MEMORANDUM REPORT ARBRL-MR-03110

(Supersedes IMR No. 678)

ANALYSIS OF MAN-IN-THE-LOOP CONTROL SYSTEMS IN THE PRESENCE OF NONLINEARITIES

Robert T. Gschwind
Irving L. Chidsey

June 1981

US ARMY ARMAMENT RESEARCH AND DEVELOPMENT COMMAND
BALLISTIC RESEARCH LABORATORY
ABERDEEN PROVING GROUND, MARYLAND

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
MEMORANDUM RE[illegible] ARBRL-MR-03110

2. GOVT ACCESSION NO.
AD-A101 574

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
ANALYSIS OF MAN-IN-THE-LOOP CONTROL SYSTEMS IN THE PRESENCE OF NONLINEARITIES

5. TYPE OF REPORT & PERIOD COVERED

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
Robert T. Gschwind
Irving L. Chidsey

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS
US Army Ballistic Research Laboratory
ATTN: DRDAR-BLB
Aberdeen Proving Ground, MD 21005

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
RDT&E 1L161102A91A

11. CONTROLLING OFFICE NAME AND ADDRESS
USA Armament Research & Development Command
US Army Ballistic Research Labortory
ATTN: DRDAR-BL
Aberdeen Proving Ground, MD 21005

12. REPORT DATE
JUNE 1981

13. NUMBER OF PAGES
45

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)

15. SECURITY CLASS. (of this report)
UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)
Approved for public release: distribution unlimited.

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

18. SUPPLEMENTARY NOTES
This report supersedes IMR No. 678, dated March 1980.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
Feedback Control
Human Control
Nonlinear Systems

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

The BRL and HEL are jointly investigating the effects of system nonlinearities on the accuracy of turret control systems with human operators in the loop. The system response at very low rates (one milliradian per second and less) is degraded because of the increased relative importance of nonlinear elements such as coulomb friction, backlash, and dead space. Good low rate response is necessary for accurate tracking of long range targets with laser designators and guided missile directors. This report describes the first phase of the BRL contribution to the joint project. (Cont'd on reverse side)

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

UNCLASSIFIED 393471

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Item 20. ABSTRACT (Continuation)

This phase developed a simplified simulation of a turret control with a human transfer function. There is an adaptive algorithm to adjust the coefficients of the human transfer function to account for changes in the system characteristics. Backlash, coulomb friction, and dead space are introduced and their effects on system response and loop performance are documented.

The next phase will adjust the simulation to agree with the system response of a concurrent turret measurement program. It will compare the loop performance to a concurrent experiment with a real man-in-the-loop. It will relate loop performance (tracking accuracy) to system response at low rates.

TABLE OF CONTENTS

LIST OF FIGURES

PRECEDING PAGE BLANK-NOT FILMED

I. STATEMENT OF THE PROBLEM

The goal of this project is to improve the accuracy of rate control systems. The traditional direct-fire gun-type turret has an accuracy requirement on the order of 0.2 milliradians which is sufficient for ranges of from one to two kilometers. Laser designators and guided missile directors are designed for twice the range and therefore require twice the accuracy, i.e., approximately 0.1 milliradians. Even the best modern turret systems have trouble achieving this order of accuracy all the time. This effort attempts to determine which system characteristics are limiting tracking accuracy. It will then be possible to write specifications in terms of hardware characteristics rather than in terms of system performance goals.

The premise of this approach is that the solution can be found in the nonlinear behavior of the turret systems at very low rates. The tracking rates of interest and changes in rate that are called for to achieve precise tracking are very small. A 30-kilometer-per-hour target, for example, at a range of 4 kilometers and a heading of 30 degrees would result in a 1-milliradian-per-second crossing rate. The gunner will command small changes about this nominal rate in an attempt to reduce the tracking error. We can infer the magnitude of these changes to be less than 0.3 milliradians per second (referred to the output) by the following logic. Since the desired error tolerance is on the order of 0.1 milliradians the gunner should have the ability to make commands at his bandwidth (3 radians per second) which would result in amplitudes of 0.1 milliradian. Assuming a sinusoidal input of 0.3-milliradian-per-second amplitude and 3-radian-per-second frequency the output would be a sinusoid with 0.1-milliradian amplitude.

The nonlinear elements in the turret, i.e., backlash, coulomb friction, and deadspace, will have more effect on the turret response at low rates and for small commands. The gunner will notice a decrease in gain and an increase in phase lag. He will attempt to compensate for the turret changes within limits, but eventually the tracking error will increase above what it would be for a linear system. Moreover, the human's ability to compensate will vary considerably with training and between people. The result will be unpredictable system performance due to operator differences rather than hardware differences.

Traditionally the turret specifications for low rate tracking have been system performance specifications with a human operator in the loop, e.g., 0.1-milliradian root-mean-square error when tracking a 1.0-milliradian-per-second target for 10 seconds. It would probably be better to specify the accuracy requirements against sinusoidal inputs and better still to specify the system characteristics without a human operator in the loop. Enough analysis of linear control systems with human operators has been done to relate accuracy to system characteristics (gain and phase lag) and input power spectra at least for linear control systems. This report addresses the system performance at low rates when the nonlinear system characteristics are important. Ultimately this work should lead to specifications of system characteristics which are required to achieve any specified degree of accuracy.

II. APPROACH

The overall solution to the problem requires three distinct tasks to be carried out by three different organizations.

a. First, there is the characterization of a typical turret control system at very low rates. An M60A3 turret control is being used because it is available now and in the future and it is typical of current military turret control systems. Frequency response and transient response will be measured at various amplitudes from low rates near or below threshold up to high rates in the linear region. This work is being done under contract by General Electric, Pittsfield, Massachusetts.

b. Second, there are experiments with humans in the loop and with computer simulated turret response. These tests will characterize the human operator and they will allow for some limited parameter variations of turret response. Human Engineering Laboratory is doing this work.

c. Third, there is an analysis task with both the turret response and the human operator simulated by a computer. This task allows for parametric variations of turret response just as with the human experiments; but the computer allows for almost unlimited parameter variations because the trials are faster and they are without random human variation. This report describes the Phase I effort on the third task.

The Phase I effort develops the methods and computer codes based on assumed system characteristics. The Phase II effort will do it all again but with accurate system data from the other tasks.

The main task of Phase I was to develop a self-optimizing human operator model. The model must minimize root mean square (rms) tracking error, subject to constraints on human behavior, in a consistent and rational manner. This model was then used to determine the effects of various kinds of nonlinearities on system performance.

III. PROCEDURE FOR MAN-MODEL OPTIMIZATION

The Phase I effort had a goal of developing techniques for evaluating nonlinearities. This included the man model, the adaptive algorithm for the man model, models of nonlinearities and some limited data to show how well the models work. Complicated turret dynamics were not important at this stage; consequently a very simplified model of turret response was incorporated. Figure 1 shows a block diagram of the control loop and the linear models that were used for the man and turret. The reference signal was either white noise, a sinusoid, or a maneuvering tank. The human model was a conventional linear model from Sheridan[1] to which a noise remnant was added for reasons that will be explained later. The turret response was given a time constant of 0.1 seconds and a gain of 0.01 radians per second per radian. These parameters are about right for a tank turret at very low rates. Nonlinearities were introduced in the digital simulation at the points shown in Figure 1.

The notation in Figure 1 was chosen to be consistent with the digital simulation shown in its entirety in the Appendix. The linear transfer functions were simulated by the step invariant zeta transform[2] method.

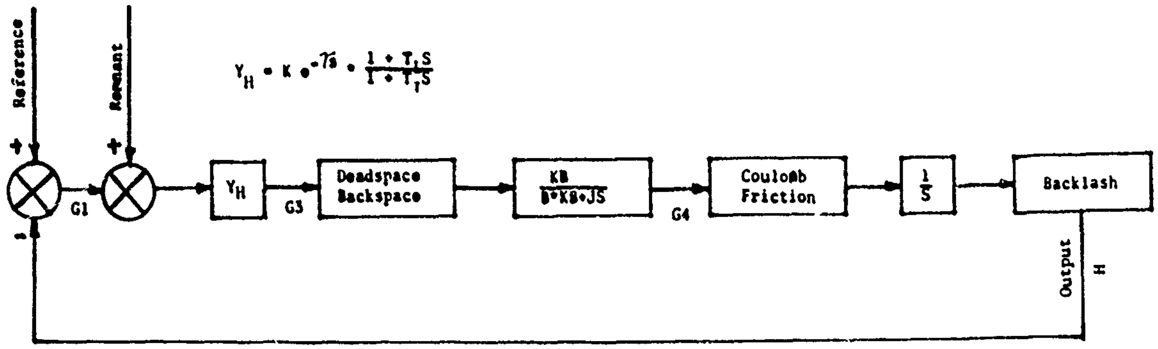

Figure 1. Block Diagram of Control Loop .

[1]Man-Machine Systems, Sheridan and Ferrell, 1974, MIT Press.

[2]Digital Signal Analysis, Stearns, 1975, Hayden Book Company Inc.

An algorithm for optimizing the human transfer function was developed while using white noise for a reference signal. It minimized rms tracking error but with a penalty for open loop gain margin less than 6 db and open loop phase margin less than 45 degrees. Secant functions accomplished the penalty by giving a cost factor of one at 6 db and 45 degrees and a cost factor of infinity at 0 db and 0 degrees.

secant [15 (Gain Margin -6)], secant [2 (Phase Margin -45)]

The loop was warmed up for a few seconds, run for 52 seconds at a time step of 0.01 seconds, and sampled every .05 seconds till 1024 samples of input to the man and output from the plant were stored. A fast fourier transform was taken of the input and output. The open loop gain and phase were calculated from zero to ten radians per second at intervals of one radian per second by adding the complex numbers in every eight cells and then dividing the absolute values to get gain and subtracting the angles to get phase. The computer program plots the resulting gain and phase and it calculates the cost. It selects new values for the human transfer function and then repeats the simulation and cost calculations until a minimum is established.

The human transfer function optimization algorithm worked fine with a white noise input. There was enough power at every frequency of interest to make good calculations of gain and phase. The gain calculation becomes noisy when the input (the denominator) gets near zero. The phase calculation gets noisy when either the input or output power gets too small to make an accurate phase measurement. The algorithm requires smooth monotonically-decreasing gain and phase for at least one frequency band beyond 180 degree phase lag. Such data were obtained with a white noise input with and without nonlinearities in the loop. Unfortunately real targets do not present a white noise tracking spectrum.

A realistic target motion was constructed from the following considerations:

a. The algorithm wants as much power as possible and so does the maneuvering target; therefore a course made of segments of 0.2g turns was used.

b. The target wants to move forward rather than go in a circle; therefore the turns were limited to plus and minus 45 degrees from the line of sight between the target and tracker.

c. The algorithm wants power in each one-radian-per-second frequency band, therefore the turning radius and speed were selected to produce a fundamental frequency at 0.5 radian per second so that the harmonics would fall at 1.5, 2.5, 3.5, ..., radians per second. The relationships for radial acceleration were used.

radial acceleration = $rw^2 = v^2/r = 0.2g$

r = radius = 8 meters

w = angular rate = 0.5 radians per second

g = acceleration of gravity = 10 meters per second squared

d. The range to target was set at four kilometers to reduce the angular tracking rates to the low rates of interest.

Maximum rate = 4 m/s • (sin 45)/4 km = 0.7 mrad/s

These considerations resulted in a course which was roughly sinusoidal at 0.5 radians per second. The abrupt changes in radial acceleration every 90 degrees of turn gave strong enough harmonics to allow the algorithm to calculate gain and phase when there were no nonlinearities in the loop, although it did require double precision in the calculations. When nonlinearities were introduced the gain and phase curves became noisy and the optimization algorithm would not work.

Additional power was required at both the input and output at frequencies of interest (1.5 through 8.5) to make the optimization algorithm work properly. Fortunately the addition of noise power is justified as the so-called remnant term of the human transfer function (Y_H). Although it can be added either before or after the linear portion of Y_H, here it is added before Y_H to enhance the optimization algorithm operation, but after the rms calculation to avoid improperly affecting it. The appropriate amount of noise was calculated by the following steps:

a. Sheridan page 241 shows the noise power to be 20 percent of the total power at the output of the man. Page 242 shows it to be uniform with frequency.

b. The Y_H can be approximated by a pure gain for power calculations, because the transportation delay does not affect power and the lead-lag terms are very small.

c. Sample trials have shown the rms error to be approximately 0.3 milliradians.

d. There are approximately eight bands of interest.

e. Therefore power from a sinusoid with an rms amplitude of 0.05 milliradians should be introduced at each frequency band.

$$0.3\sqrt{0.2/8} = 0.05$$

This additional power helped but sooner or later as the magnitude of the nonlinearities was increased the algorithm would become too noisy. There are still a couple of tricks to try, i.e., longer running time and extrapolation of the phase curve to 180 degrees lag rather than interpolation as was done here. These will be tried in the Phase II effort. The current effort was finished by using the simple expedient of minimizing rms error and forgetting about the phase and gain calculations. This procedure raised the gain until the system went unstable. It is a consistent method but it is probably not typical of human operation.

IV. EFFECTS OF NONLINEARITIES ON THE TURRET RESPONSE

Three nonlinearities were added one at time. Deadspace was added at G3 on Figure 1. It corresponds to the deadspace in a gunner's control for the first couple of degrees of rotation. Coulomb friction was added at G4. It corresponds to the friction on the turret itself. Backlash was applied to the output at H and it can also be applied to the input at G3.

Figure 2 through Figure 6 show the effects of these nonlinearities on the gain and phase characteristics of the turret, i.e., from G3 to H. Figure 2 shows the turret with no nonlinearities for a comparison. The turret parameters were B = 10, KB = 1.0, and J = 1.0. These curves were generated by using a single sinusoid by itself at each frequency. The family of curves in Figure 3 through Figure 6 represents successive doubling

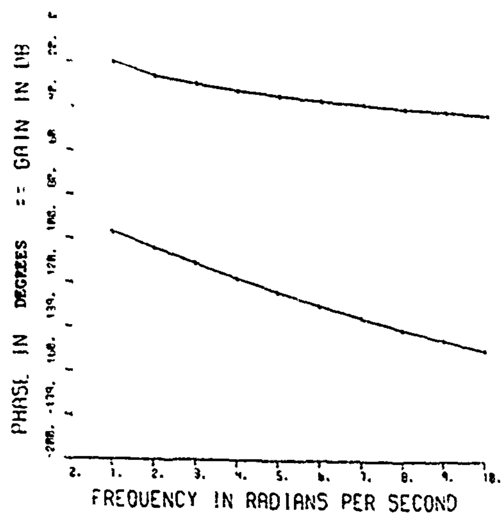

Figure 2. Turret Response Without Nonlinearities.

of the ratio of the nonlinearity to the input. It was done by changing the input rather than the turret. This procedure is equivalent to the way the test data would be collected on a turret. The input amplitudes were 0.125, 0.25, 0.5, 1, 2, 4 and 8 milliradians.

The phase lag shown in Figure 2 points out a limitation of this methodology. A quick calculation would predict a lag of 135 degrees at 10 radians per second. The figure shows a lag of 150 degrees. The difference of 15 degrees must be due to the analysis technique which uses the Zeta transform and the Fast Fourier Transform. The time step used with the simulation can account for 6 degrees of error (0.01 seconds x 10 radians per second x 60 degrees per radian). The rest is either due to the FFT or it is unknown.

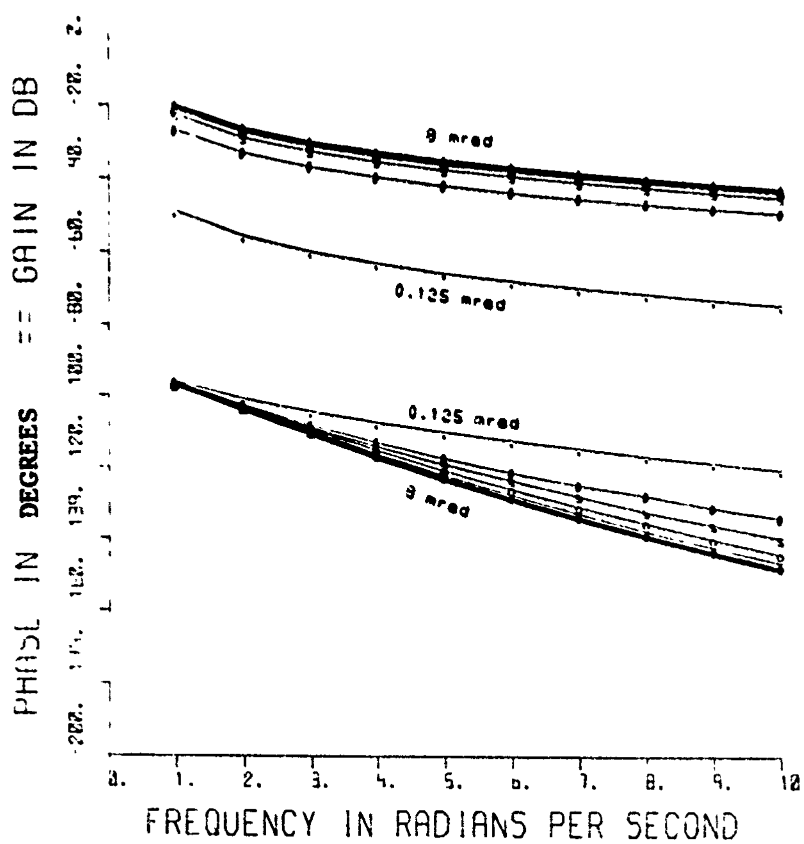

Figure 3. Turret Response with Coulomb Friction.

Figure 3 shows the effect of coulomb friction applied to G4. The magnitude was 0.1 pound-foot applied to a turret of one slug-foot-squared polar moment. The ratio is about right since a tank has a 22,000 slug-feet-squared polar moment and about 2000 pound-feet of coulomb friction referred to the turret. The gain decreased as the input was decreased but the phase lag decreased as well.

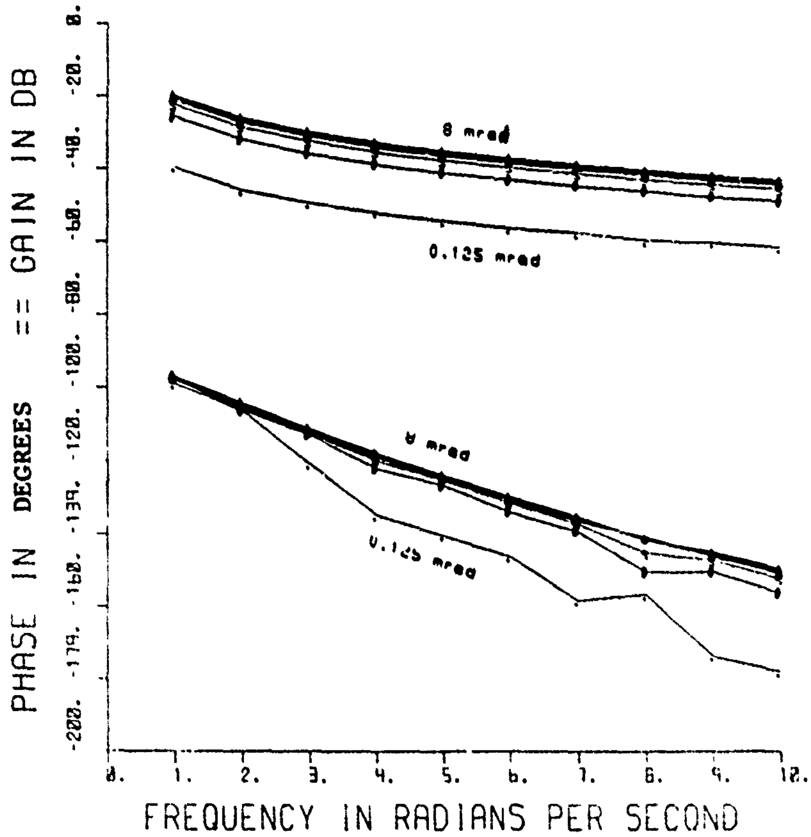

Figure 4. Turret Response With Dead Space at the Control Handle.

Figure 4 shows the effect of deadspace applied at G3. The magnitude of the deadspace was 0.1 milliradians. A typical turret might have 0.04 radians deadspace at the turret control handle. The turret gain during these turret response runs was 0.1 compared to a typical turret gain of 0.02 radians per second per radian. Obviously the problem will require new coefficients for a quantitative analysis but these figures show the trends.

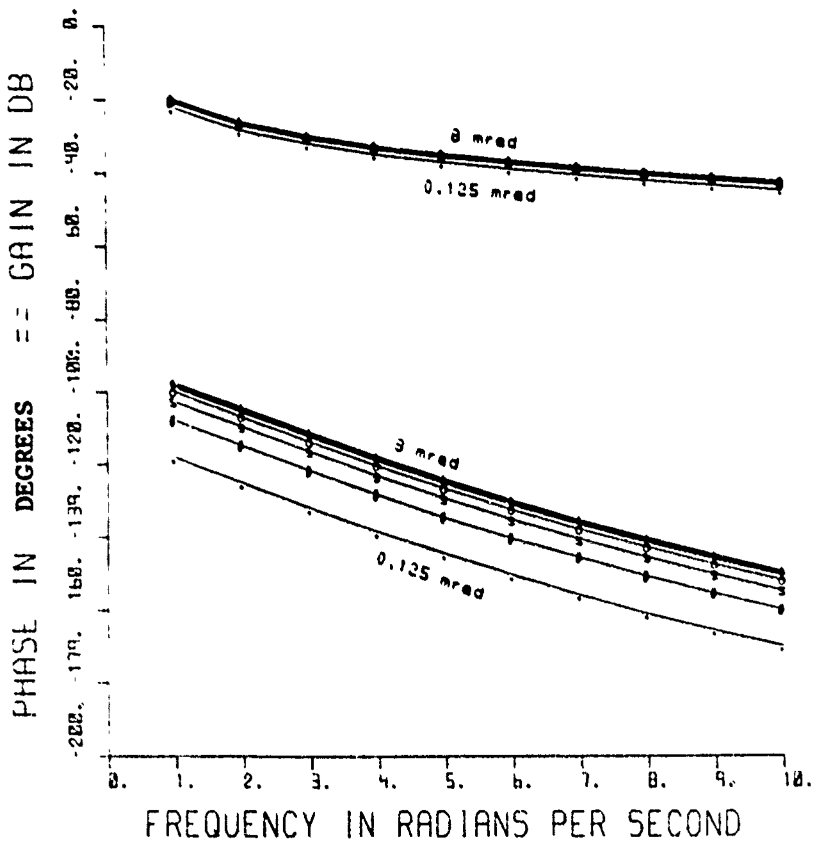

Figure 5. Turret Response With Backlash at the Control Handle.

Figure 5 shows the effect of backlash applied at the turret control. The magnitude of the backlash was 0.04 milliradians. Once again the level chosen was not necessarily representative of real turrets, however it does show the relative effects of backlash on gain and phase. Backlash at the control handle will cause a phase lag without changing the gain substantially.

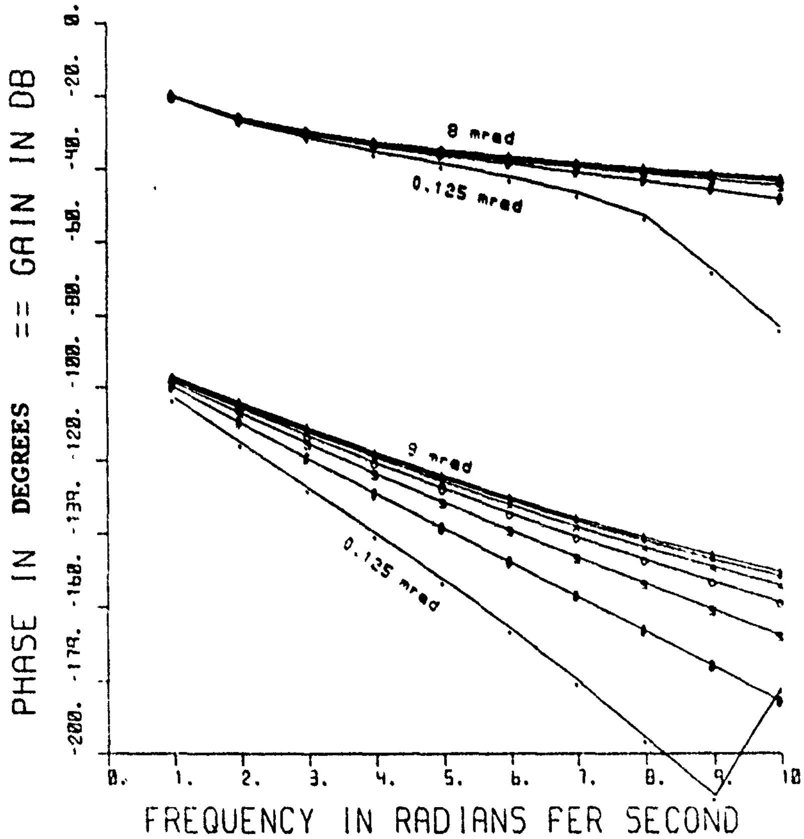

Figure 6. Turret Response With Backlash at the Turret Output.

Figure 6 shows the effect of 0.001 milliradian of backlash on the output. Here the backlash can be seen to have a greater effect at higher frequencies as compared to backlash on the input. The reduction in amplitude with increased frequency at the output causes this effect.

The objective of presenting these figures is to indicate that it will be possible to shape the gain-phase characteristics of the turret model. This will be done when the test data from the tank turret become available.

V. EFFECTS OF NONLINEARITIES ON LOOP RESPONSE

The intent at this point was to calculate the rms error for the closed-loop system when tracking the target course developed earlier. The gain-phase plots for the open-loop response of the man and turret were also of interest, but as explained earlier the gain-phase plots were usable only for the condition with low levels of nonlinearities. Figure 7 shows these plots for a condition with no nonlinearities. This condition had a phase margin of 41 degrees, a gain margin of 4.8 db and an rms error of 0.31 milliradians. The cross-over frequency (0 db gain) was 3.0 radians per second.

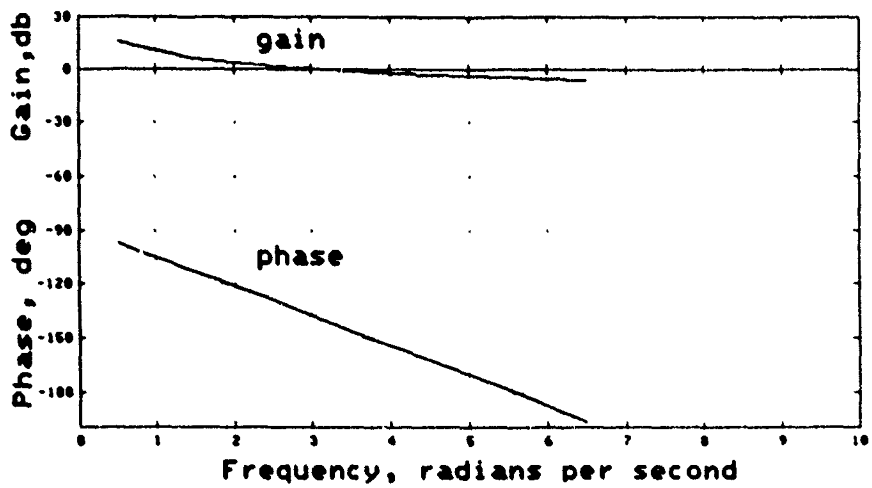

Figure 7. Gain-Phase Plot Without Nonlinearities.

When the nonlinearities were added the loop was optimized for minimum rms error. The rms error for no nonlinearities dropped to 0.26 milliradians but the loop was not nearly as stable. The phase margin was only 13 degrees and the gain margin was 1.4 db. The growth in rms error with increased levels of nonlinearities is shown in Figures 8, 9, and 10. The deadspace in Figure 8 is at the control handle. The turret gain was changed to 0.01 (BK = 0.10, B = 100) for these runs. The coulomb friction in Figure 9 was applied to G4. The backlash in Figure 10 was applied to the control handle.

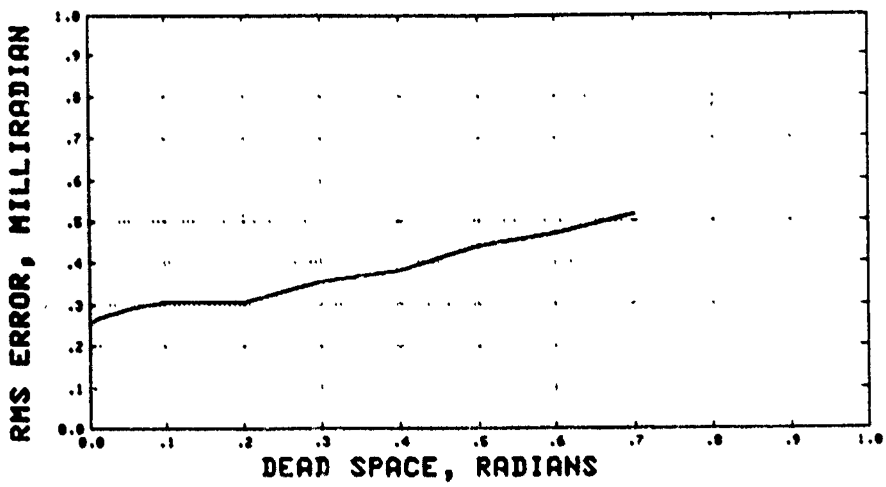

Figure 8. Tracking Error with Dead Space at the Control Handle.

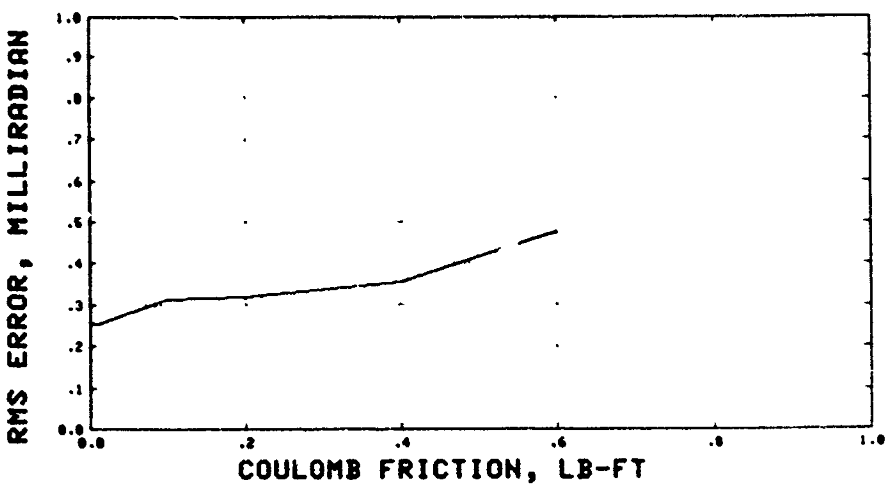

Figure 9. Tracking Error With Coulomb Friction at the Turret Output.

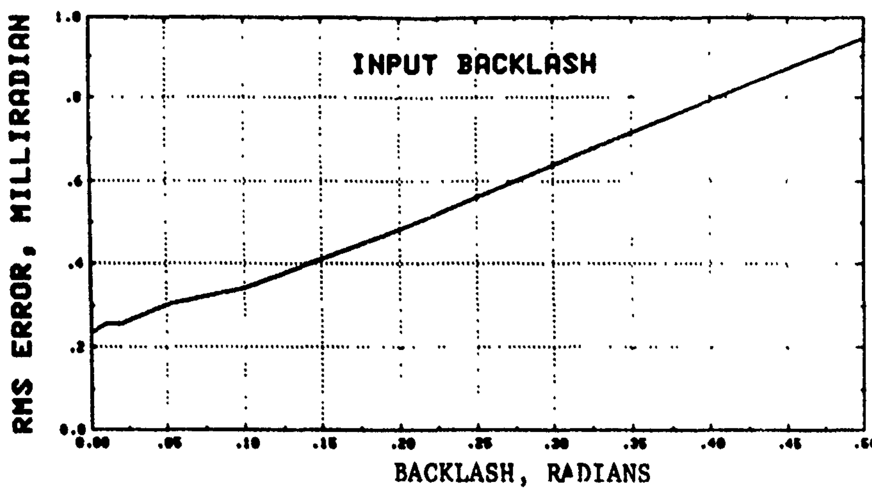

Figure 10. Tracking Error with Backlash at the Control Handle.

The only conclusion that can be reached at this time regarding Figures 8, 9, and 10 is that the level of nonlinearities that were used did have an influence on tracking error. It remains to be seen if these are the appropriate levels. The turret measurement tests will determine the appropriate levels to use. It also remains to be proven that the man-model used here is appropriate for this task. The human tracking tests will determine that.

VI. SUMMARY

A control loop with a man-model and with provision for nonlinearities was developed. An optimization algorithm for the adaptive man-model worked well for low levels of nonlinearities, but it had to be simplified to work for high levels of nonlinearities. Nonlinearities were shown to influence tracking error.

The next phase of this effort will have the benefit of quantitative descriptions of the turret response. The turret will be simulated in more detail and the correct parameter values will be used for nominal conditions. Another attempt will be made to improve the adaptive man-model to work with the appropriate nonlinearities.

APPENDIX

COMPUTER PROGRAM

Jan 2 10:41

```
c       PROGRAM CONTRL
c
c       THIS PROGRAM SIMULATES A TANK GUNNER TRACKING A TARGET.
c       ON DEMAND IT WILL ADJUST THE PARAMETERS OF THE MAN
c       TO GIVE THE LOWEST COST OR ERROR.
c       IT WILL ALSO PRODUCE BODE PLOTS ON PRINTER AND IN A
c       FILE FOR USE BY A PLOTTER.
c
c       THE PROGRAM IS INTERACTIVE, AND PROMPTS ALL INPUTS.
c       A CARRIAGE RETURN IS SUFFICIENT FOR AN ANSWER  OF
c       ZERO OR NO.
c
c       TWO PROMPTS REQUIRE MULTIPLE INPUTS
c       ON ONE LINE, SEPERATED BY COMMAS.
c         FIRST SET:
c           TRAILING ZERO VALUES MAY BE IGNORED,
c           COMMAS ARE SUFFICIENT FOR NONTRAILING ZERO FIELDS.
c           TAU         TRANSPORT DELAY IN SECONDS
c           B           VISCOUS FRICTION
c           BK          PLANT GAIN
c           BKLSH       BACKLASH AT OUTPUT IN RADIANS
c           CF          COULOMB FRICTION
c           DEDSPC      DEADSPACE AT INPUT IN RADIANS
c
c         SECOND SET:
c           TI          INTEGRATION TIME IN SECONDS
c           K           GAIN OF MAN
c           TL          LEAD TIME IN SECONDS
c
c           INSTEAD OF THE SECOND SET, DEFAULTS OF
c           TI = .01, K = 2.5 * B, AND TI = TI + 1/B
c           MAY BE CALLED BY A CARRIAGE RETURN.
c
c       THE PLANT IS NORMALIZED TO A MASS OF 1.
c
c       SUBROUTINES IN THE PACKAGE:
c
c       THE MAIN PART OF THE PROGRAM DOES ALL THE INTERACTIVE
c       CONVERSATION AND CALLS PLANT0, AUTO, FUN, AND MACHINE.
c
c       SUBROUTINE AUTO
c       SETS UP THE AUTOMATIC OPTIMIZATION
c       CALLS FUN AND FNMIN.
c       THE PARAMETER ACRCY IN AUTO TELLS FNMIN
c       THE PRECISION DESIRED.
c       THE AUTOMATIC MINIMIZATION ALSO TERMINATES IF TI
c       BECOMES LESS THAN .005
c
c       SUBROUTINE FNMIN
c       DOES THE AUTOMATIC OPTIMIZATION
c       CALLS FUN
c       IF AUTOMATIC MINIMIZATION IS CHOSEN
c       FNMIN SYSTEMATICALY VARIES
c       X( 1 ) = 1 / TI, X( 2 ) = K, AND X( 3 ) = TL
```

```
c       TO MINIMIZE THE COST RETURNED BY FUNCTION FUN.
c       X( 1 ) IS INVERTED TO AVOID NEGATIVE VALUES.
c
c       FUNCTION FUN
c       MODELS THE CONTROL LOOP
c       COMPUTES THE COST OF TRACKING
c       CALLS MAN0, PLANT1, STATS0, MAN1, TGTS, STATS1, PLANT2,
c       STATS2, STATSW, AND FFT.
c       THE PARAMETER NN IS USED AS A FLAG IN FUN:
c       NN > 0,  AUTOMATIC REDUCTION, NO BODE PLOT.
c       NN = 0,  BODE PLOT OF MAN-MACHINE SYSTEM
c         USING THE MODEL TARGET AS INPUT.
c       NN = -1,  BODE PLOT OF MACHINE WITH MODEL TARGET
c         AS INPUT TO MAN.
c         NOTE THAT IN THIS CASE THE TARGET IS FILTERED THRU
c         THE MAN AND THE PLOT IS THEREFORE AN IMPLICIT FUNCTION
c         OF THE MAN.
c       NN < -1,  BODE PLOT OF MACHINE WITH SINE WAVE INPUT.
c
c       SUBROUTINE MACHINE
c       MAKES BODE PLOTS OF THE MACHINE
c       CALLS FUN AND PLOT.
c
c       SUBROUTINE PLOT
c       PRODUCES A PRETTY BODE PLOT IN A FILE READY FOR PLOTTING
c       PLOT ASSUMES THE PLOTTING PACKAGE
c       TIG ( TERMINAL INDEPENDENT GRAPHICS ) WHICH WAS
c       WRITEN IN C AND REQUIRES A C COMPILER.
c
c       SUBROUTINE FFT
c       FAST FOURIER TRANSFORM
c
c       THE MAN:
c
c         SUBROUTINE MAN0
c         INITIALIZES THE MAN
c         CALLS TGTA0
c
c         SUBROUTINE MAN1
c         THE MAN'S PART OF THE CONTROL LOOP
c         CALLS TGTA1
c
c       THE PLANT:
c
c         SUBROUTINE PLANT0
c         INITIALIZES THE PLANT
c
c         SUBROUTINE PLANT1
c         RESETS PLANT AT START OF EACH RUN
c
c         SUBROUTINE PLANT2
c         THE MACHINERY'S PART OF THE CONTROL LOOOP
c
c       THE STATS:
```

```
c
c         SUBROUTINE STATS0
c         INITIALIZES STATISTICS PACKAGE
c
c         SUBROUTINE STATS1
c         COLLECTS THE DATA
c
c         SUBROUTINE STATS2
c         FINDS RMS ERROR, MEAN INPUT OFFSET, ETC
c
c         SUBROUTINE STATSW
c         WRITES THE STATS
c
c       THE TARGET:
c
c         SUBROUTINE TGTA0
c         SETS UP TARGET
c
c         SUBROUTINE TGTA1
c         RUNS TARGET
c
c         SUBROUTINE TGTS
c         SINE WAVES FOR BODE PLOTS
c
c       THE PROGRAM WAS WRITEN TO BE RUN ON A PDP 11/70
c       USING THE CULC F4P COMPILER AND THE UNIX OPERATING
c       SYSTEM.
c       THE PROGRAM IS IMPLICIT DOUBLE PRECISION,
c       AND USES COMPLEX ARITHMATIC.
c       GENERIC NAMES HAVE BEEN USED FOR FUNCTIONS, IE ABS,
c       AND THE F4P COMPILER SELECTS THE PROPER FUNCTION,
c       IE ABS, CABS, DABS, ETC.
c
c
        common b, delta, deltat,
     +  jmax, jmaxt, jmod, k, nc, nt, pi, tmax, unlin
        common / x / x( 10 )
        external fun
        real * 8 jj, kk

        write ( 6, 5 )
5       format ( ' NOW TYPE VALUES FOR TAU, B, BK, BACKLASH, ' /
     +  ' COULOMB FRICTION, AND DEAD SPACE ' )
        accept 10, tau, b, bk, bklsh, cf, dedspc
10      format ( 7f10.0 )
        unlin = abs( bklsh ) + abs( cf ) + abs( dedspc )
        k = 10
        jmod = 5
        jmax = 2 ** k
        jmaxt = jmax * jmod
        tstep = 2.
        pi = 3.1415926535
        nc = 8
        nn = 0
```

```
      fno = no
      tmax = fno * 2. * pi
      delta = tmax / float( jmax )
      deltat = tmax / float( jmaxt )
      write ( 6, 15 ) tau, delta, b, bk, bklsh, of, dedspc, jmax
15    format ( / '  TAU  DELTA   B    BK   BKLSH  CLFR DEDSP JMAX ' /
     + 2f6.3, f6.1, f6.3, 3pf6.2, 0pf6.3, 3pf6.2, i5 / )
      nt = tau / deltat + .5
      call plant0( b, bk, bklsh, bklshi, of, dedspc, deltat )
      write ( 6, 20 )
20    format ( '  DO YOU WANT TO LOOK AT THE MACHINERY? ' /
     + ' TO ANSWER, TYPE 1 FOR YES, 0 FOR NO. ' )
      accept 30, m
      write ( 6, 25 )
25    format ( / ' DO YOU WANT AUTOMATIC REDUCTION ? ' /
     + ' TYPE 1 FOR YES, OR 0 FOR NO ' )
      accept 30, n
30    format ( i10 )
35      write ( 6, 40 )
40      format ( / ' NOW TYPE VALUES FOR TI, K, AND TL ' )
        accept 10, ti, kk, tl
        if ( ti .gt. 0. ) go to 45
          ti = .01
          kk = 2.5 * b
          tl = ti + 1. / b
45      x( 1 ) = 1. / ti
        x( 2 ) = kk
        x( 3 ) = tl
        if ( n .gt. 0 ) call auto
        cost = fun( x, nn )
        if ( m .eq. 1 ) call machine
        if ( n .le. 0 ) go to 35
      stop

      end

      subroutine auto

      implicit double precision ( a-h, o-z )
      common b, delta, deltat,
     + jmax, jmaxt, jmod, kj, nc, nt, pi, tmax, unlin
      dimension eps( 10 )
      common / x / x( 10 )
      real * 8 kk
      external fun

      k = 1
      nn = 0
      cost = fun( x, nn )
      ti = 1. / x( 1 )
      kk = x( 2 )
      tl = x( 3 )
      write ( 6, 15 ) cost, ti, kk, tl
15    format ( / ' THE INITIAL COST WAS ', 3pf8.3 /
```

```
      +   ' FOR TI = ', 0pf7.4, ' KK = ', f5.1, ', TL = ', f7.4 //
      +   '    TI       KK       TL       RMS  PHASMR GAINMR  COST   SLOPE' / )
          acrcy = .05
          e = 5.
          n = 3
          do 5 i = 1, n
5           eps( i ) = acrcy * x( i )
          call fnmin( n, x, cost, fun, e, eps, k )
          ti = 1. / x( 1 )
          kk = x( 2 )
          tl = x( 3 )
          write ( 6, 10 ) cost, ti, kk, tl
10        format ( / ' THE MINIMUM COST WAS ' 3pf6.3 /
      +    ' FOR TI = ', 0pf7.4, ', KK = ', f5.1, ', TL = ', f7.4 )
          return

          end

          function fun( x, nn )

          implicit double precision ( a-h, o-z )
          common b, delta, deltat,
      +    jmax, jmaxt, jmod, k, nc, nt, pi, tmax, unlin
          common / gp / gaan( 10, 7 ), phaas( 10, 7 )
          dimension gain( 81 ), phase( 81 )
          dimension gl( 100 )
          dimension frin( 1024 ), fiin( 1024 ), frout( 1024 ),
      +    fiout( 1024 )
          dimension x( 10 )
          complex cin, cout, crin, crout, scrin, scrout, cxx, cyy
          real*4 fiini, fiouti, frini, frouti
          data costmn / 1. /

          if ( x( 1 ) .le. 0. .or. x( 2 ) .le. 0. .or. x( 3 ) .le. 0. )
      +    cost = 1.e+2
          if ( x( 1 ) .le. 0. .or. x( 2 ) .le. 0. .or. x( 3 ) .le. 0. )
      +    go to 900
          call man0( deltat, nt, x )
          call plan''( h )
          points = jmax
          if ( nn .eq. 0 ) write ( 6, 55 )
55        format ( / 27x, 4h-1.0, 6x, 4h-0.5, 7x, 3h0.0, 7x, 3h0.5,
      +    '        1.0' /
      +    '    SEC     REF      IN     OUT +', 4( 9x, 1h+ ) )
          if ( nn .eq. -1 ) write ( 6, 56 )
56        format ( 27x, '-250       -125.       0.0       125.'
      +    '        250' /
      +    '    SEC     REF      IN     OUT +', 4( 9x, 1h+ ) )
        call stats0( delta, jmax )
        jcm = 2
        jmx = jmaxt
        jcl = jcm
        if ( nn .ge. -1 ) jcl = 1
c          NOW MODEL THE CONTROL LOOP
```

```
        do 101 jct = jcl, jcm
          jm = 0
          if ( jct .eq. jcm ) jmx = jmaxt
          do 100 j = 1, jmx
            if ( nn .ge. - 1 ) call manl( h, j, nt, diffdt, difft, g3 )
            if ( nn .lt. -1 ) call tgts( j, g3 )
            if ( jct .lt. jcm ) go to 90
              if ( mod( j, jmod ) .ne. 0 ) go to 90
                jm = jm + 1
              if ( nn .ge. 0 ) frin( jm ) = diffdt
              if ( nn .lt. 0 ) frin( jm ) = g3
              frout( jm ) = h
              fiin( jm ) = 0.
              fiout( jm ) = 0.
              if ( nn .ge. -1 )
     +         call stats1( difft, frin( jm ), frout( jm ), jm )
90          call plant2( g3, h )
100       continue
101     continue
        if ( nn .ge. -1 )
     + call stats2( avein, aveout, rms )
          do 120 j = 1, jmax
            tj = j
            frin( j ) = frin( j ) - avein
            frout( j ) = frout( j ) - aveout
120        continue
121     continue
          if ( nn .gt. 0 .or. nn .lt. -1 ) go to 170
          do 150 j = 1, 199, 3
            t = float( j ) * delta
            if ( nn .eq. 0 ) n = frin( j ) * 20000.
            if ( nn .lt. 0 ) n = frin( j ) * 80.
            ref = frin( j ) + frout( j )
            if ( n .gt. -21 )
     +       write ( 6, 140 ) t, ref, frin( j ), frout( j )
            if ( n .le. -21 ) write ( 6, 142 ) t, ref, frin( j ), frout( j )
150       continue
140       format ( 1x, f6.2, 3p3f7.2, <n+21>x, 1h* )
141       format ( 1x, 4e10.3 )
142       format ( f7.2, 3p3f7.2 )
170     if ( nn .eq. 0 ) call statsw
175       call fft ( frin, fiin, k )
          call fft ( frout, fiout, k )
          nav = nc
          if ( nn .lt. -1 ) nav = 1
          irads = 1
          iradm = 42
          if ( nn .lt. -1 ) iradm = 1 - nn
          if ( nn .lt. -1 ) irads = iradm
          do 500 irad = irads, iradm
            scrin = 0.
            scrout = 0.
            do 400 j = 1, nav
              i = nav * ( irad - 1 ) + j
```

```
            frini = frin( i )
            fiini = fiin( i )
            cin = cmplx( frini, fiini )
            frouti = frout( i )
            fiouti = fiout( i )
            cout = cmplx( frouti, fiouti )
            crin = cin * conjg( cin )
            crout = cout * conjg( cin )
      if ( abs( crin ) .le. 1.e-10
     + .or. nn .lt. -1 .or. nn .gt. 0
     + .or. i .gt. 80 ) go to 391
      cxx = crout / crin
      cyy = abs( cxx )
      gan = 20. * log10( real( cyy ) )
      faz = atan2( aimag( crout ), real( crout ) )
      faz =  180. * faz / pi
      eni = real( crin )
      eni = sqrt( eni )
      cyy = abs( cout )
      eno = real( cyy )
      if ( abs( eno ) .gt. 1.e-10 ) write ( 6, 390 )
     + i, frin( i ), fiin( i ), frout( i ), fiout( i ),
     + gan, faz, eni, eno
390   format ( i5, 4f8.4, 2f8.2, 2f8.4 )
391   continue
            scrin = scrin + crin
            scrout = scrout + crout
400        continue
           if ( scrout .eq. 0. ) write ( 6, 450 )
           if ( abs( scrout ) .gt. 1.e+10 )write ( 6, 451 )
     +      1. / x( 1 ), x( 2 ), x( 3 )
451        format ( / f7.4, f7.2, f7.4, ' UNSTABLE ' / )
450        format (  / ' INPUT TOO SMALL ' /  )
           if ( scrout .eq. 0. .or. abs( scrout ) .gt. 1.e+10 )
     +      cost = 1.e+1
           if ( scrout .eq. 0. .or. abs( scrout ) .gt. 1.e+10 )
     +      go to 900
           cxx = 0.
           if ( abs( scrin ) .gt. 1.e-30 ) cxx = scrout / scrin
           cyy = abs( cxx )
           gain( irad ) = real( cyy )
           phase( irad ) = atan2( aimag( scrout ), real( scrout ) )
           if ( phase( irad ) .gt. 0. )
     +      phase( irad ) = phase( irad ) - 2. * pi
500      continue
         if ( nn .gt. 0 .or. nn .lt. -nc ) go to 540
         if ( nn .eq. 0 ) idbmn = -10
         if ( nn .eq. -1 ) idbmn = -60
         if ( nn .lt. -1 ) idbmn = -70
         idbmx = idbmn + 30
         write ( 6, 510 ) ( idb, idb = idbmn, idbmx, 10 )
510      format ( / 20x, 'PHASE SHIFT', 20x, 7hDB GAIN /
     +   '  FREQ  PH  GAIN -180', 15x, '-90',
     +    i4, 3( 7x, i3 ) )
```

```
        write ( 6, 530 )
530     format ( 1x, 1h+, 4( 8x, 1h+ ), 3x, 1h+, 3( 9x, 1h+ ) )
540     xx = 180. / pi
        radf = 2. * pi * float( nav ) / tmax
        npm = 52
        if ( nn .lt. 0 ) npm = npm - idbmn - 10
        i1 = 1
        im = 40
        if ( nn .lt. -1 ) i1 = 1 - nn
        if ( nn .lt. -1 ) im = 1 - nn
        do 600 i = i1, im
          if ( gain( i ) .le. 1. .or. gain( i+1 ) .gt. 1. )
     +     go to 590
          phasmr = 180. + xx * ( phase( i ) -
     +     ( phase( i ) - phase( i+1 ) )
     +     * ( gain( i ) - 1. )
     +     / ( gain( i ) - gain( i+1 ) ) )
          if ( i .eq. 1 )
     +     slope = 20. * log10( gain( i ) / gain( i+1 ) )
     +     / log10( 1. / 3. )
          if ( i .gt. 1 )
     +    slope = 20. * log10( gain( i-1 ) / gain( i+2 ) )
     +     / log10( ( float( i-1 ) - .5 )
     +     / ( float( i+1 ) + .5 ) )
590       n = phase( i ) * xx
          np = n / 5
          gan = 20. * log10( gain( i ) )
          m = gan
          rf = radf * ( float( i) - .5 )
          if ( nn .lt. -1 ) rf = rf - .5 / float( nc )
          nrf = rf + .001
          if ( nn .eq. -nc ) iii = iii + 1
          if ( nn .ge. -1 .or. nrf .lt. 1 .or. iii .lt. 1 ) go to 595
          if ( nn .lt. -1 ) gaan( nrf, iii ) = gan
          if ( nn .lt. -1 ) phans( nrf, iii ) = phase( i ) * xx
595       continue
          if ( nn .le. 0 .and. np .gt. -55 .and. m .gt. 2-npm )
     +     write ( 6, 610 ) rf, n, gan
          if ( nn .le. 0 .and. ( np .le. -55 .or. m .le. 2-npm )
     +    .and. gan .gt. -100. )
     +     write ( 6, 611 ) rf, n, gan
          if ( phase( i ) .gt. -pi .and. phase( i+1 ) .lt. -pi )
     +     gainmr = - 20. * log10( gain( i )
     +     - ( gain( i ) - gain( i+1 ) )
     +     * ( phase( i ) + pi )
     +     / ( phase( i ) - phase( i+1 ) ) )
        if ( nn .lt. -1 ) go to 600
          if ( phase( i ) .lt. -pi .and. gain( i ) .lt. 1. ) go to 700
600     continue
610     format ( 1x, f5.2, i5, f5.1, t<np+56>, 1h*, t<m+npm>, 1h+ )
611     format ( 1x, f5.2, i5, f5.1 )
620     format ( 1x, f10.1, i10, f10.3 )
        if ( nn .ge. -1 ) write ( 6, 630 )
630     format ( ' FELL THRU 600 LOOP ' )
```

```
700     cost = rms
      go to 701
        if ( phasmr .ne. 0. )
     +   cost = rms / sin( phasmr * pi / 90. )
        if ( gainmr .lt. 6. .and. gainmr .gt. 0. )
     +   cost = cost / sin( gainmr / 6. * pi / 2. )
701   continue
        if ( nn .eq. 0 )
     +   write ( 6, 750 ) phasmr, gainmr, cost, slope
750     format ( / ' THE PHASE MARGIN IS ', f5.0, ' DEGREES ' /
     +   ' THE GAIN MARGIN IS ', f6.1, ' DB. ' /
     +   ' THE COST IS ', 3pf12.3 /
     +   ' THE SLOPE IS ', 0pf6.1, ' DB PER DECADE. ' / )
        ti = 1. / x( 1 )
        if ( nn .gt. 0 .and. cost .lt. 1. ) write ( 6, 800 )
     +   ti, x( 2 ), x( 3 ), rms, phasmr, gainmr, cost, slope
800     format ( f7.4, f7.2, f7.4, 3pf7.3, 0p2f7.3, 3pf7.3, 0pf7.2 )
      if ( ti .lt. .005 .and. cost .lt. costmn ) cost = - cost
      costmn = min( cost, costmn )
        fun = cost
900     return

        end

        subroutine machine

        implicit double precision ( a-h, o-z )
        common b, delta, deltat,
     +   jmax, jmaxt, jmod, kj, nc, nt, pi, tmax, urlin
        common / gp / gain( 10, 7 ), phase( 10, 7 )
        common / mark / mark( 8 )
        common / tgts / a, w
        common / x / x( 10 )
        byte cr
        data mark / '+', '*', 'x', 'o', 's', '#', '.', 'n' /
        cr = "015

        write ( 6, 5 )
5       format ( / ' NOW LOOK AT MACHINE ONLY ' / )
        n = -1
        cost = fun( x, n )
        write ( 6, 10 )
10      format ( / ' BODE PLOTS FOR SINUSOIDAL DRIVING FUNCTIONS ' / )
        do 200 k = 1, 7
          a = .1 / 2 ** ( k-1 )
          write ( 6, 20 ) a
20        format ( / 1x, 3pf6.2, ' MILLIRADIANS PER SECOND AMPLITUDE. ' )
          km = k
          do 100 i = 1, 10
            w = deltat * float( i )
            n = - nc * i
            cost = fun( x, n )
            if ( cost .gt. 1. ) go to 300
100       continue
```

```
          if ( unlin .eq. 0 ) go to 300
200     continue
300     write ( 6, 320 )
320     format ( // ' CONSOLIDATED PLOT ' //
     +   20x,  ' PHASE SHIFT ', 20x, ' DB GAIN '
     +   / '  FREQ  -225     -180     -135     -90 '
     +   ' -70        -60        -50       -40' /
     +   10x, 1h+, 3( 8x, 1h+ ), 3x, 1h+, 3(9x, 1h+ ) )
        do 400 i = 1, 10
          write ( 6, 330 ) i
330       format ( i5, $ )
          itpm = uwrite ( 1, cr, 1 )
          do 350 k = 1, km
            m = gaan( i, k )
            np = phaas( i, k ) / 5.
            if ( m .gt. -100 .and. np .ge. -54
     +       .and. m .lt. 0 .and. np .lt. 0 )
     +      write ( 6, 340 ) mark( k ), mark( k )
340         format ( t<np+56>, a1, t<m+112>, a1, $ )
            itpm = uwrite( 1, cr  1 )
350       continue
          write ( 6, 360 )
360       format ( 1x )
400     continue
        call plot( km )
        return

        end

        subroutine plot( n )

        implicit double precision ( a-h, o-z )
        common / gp / gaan( 10, 7 ), phaas( 10, 7 )
        common / mark / mark( 8 )
        dimension ixp( 10 ), iyp( 10 ), iyg( 10 )
        byte dev( 4 ), file( 10 )
        external ffaxis, ffline, ffneworigin, ffoutput
        data dev / 'v', 't', 'c', 0 /
        data file / 'd', 'a', 't', 'a', 'p', 'l', 'o', 't',
     +   2 * 0 /

        call callc( ffoutput, file, 0 )
        call callc( ffneworigin, 1000, 1000 )
        call callc( ffaxis, ' FREQUENCY IN RADIANS PER SECOND ',
     +   0, 0, 6000, 0.0, 0, 0., 1., 600 )
        call callc( ffaxis, ' PHASE IN RADIANS  == GAIN IN DB ',
     +   0, 0, -6000, 90., 0, -200., 20., 600 )
        jmx = 10
        do 100 i = 1, n
          do 90 j = 1, 10
            iyg( j ) = gaan( j, i ) * 30. + 6000.
            iyp( j ) = phaas( j, i ) * 30. + 6000.
            ixp( j ) = j * 600
90        continue
```

```
          call callc( ffline, ixp, iyp, 10, mark( i ), 1 )
          call callc( ffline, ixp, iyg, 10, mark( i ), 1 )
100     continue
        return

        end


      subroutine tgta0

      implicit double precision ( a-h, o-z )
      common b, delta, deltat,
     + jmax, jmaxt, jmod, k, nc, nt, pi, tmax, unlin
      common / tgta / dt, h, hdt, hlim, ood, pos, vdt

      dist0 = 4000.
      dt = deltat
      hlim = pi / 5.
      pos = 0.
      h = hlim
      vel = 10.
      hdot = vel / 50.
      hdt = hdot * deltat
      ood = 1. / dist0
      vdt = vel * deltat
      return
c
      end
c
      subroutine tgta1( j, dither, g )
c
      implicit double precision ( a-h, o-z )
      common / tgta / dt, h, hdt, hlim, ood, pos, vdt
c
      if ( abs( h ) .ge. hlim .and. h * hdt .gt. 0. )
     + hdt = - hdt
      h = h + hdt
      pos = pos + sin( h ) * vdt
      dtf = dt * float( j )
      angle = ood * pos
      dither = ood * ( .1 * sin( 1.5 * dtf )
     + + .1 * sin( 1. + 2.5 * dtf )
     + + .1 * ( sin( 2. + 3.5 * dtf )
     + + sin( 3. + 4.5 * dtf ) + sin( 4. + 5.5 * dtf )
     + + sin( 5. + 6.5 * dtf ) + sin( 6. + 7.5 * dtf ) ) )
      g = angle
      return

      end
c
        subroutine tgts( j, g )

        implicit double precision ( a-h, o-z )
        common / tgts / a, w
```

```
c
        g = a * sin( w * float( j ) )
        return
c
        end

      subroutine fnmin(n,x,fx,fun,e,eps,k)

      implicit double precision ( a-h, o-z )
      dimension x(10),eps(10),se(10),q(10),h(10,10),xi(10),xo(10)
      real*8 mj,lmda,l1,l2,l3, lmin, mjfot
c
        mjfot = 2.
c         reduced from 20. in BRL version to tame subroutine
      m=n
      do 1 i=1,m
      se(i)=eps(i)
      q(i)=se(i)*e
      xi(i)=x(i)
      xo(i)=xi(i)
      do 2 j=1,m
    2 h(i,j)=0.0
    1 h(i,i)=1.0
c                 ic is the iteration counter and jc is the
c                 function evaluation counter.
      ic=1
      jc=0
      ir3=5
      go to 112
    3 imax=20*m
      fmin=fbar
      f0=fbar
      fj=fbar
      del=0.0
      assign 30 to ir1
c                        begin iteration
   50 do 41 j=1,m
      qj=q(j)
      mj= mjfct *qj
      go to 100
   30 q(j)=max(se(j),abs(lmda))
      if(abs(del).gt.abs(fj-fbar))goto 41
      del=fj-fbar
      jd=j
   41 fj=fbar
c              check convergence
      if(ic.ge.imax)goto 91
      ir2=1
      kl=1
      psi1=0.0
      emin=200.
      do 63 i=1,m
      t2=abs(xi(i)-xo(i))
      if(t2.eq. 0.) go to 63
```

```
      if(t2.ge.eps(i)) ir2=2
      psil=psil+t2*t2
      t3=eps(i)/t2
      if(t3.lt.emin) emin=t3
   63 continue
      go to (90,70),ir2
c                            check desirability of new direction
   70 do 73 i=1,m
   73 x(i)=xi(i)+xi(i)-xo(i)
      ir3=6
      go to 112
   75 f1=f0
      psil=sqrt(psil)
      emin=emin*psil
      l1=-psil
      f2=fmin
      l2=0.0
      f3=fbar
      l3=psil
      if(f3.ge.f1)goto 72
      if((f1-(f2+f2)+f3)*(f1-f2-del)**2.ge..5*del*(f1-f3)**2)goto 72
c        compute new direction and use directions
c          i=1,2,3,...,jd-1,jd+1,...,n,new
      jj=m-1
      if(jd-m)81,83,81
   81 do 82 i=jd,jj
      se(i)=se(i+1)
      q(i)=q(i+1)
      do 82 j1=1,m
   82 h(i,j1)=h(i+1,j1)
   83 do 84 j1=1,m
   84 h(m,j1)=(xi(j1)-xo(j1))/psil
      se(m)=emin
      q(m)=psil
      qj=psil
        mj = mjfct * psil
      j=m
      assign 72 to ir1
      go to 400
c                    prepare for new iteration
   72 do 71 i=1,m
   71 xo(i)=xi(i)
      f0=fmin
      fj=fmin
      del=0
      ic=ic+1
      assign 30 to ir1
      go to 50
c                    prepare to return
   91 k1=2
   90 do 92 i=1,m
   92 x(i)=xi(i)
      fx=fmin
      if(k)93,96,93
```

```
   93 k=k1
   97 return
   96 if(k1-1)94,97,94
   94 write(6,95) imax,k1
   95 format(24h funmin--not converged--,i3,15h iterations,k =,i2)
      stop
c     find minimum along a line (initial steps)
  100 l2=0
      f2=fmin
      lmda=qj
      ir3=1
      go to 110
  102 if(fbar.gt.f2)goto 103
      l1=l2
      f1=f2
      l2=lmda
      f2=fbar
      lmda=qj+qj
      ir3=2
      go to 110
  105 l3=lmda
      f3=fbar
      go to 400
  103 l3=lmda
      f3=fbar
      lmda=-qj
      ir3=3
      go to 110
  104 l1=lmda
      f1=fbar
c     find minimum along a line
  400 t1=l2-l3
      t2=l3-l1
      t3=l1-l2
      t4=t1*t2*t3
      t5=t1*f1+t2*f2+t3*f3
      t4=t5/t4
      t1=l1*l1
      t2=l2*l2
      t3=l3*l3
      lmda=.5*((t2-t3)*f1+(t3-t1)*f2+(t1-t2)*f3)/t5
      if(t4)401,402,402
 401  if(abs(lmda)-mj)403,403,402
  402 if(f1.lt.f3)goto 404
      lmda=mj
      go to 403
  404 lmda=-mj
  403 if(f1.lt.f2)goto 405
      if(f3.lt.f2)goto 406
      lmin=l2
      fmin=f2
  407 if(abs(lmda-lmin).lt.se(j)) go to 471
      if(lmda.eq.0.0)goto 408
      if(abs((lmda-lmin)/lmda).lt..03)goto 471
```

```
  408 ir3=4
      go to 110
  405 lmin=l1
      fmin=f1
      go to 407
  406 lmin=l3
      fmin=f3
      go to 407
  480 if(lmda.gt.l2)goto 481
      if(lmda.lt.l1)goto 482
      if(fbar.lt.f2)goto 483
  486 l1=lmda
      f1=fbar
      go to 400
  481 if(lmda.gt.l3)goto 484
      if(fbar.lt.f2)goto 485
  487 l3=lmda
      f3=fbar
      go to 400
  482 l3=l2
      f3=f2
      l2=l1
      f2=f1
      go to 486
  483 l3=l2
      f3=f2
  488 l2=lmda
      f2=fbar
      go to 400
  484 l1=l2
      f1=f2
      l2=l3
      f2=f3
      go to 487
  485 l1=l2
      f1=f2
      go to 488
  471 lmda=lmin
      fbar=fmin
      do 473 i=1,m
  473 xi(i)=xi(i)+lmda*h(j,i)
      go to ir1, (30,72)
c                           prepare to evaluate function
  110 do 111 i=1,m
  111 x(i)=xi(i)+lmda*h(j,i)
  112 jc=jc+1
      fbar=fun(x,m)
c  special for control  c  c  c  c  c  c  c  c
      if ( fbar .le. 0. ) go to 91
c  special for control  c  c  c  c  c  c  c  c
      go to (102,105,104,480,3,75        ),ir3

      end
```

Jan 2 10:41

```
      subroutine fft( fr, fi, k  )

      implicit double precision ( a-h, o-z )
      dimension fr( 1024 ), fi( 1024 )

      n = 2 ** k
      mr = 0
      nn = n - 1
      do 2 m = 1, nn
         l = n
1        l = l / 2
         if ( mr + l .gt. nn ) go to 1
           mr = mod( mr, l ) + l
           if ( mr .le. m ) go to 2
             tr = fr( m + 1 )
             fr( m + 1 ) = fr( mr + 1 )
             fr( mr + 1 ) = tr
             ti = fi( m + 1 )
             fi( m + 1 ) = fi(mr + 1 )
             fi( mr + 1 ) = ti
2         continue
          l = 1
3         if ( l .ge. n ) return
            istep = 2 * l
            el = l
            do 4 m = 1, l
              a = 3.1415926535 * float( 1 - m ) / el
              wr = cos( a )
              wi = sin( a )
              do 4 i = m, n, istep
                j = i + l
                tr = wr * fr( j ) - wi * fi( j )
                ti = wr * fi( j ) + wi * fr( j )
                fr( j ) = fr( i ) - tr
                fi( j ) = fi( i ) - ti
                fr( i ) = fr( i ) + tr
                fi( i ) = fi( i ) + ti
4             continue
              l = istep
              go to 3

        end

        subroutine man0( delta, nt, x )
c
        implicit double precision ( a-h, o-z )
        real*8 kk
        common / man / g1( 320 ), g2, g3, kk, expt, tlti, tltme
        dimension x( 10 )
c
        ti = 1. / x( 1 )
        kk = x( 2 )
        tl = x( 3 )
        tti = delta / ti
```

```
      expt = exp( -tti )
      t1ti = t1 / ti
      t1tme = 1. - t1ti - expt
      do 50 n = 1, nt
50      g1( n ) = 0.
      g2 = 0.
      g3 = 0.
      call tgta0
      return
c
      end
c
      subroutine man1( h, j, nt, diffdt, difft, gm )
c
      implicit double precision ( a-h, o-z )
      real*8 kk
      common / man / g1( 320 ), g2, g3, kk, expt, t1ti, t1tme
c
      do 60 n = 1, nt-1
60      g1( n ) = g1( n+1 )
      call tgta1( j, dither, g )
      difft = g - h
      diffdt = g + dither - h
      g1( nt ) = diffdt
      g21 = g2
      g2 = kk * g1( 1 )
      g3 = expt * g3 + t1tme * g21 + t1ti * g2
      gm = g3
      return
c
      end

      subroutine plant0( b, kb, bk, bki, cf, ds, dt )
c
      implicit double precision ( a-h, o-z )
      real*8 jj, kb
      common / plant / expb, bdxpb, bklsh, bklshi, cfdt, dedspc,
     +  delta, g3b, g4, hn, hnb1
c
      jj = 1.
      delta = dt
      tj = delta / jj
      expb = exp( -b * kb * tj )
      bdxpb = ( 1. - expb ) / b
      bklsh = bk
      bklshi = bki
      cfdt = cf * tj
      dedspc = ds
      return
c
      end
c
      subroutine plant1( h )
c
```

```
      implicit double precision ( a-h, o-z )
      common / plant / expb, bdxpb, bklsh, bklshi, cfdt, dedspc,
     +  delta, g3b, g4, hn, hnbl
c
      g3b = 0.
      g4 = 0.
      h = 0.
      hn = 0.
      hnbl = 0.
      return
c
      end
c
      subroutine plant2( g3, h )
c
      implicit double precision ( a-h, o-z )
      common / plant / expb, bdxpb, bklsh, bklshi, cfdt, dedspc,
     +  delta, g3b, g4, hn, hnbl
c
      g3bl = g3b
      g3b = 0.
      if ( abs( g3 ) .gt. dedspc )
     +  g3b = g3 - sign( dedspc, g3 )
c     if ( abs( g3b - g3bl ) .gt. bklshi )
c    +   g3b = g3b - sign( bklshi, g3b - g3bl )
      g4 = expb * g4 + bdxpb * g3bl
      g4 = g4 - sign( min( cfdt, abs( g4 ) ), g4 )
      hnbl = hnbl + delta * g4
      if ( abs( hnbl - hn ) .gt. bklsh )
     +  hn = hnbl - sign( bklsh, hnbl - hn )
      h = hn
      return
c
      end

      subroutine stats0( dt, jmax )
c
      implicit double precision ( a-h, o-z )
      common / stats / delta, points, tmax,
     +  sum, sumsq, sumin, sumout, sumi2, sumo2,
     +  sumt, sumt2, sumit, sumot
c
      delta = dt
      points = jmax
      tmax = points * delta
      sum = 0.
      sumsq = 0.
      sumin = 0.
      sumout = 0.
      sumi2 = 0.
      sum02 = 0.
      sumt = 0.
      sumt2 = 0.
      sumit = 0.
```

```
      sumot = 0.
      return
c
      end
c
      subroutine stats1 ( difft, frin, frout, j )
c
      implicit double precision ( a-h, o-z )
      common / stats / delta, points, tmax,
     +  sum, sumsq, sumin, sumout, sumi2, sumo2,
     +  sumt, sumt2, sumit, sumot
c
      dt = difft
      sum = sum + dt * delta
      sumsq = sumsq + dt * dt * delta
      sumin = sumin + frin
      sumout = sumout + frout
      sumi2 = sumi2 + frin * frin
      sumo2 = sumo2 + frout * frout
      tj = j
      sumt = sumt + tj
      sumt2 = sumt2 + tj * tj
      sumit = sumit + frin * tj
      sumot = sumot + frout * tj
      return
c
      end
c
      subroutine stats2 ( avin, avout, rmsd )
c
      implicit double precision ( a-h, o-z )
      common / stats / delta, points, tmax,
     +  sum, sumsq, sumin, sumout, sumi2, sumo2,
     +  sumt, sumt2, sumit, sumot
      common / statw / rms, avein, aveout,
     +  devin, devout, devid, devod, ai, bi, ao, bo
c
      rms = sqrt( sumsq / tmax )
      rmsd = rms
      avein = sumin / points
      avin = avein
      aveout = sumout / points
      avout = aveout
      si2 = sumi2 - sumin * avein
      so2 = sumo2 - sumout * aveout
      pts1 = points - 1.
      devin = sqrt( si2 / pts1 )
      devout = sqrt( so2 / pts1 )
      avet = sumt / points
      st2 = sumt2 - sumt * avet
      sit = sumit - sumin * avet
      sot = sumot - sumout * avet
      bi = sit / st2
      bo = sot / st2
```

```
        ai = avein - bi * avet
        ao = aveout - bo * avet
        pts2 = points - 2
        devid = sqrt ( ( si2 - bi * sit ) / pts2 )
        devod = sqrt ( ( so2 - bo * sot ) / pts2 )
        return
c
        end

        subroutine statsw
c
        implicit double precision ( a-h, o-z )
        common / statw / rms, avein, aveout, devin, devout,
     +  devid, devod, ai, bi, ao, bo
c
        write ( 6, 100 ) rms, avein, aveout, devin, devout,
     +  devid, devod, ai, bi, ao, bo
100     format ( '  RMS AVEIN AVEOUT DEVIN DEVOUT '
     +  'DEVID DEVOD  AI    BI     AO     BO '
     +  / 3p11f6.3 / )
        return
c
        end
```

DISTRIBUTION LIST

No. of Copies	Organization
12	Commander Defense Technical Info Center ATTN: DDC-DDA Cameron Station Alexandria, VA 22314
1	Director Defense Advanced Research Projects Agency Tactical Technical Office ATTN: Dr. James Tegnelia 1400 Wilson Boulevard Arlington, VA 22209
1	Director Institute for Defense Analyses ATTN: Dr. Bruce J. Whittemore 400 Army Navy Drive Arlington, VA 22202
1	Deputy Secretary of Defense R&E Tactical Warfare Programs ATTN: Mr. David Hardison Washington, DC 20310
1	Commander US Army Materiel Development and Readiness Command ATTN: DRCDMD-ST 5001 Eisenhower Avenue Alexandria, VA 22333
2	Commander US Army Armament Research and Development Command ATTN: DRDAR-TSS Dover, NJ 07801
5	Commander US Army Armament Research and Development Command ATTN: DRDAR-AC DRDAR-SC DRDAR-SCS DRDAR-SCF DRDAR-SE Dover, NJ 07801
8	Commander US Army Armament Research and Development Command ATTN: DRDAR-LC, Dr. J. Frasier DRDAR-LCW, Mr. H. Garver DRDAR-LCN, Mr. F. Saxe DRDAR-LCU, Mr. A. Moss DRDAR-LCS, Mr. Gregorits Mr. Killen, Mr. Brooks DRDAR-LCS, Mr. Randers-Pehrson Dover, NJ 07801
1	Commander US Army Armament Research and Development Command ATTN: DRDAR-AS Dover, NJ 07801
2	Director US Army ARRADCOM Benet Weapons Laboratory ATTN: DRDAR-LCB-TL DRDAR-LCB, Mr. M. Dale Watervliet, NY 12189
1	Commander US Army Armament Materiel Readiness Command ATTN: DRSAR-LEP-L, Tech Lib Rock Island, IL 61202
1	Commander US Army Aviation Research and Development Command ATTN: DRSAV-E P.O. Box 209 St. Louis, MO 63166
1	Director US Army Air Mobility Research and Development Laboratory Ames Research Center Moffett Field, CA 94035

DISTRIBUTION LIST

No. of Copies	Organization
1	Director Applied Technology Laboratory US Army Research & Technology Laboratories ATTN: DAVDL Fort Eustis, VA 23604
1	Commander US Army Communications Rsch and Development Command ATTN: DRDCO-PPA-SA Fort Monmouth, NJ 07703
1	Commander US Army Electronics Research and Development Command Technical Support Activity ATTN: DELSD-L Fort Monmouth, NJ 07703
4	Commander US Army Harry Diamond Labs ATTN: DELHD-TI DELHD-SA, Mr. W. Pepper Mr. J. Salerno DELHD-D-OE-ES, Mr. Danner 2800 Powder Mill Road Adelphi, MD 20783
1	Director Office of Missile Electronic Warfare ATTN: DELEW-WLH-SF, Dr. R. Clawson White Sands Missile Range, NM 88002
1	Commander US Army Night Vision and Electro-Optics Laboratory ATTN: DELND-NV-VI, Mr. J. Dehne Fort Belvoir, VA 22060
2	Commander US Army Missile Command ATTN: DRSMI-R DRSMI-YDL Redstone Arsenal, AL 35809
3	Commander US Army Missile Command ATTN: DRSMI-CA, Dr. D. McDaniels DRSMI-EA, Mr. B. Harwell DRSMI-OT, Mr. K. Taylor Redstone Arsenal, AL 35809
3	Commander US Army Missile Command ATTN: COL Williamson, TOW PM Redstone Arsenal, AL 35809
1	Commander US Army Tank Automotive Research & Development Command ATTN: DRDTA-UL Warren, MI 48090
1	Commander US Army Training and Doctrine Command ATTN: ATCG, Dr. Marvin Pastel Fort Monroe, VA 23651
2	Director US Army TRADOC Systems Analysis Activity ATTN: ATAA-SL, Tech Lib ATAA-TF, Mr. B. Watson White Sands Missile Range, NM 88002
1	Commander US Army Armor Center and School ATTN: ATZK-CD-SD, Mr. G. Hilkemeyer Fort Knox, KY 40121
3	Commander US Army Infantry Center ATTN: ATSH-DCG, Mr. P. Ferguson Mr. G. Hardgrove ATSH-CD-SD-F, Mr. Ramsey Fort Benning, GA 31905

DISTRIBUTION LIST

No. of Copies	Organization	No. of Copies	Organization
2	ADTC/DLMIT (J. Constantine; J. Burda) Eglin AFB, FL 32542	1	AFELM, The Rand Corporation ATTN: Library-D 1700 Main Street Santa Monica, CA 90406
2	AFATL (Dr. J.R. Mayersak, Chief Scientist) Eglin AFB, FL 32542	1	AVCO Systems Division ATTN: Mr. Pete Kenyon 201 Lowell Street Wilmington, MA 01887

Aberdeen Proving Ground

Dir, USAMSAA
ATTN: DRXSY-D
DRXSY-MP, Mr. H. Cohen
DRXSY-GI, MAJ G. DeGelo
Mr. D. Kirk
DRXSY-DS, Mr. J. Kramar
DRXSY-T, Mr. A. Reid
DRXSY-G, Mr. R. Conroy
Mr. C. Odom
Mr. W. Clifford
DRXSY-GA, J. Brewer
Mr. J. Chernick
Dir, HEL
ATTN: Mr. D. Giordano
Mr. G. L. Horley
CDR, USATECOM
ATTN: DRSTE-TO-F
Dir, USA CSL
Bldg E3516, EA
ATTN: DRDAR-CLB-PA

USER EVALUATION OF REPORT

Please take a few minutes to answer the questions below; tear out this sheet, fold as indicated, staple or tape closed, and place in the mail. Your comments will provide us with information for improving future reports.

1. BRL Report Number ________________

2. Does this report satisfy a need? (Comment on purpose, related project, or other area of interest for which report will be used.)

____

____

____

3. How, specifically, is the report being used? (Information source, design data or procedure, management procedure, source of ideas, etc.) ________________

____

____

4. Has the information in this report led to any quantitative savings as far as man-hours/contract dollars saved, operating costs avoided, efficiencies achieved, etc.? If so, please elaborate.

____

____

5. General Comments (Indicate what you think should be changed to make this report and future reports of this type more responsive to your needs, more usable, improve readability, etc.) ________________

____

____

____

6. If you would like to be contacted by the personnel who prepared this report to raise specific questions or discuss the topic, please fill in the following information.

Name: ________________

Telephone Number: ________________

Organization Address: ________________

____

____

FOLD HERE

Director
US Army Ballistic Research Laboratory
Aberdeen Proving Ground, MD 21005

OFFICIAL BUSINESS
PENALTY FOR PRIVATE USE, $300

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO 12062 WASHINGTON, DC

POSTAGE WILL BE PAID BY DEPARTMENT OF THE ARMY

Director
US Army Ballistic Research Laboratory
ATTN: DRDAR-TSB
Aberdeen Proving Ground, MD 21005

FOLD HERE