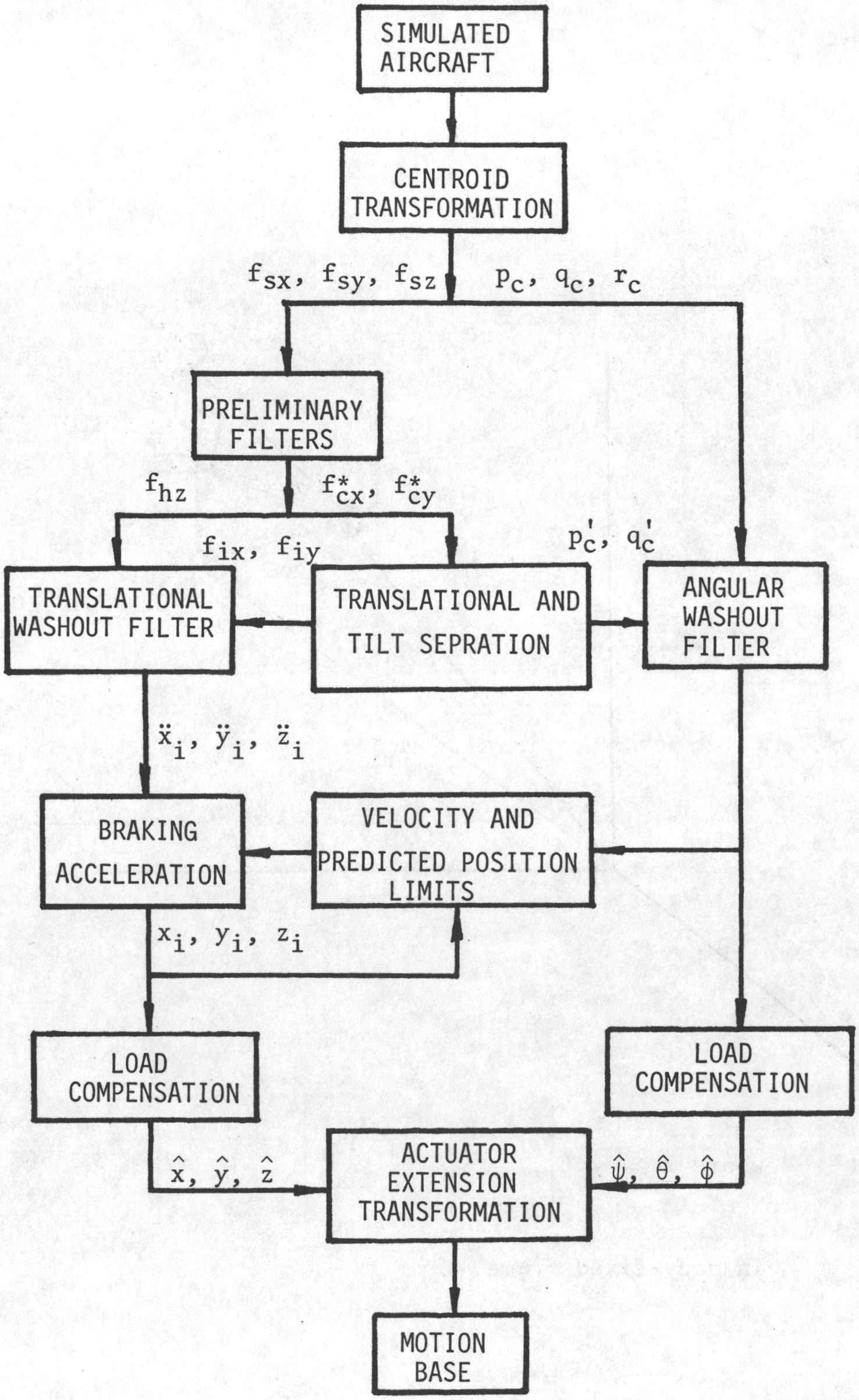


FIGURE 4.1 BLOCK DIAGRAM OF LINEAR WASHOUT FILTER [1]

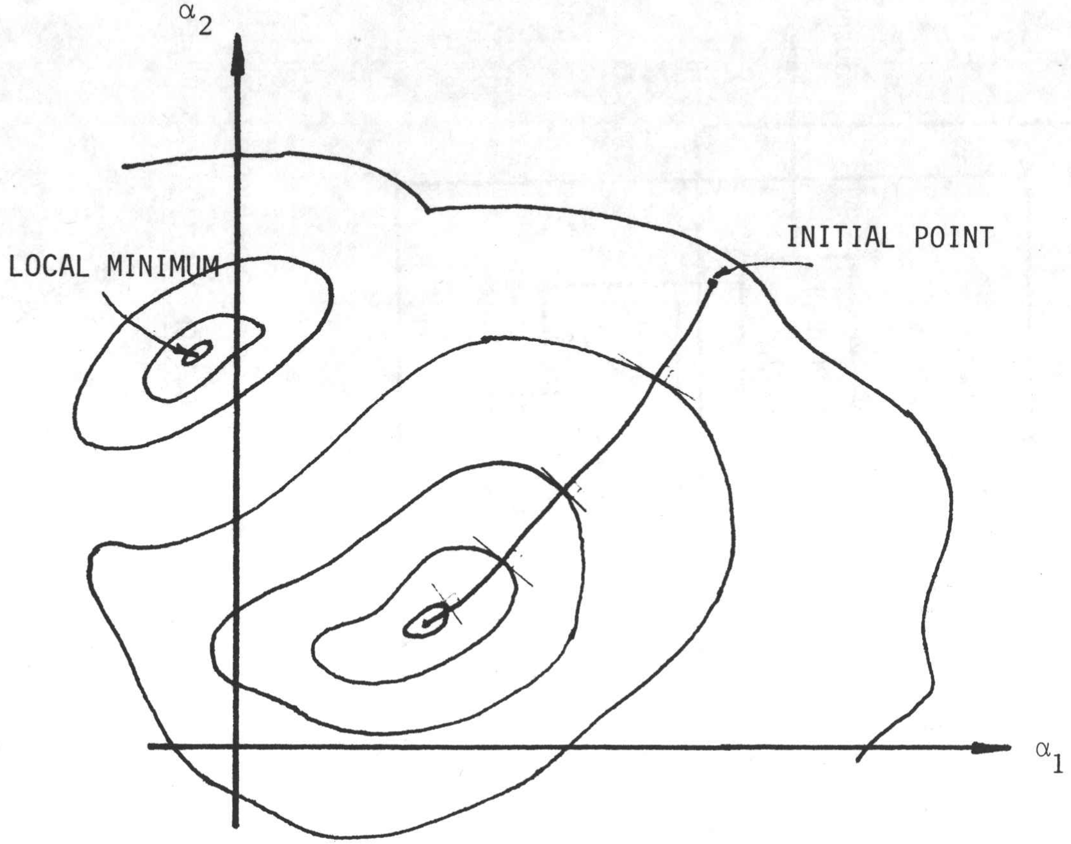


FIGURE 5.1 CONTOUR LINES AND STEEPEST DESCENT IN PARAMETER SPACE

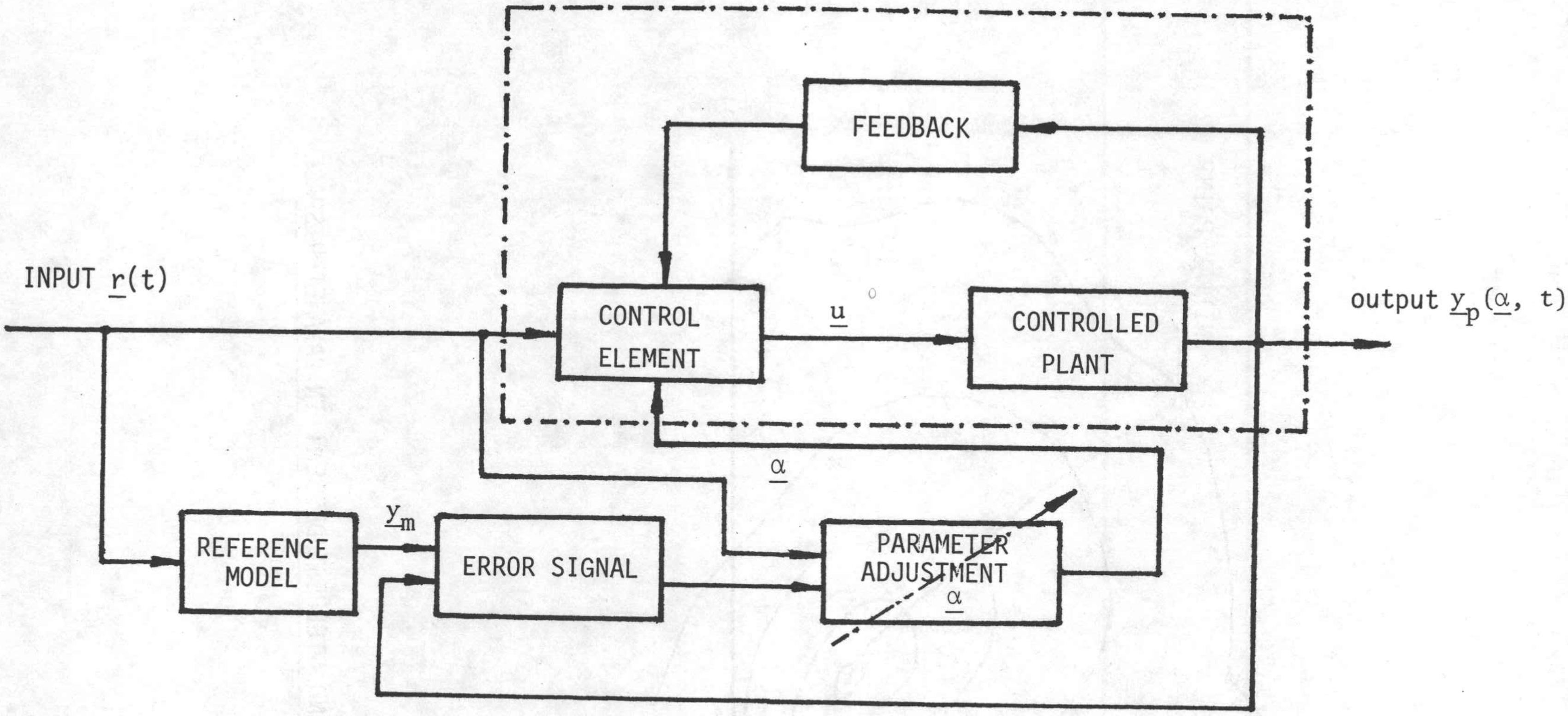


FIGURE 5.2 BLOCK DIAGRAM OF MODEL REFERENCE ADAPTIVE CONTROL SYSTEM

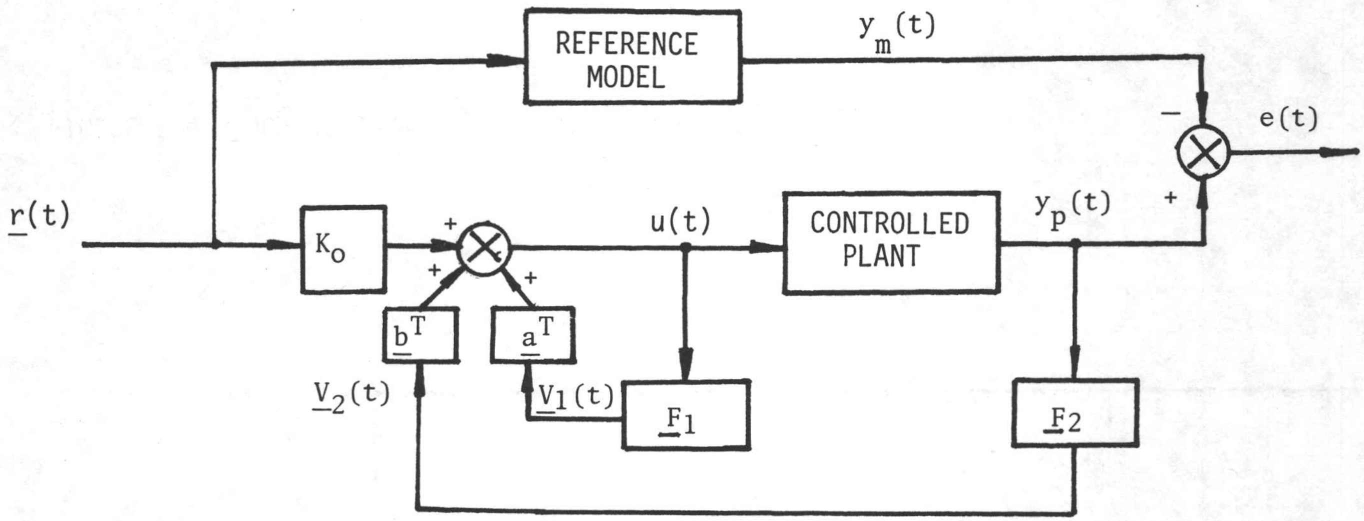


FIGURE 5.3 BLOCK DIAGRAM OF M.R.A.C. (DIRECT CONTROL)

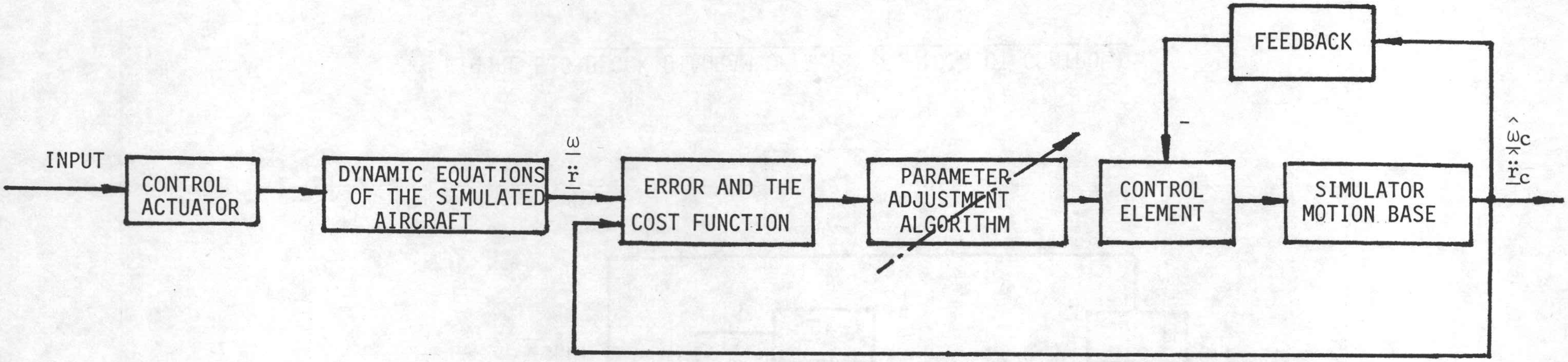


FIGURE 5.4 BLOCK DIAGRAM OF THE ADAPTIVE WASHOUT FILTER

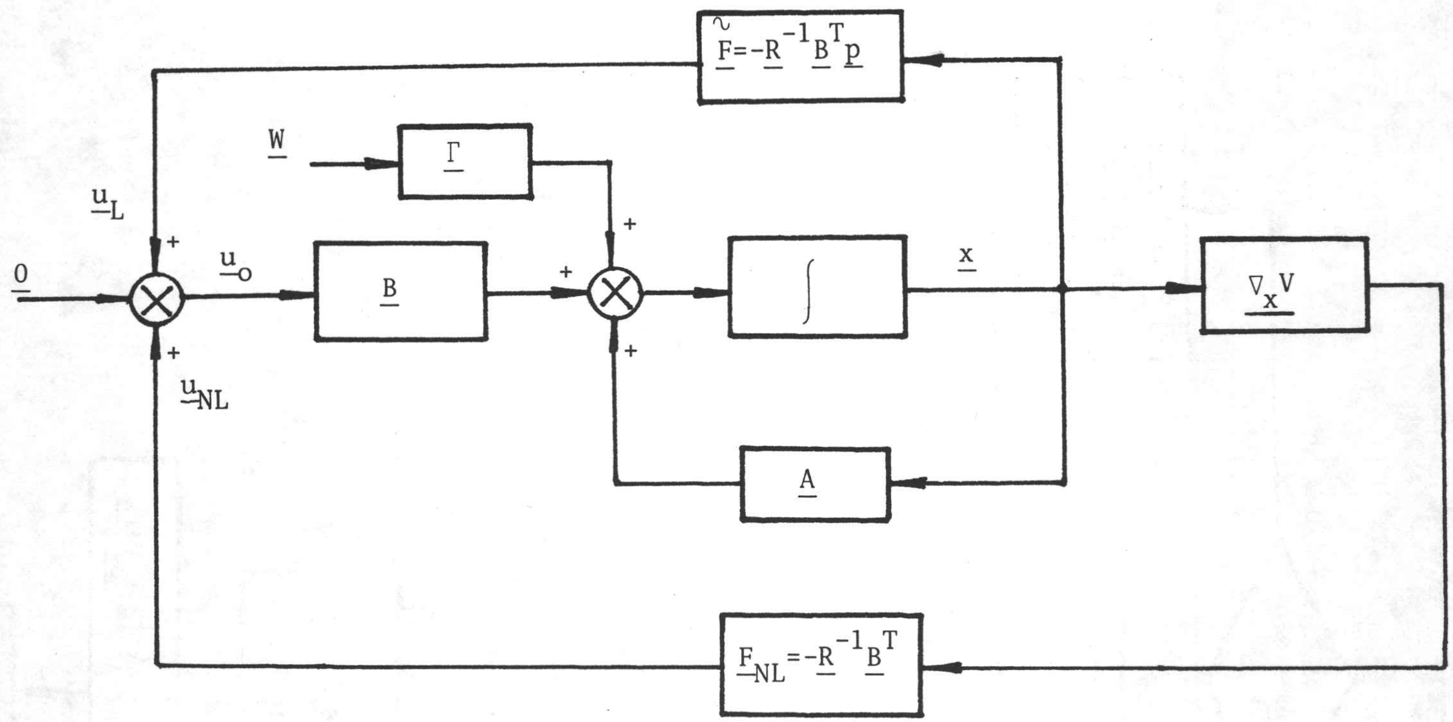


FIGURE 6.1 NONLINEAR OPTIMAL CONTROL DIAGRAM

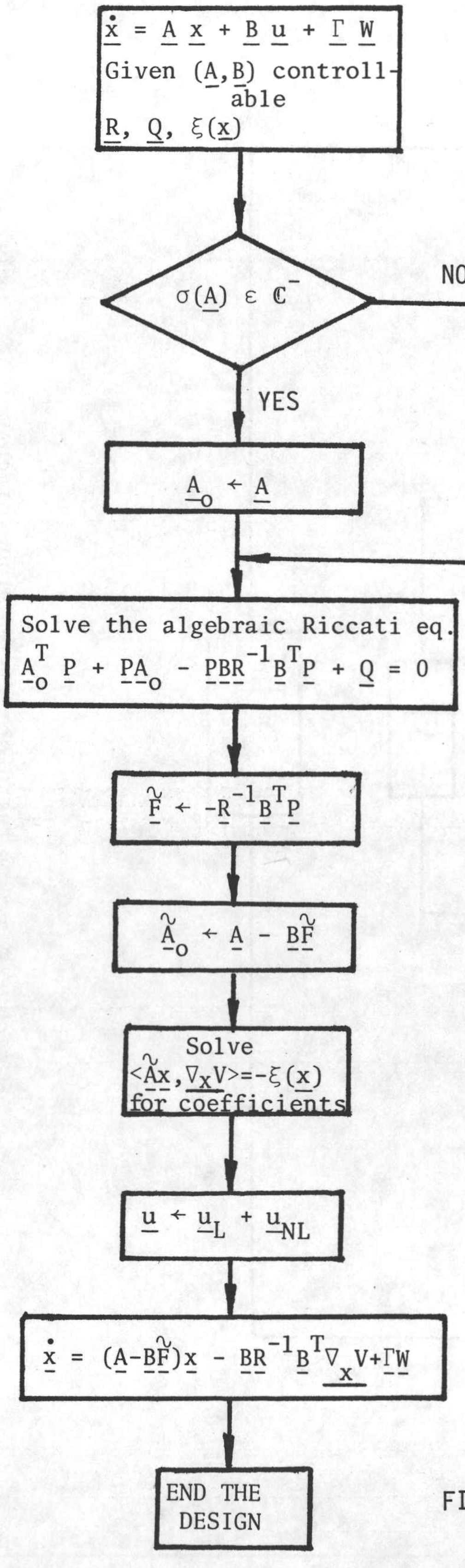


FIGURE 6.2 DESIGN PROCEDRURES FOR N.O.C.

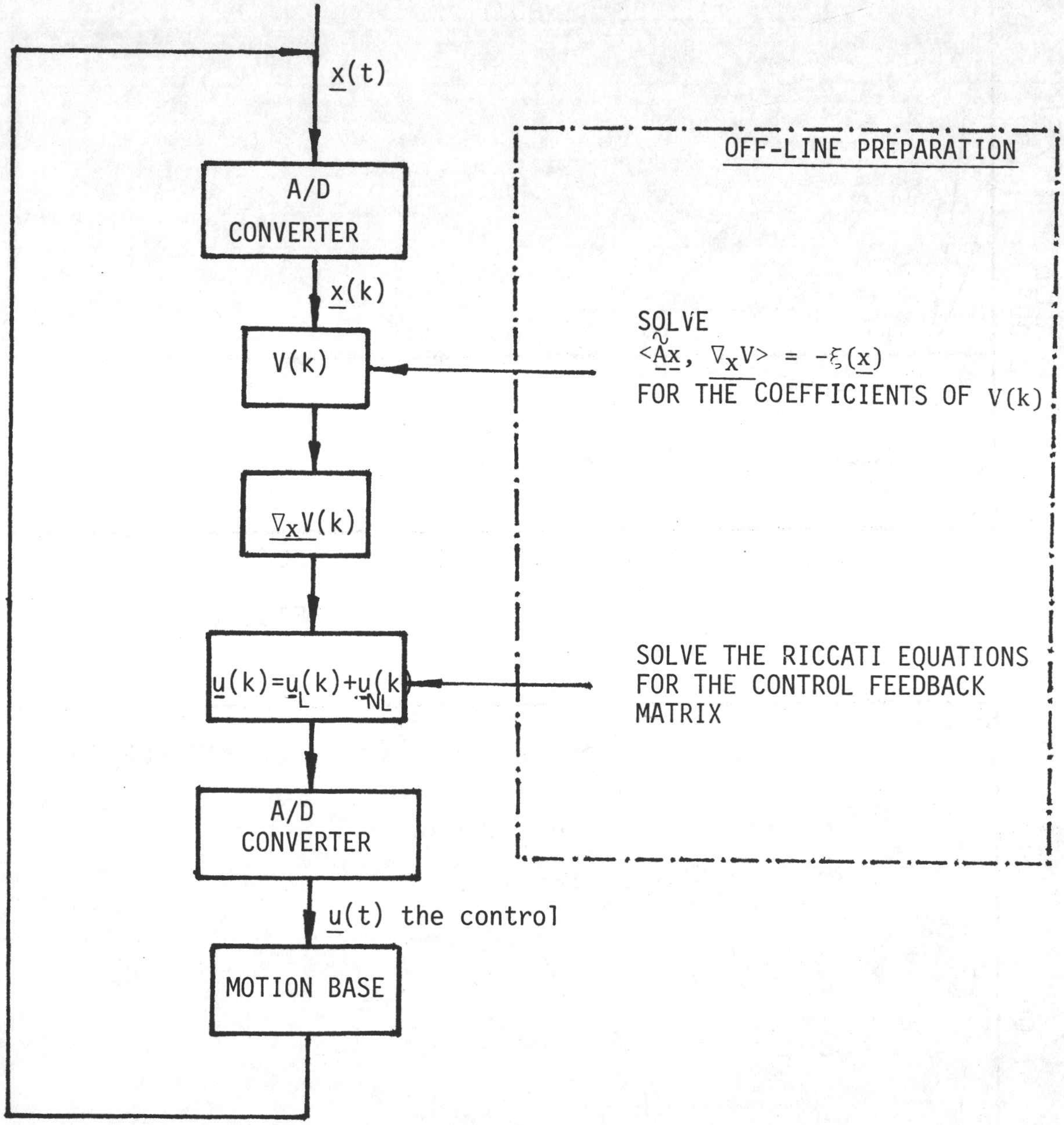


FIGURE 6.3 THE FLOW CHART FOR REAL TIME DIGITAL CONTROL

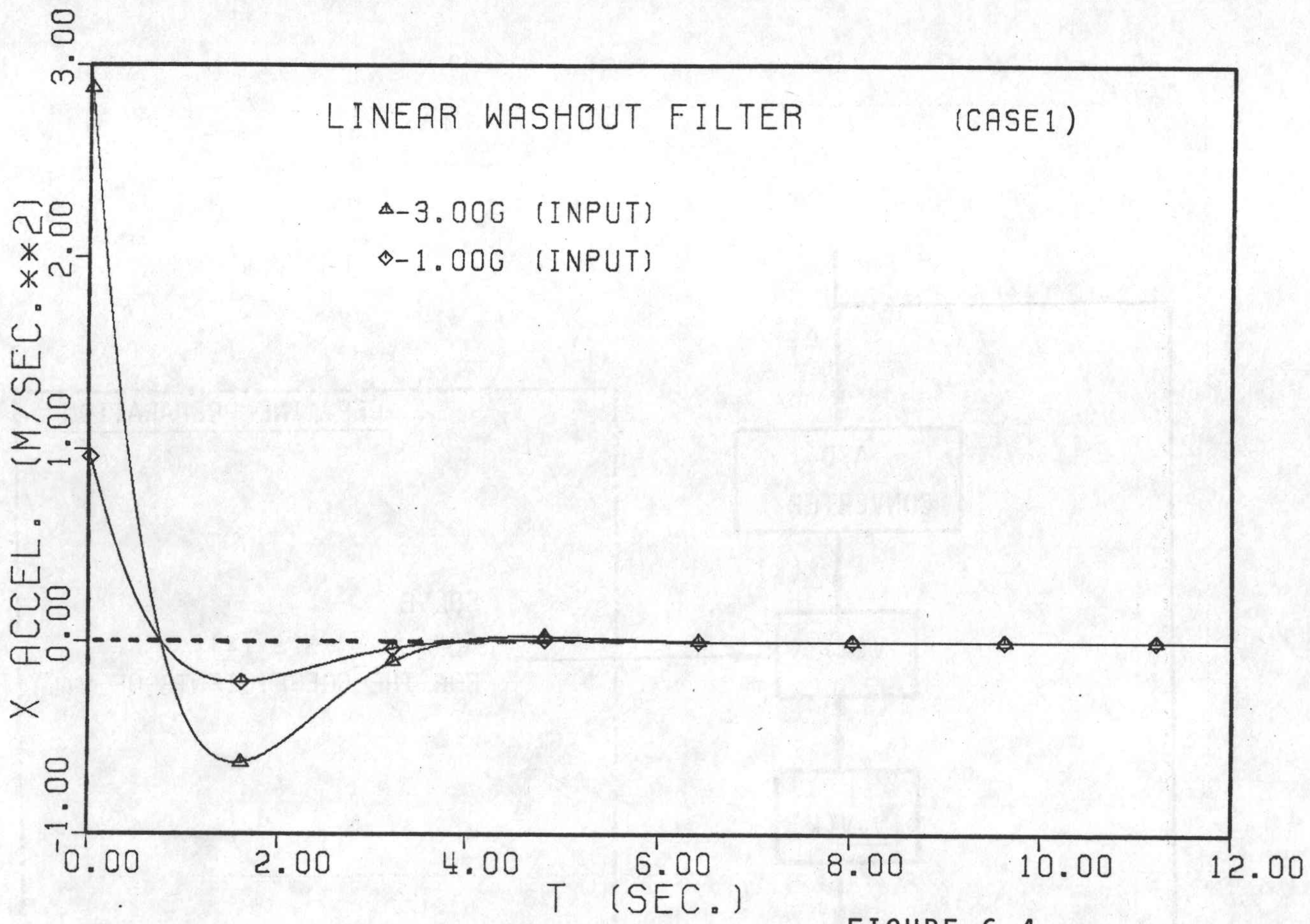


FIGURE 6.4

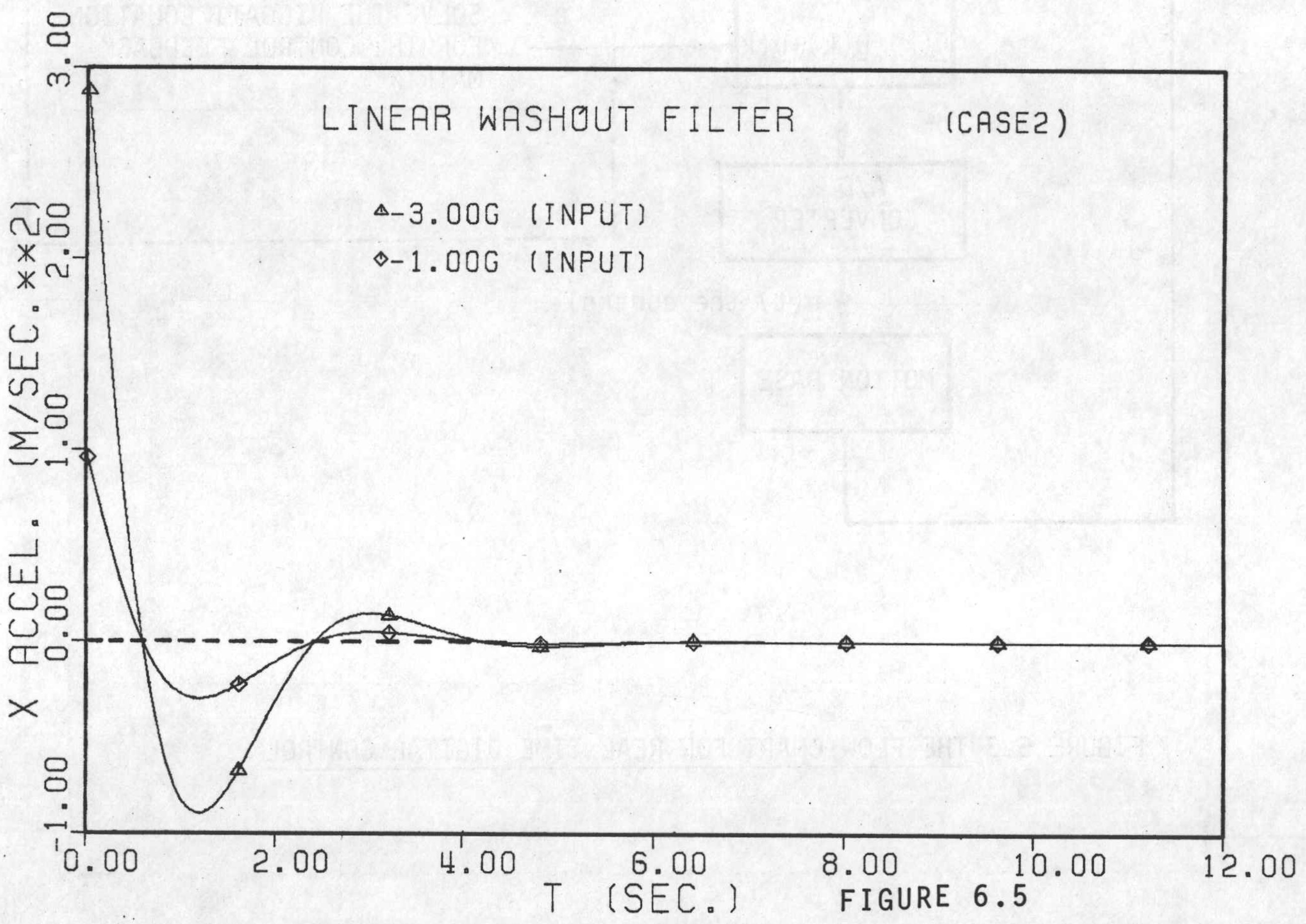


FIGURE 6.5

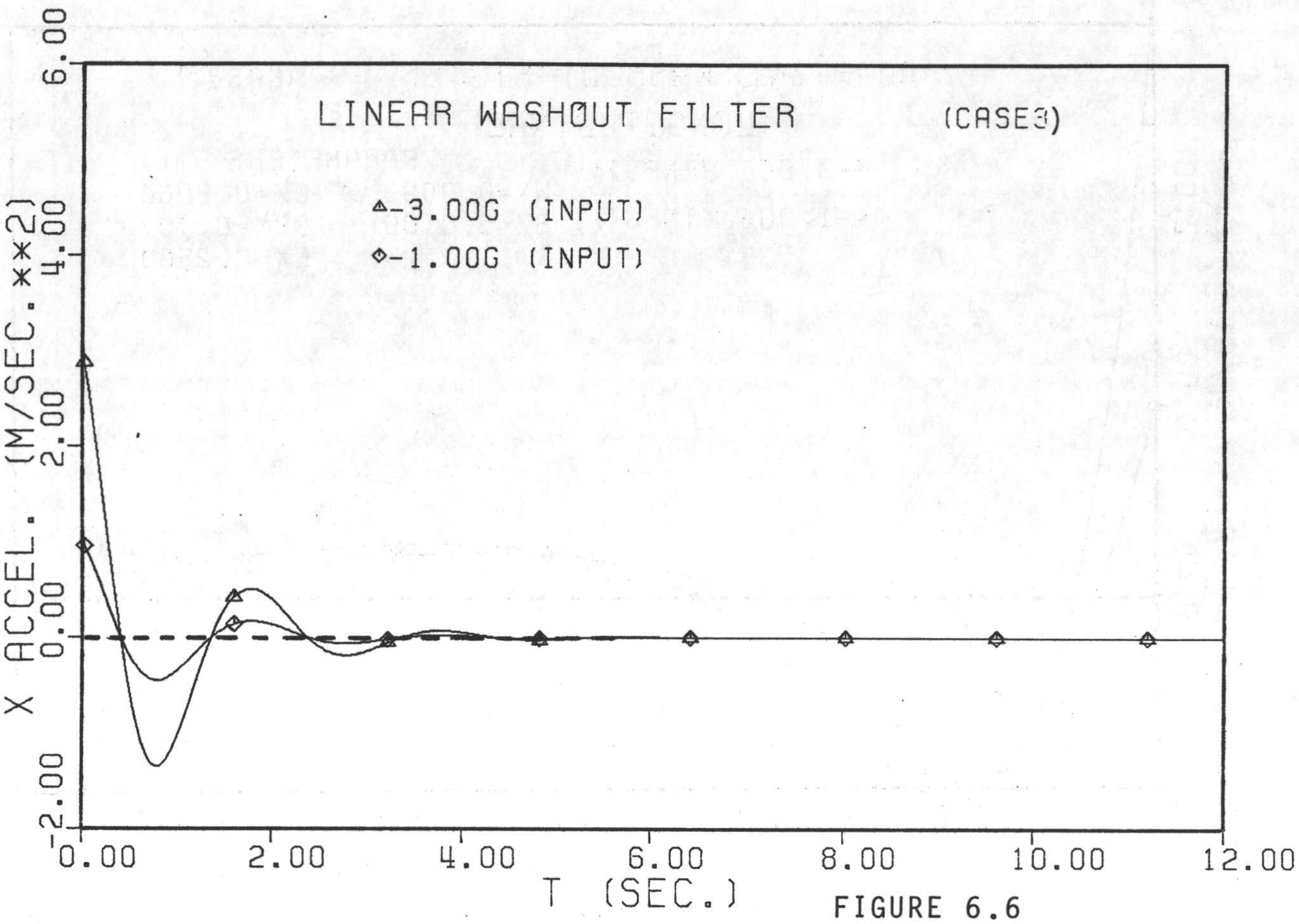


FIGURE 6.6

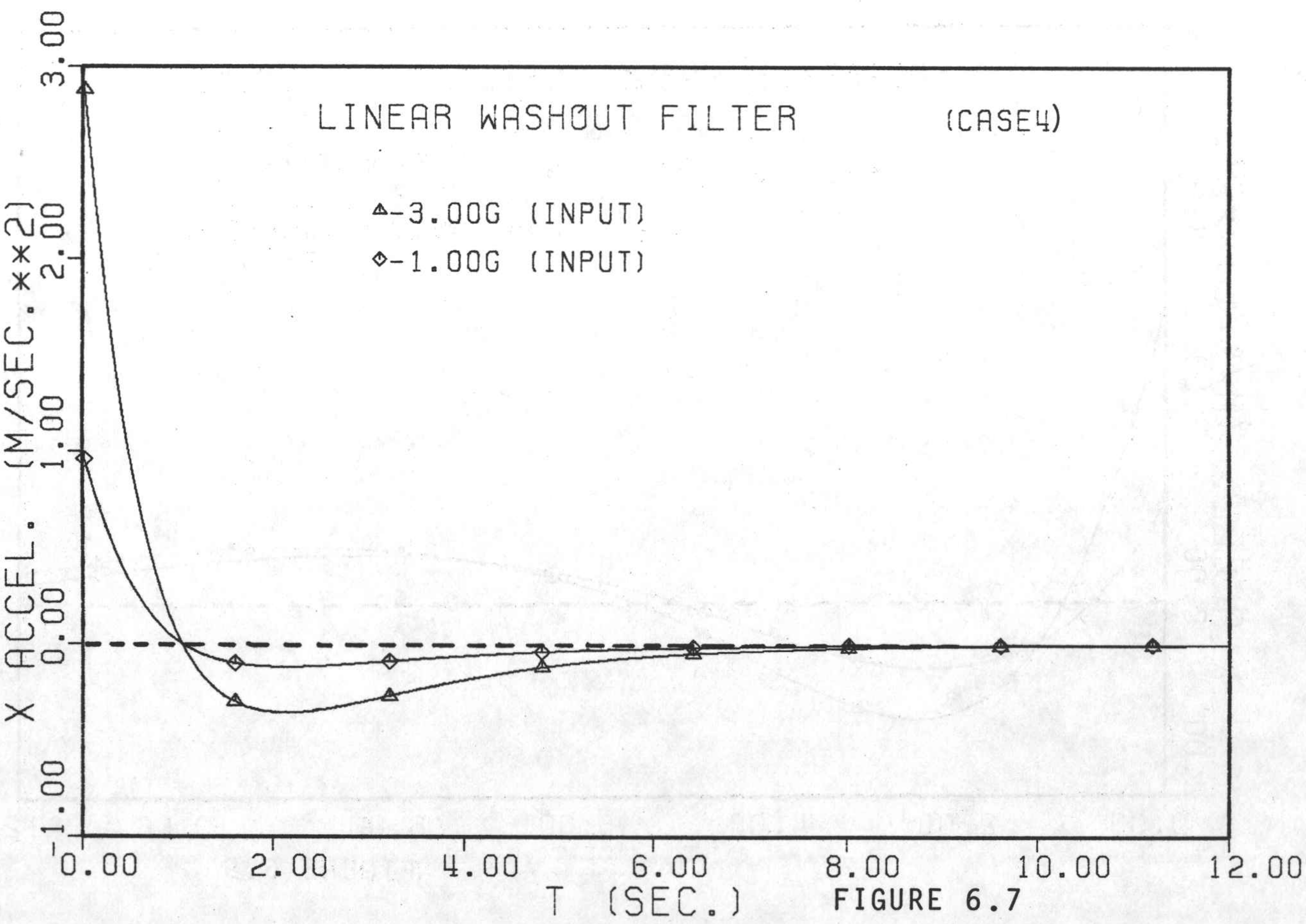


FIGURE 6.7

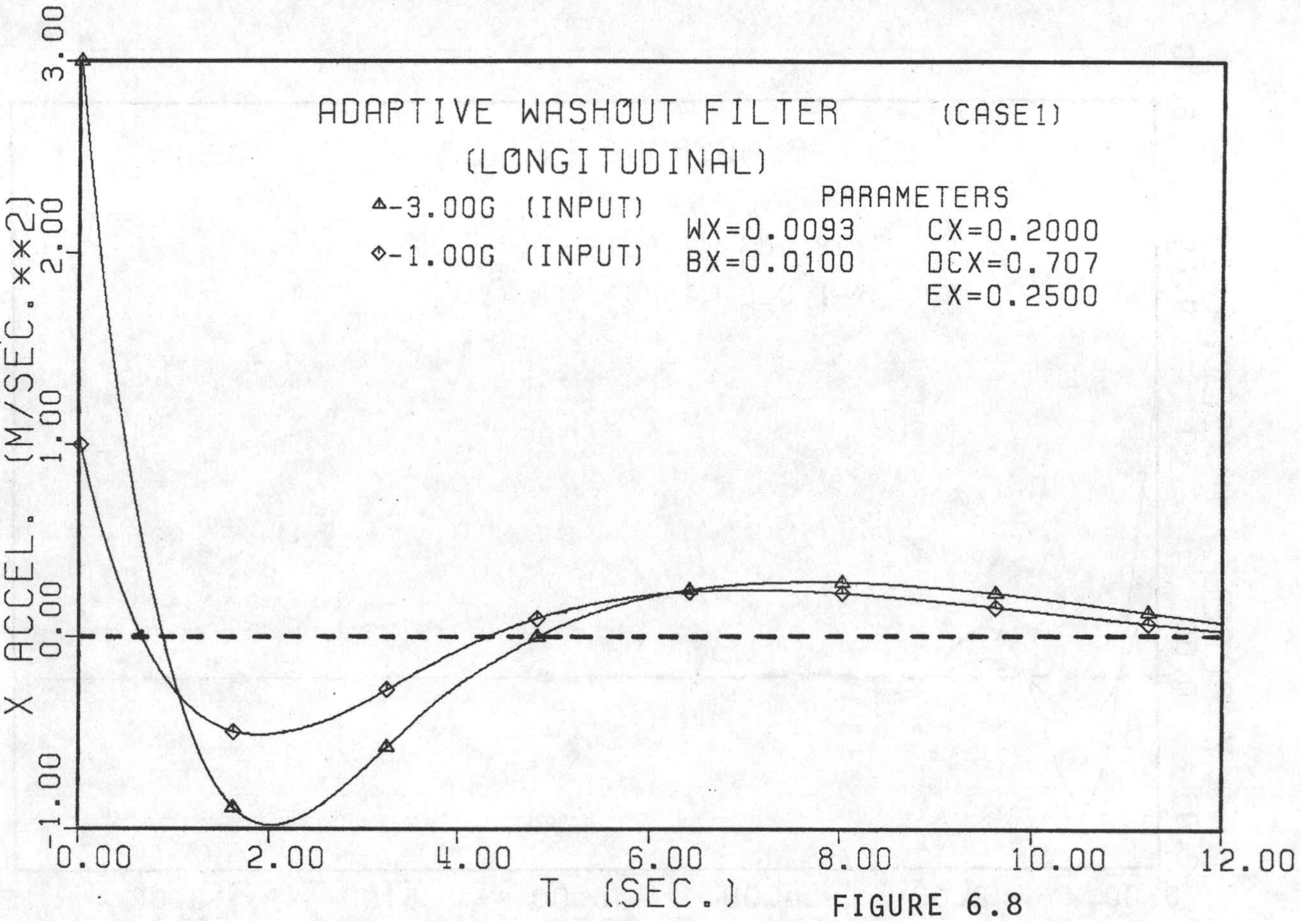


FIGURE 6.8

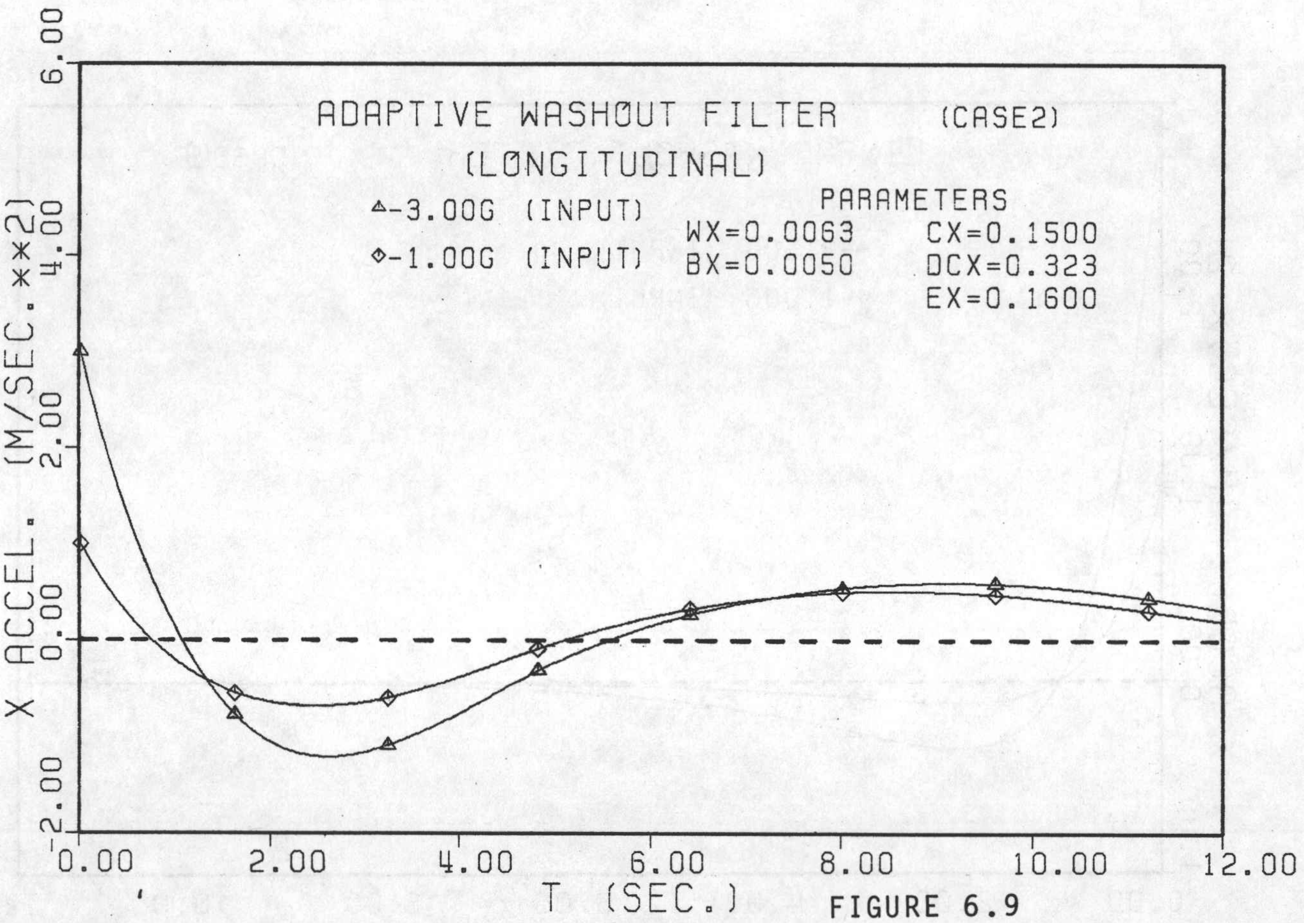


FIGURE 6.9

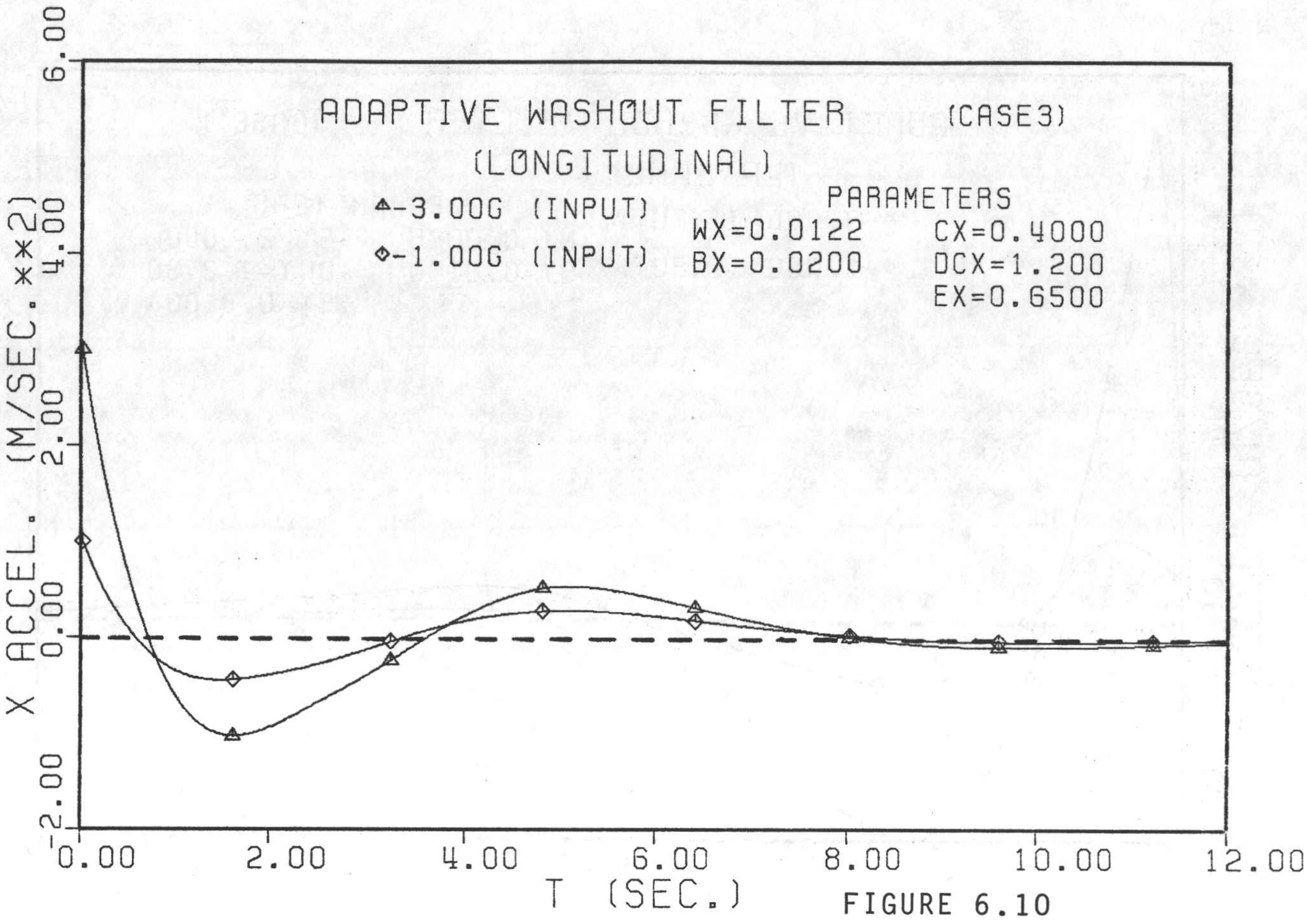


FIGURE 6.10

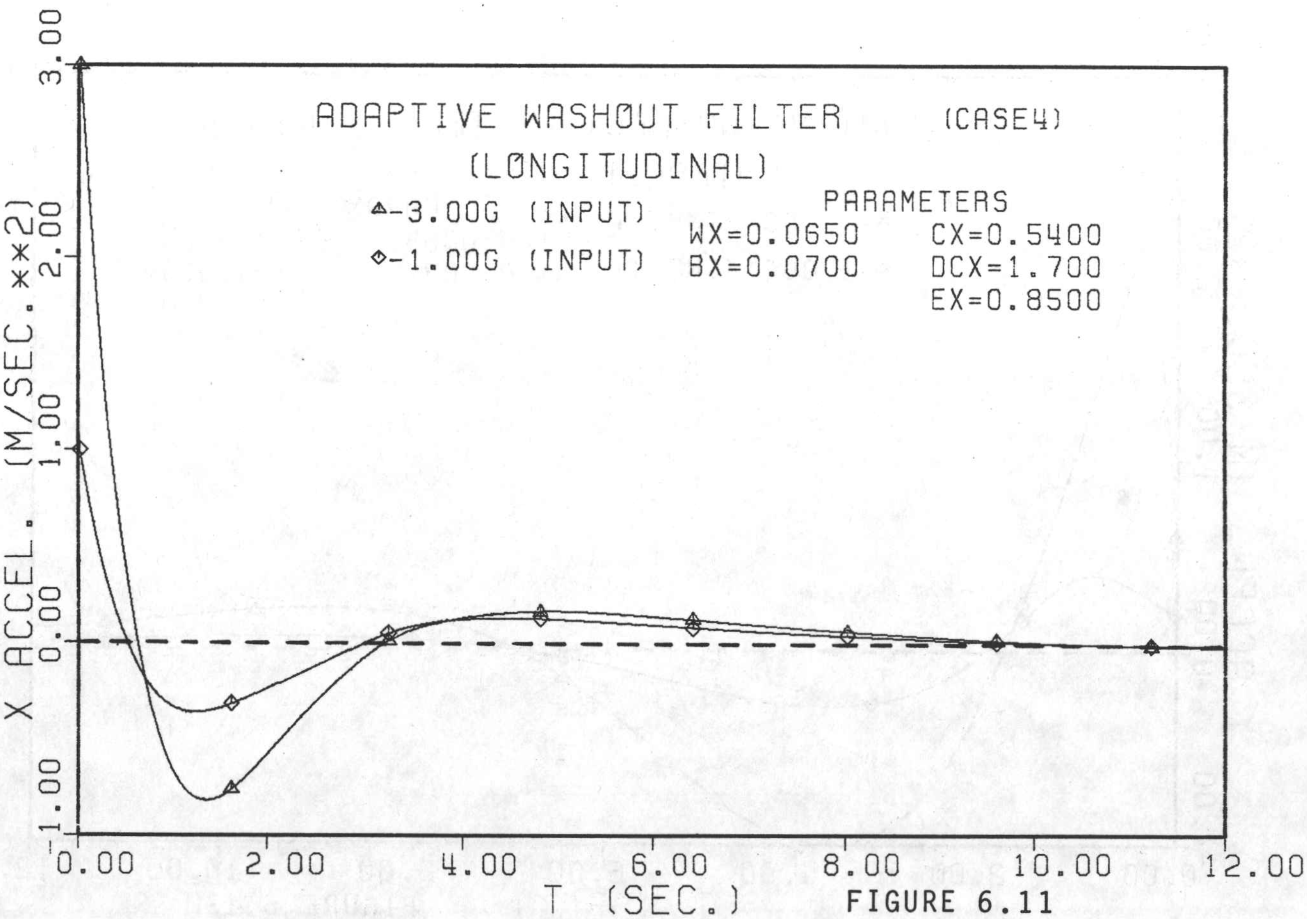


FIGURE 6.11

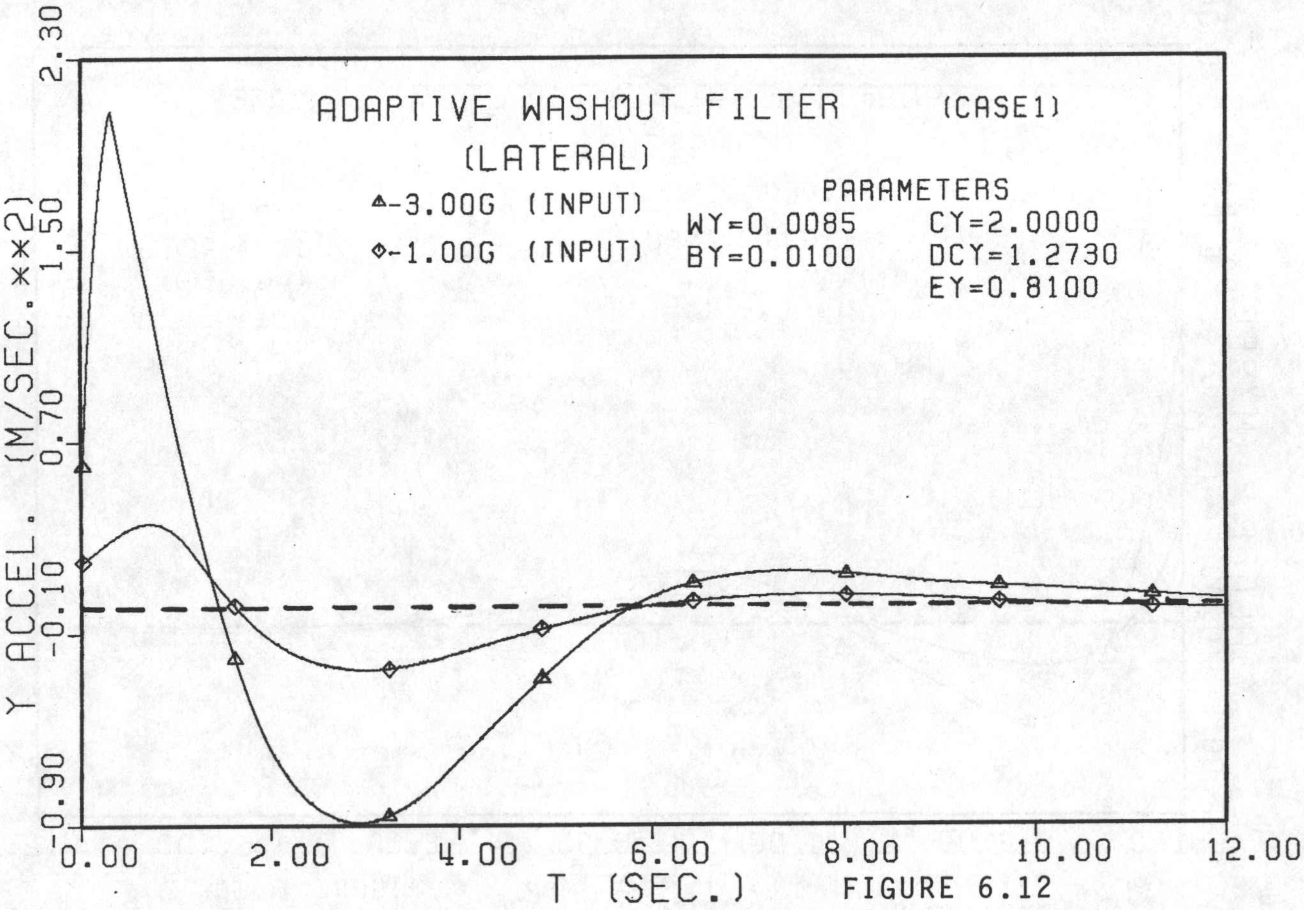


FIGURE 6.12

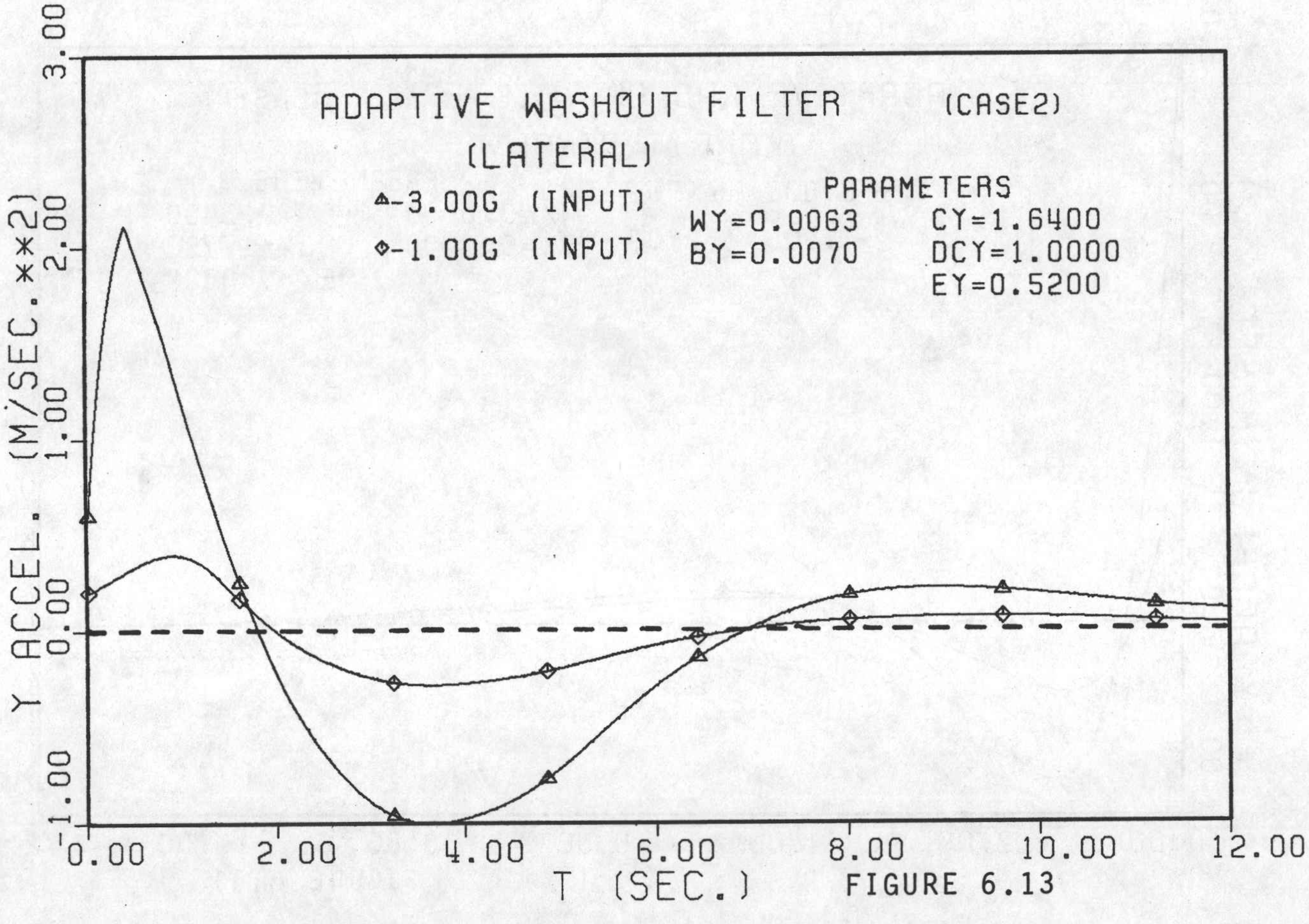


FIGURE 6.13

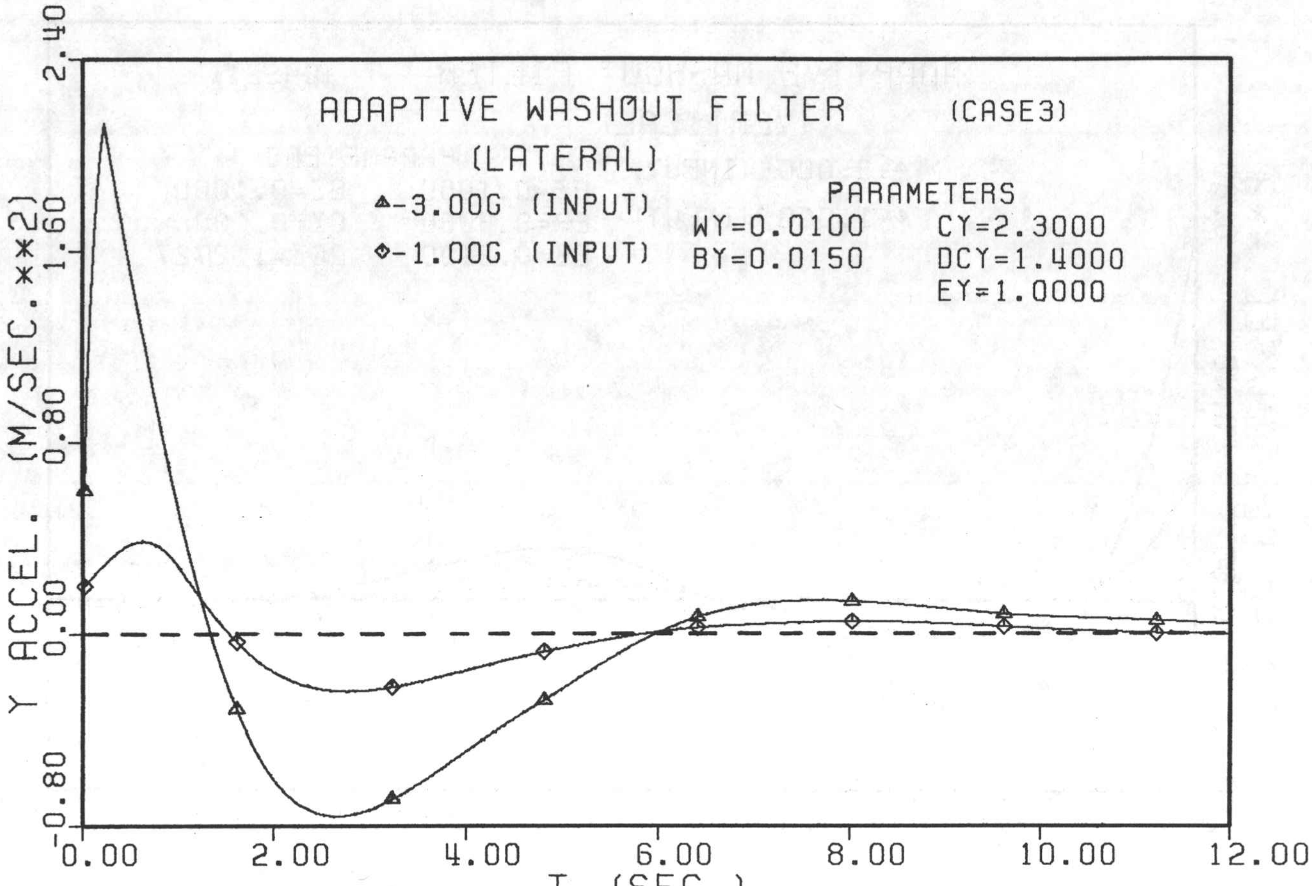


FIGURE 6.14

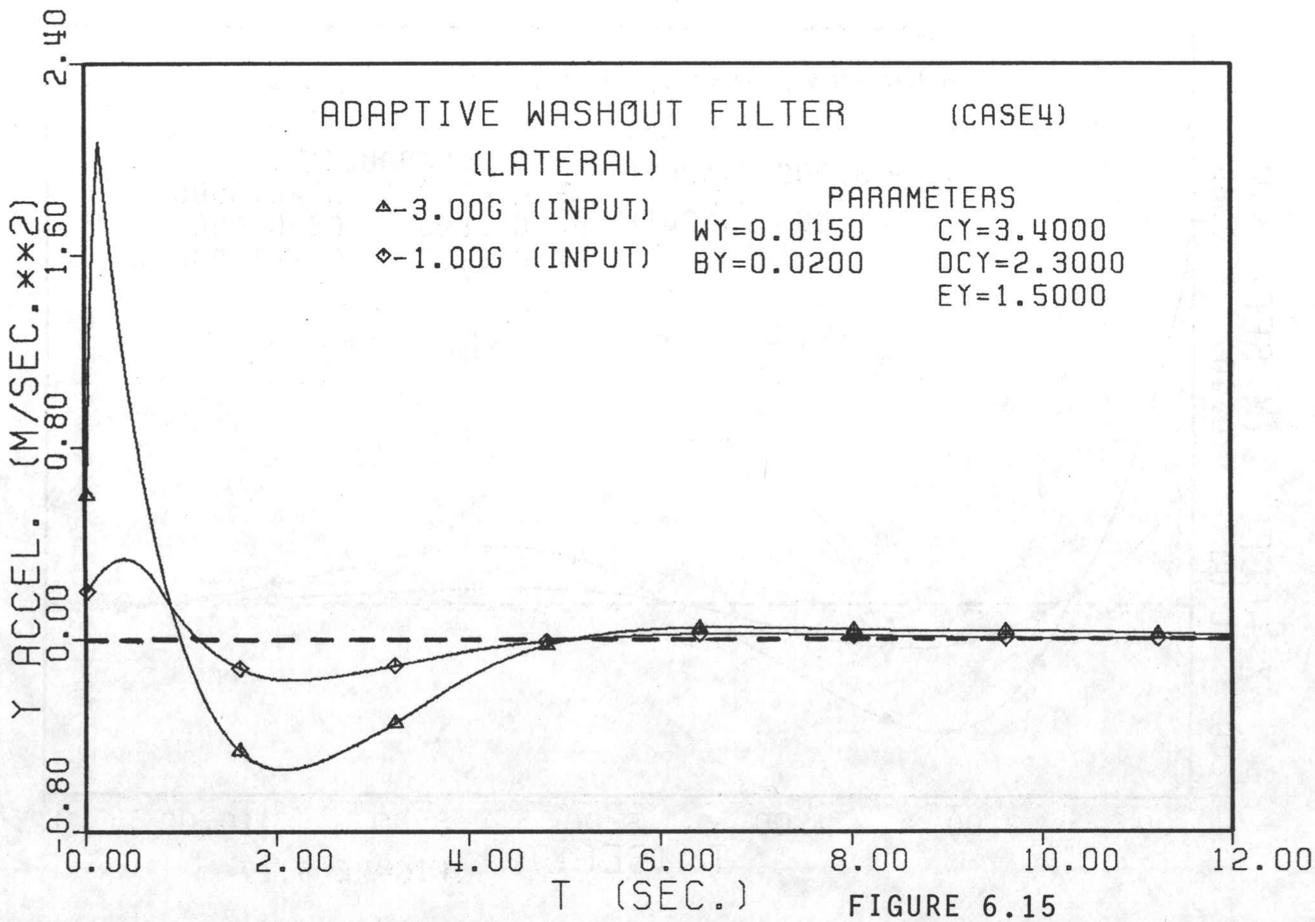


FIGURE 6.15

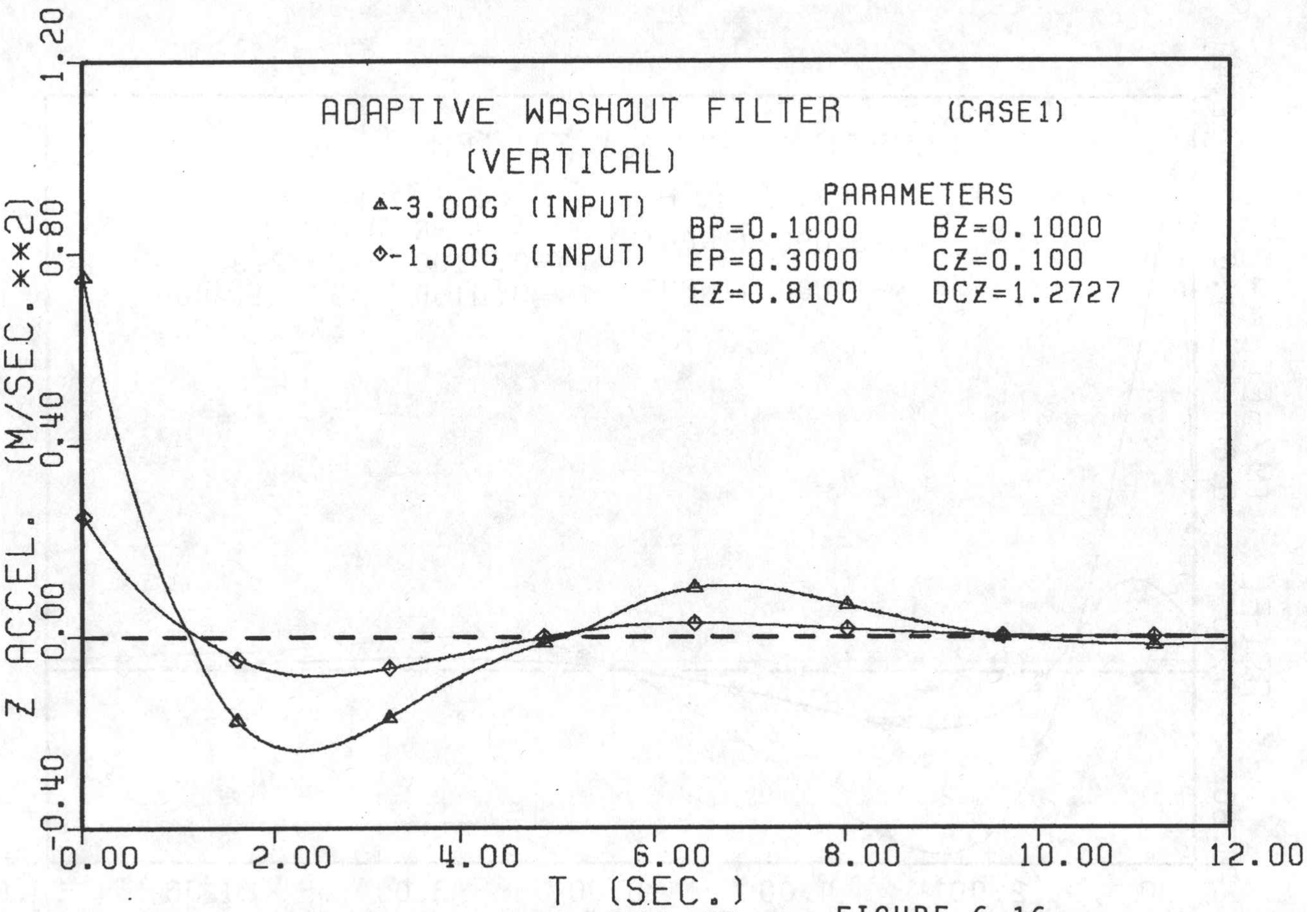


FIGURE 6.16

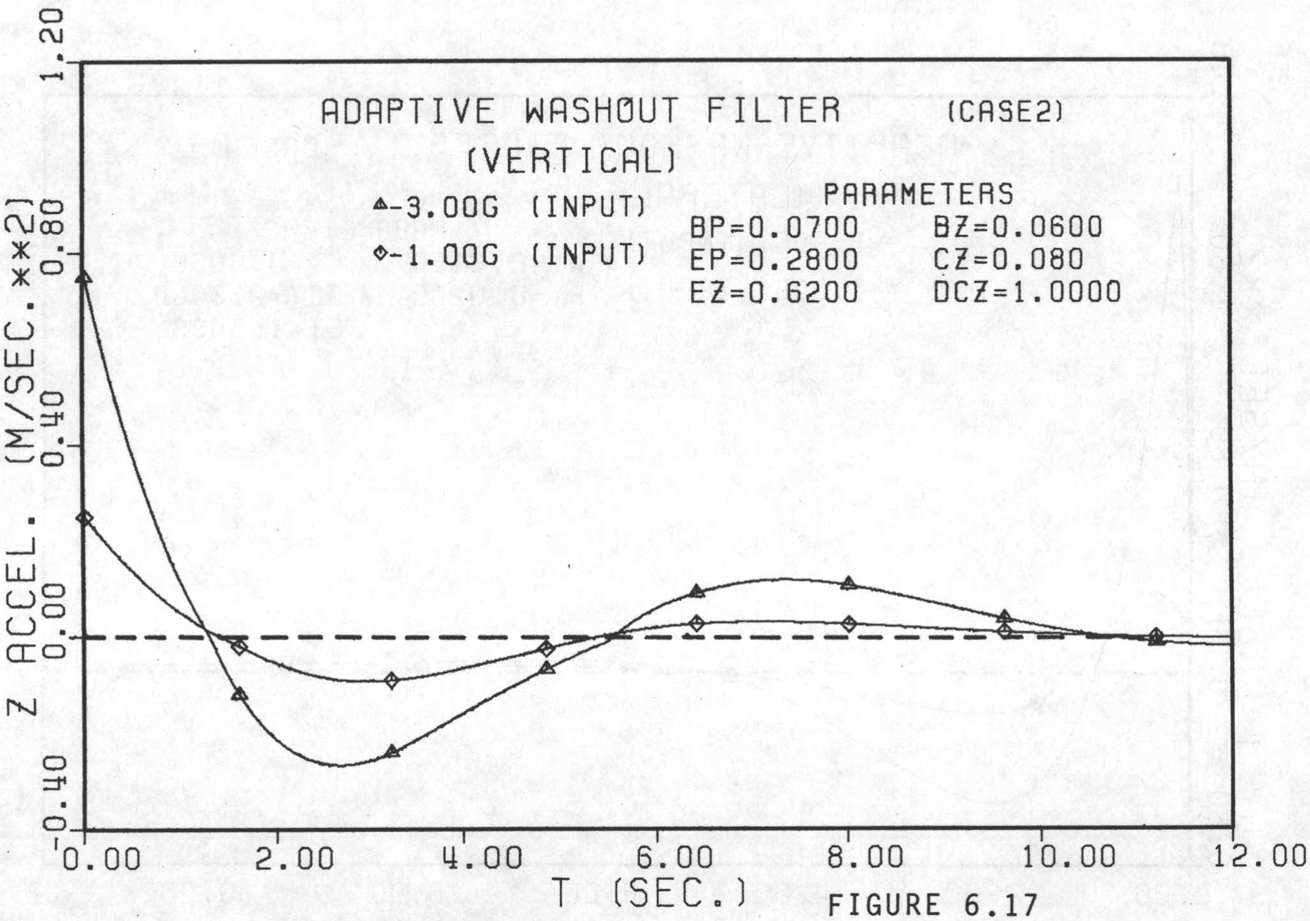


FIGURE 6.17

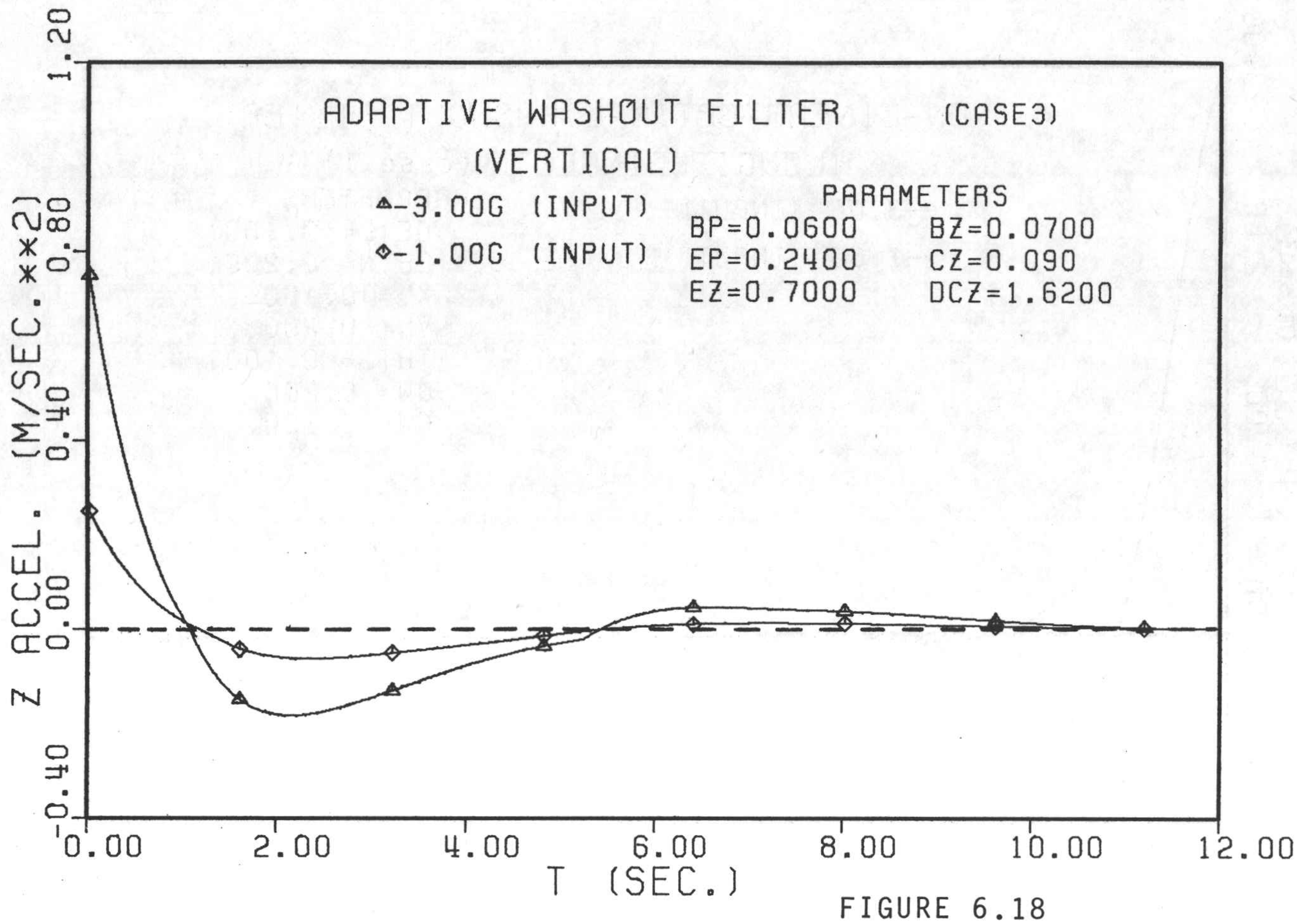


FIGURE 6.18

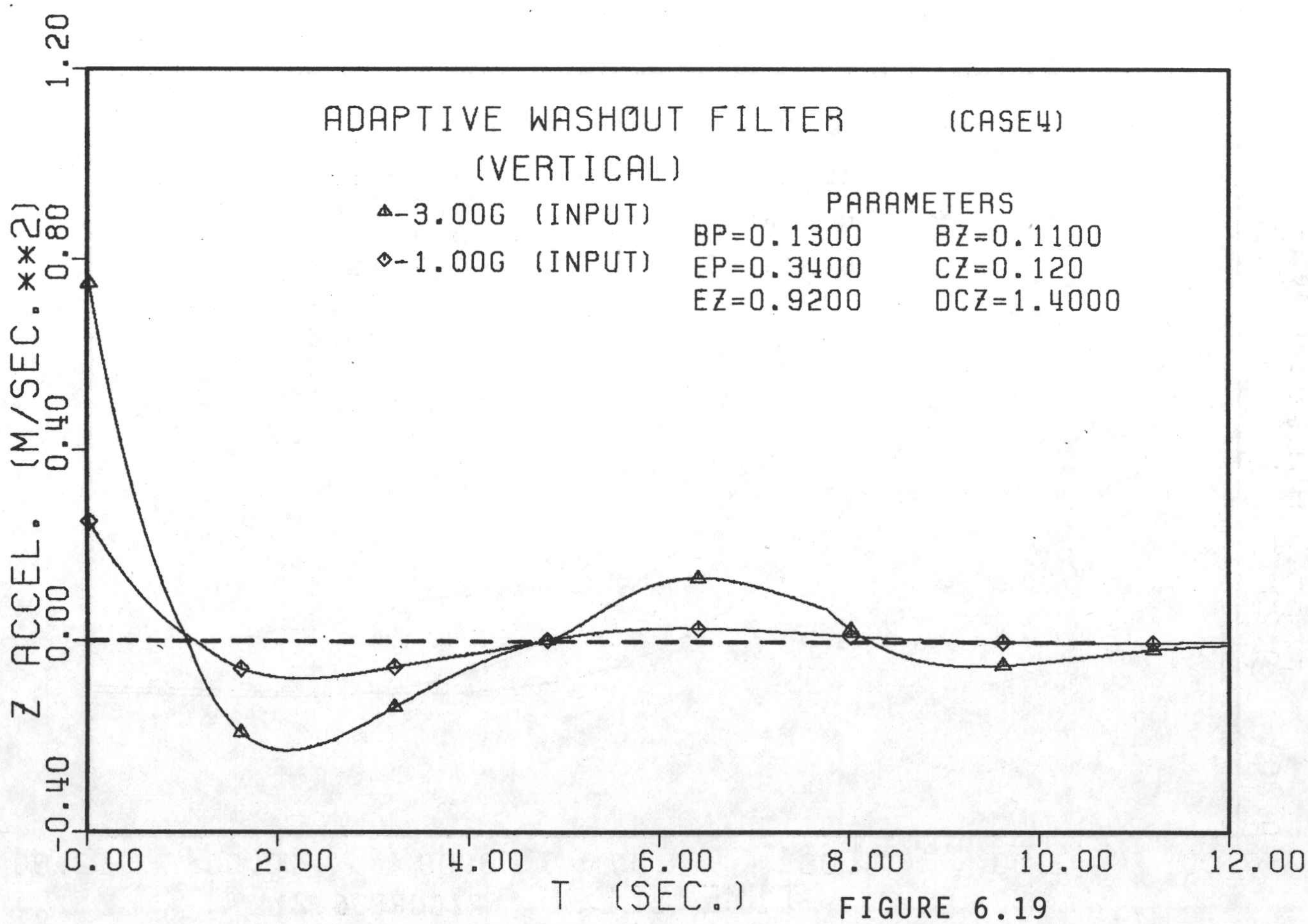


FIGURE 6.19

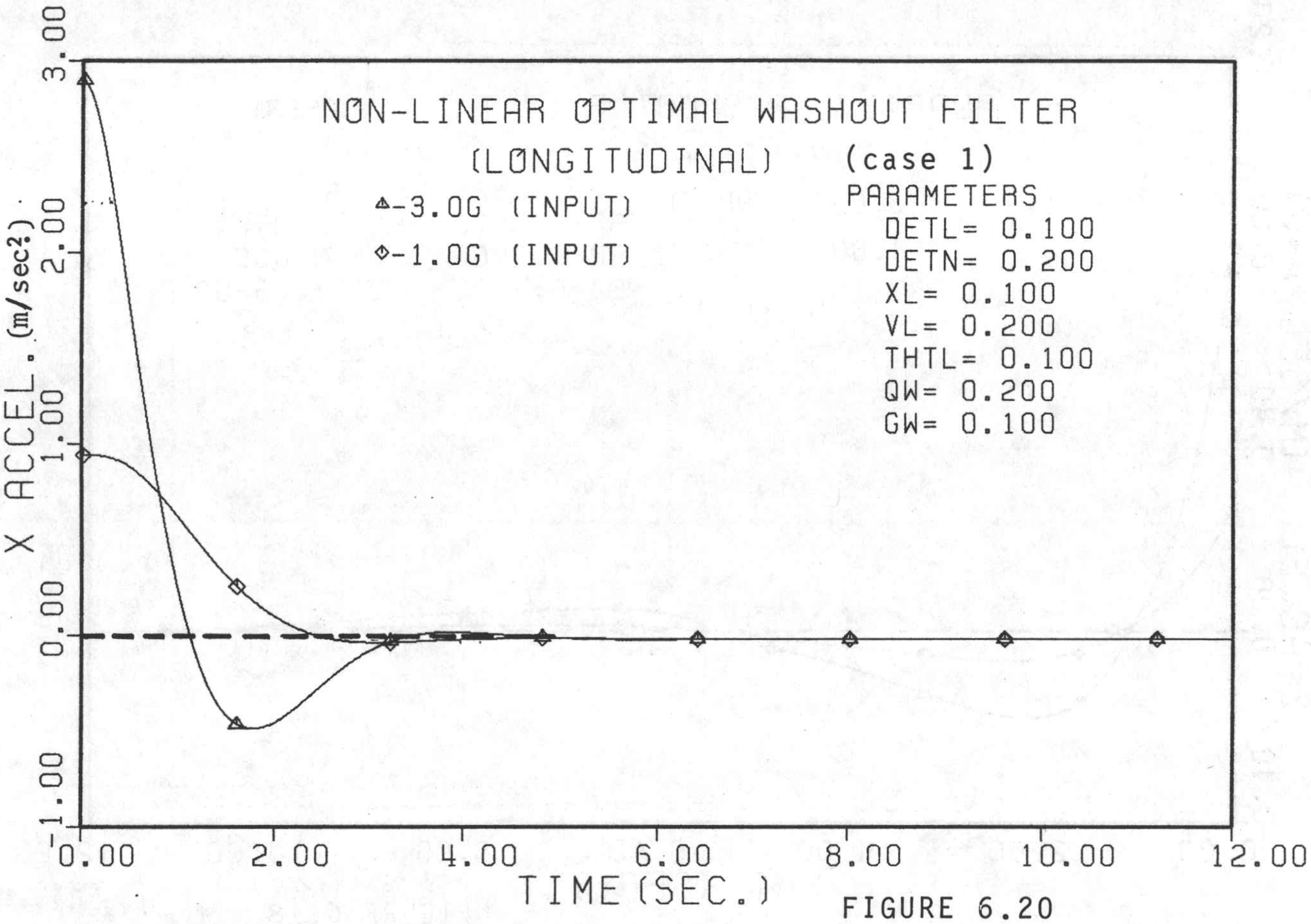


FIGURE 6.20

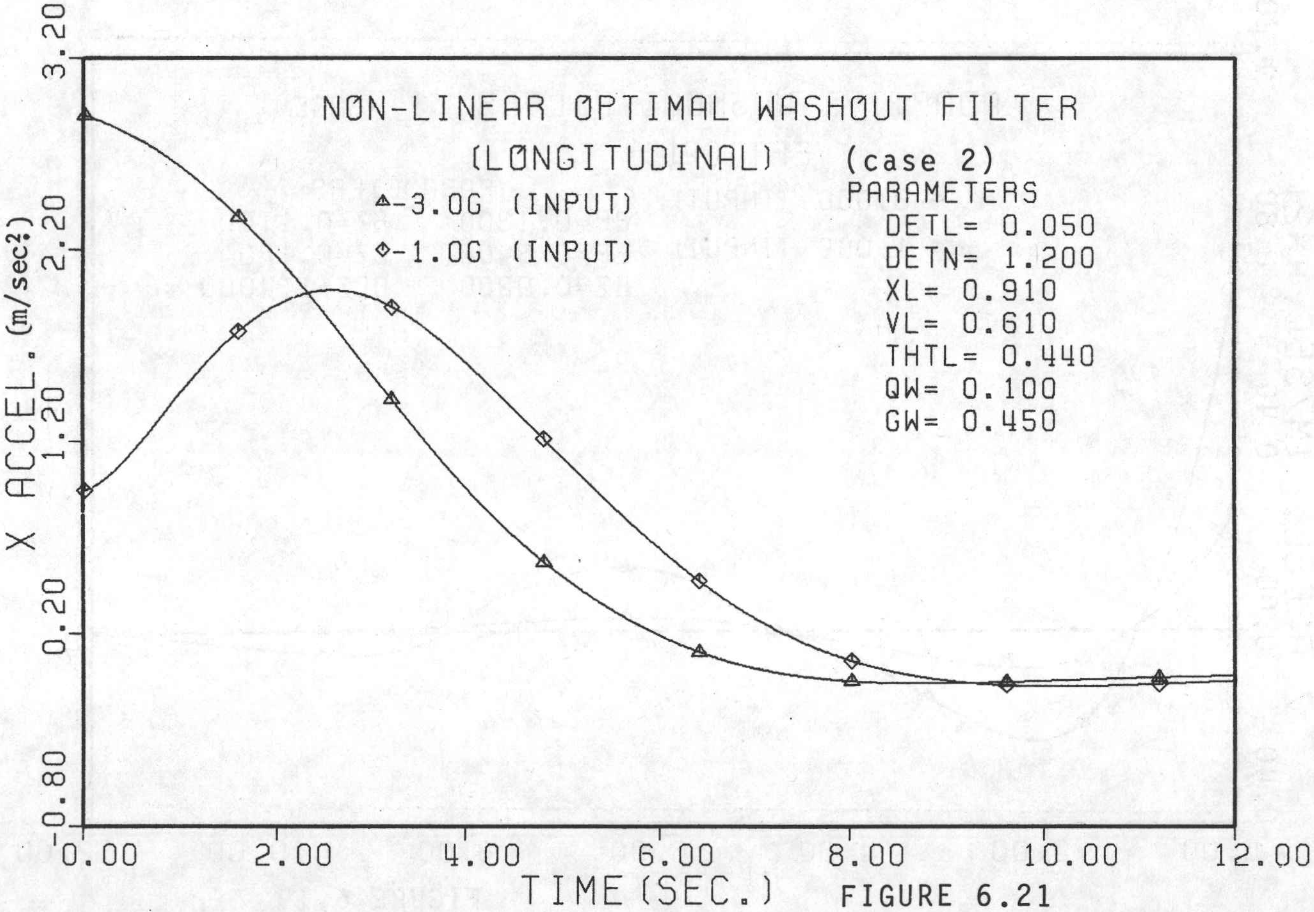


FIGURE 6.21

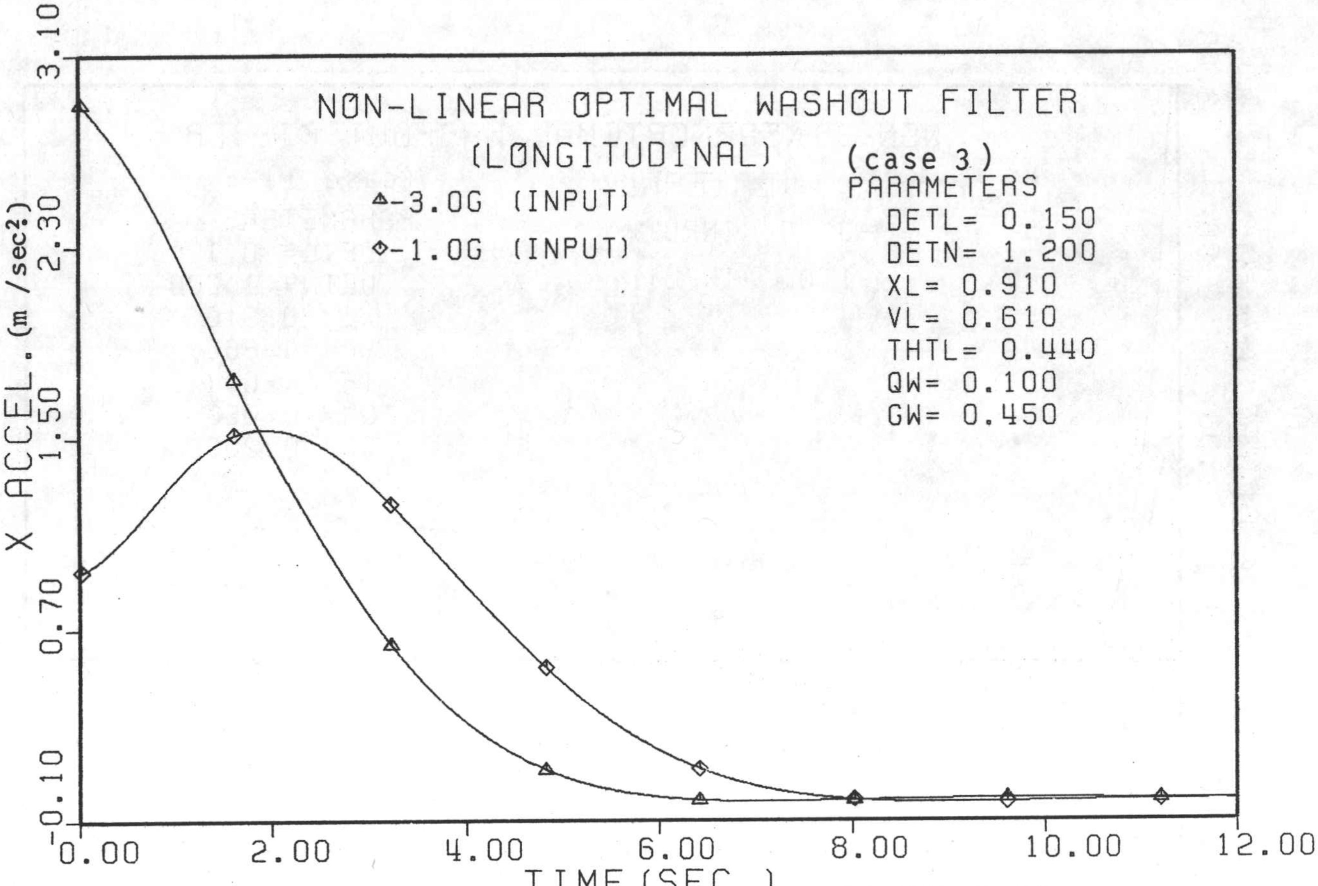


FIGURE 6.22

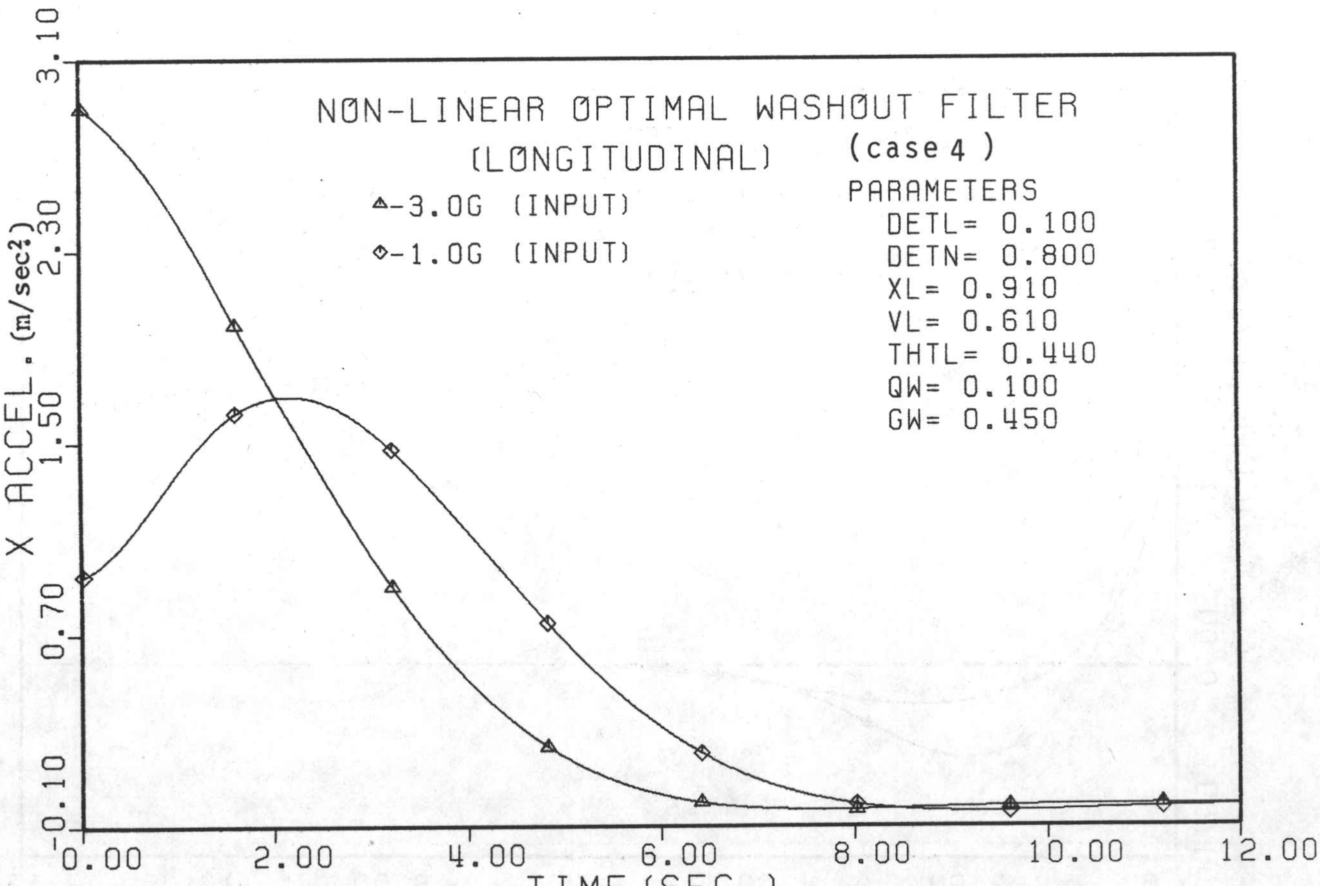


FIGURE 6.23

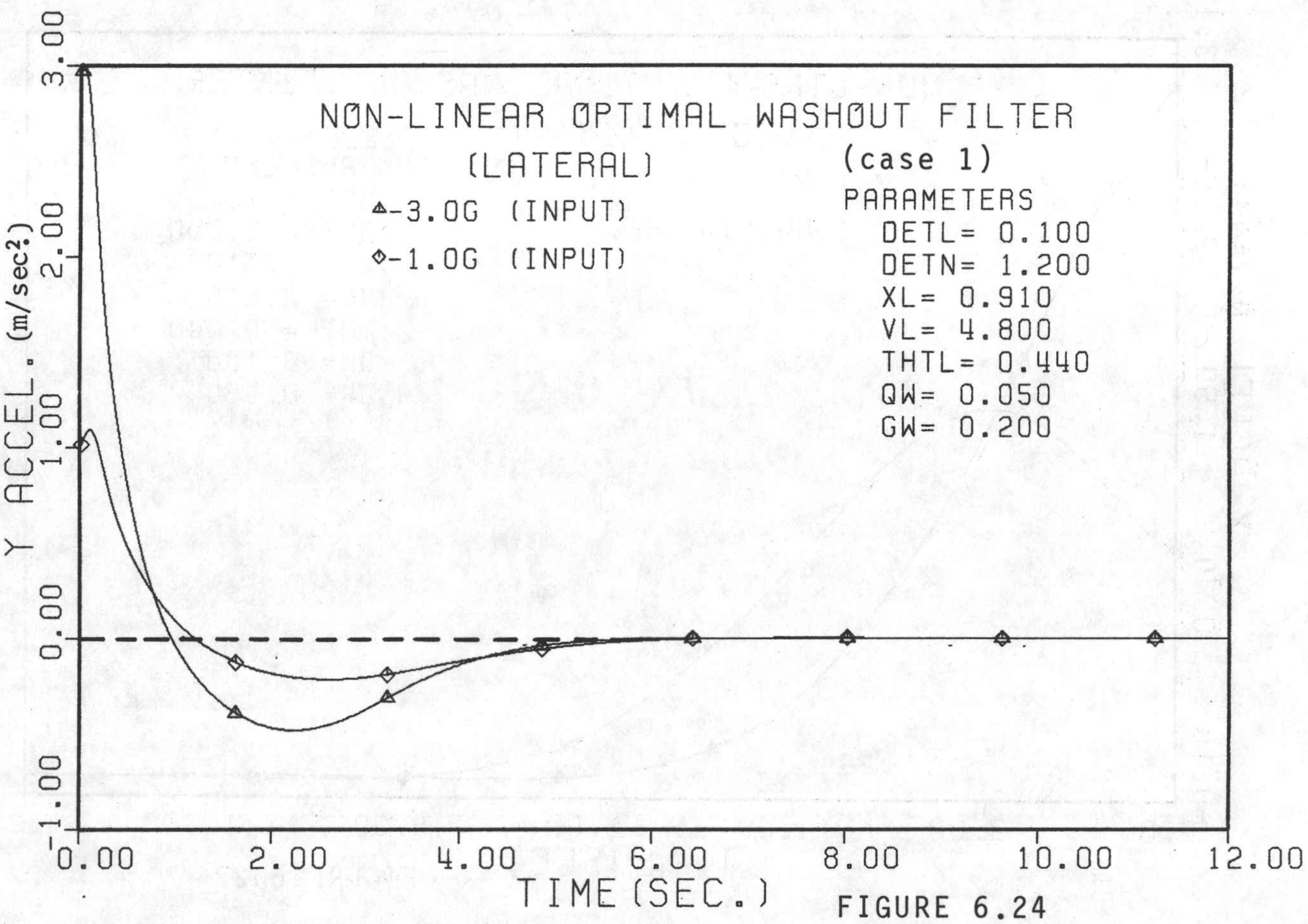


FIGURE 6.24

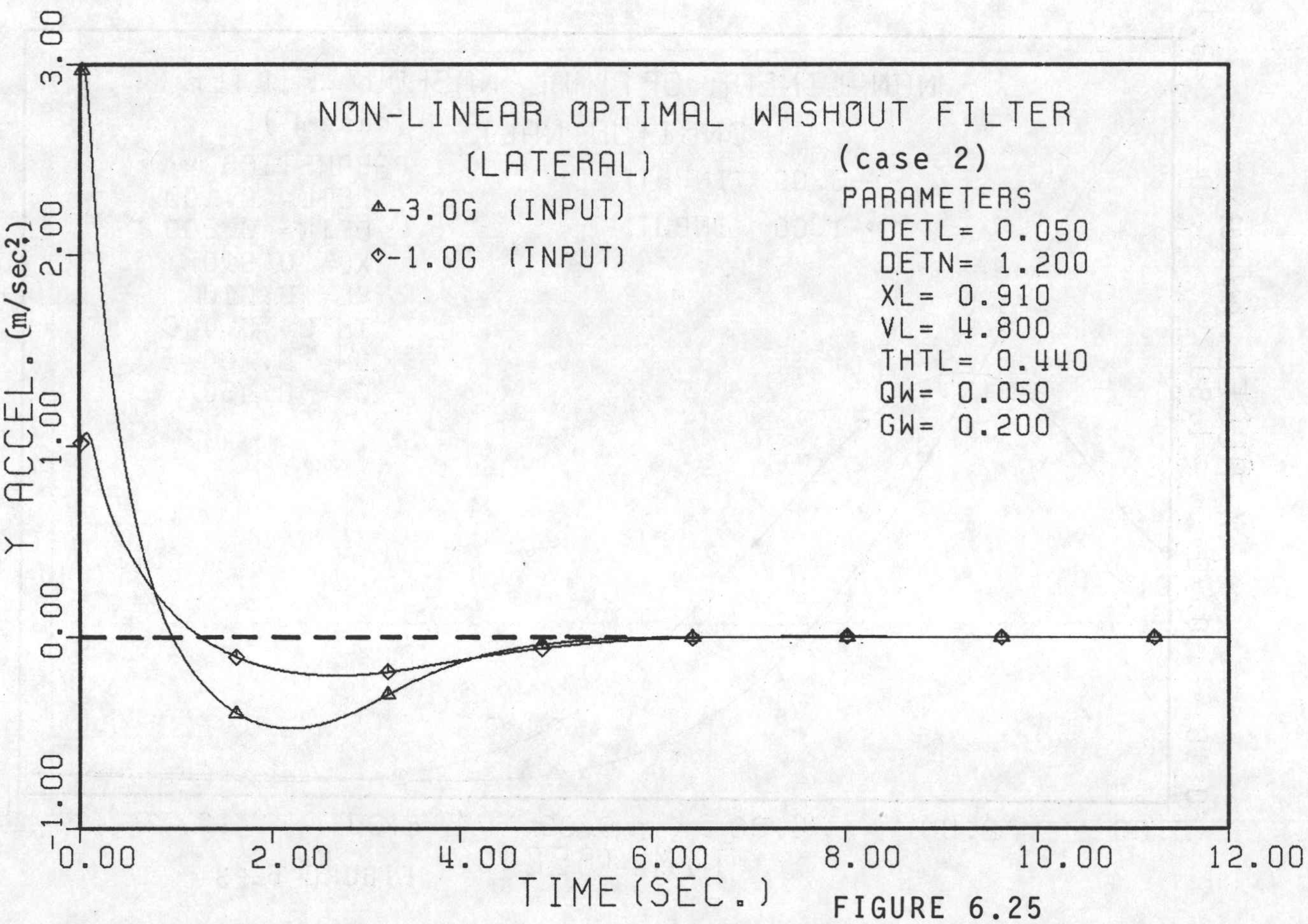


FIGURE 6.25

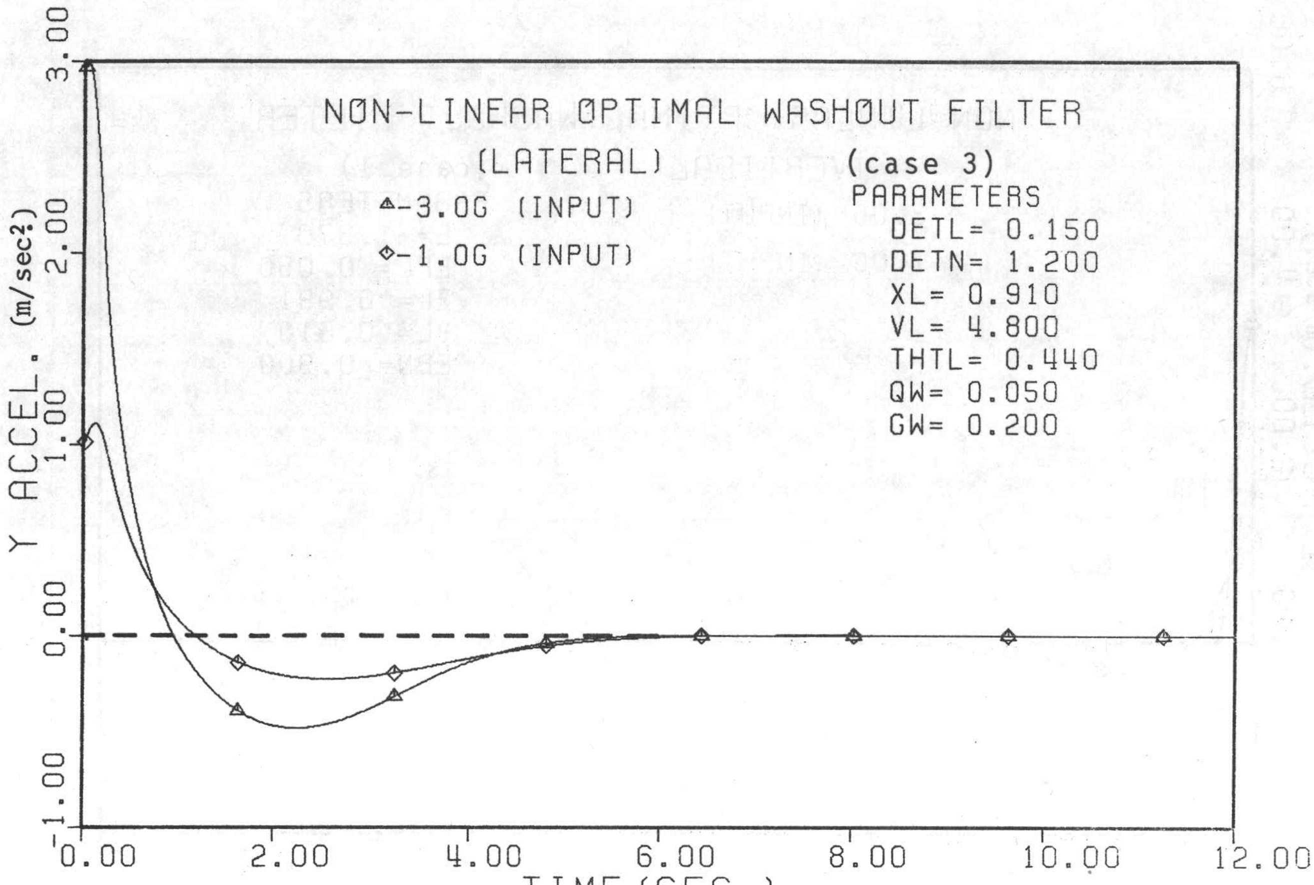


FIGURE 6.26

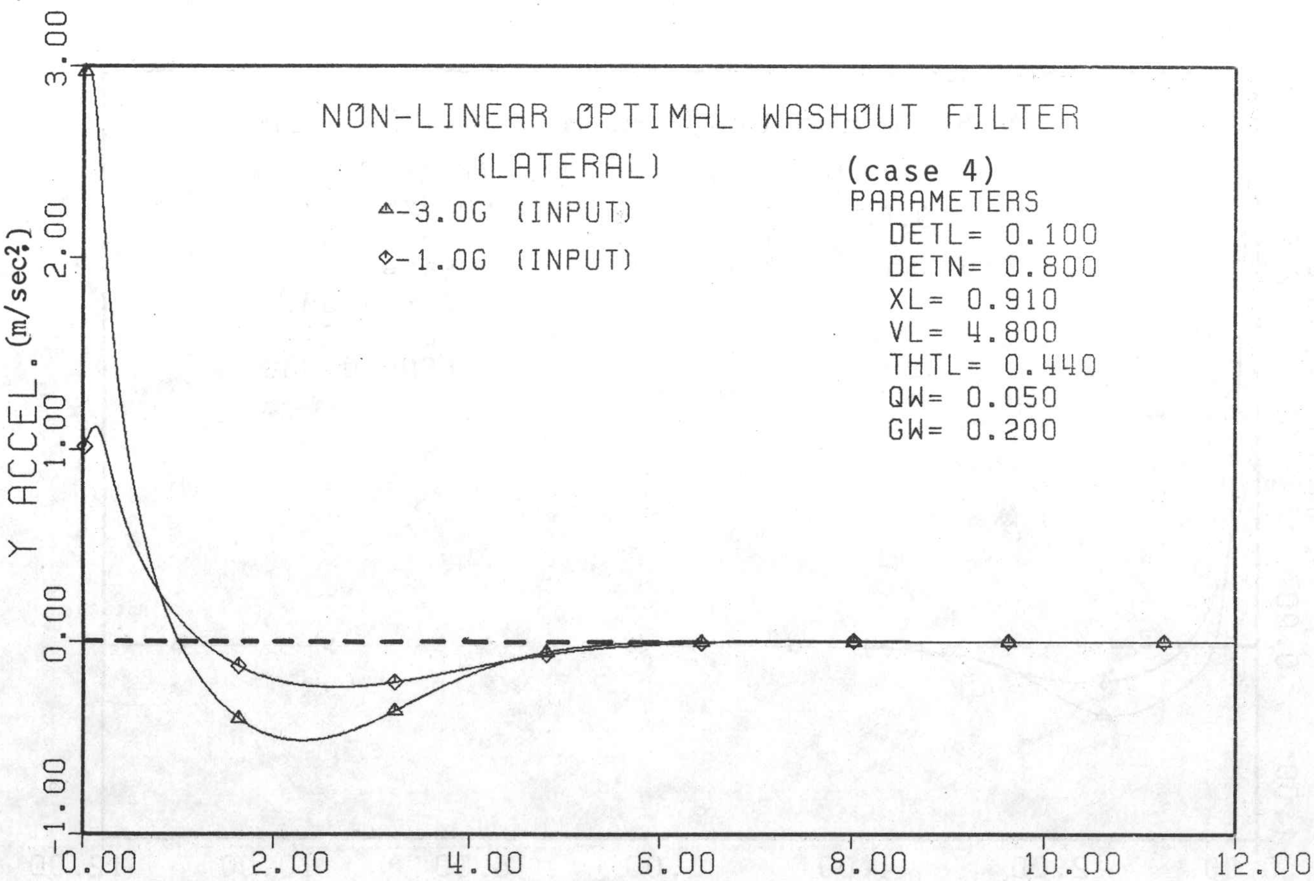


FIGURE 6.27

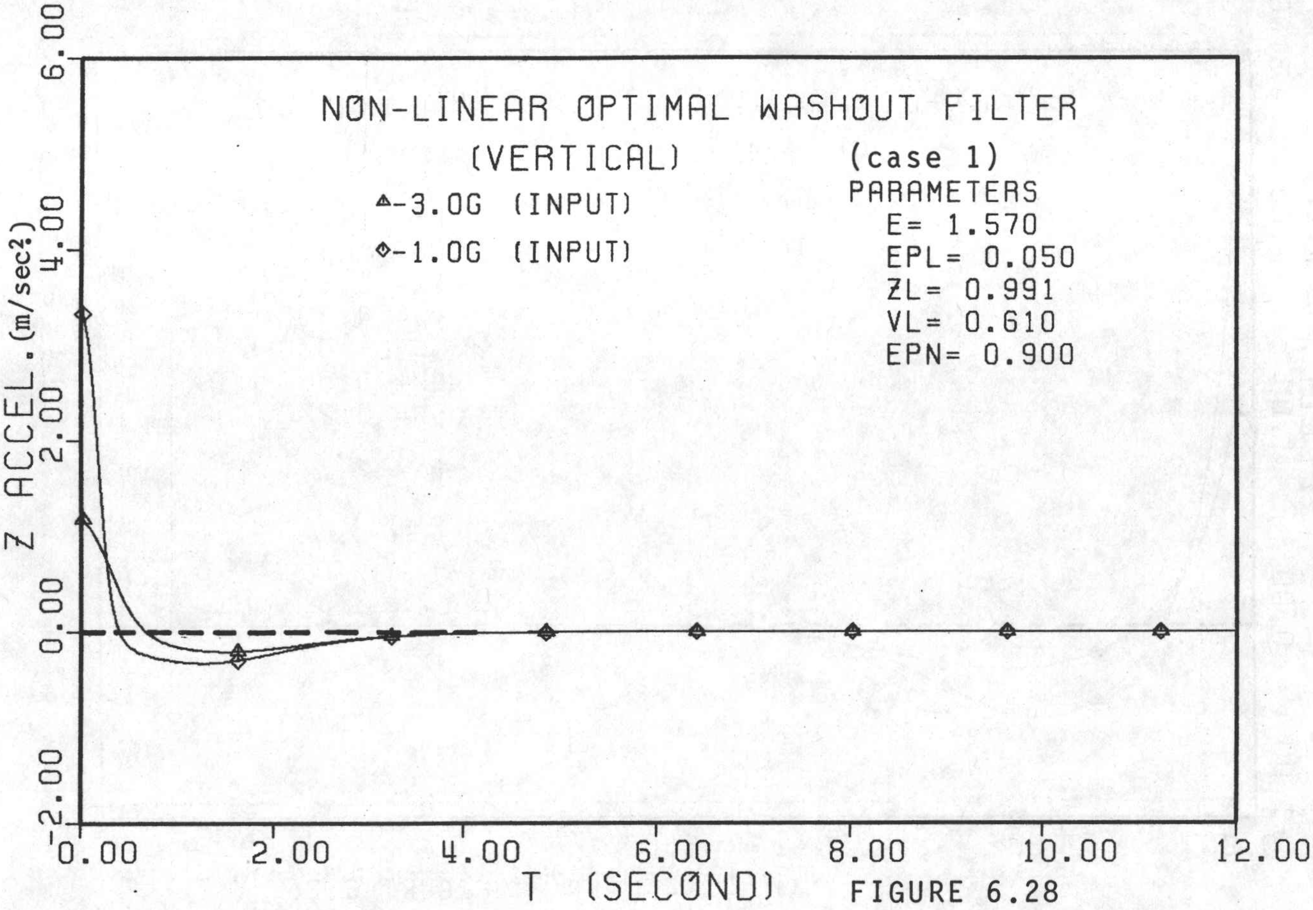


FIGURE 6.28

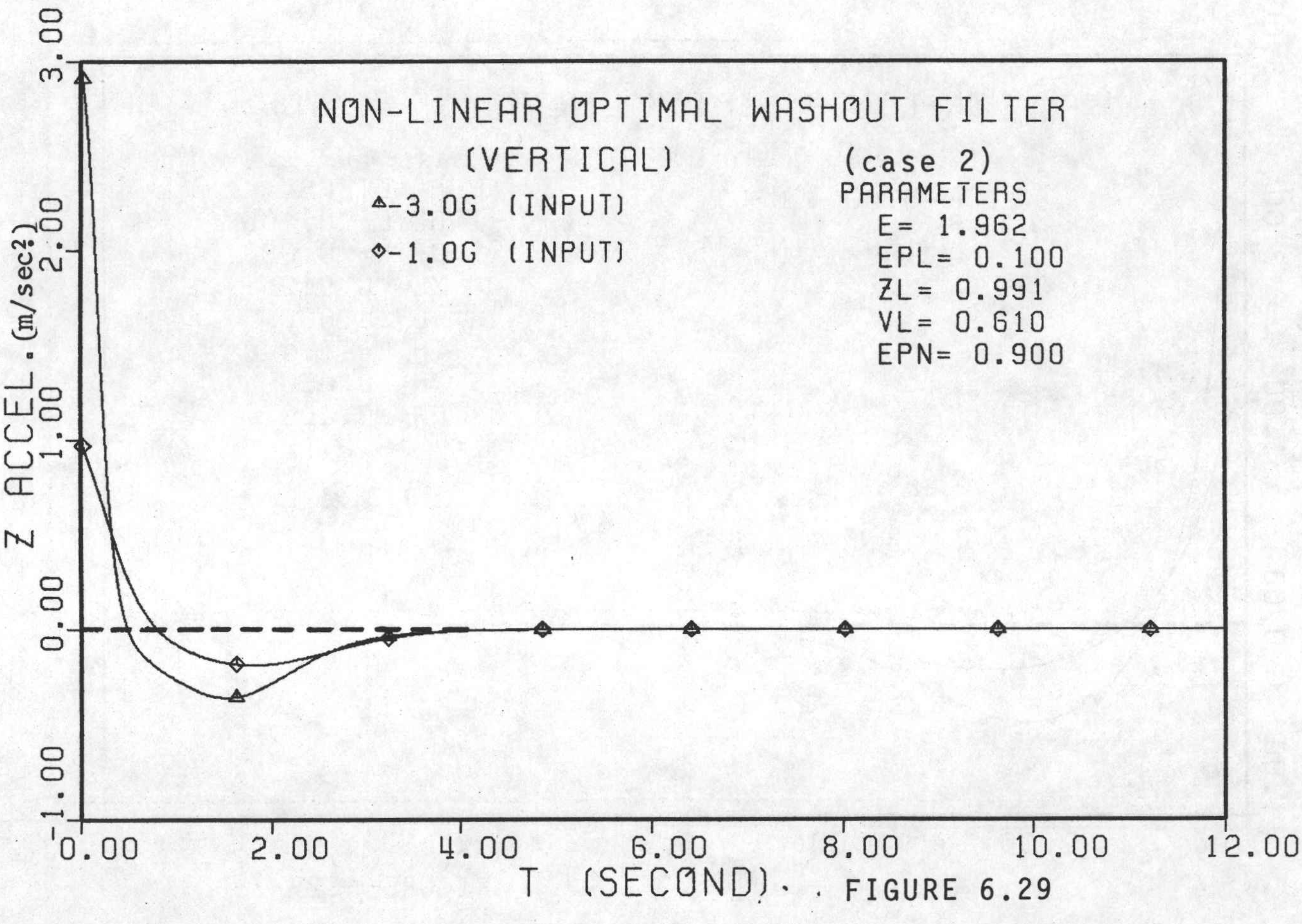


FIGURE 6.29

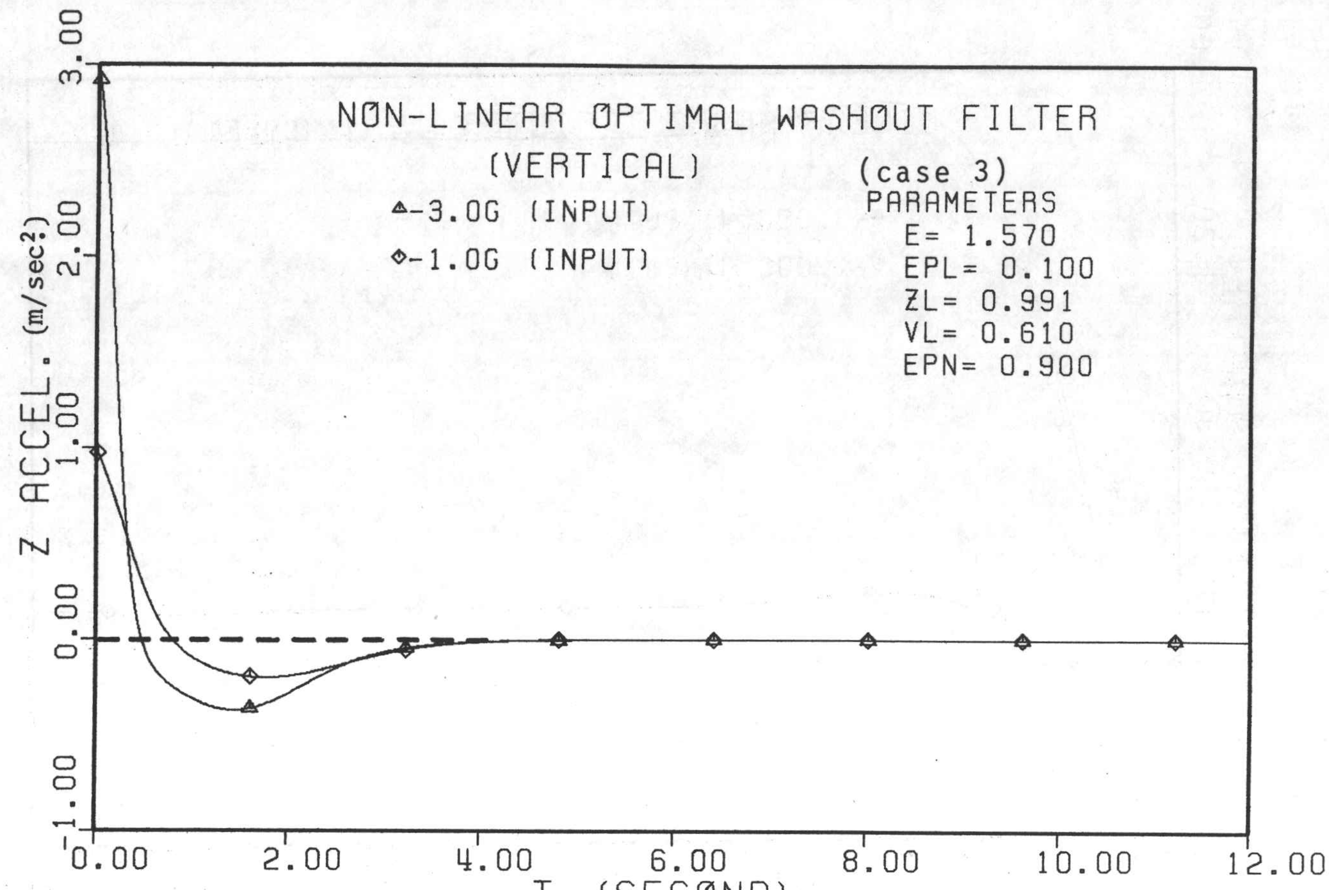


FIGURE 6.30

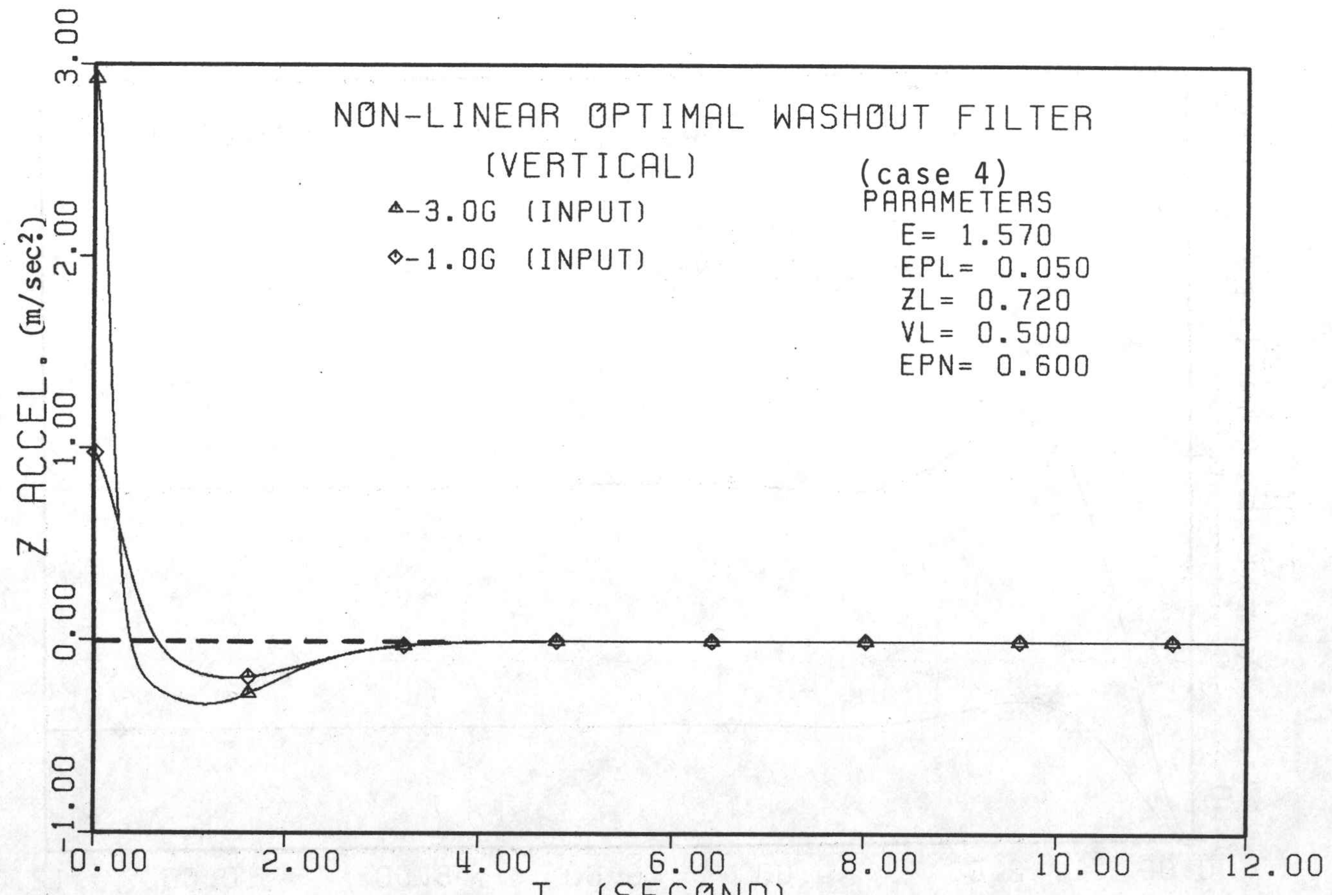


FIGURE 6.31

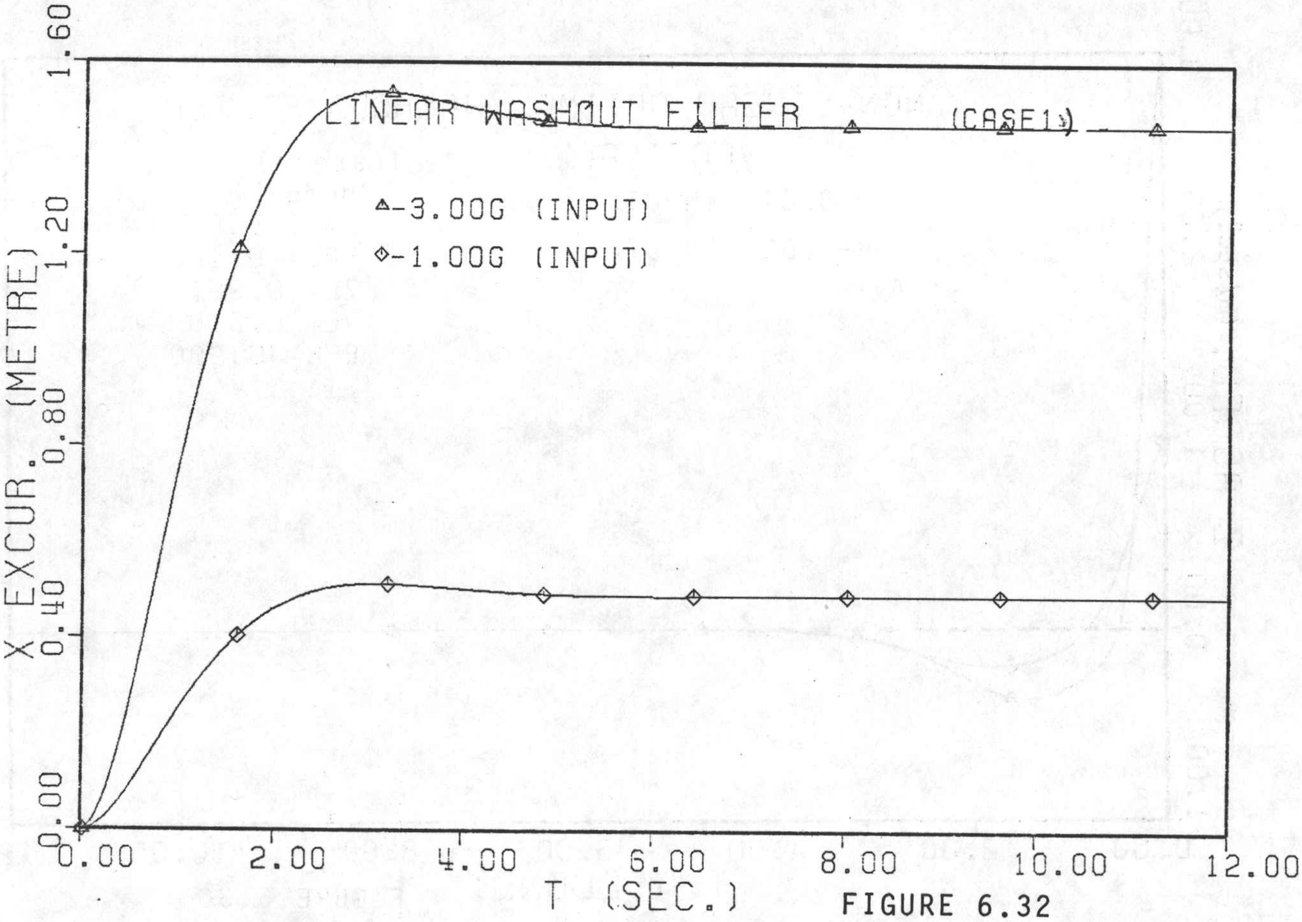


FIGURE 6.32

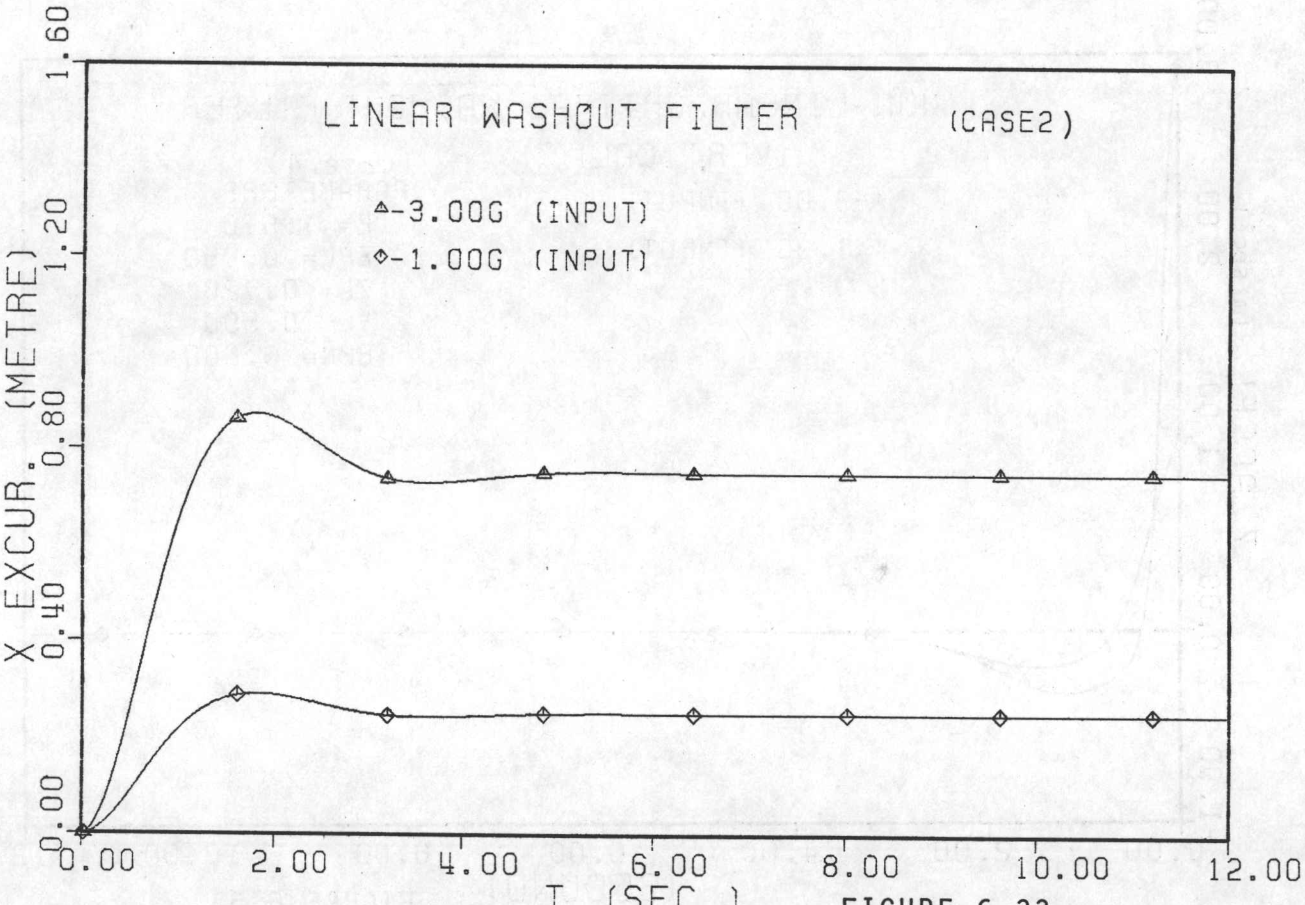


FIGURE 6.33

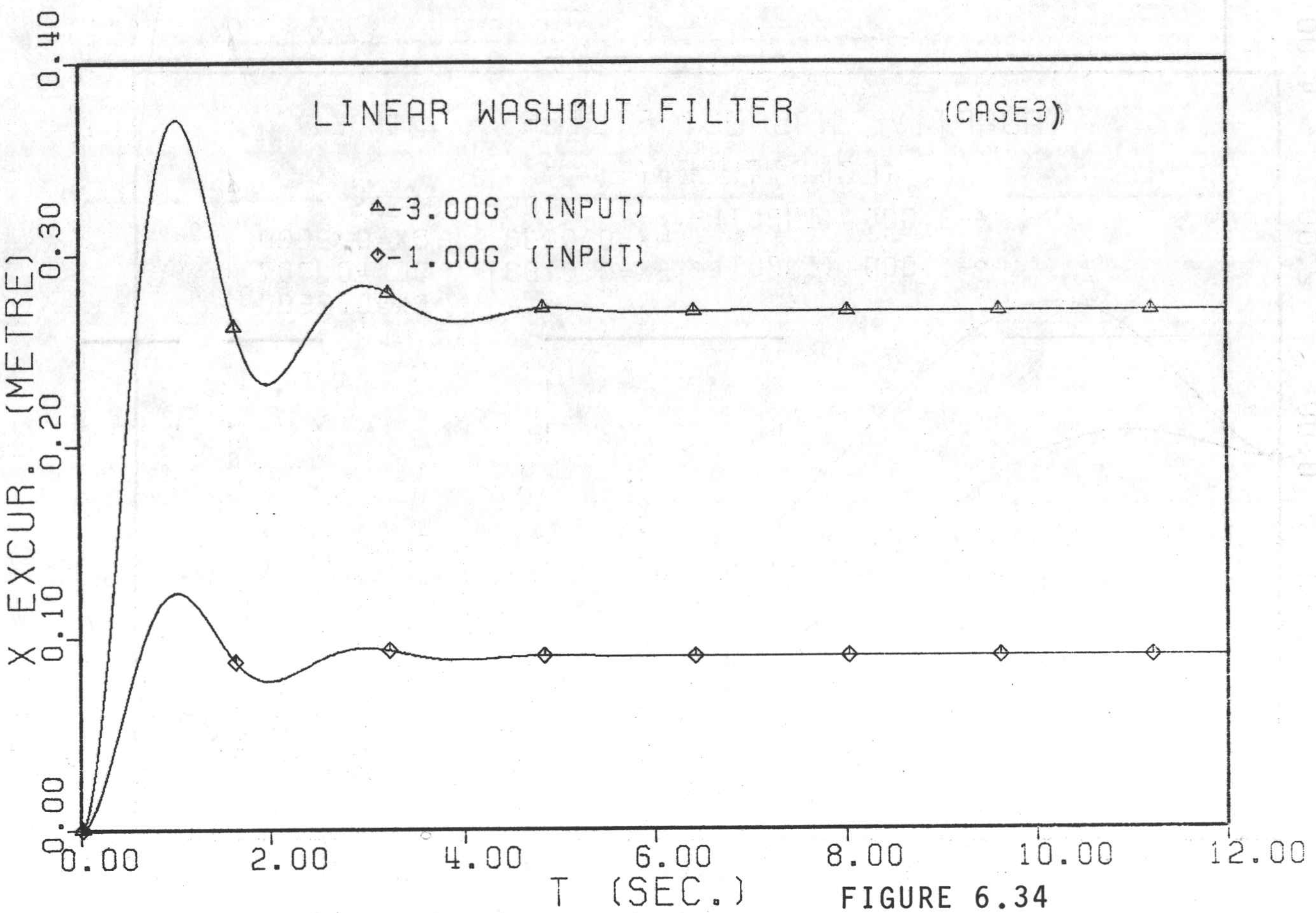


FIGURE 6.34

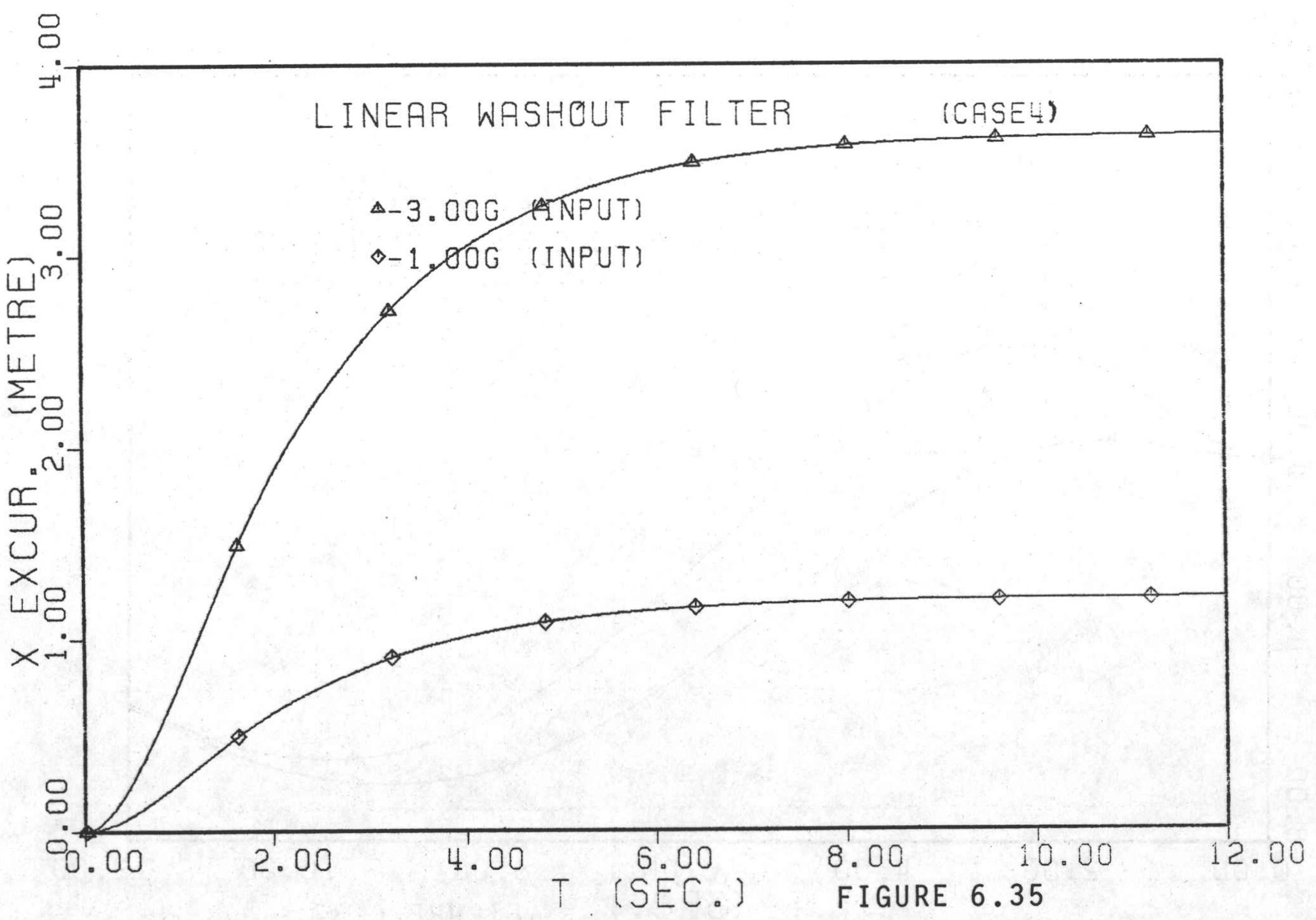


FIGURE 6.35

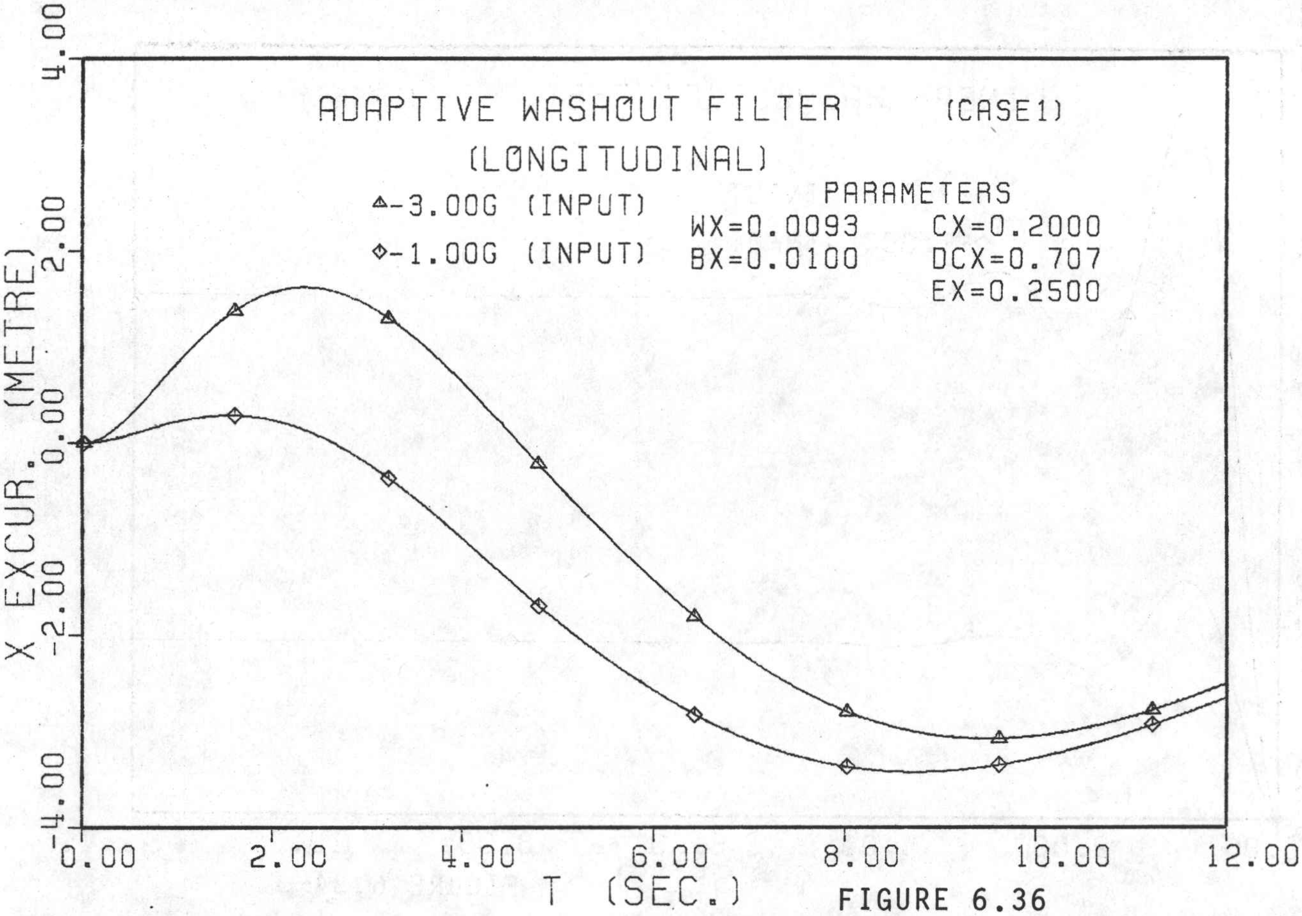


FIGURE 6.36

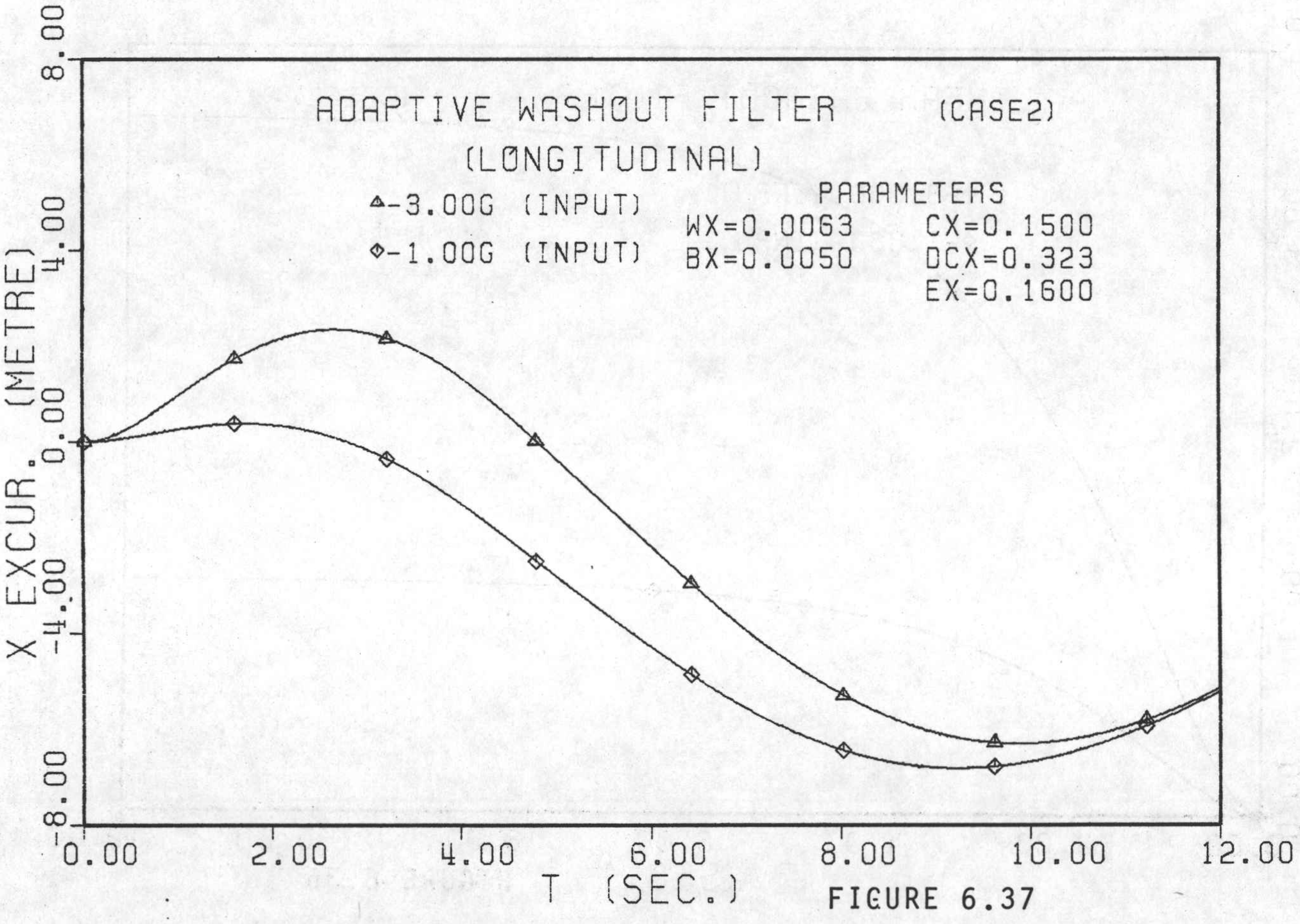


FIGURE 6.37

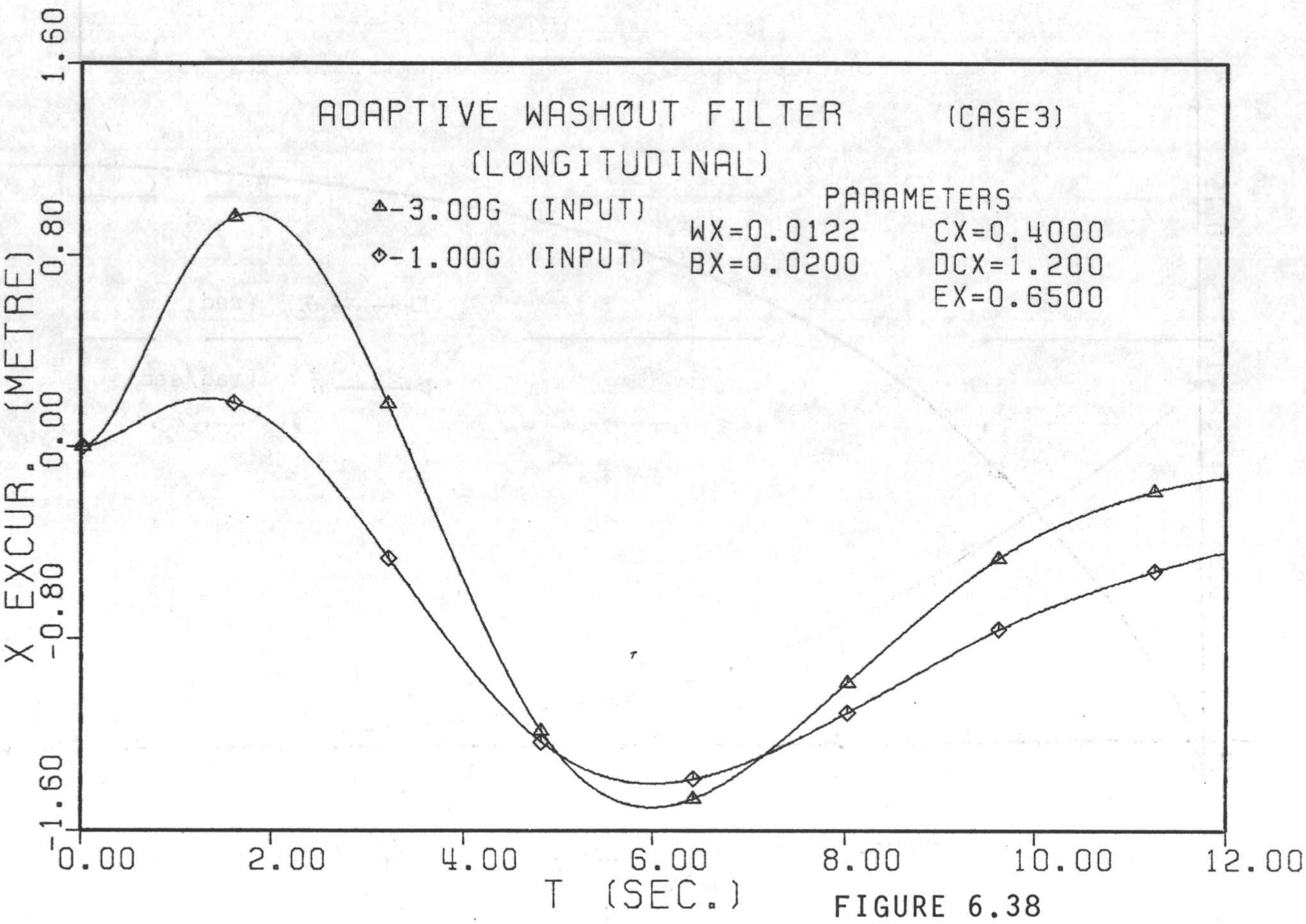


FIGURE 6.38

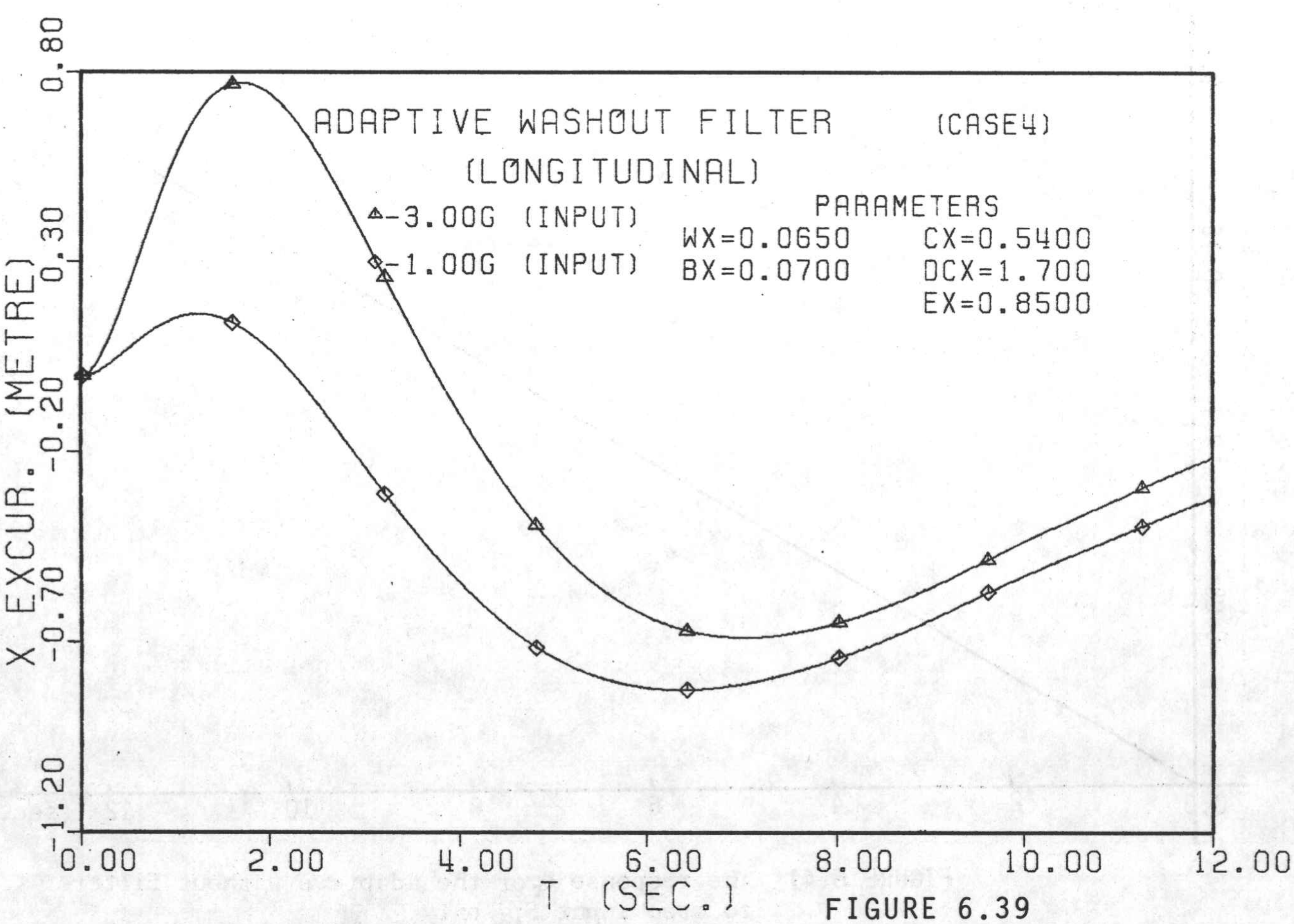


FIGURE 6.39

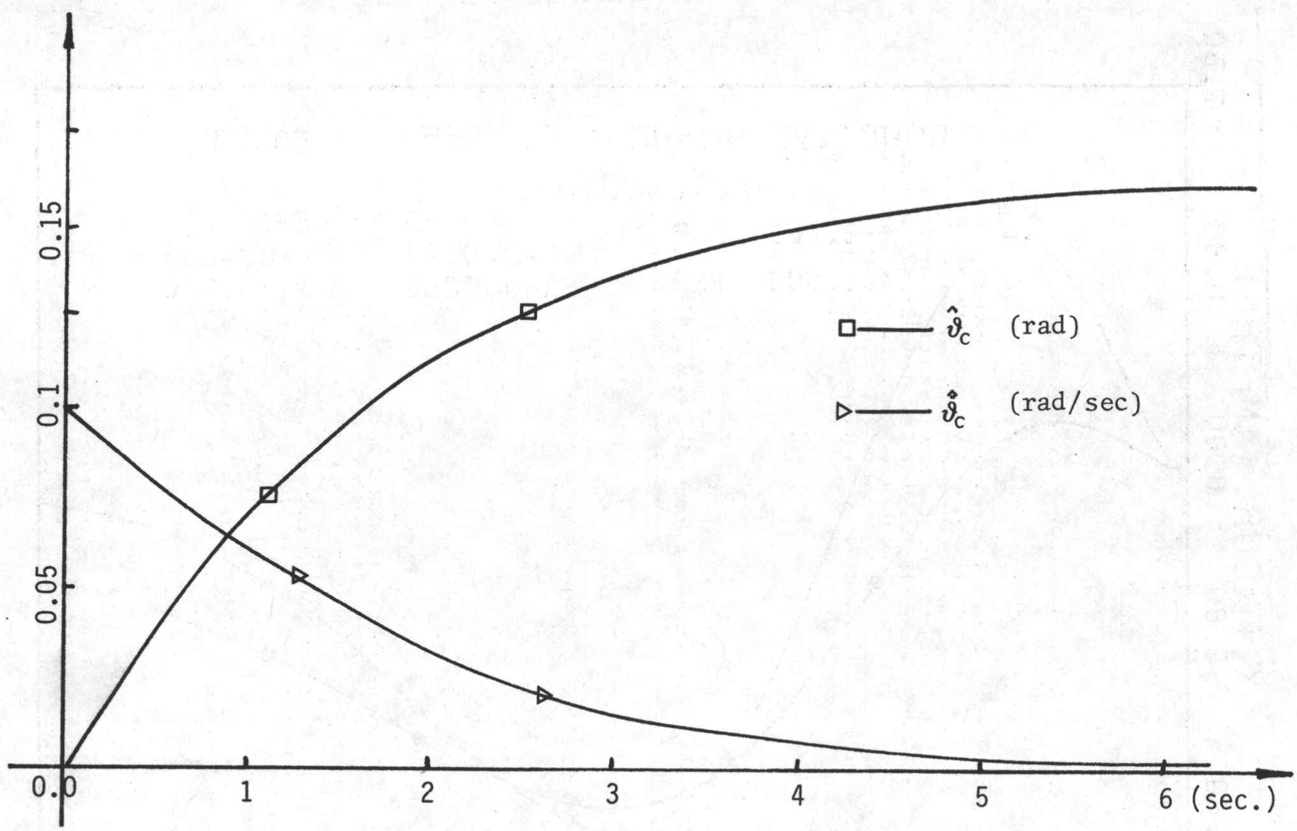


FIGURE 6.40 The responses $\hat{\theta}_c$ and $\hat{\dot{\theta}}_c$ of the adaptive washout filter.

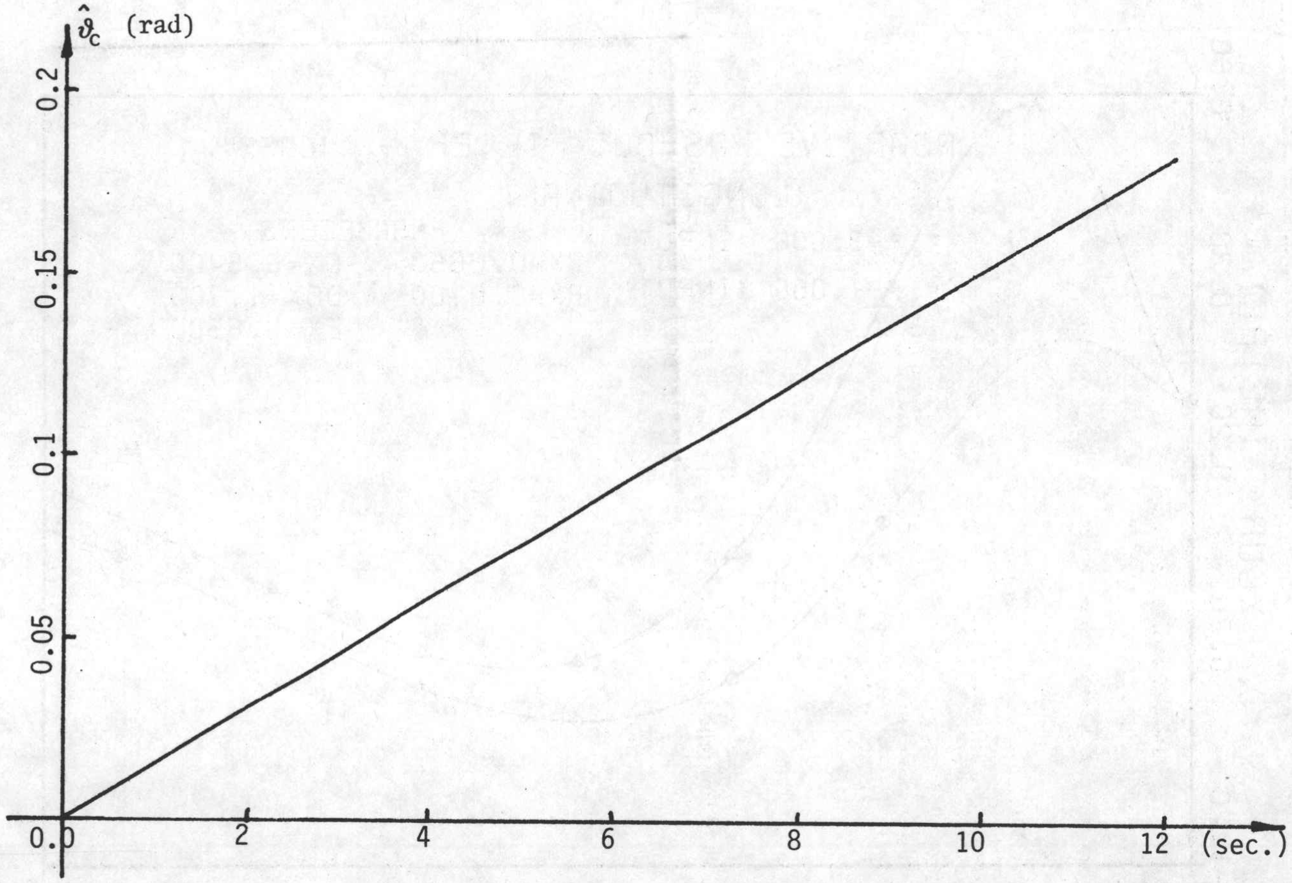


FIGURE 6.41 The response $\hat{\theta}_c$ of the adaptive washout filter to step input f_{ix} only.

UTIAS Technical Note No. 246

Institute for Aerospace Studies, University of Toronto (UTIAS)
4925 Dufferin Street, Downsview, Ontario, Canada, M3H 5T6

A STUDY OF WASHOUT FILTERS FOR A SIMULATOR MOTION BASE

Liu, Zhi-Qiang

1. Flight simulation 2. Nonlinear optimal control 3. Washout filters

I. Liu, Zhi-Qiang II. UTIAS Technical Note No. 246

The conventional linear washout filter and coordinated adaptive washout filter for a six-degree-of-freedom flight simulator are surveyed. A nonlinear optimal washout filter based on nonlinear regulator and optimal control theories has been synthesized. The proposed nonlinear optimal washout filter is capable of producing the drive signal according to the magnitudes of inputs while it minimizes the given performance criterion. For each channel four different cases are tested using computer simulation. Comparisons are made with the results obtained from a linear washout filter and an adaptive washout filter. The observation is that the nonlinear optimal and adaptive washout filters are superior to the linear washout filters in some aspects. Recommendations for future work and improvement are also included.

Available copies of this report are limited. Return this card to UTIAS, if you require a copy.

UTIAS Technical Note No. 246

Institute for Aerospace Studies, University of Toronto (UTIAS)
4925 Dufferin Street, Downsview, Ontario, Canada, M3H 5T6

A STUDY OF WASHOUT FILTERS FOR A SIMULATOR MOTION BASE

Liu, Zhi-Qiang

1. Flight simulation 2. Nonlinear optimal control 3. Washout filters

I. Liu, Zhi-Qiang II. UTIAS Technical Note No. 246

The conventional linear washout filter and coordinated adaptive washout filter for a six-degree-of-freedom flight simulator are surveyed. A nonlinear optimal washout filter based on nonlinear regulator and optimal control theories has been synthesized. The proposed nonlinear optimal washout filter is capable of producing the drive signal according to the magnitudes of inputs while it minimizes the given performance criterion. For each channel four different cases are tested using computer simulation. Comparisons are made with the results obtained from a linear washout filter and an adaptive washout filter. The observation is that the nonlinear optimal and adaptive washout filters are superior to the linear washout filters in some aspects. Recommendations for future work and improvement are also included.

Available copies of this report are limited. Return this card to UTIAS, if you require a copy.

UTIAS Technical Note No. 246

Institute for Aerospace Studies, University of Toronto (UTIAS)
4925 Dufferin Street, Downsview, Ontario, Canada, M3H 5T6

A STUDY OF WASHOUT FILTERS FOR A SIMULATOR MOTION BASE

Liu, Zhi-Qiang

1. Flight simulation 2. Nonlinear optimal control 3. Washout filters

I. Liu, Zhi-Qiang II. UTIAS Technical Note No. 246

The conventional linear washout filter and coordinated adaptive washout filter for a six-degree-of-freedom flight simulator are surveyed. A nonlinear optimal washout filter based on nonlinear regulator and optimal control theories has been synthesized. The proposed nonlinear optimal washout filter is capable of producing the drive signal according to the magnitudes of inputs while it minimizes the given performance criterion. For each channel four different cases are tested using computer simulation. Comparisons are made with the results obtained from a linear washout filter and an adaptive washout filter. The observation is that the nonlinear optimal and adaptive washout filters are superior to the linear washout filters in some aspects. Recommendations for future work and improvement are also included.

Available copies of this report are limited. Return this card to UTIAS, if you require a copy.