DOE/CE/90025--3

# CAM/LIFTER FORCES AND FRICTION

DOE/CE/90025--3

DE92 011573

David J. Gabbey, Graduate Student
Jongmin Lee, Graduate Student
Donald J. Patterson, Professor

Final Report

for the Period September 15, 1988 - November 30, 1991

The project was supported by the:

**U.S. Department of Energy**

Heavy Duty Transportation Technology Program
John Fairbanks

Energy Conservation and Utilization Technology Program
David Mello

and was work performed by the:
Department of Mechanical Engineering and Applied Mechanics
W. E. Lay Automotive Laboratory
The University of Michigan
Ann Arbor, Michigan 48109

February, 1992



Prepared for

THE U. S. DEPARTMENT OF ENERGY
AGREEMENT NO. DE-FG02-88CE90025

ACKNOWLEDGEMENTS

The authors wish to thank the U.S. Department of Energy, and especially Mr. John Fairbanks and Mr. David Mello for their technical and financial support. Thanks also go to Mr. Greg Woodworth, Mr. Nick Skull, and Mr. Terry Shaw of the Cummins Engine Company for providing extensive technical assistance throughout the program. In addition, we thank Mr. Russell R. Mitchell, Mr. George Schwartz and Mr. Kevin Morrison for their contributions.

TABLE OF CONTENTS

# LIST OF FIGURES

FIGURE

ABSTRACT

This report details the procedures used to measure the cam/lifter forces and friction. The present effort employed a Cummins LTA-10, and focuses on measurements and dynamic modeling of the injector train. The program was sponsored by the U.S. Department of Energy in support of advanced diesel engine technology. The injector train was instrumented to record the instantaneous roller speed, roller pin friction torque, pushrod force, injector link force and cam speed. These measurements, together with lift profiles for pushrod and injector link displacement, enabled the friction work loss in the injector train to be determined. Other significant design criteria such as camshaft roller follower slippage and maximum loads on components were also determined. Future efforts will concentrate on the dynamic model, with tests run as required for correlation.

## INTRODUCTION

As regulations limiting the quantities of exhaust emissions from diesel engines are made increasingly strict, advances in diesel engine technology must be made to keep pace. A significant factor for the reduction of particulates is known to be the fuel injection pressure; higher pressures result in better atomization of the fuel, resulting in faster, more complete combustion, higher efficiency and fewer particulates. In general, it is hoped that injection pressures double current practice can be used.

To increase the injection pressure without sacrificing reliability in a Cummins L-10 engine, the injector train must be strengthened from camshaft to injector. Thus, the incentive driving this research was the need to develop the technology to measure the magnitude, duration and cause of various forces in the injector train.

This project was initiated in September of 1988. It is jointly funded within the U.S. Department of Energy by the Heavy Duty Transportation Technology Program through Mr. John Fairbanks, and the Energy Conservation and Utilization Technology Program, Mr. David Mello. Originally the experimental efforts centered around instrumenting the exhaust valvetrain of the number one cylinder of the L-10 engine. The physical access to this area of the cylinder head was good, and the techniques were relatively easy to implement there. This effort was detailed in Reference 1. After successfully completing this, the apparatus was transferred to the injector train. Now efforts are being made to develop a dynamic model of the injector train in the L-10 engine.

**Reference 1:** Mitchell and Patterson, INJECTOR AND VALVE TRAIN FRICTION STUDIES, CUMMINS L-10 ENGINE; Progress Report, 01/10/91 to 31/12/91; The University of Michigan.

HARDWARE OVERVIEW

A "broken-in" Cummins L-10 engine, an inline six cylinder, four-stroke, direct injection diesel, was used for the project (Figure 1). Since firing the engine was not a necessity for examining the characteristics of the valve and injector trains, all pistons and connecting rods were removed. An electronically controlled fuel pump replaced the original, and the engine's oil capacity was reduced to decrease the warm-up period and minimize the volume required of any special test oils. The oil galleries were modified so that the valve and injector trains were lubricated independently of the rest of the engine by an external pump and heater. Reference 1 contains details of the set-up.

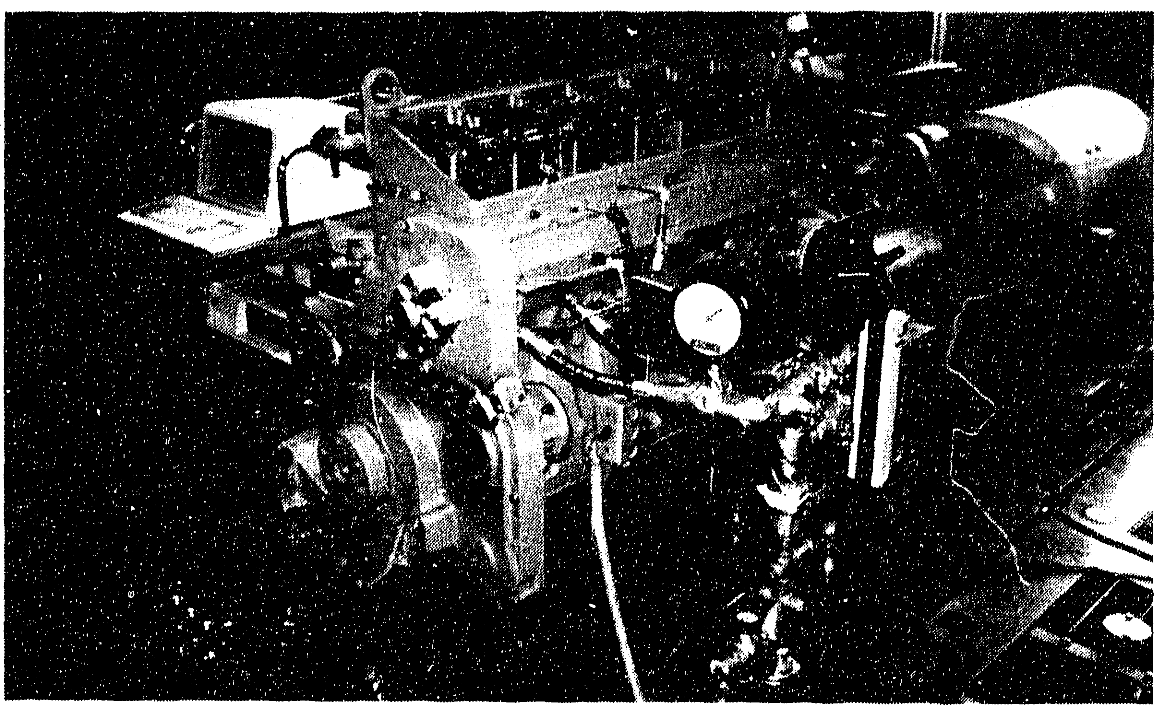


Figure 1: Overview: Cummins L-10 engine installed in Room 244 of the University of Michigan Automotive Laboratory

Because of space considerations, cylinder number two was instrumented for the injector train. Within its bore was a specially machined funnel to collect the injected fuel, which then drained to a resevoir outside of the block. The other five injectors were blocked off to prevent fuel dilution of the oil in the sump. Cylinder two had only the injector train installed due to limited space. The remaining cylinders had only the valve train installed, because the injectors would be damaged if operated without fuel (Figure 2). Having most of the valve train present served to reduce the camshaft speed variation. The injector was adjusted to Cummins' specifications, while the lash on the valves was reduced to zero because they did not warm up during a run. This also decreased the speed variation.

When motored by the dynomometer, the controllable variables included:

1. Engine/camshaft speed (100-2200 rpm at crank)
2. Oil temperature (ambient-200°F)
3. Oil pressure (0-65 psi)
4. Oil type
5. Fuel pressure to injector (0-185 psi)

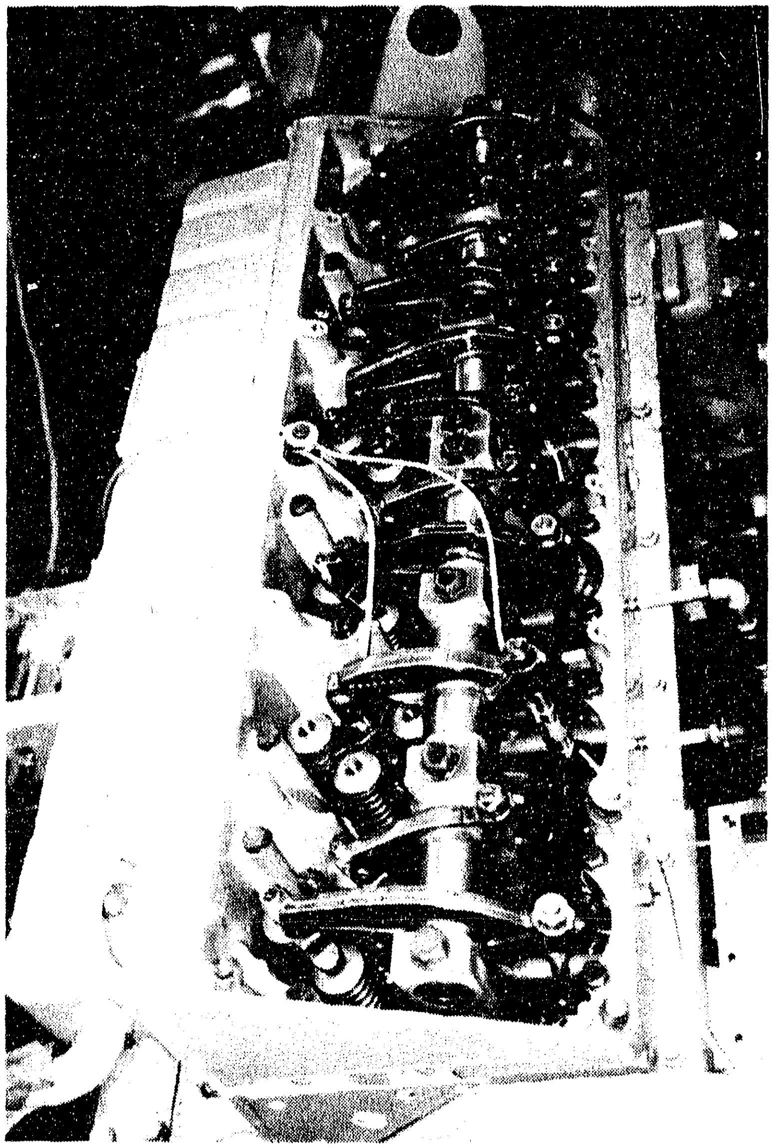


Figure 2: Upper Valve and Injector Trains

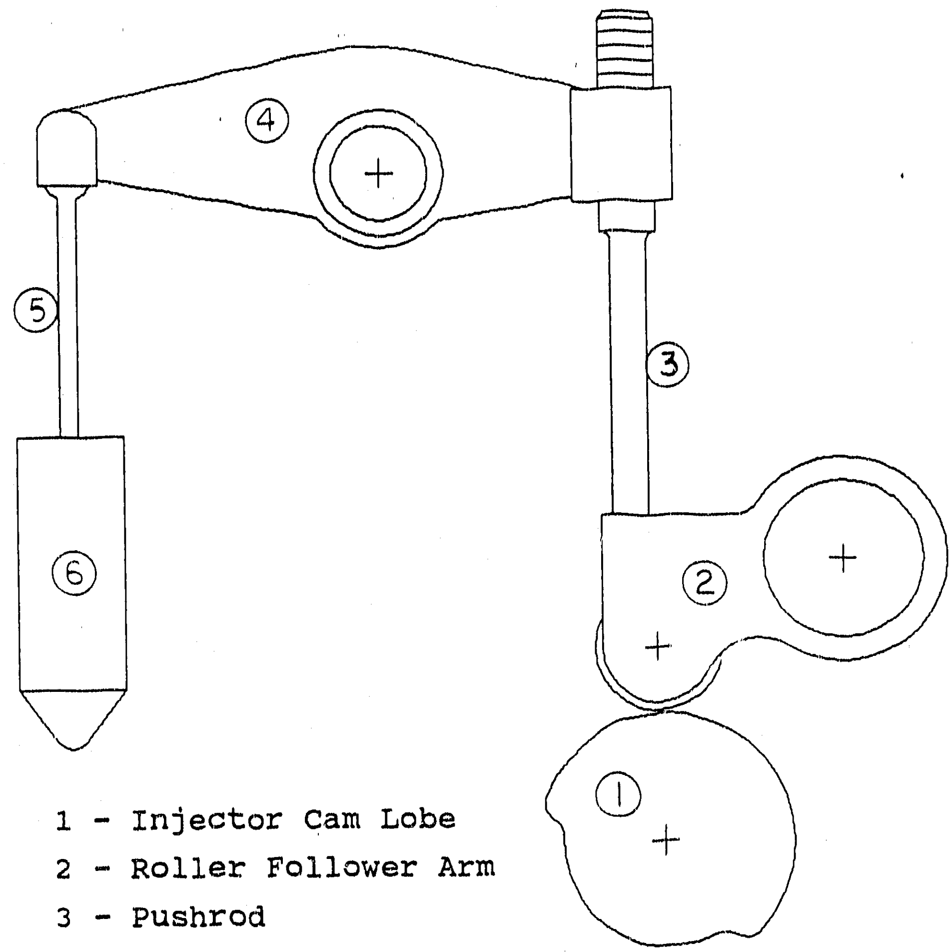
## THEORETICAL CONSIDERATIONS

Upper injector train net work over 360° of cam rotation is the difference between the work flowing into the pushrod during injection and that recovered when refilling. This work includes friction plus fluid pumping. Figure 3 is a schematic of the injector train. The rate of work, or power, transferred by the pushrod is the product of the force and the velocity in the direction of the force. Thus, under steady state conditions the net work is:

$$\text{NET WORK} = \int(\text{FORCE}*\text{VELOCITY})\,dt$$

where the integration is over one cycle.

The roller/cam instantaneous tractive force is determined from the net roller torque divided by the roller radius. The instantaneous roller torque arises from a combination of the instantaneous inertia torque and the roller pin friction torque. The inertia torque is determined from the product of the instantaneous angular acceleration, and the inertia of the roller. The instantaneous acceleration is determined from the speed variation of the roller with respect to its axis, and the motion of that axis as the rocker swings to move the pushrod. To summarize, the roller friction torque, angular acceleration, inertia and radius, and the follower arm angular acceleration are needed to determine the tractive force. The reader is referred to Reference 1 for more detail on the design and implementation of the measurements of pushrod force, roller angular acceleration and roller pin friction torque.



1 - Injector Cam Lobe
2 - Roller Follower Arm
3 - Pushrod
4 - Rocker Arm
5 - Injector Link
6 - Injector

Figure 3: Injector Train Schematic

## EXPERIMENTAL CONSIDERATIONS

The instantaneous measurements collected as a function of camshaft rotational angle were: pushrod and injector link forces; camshaft speed; roller follower pin torque; roller speed.

### Pushrod Force and Velocity

Forces were measured by means of full bridge strain gage circuits, one on the pushrod and one on the injector link (Figure 4). These were carefully calibrated in a lever arm fixture using a series of known masses to produce calibration curves presented as Figures 5 and 6. Both curves are linear and show no hysteresis. The wires from the components were firmly anchored and carefully routed to prevent breakage due to fatigue.

The pushrod velocity is required to compute the work done by the pushrod through one cycle, and is derived from an LVDT measurement of its displacement parallel to the force. The displacement at each degree of camshaft rotation was stored in a file read by the data processing program FLIFTER5. The same method is used for the injector link.

The acceleration of the camshaft roller follower arm is required for the calculation of roller tractive force. It is based on the double differentiation of the pushrod displacement.

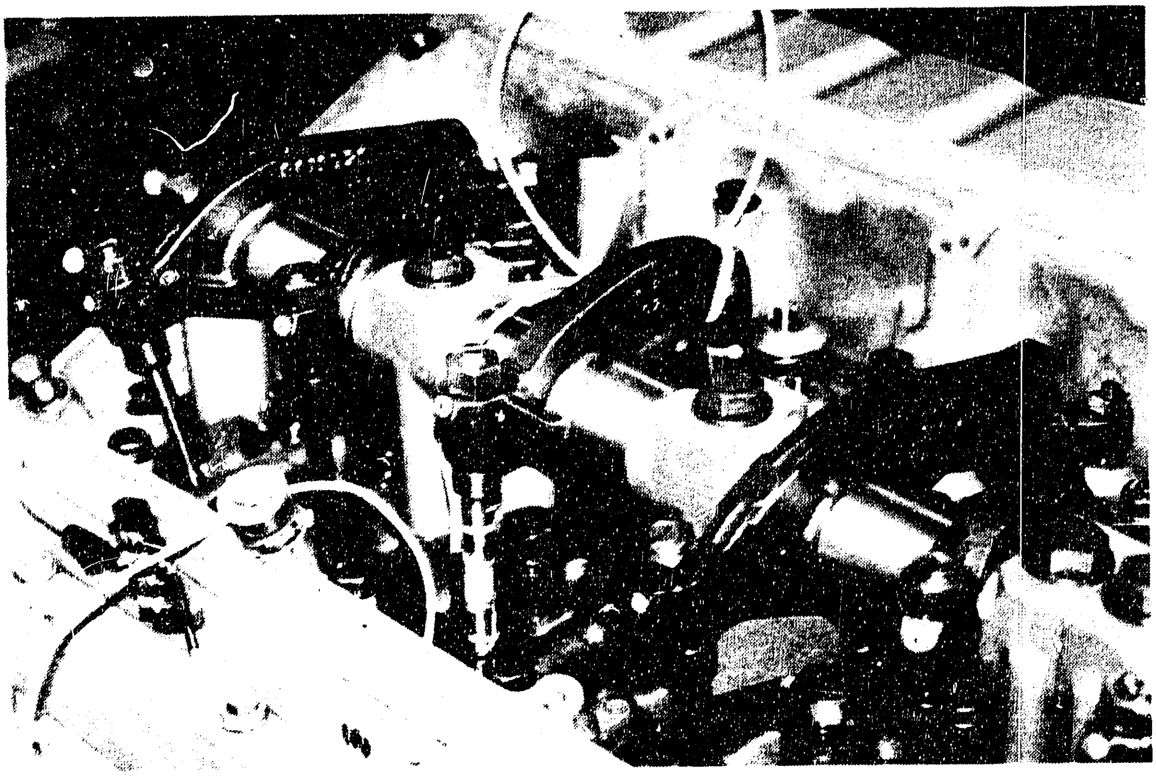
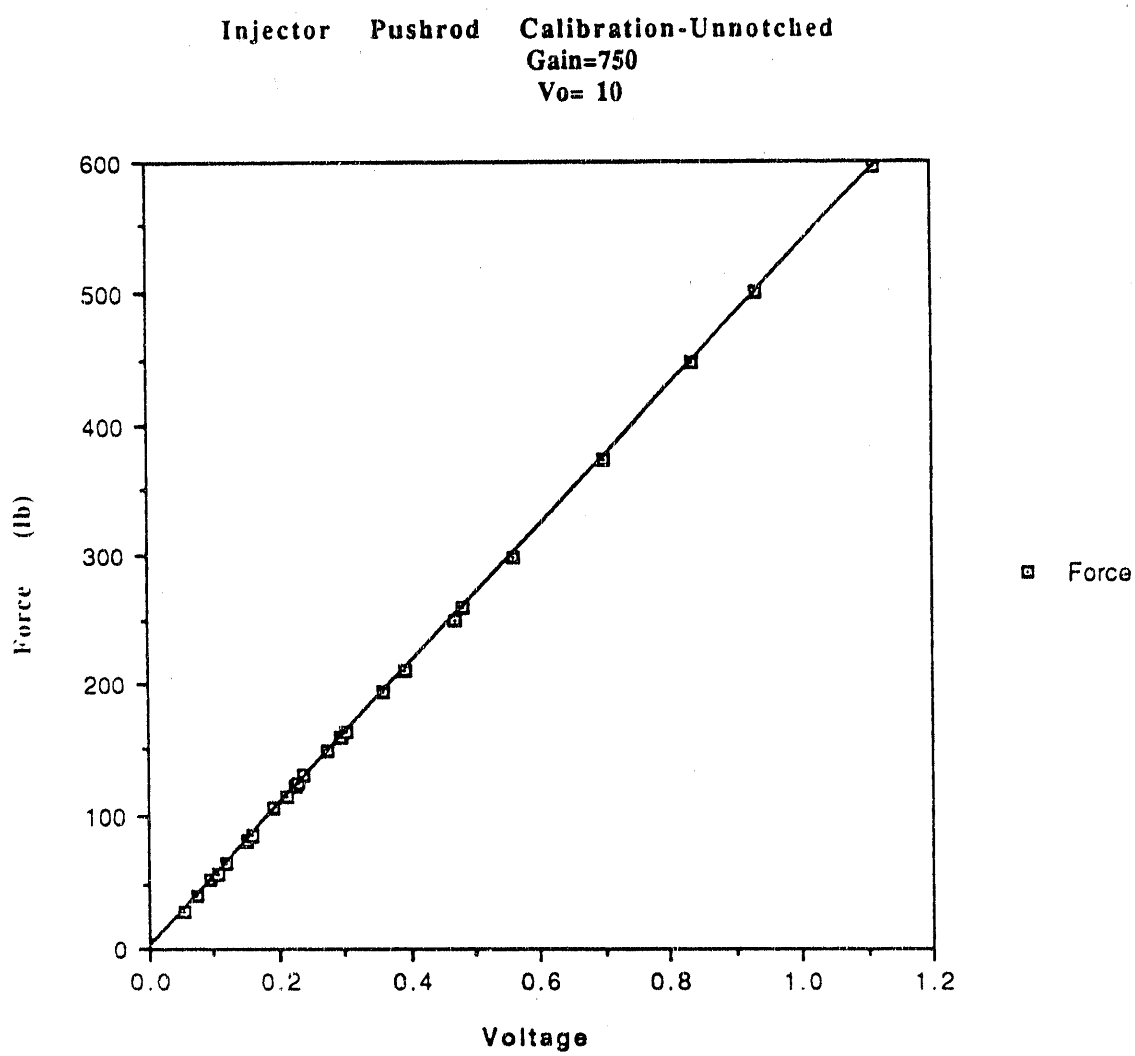
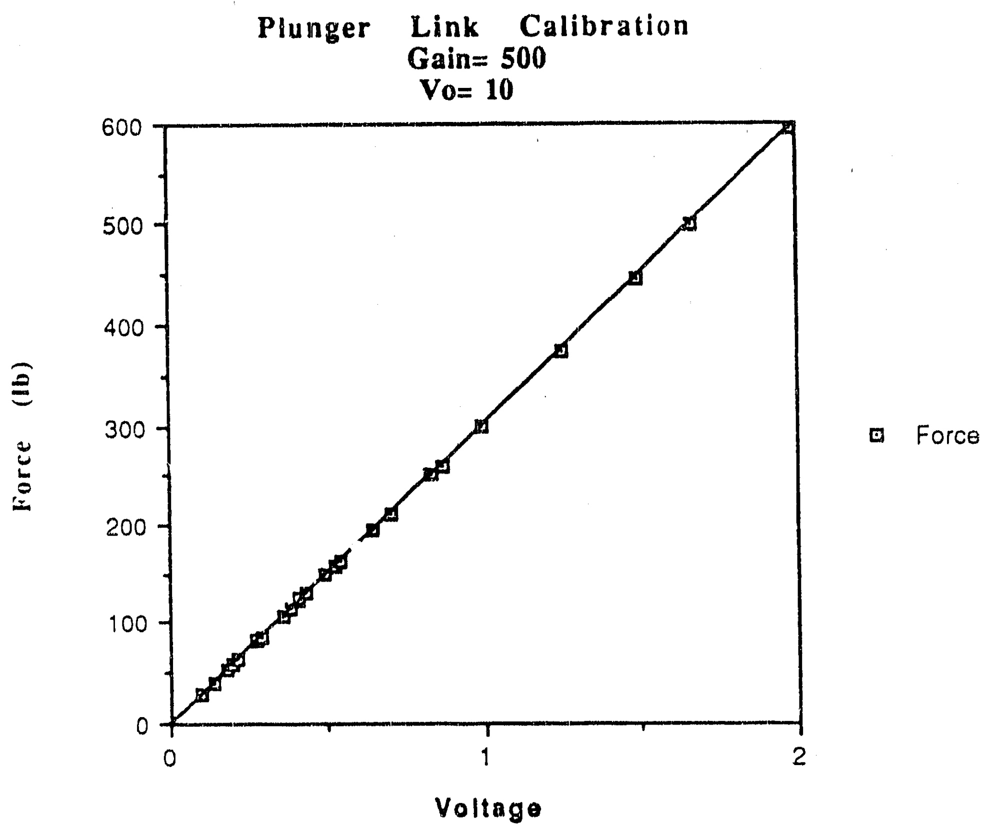


Figure 4: Upper Injector Train Detail. The two uppermost wires connect to the injector pushrod(left) and link (right) strain gage bridges.




Force = 1.9950 + 532.41(Voltage) (lb)

Figure 5




Force = - 3.5736e-3 + 300.03(Voltage)

Figure 6

Camshaft Speed

The cam speed is monitored during each run by a Datametrics optical speed encoder generating 720 pulses per revolution, driven from the front of the camshaft. In addition to speed, it also sends out a pulse to trip the timers in the data acquisition system, signaling the start of a cycle.

Roller Follower Pin Torque

A standard injector cam lobe follower was extensively modified in order to measure both the roller speed and roller pin torque instantaneously. Space restrictions within the engine made this a challenge.

The primary change was to mount the roller's pin, originally fixed in the follower arm, within two rolling element bearings. This necessitated building up the follower arm with weld material and remachining it, since the bearings were larger in diameter than the pin. A retainer was made to prevent the bearings from shifting. (This had happened previously with disasterous results.) The torque arm, machined from steel, was screwed rigidly to the roller's pin at one end, and has a loop at the other end around a pin set in the follower arm body (Figure 7). This prevents the roller's pin from rotating, and the torque required to prevent this is measured by a half bridge of strain gages on the sides of the torque arm's body. In addition, the torque arm has an oil passage drilled through it to supply the roller pin/roller interface with oil, since the original path was blocked by the bearings. The pin also had to be redrilled for this modification.

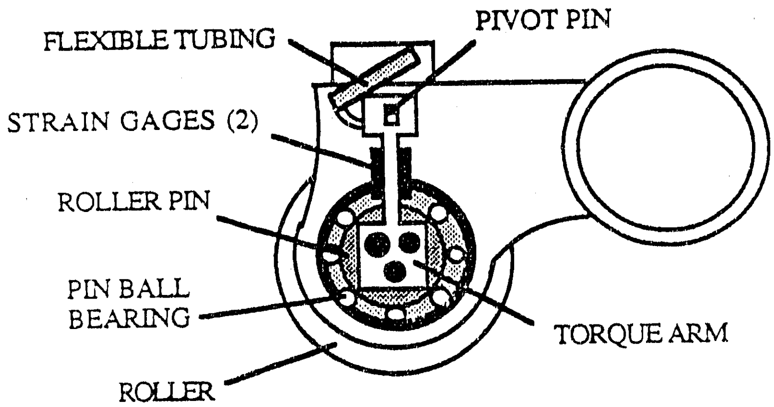



Figure 7: Torque Arm Assembly

As with the pushrod and injector link, the torque arm was calibrated in a special fixture using a series of known masses. Its curve was also linear and free of hysteresis(Figure 8).

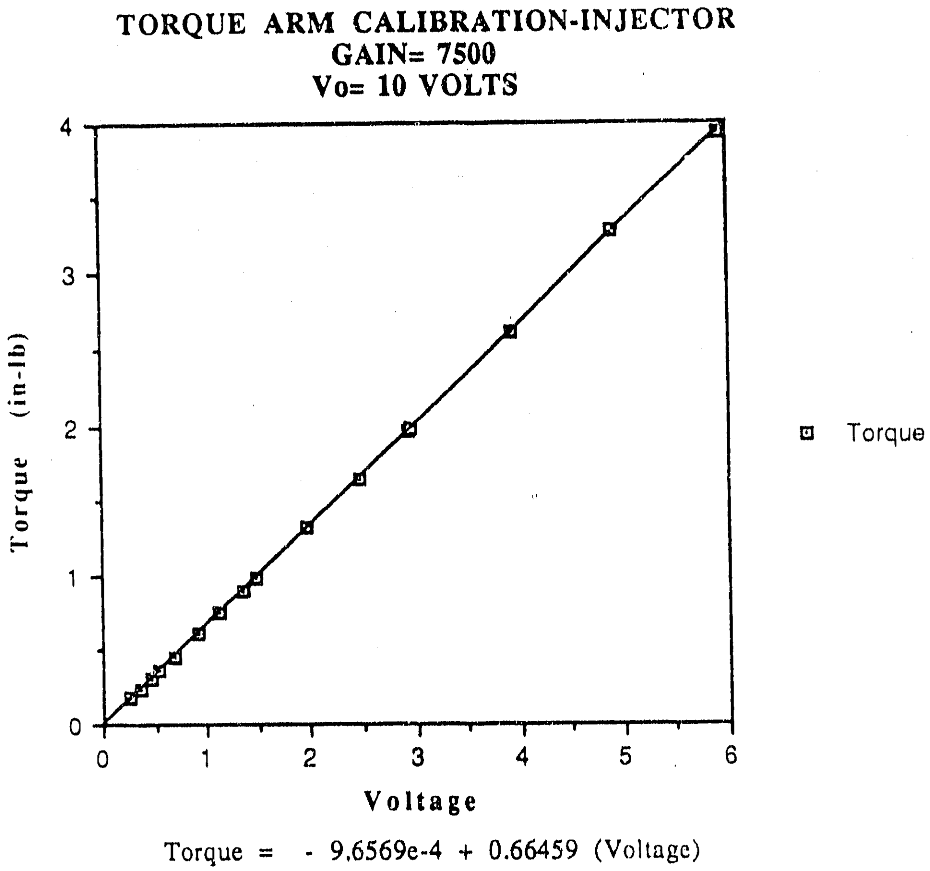

TORQUE ARM CALIBRATION-INJECTOR
GAIN= 7500
Vo= 10 VOLTS
4
3
2
1
0
Torque (in-lb)
0
1
2
3
4
5
6
Voltage
Torque
Torque = - 9.6569e-4 + 0.66459 (Voltage)

Figure 8

Roller Speed

In order to measure the roller speed, a custom encoder disc was designed, manufactured and affixed to the side of the roller opposite the torque arm (Figure 9). An optical pickup with its own light source was mounted just outside of the engine. It has a twelve inch fiber optic probe which was long enough to not only reach the encoder disc but also to bend in a coil around the follower's pivot for fatigue minimization during operation. See Reference 1 for details.

The final addition to the roller follower assembly was a capillary tube supplying compressed air to the encoder disc, just ahead of the optical pickup. This reduced the oil film thickness so that the pickup received reliable signals.

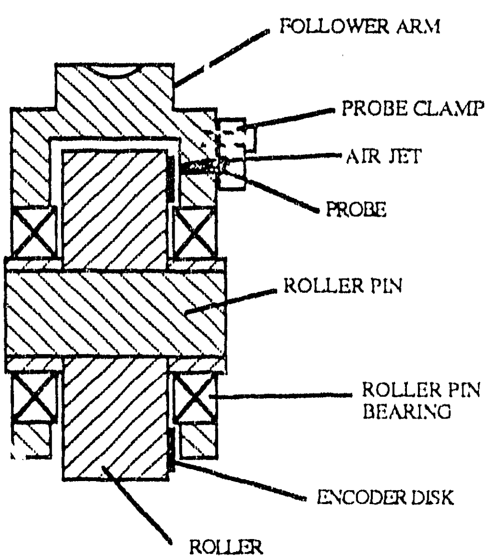



Figure 9: Cross Section of Roller Speed System

Data Acquisition System

The key to the success of the data acquisition system used in this project is its ability to simultaneously sample five channels, analog and digital: pushrod and injector link force; roller pin torque; roller speed; cam speed. To do this a series of sample and hold modules are used in parallel; each monitors its own channel and saves the signal when triggered. The signal is held until the computer has a chance to store it as data for the particular cam angle at which it was aquired at. A top dead center pulse from the cam speed encoder is used as a reference point for the system to define one cycle.

The computer used is an IBM PC, modified for data collection . The FORTRAN program VALVEIN3 (Appendix 1) was written to collect the test data; during a test it also continuously displayed five cycles of each of the five channels. To prevent inaccuracies from averaging too many cycles in a run, the program stops storing data after ten cycles, although the display is still updated. Each run's data is stored on a disk with the required constants for transferral to the data processing system.

Data Processing System

The data is processed by the FORTRAN program FLIFTER5 (Appendix 2) on a 386 IBM PC. The program takes the test data from the disk, giving the user the option of processing one cycle at a time or averaging all ten. It provides detailed plots, versus cam angle, of all measured data as well as the derivations based on it:

1. Pushrod velocity
2. Injector link velocity
3. Pushrod acceleration
4. Pushrod work
5. Averaged roller speed
6. Roller relative acceleration
7. Averaged roller relative acceleration
8. Follower arm acceleration
9. Inertia torque
10. Roller tractive force
11. Roller rotation relative to cam rotation

Note that some of the above graphs are labelled as averaged. This means that a five point sliding average was used to smooth the curves - in fact, except for number 6. above, all acceleration plots and those based on them used the five point sliding average, since differentiation tends to accentuate noise significantly. All results presented in this report are also based on the average of ten consecutive cycles.

RESULTS

To date there have been 22 successful tests run using the instrumented injector train, plus dozens of other informal tests performed while adjusting the equipment. Each test was composed of three to nine runs. Between runs within a test, only one variable, such as cam speed, would be changed. A complete set of data plots for Test #9102102 is included as Appendix 3. This test simulated the conditions under which the engine produces its maximum torque: 1350 crankshaft rpm; 140 psi fuel supply pressure; 45 psi oil pressure. All engine components were as supplied by Cummins Engine Co., except those modified as described in the Experimental Considerations section.

The following examination of test results refers to a comparison between Test #9102102, described above, and Test #9102201. The latter test was run at a steady 140 crankshaft rpm with all other variables the same. This minimized the dynamic forces. While examining these plots it may be useful to refer to Figure 18 of the injector displacement, noting that the injection of fuel occurred during the transition from base circle to maximum lift. Beyond this, the injector remains closed for a period (while on the outer base circle) to prevent the combustion gases from entering the injector.

Pushrod Force

The effect of engine speed is readily apparent in the plots of pushrod force, Figure 10. The higher speed curve rises sooner because of the compression of the fuel in the injector, which occurs before it can be sprayed out due to the restriction of the nozzle. The low speed run does not display this trait, since the fuel is injected at the same

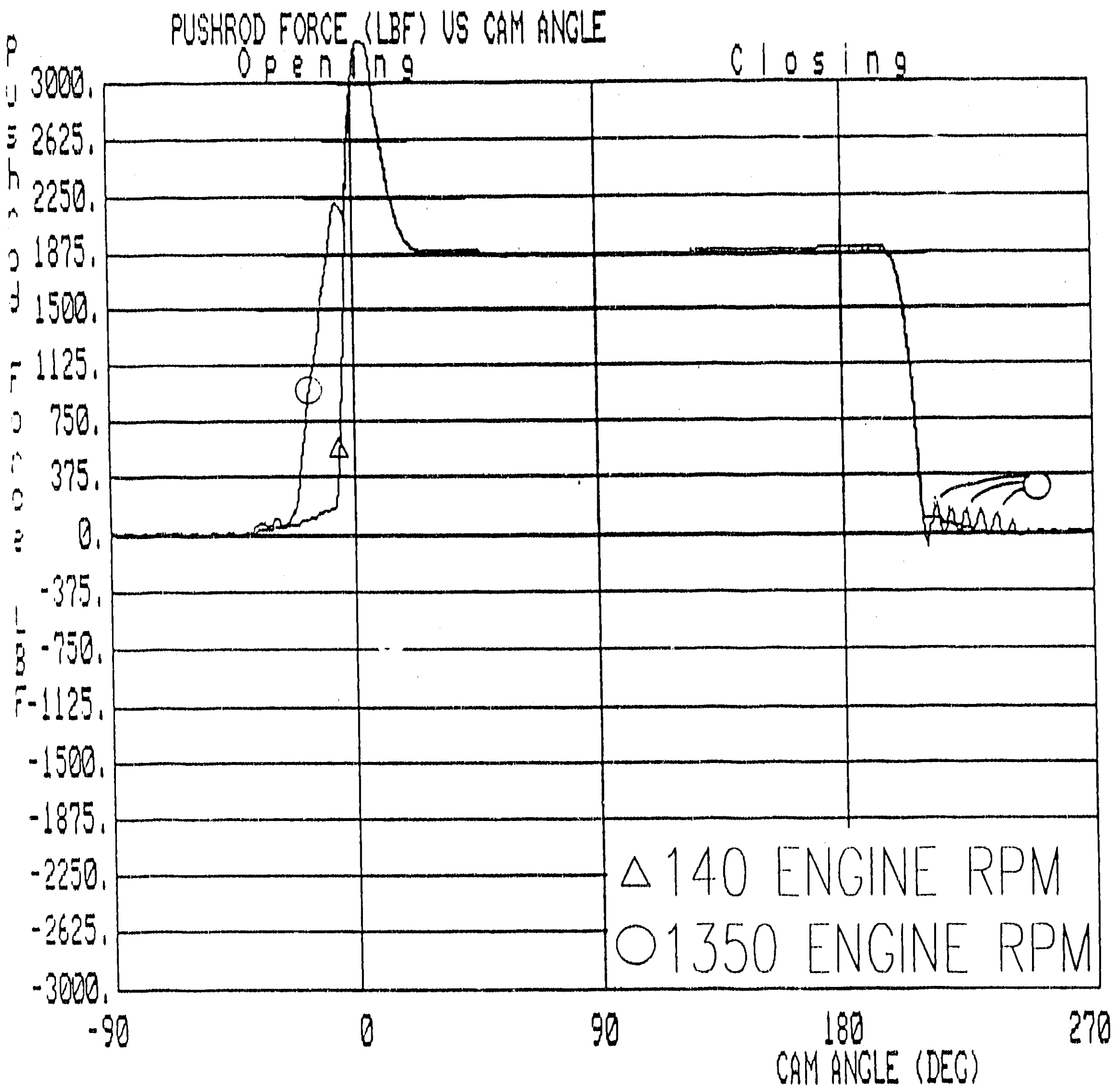



Figure 10: Injector Pushrod Force

rate at which it is compressed . The large peak was caused by the injector bottoming on its seat. As seen by the injector displacement curve, this pressure was relaxed after initial combustion, and the force settled to a steady state value at 1875 pounds. Afterwards, the force droped as the cam allowed the injector to refill.

Another characteristic at the higher engine speed is the bouncing of the pushrod/roller follower assembly upon its return to the base circle. This phenomenon is first detectable at about 800 rpm, and increases in magnitude with increasing speed. Although relatively small in terms of force, the bounces produce very sharp peaks which would tend to accelerate wear by delaying the formation of an elastohydrodynamic lubrication state.

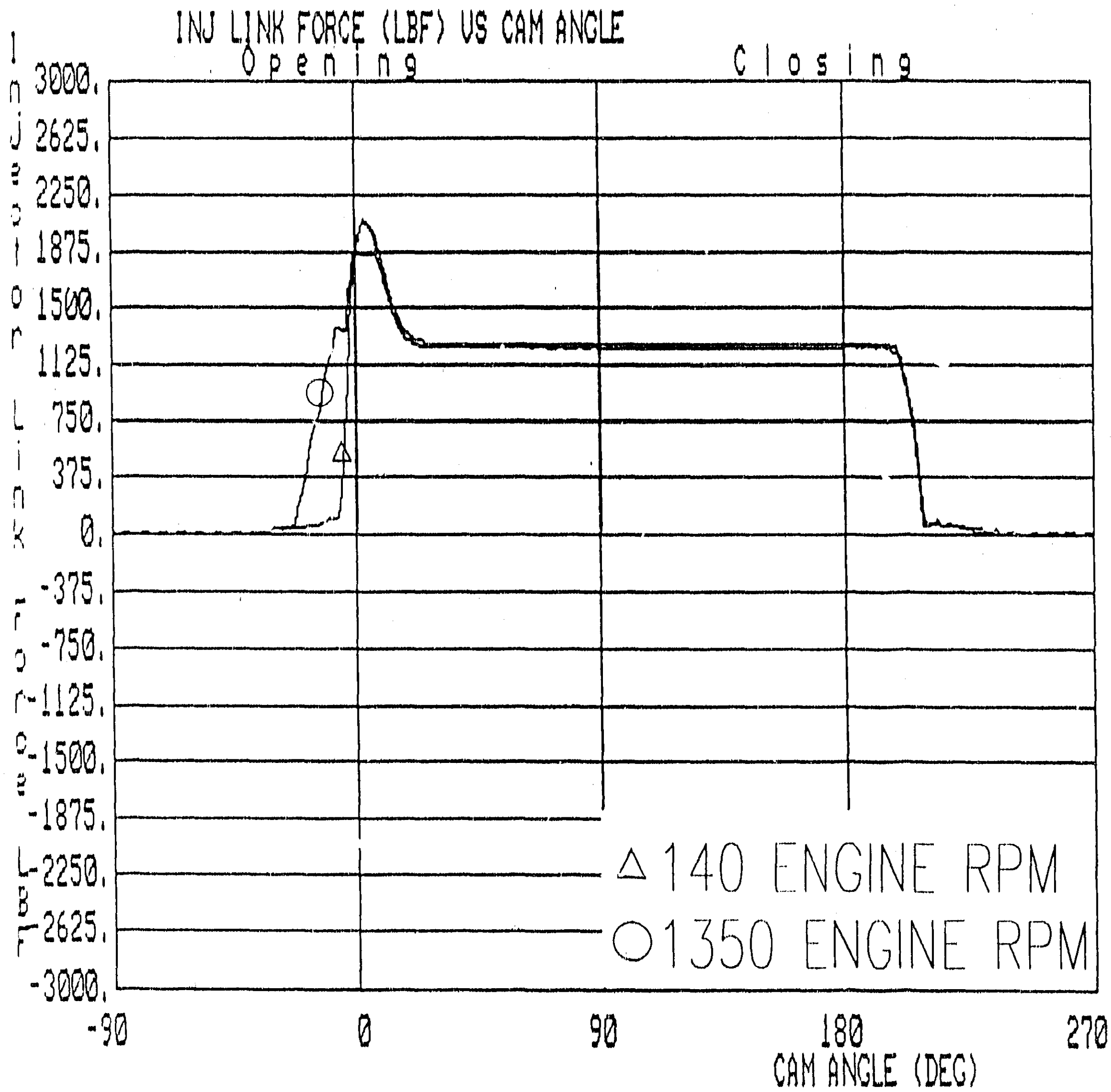



Figure 11: Injector Link Force

Injector Link Force

As was expected, the injector link force curves (Figure 11) were very similar to those at the pushrod. The magnitude of the force on the outer base circle is roughly 67% that of the pushrod's. This arises from the effective rocker arm ratio. However, at the peak, the ratio between the forces dropped to 61%, indicating that there was a significant combination of inertia and friction in the rocker arm area. Since the high and low speed peaks are virtually the same, it would appear that most of the difference is due to friction. If the injector link work were calculated (presently the processing program does not do this), subtracting it from the pushrod work would leave the amount of friction work lost in the upper injector train: rocker pivot plus injector and pushrod ends.

Note also how much the bouncing seen at the pushrod has been damped by the rocker arm - it is barely visible in the high speed injector link force plot.

Roller Friction Torque

As shown in Figure 12, roller friction torque is also very dependent on the engine speed. The low speed curve exhibits a sharp rise as the roller travels up the ramp of the cam. This is a result of the increasing force on the roller, plus the fact that the step in the cam initially exerts a force not acting on a line through the axis of the roller and the cam, increasing the torque on both. An even sharper rise is seen on the high speed curve as the roller contacts the beginning of the ramp. This arised from increased inertia. After the initial contact the torque drops slightly, then recovers as the roller experiences the longest "lever arm" of the cam.

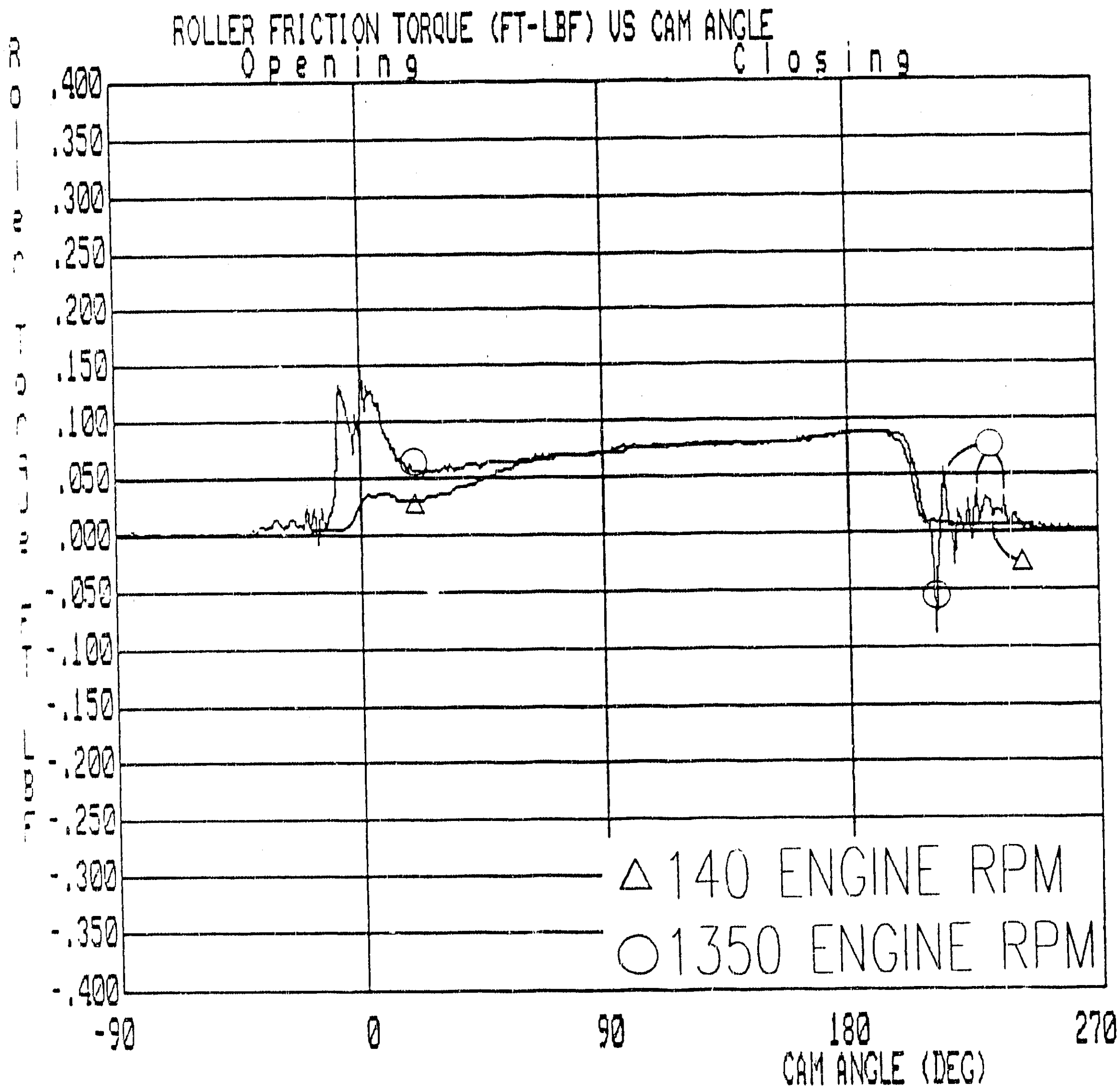



Figure 12: Injector Roller Pin Torque

Beyond the first peaks, both curves drop as the outer base circle is reached. In fact, the high speed curve drops substantially because (it is believed) the roller floats off of the cam slightly. As it rides on the outer base circle the spinning roller gradually matches the surface speed of the cam, increasing the friction torque as it does so. The low speed curve is similar, after roughly 70° of cam angle. The gradual rise in friction torque along the outer base circle is attributed to the oil film decreasing in thickness between the cam and roller, and the roller and its pin. It appears that elastohydrodynamic lubrication does not fully develop.

Returning to the base circle, the torque on the low and high speed rollers drops with the load. The high speed curve again shows signs of the roller bouncing on the cam, based on the corresponding spikes in the friction torque.

Roller Tractive Force

The roller tractive force curves (Figure 13) are based on the differentiation of measured data, and as such the high speed curve displays many jumps. Since at low speeds the dynamic effects are minimized, the 140 rpm curve almost duplicates that of the roller friction torque.

The high speed curve shows the same general trends seen in the friction torque plot; it has the double rise from contacting the ramp, a tractive force of zero as it floats past it, a gradual rise as traction increases on the outer base circle, and a force of zero as it floats and bounces on its return to the base circle.

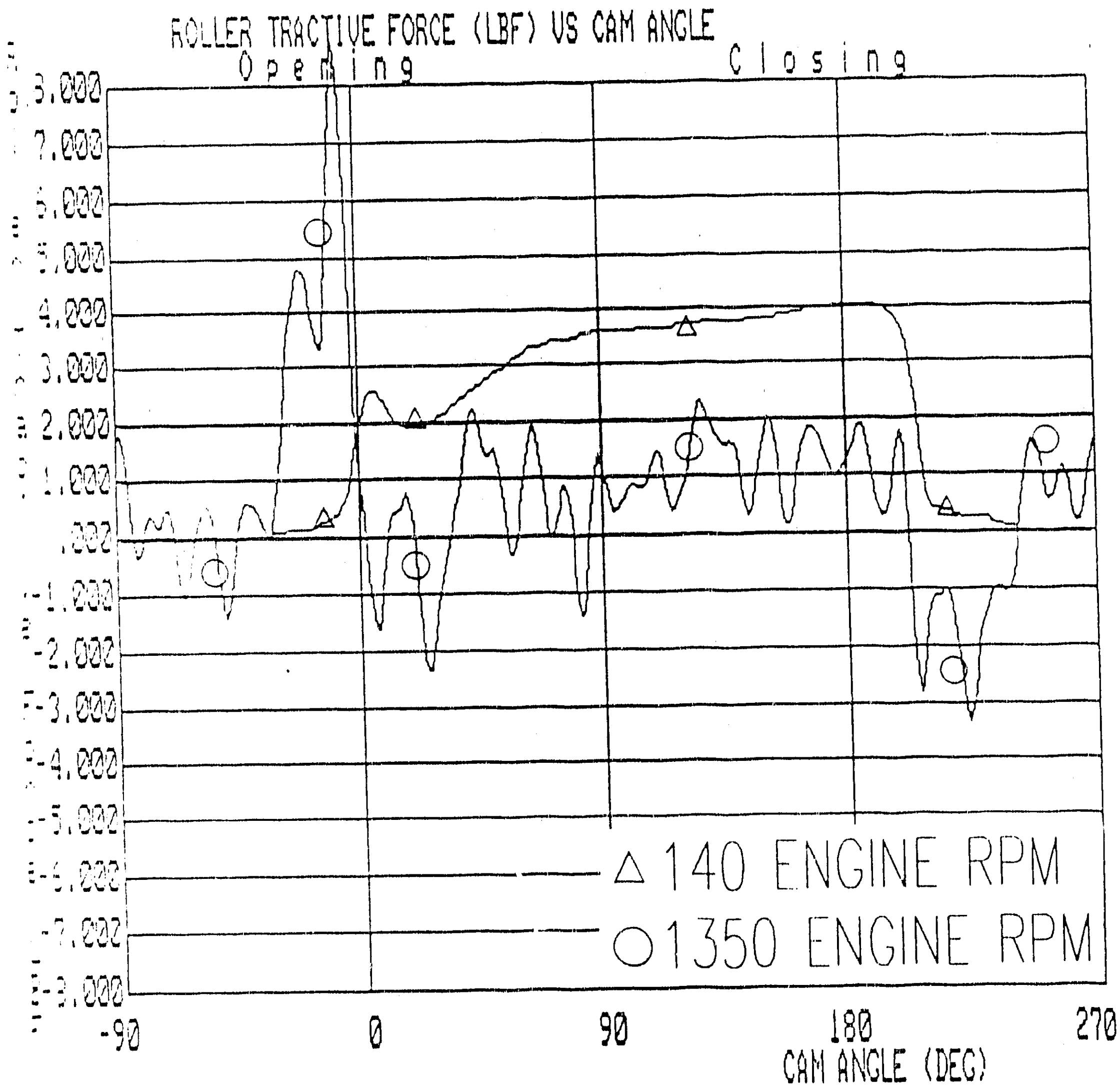



Figure 13: Injector Roller Tractive Force

Roller Rotation Relative to Cam Angle

Figure 14 compares the rotation of the roller relative to the rotation of the cam for both speeds. When the roller travels up the ramp, its higher rotational inertia at the higher engine speed causes it to lag behind the increasing rotation seen at the lower speed. At the peak the curves almost match, but afterwards the high speed roller floats and its rotation does not decrease as quickly as it does for the low speed case, with no slip. Eventually the curves coincide again, until the roller drops back to the base circle, where the high speed roller bounces, losing traction and lagging behind the low speed curve once more.

Pushrod Work

The pushrod work curves, Figure 15, very clearly show the extra work done at high speeds to compress the fuel. However, the work to compress the fuel is quite small compared to the work done on the injector train components as the injector bottoms against its cup. As the negative portion of the curve shows, some of this is returned to the system, with the remainder being lost as friction.

The bouncing mentioned previously also shows up as a series of positive and negative spikes at about 210° of cam angle, just after the large negative loop signifying the work done as the roller returns to the base circle. The low speed curve shows no bouncing, instead settling down immediately to the zero axis.

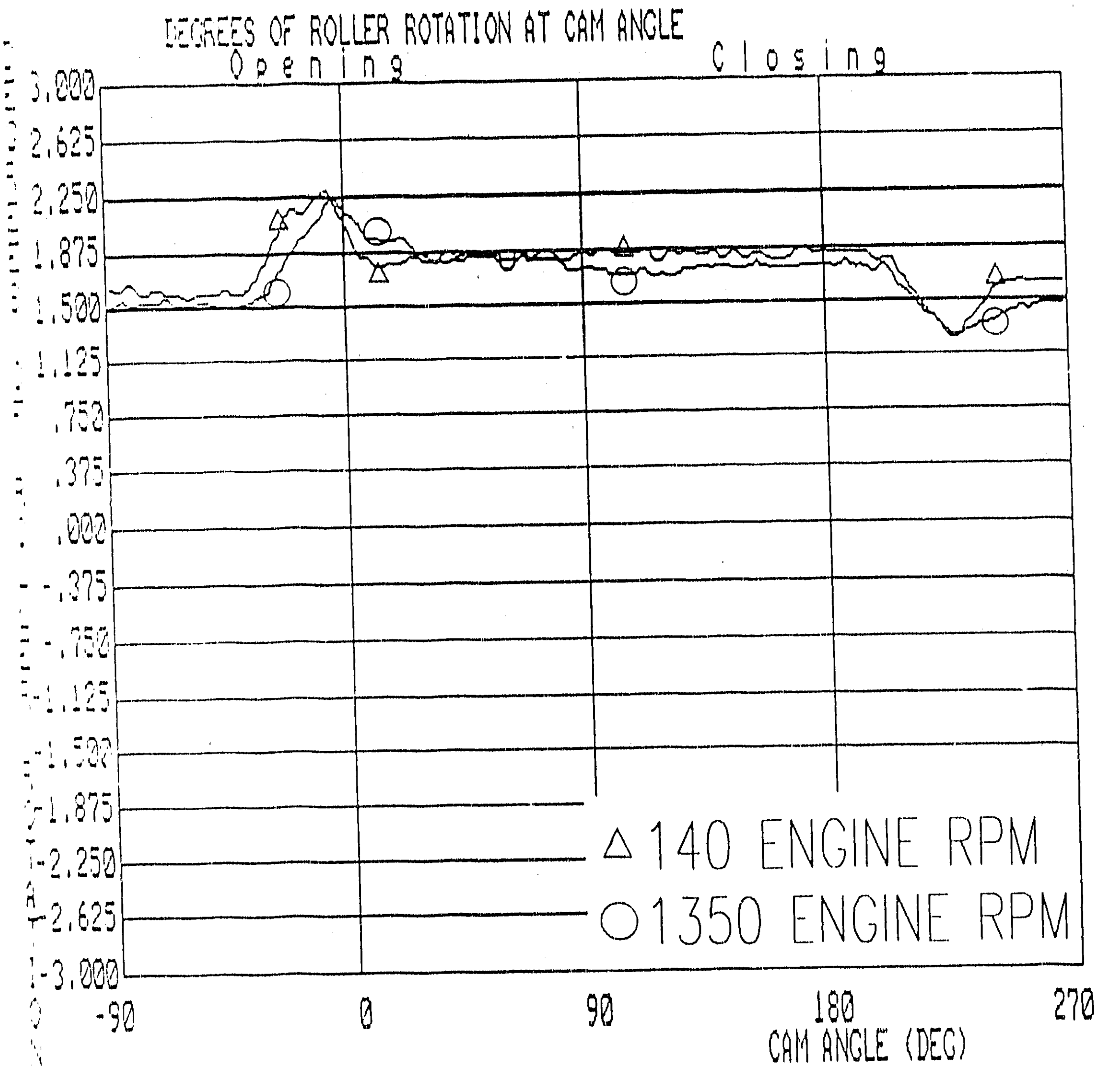



Figure 14: Roller Rotation Relative to Cam Rotation

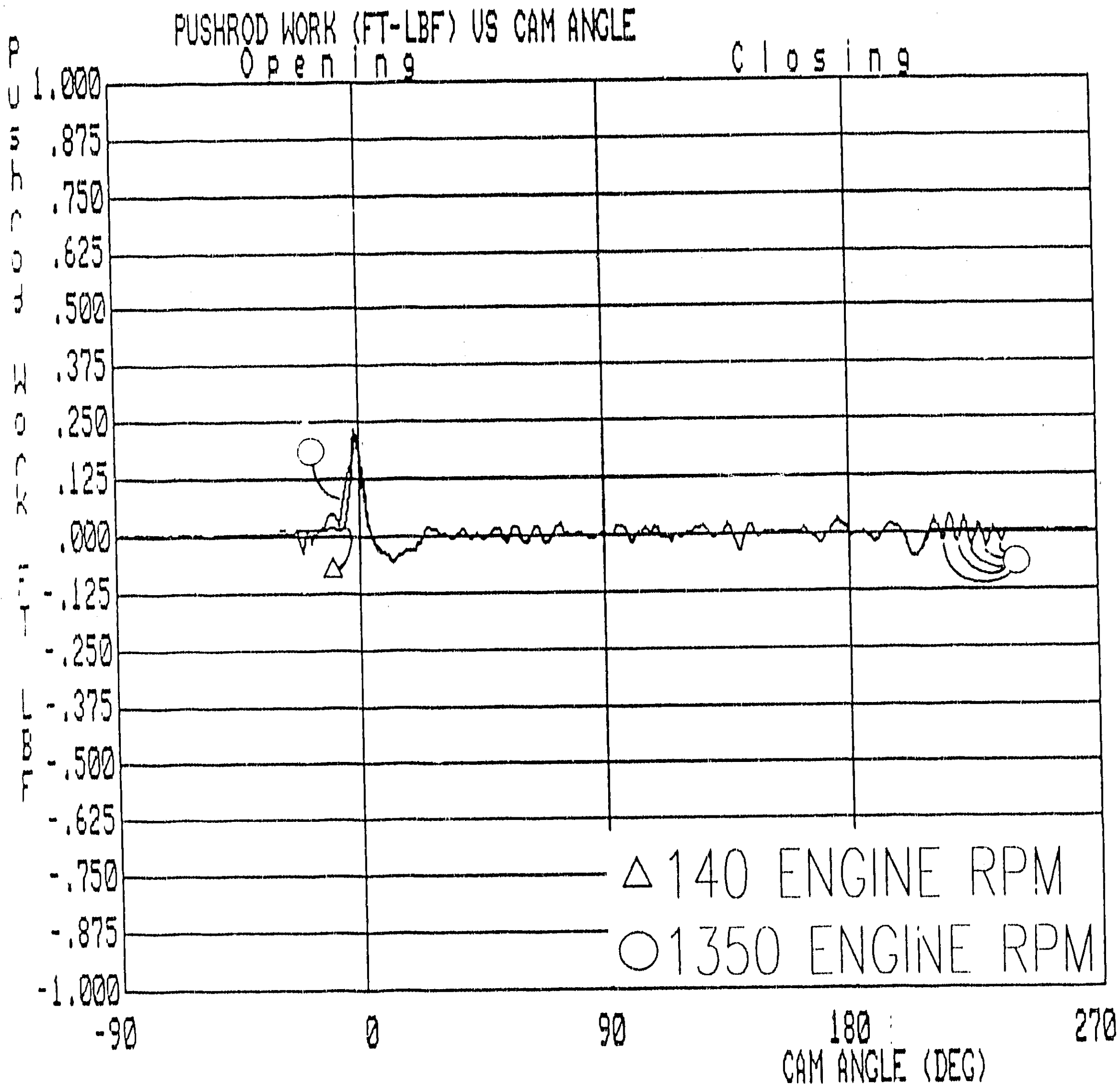



Figure 15: Injector Pushrod Work

CONCLUSION

The experimental techniques developed in the course of this project have proven to be very reliable. The wires from the pushrod, injector link and roller follower have never fatigued in over one hundred runs, nor have any of the modified components broken. Both the data acquisition and processing systems are very efficient, complete and user friendly.

The results are very repeatable and consistent. Thus far, the examination of the injector train has revealed potential wear problems in the floating of the roller on the outer base circle at high speeds, as well as its bouncing when returning to the base circle. It was found that the highest forces occured during injection, with peak loads almost 80% greater than on the outer base circle. Therefore, any increases in injection pressure must be accompanied by redesigned components. The effect of more gradual ramps up and down the cam would reduce most of the shocks to the system. Adequate lubrication appears to exist at idle speed and above,limiting friction to reasonable levels.

Future efforts will include development of the computer-based dynamic model. Tests will be run with the existing equipment using ceramic components. It is expected that the dynamic model will help us better understand the experimental data.

## INTRODUCTION TO ANALYTICAL REPORT

Increasing restrictions on exhaust emissions and competitive pressures for fuel economy are expected to require higher injection pressures and shorter injection duration. In turn this is expected to create problems of wear and friction in the injector train.

The computer simulation of the injection system provides the means to evaluate new designs and to predict the cause of problems. Our computer simulation will be verified by comparision with experimental data.

The kinematic and dynamic models of the Cummins L-10 injector train were developed to predict the forces and friction of each part, especially between the cam and roller follower. A general study of existing papers related to injector trains showed that a high speed dynamic model which did not include friction, was not sufficiently accurate to successfully predict the behavior of injector train. From our experimental data, it has been concluded that the lubrication is mainly mixed rather than elastohydrodynamic.

The following sections describe the mathematical model of kinematic and dynamic analysis accomplished to date.

MODELING

Kinematic model

The injector cam profile and the static displacement of the push rod were determined analytically based on Cummins' camshaft manufacturing data. This is needed to predict slippage between the roller follower and camshaft, and the effect of the cam profile on the frictional work. Equations describing the kinematics of the injector train have been developed by using the Finite Difference Method as follows.

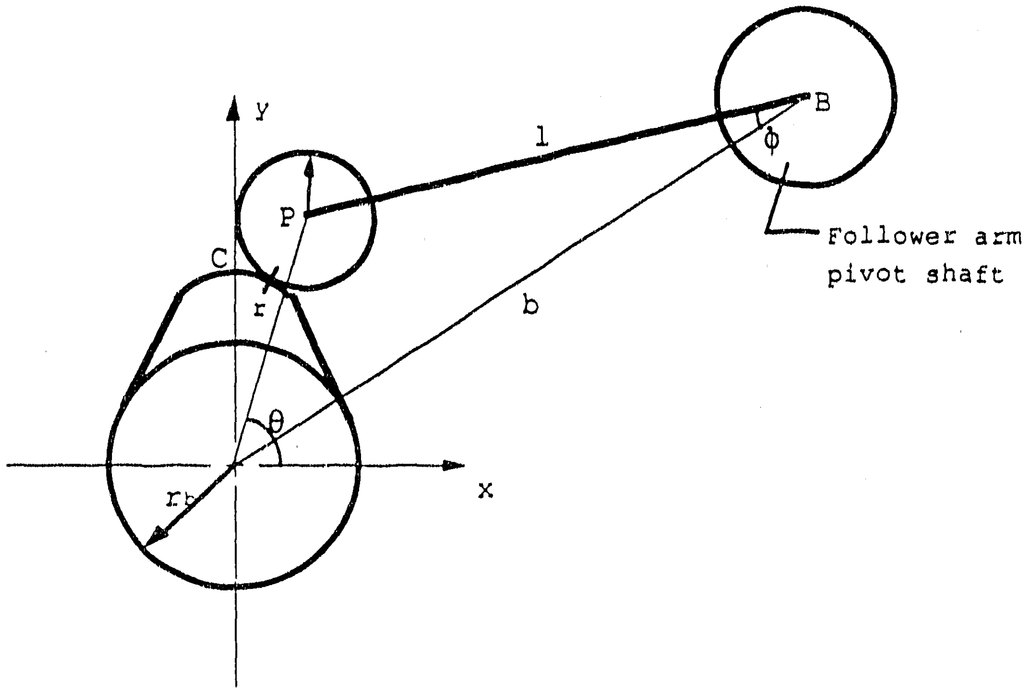



Fig.16 Cam And follower schematics

Instantaneous contact point C(x,y) can be obtained by solving the equations below.

$$F(x,y,\theta) = (x - r\cos\theta)^2 + (y - r\sin\theta)^2 - r_f^2 = 0 \qquad -(1)$$

$$\frac{\partial F}{\partial \theta} = 0 \qquad -(2)$$

from (1)and (2),

$$x = r\cos\theta \pm \frac{r_f}{\sqrt{1 + \frac{A^2}{B^2}}}$$

$$Y = \frac{Ax + r\frac{ds}{d\theta}}{B}$$

where, $A(\theta) = r\sin\theta - \frac{ds}{d\theta}\cos\theta$

$$B(\theta) = r\cos\theta + \frac{ds}{d\theta}\sin\theta$$

r(θ): Camshaft manufacturing data (given)

F(x,y,θ): General equation representing the center of moving roller follower.

Instantaneous contact point C(x,y) was calculated from the above by FDM and plotted in Fig.17.

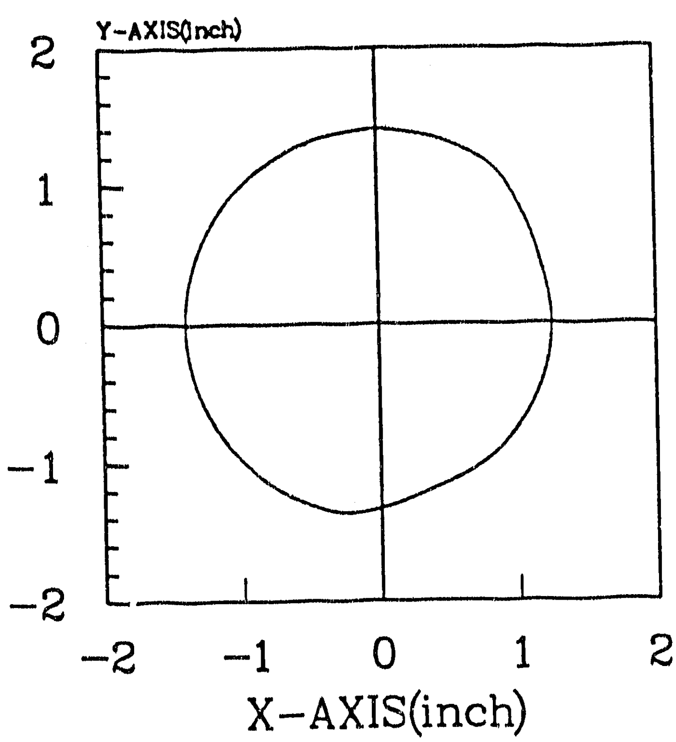



Fig.17 Cam profile

Follower arm rotation angle $\phi$ was computed as shown below.

$$r^2 = l^2 + b^2 - 2lb\cos\phi$$

$$\phi = \cos^{-1}\left(\frac{l^2 + b^2 - r^2}{2lb}\right)$$

Injector displacement was obtained with this value by assuming small motion ,and is plotted in Fig.18.

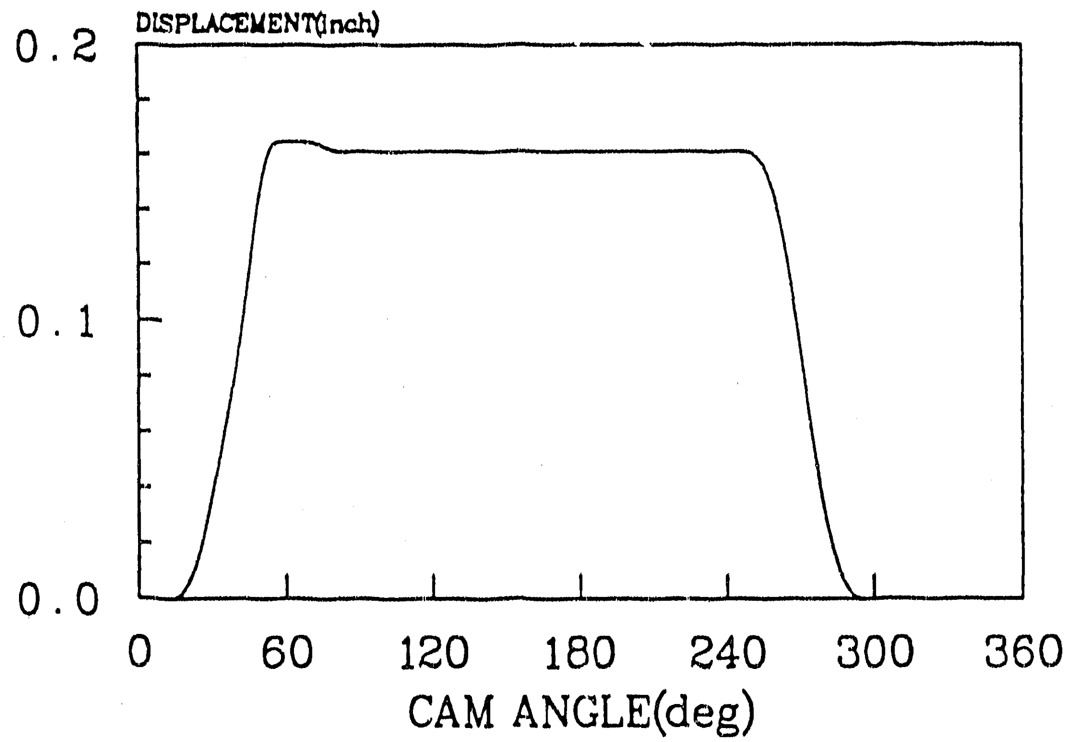
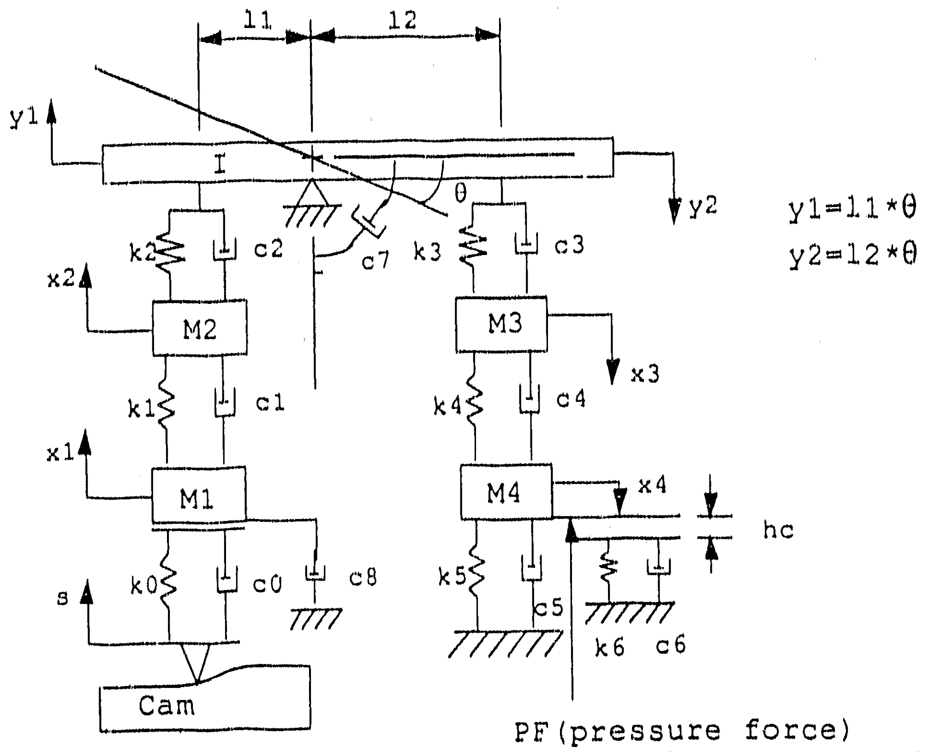



Fig.18. Injector displacement

Dynamic model

A schematic mathematical model of the L-10 unit injector system is outlined in Fig.19. This model consists of 5 main masses which represents 5 degrees of freedom. This dynamic linkage simulation has the capability of predicting the jump of each component. Input to the model( displacement determined by cam profile) was obtained by kinematical analysis with manufacturing data. Thus, the dynamic effects of the injector train's motion can be investigated by comparing kinematic motion and dynamic motion. The stiffness coefficient between each component will be determined theoretically on the basis of the Hertzian contact model.




| | |
|---|---|
| M1 | Equivalent mass of follower |
| M2 | Mass of pushrod |
| M3 | Mass of injector rod |
| M4 | Mass of plunger |
| I | Mass moment inertia of rocker arm |

| Spring Constant | Damping Coeff. | Position |
|---|---|---|
| k0 | c0 | Between cam and follower |
| k1 | c1 | Between follower and pushrod |
| k2 | c2 | Between pushrod and rocker arm |
| k3 | c3 | Between rocker arm and injector link |
| k4 | c4 | Between injector link and plunger |
| k5 | c5 | Plunger spring |
| k6 | c6 | Between Plunger and cup |
| | c7 | Between rocker arm shaft and rocker arm (rotational) |
| | c8 | External damping |

Fig.19 Schematic of dynamic model

Equations describing the motion of the injector train are derived below.
for mass M1:

(a) if jump does not occur

$$M_1\ddot{x}_1 + (c_0 + c_1 + c_8)\dot{x}_1 + (k_0 + k_1)x_1 - c_1\dot{x}_2 - k_1x_2 = c_0\dot{s} + k_0s \qquad -(1)$$

(b) if jump occurs

$$M_1\ddot{x}_1 + (c_1 + c_8)\dot{x}_1 + k_1x_1 - c_1\dot{x}_2 - k_1x_2 = 0 \qquad -(1')$$

for mass M2:

$$M_2\ddot{x} + (c_1 + c_2)\dot{x}_2 + (k_1 + k_2)x_2 + c_1\dot{x}_1 - k_1x_1 - c_2l_1\dot{\theta} - k_2l_1\theta = 0 \qquad -(2)$$

for mass moment of inertia I:

$$I\ddot{\theta} + (c_2l_1^2 + c_3l_2^2)\dot{\theta} + (k_2l_1^2 + k_3l_2^2)\theta - c_2l_1\dot{x}_2 - k_2l_1x_2 - c_3l_2\dot{x}_3 - k_3l_2x_3 = 0 \qquad -(3)$$

for mass of M3:

$$M_3\ddot{x}_3 + (c_3 + c_4)\dot{x}_3 + (k_3 + k_4)x_3 - c_3l_2\dot{\theta} - k_3l_2\theta - c_4\dot{x}_4 - k_4x_4 = 0 \qquad -(4)$$

for mass of M4:

(a) if $x_4 < h_c$

$$M_4\ddot{x}_4 + (c_4 + c_5)\dot{x}_4 + (k_4 + k_5)x_4 - c_4\dot{x}_3 - k_4x_3 = -PF \qquad -(5)$$

(b) if $x_4 \geq h_c$

$$M_4\ddot{x}_4 + (c_4 + c_5 + c_6)\dot{x}_4 + (k_4 + k_5 + k_6)x_4 - c_4\dot{x}_3 - k_4x_3 = -PF \qquad -(5')$$

where $h_s$ = solid fuel height

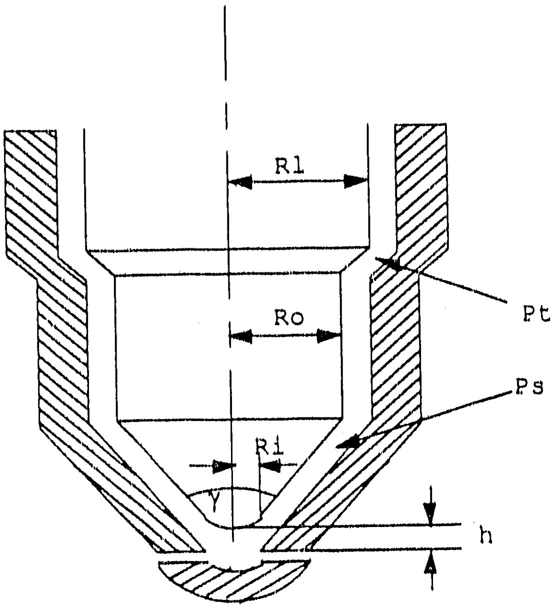
$h_c$ = initial separation between plunger and cup

and

$PF = 0$ when $x_4 < h_s$

$PF = PF(x_4)$ when $x_4 \geq h_s$

The model and equations of simulation to describe the throttling effects of the injector were made. This model approximates the throttling effects by representing the fuel flow from the injector to the cylinder as two orifices in series.




Pt: trapped pressure
Ps: sac pressure
Pc: cylinder pressure
PF: pressure force in simulation
At: π(R1^2 - Ro^2)
As: πRo^2
h: variable separation between plunger and cup
Ac: flow area around cup
A: total area of nozzle hole

Fig.20 Injector nozzle schematics

PF=Pt*At + Ps*As

Pt and Ps will be computed as follows.

$$Q = Cd_c A_c \sqrt{\frac{2}{\rho}(P_t - P_s)} = CdA \sqrt{\frac{2}{\rho}(P_s - P_c)}$$

from this,

$$P_s = P_c + \frac{1}{1 + (\frac{CdA}{Cd_c A_c})^2}(P_t - P_c) \qquad -(1)$$

Q: Flow out of the injector

from the continuity equation,

net fuel flow in = nozzle flow + compressible flow + leakage flow

$$\pi R_l^2 \frac{du}{dt} = CdA \sqrt{\frac{2}{\rho}(P_t - P_c)\frac{1}{1+(\frac{CdA}{Cd_c A_c})^2}} + \frac{V_i - \pi R_l^2 u}{B_T}\frac{dP_t}{dt} + K_L P_t \qquad -(2)$$

$V_i$ :Initial injector volume

u :Plunger displacement

$B_T$:Isothermal bulk modulus

$K_L$:Leakage flow coefficient

$Cd$:Flow discharge coefficient

Trapped pressure, Pt can be calculated by (2) and will be substituted into (1) to obtain Ps.

## MODELING SUMMARY

The equations for the kinematical analysis of the injector train have been derived and confirmed by experimental measurements. It is expected that the injector motion prediction is feasible from computer simulation.

Regarding the mechanical vibration of the injector train at high injection pressures, a dynamic model with 5 degrees of freedom was made and a comparison with experimental data will be performed. The results of the simulation will be obtained after completing the following three steps.

1. Equations describing the throttling effects of the injector will be solved numerically to estimate the pressure force, PF.
2. Dynamic analysis of the injector train linkage will be performed with the force PF. Force and displacement of each part can be calculated at this time.
3. A detailed investigation of the frictional phenomena between the cam and follower will be made attainable by using the forces of each part obtained in steps 1. and 2.

## APPENDIX 1

## VALVEIN3 PROGRAM

```
$DEBUG
C     ********************************************************************
      PROGRAM VALVEIN3
C     ********************************************************************
C     *       This program is for getting test conditions,               *
C     *       calling the data acquisition routine and                   *
C     *       storing the data on disk for the L10 valve                 *
C     *       train friction study.  The program reads                   *
C     *       in and stores 4 channels of 3600 points.                   *
C     *                                                                  *
C     *       MUST BE LINKED WITH FOR2GRPH.OBJ AND VALADC.OBJ            *
C     ********************************************************************
      CHARACTER*14 STR,IOSTR,BLANKS,FSTR
      CHARACTER*1  ETYPE,ACTYPE,STRING(14)
      CHARACTER*10 DATE,TIM,Y1,ABRAND,ANO,LUBRIC
      CHARACTER*10 BUFSTR,FTYPE
      INTEGER*4    ICLOCK
      INTEGER*2    PRFORC(3600),ROTORQ(3600),CAMSPD(3600),ROLSPD(3600)
      INTEGER*2    INJECT(3600),STAT,NUMCON,CTLWRD
      EXTERNAL     ICLOCK
      EQUIVALENCE  (STR,STRING)
      COMMON/ADCBL/PRFORC,ROTORQ,CAMSPD,ROLSPD,INJECT,STAT,NUMCON,CTLWRD
      COMMON/CONST/PRODZ,PRODS,ROTRQZ,ROTRQS,ROLLI,FINJTZ,FINJTS,ENCODES
     &             ,ROLENS
C     --------------------- set default values -----------------------
      BLANKS='              '
      HOUR=0.0
      ISTART=1
C     WRITE(*,*) 'LIST OF ENGINE TYPES:'
C     WRITE(*,*) 'A = ADIABATIC CUMMINS ENGINE RM 249'
C     WRITE(*,*) 'F = CUMMINS ENGINE RM 245'
C     WRITE(*,*) 'K = CADILLAC ENGINE IMEP TECHNIQUE'
C     WRITE(*,*) 'L = CADILLAC ENGINE FIXED SLEEVE TECHNIQUE'
C     WRITE(*,*) 'V = VALVE TRAIN ENGINE'
      ETYPE='C'
C     --- ENGINE TYPE C CORRESPONDS TO CAM FRICTION TESTS ON CUMMINS L10
      WRITE(*,*) 'ENGINE TYPE=',ETYPE
C     WRITE(*,*) 'Enter engine type (A,F,K,L,OR V)'
C     READ(*,'(A1)') ETYPE
      WRITE(FSTR,'(A7,A1)') 'DEFAULT',ETYPE
      OPEN(1,FILE=FSTR,STATUS='OLD',FORM='FORMATTED')
      READ(1,* ) IFIRE,IRPM,VAC,IOILP,NCYCLE,NDPC,NSET,EHOUR
      READ(1,10) LUBRIC,ABRAND,ANO
 10   FORMAT(4A10)
      CLOSE(1)
C     ------------------------------------------------------------------
C     BELOW IS A LISTING OF WHAT THE DEFAULT VARIABLES STAND FOR
C     ------------------------------------------------------------------
C     IFIRE   =   1 FOR FIRING, 0 FOR MOTORING
C     IRPM    =   ENGINE RPM
C     VAC     =   MANIFOLD VACUUM
C     IOILP   =   OIL PRESSURE
C     NCYCLE  =   NUMBER OF ENGINE CYCLES OF DATA
C     NSET    =   NUMBER OF ENGINE SETS OF DATA
C     NDPC    =   NUMBER OF DEGREES (SAMPLES) PER CYCLE
```

```
C       EHOUR   =   ENGINE HOURS
C       LUBRIC  =   LUBRICANT TYPE
C       ABRAND  =   STRAIN GAUGE AMPLIFIER BRAND NAME
C       ANO     =   STRAIN GAUGE AMPLIFIER NUMBER
C       BUFSTR  =   DATA 6000 BUFFER FILE NAMES
C       ---------------------- READ CONSTANTS FILE ---------------------
        WRITE(FSTR,'(A5,A1)') 'CONST',ETYPE
        OPEN(1,FILE=FSTR,STATUS='OLD',FORM='FORMATTED')
        READ(1,'(A7)') FTYPE
        IF(FTYPE.EQ.'CONSTC1') THEN
        READ(1,* ) PRODZ,PRODS,ROTRQZ,ROTRQS,ROLLI,FINJTZ,FINJTS,ENCODES
        READ(1,* ) ROLENS
C       ---- CHECK CONSTANTS FILE FOR THE MEANING OF THESE CONSTANTS ----
        CLOSE(1)
        ELSE
        WRITE(*,*) 'CONSTANTS FILE IS INCOMPATIBLE'
        ENDIF
C       -------------- get the test conditions of the run ---------------
        ACTYPE='T'
        WRITE(*,*) 'ENGINE TYPE=',ETYPE,', ACQUISITION DEVICE=',ACTYPE
        WRITE(*,*) ' ENTER:                                     NOW SET AT'
        CALL GETSTR('       DATE (eg. "1-24-83")                 ',  DATE)
        CALL GETREL('       BAROMETRIC PRESS. (IN HG)            ',BARPRS)
        CALL GETSTR('       AMP. BRAND                           ',ABRAND)
        CALL GETSTR('       AMP. NUMBER                          ',   ANO)
        CALL GETSTR('       LUBRICANT SAE:                       ',LUBRIC)
C       ------------ Branch to individual acquisition device ------------
20      IF(ACTYPE.EQ.'T') CALL TECMAR  (NDPC,NSET,NCYCLE,YSCL)
30      CALL GETINT('       TEST NO.                             ', NTEST)
        CALL GETINT('       RUN NO.                              ',  NRUN)
        ITDRY=0
        ITWET=0
        I=IFIRE+1
        CALL GETINT('       FIRING=2 MOTORING=1                  ',     I)
        IFIRE=I-1
        CALL GETINT('       ENGINE RPM                           ',  IRPM)
        CALL GETREL('       MANIFOLD VAC. (IN HG)                ',   VAC)
        CALL GETINT('       OIL PRESSURE (PSI)                   ', IOILP)
        CALL GETINT('       OIL FLOW (CFM?)                      ', IOILF)
        CALL GETREL('       DYNO READING (LBF)                   ',TORQUE)
C       --------- skip these questions if it is a motoring test ----------
        IF(IFIRE.EQ.1) THEN
        CALL GETINT('       SPARK ADVANCE (DEG BTDC)             ',ISPKAD)
        CALL GETREL('       FUEL WEIGHT (LBF)                    ',FUELWT)
        CALL GETREL('       ELAPSED TIME (MIN)                   ',ELPTIM)
        CALL GETINT('       EXHAUST TEMP F                       ', ITEXH)
        END IF
        CALL GETINT('       WATER TEMP F                         ', ITWAT)
        CALL GETINT('       OIL SUMP TEMP F                      ',ITOILS)
        CALL GETINT('       OIL GALLERY TEMP F                   ',ITOILG)
        CALL GETINT('       STRAIN GUAGE TEMP F                  ',  ITSG)
        CALL GETINT('       INLET AIR TEMP F                     ', ITAIR)
        AINP=(BARPRS-VAC)/2.03603
        CLOCK=ICLOCK(K)/65454.54
        IF(ISTART.EQ.1) THEN
```

```
      ISTART=0
      OCLOCK=CLOCK
      END IF
      HOUR=HOUR+CLOCK-OCLOCK
      IF(HOUR.GE.13.0) HOUR=HOUR-12.0
      CALL GETTIME('       TIME (e.g. 10:34)                 ',HOUR,TIM)
      EHOUR=EHOUR+CLOCK-OCLOCK
      OCLOCK=CLOCK
      CALL GETTIME('       ENGINE HOURS (e.g.145:05)         ',EHOUR,Y1)
      I=1
      CALL GETINT('ENTER  ( 1 = SAVE,     2 = RE-ENTER )   ',         I)
      IF(I.EQ.2) GOTO 30
C     --- save new default file
      WRITE(FSTR,'(A7,A1)') 'DEFAULT',ETYPE
      OPEN(1,FILE=FSTR,STATUS='NEW',FORM='FORMATTED')
      WRITE(1, *) IFIRE,IRPM,VAC,IOILP,NCYCLE,NDPC,NSET,EHOUR
      WRITE(1,10) LUBRIC,ABRAND,ANO
      CLOSE(1)
      WRITE(STR,'(I3,I2)') NTEST,NRUN
      DO 40 I=13,1,-1
40    IF(STRING(I).EQ.' '.AND.STRING(I+1).NE.' ') STRING(I)='0'
C     ------------ SAVE DATA ON DISK FOR FRICTION PROGRAM -------------
      WRITE(IOSTR,'(2A1,A5)') 'D',ETYPE,STR
      WRITE(*,*) 'ENTER DATA FILE NAME:',IOSTR
      READ(*,'(A14)') STR
      IF(STR.NE.BLANKS) IOSTR=STR
      WRITE(STR,'(A2,A12)') 'B:',IOSTR
      OPEN(1,FILE=STR,STATUS='NEW',FORM='BINARY')
      K=NCYCLE*NDPC*NSET
      WRITE(*,*) 'STORING DATA ON THE DISK...'
      WRITE(1) 'CVALVE-3'
      WRITE(1)  K
      WRITE(1) (PRFORC(J),J=1,K)
      WRITE(1) (ROTORQ(J),J=1,K)
      WRITE(1) (INJECT(J),J=1,K)
      WRITE(1) (CAMSPD(J),J=1,K)
      WRITE(1) (ROLSPD(J),J=1,K)
      WRITE(1) NTEST,NRUN,ITDRY,ETYPE,NDPC,
     &         ITWET,IRPM,IOILP,IOILF,ISPKAD,ITEXH,
     &         ITWAT,ITOILS,ITOILG,ITSG,ITAIR,AINP,IFIRE,EHOUR,
     &         BARPRS,VAC,TORQUE,FUELWT,ELPTIM,YSCL,
     &         DATE,TIM,LUBRIC,ABRAND,ANO
      CLOSE(1)
      IF(FTYPE.EQ.'CONSTC1') THEN
      WRITE(FSTR,'(A1,A13)') 'C',IOSTR
      WRITE(STR,'(A2,A12)') 'B:',FSTR
      OPEN(1,FILE=STR,STATUS='NEW',FORM='FORMATTED')
      WRITE(1,*) FTYPE
      WRITE(1,*) PRODZ,PRODS,ROTRQZ,ROTRQS,ROLLI,
     &           FINJTZ,FINJTS,ENCODS,ROENCS
      ELSE
      WRITE(*,*) 'COPY CORRECT CONSTANTS FILE TO DATA DISK'
      END IF
      NRUN=NRUN+1
      I=1
```

```
      CALL GETINT('ENTER 2=EXIT 1=TAKE DATA                  ',I)
      IF(I.EQ.1) GOTO 20
      STOP
      END
C     ******************************************************************
      SUBROUTINE TECMAR(NDPC,NSET,NCYCLE,YSCL)
C     ******************************************************************
C     -        This subroutine is for taking data from the tecmar      -
C     -        a/d boards by calling an assembly language routine.     -
C     ------------------------------------------------------------------
      INTEGER*2 PRFORC(3600),ROTORQ(3600),CAMSPD(3600),ROLSPD(3600)
      INTEGER*2 INJECT(3600),STAT,NUMCON,CTLWRD
      COMMON/ADCBL/PRFORC,ROTORQ,CAMSPD,ROLSPD,INJECT,STAT,NUMCON,CTLWRD
      EXTERNAL KEYIN
      STAT=0
      YSCL=2048.0/5.0
C     -------------------- get data and display it -------------------
      WRITE(*,*) 'HIT RETURN TO START AS WELL AS STOP TAKING DATA'
      READ(*,'(I1)') I
      CALL GMODE
      CALL CLRSCR
10    CTLWRD=04
      NUMCON=NCYCLE*NDPC
      CALL VALADC
      CALL PLT(NDPC,NSET,NCYCLE,YSCL)
      IKEY=KEYIN(IDUMMY)
      IF(IKEY.NE.13) GOTO 10
      CALL TMODE
      IF(STAT.GE.64) WRITE(*,*) '**** D A T A  O V E R R U N ****',STAT
      RETURN
      END
C     ******************************************************************
      SUBROUTINE GETTIME(STRING,HOUR,TI)
C     ******************************************************************
C     -           This subroutine gets hours and engine hours          -
C     ------------------------------------------------------------------
      CHARACTER*35 STRING
      CHARACTER*10 TIM,STR,TI
      CHARACTER*1  TIME(10)
      EQUIVALENCE  (TIM,TIME)
      IHOUR=INT(HOUR)
      IMIN=60*(HOUR-IHOUR)
      WRITE(TIM,'(I4,A1,I2)') IHOUR,':',IMIN
      IF(IMIN.LT.10) TIME(6)='0'
      WRITE(*,*) STRING,TIM
      READ(*,'(A10)') STR
      IF(STR.NE. '          ') THEN
      TIM=STR
      DO 10 I=1,10
10    IF(TIME(I).EQ.':') WRITE(STR,'(A2,I1,A6)') '(I',I-1,',X,I2)'
      READ(TIM,STR) IHOUR,IMIN
      HOUR=IHOUR+IMIN/60.0
      END IF
      TI=TIM
      RETURN
```

```
      END
C     ****************************************************************
      SUBROUTINE GETSTR(QUESTION,ANSWER)
C     ****************************************************************
C     -      This subroutine reads in a string from the keyboard       -
C     ----------------------------------------------------------------
      CHARACTER*40 QUESTION
      CHARACTER*10 ANSWER,STR
10    WRITE(*,'(A1,A40,A10)') ' ',QUESTION,ANSWER
      READ(*,'(A10)',ERR=20) STR
      IF(STR.NE.'          ') ANSWER=STR
      RETURN
20    WRITE(*,*) '*** Input error.  Please re-enter data ***'
      GOTO 10
      END
C     ****************************************************************
      SUBROUTINE GETREL(QUESTION,ANSWER)
C     ****************************************************************
C     ---     This subroutine reads a real number from keyboard      ---
C     ----------------------------------------------------------------
      CHARACTER*40 QUESTION
      CHARACTER*10 STR
      CHARACTER*1  STRING(10)
      EQUIVALENCE  (STR,STRING)
10    WRITE(*,'(A1,A40,F16.8)') ' ',QUESTION,ANSWER
      READ(*,'(A10)',ERR=40) STR
      J=0
      DO 20 I=1,10
      IF(STRING(I).EQ.'.') GOTO 30
20    IF(STRING(I).NE.' ') J=I
      IF(J.NE.0) STRING(J+1)='.'
30    IF(STR.NE.'          ') READ(STR,'(BN,F10.4)',ERR=40) ANSWER
      RETURN
40    WRITE(*,*) '*** Input error. Please re-enter data ***'
      GOTO 10
      END
C     ****************************************************************
      SUBROUTINE GETINT(QUESTION,IANSWER)
C     ****************************************************************
C     ---   This subroutine reads in an integer from the keyboard    ---
C     ----------------------------------------------------------------
      CHARACTER*40 QUESTION
      CHARACTER*9 STR
10    WRITE(*,*) QUESTION,IANSWER
      READ(*,'(A9)',ERR=20) STR
      IF(STR.NE.'         ') READ(STR,'(BN,I9)',ERR=20) IANSWER
      RETURN
20    WRITE(*,*) '*** Input error.  Please re-enter data ***'
      GOTO 10
      END
C     ****************************************************************
      SUBROUTINE GETYN(QUESTION,IANSWER)
C     ****************************************************************
C     ---  THIS SUBROUTINE READS IN THE ANSWER TO A YES/NO QUESTION  ---
C     ----------------------------------------------------------------
```

```
      CHARACTER*40 QUESTION
      CHARACTER*1 YN
      YN='Y'
      IF(IANSWER.EQ.0) YN='N'
10    WRITE(*,*) QUESTION,YN
      READ(*,'(A1)',ERR=20) YN
      IF(YN.EQ.'Y'.OR.YN.EQ.'y') IANSWER=1
      IF(YN.EQ.'N'.OR.YN.EQ.'n') IANSWER=0
      IF(IANSWER.NE.2) RETURN
20    WRITE(*,*) '*** Input error.  Please re-enter data ***'
      GOTO 10
      END
C     *****************************************************************
      SUBROUTINE PLT(NDPC,NSET,NCYCLE,YSCL)
C     *****************************************************************
C     - This subroutine is for ploting data from the tecmar a/d boards -
C     -----------------------------------------------------------------
      INTEGER*2 PRFORC(3600),ROTORQ(3600),CAMSPD(3600),ROLSPD(3600)
      INTEGER*2 INJECT(3600),STAT,NUMCON,CTLWRD
      CHARACTER*8 TITLE(5)
      COMMON/ADCBL/PRFORC,ROTORQ,CAMSPD,ROLSPD,INJECT,STAT,NUMCON,CTLWRD
      COMMON/CONST/PRODZ,PRODS,ROTRQZ,ROTRQS,ROLLI,FINJTZ,FINJTS,ENCODES
     &            ,ROLENS
      TITLE(1)='PR FORCE'
      TITLE(2)='ROL TORQ'
      TITLE(3)='INJ LNK '
      TITLE(4)='CAM RPM '
      TITLE(5)='ROL RPM '
      CALL CLRSCR
      CALL TEXTF(150,15,40,'CUMMINS L-10 VALVE TRAIN FRICTION STUDY ')
C     OPEN(3,FILE='LPT1')
      DO 1000 I=1,5
C     ------------------------- draw axies ---------------------------
      CALL PUTPT( 96,55*I-25)
      CALL DLINE(100,55*I-25)
      CALL DLINE(100,55*I+25)
      CALL DLINE(700,55*I+25)
      CALL TEXTF(1,55*I+5,8,TITLE(I))
      IF(I.EQ.1) THEN
      CALL TEXTF(50,55*I-25,4,' 5 V')
      ELSE IF(I.EQ.2) THEN
      CALL TEXTF(50,55*I-25,4,' 5 V')
      ELSE IF(I.EQ.3) THEN
      CALL TEXTF(50,55*I-25,4,' 5 V')
      ELSE
      CALL TEXTF(50,55*I-25,4,'1500')
      END IF
C     ------------------------- draw lines ---------------------------
      DO 100 J=1,1800
      IF(I.EQ.1) THEN
      IY=-PRFORC(J)/YSCL/5.0*25.0+I*55
      ELSE IF(I.EQ.2) THEN
      IY=-ROTORQ(J)/YSCL/5.0*25.0+I*55
      ELSE IF(I.EQ.3) THEN
      IY=-INJECT(J)/YSCL/5.0*25.0+I*55
```

```
      ELSE IF(I.EQ.4) THEN
      K=CAMSPD(J)
      IF(K.LE.0) K=K+65536
      IY=-1000000.0/K*60.0/720.0/1500.0*70.0+I*55.0+25.0
      ELSE
      K=ROLSPD(J)
      IF(K.LE.0) K=K+65536
C     WRITE(3,*) 'K=',K,ROLSPD(J)
      IY=-1000000.0/K*60.0/120.0/1500.0*70.0+I*55.0+25.0
      END IF
      IX=100+J/1800.0*600.0
      CALL PUTPT(IX,IY)
      CALL DLINE(IX,IY)
100   CONTINUE
C     ------------------------- end of plot ---------------------------
1000  CONTINUE
      RETURN
      END
```

# APPENDIX 2

# FLIFTER5 PROGRAM

```
$DEBUG
      PROGRAM FLIFTER5
C
C ****************************************************
C * This program is for taking data from the disk,    *
C * calculating valve train friction, and plotting.   *
C ****************************************************
C
      CHARACTER*14 FILSTR,IOSTR,STR,CFILE
      CHARACTER*1 ETYPE
      CHARACTER*10 ZTYPE
      CHARACTER*8 FTYPE,FNAME
      CHARACTER*60 COMMENT,TITLE,HEADING,HEAD
      CHARACTER*70 STR70
      CHARACTER*10 DATE,TIME,TBRAND,TNO,ABRAND,ANO,LUBRIC,STRING
      INTEGER*2 PRFORC(3600),ROTORQ(3600),CAMSPD(3600),ROLLSP(3600)
      INTEGER*2 CAMAGL(2000),ITDC(20),LVDT(3600),ITDCD(20),ERTDC
      INTEGER*2 SHIFT,ANS,INJECT(3600),LVDTIJ(3600)
      INTEGER*4 ROLSPD(3600),REDSPD(2000)
      REAL*4 PRF(360),RTQ(360),CSP(360),RSP(360),TRQIN(360),ARMTH(360)
      REAL*4 DGMRK(119),EDGMRK(2000),ROLRPM(3600),ROLA(360),TRFRC(360)
      REAL*4 ARMVL(360),ARMAC(360),REV(360),IJF(360),IJL(360),IJV(360)

      REAL*4 PRL(360),LZ,LS,ZMAX,RTQZ1
      REAL*4 PRV(360),PRA(360),PRW(360)
      COMMON ROLRPM,ROLA,TRFRC
      COMMON REDSPD,CAMAGL,LVDT,DGMRK,EDGMRK
      COMMON PRFORC,ROTORQ,CAMSPD,ROLLSP,ROLSPD
      COMMON PRF,PRL,PRV,PRA,PRW
      COMMON/PLOT/STRING
C   DGMRK=ROLLER ENCODER DEGREES PER MARK
C   ROLA=ROLLER ACCELERATION
C   PRF=PUSH ROD FORCE

C   PRL=PUSHROD LIFT
C   PRV=PUSHROD VELOCITY
C   PRA=PUSHROD ACCELERATION
C   TRQIN=ROLLER INERTIA TORQUE
C   TRFRC=ROLLER TRACTIVE FORCE
C   PRW=PUSH ROD WORK
C
C      ......This will read in the data file for calculations....
3      WRITE (*,5)
5      FORMAT( /,5X,' enter file name (e.g. DC12345),*** to exit: '$ )
       READ (*,8) IOSTR
8      FORMAT (A9)
       IF(IOSTR.EQ.'***')STOP
       STRING=IOSTR
       WRITE (FILSTR,'(A1,A1,A9)') 'B',':',IOSTR
       WRITE (*,*) 'Reading data from disk'
10     OPEN(1,FILE=FILSTR,STATUS='OLD',FORM='BINARY')
       READ (1) FTYPE
       IF (FTYPE.NE.'CVALVE-1'.AND.FTYPE.NE.'CVALVE-3') THEN
        WRITE (*,*) 'INCORRECT DATA FILE'
        STOP
```

```
      ENDIF
      READ (1) NPTS
      READ (1) (PRFORC(J) ,J=1,NPTS)
      READ (1) (ROTORQ(J) ,J=1,NPTS)
      IF(FTYPE.EQ.'CVALVE-3') READ(1) (INJECT(J),J=1,NPTS)
      READ (1) (CAMSPD(J) ,J=1,NPTS)
      READ (1) (ROLLSP(J) ,J=1,NPTS)
      READ(1) NTEST,NRUN,ITDRY,ETYPE,NDPC,
     1        ITWET,IRPM,IOILP,IOILF,ISPKAD,ITEXH,
     1        ITWAT,ITOILS,ITOILG,ITSG,ITAIR,AINP,IFIRE,EHOUR,
     1        BARPRS,VAC,TORQUE,FUELWT,ELPTIM,YSCL,
     1        DATE,TIME,LUBRIC,ABRAND,ANO
      CLOSE(1)
      WRITE(*,*)'NPTS=',NPTS
C.....................Read constants file.....................
      WRITE(*,*) 'Reading calibration constants'
      WRITE(FILSTR,'(A1,A13)')'C',IOSTR
      WRITE(STR,'(A2,A12)')'B:',FILSTR
      WRITE(*,*) STR
      OPEN(1,FILE=STR,STATUS='OLD',FORM='FORMATTED')
      WRITE(*,*) 'OPEN'
      READ (1,'(A7)') ZTYPE
      WRITE(*,*) ZTYPE
      IF (FTYPE.EQ.'CVALVE-3') THEN
       READ (1,*) PRZ,PRS,RTQZ,RTQS,RINER,LZ,LS,ENS,RENS,FINJTZ,FINJTS

       WRITE(*,*) PRZ,PRS,RTQZ,RTQS,RINER,LZ,LS,ENS,RENS,FINJTZ,FINJTS
      ELSE
       READ (1,*) PRZ,PRS,RTQZ,RTQS,RINER,LZ,LS,ENS,RENS
       WRITE(*,*) PRZ,PRS,RTQZ,RTQS,RINER,LZ,LS,ENS,RENS,FINJTZ,FINJTS
      ENDIF
C  RTQZ=ROLLER TORQUE ZERO TORQUE (FT-Lb)
C  RTQS=ROLLER TORQUE SLOPE (VOLTS/FT-LB)
C  PRS=PUSHROD STRAIN GAUGE SLOPE (VOLTS/Lb)
C  PRZ=PUSHROD ZERO FORCE VOLTAGE
C  RINER=ROLLER INERTIA (SLUG-FT**2)
C  LZ=LVDT ZERO (FT)
C  LS=LVDT LIFT SLOPE (VOLTS/FT)
C  FINJTS=INJ LINK FORCE SLOPE (VOLTS/LB)
C  FINJTZ=INJ LINK ZERO FORCE VOLTAGE
C
C             Input lift file
C

      WRITE (*,15)
15     FORMAT( /,5X,' enter LVDT file name (e.g. LVDTRD): '$ )
      READ (*,18) IOSTR
18     FORMAT (A9)
      STRING=IOSTR
      WRITE (FILSTR,'(A1,A1,A9)') 'B',':',IOSTR
      WRITE (*,*) 'Reading data from disk'
20    OPEN(1,FILE=FILSTR,STATUS='OLD',FORM='BINARY')
      READ (1) FTYPE
      IF (FTYPE .NE. 'LIFTER-1') THEN
       WRITE (*,*) 'INCORRECT DATA FILE'
```

```
        STOP
       ENDIF
       READ (1) (LVDT(J) ,J=1,NPTS)
       CLOSE(1)

       WRITE (*,25)
25      FORMAT( /,5X,' enter LVDT file name (e.g. LVDTLK): '$ )
       READ (*,28) IOSTR
28      FORMAT (A9)
       STRING=IOSTR
       WRITE (FILSTR,'(A1,A1,A9)') 'B',':',IOSTR
       WRITE (*,*) 'Reading data from disk'
30     OPEN(1,FILE=FILSTR,STATUS='OLD',FORM='BINARY')
       READ (1) FTYPE
       IF (FTYPE .NE. 'LIFTER-1') THEN
        WRITE (*,*) 'INCORRECT DATA FILE'
        STOP
       ENDIF
       READ (1) (LVDTIJ(J) ,J=1,NPTS)
       CLOSE(1)


C...................READ ROLLER ERROR SPEED FILE................
       WRITE (*,*) 'ENTER ROLLER ERROR FILE e.g. DEGMRK (RETURN=NONE)'
       READ (*,'(A14)') CFILE
       IF (CFILE .NE. ' ') THEN
        WRITE (FILSTR,'(2A1,A12)') 'B',':',CFILE
        OPEN (1,FILE=FILSTR,STATUS='OLD',FORM='FORMATTED')
        READ (1,*) ERTDC
        READ (1,'(4F7.5)') (DGMRK(K),K=1,119)
        CLOSE(1)
       ENDIF
       WRITE(*,*) 'ERTDC',ERTDC
       WRITE(*,*) 'DGMRK',DGMRK(3)


C
C               roller speed
       WRITE(*,*) 'NPTS=',NPTS

       DO 310 I=1,NPTS
          ROLSPD(I)=ROLLSP(I)
          IF(ROLSPD(I).LT.0)ROLSPD(I)=ROLSPD(I)+65536
310    CONTINUE
C      DO 311 I=1,60
C              WRITE(*,*)ROLSPD(I),ROLLSP(I)
C311     CONTINUE
       WRITE(*,*)'DO YOU WANT SPEED CALIBRATION(Y=1,N=2)'
       READ(*,*)ANS
       IF(ANS.EQ.2)GOTO 393

C *****************************************************************
C *****************************************************************
C                         USED FOR ROLLER CALIBRATION
```

```
C
C              Reduce roller speed from 360 marks/rev to 119 marks/rev
C
      IDEG=1
      REDSPD(1)=ROLSPD(1)
      CAMAGL(1)=1
      DO 315 I=2,NPTS-1
         IF(ROLSPD(I).NE.ROLSPD(I+1))THEN
              IDEG=IDEG+1
              REDSPD(IDEG)=ROLSPD(I+1)
              CAMAGL(IDEG)=I
         ENDIF
315   CONTINUE
      WRITE(*,*)'IDEG=',IDEG
      OPEN (3,FILE='LPT1')
C     DO 317 I=1,20
C             WRITE(3,*)ROLSPD(I+104),ROLSPD(I+226)
C317   CONTINUE


C
C              Correct for roller speed noise

      IF(REDSPD(1).EQ.0)REDSPD(1)=REDSPD(IDEG)
      IF(REDSPD(1).EQ.0)REDSPD(1)=REDSPD(IDEG-1)
      IF(REDSPD(1).EQ.0)THEN
         WRITE(*,*)'TOO MANY ZEROS INPUT NEW FILE'
         GOTO 3
      ENDIF
      DO 318 I=2,IDEG
          IF(REDSPD(I).EQ.0)REDSPD(I)=REDSPD(I-1)+1
318   CONTINUE
      DO 319 I=1,60
         WRITE(*,*)'REDSPD=',REDSPD(I)
319   CONTINUE

      AVGSPD=0.0
      R=REDSPD(IDEG)
      R2=REDSPD(2)
      R1=REDSPD(1)

      AVGSPD=(R+R2)/2.0
      WRITE(*,*)'AVGSPD1=',AVGSPD
      IF(R1.LT.AVGSPD*.25.OR.R1.GT.2.25*AVGSPD)THEN
           REDSPD(1)=AVGSPD
           WRITE(*,*)'Roller speed noise found at 1',AVGSPD,REDSPD(1)

      ENDIF
      R=REDSPD(IDEG)
      R2=REDSPD(1)
      R1=REDSPD(IDEG-1)
      AVGSPD=(R1+R2)/2.0
      WRITE(*,*)'AVGSPD END=',AVGSPD
```

```
      IF(R.LT.AVGSPD*.25.OR.R.GT.2.25*AVGSPD)THEN
              REDSPD(IDEG)=AVGSPD
              WRITE(*,*)'Roller speed noise found at 3600',AVGSPD,
     +        REDSPD(IDEG)
      ENDIF
      DO 320 I=2,IDEG-1
          R=REDSPD(I)
          R1=REDSPD(I-1)
          R2=REDSPD(I+1)
          AVGSPD=(R1+R2)/2.0
          IF(R.LT.AVGSPD*.25.OR.R.GT.2.25*AVGSPD)THEN
              REDSPD(I)=AVGSPD

              WRITE(*,*)'Roller speed noise found at ',I,AVGSPD,
     +        REDSPD(I)
          ENDIF
320   CONTINUE
C
C             find TDC
C
      AVGSPD=0.0
      ICOUNT=0
      DO 330 I=2,IDEG-1
         AVGSPD=(REDSPD(I-1)+REDSPD(I+1))/2
         IF(REDSPD(I).LT.AVGSPD*2.25.AND.REDSPD(I).GT.AVGSPD*1.75)THEN
              ICOUNT=ICOUNT+1
              ITDC(ICOUNT)=I

         ENDIF
330   CONTINUE
      IF(ITDC(1).EQ.120)THEN
         DO 340 I=1,ICOUNT
              ITDC(I)=ITDC(I)-119
340      CONTINUE
      ENDIF




      DO 345 I=1,ICOUNT-1

         ITDCD(I)=ITDC(I+1)-ITDC(I)
         WRITE(3,*)'ITDC=',ITDC(I)
         WRITE(3,*)'DIFF',ITDCD(I)
345   CONTINUE


C
C             shift error file
C
      DO 350 I=1,119
         SHIFT=I+ERTDC-ITDC(1)
         IF(SHIFT.GT.119)SHIFT=SHIFT-119
```

```
         IF(SHIFT.LT.1)SHIFT=119+SHIFT
         EDGMRK(I)=DGMRK(SHIFT)
350   CONTINUE
C
C  Expand out the degrees per mark to calculate actual roller speed
C
      DO 360 I=1,119
         DO 360 J=1,16
360         EDGMRK(I+J*119)=EDGMRK(I)


C
C            convert to RPM and back to 360 marks/rev
C
C
      DO 365 I=1,3600
             ROLRPM(I)=0.0
365   CONTINUE
      DO 370 I=1,IDEG
            ROLRPM(CAMAGL(I))=(EDGMRK(I)/REDSPD(I))*1000000*60/360
370   CONTINUE
C     CALL PLT(REDSPD,2000.,TITLE,1,HEADING,ACRPM)
C     TITLE='REDSPD'
C     CALL PLT(ROLRPM,2000.,TITLE,1,HEADING,ACRPM)
C     TITLE='ROLRPM'


C     DO 373 I=1,300
C       WRITE(3,*)I,ROLLSP(I),ROLSPD(I),REDSPD(I),EDGMRK(I)
C       WRITE(3,*)CAMAGL(I),ROLRPM(CAMAGL(I))
C       WRITE(3,*)I

C       WRITE(3,*)'REDSPD=',REDSPD(I)
C       WRITE(3,*)'ROLRPM=',ROLRPM(CAMAGL(I))
C       WRITE(3,*)'EDGMRK=',EDGMRK(I)
C373   CONTINUE
      CLOSE (3)


C
C            fill in blank spots
C
      IF(ROLRPM(1).EQ.0)ROLRPM(1)=ROLRPM(CAMAGL(1))
      DO 390 I=2,3600
         IF(ROLRPM(I).EQ.0)ROLRPM(I)=ROLRPM(I-1)
390   CONTINUE
C
C ******************************************************************
C ******************************************************************
C                      END OF ROLLER CALIBRATION
C
C
393   IF(ANS.EQ.1)GOTO 399
      IF(ROLSPD(1).EQ.0)ROLSPD(1)=ROLSPD(NPTS)
      IF(ROLSPD(1).EQ.0)ROLSPD(1)=ROLSPD(NPTS-1)
```

```
      IF(ROLSPD(1).EQ.0)THEN
         WRITE(*,*)'TOO MANY ZEROS INPUT NEW FILE'
         GOTO 3
      ENDIF
      DO 394 I=2,NPTS

          IF(ROLSPD(I).EQ.0)ROLSPD(I)=ROLSPD(I-1)
394   CONTINUE
      DO 395 I=1,NPTS
         ROLRPM(I)=0.0
395   CONTINUE
      DO 396 I=1,NPTS
        ROLRPM(I)=(3.0/ROLSPD(I))*1000000.*60./360.
396   CONTINUE
      DO 397 I=2,NPTS
         IF(ROLRPM(I).LT.ROLRPM(I-1)*.75.OR.ROLRPM(I).GT.ROLRPM(I-1)*
     +   1.25)ROLRPM(I)=ROLRPM(I-1)
397   CONTINUE
C     DO 398 I=1,400
C        WRITE(*,*)'ROLRPM',ROLRPM(I)
C398   CONTINUE


C.........Average torque and rod force and convert to (FT-LB) AND (LBF)..
399   WRITE (*,*) 'Calculating'
      WRITE (*,*) 'ENTER CYCLE NUMBER TO ANALIZE (11=AVERAGE)'
      READ (*,*) I
      ISTART=1
      IF (I .NE. 11) THEN
       ISTART=I
       NACYC=I
      ENDIF
      IF (I .EQ. 11) NACYC=NPTS/NDPC
C  NACYC=NUMBER OF CYCLES OF DATA TO USE
      ACRPM=0.0
      DO 400 I=1,NDPC
       PRF(I)=0.0
       IJF(I)=0.0
       RTQ(I)=0.0
       CSP(I)=0.0
       RSP(I)=0.0
       PRL(I)=0.0
       IJL(I)=0.0
       DO 405 L=ISTART,NACYC
        IDEG=(L-1)*NDPC+I
        PRF(I)=PRF(I)+PRFORC(IDEG)/YSCL/(NACYC-ISTART+1)
        IJF(I)=IJF(I)+INJECT(IDEG)/YSCL/(NACYC-ISTART+1)
        RTQ(I)=RTQ(I)+ROTORQ(IDEG)/YSCL/(NACYC-ISTART+1)
        J=CAMSPD(IDEG)
        IF (J .LE. 0) J=J+65536
        CSP(I)=CSP(I)+1000000/J*60/720/(NACYC-ISTART+1)
        RSP(I)=RSP(I)+ROLRPM(IDEG)/(NACYC-ISTART+1)
        PRL(I)=PRL(I)+LVDT(IDEG)/YSCL/(NACYC-ISTART+1)
        IJL(I)=IJL(I)+LVDTIJ(IDEG)/YSCL/(NACYC-ISTART+1)
405    CONTINUE
```

```
      ACRPM=ACRPM+CSP(I)/NDPC
400   CONTINUE
C  PRF=PUSHROD FORCE IN VOLTS
C  IJF=INJ LINK FORCE IN VOLTS
C  RTQ=ROLLER TORQUE IN VOLTS
C  CSP=CAMSPEED IN RPM
C  RSP=ROLLER SPD IN RPM
C  PRL=PUSHROD LIFT IN VOLTS
C  IJL=INJ LINK LIFT IN VOLTS
      OPEN(3,FILE='LPT1')
C     WRITE(3,*)'PRF,PRL,CSP,RSP,RTQ'
C     DO 407 I=100,200
C       WRITE(3,*)PRF(I),PRL(I),CSP(I),RSP(I),RTQ(I)
C407   CONTINUE


      RTQZ1=(RTQ(45)+RTQ(49)+RTQ(52)+RTQ(55))/4.
      PRZ=(PRF(45)+PRF(49)+PRF(52)+PRF(55))/4.
      FINJTZ=(IJF(45)+IJF(49)+IJF(52)+IJF(55))/4.
      DO 410 I=1,NDPC
       PRF(I)=(PRF(I)-PRZ)*PRS
       IJF(I)=(IJF(I)-FINJTZ)*FINJTS
C410     RTQ(I)=(RTQ(I)-RTQZ1)*RTQS-RTQZ
410    RTQ(I)=(RTQ(I)-RTQZ1)*RTQS-RTQZ

      Y=2400.
      WRITE(*,*)'Input Ymax P Rod force (=2400)'
      READ(*,*)Y
      HEAD='PUSHROD FORCE (LBF) VS CAM ANGLE'
      TITLE='Pushrod Force LBF'
      CALL PLT(PRF,Y,TITLE,1,HEAD,ACRPM)
      Y=2400.
      WRITE(*,*)'Input Ymax Inj Link Force(=2400)'
      READ(*,*)Y
      HEAD='INJ LINK FORCE (LBF) VS CAM ANGLE'
      TITLE='Injector Link Force LBF'
      CALL PLT(IJF,Y,TITLE,1,HEAD,ACRPM)
      Y=0.4
      WRITE(*,*)'Input Ymax-Roller Torq(=0.40)'
      READ(*,*)Y
      TITLE='Roller Torque FT LBF'
      HEAD='ROLLER FRICTION TORQUE (FT-LBF) VS CAM ANGLE'
      CALL PLT(RTQ,Y,TITLE,1,HEAD,ACRPM)
      Y=1000.
      WRITE(*,*)'Input Ymax-Cam Speed (=1000)'
      READ(*,*)Y
      TITLE='Cam Speed RPM'
      HEAD='CAM SPEED (RPM) VS CAM ANGLE'
      CALL PLT(CSP,Y,TITLE,1,HEAD,ACRPM)

C..........Read ECCENTRICITY file....................................
C440   OPEN (1,FILE='A:AVECC',STATUS='OLD',FORM='FORMATTED')
C      READ (1,430) N,N,N,N,STR,N,A,A,N,N
C      READ (1,435) (ECC(K),K=1,NDPC)
C      CLOSE (1)
```

```
C..........Convert pushrod lift to feet.......................
      P=(PRL(45)+PRL(52)+PRL(55))/3.
      PINJ=(IJL(45)+IJL(52)+IJL(55))/3.
      DO 450 I=1,NDPC
       PRL(I)=(PRL(I)-P)*LS/12.
       IJL(I)=(IJL(I)-PINJ)*LS/12.
C   NOTE: THE PRL AND IJL MUST BE /12 SINCE CAL CONST IS IN INCHES
450   CONTINUE
      CALL SLAVG(PRL,3)
      CALL SLAVG(IJL,3)
      ZMAX=PRL(1)
      DO 455 I=1,NDPC
        IF(ZMAX.LT.PRL(I))THEN
          ZMAX=PRL(I)
          PMAX=I
        ENDIF
455   CONTINUE
      WRITE(*,*)'MAX POINT=',ZMAX,'  AT ',PMAX
      DO 460 I=1,10
        WRITE(*,*)PRL(I),I
460   CONTINUE


      TITLE='Pushrod Lift FT'
      HEAD='PUSHROD LIFT (FT) VS CAM ANGLE'
      CALL PLT(PRL,.04,TITLE,1,HEAD,ACRPM)
      TITLE='Inj Rod Lift FT'
      HEAD='INJ ROD LIFT (FT) VS CAM ANGLE'
      CALL PLT(IJL,.04,TITLE,1,HEAD,ACRPM)


C..........Calculate pushrod velocity (FT/SECOND)..................
      IF (ACRPM .EQ. 0) THEN
       WRITE (*,*) 'ENTER ACRPM'
       READ (*,*) ACRPM
      ENDIF
      DPS=ACRPM*360/60
C   DPS=DEGREES PER SECOND
      DO 500 I=1,NDPC
       IMINUS=I-1
       IPLUS=I+1
       IF (IMINUS .LT. 1) IMINUS=NDPC
       IF (IPLUS .GT. NDPC) IPLUS=1
       PRV(I)=DPS*(PRL(IPLUS)-PRL(IMINUS))/2.
500    IJV(I)=DPS*(IJL(IPLUS)-IJL(IMINUS))/2.
       CALL SLAVG(PRV,5)
       CALL SLAVG(IJV,5)
       Y=2.6
       WRITE(*,*)'Input Ymax P-Rod Vel (=2.6)'
       READ(*,*)Y
       TITLE='Pushrod Velocity FT/SEC'
       HEAD='PUSHROD VELOCITY (FT/SECOND) VS CAM ANGLE'
       CALL PLT(PRV,Y,TITLE,1,HEAD,ACRPM)
       Y=2.6
       WRITE(*,*)'Input Ymax Ij-Rod Vel (=2.6)'
```

```
      READ(*,*)Y
      TITLE='Inj Rod Velocity FT/SEC'
      HEAD='INJ ROD VELOCITY (FT/SECOND) VS CAM ANGLE'
      CALL PLT(IJV,Y,TITLE,1,HEAD,ACRPM)

C...........Calculate pushrod acceleration (FT/SECOND**2)..........
      DO 550 I=1,NDPC
       IMINUS=I-1
       IPLUS=I+1
       IF (IMINUS .LT. 1) IMINUS=NDPC
       IF (IPLUS .GT. NDPC) IPLUS=1
550    PRA(I)=DPS*(PRV(IPLUS)-PRV(IMINUS))/2
      CALL SLAVG(PRA,5)
      Y=400.
      WRITE(*,*)'Input Ymax P-Rod Accel (=400)'
      READ(*,*)Y
      TITLE='Pushrod Accel FT/SEC**2'
      HEAD='PUSHROD ACCELERATION (FT/SECOND**2) VS CAM ANGLE'
      CALL PLT(PRA,Y,TITLE,1,HEAD,ACRPM)

C..........Convert ECCENTRICITY to feet.........................
C615   DO 620 I=1,NDPC
C620    ECC(I)=ECC(I)/12.
C      CALL SLAVG(ECC,3)
C      CALL XSHIFT(ECC)
C      TITLE='CAM LOBE ECCENTRICITY (FT)'
C      CALL PLT(ECC,.04,TITLE,1,TITLE,ACRPM)
C...........Calculate push rod work.................................
      APRW=0.0
      DO 625 I=1,NDPC
       PRW(I)=PRF(I)*PRV(I)/DPS-IJF(I)*IJV(I)/DPS
       APRW=APRW+PRW(I)/NDPC
625   CONTINUE
      Y=1.
      WRITE(*,*)'Input Ymax Work (=1)'
      READ(*,*)Y
      TITLE='Pushrod Work FT LBF'
      HEAD='PUSHROD WORK (FT-LBF) VS CAM ANGLE'
      CALL PLT(PRW,Y,TITLE,1,HEAD,ACRPM)
C
C              roller acceleration

      Y=1400.
      WRITE(*,*)'Input Ymax Rol Spd (=1400)'
      READ(*,*)Y
      TITLE='Roller Speed RPM'
      HEAD='ROLLER SPEED (RPM) VS CAM ANGLE'
      CALL PLT(RSP,Y,TITLE,1,HEAD,ACRPM)
      CALL SLAVG(RSP,5)
      HEAD='AVERAGED ROLLER SPEED (RPM) VS CAM ANGLE'
      CALL PLT(RSP,Y,TITLE,1,HEAD,ACRPM)
```

```
C
      ROLA(1)=DPS*(RSP(2)-RSP(NDPC))/2.*2.0*3.14159/60
      ROLA(NDPC)=DPS*(RSP(1)-RSP(NDPC-1))/2.*2.0*3.14159/60
      DO 630 I=2,NDPC-1
        ROLA(I)=DPS*(RSP(I+1)-RSP(I-1))/2.*2.0*3.14159/60
630   CONTINUE
      DO 632 I=1,NDPC
C       ARMTH(I)=ASIN(PRL(I)*5.8065)  **FOR EXHAUST
        ARMTH(I)=ASIN(PRL(I)*8.3624)
632   CONTINUE
      ARMVL(1)=DPS*(ARMTH(2)-ARMTH(NDPC))/2.
      ARMVL(NDPC)=DPS*(ARMTH(1)-ARMTH(NDPC-1))/2.
      DO 635 I=2,NDPC-1
        ARMVL(I)=DPS*(ARMTH(I+1)-ARMTH(I-1))/2.
635   CONTINUE
      ARMAC(1)=DPS*(ARMVL(2)-ARMVL(NDPC))/2.
      ARMAC(NDPC)=DPS*(ARMVL(1)-ARMVL(NDPC-1))/2.
      DO 637 I=2,NDPC-1
        ARMAC(I)=DPS*(ARMVL(I+1)-ARMVL(I-1))/2.

637   CONTINUE


      CALL SLAVG(ARMAC,5)

C
C             roller inertia and tractive forces
C
      Y=5000.
      WRITE(*,*)'Input Ymax Rol Accel  (=5000)'
      READ(*,*)Y
      TITLE='Rel. Accel. RAD/SEC**2'
      HEAD='ROLLER RELATIVE ACCELERATION  (RAD/SEC**2)'
      CALL PLT(ROLA,Y,TITLE,1,HEAD,ACRPM)
      CALL SLAVG(ROLA,5)

      HEAD='ROLLER RELATIVE ACCELERATION  (RAD/SEC**2)-AVG'
      CALL PLT(ROLA,Y,TITLE,1,HEAD,ACRPM)
      CALL SLAVG(ARMAC,5)
      TITLE='Follower Accel. RAD/SEC**2'
      HEAD='FOLLOWER ARM ACCELERATION(RAD/SEC**2)'
      WRITE(*,*)'Input Ymax arm accel  (=2000)'
      READ(*,*)Y
      CALL PLT(ARMAC,Y,TITLE,1,HEAD,ACRPM)
      RAD=0.06771

      DO 640 I=1,NDPC
         TRQIN(I)=RINER*(ROLA(I)-ARMAC(I))
640   CONTINUE
      CALL SLAVG(TRQIN,5)
      DO 650 I=1,NDPC
```

```
         TRFRC(I)=(RTQ(I)+TRQIN(I))/RAD
650   CONTINUE
      CALL SLAVG(TRFRC 5)
      Y=0.4
      WRITE(*,*)'Input Ymax Iner Torq (=0.4)'
      READ(*,*)Y
      TITLE='Inertia Torque FT LBF'
      HEAD='INERTIA TORQUE (FT-LB) VS CAM ANGLE'
      CALL PLT (TRQIN,Y,TITLE,1,HEAD,ACRPM)
      Y=8.0
      WRITE(*,*)'Input Ymax Tract Force (=8)'
      READ(*,*)Y
      TITLE='Roller Tractive Force LBF'
      HEAD='ROLLER TRACTIVE FORCE (LBF) VS CAM ANGLE'
      CALL PLT (TRFRC,Y,TITLE,1,HEAD,ACRPM)
      CALL SLAVG(CSP,5)
C     REV(1)=(RSP(360)+RSP(1))/2.*(2./(CSP(360)+CSP(1)))
      REV(1)=(RSP(360)+RSP(1))/2./ACRPM
      DO 700 I=2,360
      REV(I)=(RSP(I-1)+RSP(I))/2./ACRPM
C     REV(I)=(RSP(I-1)+RSP(I))/2.*(2./(CSP(I-1)+CSP(I)))

700   CONTINUE
      WRITE(*,*)'Input roller rotation'
      READ(*,*)Y
      TITLE='DEGREES OF ROLLER ROTATION'
      HEAD='DEGREES OF ROLLER ROTATION AT CAM ANGLE'
      CALL PLT (REV,Y,TITLE,1,HEAD,ACRPM)




      CLOSE(3)
C.................send test conditions to printer.........................
1200  WRITE (*,*) 'ENTER RETURN TO PRINT TEST CONDITIONS'
      READ (*,'(A1)') IOSTR
1300  OPEN (3,FILE='LPT1')
      WRITE (3,'(/////////)')
      WRITE (3,1337)
1337  FORMAT(' ',)
      WRITE (3,1340) DATE
1340  FORMAT ('            DATE                  : ',A8)
      WRITE (3,1350) TIME
1350  FORMAT ('            TIME                  : 'A8)
      WRITE (3,*) ' '
      WRITE (3,1352) ETYPE,NTEST
1352  FORMAT ('            TEST NO.              : ',A1,' -',I4)
      WRITE (3,1353) NRUN
1353  FORMAT ('            RUN NO.               : ',I2)
      WRITE (3,*) ' '
      WRITE (3,1354) BARPRS
1354  FORMAT ('            BAR PRESS             =',F6.2,'IN HG')
```

```
      WRITE (3,1356) ITDRY
1356  FORMAT ('              TDRY                    =',I3,' F')
      WRITE (3,1358) ITWET
1358  FORMAT ('              TWET                    =',I3,' F')
      WRITE (3,*) ' '
      WRITE (3,1364) TBRAND,TNO
1364  FORMAT ('              TRANS.                  : ',A4,A8)
      WRITE (3,1366) ABRAND,ANO
1366  FORMAT ('              AMP.                    : ',A4,A8)
      WRITE (3,1368) TS
1368  FORMAT ('              PRESS TR. SLOPE    :',F8.4)
      WRITE (3,*) ' '
      IF(IFIRE .EQ. 1) WRITE (3,*) '          F I R I N G'
      IF(IFIRE .EQ. 0) WRITE (3,*) '          M O T O R I N G'
      WRITE (3,*) ' '
      WRITE (3,1370) ACRPM
1370  FORMAT ('              CAMSHAFT RPM       = ',F6.1)
      WRITE (3,1372) VAC
1372  FORMAT ('              MAN VACUUM IN HG  =',F6.2)
      WRITE (3,1374) AINP
1374  FORMAT ('              INT ABS PRESS-PSI =',F6.3)
      IF(IFIRE .EQ. 0)GOTO 1390
      WRITE (3,1380) ISPKAD
1380  FORMAT ('              SPARK ADV-DEG BTDC=',I3)
      WRITE (3,1382) FUELWT
1382  FORMAT ('              FUEL WEIGHT-LBF   =',F8.6)
      WRITE (3,1384) ELPTIM
1384  FORMAT ('              ELAPSED TIME-MIN  =',F8.4)
      WRITE (3,1386) ITEXH
1386  FORMAT ('              EXHAUST TEMP-F    =',I5)
1390  WRITE (3,*) ' '
      IF(IFIRE .EQ. 0) WRITE (3,*) ' '
      IF(IFIRE .EQ. 0) WRITE (3,*) ' '
      IF(IFIRE .EQ. 0) WRITE (3,*) ' '
      IF(IFIRE. EQ. 0) WRITE (3,*) ' '
      WRITE (3,1360) LUBRIC
1360  FORMAT ('              LUBRICANT          : ',A8)
      WRITE (3,1376) IOILP
1376  FORMAT ('              OIL PRESS-PSI      =',I3)
      WRITE (3,1394) ITOILS
1394  FORMAT ('              SUMP OIL TEMP-F    =',I4)
      WRITE (3,1392) ITOILG
1392  FORMAT ('              GALLERY OIL TEMP-F  =',I4)
      WRITE (3,1396) ITSG
1396  FORMAT ('              STRAIN GAGE TEMP-F=',I4)
      WRITE (3,*) ' '
      WRITE (3,1400) TORQUE
1400  FORMAT ('              ENG TORQUE (FT-LBF)=',F6.2)
      WRITE (3,1402) TORQUE*1.2415
1402  FORMAT ('              ENGINE MEP (PSI)   =',F6.2)
C
      WRITE(3,1660) APRW
1660  FORMAT('            PR FRICTION WORK (FT-LBF)  =',F6.5)
      WRITE(3,1665) FMEP,EXHINT,COMPWR
1665  FORMAT('            PR FMEP          (PSI)      =',F6.2,
```

```
     +       '(IE=',F5.2,',PC=',F5.2,')')
1690     CONTINUE
      WRITE(3,*) ' '
      WRITE(3,1695) ZER
1695  FORMAT('             ZERO AXIS OF AVG (LBF)       =',F6.2)
      WRITE(3,1696) EHOUR
1696  FORMAT('             ENGINE HOURS                 =',F6.2)
      WRITE(*,*)'Enter first comment line.'
      READ (*,'(A60)') COMMENT
      WRITE(3,*) '             NOTE: ',COMMENT
      WRITE(*,*)'Enter second comment line.'
      READ (*,'(A60)') COMMENT
      WRITE(3,*) '                   ',COMMENT
      WRITE(3,*) ' '
1740  WRITE (3,1760)
1760  FORMAT (' '///)
      WRITE (3,'(//////////)')

      WRITE(3,'(A1)') '1'
      CLOSE (3)
CC......................save friction data on disk for MTS...............
C      READ (STRING,'(1X,A8)') FILSTR
C      WRITE (IOSTR,'(A1,A8)') 'F',FILSTR
C1800  WRITE (*,1810) IOSTR
C1810  FORMAT (/,5X,' Enter friction storage file name:',A14 $)
C      READ (*,'(A14)') FILSTR
C      IF (FILSTR .EQ. '              ') FILSTR=IOSTR
C      WRITE(IOSTR,'(A1,A1,A9)') 'A',':',FILSTR
C      OPEN (1,FILE=IOSTR,STATUS='NEW',FORM='FORMATTED')
C      WRITE (1,1825) (0.0,L=1,18)
C1825  FORMAT(F8.2)
C      WRITE (1,1825) 1.0
C      WRITE (1,1830) NACYC,IFIRE,NTEST,NRUN,LUBRIC,IRPM,AINP,FMEP
C     +   ,ITWAT,ITOIL
C1830  FORMAT(4I4,A8,I4,2F6.2,2I4)
C1835  FORMAT(10F8.2)
C      WRITE (1,1835) (FR(K),K=1,NDPC)
C      CLOSE (1)
      GOTO 3
      STOP
      END

      SUBROUTINE SLAVG(PR,N)
C...............This subroutine does an N point sliding average......
      REAL*4 PR(360),AVG(360)
C
      DO 100 I=1,360
       AVG(I)=0.
       DO 100 J=-(N/2),N/2
        ITHETA=I+J
        IF (ITHETA .LT. 1) ITHETA=360+ITHETA
        IF (ITHETA .GT. 360) ITHETA=ITHETA-360
100     AVG(I)=AVG(I)+PR(ITHETA)/N
      DO 200 I=1,360
200    PR(I)=AVG(I)
```

```
      RETURN
      END

      SUBROUTINE XSHIFT(ECC)
C..............This subroutine shifts the eccentricity data to match
C              the maximum lift data in the X direction..............
      REAL*4 ECC(360),ECCTEMP(360)
C PRTEMP AND TQTEMP ARE TEMPORARY BUFFERS FOR PRF AND TQ

C  MAX LIFT OCCURS AT 184 DEGREES
      ILIFT=184
      DO 100 I=170,190
       IF (ECC(I) .GT. 0.0) GOTO 150
100   CONTINUE
      WRITE (*,*) 'ERROR IN FINDING ZERO ECCENTRICTY'
150   WRITE (*,*) 'RECOMENDED SHIFT IS',ILIFT-I,' DEGREES RIGHT.'
      WRITE (*,*) 'ENTER DESIRED SHIFT.'
      READ (*,*) ISHIFT
      DO 200 I=1,360
       J=I+ISHIFT
       IF (J .LT. 1   ) J=360-J
       IF (J .GT. 360) J=J-360
200    ECCTEMP(J)=ECC(I)
      DO 300 I=1,360
300    ECC(I)=ECCTEMP(I)
      RETURN
      END

      FUNCTION LEN(STRING)
C  THIS FUNCTION RETURNS THE LENGTH OF THE GIVEN STRING
      CHARACTER*1 STRING(60)
      L=1
      DO 100 I=1,80
100    IF (STRING(I) .NE. ' ') L=I
      LEN=L
      RETURN
      END

      SUBROUTINE TEXTF(IX,IY,L,TITLE1)
C  THIS SUBROUTINE CONVERTS TITLE DATA TO HALO FORMAT
      CHARACTER*1 TITLE1(80),TITLE2(80)
      IF (L .EQ. -1) L=LEN(TITLE1)
      DO 100 I=1,L
100    TITLE2(I+1)=TITLE1(I)
      TITLE2(1)='~'
      TITLE2(L+2)='~'
      CALL MOVTCA(IX*1.,IY*1.)
      CALL TEXT(TITLE2)
      RETURN
      END

      SUBROUTINE PUTPT(IX,IY)
C  THIS SUBROUTINE CONVERTS THE CALL TO A HALO CALL
      CALL MOVABS(IX*1.,IY*1.)
      RETURN
```

```
      END

      SUBROUTINE DLINE(IX,IY)
C  THIS SUBROUTINE CONVERTS THE CALL TO A HALO CALL
      CALL LNABS(IX*1.,IY*1.)
      RETURN
      END

      SUBROUTINE TMODE
C  THIS SUBROUTINE SWITCHES THE SCREEN BACK TO TEXT MODE
      CALL CLOSEG
      RETURN
      END

      SUBROUTINE GMODE
C  THIS SUBROUTINE INITIALIZES GRAPHICS MODE
      INTEGER*2 MODE,MODE1,MODE2
      CALL SETDEV('/\FORTRAN\HALO\HALOIBMP.DEV/')
      MODE=12
      CALL INITGR(MODE)
      MODE=1
      CALL SETIEE(MODE)
      CALL INQCRA(MODE)
      CALL SETWOR(0.,348.,719.,0.)
      CALL SETFON('/\FORTRAN\HALO\HALO001.FNT/')
      CALL SETSTC(MODE,MODE)
      MODE=0
      MODE1=1
      MODE2=2
      CALL SETTEX(MODE2,MODE1,MODE,MODE)
      RETURN
      END

      SUBROUTINE CLRSCR
      RETURN
      END

      SUBROUTINE PSCREEN
C  THIS SUBROUTINE PRINTS THE GRAPHICS SCREEN ON THE LASER JET
      INTEGER*2 PATTR(17)
C
      CALL SETPRN('/\FORTRAN\HALO\HALOLJTP.PRN/')
      DO 710 I=1,17
710     PATTR(I)=-1
C  THERE IS A SHIFT OF ONE IN THE PATTR ARRAY COMPARED TO THE MANUAL
      PATTR(1)=1700
      PATTR(2)=1700
      PATTR(6)=0
      PATTR(8)=3
      CALL SETPAT(PATTR)
      CALL GPRINT
      RETURN
      END

      SUBROUTINE PLT(DATA,YMAX,YTITLE,MEAN,HEADING,RPM)
```

```
C ***********************************************
C *  THIS ROUTINE PLOTS PR FRICTION ON THE SCREEN *
C ***********************************************
      REAL*4 DATA(360),SDATA(360)
      CHARACTER*15 MSG
      CHARACTER*60 TITL,YTITLE,HEADING
      CHARACTER*1 MG(15),TITLE(60),CH
      INTEGER*2 ILINE(32)
      EQUIVALENCE (TITL,TITLE),(MSG,MG)
C .................set up axes.............................
      CALL GETYN('DO YOU WANT THE PLOT PRINTED                   ',PRNT)
      CALL GMODE
      CALL CLRSCR
      SCL=YMAX/80.
C  DRAW X-AXIS AND LABEL
      DO 20 I=-180,180,90
       J=67+(I*2+359)*645./720.
       N=67+(I*2+345)*645./720.
       CALL PUTPT(J,24)
       CALL DLINE(J,314)
CCC
       WRITE (MSG,'(I4)') I+90
CCC
20     CALL TEXTF(N-15,333,4,MG)
      TITL='CAM ANGLE (DEG)'
      CALL TEXTF (500,345,20,TITLE)
C  DRAW HEADING
      TITL=HEADING
      CALL TEXTF(111,12,60,TITLE)
      TITL='O p e n i n g'
      CALL TEXTF(155,23,13,TITLE)
      TITL='C l o s i n g'
      CALL TEXTF(480,23,13,TITLE)
C  DRAW Y-AXIS AND LABEL
      DO 100 I=-80,80,10
       Z=I
       J=169-I*145./80.
       CALL PUTPT (67,J)
       CALL DLINE(710,J)
       IF (SCL .GE. .125) WRITE (MSG,'(F6.0)') Z*SCL
       IF (SCL .LT. .125) WRITE (MSG,'(F6.3)') Z*SCL
       CALL TEXTF (10,J+5,6,MG)
100   CONTINUE
      TITL=YTITLE
      DO 150 I=1,40
       CH=TITLE(I)
       CALL TEXTF(2,6+I*13,1,CH)
150   CONTINUE
      IF (MEAN .EQ. 1) THEN
       AVG=0.0
       DO 180 I=1,360
180     AVG=AVG+DATA(I)/360.
       WRITE (MSG,'(A4,F9.5)') 'AVG ',AVG
       CALL TEXTF(590,277,12,MG)
       WRITE (MSG,'(A8,F5.0)') 'CAM RPM ',RPM
```

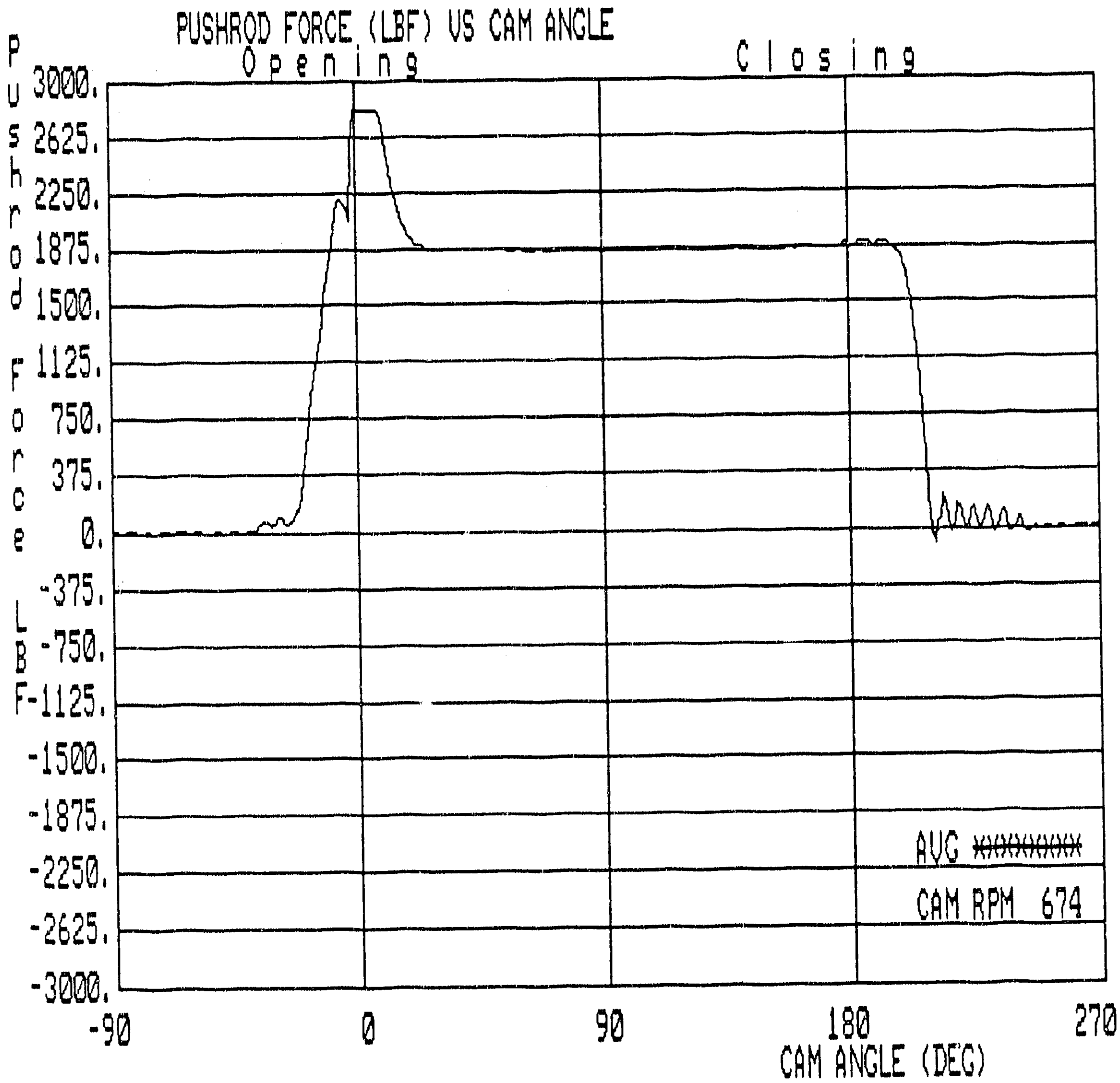
```
         CALL TEXTF(590,295,12,MG)
        ENDIF
C.................plot actual data............................
        PT=0.
        DO 198 J=1,360
          ISHIFT=J+327
          IF(ISHIFT.GT.360)ISHIFT=ISHIFT-360
          SDATA(ISHIFT)=DATA(J)
198     CONTINUE
        DO 200 I=1,360
         IPLOT=169-SDATA(I)*145./80./SCL
         IX=67+(I-1)*645./360.
         IF (IPLOT .GT. 314) IPLOT=314
         IF (IPLOT .LT. 10) IPLOT=10
         IF (PT .EQ. 0.) THEN
             CALL PUTPT(IX,IPLOT)
             PT=.1
         ELSE
             CALL DLINE(IX,IPLOT)
         ENDIF
200     CONTINUE
        IF (PRNT .EQ. 0) THEN
         READ (*,'(A1)') CH
        ELSE
         CALL PSCREEN
        ENDIF
        CALL TMODE
        RETURN
        END

        SUBROUTINE GETREL(QUESTION,ANSWER)
C  THIS SUBROUTINE READS IN A REAL NUMBER FROM THE KEYBOARD
        CHARACTER*40 QUESTION
        CHARACTER*10 STR
        CHARACTER*1 STRING(10)
        EQUIVALENCE (STR,STRING)
C
50      WRITE (*,*) QUESTION,ANSWER
        READ (*,'(A10)',ERR=100) STR
        J=0
        DO 75 I=1,10
         IF (STRING(I) .EQ. '.') GOTO 80
75       IF (STRING(I) .NE. ' ') J=I
        IF (J .NE. 0) STRING(J+1)='.'
80      IF (STR .NE. '          ') READ (STR,'(BN,F10.4)',ERR=100) ANSWER
        RETURN
100     WRITE (*,*) '*** Input error.  Please re-enter data ***'
        GOTO 50
        STOP
        END

        SUBROUTINE GETINT(QUESTION,IANSWER)
C  THIS SUBROUTINE READS IN AN INTEGER FROM THE KEYBOARD
        CHARACTER*40 QUESTION
        CHARACTER*9 STR
```

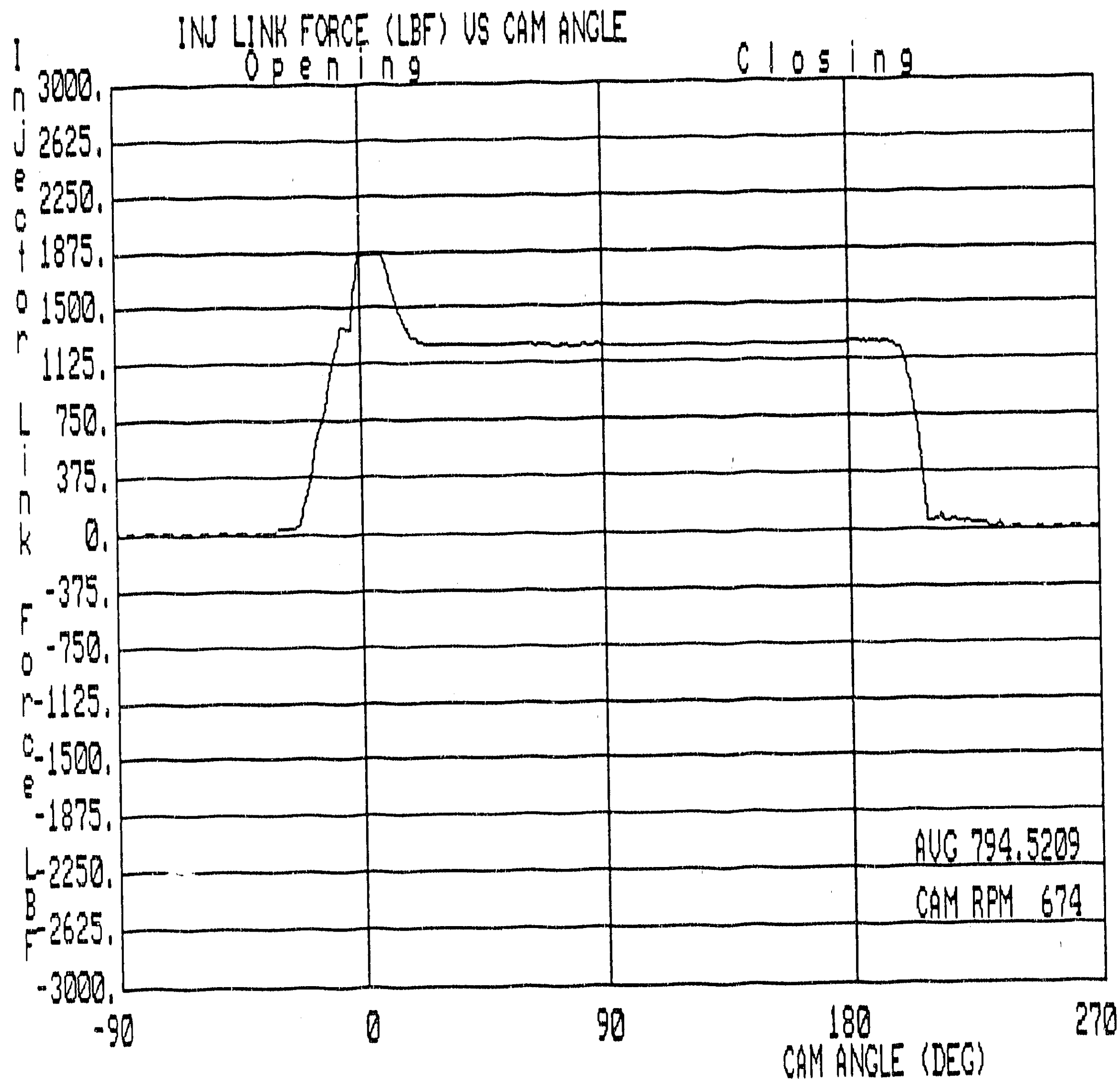
```
      CHARACTER*4 FMT
C
50    WRITE (*,*) QUESTION,IANSWER
      READ (*,'(A9)',ERR=100) STR
      IF (STR .NE. '         ') READ (STR,'(BN,I9)',ERR=100) IANSWER
      RETURN
100   WRITE (*,*) '*** Input error.  Please re-enter data ***'
      GOTO 50
      STOP
      END

      SUBROUTINE GETYN(QUESTION,ANSWER)
C  THIS SUBROUTINE READS IN THE ANSWER TO A YES/NO QUESTION
      CHARACTER*40 QUESTION
      CHARACTER*1 YN
C
50    WRITE (*,*) QUESTION,'(Y/N)'
      READ (*,'(A1)',ERR=100) YN
      IF (YN.EQ.'Y' .OR. YN.EQ.'y') THEN
       ANSWER=1
       RETURN
      ENDIF
      IF (YN.EQ.'N' .OR. YN.EQ.'n') THEN
       ANSWER=0
       RETURN
      ENDIF
100   WRITE (*,*) '*** Input error.  Please re-enter data ***'
      GOTO 50
      STOP
      END
```

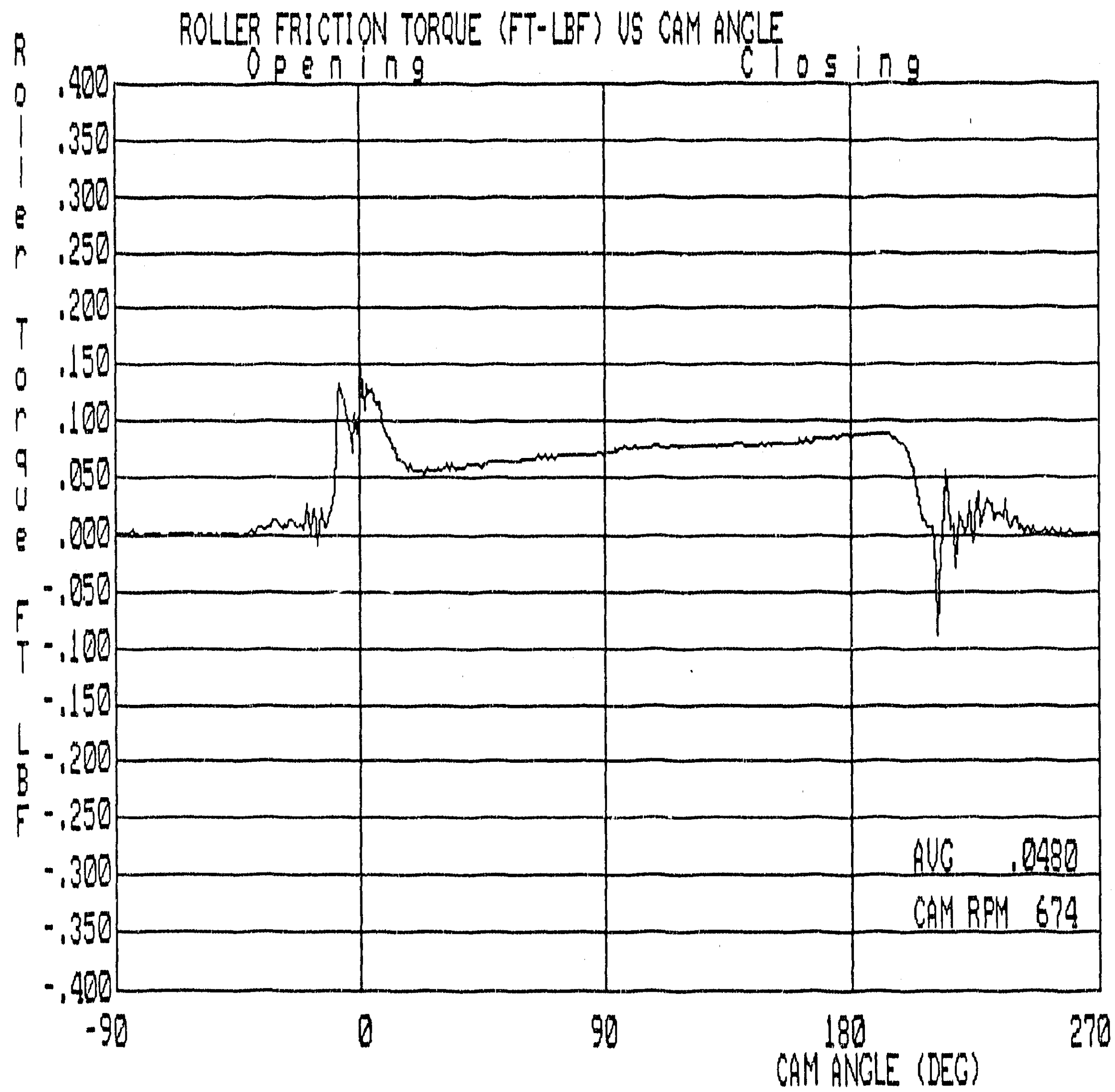
# APPENDIX 3

# TEST 9102102 DATA PLOTS

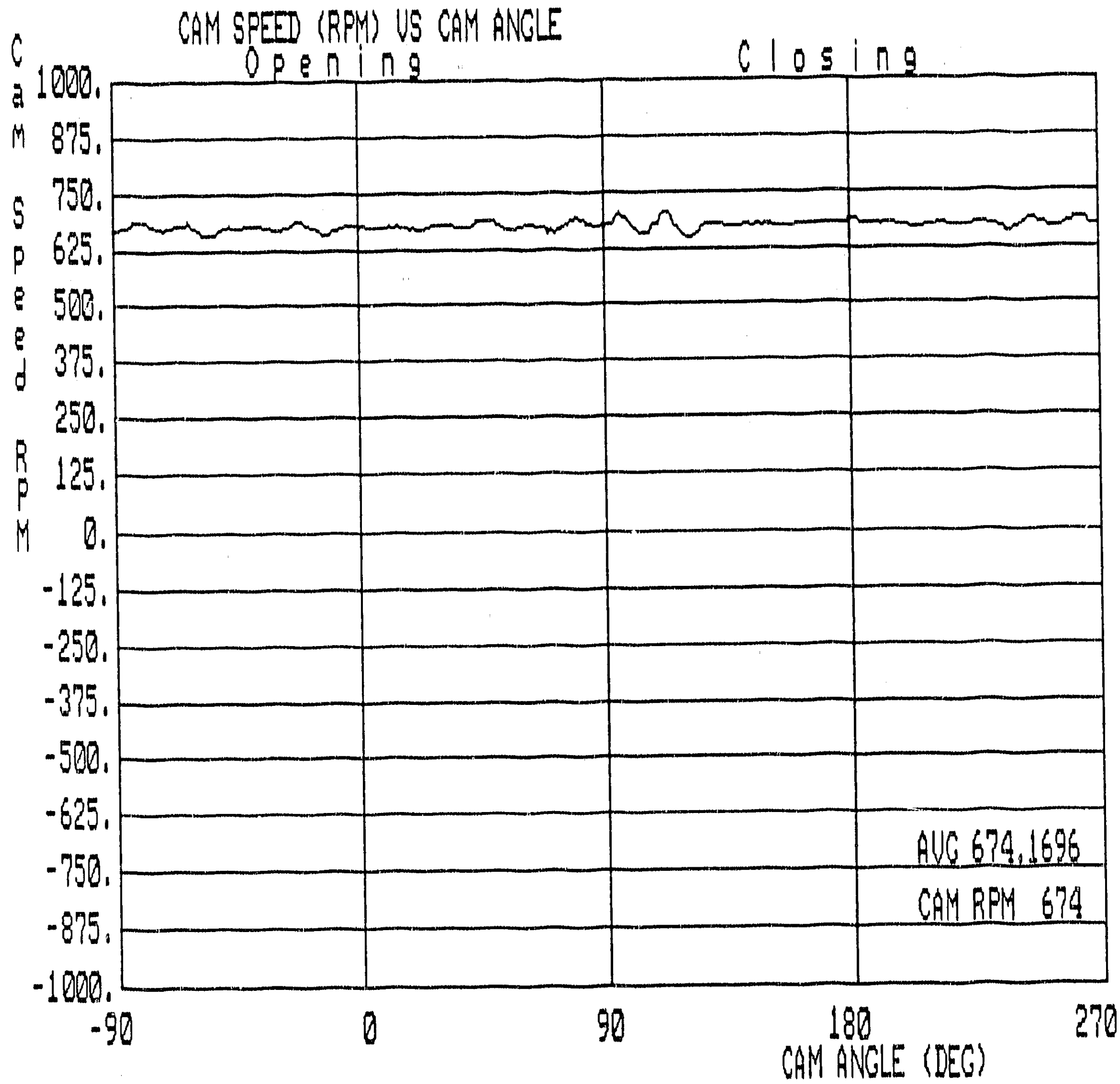
# MAXIMUM TORQUE CONDITIONS

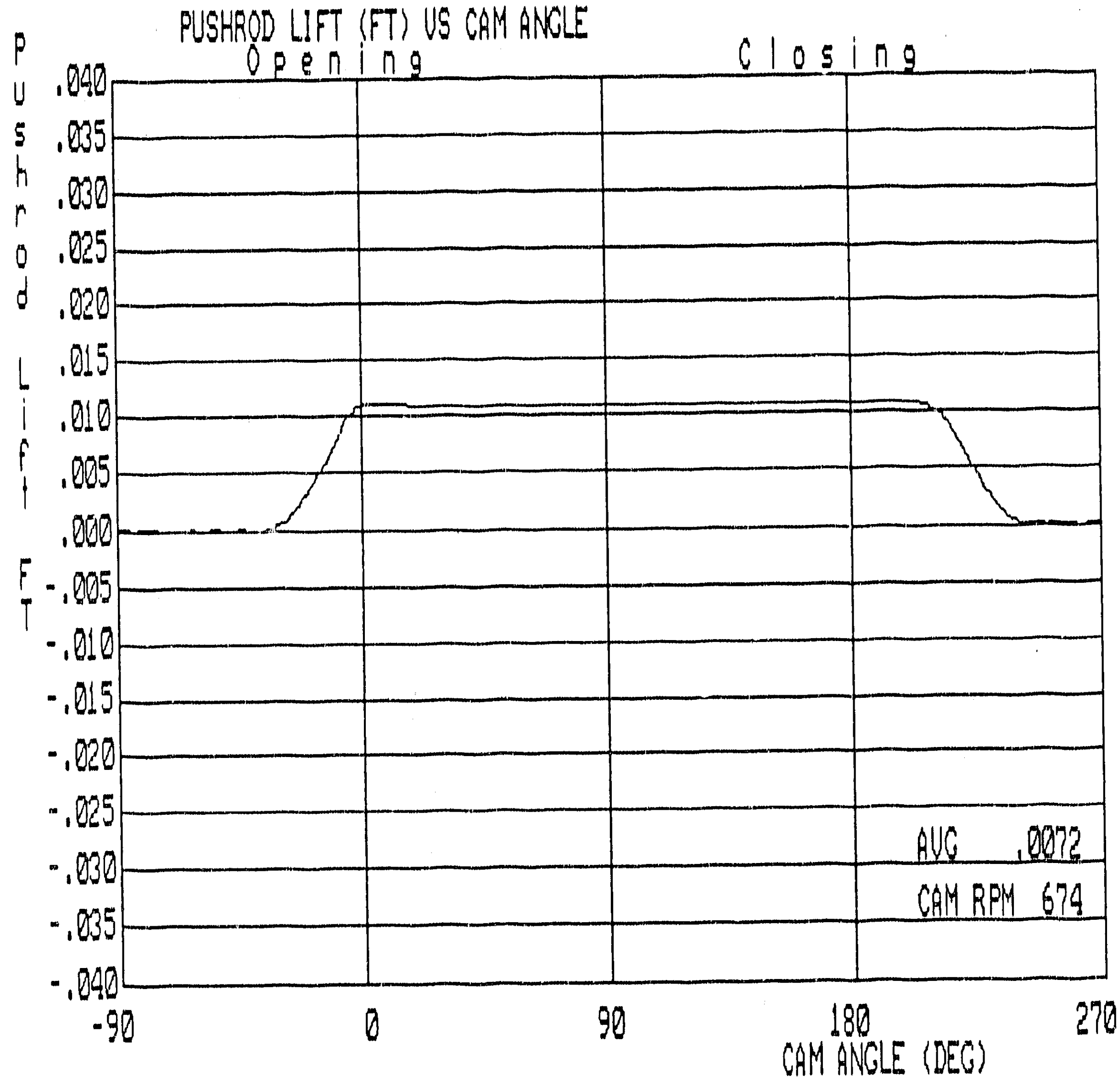

PUSHROD FORCE (LBF) VS CAM ANGLE
Opening
Closing
Pushrod Force LBF
3000.
2625.
2250.
1875.
1500.
1125.
750.
375.
0.
-375.
-750.
-1125.
-1500.
-1875.
-2250.
-2625.
-3000.
-90
0
90
180
270
CAM ANGLE (DEG)
AVG XXXXXXXX
CAM RPM 674

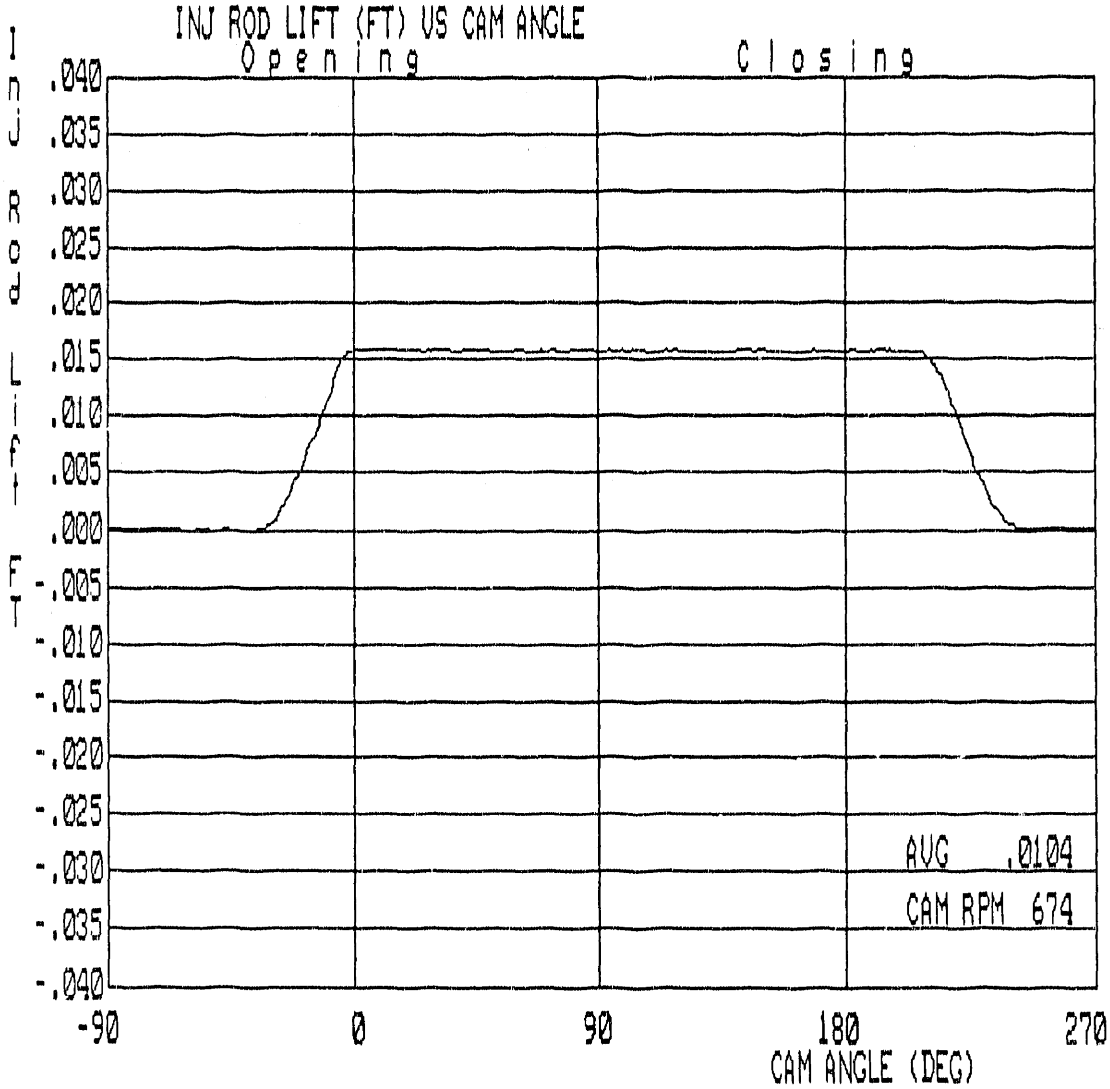

INJ LINK FORCE (LBF) VS CAM ANGLE
Opening
Closing
Injector Link Force LBF
3000.
2625.
2250.
1875.
1500.
1125.
750.
375.
0.
-375.
-750.
-1125.
-1500.
-1875.
-2250.
-2625.
-3000.
-90
0
90
180
270
CAM ANGLE (DEG)
AVG 794.5209
CAM RPM 674

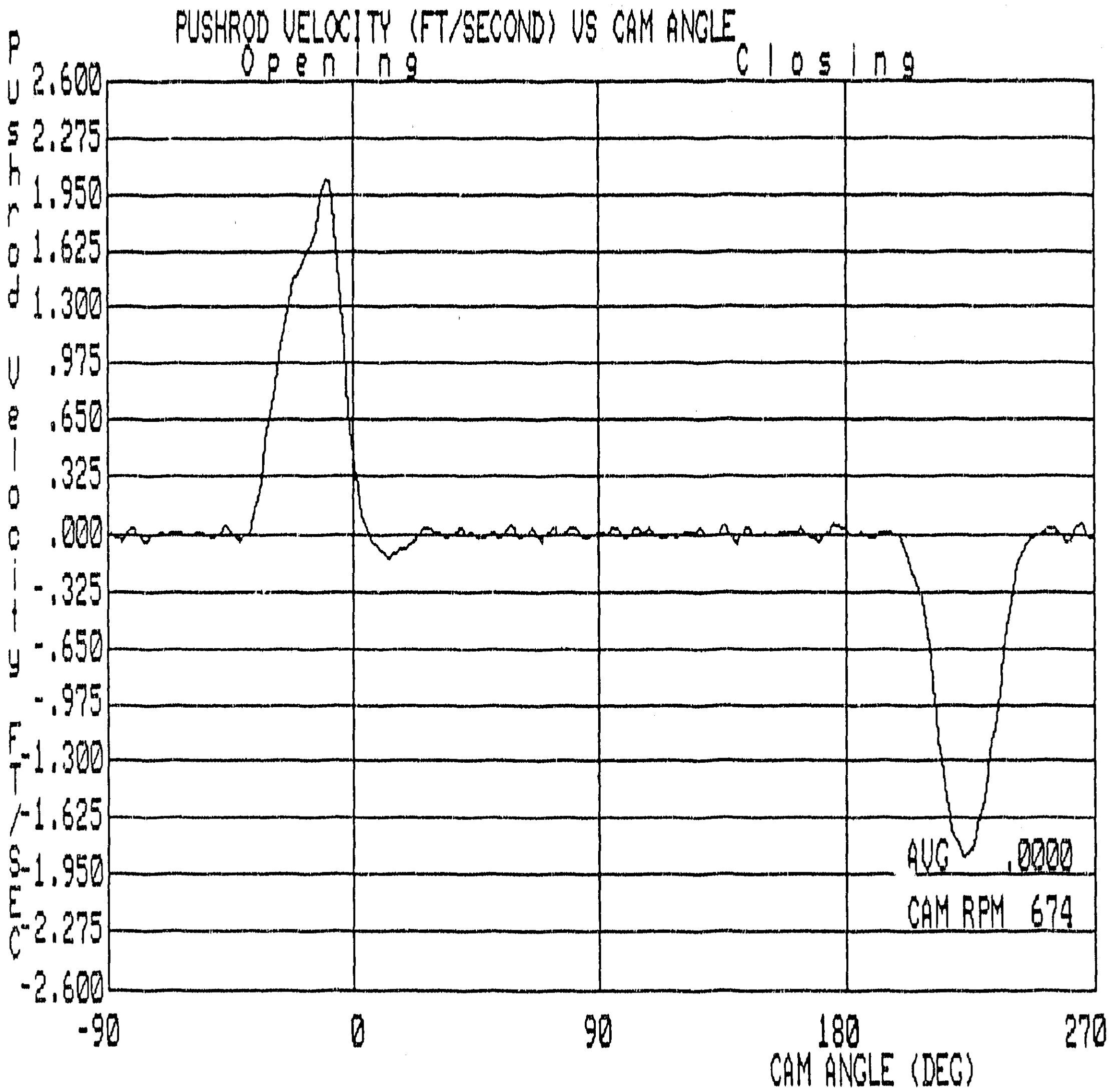


ROLLER FRICTION TORQUE (FT-LBF) VS CAM ANGLE
Opening
Closing
Roller Torque FT LBF
.400
.350
.300
.250
.200
.150
.100
.050
.000
-.050
-.100
-.150
-.200
-.250
-.300
-.350
-.400
AVG .0480
CAM RPM 674
-90
0
90
180
270
CAM ANGLE (DEG)

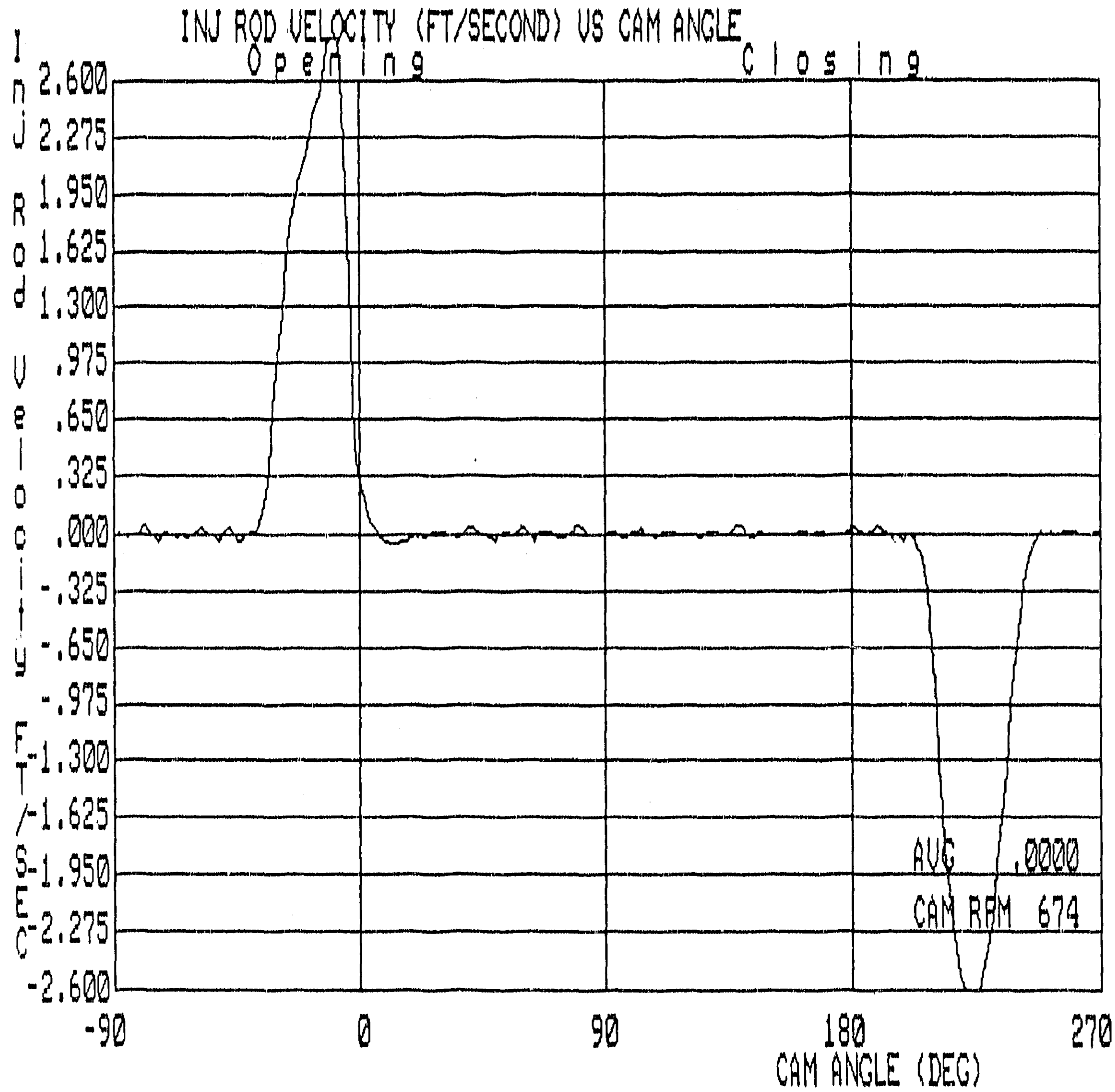

CAM SPEED (RPM) VS CAM ANGLE
Opening
Closing
Cam Speed RPM
1000.
875.
750.
625.
500.
375.
250.
125.
0.
-125.
-250.
-375.
-500.
-625.
-750.
-875.
-1000.
-90
0
90
180
270
CAM ANGLE (DEG)
AVG 674.1696
CAM RPM 674

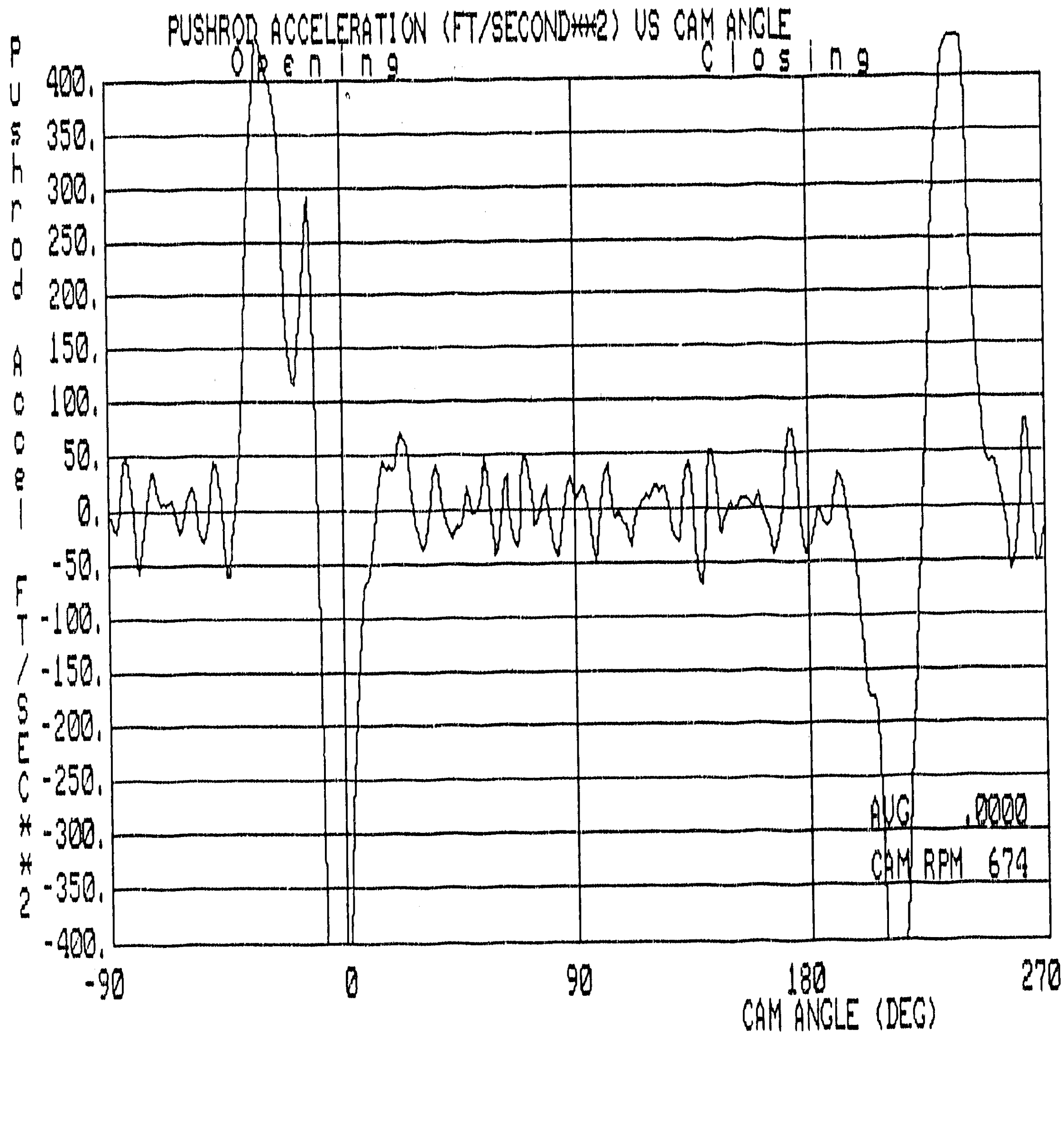

PUSHROD LIFT (FT) VS CAM ANGLE
Opening
Closing
Pushrod Lift FT
.040
.035
.030
.025
.020
.015
.010
.005
.000
-.005
-.010
-.015
-.020
-.025
-.030
-.035
-.040
-90
0
90
180
270
CAM ANGLE (DEG)
AVG .0072
CAM RPM 674


INJ ROD LIFT (FT) VS CAM ANGLE
Opening
Closing
Inj Rod Lift FT
.040
.035
.030
.025
.020
.015
.010
.005
.000
-.005
-.010
-.015
-.020
-.025
-.030
-.035
-.040
-90
0
90
180
270
CAM ANGLE (DEG)
AVG .0104
CAM RPM 674

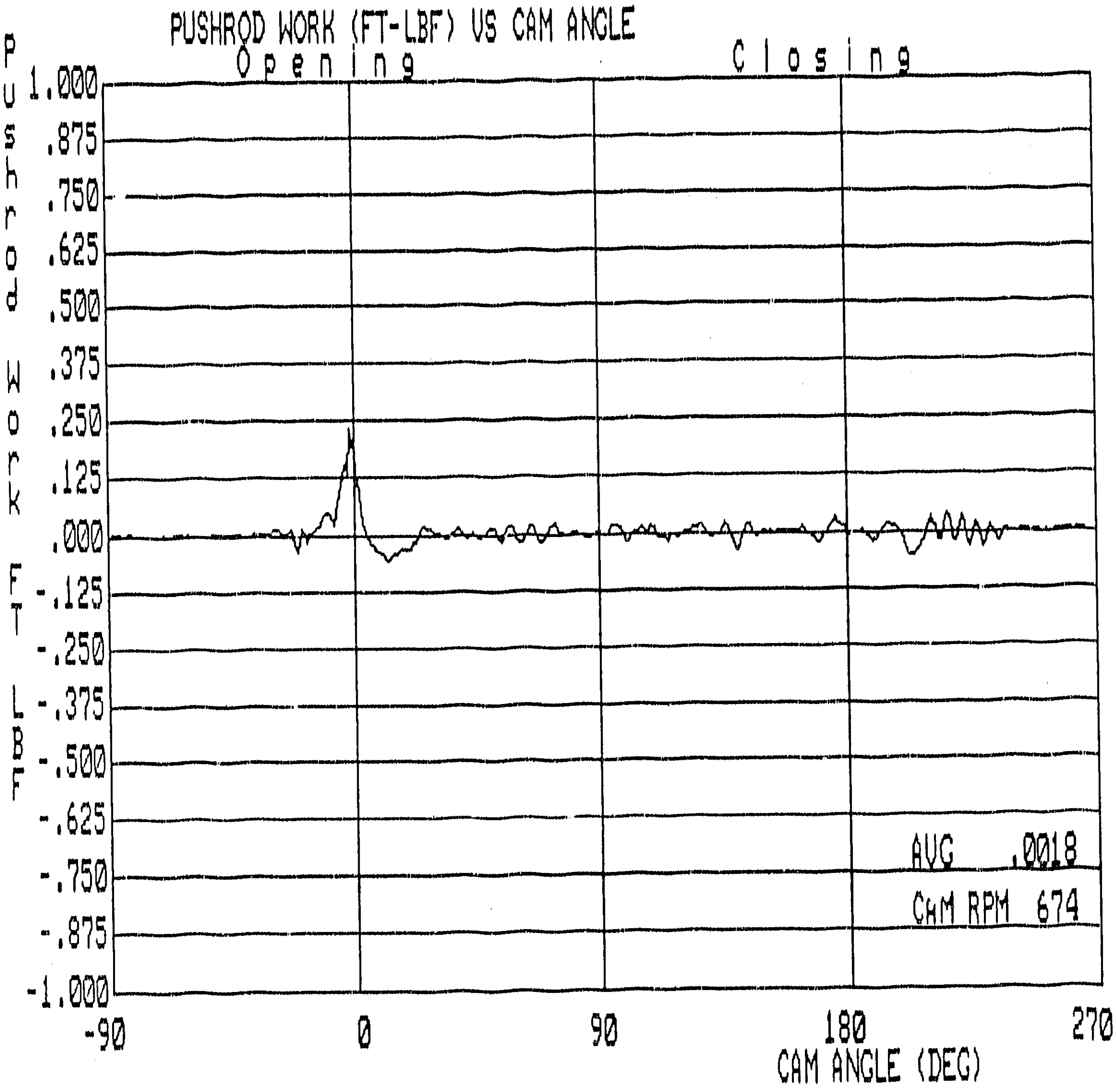


PUSHROD VELOCITY (FT/SECOND) VS CAM ANGLE
Opening
Closing
Pushrod Velocity FT/SEC
2.600
2.275
1.950
1.625
1.300
.975
.650
.325
.000
-.325
-.650
-.975
-1.300
-1.625
-1.950
-2.275
-2.600
-90
0
90
180
270
CAM ANGLE (DEG)
AVG .0000
CAM RPM 674

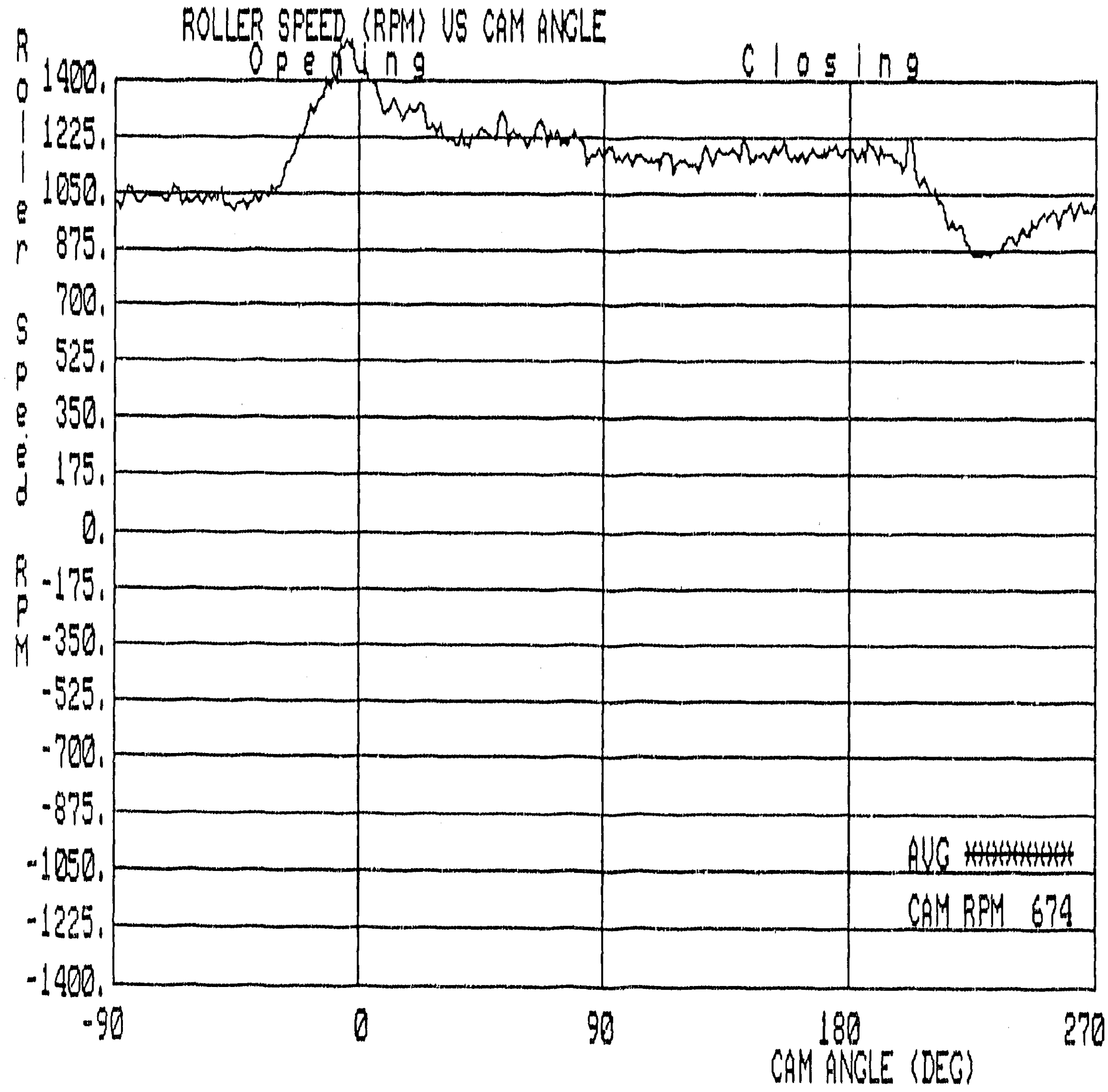

INJ ROD VELOCITY (FT/SECOND) VS CAM ANGLE
Opening
Closing
Inj Rod Velocity FT/SEC
2.600
2.275
1.950
1.625
1.300
.975
.650
.325
.000
-.325
-.650
-.975
-1.300
-1.625
-1.950
-2.275
-2.600
AVG .0000
CAM RPM 674
-90
0
90
180
270
CAM ANGLE (DEG)

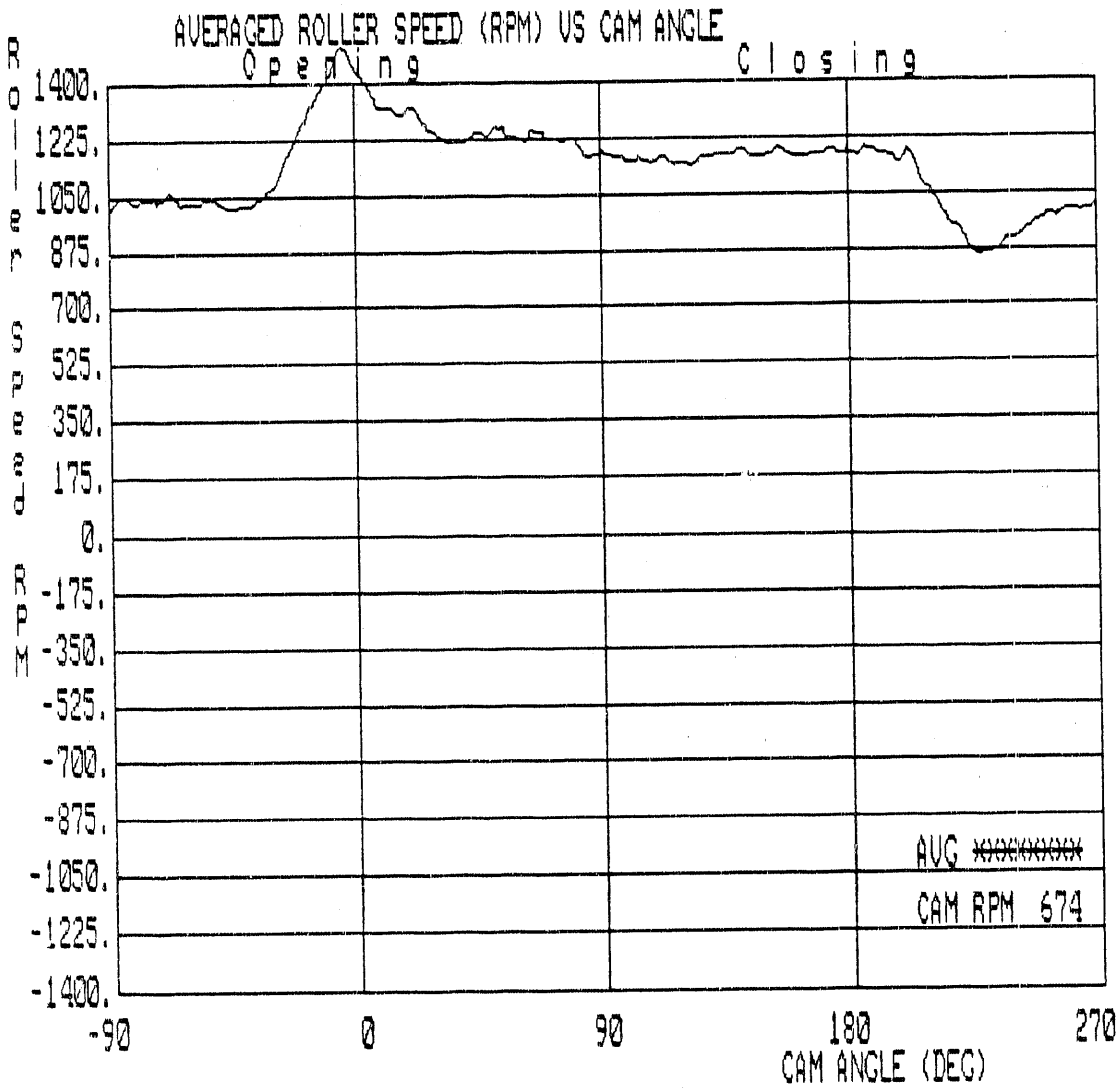


PUSHROD ACCELERATION (FT/SECOND**2) VS CAM ANGLE
Opening
Closing
Pushrod Accel FT/SEC**2
400.
350.
300.
250.
200.
150.
100.
50.
0.
-50.
-100.
-150.
-200.
-250.
-300.
-350.
-400.
-90
0
90
180
270
CAM ANGLE (DEG)
AVG .0000
CAM RPM 674

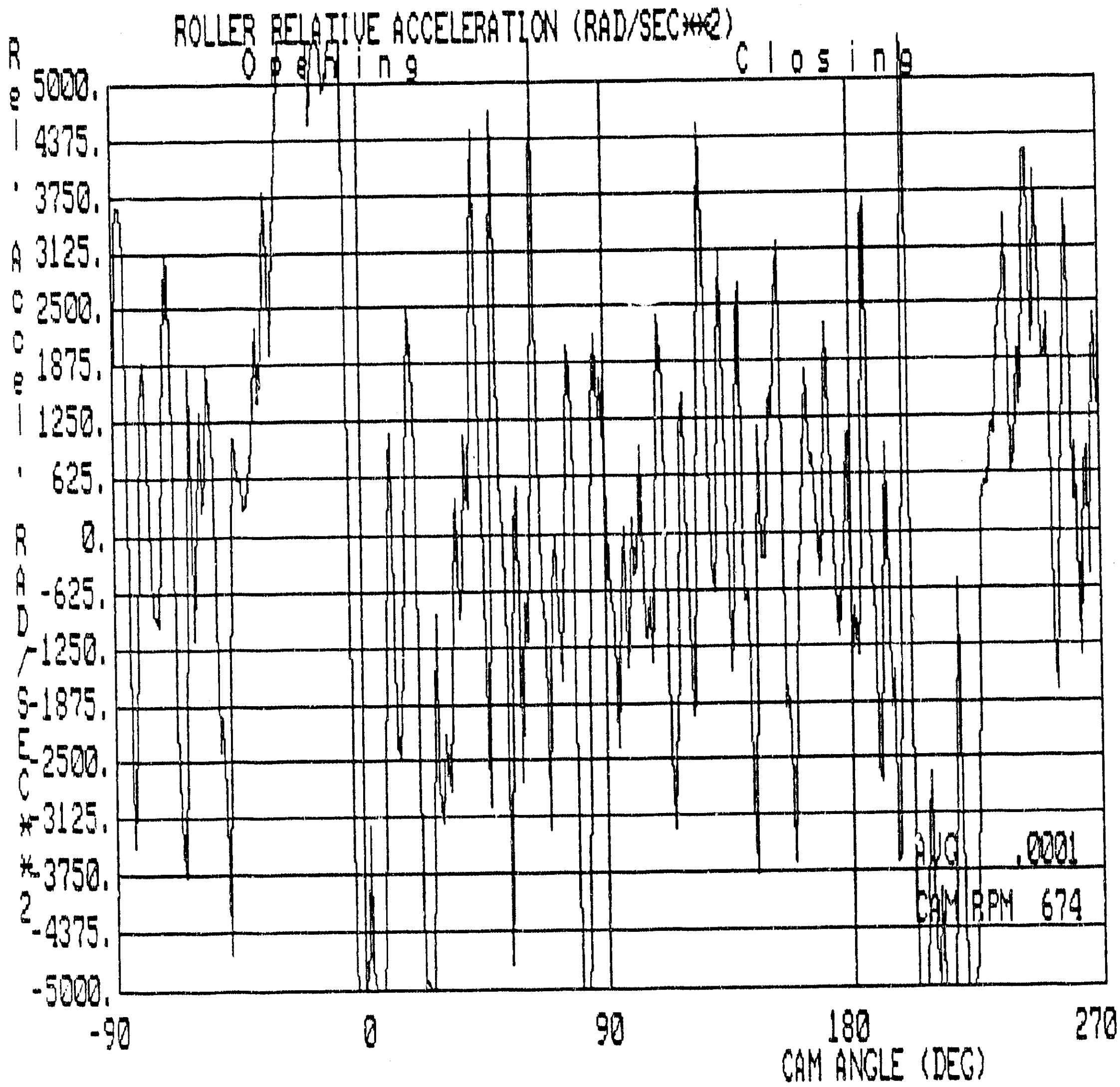


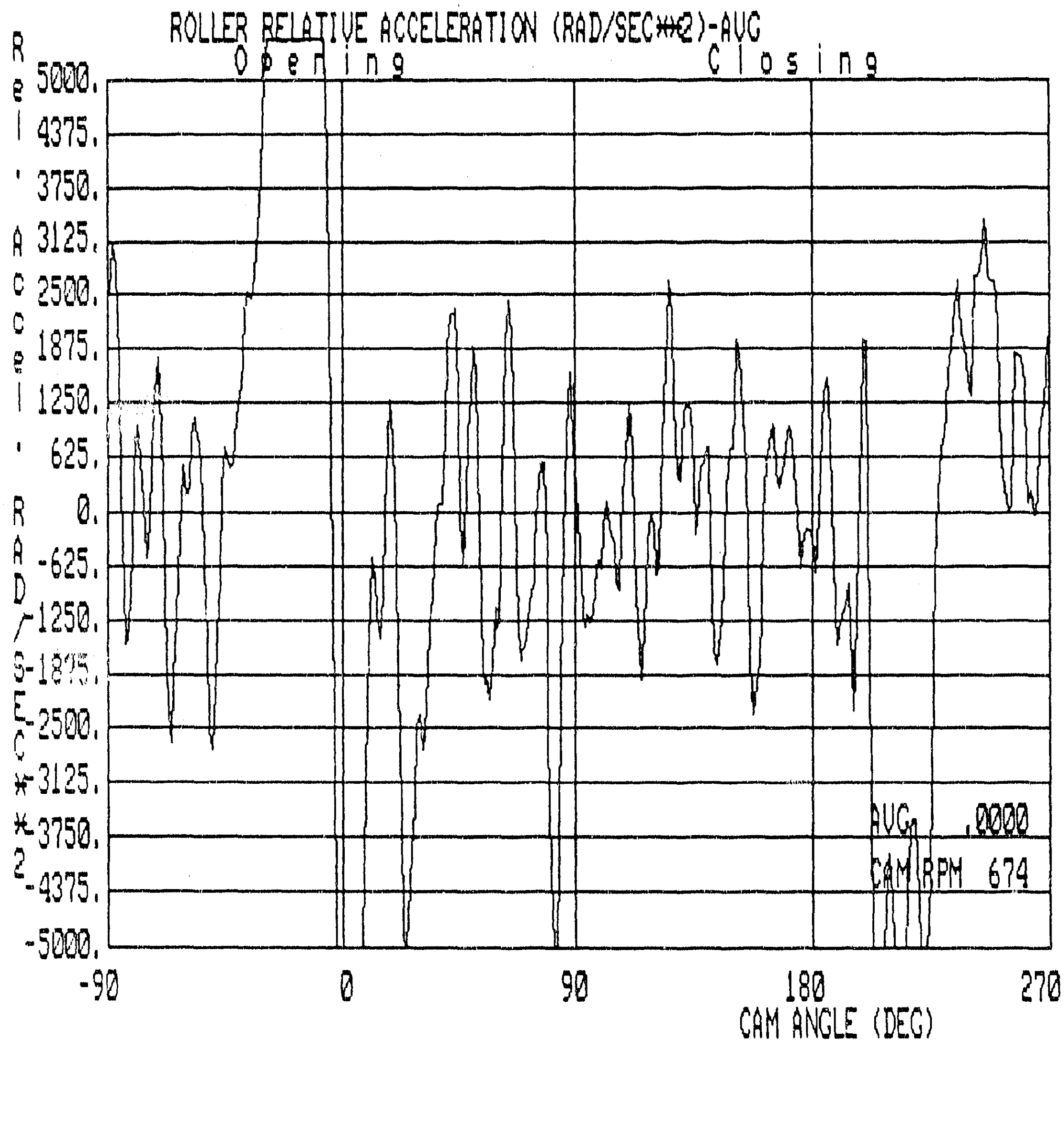
PUSHROD WORK (FT-LBF) VS CAM ANGLE
Opening
Closing
Pushrod Work FT LBF
1.000
.875
.750
.625
.500
.375
.250
.125
.000
-.125
-.250
-.375
-.500
-.625
-.750
-.875
-1.000
-90
0
90
180
270
CAM ANGLE (DEG)
AVG .0018
CAM RPM 674


ROLLER SPEED (RPM) VS CAM ANGLE
Opening
Closing
Roller Speed RPM
1400.
1225.
1050.
875.
700.
525.
350.
175.
0.
-175.
-350.
-525.
-700.
-875.
-1050.
-1225.
-1400.
-90
0
90
180
270
CAM ANGLE (DEG)
AVG XXXXXXXX
CAM RPM 674

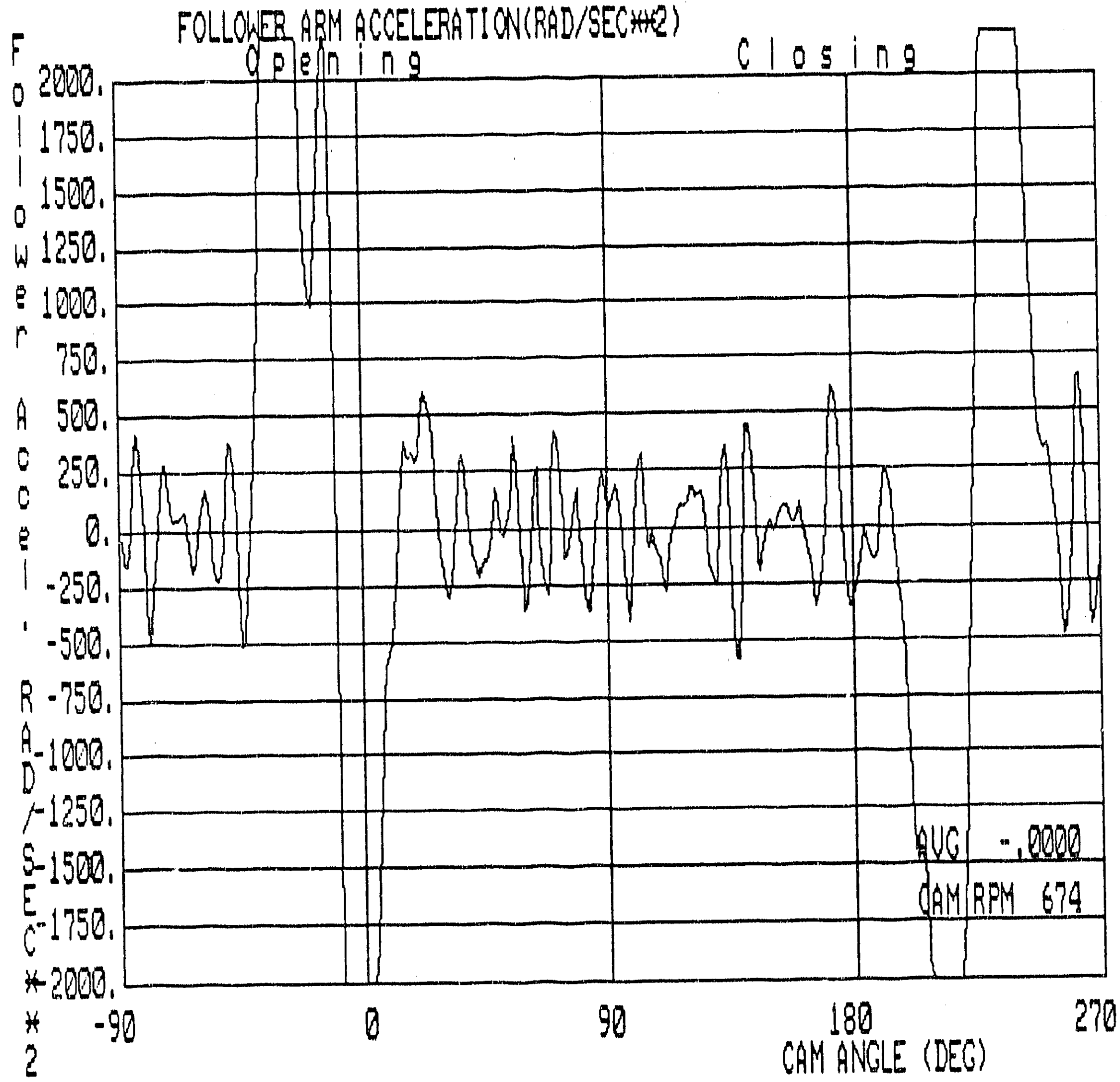


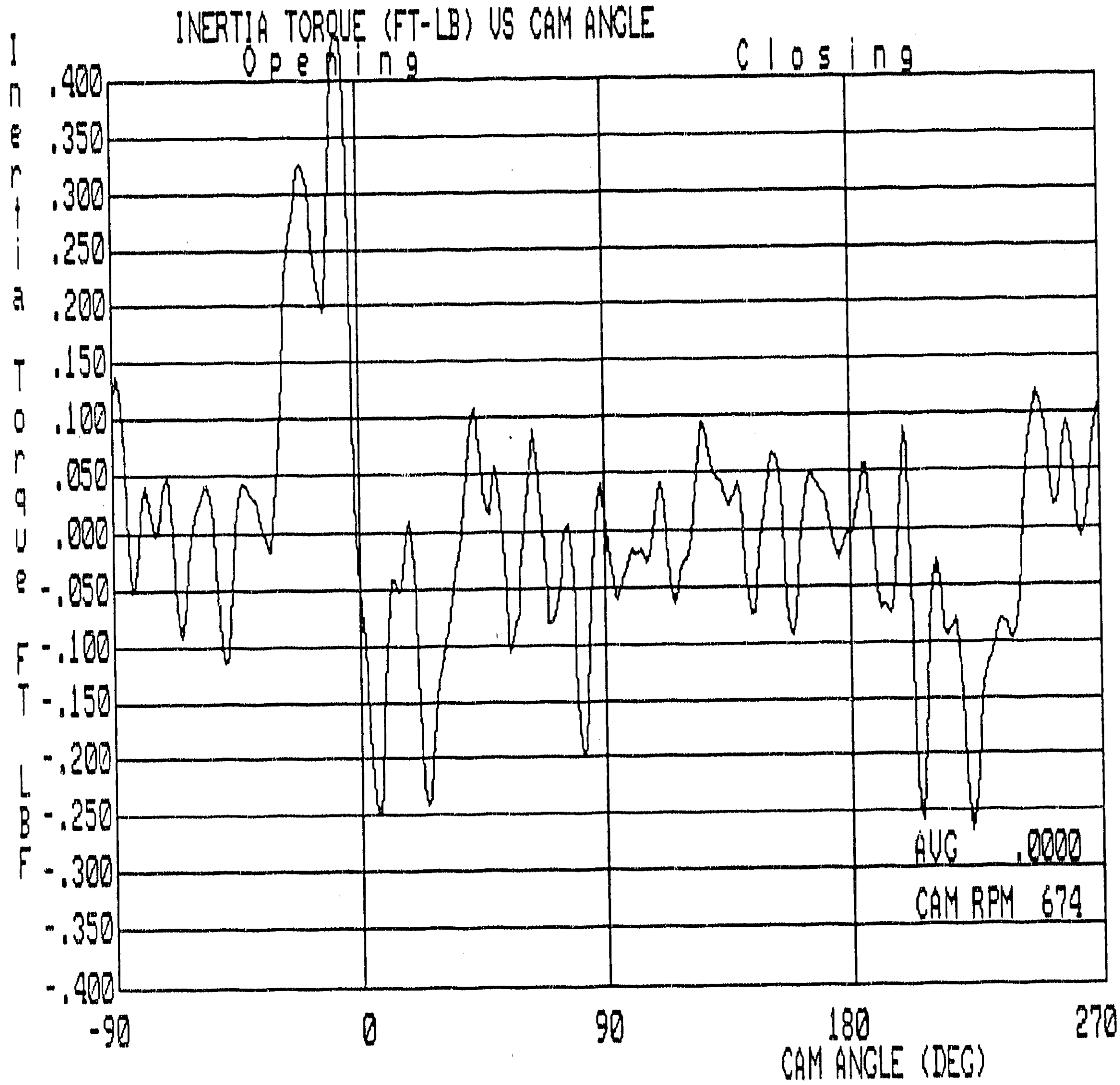
AVERAGED ROLLER SPEED (RPM) VS CAM ANGLE
Opening
Closing
Roller Speed RPM
1400.
1225.
1050.
875.
700.
525.
350.
175.
0.
-175.
-350.
-525.
-700.
-875.
-1050.
-1225.
-1400.
AVG XXXXXXXX
CAM RPM 674
-90
0
90
180
270
CAM ANGLE (DEG)

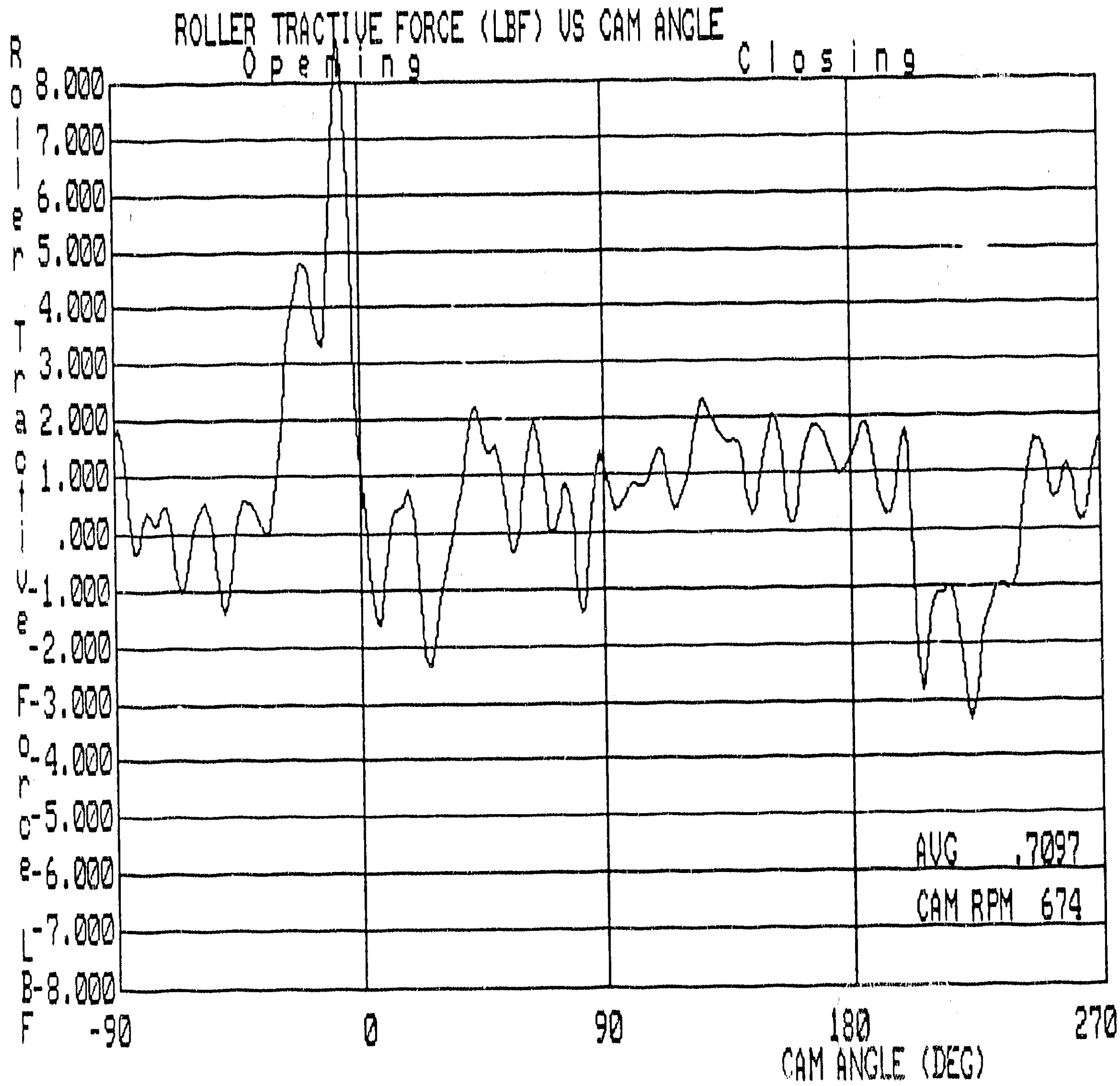

ROLLER RELATIVE ACCELERATION (RAD/SEC**2)
Opening
Closing
Rel. Accel. RAD/SEC**2
5000.
4375.
3750.
3125.
2500.
1875.
1250.
625.
0.
-625.
-1250.
-1875.
-2500.
-3125.
-3750.
-4375.
-5000.
-90
0
90
180
270
CAM ANGLE (DEG)
AVG .0001
CAM RPM 674

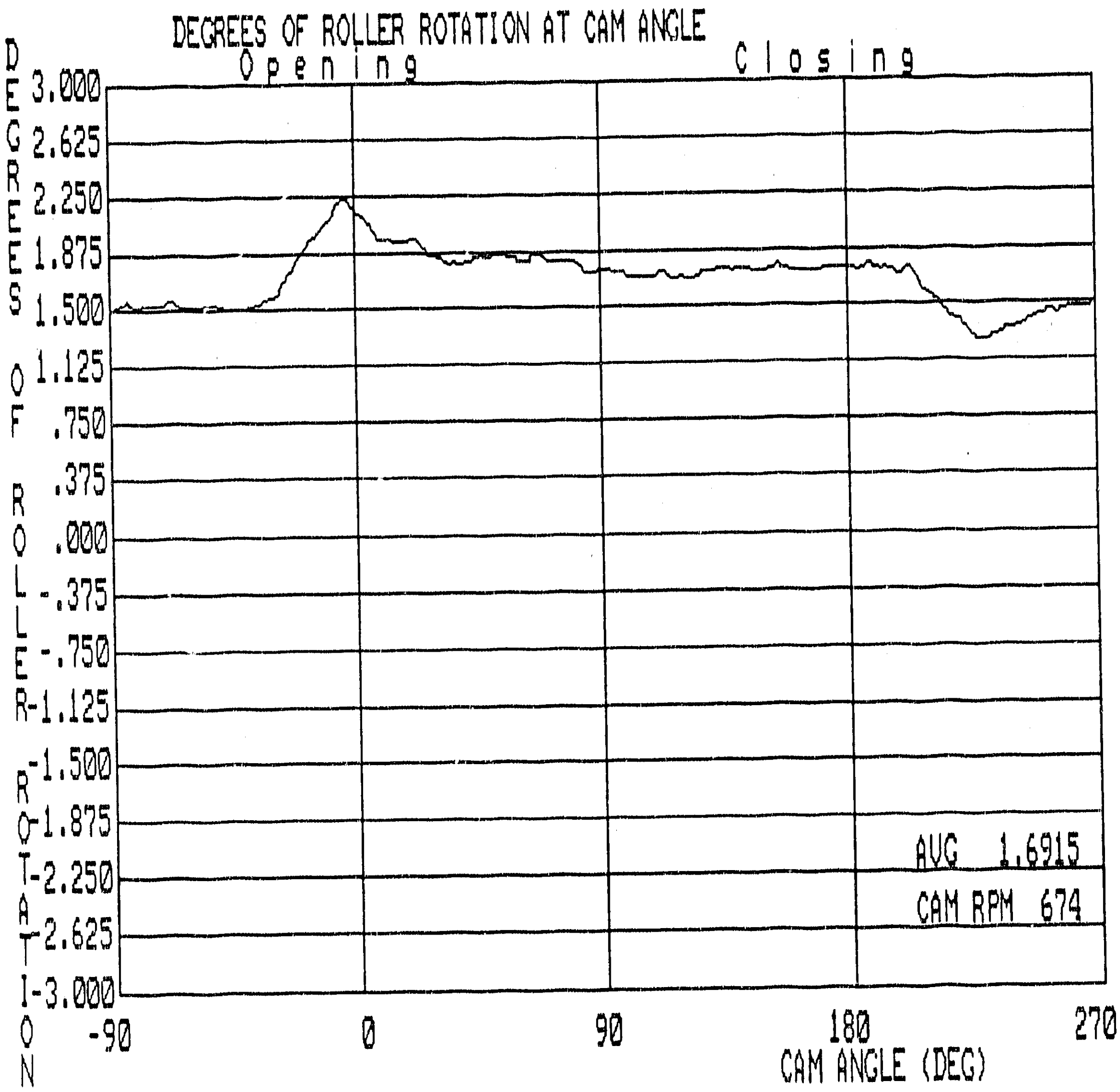


ROLLER RELATIVE ACCELERATION (RAD/SEC**2)-AVG
Opening
Closing
Rel. Accel. RAD/SEC**2
5000.
4375.
3750.
3125.
2500.
1875.
1250.
625.
0.
-625.
-1250.
-1875.
-2500.
-3125.
-3750.
-4375.
-5000.
-90
0
90
180
270
CAM ANGLE (DEG)
AVG .0000
CAM RPM 674


FOLLOWER ARM ACCELERATION(RAD/SEC**2)
Opening
Closing
Follower Accel. RAD/SEC**2
2000.
1750.
1500.
1250.
1000.
750.
500.
250.
0.
-250.
-500.
-750.
-1000.
-1250.
-1500.
-1750.
-2000.
-90
0
90
180
270
CAM ANGLE (DEG)
AVG -.0000
CAM RPM 674


INERTIA TORQUE (FT-LB) VS CAM ANGLE
Opening
Closing
Inertia Torque FT LBF
.400
.350
.300
.250
.200
.150
.100
.050
.000
-.050
-.100
-.150
-.200
-.250
-.300
-.350
-.400
-90
0
90
180
270
CAM ANGLE (DEG)
AVG .0000
CAM RPM 674


ROLLER TRACTIVE FORCE (LBF) VS CAM ANGLE
Opening
Closing
Roller Tractive Force LBF
8.000
7.000
6.000
5.000
4.000
3.000
2.000
1.000
.000
-1.000
-2.000
-3.000
-4.000
-5.000
-6.000
-7.000
-8.000
-90
0
90
180
270
CAM ANGLE (DEG)
AVG .7097
CAM RPM 674


DEGREES OF ROLLER ROTATION AT CAM ANGLE
Opening
Closing
DEGREES OF ROLLER ROTATION
3.000
2.625
2.250
1.875
1.500
1.125
.750
.375
.000
-.375
-.750
-1.125
-1.500
-1.875
-2.250
-2.625
-3.000
-90
0
90
180
270
CAM ANGLE (DEG)
AVG 1.6915
CAM RPM 674

# END

# DATE FILMED

5/13/92