PARTICULATE-INDUCED SEIZURE OF PISTON-BORE ASSEMBLIES

By

TOKUNOSUKE ITO

Bachelor of Science in Mechanical Engineering
Oklahoma State University
Stillwater, Oklahoma
May, 1982

Master of Science
Oklahoma State University
Stillwater, Oklahoma
May, 1985

Submitted to the Faculty of the Graduate College
of the Oklahoma State University
in partial fulfillment of the requirements for
the Degree of
DOCTOR OF PHILOSOPHY
July, 1987

PARTICULATE-INDUCED SEIZURE OF

PISTON-BORE ASSEMBLIES

Thesis Approved:

E. C. Fitch
Thesis Adviser

David G. Lilley

Paul M. Moretti

L. G. Rockley

Norman N. Durham
Dean of the Graduate College

ACKNOWLEDGEMENTS

I would like to express my appreciation to people who supported me in the course of pursuing my Ph.D.

I am very thankful to Dr. Fitch, my major adviser, for his encouragement and seasoned advice. I would also like to express my appreciation to Dr. David Lilley, Dr. Peter Moretti, and Dr. Mark Rockley for their keen advice which helped me break through many difficulties encountered throughout my research.

To my collegues at Fluid Power Research Center, Dr. I.T. Hong, Michael Arrington, Luis Isaza, Ibrahim Khalil, and Gabriel Silva, I extend my thanks for their assistance in fabricating the experimental apparatus and conducting the experiments.

I would also like to thank Ailisha O'Sullivan for helping to edit and format the figures of the dissertation.

Finally, I extend my utmost appreciation to my wife, Linda, for typing this dissertation and providing constant encouragement and support throughout my Ph.D. program.

TABLE OF CONTENTS

LIST OF FIGURES

Figure		Page

Figure | Page

Figure		Page

LIST OF TABLES

CHAPTER I

INTRODUCTION

There are two distinct modes of seizure. One is the seizure caused by welding between two sliding surfaces. The other seizure is caused by a third body which enters and wedges between two sliding surfaces. This third body bridges the clearance and prevents sliding motion. This is called particulate-induced seizure. There are two types of this mode of seizure. One type occurs when the third body, called a transfer element, is a result of surface contact wear and is a mixture of the two sliding surface materials. These transfer elements are known to grow in size as they combine together, and they cause seizure when they bridge across the two sliding surfaces. The other type occurs when the third body is composed of material different than that of the sliding surfaces. As the fluid media flows between the sliding surfaces, it transports foreign particulates which induce seizure. The following study examines the nature of this second type of particulate-induced seizure known to occur among piston-bore assemblies.

For years, piston-bore assemblies have been used for various industrial applications, and as the fluid power industry has grown, these applications have become more sophisticated and extensive. However, research scientists have never understood the classical problem of piston-bore assemblies: seizure of the assembly. Especially with the recent stringent requirements imposed upon the assembly such as high pressure and tight clearance, understanding the mechanism of particulate-induced seizure and the force required to break the seizure has become critical. Although researchers have attempted to analyze the process of particulate-induced seizure, none have offered a successful method to predict the occurence of seizure nor to assess the force required to break seizure. Researchers attempted to prevent particulate-induced seizure by applying dither to the assembly, but as of yet, this method has not proven successful.

This study therefore examines the nature of the particulate-induced seizure of a piston-bore assembly in order to predict the force required to break seizure and thus avoid catastrophic failures associated with the seizure of a piston-bore assembly. Three theoretical models were first developed. The first model is the silt process model which describes how solid particles silt in the clearance of a piston-bore assembly. The second model is the particulate-induced seizure model developed from the stress relation of three interacting bodies: the piston, the bore, and the particle. The force required to break the seizure of a piston-bore assembly was found from the three body stress relation and the assembly geometry. The third model, the dither model, predicts that when a piston-bore assembly is treated as a filter, the higher the dither frequency, the lower the assembly filtration efficiency.

To verify the integrity of these models, a computer simulation program that combines the first two models is developed. Experiments for each model are performed, and the results are compared against the computer simulation results.

CHAPTER II

PREVIOUS INVESTIGATIONS

The study of seizure on piston-bore assemblies began in the early 1950's. Sweeney [1] conducted experimental and analytical studies on the phenomenon called "hydraulic lock." He believed that when an axial pressure exists on one surface of a piston, the non-uniform pressure distribution in the clearance of the piston-bore assembly caused the piston to adhere to the bore. Sweeney observed that when the measured leakage flow corresponds to the mathematical leakage flow model with an eccentricity of one, hydraulic lock occurs. This suggests that when the piston sticks to the bore, indicating an eccentricity of one, more frictional force exists to retard the spool movement.

In contrast, Stringer [2] stated that dirt in the oil caused hydraulic lock, and this locking force is ten times as great as those reported by Sweeney.

Mannam [3] summarized the analytical investigation of hydraulic lock. He presented an approximate pressure distribution in the clearance assuming that the hydraulic fluid is incompressible and has constant viscosity. The analytical equation he used to obtain the approximate solution is

$$\frac{\partial}{\partial x}\left(h^3 \frac{\partial P}{\partial x}\right) + \frac{\partial}{\partial y}\left(h^3 \frac{\partial P}{\partial y}\right) = 0 \qquad (1)$$

where

p = the pressure at point x,y on the surface of the piston

h = radial clearance

x = axial direction

y = circumferential direction

Figure 1 presents possible applications of the equation.

When the land surfaces of the piston and bore are perfectly parallel to each other in the clearance (Fig.1a), there is no lateral force. In addition, a movement will be generated to correct any tilting of the piston axis relative to the bore axis if the tilting occurs (Fig. 1b). When the land of the piston has a uniform taper in the axial direction, there will be no resultant lateral force, provided that there is no eccentricity. However, if an eccentricity exists, then a clearance forming a taper diverging in the direction of axial flow would have a lateral resultant force proportional to the magnitude of the eccentricity (Fig. 1c). On the other hand, a tapered piston with a converging clearance in the direction of axial flow has the centralizing resultant force to reduce the eccentricity (Fig. 1d).

Mannam also indicated that the grooves in the piston land and stepped land reduce this lateral force; therefore, less hydraulic lock force occurs. Mannam did not give any details about the effect of dirt, although he was aware of its importance.

In 1966, Dransfield et. al. [4] also analytically investigated the hydraulic lock. To simulate the hydraulic lock they developed a mathematical model which includes the eccentricity and elliptic shape of

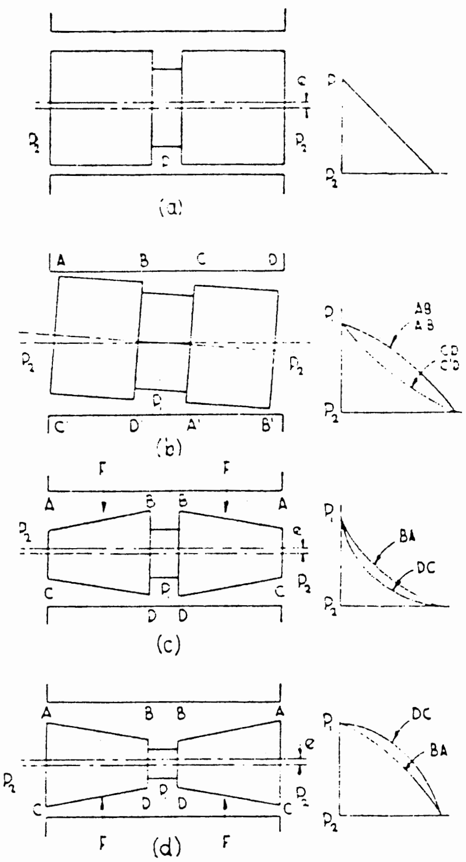


Figure. 1.   Examples of Piston Pairs [3].

a piston cross section. They found that the elliptic shape increases the locking force.

In 1967, Dransfield [5] published experimental results on the hydraulic lock using the fixture shown in Fig. 2. Dransfield measured two parameters out of this experimental fixture: 1) leakage, and 2) force required to move the yoke. Dransfield and Bruce [6] developed the leakage equation including the pressure effect on the piston-bore geometry and pressure and temperature effect on viscosity. Figure 3 shows the measured results of both leakage and force required to move the yoke. Dransfield found that the hydraulic lock force is a function of the axial pressure, and, as the applied pressure increases, the measured force increases to its maximum peak and decreases to zero. The leakage flow correlates closely with the hydraulic lock force. When the hydraulic force disappears at a certain pressure level, the leakage flow also decreases from an eccentricity one leakage down to an eccentricity near zero.

The existence of a maximum peak and gradual decrease of hydraulic lock force, first found and analyzed by Whiteman [7], is further researched by Dransfield and Bruce. They considered that very high axial pressure, creating the expansion of the clearance, also caused the decrease in the lock force. High axial pressure reduces the diameter; thus, the clearance forms a taper converging in the direction of the axial leakage flow. Consequently, the assymetric pressure distribution in the clearance acts to reduce the eccentricity.

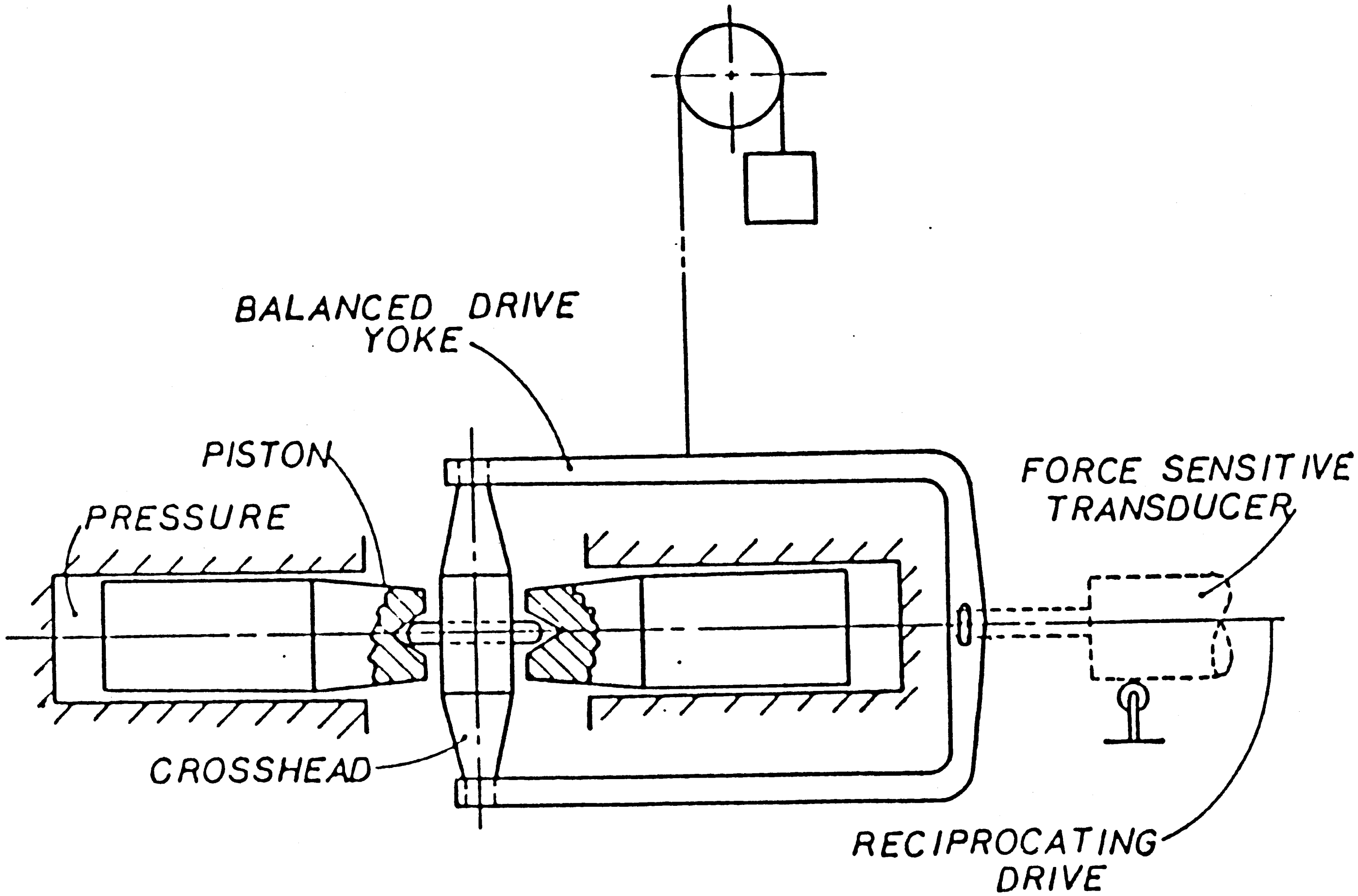


Figure. 2. Experimental Piston-Cylinder Set Up [5].

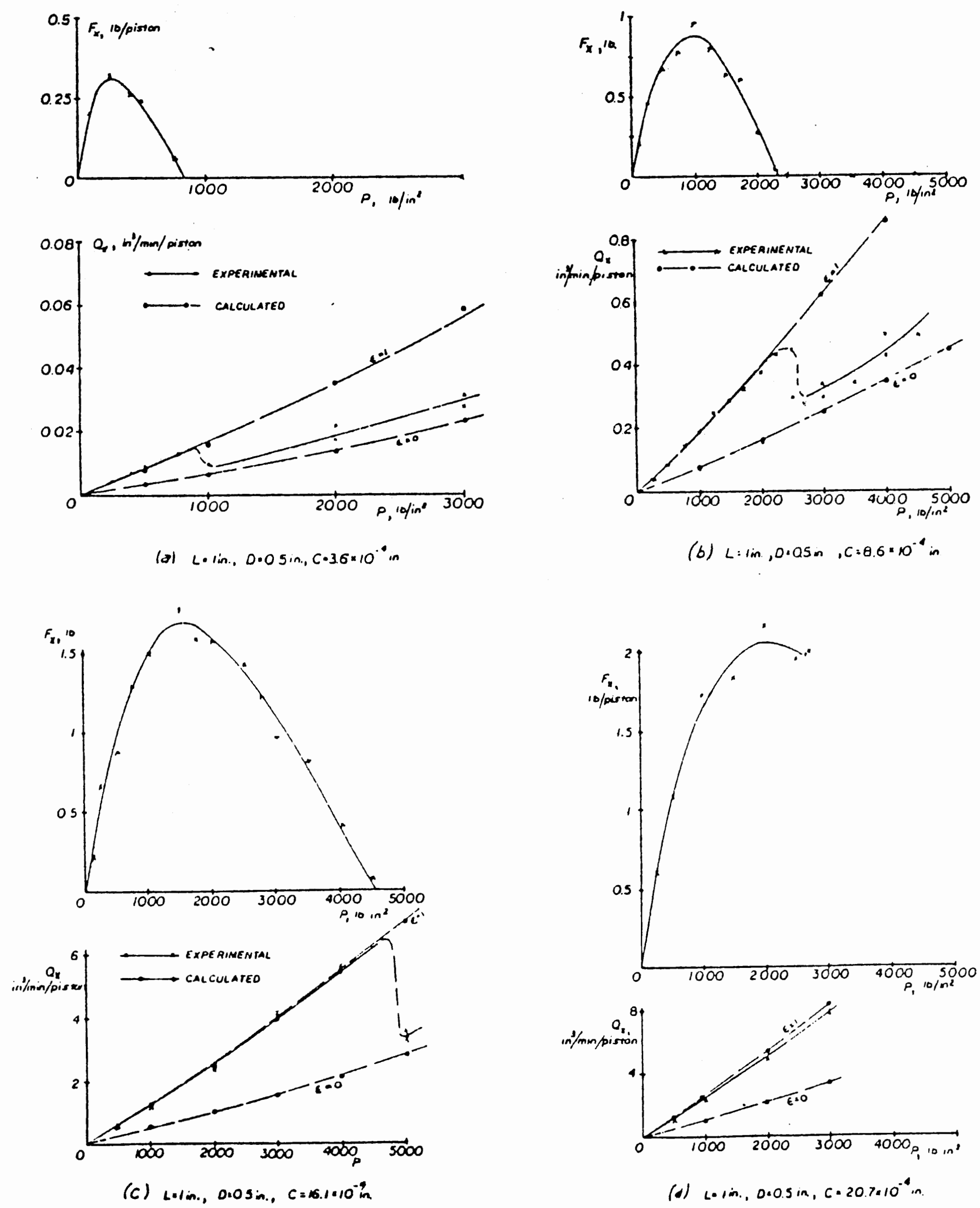


Figure. 3. Break-Out Force (Fx) and Leakage Flow Rate (Qx) Versus Pressure Characteristics, for Various Diametral Clearances (C) [6].

Dransfield and Bruce revealed other important evidence:

(1) The larger the clearance, the greater the maximum force required to overcome the seizure.

(2) The decrease in leakage flow takes place within a 100 to 500 psi range of pressure changes at which the lock force decreases close to zero.

In 1970, Kamijo and others [8] investigated the importance of the curve effect in the spool axis. They pointed out that if the effect of elastic deformation occurred only as Whiteman and Dransfield suggested, then the lock force reduction would occur at a much higher pressure (Fig. 4). When Kamijo and others included the curving effect in the spool axis and calculated the pressure distribution and its resultant force, they found a close agreement with the experimental result shown in Fig. 4 (elastic deformation and inclination of spool).

Kamijo and others, who built the experimental apparatus to measure the pressure distribution in the piston-bore assembly clearance (Fig. 5), were the first to prove experimentally that an assymetric pressure distribution exists in the clearance (Fig. 6). Figure 6 also shows the variable viscosity pressure curve, which takes into account the pressure and temperature effect on viscosity and shows a better fit to the actual pressure values than the constant viscosity pressure curve.

Using the same apparatus, Kamijo et. al. analyzed the dirt effect on pressure distribution in the clearance. Dirt filtered by a 10-micrometre mesh was injected into hydraulic fluid circulating through the test apparatus. Figure 7 demonstrates the measured result of the pressure

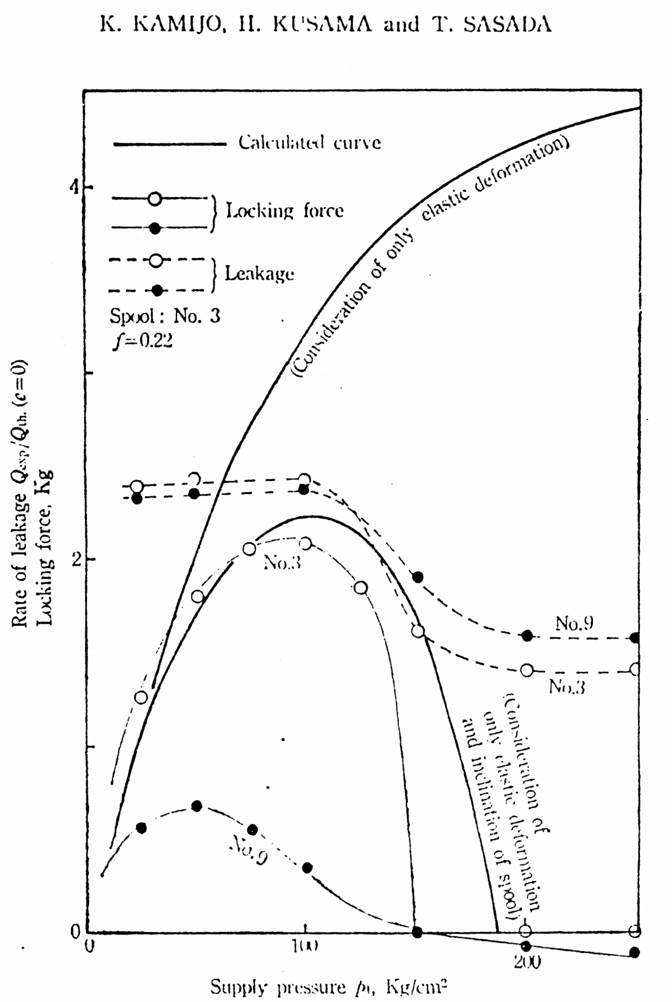
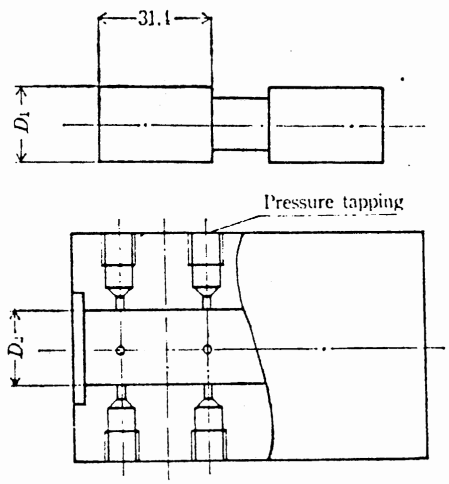
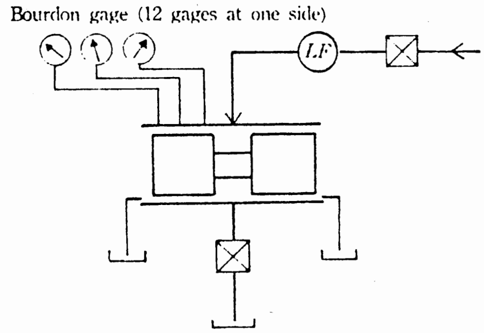


Figure. 4. Locking Force and Leakage at Higher Pressure [8].



Spool and cylinder to measure pressure distribution.



Circuit to measure pressure distribution.

Figure. 5. Pressure Distribution Measurement Apparatus [8].

Hydraulic Lock on Spool Valve (2nd Report)

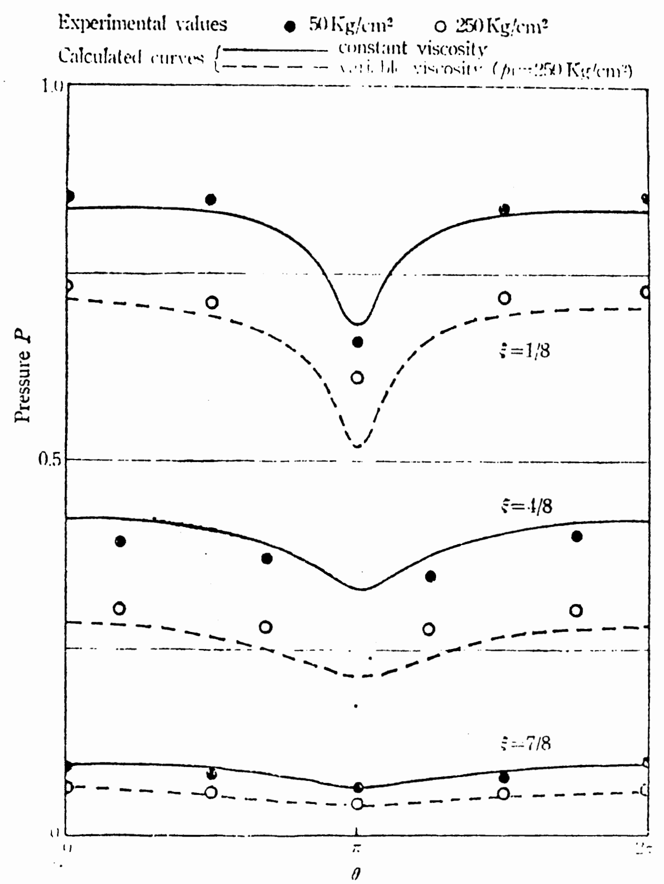


Figure. 6. Pressure Distribution Around Tapered Spool [8].

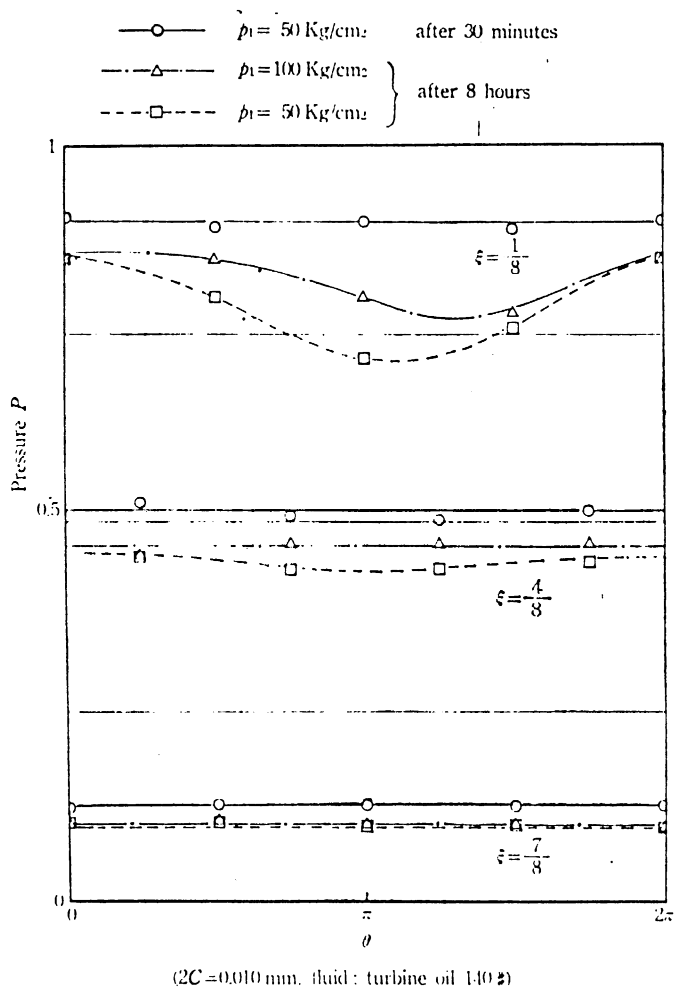


Figure. 7. Influence of Dirts on Pressure Distributions With Parallel Spool [8].

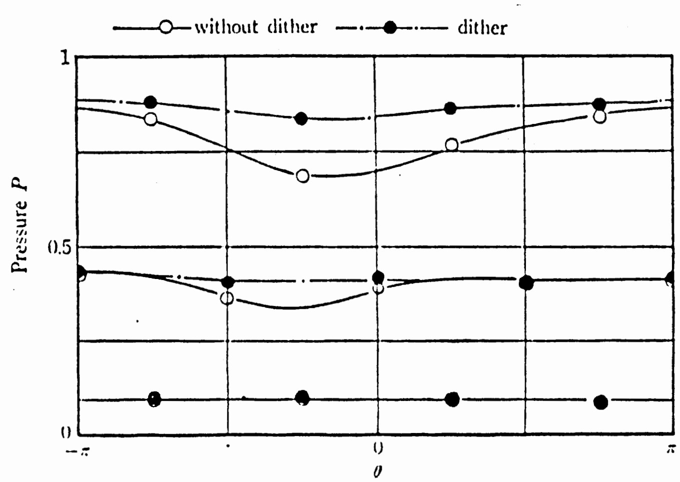
distribution. Kamijo and others verified in the same report that the hydraulic lock has a very short delay or saturation time. Therefore, the long development time observed in Fig. 7 proves that the dirt causes the non-uniform pressure distribution. In addition, Figure 7 suggests that dirt tends to accumulate in the neighborhood of the inlet. This agrees with the work of Laurenson [9]. He also found that when the clearance was increased to 22 micrometres under the same test conditions as before, no pressure fall took place.

Kamijo and others continued their experiments to apply dither on the piston of the piston-bore assembly. The dither amplitude was fixed to be 0.05 mm., and the frequency was also set at 20 Hz. Figure 8 shows the dither effect on the clearance pressure distribution. Obviously, the dither is capable of preventing or minimizing the non-uniform pressure distribution in the clearance.

In 1976, Surjatmadja and Fitch [10] published a series of papers on particulate-induced seizure and proposed four modes of particulate-induced seizure. No experimental results were presented.

Iyenger [11] also conducted experiments on particulate-induced seizure in 1976. He observed that the force required to break seizure increases with time until the force reaches a saturation level. He found that ACFTD 0-5 micrometre, lower-cut dust causes the most severe lock on the test valve with a design clearance between 3.8 to 7.6 micrometres.

Inoue [12], in 1980, developed an analytical model to predict the force required to break particulate-induced seizure, and the model incorporates the derivative of a constant filteration model to account



(C=0.006 mm, fluid : turbine oil 140)

Figure. 8. Effect of Dither on Pressure Distributions With Parallel Spool [8].

for the silt process of particles and the reduction in leakage flow through the clearance. Due to many unknown parameters included in his model, one who plans to use the model must first conduct experiments to identify unknown parameters. He also presented experimental data demonstrating the strong influence of particle size distribution on particulate-induced seizure force, and he found the induced force to be over 50 lbf, which is considerably higher than the force required to break the hydraulic lock.

Nikitin and Chekov [13] reported experimental results on the dither effect to seizure (Fig. 9). However, the report lacks detailed information of piston-bore geometries and test conditions.

Nair [14] surveyed the work performed on the study of dither. He presented the proportional control valve performance data illustrating that dither reduces the effect of particulates in the fluid.

Ito [15] hypothesized that particles close to the clearance size are the major source of particulate-induced seizure. He presented experimental data supporting his postulate.

Sasada and Michina [16] verified with their experiments that wear particles close to the size of the piston-bore clearance caused seizure.

One of the difficulties in analyzing the particulate-induced seizure is the fact that in the past no researcher had a quick and convenient method to analyze the size distribution of the particulate or dirt. As a result, researchers could only roughly regulate the size distribution by filtering particles through a mesh having an approximate size. This approach normally could only provide a coarse control over the upper

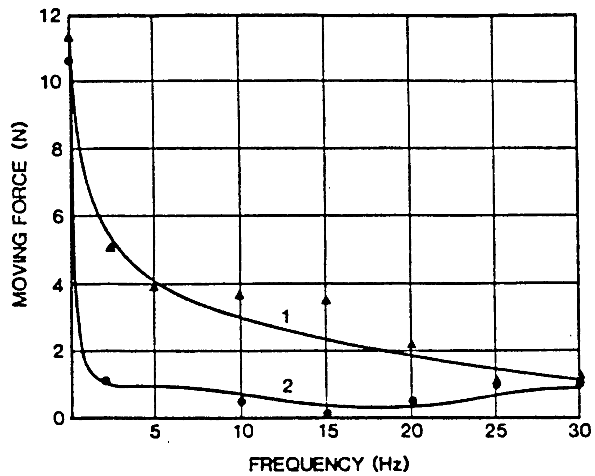


Figure. 9a. Frequency of Oscillations on the Force Necessary for Movement of Plunger of Spool Valve [13].

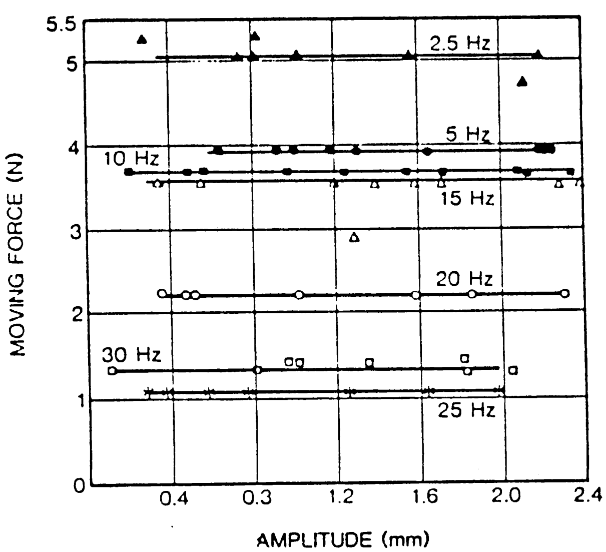


Figure. 9b. Effect of Magnitude of Amplitude and Frequency of Oscillations on the Force Necessary for Movement of Plunger of Spool Valve [13].

limits of particle size.

From the early 1970's to the mid 1970's, Erwin and Bensch [17] established the ACFTD particle size distribution. Cole, and later Bensch and Fitch [18, 19], presented an analytical model to describe particle size distribution, and this model was applied to ACFTD. Though useful particle counting techniques and particle size distribution models are now available, no researcher to date has used this technology to investigate the particulate-induced seizure.

Since particulate-induced seizure occurs only when particles are deposited in the clearance, this silting process can be investigated by treating the piston-bore assembly as a filter.

Fluid Power Research Center (FPRC) has investigated the particle separation mechanisms in a fibrous filter for many years and has identified the following types:

1. Seiving
2. Direct Interception
3. Inertial Impaction
4. Brownian Diffusion
5. Gravity Settling
6. Electrostatic Precipitation
7. Obliteration

The second through the fifth mechanisms are minor mechanisms compared to the others because although they may capture particles, they have very poor retention (dirt holding) capability. In addition, the seventh mechanism becomes significant only at the freezing temperature (0 degrees

Celsius) [26]. Therefore, the sieving and electrostatic precipitation mechanisms are the two known to most effectively capture and retain particles, thus being the main mechanisms of the silt process in piston-bore assemblies. Of these two mechanisms, sieving contributes to the seizure the most because particles large enough to bridge the clearance are suspected of being the major cause of the seizure.

To summarize, there is no analytical model which could estimate the required force to break particulate-induced seizure for a given set of conditions of contaminant size, fluid flow rate, piston-bore design, and pressure drop. Furthermore, the theoretical and experimental studies of the effect of dither on particulate-induced seizure are seriously lacking.

CHAPTER III

DEVELOPMENT OF THEORETICAL MODELS

Silt Process (Silt Beta) Model

To analyze how particles silt in the clearance of a piston-bore assembly, the system must be first described and then analyzed according to known models developed in the current work. The description of the assembly system starts with a list of vital inputs and outputs listed in Fig. 10, and then the relationships of these parameters are investigated.

The relationship between the axial pressure difference and the leakage flow through the clearance is described by:

$$Q(t) = \frac{\pi D h^3 (1 + 1.5 (e/h)^2)}{12 \mu L} (Pu - Pd) \qquad (2)$$

where

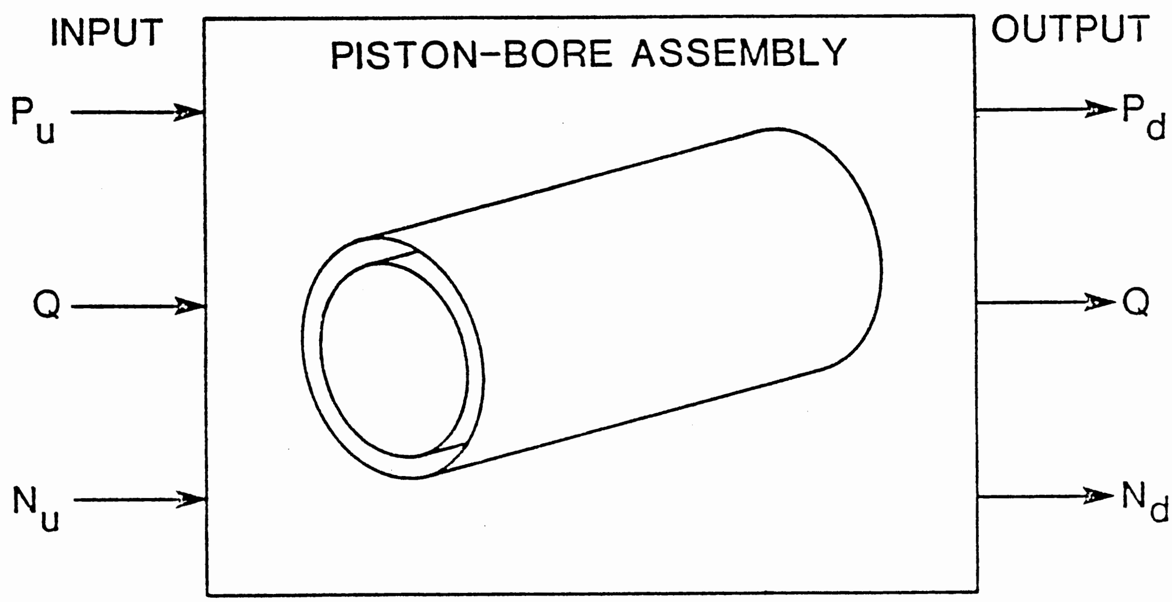
e = eccentricity

h = nominal clearance

Pu = upstream axial pressure

Pd = downstream axial pressure

D = bore diameter



SUBSCRIPT

u - UPSTREAM
d - DOWNSTREAM

PARAMETER

P - PRESSURE
Q - LEAKAGE FLOW
N - PARTICLE SIZE DISTRIBUTION

Figure. 10. Input and Output of Piston-Bore Assemblies.

μ = viscosity (including pressure and temperature effect)

L = piston land length

However, as soon as particulate contaminants enter the clearance of the piston-bore assembly, the leakage flow decreases. Particles silt up in the clearance, causing the effective clearance size h to decrease with time. In order to model the decrease of the leakage flow, the particle size distribution upstream (Nu) and downstream (Nd) from the assembly have to be defined. The difference between the two particle size distributions indicates the process of particle silting per unit volume of leakage through the clearance of the piston-bore assembly. The upsteam and downstream particle size distribution are mathematically modeled [17-19], and the difference between the two distibutions are represented by the Beta rating system. The ratio of the number of particles for a given size upstream vs. downstream is the Beta value used in the Beta rating system. The larger the value of Beta, the more the particles at the evaluated particle size are retained by the assembly. Therefore, the Beta value evaluated at the nominal clearance size will be referred to as the silt Beta β. The silt Beta value and the number of particles greater than 1 micrometer downstream (NTd) define a downstream particle size distribution.

Therefore, the volume of particles silted in the clearance can be found by a three step procedure:

Step 1. Find particle size distribution upstream (NDu) and downstream (NDd):

$$NDu = NTu\ e^{(-Bu\ \ln^2(D))} \tag{3}$$

$$Bd = Bu + \ln(NTd\ /\ NTu\ \beta\ )\ /\ \ln^2(h) \tag{4}$$

$$NDd = NTd\ e^{(-Bd\ \ln^2(D))} \tag{5}$$

where

h = nominal clearance size.

Step 2. Find the total volume occupied by each particle size distribution per unit volume of fluid (see references 17 to 19 for detailed derivations):

$$Vu = \frac{\pi\ NTu}{6\ S}\ \{1 + 3\sqrt{\frac{\pi}{Bu}}\ e^{(9/(4Bu))} \int_{-\infty}^{\frac{3}{\sqrt{2B_u}}} \frac{1}{\sqrt{2\pi}}\ e^{(-x^2/2)}\ dx\} \tag{6}$$

$$Vd = \frac{\pi\ NTd}{6\ S}\ \{1 + 3\sqrt{\frac{\pi}{Bd}}\ e^{(9/(4Bd))} \int_{-\infty}^{\frac{3}{\sqrt{2B_d}}} \frac{1}{\sqrt{2\pi}}\ e^{(-x^2/2)}\ dx\} \tag{7}$$

where

$x/\sqrt{2} = y - 3/(2\sqrt{B})$

$y = Bi\ \ln D$

'i' = subscript indicating either 'u' or 'd'.

'u' = upstream

'd' = downstream

Step 3. Compute the total volume of particles retained in the clearance by the difference between upstream and downstream volume.

$$Vt = Vu - Vd \tag{8}$$

In addition, find the frequency (number of particles for a given size) versus size by application of the equation:

$$f(D)u = 2\ NTu\ Bu\ \ln(D)\ /\ D\ e^{(-Bu\ (\ln^2(D)))} \tag{9}$$

The above equation is derived by taking a derivative of Eq. 3 with respect to particle size D. Thus, the frequency versus the size of particles retained in the clearance after a unit volume of leakage flow passes through the clearance is:

$$f(D)t = f(D)u - f(D)d \tag{10}$$

Particles silted and retained in the clearance of the piston-bore assembly affect the leakage flow through the clearance described by Eq. 1. In order to find the reduction of the clearance size, the total volume of particles retained in the clearance within a unit time of delta t is:

$$\Delta S = \int_{t}^{t+\Delta t} Vt(t)\ Q(t)\ dt \tag{11}$$

Therefore, the reduction of clearance in unit time delta t is:

$$\epsilon = \Delta S \ / \ (2 \pi L (D - h)) \tag{12}$$

Then the clearance reduces to:

$$h = h - 2 \epsilon \tag{13}$$

The flow chart shown in Fig. 11 summarizes the entire silt process. Notice three different particle retention cases exist. The above steps describe the particle retension (silt Beta) case 1; however, the same

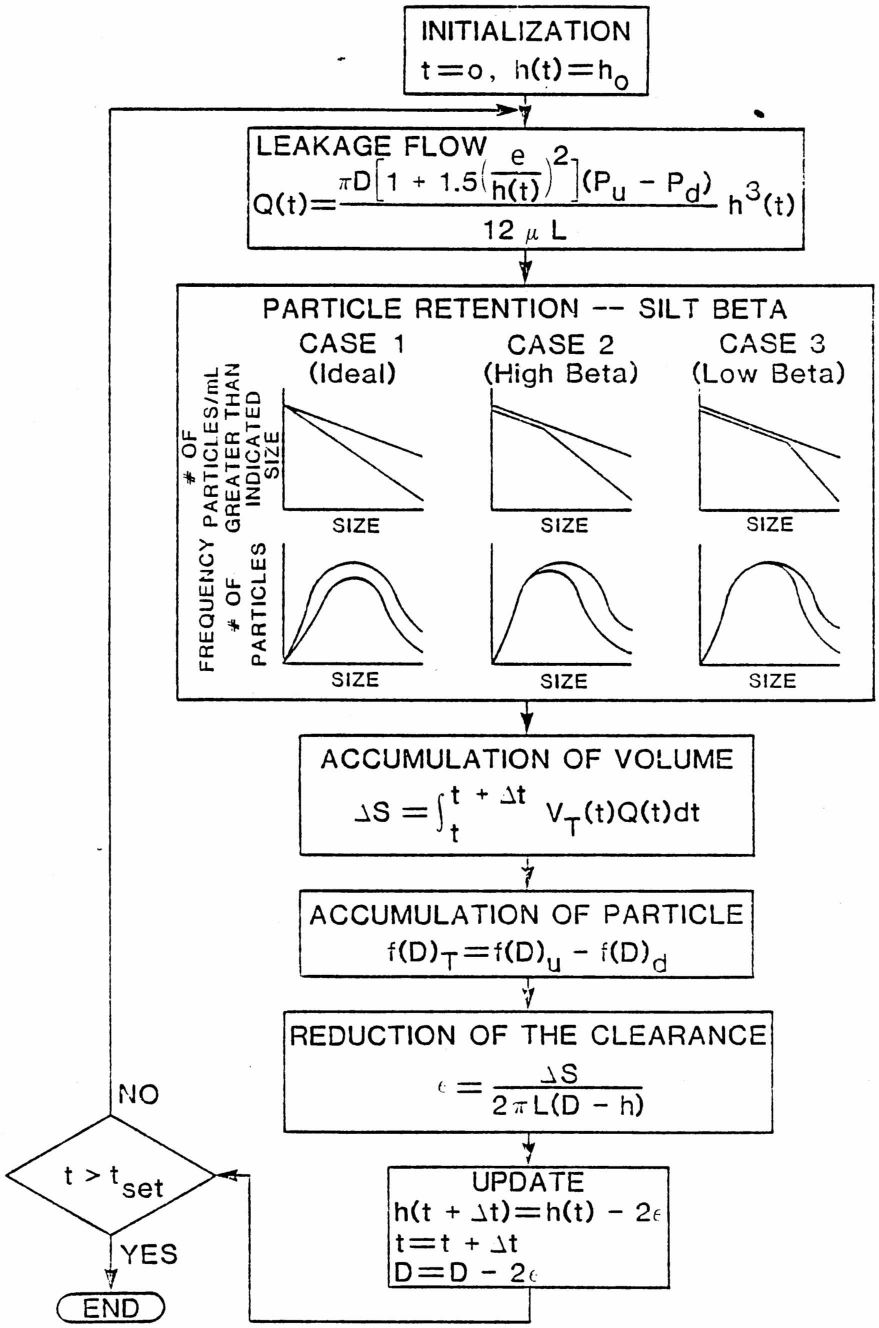


Figure 11. Silt Process Algorithm

approach applies to the linear approximation of cases 2 and 3. By fitting straight lines to the downstream particle size distribution and applying the above steps for each straight line segment, the silt volume Vt can be found. Then, the application of Eq. 11 through 13 enables an estimate to be made of the clearance and leakage reduction.

Particulate-Induced Seizure Model

Ito et. al. [21] obtained successful three body wear model results using the following hypothesis: a particle in the clearance of a piston-bore assembly continues to indent into both constraining surfaces of the piston-bore assembly as long as a sufficient force exists to maintain the surface pressures P1 & p1 equal to or above the lower yield strength of the two contacting surface materials (see Fig. 12). This means that when a particle has higher yield strength than the material of a piston-bore assembly, the particle indents into both surfaces of the piston-bore assembly, and when a particle has lower yield strength, the particle deforms.

Thus, the three body wear model was modified to predict the necessary force to break seizure. The criterion is that the contact pressure be equal to the yield strength (Fig. 12). Consequently,

$$P1 = Syc = FL \cos\theta \cos\gamma \,/\, (dc\ 1) \tag{14}$$

Solving for FL gives:

$$FL = Syc \times dc\ 1 \,/\, (\cos\theta \cos\gamma) \tag{15}$$

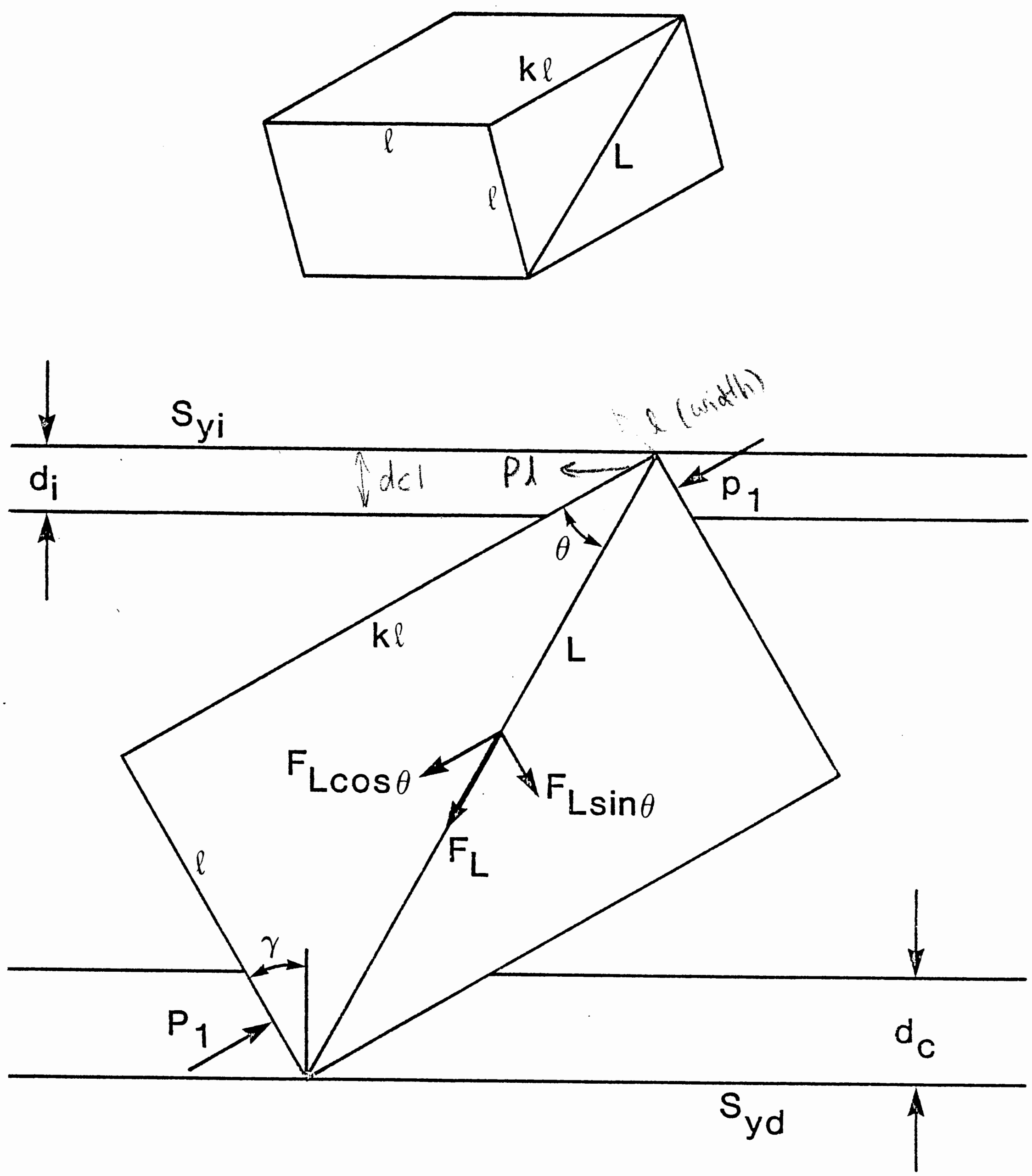


Figure. 12. Three Body Model.

Similarly, the force FL on the other surface is:

$$FL = Syi\ di\ l\ /\ (\cos\theta\ \cos\gamma) \tag{16}$$

Equations 15 and 16 are combined to eliminate FL, and the result is:

$$Syc\ /\ Syi = di\ /\ dc \tag{17}$$

The above equation reveals an important relationship between the indentation depths and the yield strengths of a piston-bore assembly. Futhermore, the component of force FL that resists the movement of the spool is:

$$FP = FL\ \cos(\theta + \gamma) \tag{18}$$

The geometrical relationship between the depth of indentations, particle size, and clearance size is given by:

$$di + dc = L\ \sin(\theta + \gamma) - H \tag{19}$$

From Eq. 17 and 19, the following equation can be derived:

$$dc = (L\ \sin(\theta + \gamma) - H)\ /\ (1 + Syc\ /\ Syi) \tag{20}$$

Equation 20 replaces dc in Eq. 15, resulting in:

$$FL = \frac{Syc\ l\ (L\ \sin(\theta + \gamma) - H)}{\cos\theta\ \cos\gamma\ (1 + Syc\ /\ Syi)} \tag{21}$$

From the particle geometry,

$$l = L \sin\theta \tag{22}$$

and Eq. 18, 21, and 22 may be combined to solve for FP:

$$FP = alpha \frac{\{\sin(\theta + \gamma) - H / L\} \sin\theta}{\cos\theta \cos\gamma} \cos(\theta + \gamma) \tag{23}$$

where

$$alpha = L^2 / (1 / Syc + 1 / Syi) \tag{24}$$

The above equation defines the force needed to achieve the rake angle.

The parameters of the above equation have the following limits. The limits of the rake angle are:

$$0 <= \gamma <= \left(\pi / 2\right) - \theta \tag{25}$$

The ratio limits of the clearance size H to the size of a particle L is:

$$\sin\theta <= H / L <= 1 \tag{26}$$

Moreover, the length-to-width ratio [21] of a particle in nature is limited to:

$$1 <= k <= 1.5 \tag{27}$$

In order to closely simulate the conditions at which particulate-induced seizure occurs, the particulate-induced force equation must be applied in light of the piston-bore clearance information and the probablistic treatment on the particles' location inside the clearance. To accomplish such a task, the following two subsections develop two mathematical models which are essential for the prediction of the particulate-induced seizure of a given piston-bore assembly. These models are to find:

(1) the force variation for a given piston-bore clearance condition and particle size.

(2) the probability that particles contribute to the particulate-induced seizure.

Effect of Clearance Size and Particle Size on Particulate-Induced Seizure Force

If the length-to-width ratio of an ACFTD particle is 1.49 (k = 1.49) [20], then the ratio limits of the clearance size to a particle size is:

$$0.5573\ (\theta = \mathrm{atn}(1/1.49)) <= H/L <= 1 \qquad (28)$$

And, the limits of the rake angle become:

$$0 <= \gamma <= 56.13 \qquad (29)$$

Assuming that the value of alpha in Eq. 24 is unity, the effect of the rake angle and the ratio of the clearance H to the particle size L was investigated by developing the simulation program of Eq. 23 and 24. The result of the program is graphically presented in Fig. 13. This figure clearly illustrates how the particulate-induced force varies

H/L	.557	.704	.852	1
FPmax	.1164	.0675	.0263	0
FPavg	.0781	.0363	.0098	0

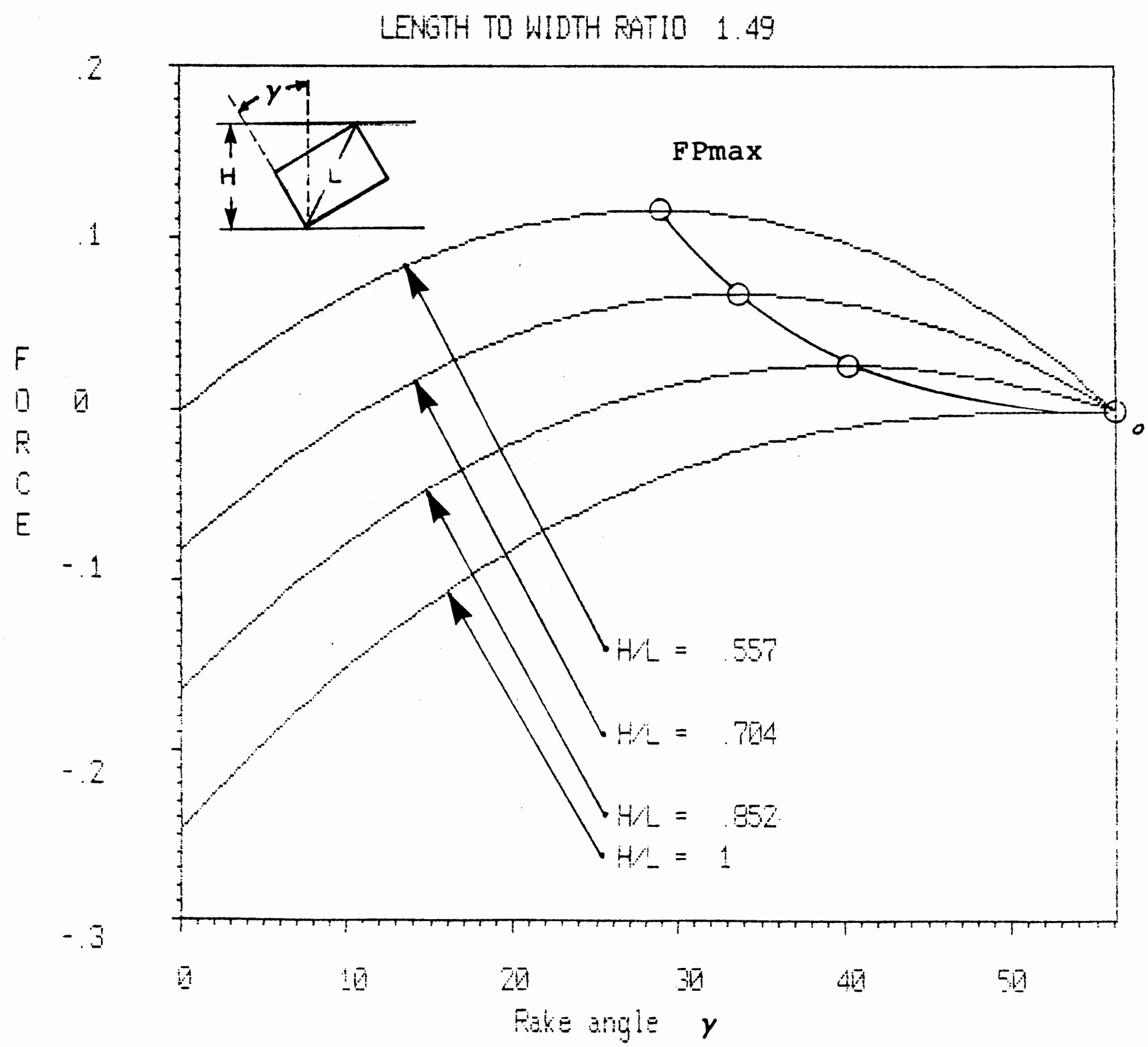


Figure. 13. Seizure Force.

according to a given rake angle: the size ratio of the clearance H to the particle size L. Note that the negative force is meaningless here because in order for a size L particle to touch the surfaces of the clearance and generate the induced force, the particle must rotate by increasing the rake angle and indenting the surfaces. Hence, no matter how much the rake angle increases, no force will be generated when the ratio H/L is one since the particle never indents the surfaces.

On the other hand, when the ratio H/L is less than unity, the particle eventually makes the indentation as the rake angle increases; thus, positive induced force occurs.

If the particle with its length-to-width ratio (k) is 1.49, the minimum of the ratio H/L is approximately 0.557. This occurs because when the ratio H/L becomes smaller than 0.557, the width of the particle becomes bigger than the size of the clearance, and the particle can no longer enter inside the clearance space.

Figure 14 plots the maximum and average force against the ratio H/L. FPmax and FPavg are smooth functions of H/L.

$$FPmax = f(H/L) \tag{30}$$

$$FPavg = g(H/L) \tag{31}$$

The straight line that connects the four data points shown in Fig. 14 closely approximates the true values between data points. Therefore, the computer simulation results, shown later, use the data derived from the straight line.

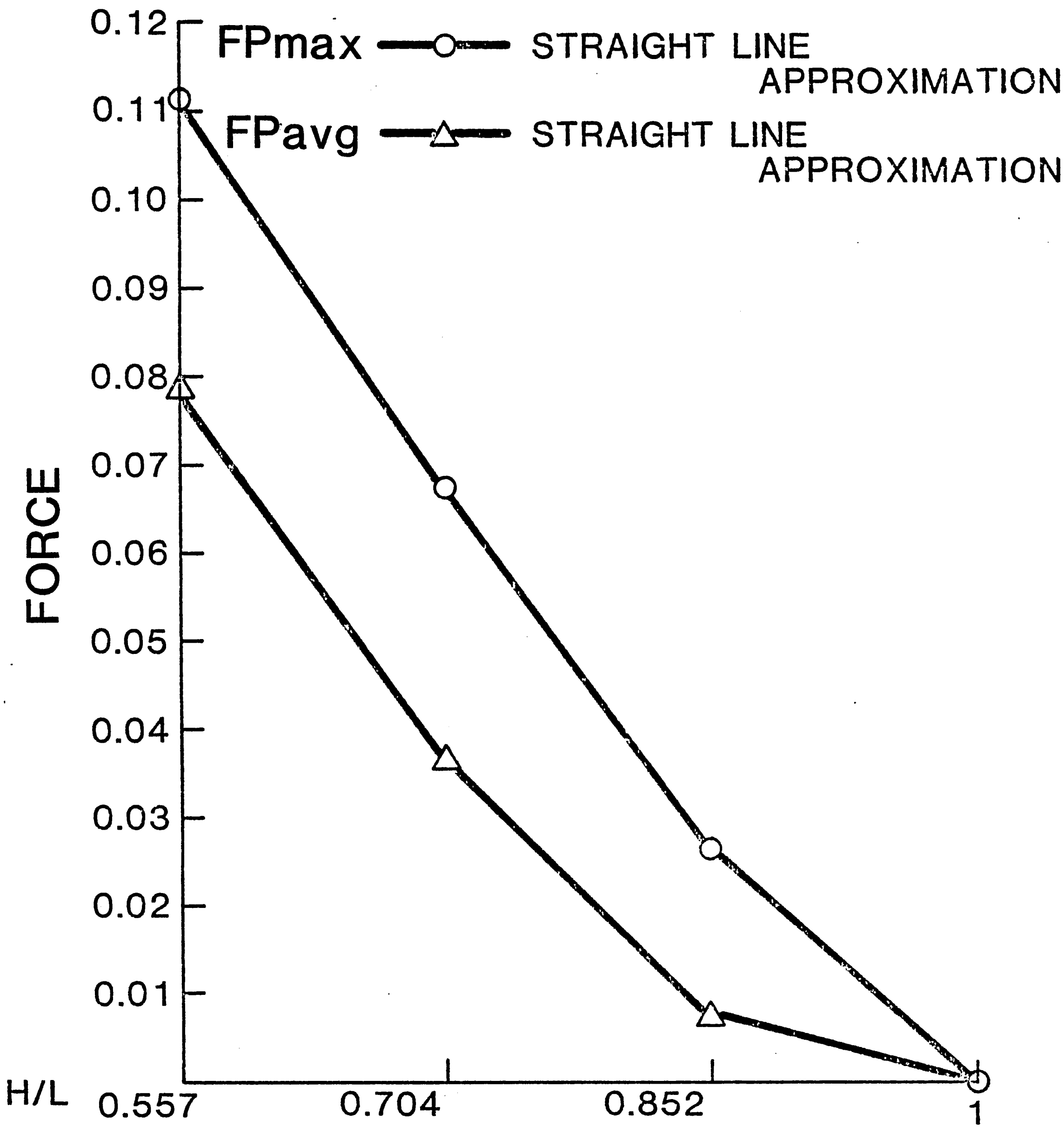


Figure. 14. Seizure Force Versus H/L.

Contribution of Silt Particles to Particulate-Induced Seizure Force

Assuming that the piston-bore assembly exists with an eccentricity of one (a good assumption according to Dransfield's experiment), the cross sectional area that the size L particle can enter is presented in Fig. 15 and marked with the angle w1. In addition, the cross sectional area in which size L particles can contribute to the particulate-induced force is designated by the angle difference between w1 and w2, and the corresponding magnitude for FPmax and FPavg is also shown in Fig. 15. Thus, if the clearance is specified as a function of w,

$$h = h(w) = \sqrt{R^2 + e^2 - 2\ R\ e\ \cos(w)} - r \qquad (32)$$

where

R = radius of the bore

e = eccentricity distance

r = radius of the piston

then, the cross sectional area pointed by w can be described by:

$$\text{Area}(w,0) = \int_0^w h(w)\ dw \qquad (33)$$

And, the area between w1 and w2 is:

$$\text{Area}(w_2,w_1) = \text{Area}(w_2,0) - \text{Area}(w_1,0) \qquad (34)$$

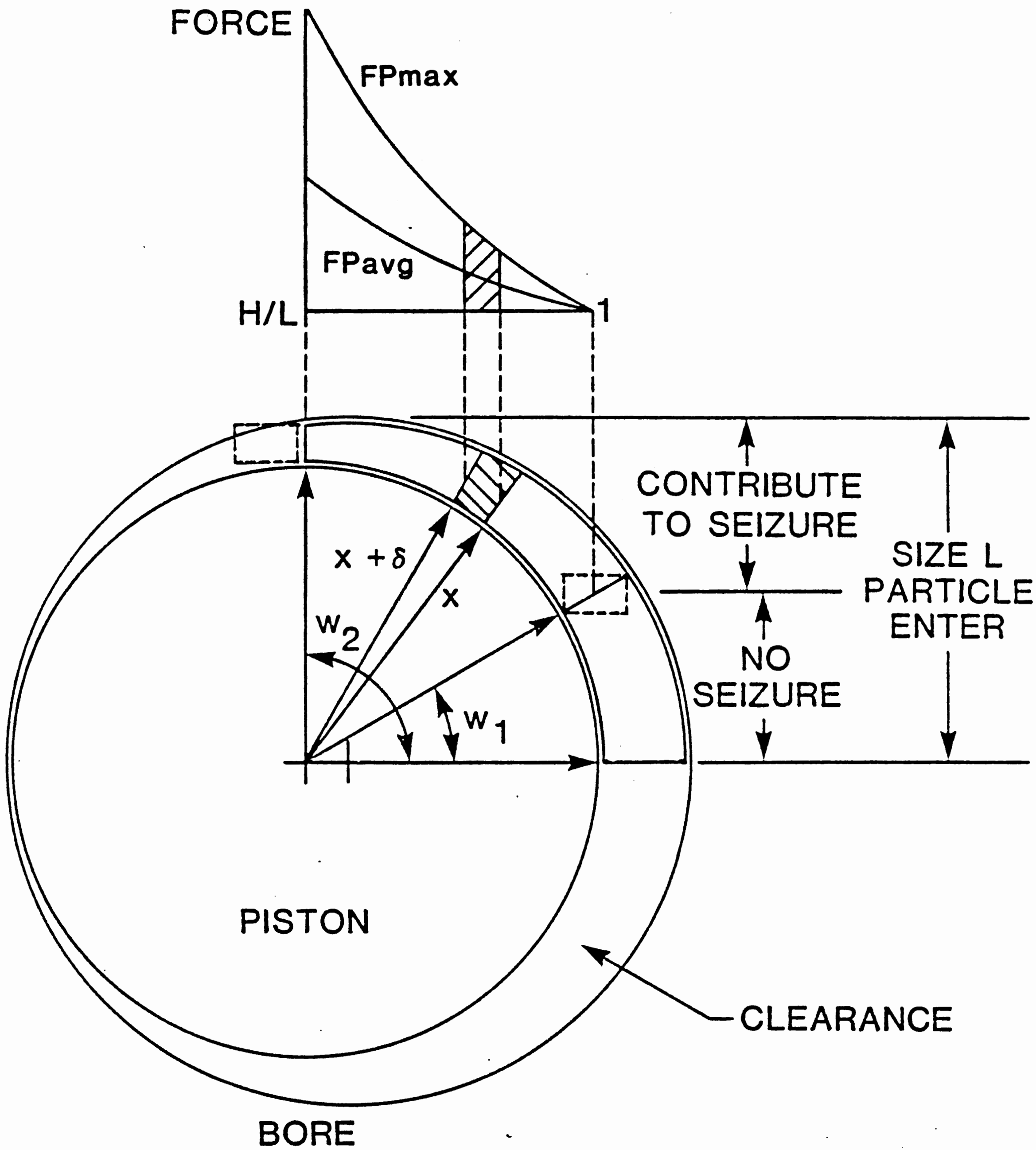


Figure. 15. Contribution of Silt Particles to Particulate-Induced Seizure.

If the total number of size L particles silted in the clearance is MD(L) (see Fig. 10), then particles among MD(L) could resist the spool movement and contribute to the seizure, calculated by:

$$MS(L) = MD(L) \frac{Area(w_2,w_1)}{Area(w_2,0)} \tag{35}$$

Furthermore, the maximum induced force varies depending upon the position of w_x as Fig. 15 shows. The total maximum particulate-induced force due to the silted size L particles is:

$$FI(L) = MS(L) \sum_{x=1}^{2} \{ \frac{Area(w_{x+\delta},w_x)}{Area(w_2,w_1)} f(h(w_{x+.5\delta})/L)\} \tag{36}$$

The maximum total induced force covering the entire size range of silted particles is:

$$FMAX = \sum_{L=1}^{LM} FI(L) \tag{37}$$

where

$LM = L \sin(atn(1/k))$

When the function g() replaces the function f() in Eq. 36 (see Eq. 30 & 31), the result is the average total induced force:

$$FAVG = \sum_{L=1}^{LM} FI(L) \tag{38}$$

Particulate-Induced Seizure Model's Validity Investigation by Computer Simulation

The simulation program (Appendix A) was developed in two parts. The first part simulates the process of particles silting in the clearance of the piston-bore assembly, and the second part simulates the particulate-induced seizure force. Table I shows a typical input for both parts.

The silt process program simulation contains the algorithm presented in Fig. 11 and gives four graphic outputs: clearance vs. time, leakage flow vs. time, silt volume vs. time, and number of particles (frequency) vs. size. Figures 16 through 19 show the example outputs based upon the inputs illustrated in Table I. The clearance and leakage flow reduction agree with the previous investigators' observations and experimental data [1, 2, 5, & 9]. Figure 18 demonstrates that the accumulation of silt per unit leakage flow volume increases as time increases. This indicates that the piston-bore assembly gradually becomes efficient in capturing particles flowing through the clearance of the assembly. Figure 19 is the result of Eq. 10. Notice that although the internal nominal clearance is 25 micrometers, large numbers of particles less than 25 micrometers in size have silted in the clearance. The above result can be attributed to the silt Beta and NTd assumption, and these assumptions have been verified by experiments to be described shortly.

The particulate-induced seizure simulation program has two output values: FMAX and FAVG. One of the four selected parameters (time, gravimetric level, axial pressure, and silt Beta) was varied while the

TABLE I

INPUT PARAMETERS

1 ---	H0 : initial clearance micrometer	25
2 ---	D : inner dia. of the bore inches	.2501969
3 ---	PU : Upstream pressure lbf/in^2	190
4 ---	PD : Down stream pressre lbf/in^2	10
5 ---	L : Silt land length in inches	.25
6 ---	SYP: Yield strength of particle lbf/in^2	58000
7 ---	SYB: Yield Strength of Bore lbf/in^2	10000
8 ---	SYS: Yield Strength of Spool lbf/in^2	30000
9 ---	S : shape factor	1
10---	G : Concentration mg/L	50
11---	BETA : Retention characteristic (Nu/Nd)	2
12---	ECC: (0<= ECC <=1) Eccentricity	1
13---	DT : Time step in sec.	6
14---	TSET : Total length of time. (sec)	60
15---	GLN : Total number of grooves	0
16---	GL : The average length of grooves (in.)	.01

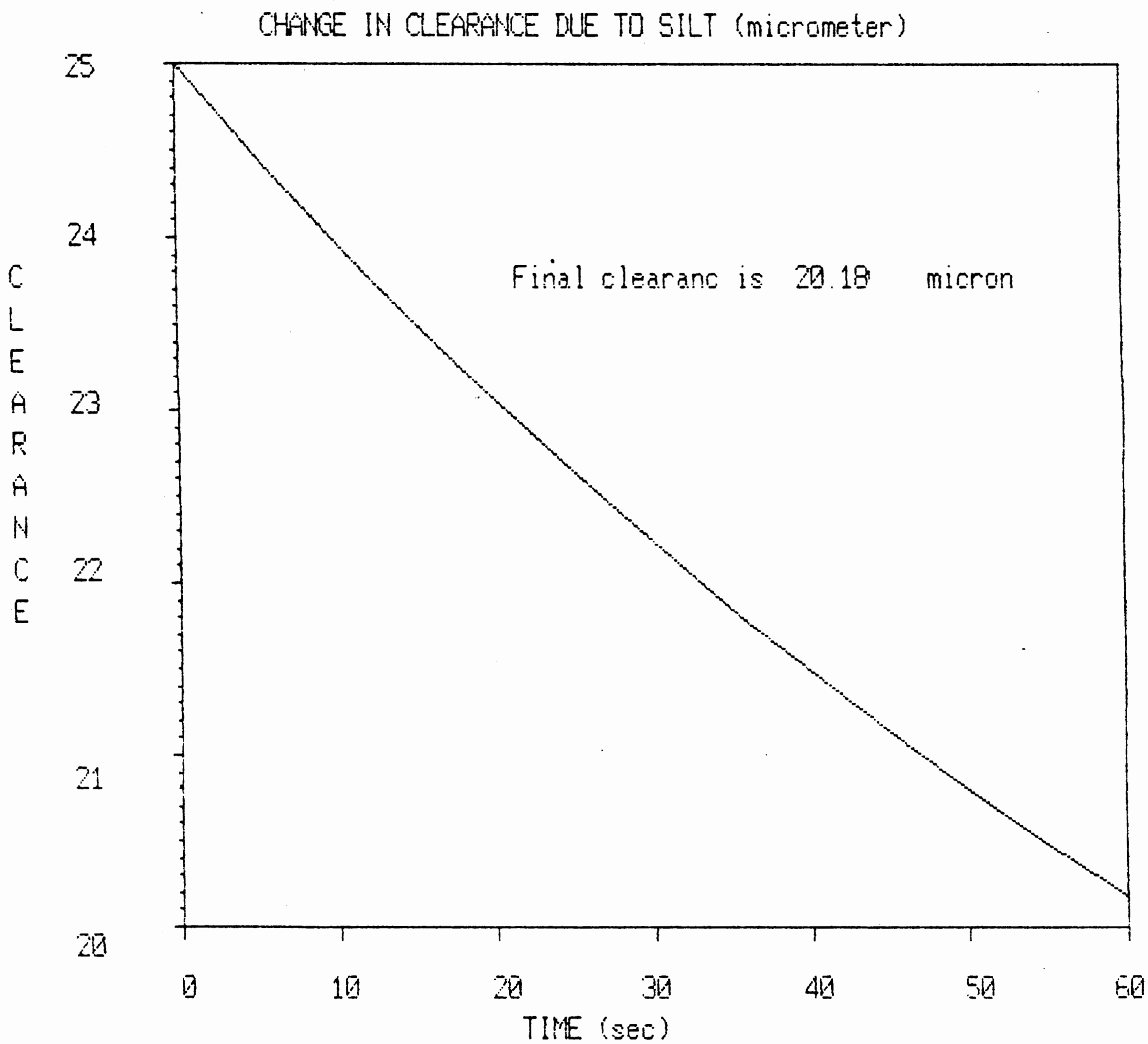


Figure. 16. Change in Clearance Due to Silt.

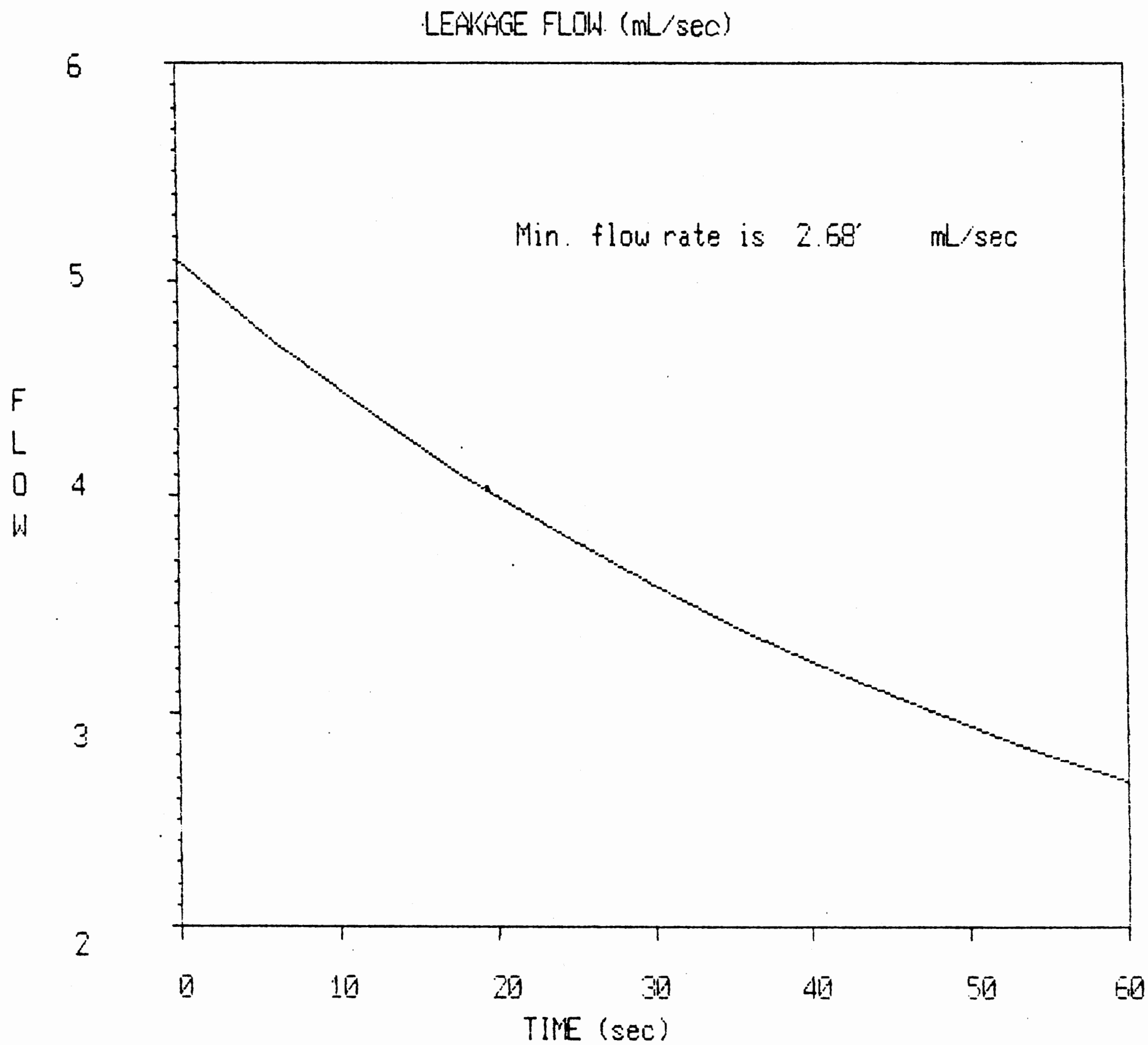


Figure. 17. Leakage Flow.

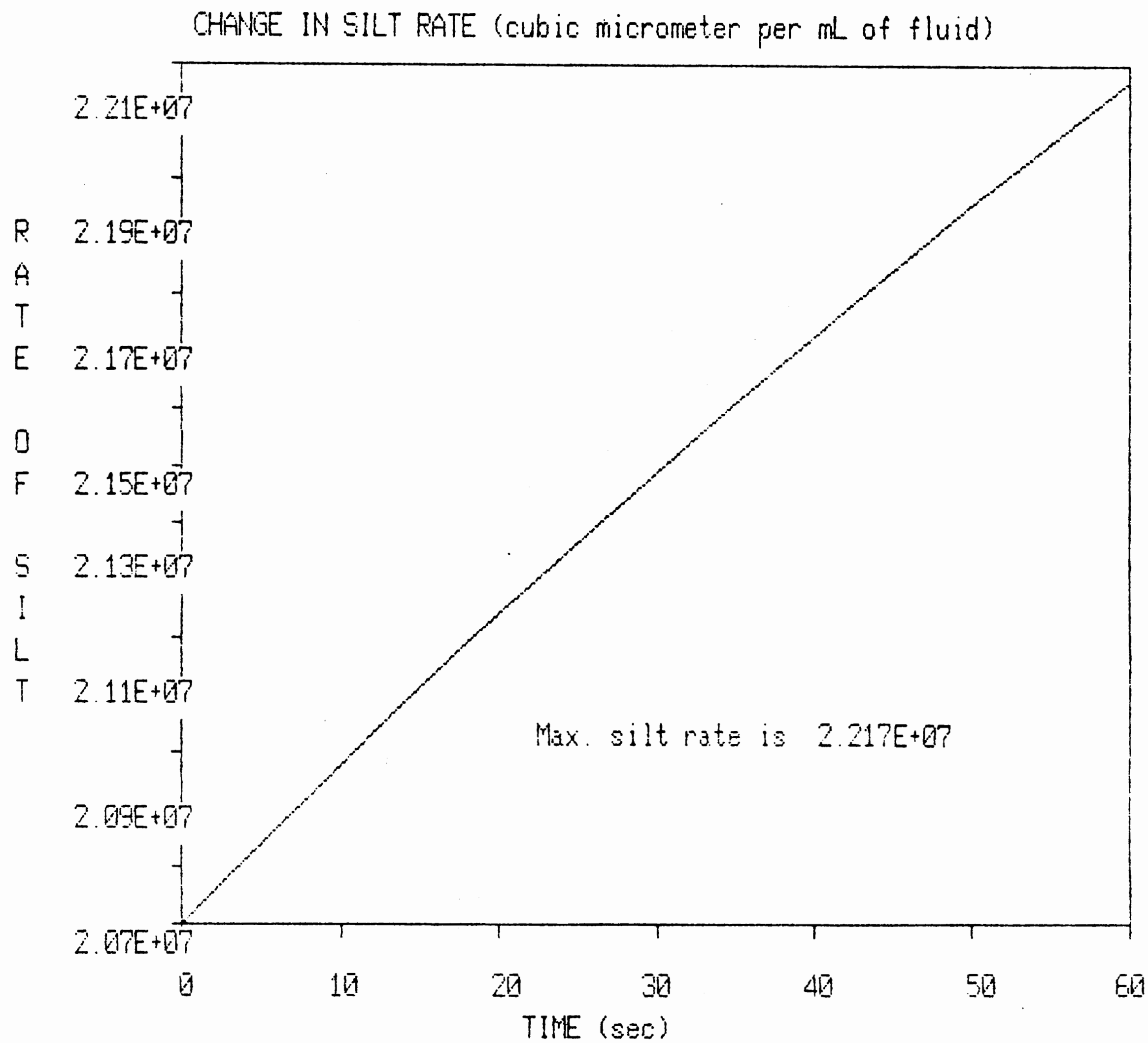


Figure. 18. Change in Silt Rate.

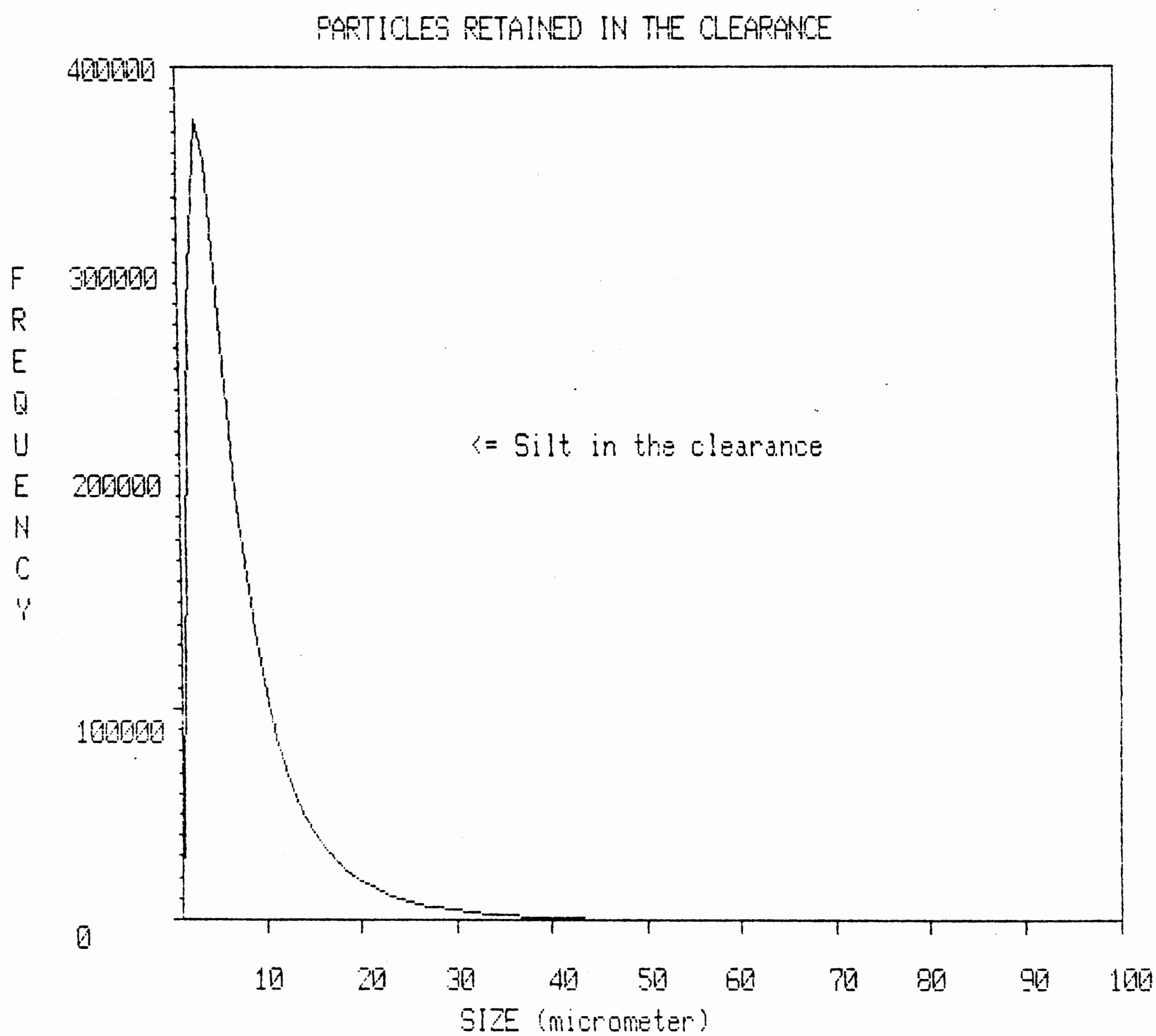


Figure. 19. Particle Retained in the Clearance.

others were held constant to investigate the parametric effect on the seizure force. Figures 20 through 23 exhibit the result. Figures 20 through 22 agree with previous experimental results [1, 3, 4, 7, 8, 11, 12, and 13]. The seizure force FAVG increases as a parameter increases, yet the increasing trend is concave downward indicating the possible terminal saturation level, which has also been observed by previous investigators. Figure 23 of the seizure force vs. silt Beta clearly displays the existence of the saturation level and demonstrates a steep increase in the seizure force at a silt Beta value between 1 and 10. These results show that the simulation programs adequately describe observed key characteristics of the dynamics of seizure observed in the previous experiments by other workers.

Dither Model

Among the paramters shown in Table I, silt Beta βs is the only parameter that dither can affect. As Figure 23 shows, when the silt Beta value decreases, FAVG decreases accordingly (so does FMAX); therefore, when the dither is applied to the piston-bore assembly, silt Beta must decrease. Kamijo's experimental results (Fig. 8) support this postulate, as his results clearly show the decrease in dirt damming and piling in the piston-bore clearance when dither is applied to the piston. Hence, the results of dither should appear as a change in the downstream particle size distribution NDd and may be measured by monitoring NDd. In order to estimate NDd, according to Eq. 4 & 5, both silt Beta value and

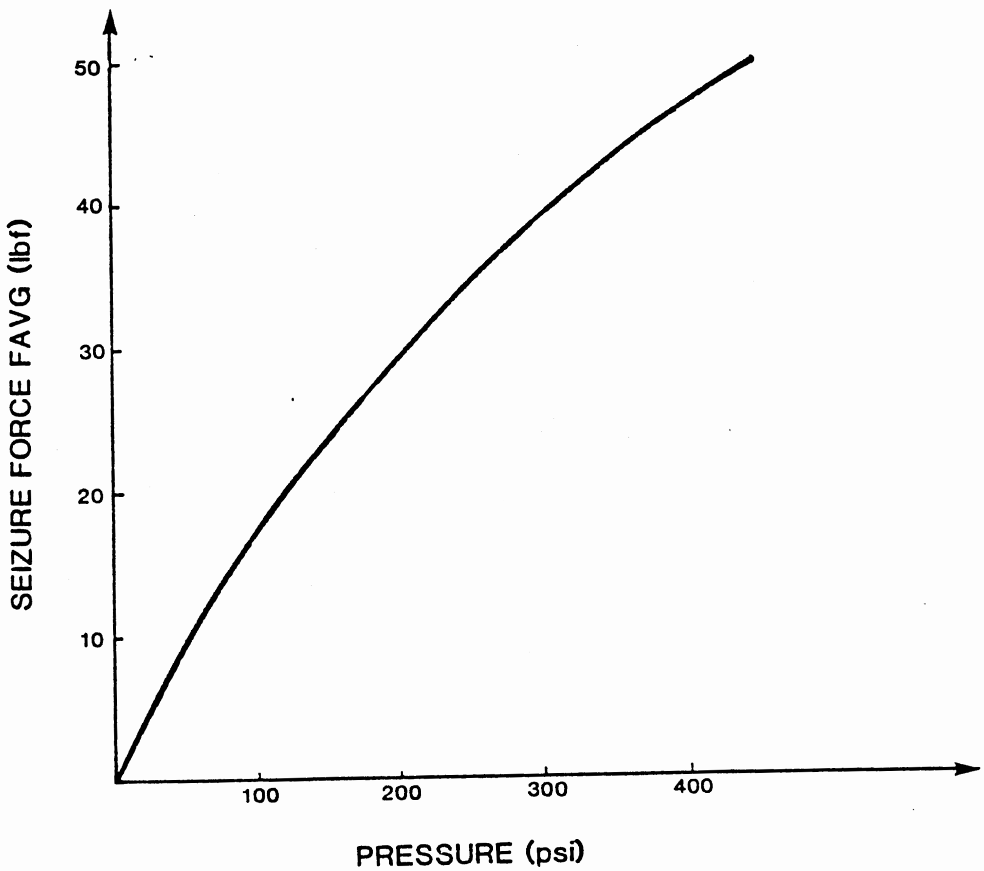


Figure. 20. Seizure Force Versus Pressure.

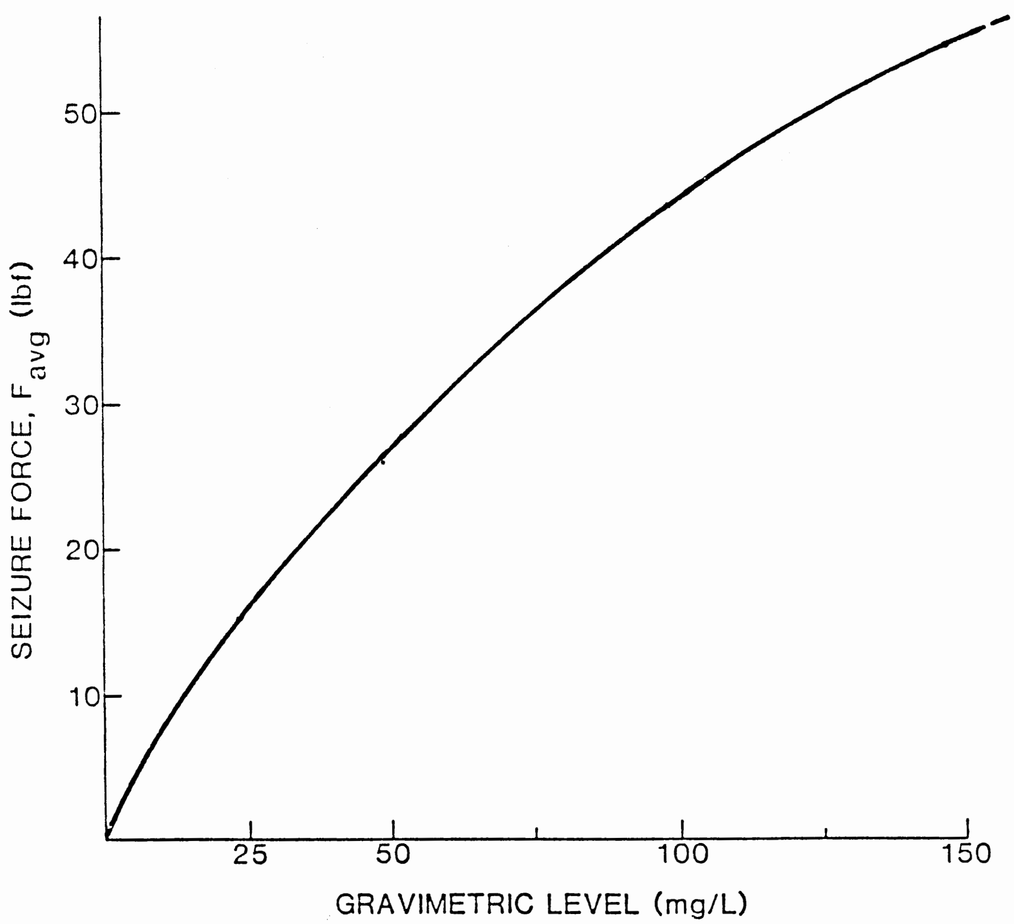


Figure. 21. Seizure Force Versus Gravimetric Level.

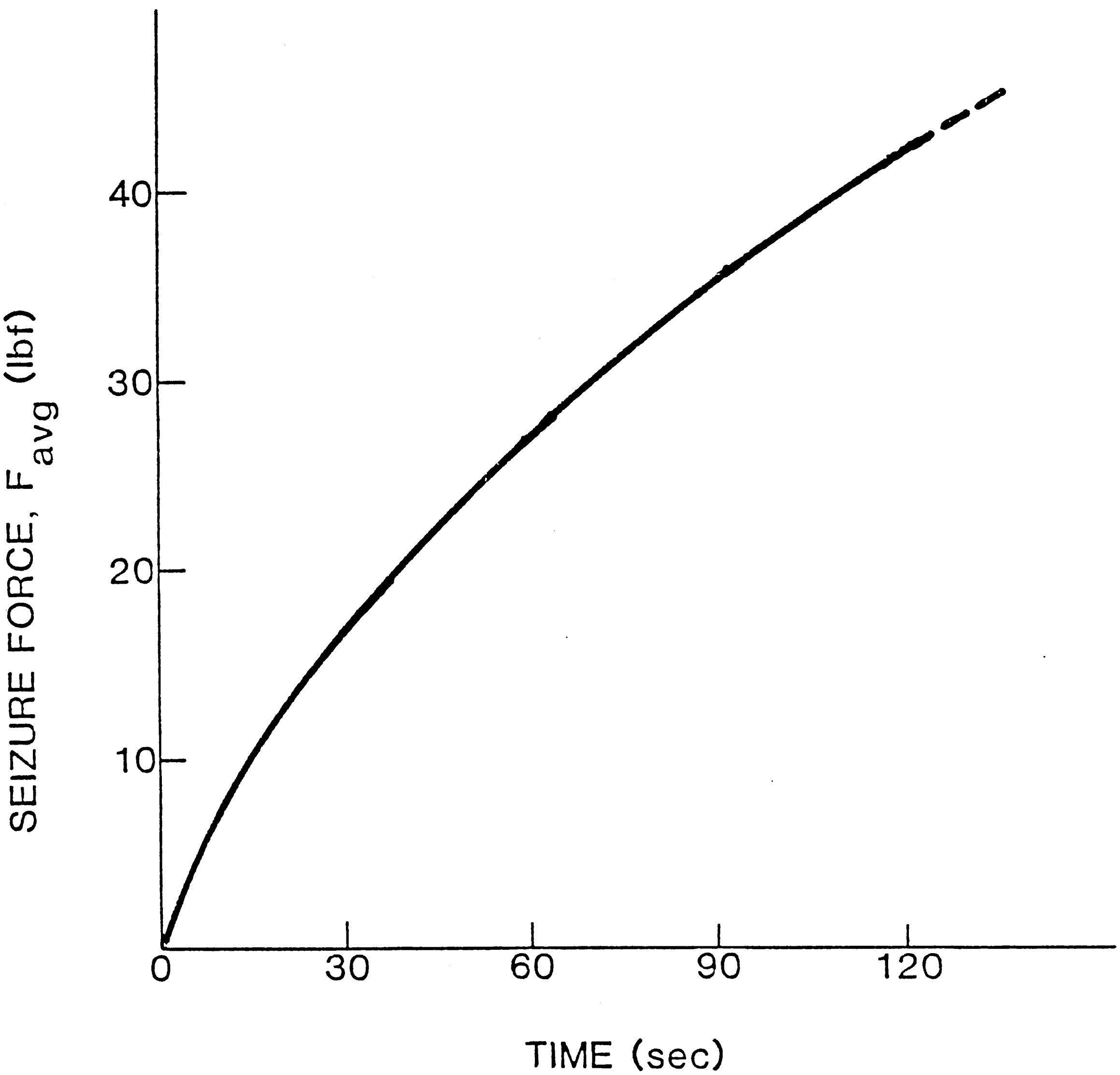


Figure. 22. Seizure Force Versus Time.

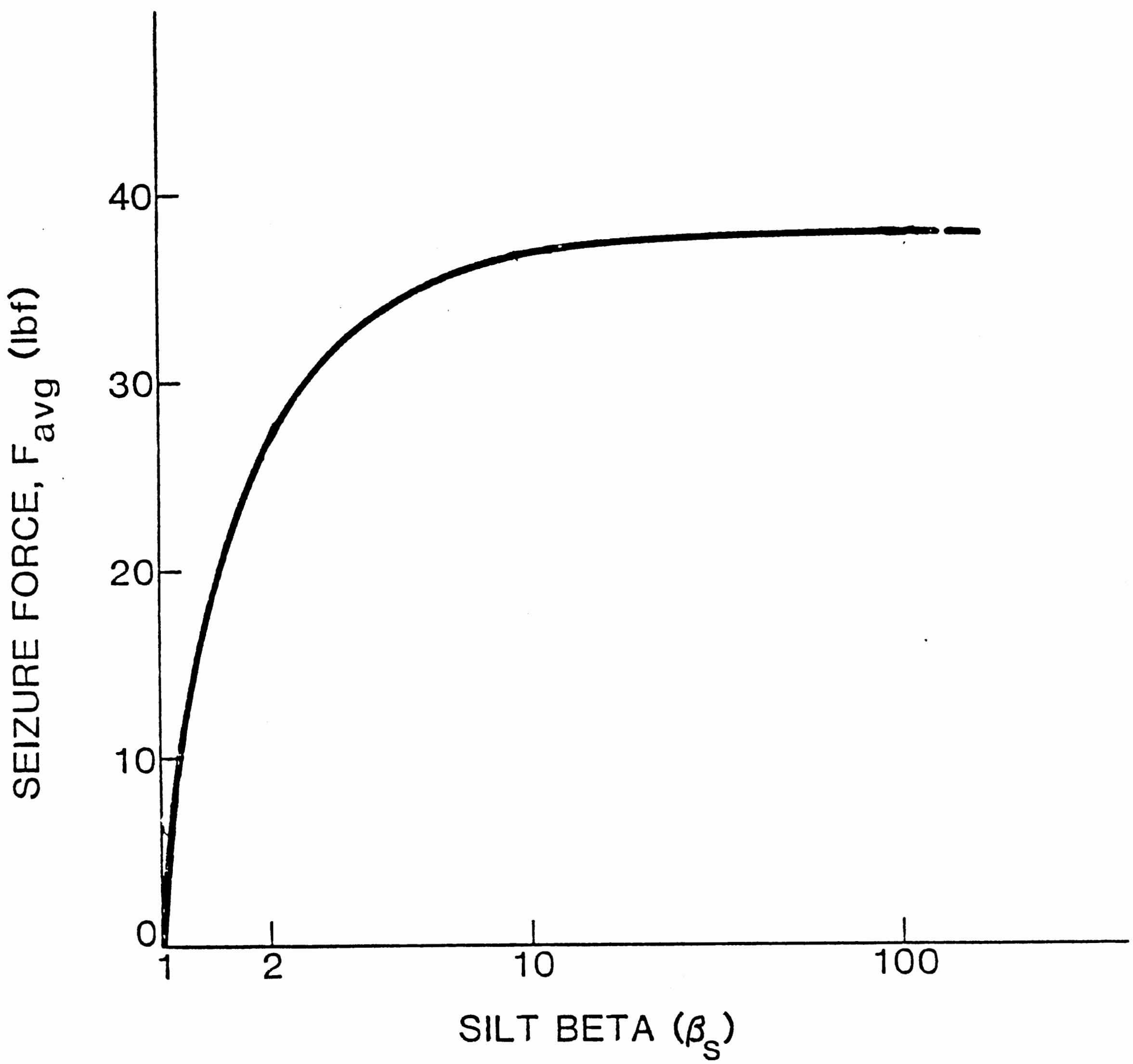


Figure. 23. Seizure Force Versus Silt Beta.

NTd must be measured.

It should be noted that in the presence of dither, various parameters and associated symbols are altered: note that the dither changes NDd to NPDd and its coefficients to βp and NPTd. The relationship between β and βp is assumed to be similar to Nikitin's experimental results (Fig.9). The force required to break the seizure decreases as a function of frequencey--in a manner similar to exponential decay. Thus, the logarithm of silt Beta ratio also decreases:

$$\frac{\ln(\beta p)}{\ln(\beta)} = A\ e^{-Cw} \tag{39}$$

where A and C are constants for the exponential decay function, and w is the frequency of the controlled pulsation.

Solving for βp gives:

$$\beta p = \beta^{Ae^{-Cw}} \tag{40}$$

For a given value of βp, the range of NPTd can be determined as follows. From the definition of silt Beta,

$$\beta p = \frac{NTu\ e^{(-Bu\ \ln^2(h))}}{NPTd\ e^{(-Bd\ \ln^2(h))}} \tag{41}$$

where h is the nominal clearance at the instance when the silt Beta value is obtained. Solving for Bd gives:

$$Bd = Bu + \frac{\ln(\beta p)\ Ga}{\ln^2(h)} \tag{42}$$

where

$$Ga = NPTd\ /\ NTu \tag{43}$$

$$0 < Ga < 1 \tag{44}$$

From now on, the NPTd range is specified by the range of Ga because NTu is a constant. The range exhibited by Eq. 44 is further narrowed by the non-desorption constraints. When the following inequality is satisfied, fine particle size designated by F (smaller than h) does not desorb from the piston-bore clearance:

$$\frac{2\ NTu\ Bu\ \ln(F)}{F} e^{(-Bu\ \ln^2 F)} > \frac{2\ NPTd\ Bd\ \ln(F)}{F} e^{(-Bd\ \ln^2 F)} \tag{45}$$

Eliminating and organizing terms, the inequality becomes:

$$\ln(Bu\ /\ (Ga\ Bd)) > (Bu - Bd)\ \ln^2 F \tag{46}$$

However, Eq. 42 is the same as:

$$Bu - Bd = -\ n\ \ln(\beta p\ Ga) \tag{47}$$

where

$$n = 1\ /\ \ln^2(h) \tag{48}$$

Replacing Bu - Bd in Eq. 46 with Eq. 47, the inequality becomes:

$$Bd < \frac{Bu}{Ga\ (\beta p\ Ga)^{-n\ ln^2 F}} \tag{49}$$

The result of dividing both sides of Eq. 49 with Bu and organizing it is:

$$\frac{Bd}{Bu} < \frac{(\beta p\ Ga)^{n\ ln^2 F}}{Ga} \tag{50}$$

When the desorption constraint Bd/Bu > 1 is satisfied, retained particles with sizes greater than h do not desorb from the piston-bore assembly. As a result, Bd/Bu > 1 and the insertion of Eq. 48 into Eq. 42 yields:

$$\frac{Bd}{Bu} = 1 + \frac{n\ ln(\beta p\ Ga)}{Bu} > 1 \tag{51}$$

Solving Eq. 51 for Ga results in:

$$Ga > 1 / \beta p \tag{52}$$

Figure 24 illustrates the range of Ga (= NPTd/NTd) as a function of Bd/Bu, where desorption constraints limit the range of Ga. Therefore, by the experimental determination of silt Beta (see Eq. 39), the range of non-desorption NPTd can be determined. If the particle counter analysis reveals that the actual NPTd is outside of the non-desorption range, then the size range of the desorption can be identified.

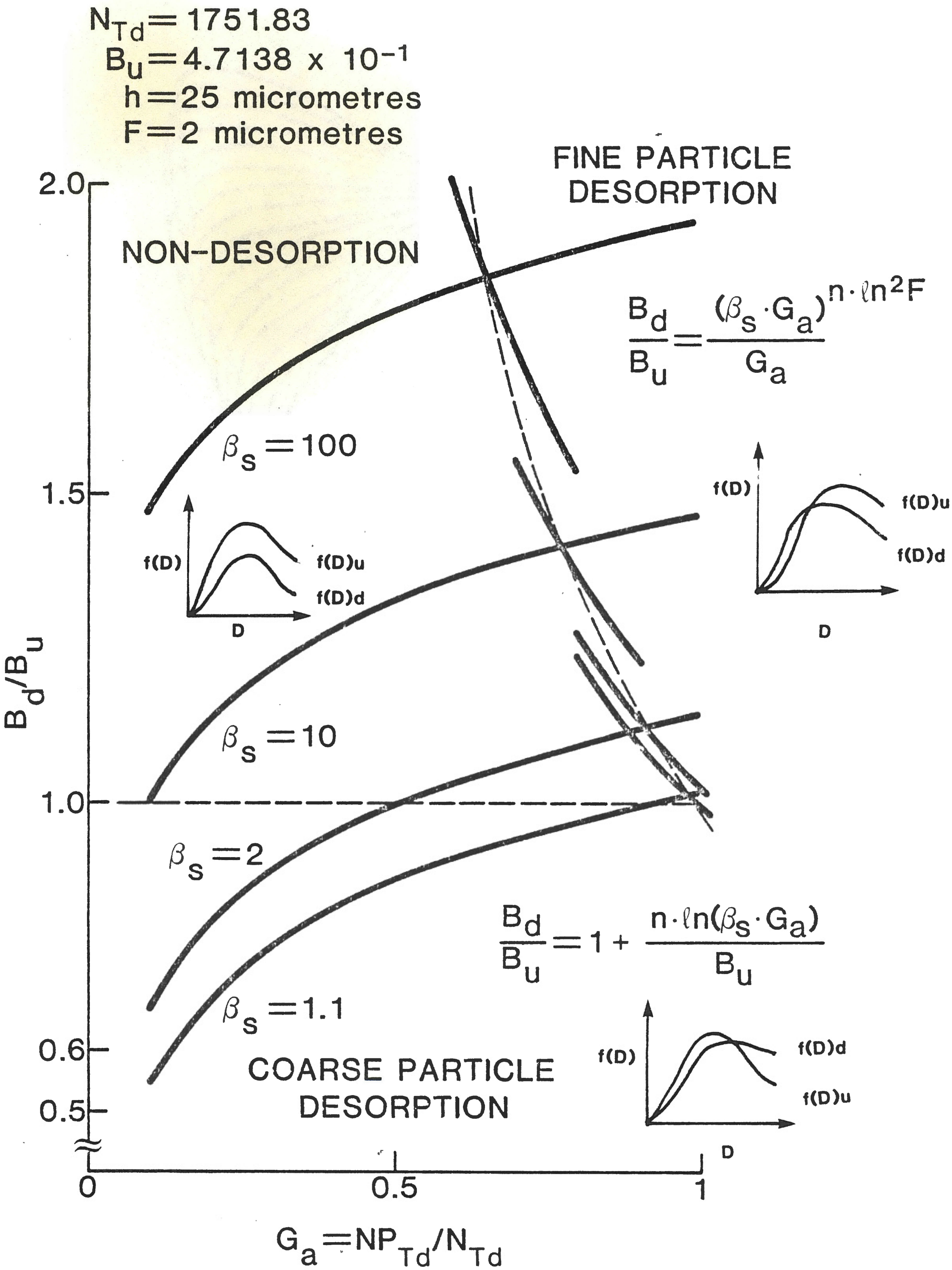


Figure. 24. Desorption.

CHAPTER IV

VERIFICATION OF SILT PROCESS MODEL

Experimental Scheme

As a result of the simulation studies, several experimental objectives ensue. The first is to investigate whether the leakage through a piston-bore assembly follows the theory that is presented in the previous chapter, and the second is to develop from the experimental data an equation describing the silt Beta as a function of particle size.

The leakage equation (Eq. 2) shows that the eccentricity of a piston in the bore greatly influences the leakage. This was confirmed by many researchers in the past. Kamijo and Dransfield [5, 8], in particular, experimentally proved that when the pressure difference across the clearance of a piston-bore assembly is below 1000 psi, then the eccentricity is always close to one.

Besides the eccentricity, the absolute viscosity also affects the outcome of the leakage flow. Since Dexron II contains additives, the effect of this additive over the absolute viscosity is investigated by varing the shear rate of the leakage through the clearance of a piston-bore assembly.

Since no information on the silt Beta values of a given piston-bore assembly is available, first the method to identify a silt Beta value at each particle size is developed, and then, using this method, the silt Beta values versus the particle size of a given piston-bore assembly is determined. The variation of silt Beta values is investigated for three different clearances with two different piston land lengths (a total of six different cases). These silt Beta values obtained through the experiments are then implemented in the computer simulation program for analyzing the validity of the silt process model.

Development of Experimental Facility

The generic test housing (Fig. 25) and piston and bore test pieces were fabricated. A combination of a piston and bore was installed inside the housing, as shown in Fig. 25. The actuation rod controlled the position of the piston with respect to the bore. When the actuation rod inserted the piston inside the bore, as shown in Fig. 25, the nominal clearance between the piston and bore was established by their diameter difference. A complete set of piston and bores with their dimensions is listed in Table II.

Table II has two measurements. The above measurement was performed optically by STADCO Automatic Co. in Fairborn, Ohio. The lower tabulated value was obtained by the precision measurement device (minimum resolution of 0.254 micrometer) manufactured by Pratt & Whitney.

Pistons No.1 through 9 and bore No. 1 were made out of soft cold rolled steel 12L/4. Pistons No. 10 through 18 and bore No. 2 were made

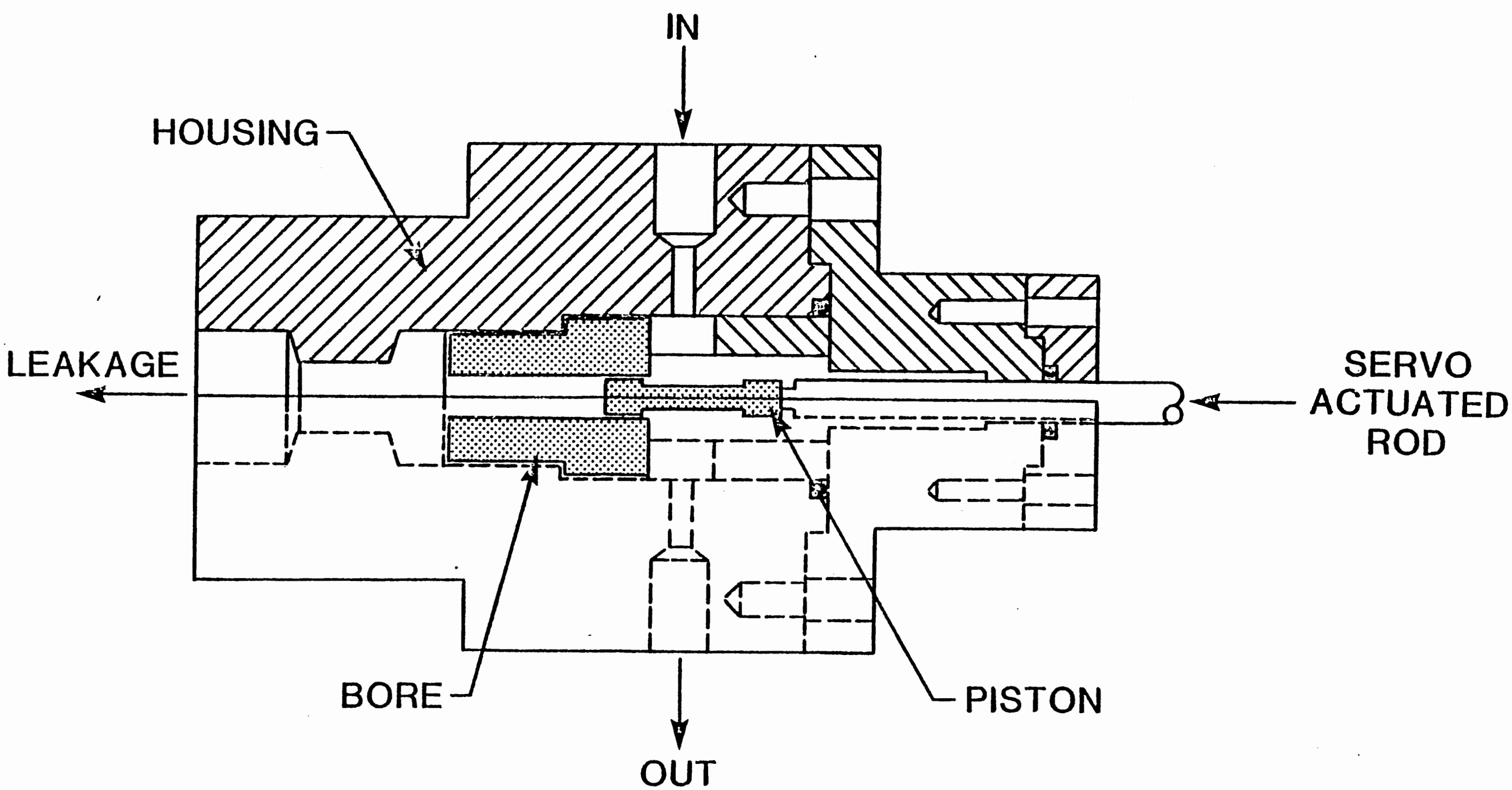
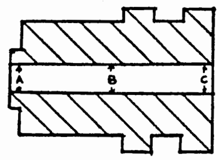


Figure. 25.   Piston-Bore Assemblies and Housing.

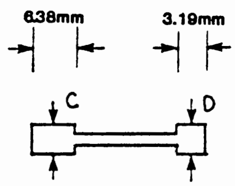
TABLE II

PISTON-BORE DIMENSION



BORE DIMENSION (mm)

No. Hardness Brinell	A	B	C
1 Hb=120	6.350	6.350	6.350
2 Hb=400	6.351	6.351	6.351
3 Hb=725	6.351	6.351	6.351



PISTON DIMENSION (mm)

No. Hardness Brinell	C	D						
1 Hb=120	6.341 6.3375	6.341 6.3381	11 Hb=400	6.339 6.3373	6.341 6.3373	21 Hb=725	6.341 6.3383	6.341 6.3398
2 Hb=120	6.341 6.3406	6.341 6.3428	12 Hb=400	6.339 6.3381	6.339 6.3385	22 Hb=725	6.332 6.3284	6.332 6.3309
3 Hb=120	6.341 6.3373	6.341 6.3423	13 Hb=400	6.330 6.3298	6.330 6.3298	23 Hb=725	6.332 6.3322	6.332 6.3322
4 Hb=120	6.329 6.3296	6.329 6.3296	14 Hb=400	6.330 6.3296	6.330 6.3309	24 Hb=725	6.332 6.3322	6.330 6.3322
5 Hb=120	6.332 6.3322	6.330 6.3309	15 Hb=400	6.329 6.3303	6.330 6.3296	25 Hb=725	6.322 6.3220	6.322 6.3220
6 Hb=120	6.332 6.3296	6.330 6.3322	16 Hb=400	6.320 6.3220	6.320 6.3220	26 Hb=725	6.320 6.3220	6.322 6.3220
7 Hb=120	6.320 6.3220	6.320 6.3220	17 Hb=400	6.319 6.3195	6.319 6.3195	27 Hb=725	----- 6.3474	----- -----
8 Hb=120	6.322 6.3220	6.322 6.3220	18 Hb=400	6.322 6.3220	6.322 6.3220	28 Hb=725	----- 6.3474	----- -----
9 Hb=120	6.322 6.3220	6.320 6.3220	19 Hb=725	6.339 6.3398	6.341 6.3398	29 Hb=725	----- 6.3449	----- -----
10 Hb=400	6.339 6.3398	6.341 6.3373	20 Hb=725	6.341 6.3398	6.341 6.3398	21 Hb=725	----- 6.3449	----- -----

of cold rolled steel 1018, carburized, and have a hardness of 43 to 45 on the Rockwell C scale. Pistons No. 19 through 26 and bore No. 3 were made of A-2 and hardened to achieve a 62 to 64 hardness on the Rockwell C scale.

The test hydraulic circuitry, shown in Fig. 26, was developed and confirmed to control the following parameters:

1. temperature
2. pressure
3. cleanliness through high Beta value filter
4. concentration of the injected particles for the entire test period

The accumulator was oriented vertically to avoid providing any space where the injected particles might settle.

Leakage Flow Test

The leakage flow of a piston-bore assembly under clean fluid is well defined, and many researchers [6, 8] have experimentally proved the integrity of Eq. 2. Dransfield [5] verified the validity of Eq. 2 with Shell Tellus 27 (known as a non-additive industrial mineral hydraulic oil), and Kamijo et al. [8] verified its validity with turbine oil #140. Since Equation 2 was derived from the Poiseuille flow equation, the equation is thus valid for laminar flow and for cases where the constant viscosity and density assumption is appliable.

For this study, Dexron II was chosen as a test fluid because of its

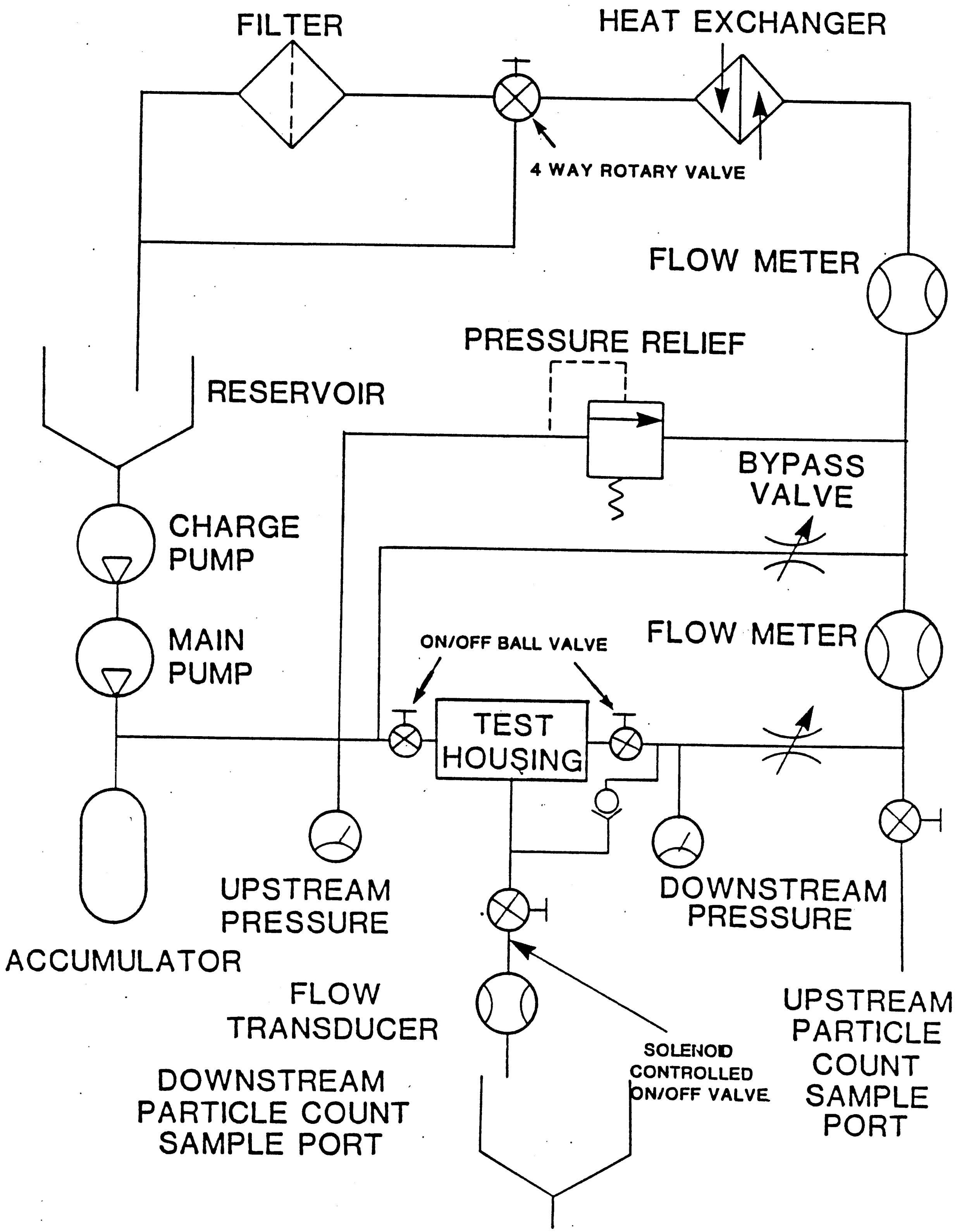


Figure. 26. Hydraulic Circuitry.

wide industrial use; however, it had to be experimentally tested as to whether it follows Eq. 2.

The test housing was installed and connected to the test hydraulic circuitry as shown in Fig. 26. The piston was inserted into the bore as Fig. 25 shows, and pressure was applied such that there was 1000 psi difference between the upstream and downstream of the leakage flow through the piston-bore clearance.

The combinations of the piston-bore assemblies are displayed in Table III. Bore No. 3 and pistons No. 19, 23, and 26 were made out of the same material and have the same hardness; therefore, the temperature effect over the nominal clearance of a given combination of a piston-bore assembly was minimized. Moreover, each piston had the land length of 3.19mm and 6.38mm; thus, the effect of land length over the leakage flow could be investigated. The leakage flow test results of Table III were plotted against the theoretical leakage flow derived from Eq. 2. The values used for the theoretical calculations are

viscosity :	μ = 2.32E-06 lbf/in^2
clearance :	h = 5, 10 and 15 μm
land length :	L = 3.19 and 6.38 mm.
upstream pressure :	Pu = 1000 lbf/in^2
downstream pressure :	Pd = 0 lbf/in^2 (atomospheric pressure)
spool diameter :	D = 6.320, 6.330 and 6.340 mm.
eccentricity :	e = 0 and 1.

Figures 27 and 28 show that all the experimental results stay within both

TABLE III

LEAKAGE FLOW TEST (DEXRON II CLEAN FLUID)

BORE No.	SPOOL No.	NOMINAL CLEARANCE μm	LAND LENGTH e; eccentricity	
			6.38 mm	3.19 mm
3	19	5	2.73 mL/min e=1.222(h=5) e=0.764(h=6)	6.96 mL/min e=1.184(h=5) e=.963(h=6)
3	23	10	10.0 mL/min e=0.568(h=5)	21.6 mL/min e=0.634(h=5)
3	26	15	40.0 mL/min e=0.711(h=5)	97.0 mL/min e=0.868(h=5)

h = actual clearance size in micrometer.

extremes of eccentricity, except for the results for a nominal clearance of 5 micrometers. A close examination of Table II shows that the bore No. 3 and piston No. 19 combination has a clearance of 6 micrometers, according to the measurement done by the Pratt & Whitney precision measurement device. If the clearance of 6 micrometers were used instead of 5 micrometers in Eq. 2, the estimated eccentricities would change as shown in Table III and would be less than unity.

Dransfield and Sasada's experimental data show that a given piston-bore assembly always presents eccentricities above 0.9 and close to 1 for Newtonian fluids. However, the eccentricity of the experimental data shown in Fig. 27 and 28 are not always above 0.9.

Since Dexron II is known to contain proprietary additives, it may have significant non-Newtonian characteristics. To examine the non-Newtonian characteristics of Dexron II, a copper capillary of 190 inches with 1.4mm inner diameter was used. One end of the capillary was left open to atmospheric pressure, and Dexron II pressured higher than atmospheric pressure was applied to the other end. When the higher pressure was fixed, the pressure drop per unit length became constant. The leakage through the capillary was collected in a graduated cylinder, and the time taken to collect the leakage was measured. The leakage flow rate was calculated by the leakage volume in the graduated cylinder divided by the time needed to collect the volume. The fully developed laminar equation (Eq. 53) was used to find the absolute viscosity:

$$\mu = \pi D^4 (Pu - Pd) / (128 Q L) \tag{53}$$

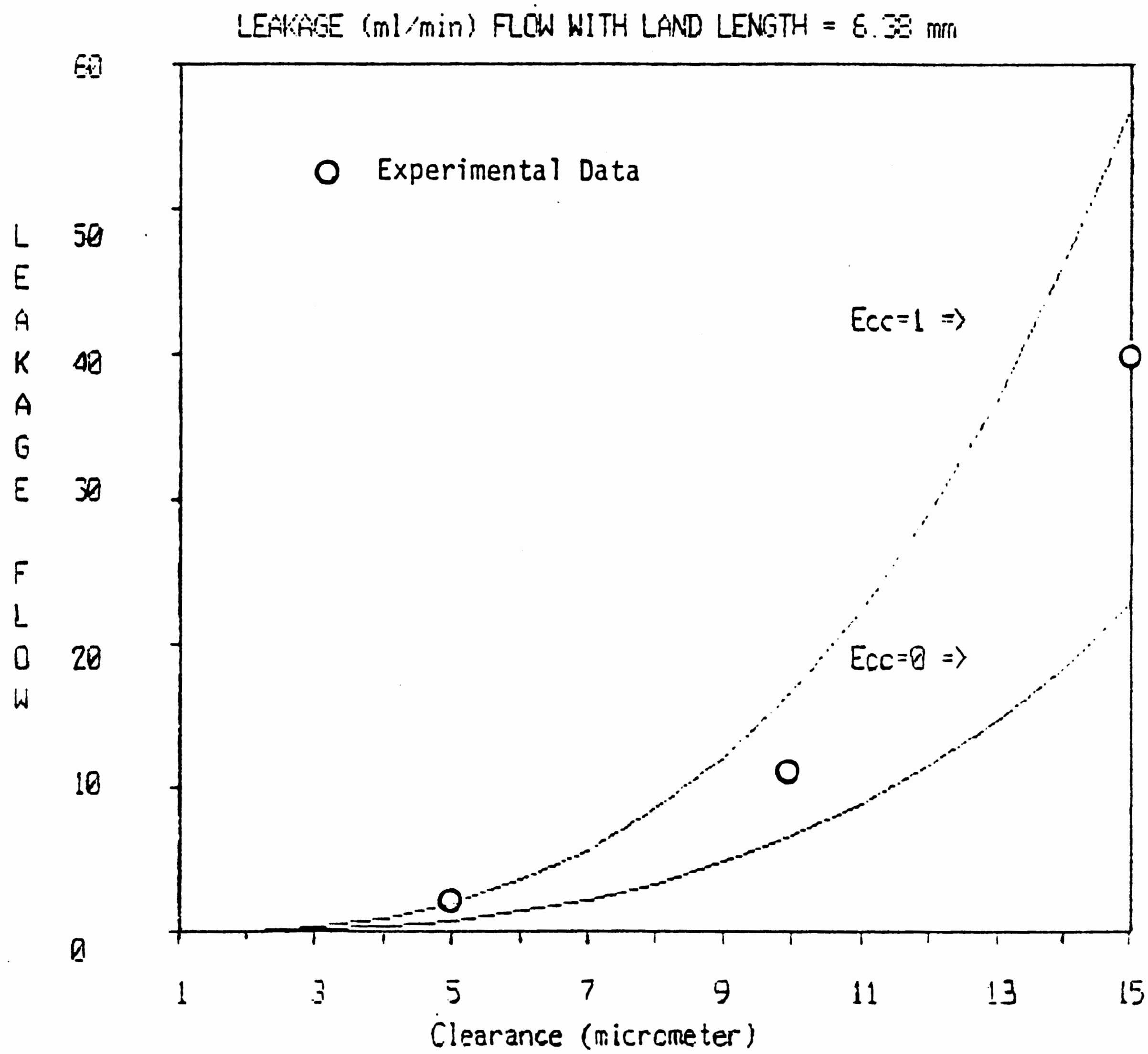


Figure. 27. Leakage Flow of a Piston-Bore Assembly With 6.38mm Land Length.

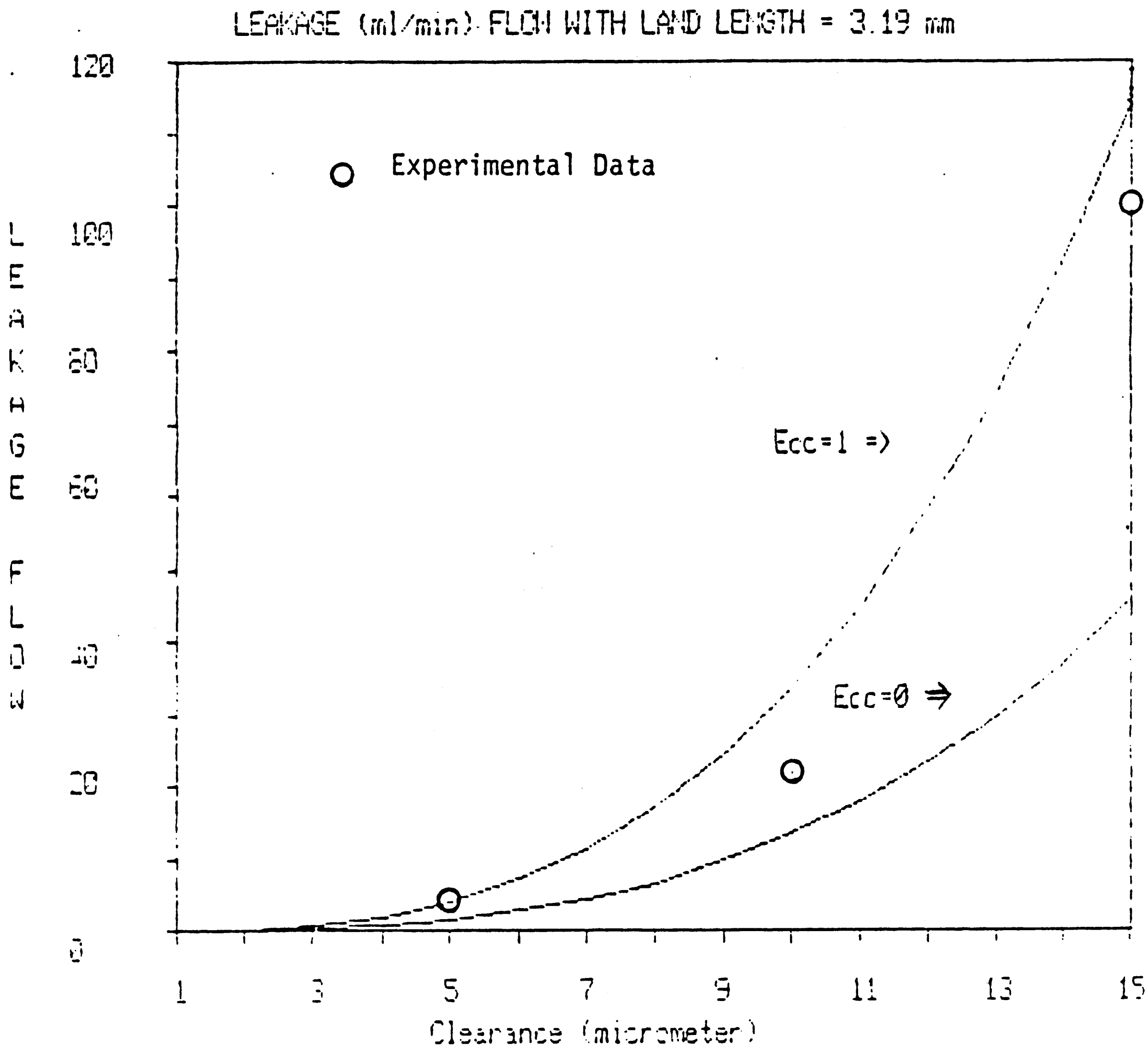


Figure. 28. Leakage Flow of a Piston-Bore Assembly With 3.19mm Land Length.

where

D = inner diameter of capillary

Q = leakage flow rate

Pu = higher pressure (end of capillary)

Pd = lower pressure (end of capillary)

L = length of capillary

In addition, the highest shear rate, which occurs at the wall of a capillary, was also calculated by:

$$\left.\frac{du}{dr}\right|_{r=R} = - \frac{4\ VAVG}{R} \tag{54}$$

where R is the inner radius of the capillary, and VAVG is calculated by dividing the leakage by the cross sectional area of the capillary.

Figure 29 shows the experimental result of absolute viscosity versus the shear rate at the capillary wall. This figure clearly demonstrates that the absolute viscosity increases as the shear rate increases. Yet, the increase is nonlinear. To correlate this capillary test data with the piston-bore assemblies' test, the shear rate of the leakage flow through the piston-bore assemblies must be known.

An accurate value of the shear rate of the leakage flow through the piston-bore assemblies can not readily be calculated; however, the shear rate was estimated by the following approach. If the piston is assumed to be concentric with the bore, then the average flow velocity could be calculated by:

$$VAVG = Q \,/\, (\pi h\ (D - h)) \tag{55}$$

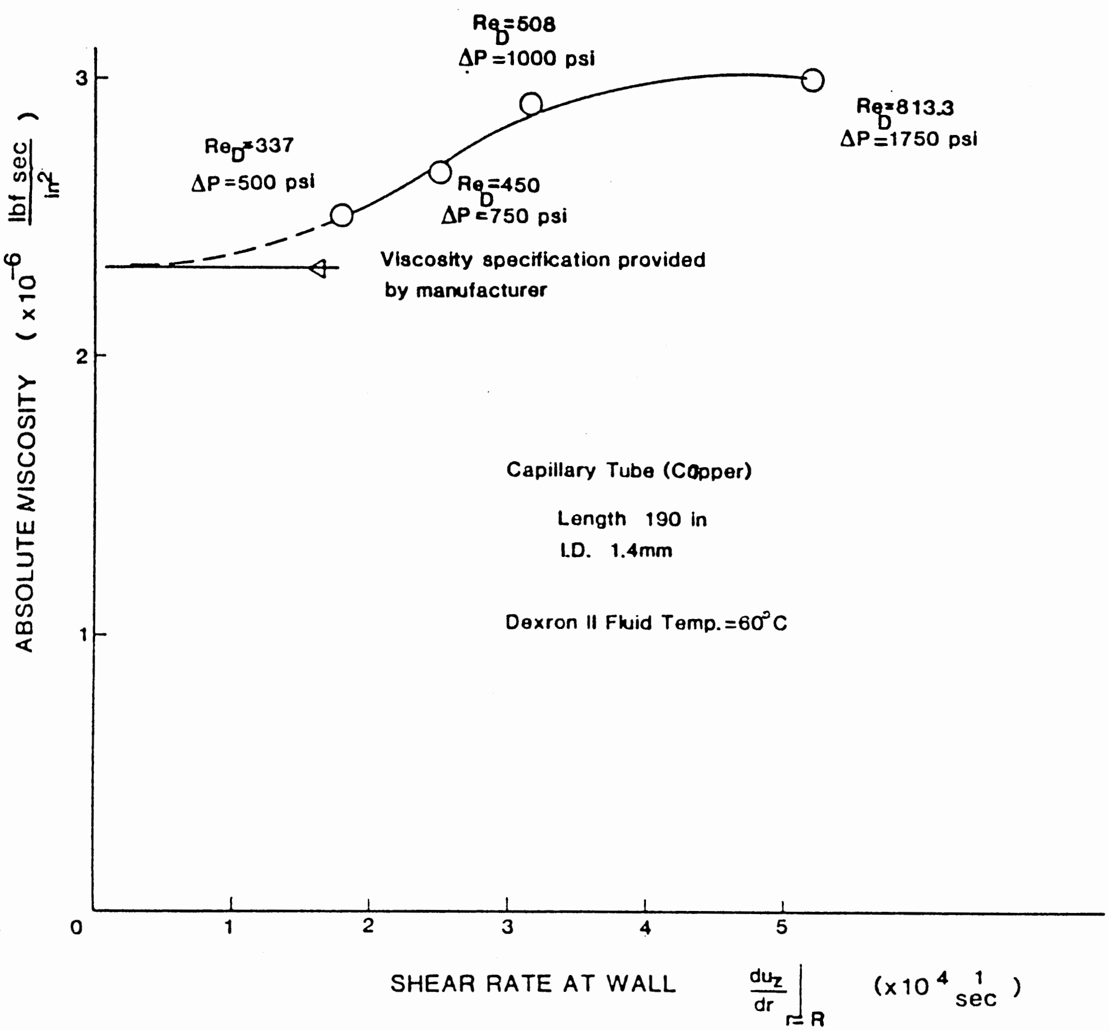


Figure. 29. Absolute Viscosity Versus Shear Rate at the Wall of the Capillary Tube.

where

Q = the leakage flow

h = the clearance size

D = the inner diameter of the bore

The shear rate was then estimated by using the fully developed laminar flow equation for flat plates and is:

$$\left.\frac{du}{dy}\right|_{y=h/2} = -\frac{6\ VAVG}{h} \qquad (56)$$

where u, h and y are explained in Fig. 30. In reality, the piston is not normally concentric with the bore; therefore, if the actual leakage flow (leakage under eccentric condition with a higher flow rate than concentric condition) is used in Eq. 55 to find the estimated shear rate in Eq. 56, this shear rate becomes approximately the average shear rate of the eccentric condition.

Based upon the data in Table III, the average shear rate was calculated using Eq. 55 and 56 in Table IV. It is evident that the leakage through the clearance of a piston-bore assembly experiences a shear rate 9 to 50 times higher than the shear rate found in the capillary test. However, from the experimental data shown in Fig. 27 and 28, the absolute viscosity does not increase at the same rate as the shear rate. If the absolute viscosity were to increase in proportion to the shear rate, the leakage would have been 1/9 or 1/50 of what actually was measured. Furthermore, it is reasonable to postulate that the increased absolute viscosity due to the high shear rate results in a lower leakage flow in

For the case of π (D−h) >> h

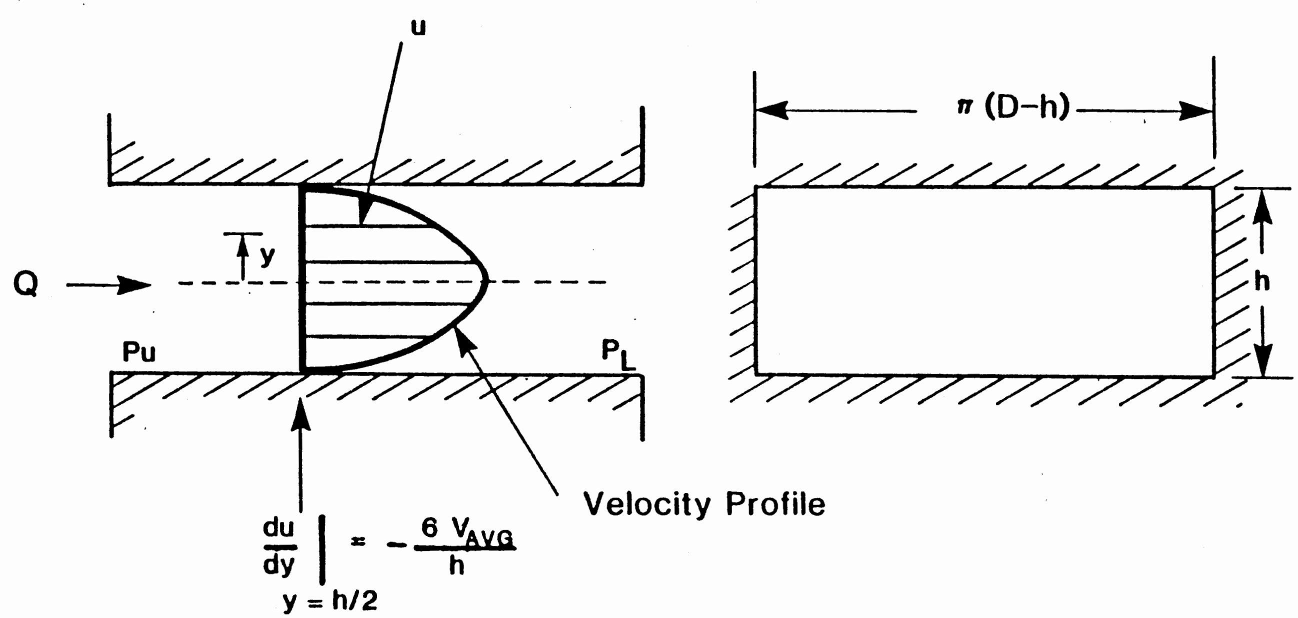


Figure. 30. Laminar Flow Through Flat Plates.

TABLE IV

AVERAGE SHEAR RATE

BORE No.	SPOOL No.	NOMINAL CLEARANCE μm	LAND LENGTH	
			6.38 mm	3.19 mm
3	19	5	452700 l/sec	1154000 l/sec
3	23	10	556200 l/sec	1201000 l/sec
3	26	15	955700 l/sec	2318000 l/sec

the piston-bore assemblies than the leakage flow calculated by the low shear rate absolute viscosity. For these reasons, many of the estimated eccentricities in Table III are below 0.9.

Leakage Flow Test Under Contamination

Though Laurenson [9] indicated in his work that the leakage through the capillary and fine slots seem to decrease exponentially when solid particles silt in the passage, no experimental data is available to support his claim for the case of a piston-bore assembly. In addition, Laurenson used siliconcarbide particles, which are known to have a narrow band size distribution. He did not use ACFTD full distribution, which has a wide band size distribution. This experiment therefore investigates the change in the leakage flow through a piston-bore assembly clearance when the fluid is contaminated with ACFTD full size distribution.

ACFTD full distribution was first injected into the reservoir to achieve a 100 mg/L gravimetric level of the test fluid (Dexron II) circulating through the test hydraulic circuit. Prior to this injection, the filter system was isolated by the valve at the upstream of the filter system. Then, the No. 19 piston was inserted into the No. 3 bore to form the nominal 5 micrometer test clearance between the land of the piston and bore surface. Initially, the test housing was completely isolated from the test circuitry by two ball valves: one at the upstream and the other at the downstream of the test housing. Hence, there was no axial pressure on the piston-bore assembly. Then, the closed ball valves were opened to introduce 1000 psi axial pressure on the piston-bore assembly. The constant axial pressure was well maintained by controlling the bypass valve.

The leakage flow was collected in a clean bottle for the analysis of Nd: the downstream particle size distribution. When the leakage came out in the form of droplets, these droplets were counted and timed to investigate the silt accumulation in the clearance. Prior to the injection of dust, while the test fluid was clean, the volume of a droplet coming out of the 1/4" tube was determined to be 0.0392 ml/drop and out of the 3/8" tube was 0.0335 ml/drop. The system fluid temperature was always maintained at 60 degrees Celsius plus or minus 2 degrees.

Figure 31 shows the result of the leakage volume in the number of droplets versus time in seconds. The total number of droplets was no more than 17 over a period of 10 minutes for the combination of bore No. 3 and spool No. 19. The leakage flow clearly demonstrated a progressively decreasing trend and deviated further away from the leakage flow with clean fluid as the silting time increased.

If Laurenson's postulate is correct, the leakage can be described by:

$$Q(t) = Qo\ e^{-kt} \qquad (57)$$

where

Qo = the leakage under clean fluid

k = a constant

t = time

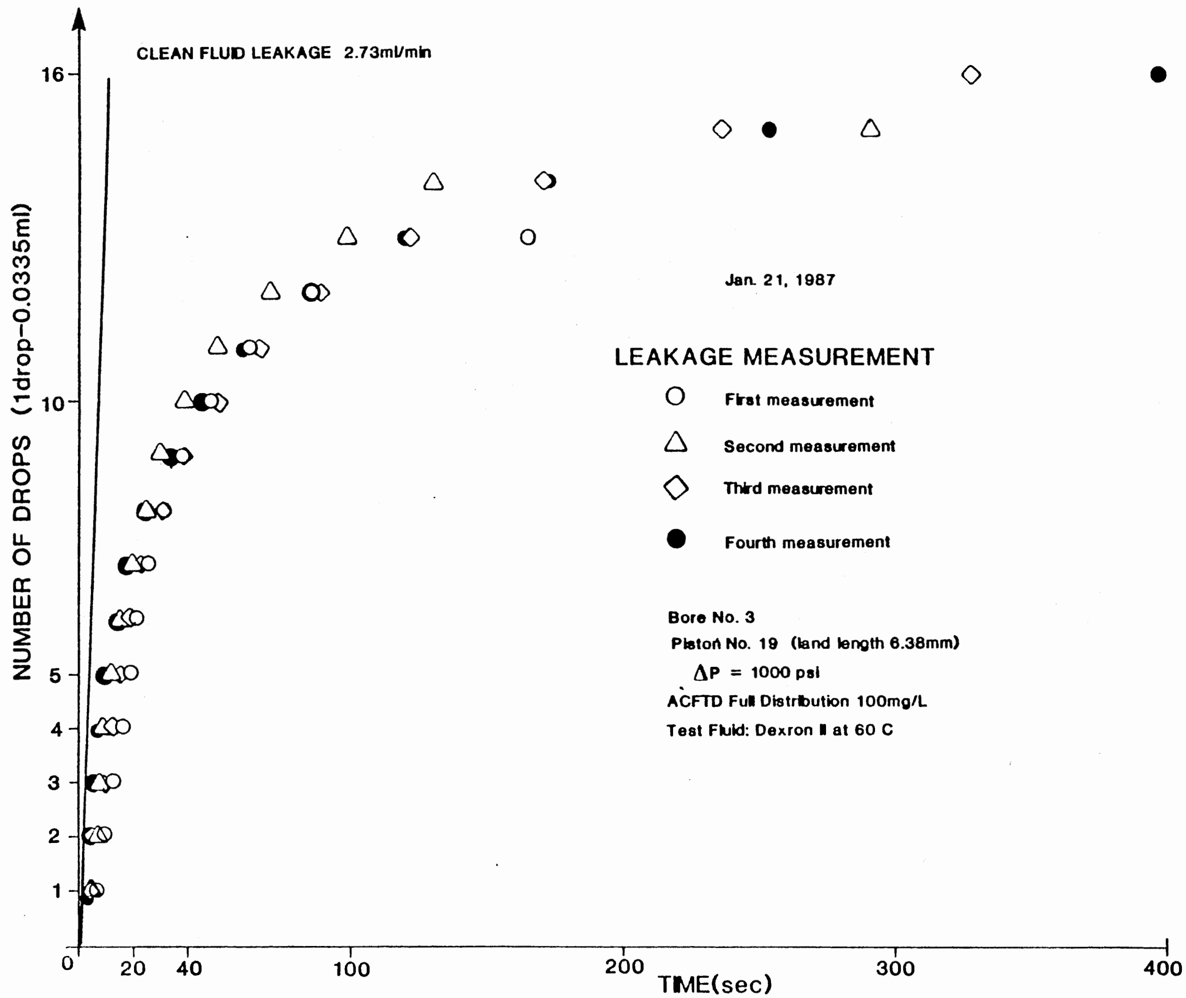


Figure. 31. Piston-Bore Assembly Leakage Volume Under Contamination (100 mg/L).

Equation 57 can be modified as:

$$\ln (Q(t) / Qo) = -kt \tag{58}$$

Equation 58 shows that the logarithm of Q(t) divided by Qo has a linear relationship with time. Figure 32 was derived from Figure 31 to investigate the relationship between the leakage flow and time, as Fig. 25 shows that the leakage passing through the clearance must travel until either the end of the bore piece or the end of the housing to form a droplet. There was a delayed response due to the long passage. The first 20 seconds of Fig. 32 clearly demonstrates the delayed response of the leakage flow rate, which is not conspicuous in Fig. 31. After 20 seconds, the leakage flow rate quickly diverged from the linear relationship. The same leakage test was repeated with 25 mg/L of ACFTD full distribution instead of 100 mg/L.

Figure 33 shows higher leakage volume with a moderately decreasing trend. Since there is a concentration of 25 mg/L of solid particles, it is very logical to have more leakage volume because there are fewer particles to block the passage than when the concentration is 100 mg/L. Figure 34 shows the leakage flow rate analysis. As with Figure 32, there is a delayed response at the beginning, and then later the flow rate deviates from the straight line.

For the first 70% to 80% of the total leakage volume, Laurenson's postulate may correctly predict the reduction of the leakage flow; however, the experimental data of Fig. 32 and 34 clearly show that the

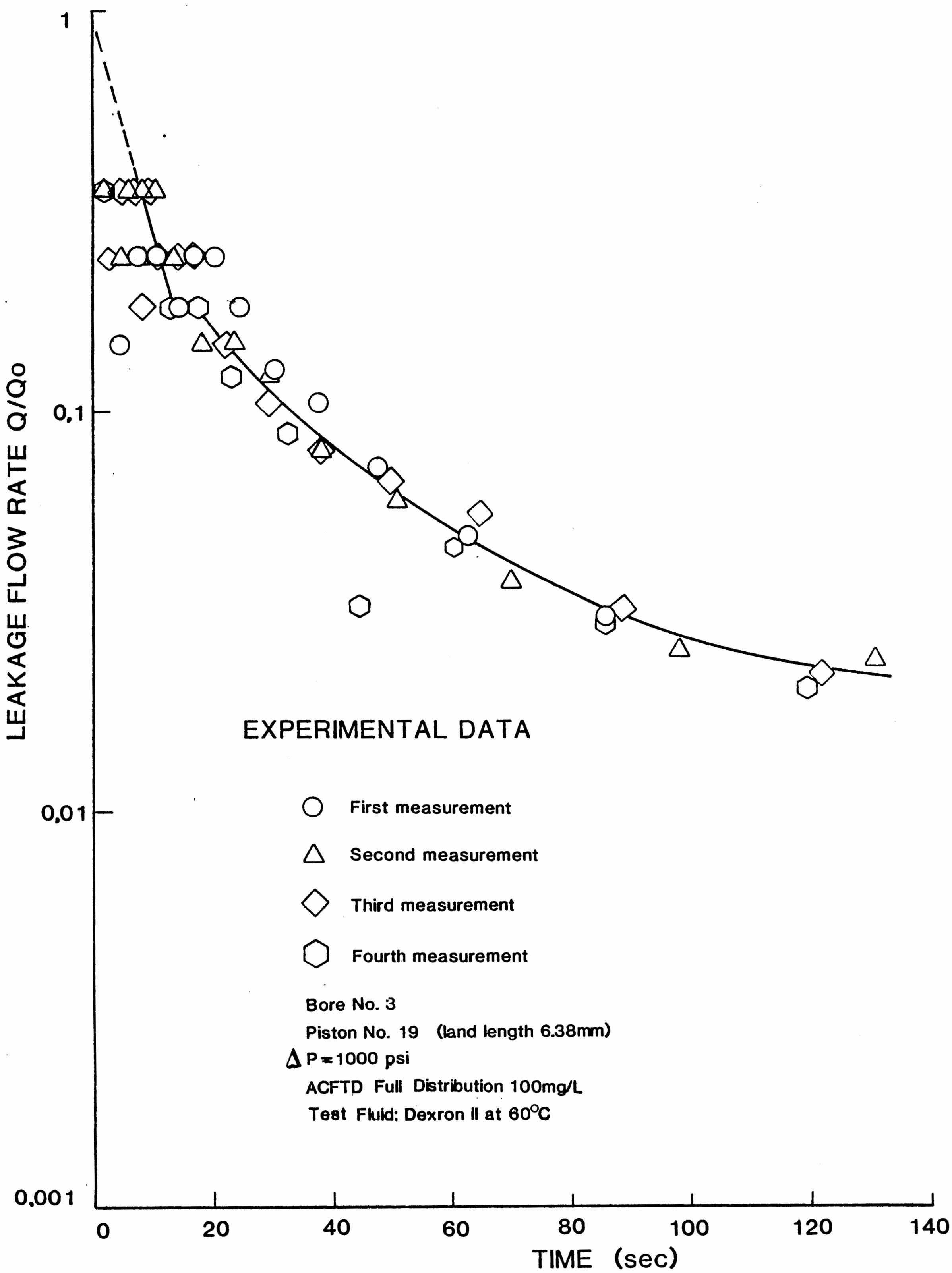


Figure. 32. Piston-Bore Assembly Leakage Flow Rate Under Contamination (100 mg/L).

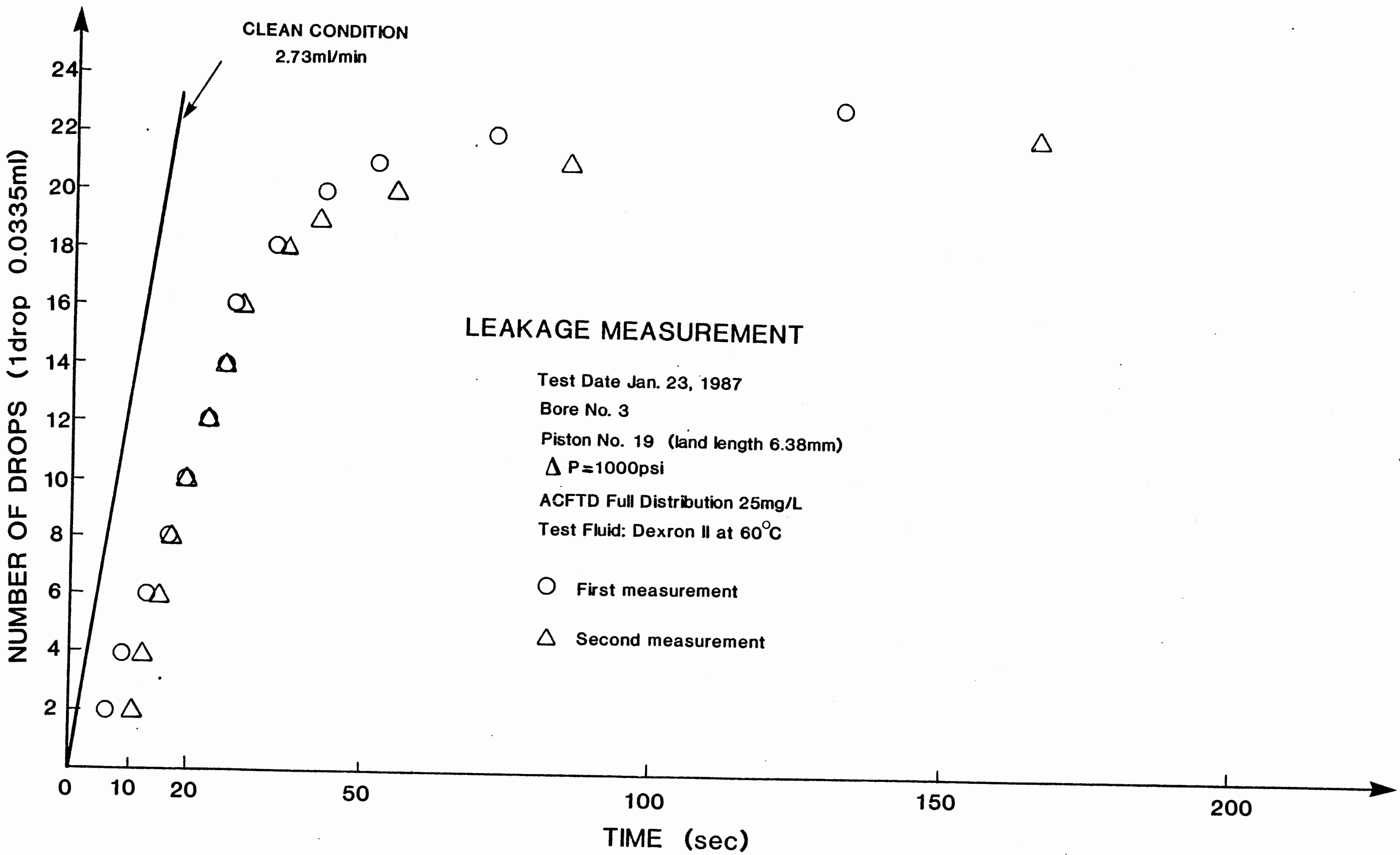


Figure. 33. Piston-Bore Assembly Leakage Volume Under Contamination (25 mg/L).

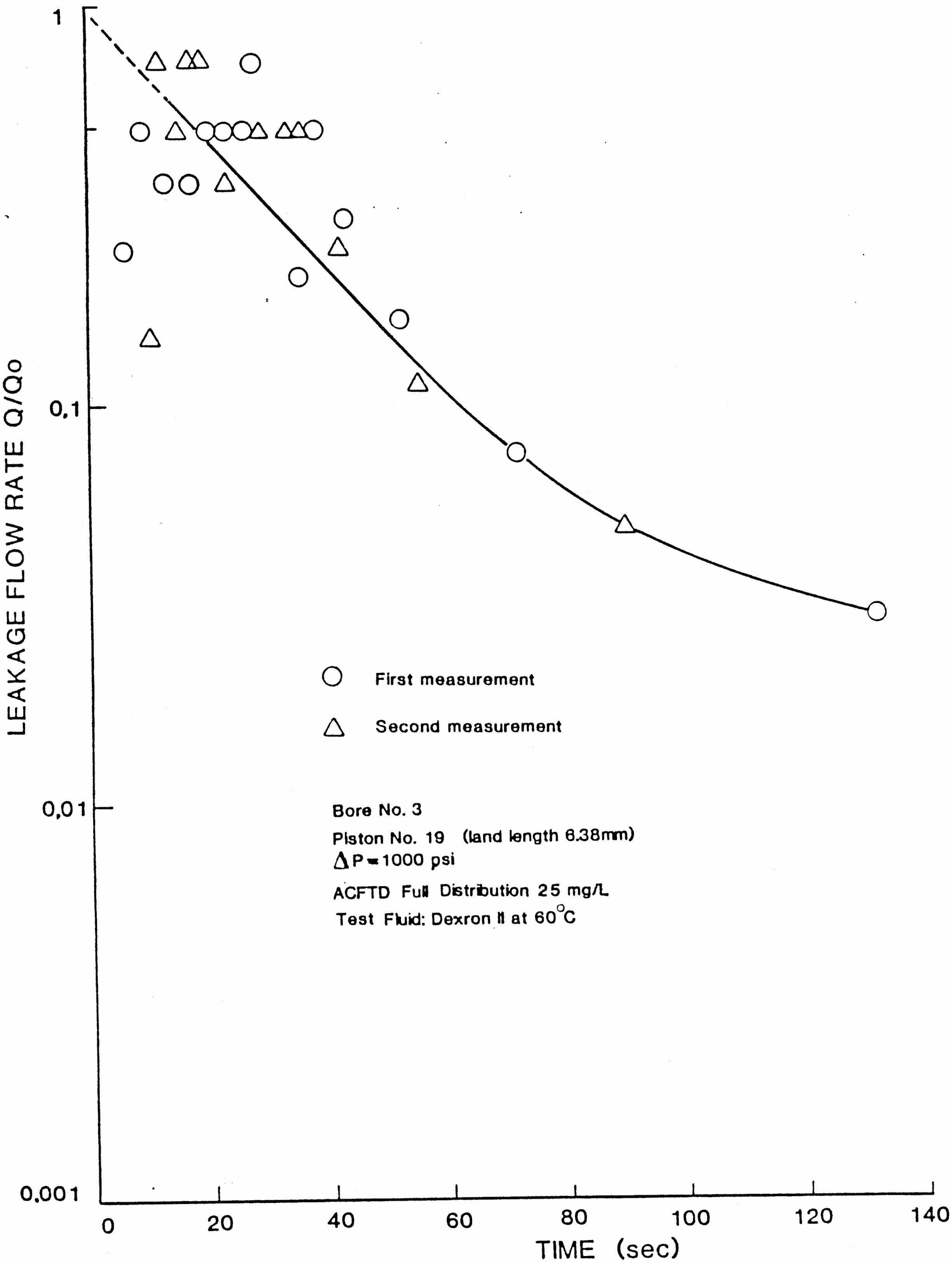


Figure. 34. Piston-Bore Assembly Leakage Flow Rate Under Contamination (25 mg/L).

leakage persists longer than what the initial leakage flow rate projects. The following hypothesis may explain why the leakage flow rate behaves the way Fig. 32 and 34 show. Since Laurenson selected solid particles that have a narrow range of distribution for his experiment, if it is assumed that each particle silted in a piston-bore assembly clearance equally contributes to the reduction of the leakage flow through the clearance, the following differential equation should hold to be true:

$$dQ / dt = -kQ \tag{59}$$

Assuming that particles are uniformly distributed in the fluid, a unit leakage flow rate through the clearance should result in a corresponding decrease in the rate of change in the leakage flow rate per unit time. The solution of Eq. 59 is Eq. 57, and this supports Laurenson's work.

However, when the solid particles have a wide distribution such as ACFTD full distribution, the leakage flow follows Eq. 59 at first, because for a given clearance size, only a particle whose size is close to the clearance can silt. These particles also play a major role in reducing the leakage flow rate at the beginning of the leakage test (see Fig. 35a). Smaller particles silt around the particles whose sizes are close to the clearance size and form a caking condition, as shown in Fig. 35(b). As a result, when the initial particles whose sizes are close to the clearance size no longer have a space to silt, only smaller sized particles can wander through the tortuous path of the cake

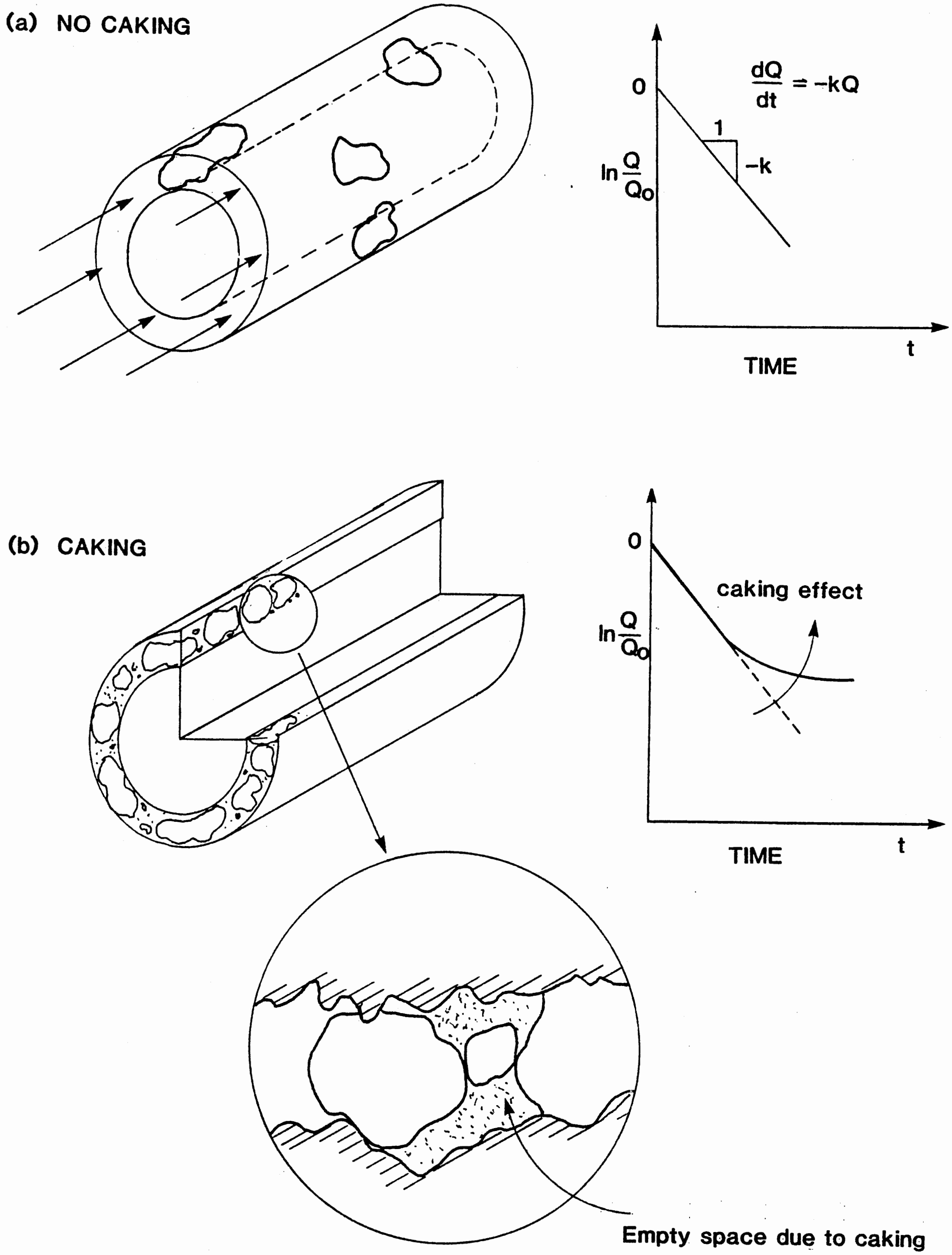


Figure. 35. Cake Formation in the Clearance of the Piston-Bore Assembly.

formation and silt. Hence, Eq. 59 is no longer valid and thus becomes:

$$dQ / dt < -kQ \tag{60}$$

Investigation of Background Count

The leakage flow test of the No. 3 bore and No. 19 piston (with 6.38mm land length) combination gives no more than 17 drops (Fig. 31) when the fluid Dexron II contains 100 mg/L ACFTD full distribution. The total 17 drops or less is not a sufficient sample of fluid to dilute with the super clean fluid and to run through the particle analyzer. Since the minimum necessary sample fluid for particle analysis is about 60 cc, 17 drops only amounts to 0.5695 ml and must be diluted with super clean fluid that is nearly 700 times more in volume to obtain 60 cc. When the sample fluid is diluted 100 times, the particle count of the sample fluid becomes so diluted that it may become impossible to conduct a particle count analysis.

To assess the silt Beta of a piston-bore assembly, the instruments and processes to identify silt Beta had to be qualified (known to be able to achieve the desired task). The leakage flow was collected in a clean bottle. The bottle was cleansed so that when the fluid was added, there was no more than 1.5 particles greater that 10 micrometers per milliliter of super clean fluid. In addition, the super clean fluid itself had a cleanliness level of less than one particle per milliliter greater than five micrometers. The sample fluid in the clean bottle was later diluted to achieve sufficient fluid volume for the particle analyzer to analyze the particle size distribution.

The most critical process that might destroy the particle size distribution information is this dilution process. Thus, careful attention was paid in order to statistically estimate how much particle size distribution information may have been corrupted through the dilution process. The identified level of corruption due to the dilution process is hereafter called "background count." The expected particle count result of the sampled fluid should be ten times or higher than the background count so that the sample fluid count will not be corrupted by the background.

The minimun sample fluid volume for the particle analyzer is 60 cc. Thus, the total sample fluid volume had to be high enough so that the added super clean fluid would not dilute excessively. Since the minimun concentration that was used in the silt Beta test was 25 mg/L, the nominal particle count of 25 mg/L ACFTD full distribution greater than 5 micrometers was 12917. If this were diluted x times, the diluted count would be:

$$\text{diluted count} = 12917 \,/\, x \qquad (61)$$

The background count should not exceed 10% of the diluted count; then,

$$\text{background count} < 0.1\ (\text{diluted count}) \qquad (62)$$

Combining Equations 61 and 62, Eq. 63 results in:

$$x < 1291.7 \,/\, \text{background count} \qquad (63)$$

So, as soon as the background count is available, the maximum allowable dilution factor x can be found.

Figure 36 shows the process employed to make the sample for the particle analyzer. The leakage volume was first poured into a pipet to measure the volume accurately. It took approximately 10 minutes to transfer most of the fluid in the 15 cc bottle to the pipet. Then, the fluid was poured into a 125 cc clean bottle. Step 3 and 4 were repeated at least twice to transfer particles from the pipet to the 125 cc bottle. To transfer the remaining particles in the 15 cc bottle to the 125 cc bottle, a known super clean fluid measured by a graduated cylinder was poured into the 15 cc bottle. Then, the 15 cc bottle was shaken vigorously with its clean cap on, and the fluid was poured into the 125 cc bottle. These steps, 5, 6, 7, and 8, were repeated at least twice.

After each step was completed, the pipet, graduated cylinder, and rubber glove were immediately flushed with ether to clean off any residual solid particles on their surface.

As Figure 36 shows, steps 1, 4, 6, and 7 may introduce additional particles because the leakage comes in contact with surfaces that may have foreign particles. As a result, due to the intrusion of foreign particles, it was important to know how many particles there would be in the 125 cc sample bottle after the dilution process.

By following all of the steps in Fig. 36 with super clean fluid in the 15 cc sample bottle instead of the sample fluid, the background count in the 125 cc bottle was identified. The results of 54 analyses are plotted in Fig. 37. As Figure 37 shows, the number of particles per

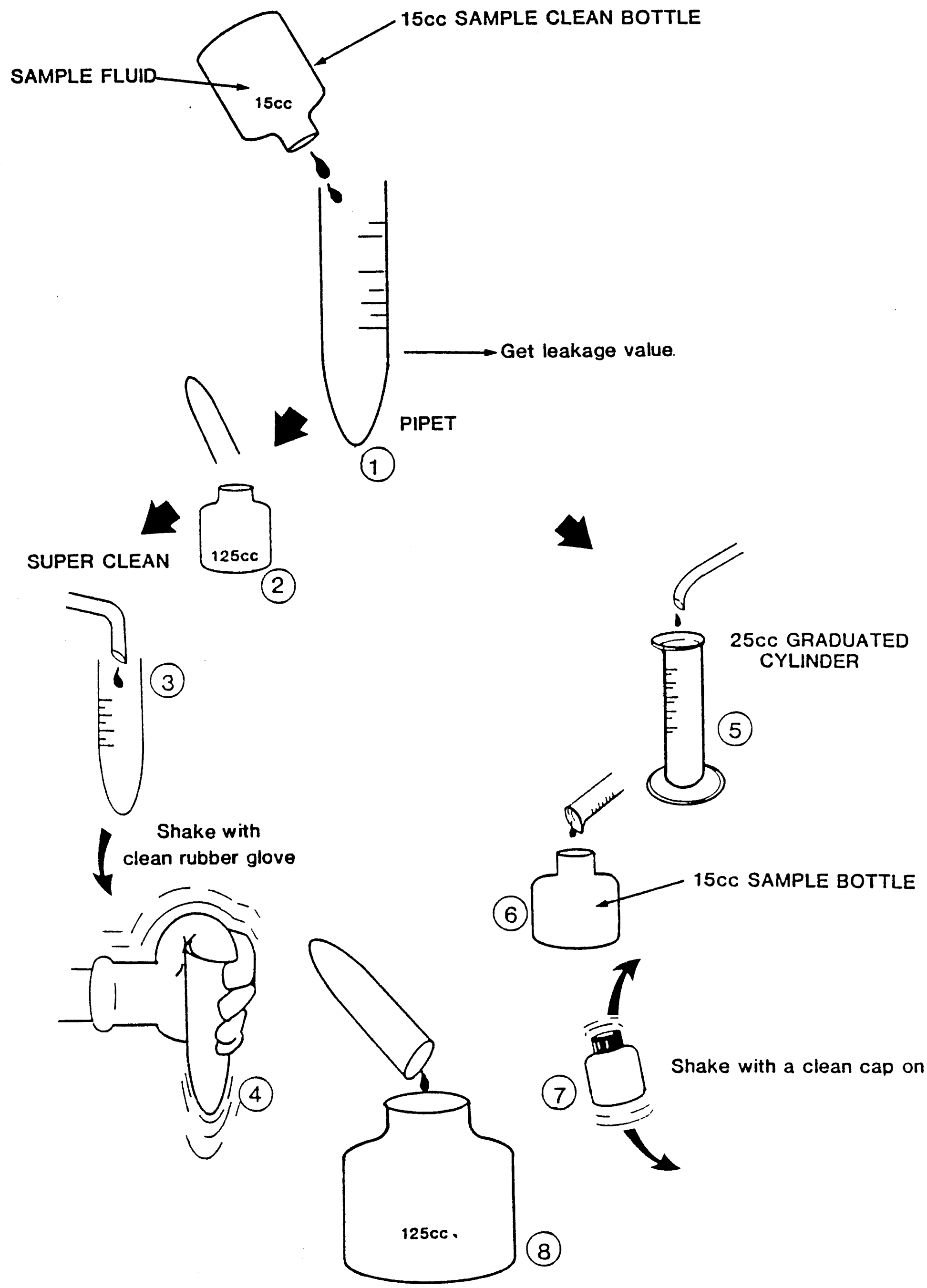


Figure 36. Dilution Process.

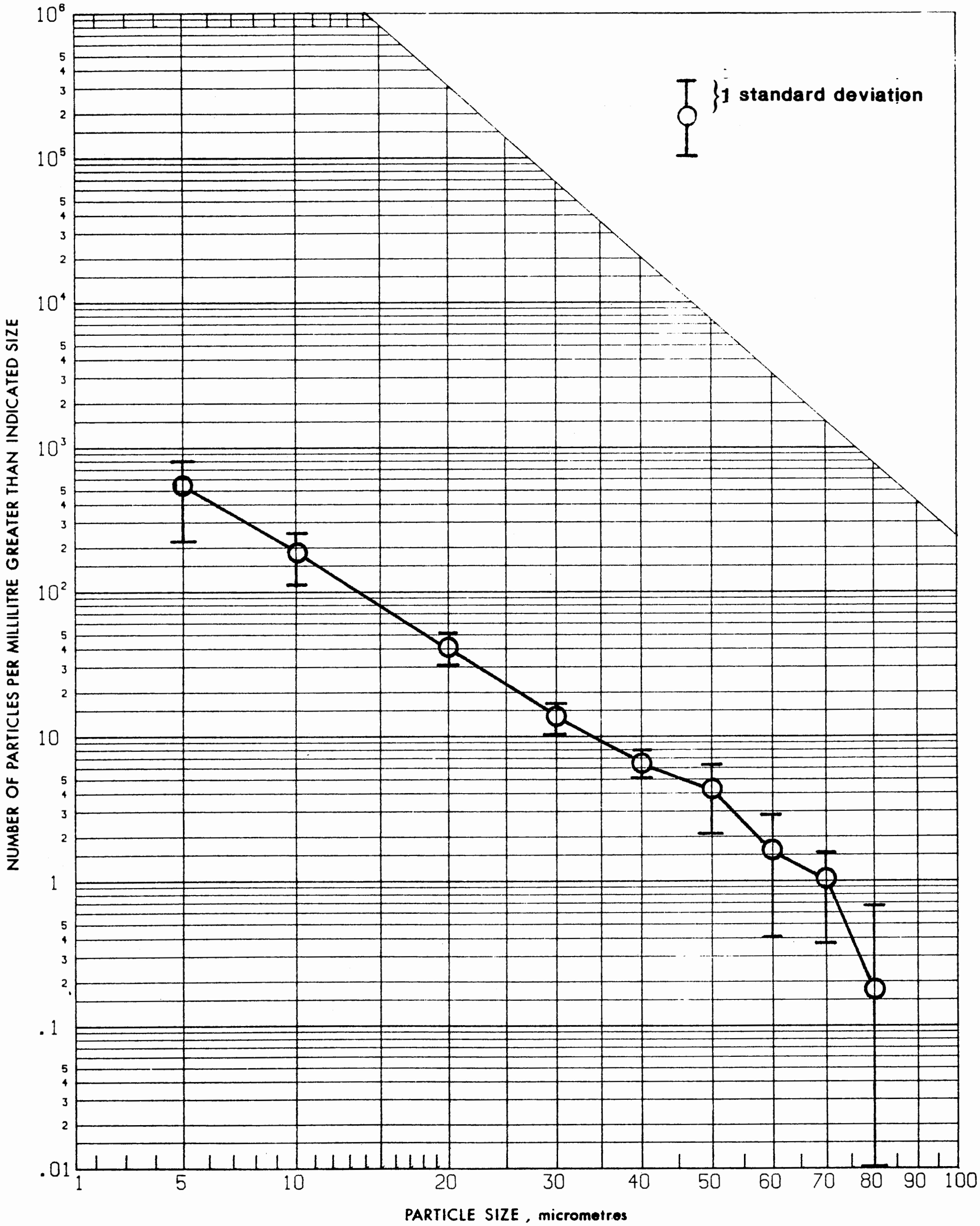


Figure 37. Dilution Process of Background Count (I).

milliliter greater than 5 micrometers is 563. Using Eq 63, the maximum allowable dilution factor is 2.29. Figure 38 shows just how important it is to maintain a high level of cleanliness throughout the dilution process. The data specified by the triangle were obtained when bare fingers were used in step 4 of Fig. 36. The background was tremendously corrupted.

The Hiac-Royco Particle Size Analyzer Model : PC-320 with the Hiac Particle Counting Sensor G3-90-68sp were used to analyze the upstream and downstream particle size distribution (Fig. 39). The counter and sensors were calibrated per ISO standard 4402. The sample bottles were shaken vigorously by a paint shaker for 15 minutes. Then, they were placed in an ultrasonic bath to eliminate the bubbles in the fluid. After that, the bottle cap was taken off and the bottle was placed in a vaccum chamber to further degas the fluid until no significant bubbles remained. Finally, the sample bottle was placed in the chamber shown in Fig. 39 to conduct particle counts. An air pump pressurized the chamber causing the sample fluid to travel up the channel which passes through the particle counting sensor. When the fluid level in the metering tube hit the lower level sensor, the particle counter began counting until the fluid level reached the upper level sensor. By positioning the level sensors so that the volume passing through the particle counter sensor was known, the number of particles were counted in the counter and were then described in a per milliliter basis. From the information obtained from the calibration method mentioned earlier, the counter was set to measure the number of particles at the desired size or greater.

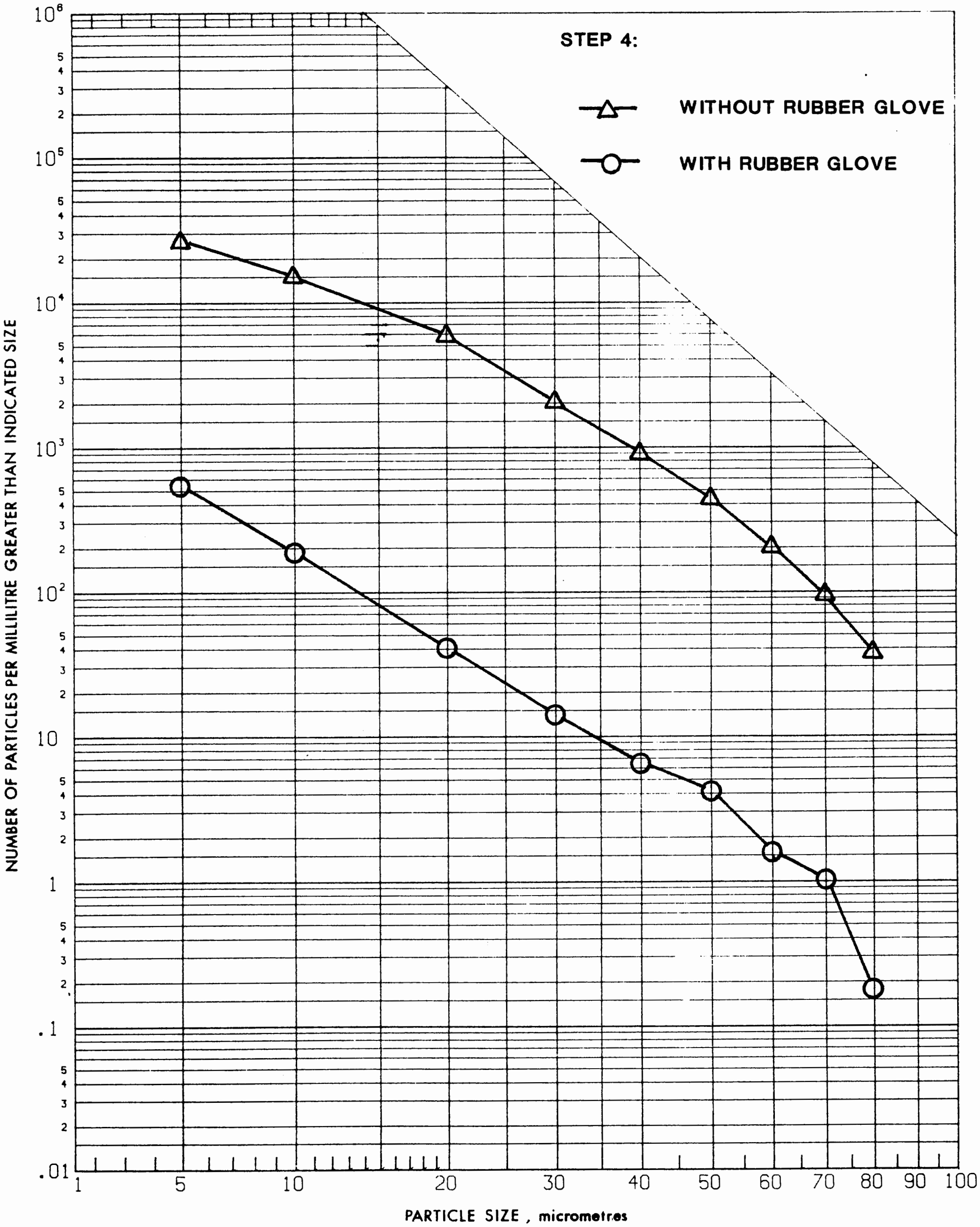


Figure 38. Dilution Process of Background Count (II).

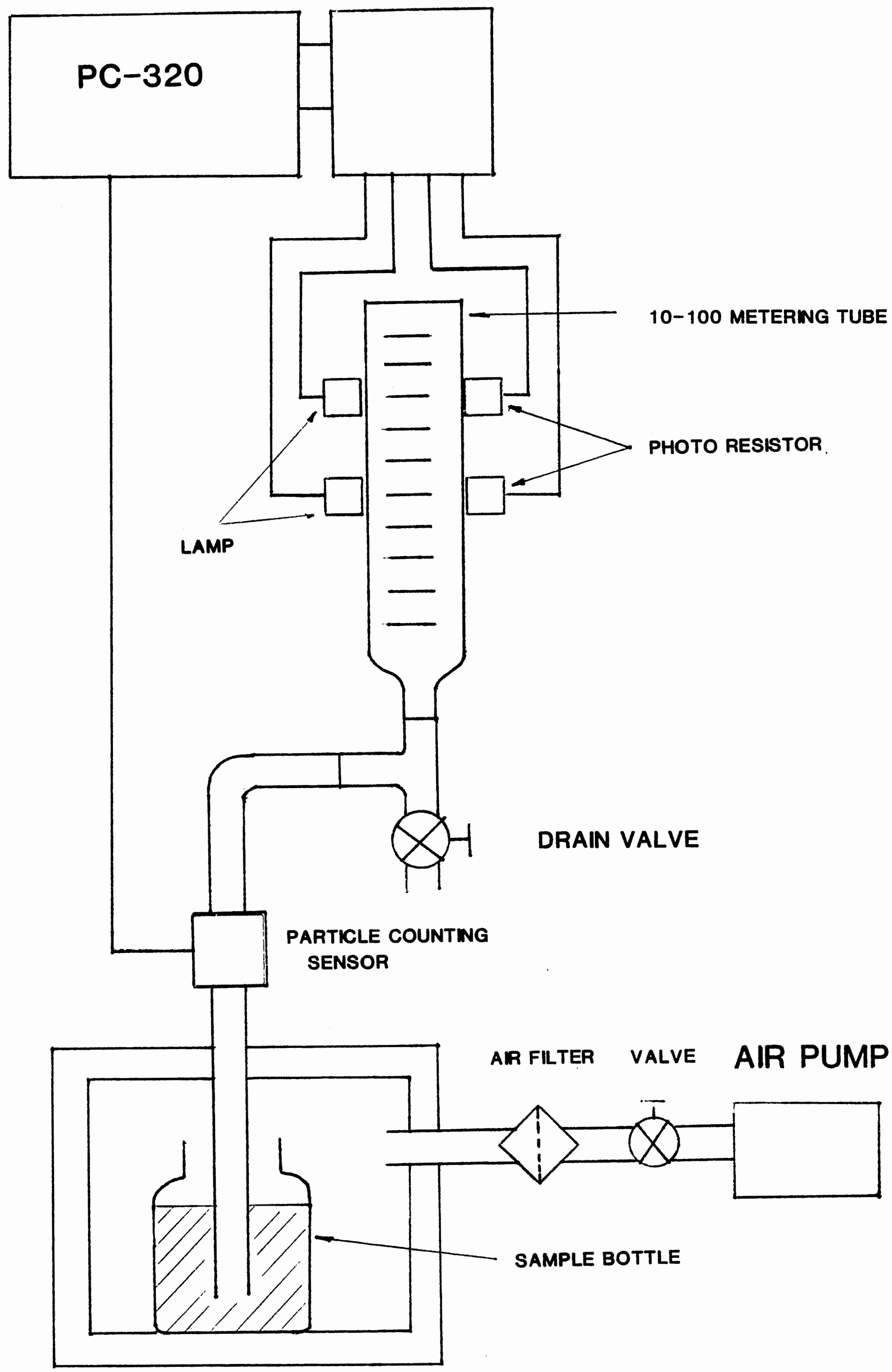


Figure 39. Particle Count Analysis Facility.

Silt Beta Experiment

The leakage was collected in clean bottles while the system pressure was set at 1000 psi. The sample position of the leakage is called "downstream" and the collected leakage itself is the downstream sample (see Fig. 2). The upstream sample was collected from the main test circiutry. In order to collect a sufficient amount of fluid for the particle analysis, the leakage test was repeated several times. Prior to each leakage test, the clearance was first flushed by the test fluid circulating through the hydraulic circuitry. This was done simply by pulling the piston out of the bore. Then, the test housing was isolated from the rest of the test circuitry by closing both of the ball valves located right next to the test housing. After that, the piston was positioned back inside the bore to establish the test clearance, as in Fig. 25.

To flush residual particles in the bore surface and the test housing, the nozzle of a plastic bottle containing ether was inserted as shown in Fig. 40. This process was repeated several times to ensure that the flushing was performed properly.

The downstream and upstream samples were collected as Table V shows. The samples were collected within a 10 minute period, which was sufficient to determine the last drop the system could collect. The detailed results of Test Code 2I are shown in Fig. 41. Test Code 1I contains the upstream sample and 2I contains the downstream sample particle size distribution, as Fig. 41 shows. The difference between the

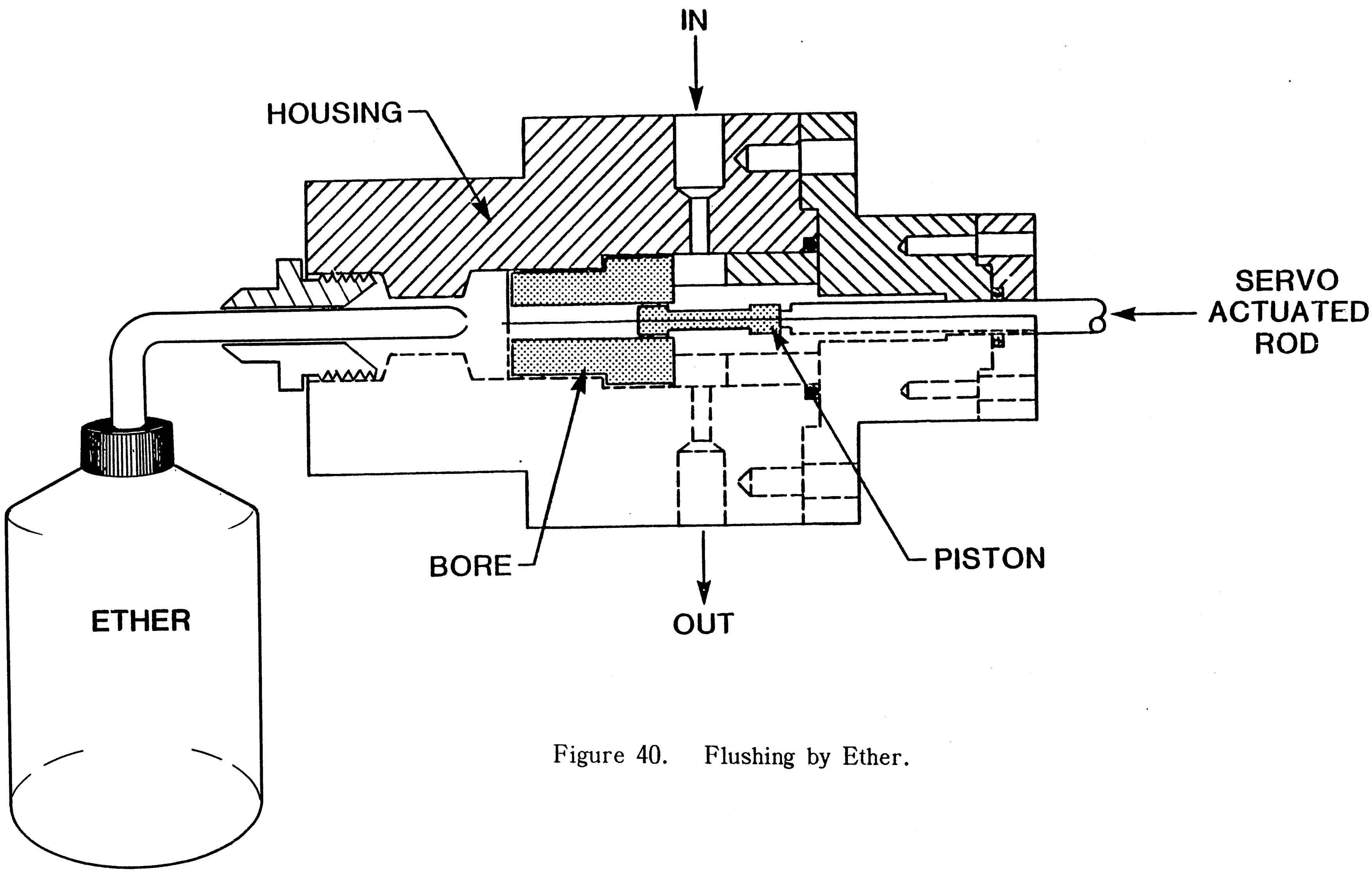


Figure 40. Flushing by Ether.

TABLE V

SILT BETA ASSESSMENT TEST

<TEST CONDITIONS>

BORE NO. 3
CONTAMINANT: ACFTD FULL DISTRIBUTION
CONCENTRATION: 25 mg/L
PRESSURE DIF.: 1000 psi
TEST FLUID: Dexron II at 60 degrees celsius

NOMINAL CLEARANCE SIZE(m)	SPOOL NO.	LAND LENGTH (mm)	TEST CODE	SAMPLE POSITION
5	19	6.38	1I 2I 3I	UPSTREAM DOWNSTREAM DOWNSTREAM
5	19	3.19	1J 2J 3J	UPSTREAM DOWNSTREAM DOWNSTREAM
10	23	6.38	1K 3K	UPSTREAM DOWNSTREAM
10	23	3.19	1L 3L	UPSTREAM DOWNSTREAM
15	25	6.38	1G 2G	UPSTREAM DOWNSTREAM
15	25	3.19	1H 2H 4H	UPSTREAM DOWNSTREAM DOWNSTREAM

***** 1 & 2H TEST RESULT ******

Bore No.3 - Piston No.19, Land Length (6.38 mm)
5 micrometers nominal clearance

Size	Distribution of 25 mg/L	Upstream	Downstream	BETA
5	12917.2	11008.8	7279.271	1.512349
10	3597.918	2878.8	950.8991	3.027451
20	637.1161	487.1	113.8693	4.277713
30	187.6002	132.4	23.8326	5.555416
40	71.72498	48.5	5.0554	9.593702
50	32.244	22.4	2.8888	7.754084
60	16.20372	8.729999	.238326	36.63049
70	8.837601	3.81	.238326	15.9865
80	5.133417	2.09	.01	****

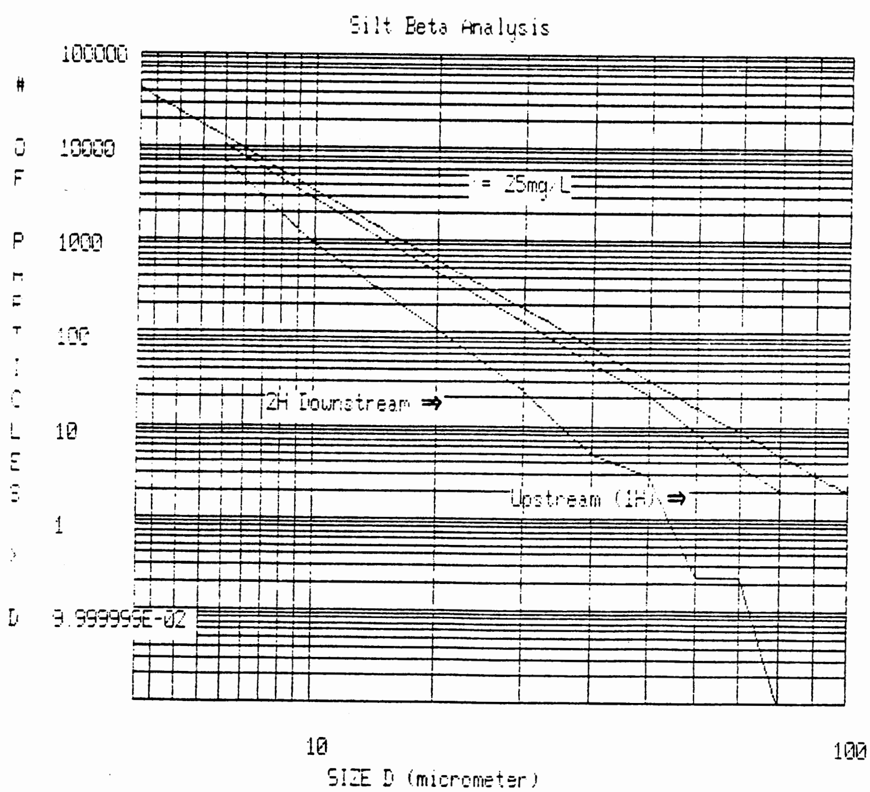


Figure 41. Particle Size Distribution Analysis.

upstream and downstream clearly demonstrates that the piston-bore assembly (bore No. 3 and piston No. 19 with 6.19 mm land) filtered particles as the leakage passed through the clearance of the piston-bore assembly.

The combination of bore No. 3 and piston No. 19 gives the nominal clearance size of 5 micrometers. Therefore, when the piston is fully eccentric with the bore, the largest clearance is 10 micrometers. Since the average length to width ratio of ACFTD full distribution is 1.49/1, it is possible that a particle with a length of 1.49 micrometers and a width of 10 micrometers can enter the clearance, as shown in Fig. 42. Moreover, the length to width ratio 1.49 is the average ratio, and careful microscopic observation of ACFTD shows that the length to width ratio of ACFTD does not exceed 3 to 1. Therefore, there should be no particles greater than 30 micrometers at downstream for the combination of the No. 3 bore and No. 19 piston (having a nominal clearance of 5 micrometers).

Though Figure 41 shows the presence of particles above 30 micrometers at downstream, as the background count of Fig. 37 presents, the downstream counts are corrupted by the background counts. The downstream particle count at 5 micrometers is the most accurate value because the average background count at 5 micrometers only amounts to 7.7% of the downstream count. However, the average background count comes close to 19% of the downstream particle count at 10 micrometers; thus, the downstream count at 10 micrometers is less accurate than the particle count at 5 micrometers. The downstream particle count at larger than 10

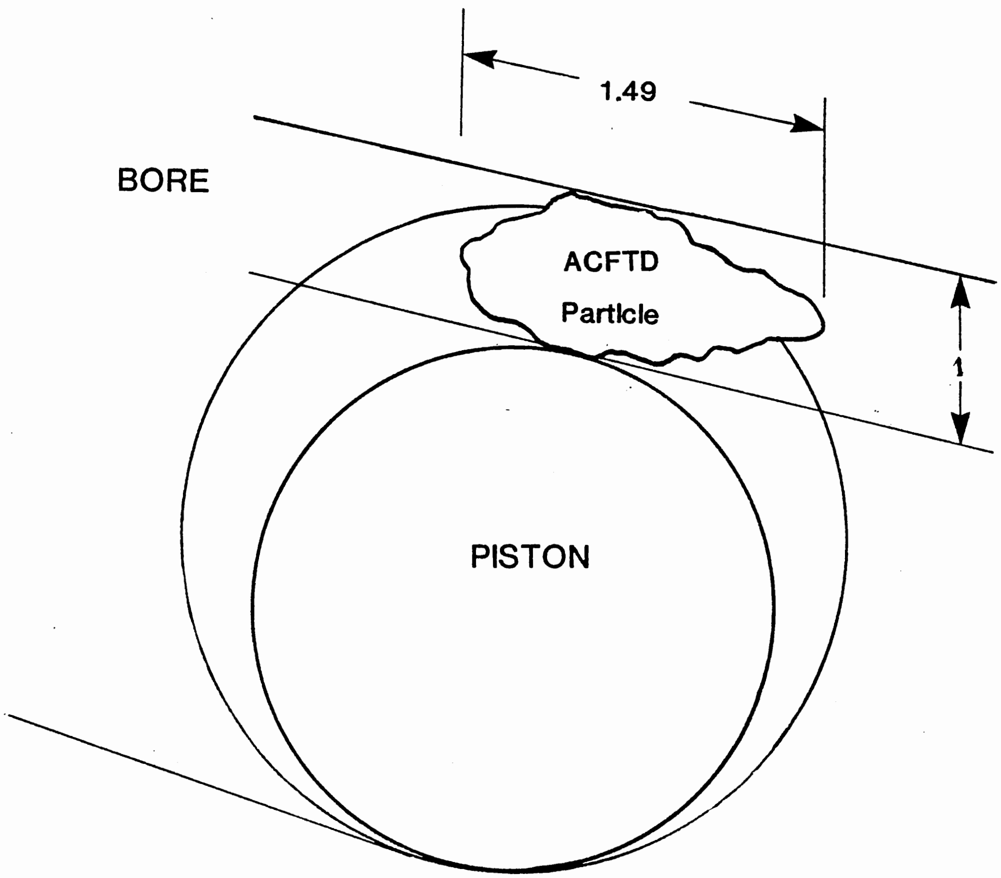


Figure 42. Maximum Entrained Particle Size.

micrometers becomes less accurate as the particle size becomes larger. Therefore, 5 m and 10 m are the two datapoints with the least error. Based on these datapoints, the following silt Beta model is developed.

Silt Beta Model

Tessman and Fitch [23] presented the filtration Beta of hydraulic filter elements as a function of particle size (Fig. 43). This relation can be closely approximated by:

$$\ln \beta = M + R \ln^2 D \tag{64}$$

where

M & R = constants

β = filtration Beta

D = particle size

Hydraulic filter elements have a certain pore size distribution, and the maximum size at which particles can pass through is not definite, whereas a piston-bore assembly can have the maximum limit because of its known geometrical configuration. Therefore, the upper limit size constraint was added in Eq. 64, and the silt Beta model for a piston-bore assembly became:

$$\ln \beta = M + R \ln^2 D - U \ln (\ln^2 C - \ln^2 D) \tag{65}$$

where

U = constant

C = upper limit size in micrometers

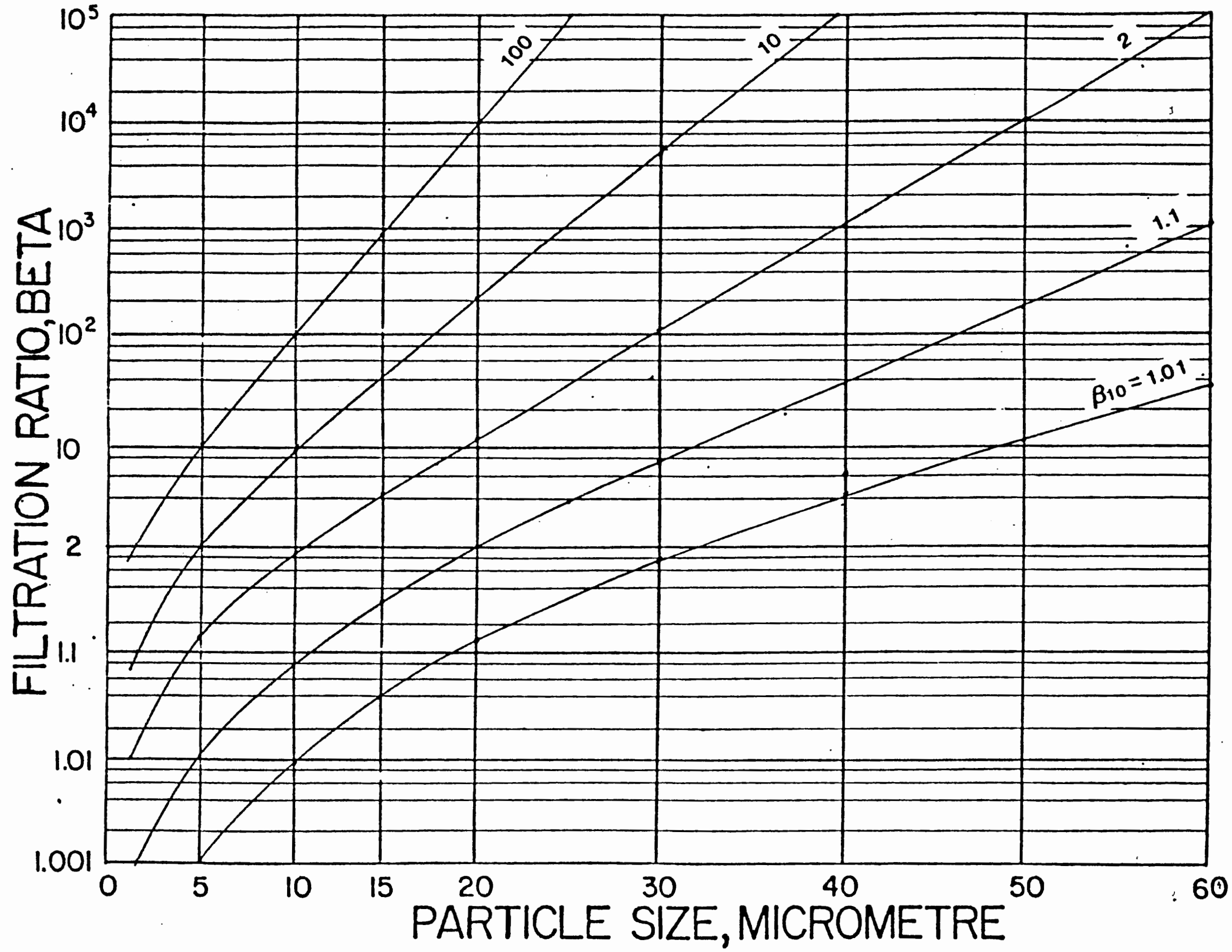


Figure 43. Log Normal Filter Model.

Three experimental data points are needed to determine three constants in Eq. 65. However, as found in previous sections, only the data at 5 micrometers and 10 micrometers can maintain the average error percentage less than 20 percent. As a consequence, further data was taken from the following assumption. If the number of particles filtered per milliliter greater than 5 micrometers is described as:

$$\Delta_5 = Nu_5 - Nd_5 \tag{66}$$

then the particles filtered at greater than 1 micrometer is:

$$\Delta_1 = Nu_1 - Nd_1 \tag{67}$$

and the postulate is:

$$\Delta_1 \geq \Delta_5 \tag{68}$$

This assumption is valid as long as there is no significant desorption of once filtered particles in the clearance. Figure 44 describes why. When the upstream and downstream particle size distribution is described in the graph as frequency versus size, it will be as shown in Fig. 44. What is described by Eq 66 is the area between the upstream and downstream greater than 5 micrometers; the area between the upstream and downstream greater than 1 micrometer is described by Eq 67. As long as there is no desorption of silted particles, the downstream curve will not exceed the upstream, and thus Eq. 68 holds true. Combining Equations 66 through 68,

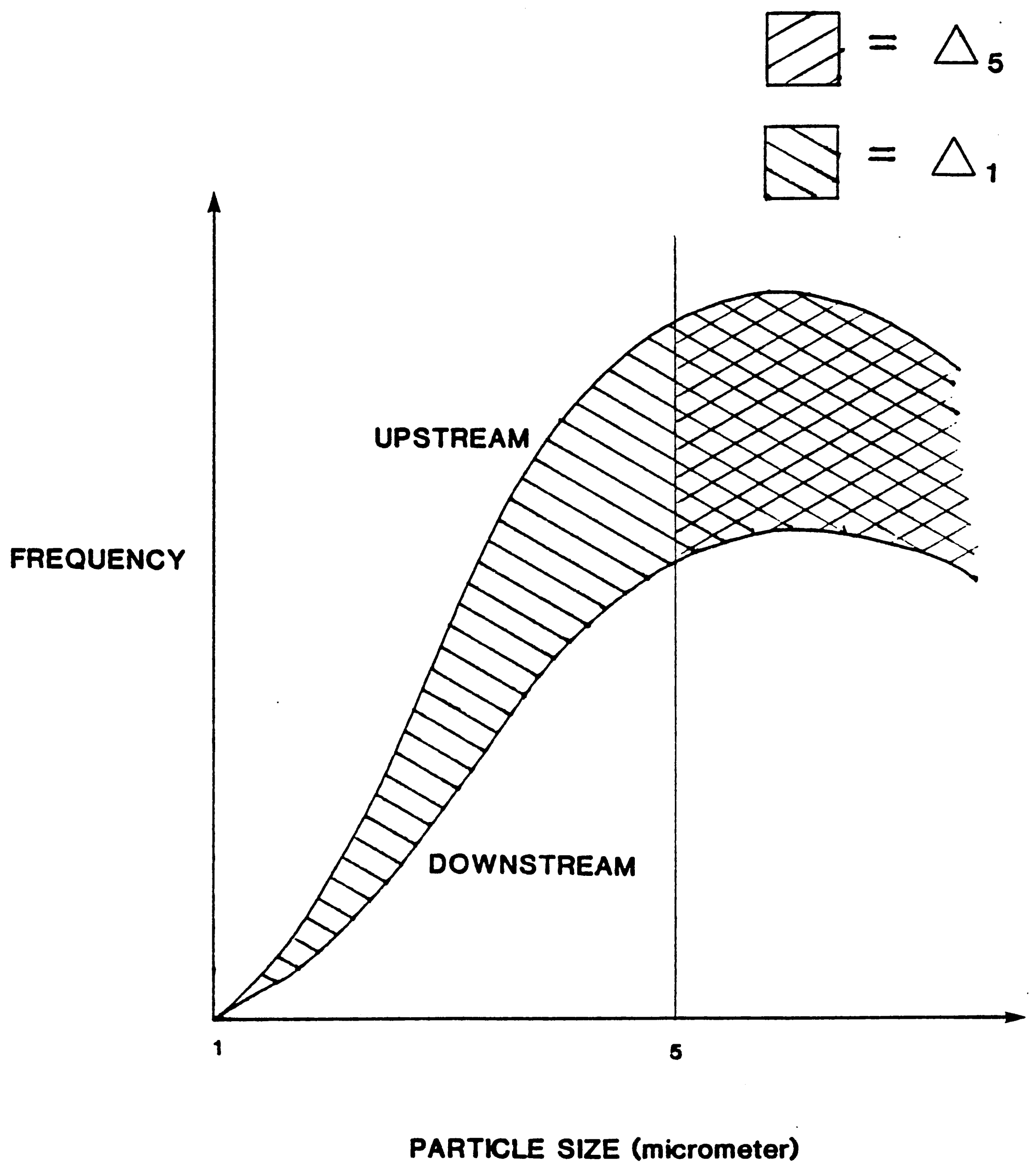


Figure 44. Frequency Versus Particle Size.

Eq. 69 results in:

$$Nu_1 \left(1 - \frac{Nd_1}{Nu_1}\right) \geq Nu_5 \left(1 - \frac{Nd_5}{Nu_5}\right) \tag{69}$$

Using the definition of Beta,

$$\beta = Nd \;/\; Nu \qquad \beta = \frac{N_u}{N_d} \tag{70}$$

Equation 69 becomes:

$$\frac{Nu_1}{Nd_{1/5}} \left(1 - \frac{1}{\beta_1}\right) \geq \left(1 - \frac{1}{\beta_5}\right) \tag{71}$$

Since the upstream is ACFTD full distribution, Nu1 divided by Nu5 is always 1.259, as follows:

$$T = \frac{Nu_1}{Nu_5} = e^{Bu\,(\ln^2(5) - \ln^2(1))} = 1.259 \tag{72}$$

Solving for β_1, Eq. 71 becomes:

$$\beta_1 \geq \frac{1}{1 - (1 - 1/\beta_5)(1/T)} \tag{73}$$

It is also known from Fig. 43 that:

$$\beta_5 > \beta_1 \tag{74}$$

Figure 45 shows the boundary of values that β_1 can take, provided that there is no desorption of silted particles.

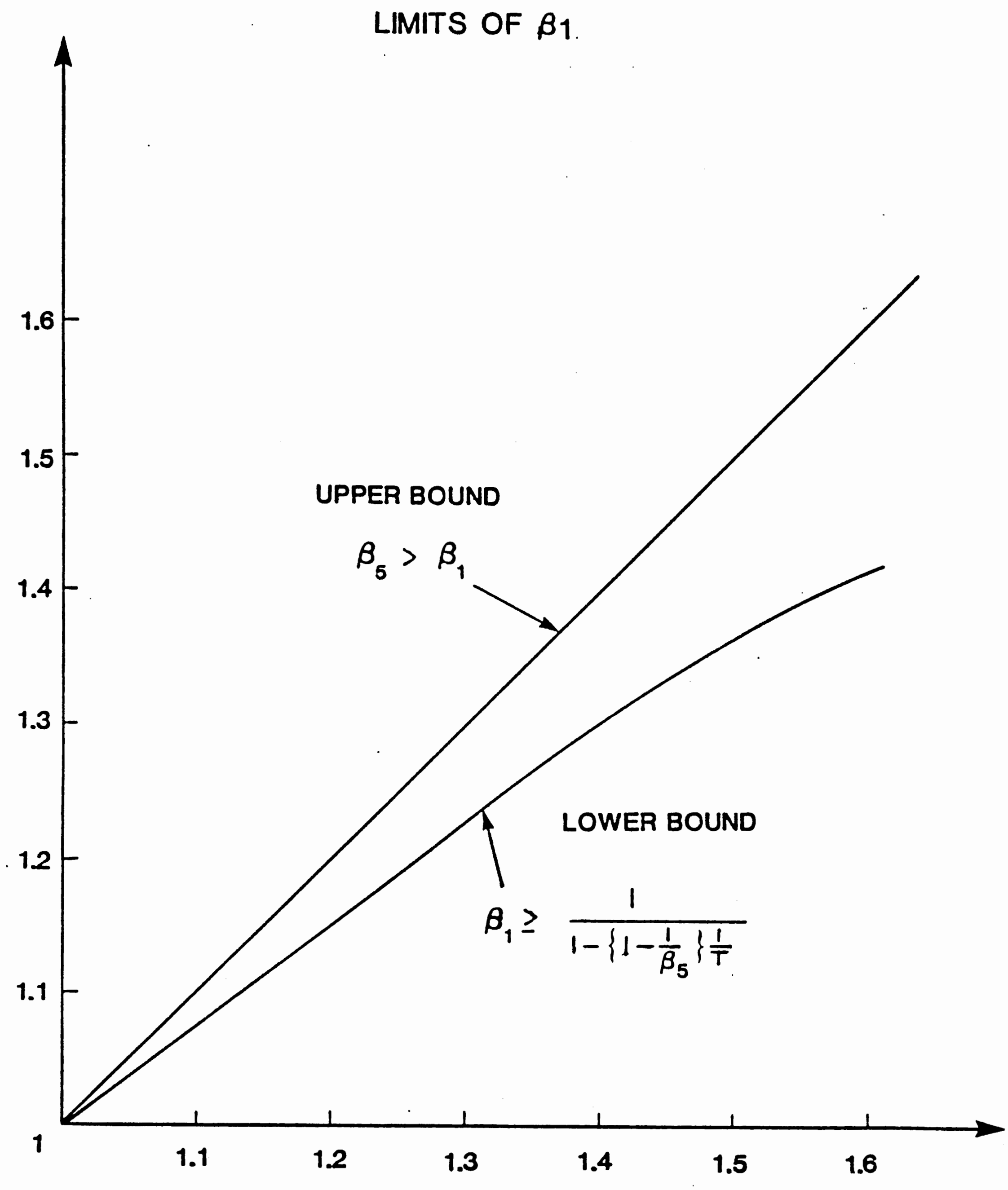


Figure 45. Limits of β_1.

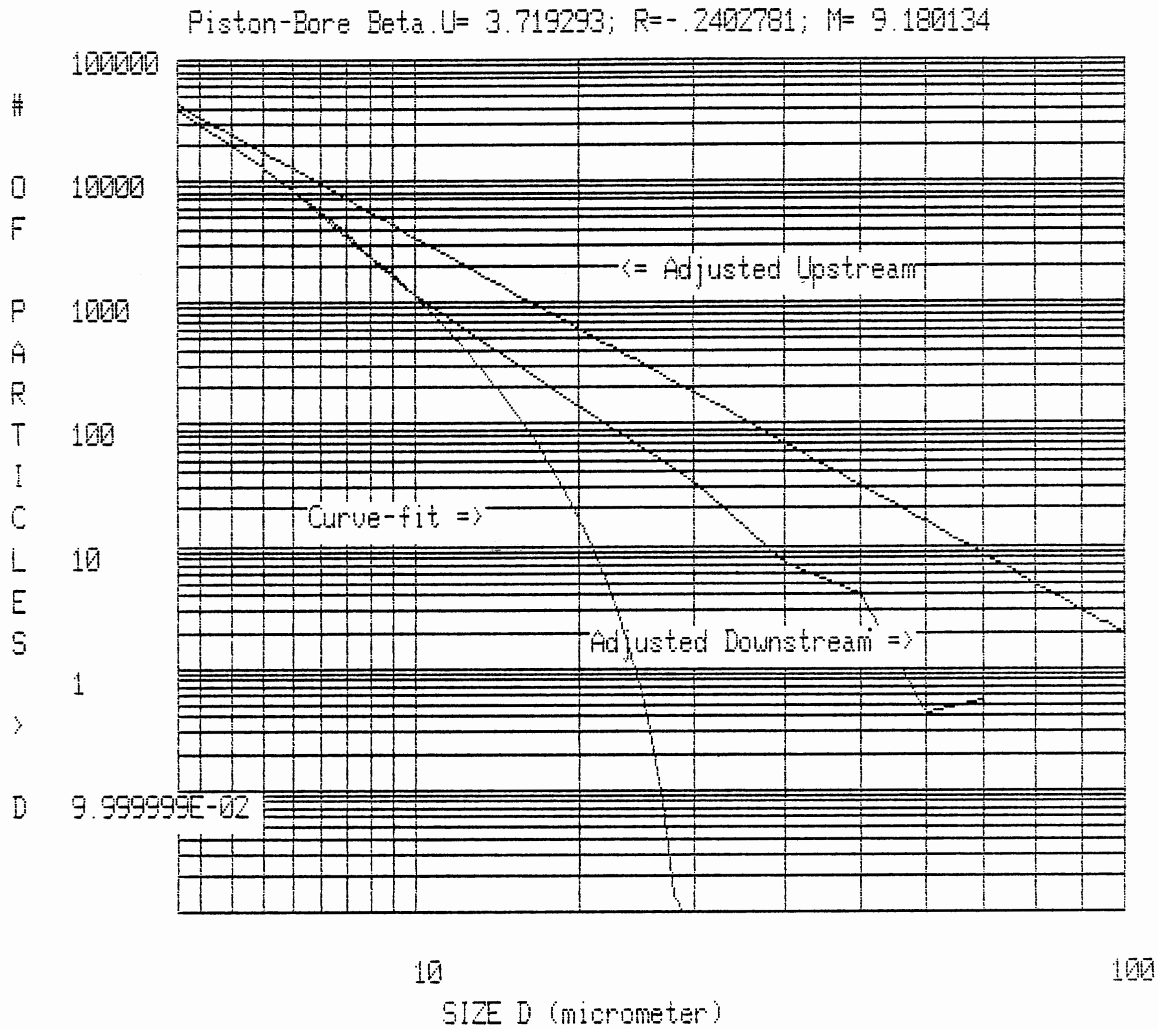
For the third data to find the constants of Eq. 65, Eq. 75 was chosen because this silt Beta value:

$$\beta_1 = \frac{1}{1 - (1 - 1 / \beta_5)(1 / T)} \qquad (75)$$

is the most conservative value, and the error involved with this choice is a maximum of 11 percent at $\beta_5 = 1.6$, and at the smaller value of β_5, the error is less. Keeping the same silt Beta value at each particle size shown in Fig. 41, the downstream was redrawn in relation to the theoretical 25 mg/L ACFTD full distribution as the upstream particle size distribution in Fig. 46. Figure 46 demonstrates a good agreement between the silt model curve of the downstream distribution and the experimental data for the 1 to 10 micrometers range; it also demonstrates a diverging deviation after 10 micrometers. As a consequence, this silt model curve satisfies two important conditions:

1. The model fits well between 1 to 10 micrometers where the actual downstream data is the most accurate.
2. The curve satisfies the maximum particle size at which no particle should be found in the downstream.

The silt Beta test data analyses for 3I through 4H shown in Table V are presented in Appendix B. The silt models found from the silt Beta test are plotted in Fig. 47. In the figure, hydraulic filter Beta curves are also superimposed. This figure clearly demonstrates that the smaller the nominal clearance is, the higher the silt Beta is. Moreover, the silt model tends to have better Beta values than the hydraulic filter



Size micron	Upstream	Downstream	Beta
1	43797.5	40067.96	1.09308
5	12917.2	8541.142	1.51235
10	3597.918	1188.432	3.02745
15	1380.819	170.7609	8.08627
20	637.1161	18.3983	34.62907
25	331.3792	.718631	461.1257
30	187.6002	.01	****
35	113.1623	.01	****
40	71.72498	.01	****
45	47.30807	.01	****
50	32.244	.01	****

Figure 46. Silt Beta Model Fit.

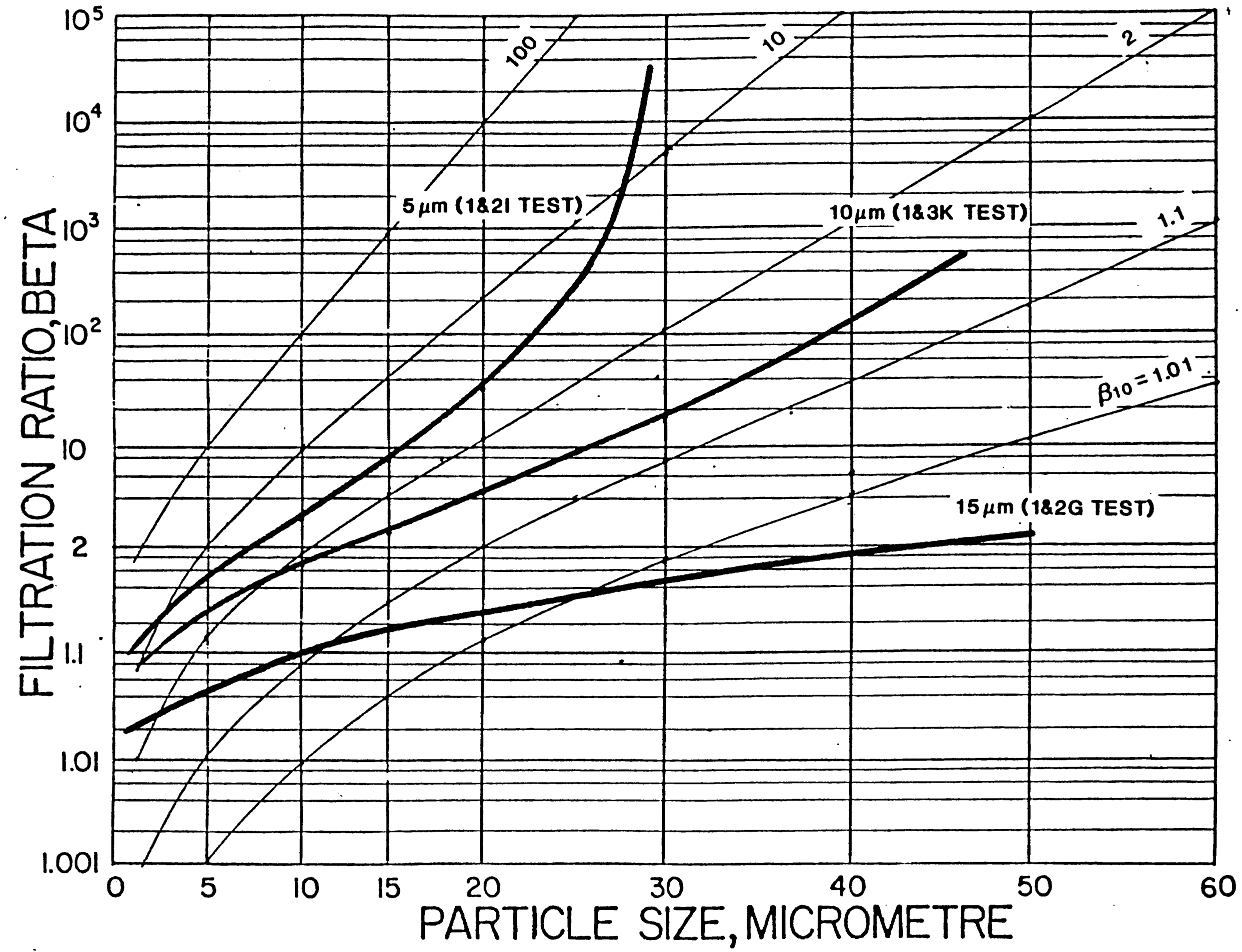


Figure 47. Piston-Bore Assembly Silt Beta (6.38mm Land Length).

Beta at below the 10 micrometers range. Since the silt model is derived from data at 1, 5, and 10 micrometers with their maximum error percentage at 3.3, 7.7, and 19 respectively, data in this range maintains high accuracy, and Beta values of piston-bore assemblies tend to maintain higher values than the hydraulic filter in this range. Furthermore, the Beta values in Fig. 47 were converted into the filter efficiency by the equation:

$$\eta = 1 - 1 / \beta \tag{76}$$

and plotted in Fig. 48. Hydraulic filter elements' filter efficiency [24] lines were also plotted to compare with the filter efficiency of piston-bore assemblies. This figure shows that hydraulic filter elements are straight lines, indicating that:

$$\eta = \int_{-\infty}^{Xe} \frac{1}{\sqrt{2\pi}} e^{-x^2/2} \, dx \tag{77}$$

and Xe is expressed as:

$$Xe = k_1 \log D + k_2 \tag{78}$$

where

k1 and k2 = constants

D = particle size in micrometers

On the other hand, the filteration efficiency is nonlinear, and thus the

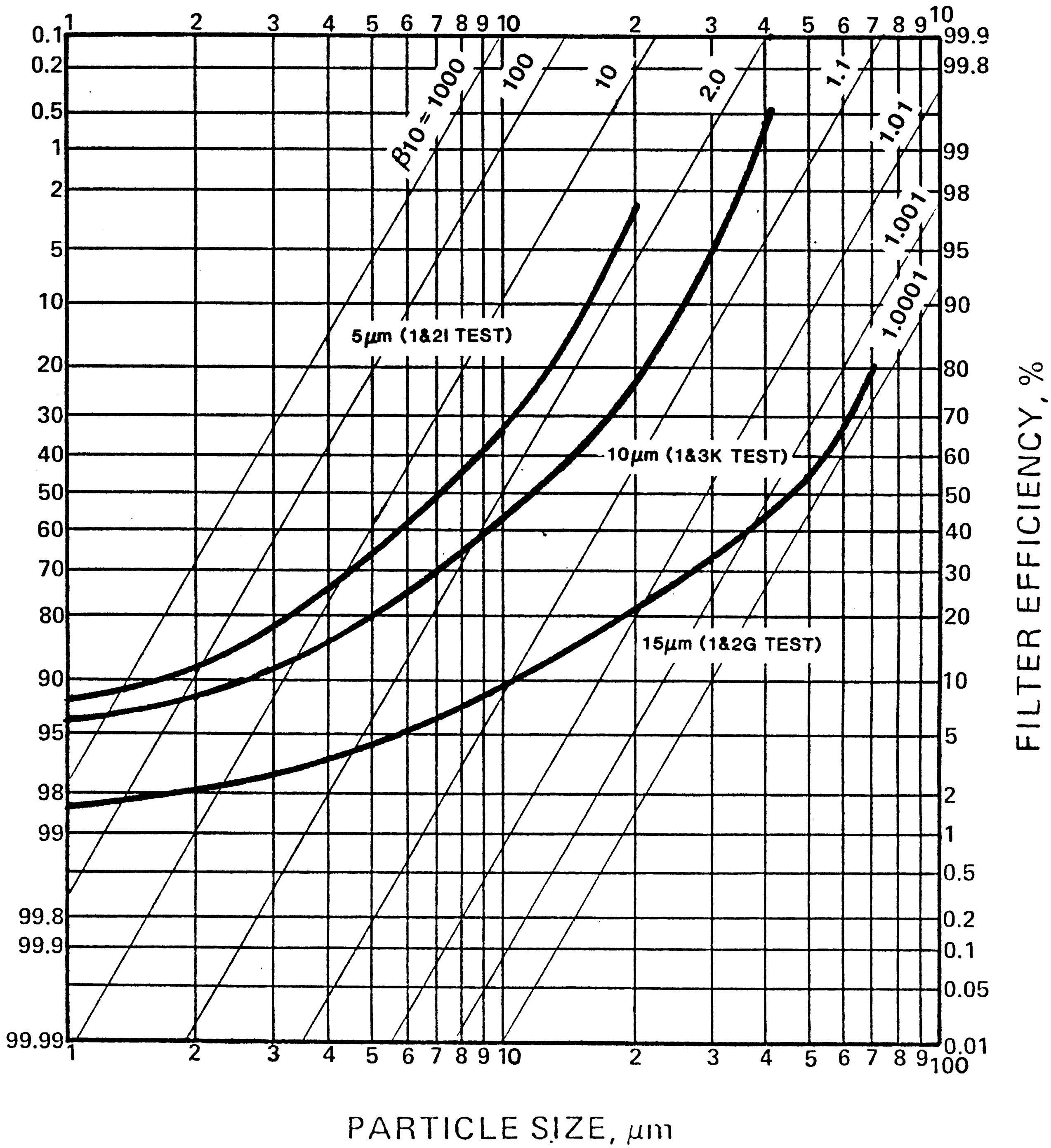


Figure 48. Piston-Bore Assembly Filtration Efficiency (6.38mm Land Length).

equation of Xe for the piston-bore assemblies needs to be changed to express nonlinearity.

The significance of this nonlinearity found among the piston-bore assemblies' filter efficiency is argued in the following. If there are two straight lines in the filter efficiency graph as shown in Fig. 49a, the separation efficiency at each particle size versus the logarithm of particle size displays two corresponding normal distributions (Fig. 49b). When the slope is steep, the separation efficiency shows narrow band normal distribution (the small standard deviation), and when the slope is moderate, the normal distribution has the large standard deviation. However, when the filter efficiency curve is nonlinear, such as Fig. 49c, the resulting separation efficiency is as illustrated in Fig. 49d. This indicates that over a period of time, the filtration efficiency progressively improved, and the assembly became capable of filtering finer particles. Figure 48 shows the progressive improvement by demonstrating the time average of the assembly filtration efficiency. This progressive improvement in filtration efficiency is observed among hydraulic filters and is called "caking."

The cake formation was previously analyzed based upon the leakage experiment, and here is further evidence that supports the presence of caking. As the cake formation builds up in the clearance of a piston-bore assembly, finer particles start silting in the tortuous path of the cake formation. As a result, the mean separation efficiency shifts to a smaller size. The separation efficiency curve of Fig. 48 is shown in Fig. 50 and clearly demonstrates the gradual slope of curves at a smaller

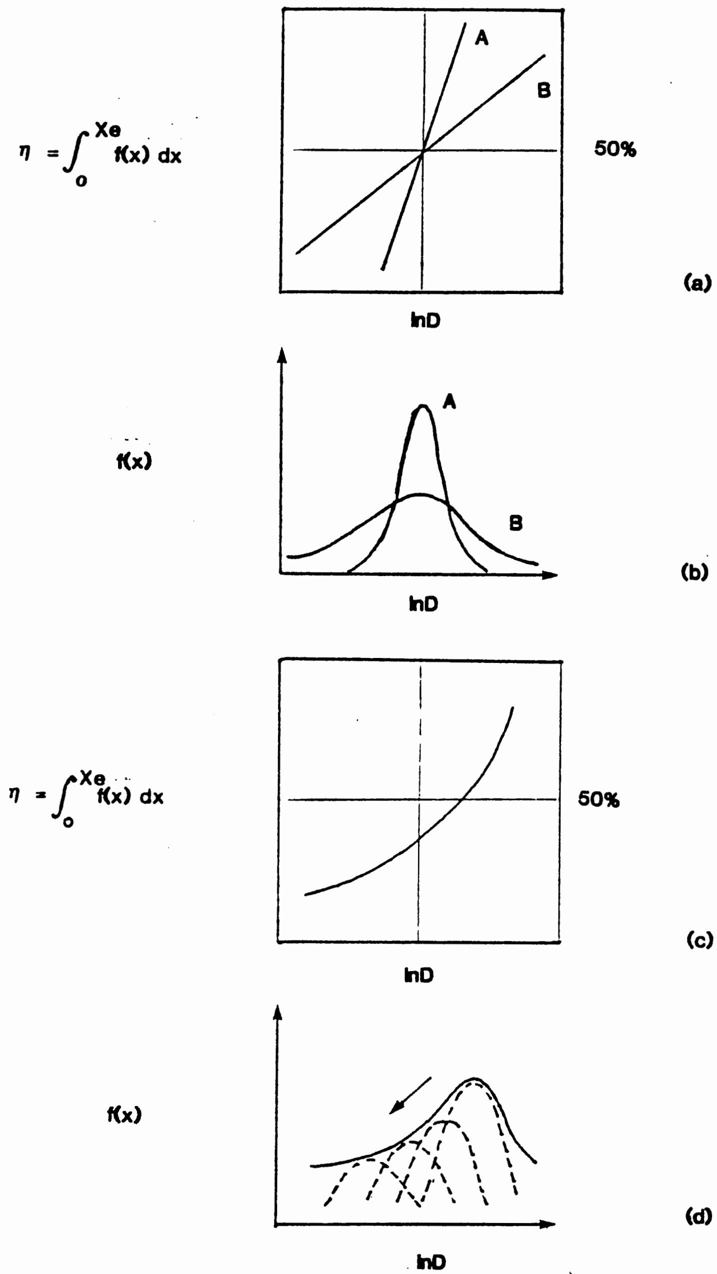


Figure 49. Analysis of Cake Formation.

particle size range than the mean size, and the steep slope of the curve at a larger particle size range than the mean size.

Furthermore, Fig. 48 shows that the particle size at which 50% filter efficiency of a piston-bore assembly locates is slighty bigger than the nominal clearance size of the assembly. For the case of 15 micrometers nominal clearance, the main filtration efficiency occurs at quite a larger size than 15 micrometers. This could be explained by the fact that as the clearance increases, the initial leakage flow and shear rate increase enough to prevent fine particles from silting in the clearance. Therefore, the overall filtration efficiency degrades.

The effect of land length on the assembly filtration efficiency was investigated. Half land length (3.19mm) tests are presented in Fig. 51 and 52. Figure 51 shows the filtration ratio versus particle size, and Fig. 52 shows filter efficiency versus particle size. The comparison between Fig. 48 and 52 exhibits that the assembly with 6.38mm land length has a higher filtration efficiency than the one with 3.19mm land length when both assemblies have the same clearance size.

Evaluation of the Simulation Program for the Silt Process Model

Though silt Beta values are now known (Fig. 47 and 52), these figures show the time average silt Beta over a period of time T. The silt Beta values at the beginning of the silt process (when solid particles are introduced) is slower than at the end of the silt process

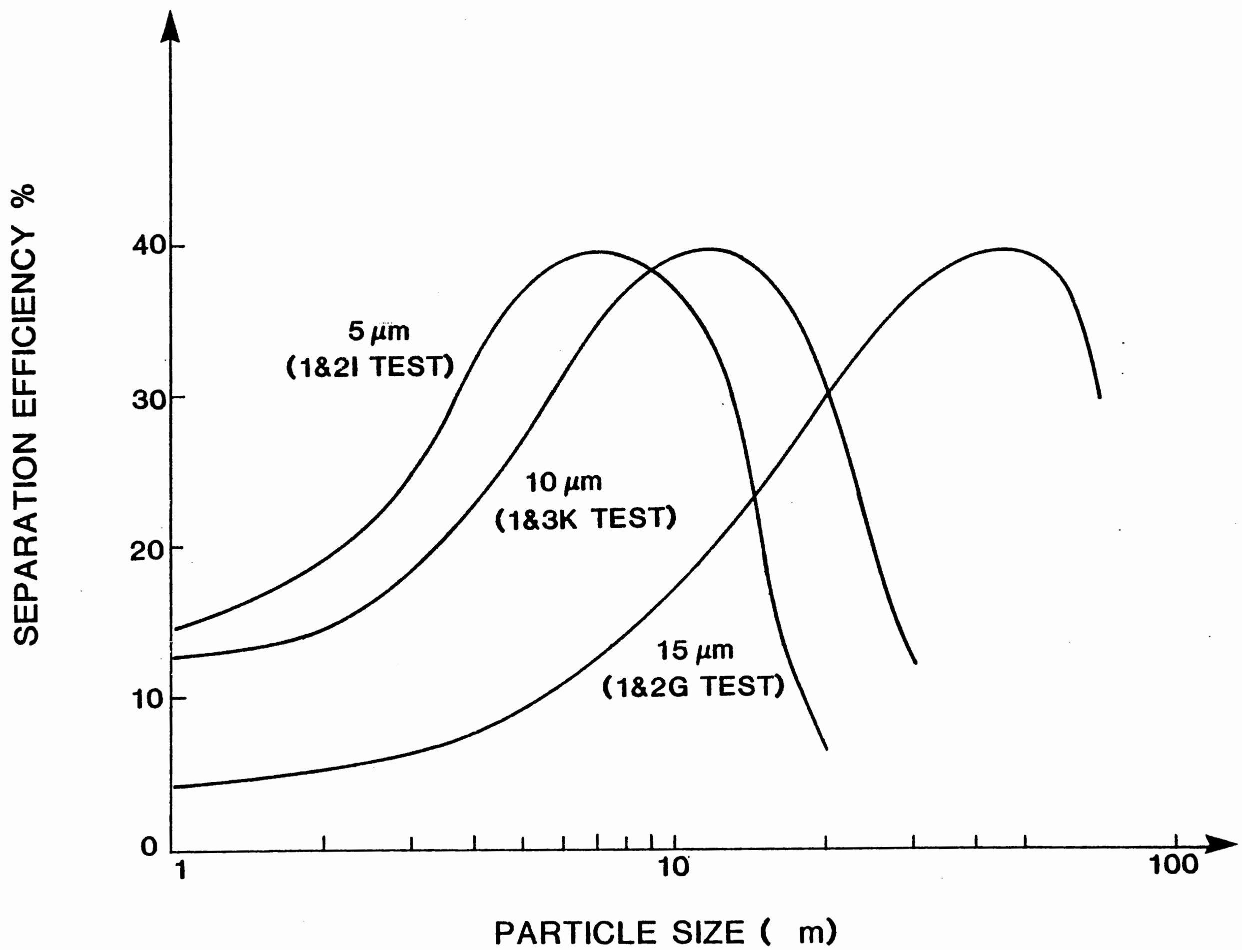


Figure 50. Piston-Bore Assembly Separation Efficiency.

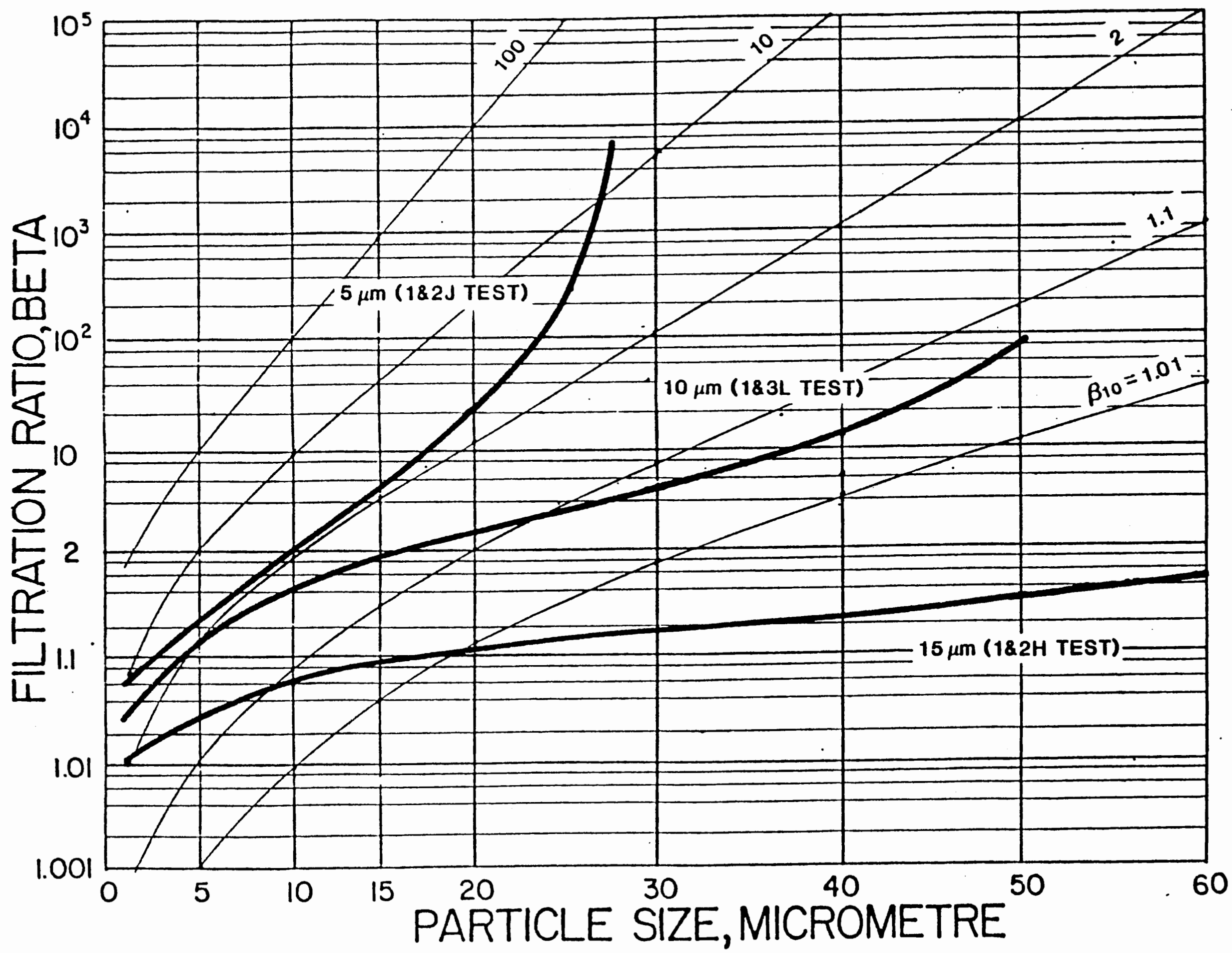


Figure 51. Piston-Bore Assembly Silt Beta (3.19mm Land Length).

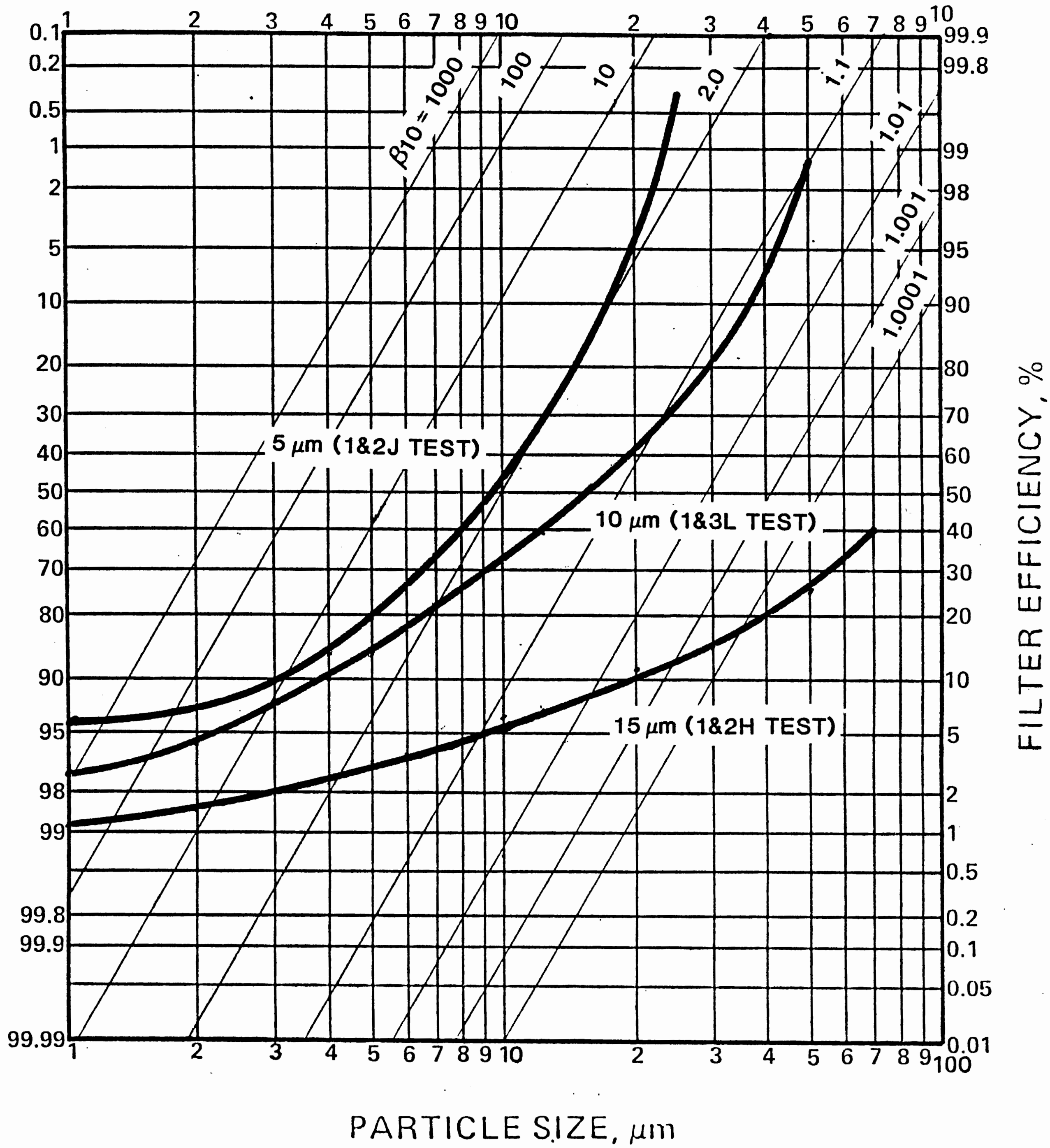


Figure 52. Piston-Bore Assembly Filtration Efficiency (3.19mm Land Length).

(when the leakage diminishes). Thus, the average silt Beta is represented by:

$$\bar{\beta}s = \left(\int_0^T \beta \, dt \right) / T \tag{79}$$

Equation 79 can also be shown as:

$$\bar{\beta}s = Nu / \overline{Nd} \tag{80}$$

where the upstream particle size distribution is not a function of time. It is a constant over a test period. Since we know Nu and Nd, the time average of the particle frequency distribution upstream f(D)u and downstream f(D)d can be calculated from Eq. 9. Thus, the particle frequency distribution of particles silted in the clearance of a piston-bore assembly is:

$$\overline{f(D)t} = f(D)u - \overline{f(D)d} \tag{81}$$

However, the unit associated with each term in Eq. 81 is the number of particles at size D per milliliter of the leakage. Hence, the total leakage volume has to be known to identify the particle frequency distribution after a test of duration T and is defined mathematically by:

$$F(D)t = \overline{f(D)t} \int_0^T Q \, dt \tag{82}$$

or

$$F(D)t = \overline{f(D)t} \; Vol \tag{83}$$

where

$$Vol = \int_0^T Q \, dt \tag{84}$$

In a similar fashion, Eq. 8 becomes:

$$\overline{Vt} = Vu - \overline{Vd} \tag{85}$$

where

$\overline{Vd}$ = the time average volume of the downstream particle size distribution per unit leakage volume.

$\overline{Vt}$ = the time average volume of the particles in the piston-bore clearance.

Thus, the total silt particle volume is:

$$S = \int_0^T \overline{Vt} \quad Q(t)dt \tag{86}$$

Substituting Eq. 84, this becomes:

$$S = \overline{Vt} \; Vol \tag{87}$$

Both terms on the right side of Eq. 87 can be experimentally measured to identify the silt volume S; however, the way this silt volume blocks the leakage flow through the clearance must be investigated. To do so, the silt models shown in Fig. 47 and Fig. 52 are implemented in the simulation program. For a piston-bore assembly that has a clearance size other than 5, 10, or 15 micrometers, and a land length other than 6.38mm and 3.19mm, the program has also been implemented to linearly interpolate and/or extrapolate (according to the requested range of silt Beta values) from data shown in Fig. 47 and Fig. 52.

The experimental results shown in Fig. 33 are used to investigate the performance of the simulation program. In Figure 53, the experimental results are compared with the simulation results. The curve marked A = 1 is the simulation result. The simulation program predicts far more leakage volume than is observed experimentally. This indicates that the silt particles in the clearance are piling up in such a way that the leakage flow through the clearance is reduced more than is predicted by the silt model process. To better define the effectiveness of this leakage flow reduction, Eq. 87 may be modified to have a constant multiplier, A:

$$S = \overline{Vt}\ Vol\ A \tag{88}$$

A was varied, as Fig. 53 shows, to define the effective particle reduction of the leakage flow. When the value of A is 40, the simulated leakage volume approaches the experimental results at short times.

The discrepancy between the silt process model and the experimental data is illustrated in Fig. 54. The silt process model hypothesizes that the reduction of the leakage flow can be calculated by uniformly decreasing the clearance size (thus decreasing the clearance space) according to the volume of particles silted in the clearance (Fig. 54a). However, if the theory is correct, the experimental results should show 40 times more silt volume than what is actually measured. This experimental result indicates that the silt volume does not uniformly spread and decrease the clearance size, but instead, tends to agglomerate

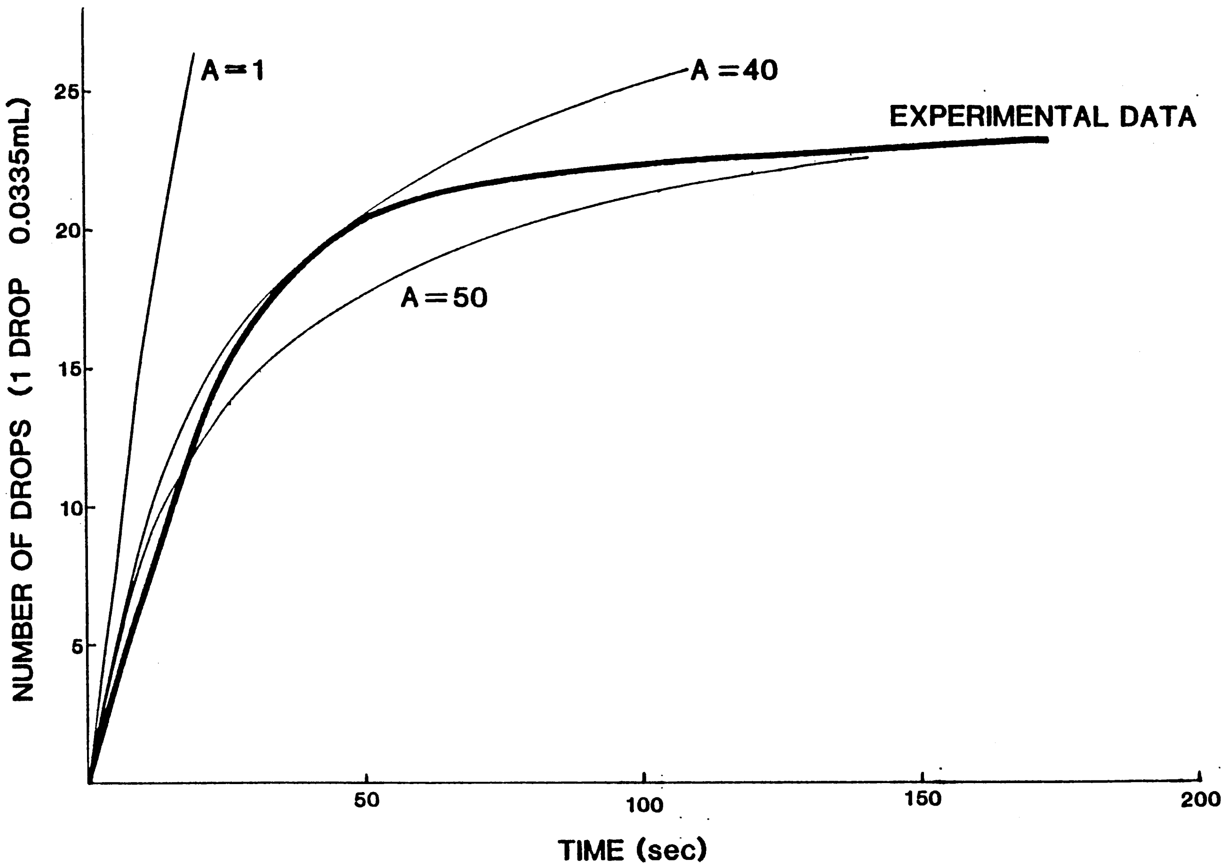


Figure 53. Constant Multiplier Analysis.

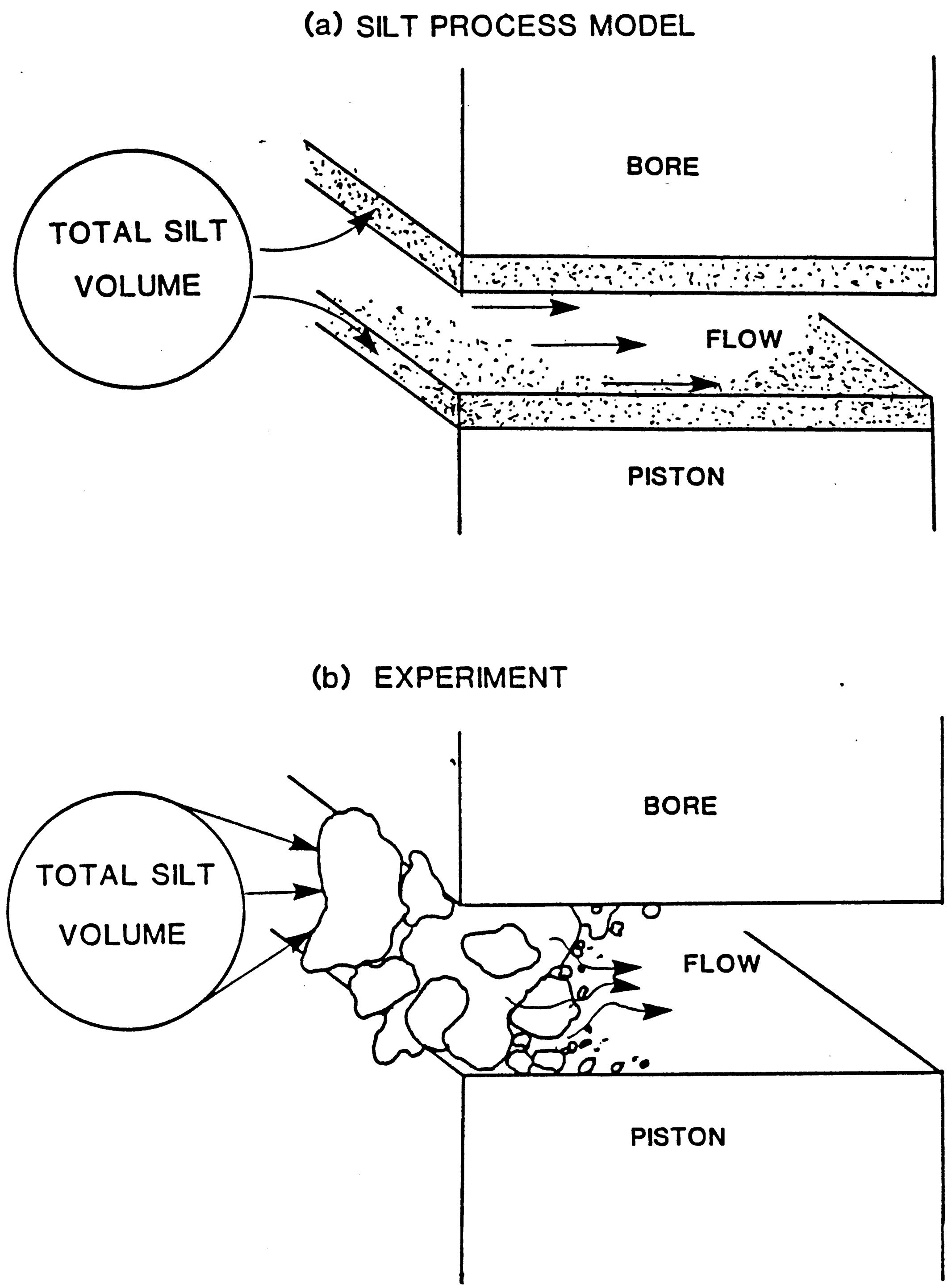


Figure 54. Silt Pattern (Theory and Practice).

somewhere in the clearance and effectively decrease the leakage.

Further analysis was performed to investigate the discrepancy. The rate of silt volume accumulation in the clearance, for the case of Fig. 33, was found to be 1.409E07 cubic micrometers per milliliter of the leakage volume passing through the clearance of the piston-bore assembly. From Fig. 33, there are no more than 24 drops for the piston-bore assembly under the specified test condition in the same figure. Therefore, the total leakage volume is no more that 0.804 milliliter (24 drops x 0.0335 ml/drop). The total silt volume accumulated in the clearance is then calculated by:

$$(1.409 \times 10^7\ \mu m^3/mL \text{ of leakage}) \times (0.804 \text{ mL of leakage}) = 1.133 \times 10^7\ \mu m^3 \tag{89}$$

The total space available in the clearance of the piston-bore is calculated by:

$$Vcl = \pi\ L\ h\ (D - h) \tag{90}$$

where

L = 6.38mm ; land length

h = 6 μm ; clearance size

D = 6.351mm ; inner diameter of the bore

and is 7.63E10 cubic micrometers. Thus, while the silt volume (Eq. 89) found from the experiments only amounts to 1.79 % of the clearance space available, the silt particles block the leakage with a calculated efficiency corresponding to a summed volume of 60 to 80 percent of the clearance space occupied. Clearly, the model needs refinement. One explanation for this discrepancy is that the silt particles are locally concentrated in the clearance space and effectively increase resistance to the leakage flow through the clearance. Moreover, this packing takes place preferentially at the entrance to the clearance of the piston-bore assembly, as Fig. 54b shows. Results obtained by Kamijo [8] and Laurenson [9] support this contention.

To reconcile this discrepancy between the theory and the experimental results, the variation of the A value in Eq. 88 was investigated for different geometrical configurations of piston-bore assemblies. If A varies according to some non-dimensional parameters associated with the leakage flow through the clearance, then the use of the constant multiplier A in Eq. 88 is a valid approach for reconciling the discrepancy.

Table VI summarizes the result of finding A values, along with other information, for various assemblies. This table reveals several important facts. For instance, it was found that the larger the clearance, the larger the leakage and the total silt volume in the clearance. Furthermore, among piston-bore assemblies with the same clearance size, those with a land length of 3.19mm have a higher total silt volume in the clearance than those with a land length of 6.38mm. The piston-bore assemblies with a 3.19mm land length are less efficient in filtering particles than the assemblies with a 6.38mm land length (compare Fig. 47 and 51); however, by the end of the filtering process,

TABLE VI

SILT VOLUME ANALYSIS

CLEARANCE SIZE μm	SPOOL NO.	ACFTD CONCEN-TRATION mg/L	PRESSURE DIFFERENCE psi	LAND LENGTH mm	SILT VOL. / LEAKAGE μm^3/mL	AVERAGE LEAKAGE mL	TOTAL SILT VOLUME μm^3	% OF SILT VOLUME PER CLEARANCE SPACE	VALUE OF 'A'
6	19	25	1000	6.38	1.409E07	0.804	1.133E07	1.48	50
5	20	25	1000	3.19	1.354E07	1.325	1.794E07	5.64	11-12
9.5	23	25	1000	6.38	1.236E07	1.918	2.371E07	1.96	39-40
9.5	23	25	1000	3.19	1.075E07	2.43	2.612E07	2.163	17-18
14.5	25	25	1000	6.38	5.939E06	14.1	8.373E07	4.547	14-15
14.5	25	25	1000	3.19	3.583E06	13.49	4.833E07	5.249	14-15

the assemblies with a 3.19mm land length retain more particles in the clearance because these assemblies allow a larger amount of total leakage volume to pass through the clearance.

When the value of A (in logarithmic scale) is plotted against the average shear rate at the wall of the assembly clearance (in logarithmic scale), there seems to be a linear relation between the A values and the shear rates (Fig. 55). The linear regression of A versus shear rate is:

$$A = 2710381\ (\text{Shear Rate})^{-0.8547663} \tag{91}$$

The correlation coefficient of the above equation is 0.81. Thus, Equation 91 may be incorporated into the simulation program to account for the discrepancy between the experimental data and the silt process model.

For a given piston-bore assembly, only the pressure across the clearance of the assembly and the fluid viscosity determine the shear rate. At a constant temperature of fluid, it can be assumed that the fluid viscosity does not vary significantly. Since Dexron II is maintained at a constant temperature throughout the experiment, while the leakage flowing through the clearance is laminar flow, the pressure across the clearance becomes the only parameter that determines the shear rate of the assembly. Thus, the higher the pressure, the higher the shear rate. However, the constant multiplier A decreases as the pressure across the clearance increases. At high pressure, there exists a high leakage flow, and the rate of the silt volume accumulation in the

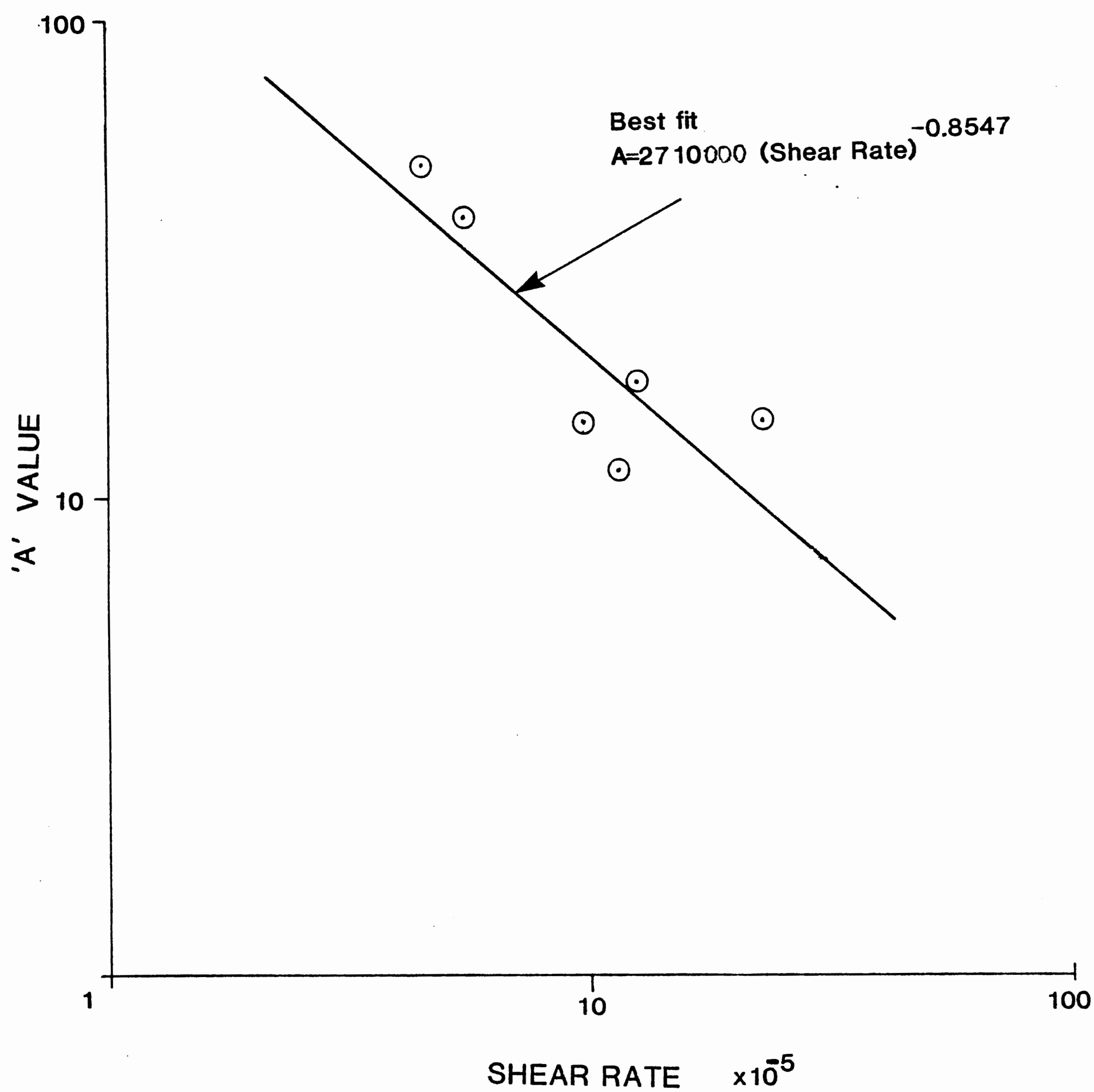


Figure 55. 'A' Value Versus Shear Rate.

clearance is high. This silt volume, however, does not block the leakage as effectively as the silt volume accumulating under a lower shear rate. When the fluid is first introduced into the clearance (at the beginning of the silting process), the average shear rate at the wall of the clearance is high. The leakage flow has the ability to transport particles finer than the clearance size deep into the clearance and even through the clearance. Consequently, silt particles are spreading and their build up is thin, thus causing the A values to decrease.

CHAPTER V

VERIFICATION OF PARTICULATE-INDUCED SEIZURE MODEL

Experimental Considerations

A review of past research [11, 12, & 25] reveals that the motion of the mechanism to measure the force required to break seizure must be well controlled so that each motion of the mechanism is repeatable and consistent.

For example, in contaminant diagnostic monitor research [25] seizure force has been used as a method to evaluate the contamination level of hydraulic systems (Fig. 56). The monitor used employed a weight system to actuate the tapered piston (the spool in the figure) in the bore housing as shown in Fig. 57. However, this actuation system had the following problems:

1. The correct weight to use for a given test condition was difficult to calculate in advance.
2. No actuator motion control was available.
3. The time allowed for the silt process to take place was imprecise because of a lack of automation.

The lack of an actuator motion control was a particularly serious

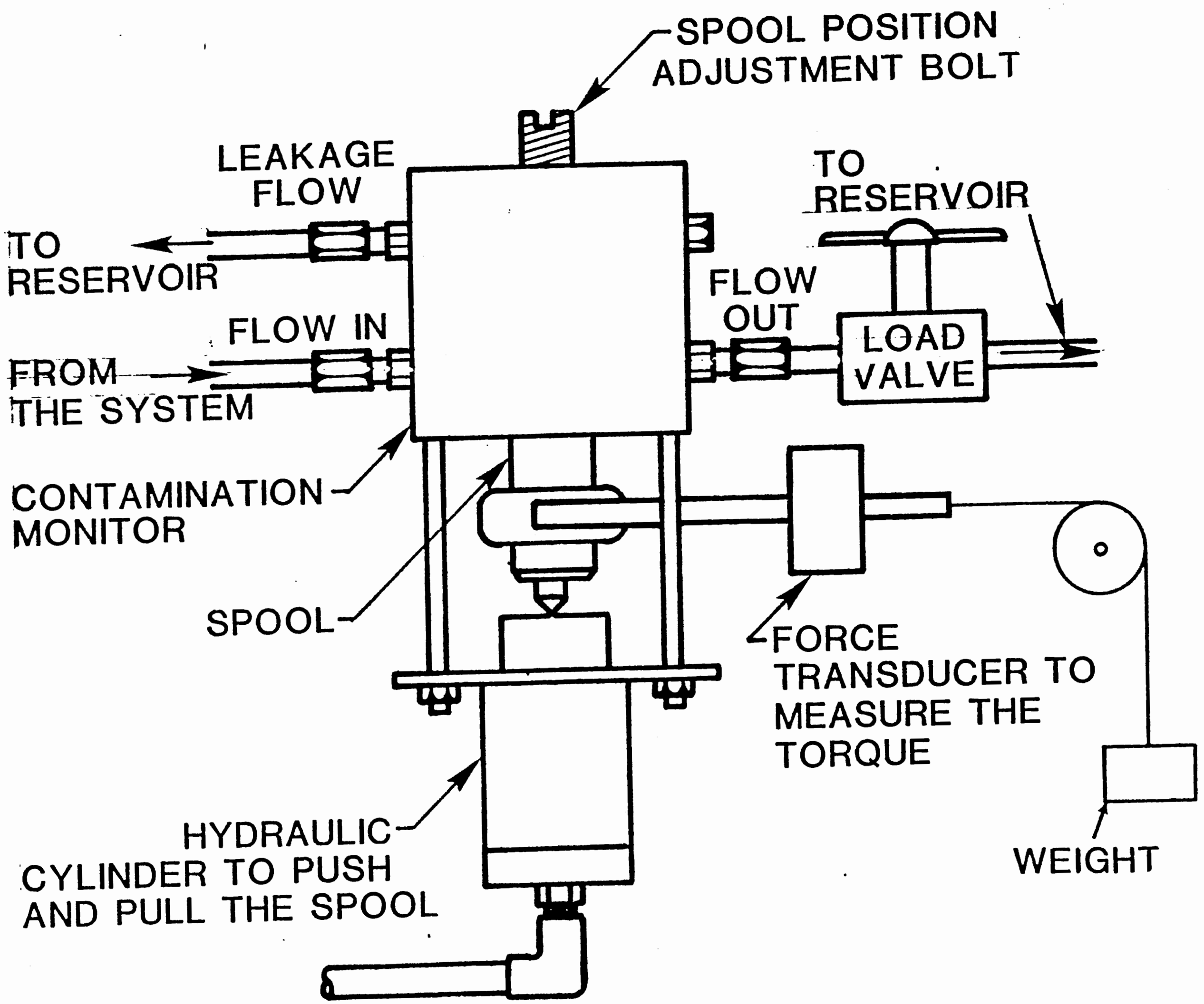


Figure 56. DOE Test Valve and System Set Up.

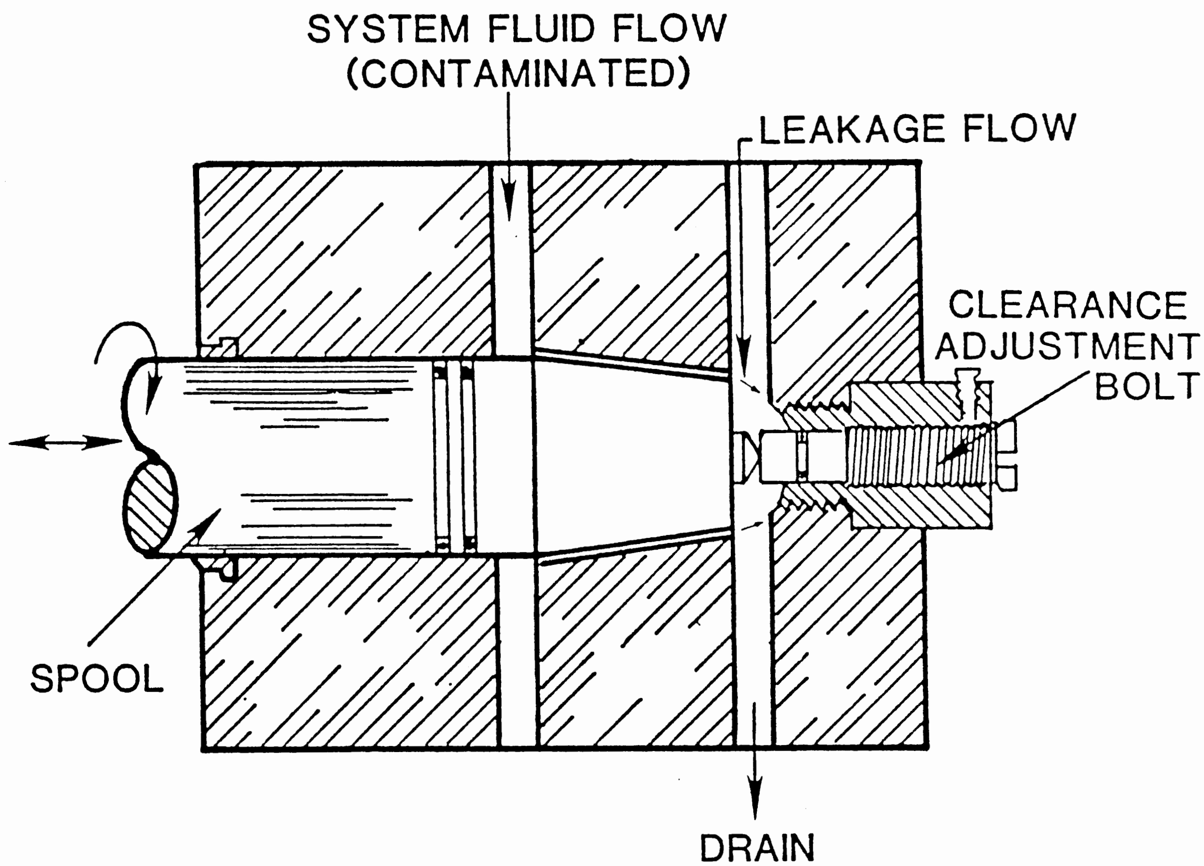


Figure 57. DOE Test Valve.

shortcoming. When the angular velocity of the spool rotation changed, the viscous drag changed. Ideally, the spool rotational motion was maintained constant throughout the test; however, when ACFTD was injected, particles entrained in the clearance impeded the rotational motion of the spool. Sometimes, a given weight could not overcome the seizure force generated by particles, and the spool rotation stopped in the middle of the process. In short, the rotational motion was impeded in an uncontrolled manner.

An improved actuator and motion controller for the actuator is definitely needed to verify the particulate-induced seizure model.

Development of Experimental Facility-Hardware

Mechanical hardware was designed (Fig. 58) and built to achieve this precise motion control. The rotational motion of the rotor at the top of the whole assembly was converted into linear motion through linkages and spheric joints. The force sensor was placed between the two linear bearing systems. The position sensor was attached to the linkage above the force transducer so that the position sensor would not affect the force measurement. A coupling was provided between the linkage and the actuation rod to ease possible misalignments. A hydraulic motor and hydraulic servovalves were chosen to drive the rotor because of their high frequency response and wide range of controllable speed. In addition, this electrohydraulic servo system could generate more power than an electrical motor of equivalent physical size. An additional

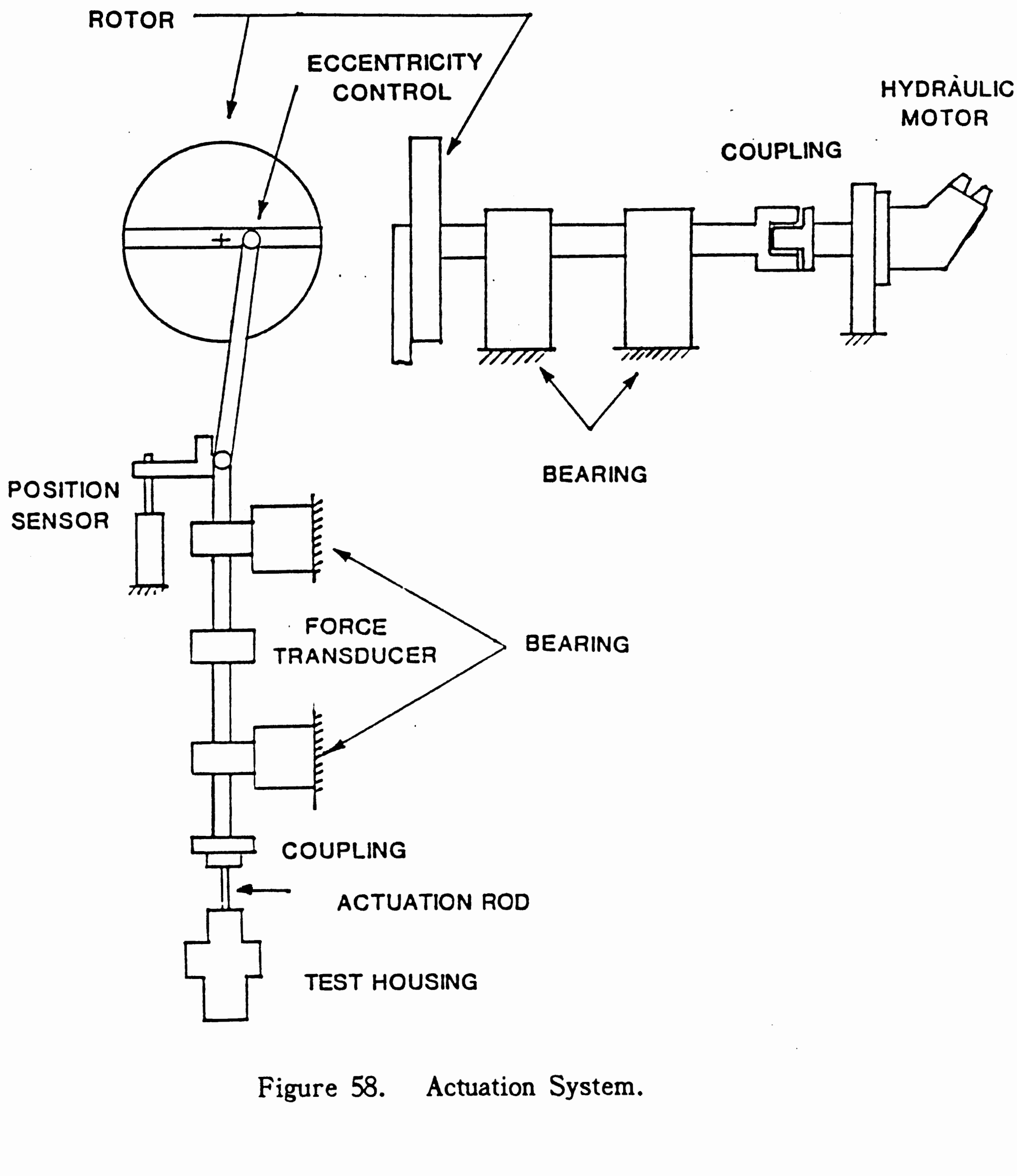


Figure 58. Actuation System.

power unit for this electrohydraulic servo system was built to supply constant pressure to the hydraulic servovalve.

Electronics hardware was either purchased or developed to achieve the desired motion control of the actuation system. A microcomputer system was developed to control the mechanics. This low level system was also designed to communicate with other computer systems (Fig. 59). The machine interface microcomputer consists of six major functions: main processor, communication, memory, on/off input and output, D/A, and A/D. The heart of this device is the control function designed around the Intel 80188 processor. The IEEE 488 card enables the machine interface to transfer information to and from a personal computer (Tandy 2000) at the speed of 500k bytes per second. The combination of a 12 bits D/A (digital to analog converter) and a power operational amplifier drives the hydraulic servovalve. The output of the position sensor from the servovalve was fed directly into a 12 bits A/D (analog to digital converter); however, the output of the force transducer was fed through the signal conditioning units to achieve noise suppressed signal amplification. The schematic diagram of this unit is shown in Fig. 60. The signal conditioning unit was fully equipped with noise rejection capability, a low pass filter, and laser trimmed gain accuracy. The I/0 unit and opto-electronic unit turn on and off the 120 volts solenoid valve placed downstream of the test housing.

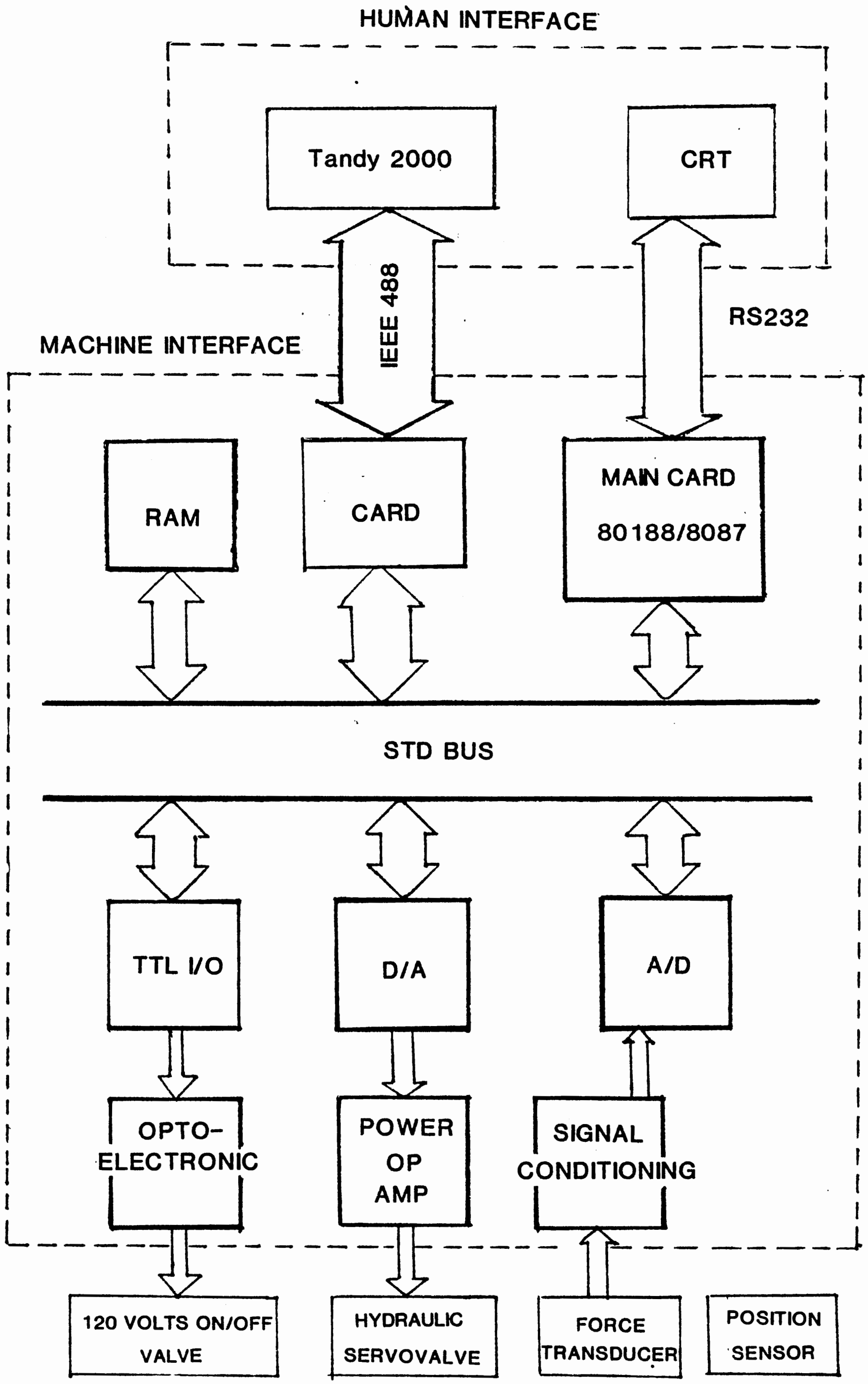


Figure 59. Microcomputer System Hardware.

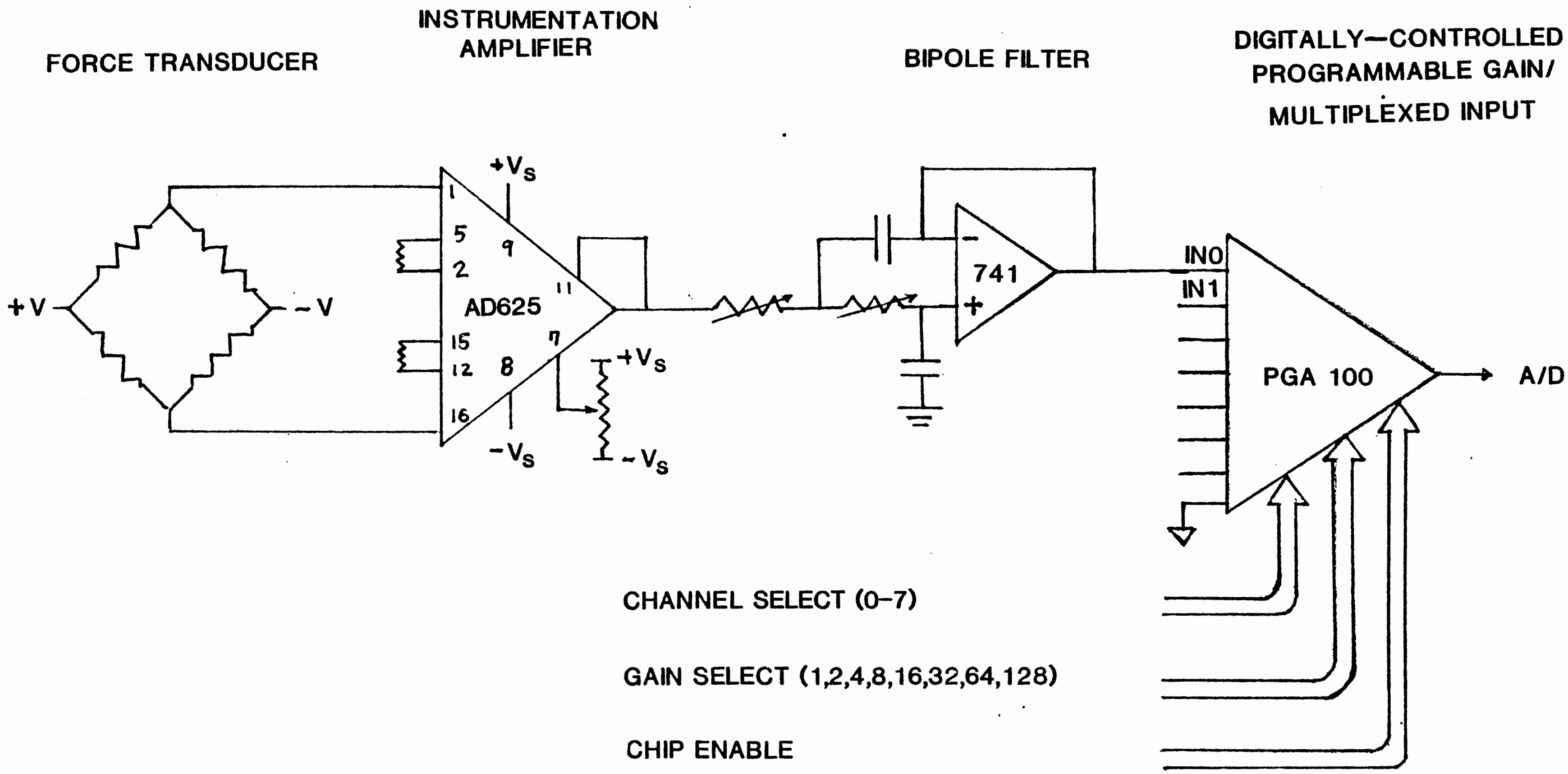


Figure 60. Signal Conditioning Unit.

Development of Experimental Facility-Software

The hardware part of the system therefore consisted of a test bed, a low level computer, and a high level computer. Software was written to effect a smooth interface between these independent units. Critical to the success of this software was careful attention to the operational priorities of each unit. The communication between the two computers was initiated only upon request by the master (high level) computer. The details of the low level operation of the test bed were looked after entirely by the slave computer. Because of the need for high speed and precise timing of the various operations, the slave computer was designed and built with a minimal number of allowable interrupts.

The main task of the slave computer is the tight loop of scanning the request from the master computer (Fig. 61). Then, the scanning routine first finds whether the slave computer is to listen to what the master computer talks, or to talk to the master computer. The master must become the talker and the slave the listener when the communication is initiated; then, only when the master computer requests the slave computer to talk, the slave computer talks. The scan routine algorithm shown in Fig. 61 achieves this listener-talker alternation.

When the slave computer listens to the master computer, the master computer gives a series of instructions and information to achieve the desired task. Therefore, these instructions must be decoded (and executed if so required) immediately. While the master and slave computer communicate, both computers are executing their task at a much

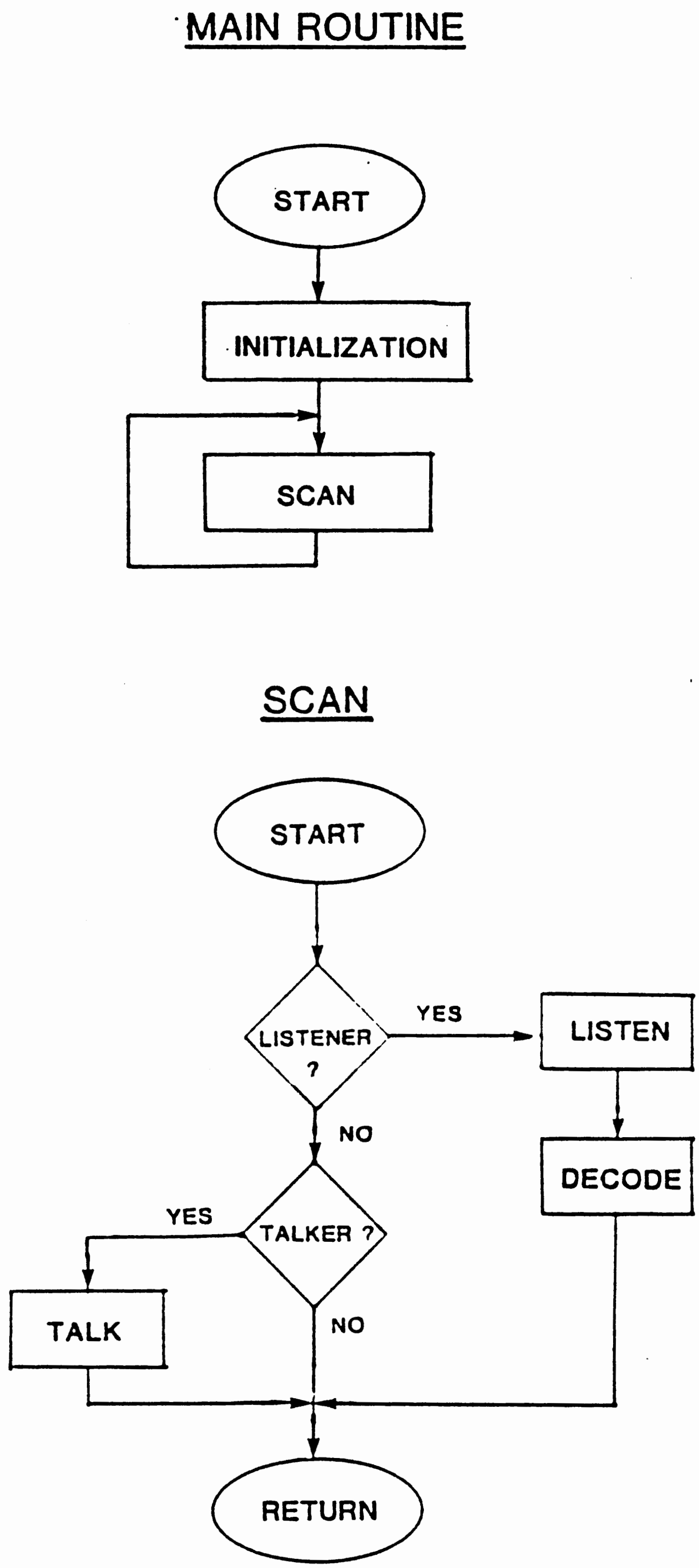


Figure 61. Algorithm of Slave Computer.

recuced speed because the response time of the communication hardware between computers is slow compared to the speed of the computers themselves. To minimize this delay the instruction set was written in as compact a manner as feasible.

Figure 62 illustrates the approach used with the slave computer for this experiment. The first byte received determines the type of operation. As the example in the figure shows, if this byte is 00H (zero-zero in hexadecimal), then the task is A/D conversion. The information of the channel number and the required number of conversion then follows as Fig. 62 shows. This way the master computer can access as many as 256 channels and trigger as many as 65536 A/D conversions by sending a mere four bytes of instructions to the slave computer.

A total of eight levels of task selections were implemented in the slave computer (Fig. 63), and the source code for this software is presented in Appendix C. The software shown in Appendix C was first developed on a Tandy 2000 (the master computer) and subsequently downloaded through an RS 232C serial port to the slave computer. The file downloading program was written in a combination of C language and Assembly. This program used the BIOS (basic input and output sevice routine) interrupt 14H (fourteen in hexadecimal) to drive the RS 232C of the Tandy 2000 and 16H to control the keyboard buffer. The software for the communications between the Tandy 2000 and the slave computer is presented in Appendix D.

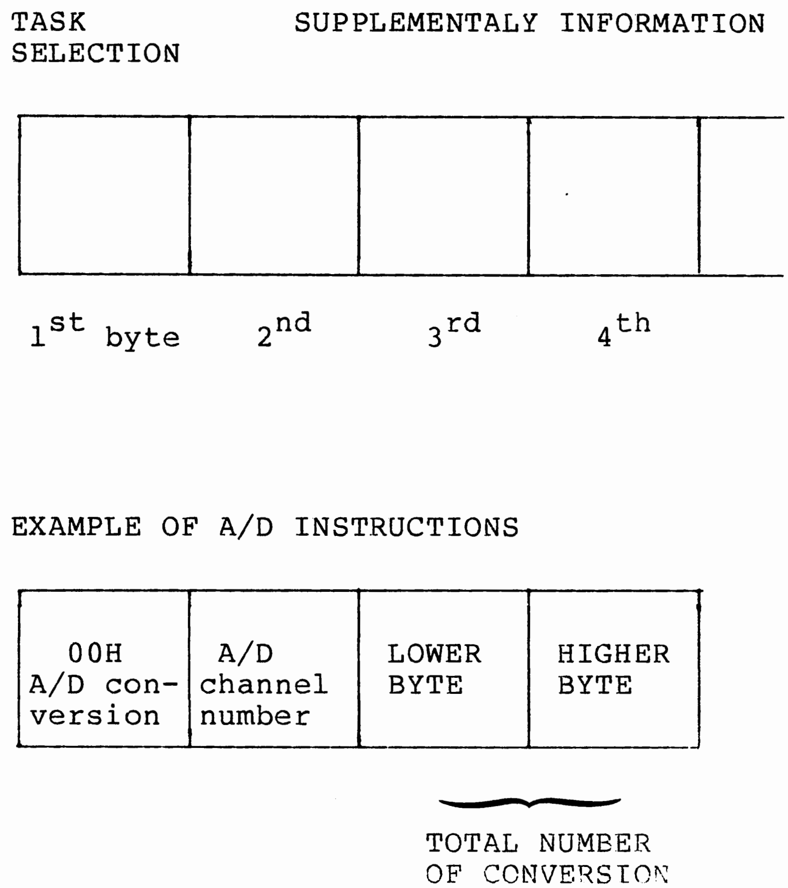


Figure 62. String of Instructions.

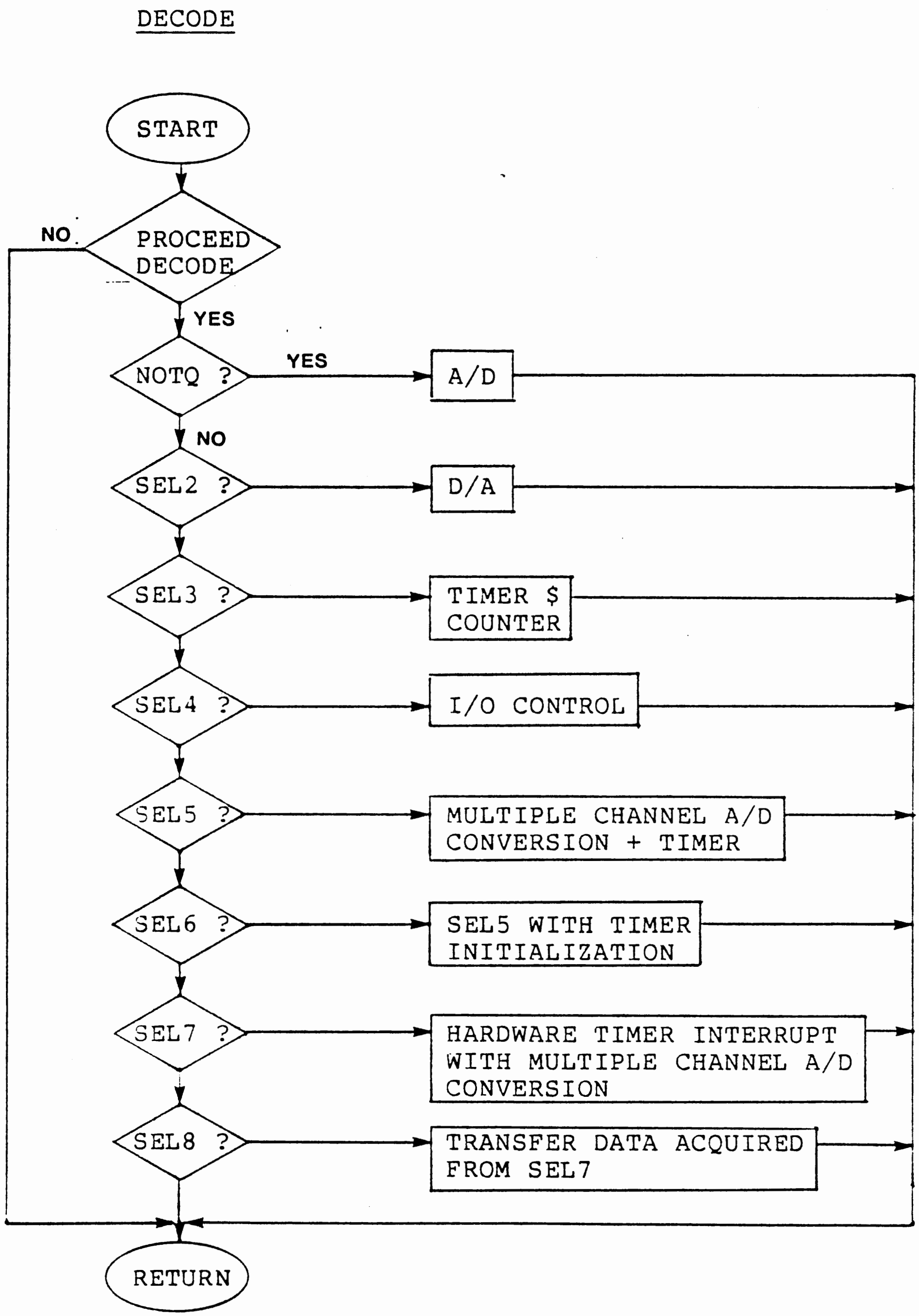


Figure 63. Decoding Routine.

Actuation Mechanism Control

The rotor which controls the position of a piston in the test housing is driven by the hydraulic motor shown in Fig. 64. The speed of the rotor is regulated by the hydraulic servovalve. The orifice of the servovalve opens by an amount proportional to the input current delivered by the power amplifier and D/A. A pressure relief valve maintains a constant supply of pressure to the servovalve constant. A heat exchanger keeps the temperature of the hydraulic fluid constant, and a hydraulic filter cleans the hydraulic fluid as this fluid moves through the hydraulic circuitry.

While the rotor turns, the position of the piston is monitored by a position sensor. When the piston is out of the bore (Fig. 65b), the downstream on/off solenoid valve is closed so that the fluid flushed through the piston-bore assembly can escape through the check valve to the reservoir. When the piston is inside the bore (Fig. 65a), the solenoid valve remains open so that the pressure difference between the upstream and downstream sides of the piston is referenced to atmospheric pressure. Under this condition, the leakage starts upstream and moves downstream through the clearance. These flushing and silting conditions must be well coordinated by the software which controls the actuation mechanism. The force required to pull the piston out of the bore must be measured. This force will be equal to the initial friction when the hydraulic fluid is filtered and clean. However, when the fluid contains solid particles such as ACFTD full distribution, the force required to

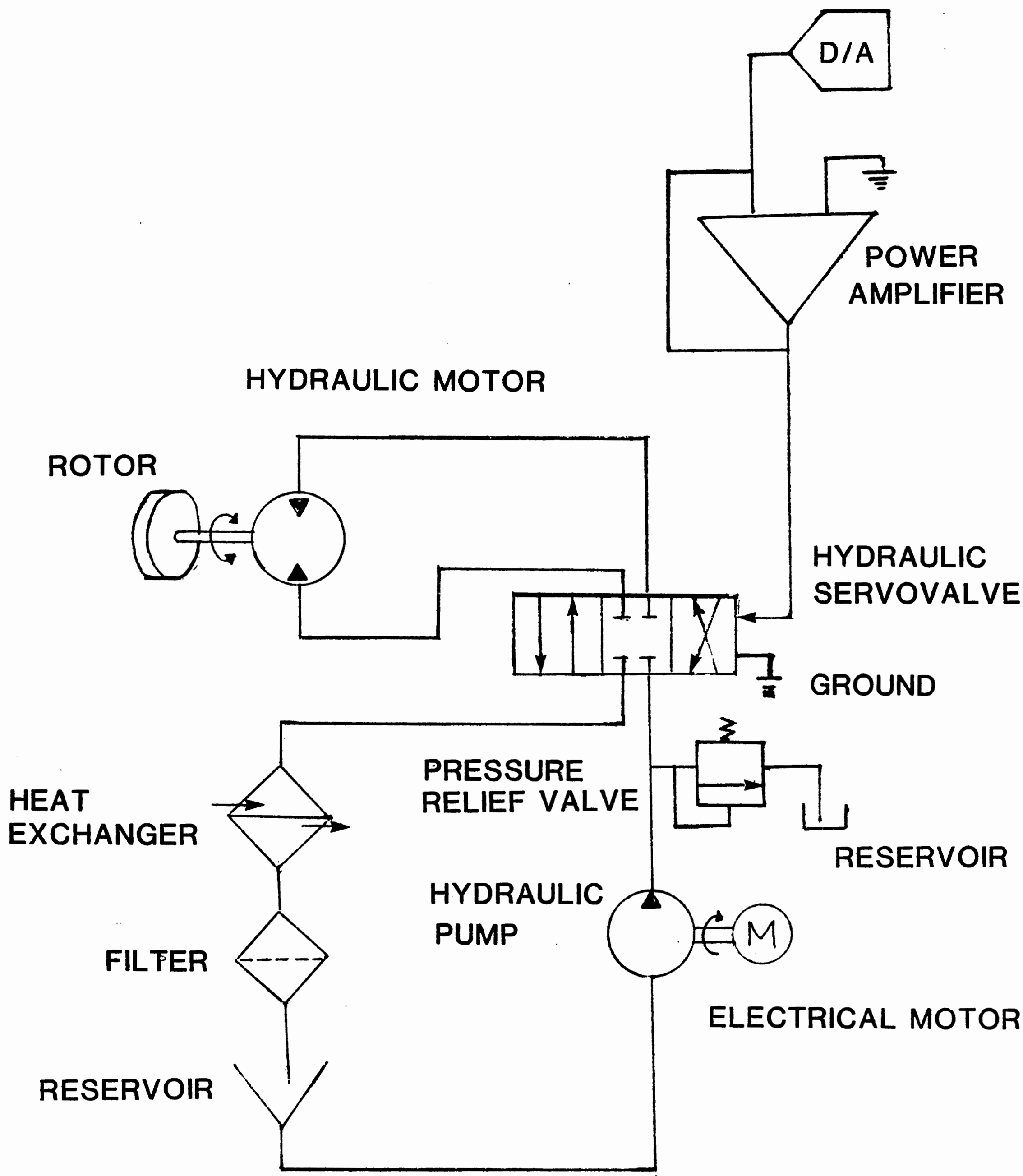


Figure 64. Hydraulic Circuitry of Actuation System.

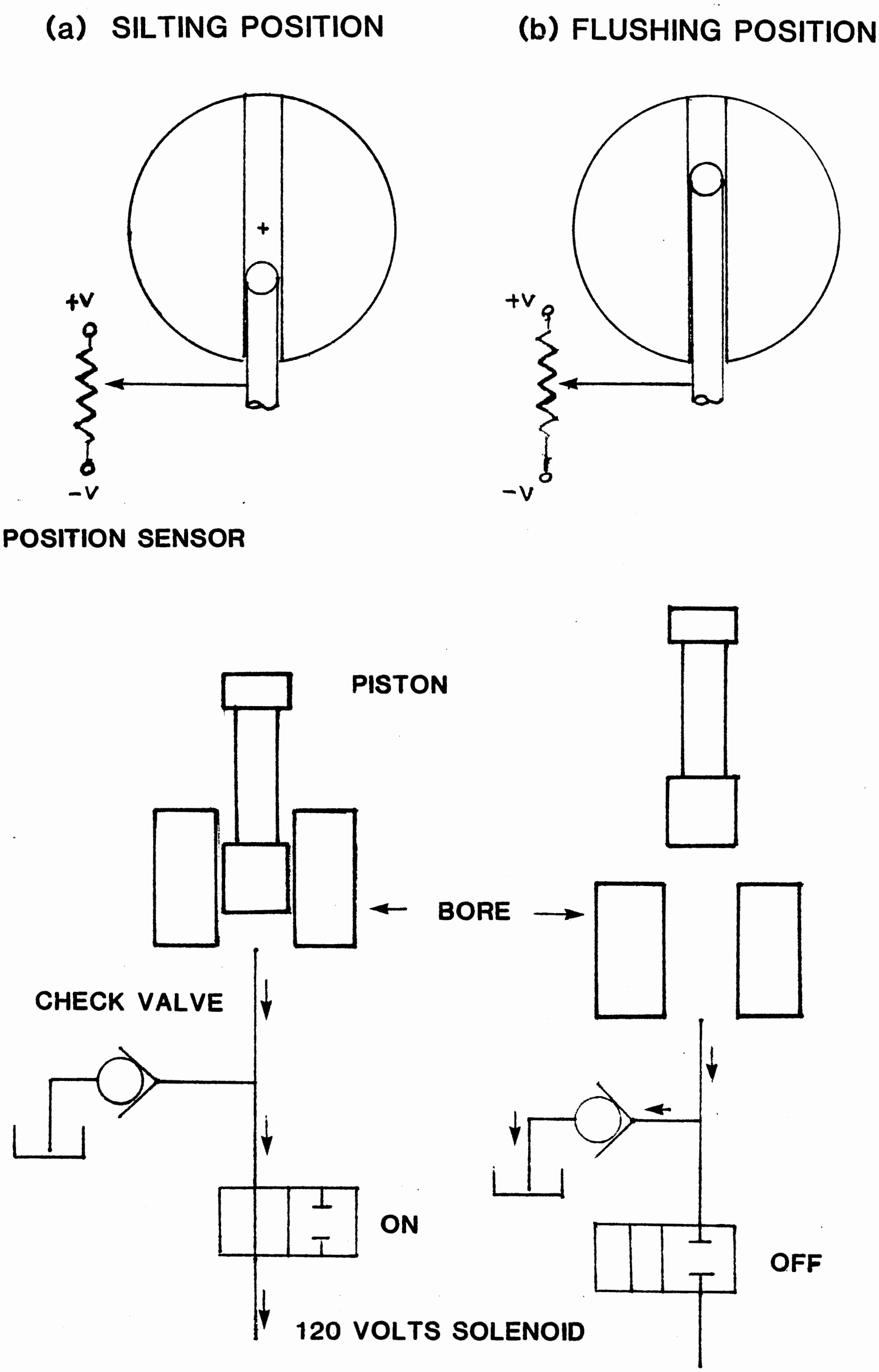


Figure 65. Piston Actuation Mechanism.

pull the piston out of the bore is expected to increase beyond the initial friction. This increase in force is the force required to break seizure. The algorithm used by the master computer is shown in Fig. 66. The main program sends the instructions to close the solenoid valve at the downstream of the test housing. Then, the master sends instructions which cause the rotor to turn so that the maximum and minimum position of the piston are checked. The rotation of the rotor also serves to flush the piston-bore assembly. The position of the piston is monitored throughout this motion by feed back control to the A/D converter which is suitably acknowledged by the master. This feed back control lowers the piston, inserts the piston into the bore, and stops the motion of the piston for the silt process. As soon as the final position of the piston is set, the master computer commands the slave computer to open the solenoid valve and allow the silt process to take place in the clearance of the piston-bore assembly for a pre-defined time (stationary time). The master computer keeps track of this stationary time, and, as soon as the defined stationary time is past, the master computer commands the slave computer to close the solenoid valve and initialize the hardware timer interrupt routine.

The hardware timer interrupt routine, when programmed for a certain period (e.g. 10 milliseconds), sends a signal to the hardware interrupt controller every 10 milliseconds. The hardware interrupt controller then signals the central processing unit (CPU) to stop execution of the present program and switch to the independent interrupt service program. This interrupt service program is executed every 10 milliseconds. The

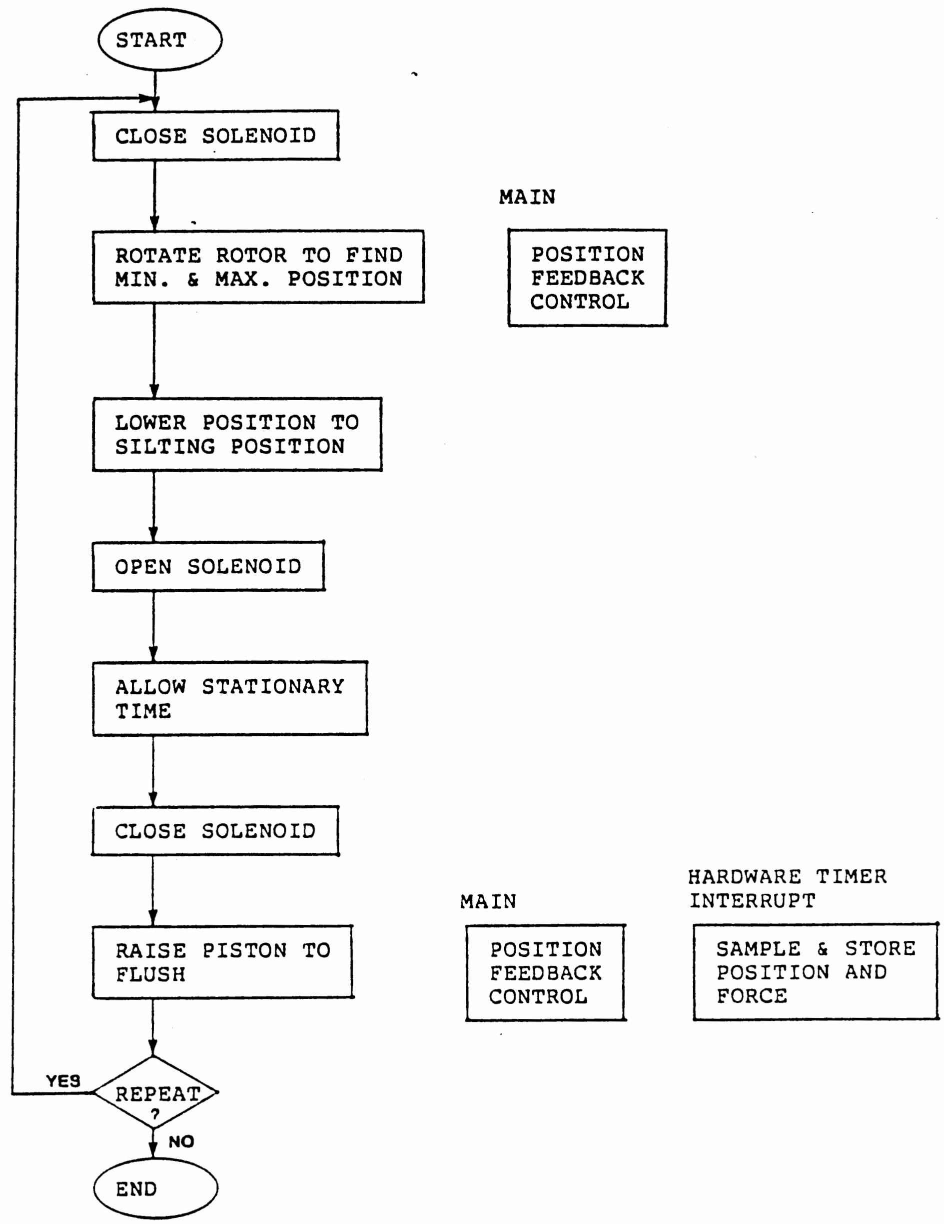


Figure 66. Actuation Algorithm for Main CPU.

duration of this program must not exceed that required by the CPU to execute this program. As soon as the CPU finishes executing the interrupt service program, the CPU returns control to the main program and resumes execution of the main program. This hardware timer interrupt procedure allows the slave computer to execute a time-critical program along with a non-time-critical (main) program. Appendix E contains a listing of the source code for this main program.

It is now apparent that the integration of the hardware and software achieves the following:

1. Precise time control of the piston movement.
2. Precise and accurate motion of a piston.
3. An actuation mechanism control with virtually no human error

Further improvements were made to reduce the initial friction between the actuation rod and the housing by adding a teflon cap seal between the actuation rod and the O ring seal.

A preliminary test run of the actuation mechanism revealed that a guide is needed to smoothly insert a test piston into a bore smoothly. Without a guide, an excessive misalignment between the axis of the piston and the axis of the bore can cause the piston to collide with the bore. A guide cap was therefore designed and built, and a 45 degrees chanfer was machined on the bore surface as shown in Fig. 67. These additions immediately corrected any misalignment problems.

Another problem was noted once testing began. This appeared to be due to flow-induced vibration of the piston. When the piston is outside the bore, the circulating hydraulic fluid hits the side of the piston.

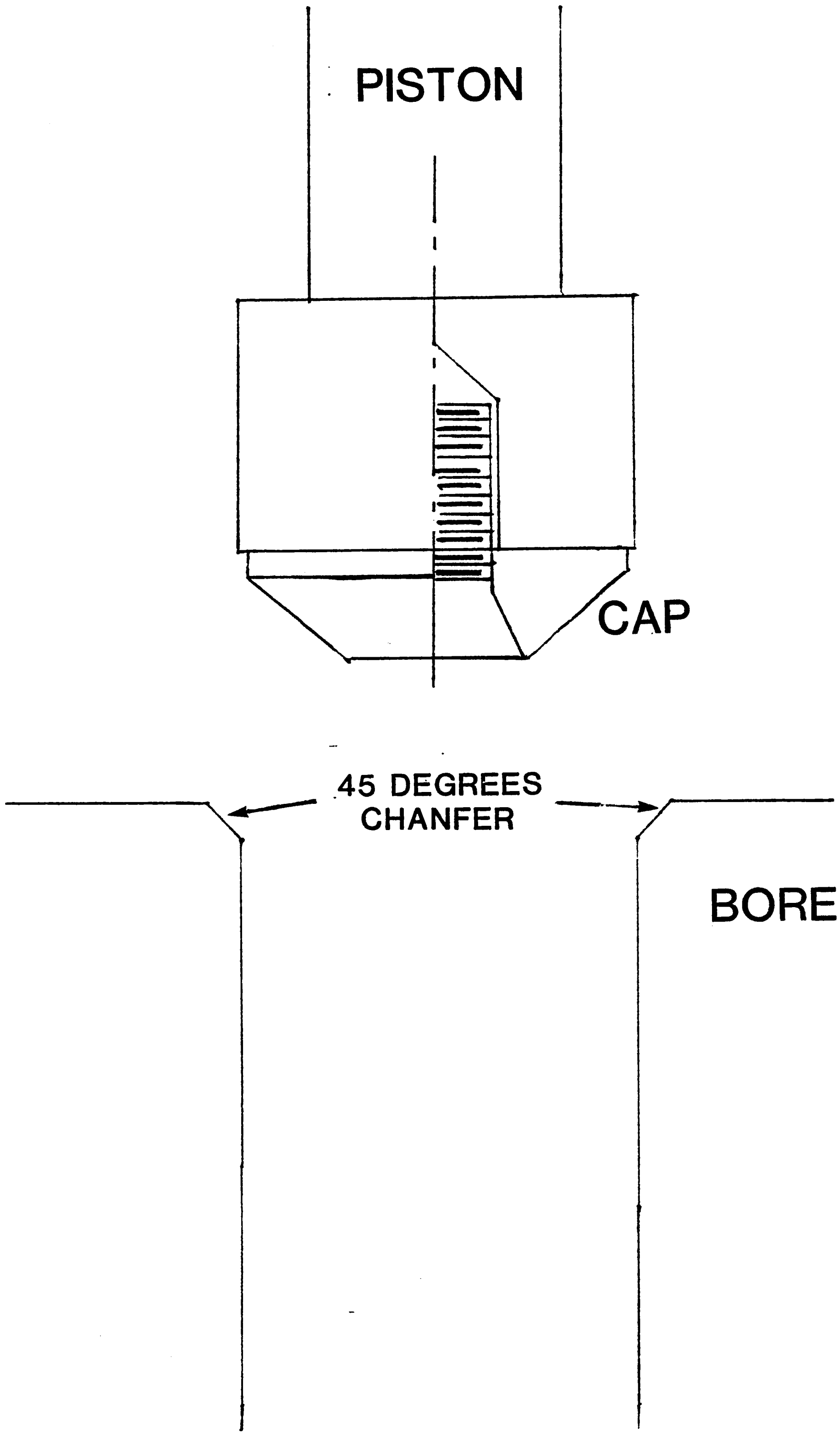


Figure 67. Alignment Improvements.

At 500 psi, this induced vibration is sufficient to unscrew the piston from the actuation rod. An effective remedy to this problem was to use a commercially available threadlocker called "LOCTITE."

Experimental Results and Analysis

The verification tests of the silt process model provide the number of particles, and their size, retained in the clearance of a piston-bore assembly. These particles in the clearance cause a sudden increase in the force which moves the piston from the bore. The experimental facilities are organized to measure this increase in force. To measure the seizure force associated with the presence of silt particles in the clearance of the piston-bore assembly, the initial friction force had to be measured first while the hydraulic fluid (Dexron II for this case) was clean. This initial friction force had to be subtracted from the total seizure force measured after the injection of solid particles. The resultant force was the additional force needed to overcome the seizure caused by the solid particles in the clearance (Fig. 68). From hereafter, this additional force is called the seizure force.

Each piston had two land lengths: 6.38mm and 3.19mm. Thus, a given pair of piston-bore assemblies could be tested for two land lengths. Each pair was tested under three different pressures (100, 200, and 500 psi) and three stationary times (15, 30, and 60 seconds), the stationary time being that allowed for particles to silt in the clearance. For each test condition, at least three measurements were performed. Thus, for

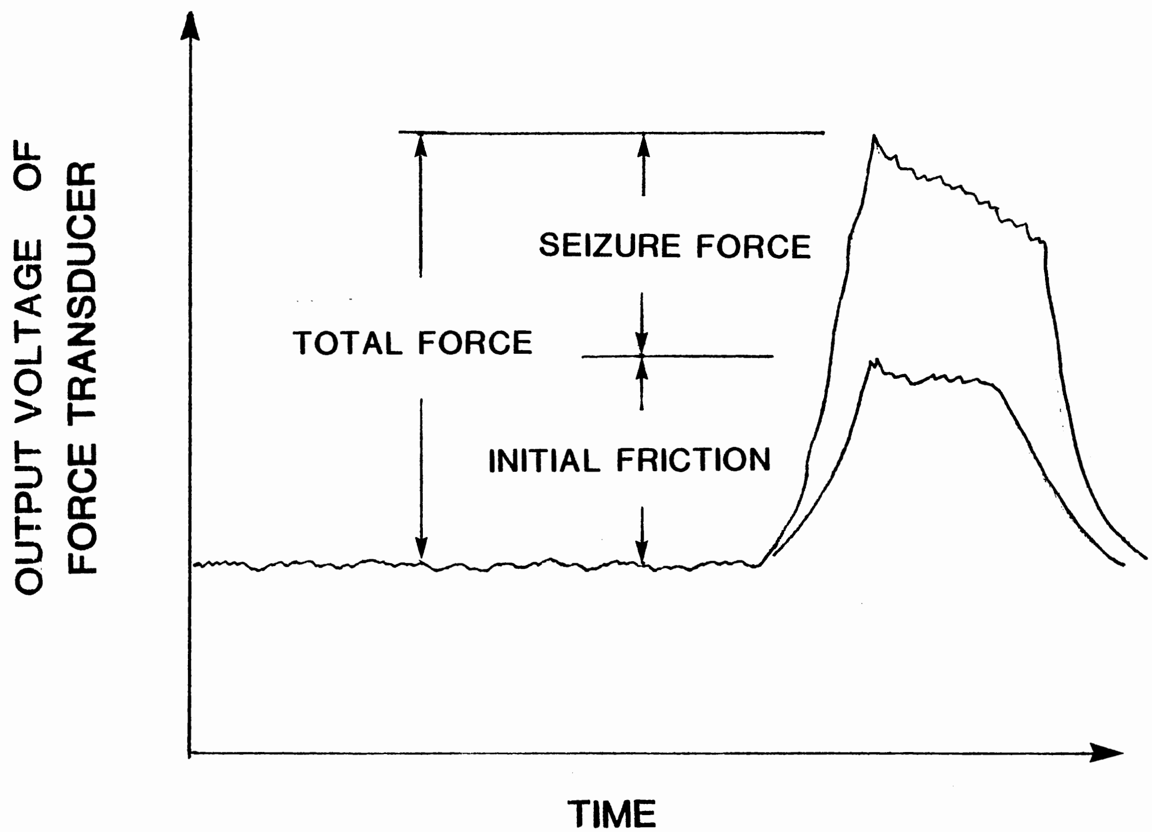


Figure 68. Seizure Force.

each pair of piston-bore assemblies, 54 measurements were made.

ACFTD full distribution was used as the source of solid particles for this particulate-induced seizure test. Enough ACFTD was injected to achieve a solids concentration of 25 mg/L. Since ACFTD full distribution can not be maintained for more than one hour in the hydraulic system used for this experiment [25], and because a set of 54 tests was needed (as explained above), each test could not be permitted to exceed a duration of one minute.

To complete the experimental set, pistons of different hardness were chosen to investigate the effect of hardness on seizure force. The pistons are listed in Table VII. The No. 3 bore with a hardness of 725 Brinell (Rockwell 62-64 on the C scale) was not swapped out for other pistons and remained in the test housing throughout the test. All the experimental results are summarized in the seizure force analysis figures in Appendix F. Three representative seizure force analysis figures were chosen for the following analysis and discussion. The piston-bore assemblies in these representative figures had the same 10 micrometer nominal clearance, and a 6.38mm land length. Only the hardness of the pistons differed. Figure 69 shows the results for the tests of the piston with a hardness of 725 Brinell. The dark lines show the average force predicted by simulation, and the circles, triangles, and rectangles with the dotted lines represent the averaged data points. Since the data points in the figure are the average of three test data, some scattering of the data was expected. Above 200 psi, the experimentally measured seizure forces were consistently less than the simulation results. For

TABLE VII

PISTON SELECTION FOR HARDNESS

BORE NO.	PISTON NO.	NOMINAL CLEARANCE μm	PISTON HARDNESS (Brinell)
3	22	10	725
	25	15	725
	1	5	120
	2	5	120
	5	10	120
	8	15	120
	11	5	400
	14	10	400
	17	15	400

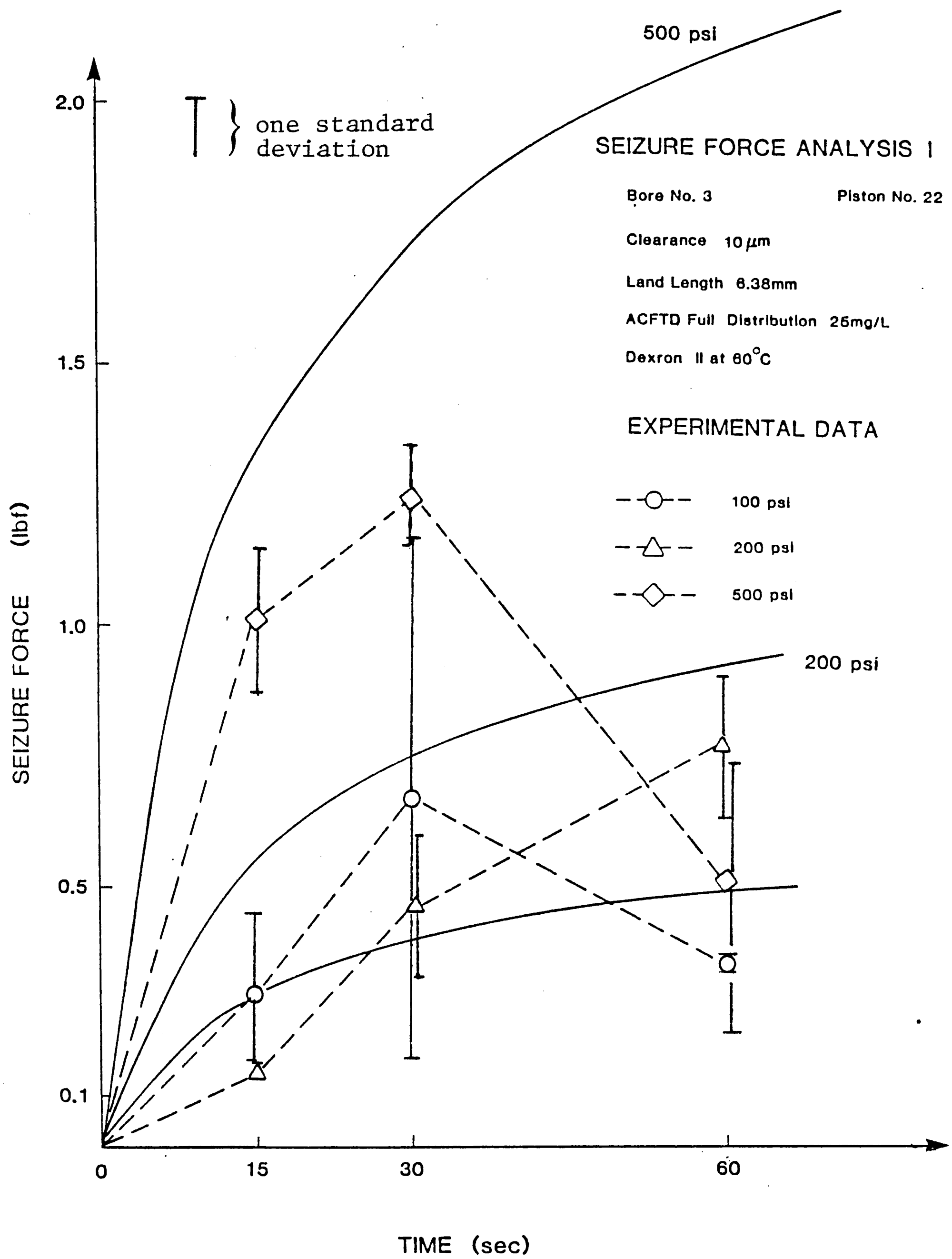


Figure 69. Seizure Force Analysis (I).

the assemblies described by the data in Fig. 69, both the piston and bore were harder than ACFTD, which has an estimated hardness of 500 Brinell. Therefore, the particles were expected to be deformed or crushed under the experimental conditions. It is also possible that Dexron II acted as a lubricant and helped the solid particles to reorient themselves through sliding or cutting the piston and/or bore surfaces. Microscopic observation (Fig. 70) clearly supports that conspicuous cutting action took place on the piston surface at the upstream side where solid particles silted. The controlling stress for seizure force in this case was the yield stress estimated from the hardness of silca (500 in Brinell) through:

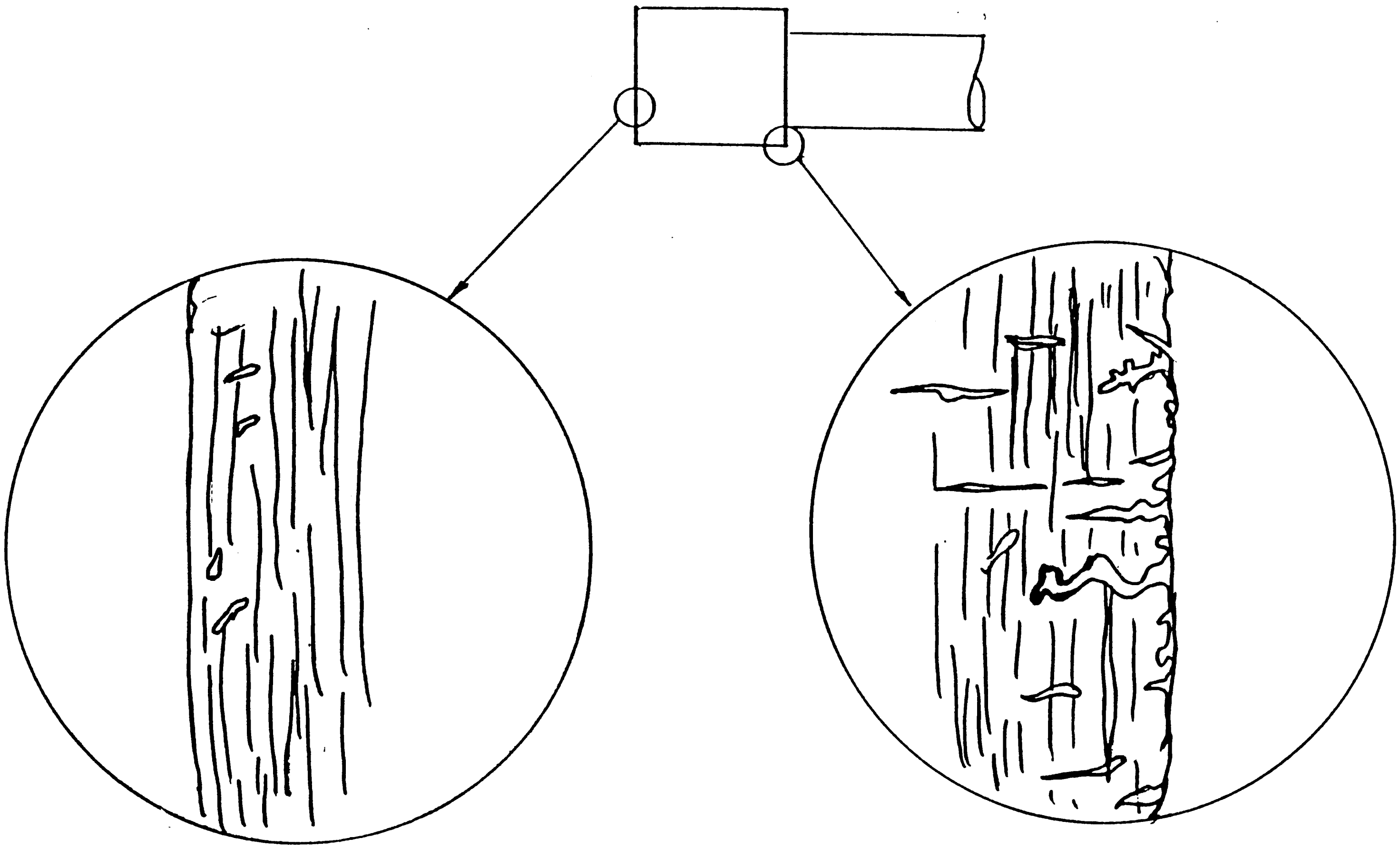
$$\text{Yield Stress} = 500\ H_B \ /\ 1.8 \qquad (92)$$

where

H_B = Brinell hardness.

In contrast to Fig. 69, the experimental data tends to be higher than the simulation results in Fig. 71. For the case in Fig. 71, the piston surface was slightly softer (400 in Brinell) than ACFTD; therefore, particles should have been able to indent into the surface of the piston. If particles only indented, then essentially, the particles should have stayed out of the piston motion, and the seizure force should have decreased. However, more than just indenting action occured.

Figure 72 is a sketch of the microscopic picture of the upstream side of the surface of the No. 14 piston. In Figure 70, the ground



DOWNSTREAM x400 **UPSTREAM x400**

Figure 70. Microscopic Observation of Piston Surfaces.

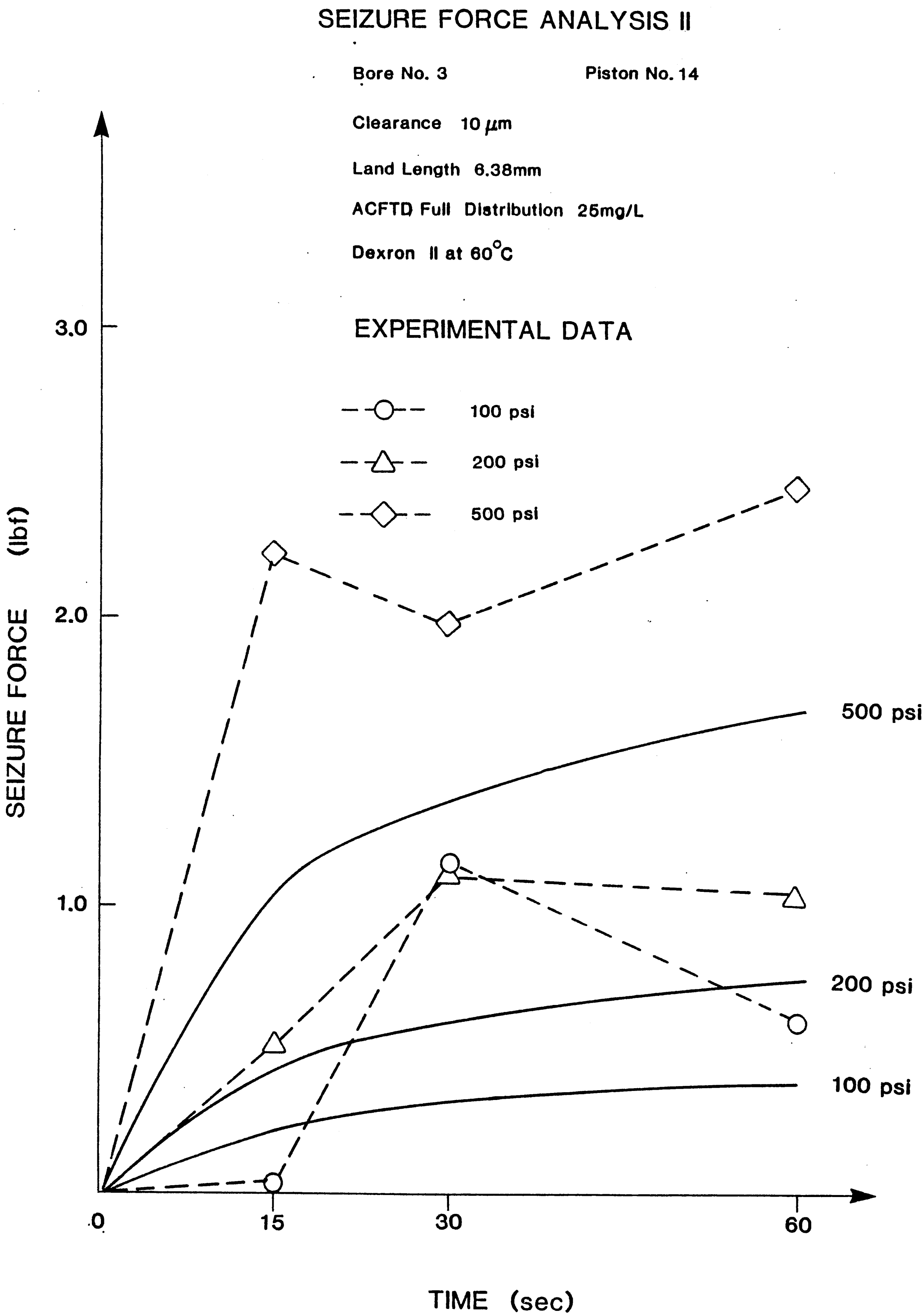


Figure 71. Seizure Force Analysis (II).

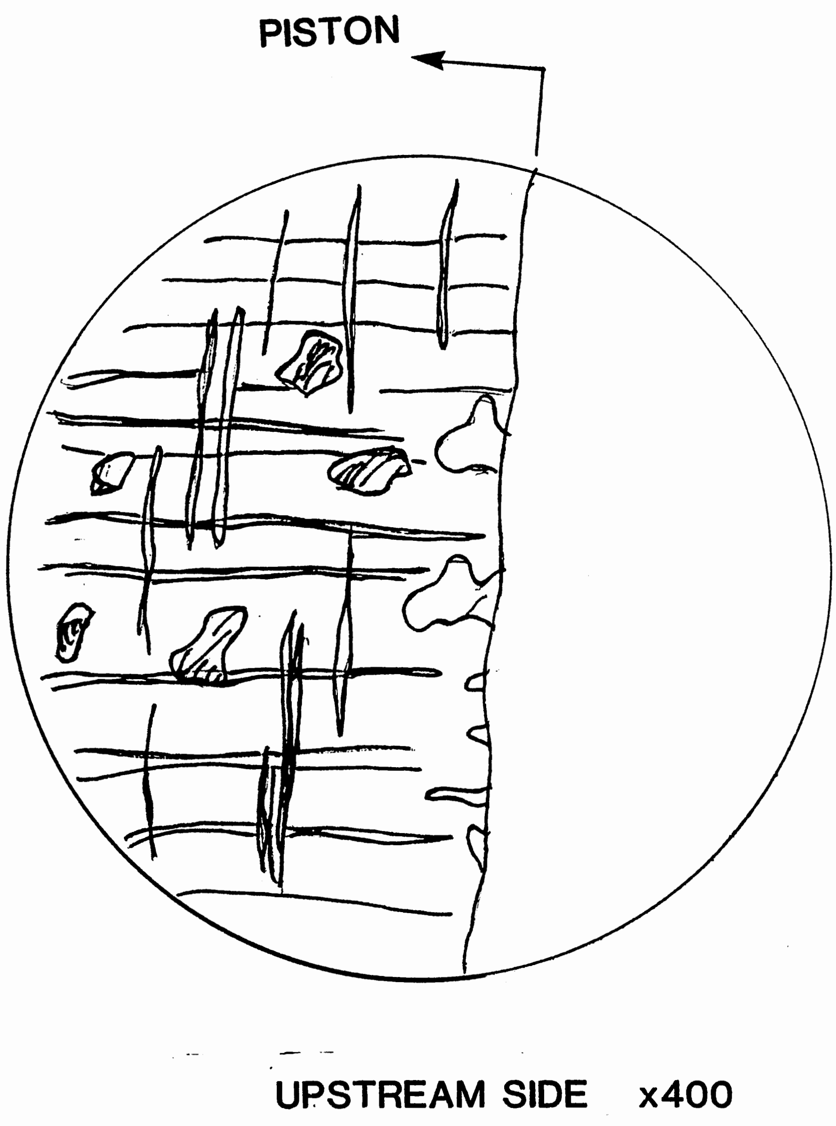


Figure 72. Microscopic Observation of Piston (Rc 43-45).

surface (verticle lines) is obvious; however, in Fig. 72, almost all trace of the grinding action is gone. Instead, there are many lateral lines that must have been created by ACFTD. In addition, there are several noticeable small holes which also must have been created by ACFTD. This means that many ACFTD not only indented into the surface causing the visible small holes, but also performed massive cutting action (shown by the lateral lines). In order for continuous cutting action to take place, there must be sufficient force to plastically deform the piston surface. This may be why the seizure force was higher than what the simulation program predicts. The particulate-induced seizure model predicts seizure force based upon the yield stress, not the ultimate stress, of interacting surfaces. Figure 73 shows the experimental and simulation results for the softest of the three pistons. The experimental data tend to be higher than the simulation data, as Fig. 71 shows. Figure 74 shows the upstream side surface of the piston used for the experiment. The edge of the surface was so destroyed that it no longer retained its original shape. Many small holes and lateral lines were obvious through the microscope.

The results shown in Fig. 71 and Fig. 74 indicate that the simulation result of seizure force will closely model the experimental result if the stress which has a higher value than the yield stress of the softest material among the piston-bore and particle is used.

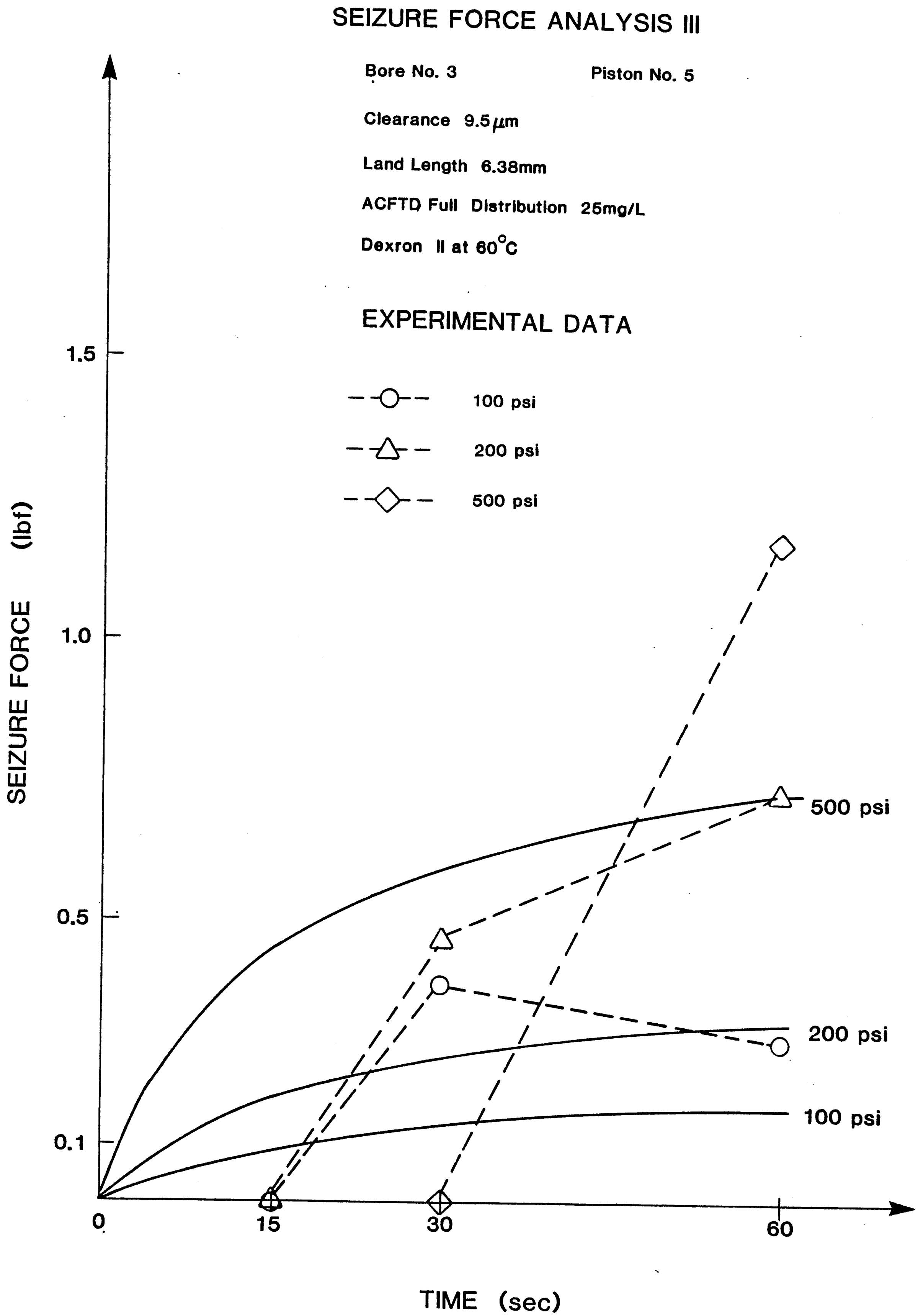


Figure 73. Seizure Force Analysis (III).

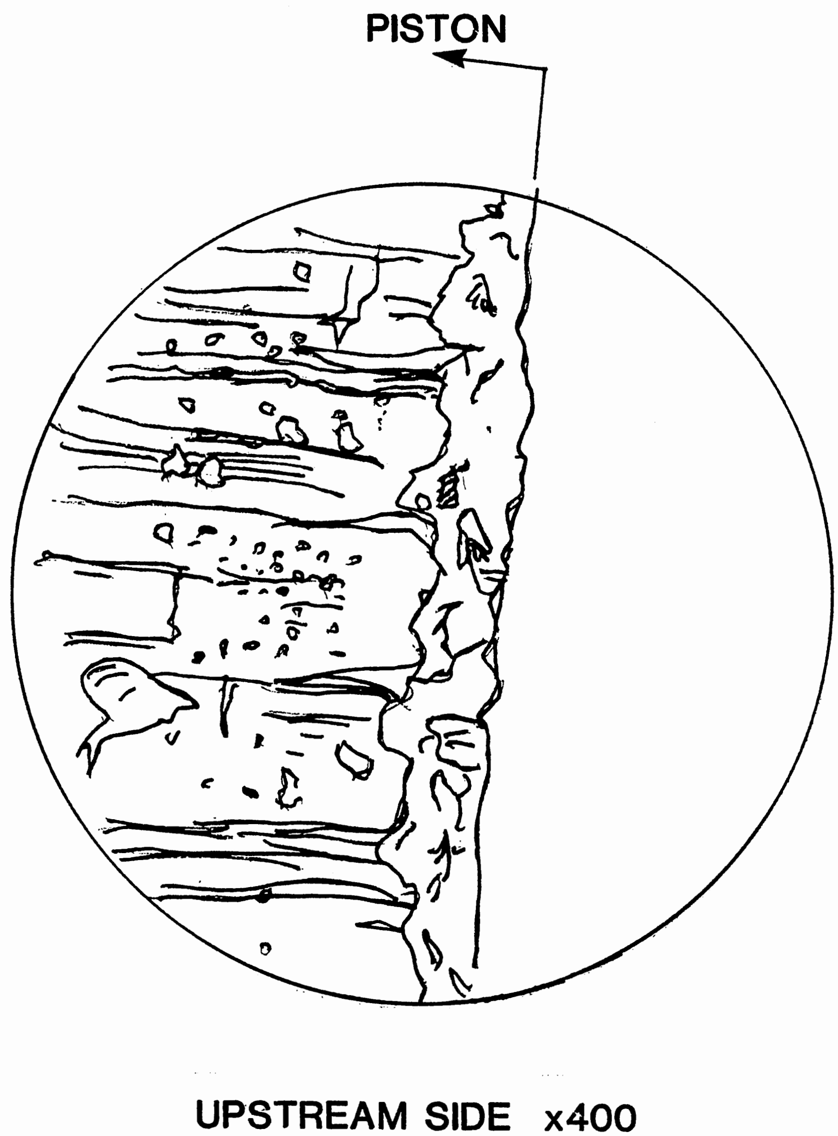


Figure 74. Microscopic Observation of Piston (H_B = 120).

CHAPTER VI

VERIFICATION OF THE DITHER MODEL

Kamijo et. al. observed that dither is capable of preventing non-uniform pressure distribution in the clearance of a piston-bore assembly, and suggested that the solid particles in the clearance are removed by dither [8]. If this is true, then with a given piston-bore assembly and a constant upstream particle size distribution, there should be an increased number of particles at the downstream side of a piston-bore assembly when dither is applied to the piston. This means that silt Beta values of a given piston-bore assembly should decrease when dither is applied. Nikitin's experimental results [13] suggest that the higher the frequency of dither, the more effective is dither in reducing the frictional force required to move the piston of a piston-bore assembly. Kamijo and Nikitin's findings suggest that silt Beta values of a given piston-bore assembly may be a function of dither frequency. The following questions arise:

1. How do silt Beta values of a piston-bore assembly vary as dither is applied to the piston?
2. How does seizure force vary when dither is applied to the piston?

Experimental Facility

The development of the piston actuation mechanism for dither required only a slight modification to the mechanism, shown in Fig. 58. The eccentricity position of the joint between the linkage and rotor was positioned such that the maximum travel of the piston rod was 0.762 millimeter plus or minus 0.08 milliliter. Thus, uni-directional rotation of the rotor caused dither action of the piston with the amplitude of 0.762 millimeter. The frequency of dither could be controlled by varying the input voltage across the servovalves.

For the determination of silt Beta values, the leakage fluid from the tested assembly was collected and analyzed by the method presented in Chapter IV.

Experimental Result and Analysis

The upstream pressure of the piston-bore assemblies was fixed at 1000 psi. The temperature of Dexron II fluid was maintained at 60 degrees Celsius. The No. 6 piston and No. 23 piston (both giving a nominal clearance of 10 micrometers with the No. 3 bore) were chosen for this dither so that the effects of piston hardness could also be studied. The hardness of piston No. 6 was 725 Brinell (harder than ACFTD), while the hardness of piston No. 23 was 120 Brinell (softer than ACFTD).

ACFTD full distribution was chosen as the solid particles source for the dither test. ACFTD was injected from the reservoir to achieve a concentration of 25 mg/L. When the fluid was filtered and clean, the upstream pressure was raised to 1000 psi, and the downstream pressure was open to atmospheric pressure to create a 1000 psi drop across the clearance of the No. 6 piston with a 6.38mm land length and the No. 3 bore. The rotor was first rotated to record the force variation under low frequency dither. The force was observed to be 2.5 lbf to -0.1 lbf peak to peak (Fig. 75). The positive force means that the force transducer experienced tensile force and the negative force means that the transducer experienced compression. At high frequency dither, the recorded force increased to roughly 3.1 lbf peak to peak (Fig. 76).

The two ball valves in the hydraulic circuitry were closed to isolate lines connected to the test housing. The piston was inserted into the bore, and while the actuation mechanism applied dither motion to the piston, the piston never completely left the bore (Fig. 77). The downstream side of the piston was flushed with ether to remove oil as well as particles, so that the leakage fluid that was collected in the clean bottle for the particle size distribution analysis was not corrupted by unwanted residual particles.

The dither frequency was set at 0.057 Hz, and the two ball valves were opened to introduce contaminated fluid (25 mg/L of ACFTD full distribution). Immediately after the introduction of solid particles to the clearance, both the tension and compression force increased (Fig. 78). The force was about 3.5 lbf to -1.5 lbf peak to peak. As time elapsed, the peak to peak force decreased.

The dither frequency was raised to 1.24 Hz and the same test was

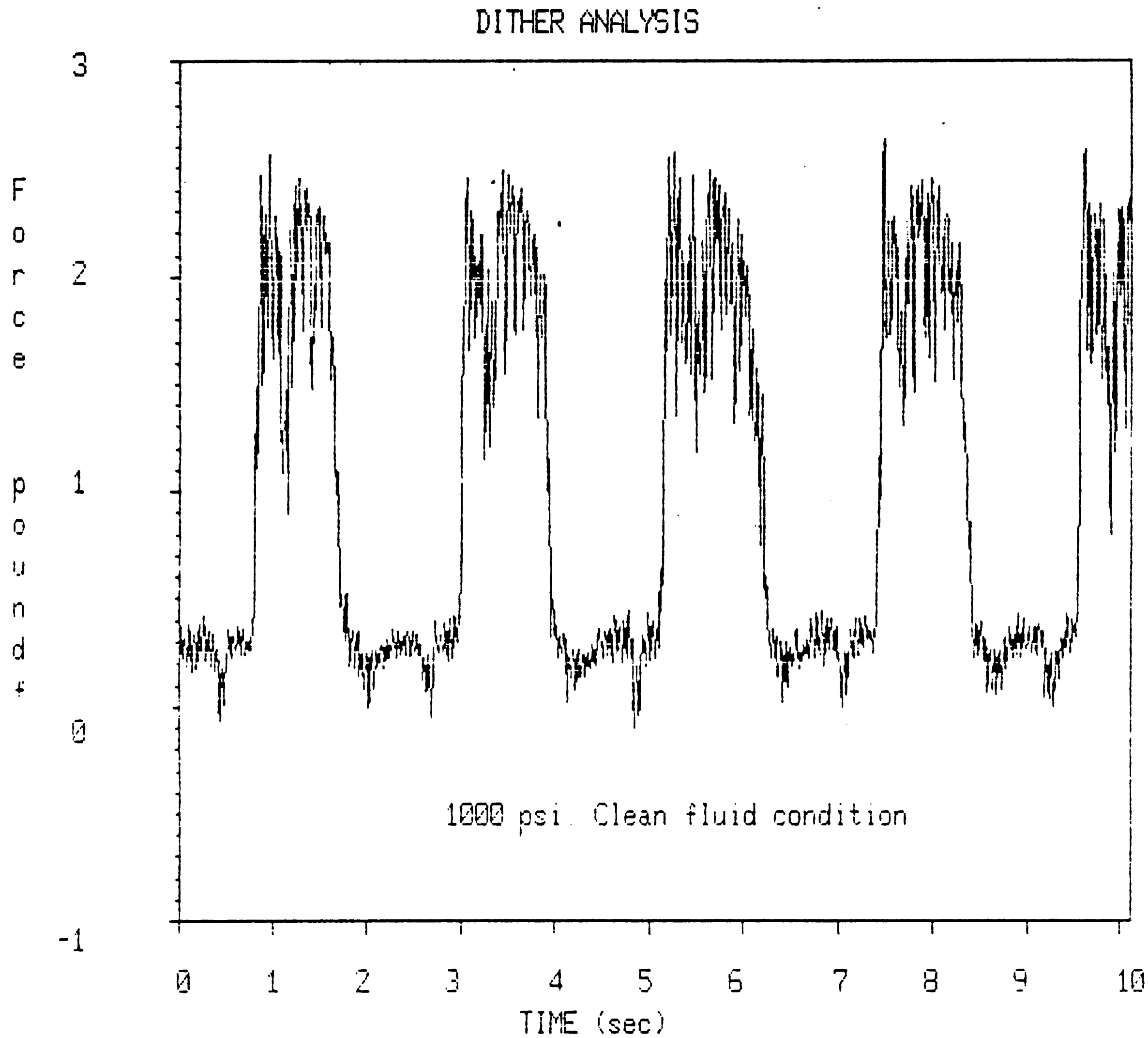


Figure 75. Dither Analysis (I).

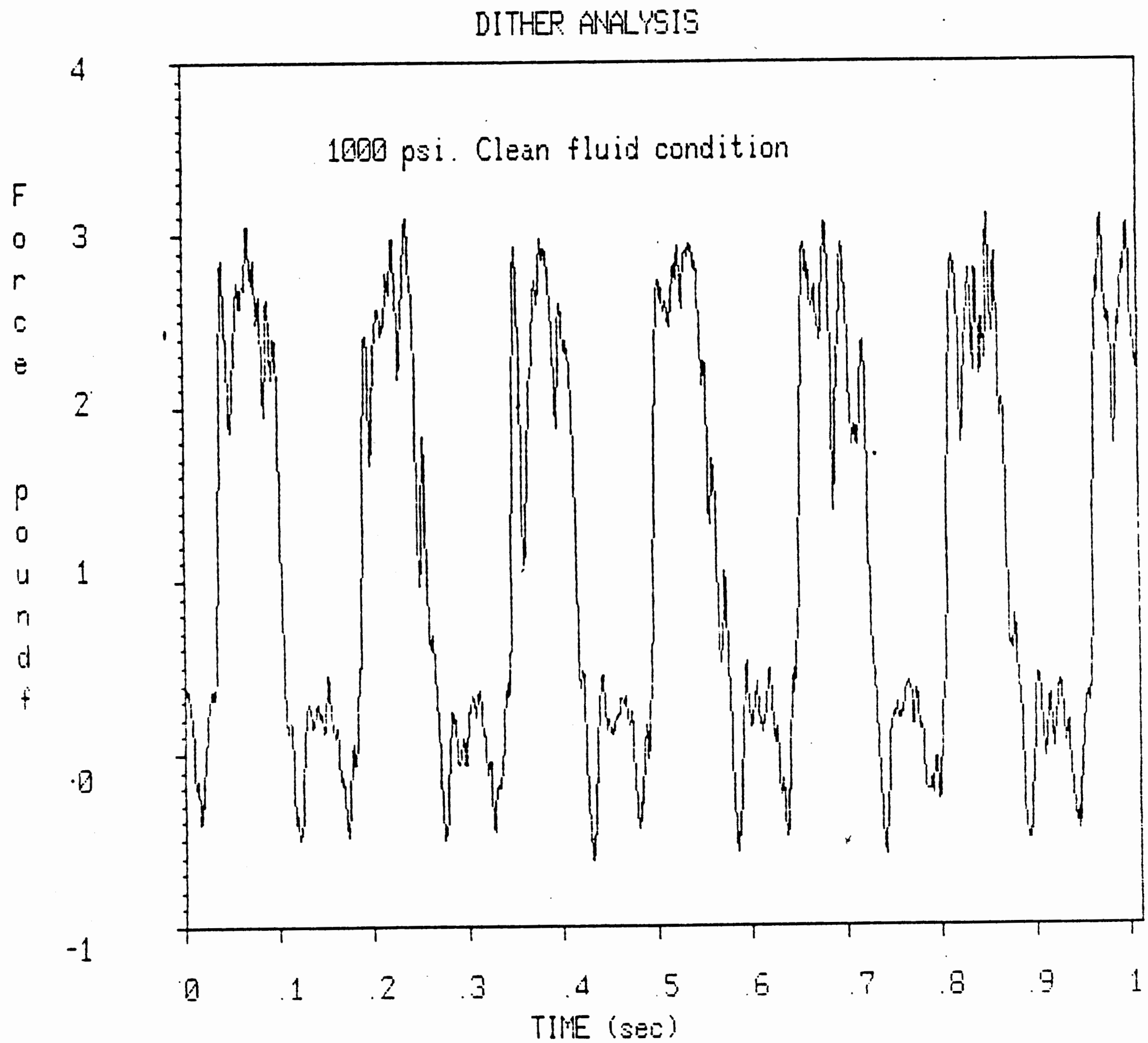


Figure 76. Dither Analysis (II).

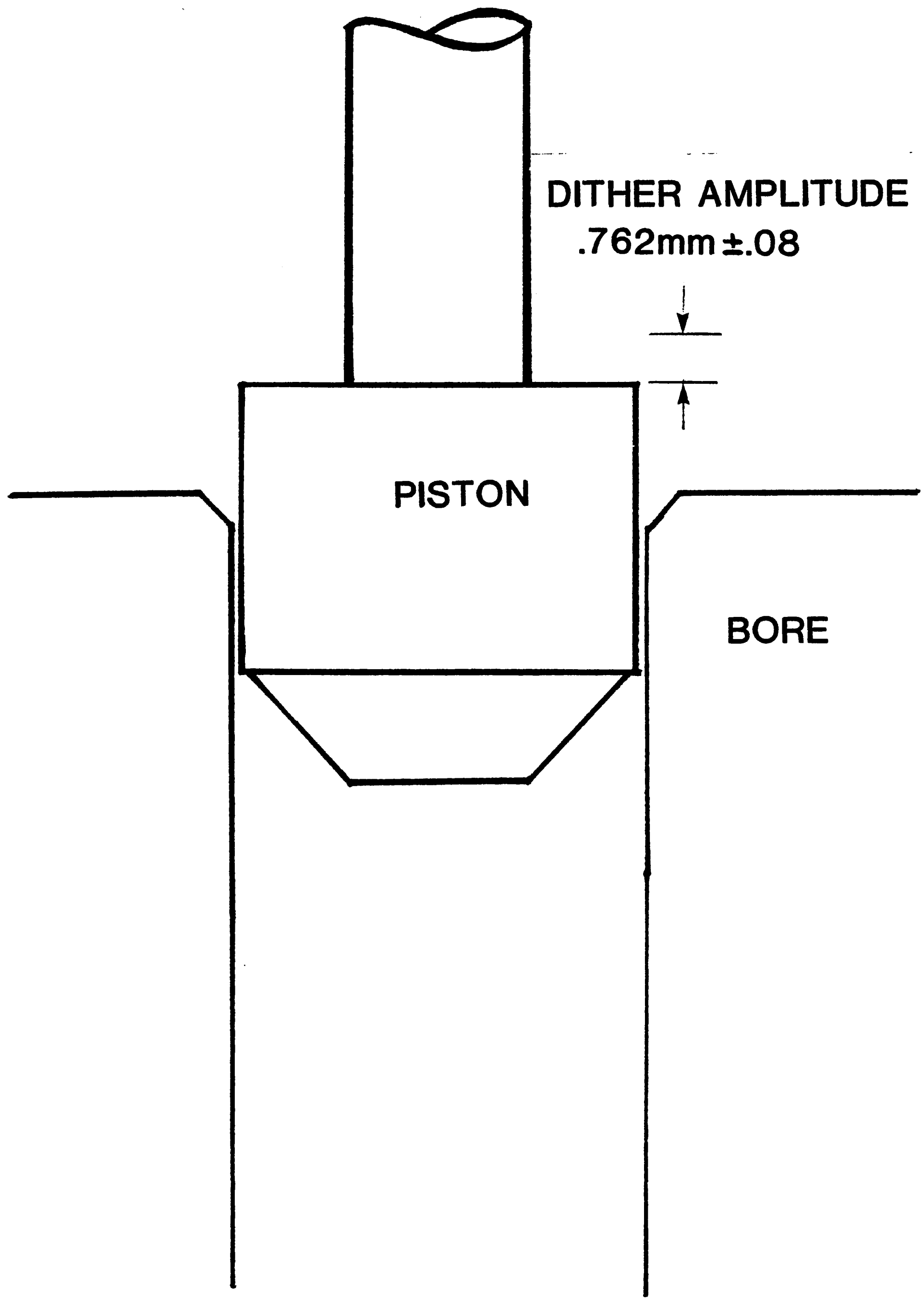


Figure 77. Dither.

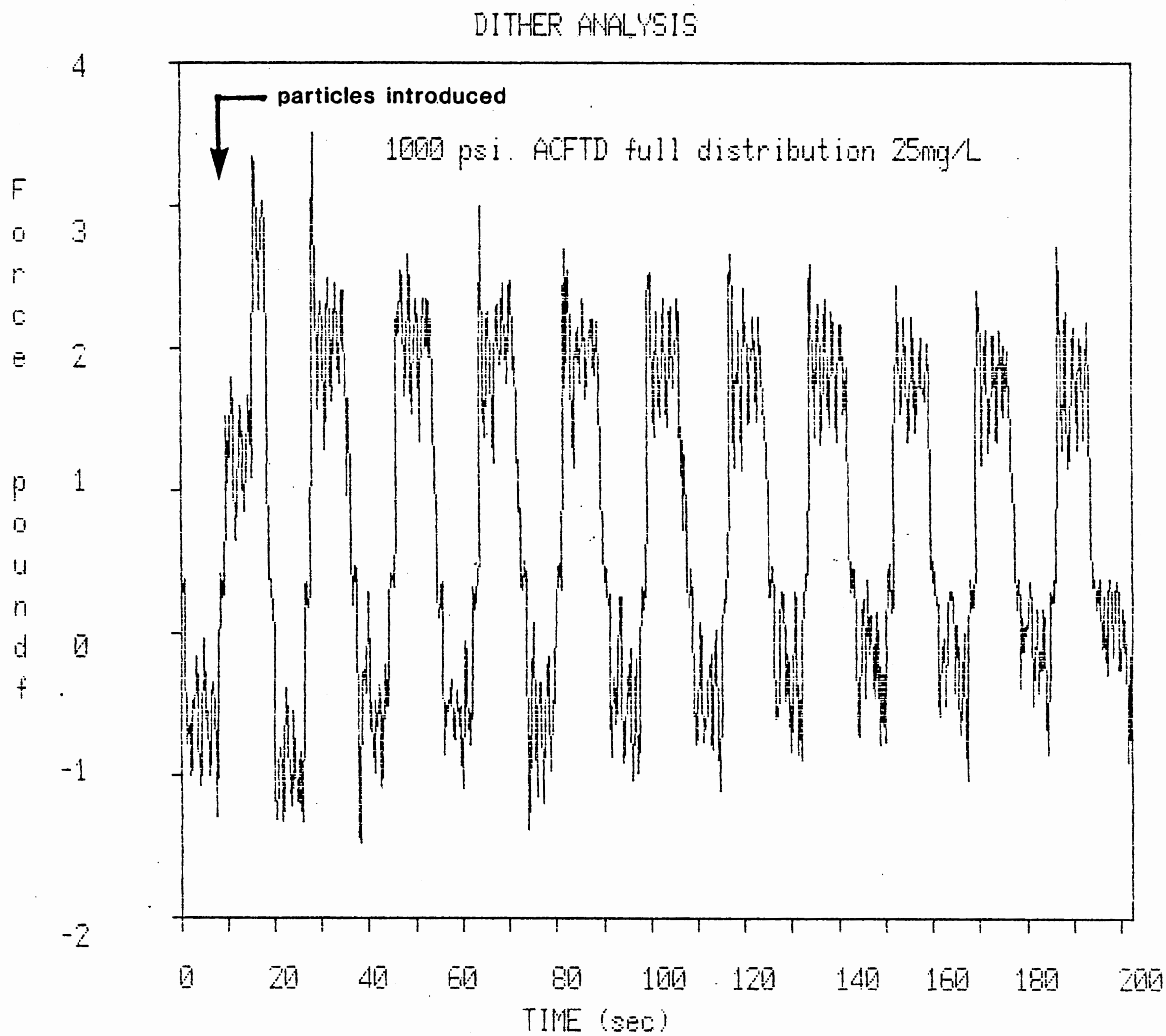


Figure 78. Dither Analysis (III).

repeated (Fig. 79). For approximately the first 20 seconds from the time that the ACFTD was introduced, the peak to peak force was affected and the maximum peak to peak force was increased to 4.5 lbf (3.5 lbf to ~~-0.1~~ -1.0 lbf). After that, the peak to peak force gradually decreased and remained at 2.5 lbf to -0.5 lbf. The dither frequency was further increased to 7.9 Hz (Fig. 80). The measured force was immediately increased to 3.8 lbf to -1.5 lbf, a difference of 5.3 lbf. The force due to solid particles lasted approximately 25 seconds after, which the force seemed to resume steady oscillation.

The difference between the upper peak and the lower peak force was plotted in Fig. 81. The peak to peak difference in the force under clean fluid showed gradual increase as the dither frequency increased. This is probably caused by the inertia of the bearing rod attached underneath the force transducer. On the other hand, the difference in the force under the contamination decreased at first, and then gradually increased. The difference between the force under clean fluid and contaminated fluid is the force needed to overcome the extra friction caused by particles entrained in the clearance. This extra friction decreased quickly at first as the dither frequency increased to 1.5 Hz, yet it did not show a diminishing trend when the dither frequency was raised above 1.5 Hz.

The downstream particle size distribution was analyzed from the leakage collected in the clean bottles. The silt Beta values were obtained and converted to filter efficiency versus particle size, as shown in Fig. 82. With a dither frequency of 0.057 Hz, the piston-bore assembly actually filtered particles better than the assembly without

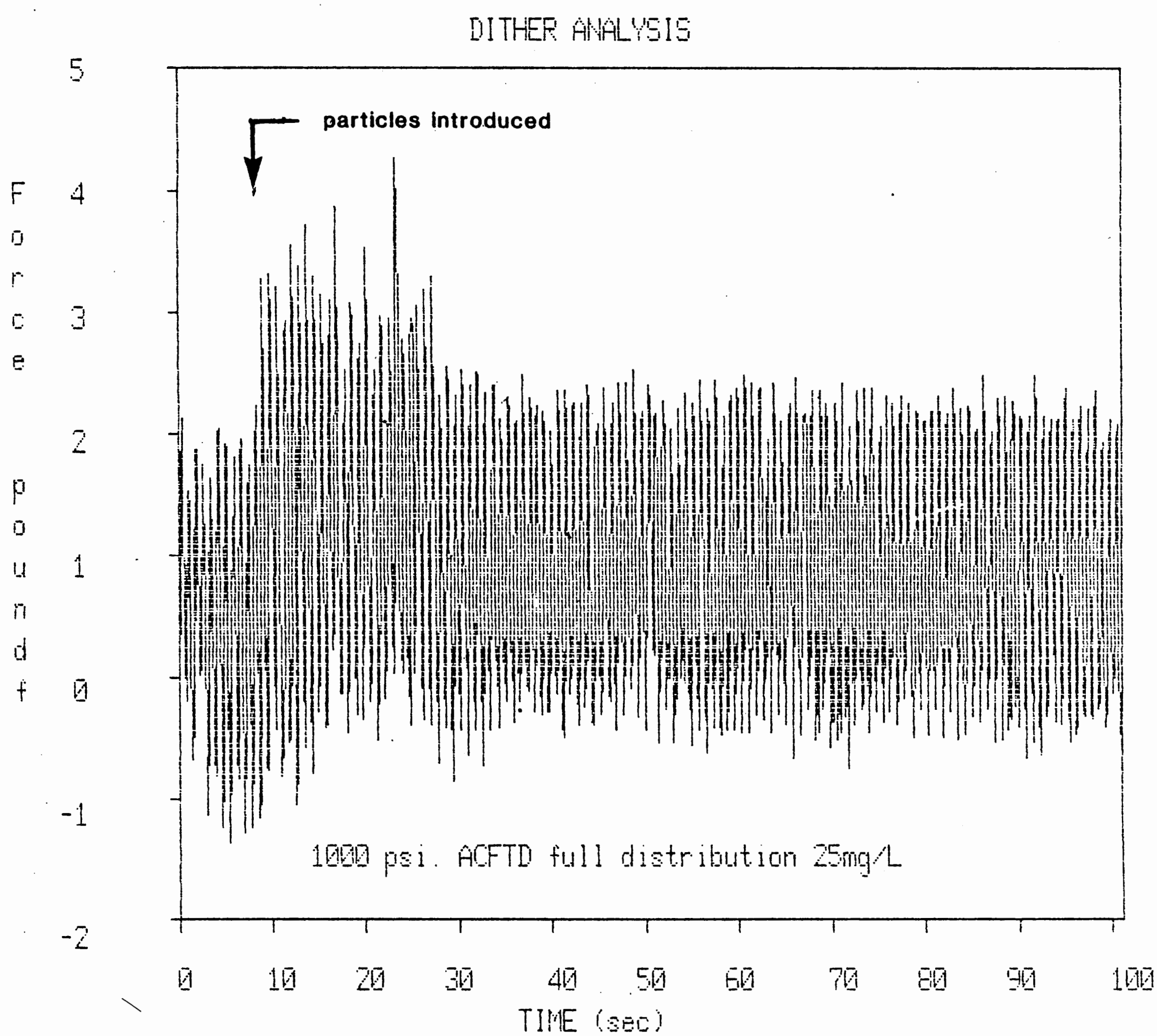


Figure 79. Dither Analysis (IV).

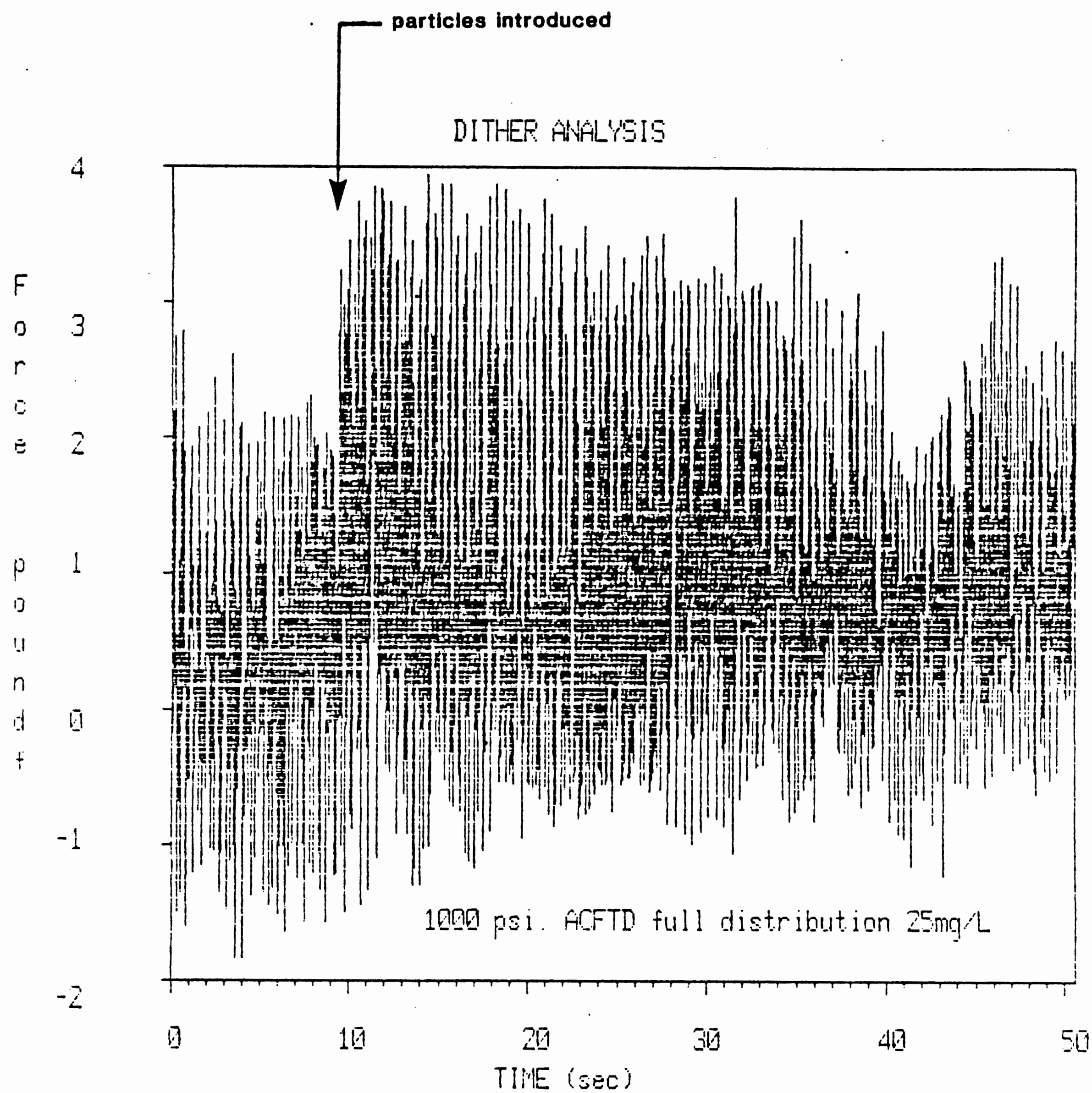


Figure 80. Dither Analysis (V).

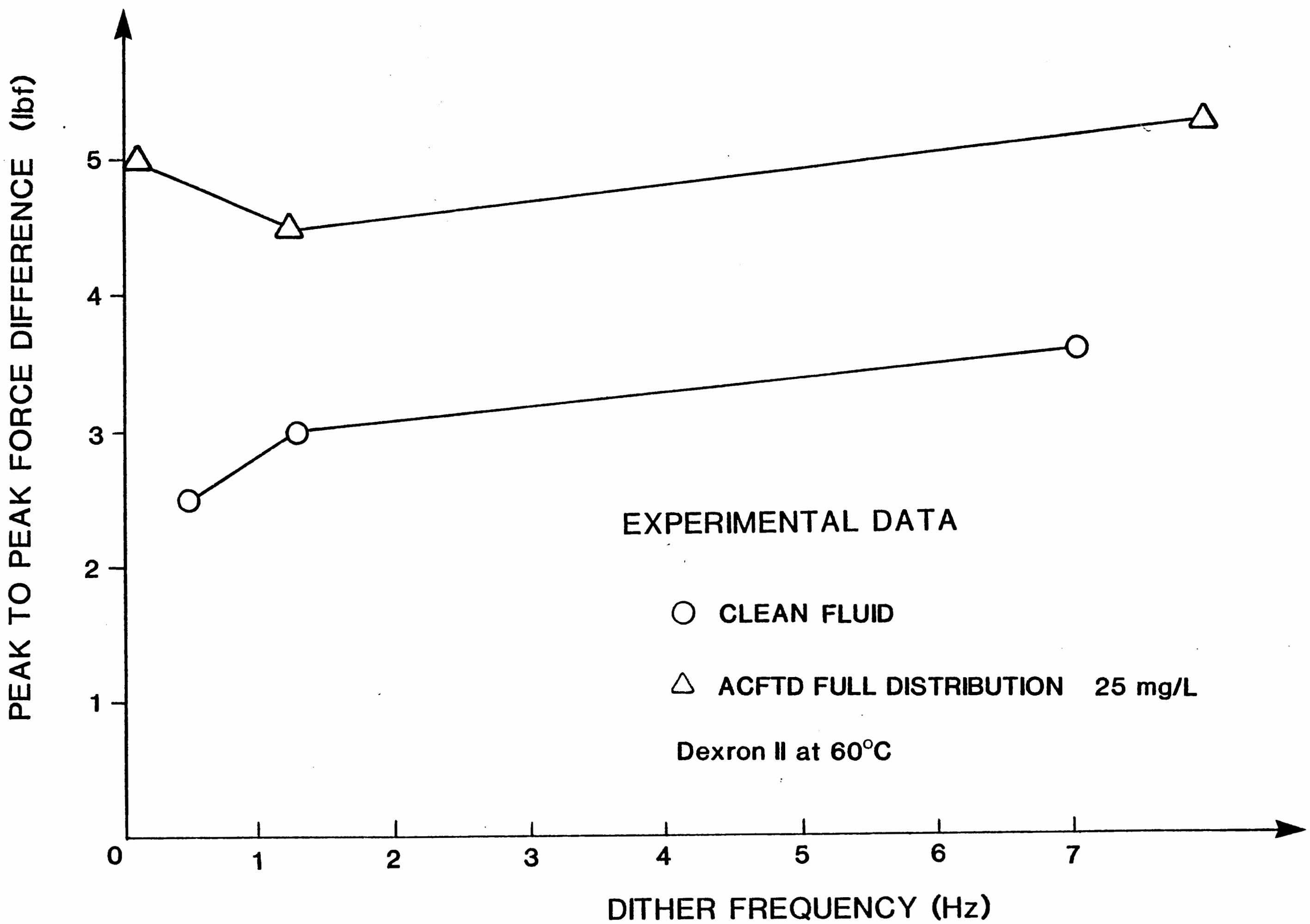


Figure 81. Friction Force Under Dither.

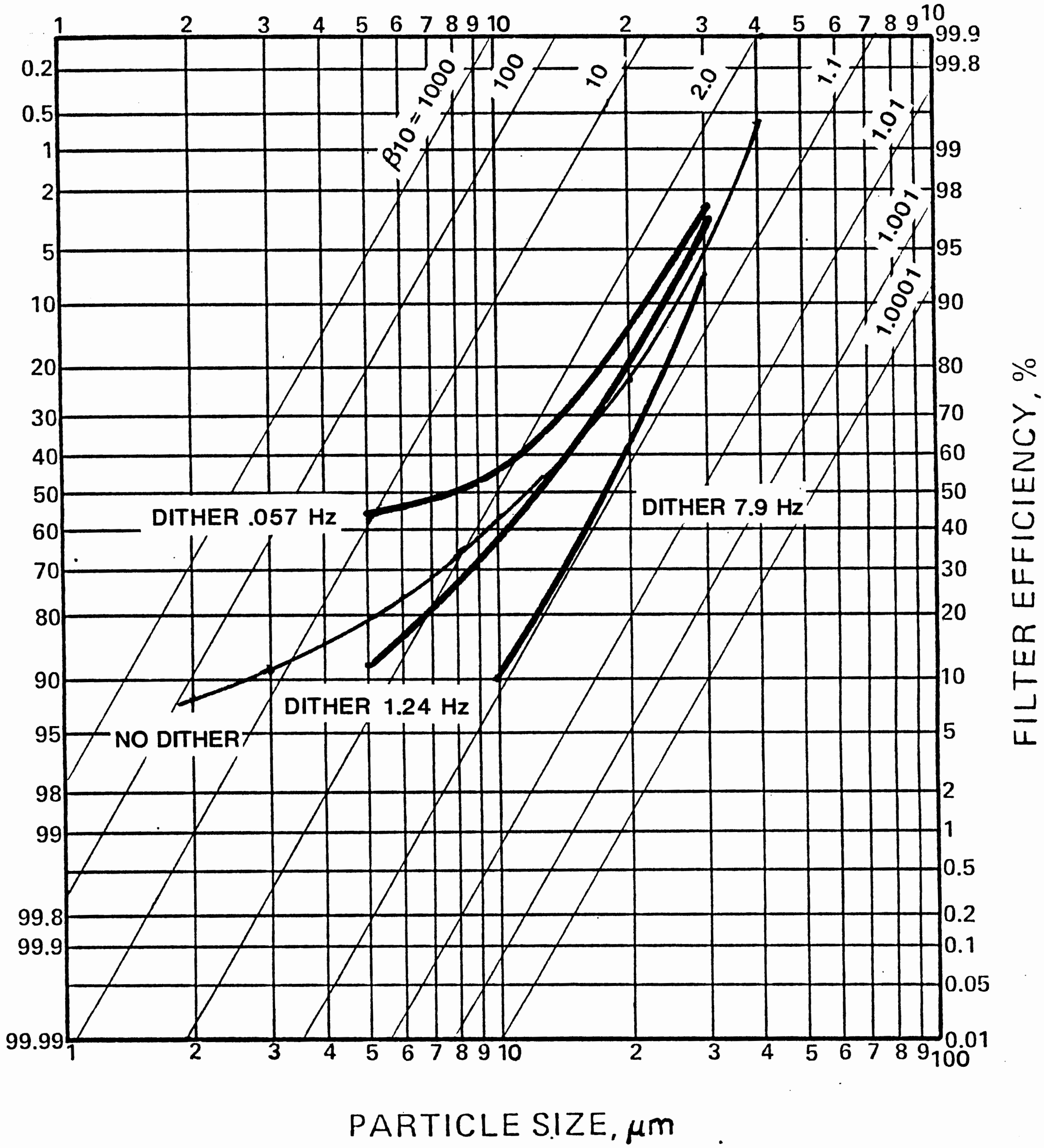


Figure 82. Effect of Dither Frequency on Filtration Efficiency.

dither. This was probably because the slow dither motion disturbed the cake formation to allow more leakage volume than the leakage under no dither. Also, the average cleanliness of the total collected leakage became cleaner than the leakage collected under no dither. Therefore, this leakage passing through the cake formation is a well filtered fluid, and so the time average downstream particle size distribution with dither showed a higher filtration efficiency than for a system without dither. However, as soon as the dither frequency exceeded 1.23 Hz, the dither motion of the piston became abrupt enough to destroy the possibility of initial cake formation or addition to the cake formation. The drastic efficiency degradation at fine particle size clearly demonstrates that there is less caking. At these high frequencies, the piston-bore assembly functioned as a poor filter. Hence, the filtration efficiency decreased as the dither frequency increased. These experimental results agree with the dither model presented in Chapter IV (Eq. 39).

The No. 23 piston was replaced by the No. 6 piston to investigate how hardness would affect the force required to actuate a piston. As Figure 83 shows, as soon as contaminated fluid was introduced into the test housing, the recorded force exhibited a "sticking" effect at the tension side with a higher peak value than for the force exhibited under a clean condition of the same assembly. At higher dither frequency, the force appeared to increase with elapsed time (Fig. 84). The peak to peak value of the force was difficult to assess with this piston-bore assembly because of unstable behavior of the system. This may be attributed to the severe wear taking place on the surface of the piston. After the

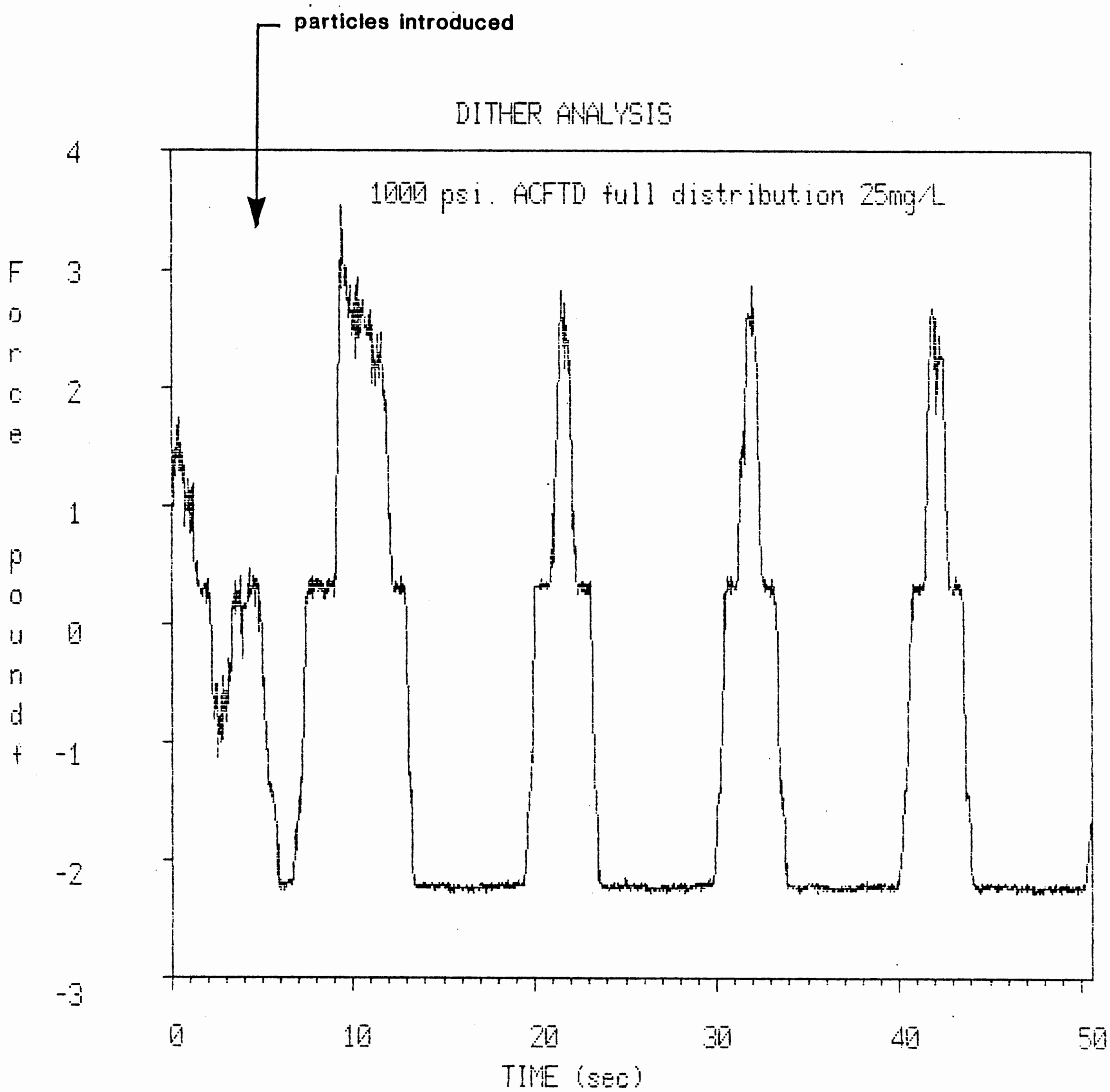


Figure 83. Dither Analysis (VI).

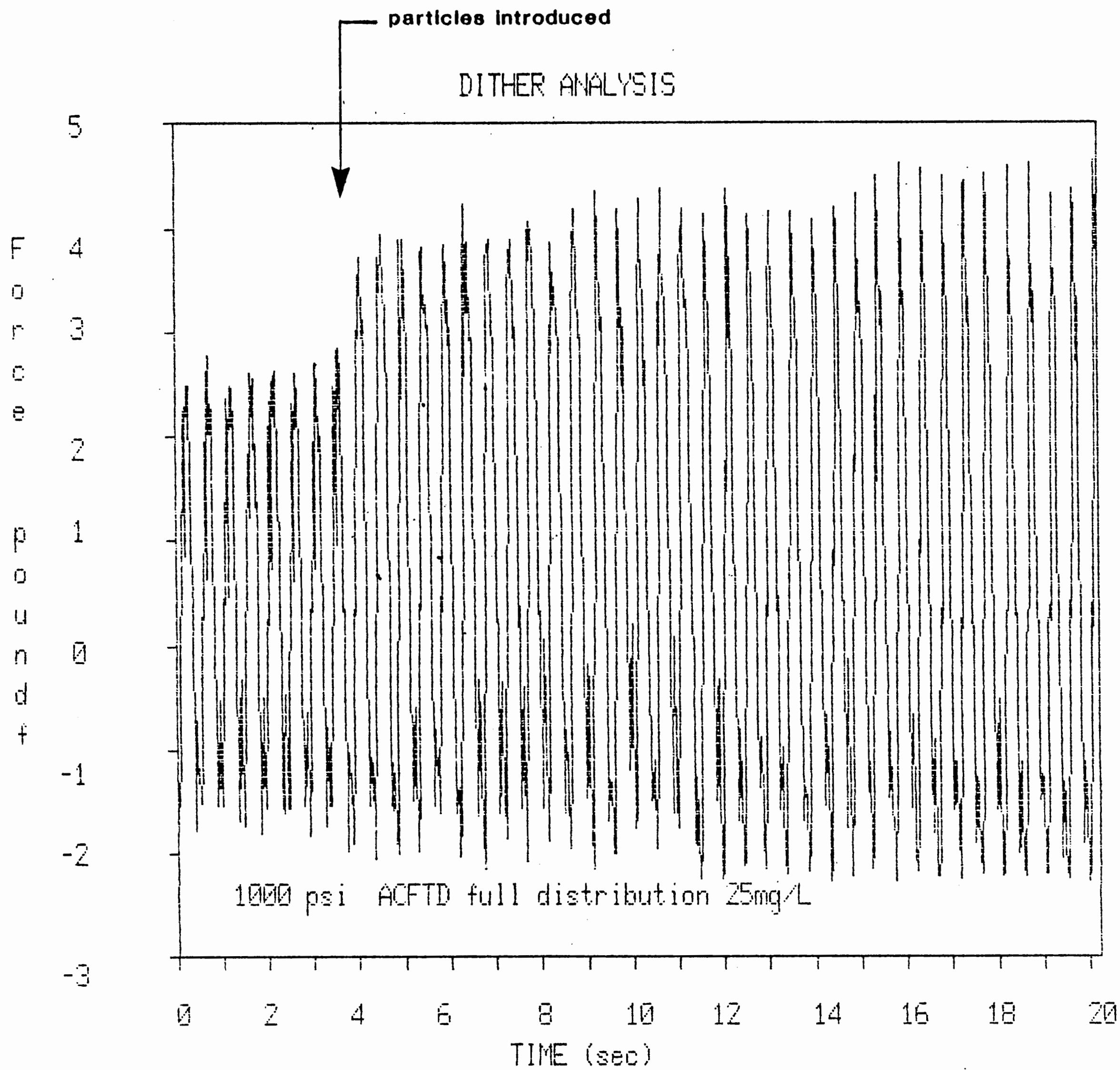


Figure 84. Dither Analysis (VII).

test, a band of wear all the way around the surface of both the No. 23 (Fig. 85) and No. 6 (Fig.86) piston was noted. Microscopic observation revealed that different wear was occuring at the upstream side compared with the downstream side surface of the piston. The difference in wear was vividly illustrated when the piston (No. ~~6~~ 23) is softer than ACFTD particles. There were many lateral lines created by the dither motion on the surface at the upstream side. This may have been caused by large particles indenting on the harder surface of the bore, cutting into the softer surface of the piston. In addition, there were many pits of various sizes that were created by solid particles under the influence of dither.

In contrast to the upstream side surface, the downstream side was less worn, and vertical lines caused by initial fabrication grinding were conspicuous, indicating minimal wear taking place. The observation of wear conditions at the upstream side indicates that larger particles which can bridge the clearance were stuck at the entrance region of the clearance and were causing severe wear. The silt process predicted the locally concentrated silt at the clearance entrance, and this upstream side wear on the piston surface clearly confirms the accuracy of that prediction. Furthermore, the wear shows that these large particles continued to stay at the entrance region and do not move downstream. This further indicates that particles which bridge the clearance at the entrance region significantly contribute to the seizure and provide extra surfaces for finer particles to silt and block the leakage path through the clearance.

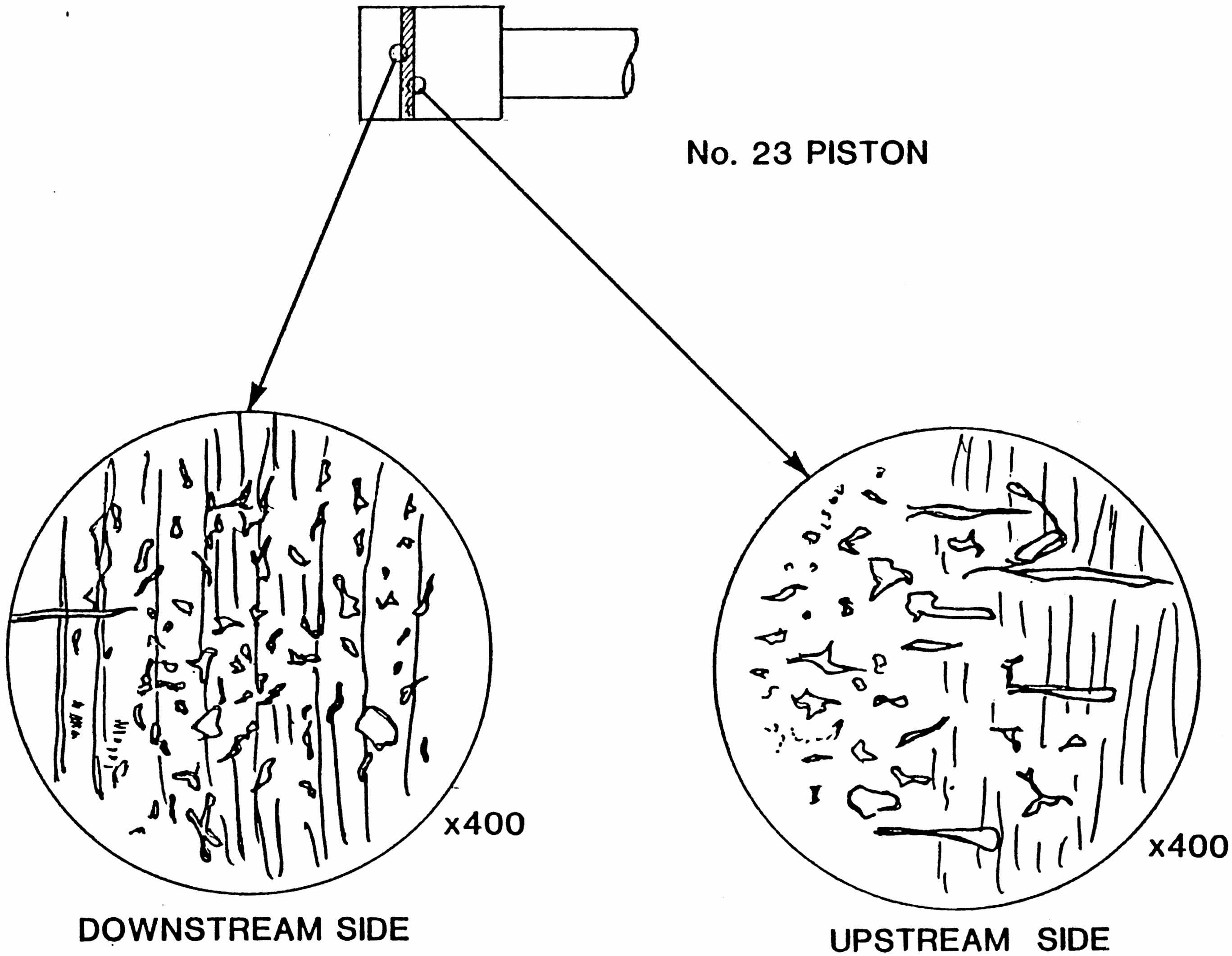


Figure 85. Microscopic Observation of Wear Caused by Dither (I).

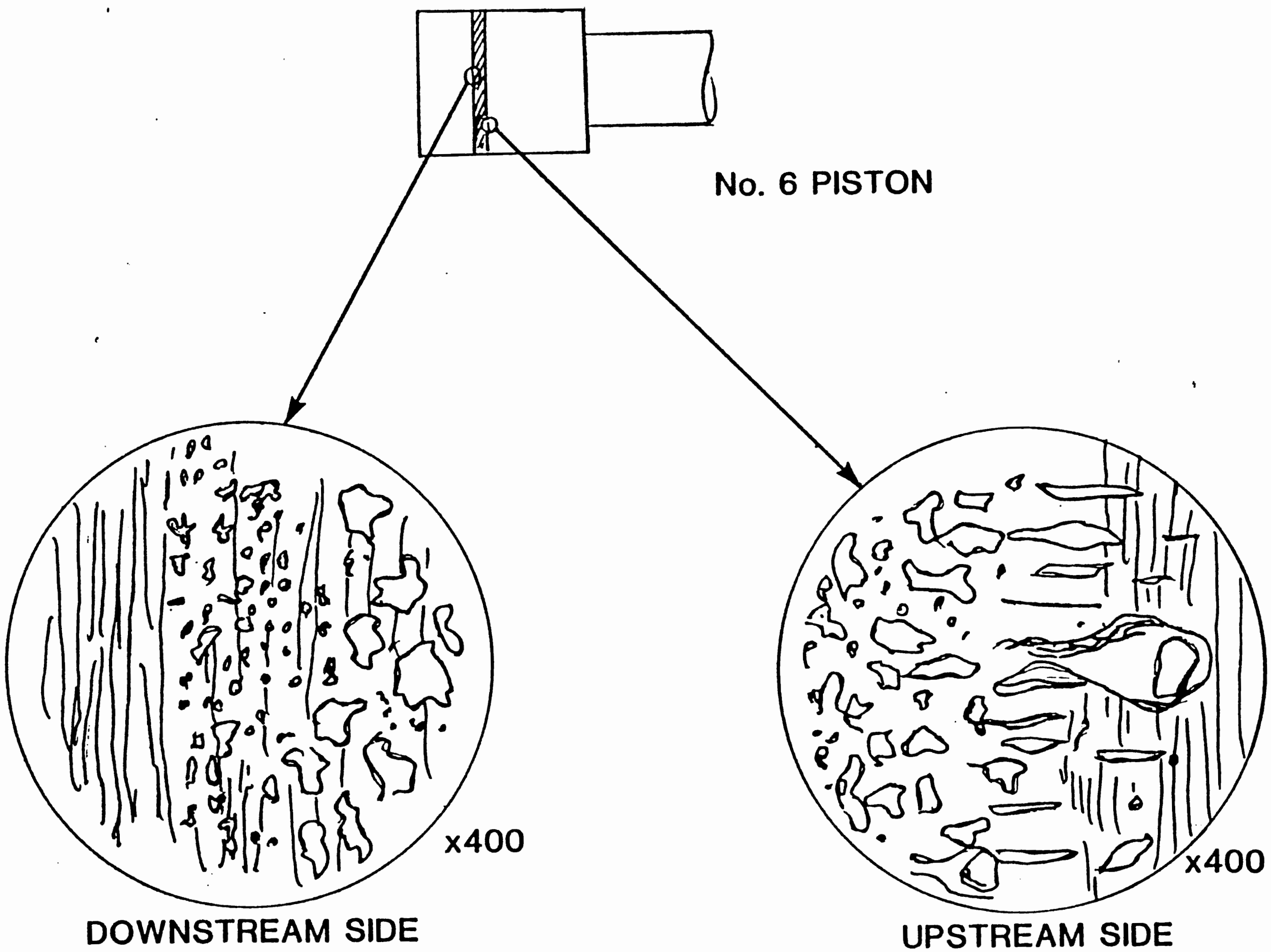


Figure 86. Microscopic Observation of Wear Caused by Dither (II).

When the piston material is softer than the particles, the serious wear on the surface of the piston discourages the application of dither. Furthermore, high frequency dither application, shown in Fig. 84, demonstrates the unpredictable and continuous uncontrollable increase in peak to peak friction force. This type of unpredictable force increase may cause catastrophic failures of the piston-bore assembly.

When the assembly materials were harder than the particles, however, very little wear was observed on the surface of the piston. The measured peak to peak force increase was stable and predictable, as presented in Fig. 81.

CHAPTER VII

SUMMARY AND CONCLUSIONS

The application of piston-bore assemblies varies from such mundane equipment as earth moving machines to as far reaching as robotics and space vehicles, and the sophistication of the assemblies has developed from the simple muscle of heavy equipment to the delicate pressure and flow control elements. Within the last decade, the control methods of these assemblies have achieved such a sophistication that they can now imitate the intricate muscle movements of living creatures. However, even with such advanced technology, particulate-induced seizure of the piston-bore assembly has hardly been understood. This lack of understanding has led to catastrophic failures and accidents in industry. Design engineers, in the past, have had no means of understanding seizure. Thus, the only solution to seizure they have had has soley been through the experience of engineers and through trial and error methods. The understanding of seizure, however, can alleviate possible catastrophic failures by allowing design engineers to predict the occurence and magnitude of the force required to break seizure. This dissertation provides the essential tools that engineers need for predicting the force required to break seizure and thus maintain normal operation of piston-bore assemblies.

The verification tests of the three theoretical models developed in this dissertation uncovered important evidence as to the mechanisms that cause particulate-induced seizure. To analyze how particles in the clearance cause seizure, the silt process had to be understood so that the number and size of particles in the clearance prior to seizure could be known. Thus, the silt process model laid the foundation of this study.

The experimental results of the silt process model revealed that the smaller clearance of a piston-bore assembly sieves finer particles, and the long land length increases the retention (dirt holding) capability by reducing the average shear rate at the wall of the clearance. The effect of shear rate was clearly demonstrated by the leakage measurement. While the initial shear rate is low, particles smaller than the clearance size are loosely captured in the assembly clearance and effectively reduce the leakage; however, at the high initial shear rate, these particles are either flushed away or moved aside from the leakage path until the leakage reduces and approaches the low shear rate.

The verification test of the particulate-induced seizure model confirmed the integrity of the silt process model and established the means for predicting the force required to break seizure. It was found that the particles must bridge across the clearance in order to contribute to the seizure force. Furthermore, the controlling stress of the force required to break seizure is the ultimate stress of the softest material of the three interacting bodies.

After establishig the method for predicting the magnitude of the seizure force, the dither model was developed to investigate the effect of dither on the filtration efficiency of the assembly. This experiment confirmed that the higher the dither frequency, the lower the filtration efficiency. This result indicates that the vibrational effect caused by dither enhanced the dislodgement of fine particles smaller than the clearance. When the particles are softer than the assembly, the application of dither to reduce the seizure force is appropriate; however, when the particles are harder than the assembly material, the application of dither may increase the seizure force due to wear which takes place on the assembly surface.

To the author's knowledge, this dissertation has demonstrated for the first time that the occurence and magnitude of seizure force can be predicted. The combination of the silt process model and the particulate-induced seizure model gives design engineers the capability to develop piston-bore assemblies which are seizure-free. It is firmly believed that this study advances the understanding of seizure and encourages extensive application of these models in the area of fluid power engineering.

REFERENCES

[1] Sweeny, D.C. "Preliminary Investigation of Hydraulic Lock." Engineering, 172 (1951), 513-516 & 580-582.

[2] Stringer, J.E.C. "Hydraulic Lock: Another Explanation." Engineering, 183 (1952), 509-511.

[3] Mannam, J. "Hydraulic and Dirt Lock in Piston Type Control Valves." University of Adelaide, South Australia, Paper from the Dept. of Mechanical Engineering, p. 15-31.

[4] Dransfield, P., D.M. Bruce, and M. Wadsworth. "A General Approach to Hydraulic Lock." The Basic Fluid Power Research Journal (BFPR), (1967), 1-15.

[5] Dransfield, P. "Hydraulic Lock With Single-Land Piston." The BFPR Journal, (1967), Appendix E.

✓ [6] Dransfield, P. "Leakage Flow Rate Past Pistons of Oil Hydraulic System Components." The BFPR Journal, (1967), Appendix D.

[7] Whiteman, K.J. "Hydraulic Lock at High Pressure." The Engineer, April 12 (1957), 554-557.

[8] Kamijo, K., H. Kusama, and T. Sasada. "Hydraulic Lock on Spool Valve." Bulletin of the Tokyo Institute of Technology, No. 100 (1970), 45-90.

✓ [9] Laurenson, I.T., and J.P. O'Donoghue. "Silting of Fine Flow Passage." Fifth International Fluid Power Symposium, Paper D2 (1978), 13-22.

✓ [10] Surjaatmadja, J.B., and E.C. Fitch. "The Characteristics of Contaminant Lock in Fluid Components--Particle--Induced Friction." Oklahoma State University, Fluid Power Research Center, Paper No. 76, October (1976), parts 1-4.

[11] Iyenger, S.K. "Effect of Particulate Contaminants on Break-Out and Actuating Forces in Spool Valves--A Case Study." Oklahoma State University, Fluid Power Research Center, Paper No. , October (1976), 76-77.

[12] Inoue, R. "Contaminant Lock in Spool-Type Directional Control Valves Parts 1-4: Experimental Exposition." The BFPR Journal, 13 (1980), 163-188.

[13] Nikitin, G.A., and S.V. Chirkov. "The Effect of Fluid Contamination on the Operating Reliability of Aircraft Hydraulic Systems." Report No. FTD-MT-24-463-69, April, 1970.

[14] Nair, K.S. "Dither--The Mysterious Killer of Friction in Fluid Power Valves." The BFPR Journal, 15 (1982), 35-42.

[15] Ito, T. "Particulate-Induced Seizure of Piston-Bore Assemblies." The Tribological System Fluid Journal, 7 (1986), 36-43.

[16] Mishina, H., and T. Sasada. "Observation of Micro-Structure in Seized Portion and Mechanism of Seizure." Transaction of ASME, 108 (1986), 128-133.

[17] Bensch, L.E. "Contamination Levels." The BFPR Journal, (1970), Supplemental Report No. 70-1.

[18] Cole, F.W. "Gravimetric Conversion of Particle Count Data." Paper presented to the Contamination and Friction Panel of SAE A-6 Committee, February, 1968.

[19] Bensch, L.E., and E.C. Fitch. "A Cleanliness Level Criterion for Particulate Matter in Liquids." Paper presented at the Advances in Filteration Symposium for the American Chemical Society Industrial & Engineering Chemistry Division, September, 1971.

[20] Fitch, E.C. An Encyclopedia of Fluid Contamination Control. Fluid Engineering Service Inc., Stillwater, Oklahoma, 1980.

[21] Ito, T., I. Khalil, and I.T. Hong. "Abrasive Wear Under Hydrodynamic Lubrication: Parts 1-2." The Fire Resistant Hydraulics Journal, 5 (1986), 125-136.

[22] Multipass Filter Test Data--Data Log Book. Oklahoma State University, Fluid Power Research Center, 1982-1987.

[23] Fitch, E.C., and R.K. Tessman. "Controlling Contaminant Wear Through Filtration." Paper P75-9 presented at the Ninth Annual Fluid Power Research Conference at Oklahoma State University, Stillwater, Oklahoma, 1975.

[24] Bensch, L.E. The Influence of Electrostatic Charge on the Filtration of Hydraulic Fluids by Fibrous Filters." Oklahoma State University, Doctoral Dissertation, 1977.

[25] "The Development of Contaminant Diagnostic Monitor." Final Report prepared of the U.S. Department of Energy, Pittsburg Mining Center, Contract No. DE-AS05-78-ET-12204, 1983.

✓ [26] Hong, I.T. and E.C. Fitch. "Tribological Effects of Obliteration on Sliding Valve Characteristics--Part I: Physics of Obliteration." The Tribological System Fluid Journal, 7 (1986), 134-141.

APPENDIX A

SEIZURE MODEL SIMULATION PROGRAM

```
100 '*******************************************************************
120 '
140 '   THIS PROGRAM VERIFIES THE INTEGRITY OF THE THREE PROPOSED
160 '   MODEL.  THE DETAIL IS EXPLAINED AFTER SUBROUTINES ( LINE# 10700-).
180 '
200 '           PROGRAMMED BY TOKUNOSUKE ITO
220 '
240 '*******************************************************************
260 '
280     CLS
300     CLEAR:CLEAR 1000
320     DIM CTABLE(180)
340     DIM HL(3),FM(3),FA(3),BFM(3),BFA(3),ANGLE(3),CAREA(3)
360     IBM = 0
380     IF IBM = 1 THEN GOSUB 860 ELSE GOSUB 480
400     GOTO 1140
420 '
440 ' -------- TANDY 2000 --
460 '
480     KEY 11,"EDIT "
500     KEY 12,"AUTO "
520     LPRINT CHR$(27);CHR$(14):'              CONDENSED CHARACTER
540     DIMC = 0:'      To avoid the duplicate dif.
560     WIDTH LPRINT 117:'                      SET PRINTER, 117chars/line
580     YOFFS = 350
600     SCS = 4
620     AY = 330:'              MAX. WIDTH OF YAXIS
640     YKON = (22-2)/(8-330)
660     YOFF = 2
680     YSET = 180
700     KB1 = 28
720     KB2 = 29
740     KB3 = 30
760     KB4 = 31
780     RETURN
800 '
820 ' ------- IBM HIGH RES. SETTING ---
840 '
860 '
880     YOFFS = 180
900     SCS = 2
920     AY = 170
940     YKON = (2-23)/(170-0)
960     YOFF = 0
980     YSET = 85
1000     KB1 = 54:'      AT KEY CODE
1020     KB2 = 52
1040     KB3 = 56
1060     KB4 = 50
1080 '
1100     RETURN
1120 '
1140     GOTO 19040:'JUMP TO MAIN ROUTINE
1160 '
1180 '***********  HIGH RESOLUTION GRAPHICS ***************
1200 '
1220     SCREEN (SCS):'            HIGHEST RESOLUTION
1240     KEY OFF:'ELIMINATE FUNCTION KEY DISPLAY
1260     RETURN
1280 '
```

```
1300 '************** END OF HIGH RESOLUTION (TURN_ON)**************
1320 '
1340 '
1360 '
1380     LINE (XD1,YD1)-(XD2,YD2),K AND &HF0F
1400 '=================================
1420 '
1440 '   SUBROUTINE
1460 '
1480 '---------------------------------
1500 '
1520     XD1=X1+90:YD1=YOFFS-Y1
1540     XD2=X2+90:YD2=YOFFS-Y2
1560     LINE (XD1,YD1)-(XD2,YD2),CLR
1580     RETURN
1600 '
1620 '
1640 '****  BOX DRAWING *****************
1660 '
1680     CLR=1:'                 COLOR CODE
1700     X1=0:Y1=0
1720     X2=510:Y2=0:GOSUB 1500
1740     X1=510:Y1=AY:GOSUB 1500
1760     X2=0:Y2=AY:GOSUB 1500
1780     X1=0:Y1=0:GOSUB 1500
1800     RETURN
1820 '
1840 '************************************
1860 '
1880 '*************** MAX. & MIN. ***********************8
1900 '
1920 '@@@@@@  X-AXIS FIRST @@@@@@@
1940 '
1960     IF I=SS(J) THEN K=S(I):PX=X(I,K):XMAX=PX:XMIN=PX
1980     FOR K=S(I) TO Z(I)
2000             PX=X(I,K)
2020             IF XMIN > PX THEN XMIN = PX
2040             IF XMAX < PX THEN XMAX = PX
2060     NEXT K
2080 '
2100 '-------- Y-AXIS  -----------------------------
2120 '
2140     IF I=SS(J) THEN K=S(I):PY=Y(I,K):YMAX=PY:YMIN=PY
2160     FOR K=S(I) TO Z(I)
2180             PY=Y(I,K)
2200             IF YMIN > PY THEN YMIN=PY
2220             IF YMAX < PY THEN YMAX = PY
2240 '           PRINT"YMAX  ";YMAX,"YMIN  ";YMIN
2260 '
2280    NEXT K
2300 '
2320     RETURN
2340 '
2360 '****************** END OF MAX. & MIN. **************
2380 '
2400 '******** TITLE & AXIES LABEL CONTROL ****
2420 '
2440 '
2460 '   T$="TEST GRAPHING"
2480 '   XL$="TIME (sec)"
```

```
2500 '   YL$="OUTPUT PARAMETER"
2520  '
2540     IF LEN(T$(J))>80 THEN T$(J)=LEFT$(T$(J),80)
2560     IF LEN(XL$(J))>80 THEN XL$(J)=LEFT$(XL$(J),80)
2580     IF LEN(YL$(J))>24 THEN YL$(J)=LEFT$(YL$(J),24)
2600     N= INT((80-LEN(XL$(J)))/2):'   MAINTAIN THE ORDER FOR IBM
2620     LOCATE 25,N:PRINT XL$(J)
2640     N= INT((80-LEN(T$(J)))/2)
2660     LOCATE 1,N:PRINT T$(J)
2680    M=0
2700    NM=INT((24-LEN(YL$(J)))/2)
2720    FOR N=NM TO LEN(YL$(J))+NM
2740    M=M+1
2760     LOCATE N,1:PRINT MID$(YL$(J),M,1)
2780     NEXT N
2800     RETURN
2820 '
2840 '******* END OF TITLES ******************
2860             SDELTA=INT((BASE + DELTA * F)*10^(POWER+4))/(10^(POWER+4))
2880 '********  TIC MARK FOR X-AXIS ***********
2900 '
2920 '    XMAX=3.2:XMIN=-1.6
2940    CLR=1
2960     AX=510:'                  WIDTH OF GRAPHICS
2980     POWER=0:'        POWER INCREMENT TO FIND THE SPACE BETWEEN XMAX,XMIN
3000     IF XMAX >= 0 THEN SIGNMAX=1 ELSE SIGNMAX=-1
3020     IF XMIN >= 0 THEN SIGNMIN=1 ELSE SIGNMIN=1:'**** NOTE INT(-1.1)=-2
3040     LX=INT(XMAX*10^POWER +.5*SIGNMAX)
3060     SX=INT(XMIN*10^POWER + .5*SIGNMIN)
3080     DE=LX-SX
3100 '   PRINT"DE ";DE,"LX ";LX,"SX ";SX,"POWER  ";POWER
3120     IF DE < 2 THEN POWER=POWER + 1:GOTO 3040
3140     IF DE > 20 THEN POWER=POWER - 1:GOTO 3040
3160     IF XMAX < LX*10^(-POWER) THEN LX=LX-1
3180     IF XMIN > SX*10^(-POWER) THEN SX=SX+1
3200     DE=LX-SX:'PRINT "DE ";DE;"POWER  ";POWER;"SX  ";SX
3220     DELTA = 10^(-POWER)
3240     BASE=SX*DELTA
3260     FOR F=0 TO DE
3280             SDELTA=INT((BASE + DELTA * F)*10^(POWER+4)+.5)/(10^(POWER+4))
3300            '***** GOOD ALGORITHM ******
3320            '
3340            S$=STR$(SDELTA)
3360            CK$=RIGHT$(S$,4)
3380            IF LEFT$(CK$,1)<>"E" THEN SDELTA=VAL(LEFT$(S$,5)):GOTO 3460
3400                    EX=VAL(RIGHT$(CK$,2))
3420                    IF MID$(CK$,2,1)="+" THEN SDELTA=VAL(LEFT$(S$,5))*10^EX
3440                    IF MID$(CK$,2,1)="-" THEN SDELTA=VAL(LEFT$(S$,5))*10^(-E
X)
3460 '
3480             X1=INT(AX/(XMAX-XMIN)*(SDELTA - XMIN))
3500             X2=X1:SMALL1=X1
3520             Y1=0:Y2=-5
3540             'PRINT X1,X2,Y1,Y2
3560             GOSUB 1500
3580             IF DE <=10 THEN 3640
3600             JUDGE= INT(F/2)
3620             IF F <> JUDGE*2 THEN 3800
3640             LOCATE 24,INT(X1/8)+11:PRINT ;SDELTA;:'       LABELING
3660             IF  F < 1 OR DE > 5 THEN  3800:'             FINER RESOLUTION
```

```
3680                            DIFF=SMALL1-SMALL0
3700                    FOR N=1 TO 9
3720                            X1=SMALL0 + INT(DIFF/10*N+.5):X2=X1
3740                            Y1=0:Y2=-2
3760                            GOSUB 1500
3780                    NEXT N
3800            SMALL0=SMALL1
3820     NEXT F
3840 '
3860    IF DE > 5 THEN GOTO 4340
3880 '==== UPPER END TIC MARK CONTROL =====
3900 '
3920    NUM=0
3940    NUM=NUM+1
3960    X1=SMALL1 + INT(DIFF/10*NUM+.5)
3980 '  PRINT"X1  ";X1;" SMALL1  ";SMALL1;" SMALL  ";SMALL;" NUM  ";NUM
4000    IF X1 >= AX THEN 4160
4020    X2=X1
4040    Y1=0:Y2=-2
4060    GOSUB 1500
4080    GOTO 3940
4100 '
4120 '==== LOWER END TIC MARK CONTROL =====
4140 '
4160    SMALL0=INT(AX/(XMAX-XMIN)*(BASE-XMIN))
4180    NUM=0
4200    NUM=NUM+1
4220    X1=SMALL0-INT(DIFF/10*NUM+.5)
4240    IF X1 =< 0 THEN GOTO 4340
4260    X2=X1
4280    Y1=0:Y2=-2:'                       REDUNDANT
4300    GOSUB 1500
4320    GOTO 4200
4340     RETURN
4360 '
4380 '************** END OF X-AXIS TIC MARK ********************
4400 '
4420 '******************** TIC MARK FOR Y AXIS ****************
4440 '
4460 '   YMAX=2.25:YMIN=-1.1:'              DUMMY VALUES
4480    CLR=1
4500    POWER=0:'                          INITIALIZATION OF POWER
4520     '
4540 '
4560    LY=INT(YMAX*10^POWER + .5)
4580    SY=INT(YMIN*10^POWER)
4600    DE=LY-SY
4620    IF DE < 2 THEN POWER = POWER + 1:GOTO 4560
4640    IF DE > 20 THEN POWER = POWER - 1:GOTO 4560
4660    IF YMAX > LY*10^(-POWER) THEN LY=LY+1
4680    IF YMIN < SY*10^(-POWER) THEN SY=SY-1:'         THIS WILL BE RARE
4700    DE=LY-SY
4720    DELTA=10^(-POWER):'                THIS IS EQUIVALENT TO 1 UNIT IN GRAPH
4740    BASE=SY*DELTA:'                    THIS IS THE BOTTOM VALUE OF Y-AXIS
4760    YDMAX=LY*10^(-POWER):YDMIN=SY*10^(-POWER):'      Y HAS BOUNDARY LARGER TH
AN YMAX,YMIN
4780    FOR F=0 TO DE
4800            SDELTA=INT((BASE + DELTA * F)*(10^(POWER+4))+.5)/(10^(POWER+4)):
'       INCREMENT EACH UNIT FROM BOTTOM
```

```
4820            Y1=INT(AY/(YDMAX-YDMIN)*(SDELTA-YDMIN))
4840            Y2=Y1:SMALL1=Y1
4860            X1=0:X2=-5
4880            GOSUB 1500:'                    DRAW LINE
4900            IF DE <= 10 THEN 4960:'        IF INCREMENT IS WITHIN 10,WRITE N
UMBER DIRECTLY
4920            JUDGE=INT(F/2):'               THIS IS TO WRITE EVERY OTHER TIC
MARK #
4940            IF F <> JUDGE*2 THEN 5140
4960            POSI= 23+INT(YKON*(Y1-YOFF))
4980             LOCATE POSI,4:PRINT SDELTA
5000          ' PRINT "F  ";F;" POSI  ";POSI;" Y1  ";Y1
5020             IF F < 1 OR DE > 5 THEN 5140
5040              DIFF=SMALL1-SMALL0
5060             FOR N=1 TO 9
5080                     Y1=SMALL0 + INT(DIFF/10*N +.5):Y2=Y1:X2=-2
5100                    GOSUB 1500:'**** NOTE    SPEC OF X2,X1 IGNORED
5120             NEXT N
5140            SMALL0=SMALL1
5160 '
5180 '
5200 '
5220    NEXT F
5240    YMAX=YDMAX:'            NEW YMAX VALUE
5260    YMIN=YDMIN:'            NEW YMIN VALUE FOR THE GRAPHICS PURPOSE
5280 '
5300    RETURN
5320 '
5340 '******* END OF Y-AXIS TIC MARK ******
5360 '
5380 '********* DRAWING ROUTINE ***********
5400 '
5420    K=S(I)
5440    X1=INT(AX/(XMAX-XMIN)*(X(I,K)-XMIN))
5460    Y1=INT(AY/(YMAX-YMIN)*(Y(I,K)-YMIN))
5480 '
5500    FOR K=S(I)+1 TO Z(I)
5520            X2=INT(AX/(XMAX-XMIN)*(X(I,K)-XMIN))
5540            Y2=INT(AY/(YMAX-YMIN)*(Y(I,K)-YMIN))
5560 '
5580 '          IF I <> INT(I/2)*2 THEN CLR=1:GOTO 4500
5600 '          CLR=(-1)^K:CLR=(ABS(CLR)+CLR)/2
5620            CLR=1
5640            GOSUB 1500
5660            X1=X2:Y1=Y2
5680    NEXT K
5700    GOSUB 5820:'            LINE LABLING
5720    RETURN
5740 '
5760 '********* END OF DRAWING  **********
5780 '
5800 '******* LINE LABLE POSITIONING ************
5820 '
5840    XSET=255
5860    A$=INKEY$:IF A$="" THEN GOSUB 6140:GOTO 5860
5880    DIS=ASC(A$)
5900    IF DIS=KB1 THEN XSET=XSET+10
5920    IF DIS=KB2 THEN XSET=XSET-10
5940    IF DIS=KB3 THEN YSET=YSET+10
5960    IF DIS=KB4 THEN YSET=YSET-10
```

```
5980    IF DIS < > 32 THEN 5860
6000    POSI=23+INT(YKON*(YSET-YOFF))
6020    LOCATE POSI,INT(XSET/8)+11:PRINT NA$(J,I)
6040    RETURN
6060 '
6080 '******* END OF LINE LABLE ****************
6100 '
6120 '****** FLASHING  POINT ******************
6140 '
6160    INDICATOR=0:'                     0= DOT ON, 1= DOT OFF
6180    XFLASH=XSET + 90
6200   YFLASH=YOFFS - YSET
6220    IF POINT(XFLASH,YFLASH)=1 THEN INDICATOR=1
6240    PSET(XFLASH,YFLASH)
6260    FOR N=0 TO 10:NEXT N
6280    PRESET(XFLASH,YFLASH),0
6300    FOR N=0 TO 10: NEXT N
6320    IF INDICATOR=1 THEN PSET(XFLASH,YFLASH)
6340    RETURN
6360 '
6380 '************* END OF FLASHIBG **************
6400 '
6420 '*********** DOTTING THE DATA POINT ***************
6440 '
6460    FOR L=1 TO LIMIT
6480            XC = INT(AX/(XMAX - XMIN) * (XX(L) - XMIN)) + 90
6500            YC = YOFFS - INT(AY/(YMAX - YMIN) * (YY(L) - YMIN))
6520            CIRCLE(XC,YC),2
6540    NEXT L
6560    RETURN
6580 '
6600 '******************** END OF DATA MARK ***************
6620 '
6640 '******** GRAPHICS STARTS **************
6660 '
6680    FOR J=JS TO JEND
6700            FOR I=SS(J) TO ZZ(J)
6720                    GOSUB 1940:'            MAX. & MIN.
6740            NEXT I
6760 '
6780 '
6800           GOSUB 1200:'       TURN-ON HIGH RESOLUTION SCREEN
6820           GOSUB 2440:'     TITLE OF THE GRAPH
6840           GOSUB 1660:'     SET-UP BOX
6860           GOSUB 2900:'     TIC MARK ON X-AXIS
6880           GOSUB 4440:'    TIC MARK OF Y-AXIS
6900 '
6920            FOR I = SS(J) TO ZZ(J)
6940                    IF SAMPLE(0)=1 THEN GOSUB 6440
6960                    GOSUB 5400:'             DRAWING ROUTINE
6980            NEXT I
7000            IF IBM = 0 THEN LPRINT CHR$(27) CHR$(19)
7020            A$=INKEY$:IF A$="" THEN 7020 ELSE A$=""
7040            IF IBM = 0 THEN LPRINT CHR$(27) CHR$(20)
7060            SCREEN(0):KEY ON
7080    NEXT J
7100 '
7120    RETURN
7140 '
7160 '************** END OF GRAPHICS *****************
```

```
7180 '
7200  '
7220 '
7240 '******* GRAPHICS CONTROL ************
7260 '
7280 '  ARRAY OF X(W,XX),Y(W,XX)
7300 '          W = MAX. DATA SET - 1
7320 '          XX = MAX. DATA POINTS - 1
7340 '
7360 '  I,J,K - CONTROL
7380 '  *** NOTE ****
7400 '          I,J,K ARE USED TO CONTROL DATA SET & POINTS
7420 '          DO NOT CHANGE IN THE MIDDLE OF GRAPHICS ROUTINE
7440 '
7460 '          J CONTOLS NUMBER OF PICTURES TO BE DRAWN
7480 '          I CONTROLS NUMBER OF DATA SETS TO BE DRAWN IN ONE GRAPH
7500 '          K CONTROLS DATA POINTS IN A DATA SET
7520 '
7540 '          FOR J=0 TO JEND
7560 '                  FOR I=SS(J) TO ZZ(J)
7580 '                          FOR K=S(I) TO Z(I)
7600 '                                  X(I,K)
7620 '                                  Y(I,K)
7640 '                          NEXT K
7660 '                  NEXT I
7680 '          NEXT J
7700 '
7720 '
7740 '
7760 ' -------------------------------------------
7780 '
7800 ' ----------- Print Out Routine -----------
7820 '
7840    PRINT "SIZE & NUMBER OF PARTICLES"
7860    C$(0)=" micron     ":C$(1) = " particles"
7880 '
7900    FOR I= 1 TO XX+1 STEP 2
7920            LPRINT I;C$(0);Y(3,I);C$(1),I+1;C$(0);Y(3,I+1);C$(1)
7940    NEXT I
7960    RETURN
7980 '
8000 ' ----------------------------------------
8020 '
8040 '
8060    CLS: CL=15
8080    C$(1)="Graphics output selection"
8100    C$(2)="1 - Force only            "
8120    C$(3)="2 - Leakage only"
8140    C$(4)="3 - Clerance only"
8160    C$(5)="4 - Frequency vs. Size only"
8180    C$(6)="5 - Silt rate only"
8200    C$(7)="6 - Print Out of Freq. vs. Size"
8220    C$(8)="7 - All"
8240 '
8260    FOR RW=3 TO 10
8280            LOCATE RW,CL:PRINT C$(RW-2)
8300    NEXT RW
8320 '
8340    A$=INKEY$:IF A$="" THEN 8340
8360    A=VAL(A$):A$=""
```

```
8380    IF A=1 THEN JS=4:JEND=4:GOTO 8540
8400    IF A = 2 THEN JS=0:JEND =0:GOTO 8540
8420    IF A = 3 THEN JS=2:JEND=2:GOTO 8540
8440    IF A =4 THEN JS=3:JEND=3:GOTO 8540
8460    IF A = 5 THEN JS=1:JEND=1:GOTO 8540
8480    IF A = 6 THEN GOSUB 7820:GOTO 8040
8500    IF A <> 7 THEN CLS:GOTO 8260
8520    JS=0
8540            SS(0)=0:ZZ(0)=0:'              I-CONTROL
8560                    S(0)=0:Z(0)=XX1:'SAMPLE(0)=1:'        K-CONTROL
8580            SS(1) = 1:ZZ(1) = 1
8600                    S(1)=0:Z(1)=XX1
8620            SS(2) = 2:ZZ(2) = 2
8640                    S(2)=0:Z(2) = XX1
8660            SS(3) = 3:ZZ(3) = 3
8680                    S(3)=0:Z(3)=XX
8700            SS(4) = 4:ZZ(4) = 4
8720                    S(4)=0:Z(4)=XX
8740 '
8760    RETURN
8780    '
8800 ' --------------------------------------------
8820 '
8840 '
8860     W=4:'                     W=X;X+1 IS NUMBER OF DATA SET YOU HAVE
8880     XLMT=300:'                  XX=?;XX-1 IS NUMBER OF ELEMENTS
8900     JEND=4:'                    JEND=#;#+1 DRAWINGS
8920     DIM X(W,XLMT),Y(W,XLMT):'T(XX)
8940 '
8960 '
8980 '------------- AXIS LABELS ---------------------
9000 '
9020    XL$(0)="TIME (sec)"
9040    YL$(0)="FLOW"
9060    T$(0)="LEAKAGE FLOW (mL/sce)"
9080    XL$(1)="TIME (sec)"
9100    YL$(1)="RATE OF SILT"
9120    T$(1)="CHANGE IN SILT RATE (cubic micrometer per mL of fluid)"
9140    XL$(2)="TIME (sec)"
9160    YL$(2)="CLEARANCE"
9180    T$(2)="CHANGE IN CLEARANCE DUE TO SILT (micrometer)"
9200    XL$(3)="SIZE (micrometer)"
9220    YL$(3)="FREQUENCY"
9240    T$(3)="PARTICLES RETAINED IN THE CLEARANCE"
9260    XL$(4)="FORCE (lbf)"
9280    YL$(4)="PROBABILITY"
9300    T$(4)="PROBABILITY OF FORCE REQUIRED TO BREAK SEIZURE"
9320    NA$(0,0)="Min. flow rate is "+STR$(Y(0,XX1))+" mL/sec"
9340    NA$(1,1)="Max. silt rate is "+STR$(Y(1,XX1))
9360    NA$(2,2)="Final clearanc is "+STR$(Y(2,XX1))+" micron"
9380    NA$(3,3)="<= Silt in the clearance"
9400    NA$(4,4)=" Max Force is "+STR$(FORCE)+" lbf"
9420 '
9440 '
9460     RETURN
9480 '
9500 '************* END OF GRAPHICS CONTROL ********************
9520 '
9540 '**************** MAIN PROGRAM SUBROUTINES **********************
9560 '
```

```
9580 ' ------------- PRE-CALCULATION OF AREA UNDER STANDARD NORMAL CURVE --
9600 '
9620   DIM NORMAL(401)
9640   GOTO 9780
9660 '
9680 ' ----- FUNCTION TO BE INTEGERATED ----
9700 '
9720   F = CONST * EXP(-X*X/2)
9740   RETURN
9760 '
9780 '
9800   PI = 3.1415927#
9820   FX = .5         :'      AT X=0 ,FX=0.5
9840   NORMAL(0) = FX
9860   H = .01
9880   CONST = 1/SQR(2*PI)
9900 '
9920   CLS:LOCATE 12,5:PRINT "wait a minute"
9940   FOR J=0 TO 4 STEP H
9960           P = J * 100
9980           X = J
10000           GOSUB 9700
10020           K1 = H*F
10040           X=J+H/2
10060           GOSUB 9700
10080           K2 = H*F
10100           K3 = K2
10120           X = J + H
10140           GOSUB 9700
10160           K4 = H * F
10180           FX = FX + (K1 + 2*K2 + 2*K3 +K4)/6
10200           NORMAL(P+1) = FX
10220 '         PRINT P,NORMAL(P)
10240   NEXT J
10260 '
10280   RETURN
10300 '
10320 ' --------------------------------------------
10340 '
10360 '  ----------- reassignment of Variables --------
10380 '
10400  H0 = R(1):D=R(2):PU=R(3):PD=R(4):L=R(5):SYP=R(6):SYB=R(7):SYS=R(8)
10420  S=R(9):G=R(10):BETA=R(11):ECC=R(12):DT=R(13):TSET=R(14)
10440   GLN=R(15):GL=R(16)
10460  RETURN
10480 '
10500 ' -----------------------------------------------
10520 '
10540 ' ------------- PRINT STATEMENTS -------
10560 '
10580   LOCATE RW,CL:PRINT B$;VR
10600   RETURN
10620 '
10640 ' ---------------------------------------
10660 '
10680 '
10700 ' -------------- INITIALIZATION & ASSIGNMENTOF PARAMETER -----
10720 '
10740 '  The required inputs
10760 '
```

```
10780 '          H0      ;       initial clearance
10800 '          D       ;       inner dia. of the bore  (in.)
10820 '          MU      ;       absolute viscosity  (lbf * sec^2 / in. )
10840 '          PU      ;       upstream pressure  (psi)
10860 '          PD      ;       downstream pressure  (psi)
10880 '          L       ;       silt land length  (in)
10900 '          SYP     ;       Yield Strength of Particle (lbf/in^2)
10920 '          SYB     ;       Yield Strength of Bore  (lbf/in^2)
10940 '          SYS     ;       Yield Strength of Spool (lbf/in^2)
10960 '
10980 '  ******  PARTICLE SIZE DISTRIBUTION INFORMATION *****
11000 '
11020 '  <Upstream>
11040 '          NTU     ;       number of particle greater than 1 micrometer in

11060 '                          1 mg/L of ACFTD
11080 '          BU      ;       slope of particle size distribution in log - lo
g
11100 '                          normal
11120 '          G       ;       gravimeteric level  (mg/L)
11140 '
11160 '  <Downstream>
11180 '          NTD     ;       number of particle greater than 1 micrometer in

11200 '                          1 mg/L of ACFTD
11220 '          BD      ;       slope of particle size distribution in log - lo
g
11240 '                          normal
11260 '
11280 '----------------------------------------
11300 '
11320 '----------- initialization -----------
11340 '
11360   H0  = 25          :' micrometer
11380   D   = 6.355/25.4 :' inches
11400   MU = 2.889E-06    :' lbf*sec/in^2  (Dexron II at 80 C => 20 centistokes)

11420   PU  = 190         :' lbf/in^2
11440   PD  = 10          :' lbf/in^2
11460   L   = .25         :' rough estimate
11480   SYP = 58000!       :' yield strength of particle
11500   SYB = 10000        :' yield strength of bore
11520   SYS = 30000       :' yield strength of spool
11540   MI = .0001/2.54     :'       converts micrometer to inch
11560   S   = 1            :' shape factor
11580   G    = 50           :' mg/L
11600   BETA = 2          :'  (Nu/Nd)
11620   ECC   = 1         :' (0<= ECC <=1)  Eccentricity
11640   T = 0             :' (sec)
11660   DT   = 6          :' (time step)
11680   I = 0             :' counter
11700   TSET = 60         :' sec;
11720   CIML = 2.54^3     :'  cubic inches to milli-liter
11740  GLN = 0           :' number of grooves
11760  GL = .01         :' average length of grooves (inches)
11780  HS = 400          :' hardness of spool (Brinell)
11800 '
11820 ' ----------------------------------------
11840 '
11860  '
```

```
11880 ' ------------- ASSIGNMENTS ----------------------------
11900 '
11920    DIM R(17)
11940    R(1) = H0:R(2)=D:R(3)=PU:R(4)=PD:R(5)=L:R(6)=SYP
11960    R(7)=SYB:R(8)=SYS:R(9)=S:R(10)=G:R(11)=BETA:R(12)=ECC
11980    R(13)=DT:R(14)=TSET:R(15)=GLN:R(16)=GL
12000 '
12020 ' -----------------------------------------------------
12040 '
12060 '
12080 '
12100 '
12120   DIM C$(20)
12140 '
12160    CLS
12180    LOCATE 2,25:PRINT"MENU"
12200    C$(1)="1 ---  H0 : initial clearance micrometer        "
12220    C$(2)="2 ---  D  : inner dia. of the bore inches       "
12240    C$(3)="3 ---  PU : Upstream pressure lbf/in^2          "
12260    C$(4)="4 ---  PD : Down stream pressre lbf/in^2        "
12280    C$(5)="5 ---  L : Silt land length in inches           "
12300    C$(6)="6 ---  SYP: Yield strength of particle lbf/in^2 "
12320    C$(7)="7 ---  SYB: Yield Strength of Bore  lbf/in^2    "
12340    C$(8)="8 ---  SYS: Yield Strength of Spool  lbf/in^2   "
12360    C$(9)="9 ---  S  : shape factor                        "
12380    C$(10)="10---  G  : Concentration  mg/L                 "
12400    C$(11)="11---  BETA : Retention characteristic (Nu/Nd)  "
12420    C$(12)="12---  ECC: (0<= ECC <=1)  Eccentricity         "
12440    C$(13)="13---  DT : Time step in sec.                   "
12460    C$(14)="14---  TSET : Total length of time.  (sec)      "
12480    C$(15)="15--- GLN : Total number of grooves             "
12500    C$(16)="16--- GL : The average length of grooves  (in.) "
12520   CL=10
12540   FOR RW=4 TO 19
12560          B$=C$(RW-3):VR=R(RW-3)
12580          GOSUB 10560
12600   NEXT RW
12620 '
12640    PRINT" **** Do you want change the value (Y/N)"
12660   A$=INKEY$:IF A$="" THEN 12660
12680   IF A$="n" OR A$="N" THEN A$="":GOTO 12780
12700   INPUT" ---- type in the number of the parameter you want to change";CHOI
CE
12720   INPUT" ---- type in the new value";R(CHOICE)
12740   GOSUB 10380:GOTO 12140
12760 '
12780   RETURN
12800 '
12820 '
12840 ' ----- CALCULATION OF THE VOLUME OF A GIVEN PARTICLE SIZE
12860 ' ----- DISTRIBUTION
12880 '
12900 '  <INPUT>          NT & B
12920 '  <OUTPUT>         V
12940 '  <Variable>       UPL ---- Upper Limit
12960 '                   NOA ---- Area Under Normal Distribution
12980 '
13000   UPL = 3/SQR(2*B)
13020   UPL = INT(UPL * 100 + .5)
13040   IF UPL > 400 THEN NOA = 1 ELSE NOA = NORMAL(UPL)
```

```
13060 '
13080    V = PI*NT*(1+3*SQR(PI/B)*EXP(9/4/B)*NOA)/6/S
13100    RETURN
13120 '
13140 ' -------------------------------
13160 '
13180 ' ************* TABLE OF PROBABILITY FOR CLEARANCE AREA ********
13200 '
13220   F = SQR(RAD^2 + EXE^2 - 2*RAD*EXE*COS(TH)) - RADS
13240   RETURN
13260 '
13280 ' -----------------------------------------------
13300 '
13320 ' ---------- INITIALIZATION --------
13340 '
13360 ' DIM CTABLE(180)
13380   CLS:PRINT"MAKING A TABLE. IT WILL SOON BE OVER !"
13400   IM = 2.54 * 10000:'      INCHES TO MICRON
13420   RAD = D*IM/2
13440   RADS = .5*(D*IM - 2*H0)
13460   EXE = H0 * ECC
13480   H  = PI/180
13500 '
13520 '
13540   FOR J=0 TO 180
13560           LOCATE 1,60:PRINT J
13580           THETA = J*H
13600           TH = THETA
13620           GOSUB 13200
13640           K1 = H*F
13660           TH = THETA + H/2
13680           GOSUB 13200
13700           K2 = H*F
13720           K3 = K2
13740           TH = THETA + H
13760           GOSUB 13200
13780           K4 = H * F
13800           FX = FX + (K1+2*K2+2*K3+K4)/6
13820           CTABLE(J) = FX
13840 '         PRINT J,THETA,CTABLE(J)
13860   NEXT J
13880 '
13900   MAXAREA = CTABLE(180)
13920 '
13940   RETURN
13960 '
13980 ' ***************************************
14000 '
14020 ' ---------- BINARY SEARCH -------------
14040 '
14060   FR = (BETA*GM)^(KCON*KON1)/GM-KCON*LOG(BETA*GM)/BU -1
14080   RETURN
14100 '
14120 ' -----------------------------------
14140 '
14160 '
14180 '  ----- PRELIMINARY CALCULATIONS ---
14200 '
14220 '
14240   NTU = 1751.9    :' particles/ 1 mg/L of ACFTD
```

```
14260   NTU = NTU * G
14280   BU = (LOG(NTU) - LOG(2.869*G))/(LOG(40))^2:'    2.869 particles- 40 micr
on / mg/L of ACFTD
14300   DP = PU - PD     :'      pressure difference
14320   HT = H0          :'  (HT is variable)
14340    L =  L - GLN * GL :'   reduction of effective silt land
14360 ' ---- BINARY SEARCH
14380   LGAM = 1/BETA
14400   D = 2
14420   KON1 = (LOG(D))^2
14440   EPSILON = .000001
14460   GL = LGAM
14480   GU = .999
14500 '
14520 ' --------------------------------
14540 '
14560 '
14580 ' --------- CALCULATION OF THE LEAKAGE FLOW
14600 '
14620    Q = (PI*D*(1+1.5*ECC)*DP)/(12*MU*L)*((HT*MI)^3)
14640    Q = Q * CIML    :' converts to milli-liter
14660 '
14680 ' ----------- UPSTREAM VOLUME -------------
14700 '
14720    NT = NTU
14740    B = BU
14760    GOSUB 12980
14780    VU = V
14800 '
14820 ' ----------- DOWNSTREAM VOLUME --------
14840 '
14860 '          Calculate number of particles at the size HT with
14880 '          particle capture (retention) efficiency BETA
14900 '
14920  '  ---- BINARY SEARCH FINDS THE APPROPRIATE NTD,BD,GAMMA
14940 '
14950   GU = .999:GL = 1/BETA
14960   KCON = 1/(LOG(HT))^2
14980   XO = KCON * KON1
15000 '
15020   GM = GU
15040   GOSUB 14040
15050 ' LOCATE 9,10:PRINT"UP  ";GU,"DOWN ";GL,"ERROR ";FR
15060   IF FR>0 THEN GU = (1+GU)/2:GOTO 15020
15080 '
15100   GM = (GL+GU)/2
15120   GOSUB 14040
15140   IF ABS(FR) < EPSILON THEN 15240
15160   IF FR > 0 THEN GL=GM
15180   IF FR < 0 THEN GU = GM
15190 ' LOCATE 10,10:PRINT"UP  ";GU,"DOWN ";GL,"ERROR ";FR
15200   GOTO 15100
15220 '
15240   UGAM = GM
15260   GAMMA = LGAM+.9*(UGAM-LGAM)
15270   LOCATE 10,10:PRINT"FINAL GAMMA  ";GAMMA
15280   NTD = GAMMA * NTU
15300   BD = BU + LOG(GAMMA * BETA)*KCON
15320 '
15340   NT = NTD
```

```
15360   B = BD
15380   GOSUB 12980
15400   VD = V
15420 '
15440 ' ------ REMAINED VOLUME
15460 '
15480   VT = VU-VD      :'  Cubic micrometer retained / ml of fluid passed clear
ance
15500 '
15520 ' ------ TOTAL SILTED VOLUME
15540 '
15560   SILT = VT * Q * DT * MI^3
15580 '
15600 ' ------ SILT HEIGHT
15620 '
15640   EPSI = (SILT/(2*PI*L*(D-HT*MI)))/MI
15660 '
15680 ' GOTO 12300
15700 ' ------ FREQUENCY VS SIZE
15720 '
15740   IF FIRST =1 THEN GOTO 15840
15760   FIRST =1
15780   FOR IP = 1 TO 100
15800           X(3,IP-1) = IP
15820   NEXT IP
15840 '
15860   FOR IP = 1 TO 100
15880           FUD = 2*NTU*BU*LOG(IP)/IP*EXP(-BU*(LOG(IP))^2)
15900           FDD = 2*NTD*BD*LOG(IP)/IP*EXP(-BD*(LOG(IP))^2)
15920           TMP = FUD - FDD
15940           IF TMP <0 THEN TMP = 0
15960           Y(3,IP-1) = TMP*Q*DT + Y(3,IP-1)
15980           LOCATE 2,5:PRINT I,IP
16000   NEXT IP
16020 '
16040 ' ------ STORE THE VALUES
16060 '
16080   X(0,I) = T
16100   X(1,I) = X(0,I)
16120   X(2,I) = X(1,I)
16140 '
16160   Y(0,I) = Q      :' (ml/sec)
16180   Y(1,I) = VT     :'  (cubic micrometers / ml of fluid)
16200   Y(2,I) = HT
16220 ' PRINT Q;" ML/SEC ",VT;" MICRON^3/ML",HT;" MICRON",EPSI" SILT micron"
16240 ' STOP
16260 '
16280 ' ---- INCREMENT STEP
16300 '
16320   I = I +1
16340   D = D-2*EPSI*MI
16360   HT = HT - 2*EPSI
16380   T = T+DT
16400 '
16420   IF HT < 1 THEN  17220
16440   IF T<= TSET THEN 14600
16460 '
16480 ' ******  FORCE CALCULATION **********************
16500 '
16520    GOTO 17220 :' SKIP SUBROUTINE
```

```
16540 ' ------- STEP TO CALCULATE FORCE ----
16560 '
16580    XINT = (RAD*RAD + EXE*EXE - (SIZEX + RADS)^2)/(2*RAD*EXE)
16600    IF XINT = 1 THEN THAI = 0:RETURN
16620    IF XINT = -1 THEN THAI = PI:RETURN
16640    THAI = -ATN(XINT/SQR(-XINT*XINT+1)) + PI/2
16660    RETURN
16680 '
16700    FOR I = SEL TO 3
16720            SIZEX = HL(I) * P
16740            GOSUB 16560
16760            ANGLE(I) = THAI
16780    NEXT I
16800 '
16820    SLP = (TMP-HL(SEL-1))/(HL(SEL)-HL(SEL-1))
16840    BFM(SEL-1)=(FM(SEL)-FM(SEL-1))*SLP + FM(SEL-1)
16860    BFM(SEL-1) = (BFM(SEL-1)+FM(SEL))/2
16880    BFA(SEL-1)=(FA(SEL)-FA(SEL-1))*SLP + FA(SEL-1)
16900    BFA(SEL-1) = (BFA(SEL-1)+FA(SEL))/2
16920 '
16940    RETURN
16960 '
16980 ' ---------------------------------------
17000 '
17020    TMP = PCK/P:ANGLE(0)=PI
17040    IF (TMP > HL(1)) AND (TMP < HL(0)) THEN SEL=1:GOSUB 16700:RETURN
17060    IF TMP>=HL(2) AND TMP<HL(1) THEN SEL=2:ANGLE(1)=PI:GOSUB 16700:RETURN
17080    IF TMP>HL(3) AND TMP<HL(2) THEN SEL=3:ANGLE(2)=PI:ANGLE(1)=PI:GOSUB 167
00:RETURN
17100    IF TMP = HL(3) THEN ANGLE(3) = PI:ANGLE(2)=PI:ANGLE(1)=PI
17120    RETURN
17140 '
17160 ' ---------------------------------------
17180 '
17200 ' -------- Initialization ----------------
17220 '
17240 ' DIM HL(3),FM(3),FA(3),BFM(3),BFA(3),ANGLE(3),CAREA(3)
17260    TH = ATN(1/1.49)
17280 '
17300    HL(0) = 1       :'     H/L  ratio
17320    HL(1) = .8524232
17340    HL(2) = .7048464
17360    HL(3) = .557269
17380 '
17400    FM(3) = .1116427
17420    FM(2) = 6.756162E-02
17440    FM(1) = 2.633637E-02
17460    FM(0) = 0
17480 '
17500    FA(3) = 7.819522E-02
17520    FA(2) = .0363837
17540    FA(1) = 9.836434E-03
17560    FA(0) = 0
17580 '
17600   IF SYP> SYB THEN SYC=SYB ELSE SYC=SYP
17620   IF SYP> SYS THEN SYI=SYS ELSE SYI=SYP
17640 '
17660 '-----------------------------------------------
17680 '
17700 '
```

```
18840            Y(4,I)=EXP(-(X(4,I)-FORCEA)^2/2/SIGMA^2)/SIGMA/SQR(PI*2)
18860   NEXT I
18880 '
18900 '
18920    LPRINT "TOTAL MAX. FORCE = ";FORCE,"TOTAL AVG. FORCE = ";FORCEA
18940   RETURN
18960 '
18980 ' ---------------------------------------------
19000 '
19020 ' ****************** MAIN PROGRAM CONTROL ******************
19040 '
19060            GOSUB 9600:'    TABLE OF AREA  UNDER NORMAL CURVE
19080            GOSUB 11340:'   PARAMETER INITIALIZATION & REASSIGNMENTS
19100 '
19120            FIRSTRUN = FIRSTRUN + 1
19140            XX1 = INT(TSET/DT + .5):XX=99
19160            IF XX1<=300 THEN 19260
19180            PRINT" ***** TOO MANY ITERATIONS ******"
19200            PRINT" INCREASE TIME STEP OR DECREASE LENGTH OF TIME"
19220            PRINT" PRESS ANY BAR "
19240            A$ = INKEY$:IF A$="" THEN 19240:GOTO 260
19260 '
19280            GOSUB 13340:'   TABLE OF AREA UNDER CLEARANCE
19300            CLS
19320            LOCATE 5,10:PRINT" TOTAL NUMBER OF SILT PROCESS ITERATION  ";(XX
1+1)*(XX+1)
19340 '
19360            IF FIRSTRUN >=2 THEN 19400
19380            GOSUB 8840:'    ARRAY SPACE ALLOCATION
19400            GOSUB 14240:'   MEAT
19420            GOSUB 8040:'    SET GRAPHICS PARAMETER
19440            GOSUB 9320:'    GET MAX FORCE
19460            GOSUB 6660:'    JUMP TO GRAPHICS CONTROL
19480            JEND =4:'       Rcovery from J - Control of graphics
19500 '
19520 '
19540   LOCATE 5,10:PRINT " FORCE REQUIRED TO BREAK SEIZURE IS  ";FORCE,"lbf"
19560   PRINT
19580   PRINT"                 A --- Back to the graphics"
19600   PRINT"                 B --- Try other parameters"
19620   PRINT"                 Q --- QUIT"
19640   PRINT"
19660   PRINT"               Please select your choice (A or B)"
19680   A$=INKEY$:IF A$="" THEN 19680
19700   IF A$="a" OR A$="A" THEN GOTO 19420
19720   IF A$="b" OR A$="B" THEN GOTO 19820
19740   IF A$="q" OR A$="Q" THEN END
19760   CLS:GOTO 19560
19780 '
19800 ' ----clean up --
19820 '
19840   I=0:T =0
19860   FOR IP= 1 TO 100
19880            Y(3,IP-1)=0
19900   NEXT IP
19920 '
19940   GOSUB 12160:GOTO 19100
19960 '
19980 PRINT "ERROR CODE ";ERR,"OCCUR AT THE LINE";ERL
```

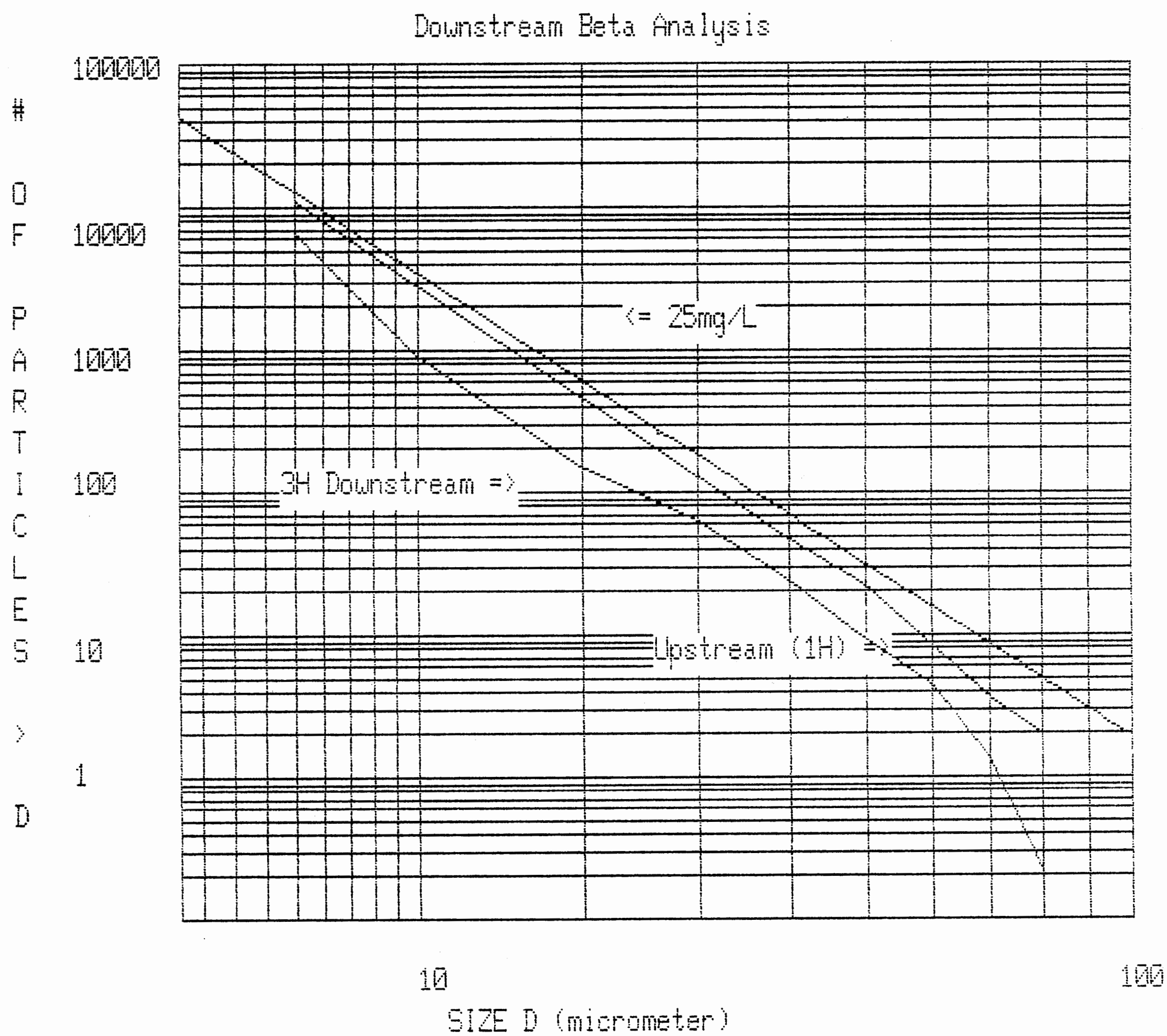
APPENDIX B

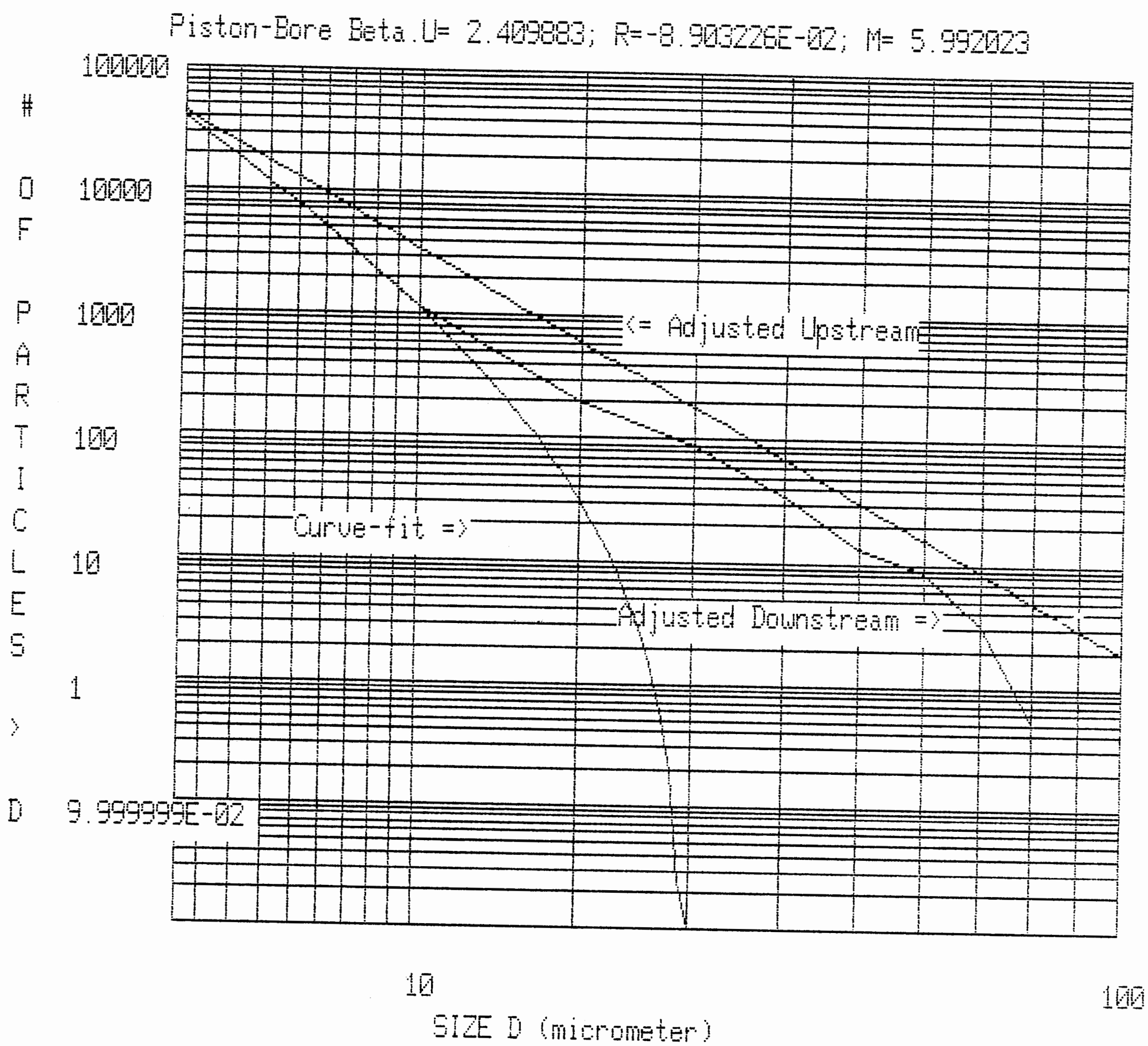
SILT BETA ANALYSIS

***** 1 & 3H TEST RESULT ******

Bore No.3 - Piston No.19, Land Length (6.38 mm)
5 micrometers nominal clearance

Size	Distribution of 25 mg/L	Upstream	Downstream	BETA
5	12917.2	11008.8	6781.337	1.623397
10	3597.918	2878.8	944.0335	3.049468
20	637.1161	487.1	154.7208	3.148252
30	187.6002	132.4	63.5448	2.083569
40	71.72498	48.5	24.36	1.990969
50	32.244	22.4	9.514321	2.354346
60	16.20372	8.729999	4.64232	1.880525
70	8.837601	3.81	1.392	2.737069
80	5.133417	2.09	.231768	9.017638



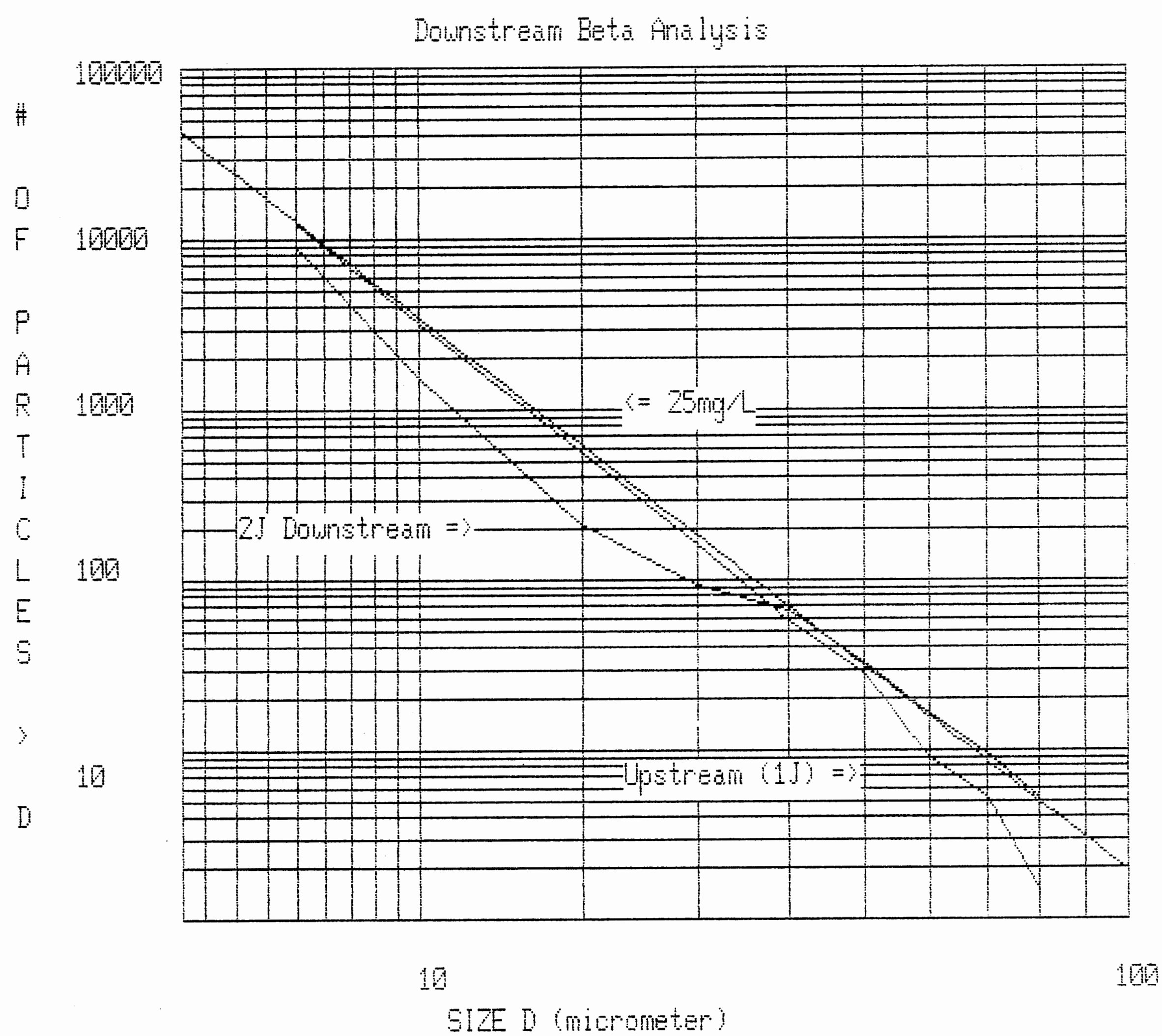


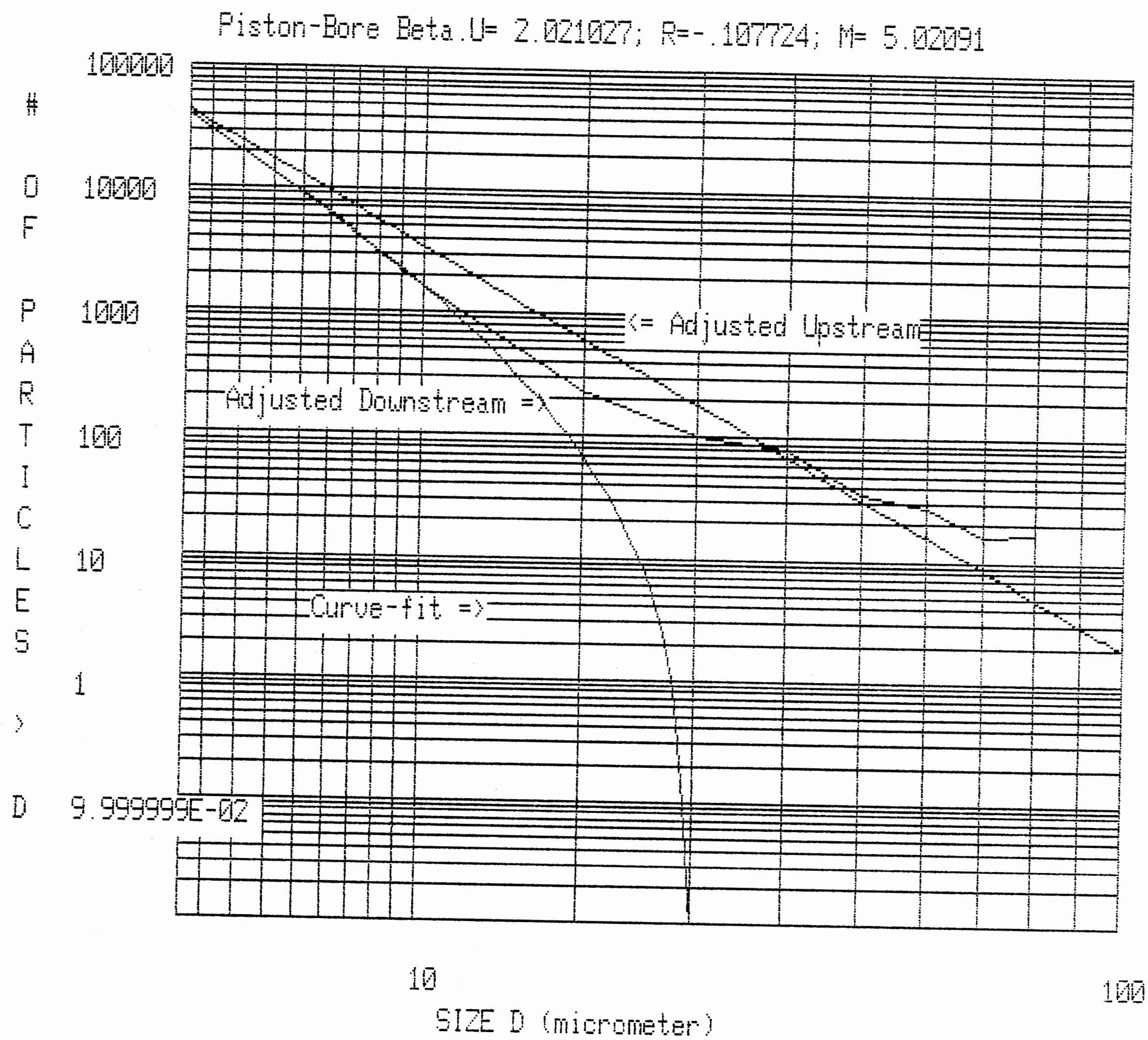
Size micron	Adjusted Upstream	Fitted Downstream	Beta
1	43797.5	39570.05	1.106835
5	12917.2	7956.898	1.623396
10	3597.918	1179.851	3.049467
15	1380.819	209.2399	6.599215
20	637.1161	33.72313	18.89256
25	331.3792	2.987863	110.9084
30	187.6002	.01	****
35	113.1623	.01	****
40	71.72498	.01	****
45	47.30807	.01	****
50	32.244	.01	****
55	22.59059	.01	****
60	16.20372	.01	****
65	11.86098	.01	****
70	8.837601	.01	****
75	6.688775	.01	****
80	5.133417	.01	****
85	3.989182	.01	****
90	3.135055	.01	****
95	2.489046	.01	****

***** 1 & 2J TEST RESULT ******

Bore No.3 - Piston No.19, Land Length (3.19 mm)
5 micrometers nominal clearance

Size	Distribution of 25 mg/L	Upstream	Downstream	BETA
5	12917.2	12543.7	9140.876	1.372265
10	3597.918	3353.2	1575.188	2.128762
20	637.1161	592.4	221.7397	2.671601
30	187.6002	166.5	94.34101	1.764874
40	71.72498	59.1	66.729	.8856719
50	32.244	28.9	32.7509	.8824184
60	16.20372	9.25	16.3371	.5661959
70	8.837601	5.42	9.971	.5435765
80	5.133417	1.58	5.369001	.294282



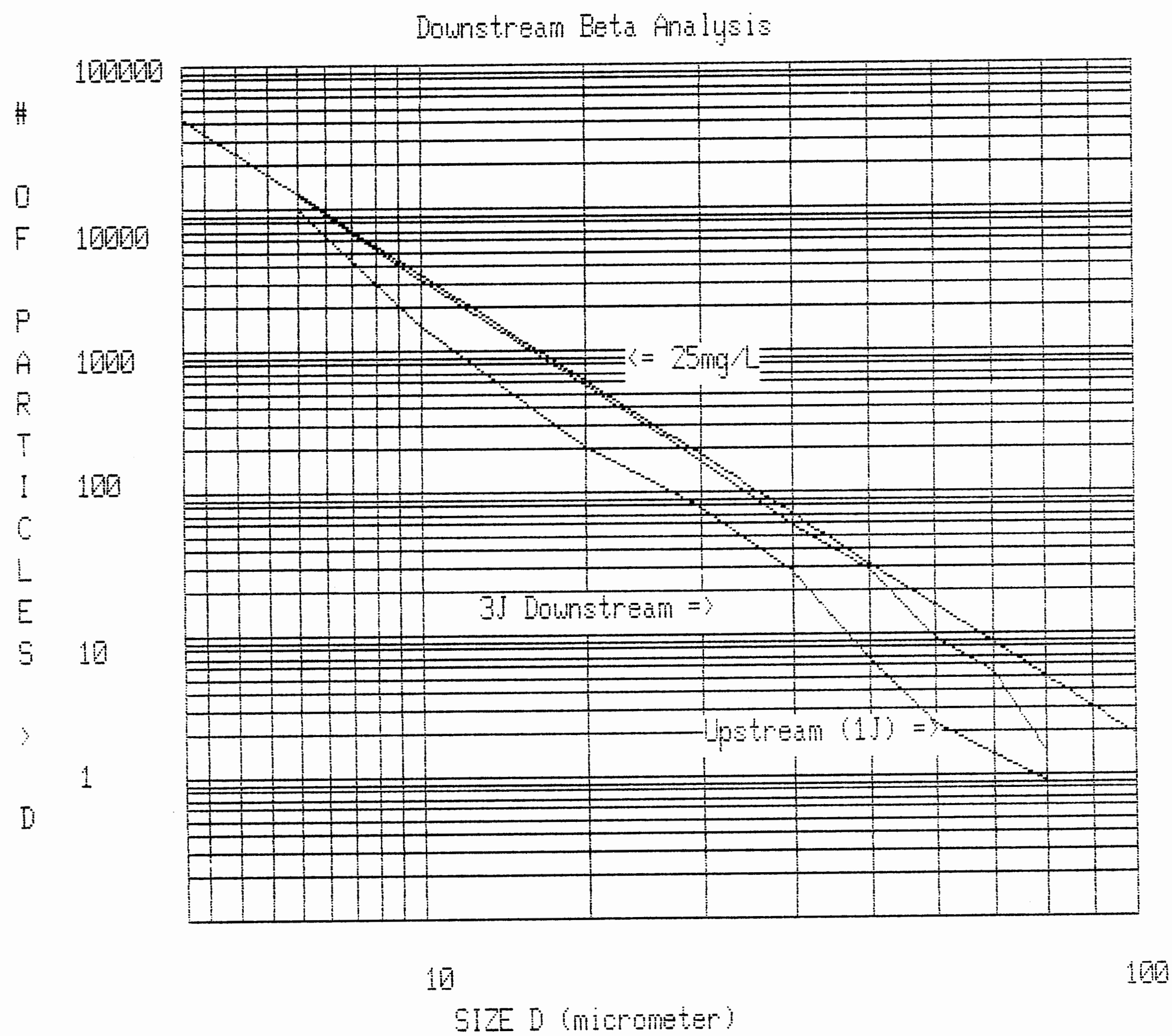


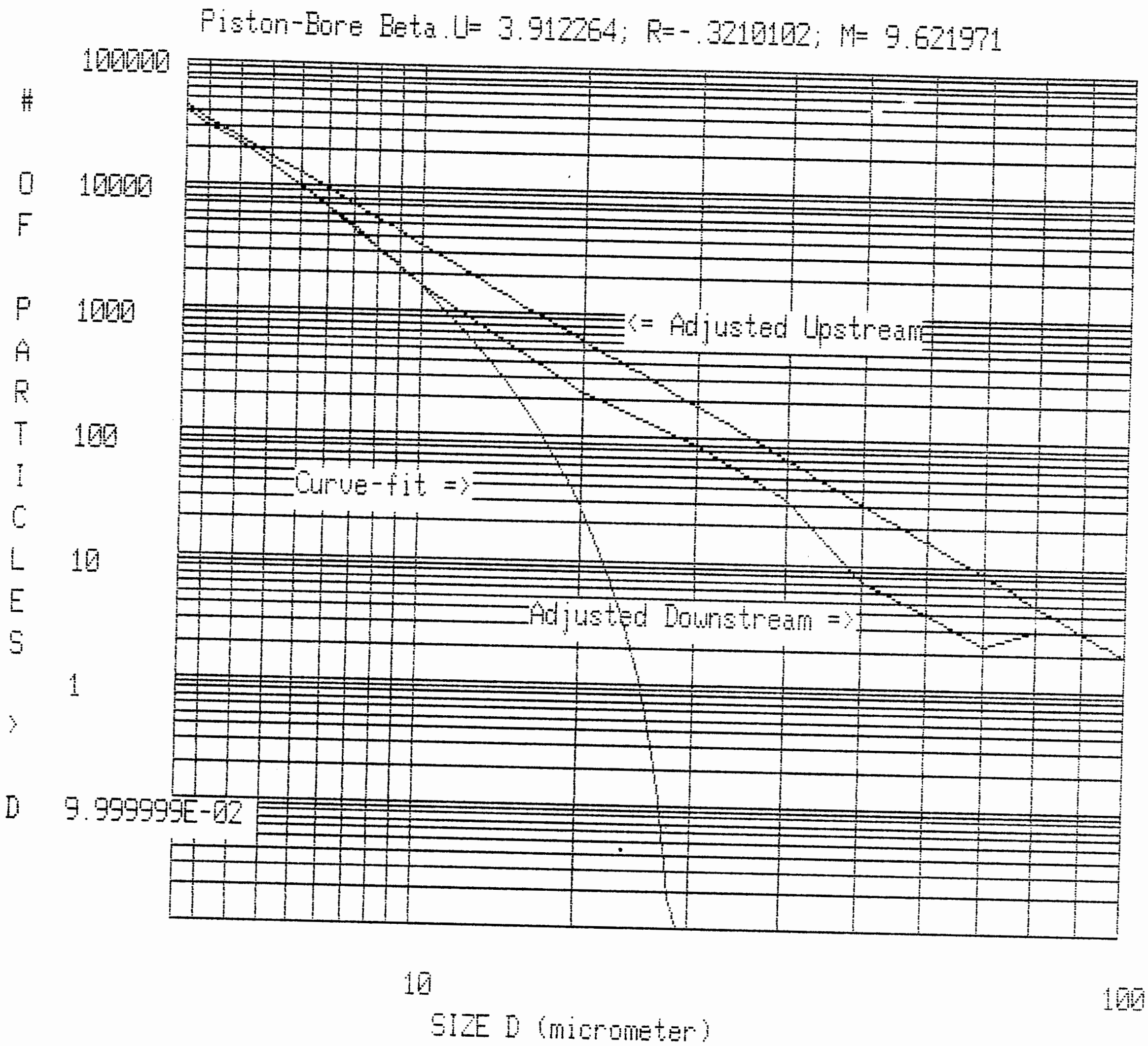
Size micron	Adjusted Upstream	Fitted Downstream	Beta
1	43797.5	40394.7	1.084239
5	12917.2	9413.055	1.372264
10	3597.918	1690.146	2.128761
15	1380.819	363.0846	3.803023
20	637.1161	73.20782	8.702842
25	331.3792	9.035084	36.67693
30	187.6002	.01	****
35	113.1623	.01	****
40	71.72498	.01	****
45	47.30807	.01	****
50	32.244	.01	****
55	22.59059	.01	****
60	16.20372	.01	****
65	11.86098	.01	****
70	8.837601	.01	****
75	6.688775	.01	****

***** 1 & 3J TEST RESULT ******

Bore No.3 - Piston No.19, Land Length (3.19 mm)
5 micrometers nominal clearance

Size	Distribution of 25 mg/L	Upstream	Downstream	BETA
5	12917.2	12543.7	10028.01	1.250866
10	3597.918	3353.2	1554.45	2.157162
20	637.1161	592.4	219.2859	2.701496
30	187.6002	166.5	78.30091	2.126412
40	71.72498	59.1	27.474	2.151125
50	32.244	28.9	6.7239	4.298101
60	16.20372	9.25	2.3859	3.876944
70	8.837601	5.42	1.446	3.748271
80	5.133417	1.58	.9398999	1.68103



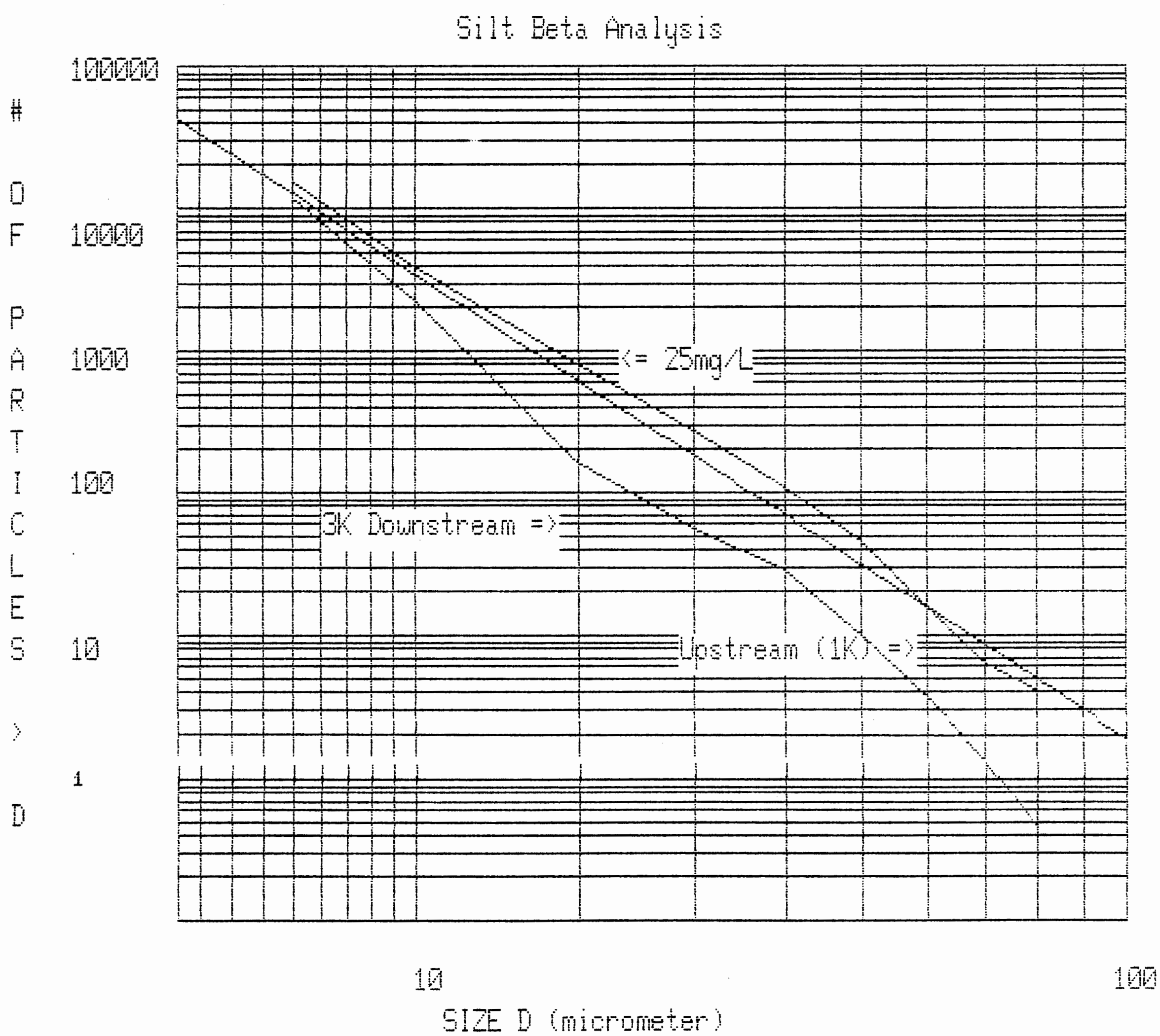


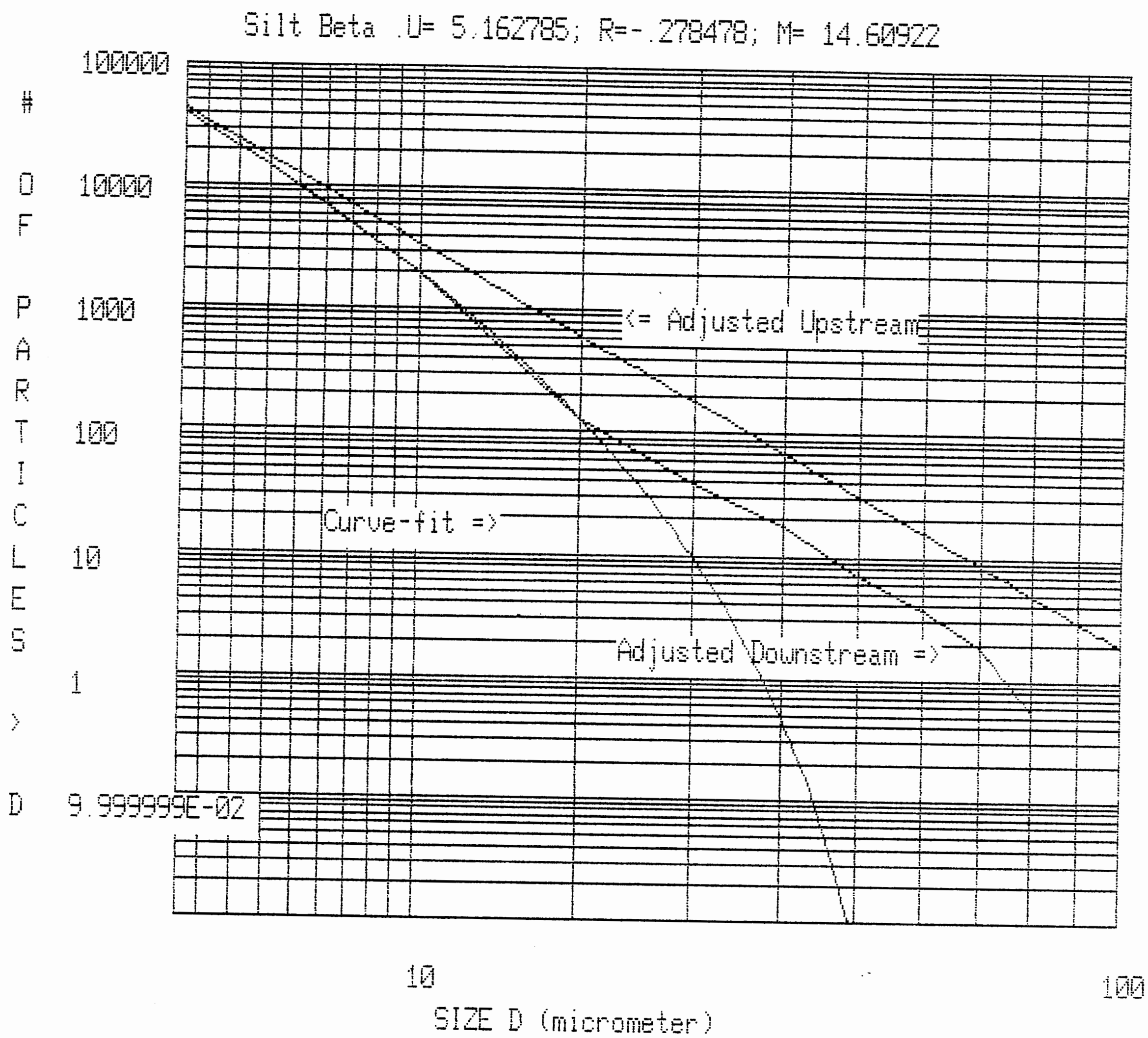
Size micron	Adjusted Upstream	Fitted Downstream	Beta
1	43797.5	41281.78	1.06094
5	12917.2	10326.6	1.250867
10	3597.918	1667.893	2.157164
15	1380.819	261.6242	5.277871
20	637.1161	29.2375	21.79106
25	331.3792	1.097602	301.9119
30	187.6002	.01	****
35	113.1623	.01	****
40	71.72498	.01	****
45	47.30807	.01	****
50	32.244	.01	****
55	22.59059	.01	****
60	16.20372	.01	****
65	11.86098	.01	****
70	8.837601	.01	****
75	6.688775	.01	****

***** 1 & 3K TEST RESULT ******

Bore No.3 - Piston No.23, Land Length (6.38 mm)
10 micrometers nominal clearance

Size	Distribution of 25 mg/L	Upstream	Downstream	BETA
5	12917.2	15258	12251.87	1.245361
10	3597.918	4096.6	2327.531	1.760063
20	637.1161	822.6	170.343	4.82908
30	187.6002	274.8	56.54967	4.859445
40	71.72498	112.9	28.5307	3.957141
50	32.244	46.1	10.0243	4.598825
60	16.20372	16.19	3.7153	4.357656
70	8.837601	6.38	1.402	4.550642
80	5.133417	4.17	.4907	8.498064



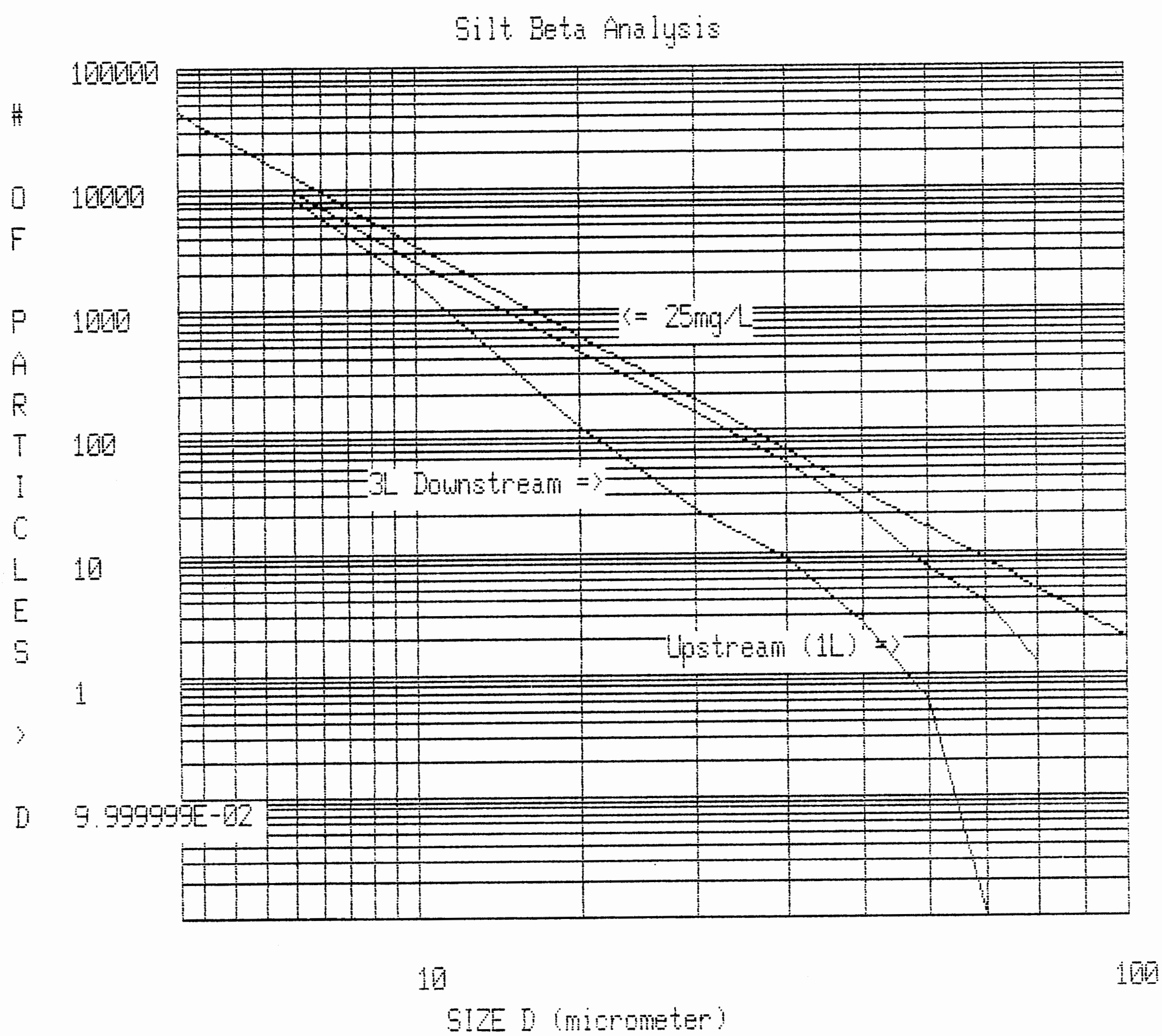


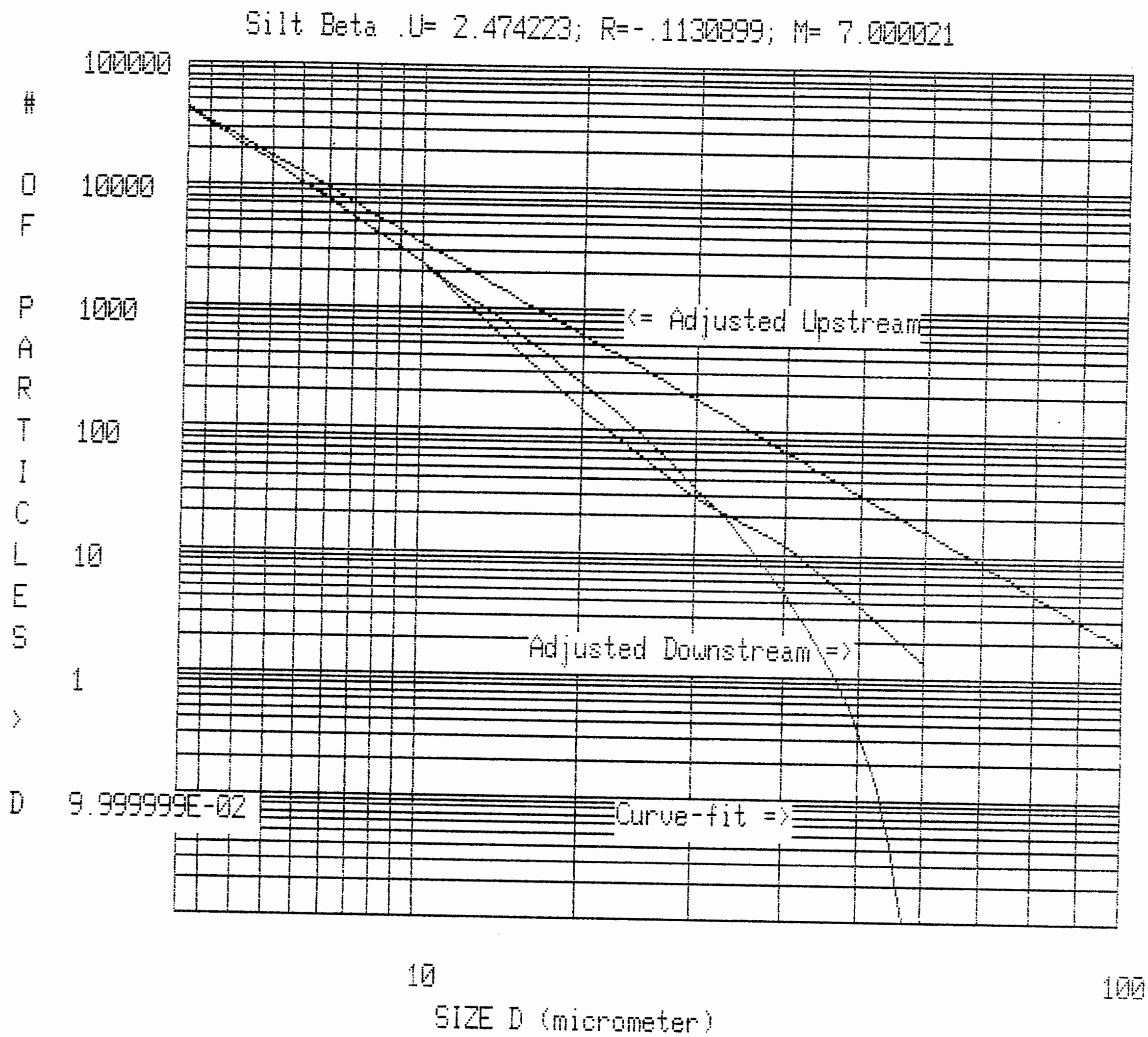
Size micron	Adjusted Upstream	Fitted Downstream	Beta
1	43797.5	40791.38	1.073695
5	12917.2	10372.26	1.245361
10	3597.918	2044.198	1.760063
15	1380.819	501.8064	2.751696
20	637.1161	135.4207	4.704718
25	331.3792	37.36605	8.868457
30	187.6002	9.965322	18.8253
35	113.1623	2.421582	46.73074
40	71.72498	.4946596	144.9987
45	47.30807	7.398339E-02	639.4418
50	32.244	.01	****
55	22.59059	.01	****
60	16.20372	.01	****
65	11.86098	.01	****
70	8.837601	.01	****
75	6.688775	.01	****

***** 1 & 3L TEST RESULT ******

Bore No.3 - Piston No.23, Land Length (3.19 mm)
10 micrometers nominal clearance

Size	Distribution of 25 mg/L	Upstream	Downstream	BETA
5	12917.2	9735.5	8324.379	1.169517
10	3597.918	2587.7	1773.875	1.458784
20	637.1161	469.7	114.9471	4.086228
30	187.6002	150.2	23.0141	6.526434
40	71.72498	54.4	9.255	5.877904
50	32.244	21.6	2.67161	8.085013
60	16.20372	7.43	.617	12.04214
70	8.837601	3.94	.01	****
80	5.133417	1.31	.01	****



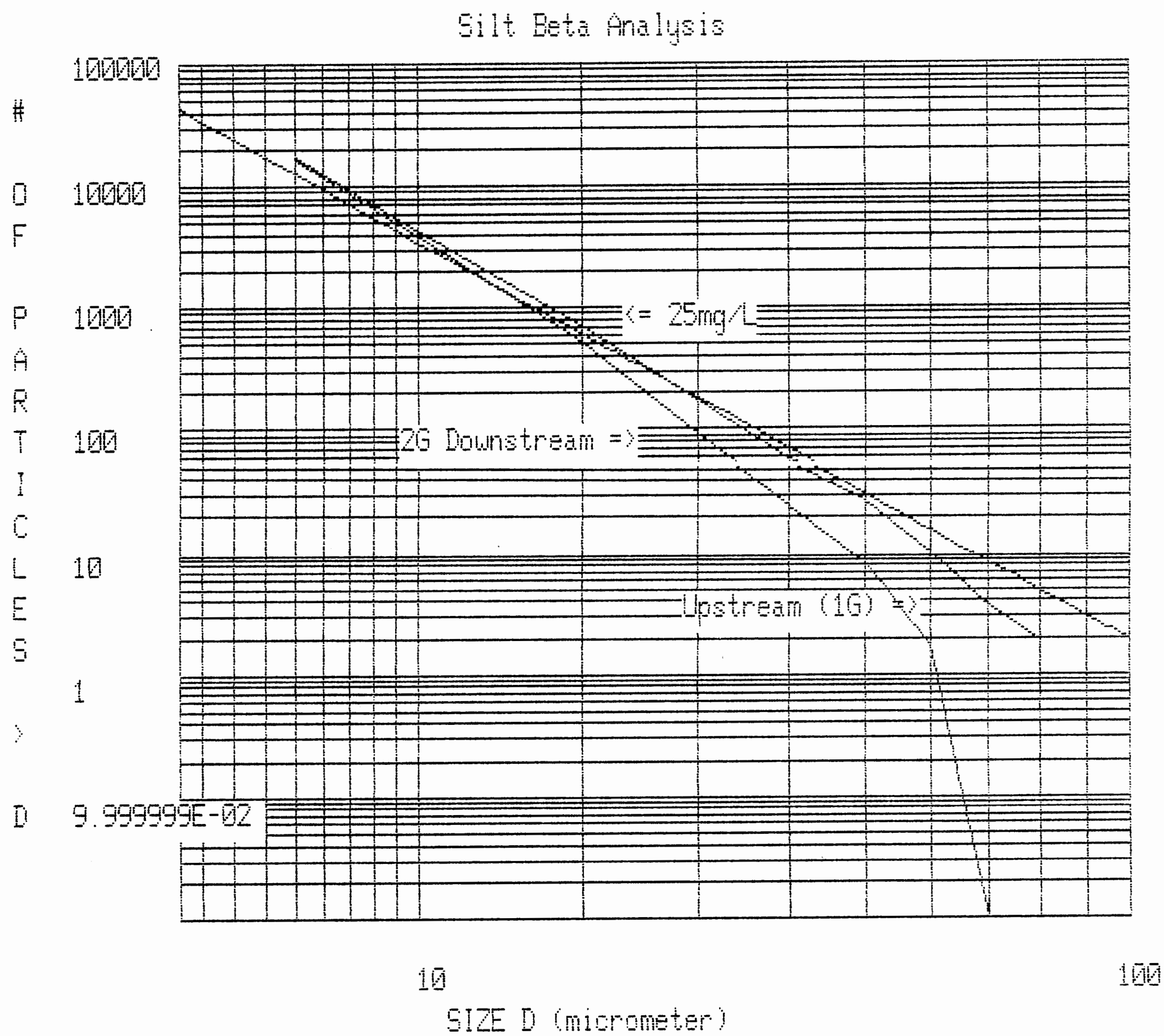


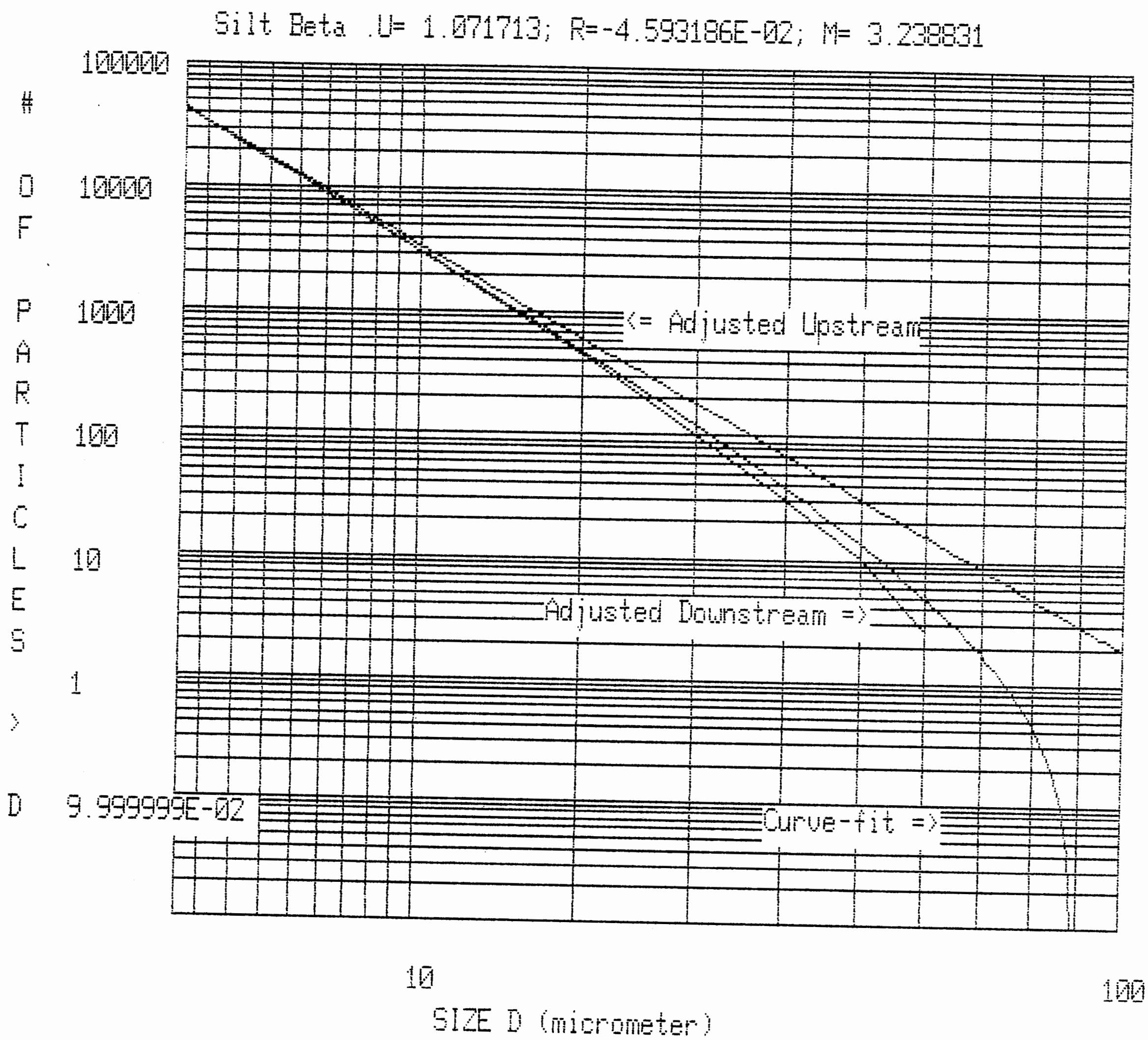
Size micron	Adjusted Upstream	Fitted Downstream	Beta
1	43797.5	42386.36	1.033292
5	12917.2	11044.9	1.169517
10	3597.918	2466.381	1.458785
15	1380.819	733.105	1.883521
20	637.1161	252.989	2.518355
25	331.3792	94.40591	3.510153
30	187.6002	36.35544	5.160169
35	113.1623	13.87771	8.154248
40	71.72498	5.012532	14.30913
45	47.30807	1.594671	29.66635
50	32.244	.384953	83.76088
55	22.59059	4.365352E-02	517.4977
60	16.20372	.01	****
65	11.86098	.01	****
70	8.837601	.01	****
75	6.688775	.01	****

***** 1 & 2G TEST RESULT ******

Bore No.3 - Piston No.25, Land Length (6.38 mm)
15 micrometers nominal clearance

Size	Distribution of 25 mg/L	Upstream	Downstream	BETA
5	12917.2	17444	16652.75	1.047515
10	3597.918	4446.5	4018.49	1.10651
20	637.1161	730.5	540.44	1.351677
30	187.6002	175.1	95.58	1.831973
40	71.72498	59.2	25.075	2.360917
50	32.244	27.7	8.850001	3.129943
60	16.20372	10.74	1.77	6.067796
70	8.837601	3.8	.01	****
80	5.133417	2.01	.01	****



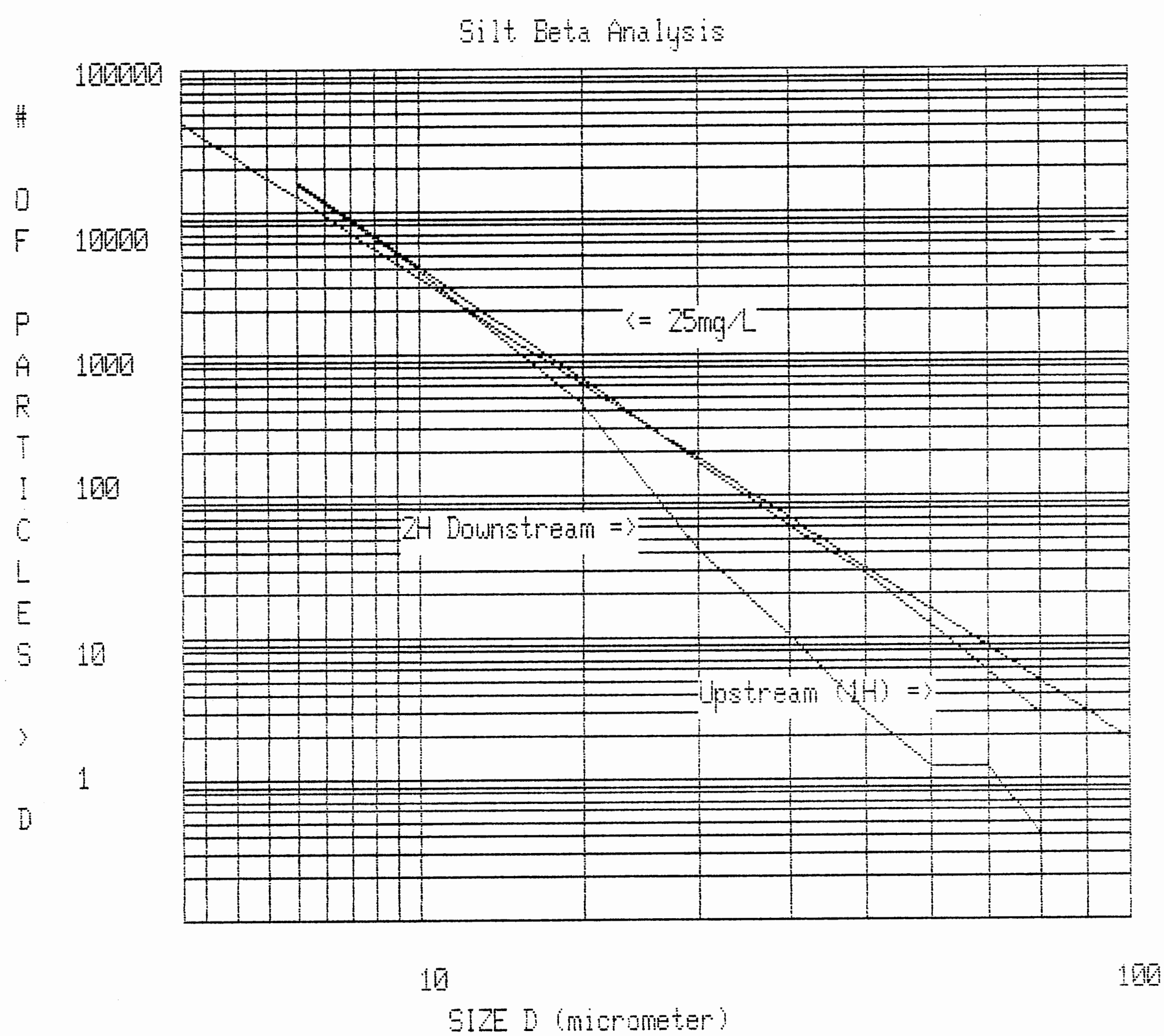


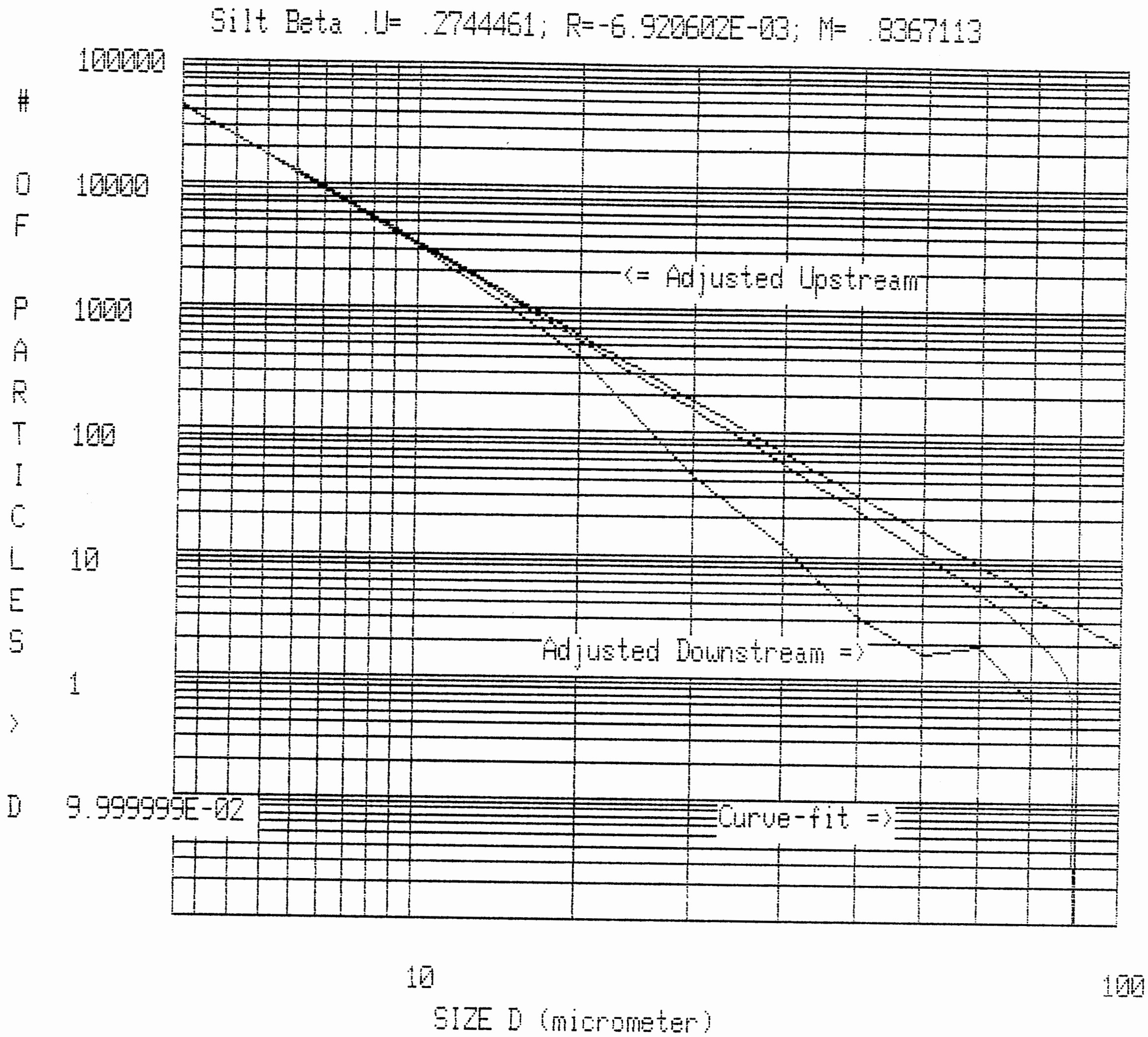
Size micron	Adjusted Upstream	Fitted Downstream	Beta
1	43797.5	43006.26	1.018398
5	12917.2	12331.28	1.047515
10	3597.918	3251.59	1.10651
15	1380.819	1170.609	1.179573
20	637.1161	503.1112	1.266353
25	331.3792	242.0985	1.368778
30	187.6002	125.8875	1.490221
35	113.1623	69.18451	1.635659
40	71.72498	39.57649	1.812313
45	47.30807	23.29494	2.03083
50	32.244	13.9738	2.307461
55	22.59059	8.466728	2.668161
60	16.20372	5.132644	3.156993
65	11.86098	3.076723	3.85507
70	8.837601	1.792868	4.929309
75	6.688775	.9858079	6.78507

***** 1 & 2H TEST RESULT ******

Bore No.3 - Piston No.25, Land Length (3.19 mm)
15 micrometers nominal clearance

Size	Distribution of 25 mg/L	Upstream	Downstream	BETA
5	12917.2	16690	16168.66	1.032244
10	3597.918	4291.4	4045.958	1.060664
20	637.1161	700.6	469.2	1.49318
30	187.6002	177	42.96751	4.119393
40	71.72498	62.7	10.62075	5.903538
50	32.244	28.8	2.97075	9.694522
60	16.20372	12.77	1.275	10.01569
70	8.837601	5.97	1.275	4.682353
80	5.133417	3.05	.42075	7.24896





Size micron	Adjusted Upstream	Fitted Downstream	Beta
1	43797.5	43276.16	1.012047
5	12917.2	12513.71	1.032244
10	3597.918	3392.139	1.060664
15	1380.819	1268.165	1.088832
20	637.1161	570.0494	1.117651
25	331.3792	288.7042	1.147816
30	187.6002	158.9898	1.179951
35	113.1623	93.15933	1.214718
40	71.72498	57.24731	1.252897
45	47.30807	36.51764	1.295486
50	32.244	23.99406	1.343832
55	22.59059	16.13773	1.399862
60	16.20372	11.04937	1.466484
65	11.86098	7.660099	1.548411
70	8.837601	5.343157	1.654004
75	6.688775	3.715622	1.800177

APPENDIX C

SLAVE COMPUTER PROGRAM

```
;
;
MADDR     EQU    180H         ;STARTING ADDRESS OF 8256 MUART

MODE      equ    186H
P1        equ    188H         ;Port1  control
P2IO      equ    192H         ;Port2 IO register
T2        equ    196H
T3        equ    198H
T4        equ    19aH
T5        equ    19cH

BREAK     EQU    018H         ;IMMEDIALTY BREAK FROM ROUTINE--CTRL X.
XON       EQU    011H         ;TRANSMIT ON CHARACTER
XOFF      EQU    013H         ;TRANSMIT OFF CHARACTER
HOLD      EQU    016H         ;HOLD LOOP IDENTFIER-- CTRL V.
MUXCV     EQU    0fffbh       ;A/D channel address
ADCHI     EQU    0fffdh       ;A/D high byte data
ADCLO     EQU    0fffch       ;A/D low byte data
DABASE    EQU    0ff00h       ;D/A base address
TALK      equ    0aah
C8256     equ    0ddH         ;2-4 & 3-5 counters, port2 = output
P1IN      equ    00H          ;port1 is input
HBYTE     EQU    0ffH         ;write higher byte cause HBYTE * 256 + 255 to be
                              ;stored
;
;<timer address>
;

TMR0          equ       0ff50H        ;timer0 register
MAXA0         equ       0ff52H        ;maximum count A, timer0
MAXB0         equ       0ff54H        ;maximum count B, timer0
MTMR0         equ       0ff56H        ;timer0 mode control

TMR1          equ       0ff58H        ;timer1 register
MAXA1         equ       0ff5aH        ;maximum count A, timer1
MAXB1         equ       0ff5cH        ;maximum count B, timer1
MTMR1         equ       0ff5eH        ;timer1 mode control

TMR2          equ       0ff60H        ;timer2 register
MAXA2         equ       0ff62H        ;maximum count A, timer2
MTMR2         equ       0ff66H        ;timer2 mode control

Counter       equ       0c00dH        ;external event counter
Timer         equ       0c009H        ;timer1 controlled by timer2
Prescale      equ       0c001H        ;trig timer2 as a prescaler
scale         equ       250           ;NOTE this is in decimal.
                                      ;250 * 8e-7 gives 200 microseconds
ATD_Int       equ       1110000000001001B     ; e009h scaled by timer2, int. e
T10msec       equ       50            ;200 usec * 50 = 10 msec
T500msec      equ       2500          ;200 usec * 2500 = 500 msec
T1sec         equ       5000          ;200 usec * 5000 = 1 sec

max           equ       0ffffH        ;maximum count
clr           equ       0000H         ;clear count. this gives 65536 max.
example       equ       50000        ;decimal

;
;
;   IEEE-488 names
```

```
;
auxcmd          equ         03h

reset           equ         80h
rstclr          equ         00h

init55          equ         02h
adr_8255        equ         0ch
addr            equ         04h
int0            equ         00h
int1            equ         01h
adstat          equ         02h

hdfa            equ         83h
hdaclr          equ         03h

eoimk           equ         08h
bim             equ         20h
bom             equ         10h
din             equ         07h
rhdf            equ         02h
etx             equ         03h

dout            equ         07h
feoi            equ         08h

MA              equ         04h
LADS            equ         04h
TADS            equ         02h
ZERO            equ         30h  ;ASCII 0
ONE             equ         31h  ;ASCII 1
Last            equ         02h

;<decoding INSTRUCTION SET>

Do_ATD         equ         00h  ;Single shot A/D
SingDTA         equ         01h  ;Single shot D/A
SelecATD        equ         0ah  ;selective ATD
ISelecATD       equ         0bh  ;selective ATD with resetting of timer
Int_ATD          equ         0ch  ;start interrupt ATD
Trans           equ         0dh  ;tansfer data from ATDstore to databuf

Freq            equ         010h ;counter-(timer0) & timer-(timer1)
IO              equ         012h ;io control on 8255 (IEEE488 card)


;<array>

Quit            equ         0ffh  ;Quit decoding

;<distalk>
Quitalk         equ         0ffh   ;quit talking

ZEROV           equ         0800H  ;offset binary equivalent 0 volt

;<Slective ATD>
ten             equ          10
uno             equ          1
```

```
;<interrupt >

int8ip          equ     8h
timer_int_ctrl  equ     0ff32h
eoi_register    equ     0ff22h
non_spec        equ     8000h
```

```
public    count, array, distalk, databuf, ATDstore, MAXCOUNT, POINTER
public    saveINT8cs, saveINT8ip, memset, memPoint
;
dataseg   segment

          count         dw         ?       ;note count stores word number info.
                                           ;NOT bytes

          MAXCOUNT      dw        ?
          POINTER       dw        ?
          saveINT8cs    dw        ?
          saveINT8ip    dw        ?
          databuf       dw        800 dup(?)
          ATDstore      dw        800 dup(?)
          memPoint      dw        ?
          memset        db        ?
          distalk       db        ?
          array         db        3 dup(?)
dataseg   ends
;

include    b:name.lst


;
codeseg   segment
;

public    crlf, conv, bout, CHROUT

extrn     IntrptATD:near, ISATD:near, SATD:near, IO1:near, CF:near, DTA:near
extrn     DTAS:near, ATDS:near, ATD:near, data_trans:near
;
ASSUME    CS:codeseg,DS:dataseg
;
;
;


;
; ***************************** Segment Assignment ****************
;

          mov           ax,dataseg       ;Stack grows downward; therefore,
          mov           ss,ax            ;the address can be ss=ds
          mov           es,ax             ;es is data buffer segment
          mov           ds,ax
;
; **************************************************************
;
; ************************************************************
;
;     Listener & Talker rotuines
;
;        Programmed by Ito, Tokunosuke
;
; ************************************************************
;
              call      IRS232
```

```
              call      initlt                  ;initialize 9914 as listener
              call      initmr                  ;initialize timer
              call      initDTA
              call      initmem                 ;initialize memory
st2:          call      scan          ;scan to determine talker or listener
              jmp       st2
;
;
; ------------------------------------
;
; initmem  ---- initialize memory
;                  memset = 00h
;
;    Entry & Exit condition ---- none
;
; -----------------------------------
;
initmem         proc            near
                push            ax
                push            di

                mov             di,offset memset
                mov             al,00h
                mov             [di],al

                pop             di
                pop             ax
                ret
initmem         endp
;
;
; ------------------------------------
;
; initDTA  --- initialize DTA to 0 voltage
;              this is necessary so that external devices will not be accidental
;              driven
;
;    Entry &  Exit condition ---- none
;
; ---------------------------------------------
;
initDTA:      push      ax
              push      dx

              mov       ax,ZEROV           ;set 0 volt
              mov       dx,0               ;channel zero
              call      DTA
              mov       dx,1               ;channel 1
              call      DTA
              mov       dx,2               ;channel 2
              call      DTA
              mov       dx,3               ;channel 3
              call      DTA

              pop       dx
              pop       ax
              ret
;
;
;
```

```
;
; ----------------------------------------------
;
;   initmr ---- initialize timer routine
;
;       Entry & Exit condition --.-- none
;
; -------------------------------------------
;
initmr:         push      ax
                push      dx

                 mov       dx,MAXA0                 ;timer0 maxA count = ffffH
                 mov       ax,max
                 out       dx,ax

                 mov       dx,MTMR0                 ;initialize tmr0 as counter
                 mov       ax,Timer
                 out       dx,ax

                mov       dx,MAXA2                 ;initialize timer2 with scaled
                mov       ax,scale                 ;value
                out       dx,ax

                mov       dx,MTMR2                 ;initialize timer2 as prescaler
                mov       ax,Prescale
                out       dx,ax

                mov       dx,MAXA1                 ;set maxA count of timer1
                mov       ax,example
                out       dx,ax
;                mov       dx,MAXB1
;                out       dx,ax

                mov       dx,MTMR1                 ;initialize tmr1 as event timer
                mov       ax,Timer
                out       dx,ax

                mov       dx,MODE                  ;initialize 8256
                mov       al,C8256                 ;2-4 & 3-5 counter, port2 outpu
                out       dx,al

                mov       dx,P1                    ;initialize port1
                mov       al,P1IN                  ;as input
                out       dx,al

                mov       dx,T4                    ;set initial value
                mov       al,HBYTE                 ;ffffH
                out       dx,al
                mov       dx,T5
                out       dx,al

                mov       dx,P2IO                  ;initialize port2 register
                mov       al,HBYTE                 ;set all bits high
                out       dx,al

                mov       ax,clr
                mov       dx,TMR1                  ;reset timer
                out       dx,ax
                mov       dx,TMR2                  ;reset prescaler
```

```
                out         dx,ax
                pop         dx
                pop         ax
                ret
;
;
; ----------------------------------------------------------
;
;    scan  --- scan address status and trigger
;              listener or talker routine
;
;        Entry condition ----- none
;
;        Exit condition ---- none
;
; -------------------------------------------------------
;
scan:           push        dx              ;save registers to make
                push        si
                push        cx
                push        ax              ;independent routine


                mov         dx,adstat       ;get address of address status register
                in          al,dx           ;get content

                test        al,LADS         ;listener ?
                jz          next            ;no.

                mov         al,'L'
                call        CHROUT

                call        listen      ;listener primary address start

                call        decode          ;decode what has been received

                jmp         return          ;return

next:           test        al,TADS
                jz          return          ;nop

                mov         al,'T'
                call        CHROUT
                call        inittalker      ;talker !

;*b
                mov         al,'*'
                call        CHROUT
;*e


return:         pop         ax
                pop         cx
                pop         si
                pop         dx
                ret
;
;
;
```

```
; ---------------------------------------------
;
;  relax  --- delay loop
;
;    Entry & EXit condition --- none
;
; -------------------------------------------
;
relax:          push      cx
                mov       cx,0fffh
bushed:         nop
                loop      bushed
                pop       cx
                ret
; --------------------------------------------
;
;


; -------------------------------------------------------
;
;  decode ----- decoding the instructions
;
;    Entry conditions  ----- databuf has input data
;
;     Exit condition  ---- databuf has the result
;
; -----------------------------------------------
;
decode:         push      si
                push      ax              ;
                push      es

                mov       ax,ds
                mov       es,ax           ;es = ds

                mov       si,offset array
                mov       al,es:[si]          ;get instruction
                cmp       al,Quit             ;quit decoding ? Quit=0ffh
                jnz       notq                ;get out this routine promptly
                mov       al,00h              ;clean 1st byte of array
                mov       es:[si],al
                jmp       lsel

notq:           mov       si,offset databuf
                mov       al,es:[si]
                cmp       al,Do_ATD           ;is this ATDS
                jnz       sel2                ;no. Go to 2nd choice
                inc       si                  ;make si points the next byte
                call      ATDS                ;NOTE si,ax must be preserved
                jmp       lsel

sel2:           cmp       al,SingDTA          ;is this DTAS
                jnz       sel3                ;no. jump
                inc       si                  ;point to next byte
                call      DTAS
                jmp       lsel

sel3:           cmp       al,Freq             ;timer & counter ?
                jnz       sel4                ;if not jump
                mov       al,'F'
```

```
                call    CHROUT
                call    CF
                jmp     lsel                ;any more info.

sel4:           cmp     al,IO               ;IO control ?
                jnz     sel5
                inc     si                  ;point to the next byte
                call    IO1
                jmp     lsel

sel5:           cmp     al,SelecATD
                jnz     sel6
                call    SATD                ;call selective ATD
                jmp     lsel

sel6:           cmp     al,ISelecATD       ;initialize timer
                jnz     sel7
                call    ISATD
                jmp     lsel

sel7:           cmp     al,Int_ATD
                jnz     sel8
                inc     si                  ;si points to no. of ATD conversion
                call    IntrptATD
                jmp     lsel

sel8:           cmp     al,Trans
                jnz     lsel
                inc     si
                call    data_trans
                jmp     lsel

lsel:           pop     es
                pop     ax
                pop     si
                ret
;
; -----------------------------------------------------
;
;   bout   --- binary output
;
;       Entry condition --- ax contains a word to be outputted
;
;       Exit condition ----  none
;
; -----------------------------------------------------
;
bout:           push    cx
                push    ax          ;save register
                mov     cl,16       ;one word rotation
trya:           rol     ax,1        ;left rotate
                jc      p1          ;jump if carry
                push    ax
                mov     al,ZERO     ;no jump, send 0
                call    CHROUT
                pop     ax
                jmp     p2          ;jump to loop

p1:             push    ax
                mov     al,ONE      ;send 1
```

```
                call      CHROUT
                pop       ax

p2:             loop      trya
                pop       ax
                pop       cx
                ret
;

; ----------------------------------------------------
;
;    rdchr --- read character
;
;        Entry condition ----- none
;
;        Exit condition ----- none
;
; ----------------------------------------------------
;
rdchr:            call        CHRIN
                  call        CHROUT
                  ret
;
; -----------------------------------------------------
;
;     listen - listener routines
;
;       Entry condition ----- none
;
; -----------------------------------------------------
;
listen:         push      ax
                push      dx
                push      si

                mov       dx,auxcmd
                mov       al,hdfa
                out       dx,al

                lea       si, databuf

ls1:            mov       dx,adstat               ;check for talker
                in        al,dx
                test      al,TADS                 ;talker ?
                jnz       get_out                 ;yes! get out listener routine

                mov       dx,int0                 ;BI or EOI ?
                in        al,dx
                and       al,eoimk+bim
                jz        ls1                     ;not BI or EOI

                call      recvlt                  ;yes! call recvlt

                lea       si,distalk              ;enable talk
                mov       al,00h
                mov       es:[si],al

                jmp       ls2
```

```
get_out:        lea       si,array
                mov       al,0ffh                 ;tellling decode to quit
                mov       es:[si],al

ls2:            mov       dx,auxcmd
                mov       al,hdaclr
                out       dx,al

                pop       si
                pop       dx
                pop       ax
                ret
;
;
; -----------------------------------------------------------
;
;     inittalker --- talker routines
;
;     Entry condition ---- none
;
; ----------------------------------------------------------
;
;
inittalker:     push      si
                push      ax

                mov       si,offset memset
                mov       al,[si]                 ;byte transfer
                cmp       al,00h
                jnz       altl

                mov       si,offset databuf       ;get pointer to outut buffer
                jmp       resetmem

altl:           mov       si,offset memPoint      ;get starting address
                mov       ax,[si]
                mov       si,offset ATDstore      ;the beginning of ATDstore
                add       si,ax                   ;add offset

resetmem:       call      sendlt                  ;send them to gpib
                mov       si,offset memset
                mov       al,00h
                mov       [si],al

;*b
                mov        al,'E'
                call       CHROUT
;*e

                pop       ax
                pop       si
                ret
;
;
; ----------------------------------------------------------------------------
;
;    initlt ---- initialize 9914   seems to work for both talker & listener
;
;      Entry condition -------  none
;
```

```
;       Exit condition  --------  none
;
; ---------------------------------------------------------------------
;
initlt:         push      ax                          ;save working register
                push      dx
                mov       dx,auxcmd                   ;point to auxcmd (03h) register
                mov       al,reset                    ;reset (80h)
                out       dx,al
                mov       al,rstclr                   ;rstclr (00h)
                out       dx,al

;
                mov        al,init55                    ;initialize 8255 init55 (02h)
                mov        dx,adr_8255+03h                ;load control register addr
                out       dx,al

                mov        dx,adr_8255+1                ;get switch address
                in        al,dx                        ;get switch settings

                and        al,01fh                        ;protect 5 lowest bits
                mov        dx,addr                        ;point to addr register of
                out        dx,al                        ;addr (04h)
                xor        al,al                        ;clean up al
                mov        dx,int0
                out        dx,al
                mov        dx,int1
                out        dx,al                        ;mask out all interrupt

                pop        dx
                pop        ax                            ;recover registers
                ret

;
;
; --------------------------------------------------------
;
;      string ----- outputting the string of data
;
;       Entry condition ---------- si contains the effective address
;
;       Exit condition ----------- si destroyed
;
; -------------------------------------------------------
;
string:           push         ax
checkl:           mov            al,cs:[si]                ;get a byte
              cmp           al,'$'                     ;is it yhe end of text
              je           fnprint                     ;yes jump
              call        CHROUT                     ;output it to screen
              inc            si
              jmp            checkl
fnprint:      pop            ax
              ret
;
;
; -------------------------------------------------------
;
;       recvlt ------ receive data
;                      data is received until EOI is sent with
```

```
;                       last byte ETX is added as last byte to
;                       indicate end of buffer
;
;         Entry condition ------ si points to data buffer
;                                al contains the status of int0
;
;        Exit condition ------- si only have been destroyed
;
; -------------------------------------------------------------
;
recvlt:     push        ax
            push        cx
            push        dx                          ;save ax & dx
            push        si
            push        di
            jmp         recv                        ;jump int0 state acquisition

recvlt1:    mov         dx,int0
            in          al,dx                       ;int0 (00h) contains status

recv:       and         al,eoimk+bim               ;EOI or BI ?
            jz          recvlt1                      ;wait until EOI or BI is set
            push        ax                          ;save status byte
            and         al,eoimk                   ;test for EOI
            jnz         recvlt2                      ;jump if EOI
            pop         ax                          ;restore status

            mov         dx,din                      ;now we know that status has B
            in          al,dx                      ;get data byte din (07h)
            mov         es:[si],al                  ;save

            mov         dx,auxcmd
            mov         al,rhdf                      ;rhdf-release RFD holdoff
            out         dx,al                      ;rhdf (02h)
            inc         si
            jmp         recvlt1                      ;get more

recvlt2:    pop           ax                          ;restore status byte
;
recvlt2a:   and           al,bim                      ;byte in ?
            jnz           recvlt5                      ;if so jump. bim (20h)
            mov           dx,int0
            in           al,dx
            jmp           recvlt2a                   ;different from original !!!

recvlt5:    mov           dx,din
            in            al,dx
            mov           es:[si],al                  ;store away

            inc           si                          ;this is to leave ETX in da
            mov           al,etx                      ;buffer. ETX (03h)
            mov           es:[si],al

recvlt3:    mov           dx,auxcmd
            mov           al,rhdf                      ;release Holdoff
            out           dx,al

   ;         mov           dx,auxcmd
   ;         mov           al,hdaclr                  ;clear Hold-off
   ;        out           dx,al                     ;to 9914
```

```
            pop             di
            pop             si
            pop             dx
            pop             cx
            pop             ax
            ret
;
;
;
; ------------------------------------------------------------
;
;    sendlt --- send data to IEEE488
;
;     Entry condition ---- es = ds
;                          si contains the address of data buffer
;
;     Exit condition ----  si destroyed
;
; -----------------------------------------------------------
;
sendlt:
            push         ax
            push         cx
            push         dx
            push         di
            push         es

            mov          ax,ds
            mov          es,ax                  ;es = ds

            mov          di,offset distalk      ;check previous process
            mov          al,es:[di]
            cmp          al,Quitalk
            jz           disable

            mov          di,offset count         ;get numbet of words to transf
            mov          cx,es:[di]
            mov          ax,cx                  ;get no. of bytes from words
            add          cx,ax                  ;cx = content of count * 2

;*b
            push           cx
            push           si
            call           crlf
            mov            al,'N'
            call           CHROUT
            mov            ax,cx
            call           conv
            pop            si
            pop            cx
;*e

sendltl:    mov              al,es:[si]             ;si already pionts to the t
                                                 ;of memory set by memset
            mov              dx,dout             ;if not send out.
            out              dx,al

;            call             bout                ;check

sendwait:    mov             dx,int0
```

```
          in          al,dx               ;get status
          and          al,bom               ;check byte out mask bom (10h)
          jz          sendwait            ;if BO set then jump to repeat

          cmp          cx,Last            ;is this one before the last byte
          jnz          sendlt2

          mov          dx,auxcmd          ;send EOI
          mov          al,feoi
          out          dx,al

sendlt2:  inc          si
          loop         sendlt1            ;

          mov         di,offset distalk   ;mark that talk process is finishe
          mov         al,Quitalk
          mov         es:[di],al

disable:  pop          es
          pop          di
          pop          dx
          pop          cx
          pop          ax
          ret
;
```

```
;
; ----------------------------------------------------------------
;
;   crlf   subroutine   - sends CR & LF  to consol
;
;          Entry condition ------- none
;
;          Exit condition ------- destroies none
;
; ---------------------------------------------------------------
;
crlf:          push      ax           ;save ax register
               mov       al,0ah       ;line feed
               call      CHROUT       ;out to consol
               mov       al,0dh       ;carriage return
               call      CHROUT       ;send it to consol
               pop       ax           ;restore the register
               ret                    ;back to main routine
;
;
; ----------------------------------------------------------------
;
;  conv      subroutine
;
```

```
;               Entry condition -------- ax has binary number
;
;               Exit condition ---------
;
; ---------------------------------------------------------------
;
conv:   push    ds              ;save registers
        push    di
        push    dx
        push    cx              ;
        push    bx
        push    ax              ;ax is stored temporalilly
        mov     ax,dataseg      ;get dataseg in place
        mov     ds,ax
        lea     di,array        ;get pointer to array
        mov     al,0            ;clean up of the array
        mov     cx,3            ;0-2, 3 bytes
cleanup: mov    [di],al         ;cleanup loop
        inc     di              ;next byte
        loop    cleanup

        pop     dx              ;get dx(16 bit binary) back formally in ax
        lea     di,array        ;get pointer to array again
getbcd: mov     ax,dx
        mov     dx,0            ;cleanup upper byte
        mov     cx,10           ;divisor of 10
        div     cx              ;(dx:ax)/cx => ax=quotient,dx=remainder
        xchg    ax,dx           ;ax=remainder, dx=quotient
        mov     bl,al           ;bl has the remainder (al & bl are less than 1
        mov     ax,dx           ;ax=dx=quotient
        mov     dx,0            ;cleanup upper byte
        mov     cx,10           ;divisor of 10
        div     cx              ;(dx:ax)/cx => ax=quotient, dx=remainder
        xchg    ax,dx           ;dx=quotient, ax=remainder
        mov     cl,4            ;rotate left 4 bits
        rol     al,cl
        and     al,0f0h         ;clean up the last 4 bits
        or      al,bl           ;combine and the result is in al
        mov     [di],al         ;store away
        inc     di
        cmp     dx,0            ;if dx(quotient)=0, finish.
        jnz     getbcd          ;still remaining to convert
;
;   output ASCII converted from BCD
;
        dec     di              ;point to the last byte stored
;        mov     dh,0            ;test against zero
;        mov     bl,0            ;if bl=0, leading 0. if bl=1, non leading 0
cont:   mov     al,[di]         ;get packed BCD byte
        dec     di              ;and point to next
        mov     dl,al           ;save it in dl
        mov     cl,4            ;rotate left 4 times
        rol     al,cl
        and     al,0fh          ;first BCD
;        or      dh,al           ;dh=0
;        jz      bcdl            ;if leading zero, skip!
;        inc     bl              ;from now on zero will be sent as number
        add     al,30h          ;add 48D(30h) to make ASCII
        call    CHROUT          ;send it
        mov     al,dl           ;dl has the original packed BCD
```

```
        and       al,0fh          ;lower nipple
;       or        bl,al           ;if both bl & al are zero, then leading zero
;       jz        bcd2            ;jump if leading zero
;       inc       bl              ;not leading zero
        add       al,30h          ;make ASCII
        call      CHROUT

        mov       bx,offset array
        cmp       di,bx           ;if di >= bx then jmp
        jge       cont

;
        pop       bx
        pop       cx
        pop       dx
        pop       di
        pop       ds
        ret
;
;-----------------------------------------------------------------


;
;
CHRINI:
        CALL    IRS232
        MOV     AL,2AH          ;ASCII- (*)
        CALL    CHROUT
;
;
        MOV     CX,500          ;SET LOOP COUNTER
;
INLOOP: MOV     AX,1
        CALL    DELAY           ;WAIT FOR TIME DELAY
        CALL    CSTATUS           ;CHECK FOR CHARACTER AVAILABLE
        JNZ     SETBAUD           ;IF AVAILABLE GET CHARACTER
back:   LOOP    INLOOP          ;KEEP TRYING
        JMP     CHRINI          ;TRY STARTING OVER
;
;
SETBAUD:push    ax              ;save key registers
        push    cx
        CALL    CHRIN           ;GO GET FIRST CHARACTER
        AND     AL,5FH          ;MASK BITS
        CMP     AL,55H          ;CHECK FOR (U) INPUT
        jz      ENDINIT         ; yes u or U
        pop     cx              ;restore registers
        pop     ax
        jmp     back            ;back to the road

;
ENDINIT:    RET             ;RETURN INITIALIZATION COMPLETE
;
;
;
;
;
;    SOFTWARE DELAY LOOP (AX) NUMBER OF ITTERATIONS ON ENTRY
;
;
DELAY:  PUSH    CX
```

```
DELAY0: PUSH    AX
        MOV     CX,50
DELAY1: MUL     AL
        LOOP    DELAY1
        POP     AX
        DEC     AX
        JNZ     DELAY0
        POP     CX
        RET
;
;
;
;
;
;    CHROUT:    CHARACTER OUTPUT ROUTINE
;            DATA IN (AL) IS OUTPUT TO THE CONSOLE
;
;
CHXOUT:
CHROUT:         PUSH    DX
                PUSH    AX
                xor     ah,ah
WRCH1:          CALL    CSTATUS
                JZ      WRCHR2          ;nothing coming in, transmit!
                CALL    CHRIN           ;something on the line
                CMP     AL,BREAK        ;Is it break ?
                JZ      RESTART         ;If so jump.
                CMP     AL,HOLD         ;CAUSE CHROUT ROUTINE TO PAUSE
                JNZ     WR1             ;if not hold jump to check XOFF
                CALL    RS_WAIT         ;
WR1:            CMP     AL,XOFF         ;CHECK FOR XOFF
                JNZ     WRCHR2         ;JMP IFF NOT
;
WRCH2:          CALL    CSTATUS
                JZ      WRCH2          ;WAIT FOR X_ON
                CALL    CHRIN
                CMP     AL,BREAK
                JE      RESTART
                CMP     AL,HOLD
                JNE     WR2
                CALL    RS_WAIT
WR2:            CMP     AL,XON
                JNZ     WRCH2          ;JUMP IF NOT X-ON
;
;
WRCHR2:         MOV     DX,MADDR+1EH    ;POINT TO STATUS REG
                IN      AL,DX
                TEST    AL,20H          ;CHECK FOR TRANSMIT BUFFER EMPTY
                JZ      WRCH1          ;WAIT IF BUSY
                POP     AX         ;GET CHARACTER BACK
                MOV     DX,MADDR+0EH    ;ADDRESS OF TRANSMIT BUFFER
                OUT     DX,AL          ;SEND IT
                POP     DX
                RET            ;RETURN TO CALLER
;
RESTART:        ret           ;GO TO WARM START
;
;
RS_WAIT:
                CALL    CHRIN
```

```
                CMP     AL,BREAK     ;IF CTRL-X IS ENTERED, JUMP TO WARM START
                JE      RESTART
                CMP     AL,HOLD         ;IF CTRL-V IS ENTERED, WAIT UNITL ANOTHER
                JNE     RS_WAIT         ;IS ENTERED.
                RET
;
;
;
;
;
;    CHRIN:     CHARACTER INPUT ROUTINE
;             CHARACTER WILL BE RETURNED IN THE (AL) REG
;
;
CHXIN:
CHRIN:  CALL     CSTATUS
        JNZ     CHRIN1
        JMP     CHRIN
CHRIN1:         PUSH    DX
        MOV     DX,MADDR+0EH     ;INPUT BUFFER ADDRESS
        IN      AL,DX        ;GET CHARACTER AVAILABLE
        STC               ;SET OK FLAG (NO TIME OUT)
        POP     DX          ;RESTORE REGISTERS
        RET               ;RETURN TO CALLER
;
;
;
;
;
;
;    CSTATUS:    CONSOLE STATUS ROUTINE
;             STATUS INFORMATION RETURNED IN (AL) REG.
;
;        (AL) = 00H    NO CHARACTER AVAILABLE
;        (AL) = 0FFH    CHARACTER AVAILABLE
;
CSTATUS:PUSH     DX          ;SAVE REGISTWER TO BE USED
        MOV      DX,MADDR+1EH    ;ADDRESS OF STATUS REGISTER
        IN       AL,DX         ;GET STATUS BYTE
        AND      AL,40H         ;MASK OUT BITS
        JNZ      CON1          ;JUMP IF AVAILABLE
        POP      DX          ;RESTORE REGISTERS
        XOR      AL,AL         ;00 INTO (AL) REG
        RET
CON1:   POP      DX          ;RESTORE REGISTERS
        OR       AL,0FFH         ;0FFH INTO (AL) REG
        RET
;
;
;
;
;
;    CONSOLE INITIALIZATION BEGINS HERE
;
IRS232:
        PUSH    AX
        PUSH    DX
        PUSH    SI
        PUSHF
        MOV    DX,MADDR+0H
```

```
        MOV     AL,02H
        OUT     DX,AL           ;SET 8086 MODE
;
        MOV     AL,034h         ;9600 baud rate
        MOV     DX,MADDR+2H
        OUT     DX,AL            ;SET BAUD RATE
;
        MOV     AL,0C0H
        MOV     DX,MADDR+4H
        OUT     DX,AL           ;ENABLE RECEIVER
        AND     AL,81H
        MOV     DX,MADDR+4H
        OUT     DX,AL           ;RESET MUART
        POPF
        POP     SI
        POP     DX
        POP     AX
        RET
;
;
msg1    db         'Ad sta   $'

msg2    db         'int0 status is   $'

msg3    db         'Enetered Talker State... $'

msg4     db        'Ready to receive data...(Listener) $'

msg5    db         'End detected in int0    $'

msg6    db         'GPIB status ...  $'

msg7    db         'int1 status is   $'
;
;
codeseg  ends
;
;
    END
```

```
dataseg         segment
                extrn count:word, array:byte, distalk:byte, databuf:word
                extrn ATDstore:word, MAXCOUNT:word, POINTER:word
                extrn saveINT8cs:word, saveINT8ip:word, memset:byte
                extrn memPoint:word
dataseg         ends

include         b:name.lst


codeseg         segment

ASSUME          cs:codeseg, ds:dataseg

                public  IntrptATD, ISATD, SATD, IO1, CF, DTAS, DTA
                public  ATDS, ATD, data_trans

                extrn   crlf:near, conv:near, bout:near, CHROUT:near
;
;
; ------------------------------------
;
;  data_trans  ----- data transfer from ATDstore to databuf
;                                 from MAXCOUNT to count
;
;    Entry & Exit condirion ------ none
;
; ---------------------------------------
;
data_trans      proc     near
                push     ax
                push     bx
                push     cx
                push     dx
                push     es
                push     si
                push     di


                mov      ax,[si]                  ;si contains the pointer to
                                                  ;the number of words to be
                                                  ;transfered
                shl      ax,1                     ;2 channels (double the word
                                                  ;number)

                mov      di,offset count
                mov      [di],ax
                inc      si
                inc      si                       ;get next word which contains
                                                  ;the beginning of the address
                                                  ;for data transfer
                mov      ax,[si]                  ;address is in ax
                mov      cl,2                     ;4*ax to get proper offset
                shl      ax,cl
                mov      di,offset memPoint       ;store back into memPoint
                                                  ;so that inittalker routine
                mov      [di],ax                  ;can figure out the correct
                                                  ;address
```

```
                mov     si,offset memset          ;memory set for ATDstore
                mov     al,01h
                mov     [si],al


                mov     bx,offset count             ;get no. of words to be
                mov     cx,[bx]                     ;transfered
                mov     bx,offset ATDstore          ;base address
                mov     di,offset memPoint          ;offset
                add     bx,[di]                     ;start address = base + offse

show:           mov     ax,[bx]
                call    conv
                call    crlf
                inc     bx
                inc     bx
                loop    show

                pop     di
                pop     si
                pop     es
                pop     dx
                pop     cx
                pop     bx
                pop     ax
                ret
data_trans      endp
;
;
; ---------------------------------
;
;  IntrptATD  ----- tap timer interrupt routine
;
;     Entry condition ---- si points the next word (count)
;
;     Exit ----- none
;
;
; ---------------------------------
;
IntrptATD:      push    ax
                push    bx
                push    dx
                push    es
                push    si


                mov     ax,ds               ;set ax = ds
                mov     es,ax               ;es = ds

;*b
                xor     ax,ax
                mov     al,'R'              ;interrupt tap enter
                call    CHROUT
                call    crlf
;*e
                pop     si                  ;recover si
```

```
push       si
mov        ax,es:[si]           ;si points the count
                                ;es=ds

mov        si,offset MAXCOUNT   ;store count in MAXCOUNT


mov        es:[si],ax           ;es = ds

xor        ax,ax                ;clean up ax
mov        si,offset POINTER    ;clean up pointer
mov        es:[si],ax

cli                             ;disable interrupt

xor        ax,ax                ;ax = 00h
mov        es,ax                ;es = 00h

mov        si,4*int8ip            ;instruction pointer of type 8 int

mov        ax,es:[si]
mov        bx,offset saveINT8ip ;save old instruction pointer
mov        [bx],ax

inc        si
inc        si                   ; a word apart for code segment


mov        ax,es:[si]           ;save codesegement
mov        bx,offset saveINT8cs
mov        [bx],ax

mov        si,4*int8ip
mov        es:[si],offset timer_int   ;es = 0000h
inc        si
inc        si
mov        es:[si],cs           ;patching pointers complete

mov        dx,MAXA0              ;set 10 msec interrupt
mov        ax,T10msec
out        dx,ax

mov        dx,MTMR0             ;enable timer interrupt
mov        ax,ATD_Int
out        dx,ax

mov        dx,timer_int_ctrl     ;unmask interrupt
mov        ax,clr                ;highest priority
out        dx,ax

mov        dx,TMR0               ;clean up timer0 & timer2
out        dx,ax                 ;ax = 0000h from the previous code
mov        dx,TMR2
out        dx,ax

sti                              ;enable processors interrupt

pop        si
pop        es
pop        dx
```

```
              pop       bx
              pop       ax
              ret
;
;
; ---------------------------------------
;
;  timer_int ---- timer interrupt
;
;     Entry & Exit condition ---- none
;
; ---------------------------------------
;
timer_int     proc      near
              push      ax
              push      bx
              push      cx
              push      dx
              push      es
              push      si
              push      di

              mov       ax,ds                     ;ax = ds
              mov       es,ax                     ;es = ds

              mov       bx,offset POINTER         ;get present pointer
              mov       ax,[bx]
              inc       ax                        ;increment by 1
              mov       [bx],ax                   ;store it back right away !


              mov       si,offset MAXCOUNT        ;compare POINTER with MAXCOUNT
              mov       bx,es:[si]                ;get MAXCOUNT       es=ds
              cmp       ax,bx
              jg        finito                    ;if pointer is greater than MAX
                                                  ;then go to finish routine
              dec       ax                        ;decrement to start from 0

              mov       cx,2                      ;multiply ax with 4 by 2 shift
              shl       ax,cl
              mov       si,offset ATDstore
              add       si,ax                     ;si + ax(pointer) gives the beg
                                                  ;of the storage

              mov       ax,2                      ;2nd  channel (position)
              call      ATD
              mov       es:[si],ax                ;store    es=ds

              xor       ax,ax                     ;clean ax to get 0 channel (for
              add       si,2                      ;increment pointer by a word
              call      ATD
              mov       es:[si],ax                ;es=ds

out_of_int:   mov       dx,eoi_register           ;non-iRMAX 86 non specific
              mov       ax,non_spec               ;end of interrupt
              out       dx,ax

              jmp       end_int
```

```
finito:         cli                                     ;clear interrupt

;*b
                mov     al,'E'
                call    CHROUT
                call    crlf
;*e

                xor     ax,ax                           ;ax = 0000h
                mov     es,ax                           ;es = 0000h

                mov     bx,offset saveINT8ip            ;repatch the old info.
                mov     ax,[bx]                         ;get saved ip info.
                mov     si,4*int8ip                     ;get the address of int8
                mov     es:[si],ax                      ;repatch ip  es=0000h

                mov     bx,offset saveINT8cs            ;get address where cs saved
                mov     ax,[bx]                         ;transfer contyent to ax
                inc     si
                inc     si                              ;get address of int8cs
                mov     es:[si],ax                      ;es = 0000h

                mov     dx,MTMR0                        ;disable timer interrupt
                mov     ax,Timer
                out     dx,ax

                sti                                     ;repatch ended. enable interrup

                jmp     out_of_int

end_int:        pop     di
                pop     si
                pop     es
                pop     dx
                pop     cx
                pop     bx
                pop     ax

                iret

timer_int       endp

;
;
; -----------------------------------
;
;  ISATD  ------ SATD with timer initialization
;
;     Entry & Exit  condition ----- none
;
; -----------------------------------
;
ISATD:          push    ax
                push    dx
                mov     dx,TMR0                         ;reset timer0
                mov     ax,clr
                out     dx,ax
                mov     dx,TMR2
```

```
                out     dx,ax
                call    SATD
                pop     dx
                pop     ax
                ret
;
; ------------------------------------------
;
;   SATD ---- selective ATD
;
;     Entry & Exit condition ---- none
;
; --------------------------------------
;
SATD:           push    ax
                push    bx
                push    cx
                push    dx
                push    si


                lea     si,count
                mov     es:[si],3               ;3 words transfer


                lea     si,databuf              ;load pointer to databuf
                mov     ax,clr                  ;clean up buffer
                mov     es:[si],ax              ;3 words
                add     si,2                    ; 1-position
                mov     es:[si],ax              ; 2-force
                add     si,2                    ; 3-timer1 prescaled by timer2
                mov     es:[si],ax

                mov     ax,clr                  ;reset timer1 & timer2(prescale
                mov     dx,TMR1
                out     dx,ax

                mov     cx,ten                  ;# of data sets
                mov     bx,uno                  ;counter for 200 usec

again:          lea     si,databuf
                xor     ah,ah
                mov     al,2                    ;channel # of position
                call    ATD
                add     es:[si],ax              ;store in databuf
                xor     ax,ax                   ;next is 0 channel=force
                add     si,2                    ;advance pointer
                call    ATD                     ;get force
                add     es:[si],ax
                mov     dx,TMR1                 ;200 microsecond wait
again1:         in      ax,dx                   ;get timer1 count
                cmp     ax,bx
                jl      again1                  ;wait until 200 usec
                inc     bx                      ;increment for next
                loop    again

                lea     si,databuf
                mov     cx,2
                mov     bx,ten
```

```
again2:        xor       dx,dx
               mov       ax,es:[si]
               div       bx
               mov       es:[si],ax
               add       si,2
               loop      again2

               mov       dx,TMR0                    ;timer 1 count
               in        ax,dx
               mov       es:[si],ax

               pop       si
               pop       dx
               pop       cx
               pop       bx
               pop       ax
               ret
;
; ------------------------------------------
;
;   IO1 ---- Industrial IO control
;
;     Entry condition ------ si points to a byte data
;
;     Exit condition ------  none
;
; ------------------------------------------
;
IO1:           push      ax
               push      dx
               mov       al,es:[si]
               mov       dx,P2IO
               out       dx,al
               pop       dx
               pop       ax
               ret
;
;
; --------------------------------------------
;
;   CF ---- Timer0 is counter & timer1 is timer prescaled by
;           timer2 at 200 microseconds.
;
;     Entry conditon ------ none
;
;     Exit condition ------ databuf contains data as follows
;
;       low byte counter: high byte counter: low byte timer : high byte timer
;
; --------------------------------------------
;
CF:            push      ax
               push      bx
               push      dx
               push      si

               lea       si,databuf

               mov       dx,T4
               in        al,dx
```

```
            mov     ah,al
            mov     dx,T2
            in      al,dx

            mov     bx,ax
            mov     ax,max
            sub     ax,bx

            mov     es:[si],ax
            call    bout
            call    crlf
            add     si,2
            mov     dx,TMR1
            in      ax,dx
            mov     es:[si],ax
            call    bout
            call    crlf
            call    crlf

            lea     si,count
            mov     ax,0002h            ;2 words (means 4 bytes in sendl
            mov     es:[si],ax

            mov     ax,clr              ;clean up routine
            mov     dx,TMR1             ;reset timer1
            out     dx,ax
            mov     dx,TMR2             ;reset timer2
            out     dx,ax

            mov     dx,T4
            mov     al,HBYTE
            out     dx,al

            pop     si
            pop     dx
            pop     bx
            pop     ax
            ret
;



;
; --------------------------------------------------
;
;   DTAS  ---- single shot ATD
;
;     Entry condition ----- si points to the channel data
;
;     Exit condition ------ si destroyed
;
; -------------------------------------------------------
;
DTAS:       push    ax
            push    dx
            mov     dl,es:[si]           ;get channel number
            xor     dh,dh
            inc     si
```

```
            mov      ax,es:[si]            ;get D/A output value
            call     DTA
            pop      dx
            pop      ax
            ret
;
;
; ---------------------------------------------
;
;  DTA ---- D/A conversion routine
;
;  Entry condition ------ ax = data
;                         dx = channel no. (0-3)
;
;  Exit condition ----- ax,dx destroyed
;
; ----------------------------------------------
;
DTA:        push     bx
            push     es

            push     ax
            mov      ax,0000h
            mov      es,ax
            pop      ax

            mov      bx,DABASE
            add      bx,dx
            add      bx,dx
            mov      es:[bx],ax

            pop      es
            pop      bx
            ret
;
;
; ------------------------------------------------
;
;  ATDS ----  ATD   routine
;
;   Entry condition ----- si points to data buffer that contains
;                         the channel number
;
;   Exit condition -----  databuffer contains ATD result
;
; ------------------------------------------------------
;
ATDS:
            push     ax
            push     cx
            push     dx
            push     di

            mov      dl,es:[si]         ;get channel number
            inc      si
            mov      cx,es:[si]         ;get number of conversion to take pl
            lea      di,count           ;note it is 2 bytes (word)
            mov      es:[di],cx         ;store away this count

            lea      si,databuf         ;now points the top of the  storage
```

```
continue:     mov      al,dl               ;get channel number
              xor      ah,ah               ;clean higher byte
              call     ATD                 ;get ATD result
              mov      es:[si],ax          ;store
              add      si,2                ;increment to avoid a word just save
              loop     continue

              pop      di
              pop      dx
              pop      cx
              pop      ax
              ret
;
;
; ----------------------------------------------------------------
;
;   ATD   subroutine
;
;         Entry condition ------ ax contains channel number
;
;         Exit condition ------- ax contains A/D conversion result
;
; ----------------------------------------------------------------
;
ATD:
              push     si
              push     es
              push     ax                  ;save ax temporalilly
              xor      ax,ax               ;set es = 0000h
              mov      es,ax
              pop      ax                  ;get channel no. back
              mov      si,MUXCV            ;send channel no.
              cli
              mov      es:[si],ax
repeat:       mov      si,ADCHI            ;check busy bit
              mov      al,es:[si]
              rol      al,1                ;Busy ?
              jc       repeat
              mov      si,ADCLO            ;get a Word
              mov      ax,es:[si]
              sti

              pop      es
              pop      si
              ret

;
;
codeseg  ends
;
;
    END
```

APPENDIX D

FILE DOWNLOADING PROGRAM

(INTEL HEX FORMAT)

```
#include        "stdio.h"

#define         NZ          16384          /* 15 th bits */
#define         Baud        224            /* 9600 */
#define         Sb          4              /* 2 stop bits */
#define         Wd          3              /* b bits word length */
#define         Comst       768            /* communication status check */
#define         Reciv       512            /* receive character  */
#define         Transm      256            /* transmit character  */
#define         Kb_scan     256            /* non-destructive keyboard scan */
#define         Kb_get      0              /* read keyboard buffer  */
#define         DR          256            /* rs232  dsr  */
#define         Flush       1024           /* ah = 4 */
#define         XONF        768            /* xon/xoff    */
#define         CW          23             /* Cntl  W     */
#define         CNTR        18             /* Cntl  R     */
#define         CC          03             /* Cntl  C     */
#define         TR          24576          /* Transmitter register empty */
#define         DATASIZE    15             /* ARRAY SIZE */
#define         FILESIZE    10             /* file name size */
#define         ARBI        15             /* temporaly character storage */
#define         CR          13             /* Carriage return */
#define         LF          10             /* Line feed */
#define         H_wipe      255            /* Wipe out higher byte */
#define         ESC         27             /* Escape code */
#define         SUB         26             /* Subtract */

main()
{

    char  k,file_name[FILESIZE], mem_loc[DATASIZE], temp[ARBI];
    int a,b,c,d,i,Skip;

    Skip = 0;
    a = Baud;               /*  al = 1110011l, 111 = 9600 baud  */
    a = a + Sb;              /*  1 (2^2) = 2 stop bits    */
    a = a + Wd;              /*  11 = 8 bits word length  */
    d = XONF;                /*  dx = 000001100000000,  Xon/Xoff terminal & host *

    a = serial(a,d);        /* reset the serial communication */

    a = Flush;              /* Flush the serial communication */
    a = serial(a,d);


    while(1)
    {
       recicom(&a);
       putchar(a);

       a = Kb_scan;                 /* ah =1 */
       KBscan(&a,&b,&d);             /* non-destructive keyboard scan */

       if (d == 0)                  /* found typed character */
       {
          if (a == CNTR)              /* download Intel hex coded prog.  */
          {
              printf("\nEnter filename to download:");
              scanf("%s",file_name);
```

```
            mem_loc[0] = 'R';
            mem_loc[1] = ' ';
            printf("\nEnter address to send to <seg:offset>:");
            scanf("%s",temp);
            for(i=0; temp[i] != '\0'; ++i)    /* IMPORTANT */
                                              /* scanf() places '\0'=0 at the
                                                 end of the string */
            {
               mem_loc[i+2] = temp[i];
            }

            mem_loc[i+2] = 13;
            mem_loc[i+3] = '\0';
            printf("down loading information is   %s\n",mem_loc);

            sends(mem_loc);


            printf("\n\n FILE DOWN LOADING STARTS \n\n");

            sendf(file_name);

            printf("\n File down-loading complete\n");

            a = SUB;      /* SUB & ESC sequence terminate downloading */
            sendc(a,0);
            a = ESC;
            sendc(a,0);

            Skip = 1;      /* Skip the return code check */

         }
         if (a == CW)
         {
            printf("Enter filename to upload:");
            scanf("%s",file_name);
         }
         if (a == CC)  break;

         if (Skip == 1)
         {
            Skip = 0;
         }
         else if (Skip == 0)
         {
            sendc(a,1);
         }

      }
   }
}

sends(array)
char  array[DATASIZE];
{
   int     a,b,i;

   for(i=0; array[i] != '\0'; ++i)
```

```
   {
      b = array[i];
                       /*  printf("%d data of array to be sent is %d",i,b);*/
      sendc(b,1);
   }
}

recicom(x)
int *x;
{
   int  a,b;

   a = serial(Comst,b);
   if((a & DR) == DR)     /* yes ready */
   {
      a = serial(Reciv,b);
      *x = a;
   }
}

sendc(a,echo)
int a,echo;
{
   int b;
   while(1)
   {

                         /*  printf("\n character to be sent is  %d",a);*/

      b = serial(Comst,b);

                          /*  printf("\n sendc status is %d",b); */

      if ((b & DR) == DR)
      {
         printf("\n found data while tring to transmit  %d  ",a);
         recicom(&b);
         putchar(b);
      }

      if ((b & TR) != TR) printf("\n Transmitter crowded\n");

      if ((b & TR) == TR)
      {
         b = Transm + a;
         b = serial(b,b);

         if (echo == 1)
         {
            recicom(&b);
/*            printf("\n OUTLOOP CHECK FOR RETURN  %d ",(b & H_wipe));*/
            while (b != a)
            {
               if ((b == CR) || (b == LF) || (b == '<'))
               {
                  putchar(b);
                  break;
               }
               recicom(&b);
/*               printf("\n INLOOP CHECK FOR RETURN  %d ",(b & H_wipe));*/
```

```
            }                   /* while(b != a) */
          }                     /* if (echo == 1) */
          break;                /* get out 'while(1)' loop */
        }          /* if ((b & TR) ==TR) */
    }         /* while(1) */

}      /* sendc() */

sendf(name)
char  name[FILESIZE];
{

   int  a,b;
   char c;
   FILE *fp, *fopen();

   fp = fopen(name, "r");

   while((c = getc(fp)) != EOF)
   {
/*      printf("\n down loading information  %d  %c  ",c,c);*/
      sendc((int)c,0);
   }
}
```

```
;
;
;      This program provide basic bios call technic
;
codeseg  segment
         assume        cs:codeseg
         public        serial_,KBscan_

serial_  proc          near
         push          bp
         mov           bp,sp
         mov           ax,[bp+4]        ;get first argument
         mov           dx,[bp+6]        ;get second
         int           14h              ;call bios interrupt for RS232

         mov           sp,bp
         pop           bp
         ret
serial_  endp

KBscan_  proc          near             ;non-destructive keyboard scan
         push          bp
         mov           bp,sp
         mov           bx,[bp+4]
         mov           ax,[bx]
         int           16h              ;call non-destructive keyborad
         mov           cx,ax            ;make a copy
         lahf                           ;scan
         and           ax,04000h        ;mask off except 15th bit
         mov           bx,[bp+8]        ;get third argument address
         mov           [bx],ax          ;transfer ax anded result
         cmp           ax,04000h        ;if no character
         jz            back             ;return

         mov           ax,cx            ;copy back
         xor           ch,ch            ;clean up higher byte
         mov           bx,[bp+4]        ;first argument address
         mov           [bx],cx          ;ASCII values of a character
         xor           al,al            ;clean up lower bytes
         mov           bx,[bp+6]        ;second argument address
         mov           [bx],ax          ;keyboard scan code

         mov           ah,03h           ;flush keyborad buffer
         int           16h
back:    mov           sp,bp
         pop           bp
         ret
KBscan_  endp


codeseg  ends
         end
```

APPENDIX E

MASTER COMPUTER PROGRAM

```
100      CLEAR   ,61184!          ' BASIC Declarations
110      IBINIT1 = 61184!
120      IBINIT2 = IBINIT1 + 3    ' Lines 1 through 6 MUST be included in your p
130      BLOAD "bib.m",IBINIT1
140      CALL IBINIT1(IBFIND,IBTRG,IBCLR,IBPCT,IBSIC,IBLOC,IBPPC,IBBNA,IBONL,IB
150      CALL IBINIT2(IBGTS,IBCAC,IBWAIT,IBPOKE,IBWRT,IBWRTA,IBCMD,IBCMDA,IBRD,
160      REM Optionally include the following declarations in your program.
170      REM They provide appropriate mnemonics by which
180      REM to reference commonly used values.  Some mnemonics (GET%, ERR%,
190     REM END%, ATN%) are preceded by "B" in order to distinguish them from
200     REM BASIC keywords.
210     REM
220     REM GPIB Commands
230     UNL% = &H3F      ' GPIB unlisten command
240     UNT% = &H5F      ' GPIB untalk command
250     GTL% = &H1       ' GPIB go to local
260     SDC% = &H4       ' GPIB selected device clear
270     PPC% = &H5       ' GPIB parallel poll configure
280     BGET% = &H8      ' GPIB group execute trigger
290     TCT% = &H9       ' GPIB take control
300     LLO% = &H11      ' GPIB local lock out
310     DCL% = &H14      ' GPIB device clear
320     PPU% = &H15      ' GPIB ppoll unconfigure
330     SPE% = &H18      ' GPIB serial poll enable
340     SPD% = &H19      ' GPIB serial poll disable
350     PPE% = &H60      ' GPIB parallel poll enable
360     PPD% = &H70      ' GPIB parallel poll disable
370     REM
380     REM GPIB status bit vector
390     REM global variable IBSTA% and wait mask
400     BERR% = &H8000  ' Error detected
410     TIMO% = &H4000  ' Timeout
420     BEND% = &H2000  ' EOI or EOS detected
430     SRQI% = &H1000  ' SRQ detected by CIC
440     RQS% = &H800     ' Device needs service
450     CMPL% = &H100    ' I/O completed
460     LOK% = &H80      ' Local lockout state
470     REM% = &H40      ' Remote state
480     CIC% = &H20      ' Controller-In-Charge
490     BATN% = &H10     ' Attention asserted
500     TACS% = &H8      ' Talker active
510     LACS% = &H4      ' Listener active
520     DTAS% = &H2      ' Device trigger state
530     DCAS% = &H1      ' Device clear state
540     REM
550     REM Error messages returned in global variable IBERR%
560     EDVR% =  0       ' DOS error
570     ECIC% =  1       ' Function requires board to be CIC
580     ENOL% =  2       ' Write function detected no Listeners
590     EADR% =  3       ' Interface board not addressed correctly
600     EARG% =  4       ' Invalid argument to function call
610     ESAC% =  5       ' Function requires board to be SAC
620     EABO% =  6       ' I/O operation aborted
630     ENEB% =  7       ' Non-existent interface board
640     EOIP% = 10       ' I/O operation started before previous operation compl
650     ECAP% = 11       ' No capability for operation
660     EFSO% = 12       ' File system operation error
670     EBUS% = 14       ' Command error during device call
680     ESTB% = 15       ' Serial poll status byte lost
690     ESRQ% = 16       ' SRQ remains asserted
```

```
700      REM
710      REM EOS mode bits
720      BIN% = &H1000    ' Eight bit compare
730      XEOS% = &H800    ' Send EOI with EOS byte
740      REOS% = &H400    ' Terminate read on EOS
750      REM
760      REM Timeout values and meanings
770      TNONE% =   0     ' Infinite timeout (disabled)
780      T10US% =   1     ' Timeout of 10 us (ideal)
790      T30US% =   2     ' Timeout of 30 us (ideal)
800      T100US% =  3     ' Timeout of 100 us (ideal)
810      T300US% =  4     ' Timeout of 300 us (ideal)
820      T1MS% =    5     ' Timeout of 1 ms (ideal)
830      T3MS% =    6     ' Timeout of 3 ms (ideal)
840      T10MS% =   7     ' Timeout of 10 ms (ideal)
850      T30MS% =   8     ' Timeout of 30 ms (ideal)
860      T100MS% =  9     ' Timeout of 100 ms (ideal)
870      T300MS% = 10     ' Timeout of 300 ms (ideal)
880      T1S% =    11     ' Timeout of 1 s (ideal)
890      T3S% =    12     ' Timeout of 3 s (ideal)
900      T10S% =   13     ' Timeout of 10 s (ideal)
910      T30S% =   14     ' Timeout of 30 s (ideal)
920      T100S% =  15     ' Timeout of 100 s (ideal)
930      T300S% =  16     ' Timeout of 300 s (ideal)
940      T1000S% = 17     ' Timeout of 1000 s (maximum)
950      REM
960      REM Miscellaneous
970      S% = &H8         ' Parallel Poll sense bit
980      LF% = &HA        ' Line feed character
990      REM
1000      REM Application program variables passed to
1010      REM GPIB functions
1020      REM
1030      CMD$ = SPACE$(10)        ' command buffer
1040      RD$ = SPACE$(255)        ' read data buffer
1050      WRT$ = SPACE$(255)       ' write data buffer
1060      BNAME$ = SPACE$(7)       ' board name buffer
1070      BDNAME$ = SPACE$(7)      ' board or device name buffer
1080      FLNAME$ = SPACE$(50)     ' file name buffer
1090 '
1100 '
1110 ' ******************************
1120 '
1130    REM ****************************************************************
1140    REM  CLOCK.BAS   REV. C.1
1150    REM  NATIONAL INSTRUMENTS SOFTWARE FOR GPIB-PC2000
1160    REM  TIME OF DAY CLOCK (NATIONAL SEMICONDUCTOR MM58167A)
1170    REM  (c) Copyright 1985, National Instruments
1180    REM  All rights reserved
1190    REM ****************************************************************
1200    REM
1210    REM      GOSUB 10200 ... GETALARM
1220    REM      GOSUB 10300 ... SETALARM
1230    REM      GOSUB 10400 ... GETCLOCK
1240    REM      GOSUB 10500 ... SETCLOCK
1250    REM      GOSUB 10600 ... SETFREQ
1260    REM      GOSUB 10800 ... OUTCLK
1270    REM
1280    REM ****************************************************************
1290    REM  SUBROUTINE:  INITIALIZE PROGRAM VARIABLES
```

```
1300   REM *************************************************************
1310   REM
1320   REM         REGISTER ADDRESSES
1330   CL.TM.MSEC% = 0
1340   CL.TM.SEC% = 2
1350   CL.TM.MIN% = 3
1360   CL.TM.HR% = 4
1370   CL.TM.DAY% = 5
1380   CL.TM.DATE% = 6
1390   CL.TM.MO% = 7
1400   CL.ALRM.MSEC% = 8
1410   CL.ALRM.CDSEC% = 9
1420   CL.ALRM.SEC% = 10
1430   CL.ALRM.MIN% = 11
1440   CL.ALRM.HR% = 12
1450   CL.ALRM.DAY% = 13
1460   CL.ALRM.DATE% = 14
1470   CL.ALRM.MO% = 15
1480   CL.ISR% = 16
1490   CL.IMR% = 17
1500   CL.CRST% = 18
1510   CL.ARST% = 19
1520   CL.OVFL% = 20
1530   CL.GO% = 21
1540   CL.IE% = 22
1550   DIM TM%(8)       ' ARRAY FOR PASSING TIME PARAMETERS TO/FROM SUBROUTINES
1560   MSEC% = 0        ' TEN THOUSANDTHS OF SECONDS
1570   CDSEC% = 1       ' HUNDREDTHS AND TENTHS OF SECONDS
1580   SEC% = 2         ' SECONDS
1590   MIN% = 3         ' MINUTES
1600   HR% = 4          ' HOURS
1610   DAY% = 5         ' DAY OF THE WEEK
1620   DATE% = 6        ' DAY OF THE MONTH
1630   MO% = 7          ' MONTH
1640   YR% = 8          ' YEAR
1650   REM         REGISTER ADDRESSES
1660   ISR% = 0         ' INTERRUPT STATUS REGISTER
1670   IMR% = 1         ' INTERRUPT MASK (CONTROL) REGISTER
1680   CRST% = 2        ' COUNTER RESET
1690   ARST% = 3        ' ALARM RESET (RAM)
1700   OVFL% = 4        ' OVERFLOW (STATUS) BIT
1710   GO% = 5          ' GO COMMAND - USED FOR PRECISE STARTING OF CLOCK
1720   IE% = 6          ' INTERRUPT ENABLE (STANDBY INTERRUPT)
1730   REM         CLOCK CONTROL REGISTERS
1740   CCR% = &H81E     ' CLOCK CONTROL REGISTER
1750   CDR% = &H81C     ' CLOCK DATA REGISTER
1760   IE% = &H20       ' INTERRUPT ENABLE  (master and separate bits) (5&6)
1770   RESULT% = 0
1780   REM     ***** ASSEMBLY LANGUAGE ROUTINES STORED AS DATA ******
1790   DIM ASM%(66)
1800  DATA &H5590, &Hec8b, &H5e8b, &Hba06, &H0000, &H68e8, &H8800, &Hba07
1810  DATA &H0001, &H60e8, &H8800, &H0247, &H02ba, &He800, &H0057, &H4788
1820  DATA &Hba04, &H0003, &H4ee8, &H8800, &H0647, &H04ba, &He800, &H0045
1830  DATA &H4788, &Hba08, &H0005, &H3ce8, &H8800, &H0a47, &H06ba, &He800
1840  DATA &H0033, &H4788, &Hba0c, &H0007, &H2ae8, &H8800, &H0e47, &H14ba
1850  DATA &He800, &H0021, &Hc084, &Haf75, &He58b, &Hca5d, &H0002, &H5590
1860  DATA &Hec8b, &H5e8b, &H8b08, &He817, &H000b, &H5e8b, &H8806, &H8b07
1870  DATA &H5de5, &H04ca, &H8a00, &Hbac2, &H081e, &Hbaee, &H081c, &Hc3ec
1880  DATA &Hcbfa, &Hcbfb
1890   FOR I = 0 TO 65
```

```
1900   READ ASM%(I)
1910   NEXT I
1920 '
1930     BDNAME$ = "DEV1"
1940     CALL   IBFIND(BDNAME$,DEV1%)
1950    BDNAME$ = "GPIB0"
1960    CALL   IBFIND(BDNAME$,GPIB0%)
1970    '
1980    CALL    IBCLR(GPIB0%)
1990 '
2000     IF DEV1% < 0 THEN PRINT"ERROR WITH DEV1  ":STOP
2010    GOTO 5760
2020 '
2030 '
2040 ' ********************************
2050 '
2060 '
2070   REM
2080   REM  ****************************************************************
2090   REM  SUBROUTINE:  GETALARM
2100   REM     RETURN THE ALARM SETTING IN TM.
2110   REM  ****************************************************************
2120   LOCKOUT = VARPTR(ASM%(64))
2130   CALL LOCKOUT                 ' LOCK OUT INTERRUPTS
2140   A% = CL.ALRM.SEC%
2150   INCLK = VARPTR(ASM%(47)) : CALL INCLK(A%,TM%(SEC%))
2160   A% = CL.ALRM.MIN%
2170   INCLK = VARPTR(ASM%(47)) : CALL INCLK(A%,TM%(MIN%))
2180   A% = CL.ALRM.HR%
2190   INCLK = VARPTR(ASM%(47)) : CALL INCLK(A%,TM%(HR%))
2200   A% = CL.ALRM.DAY%
2210   INCLK = VARPTR(ASM%(47)) : CALL INCLK(A%,TM%(DAY%))
2220   A% = CL.ALRM.DATE%
2230   INCLK = VARPTR(ASM%(47)) : CALL INCLK(A%,TM%(DATE%))
2240   A% = CL.ALRM.MO%
2250   INCLK = VARPTR(ASM%(47)) : CALL INCLK(A%,TM%(MO%))
2260   TM%(MSEC%) = 0
2270   TM%(CDSEC%) = 0
2280   TM%(YR%) = 0
2290   UNLOCK = VARPTR(ASM%(65))
2300   CALL UNLOCK                ' RESTORE INTERRUPTS
2310   RETURN
2320   REM
2330   REM  ****************************************************************
2340   REM  SUBROUTINE:  SETALARM
2350   REM     SET THE ALARM TIME AS INDICATED BY TM.
2360   REM             NOTE: THE USER MUST SET UP INTERRUPT 23 BEFORE
2370   REM                   CALLING THIS FUNCTION
2380   REM  ****************************************************************
2390   LOCKOUT = VARPTR(ASM%(64))
2400   CALL LOCKOUT                 ' LOCK OUT INTERRUPTS
2410   IF (TM% (YR%) <> 0) THEN GOTO 2470
2420   IESTATE% = 0
2430   A% = CL.IMR%
2440   D% = 0
2450   GOSUB 3690               ' OUTCLK
2460   GOTO 2690
2470   A% = CL.ALRM.SEC%
2480   D% = TM% (SEC%)
2490   GOSUB 3690               ' OUTCLK
```

```
2500    A% = CL.ALRM.MIN%
2510    D% = TM% (MIN%)
2520    GOSUB 3690              ' OUTCLK
2530    A% = CL.ALRM.HR%
2540    D% = TM% (HR%)
2550    GOSUB 3690              ' OUTCLK
2560    A% = CL.ALRM.DAY%
2570    D% = &HCC
2580    GOSUB 3690              ' OUTCLK
2590    A% = CL.ALRM.DATE%
2600    D% = TM% (DATE%)
2610    GOSUB 3690              ' OUTCLK
2620    A% = CL.ALRM.MO%
2630    D% = TM% (MO%)
2640    GOSUB 3690              ' OUTCLK
2650    IESTATE% = IE%
2660    A% = CL.IMR%
2670    D% = 1
2680    GOSUB 3690              ' OUTCLK
2690    UNLOCK = VARPTR(ASM%(65))
2700    CALL UNLOCK               ' RESTORE INTERRUPTS
2710    RETURN
2720    REM
2730    REM  **************************************************************
2740    REM  SUBROUTINE:  GETCLOCK
2750    REM     PUT THE CURRENT DAY, DATE AND TIME IN TM.
2760    REM  **************************************************************
2770    LOCKOUT = VARPTR(ASM%(64))
2780    CALL LOCKOUT                 ' LOCK OUT INTERRUPTS
2790    INCLOCK = VARPTR(ASM%(0))
2800    CALL INCLOCK(TM%(0))
2810    A% = CL.ALRM.CDSEC% OR IESTATE%
2820    INCLK = VARPTR(ASM%(47))
2830    CALL INCLK(A%, RESULT%)
2840    RESULT% = RESULT% AND &H33
2850 '  PRINT "result & h33", HEX$(RESULT%)
2860    TM%(YR%) = ((RESULT% \ 4) OR RESULT%) AND &HF
2870    A% = CL.ALRM.MSEC
2880    INCLK = VARPTR(ASM%(47))
2890    CALL INCLK(A%, RESULT%)
2900    IF ((TM%(MO%) > 6) OR NOT (RESULT% AND &H20)) THEN GOTO 2950
2910    TM%(YR%) = TM%(YR%) + 1
2920    D% = ((TM%(YR%) * 4) OR TM%(YR%)) OR &HCC
2930    A% = CL.ALRM.CDSEC%
2940    GOSUB 3690      ' OUTCLK
2950    A% = CL.ALRM.MSEC%
2960    IF TM%(MO%) <= 6 THEN GOTO 2990
2970    D% = &HE0
2980    GOTO 3000
2990    D% = &HC0
3000    GOSUB 3690      ' OUTCLK
3010    UNLOCK = VARPTR(ASM%(65))
3020    CALL UNLOCK               ' RESTORE INTERRUPTS
3030    RETURN
3040    REM
3050    REM  **************************************************************
3060    REM  SUBROUTINE:  SETCLOCK
3070    REM     SET THE CLOCK TO THE TIME INDICATED BY TM.
3080    REM  **************************************************************
3090    LOCKOUT = VARPTR(ASM%(64))
```

```
3100   CALL LOCKOUT                ' LOCK OUT INTERRUPTS
3110   A% = CL.CRST%
3120   D% = &HFF
3130   GOSUB 3690      ' OUTCLK
3140   A% = CL.TM.MSEC%
3150   D% = 0
3160   GOSUB 3690      ' OUTCLK
3170   A% = CL.TM.CDSEC%
3180   D% = 0
3190   GOSUB 3690      ' OUTCLK
3200   A% = CL.TM.SEC%
3210   D% = TM% (SEC%)
3220   GOSUB 3690      ' OUTCLK
3230   A% = CL.TM.MIN%
3240   D% = TM% (MIN%)
3250   GOSUB 3690      ' OUTCLK
3260   A% = CL.TM.HR%
3270   D% = TM% (HR%)
3280   GOSUB 3690      ' OUTCLK
3290   A% = CL.TM.DAY%
3300   D% = TM% (DAY%)
3310   GOSUB 3690      ' OUTCLK
3320   A% = CL.TM.DATE%
3330   D% = TM% (DATE%)
3340   GOSUB 3690      ' OUTCLK
3350   A% = CL.TM.MO%
3360   D% = TM% (MO%)
3370   GOSUB 3690      ' OUTCLK
3380   A% = CL.ALRM.CDSEC%
3390   D% = ((TM%(YR%) * 4) OR TM%(YR%)) OR &HCC
3400   GOSUB 3690      ' OUTCLK
3410   A% = CL.ALRM.MSEC%
3420   IF (TM% (MO%) <= 6) THEN GOTO 3450
3430   D% = &HE0
3440   GOTO 3460
3450   D% = &HC0
3460   GOSUB 3690      ' OUTCLK
3470   UNLOCK = VARPTR(ASM%(65))
3480   CALL UNLOCK                 ' RESTORE INTERRUPTS
3490   RETURN
3500   REM
3510   REM  ***********************************************************
3520   REM  SUBROUTINE:  SETFREQ
3530   REM      SET THE PERIODIC INTERRUPT FREQUENCY
3540   REM      ACCORDING TO THE BITS IN N.
3550   REM  ***********************************************************
3560   LOCKOUT = VARPTR(ASM%(64))
3570   CALL LOCKOUT                ' LOCK OUT INTERRUPTS
3580   A% = CL.IMR%
3590   IF N% = 0 THEN GOTO 3620         ' SET/CLEAR IE ON NEXT IN/OUT CALL
3600   IESTATE% = 0
3610   GOTO 3630
3620   IESTATE% = IE%
3630   D% = N%
3640   GOSUB 3690      ' OUTCLK
3650   UNLOCK = VARPTR(ASM%(65))
3660   CALL UNLOCK                 ' RESTORE INTERRUPTS
3670   RETURN
3680   REM
3690   REM  ===========================================================
```

```
3700   REM  SUBROUTINE:  OUTCLK
3710   REM  WRITE DATA IN D% TO CLOCK REGISTER A%.
3720   REM  ============================================================
3730   REM
3740   OUT CCR%, A% OR IESTATE%
3750   OUT CDR%, D%
3760   RETURN
3770  ' *****************
3780 '
3790 ' **************************
3800 '
3810 '   INTEGER READ ROUTINE
3820 '
3830 '   WRT$ CONTAINS THE NECESSARY COMMAND & DATA
3840 '   MASK% CONTAINS PROCESS CHECK
3850    CALL IBWRT(DEV1%,WRT$)
3860    CALL IBWAIT(DEV1%,MASK%)
3870    RETURN
3880 '
3890 ' ****************************************
3900 '
3910 ' ************************************
3920 '
3930 '   INTEGER READ ROUTINE
3940 '
3950 '   RD% IS AN ARRAY
3960 '   CNT%  =  # OF BYTES TO BE READ
3970 '
3980    CALL IBRDI(DEV1%,RD%(M%),CNT%)
3990    CALL IBWAIT(DEV1%,MASK%)
4000    RETURN
4010 '
4020 ' ***********************************
4030 '
4040 ' *** STRICT 2 BYTES ONLY ***********
4050 '
4060    CALL IBRDI(DEV1%,RD%(0),CN%)
4070    CALL IBWAIT(DEV1%,MASK%)
4080    RETURN
4090 '
4100 ' ***********************************
4110 '
4120 ' ************************************
4130 '
4140 '   ASCII READ ROUTINE
4150 '
4160 '   RD$ IS AN 255 BYTES CHARACTER STORAGE
4170 '
4180    CALL IBRD(DEV1%,RD$)
4190    CALL IBWAIT(DEV1%,MASK%)
4200    RETURN
4210 '
4220 ' ****************************************
4230 '
4240 ' ************** CONTROL OUTPUT ****
4250 '
4260     HIGH% = INT(SERVO%/FULL%)
4270     LOW% = SERVO% - HIGH%*FULL%
4280     WRT$=CHR$(1)+CHR$(1)+CHR$(LOW%)+CHR$(HIGH%)
4290     GOSUB 3850:RETURN
```

```
4300 '
4310 ' *********************************
4320 '
4330 ' *****************   COMPLETE STOP *******
4340 '
4350    SERVO% = MIDDLE%
4360    GOSUB 4250:PRINT" SERVO STOP "
4370 '
4380 '
4390    PASS% = 0
4400 '
4410 '
4420    WRT$ = ATD2$
4430    GOSUB 3850:GOSUB 4050
4440    DOLD% = RD%(0)
4450    FOR L% = 1 TO 10
4460            WRT$ = ATD2$
4470            GOSUB 3850:GOSUB 4050
4480            PRINT" CURRENT P";RD%(0)
4490            DNEW% = RD%(0)
4500            DIF%(L%) = ABS(DOLD%-DNEW%)
4510            IF DIF%(L%) >= 4 THEN PASS% = 1
4520            DIF%(L%)=0
4530            DOLD% = DNEW%
4540    NEXT L%
4550 '
4560    IF PASS% > 0 THEN PRINT" NOT PASS ":GOTO 4390
4570 '
4580    PRINT"COMPLETE STOP !!! "
4590    RETURN
4600 '
4610 ' *********************************
4620 '
4630 ' ****** POSITION CONTROL *****
4640 '
4650    PRINT"ENTER POSITION CONTROL"
4660    GOSUB 4340     :' COMPLETE STOP
4670 '
4680    WRT$ = ATD11$:M% = UP%        :' FIRST 3 BYTES CAN NOT BE USED
4690    GOSUB 3850:GOSUB 3970
4700 '  PRINT" TAKE A LOOK ";RD%(M%),RD%(M%+1),RD%(M%+2)
4710 '  STOP
4720    M% = M% + UP%
4730 '
4740 '
4750    SERVO%= NULL% + INC% * DIR%       :'INITIAL INPUT
4760    GOSUB 4250
4770 '
4780    WRT$ = ATD10$                      :'GET NEW POSITION
4790    GOSUB 3850:GOSUB 3970
4800    DNEW% = RD%(M%)
4810 '  PRINT"TARGET ";SETL%,"CURRENT ";RD%(M%),"SERVO ";SERVO%
4820 '
4830    M% = M% + UP%
4840 '
4850 '
4860    IF ABS(DNEW% - DOLD%) <= 50  THEN SERVO% = SERVO% + DIR%:GOSUB 4250:GOT
4870 '  PRINT"MOVING"
4880 '
4890    WRT$ = ATD10$
```

```
4900      GOSUB 3850:GOSUB 3970
4910      M% = M% + UP%
4920 '    PRINT"SET TARGET ";SETL%,"CURRENT ";RD%(M%),"SERVO ";SERVO%
4930 '
4940      IF ABS(SETL% - RD%(M%-UP%)) >= LIMITP% THEN 4890
4950      M% = M% + UP%
4960  '
4970      GOSUB 4340:'COMPLETE STOP
4980 '
4990 '    FOR I%=0 TO M%-UP%:PRINT I%,RD%(I%),RD%(I%+1),RD%(I%+2):NEXT I%
5000 '    STOP
5010 '
5020 '
5030      RETURN
5040 '
5050 ' *************************
5060 '
5070 ' ************** MIN & MAX POSITION ******
5080 '
5090      PRINT"ENTER MIN & MAX ROUTINE"
5100      SERVO% = NULL% + INC%
5110      GOSUB 4250
5120 '
5130      WRT$ = ATD2$
5140      GOSUB 3850:GOSUB 4050
5150      DNEW% = RD%(0)
5160      IF ABS(DNEW% - DOLD%) <= LIMIT% THEN SERVO% = SERVO% + 1:GOTO 5110
5170      PRINT"MOVING"
5180 '
5190      CK% = 0:CK1% = 0:UD% = 0:NO%=0
5200      WRT$ = ATD2$
5210      GOSUB 3850:GOSUB 4050
5220 '
5230      MAX% = RD%(0)
5240      MINI% = RD%(0)
5250 '    PRINT"MIN ";MINI%,"MAX ";MAX%
5260 '
5270      WRT$ = ATD2$
5280      GOSUB 3850:GOSUB 4050
5290      IF RD%(0) > MAX% THEN MAX% = RD%(0):CK% = 1
5300      IF RD%(0) < MINI% THEN MINI% = RD%(0):CK% = -1
5310      IF CK% <> CK1% THEN CK1% = CK%:UD% = UD% +1
5320      NO% = NO% +1
5330 '    PRINT"MIN ";MINI%,"MAX ";MAX%,"CK ";CK%,"CK1 ";CK%1,"UD ";UD%,"NO ";NO%
5340      IF (UD% >= 3) OR (NO% > 150) THEN RETURN
5350 '
5360      GOTO 5270
5370 '
5380 ' *************************
5390 '
5400 ' ************** STATIONARY TIME *********
5410 '
5420 '    SETTIME% = 10
5430     CLS
5440     HOUR%=0:INCMIN%=0
5450      GOSUB 2730
5460      INITMIN% = VAL(HEX$(TM%(MIN%)))
5470      INITSEC% = VAL(HEX$(TM%(SEC%)))
5480     INITSEC% = INITSEC% + SETTIME%
5490     IF INITSEC% >  60 THEN INCMIN% =INT(INITSEC% / 60)
```

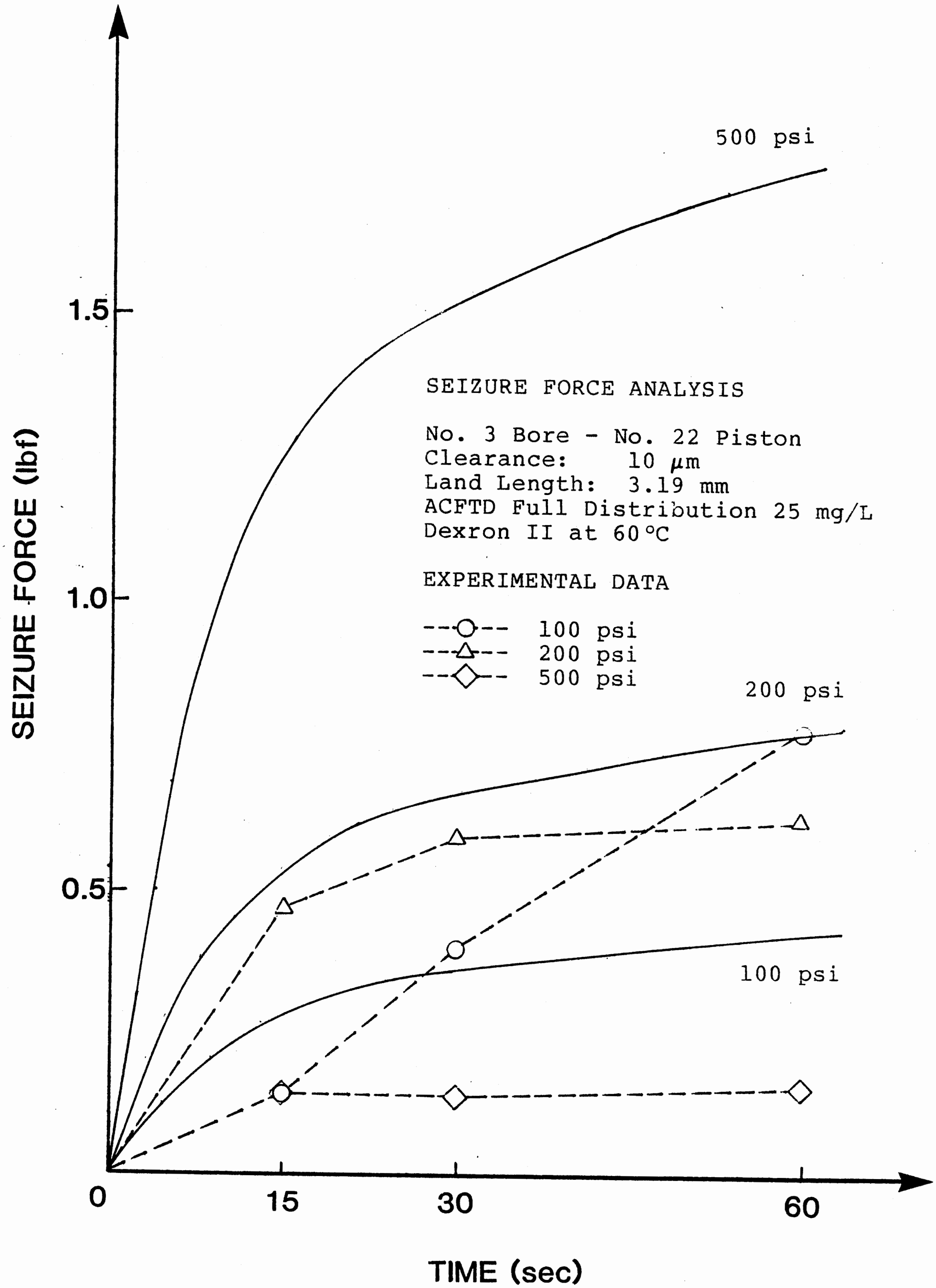
```
5500    INITMIN% = INITMIN% + INCMIN%
5510    INITSEC% = INITSEC% - 60 * INCMIN%
5520    IF INITMIN% > 59 THEN HOUR% =INT(INITMIN% / 60)
5530    INITMIN% = INITMIN% - HOUR% * 60
5540 '
5550    GOSUB 2730
5560 '
5570    INTERMIN% = VAL(HEX$(TM%(MIN%)))
5580    INTERSEC% = VAL(HEX$(TM%(SEC%)))
5590 '
5600    TESTMIN% = INITMIN% - INTERMIN% + HOUR% * 60
5610    TESTSEC% = INITSEC% - INTERSEC%
5620 '
5630 '   IF TESTMIN% < 0 THEN TESTMIN% = TESTMIN% + 60
5640     IF TESTSEC% < 0 THEN TESTSEC% = TESTSEC% + 60:TESTMIN%=TESTMIN%-1
5650 '
5660 '
5670    LOCATE 5,10:PRINT TESTMIN% * 60 + TESTSEC%;"  sec "
5680    IF INITMIN% =INTERMIN% AND INTERSEC%=INITSEC% THEN RETURN
5690 '
5700     GOTO 5550
5710 '
5720 '
5730 ' ********************************
5740 '
5750 ' ***** MAIN ROUTIN  ****************
5760 '
5770     CNT% = 6:CN% = 2
5780     MASK% = &H100
5790     DIM RD%(1000),DIF%(10)
5800     SETL% =  1560:'  MIDDLE POSITION
5810     TOP% = 2000
5820     BOTTOM% = 1200
5830     FULL% = 256
5840     LIMIT% = 10
5850     LIMITP% = 15
5860     NULL% = 2048
5870 '
5880     INTERRUPT% = 0
5890     MIDDLE%=2048
5900     GOUP%=2100
5910     GODOWN%=1950
5920     INC% = 50
5930     UP% = 3
5940 '
5950     ATD0$ = CHR$(0)+CHR$(0)+CHR$(1)+CHR$(0)
5960     ATD2$ = CHR$(0)+CHR$(2)+CHR$(1)+CHR$(0)
5970 '   DTA$  = CHR$(1)+CHR$(1)+CHR$(LOW%)+CHR$(HIGH%)
5980     FC$ = CHR$(16)
5990     IOFF$ = CHR$(18)+CHR$(31):'   31-OFF, 30-ON
6000     ION$ = CHR$(18)+CHR$(30)
6010     ATD10$ = CHR$(10)
6020     ATD11$ = CHR$(11)
6030 '
6040    ATD12$=CHR$(12)+CHR$(144)+CHR$(1)
6050    ADBASE$=CHR$(13)+CHR$(100)+CHR$(0)
6060    ATD13$(0)=ADBASE$+CHR$(0)+CHR$(0)
6070    ATD13$(1)=ADBASE$+CHR$(100)+CHR$(0)
6080    ATD13$(2)=ADBASE$+CHR$(200)+CHR$(0)
6090    ATD13$(3)=ADBASE$+CHR$(44)+CHR$(1)
```

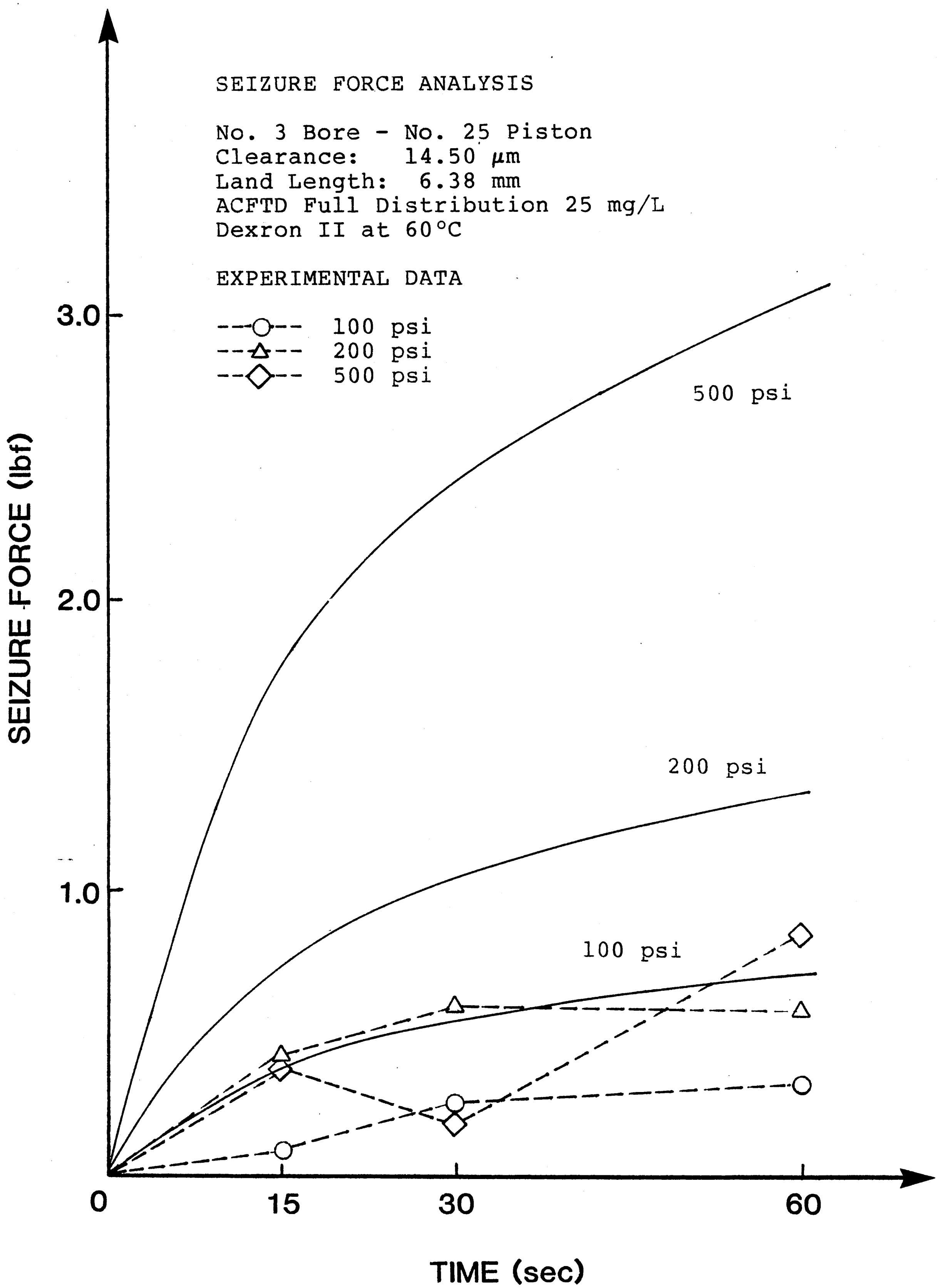
```
6100 '
6110 '
6120      WRT$ = IOFF$
6130      GOSUB 3850:PRINT "CLOSE L-VALVE"
6140 '
6150      GOSUB 4340                          :'COMPLETE STOP
6160 '
6170      INPUT"WHAT PRESSURE (PSI)";PRESSURE
6180      INPUT"STATIONARITY TIME (SEC)";SETTIME%
6190 '
6200      GOSUB 5080                          :' MIN MAX POSITION SEARCH
6210 '
6220      PRINT"FINAL MAX IS ";MAX%,"FINAL MIN IS ";MINI%,"NO ";NO%
6230 '
6240      IF MAX%-MIN% <300 THEN 6190:' REPEAT
6250 '
6260 '
6270      GOSUB 4340                            :' COMPLETE STOP
6280 '
6290      WRT$ = ATD2$
6300      GOSUB 3850:GOSUB 4050
6310      IF ABS(MAX% - RD%(0)) <= LIMIT% THEN PRINT"ALREADY AT THE TOP":GOTO 637
6320 '
6330      PRINT"NOW POSITION CONTROL FOR TOP POSITION "
6340      SETL% = MAX%:DIR% = 1
6350      GOSUB 4640:PRINT"SPOOL AT THE TOP"
6360 '
6370      SETL% = MINI%:DIR% = -1             :' SET SPOOL AT THE BOTTOM
6380      GOSUB 4640:PRINT"SPOOL AT THE BOTTOM"
6390      '
6400      WRT$ = ION$
6410      GOSUB 3850:PRINT"LEAKAGE-VALVE OPEN"
6420 '
6430      GOSUB 5410                      :' TIMER
6440 '
6450      WRT$ = IOFF$
6460      GOSUB 3850:PRINT"LEAKAGE  VALVE CLOSED"
6470 '
6480      WRT$=ATD12$   :'TAP INTERRUPT
6490      GOSUB 3850
6500 '
6510 '
6520      SETL% = MAX%:DIR%=1
6530      GOSUB 4640
6540      PRINT"SPOOL AT THE TOP"
6550 '
6560 '
6570      CNT%=400
6580      FOR J=0 TO 3
6590              WRT$=ATD13$(J)
6600              GOSUB 3850
6610              M%=J*200
6620              GOSUB 3970
6630      NEXT J
6640 '
6650      CNT%=6
6660 '
6670      '
6680      GOSUB 2730
6690      FILENAME$="B:"+"A"+HEX$(TM%(DATE%))+HEX$(TM%(HR%))+HEX$(TM%(MIN%))+".S"
```

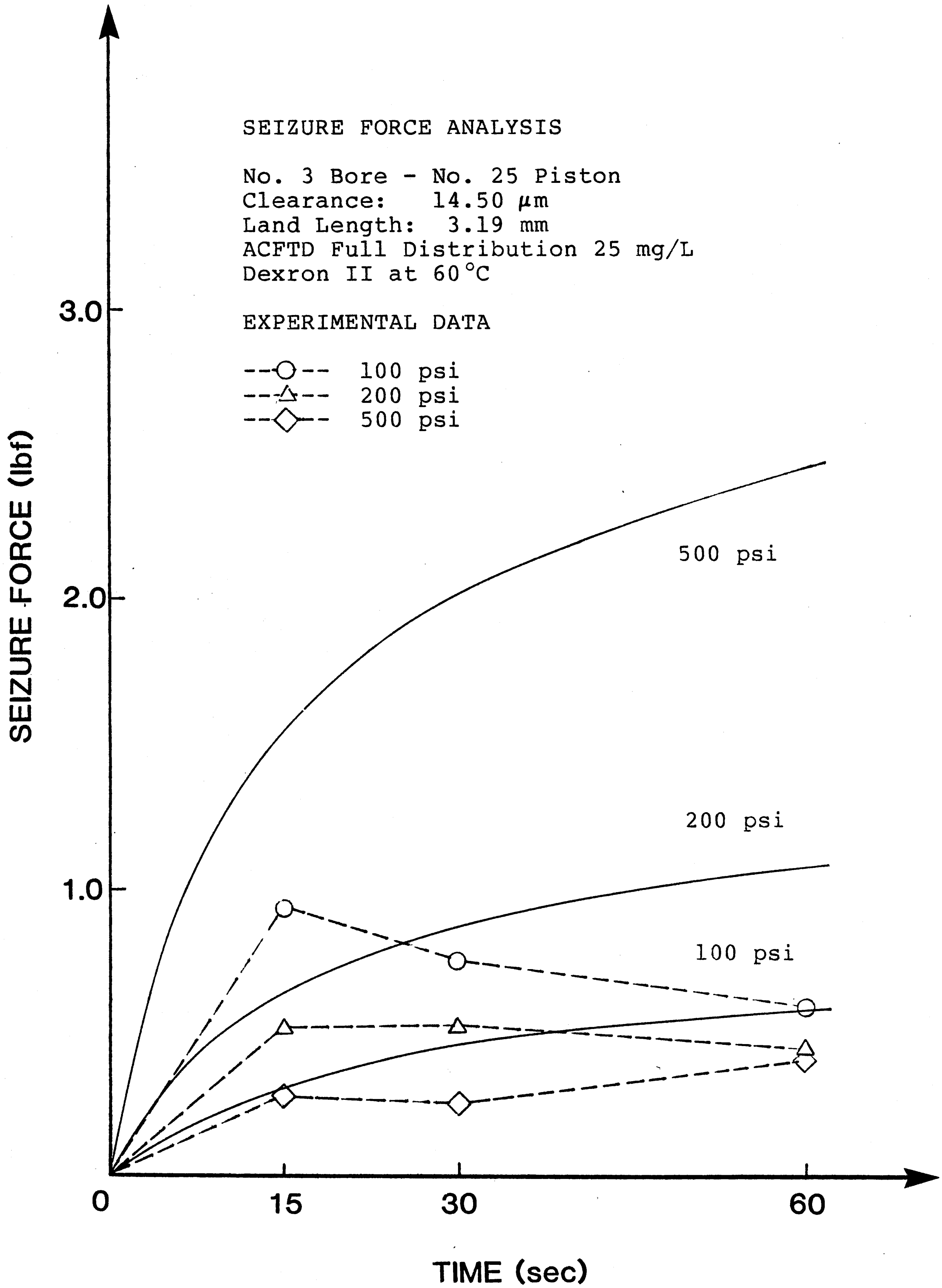
```
6700     PRINT"FILE IS  ";FILENAME$
6710 '
6720     OPEN "R",1,FILENAME$,4
6730     FIELD 1,2 AS P$,2 AS F$
6740 '
6750     FOR I% = 0 TO 799 STEP 2
6760            LSET P$=MKI$(RD%(I%))
6770            LSET F$=MKI$(RD%(I%+1))
6780            PUT 1,INT((I%+1)/2)+1
6790     NEXT I%
6800 '
6810     CLOSE 1
6820 '
6830     PRINT"DATA STORED UNDER FILENAME  ";FILENAME$
6840     LPRINT"FILENAME  ";FILENAME$,"PRESSURE ";PRESSURE;" PSI","TIME ";SETTIM
6850     INPUT"WOULD YOU LIKE TO SEE THE GRAPH OF THIS RESULT ";A$
6860     IF A$ = "Y" OR A$ = "y" THEN RUN"B:GRAPH.BAS"
6870     IF A$ ="N" OR A$ = "n" THEN 6170
6880     GOTO 6850
6890 '
6900 '  GOTO 5960
6910 '
6920 '
```

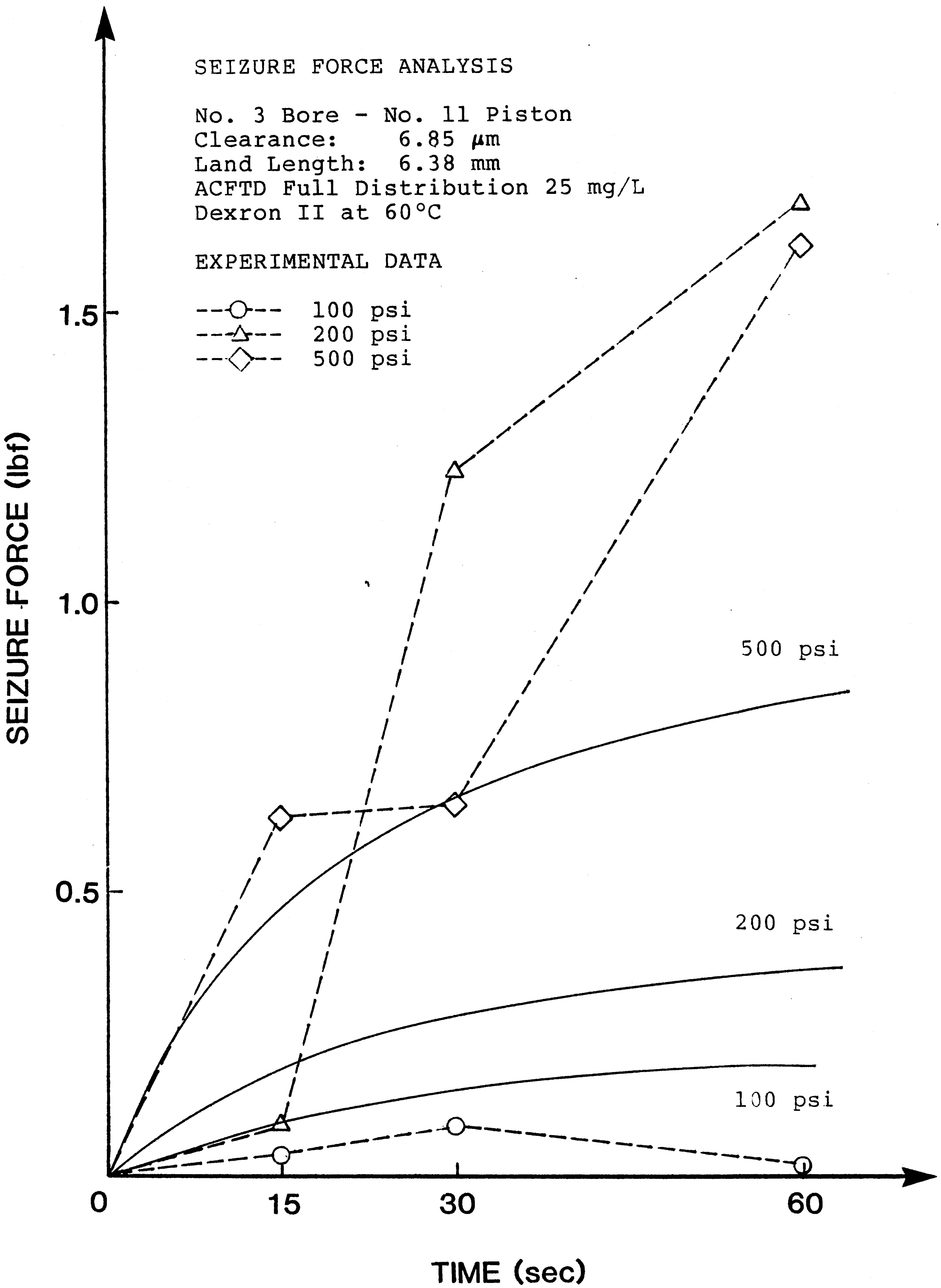
APPENDIX F

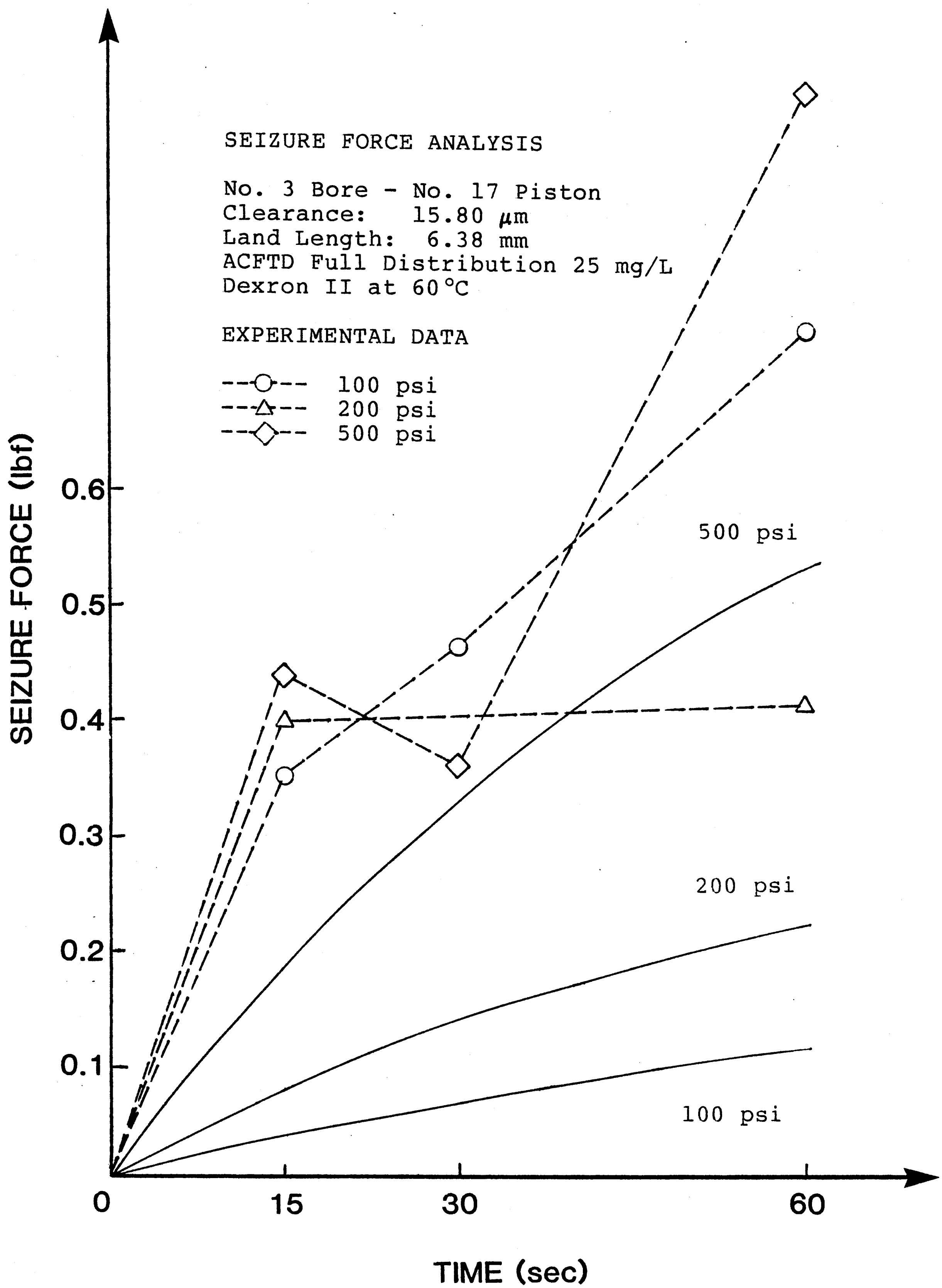
SEIZURE FORCE ANALYSIS

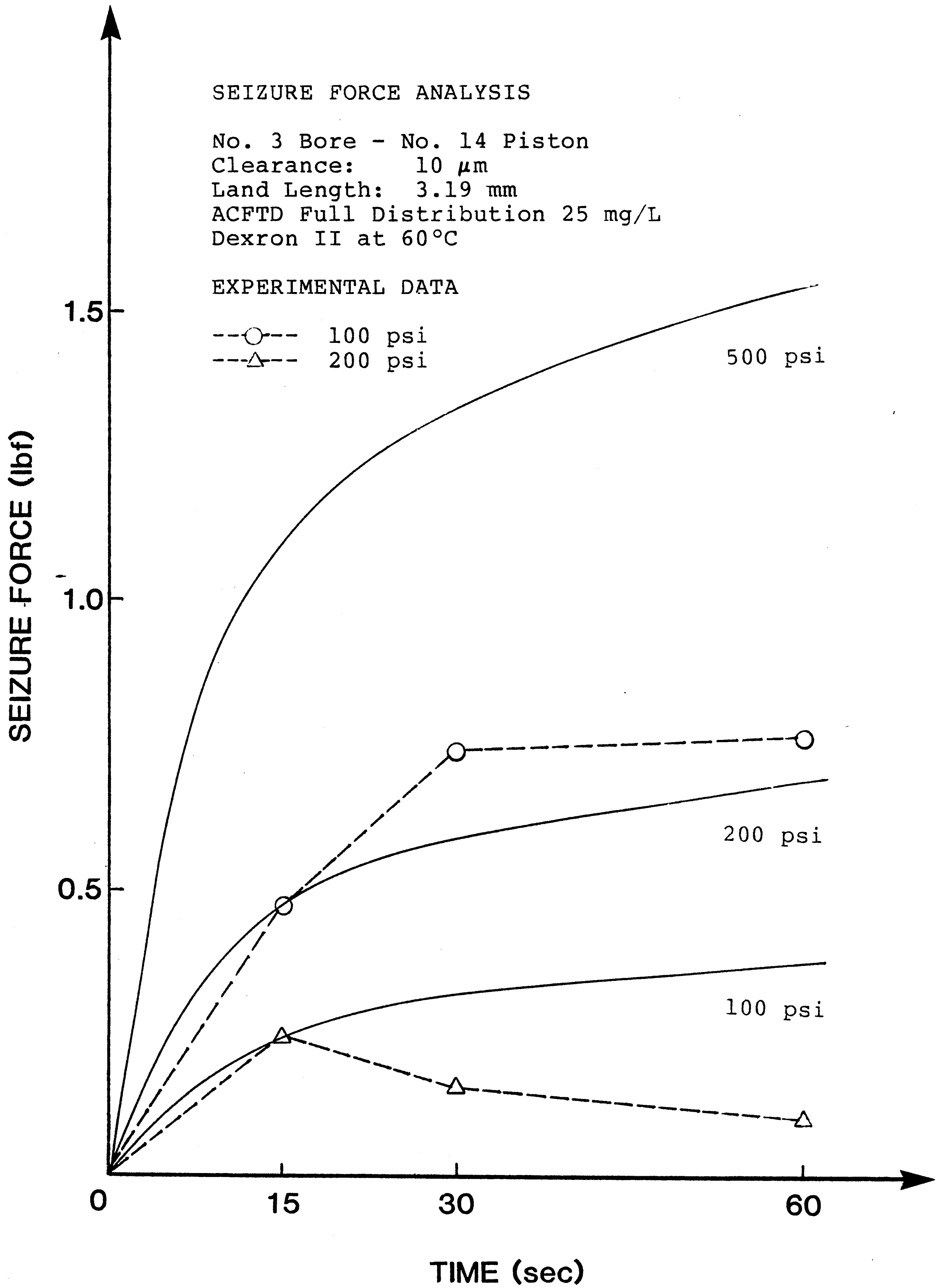

500 psi
1.5
SEIZURE FORCE ANALYSIS
No. 3 Bore - No. 22 Piston
Clearance: 10 μm
Land Length: 3.19 mm
ACFTD Full Distribution 25 mg/L
Dexron II at 60°C
EXPERIMENTAL DATA
100 psi
200 psi
500 psi
SEIZURE FORCE (lbf)
1.0
200 psi
0.5
100 psi
0
15
30
60
TIME (sec)

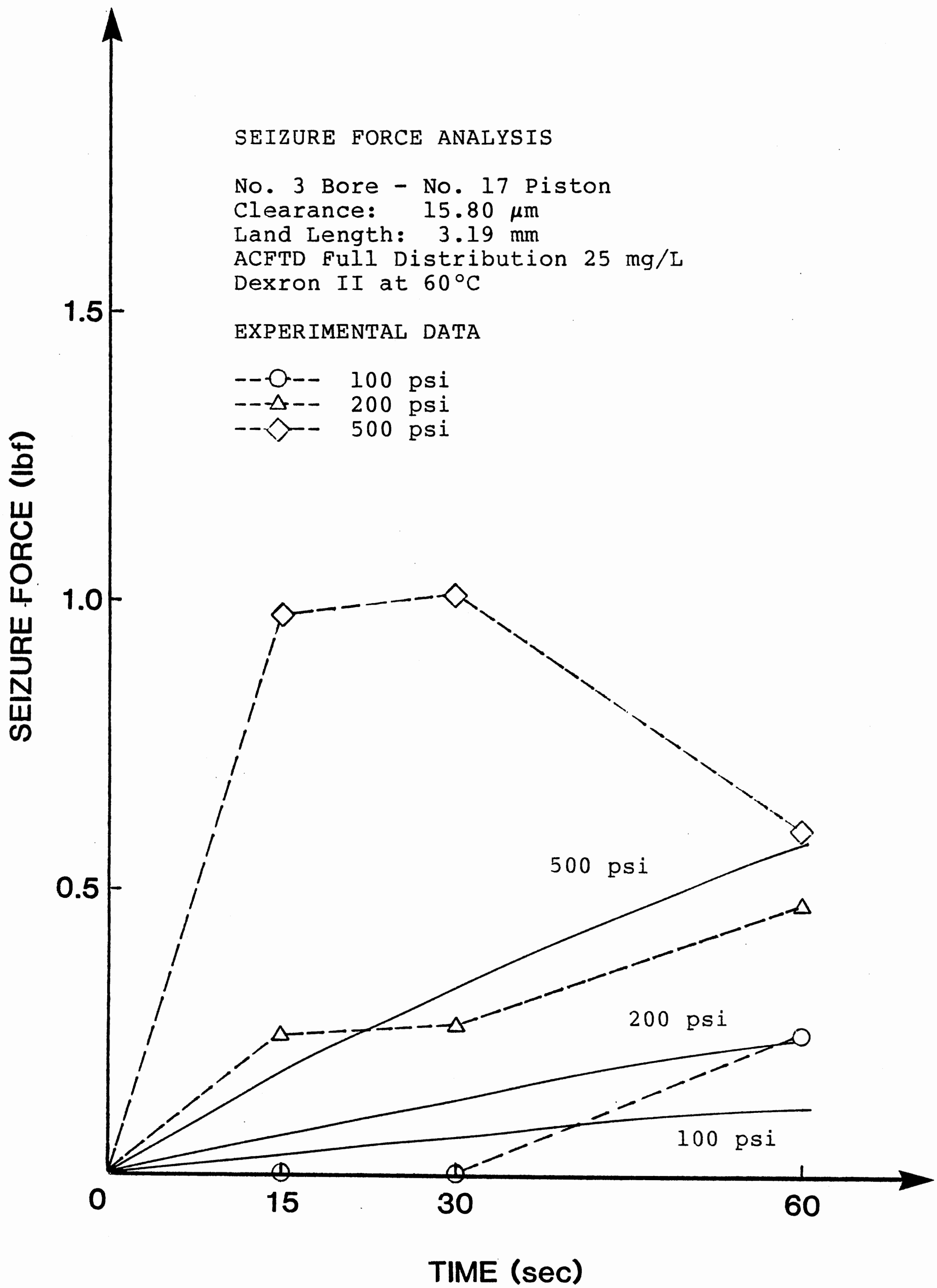

SEIZURE FORCE ANALYSIS
No. 3 Bore - No. 25 Piston
Clearance: 14.50 μm
Land Length: 6.38 mm
ACFTD Full Distribution 25 mg/L
Dexron II at 60°C
EXPERIMENTAL DATA
100 psi
200 psi
500 psi
500 psi
200 psi
100 psi
SEIZURE FORCE (lbf)
3.0
2.0
1.0
0
15
30
60
TIME (sec)

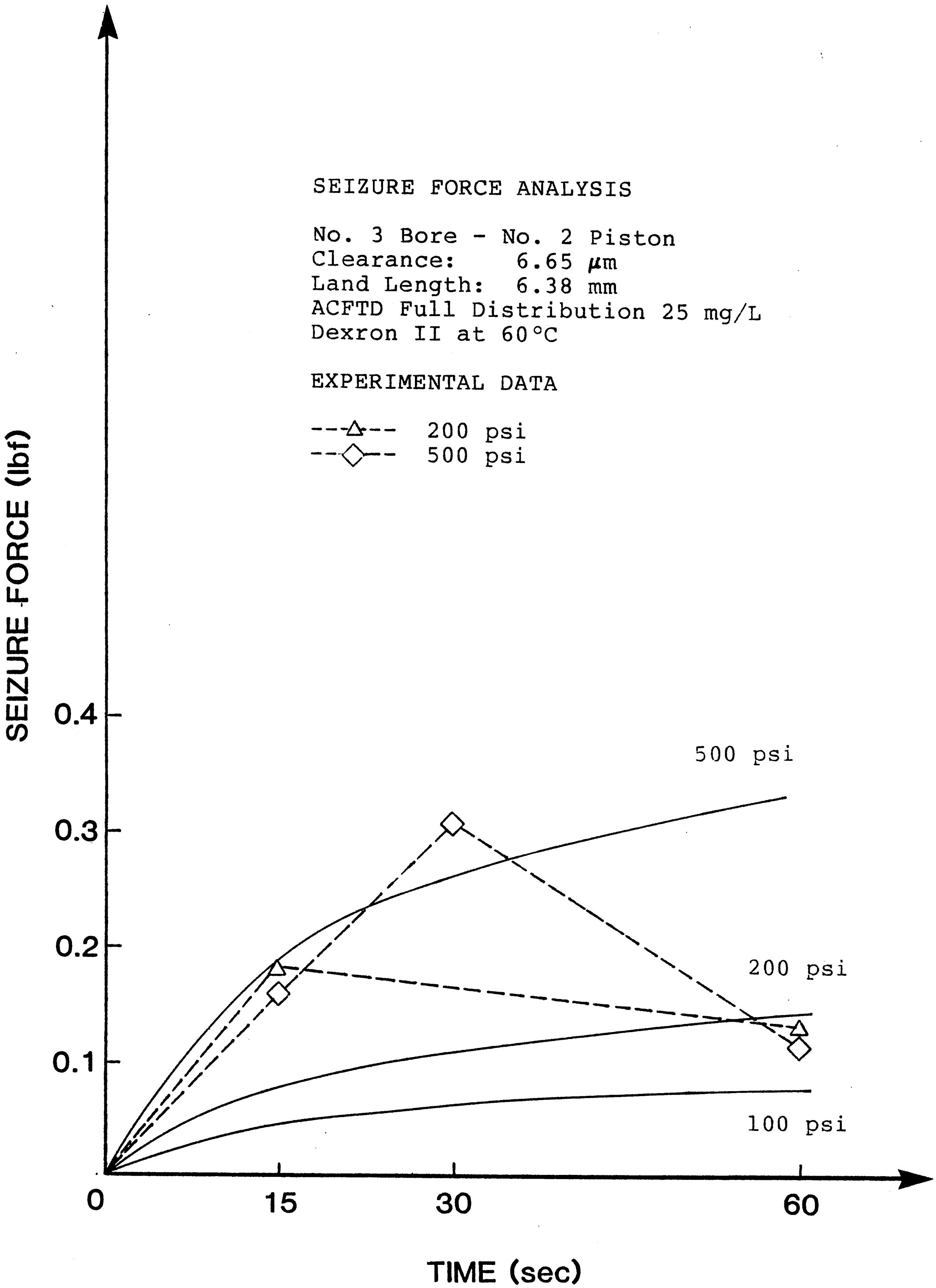

SEIZURE FORCE ANALYSIS
No. 3 Bore - No. 25 Piston
Clearance: 14.50 μm
Land Length: 3.19 mm
ACFTD Full Distribution 25 mg/L
Dexron II at 60°C
EXPERIMENTAL DATA
100 psi
200 psi
500 psi
500 psi
200 psi
100 psi
SEIZURE FORCE (lbf)
3.0
2.0
1.0
0
15
30
60
TIME (sec)

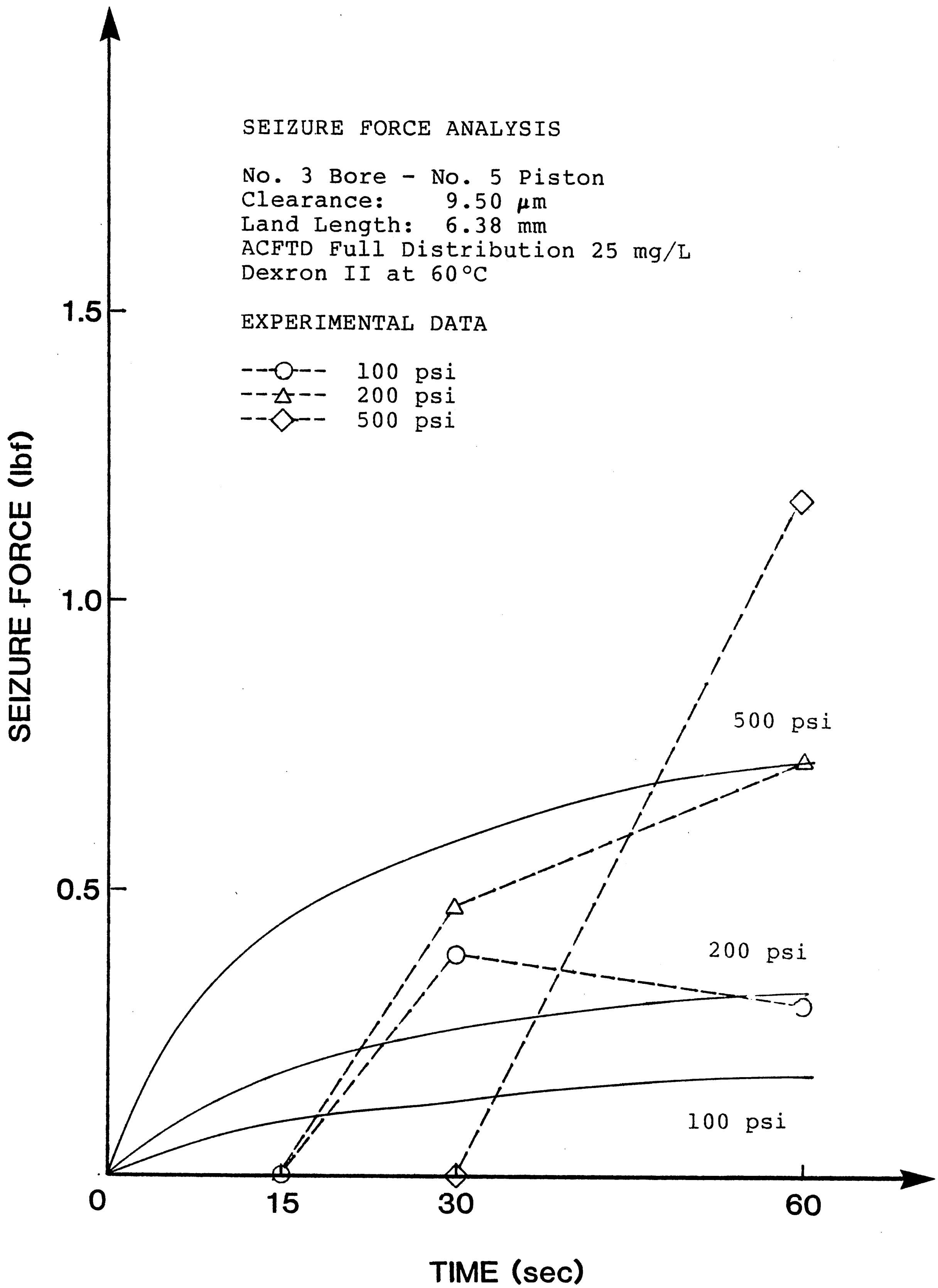

SEIZURE FORCE ANALYSIS
No. 3 Bore - No. 11 Piston
Clearance: 6.85 μm
Land Length: 6.38 mm
ACFTD Full Distribution 25 mg/L
Dexron II at 60°C
EXPERIMENTAL DATA
100 psi
200 psi
500 psi
SEIZURE FORCE (lbf)
1.5
1.0
0.5
500 psi
200 psi
100 psi
0
15
30
60
TIME (sec)

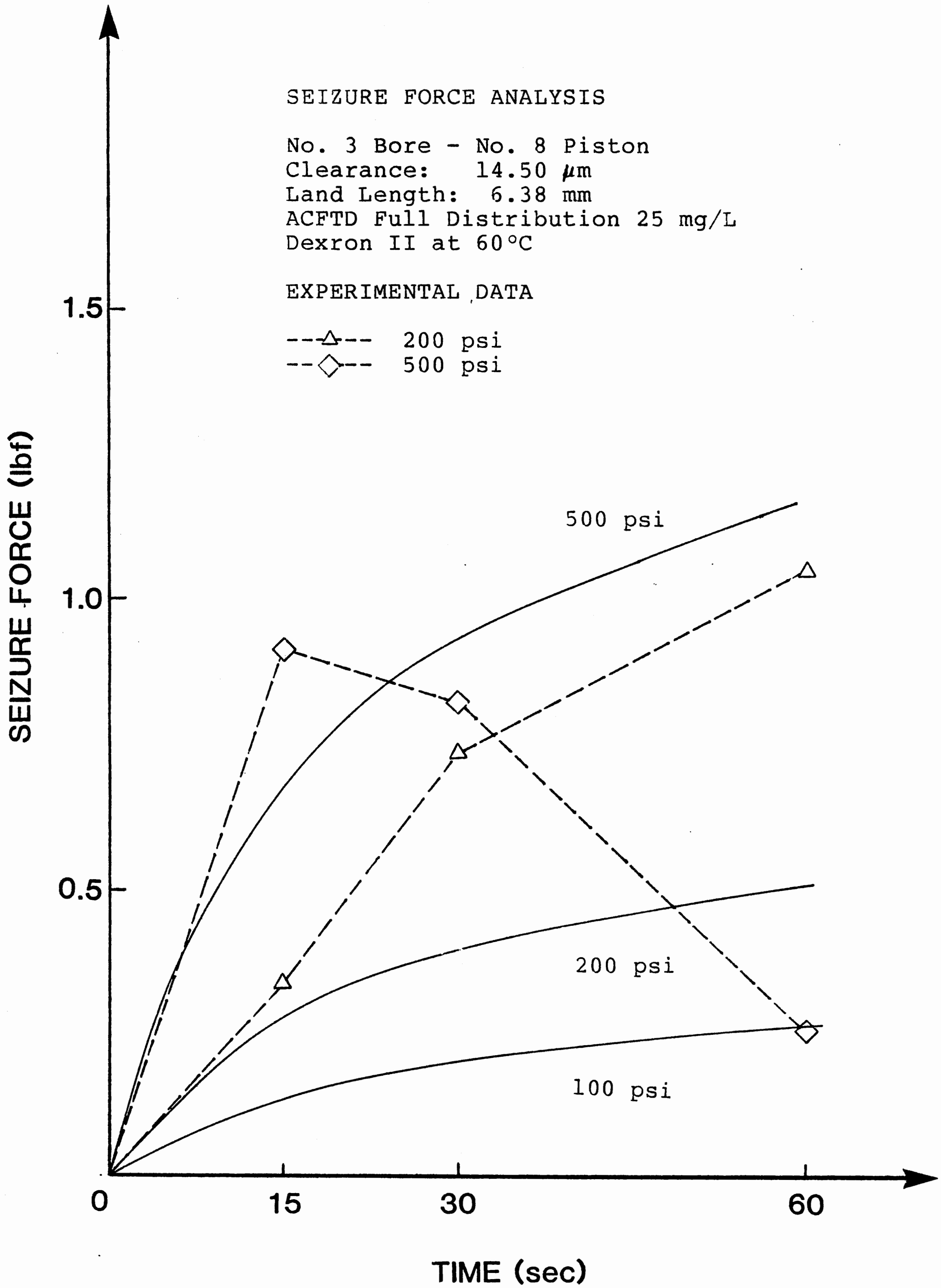

SEIZURE FORCE ANALYSIS
No. 3 Bore - No. 17 Piston
Clearance: 15.80 μm
Land Length: 6.38 mm
ACFTD Full Distribution 25 mg/L
Dexron II at 60°C
EXPERIMENTAL DATA
100 psi
200 psi
500 psi
500 psi
200 psi
100 psi
SEIZURE FORCE (lbf)
0.6
0.5
0.4
0.3
0.2
0.1
0
15
30
60
TIME (sec)

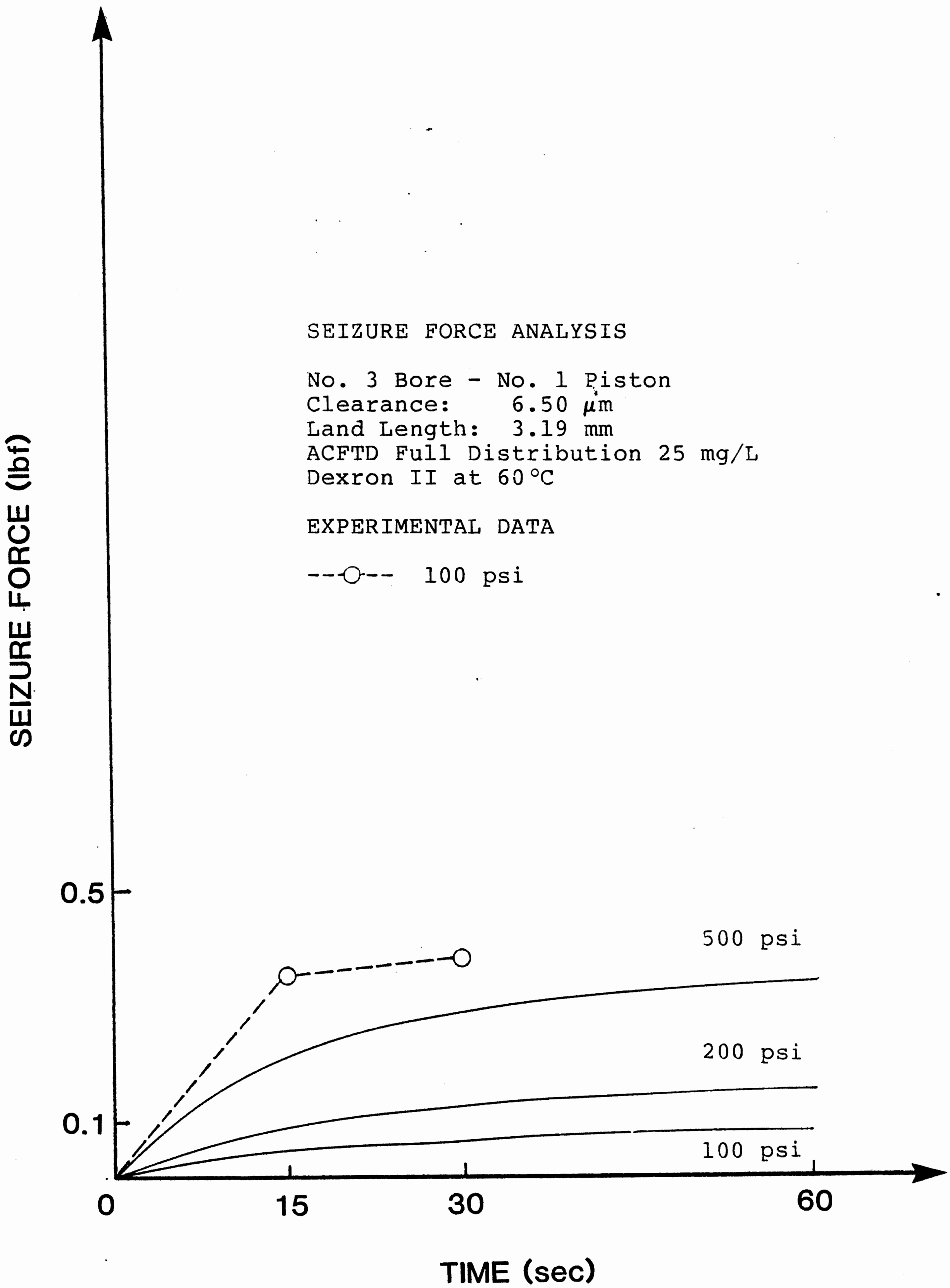

SEIZURE FORCE ANALYSIS
No. 3 Bore - No. 14 Piston
Clearance: 10 μm
Land Length: 3.19 mm
ACFTD Full Distribution 25 mg/L
Dexron II at 60°C
EXPERIMENTAL DATA
100 psi
200 psi
500 psi
200 psi
100 psi
1.5
1.0
0.5
SEIZURE FORCE (lbf)
0
15
30
60
TIME (sec)

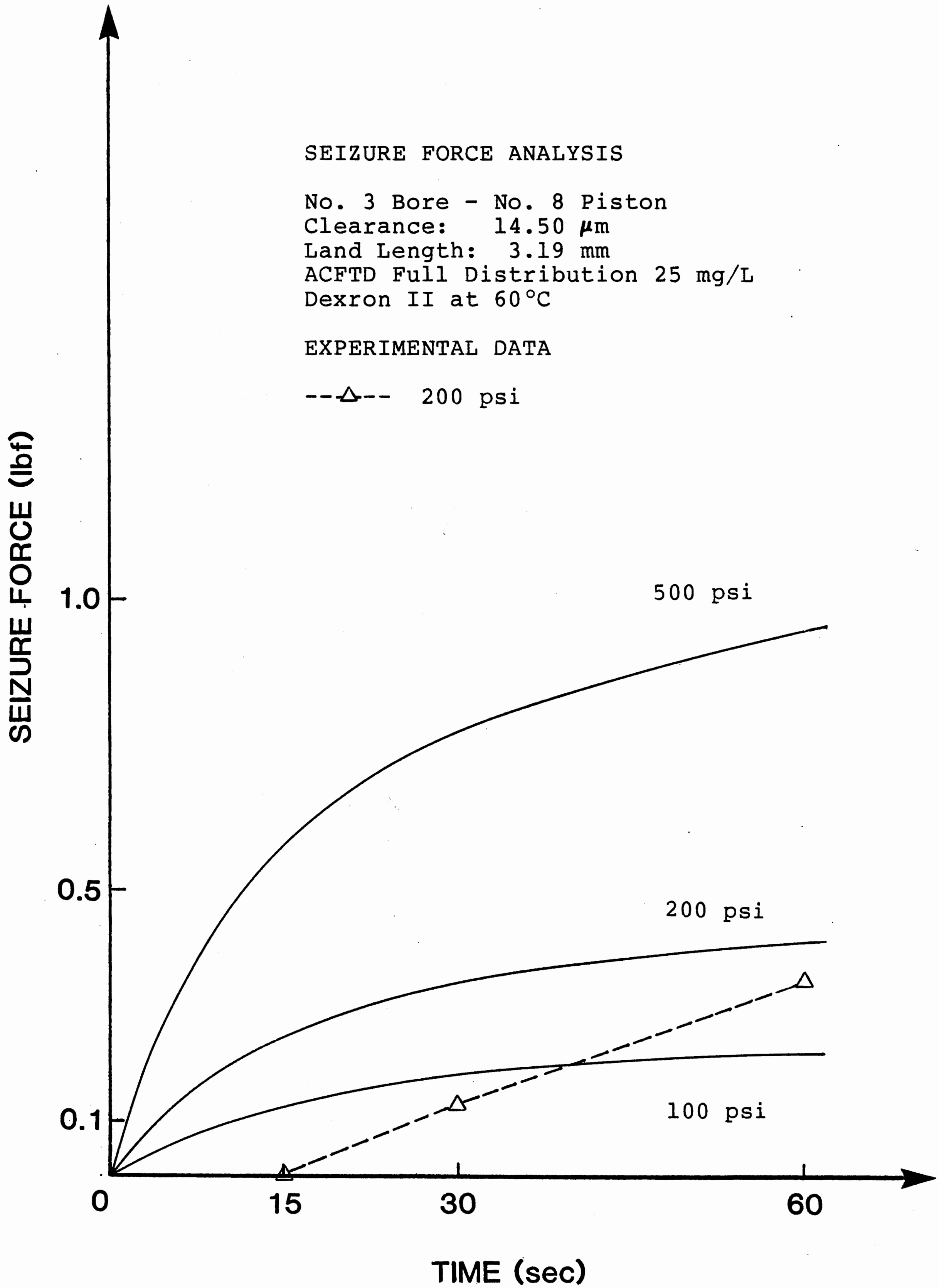

SEIZURE FORCE ANALYSIS
No. 3 Bore - No. 17 Piston
Clearance: 15.80 µm
Land Length: 3.19 mm
ACFTD Full Distribution 25 mg/L
Dexron II at 60°C
EXPERIMENTAL DATA
100 psi
200 psi
500 psi
500 psi
200 psi
100 psi
SEIZURE FORCE (lbf)
1.5
1.0
0.5
0
15
30
60
TIME (sec)


SEIZURE FORCE ANALYSIS
No. 3 Bore - No. 2 Piston
Clearance: 6.65 μm
Land Length: 6.38 mm
ACFTD Full Distribution 25 mg/L
Dexron II at 60°C
EXPERIMENTAL DATA
200 psi
500 psi
SEIZURE FORCE (lbf)
0.4
0.3
0.2
0.1
500 psi
200 psi
100 psi
0
15
30
60
TIME (sec)


SEIZURE FORCE ANALYSIS
No. 3 Bore - No. 5 Piston
Clearance: 9.50 μm
Land Length: 6.38 mm
ACFTD Full Distribution 25 mg/L
Dexron II at 60°C
EXPERIMENTAL DATA
100 psi
200 psi
500 psi
500 psi
200 psi
100 psi
1.5
1.0
0.5
0
15
30
60
SEIZURE FORCE (lbf)
TIME (sec)


SEIZURE FORCE ANALYSIS
No. 3 Bore - No. 8 Piston
Clearance: 14.50 μm
Land Length: 6.38 mm
ACFTD Full Distribution 25 mg/L
Dexron II at 60°C
EXPERIMENTAL DATA
200 psi
500 psi
500 psi
200 psi
100 psi
1.5
1.0
0.5
0
15
30
60
SEIZURE FORCE (lbf)
TIME (sec)


SEIZURE FORCE ANALYSIS
No. 3 Bore - No. 1 Piston
Clearance: 6.50 μm
Land Length: 3.19 mm
ACFTD Full Distribution 25 mg/L
Dexron II at 60°C
EXPERIMENTAL DATA
---O--- 100 psi
SEIZURE FORCE (lbf)
0.5
0.1
500 psi
200 psi
100 psi
0
15
30
60
TIME (sec)


SEIZURE FORCE ANALYSIS
No. 3 Bore - No. 8 Piston
Clearance: 14.50 μm
Land Length: 3.19 mm
ACFTD Full Distribution 25 mg/L
Dexron II at 60°C
EXPERIMENTAL DATA
200 psi
500 psi
200 psi
100 psi
SEIZURE FORCE (lbf)
1.0
0.5
0.1
0
15
30
60
TIME (sec)

VITA

Tokunosuke Ito

Candidate for the Degree of

Doctor of Philosophy

Thesis: PARTICULTATE-INDUCED SEIZURE OF PISTON-BORE ASSEMBLIES

Major Field: Mechanical Engineering

Biographical:

Personal Data: Born in Tokyo, Japan, on March 15, 1958; son of Mr. and Mrs. Seihachi Ito; married in Stillwater, Oklahoma, on May 20, 1986, to Linda Ann Stark.

Education: Graduated from Igusa High School, Tokyo, Japan, in March, 1977; received a Bachelor of Science degree in Mechanical Engineering with an Aerospace Option from Oklahoma State University in May, 1982; received a Master of Science degree in Mechanical Engineering from Oklahoma State University in May, 1985; completed requirements for the Doctor of Philosophy degree at Oklahoma State University in July, 1987.

Professional Experience: Project Associate, Fluid Power Research Center, Stillwater, Oklahoma, 1982-1983; Project Engineer, Fluid Power Research Center, 1984-1985; Technical Translator (English/Japanese) for Fluid Power Research Center, summers 1984-1985; Project Engineer, Fluid Power Research Center, 1986-1987; Graduate Teaching Assistant, Oklahoma State University, 1986-1987.