R-TR-77-017

20

A MATHEMATICAL MODEL OF THE 30 MM ADVANCED MEDIUM CALIBER WEAPON SYSTEM (AMCAWS-30)

MICHAEL R. KANE

APRIL 1977

FINAL REPORT

U.S. ARMY ARMAMENT COMMAND

SMALL CALIBER WEAPONS SYSTEMS LABORATORY

Distribution Statement

Approved for public release; distribution unlimited.

Disclaimer:

The findings of this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

DISPOSITION INSTRUCTIONS:

Destroy this report when it is no longer needed.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER R-TR-77-017	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A MATHEMATICAL MODEL OF THE 30 MM ADVANCED MEDIUM CALIBER WEAPON SYSTEM (AMCAWS-30)		5. TYPE OF REPORT & PERIOD COVERED Final rept.
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Michael R. Kane		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Commander, Rock Island Arsenal DRDAR-SCS Rock Island, IL 61201		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 1W662603AH78 01.01
11. CONTROLLING OFFICE NAME AND ADDRESS Same as 9		12. REPORT DATE April 77
		13. NUMBER OF PAGES 265
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) RIA-R-TR-77-017		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for Public Release; Distribution Unlimited.

259p.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Report fulfills DARCOM thesis requirement for the author's DARCOM sponsored long-term CAD-E training at the University of Michigan.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

1. High impulse
2. Externally powered
3. Mathematical model
4. Numerical integration
5. d'Alembert force method
6. Computer program
7. Weapon model
8. FORTRAN

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A mathematical model for the AMCAWS-30MM weapon is developed using the generalized d'Alembert force equations. The development of the one degree of freedom differential equation of motion for the weapon is shown. The equation accounts for operations including feed, eject, chamber locking, round crush-up, chamber translation, face cam rotation, and drum cam rotation. The resultant equation is numerically integrated to obtain the time response of position, velocity, acceleration, and force for the major components. The solution is based on the known drive motor characteristics.

DD FORM 1473, 1 JAN 73 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

408 247

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

LB

The numerical integration is done using the HPCG subroutine out of the IBM SSP Math Library. The total program is modularized and inclusions of additional parts or design changes in the weapon can be incorporate without extensive revision of the program. The program is written in FORTRAN.

A MATHEMATICAL MODEL OF THE 30 MM
ADVANCED MEDIUM CALIBER WEAPON SYSTEM
(AMCAWS-30)*

by

Michael R. Kane

111

(This page is blank)

iv

ACKNOWLEDGMENT

This work was undertaken to meet the DARCOM project requirement while the author was engaged in DARCOM sponsored long term CAD-E training at the University of Michigan. The project was completed after the author returned to the advanced concepts group of Gen. T. J. Rodman Laboratory at Rock Island Arsenal.

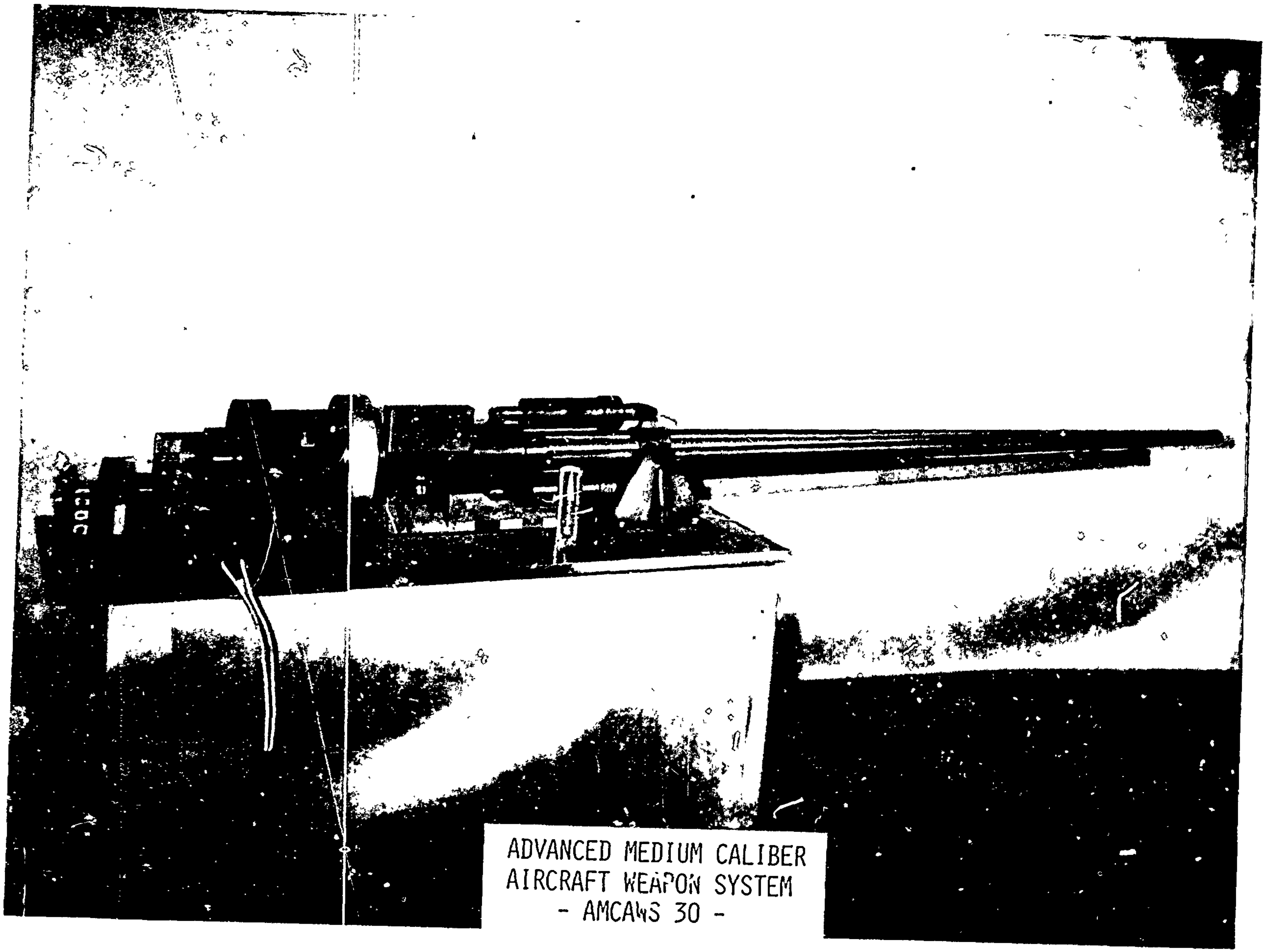
Prof. M. A. Chace (Chairman, CAD-E Department) of the University of Michigan was the faculty advisor for the project. His guidance was invaluable.

ABSTRACT

A mathematical model for the AMCAWS-30MM weapon is developed using the generalized d'Alembert force equations. The development of the one degree of freedom differential equation of motion for the weapon is shown. The equation accounts for operations including feed, eject, chamber locking, round crush-up, chamber translation, face cam rotation, and drum cam rotation. The resultant equation is numerically integrated to obtain the time response of position, velocity, acceleration, and force for the major components. The solution is based on the known drive motor characteristics.

The numerical integration is done using the HPCG subroutine out of the IBM SSP Math Library. The total program is modularized and inclusions of additional parts or design changes in the weapon can be incorporated without extensive revision of the program. The program is written in FORTRAN.

*The AMCAWS-30MM weapon is currently under development in the Advanced Concepts Group, Aircraft and Air Defense Weapons Systems Directorate, General Thomas J. Rodman Laboratory, Rock Island, Ill. (SARRI-LW-A). AMCAWS is a 30 milimeter single barrel weapon that utilizes an aluminum cased, fully telescoped, and consolidated propellant round with ballistic characteristics slightly better than GAU-8 rounds. The prototype weapon fires ten round bursts at a nominal rate of 120 spm. The second prototype weapon has a nominal rate in excess of 400 spm.



ADVANCED MEDIUM CALIBER
AIRCRAFT WEAPON SYSTEM
- AMCAWS 30 -

TABLE OF CONTENTS

1. OBJECTIVE

The first prototype AMCAWS-30 weapon capable of operation in an automatic mode has existed since 2 May 74. That prototype has since shown a high degree of reliability in many ten round bursts and other lesser firing schedules.

Those first automatic firings were the result of a design process historically similar to the design of other medium caliber weapons. A new concept, approach, or need leads the designer to develop the design, using tools generally available to the draftsman (assemblies, sections, blowups, etc.). Parts are sized by a coarse static force analysis or by the intuition of the designer. Operational or dynamic forces are not investigated extensively. The drive motor, for example, on the AMCAWS was chosen because it was available and it was felt that it was "big enough". Throughout the complete design cycle including the initial layouts, a few part redesigns, and a very circumspect assembly there were several questions that begged answers. The questions included:

(1) What is the complete position description of all the major components during a firing cycle?

(2) What is the response of the weapon as a whole to different drive motors?

(3) What are the forces between parts during weapon operation?

(4) What is the effect on weapon performance when parts are redesigned?

The AMCAWS-30 mathematical model provides the ability to answer these questions. The model is one degree of freedom and accounts for inertial, translational, and dissipative forces. Modeled components include the feed, eject, chamber, lock, and the drum/face cams. The model employs generalized d'Alembert forces to develop the differential equation of motion and uses functional relationships developed by a preliminary program to establish a component's positional dependence on a single coordinate, the motor input angle.

The purpose of this report is threefold. First, it is intended to document the computer programs that make up the AMCAWS mathematical model package. The programs are discussed and listed in the appendices and contain excellent internal documentation. Second, the report is to describe the operational characteristics of the weapon. A brief ammunition description is in the next section and a detailed ammunition description can be found in the ammunition final report (1)* Third, the report will document the actual modeling procedure.

The model package discussed in this report treats only the first prototype weapon. The procedure employed, is, however, general and can be applied to a wide variety of weapon systems and subsystems.

*Numbers in parentheses designate References at end of paper.

2. AMMUNITION WEAPON DESCRIPTION

The AMCAWS 30 weapon is the result of an advanced development program for a high performance 30mm automatic weapon system. The weapon has been designed, developed and fabricated in-house at Rock Island Arsenal. The weapon is externally powered and various cams accomplish the feeding, firing, and ejecting of the round. The weapon treated by this report is the first prototype weapon which fires up to ten round bursts at a 121 shots per minute (spm) rate. Parenthetically, the second prototype weapon has a design rate of 500 spm. The second prototype is, as of the date of this report, some 90% fabricated. While the detailed description of the first prototype weapon may seem long and involved, the weapon itself can be characterized as clean and simple. Various cams ensure positive motions and the lateral feeding and ejecting of ammunition permits the absence of some of the more complicated extracting mechanisms used on more conventional weapons.

2.1 AMMUNITION

The AMCAWS 30 ammunition (Figure 1) is an aluminum cased and fully telescoped round. The case is one piece and has a .75 degree radial taper to the base. The main charge is consolidated and concentric about the projectile. The full weight of the round is about 9595 grains (1.37 pounds). Extraction forces after firing are very low. A complete ammunition description can be found in the ammunition final report (1).

2.2 DRUM CAM

The drum cam assembly (Figure 2) has several functions. Torque from the drive unit is transmitted to the weapon through the 151 tooth external gear on the outside diameter of the drum. An internal cam path (Figure 3)

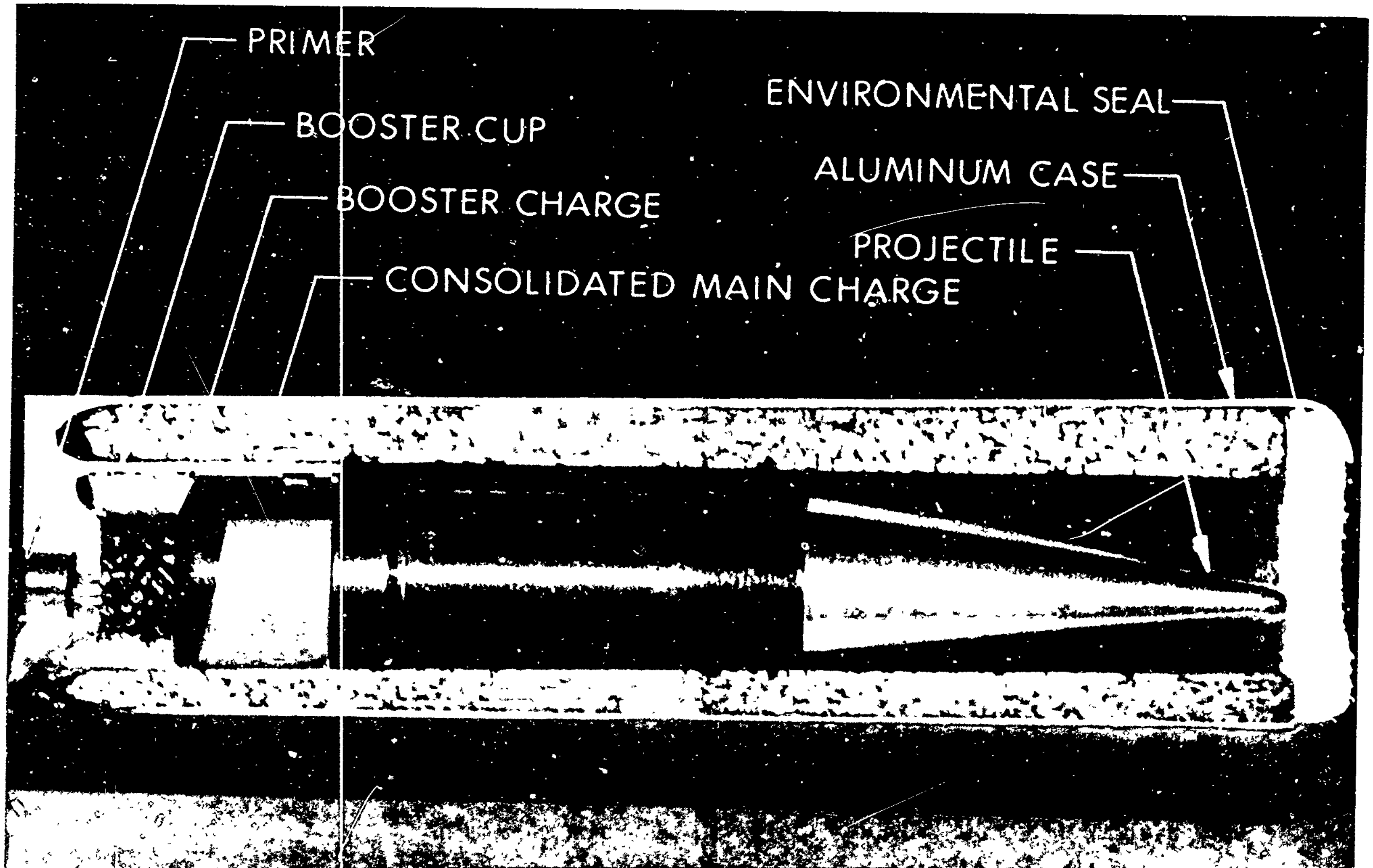


FIGURE 1

AMCAWS-30 CUTAWAY ROUND

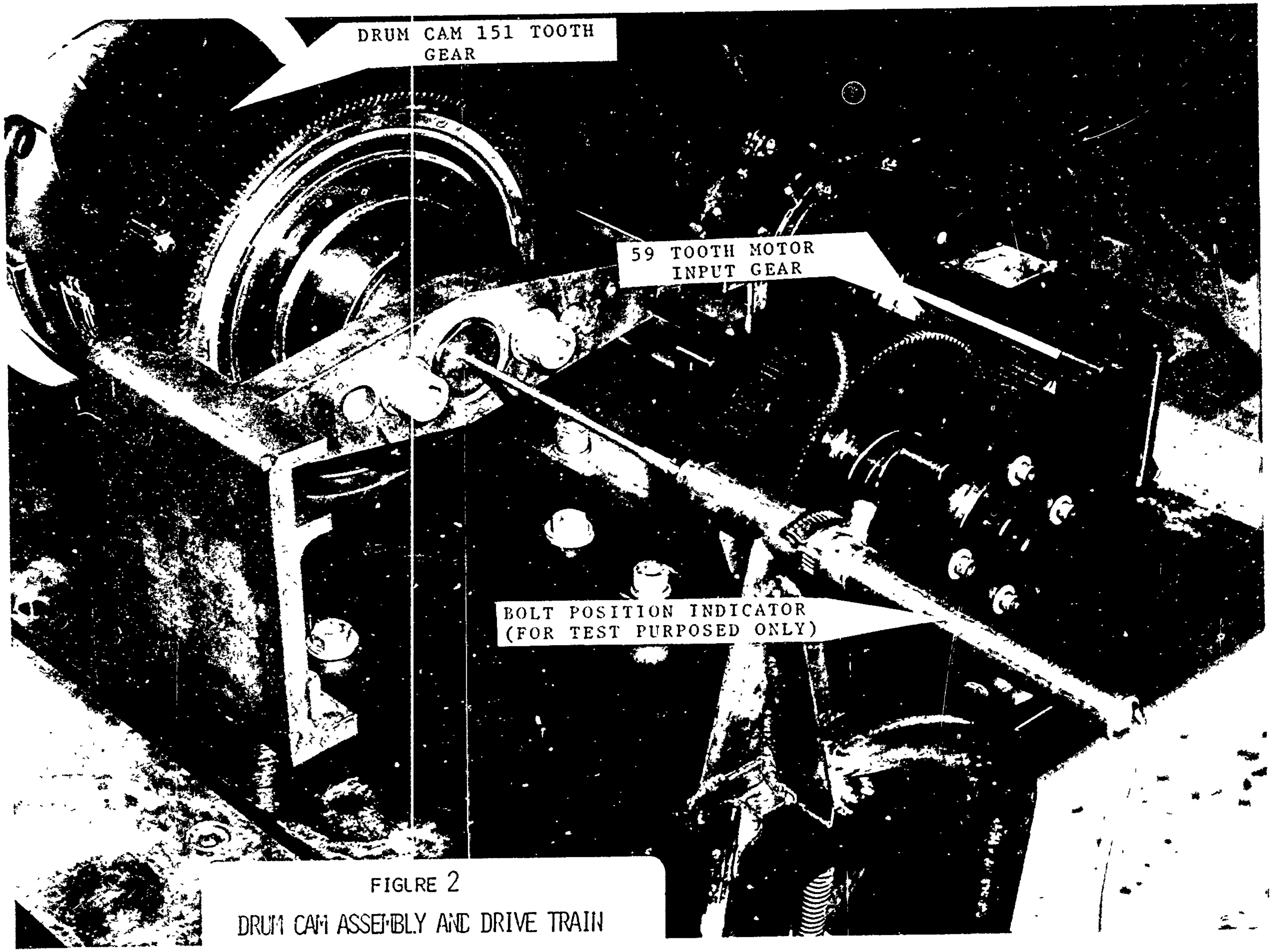


FIGURE 2

DRUM CAM ASSEMBLY AND DRIVE TRAIN

FIGURE 3

INTERNAL DRUM CAM CAMPATH

controls the chamber motion of the weapon. A follower stud cantilevered from the chamber and passing through a receiver slot follows the drum cam path, thus providing the proper chamber motion. A lump cam fixed to the inside diameter of the drum initiates the lock and unlock sequence (Figure 4). The drum cam is concentric to the weapon centerline (Figure 5) and is located over the rear half of the receiver. The face cam is locked to the front of the drum cam at weapon assembly (Figure 4).

2.3 FACE CAM

The face cam (Figure 6,7) has cam paths that control the feed and eject functions for the weapon. The face cam is fixed to the drum cam at weapon assembly and is thus timed to the chamber and locking motions. The feed cam path (Figure A1-10) is followed by the feed rocker arm which transmits rotation to the feeding pawls through the feed shaft. The feed shaft is the center of rotation of the rocker and the pawls and is fixed via supports to the receiver. The total feed mechanism (Figure 8) places a new round of ammunition at the center of the chamber. While the new round is being placed, the fired case is being ejected from the system. The eject campath (Figure A1-9) is followed by the eject rocker arm which transmits rotation to the eject pawl through the eject shaft. The center of rotation of this rocker and pawl is the eject shaft which is fixed to the receiver. The total eject mechanism (Figure 9) must allow the fired round to escape the chamber centerline. This is accomplished by waiting until the chamber is fully rearward and swinging the pawl up into an exposed chamber area so that the fired round can pass under the pawl. As the fired round is passing under the pawl the pawl moves down into a dwell position that will cause a positive stop for the new round being presented. After the stop is made the pawl swings out of the chamber area so that the chamber can be brought forward for a firing.

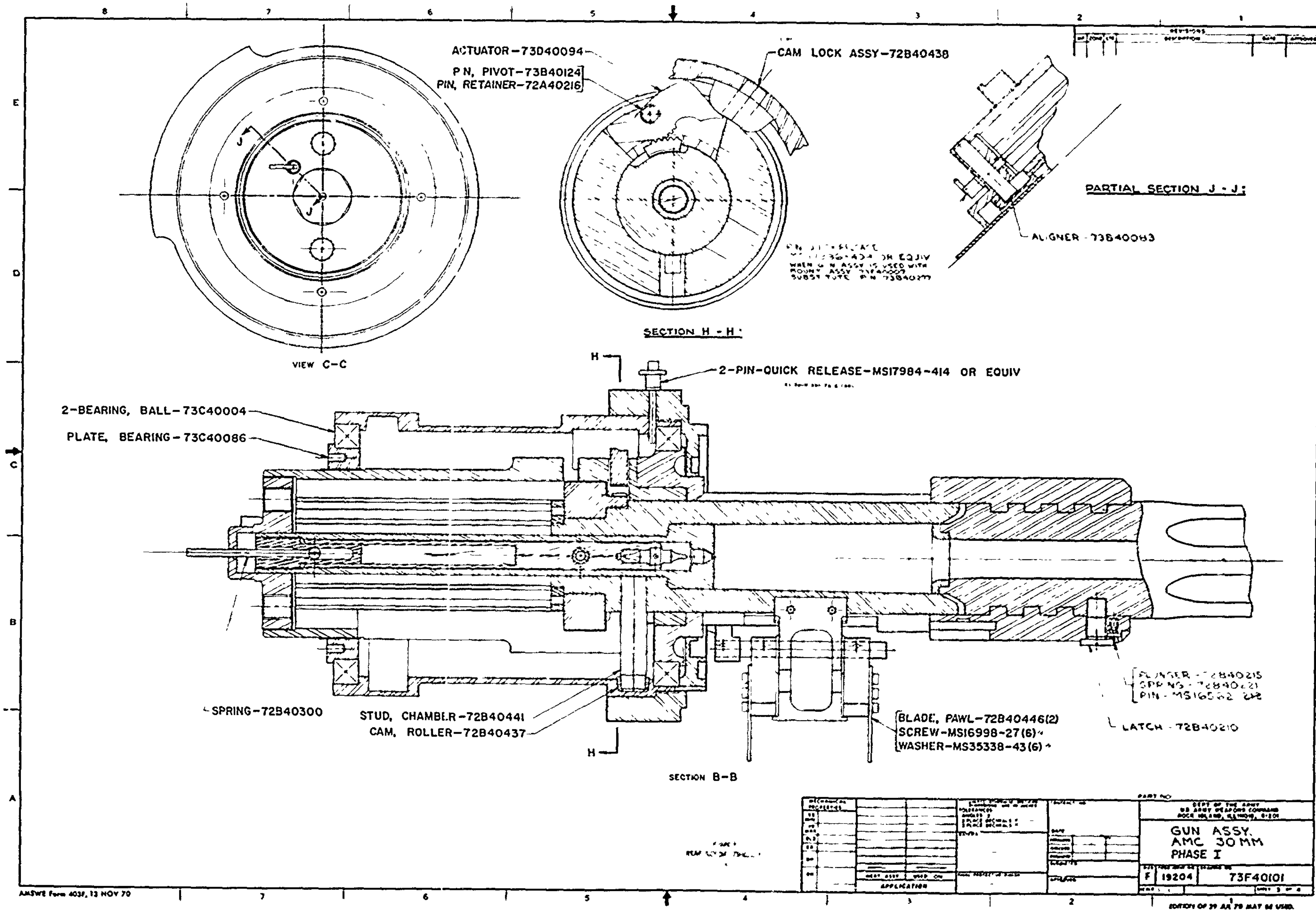
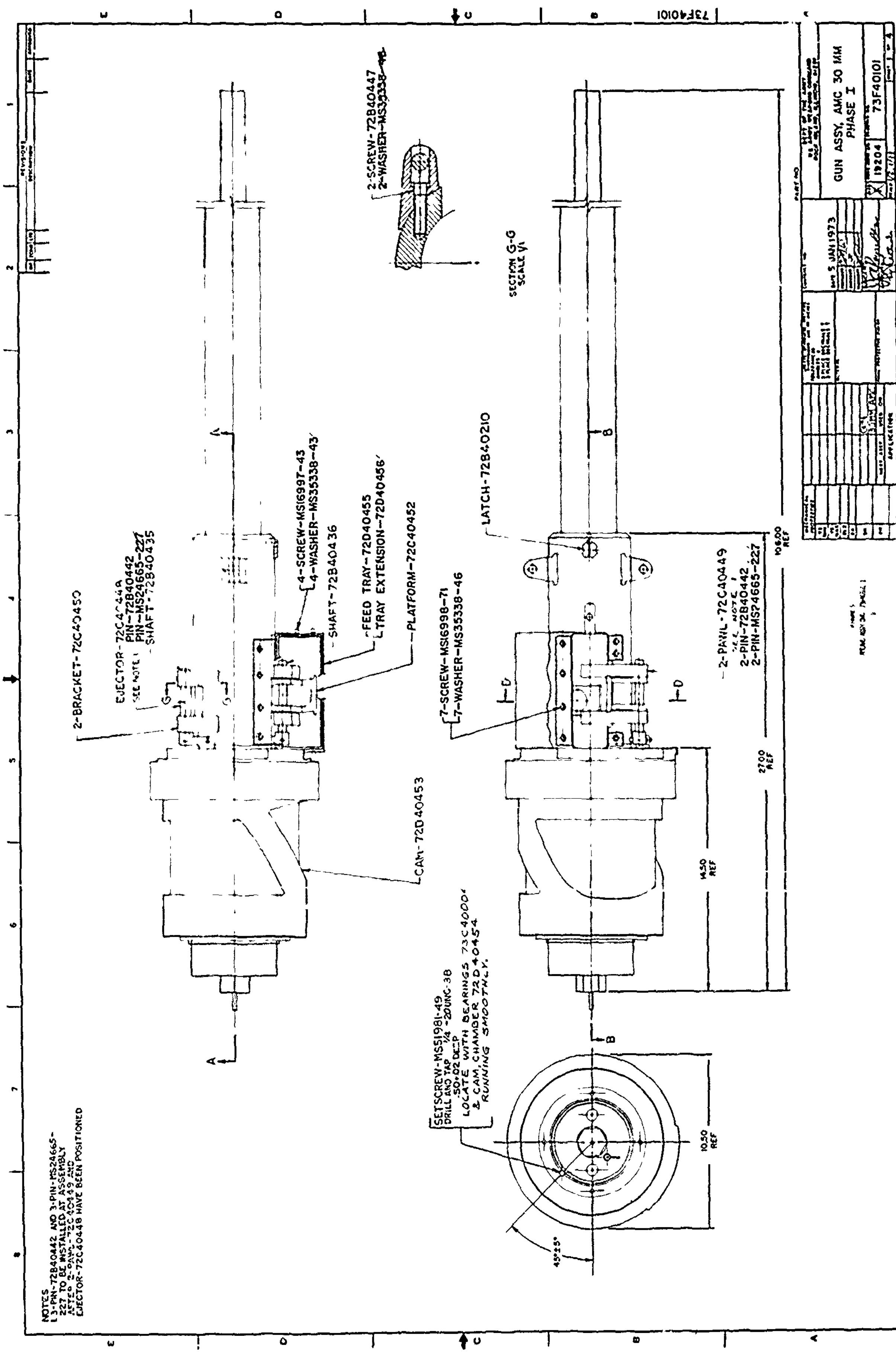

ACTUATOR-73D40094
P N, PIVOT-73B40124
PIN, RETAINER-72A40216
CAM LOCK ASSY-72B40438
PARTIAL SECTION J - J
ALIGNER - 73B40083
VIEW C-C
SECTION H - H
2-PIN-QUICK RELEASE-MS17984-414 OR EQUIV
2-BEARING, BALL-73C40004
PLATE, BEARING-73C40086
SPRING-72B40300
STUD, CHAMBER-72B40441
CAM, ROLLER-72B40437
SECTION B-B
BLADE, PAWL-72B40446(2)
SCREW-MS16998-27(6)
WASHER-MS35338-43(6)
LATCH - 72B40210
GUN ASSY.
AMC 30MM
PHASE I
19204
73F40101
APPLICATION
AMSWE Form 403F, 13 NOV 70

Fig. 4


73F40101
2-SCREW-72B40447
2-WASHER-MS35338-46
SECTION G-G
SCALE 1/1
2-BRACKET-72C40450
EJECTOR-72C40444
PIN-72B40442
PIN-MS24665-227
SHAFT-72B40435
4-SCREW-MS16997-43
4-WASHER-MS35338-43
SHAFT-72B40436
FEED TRAY-72D40455
TRAY EXTENSION-72D40456
PLATFORM-72C40452
CAM-72D40453
LATCH-72B40210
7-SCREW-MS16998-71
7-WASHER-MS35338-46
2-PAWL-72C40449
2-PIN-72B40442
2-PIN-MS24665-227
SETSCREW-MS51981-49
DRILL AND TAP 1/4-20UNC-3B
.50+.02 DEEP
LOCATE WITH BEARINGS 73C40004
& CAM, CHAMBER 72D40454
RUNNING SMOOTHLY.
106.00 REF
27.00 REF
14.50 REF
10.50 REF
45°±5°
NOTES
1. 3-PIN-72B40442 AND 3-PIN-MS24665-227 TO BE INSTALLED AT ASSEMBLY AFTER 2-PAWL-72C40449 AND EJECTOR-72C40448 HAVE BEEN POSITIONED
GUN ASSY, AMC 30 MM
PHASE I
19204
73F40101
5 JAN 1973

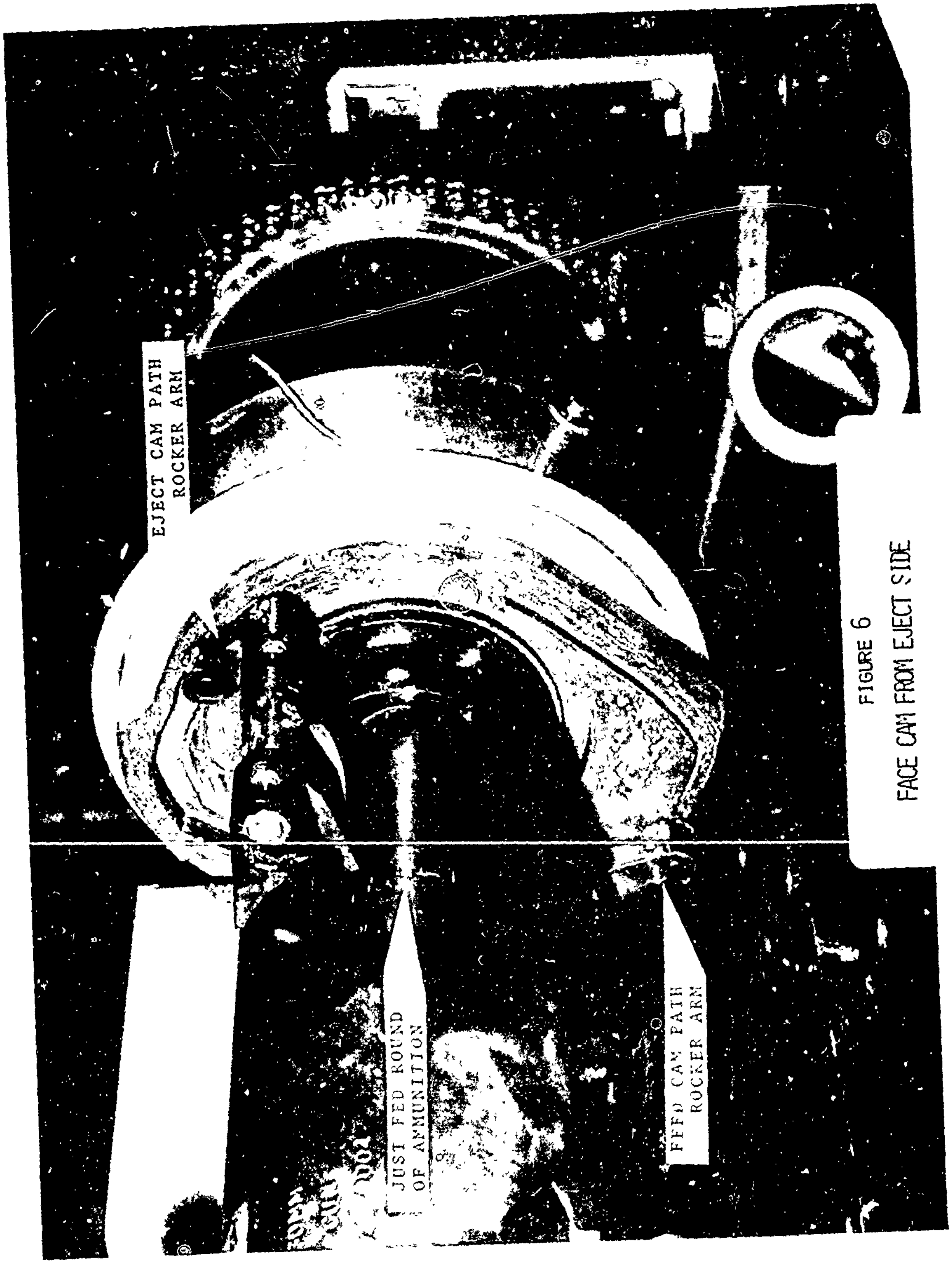


FIGURE 6
FACE CAM FROM EJECT SIDE

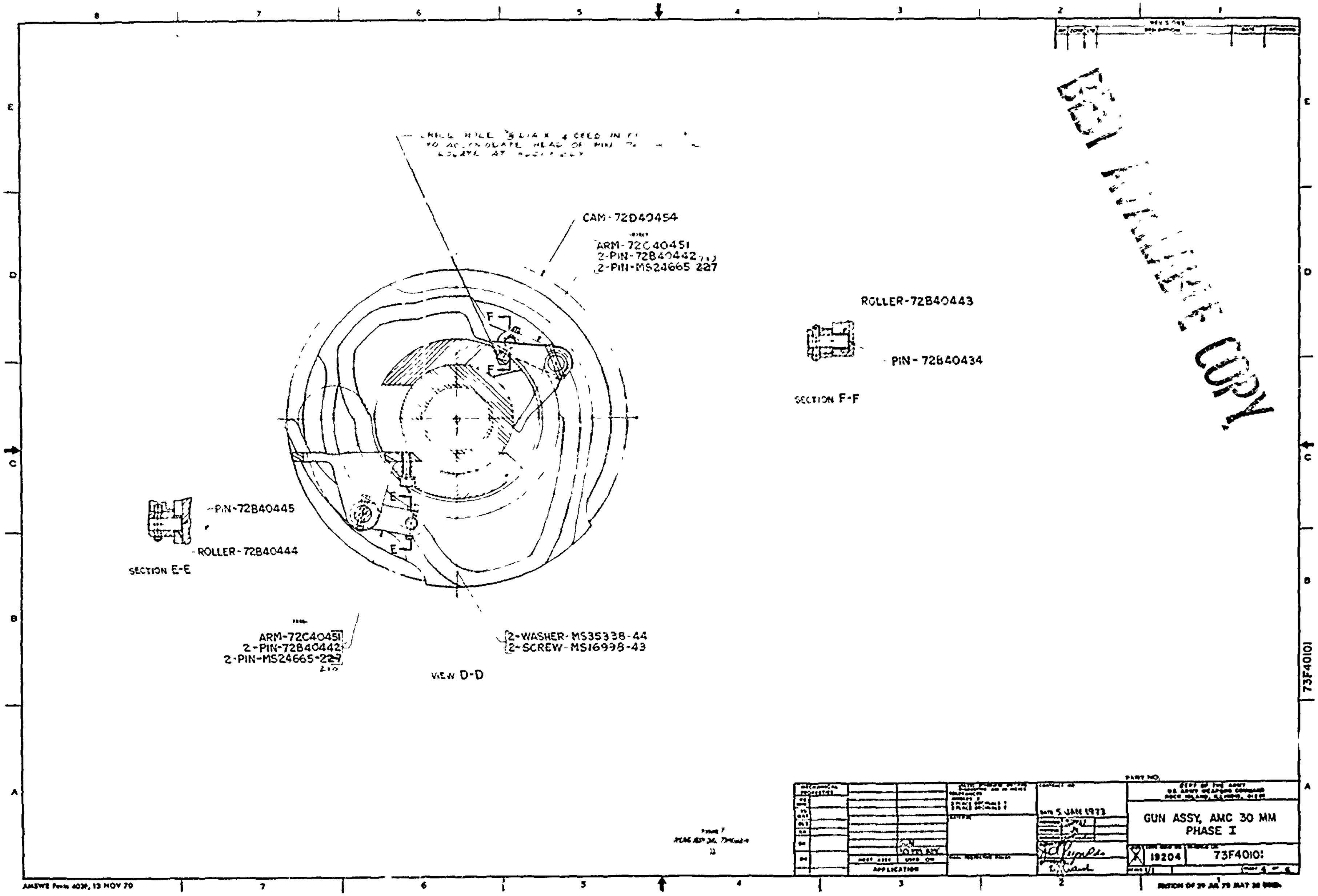

CAM-72D40454
ARM-72C40451
2-PIN-72B40442
2-PIN-MS24665 227
ROLLER-72B40443
PIN-72B40434
SECTION F-F
PIN-72B40445
ROLLER-72B40444
SECTION E-E
ARM-72C40451
2-PIN-72B40442
2-PIN-MS24665-227
2-WASHER-MS35338-44
2-SCREW-MS16998-43
VIEW D-D
GUN ASSY, AMC 30 MM
PHASE I
19204
73F40101
5 JAN 1973
73F40101
AMSWE Form 4039, 13 NOV 70

Fig. 7

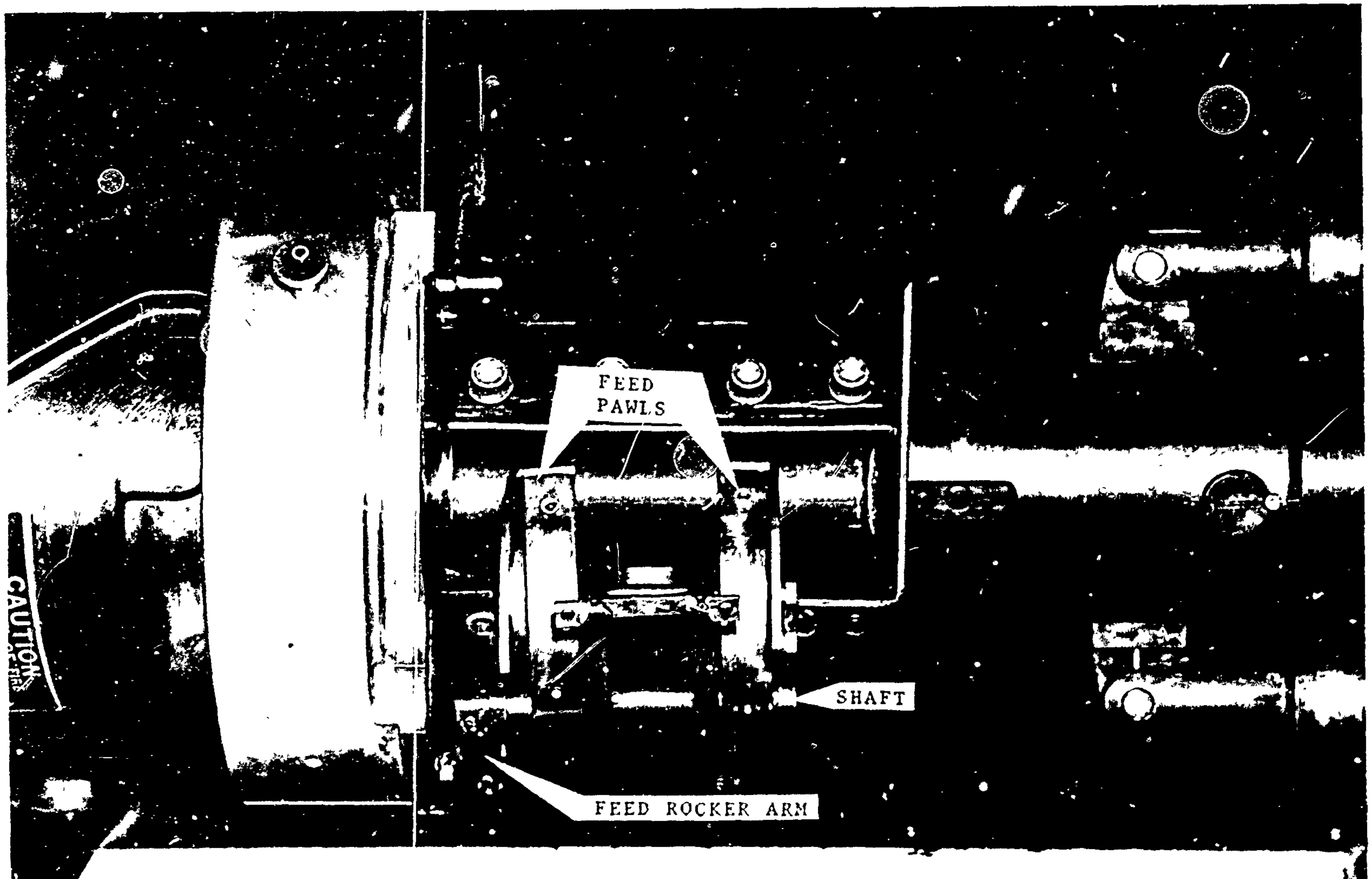


FIGURE 3
FEED MECHANISM

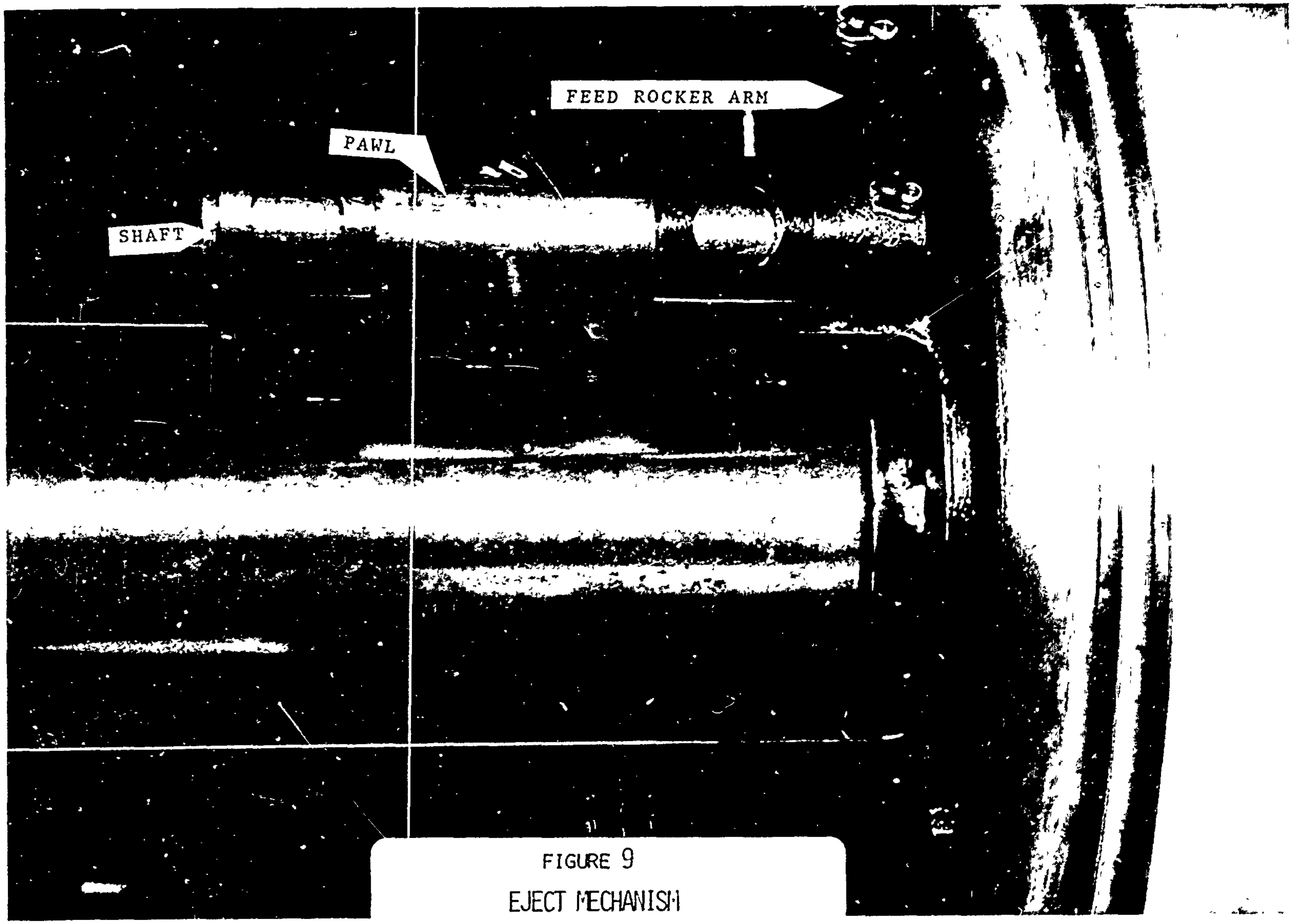


FIGURE 9
EJECT MECHANISM

2.4 LOCKING/SEAR ASSEMBLIES

As the cam drum rotates, the lump cam, located on the cam drum inside diameter (Figure 4) contacts the actuator. The unidirectional rotary motion of the cam drum is changed to a bidirectional oscillating motion of the actuator by their cam/follower relationship. The motion (Figure A1-13) is controlled by the profile of the contact surface of the actuator. As the lump cam moves along the actuator, it forces (cams) the actuator to rotate about its own pivot point. As the actuator rotates, it in turn rotates the lock ring to its locked position via a set of gear teeth on the actuator and mating teeth on the lock ring. Once in the locked position, the lump cam rides on the dwell portion of the actuator profile. Since the lump cam and actuator are in constant contact during this period no motion can occur, thus positive locking during the total lock dwell results. As the lump cam moves along the actuator, it contacts the unlock portion of the actuator profile. The actuator is forced (cammed) back to its original position and through the gear teeth it returns the lock ring to its original unlocked position.

Located on the inside diameter of the lock ring is a small cam path (Figure 10). Riding on this small cam path is a bean-like object called a sear extension (Figure 11). As the lock ring is rotated, the small cam path lifts the sear extension. When the lock ring reaches its fully locked position, the sear extension is at the top of the small cam path and just enough lift has occurred so that the sear extension releases the weapon's main sear in the bolt assembly. If the lock ring does not turn to full lock, the sear extension cannot reach full lift. This feature prevents the gun from being fired in any position other than the desired full lock position.

Since the lock ring is concentric about the longitudinal axis of the gun, the load on the receiver is also concentric about this axis, and therefore induces no bending moments to the receiver.

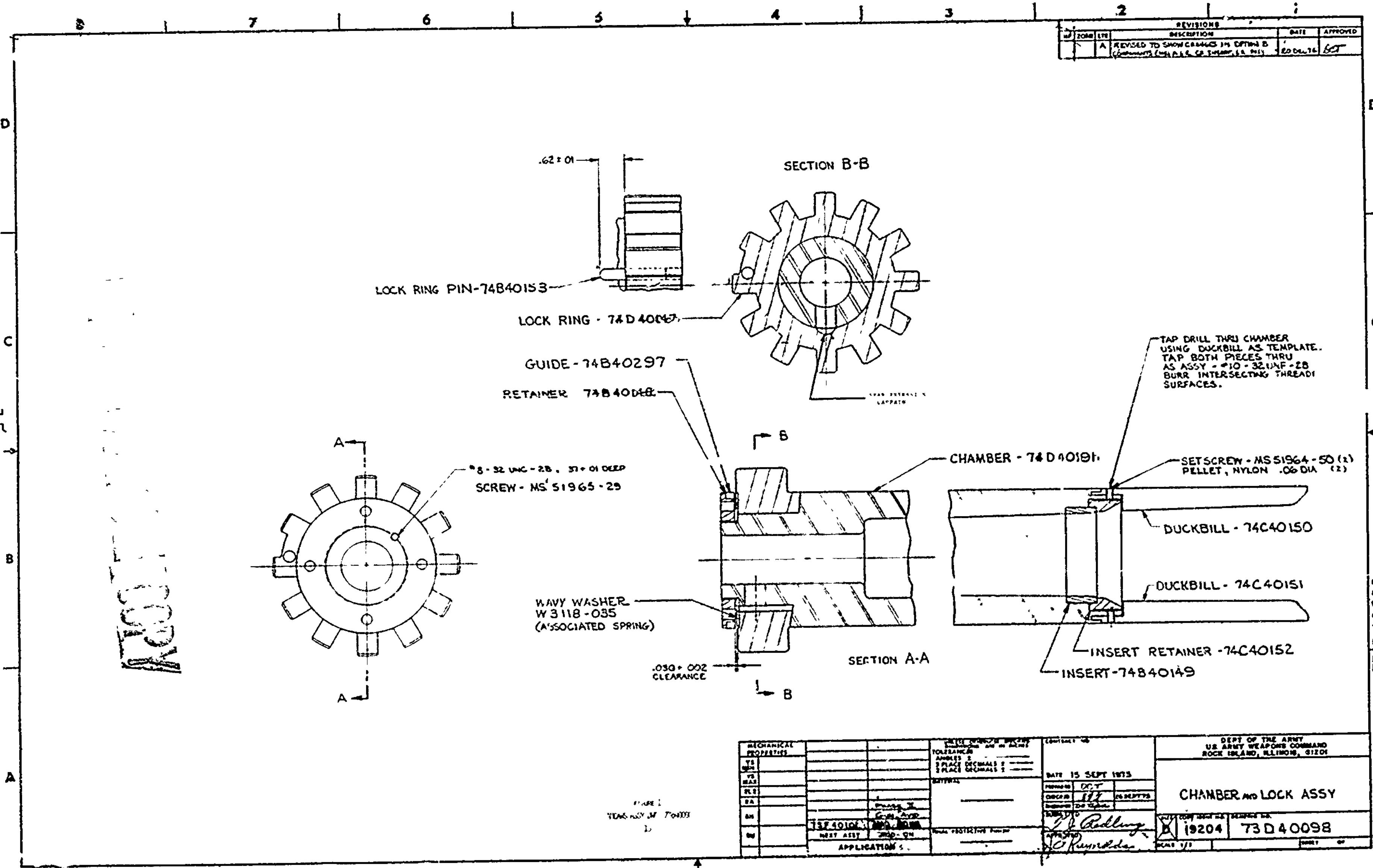

SECTION B-B
.62 ± .01
LOCK RING PIN-74B40153
LOCK RING - 74D40047
GUIDE-74B40297
RETAINER 74B40048
TAP DRILL THRU CHAMBER USING DUCKBILL AS TEMPLATE. TAP BOTH PIECES THRU AS ASSY - #10-32 UNF-2B BURR INTERSECTING THREAD SURFACES.
#8-32 UNC-2B, .37 ± .01 DEEP
SCREW - MS 51965-29
CHAMBER - 74D40191
SETSCREW - MS 51964-50 (2)
PELLET, NYLON .06 DIA (2)
DUCKBILL - 74C40150
DUCKBILL - 74C40151
WAVY WASHER W3118-035 (ASSOCIATED SPRING)
.030 ± .002 CLEARANCE
SECTION A-A
INSERT RETAINER -74C40152
INSERT-74B40149
REVISIONS
A REVISED TO SHOW CHANGES
DATE 15 SEPT 1973
DEPT OF THE ARMY U.S. ARMY WEAPONS COMMAND ROCK ISLAND, ILLINOIS, 61201
CHAMBER AND LOCK ASSY
19204
73D40098
MECHANICAL PROPERTIES
APPLICATION
NEXT ASSY
AMSWE Form 4030, 29 Jul 69

Fig. 10

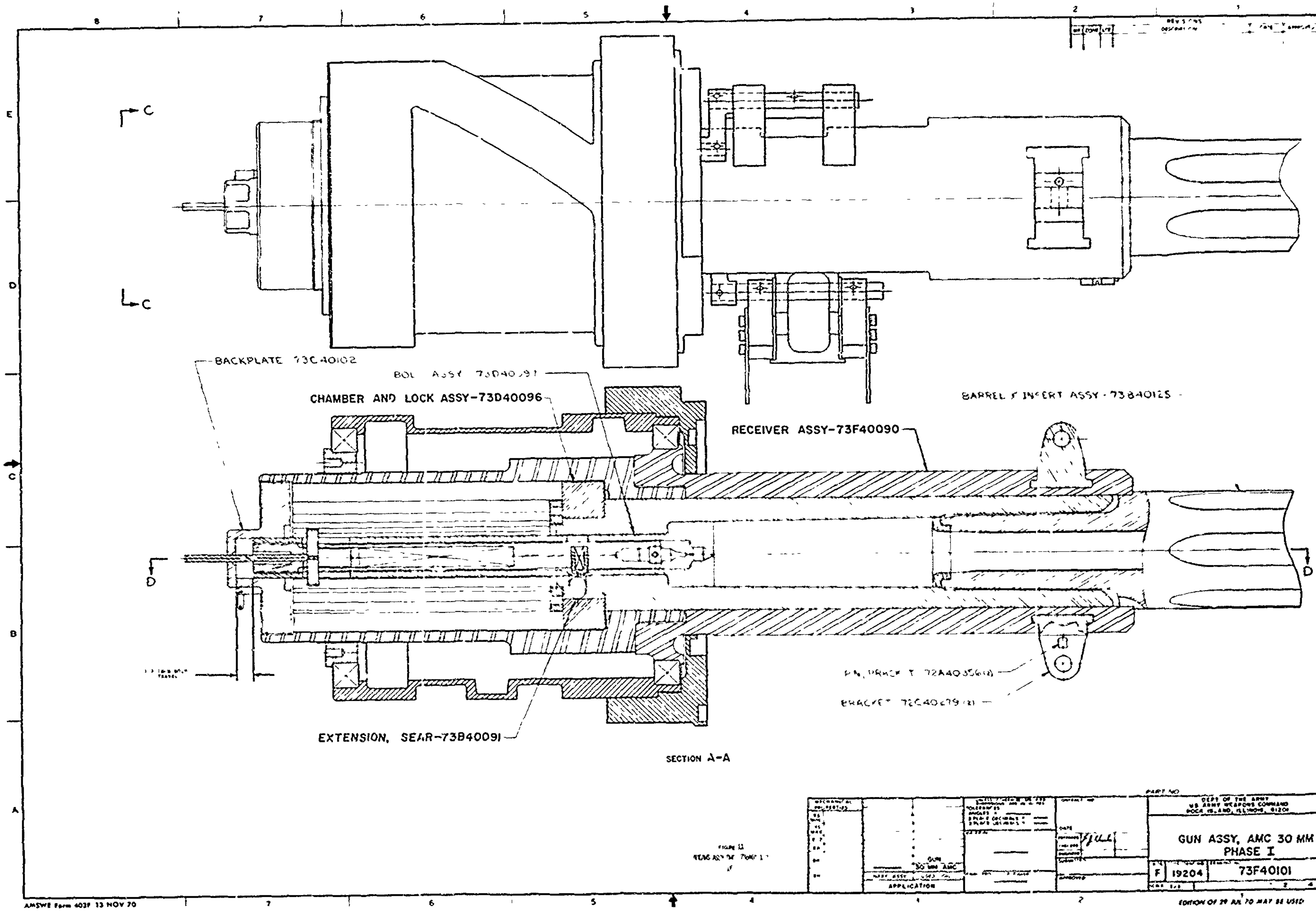

BACKPLATE 73C40102
BOL ASSY 73D40091
CHAMBER AND LOCK ASSY-73D40096
RECEIVER ASSY-73F40090
BARREL & INSERT ASSY-73B40125
EXTENSION, SEAR-73B40091
SECTION A-A
GUN ASSY, AMC 30 MM
PHASE I
19204
73F40101
AMSWE Form 403F 13 NOV 70

Fig. 11

2.5 BOLT

The bolt assembly is pictured in Figure 12. The firing pin is contained in the bolt housing and is driven by a spring with a rate (at assembled height) of about seven pounds per inch. The firing pin is held in a ready-to-fire position by the sear extending through the sear hole in the bolt housing. The firing pin is seared off as discussed in the Locking/Sear explanation. The firing pin can only be released from its ready-to-fire position when the weapon is fully locked. The bolt has about one-half inch of total travel relative to the receiver along the centerline of the weapon. In operation, the firing pin is seared off by the action of fully locking the lock ring. The firing pin drives forward and discharges the round. A fixed time elapses that allows the projectile to exit the barrel and the high pressure exhaust gases to bleed off. The lock ring unlocks and the chamber begins to move back. The fired round's case, the chamber assembly, and the bolt assembly move rearward as a unit for about the half inch indicated in Figure 11 and 13. At this point the bolt housing impacts the backplate and the force of this impact frees the round case from the chamber wall. The chamber assembly continues to move over the now stationary bolt assembly, thus exposing the fired round (Figure 14). The chamber-lock ring assembly moves rearward so far as to interfere with the bolt crosspin (Figure 12) and move it from the fully forward position (because the firing pin has been seared off) to its fully rearward position (as shown). In the fully rearward position, the firing spring is recompressed and the sear drops into the sear hole. The chamber is also fully rearward and load/eject operations take place. The bolt is ready for another cycle. The shoulder of the chamber interferes with the shoulder on the bolt head to bring the bolt away from the baseplate (Figure 13) and secure the round for the next firing. Figure 15 shows a round and all the components in a now ready to fire position.

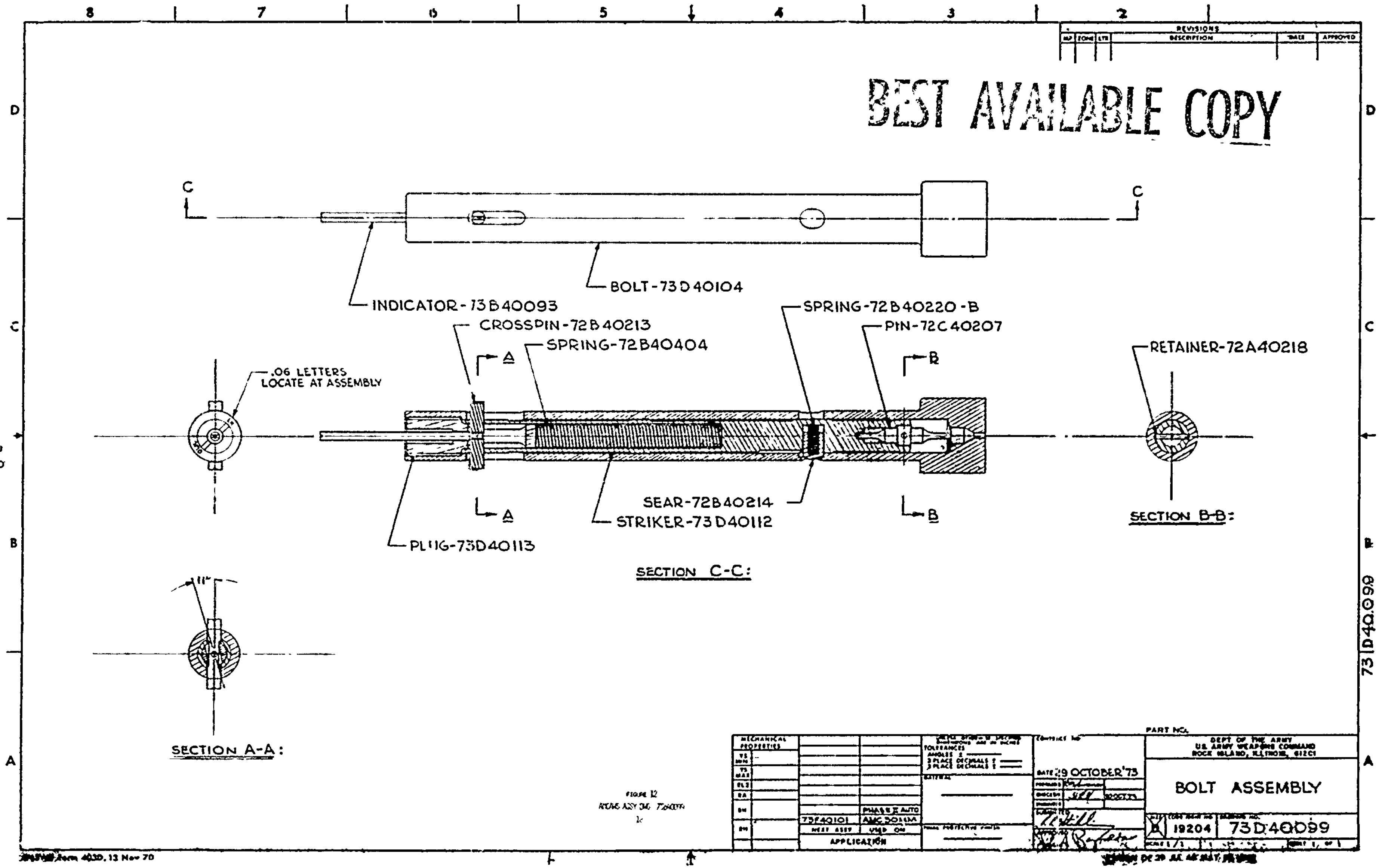

BEST AVAILABLE COPY
INDICATOR-73B40093
BOLT-73D40104
CROSSPIN-72B40213
SPRING-72B40404
SPRING-72B40220-B
PIN-72C40207
RETAINER-72A40218
.06 LETTERS
LOCATE AT ASSEMBLY
SEAR-72B40214
STRIKER-73D40112
PLUG-73D40113
SECTION C-C:
SECTION B-B:
SECTION A-A:
29 OCTOBER '73
BOLT ASSEMBLY
19204
73D40099
APPLICATION

Fig. 12

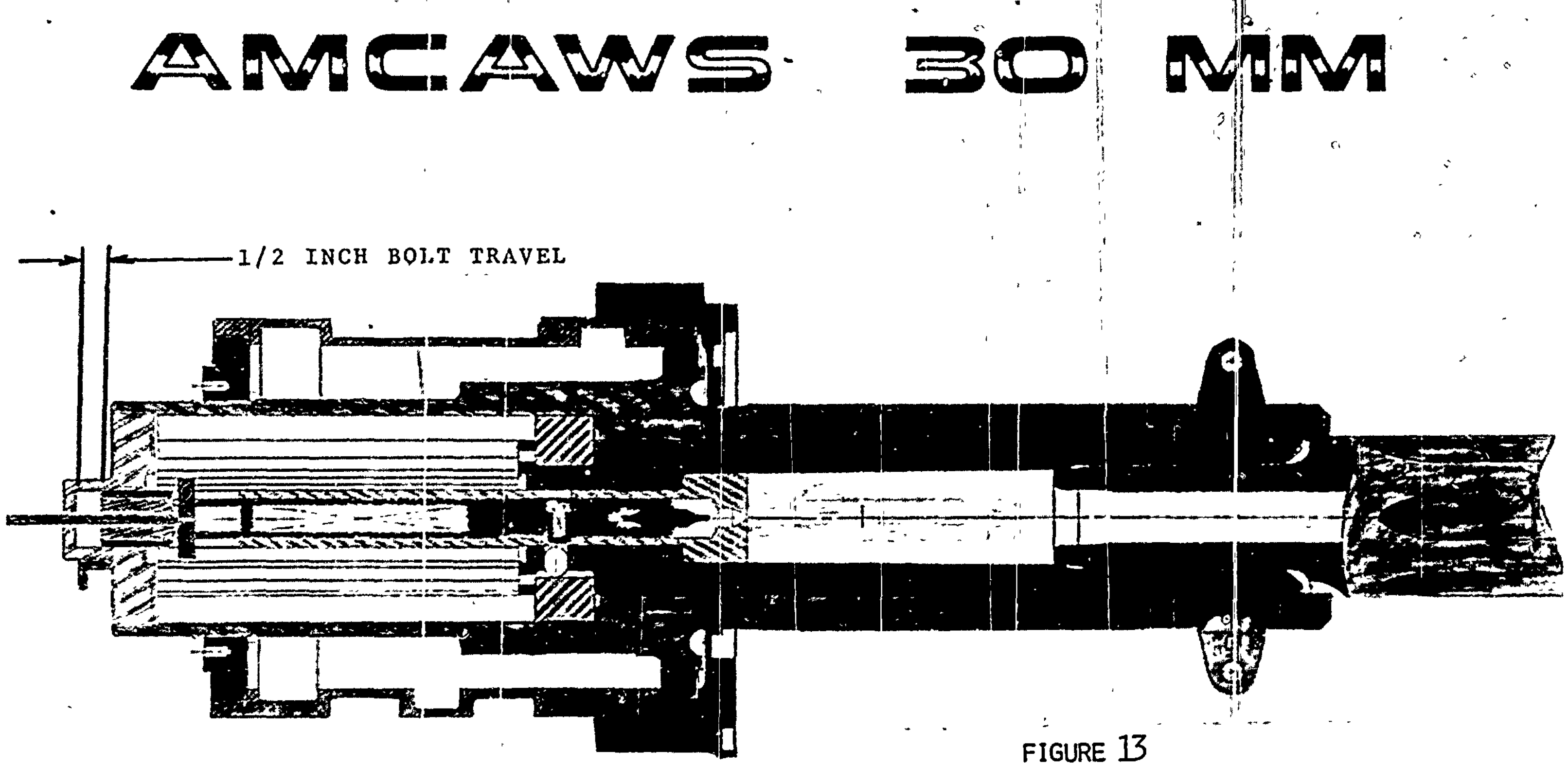


FIGURE 13
AMCAWS LAYOUT

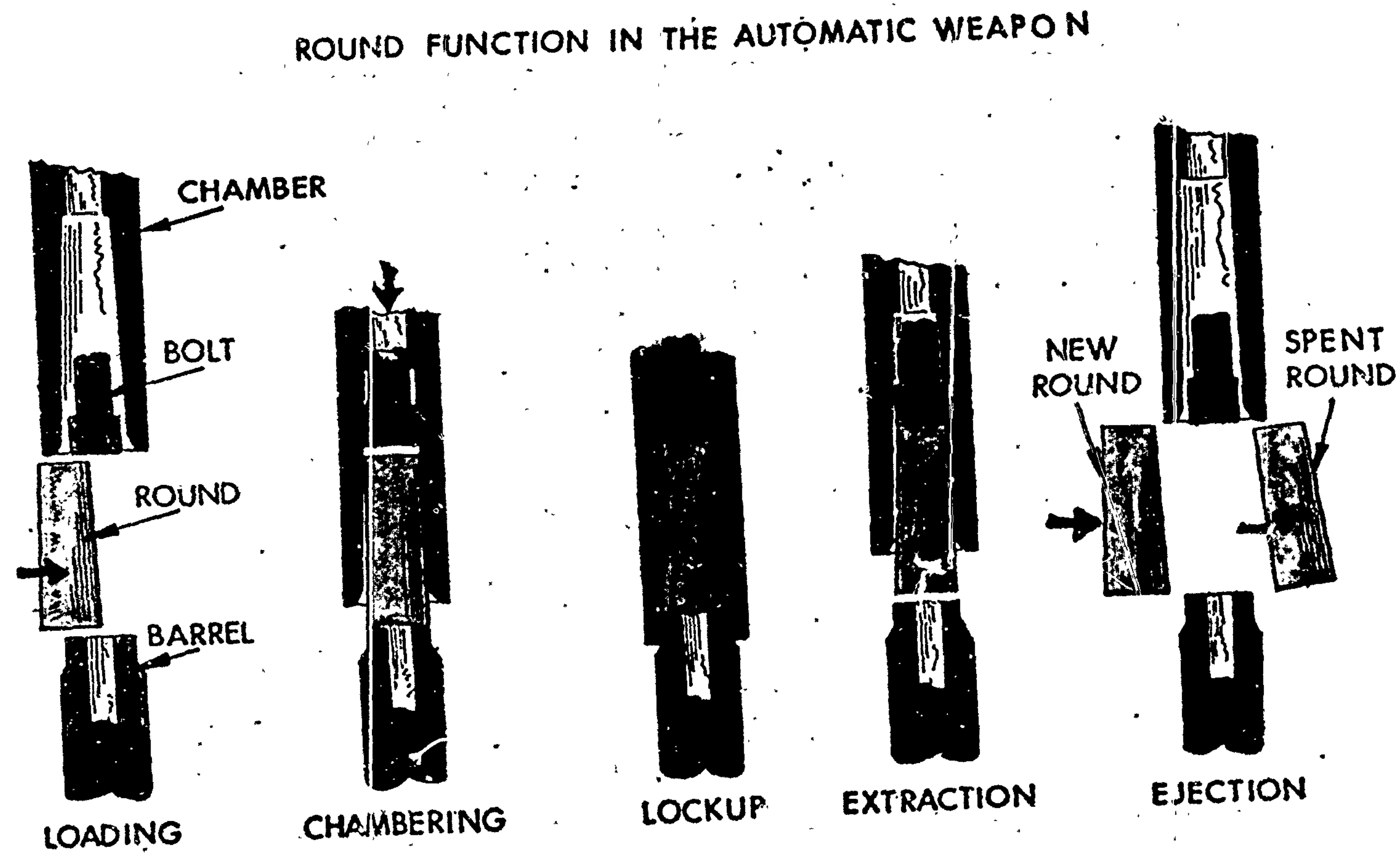
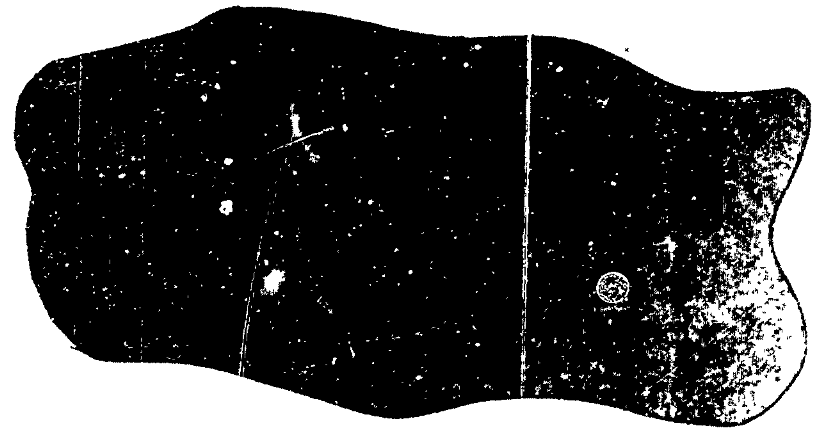


FIGURE 14

ROUND FUNCTION SEQUENCE



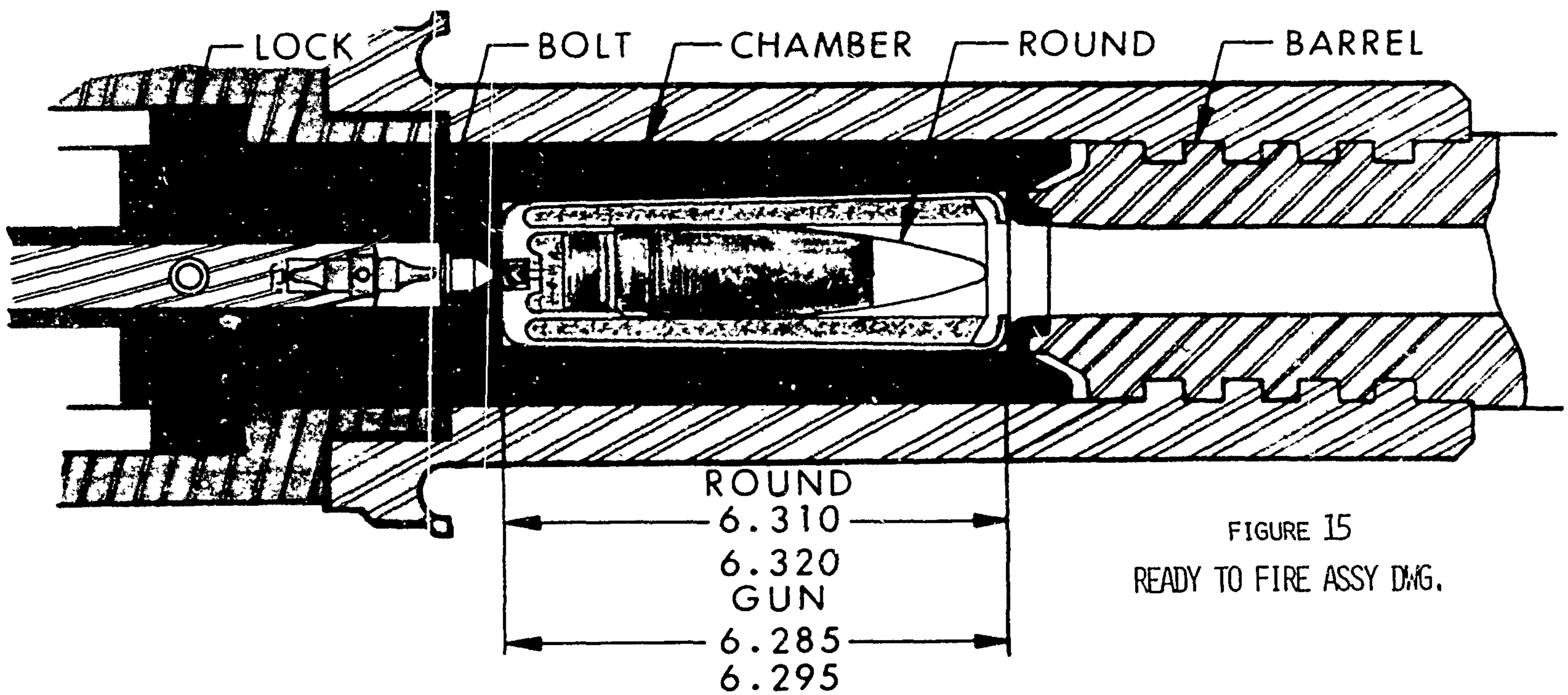


FIGURE 15
READY TO FIRE ASSY DWG.

2.6 CHAMBER

The chamber is pictured in Figure 10 with the lock ring in place. The chamber is translated by the action of the drum cam path through the follower stud. The stud is cantilevered from the chamber to the drum cam path. The stud is constrained to move within the receiver slot shown in Figure 16. In the automatic weapon assembly the handle seen in Figure 16 is replaced by the follower stud.

The follower stud is cantilevered as shown in Figure 4. The duckbills of Figure 10 are essentially extensions of the top and bottom of the chamber and serve to keep the round very close to the weapon's centerline during feed and eject. The matching tapers on the chamber and round cause line to line fit of the round and the chamber during the lockup phase of Figure 14. The chamber provides the support during the peak pressures, although the round itself performs all obturation functions (1). Figures 13 and 14 show the chamber's relationships to the other components just prior to firing.

2.7 RECEIVER

The receiver (Figure 17) essentially acts as a housing for the chamber and bolt assemblies. The front end of the receiver has a set of lugs which hold the barrel. The rear of the receiver has flanges cut that match the lock ring lugs (Figure 10) and against which the lock ring lugs seat during firing. In the unlocked position the lock ring lugs (thus the chamber) can move down the receiver, but the locking operation turns the lock ring 15°, which lines up the lock ring lugs and the corresponding receiver flanges. The receiver is slotted (Figure 16 and 17) to provide a guide for the chamber-to-drum cam path follower stud. The forward area of the receiver, near the barrel lugs, is cut to allow brackets that connect to the buffer packs. The receiver window, just to the rear of the barrel lugs, is cut to allow the ammunition to be fed and ejected.

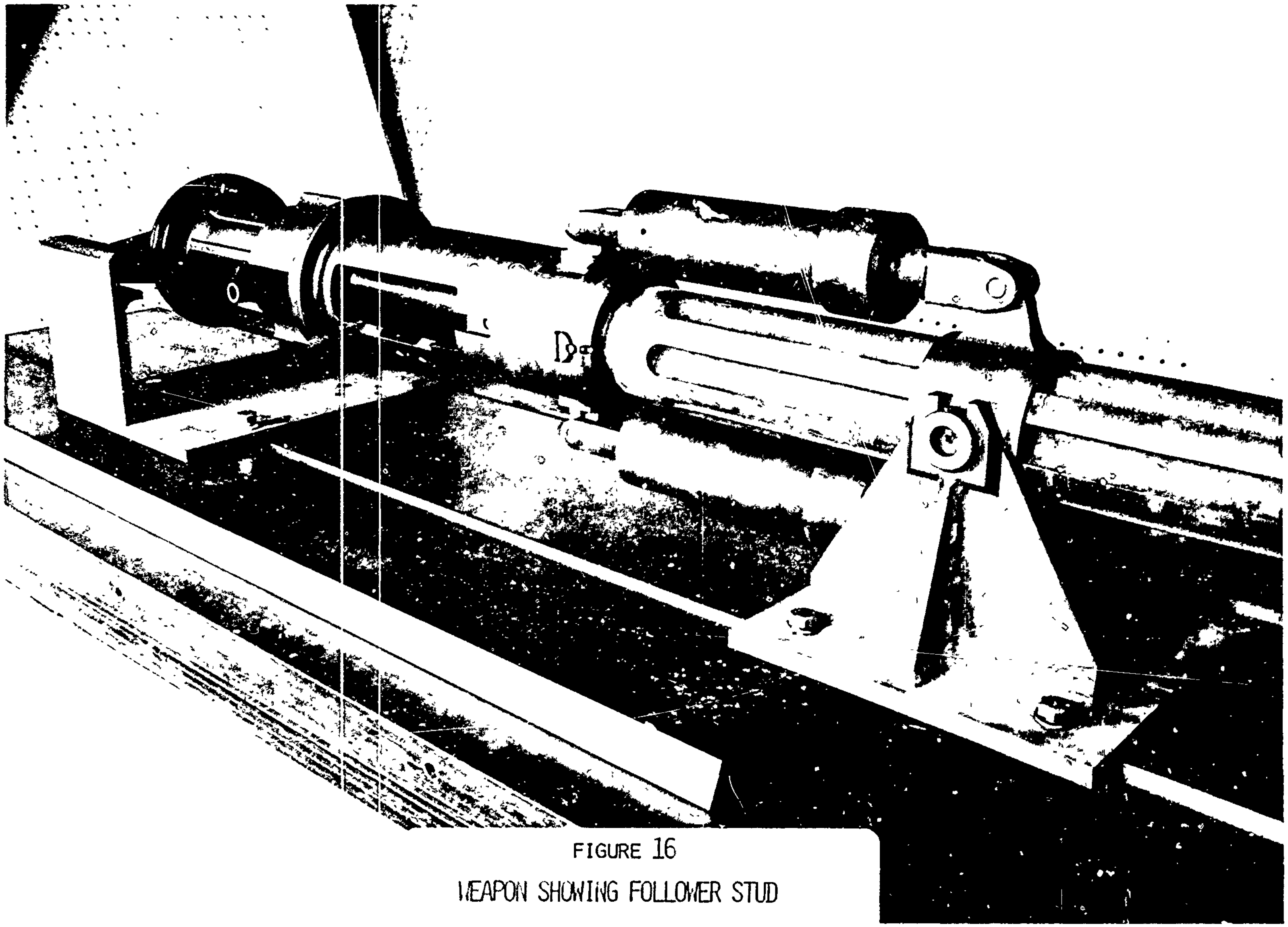
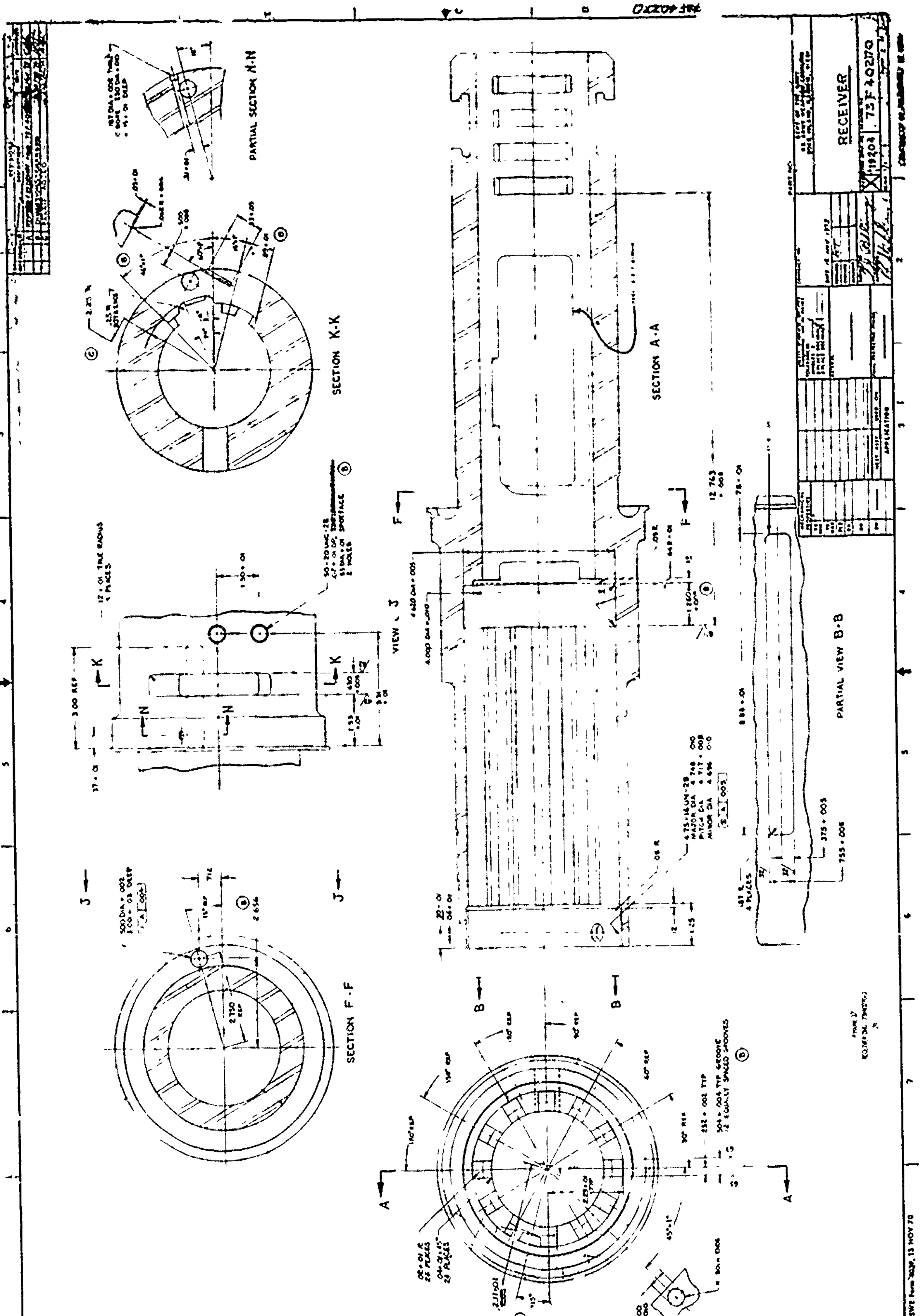


FIGURE 16

WEAPON SHOWING FOLLOWER STUD



RECEIVER
SECTION A-A
SECTION K-K
PARTIAL SECTION N-N
SECTION F-F
PARTIAL VIEW B-B
VIEW J

Fig. 17

2.8 BUFFERS

The buffer assembly drawing (Figure B) shows one of the two identical buffers. The buffers are mounted on the top and bottom of the weapon, as can be seen in Figure 19. The ring seen in Figure 16 onto which the front end of the buffer packs fasten clears the barrel and allows the barrel to move up and back within it. The ring has trunion mounts on both sides which allows the gun to fasten onto its firing platform. As can be seen from the assembly drawing (Figure 18) the buffers have the same preload and spring-rate in both recoil and counter-recoil. These two spring factors are variable simply by changing the belville spring packs. Currently the weapon has nineteen sets of double springs in each buffer pack which gives a preload and rate per buffer of 3500 pounds and 10,700 pounds per inch respectively, although Figure 18 indicates triple spring sets in an elongated buffer.

Upon firing, a peak force of up to 200,000 pounds is developed against the flange lugs on the receiver. This force (more properly, the firing impulse) is transmitted through the receiver to the buffers which lengthen to absorb the recoil energy of the firing. The buffer begins to counter recoil and eventually damps out prior to the next shot. Typically recoil is (for the preload and rate discussed) 1.20 inch and the counterrecoil is .75 inch. The system is completely damped in .120 seconds or 1/5 of a cycle at 121 spm.

2.9 BARREL

The barrel joins the receiver with a set of lugs as shown in Figure 15. Mechanically the barrel acts only as a recoiling mass although rifling torques (2) might be significant in a multiple degree of freedom model. Note the cut away portion of the barrel at the muzzle end in Figure 13. This gives clearance to the chamber duckbills (Figure 10) during the chamber forward portion of the firing cycle.

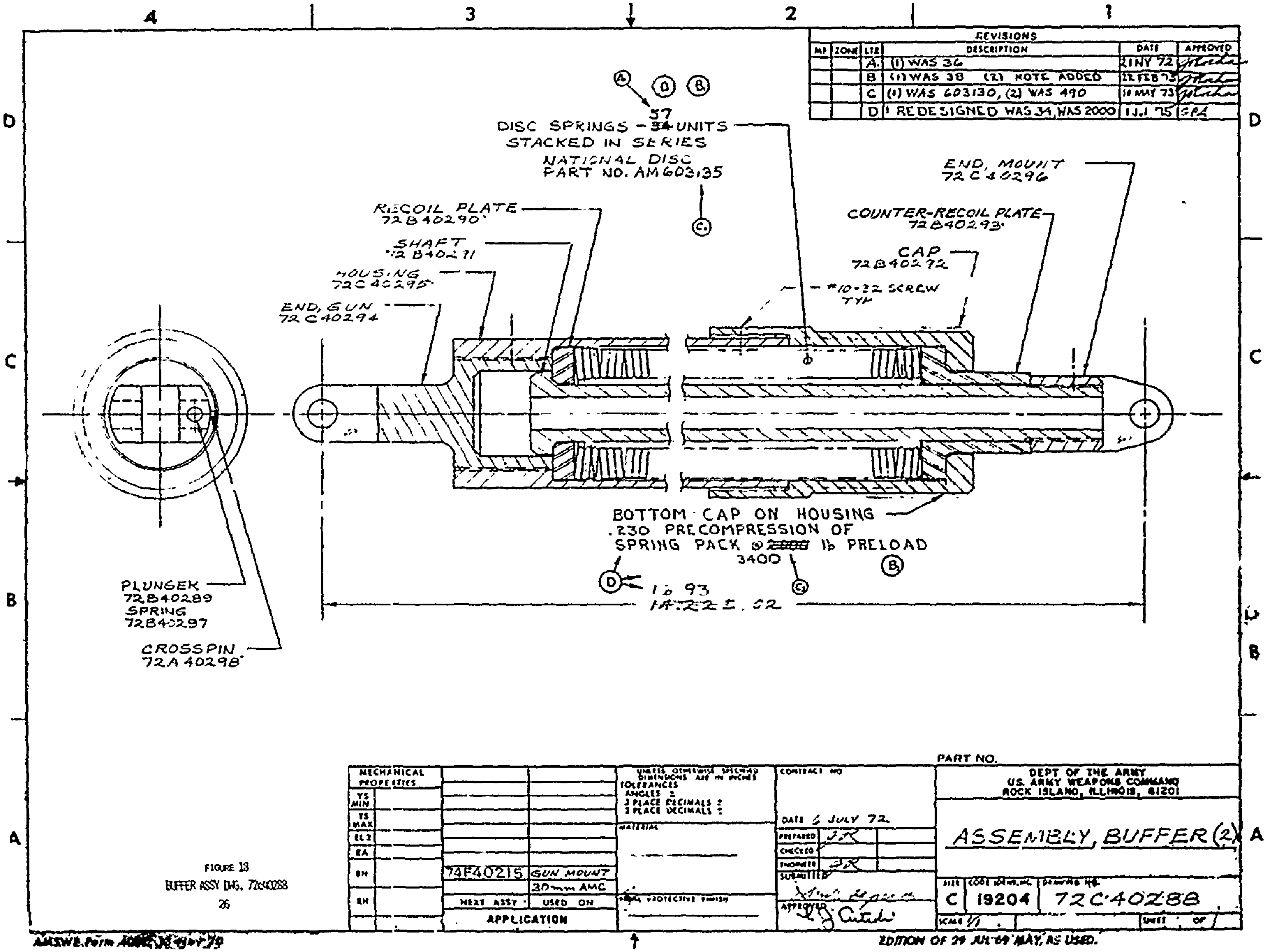

REVISIONS
A (1) WAS 36 | 21 NY 72
B (1) WAS 38 (2) NOTE ADDED | 22 FEB 73
C (1) WAS 603130, (2) WAS 490 | 18 MAY 73
D REDESIGNED WAS 34, WAS 2000
DISC SPRINGS - 57 UNITS
STACKED IN SERIES
NATIONAL DISC
PART NO. AM 603135
END, MOUNT
72C40296
COUNTER-RECOIL PLATE
72B40293
CAP
72B40292
#10-32 SCREW
TYP
RECOIL PLATE
72B40290
SHAFT
72B40271
HOUSING
72C40295
END, GUN
72C40294
BOTTOM CAP ON HOUSING
.230 PRECOMPRESSION OF
SPRING PACK @ 3400 lb PRELOAD
16.93
PLUNGER
72B40289
SPRING
72B40297
CROSSPIN
72A40298
MECHANICAL PROPERTIES
74F40215
GUN MOUNT
30mm AMC
NEXT ASSY
USED ON
APPLICATION
UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES
TOLERANCES
ANGLES ±
3 PLACE DECIMALS ±
2 PLACE DECIMALS ±
MATERIAL
FINAL PROTECTIVE FINISH
CONTRACT NO
DATE 6 JULY 72
PREPARED
CHECKED
ENGINEER
SUBMITTED
APPROVED
PART NO.
DEPT OF THE ARMY
U.S. ARMY WEAPONS COMMAND
ROCK ISLAND, ILLINOIS, 61201
ASSEMBLY, BUFFER (2)
C 19204 72C40288
SCALE 1/1
SHEET OF
FIGURE 18
BUFFER ASSY DWG. 72c40288
26
EDITION OF 29 JUL 69 MAY BE USED.

Fig. 18

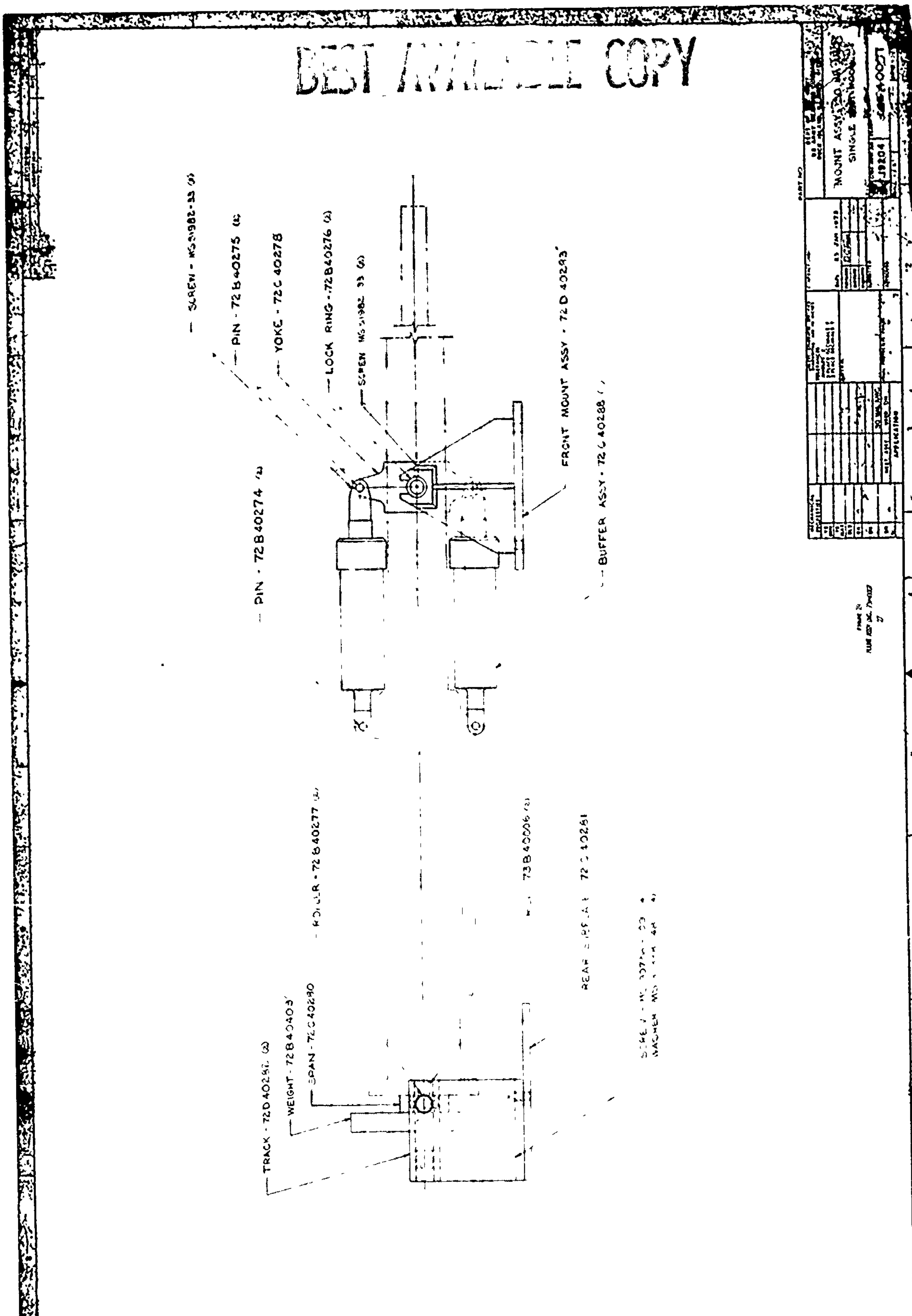

PIN - 72B40275 (2)
YOKE - 72C 40278
LOCK RING - 72B40276 (2)
FRONT MOUNT ASSY - 72D 40293
BUFFER ASSY - 72C40288
PIN - 72B40274 (2)
TRACK - 72D40282 (2)
WEIGHT - 72B40403
SPAN - 72C40290

Fig. 19

2.10 DRIVE TRAIN

The gear drive assembly (Figures 2 and 20) allows four final shots per minute rates, 90, 121, 181, and 242. The prototype weapon is limited to 121 spm because there is insufficient lock time (time necessary to sear off the firing pin, obtain projectile exit, and bleed of the high-pressure gasses) designed into the drum cam path to allow the higher rates. At 121 spm the 59 tooth gear from the motor drives a 120 tooth pickup gear. The 16 tooth pinion gear which is part of the shaft for the 120 tooth gear then drives the 151 tooth gear on the outside diameter of the drum cam (Figure 2). The torque-speed curve for the motor is shown in Figure 21. This curve is from data supplied by Aeronutronic-Ford. The drive motor itself is from an XM-140 system. Since a firing occurs once every 360° rotation of the drum cam, a quick calculation indicates in the 121 spm configuration a firing cycle completes every 6910° rotation of the 59 tooth motor gear. The 59 tooth gear angle is the input angle for the mathematical model developed. Zero degrees input angle is defined so that the drum cam is also at zero degrees (as a reference, the weapon actually fires when the drum cam is at about 33° rotation, or 541° rotation of the 59 tooth gear).

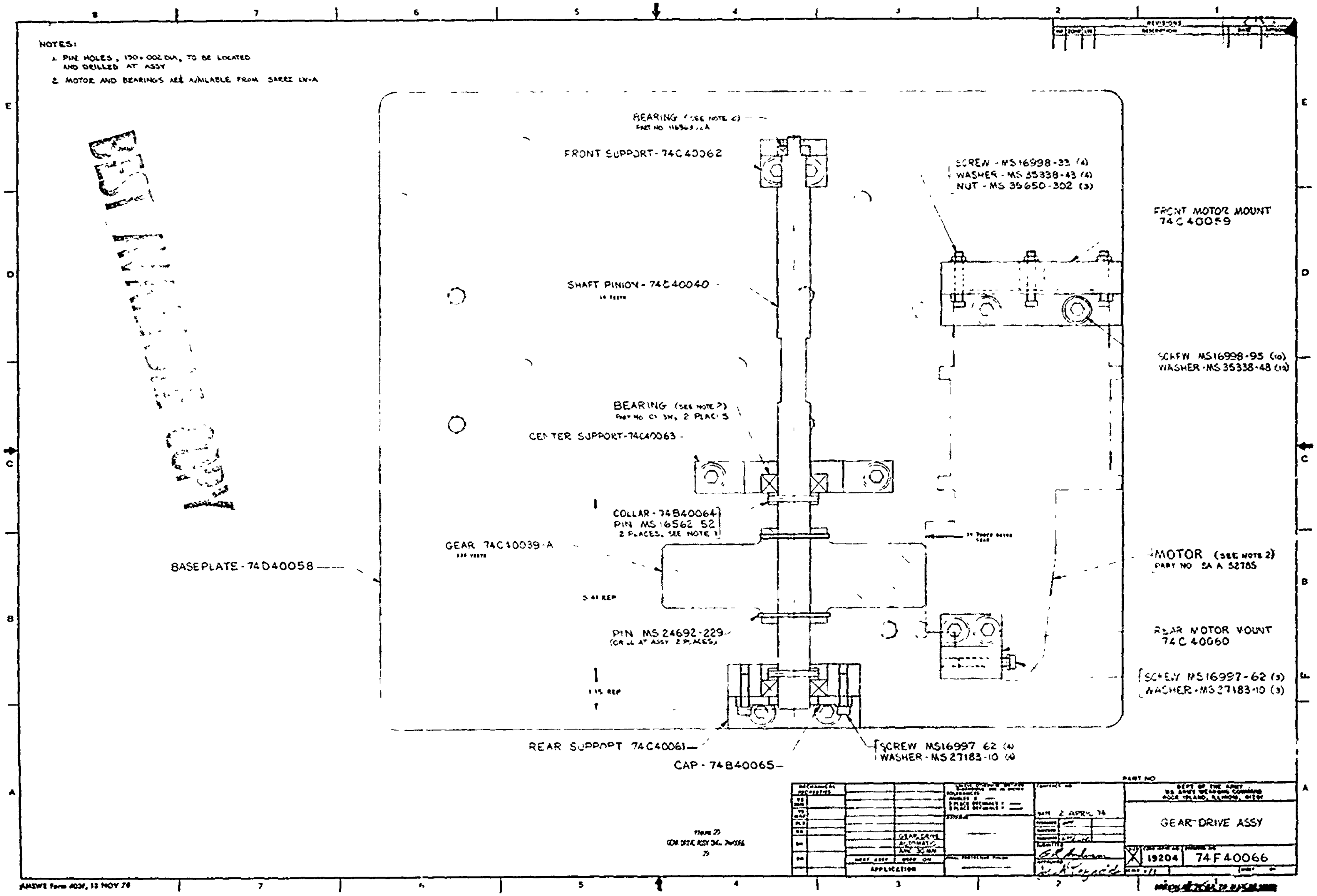

NOTES:
1. PIN HOLES, .130+.002 DIA, TO BE LOCATED AND DRILLED AT ASSY
2. MOTOR AND BEARINGS ARE AVAILABLE FROM SAREZ LV-A
BEARING (SEE NOTE 2)
FRONT SUPPORT - 74C40062
SHAFT PINION - 74C40040
BEARING (SEE NOTE 2)
CENTER SUPPORT - 74C40063
COLLAR - 74B40064
PIN MS16562 52
2 PLACES, SEE NOTE 1
GEAR 74C40039-A
BASEPLATE - 74D40058
PIN MS 24692-229
REAR SUPPORT 74C40061
CAP - 74B40065
SCREW MS16997 62 (4)
WASHER - MS 27183-10 (4)
SCREW - MS16998-33 (4)
WASHER - MS 35338-43 (4)
NUT - MS 35650-302 (3)
FRONT MOTOR MOUNT
74C40059
SCREW MS16998-95 (10)
WASHER - MS 35338-48 (10)
MOTOR (SEE NOTE 2)
PART NO SA A 52785
REAR MOTOR MOUNT
74C40060
SCREW MS16997-62 (3)
WASHER - MS 27183-10 (3)
GEAR DRIVE ASSY
19204
74F40066
2 APRIL 74
APPLICATION

Fig. 20

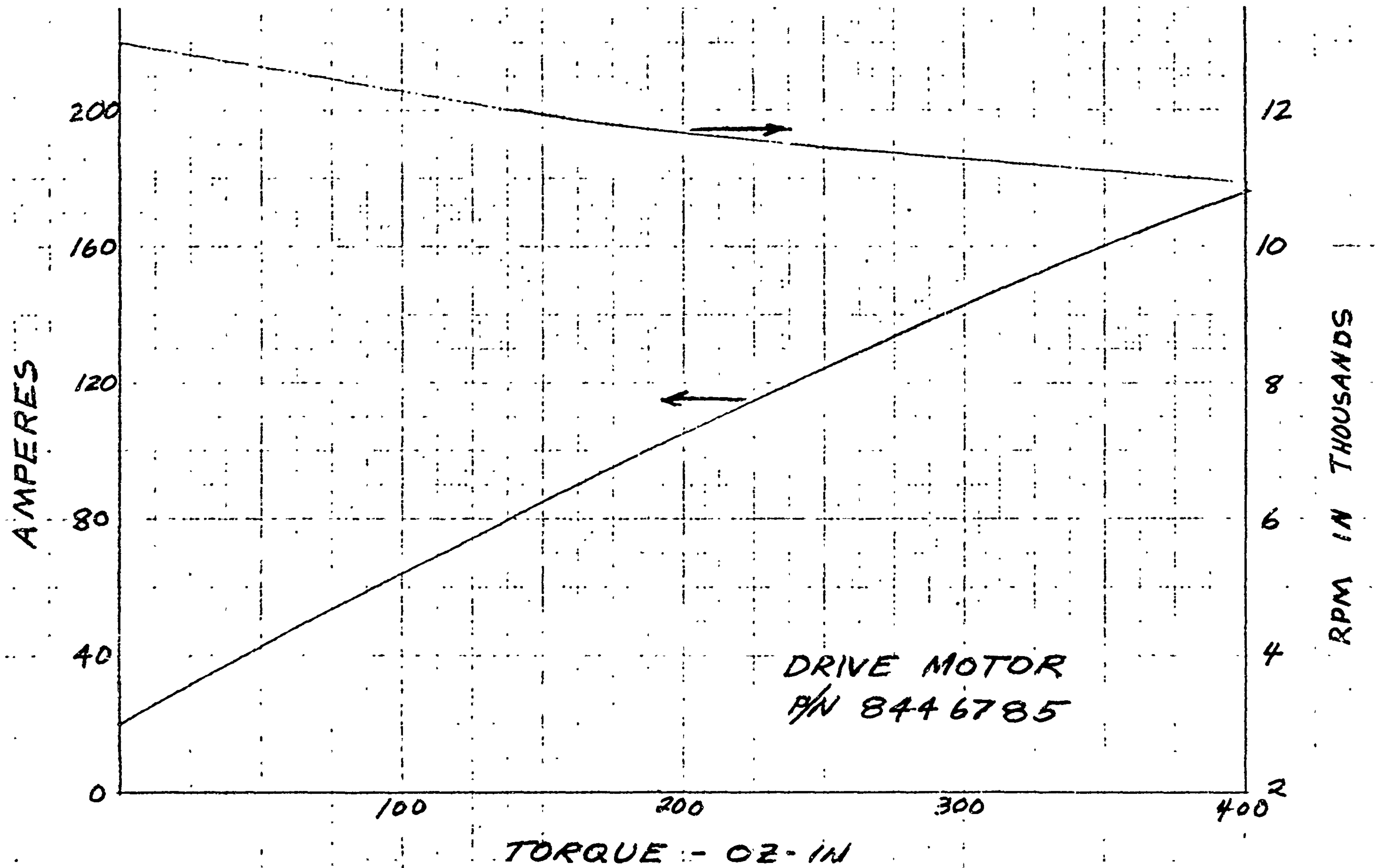


FIGURE 21

INPUT MOTOR TORQUE

3. GENERALIZED d'ALEMBERT FORCE METHOD

Obtaining the differential equation of motion for a dynamic system is obviously one of the important things that must be done in order to achieve the position description (the solution) of the system. Utilization of d'Alembert's principal (F-Mä=0) and virtual work arguments allows a derivation of the generalized d'Alembert equation, with constraints (5). The equation

$$\sum_{j=1}^{M} \vec{F}_j \cdot \frac{\partial a_j}{\partial q_i} - \sum_{\ell=1}^{L} \vec{\lambda}_\ell \cdot \frac{\partial \phi_\ell}{\partial q_i} = 0 \qquad (3.1)$$

explained more fully in Appendix 2 and Table 3.1, allows the methodical generation of the differential equation of motion. The d'Alembert equation as expressed in Eq. (3.1) handles, for a generalized coordinate set and any degree of freedom, external forces applied to the system, d'Alembert forces, and closed loop constraints. The formulation is not limited to linear or "linearized" motion, in fact, DRAM (6) is based on Eq. (3.1) and the DRAM program development is partly based on a need for a general program to facilitate computer aided design of large, linear or non-linear, displacement systems of the type found in most machines.

The d'Alembert equation (3.1) reduces to

$$\sum_{j=1}^{M} \vec{F}_j \cdot \frac{\partial a_i}{\partial \phi_2} = 0 \qquad (3.2)$$

for the AMCAWS 30 system (Appendix 2). The AMCAWS system is somewhat simple, having only one degree of freedom. Representation of the AMCAWS system as single degree of freedom is achieved by the representation of the various weapon cams as motion generators. Another simplifying factor is that AMCAWS is essentially a two dimensional system.

Since there are over twenty effective forces that must be considered, an explanation of the procedure used to obtain the differential equation of motion using the AMCAWS weapon as an example would be unnecessarily detailed. The example chosen for illustration is the simple pendulum shown in Figure 22.

The pendulum is a one degree of freedom system in two dimensions. Eq. (3.2) holds and using the terms indicated in Figure 22 Eq. (3.2) can be written

$$\sum_{j=1}^{3} \vec{F}_j \cdot \frac{\partial \vec{a}_j}{\partial \Theta_2} = 0 \tag{3.3}$$

There are three effective forces (rotational and translational d'Alembert forces and gravity) acting on the pendulum bar, hence j = 1,2,3. $\vec{a}_j$ is a vector from some point in ground to the point of application of the $\vec{F}_j$ being considered. Θ_2 is the angle of the bar measured as indicated in the figure and is, of course, the degree of freedom.

The blow-by-blow procedure of determining the differential equation of motion is a relatively straightforward application of vector analysis.

For j=1, $\vec{F}_1$ is the rotational d'Alembert force $-I_2\ddot{\Theta}_2\,\hat{k}$.

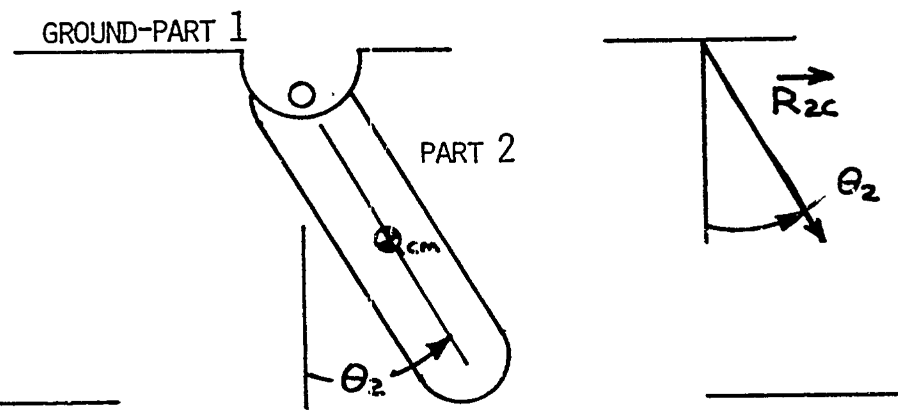
$$\vec{F}_1 = -I_2\ddot{\Theta}_2\,\hat{k} \tag{3.4}$$

$\ddot{\Theta}_2$ is the second time derivative of Θ_2 (angular acceleration in two dimensions). $\hat{k}$ is the unit vector about which the rotation takes place and is defined in Figure 22 (all the coordinates in this report are right-handed). I_2 is the moment of inertia of part 2 about the center of rotation, point 0. Note that part 1 is, by convention, ground.

The point of application from some point in ground to the point of application of $\vec{F}_1$ is $\vec{a}_1$ with

$$\vec{a}_1 = \vec{const} + \Theta_2\,\hat{k} \tag{3.5}$$

SIMPLE PENDULUM



GROUND IS PART 1

PENDULUM IS PART 2

TABLE 1

J	$\vec{F}_J$	$\vec{A}_J$	$\frac{\partial \vec{A}_J}{\partial \theta_2}$	$F_J \bullet \frac{\partial \vec{A}_J}{\partial \theta_2}$
1	$-M\ G\hat{J}$	$\vec{R}_{2c}$	$(\hat{K} \times \vec{R}_{2c})$	$-R_{2c} M_2 G \sin\theta_2$
2	$-M\ \ddot{\vec{R}}_{2c}$	$\vec{R}_{2c}$	$(\hat{K} \times \vec{R}_{2c})$	$-R_{2c} M_2 \ddot{\theta}_2$
3	$-I_2\ \ddot{\theta}_2$	$\theta_2\ \hat{K}$	$\hat{K}$	$-I_2\ \ddot{\theta}_2$

SUMMATION OF THE DOT PRODUCT TERMS

$$-(I_2 + M_2 R_{2c}^2)\ddot{\theta}_2 - M_2 R_{2c} G \sin\theta_2 = 0$$

AND AFTER REARRANGEMENT AND SUBSTITUTION (RHS -- RIGHT HAND SIDE)

$$[A]\{\ddot{\theta}\} = \{RHS\}$$

FIGURE 22

SIMPLE PENDULUM

The partial derivative with respect to θ_2 is

$$\frac{\partial \vec{a}_1}{\partial \theta_2} = \hat{k}. \qquad (3.6)$$

The dot product is then

$$\vec{F}_1 \cdot \frac{\partial \vec{a}_1}{\partial \theta_2} = -I\ddot{\theta}_2 \qquad (3.7)$$

for j=2, $\vec{F}_2$ is the translational d'Alembert force $-m\ddot{\vec{p}}_{2c}$. $\ddot{\vec{p}}_{2c}$ is the second time derivative of a vector from a point in ground to the center of mass of the bar.

Since

$$\vec{p}_{2c} = \vec{const} + \vec{r}_{2c} , \qquad (3.8)$$

$\ddot{\vec{p}}_{2c}$ is identically $\ddot{\vec{r}}_{2c}$. The point of application of $\vec{F}_2$ from the ground point is $\vec{a}_2$, with

$$\vec{a}_2 = \vec{const} + \vec{r}_{2c} \qquad (3.9)$$

and *

$$\frac{\partial \vec{a}_2}{\partial \theta_2} = (\hat{k} \times \vec{r}_{2c}). \qquad (3.10)$$

$\ddot{\vec{r}}_{2c}$ must be expanded, eventually into $\hat{i}$ and $\hat{j}$ components. In two dimensions,

$$\ddot{\vec{r}}_{2c} = \ddot{r}_{2c}\hat{r}_{2c} + 2\dot{r}_{2c}\dot{\theta}_2 (\hat{k} \times \hat{r}_{2c}) - \dot{\theta}_2^2 \vec{r}_{2c} + \ddot{\theta}_2 (\hat{k} \times \vec{r}_{2c}) \qquad (3.11)$$

Since the bar is of fixed length there is no change in length in time and

$$\dot{r}_{2c} = \ddot{r}_{2c} = 0 \qquad (3.12)$$

---

* $\frac{\partial \vec{r}_{2c}}{\partial \theta_2} = \frac{\partial \vec{r}_{2c}}{\partial t} \frac{\partial t}{\partial \theta_2} = \frac{\dot{\vec{r}}_{2c}}{\dot{\theta}_2} = \frac{\dot{\theta}_2(\hat{k} \times \vec{r}_{2c})}{\dot{\theta}_2} = (\hat{k} \times \vec{r}_{2c})$ where $\vec{r}_{2c}$ is inextensiable and moves entirely in a plane.

yielding for Eq. (3.11)

$$\ddot{\vec{r}}_{2c} = \ddot{\theta}_2 \; (\hat{k} \times \vec{r}_{2c}) - \dot{\theta}_2^2 \vec{r}_{2c}. \tag{3.13}$$

Thus

$$\vec{F}_2 = -m_2\{\ddot{\theta}_2(\hat{k} \times \vec{r}_{2c}) - \dot{\theta}_2^2\vec{r}_{2c}\} \tag{3.14}$$

and the dot product becomes

$$\vec{F}_2 \cdot \frac{\partial \vec{a}_2}{\partial \theta_2} = -mr_{2c}^2\ddot{\theta}_2. \tag{3.15}$$

for j=3, $\vec{F}_3$ is the gravity force. The gravity field is considered to act at the mass center with

$$\vec{F}_3 = -mg\hat{j}. \tag{3.16}$$

The point of application (a_3) is vector a_2. The dot product for j=3 is then

$$\vec{F}_3 \cdot \frac{\partial \vec{a}_3}{\partial \theta_2} = \vec{F}_3 \cdot \frac{\partial \vec{a}_2}{\partial \theta_2} = (-mg\;\hat{j}) \cdot (\hat{k} \times \vec{r}_{2c}) = -r_{2c}mg \sin \theta_2 \tag{3.17}$$

The sum of the right hand sides of equations (3.7), (3.15), and (3.17), when set to zero, is the differential equation of motion for the pendulum system of Figure 28.

$$\sum_{j=1}^{3} \vec{F}_j \cdot \frac{\partial \vec{a}_j}{\partial \theta_2} = -I\ddot{\theta}_2 - mr_{2c}^2\ddot{\theta}_2 - r_{2c}mg \sin \theta_2 = 0 \tag{3.18}$$

$$= -(I + mr_{2c}^2)\ddot{\theta}_2 - r_{2c}mg \sin \theta_2 = 0 \tag{3.19}$$

Eq. (3.19) is the differential equation of motion. Generally, in development of such an equation, translational d'Alembert forces and gravity forces are lumped, since they have the same point of application. A tabular form of bookkeeping, such as used throughout Appendix 2 and in Table 1, is useful in documenting the procedure without undue space or verbiage. Table 1 is the development of the differential equation of motion for the system of Figure 22 with the translational d'Alembert force and gravity force combined.

The equation of motion for this particular example is easily obtained with any number of other approaches. The simplicity of the generalized d'Alembert Force procedure claimed is not overwhelmingly apparent in a trivial example such as the pendulum, but for additional degrees of freedom or a single degree-of-freedom system with as many effective forces as the AMCAWS 30 the method provides an efficient well defined procedure for generating the differential equation of motion for dynamic systems.

4. MATHEMATICAL MODEL FOR AMCAWS 30

The AMCAWS 30 MM weapon system, while seemingly difficult to describe (Section 2), is easily modeled. This is because the model need not be as detailed as the description. Many parts can be lumped, other parts can be ignored, and some complex operating characteristics can be simplified (as long as accuracy is maintained).

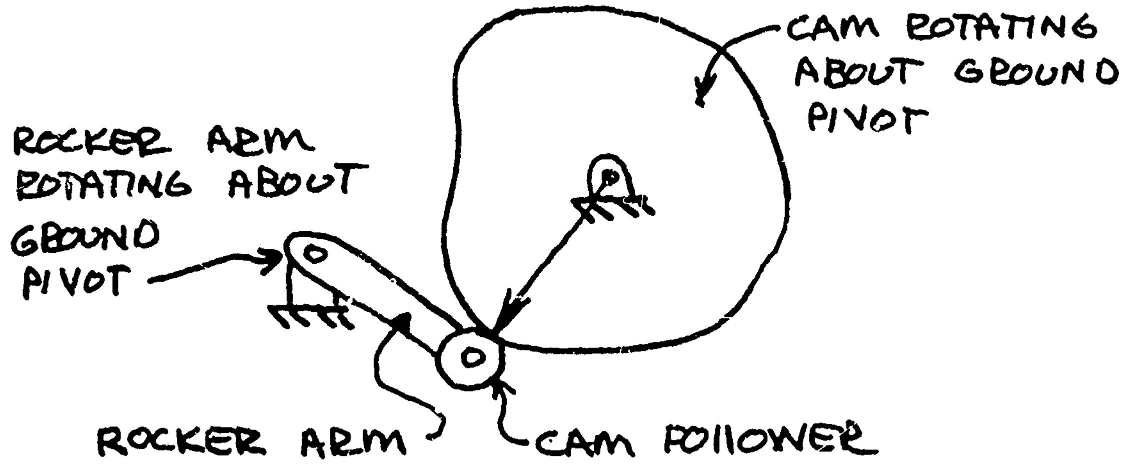
The component actions and interactions of major interest are those associated with feeding, ejecting, chambering, and locking. Drive motor torque requirements are also of major interest. A discussion of the simplifications and a defense of why some parts and components are not included is pertinent.

The greatest simplification in the model evolves from the fact that AMCAWS is one degree-of-freedom. The specification of the input angle (which is, again, the degree-of-freedom) in turn specifies all the positions of the major components listed. The specification of the first and second time derivatives of the input motor in turn specifies all the component velocities and forces. This fact can be exploited by creating a table that allows each component's position to be described as a function of the input angle. This has been done and is the Functional Relationships Computer Program (FRCP, Appendix 1). The FRCP, using the geometric constraints of the cams and followers, generates an extensive table of each component's position versus motor input angle. While the FRCP is more fully discussed in Appendix 1, briefly the program is a FORTRAN description of the weapon that has the positional table as primary output. The program traces through one firing cycle in 360 steps. For a given weapon geometric configuration (cam rises, follower arm lengths as opposed to a given weapon mass configuration), the FRCP need be run only once. The single degree of freedom allows an "uncoupling" of the major components and allows each one to be considered separately. The FRCP develops the positional response of each component with respect to the degree-of-freedom (input angle). The tables enable the dynamic program (Appendix 3) to treat each component as essentially a separate problem completely unrelated to any other component. Appendix 2, which is the detailed development of the differential equation of motion for the AMCAWS-30, does exactly that. The simplification is that each component has a "local" degree-of-freedom which is explicitly a function of the motor input angle. The terms contributing to the equation of motion are easily identified and calculated in terms of the local coordinate. As a final step the dot product terms of the components are expressed in terms of the motor input angle. The dynamic effects of an individual component on the rest of the system are correctly accounted for when the terms from each component are summed, but this interaction need not be considered when developing the terms for the individual component.

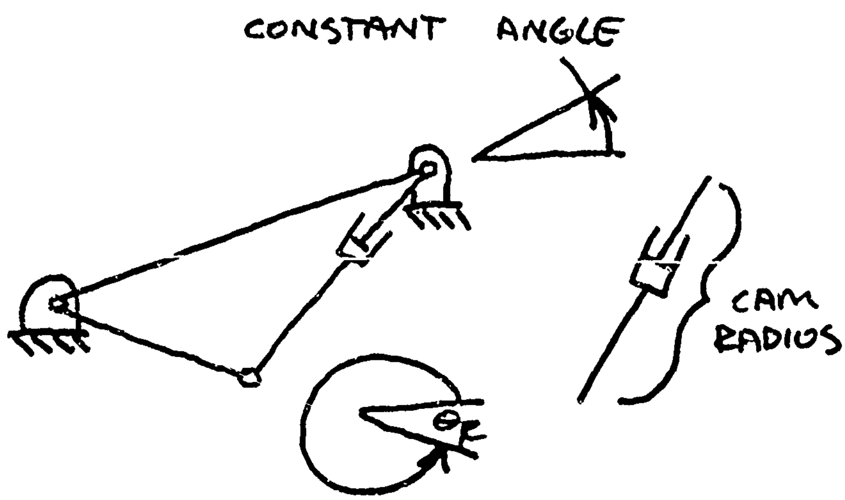
A second simplification occurs in the FRCP itself. The feed system (Figure 7) is composed of a face cam path of some width, a roller bearing follower of diameter slightly less than the cam path width and the rocker arm which transmits motion through the shaft to the dual feed pawls. The cam path to rollerbearing contact is an example of contact between higher pairs (as contrasted with lower pairs) and position, velocity, and force solution are not trivial (3,4) in terms of difficulty of incorporation into the dynamic model itself and CPU time of running. While the higher order pairs (all the cam contacts) could all be simulated with the generalized d'Alembert force procedure (4) the increase in accuracy of the positional solution and the velocity and force solutions is not felt to be significant enough to warrant inclusion at this time. The simplification made is, for the feed and eject sides, a pinned linkage (shown in Figure 23). The length of bar s corresponds to the rise of the cam for the given rotation. The internal angles of the linkage are computed and eventually all the angles of Figure 23 can be computed. The positional solution for the feed and eject components are achieved in this manner. The velocity and acceleration solutions are accomplished in the dynamic model. Part to part forces, such as cam path to roller bearing are determined as the result of force equilibrium. The recasting of the higher pair feed and eject contacts into lower pair pinned joints is an important simplification.

There are other higher pair contacts that have been recast into more easily solved problems. The drum cam-chamber motion is through a follower on the stud. The FRCP treats chamber motion entirely as a result of a displacement function, the function itself being the rise versus rotation data on the drum cam drawing. The gear sets in the drive assembly are higher order pairs. The common simplification of assuming no losses through an individual set and assigning a total loss proportional to the power transmission through the entire assembly is made in the dynamic model. The live round being fed into the chamber is a higher pair contact, since the round slides on the feed pawl surface during the load cycle. This pair has been ignored by placing the actual ammunition mass at the pawl end during the time the round is being placed into the chamber and setting that mass to zero while the feed pawls retract. The lump cam-clevis pair for the locking ring is a higer order pair. The position solution for this pair was achieved by a 10x cardboard model of the cam surfaces. Quite sophisticated.

Two major assemblies are not explicitly in the model. The bolt assembly appears as a translating mass in the chamber routines and the sear spring compression is treated as a spring force acting on the chamber. The firing



- GENERAL CAM MECHANISM -



SIMPLIFIED CAM
- PINNED LINKAGE -

FIGURE 23
CAM CONTACT "PINNED LINKAGE"

pin travel after sear off is not pertinent to the overall model and is not modeled. The buffer assembly is not treated at all. This, in effect, ignore any recoil of the weapon due to the round impulse. The recoil of the weapon is an independent degree-of-freedom that could be included at a later date if deemed necessary. Firing pin motion could also be treated as even another degree-of-freedom. Neither of these two possible additional degrees of freedom is important to the major parameters of current interest (feeding, ejecting, chambering, locking, torque).

The power input into the weapon system is modeled as a torque versus RPM curve (Figure 21) for the drive motor. A table lookup yields the torque input to the system from the drive motor for any specified RPM. Evaluating different motors is only a matter of substituting the different torque curves.

An extremely important aspect of the model is the drag forces and other sources of power loss through the weapon. These are not yet implement

The mathematical model is the differential equation of motion for the system described. The complete equation for that system is

$$
\begin{aligned}
\ddot{\theta}_2 \quad & -I_{59} - C^2 I_{120} \\
& -(\Theta_3')^2 (I_{drum} + I_{face}) \\
& -(\Theta_4')^2 [FI_{pawl} + FI_{rock} + FI_{shaft} + FM_{pawl}(FP_{cm})^2 \\
& \qquad + FM_{rock}(FR_{cm})^2 + FM_{ammo}(FP_e)^2] \\
& -(\Theta_5')^2 [EI_{pawl} + EI_{rock} + EI_{shaft} + EM_{pawl}(EP_{cm})^2 \\
& \qquad + EM_{rock}(ER_{em})^2] \\
& -(\Theta_6')^2 (I_{lock}) \\
& +(R_7')^2 (VCHMBR)
\end{aligned}
$$

$$+\dot{\Theta}_2^2 \quad -\Theta_3' \, \Theta_3''(I_{drum} + I_{face})$$

$$-\Theta_4' \, \Theta_4''[FI_{pawl} + FI_{rock} + FI_{shaft}$$

$$+FM_{pawl}(FP_{cm})^2 + FM_{rock}(FR_{cm})^2$$

$$+FM_{ammo}(PE)^2]$$

$$-\Theta_5'\Theta_5''[EI_{pawl} + EI_{rock} + EI_{shaft}$$

$$+EM_{pawl}(EP_{cm})^2 + EM_{rock}(ER_{cm})^2]$$

$$-\Theta_6'\Theta_6''(I_{lock})$$

$$+R_7'R_7'' \, (V_{chmbr})$$

$$+ T_{motor} - C_{motor}$$

$$-\Theta_4'[EM_{pawl}(EP_{cm}) \, g \cos\Theta_4 - EM_{rock}(ER_{cm}) \, 9 \cos\Theta_{rf} + T_{feed})$$

$$-\Theta_5'[EM_{pawl}(EP_{cm}) \, g \cos\Theta_5 + EN_{rock}(ER_{cm}) \, 9 \cos\Theta_{rf}]$$

$$-\Theta_6'(T_{lock})$$

$$-R_7'[CRUSH - SEAR + C_{fchmbr}(R'\dot{\Theta}_2/ABS(R'\dot{\Theta}_2))]$$

$$= 0 \qquad (4.1.a)$$

which is rewritten

$$a\ddot{\Theta} + b\dot{\Theta}^2 = 0 \qquad (4.1.b)$$

This ordinary differential equation is integrated numerically using the HPCG routine out of the IBM-SSP library (all the routines used by the FRCP and the dynamic program are included with their respective listings). HPCG uses Hamming's predictor-corrector method coupled with a Ralston modified Runge-Kutta procedure for start-up values (7). HPCG is quite general because it requires two user supplied external subroutines FCT and OUTP. FCT is the routine that must evaluate the constants of Eq. (4.1.b) and OUTP is the output vehicle for HPCG.

The construction of the dynamic program is shown in Figure 24. The OUTP and FCT blocks are expanded in Figures 24 and 25, respectively. Both the dynamic program and the FRCP are reasonably well documented and so a more

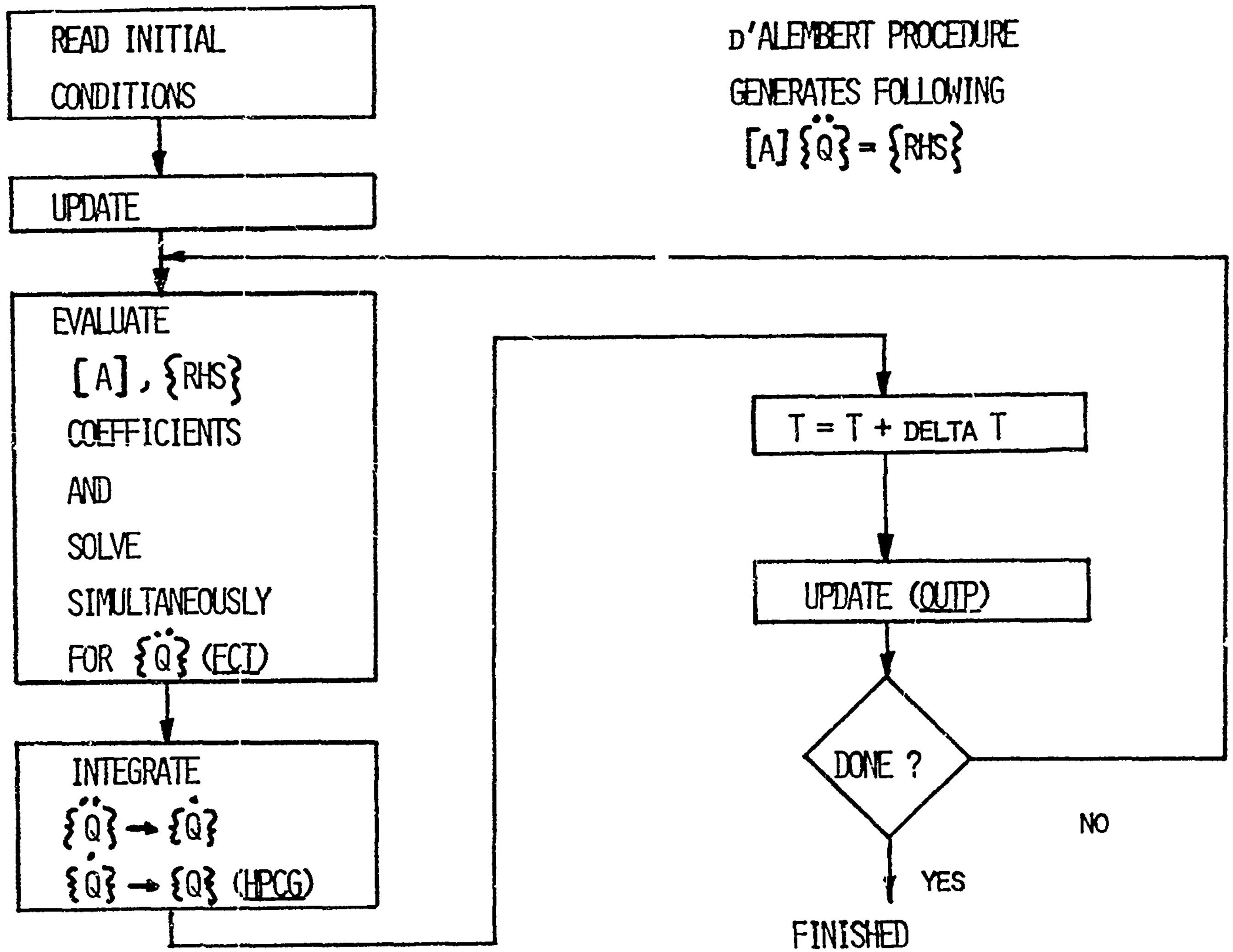


FIGURE 24

DYNAMIC PROGRAM BLOCK DIAGRAM

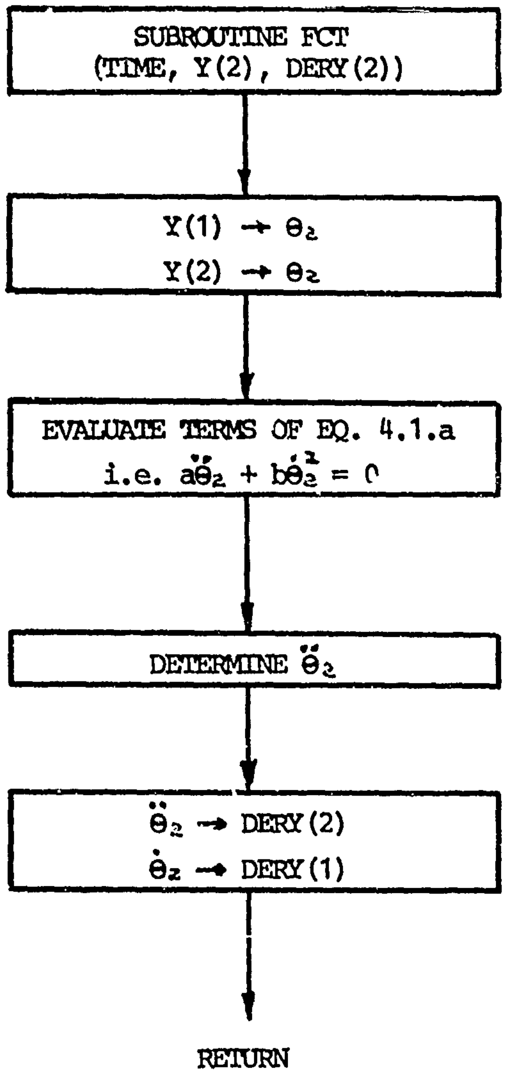


FIGURE 25

FCT BLOCK DIAGRAM

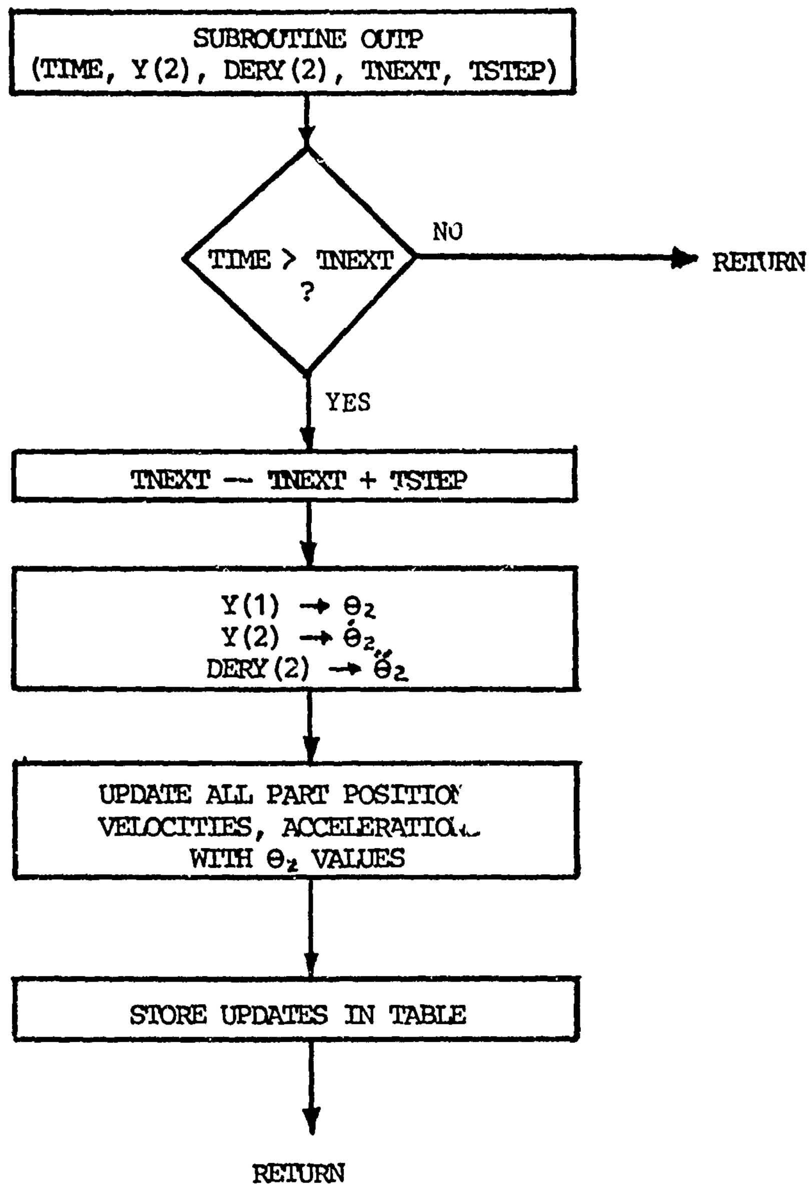


FIGURE 26

OUTP BLOCK DIAGRAM

detailed description of the program can be found in Appendix 4 and 1. The program has some interactive capability in input which was used during initial development on MTS (Michigan Terminal System, University of Michigan) but difficulty of doing interactive work and limited disk availability at AVSCOM almost demands the dynamic program be run with its batch default.

The batch default causes the dynamic program to have the following initial conditions:

Initial motor input angle position	= 0
Initial motor input velocity	= 0
Initial motor input acceleration	= 0
Initial time	= 0
Final time	= 2.5 sec

Output steps every .01 seconds

Figures 27 through 29 show the response of the motor input angle for a typical run.

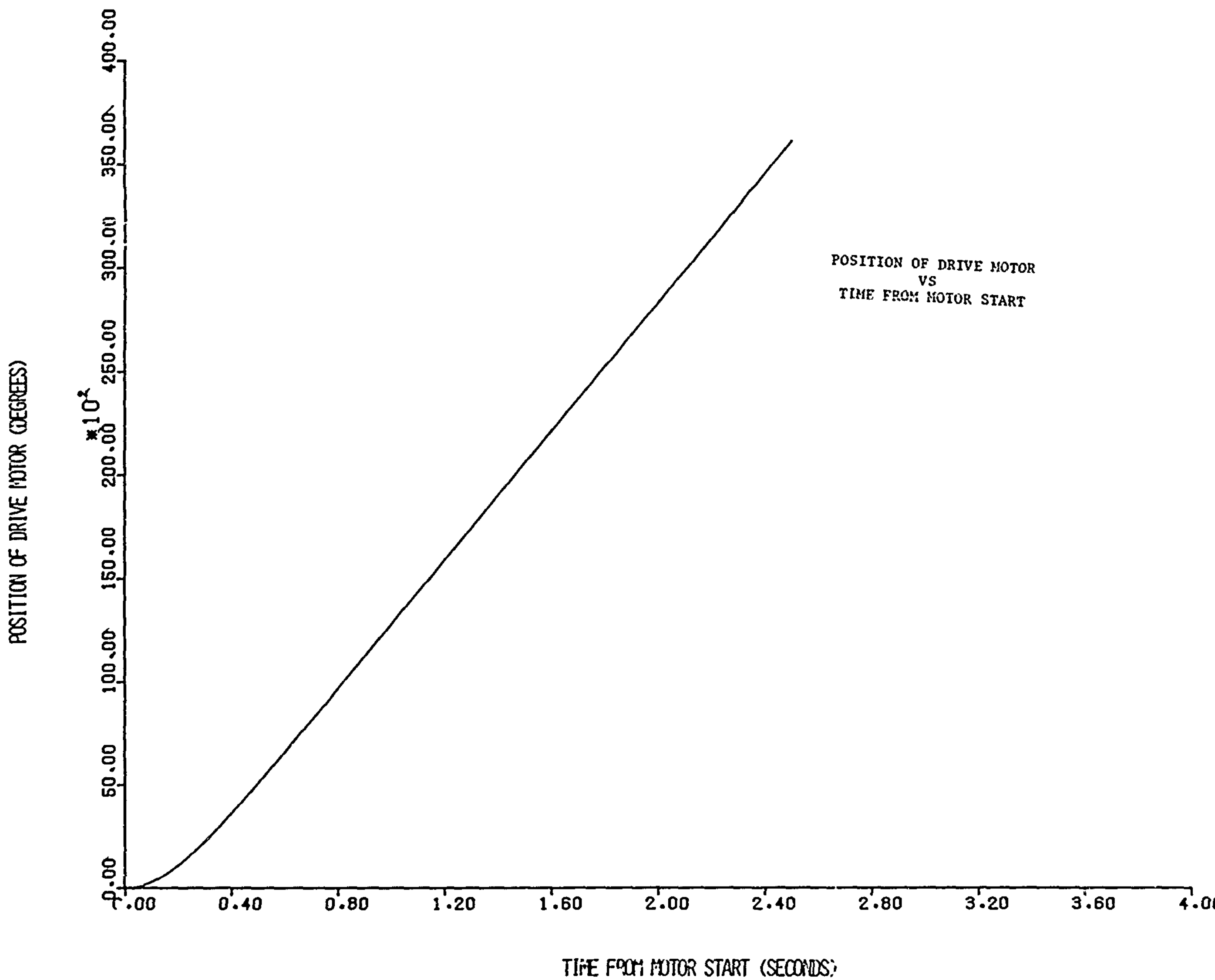


FIGURE 27
DRIVE MOTOR POSITION VS TIME

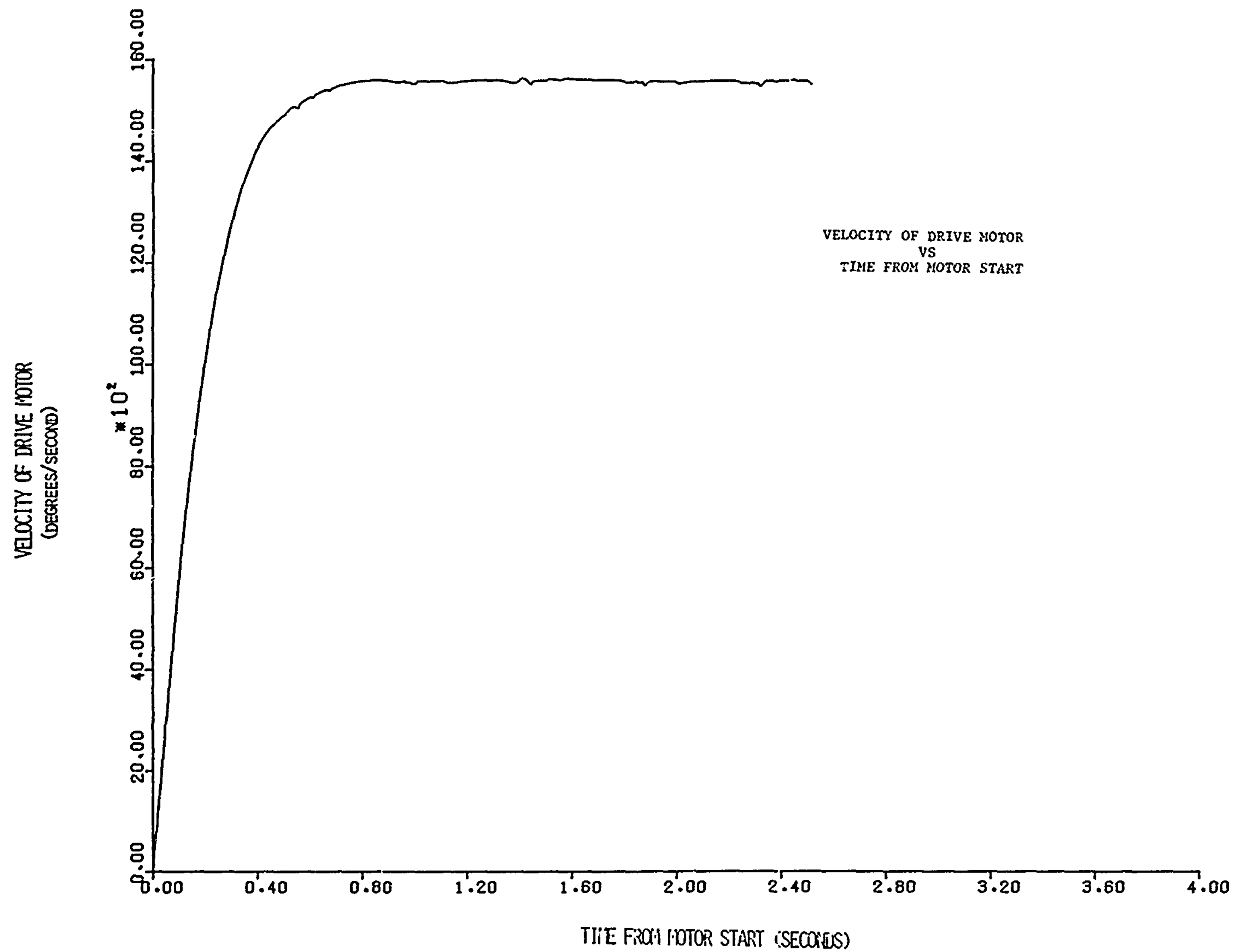


FIGURE 28
DRIVE MOTOR VELOCITY VS TIME

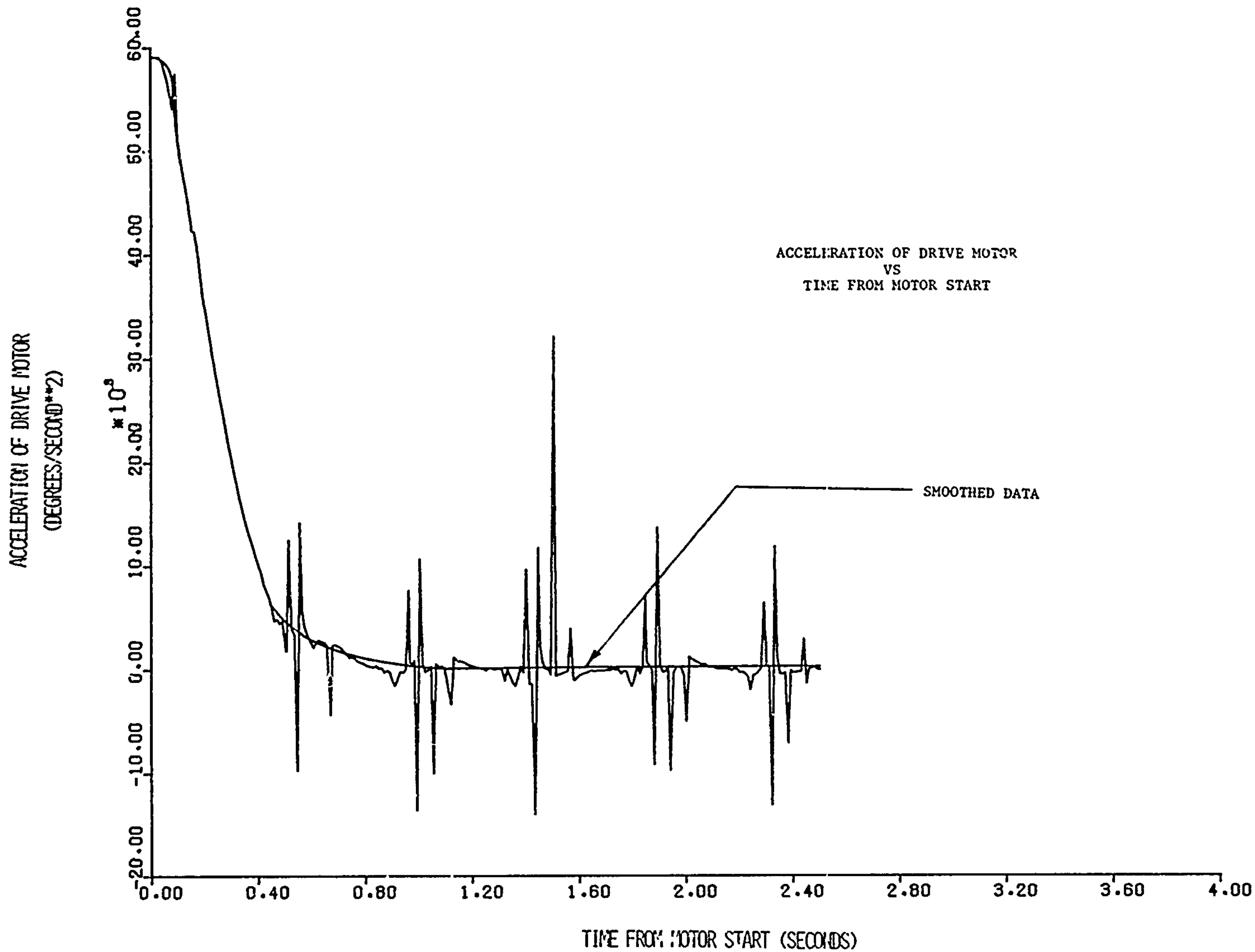


FIGURE 29
DRIVE MOTOR ACCELERATION VS TIME

5. MODEL INPUT

The mathematical model for the AMCAWS-30 uses as input a table of position solutions for each major component versus input motor angle. The range of values for the table is such that one complete firing cycle is described. The table is shown in Table 5.1 along with a description of the element entries. The table, although on disk file, is constructed as if it were on 80 character cards and thus the resulting 2 card groups seen in Table 5.1 Format for the table is format (' ', I5, 3F16.4) and format (' ', I5,4F16.4) for the two lines. This position table is the only input necessary to the AMCAWS-30 mathematical modeling program.

The table itself is, as discussed, generated by the functional relationships computer program (FRCP) which is listed in Appendix A-3. The inputs here consist of smoothed drawing data for the single turn cam AMCAWS weapon. The data required for the FRCP is data for the feed cam, eject cam, drum cam and lock cam. This data is taken from the engineering drawings for these components. The drawing data was tabulated and read into the FRCP with a

Format (2F17.4)

statement.

There are no other inputs to either program.

Since the mathematical model program is based on its numerical integration routine, it is essential that the input data have no discontinuities about which the integrating routine would cycle and ultimately stop. This did occur with the original drawing data and thus the "sharp" corners were smoothed with a cubic fitting routine that forced a match in the zero and first derivatives at the end points of each region to be smoothed. The smoothing is done on the drawing data and thus the table output is also smoothed. Comparisons between original and smoothed drawing data can be seen in Figures 30, 31 and 32. The drum cam data was well behaved enough to use without smoothing and is shown in Figure 33.

The smoothing routine and program listing can be found in Appendix A-5.

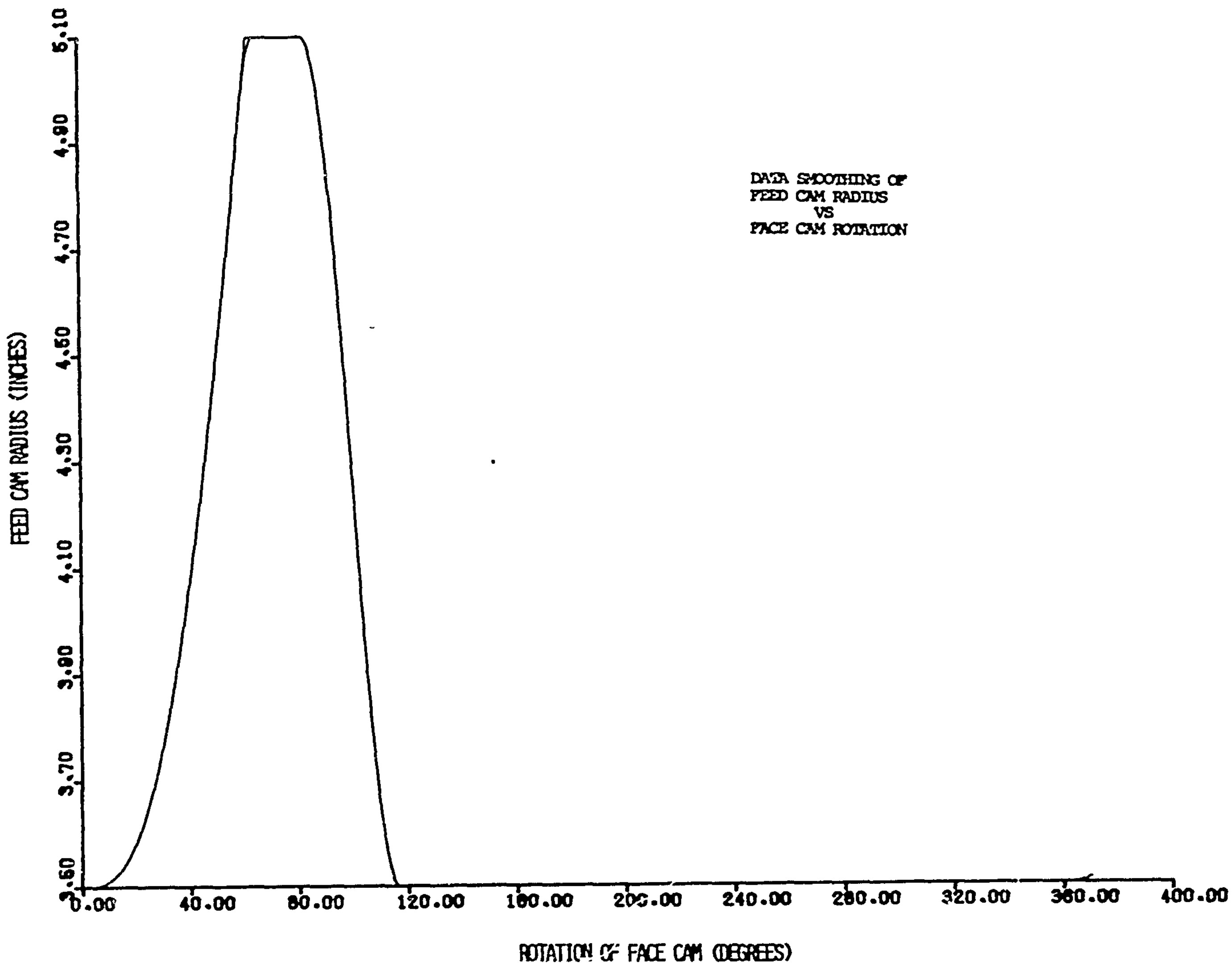


FIGURE 30
SMOOTHED FEED DATA

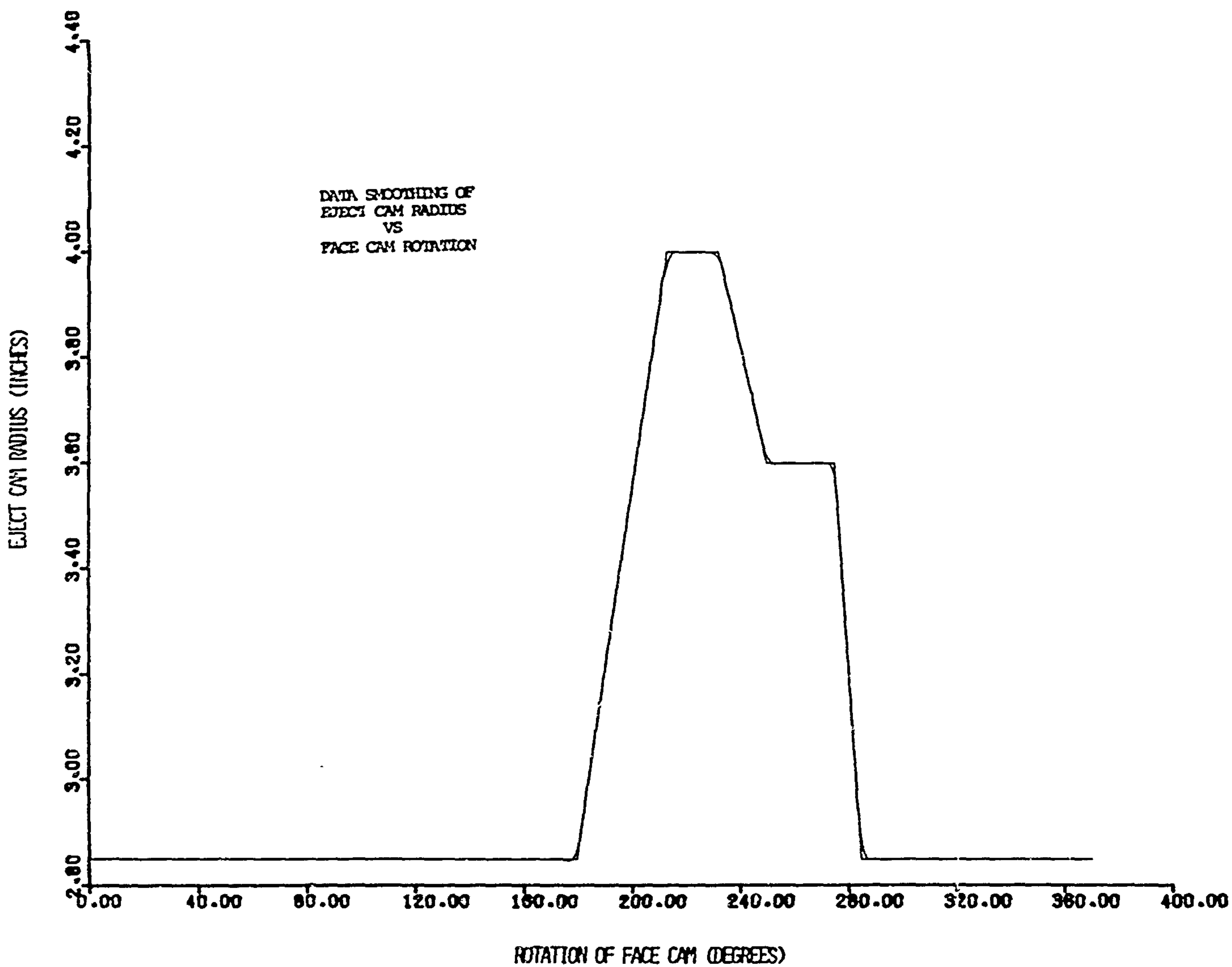


FIGURE 31
SMOOTHED EJECT DATA

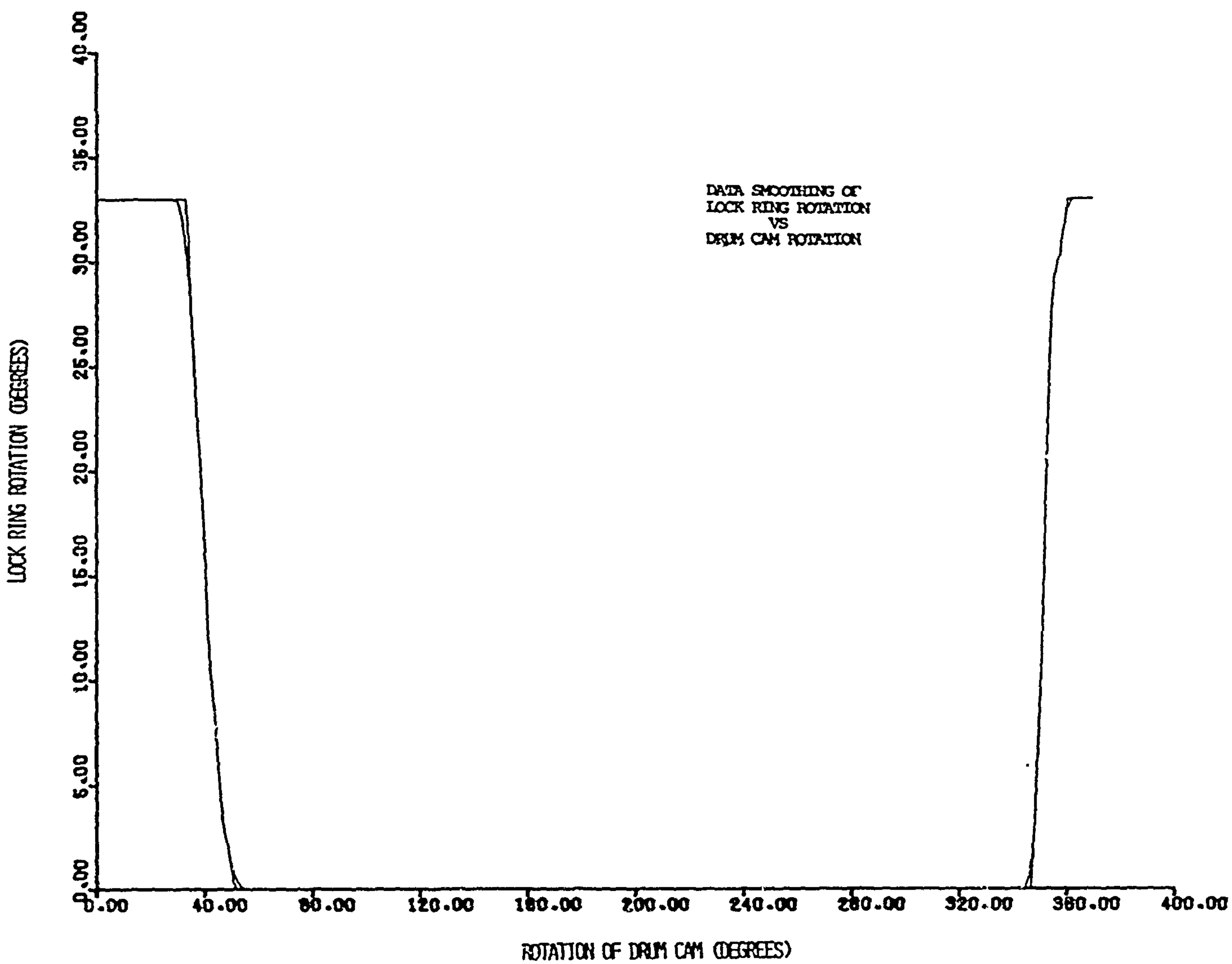


FIGURE 32
SMOOTHED LOCK DATA

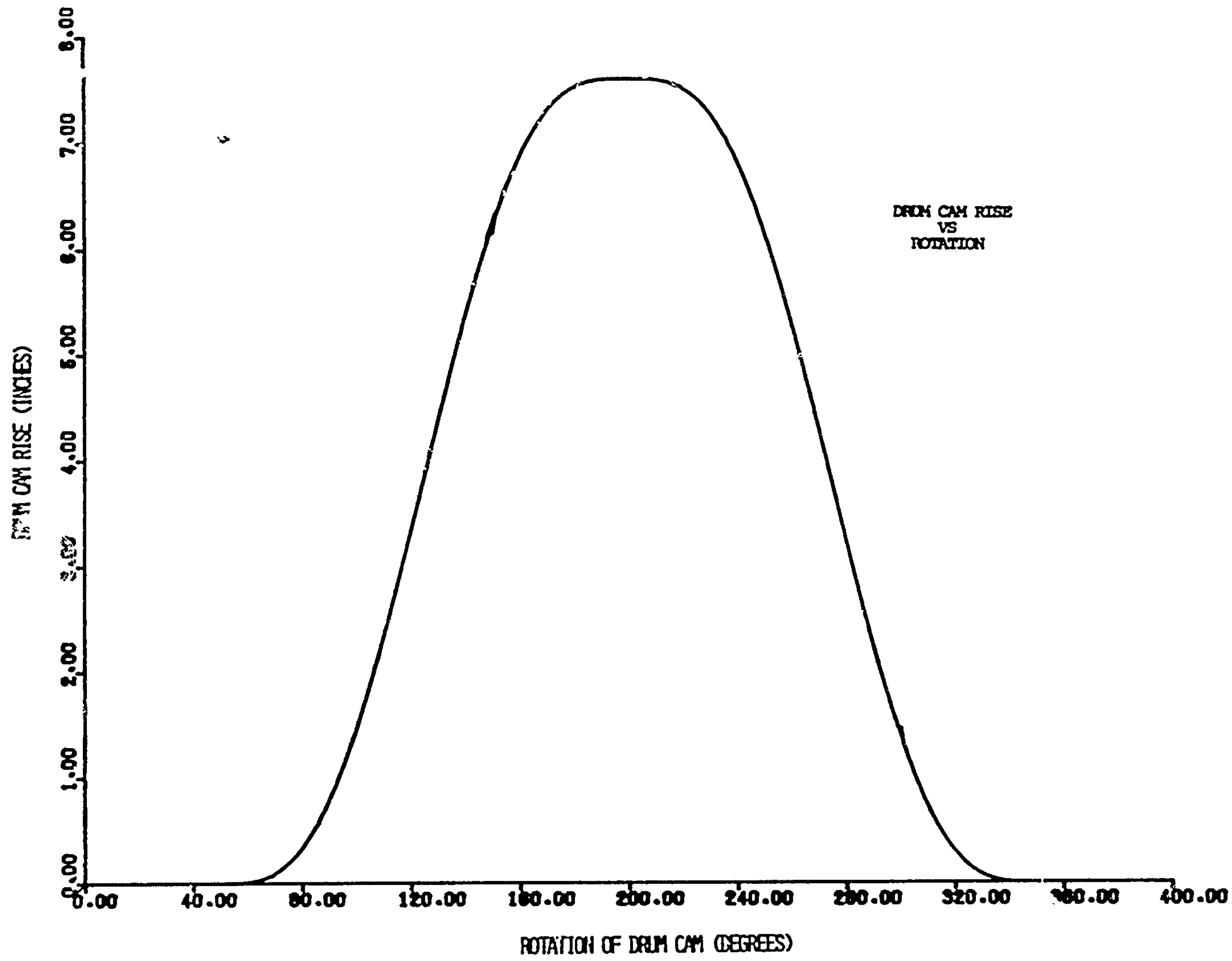


FIGURE 33
DRUM CAM DATA

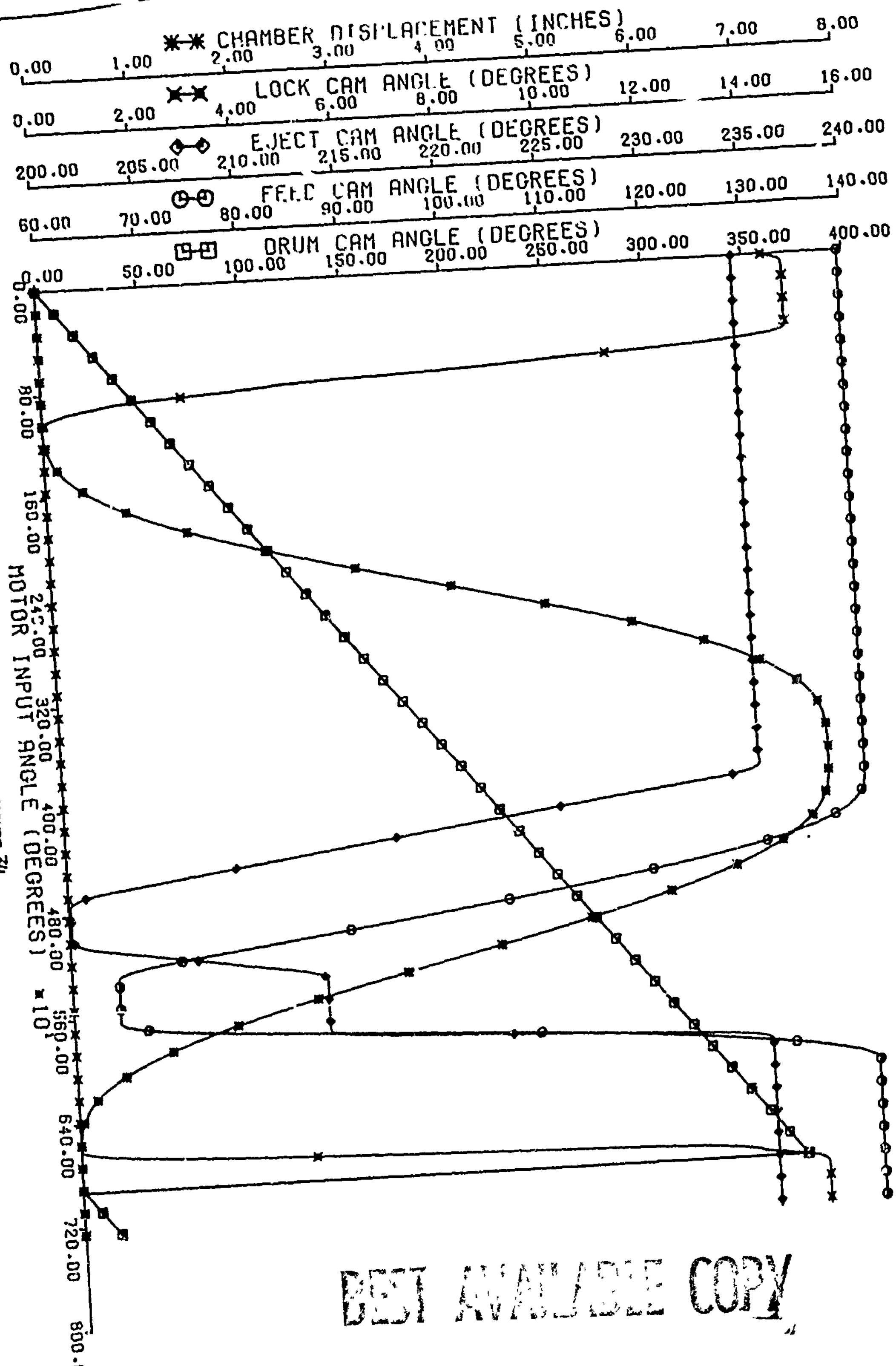


FIGURE 34

WEAPON TIMING

6.0 CONCLUSIONS

The three primary objectives of this report have been largely met. The computer programs that comprise the model package have been well documented. The modeling procedure itself has been explained in detail. The operating characteristics of the AMCAWS-30 weapon have been documented. As a bonus, the dynamic model itself seems to work well and the results seem to be similar to the actual weapon.

Figures 27, 28, and 29 are graphical output showing the response of the input drive motor gear over the 0.0 to 2.5 second time range considered. The spikes in the acceleration curve are certainly not present in the actual weapon and might be induced in the interpolation routine or by the highly discreet nature of the various fast acting cams. The model is not "correct" in the purest sense until the source of these spikes are tracked down and the source is either eliminated or justified. The smoothed data line for the acceleration curve represents the more realistic situation.

The computer model package operates at some disadvantage. Since the cancellation of AMCAWS funding in late 1976 there has been understandably little interest or enthusiasm in verifying the model against the actual weapon. As such even the masses and inertias (Table A2-1) are the result of calculations and not measurements. While the calculated and actual values probably do not differ greatly that is a known source of model error. The weapon has not been built up and fired since this modeling project was started, although a testing program to aid in the model verification was planned.

This report then cannot document an extensive verification, although the position, velocity, and acceleration curves match rough data that could be found from early 1975 gun firings.

Hopefully, the d'Alembert procedure might serve as a basis for modeling efforts for other Army developmental weapons or mechanisms. The procedure is relatively easy to apply and program and the cost is reasonable. The dynamic program in this report was run during prime hours at AVSCOM S&E computers for about fifteen dollars.

REFERENCES

1. Scott, L. AMCAWS-30MM Ammunition Development and Evaluation, Final Report under Contract DAAA25-73-C-043 Hercules Inc., 1 August 1973.

2. Kane, M. R., Rotating Band Torques and Stresses on AMCAWS-30MM Copper Banded Projectiles, Technical Report, Aircraft & Air Defense Weapon's Systems Directorate of Gen. T. Rodman Labortorary, Rock Island Arsenal, Report R-TR-75-022, May 1975.

3. Chace, M. A., Dynamics of Machinery Systems - A Vector/Computer Oriented Approach, Prentice-Hall, 1971.

4. Kass, R. C. and Chace, M. A., "An Approach to the Simulation of Two-Dimensional Higher Pair Contacts," Proceedings of the Fourth World Congress on the Theory of Machines & Mechanisms, Newcastle upon Tyne, England, September 1975.

5. Chace, M. A., Published and unpublished Notes presented to University of Michigan M.E. 540 students, Fall 1975.

6. Chace, M. A. and Angell, J. C., Users Guide to DRAM (Dynamic Response of Articulated Machinery), Design Engineering Computer Aids, Department of Mechanical Engineering, University of Michigan, 3rd Edition, April 1975.

7. System/360 Scientific Subroutine Package, Versian III, IBM Application Program, Publication GH20-0205-4, Fifth Edition, August 1970.

8. Carnahan, B., Luther, H. A., and Wilkes, J. O., Applied Numerical Methods, John Wiley & Sons, 1969.

9. Rothbart, H. A., Cams, John Wiley & Sons, 1956.

10. Tann, P. W., Cam Design and Manufacture, The Industrial Press, 1965.

11. Merritt, H. E. , Gear Engineering, John Wiley & Sons, 1971.

APPENDIX 1

FUNCTIONAL RELATIONSHIPS COMPUTER PROGRAM

The Functional Relationships Computer Program (FRCP) specifies, as output, the positional relationships versus the input drive motor angle of the components in the AMCAWS-30 weapon. Since the AMCAWS-30 model is a one degree of freedom system and the degree of freedom is the input drive motor angle, specification of an input angle also specified the position of all the other gun components. Given that there are no part failures (the weapon model uses fantastic materials that cannot fail) these various positional relationships are unchanging.

The FRCP uses the various part geometries, cam paths, and assembly angles to evaluate each part position given the input motor angle. Figure A1-1 is the basic flow chart for the program. The programs needs as input the drawing data for the drum cam, the eject and feed cam paths for the face cam and the various offset angles at which the components were assembled. A incremental Θ is chosen and the program loops through the six function calls until one complete firing cycle is over. The results of each loop are tabulated.

The program itself is listed in Appendix 3 while the output from the program, in graphical form, is shown in Figures A1-2 thru A1-7. Figures A1-8 thru A1-11 are the unaltered cam drawing data in graphical form and are included here for report completeness.

Figures A1-12 and A1-13 illustrate the basis of program functions FUN43F and FUN53F.

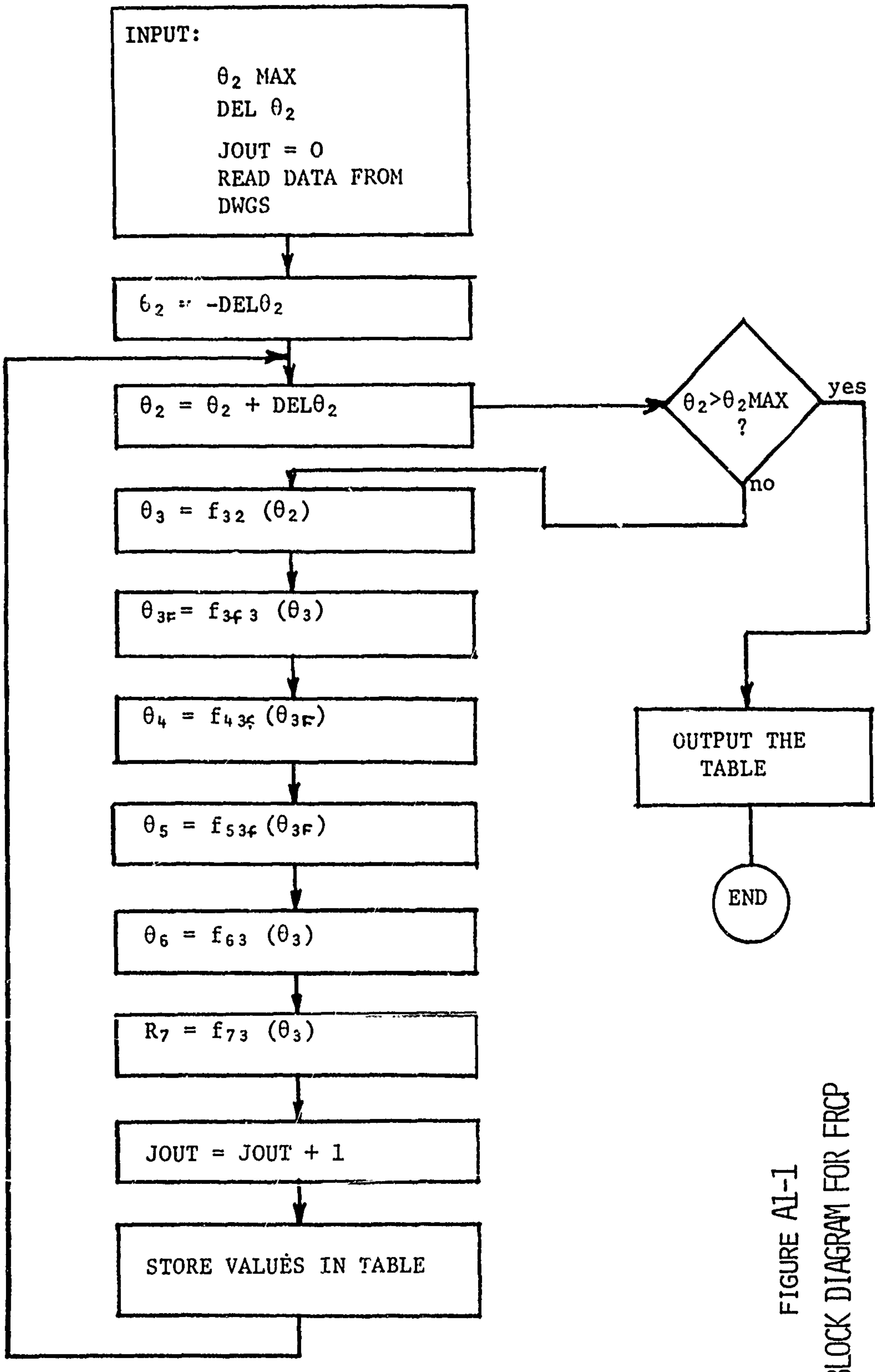


FIGURE A1-1
BLOCK DIAGRAM FOR FRCP

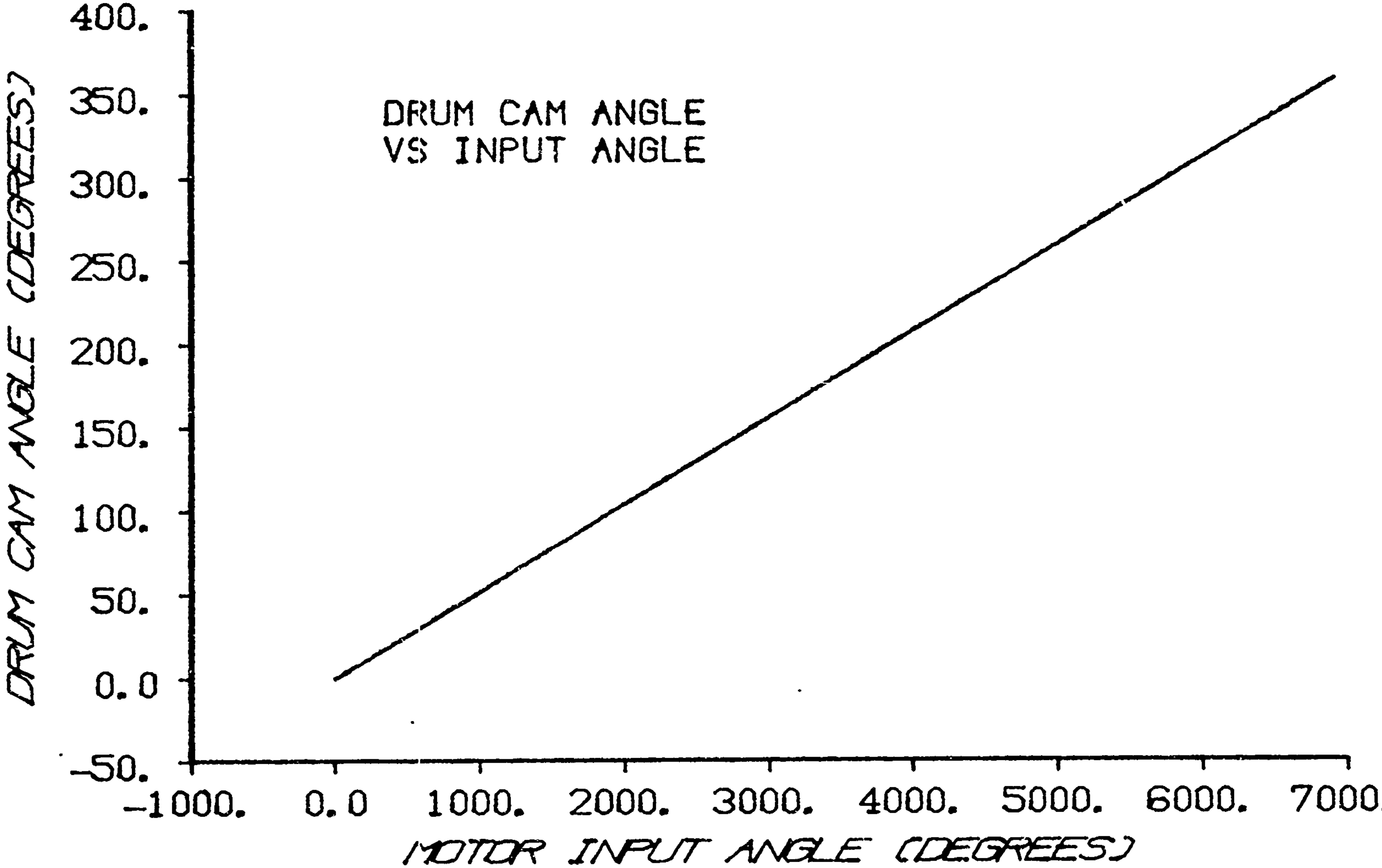


FIGURE A1-2

DRUM CAM ANGLE VS INPUT ANGLE

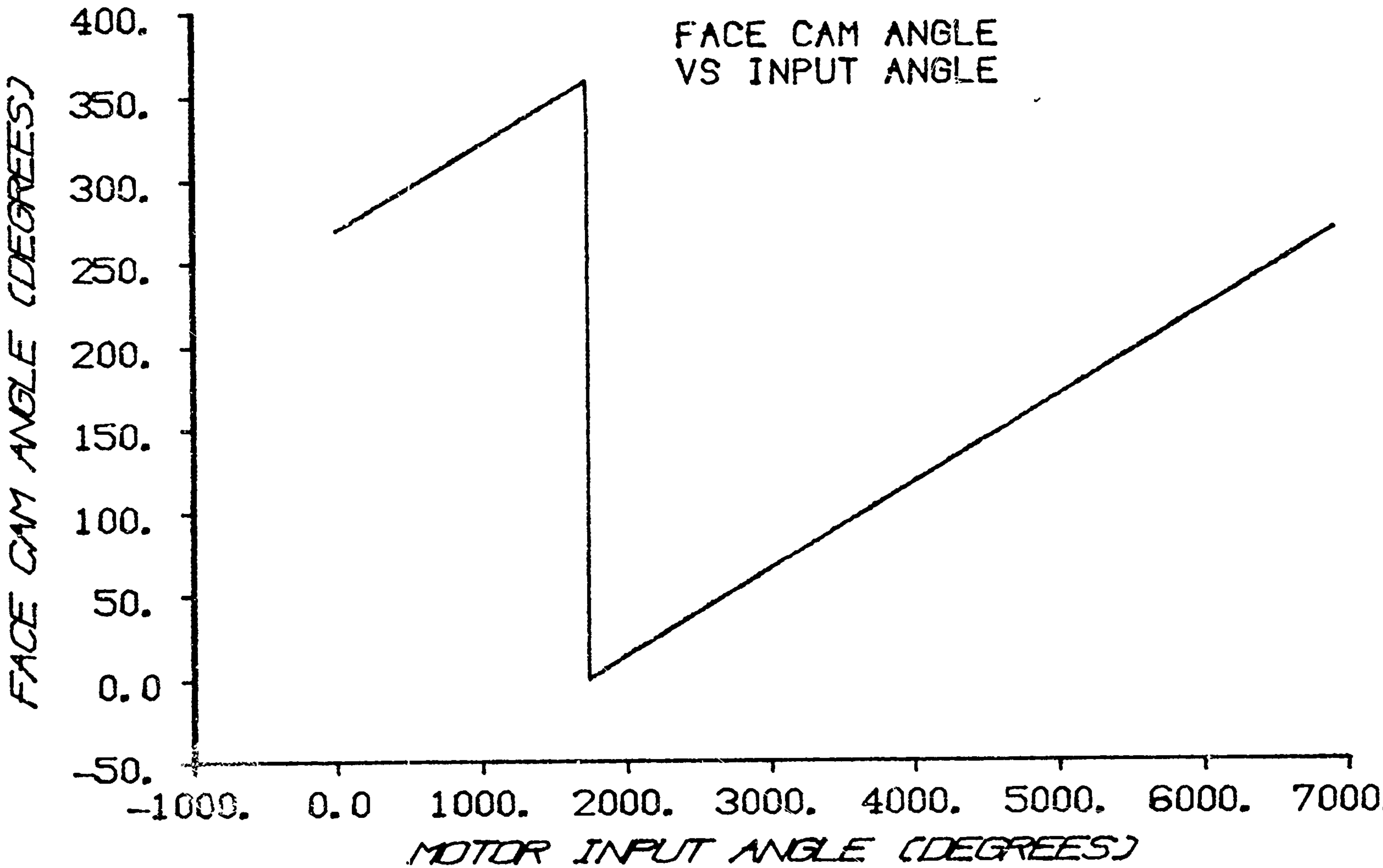


FIGURE A1-3

FACE CAM ANGLE VS INPUT ANGLE

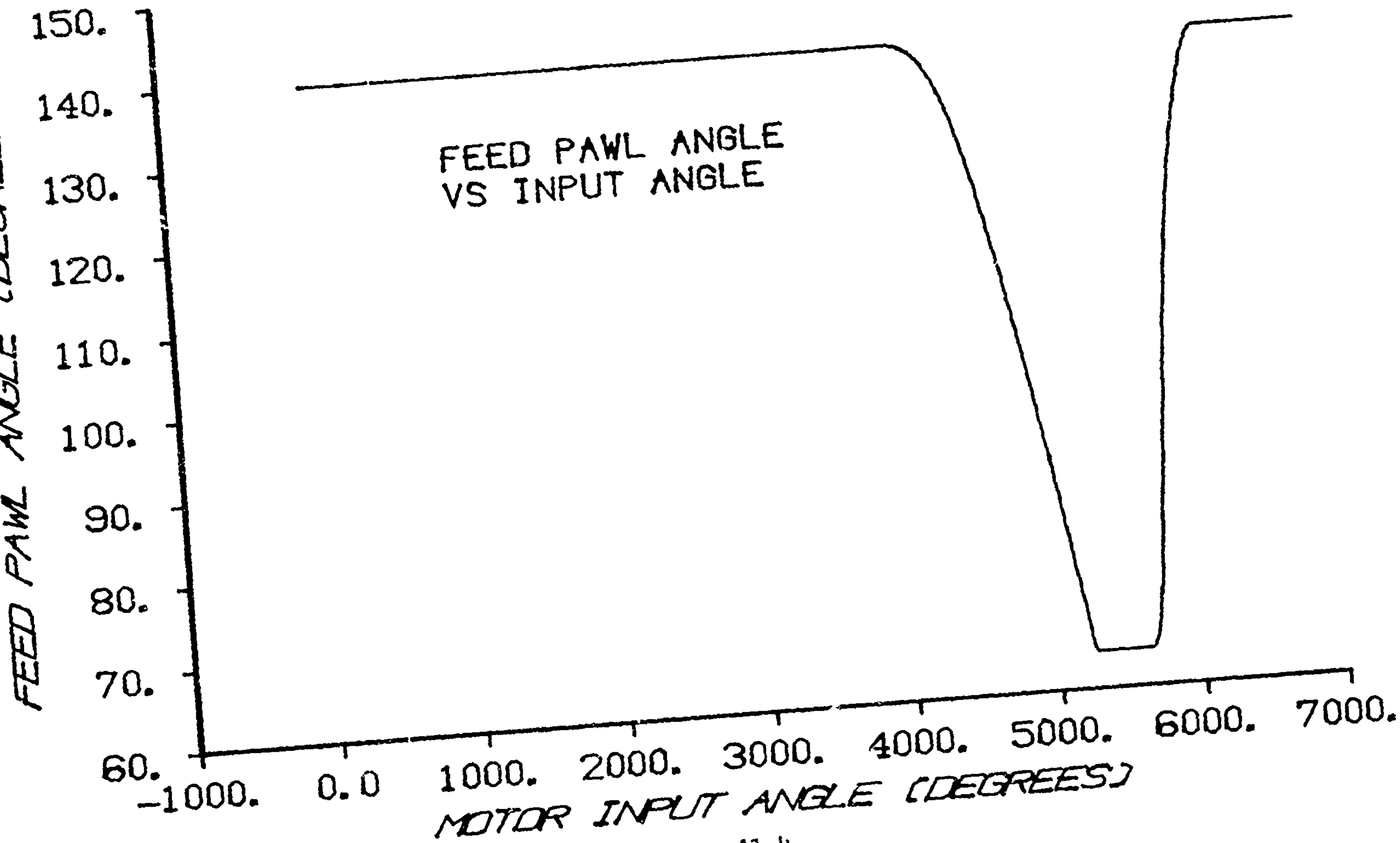


FIGURE A1-4

FEED PAWL ANGLE VS INPUT ANGLE

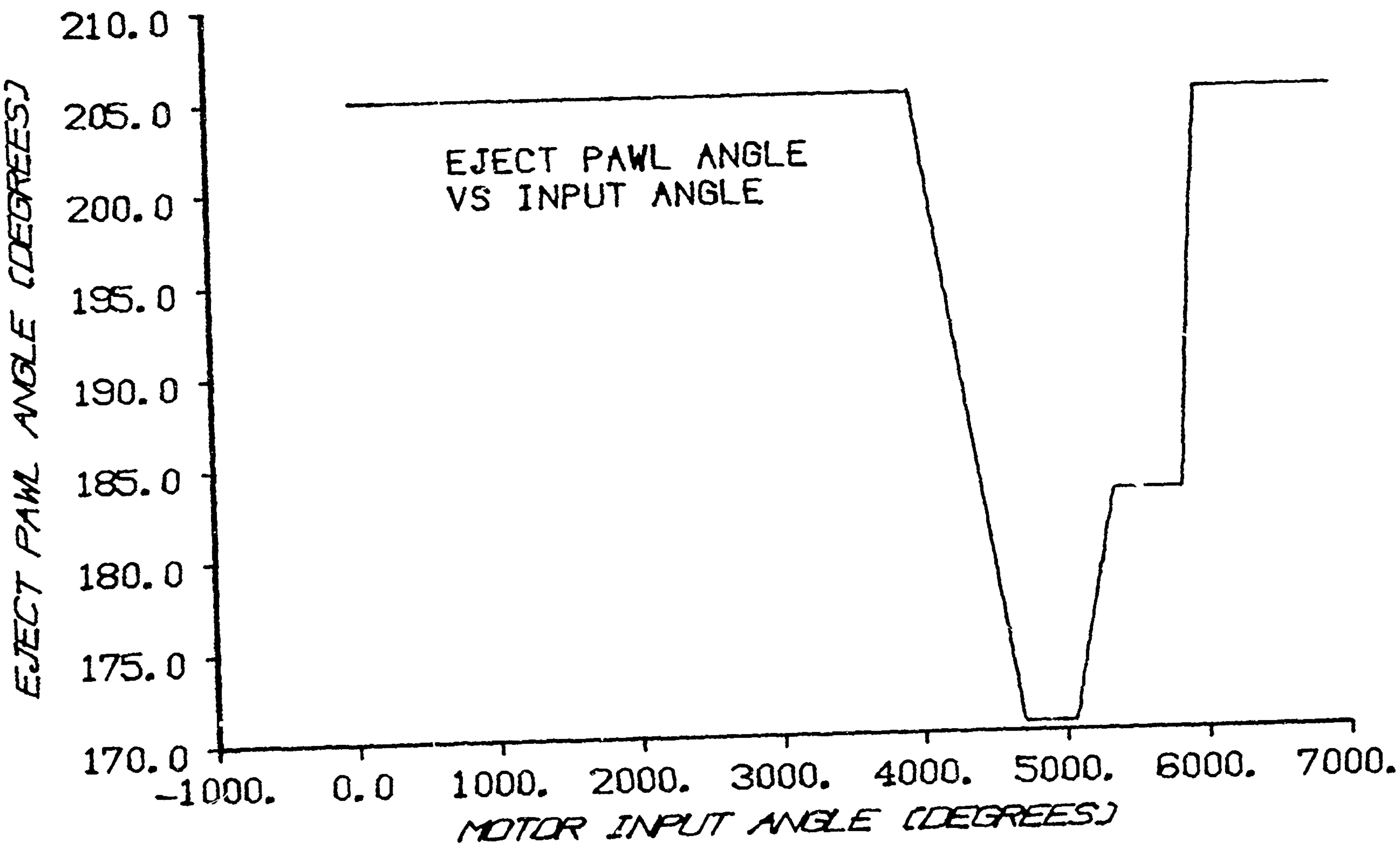


FIGURE A1-5

EJECT PAWL ANGLE VS INPUT ANGLE

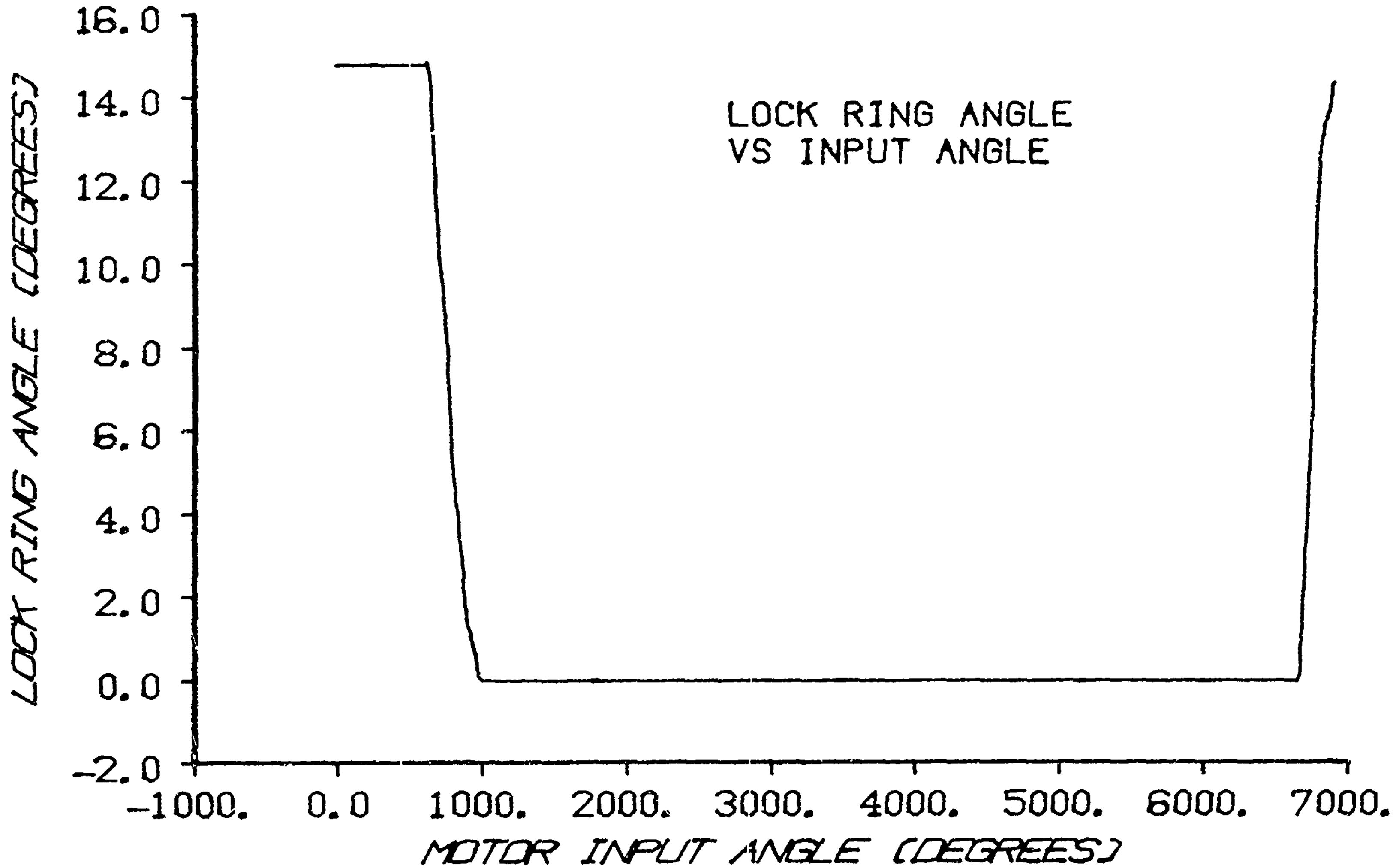


FIGURE A1-6

LOCK RING ANGLE VS INPUT ANGLE

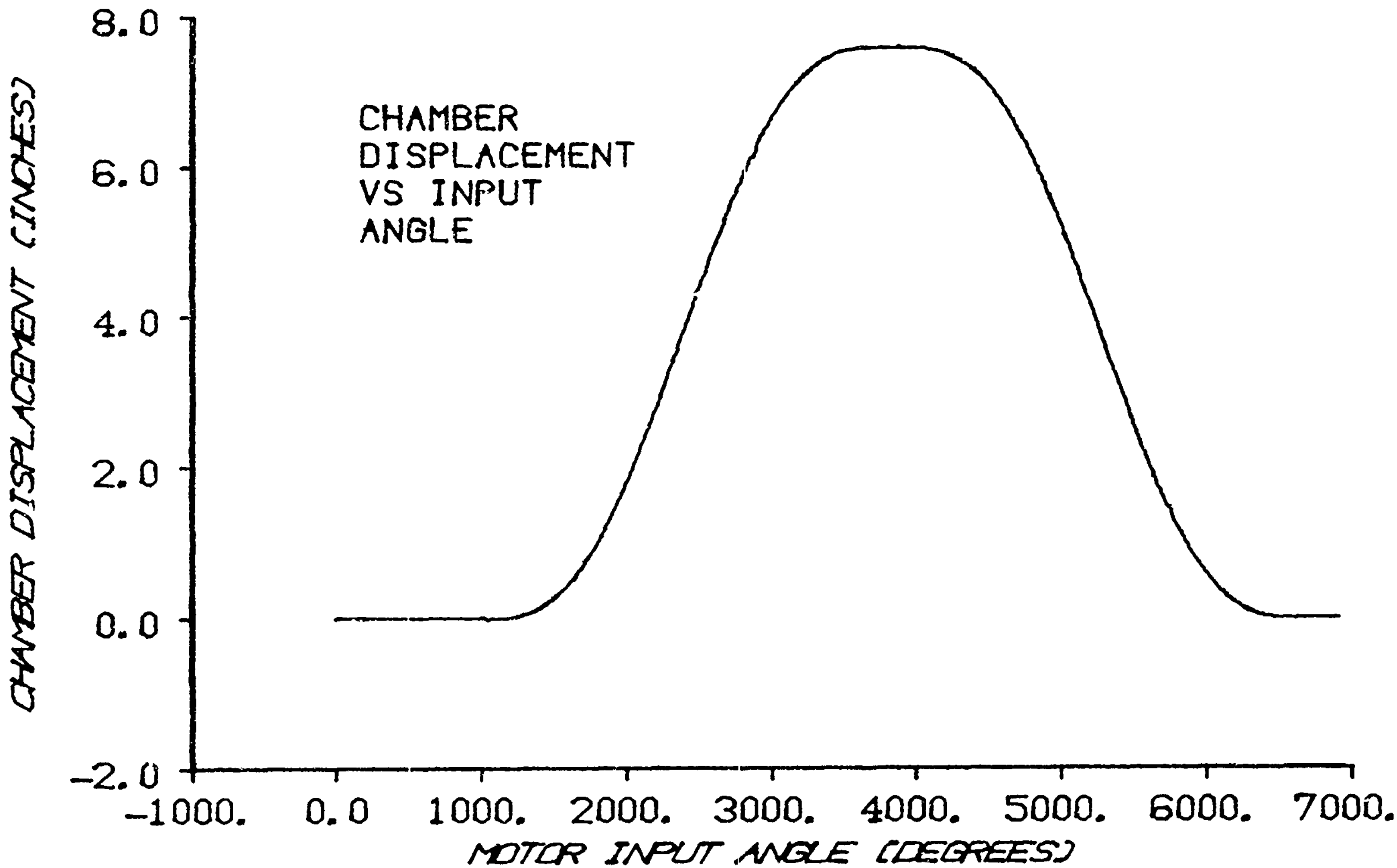


FIGURE A1-7

CHAMBER DISPLACEMENT VS INPUT ANGLE

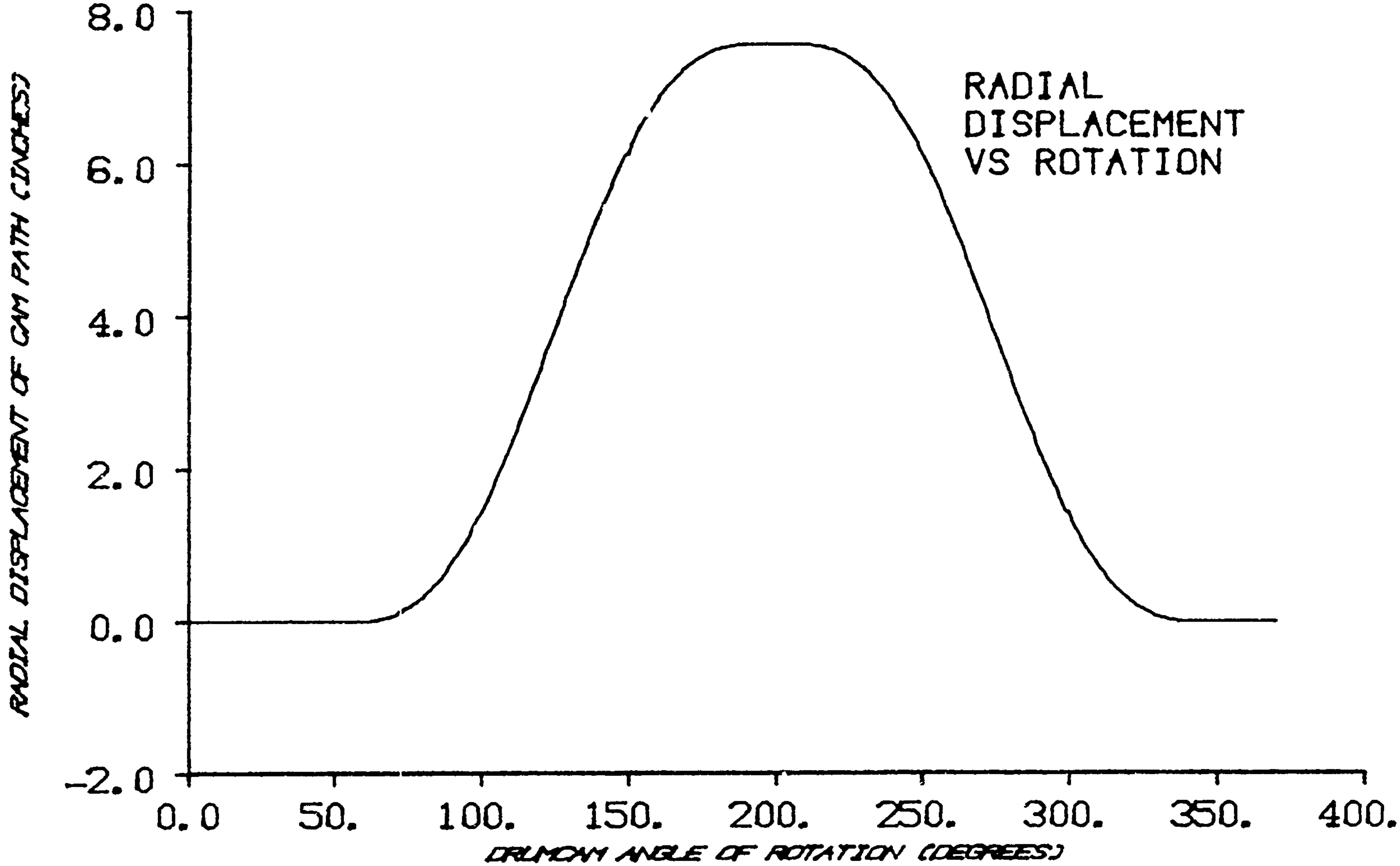


FIGURE A1-8

RADIAL DISPLACEMENT VS ROTATION

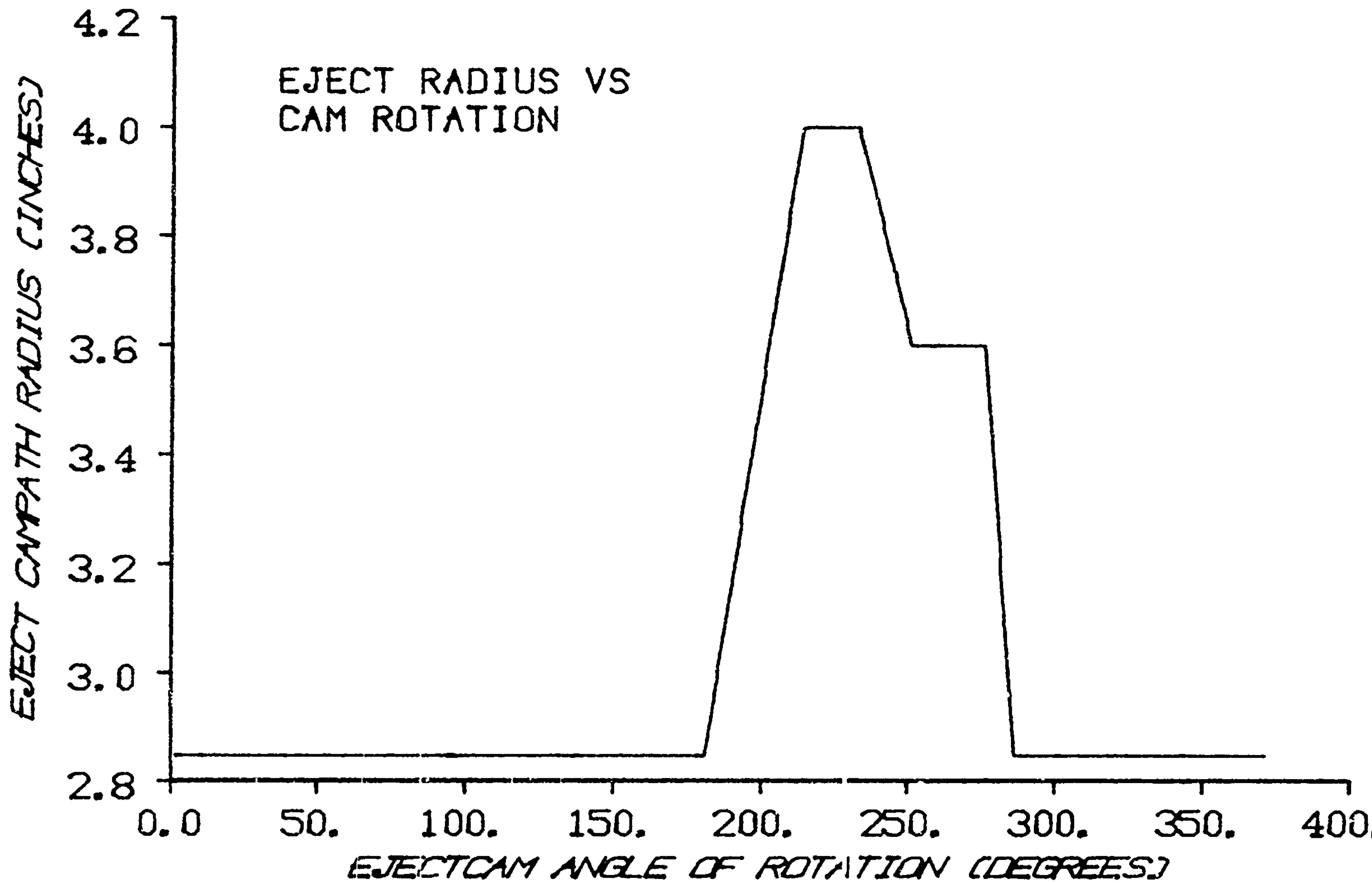


FIGURE A1-9

EJECT RADIUS VS CAM ROTATION

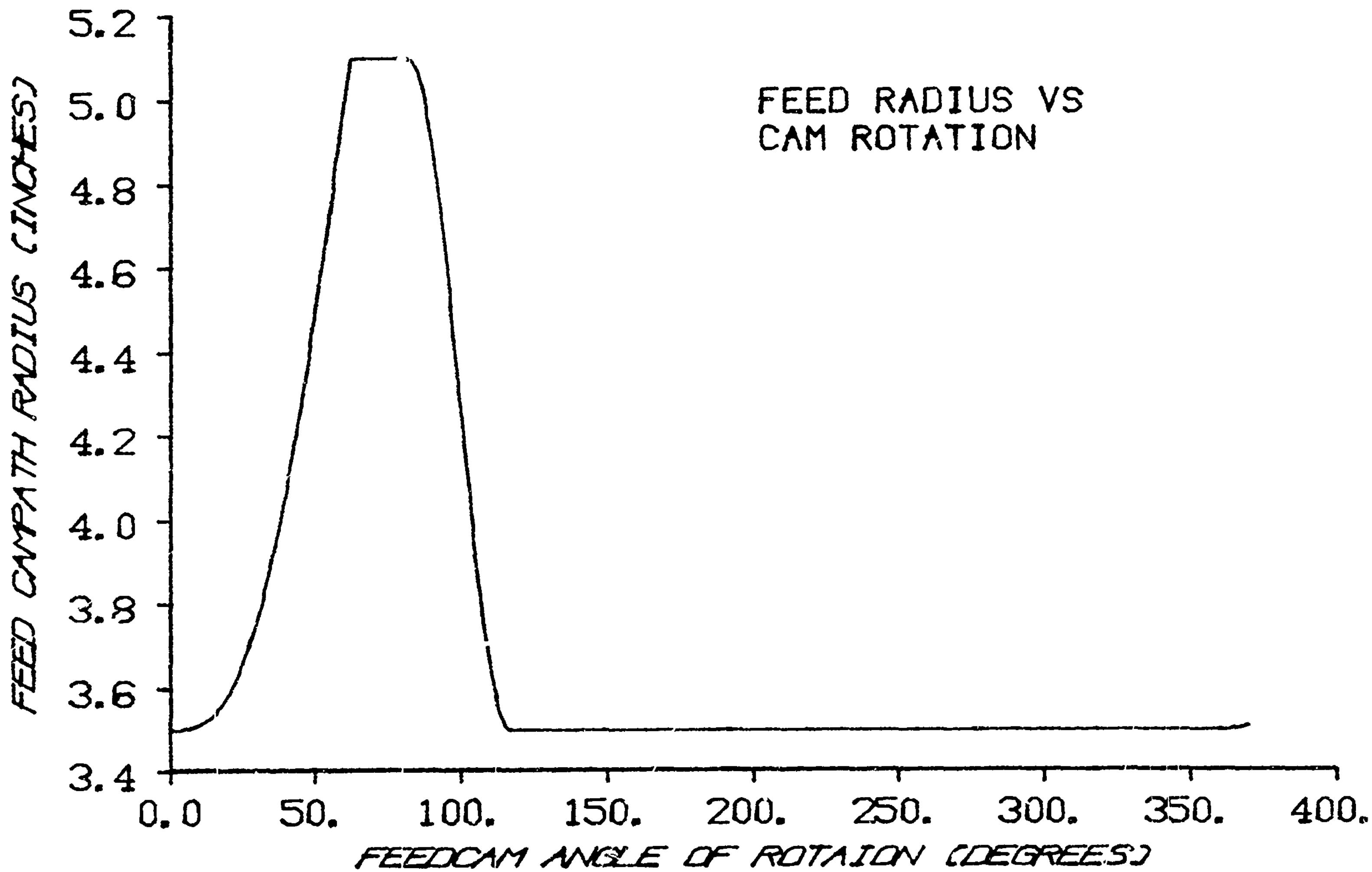


FIGURE A1-10

FEED RADIUS VS CAM ROTATION

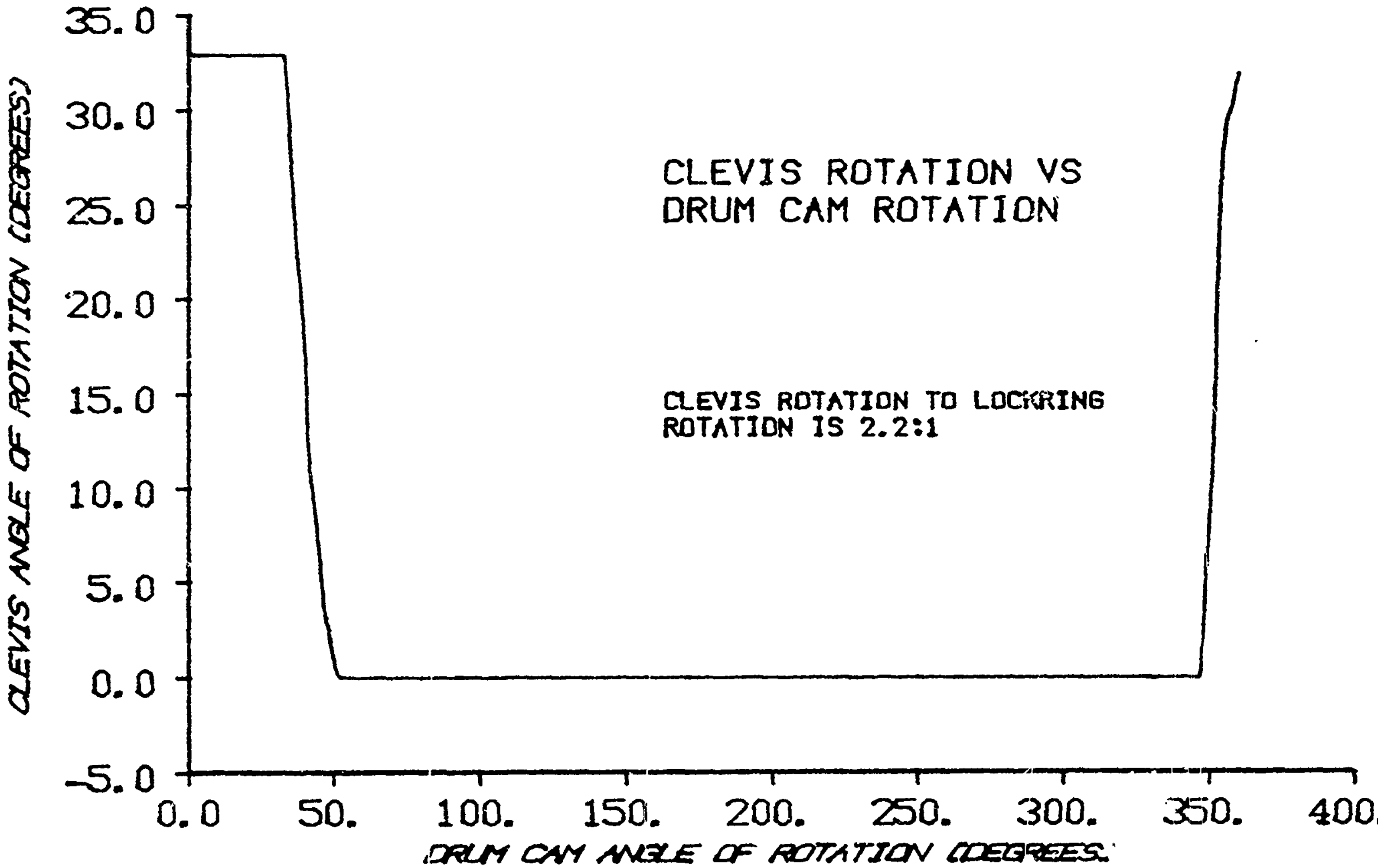
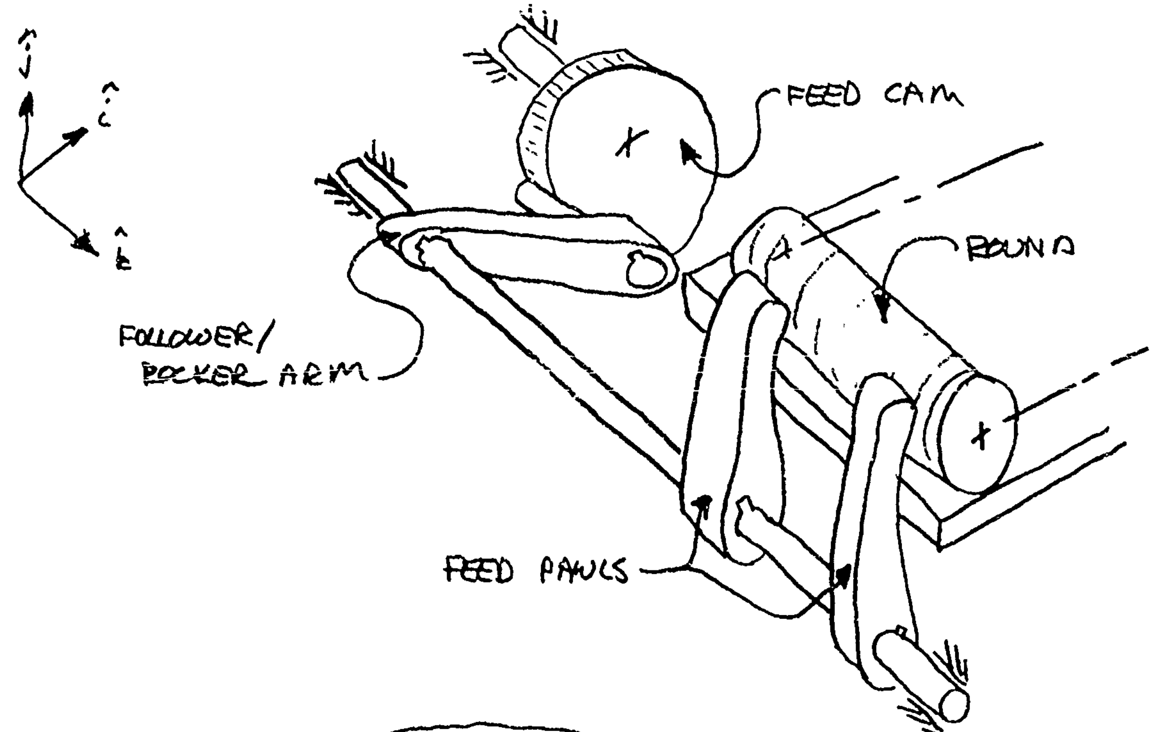


FIGURE A1-11

CLEVIS ROTATION VS DRUM CAM ROTATION



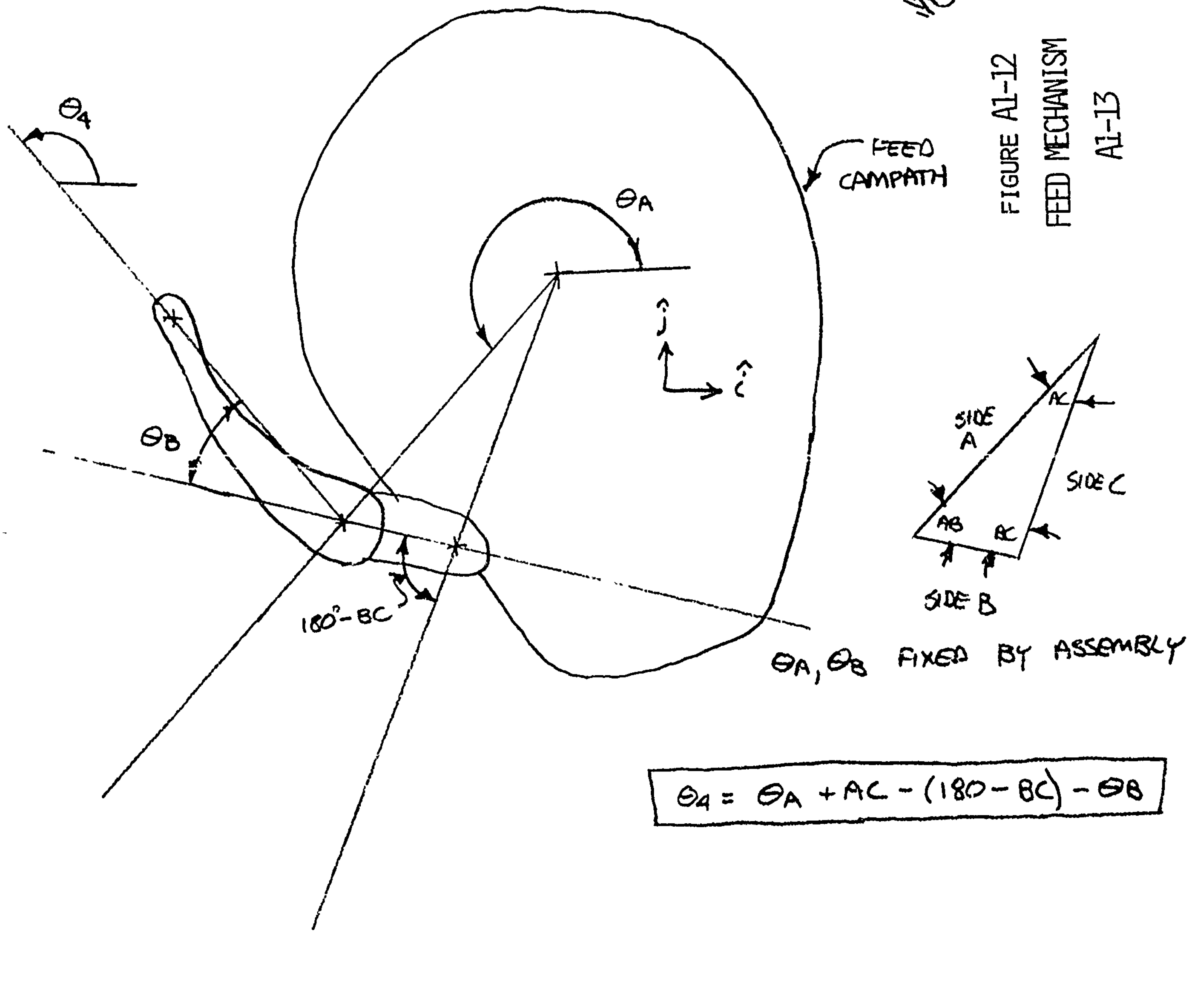
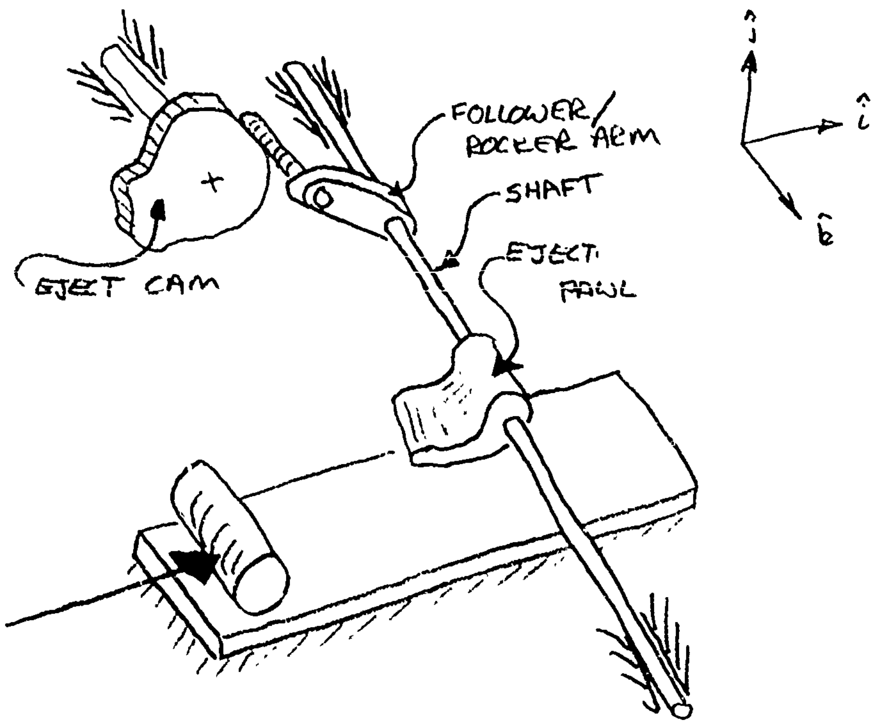


FIGURE A1-12
FEED MECHANISM



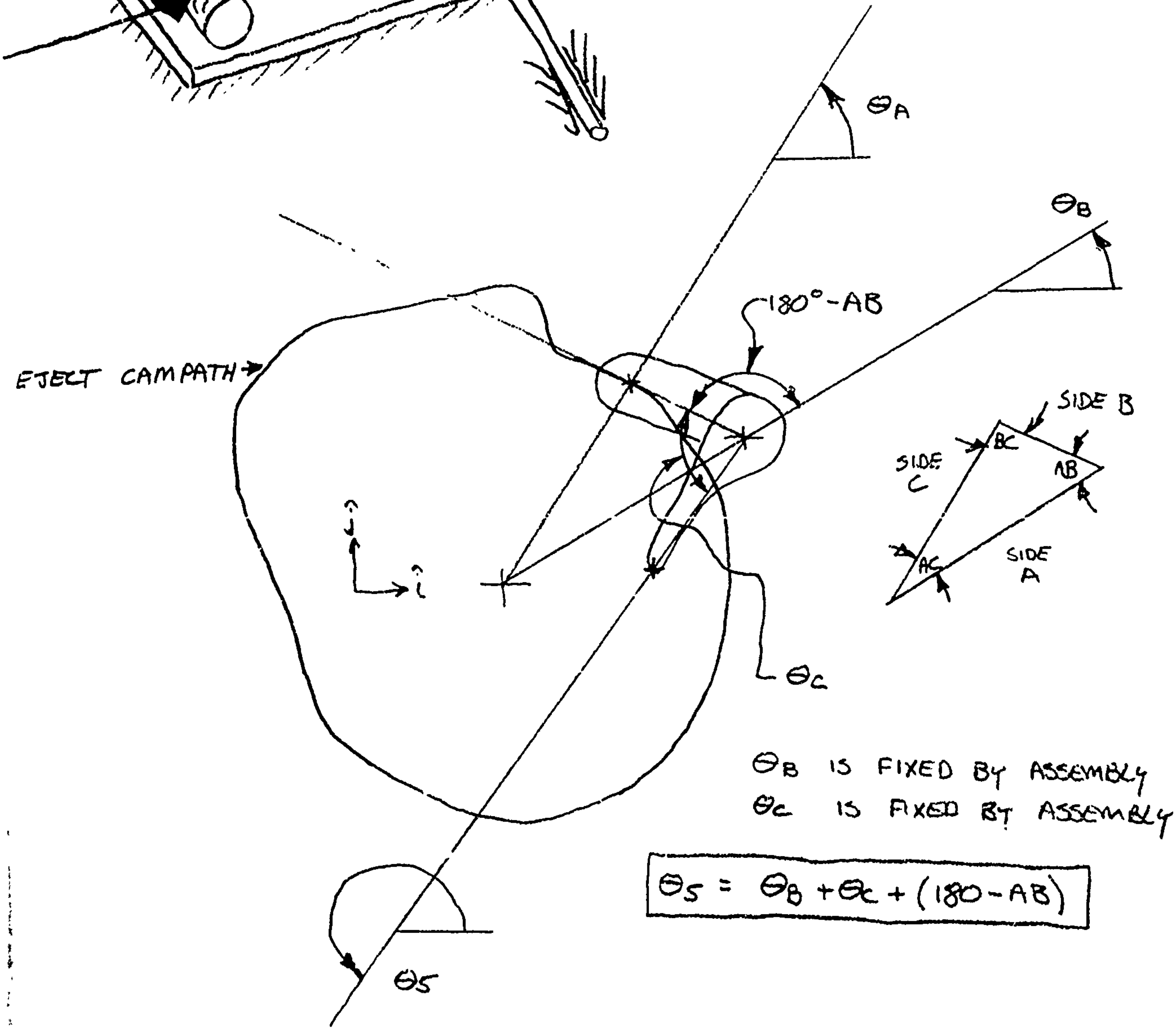


FIGURE A1-13
EJECT MECHANISM

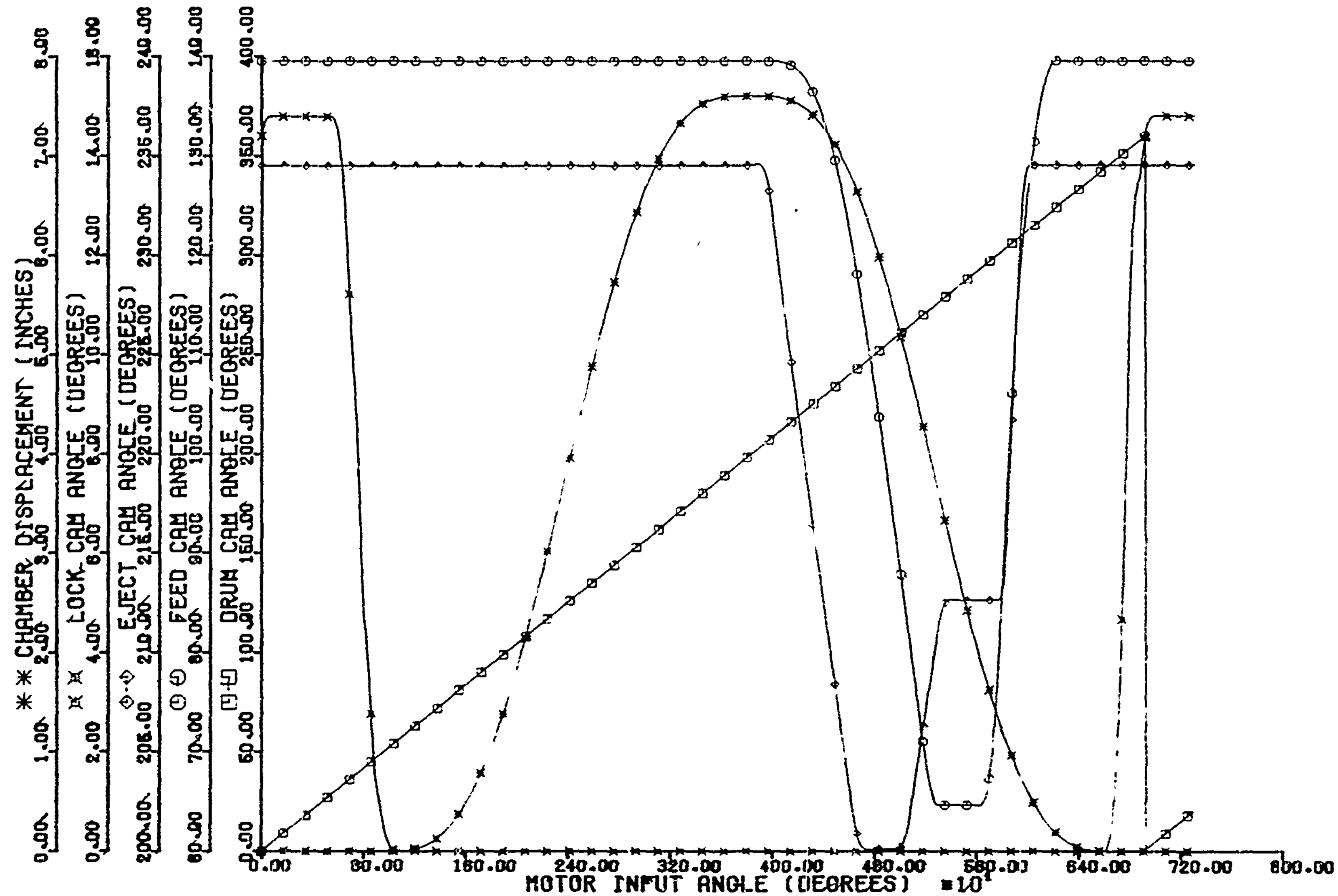


FIGURE A1-14

COMPLETE WEAPON TIMING

APPENDIX 2

Generalized d'Alembert Force
Procedure and AMCAWS-30
Equation of Motion Development

A2.1 - INTRODUCTION

The systematic derivation of the differential equaiton of motion for the AMCAWS-30 weapon was accomplished by the application of the generalized d'Alembert force equation

$$\sum_{i=1}^{M} \vec{F}_i \cdot \frac{\partial \vec{\rho}_i}{\partial q_j} + \sum_{\ell=1}^{L} \vec{\lambda}_\ell \cdot \frac{\partial P_\ell}{\partial q_j} = 0 \qquad \text{(A2.1)}$$

where,

M = number of d'Alembert forces in the total system

$\vec{F}$ = the d'Alembert force

$\vec{\rho}$ = the position vector from a point in ground to the point of application of the d'Alembert force

i = the index of the particular d'Alembert force being considered

q_j = the degree of freedom being considered

$\frac{\partial \vec{\rho}_i}{\partial q_j}$ = the partial of ρ_i with respect to the degree of freedom

$\vec{\lambda}$ = closed loop chord contact forces

$\vec{P}$ = positional vector spanning the closed loop chord

L = number of independent closed lopps

ℓ = loop being considered

j = degree of freedom

Two facts allow a significant simplification of an already simple equation. First, the AMCAWS-30 is a one degree of freedom system. This sets the index $j = 1$. Second, the development of functional relationships describing the response of the major components to the motor input angle, Θ_2, allowed the analysis to proceed with no closed loops (Appendix 1). This sets the index $L=0$. The generalized d'Alembert equation then becomes

$$\sum_{i=1}^{M} \vec{F}_i \cdot \frac{\partial \vec{\rho}_i}{\partial (\Theta_2)} = 0 \qquad \text{(A2.2)}$$

The problem at hand is then the cranking out of the $\vec{F}_i$'s and $\vec{\rho}_i$'s for the six major components involved. The detailed development follows. The black box idealization of Figure A2-4 is somewhat altered in that the reference frames for each component are oriented as they actually are on the weapon. Throughout, the development the $\hat{i},\hat{j},\hat{k}$ reference frame holds where $\hat{k}$ is aligned in the direction of projectile travel after a firing, the $\hat{i}$ axis is described by a horizontal line, and $\hat{j}$ corresponds to a vertical line (Figure A2-1). A single dot (i.e. $\dot{\Theta}_3$) refers to the derivative of Θ_3 with respect to Θ_2 (the degree of freedom). Similarly, double dots and primes (i.e. $\ddot{\Theta}_3$ and Θ_3'') are the second derivatives with respect to time and with respect to Θ_2. The interpolation routines scan the tabular data (coordinate of interest vs Θ_2) and returns a local 5th degree polynominal that fits the data in the region of interest. If Θ is the coordinate of interest, for any specific Θ_2,

$$\theta = a_0 + a_1\theta_2 + a_2\theta_2^2 + a_3\theta_2^3 + a_4\theta_2^4 + a_5\theta_2^5 \qquad \text{(A2.3)}$$

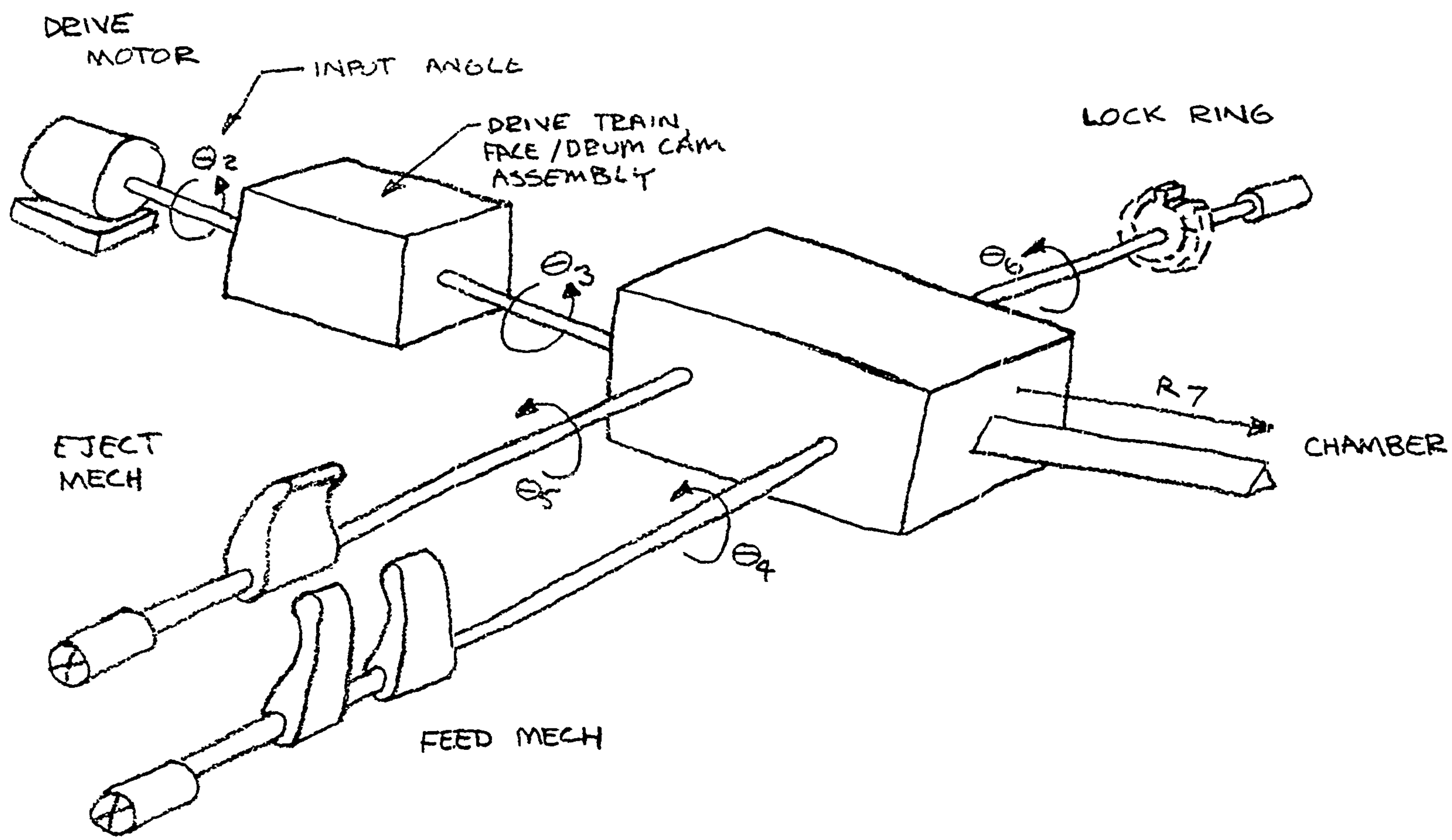
$$\theta' = \frac{\partial\theta}{\partial\theta_2} = a_1 + 2a_2\theta_2 + 3a_3\theta_2^2 + 4a_4\theta_2^3 + 5a_5\theta_2^4 \qquad \text{(A2.4)}$$

$$\theta'' = \frac{\partial^2\theta}{\partial\theta_2^2} = 2a_2 + 6a_3\theta_2 + 12a_4\theta_2^2 + 20a_5\theta_2^3 \qquad \text{(A2.5)}$$

$$\dot{\theta} = \theta'\dot{\theta}_2 \qquad \text{(A2.6)}$$

$$\ddot{\theta} = \theta'\ddot{\theta}_2 + \theta'' (\dot{\theta}_2)^2 \qquad \text{(A2.7)}$$

With these relationships the development of the differential equation of motion can proceed by applying the "reduced" d'Alembert equation to each major component and then summing all the terms into a single equation.



BLACK BOX IDEALIBATION OF AMCAWS·30 SYSTEM

FIGURE A2-1

AMCAWS-30 BLACK BOX

5/8/76

A-2 - LOCAL COORDINATE 2, DRIVE MOTOR INPUT

The gun parts associated with this coordinate are the input motor (409EH18, Western Gear Motor) and the gear unit transferring torque from the motor to the weapon itself. The parts and reference frame are shown in Figure A2-2.

TERMS:

I_{59} = Mass moment of inertia of the 59 tooth drive gear

I_{120} = Mass moment of inertia of the 120 tooth transfer gear (including the 16 tooth output gear shaft)

Θ_2 = Motor input angle measured as indicated

Θ_{120} = Angle of 120 tooth gear measured as indicated

T = Torque drive applied at the 59 tooth gear (a function of $\dot{\Theta}_2$)

$\vec{C}_f$ = Frictional losses through the drive train but considered to act on the 59 tooth gear.

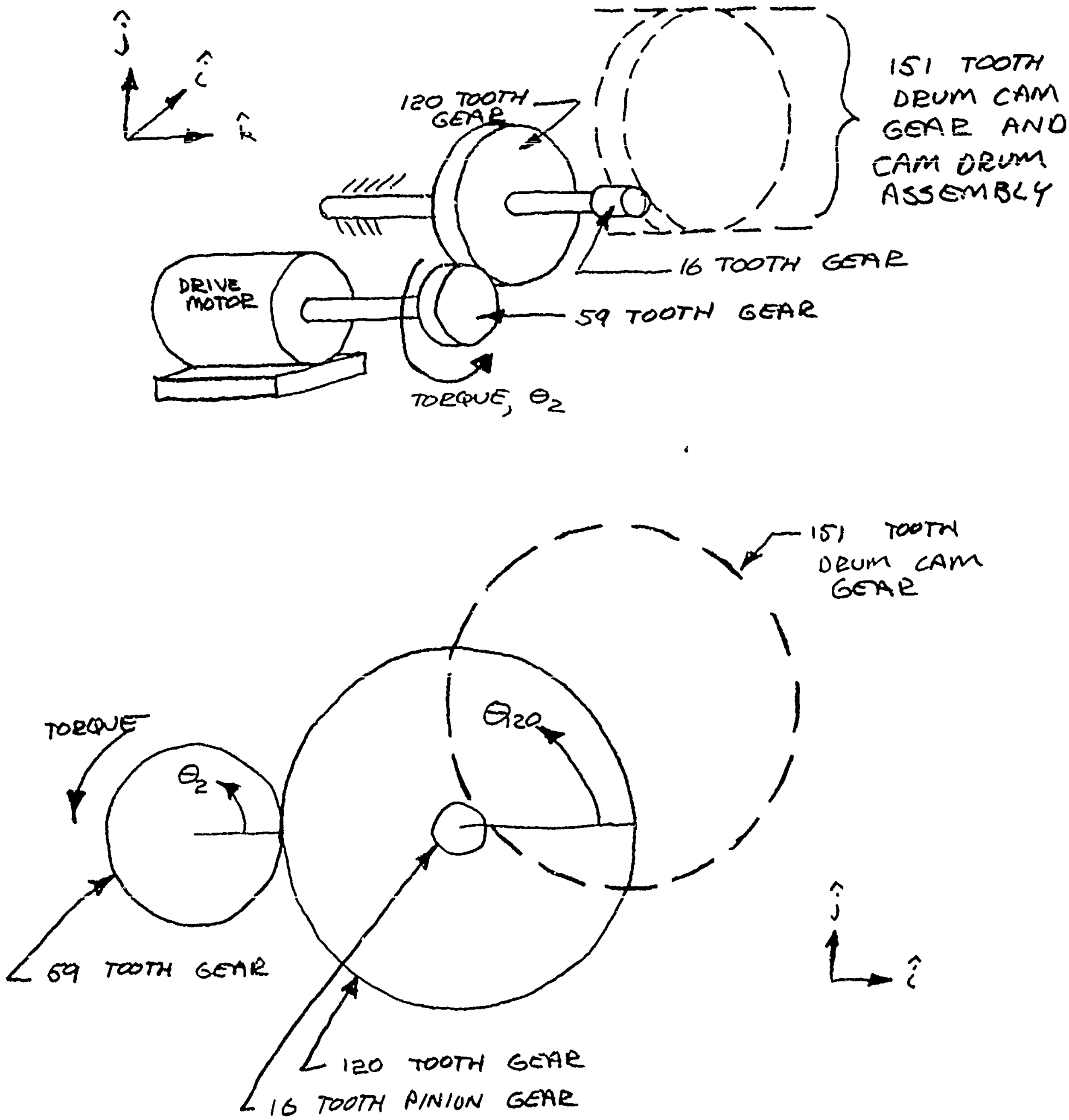


FIGURE A2-2

DRIVE COORDINATES

Relationships

$$\Theta_2 = \Theta_2$$

$$\Theta_{120} = (59/120)\Theta_2 = -C\Theta_2$$

$$\dot{\Theta}_{120} = -C\dot{\Theta}_2$$

$$\ddot{\Theta}_{120} = -C\ddot{\Theta}_2$$

$$\vec{C}_f = -C_f\hat{k}$$

Development:

i	Fi	$\vec{\rho}$	$\frac{\partial\vec{\rho}_i}{\partial\Theta_2}$	$\vec{F}i \cdot \frac{\partial\rho_i}{\partial\Theta_2}$
1	$-I_{59}\ddot{\Theta}_2\hat{k}$	$\Theta_2\hat{k}$	$\hat{k}$	$-I_{59}\ddot{\Theta}_2$
2	$-I_{120}\ddot{\Theta}_2\hat{k}$	$\Theta_{120}\hat{k}$	$-C\hat{k}$	$CI_{120}\ddot{\Theta}-C\ddot{\Theta}_2$
3	$T\hat{k}$	$\Theta_2\hat{k}$	$\hat{k}$	T
4	$-C_f\hat{k}$	$\Theta_2\hat{k}$	$\hat{k}$	$-C_f$

in terms of Θ_2,

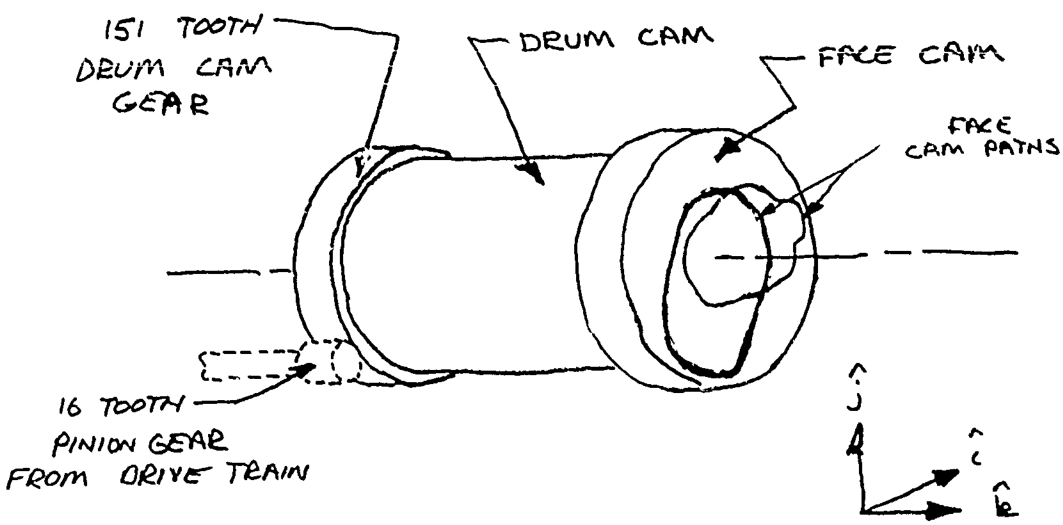
$$\sum_{i=1}^{4} \vec{F}_i \cdot \frac{\partial\vec{\rho}_i}{\partial\Theta_2} =$$

$$-I_{59}\ddot{\Theta}_2 - C^2 I_{120}\ddot{\Theta}_2 + T - C_f$$

$$= \ddot{\Theta}_2\,(-I_{59} - C^2 I_{120}) + T - C_f \qquad \text{(A2.8)}$$

A2.3 - LOCAL COORDINATE 3, DRUM/FACE CAMS

The gun parts associated with this coordinate are the drum cam and the face cam. Although in practice they are assembled so that they have no relative motion between them, they are treated separately in keeping with the practice of trying to do the analysis on very primary level so any program revisions necessated by part changes are minor. The parts and reference frame are shown in Figure A2-3.



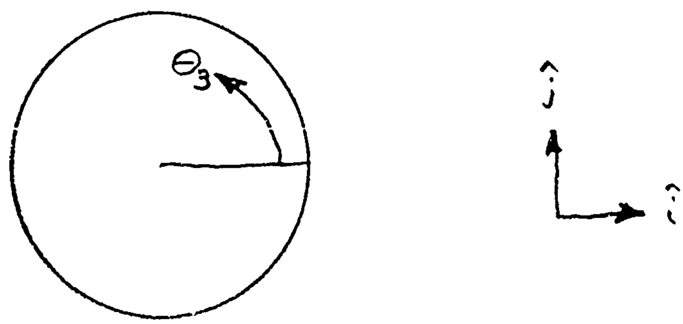


FIGURE A2-3

DRUM/FACE CAM COORDINATES

Terms:

IDRUM = mass moment of inertia of the drum cam

IFACE = mass moment of inertia of the face cam

Θ_3 = angle made by the zero point of the drum cam with the i axis, measured as shown.

Θ_F = angle made by the zero point of the face cam with the i axis, measured as shown.

Relationships:

$\theta_F = \theta_3$ + constant angle $= \theta_3 + A$

$\dot{\theta}_F = \dot{\theta}$

$\ddot{\theta}_F = \ddot{\theta}_3$

$\theta_3 = a_0 + a_1\theta_2 + a_2\theta_2^2 + a_3\theta_2^3 + a_4\theta_2^4 + a_5\theta_2^5$ for a given Θ_2

Development:

i	$\vec{F}i$	$\vec{\rho}i$	$\frac{\partial\vec{\rho}i}{\partial\theta_2}$	$\vec{F}i\,\frac{\partial\vec{\rho}i}{\partial\theta_2}$
5	$-IDRUM\ddot{\theta}_3\hat{k}$	$\theta_3\hat{k}$	$\theta_3'\hat{k}$	$-\theta_3' IDRUM\ddot{\theta}_3$
6	$-IFACE\ddot{\theta}_F\hat{k}$	$\theta_F\hat{k}$	$\theta_3'\hat{k}$	$-\theta_3' IFACE\ddot{\theta}_3$

in terms of θ_2

$$\sum_{j=5}^{6} \vec{F}i \cdot \frac{\partial\vec{\rho}i}{\partial\theta_2}$$

$$= -\theta_3'\ IDRUM(\theta_3'\ddot{\theta}_2 + \theta_3''\dot{\theta}_2^2)$$

$$-\theta_3'\ IFACE\ (\theta_3'\ddot{\theta}_2 + \theta_3''\dot{\theta}_2^2)$$

$$= \ddot{\theta}_2\ [-(\theta_3')^2(IDRUM + IFACE)]$$

$$-\theta_3'\theta_3''\dot{\theta}_2^2\ (IDRUM + IFACE) \quad (A2.9)$$

A2.4 - LOCAL COORDINATE 4, FEED MECHANISM

The gun parts associated with this coordinate are the feed pawls, the feed shaft, and the feed c follower which is called a rocker. The parts and reference frame are shown in Figure A2-4.

Terms:

IPAWL = mass moment of inertia of the pawl about the shaft

IROCK = mass moment of inertia of the rocker/follower about the shaft

ISHAFT = mass moment of inertia of the shaft

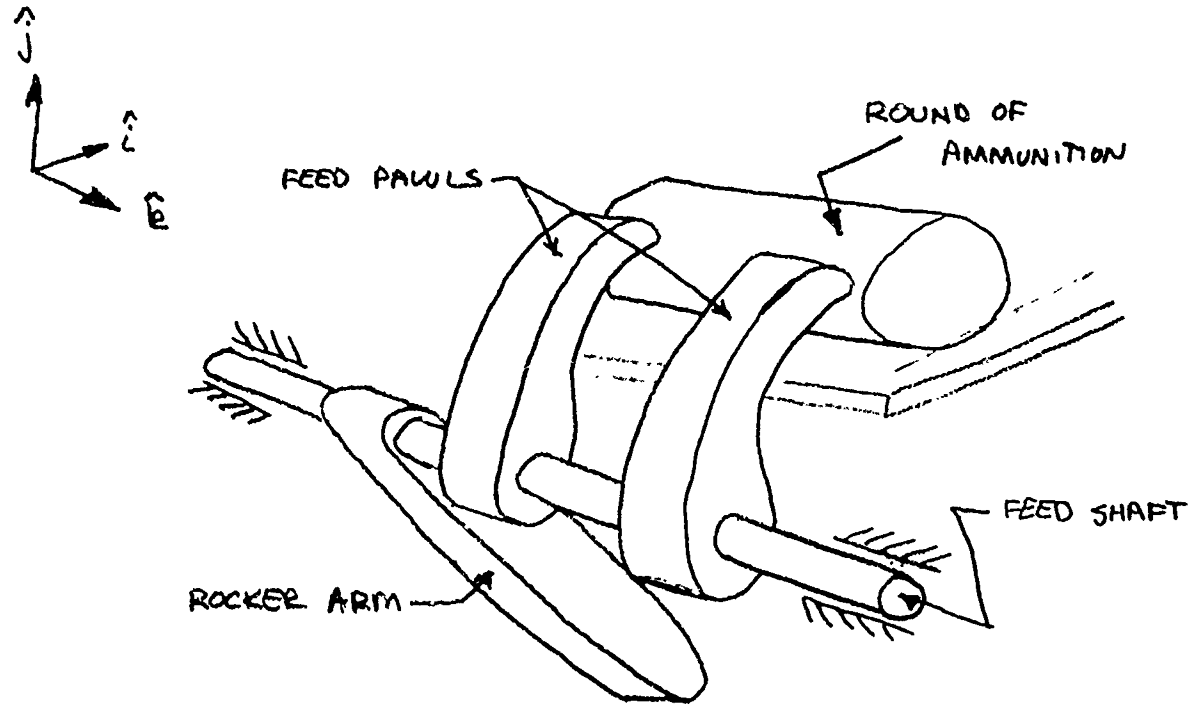
MPAWL = mass of the pawl

MROCK = mass of the rocker/follower

MAMMO = mass of the ammunition

Θ_2 = pawl angle measured as indicated

Θ_r = rocker angle measured as indicated



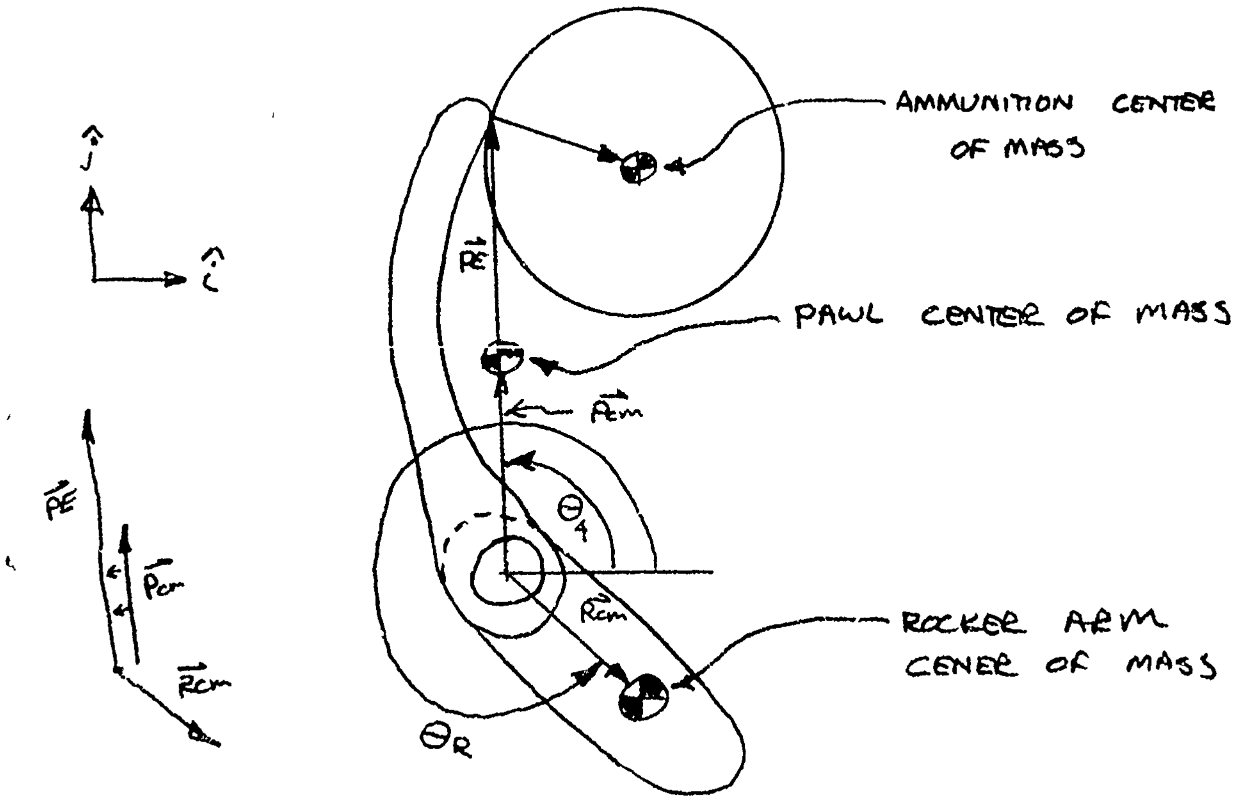


FIGURE A2-4

FEED COORDINATES

PCM = Distance from shaft to PAWL center of mass

RCM = Distance from shaft to rocker

PE = Distance from shaft to end of PAWL

$\vec{P}$ = positional vector from ground to pawl center of mass

$\vec{A}$ = positional vector from ground to ammo center of mass

$\vec{R}$ = positional vector from ground to rocker center of mass

T = torque needed to pull ammo belt.

Relationships:

$$\theta_r = \theta_4 + \text{constant angle} = \theta_4 + C$$

$$\dot{\theta}_r = \dot{\theta}_4$$

$$\ddot{\theta}_r = \ddot{\theta}_4$$

$$\vec{P} = \overline{\overrightarrow{const1}} + \overrightarrow{pcm}$$

$$\vec{A} = \overline{\overrightarrow{const2}} + \overrightarrow{PE} + (DAMMO/2)\hat{i}$$

$$\vec{R} = \overline{\overrightarrow{const3}} + \overrightarrow{RCM}$$

Development:

i	$\vec{F}i$	$\vec{R}i$	$\frac{\partial \rho i}{\partial \theta_2}$	$\vec{F}i \cdot \frac{\partial \rho i}{\partial \theta_2}$
7	$-IPAWL\ddot{\theta}_4\hat{k}$	$\theta_4\hat{k}$	$\theta_4'\hat{k}$	$-IPAWL\ddot{\theta}_4\theta_4'$
8	$-MPAWL(\ddot{\vec{p}}+g\hat{j})$	$\vec{p}$	$PCM\theta_4'(-\sin\theta_4\hat{i} + \cos\theta_4\hat{j})$	$-MPAWL\ (PCM)^2\theta_4'$ $\{\ddot{\theta}_4 + \cos\theta_4(g/PCM)\}$
9	$-IROCK\ddot{\theta}_r\hat{k}$	$\theta_r\hat{k}$	$\theta_4'\hat{k}$	$-IROCK\ddot{\theta}_4\theta_4'$
10	$-MROCK\ (\ddot{\vec{R}}+g\hat{j})$	$\vec{R}$	$RCM\theta_r'(-\sin\theta_r\hat{i}+\cos\theta_r\hat{j})$	$-MROCK\ (RCM)^2\theta_r'$ $\{\ddot{\theta}_r+\cos\theta_r(g/RCM)\}$
11	$-ISHAFT\ddot{\theta}_4\hat{k}$	$\theta_4\hat{k}$	$\theta_4'\hat{k}$	$-ISHAFT\ \ddot{\theta}_4\theta_4'$
12	$-MAMMO\ \ddot{\vec{A}}$	$\vec{A}$	$PE\theta_4'(-\sin\theta_4\hat{i}+\cos\theta_4\hat{j})$	$-MAMMO(PE)^2\theta_4'\ddot{\theta}_4$
13	$-T\hat{k}$	$\theta_4\hat{k}$	$\theta_4'\hat{k}$	$-T\theta_4'$

in terms of θ_2

$$\sum_{i=7}^{13} \vec{F}_i \cdot \frac{\partial \vec{\rho}_i}{\partial \Theta_2}$$

$$= \ddot{\Theta}_2(-(\Theta_4')^2\{IPAWL + IROCK + ISHAFT + MPAWL(PCM)^2 + MROCK(RCM)^2$$

$$+ MAMMO(PE)^2\}$$

$$- \dot{\Theta}_2^2(\Theta_4'\Theta_4'')\{IPAWL + IROCK + ISHAFT + MPAWL(PCM)^2 + MROCK(RCM)^2$$

$$+ MAMMO(PE)^2\}$$

$$-\Theta_4'\{MPAWL(PCM)g\cos\Theta_4 + MROCK(RCM)g\cos\Theta_r + T\} \quad (A2.10)$$

NOTE, however that the ammunition is only present during the portion of the cycle during which the ammo is transferred to the chamber for firing. Thus the term

MAMMO = (mass of AMMO) Unit $(\dot{\Theta}_4)$

where unit (Θ_4) = 1 when $\dot{\Theta}_4 < 0$, or since $\dot{\Theta}_2 >, 0$, when $\Theta_4' \pm 0$

= 0 when $\dot{\Theta}_4 \geq 0$, or since $\dot{\Theta}_2 >, 0$, when $\Theta_4' >, 0$

Also, there is currently no ammo belt system yet fabricated so that T is always zero.

A2.5 - LOCAL COORDINATE 5, EJECT MECHANISM

The gun parts associated with this coordinate are the eject pawl, the eject shaft, and the eject cam path fall over which is called a rocker. The parts and reference frame are shown in Figure A2-5.

Terms:

IPAWL = mass moment of inertia of the pawl about the shaft

IROCK = mass moment of inertia of the rocker about the shaft

ISHAFT = mass moment of inertia of the shaft

MPAWL = mass of the eject pawl

MROCK = mass of the rocker

Θ_5 = pawl angle measured as shown

Θ_r = rocker angle measured as shown

PCM = distance from shaft to pawl center of mass

RCM = distance from shaft to rocker center of mass

$\vec{P}$ = positional vector from ground to pawl center of mass

$\vec{R}$ = positional vector from ground to rocker center of mass

Relationships:

$$\Theta_r = \Theta_5 - \text{constant angle} = \Theta_5 - C$$

$$\Theta_r = \dot{\Theta}_5$$

$$\ddot{\Theta}_r = \ddot{\Theta}_5$$

$$\vec{P} = \vec{const} + \vec{PCM}$$

$$\vec{R} = \vec{const} + \vec{RCM}$$

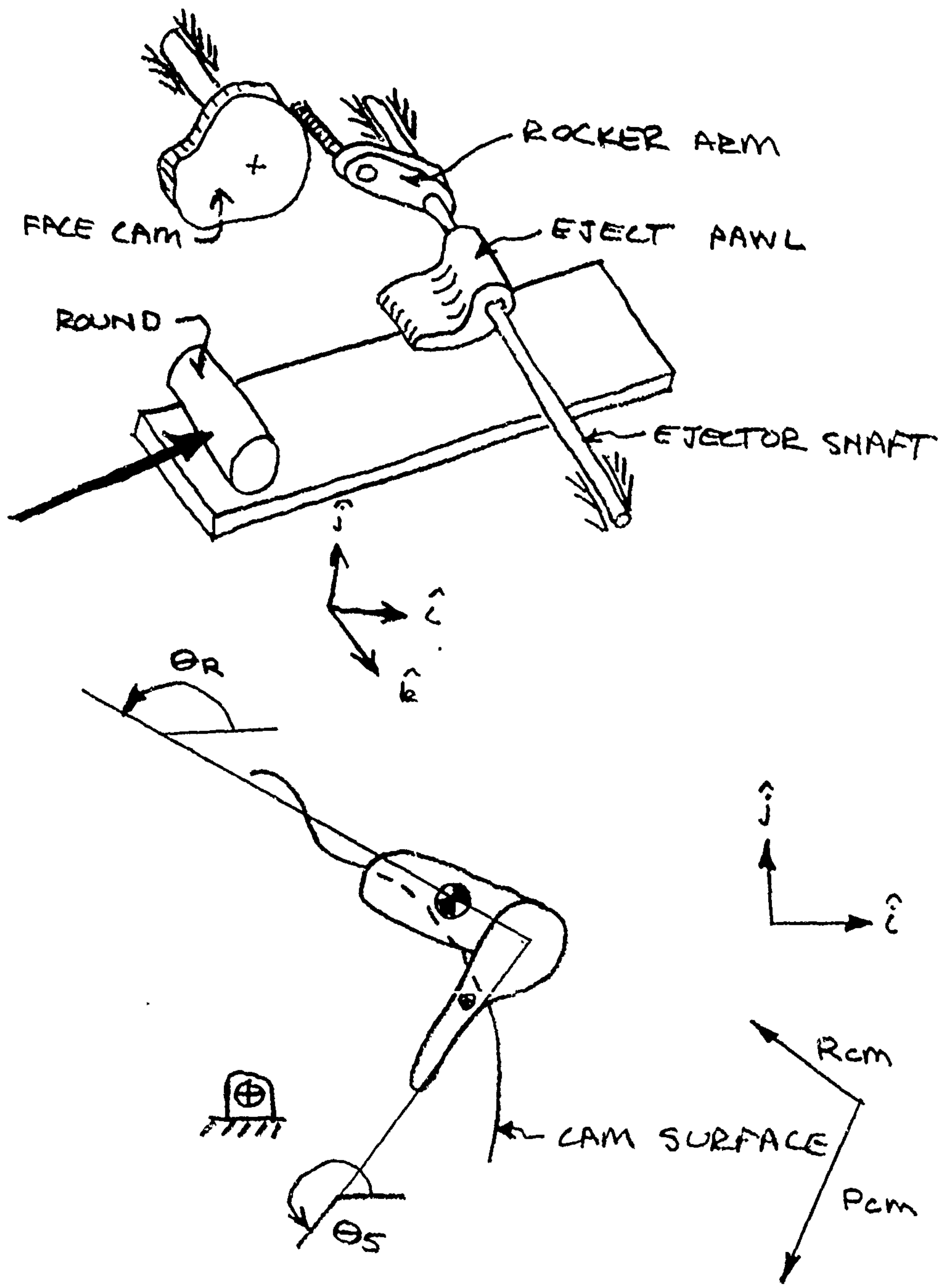


FIGURE A2-5

EJECT COORDINATES

Development:

i	$\vec{F}_i$	$\vec{P}_i$	$\frac{\partial \vec{P}_i}{\partial \Theta_2}$	$\vec{F}_i \cdot \frac{\partial \vec{\rho}_i}{\partial \Theta_2}$
14	$-IPAWL\ddot{\Theta}_5\hat{k}$	$\Theta_5\hat{k}$	$\Theta_5'\hat{k}$	$-IPAWL\ddot{\Theta}_5\Theta_4'$
15	$-MPAWL(\ddot{\vec{P}}+g\hat{j})$	$\vec{P}$	$Pcm\Theta_5'(-\sin\Theta_5\hat{i}+\cos\Theta_5\hat{j})$	
16	$-IROCK\ddot{\Theta}_5\hat{k}$	$\Theta_5\hat{k}$	$\Theta_5'\hat{k}$	$-IROCK\ddot{\Theta}_5\Theta_5'$
17	$-MROCK(\ddot{\vec{R}}+g\hat{j})$	$\vec{R}$	$Rcm\Theta_r(-\sin\Theta_r\hat{i}+\cos\Theta_r\hat{j})$	
18	$-ISHAFT\ddot{\Theta}_5\hat{k}$	$\Theta_5\hat{k}$	$\Theta_5'\hat{k}$	$-ISHAFT\Theta_5\Theta_5'$

$$-MPAWL(Pcm)^2\Theta_5'\{\ddot{\Theta}_5+\cos\Theta_5(g/Pcm)\}$$

$$-MROCK(R_{cm})^2\Theta'\{\ddot{\Theta}_r+\cos\Theta_r(g/Rcm)\}$$

in terms of $\dot{\Theta}_2$,

$$\sum_{14}^{18} \vec{F}_i \cdot \frac{\partial \vec{\rho}_i}{\partial \Theta_2}$$

$$= -(\Theta_5')^2\{IPAWL + IROCK + ISHAFT + MPAWL(PCM)^2 + MROCK(RCM)^2\}\ddot{\Theta}_2$$

$$-(\Theta_5'\Theta_5'')\{IPAWL + IROCK + ISHAFT + MPAWL(PCM)^2 + MROCK(RCM)^2\}\dot{\Theta}_2^2$$

$$-(\Theta_5')\{MPAWL(PCM)g\cos\Theta_5 + MROCK(RCM)g\cos\Theta_r\} \quad \text{(A2.11)}$$

A2.6 - LOCAL COORDINATE 6, LOCK RING

The gun parts associated with this coordinate is the lock ring. The part and reference frame is shown in Figure A2-6.

Terms:

ILOCK = mass moment of inertia of lock ring

T = additional torques acting on the lock ring, including friction

Θ_6 = angle from lock ring center to zero point of ring from $\hat{i}$ axis

Relationships:

$$\Theta_6 = a_0 + a_1\Theta_2 + a_2\Theta_2^2 + a_3\Theta_2^3 + a_4\Theta_2^4 + a_5\Theta_2^5 \text{ for given } \Theta_2$$

Development:

i	$\vec{F}_i$	$\vec{\rho}_i$	$\frac{\partial\vec{\rho}_i}{\partial\Theta_2}$	$\vec{F}_i \cdot \frac{\partial\vec{\rho}_i}{\partial\Theta_2}$
19	$-LOCK\ddot{\Theta}_6\hat{k}$	$\Theta_6\hat{k}$	$\Theta_6'\hat{k}$	$-ILOCK\ddot{\Theta}_6\Theta_6'$
20	$-T\hat{k}$	$\Theta_6\hat{k}$	$\Theta_6'\hat{k}$	$-T\Theta_6'$

in terms of Θ_2

$$\sum_{i=19}^{20} \vec{F}_i \cdot \frac{\partial\vec{\rho}_i}{\partial\Theta_2}$$

$$= \ddot{\Theta}_2(-ILOCK(\Theta_6')^2) + \dot{\Theta}_2^2(-ILOCK\ \Theta_6'\Theta_6'') - T\Theta_6' \qquad \text{(A2.12)}$$

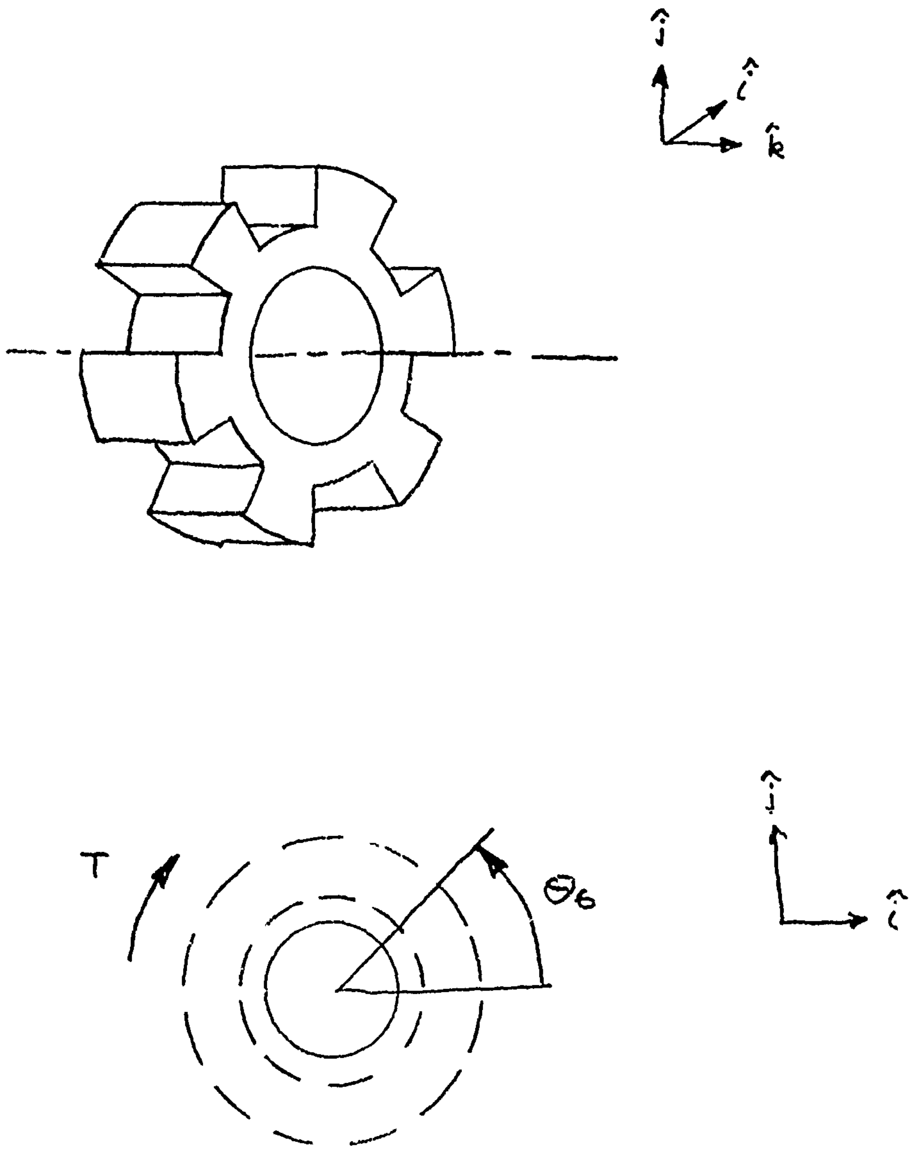


FIGURE A2-6

LOCK RING COORDINATES

A2.7 - LOCAL COORDINATE 7, CHAMBER

The gun parts associated with this coordinate are the chamber and bolt assemblies. As discussed in Section 2.5, the bolt and chamber translate as a unit only a small distance before the bolt becomes stationary. The actual mass that is translating is the mass of interest. The chamber assembly includes everything shown in Figure 9. The parts and reference frame is shown in Figure A2-7.

Terms:

$\vec{S}$ = vector from ground to fixed point on chamber

Crush = spring force acting on chamber due to round crush up

Sear = spring force acting on chamber during researing of firing pen spring

$\vec{C}_f$ = friction acting to resist chamber motion

VCHMBR = virtual (actual) translating mass

Relationships:

$$R = a_0 + a_1\Theta_2 + a_2\Theta_2^2 + a_3\Theta_2^3 + a_4\Theta_2^4 + a_5\Theta_5^5$$

$$\vec{R} = -R\hat{k}$$

$$\vec{S} = \text{const} + \vec{R}$$

$$\vec{CRUSH} = -(\text{CRUSH})\,\hat{k}$$

$$\vec{SEAR} = (\text{SEAR})\,\hat{k}$$

$$\vec{C}_f = C_f\hat{k} \text{ when } \dot{R} > 0$$

$$-C_f\hat{k} \text{ when } \dot{R} \leq 0$$

$$\ddot{\vec{S}} + \ddot{\vec{R}} = \ddot{\vec{R}}$$

$$\frac{\partial(\vec{S}+\vec{R}_1)}{\partial\Theta_2} \qquad \frac{\partial R}{\partial\Theta_2} = R^1$$

i	$\vec{F}_i$	$\vec{\rho}_i$	$\frac{\partial \rho i}{\partial \Theta_2}$	$\vec{F}i \cdot \frac{\partial \vec{\rho} i}{\partial \Theta_2}$
21	$-VCHMBR(\ddot{\vec{S}}+\ddot{\vec{R}}_1)$	$\vec{S}+\vec{R}$	$-R'\hat{k}$	$(VCHMBR)R'\ddot{R}$
22	$\vec{CRUSH}$	$\vec{R}$	$-R'\hat{k}$	$(CRUSH)R'$
23	$\vec{SEAR}$	$\vec{R}$	$-R'\hat{k}$	$-(SEAR)R'$
24	$\vec{C}_f$	$\vec{R}$	$-k'\hat{k}$	$+C_f R'(\frac{\dot{R}}{\lvert\dot{R}\rvert})$

in terms of Θ_2

$$\sum_{C=21}^{24} \vec{F}i \frac{\partial \vec{\rho}_i}{\partial \Theta_2}$$

$$= VCHMBR\ R'(R'\ddot{\Theta}_2 + R''\dot{\Theta}_2^2) + (CRUSH)R' - (SEAR)R' + C_f R'(R'\dot{\Theta}_2/ABS(R'\dot{\Theta}_2))$$

$$= \ddot{\Theta}_2(VCHMBR(R')^2) + \dot{\Theta}_2^2(VCHMBR\ R'R'') + R'\ (CRUSH - SEAR + C_f(R'\dot{\Theta}_2/ABS(R'\dot{\Theta}_2)) \quad \text{(As.13)}$$

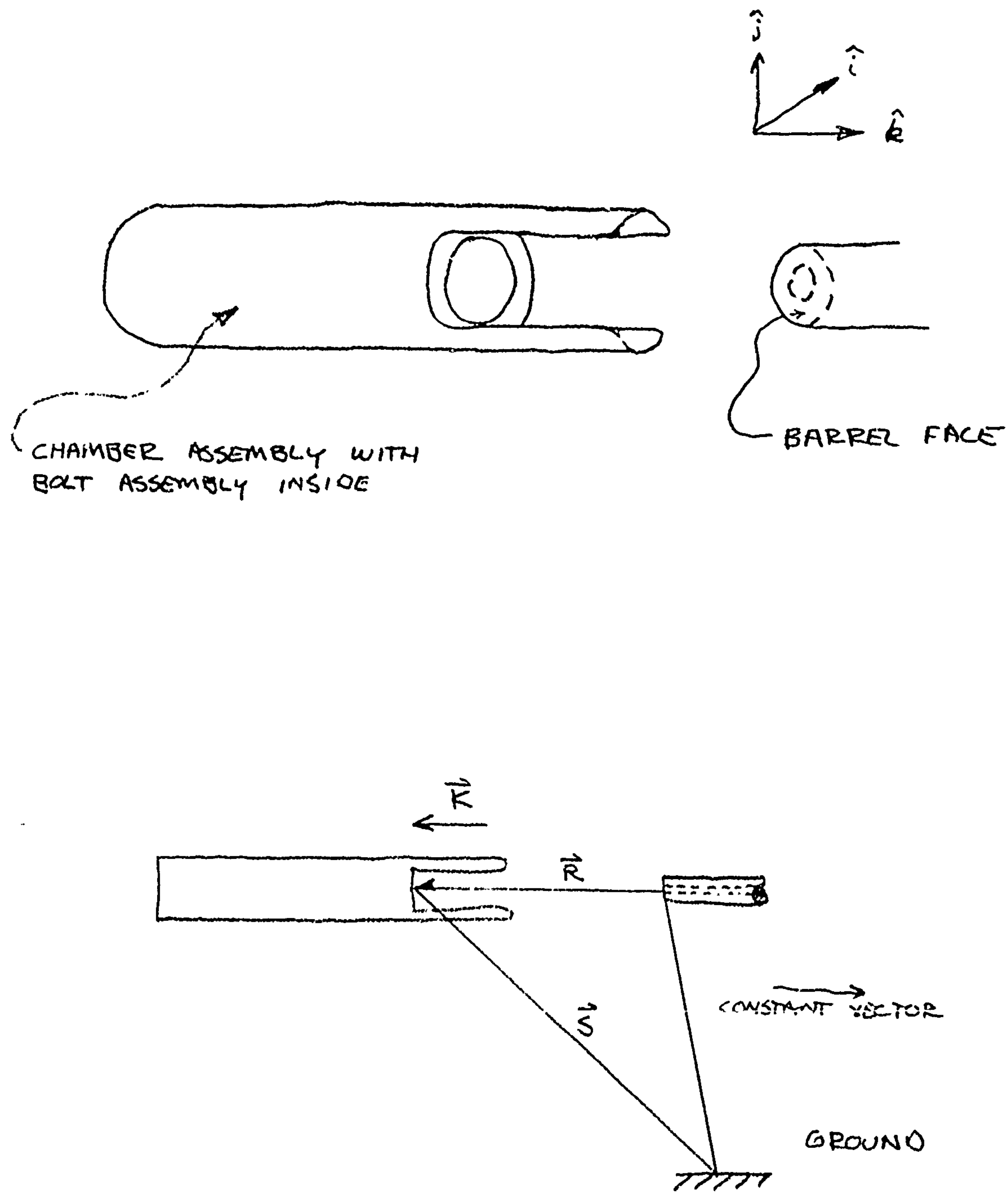


FIGURE A2-7

CHAMBER COORDINATES

A2.8 - GENERAL DEVELOPMENT OF A TRANSLATING d'ALEMBERT FORCE

Let the positional vector be PCM and have it acting at angle as shown in the Figure A-8 as always, Θ is a function of Θ_2

Terms:

$\vec{P}$ = positional vector from ground to the center of mass

Θ = angle measured as shown

PCM = magnitude of distance from origin to center of mass

g = acceleration due gravity

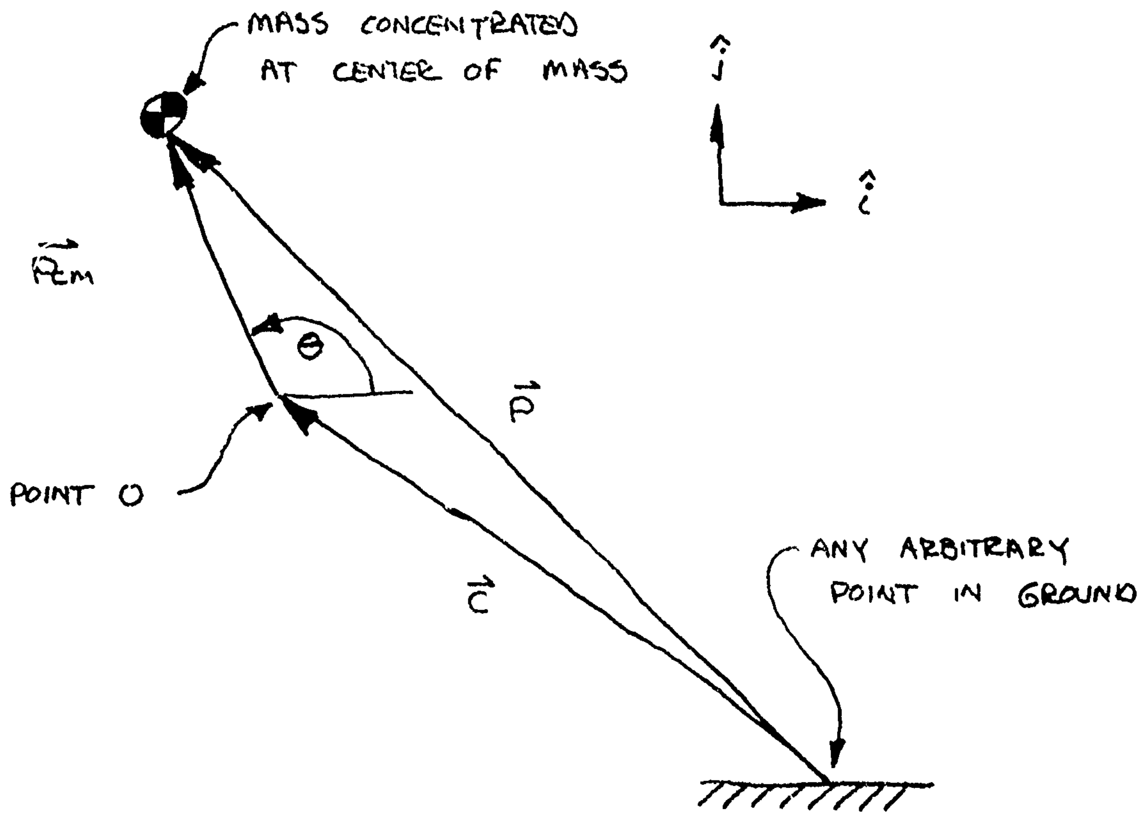
m = mass of part, considered concentrated at center of mass

Relationahips:

$$\Theta = a + a_1\Theta_2 + a_2\Theta_2^2 + a_3\Theta_2^3 + a_4\Theta_2^4 + a_5\Theta_2^5 \qquad \text{(A2-3)}$$

Development:

The only force considered here is the d'Alembert force associated with translation of the center of mass.



NOTES:

$\vec{P}$ IS VECTOR FROM A GROUND POINT TO CM

$\vec{C}$ IS VECTOR FROM THE GROUND POINT TO THE AXIS OF ROTATION FOR THE PART, POINT O

$\vec{R}_{cm}$ IS THE VECTOR FROM O TO THE CM AND THUS HAS FIXED LENGTH

GRAVITY ACTS IN $-\hat{j}$ DIRECTION WITH MAGNITUDE g

FIGURE A2-8

GENERAL TRANSLATING MASS

The positional vector $\vec{P}$ is

$\vec{P} = \vec{C} + \vec{P}_{cm}$, where $\vec{C}$ is a vector from some point in ground to point 0, which is fixed relative to ground.

The second time derivative of the vector $\vec{P}$(in two dimensions) is

$$\ddot{\vec{P}} = 0 + \ddot{\vec{P}}CM$$

$$= (\ddot{P}_{cm})(\hat{P}_{cm}) + \ddot{\Theta}(\hat{k}X\vec{P}_{cm}) - \dot{\Theta}^2\vec{P}_{cm} + 2\dot{\Theta}(\hat{k}X(\dot{P}CM)\hat{P}CM) \qquad (A2.14)$$

however, the length P_{cm} = constant

thus $\dot{P}_{cm} = 0$

and $\ddot{P}_{cm} = 0$, yielding

$$\ddot{\vec{P}} = \ddot{\Theta}(\hat{k}X\vec{P}_{cm}) - (\dot{\Theta})^2 \vec{P}_{cm} \qquad (A2.14.a)$$

since

$$\vec{P}_{cm} = P_{cm}(COS\Theta\hat{i} + SIN\Theta\hat{j})$$

$$(\hat{k}X\vec{P}CM) = P_{cm}(-SIN\Theta\hat{i} + COS\Theta\hat{j}), \text{ therefore}$$

$$\ddot{\vec{P}} = \ddot{\Theta}P_{cm}(-SIN\Theta\hat{i} + COS\Theta\hat{j}) - \dot{\Theta}^2P_{cm}(COS\Theta\hat{i} + SIN\Theta\hat{j})$$

$$= \{(-\sin\Theta)(\ddot{\Theta})(PCM) - (\dot{\Theta}^2)(P_{cm})(\cos\Theta)\}\hat{i}$$

$$+\{(\cos\Theta)(\ddot{\Theta})(PCM) - (\dot{\Theta}^2)(P_{cm})(\sin\Theta)\}\hat{j} \qquad (A2.14.b)$$

$\vec{F}$ for this case is

$$\vec{F} = -M[\ddot{\vec{P}} + g\hat{j}] =$$

$$-MP_{cm} \quad \{(-\sin\Theta)\ddot{\Theta} - \dot{\Theta}^2\cos\Theta\}\hat{i}$$

$$+ \{(\cos\Theta)\ddot{\Theta} - \dot{\Theta}^2\sin G + \frac{3}{P_{cm}}\}\hat{j} \qquad (A2.15)$$

$\vec{P} = \vec{C} + P_{cm}^{\rightarrow}$, AND

$$\frac{\partial \vec{\rho}}{j\Theta_2} = 0 + \frac{\partial(P_{cm}^{\rightarrow})}{\partial\Theta_2}$$

$$\frac{j}{j\Theta_2} \quad \{P_{cm}(\cos\Theta\hat{i} + \sin\Theta\hat{j})\}$$

$$= P_{cm}\frac{d\Theta}{d\Theta_2}\ (-\sin\Theta\hat{i} + \cos\Theta\hat{j})$$

$$= P_{cm}\Theta'(-\sin\Theta\hat{i} \_ \cos\Theta\hat{j})$$

$$= (P_{cm}\ \Theta'-\sin\Theta)\hat{i} + (P_{cm}\Theta'+\cos\Theta)\hat{j} \qquad \text{(A2.16)}$$

and $\vec{F} \cdot \frac{d\vec{P}}{\partial\Theta_2}$

$$= M(P_{cm})^2\Theta'\{+\sin\Theta(+\sin\Theta)\ddot{\Theta} + \sin\Theta\dot{\Theta}^2\cos\Theta$$

$$+\cos\Theta(\cos\Theta)\ddot{\Theta} - \cos\Theta\dot{\Theta}^2\sin\Theta + \cos\Theta\frac{g}{P_{cm}}\ \}$$

$$= -M(P_{cm})^2\Theta'\ \{+\ddot{\Theta}(\sin^2+\cos^2) + \dot{\Theta}^2(\sin\Theta\cos\Theta-\sin\Theta\cos\Theta + \cos\Theta\ \frac{g}{P_{cm}}\}$$

$$= -M(P_{cm})2\ \Theta'\{+\ddot{\Theta} + \cos\Theta\frac{g}{P_{cm}}\ \}$$

which in terms of explains as follows

$$= -M(P_{cm})^2\ \Theta'\{(\Theta'\ddot{\Theta}_2+\Theta''\dot{\Theta}_2^2) + (g/P_{cm})\cos\Theta\}$$

$$= \ddot{\Theta}_2\{-M(P_{cm})^2(\Theta')^2\} + \dot{\Theta}_2^2\ \{-M(P_{cm})^2\ \Theta'\Theta''\}$$

$$-M(P_{cm})\ g\ \cos\Theta\ \Theta' \qquad \text{(A2.17)}$$

TABLE OF MASS AND INERTIA

Part Name	Mass (Lb-Sec**2/Ft)	I Mass (Ft-Lb-Sec**2)
59 Gear	.2094	.000949
120 Gear	.4816	.01546
Face Cam	.3525	.04406
Drum Cam	.8540	.0997
Feed Shaft		.000018647
Feed Rocker	.0049232	.00002724
Feed Pawl	.010888	.00032958
Ammunition	.04255	
Eject Shaft		.000016782
Eject Rocker	.0049232	.00002724
Eject Pawl	.017473	.00016703
Lock Ring	.10864	.0020653
Chamber	.68329	

TABLE A2-1

A P P E N D I X A-3

PROGRAM LISTING, FRCP

```
            C---------- ABSTRACT:  THIS MAIN PROGRAM IS THE EXECUTIVE WHICH
            C----------     CALLS THE VARIOUS ROUTINES.  THE OVERALL OBJECTIVE
            C----------     OF THE PROGRAM IS TO DEVELOP POSITIONAL RELATIONS
            C----------     BETWEEN THE VARIOUS DEPENDENT VARIABLES WHICH
            C----------     DETERMINE PART POSITIONS FOR THE VARIOUS WEAPON
            C----------     COMPONENTS.  THIS PORGRAM ESSENTIALLY MASSAGES
            C----------     AVAILABLE DATA (FROM DRAWINGS, ETC.) AND PREPARES
            C----------     OUTPUT DATA IN A FORM SUITABLE FOR USE BY THE
            C----------     AMCAWS-30 MATHEMATICAL MODEL PROGRAM.
            C
            C
            C---------- PREFORM THE INITIALIZATION NECESSARY FOR PROGRAM
            C
 0001             CALL INPUT
            C
            C---------- COMPUTE THE VARIOUS RELATIONSHIPS FOR PART POSITIONS
            C
 0002             CALL CMPUTE(JMAX)
            C
            C---------- PREPARE THE OUTPUT FOR THE AMCAWS-30 MATH PROGRAM
            C
 0003             CALL OUTPUT(JMAX)
            C
 0004             CALL PLOTNG(JMAX)
            C
 0005                 STOP
 0006                 END
```

```
0001            SUBROUTINE INPUT
          C
          C---------- ABSTRACT: SUBROUTINE INPUT SETS UP THE VARIABLES
          C----------      NECESARY FOR RUNNING THE AMCAWS-30 FUNCTIONAL
          C----------      RELATIONSHIPS PROGRAM
          C
          C
0002            COMMON/INPUTS/O2MAX,DELO2
0003            COMMON/DWGS1/FTH(1000),FR(1000),NPTSF
0004            COMMON/DWGS2/ETH(1000),ER(1000),NPTSE
0005            COMMON/DWGS3/DTH(1000),DR(1000),NPTSD
0006            COMMON/DWGS4/DLTH(1000),ALTH(1000),NPTSL
          C
          C
          C---------- PREFORM INPUT
          C
0007            O2MAX=7294.06779
0008            DELO2=O2MAX/380.
          C
          C
          C---------- FEEDCAM DATA INPUT FROM FILE 10
          C
0009            J=0
0010      100   CONTINUE
          C        READ( 6,1000,ERR=100,END=200) X,Y
          C        READ(10,1000,ERR=100,END=200) X,Y
0011               READ(10,1000,ERR=100,END=200) X,Y
0012               J=J+1
0013               IF(X.EQ.380.) NPTSF=J
0014               FTH(J)=X
0015               FR(J)=Y
0016               GO TO 100
0017      200   CONTINUE
          C
          C---------- EJECTCAM DATA INPUT FROM FILE 11
          C
0018            J=0
0019      300   CONTINUE
0020            READ(11,1000,ERR=300,END=400) X,Y
0021               J=J+1
0022               IF(X.EQ.380.) NPTSE=J
0023               ETH(J)=X
0024               ER(J)=Y
0025               GO TO 300
0026      400   CONTINUE
          C
          C---------- DPUMCAM DATA INPUT FROM FILE 12
          C
0027            J=0
0028      500   CONTINUE
0029               READ(12,1000,ERR=500,END=600) X,Y
0030               J=J+1
0031               IF(X.EQ.380.) NPTSD=J
```

```
                DTH(J)=X
                DR(J)=Y
                GO TO 500
      600   CONTINUE
      C
      C---------- LOCKCAM DATA INPUT FORM FIL 13
      C
            J=0
      700   CONTINUE
                READ(13,1000,ERR=700,END=800) X,Y
                J=J+1
                IF(X.EQ.380.) NPTSL=J
                OLTH(J)=X
               ALTH(J)=Y
                GO TO 700
      800   CONTINUE
                RETURN
      1000  FORMAT(2F17.4)
                END
```

```
0001              SUBROUTINE CMPUTE(JMAX)
            C
            C---------- ABSTRACT: SUBROUTINE CMPUTE MAKES THE FUNCTION
            C----------     CALLS THAT SET UP THE ROTATION AND DISPLACEMENT
            C----------     FUNCTIONS.
            C
            C
0002              COMMON/INPUTS/O2MAX,DELO2
0003              COMMON/OUTVAL/O2,O3,O3F,O4,O5,O6,S7,JOUT
            C
            C
0004              JOUT=0
0005              O2=-DELO2
            C
            C---------- LOOP FROM 0. TO O2MAX AND CALL FUNCTIONS
            C
0006        100   CONTINUE
0007                 O2=O2+DELO2
0008                 IF(O2.GT.O2MAX) GO TO 200
0009                 O3=FUN32(O2)
0010                 O3F=FUN3F3(O3)
0011                 O4=FUN43F(O3F)
0012                 O5=FUN53F(O3F)
0013                 O6=FUN63(O3)
0014                 S7=FUN73(O3)
0015                 JOUT=JOUT+1
0016                 CALL OUT1
0017              GO TO 100
            C
            C---------- HAVE REACHED O2MAX
            C
0018        200   CONTINUE
0019              JMAX=JOUT
0020                 RETURN
0021                 END
```

FORTRAN IV G LEVEL 21 OUTPUT DATE = 77099 09/20/55 PAGE 0001

```
0001           SUBROUTINE OUTPUT(JMAX)
          C
          C---------- ABSTRACT: SUBROUTINE OUTPUT PUTS OUT THE ARRAYS
          C----------      OF POSITIONAL INFORMATION RELATING PART POSITION
          C----------      AND ROTATIONS TO THE MOTOR INPUT ANGLE.
          C
          C
0002           COMMON/VECTOR/TH2(2001),TH3(2001),TH3F(2001),TH4(2001),
             1     TH5(2001),TH6(2001),DISPL7(2001)
0003           COMMON/OUTVAL/02,03,03F,04,05,06,S7,JOUT
          C
          C
0004           DO 100 J=1,JMAX
0005              WRITE(9,1000) J,TH2(J),TH3(J),TH3F(J)
0006              WRITE(9,1000) J,TH4(J),TH5(J),TH6(J),DISPL7(J)
0007              WRITE(6,1000) J,TH2(J),TH3(J),TH3F(J)
0008              WRITE(6,1000) J,TH4(J),TH5(J),TH6(J),DISPL7(J)
0009           WRITE(6,1002)
0010      100  CONTINUE
          C
          C    DO 200 J=1,JMAX
          C       WRITE(1,1001)TH2(J),TH3(J)
          C       WRITE(2,1001)TH2(J),TH3F(J)
          C       WRITE(3,1001)TH2(J),TH4(J)
          C       WRITE(4,1001)TH2(J),TH5(J)
          C       WRITE(8,1001)TH2(J),TH6(J)
          C       WRITE( 6,1001)TH2(J),DISPL7(J)
          C 200 CONTINUE
          C
          C
          C
0011              RETURN
0012     1000  FORMAT(' ',I5,4F16.4)
0013     1001  FORMAT(' ',2F16.4)
0014     1002  FORMAT(' ')
0015              END
```

```
0001            SUBROUTINE OUT1
           C
           C----------- ABSTRACT: SUBROUTINE OUT1 PLACES RESULTS OF
           C-----------      SUBROUTINE CMPUTE INTO ARRAYS FOR LATER
           C-----------      USE BY THE OUTPUT ROUTINE.
           C
           C
0002            COMMON/VECTOR/TH2(2001),TH3(2001),TH3F(2001),TH4(2001),
               1     TH5(2001),TH6(2001),DISPL7(2n01)
0003            COMMON/OUTVAL/02,03,03F,04,05,06,S7,JOUT
           C
           C
0004            TH2(JOUT)=02
0005            TH3(JOUT)=03
0006            TH3F(JOUT)=03F
0007            TH4(JOUT)=04
0008            TH5(JOUT)=05
0009            TH6(JOUT)=06
0010            DISPL7(JOUT)=S7
           C
           C
0011               RETURN
0012               END
```

```
0001              FUNCTION FUN32(TH2)
            C
            C----------- ABSTRACT: FUNCTION FUN32 RETURNS
            C-----------     THE DEPENDENT TH3 VALUE, GIVEN TH2.
            C
            C
0002              DATA RATIO/.052097/
            C
            C
0003              TH3=RATIO*TH2
0004              TH3=AMOD(TH3,360.000)
0005              FUN32=TH3
            C
            C
0006                  RETURN
0007                  END
```

```
0001          FUNCTION FUN3F3(TH3)
        C
        C-----------  ABSTRACT: FUNCTION FUN3F3 PROVIDES A FACECAM ANGLE
        C-----------      VS DRUMCAM ANGLE THAT WILL ULTIMATELY
        C-----------      PROVIDE A FACECAM VS MOTOR INPUT ANGLE TABLE
        C-----------      (TH3F VS TH2).
        C-----------      ALL ANGLES IN
        C-----------      DEGREES MEASURED COUNTER-CLOCKWISE
        C-----------      FROM I DIRECTION LOOKING IN -K DIRECTION
        C
        C
0002          TH3F=TH3-90.0+720.
0003          TH3F=AMOD(TH3F,360.0000)
        C
        C
0004          FUN3F3=TH3F
        C
        C
0005             RETURN
0006             END
```

```
0001            FUNCTION FUN43F(TH3F)
          C
          C---------- ABSTRACT: FUNCTION FUN43F WILL PROVIDE
          C----------      A FEED PAWL ANGLE VS FACECAM ANGLE THAT
          C----------      WILL ULTIMATELY CREATE A FEED PAWL ANGLE
          C----------      VS MOTOR INPUT ANGLE. (TH4 VS TH2).
          C----------      ALL ANGLES IN
          C----------      DEGREES MEASURED COUNTER-CLOCKWISE
          C----------      FROM I DIRECTION LOOKING IN -K DIRECTION
          C
          C
0002            DIMENSION QDUM(20)
0003            COMMON/DWGS1/FTH(1000),FR(1000),NPTSF
0004            DATA A/4.2000/,B/1.3480/,THAC/17.2800/
0005            DATA THCNST/226.0000/,TH3/36.0000/
          C
          C
          C---------- DETERMINE CURRENT RADIUS USING THCNST,AC,TH3F
          C
0006            DEG=THCNST+THAC+TH3F
0007            DEG=AMOD(DEG,360.0000)
0008            CALL INTERP(FTH,FR,NPTSF,DEG,R,DR,D2R,0,0)
          C        R=SAINT(NPTSF,FTH,FR,DEG,5,QDUM)
          C
          C---------- DETERMINE INTERIOR ANGLES, R IS SIDE C
          C
0009            C=R
0010            CALL ANGLES(A,B,C,BC,AC,AB)
0011            THAC=AC
          C
          C
0012            TH4=THCNST+AC-(180.-BC)-TH3
          C
0013            FUN43F=TH4
          C
          C
0014               RETURN
0015               END
```

```
0001          FUNCTIO  FUN53F(TH3F)
        C
        C---------- ABSTRACT: FUNCTION FUN53F PROVIDES AN EJECT
        C----------      PAWL ANGLE VS FACE CAM ANGLE THAT
        C----------      WILL ULTIMATELY CREATE AND EJECT
        C----------      PAWL ANGLE VS INPUT MOTOR ANGLE
        C----------      (TH5 VS TH2).  ALL ANGLES IN
        C----------      DEGREES MEASURED COUNTER-CLOCKWISE
        C----------      FROM I DIRECTION LOOKING IN -K DIRECTION
        C
        C
0002          DIMENSION QDUM(20)
0003          COMMON/DWGS2/ETH(1000),ER(1000),NPTSE
0004          DATA A/3.5100/,B/2.0000/,THAC/34.7300/
0005          DATA THCNST/29.5000/,TH3/79.3000/,TH1/29.5000/
        C     DATA THCNST/29.5000/,TH3/79.3000/
        C
        C
        C---------- DETERMINE CURRENT RADIUS USING TH1,AC,TH3F
        C
0006          DEG=THCNST+THAC+TH3F
0007          DEG=AMOD(DEG,360.0000)
0008          CALL INTERP(ETH,ER,NPTSE,DEG,R,DR,D2R,0,0)
        C        R=SAINT(NPTSE,ETH,ER,DEG,5,QDUM)
        C
        C---------- DETERMINE INTERIOR ANGLES, NOTE R IS SIDE C
        C
0009          C=R
0010          CALL ANGLES(A,B,C,BC,AC,AB)
0011          THAC=AC
        C
        C
0012          TH5=TH1+(180.-AB)+TH3
0013          FUN53F=TH5
        C
        C
0014          RETURN
0015          END
```

```
0001          FUNCTION FUN63(TH3)
         C
         C---------- ABSTRACT: FUNCTION FUN63 PROVIDES THE LOCK RING
         C----------      ANGLE VS DRUMCAM ANGLE THAT WILL
         C----------      ULTIMATELY PROVIDE A LOCK RING ANGLE
         C----------      VS MOTOR INPUT ANGLE.     (TH6 VS TH2).
         C----------      ALL ANGLES IN
         C----------      DEGREES MEASURED COUNTER-CLOCKWISE
         C----------      FROM I DIRECTION LOOKING IN -K DIRECTION
         C
         C
0002          DIMENSION QDUM(20)
         C    COMMON/DWGS3/DTH(1000),DR(1000),NPTSD
0003          COMMON/DWGS4/DLTH(1000),ALTH(1000),NPTSL
         C
         C
         C---------- DETERMINE LOCK RING ANGLE USING TH3
         C
0004          CALL INTERP(DLTH,ALTH,NPTSL,TH3,R,DR,D2R,0,0)
         C       R=SAINT(NPTSL,DLTH,ALTH,TH3,5,QDUM)
0005          R=R/2.23
         C    IF(R.LT.0.0) R=0.0
         C    IF(R.GT.15.) R=15.
         C
         C
0006          TH6=R
0007          FUN63=TH6
         C
         C
0008             RETURN
0009             END
```

```
0001            FUNCTION FUN73(TH3)
          C
          C---------- ABSTRACT: FUNCTION FUN73 PROVIDES THE CHAMBER
          C----------     DISPLACEMENT VS DRUMCAM ANGLE THAT
          C----------     WILL ULTIMATELY PROVIDE THE CHAMBER
          C----------     SIDPLACEMENT VS MOTOR INPUT ANGLE
          C----------     (S7 VS TH2). ALL ANGLES IN
          C----- ----     DEGREES MEASURED COUNTER-CLOCKWISE
          C----------     FROM I DIRECTION LOOKING IN -K DIRECTION
          C
          C
0002            DIMENSION QDUM(20)
0003            COMMON/DWGS3/DTH(1000),DR(1000),NPTSD
          C
          C
          C---------- DETERMINE DISPL7 KNOWING TH3
          C
0004            CALL INTERP(DTH,DR,NPTSD,TH3,R,DR,D2R,0,0)
          C        R=SAINT(NPTSD,DTH,DR,TH3,5,QDUM)
          C
          C---------- R IS THE DESIRED DISPL7
          C
0005            DISPL7=R
0006            FUN73=DISPL7
          C
          C
0007               RETURN
0008               END
```

```
0001          SUBROUTINE ANGLES(A,B,C,BC,AC,AB)
        C
        C----------  SUBROUTINE ANGLES ACCEPTS THE 3 SIDES OF ANY TRIANGLE AND CALCULATE
        C----------  THE INTERNAL ANGLES
        C----------  A,B,C ARE THE SIDE LENGTHS
        C----------  AB IS THE ANGLE BETWEEN SIDES A AND B
        C
        C---------- DETERMINE THE LONGEST SIDE
        C
0002          TEST=AMAX1(A,B,C)
0003          IF(TEST.EQ.A) GO TO 100
0004          IF(TEST.EQ.B) GO TO 200
0005          IF(TEST.EQ.C) GO TO 300
        C
        C----------  SHOULD NEVER BE HERE
        C
0006             WRITE(6,1000)
0007             GO TO 100
        C
0008    100   CONTINUE
0009          CALL ANGL(A,B,C,BC,AC,AB)
0010             GO TO 400
0011    200   CONTINUE
0012             CALL ANGL(B,A,C,AC,BC,AB)
0013             GO TO 400
0014    300   CONTINUE
0015             CALL ANGL(C,A,B,AB,BC,AC)
0016    400   CONTINUE
0017          TEST=ABS(AB+AC+BC-180.)
0018          IF(TEST.LT..1) GO TO 9000
        C
        C----------  IF HERE, ALGORITHM HAS FAILED. FOR NOW DO NOTHING
        C
0019          WRITE(6,1001)
0020          GO TO 9000
0021    9000  CONTINUE
0022             RETURN
0023    1000  FORMAT(' PASSED NO TESTS IN ANGLES')
0024    1001  FORMAT(' INTERIOR ANGLES DO NOT TOTAL TO 180.')
0025             END
```

```
FORTRAN IV G LEVEL  21                ANGL              DATE = 77099        09/20/55              PAGE 0001

 0001             SUBROUTINE ANGL(A,B,C,BC,AC,AB)
            C
            C----------  CALCULATE INTERIOR ANGLES OF A TRIANGLE WHEN A IS
            C----------  THE LARGEST SIDE AND A,B,C ARE KNOWN
            C
 0002             BC=ARCOS(((B*B+C*C-A*A)/(2.*B*C)))
 0003             AC=ARSIN(B*SIN(BC)/A)
 0004             AB=ARSIN(C*SIN(BC)/A)
            C
 0005             BC=BC*57.29578
 0006             AC=AC*57.29578
 0007             AB=AB*57.29578
 0008                RETURN
 0009                END
```

```
0001          SUBROUTINE INTERP(XV,YV,NPTS,X,YINTER,DYDX,D2YDX2,ISW,IWRAP)
        C
        C---------- ABSTRACT: SUBROUTINE INTERP SEARCHES THE
        C----------     XV AND YV VECTORS AND EXTRACTS 6
        C----------     LOCAL DATA POINTS.  THESE ARE PASSED TO
        C----------     THE POLATE ROUTINE THAT FITS A 5TH
        C----------     DEGREE POLYNOMINAL TO THEM.
        C
        C
0002          DIMENSION XV(1),YV(1),XPOLY(6),YPOLY(6),COEFF(6)
        C
        C
        C---------- DETERMINE STEP SIZE BETWEEN ADJACENT X(I)
        C---------- THEN ESTABLISH THE "LOCAL" REGION IN ARRAYS
        C
0003          M=5
0004          N=6
0005          XVAL=X
0006          STEP=XV(2)-XV(1)
0007          NSTEP=IFIX(XVAL/STEP + .5)
0008          IF(NSTEP.LT.5 .AND. IWRAP.EQ.0) NSTEP=4
0009          IF(NSTEP.LT.5 .AND. IWRAP.EQ.0) GO TO 50
0010          IF((NPTS-2).LT.NSTEP .AND. IWRAP.EQ.0) NSTEP=NPTS-2
0011          IF((NPTS-2).LT.NSTEP .AND. IWRAP.EQ.0) GO TO 50
0012          IF(NSTEP.LT.5)  XVAL=XVAL+XV(NPTS)
0013          IF(NSTEP.LT.5) NSTEP=NPTS+NSTEP
        C
        C---------- SET UP THE 6 ELEMENT ARRAYS TO BE FIT
        C
0014    50    CONTINUE
0015             BASE=0.0
0016          DO 100 J=1,6
0017             JVAL=NSTEP-4+J
        C        XPOLY(J)=XV(JVAL)
0018                 XPOLY(J)=BASE + STEP*FLOAT(J)
0019             YPOLY(J)=YV(JVAL)
0020    100   CONTINUE
0021              OFFSET=XV(NSTEP-3)-XPOLY(1)
0022              XVAL=XVAL-OFFSET
        C
        C---------- ARRAYS ARE NOW TO BE FIT
        C
0023          CALL POLATE(XPOLY,YPOLY,COEFF)
        C
        C---------- EVALUATE POLY FOR 0,1,2 DERIVITIVES
        C---------- AT VALUE XVAL BASED ON ISW
        C
0024          P=COEFF(6)
0025          DO 200 J=1,5
0026             I=5-J
0027             P=P*XVAL+COEFF(I+1)
0028    200   CONTINUE
        C
```

```
          IF(ISW.LE.1) GO TO 9999
C
          DO 300 J=1,5
             COEFF(J)=J*COEFF(J+1)
  300     CONTINUE
C
          DP=COEFF(5)
          DO 400 J=1,4
             I=4-J
             DP=DP*XVAL+COEFF(I+1)
  400     CONTINUE
C
          DO 500 J=1,4
             COEFF(J)=J*COEFF(J+1)
  500     CONTINUE
C
          D2P=COEFF(4)
          DO 600 J=1,3
             I=3-J
             D2P=D2P*XVAL+COEFF(I+1)
  600     CONTINUE
C
C
          DYDX=DP
          D2YDX2=D2P
C
 9999     CONTINUE
C
          YINTER=P
C
C
             RETURN
             END
```

```
0001              SUBROUTINE POLATE(XV,YV,COEFF)
            C
            C---------- ABSTRACT: SUBROUTINE POLATE PREFORMS A
            C----------     POLYNUMINAL REGRESSION ON THE
            C----------     DATA POINTS IN VECTORS XY,YV AND
            C----------     RETURNS A COEFFICIENT VECTOR.  THE
            C----------     DEGREE OF THE POLY WILL BE 5.
            C
            C
0002              DIMENSION XV(1),YV(1),COEFF(1)
0003              DIMENSION FIT(36),FITINV(36),DCOEFF(6),DYPOLY(6)
0004              DOUBLE PRECISION FIT,FITINV,DCOEFF,DYPOLY
            C
            C
            C---------- INITIALIZE FIT MATRIX FOR CURVE FITTING
            C
0005              DO 100 I=1,6
0006                 FIT(I)=1.
0007                 FIT(I+6)=XV(I)
0008                 FIT(I+12)=XV(I)**2
0009                 FIT(I+18)=XV(I)**3
0010                 FIT(I+24)=XV(I)**4
0011                 FIT(I+30)=XV(I)**5
0012                 DYPOLY(I)=YV(I)
0013        100   CONTINUE
            C
            C---------- ZERO THE INVERSE MATRIX THEN MAKE IT IDENITY
            C
0014              DO 200 I=2,35
0015                 FITINV(I)=0.0
0016        200   CONTINUE
            C
0017                 FITINV(1)=1.
0018                 FITINV(8)=1.
0019                 FITINV(15)=1.
0020                 FITINV(22)=1.
0021                 FITINV(29)=1.
0022                 FITINV(36)=1.
            C
            C---------- INVERT THE FIT MATRIX USING GELG
            C
0023              CALL DGELG(FITINV,FIT,6,6,1.E-07,IERR)
            C
            C---------- DETERMINE THE POLY COEFFICIENTS
            C
0024              CALL DGMPRD(FITINV,DYPOLY,DCOEFF,6,6,1)
            C
            C---------- TRANSPOSE DOUBLE PRECISION COEFF TO SINGLE
            C
0025              DO 300 I=1,6
0026                 COEFF(I)=DCOEFF(I)
0027        300   CONTINUE
            C
0028                 RETURN
0029                 END
```

```
               C      SUBROUTINE GMPRD(A,B,R,N,M,L)
               C
               C         PURPOSE
               C            MULTIPLY TWO GENERAL MATRICES TO FORM A RESULTANT GENERAL
               C            MATRIX
               C
               C         USAGE
               C            CALL GMPRD(A,B,R,N,M,L)
               C
               C         DESCRIPTION OF PARAMETERS
               C            A - NAME OF FIRST INPUT MATRIX
               C            B - NAME OF SECOND INPUT MATRIX
               C            R - NAME OF OUTPUT MATRIX
               C            N - NUMBER OF ROWS IN A
               C            M - NUMBER OF COLUMNS IN A AND ROWS IN B
               C            L - NUMBER OF COLUMNS IN B
               C
               C         REMARKS
               C            ALL MATRICES MUST BE STORED AS GENERAL MATRICES
               C            MATRIX R CANNOT BE IN THE SAME LOCATION AS MATRIX A
               C            MATRIX R CANNOT BE IN THE SAME LOCATION AS MATRIX B
               C            NUMBER OF COLUMNS OF MATRIX A MUST BE EQUAL TO NUMBER OF ROW
               C            OF MATRIX B
               C
               C         SUBROUTINES AND FUNCTION SUBPRnGRAMS REQUIRED
               C            NONE
               C
               C         METHOD
               C            THE M BY L MATRIX B IS PREMULTIPLIED BY THE N BY M MATRIX A
               C            AND THE RESULT IS STORED IN THE N BY L MATRIX R.
               C
               C      ..............................................................
0001                  SUBROUTINE GMPRD(A,B,R,N;M,L)
               C
0002                  IMPLICIT REAL*8 (A-H,O-Z)
0003                  DIMENSION A(1),B(1),R(1)
               C
0004                  IR=0
0005                  IK=-M
0006                  DO 10 K=1,L
0007                  IK=IK+M
0008                  DO 10 J=1,N
0009                  IR=IR+1
0010                  JI=J-N
0011                  IB=IK
0012                  R(IR)=0
0013                  DO 10 I=1,M
0014                  JI=JI+N
0015                  IB=IB+1
0016               10 R(IR)=R(IR)+A(JI)*B(IB)
0017                  RETURN
0018                  END
```

```
C      SUBROUTINE DGMPRD(A,B,R,N,M,L)
C
C         PURPOSE
C            MULTIPLY TWO GENERAL MATRICES TO FORM A RESULTANT GENERAL
C            MATRIX
C
C         USAGE
C            CALL GMPRD(A,B,R,N,M,L)
C
C         DESCRIPTION OF PARAMETERS
C            A - NAME OF FIRST INPUT MATRIX
C            B - NAME OF SECOND INPUT MATRIX
C            R - NAME OF OUTPUT MATRIX
C            N - NUMBER OF ROWS IN A
C            M - NUMBER OF COLUMNS IN A AND ROWS IN B
C            L - NUMBER OF COLUMNS IN B
C
C         REMARKS
C            ALL MATRICES MUST BE STORED AS GENERAL MATRICES
C            MATRIX R CANNOT BE IN THE SAME LOCATION AS MATRIX A
C            MATRIX R CANNOT BE IN THE SAME LOCATION AS MATRIX B
C            NUMBER OF COLUMNS OF MATRIX A MUST BE EQUAL TO NUMBER OF ROW
C            OF MATRIX B
C
C         SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C            NONE
C
C         METHOD
C            THE M BY L MATRIX B IS PREMULTIPLIED BY THE N BY M MATRIX A
C            AND THE RESULT IS STORED IN THE N BY L MATRIX R.
C
C      ..................................................................
       SUBROUTINE DGMPRD(A,B,R,N,M,L)
C
       IMPLICIT REAL*8 (A-H,O-Z)
       DIMENSION A(1),B(1),R(1)
       IR=0
       IK=-M
       DO 10 K=1,L
          IK=IK+M
          DO 20 J=1,N
             IR=IR+1
             JI=J-N
             IB=IK
             R(IR)=0.D+00
             DO 30 I=1,M
                JI=JI+N
                IB=IB+1
                R(IR)=R(IR)+A(JI)*B(IB)
 30          CONTINUE
 20       CONTINUE
 10    CONTINUE
          RETURN
          END
```

```
           C      SUBROUTINE DGELG(R,A,M,N,EPS,IER)
           C
           C         PURPOSE
           C            TO SOLVE A GENERAL SYSTEM OF SIMULTANEOUS LINEAR EQUATIONS.
           C
           C         USAGE
           C            CALL DGELG(R,A,M,N,EPS,IER)
           C
           C         DESCRIPTION OF PARAMETERS
           C            R      - DOUBLE PRECISION M BY N RIGHT HAND SIDE MATRIX
           C                     (DESTROYED). ON RETURN R CONTAINS THE SOLUTIONS
           C                     OF THE EQUATIONS.
           C            A      - DOUBLE PRECISION M BY M COEFFICIENT MATRIX
           C                     (DESTROYED).
           C            M      - THE NUMBER OF EQUATIONS IN THE SYSTEM.
           C            N      - THE NUMBER OF RIGHT HAND SIDE VECTORS.
           C            EPS    - SINGLE PRECISION INPUT CONSTANT WHICH IS USED AS
           C                     RELATIVE TOLERANCE FOR TEST ON LOSS OF
           C                     SIGNIFICANCE.
           C            IER    - RESULTING ERROR PARAMETER CODED AS FOLLOWS
           C                     IER=0  - NO ERROR,
           C                     IER=-1 - NO RESULT BECAUSE OF M LESS THAN 1 OR
           C                              PIVOT ELEMENT AT ANY ELIMINATION STEP
           C                              EQUAL TO 0,
           C                     IER=K  - WARNING DUE TO POSSIBLE LOSS OF SIGNIFI-
           C                              CANCE INDICATED AT ELIMINATION STEP K+1,
           C                              WHERE PIVOT ELEMENT WAS LESS THAN OR
           C                              EQUAL TO THE INTERNAL TOLERANCE EPS TIMES
           C                              ABSOLUTELY GREATEST ELEMENT OF MATRIX A.
           C
           C         REMARKS
           C            INPUT MATRICES R AND A ARE ASSUMED TO BE STORED COLUMNWISE
           C            IN M*N RESP. M*M SUCCESSIVE STORAGE LOCATIONS. ON RETURN
           C            SOLUTION MATRIX R IS STORED COLUMNWISE TOO.
           C            THE PROCEDURE GIVES RESULTS IF THE NUMBER OF EQUATIONS M IS
           C            GREATER THAN 0 AND PIVOT ELEMENTS AT ALL ELIMINATION STEPS
           C            ARE DIFFERENT FROM 0. HOWEVER WARNING IER=K - IF GIVEN -
           C            INDICATES POSSIBLE LOSS OF SIGNIFICANCE. IN CASE OF A WELL
           C            SCALED MATRIX A AND APPROPRIATE TOLERANCE EPS, IER=K MAY BE
           C            INTERPRETED THAT MATRIX A HAS THE RANK K. NO WARNING IS
           C            GIVEN IN CASE M=1.
           C
           C         SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
           C            NONE
           C
           C         METHOD
           C            SOLUTION IS DONE BY MEANS OF GAUSS-ELIMINATION WITH
           C            COMPLETE PIVOTING.
           C
           C      ..................................................................
 0001             SUBROUTINE DGELG(R,A,M,N,EPS,IER)
           C
           C
```

```
            C
 0002             DIMENSION A(1),R(1)
 0003             DOUBLE PRECISION R,A,PIV,TB,TOL,PIVI
 0004             IF(M)23,23,1
            C
            C     SEARCH FOR GREATEST ELEMENT IN MATRIX A
 0005           1 IER=0
 0006             PIV=0.D0
 0007             MM=M*M
 0008             NM=N*M
 0009             DO 3 L=1,MM
 0010             TB=DABS(A(L))
 0011             IF(TB-PIV)3,3,2
 0012           2 PIV=TB
 0013             I=L
 0014           3 CONTINUE
 0015             TOL=EPS*PIV
            C     A(I) IS PIVOT ELEMENT. PIV CONTAINS THE ABSOLUTE VALUE OF A(I).
            C
            C
            C     START ELIMINATION LOOP
 0016             LST=1
 0017             DO 17 K=1,M
            C
            C     TEST ON SINGULARITY
 0018             IF(PIV)23,23,4
 0019           4 IF(IER)7,5,7
 0020           5 IF(PIV-TOL)6,6,7
 0021           6 IER=K-1
 0022           7 PIVI=1.D0/A(I)
 0023             J=(I-1)/M
 0024             I=I-J*M-K
 0025             J=J+1-K
            C     I+K IS ROW-INDEX, J+K COLUMN-INDEX OF PIVOT ELEMENT
            C
            C     PIVOT ROW REDUCTION AND ROW INTERCHANGE IN RIGHT HAND SIDE R
 0026             DO 8 L=K,NM,M
 0027             LL=L+I
 0028             TB=PIVI*R(LL)
 0029             R(LL)=R(L)
 0030           8 R(L)=TB
            C
            C     IS ELIMINATION TERMINATED
 0031             IF(K-M)9,18,18
            C
            C     COLUMN INTERCHANGE IN MATRIX A
 0032           9 LEND=LST+M-K
 0033             IF(J)12,12,10
 0034          10 II=J*M
 0035             DO 11 L=LST,LEND
 0036             TB=A(L)
 0037             LL=L+II
 0038             A(L)=A(LL)
```

```
 0039          11 A(LL)=TB
            C
            C     ROW INTERCHANGE AND PIVOT ROW REDUCTION IN MATRIX A
 0040          12 DO 13 L=LST,MM,M
 0041             LL=L+I
 0042             TB=PIVI*A(LL)
 0043             A(LL)=A(L)
 0044          13 A(L)=TB
            C
            C     SAVE COLUMN INTERCHANGE INFORMATION
 0045             A(LST)=J
            C
            C     ELEMENT REDUCTION AND NEXT PIVOT SEARCH
 0046             PIV=0.D0
 0047             LST=LST+1
 0048             J=0
 0049             DO 16 II=LST,LEND
 0050             PIVI=-A(II)
 0051             IST=II+M
 0052             J=J+1
 0053             DO 15 L=IST,MM,M
 0054             LL=L-J
 0055             A(L)=A(L)+PIVI*A(LL)
 0056             TB=DABS(A(L))
 0057             IF(TB-PIV)15,15,14
 0058          14 PIV=TB
 0059             I=L
 0060          15 CONTINUE
 0061             DO 16 L=K,NM,M
 0062             LL=L+J
 0063          16 R(LL)=R(LL)+PIVI*R(L)
 0064          17 LST=LST+M
            C     END OF ELIMINATION LOOP
            C
            C
            C     BACK SUBSTITUTION AND BACK INTERCHANGE
 0065          18 IF(M-1)23,22,19
 0066          19 IST=NM+M
 0067             LST=M+1
 0068             DO 21 I=2,M
 0069             II=LST-I
 0070             IST=IST-LST
 0071             L=IST-M
 0072             L=A(L)+.5D0
 0073             DO 21 J=II,NM,M
 0074             TB=R(J)
 0075             LL=J
 0076             DO 20 K=IST,MM,M
 0077             LL=LL+1
 0078          20 TB=TB-A(K)*R(LL)
 0079             K=J+L
 0080             R(J)=R(K)
 0081          21 R(K)=TB
```

```
 0082          22 RETURN
             C
             C
             C      ERROR RETURN
 0083          23 IER=-1
 0084             RETURN
 0085             END
```

```
0001           SUBROUTINE PLOTNG(JMAX)
         C
         C
0002           DIMENSION IBUF(1000)
0003           COMMON/VECTOR/TH2(2001),TH3(2001),TH3F(2001),TH4(2001),
              1     TH5(2001),TH6(2001),DISPL7(2n01)
         C
0004           CALL PLOTS(IBUF,1000,14)
0005           CALL FACTOR(1.0)
0606           CALL PLOT(5.0,-36.0,-3)
0007           CALL PLOT(2.0,2.5,-3)
0008           CALL SCALE(TH2,10.,JMAX,1)
0009           CALL SCALE( TH3,8.0,JMAX,1)
0010           CALL SCALE( TH4,8.0,JMAX,1)
0011           CALL SCALE( TH5,8.0,JMAX,1)
0012           CALL SCALE( TH6,8.0,JMAX,1)
0013           CALL SCALE( DISPL7,8.0,JMAX,1)
         C
0014           IF=JMAX+1
0015           ID=IF+1
         C
0016           CALL AXIS(0.0,0.0,27HMOTOR INPUT ANGLE (DEGREES),-27,10.,0.0,
              1          TH2(IF),TH2(ID))
0017           CALL AXIS( 0.0,0.0,24HDRUM CAM ANGLE (DEGREES),+24,
              1          8.,90.,TH3(IF),TH3(ID))
0018            CALL NEWPEN(2)
0019           CALL LINE(TH2,TH3,JMAX,1,0,0)
0020            CALL NEWPEN(1)
0021           CALL PLOT(15.0,0.0,-3)
         C
0022           CALL AXIS(0.0,0.0,27HMOTOR INPUT ANGLE (DEGREES),-27,10.,0.0,
              1          TH2(IF),TH2(ID))
0023           CALL AXIS( 0.0,0.0,24HFEED CAM ANGLE (DEGREES),+24,
              1          8.,90.,TH4(IF),TH4(ID))
0024            CALL NEWPEN(2)
0025           CALL LINE(TH2,TH4,JMAX,1,0,1)
0026            CALL NEWPEN(1)
0027           CALL PLOT(15.0,0.0,-3)
         C
0028           CALL AXIS(0.0,0.0,27HMOTOR INPUT ANGLE (DEGREES),-27,10.,0.0,
              1          TH2(IF),TH2(ID))
0029           CALL AXIS( 0.0,0.0,25HEJECT CAM ANGLE (DEGREES),+25,
              1          8.,90.,TH5(IF),TH5(ID))
0030            CALL NEWPEN(2)
0031           CALL LINE(TH2,TH5,JMAX,1,0,5)
0032            CALL NEWPEN(1)
0033           CALL PLOT(15.0,0.0,-3)
         C
0034           CALL AXIS(0.0,0.0,27HMOTOR INPUT ANGLE (DEGREES),-27,10.,0.0,
              1          TH2(IF),TH2(ID))
0035           CALL AXIS( 0.0,0.0,24HLOCK CAM ANGLE (DEGREES),+24,
              1          8.,90.,TH6(IF),TH6(ID))
0036            CALL NEWPEN(2)
```

```
0037          CALL LINE(TH2,TH6,JMAX,1,0,10)
0038           CALL NEWPEN(1)
0039          CALL PLOT(15.0,0.0,-3)
      C
0040          CALL AXIS(0.0,0.0,27HMOTOR INPUT ANGLE (DEGREES),-27,10.,0.0,
             1            TH2(IF),TH2(ID))
0041          CALL AXIS( 0.0,0.0,29HCHAMBER DISPLACEMENT (INCHES),+29,
             1            8.,90.,DISPL7(IF),DISPL7(ID))
0042           CALL NEWPEN(2)
0043          CALL LINE(TH2,DISPL7,JMAX,1,0,11)
0044           CALL NEWPEN(1)
0045          CALL PLOT(20.0,0.0,-3)
      C
0046          CALL AXIS(0.0,0.0,27HMOTOR INPUT ANGLE (DEGREES),-27,10.,0.0,
             1            TH2(IF),TH2(ID))
0047          CALL AXIS( 0.0,0.0,24HDRUM CAM ANGLE (DEGREES),+24,
             1            8.,90.,TH3(IF),TH3(ID))
0048          CALL AXIS(-0.5,0.0,24HFEED CAM ANGLE (DEGREES),+24,
             1            8.,90.,TH4(IF),TH4(ID))
0049          CALL AXIS(-1.0,0.0,25HEJECT CAM ANGLE (DEGREES),+25,
             1            8.,90.,TH5(IF),TH5(ID))
0050          CALL AXIS(-1.5,0.0,24HLOCK CAM ANGLE (DEGREES),+24,
             1            8.,90.,TH6(IF),TH6(ID))
0051          CALL AXIS(-2.0,0.0,29HCHAMBER DISPLACEMENT (INCHES),+29,
             1            8.,90.,DISPL7(IF),DISPL7(ID))
0052           CALL NEWPEN(2)
0053          CALL LINE(TH2,TH6,JMAX,1,9,10)
0054          CALL LINE(TH2,TH5,JMAX,1,9,5)
0055          CALL LINE(TH2,TH4,JMAX,1,9,1)
0056          CALL LINE(TH2,TH3,JMAX,1,9,0)
0057          CALL LINE(TH2,DISPL7,JMAX,1,9,11)
0058          BASEX=-.32
0059          BASEY=1.5
0060          BASEYY=BASEY+0.25
0061          CALL SYMBOL(BASEX,BASEY,.140,0,90.,-1)
0062          CALL SYMBOL(BASEX,BASEYY,.140,0,90.,-2)
0063          CALL SYMBOL(BASEX-.5,BASEY ,.140,1,90.,-1)
0064          CALL SYMBOL(BASEX-.5,BASEYY,.140,1,90.,-2)
0065          CALL SYMBOL(BASEX-1.0,BASEY ,.140,5,90.,-1)
0066          CALL SYMBOL(BASEX-1.0,BASEYY,.140,5,90.,-2)
0067          CALL SYMBOL(BASEX-1.5,BASEY ,.140,10,90.,-1)
0068          CALL SYMBOL(BASEX-1.5,BASEYY,.140,10,90.,-2)
0069          CALL SYMBOL(BASEX-2.0,BASEY ,.140,11,90.,-1)
0070          CALL SYMBOL(BASEX-2.0,BASEYY,.140,11,90.,-2)
      C
0071          CALL PLOT(20.,0.0,999)
0072          RETURN
0073          END
```

J J	TH2 TH4	TH3 TH5	TH3F TH6	DISPL 7
1 1	0.0 139.5305	0.0 234.5235	270.0000 14.3942	0.0
2 2	19.1949 139.5305	1.0000 234.5235	271.0000 14.6095	0.0
3 3	38.3898 139.5305	2.0000 234.5235	272.0000 14.7360	0.0
4 4	57.5847 139.5305	3.0000 234.5235	273.0000 14.7900	0.0
5 5	76.7796 139.5305	4.0000 234.5235	274.0000 14.3010	0.0
6 6	95.9745 139.5305	5.0000 234.5235	274.9998 14.7982	0.0
7 7	115.1694 139.5305	6.0000 234.5235	275.9998 14.7982	0.0
8 8	134.3643 139.5305	7.0000 234.5235	276.9998 14.7982	0.0
9 9	153.5592 139.5305	8.0000 234.5235	277.9998 14.7982	0.0
10 10	172.7541 139.5305	9.0000 234.5235	278.9998 14.7982	0.0
11 11	191.9490 139.5305	10.0000 234.5235	279.9998 14.7982	0.0
12 12	211.1439 139.5305	11.0000 234.5235	280.9998 14.7982	0.0
13 13	230.3388 139.5305	12.0000 234.5235	281.9998 14.7982	0.0
14 14	249.5337 139.5305	13.0000 234.5235	282.9998 14.7982	0.0
15 15	268.7285 139.5305	13.9999 234.5235	283.9998 14.7982	0.0
16 16	287.9233 139.5305	14.9999 234.5235	284.9998 14.7982	0.0
17 17	307.1182 139.5305	15.9999 234.5235	285.9998 14.7982	0.0
18 18	326.3130 139.5305	16.9999 234.5235	286.9998 14.7982	0.0
19 19	345.5078 139.5305	17.9999 234.5235	287.9998 14.7982	0.0
20 20	364.7026 139.5305	18.9999 234.5235	288.9998 14.7982	0.0

21	383.8975	19.9999	289.9998	
21	139.5305	234.5235	14.7982	0.0
22	403.0923	20.9999	290.9998	
22	139.5305	234.5235	14.7982	0.0
23	422.2871	21.9999	291.9998	
23	139.5305	234.5235	14.7982	0.0
24	441.4819	22.9999	292.9998	
24	139.5305	234.5235	14.7982	0.0
25	460.6768	23.9999	293.9998	
25	139.5305	234.5235	14.7982	0.0
26	479.8716	24.9999	294.9998	
26	139.5305	234.5235	14.7982	0.0
27	499.0664	25.9998	295.9998	
27	139.5305	234.5235	14.7982	0.0
28	518.2612	26.9998	296.9998	
28	139.5305	234.5235	14.7982	0.0
29	537.4561	27.9998	297.9998	
29	139.5305	234.5235	14.7982	0.0
30	556.6509	28.9998	298.9998	
30	139.5305	234.5235	14.7982	0.0
31	575.8457	29.9998	299.9998	
31	139.5305	234.5235	14.7320	0.0
32	595.0405	30.9998	300.9998	
32	139.5305	234.5235	14.5286	0.0
33	614.2354	31.9998	301.9998	
33	139.5305	234.5235	14.1811	0.0
34	633.4302	32.9998	302.9998	
34	139.5305	234.5235	13.6826	0.0
35	652.6250	33.9998	303.9998	
35	139.5305	234.5235	13.0261	0.0
36	671.8198	34.9998	304.9998	
36	139.5305	234.5235	12.2045	0.0
37	691.0146	35.9998	305.9998	
37	139.5305	234.5235	11.2110	0.0
38	710.2095	36.9998	306.9998	
38	139.5305	234.5235	10.1291	0.0
39	729.4043	37.9998	307.9998	
39	139.5305	234.5235	9.4172	0.0
40	748.5991	38.9998	308.9998	
40	139.5305	234.5235	8.5517	0.0

41	767.7939	39.9998	309.9998	
41	139.5305	234.5235	7.3831	0.0
42	786.9888	40.9997	310.9995	
42	139.5305	234.5235	5.8300	0.0
43	806.1836	41.9997	311.9995	
43	139.5305	234.5235	4.7895	0.0
44	825.3784	42.9997	312.9995	
44	139.5305	234.5235	4.2506	0.0
45	844.5732	43.9997	313.9995	
45	139.5305	234.5235	3.5876	0.0
46	863.7681	44.9997	314.9995	
46	139.5305	234.5235	2.7729	0.0
47	882.9629	45.9997	315.9995	
47	139.5305	234.5235	1.8850	0.0
48	902.1577	46.9997	316.9995	
48	139.5305	234.5235	1.3454	0.0
49	921.3525	47.9997	317.9995	
49	139.5305	234.5235	1.1152	0.0
50	940.5474	48.9997	318.9995	
50	139.5305	234.5235	0.8032	0.0
51	959.7422	49.9997	319.9995	
51	139.5305	234.5235	0.5298	0.0
52	978.9370	50.9997	320.9995	
52	139.5305	234.5235	0.3267	0.0
53	998.1318	51.9997	321.9995	
53	139.5305	234.5235	0.1834	0.0000
54	1017.3267	52.9996	322.9995	
54	139.5305	234.5235	0.0896	-0.0000
55	1036.5215	53.9996	323.9995	
55	139.5305	234.5235	0.0347	0.0001
56	1055.7163	54.9996	324.9995	
56	139.5305	234.5235	0.0085	0.0004
57	1074.9111	55.9996	325.9995	
57	139.5305	234.5235	0.0004	0.0010
58	1094.1060	56.9996	326.9995	
58	139.5305	234.5235	-0.0000	0.0020
59	1113.3008	57.9996	327.9995	
59	139.5305	234.5235	-0.0000	0.0035
60	1132.4956	58.9996	328.9995	
60	139.5305	234.5235	-0.0000	0.0056

61	1151.6904	59.9996	329.9995	0.0083
61	139.5305	234.5235	0.0	
62	1170.8853	60.9996	330.9995	0.0119
62	139.5305	234.5235	0.0	
63	1190.0801	61.9996	331.9995	0.0162
63	139.5305	234.5235	0.0	
64	1209.2749	62.9996	332.9995	0.0216
64	139.5305	234.5235	0.0	
65	1228.4697	63.9996	333.9995	0.0280
65	139.5305	234.5235	0.0	
66	1247.6646	64.9996	334.9995	0.0355
66	139.5305	234.5235	0.0	
67	1266.8594	65.9996	335.9995	0.0442
67	139.5305	234.5235	0.0	
68	1286.0542	66.9996	336.9995	0.0542
68	139.5305	234.5235	0.0	
69	1305.2490	67.9995	337.9995	0.0656
69	139.5305	234.5235	0.0	
70	1324.4438	68.9995	338.9995	0.0784
70	139.5305	234.5235	0.0	
71	1343.6387	69.9995	339.9995	0.0928
71	139.5305	234.5235	0.0	
72	1362.8335	70.9995	340.9995	0.1088
72	139.5305	234.5235	0.0	
73	1382.0283	71.9995	341.9995	0.1264
73	139.5305	234.5235	0.0	
74	1401.2231	72.9995	342.9995	0.1457
74	139.5305	234.5235	0.0	
75	1420.4180	73.9995	343.9993	0.1669
75	139.5305	234.5235	0.0	
76	1439.6128	74.9995	344.9993	0.1899
76	139.5305	234.5235	0.0	
77	1458.8076	75.9995	345.9993	0.2148
77	139.5305	234.5235	0.0	
78	1478.0024	76.9995	346.9993	0.2417
78	139.5305	234.5235	0.0	
79	1497.1973	77.9995	347.9993	0.2705
79	139.5305	234.5235	0.0	
80	1516.3921	78.9995	348.9993	0.3015
80	139.5305	234.5235	0.0	

81	1535.5869	79.9995	349.9993	
81	139.5305	234.5235	0.0	0.3345
82	1554.7817	80.9995	350.9993	
82	139.5305	234.5235	0.0	0.3696
83	1573.9766	81.9995	351.9993	
83	139.5305	234.5235	0.0	0.4069
84	1593.1714	82.9994	352.9993	
84	139.5305	234.5235	0.0	0.4464
85	1612.3662	83.9994	353.9993	
85	139.5305	234.5235	0.0	0.4881
86	1631.5610	84.9994	354.9993	
85	139.5305	234.5235	0.0	0.5321
87	1650.7559	85.9994	355.9993	
87	139.5305	234.5235	0.0	0.5682
88	1669.9507	86.9994	356.9993	
88	139.5305	234.5235	0.0	0.6367
89	1689.1455	87.9994	357.9993	
89	139.5305	234.5235	0.0	0.6774
90	1708.3403	88.9994	358.9993	
90	139.5305	234.5235	0.0	0.7304
91	1727.5352	89.9994	359.9993	
91	39.5305	234.5235	0.0	0.7856
92	746.7300	90.9994	0.9993	
92	139.5305	234.5235	0.0	0.8431
93	1765.9248	91.9994	1.9993	
93	139.5305	234.5235	0.0	0.9029
94	1785.1196	92.9994	2.9993	
94	139.5305	234.5235	0.0	0.9649
95	1804.3145	93.9994	3.9993	
95	139.5305	234.5235	0.0	1.0292
96	1823.5093	94.9993	4.9993	
96	139.5305	234.5235	0.0	1.0956
97	1842.7041	95.9993	5.9993	
97	139.5305	234.5235	0.0	1.1642
98	1861.8989	96.9993	6.9993	
98	139.530[illegible]	234.5235	0.0	1.2350
99	1881.0937	97.9993	7.9993	
99	139.5305	234.5235	0.0	1.3077
100	1900.2886	98.9993	8.9993	
100	139.5305	234.5235	0.0	1.3827

101	1919.4834	99.9993	9.9993	
101	139.5305	234.5235	0.0	1.4596
102	1938.6782	100.9993	10.9993	
102	139.5305	234.5235	0.0	1.5386
103	1957.8730	101.9993	11.9993	
103	139.5305	234.5235	0.0	1.6194
104	1977.0679	102.9993	12.9993	
104	139.5305	234.5235	0.0	1.7022
105	1996.2627	103.9993	13.9993	
105	139.5305	234.5235	0.0	1.7868
106	2015.4575	104.9993	14.9993	
106	139.5305	234.5235	0.0	1.8731
107	2034.6523	105.9993	15.9993	
107	139.5305	234.5235	0.0	1.9612
108	2053.8472	106.9993	16.9993	
108	139.5305	234.5235	0.0	2.0508
109	2073.0420	107.9993	17.9990	
109	139.5305	234.5235	0.0	2.1421
110	2092.2368	108.9993	18.9990	
110	139.5305	234.5235	0.0	2.2348
111	2111.4316	109.9992	19.9990	
111	139.5305	234.5235	0.0	2.3289
112	2130.6265	110.9992	20.9990	
112	139.5305	234.5235	0.0	2.4244
113	2149.8213	111.9992	21.9990	
113	139.5305	234.5235	0.0	2.5211
114	2169.0161	112.9992	22.9990	
114	139.5305	234.5235	0.0	2.6190
115	2188.2109	113.9992	23.9990	
115	139.5305	234.5235	0.0	2.7180
116	2207.4058	114.9992	24.9990	
116	139.5305	234.5235	0.0	2.8180
117	2226.6006	115.9992	25.9990	
117	139.5305	234.5235	0.0	2.9189
118	2245.7954	116.9992	26.9990	
118	139.5305	234.5235	0.0	3.0206
119	2264.9902	117.9992	27.9990	
119	139.5305	234.5235	0.0	3.1230
120	2284.1851	118.9992	28.9990	
120	139.5305	234.5235	0.0	3.2261

121	2303.3799	119.9992	29.9990	
121	139.5305	234.5235	0.0	3.3297
122	2322.5747	120.9992	30.9990	
122	139.5305	234.5235	0.0	3.4337
123	2341.7695	121.9991	31.9990	
123	139.5305	234.5235	0.0	3.5381
124	2360.9644	122.9991	32.9990	
124	139.5305	234.5235	0.0	3.6427
125	2380.1592	123.9991	33.9990	
125	139.5305	234.5235	0.0	3.7475
126	2399.3540	124.9991	34.9990	
126	139.5305	234.5235	0.0	3.8523
127	2418.5488	125.9991	35.9990	
127	139.5305	234.5235	0.0	3.9571
128	2437.7437	126.9991	36.9990	
128	139.5305	234.5235	0.0	4.0617
129	2456.9385	127.9991	37.9990	
129	139.5305	234.5235	0.0	4.1661
130	2476.1333	128.9991	38.9990	
130	139.5305	234.5235	0.0	4.2701
131	2495.3281	129.9991	39.9990	
131	139.5305	234.5235	0.0	4.3737
132	2514.5229	130.9991	40.9990	
132	139.5305	234.5235	0.0	4.4768
133	2533.7178	131.9991	41.9990	
133	139.5305	234.5235	0.0	4.5792
134	2552.9126	132.9991	42.9990	
134	139.5305	234.5235	0.0	4.6809
135	2572.1074	133.9991	43.9990	
135	139.5305	234.5235	0.0	4.7818
136	2591.3022	134.9991	44.9990	
136	139.5305	234.5235	0.0	4.8818
137	2610.4971	135.9991	45.9990	
137	139.5305	234.5235	0.0	4.9808
138	2629.6919	136.9990	46.9990	
138	139.5305	234.5235	0.0	5.0787
139	2648.8867	137.9990	47.9990	
139	139.5305	234.5235	0.0	5.1754
140	2668.0815	138.9990	48.9990	
140	139.5305	234.5235	0.0	5.2709

141	2687.2764	139.9990	49.9990	
141	139.5305	234.5235	0.0	5.3650
142	2706.4712	140.9990	50.9988	
142	139.5305	234.5235	0.0	5.4577
143	2725.6660	141.9990	51.9988	
143	139.5305	234.5235	0.0	5.5490
144	2744.8608	142.9990	52.9988	
144	139.5305	234.5235	0.0	5.6386
145	2764.0557	143.9990	53.9988	
145	139.5305	234.5235	0.0	5.7267
146	2783.2505	144.9990	54.9988	
146	139.5305	234.5235	0.0	5.8130
147	2802.4453	145.9990	55.9988	
147	139.5305	234.5235	0.0	5.8976
148	2821.6401	146.9990	56.9988	
148	139.5305	234.5235	0.0	5.9804
149	2840.8350	147.9990	57.9988	
149	139.5305	234.5235	0.0	6.0612
150	2860.0298	148.9989	58.9988	
150	139.5305	234.5235	0.0	6.1402
151	2879.2246	149.9989	59.9980	
151	139.5305	234.5235	0.0	6.2172
152	2898.4194	150.9989	60.9988	
152	139.5305	234.5235	0.0	6.2921
153	2917.6143	151.9989	61.9988	
153	139.5305	234.5235	0.0	6.3649
154	2936.8091	152.9989	62.9988	
154	139.5305	234.5235	0.0	6.4357
155	2956.0039	153.9989	63.9988	
155	139.5305	234.5235	0.0	6.5043
156	2975.1987	154.9989	64.9988	
156	139.5305	234.5235	0.0	6.5707
157	2994.3936	155.9989	65.9988	
157	139.5305	234.5235	0.0	6.6350
158	3013.5884	156.9989	66.9988	
158	139.5305	234.5235	0.0	6.6970
159	3032.7832	157.9989	67.9988	
159	139.5305	234.5235	0.0	6.7568
160	3051.9780	158.9989	68.9988	
160	139.5305	234.5235	0.0	6.8143

161	3071.1729	159.9989	69.9988	
161	139.5305	234.5235	0.0	6.8695
162	3090.3677	160.9989	70.9988	
162	139.5305	234.5235	0.0	6.9225
163	3109.5625	161.9989	71.9988	
163	139.5305	234.5235	0.0	6.9732
164	3128.7573	162.9989	72.9988	
164	139.5305	234.5235	0.0	7.0217
165	3147.9521	163.9988	73.9988	
165	139.5305	234.5235	0.0	7.0678
166	3167.1470	164.9988	74.9988	
166	139.5305	234.5235	0.0	7.1118
167	3186.3418	165.9988	75.9988	
167	139.5305	234.5235	0.0	7.1536
168	3205.5366	166.9988	76.9988	
168	139.5305	234.5235	0.0	7.1931
169	3224.7314	167.9988	77.9988	
169	139.5305	234.5235	0.0	7.2304
170	3243.9263	168.9988	78.9988	
170	139.5305	234.5235	0.0	7.2655
171	3263.1211	169.9988	79.9988	
171	139.5305	234.5235	0.0	7.2935
172	3282.3159	170.9988	80.9988	
172	139.5305	234.5235	0.0	7.3294
173	3301.5107	171.9988	81.9988	
173	139.5305	234.5235	0.0	7.3583
174	3320.7056	172.9988	82.9988	
174	139.5305	234.5235	0.0	7.3852
175	3339.9004	173.9988	83.9985	
175	139.5305	234.5235	0.0	7.4101
176	3359.0952	174.9988	84.9985	
176	139.5305	234.5235	0.0	7.4331
177	3378.2900	175.9988	85.9985	
177	139.5305	234.5235	0.0	7.4542
178	3397.4849	176.9987	86.9985	
178	139.5305	234.5235	0.0	7.4735
179	3416.6797	177.9987	87.9985	
179	139.5305	234.5235	0.0	7.4912
180	3435.8745	178.9987	88.9985	
180	139.5305	234.5235	0.0	7.5072

181	3455.0693	179.9987	89.9985	
181	139.5305	234.5235	0.0	7.5216
182	3474.2642	180.9987	90.9985	
182	139.5305	234.5235	0.0	7.5344
183	3493.4590	181.9987	91.9985	
183	139.5305	234.5235	0.0	7.5458
184	3512.6538	182.9987	92.9985	
184	139.5305	234.5235	0.0	7.5558
185	3531.8486	183.9987	93.9985	
185	139.5305	234.5235	0.0	7.5645
186	3551.0435	184.9987	94.9985	
186	139.5305	234.5235	0.0	7.5720
187	3570.2383	185.9987	95.9985	
187	139.5305	234.5235	0.0	7.5784
188	3589.4331	186.9987	96.9985	
188	139.5305	234.5235	0.0	7.5837
189	3608.6279	187.9987	97.9985	
189	139.5305	234.5235	0.0	7.5881
190	3627.8228	188.9987	98.9985	
190	139.5305	234.5235	0.0	7.5916
191	3647.0176	189.9987	99.9985	
191	139.5305	234.5235	0.0	7.5944
192	3666.2124	190.9986	100.9985	
192	139.5305	234.5235	0.0	7.5964
193	3685.4072	191.9986	101.9985	
193	139.5305	234.5235	0.0	7.5979
194	3704.6021	192.9986	102.9985	
194	139.5305	234.5235	0.0	7.5989
195	3723.7969	193.9986	103.9985	
195	139.5305	234.5235	0.0	7.5995
196	3742.9917	194.9986	104.9985	
196	139.5305	234.5235	0.0	7.5998
197	3762.1865	195.9986	105.9985	
197	139.5305	234.5235	0.0	7.6000
198	3781.3813	196.9986	106.9985	
198	139.5305	234.5235	0.0	7.6000
199	3800.5762	197.9986	107.9985	
199	139.5305	234.5235	0.0	7.6000
200	3819.7710	198.9986	108.9985	
200	139.5305	234.5236	0.0	7.6000

201	3838.9658	199.9986	109.9985	
201	139.5305	234.5250	0.0	7.6000
202	3858.1606	200.9986	110.9985	
202	139.5305	234.5568	0.0	7.6000
203	3877.3555	201.9986	111.9985	
203	139.5305	234.5961	0.0	7.6000
204	3896.5503	202.9986	112.9985	
204	139.5305	234.5902	0.0	7.6000
205	3915.7451	203.9986	113.9985	
205	139.5305	234.4985	0.0	7.5999
206	3934.9399	204.9986	114.9985	
206	139.5305	234.2677	0.0	7.5996
207	3954.1348	205.9985	115.9985	
207	139.5305	233.8559	0.0	7.5990
208	3973.3296	206.9985	116.9985	
208	139.5028	233.2100	0.0	7.5980
209	3992.5244	207.[illegible]985	117.9983	
209	139.5316	232.2967	0.0	7.5965
210	4011.7192	208.9985	118.9983	
210	139.5232	231.3046	0.0	7.5944
211	4030.9141	209.9985	119.9983	
211	139.5116	230.3269	0.0	7.5917
212	4050.1089	210.9985	120.9983	
212	139.4915	229.3552	0.0	7.5881
213	4069.3037	211.9985	121.9983	
213	139.4573	22[illegible].3960	0.0	7.5838
214	4088.4985	212.9985	122.9983	
214	139.4059	227.4440	0.0	7.5784
215	4107.6914	213.9984	123.9983	
215	139.3391	226.4972	0.0	7.5720
216	4126.8828	214.9982	124.9980	
216	139.2863	225.5599	0.0	7.5645
217	4146.0742	215.9980	125.9978	
217	139.1228	224.6259	0.0	7.5558
218	4165.2656	216.9978	126.9978	
218	138.9798	223.6995	0.0	7.5458
219	41[illegible]4.4570	217.9976	127.9976	
219	138.8020	222.7788	0.0	7.5344
220	4203.6484	218.9975	128.9973	
220	138.5919	221.8588	0.0	7.5216

221	4222.8398	219.9973	129.9971	
221	138.3427	220.9470	0.0	7.5072
222	4242.0312	220.9971	130.9971	
222	138.0497	220.0384	0.0	7.4912
223	4261.2227	221.9969	131.9968	
223	137.7166	219.1297	0.0	7.4737
224	4280.4141	222.9967	132.9966	
224	137.3385	218.2278	0.0	7.4544
225	4299.6055	223.9965	133.9963	
225	136.9101	217.3279	0.0	7.4332
226	4318.7969	224.9963	134.9963	
226	136.4313	216.4275	0.0	7.4102
227	4337.9883	225.9962	135.9961	
227	135.8969	215.5325	0.0	7.3853
228	4357.1797	226.9960	136.9958	
228	135.3101	214.6384	0.0	7.3584
229	4376.3711	227.9958	137.9956	
229	134.6627	213.7442	0.0	7.3296
230	4395.5625	228.9956	138.9954	
230	133.9579	212.8537	0.0	7.2986
231	4414.7539	229.9954	139.9954	
231	133.1897	211.9629	0.0	7.2657
232	4433.9453	230.9952	140.9951	
232	132.3665	211.0723	0.0	7.2306
233	4453.1367	231.9950	141.9949	
233	131.4797	210.1842	0.0	7.1933
234	4472.3281	232.9949	142.9946	
234	130.5319	209.2955	0.0	7.1538
235	4491.5195	233.9947	143.9946	
235	129.5215	208.4057	0.0	7.1121
236	4510.7109	234.9945	144.9944	
236	128.4409	207.5182	0.0	7.0681
237	4529.9023	235.9943	145.9941	
237	127.3330	206.6296	0.0	7.0321
238	4549.0937	236.9941	146.9939	
238	126.1563	205.7392	0.0	6.9637
239	4568.2852	237.9939	147.9939	
239	124.9257	204.8502	0.0	6.9228
240	4587.4766	238.9937	148.9937	
240	123.6446	203.9603	0.0	6.8700

241	4606.6680	239.9936	149.9934	
241	122.3155	203.0680	0.0	6.8148
242	4625.8594	240.9934	150.9932	
242	120.9423	202.1837	0.0	6.7573
243	4645.0508	241.9932	151.9932	
243	119.5255	201.4463	0.0	6.6975
244	4664.2422	242.9930	152.9929	
244	118.0711	200.8886	0.0	6.6355
245	4683.4336	243.9928	153.9927	
245	116.5822	200.4997	0.0	6.5713
246	4702.6250	244.9926	154.9924	
246	115.0579	200.2543	0.0	6.5049
247	4721.8164	245.9924	155.9924	
247	113.5061	200.1377	0.0	6.4363
248	4741.0078	246.9923	156.9922	
248	111.9236	200.1033	0.0	6.3656
249	4760.1992	247.9921	157.9919	
249	110.3203	200.1064	0.0	6.2928
250	4779.3906	248.9919	158.9917	
250	108.6890	200.1064	0.0	6.2178
251	4798.5820	249.9917	159.9917	
251	107.0389	200.1064	0.0	6.1409
252	4817.7734	250.9915	160.9915	
252	105.3699	200.1064	0.0	6.0620
253	4836.9648	251.9913	161.9912	
253	103.6813	200.1064	0.0	5.9812
254	4856.1562	252.9911	162.9910	
254	101.9794	200.1064	0.0	5.8984
255	4875.3477	253.9910	163.9910	
255	100.2570	200.1064	0.0	5.8139
256	4894.5391	254.9908	164.9907	
256	98.5234	200.1064	0.0	5.7276
257	4913.7305	255.9936	165.9905	
257	96.7759	200.1064	0.0	5.6395
258	4932.9219	256.9902	166.9902	
258	95.0133	200.1064	0.0	5.5500
259	4952.1133	257.9900	167.9900	
259	93.2414	200.1064	0.0	5.4587
260	4971.3047	258.9900	168.9900	
260	91.4528	200.1063	0.0	5.3660

261	4990.4961	259.9897	169.9897	
261	89.6535	200.1284	0.0	5.2720
262	5009.6875	260.9895	170.9895	
262	87.8446	200.2124	0.0	5.1765
263	5028.8789	261.9893	171.9893	
263	86.0213	200.3982	0.0	5.0798
264	5048.0703	262.9893	172.9893	
264	84.1819	200.7399	0.0	4.9820
265	5067.2617	263.9890	173.9890	
265	82.3352	201.3044	0.0	4.8830
266	5086.4531	264.9888	174.9888	
266	80.4689	202.0998	0.0	4.7830
267	5105.6445	265.9885	175.9885	
267	78.5987	202.9447	0.0	4.6822
268	5124.8359	266.9885	176.9885	
268	76.6982	203.7921	0.0	4.5805
269	5144.0273	267.9883	177.9883	
269	74.8002	204.6341	0.0	4.4781
270	5163.2187	268.9880	178.9880	
270	72.8669	205.4664	0.0	4.3750
271	5182.4102	269.9878	179.9878	
271	71.0146	206.2906	0.0	4.2715
272	5201.6016	270.9878	180.9876	
272	69.3373	207.1070	0.0	4.1675
273	5220.7930	271.9875	181.9875	
273	67.8501	207.9187	0.0	4.0631
274	5239.9844	272.9873	182.9873	
274	66.5800	208.5444	0.0	3.9585
275	5259.1758	273.9871	183.9871	
275	65.5714	209.4997	0.0	3.8538
276	5278.3672	274.9871	184.9871	
276	64.8955	210.3047	0.0	3.7490
277	5297.5586	275.9868	185.9868	
277	64.6406	211.0939	0.0	3.6442
278	5316.7500	276.9866	186.9866	
278	64.6398	211.7946	0.0	3.5396
279	5335.9414	277.9863	187.9863	
279	64.6398	212.2542	0.0	3.4352
280	5355.1328	278.9863	188.9863	
280	64.6398	212.5116	0.0	3.3312

281	5374.3242	279.9861	189.9861	
281	64.6398	212.6365	0.0	3.2276
282	5393.5156	280.9858	190.9858	
282	64.6398	212.6724	0.0	3.1246
283	5412.7070	281.9856	191.9856	
283	64.6398	212.6655	0.0	3.0222
284	5431.8984	282.9854	192.9854	
284	64.6398	212.6483	0.0	2.9205
285	5451.0898	283.9854	193.9854	
285	64.6398	212.6467	0.0	2.8196
286	5470.2812	284.9851	194.9851	
286	64.6398	212.6466	0.0	2.7196
287	5489.4727	285.9849	195.9849	
287	64.6398	212.6466	0.0	2.6206
288	5508.6641	286.9846	196.9846	
288	64.6398	212.6466	0.0	2.5227
289	5527.8555	287.9846	197.9846	
289	64.6398	212.6466	0.0	2.4260
290	5547.0469	288.9844	198.9844	
290	64.6398	212.6466	0.0	2.3305
291	5566.2383	289.9841	199.9841	
291	64.6398	212.6466	0.0	2.2364
292	5585.4297	290.9839	200.9839	
292	64.6402	212.6466	0.0	2.1437
293	5604.6211	291.9839	201.9839	
293	64.5996	212.6466	0.0	2.0524
294	5623.8125	292.9836	202.9836	
294	64.5568	212.6466	0.0	1.9628
295	5643.0039	293.9834	203.9834	
295	64.6526	212.6466	0.0	1.8746
296	5662.1953	294.9832	204.9332	
296	65.0370	212.6466	0.0	1.7883
297	5681.3867	295.9832	205.9832	
297	65.9085	212.6466	0.0	1.7037
298	5700.5781	296.9829	206.9829	
298	67.3202	212.6466	0.0	1.6209
299	5719.7695	297.9827	207.9827	
299	69.3806	212.6466	0.0	1.5401
300	5738.9609	298.9824	208.9824	
300	72.1793	212.6469	0.0	1.4611

301	5758.1523	299.9824	209.9824	
301	75.7725	212.6385	0.0	1.3841
302	5777.3437	300.9822	210.9822	
302	80.1422	212.6831	0.0	1.3091
303	5796.5352	301.9819	211.9819	
303	85.1705	212.9945	0.0	1.2363
304	5815.7266	302.9817	212.9817	
304	90.6074	213.8466	0.0	1.1655
305	5834.9180	303.9817	213.9817	
305	96.1208	215.6764	0.0	1.0968
306	5854.1094	304.9814	214.9814	
306	101.3765	218.5575	0.0	1.0304
307	5873.3008	305.9812	215.9812	
307	106.1510	221.7124	0.0	0.9661
308	5892.4922	306.9810	216.9810	
308	110.3696	224.7781	0.0	0.9041
309	5911.6836	307.9810	217.9810	
309	114.0807	227.6341	0.0	0.8442
310	5930.8750	308.9807	218.9807	
310	117.3700	230.2688	0.0	0.7867
311	5950.0664	309.9805	219.9805	
311	120.3149	232.5148	0.0	0.7315
312	5969.2578	310.9802	220.9802	
312	122.9725	233.7642	0.0	0.6784
313	5988.4492	311.9800	221.9800	
313	125.3846	234.3376	0.0	0.6277
314	6007.6406	312.9800	222.9800	
314	127.5831	234.5210	0.0	0.5791
315	6026.8320	313.9797	223.9797	
315	129.5758	234.5263	0.0	0.5330
316	6046.0234	314.9795	224.9795	
316	131.3932	234.5236	0.0	0.4890
317	6065.2148	315.9792	225.9792	
317	133.0281	234.5235	0.0	0.4472
318	6084.4062	316.9792	226.9792	
318	134.4976	234.5235	0.0	0.4077
319	6103.5977	317.9790	227.9790	
319	135.7935	234.5235	0.0	0.3704
320	6122.7891	318.9788	228.9788	
320	136.9149	234.5235	0.0	0.3352

321	6141.9805	319.9785	229.9785	
321	137.8559	234.5235	0.0	0.3022
322	6161.1719	320.9785	230.9785	
322	138.6050	234.5235	0.0	0.2712
323	6180.3633	321.9783	231.9783	
323	139.1650	234.5235	0.0	0.2423
324	6199.5547	322.9780	232.9780	
324	139.4[illegible]17	234.5235	0.0	0.2154
325	6218.7461	323.9778	233.9778	
325	139.5848	234.5235	0.0	0.1904
326	6237.9375	324.9778	234.9778	
326	139.5735	234.5235	0.0	0.1674
327	6257.1289	325.9775	235.9775	
327	139.5335	234.5235	0.0	0.1463
328	6276.3203	326.9773	236.9773	
328	139.5305	234.5235	0.0	0.1268
329	6295.5[illegible]	327.9771	237.9771	
329	139.5305	234.5235	0.0	0.1092
330	6314.7031	328.9771	238.9771	
330	139.5305	234.5235	0.0	0.0931
331	6333.8945	329.9768	239.9768	
331	139.5305	234.5235	0.0	0.0787
332	6353.0859	330.9766	240.9766	
332	139.5305	234.5235	0.0	0.0659
333	6372.2773	331.9763	241.9763	
333	139.5305	234.5235	0.0	0.0545
334	6391.4687	332.9763	242.9763	
334	139.5305	234.5235	0.0	0.0444
335	6410.6602	333.9761	243.9761	
335	139.5305	234.5235	0.0	0.0357
336	6429.8516	334.9758	244.9758	
336	139.5305	234.5235	0.0	0.0282
337	6449.0430	335.9756	245.9756	
337	139.5305	234.5235	0.0	0.0217
338	6468.[illegible]344	336.9753	246.9753	
338	139.5305	234.5235	0.0	0.0163
339	6487.4258	337.9753	247.9753	
339	139.5305	234.5235	0.0	0.0120
340	6506.6172	338.9751	248.9751	
340	139.5305	234.5235	0.0	0.0085

341	6525.8086	339.9749	249.9749	
341	139.5305	234.5235	0.0	0.0057
342	6545.0000	340.9746	250.9746	
342	139.5305	234.5235	0.0	0.0036
343	6564.1914	341.9746	251.9746	
343	139.5305	234.5235	0.0	0.0021
344	6583.3828	342.9744	252.9744	
344	139.5305	234.5235	0.0001	0.0011
345	6602.5742	343.9741	253.9741	
345	139.5305	234.5235	-0.0003	0.0005
346	6621.7656	344.9739	254.9739	
346	139.5305	234.5235	0.0414	0.0002
347	6640.9570	345.9739	255.9739	
347	139.5305	234.5235	0.2038	0.0000
348	6660.1484	346.9736	256.9736	
348	139.5305	234.5235	0.5401	-0.0000
349	6679.3398	347.9734	257.9734	
349	139.5305	234.5235	1.1025	0.0000
350	6698.5312	348.9731	258.9731	
350	139.5305	234.5235	1.9437	0.0
351	6717.7227	349.9731	259.9731	
351	139.5305	234.5235	3.1164	0.0
352	6736.9141	350.9729	260.9729	
352	139.5305	234.5235	4.6724	0.0
353	6756.1055	351.9727	261.9727	
353	139.5305	234.5235	6.6673	0.0
354	6775.2969	352.9724	262.9724	
354	139.5305	234.5235	8.9048	0.0
355	6794.4883	353.9724	263.9724	
355	139.5305	234.5235	11.1576	0.0
356	6813.6797	354.9722	264.9722	
356	139.5305	234.5235	12.5313	0.0
357	6832.8711	355.9719	265.9719	
357	139.5305	234.5235	13.1030	0.0
358	6852.0625	356.9717	266.9717	
358	139.5305	234.5235	13.4463	0.0
359	6871.2539	357.9717	267.9717	
359	139.5305	234.5235	13.5824	0.0
360	6890.4453	358.9714	268.9714	
360	139.5305	234.5235	14.0092	0.0

361	6909.6367	359.9712	269.9712	
361	139.5305	234.5235	14.3732	0.0
362	6928.8281	0.9709	270.9709	
362	139.5305	234.5235	14.6046	0.0
363	6948.0195	1.9709	271.9709	
363	139.5305	234.5235	14.7334	0.0
364	6967.2109	2.9707	272.9707	
364	139.5305	234.5235	14.7892	0.0
365	6986.4023	3.9705	273.9705	
365	139.5305	234.5235	14.8010	0.0
366	7005.5937	4.9702	274.9702	
366	139.5305	234.5235	14.7983	0.0
367	7024.7852	5.9700	275.9700	
367	139.5305	234.5235	14.7982	0.0
368	7043.9766	6.9700	276.9700	
368	139.5305	234.5235	14.7982	0.0
369	7063.1680	7.9697	277.9697	
369	139.5305	234.5235	14.7982	0.0
370	7082.3594	8.9695	278.9695	
370	139.5305	234.5235	14.7982	0.0
371	7101.5500	9.9692	279.9692	
371	139.5305	234.5235	14.7982	0.0
372	7120.7422	10.9692	280.9692	
372	139.5305	234.5235	14.7982	0.0
373	7139.9336	11.9690	281.9690	
373	139.5305	234.5235	14.7982	0.0
374	7159.1250	12.9688	282.9687	
374	139.5305	234.5235	14.7982	0.0
375	7178.3164	13.9685	283.9685	
375	139.5305	234.5235	14.7982	0.0
376	7197.5078	14.9685	284.9685	
376	139.5305	234.5235	14.7982	0.0
377	7216.6992	15.9683	285.9683	
377	139.5305	234.5235	14.7982	0.0
378	7235.8906	16.9680	286.9680	
378	139.5305	234.5235	14.7982	0.0
379	7255.0820	17.9678	287.9678	
379	139.5305	234.5235	14.7982	0.0
380	7274.2734	18.9678	288.9678	
380	139.5305	234.5235	14.7982	0.0

A P P E N D I X A-4

PROGRAM LISTING, DYNAMIC MODEL

```
              C
 0001                DIMENSION DUM2(6),DUM3(6),DUM4(6),DUM5(6),DUM6(6),DUM7(6)
              C
              C---------- THIS IS THE MAIN PROGRAM EXECUTIVE
              C
 0002                CALL INPUT
              C
 0003                CALL DATAIN
              C
 0004                CALL CMPUTE
              C
 0005                CALL TBLOUT(10,DUM2,DUM3,DUM4,DUM5,DUM6,DUM7,X1,X2,X3)
              C
 0006                STOP
 0007                END
```

```
0001          SUBROUTINE INPUT                                                 00000160
        C                                                                      00000170
        C---------- ABSTRACT: SUBROUTINE INPUT PREFORMS AS THE CALLING ROUTINE F00000180
        C----------       THE WEAPON INIT                                      00000190
        C                                                                      00000200
        C                                                                      00000210
        C                                                                      00000220
0002          COMMON/TIMES/TINIT,TFINAL,TNOW,TOUT,TSTEP,TNEXT                  00000230
0003          COMMON/ANGLE2/TH2,DTH2,D2TH2                                     00000240
0004          COMMON/SWITCH/IACT                                               00000250
        C   VARIABLE      TYPE       DESCRIPTION                               00000260
        C                                                                      00000270
        C---- COMMON TIMES                                                     00000280
        C TINIT          REAL    INITIAL TIME FOR INTEGRATION (SECONDS)        00000290
        C TFINAL         REAL    FINAL AND TERMINATING TIME FOR INTEGRATION (SECON00000300
        C TNOW           REAL    TIME TO WHICH THE INTEGRATION HAS PROGRESSED (SEC00000310
        C TOUT           REAL    TIME STEP THAT MUST BE EXCEEDED TO GET AN OUTPUT 00000320
        C TSTEP          REAL    INITIAL TIME STEP SIZE TO START INTEGRATION (SEC00000330
        C TNEXT          REAL    ABSOLUTE TIME FOR NEXT OUTPUT STEP. TNEXT2=TNEXT100000340
        C---- COMMON ANGLE2                                                    00000350
        C TH2XTL         REAL    ROTATION OF INPUT MOTOR GEAR (DEGREES)        00000360
        C DTH2           REAL    FIRST TIME DERIVITIVE OF TH2 (DEG/SEC)        00000370
        C D2TH2          REAL    SECOND TIME DERIVATIVE OF TH2 (DEG/SEC**2)    00000380
        C---- COMMON SWITCH                                                    00000390
        C IACT           INTEGER INTERACTIVE INPUT SWITCH. 2 IS INTERACTIVE, 0 IS 00000400
        C                                                                      00000410
        C----------------------------------------------------------------------00000420
        C---------- ASK USER IF HE WANTS DEFAULT VALUES OF WISHES TO INPUT INTER00000430
        C---------- SET INTERACTIVE SWITCH (IACT) ACCORDINGLY                  00000440
        C                                                                      00000450
        C---------- FOR AVSCOM, FORGET ABOUT INTERACTIVE WORK                  00000460
        C                                                                      00000470
        C                                                                      00000480
        C                                                                      00000490
        C---------- WHEN HERE, THE USER WANTS DEFAULT VALUES                   00000500
        C                                                                      00000510
0005          TH2=0.                                                           00000520
0006          DTH2=0.                                                          00000530
0007          D2TH2=0.                                                         00000540
0008          TINIT=0.                                                         00000550
0009          TFINAL=2.5                                                       00000560
0010          TNOW=0.                                                          00000570
0011          TOUT=.01                                                         00000580
0012          TSTEP=.01                                                        00000590
0013    9000  CONTINUE                                                         00000600
0014          TNEXT=0.0                                                        00000610
        C                                                                      00000620
        C---------- UPDATE ALL THE OTHER ANGLE COMMONS AND DISPL COMMON        00000630
        C                                                                      00000640
0015          CALL UPDATE                                                      00000650
        C                                                                      00000660
        C---------- INITIALIZATION IS COMPLETED                                00000670
        C                                                                      00000680
```

```
 0016              RETURN                                                  00000690
 0017        1000  FORMAT(' ENTER 1 IF YOU WISH DEFALUT VALUES',/,          00000700
                 2      'ENTER 2 IF YOU WISH TO SPECIFY INITIAL CONDITIONS',/,  00000710
                 3      'THEN HIT RETURN')                                  00000720
 0018        1001  FORMAT(I1)                                               00000730
 0019              END                                                      00000740
```

```
0001             SUBROUTINE DATAIN                                                    00000750
           C                                                                          00000760
           C---------- ABSTRACT: SUBROUTINE DATAIN SETS UP THE                        00000770
           C----------      SEVEN TABULAR FUNCTIONS NECESSARY TO                      00000780
           C----------      ESTABLISH THE VARIOUS DEPENDENCE ON THE                   00000790
           C----------      MOTOR INPUT ANGLE (TH2).  THE DATA IS                     00000800
           C----------      AVAILABLE ON UNIT 1 AND IS THE RESULT                     00000810
           C----------      OF THE EXECUTION OF THE AMCAWS-30                         00000820
           C----------      TABULAR FUNCTION GENERATING PROGRAM                       00000830
           C                                                                          00000840
           C                                                                          00000850
0002             COMMON/MOTOR2/AMTH2(1000),NPTS2                                      00000860
0003             COMMON/DRUM3/DTH3(1000)                                              00000870
0004             COMMON/FACE3F/FTH3F(1000)                                            00000880
0005             COMMON/FEED4/FTH4(1000)                                              00000890
0006             COMMON/EJECT5/ETH5(1000)                                             00000900
0007             COMMON/LOCK6/ALTH6(1000)                                             00000910
0008             COMMON/DISP7/RAD7(1000)                                              00000920
0009             COMMON/MTORQ/RPM(1000),TRPM(1000),NPTTQ                              00000930
           C--------------------------------------------------------------------------00000940
           C   VARIABLE      TYPE       DESCRIPTION                                   00000950
           C                                                                          00000960
           C---- COMMON MOTOR2                                                        00000970
           C AMTH2(1000)  REAL     VECTOR OF SUCESSIVE INPUT MOTOR POSITIONS (DEG)    00000980
           C NPTS2        INTEGER POINTS IN AMTH2,DRUM3,FACE3F,FEED4,EJECT5,LOCK6,D000000990
           C---- COMMON DRUM3                                                         00001000
           C DTH3(1000)   REAL     VECTOR OF DRUM  POSITIONS FOR CORRESPONDING AMTH200001010
           C---- COMMON FACE3F                                                        00001020
           C FTH3F(1000)  REAL     VECTOR OF FACE  POSITIONS FOR CORRESPONDING AMTH200001030
           C---- COMMON FEED4                                                         00001040
           C FTH4         REAL     VECTOR OF FEED POSITIONS FOR CORRESPONDING AMTH2   00001050
           C---- COMMON EJECT5                                                        00001060
           C ETH5(1000)   REAL     VECTOR OF EJECT POSITIONS FOR CORRESPONDING AMTH200001070
           C---- COMMON LOCK6                                                         00001080
           C ALTH6(1000)  REAL     VECTOR OF LOCKT POSITIONS FOR CORRESPONDING AMTH200001090
           C---- COMMON DISP7                                                         00001100
           C RAD7(1000)   REAL     VECTOR OF CHAMBER DISPLACEMENTS COR. TO AMTH2 (FT00001110
           C---- COMMON MTORQ                                                         00001120
           C RPM(1000)    REAL     VECTOR OF RPM VALUES FOR INPUT MOTOR (DEG/SEC)     00001130
           C TRPM(1000)   REAL     VECTOR OF TORQUES CORRES. TO RPM VECTOR (FT-LBS)   00001140
           C NPTTQ        INTEGER POINTS IN RPM-TRPM TORQUE/SPEED CURVE VECTORS       00001150
           C--------------------------------------------------------------------------00001160
           C---------- FILL THE COMMON AREAS FROM DATA                                00001170
           C---------- FROM UNIT 1 DEVICE                                             00001180
           C                                                                          00001190
0010       100   CONTINUE                                                             00001200
0011             READ(1,1000,END=200,ERR=900)J,AMTH2(J),DTH3(J),FTH3F(J)              00001210
0012             READ(1,1000,END=200,ERR=900)J,FTH4(J),ETH5(J),ALTH6(J),RAD7(J)       00001220
0013                   RAD7(J)=RAD7(J)/12.0                                           00001230
0014                      WRITE(6,2001) J,AMTH2(J),DTH3(J),FTH3F(J)                   00001240
0015                      WRITE(6,2001) J,FTH4(J),ETH5(J),ALTH6(J),RAD7(J)            00001250
0016       2001           FORMAT(' ',I5,4F15.4)                                       00001260
0017       1000  FORMAT(I5,4F16.4)                                                    00001270
```

```
 0018              GO TO 100                                                       00001280
             C                                                                     00001290
 0019        200   CONTINUE                                                        00001300
 0020                 NPTS2=J                                                      00001310
 0021        300   CONTINUE                                                        00001320
 0022                 READ(2,1000,END=400,ERR=900) J,RPM(J),TRPM(J)                00001330
 0023                       TRPM(J)=TRPM(J)/12.0                                   00001340
 0024                       WRITE(6,2001)J,RPM(J),TRPM(J)                          00001350
 0025                 GO TO 300                                                    00001360
             C                                                                     00001370
 0026        400   CONTINUE                                                        00001380
 0027                 NPTTQ=J                                                      00001390
             C                                                                     00001400
             C                                                                     00001410
             C---------- DATA IS NOW IN PROPER COMMONS                             00001420
             C                                                                     00001430
             C---------- ERROR HANDLER IF NECESSARY                                00001440
             C                                                                     00001450
 0028              GO TO 9999                                                      00001460
 0029        900   CONTINUE                                                        00001470
 0030                 WRITE(6,1001) J                                              00001480
 0031        9999  CONTINUE                                                        00001490
             C                                                                     00001500
             C                                                                     00001510
 0032              RETURN                                                          00001520
 0033        1001  FORMAT(' UNIT READ ERROR, UNIT 1 OR 2, J=',I5)                  00001530
 0034              END                                                             00001540
```

```
0001          SUBROUTINE CMPUTE
        C
        C---------- ABSTRACT:  SUBROUTINE CMPUTE HANDLES THE NUMERICAL
        C----------      INTEGRATION BY A CALL TO THE IBM-SSP SUBROUTINE
        C----------      HPCG (HAMMING PREDICTOR-CORRECTOR, MODIFIED)
        C----------      CMPUTE SETS UP THE PARAMETERS USED IN THE CALLING
        C----------      ARRAYS AND THEN HANDS CONTROL OF THE PROGRAM TO
        C----------      HPCG.  HPCG REQUIRES TWO PROGRAMMER SUPPLIED
        C----------      ROUTINES CALLED FCT AND OUTP
        C
0002          COMMON/TIMES/TINIT,TFINAL,TNOW,TOUT,TSTEP,TNEXT
0003          COMMON/ANGLE2/TH2,DTH2,D2TH2
0004          DIMENSION Y(2),DERY(2),PRMT(5),AUX(16,4)
0005          EXTERNAL FCT,OUTP
        C-----------------------------------------------------------------
        C   VARIABLE     TYPE      DESCRIPTION
        C
        C---- COMMON TIMES
        C TINIT         REAL    INITIAL TIME FOR INTEGRATION (SECONDS)
        C TFINAL        REAL    FINAL AND TERMINATING TIME FOR INTEGRATION (SECON
        C TNOW          REAL    TIME TO WHICH THE INTEGRATION HAS PROGRESSED (SEC
        C TOUT          REAL    TIME STEP THAT MUST BE EXCEEDED TO GET AN OUTPUT
        C TSTEP         REAL    INITIAL TIME STEP SIZE TO START INTEGRATION (SEC
        C TNEXT         REAL    ABSOLUTE TIME FOR NEXT OUTPUT STEP. TNEXT2=TNEXT1
        C---- COMMON ANGLE2
        C TH2           REAL    ROTATION OF INPUT MOTOR GEAR (DEGREES)
        C DTH2          REAL    FIRST TIME DERIVITIVE OF TH2 (DEG/SEC)
        C D2TH2         REAL    SECOND TIME DERIVATIVE OF TH2 (DEG/SEC**2)
        C
        C Y(1)          REAL    INITIAL ANGLE OF INPUT MOTOR (DEGREES)
        C Y(2)          REAL    INITIAL ANGULAR VELOCITY OF INPUT MOTOR  (DEG/SEC
        C DERY(1)       REAL    ERROR WEIGHT FACTOR FOR INTEGRATING ROUTINE
        C DERY(2)       REAL    ERROR WEIGHT FACTOR FOR INTEGRATING ROUTINE
        C PRMT(1)       REAL    INITIAL TIME SUPPLIED TO HPCG ROUTINE, TINIT (SEC
        C PRMT(2)       REAL    FINAL TIME GIVEN TO HPCG, =TFINAL (SECONDS)
        C PRMT(3)       REAL    OUTPUT OCCURS EVERY PRMT(3) SECONDS
        C PRMT(4)       REAL    ABSOLUTE ERROR UPON WHICH INTEGRATING TIME STEP 10
        C PRMT(5)       REAL    UNUSED HERE.  CAN BE USED OUT OF OUTP ROUTINE
        C AUX(16,4)     REAL    SCRATCH VECTOR USED IN HPCG
        C NDIM          INTEGER NUMBER OF ELEMENTS IN DERY VECTOR
        C-----------------------------------------------------------------
        C
        C---------- SET UP THE PRMT VECTOR, PRMT(5) IS SET INTERNALLY
        C
0006          PRMT(1)=TINIT
0007          PRMT(2)=TFINAL
0008          PRMT(3)=TSTEP
0009          PRMT(4)=40.
        C
        C---------- SET UP THE WEIGHTED DERY VECTOR, NOTE THAT THE
        C---------- SUM OF THE DERY ELEMENTS MUST BE 1.
        C
0010          DERY(1)=.5
```

```
 0011         DERY(2)=.5
           C
           C---------- SET UP THE Y VECTOR WITH THE ANGLE2 INITIAL
           C---------- CONDITIONS
           C
 0012         Y(1)=TH2
 0013         Y(2)=DTH2
           C
           C---------- DIMENSION OF THE PROBLEM IS 2
           C
 0014         NDIM=2
           C
           C---------- ARGUMENT LIST IS NOW COMPLETE. GIVE CONTROL
           C---------- OF THE NUMERICAL INTEGRATION PROCEDURE TO
           C---------- IBM-SSP SUBROUTINE HPCG
           C
 0015            CALL HPCG(PRMT,Y,DERY,NDIM,IHLF,FCT,OUTP,AUX)
           C
           C
 0016            RETURN
 0017            END
```

```
C      SUBROUTINE HPCG(PRMT,Y,DERY,NDIM,IHLF,FCT,OUTP,AUX)
C
C         PURPOSE
C            TO SOLVE A SYSTEM OF FIRST ORDER ORDINARY GENERAL
C            DIFFERENTIAL EQUATIONS WITH GIVEN INITIAL VALUES.
C
C         USAGE
C            CALL HPCG (PRMT,Y,DERY,NDIM,IHLF,FCT,OUTP,AUX)
C            PARAMETERS FCT AND OUTP REQUIRE AN EXTERNAL STATEMENT.
C
C         DESCRIPTION OF PARAMETERS
C            PRMT   - AN INPUT AND OUTPUT VECTOR WITH DIMENSION GREATER
C                     OR EQUAL TO 5, WHICH SPECIFIES THE PARAMETERS OF
C                     THE INTERVAL AND OF ACCURACY AND WHICH SERVES FOR
C                     COMMUNICATION BETWEEN OUTPUT SUBROUTINE (FURNISHED
C                     BY THE USER) AND SUBROUTINE HPCG. EXCEPT PRMT(5)
C                     THE COMPONENTS ARE NOT DESTROYED BY SUBROUTINE
C                     HPCG AND THEY ARE
C            PRMT(1)- LOWER BOUND OF THE INTERVAL (INPUT),
C            PRMT(2)- UPPER BOUND OF THE INTERVAL (INPUT),
C            PRMT(3)- INITIAL INCREMENT OF THE INDEPENDENT VARIABLE
C                     (INPUT),
C            PRMT(4)- UPPER ERROR BOUND (INPUT). IF ABSOLUTE ERROR IS
C                     GREATER THAN PRMT(4), INCREMENT GETS HALVED.
C                     IF INCREMENT IS LESS THAN PRMT(3) AND ABSOLUTE
C                     ERROR LESS THAN PRMT(4)/50, INCREMENT GETS DOUBLED.
C                     THE USER MAY CHANGE PRMT(4) BY MEANS OF HIS
C                     OUTPUT SUBROUTINE.
C            PRMT(5)- NO INPUT PARAMETER. SUBROUTINE HPCG INITIALIZES
C                     PRMT(5)=0. IF THE USER WANTS TO TERMINATE
C                     SUBROUTINE HPCG AT ANY OUTPUT POINT, HE HAS TO
C                     CHANGE PRMT(5) TO NON-ZERO BY MEANS OF SUBROUTINE
C                     OUTP. FURTHER COMPONENTS OF VECTOR PRMT ARE
C                     FEASIBLE IF ITS DIMENSION IS DEFINED GREATER
C                     THAN 5. HOWEVER SUBROUTINE HPCG DOES NOT REQUIRE
C                     AND CHANGE THEM. NEVERTHELESS THEY MAY BE USEFUL
C                     FOR HANDING RESULT VALUES TO THE MAIN PROGRAM
C                     (CALLING HPCG) WHICH ARE OBTAINED BY SPECIAL
C                     MANIPULATIONS WITH OUTPUT DATA IN SUBROUTINE OUTP.
C            Y      - INPUT VECTOR OF INITIAL VALUES.  (DESTROYED)
C                     LATERON Y IS THE RESULTING VECTOR OF DEPENDENT
C                     VARIABLES COMPUTED AT INTERMEDIATE POINTS X.
C            DERY   - INPUT VECTOR OF ERROR WEIGHTS.  (DESTROYED)
C                     THE SUM OF ITS COMPONENTS MUST BE EQUAL TO 1.
C                     LATERON DERY IS THE VECTOR OF DERIVATIVES, WHICH
C                     BELONG TO FUNCTION VALUES Y AT A POINT X.
C            NDIM   - AN INPUT VALUE, WHICH SPECIFIES THE NUMBER OF
C                     EQUATIONS IN THE SYSTEM.
C            IHLF   - AN OUTPUT VALUE, WHICH SPECIFIES THE NUMBER OF
C                     BISECTIONS OF THE INITIAL INCREMENT. IF IHLF GETS
C                     GREATER THAN 10, SUBROUTINE HPCG RETURNS WITH
C                     ERROR MESSAGE IHLF=11 INTO MAIN PROGRAM.
C                     ERROR MESSAGE IHLF=12 OR IHLF=13 APPEARS IN CASE
```

```
      C                PRMT(3)=0 OR IN CASE SIGN(PRMT(3)).NE.SIGN(PRMT(2)-00002820
      C                PRMT(1)) RESPECTIVELY.                             00002830
      C          FCT   - THE NAME OF AN EXTERNAL SUBROUTINE USED. IT      00002840
      C                COMPUTES THE RIGHT HAND SIDES DERY OF THE SYSTEM   00002850
      C                TO GIVEN VALUES OF X AND Y. ITS PARAMETER LIST     00002860
      C                MUST BE X,Y,DERY. THE SUBROUTINE SHOULD NOT        00002870
      C                DESTROY X AND Y.                                   00002880
      C          OUTP  - THE NAME OF AN EXTERNAL OUTPUT SUBROUTINE USED.  00002890
      C                ITS PARAMETER LIST MUST BE X,Y,DERY,IHLF,NDIM,PRMT.00002900
      C                NONE OF THESE PARAMETERS (EXCEPT, IF NECESSARY,    00002910
      C                PRMT(4),PRMT(5),...) SHOULD BE CHANGED BY          00002920
      C                SUBROUTINE OUTP. IF PRMT(5) IS CHANGED TO NON-ZERO,00002930
      C                SUBROUTINE HPCG IS TERMINATED.                     00002940
      C          AUX   - AN AUXILIARY STORAGE ARRAY WITH 16 ROWS AND NDIM 00002950
      C                COLUMNS.                                           00002960
      C                                                                   00002970
      C       REMARKS                                                     00002980
      C          THE PROCEDURE TERMINATES AND RETURNS TO CALLING PROGRAM, IF 00002990
      C          (1) MORE THAN 10 BISECTIONS OF THE INITIAL INCREMENT ARE 00003000
      C              NECESSARY TO GET SATISFACTORY ACCURACY (ERROR MESSAGE 00003010
      C              IHLF=11),                                            00003020
      C          (2) INITIAL INCREMENT IS EQUAL TO 0 OR HAS WRONG SIGN    00003030
      C              (ERROR MESSAGES IHLF=12 OR IHLF=13),                 00003040
      C          (3) THE WHOLE INTEGRATION INTERVAL IS WORKED THROUGH,    00003050
      C          (4) SUBROUTINE OUTP HAS CHANGED PRMT(5) TO NON-ZERO.     00003060
      C                                                                   00003070
      C       SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED               00003080
      C          THE EXTERNAL SUBROUTINES FCT(X,Y,DERY) AND               00003090
      C          OUTP(X,Y,DERY,IHLF,NDIM,PRMT) MUST BE FURNISHED BY THE USER.00003100
      C                                                                   00003110
      C       METHOD                                                      00003120
      C          EVALUATION IS DONE BY MEANS OF HAMMINGS MODIFIED PREDICTOR- 00003130
      C          CORRECTOR METHOD. IT IS A FOURTH ORDER METHOD, USING 4   00003140
      C          PRECEEDING POINTS FOR COMPUTATION OF A NEW VECTOR Y OF THE 00003150
      C          DEPENDENT VARIABLES.                                     00003160
      C          FOURTH ORDER RUNGE-KUTTA METHOD SUGGESTED BY RALSTON IS  00003170
      C          USED FOR ADJUSTMENT OF THE INITIAL INCREMENT AND FOR     00003180
      C          COMPUTATION OF STARTING VALUES.                          00003190
      C          SUBROUTINE HPCG AUTOMATICALLY ADJUSTS THE INCREMENT DURING 00003200
      C          THE WHOLE COMPUTATION BY HALVING OR DOUBLING.            00003210
      C          TO GET FULL FLEXIBILITY IN OUTPUT, AN OUTPUT SUBROUTINE  00003220
      C          MUST BE CODED BY THE USER.                               00003230
      C          FOR REFERENCE, SEE                                       00003240
      C          (1)  RALSTON/WILF, MATHEMATICAL METHODS FOR DIGITAL      00003250
      C               COMPUTERS, WILEY, NEW YORK/LONDON, 1960, PP.95-109. 00003260
      C          (2)  RALSTON, RUNGE-KUTTA METHODS WITH MINIMUM ERROR BOUNDS,00003270
      C               MTAC, VOL.16, ISS.80 (1962), PP.431-437.            00003280
      C                                                                   00003290
      C       ............................................................00003300
      C                                                                   00003310
0001          SUBROUTINE HPCG(PRMT,Y,DERY,NDIM,IHLF,FCT,OUTP,AUX)         00003320
      C                                                                   00003330
      C                                                                   00003340
```

```
0002           DIMENSION PRMT(1),Y(1),DERY(1),AUX(16,1)
0003           N=1
0004           IHLF=0
0005           X=PRMT(1)
0006           H=PRMT(3)
0007           PRMT(5)=0.
0008           DO 1 I=1,NDIM
0009           AUX(16,I)=0.
0010           AUX(15,I)=DERY(I)
0011         1 AUX(1,I)=Y(I)
0012           IF(H*(PRMT(2)-X))3,2,4
         C
         C     ERROR RETURNS
0013         2 IHLF=12
0014           GOTO 4
0015         3 IHLF=13
         C
         C     COMPUTATION OF DERY FOR STARTING VALUES
0016         4 CALL FCT(X,Y,DERY)
         C
         C     RECORDING OF STARTING VALUES
0017           CALL OUTP(X,Y,DERY,IHLF,NDIM,PRMT)
0018           IF(PRMT(5))6,5,6
0019         5 IF(IHLF)7,7,6
0020         6 RETURN
0021         7 DO 8 I=1,NDIM
0022         8 AUX(8,I)=DERY(I)
         C
         C     COMPUTATION OF AUX(2,I)
0023           ISW=1
0024           GOTO 100
         C
0025         9 X=X+H
0026           DO 10 I=1,NDIM
0027        10 AUX(2,I)=Y(I)
         C
         C     INCREMENT H IS TESTED BY MEANS OF BISECTION
0028        11 IHLF=IHLF+1
0029           X=X-H
0030           DO 12 I=1,NDIM
0031        12 AUX(4,I)=AUX(2,I)
0032           H=.5*H
0033           N=1
0034           ISW=2
0035           GOTO 100
         C
0036        13 X=X+H
0037           CALL FCT(X,Y,DERY)
0038           N=2
0039           DO 14 I=1,NDIM
0040           AUX(2,I)=Y(I)
0041        14 AUX(9,I)=DERY(I)
0042           ISW=3
```

```
0043           GOTO 100
         C
         C     COMPUTATION OF TEST VALUE DELT
0044        15 DELT=0.
0045           DO 16 I=1,NDIM
0046        16 DELT=DELT+AUX(15,I)*ABS(Y(I)-AUX(4,I))
0047           DELT=.06666667*DELT
0048           IF(DELT-PRMT(4))19,19,17
0049        17 IF(IHLF-10)11,18,18
         C
         C     NO SATISFACTORY ACCURACY AFTER 10 BISECTIONS. ERROR MESSAGE.
0050        18 IHLF=11
0051           X=X+H
0052           GOTO 4
         C
         C     THERE IS SATISFACTORY ACCURACY AFTER LESS THAN 11 BISECTIONS.
0053        19 X=X+H
0054           CALL FCT(X,Y,DERY)
0055           DO 20 I=1,NDIM
0056           AUX(3,I)=Y(I)
0057        20 AUX(10,I)=DERY(I)
0058           N=3
0059           ISW=4
0060           GOTO 100
         C
0061        21 N=1
0062           X=X+H
0063           CALL FCT(X,Y,DERY)
0064           X=PRMT(1)
0065           DO 22 I=1,NDIM
0066           AUX(11,I)=DERY(I)
0067        22 Y(I)=AUX(1,I)+H*(.375*AUX(8,I)+.7916667*AUX(9,I)
              1-.2083333*AUX(10,I)+.04166667*DERY(I))
0068        23 X=X+H
0069           N=N+1
0070           CALL FCT(X,Y,DERY)
0071           CALL OUTP(X,Y,DERY,IHLF,NDIM,PRMT)
0072           IF(PRMT(5))6,24,6
0073        24 IF(N-4)25,200,200
0074        25 DO 26 I=1,NDIM
0075           AUX(N,I)=Y(I)
0076        26 AUX(N+7,I)=DERY(I)
0077           IF(N-3)27,29,200
         C
0078        27 DO 28 I=1,NDIM
0079           DELT=AUX(9,I)+AUX(9,I)
0080           DELT=DELT+DELT
0081        28 Y(I)=AUX(1,I)+.3333333*H*(AUX(8,I)+DELT+AUX(10,I))
0082           GOTO 23
         C
0083        29 DO 30 I=1,NDIM
0084           DELT=AUX(9,I)+AUX(10,I)
0085           DELT=DELT+DELT+DELT
```

```
0086          30 Y(I)=AUX(1,I)+.375*H*(AUX(8,I)+DELT*AUX(11,I))
0087             GOTO 23
            C
            C    THE FOLLOWING PART OF SUBROUTINE HPCG COMPUTES BY MEANS OF
            C    RUNGE-KUTTA METHOD STARTING VALUES FOR THE NOT SELF-STARTING
            C    PREDICTOR-CORRECTOR METHOD.
0088         100 DO 101 I=1,NDIM
0089             Z=H*AUX(N+7,I)
0090             AUX(5,I)=Z
0091         101 Y(I)=AUX(N,I)+.4*Z
            C    Z IS AN AUXILIARY STORAGE LOCATION
            C
0092             Z=X+.4*H
0093             CALL FCT(Z,Y,DERY)
0094             DO 102 I=1,NDIM
0095             Z=H*DERY(I)
0096             AUX(6,I)=Z
0097         102 Y(I)=AUX(N,I)+.2969776*AUX(5,I)+.1587596*Z
            C
0098             Z=X+.4557372*H
0099             CALL FCT(Z,Y,DERY)
0100             DO 103 I=1,NDIM
0101             Z=H*DERY(I)
0102             AUX(7,I)=Z
0103         103 Y(I)=AUX(N,I)+.2181004*AUX(5,I)-3.050965*AUX(6,I)+3.832865*Z
            C
0104             Z=X+H
0105             CALL FCT(Z,Y,DERY)
0106             DO 104 I=1,NDIM
0107         1040Y(I)=AUX(N,I)+.1747603*AUX(5,I)-.5514807*AUX(6,I)
                1+1.205535*AUX(7,I)+.1711848*H*DERY(I)
0108             GOTO(9,13,15,21),ISW
            C
            C    POSSIBLE BREAK-POINT FOR LINKAGE
            C
            C    STARTING VALUES ARE COMPUTED.
            C    NOW START HAMMINGS MODIFIED PREDICTOR-CORRECTOR METHOD.
0109         200 ISTEP=3
0110         201 IF(N-8)204,202,204
            C
            C    N=8 CAUSES THE ROWS OF AUX TO CHANGE THEIR STORAGE LOCATIONS
0111         202 DO 203 N=2,7
0112             DO 203 I=1,NDIM
0113             AUX(N-1,I)=AUX(N,I)
0114         203 AUX(N+6,I)=AUX(N+7,I)
0115             N=7
            C
            C    N LESS THAN 8 CAUSES N+1 TO GET N
0116         204 N=N+1
            C
            C    COMPUTATION OF NEXT VECTOR Y
0117             DO 205 I=1,NDIM
0118             AUX(N-1,I)=Y(I)
```

```
0119      205 AUX(N+6,I)=DERY(I)                                              00004940
0120          X=X+H                                                           00004950
0121      206 ISTEP=ISTEP+1                                                   00004960
0122          DO 207 I=1,NDIM                                                 00004970
0123         0DELT=AUX(N-4,I)+1.333333*H*(AUX(N+6,I)+AUX(N+6,I)-AUX(N+5,I)+   00004980
             1AUX(N+4,I)+AUX(N+4,I))                                          00004990
0124          Y(I)=DELT-.9256198*AUX(15,I)                                    00005000
0125      207 AUX(16,I)=DELT                                                  00005010
        C     PREDICTOR IS NOW GENERATED IN ROW 16 OF AUX, MODIFIED PREDICTOR 00005020
        C     IS GENERATED IN Y. DELT MEANS AN AUXILIARY STORAGE.             00005030
        C                                                                     00005040
0126          CALL FCT(X,Y,DERY)                                              00005050
        C     DERIVATIVE OF MODIFIED PREDICTOR IS GENERATED IN DERY           00005060
        C                                                                     00005070
0127          DO 208 I=1,NDIM                                                 00005080
0128         0DELT=.125*(9.*AUX(N-1,I)-AUX(N-3,I)+3.*H*(DERY(I)+AUX(N+6,I)+   00005090
             1AUX(N+6,I)-AUX(N+5,I)))                                         00005100
0129          AUX(16,I)=AUX(16,I)-DELT                                        00005110
0130      208 Y(I)=DELT+.07438017*AUX(16,I)                                   00005120
        C                                                                     00005130
        C     TEST WHETHER H MUST BE HALVED OR DOUBLED                        00005140
0131          DELT=0.                                                         00005150
0132          DO 209 I=1,NDIM                                                 00005160
0133      209 DELT=DELT+AUX(15,I)*ABS(AUX(16,I))                              00005170
0134          IF(DELT-PRMT(4))210,222,222                                     00005180
        C                                                                     00005190
        C     H MUST NOT BE HALVED. THAT MEANS Y(I) ARE GOOD.                 00005200
0135      210 CALL FCT(X,Y,DERY)                                              00005210
0136          CALL OUTP(X,Y,DERY,IHLF,NDIM,PRMT)                              00005220
0137          IF(PRMT(5))212,211,212                                          00005230
0138      211 IF(IHLF-11)213,212,212                                          00005240
0139      212 RETURN                                                          00005250
0140      213 IF(H*(X-PRMT(2)))214,212,212                                    00005260
0141      214 IF(ABS(X-PRMT(2))-.1*ABS(H))212,215,215                         00005270
0142      215 IF(DELT-.02*PRMT(4))216,216,201                                 00005280
        C                                                                     00005290
        C                                                                     00005300
        C     H COULD BE DOUBLED IF ALL NECESSARY PRECEEDING VALUES ARE       00005310
        C     AVAILABLE                                                       00005320
0143      216 IF(IHLF)201,201,217                                             00005330
0144      217 IF(N-7)201,218,218                                              00005340
0145      218 IF(ISTEP-4)201,219,219                                          00005350
0146      219 IMOD=ISTEP/2                                                    00005360
0147          IF(ISTEP-IMOD-IMOD)201,220,201                                  00005370
0148      220 H=H+H                                                           00005380
0149          IHLF=IHLF-1                                                     00005390
0150          ISTEP=0                                                         00005400
0151          DO 221 I=1,NDIM                                                 00005410
0152          AUX(N-1,I)=AUX(N-2,I)                                           00005420
0153          AUX(N-2,I)=AUX(N-4,I)                                           00005430
0154          AUX(N-3,I)=AUX(N-6,I)                                           00005440
0155          AUX(N+6,I)=AUX(N+5,I)                                           00005450
0156          AUX(N+5,I)=AUX(N+3,I)                                           00005460
```

```
0157            AUX(N+4,I)=AUX(N+1,I)                                           00005470
0158            DELT=AUX(N+6,I)+AUX(N+5,I)                                      00005480
0159            DELT=DELT+DELT+DELT                                             00005490
0160        2210AUX(16,I)=8.962963*(Y(I)-AUX(N-3,I))-3.361111*H*(DERY(I)+DELT   00005500
               1+AUX(N+4,I))                                                    00005510
0161            GOTO 201                                                        00005520
          C                                                                     00005530
          C                                                                     00005540
          C     H MUST BE HALVED                                                00005550
0162        222 IHLF=IHLF+1                                                     00005560
0163            IF(IHLF-10)223,223,210                                          00005570
0164        223 H=.5*H                                                          00005580
0165            ISTEP=0                                                         00005590
0166            DO 224 I=1,NDIM                                                 00005600
0167           0Y(I)=.00390625*(80.*AUX(N-1,I)+135.*AUX(N-2,I)+40.*AUX(N-3,I)+  00005610
               1AUX(N-4,I))-.1171875*(AUX(N+6,I)-6.*AUX(N+5,I)-AUX(N+4,I))*H    00005620
0168           0AUX(N-4,I)=.00390625*(12.*AUX(N-1,I)+135.*AUX(N-2,I)+           00005630
               1108.*AUX(N-3,I)+AUX(N-4,I))-.0234375*(AUX(N+6,I)+18.*AUX(N+5,I)- 00005640
               29.*AUX(N+4,I))*H                                                00005650
0169            AUX(N-3,I)=AUX(N-2,I)                                           00005660
0170        224 AUX(N+4,I)=AUX(N+5,I)                                           00005670
0171            X=X-H                                                           00005680
0172            DELT=X-(H+H)                                                    00005690
0173            CALL FCT(DELT,Y,DERY)                                           00005700
0174            DO 225 I=1,NDIM                                                 00005710
0175            AUX(N-2,I)=Y(I)                                                 00005720
0176            AUX(N+5,I)=DERY(I)                                              00005730
0177        225 Y(I)=AUX(N-4,I)                                                 00005740
0178            DELT=DELT-(H+H)                                                 00005750
0179            CALL FCT(DELT,Y,DERY)                                           00005760
0180            DO 226 I=1,NDIM                                                 00005770
0181            DELT=AUX(N+5,I)+AUX(N+4,I)                                      00005780
0182            DELT=DELT+DELT+DELT                                             00005790
0183           0AUX(16,I)=8.962963*(AUX(N-1,I)-Y(I))-3.361111*H*(AUX(N+6,I)+DELT 00005800
               1+DERY(I))                                                       00005810
0184        226 AUX(N+3,I)=DERY(I)                                              00005820
0185            GOTO 206                                                        00005830
0186            END                                                             00005840
```

```
0001         SUBROUTINE FCT(T,Y,DERY)
          C
          C---------- ABSTRACT:  SUBROUTINE FCT MUST EVALUATE THE
          C----------      COEFFICIENTS FOR THE SECOND ORDER COEFFICIENT
          C----------      OF THE D2TH2 TERM AND THE LOW ORDER AND CONSTANT
          C----------      TERMS FOR THE RIGHT HAND SIDE OF THE ONE DEGREE
          C----------      OF FREEDOM SYSTEM DEVELOPED. FCT ' CALLED BY
          C----------      IBM-SSP SUBROUTINE HPCG TO EVALU     E  D2TH2
          C----------      TERM IN THE  A11  D2TH2 = F1  EXP  SSION.  BOTH
          C----------       A11  AND  F1  ARE KNOWN OR CAN BE DETERMINED.
          C
0002         DIMENSION Y(1),DERY(1)
0003         DATA A11MIN/1.E70/,D2TH2/1.E70/
          C------------------------------------------------ --------------------
          C--------------------------------------------------------------------
          C   VARIABLE     TYPE      DESCRIPTION
          C
          C T             REAL    TIME TO WHICH THE INTEGRATION HAS PROGRESSED (SEC
          C Y(1)          REAL    POSITION OF INPUT MOTOR (DEGREES)
          C Y(2)          REAL    ANGULAR VELOCITY OF INPUT MOTOR (DEG/SEC)
          C DERY(1)       REAL    FIRST DERIVITIVE OF Y(1) AT TIME T (DEG/SEC)
          C DERY(2)       REAL    SECOND DERIVITIVE OF Y(1) AT TIME T (DEG/SEC**2)
          C A11MIN        REAL    MIN VALUE OF A11 Y PREVENTS A ZERO DIVIDE, HOPEFU
          C TH2           REAL    INPUT MOTOR ANGLE AT TIME T (DEGREES)
          C DTH2          REAL    ANGULAR VELOCITY OF MOTOR INPUT ANGLE AT TIME T (
          C D2TH2         REAL    ANGULAR ACC. THAT SOLVES ABOVE EQUATION (DEG/SEC*
          C----- NOTE THAT TH2,DTH2,D2TH2 ARE ONLY LOCAL TO FCT,VALUES,COEFF# SUBS
          C A11,ALHS      REAL    LEFT HAND SIDE OF DIFFERENTIAL EQUATION OF MOTION
          C F1,ARHS       REAL    RIGHT HAND SIDE OF DIFF. EQ. OF MOTION
          C--------------------------------------------------------------------
          C
          C---------- EVALUATE THE CONSTANTS,  A11 , AND  F1
          C
0004         TH2=Y(1)
0005         DTH2=Y(2)
          C
0006         CALL VALUES(TH2,DTH2,ALHS,ARHS)
0007            A11=ALHS
0008            F1=-ARHS
          C
          C---------- CHECK FOR ZERO A11 TERM, WHICH WOULD BLOW
          C---------- PROGRAM. IF ZERO FIXUP SO RUN CAN CONTINUE
          C
0009         IF(A11.NE.0.) GO TO 100
0010            A11=A11MIN
0011            WRITE(6,1000) TH2,DTH2,D2TH2
          C
0012      100   CONTINUE
          C
          C---------- DETERMINE THE NEW D2TH2 AND UPDATE A11MIN IF
          C---------- NECESSARY
          C
0013         D2TH2=F1/A11
```

```
0014          IF(A11.LT.A11MIN) A11MIN=A11                                00006380
        C                                                                 00006390
        C---------- REPLACE INTO VECTOR DERY                              00006400
        C                                                                 00006410
                                                                          00006420
0015          DERY(1)=DTH2                                                00006430
0016          DERY(2)=D2TH2                                               00006440
        C                                                                 00006450
        C                                                                 00006460
0017             RETURN                                                   00006470
0018    1000  FORMAT(' ZERO A11 OUT OF FCT',/,                            00006480
             2     'OCCURED WHEN',F16.4,'=TH2',F16.4,'=DTH2',/,           00006490
             3     F16.4,'=D2TH2',/,'ZERO A11 IS RELPACED BY ',F16.4)     00006500
0019             END
```

```
0001          SUBROUTINE VALUES(TH2,DTH2,ALEFT,ARIGHT)                          00006510
        C                                                                      00006520
        C---------- ABSTRACT: SUBROUTINE VALUES DETERMINES THE                 00006530
        C----------      LEFT AND RIGHT HAND SIDES OF THE                      00006540
        C----------      DIFFERENTIAL EQUATION.  ALEFT CORRESPONDS             00006550
        C----------      TO THE D2DTH2 COEFFICIENT                             00006560
        C----------      WHILE ARIGHT CORRESPONDS TO THE                       00006570
        C----------      KNOWN RIGHT HAND SIDE.                                00006580
        C----------------------------------------------------------------------00006590
        C  VARIABLE     TYPE      DESCRIPTION                                  00006600
        C                                                                      00006610
        C ALEFT        REAL    SUM OF INDIVIDUAL LEFT AAND SIDE COMPONENTS OF DI00006620
        C ALEFT        REAL    SUM OF INDIVIDUAL LEFT HAND SIDE COMPONENTS OF DI00006630
        C AL2          REAL    COEFF OF 2ND DERIV. FOR INPUT MOTOR COMPONENTS   00006640
        C AR2          REAL    1ST DERIV. AND CONSTANTS VALUE FOR INPUT MOTOR CO000006650
        C AL3          REAL    COEFF. OF  ND DERIV. FOR DRUM/FACE CAM COMPONENTS00006660
        C AL4          REAL    CONSTANTS & 1ST DER VALUES FOR DRUM/FACE CAM COMP00006670
        C AR4          .REAL   CONSTANT, & 1ST DER VALUES FOR COMPONENTS OF FEED00006680
        C AL5          REAL    COEFF OF 2ND DERIVITIVE FOR COMPONENTS OF EJECT M00006690
        C AR5          REAL    CONSTANTS & 1ST DER VALUES FOR COMPONENOS OF EJEC00006700
        C AL6          REAL    COEFF OF 2ND DERIVITIVE FOR COMPONENTS OF LOCK RI00006710
        C AR6          REAL    CONSTANTS & 1ST DER VALUES FOR COMPONENOS OF LOCK00006720
        C AL7          REAL    COEFF OF 2ND DERIVITIVE FOR9COMPONENTS O> CHAMBER00006730
        C AR7          REAL    CONSTANTS & 1ST DER VALUES FOR COMPONENTS OF CHAM00006740
        C TH2          REAL    INPUT MOTOR ANGLE AT TIME T (DEGREES)            00006750
        C DTH2         REAL    ANGULAR VELOCITY OF MOTOR INPUT ANGLE AT TIME T  00006760
        C----- NOTE THAT TH2,DTH2,D2TH2 ARE ONLY LOCAL TO FCT,VALUES,COEFF# SUBS00006770
        C----------------------------------------------------------------------00006780
        C---------- FIND THE INDIVIDUAL COEFFICIENTS FROM THE                  00006790
        C---------- COMPONENT PARTS                                            00006800
        C                                                                      00006810
0002          CALL COEF2(TH2,DTH2,AL2,AR2)                                      00006820
0003          CALL COEF3(TH2,DTH2,AL3,AR3)                                      00006830
0004          CALL COEF4(TH2,DTH2,AL4,AR4)                                      00006840
0005          CALL COEF5(TH2,DTH2,AL5,AR5)                                      00006850
0006          CALL COEF6(TH2,DTH2,AL6,AR6)                                      00006860
0007          CALL COEF7(TH2,DTH2,AL7,AR7)                                      00006870
        C                                                                      00006880
        C---------- COMPUTE ALEFT,ARIGHT AND RETURN                            00006890
        C                                                                      00006900
0008          ALEFT=AL2+AL3+AL4+AL5+AL6+AL7                                     00006910
0009          ARIGHT=AR2+AR3+AR4+AR5+AR6+AR7                                    00006920
        C                                                                      00006930
        C                                                                      00006940
0010             RETURN                                                        00006950
0011             END                                                           00006960
```

```
0001        SUBROUTINE COEF2(TH2,DTH2,AL2,AR2)                              00006970
      C                                                                     00006980
      C                                                                     00006990
      C                                                                     00007000
      C                                                                     00007010
      C                                                                     00007020
0002        COMMON/MTORQ/RPM(1000),TRPM(1000),NPTTQ                         00007030
0003        REAL I59/.000949/,I120/.01546/,RATIO/.4916666/                  00007040
      C---------------------------------------------------------------------00007050
      C   VARIABLE     TYPE      DESCRIPTION                                00007060
      C                                                                     00007070
      C---- COMMON MTORQ                                                    00007080
      C RPM(1000)     REAL    VECTOR OF RPM VALUES FOR INPUT MOTOR (DEG/SEC) 00007090
      C TRPM(1000)    REAL    VECTOR OF TORQUES CORRES. TO RPM VECTOR (FT-LBS) 00007100
      C NPTTQ         INTEGER POINTS IN RPM-TRPM TORQUE/SPEED CURVE VECTORS  00007110
      C                                                                     00007120
      C AL2           REAL    COEFF OF 2ND DERIV. FOR INPUT MOTOR COMPONENTS 00007130
      C AR2           REAL    1ST DERIV. AND CONSTANTS VALUE FOR INPUT MOTOR CO00007140
      C I59           REAL    INERTIA MOMENT OF 59TTH GEAR (FT-LB-SEC**2)    00007150
      C I120          REAL    INERTIA MOMENT OF 120 TTH IDLER GEAR (FT-LB-SEC**00007160
      C TORQUE        REAL    TORQUE AT DTH2 INTERPOLATED FROM RPM-TRPM CURVE (00007170
      C TH2           REAL    INPUT MOTOR ANGLE AT TIME T (DEGREES)          00007180
      C DTH2          REAL    ANGULAR VELOCITY OF MOTOR INPUT ANGLE AT TIME T 00007190
      C----- NOTE THAT TH2,DTH2,D2TH2 ARE ONLY LOCAL TO FCT,VALUES,COEFF# SUBS00007200
      C---------------------------------------------------------------------00007210
      C                                                                     00007220
      C                                                                     00007230
      C                                                                     00007240
      C---------- COMPUTE AL2,AR2                                           00007250
      C                                                                     00007260
0004        AL2=-(I59+I120*RATIO**2)                                        00007270
0005        CALL INTERP(RPM,TRPM,NPTTQ,DTH2/6.,TORQUE,DUM1,DUM2,1,0)        00007280
0006        AR2=TORQUE - .1*TORQUE                                          00007290
      C                                                                     00007300
      C                                                                     00007310
0007          RETURN                                                        00007320
0008          END                                                           00007330
```

```
0001          SUBROUTINE COEF3(TH2,DTH2,AL3,AR3)                              00007340
        C                                                                     00007350
        C                                                                     00007360
        C                                                                     00007370
        C                                                                     00007380
        C                                                                     00007390
0002          COMMON/DRUM3/DTH3(1000)                                         00007400
0003          COMMON/MOTOR2/AMTH2(1000),NPTS2                                 00007410
0004          COMMON/UPDTE3/TH3,D1TH3,D2TH3                                   00007420
0005          REAL IDRUM/.0997/,IFACE/.04406/                                 00007430
        C---------------------------------------------------------------------00007440
        C   VARIABLE      TYPE      DESCRIPTION                               00007450
        C                                                                     00007460
        C TH2          REAL    INPUT MOTOR ANGLE AT TIME T (DEGREES)          00007470
        C DTH2         REAL    ANGULAR VELOCITY OF MOTOR INPUT ANGLE AT TIME T (00007480
        C----- NOTE THAT TH2,DTH2,D2TH2 ARE ONLY LOCAL OO FCT,VALUES,COEFF# SUBS00007490
        C AL3          REAL    COEFF. OF 2ND DERIV. FOR DRUM/FACE CAM COMPONENTS00007500
        C---- COMMON MOTOR2                                                   00007510
        C AMTH2(1000)  REAL    VECTOR OF SUCESSIVE INPUT MOTOR POSITIONS (DEG) 00007520
        C NPTS2        INTEGER POINTS IN AMTH2,DRUM3,FACE3F,FEED4,EJECT5,LOCK6,D00007530
        C---- COMMON DRUM3                                                    00007540
        C DTH3(1000)   REAL    VECTOR OF DRUM  POSITIONS FOR CORRESPONDING AMTH200007550
        C                                                                     00007560
        C IDRUM        REAL    INERTIA MOMENT OF DRUM CAM (FT-LB-SEC**2)      00007570
        C IFACE        REAL    INERTIA MOMENT OF THE FACE CAM (FT-LB-SEC**2)  00007580
        C SUM          REAL    SUM OF FACE AND DRUM MOMENTS OF INERTIA (FT-LB-SE00007590
        C---------------------------------------------------------------------00007600
        C                                                                     00007610
        C                                                                     00007620
0006          CALL INTERP(AMTH2,DTH3,NPTS2,AMOD(TH2,6910.17),TH3,D1TH3,D2TH3,3,000007630
             1            )                                                   00007640
        C                                                                     00007650
        C                                                                     00007660
0007          SUM=IDRUM+IFACE                                                 00007670
0008          AL3=-(D1TH3**2)*SUM                                             00007680
0009          AR3= -(DTH2**2)*(D1TH3*D2TH3*SUM)                               00007690
        C                                                                     00007700
        C                                                                     00007710
0010             RETURN                                                       00007720
0011             END                                                          00007730
```

```
0001          SUBROUTINE COEF4(TH2,DTH2,AL4,AR4)
        C
        C
        C
        C
        C
0002          COMMON/FEED4/FTH4(1000)
0003          COMMON/MOTOR2/AMTH2(1000),NPTS2
0004          COMMON/UPDTE4/TH4,D1TH4,D2TH4
0005          REAL ISHFT/.000018647/,IROCK/.00002724/,IPAWL/.00032958/
0006          REAL MAMMO/.04255/,MROCK/.0049232/,MPAWL/.010888/
0007          REAL THCNST/215./,GRAV/32.2/,RCM/.08333/,PE/.3000/
0008          DATA PCM/.166670/
        C-----------------------------------------------------------------
        C   VARIABLE     TYPE       DESCRIPTION
        C
        C---- COMMON EJECT5
        C FTH4          REAL    VECTOR OF FEED POSITIONS FOR CORRESPONDING AMTH2
        C---- COMMON MOTOR2
        C AMTH2(1000)   REAL    VECTOR OF SUCESSIVE INPUT MOTOR POSITIONS (DEG)
        C NPTS2         INTEGER POINTS IN AMTH2,DRUM3,FACE3F,FEED4,EJECT5,LOCK6,D
        C
        C TH2           REAL    INPUT MOTOR ANGLE AT TIME T (DEGREES)
        C DTH2          REAL    ANGULAR VELOCITY OF MOTOR INPUT ANGLE AT TIME T (
        C----- NOTE THAT TH2,DTH2,D2TH2 ARE ONLY LOCAL OD FCT,VALUES,COEFF# SUBS
        C AL4           REAL    CONSTANTS & 1ST DER VALUES FOR DRUM/FACE CAM COMP
        C AR4           REAL    CONSTANTS & 1ST DER VALUES FOR COMPONENTS OF FEED
        C ISHFT         REAL    INERTIA MOMENT OF FEED SHAFT (FT-LB-SEC**2)
        C IROCK         REAL    INERTIA MOMENT OF ROCKER PAWL (FT-LB-SEC**2)
        C IPAWL         REAL    INERTIA MOMENT OF FEED PAWL (FT-LB-SEC**2)
        C AMMO          REAL    MASS OF THE AMMO ROUND (LB-SEC**2/FT)
        C MROCK         REAL    MASS OF THE ROCKER ARM (LB-SEC**2/FT)
        C MPAWL         REAL    MASS OF THE FEED PAWL (LB-SEC**2/FT)
        C THCNST        REAL   THE CONSTANT ANGLE BETWEEN FEED PAWL AND ROCKER A
        C GRAV          REAL    ACCELERATION DUE GRAVITY (FT/SEC**2)
        C RCM           REAL    LENGTH FROM S  FT CENTER TO ROCKER MASS CENTER (F
        C PCM           REAL    LENGTH FROM SHAFT CENTER TO PAWL MASS CENTER (FT)
        C PE            REAL    LENGTH FROM SHAFT CENTER TO PAWL END (FT)
        C C4            REAL    COSINE OF FEED PAWL ANGLE
        C C4C           REAL    COSINE OF ROCKER ARM ANGLE
0009           CONTINUE
        C ARG1          REAL
        C ARG2          REAL
        C-----------------------------------------------------------------
        C
        C
0010          CALL INTERP(AMTH2,FTH4,NPTS2,AMOD(TH2,6910.17),TH4,D1TH4,D2TH4,3,0
             1            )
        C
        C
0011          C4=COS(TH4/57.2957795)
0012          C4C=COS(TH4/57.2957795+THCNST/57.2957795)
        C
```

```
              C
              C
 0013              AL4=-(DITH4**2)*(IPAWL+MPAWL*PCM**2+IROCK+MROCK*RCM**2+
                  1     ISHFT+UNIT(-DITH4)*MAMMO*PE**2)
              C
              C
 0014              AR4=-(DTH2**2*DITH4*D2TH4)*(IPAWL+MPAWL*PCM**2+IROCK+
                  1     MROCK*RCM**2+ISHFT+UNIT(-DITH4)*MAMMO*PE**2)
                  2    -GRAV*DITH4*(PCM*MPAWL*C4+RCM*MROCK*C4C)
                  3    -TOR(AR3,ARG1)*DITH4
              C
              C
              C
              C
 0015                RETUR+
 0016                END
```

```
0001          SUBROUTINE COEFS(TH2,DTH2,AL5,AR5)                               00008430
        C                                                                      00008440
        C                                                                      00008450
        C                                                                      00008460
        C                                                                      00008470
        C                                                                      00008480
0002          COMMON/EJECT5/ETH5(1000)                                         00008490
0003          COMMON/MOTOR2/AMTH2(1000),NPTS2                                  00008500
0004          COMMON/UPDTE5/TH5,D1TH5,D2TH5                                    00008510
0005          REAL ISHFT/.00016782/,IROCK/.00002724/,IPAWL/.00016703/          00008520
0006          REAL MROCK/.0049232/,MPAWL/.017473/                              00008530
0007          REAL THCNST/79.5/,GRAV/32.2/,RCM/.08333/,PE/.3000/               00008540
0008          DATA PCM/.166670/                                                00008550
        C                                                                      00008560
        C----------------------------------------------------------------------00008570
        C   VARIABLE     TYPE      DESCRIPTION                                 00008580
        C---- COMMON EJECT5                                                    00008590
        C ETH5(1000)   REAL    VECTOR OF EJECT POSITIONS FOR CORRESPONDING AMTH200008600
        C---- COMMON MOTOR2                                                    00008610
        C AMTH2(1000)  REAL    VECTOR OF SUCESSIVE INPUT MOTOR POSITIONS (DEG) 00008620
        C NPTS2        INTEGER POINTS IN AMTH2,DRUM3,FACE3F,FEED4,EJECT5,LOCK6,D00008630
        C                                                                      00008640
        C TH2          REAL    INPUT MOTOR ANGLE AT TIME T (DEGREES)           00008650
        C DTH2         REAL    ANGULAR VELOCITY OF MOTOR INPUT ANGLE AT TIME T (00008660
        C----- NOTE THAT TH2,DTH2,D2TH2 ARE ONLY LOCAL OD FCT.VALUES,COEFF# SUBS00008670
        C AL5          REAL    COEFF OF 2ND DERIVITIVE FOR COMPONENTS OF EJECT M00008680
        C AR5          REAL    CONSTANTS & 1ST DER VALUES FOR COMPONENOS OF EJEC00008690
        C ISHFT        REAL    INERTIA MOMENT OF EJECT SHAFT (FT-LB-SEC**2)    00008700
        C IROCK        REAL    INERTIA MOMENT OF ROCKER PAWL (FT-LB-SEC**2)    00008710
        C IPAWL        REAL    INERTIA MOMENT OF EJECT PAWL (FT-LB-SEC**2)     00008720
        C MROCK        REAL    MASS OF THE ROCKER ARM (LB-SEC**2/FT)           00008730
        C MPAWL        REAL    MASS OF THE EJECT PAWL (LB-SEC**2/FT)           00008740
        C THCNST       REAL   THE CONSTANT ANGLE BETWEEN EJECT PAWL AND ROCKER A00008750
        C GRAV         REAL    ACCELERATION DUE GRAVITY (FT/SEC**2)            00008760
        C RCM          REAL    LENGTH FROM SHAFT CENTER TO ROCKER MASS CENTER (F00008770
        C PCM          REAL    LENGTH FROM SHAFT CENTER TO PAWL MASS CENTER (FT)00008780
        C PE           REAL    LENGTH FROM SHAFT CENTER TO PAWL END (FT)       00008790
        C C5           REAL    COSINE OF EJECT PAWL ANGLE                      00008800
        C C4C          REAL    COSINE OF ROCKER ARM ANGLE                      00008810
        C----------------------------------------------------------------------00008820
        C                                                                      00008830
        C                                                                      00008840
0009          CALL INTERP(AMTH2,ETH5,NPTS2,AMOD(TH2-5910.17),TH5,D1TH5,D2TH5,3,000008850
             1              )                                                  00008860
        C                                                                      00008870
        C                                                                      00008880
0010          C5=COS(TH5/57.2957795)                                           00008890
0011          C5C=COS(TH5/57.2957795 - THCNST/57.2957795)                      00008900
        C                                                                      00008910
        C                                                                      00008920
        C                                                                      00008930
0012          AL5=-(D1TH5**2)*(IPAWL+MPAWL*PCM**2+IROCK+MROCK*RCM**2+          00008940
             1       ISHFT)                                                    00008950
```

```
            C
            C
 0013             AR5=-(DTH2**2*D1TH5*D2TH5)*(IPAWL+MPAWL*PCM**2+IROCK+
                 1      MROCK*PCM**2+ISHFT)
                 2    -GRAV*D1TH5*(PCM*MPAWL*C5+RCM*MROCK*C5C)
            C
            C
            C
 0014               RETURN
 0015               END
```

```
0001              SUBROUTINE COEF6(TH2,DTH2,AL6,AR6)                                 00009060
            C                                                                        00009070
            C                                                                        00009080
            C                                                                        00009090
            C                                                                        00009100
            C                                                                        00009110
0002              REAL ILOCK/.0020653/                                               00009120
0003              COMMON/LOCK6/ALTH6(1000)                                           00009130
0004              COMMON/MOTOR2/AMTH2(1000),NPTS2                                    00009140
0005              COMMON/UPDTE6/TH6,DTH6,D2TH6                                       00009150
            C                                                                        00009160
            C------------------------------------------------------------------------00009170
            C  VARIABLE      TYPE       DESCRIPTION                                  00009180
            C---- COMMON LOCK6                                                       00009190
            C ALTH6(1000)  REAL    VECTOR OF LOCKT POSITIONS FOR CORRESPONDING AMTH200009200
            C---- COMMON MOTOR2                                                      00009210
            C AMTH2(1000)  REAL    VECTOR OF SUCESSIVE INPUT MOTOR POSITIONS (DEG)   00009220
            C NPTS2        INTEGER POINTS IN AMTH2,DRUM3,FACE3F,FEED4,EJECT5,LOCK6,D00009230
            C                                                                        00009240
            C TH2          REAL    INPUT MOTOR ANGLE AT TIME T (DEGREES)             00009250
            C DTH2         REAL    ANGULAR VELOCITY OF MOTOR INPUT ANGLE AT TIME T (D00009260
            C----- NOTE THAT TH2,DTH2,D2TH2 ARE ONLY LOCAL TO FCT,VALUES,COEFF# SUBS00009270
            C------------------------------------------------------------------------00009280
            C                                                                        00009290
            C                                                                        00009300
0006              CALL INTERP(AMTH2,ALTH6,NPTS2,AMOD(TH2,6910.17),TH6,DTH ,D2TH6,3,000009310
                 1            )                                                      00009320
            C                                                                        00009330
            C                                                                        00009340
0007              AL6=-(DTH6**2)*ILOCK                                               00009350
0008              AR6=-(DTH2**2)*(DTH6*D2TH6*ILOCK)-DTH6*TOR(ARG1,ARG2)              00009360
            C                                                                        00009370
            C                                                                        00009380
0009                 RETURN                                                          00009390
0010                 END                                                             00009400
```

```
0001              SUBROUTINE COEF7(TH2,DTH2,AL7,AR7)
            C
            C
            C
            C
0002              COMMON/DISP7/RAD7(1000)
0003              COMMON/MOTOR2/AMTH2(1000),NPTS2
0004              COMMON/UPDTE7/R7,DR7,D2R7
0005              DATA CFRICT/.9/
            C-----------------------------------------------------------------
            C   VARIABLE      TYPE       DESCRIPTION
            C
            C---- COMMON DISP7
            C RAD7(1000)    REAL    VECTOR OF CHAMBER DISPLACEMENTS COR. TO AMTH2 (FT
            C---- COMMON MOTOR2
            C AMTH2(1000)   REAL    VECTOR OF SUCESSIVE INPUT MOTOR POSITIONS (DEG)
            C NPTS2         INTEGER POINTS IN AMTH2,DRUM3,FACE3F,FEED4,EJECT5,LOCK6,D
            C TH2           REAL    INPUT MOTOR ANGLE AT TIME T (DEGREES)
            C DTH2          REAL    ANGULAR VELOCITY OF MOTOR INPUT ANGLE AT TIME T (
            C----- NOTE THAT TH2,DTH2,D2TH2 ARE ONLY LOCAL TO FCT,VALUES,COEFF# SUBS
            C AL7           REAL    COEFF OF 2ND DERIVITIVE FOR9COMPONENTS OF CHAMBER
            C AR7           REAL    CONSTANTS & 1ST DER VALUES FOR COMPONENTS OF CHAM
            C VCHMBR        REAL    TRANSLATING MASS ASSOCIATED WITH CHAMBER MOTION (
            C-----------------------------------------------------------------
            C
            C
0006              CALL INTERP(AMTH2,RAD7,NPTS2,AMOD(TH2,6910.17),R7,DR7,D2R7,3,0)
0007              CALL CHAMBR(CRUSH,SEAR,VCHMBR)
            C
            C
0008              AL7=(DR7**2)*VCHMBR
0009              AR7=(DTH2**2)*VCHMBR*DR7*D2R7 + DR7*(CRUSH-SEAR)
0010                 IF(DR7.NE.0.) AR7=AR7+CFRICT*(DR7/ABS(DR7))
            C
0011                 RETURN
0012                 END
```

```
0001         SUBROUTINE OUTP(T,Y,DERY,IHLF,NDIM,PRMT)
      C
      C---------- ABSTRACT:  SUBROUTINE OUTP MUST MAKE THE DECISION
      C----------     ON WHAT TO DO WITH THE RESULTS OF THE
      C----------     JUST COMPLETED INTEGRATION STEP.  NOTE
      C----------     THAT TOUT IS THE NEXT OUTOUT TIME VALUE
      C----------     SO THAT IT IS THE TEST VALUE
      C
0002         COMMON/TIMES/TINIT,TFINAL,TNOW,TOUT,TSTEP,TNEXT
0003         COMMON/ANGLE2/TH2,DTH2,D2TH2
0004         DIMENSION Y(1),DERY(1),PRMT(1)
      C------------------------------------------------------------------
      C   VARIABLE     TYPE      DESCRIPTION
      C
      C---- COMMON TIMES
      C TINIT        REAL    INITIAL TIME FOR INTEGRATION (SECONDS)
      C TFINAL       REAL    FINAL AND TERMINATING TIME FOR INTEGRATION (SECON
      C TNOW         REAL    TIME TO WHICH THE INTEGRATION HAS PROGRESSED (SEC
      C TOUT         REAL    TIME STEP THAT MUST BE EXCEEDED TO GET AN OUTPUT
      C TSTEP        REAL    INITIAL TIME STEP SIZE TO START INTEGRATION (SECO
      C TNEXT        REAL    ABSOLUTE TIME FOR NEXT OUTPUT STEP. TNEXT2=TNEXT1
      C---- COMMON ANGLE2
      C TH2          REAL    ROTATION OF INPUT MOTOR GEAR (DEGREES)
      C DTH2         REAL    FIRST TIME DERIVITIVE OF TH2 (DEG/SEC)
      C D2TH2        REAL    SECOND TIME DERIVATIVE OF TH2 (DEG/SEC**2)
      C
      C T            REAL    TIME TO WHICH THE INTEGRATION HAS PROGRESSED (SEC
      C Y(1)         REAL    POSITION OF INPUT MOTOR (DEGREES)
      C Y(2)         REAL    ANGULAR VELOCITY OF INPUT MOTOR (DEG/SEC)
      C DERY(1)      REAL    FIRST DERIVITIVE OF Y(1) AT TIME T (DEG/SEC)
      C DERY(2)      REAL    SECOND DERIVITIVE OF Y(1) AT TIME T (DEG/SEC**2)
      C PRMT(1)      REAL    INITIAL TIME SUPPLIED TO HPCG ROUTINE, TINIT (SEC
      C PRMT(2)      REAL    FINAL TIME GIVEN TO HPCG, =TFINAL (SECONDS)
      C PRMT(3)      REAL    OUTPUT OCCURS EVERY PRMT(3) SECONDS
      C PRMT(4)      REAL    ABSOLUTE ERROR UPON WHICH INTEGRATING TIME STEP I
      C PRMT(5)      REAL    UNUSED HERE.  CAN BE USED OUT OF OUTP ROUTINE
      C AUX(16,4)    REAL    SCRATCH VECTOR USED IN HPCG
      C NDIM         INTEGER NUMBER OF ELEMENTS IN DERY VECTOR
      C------------------------------------------------------------------
0005          CONTINUE
      C
      C---------- RETURN IF NOT TO AN OUTPUT STEP
      C
0006            IF(T.LT.TNEXT) RETURN
      C
      C---------- TIME TO DO AN OUTPUT, UPDATE THE SYSTEM
      C
0007         TNEXT=TNEXT+TOUT
0008         TH2=Y(1)
0009         DTH2=Y(2)
0010         TNOW=T
0011         D2TH2=DERY(2)
      C
```

```
0012            CALL UPDATE                                         00010300
          C                                                         00010310
          C---------- CALL THE OUTPUT ROUTINES                      00010320
          C                                                         00010330
0013            CALL OUTPUT                                         00010340
          C                                                         00010350
          C                                                         00010360
0014               RETURN                                           00010370
0015               END                                              00010380
```

```
 0001                 SUBROUTINE OUTPUT                                               00010390
           C                                                                          00010400
 0002                 COMMON/TIMES/TINIT,TFINAL,TNOW,TOUT,TSTEP,TNEXT                 00010410
 0003                 COMMON/ANGLE2/TH2,DTH2,D2TH2                                    00010420
 0004                 DATA ITHRU/0/                                                   00010430
           C                                                                          00010440
           C                                                                          00010450
           C                                                                          00010460
           C                                                                          00010470
 0005                 IF(ITHRU.GT.5) GO TO 25                                         00010480
 0006                    WRITE( 6,999)                                                00010490
 0007      999            FORMAT('1',///)                                             00010500
 0008                    ITHRU=10                                                     00010510
 0009        25       CONTINUE                                                        00010520
 0010                 WRITE( 6,1000) TH2,DTH2,D2TH2,TNOW                              00010530
           C          WRITE(12,1000) TH2,DTH2,D2TH2,TNOW                              00010540
 0011      1000       FORMAT(' ',4F20.5)                                              00010550
           C                                                                          00010560
           C                                                                          00010570
           C                                                                          00010580
 0012                  WRITE(13,1234) TNOW,TH2                                        00010590
 0013                  WRITE(14,1234) TNOW,DTH2                                       00010600
 0014                  WRITE(15,1234) TNOW,D2TH2                                      00010610
 0015      1001        FORMAT(2F16.4)                                                 00010620
 0016      1234        FORMAT(' ',2F20.5)                                             00010630
           C                                                                          00010640
           C                                                                          00010650
           C                                                                          00010660
           C                                                                          00010670
 0017                 RETURN                                                          00010680
 0018                 END                                                             00010690
```

```
0001              SUBROUTINE ANGLES(A,B,C,BC,AC,AB)
          C
          C----------  SUBROUTINE ANGLES ACCEPTS THE 3 SIDES OF ANY TRIANGLE AND C
          C----------  THE INTERNAL ANGLES
          C----------  A,B,C ARE THE SIDE LENGTHS
          C----------  AB IS THE ANGLE BETWEEN SIDES A AND B
          C
          C---------- DETERMINE THE LONGEST SIDE
          C
0002              TEST=AMAX1(A,B,C)
0003              IF(TEST.EQ.A) GO TO 100
0004              IF(TEST.EQ.B) GO TO 200
0005              IF(TEST.EQ.C) GO TO 300
          C
          C----------  SHOULD NEVER BE HERE
          C
0006                 WRITE(6,1000)
0007                 GO TO 100
          C
0008      100     CONTINUE
0009              CALL ANGL(A,B,C,BC,AC,AB)
0010                 GO TO 400
0011      200     CONTINUE
0012                 CALL ANGL(B,A,C,AC,BC,AB)
0013                 GO TO 400
0014      300     CONTINUE
0015                 CALL ANGL(C,A,B,AB,BC,AC)
0016      400     CONTINUE
0017              TEST=ABS(AB+AC+BC-180.)
0018              IF(TEST.LT..1) GO TO 9000
          C
          C----------  IF HERE, ALGORITHM HAS FAILED. FOR NOW DO NOTHING
          C
0019              WRITE(6,1001)
0020              GO TO 9000
0021      9000    CONTINUE
0022                 RETURN
0023      1000    FORMAT(' PASSED NO TESTS IN ANGLES')
0024      1001    FORMAT(' INTERIOR ANGLES DO NOT TOTAL TO 180.')
0025                 END
```

```
0001          SUBROUTINE ANGL(A,B,C,BC,AC,AB)                              00011100
        C                                                                  00011110
        C-----------  CALCULATE INTERIOR ANGLES OF A TRIANGLE WHEN A IS    00011120
        C-----------  THE LARGEST SIDE AND A,B,C ARE KNOWN                 00011130
        C                                                                  00011140
0002          BC=ARCOS(((B*B+C*C-A*A)/2.*B*C))                             00011150
0003          AC=ARSIN(B*SIN(BC)/A)                                        00011160
0004          AB=ARSIN(C*SIN(BC)/A)                                        00011170
        C                                                                  00011180
        C                                                                  00011190
        C                                                                  00011200
0005          AB=AB*57.2957795                                             00011210
0006          AC=AC*57.2957795                                             00011220
0007          BC=BC*57.2957795                                             00011230
        C                                                                  00011240
0008          RETURN                                                       00011250
0009          END                                                          00011260
```

```
FORTRAN IV G LEVEL  21                  TOR                 DATE = 77104          06/14/24            PAGE 0001

 0001            FUNCTION TOR(X,Y)                                                   00011270
          C                                                                          00011280
          C                                                                          00011290
          C                                                                          00011300
          C                                                                          00011310
 0002              TOR=0.0                                                           00011320
          C                                                                          00011330
          C                                                                          00011340
 0003            RETURN                                                              00011350
 0004            END                                                                 00011360
```

```
FORTRAN IV G LEVEL  21                UNIT                DATE = 77104        06/14/24          PAGE 0001
                                                                              00011370
 0001              FUNCTION UNIT(X)                                           00011380
             C                                                                00011390
             C                                                                00011400
             C                                                                00011410
             C                                                                00011420
             C                                                                00011430
 0002              UNIT=1.0                                                   00011440
 0003              IF(X.LE.0.0) UNIT=0.0                                      00011450
             C                                                                00011460
             C                                                                00011470
 0004                 RETURN                                                  00011480
 0005              END
```

```
0001          SUBROUTINE UPDATE
           C
           C
           C---------- ABSTRACT:  SUBROUTINE UPDATE IS THE EXECUTIVE WHICH
           C---------- CAUSES DETERMINATION OF PART POSITIONS,VELOCITIES,
           C---------- ACCELERATIONS,AND FORCES FOR THE MAJOR MODELED
           C---------- COMPONENTS.  EACH MAJOR SUB-ASSEMBLY IS UPDATED
           C---------- BY AN INDIVIDUAL SUBROUTINE AND A LARGE TABLE IS GENERATED W
           C---------- IS THE TIME HISTORY AND UPDATED VALUES FOR THE COMPONENTS OF
           C---------- INTEREST.
           C
           C
0002          DIMENSION VEC2(6),VEC3(6),VEC4(6),VEC5(6),VEC6(6),VEC7(6)
0003          COMMON/ANGLE2/TH2,DTH2,D2TH2
0004          DATA ITHRU/0/
           C
           C-------------------------------------------------------------------
           C   VARIABLE     TYPE       DESCRIPTION
           C---- COMMON ANGLE2
           C TH2           REAL    ROTATION OF INPUT MOTOR GEAR (DEGREES)
           C DTH2          REAL    FIRST TIME DERIVITIVE OF TH2 (DEG/SEC)
           C D2TH2         REAL    SECOND TIME DERIVATIVE OF TH2 (DEG/SEC**2)
           C
           C VEC3(1)        REAL    ROTATION OF THE DRUM CAM(DEG)
           C     3(2)       REAL    ANG VELOCITY OF DRUM CAM(DEG/SEC)
           C     3(3)       REAL    ANG ACC OF DRUM CAM(DEG/SEC**2)
           C     3(4)       REAL    TORQUE ACTING ON DRUM CAM
           C VEC4(1)        REAL    ROTATION OF THE FEED PAWL(DEG)
           C     4(2)       REAL    ANG VELOCITY OF FEED PAWL(DEG/SEC)
           C     4(3)       REAL    ANG ACC OF FEED PAWL(DEG/SEC**2)
           C     4(4)       REAL    FORCE ACCELERATING AMMO
           C     4(5)       REAL    TORQUE ACTING ON FEED ABOUT SHAFT
           C     4(6)       REAL    FORCE ACTING ON ROLLER BEARING
           C VEC5(1)        REAL    ROTATION OF THE EJECT PAWL(DEG)
           C     5(2)       REAL    ANG VELOCITY OF EJECT PAWL(DEG/SEC)
           C     5(3)       REAL    ANG ACC OF EJECT PAWL(DEG/SEC**2)
           C     5(4)       REAL    TORQUE ACTING ON EJECT ABOUT SHAFT
           C     5(5)       REAL    FORCE ACTING ON ROLLER BEARING
           C VEC6(1)        REAL    ROTATION OF THE LOCK CAM(DEG)
           C     6(2)       REAL    ANG VELOCITY OF LOCK CAM(DEG/SEC)
           C     6 3)       REAL    ANG ACC OF LOCK CAM(DEG/SEC**2)
           C     6(4)       REAL    TORQUE ACTING ON LOCK CAM
           C     6(5)       REAL    FORCE ON THE ACTUATOR GEAR SET
0005            CONTINUE
           C VEC7(1)        REAL    DISPLACEMENT OF THE CHAMBER (IN)
           C     7(2)       REAL    VELOCITY OF CHAMBER(IN/SEC)
           C     7(3)       REAL    ACC OF CHAMBER(IN/SEC**2)
           C     7(4)       REAL    CRUSH FORCE EXERTED BY ROUND (LBS)
           C     7(5)       REAL    SEAR FORCE DUE RE-SEARING (LBS)
           C     7(6)       REAL    FORCE ON THE CHAMBER TO DRUM CAM STUD (LBS)
           C-------------------------------------------------------------------
           C
           C---------- FILL OUT POSSIBLE UNUSED VECTOR LOCATIONS
```

```
             C                                                                  00012020
 0006              IF(ITHRU.GT.5) GO TO 200                                     00012030
 0007                 DO 100 J=5,6                                              00012040
 0008                    VEC3(J)=99999.9                                        00012050
 0009                    VEC4(J)=99999.9                                        00012060
 0010                    VEC5(J)=99999.9                                        00012070
 0011                    VEC6(J)=99999.9                                        00012080
 0012                    VEC7(J)=99999.9                                        00012090
 0013        100      CONTINUE                                                  00012100
 0014              ITHRU=10                                                     00012110
 0015        200   CONTINUE                                                     00012120
 0016              CALL UPDT3(TH2,DTH2,D2TH2,VEC3)                              00012130
 0017              CALL UPDT4(TH2,DTH2,D2TH2,VEC4)                              00012140
 0018              CALL UPDT5(TH2,DTH2,D2TH2,VEC5)                              00012150
 0019              CALL UPDT6(TH2,DTH2,D2TH2,VEC6)                              00012160
 0020              CALL UPDT7(TH2,DTH2,D2TH2,VEC7)                              00012170
             C                                                                  00012180
             C                                                                  00012190
             C---------- GENERATE THE OUTPUT TABLE                              00012200
             C                                                                  00012210
 0021              CALL TBLOUT(1,VEC2,VEC3,VEC4,VEC5,VEC6,VEC7,TH2,DTH2,D2TH2)  00012220
             C                                                                  00012230
             C                                                                  00012240
 0022                 RETURN                                                    00012250
 0023                 END                                                       00012260
```

```
0001             SUBROUTINE UPDT3(TH2,DTH2,D2TH2,VEC)                         00012270
           C                                                                  00012280
           C---------- ABSTRACT: SUBR UPDT3 UPDATES SOME OF THE FACE AND DRUM   00012290
           C---------- PARAMETERS.  THE PARAMETERS ARE SHOWN IN THE VARIABLE LIST. 00012300
           C                                                                  00012310
           C                                                                  00012320
0002             DIMENSION VEC(1)                                             00012330
0003             COMMON/UPDTE3/TH3,DTH3,D2TH3                                 00012340
0004             REAL IDRUM/.0997/,IFACE/.04406/                              00012350
0005             DATA DG2RAD/.0174533/                                        00012360
           C                                                                  00012370
           C------------------------------------------------------------------00012380
           C   VARIABLE     TYPE      DESCRIPTION                             00012390
           C                                                                  00012400
           C---- COMMON UPDTE3                                                00012410
           C TH3          REAL    ROTATION OF DRUM CAM (DEG)                  00012420
           C DTH3         REAL    DERIVITIVE WRT TH2 OF TH3                   00012430
           C D2TH3        REAL    SECOND DER WRT TH2 OF TH3                   00012440
           C                                                                  00012450
           C TH2          REAL    ROTATION OF INPUT MOTOR GEAR (DEGREES)      00012460
           C DTH2         REAL    FIRST TIME DERIVITIVE OF TH2 (DEG/SEC)      00012470
           C D2TH2        REAL    SECOND TIME DERIVATIVE OF TH2 (DEG/SEC**2)  00012480
           C VEC(1)       REAL    ROTATION OF THE DRUM CAM(DEG)               00012490
           C    (2)       REAL    ANG VELOCITY OF DRUM CAM(DEG/SEC)           00012500
           C    (3)       REAL    ANG ACC OF DRUM CAM(DEG/SEC**2)             00012510
           C    (4)       REAL    TORQUE ACTING ON DRUM CAM (FT-LBS)          00012520
           C DG2RAD       REAL    CONVERS DEGREES TO RADIANS                  00012530
           C IDRUM        REAL    INERTIA MOMENT OF DRUM CAM  (FT-LB-SEC**2)  00012540
           C IFACE        REAL    INERTIA MOMENT OF THE FACE CAM (FT-LB-SEC**2) 00012550
           C ITOTAL       REAL    SUM OF FACE AND DRUM MOMENTS OF INERTIA (FT-LB-SE00012560
           C------------------------------------------------------------------00012570
           C                                                                  00012580
           C                                                                  00012590
           C---------- OBTAIN THE VEC PARAMETERS                              00012600
           C                                                                  00012610
0006             VEC(1)=TH3                                                   00012620
0007             VEC(2)=DTH3*DTH2                                             00012630
0008             VEC(3)=DTH3*D2TH2 + D2TH3*(DTH2**2)                          00012640
0009             VEC(4)=(IDRUM+IFACE)*VEC(3)*DG2RAD                           00012650
           C                                                                  00012660
           C                                                                  00012670
0010                RETURN                                                    00012680
0011                END                                                       00012690
```

```
0001          SUBROUTINE UPDT4(TH2,DTH2,D2TH2,VEC)                           00012700
        C                                                                   00012710
        C---------- ABSTRACT:  UPDT4 UPDATES SOME OF THE FEED PARAMETERS    00012720
        C---------- THE PARAMETERS ARE SHOWN IN THE VARIABLE LIST.          00012730
0002          DIMENSION VEC(1)                                              00012740
0003          COMMON/UPDTE4/TH4,DTH4,D2TH4                                  00012750
0004          REAL IPAWL/.00032958/,IROCK/.00002724/,ISHFT/.000018647/      00012760
0005          REAL MAMMO/.04255/,RE/.166667/,PE/.3000/                      00012770
0006          DATA DG2RAD/.0174533/                                         00012780
        C                                                                   00012790
        C-------------------------------------------------------------------00012800
        C   VARIABLE     TYPE       DESCRIPTION                             00012810
        C                                                                   00012820
        C---- COMMON UPDTE4                                                 00012830
        C TH4            REAL    ROTATION OF FEED PAWL (DEG)                00012840
        C DTH4           REAL    DERIVITIVE WRT TH2 OF TH4                  00012850
        C D2TH4          REAL    SECOND DER WRT TH2 OF TH4                  00012860
        C                                                                   00012870
        C TH2            REAL    ROTATION OF INPUT MOTOR GEAR (DEGREES)     00012880
        C DTH2           REAL    FIRST TIME DERIVITIVE OF TH2 (DEG/SEC)     00012890
        C D2TH2          REAL    SECOND TIME DERIVATIVE OF TH2 (DEG/SEC**2) 00012900
        C VEC(1)         REAL    ROTATION OF THE FEED PAWL(DEG)             00012910
        C    (2)         REAL    ANG VELOCITY OF FEED PAWL(DEG/SEC)         00012920
        C    (3)         REAL    ANG ACC OF FEED PAWL(DEG/SEC**2)           00012930
        C    (4)         REAL    FORCE ACCELERATING AMMO                    00012940
        C    (5)         REAL    TORQUE ACTING ON FEED ABOUT SHAFT          00012950
        C    (6)         REAL    FORCE ACTING ON ROLLER BEARING             00012960
        C ISHFT          REAL    INERTIA MOMENT OF FEED SHAFT (FT-LB-SEC**2)  00012970
        C IROCK          REAL    INERTIA MOMENT OF ROCKER PAWL (FT-LB-SEC**2) 00012980
        C IPAWL          REAL    INERTIA MOMENT OF FEED PAWL (FT-LB-SEC**2) 00012990
        C AMMO           REAL    MASS OF THE AMMO ROUND (LB-SEC**2/FT)      00013000
        C MROCK          REAL    MASS OF THE ROCKER ARM (LB-SEC**2/FT)      00013010
        C MPAWL          REAL    MASS OF THE FEED PAWL (LB-SEC**2/FT)       00013020
        C RE             REAL    LENGTH TO ROCKER END (FT)                  00013030
        C PE             REAL    LENGTH TO PAWL END (FT)                    00013040
        C DG2RAD         REAL    CONVERS DEGREES TO RADIANS                 00013050
        C IDRUM          REAL    INERTIA MOMENT OF DRUM CAM  (FT-LB-SEC**2) 00013060
        C IFACE          REAL    INERTIA MOMENT OF THE FACE CAM (FT-LB-SEC**2) 00013070
0007           CONTINUE                                                     00013080
        C ITOTAL         REAL    SUM OF FACE AND DRUM MOMENTS OF INERTIA (FT-LB-SE00013090
        C-------------------------------------------------------------------00013100
        C                                                                   00013110
        C---------- OBTAIN THE VEC PARAMETERS                               00013120
        C                                                                   00013130
0008          VEC(1)=TH4                                                    00013140
0009          VEC(2)=DTH4*DTH2                                              00013150
0010          VEC(3)=DTH4*D2TH2 + D2TH4*(DTH2**2)                           00013160
0011          VEC(4)=VEC(3)*DG2RAD*PE*MAMMO*UNIT(-DTH4)                     00013170
0012          VEC(5)=VEC(4)*PE + ITOTAL*VEC(3)*DG2RAD                       00013180
0013          VEC(6)=VEC(5)/RE                                              00013190
0014             RETURN                                                     00013200
0015             END                                                        00013210
```

```
0001          SUBROUTINE UPDTS(TH2,DTH2,D2TH2,VEC)
        C
        C---------- ABSTRACT: SUBROUTINE UPDTS UPDATES SOME OF THE EJECT
        C---------- PARAMETERS.  THE PARAMETERS ARE SHOWN IN THE
        C
        C---------- VARIABLE LIST.
        C
        C
        C-----------------------------------------------------------------
        C   VARIABLE     TYPE      DESCRIPTION
        C
        C---- COMMON UPDTES
        C TH5           REAL    ROTATION OF EJECT PAWL (DEG)
        C DTH5          REAL    DERIVITIVE WRT TH2 OF TH5
        C D2TH5         REAL    SECOND DER WRT TH2 OF TH5
        C
        C TH2           REAL    ROTATION OF INPUT MOTOR GEAR (DEGREES)
        C DTH2          REAL    FIRST TIME DERIVITIVE OF TH2 (DEG/SEC)
        C D2TH2         REAL    SECOND TIME DERIVATIVE OF TH2 (DEG/SEC**2)
        C VEC(1)        REAL    ROTATION OF THE EJECT PAWL(DEG)
        C    (2)        REAL    ANG VELOCITY OF EJECT PAWL(DEG/SEC)
        C    (3)        REAL    ANG ACC OF EJECT PAWL(DEG/SEC**2)
        C    (4)        REAL    TORQUE ACTING ON EJECT ABOUT SHAFT
        C    (5)        REAL    FORCE ACTING ON ROLLER BEARING
        C ISHFT         REAL    INERTIA MOMENT OF EJECT SHAFT (FT-LB-SEC**2)
        C IROCK         REAL    INERTIA MOMENT OF ROCKER PAWL (FT-LB-SEC**2)
        C IPAWL         REAL    INERTIA MOMENT OF EJECT PAWL (FT-LB-SEC**2)
        C MROCK         REAL    MASS OF THE ROCKER ARM (LB-SEC**2/FT)
0002           CONTINUE
        C MPAWL         REAL    MASS OF THE EJECT PAWL (LB-SEC**2/FT)
        C RE            REAL    LENGTH TO ROCKER END (FT)
        C DG2RAD        REAL    CONVERS DEGREES TO RADIANS
        C-----------------------------------------------------------------
        C
        C
0003          DIMENSION VEC(1)
0004          COMMON/UPDTES/TH5,DTH5,D2TH5
0005          REAL IPAWL/.00015703/,IROCK/.00002724/,ISHFT/.000016782/
0006          DATA ITHRU/1/,GC/986.088/,DG2RAD/.0174533/
0007          DATA RE/.166667/
        C
        C
        C---------- OBTAIN VEC PARAMETERS
        C
0008          VEC(1)=TH5
0009          VEC(2)=DTH5*DTH2
0010          VEC(3)=DTH5*D2TH2 + D2TH5*(DTH2**2)
0011          VEC(4)=VEC(3)*DG2RAD*ITOTAL
0012          VEC(5)=VEC(4)/RE
        C
        C
0013             RETURN
0014             END
```

```
0001            SUBROUTINE UPDT6(TH2,DTH2,D2TH2,VEC)
          C
          C---------- ABSTRACT: SUBROUTINE UPDT6 UPDATES SOME OF THE LOCK
          C---------- RING PARAMETERS.  THE PARAMETERS ARE SHOWN IN THE
          C---------- VARIABLE LIST.
          C
          C
0002            DIMENSION VEC(1)
0003            COMMON/UPDTES/TH6,DTH6,D2TH6
0004            REAL ILOCK/.0020653/,PIN/.16083/
0005            DATA DG2RAD/.0174533/
          C
          C
          C-----------------------------------------------------------------
          C   VARIABLE     TYPE      DESCRIPTION
          C
          C---- COMMON UPDTES
          C TH6          REAL    ROTATION OF LOCK (DEG)
          C DTH6         REAL    DERIVITIVE WRT TH2 OF TH6
          C D2TH6        REAL    SECOND DER WRT TH2 OF TH6
          C
          C TH2          REAL    ROTATION OF INPUT MOTOR GEAR (DEGREES)
          C DTH2         REAL    FIRST TIME DERIVITIVE OF TH2 (DEG/SEC)
          C D2TH2        REAL    SECOND TIME DERIVATIVE OF TH2 (DEG/SEC**2)
          C VEC(1)       REAL    ROTATION OF THE LOCK CAM(DEG)
          C    (2)       REAL    ANG VELOCITY OF LOCK CAM(DEG/SEC)
          C    (3)       REAL    ANG ACC OF LOCK CAM(DEG/SEC**2)
          C    (4)       REAL    TORQUE ACTING ON LOCK CAM
          C    (5)       REAL    FORCE ON THE ACTUATOR GEAR SET
          C DG2RAD       REAL    CONVERS DEGREES TO RADIANS
          C ILOCK        REAL    INERTIA MOMENT OF LOCK (FT-LB-SEC**2)
          C PIN          REAL    DISTANCE FROM LOCK CENTER TO GEARS(FT)
          C-----------------------------------------------------------------
          C
          C
          C
          C---------- OBTAIN VEC PARAMETERS
          C
0006            VEC(1)=TH6
0007            VEC(2)=DTH6*DTH2
0008            VEC(3)=DTH6*D2TH2 + D2TH6*(DTH2**2)
0009            VEC(4)=VEC(3)*DG2RAD*ILOCK
0010            VEC(5)=VEC(4)/PIN
          C
          C
0011              RETURN
0012              END
```

```
0001         SUBROUTINE UPDT7(TH2,DTH2,D2TH2,VEC)
        C
        C---------- ABSTRACT:  SUBROUTINE UPDT7 UPDATES SOME OF THE CHAMBER
        C---------- PARAMETERS. THE PARAMETERS ARE SHOWN IN THE
        C---------- VARIABLE LIST.
        C
0002         DIMENSION VEC(1)
0003         COMMON/UPDTE7/R7,DR7,D2R7
        C-----------------------------------------------------------------
        C   VARIABLE     TYPE      DESCRIPTION
        C
        C---- COMMON UPDTE7
        C R7            REAL    DISPLACEMENT OF CHAMBER (IN)
        C DR7           REAL    DERIVITIVE WRT TH5 OF R7
        C D2R7          REAL    SECOND DER WRT TH2 OF R7
        C
        C CRUSH          REAL    CRUSH FORCE EXERTED BY ROUND (LBS)
        C MBOLT          REAL    WEIGHT OF THE BOLT ASSY (LBS)
        C VMASS          REAL    THE ACTUAL TRANSLATING MASS
        C RESEAR         REAL    SEAR FORCE DUE RE-SEARING (LBS)
        C TH2            REAL    ROTATION OF INPUT MOTOR GEAR (DEGREES)
        C DTH2           REAL    FIRST TIME DERIVITIVE OF TH2 (DEG/SEC)
        C D2TH2          REAL    SECOND TIME DERIVATIVE OF TH2 (DEG/SEC**2)
        C VEC(1)         REAL    DISPLACEMENT OF THE CHAMBER (IN)
        C    (2)         REAL    VELOCITY OF CHAMBER(IN/SEC)
        C    (3)         REAL    ACC OF CHAMBER(IN/SEC**2)
        C    (4)         REAL    CRUSH FORCE EXERTED BY ROUND (LBS)
        C    (5)         REAL    SEAR FORCE DUE RE-SEARING (LBS)
        C    (6)         REAL    FORCE ON THE CHAMBER TO DRUM CAM  STUD (LBS)
        C DG2RAD         REAL    CONVERS DEGREES TO RADIANS
        C ROUND          REAL    NOMINAL UNCRUSHED LENGTH OF THE ROUND
0004          CONTINUE
        C HDSP           REAL    NOMINAL HEADSPACE OF THE WEAPON (FT)
        C CRATE          REAL    CRUSH SPRING RATE (LBS/FT)
        C SRATE          REAL    SEAR SPRING RATE (LBS/FT)
        C SPRE           REAL    SEAR SPRING PRELOAD (LBS)
        C-----------------------------------------------------------------
        C
        C---------- OBTAIN VEC PARAMETERS
        C
0005         VEC(1)=R7*12.0
0006         VEC(2)=DR7*DTH2*12.0
0007         VEC(3)=DR7*D2TH2 + D2R7*(DTH2**2)
        C
        C---------- GET ALL THE CURRENT CHAMBER INFORMATION
        C
0008         CALL CHAMBR(CRUSH,RESEAR,VMASS)
0009         VEC(4)=CRUSH
0010         VEC(5)=RESEAR
0011         VEC(6)=VEC(3)*VMASS
0012         VEC(3)=VEC(3)*12.0
0013            RETURN
0014            END
```

```
0001          SUBROUTINE CHAMBR(CRUSH,RESEAR,VMASS)                          00014750
        C                                                                    00014760
        C---------- ABSTRACT:  SUBROUTINE CHAMBR DETERMINES THE POSITION OF THE 00014770
        C---------- CHAMBER AND CALCULATES THE CRUSH UP AND RESEARING FORCEES   00014780
        C---------- THAT MIGHT BE ACTING ON THE CHAMBER.  THE ACTUAL TRANSLATING00014790
        C---------- MASS IS ALSO DETERMINED.                                 00014800
        C                                                                    00014810
0002          COMMON/UPDTE7/R7,DR7,D2R7                                      00014820
0003          COMMON/UPDTE3/TH3,DTH3,D2TH3                                   00014830
0004          REAL MCHMBR/.6211/,MBOLT/.2795/                                00014840
0005          DATA CRATE/280800./,SRATE/84.0/,SPRE/33.0/                     00014850
0006          DATA ROUND/.52625/,HDSP/.52417/                                00014860
        C                                                                    00014870
0007    10    CONTINUE                                                       00014880
        C                                                                    00014890
        C---------- CRUSHUP ON ROUND OCCURS ONLY WHEN TH3 IS -NOT- IN 33-250  00014900
        C---------- DEGREE RANGE OF DRUM CAM  ROTATION.                      00014910
        C                                                                    00014920
0008          IF(TH3.GT.33.  .AND.  TH3.LT.250.) GO TO 100                   00014930
0009             CRUSH=(ROUND-(R7+HDSP))*CRATE                               00014940
0010             IF(CRUSH.LT.0.0) GO TO 100                                  00014950
0011             GO TO 200                                                   00014960
0012    100   CONTINUE                                                       00014970
0013             CRUSH=0.0                                                   00014980
0014    200   CONTINUE                                                       00014990
        C                                                                    00015000
        C---------- RE-SEAR FORCE WILL ONLY OCCUR WHEN TH3  -IS- IN THE      00015010
        C---------- 150-250 DEGREE PORTION OF DRUM CAM  ROTATION.            00015020
        C                                                                    00015030
0015          IF(.NOT. (TH3.GT.150.  .AND.  TH3.LT.250.)) GO TO 300          00015040
        C                                                                    00015050
        C---------- RE-SEARING OCCURS ONLY DURING THE LAST INCH OF CHAMBER   00015060
        C---------- MOTION, FROM R7=6.6 TO R7=7.6, .550 TO .633333 FT        00015070
        C                                                                    00015080
0016             IF(R7.LT..550) GO TO 300                                    00015090
0017                RESEAR=SPRE+SRATE*(R7-.550)                              00015100
0018                GO TO 400                                                00015110
0019    300   CONTINUE                                                       00015120
0020             RESEAR=0.0                                                  00015130
0021    400   CONTINUE                                                       00015140
        C                                                                    00015150
        C---------- NOW DETERMINE THE VIRTUAL MASS OF THE TRANSLATING CHAMBER. 00015160
        C---------- THE BOLT ONLY TRAVELS ABOUT 1/2 INCH WITH THE CHAMBER    00015170
        C---------- WHEN R7 IS LESS THAN .5 INCH, .0417 FT                   00015180
        C                                                                    00015190
0022          IF(R7.GT..0417) GO TO 500                                      00015200
0023             VMASS=MCHMBR+MBOLT                                          00015210
0024             GO TO 600                                                   00015220
0025    500   CONTINUE                                                       00015230
0026             VMASS=MCHMBR                                                00015240
0027    600   CONTINUE                                                       00015250
0028             RETURN                                                      00015260
0029             END                                                         00015270
```

```
 0001          SUBROUTINE TBLOUT(IVAL,VEC2,VEC3,VEC4,VFC5,VEC6,VEC7,TH2,DTH2,
              1                    D2TH2)
           C
           C
           C
 0002          DIMENSION VEC2(1),VEC3(1),VEC4(1),VEC5(1),VEC6(1),VEC7(1)
 0003          DIMENSION V3(300,6),V4(300,6),V5(300,6),V6(300,6),V7(300,6)
 0004          REAL TIME(300),MOT(300),DMOT(300),D2MOT(300)
 0005          COMMON/TIMES/TINTI,TFINAL,TNOW,TOUT,TSTEP,TNEXT
 0006          DATA J/0/
           C
           C--------------------------------------------------------------------
           C   VARIABLE      TYPE      DESCRIPTION
           C IVAL            INTEGER .LT.5 THEN ADD TO TABLE,.GT.5 THEN OUTPUT TABLE
           C TH2             REAL    ROTATION OF INPUT MOTOR GEAR (DEGREES)
           C DTH2            REAL    FIRST TIME DERIVITIVE OF TH2 (DEG/SEC)
           C D2TH2           REAL    SECOND TIME DERIVATIVE OF TH2 (DEG/SEC**2)
           C    3(2)         REAL    ANG VELOCITY OF DRUM CAM (DEG/SEC)
           C    3(3)         REAL    ANG ACC OF DRUM CAM (DEG/SEC**2)
           C    3(4)         REAL    TORQUE ACTING ON DRUM CAM  (FT-LBS)
           C VEC4(1)         REAL    ROTATION OF THE FEED PAWL(DEG)
           C    4(2)         REAL    ANG VELOCITY OF FEED PAWL(DEG/SEC)
           C    4(3)         REAL    ANG ACC OF FEED PAWL(DEG/SEC**2)
           C    4(4)         REAL    FORCE ACCELERATING AMMO
           C    4(5)         REAL    TORQUE ACTING ON FEED ABOUT SHAFT
           C    4(6)         REAL    FORCE ACTING ON ROLLER BEARING
           C VEC5(1)         REAL    ROTATION OF THE EJECT PAWL(DEG)
           C    5(2)         REAL    ANG VELOCITY OF EJECT PAWL(DEG/SEC)
           C    5(3)         REAL    ANG ACC OF EJECT PAWL(DEG/SEC**2)
           C    5(4)         REAL    TORQUE ACTING ON EJECT ABOUT SHAFT
           C    5(5)         REAL    FORCE ACTING ON ROLLER BEARING
           C VEC6(1)         REAL    ROTATION OF THE LOCK CAM(DEG)
           C    6(2)         REAL    ANG VELOCITY OF LOCK CAM(DEG/SEC)
           C    6(3)         REAL    ANG ACC OF LOCK CAM(DEG/SEC**2)
           C    6(4)         REAL    TORQUE ACTING ON LOCK CAM
           C    6(5)         REAL    FORCE ON THE ACTUATOR GEAR SET
 0007            CONTINUE
           C VEC7(1)         REAL    DISPLACEMENT OF THE CHAMBER (IN)
           C    7(2)         REAL    VELOCITY OF CHAMBER(IN/SEC)
           C    7(3)         REAL    ACC OF CHAMBER(IN/SEC**2)
           C    7(4)         REAL    CRUSH FORCE EXERTED BY ROUND (LBS)
           C    7(5)         REAL    SEAR FORCE DUE RE-SEARING (LBS)
           C    7(6)         REAL    FORCE ON THE CHAMBER TO DRUM CAM STUD (LBS)
           C--------------------------------------------------------------------
           C
           C
 0008          IF(IVAL.GT.10) GO TO 200
 0009          J=J+1
 0010          DO 100 M=1,6
 0011            V3(J,M)=VEC3(M)
 0012            V4(J,M)=VEC4(M)
 0013            V5(J,M)=VEC5(M)
 0014            V6(J,M)=VEC6(M)
```

```
0015                 V7(J,M)=VEC7(M)                                                 00015930
0016         100   CONTINUE                                                          00015940
0017               TIME(J)=TNOW                                                      00015950
0018               MOT(J)=TH2                                                        00015960
0019               DMOT(J)=DTH2                                                      00015970
0020               D2MOT(J)=D2TH2                                                    00015980
             C                                                                       00015990
             C---------- IS TABLE COMPLETE? IF NOT, THEN RETURN                      00016000
             C                                                                       00016010
0021               IF(IVAL.LT.5) RETURN                                              00016020
             C                                                                       00016030
0022         200   CONTINUE                                                          00016040
             C                                                                       00016050
             C---------- IF HERE, OUTPUT THE TABLE                                   00016060
             C                                                                       00016070
             C---------- DRUM CAM                                                    00016080
             C                                                                       00016090
0023                WRITE(6,1000)                                                    00016100
0024                WRITE(6,3003)                                                    00016110
0025                DO 300 I=1,J                                                     00016120
0026                   WRITE(6,1001) I,TIME(I),MOT(I),V3(I,1),V3(I,2),V3(I,3),V3(I,4)00016130
0027         300    CONTINUE                                                         00016140
             C                                                                       00016150
             C---------- FEED MECHANISM                                              00016160
             C                                                                       00016170
0028                WRITE(6,1000)                                                    00016180
0029                WRITE(6,4003)                                                    00016190
0030                DO 400 I=1,J                                                     00016200
0031                  WRITE(6,1001) I,TIME(I),MOT(I),V4(I,1),V4(I,2),V4(I,3),        00016210
                2            V4(I,4),V4(I,5),V4(I,6)                                 00016220
0032         400   CONTINUE                                                          00016230
             C                                                                       00016240
             C---------- EJECT MECHANISM                                             00016250
             C                                                                       00016260
0033                WRITE(6,1000)                                                    00016270
0034                WRITE(6,5003)                                                    00016280
0035                DO 500 I=1,J                                                     00016290
0036                   WRITE(6,1001) I,TIME(I),MOT(I),V5(I,1),V5(I,2),V5(I,3),       00016300
                1             V5(I,4),V5(I,5)                                        00016310
0037         500    CONTINUE                                                         00016320
             C                                                                       00016330
             C---------- LOCK CAM                                                    00016340
             C                                                                       00016350
0038                WRITE(6,1000)                                                    00016360
0039                WRITE(6,6003)                                                    00016370
0040                DO 600 I=1,J                                                     00016380
0041                  WRITE(6,1001) I,TIME(I),MOT(I),V6(I,1),V6(I,2),V6(I,3),        00016390
                2            V6(I,4),V6(I,5)                                         00016400
0042         600   CONTINUE                                                          00016410
             C                                                                       00016420
             C---------- CHAMBER ASSEMBLY                                            00016430
             C                                                                       00016440
0043                WRITE(6,1000)                                                    00016450
```

```
0044              WRITE(6,7003)                                                         00016460
0045              DO 700 I=1,J                                                          00016470
0046                 WRITE(6,1001) I,TIME(I),MOT(I),V7(I,1),V7(I,2),V7(I,3),            00016480
               1          V7(I,4),V7(I,5),V7(I,6)                                       00016490
0047         700  CONTINUE                                                              00016500
             C                                                                          00016510
0048         1000 FORMAT('1')                                                           00016520
0049         1001 FORMAT(' ',I4,8F16.5)                                                 00016530
0050         3003 FORMAT(' ','DRUM CAM (DC) PARAMETERS',/,                              00016540
               2   8X,'TIME (SEC)       MOTOR (DEG)     DC ROTATION',5X,                00016550
               3   'DC VELOCITY    DC ACCELERATION   DC TORQUE',/,41X,                  00016560
               4   '(DEG)',9X,'(DEG/SEC)',7X,'(DEG/SEC**2)        (FT-LBS)',/)          00016570
0051         4003 FORMAT(' ','FEED MECHANISM (FM) PARAMETERS',/,                        00016580
               2   8X,'TIME (SEC)       MOTOR (DEG)     FM ROTATION',5X,                00016590
               3   'FM VELOCITY    FM ACCELERATION  SHAFT TORQUE',6X,                   00016600
               4   'CAM FORCE    FORCE ON AMMO',/,41X,                                  00016610
               5   '(DEG)',9X,'(DEG/SEC)',7X,'(DEG/SEC**2)        (FT-LBS)',9X,         00016620
               6   '(LBS)',11X,'(LBS)',/)                                               00016630
0052         5003 FORMAT(' ','EJECT MECHANISM (EM) PARAMETERS',/,                       00016640
               2   8X,'TIME (SEC)       MOTOR (DEG)     EM ROTATION',5X,                00016650
               3   'EM VELOCITY    EM ACCELERATION  SHAFT TORQUE',6X,                   00016660
               4   'CAM FORCE    ',/,41X,                                               00016670
               5   '(DEG)',9X,'(DEG/SEC)',7X,'(DEG/SEC**2)        (FT-LBS)',9X,         00016680
               6   '(LBS)',/)                                                           00016690
0053         6003 FORMAT(' ','LOCK MECHANISM (LM) PARAMETERS',/,                        00016700
               2   8X,'TIME (SEC)       MOTOR (DEG)     LM ROTATION',5X,                00016710
               3   'LM VELOCITY    LM ACCELERATION   LOCK TORQUE',6X,                   00016720
               4   'GEAR FORCE    ',/,41X,                                              00016730
               5   '(DEG)',9X,'(DEG/SEC)',7X,'(DEG/SEC**2)        (FT-LBS)',9X,         00016740
               6   '(LBS)',/)                                                           00016750
0054         7003 FORMAT(' ','CHAMBER ASSEMBLY (CA) PARAMETERS',/,                      00016760
               2   8X,'TIME (SEC)       MOTOR (DEG)     CA   DISPL   ',5X,              00016770
               3   'CA VELOCITY    CA ACCELERATION  CRUSH FORCE',4X,                    00016780
               4   'RESEAR FORCE     STUD FORCE',/,41X,                                 00016790
               5   '(DEG)',9X,'(DEG/SEC)',7X,'(DEG/SEC**2)        (FT-LBS)',9X,         00016800
               6   '(LBS)',11X,'(LBS)',/)                                               00016810
0055              RETURN                                                                00016820
0056              END                                                                   00016830
```

```
0001          SUBROUTINE INTERP(XV,YV,NPTS,X,YINTER,DYDX,D2YDX2,ISW,IWRAP)
        C
        C---------- ABSTRACT: SUBROUTINE INTERP SEARCHES THE
        C----------     XV AND YV VECTORS AND EXTRACTS 6
        C----------     LOCAL DATA POINTS.  THESE ARE PASSED TO
        C----------     THE POLATE ROUTINE THAT FITS A 5TH
        C----------     DEGREE POLYNOMINAL TO THEM.
        C
        C
0002          DIMENSION XV(1),YV(1),XPOLY(6),YPOLY(6),COEFF(6)
0003          REAL OFFSET
        C
        C
        C---------- DETERMINE STEP SIZE BETWEEN ADJACENT X(I)
        C---------- THEN ESTABLISH THE "LOCAL" REGION IN ARRAYS
        C
0004          IF(NPTS .LT. 6)RETURN
0005          M=5
0006          N=6
0007          XVAL=X
0008          STEP=XV(2)-XV(1)
0009          NSTEP=IFIX(XVAL/STEP + .5)
0010          IF( .NOT. (NSTEP.LT.5)) GO TO 20
0011              NSTEP=4
0012              OFFSET=0.0
0013              GO TO 50
0014    20    CONTINUE
0015          IF( .NOT. ((NPTS-2).LT.NSTEP)) GO TO 40
0016              NSTEP=NPTS-2
0017              OFFSET=XV(NSTEP-4)
0018              GO TO 50
0019    40    CONTINUE
0020              OFFSET=XV(NSTEP-4)
0021    50    CONTINUE
        C
        C---------- SET UP THE 6 ELEMENT ARRAYS TO BE FIT
        C
0022          DO 100 J=1,6
0023              JVAL=NSTEP-4+J
0024              XPOLY(J)=XV(JVAL)-OFFSET
0025              YPOLY(J)=YV(JVAL)
0026    100   CONTINUE
0027          XVAL=XVAL-OFFSET
        C
        C---------- ARRAYS ARE NOW TO BE FIT
        C
0028          CALL POLATE(XPOLY,YPOLY,COEFF)
        C
        C---------- EVALUATE POLY FOR 0,1,2 DERIVITIVES
        C---------- AT VALUE XVAL BASED ON ISW
        C
0029          P=COEFF(6)
0030          DO 200 J=1,5
```

```
0031                I=5-J                                      00017370
0032                P=P*XVAL+COEFF(I+1)                        00017380
0033        200  CONTINUE                                      00017390
            C                                                  00017400
0034             IF(ISW.LE.1) GO TO 9999                       00017410
            C                                                  00017420
0035             DO 300 J=1,5                                  00017430
0036                COEFF(J)=J*COEFF(J+1)                      00017440
0037        300  CONTINUE                                      00017450
            C                                                  00017460
0038             DP=COEFF(5)                                   00017470
0039             DO 400 J=1,4                                  00017480
0040                I=4-J                                      00017490
0041                DP=DP*XVAL+COEFF(I+1)                      00017500
0042        400  CONTINUE                                      00017510
            C                                                  00017520
0043             DO 500 J=1,4                                  00017530
0044                COEFF(J)=J*COEFF(J+1)                      00017540
0045        500  CONTINUE                                      00017550
            C                                                  00017560
0046             D2P=COEFF(4)                                  00017570
0047             DO 600 J=1,3                                  00017580
0048                I=3-J                                      00017590
0049                D2P=D2P*XVAL+COEFF(I+1)                    00017600
0050        600  CONTINUE                                      00017610
            C                                                  00017620
            C                                                  00017630
0051             DYDX=DP                                       00017640
0052             D2YDX2=D2P                                    00017650
            C                                                  00017660
0053        9999 CONTINUE                                      00017670
            C                                                  00017680
0054             YINTER=P                                      00017690
            C                                                  00017700
            C                                                  00017710
0055                RETURN                                     00017720
0056                END                                        00017730
```

```
0001             SUBROUTINE POLATE(XV,YV,COEFF)                                      00017620
           C                                                                         00017630
           C---------- ABSTRACT: SUBROUTINE POLATE PREFORMS A                        00017640
           C----------     POLYNOMINAL REGRESSION ON THE                             00017650
           C----------     DATA POINTS IN VECTORS XY,YV AND                          00017660
           C----------     RETURNS A COEFFICIENT VECTOR.  THE                        00017670
           C----------     DEGREE OF THE POLY WILL BE 5.                             00017680
           C                                                                         00017690
0002             DIMENSION XV(1),YV(1),COEFF(1)                                      00017700
0003             DIMENSION FIT(36),FITINV(36),DCOEFF(6),DYPOLY(6)                    00017710
0004             DOUBLE PRECISION FIT,FITINV,DCOEFF,DYPOLY                           00017720
           C                                                                         00017730
           C                                                                         00017740
           C---------- INITIALIZE FIT MATRIX FOR CURVE FITTING                       00017750
           C                                                                         00017760
0005             DO 100 I=1,6                                                        00017770
0006                FIT(I)=1.                                                        00017780
0007                FIT(I+6)=XV(I)                                                   00017790
0008                FIT(I+12)=XV(I)**2                                               00017800
0009                FIT(I+18)=XV(I)**3                                               00017810
0010                FIT(I+24)=XV(I)**4                                               00017820
0011                FIT(I+30)=XV(I)**5                                               00017830
0012                DYPOLY(I)=YV(I)                                                  00017840
0013       100   CONTINUE                                                            00017850
           C                                                                         00017860
           C---------- ZERO THE INVERSE MATRIX THEN MAKE IT IDENITY                  00017870
           C                                                                         00017880
0014             DO 200 I=2,35                                                       00017890
0015                FITINV(I)=0.0                                                    00017900
0016       200   CONTINUE                                                            00017910
           C                                                                         00017920
0017                FITINV(1)=1.                                                     00017930
0018                FITINV(8)=1.                                                     00017940
0019                FITINV(15)=1.                                                    00017950
0020                FITINV(22)=1.                                                    00017960
0021                FITINV(29)=1.                                                    00017970
0022                FITINV(36)=1.                                                    00017980
           C                                                                         00017990
           C---------- INVERT THE FIT MATRIX USING GELG                              00018000
           C                                                                         00018010
0023             CALL CGELG(FITINV,FIT,6,6,1.E-07,IERR)                              00018020
           C                                                                         00018030
           C---------- DETERMINE THE POLY COEFFICIENTS                               00018040
           C                                                                         00018050
0024             CALL DGMPRD(FITINV,DYPOLY,DCOEFF,6,6,1)                             00018060
           C                                                                         00018070
           C---------- TRANSPOSE DOUBLE PRECISION COEFF TO SINGLE                    00018080
           C                                                                         00018090
0025             DO 300 I=1,6                                                        00018100
0026                COEFF(I)=DCOEFF(I)                                               00018110
0027       300   CONTINUE                                                            00018120
0028                RETURN                                                           00018130
0029                END                                                              00018140
```

```
              C      SUBROUTINE GMPRD(A,B,R,N,M,L)                                       00018300
              C                                                                          00018310
              C         PURPOSE                                                          00018320
              C            MULTIPLY TWO GENERAL MATRICES TO FORM A RESULTANT GENERAL     00018330
              C            MATRIX                                                        00018340
              C                                                                          00018350
              C         USAGE                                                            00018360
              C            CALL GMPRD(A,B,R,N,M,L)                                       00018370
              C                                                                          00018380
              C         DESCRIPTION OF PARAMETERS                                        00018390
              C            A - NAME OF FIRST INPUT MATRIX                                00018400
              C            B - NAME OF SECOND INPUT MATRIX                               00018410
              C            R - NAME OF OUTPUT MATRIX                                     00018420
              C            N - NUMBER OF ROWS IN A                                       00018430
              C            M - NUMBER OF COLUMNS IN A AND ROWS IN B                      00018440
              C            L - NUMBER OF COLUMNS IN B                                    00018450
              C                                                                          00018460
              C         REMARKS                                                          00018470
              C            ALL MATRICES MUST BE STORED AS GENERAL MATRICES               00018480
              C            MATRIX R CANNOT BE IN THE SAME LOCATION AS MATRIX A           00018490
              C            MATRIX R CANNOT BE IN THE SAME LOCATION AS MATRIX B           00018500
              C            NUMBER OF COLUMNS OF MATRIX A MUST BE EQUAL TO NUMBER OF ROW00018510
              C            OF MATRIX B                                                   00018520
              C                                                                          00018530
              C         SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED                    00018540
              C            NONE                                                          00018550
              C                                                                          00018560
              C         METHOD                                                           00018570
              C            THE M BY L MATRIX B IS PREMULTIPLIED BY THE N BY M MATRIX A  00018580
              C            AND THE RESULT IS STORED IN THE N BY L MATRIX R.              00018590
              C                                                                          00018600
              C      ..................................................................00018610
 0001                SUBROUTINE GMPRD(A,B,R,N,M,L)                                       00018620
              C                                                                          00018630
 0002                IMPLICIT REAL*8 (A-H,O-Z)                                           00018640
 0003                DIMENSION A(1),B(1),R(1)                                            00018650
              C                                                                          00018660
 0004                IR=0                                                                00018670
 0005                IK=-M                                                               00018680
 0006                DO 10 K=1,L                                                         00018690
 0007                IK=IK+M                                                             00018700
 0008                DO 10 J=1,N                                                         00018710
 0009                IR=IR+1                                                             00018720
 0010                JI=J-N                                                              00018730
 0011                IB=IK                                                               00018740
 0012                R(IR)=0                                                             00018750
 0013                DO 10 I=1,M                                                         00018760
 0014                JI=JI+N                                                             00018770
 0015                IB=IB+1                                                             00018780
 0016             10 R(IR)=R(IR)+A(JI)*B(IB)                                             00018790
 0017                RETURN                                                              00018800
 0018                END                                                                 00018810
```

```
          C      SUBROUTINE DGMPRD(A,B,R,N,M,L)
          C
          C         PURPOSE
          C            MULTIPLY TWO GENERAL MATRICES TO FORM A RESULTANT GENERAL
          C            MATRIX
          C
          C         USAGE
          C            CALL GMPRD(A,B,R,N,M,L)
          C
          C         DESCRIPTION OF PARAMETERS
          C            A - NAME OF FIRST INPUT MATRIX
          C            B - NAME OF SECOND INPUT MATRIX
          C            R - NAME OF OUTPUT MATRIX
          C            N - NUMBER OF ROWS IN A
          C            M - NUMBER OF COLUMNS IN A AND ROWS IN B
          C            L - NUMBER OF COLUMNS IN B
          C
          C         REMARKS
          C            ALL MATRICES MUST BE STORED AS GENERAL MATRICES
          C            MATRIX R CANNOT BE IN THE SAME LOCATION AS MATRIX A
          C            MATRIX R CANNOT BE IN THE SAME LOCATION AS MATRIX B
          C            NUMBER OF COLUMNS OF MATRIX A MUST BE EQUAL TO NUMBER OF ROW
          C            OF MATRIX B
          C
          C         SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
          C            NONE
          C
          C         METHOD
          C            THE M BY L MATRIX B IS PREMULTIPLIED BY THE N BY M MATRIX A
          C            AND THE RESULT IS STORED IN THE N BY L MATRIX R.
          C      ..................................................................
0001             SUBROUTINE DGMPRD(A,B,R,N,M,L)
          C
0002             IMPLICIT REAL*8 (A-H,O-Z)
0003             DIMENSION A(1),B(1),R(1)
0004             IR=0
0005             IK=-M
0006             DO 10 K=1,L
0007                IK=IK+M
0008                DO 20 J=1,N
0009                   IR=IR+1
0010                   JI=J-N
0011                   IB=IK
0012                   R(IR)=0.D+00
0013                   DO 30 I=1,M
0014                      JI=JI+N
0015                      IB=IB+1
0016                      R(IR)=R(IR)+A(JI)*B(IB)
0017         30        CONTINUE
0018         20     CONTINUE
0019         10  CONTINUE
0020                RETURN
0021                END
```

```
      C      SUBROUTINE DSELG(R,A,M,N,EPS,IER)                            00019360
      C                                                                   00019370
      C         PURPOSE                                                   00019380
      C            TO SOLVE A GENERAL SYSTEM OF SIMULTANEOUS LINEAR EQUATIONS. 00019390
      C                                                                   00019400
      C         USAGE                                                     00019410
      C            CALL DSELG(R,A,M,N,EPS,IER)                            00019420
      C                                                                   00019430
      C         DESCRIPTION OF PARAMETERS                                 00019440
      C            R      - DOUBLE PRECISION M BY N RIGHT HAND SIDE MATRIX 00019450
      C                     (DESTROYED). ON RETURN R CONTAINS THE SOLUTIONS 00019460
      C                     OF THE EQUATIONS.                             00019470
      C            A      - DOUBLE PRECISION M BY M COEFFICIENT MATRIX    00019480
      C                     (DESTROYED).                                  00019490
      C            M      - THE NUMBER OF EQUATIONS IN THE SYSTEM.        00019500
      C            N      - THE NUMBER OF RIGHT HAND SIDE VECTORS.        00019510
      C            EPS    - SINGLE PRECISION INPUT CONSTANT WHICH IS USED AS 00019520
      C                     RELATIVE TOLERANCE FOR TEST ON LOSS OF        00019530
      C                     SIGNIFICANCE.                                 00019540
      C            IER    - RESULTING ERROR PARAMETER CODED AS FOLLOWS    00019550
      C                     IER=0  - NO ERROR,                            00019560
      C                     IER=-1 - NO RESULT BECAUSE OF M LESS THAN 1 OR 00019570
      C                              PIVOT ELEMENT AT ANY ELIMINATION STEP 00019580
      C                              EQUAL TO 0,                          00019590
      C                     IER=K  - WARNING DUE TO POSSIBLE LOSS OF SIGNIFI- 00019600
      C                              CANCE INDICATED AT ELIMINATION STEP K+1, 00019610
      C                              WHERE PIVOT ELEMENT WAS LESS THAN OR 00019620
      C                              EQUAL TO THE INTERNAL TOLERANCE EPS TIMES 00019630
      C                              ABSOLUTELY GREATEST ELEMENT OF MATRIX A. 00019640
      C                                                                   00019650
      C         REMARKS                                                   00019660
      C            INPUT MATRICES R AND A ARE ASSUMED TO BE STORED COLUMNWISE 00019670
      C            IN M*N RESP. M*M SUCCESSIVE STORAGE LOCATIONS. ON RETURN 00019680
      C            SOLUTION MATRIX R IS STORED COLUMNWISE TOO.            00019690
      C            THE PROCEDURE GIVES RESULTS IF THE NUMBER OF EQUATIONS M IS 00019700
      C            GREATER THAN 0 AND PIVOT ELEMENTS AT ALL ELIMINATION STEPS 00019710
      C            ARE DIFFERENT FROM 0. HOWEVER WARNING IER=K - IF GIVEN - 00019720
      C            INDICATES POSSIBLE LOSS OF SIGNIFICANCE. IN CASE OF A WELL 00019730
      C            SCALED MATRIX A AND APPROPRIATE TOLERANCE EPS, IER=K MAY BE 00019740
      C            INTERPRETED THAT MATRIX A HAS THE RANK K. NO WARNING IS 00019750
      C            GIVEN IN CASE M=1.                                     00019760
      C                                                                   00019770
      C         SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED             00019780
      C            NONE                                                   00019790
      C                                                                   00019800
      C         METHOD                                                    00019810
      C            SOLUTION IS DONE BY MEANS OF GAUSS-ELIMINATION WITH    00019820
      C            COMPLETE PIVOTING.                                     00019830
      C                                                                   00019840
      C      ..............................................................00019850
0001         SUBROUTINE DSELG(R,A,M,N,EPS,IER)                            00019860
      C                                                                   00019870
      C                                                                   00019880
```

```
           C                                                           00019890
 0002            DIMENSION A(1),R(1)                                   00019900
 0003            DOUBLE PRECISION R,A,PIV,TB,TOL,PIVI                  00019910
 0004            IF(M)23,23,1                                          00019920
           C                                                           00019930
           C     SEARCH FOR GREATEST ELEMENT IN MATRIX A               00019940
 0005          1 IER=0                                                 00019950
 0006            PIV=0.D0                                              00019960
 0007            MM=M*M                                                00019970
 0008            NM=N*M                                                00019980
 0009            DO 3 L=1,MM                                           00019990
 0010            TB=DABS(A(L))                                         00020000
 0011            IF(TB-PIV)3,3,2                                       00020010
 0012          2 PIV=TB                                                00020020
 0013            I=L                                                   00020030
 0014          3 CONTINUE                                              00020040
 0015            TOL=EPS*PIV                                           00020050
           C     A(I) IS PIVOT ELEMENT. PIV CONTAINS THE ABSOLUTE VALUE OF A(I).  00020060
           C                                                           00020070
           C                                                           00020080
           C     START ELIMINATION LOOP                                00020090
 0016            LST=1                                                 00020100
 0017            DO 17 K=1,M                                           00020110
           C                                                           00020120
           C     TEST ON SINGULARITY                                   00020130
 0018            IF(PIV)23,23,4                                        00020140
 0019          4 IF(IER)7,5,7                                          00020150
 0020          5 IF(PIV-TOL)6,6,7                                      00020160
 0021          6 IER=K-1                                               00020170
 0022          7 PIVI=1.D0/A(I)                                        00020180
 0023            J=(I-1)/M                                             00020190
 0024            I=I-J*M-K                                             00020200
 0025            J=J+1-K                                               00020210
           C     I+K IS ROW-INDEX, J+K COLUMN-INDEX OF PIVOT ELEMENT   00020220
           C                                                           00020230
           C     PIVOT ROW REDUCTION AND ROW INTERCHANGE IN RIGHT HAND SIDE R  00020240
 0026            DO 8 L=K,NM,M                                         00020250
 0027            LL=L+I                                                00020260
 0028            TB=PIVI*R(LL)                                         00020270
 0029            R(LL)=R(L)                                            00020280
 0030          8 R(L)=TB                                               00020290
           C                                                           00020300
           C     IS ELIMINATION TERMINATED                             00020310
 0031            IF(K-M)9,18,18                                        00020320
           C                                                           00020330
           C     COLUMN INTERCHANGE IN MATRIX A                        00020340
 0032          9 LEND=LST+M-K                                          00020350
 0033            IF(J)12,12,10                                         00020360
 0034         10 II=J*M                                                00020370
 0035            DO 11 L=LST,LEND                                      00020380
 0036            TB=A(L)                                               00020390
 0037            LL=L+II                                               00020400
 0038            A(L)=A(LL)                                            00020410
```

```
 0039          11 A(LL)=TB                                                  00020420
           C                                                                00020430
           C      ROW INTERCHANGE AND PIVOT ROW REDUCTION IN MATRIX A       00020440
 0040          12 DO 13 L=LST,MM,M                                          00020450
 0041             LL=L+I                                                    00020460
 0042             TB=PIVI*A(LL)                                             00020470
 0043             A(LL)=A(L)                                                00020480
 0044          13 A(L)=TB                                                   00020490
           C                                                                00020500
           C      SAVE COLUMN INTERCHANGE INFORMATION                       00020510
 0045             A(LST)=J                                                  00020520
           C                                                                00020530
           C      ELEMENT REDUCTION AND NEXT PIVOT SEARCH                   00020540
 0046             PIV=0.D0                                                  00020550
 0047             LST=LST+1                                                 00020560
 0048             J=0                                                       00020570
 0049             DO 16 II=LST,LEND                                         00020580
 0050             PIVI=-A(II)                                               00020590
 0051             IST=II+M                                                  00020600
 0052             J=J+1                                                     00020610
 0053             DO 15 L=IST,MM,M                                          00020620
 0054             LL=L-J                                                    00020630
 0055             A(L)=A(L)+PIVI*A(LL)                                      00020640
 0056             TB=DABS(A(L))                                             00020650
 0057             IF(TB-PIV)15,15,14                                        00020660
 0058          14 PIV=TB                                                    00020670
 0059             I=L                                                       00020680
 0060          15 CONTINUE                                                  00020690
 0061             DO 16 L=K,NM,M                                            00020700
 0062             LL=L+J                                                    00020710
 0063          16 R(LL)=R(LL)+PIVI*R(L)                                     00020720
 0064          17 LST=LST+M                                                 00020730
           C      END OF ELIMINATION LOOP                                   00020740
           C                                                                00020750
           C                                                                00020760
           C      BACK SUBSTITUTION AND BACK INTERCHANGE                    00020770
 0065          18 IF(M-1)23,22,19                                           00020780.
 0066          19 IST=MM+M                                                  00020790
 0067             LST=M+1                                                   00020800
 0068             DO 21 I=2,M                                               00020810
 0069             II=LST-I                                                  00020820
 0070             IST=IST-LST                                               00020830
 0071             L=IST-M                                                   00020840
 0072             L=A(L)+.5D0                                               00020850
 0073             DO 21 J=II,NM,M                                           00020860
 0074             TB=R(J)                                                   00020870
 0075             LL=J                                                      00020880
 0076             DO 20 K=IST,MM,M                                          00020890
 0077             LL=LL+1                                                   00020900
 0078          20 TB=TB-A(K)*R(LL)                                          00020910
 0079             K=J+L                                                     00020920
 0080             R(J)=R(K)                                                 00020930
 0081          21 R(K)=TB                                                   00020940
```

```
0082          22 RETURN                                              00020950
            C                                                        00020960
            C                                                        00020970
            C        ERROR RETURN                                    00020980
0083          23 IER=-1                                              00020990
0084             RETURN                                              00021000
0085             END                                                 00021010
```

θ_2 (DEG)	$\dot{\theta}_2$ (D/S)	$\ddot{\theta}_2$ (D/S²)	TIME (SEC)
0.0	0.0	59096.7617	0.0
2.95483	590.96704	59096.7617	0.01000
11.81933	1181.93335	59096.7617	0.02000
26.59344	1772.89795	59096.7148	0.03000
47.27733	2362.58423	58730.9687	1.04006
73.82103	2945.10229	57901.1680	0.05000
106.15965	3520.11572	56920.0703	0.06000
144.19530	4083.24780	55615.5430	0.07000
187.77628	4631.78125	54075.6992	0.08000
235.77397	5181.73828	57502.6211	0.09000
291.69507	5749.07812	51495.2461	0.10000
366.80444	6384.75000	48803.3750	0.11250
450.43066	6982.94141	46810.2695	0.12500
522.57178	7443.44531	45264.7539	0.13500
639.27368	8102.93359	42436.7109	0.15000
722.36401	8523.12500	42221.6641	0.16000
809.78125	8941.42969	40695.0625	0.17000
901.08569	9335.79297	38505.7852	0.18000
996.39502	9710.42187	36218.4883	0.19000
1095.25439	10063.9648	34760.8867	0.20000
1197.64966	10404.8594	33254.5547	0.21000
1303.29712	10729.5586	31715.9180	0.22000
1412.16650	11039.6328	30062.2422	0.23000
1524.01587	11331.4258	28572.3477	0.24000
1638.74756	11609.9102	27102.1055	0.25000
1756.16821	11873.7031	25676.8203	0.26000
1876.16943	12123.6094	24316.6719	0.27000
1998.59619	12359.6523	22859.0508	0.28000
2123.30615	12581.0898	21469.0039	0.29000
2250.17261	12789.5234	20261.5820	0.30000
2379.06226	12985.0039	18993.6055	0.31000
2509.84375	13169.7930	17802.5508	0.32000
2642.41748	13341.7461	16561.7617	0.33000
2776.63525	13501.8203	15526.0898	0.34000
2912.42310	13651.9844	14470.9336	0.35000
3049.63867	13791.7500	13525.4414	0.36000
3188.22461	13922.6445	12560.8203	0.37000
3328.06543	14044.9414	11791.6133	0.38000
3469.08911	14153.6992	10981.5156	0.39000
3611.21289	14264.4023	10147.7266	0.40000
3754.34692	14361.8203	9356.73047	0.41000
3898.42261	14450.3281	8251.36719	0.42000
4043.30591	14529.6992	7501.77344	0.43000
4188.99437	14601.6250	6598.08984	0.44000
4335.30469	14665.8555	5875.50391	0.45000
4482.23828	14718.6680	4584.28516	0.46000
4629.63281	14764.5195	4915.93359	0.47000
4777.57812	14813.3828	4449.51562	0.48000
4925.84766	14857.7773	4589.20703	0.49000
5974.73437	14895.6797	2164.48926	0.50000
5149.21875	14905.3320	1779.16064	0.50500
5261.32031	14963.1211	12550.6406	0.51250
5411.26562	15019.0859	3983.45190	0.52250
5599.14453	15064.5781	3438.36060	0.53500
5749.87891	15060.2891	-9702.39062	0.54500
5900.17578	15034.6406	14209.6641	0.55500

6013.33203	15123.3008	5642.34375	0.56250
6164.80078	15169.0391	4091.80737	0.57250
6316.66016	15202.8437	3327.51099	0.58250
6468.83984	15233.6602	2809.43188	0.59250
6659.48437	15270.4844	2151.84839	0.60500
6773.96484	15253.2266	2571.46582	0.61250
6926.94922	15315.2227	2893.30566	0.62250
7118.53125	15348.7187	2758.30078	0.63500
7272.19531	15374.3437	2539.82983	0.64500
7426.05859	15400.1211	2391.57129	0.65500
7657.28906	15405.6289	-4325.06641	0.67000
7734.26953	15390.8242	2345.00244	0.67500
7888.37500	15432.0391	2424.74146	0.68500
8042.83984	15454.7695	2279.46948	0.69500
8197.47266	15475.6172	2175.76758	0.70500
8429.94531	15504.1055	1779.77808	0.72000
8585.06641	15520.3945	1456.92187	0.73000
8740.32812	15533.1602	1147.19214	0.74000
8895.69922	15544.5391	1224.47754	0.75000
9051.19531	15556.6211	1083.84595	0.76000
9206.77734	15565.4805	952.30737	0.77000
9362.47266	15573.9951	711.32397	0.78000
9518.23047	15580.7187	644.03491	0.79000
9674.05859	15586.4141	463.31714	0.80000
9829.92578	15590.4609	374.09595	0.81000
9985.84375	15593.7891	307.94873	0.82000
10141.7812	15595.6289	248.13017	0.83000
10297.7500	15599.4180	361.26001	0.84000
10453.7578	15602.3008	125.27588	0.85000
10609.7539	15603.4492	167.31493	0.86000
10765.8125	15603.3164	-257.69604	0.87000
10921.7773	15601.2344	-27.15523	0.88000
11077.8203	15600.1523	-219.96823	0.89000
11233.7461	15594.9102	-1065.71289	0.90000
11389.6211	15580.9531	-1515.16455	0.91000
11545.3555	15568.0195	-774.97534	0.92000
11701.0273	15564.0898	-156.42354	0.93000
11856.6211	15563.0703	-277.51343	0.94000
12012.2187	15561.0039	76.88417	0.95000
12167.8555	15590.3320	7674.27734	0.96000
12246.4687	15561.4453	715.29469	0.96500
12343.7461	15558.8203	394.79175	0.97125
12518.7617	15565.8398	875.53198	0.98250
12674.2852	15522.6992	-13599.5117	0.99250
12829.2266	15531.3008	10709.9453	1.00250
12984.8867	15593.2031	643.70117	1.01250
13140.6602	15587.6242	-171.56885	1.02249
13335.4492	15586.5391	-42.19145	1.03499
13491.3047	15589.5352	323.22485	1.04499
13647.1562	15568.0234	-9955.09375	1.05499
13763.8477	15589.1445	575.41577	1.06249
13919.6541	15585.9297	279.61035	1.07248
14114.4258	15590.5234	303.75122	1.08498
14270.3242	15592.5117	162.97150	1.09498
14504.1875	15588.6328	-1955.22217	1.10998
14659.8672	15560.9844	-3365.28613	1.11998
14815.3594	15541.8320	1218.02954	1.12998
14971.0585	15560.4375	958.34106	1.13998
15126.3359	15565.7539	781.02637	1.14998
15282.2930	15569.3477	801.11890	1.15998

15437.8828	15578.2969	694.05469	1.16998
15593.7461	15583.8984	522.41968	1.17997
15749.5703	15589.0039	419.08887	1.18997
15905.4648	15593.1602	357.86182	1.19997
16061.3984	15595.3086	260.78198	1.20997
16217.3555	15598.5586	198.06180	1.21997
16373.3437	15600.2539	169.48607	1.22997
16529.3398	15601.7734	133.93829	1.23997
16685.3477	15602.3750	-47.43010	1.24997
16841.3516	15602.2773	120.97285	1.25997
16997.3828	15603.7656	154.31264	1.26997
17153.4023	15605.1914	57.68726	1.27997
17309.4453	15605.4102	54.18924	1.28997
17465.4922	15605.1641	107.53809	1.29997
17621.5430	15607.0156	17.76927	1.30997
17777.5977	15603.0820	-1035.42163	1.31996
17933.5195	15595.1875	50.06320	1.32996
18089.5859	15594.0820	-588.69531	1.33996
18245.3203	15585.4102	-1176.40430	1.34996
18401.2070	15570.1211	-1532.37451	1.35996
18556.7617	15557.7969	-756.66040	1.36996
18712.3516	15555.4883	326.03296	1.37996
18867.8906	15558.4219	-276.51196	1.38996
19023.3867	15590.7031	9579.98437	1.39996
19101.3008	15644.9883	4818.26172	1.40496
19218.5000	15647.5469	-1379.20557	1.41246
19374.8572	15631.9102	-1335.03564	1.42245
19570.0273	15573.4375	-13992.7031	1.43495
19725.3555	15523.9570	11791.2500	1.44495
19842.0781	15585.4336	2271.66992	1.45245
19997.9766	15598.5234	675.99512	1.46245
20153.9023	15598.2031	55.80971	1.47244
20309.8320	15597.3867	103.96130	1.48244
20465.7461	15595.7656	-455.89819	1.49244
20621.5078	15589.8828	32263.1016	1.50244
20757.9609	15625.0898	-503.02954	1.51118
20914.1133	15620.0312	-473.96021	1.52118
21070.2578	15615.1562	-388.02441	1.53118
21226.3828	15612.7383	-273.53687	1.54118
21460.5156	15605.1602	-137.74188	1.55618
21616.5664	15619.6367	3979.78638	1.55617
21773.1328	15643.3398	-822.71094	1.57617
21851.1875	15647.2305	-1037.47021	1.58117
22007.5352	15637.3789	-732.57544	1.59117
22163.8867	15629.2383	-532.70508	1.60117
22320.1328	15624.7344	-386.00928	1.61117
22554.5156	15617.5039	-285.54517	1.62517
22710.5484	15615.2656	-174.32469	1.63517
22866.7969	15613.9766	-68.35913	1.64616
23022.9102	15613.3711	-131.71103	1.65616
23179.0117	15612.2305	-105.91069	1.66616
23335.1250	15611.3398	-45.75533	1.67516
23491.2227	15610.9219	-78.09987	1.68516
23647.3164	15610.0586	-69.10861	1.69516
23803.4062	15609.4023	-43.06372	1.70516
23959.4844	15609.0742	-33.70285	1.71616
24115.5547	15609.2422	105.35754	1.72616
24271.6406	15609.8711	-49.61833	1.73516
24427.7070	15609.5195	28.09495	1.74616
24583.8125	15609.1502	-369.83495	1.75516

24739.8242	15605.0469	-121.46631	1.76616
24895.9062	15603.5977	-167.41743	1.77616
25051.8789	15598.7930	-1100.27319	1.78615
25207.7773	15584.2187	-1595.40259	1.79615
25363.5508	15570.5625	-785.50464	1.80615
25519.2383	15568.7031	350.24658	1.81615
25674.9258	15571.4023	-367.30444	1.82615
25830.5391	15568.2109	223.09703	1.83615
25986.3437	15595.2070	7131.04297	1.84615
26064.9687	15567.2773	702.44629	1.85115
26220.6094	15565.0156	415.54175	1.86114
26376.2422	15571.0156	-22.71590	1.87114
26531.5703	15502.2969	-9184.44922	1.88114
25586.6836	15565.9414	13715.3203	1.89114
26842.5742	15594.3516	392.21875	1.90114
26998.4766	15593.7695	-197.45580	1.91113
27154.3594	15592.3828	-169.15584	1.92113
27310.2695	15594.4727	264.47192	1.93113
27466.1719	15573.3477	-9720.69453	1.94113
27621.8359	15589.3437	-347.67969	1.95113
27777.7344	15592.2734	168.09805	1.96112
27933.6953	15590.8750	259.85522	1.97112
28089.5977	15592.7227	158.51859	1.98112
28245.5000	15594.1836	-560.28296	1.99112
28401.3437	15571.8945	-4993.37500	2.00112
28556.8555	15543.0586	1258.08154	2.01112
28712.2891	15555.9805	1025.11523	2.02111
28867.8984	15565.4805	369.41064	2.03111
29023.5703	15573.2266	732.07202	2.04111
29179.2812	15580.5977	587.46362	2.05111
29335.1094	15585.8555	506.35595	2.06111
29490.9727	15590.9883	543.23120	2.07111
29646.9023	15595.4805	280.83521	2.08111
29802.8398	15597.6680	221.92377	2.09111
29958.8281	15599.6914	235.43085	2.10111
30114.8164	15601.8750	133.90338	2.11111
30270.8242	15602.7305	57.79265	2.12111
30426.8359	15603.3164	98.45195	2.13110
30582.8633	15604.3203	74.71588	2.14110
30738.8945	15604.9961	59.34158	2.15110
30894.9375	15605.4648	50.60559	2.16110
31050.9844	15605.4375	-91.63829	2.17110
31207.0117	15604.9062	66.03467	2.18110
31363.0703	15605.5937	40.40504	2.19110
31519.0977	15604.9219	-282.68823	2.20110
31675.1172	15601.7266	-207.56853	2.21110
31831.1250	15599.0391	-382.95165	2.22110
31987.0586	15594.1875	-545.72165	2.23110
32142.9687	15582.2656	-1943.44019	2.24110
32298.6211	15565.0039	-534.74316	2.25110
32454.4023	15562.7031	-473.08399	2.26110
32609.8281	15559.2109	-184.13727	2.27109
32765.5195	15557.4219	126.50465	2.28109
32921.0312	15583.0352	6487.88281	2.29109
33077.3828	15550.7148	449.62915	2.30109
33232.8906	15555.1406	230.79922	2.31109
33427.0625	15492.5742	-13135.1641	2.32359
33543.1328	15539.9219	11921.9844	2.33108
33699.7656	15600.8633	1181.39331	2.34108
33854.7461	15605.3906	-360.55479	2.35107

34010.7656	15604.5039	-408.37305	2.36107
34166.7500	15602.3398	-324.73486	2.37107
34361.7266	15584.3320	-7145.44141	2.38357
34478.4844	15612.7539	-64.34550	2.39106
34634.5547	15613.5547	-314.43774	2.40106
34790.6055	15610.9414	-251.71407	2.41106
34946.6836	15608.5312	-194.86621	2.42105
35102.7344	15606.9336	-169.19214	2.43105
35258.7812	15619.2656	3005.13330	2.44105
35415.2852	15634.5352	-1363.30005	2.45105
35571.8242	15600.2383	80.39467	2.46105
35727.7422	15603.5898	96.65274	2.47105
35883.7148	15605.0742	302.14258	2.48104
36039.7461	15605.0859	61.72400	2.49104
36195.7891	15606.6172	20.11620	2.50104

DRUM CAM (DC) PARAMETERS

	TIME (SEC)	MOTOR (DEG)	DC ROTATION (DEG)	DC VELOCITY (DEG/SEC)	DC ACCELERATION (DEG/SEC**2)	DC TORQUE (FT-LBS)
1	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	3078.77368	7.72490
3	0.01000	2.95483	0.15394	30.78770	3078.77368	7.72490
4	0.02000	11.81933	0.61575	61.57538	3078.77368	7.72490
5	0.03000	26.59344	1.38544	92.36296	3078.77124	7.72490
6	0.04000	47.27733	2.46301	123.09394	3059.71704	7.67709
7	0.05000	73.82103	3.84587	153.48358	3016.48682	7.56862
8	0.06000	106.15965	5.53062	183.38805	2965.37427	7.44038
9	0.07000	144.19530	7.51217	212.72563	2897.41211	7.26985
10	0.08000	187.77528	9.78261	241.30267	2817.19067	7.06857
11	0.09000	236.77397	12.33525	269.95386	2995.72363	7.51652
12	0.10000	291.69507	15.19641	299.53442	2681.03149	6.72693
13	0.11250	366.80444	19.10939	332.62793	2542.35840	6.37899
14	0.12500	450.43066	23.46608	363.79272	2438.69214	6.11889
15	0.13500	522.57176	27.22437	387.81104	2353.02173	5.90393
16	0.15000	639.27368	33.30421	422.14136	2210.84106	5.54719
17	0.16000	722.36401	37.63300	444.03223	2199.63770	5.51908
18	0.17000	809.78125	42.18712	465.86353	2116.47607	5.31042
19	0.18000	901.08569	46.94382	486.42236	2006.05029	5.03335
20	0.19000	996.39502	51.90919	505.88135	1870.86035	4.69414
21	0.20000	1095.25439	57.05942	524.30615	1810.95093	4.54383
22	0.21000	1197.64966	62.39394	542.05567	1732.47485	4.34693
23	0.22000	1303.29712	67.89781	558.90381	1651.91040	4.14478
24	0.23000	1412.16650	73.55960	575.08374	1566.16392	3.92954
25	0.24000	1524.01587	79.39665	590.33740	1488.54419	3.73488
26	0.25000	1638.74756	85.37378	604.92983	1429.67358	3.58717
27	0.26000	1756.16821	91.49109	618.58862	1337.69482	3.35639
28	0.27000	1876.16943	97.74274	631.60962	1271.83618	3.19114
29	0.28000	1998.59619	104.12085	643.90527	1190.89648	2.98806
30	0.29000	2123.30615	110.61780	655.45459	1184.43262	2.97184
31	0.30000	2250.17261	117.22722	666.30054	1055.57520	2.64853
32	0.31000	2379.06226	123.94194	676.51807	963.32861	2.42209
33	0.32000	2509.84375	130.75531	686.11157	927.46606	2.32709
34	0.33000	2642.41748	137.66193	695.09131	945.70654	2.37537
35	0.34000	2776.63525	144.65433	703.40918	808.86841	2.02952
36	0.35000	2912.42310	151.72842	711.23389	759.34570	1.90526
37	0.36000	3049.63867	158.87700	718.51392	704.63989	1.76800
38	0.37000	3188.22461	166.09688	725.30859	645.81689	1.62041
39	0.38000	3328.06543	173.38220	731.70459	614.31201	1.54136
40	0.39000	3469.08911	180.72908	737.63086	572.10791	1.43547
41	0.40000	3611.21289	188.13335	743.13770	528.66992	1.32648
42	0.41000	3754.34692	195.59015	748.21313	487.46094	1.22308
43	0.42000	3898.42261	203.09612	752.82397	429.87427	1.07859
44	0.43000	4043.30591	210.64405	756.90698	395.03223	0.99368
45	0.44000	4188.98437	218.23344	760.72217	428.49048	1.07512
46	0.45000	4335.30469	225.85635	764.12793	313.92017	0.78765
47	0.46000	4482.23828	233.51120	766.78467	135.36295	0.33964
48	0.47000	4629.63281	241.18994	769.10034	282.91626	0.70986
49	0.48000	4777.57812	248.89745	771.73926	261.06177	0.65503
50	0.49000	4925.84766	256.62159	773.98062	148.75398	0.37324
51	0.50000	5074.73437	264.37817	775.91064	340.63965	0.85469
52	0.50500	5149.21875	268.25854	776.39867	193.05457	0.48439
53	0.51250	5261.32031	274.09863	779.53467	801.19824	2.01027
54	0.52250	5411.26562	281.91040	782.39478	192.66731	0.48342
55	0.53500	5599.14453	291.69824	785.01147	259.09351	0.37518

56	0.54500	5749.87891	299.55103	784.76660	-366.34595	-0.91919
57	0.55500	5900.17578	307.38110	783.33667	1011.96631	2.53911
58	0.56250	6013.33203	313.27512	788.01099	-20.84888	-0.05231
59	0.57250	6164.80078	321.16724	790.42993	0.05234	0.00013
60	0.58250	6316.66016	329.07861	792.21094	33.52705	0.08412
61	0.59250	6468.83984	337.00584	793.62524	283.20605	0.71059
62	0.60500	6659.48437	346.93896	795.36572	-24.36969	-0.06115
63	0.61250	6773.96484	352.90283	794.62036	229.33350	0.57542
64	0.62250	6926.94922	0.87426	797.93164	150.73302	0.37820
65	0.63500	7118.53125	10.85514	799.62476	143.69965	0.36055
66	0.64500	7272.19531	18.86050	800.96167	135.04910	0.33885
67	0.65500	7426.05859	26.87527	802.34961	193.97777	0.48671
68	0.67000	7657.28906	38.92279	802.59277	-225.32460	-0.56536
69	0.67500	7734.26953	42.93317	801.79980	87.45496	0.21943
70	0.68500	7888.37500	50.96164	803.96851	126.32265	0.31695
71	0.69500	8042.83984	59.00876	805.15283	118.75436	0.29796
72	0.70500	8197.47266	67.06473	806.26880	47.26794	0.11860
73	0.72000	8429.94531	79.17584	807.72314	92.72174	0.23265
74	0.73000	8585.06541	87.25714	808.57178	75.90178	0.19044
75	0.74000	8740.32812	95.34575	809.16602	229.29829	0.57533
76	0.75000	8895.69922	103.44020	809.82959	63.79204	0.16006
77	0.76000	9051.19531	111.54105	810.45850	17.80133	0.04467
78	0.77000	9206.77734	119.64645	810.97266	44.40295	0.11141
79	0.78000	9362.47266	127.75768	811.36426	37.05809	0.09298
80	0.79000	9518.23047	135.87224	811.70508	-10.58400	-0.02656
81	0.80000	9674.05859	143.99037	812.01123	24.13759	0.06056
82	0.81000	9829.92578	152.11055	812.22632	22.44557	0.05532
83	0.82000	9985.84375	160.23347	812.39526	16.04330	0.04025
84	0.83000	10141.7812	168.35730	812.54321	12.92691	0.03243
85	0.84000	10297.7500	176.48283	812.62549	-98.10542	-0.24615
86	0.85000	10453.7578	184.61037	812.83887	6.52654	0.01638
87	0.86000	10609.7539	192.73727	812.88672	-37.13591	-0.09318
88	0.87000	10765.8125	200.86749	812.89185	-13.42530	-0.03369
89	0.88000	10921.7773	208.99275	812.78687	3.70088	0.00931
90	0.89000	11077.8203	217.12213	812.70972	-11.90322	-0.02987
91	0.90000	11233.7461	225.24539	812.47192	25.12585	0.06304
92	0.91000	11389.6211	233.36610	811.77197	-257.06934	-0.64501
93	0.92000	11545.3555	241.47931	810.99463	161.62437	0.40553
94	0.93000	11701.0273	249.58937	810.83130	-14.99984	-0.03764
95	0.94000	11856.6211	257.69507	810.78491	191.55727	0.48063
96	0.95000	12012.2187	265.80127	810.64185	72.37943	0.18161
97	0.96000	12167.8555	273.90942	812.18091	497.08472	1.24723
98	0.96500	12245.4687	278.00488	810.68921	163.82042	0.41104
99	0.97125	12343.7461	283.07275	810.57544	180.12204	0.45194
100	0.99250	12518.7617	292.19043	811.10156	-195.99930	-0.49178
101	0.99250	12674.2852	300.29297	808.81592	-1055.80566	-2.64911
102	1.00250	12829.2256	308.35499	809.22876	126.13232	0.31648
103	1.01250	12984.8857	316.47437	811.95532	379.45557	0.95209
104	1.02249	13140.6602	324.58936	812.26318	126.51671	0.31744
105	1.03499	13335.4492	334.73730	811.99414	101.01700	0.25346
106	1.04499	13491.3047	342.85718	812.00024	-169.68848	-0.42576
107	1.05499	13647.1562	350.97656	810.85693	-635.44653	-1.59590
108	1.06249	13763.8477	357.05591	812.14331	179.71335	0.45092
109	1.07248	13919.6641	5.17471	812.03467	14.56691	0.03555
110	1.08498	14114.4258	15.32119	812.28076	-51.08380	-0.12817
111	1.09498	14270.3242	23.44302	812.32886	8.49038	0.02130
112	1.10998	14504.1875	35.62662	812.12671	-96.70430	-0.24264
113	1.11998	14659.8672	43.73700	810.68823	-167.07649	-0.41921
114	1.12998	14815.3594	51.83774	809.68213	28.91371	0.07255
115	1.13998	14971.0536	59.94915	810.65796	49.92703	0.12527

116	1.14998	15126.3359	68.03860	810.87744	82.71118	0.20753
117	1.15998	15282.2930	76.16354	811.12207	41.73618	0.10472
118	1.16998	15437.8828	84.26930	811.64331	-4.65164	-0.01167
119	1.17997	15593.7461	92.38934	911.88013	27.21667	0.06829
120	1.18997	15749.5703	100.50728	812.14600	21.83340	0.05478
121	1.19997	15905.4648	108.62897	812.36279	-0.32829	-0.00082
122	1.20997	16061.3994	116.75261	812.52661	13.58605	0.03409
123	1.21997	16217.3555	124.87745	812.64722	17.83720	0.04476
124	1.22997	16373.3437	133.00404	812.73218	8.82977	0.02215
125	1.23997	16529.3398	141.13092	812.81128	6.97782	0.01751
126	1.24997	16685.3477	149.25841	812.76807	108.34869	0.27186
127	1.25997	16841.3516	157.38582	812.83765	6.30225	0.01581
128	1.26997	16997.3828	165.51455	812.91577	-36.59045	-0.09181
129	1.27997	17153.4023	173.64275	812.98950	3.00535	0.00754
130	1.28997	17309.4453	181.77208	813.00073	2.82310	0.00708
131	1.29997	17465.4922	189.90172	813.02905	-41.90919	-0.10515
132	1.30997	17621.5430	198.03143	813.08447	0.92572	0.00232
133	1.31996	17777.5977	206.16138	812.79395	21.41220	0.05373
134	1.32996	17933.5195	214.28450	812.45581	-9.71982	-0.02439
135	1.33996	18089.5859	222.41508	812.39038	-23.35052	-0.05859
136	1.34996	18245.3203	230.52837	811.92871	-71.23912	-0.17975
137	1.35996	18401.2070	238.64957	811.14524	-54.56219	-0.13590
138	1.36996	18556.7617	245.75354	810.59790	-11.32091	-0.02841
139	1.37996	18712.3516	254.85934	810.34009	-55.19897	-0.13850
140	1.38996	18867.8906	262.96191	810.75513	-62.31030	-0.15634
141	1.39996	19023.3857	271.06299	812.42651	379.22583	0.95151
142	1.40495	19101.3008	275.12207	815.24414	73.92143	0.18548
143	1.41246	19218.5000	281.22754	815.20728	-88.59637	-0.22230
144	1.42245	19374.8672	289.37402	814.20630	182.14789	0.45702
145	1.43495	19570.0273	299.54126	811.76465	-577.28296	-1.44845
146	1.44495	19725.3555	307.63330	808.94141	734.50562	1.84294
147	1.45245	19842.0781	313.71460	811.87842	-125.27484	-0.31433
148	1.46245	19997.9756	321.83643	812.47437	-163.08568	-0.40920
149	1.47244	20153.9023	329.95947	812.43213	-128.81532	-0.32321
150	1.48244	20309.8320	338.08275	812.76929	-146.81227	-0.36836
151	1.49244	20465.7461	346.20557	812.70508	-291.09790	-0.73039
152	1.50244	20621.5078	354.32031	812.29956	1288.45288	3.23284
153	1.51118	20757.9609	1.43043	814.02271	-31.41612	-0.07883
154	1.52118	20914.1133	9.56553	813.75928	-24.69197	-0.06195
155	1.53118	21070.2576	17.70012	813.56250	-13.27355	-0.03330
156	1.54118	21226.3828	25.83376	813.28613	-29.60391	-0.07177
157	1.55618	21460.5156	38.03142	812.99779	-7.17599	-0.01801
158	1.56617	21615.5654	46.15115	813.74194	207.33643	0.52022
159	1.57617	21773.1328	54.31775	815.02905	-104.29030	-0.26167
160	1.58117	21851.1875	58.38420	815.17944	-54.04947	-0.13561
161	1.59117	22007.5352	65.52951	814.66895	-53.36440	-0.13390
162	1.60117	22163.8857	74.57491	814.24219	-27.75253	-0.06963
163	1.61117	22320.1328	82.81487	813.91138	-38.22545	-0.09591
164	1.62617	22554.5156	95.02547	813.52026	23.09720	0.05795
165	1.63617	22710.6484	103.15955	813.51416	-9.08187	-0.02279
166	1.64616	22866.7959	111.29442	813.50098	-54.95911	-0.13790
167	1.65616	23022.9102	119.42749	813.41553	-6.86181	-0.01722
168	1.66616	23179.0117	127.55985	813.35620	-5.51758	-0.01384
169	1.67616	23335.1250	135.69295	813.30811	-29.74899	-0.07464
170	1.68616	23491.2227	143.82512	813.28784	-4.06880	-0.01021
171	1.69615	23647.3154	151.95711	813.24634	2.02344	0.00508
172	1.70616	23803.4062	160.08897	813.20874	-2.24352	-0.00563
173	1.71615	23959.4844	168.22014	813.19165	-1.75584	-0.00441
174	1.72616	24115.5547	176.35099	813.15454	-90.23256	-0.22640
175	1.73615	24271.6406	184.48254	813.23315	-2.58500	-0.00549

176	1.74616	24427.7070	192.61313	813.20996	-45.25417	-0.11355
177	1.75616	24583.8125	200.74579	813.14404	-19.26743	-0.04834
178	1.76616	24739.8242	208.87350	812.98462	0.15035	0.00038
179	1.77616	24895.9062	217.00491	812.88916	-9.54167	-0.02394
180	1.78615	25051.878	225.13062	812.66406	8.42046	0.02113
181	1.79615	25207.7773	233.25255	811.96338	-209.85748	-0.52555
182	1.80615	25363.5506	241.36780	811.10474	86.18420	0.21524
183	1.81615	25519.2383	249.47867	811.08691	-41.02216	-0.10293
184	1.82615	25674.9258	257.58911	811.19238	178.76428	0.44853
185	1.83615	25830.5391	265.69629	811.01001	57.46159	0.14418
186	1.84615	25986.3437	273.81323	812.47705	443.40137	1.11253
187	1.85115	26064.9687	277.90918	810.97998	133.77261	0.33565
188	1.86114	26220.6094	286.01758	810.95337	80.13707	0.20107
189	1.87114	26375.2422	294.12549	811.21973	22.91066	0.05748
190	1.88114	26531.5703	302.21753	807.64014	-492.19043	-1.23495
191	1.89114	26686.6836	310.29858	811.00464	660.44800	1.65712
192	1.90114	26842.5742	318.42017	812.26807	339.53979	0.85193
193	1.91113	26998.4766	326.54199	812.34521	94.62389	0.23742
194	1.92113	27154.3594	334.66309	812.28906	97.78813	0.24536
195	1.93113	27310.2695	342.78589	812.27539	-203.92345	-0.51166
196	1.94113	27466.1719	350.90796	811.14600	-665.58984	-1.67002
197	1.95113	27621.8359	359.01758	811.96191	-107.53307	-0.26981
198	1.96112	27777.7344	7.14057	812.31323	8.75743	0.02197
199	1.97112	27933.6953	15.26562	812.30273	-28.62325	-0.07182
200	1.98112	28089.5977	23.38768	812.33984	8.25840	0.02072
201	1.99112	28245.5000	31.50969	812.41602	-29.18927	-0.07324
202	2.00112	28401.3437	39.62874	811.25513	-282.15916	-0.70796
203	2.01112	28556.8555	47.73038	809.75244	65.54272	0.16445
204	2.02111	28712.2891	55.82797	810.42798	60.15889	0.15094
205	2.03111	28867.8984	63.93481	810.92065	45.29399	0.11365
206	2.04111	29023.5703	72.04480	811.32422	38.13901	0.09569
207	2.05111	29179.2812	80.15594	811.70825	30.60529	0.07679
208	2.06111	29335.1094	88.27509	811.98218	26.37979	0.06619
209	2.07111	29490.9727	95.39508	812.30151	-84.40973	-0.21179
210	2.08111	29646.9023	104.51857	812.48340	14.63077	0.03671
211	2.09111	29802.8398	112.64241	812.59839	18.59821	0.04666
212	2.10111	29958.8281	120.76900	812.69849	-21.70271	-0.05445
213	2.11111	30114.8164	128.89546	812.81665	6.97600	0.01750
214	2.12111	30270.8242	137.02296	812.75073	40.18839	0.10084
215	2.13110	30426.8359	145.15076	812.89185	5.12908	0.01287
216	2.14110	30582.8633	153.27928	812.94409	3.89248	0.00977
217	2.15110	30738.8945	161.40810	812.97925	3.09152	0.00776
218	2.16110	30894.9375	169.53743	813.00366	2.63641	0.00661
219	2.17110	31050.9844	177.66695	813.02441	93.13914	0.23369
220	2.18110	31207.0117	185.79558	812.97461	3.44022	0.00863
221	2.19110	31363.0703	193.92572	813.01392	9.08891	0.02280
222	2.20110	31519.0977	202.05434	812.97534	-14.72733	-0.03695
223	2.21110	31675.1172	210.18245	812.80884	-10.81900	-0.02715
224	2.22110	31831.1250	218.30997	812.69385	69.73746	0.17498
225	2.23110	31987.0586	226.43375	812.44165	-221.78366	-0.55647
226	2.24110	32142.9687	234.55614	811.78052	-57.94676	-0.14539
227	2.25110	32298.6211	242.66515	810.92896	-42.76021	-0.10729
228	2.26110	32454.4023	250.78085	810.75952	-24.64590	-0.06184
229	2.27109	32609.8281	258.87793	810.77417	-50.15254	-0.12584
230	2.28109	32765.5195	266.98877	810.70093	-60.74057	-0.15240
231	2.29109	32921.0312	275.09058	812.02222	189.57945	0.47567
232	2.30109	33077.3828	283.23584	810.19067	237.36172	0.59582
233	2.31109	33232.8906	291.33740	810.49194	278.32983	0.69835
234	2.32358	33427.0625	301.45337	806.97900	-327.55332	-0.82211
235	2.33108	33543.1328	307.50000	809.75000	783.12891	1.96494

236	2.34108	33698.7656	315.60815	812.70728	93.39699	0.23434
237	2.35107	33854.7461	323.73437	813.00195	37.28123	0.09354
238	2.36107	34010.7656	331.86255	812.91504	66.12245	0.16591
239	2.37107	34166.7500	339.98853	812.84448	46.79245	0.11741
240	2.38357	34361.7266	350.14648	812.07446	-582.80176	-1.46230
241	2.39106	34478.4844	356.22900	813.39429	-26.07263	-0.06542
242	2.40106	34634.5547	4.36131	813.42187	-16.38130	-0.04110
243	2.41106	34790.6055	12.49111	813.28564	-13.11359	-0.03290
244	2.42105	34946.6836	20.62225	813.16333	-10.15202	-0.02547
245	2.43105	35102.7344	28.75198	813.08203	-1.72947	-0.00434
246	2.44105	35258.7812	36.88164	813.72266	156.61163	0.39295
247	2.45105	35415.2852	45.03496	814.51807	-69.98241	-0.17559
248	2.46105	35571.8242	53.19012	812.54844	89.20624	0.22383
249	2.47105	35727.7422	61.31305	812.90601	5.03535	0.01263
250	2.48104	35883.7148	69.43875	813.02881	-122.58806	-0.30758
251	2.49104	36039.7461	77.56754	813.03613	2.17371	0.00545
252	2.50104	36195.7891	85.69587	813.06494	8.17526	0.02051
253	2.50104	0.0	0.0	0.0	0.0	0.0

FEED MECHANISM (FM) PARAMETERS

	TIME (SEC)	MOTOR (DEG)	FM ROTATION (DEG)	FM VELOCITY (DEG/SEC)	FM ACCELERATION (DEG/SEC**2)	SHAFT TORQUE (FT-LBS)	CAM FORCE (LBS)	FORCE ON AMMO (LBS)
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	139.53050	0.0	-0.00000	-0.00000	0.00639	0.03831
3	0.01000	2.95483	139.53050	-0.00000	-0.00000	-0.00000	0.00239	0.01436
4	0.02000	11.81933	139.53050	0.00000	0.00000	0.0	-0.01415	-0.08490
5	0.03000	26.59344	139.53050	0.00000	0.00000	0.0	-0.02311	-0.13867
6	0.04000	47.27733	139.53050	0.00000	0.00000	0.0	-0.45157	-2.70943
7	0.05000	73.82103	139.53050	0.00000	0.00000	0.0	-4.83046	-28.98270
8	0.06000	106.15965	139.53049	0.00000	0.00000	0.0	-5.62919	-33.77507
9	0.07000	144.19530	139.53049	0.00000	0.00000	0.0	-7.33554	-44.01917
10	0.08000	187.77528	139.53049	0.00000	0.00000	0.0	-11.27948	-67.67676
11	0.09000	236.77397	139.53049	0.00000	0.00000	0.0	-19.36844	-116.21040
12	0.10000	291.69507	139.53049	-0.00000	-0.00000	-0.00000	54.59380	327.56201
13	0.11250	366.80444	139.53049	-0.00000	-0.00000	-0.00000	53.03960	318.23682
14	0.12500	450.43066	139.53049	-0.00000	-0.00000	-0.00000	83.15187	498.91016
15	0.13500	522.57178	139.53049	-0.00000	-0.00000	-0.00000	78.39244	470.35352
16	0.15000	639.27368	139.53049	-0.00000	-0.00000	-0.00000	98.48895	590.93237
17	0.16000	722.36401	139.53049	-0.00000	-0.00000	-0.00000	62.29385	373.76221
18	0.17000	809.78125	139.53049	-0.00000	-0.00000	-0.00000	109.20630	655.23633
19	0.18000	901.08569	139.53049	-0.00000	-0.00000	-0.00000	97.64070	585.84302
20	0.19000	996.39502	139.53049	-0.00000	-0.00000	-0.00000	102.47989	614.87793
21	0.20000	1095.25439	139.53049	-0.00000	-0.00000	-0.00000	124.53221	747.19165
22	0.21000	1197.64966	139.53049	-0.00000	-0.00001	-0.00000	172.56261	1035.97363
23	0.22000	1303.29712	139.53049	-0.00000	-0.00000	-0.00000	123.69138	742.14673
24	0.23000	1412.16650	139.53049	-0.00000	-0.00000	-0.00000	98.08357	588.50024
25	0.24000	1524.01587	139.53049	-0.00000	-0.00001	-0.00000	204.86046	1229.16040
26	0.25000	1638.74756	139.53049	-0.00000	-0.00001	-0.00000	211.31259	1267.87305
27	0.26000	1756.16821	139.53049	-0.00000	-0.00001	-0.00000	241.05710	1446.33960
28	0.27000	1876.16943	139.53049	-0.00000	-0.00000	-0.00000	137.96355	827.77954
29	0.28000	1998.59619	139.53049	-0.00000	-0.00001	-0.00000	196.65067	1179.90161
30	0.29000	2123.30615	139.53049	-0.00000	-0.00000	-0.00000	132.88852	797.32959
31	0.30000	2250.17261	139.53049	-0.00000	-0.00001	-0.00000	228.89168	1373.34741
32	0.31000	2379.06226	139.53049	-0.00000	-0.00001	-0.00000	187.50111	1125.00439
33	0.32000	2509.84375	139.53049	-0.00000	-0.00001	-0.00000	164.43955	985.62939
34	0.33000	2642.41748	139.53049	-0.00000	-0.00001	-0.00000	155.39529	932.35987
35	0.34000	2776.63525	139.53049	-0.00000	-0.00001	-0.00000	158.04533	948.27002
36	0.35000	2912.42310	139.53049	-0.00000	-0.00001	-0.00000	172.54234	1035.25195
37	0.36000	3049.63867	139.53049	-0.00000	-0.00001	-0.00000	200.14058	1200.84106
38	0.37000	3188.22461	139.53049	-0.00000	-0.00001	-0.00000	244.48074	1466.88159
39	0.38000	3328.06543	139.53049	-0.00000	-0.00001	-0.00000	310.45215	1862.70923
40	0.39000	3469.08911	139.53049	-0.00000	-0.00001	-0.00000	185.64731	1113.88159
41	0.40000	3611.21289	139.53049	-0.00000	-0.00001	-0.00000	264.16333	1594.97681
42	0.41000	3754.34692	139.53049	-0.00000	-0.00001	-0.00000	168.70036	1012.20020
43	0.42000	3898.42261	139.53049	-0.00000	-0.00001	-0.00000	263.11548	1578.68970
44	0.43000	4043.30591	139.50035	-16.65686	-9520.89062	-2.12117	****************	****************
45	0.44000	4188.98437	138.75076	-168.65324	-15543.9062	-3.46304	****************	****************
46	0.45000	4335.30469	135.97501	-424.00806	-32501.3164	-7.26329	****************	****************
47	0.46000	4482.23928	130.01825	-775.50293	-38531.5195	-8.58449	****************	****************
48	0.47000	4629.63281	120.65725	-1079.41260	-26144.4453	-5.82475	****************	****************
49	0.48000	4777.57812	108.84427	-1268.19092	-15378.3711	-3.42516	****************	****************
50	0.49000	4925.94766	95.66457	-1365.04658	-8354.27734	-1.36126	****************	****************
51	0.50000	5074.73437	81.61292	-1443.59385	-19193.7656	-4.27620	****************	****************
52	0.50500	5149.21375	74.28598	-1480.86621	-33319.0117	-7.42318	****************	****************
53	0.51250	5261.32031	65.47772	-538.89819	205924.375	45.37814	****************	****************
54	0.52250	5411.26562	64.63979	0.00000	0.00000	0.0	-57.17070	-343.02344
55	0.53500	5599.14453	54.61503	-40.67044	-25195.9414	-5.93545	****************	****************

56	0.54500	5749.87891	74.12500	2863.67822	493320.187	0.0	********************************	
57	0.55500	5900.17578	111.91002	2938.29469	-283597.937	0.0	********************************	
58	0.56250	6013.33203	128.19601	1604.75732	-127738.250	0.0	********************************	
59	0.57250	6164.80078	138.72484	486.81982	-117630.375	0.0	********************************	
60	0.58250	6316.66016	139.53061	1.05517	870.71484	0.0	********************************	
61	0.59250	6468.83984	139.53049	-0.00000	-0.00000	-0.00000	34.79472	208.76793
62	0.60500	6659.48437	139.53049	-0.00000	-0.00000	-0.00000	31.29500	187.76962
63	0.61250	6773.96484	139.53049	-0.00000	-0.00000	-0.00000	29.39609	176.37617
64	0.62250	6926.94922	139.53050	0.00000	0.00000	0.0	-1.80970	-10.85816
65	0.63500	7118.53125	139.53049	0.00000	0.00000	0.0	-124.25261	-745.51416
66	0.64500	7272.19531	139.53049	-0.00000	-0.00001	-0.00000	244.76599	1468.59302
67	0.65500	7426.05859	139.53049	-0.00000	-0.00001	-0.00000	248.91231	1493.47095
68	0.67000	7657.28906	139.53049	-0.00000	-0.00001	-0.00000	258.89722	1553.38013
69	0.67500	7734.26953	139.53049	-0.00000	-0.00001	-0.00000	260.81372	1564.87915
70	0.68500	7888.37500	139.53049	-0.00000	-0.00001	-0.00000	268.50488	1511.02612
71	0.69500	8042.83984	139.53049	-0.00000	-0.00001	-0.00000	279.98340	1579.89697
72	0.70500	8197.47266	139.53049	-0.00000	-0.00001	-0.00000	293.86523	1763.18799
73	0.72000	8429.94531	139.53049	-0.00000	-0.00001	-0.00000	322.27710	1933.65869
74	0.73000	8585.06641	139.53049	-0.00000	-0.00001	-0.00000	344.12207	2064.72827
75	0.74000	8740.32812	139.53049	-0.00000	-0.00001	-0.00000	368.88354	2213.29687
76	0.75000	8895.69922	139.53049	-0.00000	-0.00001	-0.00000	396.49780	2378.98218
77	0.76000	9051.19531	139.53049	-0.00000	-0.00001	-0.00000	188.99678	1133.97852
78	0.77000	9206.77734	139.53049	-0.00000	-0.00001	-0.00000	208.25117	1249.50464
79	0.78000	9362.47266	139.53049	-0.00000	-0.00001	-0.00000	230.00760	1380.04272
80	0.79000	9518.23047	139.53049	-0.00000	-0.00001	-0.00000	254.03076	1524.18164
81	0.80000	9674.05859	139.53049	-0.00000	-0.00001	-0.00000	280.60229	1683.61035
82	0.81000	9829.92578	139.53049	-0.00000	-0.00001	-0.00000	309.53027	1857.17798
83	0.82000	9985.84375	139.53049	-0.00000	-0.00001	-0.00000	341.16895	2047.00977
84	0.83000	10141.7812	139.53049	-0.00000	-0.00001	-0.00000	375.27393	2251.63916
85	0.84000	10297.7500	139.53049	-0.00000	-0.00001	-0.00000	412.18433	2473.10107
86	0.85000	10453.7578	139.53049	-0.00000	-0.00001	-0.00000	202.60162	1215.60742
87	0.86000	10609.7539	139.53049	-0.00000	-0.00001	-0.00000	226.38455	1361.30469
88	0.87000	10765.8125	139.53049	-0.00000	-0.00001	-0.00000	253.83435	1523.00317
89	0.88000	10921.7773	139.52342	-33.50121	-26605.1211	-5.92761	********************************	
90	0.89000	11077.8203	138.96571	-92.33604	-12983.0820	-2.89252	********************************	
91	0.90000	11233.7461	136.30382	-421.47583	-37890.7812	-8.44174	********************************	
92	0.91000	11389.6211	130.15461	-812.72656	-42935.8242	-9.56573	********************************	
93	0.92000	11545.3555	120.25931	-1148.55640	-29665.1445	-6.60913	********************************	
94	0.93000	11701.0273	107.70465	-1339.43042	-9219.73047	-2.05407	********************************	
95	0.94000	11856.6211	93.76494	-1438.24683	-6483.97266	-1.44457	********************************	
96	0.95000	12012.2187	78.94930	-1517.10083	-7023.71094	-1.56482	********************************	
97	0.96000	12167.8555	65.63882	-716.61353	212833.625	47.41747	********************************	
98	0.96500	12246.4687	64.63905	-31.40706	-29738.9102	-6.62557	********************************	
99	0.97125	12343.7451	64.63979	0.00000	0.00000	0.0	-58.72499	-352.34912
100	0.98250	12518.7617	64.58673	-50.48087	5339.99609	1.18970	********************************	
101	0.99250	12674.2852	77.05104	3423.99292	494215.812	0.0	********************************	
102	1.00250	12829.2256	115.39061	2596.64600	-253791.687	0.0	********************************	
103	1.01250	12984.8857	133.77278	1190.49683	-91341.0000	0.0	********************************	
104	1.02249	13140.5602	139.58731	-19.49080	-48240.9258	-10.74766	********************************	
105	1.03499	13335.4492	139.53049	-0.00000	-0.00000	-0.00000	22.36960	134.21733
106	1.04499	13491.3047	139.53049	-0.00000	-0.00000	-0.00000	28.19609	169.17521
107	1.05499	13647.1552	139.53049	-0.00000	-0.00000	-0.00000	34.58167	208.08961
108	1.06249	13763.8477	139.53049	-0.00000	-0.00000	-0.00000	39.55793	237.94711
109	1.07248	13919.6641	139.53049	0.00000	0.00001	0.0	-154.01465	-924.08594
110	1.08498	14114.4258	139.53049	-0.00000	-0.00001	-0.00000	416.84009	2501.03564
111	1.09498	14270.3242	139.53049	-0.00000	-0.00001	-0.00000	399.57422	2397.44067
112	1.10998	14504.1875	139.53049	-0.00000	-0.00001	-0.00000	204.99913	1229.99243
113	1.11998	14659.8672	139.53049	-0.00000	-0.00001	-0.00000	225.32425	1351.94287
114	1.12998	14815.3594	139.53049	-0.00000	-0.00001	-0.00000	245.33459	1472.00464
115	1.13998	14971.0586	139.53049	-0.00000	-0.00001	-0.00000	270.16504	1520.98706

116	1.14998	15126.3359	139.53049	-0.00000	-0.00001	-0.00000	291.05322	1746.31592
117	1.15998	15282.2930	139.53049	-0.00000	-0.00001	-0.00000	321.83398	1931.00000
118	1.16998	15437.8828	139.53049	-0.00000	-0.00001	-0.00000	343.93335	2099.59595
119	1.17997	15593.7461	139.53049	-0.00000	-0.00001	-0.00000	383.03721	2301.81885
120	1.18997	15749.5703	139.53049	-0.00000	-0.00001	-0.00000	183.92659	1103.55737
121	1.19997	15905.4648	139.53049	-0.00000	-0.00001	-0.00000	205.68384	1234.10059
122	1.20997	16061.3984	139.53049	-0.00000	-0.00001	-0.00000	229.62247	1377.73193
123	1.21997	16217.3555	139.53049	-0.00000	-0.00001	-0.00000	255.73103	1534.38306
124	1.22997	16373.3437	139.53049	-0.00000	-0.00001	-0.00000	284.24463	1705.46436
125	1.23997	16529.3398	139.53049	-0.00000	-0.00001	-0.00000	315.02981	1890.16919
126	1.24997	16685.3477	139.53049	-0.00000	-0.00001	-0.00000	348.20166	2089.20581
127	1.25997	16841.3516	139.53049	-0.00000	-0.00001	-0.00000	383.35796	2301.94336
128	1.26997	16997.3828	139.53049	-0.00000	-0.00001	-0.00000	185.60231	1113.61157
129	1.27997	17153.4023	139.53049	-0.00000	-0.00001	-0.00000	208.65594	1251.93311
130	1.28997	17309.4453	139.53049	-0.00000	-0.00001	-0.00000	233.91925	1403.51270
131	1.29997	17465.4922	139.53049	-0.00000	-0.00001	-0.00000	261.35864	1568.14868
132	1.30997	17621.5430	139.53049	-0.00000	-0.00001	-0.00000	291.05542	1746.32910
133	1.31996	17777.5977	139.52924	-8.11554	-20965.1914	-4.67086	***************	***************
134	1.32996	17933.5195	139.32019	-51.28806	17134.0039	3.81730	***************	***************
135	1.33996	18089.5859	137.56488	-302.80566	-33616.5820	-7.48948	***************	***************
136	1.34996	18245.3203	132.75748	-669.76099	-34674.4922	-7.72517	***************	***************
137	1.35996	18401.2070	124.09119	-1045.73584	-32764.1055	-7.29955	***************	***************
138	1.36996	18556.7617	112.30412	-1289.43140	-18510.8750	-4.12406	***************	***************
139	1.37996	18712.3516	98.75179	-1407.44800	-5847.99219	-1.30288	***************	***************
140	1.38996	18867.8906	86.23149	-1496.15503	-6022.76562	-1.34182	***************	***************
141	1.39996	19023.3867	69.21889	-1267.54492	129948.875	28.95146	***************	***************
142	1.40496	19101.3008	64.83427	-345.05543	287805.187	64.12045	***************	***************
143	1.41246	19218.5000	64.63979	0.00000	0.00000	0.0	-56.14722	-336.88257
144	1.42245	19374.8672	64.63979	0.00000	0.00000	0.0	-52.24561	-313.47900
145	1.43495	19570.0273	74.08939	2955.82471	527291.187	0.0	***************	***************
146	1.44495	19725.3555	112.84401	2942.34082	-291468.562	0.0	***************	***************
147	1.45245	19842.0781	129.06648	1580.00562	-127070.812	0.0	***************	***************
148	1.46245	19997.9766	139.09825	395.62793	-149057.187	0.0	***************	***************
149	1.47244	20153.9023	139.53047	0.11794	172.61935	0.0	***************	***************
150	1.48244	20309.8320	139.53049	-0.00000	-0.00000	-0.00000	41.19847	247.19035
151	1.49244	20465.7461	139.53049	-0.00000	-0.00000	-0.00000	49.63661	297.81909
152	1.50244	20621.5078	139.53049	-0.00000	-0.00000	-0.00000	58.48513	350.91602
153	1.51118	20757.9609	139.53050	0.00000	0.00000	0.0	-1.40954	-8.45724
154	1.52118	20914.1133	139.53049	0.00000	0.00000	0.0	-106.22581	-637.35352
155	1.53118	21070.2578	139.53050	0.00000	-0.00000	0.0	60.20271	361.21558
156	1.54118	21226.3828	139.53049	-0.00000	-0.00001	-0.00000	246.67877	1480.06958
157	1.55618	21460.5156	139.53049	-0.00000	-0.00001	-0.00000	290.69312	1744.15527
158	1.56617	21616.5664	139.53049	-0.00000	-0.00001	-0.00000	323.33518	1940.01318
159	1.57617	21773.1328	139.53049	-0.00000	-0.00001	-0.00000	366.11499	2196.68555
160	1.58117	21851.1875	139.53049	-0.00000	-0.00001	-0.00000	385.16455	2310.98267
161	1.59117	22007.5352	139.53049	-0.00000	-0.00001	-0.00000	188.84993	1133.09741
162	1.60117	22163.8967	139.53049	-0.00000	-0.00001	-0.00000	215.25922	1291.55273
163	1.61117	22320.1328	139.53049	-0.00000	-0.00001	-0.00000	243.16514	1459.00586
164	1.62617	22554.5156	139.53049	-0.00000	-0.00001	-0.00000	289.85352	1739.11768
165	1.63617	22710.6484	139.53049	-0.00000	-0.00001	-0.00000	322.74097	1936.44189
166	1.64616	22866.7969	139.53049	-0.00000	-0.00001	-0.00000	358.41870	2150.50806
167	1.65616	23022.9102	139.53049	-0.00000	-0.00001	-0.00000	396.24341	2377.45581
168	1.66616	23179.0117	139.53049	-0.00000	-0.00001	-0.00000	193.67072	1152.02197
169	1.67616	23335.1250	139.53049	-0.00000	-0.00001	-0.00000	218.32104	1309.92358
170	1.68616	23491.2227	139.53049	-0.00000	-0.00001	-0.00000	244.98607	1469.91357
171	1.69615	23647.3154	139.53049	-0.00000	-0.00001	-0.00000	273.82959	1642.97437
172	1.70616	23803.4052	139.53049	-0.00000	-0.00001	-0.00000	304.95093	1829.70190
173	1.71616	23959.4844	139.53049	-0.00000	-0.00001	-0.00000	338.32788	2029.96338
174	1.72616	24115.5547	139.53049	-0.00000	-0.00001	-0.00000	374.08911	2244.53027
175	1.73616	24271.6406	139.53049	-0.00000	-0.00001	-0.00000	412.56040	2475.35752

176	1.74616	24427.7070	139.53049	-0.00000	-0.00001	-0.00000	203.31081	1219.86230
177	1.75616	24583.8125	139.53049	-0.00000	-0.00001	-0.00000	228.71701	1372.29932
178	1.76616	24739.8242	139.52802	-28.81087	-38186.3242	-8.50758	******************************	
179	1.77616	24895.9062	133.97897	-92.24298	10386.1914	2.31395	******************************	
180	1.78615	25051.8789	135.36299	-416.29712	-36878.5625	-8.21622	******************************	
181	1.79615	25207.7773	130.27789	-806.95630	-41865.7656	-9.32733	******************************	
182	1.80615	25363.5508	120.41696	-1144.69019	-29202.8984	-6.50515	******************************	
183	1.81615	25519.2393	107.88585	-1340.44556	-2558.42310	-0.56999	******************************	
184	1.82615	25674.9258	93.95261	-1438.20728	-5890.53516	-1.31236	******************************	
185	1.83615	25830.5391	79.14575	-1517.08154	-4272.96484	-0.95198	******************************	
186	1.84615	25986.3437	65.72523	-741.84546	207762.000	46.28755	******************************	
187	1.85115	26064.9687	64.64253	-27.15811	-41925.8711	-9.34072	******************************	
188	1.86114	26220.6094	64.63979	0.00000	0.00000	0.0	-59.72975	-358.37769
189	1.87114	26376.2422	64.68686	212.88474	206290.937	0.0	******************************	
190	1.88114	26531.5703	86.42979	4343.56250	220000.750	0.0	******************************	
191	1.89114	26686.6836	121.19069	2194.48364	-177232.125	0.0	******************************	
192	1.90114	26842.5742	136.30907	917.99268	-111335.687	0.0	******************************	
193	1.91113	26998.4766	139.52631	0.94677	30453.3867	0.0	******************************	
194	1.92113	27154.3594	139.53049	-0.00000	-0.00000	-0.00000	19.19582	115.17467
195	1.93113	27310.2695	139.53049	-0.00000	-0.00000	-0.00000	24.68562	148.11345
196	1.94113	27466.1719	139.53049	-0.00000	-0.00000	-0.00000	30.85582	185.13455
197	1.95113	27621.8359	139.53049	-0.00000	-0.00000	-0.00000	37.31073	223.86395
198	1.96112	27777.7344	139.53049	0.00000	0.00000	0.0	-151.29404	-907.76245
199	1.97112	27933.6953	139.53049	-0.00000	-0.00001	-0.00000	403.10034	2418.59741
200	1.98112	28089.5977	139.53049	-0.00000	-0.00001	-0.00000	383.43799	2300.62354
201	1.99112	28245.5000	139.53049	-0.00000	-0.00001	-0.00000	184.36183	1106.16870
202	2.00112	28401.3437	139.53049	-0.00000	-0.00001	-0.00000	204.90512	1229.42822
203	2.01112	28556.8555	139.53049	-0.00000	-0.00001	-0.00000	223.58362	1341.49902
204	2.02111	28712.2891	139.53049	-0.00000	-0.00001	-0.00000	243.75282	1462.51392
205	2.03111	28867.8984	139.53049	-0.00000	-0.00001	-0.00000	267.12573	1602.75122
206	2.04111	29023.5703	139.53049	-0.00000	-0.00001	-0.00000	292.76831	1756.60645
207	2.05111	29179.2812	139.53049	-0.00000	-0.00001	-0.00000	320.61279	1923.67285
208	2.06111	29335.1094	139.53049	-0.00000	-0.00001	-0.00000	351.83545	2111.00854
209	2.07111	29490.9727	139.53049	-0.00000	-0.00001	-0.00000	385.55994	2313.94897
210	2.08111	29646.9023	139.53049	-0.00000	-0.00001	-0.00000	186.01540	1116.09009
211	2.09111	29802.8398	139.53049	-0.00000	-0.00001	-0.00000	208.31949	1249.91431
212	2.10111	?9953.8281	139.53049	-0.00000	-0.00001	-0.00000	233.02374	1398.13965
213	2.11111	30114.8154	139.53049	-0.00000	-0.00001	-0.00000	259.75952	1558.55396
214	2.12111	30270.8242	139.53049	-0.00000	-0.00001	-0.00000	288.79587	1732.77783
215	2.13110	30426.8359	139.53049	-0.00000	-0.00001	-0.00000	320.09473	1920.56445
216	2.14110	30582.8633	139.53049	-0.00000	-0.00001	-0.00000	353.93579	2123.61060
217	2.15110	30738.8945	139.53049	-0.00000	-0.00001	-0.00000	390.24146	2341.44409
218	2.16110	30894.9375	139.53049	-0.00000	-0.00001	-0.00000	189.61044	1137.66040
219	2.17110	31050.9844	139.53049	-0.00000	-0.00001	-0.00000	213.24883	1279.49048
220	2.18110	31207.0117	139.53049	-0.00000	-0.00001	-0.00000	238.71503	1432.28735
221	2.19110	31363.0703	139.53049	-0.00000	-0.00001	-0.00000	266.57847	1600.06763
222	2.20110	31519.0977	139.53049	-0.00000	-0.00001	-0.00000	295.43528	1778.61426
223	2.21110	31675.1172	139.51105	-2.50873	-4783.89062	-1.06581	******************************	
224	2.22110	31831.1250	138.73373	-181.40750	-8319.69922	-1.85356	******************************	
225	2.23110	31987.0586	135.64572	-475.25415	-31116.4453	-6.93247	******************************	
226	2.24110	32142.9687	128.92238	-881.81616	-41281.0352	-9.19705	******************************	
227	2.25110	32298.6211	118.55179	-1184.55984	-22294.5820	-4.96704	******************************	
228	2.26110	32454.4023	105.72318	-1358.10449	-13341.6133	-2.97239	******************************	
229	2.27109	32609.8281	91.65392	-1455.81396	-10415.8437	-2.32056	******************************	
230	2.28109	32765.5195	75.69702	-1552.14917	-7113.23828	-1.58477	******************************	
231	2.29109	32921.0312	64.84784	-354.75854	283507.562	63.16299	******************************	
232	2.30109	33077.3828	64.63979	0.00000	0.00000	0.0	-55.28560	-331.71289
233	2.31109	33232.8906	64.63477	-20.13963	-42930.7187	-9.56459	******************************	
234	2.32358	33427.0625	82.44212	4054.13184	369678.937	0.0	******************************	
235	2.33109	33543.1328	112.35504	2994.50806	-300361.312	0.0	******************************	

236	2.34108	33698.7656	132.44289	1309.32300	-121341.250	0.0	********************************	
237	2.35107	33854.7461	139.57043	60.19624	-85668.3750	0.0	********************************	
238	2.36107	34010.7656	139.53049	-0.00000	-0.00000	-0.00000	28.41626	170.49722
239	2.37107	34166.7500	139.53049	-0.00000	-0.00000	-0.00000	35.45369	212.72173
240	2.38357	34361.7256	139.53049	-0.00000	-0.00000	-0.00000	45.45367	272.72144
241	2.39106	34478.4844	139.53049	-0.00000	-0.00000	-0.00000	51.59172	309.54956
242	2.40106	34634.5547	139.53050	0.00000	0.00001	0.0	-201.77512	-1210.64819
243	2.41106	34790.6055	139.53049	0.00000	0.00001	0.0	-181.26375	-1087.58032
244	2.42105	34946.6836	139.53049	-0.00000	-0.00001	-0.00000	204.72493	1228.34717
245	2.43105	35102.7344	139.53049	-0.00000	-0.00001	-0.00000	229.58280	1378.09399
246	2.44105	35259.7812	139.53049	-0.00000	-0.00001	-0.00000	257.21777	1543.30347
247	2.45105	35415.2852	139.53049	-0.00000	-0.00001	-0.00000	292.62695	1755.75830
248	2.46105	35571.8242	139.53049	-0.00000	-0.00001	-0.00000	329.90210	1979.40869
249	2.47105	35727.7422	139.53049	-0.00000	-0.00001	-0.00000	362.97998	2177.87549
250	2.48104	35883.7148	139.53049	-0.00000	-0.00001	-0.00000	399.06323	2394.37476
251	2.49104	36039.7461	139.53049	-0.00000	-0.00001	-0.00000	194.81343	1168.87817
252	2.50104	36195.7891	139.53049	-0.00000	-0.00001	-0.00000	218.88916	1313.33228
253	2.50104	0.0	0.0	0.0	0.0	0.0	0.0	0.0

EJECT MECHANISM (EM) PARAMETERS

	TIME (SEC)	MOTOR (DEG)	EM ROTATION (DEG)	EM VELOCITY (DEG/SEC)	EM ACCELERATION (DEG/SEC**2)	SHAFT TORQUE (FT-LBS)	CAM FORCE (LBS)
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	234.52350	0.0	-0.00000	-0.00100	-0.00600
3	0.01000	2.95483	234.52350	-0.00000	-0.00000	-0.00039	-0.00231
4	0.02000	11.81933	234.52350	-0.00000	-0.00000	-0.03555	-0.21330
5	0.03000	26.59344	234.52350	-0.00000	-0.00000	-0.28042	-1.68250
6	0.04000	47.27733	234.52350	-0.00000	-0.00000	-0.63946	-3.83674
7	0.05000	73.82103	234.52350	-0.00000	0.00000	1.87823	11.26934
8	0.06000	106.15965	234.52348	-0.00000	0.000[illegible]	1.69019	10.14112
9	0.07000	144.19530	234.52348	-0.00000	0.00000	2.29129	13.74773
10	0.08000	187.77528	234.52348	-0.00000	0.00000	4.18051	25.08299
11	0.09000	235.77397	234.52348	-0.00000	0.00000	8.40058	50.40338
12	0.10000	291.69507	234.52348	-0.00000	-0.00000	-43.75214	-262.51221
13	0.11250	365.80444	234.52348	-0.00000	-0.00000	-62.85841	-377.14966
14	0.12500	450.43066	234.52348	-0.00000	-0.00001	-97.82593	-586.95435
15	0.13500	522.57178	234.52348	-0.00000	-0.00000	-92.66182	-555.96973
16	0.15000	639.27368	234.52348	-0.00000	-0.00001	-116.21716	-697.30151
17	0.16000	722.36401	234.52348	-0.00000	-0.00000	-74.54481	-447.86792
18	0.17000	809.78125	234.52348	-0.00000	-0.00001	-129.16513	-774.98926
19	0.18000	901.08569	234.52348	-0.00000	-0.00001	-116.10312	-696.61719
20	0.19000	996.39502	234.52348	-0.00000	-0.00001	-121.95221	-731.71167
21	0.20000	1095.25439	234.52348	-0.00000	-0.00001	-147.68588	-886.11353
22	0.21000	1197.64966	234.52349	-0.00000	-0.00001	-203.34579	-1220.07337
23	0.22000	1303.29712	234.52348	-0.00000	-0.00001	-147.22528	-883.34985
24	0.23000	1412.16650	234.52348	-0.00000	-0.00001	-117.70776	-706.24512
25	0.24000	1524.01587	234.52348	-0.00000	-0.00001	-241.24323	-1447.45654
26	0.25000	1638.74756	234.52348	-0.00000	-0.00001	-248.94891	-1493.69043
27	0.26000	1756.16821	234.52348	-0.00000	-0.00001	-283.35156	-1700.10596
28	0.27000	1876.16943	234.52348	-0.00000	-0.00001	-164.82158	-988.92749
29	0.28000	1998.59619	234.52348	-0.00000	-0.00001	-232.87595	-1397.25903
30	0.29000	2123.30515	234.52348	-0.00000	-0.00001	-159.26382	-955.58105
31	0.30000	2250.17261	234.52348	-0.00000	-0.00001	-270.44409	-1622.66138
32	0.31000	2379.06226	234.52348	-0.00000	-0.00001	-222.92513	-1337.55420
33	0.32000	2509.84375	234.52349	-0.00000	-0.00001	-196.38220	-1178.29077
34	0.33000	2642.41748	234.52348	-0.00000	-0.00001	-186.01973	-1116.11621
35	0.34000	2776.63525	234.52348	-0.00000	-0.00001	-189.22853	-1135.36890
36	0.35000	2912.42310	234.52348	-0.00000	-0.00001	-206.19533	-1237.16943
37	0.36000	3049.63967	234.52349	-0.00000	-0.00001	-238.30988	-1429.85645
38	0.37000	3188.22461	234.52348	-0.00000	-0.00002	-289.55308	-1737.91504
39	0.38000	3328.06543	234.52348	-0.00000	-0.00002	-365.65527	-2193.92725
40	0.39000	3467.08911	234.52348	-0.00000	-0.00001	-221.84923	-1331.09277
41	0.40000	36[illegible].21289	234.52349	-0.00000	-0.00002	-312.72217	-1876.32935
42	0.41000	37[illegible]34572	234.52348	-0.00000	-0.00001	-202.31960	-1213.90918
43	0.42000	3898.42261	234.58534	-40.59589	-52374.1289	******************	******************
44	0.43000	4043.30591	229.10009	-735.98511	-4022.81909	******************	******************
45	0.44000	4188.98437	222.56169	-599.52148	-709.81641	******************	******************
46	0.45000	4335.30469	215.65725	-582.11377	2577.82886	******************	******************
47	0.46000	4482.23828	208.83504	-582.53247	-384.53247	******************	******************
48	0.47000	4629.63281	202.01921	-531.50293	104572.875	******************	******************
49	0.48000	4777.57812	200.10693	-4.11779	-5975.06641	******************	******************
50	0.49000	4925.84766	200.10638	-0.00000	-0.00000	-33.56232	-201.37350
51	0.50000	5074.73437	201.58972	503.02832	130179.687	******************	******************
52	0.50500	5149.21875	204.85082	550.04321	-6795.85937	******************	******************
53	0.51250	5261.32031	209.62575	578.31592	-60131.4961	******************	******************
54	0.52250	5411.26562	212.66661	-11.51521	-5905.93750	******************	******************
55	0.53500	5599.14453	212.64659	-0.00000	-0.00000	-33.89087	-203.3448[illegible]

56	0.54500	5749.87891	212.64178	-8.25502	1818.72119	**************	**************
57	0.55500	5900.17578	225.94923	2256.33008	-138257.500	**************	**************
58	0.56250	6013.33203	234.53416	18.85678	-77523.0625	**************	**************
59	0.57250	6164.80078	234.52348	0.00000	0.00000	8.23444	49.40552
60	0.58250	6316.66016	234.52348	0.00000	0.00000	14.69263	88.15562
61	0.59250	6468.83984	234.52348	0.00000	0.00000	19.58591	117.51521
62	0.60500	6659.48437	234.52348	0.00000	0.00000	23.93765	143.62564
63	0.61250	6773.96484	234.52348	0.00000	0.00000	26.01506	156.09006
64	0.62250	6926.94922	234.52350	-0.00000	-0.00000	-9.15387	-54.92313
65	0.63500	7118.53125	234.52348	-0.00000	0.00000	50.96578	305.80005
66	0.64500	7272.19531	234.52348	-0.00000	-0.00002	-291.55884	-1749.34961
67	0.65500	7426.05859	234.52348	-0.00000	-0.00002	-296.38501	-1778.30564
68	0.67000	7657.28906	234.52348	-0.00000	-0.00002	-307.93335	-1847.59644
69	0.67500	7734.26953	234.52348	-0.00000	-0.00002	-310.14111	-1860.84302
70	0.68500	7888.37500	234.52348	-0.00000	-0.00002	-319.07764	-1914.46216
71	0.69500	8042.83984	234.52348	-0.00000	-0.00002	-332.36060	-1994.15967
72	0.70500	8197.47266	234.52348	-0.00000	-0.00002	-348.40381	-2090.41870
73	0.72000	8429.94531	234.52348	-0.00000	-0.00002	-381.17187	-2287.02686
74	0.73000	8585.06641	234.52348	-0.00000	-0.00002	-406.31982	-2437.91406
75	0.74000	8740.32812	234.52348	-0.00000	-0.00002	-434.77979	-2608.67358
76	0.75000	8895.69922	234.52348	-0.00000	-0.00002	-466.47095	-2798.82031
77	0.76000	9051.19531	234.52348	-0.00000	-0.00001	-226.95619	-1361.73437
78	0.77000	9206.77734	234.52348	-0.00000	-0.00001	-249.38283	-1496.29395
79	0.78000	9362.47266	234.52348	-0.00000	-0.00001	-274.65942	-1647.95337
80	0.79000	9518.23047	234.52348	-0.00000	-0.00002	-302.50269	-1815.01245
81	0.80000	9674.05859	234.52348	-0.00000	-0.00002	-333.22778	-1999.36279
82	0.81000	9829.92578	234.52348	-0.00000	-0.00002	-366.60132	-2199.60352
83	0.82000	9985.84375	234.52348	-0.00000	-0.00002	-403.02441	-2418.14185
84	0.83000	10141.7812	234.52348	-0.00000	-0.00002	-442.20505	-2653.23120
85	0.84000	10297.7500	234.52348	-0.00000	-0.00003	-484.52881	-2907.16724
86	0.85000	10453.7578	234.52348	-0.00000	-0.00001	-242.84218	-1457.05005
87	0.86000	10609.7539	234.52348	-0.00000	-0.00001	-271.36470	-1626.38501
88	0.87000	10765.8125	234.55070	36.36185	17467.6211	**************	**************
89	0.88000	10921.7773	231.31117	-794.02271	32367.1094	**************	**************
90	0.89000	11077.8203	223.58488	-748.67041	4157.33203	**************	**************
91	0.90000	11233.7461	216.20352	-729.90454	5022.26562	**************	**************
92	0.91000	11389.6211	208.96504	-722.76050	-10.65334	-202971392.	**************
93	0.92000	11545.3555	201.79027	-613.78931	178072.937	**************	**************
94	0.93000	11701.0273	200.10577	0.27976	3915.98169	**************	**************
95	0.94000	11856.6211	200.10638	-0.00192	-28.09261	-535230976.	**************
96	0.95000	12012.2187	202.75917	576.70801	-30854.3164	**************	**************
97	0.96000	12167.8555	209.41251	908.18799	115447.125	**************	**************
98	0.96500	12246.4687	212.25015	257.36304	-155735.625	**************	**************
99	0.97125	12343.7461	212.64717	-9.78471	11624.3750	**************	**************
100	0.98250	12519.7617	212.64659	-0.00000	-0.00000	-74.58969	-447.53711
101	0.99250	12674.2852	212.54045	13.71201	55258.5781	**************	**************
102	1.00250	12829.2256	223.67275	2153.47485	-141739.625	**************	**************
103	1.01250	12984.8857	234.52015	-3.94382	14489.3789	**************	**************
104	1.02249	13140.6602	234.52348	0.00000	0.00000	43.78688	262.72070
105	1.03499	13335.4492	234.52348	0.00000	0.00000	36.74556	220.47284
106	1.04499	13491.3047	234.52348	0.00000	0.00000	30.04503	180.26982
107	1.05499	13647.1552	234.52348	0.00000	0.00000	22.15097	132.90556
108	1.06249	13763.8477	234.52348	0.00000	0.00000	16.71188	100.27110
109	1.07248	13919.5641	234.52348	-0.00000	0.00000	68.14453	408.86621
110	1.08498	14114.4258	234.52348	-0.00000	-0.00002	-340.24731	-2041.47998
111	1.09498	14270.3242	234.52348	-0.00000	-0.00002	-470.06885	-2820.40747
112	1.10999	14504.1875	234.52348	-0.00000	-0.00001	-245.61774	-1473.70337
113	1.11998	14659.8672	234.52348	-0.00000	-0.00001	-269.20752	-1615.24194
114	1.12998	14815.3594	234.52348	-0.00000	-0.00002	-292.39355	-1754.35791
115	1.13998	14971.0536	234.52348	-0.00000	-0.00002	-321.14062	-1926.83984

116	1.14998	15126.3359	234.52348	-0.00000	-0.00002	-345.26782	-2071.60278
117	1.15998	15282.2930	234.52348	-0.00000	-0.00002	-380.74534	-2284.47363
118	1.16998	15437.8828	234.52348	-0.00000	-0.00002	-413.07910	-2478.46973
119	1.17997	15593.7461	234.52348	-0.00000	-0.00002	-451.78320	-2710.69385
120	1.18997	15749.5703	234.52348	-0.00000	-0.00001	-221.07011	-1326.41797
121	1.19997	15905.4648	234.52348	-0.00000	-0.00001	-246.42064	-1478.51978
122	1.20997	16061.3984	234.52348	-0.00000	-0.00001	-274.23437	-1645.40308
123	1.21997	16217.3555	234.52348	-0.00000	-0.00002	-304.48950	-1826.93335
124	1.22997	16373.3437	234.52348	-0.00000	-0.00002	-337.44971	-2024.69434
125	1.23997	16529.3398	234.52348	-0.00000	-0.00002	-372.95068	-2237.69971
126	1.24997	16685.3477	234.52348	-0.00000	-0.00002	-411.12085	-2466.72021
127	1.25997	16841.3516	234.52348	-0.00000	-0.00002	-451.83374	-2710.99707
128	1.26997	16997.3828	234.52348	-0.00000	-0.00001	-223.03596	-1338.21899
129	1.27997	17153.4023	234.52348	-0.00000	-0.00001	-249.88055	-1499.32837
130	1.28997	17309.4453	234.52348	-0.00000	-0.00001	-279.22827	-1675.36621
131	1.29997	17465.4922	234.52348	-0.00000	-0.00002	-311.00879	-1866.04907
132	1.30997	17621.5430	234.52350	0.01983	62.98404	1199995140.	7199956990.
133	1.31996	17777.5977	233.76839	-452.35791	-160017.937	***************	***************
134	1.32996	17933.5195	226.22756	-764.33105	7562.35156	***************	***************
135	1.33996	18089.5859	218.75067	-734.59863	8545.01953	***************	***************
136	1.34996	18245.3203	211.48796	-723.32178	925.69653	***************	***************
137	1.35996	18401.2070	204.25688	-722.24927	-1757.20190	***************	***************
138	1.36996	18556.7617	200.10675	-15.69490	28860.2070	***************	***************
139	1.37996	18712.3516	200.10638	-0.00000	-0.00000	-48.86787	-293.20554
140	1.38996	18867.8906	200.72810	352.67822	144517.312	***************	***************
141	1.39996	19023.3857	207.16837	560.22314	-2773.41187	***************	***************
142	1.40496	19101.3008	210.39035	509.10767	-43499.4727	***************	***************
143	1.41246	19218.5000	212.67242	-3.42737	-20699.8320	***************	***************
144	1.42245	19374.8672	212.64659	-0.00000	-0.00000	-93.97830	-563.86865
145	1.43495	19570.0273	212.64188	-8.56004	1569.64282	***************	***************
146	1.44495	19725.3555	226.66765	2282.45020	-144244.562	***************	***************
147	1.45245	19842.0781	234.53221	-14.36829	-32548.9844	***************	***************
148	1.46245	19997.9756	234.52348	0.00000	0.00000	31.54199	189.25159
149	1.47244	20153.9023	234.52348	0.00000	0.00000	23.79404	142.76395
150	1.48244	20309.8320	234.52348	0.00000	0.00000	14.97132	89.82775
151	1.49244	20465.7461	234.52348	0.00000	0.00000	5.08766	30.52592
152	1.50244	20621.5078	234.52348	0.00000	-0.00000	-4.68865	-28.13187
153	1.51118	20757.9609	234.52350	-0.00000	-0.00000	-18.77205	-112.63206
154	1.52118	20914.1133	234.52349	-0.00000	0.00000	39.07079	234.42426
155	1.53118	21070.2578	234.52348	-0.00000	-0.00001	-140.40285	-842.41528
156	1.54118	21226.3828	234.52348	-0.00000	-0.00002	-294.02319	-1764.13574
157	1.55618	21460.5156	234.52348	-0.00000	-0.00002	-344.89795	-2069.38354
158	1.56617	21616.5654	234.52348	-0.00000	-0.00002	-382.54712	-2295.27832
159	1.57617	21773.1328	234.52348	-0.00000	-0.00002	-431.75757	-2590.54028
160	1.58117	21851.1875	234.52348	-0.00000	-0.00002	-453.52769	-2721.76074
161	1.59117	22007.5352	234.52348	-0.00000	-0.00001	-226.85188	-1361.10864
162	1.60117	22163.8857	234.52348	-0.00000	-0.00001	-257.58740	-1545.52124
163	1.61117	22320.1328	234.52348	-0.00000	-0.00002	-289.96924	-1739.81201
164	1.62517	22554.5156	234.52348	-0.00000	-0.00002	-343.94385	-2063.65394
165	1.63617	22710.6494	234.52348	-0.00000	-0.00002	-381.84912	-2291.09033
166	1.64616	22865.7959	234.52348	-0.0000	-0.00002	-422.87744	-2537.25977
167	1.65616	23022.9102	234.52348	-0.00000	-0.00002	-466.28247	-2797.68945
168	1.66616	23179.0117	234.52348	-0.00000	-0.00001	-232.45061	-1394.70093
169	1.67616	23335.1250	234.52348	-0.00000	-0.00001	-261.12866	-1566.76880
170	1.68616	23491.2227	234.52348	-0.00000	-0.00002	-292.06104	-1752.36279
171	1.69616	23647.3154	234.52348	-0.00000	-0.00002	-325.43140	-1952.58447
172	1.70616	23803.4052	234.52348	-0.00000	-0.00002	-361.34741	-2168.08032
173	1.71616	23959.4844	234.52348	-0.00000	-0.00002	-399.77783	-2398.66235
174	1.72616	24115.5547	234.52348	-0.00000	-0.00002	-440.86450	-2645.18189
175	1.73616	24271.6406	234.52348	-0.00000	-0.00003	-485.09933	-2910.53538

176	1.74616	24427.7070	234.52348	-0.00000	-0.00001	-243.67404	-1462.04126
177	1.75616	24583.8125	234.54547	33.47658	21106.9453	********************************	
178	1.76616	24739.8242	231.42702	-799.05811	32989.2227	********************************	
179	1.77616	24895.9062	223.69290	-749.49780	4997.58594	********************************	
180	1.78615	25051.8789	216.30669	-730.71021	3931.09229	********************************	
181	1.79615	25207.7773	209.06615	-722.86450	-504.32617	********************************	
182	1.80615	25363.5508	201.87627	-636.58813	152538.562	********************************	
183	1.81615	25519.2383	200.10345	-2.99287	13588.1562	********************************	
184	1.82615	25674.9258	200.10638	0.00120	-19.82202	-377656576.	***************
185	1.83615	25830.5391	202.70125	681.05884	-30926.1484	********************************	
186	1.84615	25986.3437	209.30585	890.69189	183970.625	********************************	
187	1.85115	26064.9687	212.22864	277.07007	-167765.937	********************************	
188	1.86114	26220.6094	212.64661	0.40319	495.22192	9435156480.	***************
189	1.87114	26376.2422	212.64659	-0.00000	-0.00000	-68.65422	-411.92432
190	1.88114	26531.5703	213.13673	542.60962	396777.437	********************************	
191	1.89114	26686.6836	233.06734	1258.96484	-828540.937	********************************	
192	1.90114	26842.5742	234.52353	0.08090	-192.05516	********************************	
193	1.91113	26998.4756	234.52348	0.00000	0.00000	45.75195	274.51099
194	1.92113	27154.3594	234.52348	0.00000	0.00000	40.43474	242.60796
195	1.93113	27310.2695	234.52348	0.00000	0.00000	34.14438	204.86588
196	1.94113	27466.1719	234.52348	0.00000	0.00000	26.65042	159.90222
197	1.95113	27621.8359	234.52348	0.00000	0.00000	19.43445	116.60546
198	1.96112	27777.7344	234.52348	-0.00000	0.00000	66.50893	399.05273
199	1.97112	27933.6953	234.52348	-0.00000	-0.00002	-327.61938	-1965.71240
200	1.98112	28089.5977	234.52348	-0.00000	-0.00002	-451.56689	-2709.39500
201	1.99112	28245.5000	234.52349	-0.00000	-0.00001	-221.58148	-1329.48633
202	2.00112	28401.3437	234.52349	-0.00000	-0.00001	-245.48997	-1472.93701
203	2.01112	28556.8555	234.52348	-0.00000	-0.00001	-267.17456	-1603.04419
204	2.02111	28712.2891	234.52349	-0.00000	-0.00002	-290.57593	-1743.45215
205	2.03111	28867.8984	234.52348	-0.00000	-0.00002	-317.63354	-1905.79761
206	2.04111	29023.5703	234.52349	-0.00000	-0.00002	-347.25464	-2083.52363
207	2.05111	29179.2812	234.52349	-0.00000	-0.00002	-379.35425	-2276.12109
208	2.06111	29335.1094	234.52348	-0.00000	-0.00002	-415.27490	-2491.64453
209	2.07111	29490.9727	234.52349	-0.00000	-0.00002	-454.11255	-2724.66992
210	2.08111	29646.9023	234.52349	-0.00000	-0.00001	-223.51184	-1341.06835
211	2.09111	29802.8398	234.52348	-0.00000	-0.00001	-249.49037	-1496.93921
212	2.10111	29959.8291	234.52349	-0.00000	-0.00001	-278.18359	-1669.09814
213	2.11111	30114.8164	234.52349	-0.00000	-0.00002	-309.15430	-1854.92212
214	2.12111	30270.8242	234.52349	-0.00000	-0.00002	-342.70579	-2056.23557
215	2.13110	30426.8359	234.52348	-0.00000	-0.00002	-378.78735	-2272.71973
216	2.14110	30582.8633	234.52348	-0.00000	-0.00002	-417.71387	-2506.27832
217	2.15110	30738.8945	234.52349	-0.00000	-0.00002	-459.38843	-2756.32520
218	2.16110	30894.9375	234.52348	-0.00000	-0.00001	-227.71230	-1366.27100
219	2.17110	31050.9844	234.52349	-0.00000	-0.00001	-255.22888	-1531.37036
220	2.18110	31207.0117	234.52348	-0.00000	-0.00001	-284.78809	-1708.72510
221	2.19110	31363.0703	234.52349	-0.00000	-0.00002	-317.16040	-1902.95874
222	2.20110	31519.0977	234.59750	19.73795	-31759.2266	********************************	
223	2.21110	31675.1172	230.14964	-782.64722	1300.20898	********************************	
224	2.22110	31831.1250	222.49153	-747.35913	-701.50317	********************************	
225	2.23110	31987.0586	215.14220	-725.09814	-3933.22290	********************************	
226	2.24110	32142.9687	207.90694	-720.36572	2054.42334	********************************	
227	2.25110	32298.6211	201.05211	-427.29321	114157.250	********************************	
228	2.26110	32454.4023	200.10641	-0.07127	-234.84541	********************************	
229	2.27109	32609.8281	200.10591	1.98808	9439.78516	********************************	
230	2.28109	32765.5195	203.79262	687.03076	-486.31665	********************************	
231	2.29109	32921.0312	210.37354	509.21875	-66434.5625	********************************	
232	2.30109	33077.3828	212.64545	-7.17527	14249.5234	********************************	
233	2.31109	33232.8906	212.64659	-0.00000	-0.00000	-89.20090	-535.20435
234	2.32359	33427.0625	212.77570	226.78322	274140.375	********************************	
235	2.33108	33543.1328	225.28949	2308.94678	-145219.687	********************************	

236	2.34108	33698.7656	234.52475	-0.73926	-8782.22266******************************		
237	2.35107	33854.7461	234.52348	0.00000	0.00000	37.10735	222.64365
238	2.36107	34010.7656	234.52348	0.00000	0.00000	29.89752	179.38478
239	2.37107	34166.7500	234.52348	0.00000	0.00000	21.70374	130.22215
240	2.38357	34361.7266	234.52348	0.00000	0.00000	9.75488	58.52914
241	2.39106	34478.4844	234.52349	0.00000	0.00000	2.93563	17.61374
242	2.40106	34634.5547	234.52350	0.00000	0.00001	122.33482	734.00732
243	2.41106	34790.6055	234.52348	0.00000	0.00000	84.07433	504.44482
244	2.42105	34946.6836	234.52348	-0.00000	-0.00001	-245.31941	-1471.90747
245	2.43105	35102.7344	234.52348	-0.00000	-0.00001	-274.31494	-1645.88647
246	2.44105	35258.7812	234.52348	-0.00000	-0.00002	-306.23533	-1837.41431
247	2.45105	35415.2852	234.52348	-0.00000	-0.00002	-347.16382	-2082.97876
248	2.46105	35571.8242	234.52348	-0.00000	-0.00002	-390.07227	-2340.42896
249	2.47105	35727.7422	234.52348	-0.00000	-0.00002	-428.10156	-2568.60425
250	2.48104	35883.7148	234.52348	-0.00000	-0.00002	-469.50269	-2817.01050
251	2.49104	36039.7461	234.52348	-0.00000	-0.00001	-233.77546	-1402.65601
252	2.50104	36195.7891	234.52348	-0.00000	-0.00001	-261.78394	-1570.70044
253	2.50104	0.0	0.0	0.0	0.0	0.0	0.0

LOCK MECHANISM (LM) PARAMETERS

	TIME (SEC)	MOTOR (DEG)	LM ROTATION (DEG)	LM VELOCITY (DEG/SEC)	LM ACCELERATION (DEG/SEC**2)	LOCK TORQUE (FT-LBS)	GEAR FORCE (LBS)
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	14.79820	0.0	-0.00000	-0.00000	-0.00000
3	0.01000	2.95483	14.79820	-0.00000	-0.00000	-0.00000	-0.00000
4	0.02000	11.81933	14.79820	-0.00000	-0.00000	-0.00000	-0.00000
5	0.03000	26.59344	14.79820	-0.00000	-0.00000	-0.00000	-0.00000
6	0.04000	47.27733	14.79820	-0.00000	-0.00000	-0.00000	-0.00000
7	0.05000	73.82103	14.79820	-0.00000	0.00000	0.00000	0.00000
8	0.06000	106.15965	14.79820	-0.00000	0.00000	0.00000	0.00000
9	0.07000	144.19530	14.79820	-0.00000	0.00000	0.00000	0.00000
10	0.08000	187.77528	14.79820	-0.00000	0.00000	0.00000	0.00000
11	0.09000	236.77397	14.79820	0.00000	0.00000	0.00000	0.00000
12	0.10000	291.69507	14.79820	-0.00000	-0.00000	-0.00000	-0.00000
13	0.11250	366.80444	14.79820	-0.00000	-0.00000	-0.00000	-0.00000
14	0.12500	450.43066	14.79820	-0.00000	-0.00000	-0.00000	-0.00000
15	0.13500	522.57178	14.79820	-0.00000	-0.00000	-0.00000	-0.00000
16	0.15000	639.27368	13.49980	-263.72998	-29920.1406	-1.07851	-6.70590
17	0.16000	722.35401	9.63173	-284.09521	75910.6875	2.73269	16.99120
18	0.17000	809.73125	4.68554	-232.81740	109133.812	3.93387	24.45979
19	0.18000	901.03569	1.36282	-157.47238	103220.125	3.72070	23.13438
20	0.19000	995.39502	0.19435	-62.28990	11561.9062	0.41576	2.59133
21	0.20000	1095.25439	0.00004	0.31376	-105.06491	-0.00382	-0.02377
22	0.21000	1197.64966	0.0	0.0	0.0	0.0	0.0
23	0.22000	1303.29712	0.0	0.0	0.0	0.0	0.0
24	0.23000	1412.16650	0.0	0.0	0.0	0.0	0.0
25	0.24000	1524.01587	0.0	0.0	0.0	0.0	0.0
26	0.25000	1638.74756	0.0	0.0	0.0	0.0	0.0
27	0.26000	1756.16821	0.0	0.0	0.0	0.0	0.0
28	0.27000	1875.16943	0.0	0.0	0.0	0.0	0.0
29	0.28000	1998.59519	0.0	0.0	0.0	0.0	0.0
30	0.29000	2123.30515	0.0	0.0	0.0	0.0	0.0
31	0.30000	2250.17261	0.0	0.0	0.0	0.0	0.0
32	0.31000	2379.06226	0.0	0.0	0.0	0.0	0.0
33	0.32000	2509.84375	0.0	0.0	0.0	0.0	0.0
34	0.33000	2642.41748	0.0	0.0	0.0	0.0	0.0
35	0.34000	2776.63525	0.0	0.0	0.0	0.0	0.0
36	0.35000	2912.42310	0.0	0.0	0.0	0.0	0.0
37	0.36000	3049.63867	0.0	0.0	0.0	0.0	0.0
38	0.37000	3188.22461	0.0	0.0	0.0	0.0	0.0
39	0.38000	3328.06543	0.0	0.0	0.0	0.0	0.0
40	0.39000	3469.08911	0.0	0.0	0.0	0.0	0.0
41	0.40000	3611.21289	0.0	0.0	0.0	0.0	0.0
42	0.41000	3754.34692	0.0	0.0	0.0	0.0	0.0
43	0.42000	3898.42261	0.0	0.0	0.0	0.0	0.0
44	0.43000	4043.30591	0.0	0.0	0.0	0.0	0.0
45	0.44000	4188.98437	0.0	0.0	0.0	0.0	0.0
46	0.45000	4335.30469	0.0	0.0	0.0	0.0	0.0
47	0.46000	4482.23828	0.0	0.0	0.0	0.0	0.0
48	0.47000	4629.63281	0.0	0.0	0.0	0.0	0.0
49	0.48000	4777.57812	0.0	0.0	0.0	0.0	0.0
50	0.49000	4925.84766	0.0	0.0	0.0	0.0	0.0
51	0.50000	5074.73437	0.0	0.0	0.0	0.0	0.0
52	0.50500	5149.21875	0.0	0.0	0.0	0.0	0.0
53	0.51250	5261.32031	0.0	0.0	0.0	0.0	0.0
54	0.52250	5411.26562	0.0	0.0	0.0	0.0	0.0
55	0.53500	5599.14453	0.0	0.0	0.0	0.0	0.0

56	0.54500	5749.87891	0.0	0.0	0.0	0.0	0.0
57	0.55500	5900.17578	0.0	0.0	0.0	0.0	0.0
58	0.56250	6013.33203	0.0	0.0	0.0	0.0	0.0
59	0.57250	6164.80078	0.0	0.0	0.0	0.0	0.0
60	0.58250	6316.66016	0.0	0.0	0.0	0.0	0.0
61	0.59250	6468.83984	0.0	0.0	0.0	0.0	0.0
62	0.60500	6659.48437	0.52503	343.57397	140708.062	5.07200	31.53642
63	0.61250	6773.96484	8.74693	1807.48169	8490.37109	0.30605	1.90292
64	0.62250	6926.94922	14.79820	-0.00000	-0.00000	-0.00000	-0.00000
65	0.63500	7118.53125	14.79820	-0.00000	0.00000	0.00000	0.00000
66	0.64500	7272.19531	14.79820	-0.00000	-0.00000	-0.00000	-0.00000
67	0.65500	7426.05859	14.79820	-0.00000	-0.00000	-0.00000	-0.00000
68	0.67000	7657.28906	8.63449	-847.23975	-303273.500	-10.93188	-67.97166
69	0.67500	7734.26953	4.28102	-365.24023	-26309.1562	-0.94931	-5.89636
70	0.68500	7888.37500	0.33322	-138.70840	38894.3477	1.40200	8.71726
71	0.69500	8042.83984	0.00000	0.23856	320.76099	0.01156	0.07189
72	0.70500	8197.47266	0.0	0.0	0.0	0.0	0.0
73	0.72000	8429.94531	0.0	0.0	0.0	0.0	0.0
74	0.73000	8585.06641	0.0	0.0	0.0	0.0	0.0
75	0.74000	8740.32812	0.0	0.0	0.0	0.0	0.0
76	0.75000	8895.69922	0.0	0.0	0.0	0.0	0.0
77	0.76000	9051.19531	0.0	0.0	0.0	0.0	0.0
78	0.77000	9206.77734	0.0	0.0	0.0	0.0	0.0
79	0.78000	9362.47266	0.0	0.0	0.0	0.0	0.0
80	0.79000	9518.23047	0.0	0.0	0.0	0.0	0.0
81	0.80000	9674.05859	0.0	0.0	0.0	0.0	0.0
82	0.81000	9829.92578	0.0	0.0	0.0	0.0	0.0
83	0.82000	9985.84375	0.0	0.0	0.0	0.0	0.0
84	0.83000	10141.7812	0.0	0.0	0.0	0.0	0.0
85	0.84000	10297.7500	0.0	0.0	0.0	0.0	0.0
86	0.85000	10453.7578	0.0	0.0	0.0	0.0	0.0
87	0.86000	10609.7539	0.0	0.0	0.0	0.0	0.0
88	0.87000	10765.8125	0.0	0.0	0.0	0.0	0.0
89	0.88000	10921.7773	0.0	0.0	0.0	0.0	0.0
90	0.89000	11077.8203	0.0	0.0	0.0	0.0	0.0
91	0.90000	11233.7461	0.0	0.0	0.0	0.0	0.0
92	0.91000	11389.6211	0.0	0.0	0.0	0.0	0.0
93	0.92000	11545.3555	0.0	0.0	0.0	0.0	0.0
94	0.93000	11701.0273	0.0	0.0	0.0	0.0	0.0
95	0.94000	11856.6211	0.0	0.0	0.0	0.0	0.0
96	0.95000	12012.2187	0.0	0.0	0.0	0.0	0.0
97	0.96000	12167.8555	0.0	0.0	0.0	0.0	0.0
98	0.96500	12246.4687	0.0	0.0	0.0	0.0	0.0
99	0.97125	12343.7461	0.0	0.0	0.0	0.0	0.0
100	0.98250	12518.7617	0.0	0.0	0.0	0.0	0.0
101	0.99250	12674.2852	0.0	0.0	0.0	0.0	0.0
102	1.00250	12829.2266	0.0	0.0	0.0	0.0	0.0
103	1.01250	12984.8857	0.0	0.0	0.0	0.0	0.0
104	1.02249	13140.6602	0.0	0.0	0.0	0.0	0.0
105	1.03499	13335.4492	0.0	0.0	0.0	0.0	0.0
106	1.04499	13491.3047	0.00009	0.06983	-155.29436	-0.00596	-0.03705
107	1.05499	13647.1562	4.57923	1433.65088	287590.937	10.36622	64.45454
108	1.06249	13763.8477	13.47413	265.13184	-81294.6875	-2.93037	-18.22031
109	1.07248	13919.6641	14.79820	0.00000	0.00000	0.00000	0.00000
110	1.08498	14114.4258	14.79820	-0.00000	-0.00000	-0.00000	-0.00000
111	1.09499	14270.3242	14.79820	-0.00000	-0.00000	-0.00000	-0.00000
112	1.10998	14504.1875	11.59955	-825.12573	-97548.1250	-3.51525	-21.86313
113	1.11998	14659.8672	3.79769	-508.50391	-259248.687	-9.34495	-58.10452
114	1.12998	14815.3594	0.20327	-102.37206	30422.4297	1.09661	6.51847
115	1.13998	14971.0536	-0.00000	0.01475	24.87752	0.00090	0.00558

116	1.14998	15126.3359	0.0	0.0	0.0	0.0	0.0
117	1.15998	15282.2930	0.0	0.0	0.0	0.0	0.0
118	1.16998	15437.8828	0.0	0.0	0.0	0.0	0.0
119	1.17997	15593.7461	0.0	0.0	0.0	0.0	0.0
120	1.18997	15749.5703	0.0	0.0	0.0	0.0	0.0
121	1.19997	15905.4648	0.0	0.0	0.0	0.0	0.0
122	1.20997	16061.3984	0.0	0.0	0.0	0.0	0.0
123	1.21997	16217.3555	0.0	0.0	0.0	0.0	0.0
124	1.22997	16373.3437	0.0	0.0	0.0	0.0	0.0
125	1.23997	16529.3398	0.0	0.0	0.0	0.0	0.0
126	1.24997	16685.3477	0.0	0.0	0.0	0.0	0.0
127	1.25997	16841.3516	0.0	0.0	0.0	0.0	0.0
128	1.26997	16997.3828	0.0	0.0	0.0	0.0	0.0
129	1.27997	17153.4023	0.0	0.0	0.0	0.0	0.0
130	1.28997	17309.4453	0.0	0.0	0.0	0.0	0.0
131	1.29997	17465.4922	0.0	0.0	0.0	0.0	0.0
132	1.30997	17621.5430	0.0	0.0	0.0	0.0	0.0
133	1.31996	17777.5977	0.0	0.0	0.0	0.0	0.0
134	1.32996	17933.5195	0.0	0.0	0.0	0.0	0.0
135	1.33996	18089.5859	0.0	0.0	0.0	0.0	0.0
136	1.34996	18245.3203	0.0	0.0	0.0	0.0	0.0
137	1.35996	18401.2070	0.0	0.0	0.0	0.0	0.0
138	1.36996	18556.7617	0.0	0.0	0.0	0.0	0.0
139	1.37996	18712.3516	0.0	0.0	0.0	0.0	0.0
140	1.38996	18867.8906	0.0	0.0	0.0	0.0	0.0
141	1.39996	19023.3867	0.0	0.0	0.0	0.0	0.0
142	1.40496	19101.3008	0.0	0.0	0.0	0.0	0.0
143	1.41246	19218.5000	0.0	0.0	0.0	0.0	0.0
144	1.42245	19374.8672	0.0	0.0	0.0	0.0	0.0
145	1.43495	19570.0273	0.0	0.0	0.0	0.0	0.0
146	1.44495	19725.3555	0.0	0.0	0.0	0.0	0.0
147	1.45245	19842.0781	0.0	0.0	0.0	0.0	0.0
148	1.46245	19997.9766	0.0	0.0	0.0	0.0	0.0
149	1.47244	20153.9023	0.0	0.0	0.0	0.0	0.0
150	1.48244	20309.8320	0.0	0.0	0.0	0.0	0.0
151	1.49244	20465.7461	0.26572	234.20360	121949.000	4.39220	27.30960
152	1.50244	20621.5078	11.91409	1370.95898	-879457.625	-31.70262	-197.11987
153	1.51118	20757.9609	14.79820	-0.00000	-0.00000	-0.00000	-0.00000
154	1.52118	20914.1133	14.79820	-0.00000	0.00000	0.00000	0.00000
155	1.53118	21070.2578	14.79820	0.00000	-0.00000	-0.00000	-0.00000
156	1.54118	21226.3828	14.79820	-0.00000	-0.00000	-0.00000	-0.00000
157	1.55618	21460.5156	9.39971	-448.11060	-1805.88159	-0.06510	-0.40475
158	1.56617	21616.5664	1.75526	-624.19434	273331.750	9.85259	61.26093
159	1.57617	21773.1328	0.02371	-24.69957	16966.1797	0.51157	3.80257
160	1.58117	21851.1875	-0.00029	-0.52213	979.76270	0.03532	0.21959
161	1.59117	22007.5352	0.0	0.0	0.0	0.0	0.0
162	1.60117	22163.8867	0.0	0.0	0.0	0.0	0.0
163	1.61117	22320.1328	0.0	0.0	0.0	0.0	0.0
164	1.62617	22554.5156	0.0	0.0	0.0	0.0	0.0
165	1.63617	22710.6484	0.0	0.0	0.0	0.0	0.0
166	1.64616	22866.7959	0.0	0.0	0.0	0.0	0.0
167	1.65616	23022.9102	0.0	0.0	0.0	0.0	0.0
168	1.66616	23179.0117	0.0	0.0	0.0	0.0	0.0
169	1.67616	23335.1250	0.0	0.0	0.0	0.0	0.0
170	1.68616	23491.2227	0.0	0.0	0.0	0.0	0.0
171	1.69616	23647.3164	0.0	0.0	0.0	0.0	0.0
172	1.70616	23803.4062	0.0	0.0	0.0	0.0	0.0
173	1.71616	23959.4844	0.0	0.0	0.0	0.0	0.0
174	1.72616	24115.5547	0.0	0.0	0.0	0.0	0.0
175	1.73616	24271.6406	0.0	0.0	0.0	0.0	0.0

176	1.74616	24427.7070	0.0	0.0	0.0	0.0	0.0
177	1.75616	24583.8125	0.0	0.0	0.0	0.0	0.0
178	1.76616	24739.8242	0.0	0.0	0.0	0.0	0.0
179	1.77616	24895.9062	0.0	0.0	0.0	0.0	0.0
180	1.78615	25051.8789	0.0	0.0	0.0	0.0	0.0
181	1.79615	25207.7773	0.0	0.0	0.0	0.0	0.0
182	1.80615	25363.5508	0.0	0.0	0.0	0.0	0.0
183	1.81615	25519.2383	0.0	0.0	0.0	0.0	0.0
184	1.82615	25674.9258	0.0	0.0	0.0	0.0	0.0
185	1.83615	25830.5391	0.0	0.0	0.0	0.0	0.0
186	1.84615	25986.3437	0.0	0.0	0.0	0.0	0.0
187	1.85115	26064.9687	0.0	0.0	0.0	0.0	0.0
188	1.86114	26220.6094	0.0	0.0	0.0	0.0	0.0
189	1.87114	26375.2422	0.0	0.0	0.0	0.0	0.0
190	1.88114	26531.5703	0.0	0.0	0.0	0.0	0.0
191	1.89114	26686.6836	0.0	0.0	0.0	0.0	0.0
192	1.90114	26842.5742	0.0	0.0	0.0	0.0	0.0
193	1.91113	26998.4766	0.0	0.0	0.0	0.0	0.0
194	1.92113	27154.3594	0.0	0.0	0.0	0.0	0.0
195	1.93113	27310.2695	0.00009	0.08143	-121.51236	-0.00438	-0.02723
196	1.94113	27465.1719	4.55863	1409.77856	285168.875	10.27928	63.91394
197	1.95113	27621.8359	14.03304	484.17114	56128.9492	2.02324	12.57999
198	1.96112	27777.7344	14.79820	0.00000	0.00000	0.00000	0.00000
199	1.97112	27933.6953	14.79820	-0.00000	-0.00000	-0.00000	-0.00000
200	1.98112	28089.5977	14.79820	-0.00000	-0.00000	-0.00000	-0.00000
201	1.99112	28245.5000	14.36909	-283.46021	-92964.5625	-3.35103	-20.83583
202	2.00112	28401.3437	7.85521	-980.89307	-218526.187	-7.87706	-48.97754
203	2.01112	28556.8555	1.15367	-149.12509	49077.3594	1.76906	10.99954
204	2.02111	28712.2891	0.00096	-3.26076	6164.60937	0.22221	1.38165
205	2.03111	28867.8984	0.0	0.0	0.0	0.0	0.0
206	2.04111	29023.5703	0.0	0.0	0.0	0.0	0.0
207	2.05111	29179.2812	0.0	0.0	0.0	0.0	0.0
208	2.06111	29335.1094	0.0	0.0	0.0	0.0	0.0
209	2.07111	29490.9727	0.0	0.0	0.0	0.0	0.0
210	2.08111	29646.9023	0.0	0.0	0.0	0.0	0.0
211	2.09111	29802.8398	0.0	0.0	0.0	0.0	0.0
212	2.10111	29958.8281	0.0	0.0	0.0	0.0	0.0
213	2.11111	30114.8154	0.0	0.0	0.0	0.0	0.0
214	2.12111	30270.8242	0.0	0.0	0.0	0.0	0.0
215	2.13110	30426.8359	0.0	0.0	0.0	0.0	0.0
216	2.14110	30582.8633	0.0	0.0	0.0	0.0	0.0
217	2.15110	30738.8945	0.0	0.0	0.0	0.0	0.0
218	2.16110	30894.9375	0.0	0.0	0.0	0.0	0.0
219	2.17110	31050.9844	0.0	0.0	0.0	0.0	0.0
220	2.18110	31207.0117	0.0	0.0	0.0	0.0	0.0
221	2.19110	31363.0703	0.0	0.0	0.0	0.0	0.0
222	2.20110	31519.0977	0.0	0.0	0.0	0.0	0.0
223	2.21110	31675.1172	0.0	0.0	0.0	0.0	0.0
224	2.22110	31831.1250	0.0	0.0	0.0	0.0	0.0
225	2.23110	31987.0586	0.0	0.0	0.0	0.0	0.0
226	2.24110	32142.9687	0.0	0.0	0.0	0.0	0.0
227	2.25110	32298.6211	0.0	0.0	0.0	0.0	0.0
228	2.26110	32454.4023	0.0	0.0	0.0	0.0	0.0
229	2.27109	32609.8281	0.0	0.0	0.0	0.0	0.0
230	2.28109	32765.5195	0.0	0.0	0.0	0.0	0.0
231	2.29109	32921.0312	0.0	0.0	0.0	0.0	0.0
232	2.30109	33077.3828	0.0	0.0	0.0	0.0	0.0
233	2.31109	33232.8906	0.0	0.0	0.0	0.0	0.0
234	2.32358	33427.0625	0.0	0.0	0.0	0.0	0.0
235	2.33109	33543.1328	0.0	0.0	0.0	0.0	0.0

236	2.34108	33698.7656	0.0	0.0	0.0	0.0	0.0
237	2.35107	33854.7461	0.0	0.0	0.0	0.0	0.0
238	2.36107	34010.7656	0.0	0.0	0.0	0.0	0.0
239	2.37107	34166.7500	0.0	0.0	0.0	0.0	0.0
240	2.38357	34361.7256	3.35762	1155.68896	258160.500	9.30572	57.96063
241	2.39106	34478.4844	13.18603	250.76680	-33013.5781	-1.19002	-7.39922
242	2.40106	34634.5547	14.79820	0.00000	0.00000	0.00000	0.00000
243	2.41106	34790.6055	14.79820	0.00000	0.00000	0.00000	0.00000
244	2.42105	34946.6836	14.79820	-0.00000	-0.00000	-0.00000	-0.00000
245	2.43105	35102.7344	14.30006	-2.27782	-20069.6836	-0.72344	-4.49815
246	2.44105	35258.7812	10.25148	-855.36108	155338.625	5.59938	34.81552
247	2.45105	35415.2852	2.74294	-692.38916	-27133.9062	-0.97308	-6.08143
248	2.46105	35571.8242	0.07647	-53.07686	24359.8359	0.87908	5.45969
249	2.47105	35727.7422	0.0	0.0	0.0	0.0	0.0
250	2.48104	35883.7148	0.0	0.0	0.0	0.0	0.0
251	2.49104	36039.7461	0.0	0.0	0.0	0.0	0.0
252	2.50104	36195.7891	0.0	0.0	0.0	0.0	0.0
253	2.50104	0.0	0.0	0.0	0.0	0.0	0.0

CHAMBER ASSEMBLY (CA) PARAMETERS

	TIME (SEC)	MOTOR (DEG)	CA DISPL (DEG)	CA VELOCITY (DEG/SEC)	CA ACCELERATION (DEG/SEC**2)	CRUSH FORCE (FT-LBS)	RESEAR FORCE (LBS)	STUD FORCE (LBS)
1	0.0	0.0	0.0	0.0	0.0	584.05371	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	584.05371	0.0	0.0
3	0.01000	2.95483	0.0	0.0	0.0	584.05371	0.0	0.0
4	0.02000	11.81933	0.0	0.0	0.0	584.05371	0.0	0.0
5	0.03000	26.59344	0.0	0.0	0.0	584.05371	0.0	0.0
6	0.04000	47.27733	0.0	0.0	0.0	584.05371	0.0	0.0
7	0.05000	73.82103	0.0	0.0	0.0	584.05371	0.0	0.0
8	0.06000	106.15965	0.0	0.0	0.0	584.05371	0.0	0.0
9	0.07000	144.19530	0.0	0.0	0.0	584.05371	0.0	0.0
10	0.08000	187.77528	0.0	0.0	0.0	584.05371	0.0	0.0
11	0.09000	236.77397	0.0	0.0	0.0	584.05371	0.0	0.0
12	0.10000	291.69507	0.0	0.0	0.0	584.05371	0.0	0.0
13	0.11250	366.80444	0.0	0.0	0.0	584.05371	0.0	0.0
14	0.12500	450.43066	0.0	0.0	0.0	584.05371	0.0	0.0
15	0.13500	522.57178	0.0	0.0	0.0	584.05371	0.0	0.0
16	0.15000	639.27368	0.0	0.0	0.0	0.0	0.0	0.0
17	0.16000	722.36401	0.0	0.0	0.0	0.0	0.0	0.0
18	0.17000	809.78125	0.0	0.0	0.0	0.0	0.0	0.0
19	0.18000	901.08569	0.0	0.0	0.0	0.0	0.0	0.0
20	0.19000	996.39502	0.0	0.0	0.0	0.0	0.0	0.0
21	0.20000	1095.25439	0.00207	0.66242	141.38092	0.0	0.0	10.61064
22	0.21000	1197.64966	0.01813	2.79195	445.63379	0.0	0.0	33.44481
23	0.22000	1303.29712	0.05454	5.99738	-1617.14673	0.0	0.0	-121.35685
24	0.23000	1412.16650	0.15755	12.27180	678.14648	0.0	0.0	50.89490
25	0.24000	1524.01587	0.31442	19.41730	651.40234	0.0	0.0	48.88774
26	0.25000	1638.74756	0.54764	23.81201	-5225.99219	0.0	0.0	-270.48853
27	0.26000	1756.16921	0.87250	37.22726	238.68201	0.0	0.0	12.35378
28	0.27000	1876.16943	1.23887	46.20251	872.70752	0.0	0.0	45.16988
29	0.28000	1998.59519	1.79720	55.14091	791.56787	0.0	0.0	40.97023
30	0.29000	2123.30515	2.33782	62.70477	682.68823	0.0	0.0	35.33479
31	0.30000	2250.17261	3.04389	68.11208	407.71875	0.0	0.0	21.10283
32	0.31000	2379.06226	3.74151	70.94577	136.28494	0.0	0.0	7.05388
33	0.32000	2509.84375	4.45172	70.63701	-203.17181	0.0	0.0	-10.51583
34	0.33000	2642.41748	5.14295	67.07370	-512.60228	0.0	0.0	-26.50453
35	0.34000	2776.63525	5.73347	60.49393	-879.06419	0.0	0.0	-45.49786
36	0.35000	2912.42310	6.34565	51.02579	-2190.61304	0.0	0.0	-113.38246
37	0.36000	3049.53867	6.80742	40.67664	-1170.76025	0.0	34.45190	-60.59560
38	0.37000	3188.22461	7.15757	29.31572	-1199.77954	0.0	36.90302	-62.09859
39	0.38000	3328.06543	7.39499	18.38603	-1049.35474	0.0	38.56487	-54.31285
40	0.39000	3469.08911	7.53110	9.17488	-816.15332	0.0	39.51770	-42.24274
41	0.40000	3611.21289	7.53863	2.87581	-485.67871	0.0	39.92039	-25.13792
42	0.41000	3754.34592	7.59992	0.14817	-29.81223	0.0	39.99947	-1.54303
43	0.42000	3898.42261	7.60000	-0.02186	-57.54227	0.0	39.99997	-2.97829
44	0.43000	4043.30591	7.53948	-2.83506	-552.11841	0.0	39.92539	-28.57672
45	0.44000	4188.98437	7.53225	-6.04524	6740.25391	0.0	39.52573	348.86426
46	0.45000	4335.30469	7.33890	-19.57260	-1180.38647	0.0	38.52229	-61.09483
47	0.46000	4462.23828	7.13254	-32.00558	-1316.38135	0.0	36.72780	-68.13370
48	0.47000	4629.63281	6.74562	-45.39264	-1452.24243	0.0	34.01935	-75.16565
49	0.48000	4777.57812	6.22497	-58.56226	-1227.75610	0.0	0.0	-63.54660
50	0.49000	4925.84766	5.53315	-69.41640	-832.93970	0.0	0.0	-43.11157
51	0.50000	5074.73437	4.84415	-77.58710	-502.33521	0.0	0.0	-26.00003
52	0.50500	5149.21875	4.45029	-79.91760	-450.55200	0.0	0.0	-23.31981
53	0.51250	5261.32031	3.84269	-81.67354	-127.62103	0.0	0.0	-6.60545
54	0.52250	5411.26562	3.02987	-79.93475	377.59253	0.0	0.0	19.54355
55	0.53500	5599.14453	2.07830	-71.42545	922.59863	0.0	0.0	47.75217

56	0.54500	5749.87891	1.41743	-59.69763	1711.56250	0.0	0.0	88.58762
57	0.55500	5900.17578	0.87991	-47.02263	1092.83032	0.0	0.0	56.56306
58	0.56250	6013.33203	0.56511	-36.87286	1691.27612	0.0	0.0	87.53763
59	0.57250	6164.80078	0.26555	-23.42027	1316.54004	0.0	0.0	98.80630
60	0.58250	6316.66016	0.09154	-12.06980	1023.96460	0.0	0.0	76.84854
61	0.59250	6468.83984	0.01615	-3.79880	715.46973	206.18291	0.0	53.69600
62	0.60500	6659.48437	-0.00003	0.06800	58.08842	584.13745	0.0	4.35954
63	0.61250	6773.96484	0.0	0.0	0.0	584.05371	0.0	0.0
64	0.62250	6926.74922	0.0	0.0	0.0	584.05371	0.0	0.0
65	0.63500	7118.53125	0.0	0.0	0.0	584.05371	0.0	0.0
66	0.64500	7272.19531	0.0	0.0	0.0	584.05371	0.0	0.0
67	0.65500	7426.05859	0.0	0.0	0.0	584.05371	0.0	0.0
68	0.67000	7657.28906	0.0	0.0	0.0	0.0	0.0	0.0
69	0.67500	7734.26953	0.0	0.0	0.0	0.0	0.0	0.0
70	0.68500	7888.37500	0.0	0.0	0.0	0.0	0.0	0.0
71	0.69500	8042.83984	0.00562	1.94765	399.60132	0.0	0.0	29.99007
72	0.70500	8197.47266	0.05490	9.68755	915.59790	0.0	0.0	68.71562
73	0.72000	8429.94531	0.30720	26.23604	1291.51465	0.0	0.0	96.92816
74	0.73000	8585.06641	0.65677	59.12297	-15186.5742	0.0	0.0	-1821.19824
75	0.74000	8740.32812	1.11909	55.18683	1522.08301	0.0	0.0	78.78047
76	0.75000	8895.69922	1.73933	68.48915	985.98462	0.0	0.0	51.03291
77	0.76000	9051.19531	2.47666	75.41043	737.35327	0.0	0.0	38.16417
78	0.77000	9206.77734	3.29311	84.06819	255.00128	0.0	0.0	13.19844
79	0.78000	9362.47266	4.14092	84.66040	-174.31560	0.0	0.0	-9.02228
80	0.79000	9518.23047	4.96830	80.03989	-712.06079	0.0	0.0	-36.85507
81	0.80000	9674.05859	5.72595	70.93344	-1097.73853	0.0	0.0	-56.81711
82	0.81000	9829.92578	6.37277	57.13953	-1644.87842	0.0	0.0	-85.23967
83	0.82000	9985.84375	6.88212	43.51314	-1432.39331	0.0	34.97487	-74.13829
84	0.83000	10141.7812	7.24325	28.79276	-1479.21167	0.0	37.50272	-76.56152
85	0.84000	10297.7500	7.46377	15.76344	-987.54858	0.0	39.04642	-51.11386
86	0.85000	10453.7578	7.56922	5.98855	-752.31812	0.0	39.78456	-38.93872
87	0.86000	10609.7539	7.59869	0.71329	-324.67871	0.0	39.99080	-16.80482
88	0.87000	10765.8125	7.60000	-0.00000	-0.00000	0.0	39.99998	-0.00000
89	0.88000	10921.7773	7.59441	-1.95941	-408.73389	0.0	39.96089	-21.15538
90	0.89000	11077.8203	7.54445	-8.82653	-933.47534	0.0	39.61122	-48.31512
91	0.90000	11233.7461	7.40419	-19.80145	-1283.12573	0.0	38.62930	-66.41245
92	0.91000	11389.6211	7.13857	-33.62480	-1447.56738	0.0	36.77002	-74.92368
93	0.92000	11545.3555	6.72827	-48.77684	-453.42407	0.0	33.89792	-23.46846
94	0.93000	11701.0273	6.17205	-62.50518	-1174.25171	0.0	0.0	-60.77731
95	0.94000	11856.6211	5.48582	-74.36183	-1219.37378	0.0	0.0	-63.11275
96	0.95000	12012.2137	4.70113	-81.92516	-525.63501	0.0	0.0	-27.20599
97	0.96000	12167.8555	3.86192	-85.08286	-99.95529	0.0	0.0	-5.17352
98	0.96500	12246.4687	3.43325	-84.54340	231.40100	0.0	0.0	11.97693
99	0.97125	12343.7461	2.91163	-82.07689	533.75464	0.0	0.0	27.62624
100	0.98250	12518.7617	2.03369	-73.14853	1162.36377	0.0	0.0	60.16230
101	0.99250	12674.2852	1.36171	-60.57499	-824.75806	0.0	0.0	-42.68810
102	1.00250	12829.2256	0.82177	-46.54798	1731.57080	0.0	0.0	89.62321
103	1.01250	12984.8857	0.42736	-32.07072	1487.42212	0.0	0.0	111.63103
104	1.02249	13140.6602	0.17608	-18.49471	1335.18408	0.0	0.0	109.28059
105	1.03499	13335.4492	0.02989	-5.86221	567.95654	0.0	0.0	50.13013
106	1.04499	13491.3047	0.00119	-0.57490	271.55234	556.11963	0.0	20.38751
107	1.05499	13647.1562	0.0	0.0	0.0	584.05371	0.0	0.0
108	1.06249	13763.8477	0.0	0.0	0.0	584.05371	0.0	0.0
109	1.07248	13919.6641	0.0	0.0	0.0	584.05371	0.0	0.0
110	1.08498	14114.4258	0.0	0.0	0.0	584.05371	0.0	0.0
111	1.09498	14270.3242	0.0	0.0	0.0	584.05371	0.0	0.0
112	1.10998	14504.1675	0.0	0.0	0.0	0.0	0.0	0.0
113	1.11998	14659.9672	0.0	0.0	0.0	0.0	0.0	0.0
114	1.12998	14815.3594	0.0	0.0	0.0	0.0	0.0	0.0
115	1.13998	14971.0586	0.00815	2.44272	535.52769	0.0	0.0	40.19885

116	1.14998	15126.3359	0.06599	7.97100	-5011.14844	0.0	0.0	-376.08667
117	1.15998	15282.2930	0.21908	21.27145	1331.42041	0.0	0.0	99.92308
118	1.16998	15437.8828	0.49974	35.25810	1541.67090	0.0	0.0	115.70238
119	1.17997	15593.7461	0.92682	50.15285	1415.16895	0.0	0.0	73.24678
120	1.18997	15749.5703	1.49946	64.17888	1429.76685	0.0	0.0	74.00233
121	1.19997	15905.4648	2.20032	75.44072	824.69897	0.0	0.0	42.68504
122	1.20997	16061.3984	2.99545	82.78993	479.88135	0.0	0.0	24.83784
123	1.21997	16217.3555	3.83956	85.14056	-45.02480	0.0	0.0	-2.33041
124	1.22997	16373.3437	4.66140	82.35945	-505.93066	0.0	0.0	-26.18611
125	1.23997	16529.3398	5.46983	74.63824	-918.61645	0.0	0.0	-47.54605
126	1.24997	16685.3477	6.16031	62.63705	-2372.48437	0.0	0.0	-122.79582
127	1.25997	16841.3516	6.72037	48.75954	-1352.14966	0.0	33.84259	-69.98502
128	1.26997	16997.3828	7.13364	33.95499	-1516.48657	0.0	36.73550	-78.49080
129	1.27997	17153.4023	7.40145	20.00966	-1299.34863	0.0	38.61021	-67.25211
130	1.28997	17309.4453	7.54333	8.97471	-872.24585	0.0	39.60332	-45.14598
131	1.29997	17465.4922	7.59416	2.06235	-462.99463	0.0	39.95909	-23.96384
132	1.30997	17621.5430	7.60000	0.03535	33.45605	0.0	40.00000	1.73163
133	1.31996	17777.5977	7.59887	-0.69019	-276.03735	0.0	39.99207	-14.28724
134	1.32996	17933.5195	7.56999	-5.84546	-773.98096	0.0	39.78990	-40.05997
135	1.33996	18089.5859	7.46440	-17.95152	3875.60156	0.0	39.05078	200.59465
136	1.34996	18245.3203	7.24728	-28.56450	-1554.61694	0.0	37.53096	-80.46437
137	1.35996	18401.2070	6.89159	-46.76321	-21446.1328	0.0	35.04112	-1110.01611
138	1.36996	18556.7617	6.33267	-57.75146	-1350.76050	0.0	0.0	-69.91310
139	1.37996	18712.3516	5.73905	-70.50798	-1228.61719	0.0	0.0	-63.59117
140	1.38996	18867.8906	4.93464	-79.71808	-772.17896	0.0	0.0	-39.96669
141	1.39996	19023.3867	4.15962	-84.72903	-340.93677	0.0	0.0	-17.64632
142	1.40496	19101.3008	3.73481	-85.46280	-22.46910	0.0	0.0	-1.16296
143	1.41246	19218.5000	3.09975	-83.58397	393.99390	0.0	0.0	20.39246
144	1.42245	19374.8672	2.29361	-76.81078	1032.20874	0.0	0.0	53.42538
145	1.43495	19570.0273	1.41822	-61.56624	1861.61572	0.0	0.0	96.35411
146	1.44495	19725.3555	0.85476	-48.20396	1331.93481	0.0	0.0	68.93872
147	1.45245	19842.0781	0.54498	-36.94009	1661.34302	0.0	0.0	85.98833
148	1.46245	19997.9756	0.24627	-22.86908	1384.78149	0.0	0.0	103.92784
149	1.47244	20153.9023	0.07893	-11.00697	1129.18018	0.0	0.0	84.66992
150	1.48244	20309.8320	0.01159	-3.02922	519.34326	312.78076	0.0	38.90166
151	1.49244	20465.7461	-0.00003	-0.09578	160.22702	584.92349	0.0	12.02504
152	1.50244	20621.5078	0.0	0.0	0.0	584.05371	0.0	0.0
153	1.51118	20757.9609	0.0	0.0	0.0	584.05371	0.0	0.0
154	1.52118	20914.1133	0.0	0.0	0.0	584.05371	0.0	0.0
155	1.53118	21070.2578	0.0	0.0	0.0	584.05371	0.0	0.0
156	1.54118	21226.3828	0.0	0.0	0.0	584.05371	0.0	0.0
157	1.55618	21460.5156	0.0	0.0	0.0	0.0	0.0	0.0
158	1.56617	21616.5654	0.0	0.0	0.0	0.0	0.0	0.0
159	1.57617	21773.1328	0.00017	0.21013	150.49234	0.0	0.0	11.29445
160	1.58117	21851.1875	0.00423	1.64789	424.16968	0.0	0.0	31.83394
161	1.59117	22007.5352	0.04933	5.17620	897.55298	0.0	0.0	67.36133
162	1.60117	22163.8857	0.18224	18.97775	1172.97461	0.0	0.0	88.03174
163	1.61117	22320.1328	0.43895	32.71567	1449.46582	0.0	0.0	108.78239
164	1.62617	22554.5156	1.09737	54.90837	1412.71899	0.0	0.0	73.11998
165	1.63617	22710.6494	1.71566	68.42329	1194.58423	0.0	0.0	61.31210
166	1.64616	22866.7959	2.45284	78.50351	734.57300	0.0	0.0	38.02028
167	1.65616	23022.9102	3.27044	84.27498	218.51642	0.0	0.0	11.31004
168	1.66616	23179.0117	4.[illegible]2027	84.90811	-151.19398	0.0	0.0	-7.82555
169	1.67616	23335.1250	4.95060	80.35417	-706.43604	0.0	0.0	-36.56395
170	1.68616	23491.2227	5.71149	71.26131	-1030.81519	0.0	0.0	-53.35327
171	1.69616	23647.3154	6.36194	57.59180	-2338.92505	0.0	0.0	-121.05885
172	1.70616	23803.4052	6.87435	43.81329	-1451.88867	0.0	34.92053	-75.14734
173	1.71616	23959.4844	7.23836	29.06366	-1458.80696	0.0	37.46855	-75.50540
174	1.72616	24115.5547	7.45120	15.93818	-1042.10474	0.0	39.02843	-53.93759
175	1.73616	24271.6406	7.55827	6.11377	-773.31519	0.0	39.77792	-40.02550

176	1.74616	24427.7070	7.59857	0.76390	-333.86792	0.0	39.99001	-17.28044
177	1.75616	24583.8125	7.60000	-0.00000	-0.00000	0.0	39.99998	-0.00000
178	1.76616	24739.8242	7.59470	-1.89950	-414.00977	0.0	39.96288	-21.42845
179	1.77616	24895.9062	7.54577	-8.69436	-926.14819	0.0	39.62006	-47.93588
180	1.78615	25051.8789	7.40697	-19.62723	-1251.38452	0.0	38.64880	-64.76956
181	1.79615	25207.7773	7.14326	-33.42886	-1447.64355	0.0	36.80284	-74.92761
182	1.80615	25363.5508	6.73498	-48.68501	-975.16699	0.0	33.94482	-50.47301
183	1.81615	25519.2383	6.18055	-62.40639	-1022.16699	0.0	0.0	-52.90565
184	1.82615	25674.9258	5.49552	-74.24002	-1252.44019	0.0	0.0	-64.82420
185	1.83615	25830.5391	4.71176	-81.89594	-513.52148	0.0	0.0	-26.57901
186	1.84615	25986.3437	3.87200	-85.10817	-94.80286	0.0	0.0	-4.90684
187	1.85115	26064.9687	3.44322	-84.60139	207.34387	0.0	0.0	10.73177
188	1.86114	26220.6094	2.61735	-79.83606	723.77661	0.0	0.0	37.46147
189	1.87114	26376.2422	1.85215	-70.75964	1261.83472	0.0	0.0	65.31046
190	1.88114	26531.5703	1.21966	-56.74446	967.52100	0.0	0.0	50.07727
191	1.89114	26686.6836	0.71436	-43.36931	1303.10303	0.0	0.0	67.44643
192	1.90114	26842.5742	0.35463	-28.70421	1374.96289	0.0	0.0	103.19095
193	1.91113	26998.4766	0.13509	-15.76981	1027.97705	0.0	0.0	77.14967
194	1.92113	27154.3594	0.03043	-5.92597	678.12402	0.0	0.0	50.89320
195	1.93113	27310.2695	0.00125	-0.69922	276.54419	554.69702	0.0	20.75464
196	1.94113	27466.1719	0.0	0.0	0.0	584.05371	0.0	0.0
197	1.95113	27621.8359	0.0	0.0	0.0	584.05371	0.0	0.0
198	1.96112	27777.7344	0.0	0.0	0.0	584.05371	0.0	0.0
199	1.97112	27933.6953	0.0	0.0	0.0	584.05371	0.0	0.0
200	1.98112	28089.5977	0.0	0.0	0.0	584.05371	0.0	0.0
201	1.99112	28245.5000	0.0	0.0	0.0	584.05371	0.0	0.0
202	2.00112	28401.3437	0.0	0.0	0.0	0.0	0.0	0.0
203	2.01112	28556.8555	0.0	0.0	0.0	0.0	0.0	0.0
204	2.02111	28712.2891	0.00087	0.58040	251.48328	0.0	0.0	18.87381
205	2.03111	28867.8984	0.02756	5.52727	659.59814	0.0	0.0	49.50284
206	2.04111	29023.5703	0.12739	17.87590	3841.26562	0.0	0.0	288.28687
207	2.05111	29179.2812	0.33987	27.83414	1380.32886	0.0	0.0	103.59366
208	2.06111	29335.1094	0.68349	20.05383	25105.3594	0.0	0.0	1299.41138
209	2.07111	29490.9727	1.19197	57.34219	1414.87427	0.0	0.0	73.23154
210	2.08111	29646.9023	1.83140	70.14079	1122.63281	0.0	0.0	58.10559
211	2.09111	29802.8398	2.58393	79.69543	787.92041	0.0	0.0	40.78143
212	2.10111	29958.8281	3.40972	84.60385	253.52802	0.0	0.0	13.12219
213	2.11111	30114.8164	4.25935	84.37996	-307.39038	0.0	0.0	-15.91002
214	2.12111	30270.8242	5.08103	79.08188	-794.78369	0.0	0.0	-41.13667
215	2.13110	30426.8359	5.82593	69.16716	-1131.18018	0.0	0.0	-58.54799
216	2.14110	30582.8633	6.45529	56.56294	-1697.91333	0.0	0.0	-87.88115
217	2.15110	30738.8945	6.94354	41.41702	-1595.48437	0.0	35.40480	-82.57961
218	2.16110	30894.9375	7.28353	26.76347	-1359.42651	0.0	37.78471	-70.35165
219	2.17110	31050.9844	7.48555	14.05666	-1187.06763	0.0	39.19893	-61.44064
220	2.18110	31207.0117	7.57718	4.94490	-707.90698	0.0	39.84030	-36.64009
221	2.19110	31363.0703	7.59947	0.37323	-196.31305	0.0	39.99626	-10.16084
222	2.20110	31519.0977	7.60000	-0.00000	-0.00000	0.0	39.99998	-0.00000
223	2.21110	31675.1172	7.59112	-2.63740	-629.54810	0.0	39.93785	-32.58437
224	2.22110	31831.1250	7.53167	-5.67940	8884.03125	0.0	39.52171	459.82275
225	2.23110	31987.0586	7.37377	-21.75818	-1328.92676	0.0	38.41541	-68.78302
226	2.24110	32142.9687	7.09738	-36.03967	119.08269	0.0	36.41164	6.16352
227	2.25110	32298.6211	6.65607	-50.59018	-1418.20874	0.0	33.39247	-73.40411
228	2.26110	32454.4023	6.07579	-64.46193	-1323.69141	0.0	0.0	-68.77084
229	2.27109	32609.8281	5.37644	-75.52530	-759.10278	0.0	0.0	-39.28989
230	2.28109	32765.5195	4.59044	-82.79945	-509.67871	0.0	0.0	-26.38011
231	2.29109	32921.0312	3.73812	-85.12440	-35.51691	0.0	0.0	-1.89005
232	2.30109	33077.3828	2.89512	-81.92569	538.42896	0.0	0.0	27.86818
233	2.31109	33232.8906	2.11124	-74.18102	928.62695	0.0	0.0	48.06418
234	2.32358	33427.0625	1.27375	-59.51309	4924.57031	0.0	0.0	254.88753
235	2.33108	33543.1328	0.87273	-48.47719	1310.32935	0.0	0.0	67.82045

236	2.34108	33698.7656	0.46244	-33.74258	1531.00366	0.0	0.0	114.90181
237	2.35107	33854.7461	C.19631	-19.93600	1268.22290	0.0	0.0	95.18013
238	2.36107	34010.7656	0.05573	-8.86019	844.41870	0.0	0.0	63.37363
239	2.37107	34166.7500	0.00567	-1.98653	462.92310	451.47998	0.0	34.74237
240	2.38357	34361.7266	0.0	0.0	0.0	584.05371	0.0	0.0
241	2.39106	34478.4844	0.0	0.0	0.0	584.05371	0.0	0.0
242	2.40106	34634.5547	0.0	0.0	0.0	584.05371	0.0	0.0
243	2.41106	34790.6055	0.0	0.0	0.0	584.05371	0.0	0.0
244	2.42105	34946.6836	0.0	0.0	0.0	584.05371	0.0	0.0
245	2.43105	35102.7344	0.0	0.0	0.0	584.05371	0.0	0.0
246	2.44105	35258.7812	0.0	0.0	0.0	0.0	0.0	0.0
247	2.45105	35415.2852	0.0	0.0	0.0	0.0	0.0	0.0
248	2.46105	35571.8242	0.00001	0.03194	79.54527	0.0	0.0	5.96987
249	2.47105	35727.7422	0.01321	3.47876	354.61597	0.0	0.0	26.61394
250	2.48104	35883.7148	0.07198	13.25315	33784.3008	0.0	0.0	2535.51147
251	2.49104	36039.7461	0.25782	23.51338	1270.00049	0.0	0.0	95.31354
252	2.50104	36195.7891	0.55732	28.08549	2321.32178	0.0	0.0	120.14774
253	2.50104	0.0	0.0	0.0	0.0	0.0	0.0	0.0

APPENDIX A-5

DRAWING DATA SMOOTHING

The numerical integration routine (IBM-SSP routine HPCG) can quite possibly iterate excessively about a first derivative discontinuity such as exists in the drawing data for the feed, eject, and lock ring cam data. To avoid this excessive iteration the data has been treated to "smooth" the sharp corners. The "smoothing" algorithm used is a cubic fit that fits exactly at the endpoints in both displacement and slope.

Development:

CUBIC FIT

$$Y_1 = a_3X_1^3 + a_2X_1^2 + a_1X_1 + a_0$$

$$Y_2 = a_3Y_2^3 + a_2X_2^2 + a_1X_2 + a_0$$

$$M_1 = 3a_3X_1^2 + 2a_2X_1 + a_1$$

$$M_2 = 3a_3X_2^2 + 2a_2X_2 + a_1$$

Unknowns are a_0, a_1, a_2, a_3

$$a_3(X_1^3) + a_2(X_1^2) + a_1(X_1) + a_0(1) = Y_1$$

$$a_3(X_2^3) + a_2(X_2^2) + a_1(X_2) + a_0(1) = Y_2$$

$$a_3(3X_1^2) + a_2(2X_1) + a_1(1) + a_0(0) = M_1$$

$$a_3(3X_2^2) + a_2(2X_2) + a_1(1) + a_0(0) = M_2$$

$$\begin{matrix} X_1^3 & X_1^2 & X_1 & 1 \\ X_2^3 & X_2^2 & X_2 & 1 \\ 3X_1^2 & 2X_1 & 1 & 0 \\ 3X_2^2 & 2X_2 & 1 & 0 \end{matrix} \quad \begin{matrix} a_3 \\ a_2 \\ a_1 \\ a_0 \end{matrix} = \begin{matrix} Y_1 \\ Y_2 \\ M_1 \\ M_2 \end{matrix}$$

which is of form

[A]{a} = {RHS} where

only {a} is unknown. IBM-SSP

sinΘ can be used to determine {a}

when {a} is found, the Y values in the region of interest are then

Y(J) = a(4) + X(J)(a(3) + X(J)(a(2) + X(J)(a(1))))

For Range of X(J) between and including endpoints X1 and X2.

PROGRAM: The original drawing data and the regions of each curve to be fit must be supplied to the fitting program. The graphical output only is included as the program output.

```
0001              REAL X(1600),Y(1000),Z(1000)
0002              DATA J/0/,N/0/
            C
            C-  INPUT DATA
            C
0003        10    CONTINUE
0004              J=J+1
0005              READ(5,100,ERR=101) X(J),Y(J)
0006               IF(X(J) .GT. 900.) GO TO 101
0007              N=N+1
0008              Z(J)=Y(J)
0009              GO TO 10
            C
0010        101   CONTINUE
            C
            C
0011              READ(5,200,END=201) ISTART,IEND
0012               WRITE(6,4001) ISTART,IEND
0013        4001      FORMAT(' ',2I10)
0014              CALL FIT3(X,Z,ISTART,IEND)
0015              GO TO 101
0016        201   CONTINUE
            C
0017              DO 700 J=1,N
0018               DIFF=Y(J)-Z(J)
0019              WRITE(6,4005)X(J),Y(J),Z(J),DIFF,I
0020        4005   FORMAT(' ',4F16.4,I10        )
0021        700   CONTINUE
            C
            C     CALL PUNCHG(X,Z,N)
0022              CALL P   ING(X,Y,Z,N)
            C
0023               STOP
0024        100   FORMAT(2F16.4)
0025        200   FORMAT(2I10)
0026                 END
```

```
0001                 SUBROUTINE PUNCHG(X,Z,N)
0002                 REAL X(1),Z(1)
             C
0003                 DO 300 J=1,N
0004                    PUNCH260,X(J),Z(J)
0005         260        FORMAT(2F16.4)
0006         300     CONTINUE
             C
             C
0007                      RETURN
0008                      END
```

```
      SUBROUTINE PLOTNG(X,Y,Z,N)
      REAL X(1),Y(1),Z1(1),Z2(1),Z (1)
       DIMENSION IBUF(1000)
      CALL PLOTS(IBUF,1000,14)
      CALL FACTOR(1.0)
      CALL NEWPEN(1)
      CALL PLOT(10.,-36.,-3)
      CALL PLOT(0.0,2.5,-3)
      CALL SCALE(X,10.,N,1)
      CALL SCALE(Y,8.,N,1)
      X1=X(N+1)
      X2=X(N+2)
      Y1=Y(N+1)
      Y2=Y(N+2)
       Z(N+1)=Y1
       Z(N+2)=Y2
      CALL AXIS(0.,0.,8H        ,-8,10.,0.,X1,X2)
      CALL AXIS(0.,0.,8H        ,+8,8.,90.,Y1,Y2)
      CALL LINE(X,Y ,N,1, 0,0)
      CALL LINE(X,Z ,N,1, 0,2)
      CALL PLOT(20.,0.,999)
      RETURN
       END
```

```
      SUBROUTINE FIT3(X,Y,ISTART,IEND)
C
C- ABSTRACT:  SUBROUTINE FIT3 GIVES A SMOOTH CUBIC FIT WITH EXACT FIT IN THE
C-    0 AND 1 DERIVATIVE FOR THE ENDPOINTS OF THE INDICATED INTERVAL...WHICH
C-    IS DETERMINED BY THE RANGE FROM X(ISTART+1) THRU X(IEND-1)
C
C
      REAL X(1),Y(1),M1,M2
      DOUBLE PRECISION DX(30),DY(30),DRHS(4),DA(4,4)
C
C- DETERMINE THE SLOPES FROM END DATA AND CHECK NUMBER OF POINTS
C
      IS=ISTART+1
      IE=IEND-1
      DELTA=X(IS)-X(ISTART)
      INUM=IFIX((X(IE)-X(IS))/DELTA) + 1
      IF(INUM.LT.6 .AND. INUM.GT.27) RETURN
C
      DO 100 J=1,INUM
      DX(J)=FLOAT(J)*DELTA
         DY(J)=Y(IS+J-1)
100   CONTINUE
      M1=(Y(IS)-Y(ISTART))/DELTA
      M2=(Y(IEND)-Y(IE))/DELTA
C
      DA(1,1)=DX(1)**3
      DA(2,1)=DX(INUM)**3
      DA(3,1)=3.D0*DX(1)*DX(1)
      DA(4,1)=3.D0*DX(INUM)*DX(INUM)
C
      DA(1,2)=DX(1)**2
      DA(2,2)=DX(INUM)**2
      DA(3,2)=2.D0*DX(1)
      DA(4,2)=2.D0*DX(INUM)
C
      DA(1,3)=DX(1)
      DA(2,3)=DX(INUM)
      DA(3,3)=1.D0
      DA(4,3)=1.D0
C
      DA(1,4)=1.D0
      DA(2,4)=1.D0
      DA(3,4)=0.D0
      DA(4,4)=0.D0
C
      DRHS(1)=DY(1)
      DRHS(2)=DY(INUM)
      DRHS(3)=M1
      DRHS(4)=M2
C
      CALL DSIMQ(DA,DRHS,4,KS)
C
      IF(KS.EQ.0) GO TO 200
```

```
0037               WRITE(6,1000)
0038        1000   FORMAT(' PROBLEMS OUT OF SIMQ'
0039        200    CONTINUE
            C
0040               DO 300 J=1,INUM
0041                  Y(IS+J-1)=DRHS(4)+
                 1       DX(J)*(DRHS(3)+DX(J)*(DRHS(2)+DX(J)*(DRHS(1))))
0042        300    CONTINUE
            C
0043                  RETURN
0044                  END
```

```
          C                                                                        SIMQ  10
          C     ...............................................................SIMQ  20
          C                                                                        SIMQ  30
          C        SUBROUTINE DSIMQ
          C                                                                        SIMQ  50
          C        PURPOSE                                                         SIMQ  60
          C           OBTAIN SOLUTION OF A SET OF SIMULTANEOUS LINEAR EQUATIONS,  SIMQ  70
          C           AX=B                                                         SIMQ  80
          C                                                                        SIMQ  90
          C        USAGE                                                           SIMQ 100
          C           CALL SIMQ(A,B,N,KS)                                          SIMQ 110
          C                                                                        SIMQ 120
          C        DESCRIPTION OF PARAMETERS                                       SIMQ 130
          C           A - MATRIX OF COEFFICIENTS STORED COLUMNWISE.  THESE ARE     SIMQ 140
          C               DESTROYED IN THE COMPUTATION.  THE SIZE OF MATRIX A IS  SIMQ 150
          C               N BY N.                                                  SIMQ 160
          C           B - VECTOR OF ORIGINAL CONSTANTS (LENGTH N). THESE ARE       SIMQ 170
          C               REPLACED BY FINAL SOLUTION VALUES, VECTOR X.             SIMQ 180
          C           N - NUMBER OF EQUATIONS AND VARIABLES. N MUST BE .GT. ONE.   SIMQ 190
          C           KS - OUTPUT DIGIT                                            SIMQ 200
          C                0 FOR A NORMAL SOLUTION                                 SIMQ 210
          C                1 FOR A SINGULAR SET OF EQUATIONS                       SIMQ 220
          C                                                                        SIMQ 230
          C        REMARKS                                                         SIMQ 240
          C           MATRIX A MUST BE GENERAL.                                    SIMQ 250
          C           IF MATRIX IS SINGULAR , SOLUTION VALUES ARE MEANINGLESS.     SIMQ 260
          C           AN ALTERNATIVE SOLUTION MAY BE OBTAINED BY USING MATRIX      SIMQ 270
          C           INVERSION (MINV) AND MATRIX PRODUCT (GMPRD).                 SIMQ 280
          C                                                                        SIMQ 290
          C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED                   SIMQ 300
          C           NONE                                                         SIMQ 310
          C                                                                        SIMQ 320
          C        METHOD                                                          SIMQ 330
          C           METHOD OF SOLUTION IS BY ELIMINATION USING LARGEST PIVOTAL   SIMQ 340
          C           DIVISOR. EACH STAGE OF ELIMINATION CONSISTS OF INTERCHANGINGSIMQ 350
          C           ROWS WHEN NECESSARY TO AVOID DIVISION BY ZERO OR SMALL       SIMQ 360
          C           ELEMENTS.                                                    SIMQ 370
          C           THE FORWARD SOLUTION TO OBTAIN VARIABLE N IS DONE IN         SIMQ 380
          C           N STAGES. THE BACK SOLUTION FOR THE OTHER VARIABLES IS       SIMQ 390
          C           CALCULATED BY SUCCESSIVE SUBSTITUTIONS. FINAL SOLUTION       SIMQ 400
          C           VALUES ARE DEVELOPED IN VECTOR B, WITH VARIABLE 1 IN B(1),   SIMQ 410
          C           VARIABLE 2 IN B(2),........, VARIABLE N IN B(N).             SIMQ 420
          C           IF NO PIVOT CAN BE FOUND EXCEEDING A TOLERANCE OF 0.0,       SIMQ 430
          C           THE MATRIX IS CONSIDERED SINGULAR AND KS IS SET TO 1. THIS   SIMQ 440
          C           TOLERANCE CAN BE MODIFIED BY REPLACING THE FIRST STATEMENT.  SIMQ 450
          C                                                                        SIMQ 460
          C     ...............................................................SIMQ 470
          C                                                                        SIMQ 480
0001            SUBROUTINE DSIMQ(A,B,N,KS)
0002            DIMENSION A(1),B(1)                                                SIMQ 500
0003            DOUBLE PRECISION A,B,BIGA,TOL,SAVE,DABS
          C                                                                        SIMQ 510
          C        FORWARD SOLUTION                                                SIMQ 520
```

```
          C                                                                 SIMQ 530
 0004           TOL=0.D0
 0005           KS=0                                                        SIMQ 550
 0006           JJ=-N                                                       SIMQ 560
 0007           DO 65 J=1,N                                                 SIMQ 570
 0008           JY=J+1                                                      SIMQ 580
 0009           JJ=JJ+N+1                                                   SIMQ 590
 0010           BIGA=0.D0
 0011           IT=JJ-J                                                     SIMQ 610
 0012           DO 30 I=J,N                                                 SIMQ 620
          C                                                                 SIMQ 630
          C        SEARCH FOR MAXIMUM COEFFICIENT IN COLUMN                 SIMQ 640
          C                                                                 SIMQ 650
 0013           IJ=IT+I                                                     SIMQ 660
 0014           IF(DABS(BIGA)-DABS(A(IJ))) 20,30,30
 0015        20 BIGA=A(IJ)                                                  SIMQ 680
 0016           IMAX=I                                                      SIMQ 690
 0017        30 CONTINUE                                                    SIMQ 700
          C                                                                 SIMQ 710
          C        TEST FOR PIVOT LESS THAN TOLERANCE (SINGULAR MATRIX)     SIMQ 720
          C                                                                 SIMQ 730
 0018           IF(DABS(BIGA)-TOL) 35,35,40
 0019        35 KS=1                                                        SIMQ 750
 0020           RETURN                                                      SIMQ 760
          C                                                                 SIMQ 770
          C        INTERCHANGE ROWS IF NECESSARY                            SIMQ 780
          C                                                                 SIMQ 790
 0021        40 I1=J+N*(J-2)                                                SIMQ 800
 0022           IT=IMAX-J                                                   SIMQ 810
 0023           DO 50 K=J,N                                                 SIMQ 820
 0024           I1=I1+N                                                     SIMQ 830
 0025           I2=I1+IT                                                    SIMQ 840
 0026           SAVE=A(I1)                                                  SIMQ 850
 0027           A(I1)=A(I2)                                                 SIMQ 860
 0028           A(I2)=SAVE                                                  SIMQ 870
          C                                                                 SIMQ 880
          C        DIVIDE EQUATION BY LEADING COEFFICIENT                   SIMQ 890
          C                                                                 SIMQ 900
 0029        50 A(I1)=A(I1)/BIGA                                            SIMQ 910
 0030           SAVE=B(IMAX)                                                SIMQ 920
 0031           B(IMAX)=B(J)                                                SIMQ 930
 0032           B(J)=SAVE/BIGA                                              SIMQ 940
          C                                                                 SIMQ 950
          C        ELIMINATE NEXT VARIABLE                                  SIMQ 960
          C                                                                 SIMQ 970
 0033           IF(J-N) 55,70,55                                            SIMQ 980
 0034        55 IQS=N*(J-1)                                                 SIMQ 990
 0035           DO 65 IX=JY,N                                               SIMQ1000
 0036           IXJ=IQS+IX                                                  SIMQ1010
 0037           IT=J-IX                                                     SIMQ1020
 0038           DO 60 JX=JY,N                                               SIMQ1030
 0039           IXJX=N*(JX-1)+IX                                            SIMQ1040
 0040           JJX=IXJX+IT                                                 SIMQ1050
```

```
0041          60 A(IXJX)=A(IXJX)-(A(IXJ)*A(JJX))                       SIMQ1060
0042          65 B(IX)=B(IX)-(B(J)*A(IXJ))                             SIMQ1070
           C                                                           SIMQ1080
           C        BACK SOLUTION                                      SIMQ1090
           C                                                           SIMQ1100
0043          70 NY=N-1                                                SIMQ1110
0044             IT=N*N                                                SIMQ1120
0045             DO 80 J=1,NY                                          SIMQ1130
0046             IA=IT-J                                               SIMQ1140
0047             IB=N-J                                                SIMQ1150
0048             IC=N                                                  SIMQ1160
0049             DO 80 K=1,J                                           SIMQ1170
0050             B(IB)=B(IB)-A(IA)*B(IC)                               SIMQ1180
0051             IA=IA-N                                               SIMQ1190
0052          80 IC=IC-1                                               SIMQ1200
0053             RETURN                                                SIMQ1210
0054             END                                                   SIMQ1220
```

28	37
48	58
343	353
358	366

1.0000	33.0000	33.0000	0.0	1
2.0000	33.0000	33.0000	0.0	2
3.0000	33.0000	33.0000	0.0	3
4.0000	33.0000	33.0000	0.0	4
5.0000	33.0000	33.0000	0.0	5
6.0000	33.0000	33.0000	0.0	6
7.0000	33.0000	33.0000	0.0	7
8.0000	33.0000	33.0000	0.0	8
9.0000	33.0000	33.0000	0.0	9
10.0000	33.0000	33.0000	0.0	10
11.0000	33.0000	33.0000	0.0	11
12.0000	33.0000	33.0000	0.0	12
13.0000	33.0000	33.0000	0.0	13
14.0000	33.0000	33.0000	0.0	14
15.0000	33.0000	33.0000	0.0	15
16.0000	33.0000	33.0000	0.0	16
17.0000	33.0000	33.0000	0.0	17
18.0000	33.0000	33.0000	0.0	18
19.0000	33.0000	33.0000	0.0	19
20.0000	33.0000	33.0000	0.0	20
21.0000	33.0000	33.0000	0.0	21
22.0000	33.0000	33.0000	0.0	22
23.0000	33.0000	33.0000	0.0	23
24.0000	33.0000	33.0000	0.0	24
25.0000	33.0000	33.0000	0.0	25
26.0000	33.0000	33.0000	0.0	26
27.0000	33.0000	33.0000	0.0	27
28.0000	33.0000	33.0000	0.0	28
29.0000	33.0000	33.0000	0.0	29
30.0000	33.0000	32.8523	0.1477	30
31.0000	33.0000	32.3987	0.6013	31
32.0000	33.0000	31.6238	1.3762	32
33.0000	33.0000	30.5120	2.4880	33
34.0000	31.0000	29.0478	1.9522	34
35.0000	27.7714	27.2157	0.5557	35
36.0000	25.0000	25.0000	0.0	36
37.0000	22.5874	22.5874	0.0	37
38.0000	21.0000	21.0000	0.0	38
39.0000	19.0698	19.0698	0.0	39
40.0000	16.4636	16.4636	0.0	40
41.0000	13.0000	13.0000	0.0	41
42.0000	10.6801	10.6801	0.0	42
43.0000	9.4784	9.4784	0.0	43
44.0000	8.0000	8.0000	0.0	44
45.0000	6.1829	6.1829	0.0	45
46.0000	4.2031	4.2031	0.0	46
47.0000	3.0000	3.0000	0.0	47
48.0000	2.4868	2.4868	0.0	48
49.0000	1.7910	1.7910	0.0	49
50.0000	1.0000	1.1813	-0.1813	50
51.0000	0.2524	0.7284	-0.4760	51
52.0000	0.0	0.4089	-0.4089	52
53.0000	0.0	0.1997	-0.1997	53
54.0000	0.0	0.0774	-0.0774	54
55.0000	0.0	0.0189	-0.0189	55
56.0000	0.0	0.0009	-0.0009	56

57.0000	0.0	-0.0000	0.0000	57
58.0000	0.0	0.0	0.0	58
59.0000	0.0	0.0	0.0	59
60.0000	0.0	0.0	0.0	60
61.0000	0.0	0.0	0.0	61
62.0000	0.0	0.0	0.0	62
63.0000	0.0	0.0	0.0	63
64.0000	0.0	0.0	0.0	64
65.0000	0.0	0.0	0.0	65
66.0000	0.0	0.0	0.0	66
67.0000	0.0	0.0	0.0	67
68.0000	0.0	0.0	0.0	68
69.0000	0.0	0.0	0.0	69
70.0000	0.0	0.0	0.0	70
71.0000	0.0	0.0	0.0	71
72.0000	0.0	0.0	0.0	72
73.0000	0.0	0.0	0.0	73
74.0000	0.0	0.0	0.0	74
75.0000	0.0	0.0	0.0	75
76.0000	0.0	0.0	0.0	76
77.0000	0.0	0.0	0.0	77
78.0000	0.0	0.0	0.0	78
79.0000	0.0	0.0	0.0	79
80.0000	0.0	0.0	0.0	80
81.0000	0.0	0.0	0.0	81
82.0000	0.0	0.0	0.0	82
83.0000	0.0	0.0	0.0	83
84.0000	0.0	0.0	0.0	84
85.0000	0.0	0.0	0.0	85
86.0000	0.0	0.0	0.0	86
87.0000	0.0	0.0	0.0	87
88.0000	0.0	0.0	0.0	88
89.0000	0.0	0.0	0.0	89
90.0000	0.0	0.0	0.0	90
91.0000	0.0	0.0	0.0	91
92.0000	0.0	0.0	0.0	92
93.0000	0.0	0.0	0.0	93
94.0000	0.0	0.0	0.0	94
95.0000	0.0	0.0	0.0	95
96.0000	0.0	0.0	0.0	96
97.0000	0.0	0.0	0.0	97
98.0000	0.0	0.0	0.0	98
99.0000	0.0	0.0	0.0	99
100.0000	0.0	0.0	0.0	100
101.0000	0.0	0.0	0.0	101
102.0000	0.0	0.0	0.0	102
103.0000	0.0	0.0	0.0	103
104.0000	0.0	0.0	0.0	104
105.0000	0.0	0.0	0.0	105
106.0000	0.0	0.0	0.0	106
107.0000	0.0	0.0	0.0	107
108.0000	0.0	0.0	0.0	108
109.0000	0.0	0.0	0.0	109
110.0000	0.0	0.0	0.0	110
111.0000	0.0	0.0	0.0	111
112.0000	0.0	0.0	0.0	112
113.0000	0.0	0.0	0.0	113
114.0000	0.0	0.0	0.0	114
115.0000	0.0	0.0	0.0	115
116.0000	0.0	0.0	0.0	116
117.0000	0.0	0.0	0.0	117

118.0000	0.0	0.0	0.0	118
119.0000	0.0	0.0	0.0	119
120.0000	0.0	0.0	0.0	120
121.0000	0.0	0.0	0.0	121
122.0000	0.0	0.0	0.0	122
123.0000	0.0	0.0	0.0	123
124.0000	0.0	0.0	0.0	124
125.0000	0.0	0.0	0.0	125
126.0000	0.0	0.0	0.0	126
127.0000	0.0	0.0	0.0	127
128.0000	0.0	0.0	0.0	128
129.0000	0.0	0.0	0.0	129
130.0000	0.0	0.0	0.0	130
131.0000	0.0	0.0	0.0	131
132.0000	0.0	0.0	0.0	132
133.0000	0.0	0.0	0.0	133
134.0000	0.0	0.0	0.0	134
135.0000	0.0	0.0	0.0	135
136.0000	0.0	0.0	0.0	136
137.0000	0.0	0.0	0.0	137
138.0000	0.0	0.0	0.0	138
139.0000	0.0	0.0	0.0	139
140.0000	0.0	0.0	0.0	140
141.0000	0.0	0.0	0.0	141
142.0000	0.0	0.0	0.0	142
143.0000	0.0	0.0	0.0	143
144.0000	0.0	0.0	0.0	144
145.0000	0.0	0.0	0.0	145
146.0000	0.0	0.0	0.0	146
147.0000	0.0	0.0	0.0	147
148.0000	0.0	0.0	0.0	148
149.0000	0.0	0.0	0.0	149
150.0000	0.0	0.0	0.0	150
151.0000	0.0	0.0	0.0	151
152.0000	0.0	0.0	0.0	152
153.0000	0.0	0.0	0.0	153
154.0000	0.0	0.0	0.0	154
155.0000	0.0	0.0	0.0	155
156.0000	0.0	0.0	0.0	156
157.0000	0.0	0.0	0.0	157
158.0000	0.0	0.0	0.0	158
159.0000	0.0	0.0	0.0	159
160.0000	0.0	0.0	0.0	160
161.0000	0.0	0.0	0.0	161
162.0000	0.0	0.0	0.0	162
163.0000	0.0	0.0	0.0	163
164.0000	0.0	0.0	0.0	164
165.0000	0.0	0.0	0.0	165
166.0000	0.0	0.0	0.0	166
167.0000	0.0	0.0	0.0	167
168.0000	0.0	0.0	0.0	168
169.0000	0.0	0.0	0.0	169
170.0000	0.0	0.0	0.0	170
171.0000	0.0	0.0	0.0	171
172.0000	0.0	0.0	0.0	172
173.0000	0.0	0.0	0.0	173
174.0000	0.0	0.0	0.0	174
.75.0000	0.0	0.0	0.0	175
176.0000	0.0	0.0	0.0	176
177.0000	0.0	0.0	0.0	177
178.0000	0.0	0.0	0.0	178

179.0000	0.0	0.0	0.0	179
180.0000	0.0	0.0	0.0	180
181.0000	0.0	0.0	0.0	181
182.0000	0.0	0.0	0.0	182
183.0000	0.0	0.0	0.0	183
184.0000	0.0	0.0	0.0	184
185.0000	0.0	0.0	0.0	185
186.0000	0.0	0.0	0.0	186
187.0000	0.0	0.0	0.0	187
188.0000	0.0	0.0	0.0	188
189.0000	0.0	0.0	0.0	189
190.0000	0.0	0.0	0.0	190
191.0000	0.0	0.0	0.0	191
192.0000	0.0	0.0	0.0	192
193.0000	0.0	0.0	0.0	193
194.0000	0.0	0.0	0.0	194
195.0000	0.0	0.0	0.0	195
196.0000	0.0	0.0	0.0	196
197.0000	0.0	0.0	0.0	197
198.0000	0.0	0.0	0.0	198
199.0000	0.0	0.0	0.0	199
200.0000	0.0	0.0	0.0	200
201.0000	0.0	0.0	0.0	201
202.0000	0.0	0.0	0.0	202
203.0000	0.0	0.0	0.0	203
204.0000	0.0	0.0	0.0	204
205.0000	0.0	0.0	0.0	205
206.0000	0.0	0.0	0.0	206
207.0000	0.0	0.0	0.0	207
208.0000	0.0	0.0	0.0	208
209.0000	0.0	0.0	0.0	209
210.0000	0.0	0.0	0.0	210
211.0000	0.0	0.0	0.0	211
212.0000	0.0	0.0	0.0	212
213.0000	0.0	0.0	0.0	213
214.0000	0.0	0.0	0.0	214
215.0000	0.0	0.0	0.0	215
216.0000	0.0	0.0	0.0	216
217.0000	0.0	0.0	0.0	217
218.0000	0.0	0.0	0.0	218
219.0000	0.0	0.0	0.0	219
220.0000	0.0	0.0	0.0	220
221.0000	0.0	0.0	0.0	221
222.0000	0.0	0.0	0.0	222
223.0000	0.0	0.0	0.0	223
224.0000	0.0	0.0	0.0	224
225.0000	0.0	0.0	0.0	225
226.0000	0.0	0.0	0.0	226
227.0000	0.0	0.0	0.0	227
228.0000	0.0	0.0	0.0	228
229.0000	0.0	0.0	0.0	229
230.0000	0.0	0.0	0.0	230
231.0000	0.0	0.0	0.0	231
232.0000	0.0	0.0	0.0	232
233.0000	0.0	0.0	0.0	233
234.0000	0.0	0.0	0.0	234
235.0000	0.0	0.0	0.0	235
236.0000	0.0	0.0	0.0	236
237.0000	0.0	0.0	0.0	237
238.0000	0.0	0.0	0.0	238
239.0000	0.0	0.0	0.0	239

240.0000	0.0	0.0	0.0	240
241.0000	0.0	0.0	0.0	241
242.0000	0.0	0.0	0.0	242
243.0000	0.0	0.0	0.0	243
244.0000	0.0	0.0	0.0	244
245.0000	0.0	0.0	0.0	245
246.0000	0.0	0.0	0.0	246
247.0000	0.0	0.0	0.0	247
248.0000	0.0	0.0	0.0	248
249.0000	0.0	0.0	0.0	249
250.0000	0.0	0.0	0.0	250
251.0000	0.0	0.0	0.0	251
252.0000	0.0	0.0	0.0	252
253.0000	0.0	0.0	0.0	253
254.0000	0.0	0.0	0.0	254
255.0000	0.0	0.0	0.0	255
256.0000	0.0	0.0	0.0	256
257.0000	0.0	0.0	0.0	257
258.0000	0.0	0.0	0.0	258
259.0000	0.0	0.0	0.0	259
260.0000	0.0	0.0	0.0	260
261.0000	0.0	0.0	0.0	261
262.0000	0.0	0.0	0.0	262
263.0000	0.0	0.0	0.0	263
264.0000	0.0	0.0	0.0	264
265.0000	0.0	0.0	0.0	265
266.0000	0.0	0.0	0.0	266
267.0000	0.0	0.0	0.0	267
268.0000	0.0	0.0	0.0	268
269.0000	0.0	0.0	0.0	269
270.0000	0.0	0.0	0.0	270
271.0000	0.0	0.0	0.0	271
272.0000	0.0	0.0	0.0	272
273.0000	0.0	0.0	0.0	273
274.0000	0.0	0.0	0.0	274
275.0000	0.0	0.0	0.0	275
276.0000	0.0	0.0	0.0	276
277.0000	0.0	0.0	0.0	277
278.0000	0.0	0.0	0.0	278
279.0000	0.0	0.0	0.0	279
280.0000	0.0	0.0	0.0	280
281.0000	0.0	0.0	0.0	281
282.0000	0.0	0.0	0.0	282
283.0000	0.0	0.0	0.0	283
284.0000	0.0	0.0	0.0	284
285.0000	0.0	0.0	0.0	285
286.0000	0.0	0.0	0.0	286
287.0000	0.0	0.0	0.0	287
288.0000	0.0	0.0	0.0	288
289.0000	0.0	0.0	0.0	289
290.0000	0.0	0.0	0.0	290
291.0000	0.0	0.0	0.0	291
292.0000	0.0	0.0	0.0	292
293.0000	0.0	0.0	0.0	293
294.0000	0.0	0.0	0.0	294
295.0000	0.0	0.0	0.0	295
296.0000	0.0	0.0	0.0	296
297.0000	0.0	0.0	0.0	297
298.0000	0.0	0.0	0.0	298
299.0000	0.0	0.0	0.0	299
300.0000	0.0	0.0	0.0	300

301.0000	0.0	0.0	0.0	301
302.0000	0.0	0.0	0.0	302
303.0000	0.0	0.0	0.0	303
304.0000	0.0	0.0	0.0	304
305.0000	0.0	0.0	0.0	305
306.0000	0.0	0.0	0.0	306
307.0000	0.0	0.0	0.0	307
308.0000	0.0	0.0	0.0	308
309.0000	0.0	0.0	0.0	309
310.0000	0.0	0.0	0.0	310
311.0000	0.0	0.0	0.0	311
312.0000	0.0	0.0	0.0	312
313.0000	0.0	0.0	0.0	313
314.0000	0.0	0.0	0.0	314
315.0000	0.0	0.0	0.0	315
316.0000	0.0	0.0	0.0	316
317.0000	0.0	0.0	0.0	317
318.0000	0.0	0.0	0.0	318
319.0000	0.0	0.0	0.0	319
320.0000	0.0	0.0	0.0	320
321.0000	0.0	0.0	0.0	321
322.0000	0.0	0.0	0.0	322
323.0000	0.0	0.0	0.0	323
324.0000	0.0	0.0	0.0	324
325.0000	0.0	0.0	0.0	325
326.0000	0.0	0.0	0.0	326
327.0000	0.0	0.0	0.0	327
328.0000	0.0	0.0	0.0	328
329.0000	0.0	0.0	0.0	329
330.0000	0.0	0.0	0.0	330
331.0000	0.0	0.0	0.0	331
332.0000	0.0	0.0	0.0	332
333.0000	0.0	0.0	0.0	333
334.0000	0.0	0.0	0.0	334
335.0000	0.0	0.0	0.0	335
336.0000	0.0	0.0	0.0	336
337.0000	0.0	0.0	0.0	337
338.0000	0.0	0.0	0.0	338
339.0000	0.0	0.0	0.0	339
340.0000	0.0	0.0	0.0	340
341.0000	0.0	0.0	0.0	341
342.0000	0.0	0.0	0.0	342
343.0000	0.0	0.0	0.0	343
344.0000	0.0	0.0000	-0.0000	344
345.0000	0.0	0.0977	-0.0977	345
346.0000	0.0	0.4687	-0.4687	346
347.0000	0.0	1.2305	-1.2305	347
348.0000	1.9000	2.5000	-0.6000	348
349.0000	5.5000	4.3945	1.1055	349
350.0000	8.0000	7.0312	0.9688	350
351.0000	10.5000	10.5273	-0.0273	351
352.0000	15.0000	15.0000	0.0000	352
353.0000	20.0000	20.0000	0.0	353
354.0000	25.0000	25.0000	0.0	354
355.0000	28.0000	28.0000	0.0	355
356.0000	29.2428	29.2428	0.0	356
357.0000	30.0000	30.0000	0.0	357
358.0000	30.3029	30.3029	0.0	358
359.0000	31.2707	31.2707	0.0000	359
360.0000	32.0000	32.0709	-0.0709	360
361.0000	33.0000	32.5793	0.4207	361

362.0000	33.0000	32.8612	0.1388	362
363.0000	33.0000	32.9818	0.0182	363
364.0000	33.0000	33.0063	-0.0063	364
365.0000	33.0000	33.0000	0.0	365
366.0000	33.0000	33.0000	0.0	366
367.0000	33.0000	33.0000	0.0	367
368.0000	33.0000	33.0000	0.0	368
369.0000	33.0000	33.0000	0.0	369
370.0000	33.0000	33.0000	0.0	370

DISTRIBUTION LIST

Defense Documentation Center Cameron Station, Bldg 5 Alexandria, VA 22314	20
Hughes Helicopter Co MS T 22 B, ATTN: Michael R. Kane Centinela Ave. At Teale St. Culver City, CA. 90230	5
Prof. M. A. Chace University of Michigan 1210 East Engineering Ann Arbor, MI 48109	2
Commander Picatinny Arsenal ATTN: Technical Library Dover, N.J. 07801	1
Commander Picatinny Arsenal ATTN: Phillip E. Townsend Dover, N.J. 07801	5
Commander Rock Island Arsenal (Technical Library) Rock Island, IL 61201	1
Commander AVSCOM Arsenal ATTN: PM-AAH, M. Corgiat 12th & Spruce Streets St. Louis, MO 63166	1
Commander Rock Island Arsenal ATTN: IBEA, R. Stickling Rock Island, IL 61201	1

AD______________ACCESSION NO.______________

UNCLASSIFIED

Small Caliber Weapons Systems Laboratory
ARRADCOM (West) DRDAR-SC
Rock Island Arsenal, Rock Island, IL 61201

A Mathematical Model of the 30MM Advanced Medium Caliber Weapon System (AMCAWS-30)

Prepared by: Michael R. Kane

Technical Report R-TR-77-017

Pages, Incl Figures, 265

A mathematical model for the AMCAWS-30MM weapon is developed using the generalized d'Alembert force equations. The development of the one degree of freedom differential equation of motion for the weapon is shown. The equation accounts for operations including feed, eject, chamber locking, round crush-up, chamber translation, face cam rotation, and drum cam rotation. The resultant equation is numerically

1. High impulse
2. Externally powered
3. Mathematical model
4. Numerical integration
5. d'Alembert force method
6. Computer program
7. Weapon model
8. FORTRAN

AD______________ACCESSION NO.______________

UNCLASSIFIED

Small Caliber Weapons Systems Laboratory
ARRADCOM (West) DRDAR-SC
Rock Island Arsenal, Rock Island, IL 61201

A Mathematical Model of the 30MM Advanced Medium Caliber Weapon System (AMCAWS-30)

Prepared by: Michael R. Kane

Technical Report R-TR-77-017

Pages, Incl Figures, 265

A mathematical model for the AMCAWS-30MM weapon is developed using the generalized d'Alembert force equations. The development of the one degree of freedom differential equation of motion for the weapon is shown. The equation accounts for operations including feed, eject, chamber locking, round crush-up, chamber translation, face cam rotation, and drum cam rotation. The resultant equation is numerically

1. High impulse
2. Externally powered
3. Mathematical model
4. Numerical integration
5. d'Alembert force method
6. Computer program
7. Weapon model
8. FORTRAN

AD______________ACCESSION NO.______________

UNCLASSIFIED

Small Caliber Weapons Systems Laboratory
ARRADCOM (West) DRDAR-SC
Rock Island Arsenal, Rock Island, IL 61201

A Mathematical Model of the 30MM Advanced Medium Caliber Weapon System (AMCAWS-30)

Prepared by: Michael R. Kane

Technical Report R-TR-77-017

Pages, Incl Figures, 265

A mathematical model for the AMCAWS-30MM weapon is developed using the generalized d'Alembert force equations. The development of the one degree of freedom differential equation of motion for the weapon is shown. The equation accounts for operations including feed, eject, chamber locking, round crush-up, chamber translation, face cam rotation, and drum cam rotation. The resultant equation is numerically

1. High impulse
2. Externally powered
3. Mathematical model
4. Numerical integration
5. d'Alembert force method
6. Computer program
7. Weapon model
8. FORTRAN

AD______________ACCESSION NO.______________

UNCLASSIFIED

Small Caliber Weapons Systems Laboratory
ARRADCOM (West) DRDAR-SC
Rock Island Arsenal, Rock Island, IL 61201

A Mathematical Model of the 30MM Advanced Medium Caliber Weapon System (AMCAWS-30)

Prepared by: Michael R. Kane

Technical Report R-TR-77-017

Pages, Incl Figures, 265

A mathematical model for the AMCAWS-30MM weapon is developed using the generalized d'Alembert force equations. The development of the one degree of freedom differential equation of motion for the weapon is shown. The equation accounts for operations including feed, eject, chamber locking, round crush-up, chamber translation, face cam rotation, and drum cam rotation. The resultant equation is numerically

1. High impulse
2. Externally powered
3. Mathematical model
4. Numerical integration
5. d'Alembert force method
6. Computer program
7. Weapon model
8. FORTRAN

integrated to obtain the time response of position, velocity, acceleration, and force for the major components. The solution is based on the known drive motor characteristics.

integrated to obtain the time response of position, velocity, acceleration, and force for the major components. The solution is based on the known drive motor characteristics.

integrated to obtain the time response of position, velocity, acceleration, and force for the major components. The solution is based on the known drive motor characteristics.

integrated to obtain the time response of position, velocity, acceleration, and force for the major components. The solution is based on the known drive motor characteristics.