TRANSIENT FLOW STABILIZATION OF

AN HYDRAULIC SERVOMECHANISM

By

ASLAUG HARALDSDOTTIR

Bachelor of Science

University of Iceland

Reykjavik, Iceland

1980

Submitted to the Faculty of the Graduate College
of the Oklahoma State University
in partial fulfillment of the requirements
for the Degree of
MASTER OF SCIENCE
May, 1982

TRANSIENT FLOW STABILIZATION OF

AN HYDRAULIC SERVOMECHANISM

Thesis Approved:

Karl N. Reid

Thesis Adviser

George Maurer

R L Lowery

Norman N Durham

Dean of Graduate College

ACKNOWLEDGMENTS

I want to express my deep appreciation to my major adviser, Dr. Karl N. Reid, for his guidance and assistance throughout this study. Also, for his excellent instruction and many interesting discussions in coursework at Oklahoma State University, which provided the background neccessary to carry this study through.

Dr. Richard L. Lowery and Dr. G. F. Mauer provided invaluable assistance in the preparation of the final manuscript and I want to thank them for their guidance.

I also want to thank Dr. B. N. Murali and Dr. Saeed Hamid for their information and encouragements in the first part of this study. Dr. Lynn R. Ebbesen provided much assistance on system's modelling for the computer and on the use of the University's IBM 370 computer. Thank you.

Financial support has been provided by the MAE department through all of my graduate program. I want to extend my sincere appreciation to the faculty for this support.

Finally, my parents deserve my deepest gratitude for their constant encouragement throughout my entire education. They have made many sacrifices to make this achievement possible.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure | Page

NOMENCLATURE

a - Flow displacement sensitivity (in^2/sec)

A - Area (in^2)

b - Flow pressure sensitivity ($in^5/lb_f \cdot sec$)

B - Level of Coulomb friction (lb_f/in^2)

C - Damping coefficient ($lb_f \cdot sec/in$)

c_d - Orifice discharge coefficient

c_r - Radial clearance between piston and housing (in)

c_3 - Leakage coefficient past main piston ($in^5/lb_f \cdot sec$)

d - Diameter (in)

f_i - Vector function of state variables (linearized)

F_i - Vector function (nonlinear)

F - Force (lb_f)

k - Ratio of specific heats, c_p/c_v

k_i - Spring constants (lb_f/in)

L - Length (in)

m - Mass ($lb_f \cdot sec^2/in$)

P - Gage pressure (psig)

Q - Volumetric flow rate (in^3/sec)

s - Laplace variable

t - Time (sec)

T - Period (sec)

U - Valve underlap (in)

V - Volume (in^3)

w_1 - Orifice width (in)

x - Displacement (in)

X - Vector of algebraic variables in nonlinear model

y - Vector of dynamic state variables in linearized model

Y - Vector of dynamic state variables in nonlinear model

α - Feedback coefficient

Δ - A small deflection in a variable

ρ - Mass density ($lb_f \cdot sec^2/in^4$)

μ - Absolute fluid viscosity ($lb_f \cdot sec/in^2$)

ω_n - Natural frequency (rad/sec)

CHAPTER I

INTRODUCTION

1.1 Background

The subject of this study is an hydromechanical servomechanism, known as a 'governing valve servomotor', that has been in use in power plants for the past several years. This servomotor controls steam valves which in turn modulate the flow of steam into the turbine in response to the external electrical load demands on the plant.

This servomechanism is subjected to a significant Coulomb friction load, and during its use it has shown some tendency toward unstable or oscillatory behaviour.

With the aid of the digital computer, this system can be analyzed in its present form and, if neccessary, some appropriate compensation introduced and tested.

1.2 Objectives and Methods of Study

The objectives of this study were as follows:

1. To investigate the dynamic behaviour of the servomechanism, so that the conditions for stability could be determined.

2. To propose a compensation design that would eliminate or lessen the possibilities of oscillatory or unstable performance.

This hydromechanical system was not available for experimental investigation, so the study was confined entirely to analytical methods. Therefore, the major steps in the investigation were:

1. Derivation of a mathematical model, that represents all the significant static and dynamic factors of the system. Because this is a hydraulic system, the model unavoidably contains several nonlinear parts.

2. Linearization of the model equations and a study of the system significant dynamic characteristics. The methods used for this part of the study were computer simulation and the Routh criterion test. Several different parameter configurations were considered.

3. Computer simulation of the complete nonlinear model including the Coulomb friction load. The computer model was set up to make it straightforward to add any neccessary compensation.

1.3 Results and Recommendations

After the nonlinear model had been thoroughly tested (see Chapter IV) for various parameter combinations, it was concluded that the system stability depends on the Coulomb friction load. This means of stabilization can be considered a rather undependable way of providing sufficient degree of stability, and therefore an alternate stabilization design is suggested. It is apparent that the most practical solution and a very effective one is 'transient flow stabilization'. This concept in fluid power control system design was first introduced by Shearer (1) for pneumatic system stabilization. The technique was shown to be quite successful in enhancing damping in a lightly damped system.

Chapter V describes the design of a transient flow stabilizer for the hydromechanical system. Simulation results show the compensation to be most effective.

A recommendation of this study is that a transient flow stabilizer be built and installed in an existing system. Then, tests should be conducted to verify that the device produces the desired improvements in the system performance.

CHAPTER II

SYSTEM DYNAMIC MODEL

2.1 Overall Operation of the Servomechanism

Figure 1 shows schematically the servomotor and its connection to the steam chest governing valves, that control the flow of steam into the turbine. The servomotor piston is connected through a link (35) to the steam chest operating lever (32). The lever is fulcrummed at the other end on the pin (4), carried in a bracket (2), which is bolted to the inlet end of the steam chest body (1). The valve stems are connected to the operating lever by links (5), (10), (14) and (18) and a set of springs provides the neccessary closing force and stability in operation.

The servomotor itself is shown schematically in Figure 2. The servomotor main piston (1) is connected to the governing valve operating lever, so that an upward movement of the piston opens the valves and a downward movement closes them.

The principal parts of the servomotor are the servomotor piston (1), the relay piston (6), which has four ports machined in the stem, the relay plunger (4), the relay bushing (7), the cam (8) and the feedback lever (9). Table I lists the main parts in more detail. A small drilled hole in the relay plunger (4) connects the high pressure oil inlet to the chamber above the relay plunger. The central hole

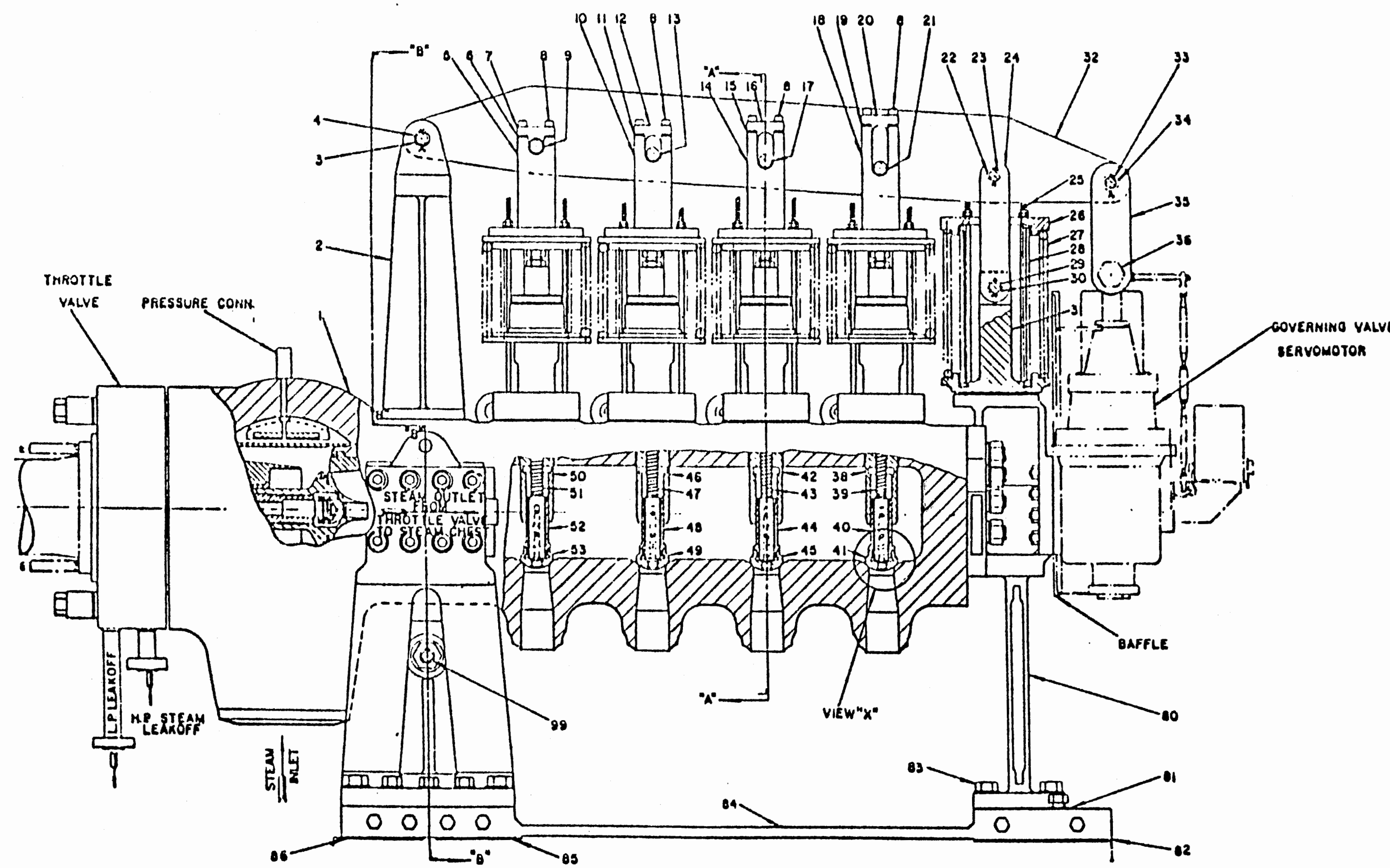


Figure 1. Servomotor and Steam Chest Governing Valves.

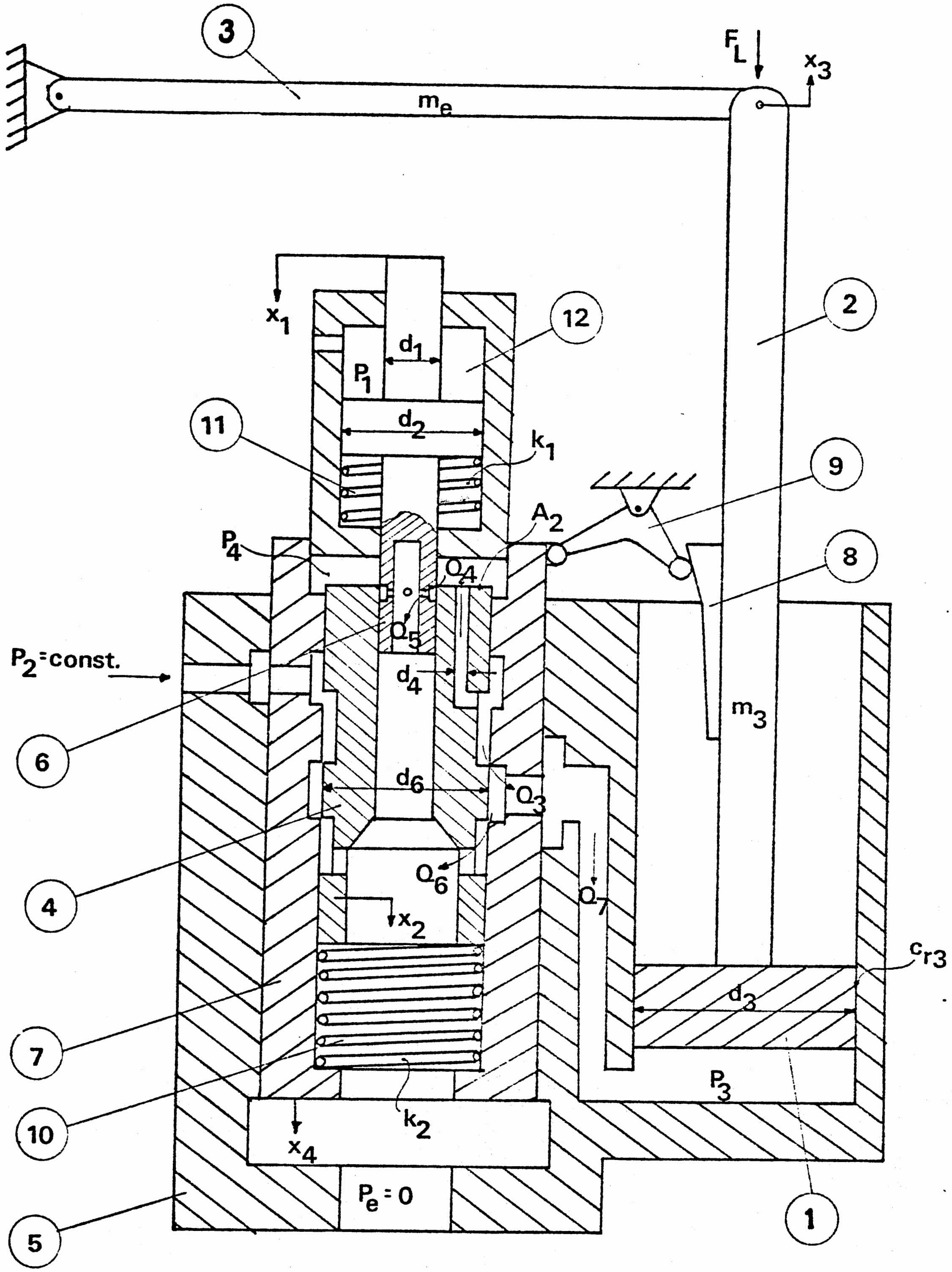


Figure 2. Schematic of the Hydraulic Servomechanism.

TABLE I

A LIST OF MAJOR PARTS OF THE SERVOMECHANISM, NUMBERS REFER TO FIGURE 2

Item No.	Item Name
1	Servomotor Piston
2	Servomotor Piston Rod
3	Governing Valves Operating Lever
4	Relay Plunger
5	Servomotor Cylinder
6	Relay Piston
7	Relay Bushing
8	Cam
9	Follow-Up Lever
10	Spring, Prestressed
11	Spring
12	Control Oil Chamber

through the entire length of the relay plunger connects the chamber above the relay plunger to drain by way of the four ports in the stem of the relay piston (6).

The pressure in the chamber above the relay plunger (4) is maintained by the ports in the relay piston stem so that the force of the spring (10) and the force due to the oil pressure (P_4) are balanced. There will be a continous flow through the ports in the stem and the small drilled hole, when the ports are open. An upward movement of the relay piston increases the flow through the ports to drain, thereby decreasing the pressure above the relay plunger until it also moves upward. Conversely, downward movement of the relay piston decreases the flow through the ports to drain, thereby increasing the pressure above the relay plunger until it also moves downward. This will cause the relay plunger to follow all movements of the relay piston as though they were connected to each other.

The control pressure (P_1) is admitted to the chamber above the relay piston (12) and exerts a force tending to move the relay piston downward. This force is opposed by the compression spring (11) which tends to move the relay piston upward. Therefore, any change in control pressure unbalances the forces on the relay piston and causes the relay piston to move until the spring force again balances the control pressure force. This movement of the relay piston in turn produces a corresponding movement of the relay plunger (4).

When the relay plunger moves downward in response to an increase in control oil pressure, high pressure oil will flow through the metering ports formed by the relay plunger and relay bushing, to the chamber beneath the servomotor piston (1), causing the piston to move upward and open the governing valves. The cam (8), mounted on the piston rod (2), then permits the follow up lever (9) to move the relay bushing in a downward direction to close the metering ports and stop the piston motion. Thus, for each governing control oil pressure, there will be a corresponding position of the relay piston, the relay plunger, the servomotor piston, the relay bushing, and the steam chest governing valves.

The system described above is an hydromechanical position feedback control system, where oil pressure is the input and piston displacement is the output.

2.2 Development of the Dynamic Model for the Original System

In order to arrive at a mathematical model of this servo system that is not too complex, but still accurate enough for this study, some simplifying assumptions have to be made. These assumptions can be summarized as follows (refer to Figure 2 and Table II for notation and further description):

1. Inertia forces and leakage effects on the relay piston (6), the relay plunger (4) and the relay bushing (7) are negligible.

2. The flow through the capillary tube orifice in the relay plunger is assumed fully developed and laminar.

3. All orifices are symmetrical. Orifice 5 has no underlap, but a radial clearance c_{r1}. Orifices 3 and 6 have an underlap of U each and a radial clearance c_{r2}.

4. All orifices have an equal and constant discharge coefficient, c_d.

5. Steady-state and dynamic flow induced forces are negligible.

6. All connecting passages are sufficiently short and wide to eliminate any resistance or 'transmission line' effects.

7. Leakage flow past the main piston is assumed fully developed and laminar. Other leakage flow within the system is negligible.

8. Supply pressure, P_2, is constant and exhaust pressure $P_e=0$.

9. Temperature and fluid properties (β,μ) are constant.

10. Changes in fluid volumes (υ_3 and υ_4) are negligible, but the time rates of change are significant.

11. Changes in fluid density ρ are negligible, but the time rate of change is significant.

The force balance on the relay piston is

$$P_1 \, \pi (d_2^2 - d_1^2)/4 = k_1 \, x_1 . \tag{2.1}$$

The force balance on the relay plunger is

$$P_4 \, A_2 = k_2 \, (x_2 - x_4) + F_s . \tag{2.2}$$

Here, F_s is the initial force in the prestressed spring (10) and A_2 is the area above the relay plunger, being acted upon by the pressure P_4.

The pressure drop across the capillary tube through the relay plunger is given by

$$P_2 - P_4 = \frac{128 \, \mu L_4 Q_4}{\pi \, d_4^4} \left(1 + \frac{2.28 \, \rho \, Q_4}{16 \, \pi \, \mu L_4} \right) . \tag{2.3}$$

The flow equations for the sharp edged orifices can be written as follows:

For $x_2 - x_1 > 0$:

$$Q_5 = \mathrm{sgn}(P_4)\ c_d \pi w_1 \sqrt{(x_2 - x_1)^2 + c_{r1}^2}\ \sqrt{(2/\rho)\ |P_4|}$$

For $x_2 - x_1 \leqslant 0$: (2.4)

$$Q_5 = \mathrm{sgn}(P_4)\ c_d \pi w_1 c_{r1} \sqrt{(2/\rho)\ |P_4|}$$

For $x_2 - x_4 + U > 0$:

$$Q_3 = \mathrm{sgn}(P_2 - P_3)\ c_d \pi d_6 \sqrt{(x_2 - x_4 + U)^2 + c_{r2}^2}\ \sqrt{(2/\rho)\,|P_2 - P_3|}$$

For $x_2 - x_4 + U \leqslant 0$: (2.5)

$$Q_3 = \mathrm{sgn}(P_2 - P_3)\ c_d \pi d_6 c_{r2} \sqrt{(2/\rho)\ |P_2 - P_3|}$$

For $x_4 - x_2 + U > 0$:

$$Q_6 = \mathrm{sgn}(P_3)\ c_d \pi d_6 \sqrt{(x_4 - x_2 + U)^2 + c_{r2}^2}\ \sqrt{(2/\rho)\ |P_3|}$$

For $x_4 - x_2 + U \leqslant 0$: (2.6)

$$Q_6 = \mathrm{sgn}(P_3)\ c_d \pi d_6 c_{r2} \sqrt{(2/\rho)\ |P_3|}\ .$$

These equations include all possible conditions of position of the relay plunger with respect to the relay piston.

Flow continuity requires that

$$\rho_3 Q_7 = \rho_3 Q_3 - \rho_3 Q_6 \tag{2.7}$$

$$\rho_3 \, Q_7 = \frac{d}{dt}(\rho_3 \upsilon_3) + c_3 P_3; \qquad c_3 = \frac{\pi \, d_3^2 \, (c_{r3}/2)^3}{12 \, \mu \, L_3} \tag{2.8}$$

$$\rho_4 Q_4 = \frac{d}{dt}(\rho_4 \upsilon_4) + \rho_4 Q_5 . \tag{2.9}$$

The time rate of change terms in equations (2.8) and (2.9) can be rewritten using the equations of state for the hydraulic fluid as follows:

$$\frac{d}{dt}(\rho_3 \upsilon_3) = \upsilon_{3i} \frac{\rho_3}{\beta} \left(\frac{d}{dt} P_3 \right) + \frac{\pi \, d_3^2}{4} \left(\frac{d}{dt} x_3 \right) \tag{2.10}$$

$$\frac{d}{dt}(\rho_4 \upsilon_4) = \upsilon_{4i} \frac{\rho_4}{\beta} \left(\frac{d}{dt} P_4 \right) + \rho_4 A_2 \left(\frac{d}{dt} x_2 \right) . \tag{2.11}$$

The force balance on the main piston can be expressed as

$$m_3 \ddot{x}_3 + C \, \dot{x}_3 + F_L = \frac{\pi d_3^2}{4} P_3 \tag{2.12}$$

where

$$F_L = m_e \ddot{x}_3 + F_{ext} \tag{2.13}$$

and

$$F_{ext} = f(x_3) . \tag{2.14}$$

Finally, the feedback relationship can be written

$$x_4 - x_3^* = \alpha \, x_3 \tag{2.15}$$

where x_3^* is a constant due to the initial displacement of the relay bushing and α is the gain given by the geometry of the feedback lever.

The final model consists of 15 nonlinear algebraic/differential equations and the following 16 unknowns: x_1, x_2, x_3, x_4, P_1, P_3, P_4, Q_3, Q_4, Q_5, Q_6, Q_7, $(\rho_3 \upsilon_3)$, $(\rho_4 \upsilon_4)$, F_L and F_{ext}. Thus, it is possible to solve (at least implicitly) for one variable (e.g. x_3) as function of another variable (e.g. P_1).

2.3 The System Parameters

Appendix A describes how the various system parameters were determined. Figure 3 shows the external load F_{ext} plotted in terms of steady-state pressure under the main piston, P_3, as a function of piston displacement, x_3. This relationship is highly nonlinear; the discontinuities are due to the successive openings of the steam valves. This graph includes implicitly all steady-state spring forces and all flow induced forces from the steam valves. The dotted line represents data that was supplied by the manufacturer of the servo motor, and the solid line shows data measured in a field installation by K. N. Reid. The solid line is the one used as a reference in this study, as it represents the most recent data available.

These data do not include any information on the load dynamic behaviour. However, it can be assumed that some level of Coulomb friction would exist between the piston and the cylinder, and between the governing valve stems and packings, and that it should be taken into account in the modelling. Based on indirect measurements in a field installation made by K. N. Reid, it is assumed that the Coulomb friction

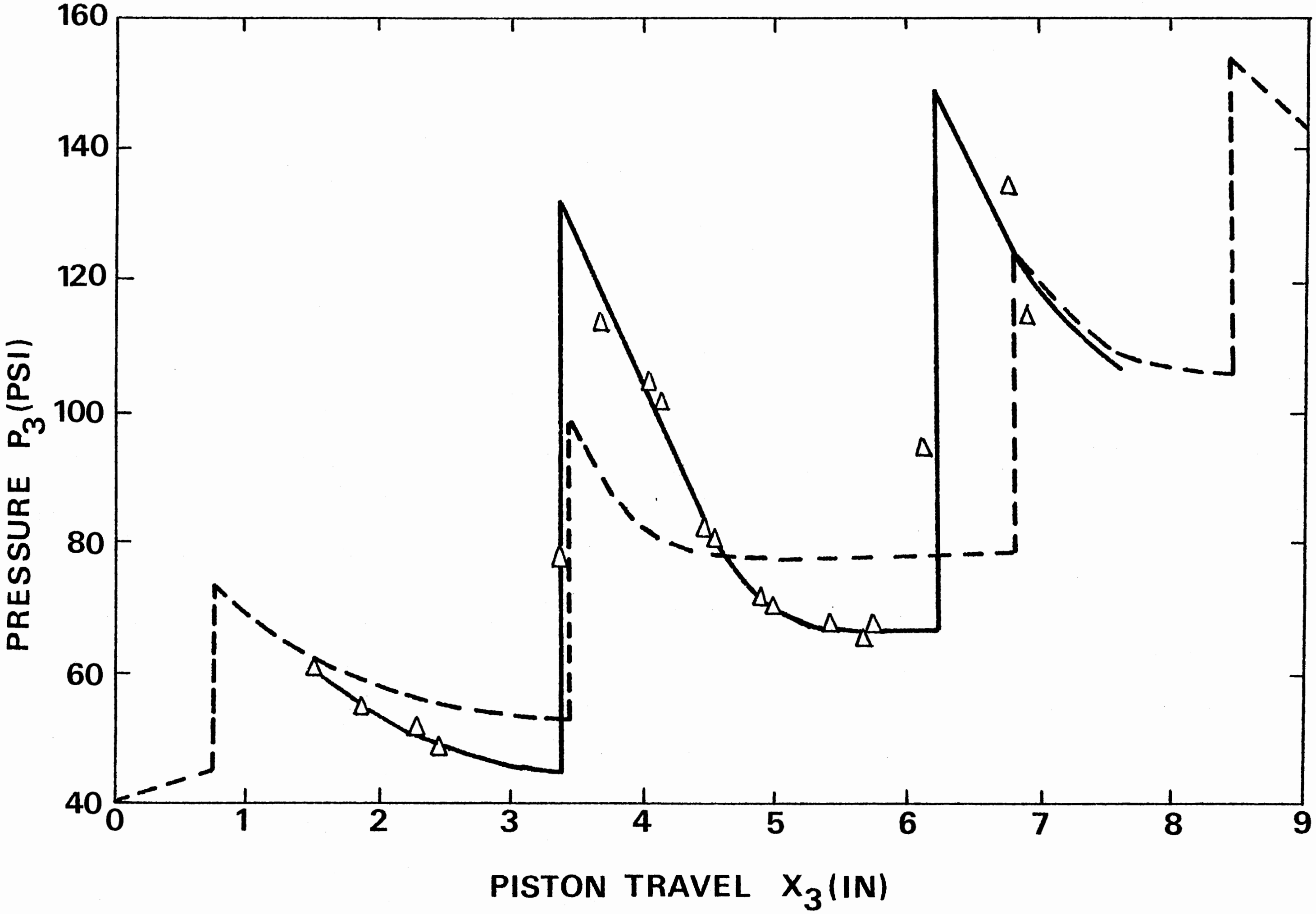


Figure 3. External Load as a Function of Piston Displacement.

level is equivalent to a value of P_3 of 5 - 10 psi.

Figure 4 shows a graph of control oil pressure, P_1, v.s. main piston displacement, x_3. The feedback gain, α, and the initial displacement of the relay bushing, x_3^*, can be determined from this relationship. This graph is needed for this purpose, because the two parameters can not be determined from drawings available for the feedback mechanism. Figure 4 shows two relationships, one representing a feedback cam with a uniform slope and the other a cam with two slightly different slopes. Curve A is believed to represent the cam used in one typical field installation (V.B. #1). Curve B represents actual measurements made by K. N. Reid in the same field installation. The disparity between the two relations has not been explained. It is believed that curve A was the relation that existed prior to an overhaul of the field system and it is known that Curve B represents the situation after this overhaul.

Appendix A gives a description of how the parameters α and x_3^* were determined by solving the steady-state equations for the servo.

Table II lists numerical values of the parameters in the mathematical model of the servo which has been presented.

There is always some uncertainty as to what value to assume for the fluid bulk modulus, β. The commonly assumed value for the type of oil used in the system, MIL5606 A, is around 230,000 psi., which is a purely theoretical value. The actual value is highly sensitive to the amount of air entrained in the fluid. Experience has shown, that a bulk modulus of the order of 110,000 psi. is a much more realistic esti-

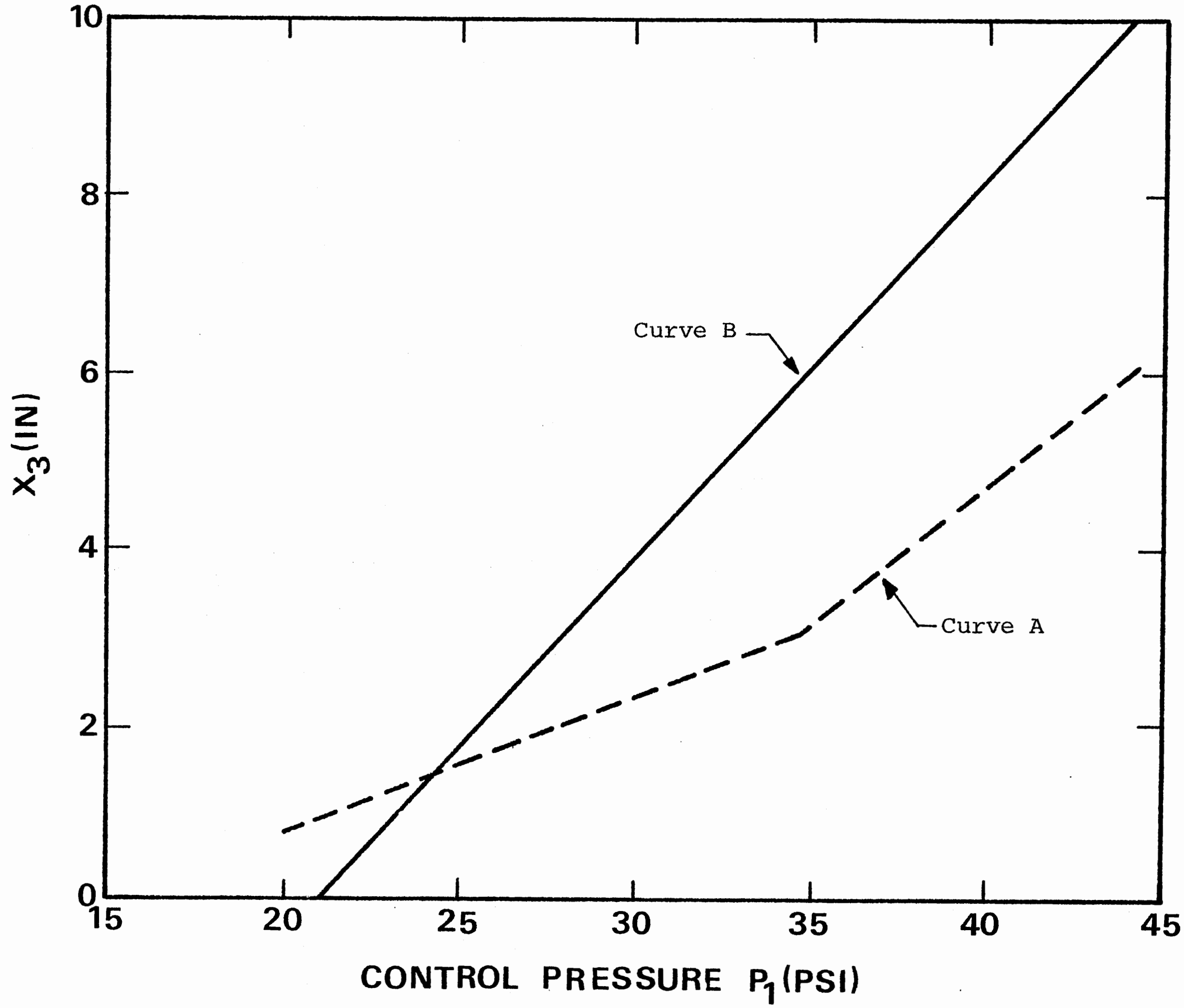


Figure 4. Main Piston Displacement as a Function of Input Control Oil Pressure.

TABLE II

NUMERICAL VALUES OF THE SYSTEM PARAMETERS

Parameter Name	Nominal Value		Description
d_1	0.996	in	Relay Piston Rod Diameter
d_2	2.494	in	Relay Piston Diameter
d_3	9.99	in	Main Piston Diameter
d_4	3/16	in	Capillary Orifice Diameter
d_6	3.495	in	Relay Plunger Diameter
w_1	1.750	in	Relay Piston Orifice Width
A_2	2.314	in^2	Relay Piston Top Area
c_{r1}	0.002	in	Relay Piston Radial Clearance
c_{r2}	0.002	in	Relay Plunger Radial Clearance
c_{r3}	0.002	in	Main Piston Clearance on Diameter
L_4	4.00	in	Capillary Orifice Length
L_3	0.8	in	Length of Leakage Path Past Main Piston
k_1	30.0	lb/in	Pilot Relay Spring Constant (11)
k_2	212.0	lb/in	Main Relay Spring Constant (10)
m_3	0.303	lb·sec/in	Piston Mass
m_e	1.3656	lb·sec/in	Lever and Valve Equivalent Mass
U	0.002	in	Main Valve Underlap
υ_4	11.6	in^3	Fluid Volume above Relay Plunger
υ_3	344.0	in^3	Fluid Volume under Main Piston
μ	$2.2 \cdot 10^{-6}$	$lb \cdot sec/in^2$	Fluid Viscosity
ρ	$8 \cdot 10^{-5}$	$lb \cdot sec^2/in^4$	Fluid Density
β	110,000	psi	Fluid Bulk Modulus of Elasticity
c_d	0.625		Orifice Discharge Coefficient
F_s	13.5	lb	Initial Force in Main Relay Spring

mate in an actual system. The model may be used to study the effect of bulk modulus on the behaviour of the system.

2.4 Solution Techniques

The primary goals of this study were (1) to show that the system can, under some conditions, become unstable, and (2) to demonstrate the use of the transient flow stabilization technique (see (1)) as a means of producing a system with an adequate degree of stability over the expected range of operating conditions. As shown in section 2.2, the model of the servo motor is a set of nonlinear, algebraic/differential equations. Further examination indicates that the model includes four independent energy storage effects; i.e. the model is fourth order. Two methods of solution were chosen to reach the goals set: (1) linearized analysis via digital simulation and Routh criterion, (2) nonlinear time domain analysis via digital computer simulation.

It is well known that a model linearized around some chosen steady-state operating point might not show the same behaviour in the neighbourhood of another point. But, linearized analysis can serve to enhance qualitative understanding of system behaviour, and to provide a reference solution against which the truer nonlinear solution can be compared. Therefore, the linearized study presented in Chapter III is intended only as a qualitative guide to the more complete study described in Chapter IV, on which the main conclusions related to objective (1) are built.

CHAPTER III

LINEARIZED SYSTEM ANALYSIS

3.1 Introduction

This chapter presents results of studies using a linearized version of the system equations. The purpose was to obtain a qualitative understanding of the system behaviour when all variables are confined to small changes, and thereby build a basis for a more complete non-linear study. Two analyses were done. The first analysis was a time-domain step response via digital computer simulation. Also, a Routh absolute stability analysis was conducted.

3.2 Derivation of the Linearized Set of Equations

The set of equations developed in section 2.2 can be linearized by considering small variations of all variables about an initial steady-state operating point. The linearized set of equations is presented below.

The force balance on the relay piston becomes

$$(\Delta P_1)\pi(d_2^2 - d_1^2)/4 = k_1\ (\Delta x_1). \tag{3.1}$$

The force balance on the prestressed spring (10) is

$$(\Delta P_4)\ A_2 = k_2\ (\Delta x_2 - \Delta x_4). \tag{3.2}$$

The flow through the capillary orifice becomes

$$(\Delta P_4) = \frac{128\ \mu\ L_4}{\pi\ d_4^4} (\Delta Q_4) \{ \ 1 + \frac{0.285\ \rho\ Q_{4i}}{\pi\ \mu\ \ L_4} \}. \tag{3.3}$$

Assuming that all orifices are operated within their underlap region and the radial clearances $c_{r1} = c_{r2} = 0$, the sharp edged orifice flow equations become

$$(\Delta Q_5) = c_d \pi w_1 \sqrt{2/\rho} \sqrt{P_{4i}} \ (\Delta x_2 - \Delta x_1) + \frac{(x_{2i} - x_{1i})}{2\sqrt{P_{4i}}} (\Delta P_4)$$

$$= a_3 \ (\Delta x_2 - \ \Delta x_1 \ ) + b_3 \ (\Delta P_4) \tag{3.4}$$

$$(\Delta Q_3) = c_d \pi d_6 \sqrt{2/\rho} \sqrt{P_2 - P_{3i}} \ (\Delta x_2 - \Delta x_4) - \frac{(x_{2i} - x_{4i} + U)}{2 \sqrt{P_2 - P_{3i}}} (\Delta P_3)$$

$$= a_1 \ (\Delta x_2 - \ \Delta x_4 \ ) - b_1 \ (\Delta P_3) \tag{3.5}$$

$$(\Delta Q_6) = c_d \pi d_6 \sqrt{2/\rho} \sqrt{P_{3i}} \ (\Delta x_4 - \Delta x_2) + \frac{(x_{4i} - x_{2i} + U)}{2 \sqrt{P_{3i}}} (\Delta P_3)$$

$$= a_2 \ (\Delta x_4 - \ \Delta x_2 \ ) + b_2 \ (\Delta P_3). \tag{3.6}$$

Here, the coefficients a_1, a_2 and a_3 are the flow displacement sensitivities and b_1, b_2 and b_3 are the flow pressure sensitivities.

From section 2.2, the flow continuity and compressibility equations can be combined and linearized to yield

$$\Delta Q_7 = \ \Delta Q_3 - \ \Delta Q_6 \tag{3.7}$$

$$\Delta Q_7 = \frac{\upsilon_{3i}}{\beta} \frac{d}{dt} (\Delta P_3) + \frac{\pi d_3^2}{4} \frac{d}{dt} (\Delta x_3) + c_3 (\Delta P_3) \tag{3.8}$$

$$\Delta Q_4 = \frac{\upsilon_{4i}}{\beta} \frac{d}{dt} (\Delta P_4) + A_2 \frac{d}{dt} (\Delta x_2) + \Delta Q_5 \; . \tag{3.9}$$

Finally, the force balance on the main piston and the feedback relation become

$$M (\Delta \ddot{x}_3) + k_3 (\Delta x_3) = \frac{\pi \, d_3^2}{4} (\Delta P_3) \tag{3.10}$$

$$(\Delta x_4) = \alpha \, (\Delta x_3) . \tag{3.11}$$

Here, M stands for the combined massed ($m_3 + m_e$), the damping on the main piston is neglected and the external load force (Figure 3) is approximated by a linear spring-like force.

3.3 A Closed Loop Transfer Function

The linearized set of equations presented in section 3.2 can be used to derive a closed-loop transfer function for the servo system. Appendix B contains this derivation; a summary of the results is included here.

The closed-loop transfer function is of fourth order with no numerator dynamics and is of the following form:

$$\frac{\Delta x_3}{\Delta P_1} = \frac{K}{a_o \, s^4 + a_1 \, s^3 + a_2 \, s^2 + a_3 \, s + a_4} \tag{3.12}$$

where the coefficients K and a_i are functions of the system parameters and the chosen initial steady-state operating point. Appendix B presents the transfer function with numerical values, corresponding to a steady-state operating point of $x_3 = 4.0$ in. Referring to Figure 3 it is clear that this point represents a condition close to the mid-range of operation of the servo in terms of the value of x_3. It corresponds to a rather high value of the pressure under the main piston, P_3, and a worst condition with regard to the negative slope of the load curve. Therefore, it should be a fairly typical operating point and the inherent dynamics of the system should become clear by focusing the analysis on this operating range.

The closed loop transfer function presented above can be used to conduct a Routh stability test on the system, which will give some information as to the system absolute stability for a given set of parameters. This analysis is presented in section 2.5.

Appendix B also contains a block diagram representing the system in its linearized, closed-loop form.

3.4 Computer Simulation of the Linearized System

A computer simulation of a dynamic system involves calculating the system repsonse to a certain input over a fixed period of time. A number of numerical integration methods are available to perform this task. In most cases, it is neccessary to reformulate the system equations as a set of first order, differential equations. A set of state variables can be defined as follows:

$$y_1 = x_3$$

$$y_2 = \dot{x}_3 = \dot{y}_1$$

$$y_3 = P_3$$

$$y_4 = P_4$$

where the y's are the state variables and x_3, P_3 and P_4 are the dynamic variables from the mathematical model.

The equations presented in section 3.2 can be combined to yield a set of first order differential equations in state variable form. From Equation (3.10), the main piston acceleration can be written as

$$\Delta\ddot{x}_3 = M^{-1} \frac{\pi d_3^2}{4} (\Delta P_3) - k_3 (\Delta x_3). \tag{3.13}$$

Combining Equation (3.8) with (3.7), (3.5), (3.6) and (3.2) yields

$$\Delta\dot{P}_3 = \frac{\beta}{\upsilon_{3i}} \left[\frac{A_2}{k_2} (a_1+a_2)(\Delta P_4) - (b_1+b_2+c_3)(\Delta P_3) - \frac{\pi d_3^2}{4} (\Delta\dot{x}_3) \right] \tag{3.14}$$

In a similar manner, combining Equations (3.1), (3.2), (3.3), (3.4), (3.9) and (3.11) gives

$$\Delta\dot{P}_4 = \left(\frac{\upsilon_{4i}}{\beta} + \frac{A_2}{k_2} \right)^{-1} \left[\frac{a_3(d_2^2 - d_1^2)}{4\, k_1} (\Delta P_1) - \left\{ \frac{a_3 A_2}{k_2} + b_3 + \frac{1}{a_4} \right\} (P_4) \right.$$

$$\left. a_3\, \alpha(\Delta x_3) - A_2\, \alpha(\Delta\dot{x}_3) \right] \tag{3.15}$$

where a_4 has been used to represent the rather lengthy constant from equation (3.3), or

$$a_4 = \frac{128\, \mu\, L_4}{\pi\, d_4^4} \left(1 + \frac{0.285\, \rho\, Q_{4i}}{\pi\, \mu\, L_4} \right). \tag{3.16}$$

Equations (3.13) to (3.15) along with the state variable definitions represent a system of four, linear first order differential equations, of the form

$$\begin{aligned} \dot{y}_1 &= f_1(y_2) \\ \dot{y}_2 &= f_2(y_1, y_3) \\ \dot{y}_3 &= f_3(y_2, y_3, y_4) \\ \dot{y}_4 &= f_4(y_1, y_2, y_4, P_1) \end{aligned} \qquad (3.17)$$

where P_1 is the control pressure input. These equations are all coupled and have to be solved simultaneously.

One of the most widely used numerical methods to solve a system of equations of this kind is the Runge-Kutta integration method. The so-called fourth-order Runge-Kutta method was chosen for this study, because it is considered to be fairly accurate and well behaved. Disadvantages of this method are that it requires the evaluation of $\dot{y}_n$ four times for each integration step, and the error involved in the numerical approximation cannot be analytically determined. The method is not very economical in computer time, but it is quite accurate. It is also relatively simple to implement on the computer and does not require any higher order derivatives to be evaluated.

A Fortran IV program was developed to simulate the system of equations (3.17); a source listing is contained in Appendix D. The linearized equations contain several steady-state parameter values, so a steady-state solution was programmed (Appendix C) and combined with the integration program to supply the neccessary values. The simulation program is set up so that the user can choose the steady-state operating point at which the system starts, and the program will calculate the neccessary parameters.

It was decided to determine the system response to a step input in control oil pressure P_1, as this is the most severe test the system could go through. In this test the input contains all frequency components. In order for the computer simulation to be valid, a small enough input step has to be chosen, so that all variables will only go through relatively small changes. This will always be a matter of estimation. In this case, the orifice opening in the relay piston can be used as a reference, that is the quantity $x_2 - x_1$. Figure 5 shows the orifice configuration formed by the relay plunger (4) and the relay piston (6) at an initial steady-state point. In order to stay within a linear operation, the orifice should not be closed; $x_2 - x_1 > 0$. Now, using the results from the steady-state calculations, a value of P_1 can be estimated which is consistent with these limits. Using Equation (3.1), which gives

$$\Delta P_1 = \frac{4\ k_1}{d_2^2 - d_1^2} (\Delta x_1)$$

and the fact that at a steady-state point of $x_3 = 4.0$ in,

$$x_1 = 4.141674 \text{ in. and}$$

$$x_2 = 4.179339 \text{ in.}$$

then

$$\Delta P_1 = \frac{4 \cdot 30}{(2.494^2 - .996^2)} (4.179339 - 4.141674) = 0.275 \text{ psi.}$$

Therefore, an appropriate value for the amplitude of the step input is $\Delta P_1 = 0.27$ psi.

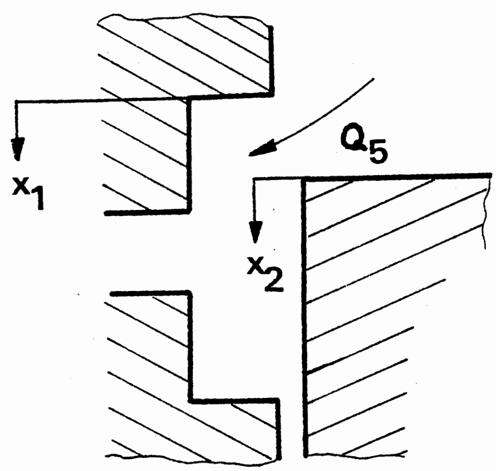


Figure 5. Orifice Configuration.

The simulation program utilizes an already developed plotting subroutine to display the time response in the four dynamic variables. If the user runs the program interactively at a Tektronix graphics terminal, a plotted output can be obtained. In any case, an output table is written into a data set, listing the time history of each of the four dynamic variables.

It is always a difficult task to estimate how small the integration time steps need to be for a system with dynamic characteristics that are barely known. In this system, examination of the various components of the system indicates that the load and the volume of oil under the main piston will likely contribute the dominant dynamic characteristics. A quick estimate of the associated natural frequency will give an idea of the order of magnitude of the required time step for integration.

If the oil under compression below the main piston is considered as a hydraulic spring, the spring constant is

$$k_s = \frac{\beta A_3^2}{\upsilon_{3i}} = \frac{110{,}000 \cdot 78.38^2}{344.0} = 1.965 \cdot 10^6 \text{ lb/in.}$$

From the load curve (Figure 3) the linearized spring constant of the load is found to be

$$k_3 = -\ 3{,}450 \text{ lb/in.}$$

Using these spring constants, which are coupled in parallel, a natural frequency can be found, which likely dominates the system response. It becomes

$$\omega_n = \sqrt{\frac{k_{eq}}{M_{eq}}} = \sqrt{\frac{1.965 \cdot 10^6 - 3{,}450}{0.303 + 1.3656}} = 1.08 \cdot 10^3 \text{ rad/sec.}$$

If this frequency content were to be noticed in the response, a time step smaller than the period corresponding to this frequency would have to be used. This period is

$$T = \frac{2\pi}{1.08 \cdot 10^3} = 5.8 \text{ millisec.}$$

Therefore, a time step should be chosen which is at least an order of magnitude less than 5.8 msec.

3.5 Simulation Results

Some system parameters are not known exactly because of manufacturing tolerances or because they are difficult to estimate in real cases. The simulation program provided an ideal tool to determine the effect that these parameters have on system stability. The tolerance parameters are mainly the valve underlap U and the radial clearances. As noted before, all radial clearances were considered zero in the derivation of the linearized orifice flow equations, so their direct influence on the system behaviour will not be determined in this part of the study. Only tolerances in the underlap U of the main orifice between the relay plunger and the bushing will be considered here.

Another variable, which is a very important factor in the dynamic behaviour of the system is the fluid bulk modulus, β. This parameter strongly affects the speed of response, and to a lesser extent, it affects the damping or degree of stability of the system. Fluid bulk modulus drastically decreases as the amount of air entrained in the fluid increases. In any real system there will always be a certain amount of air present, if only due to the action of the pump. In this study, response was determined for a high value of β, corresponding to no air content in the oil, as well as a lower value, which experience has shown is more realistic (9, page 18).

Results of the simulation studies are in the following pages. A step input of P_1 = 0.27 psi. was used in all cases shown. The steady-state operating point chosen as a starting point for the simulations was in all cases $x_3 = 4.0$ in. Referring to Figure 3, this operating point is within the very steep, negative slope region on the load curve; that is, the load force (F_{ext}) changes by approximately -3,450 lb for one inch increase in x_3. This range is expected to be the least stable operating range due to this high, negative spring like load implied in the model. Also, since all radial clearances are neglected, the model will predict somewhat less damping than should be present in the actual system.

Typical step response plots are presented in Figures 6 through 9. Figure 6 shows the output displacement x_3 as a function of time in response to a step input in P_1. This response is for a case of high value of β and a rather large valve underlap U. The case of $\alpha = 0.32$ is consistent with the value observed by K. N. Reid in an actual field installation (after overhaul). The x_3 response is overdamped with a rise time of about 0.041 sec. (using 95% of final value as a reference for defining 'rise time'). There is a slight indication of some higher order frequencies superimposed upon the overall response, and those can be seen more clearly in the plot of $\dot{x}_3$ in Figure 7 and P_3 in Figure 8. The plot of $\dot{x}_3$ shows that the velocity of the piston goes to zero at steady state, as this is a position control system.

Figures 8 and 9 show the pressure under the main piston (P_3) and the pressure above the relay plunger (P_4) respectively. These variables, especially P_3, will tend to magnify any higher order frequency content of the response. This is due to the fact that P_3 is proportional to the acceleration of the system position output, and deriva-

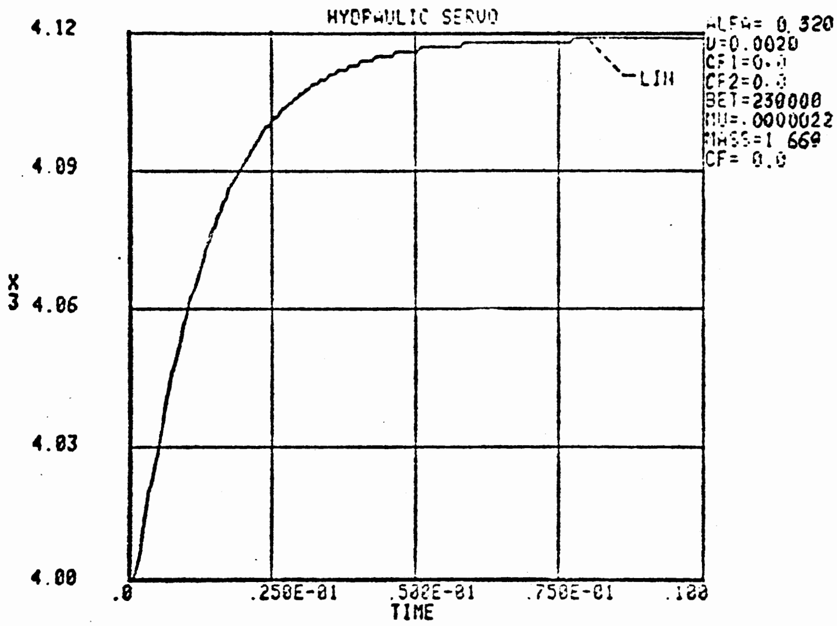


Figure 6. Linearized Analysis. Response in x_3 for $U = 0.002$ in, $\beta = 230{,}000$ psi, $\alpha = 0.32$.

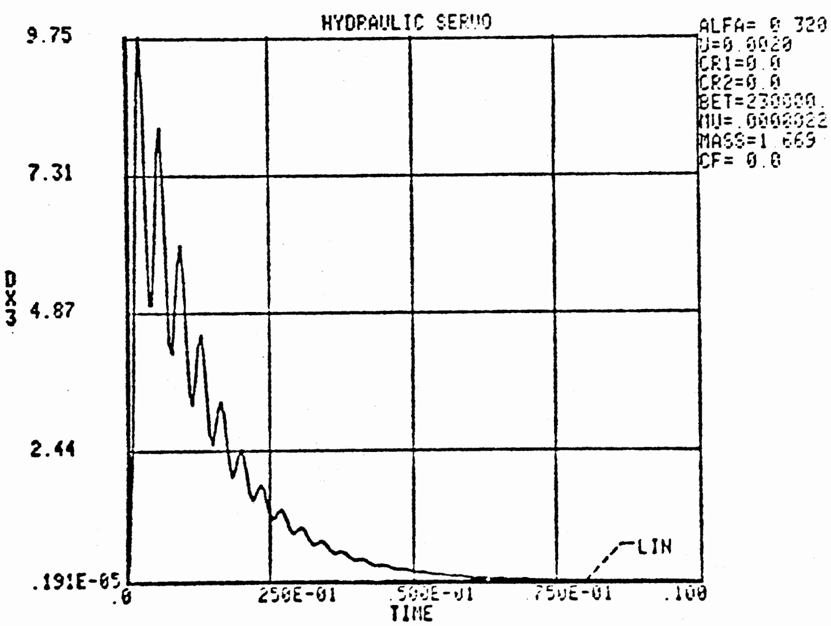


Figure 7. Linearized Analysis. Response in $\dot{x}_3$ for $U = 0.002$ in, $\beta = 230{,}000$ psi, $\alpha = 0.32$.

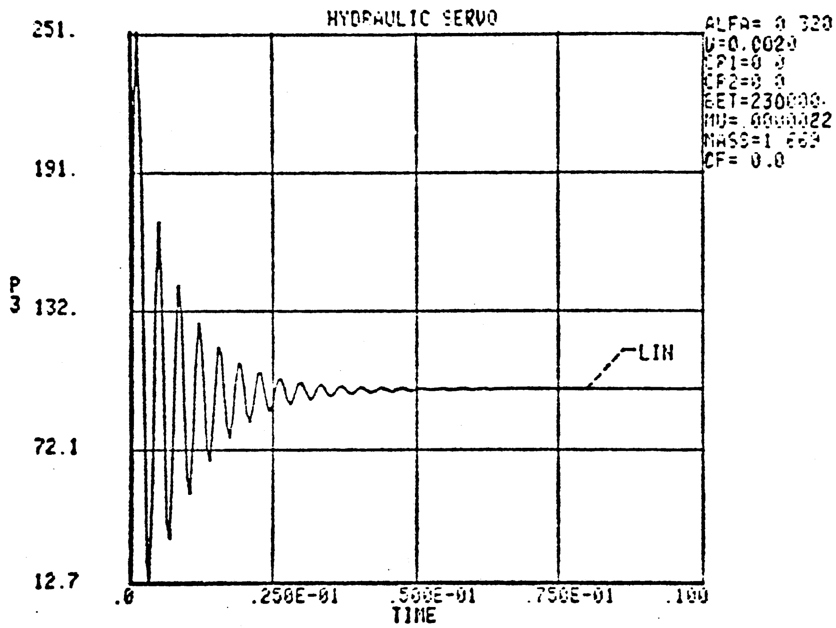


Figure 8. Linearized Analysis. Response in P_3 for U = 0.002 in, β = 230,000 psi, α = 0.32.

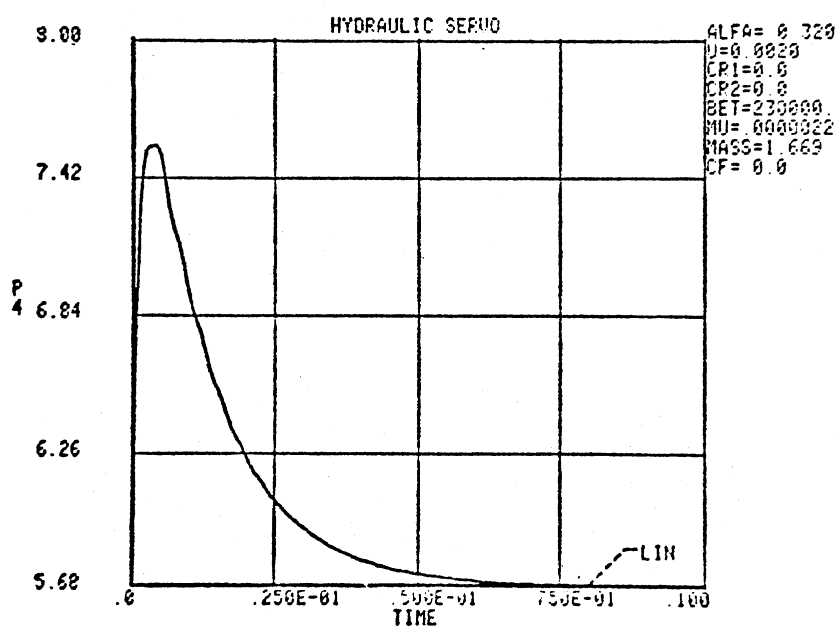


Figure 9. Linearized Analysis. Response in P_4 for U = 0.002 in, β = 230,000 psi, α = 0.32.

tive signals tend to magnify higher order components. The response in P_3 shows the effect of two complex-conjugate, lightly damped poles of the system, that the response in x_3 does not reveal.

The simulation results represented in Figure 10 are for a case with less valve underlap than that used in the case summarized above. Here, U = 0.001 in., which is within the manufacturer's tolerance of 0 to 0.002 in. This simulation shows that the system is unstable.

Figure 11 shows the results of a simulation where the value of fluid bulk modulus has been lowered to the more realistic value of 110,000 psi. Again, the result is an unstable system, and worse behaviour than for the one with U = 0.001 in. This is a model, that is quite likely a reasonably true representation of the parameters in the real system, except for the absence of Coulomb friction in the load equation. Coulomb friction could enhance system stability; but the dependence on Coulomb friction to stabilize the system is normally considered poor design.

There remains uncertainty on which feedback cam was used in the original system (before overhaul). Figure 4 demonstrated that a cam with either one of two different slopes might have been mounted on the piston rod. The results given in Figures 6, 10 and 11 were obtained for the case of $\alpha = 0.32$ (i.e. the lowest feedback gain cam). From Figure 4 it is seen that for the other possible cam, $\alpha = 0.45$ and $x_3^* = 3.38$ in. for the operating range around $x_3 = 4.0$ in. The result of the simulation with this higher feedback gain cam is shown in Figure 12. The values of β and U which produce the most stable system were used, in order to more clearly see the effect of α on the system

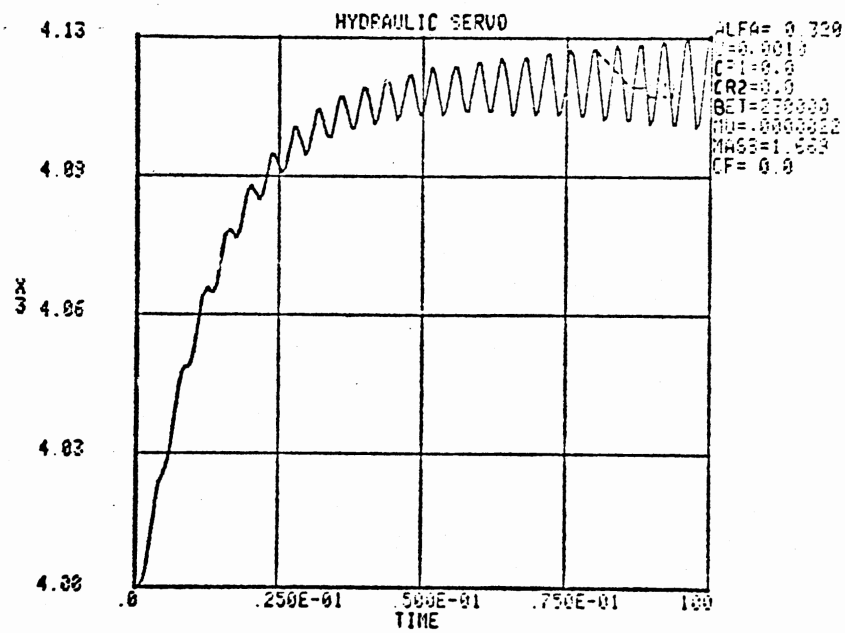


Figure 10. Linearized Analysis. Response in x_3 for U = 0.001 in, β = 230,000 psi, α = 0.32.

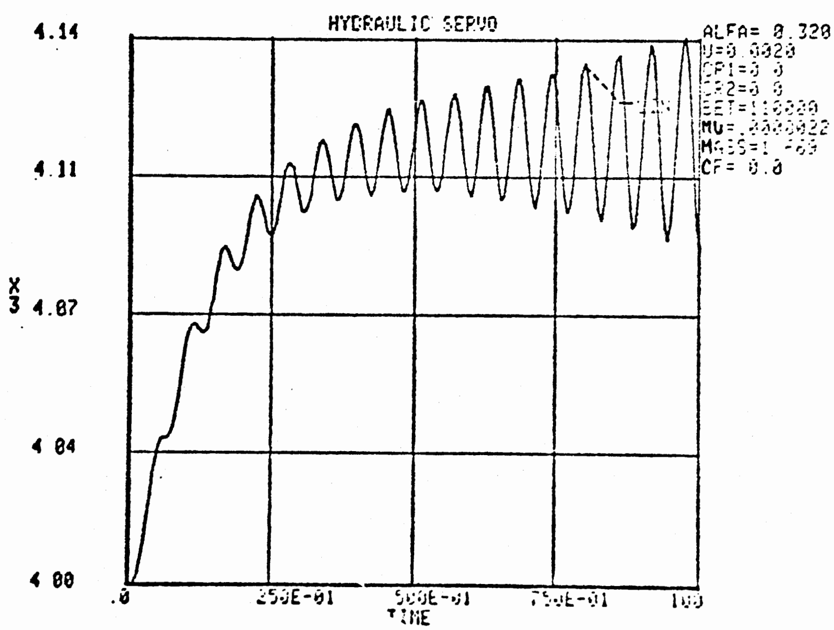


Figure 11. Linearized Analysis. Response in x_3 for U = 0.002 in, β = 110,000 psi, α = 0.32.

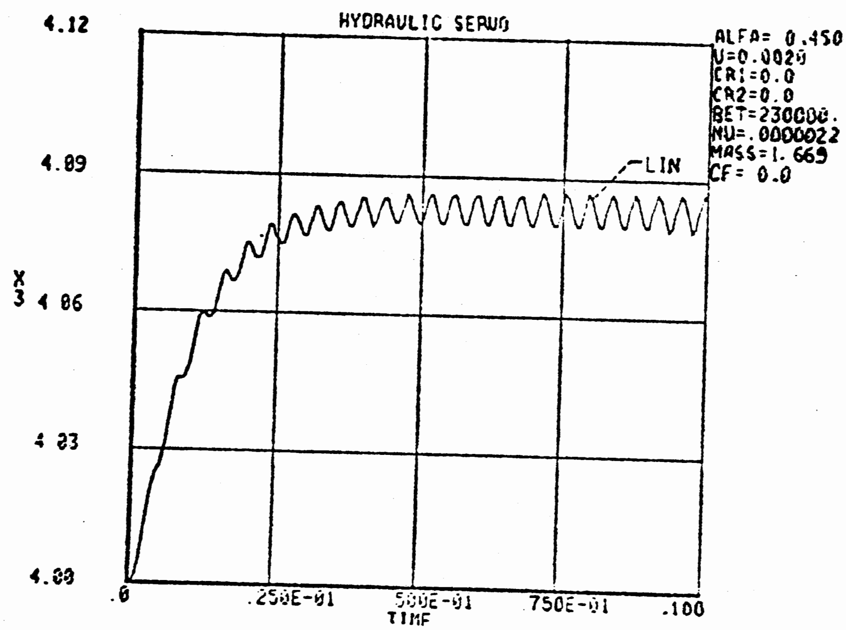


Figure 12. Linearized Analysis. Response in x_3 for U = 0.002 in, β = 230,000 psi, α = 0.45.

performance. In this case, the system is unstable. It is clear that if the system had been installed originally with a high-gain cam (and this is uncertain), it would have been less stable than with the low-gain cam.

The results of the computer simulation of the linearized system model can then be summarized as shown in Table III.

TABLE III

SUMMARY OF THE RESULTS OF THE COMPUTER SIMULATION OF THE LINEARIZED SYSTEM MODEL

U	β	α	Results
0.002 in	230,000 psi	0.32	Stable
0.001 in	230,000 psi	0.32	Unstable
0.002 in	110,000 psi	0.32	Unstable
0.002 in	230,000 psi	0.45	Unstable

It is now neccessary to determine what effect the parameters left out of the model have on the system performance, so that its stability can be assessed with higher confidence. Chapter IV will present results of a study of a more accurate nonlinear model of the system.

3.6 Routh Test of System Stability

The Routh criterion can be used as an independent check of system stability to validate the results presented in section 3.4. The Routh stability test gives information about the absolute stability of a closed-loop system, based on characteristics of the transfer function. That is, it provides a simple method to find if one or more roots of the system characteristic equation have positive real parts. However, the criterion will not yield any information concerning degree of stability.

The system closed loop transfer function was presented in section 3.3, where it was seen that the coefficients are dependent on a chosen initial steady-state operating point. An appropriate steady-state point is chosen in the mid-range of operation of the servomotor, $x_3 = 4.0$ in., and the resulting transfer function for a bulk modulus of 110,000 psi is (see Appendix B):

$$\frac{\Delta x_3}{\Delta P_1} = \frac{4.662 \cdot 10^6}{155.0 \cdot 10^{-6} s^4 + 142.9 \cdot 10^{-3} s^3 + 191.4 \cdot s^2 + 176.1 \cdot 10^3 s + 10.89 \cdot 10^6} . \tag{3.18}$$

The denominator is of the form

$$a_o\, s^4 + a_1\, s^3 + a_2\, s^2 + a_3\, s + a_4 .$$

The Routh method involves setting up a matrix of parameters as follows:

$$
\begin{array}{llll}
s^4 : & a_o & a_2 & a_4 \\
s^3 : & a_1 & a_3 & a_5 \\
s^2 : & b_1 & b_2 & \\
s^1 : & c_1 & c_2 & \\
s^0 : & d_1 & &
\end{array}
$$

where

$$b_1 = \frac{a_1a_2 - a_oa_3}{a_1}$$

$$b_2 = \frac{a_1a_4 - a_oa_5}{a_1}$$

$$c_1 = \frac{b_1a_3 - a_1b_2}{b_1}$$

$$c_2 = \frac{b_1a_5 - a_1b_3}{b_1}$$

$$d_1 = \frac{c_1b_2 - b_1c_2}{c_1} .$$

The criterion states that if in the first column (a_o, a_1, etc.) there is a change of sign of the coefficients, then there is a root with a positive real part. The number of sign changes in the first column represents the number of roots with a positive real part.

In order to validate the results obtained from the computer simulation, the stability test will be applied for the two values of β in question. First, a value of β = 110,000 psi has been implemented into Equation (3.18). The Routh array then yields

s^4:	$155.0\ 10^{-6}$	191.44	$10.89\ 10^{6}$
s^3:	$142.0\ 10^{-3}$	$176.1\ 10^{3}$	0
s^2:	$-680.3\ 10^{-3}$	$10.89\ 10^{6}$	0
s^1:	$176.1\ 10^{3}$	0	
s^0:	$10.89\ 10^{6}$		

Since the coefficient b_1 has a negative sign, there are two changes of sign in the first column. This means that the system characteristic equation has two roots with positive real parts, and therefore is unstable. This confirms the results obtained from the computer simulation of the linearized system, which showed a response with increasing oscillations in x_3 for a value of $\beta = 110{,}000$ psi. (Figure 11).

For a fluid bulk modulus $\beta = 230{,}000$ psi. the following transfer function is obtained

$$\frac{\Delta x_3}{\Delta P_1} = \frac{4.662 \cdot 10^6}{74.56 \cdot 10^{-6} s^4 + 73.38 \cdot 10^{-3} s^3 + 191.44\ s^2 + 176.1\ s + 10.89 \cdot 10^6} \qquad (3.19)$$

The Routh array then becomes slightly different, due to the different parameter values in the first three terms in the denominator.

s^4:	$74.56\ 10^{-6}$	191.44	$10.89\ 10^{6}$
s^3:	$73.38\ 10^{-3}$	$176.1\ 10^{3}$	0
s^2:	12.50	$10.89\ 10^{6}$	0

s^1: $176.1 \cdot 10^3$ 0

s^0: $10.89 \cdot 10^6$

Here, there is no change of sign in the first column, indicating that this transfer function represents a stable system. This result is in accordance with results from the computer simulation (Figures 6 - 9).

It has been mentioned before that the lower value of fluid bulk modulus is recognized as a realistic estimate of this parameter. Therefore, based on the linearized system analysis, it appears that the actual system may have too low a degree of stability. Other parameters not included in the linearized study can have an important effect on the system dynamic performance. For example, valve leakage and Coulomb friction in the load can affect system degree of stability. Results of a more comprehensive study of the nonlinear system are presented in Chapter IV.

CHAPTER IV

NONLINEAR SYSTEM ANALYSIS

4.1 Introduction

Estimation of the system model derived in section 2.2 indicates that several potentially important nonlinear effects may govern the dynamic behaviour of the system. In order to more fully understand the system behaviour it is neccessary to analyze the model derived in section 2.2 more carefully using a technique that will accomodate all its important features.

There are several methods available for analyzing nonlinear control systems. The method of describing functions, which involves a quasi-linearization of the system being studied, is used to determine whether a system could exhibit limit cycle behaviour and under what conditions. Nyquist plots are used as a graphical aid in the analysis. This method works fairly well in many cases, but it is difficult to apply to systems with more than one nonlinear element.

Other methods for analyzing nonlinear control systems have been developed in resent years. The most important ones are based on Lyapunov's stability criterion and, like describing functions, use a graphical representation of the system as the means of assessing system stability. These graphical methods are all limited to systems having only one nonlinearity and with single input and single output.

There are some related methods that will handle systems with more than one nonlinearity, but they are tedious to apply and somewhat suspect in terms of accuracy.

It appeared in this study that digital computer simulation was the most appropriate and meaningful method of analysis. A dynamic simulation program, DYSIMP, written in Fortran IV, has been developed in the Center for System Science, at Oklahoma State University. It will simulate a large variety of dynamic systems, including any nonlinear characteristics. This program was chosen for the study in this thesis.

4.2 A Nonlinear Model for Computer Simulation

The dynamic simulation program, DYSIMP, is set up to solve numerically a set of ordinary differential equations. Mathematically, the statement of the problem it will solve can be written as follows:

$$Y(t) = F(Y, X, P, t)$$

subject to

$$Y(t_o) = Y_o$$

$$G(Y, X, P, t) = 0$$

in the range $t_o \leqslant t \leqslant t_f$, where

Y is a vector of dependent state variables

X is a vector of algebraic variables

P is a vector of constant parameters

t is the independent variable

F and G are vector functions defined by the system model.

DYSIMP has three initial value integration methods available for the solution of the stated problem. These are a 'crude' Euler method, a modified Euler method, and a fourth-order Runge-Kutta integration. These methods are all implemented to allow integration of the non-linear equations using a fixed integration step. The most accurate method is the fourth-order Runge-Kutta method; this method was used in this study. The nonlinear model of the servo system derived in section 2.2 must be put in the form required by DYSIMP. First, the equations that include time derivatives of a variable must be reformulated. From Equation (2.12)

$$M \ddot{x}_3 + C \dot{x}_3 + F_{ext} = \frac{\pi d_3^2}{4} P_3 \tag{4.1}$$

where F_{ext} is to be implemented using both the steady-state load characteristic from Figure 3 and a Coulomb friction effect. Equation (4.1) may be rewritten as two first order differential equations. Two state variables can be defined as

$$Y(1) = x_3$$

$$Y(2) = \dot{x}_3 = DY(1)$$

which will result in Equation (4.1) being written as required by DYSIMP.

From Equation (2.8) and the assumption that the change in fluid density, ρ, is small, but that the rate of change of ρ is large,

$$Q_7 = \frac{\upsilon_4}{\beta} \frac{d}{dt}(P_3) + \frac{\pi d_3^2}{4} \frac{d}{dt}(x_3) + c_3 P_3. \tag{4.2}$$

In the same way Equation (2.9) may be written as

$$Q_4 = \frac{\upsilon_3}{\beta} \frac{d}{dt}(P_4) + A_2 \frac{d}{dt}(x_2) + Q_5. \tag{4.3}$$

Two additional state variables, P_3 and P_4, may be defined:

$$Y(3) = P_3$$

$$Y(4) = P_4.$$

These state variables will be adequate for setting up the model on DYSIMP.

The other equations can be written as desired, by defining the algebraic variables X(i) in such a way to make the set of equations compatible with the program. The following convention is used:

$$X(1) = x_1$$

$$X(2) = x_2$$

$$X(3) = x_4$$

$$X(4) = Q_3$$

$$X(5) = Q_5$$

$$X(6) = Q_6$$

$$X(7) = Q_7.$$

The parameters P(i) are defined as neccessary to aid in expressing the equations in a simple form. Appendix E contains a more detailed description of the formulation and a source listing of the Fortran program written to simulate the system using DYSIMP.

Equations (2.4), (2.5) and (2.6) all have to be implemented with logical operations, because their form depends on the relative positions of the relay piston, the relay plunger and its bushing. There could possibly be flow through the orifices only due to the radial clearances, or the flow could be reversed due to changes in pressure difference. Equation (2.3) is a nonlinear equation in Q_4, or more specifically a second order polynomial in this variable. At the beginning of each time step, a value of Q_4 must be calculated based on the value of P_4 from the previous time step. Therefore, at the beginning of each time step, it is neccessary to solve a second order polynomial in the variable X(7). This can be implemented rather easily into DYSIMP using the subroutine XVAL, which is called on automatically at the beginning of each time step. XVAL is therfore ideal for solving Equation (2.3) for the flow Q_4.

The most involved part of the model to implement in the computer solution is the load function. The load force, F_{ext}, includes a steady-state load function and the Coulomb friction effect. The external load can thus be written as

$$F_{ext} = F_L + B \frac{\pi d_3^2}{4} \operatorname{sgn}(\dot{x}_3) \tag{4.4}$$

where B is the level of Coulomb friction in units of pressure. Figure 13 illustrates the Coulomb friction characteristic.

It is quite instructive to consider what happens physically due to the Coulomb friction between the main piston and its cylinder. First, assume that the piston is at rest at some point, and that there is a balance of steady-state forces acting on it. That is, the pressure P_3 acting to move the piston upward is balanced by an equal force from the steam valves tending to move it downward. The level of Coulomb friction can now be assumed to be 0 (see Figure 13). An increase in oil pressure P_3 would act to unbalance the forces acting on the piston, resulting in motion of the piston if there were no Coulomb friction. However, from Figure 13 it is apparent that the increase in pressure needs to be larger than $B(\pi/4)d_3^2$ in order for the piston to move. Therefore, until P_3 has increased enough to 'break' the piston loose, there will be a force balance on the piston given by

$$\frac{\pi d_3^2}{4} P_3 = F_L + F_{CF}.$$

Therefore, the piston will not have a velocity nor acceleration until the pressure P_3 has built up high enough to overcome the level of Coulomb friction.

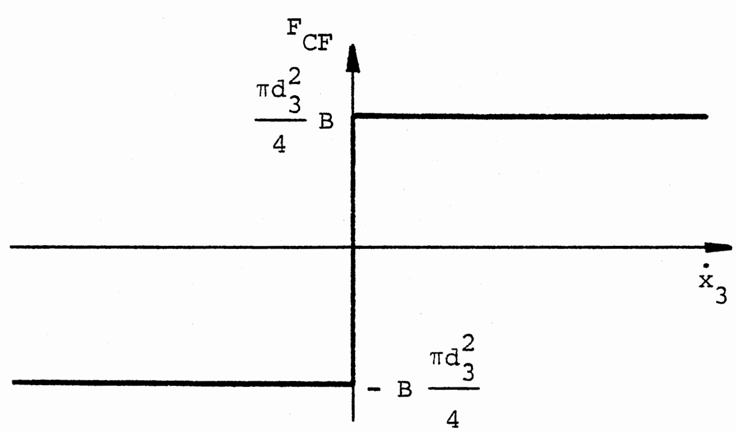


Figure 13. Coulomb Friction.

The program implementation has to detect whenever the velocity is equal to zero, or when it has passed through the zero point, and then provide the logic neccessary to describe this behaviour. Therefore, it is neccessary to retain information on the value of velocity from the previous time step, and to check on changes in sign. The Fortran implementation of this characteristic can be seen in the source listing in Appendix E.

The external steady-state load, F_L, from Figure 3 can be implemented most easily by a simple interpolation in a data table. Trying to fit an analytical function to the data represented in Figure 3 would not result in a very true representation of the actual curve. An interpolation routine is included in the program listing shown in Appendix E.

DYSIMP provides the user a choice of different output representations. The program lists the time change of all dynamic and algebraic variables as the user wishes, and prints all parameter values and initial and final values of the dynamic variables. It also produces printer plots of the dynamic variables, scaling them as needed, and has the capability to produce Calcomp plots of the same variables. The printer plots are a useful tool during the debugging and preliminary stages of using the program with a model, although they become somewhat clumsy for interpreting or presenting the solution. Therefore, printer plots were used during the development of the system model with DYSIMP, but the final solutions were obtained in graphics form, using Tektronix terminals available in the School of Mechanical and Aerospace Engineering. DYSIMP does not directly provide such representation, but by having the listed output from the program loaded directly into TSO data sets, it was relatively easy to write a short Fortran program to read this data and plot it on the Tektronix screen. Appendix F contains a listing of the program used for this purpose.

4.3 Results from a Study of the Nonlinear System Model

In the nonlinear model the following additional effects were included:

1. Valve radial clearances.
2. Variable volume under compression ($V_3 = f(x_3)$).
3. Nonlinear load force ($F_{ext} = f(x_3)$) and Coulomb friction in the load.

In the linear model, the load force was represented by a linear spring; also, the load Coulomb friction and the change in the volume under compression (but not the time rate of change) were both assumed to be negligible.

Step responses are presented in this section for the important cases listed in Table IV.

The first set of resutls, Figures 14 through 17, show the responses of all four dynamic variables to a small step input, or $\Delta P_1 = 0.27$ psi. There was no radial clearance and no Coulomb friction load for this case; also, the fluid bulk modulus, β, was 230,000 psi. These results should be compared with Figures 6 through 9 from the linear model analysis. From the plots of $\dot{x}_3$ and P_3 it is apparent that the linear model has somewhat less damping. This difference is

TABLE IV

SUMMARY OF RESULTS OF A STUDY OF THE NONLINEAR MODEL

U (in)	α	c_{r1} (in)	c_{r2} (in)	β (psi)	CF (psi)	Figures	Stability Result
0.002	0.32	0.0	0.0	230,000	0.0	14, 15, 15, 17	Stable
0.001	0.32	0.0	0.0	230,000	0.0	18	Stable
0.002	0.32	0.002	0.002	110,000	0.0	19, 20, 21	Stable, lightly damped
0.001	0.32	0.0	0.0	110,000	0.0	22, 23	Unstable
0.002	0.32	0.002	0.002	90,000	0.0	24, 25	Stable, oscillatory
0.001	0.32	0.0	0.0	110,000	5.0	26, 27, 28	Stable due to Coulomb friction
0.002	0.45	0.002	0.002	110,000	5.0	29, 30, 31	Stable due to Coulomb friction

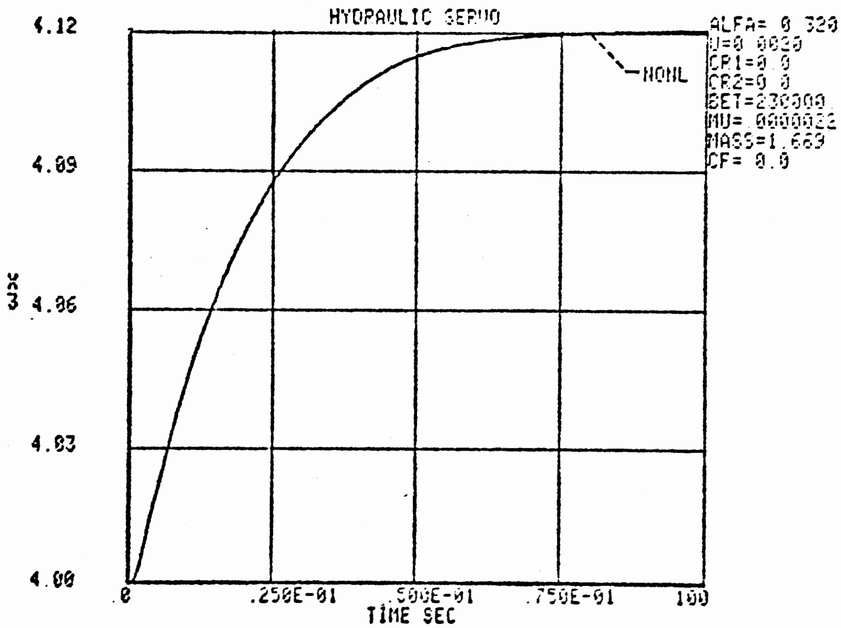


Figure 14. Nonlinear Study of Uncompensated System. x_3 for U = 0.002 in, β = 230,000 psi, F_{CF} = 0.0.

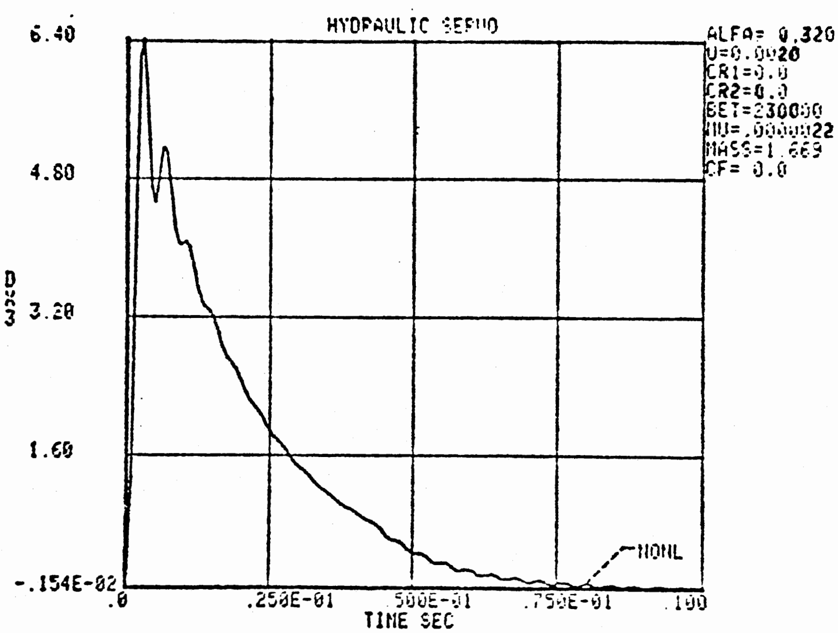


Figure 15. Nonlinear Study of Uncompensated System. $\dot{x}_3$ for U = 0.002 in, β = 230,000 psi, F_{CF} = 0.0.

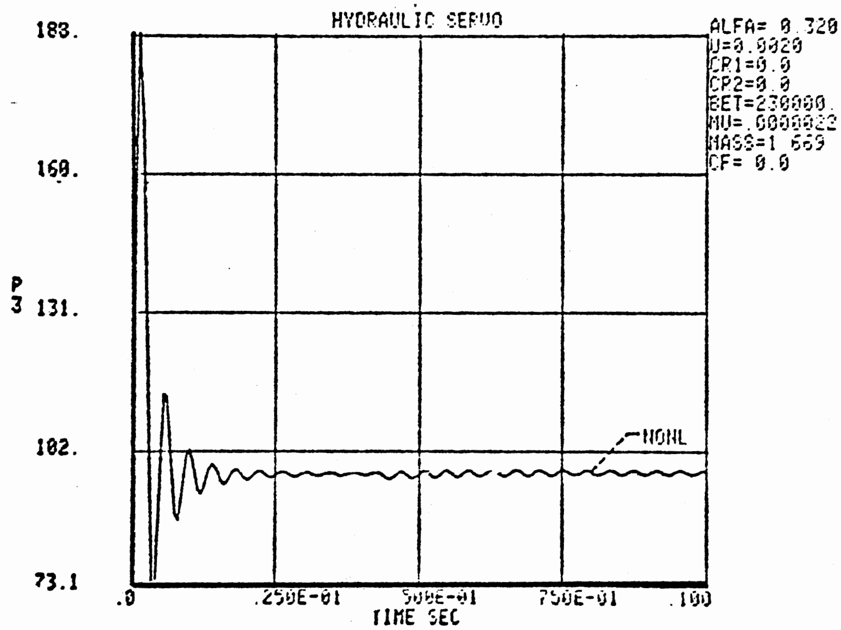


Figure 16. Nonlinear Study of Uncompensated System. P_3 for U = 0.002 in, β = 230,000 psi, F_{CF} = 0.0.

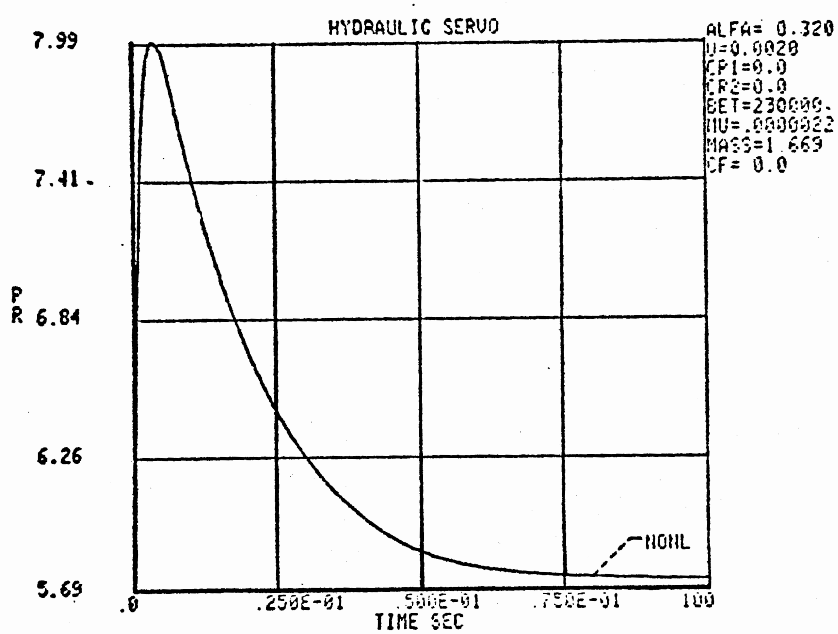


Figure 17. Nonlinear Study of Uncompensated System. P_4 for U = 0.002 in, β = 230,000 psi, F_{CF} = 0.0.

not surprising when the nature of the orifice characteristics is considered; that is the slope of the valve characteristics affects the damping and the 'average' slope is greater in the nonlinear case.

A comparison of Figures 18 and 10 shows some difference in damping between the linear and nonlinear models. The linear model is unstable in this case of small valve underlap, but the nonlinear model shows a stable repsonse. It can be concluded that the linear model predicts a less stable system than may actually exist.

The fluid bulk modulus also has a dramatic effect on the behaviour of the nonlinear model. Figures 19 through 21 show the results of a simulation where Coulomb friction was neglected, the valve underlap and radial clearances were at their 'nominal' values of 0.002 in. and β = 110,000 psi. The linear model was unstable for this case. The nonlinear model predicts that the system is stable, although the plots of $\dot{x}_3$ and P_3 show almost sustained oscillations; that is, this system has a very low degree of stability.

Figures 22 and 23 show results for reduced underlap (U=0.001 in.), zero radial clearances and negligible Coulomb friction. The system is stable, but very lightly damped for this case. Thus, with no Coulomb friction, the system with a realistic fluid bulk modulus (β= 110,000 psi.) would be on the verge of instability. This result is further underscored by the results shown in Figures 24 and 25, where underlap and radial clearances are 'nominal' again, but β is decreased to 90,000 psi. That is, if only one to two percent of air by volume was present in the system, the system could be unstable with underlaps within the manufacturing tolerances.

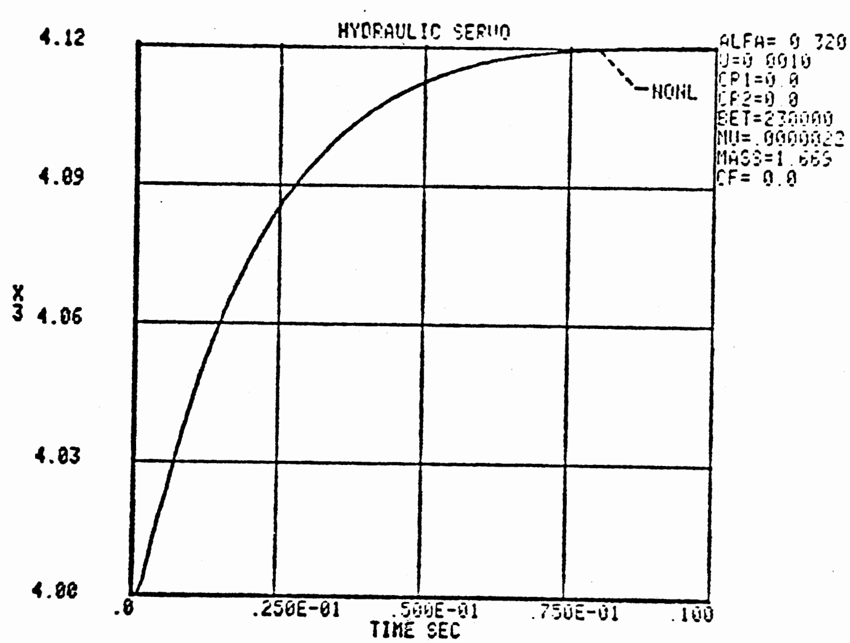


Figure 18. Nonlinear Study of Uncompensated System. x_3 for U = 0.001 in, β = 230,000 psi, F_{CF} = 0.0.

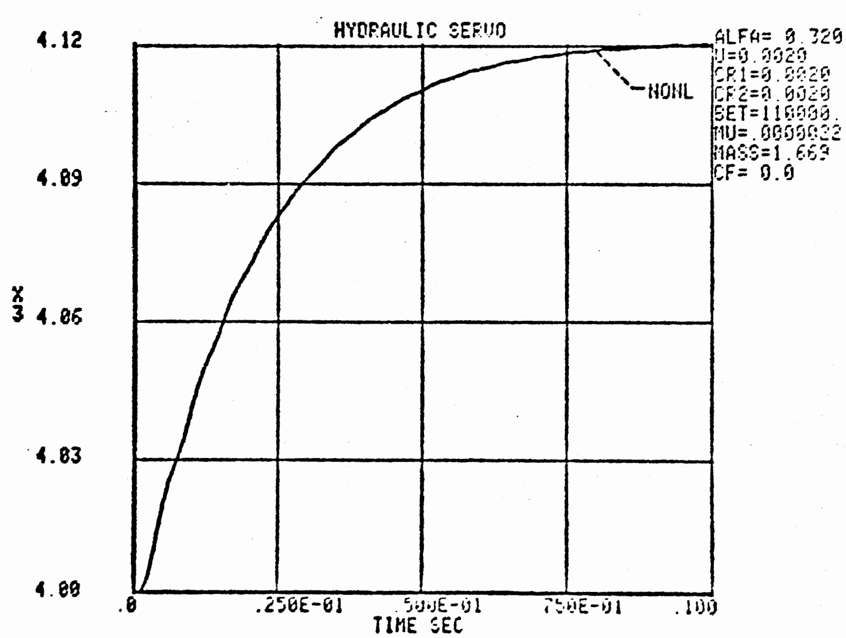


Figure 19. Nonlinear Study of Uncompensated System. x_3 for U = 0.002 in, β = 110,000 psi, $c_{r1} = c_{r2}$ = 0.002 in, F_{CF} = 0.0.

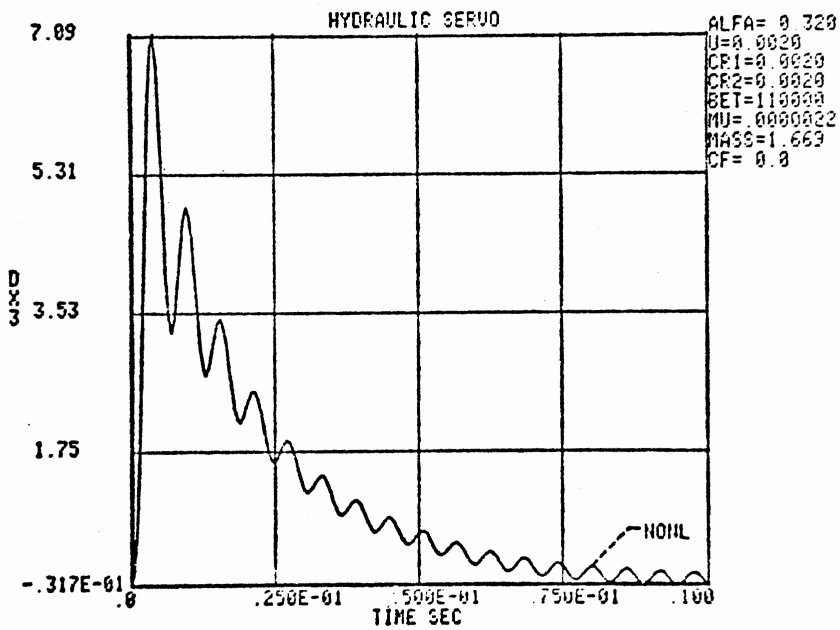


Figure 20. Nonlinear Study of Uncompensated System. $\dot{x}_3$ for U = 0.002 in, β = 110,000 psi, $c_{r1} = c_{r2}$ = 0.002 in, F_{CF} = 0.0.

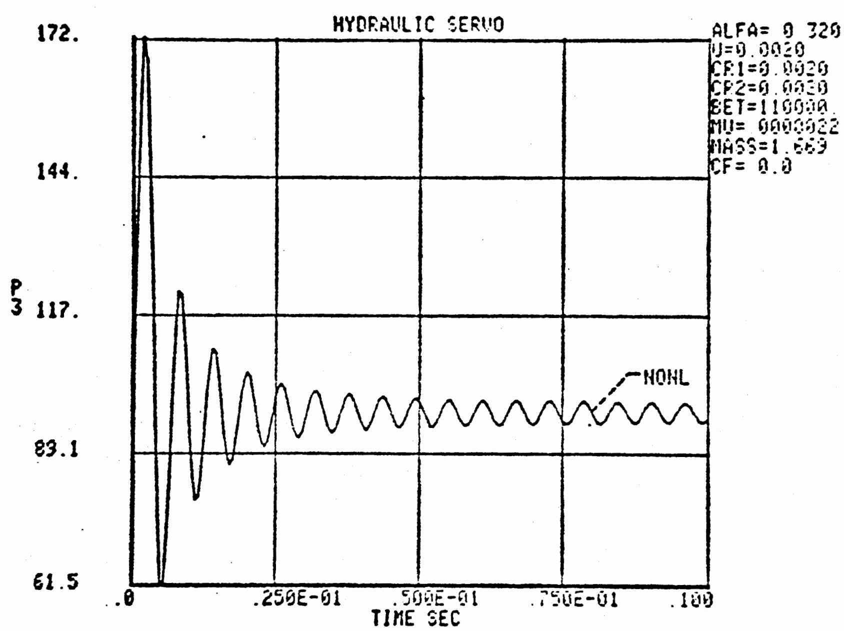


Figure 21. Nonlinear Study of Uncompensated System. P_3 for U = 0.002 in, β = 110,000 psi, $c_{r1} = c_{r2}$ = 0.002 in, F_{CF} = 0.0

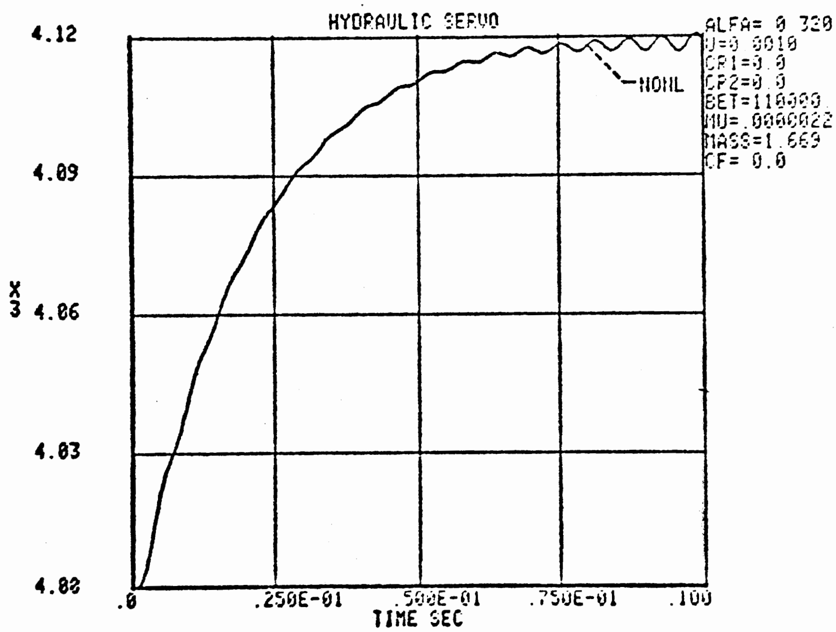


Figure 22. Nonlinear Study of Uncompensated System. x_3 for U = 0.001 in, β = 110,000 psi, $c_{r1} = c_{r2}$ = 0.0, F_{CF} = 0.0.

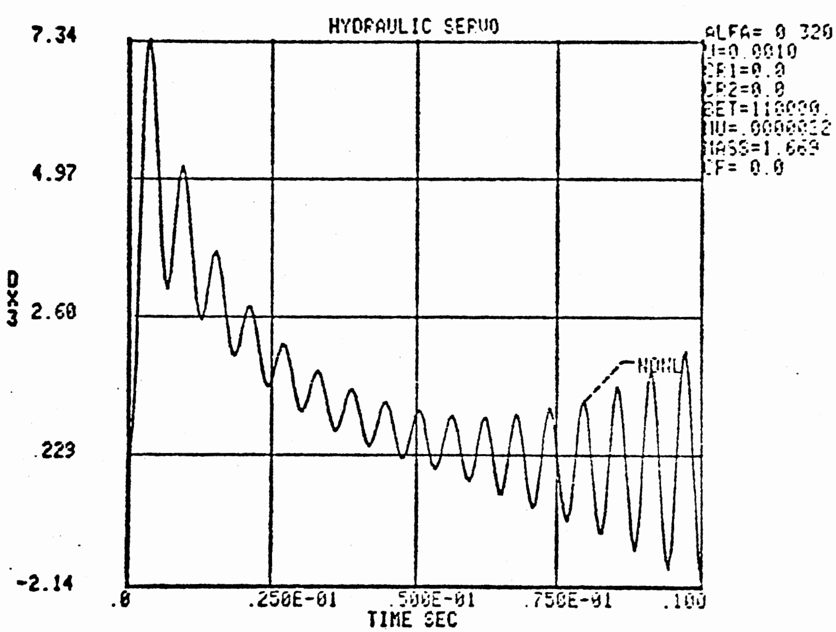


Figure 23. Nonlinear Study of Uncompensated System. $\dot{x}_3$ for U = 0.001 in, β = 110,000 psi, $c_{r1} = c_{r2}$ = 0.0, F_{CF} = 0.0.

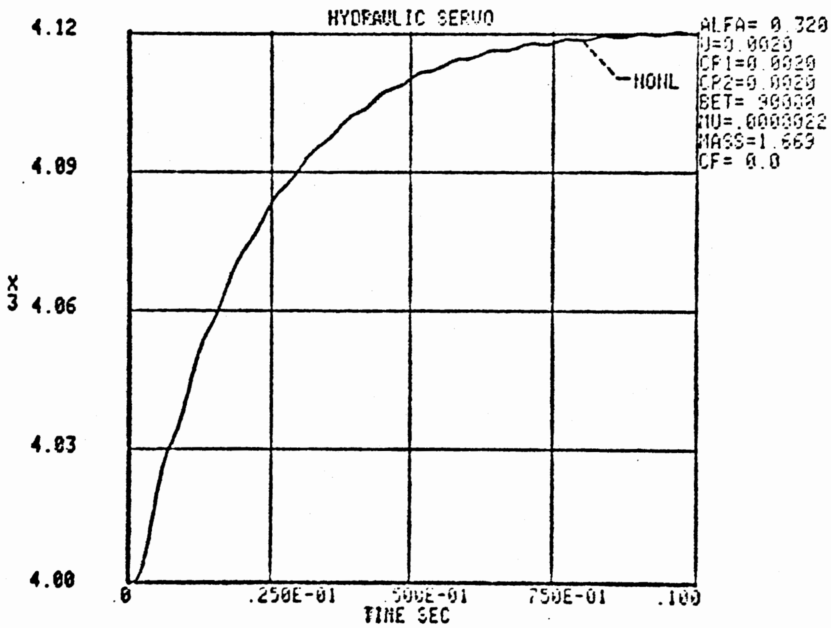


Figure 24. Nonlinear Study of Uncompensated System.
x_3 for U = 0.002 in, β = 90,000 psi,
$c_{r1} = c_{r2} = 0.002$ in, $F_{CF} = 0.0$.

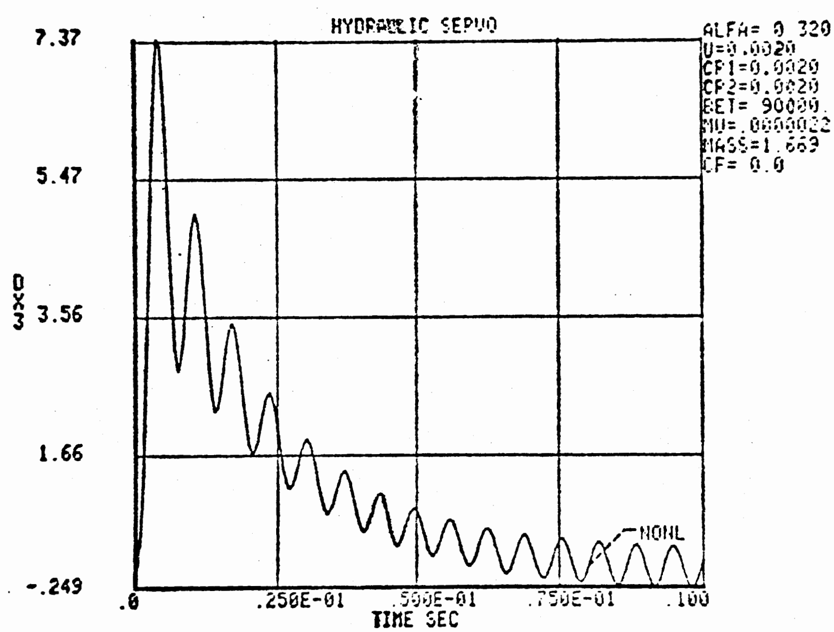


Figure 25. Nonlinear Study of Uncompensated System.
$\dot{x}_3$ for U = 0.002 in, β = 90,000 psi,
$c_{r1} = c_{r2} = 0.002$ in, $F_{CF} = 0.0$.

Coulomb friction in the load also can have a dramatic effect on the system performance. Figures 26 through 28 show the results for the same configuration as shown in Figures 22 and 23, now with a Coulomb friction level of B = 5.0 psi. added to the model. The results predict a stable system. Thus, sufficient Coulomb friction can make an otherwise unstable system stable.

It is not known with certainty which of the two possible feedback cams were present in the system before overhaul. All of the results presented above were obtained for the low gain cam believed to exist in the field system after servomotor overhaul. The 'double-slope' cam described by Figure 4 (Curve A) may have existed in the servomotor before overhaul. This latter cam produces two values of feedback gain (α) as follows:

$x_3 > 2.9$ in.: $\alpha = 0.45$ and $x_3^* = 3.38$ in.

$x_3 < 2.9$ in.: $\alpha = 0.90$ and $x_3^* = 2.10$ in.

Before overhaul, the system was observed to be very oscillatory (and perhaps unstable) at certain times. After overhaul, the system was observed to be well behaved (i.e. well damped). Figures 29 through 31 show simulation results for the value of α of 0.45, β = 110,000 psi. and Coulomb friction included. The system is stable for this case; but without Coulomb friction the system would be unstable.

Based on the results of the nonlinear simulations, it appears that system stability depends on the presence of Coulomb friction and/ or a high bulk modulus of elasticity. It is known that Coulomb friction is a somewhat erratic phenomena, and is not as predictable as viscous

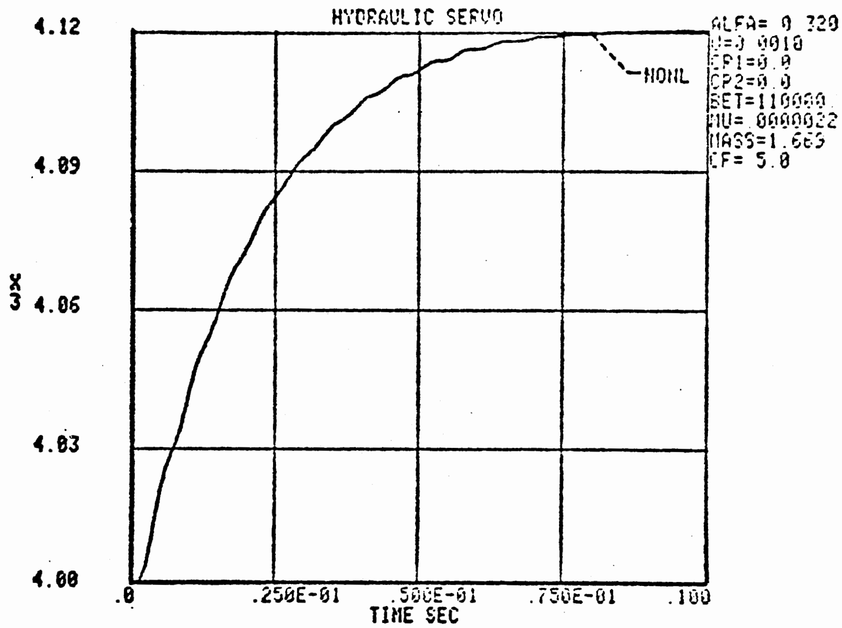


Figure 26. Nonlinear Study of Uncompensated System. x_3 for U = 0.001 in, β = 110,000 psi, $c_{r1} = c_{r2} = 0.0$, $F_{CF} = 5.0$ lb/in^2.

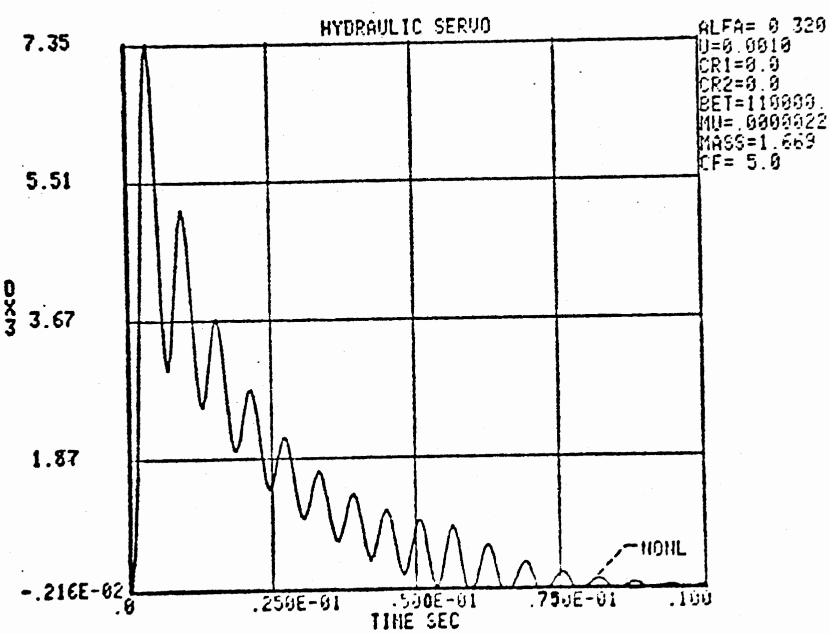


Figure 27. Nonlinear Study of Uncompensated System. $\dot{x}_3$ for U = 0.001 in, β = 110,000 psi, $c_{r1} = c_{r2} = 0.0$, $F_{CF} = 5.0$ lb/in^2.

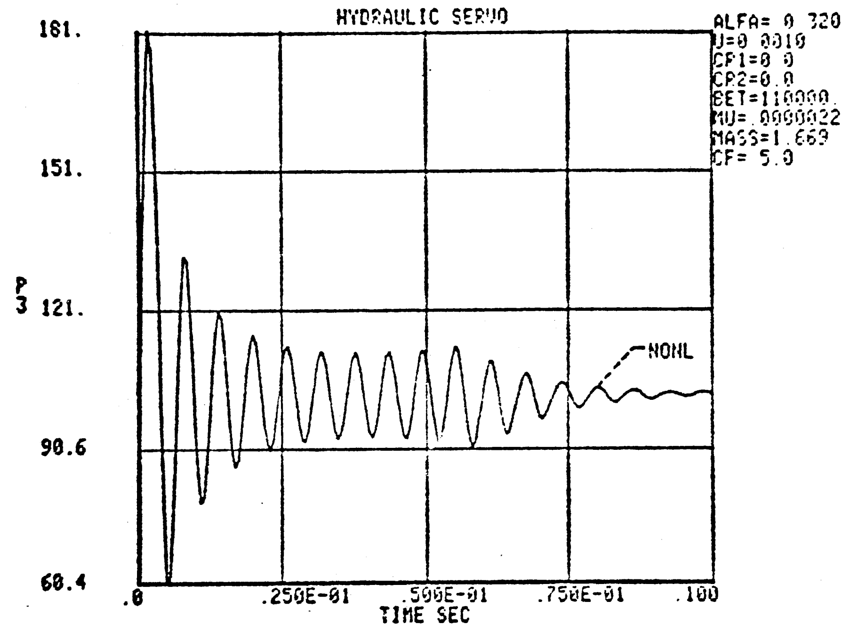


Figure 28. Nonlinear Study of Uncompensated System. P_3 for U = 0.001 in, β = 110,000 psi, $c_{r1} = c_{r2}$ = 0.0, F_{CF} = 5.0 lb/in^2.

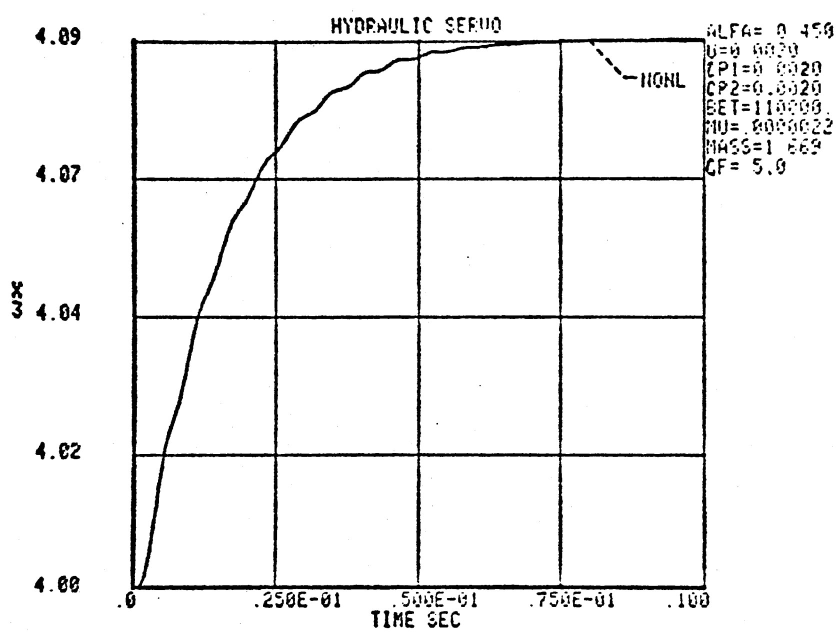


Figure 29. Nonlinear Study of Uncompensated System. x_3 for U = $c_{r1} = c_{r2}$ = 0.002 in, α = 0.45, β = 110,000 psi, F_{CF} = 5.0 lb/in^2.

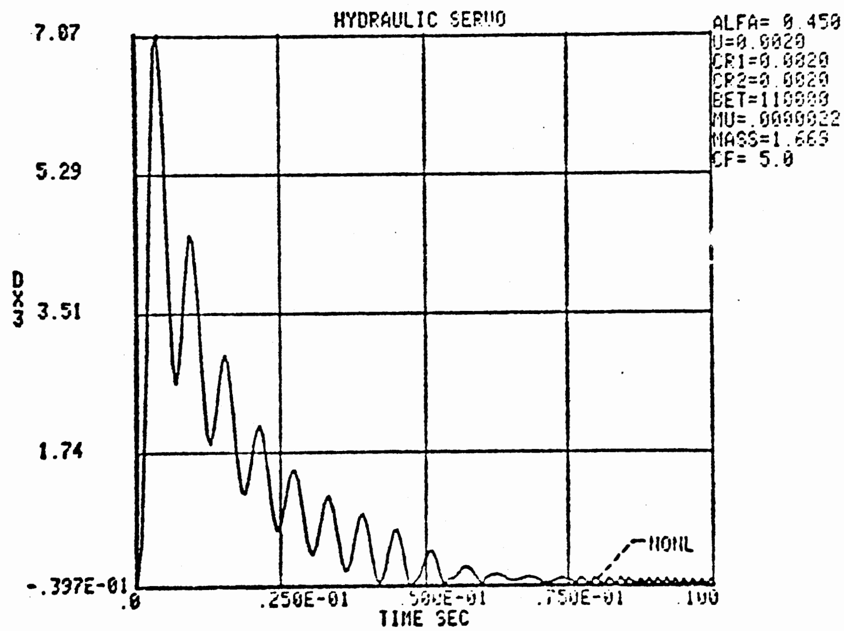


Figure 30. Nonlinear Study of Uncompensated System. $\dot{x}_3$ for $U = c_{r1} = c_{r2} = 0.002$ in, $\alpha = 0.45$, $\beta = 110{,}000$ psi, $F_{CF} = 5.0$ lb/in^2.

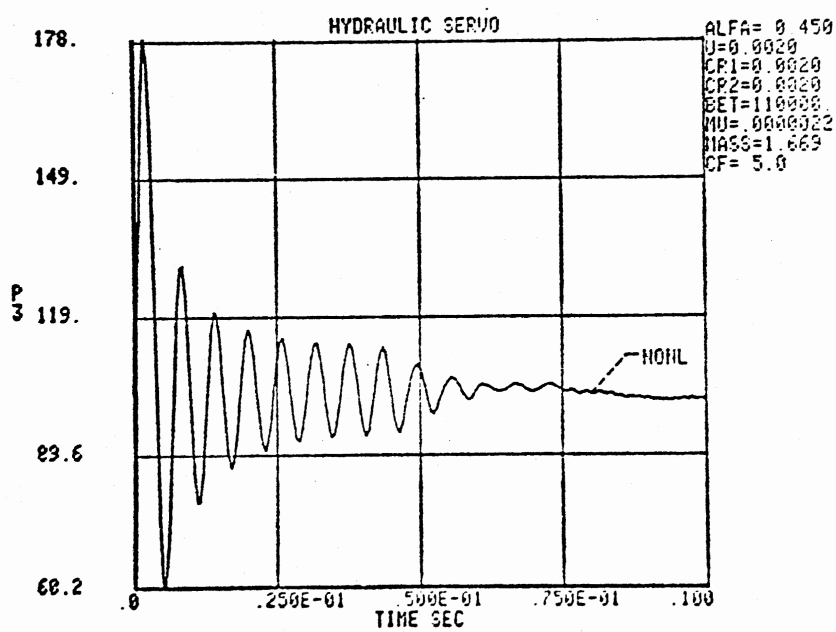


Figure 31. Nonlinear Study of Uncompensated System. P_3 for $U = c_{r1} = c_{r2} = 0.002$ in, $\alpha = 0.45$, $\beta = 110{,}000$ psi, $F_{CF} = 5.0$ lb/in^2.

friction. It is supposed that the erratic nature of the Coulomb friction load could have caused the original system (before overhaul) to exhibit instability or low degree of stability. Momentary entrainment of air into the original system (low β) also may have resulted in a lower than usual degree of stability.

All of the results presented in this section were obtained with the same initial steady-state operating point (x_3 = 4.0 in.) and step size (P_1 = 0.27 psi.) used for the linear analysis in Chapter 3. This study was not exhaustive; the effects of input step size and the location of the initial operating point were not studied. Sufficient results were obtained, however, to illustrate that the original system (before overhaul) was designed to have too low a degree of stability in view of the significant Coulomb friction load. Moreover, these results suggest the need for damping enhancement to be implemented in the existing field installations.

It is the opinion of the writer that the servomotor should be designed to have an adequate degree of stability without depending on stabilization via Coulomb friction. A simple transient flow stabilization teqhnique is introduced in Chapter V, which could be implemented very easily in field installations.

CHAPTER V

TRANSIENT FLOW STABILIZATION

5.1 Introduction

The hydraulic servomotor considered in this thesis is in use in a large number of power plant installations. Equipment changes to effect performance improvement in such an existing system are limited by cost and ease of installation considerations. Major changes requiring complete disassembly and rework normally are considered only as a last resort. An objective of this study was to design a simple compensation means that could be installed easily in the existing field locations at relatively low cost.

Damping enhancement can be achieved in several ways. The use of leakage in the valve or actuator can be effective, but only at the expense of system stiffness and power efficiency. A preferred approach which is widely recognized is to use acceleration feedback. Still another technique is to use pressure or dynamic pressure feedback. The feedback may be introduced directly into the valve design, or into the system via measurement and comparison. Since the servomotor in this study is hydromechanical, conventional acceleration or pressure feedback techniques would be complex to implement.

A technique known as transient flow stabilization was first introduced by Shearer in the book Fluid Power Control (1), for compensation in a pneumomechanical servomotor. This simple technique accomplishes a similar effect as dynamic pressure feedback, but without measurement devices and the corresponding complexity. An advantage of this technique is that increased damping is achieved during transient periods when it is needed and system operation is unaffected during steady-state periods.

Transient flow stabilization or compensation can be implemented in the hydromechanical servomotor of this study by means of a simple resistance-capacitance circuit hooked to the main piston chamber. This approach is especially attractive because the main piston chamber in the actual system already has a hole drilled and tapped for drainage of oil.

Figure 32 shows schematically how the compensation might be achieved. The 'capacitance element' is divided in two parts by a diaphragm. On the side of the diaphragm connected to the main oil chamber, the pressure is P_8. In the steady state, $P_8 = P_3$, since there is no net flow through the resistance and out of the chamber υ_8. Therefore, under steady-state conditions, the capacitance would not affect the system behaviour. On the other side of the diaphragm there is air under compression with the pressure P_a. The compressibility of the air would have a spring-like effect on the diaphragm when there are changes in the pressure P_8.

5.2 Modelling the Compensation Device

The servomotor with the added compensation can be modelled with the same set of equations as in section 2.2, except that the continuity

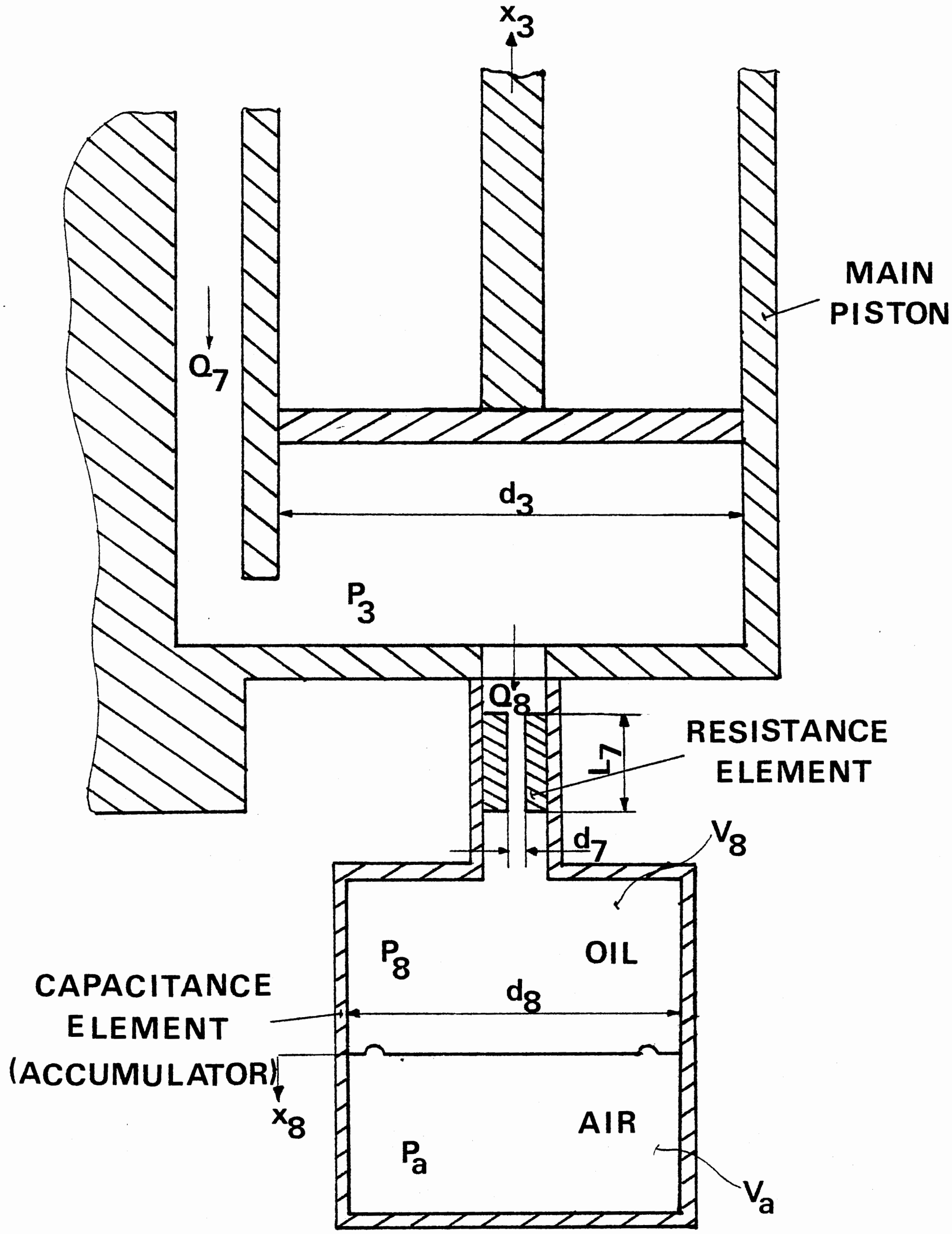


Figure 32. Schematic of the Transient Flow Stabilizer.

equation for the main piston chamber must be altered. That is, for this compensated system, Equation (2.8) is

$$\rho_7 Q_7 = \rho_8 Q_8 + \frac{d}{dt} (\rho_3 \upsilon_3) + \rho_3 c_3 P_3 \tag{5.1}$$

where Q_8 represents the flow through the resistance element into the accumulator.

In addition, it is neccessary to write an equation relating the flow through the resistance to the pressure drop across the resistance. The resistance element is assumed to be a capillary type element. In such a case, the element can be modelled by assuming fully-developed laminar flow. That is,

$$Q_8 = \frac{\pi d_7^4}{128 \mu L_7} (P_3 - P_8). \tag{5.2}$$

It is assumed that the resistance is designed such that the Reynolds number for the flow is small (say 1000). For low Reynolds numbers, an acceptable rule of thumb for choosing the diameter and length of the capillary is $L/d \geqslant 50$.

The continuity equation for the oil chamber is

$$\rho_8 Q_8 = \frac{d}{dt} (\rho_8 \upsilon_8) = \upsilon_8 \frac{d}{dt} (\rho_8) + \rho_8 \frac{d}{dt} (\upsilon_8) .$$

Or if ρ_8 is assumed to be relatively constant,

$$Q_8 = \frac{\upsilon_8}{\beta} \frac{d}{dt} (P_8) + \frac{\pi d_8^2}{4} \dot{x}_8. \tag{5.3}$$

For the air chamber it is assumed that the compression or expansion takes place adiabatically according to the relation

$$P_a = \rho_a^k \cdot \text{constant}.$$

The air chamber continuity equation is

$$0 = \frac{d}{dt}(\rho_a \upsilon_a) = \rho_a \frac{d}{dt}(\upsilon_a) - \upsilon_a \frac{d}{dt}(\rho_a). \tag{5.4}$$

From the process equation it is seen that

$$\frac{1}{P_a}\frac{d}{dt}(P_a) = \frac{k}{\rho_a}\frac{d}{dt}(\rho_a). \tag{5.5}$$

Combining Equations (5.4) and (5.5) yields

$$0 = \frac{\pi d_8^2}{4}(\dot{x}_8) - \frac{\upsilon_a}{k\,P_a}\frac{d}{dt}(P_a). \tag{5.6}$$

If the inertia of the diaphragm is neglected, a force balance requires that

$$P_a = P_8 \, .$$

Thus, Equations (5.6) and (5.3) can be combined to give

$$Q_8 = \frac{\upsilon_8}{\beta}\frac{d}{dt}(P_8) + \frac{\upsilon_a}{k\,P_8}\frac{d}{dt}(P_8)$$

or

$$Q_8 = \{\frac{\upsilon_8}{\beta} + \frac{\upsilon_a}{k\,P_8}\}\frac{d}{dt}(P_8). \tag{5.7}$$

Equations (5.1), (5.2) and (5.7) describe the static and dynamic behaviour of the transient flow stabilization circuit.

Equation (5.7) shows that the process equation for the air chamber includes the nonlinear term $\{\upsilon_a/(k\cdot P_8)\}\,\dot{P}_8$. This term will not cause any difficulty in setting up the compensated system model using DYSIMP, but if a new transfer function is desired, Equation (5.7) will need to be linearized.

It is apparent that the transient flow stabilization circuit adds a first order dynamic element into the system model; the final model is fifth order.

5.3 The Dynamics of the Compensator

Equation (5.7) can be linearized and combined with Equation (5.2) to derive a transfer function relating the pressure under the piston P_3 to the pressure in the tank, P_8. The linearized form of Equation (5.7) is

$$Q_8 = \left[\frac{\upsilon_{8i}}{\beta} + \frac{\upsilon_{ai}}{k\ P_{8i}} \right] \frac{d}{dt} (\ P_8)$$

where the subscript i represents the initial steady-state operating point. Combining the above equation with Equation (5.2) yields

$$\frac{\pi d_7^4}{128 \mu L_7} (\Delta P_3 - \Delta P_8) = \{ \frac{\upsilon_{8i}}{\beta} + \frac{\upsilon_{ai}}{k\ P_{8i}} \} \frac{d}{dt} (\Delta P_8)$$

or, in Laplace form

$$\frac{\pi d_7^4}{128 \mu L_7} (\Delta P_3) = \left[\{ \frac{\upsilon_{8i}}{\beta} + \frac{\upsilon_{ai}}{k\ P_{8i}} \} s + \frac{\pi d_7^4}{128 \mu L_7} \right] (\Delta P_8)$$

which results in the transfer function

$$\frac{\Delta P_8}{\Delta P_3} = \frac{1}{\tau_p\ s + 1} \qquad (5.8)$$

where

$$\tau_p = \frac{128 \mu L_7}{\pi d_7^4} \{ \frac{\upsilon_{8i}}{\beta} + \frac{\upsilon_{ai}}{k\ P_{8i}} \} .$$

The effect of the compensator is then to add a first order element into the system, with the time constant τ_p and a steady-state gain of 1. Therefore, it is active during transient periods, but has no effect in the steady-state periods.

The compensator can be compared to a simple electric R_eC_e circuit, where the electrical resistance (R_e) corresponds to the capillary characteristics

$$R = 128\mu L_7/(\pi d_7^4)$$

and the electrical capacitance (C_e) corresponds to the accumulator capacitance

$$C = \left[\frac{\upsilon_{8i}}{\beta} + \frac{\upsilon_{ai}}{k\, P_{8i}}\right].$$

The time constant, τ_p, is a key factor in the effect of the compensation. The characteristic time or period of the overall system (i.e. the rise time following a step input), is of the order of 0.005 sec. (see Chapters III and IV). The design problem is to select the desired value of τ_p (relative to the system characteristic time) and to size the elements.

5.4 Compensated System Transfer Function and Parameter Study

Appendix G shows the derivation of a linearized transfer function for the fifth order system with the compensation. The final form of the transfer function is as follows

$$\frac{\Delta x_3}{\Delta P_3} = \frac{K\ (\tau_p + 1\ )}{a_o\ s^5 + a_1\ s^4 + a_2\ s^3 + a_3\ s^2 + a_4\ s + a_5} \quad . \qquad (5.9)$$

The effect of the compensation is to increase the denominator by one order and add a first order component to the numerator.

The Routh criterion can be used with Equation (5.9) as was done in section 3.5, to determine the bounds on the parameter values required for system stability.

The compensator parameters can be chosen in order to arrive at different values of the time constant τ_p. The capacitance factor is

$$C = \left[\frac{\upsilon_{8i}}{\beta} + \frac{\upsilon_{ai}}{k\ P_{8i}}\right] .$$

If the same steady-state operating point is chosen as before (x_3 = 4.0 in.), then $P_{8i} = P_{3i}$ = 103.0 psi. from the steady-state condition. The constant k for the adiabatic compression of air is 1.4. The value of fluid bulk modulus is assumed β = 110,000 psi. Assuming that the air volume is approximately equal to the oil volume in the accumulator, then the capacitance becomes

$$C = \ \upsilon \left\{\frac{1}{110{,}000} + \frac{1}{1.4 \cdot 103}\right\} = 0.0069 \cdot \upsilon \ \ in^5/lb_f .$$

It is obvious that the compressibility of the hydraulic fluid is negligible compared to that of the air. Therefore, it can be assessed that the capacitance is approximately

$$C = \ \upsilon_a / \ (k\ P_{8i}) .$$

The expression for the time constant τ_p can be written

$$\tau_p = \frac{128 \cdot 2.2 \cdot 10^{-6}}{\pi} \frac{L_7}{d_7^4} \frac{\upsilon_a}{1.4 \cdot 103}$$

or $$\tau_p = 622 \cdot 10^{-9} (L_7/d_7)\upsilon_a. \qquad (5.10)$$

For the capillary flow equation to be valid, the ratio of length to diameter for the capillary should be larger than 50, or

$$L_7 / d_7 \geqslant 50. \qquad (5.11)$$

Table V shows three different sets of parameters for the compensator, which represent a range of τ_p of approximately equal to the overall system time constant to two orders of magnitude larger.

TABLE V

COMPENSATION PARAMETERS

d_7 (in)	L_7 (in)	υ_{ai} (in^3)	R ($lb \cdot sec/in^5$)	C (in^5/lb)	τ_p (sec)
0.005	2.5	20.0	35.85	0.139	4.98
0.150	7.5	98.0	1.33	0.680	0.90
0.300	15.0	100.0	0.166	0.693	0.05

These parameter values can now be used in the transfer function and a Routh test performed to see which one, if any, will render the system stable. Therefore, the set of parameters from an unstable case from section 3.5 should be used in conjunction with the added parameters of the compensation to validate the compensator design. That is, a value of bulk modulus β = 110,000 psi., no radial clearances and a valve underlap of 0.002 in. For the same operation point x_3 = 4.0 in. and a compensator time constant of τ_p = 4.98 from Table V the transfer function is

$$\frac{\Delta x_3}{\Delta P_1} = \frac{4.662 \cdot 10^6 \, (4.98\, s + 1)}{772 \cdot 10^{-6} s^5 + 714 \cdot 10^{-3} s^4 + 954\, s^3 + 872 \cdot 10^3 s^2 + 54 \cdot 10^6 s + 10.9 \cdot 10^6} .$$

The Routh array is as follows:

s^5:	$772 \cdot 10^{-6}$	954	$54 \cdot 10^6$
s^4:	$714 \cdot 10^{-3}$	$872 \cdot 10^3$	$10.9 \cdot 10^6$
s^3:	11.2	$54.0 \cdot 10^6$	0
s^2:	$-2.57 \cdot 10^6$	$10.9 \cdot 10^6$	0
s^1:	$54 \cdot 10^6$	0	
s^0:	$10.9 \cdot 10^6$		

There are two changes of sign, hence two roots of the characteristic equation have positive real parts and the resulting system is unstable. It is therefore apparent, that this would not be an adequate set of parameter values to make the system stable.

For the second case of $d_7 = 0.15$ in. and $\tau_p = 0.90$ sec. the transfer function is

$$\frac{\Delta x_3}{\Delta P_1} =$$

$$\frac{4.662 \cdot 10^6 (\, 0.90\ s\ +\ 1\,)}{28 \cdot 10^{-6} s^5 + 33 \cdot 10^{-3} s^4 + 40.6\ s^3 + 32 \cdot 10^3 s^2 + 2.1 \cdot 10^6 s + 10.9\ 10^6} .$$

The Routh array is as follows:

s^5:	$28 \cdot 10^{-6}$	40.6	$2.1 \cdot 10^6$
s^4:	$33 \cdot 10^{-3}$	$32 \cdot 10^3$	$10.9 \cdot 10^6$
s^3:	13.5	$2.1 \cdot 10^6$	0
s^2:	$27 \cdot 10^3$	$10.9 \cdot 10^6$	0
s^1:	$2.1 \cdot 10^6$	0	
s^0:	$10.9 \cdot 10^6$		

For this set of parameters, there is no change of sign in the first column, so this system is stable.

For the third case with a time constant similar to that of the system itself, $\tau_p = 0.05$ sec., the transfer function is

$$\frac{\Delta x_3}{\Delta P_1} =$$

$$\frac{4.662 \cdot 10^6 (\, 0.005\ s\ +\ 1\,)}{8.14 \cdot 10^{-6} s^5 + 23.3 \cdot 10^{-3} s^4 + 23.5\ s^3 + 9.35 \cdot 10^3 s^2 + 716 \cdot 10^3 s + 10.9 \cdot 10^6} .$$

The Routh array is as follows:

s^5:	$8.14 \cdot 10^{-6}$	23.5	$716 \cdot 10^{3}$
s^4:	$23.3 \cdot 10^{-3}$	$9.35 \cdot 10^{3}$	$10.9 \cdot 10^{6}$
s^3:	20.2	$712 \cdot 10^{3}$	0
s^2:	$8.53 \cdot 10^{3}$	$10.9 \cdot 10^{6}$	0
s^1:	$686 \cdot 10^{3}$	0	
s^0:	$10.9 \cdot 10^{6}$		

Again, there is no change of sign in the first column, so this system configuration is also stable. The results of this Routh study can then be summarized as shown in Table VI.

TABLE VI

RESULTS OF A ROUTH STUDY
OF COMPENSATED SYSTEM

R ($lb \cdot sec/in^5$)	C (in^5/lb)	τ_p (sec)	Stability Result
35.85	0.139	4.98	Unstable
1.33	0.680	0.90	Stable
0.166	0.693	0.05	Stable

Therefore, it appears that a time constant of the order of 0.90 sec. would make the system stable, but if it were any larger, stability might not be achieved. In this Routh study, the effect of the parameters R and C has not been studied individually, only the effect of their product, τ_p. In the compensated system transfer function they appear both as this product and also separately, so it could be deceiving to study only the effect of varying τ_p.

The above analysis has established bounds on the compensator parameter τ_p. Further validation of the effect of the parameters R and C can be achieved through computer simulation. The simulation tool can be used to set more accurate limits on the value of the compensator parameters needed for the desired system performance.

5.5 Nonlinear Simulation of the Compensated System

The objective of this computer simulation study of the compensated system is to validate the concept of transient flow stabilization for this hydromechanical servosystem and to establish bounds on the design parameters, R, C and τ_p. The actual design of the transient flow stabilizer will not be carried through in this thesis.

It is a relatively simple procedure to extend the model already developed for DYSIMP to include the compensator. One additional equation is required to describe the dynamic changes in the pressure P_8 and only the equation for the time derivative in P_3 has to be changed.

From Equation (5.1) the time derivative of P_3 is as follows:

$$\dot{P}_3 = \left[Q_3 - Q_6 - \frac{\pi d_7^4}{128\mu L_7} (P_3 - P_8) - A_3\dot{x}_3 - c_3P_3 \right] \frac{\beta}{\upsilon_{3i} + A_3x_3} \tag{5.12}$$

From Equations (5.2) and (5.7) the time derivative of P_8 is

$$\dot{P}_8 = \frac{\pi d_7^4}{128 \mu L_7} (P_3 - P_8) \frac{1}{\left[\frac{\upsilon_{8i}}{\beta} + \frac{\upsilon_{ai}}{k P_{8i}} \right]} . \qquad (5.13)$$

Appendix H describes the changes made in the earlier developed DYSIMP model to accomodate the compensation device, and lists the program used. This program was used to obtain simulation results for the compensator parameters as listed in Table VII. The system parameters were the same as presented in Figures 22 and 23, which was an unstable configuration. No Coulomb friction was included, in order to validate the effect of the transient flow stabilization on the system. The premise is that the system dynamic performance should be acceptable with and without a Coulomb friction load.

TABLE VII

RESULTS OF A COMPUTER SIMULATION STUDY OF COMPENSATED SYSTEM

d_7 (in)	L_7 (in)	υ_{ai} (in^3)	R ($lb \cdot sec/in^5$)	C (in^5/lb)	τ_p (sec)	Stability Result
0.10	8.2	98	7.350	0.680	5.0	Unstable
0.15	7.5	98	1.330	0.680	0.90	Stable
0.14	12.0	20	2.800	0.139	0.39	Stable, lightly damped
0.30	15.0	100	0.166	0.693	0.05	Stable
0.30	15.0	20	0.166	0.139	0.023	Stable

The results from this computer simulation study are presented in the same order as in Table VII in Figures 33 through 42. Figures 33 and 34 show an unstable response. Comparing this result with Figures 39 and 40, for approximately the same C but the latter case a much smaller R, it can be concluded that a large value of R will provide less damping. This conclusion is further supported by the results shown in Figures 37 and 38, as compared with Figures 41 and 42. Therefore, the most important design parameter from the standpoint of system stability is seen to be the resistance of the capillary type orifice.

Another consideration in the design of a compensation is the speed of response of the resulting system. It is apparent in the case of this compensation, that the speed of response of the stabilized system would literally be unaffected, given the results for the sets of parameters presented here. A comparison of Figures 37 and 41 (R is varied, C is constant) reveals that a larger R will give a slightly slower response. The difference is minimal, however. It is therefore the conclusion, that system stability is the criterion that should be used when the compensator parameters are chosen.

To summarize, the largest value of R tested, which gave a completely stable response, was R = 1.33 $lb\cdot sec/in^5$. The value of the capacitance is not so important, and could be anywhere between 0.139 and 0.693 in^5/lb. Therefore, the following set of parameters is suggested for the transient flow stabilizer:

$d_7 = 0.30$ in. $\qquad$ $L_7 = 15.0$ in. $\qquad$ $\upsilon_a = 100.0$ $in.^3$

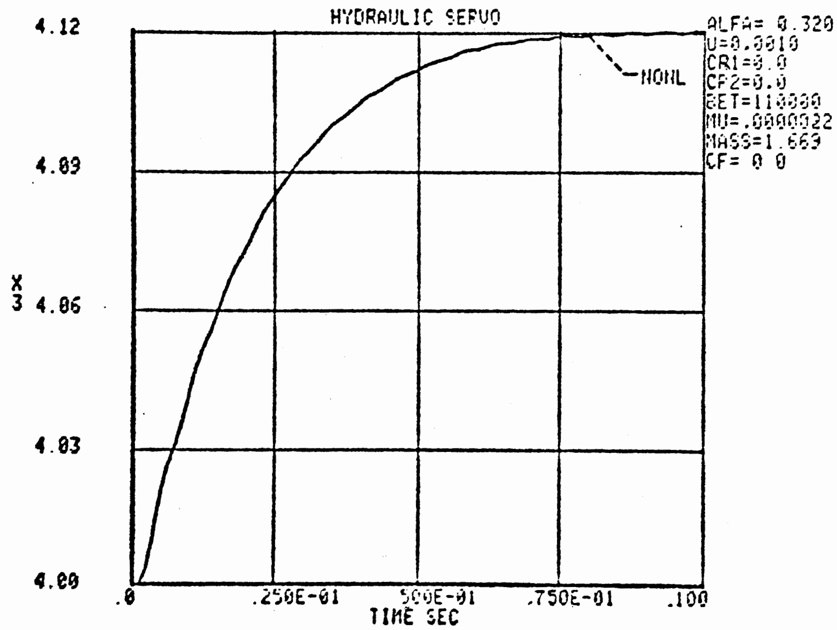


Figure 33. Nonlinear Study of Compensated System. x_3 for τ_p = 5.0 sec.

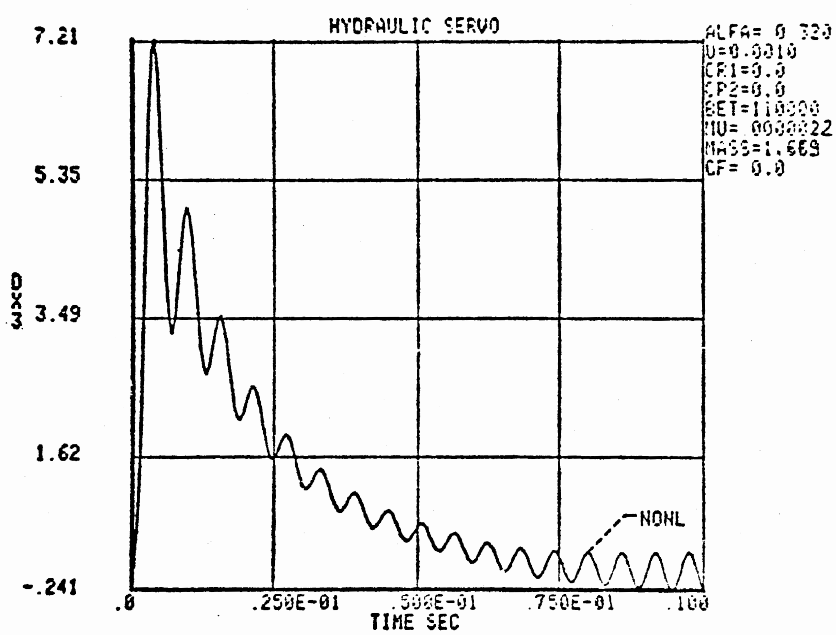


Figure 34. Nonlinear Study of Compensated System. $\dot{x}_3$ for τ_p = 5.0 sec.

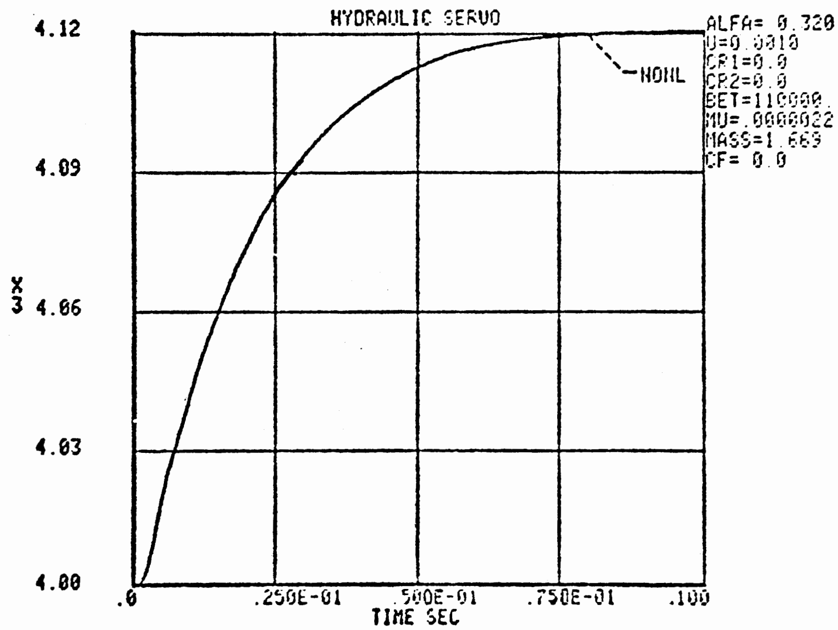


Figure 35. Nonlinear Study of Compensated System.
x_3 for τ_p = 0.90 sec.

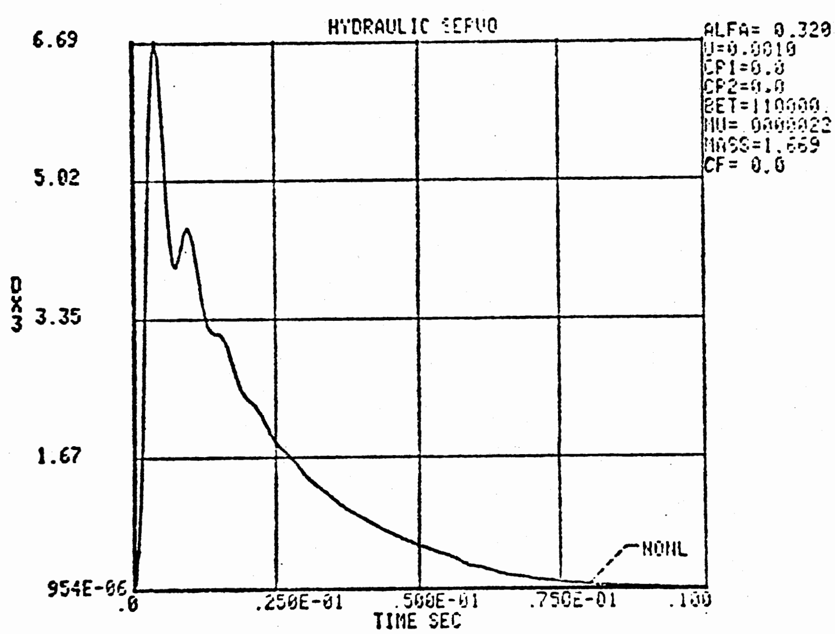


Figure 36. Nonlinear Study of Compensated System.
$\dot{x}_3$ for τ_p = 0.90 sec.

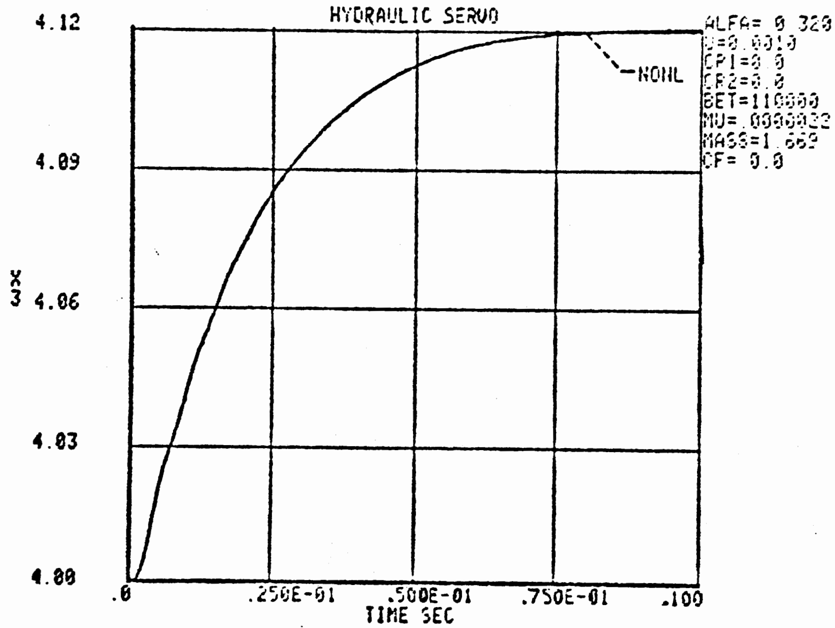


Figure 37. Nonlinear Study of Compensated System. x_3 for τ_p = 0.39 sec.

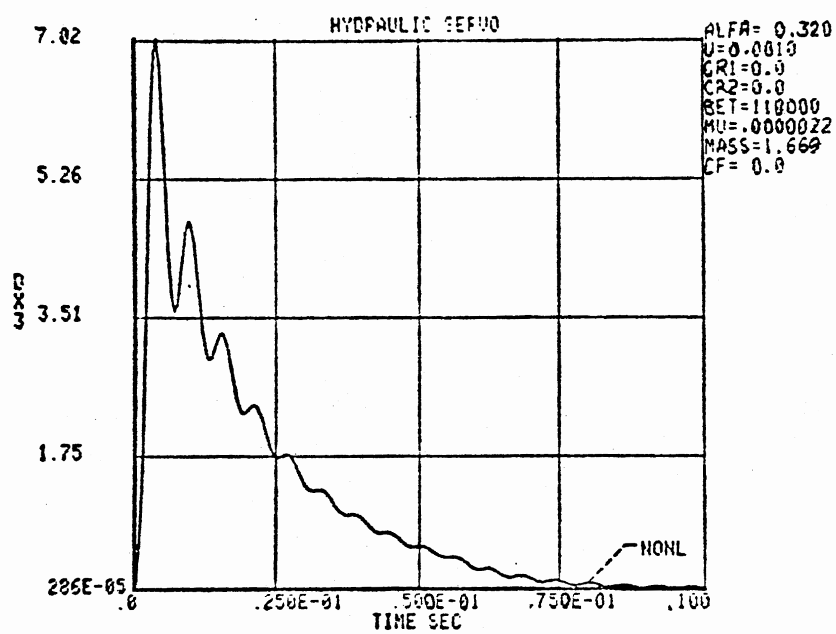


Figure 38. Nonlinear Study of Compensated System. $\dot{x}_3$ for τ_p = 0.39 sec.

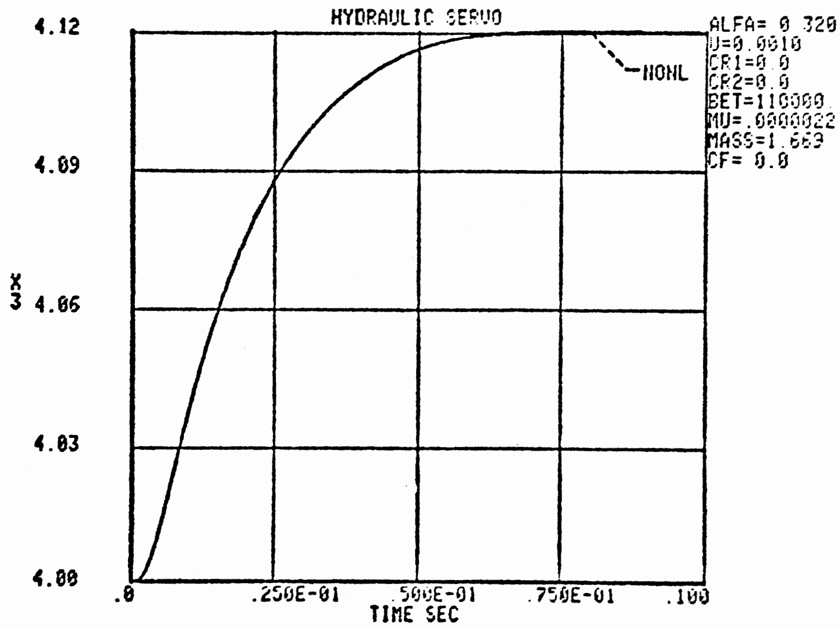


Figure 39. Nonlinear Study of Compensated System. x_3 for τ_p = 0.05 sec.

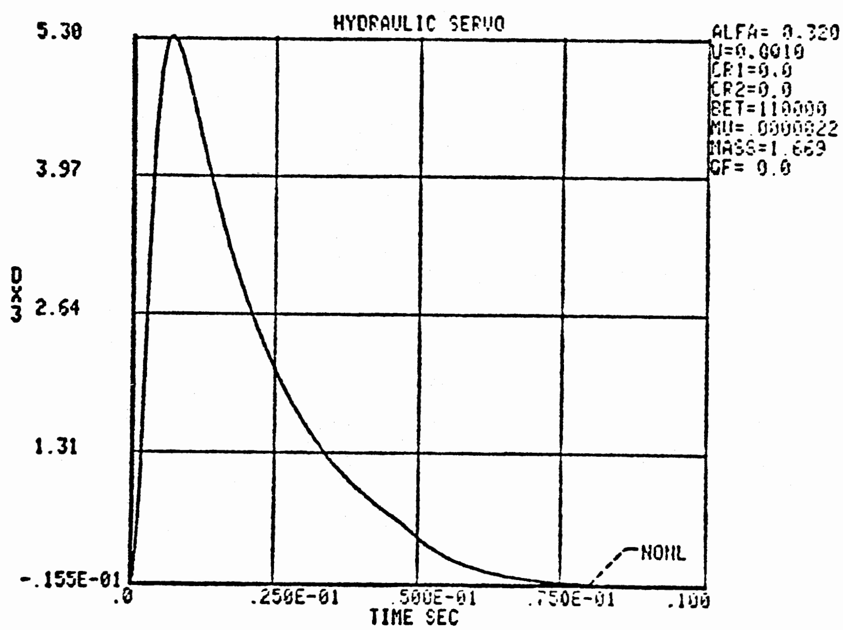


Figure 40. Nonlinear Study of Compensated System. $\dot{x}_3$ for τ_p = 0.05 sec.

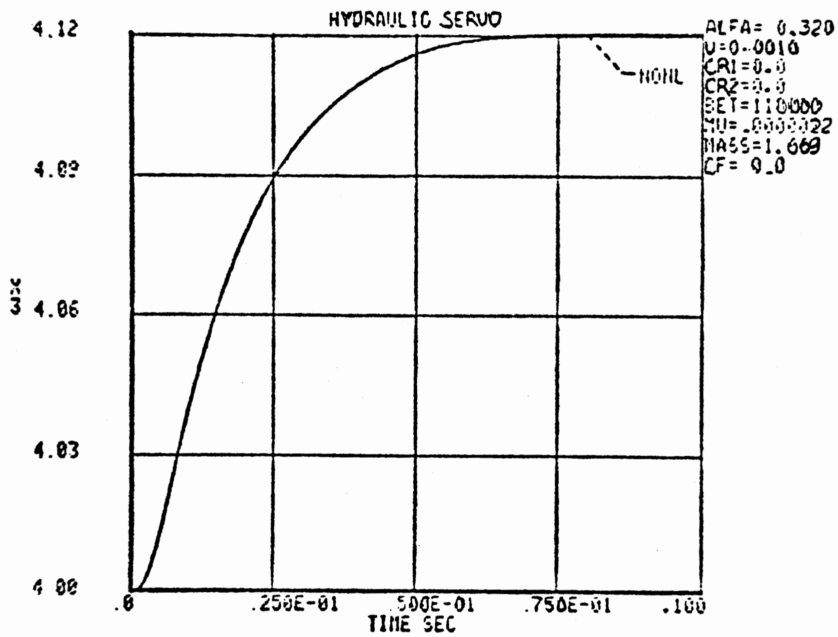


Figure 41. Nonlinear Study of Compensated System.
x_3 for τ_p = 0.023 sec.

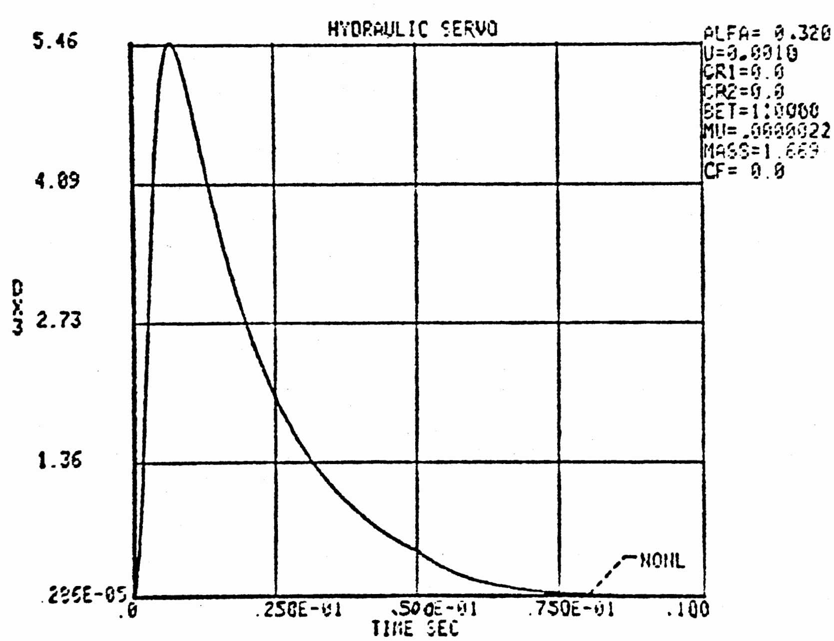


Figure 42. Nonlinear Study of Compensated System.
$\dot{x}_3$ for τ_p = 0.023 sec.

which results in

$$R = 0.166 \text{ lb}\cdot\text{sec/in}^5 \qquad C = 0.693 \text{ in}^5\text{/lb} \qquad \tau_p = 0.05 \text{ sec.}$$

This set of parameters should be within the range of practically realizable design and give a perfectly stable system even in the absence of other stabilizing effects like Coulomb friction or a high bulk modulus of the operating fluid.

CHAPTER VI

SUMMARY AND RECOMMENDATIONS

6.1 Summary

The two objectives of the preceding study as introduced in section 1.2 have been reached.

First, it has been shown that the hydromechanical servomotor, as it has been installed without damping enhancement, relies on Coulomb friction in the load or a high value of operating fluid bulk modulus for its stability. This is considered to be an undependable means of assuring stability of a system; therefore, damping enhancement is needed.

Second, it has been verified using computer simulation, that the system damping can be enhanced using a form of transient flow stabilization. A design configuration has been suggested and a parametric study through computer simulation has been conducted to establish bounds on the design parameters. It has been verified that transient flow stabilization can enhance the damping of the system under study and do so without appreciable effect on the overall system speed of response, or the input-output relationship in the steady state. It has also been reasoned that the suggested design configuration would be relatively easy to install in the field and would result in the cheapest way to enhance the system damping.

6.2 Recommendations

The computer simulation study conducted in this thesis is the first step in verifying the effectiveness of transient flow stabilization on this hydromechanical system. It still remains to choose the final design parameters and actually construct the compensator. It would be advisable to build a prototype and to install it in an existing field system. Thorough performance tests should be conducted, so that it is clear that the desired added stability and performance has been achieved. It is the opinion of the author that by using the suggested design parameters in section 5.5, the system would exhibit acceptable performance in all aspects.

SELECTED BIBLIOGRAPHY

1. Blackburn, J. F., Reethof, G. and Shearer, J. L., Fluid Power Control, M. I. T. Press, Cambridge, Mass., 1960.

2. Reid, K. N., Classnotes for MAE5453, Fluid Power Control I, Fall 1980, Unpublished, Stillwater, Oklahoma; Oklahoma State University, 1980.

3. Ogata, K., Modern Control Engineering, Prentice Hall, Englewood Cliffs, N. J., 1970.

4. Meyfarth, P. F., "Dynamic Response Plots and Design Charts for Third Order Linear Systems", Research Memorandum No. R. M. 7401-3, Massachusetts Institute of Technology, Cambridge, Mass., Sept. 1958.

5. Richardson, H. H., "The Analytical Design of Valve-Controlled Hydraulic Power and Control Systems. A Case Study.", Unpublished Paper, Massachusetts Institute of Technology, Cambridge, Mass., 1962

6. Ebbesen, L. R., Documentation on DYSIMP-2.0, Oklahoma State University, Stillwater, Oklahoma.

7. Carnahan, B., Luther, H. A. and Wilkes, J. O., Applied Numerical Methods, John Wiley and Sons, Inc., New York, 1969.

8. Conte, S. D. and deBoor, C., Elementary Numerical Analysis - An Algorithm Approach, Second Edition, McGraw-Hill, Kogakusha, 1972.

9. Merritt, H. D., Hydraulic Control Systems, John Wiley, New York, 1967.

10. Graduate College, Thesis Writing Manual, Stillwater, Oklahoma; Oklahoma State University, 1972.

APPENDIX A

DETERMINATION OF NUMERICAL VALUES FOR PARAMETERS

Available for this study were most of the detail drawings of the servomotor parts and certain performance information in the form of charts or graphs provided by the manufacturer. Reference to drawing numbers is given where appropriate. Other information is identified according to source.

<u>Diameter of relay piston rod</u>:

Drawing # 650D479 $d_1 = 0.996$ in.

<u>Diameter of relay piston</u>:

Drawing # 650D479 $d_2 = 2.494$ in.

<u>Diameter of main piston</u>:

Drawing # 650D677 $d_3 = 9.99$ in.

<u>Diameter of capillary orifice</u>:

Drawing # 651D422 $d_4 = 3/16$ in.

<u>Diameter of relay plunger</u>:

Drawing # 651D422 $d_6 = 3.495$ in.

<u>Orifice width of relay piston</u>:

Drawing # 651D422 $w_1 = 1.750$ in.

<u>Relay piston top area</u>:

Drawing # 651D422 $A_2 = 2.341$ in.^{2}

<u>Clearance between relay plunger and bushing</u>:

Drawing # 664J541 c_2 = 0.003 in. - 0.006 in.

If 0.004 in. is taken as a nominal value, the orifice radial clearance is $c_2/2 = c_{r2}$ = 0.0020 in.

<u>Clearance between relay piston and relay plunger</u>:

Drawing # 664J541 c_1 = 0.004 in.- 0.0055 in.

If 0.004 in. is taken as a nominal value, the orifice radial clearance is $c_1/2 = c_{r1}$ = 0.0020 in.

<u>Clearance between the main piston and its housing</u>:

Drawing # 664J541 c_{r3} = 0.0020 in.

<u>Capillary orifice length</u>:

Drawing # 651D422 L_4 = 4.0 in.

<u>Length of leakage path past main piston</u>:

Drawing 271A927 L_3 = 0.8 in.

This parameter is hard to estimate, and is taken to be approximately equal to the sum of the effective compressed seal lengths on the piston.

<u>Pilot relay spring constant (11)</u>:

Supplied by servomotor manufacturer k_1 = 30.0 lb/in.

<u>Main relay spring constant (10)</u>:

Supplied by servomotor manufacturer k_2 = 212.0 lb/in.

<u>Main relay spring preload</u>:

Supplied by servomotor manufacturer F_s = 13.5 lb.

<u>Volume of fluid above relay plunger</u>:

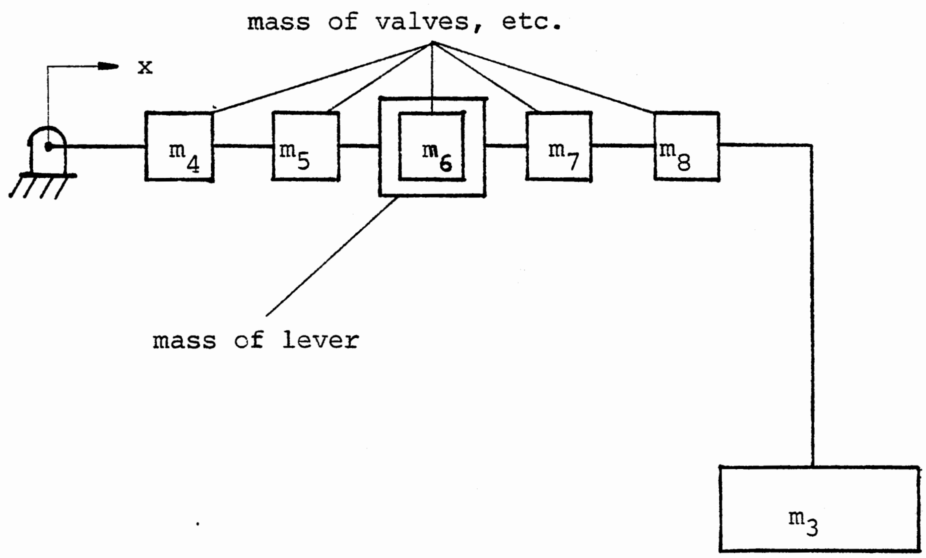
Supplied by servomotor manufacturer υ_4 = 11.6 in^3.

Volume of fluid under main piston in zero position:

Supplied by servomotor manufacturer $\upsilon_3 = 30.0\ in.^3$

Equivalent mass of steam chest lever and valves:

Drawings # 499C416 and # 664J573



$$m_e = m_4(x_4/L)^2 + m_5(x_5/L)^2 + m_6(x_6/L)^2 + m_7(x_7/L)^2 + m_8$$

$$m_4 = m_5 = m_7 = m_8 = 0.487\ lb\cdot sec^2/in.$$

$$m_6 = 0.487 + 2.487 = 2.974\ lb\cdot sec^2/in. \qquad m_e = 1.3656\ lb\cdot sec^2/in.$$

Orifice discharge coefficient:

A common estimate for this variable, which depends on orifice type, Reynolds number etc. is to set it at

$$c_d = 0.625$$

Fluid viscosity:

For MIL5605 A at the expected operating temperature

$$\mu = 2.2\cdot 10^{-6}\ lb\cdot sec/in.$$

Fluid density:

For MIL5606 A $\rho = 8 \cdot 10^{-5}$ lb·sec^2/in^4.

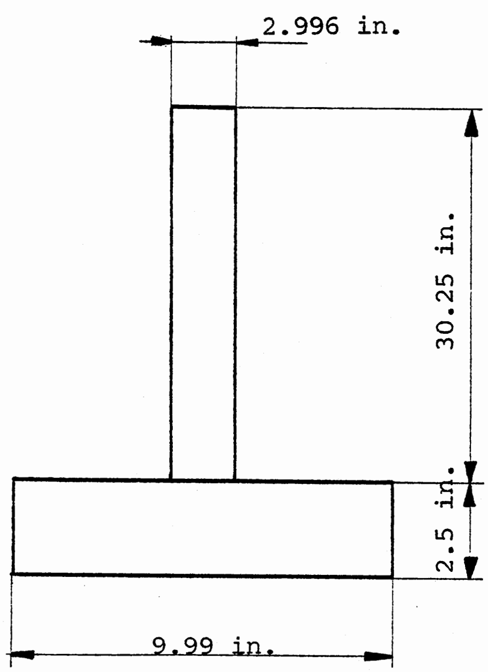
Fluid bulk modulus of elasticity:

For MIL5606 A a theoretical value of β = 230,000 psi. is often used, but experience has shown this value to be too high. This parameter depends highly on the amount of air entrained in the oil. A good value based on previous experience is

$$\beta = 110{,}000 \text{ psi.}$$

Mass of main piston and rod:

Drawing # 650D677

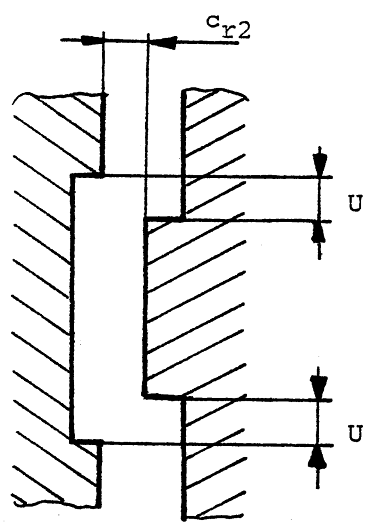


$$m_3 = \rho \cdot Vol/g = \frac{0.286}{386}\left\{\frac{\pi(9.99)^2}{4} \cdot 2.5 + \frac{\pi(2.996)^2}{4} \cdot 30.25\right\}$$

$$m_3 = 0.303 \text{ lb}\cdot\text{sec}^2/\text{in.}$$

Main valve underlap:

Drawing # 664J541



U = 0.000 - 0.002 in.

The feedback coefficients x_3^* and α:

Figure 4 shows the main piston displacement x_3 as a function of control oil pressure, P_1, for steady-state operation. This graph may be used to determine the feedback parameters, x_3^* and α.

Figure 4 shows two different cases and as discussed in section 2.3 they correspond to data measured in the field by K. N. Reid on the system after overhaul, and information supplied by the manufacturer of the servomotor (two-slope cam). There is uncertainty as to which particular feedback cam was used in the system before overhaul, so a range of feedback parameters should be investigated.

When the system is at rest in a steady state, the equations developed in section 2.2 are a set of algebraic, nonlinear equations. This equation set can be solved successively, given a certain initial steady-state operating point x_{3i}. The steady-state equations are:

$$P_1(\pi/4)\ (d_2^2 - d_1^2) = k_1 x_1 \tag{A.1}$$

$$P_4 A_2 = k_2\ (x_2 - x_4) + F_s \tag{A.2}$$

$$P_2 - P_4 = \frac{128\mu L_4 Q_4}{\pi d_4^4}\ \left(1 + \frac{2.28\rho Q_4}{16\pi\ \mu\ L_4} \right) \tag{A.3}$$

$$Q_5 = c_d \pi w_1 \sqrt{(x_2 - x_1)^2 + c_{r1}^2}\ \sqrt{(2/\rho)\ \ P_4} \tag{A.4}$$

$$Q_3 = c_d \pi d_6 \sqrt{(x_2 - x_4 + U)^2 + c_{r2}^2}\ \sqrt{(2/\rho)\ (P_2 - P_3)} \tag{A.5}$$

$$Q_6 = c_d \pi d_6 \sqrt{(x_4 - x_2 + U)^2 + c_{r2}^2}\ \sqrt{(2/\rho)\ P_3} \tag{A.6}$$

$$Q_4 = Q_5 \tag{A.9}$$

$$F_{ext} = (\pi/4)\ d_3^2\ P_3 \tag{A.10}$$

$$x_4 - x_3^* = \alpha\ x_3 \tag{A.11}$$

It is apparent, that these equations cannot be solved directly, without some iterative calculations, because of the nonlinear orifice equations. It therefore seems feasible to use the computer to obtain the steady-state solution. Successive iterations will allow determination of x_3^* and α.

Appendix C contains a source listing of a Fortran IV program, which was written to solve the steady-state equations. The solution starts with a selected value for x_{3i}. Figure 3 is used in a tabular form to calculate the external load corresponding to each value of x_{3i} and a corresponding value for P_1. The program was written to solve the set of equations for a number of values of x_{3i} and to construct a plot similar to Figure 4. In two cases, the program uses an iterative regula-falsi method to solve for the variable x_2 and also Q_4.

To estimate approximate values for x_3^* and α it can be assumed that at steady state $\Delta x_1 = \Delta x_2 = \Delta x_4$. For a given increase in control pressure, ΔP_1, a corresponding displacement of the main piston, Δx_3, can be found from Figure 4. Also, Δx_1 can be calculated from Equation (A.1) as

$$\Delta x_1 = \Delta P_1 \frac{\pi(d_2^2 - d_1^2)}{4\ k_1} = \Delta x_4.$$

From Equation (A.11)

$$\Delta x_4 = \alpha\ \Delta x_3.$$

From Figure 4, Curve B, an arbritrary change in P_1 and the corresponding change in x_3 are

$\Delta P_1 = 35.0 - 25.7 = 9.3$ psi.

$\Delta x_3 = 6.0 - 2.0 = 4.0$ in.

For this ΔP_1, the following value of Δx_4 is obtained:

$$\Delta x_4 = \frac{9.3\ \pi(2.494^2 - 0.996^2)}{4 \cdot 30.0} = 1.27 \text{ in.}$$

Therefore, $\alpha = \Delta x_4/\Delta x_3 = 1.27/4.0 = 0.32$

The value of the initial displacement, x_3^* can now be determined using Equations (A.1) and (A.11) and P_1 = 35.0 psi., x_3 = 6.0 in. That is,

$$x_1 = \frac{35.0\ (2.494^2 - 0.996^2)}{4 \cdot 30.0} = 4.79 \text{ in.}$$

and therefore

$$x_3^* = 4.79 - 0.32 \cdot 6.0 = 2.87 \text{ in.}$$

This estimate of the feedback coefficients does not take into account the leakage flow past the main piston. The complete set of steady-state equations were solved using the program in Appendix C and the results were very close to the above estimate. From the data measured by K. N. Reid, using the single slope cam, the results are:

$\alpha = 0.32$ and $x_3^* = 2.90$ in.

For the two-slope cam represented also in Figure 4, these pertinent values are

For $P_1 < 34.25$ psi:

$\alpha = 0.90$ and $x_3^* = 2.10$ in.

For $P_1 > 34.25$ psi:

$\alpha = 0.45$ and $x_3^* = 3.38$ in.

APPENDIX B

DERIVATION OF THE LINEAR TRANSFER FUNCTION

In section 3.2, the following set of linear algebraic and differential equations were derived:

$$A_1 \Delta P_1 = k_1 \Delta x_1 \tag{B.1}$$

$$A_2 \Delta P_4 = k_2 (\Delta x_2 - \Delta x_4) \tag{B.2}$$

$$M \Delta \ddot{x}_3 + k_3 \Delta x_3 = A_3 \Delta P_3 \tag{B.3}$$

$$\Delta Q_7 = \Delta Q_3 - \Delta Q_6 \tag{B.4}$$

$$\Delta Q_4 - \Delta Q_5 = A_2 \Delta \dot{x}_2 \tag{B.5}$$

$$\Delta Q_7 = (\upsilon_3/\beta) \Delta \dot{P}_3 + A_3 \Delta \dot{x}_3 + c_3 \Delta P_3 \tag{B.6}$$

$$\Delta Q_3 = a_1 (\Delta x_2 - \Delta x_4) - b_1 \Delta P_3 \tag{B.7}$$

$$\Delta Q_6 = a_2 (\Delta x_4 - \Delta x_2) + b_2 \Delta P_3 \tag{B.8}$$

$$\Delta Q_5 = a_3 (\Delta x_2 - \Delta x_1) + b_3 \Delta P_4 \tag{B.9}$$

$$-\Delta P_4 = a_4 \Delta Q_4 \tag{B.10}$$

$$\Delta x_4 = \alpha \Delta x_3 . \tag{B.11}$$

One additional simplifying assumption was made for this approximate analysis. In Equation (B.5) the compressibility effect of the oil in the chamber above the relay plunger has been deleted. This simplification makes the derivation somewhat less involved. Since the volume is relatively small, the accuracy of the solution should not be affected appreciably.

In equation (B.1), $A_1 = (\pi/4)(d_2^2 - d_1^2)$. And in Equations (B.3) and (B.6), $A_3 = (\pi/4)\ d_3^2$, which serves to make the notation simpler.

The set of 11 equations in 12 unknowns can be combined to yield the desired closed-loop transfer function. Equations (B.10), (B.9) and (B.5) can be combined to yield

$$-(\Delta P_4/a_4) - a_3\ (\Delta x_2 - \Delta x_1) - b_3(\Delta P_4) = A_2(\Delta\dot{x}_2). \qquad (B.12)$$

Also, Equations (B.4), (B.6), (B.7) and (B.8) can be combined to give

$$(\upsilon_3/\beta)(\Delta\dot{P}_3) + A_3(\Delta\dot{x}_3) + c_3(\Delta P_3) =$$

$$a_1(\Delta x_2 - \Delta x_4) - b_1(\Delta P_3) - a_2(\Delta x_4 - \Delta x_2) - b_2(\Delta P_3). \qquad (B.13)$$

The Laplace transform can be obtained for each equation assuming zero initial conditions on all variations Δ. For simplicity the same notation is used for the time and Laplace domain forms of the changed variables. Equations (B.12) and (B.13) can be rewritten as

$$\{(\upsilon_3/\beta)s + c_3 + b_1 + b_2\}(\Delta P_3) = (a_1+a_2)(\Delta x_2 - \Delta x_4) - A_3(\Delta\dot{x}_3)s \qquad (B.14)$$

$$\{(1/a_4) + b_3\}(\Delta P_4) = a_3(\Delta x_1) - (A_2 s + a_3)(\Delta x_2). \qquad (B.15)$$

From Equations (B.1), (B.2), (B.11) and (B.15)

$$\{A_2 s + \frac{k_2}{A_2}\left(\frac{1}{a_4} + b_3\right) + a_3\}(\Delta x_2) = a_3 \frac{A_1}{k_1}(\Delta P_1) + \left(\frac{1}{a_4} + b_3\right)\frac{k_2}{A_2}\alpha(\Delta x_3).$$

Defining $B_1 = \frac{k_2}{A_2}\left(\frac{1}{a_4} + b_3\right)$, the above equation reduces to

$$(A_2 s + B_1 + a_3)(\Delta x_2) = \frac{a_3 A_1}{k_1}(\Delta P_1) + B_1 \alpha(\Delta x_3) \qquad \text{(B.16)}$$

From Equations (B.3), (B.11) and (B.14)

$$\left[\frac{\upsilon_3 M}{A_3 \beta} s^3 + \frac{M}{A_3}(c_3+b_1+b_2)s^2 + \left(\frac{\upsilon_3 k_3}{A_3 \beta} + A_3\right)s + \frac{3}{A_3}(c_3+b_1+b_2)\right](\Delta x_3) = (a_1+a_2)(\Delta x_2 - \Delta x_4).$$

Setting $B_2 = (c_3+b_1+b_2)$ and $B_3 = (\upsilon_3/A_3\beta)$ gives

$$\left[B_3 M s^3 + (M\, B_2/A_3)s^2 + (B_3 k_3 + A_3)s + (k_3 B_2/A3) + (a_1+a_2)\alpha\right](\Delta x_3) = (a_1+a_2)(\Delta x_2) \qquad \text{(B.17)}$$

Finally, Equations (B.16) and (B.17) can be combined to eliminate (Δx_2). The result is

$$\left[A_2 B_3 M s^4 + \left(B_3 B_4 M + \frac{A_2 B_2 M}{A_3}\right)s^3 + \left(\frac{B_2 B_4 M}{A_3} + A_2 B_5\right)s^2 + (B_4 B_5 + A_2 B_6)s + B_4 B_6 - B_1 (a_1+a_2)\alpha\right](\Delta x_3) = \frac{a_3 A_1}{k_1}(a_1+a_2)(\Delta P_1) \qquad \text{(B.18)}$$

where

$$B_4 = B_1 + a_3$$

$$B_5 = B_3k_3 + A_3$$

$$B_6 = (k_3B_2/A_3) + (a_1+a_2)\alpha.$$

Equation (B.18) is the final, linearized closed-loop transfer function, describing the relationship between (ΔP_1) and (Δx_3).

In order to be able to determine numerical values for the parameters of Equation (B.18), a steady-state operating point must be chosen and the numerical values for the corresponding x_{1i}, x_{2i}, x_{4i}, P_{3i}, P_{4i} and Q_{4i} found. It was mentioned in Appendix A that a Fortran IV program was developed to do this task. This program is listed in Appendix C. For the values of $\alpha = 0.32$ and $x_3^* = 2.90$ in., the following results were obtained at the steady-state operating point $x_3 = 4.0$ in.:

$x_{1i} = 4.14167$ in.

$x_{2i} = 4.17934$ in.

$x_{4i} = 4.18000$ in.

$P_{3i} = 103.00$ psi.

$P_{4i} = 5.690$ psi.

By using the parameter values presented in Table II, it is possible to substitute in values for all the parameters in Equation (B.18) and to arrive at a transfer function for a small deviation from a given steady-state operating point.

One parameter remains to be determined, and that is the spring constant in the equation of motion for the main piston (B.3). From Figure 3 it is seen that the load curve has a large negative slope in the vicinity of $x_3 = 4.0$ in. The load characteristic in this region can be approximated by

$$F_{ext} \simeq k_3 \, x_3.$$

The constant k_3 is negative and from Figure 3 is found as

$k_3 = -3{,}450$ lb/in.

Numerical values for the remaining parameters in Equation (B.18) are as follows:

$A_1 = (\pi/4)(d_x^2 - d_1^2) = 4.106 \text{ in}^2.$

$k_1 = 30$ lb/in.

$A_2 = 2.341 \text{ in}^2.$

$k_2 = 212$ lb/in.

$M = 1.669 \text{ lb}\cdot\text{sec}^2/\text{in}.$

$k_3 = -3{,}450$ lb/in.

$A_3 = 78.38 \text{ in}^2.$

$\upsilon_3/\beta = 344/110{,}000 = 3.127\cdot 10^{-3} \text{ in}^5/\text{lb}.$

$$c_3 = \frac{9.99^2 \, (0.001/2)^3}{12 \cdot 2.2\cdot 10^{-6}\cdot 0.8} = 0.186\cdot 10^{-3} \text{ in}^5/\text{sec}\cdot\text{lb}.$$

$a_1 = 0.625\cdot\pi\cdot 3.495\sqrt{(2/8\cdot 10^{-5})\cdot(300-103)} = 15.2\cdot 10^3 \text{ in}^2/\text{sec}.$

$a_2 = 0.625\cdot\pi\cdot 3.495\sqrt{(2/8\cdot 10^{-5})\cdot(103)} = 11.0\cdot 10^3 \text{ in}^2/\text{sec}.$

$a_3 = 0.625\cdot\pi\cdot 1.750\sqrt{(2/8\cdot 10^{-5})\cdot(5.69)} = 1.3\cdot 10^3 \text{ in}^2/\text{sec}.$

$$b_1 = 0.625 \cdot \pi \cdot 3.495\sqrt{(2/8 \cdot 10^{-5})}\ \frac{(x_{2i} - x_{4i} + U)}{2\sqrt{300 - 103}}$$

$$= 51.8 \cdot 10^{-3}\ \text{in}^5/\text{lb} \cdot \text{sec.}$$

$$b_2 = 0.625 \cdot \pi \cdot 3.495\sqrt{(2/8 \cdot 10^{-5})}\ \frac{(x_{4i} - x_{2i} + U)}{2\ \sqrt{103}}$$

$$= 142.2 \cdot 10^{-3}\ \text{in}^5/\text{lb} \cdot \text{sec.}$$

$$b_3 = 0.625 \cdot \pi \cdot 3.495\sqrt{(2/8 \cdot 10^{-5})}\ \frac{(\ x_{2i} - x_{1i})}{2\ \sqrt{5.69}}$$

$$= 4.29\ \text{in}^5/\text{lb} \cdot \text{sec.}$$

$$a_4 \quad \frac{128 \cdot 2.2 \cdot 10^{-6} \cdot 4}{\pi\,(3/16)^4} = 290.1 \cdot 10^{-3}\ \text{sec/in}^2.$$

$$\alpha = 0.32$$

$$B_1 = \frac{212}{2.341}\ (1/290.1 \cdot 10^{-3} + 4.29) = 700.67$$

$$B_2 = 0.186 \cdot 10^{-3} + 51.8 \cdot 10^{-3} + 142.2 \cdot 10^{-3} = 194.2 \cdot 10^{-3}$$

$$B_3 = \frac{344}{78.38 \cdot 110{,}000} = 39.9 \cdot 10^{-6}$$

$$B_4 = 700.67 + 1300 = 200.67$$

$$B_5 = 39.9 \cdot 10^{-6}\ (-3{,}450) + 78.38 = 78.24$$

$$B_6 = \frac{(-3{,}450) \cdot 194.2 \cdot 10^{-3}}{78.38} + 26.2 \cdot 10^3 \cdot 0.32 = 8.38 \cdot 10^3.$$

Using these values in Equation (B.18) yields the transfer function

$$\frac{\Delta x_3}{\Delta P_1} = \frac{4.662 \cdot 10^6}{155.9 \cdot 10^{-6} s^4 + 142.9 \cdot 10^{-3} s^3 + 191.44\ s^2 + 176.1 \cdot 10^3 + 10.89 \cdot 10^6} \tag{B.19}$$

A block diagram representing the system is shown in Figure 43. In this diagram the complete nonlinear load on the system is shown, but other parts of the system (valve characteristics) are in their linearized form for clarity.

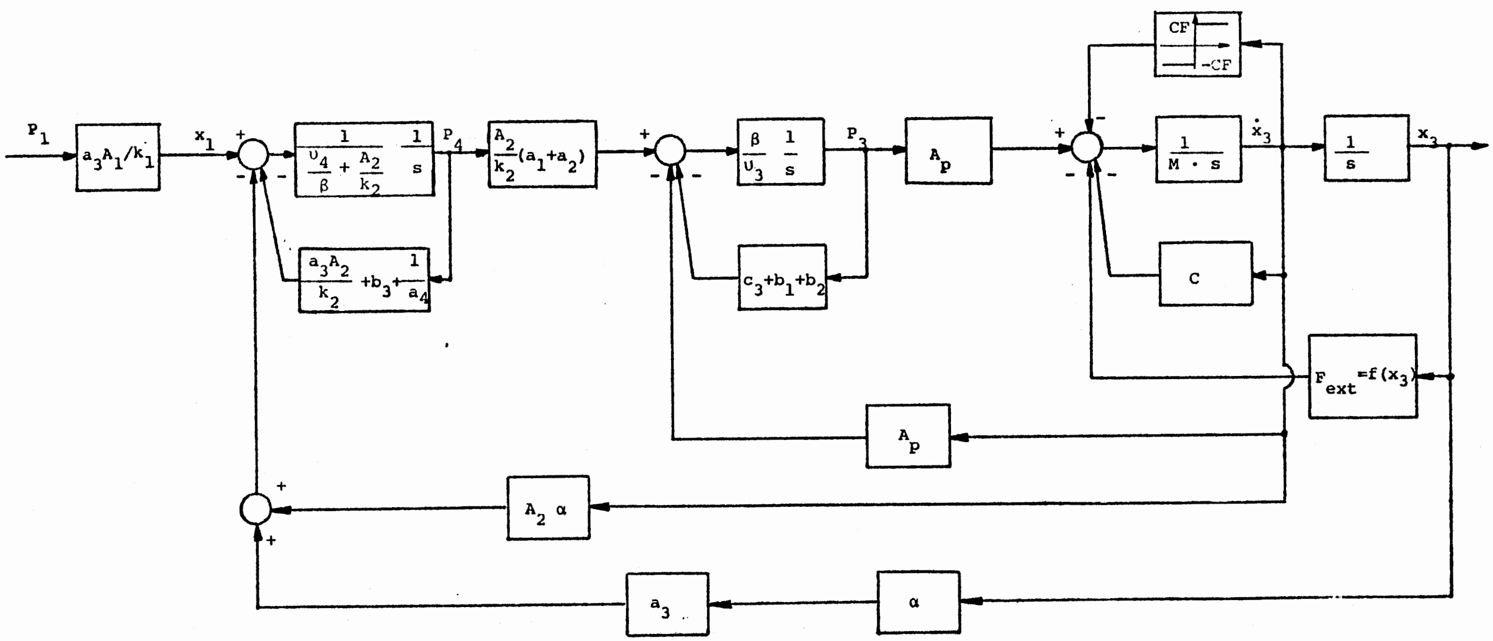


Figure 43. Block Diagram of the Uncompensated System.

APPENDIX C

THE STEADY-STATE SOLUTION

The complexity of the system of equations required that a computer program be developed to obtain the steady-state solution. This complexity was even more apparent when it was seen that the feedback coefficients, α and x_3^*, could not be determined directly from the system drawings.

Equations (A.1) through (A.11) must be solved iteratively. To start the solution, it is neccessary to choose a value of the piston position. For a particular value of piston position, x_3, a value of the steady-state load, F_{ext}, can be determined from Figure 3. This in turn gives a value of pressure under the piston, P_3, from Equation (A.10). Equation (A.11) can be used to determine a value of relay bushing displacement, x_4. Then Equations (A.8), (A.5), (A.6) and (A.7) can be combined to obtain a value of x_2. These latter equations combine to give a nonlinear equation in x_2, which has to be solved numerically. The program uses the Secant method to reach a solution to this equation and another nonlinear equation resulting from Equation (A.3). With a value for x_2 determined, Equation (A.2) can be used to find the pressure P_4 and Equation (A.3) can be used to approximate Q_4. There then remains only to use Equation (A.1) to find the control oil pressure, P_1.

A computer program was written based on the above sequence. The program was used to obtain the steady-state values of all system variables. These values provided the initial values for the dynamic simulation, as well as information to calculate the linearized coefficients in the linear simulation program. This program was also used to find values of the feedback coefficients, α and x_3^*, by comparing the calculated x_3 v.s. P_1 relation to the experimentally derived relations in Figure 4. This calculation was done for a range of coefficients, until a good match was obtained.

The program which follows consists of a main program and several subroutines. Subroutine STSOLN incorporates the equations from Appendix A, calling on FTABLE to interpolate in the load data table, and ROOT to solve the nonlinear equations. The subroutine INPAR calculates some parameters to be used in STSOLN, in order to make the implementation of the equations themselves simpler. Numerical values for the system physical parameters are set by a BLOCK DATA subprogram.

This program was set up to be run interactively on the OSU time sharing option, TSO, and was also developed with the future incorporation into the linear integration program in mind.

```
C
C-----------------------------------------------------------------
C
C       THIS PROGRAM FINDS STEADY STATE VALUES FOR THE
C       STATES TO BE USED IN THE DYNAMIC SIMULATION OF
C       HYDROMECHANICAL SERVO.
C       A VALUE OF X3 IS CHOSEN FOR THE STEADY STATE AND
C       THE PROGRAM CALCULATES THE OTHER VARIABLES AT THIS
C       PARTICULAR STEADY STATE.
C       IN TWO CASES AN ITERATIVE REGULA FALSI METHOD IS
C       USED TO SOLVE A NONLINEAR EQUATION.
C
C       ALLOCATE THE FILE LOAD.DATA UNDER FT07F001 BEFORE
C       RUNNING
C
C-----------------------------------------------------------------
        REAL PAR(20),X(12),X3TAB(29),P3TAB(29)
        COMMON /VAL/CD,RO,XMU,BETA,ALFA,CF3,C1,C2,D3,C4,C6,
       1W1,U,P2,XL3,XL4,XK1,XK2,XK3,XM3,XM4,R1,R3,V3I,V4I,
       1AR2,XST,FS,CR1,CR2
C
        ITMAX=20
        EPS=1.E-04
        NOTAB=27
C
C         INPUT THE STEADY STATE LOAD CURVE IN TABULAR FORM
C
        READ (7,202) (X3TAB(I),P3TAB(I),I=1,NOTAB)
C
C
        XST=2.1
C
        CALL INPAR(PAR)
C
        CALL STSOLN(PAR,ITMAX,EPS,XST,X,X3TAB,P3TAB,NOTAB)
C
C
  200   FORMAT(' ','INPUT XSTAR:')
  201   FORMAT(F6.3)
  202   FORMAT(2F6.2)
C
        STOP
        END
C
C
C         SOLVING FOR THE 12 STEADY STATE PARAMETERS
C
        SUBROUTINE STSOLN(PAR,ITMAX,EPS,XST,X,X3TAB,P3TAB
       1,NOTAB)
        REAL PAR(1),X(1),X3TAB(1),P3TAB(1)
C
C         INPUT A CHOSEN VALUE OF X3 FOR A STEADY STATE
C         CONDITION
C
        WRITE(10,12)PAR(8),XST
   12   FORMAT(///,' ',5X,'ALFA = ',F6.3,5X,'XST = ',F6.3,//)
C
C
        X(3)=.0
        DO 10 J=1,8
        X(3)=X(3)+1.
C
C       WRITE(6,223)
C223    FORMAT(' INPUT A VALUE FOR X3:')
C       READ(5,*)X(3)
C
        X(4)=PAR(8)*X(3)+XST
        X(6)=FTABLE(X3TAB,P3TAB,X(3),NOTAB)
C
        K=1
        XL=X(4)-.0015
        XR=X(4)+.0015
        P4=0.
C
        CALL ROOT(XL,XR,K,PAR,X(4),X(6),P4,ITMAX,EPS,X(2))
        X(7)=PAR(10)/PAR(6)*(X(2)-X(4))+PAR(13)/PAR(6)
C
        K=2
        XR=55.
        XL=40.
C
        CALL ROOT(XL,XR,K,PAR,X(4),X(6),X(7),ITMAX,EPS,X(9))
C
        X(10)=X(9)
        X(1)=X(2)-SQRT((X(9)/(PAR(2)*SQRT(X(7))))**2-PAR(16)**2)
        X(5)=PAR(5)/PAR(7)*X(1)
        X(8)=PAR(1)*SQRT(((X(2)-X(4)+PAR(11))**2+PAR(17)**2)
       1*(PAR(12)-X(6)))
        X(11)=PAR(1)*SQRT(((X(4)-X(2)+PAR(11))**2+PAR(17)**2)
       1*X(6))
        X(12)=X(8)-X(11)
C
        WRITE(10,353) (I,X(I),I=1,12)
  353   FORMAT(///,12(' X(',I2,') = ',G15.8,/))
   10   CONTINUE
        RETURN
        END
C
C       SOLVES ITERATIVELY AN EQUATION USING SECANT
C
        SUBROUTINE ROOT(X0SNG,X1SNG,K,PAR,X4SNG,P3SNG,P4SNG,
       1ITMAX,EPS,X)
        REAL*8 X0,X1,X4,P3,P4,DELTAX,DELTAF,F0,F1
        REAL PAR(1)
C
        X0=X0SNG
        X1=X1SNG
        X4=X4SNG
        P3=P3SNG
        PR=P4SNG
C
C
        F0=FUST(X0,K,PAR,X4,P3,P4)
        DELTAX=X1-X0
C
        DO 1 J=1,ITMAX
        F1=FUST(X1,K,PAR,X4,P3,P4)
        IF(DABS(F1).LE.EPS)GO TO 999
        DELTAF=F0-F1
        IF(DELTAF.EQ.0)GO TO 888
```

```
      DELTAX=F1/DELTAF*DELTAX
      X0=X1
      X1=X1+DELTAX
C     IF(DABS(DELTAX).LE.0.0001) GO TO 999
      F0=F1
 1    CONTINUE
 888  X=SNGL(X1)
C


      WRITE(6,20) K,LTMAX
      RETURN
 999  X=SNGL(X1)
 20   FORMAT('1','SECANT ITERATION FOR CASE ',I1,
     1' IN STEADY STATE SOLUTION DID NOT CONVERGE IN ',I3,
     2' ITERATIONS',//)
      RETURN
      END
C
C     GIVES THE FUNCTION EXPRESSIONS USED IN SUBROUTINE ROOT
C
      FUNCTION FUST(X,K,PAR,X4,P3,P4)
      REAL*8 X,X4,P3,P4,A,B,C,D,U,CR2,F2,FUST
      REAL PAR(1)
      A=PAR(1)
      B=PAR(3)
      C=PAR(4)
      D=PAR(5)
      U=PAR(11)
      CR2=PAR(17)
      P2=PAR(12)
C
C
      IF(K.EQ.2)GO TO 2
C
      FUST=B*P3-(A*(DSQRT(((X-X4+U)**2+CR2**2)*(P2-P3))-
     1DSQRT(((X4-X+U)**2+CR2**2)*P3)))
      GO TO 999
C
 2    FUST=P2-P4-C*X-C*D*X**2
C
 999  RETURN
      END
C
C     INTERPOLATES LINEARLY IN THE LOAD DATA TABLE
C
      FUNCTION FTABLE(VAR,FUNC,XX,M)
C
      REAL VAR(1),FUNC(1)
C
      IF(XX.LT.VAR(1)) FTABLE=FUNC(1)+(XX-VAR(1))*(FUNC(2)-
     1FUNC(1))/(VAR(2)-VAR(1))
      IF(XX.GE.VAR(M)) FTABLE=FUNC(M)+(XX-VAR(M))*(FUNC(M)-
     1FUNC(M-1))/(VAR(M)-VAR(M-1))
      IEND=M-1
      DO 20 I=1,IEND
      INT=I
      IF(XX-VAR(I)) 20,10,10
 10   IF(XX-VAR(I+1)) 30,20,20
 20   CONTINUE
      GO TO 40
 30   FTABLE=FUNC(INT)+(XX-VAR(INT))*(FUNC(INT+1)-FUNC(INT))/
     1(VAR(INT+1)-VAR(INT))
 40   CONTINUE
C
      RETURN
      END
C
C
C
C     SUBROUTINE INPAR CALCULATES THE INPARETER
C     VALUES USED IN THE FUNCTION EQUATIONS IN STSOLN
C
      SUBROUTINE INPAR(PAR)
      REAL PAR(1)
      COMMON /VAL/CD,RO,XMU,BETA,ALFA,CR3,D1,D2,D3,D4,D6,

     1W1,U,P2,XL3,XL4,XK1,XK2,XK3,XM3,XM4,R1,R3,V3I,V4I,
     1AR2,XST,FS,CR1,CR2
C
      PI=3.1415927
      PAR(1)=CD*PI*D6*SQRT(2./RO)
      PAR(2)=CD*PI*W1*SQRT(2./RO)
      PAR(3)=PI*D3*(CR3/2)**3/(12.*XMU*XL3)
      PAR(4)=128.*XMU*XL4/(PI*D4**4)
      PAR(5)=2.28*RO/(16.*PI*XMU*XL4)
      PAR(6)=AR2
      PAR(7)=PI*(D2**2-D1**2)/4.
      PAR(8)=ALFA
      PAR(9)=XK1
      PAR(10)=XK2
      PAR(11)=U
      PAR(12)=P2
      PAR(13)=FS
      PAR(14)=FFX1
      PAR(15)=D3**2*PI/4.
      PAR(16)=CR1
      PAR(17)=CR2
      PAR(18)=XST
      RETURN
      END
C
C
C     GIVING NUMERICAL VALUES TO THE CONSTANTS USED
C
      BLOCK DATA
      COMMON /VAL/CD,RO,XMU,BETA,ALFA,CR3,D1,D2,D3,D4,D6,
     1W1,U,P2,XL3,XL4,XK1,XK2,XK3,XM3,XM4,R1,R3,V3I,V4I,
     1AR2,XST,FS,CR1,CR2
C
      REAL CD/.625/,RO/8.E-05/,XMU/2.2E-06/,BETA/2.3E05/,
     1CR3/.001/,D1/.996/,D2/2.494/,D3/9.99/,D4/.1875/,
     2W1/1.75/,U/.002/,P2/300./,XL3/.8/,XL4/4./,D6/3.495/,
     3XK1/30./,XK2/212./,XK3/-3450./,XM3/.303/,XM4/1.3656/,
      REAL R3/15.375/,V3I/344./,V4I/11.6/,
     1AR2/2.348/,FS/13.5/
     1,CR1/.002/,CR2/.002/,ALFA/.32/
      END
```

APPENDIX D

LINEAR SYSTEM DYNAMIC SIMULATION PROGRAM

The following contains a brief description of the Fortran IV program which was developed to simulate the linearized version of the servo system model presented in Chapter III. A complete source listing of the program is provided at the end of this Appendix.

This program employs a fourth-order Runge-Kutta integration algorithm to obtain a numerical solution to the system of equations. The system model is fourth order, and so four equations are needed to calculate the time derivatives of the dynamic variables.

A main program sets the integration stepsize and final time, and calls upon a routine INCOND to solve the steady-state equations for a chosen initial position x_3. INCOND uses other routines to perform the neccessary calculations, including FTABLE to interpolate in the load data table and ROOT, which is a nonlinear algebraic equation solver. ROOT has to be called upon to solve for x_2 and Q_4, from Equations (A.4) and (A.3) respectively. The steady-state solution for all twelve variables is obtained from a user supplied value for x_3.

The function subprogram RUNGE is the actual integration routine. It is called by the main program and it is entered five times for each integration step. The main program also calls the subroutine DERFUN, which supplies the neccessary values of the time derivatives at each time t. The algebraic and dynamic equations in DERFUN are implemented using a set of parameters, K(i), from the subroutine PAR. Values of the system physical parameters are initialized using a BLOCK DATA subprogram.

The resulting dynamic simulation program is designed to be run interactively, using a Tektronix graphics terminal; the output is displayed in graphical form. A subprogram QCKPLT, available in the Mechanical Engineering program library, is used to do the graphics, including all scaling. The plotting subprogram prompts the user for the variable he wishes to display, giving all the dynamic variables as candidates for plotting.

The only input to the simulation program is the chosen steady-state point x_3, and then a few prompts regarding the output form.

Following is a source listing of the simulation program as it was used when obtaining the results presented in Chapter III.

```
C
C--------------------------------------------------------------
C      THIS PROGRAM IS A LINEAR SIMULATION OF A HYDRAULIC
C      SERVO SYSTEM.  IT USES A FOURTH ORDER RUNGE KUTTA
C      INTEGRATION PROCEDURE TO INTEGRATE THE 4TH ORDER
C      SYSTEM OF 1ST ORDER LINEAR DIFFERENTIAL EQUATIONS.
C      THE PROGRAM SOLVES A STEADY STATE CONDITION BEFORE
C      DOING THE DYNAMIC SIMULATION OF THE SYSTEM.  THIS
C      PROVIDES THE NECESSARY VALUES FOR THOSE COEFFICIENTS
C      IN THE DYNAMIC EQUATIONS THAT DERIVE FROM NONLINEARI-
C      TIES.  THE STEADY-STATE INITIAL CONDITION FOR THE
C      INTEGRATION IS FIXED.
C
C      ALLOCATE LOAD.DATA UNDER FT07F001
C      ALLOCATE AN OUTPUT FILE UNDER FT10F001
C
C--------------------------------------------------------------
       IMPLICIT REAL (A-H,O-Z)
       REAL K(20),XS(12),XSOL(4)
       REAL XST(12)
       INTEGER RUNGE
       REAL F(4),X(4)
       REAL X3(250),DX3(250),P3(250),P4(250),XT(250)
C
C      SET STEPSIZE H, INTEGRATION TIME TMAX, FREQUENCY OF
C      PRINTOUT IFREQ
C
       N=4
       H=.0005
       TMAX=0.1
       IFREQ=1
C
C      CALCULATING THE  STEADY STATE CON FOR USE IN THE
C      DYNAMIC EQUATIONS
C
       CALL INCOND(XS)
C
       WRITE(6,176)(I,XS(I),I=1,12)
 176   FORMAT(' THE STEADY STATE CONDITION IS: ',//,
      112(' X(',I2,')=',G15.8,/))
C
C
       CALL PAR(K,XS)
C
C      SET THE INITIAL CONDITIONS FOR X(1),X(2), ... , X(NX)
C
       T=0.
       X(1)=0.0
       X(2)=0.
       X(3)=0.
       X(4)=0.0
       ICOUNT=0
C
C      PRINT HEADINGS
C
       WRITE (10,201)
C
C        SET THE INITIAL CONDITION FOR THE INTEGRATION
C
         X3IN=4.0
         P3IN=103.0
         P4IN=5.69
         NSOL=1
C
C      CALCULATE AND PRINT SOLUTION AT TIME T
C
 10    XT(NSOL)=T
       X3(NSOL)=X3IN+X(1)
       DX3(NSOL)=0.+X(2)
       P3(NSOL)=P3IN+X(3)
       P4(NSOL)=P4IN+X(4)
       WRITE (10,202) T,X3(NSOL),DX3(NSOL),P3(NSOL),P4(NSOL)
C
C      CALL ON THE FOURTH ORDER RUNGE KUTTA FUNCTION
C
C
 11    M=RUNGE (N,X,F,T,H)
C
C      WHEN M=1 COMPUTE DERIVATIVE VALUES
C
       IF (M.NE.1) GO TO 13
C
C      THE SYSTEM OF FIRST ORDER DIFFERENTIAL EQUATIONS
C      OBTAINED FROM SUBROUTINE CERFUN
C
       CALL DERFUN (K,XS,X,F)
       GO TO 11
C
C
C      IF T EXCEEDS TMAX, TERMINATE INTEGRATION
C
 13    IF (T.LE.TMAX) GO TO 16
       GO TO 999
 16    ICOUNT=ICOUNT+1
C
C      PRINT RESULTS OR CALL DIRECTLY ON RUNGE
C
       IF (ICOUNT.NE.IFREQ) GO TO 11
       ICOUNT=0
       NSOL=NSOL+1
       GO TO 10
 999   WRITE(6,203)
       READ(6,204)M
       GO TO(1,2,3,4,5),M
 1     CALL DOCKPLT(XT,X3,NSOL,'TIME(SEC)$','X3(IN)$',
      1'LINEAR SIMULATION$',0,5,0)
       PAUSE
       GO TO 999
 2     CALL DOCKPLT(XT,DX3,NSOL,'TIME(SEC)$','DX3(IN/SEC)$',
      1'LINEAR SIMULATION$',0,5,0)
       PAUSE
       GO TO 999
 3     CALL DOCKPLT(XT,P3,NSOL,'TIME(SEC)$','P3(PSIG)$',
      1'LINEAR SIMULATION$',0,5,0)
       PAUSE
       GO TO 999
 4     CALL DOCKPLT(XT,P4,NSOL,'TIME(SEC)$','P4(FSIG)$',
      1'LINEAR SIMULATION$',C,5,0)
       PAUSE
       GO TO 999
C
C      FORMATS FOR OUTPUT STATEMENTS
C
```

```
201   FORMAT(1H ,T5,'T',T18,'X(1)',T33,'X(2)',T48,'X(3)',T63,
     1'X(4)',//)
202   FORMAT(1H ,F7.5,2X,4(G15.4))
203   FORMAT(' DO YOU WANT TO SEE THE SOLUTION PLOTTED?',/,
     1' ANS:      1=X3      2=DX3      3=P3      4=P4    VS. TIME'

     2,/,' NO PLOT=5')
204   FORMAT(I1)
5     STOP
      END
C
C-----------------------------------------------------------------
C     SUBROUTINE DERFUN DEFINES THE DYNAMIC EQUATIONS
C     TO BE INTEGRATED
C
C-----------------------------------------------------------------
      SUBROUTINE DERFUN(K,XS,X,DX)
C
      IMPLICIT REAL(A-H,O-Z)
      REAL K(1),XS(1)
      REAL    DX(1),X(1)
C     DEFINING THE INPUT PRESSURE
C
      XIN=.27
C
C     DEFINING THE ALGEBRAIC EQUATIONS
C
C
      X1=K(15)*XIN
      X2=K(19)/K(18)*X(4)+K(5)*X(1)
      DELX1=XS(2)-XS(4)+K(3)
      DELX2=XS(4)-XS(2)+K(3)
      DELX3=XS(2)-XS(1)
      Q3=K(1)*(K(2)*K(19)/K(18)*X(4)-DELX1*.5/K(2)*X(3))
      Q6=K(1)*(-K(4)*K(19)/K(18)*X(4)+DELX2*.5/K(4)*X(3))
      Q4=-X(4)/K(14)
      Q5=K(6)*(K(7)*(X2-X1)+.5*DELX3/K(7)*X(4))
C
C     DEFINING THE INTEGRATION EQUATIONS
C
      DX(1)=X(2)
      DX(2)=K(17)*X(3)-K(16)*X(1)-K(13)*X(2)
      DX(3)=1./K(9)*(Q3-Q6-K(10)*X(2)-K(11)*X(3))
      DX(4)=1./(K(12)+K(19))*(Q4-Q5-K(5)*K(18)*X(2))
C
      RETURN
      END
C
C-----------------------------------------------------------------
C     SUBROUTINE PAR EVALUATES THE PARAMETERS USED IN THE
C     DYNAMIC SIMULATION
C
C-----------------------------------------------------------------
C
      SUBROUTINE PAR(K,XS)
C
      IMPLICIT REAL(A-H,O-Z)
      REAL K(1),XS(1)
      COMMON /VAL/CD,RC,XMU,BETA,ALFA,CR3,D1,D2,D3,D4,D6,
     1W1,U,P2,XL3,XL4,XK1,XK2,XK3,XM3,XM4,R1,R3,V3I,V4I,
     2AR2,XST,FS,CR1,CR2
C     DEFINING THE PARAMETERS TO BE USED IN THE
C     SYSTEM EQUATIONS
C
      PI=3.1415927
      K(1)=CD*PI*D6*SQRT(2./RC)
      K(2)=SQRT(P2-XS(6))
      K(3)=U
      K(4)=SQRT(XS(6))
      K(5)=ALFA
      K(6)=CD*PI*W1*SQRT(2./RC)


      K(7)=SQRT(XS(7))
      K(8)=XS(2)-XS(1)
      K(9)=V3I/BETA
      K(10)=PI/4.*D3**2
      K(11)=PI*D3*(CR3/2.)**3/(12.*XMU*XL3)
      K(12)=V4I/BETA
      K(13)=XML*XL3*PI*D3/(CR3/2.*(XM3+XM4))
      K(14)=128.*XMU*XL4/(PI*D4**4)*(1.+2.*2.28*RC*XS(9)/
     1(16.*PI*XMU*XL4))
      K(15)=PI/4.*(D2**2-D1**2)/XK1
      K(16)=XK3/(XM3+XM4)
      K(17)=PI/4.*D3**2/(XM3+XM4)
      K(18)=AR2
      K(19)=AR2**2/XK2
      K(20)=P2
      WRITE(10,15) XS(5),ALFA,XK3
15    FORMAT(//,'          P1= ',F6.3,'    ALFA= ',F6.3,'    XK3= ',
     1F10.3,//)
      RETURN
      END
C
C-----------------------------------------------------------------
C     INITIALIZATION OF DATA VALUES FOR THE SYSTEM MODEL
C
C-----------------------------------------------------------------
      BLOCK DATA
C
      COMMON /VAL/CD,RO,XMU,BETA,ALFA,CR3,D1,D2,D3,D4,D6,
     1W1,U,P2,XL3,XL4,XK1,XK2,XK3,XM3,XM4,R1,R3,V3I,V4I,
     2AR2,XST,FS,CR1,CR2
C
      REAL CD/.625/,RO/8.E-05/,XML/2.2E-06/,BETA/2.3E05/,
     1CR3/.001/,D1/.996/,D2/2.494/,D3/9.95/,D4/.1875/,
     2D6/3.495/,W1/1.75/,U/.002/,P2/300./,XL3/.8/,XL4/4./,
     3XK1/30./,XK2/212./,XK3/-3449./,XM3/.303/,XM4/1.3656/
      REAL V3I/344./,V4I/11.6/,
     1AR2/2.348/,XST/3.33/,FS/13.5/,
     2CR1/.001/,CR2/.001/,
     3ALFA/.45/
      END
C
```

```
C-----------------------------------------------------------------
C
C     THE FUNCTION RUNGE EMPLOYS THE FOURTH ORDER RUNGE KUTTA
C     METHOD WITH KUTTA'S COEFFICIENTS TO INTEGRATE A SYSTEM OF
C     N SIMULTANEOUS FIRST ORDER ORDINARY DIFFERENTIAL EQUATIONS
C     F(J)=DY(J)/DX,J=1,2.....N ACROSS ONE STEP OF LENGTH H IN THE
C     INDEPENDENT VARIABLE X, SUBJECT TO INITIAL CONDITIONS Y(J),
C     J=1.2.....N.  EACH F(J), THE DERIVATIVE OF Y(J), MUST BE COM
C     PUTED FOUR TIMES PER INTEGRATION STEP BY THE CALLING PROGRAM
C     THE FUNCTION MUST BE CALLED FIVE TIMES PER STEP (PASS(1)...
C     PASS(5)) SO THAT THE INDEPENDENT VARIABLE VALUE X AND THE
C     SOLUTION VALUES Y(1)...Y(N) CAN BE UPDATED USING THE RUNGE
C     KUTTA ALGORITHM.  M IS THE PASS COUNTER.   RUNGE RETURNS AS
C     ITS VALUE 1 TO SIGNAL THAT ALL DERIVATIVES BE EVALUATED OR 0
C     TO SIGNAL THAT THE INTEGRATIONS PROCESS FOR THE CURRENT STEP
C     IS FINISHED.  SAVEY(J) IS USED TO SAVE THE INITIAL VALUE
C     OF Y(J) AND PHI(J) IS THE INCREMENT FUNTION FOR THE J(TH)
C     EQUATION
C
C-----------------------------------------------------------------
C
      FUNCTION RUNGE(N,Y,F,X,H)
C
      IMPLICIT REAL(A-H,O-Z)
      REAL Y,F,X,H


      INTEGER RUNGE
      DIMENSION PHI(50),SAVEY(50),Y(1),F(1)
      DATA M/0/
C
      M=M+1
      GO TO (1,2,3,4,5), M
C          .....PASS 1......
    1 RUNGE=1
      RETURN
C
C          .....PASS 2.....
    2 DO 22 J=1,N
      SAVEY(J)=Y(J)
      PHI(J)=F(J)
   22 Y(J)=SAVEY(J) + 0.5*H*F(J)
      X=X + 0.5*H
      RUNGE=1
      RETURN
C
C          ......PASS 3......
    3 DO 33 J=1,N
      PHI(J)=PHI(J)+2.0*F(J)
   33 Y(J)=SAVEY(J) + 0.5*H*F(J)
      RUNGE=1
      RETURN
C
C          ......PASS 4 .....
    4 DO 44 J=1,N
      PHI(J)=PHI(J) + 2.0*F(J)
   44 Y(J)=SAVEY(J)+H*F(J)
      X=X+0.5*H
      RUNGE=1
      RETURN
C          .....PASS 5 .....
    5 DO 55 J=1,N
   55 Y(J)=SAVEY(J) + (PHI(J)+F(J))*H/6.0
      M=0
      RUNGE=0
      RETURN
C
      END
C
C-----------------------------------------------------------------
C
C     THIS PROGRAM FINDS STEADY STATE VALUES FOR THE
C     STATES TO BE USED IN THE DYNAMIC SIMULATION OF
C     HYDROMECHANINCAL SERVO.
C     A VALUE OF X3 IS CHOSEN FOR THE STEADY STATE AND
C     THE PROGRAM CALCULATES THE OTHER VARIABLES AT THIS
C     PARTICULAR STEADY STATE.
C     IN TWO CASES AN ITERATIVE REGULA FALSI METHOD IS
C     USED TO SOLVE A NONLINEAR EQUATION.
C
C     ALLOCATE THE FILE LOAD.DATA UNDER F107F001 BEFORE
C     RUNNING
C
C-----------------------------------------------------------------
C
      SUBROUTINE INCOND(X)
C
      REAL PAR(20),X(12),X3TAB(29),P3TAB(29)
      COMMON /VAL/CD,RO,XMU,BETA,ALFA,CR3,C1,C2,D3,C4,C6,
     1W1,U,P2,XL3,XL4,XK1,XK2,XK3,XM3,XM4,R1,R3,V3I,V4I,
     1AR2,XST,FS,CR1,CR2
C


      ITMAX=20
      EPS=1.E-04
      NCTAB=27
C
C       INPUT THE STEADY STATE LOAD CURVE IN TABULAR FORM
C
      READ (7,202) (X3TAB(I),P3TAB(I),I=1,NCTAB)
C
C
C
      CALL INPAR(PAR)
C
      CALL STSOLN(PAR,ITMAX,EPS,XST,X,X3TAB,P3TAB,NCTAB)
C
C
  200 FORMAT(' ','INPUT XSTAR:')
  201 FORMAT(F6.3)
  202 FORMAT(2F6.2)
C
      RETURN
      END
C
C-----------------------------------------------------------------
C       SOLVING FOR THE 12 STEADY STATE PARAMETERS
C
C-----------------------------------------------------------------
C
      SUBROUTINE STSOLN(PAR,ITMAX,EPS,XST,X,X3TAB,P3TAB
     1,NCTAB)
```

```
C
      REAL PAR(1),X(1),X3TAB(1),P3TAB(1)
C
C       INPUT A CHOSEN VALUE OF X3 FOR A STEADY STATE
  C     CONDITION
C
C
      WRITE(6,223)
 223  FORMAT(' INPUT A VALUE FOR X3:')
      READ(5,*)X(3)
C
      X(4)=PAR(8)*X(3)+XST
      X(6)=FTABLE(X3TAB,P3TAB,X(3),NOTAB)
C
      K=1
      XL=X(4)-.002
      XR=X(4)+.002
      P4=0.
C
      CALL ROOT(XL,XR,K,PAR,X(4),X(6),P4,ITMAX,EPS,X(2))
      X(7)=PAR(10)/PAR(6)*(X(2)-X(4))+PAR(13)/PAR(6)
C
      K=2
      XR=55.
      XL=40.
C
      CALL ROOT(XL,XR,K,PAR,X(4),X(6),X(7),ITMAX,EPS,X(9))
C
      X(10)=X(9)
      X(1)=X(2)-SQRT((X(9)/(PAR(2)*SQRT(X(7))))**2-PAR(16)**2)
      X(5)=PAR(9)/PAR(7)*X(1)
      X(8)=PAR(1)*SQRT(((X(2)-X(4)+PAR(11))**2+PAR(17)**2)
     1*(PAR(12)-X(6)))
      X(11)=PAR(1)*SQRT(((X(4)-X(2)+PAR(11))**2+PAR(17)**2)
     1*X(6))
      X(12)=X(8)-X(11)
      RETURN

      END
C
C-----------------------------------------------------------------
C       SOLVES ITERATIVELY AN EQUATION USING SECANT
C
C-----------------------------------------------------------------
C
      SUBROUTINE ROOT(X0SNG,X1SNG,K,PAR,X4SNG,P3SNG,P4SNG,
     1ITMAX,EPS,X)
      REAL*8 X0,X1,X4,P3,P4,DELTAX,DELTAF,F0,F1
      REAL PAR(1)
C
      X0=X0SNG
      X1=X1SNG
      X4=X4SNG
      P3=P3SNG
      P4=P4SNG
C
C
      F0=FUST(X0,K,PAR,X4,P3,P4)
      DELTAX=X1-X0
C
      DO 1 J=1,ITMAX
      F1=FUST(X1,K,PAR,X4,P3,P4)
      IF(DABS(F1).LE.EPS)GO TO 999
      DELTAF=F0-F1
      IF(DELTAF.EQ.0)GO TO 888
      DELTAX=F1/DELTAF*DELTAX
      X0=X1
      X1=X1+DELTAX
      WRITE(6,330) X1,F1
 330  FORMAT(' ',2G15.9)
C     IF(DABS(DELTAX).LE.0.0001) GO TO 999
      F0=F1
 1    CONTINUE
 888  X=SNGL(X1)
C
      WRITE(6,20)K,ITMAX
      RETURN
 999  X=SNGL(X1)
 20   FORMAT('1','SECANT ITERATION FOR CASE ',I1,
     1' IN STEADY STATE SOLUTION DID NOT CONVERGE IN ',I3,
     2' ITERATIONS',//)
      RETURN
      END
C
C-----------------------------------------------------------------
C       GIVES THE FUNCTION EXPRESSIONS USED IN SUBROUTINE ROOT
C
C-----------------------------------------------------------------
      FUNCTION FUST(X,K,PAR,X4,P3,P4)
      REAL*8 X,X4,P3,P4,A,B,C,D,U,CR2,P2,FUST
C
      REAL PAR(1)
      A=PAR(1)
      B=PAR(3)
      C=PAR(4)
      D=PAR(5)
      U=PAR(11)
      CR2=PAR(17)
      P2=PAR(12)
C
      IF(K.EQ.2)GO TO 2
C
      FUST=B*P3-(A*(DSQRT(((X-X4+U)**2+CR2**2)*(P2-P3))-
     1DSQRT(((X4-X+U)**2+CR2**2)*P3)))

      GO TO 999
C
2     FUST=P2-P4-C*X-C*D*X**2
C
999   RETURN
      END
C
C-----------------------------------------------------------------
C       INTERPOLATES LINEARLY IN THE LOAD DATA TABLE
C
C-----------------------------------------------------------------
C
      FUNCTION FTABLE(VAR,FUNC,XX,M)
C
      REAL VAR(1),FUNC(1)
```

```
C
      IF(XX.LT.VAR(1)) FTABLE=FUNC(1)+(XX-VAR(1))*(FUNC(2)-
     1FUNC(1))/(VAR(2)-VAR(1))
      IF(XX.GE.VAR(M)) FTABLE=FUNC(M)+(XX-VAR(M))*(FUNC(M)-
     1FUNC(M-1))/(VAR(M)-VAR(M-1))
      IEND=M-1
      DO 20 I=1,IEND
      INT=I
      IF(XX-VAR(I)) 20,10,10
   10 IF(XX-VAR(I+1)) 30,20,20
   20 CONTINUE
      GO TO 40
   30 FTABLE=FUNC(INT)+(XX-VAR(INT))*(FUNC(INT+1)-FUNC(INT))/
     1(VAR(INT+1)-VAR(INT))
   40 CONTINUE
C
      RETURN
      END
C
C-----------------------------------------------------------------
C     SUBROUTINE PARAM CALCULATES THE PARAMETER
C     VALUES USED IN THE FUNCTION EQUATIONS IN STSCLN
C
C-----------------------------------------------------------------
      SUBROUTINE INPAR(PAR)
C
      REAL PAR(1)
      COMMON /VAL/CD,RO,XMU,BETA,ALFA,CR3,D1,D2,D3,D4,D6,
     1W1,U,P2,XL3,XL4,XK1,XK2,XK3,XM3,XM4,R1,R3,V3I,V4I,
     1AR2,XSI,FS,CR1,CR2
C
      PI=3.1415927
      PAR(1)=CD*PI*D6*SQRT(2./RO)
      PAR(2)=CD*PI*W1*SQRT(2./RO)
      PAR(3)=PI*D3*(CR3/2)**3/(12.*XMU*XL3)
      PAR(4)=128.*XMU*XL4/(PI*D4**4)
      PAR(5)=2.28*RO/(16.*PI*XMU*XL4)
      PAR(6)=AR2
      PAR(7)=PI*(D2**2-D1**2)/4.
      PAR(8)=ALFA
      PAR(9)=XK1
      PAR(10)=XK2
      PAR(11)=U
      PAR(12)=P2
      PAR(13)=FS
      PAR(14)=FEXT
      PAR(15)=D3**2*PI/4.
      PAR(16)=CR1
      PAR(17)=CR2
      PAR(18)=XSI
      RETURN

      END
```

APPENDIX E

THE DYSIMP COMPUTER MODEL OF THE SYSTEM

The statement of the problem solved by DYSIMP was discussed in Chapter IV, along with the variable definitions applied to the servo system model. Following is a more detailed discussion of how the model was set up in Fortran IV code to accomodate the use of DYSIMP to solve for the time response. Also contained in this Appendix is a source listing of the code written for DYSIMP, which follows the conventions introduced in section 4.2.

DYSIMP is set up as a processor which takes a certain kind of Fortran code and builds up a continous program from it. It takes care of invoking a Fortran compiler and the neccessary procedures for loading and executing the whole program. The user need not supply more than a minimum number of statements which define the mathematical model to be solved. For a simple problem these statements would most likely be confined totally within the subroutine DERFUN, which is called upon by DYSIMP to evaluate the time derivatives of the dynamic variables Y. DYSIMP takes care of defining the common areas needed and other standard statements within the routines defined by DYSIMP. It then depends on the complexity of the problem being solved, how much additional code the user has to supply.

In every case, the subroutine DERFUN has to be supplied. In

DERFUN, the values of the algebraic variables X(i) are first calculated based on the variable values from the previous time step, and then the time derivatives of the dynamic variables are calculated, using the X's and the parameters P(i).

A subroutine PAR, which is not a standard DYSIMP routine, was supplied in order to calculate the various coefficients from the equations derived in section 2.2. DYSIMP provides for these coefficients to be supplied as input, but this was not considered to be convenient in the present case. This subroutine is invoked by the main program before the integration routine DYS2S is called, in order to have all the parameters defined before the time response calculations start.

Subroutine XVAL is a standard DYSIMP routine; the default version consists only of a return statement. XVAL is called by DYSIMP before DERFUN at the beginning of each time step, so it is suited for performing calculations that are required once each time step. Therefore, XVAL was chosen as the routine for solving for the flowrate Q_4 from Equation (2.3), which involves solving numerically a second-order algebraic equation in Q_4. A Newton-Rapson algorithm was chosen to perform these calculations.

Another subroutine, not standard for DYSIMP, was supplied to interpolate in the load data table. This function routine, FTABLE, will interpolate linearly between data points, in a table which is read in by the subroutine PAR.

The DYSIMP data input defines the number of variables to be integrated NY, the number of parameters defined NP, and also the number of algebraic variables defined NX. The data input also contains the final integration time ENDTIM, the integration time step DELT, and the initial conditions for the dynamic variables. The initial conditions were obtained beforehand from the steady-state solution program described in Appendix C. The data input also supplies instructions as to the type of output generated by DYSIMP. All of this standard input is read by DYSIMP input processing routines.

Following is a listing of the source Fortran code, along with the JCL control language cards needed to run the job on the OSU IBM 370 computer. Also included are the control cards needed to have DYSIMP load the tabulated output into a TSO data set for later plotting on the Tektronix terminal.

```
//U11893A  JOB (11893,440-7E-3438),'DYSIMP',TIME=(0,05),
// CLASS=F,MSGCLASS=X
/*PASSWORD ?
// EXEC DYSIMP2
//DYSIMP.FT11F001 DD DSN=U11893A.NO2286.DATA,
//   DISP=(NEW,CATLG),SPACE=(TRK,(10,5),RLSE),
//   UNIT=SYSTSO,DCB=(RECFM=FB,LRECL=133,BLKSIZE=5985)
//DYSIMP.MODELIN DD *
*MAIN
C
C *****************************************************************
C *                                                               *
C *    DYNAMIC SIMULATION OF A HYDRAU - MECHANICAL SERVC          *
C *    SYSTEM.                                                    *
C *    DEVELOPED AT OKLAHOMA STATE UNIVERSITY 1981-1982           *
C *    BY ASLAUG HARALDSDOTTIR AS A PART OF A MASTER'S            *
C *    PROJECT UNDER THE SUPERVISING OF DR. KARL N. REID.         *
C *                                                               *
C *    THE NONLINEAR SIMULATION PROGRAM DYSIMP.2 IS USED          *
C *    TO SOLVE THE SYSTEM OF 4 FIRST ORDER ORDINARY              *
C *    DIFFERENTIAL EQUATIONS.  A 4TH ORDER RUNGE KUTTA           *
C *    INTEGRATION ALGORITHM IS CHOSEN.                           *
C *    DYSIMP.2 REQUIRES CERTAIN INPUT VALUES IN NAMELIST         *
C *    FORMAT AND THIS VERSION PRODUCES OUTPUT IN TABULAR         *
C *    AND PRINTER PLOT FORMAT.                                   *
C *    STANDARD DYSIMP.DATAIN INPUT:                              *
C *      NY     :  NUMBER OF DYNAMIC VARIABLES                    *
C *      NP     :  NUMBER OF SYSTEM PARAMETERS DEFINED            *
C *      NX     :  NUMBER OF ALGEBRAIC VARIABLES DEFINED          *
C *      IRK    :  SETS TYPE OF INTEGRATION ALGORITHM             *
C *      FNDTIM :  FINAL SIMULATION TIME                          *
C *      DELT   :  INTEGRATION TIME STEP                          *
C *      PRDEL  :  OUTPUT PRINT AND PLOT TIME STEP                *
C *      YI     :  INITIAL VALUES FOR DYNAMIC VARIABLES           *
C *      PLOT1  :  VARIABLES PLOTTED IN PLOT 1                    *
C *      PLOT2  :  VARIABLES PLOTTED IN PLOT 2                    *
C *      TABLE1 :  VARIABLES TABULATED IN TABLE 1                 *
C *                                                               *
C *    ADDITIONAL PROGRAM INPUT (DYSIMP.SYSIN):                   *
C *      LOAD ON SYSTEM IN TABULAR FORM                           *
C *                                                               *
C *****************************************************************
C
      IREAD=0
      IWRITE=0
      IRETRN=0
      NODISK=-1
C
C *****    SET SYSTEM VARIABLES USING SUBROUTINE PAR         *****
C
      CALL PAR
C
C *****    CALL ON DYSIMP.2 TO PERFORM SYSTEM SIMU-          *****
C *****    LATION                                            *****
C
      CALL DYS2S(IREAD,IWRITE,IRETRN,NODISK,IEND)
      IF(IEND.NE.0)  STOP 16
      STOP
      END
C
C
C
C *****************************************************************
C *                                                               *
C *    SUBROUTINE DERFUN DEFINES THE MODEL EQUATIONS, BOTH        *
C *    ALGEBRAIC AND DYNAMIC.  IT IS CALLED BY DYSIMP.2.          *
C *                                                               *
C *      X3,F3   :  THIS PAIR GIVES THE LOAD ON THE SYSTEM        *
C *                 AS A FUNCTION OF PISTON DISPLACEMENT          *
C *                 OUTPUT                                        *
C *      NOTAB   :  NUMBER OF POINTS AT WHICH LOAD IS GIVEN       *
C *      X7      :  UPDATED VALUE OF X(7) FROM XVAL               *
C *      PINT    :  INITIAL VALUE OF INPUT CONTROL PRESSURE       *
C *      DELP    :  STEP INCREASE IN INPUT PRESSURE               *
C *      DX31    :  CONTAINS THE VALUE OF Y(2) FROM PRIOR         *
C *                 TIME STEP                                     *
C *      X       :  VECTOR OF ALGEBRAIC VARIABLES                 *
C *      Y       :  VECTOR OF DYNAMIC VARIABLES                   *
C *      DY      :  VECTOR OF VARIABLES TO BE INTEGRATED          *
C *                                                               *
C *****************************************************************
C
*DECK DERFUN
      REAL X3(32),F3(32)
      COMMON /TABLE/ X3,F3,NOTAB
      COMMON /NEWT/ X7,PINT,DELP,DX31
C
C *****    DEFINE STEP INPUT IN CONTROL PRESSURE             *****
C
      XIN=PINT+DELP
      X(1)=P(1)*XIN
      X(3)=P(4)*Y(1)+P(5)
      X(2)=P(2)*Y(4)-P(3)+X(3)
      X(7)=X7
C
C *****    DEFINING THE ORIFICE FLOW VARIABLES               *****
C
C
      DELX42=X(3)-X(2)
C
      A=P(8)-Y(3)
      B=ABS(A)
      C=(X(2)-X(3)+P(7))**2+P(5)**2
      D=ABS(Y(3))
      E=(X(3)-X(2)+P(7))**2+P(5)**2
      F=ABS(Y(4))
      G=(X(2)-X(1))**2+P(10)**2
C
      IF(DELX42.GE.P(7))        GO TO 2
C
      IF(ABS(DELX42).GE.P(7))  GO TO 1
C
      X(4)=B/A*P(6)*SQRT(B*C)
      X(6)=D/Y(3)*P(6)*SQRT(D*E)
C
      GO TO 3
C
1     X(4)=B/A*P(6)*SQRT(B*C)
      X(6)=D/Y(3)*P(6)*P(9)*SQRT(E)
C
      GO TO 3
C
2     X(4)=B/A*P(6)*P(9)*SQRT(B)
      X(6)=D/Y(3)*P(6)*SQRT(D*E)
```

```
C
3       IF((X(2)-X(1)).LE.0)        GO TO 4
C
        X(5)=F/Y(4)*P(11)*SCRT(F*G)

        GO TO 5
4       X(5)=F/Y(4)*P(11)*P(10)*SCRT(F)
C
C *****   IMPLEMENTING THE LOAD ON THE SYSTEM WITH          *****
C *****   COULOMB FRICTION                                  *****
C
5       FEXT=P(14)*FTABLE(X3,F3,Y(1),NOTAB)
C
        FSTN=P(23)*P(14)
        FDRIVE=P(14)*Y(3)-FEXT
C
        DX32=DX31
        DX31=Y(2)
C
9       IF(Y(2).EQ.0.) GO TO 7
C
        IF(DX32*Y(2).GE.0.) GO TO 8
C
        IF( ABS(FDRIVE) .GT. FSTN)     GO TO 8
C
12      DY(1)=0.
        DY(2)=0.
        GO TO 11
C
7       IF(ABS(FDRIVE) .GT. FSTN)  GO TO 10
        DY(1)=.0
        DY(2)=0.
        GO TO 11
C
8       FEXT=FEXT+P(23)*P(14)*SIGN(1.0,Y(2))
C
C *****   DEFINING THE DYNAMIC EQUATIONS                  *****
C
10      DY(1)=Y(2)
        DY(2)=(P(14)*Y(3)-P(15)*Y(2)-FEXT)/P(16)
11      DY(3)=(X(4)-X(6)-P(14)*Y(2)-P(18)*Y(3))*P(17)/
       1(P(14)*Y(1)+P(24))
        DY(4)=(X(7)-X(5)-P(20)*Y(2))/P(21)
C ****************************************************************
C *                                                              *
C *     SUBROUTINE XVAL IS CALLED AT THE BEGINNING               *
C *     TIME STEP BY DYSIMP.2                                    *
C *     IT IS USED HERE TO SOLVE ITERATIVELY FOR THE FLOW        *
C *     Q4 USING A NEWTONS ALGORITHM                             *
C *                                                              *
C *        XC    :  AT FIRST IT CONTAINS A GUESS FOR THE FLOW    *
C *                 THEN XVAL UPDATES IT                         *
C *                                                              *
C ****************************************************************
C
*DECK XVAL
        COMMON /NEWT/ X0,PINT,DELF,DX31
        NTOL=15
        XTOL=1.E-04
        FTOL=1.E-04
        DO 20 N=1,NTOL
        FX0=P(12)*X0+P(12)*P(13)*X0**2+Y(4)-P(4)
C
        IF(ABS(FX0).LT.FTOL)        RETURN
C
        DERIV=P(12)+2.*P(12)*P(13)*X0
C
        IF(DERIV.EQ.0)                 GO TO 999
C
        DELTAX=FX0/DERIV
        X0=X0-DELTAX

C
        IF(ABS(DELTAX).LT.XTOL)    RETURN
C
20      CONTINUE
        WRITE(6,30)  NTOL
30      FORMAT(' ',//,' THE NEWTONS METHOD DID NOT CONVERGE',
       1' IN ',I2,' STEPS',///)
        RETURN
999     WRITE(6,31)
31      FORMAT(' ',//,' IN THE NEWTON METHOD, DERIV = 0',//)
*COPY
C
C ****************************************************************
C *                                                              *
C *     SUBROUTINE PAR SETS THE PARAMETER VALUES TO BE USED      *
C *     IN DEFINING THE MODEL IN DERFUN                          *
C *                                                              *
C ****************************************************************
C
        SUBROUTINE PAR
*CALL  ALL
        REAL X3(32),F3(32)
        COMMON /TABLE/ X3,F3,NOTAB
        DATA  CD/.625/,RO/8.E-05/,XMU/2.2E-06/,BETA/1.1E05/,
       1D1/.996/,D2/2.494/,D3/9.99/,D4/.1875/,D6/3.495/,
       2W1/1.75/,U/.002/,P2/300./,XL3/.8/,XL4/4./,
       3XK1/30./,XK2/212./,XK3/1048./,XM3/.303/,XM4/1.3656/,
       4V31/30./,V41/11.6/,AR2/2.341/,XS1/2.10/,FS/13.5/,
       5CR1/.002/,CR2/.002/,CR3/.001/,ALFA/.90/
C
        PI=22./7.
C
        P(1)=PI/4.*(D2**2-D1**2)/XK1
        P(2)=AR2/XK2
        P(3)=FS/XK2
        P(4)=ALFA
        P(5)=XS1
        P(6)=CD*PI*D6*SQRT(2./RO)
        P(7)=U
        P(8)=P2
        P(9)=CR2
        P(10)=CR1
        P(11)=CD*PI*W1*SQRT(2./RO)
        P(12)=128.*XMU*XL4/(PI*D4**4)
        P(13)=2.28*RO/(16.*PI*XMU*XL4)
        P(14)=PI/4.*D3**2
        P(15)=XMU*XL3*PI*D3/(CR3/2)
        P(16)=XM3+XM4
        P(17)=BETA
```

```
      P(18)=PI*D3*(CR3/2)**3/(12.*XMU*XL3)
      P(19)= AR2**2/XK2
      P(20)=AR2*ALFA
      P(21)=V4I/BETA+AR2**2/XK2
      P(22)=XK3
      P(24)=V3I
C
C *****    INPUT THE LOAD DATA IN TABULAR FORM            *****
C
      NOTAB=32
      READ(5,10) (X3(I),F3(I),I=1,NOTAB)
 10   FORMAT(2F6.2)
C
      P(23)=5.0
C
C
      RETURN


      END
C
C ************************************************************
C *                                                          *
C *   FUNCTION FTABLE INTERPOLATES LINEARLY IN THE LOAD      *
C *   DATA TABLE                                             *
C *                                                          *
C ************************************************************
C
C
C     INTERPOLATES LINEARLY IN THE LOAD DATA TABLE
C
      FUNCTION FTABLE(VAR,FUNC,XX,M)
C
      REAL VAR(1),FUNC(1)
C
      IF(XX.LT.VAR(1)) FTABLE=FUNC(1)+(XX-VAR(1))*(FUNC(2)-
     1FUNC(1))/(VAR(2)-VAR(1))
      IF(XX.GE.VAR(M)) FTABLE=FUNC(M)+(XX-VAR(M))*(FUNC(M)-
     1FUNC(M-1))/ (VAR(M)-VAR(M-1))
      IEND=M-1
      DO 20 I=1,IEND
      INT=I
      IF(XX-VAR(I)) 20,10,10
 10   IF(XX-VAR(I+1)) 30,20,20
 20   CONTINUE
      GO TO 40
 30   FTABLE=FUNC(INT)+(XX-VAR(INT))*(FUNC(INT+1)-FUNC(INT
     1))/(VAR(INT+1)-VAR(INT))
 40   CONTINUE
C
      RETURN
      END
C
C *****    SET INITIAL AND PROGRAM CONTROL PARAMETERS    *****
C
      BLOCK DATA
      COMMON /NEWT/ X0,PINT,DELP,DX31
      REAL X0/48.9/,PINT/41.57/,DELP/.27/,DX31/1.0/
      END
```

```
//DYSIMP.DATAIN DD *
NONLINEAR MODEL OF HYDRAULIC SERVO SYSTEM
 &DATAIN NY=4, NP=24, NX=7, IRK=4, ENDTIM=.1, DELT=.0005
 , PRDEL=.0005, YI(1)=4.,0.0,103.,5.69,  PLOT1=1,2, PLOT2=
 3, TABLE1=1,2,3,4, &END
//DYSIMP.SYSIN DD *
1.00   67.5
1.25   63.5
1.50   60.5
1.75   56.5
2.0    53.5
2.25   51.0
2.5    49.0
2.75   47.5
3.0    46.5
3.25   45.5
3.35   45.0
3.4   131.0
3.5   127.0
3.75  115.0
4.0   103.0
4.25   91.0
4.50   80.5
4.75   74.5
5.0    71.0
5.25   69.0

5.50   67.0
5.75   67.0
6.0    67.0
6.25   67.0
6.3   150.0
6.50  135.5
6.75  124.0
7.00  116.5
7.25  111.0
7.50  107.0
7.75  105.0
8.00  103.0
 8.5    0.0
```

APPENDIX F

A PROGRAM FOR GRAPHIC REPRESENTATION OF RESULTS FROM SIMULATIONS

In order to represent the results from DYSIMP in the same form as those from the linear simulation program, a Fortran program was written to plot them in Tektronix graphics. Following is a source listing of this program. It will read in the results from a DYSIMP run, which have been loaded into a TSO data set. It also will read in the results from a linear simulation run, and give the user the option of comparing the two on the same graph. It scales the two functions to be plotted so that both will fit neatly onto the same graph, and writes a few key parameter values of the system at the side of the plot. A TSO data set containing these parameter values has to be available to the program in 'Namelist form', and the program reads these values at the beginning of each run. It gives the user the option of changing any of these values to match those used in the particular case being plotted.

This program makes use of the subroutine QCKPLT, which was available in the Mechanical Engingeering computer library. The program listing follows.

```
C      THIS PROGRAM CAN PLOT TWO FUNCTIONS ON THE SAME
C      GRID WITH APPROPRIATE SCALING.
C      IT READS ONE SET OF X AND Y VECTORS FROM
C      UNIT 11. THE OTHER FROM UNIT 12.
C      THE CORRESPONDING SYSTEM PARAMETERS ARE READ FROM
C      UNIT 13
C
C
       DIMENSION X1(502),F1(502),X2(502),F2(502),XSCL(2)
       DIMENSION Y1(502,4),Y2(502,4),YSCL(2)
       INTEGER YLAB(5),ANS,YES
C
       NAMELIST /PARAM/CD,RO,XML,BETA,D1,D2,D3,D4,D6,W1,
      1U,P2,XL3,XL4,XK1,XK2,XK3,XM3,XM4,V3I,V4I,AK2,XST,
      2FS,CR1,CR2,CR3,ALFA,CF
C
       DATA YLAB/'X3$','DX3$','P3$','PR$'/,YES/'Y'/
C
C      READ IN THE SYSTEM PARAMETERS
C
       READ (13,PARAM)
       WRITE (6,PARAM)
C
       WRITE(6,117)
 117   FORMAT(' DO YOU WANT TO CHANGE ANY OF THE PARAMETERS?')
C
       READ(6,106)ANS
 106   FORMAT(A1)
C
       IF(ANS.NE.YES) GO TO 24
C
       WRITE(6,173)
 173   FORMAT( ' INPUT PARAMETERS:',/)
C
       READ (5,PARAM)
C
C      INPUT RESULTS OF A NONLINEAR SIMULATION
C
 24     K=0
 6     K=K+1
       READ(11,100) X1(K),(Y1(K,J),J=1,4)
 100   FORMAT(1X,G10.8,4(G15.12))
       IF(Y1(K,1).NE.0.) GO TO 6
       K=K-1
C
       WRITE(6,102)
 102   FORMAT(' DO YOU WANT TO COMPARE LINEAR AND ',
      1'NONLINEAR RESULTS?')
       READ(5,103) ANS
 103   FORMAT(A1)
       IF(ANS.NE.YES) GO TO 3
C
       I=0
C
C      INPUT RESULTS OF A LINEAR SIMULATION
C
 8     I=I+1
       READ (12,101) X2(I),(Y2(I,J),J=1,4)
 101   FORMAT(1X,F9.5,4(G15.4))
       IF(Y2(I,1).NE.0.0) GO TO 8
       I=I-1
C
       NP2=I
 3     NP1=K
C
 500   WRITE(6,104)
 104   FORMAT(' WHICH VARIABLE DO YOU WANT TO PLOT?',
      1'X3=1.   DX3=2.   P3=3.   P4=4')
       READ(5,105) M
 105   FORMAT(I1)
C
C     LOAD THE CHOSEN FUNCTION VALUES TO BE PLOTTED
C      INTO A VECTOR
C
       DO 21 I=1,NP1
          F1(I)=Y1(I,M)
 21       CONTINUE
C
       IF(ANS .NE. YES) GO TO 7
C
       DO 22 I=1,NP2
          F2(I)=Y2(I,M)
 22       CONTINUE
C
C      FIND THE LARGEST AND SMALLEST X AXIS VALUES
C
 7     XSCL(1)=0.
       XSCL(2)=AMAX1(X1(NP1),X2(NP2))
C
C      FIND LARGEST AND SMALLEST FI VALUES
C
       YSCL(1)=1.0E75
       YSCL(2)=-1.0E75
       DO 1 I=1,NP1
          YSCL(1)=AMIN1(YSCL(1),F1(I))
          YSCL(2)=AMAX1(YSCL(2),F1(I))
 1     CONTINUE
C
       IF(ANS.NE.YES) GO TO 4
C
C      FIND LARGER MIN AND MAX OF F1 V.S. F2
C
       DO 2 J=1,NP2
          YSCL(1)=AMIN1(YSCL(1),F2(J))
          YSCL(2)=AMAX1(YSCL(2),F2(J))
 2     CONTINUE
C      SCALING VALUES PUT INTO VECTORS TO BE PLOTTED
C
 4     X1(NP1+1)=XSCL(1)
       X1(NP1+2)=XSCL(2)
       F1(NP1+1)=YSCL(1)
       F1(NP1+2)=YSCL(2)
C
       X2(NP2+1)=XSCL(1)
       X2(NP2+2)=XSCL(2)
       F2(NP2+1)=YSCL(1)
       F2(NP2+2)=YSCL(2)
       CALL OCKPLT(X1,F1,NP1,'TIME SEC$',YLAB(M),
      1'HYDRAULIC SERVO$',1,5,'NONE ')
```

```
C
      IF(ANS.NE.YFS) GO TO 5
C
      CALL QCKPLT(X2,F2,NP2,'TIME$',YLAB(M),
     1'HYDRAULIC SERVOS',2,5,'LIN')
C
C     WRITE MORE INFORMATION ON THE PLOT
C
 5    WRITE(6,11)ALFA

 11   FORMAT(' ',////,60X,'  ALFA=',F6.3)
      WRITE(6,12)U
 12   FORMAT(' ',60X,' U=',F6.4)
      WRITE(6,13)CR1
 13   FORMAT(' ',60X,' CR1=',F6.4)
      WRITE(6,14) CR2
 14   FORMAT(' ',60X,' CR2=',F6.4)
      WRITE(6,15) BETA
 15   FORMAT(' ',60X,' BET=',F7.0)
      WRITE(6,16) XMU
 16   FORMAT(' ',60X,' MU=',F8.7)
      XMASS=XM3+XM4
      WRITE(6,17) XMASS
 17   FORMAT(' ',60X,' MASS=',F5.3)
      WRITE(6,18) CF
 18   FORMAT(' ',60X,' CF=',F4.1,24(/))
      PAUSE
C
      GO TO 500
C
      END
```

APPENDIX G

DERIVATION OF A TRANSFER FUNCTION FOR THE COMPENSATED SYSTEM

The equations neccessary to describe the dynamic behavior of the system with the pressure compensation are discussed in Chapter V. The linearized equations are listed below.

$$A_1(\Delta P_1) = k_1(\Delta x_1) \qquad (G.1)$$

$$A_2(\Delta P_4) = k_2(\Delta x_2 - \Delta x_4) \qquad (G.2)$$

$$M(\Delta x_3) + k_3(\Delta x_3) = A_3(\Delta P_3) \qquad (G.3)$$

$$\Delta Q_7 = \Delta Q_3 - \Delta Q_6 \qquad (G.4)$$

$$\Delta Q_7 = \Delta Q_8 + (\upsilon_3/\beta)(\Delta \dot{P}_3) + A_3(\Delta \dot{x}_3) + c_3(\Delta P_3) \qquad (G.5)$$

$$\Delta Q_4 - \Delta Q_5 = A_2(\Delta \dot{x}_2) \qquad (G.6)$$

$$\Delta Q_3 = a_1(\Delta x_2 - \Delta x_4) - b_1(\Delta P_3) \qquad (G.7)$$

$$\Delta Q_6 = a_2(\Delta x_4 - \Delta x_2) + b_2(\Delta P_3) \qquad (G.8)$$

$$\Delta Q_5 = a_3(\Delta x_2 - \Delta x_1) + b_3(\Delta P_4) \qquad (G.9)$$

$$-(\Delta P_4) = a_4(\Delta Q_4) \qquad (G.10)$$

$$\Delta Q_8 = a_5(\Delta P_3 - \Delta P_8) \qquad (G.11)$$

$$\Delta Q_8 = \upsilon_a/(k\ P_{8i})(\Delta \dot{P}_8) \qquad (G.12)$$

$$\Delta x_4 = \alpha(\Delta x_3). \qquad (G.13)$$

Equations (G.11) and (G.12) can be combined to obtain

$$\Delta Q_8 = 1 - \frac{1}{B_7 s + 1} a_5(\Delta P_3) \qquad (G.14)$$

where

$$B_7 = \frac{\upsilon_a}{k\ P_{8i}\ a_5}$$

$$a_5 = \frac{\pi d_7^4}{128\mu L_7}.$$

Using the same parameter simplifications as in Appendix B and combining the equations in a similar manner, the following expression relating control pressure input and displacement output can be derived:

$$\Big[A_2B_3B_7M\ s^5 + (B_3B_7B_4M + \frac{A_2B_8M}{A_3})s^4 + (\frac{B_4B_8M}{A_3} + A_2C_2)s^3 +$$

$$(B_4C_2 + A_2C_3)s^2 + (B_4C_3 + A_2B_6 - C_5B_7)s + B_4B_6 - C_5\Big] (\Delta x_3) =$$

$$\frac{a_3A_1}{k_1}(a_1+a_2)(B_7\ s + 1)\ (\Delta P_1) \qquad (G.15)$$

where

$$B_8 = B_2B_7 + a_5B_7 + \upsilon_3/\beta$$

$$C_2 = B_2M/A_3 + A_3B_7 + B_3k_3B_7$$

$$C_3 = k_3B_8/A_3 + (a_1+a_2)B_7 + A_3$$

$$C_5 = B_1\ \alpha(a_1+a_2).$$

The system closed loop transfer function is of the form

$$\frac{\Delta x_3}{\Delta P_1} = \frac{K\,(\tau_p\ s + 1)}{a_o\ s^5 + a_1\ s^4 + a_2\ s^3 + a_3\ s^2 + a_4\ s + a_5}\ . \qquad (G.16)$$

APPENDIX H

DYSIMP MODEL WITH THE COMPENSATION

In section 5.5 it was shown that only one equation has to be added to the set already implemented in DYSIMP to accomodate the added compensation. Also, a small change needs to be made in the equation for the time derivative of the pressure P_3. It is desirable to be able to choose whether or not the compensation is included in the model, and therefore some provision has to be made to allow this. Simply setting the parameters of the compensation equal to zero would result in division by zero in Equation (5.13). To overcome this difficulty, a dynamic index DYN was introduced into the model; the value is preset by the BLOCK DATA subprogram that already exists in the code. A value of 1.0 will include the pressure compensation; any other value will result in an uncompensated system model being simulated. The implementation includes two sets of equations for $\dot{P}_3$ and $\dot{P}_8$, one for uncompensated system and the other for the system with compensation.

A source listing of the program as it was run follows.

```
//U11893A  JOB (11893,440-78-3438),'DYSIMP',TIME=(0,05),
// CLASS=E,MSGCLASS=X
/*PASSWORD ?
// EXEC DYSIMP2
//DYSIMP.FT11F001 DD DSN=U11893A.NC3092.DATA,
//   DISP=(NEW,CATLG),SPACE=(TRK,(10,5),RLSE),
//   UNIT=SYSDSC,DCB=(RECFM=FB,LRECL=133,BLKSIZE=5985)
//DYSIMP.MODELIN DD *
*MAIN
C
C *************************************************************
C *                                                           *
C *   DYNAMIC SIMULATION OF A HYDRAU - MECHANICAL SERVO       *
C *   SYSTEM.                                                 *
C *   DEVELOPED AT OKLAHOMA STATE UNIVERSITY 1981-1982        *
C *   BY ASLAUG HARALDSDOTTIR AS A PART OF A MASTER'S         *
C *   PROJECT UNDER THE SUPERVISING OF DR. KARL N. REID.      *
C *                                                           *
C *   THE NONLINEAR SIMULATION PROGRAM DYSIMP.2 IS USED       *
C *   TO SOLVE THE SYSTEM OF 5 FIRST ORDER ORDINARY           *
C *   DIFFERENTIAL EQUATIONS.  A 4TH ORDER RUNGE KUTTA        *
C *   INTEGRATION ALGORITHM IS CHOSEN.                        *
C *   DYSIMP.2 REQUIRES CERTAIN INPUT VALUES IN NAMELIST      *
C *   FORMAT AND THIS VERSION PRODUCES OUTPUT IN TABULAR      *
C *   AND PRINTER PLOT FORMAT.                                *
C *   STANDARD DYSIMP.DATAIN INPUT:                           *
C *     NY     :  NUMBER OF DYNAMIC VARIABLES                 *
C *     NP     :  NUMBER OF SYSTEM PARAMETERS DEFINED         *
C *     NX     :  NUMBER OF ALGEBRAIC VARIABLES DEFINED       *
C *     IRK    :  SETS TYPE OF INTEGRATION ALGORITHM          *
C *     FNDTIM:   FINAL SIMULATION TIME                       *
C *     DELT   :  INTEGRATION TIME STEP                       *
C *     PRDEL  :  OUTPUT PRINT AND PLOT TIME STEP             *
C *     YI     :  INITIAL VALUES FOR DYNAMIC VARIABLES        *
C *     PLOT1  :  VARIABLES PLOTTED IN PLOT 1                 *
C *     PLOT2  :  VARIABLES PLOTTED IN PLOT 2                 *
C *     TABLE1:   VARIABLES TABULATED IN TABLE 1              *
C *                                                           *
C *   ADDITIONAL PROGRAM INPUT (DYSIMP.SYSIN):                *
C *     LOAD ON SYSTEM IN TABULAR FORM                        *
C *                                                           *
C *************************************************************
C
      IREAD=0
      IWRITE=0
      IRETRN=0
      NODISK=-1
C
C *****   SET SYSTEM VARIABLES USING SUBROUTINE PAR      *****
C
      CALL PAR
C
C *****   CALL ON DYSIMP.2 TO PERFORM SYSTEM SIMU-       *****
C *****   LATION                                         *****
C
      CALL DYS2S(IREAD,IWRITE,IRETRN,NODISK,IEND)
      IF(IEND.NE.0)  STOP 16
      STOP
      END
C
C
C
```

```
C *************************************************************
C *                                                           *
C *   SUBROUTINE DERFUN DEFINES THE MODEL EQUATIONS, BOTH     *
C *   ALGEBRAIC AND DYNAMIC.  IT IS CALLED BY DYSIMP.2.       *
C *                                                           *
C *     X3,F3   :  THIS PAIR GIVES THE LOAD ON THE SYSTEM     *
C *                AS A FUNCTION OF PISTON DISPLACEMENT       *
C *                OUTPUT                                     *
C *     NOTAB   :  NUMBER OF POINTS AT WHICH LOAD IS GIVEN    *
C *     X7      :  UPDATED VALUE OF X(7) FROM XVAL            *
C *     PINT    :  INITIAL VALUE OF INPUT CONTROL PRESSURE    *
C *     DELP    :  STEP INCREASE IN INPUT PRESSURE            *
C *     DX31    :  CONTAINS THE VALUE OF Y(2) FROM PRIOR      *
C *                TIME STEP                                  *
C *     DYN     :  IF DYN=1.0 THEN DYNAMIC PRESSURE COMPEN-   *
C *                SATION IS INCLUDED                         *
C *     X       :  VECTOR OF ALGEBRAIC VARIABLES              *
C *     Y       :  VECTOR OF DYNAMIC VARIABLES                *
C *     DY      :  VECTOR OF VARIABLES TO BE INTEGRATED       *
C *                                                           *
C *************************************************************
C
*DECK DERFUN
      REAL X3(32),F3(32)
      COMMON /TABLE/ X3,F3,NOTAB
      COMMON /NEWT/ X7,PINT,DELP,DX31,DYN
C
C *****    DEFINE STEP INPUT IN CONTROL PRESSURE          *****
C
      XIN=PINT+DELP
      X(1)=P(1)*XIN
      X(3)=P(4)*Y(1)+P(5)
      X(2)=P(2)*Y(4)-P(3)+X(3)
      X(7)=X7
C
C *****    DEFINING THE ORIFICE FLOW VARIABLES            *****
C
C
      DELX42=X(3)-X(2)
C
      A=P(8)-Y(3)
      B=ABS(A)
      C=(X(2)-X(3)+P(7))**2+P(9)**2
      D=ABS(Y(3))
      E=(X(3)-X(2)+P(7))**2+P(9)**2
      F=ABS(Y(4))
      G=(X(2)-X(1))**2+P(10)**2
C
      IF(DELX42.GE.P(7))      GO TO 2
C
      IF(ABS(DELX42).GE.P(7)) GO TO 1
C
      X(4)=B/A*P(6)*SQRT(B*C)
      X(6)=D/Y(3)*P(6)*SQRT(D*E)
C
      GO TO 3
C
```

```
1       X(4)=B/A*P(6)*SQRT(B*C)
        X(6)=D/Y(3)*P(6)*P(9)*SQRT(C)
C
        GO TO 3
C
2       X(4)=B/A*P(6)*P(9)*SQRT(E)
        X(6)=D/Y(3)*P(6)*SQRT(D*E)
C
3       IF((X(2)-X(1)).LE.0)      GO TO 4


C
        X(5)=F/Y(4)*P(11)*SQRT(F*G)
        GO TO 5
4       X(5)=F/Y(4)*P(11)*P(10)* SQRT(F)
C
C *****    IMPLEMENTING THE LOAD ON THE SYSTEM WITH        *****
C *****    COULOMB FRICTION                                *****
C
5       FEXT=P(14)*FTABLE(X3,F3,Y(1),NUTAB)
C
        FSTN=P(23)*P(14)
        FDRIVE=P(14)*Y(3)-FEXT
C
        DX32=DX31
        DX31=Y(2)
C
9       IF(Y(2).EQ.0.) GO TO 7
C
        IF(DX32*Y(2).GE.0.) GO TO 8
C
        IF( ABS(FDRIVE) .GT. FSTN)     GO TO 8
C
12      DY(1)=0.
        DY(2)=0.
        GO TO 11
C
7       IF(ABS(FDRIVE) .GT. FSTN)  GO TO 10
        DY(1)=.0
        DY(2)=0.
        GO TO 11
C
8       FEXT=FEXT+P(23)*P(14)*SIGN(1.0,Y(2))
C
C *****    DEFINING THE DYNAMIC EQUATIONS                    *****
C
10      DY(1)=Y(2)
        DY(2)=(P(14)*Y(3)-P(15)*Y(2)-FEXT)/P(16)
11      IF(DYN.EQ. 1.) GO TO 13
        DY(3)=(X(4)-X(6)-P(14)*Y(2)-P(18)*Y(3))*P(17)/
       1(P(14)*Y(1)+P(24))
        DY(4)=(X(7)-X(5)-P(20)*Y(2))/P(21)
        DY(5)=0.
        RETURN
13      DY(3)=(X(4)-X(6)-P(27)*(Y(3)-Y(5))-P(14)*Y(2)-
       1P(18)*Y(3))*
       1P(18)*Y(3))*P(17)/(P(14)*Y(1)+P(24))
        DY(4)=(X(7)-X(5)-P(20)*Y(2))/P(21)
        DY(5)=P(27)*(Y(3)-Y(5))/(P(28)/P(17)+P(29)/
       1(P(30)*Y(5)))
```

```
C ***************************************************************
C *                                                             *
C *     SUBROUTINE XVAL IS CALLED AT THE BEGINNING              *
C *     TIME STEP BY DYSIMP.2                                   *
C *     IT IS USED HERE TO SOLVE ITERATIVELY FOR THE FLOW       *
C *     C4 USING A NEWTONS ALGORITHM                            *
C *                                                             *
C *       XO    :  AT FIRST IT CONTAINS A GUESS FOR THE FLOW    *
C *                THEN XVAL UPDATES IT                         *
C *                                                             *
C ***************************************************************
C
*DECK XVAL
        COMMON /NEWT/ XO,PINT,DELP,DX31,DYN
        NTOL=15
        XTOL=1.E-04
        FTOL=1.E-04


        DO 20 N=1,NTOL
        FXO=P(12)*XO+P(12)*P(13)*XO**2+Y(4)-P(8)
C
        IF(ABS(FXO).LT.FTOL)       RETURN
C
        DERIV=P(12)+2.*P(12)*P(13)*XO
C
        IF(DERIV.EQ.0)                GO TO 999
C
        DELTAX=FXO/DERIV
        XO=XO-DELTAX
C
        IF(ABS(DELTAX).LT.XTOL)   RETURN
C
20      CONTINUE
        WRITE(6,30)   NTOL
30      FORMAT(' ',//,' THE NEWTONS METHOD DID NOT CONVERGE',
       1' IN ',I2,' STEPS',///)
        RETURN
999     WRITE(6,31)
31      FORMAT(' ',//,' IN THE NEWTON METHOD, DERIV = 0',//)
*COPY
C
C ***************************************************************
C *                                                             *
C *     SUBROUTINE PAR SETS THE PARAMETER VALUES TO BE USED     *
C *     IN DEFINING THE MODEL IN DERFUN                         *
C *                                                             *
C ***************************************************************
C
        SUBROUTINE PAR
*CALL ALL
        REAL X3(32),F3(32)
        COMMON /TABLE/ X3,F3,NUTAB
        DATA  CC/.625/,KO/8.E-05/,XML/2.2E-06/,BETA/.5E05/,
       1D1/.996/,D2/2.494/,D3/9.99/,C4/.1875/,D6/3.495/,
       2W1/1.75/,L/.001/,P2/300./,XL3/.8/,XL4/4./,
       3XK1/30./,XK2/212./,XK3/1048./,XM3/.303/,XM4/1.3656/,
       4V3I/30./,V4I/11.6/,AR2/2.341/,XS1/2.90/,FS/13.5/,
       5CR1/.000/,CR2/.000/,CR3/.001/,ALFA/.32/
        DATA D7/.2/,D8/5./,XL7/10./,XLOIL/5./,XLAIR/5./,
       1PCLYC/1.4/
C
        PI=22./7.
```

```
C
      P(1)=PI/4.*(D2**2-D1**2)/XK1
      P(2)=AR2/XK2
      P(3)=FS/XK2
      P(4)=ALFA
      P(5)=XST
      P(6)=CD*PI*D6*SQRT(2./RO)
      P(7)=U
      P(8)=P2
      P(9)=CR2
      P(10)=CR1
      P(11)=CD*PI*W1*SQRT(2./RO)
      P(12)=128.*XMU*XL4/(PI*D4**4)
      P(13)=2.28*RC/(16.*PI*XMU*XL4)
      P(14)=PI/4.*D3**2
      P(15)=XMU*XL3*PI*D3/(CR3/2)
      P(16)=XM3+XM4
      P(17)=BETA
      P(18)=PI*D3*(CR3/2)**3/(12.*XMU*XL3)
      P(19)=AR2**2/XK2
      P(20)=AR2*ALFA
      P(21)=V4I/BETA+AR2**2/XK2


      P(22)=XK3
      P(24)=V3I
C
C *****    INPUT THE LOAD DATA IN TABULAR FORM              *****
C
      NOTAB=32
      READ(5,10) (X3(I),F3(I),I=1,NOTAB)
 10   FORMAT (2F6.2)
C
      P(23)=0.0
      P(27)=PI*D7**4/(128.*XMU*XL7)
      P(28)=PI/4.*D8**2*XLOIL
      P(29)=PI/4.*D8**2*XLAIR
      P(30)=PCLYC
C
C
      RETURN
      END
C
C ***************************************************************
C *                                                             *
C *   FUNCTION FTABLE INTERPOLATES LINEARLY IN THE LOAD         *
C *   DATA TABLE                                                *
C *                                                             *
C ***************************************************************
C
C
C     INTERPOLATES LINEARLY IN THE LOAD DATA TABLE
C
      FUNCTION FTABLE(VAR,FUNC,XX,M)
C
      REAL VAR(1),FUNC(1)
C
      IF(XX.LT.VAR(1)) FTABLE=FUNC(1)+(XX-VAR(1))*(FUNC(2)-
     1FUNC(1))/(VAR(2)-VAR(1))
      IF(XX.GE.VAR(M)) FTABLE=FUNC(M)+(XX-VAR(M))*(FUNC(M)-
     1FUNC(M-1))/(VAR(M)-VAR(M-1))
      IEND=M-1
      DO 20 I=1,IEND
      INT=I
      IF(XX-VAR(I)) 20,10,10
 10   IF(XX-VAR(I+1)) 30,20,20
 20   CONTINUE
      GO TO 40
 30   FTABLE=FUNC(INT)+(XX-VAR(INT))*(FUNC(INT+1)-FUNC
     1(INT))/(VAR(INT+1)-VAR(INT))
 40   CONTINUE
C
      RETURN
      END
C
C *****    SET INITIAL AND PROGRAM CONTROL PARAMETERS     *****
C
      BLOCK DATA
      COMMON /NEWT/ XO,PINT,DELP,DX31,DYN
      REAL XO/48.9/,PINT/30.26/,DELP/.27/,DX31/1.0/DYN/1.0/
      END
//DYSIMP.DATAIN DD *
NONLINEAR MODEL OF HYDRAULIC SERVO SYSTEM
 &DATAIN NY=5, NP=30, NX=7, IRK=4, ENDTIM=.1, DELT=.0001
 ,PRDEL=0.0005,YI(1)=4.,0.0,103.,5.69,103.
 ,TABLE1=1,2,3,4,5, &END
//DYSIMP.SYSIN DD *
1.00   67.5
1.25   63.5
1.50   60.5

1.75   56.5
2.0    53.5
2.25   51.0
2.5    49.0
2.75   47.5
3.0    46.5
3.25   45.5
3.35   45.0
3.4   131.0
3.5   127.0
3.75  115.0
4.0   103.0
4.25   91.0
4.50   80.5
4.75   74.5
5.0    71.0
5.25   69.0
5.50   67.0
5.75   67.0
6.0    67.0
6.25   67.0
6.3   150.0
6.50  135.5
6.75  124.0
7.00  116.5
7.25  111.0
7.50  107.0
7.75  105.0
8.00  103.0
 8.5    0.0
```

VITA

Aslaug Haraldsdottir

Candidate for the Degree of

Master of Science

Thesis: TRANSIENT FLOW STABILIZATION OF AN HYDRAULIC SERVOMECHANISM

Major Field: Mechanical Engineering

Biographical:

Personal Data: Born in Reykjavik, Iceland, October 21, 1956, the daughter of Haraldur Thordarson and Maria A. Gudmundsdottir.

Education: Graduated from Kvennaskolinn i Reykjavik High School, Reykjavik, Iceland in May 1972; completed the University Entrance Examination (Studentsprof) from the Hamrahlid College, Reykjavik, Iceland in May 1975 with Music as the main field of study; extended that degree in May 1976 to include a major in Physics; received the Bachelor of Science degree in Mechanical Engineering from the University of Iceland, Reykjavik, Iceland in June 1980; completed the requirements for the Master of Science degree at Oklahoma State University in May 1982.

Professional Experience: Steel construction worker, Stalsmidjan, Reykjavik, Iceland, 1977; Welder, Stalver h.f., Reykjavik, Iceland, 1978; Design Engineer, Verkfraedistofa Gudmundar og Kristjans, Reykjavik, Iceland, 1980; Graduate Teaching and Research Assistant, School of Mechanical and Aerospace Engineering, Oklahoma State University, August 1980-May 1982.