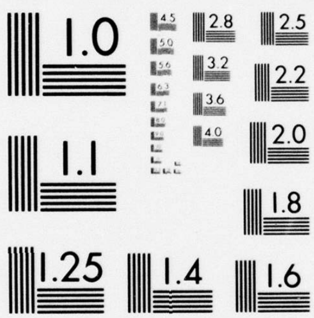
AD-A064 760 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OHIO SCH--ETC F/G 15/3.1
A PRACTICAL THREE DIMENSIONAL, 11 STATE EXTENDED KALMAN FILTER --ETC(U)
DEC 78 C W HLAVATY

UNCLASSIFIED AFIT/GGC/EE/78-7-VOL-2 NL

1 OF 3
AD
A064760


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
6.3

AFIT/GGC/EE/78-7

LEVEL III

1

ADA064760

DDC FILE COPY

6

A PRACTICAL THREE DIMENSIONAL, 11

STATE EXTENDED KALMAN FILTER FOR

USE IN A FIRE CONTROL SYSTEM

AGAINST NON-THRUSTING MISSILES.

9 Master's THESIS VOLUME II.

14 AFIT/GGC/EE/78-7

10 Charles W. Hlavaty
Capt USAF

DDC
RECEIVED
FEB 22 1979
16 A

11 Dec 78

12 214p.

Approved for public release; distribution unlimited

79 01 30 169

012 225

bpg

AFIT/GGC/EE/78-7

# A PRACTICAL THREE DIMENSIONAL, 11 STATE EXTENDED KALMAN FILTER FOR USE IN A FIRE CONTROL SYSTEM AGAINST NON-THRUSTING MISSILES

THESIS VOLUME II

Presented to the Faculty of the School of Engineering

of the Air Force Institute of Technology

Air Training Command

in Partial Fulfillment of the

Requirements for the Degree of

Master of Science

by

Charles W. Hlavaty, B.S.E.E., M.S.

Capt USAF

Graduate Electrical Engineering

December 1978

Volume II

Contents

## Appendix C

The appendix contains the graphical results of the Monte Carlo analyses of the Double Filter. In each test, the filter states are initialized to the initial true values. The tuning parameters used in the filter are (Note: The state vector is

$$[V_{mx}, \theta_t, R, \dot{R}, A_1, n, \tau_f, m/s]^T :$$

$$R = \begin{bmatrix} 3.E-5 & 0 & 0 \\ 0 & 500 & 0 \\ 0 & 0 & 100 \end{bmatrix}$$

where the measurements are on $\theta_t$, R, and $\dot{R}$. For angle measurements only,

$$R = 3.E-5$$

$$P_o = \begin{bmatrix} 100 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.E-8 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 101 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & .4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & .009 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 5 \end{bmatrix}$$

$$
Q = \begin{bmatrix}
250 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1.E-6 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 500 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 200 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 10 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & .5 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 101 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & .009
\end{bmatrix}
$$

The results of the trajectories tested are arranged as follows:

| Figure | Subject |
|---|---|
| | Double Filter with four measurements |
| C-1 to C-18 | K Set One |
| C-19 to C-36 | K Set Four, Three Dimensional |
| C-37 to C-54 | K Set Four, Two Dimensional |
| | Double Filter with two measurements |
| C-55 to C-72 | K Set Four |

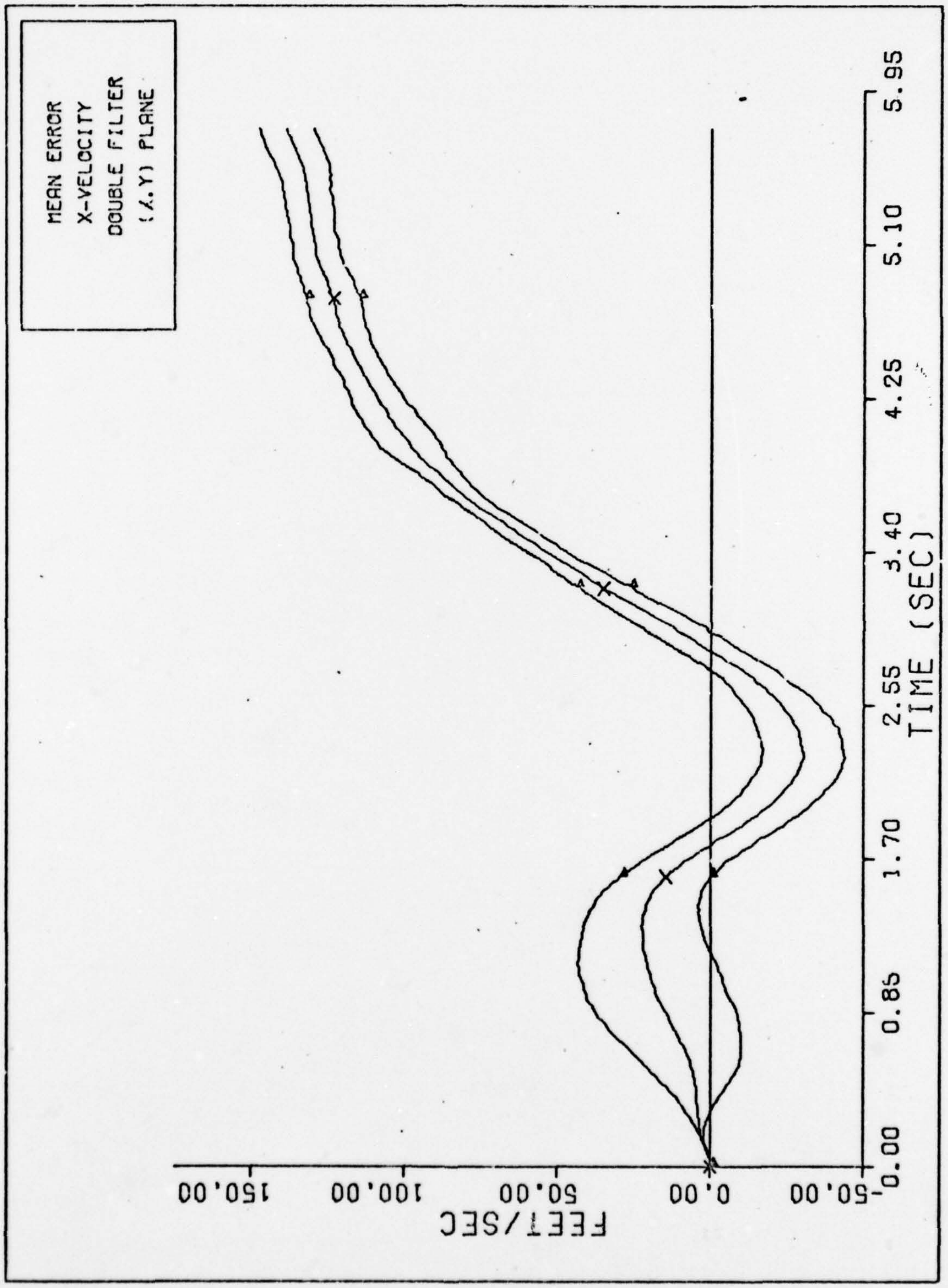



Fig. C-1 X-VELOCITY DOUBLE FILTER

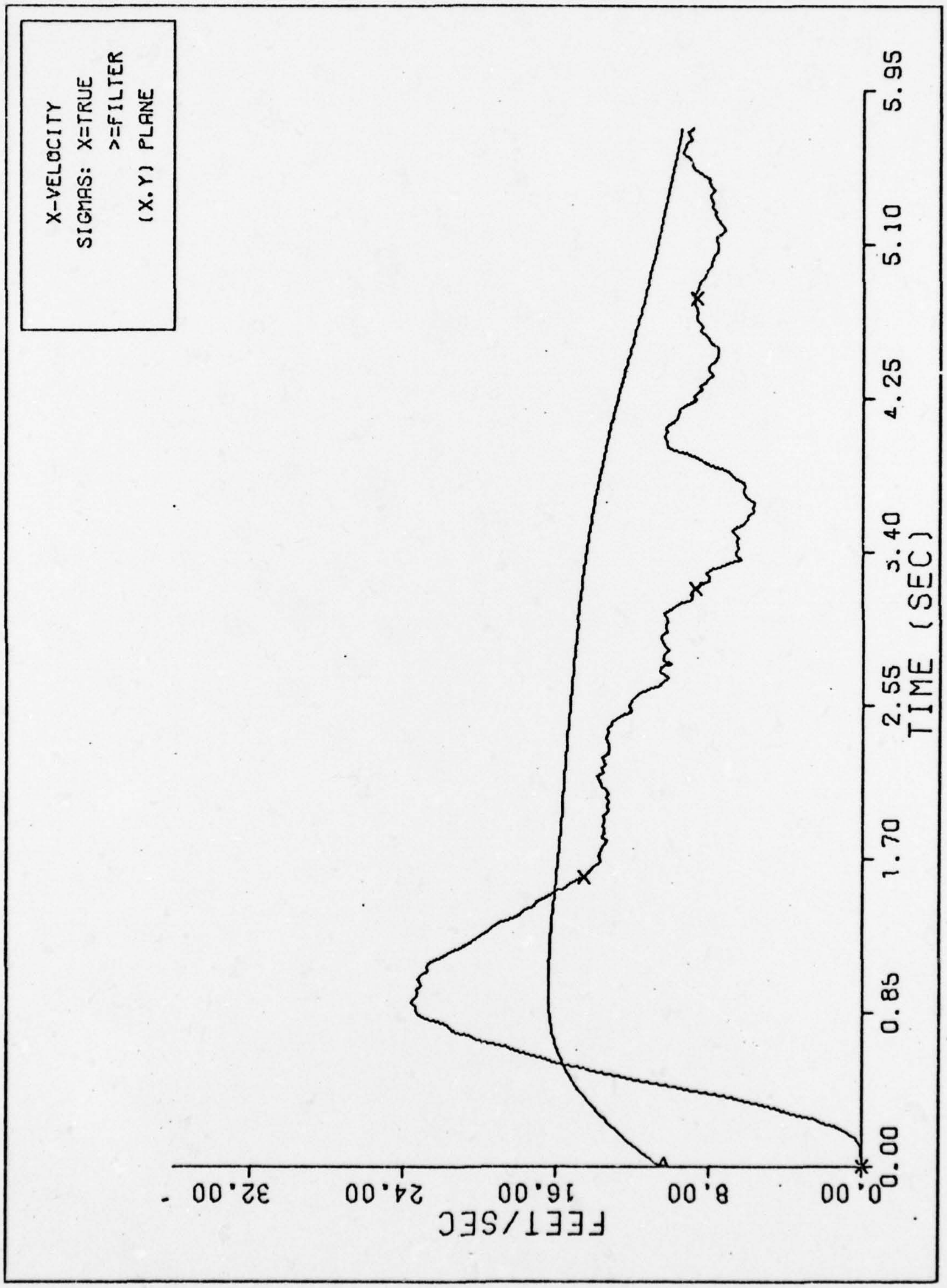



Fig. C-2 X-VELOCITY SIGMAS DOUBLE FILTER

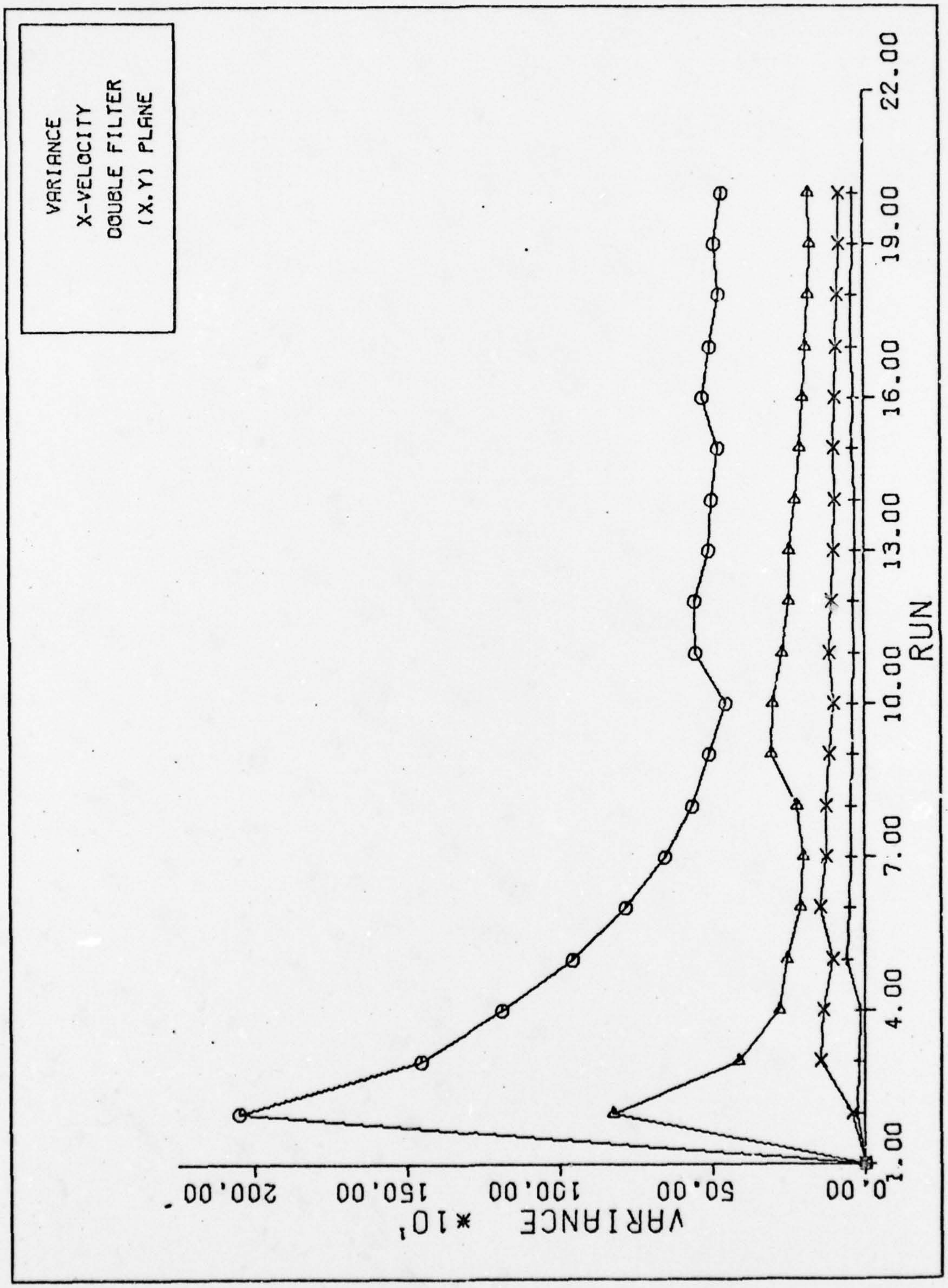



Fig. C-3 VARIANCE CONVERGENCE

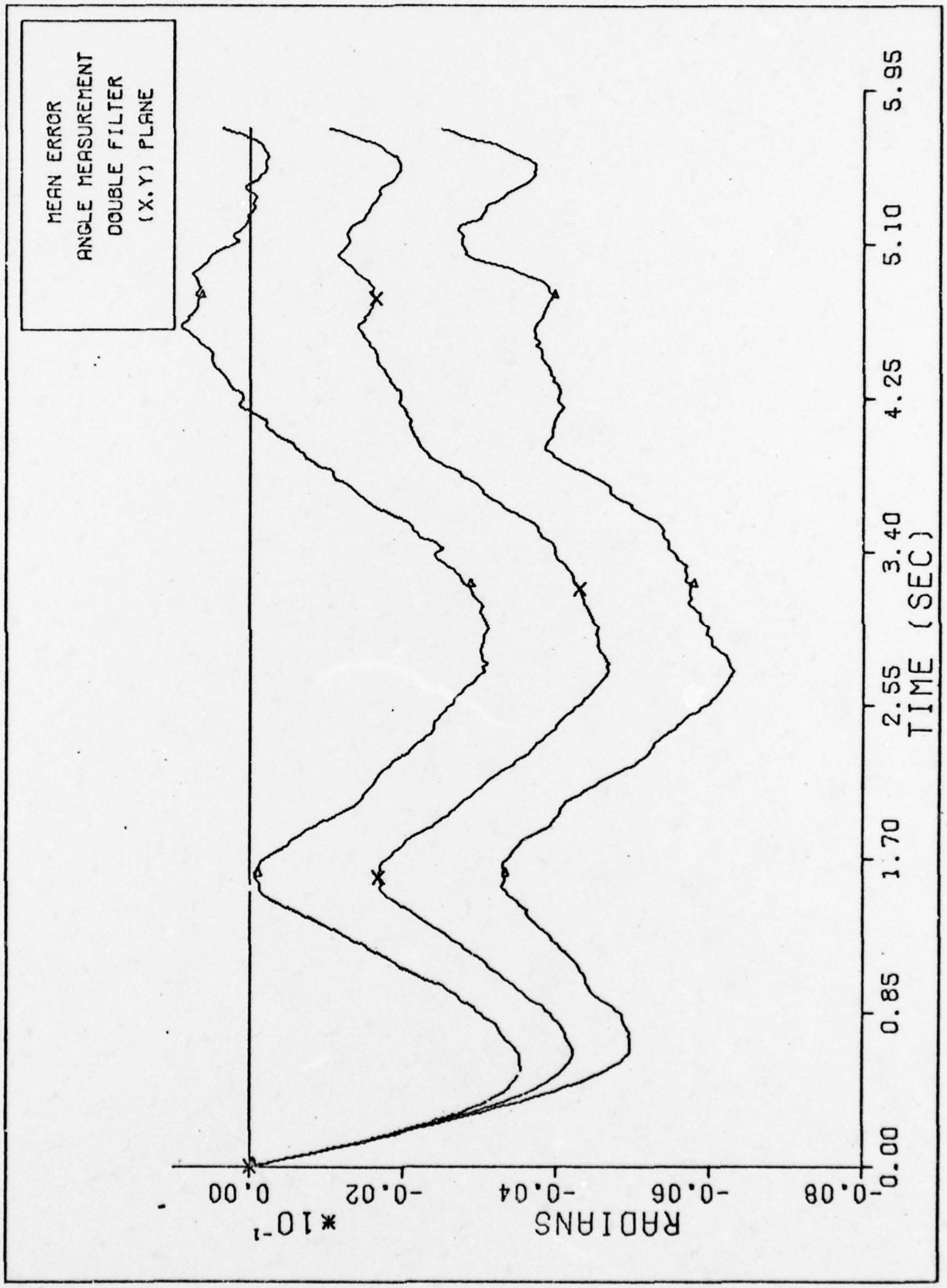



Fig. C-4 ANGLE MEASUREMENT DOUBLE FILTER

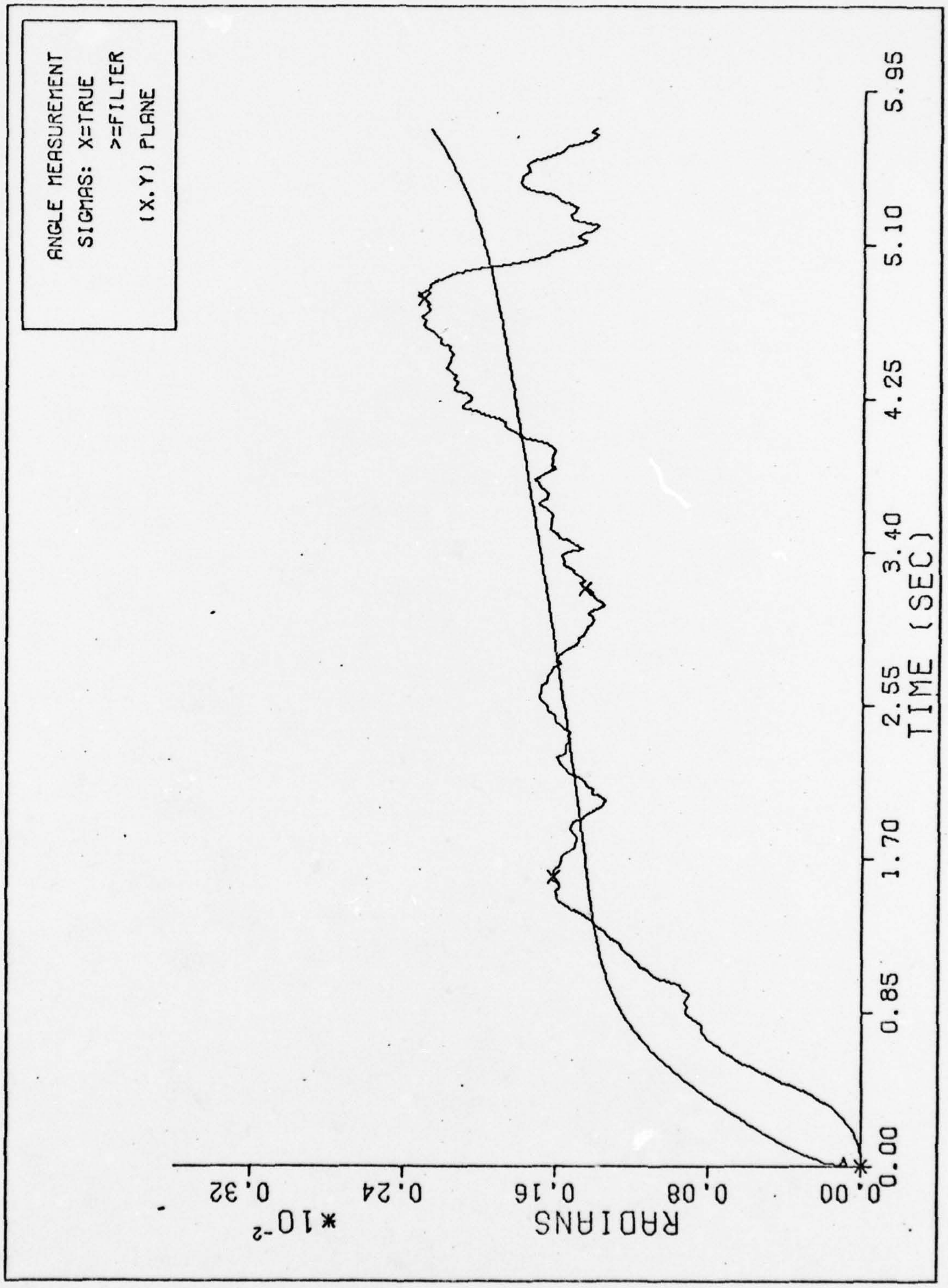



Fig. C-5 ANGLE MEASUREMENT SIGMAS DOUBLE FILTER

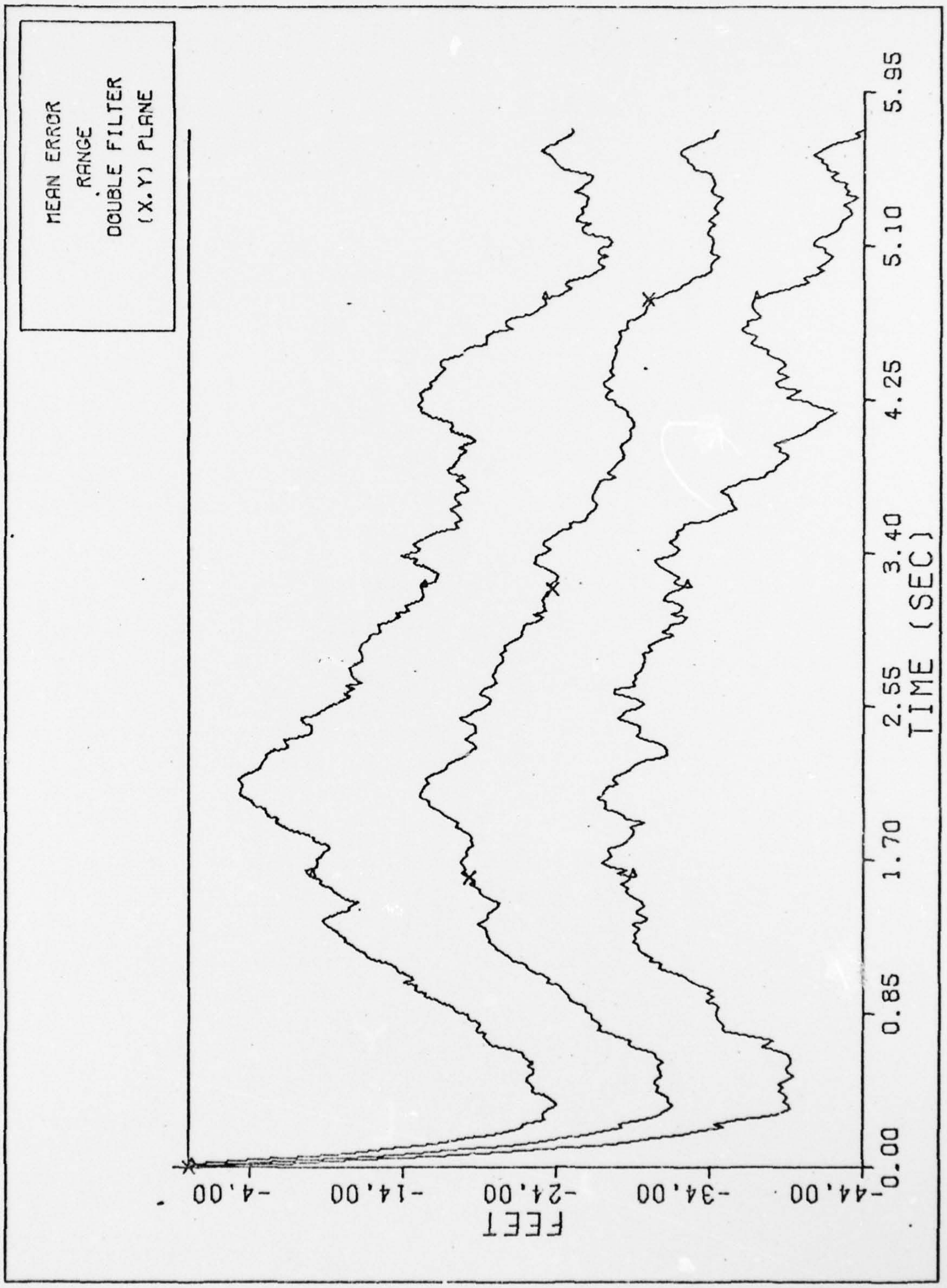



Fig. C-6 RANGE DOUBLE FILTER

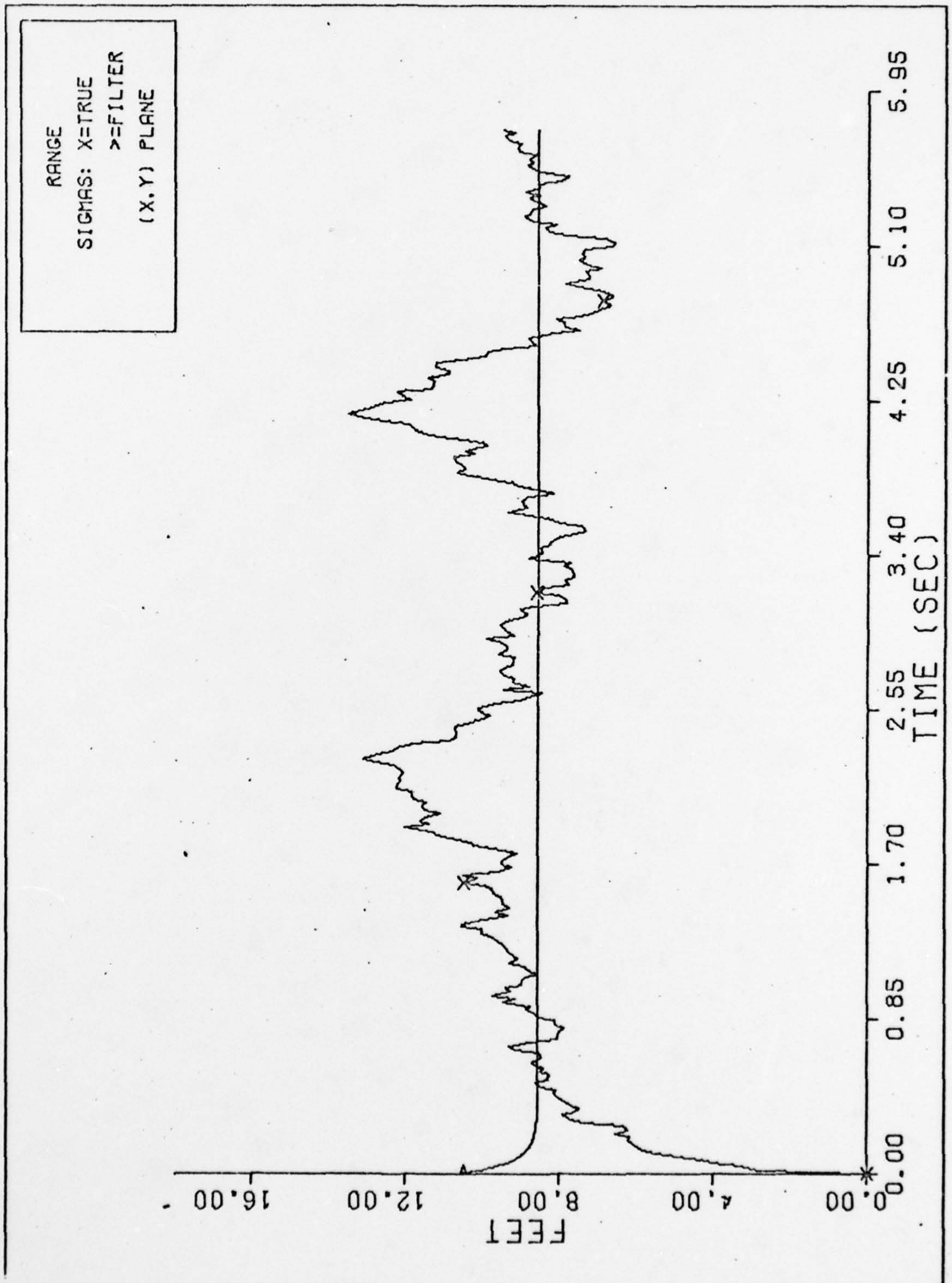



Fig. C-7 RANGE SIGMAS DOUBLE FILTER

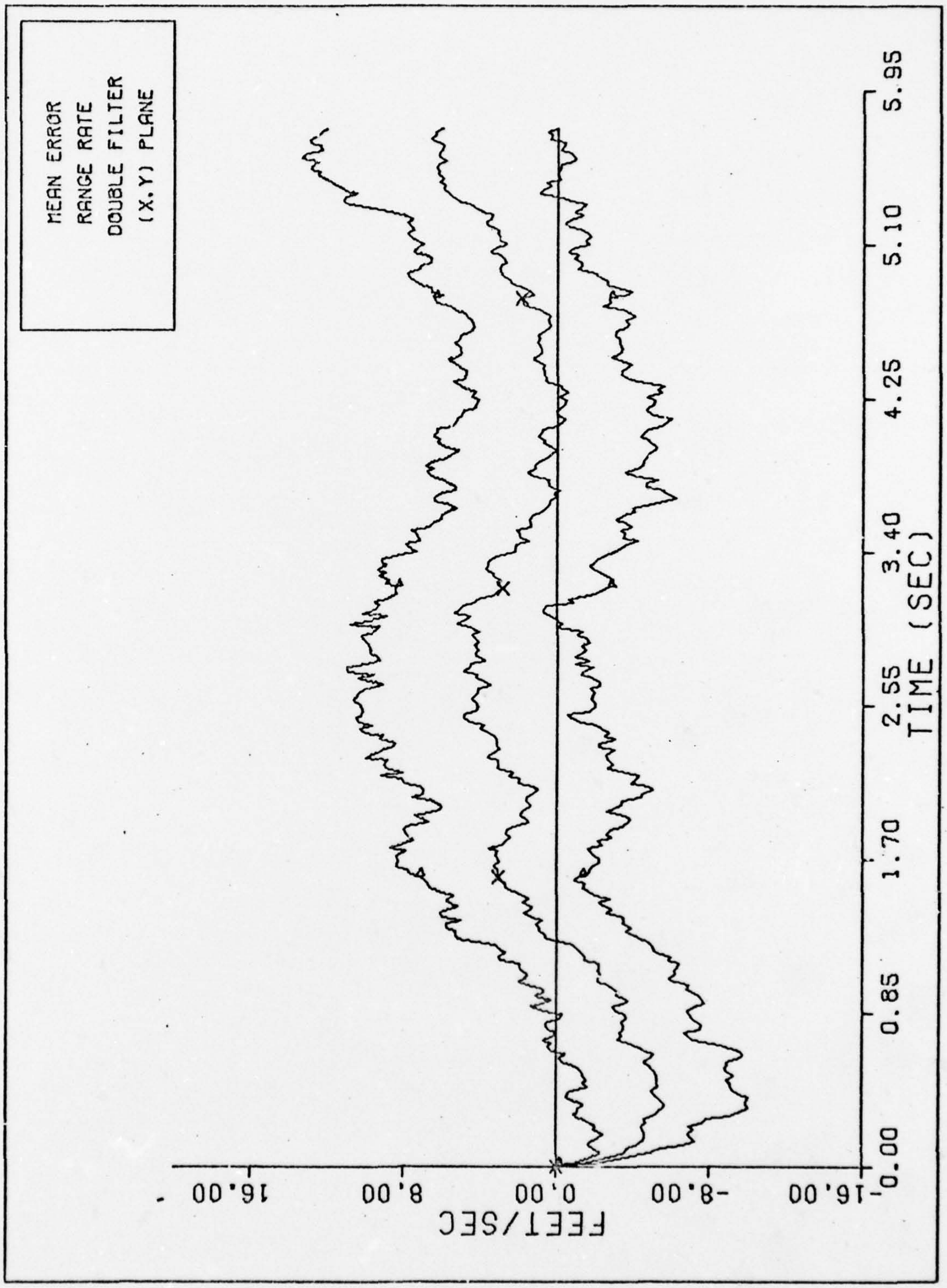



Fig. C-8 RANGE RATE DOUBLE FILTER

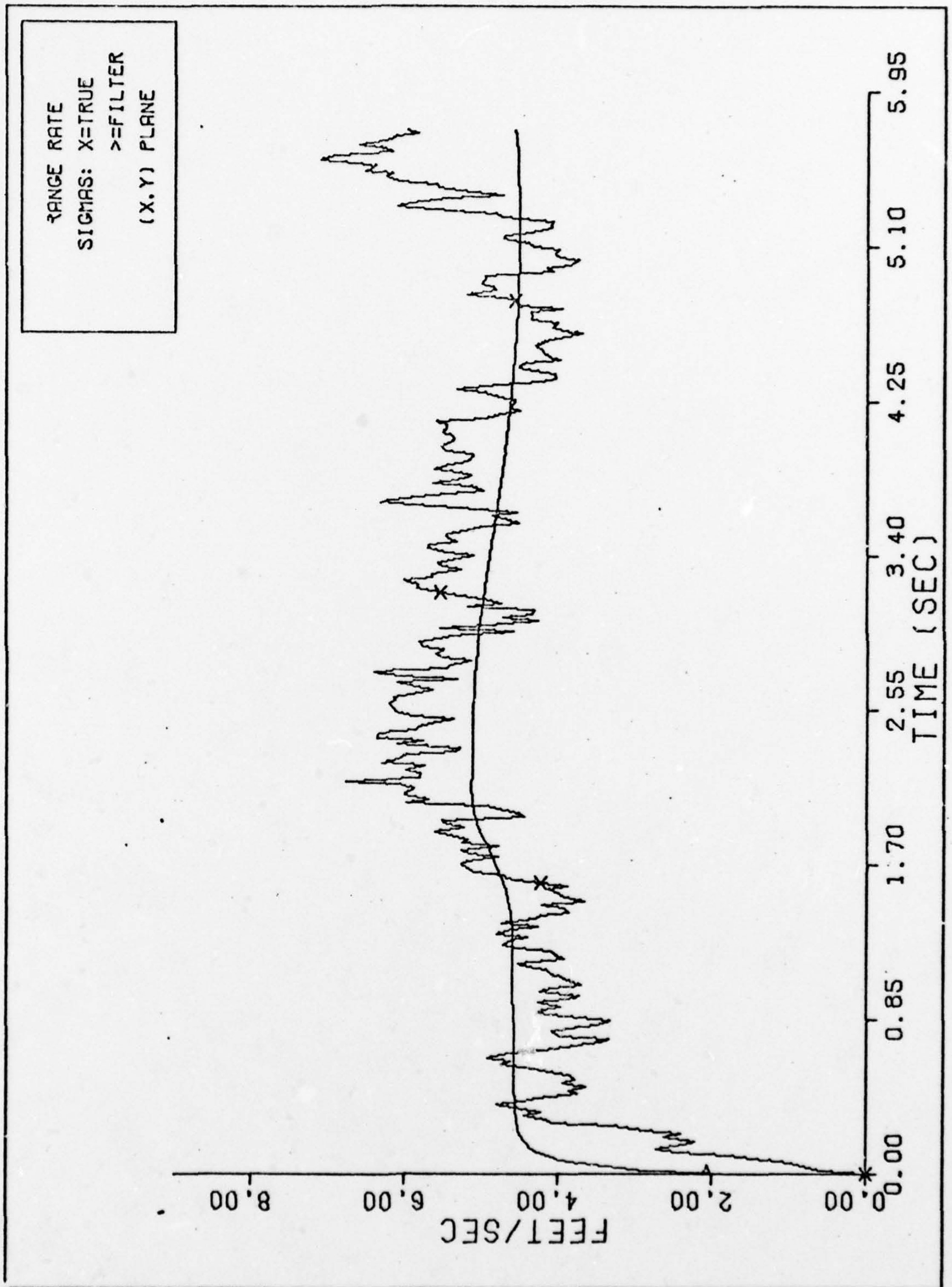



Fig. C-9 RANGE RATE SIGMAS DOUBLE FILTER

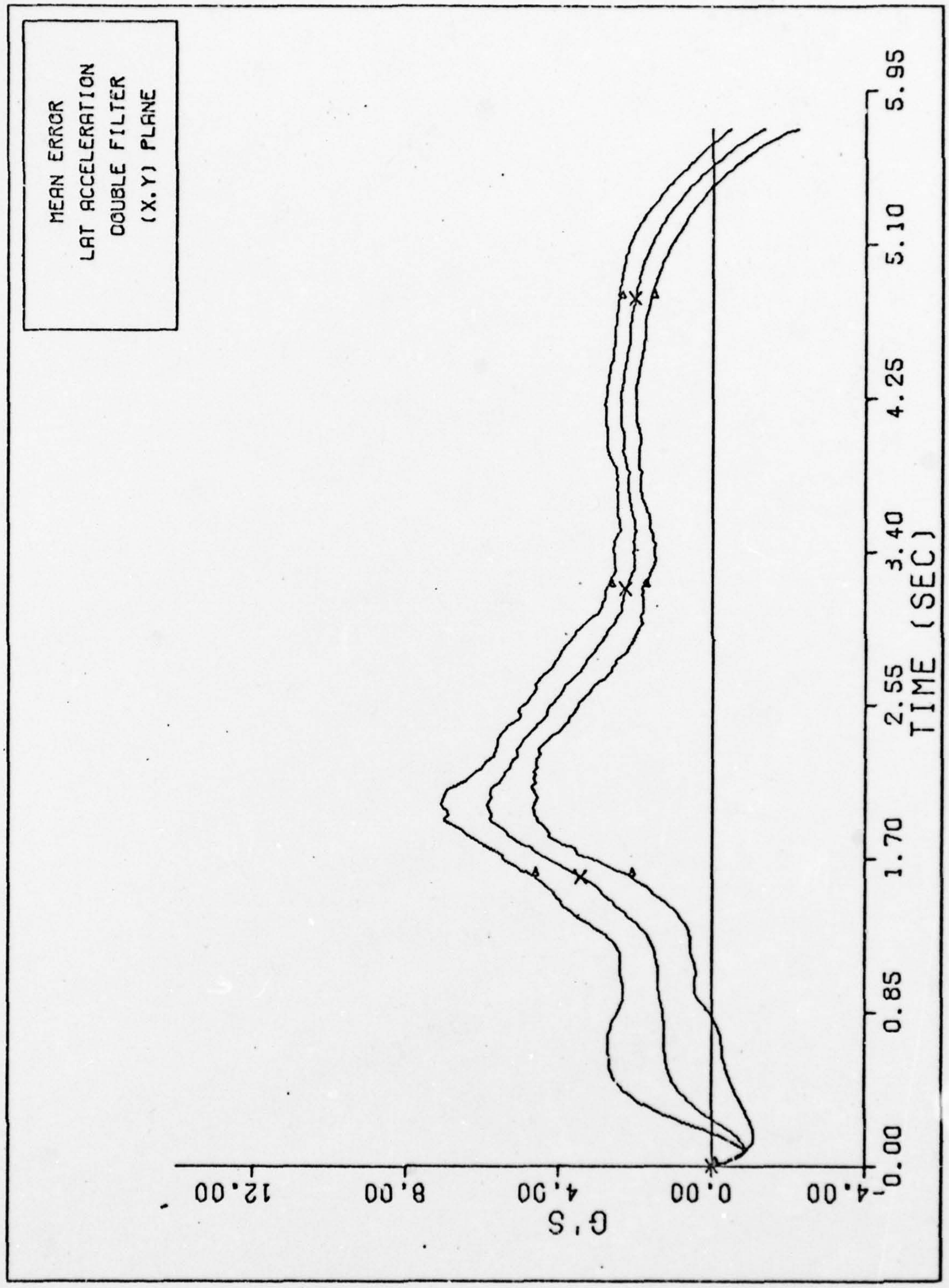



Fig. C-10 LAT ACCELERATION DOUBLE FILTER

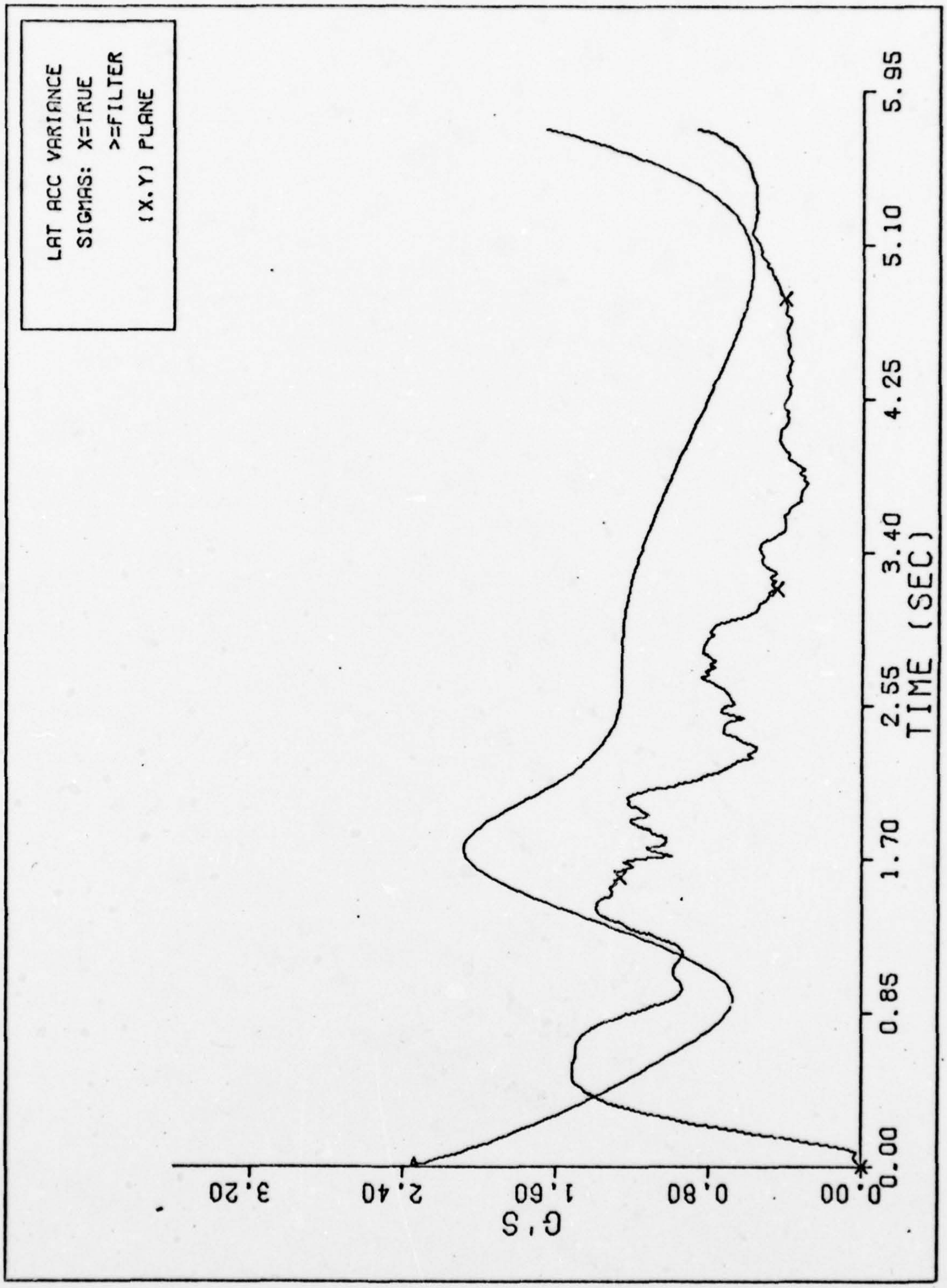



Fig. C-11 LAT ACCELERATION SIGMAS DOUBLE FILTER

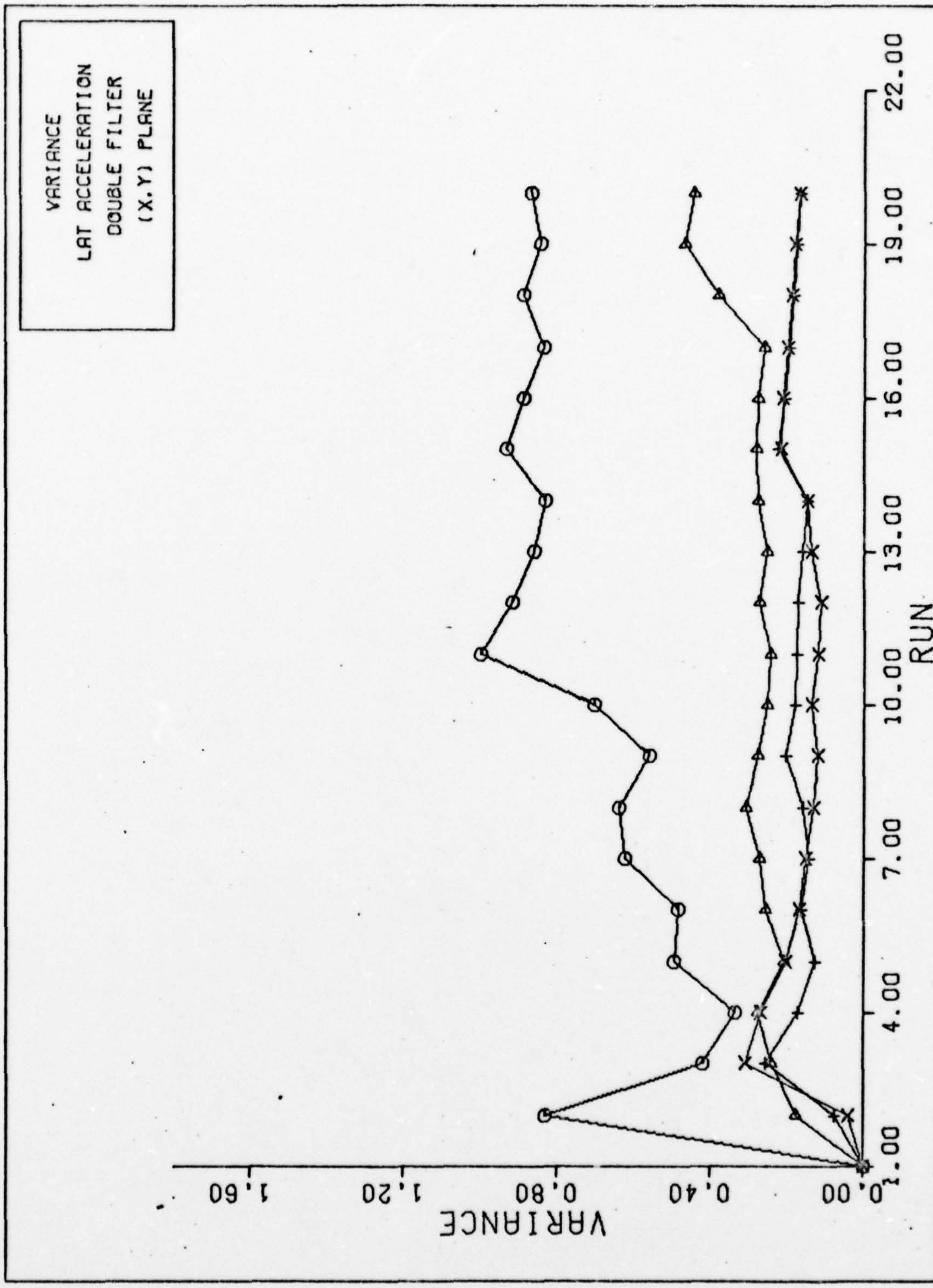



Fig. C-12 VARIANCE CONVERGENCE

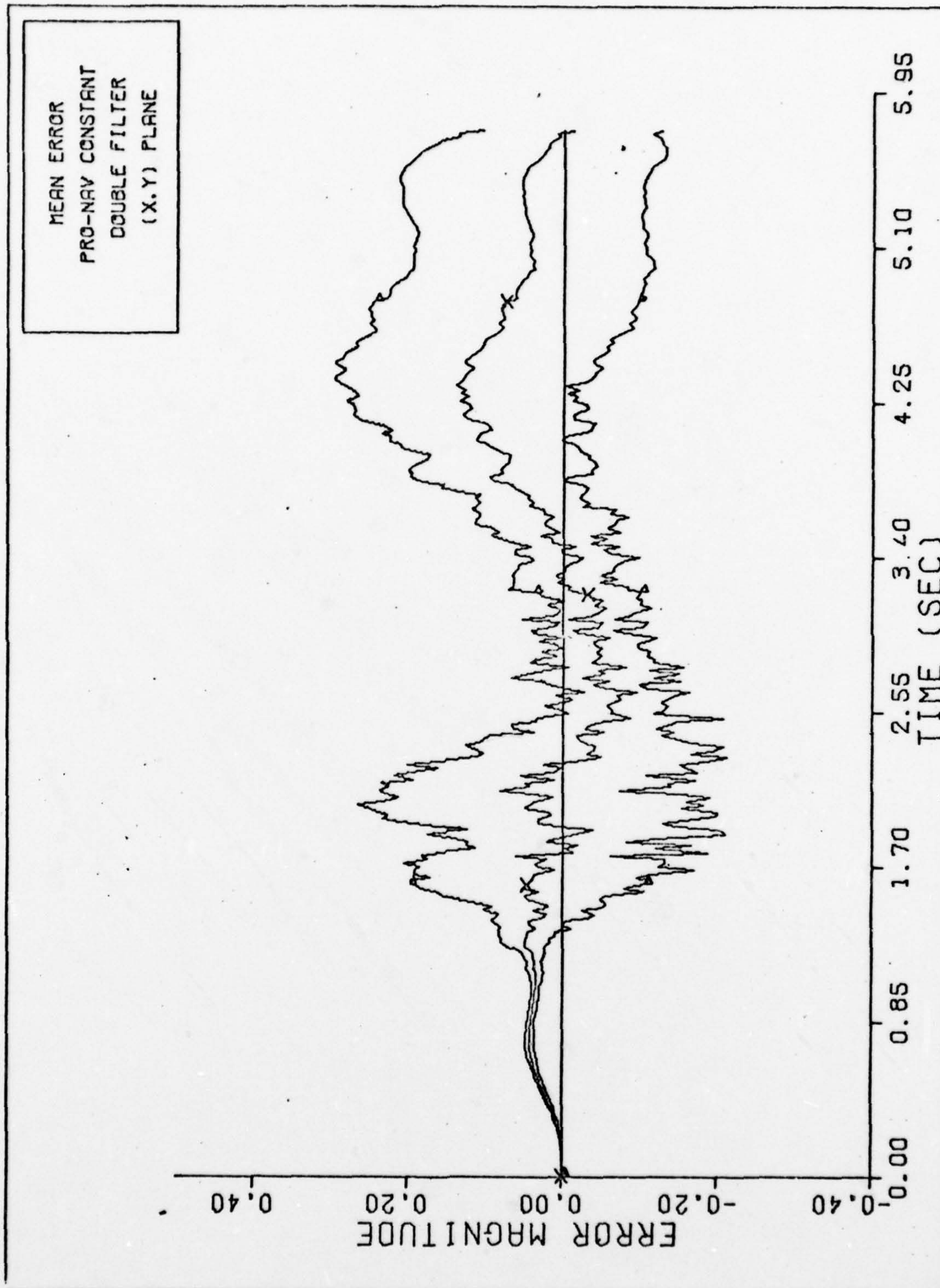



Fig. C-13 PRO-NAV CONSTANT DOUBLE FILTER

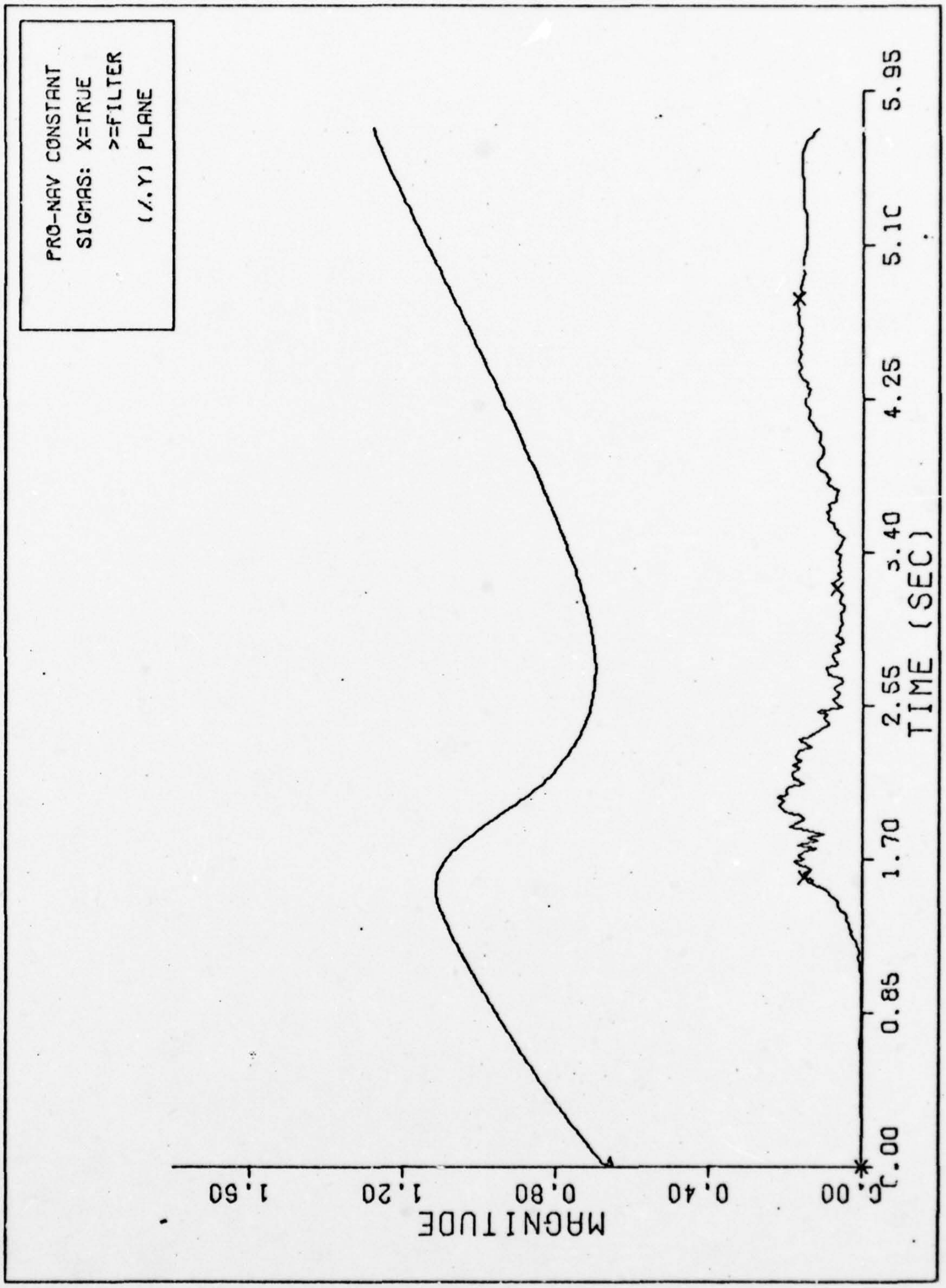



Fig. C-14 PRO-NAV CONSTANT SIGMAS DOUBLE FILTER

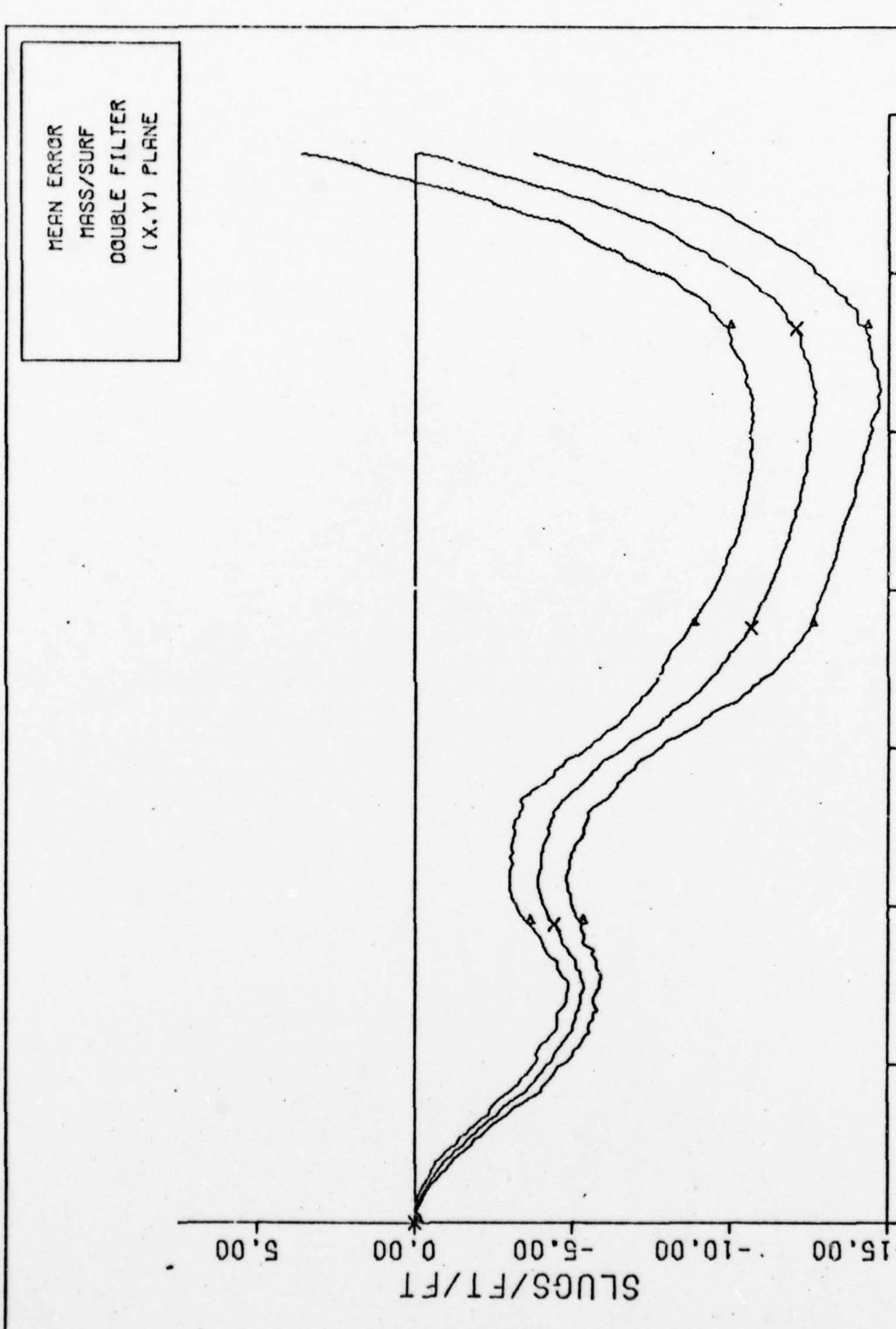



Fig. C-15 MASS/SURF DOUBLE FILTER

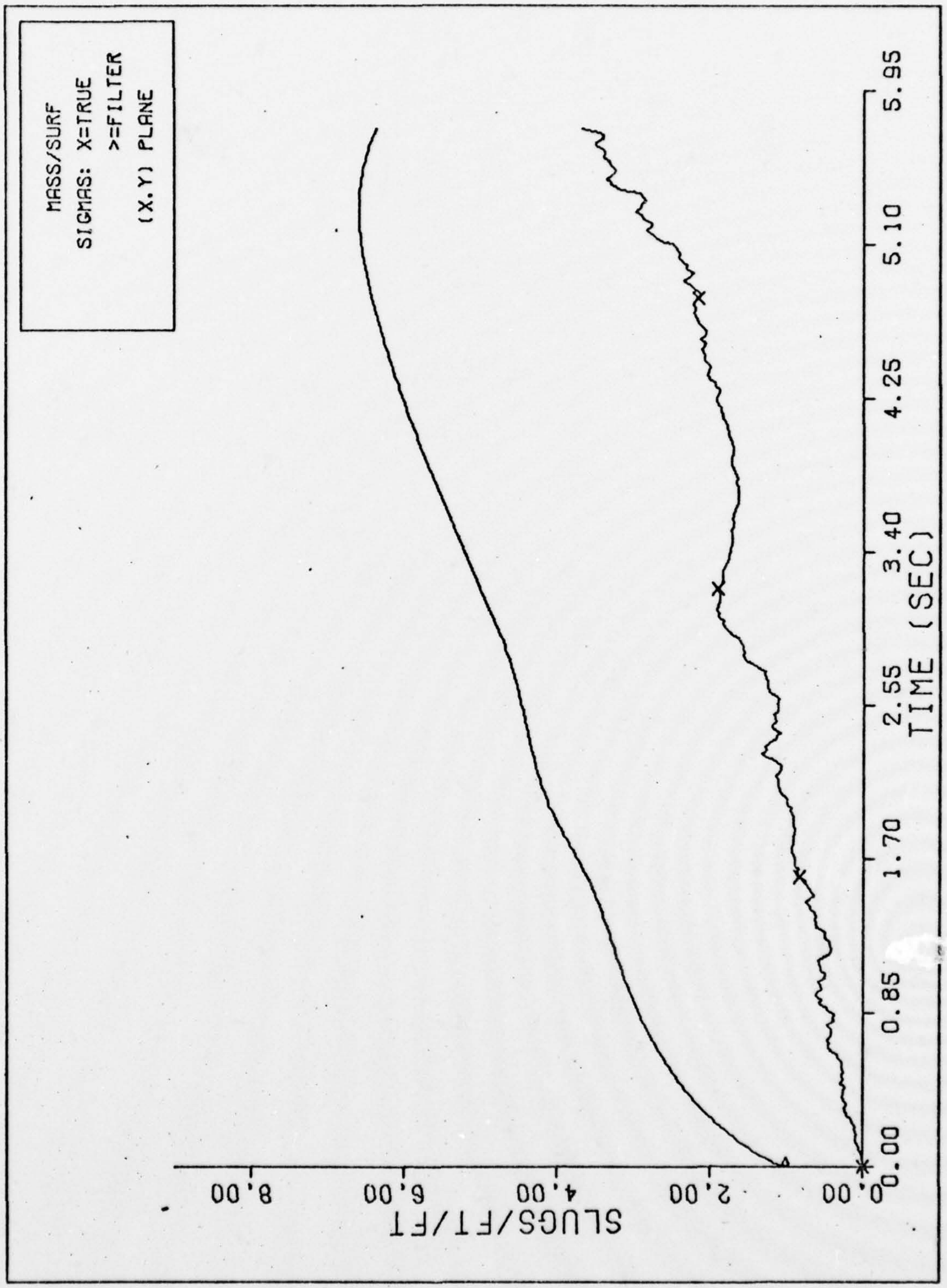



Fig. C-16 MASS/SURF SIGMAS DOUBLE FILTER

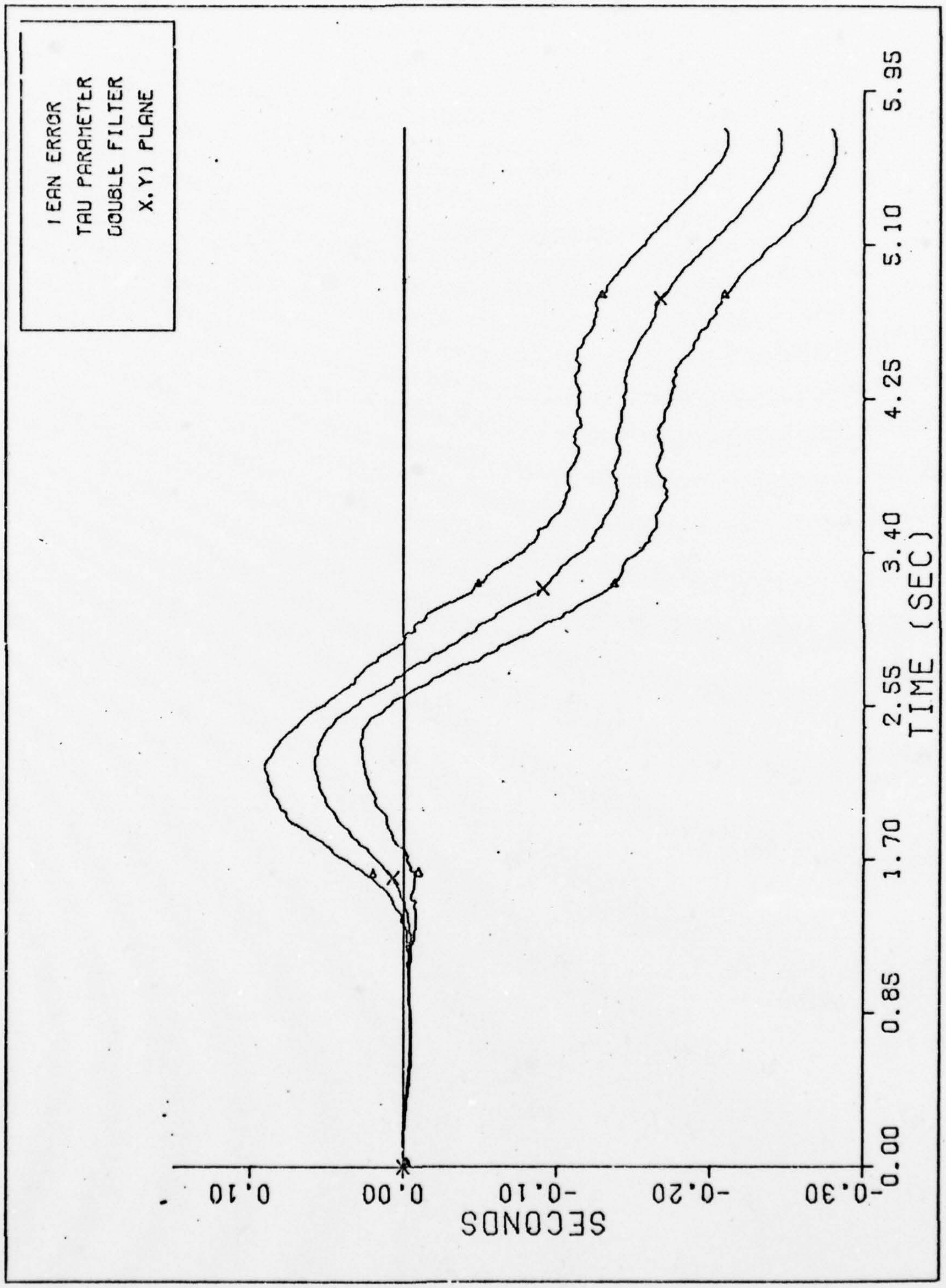



Fig. C-17 TAU PARAMETER DOUBLE FILTER

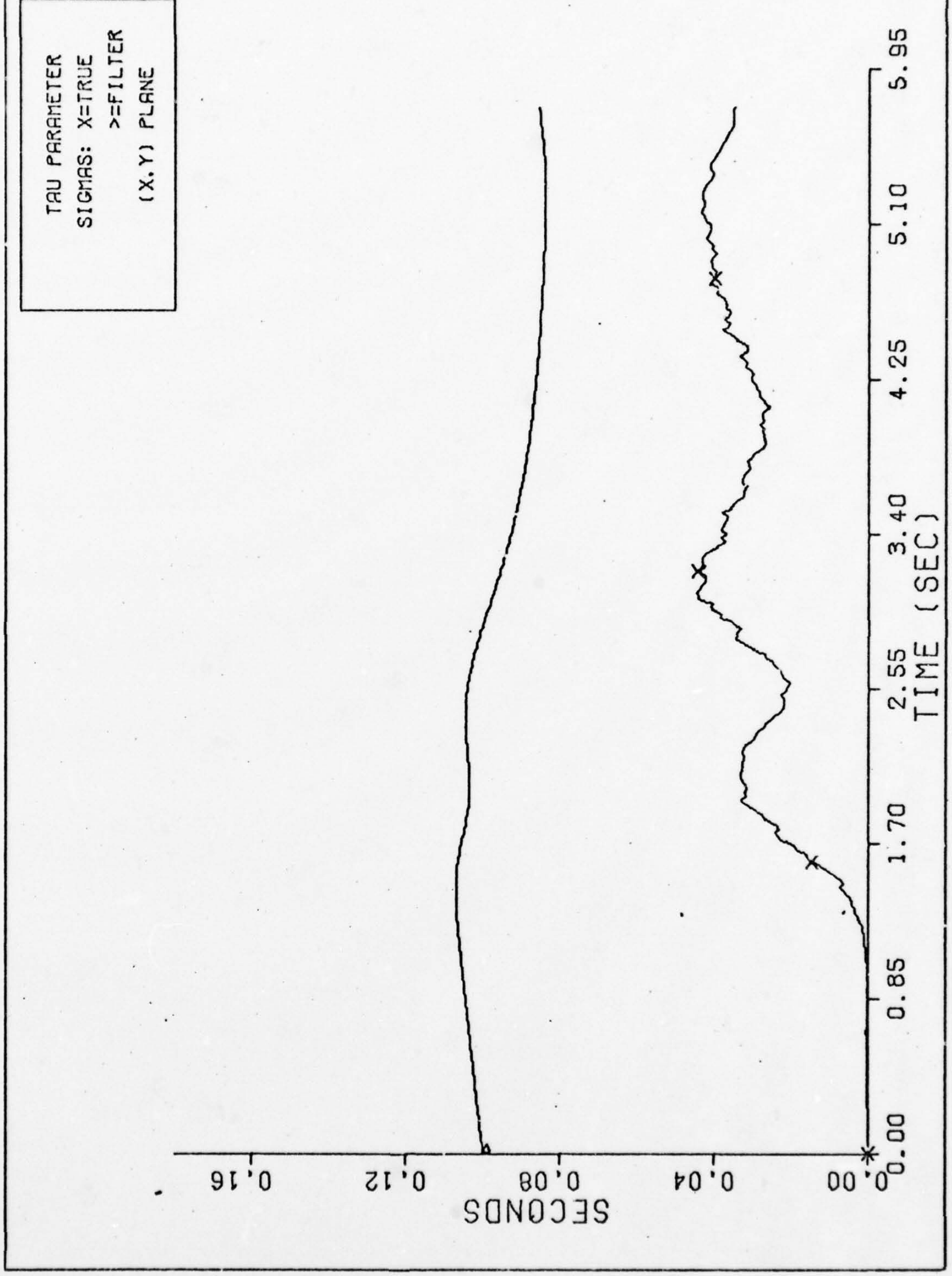



Fig. C-18 TAU PARAMETER SIGMAS DOUBLE FILTER

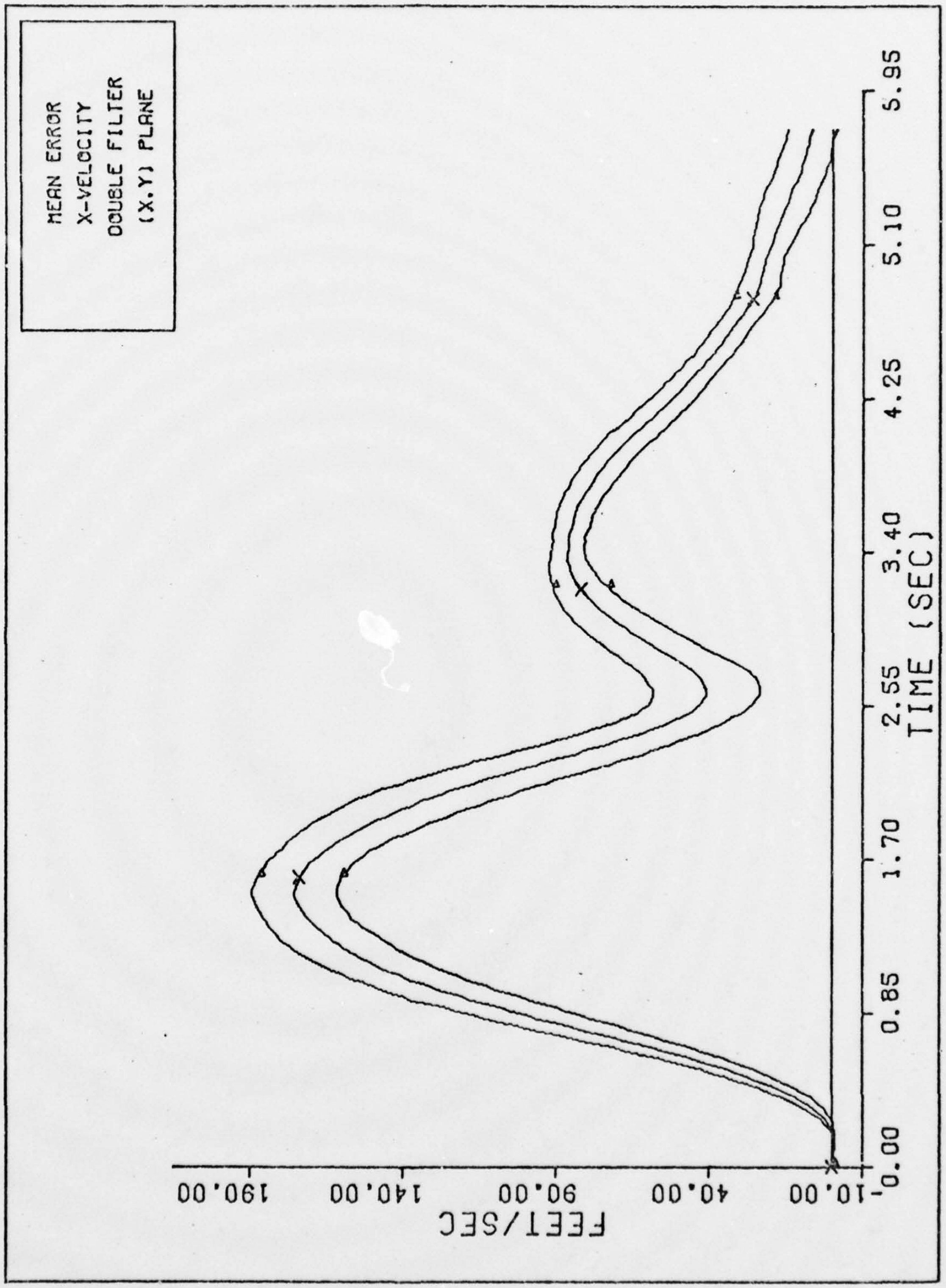



Fig. C-19 X-VELOCITY DOUBLE FILTER

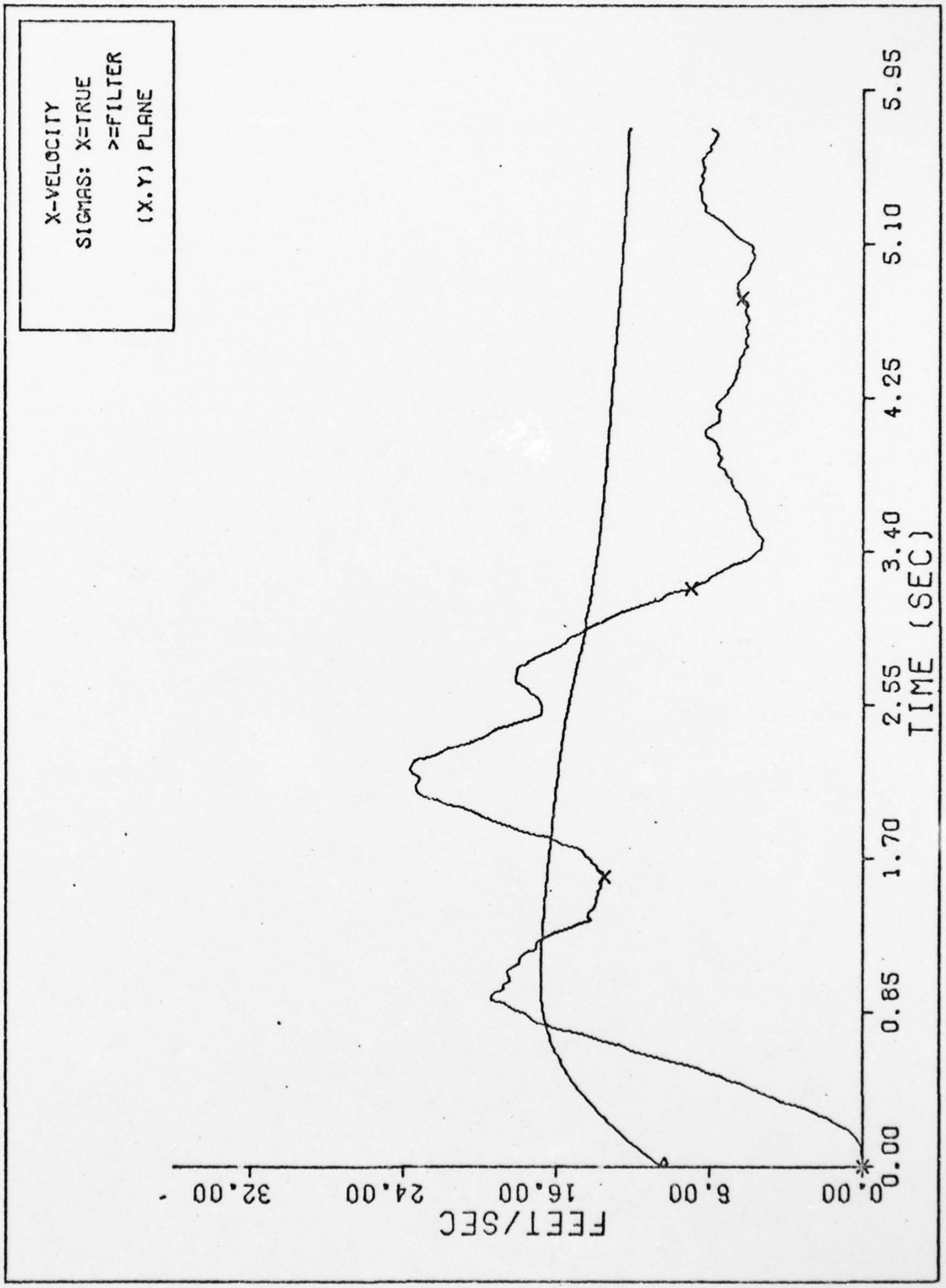



Fig. C-20 X-VELOCITY SIGMAS DOUBLE FILTER

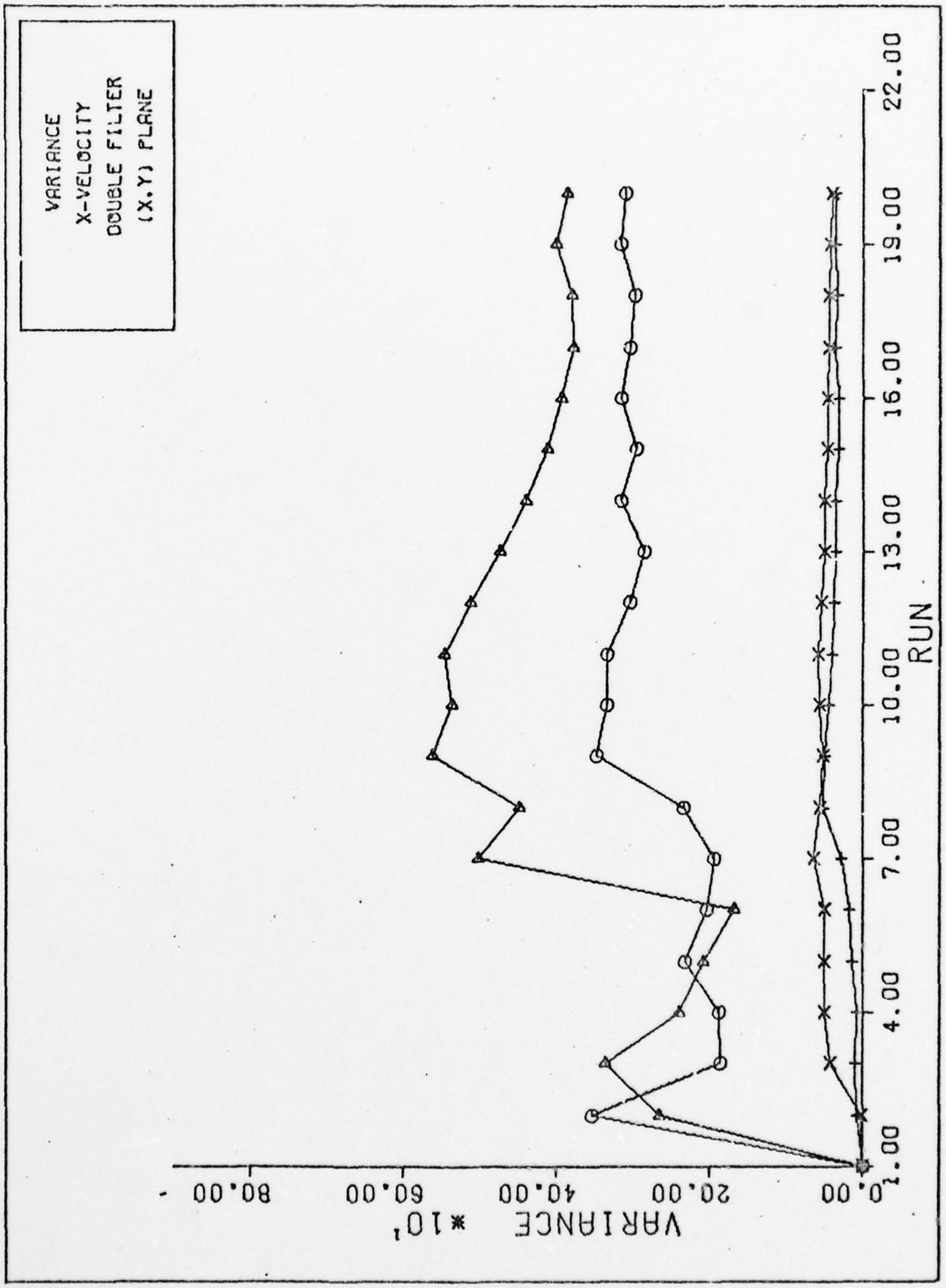



Fig. C-21 VARIANCE CONVERGENCE

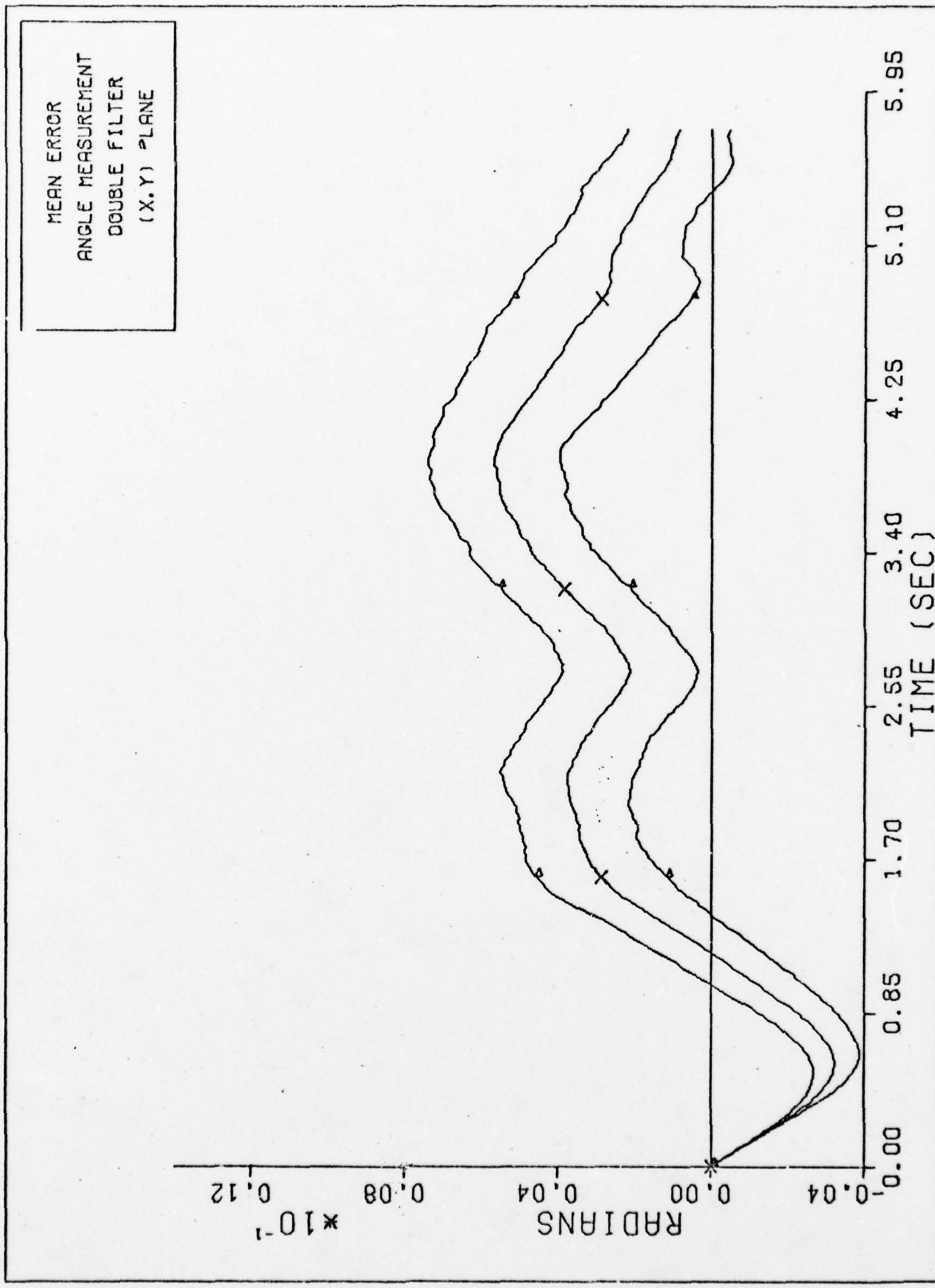



Fig. C-22 ANGLE MEASUREMENT DOUBLE FILTER

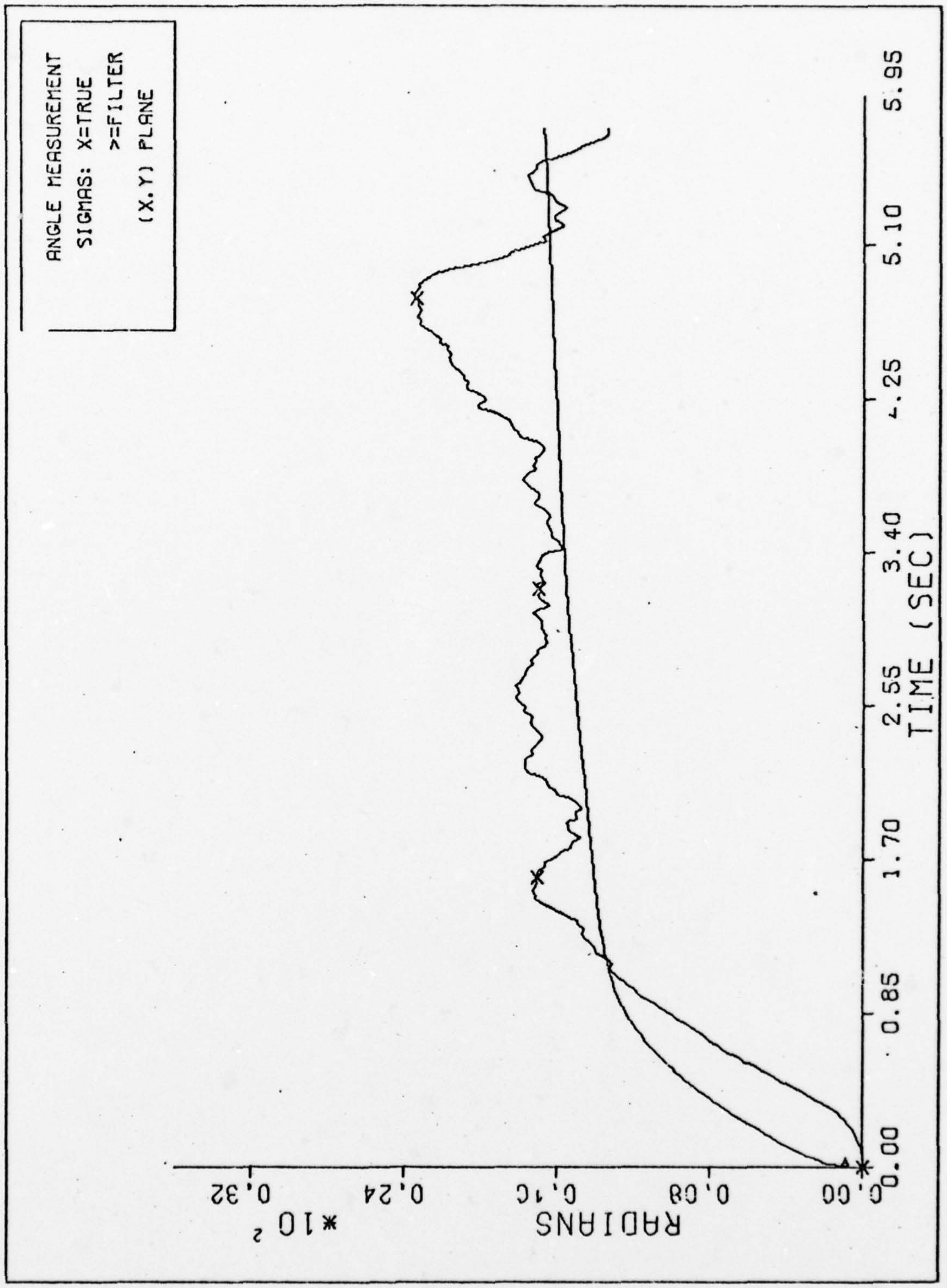



Fig. C-23 ANGLE MEASUREMENT SIGMAS DOUBLE FILTER

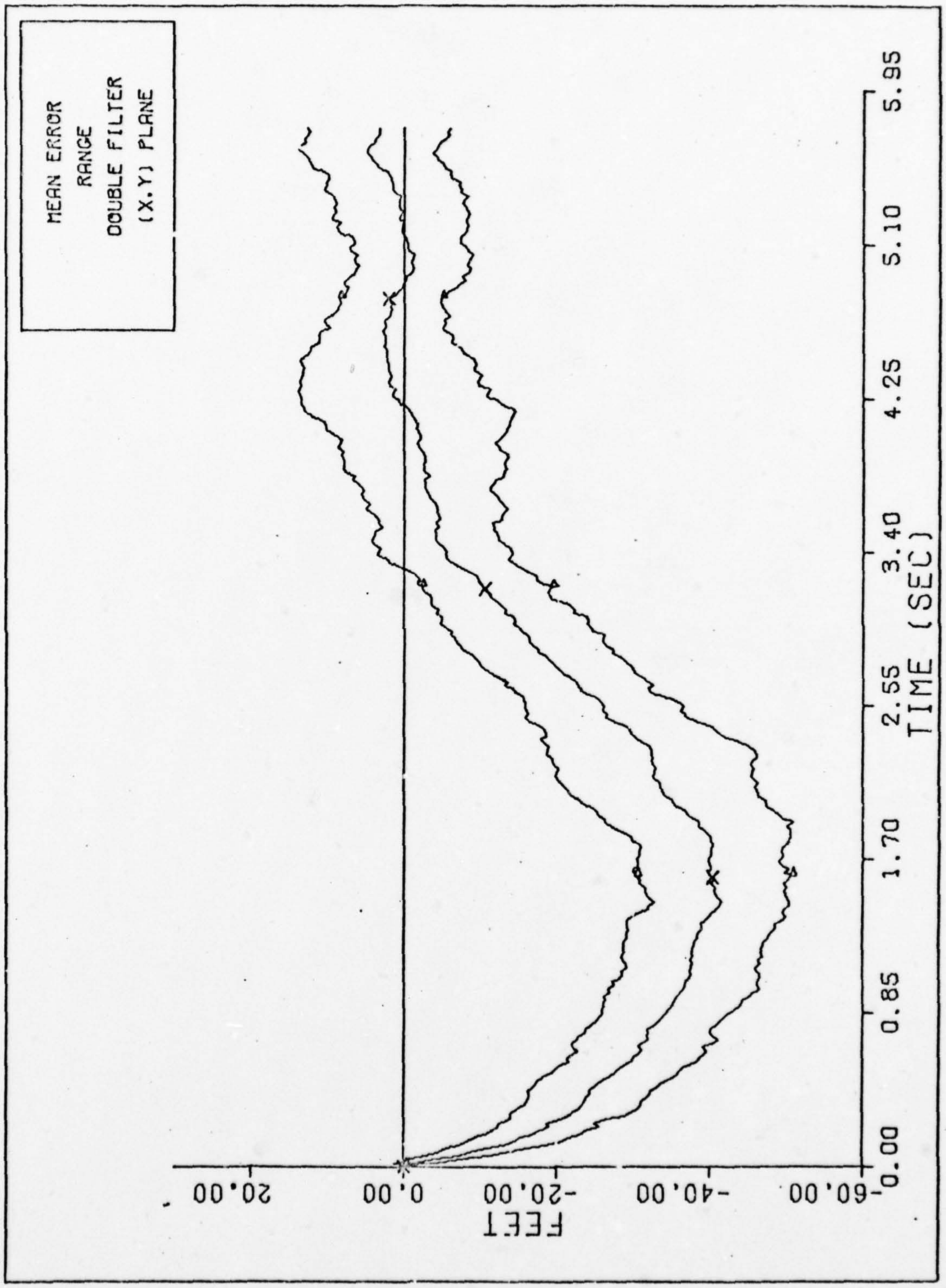



Fig. C-24 RANGE DOUBLE FILTER

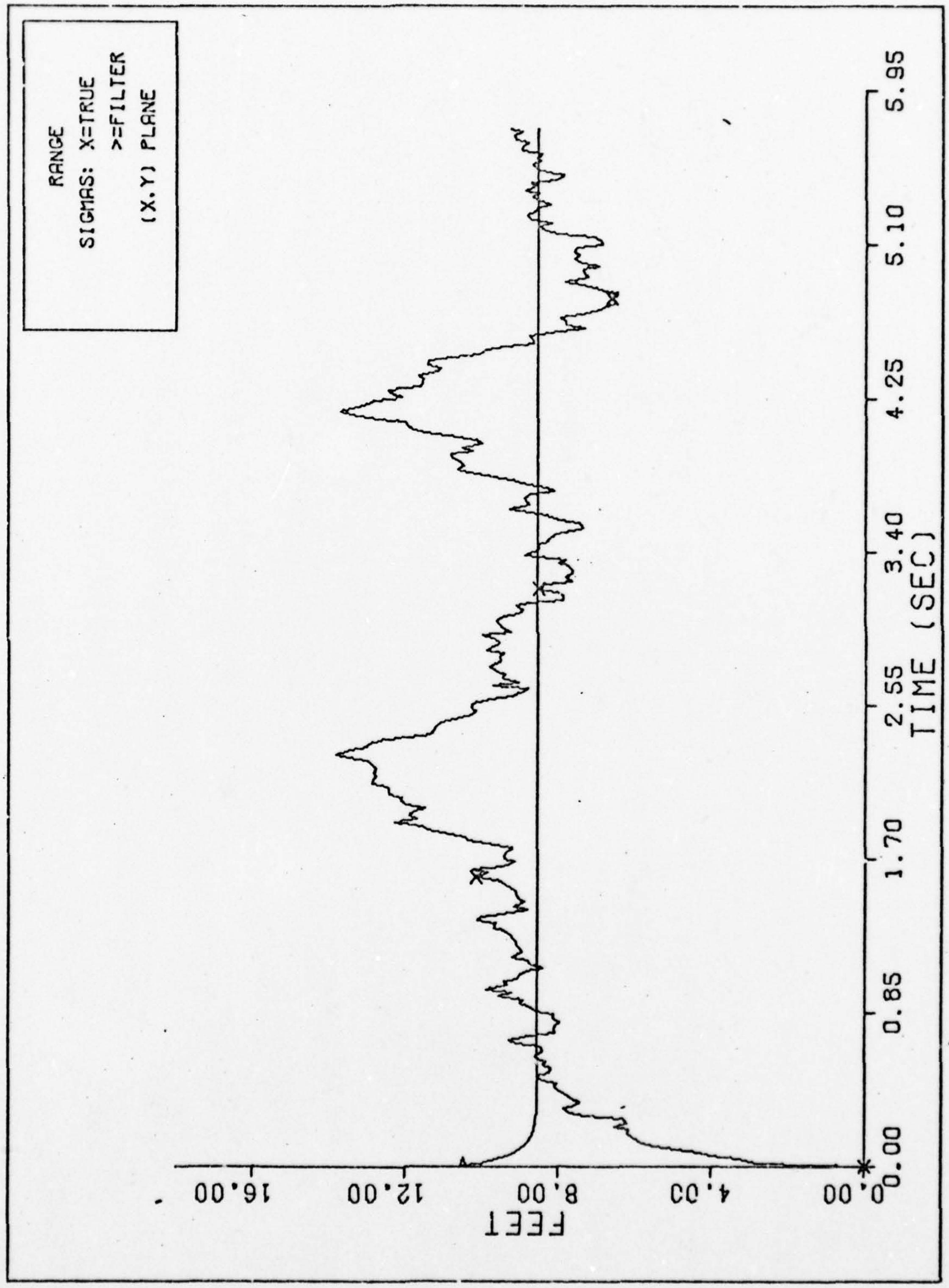



Fig. C-25 RANGE SIGMAS DOUBLE FILTER

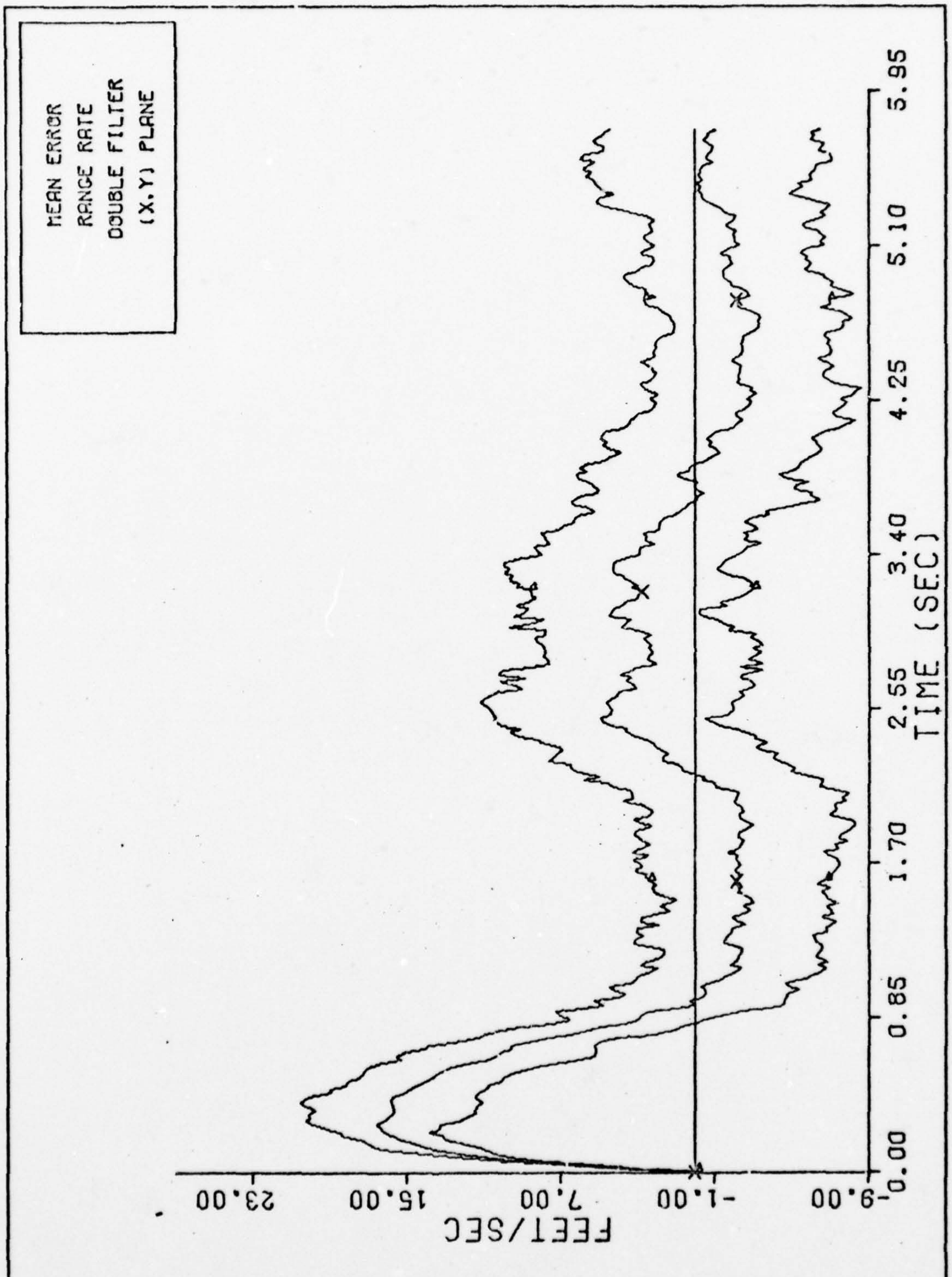



Fig. C-26 RANGE RATE DOUBLE FILTER

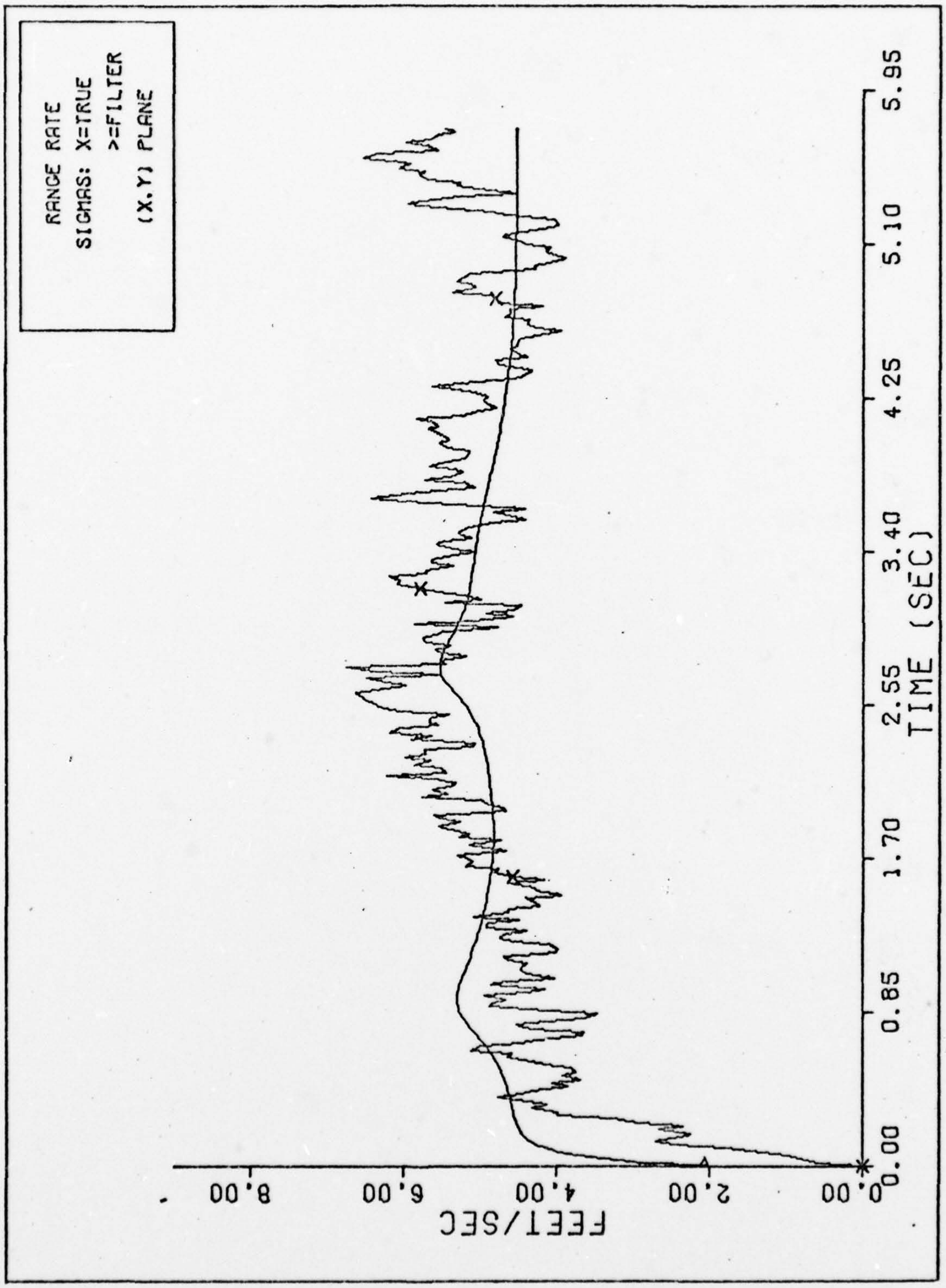



Fig. C-27 RANGE RATE SIGMAS DOUBLE FILTER

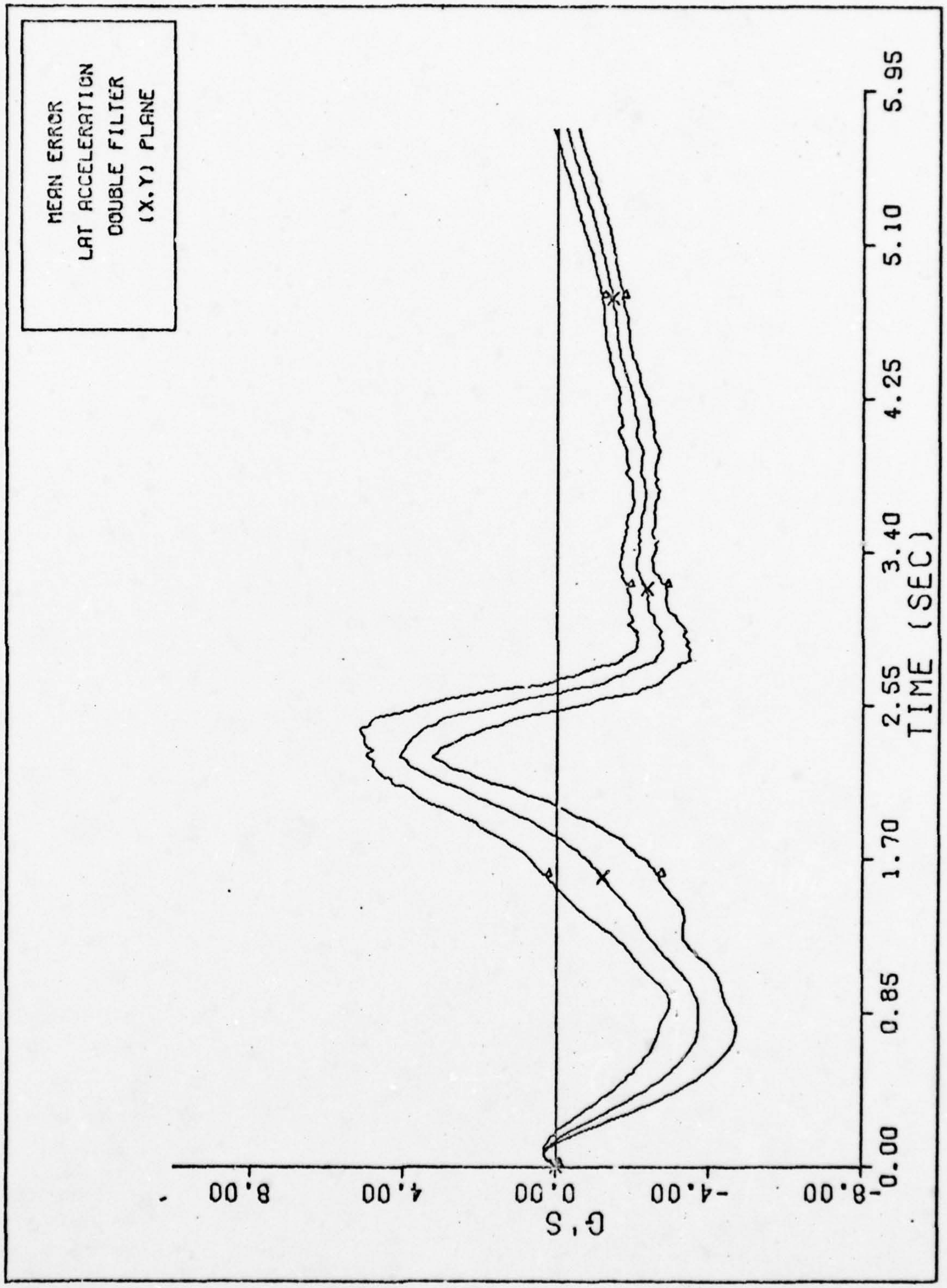



Fig. C-28 LAT ACCELERATION DOUBLE FILTER

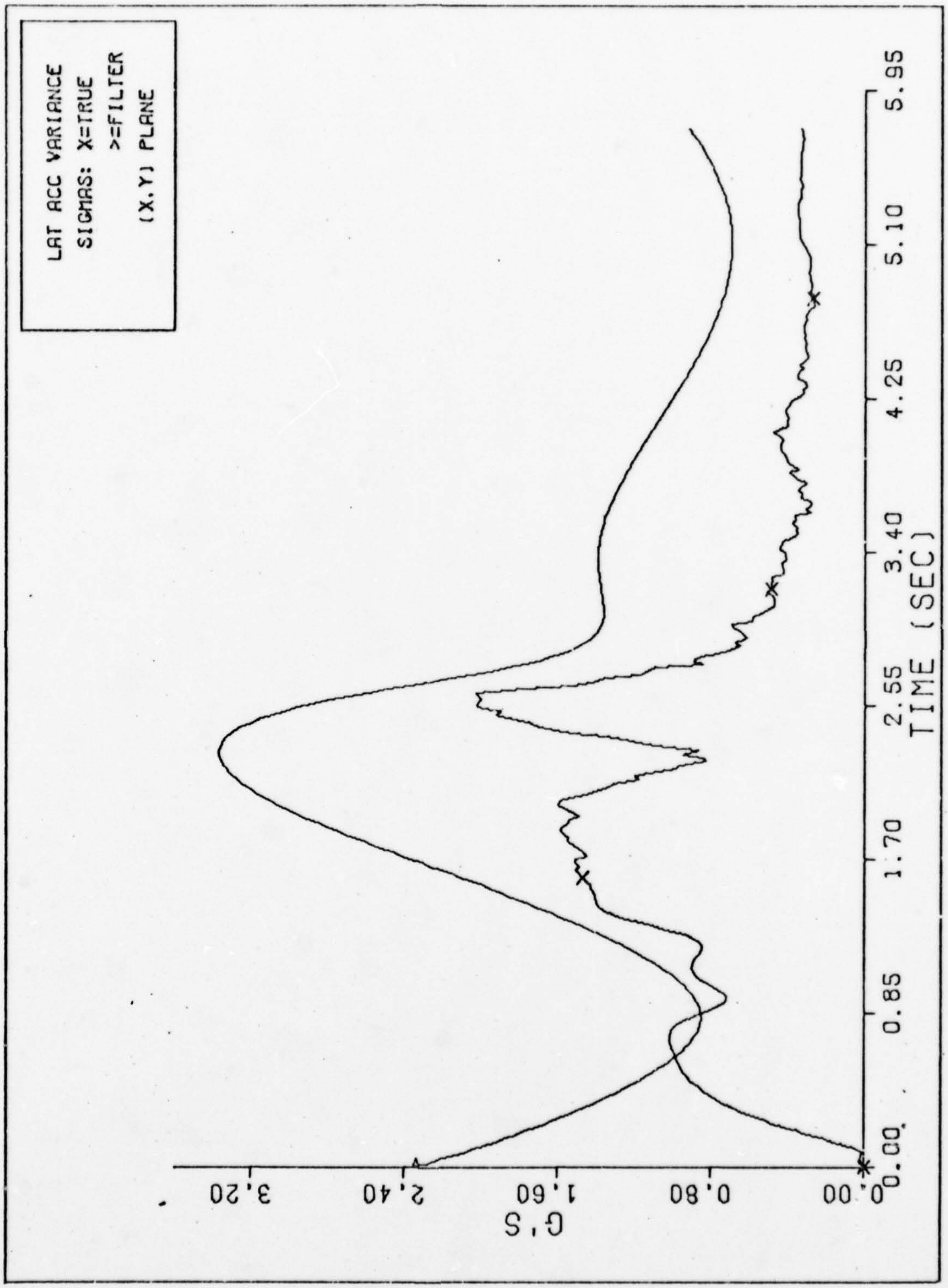



Fig. C-29 LAT ACCELERATION SIGMAS DOUBLE FILTER

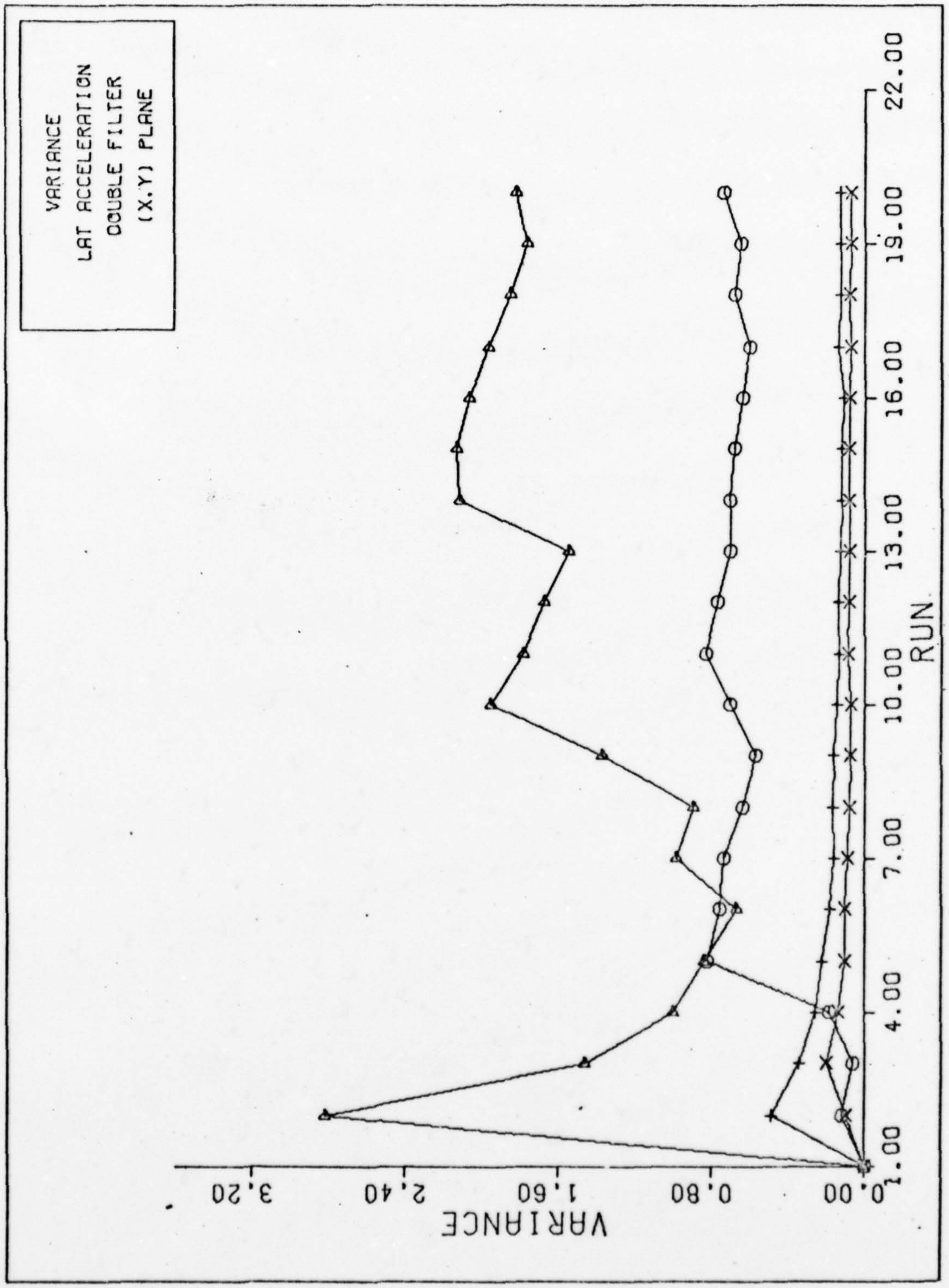



Fig. C-30 VARIANCE CONVERGENCE

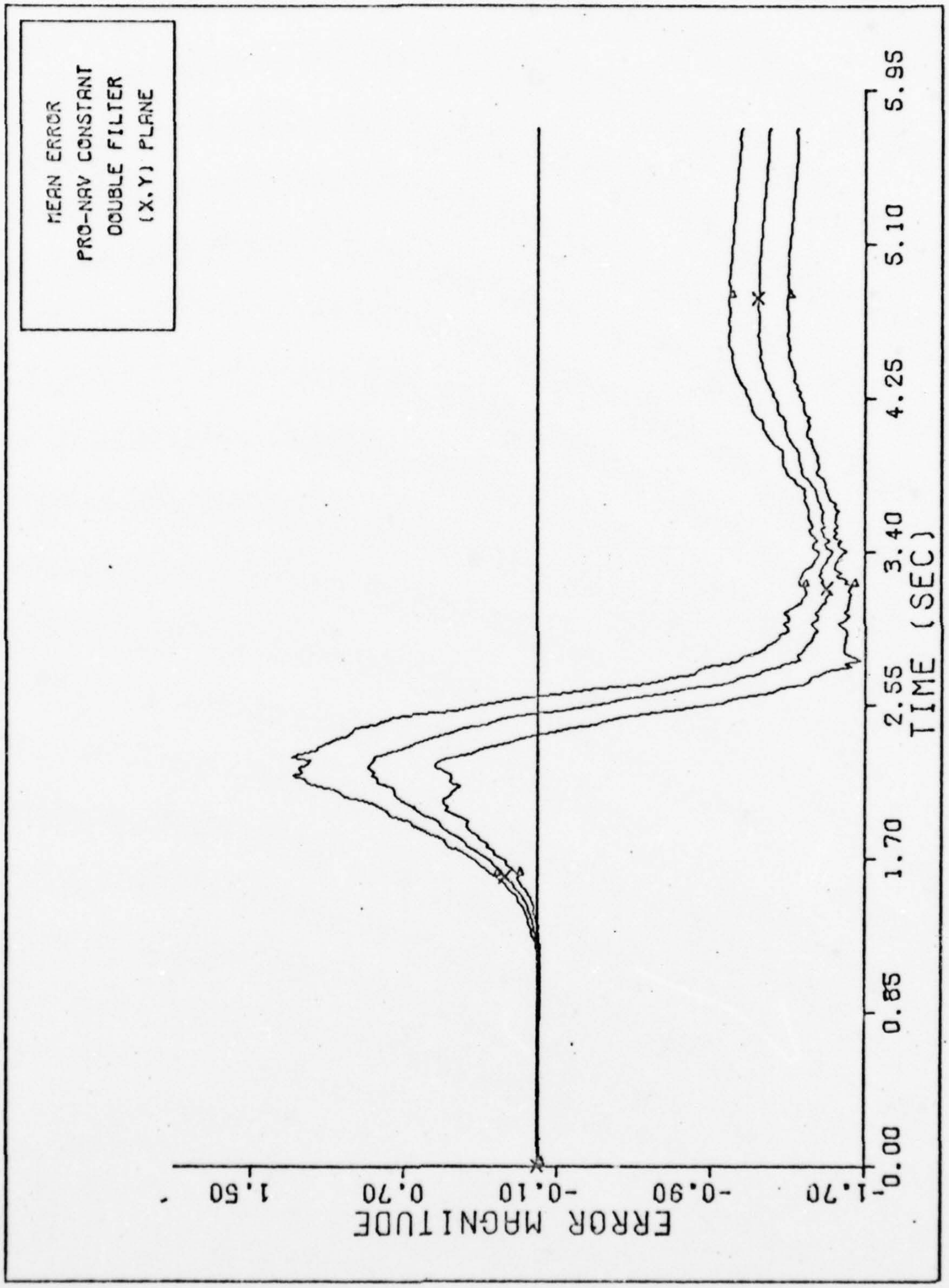



Fig. C-31 PRO-NAV CONSTANT DOUBLE FILTER

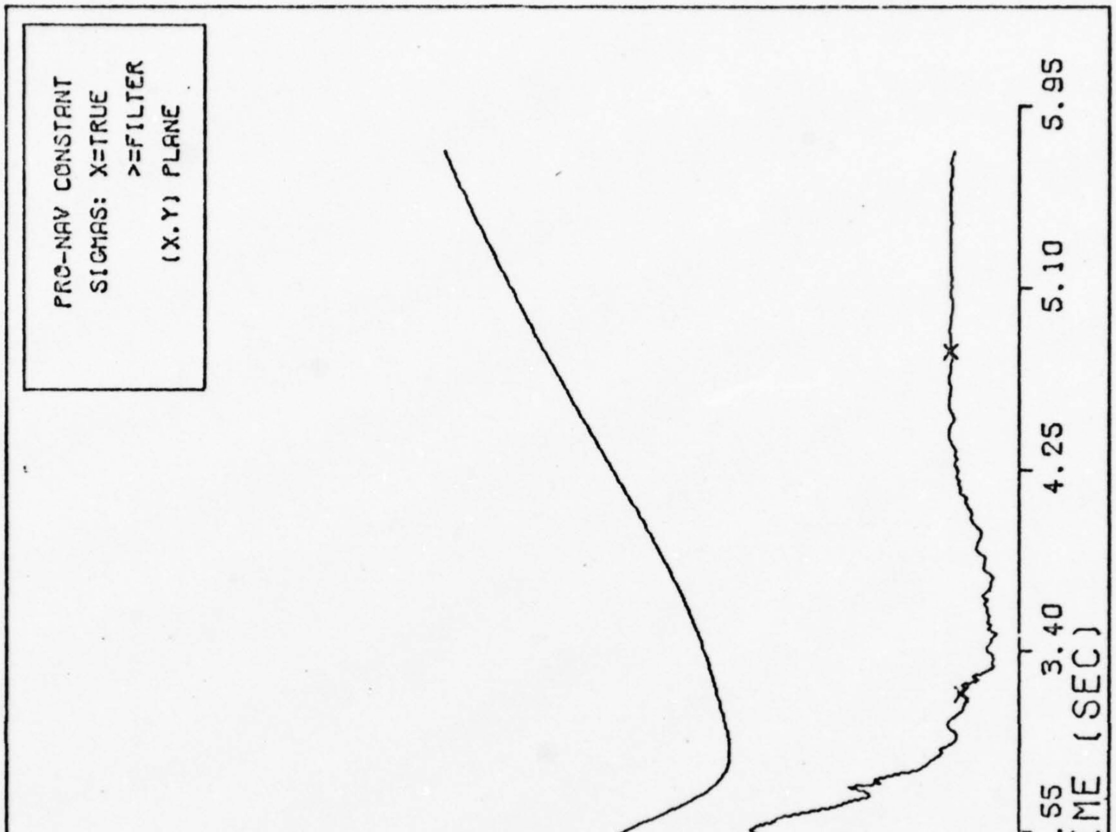



Fig. C-32 PRO-NAV CONSTANT SIGMAS DOUBLE FILTER

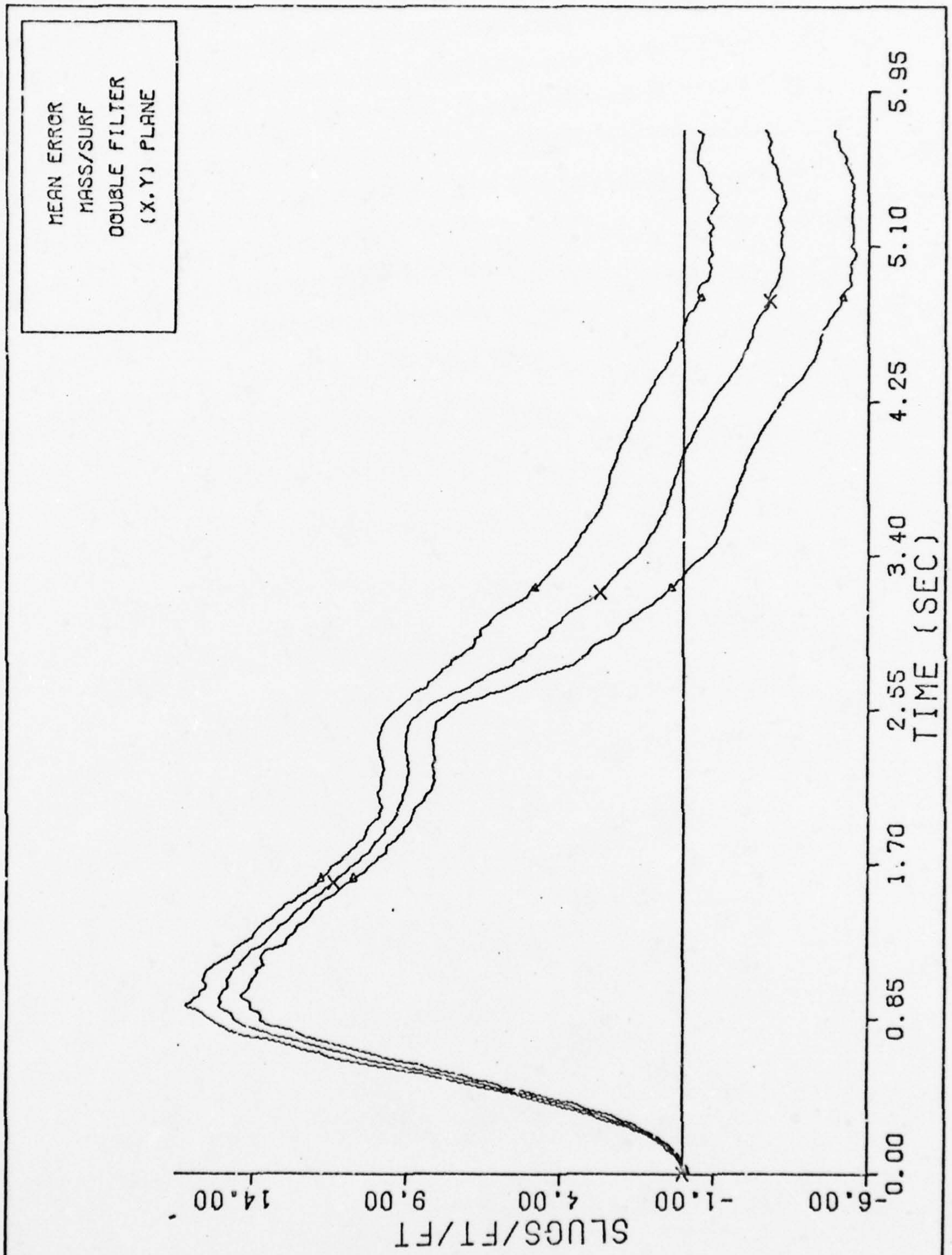



Fig. C-33 MASS/SURF DOUBLE FILTER

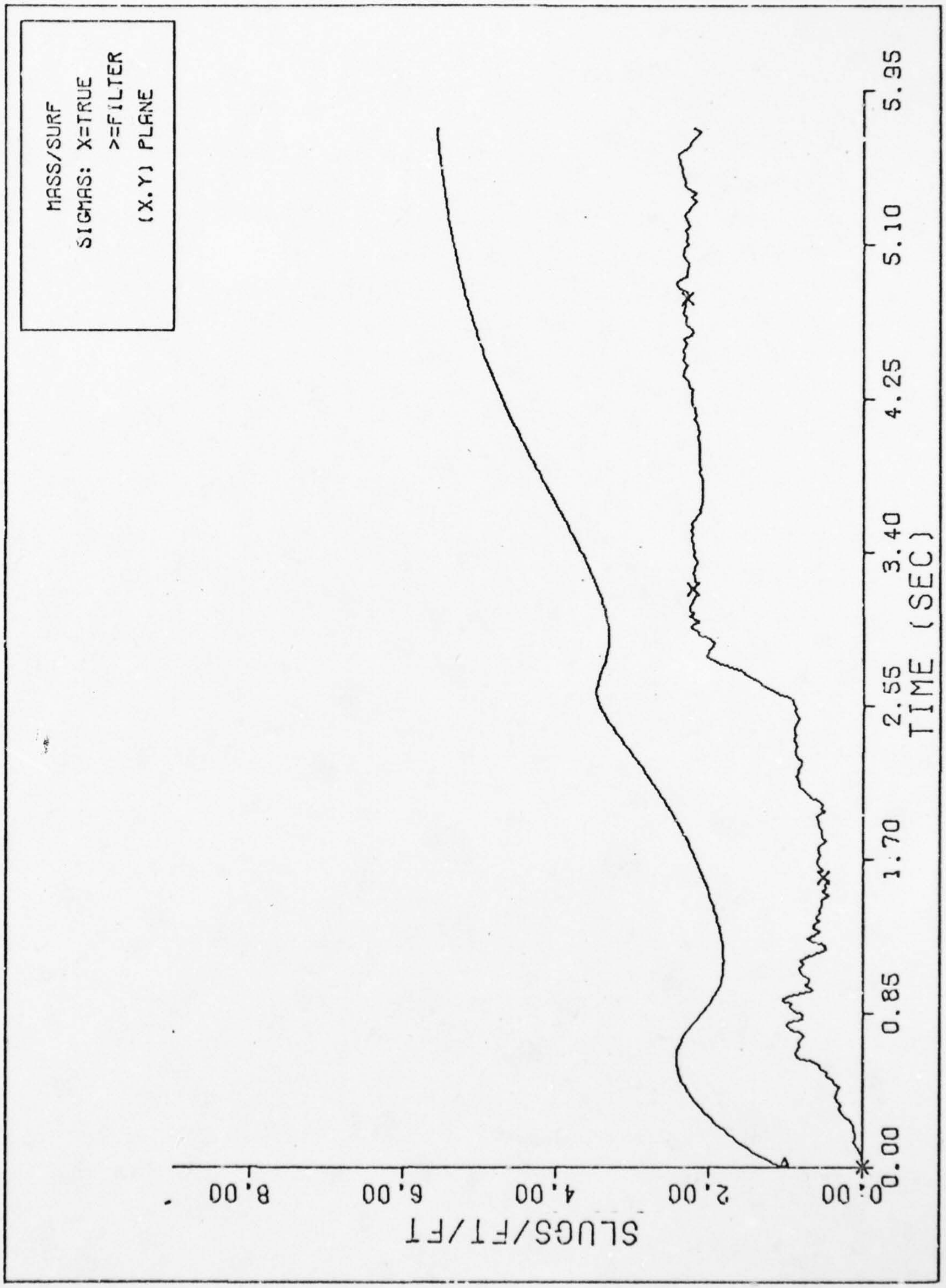



Fig. C-34 MASS/SURF SIGMAS DOUBLE FILTER

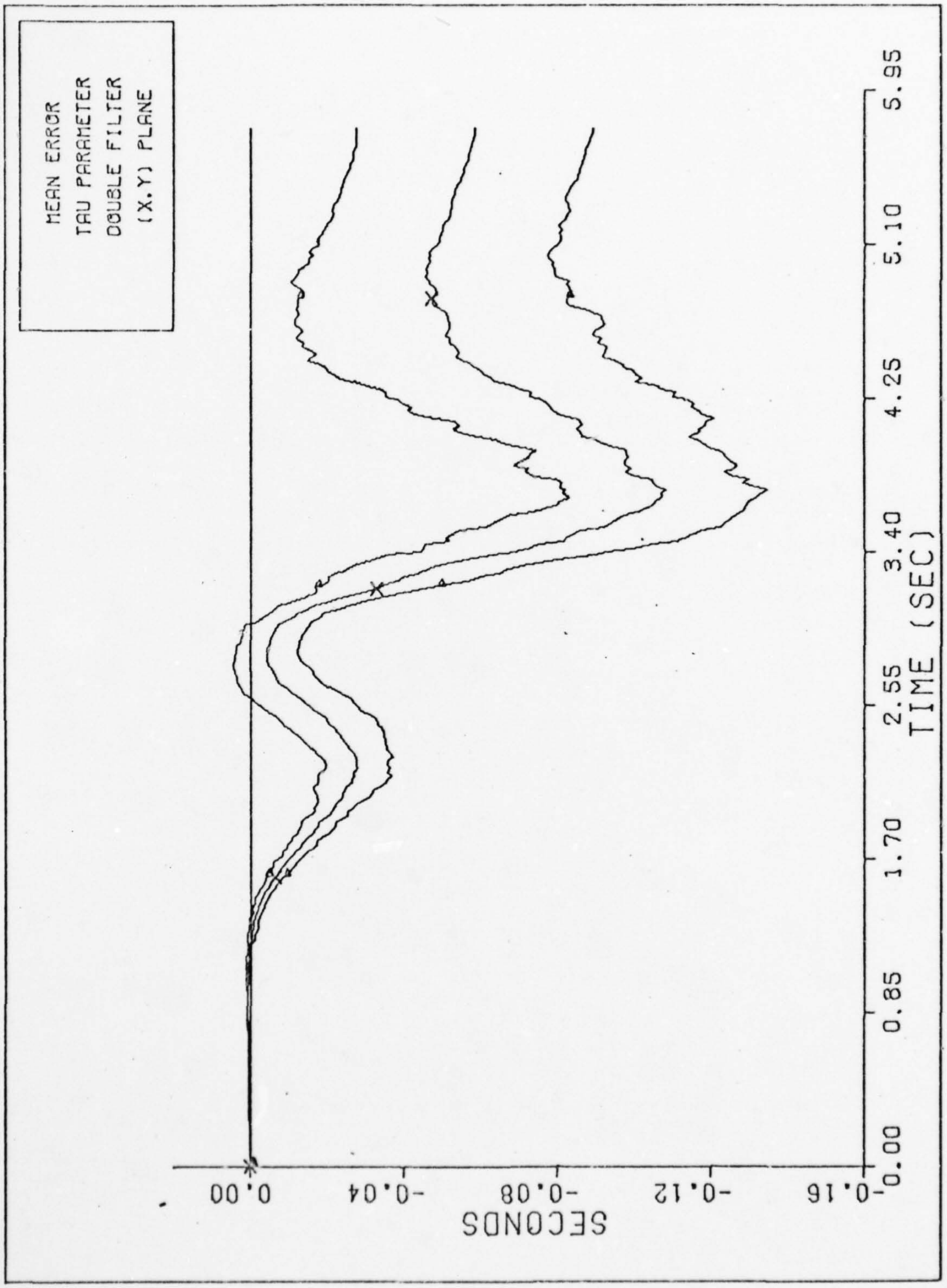



Fig. C-35 TAU PARAMETER DOUBLE FILTER

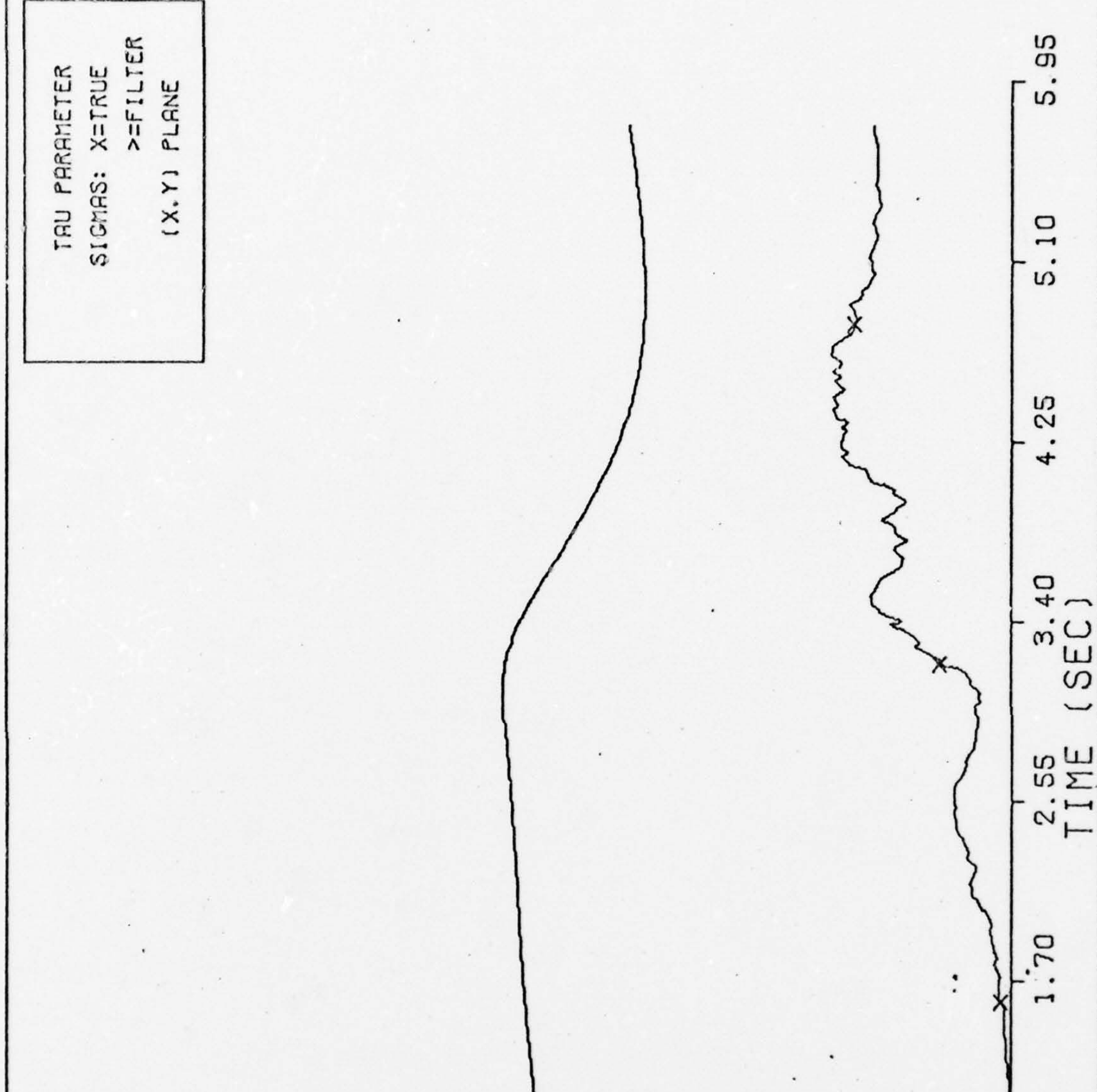



Fig. C-36 TAU PARAMETER SIGMAS DOUBLE FILTER

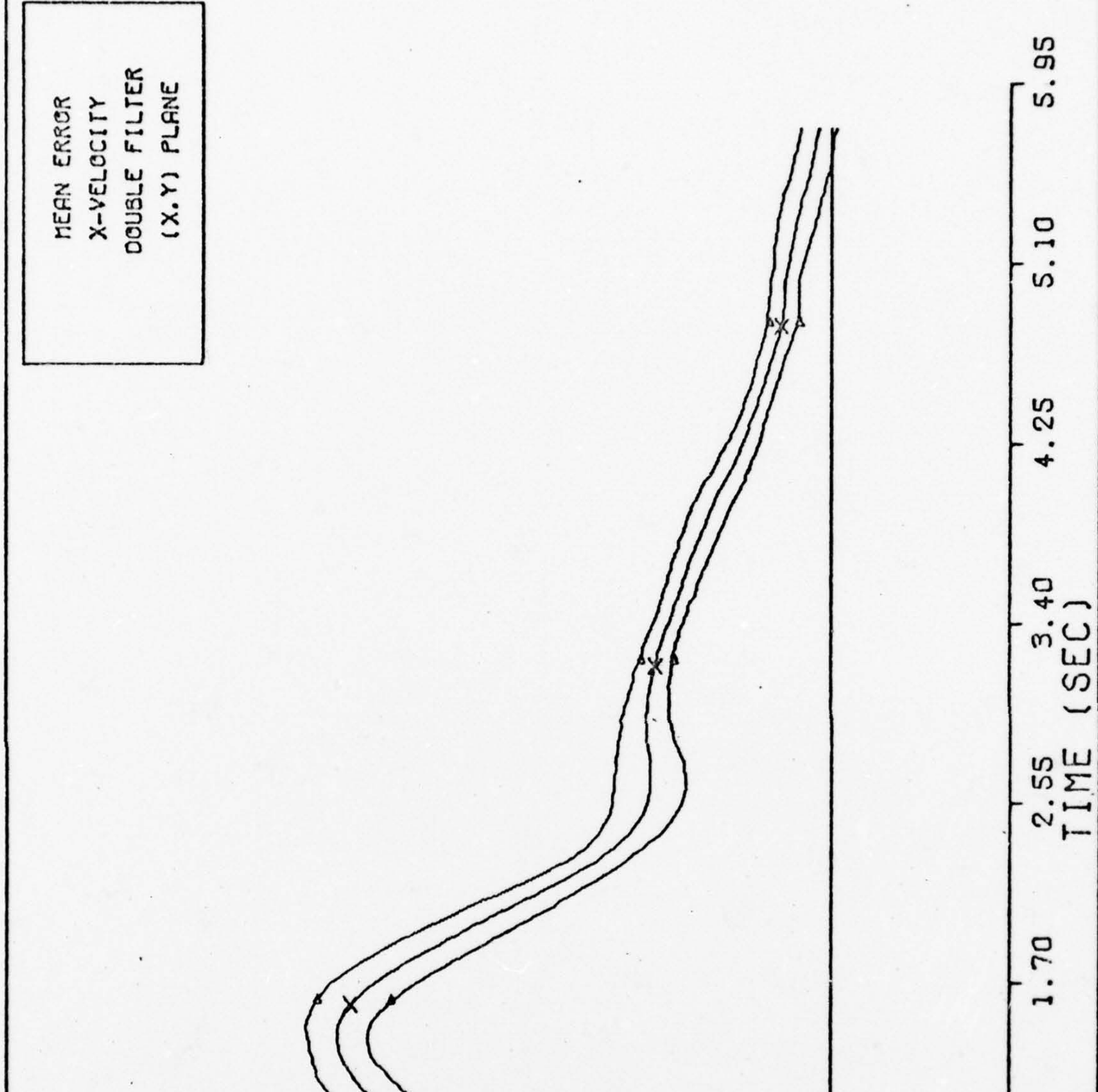



Fig. C-37 X-VELOCITY DOUBLE FILTER

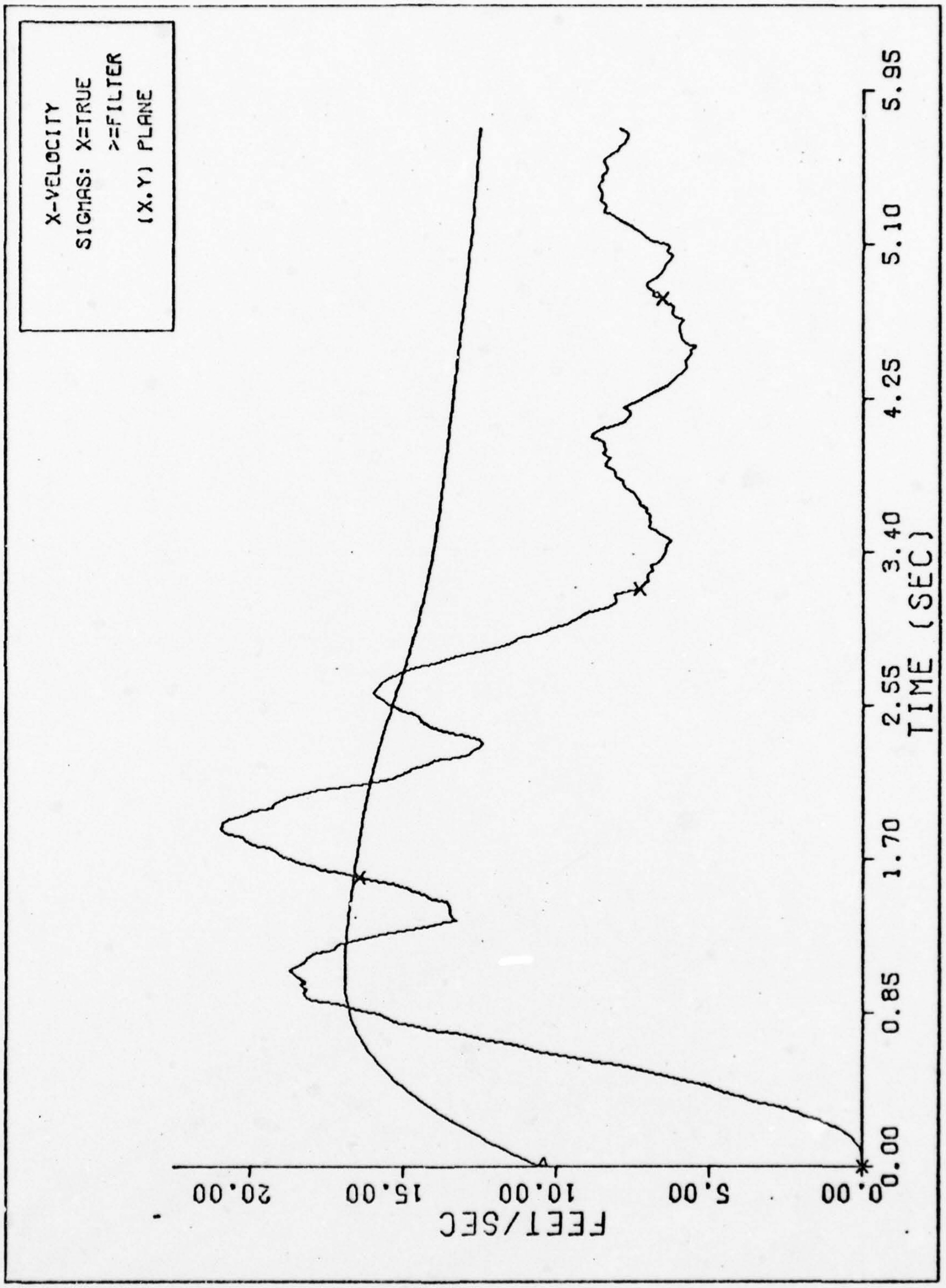



Fig. C-38 X-VELOCITY SIGMAS DOUBLE FILTER

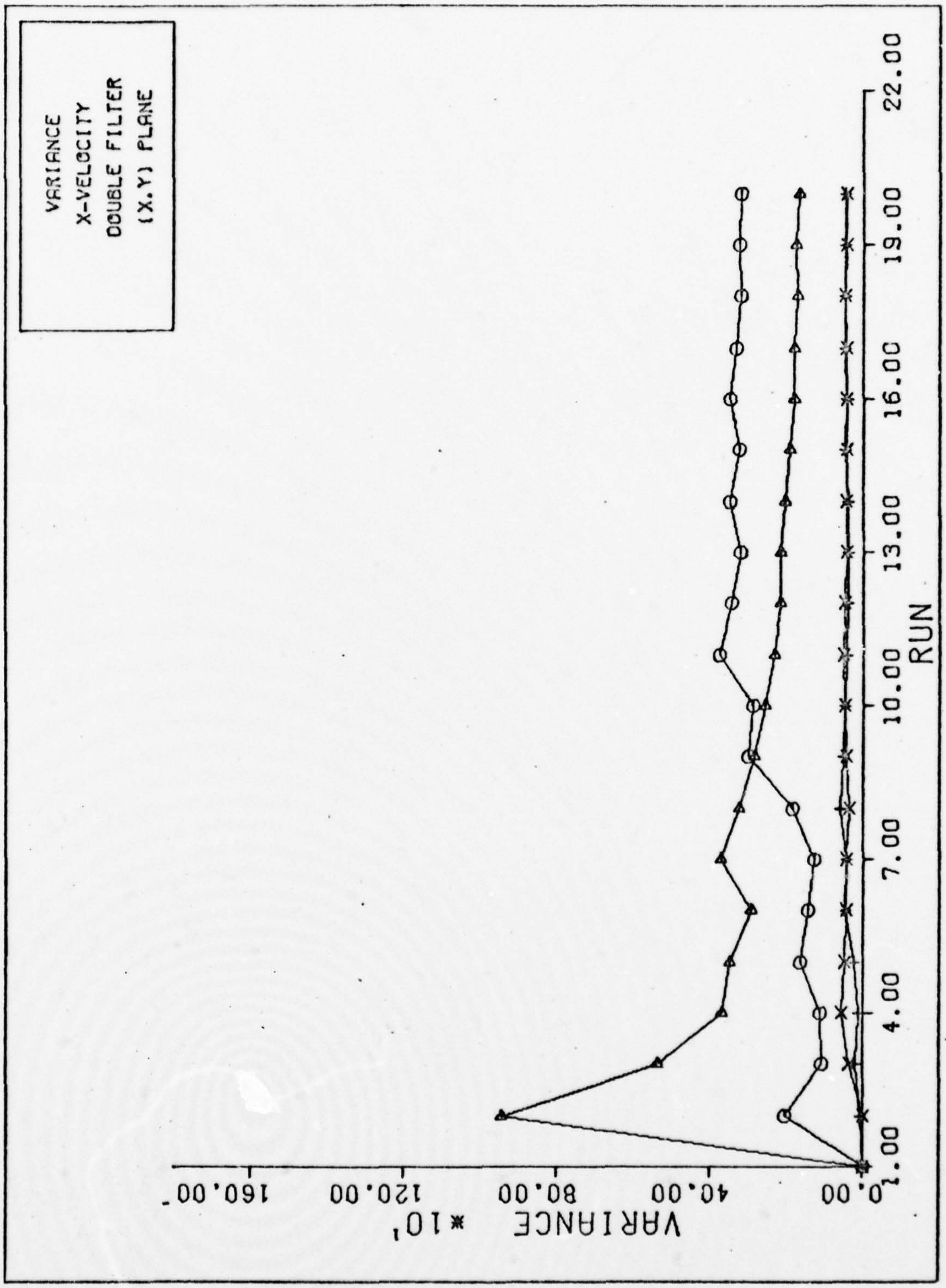



Fig. C-39 VARIANCE CONVERGENCE

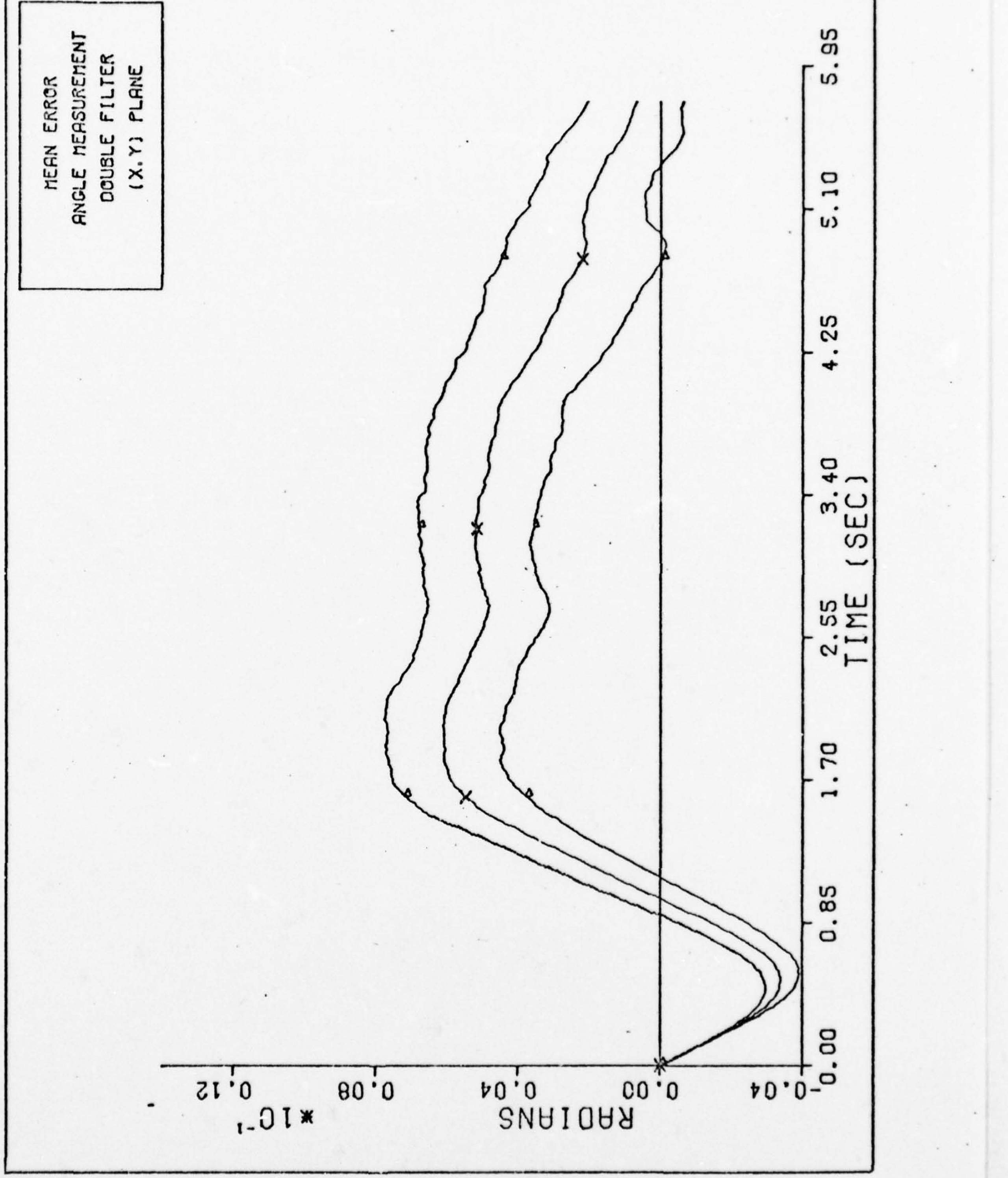



Fig. C-40 ANGLE MEASUREMENT DOUBLE FILTER

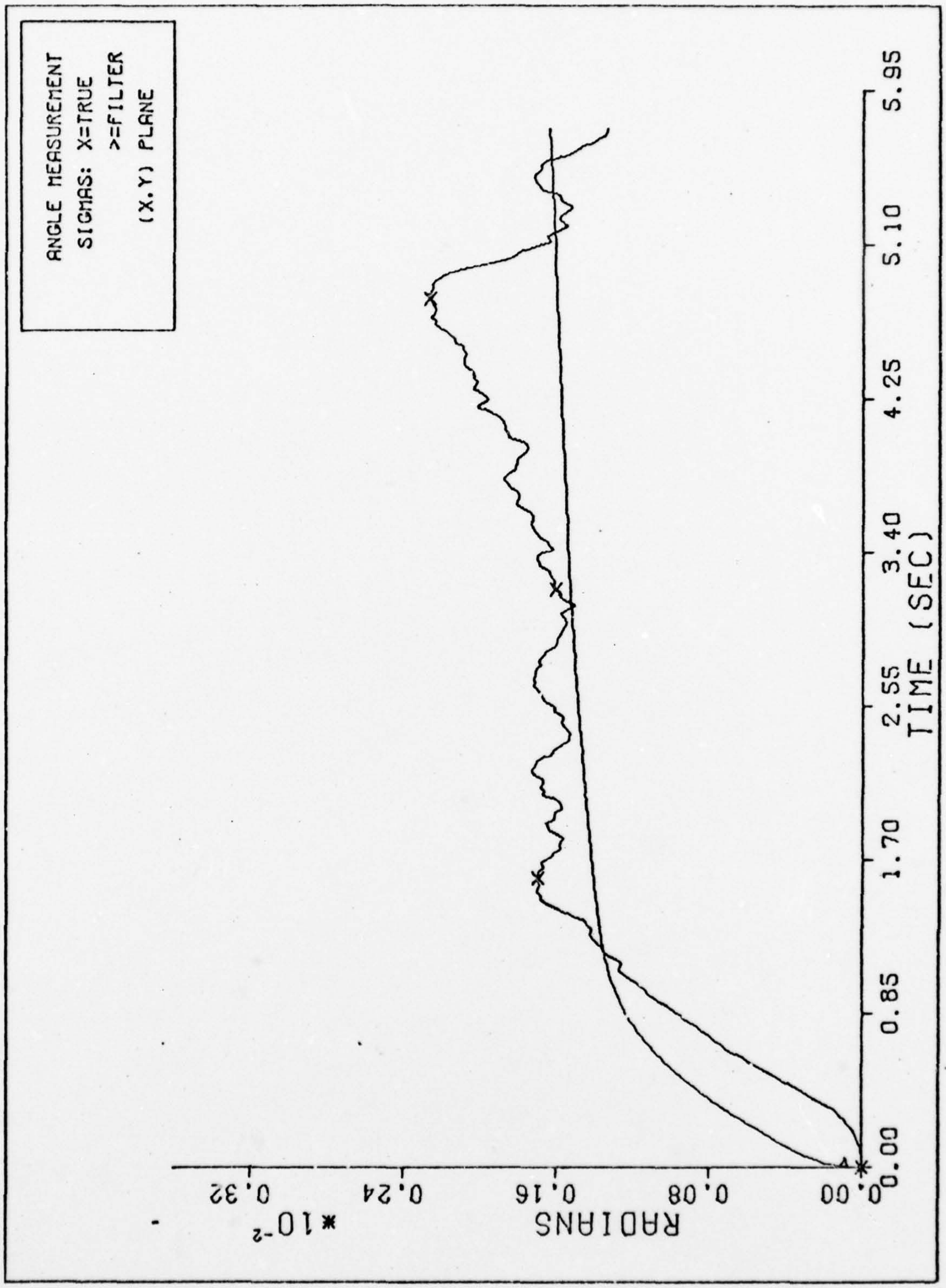



Fig. C-41 ANGLE MEASUREMENT SIGMAS DOUBLE FILTER

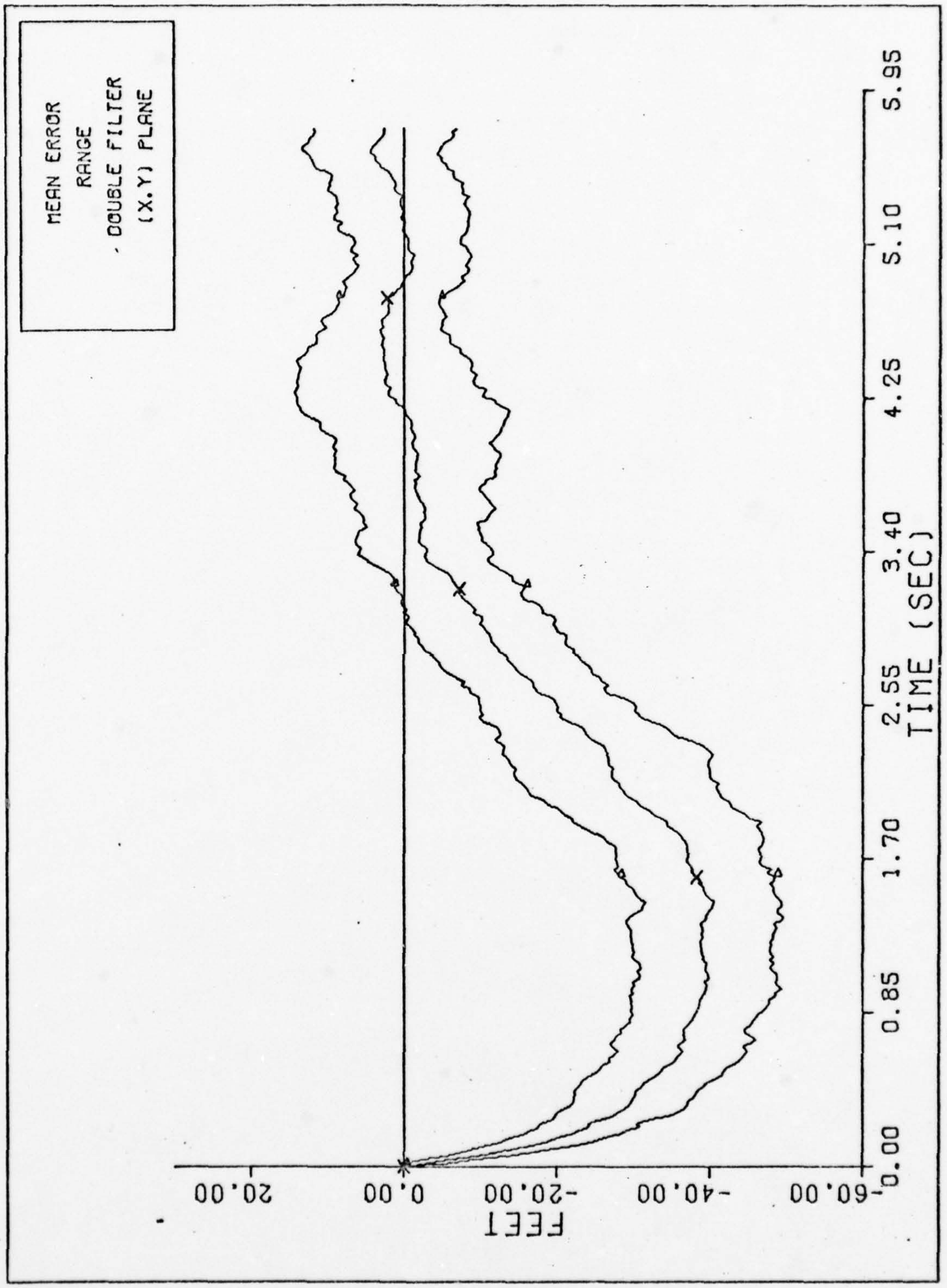



Fig. C-42

RANGE DOUBLE FILTER

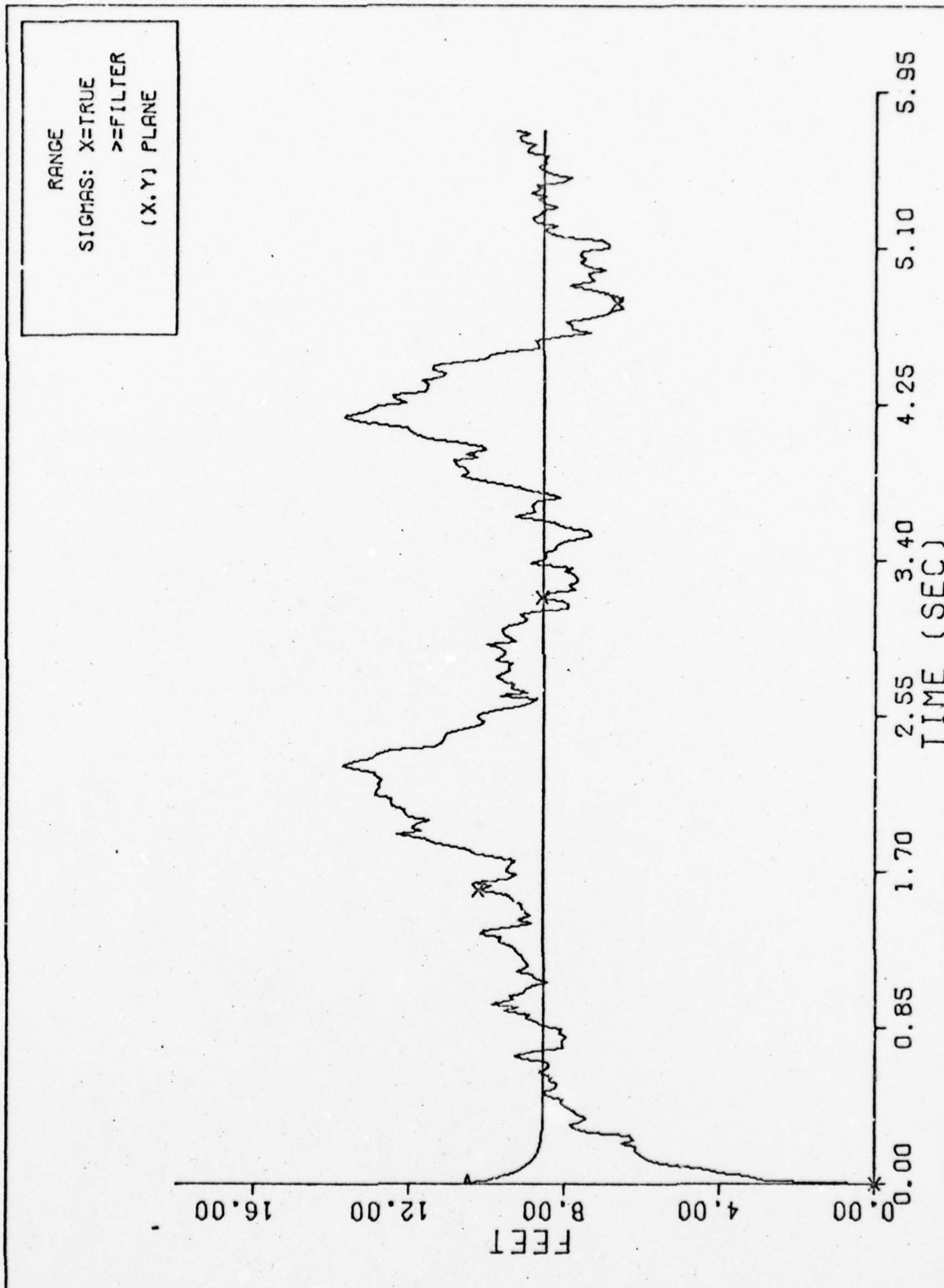



Fig. C-43 RANGE SIGMAS DOUBLE FILTER

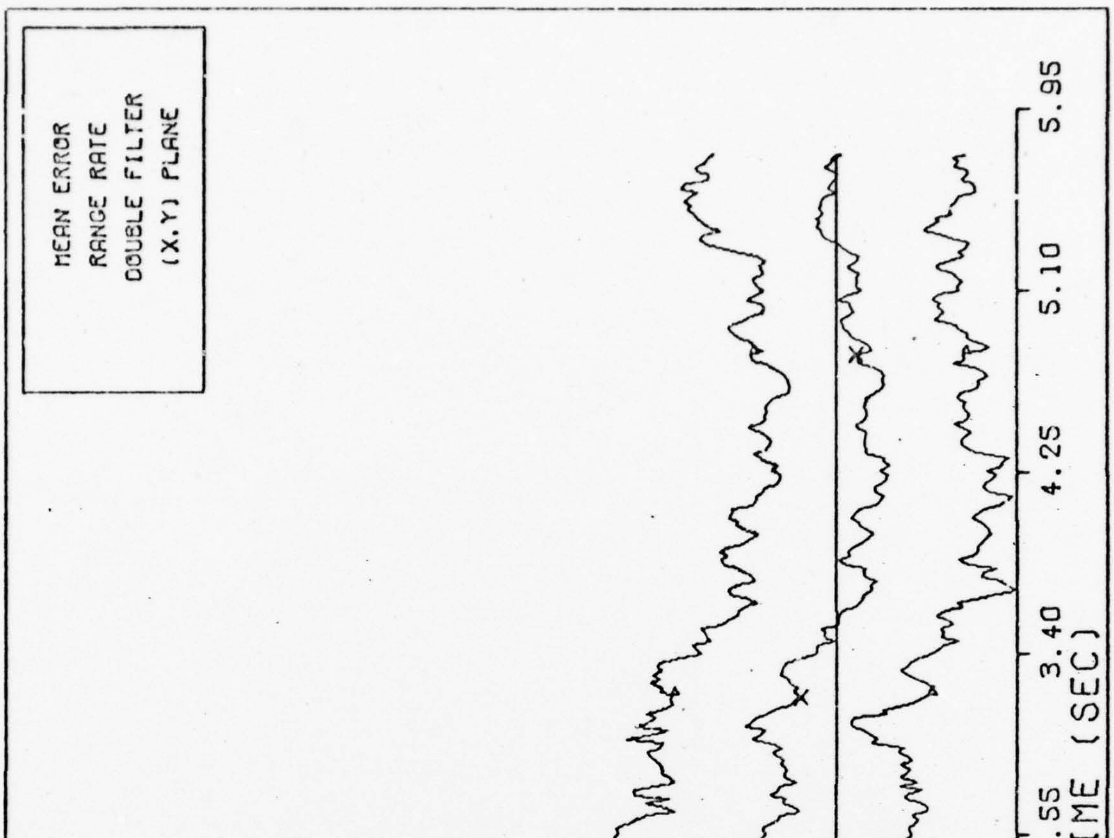



Fig. C-44 RANGE RATE DOUBLE FILTER

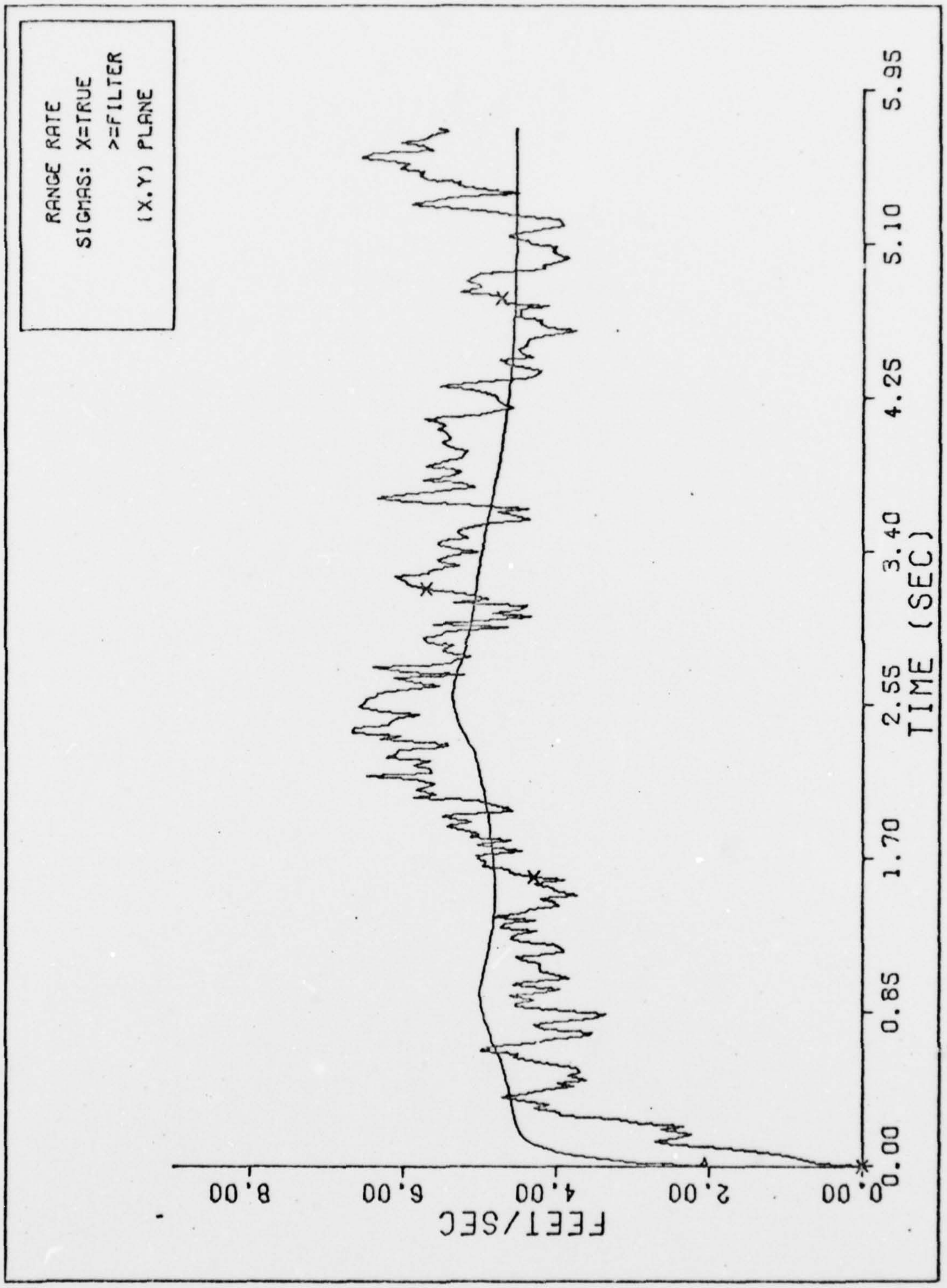



Fig. C-45 RANGE RATE SIGMAS DOUBLE FILTER

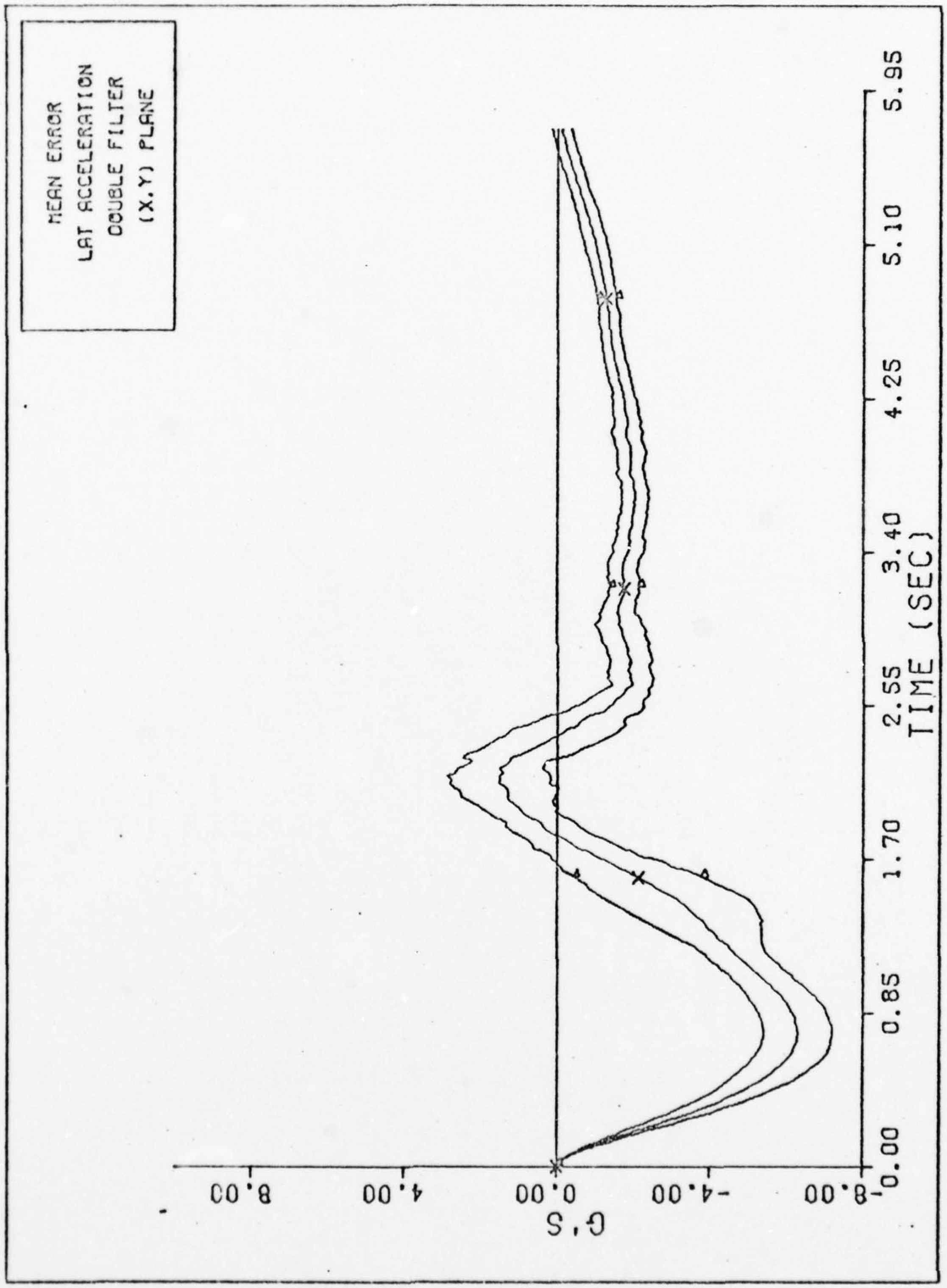



Fig. C-46 LAT ACCELERATION DOUBLE FILTER

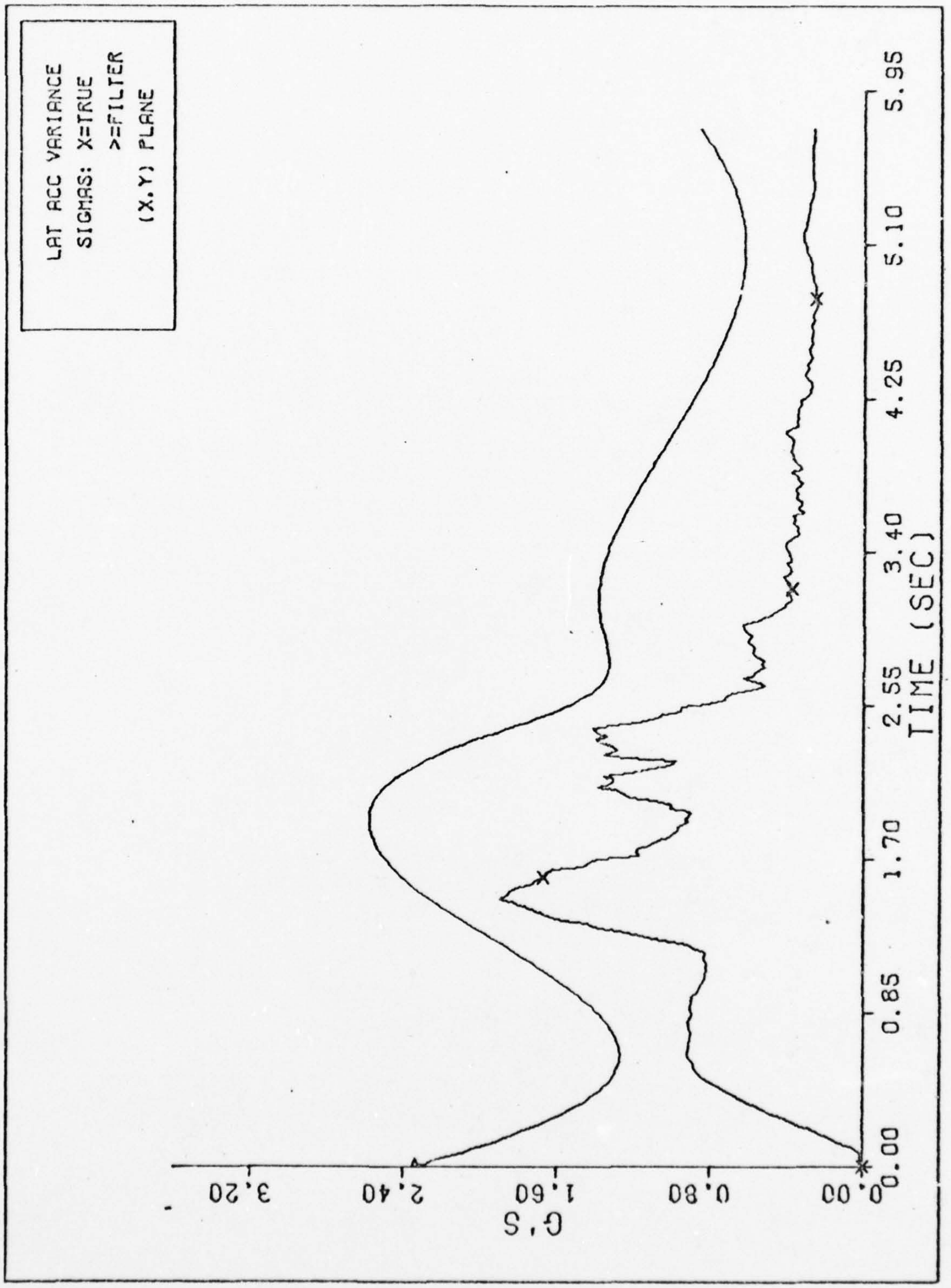



Fig. C-47 LAT ACCELERATION SIGMAS DOUBLE FILTER

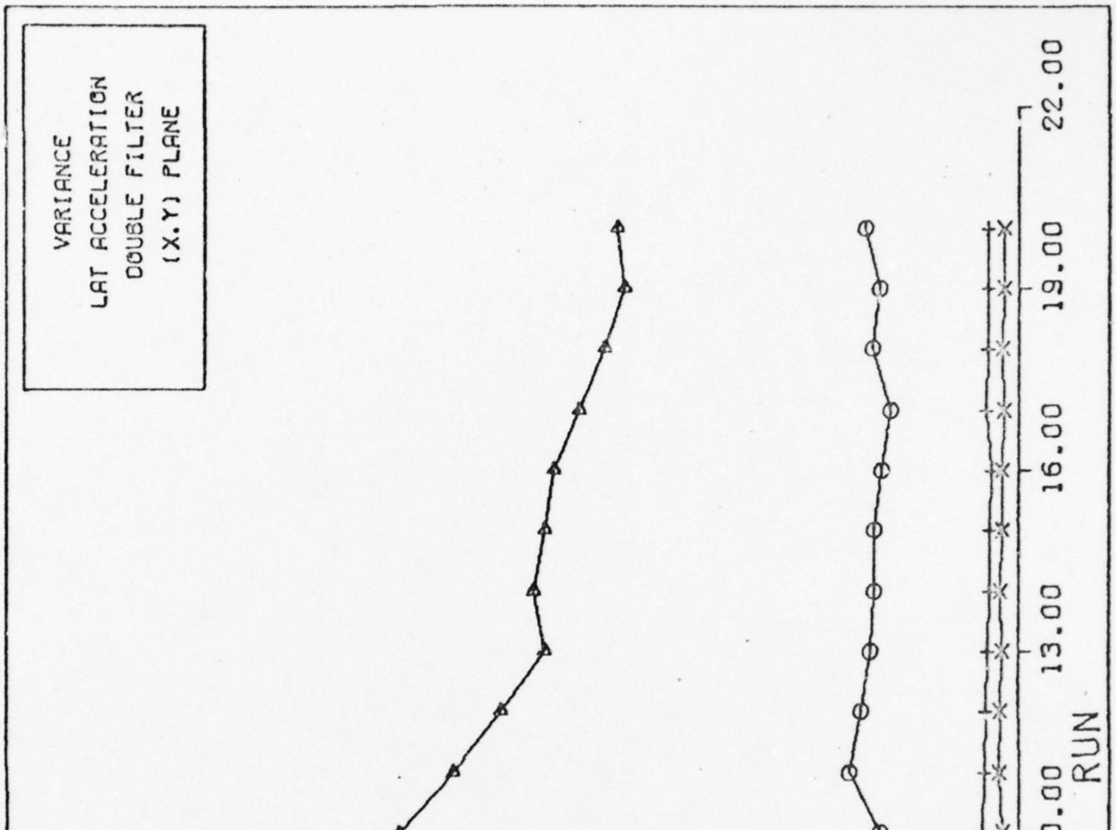



Fig. C-48 VARIANCE CONVERGENCE

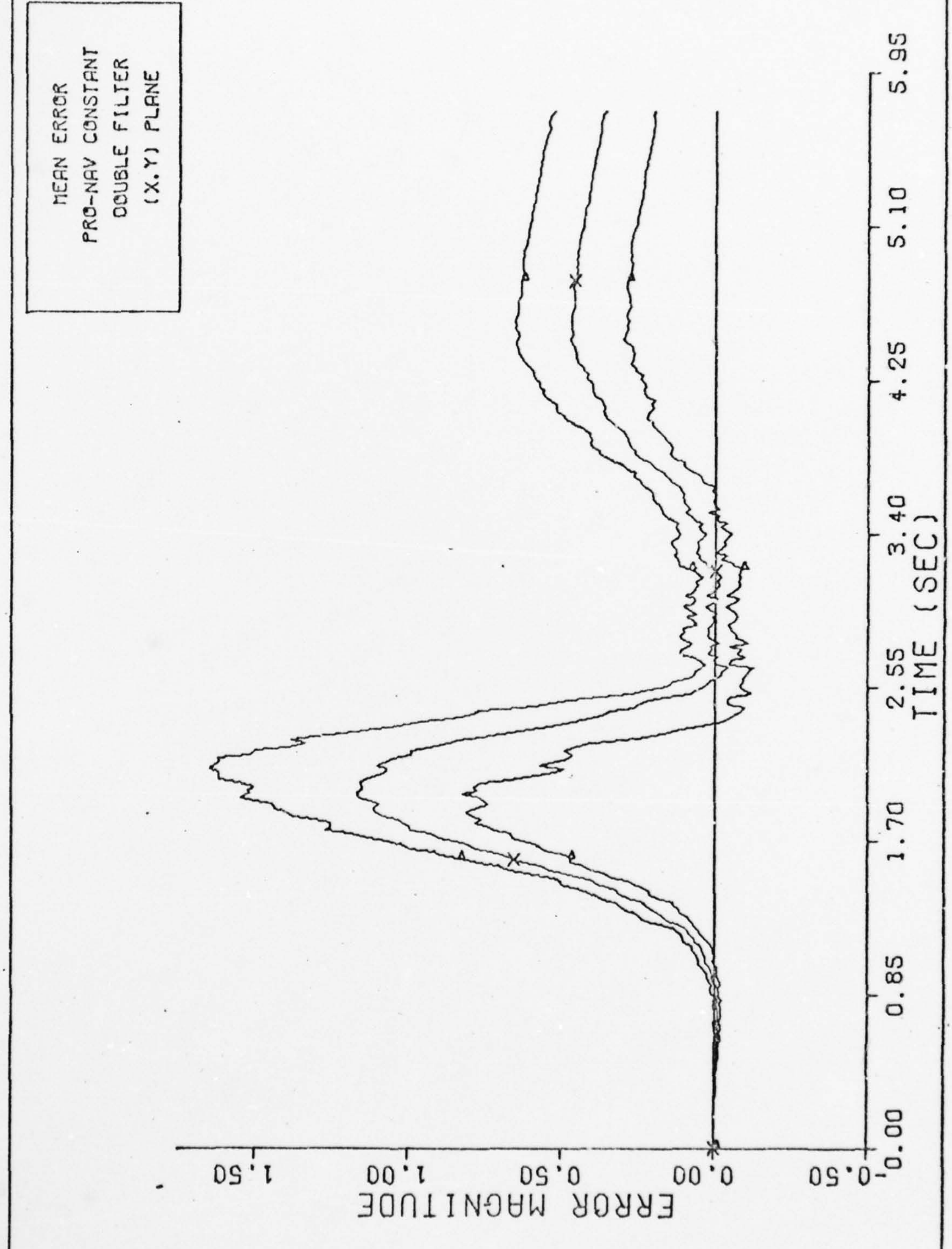



Fig. C-49 PRO-NAV CONSTANT DOUBLE FILTER

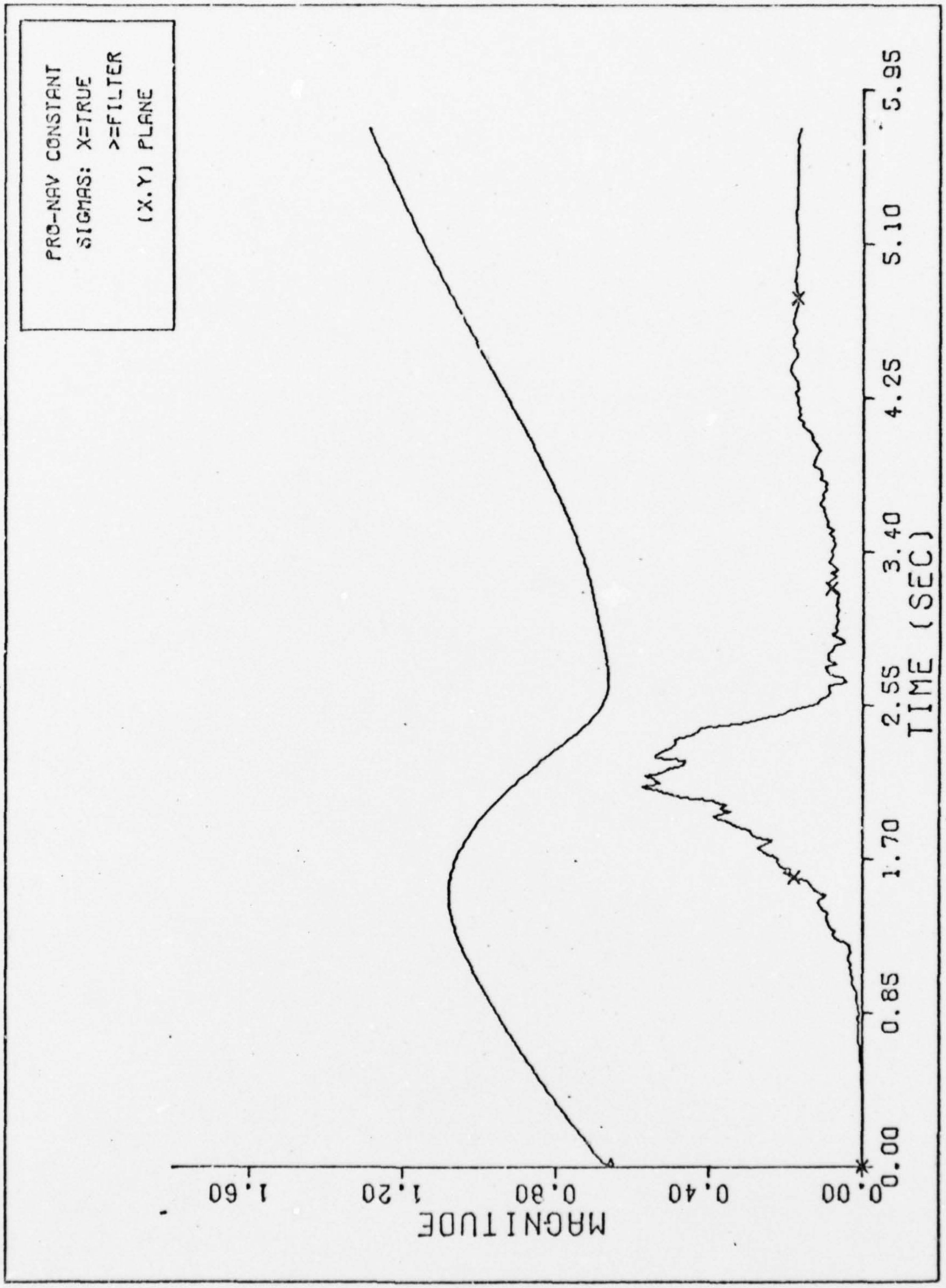



Fig. C-50 PRO-NAV CONSTANT SIGMAS DOUBLE FILTER

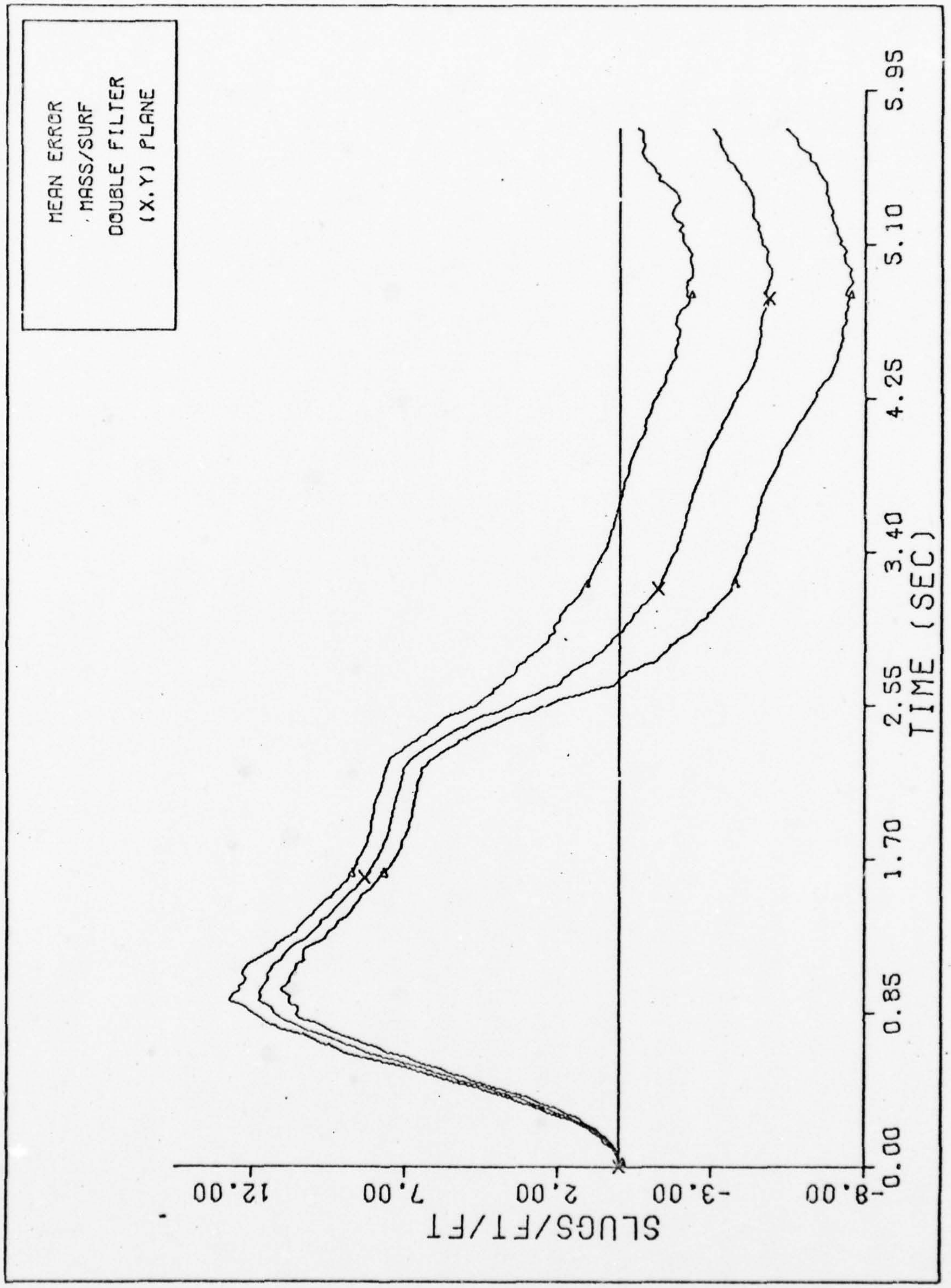



Fig. C-51 MASS/SURF DOUBLE FILTER

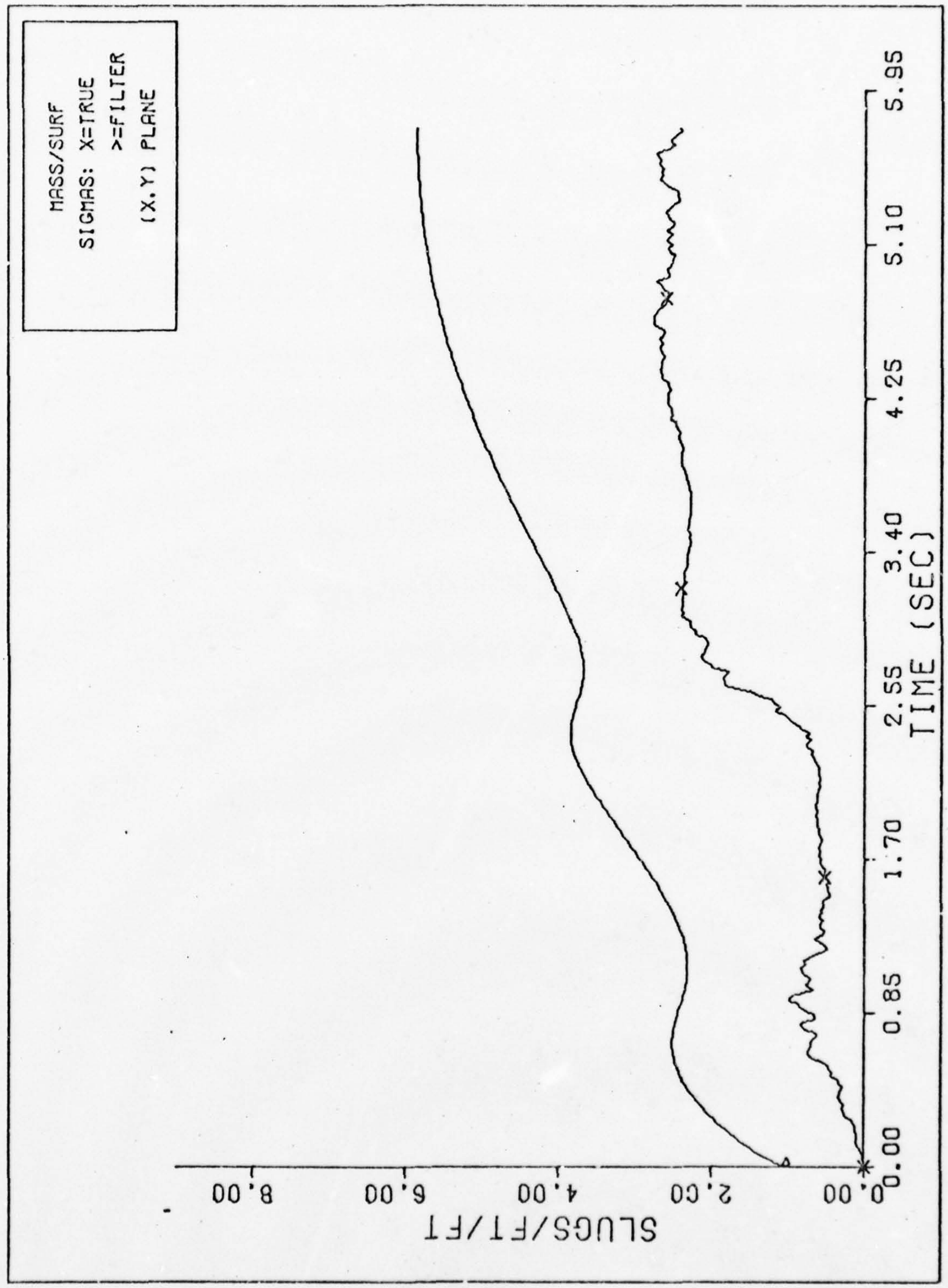



Fig. C-52 MASS/SURF SIGMAS DOUBLE FILTER

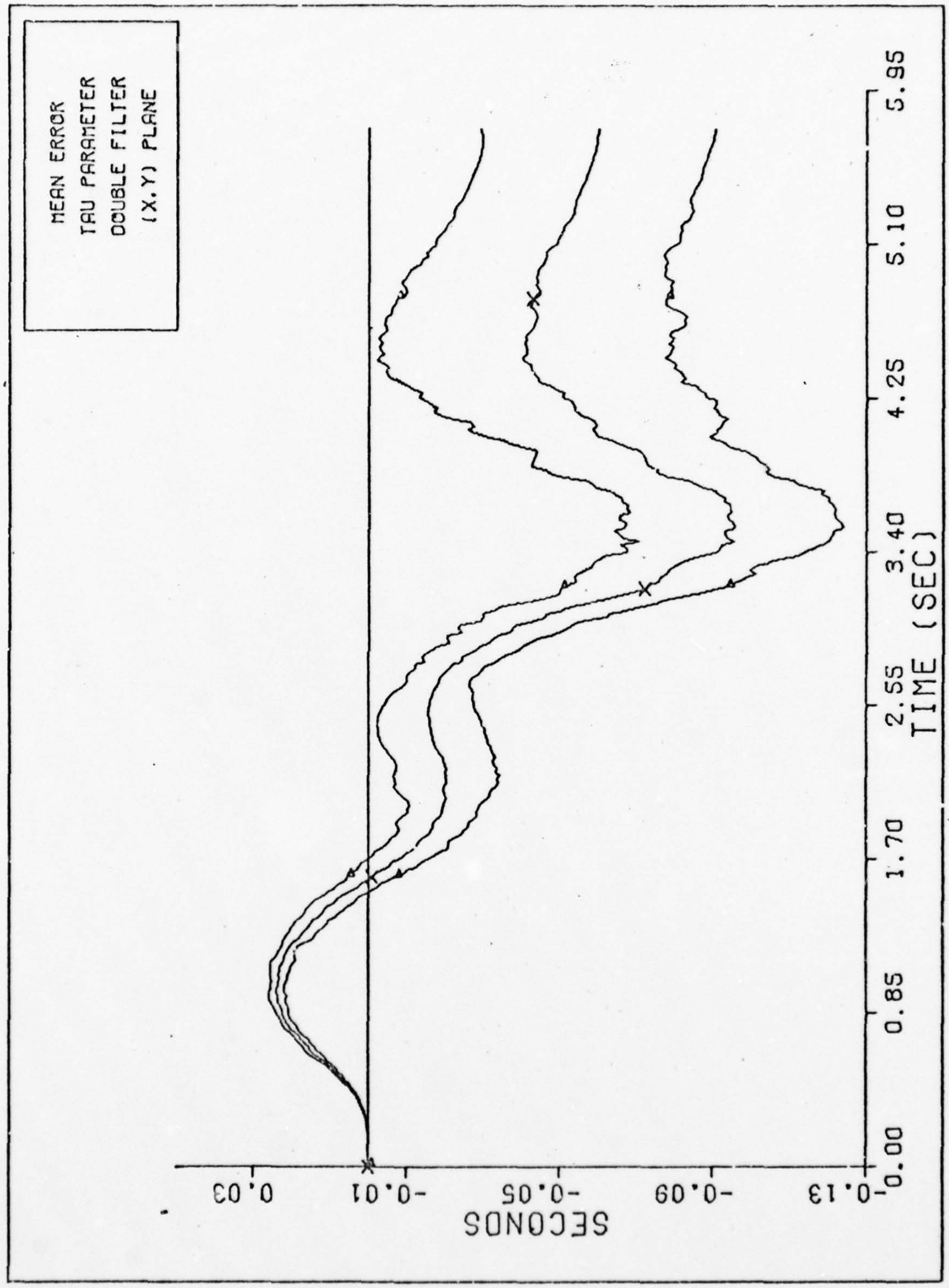



Fig. C-53 TAU PARAMETER DOUBLE FILTER

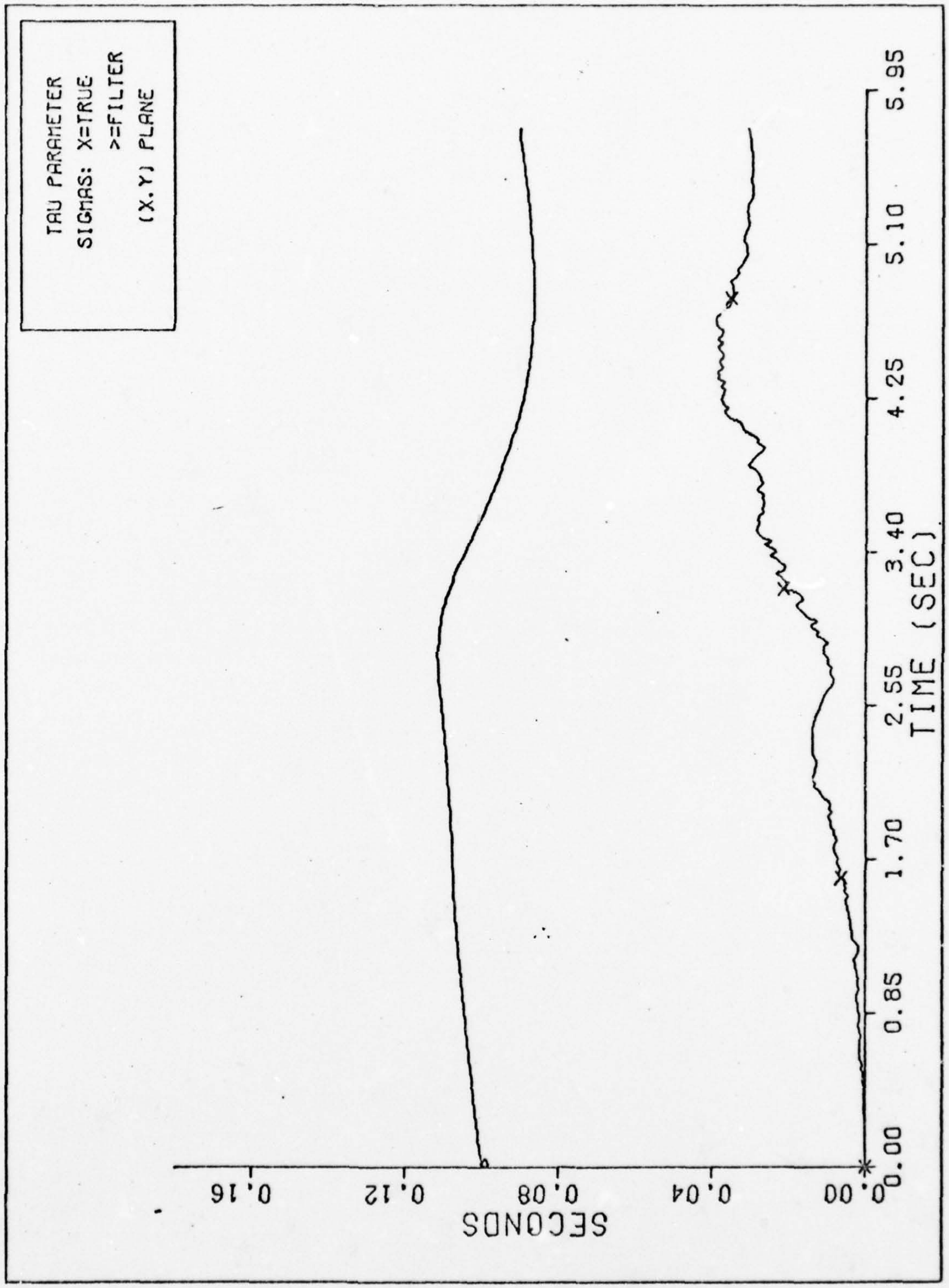



Fig. C-54 TAU PARAMETER SIGMAS DOUBLE FILTER

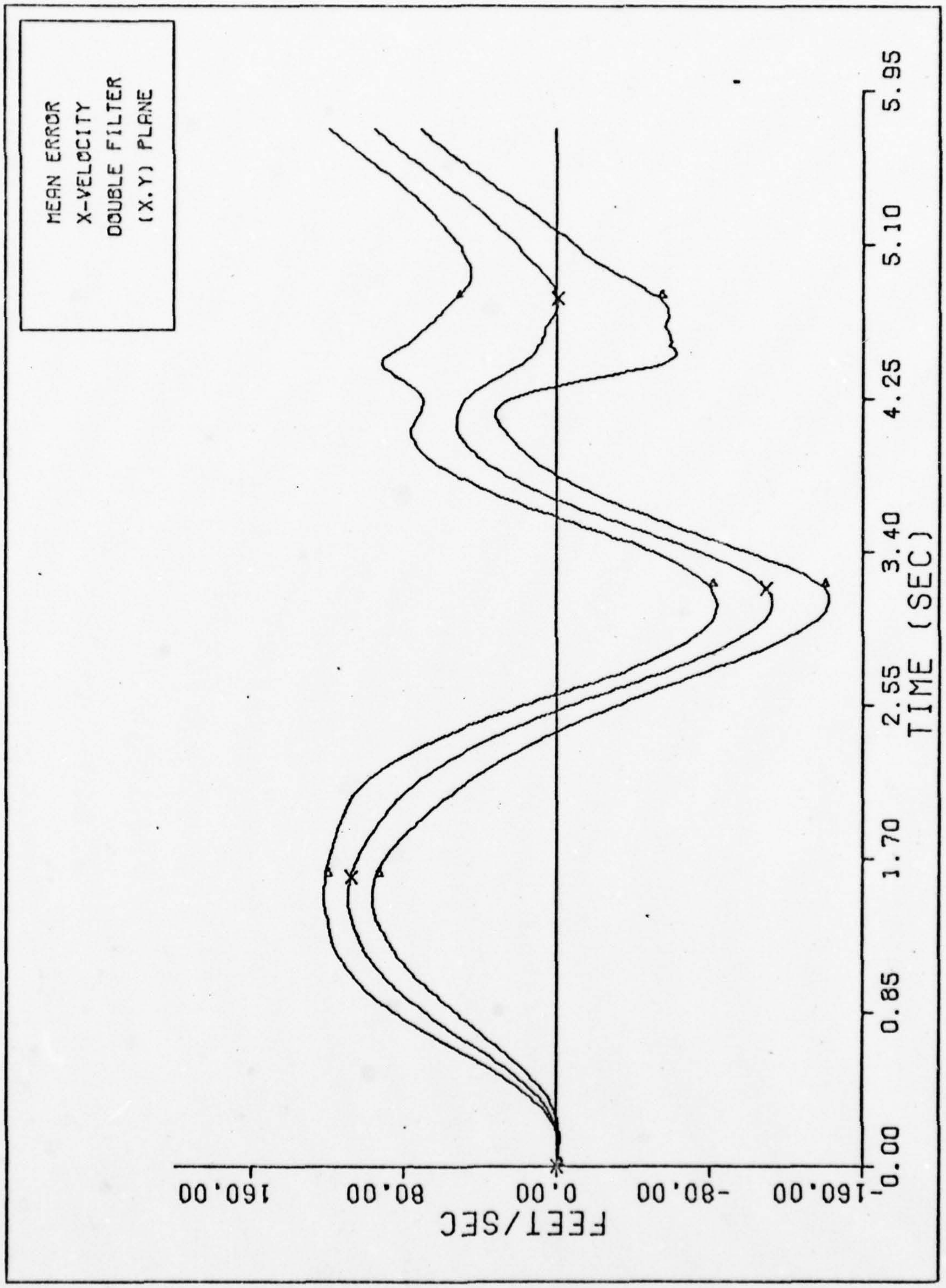



Fig. C-55 X-VELOCITY DOUBLE FILTER

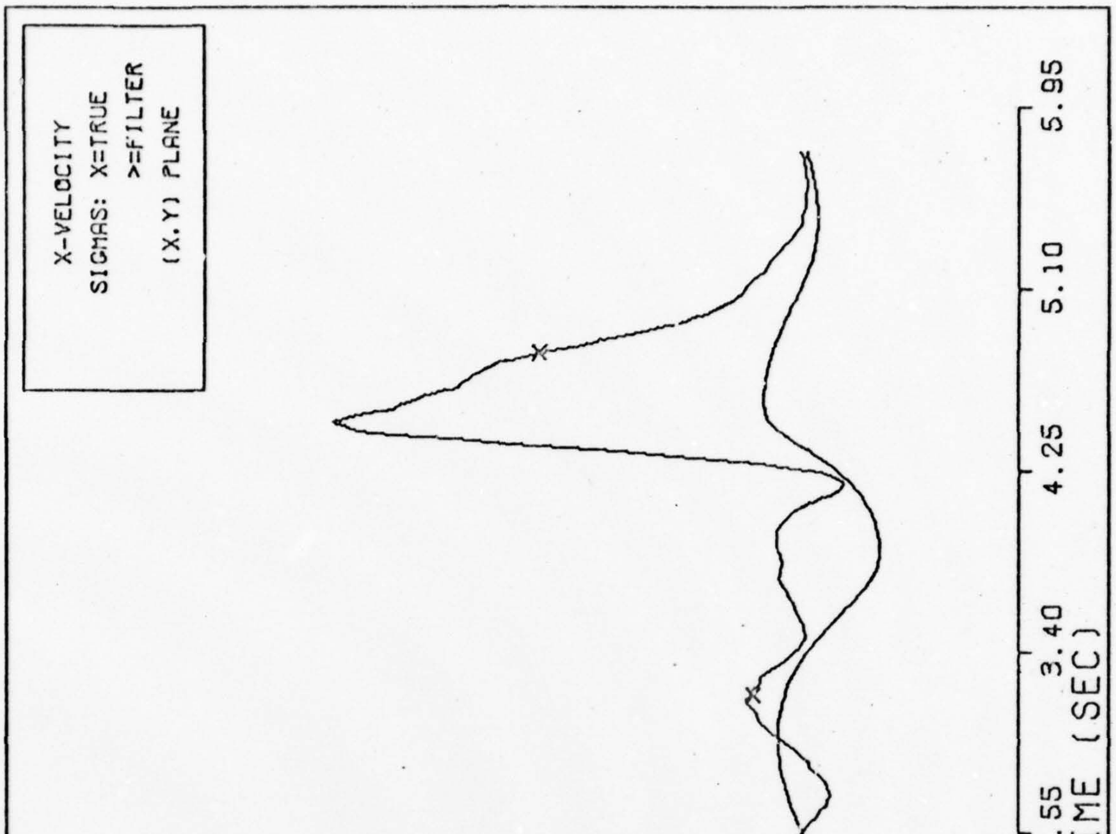



Fig. C-56 X-VELOCITY SIGMAS DOUBLE FILTER

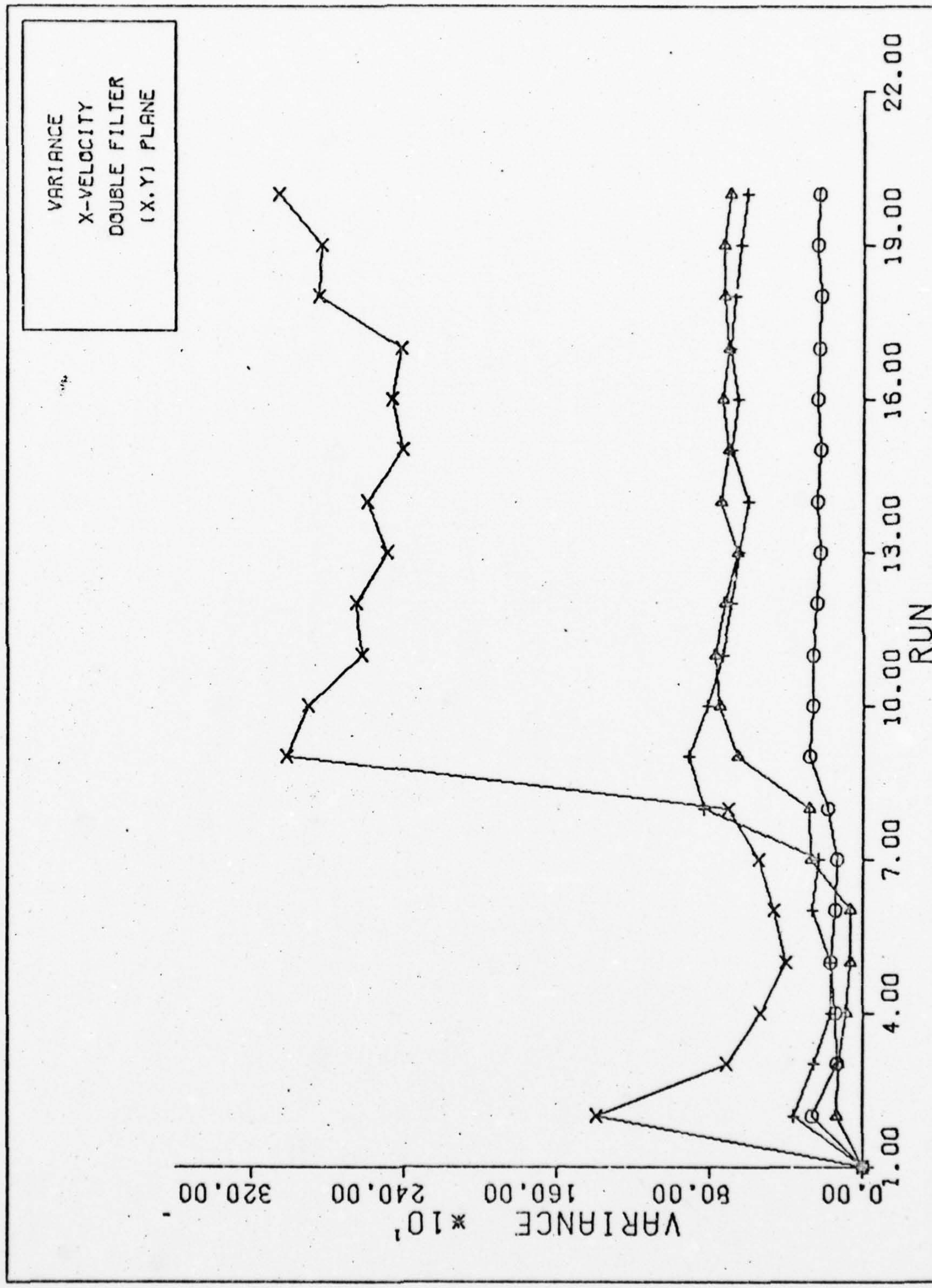



Fig. C-57 VARIANCE CONVERGENCE

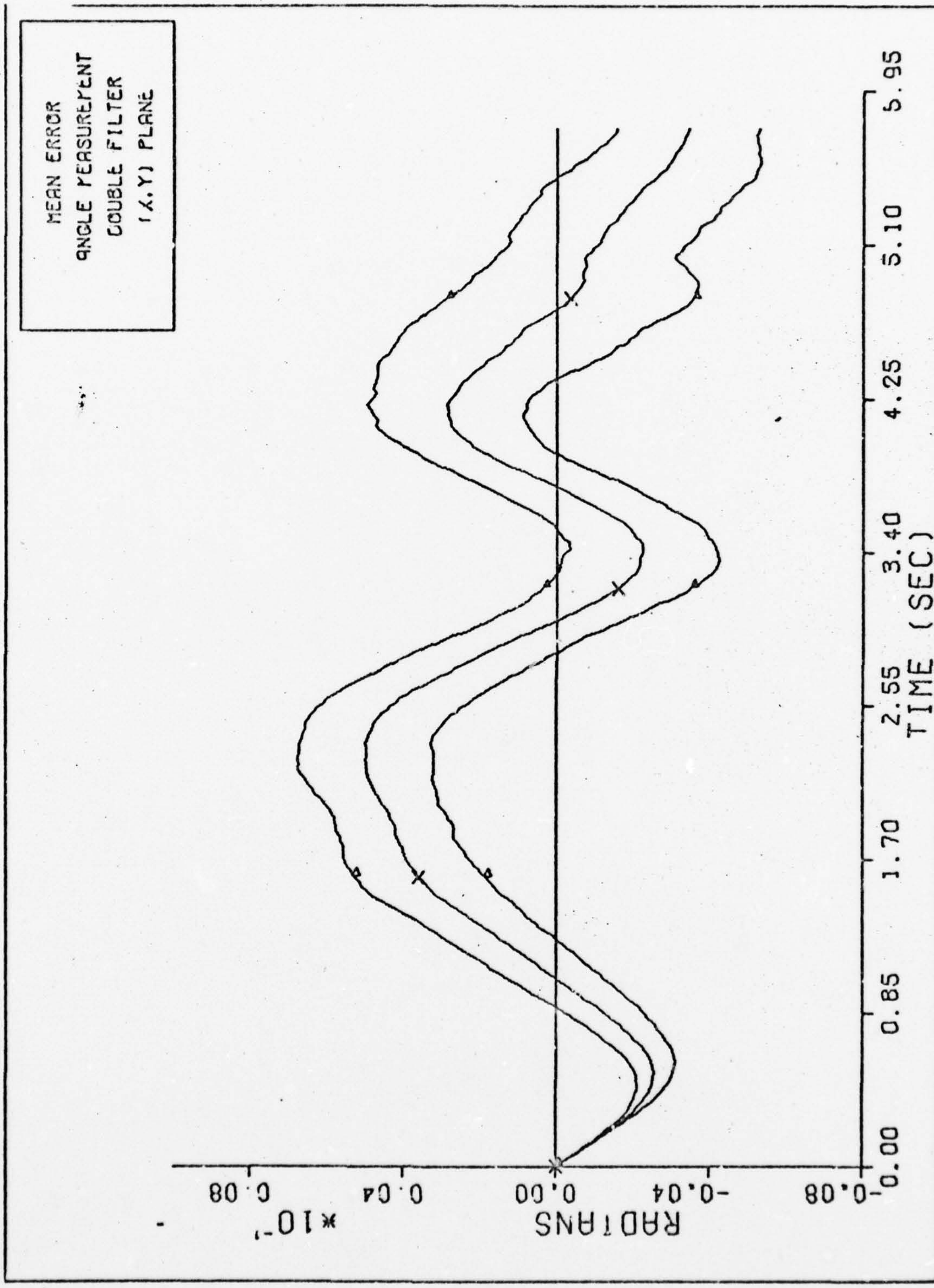



Fig. C-58 ANGLE MEASUREMENT DOUBLE FILTER

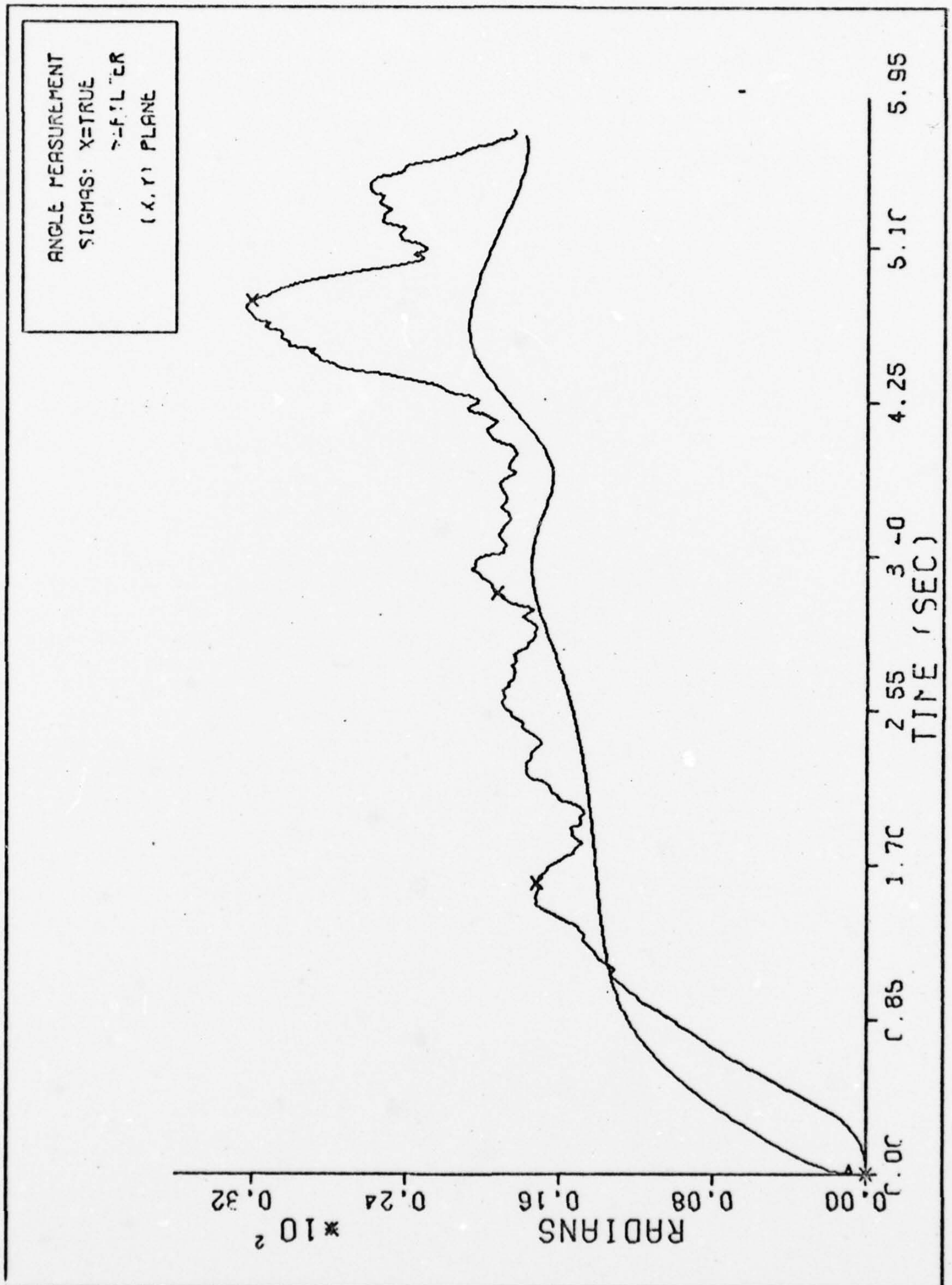



Fig. C-59 ANGLE MEASUREMENT SIGMAS DOUBLE FILTER

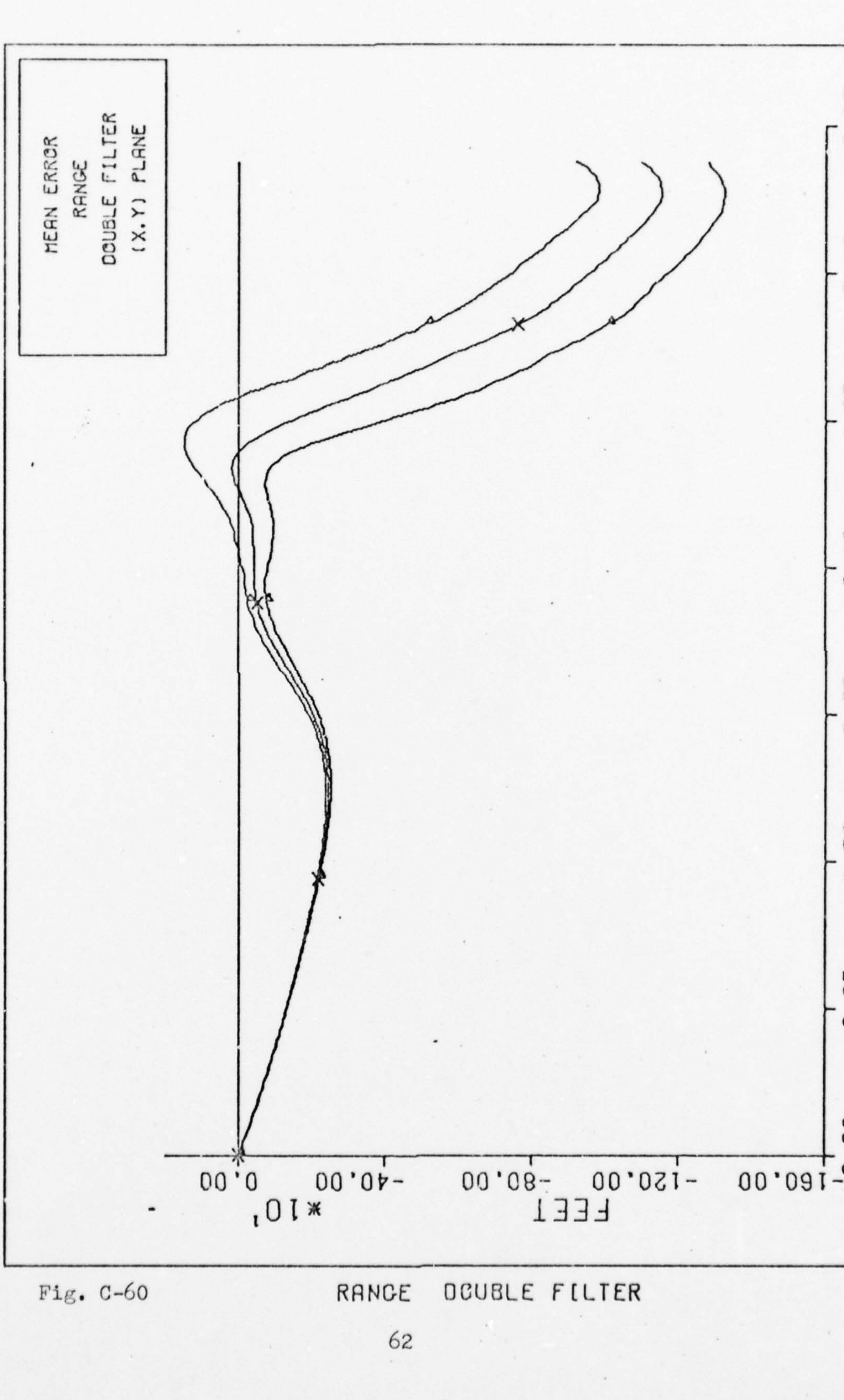



Fig. C-60 RANGE DOUBLE FILTER

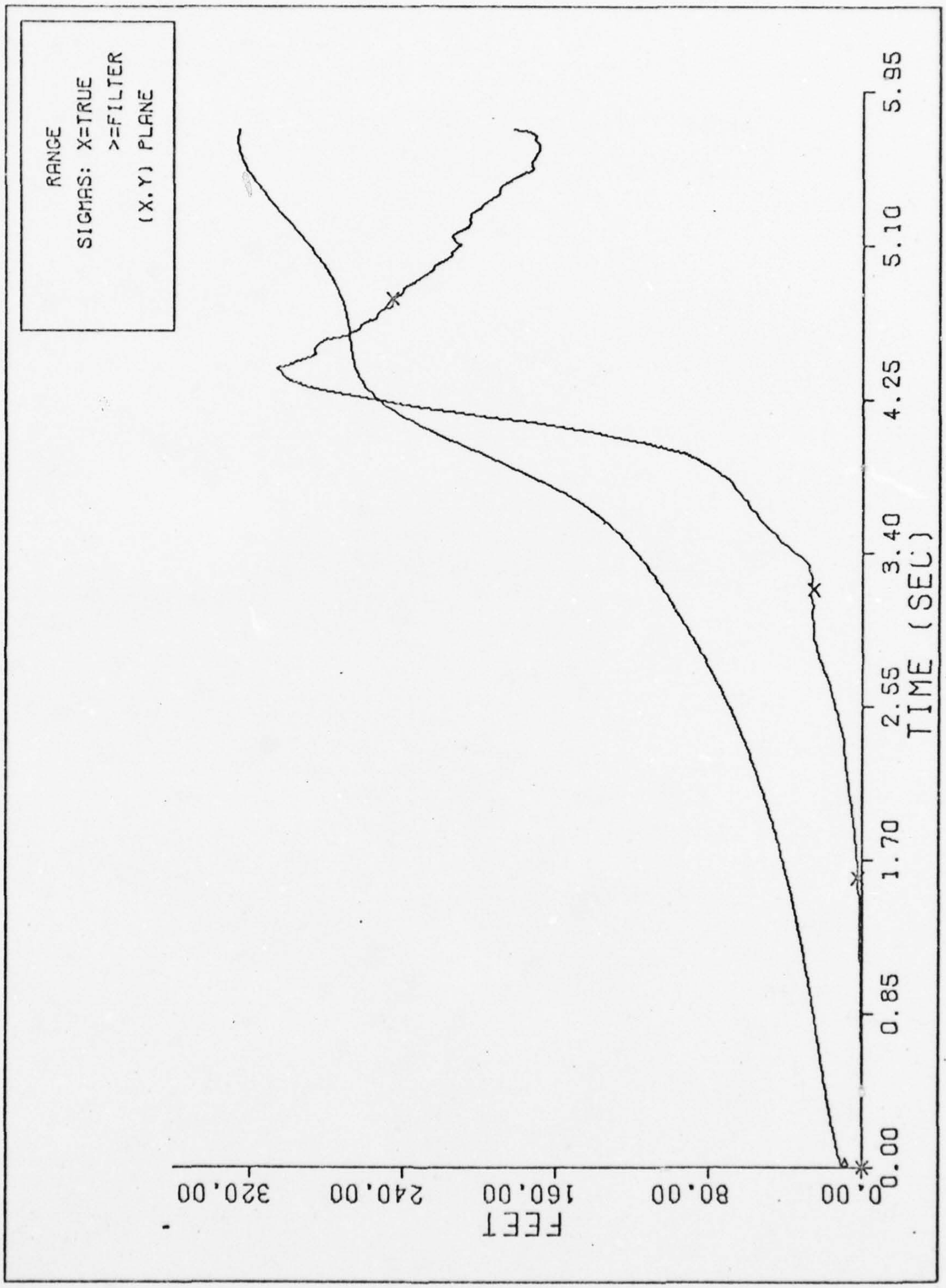



Fig. C-61 RANGE SIGMAS DOUBLE FILTER

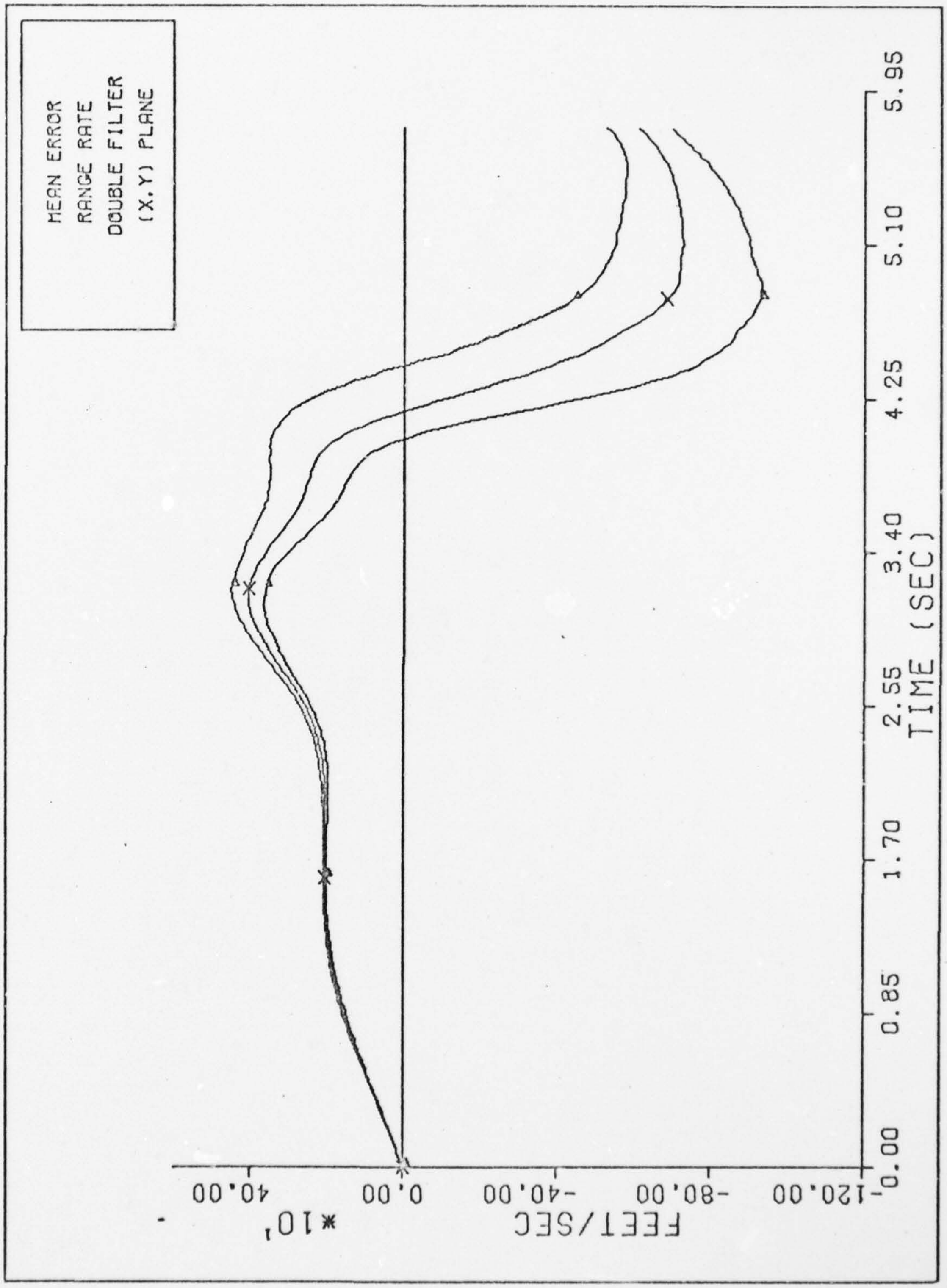



Fig. C-62 RANGE RATE DOUBLE FILTER

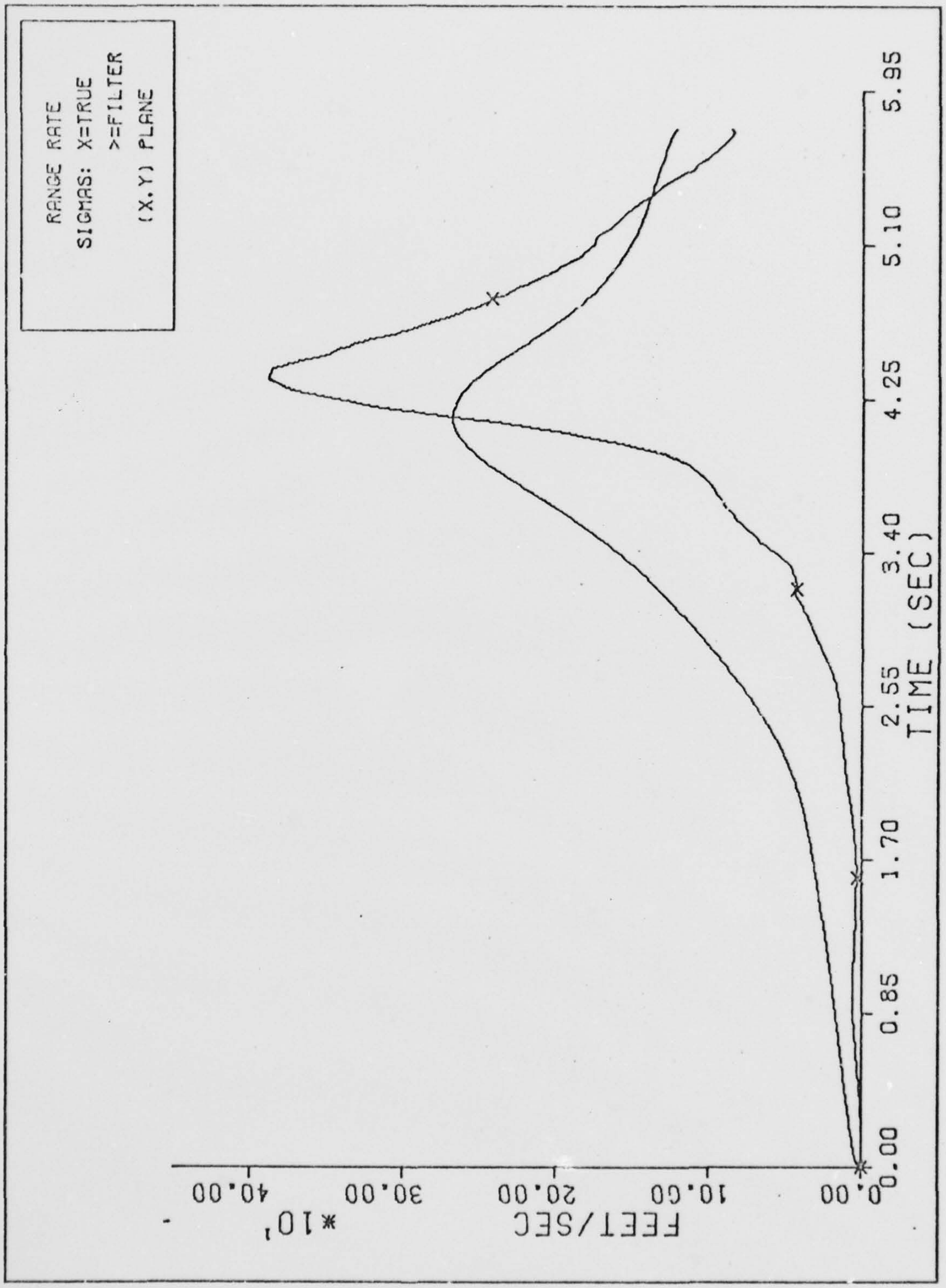



Fig. C-63 RANGE RATE SIGMAS DOUBLE FILTER

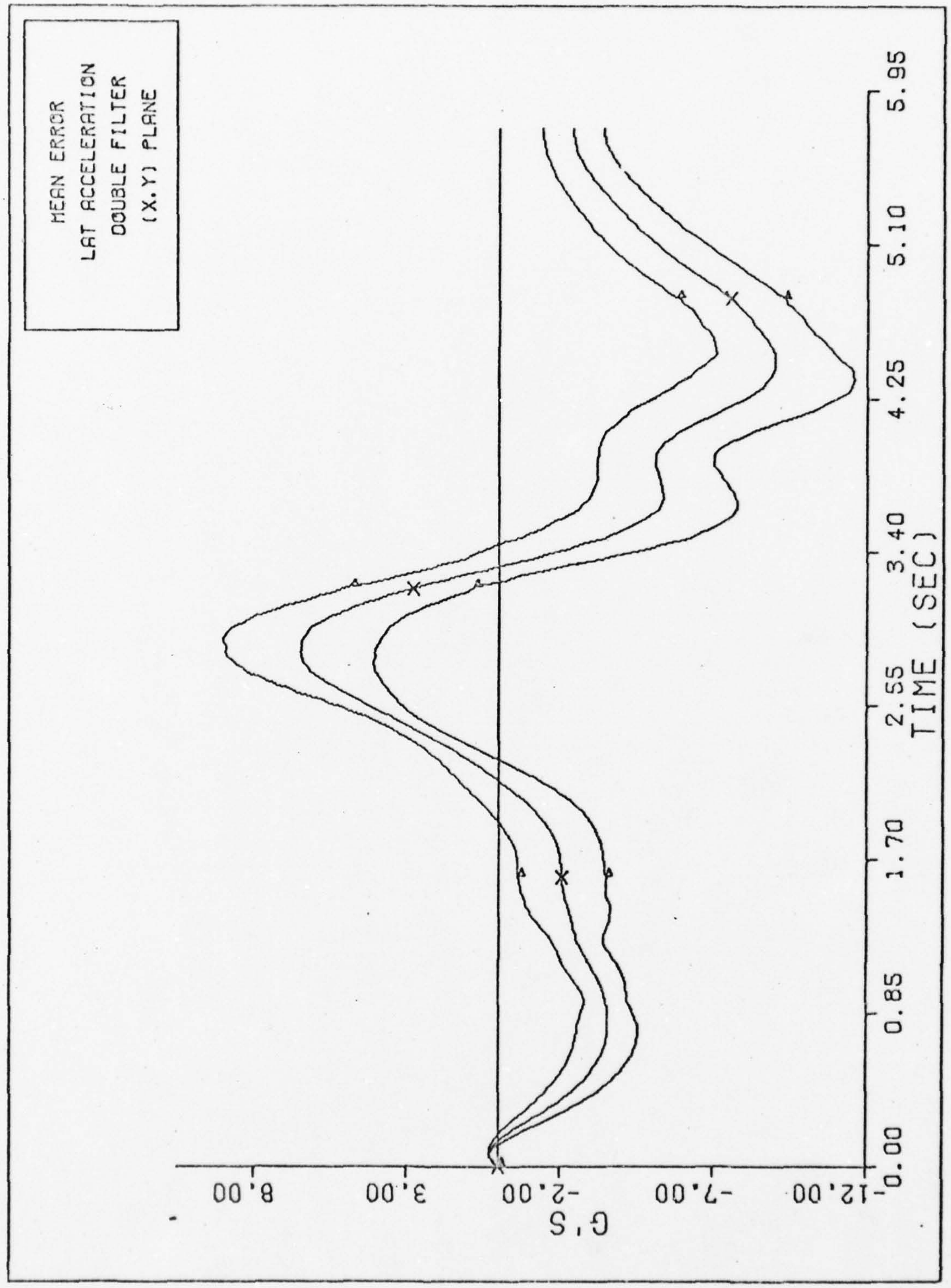



Fig. C-64 LAT ACCELERATION DOUBLE FILTER

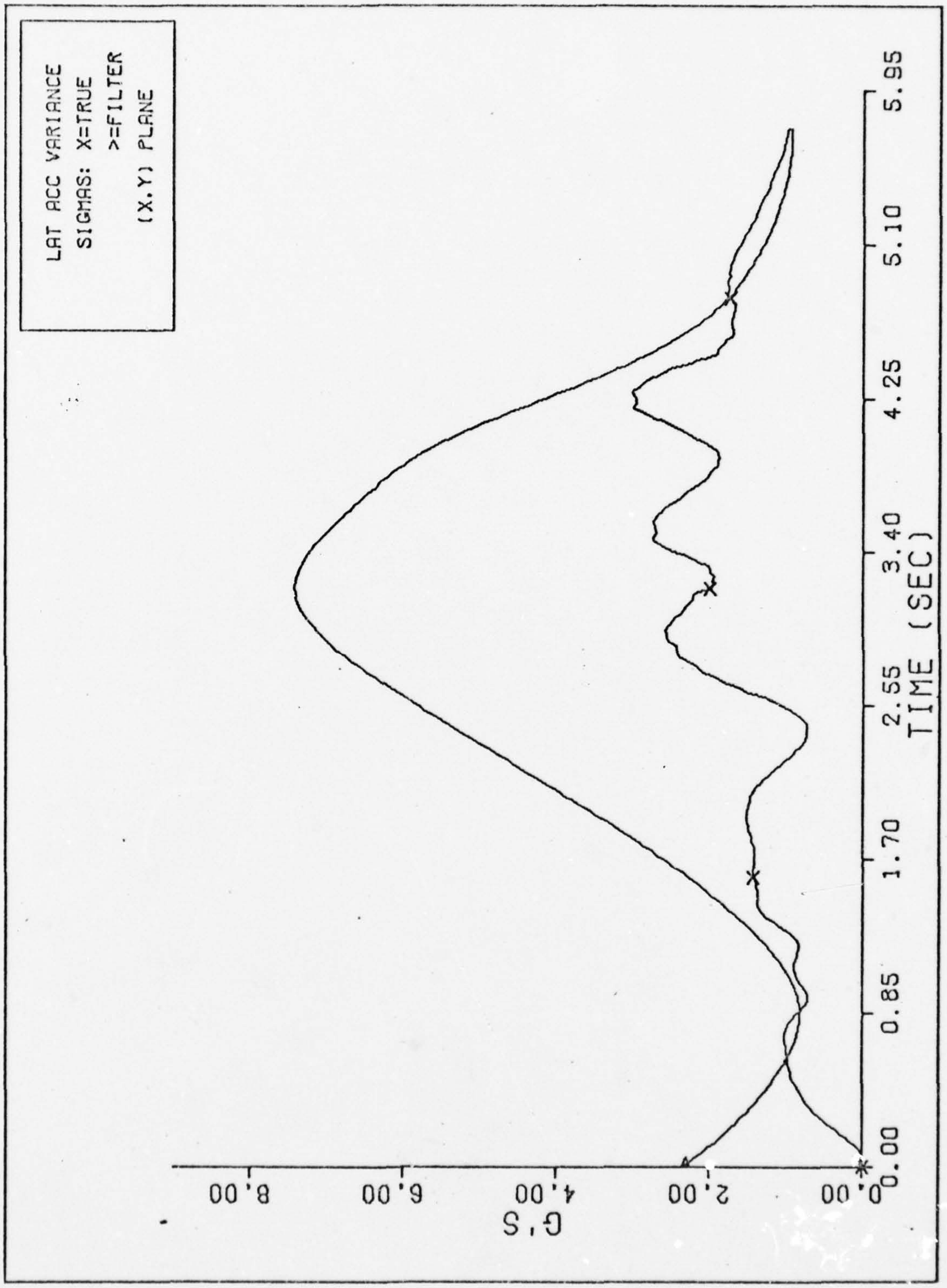



Fig. C-65 LAT ACCELERATION SIGMAS DOUBLE FILTER

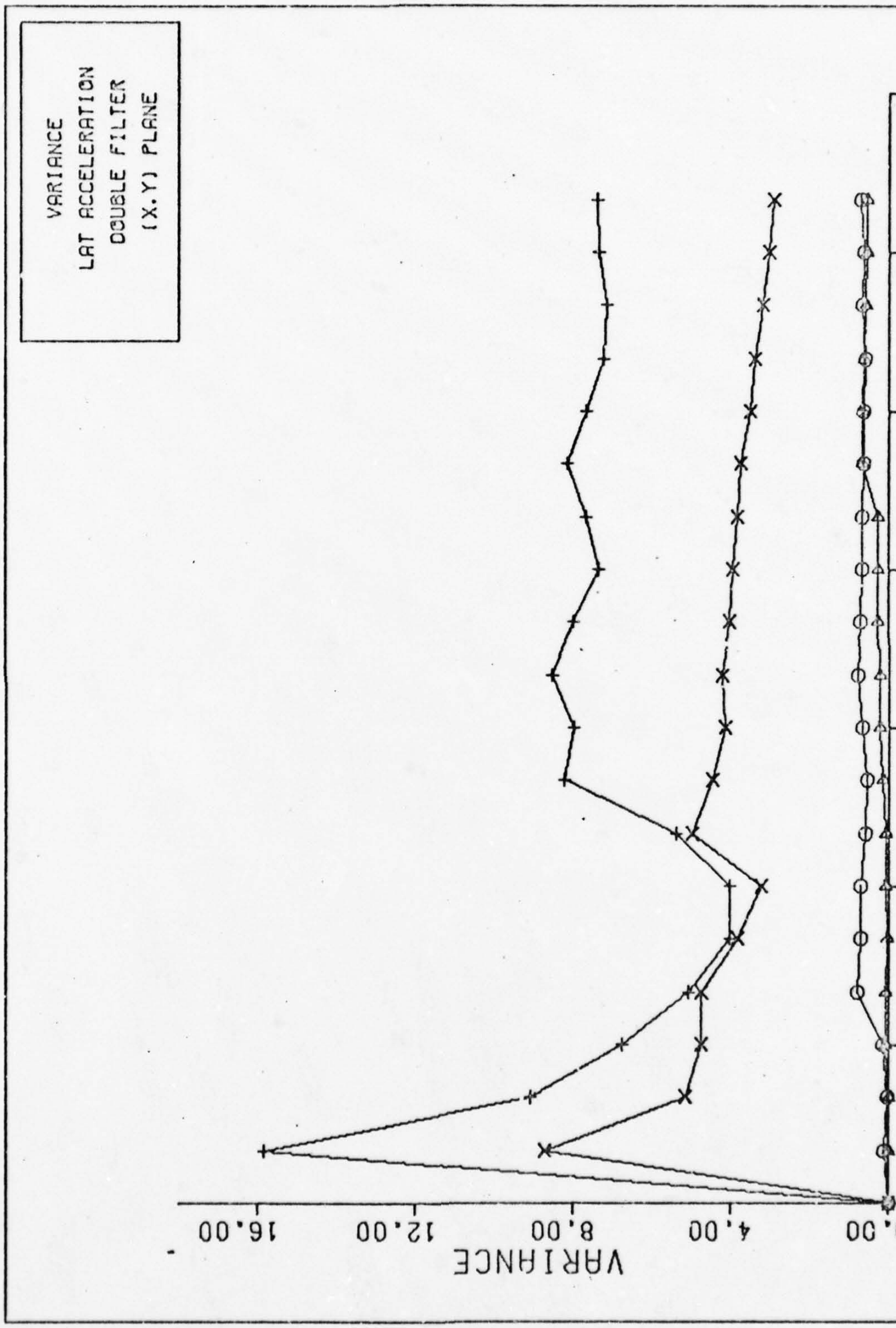



Fig. C-66 VARIANCE CONVERGENCE

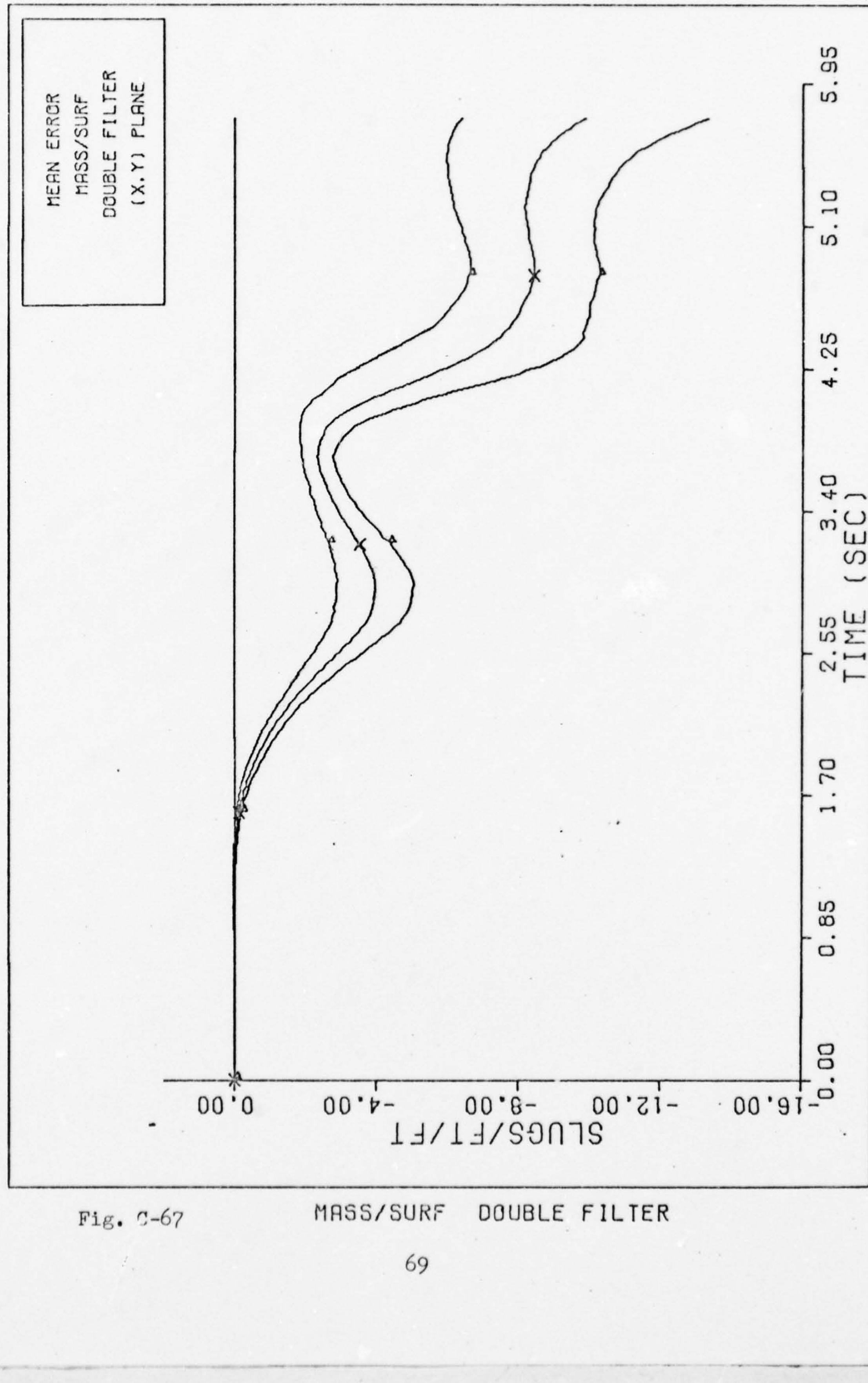



Fig. C-67 MASS/SURF DOUBLE FILTER

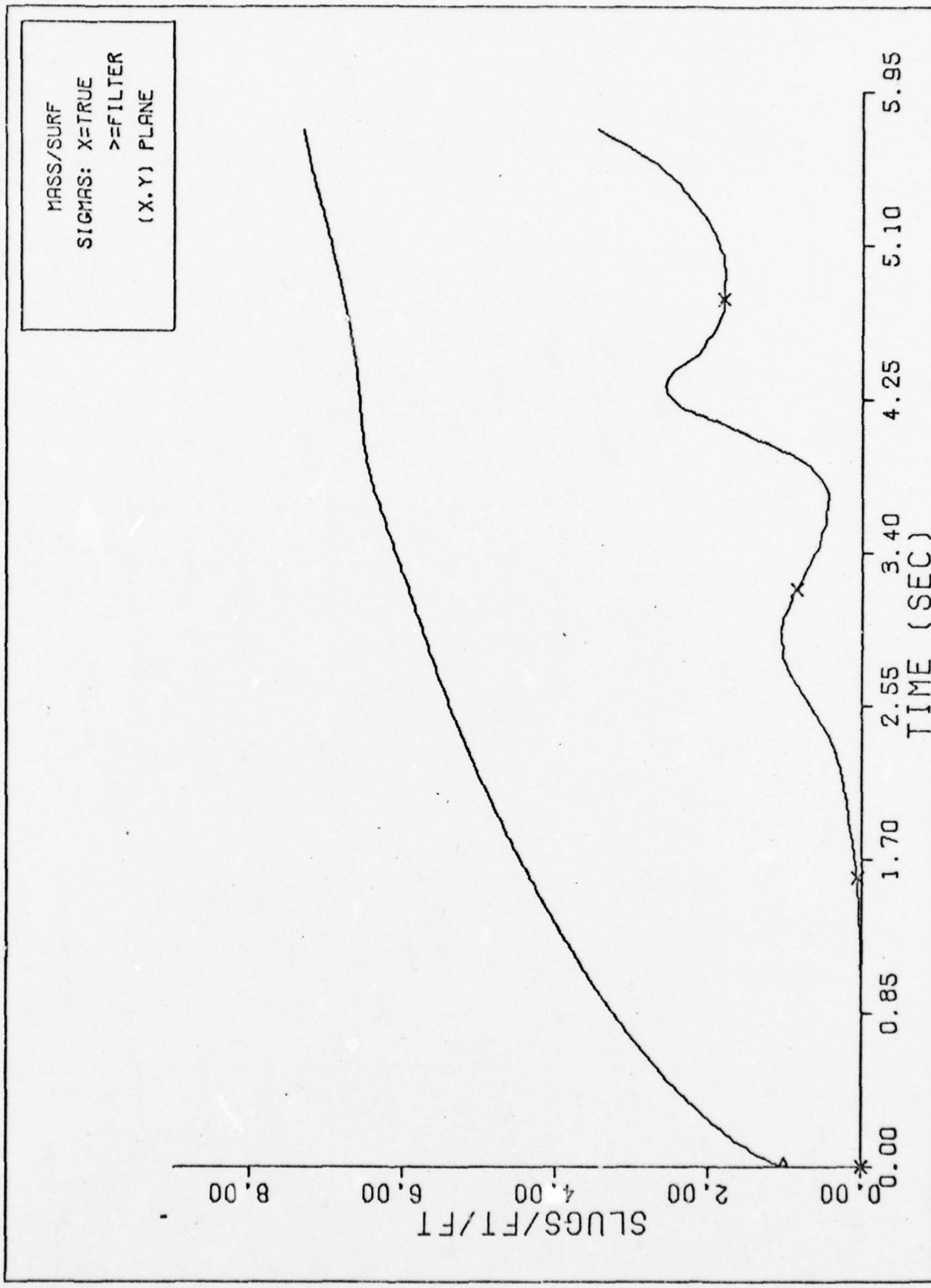



Fig. C-68 MASS/SURF SIGMAS DOUBLE FILTER

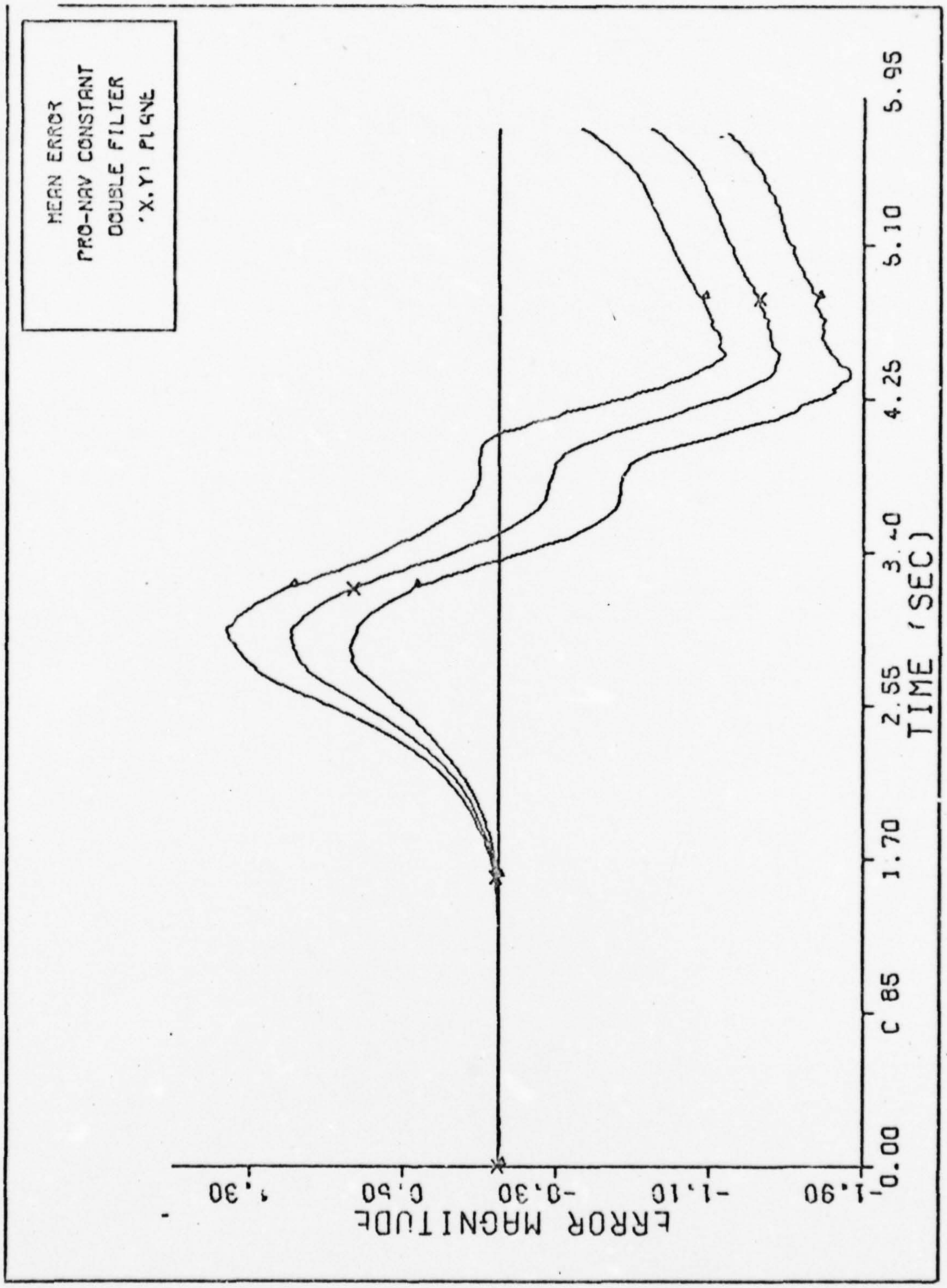



Fig. C-69 PRO NAV CONSTANT DOUBLE FILTER

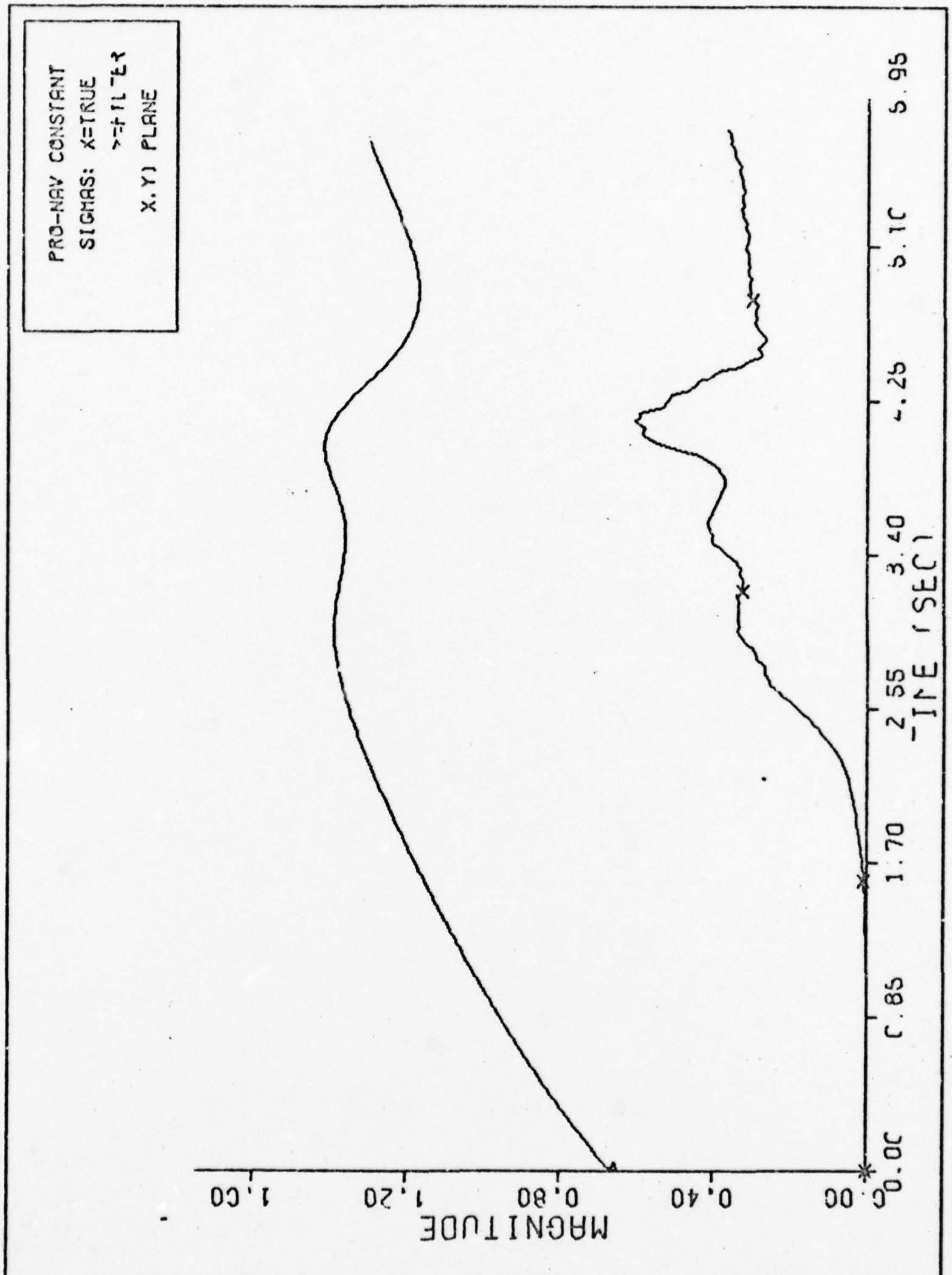



Fig. C-70 PRO NAV CONSTANT SIGMAS DOUBLE FILTER

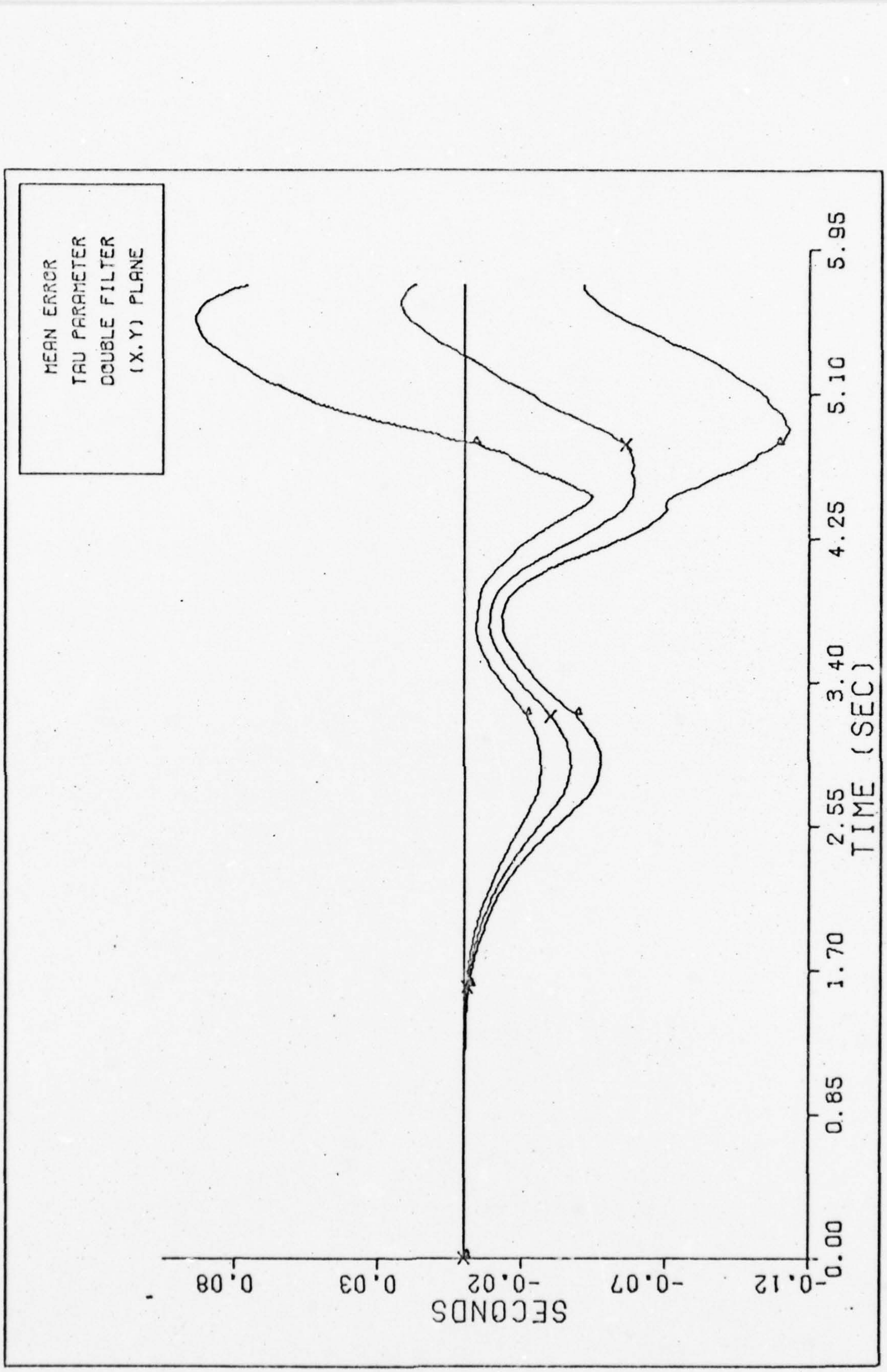



Fig. C-71 TAU PARAMETER DOUBLE FILTER

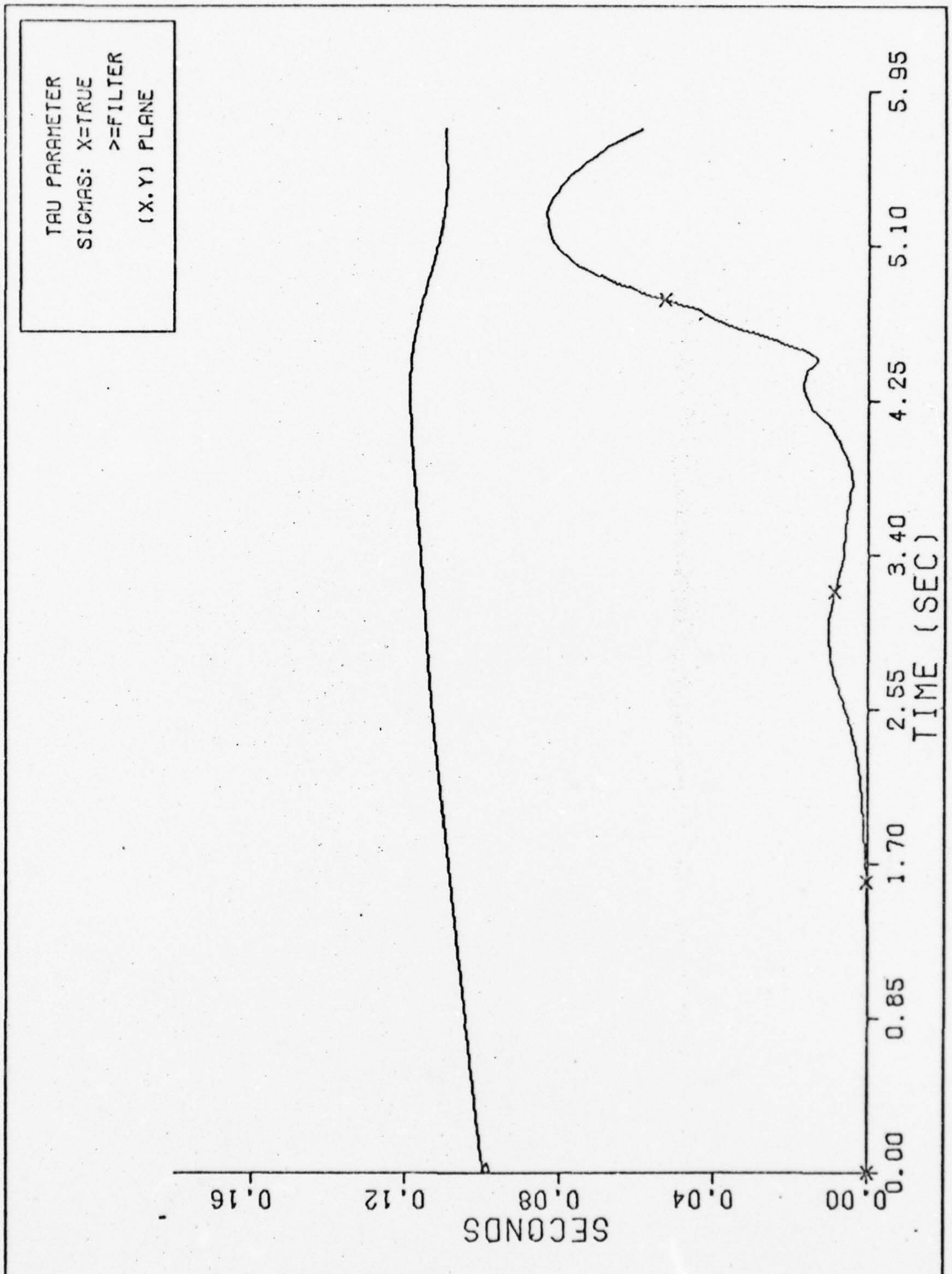



Fig. C-72 TAU PARAMETER SIGMAS DOUBLE FILTER

## Appendix D

Appendix D contains the computer software used to evaluate the 11 state filter. The only subroutine that is not contained in this appendix is Subroutine MINV - a matrix inversion subroutine. The software is organized as follows:

```
      PROGRAM EIGEN(INPUT,OUTPUT,TAPE4,TAPE5,TAPE6,TAPE8,TAPE9,DEBUG=OUT
     .PUT,TAPE2)
      COMMON/SITE/A1(11,11),A2(11,11),A4(11,6),A5(6,6),A7(11)
      COMMON/SITE1A/V(3),A(3)
      COMMON/SITE1/K1,K2,K3,K4,K5,K6,W,TS(11),TYME,DT,TRJCNT,PNK,MODE1,
     .TAP8
      COMMON/SITE2/COT2,SIT2,COA2,SIA2,COA1,SIA1,VM,Z,B,G,AD,COT1,SIT1
      COMMON/SITE3/Q(11,11),F(11,11), FS(11),FP(11,11)
      COMMON/SITE4/MEAS(6)
      COMMON/SITE5/FK(11,6)
      COMMON/SITE6/H(6,11),R(6,6)
      DIMENSION FVAR(11,288).VAR(11,288)
      DIMENSION FSDUM(11),FPDUM(11,11),RDUM(6,6),QDUM(11,11)
      DIMENSION OUT(23),VPT(20,4),VP2(20,4)
      DIMENSION XSUM(11,288),XSOS(11,288),FVARS(11,288),XM(11,288)
      INTEGER TRJCNT,RN
      LOGICAL TAP,TAP8,MODE1
      REAL MEAS,K1,K2,K3,K4,K5,K6
C                     *******************************              000300
C     PROGRAM AND INTEGRATION PARAMETERS                           000310
C     DT = PROGRAM INTEGRATION STEP SIZE
C     JJ = CONTROLS NUMBER OF SIMULATION RUNS BY # RUNS = 10 - JJ
C     MODE1 FALSE = INITIALIZES TRAJ
C     N = DIMENSION OF FSDUM, FPDUM, QDUM, AND RDUM
C     PNK = PRO NAV CONSTANT
C     TAP FALSE PRINTS STATE ERRORS ON TAPE 9
C     TAP8 TRUE = PRINTS MISSILE DATA ON TAPE4, TAPE5, AND TAPE6
C     TRJCNT  CONTROLS INTEGRATION STEP SIZE OF TRAJ BY DX = DT/TRJCNT
C                     *******************************              000320
C                     *******************************              000350
      DT=.02
      G=.707*32.174*1.E-3
      B=1.017
      Z=1.066E5
      N=11
      CALL  REED(FSDUM,FPDUM,RDUM,QDUM,N,RN)
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
      CALL RANSET(121412024)
      IF(RN.EQ.2) CALL RANSET(.27327623703++)
C                            ******************************
C     INITIALIZATION FOR 20 RUNS
C                            ******************************
      TAP=.TRUE.
      TAP5=.FALSE.
      IF(TAP)GO TO 332
      PRINT(3,358)
      PRINT(3,357)
  357 FORMAT(T3,"TIME",T14,"THETA 1",T25,"THETA 2",T36,"W 1",T47,"W 2",
     .T58,"RANGE",T69,"R-DOT",T80,"AL-1",T91,"AL-2",T102,"PRO NAV",T113,
     ."TAU",T124,"M/S",//)
  358 FORMAT(T60,"STATE ERRORS",//)
  332 CONTINUE
      DO 348 I=1,238
      DO 348 J=1,11
      XSUM(J,I)=0
      FVARS(J,I)=0
  348 XSQS(J,I)=0
      JJ=1
C                            ******************************
C     TRACKER AND MISSILE INITIALIZATION PARAMATERS
C                            ******************************
      TRJCNT=5
      PNK=4.5
C                            ******************************
C     INITIALIZATION VALUES FOR TRACKER MEASUREMENT NOISES
C
C              CORR0  (ANGLE MEAS. SCINTILLATION NOISE)
C              CORR1  (ANGLE MEAS. THERMAL NOISE)
C              CORR2  (RANGE MEAS. SCINTILLATION NOISE)
C              CORR3  (RANGE MEAS. THERMAL NOISE)
C              CORR4  (RANGE RATE MEAS. SCINTILLATION NOISE)
C              CORR5  (RANGE RATE THERMAL NOISE)
C              CORR5  (ANGLE RATE SCINTILLATION NOISE)
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
C             CORR7  (ANGLE RATE THERMAL NOISE)
C                        ******************************                    000790
      CORR0=0.000475175                                                    000800
      CORR1=0.00125                                                        000810
      CORR2=2.82843                                                        000820
      CORR3=11.7                                                           000830
      CORR4=0.5464                                                         000840
      CORR5=7.                                                             000850
      CORR6=.000045
      CORR7=.001745
C                        ******************************
C     INITIALIZATION FOR EACH RUN
C                        ******************************
999   CONTINUE                                                             000460
      MODE1=.FALSE.
      ITIME=0                                                              000300
      TYME=0
      X0=X1=X2=X3=X4=X5=0.0                                                000851
C                        ******************************                    001100
C     TRAJECTORY INITIALIZTION                                             001110
C                        ******************************                    001120
      CALL TRAJ
      MODE1=.TRUE.
C                        ******************************                    000860
C     INITIALIZTION OF FILTER STATE, COVARIANCE, Q, R, AND H MATRIX        000870
C                        ******************************                    000880
      DO 9 J=1,11
      DO 8 I=1,6
      H(I,J)=0                                                             000910
8     CONTINUE                                                             000930
9     CONTINUE                                                             000920
      H(1,1)=H(2,2)=H(3,3)=H(4,4)=H(5,5)=H(6,6)=1
      DO 11 J=1,11
      DO 10 I=1,11
      F(I,J)=0                                                             000970
10    CONTINUE                                                             000990
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
11    CONTINUE
      DO 802 J=1,11
      FS(J)=TS(J)
      DO 801 I=1,11
      FP(I,J)=FPDUM(I,J)
      Q(I,J)=QDUM(I,J)
801   CONTINUE
  802 CONTINUE
      DO 804 J=1,6
      DO 803 I=1,6
      R(I,J)=RDUM(I,J)
803   CONTINUE
  804 CONTINUE
      IF(TAP)GO TO 500
      DO 300 I=1,11
      PRINT*,"Q(",I,",",I,")=",Q(I,I)
300   CONTINUE
      DO 301 I=1,6
      PRINT*,"R(",I,",",I,")=",R(I,I)
301   CONTINUE
C                         ******************************
C     RUN PORTION
C                         ******************************
500   CONTINUE
C                         ******************************
C     COMPUTATION OF FILTER ERRORS AND FILTER SIGMAS
C                         ******************************
      OUT(1)=TYME
      DO 17 I=2,12
      J=I-1
      OUT(I)=TS(J)-FS(J)
17    CONTINUE
      DO 18 I=13,23
      J=I-12
      IF(FP(J,J).LT.0.0) PRINT*,"P(+)=",FP(J,J)," AT T=",TYME," J=",J
      OUT(I)=SQRT(FP(J,J))
```

```
16     CONTINUE                                                          001390
       ITIME=ITIME+1                                                     000310
       IF(JJ.EQ.1)WRITE(8)OUT(1)
       IF(TAF)GO TO 333
       WRITE(9,370)(OUT(I),I=1,12)
       WRITE(9,371)(OUT(I),I=13,23)
  370  FORMAT(T3,12(G9.3,2X))
  371  FORMAT(T12,11(G9.3,2X),/)
  333  CONTINUE
C                       ******************************
C      COMPUTATION OF RUN STATISTICS
C                       ******************************
       DO 310  J=1,11
       XSUM(J,ITIME)=XSUM(J,ITIME)+OUT(J+1)
       XSQS(J,ITIME)=XSQS(J,ITIME)+OUT(J+1)**2
       FVARS(J,ITIME)=FVARS(J,ITIME)+OUT(J+12)
       XM(J,ITIME)=XSUM(J,ITIME)/FLOAT(JJ)
       IF(JJ.EQ.1)GO TO 310
       FVAR(J,ITIME)=FVARS(J,ITIME)/JJ
       VAR(J,ITIME)=XSQS(J,ITIME)/FLOAT(JJ-1)-FLOAT(JJ)*XM(J,ITIME)**2/
      .FLOAT(JJ-1)
       IF(VAR(J,ITIME).LT.0.)VAR(J,ITIME)=0.
  310  CONTINUE
       IF(TYME.GT.5.73)GO TO 501
C                       ******************************
C      PROPAGATION OF TRUTH MODEL
C                       ******************************
       CALL TRAJ
C                       ******************************
C      PROPAGATION OF FILTER STATES AND FILTER COVARIANCE
C                       ******************************
       CALL DXCOVAR(2,DT)
C                       ******************************
C      COMPUTATION OF MEASUREMENT NOISE
C                       ******************************
       CALL NOISE(CORR0,0.,CORRUP)                                       000120
```

```
      X0=X0-2*X0*DT+CORRUP
      CALL NOISE(CORR1,0.,CORRUP)
      MEAS(1)=TS(1)+X0+CORRUP
      CALL NOISE(CORR0,0.,CORRUP)
      X1=X1-2*X1*DT+CORRUP
      CALL NOISE(CORR1,0.,CORRUP)
      MEAS(2)=TS(2)+X1+CORRUP
      CALL NOISE(CORR5,0.,CORRUP)
      X2=X2-.037*X2*DT+CORRUP
      CALL NOISE(CORR7,0.,CORRUP)
      MEAS(3)=TS(3)+X2+CORRUP
      CALL NOISE(CORR6,0.,CORRUP)
      X3=X3-.007*X3*DT+CORRUP
      CALL NOISE(CORR7,0.,CORRUP)
      MEAS(4)=TS(4)+X3+CORRUP
      CALL NOISE(CORR2,0.,CORRUP)
      X4=X4-2*X4*DT+CORRUP
      CALL NOISE(CORR3,0.,CORRUP)
      MEAS(5)=TS(5)+(X4+CORRUP)/1000
      CALL NOISE(CORR4,0.,CORRUP)
      X5=X5-1*X5*DT+CORRUP
      CALL NOISE(CORR5,0.,CORRUP)
      MEAS(6)=TS(6)+(X5+CORRUP)/1000
C                             *****************************
C     UPDATE OF FILTER STATES AND FILTER COVARIANCE WITH MEASUREMENTS .
C                             *****************************
      CALL EXGAIN
      CALL DXPPLUS
      CALL DXSPLUS
      GO TO 500
C                             *****************************
C     END OF RUN
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
C                                   ******************************
  501 JJ=JJ+1
      DO 317 III=60,240,60
      VPT(JJ,III/60)=VAR(7,III)
      VP2(JJ,III/60)=VAR(8,III)
  317 WRITE(8)VPT(JJ,III/60),VP2(JJ,III/60)
      IF(JJ.LE.20)GO TO 999
C                                   ******************************
C     OUTPUT OF STATISTICS
C                                   ******************************
      DO 733 I=1,11
      DO 733 J=1,238
  733 WRITE(8)XM(I,J),FVAR(I,J),VAR(I,J)
      DO 211 J=1,11
      TYME=0.
      WRITE 200,J
      DO 211 I=1,238,10
      VAR(J,I)=SQRT(VAR(J,I))
      WRITE 201,TYME,XM(J,I),FVAR(J,I),VAR(J,I)
  211 TYME=TYME+.2
  200 FORMAT(T25,"STATE  ",I2,//,T5,"TIME",T25,"MEAN",T45,"FILTER VARIAN
     .CE",T65,"VARIANCE",/)
  201 FORMAT(T5,4(G12.6,8X),/)
  9   CONTINUE
      STOP
      END
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
      SUBROUTINE TRAJ
C                                                                       000390
C                   ********************************                    002270
C     SUBROUTINE TRAJ GENERATES THE NOMINAL TRAJECTORIES FOR THE        002280
C     MONTE CARLO ANALYSIS                                              002290
C                   ********************************                    002300
C
      IMPLICIT REAL(J-N)                                                000293
      DIMENSION          VMIS(3),AMIS(3),ANCL(3),MCL(3),MISZ(3),RANG(3) 000103
      DIMENSION LOS(3),DA(3),ALDS(3),ALPCH(3),ALYAW(3),WIN(3),WMIS(3)   000113
      DIMENSION          VOWN(3),AOWN(3),P(8),Y(13),NORM(3),ALM(3)
      DIMENSION XOWN(3),XMIS(3)
      COMMON/SITE1A/V(3),B(3)
      COMMON/SITE1/K1,K2,K3,K4,K5,K6,W,TS(11),X,DY,TRJCNT,PNK,MODE1,TAP8
      LOGICAL MODE1,TAP8
      INTEGER TRJCNT
      DOUBLE PRECISION RANGE,MNORM
      IF (MODE1)GO TO 77                                                002380
C
C                   ********************************                    002390
C     INITIALIZITION OF SYSTEM PARAMETERS                               002400
C
C      DEFINITIONS
C     MASS= MISSILE MASS
C     IYY = MOMENT OF INERTIA, MISSILE YY AXIS, WHICH EQUALS
C         THE MOMENT OF INETIA OF THE MISSILE ZZ AXIS
C     OMEGA= AUTOPILOT 2ND ORDER NAT. FREQ COEFF.
C     ZETA= AUTOPILOT 2ND ORDER DAMPING COEFFICIENT
C     A= MACH CONVERSION AT 10,000 FEET STANDARD DAY
C     D= CHARACTERISTIC LENGTH OF MISSILE
C     RHO= AIR DENSITY
C     XKR= ABERATION SLOPE FOR THE NOSE CONE
C     TAU1=            SEEKER TIME CONSTANT
C     TAU2=  PROPORTIONAL NAV LAW TIME CONSTANT
C     LAM=             ACTUATOR TIME CONSTANT
C     G= ACCELERATION OF GRAVITY
```

```
C      SURF= EFFECTIVE SURFACE AREA FOR MOMENT EQUATIONS
C
C                            REFERENCE FRAMES
C      INERTIAL:  X IS HORIZONTAL, ORIGIN AT THE TRACKER.
C                 MISSILE IS IN THE (X,Y) PLANE
C                 Y AND Z ARE ELEVATED 45 DEGREES FROM HORIZONTAL
C      MISSILE BODY FRAME:  MCL IS MISSILE CENTERLINE
C                           MISZ POINTS DOWN (NEGATIVE YAW AXIS)
C                           NORM IS SOMETIMES THE RIGHT SIDE (PITCH) AXIS
C
C       SEEKER FRAME:  ANCL IS THE SEEKER CENTERLINE
C                      DA IS SOMETIMES THE RIGHT SIDE (PITCH) AXIS
C                      WIN IS SOMETIMES THE UP (YAW) AXIS
C
C       ALL VECTORS ARE EXPRESSED IN INERTIAL COORDINATES AND
C       ARE UNIT VECTORS WHERE APPROPRIATE
C
C                      *****************************
C
       MASS=4
       IYY=50
       SURF=.137
       OMEGA=7.07
       ZETA=.707
       D=.417
       A=1017
       RHO=.001066
       SIGMA=.005
       XKR=.001
       LAMN=.1
       TAU1=.075
       TAU2=0.3
       LAM=50
       G=32.174
C
C                      *****************************
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
C         CORR     (SEEKER SCINTILATION/GLINT NOISE)                          002590
C         CORR1    (SEEKER THERMAL NOISE)                                     002600
C         CORR2= RANGE RATE 1ST ORDER LAG
C         CORR3=  RANGE RATE WHITE
C                              ******************************                 002610
C
          CORR=0.003998                                                       002620
          CORR1=0.003                                                         002630
          CORR2=.8464                                                         000290
          CORR3=7.                                                            000300
C
C                              ******************************                 002640
C         INITIALIZATION OF TRACKER KINEMATICS                                002650
C                              ******************************                 002660
C
          DO 300 I=1,3                                                        000320
      300 XOWN(I)=0.                                                          000330
          VOWN(1)=K1                                                          000340
          VOWN(2)=K3+K+*W                                                     000350
          VOWN(3)=K5+K5*W*SIN(.76)
C
C                              ******************************                 002930
C         MISSILE INITIALIZATION                                              000370
C         NOTE:   VMIS(3) MUST EQUAL VOWN(3) FOR PROPER START
C                              ******************************                 003000
C
          XMIS(1)=-3+21.                                                      000400
          XMIS(2)=-9397.                                                      000410
          XMIS(3)=XOWN(3)
          VM=3051                                                             000430
          MACH=VM/A                                                           003170
          CD=2/SQRT(MACH)
          ADRAG=((RHO/(2*MASS))*SURF*CD*VM**2)/G
          VMIS(3)=VOWN(3)
          DO 50 I=1,3
       50 RANG(I)=XOWN(I)-XMIS(I)
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
      RANGE=DSQRT(ABS(RANG(1))**2.D0+ABS(RANG(2))**2.D0+ABS(RANG(3))
     .**2.D0)
      DO 753 I=1,3
      LOS(I)=RANG(I)/RANGE
  753 ANCL(I)=LOS(I)
      Y(1)=SQRT(VM**2-VMIS(3)**2)
      VMIS(1)=Y(1)*LOS(1)
      VMIS(2)=Y(1)*LOS(2)
      DO 200 I=1,3                                                  000500
      MCL(I)=VMIS(I)/VM                                             000520
  200 AMIS(I)=-ADRAG*MCL(I)                                         000193
      MISZ(1)=MCL(1)*MCL(2)                                         000540
      MISZ(2)=MCL(3)**2-MCL(1)**2                                   000550
      MISZ(3)=-MCL(2)*MCL(3)                                        000560
      MNORM=DSQRT(ABS(MISZ(1))**2.D0+ABS(MISZ(2))**2.D0+ABS(MISZ(3))
     .**2.D0)
      DO 210 I=1,3                                                  000580
  210 MISZ(I)=MISZ(I)/MNORM                                         000590
      VC =SQRT(((VMIS(1)-VOWN(1))*LOS(1))**2+((VMIS(2)-VOWN(2))*LOS(2))
     .**2+((VMIS(3)-VOWN(3))*LOS(3))**2)
C
C                   ******************************                  002770
C     STATE INITIALIZATION FOR INTERFACE
C     NOTE:  ALL STATES ARE OUTPUTTED IN KILOFEET UNITS RATHER
C            THAN FEET
C                   ******************************
      THETAY=0.
      THETAP=0.
      TS(1)=ATAN(-LOS(3)/LOS(1))
      TS(2)=ASIN(LOS(2))
      SI=SIN(TS(1))
      CO=COS(TS(1))
      SIT=SIN(TS(2))
      COT=COS(TS(2))
      B(1)=COT*AOWN(2)-SIT*(CO*AOWN(1)-SI*AOWN(3))
      B(3)=SIT*AOWN(2)+COT*(CO*AOWN(1)-SI*AOWN(3))
```

```
      B(2)=CO*AOWN(3)+SI*AOWN(1)
      V(1)=COT*VOWN(2)-SIT*(CO*VOWN(1)-SI*VOWN(3))
      V(3)=SIT*VOWN(2)+COT*(CO*VOWN(1)-SI*VOWN(3))
      V(2)=CO*VOWN(3)+SI*VOWN(1)
      VM1=COT*VMIS(2)-SIT*(CO*VMIS(1)-SI*VMIS(3))
      VM2=CO*VMIS(3)+SI*VMIS(1)
      TS(3)=(VM2-V(2))/RANGE
      TS(4)=(V(1)-VM1)/RANGE
      DO 21 I=1,3
      B(I)=B(I)/1000
   21 V(I)=V(I)/1000
      TS(5)=-RANGE/1000
      TS(6)=VC/1000
      TS(7)=TS(8)=0
      TS(9)=PNK
      TS(10)=.7
      TS(11)=23.13708E6
      DO 205 I=1,11                                                    000610
  205 Y(I)=0.
C                     ************************************             003180
C     INITIALIZTION OF AERODYNAMIC COEFFICIENTS AND                    003190
C     STABILITY DERIVATIVES FOR FLIGHT DYNAMICS
C                     ************************************             003210
      DP=(RHO/2)*VM**2                                                 003150
      CMQ=-(4947+MACH*(-2563+MACH*2125))/(2.957+MACH*(-3.078+MACH))    003230
      CMA=-(2.292+MACH*(-2.858+MACH*1.824))/(1.640+MACH*(-2.171+MACH)) 003240
      CMD=-(8.294+MACH*(-4.894+MACH*1.191))/(3.220+MACH*(-2.760+MACH)) 003250
      CNA=(1.250+MACH*(-.6398+MACH*.2834))/(2.393+MACH*(-2.132+MACH))  003260
      CND=(.3115+MACH*(-.2199+MACH*.0865))/(3.761+MACH*(-3.126+MACH))  003270
      CMA=CMA*57.295780                                                003280
      CMD=CMD*57.295780                                                003290
      CNA=CNA*57.295780                                                003300
      CND=CND*57.295780                                                003310
      MQ=(DP*SURF*D**2*CMQ)/(2*IYY*VM)                                 003330
      MA=(DP*SURF*D*CMA)/IYY                                           003340
      MD=(DP*SURF*D*CMD)/IYY                                           003350
```

```
      LD=(DP*SURF*CND)/(MASS*VM)
      LA=(DP*SURF*CNA)/(MASS*VM)
C               ******************************
C     INITIALIZATION OF AUTOPILOT FEEDBACK COEFFICIENTS AND GAIN
C               ******************************
      H1=-LA*(MA-(MD/LD)*LA+OMEGA**2+(2*ZETA*OMEGA+MQ)*(MD/LD+MQ))/(MD*
     X(MA-MD*LA/LD+OMEGA**2)+(MQ+2*ZETA*OMEGA)*(LD*MA-MD*LA))
      H2=(LD*MA*(LA-2*ZETA*OMEGA-MQ)+MD*(LA*(2*ZETA*OMEGA-LA)-OMEGA**2
     X-MA))/(VM*LD*(MD*(MA-MD*LA/LD+OMEGA**2)+(MQ+2*ZETA*OMEGA)*(LD*
     XMA-MD*LA)))
      KD=H1/VM+H2-(1/VM)*((MA+LA*MQ)/(LA*MD-MA*LD))
      DX=DY/FLOAT(TRJCNT)
      CORR=CORR*SQRT(DX)
      CORR1=CORR1*SQRT(DX)
      CORR2=CORR2*SQRT(DX)
      CORR3=CORR3*SQRT(DX)
      IF(.NOT.TAP8)GO TO 505
      PRINT(4,20)
   20 FORMAT(5X,"TIME",T15,"THETA1",T25,"THETA2",T35,"RANGE",
     .T45,"VCM",T55,"AL PITCH",T65,"AC PITCH",T75,"AL YAW",T85,"AC YAW")
      PRINT(4,15) X,TS(1),TS(2),TS(5),TS(6),TS(7),Y(8),TS(8),Y(4)
   15 FORMAT(/,5X,10(G8.2,2X))
      PRINT(5,501)
      PRINT(5,455)
      PRINT(5,450)X,XOWN(1),VOWN(1),AOWN(1),XMIS(1),VMIS(1),AMIS(1),
     .ALPCH(1),ALYAW(1)
      PRINT(5,451)  XOWN(2),VOWN(2),AOWN(2),XMIS(2),VMIS(2),AMIS(2),
     .ALPCH(2),ALYAW(2)
      PRINT(5,451)  XOWN(3),VOWN(3),AOWN(3),XMIS(3),VMIS(3),AMIS(3),
     .ALPCH(3),ALYAW(3)
      PRINT(5,455)
      PRINT(6,502)
      PRINT(6,455)
      PRINT(6,550)X,MCL(1),MISZ(1),LOS(1),ALOS(1),ANCL(1),WIN(1)
      PRINT(6,551)  MCL(2),MISZ(2),LOS(2),ALOS(2),ANCL(2),WIN(2)
      PRINT(6,551)  MCL(3),MISZ(3),LOS(3),ALOS(3),ANCL(3),WIN(3)
```

```
      PRINT(6,455)
  501 FORMAT(5X,"X",T15,"XOWN",T25,"VOWN",T35,"AOWN",T45,"XMIS",T55,"VMI
     .S",T65,"AMIS",T75,"ALPCH",T85,"ALYAW")
  502 FORMAT(5X,"X",T15,"MCL",T25,"MISZ",T35,"LOS",T45,"ALOS",T55,"ANGL"
     .,T65,"WIN")
  505 CONTINUE
      RETURN
C                             *******************************
C       INITIALIZATION COMPLETE.  THE FOLLOWING IS THE RUN PORTION
C                             *******************************
77    CONTINUE                                                          003570
      DO 78 IT=1,TRJCNT
C
C                             *******************************
C     NOISE UPDATE                                                      000110
C                             *******************************
C
      CALL NOISE(CORR,0.,CORRUP)                                        000130
      Y(9)=Y(9)-.1*Y(9)*DX+CORRUP                                       000140
      CALL NOISE(CORR1,0.,CORRUP)                                       000150
      NOISE1=Y(9)+CORRUP                                                000160
      CALL NOISE(CORR,0.,CORRUP)                                        000170
      Y(10)=Y(10)-.1*Y(10)*DX+CORRUP                                    000180
      CALL NOISE(CORR1,0.,CORRUP)                                       000190
      NOISE2=Y(10)+CORRUP                                               000200
      CALL NOISE(CORR2,0.,CORRUP)                                       000210
      Y(11)=Y(11)-Y(11)*DX+CORRUP                                       000220
      CALL NOISE(CORR3,0.,CORRUP)                                       000230
      NOISE3=Y(11)+CORRUP                                               000240
C                             *******************************
C     DETERMINATION OF ABERATION ERROR                                  000260
C                             *******************************
      THETA=ACOS(MCL(1)*LOS(1)+MCL(2)*LOS(2)+MCL(3)*LOS(3))             000330
      IF(THETA.LT.1.5)GO TO 778
      PRINT *,"MISSILE HAS LOST LOCK"
      STOP
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

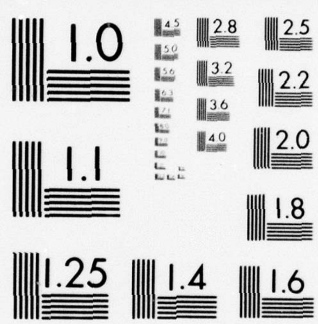
```
  778 IF (THETA.GT.-1.5)GO TO 779
      PRINT *,"MISSILE HAS LOST LOCK"
      STOP
  779 CONTINUE
      IF(THETA.EQ.0.)GO TO 103
      PHEE=XKP*THETA
      CALL TIN(MCL,LOS,THETA,PHEE,ALOS)
      CALL CROSS(MCL,LOS,NORM,DX,0)
      DO 104 I=1,3
  104 DA(I)=MCL(I)
      GO TO 105
  103 CALL CROSS(MIS7,LOS,NORM,DX,0)
      DO 106 I=1, 3
      ALOS(I)=LOS(I)
  106 DA(I)=MIS7(I)
C
C                          ******************************
C    CORRUPTION OF APPARENT LOS(ALOS) BY 3-D NOISE
C                          ******************************
C
  105 THETA=ACOS(DA(1)*ALOS(1)+DA(2)*ALOS(2)+DA(3)*ALOS(3))
      CALL TIN(DA,ALOS,THETA,NOISE1,ALOS)
      THETA=ACOS(0.)
      PHEE=NOISE2
      CALL TIN(NORM,ALOS,THETA,PHEE,ALOS)
C
C                          ******************************
C     THE SENSED LINE OF SIGHT HAS NOW BEEN PRODUCED.
C       THE FOLLOWING IS THE SEEKER MODEL, WHICH PRODUCES GUIDANCE INPUTS
C     SENSOR ERROR, E, PASSED THROUGH A FIRST ORDER LAG
C     PROVIDES THE ANGULAR RATE INPUT TO THE GUIDANCE PKG.
C                          ******************************
C
      E=     (ALOS(1)*ANCL(1)+ALOS(2)*ANCL(2)+ALOS(3)*ANCL(3))
      IF(E.GE.1.)E=1.
      E=ACOS(E)
```

```
      IF(E.EQ.0.)GO TO 120
      DO 32 I=1,3
   32 NORM(I)=ALOS(I)-ANCL(I)
      MNORM=DSQRT(ABS(NORM(1))**2.D0+ABS(NORM(2))**2.D0+ABS(NORM(3))**2.
     .D0)
      DO 63 I=1,3
   63 NORM(I)=NORM(I)/MNORM
      WIN(1)=WIN(3)=0
      WIN(2)=1.
      THETA=ACOS(ANCL(2))
      PHEE=1.57079632679489664423-THETA
      CALL TIN(ANCL,WIN,THETA,PHEE,WIN)
      CALL CROSS(ANCL,WIN,DA,DX,0)
  120 Y(12)=THETAY
      Y(13)=THETAP
      THETAY=(NORM(1)*DA(1)+NORM(2)*DA(2)+NORM(3)*DA(3))*E
      THETAP=(NORM(1)*WIN(1)+NORM(2)*WIN(2)+NORM(3)*WIN(3))*E
      Y(12)=THETAY-Y(12)
      Y(13)=THETAP-Y(13)
      WPITCH=WPITCH+Y(13)/TAU1
      WYAW=WYAW+Y(12)/TAU1
C                   *******************************
C     WIN IS NOW THE ANGULAR RATE USED TO UPDATE SEEKER POSITION
C                   *******************************
      DO 33 I=1,3
   33 WIN(I)=-WYAW*WIN(I) +WPITCH*DA(I)
      CALL CROSS(WIN,ANCL,ANCL,DX,2)
   31 CONTINUE
C
C                   *******************************
C     STATE EQUATIONS FOR MISSILE FLIGHT DYNAMICS
C
C     EQUATIONS 1 AND 8 COMPOSE THE GUIDANCE PACKAGE OF THE MISSILE
C    STATES 1 THROUGH 4 ARE YAW
C             5 THROUGH 8 ARE PITCH
C     165       MISSILE PITCH RATE
```

AD-A064 760 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OHIO SCH--ETC F/G 15/3.1
A PRACTICAL THREE DIMENSIONAL, 11 STATE EXTENDED KALMAN FILTER --ETC(U)
DEC 78 C W HLAVATY

UNCLASSIFIED AFIT/GGC/EE/78-7-VOL-2 NL

2 OF 3
AD
A064760




MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

```
C     2&5       MISSILE LATERAL ACCELERATION IN G'S                     000910
C     3&7       ELEVATOR DEFLECTION IN RAD                              000920
C     4&5       COMMANDED ACCELERATION IN G'S                           000930
C                          *****************************                003590
C                                                                       000940
      P(1)=MO*Y(1)+((G*MA)/(VM*LA))*Y( 2)+(MD-LD*MA/LA)*Y( 3)
      P( 2)=1/G*(VM*LA-H1*LAM*VM*LD)*Y(1)-(LA+H2*LAM*VM*LD)*Y( 2)-1/G*
     XLAM*VM*LD*Y( 3)+LAM*VM*LD*KD*Y( 4)
      P(3)=-H1*LAM*Y(1)-G*H2*LAM*Y( 2)-LAM*Y( 3)+G*LAM*KD*Y( 4)
      P( 4)=+PNK*VC/(G*TAU2)*WYAW   -1/TAU2*Y(4)
      P(5)=MO*Y(5)+((G*MA)/(VM*LA))*Y( 6)+(MD-LD*MA/LA)*Y( 7)           003660
      P( 6)=1/G*(VM*LA-H1*LAM*VM*LD)*Y(5)-(LA+H2*LAM*VM*LD)*Y( 6)-1/G*  003670
     XLAM*VM*LD*Y( 7)+LAM*VM*LD*KD*Y( 8)                                003680
      P(7)=-H1*LAM*Y(5)-G*H2*LAM*Y( 6)-LAM*Y( 7)+G*LAM*KD*Y( 8)         003690
      P( 8)=+PNK*VC/(G*TAU2)*WPITCH-1/TAU2*Y(8)
C                                                                       000600
C                          *****************************                003760
C     STATE UPDATES (RECTANGULAR INTEGRATION)                           003790
C                          *****************************                003800
C                                                                       000540
      Y(1)=Y(1)+P(1)*DX                                                 003810
      Y(2)=Y(2)+P(2)*DX                                                 003820
      Y(3)=Y(3)+P(3)*DX                                                 003830
      Y(4)=Y(4)+P(4)*DX                                                 003840
      Y(5)=Y(5)+P(5)*DX                                                 003850
      Y(6)=Y(6)+P(6)*DX                                                 003860
51    Y(7)=Y(7)+P(7)*DX                                                 003900
      Y(8)=Y(8)+P(8)*DX                                                 003910
      CALL CROSS(MISZ,MCL,NORM,DX,0)
  148 DO 150 I=1,3
      ALYAW(I)=NORM(I)*Y(2)                                             000241
      ALPCH(I)=-MISZ(I)*Y(5)                                            000251
C                                                                       000520
C                          *****************************                003870
C     AN ACCELERATION LIMITER WOULD GO HERE TO LIMIT ALYAW AND ALPCH
C                          *****************************                003890
```

```
C                                                                       000270
      WMIS(I)=Y(1)*MISZ(I)+Y(5)*NORM(I)                                 000263
  150 ALM(I)=ALYAW(I)+ALPCH(I)
C
C                     *******************************                   003500
C     TRAJECTORY PROPAGATION                                            003510
C     ACCELERATIONS, VELOCITIES, AND POSITION UNITS ARE FEET AND SECONDS
C                     *******************************                   003520
C
      X=X+DX                                                            004010
      AMIS(1)=(ALM(1)-ADRAG*VMIS(1)/VM)*G                               000371
      AMIS(2)=(ALM(2)-.707-ADRAG*VMIS(2)/VM)*G
      AMIS(3)=(ALM(3)-.707-ADRAG*VMIS(3)/VM)*G
      AOWN(1)=-K2*W**2*COS(W*X)                                         000392
      AOWN(2)=-K4*W**2*SIN(W*X)                                         000393
      AOWN(3)=-K6*W**2*SIN(W*X-.76)
      DO 175 I=1,3                                                      000401
      VMIS(I)=VMIS(I)+AMIS(I)*DX                                        000411
      XMIS(I)=XMIS(I)+VMIS(I)*DX                                        000421
      VOWN(I)=VOWN(I)+AOWN(I)*DX                                        000431
      XOWN(I)=XOWN(I)+VOWN(I)*DX                                        000441
  170 CONTINUE                                                          000283
C                                                                       000250
C                     *******************************                   004290
C     UPDATE OF MISSILE AERODYNAMIC COEFFICIENTS
C                     *******************************                   004310
C                                                                       000120
      VM=DSQRT(ABS(VMIS(1))**2.D0+ABS(VMIS(2))**2.D0+ABS(VMIS(3))**2.D0)
      MACH=VM/A                                                         004280
      DP=(RHO/2)*VM**2                                                  004270
      AL=DSQRT(ABS(ALM(1))**2.D0+ABS(ALM(2))**2.D0+ABS(ALM(3))**2.D0)
      CN=(AL*G*MASS*2)/(RHO*SURF*VM**2)                                 000331
      CMQ=-(+943+MACH*(-2563+MACH*2125))/(2.957+MACH*(-3.076+MACH))     004330
      CMD=-(8.294+MACH*(-4.894+MACH*1.191))/(3.220+MACH*(-2.760+MACH))
      CMA=-(2.292+MACH*(-2.856+MACH*1.824))/(1.646+MACH*(-2.171+MACH))
      CNA=(1.250+MACH*(-.6398+MACH*.2834))/(2.393+MACH*(-2.132+MACH))
```

```
      CND=(.3115+MACH*(-.2199+MACH*.0865))/(3.761+MACH*(-3.126+MACH))   004370
      CMA=CMA*57.295780                                                 004380
      CMD=CMD*57.295780                                                 004390
      CNA=CNA*57.295780                                                 004400
      CND=CND*57.295780                                                 004410
      MQ=(DP*SURF*D**2*CMQ)/(2*IYY*VM)                                  004420
      MA=(DP*SURF*D*CMA)/IYY                                            004430
      MD=(DP*SURF*D*CMD)/IYY                                            004440
      LD=(DP*SURF*CND)/(MASS*VM)                                        004450
      LA=(DP*SURF*CNA)/(MASS*VM)                                        004460
C                                                                       000100
C                   ******************************                      004470
C     UPDATE OF AUTOPILOT FEEDBACK COEFFICIENTS AND GAIN
C                   ******************************                      004490
C                                                                       000330
      H1=-LA*(MA-(MD/LD)*LA+OMEGA**2+(2*ZETA*OMEGA+MQ)*(MD/LD+MQ))/(MD* 004500
     X(MA-MD*LA/LD+OMEGA**2)+(MD+2*ZETA*OMEGA)*(LD*MA-MD*LA))           004510
      H2=(LD*MA*(LA-2*ZETA*OMEGA-MQ)+MD*(LA*(2*ZETA*OMEGA-LA)-OMEGA**2  004520
     X-MA))/(VM*LD*(MD*(MA-MD*LA/LD+OMEGA**2)+(MQ+2*ZETA*OMEGA)*(LD*    004530
     XMA-MD*LA)))                                                       004540
      KD=H1/VM+H2-(1/VM)*((MA+LA*MQ)/(LA*MD-MA*LD))                     004550
      CD=2/SQRT(MACH)+CN**2/CNA                                         004560
      ADRAG=((RHO/(2*MASS))*SURF*CD*VM**2)/G                            004580
      IF ((WMIS(1)**2+WMIS(2)**2+WMIS(3)**2).EQ.0.)GO TO 151
C                   ******************************
C     UPDATE OF MISSILE FRAME ORIENTATION, RANGE, AND CLOSING VELOCITY
C                   ******************************
      CALL CROSS(WMIS,MCL,MCL,DX,2)
      IF(Y(5).EQ.0.)GO TO 151
      CALL CROSS(WMIS,MISZ,MISZ,DX,2)
  151 DO 152 I=1,3
  152 RANG(I)=XOWN(I)-XMIS(I)
      RANGE=DSQRT(ABS(RANG(1))**2.D0+ABS(RANG(2))**2.D0+ABS(RANG(3))
     .**2.D0)
      DO 153 I=1,3                                                      000260
  153 LOS(I)=RANG(I)/RANGE                                              000270
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
      VCM=SQRT(((VMIS(1)-VOWN(1))*LOS(1))**2+((VMIS(2)-VOWN(2))*LOS(2))
     .**2+((VMIS(3)-VOWN(3))*LOS(3))**2)
      VC=VCM+NOISE3                                                     000253
   78 CONTINUE                                                          000273
C
C                            ******************************
C     UPDATE OF "TRUE STATES" AND PROGRAM INTERFACE QUANTITIES.
C                            ******************************
C
      TS(1) =ATAN(-LOS(3)/LOS(1))
      SI=SIN(TS(1))
      CO=COS(TS(1))
      TS(2) =ASIN(LOS(2))
      SIT=SIN(TS(2))
      COT=COS(TS(2))
      TS(5)=-RANGE/1000
      TS(6)=VCM/1000
      DA(1)=DA(3)=0
      DA(2)=1
      THETA=ACOS(MOL(2))
      PHEE=1.5707963267948966-THETA
      CALL  TIN(MOL,DA,THETA,PHEE,DA)
      CALL  CROSS(MOL,DA,NORM,DY,0)
      TS(7)=(DA(1)*ALM(1)+DA(2)*ALM(2)+DA(3)*ALM(3))*G
      TS(8)=(NORM(1)*ALM(1)+NORM(2)*ALM(2)+NORM(3)*ALM(3))*G
      TS(7)=TS(7)/1000
      TS(8)=TS(8)/1000
      B(1)=COT*AOWN(2)-SIT*(CO*AOWN(1)-SI*AOWN(3))
      B(3)=SIT*AOWN(2)+COT*(CO*AOWN(1)-SI*AOWN(3))
      B(2)=CO*AOWN(3)+SI*AOWN(1)
      V(1)=COT*VOWN(2)-SIT*(CO*VOWN(1)-SI*VOWN(3))
      V(3)=SIT*VOWN(2)+COT*(CO*VOWN(1)-SI*VOWN(3))
      V(2)=CO*VOWN(3)+SI*VOWN(1)
      VM1=COT*VMIS(2)-SIT*(CO*VMIS(1)-SI*VMIS(3))
      VM2=CO*VMIS(3)+SI*VMIS(1)
      TS(3)=(VM2-V(2))/RANGE
```

```
      TS(4)=(V(1)-VM1)/RANGE
      DO 22 I=1,3
      B(I)=B(I)/1000
   22 V(I)=V(I)/1000
      IF(.NOT.TAPE)RETURN
      PRINT(6,450)X,XOWN(1),VOWN(1),AOWN(1),XMIS(1),VMIS(1),AMIS(1),
     .ALPCH(1),ALYAW(1)
      PRINT(6,451)  XOWN(2),VOWN(2),AOWN(2),XMIS(2),VMIS(2),AMIS(2),
     .ALPCH(2),ALYAW(2)
      PRINT(6,451)  XOWN(3),VOWN(3),AOWN(3),XMIS(3),VMIS(3),AMIS(3),
     .ALPCH(3),ALYAW(3)
      PRINT(4,15)X,TS(1),TS(2),TS(5),TS(5),TS(7),Y(8),TS(8),Y(4)
      PRINT(6,455)
      PRINT(6,550)X,MCL(1),MISZ(1),LOS(1),ALOS(1),ANCL(1),WIN(1)
      PRINT(6,551)  MCL(2),MISZ(2),LOS(2),ALOS(2),ANCL(2),WIN(2)
      PRINT(6,551)  MCL(3),MISZ(3),LOS(3),ALOS(3),ANCL(3),WIN(3)
      PRINT(6,455)
  455 FORMAT(/,2X,".")
  551 FORMAT(15X,9(G8.2,2X))
  451 FORMAT(15X,9(G8.2,2X))
  550 FORMAT(5X,9(G8.2,2X))
  450 FORMAT(5X,9(G8.2,2X))
      RETURN
       END
      SUBROUTINE NOISE(RMSCORR,XMEAN,CORRUP)
C                          ********************************
C     SUBROUTINE NOISE GENERATES THE TRUTH MODEL NOISES
C      FOR FIRST ORDER LAG NOISE,
C     RMSCORR=(Q*DT)**.5  WHERE Q IS CONTINUOUS WHITE NOISE STRENGTH
C                         DT IS THE EULER INTEGRATION TIME STEP
C                          ********************************
      GAUSS=0.
      DO 10 I=1,12
      GAUSS=GAUSS+RANF(DUM)
   10 CONTINUE
      GAUSS=GAUSS-6.
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
CORRUP=GAUSS*RMSCORR+XMEAN
RETURN
END
```

```
      SUBROUTINE TIN(D,E,TH,PHE,C)
C
C     SUBROUTINE TIN WILL SOLVE THE FOLLOWING THREE SIMULTANEOUS EQUATIONS:
C
C                       (AXB)*C = COS(90)
C                       A * C   = COS(TH+PHE)
C                       B * C   = COS(PHE)
C
C     A AND B MUST BE UNIT VECTORS.  THE CONSTRUCTED UNIT VECTOR (C)
C     IS COPLANAR TO A AND B AND SEPERATED FROM B BY ANGLE PHE
       DIMENSION A(3),B(3),C(3),D(3),E(3)
      DOUBLE PRECISION DET,A,B,X1,X2,X3
      DO 10 I=1,3
      A(I)=D(I)
   10 B(I)=E(I)
    6  X1=A(2)*B(3)-A(3)*B(2)
       X2=A(3)*B(1)-A(1)*B(3)
       X3=A(1)*B(2)-A(2)*B(1)
      DET=X1**2+X2**2+X3**2
       D1=COS(TH+PHE)
       D2=COS(PHE)
       C(1)=((B(2)*X3-B(3)*X2)*D1+(X2*A(3)-X3*A(2))*D2)/DET
       C(2)=((B(3)*X1-B(1)*X3)*D1+(X3*A(1)-X1*A(3))*D2)/DET
       C(3)=((B(1)*X2-B(2)*X1)*D1+(X1*A(2)-X2*A(1))*D2)/DET
       RETURN
      END
```

```
      SUBROUTINE CROSS(D,E,F,DX,IT)
C     SUBROUTINE CROSS WILL CROSS TWO VECTORS, A X B = C,
C     WHERE C IS A PRODUCED UNIT VECTOR.  ZERO VECTORS OR COLINEAR
C     VECTORS ARE NOT ALLOWED.
C      IF "IT" DOES NOT EQUAL AN INTEGER 0,THEN SUBROUTINE CROSS
C     WILL PERFORM THE EULER VECTOR UPDATE
C                     B(T+1) = B(T) +(A(T) X B(T))*DX
C
C
      DIMENSION A(3),B(3),C(3),D(3),E(3),F(3)
      DOUBLE PRECISION A,B,C,CN
      DO 5 I=1,3
      A(I)=D(I)
    5 B(I)=E(I)
   20 C(1)=A(2)*B(3)-A(3)*B(2)
      C(2)=A(3)*B(1)-A(1)*B(3)
      C(3)=A(1)*B(2)-A(2)*B(1)
      IF(IT.EQ.0)GO TO 40
      DO 30 I=1,3
   30 C(I)=C(I)*DX+B(I)
   40 CN=DSQRT(C(1)**2+C(2)**2+C(3)**2)
      DO 10 I=1,3
   10 F(I)=C(I)/CN
      RETURN
      END
```

```
      SUBROUTINE DXCOVAR(N,DT)
      COMMON/SITE/B1(11,11),B2(11,11),P4(11,6),B5(6,6),B7(11)
      COMMON/SITE1/V(3),A(3)
      COMMON/SITE2/COT2,SIT2,COA2,SIA2,COA1,SIA1,VM,R,B,G,AD,COT1,SIT1
      COMMON/SITE3/Q(11,11),F(11,11),X(11),FP(11,11)
      DIMENSION XDT(8)
C
C      DXCOVAR PROPAGATES THE FILTER STATES AND THE FILTER COVARIANCE.
C       FMAT IS USED TO PROPAGATE THE FILTER COVARIANCE.
C      N IS THE NUMBER OF STEPS IN TIME INTERVAL DT.
C
      DO 100 IDID=1,N
      COT1=COS(X(1))
      SIT1=SIN(X(1))
      COT2=COS(X(2))
      SIT2=SIN(X(2))
      A2=ATAN(-(V(1)+X(4)*X(5))/(V(3)+X(5)))
      COA2=COS(A2)
      SIA2=SIN(A2)
      A1=ATAN((V(2)-X(3)*X(5))/((V(3)+X(5))*COA2-(V(1)+X(4)*X(5))*SIA2))
      COA1=COS(A1)
      SIA1=SIN(A1)
      VM=SQRT((V(1)+X(4)*X(5))**2+(V(2)-X(3)*X(5))**2+(V(3)+X(5))**2)
      AD=R/X(11)*SQRT(VM**3+B)+2*X(11)*(X(7)**2+X(8)**2)/(R+B*VM*57.3)
      CALL FMAT
      CALL MMPY(F,FP,B1,11,11,11)
      CALL MTRAN(11,11,B1,B2)
      CALL MADD(11,11,B1,B2,B1)
      CALL MADD(11,11,B1,Q,B1)
      DO 17 J=1,11
      DO 17 I=J,11
      FP(I,J)=FP(I,J)+B1(I,J)*DT/N
   17 FP(J,I)=FP(I,J)
      XDT(1)=X(3)/COT2
      XDT(2)=X(4)
      XDT(3)=(AD*SIA1-X(8)*COA1+A(2)-2*X(3)*X(6)+X(3)*X(4)*X(5)*SIT2/COT
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
     .2+G*COT1)/X(5)
      XDT(4)=(X(7)*COA2+SIA2*(SIA1*X(8)+AD*COA1)-G*COT2-A(1)-2*X(4)*X(6)
     .-X(3)**2*X(5)*SIT2/COT2-G*SIT2*SIT1)/X(5)
      XDT(5)=X(6)
      XDT(6)=X(7)*SIA2-COA2*(X(8)*SIA1+AD*COA1)-G*SIT2-A(3)+X(5)*(X(3)**
     .2+X(4)**2)+G*COT2*SIT1
      XDT(7)=(X(4)*X(6)*X(9)-X(7))/X(10)
      XDT(8)=-(X(3)*X(6)*X(9)+X(8))/X(10)
      DO 18 I=1,8
   18 X(I)=X(I)+XDT(I)*DT/N
  100 CONTINUE
      RETURN
      END
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
      SUBROUTINE FMAT                                                   000100
      COMMON/SITE1/V(3),A(3)                                            000110
      COMMON/SITE2/COT2,SIT2,COA2,SIA2,COA1,SIA1,VM,R,B,G,AD,COT1,SIT1  000130
      COMMON/SITE3/P(11,11),F(11,11),X(11),FP(11,11)                    000120
C                        ***************************************
C     SUBROUTINE FMAT EVALUATES THE F PARTIALS, WHICH ARE
C     NEEDED TO PROPAGATE THE COVARIANCE MATRIX
C
C     AD7 IS THE PARTIAL OF DRAG WITH RESPECT TO X(7)
C     AD8 IS THE PARTIAL OF DRAG WITH RESPECT TO X(8)
C    AD11 IS THE PARTIAL OF DRAG WITH RESPECT TO X(11)
C     X IS THE FILTER STATES,  F IS THE PARTIALS MATRIX, V IS THE
C     VELOCITY OF THE TRACKER, A IS THE ACCELERATION OF THE TRACKER
C                        ***************************************
      AD7=X(7)*(4*X(11)/(R*B*VM*57.3))
      AD8=X(8)*(4*X(11)/(R*B*VM*57.3))
      AD11=-R*SQRT(VM**3*B)/X(11)**2+2*(X(7)**2+X(8)**2)/(R*B*VM*57.3)
      F(1,2)=X(3)*SIT2/COT2**2                                          000180
      F(1,3)=1/COT2                                                     000190
      F(2,4)=1                                                          000200
      F(3,1)=-G*SIT1/X(5)
      F(3,2)=X(3)*X(4)/COT2**2                                          000220
      F(3,3)=-2*X(6)/X(5)+X(4)*SIT2/COT2                                000230
      F(3,4)=X(3)*SIT2/COT2                                             000240
      F(3,5)=-(AD*SIA1-X(8)*COA1+A(2)+G*COT1-2*X(3)*X(6))/X(5)**2
      F(3,6)=-2*X(3)/X(5)                                               000260
      F(3,7)=AD7*SIA1/X(5)                                              000270
      F(3,8)=AD8*SIA1/X(5)-COA1/X(5)                                    000280
      F(3,11)=AD11*SIA1/X(5)                                            000290
      F(4,1)=-G*SIT2*COT1/X(5)
      F(4,2)= G*(SIT2-COT2*SIT1)/X(5)-X(3)**2/COT2**2
      F(4,3)=-2*X(3)*SIT2/COT2                                          000310
      F(4,4)=-2*X(6)/X(5)                                               000320
      F(4,5)=-(X(7)*COA2+SIA2*(SIA1*X(8)+AD*COA1)-G*COT2-A(1)-2*X(5)    000330
     .*X(4)-G*SIT2*SIT1)/X(5)**2
      F(4,6)=-2*X(4)/X(5)                                               000350
```

```
F(4,7)=(COA2+SIA2*COA1*AD7)/X(5)
F(4,8)=SIA2/X(5)*(SIA1+COA1*AD8)
F(4,11)=SIA2*COA1*AD11/X(5)
F(5,6)=1
F(6,1)=G*COT1*COT2
F(6,2)=-G*(COT2+SIT2*SIT1)
F(6,3)=2*X(3)*X(5)
F(6,4)=2*X(4)*X(5)
F(6,5)=X(3)**2+X(4)**2
F(6,7)=SIA2-COA2*COA1*AD7
F(6,8)=-COA2*(SIA1+COA1*AD8)
F(6,11)=-COA2*COA1*AD11
F(7,4)=X(6)*X(9)/X(10)
F(7,6)=X(4)*X(9)/X(10)
F(7,7)=-1/X(10)
F(7,9)=X(4)*X(6)/X(10)
F(7,10)=-(X(4)*X(6)*X(9)-X(7))/X(10)**2
F(8,3)=-X(6)*X(9)/X(10)
F(8,6)=-X(3)*X(9)/X(10)
F(8,8)=F(7,7)
F(8,9)=-X(3)*X(6)/X(10)
F(8,10)=(X(3)*X(6)*X(9)+X(8))/X(10)**2
RETURN
END
```

```
      SUBROUTINE DYGAIN
C                          ******************************
C     SUBROUTINE DYGAIN COMPUTES THE KALMAN FILTER GAIN
C                          ******************************
      COMMON/SITE/A1(11,11),A2(11,11),A4(11,6),A5(6,6),A7(11)
      COMMON/SITE3/Q(11,11),F(11,11), FS(11),FP(11,11)
      COMMON/SITE5/FK(11,6)
      COMMON/SITE8/H(6,11),R(6,6)
      CALL  MTRAN(6,11,H,FK)
      CALL MMPY(FP,FK,A4,11,11,6)
      CALL MMPY(H,A4,A5,6,11,6)
      CALL MADD(6,6,A5,R,A5)
      CALL MINV(A5,A5,6)
      CALL MMPY(A4,A5,FK,11,6,6)
      RETURN
      END
```

```
      SUBROUTINE DXPPLUS
      COMMON/SITE/A1(11,11),A2(11,11),A4(11,6),A5(6,6),A7(11)
      COMMON/SITE3/Q(11,11),F(11,11),X(11),FP(11,11)
      COMMON/SITE5/FK(11,6)
      COMMON/SITE6/H(6,11),R(6,6)
C                         ******************************
C     SUBROUTINE DXPPLUS PERFORMS THE MEASUREMENT UPDATE OF THE
C     COVARIANCE MATRIX.
C                         ******************************
      CALL MMPY(FK,H,A1,11,6,11)
      CALL MMPY(A1,FP,A2,11,11,11)
      CALL MSUB(11,11,FP,A2,FP)
      RETURN
      END
```

```
      SUBROUTINE DXSPLUS
      COMMON/SITE/A1(11,11),A2(11,11),A4(11,6),A5(6,6),A7(11)
      COMMON/SITE3/Q(11,11),F(11,11), FS(11),FP(11,11)
      COMMON/SITE4/MEAS(6)
      COMMON/SITE5/FK(11,6)
      DIMENSION DIFF(6)
      REAL MEAS
C                       *****************************
C     SUBROUTINE DXSPLUS PERFORMS THE MEASUREMENT UPDATE OF THE
C     STATE ESTIMATES.
C                       *****************************
      DO 1 I=1,6
      DIFF(I)=MEAS(I)-FS(I)
    1 CONTINUE
      CALL MMPY(FK,DIFF,A7,11,6,1)
      CALL MADD(11,1,FS,A7,FS)
      RETURN
      END
```

```
      SUBROUTINE REED(FSDUM,FPDUM,RDUM,QDUM,N,RN)
      DIMENSION FSDUM(N),FPDUM(N,N),RDUM(6,6),QDUM(N,N)
      INTEGER N,RN
C                         ******************************
C     SUBROUTINE REED READS THE VALUES REQUIRED BY EIGEN.
C                         ******************************
      READ*,RN
      DO 1 I=1,N
      READ*,FSDUM(I)
1     CONTINUE
      DO 3 J=1,N
      DO 2 I=1,N
      FPDUM(I,J)=0
      IF(I.EQ.J) READ*,FPDUM(I,J)
2     CONTINUE
3     CONTINUE
      DO 5 J=1,6
      DO 4 I=1,6
      RDUM(I,J)=0
      IF(I.EQ.J) READ*,RDUM(I,J)
4     CONTINUE
5     CONTINUE
      DO 7 J=1,N
      DO 6 I=1,N
      QDUM(I,J)=0
      IF(I.EQ.J) READ*,QDUM(I,J)
6     CONTINUE
7     CONTINUE
      RETURN
      END
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
      SUBROUTINE MMPY(A,B,C,M,K,N)
C                           ********************************
C     SUBROUTINE MMPY MULTIPLIES TWO MATRICIES TO PRODUCE C(M,N)
C                           ********************************
      DIMENSION A(M,K),B(K,N),C(M,N)
       DO 10 J=1,N
      DO 10  I=1,M
      S=0
      DO 11 L=1,K
   11 S=S+A(I,L)*B(L,J)
   10 C(I,J)=S
      RETURN
      END
```

```
      SUBROUTINE MTRAN (K,N,G,GT)
      DIMENSION G(K,N),GT(N,K)
C                         ***************************
C     SUBROUTINE MTRAN TANSPOSES THE MATRIX
C                         ***************************
      DO 1 I=1,K
      DO 2 J=1,N
 2    GT(J,I)=G(I,J)
 1    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE  MSUB(K,N,A,B,C)
      DIMENSION A(K,N),B(K,N), C(K,N)
C                          ******************************
C     SUBROUTINE MSUB SUBTRACTS TWO MATRICES
C                          ******************************
      DO 4  J=1,N
      DO 4  I=1,K
    4 C(I,J)= A(I,J)-B(I,J)
      RETURN
      END
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
      SUBROUTINE MADD(K,N,A,B,C)
       DIMENSION A(K,N), B(K,N),C(K,N)
C                                 ******************************
C     SUBROUTINE MADD ADDS TWO MATRICES
C                                 ******************************
       DO 3 J=1,N
      DO 3 I=1,K
 3    C(I,J)= A(I,J)+B(I,J)
      RETURN
      END
```

## Appendix E

This appendix contains the graphical results of the Monte Carlo analyses of the 11 State Filter. For each K Set trajectory, the filter states are initialized to the initial true values. The tuning parameters used in the filter for all K Set trajectories are (Note: The state vector is $[\theta_1, \theta_2, \omega_1, \omega_2, R_{tm}, \dot{R}_{tm}, A_{L1}, A_{L2}, n_f, \tau_f, (m/s)_f]^T$):

$$R = \begin{bmatrix} 3.E-5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3.E-5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3.E-5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3.E-5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5.E-4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.E-4 \end{bmatrix}$$

$$P_o = \begin{bmatrix} 4.E-6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4.E.6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3.E-5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3.E-5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5.E-4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.E-4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & .009 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & .05 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.E13 \end{bmatrix}$$

$$
Q = \begin{bmatrix}
1.E-5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1.E-5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1.E-5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1.E-5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1.E-2 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 2.E-3 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 3.E-3 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 3.E-3 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & .001 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4.E12
\end{bmatrix}
$$

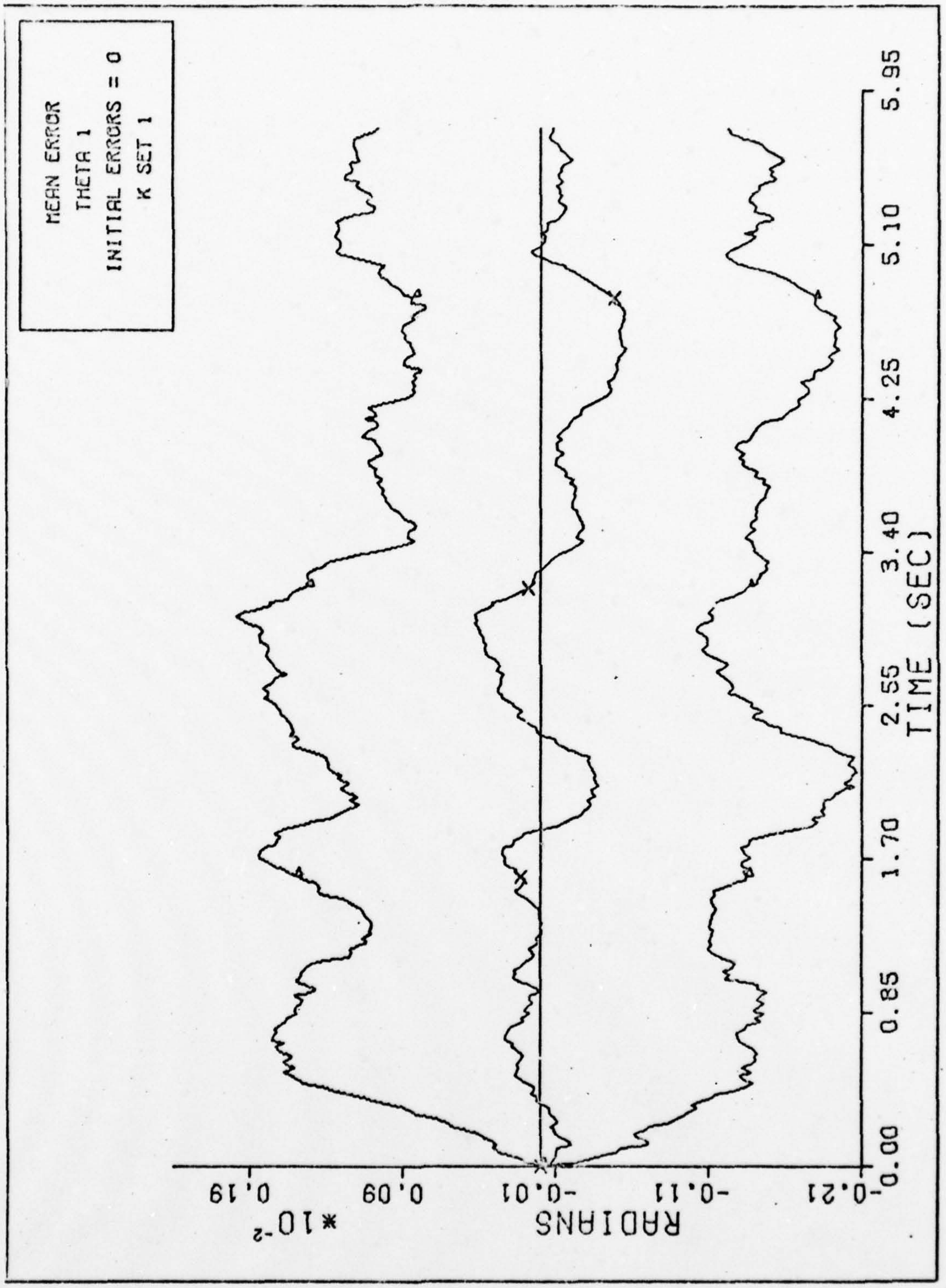



Fig. E-1 THETA 1 MEAN ERROR, 11 STATE FILTER

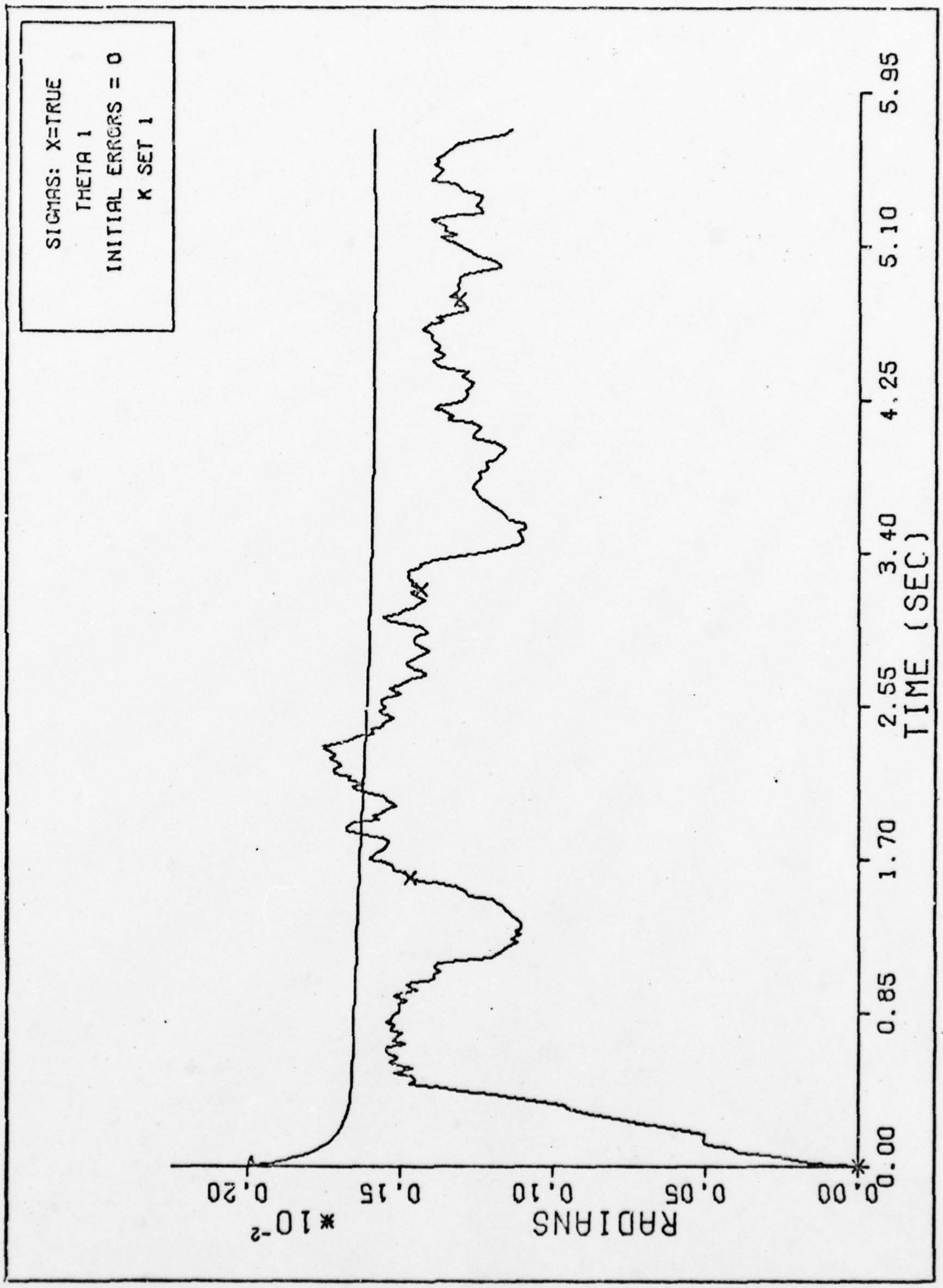



Fig. E-2 THETA 1 FILTER & TRUE SIGMAS, 11 STATE FILTER

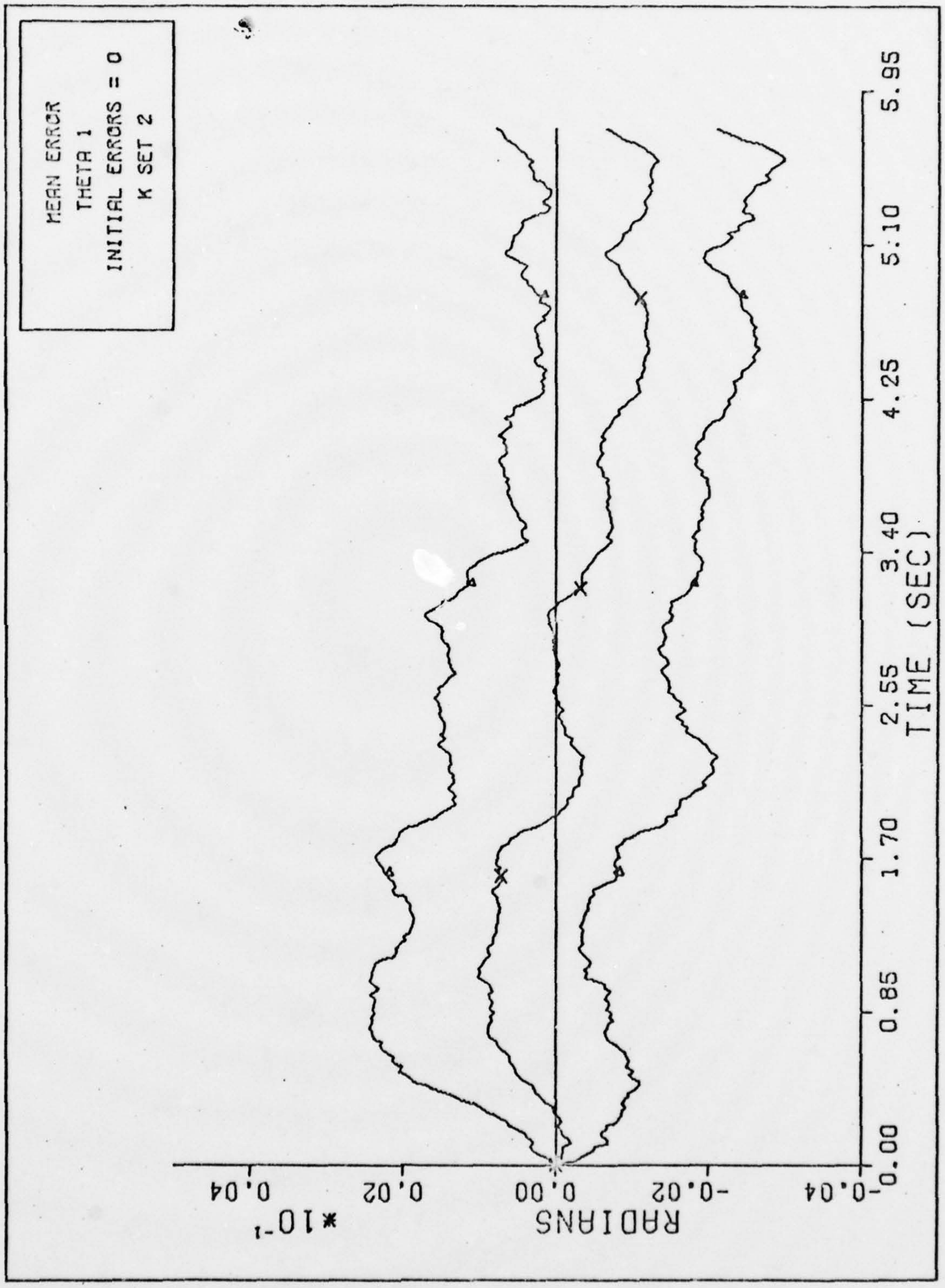



Fig. E-3 THETA 1 MEAN ERROR, 11 STATE FILTER

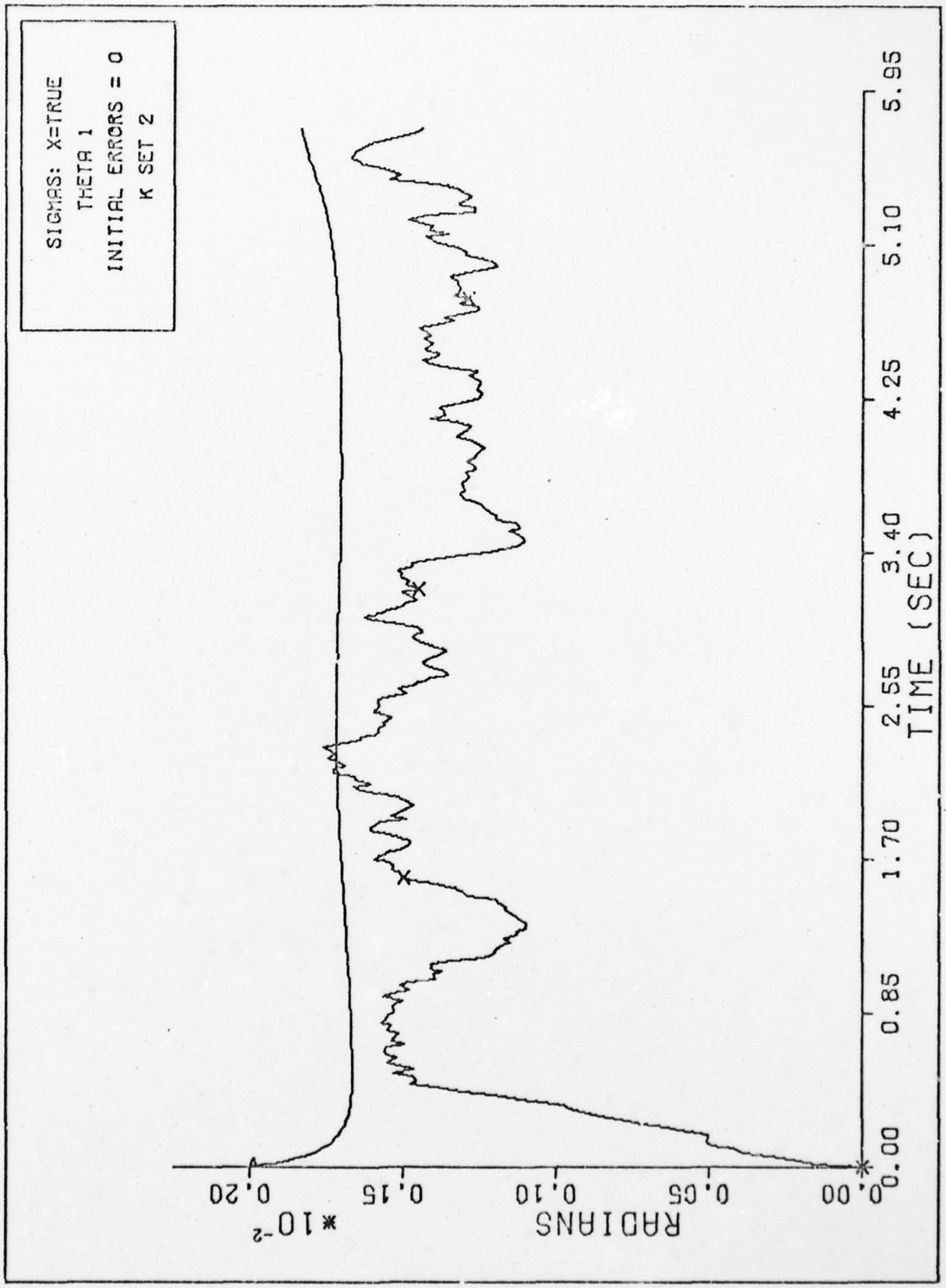



Fig. E-4 THETA 1 FILTER & TRUE SIGMAS, 11 STATE FILTER

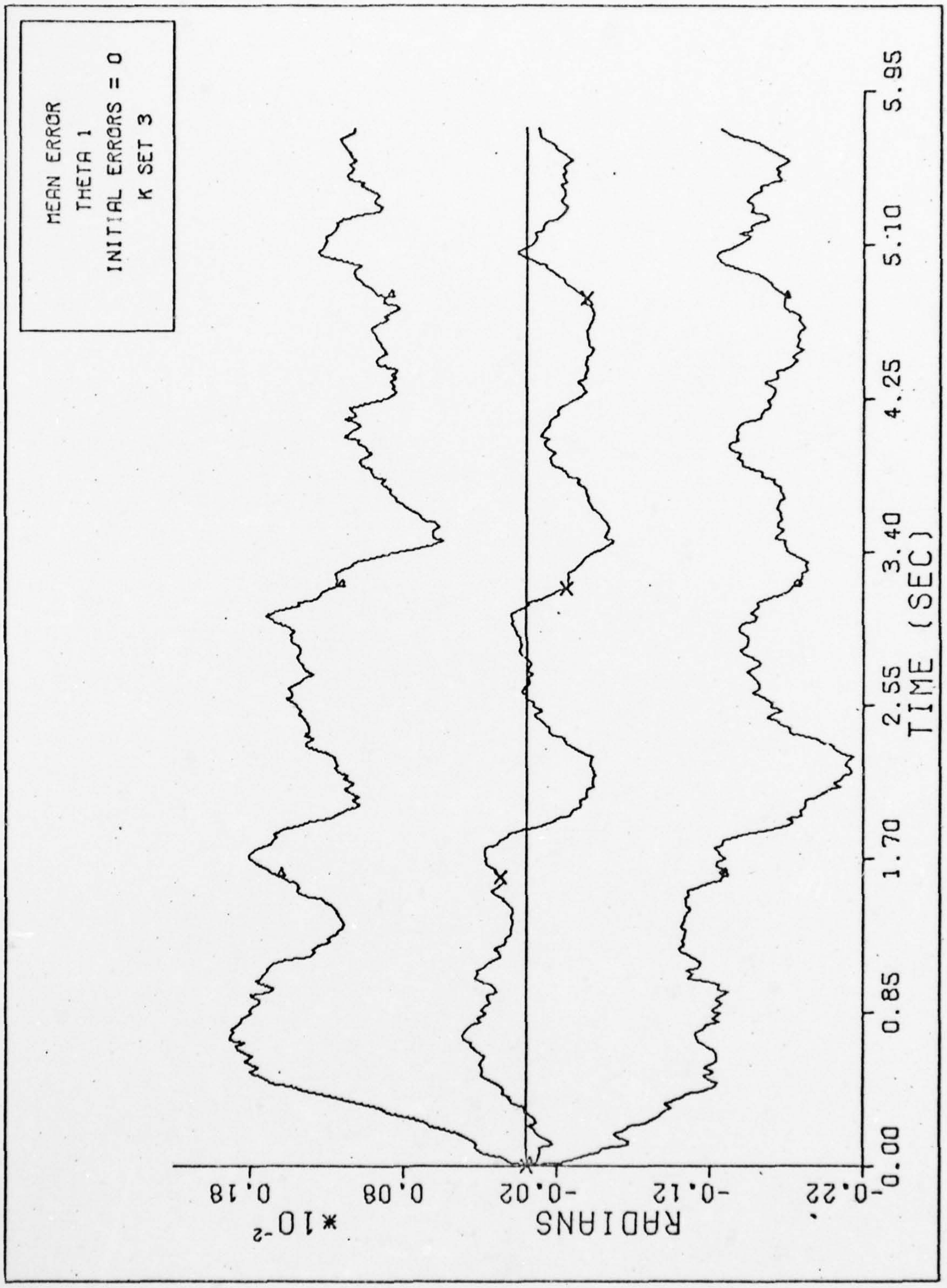



Fig. E-5 THETA 1 MEAN ERROR, 11 STATE FILTER

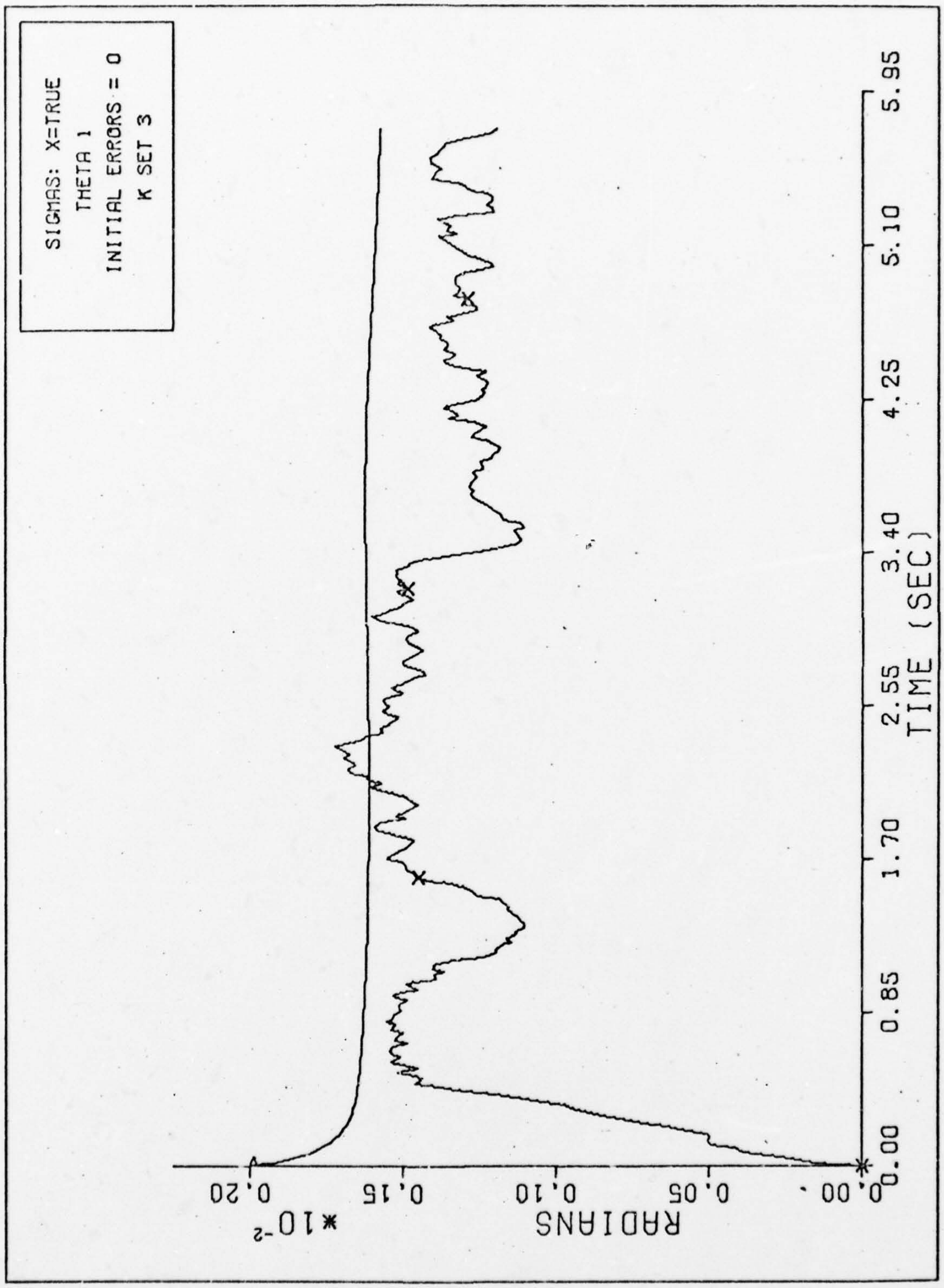



Fig. E-6 THETA 1 FILTER & TRUE SIGMAS, 11 STATE FILTER

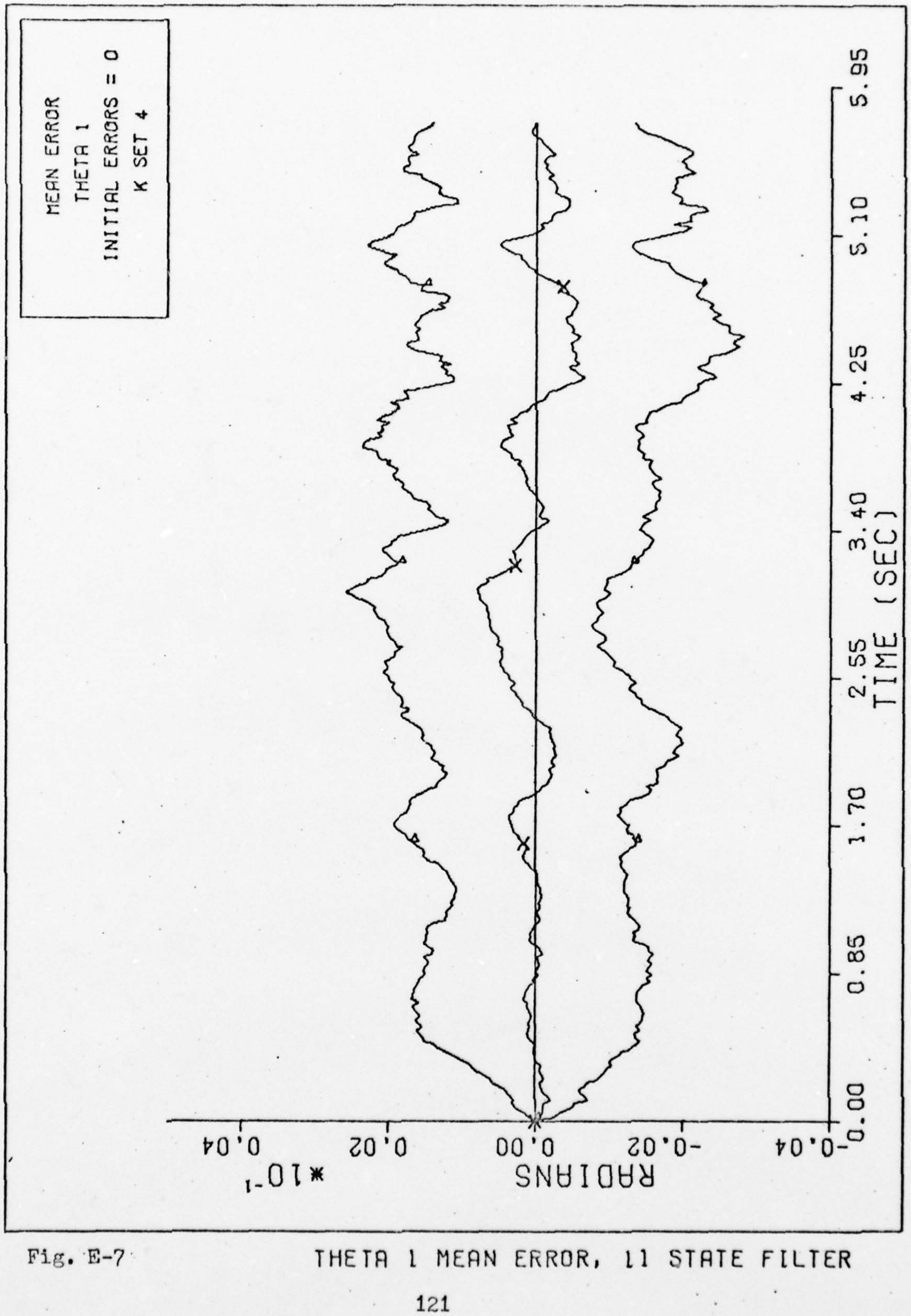



Fig. E-7 THETA 1 MEAN ERROR, 11 STATE FILTER

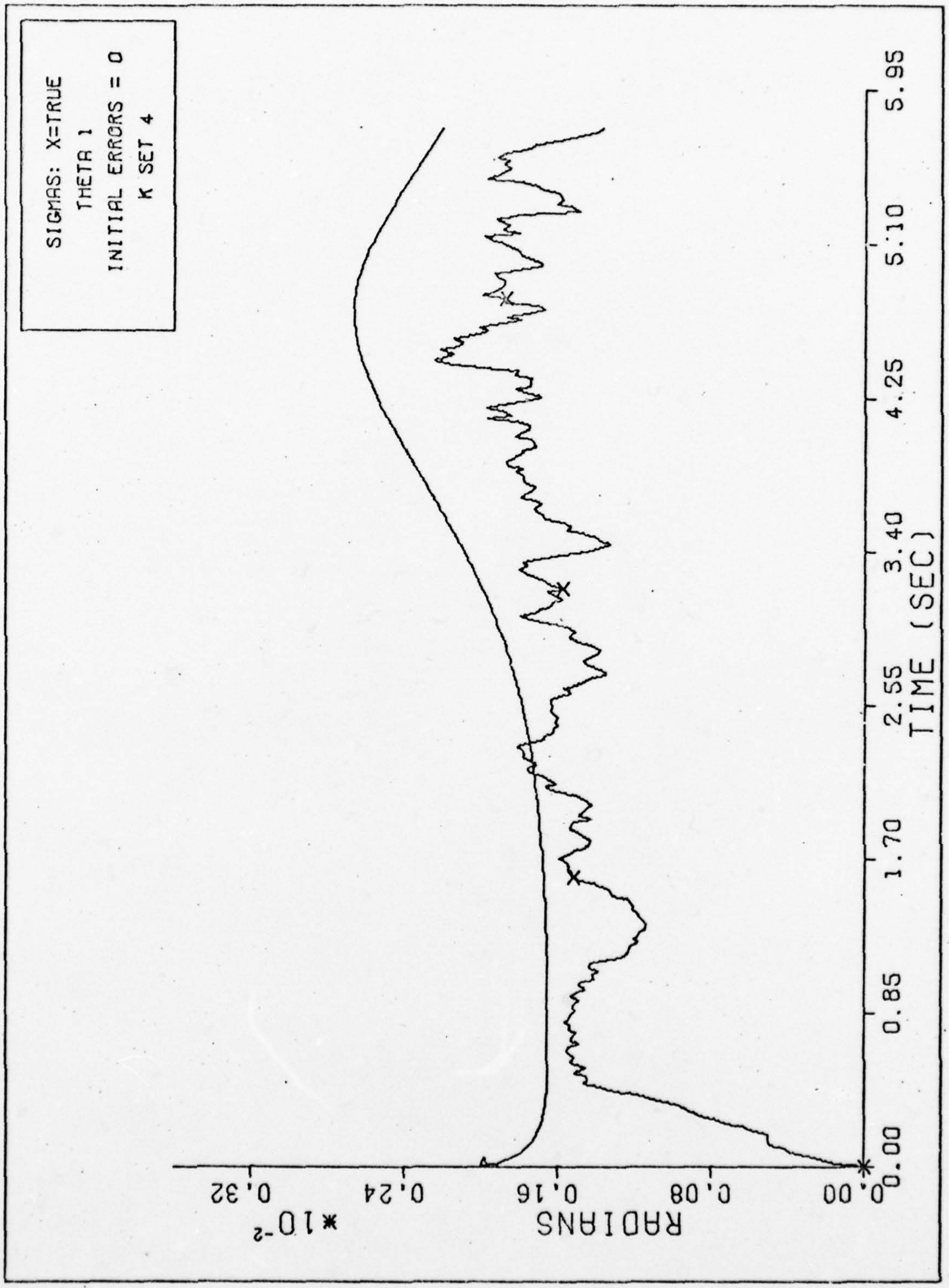



Fig. E-8 THETA 1 FILTER & TRUE SIGMAS, 11 STATE FILTER

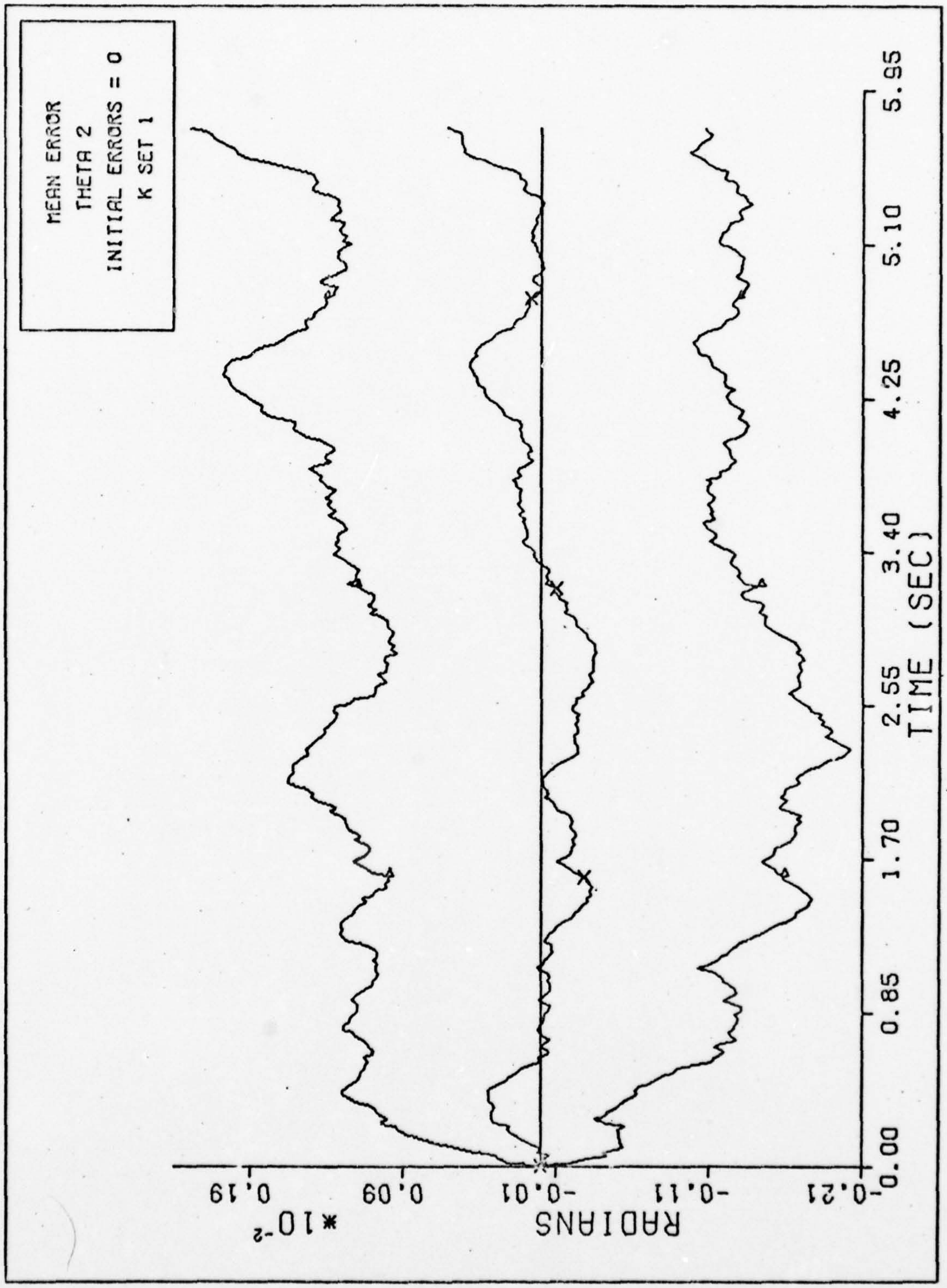



Fig. E-9 THETA 2 MEAN ERROR, 11 STATE FILTER

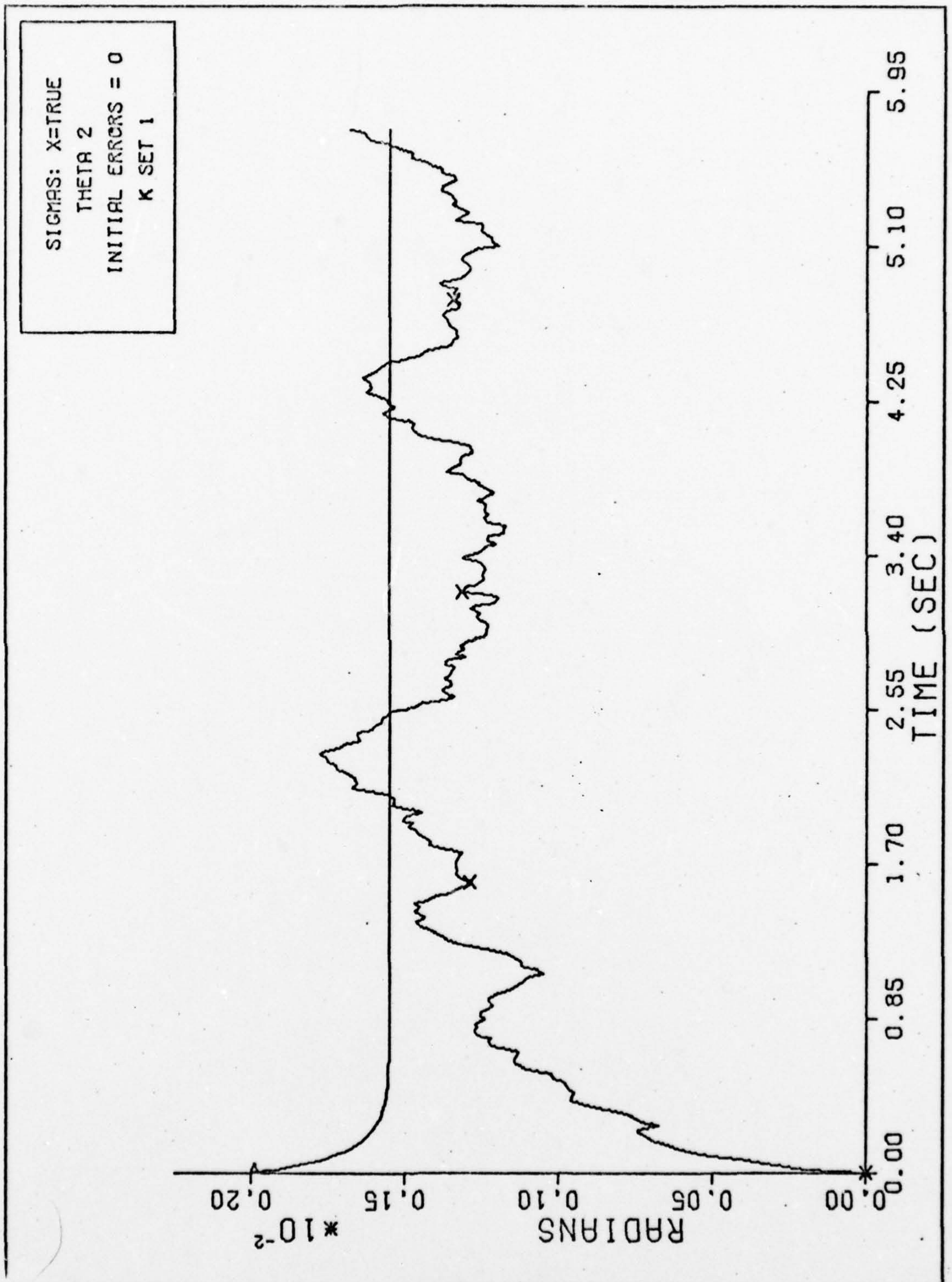



Fig. E-10 THETA 2 FILTER & TRUE SIGMAS, 11 STATE FILTER

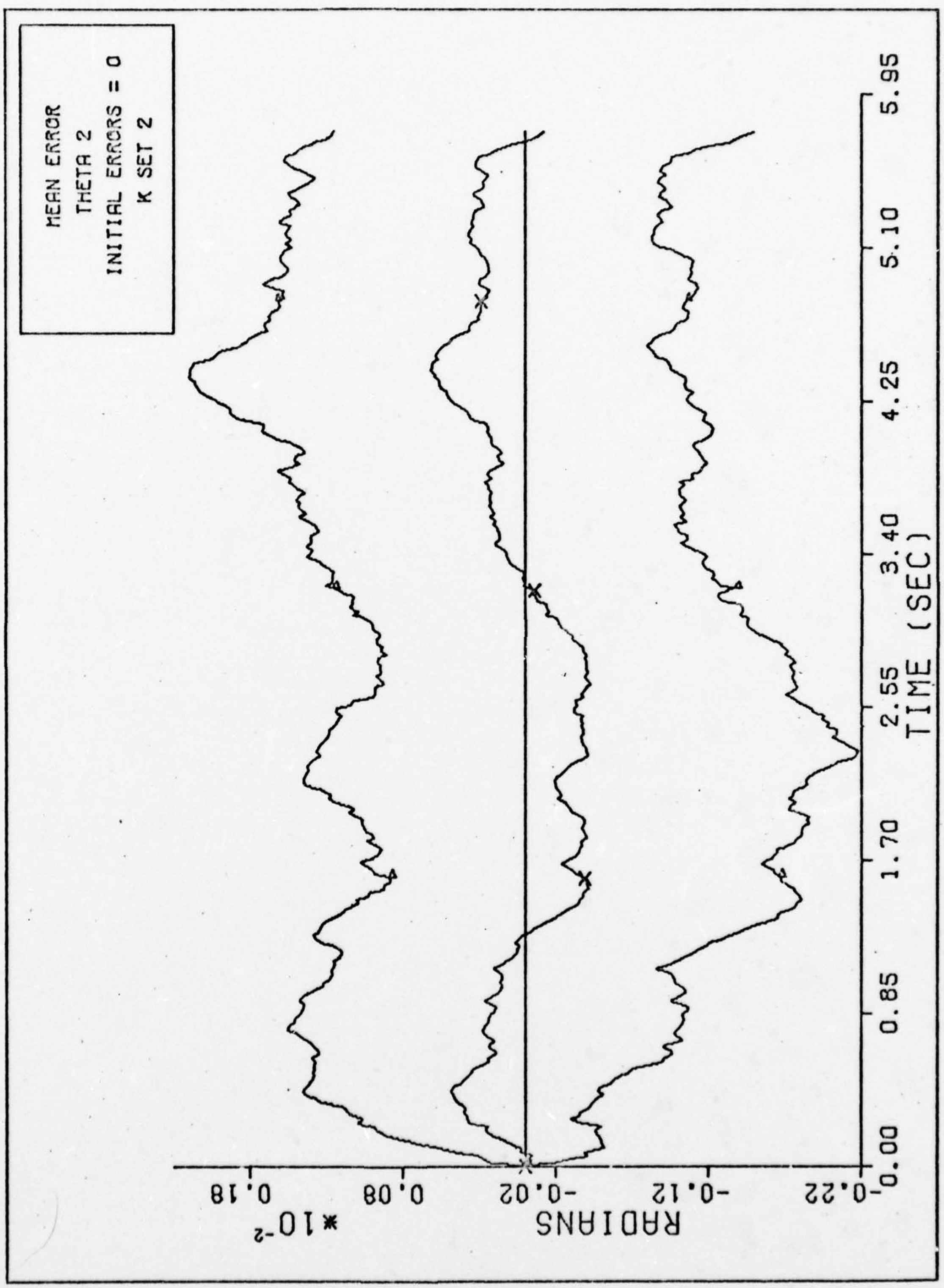



Fig. E-11 THETA 2 MEAN ERROR, 11 STATE FILTER

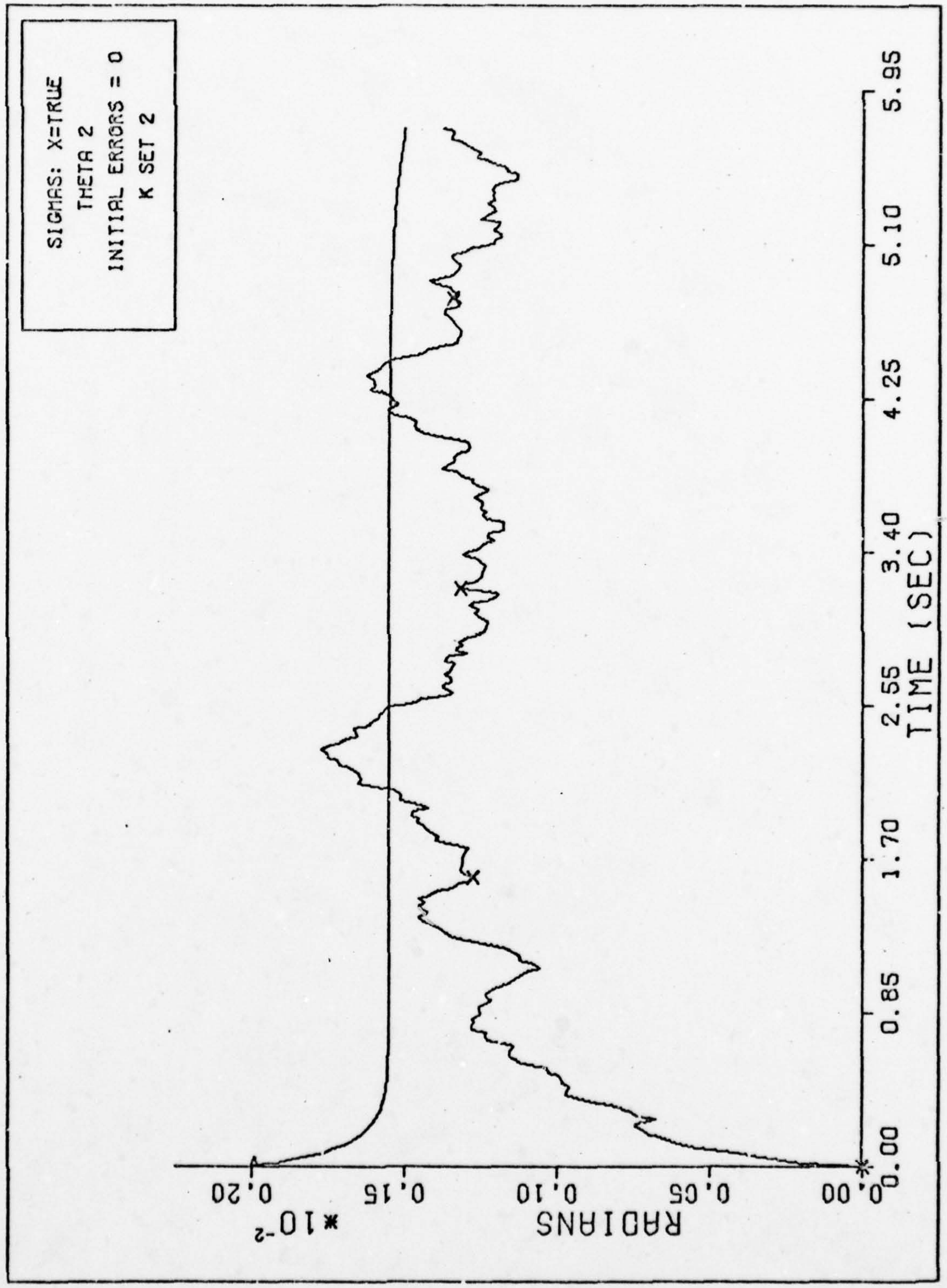



Fig. E-12 THETA 2 FILTER & TRUE SIGMAS, 11 STATE FILTER

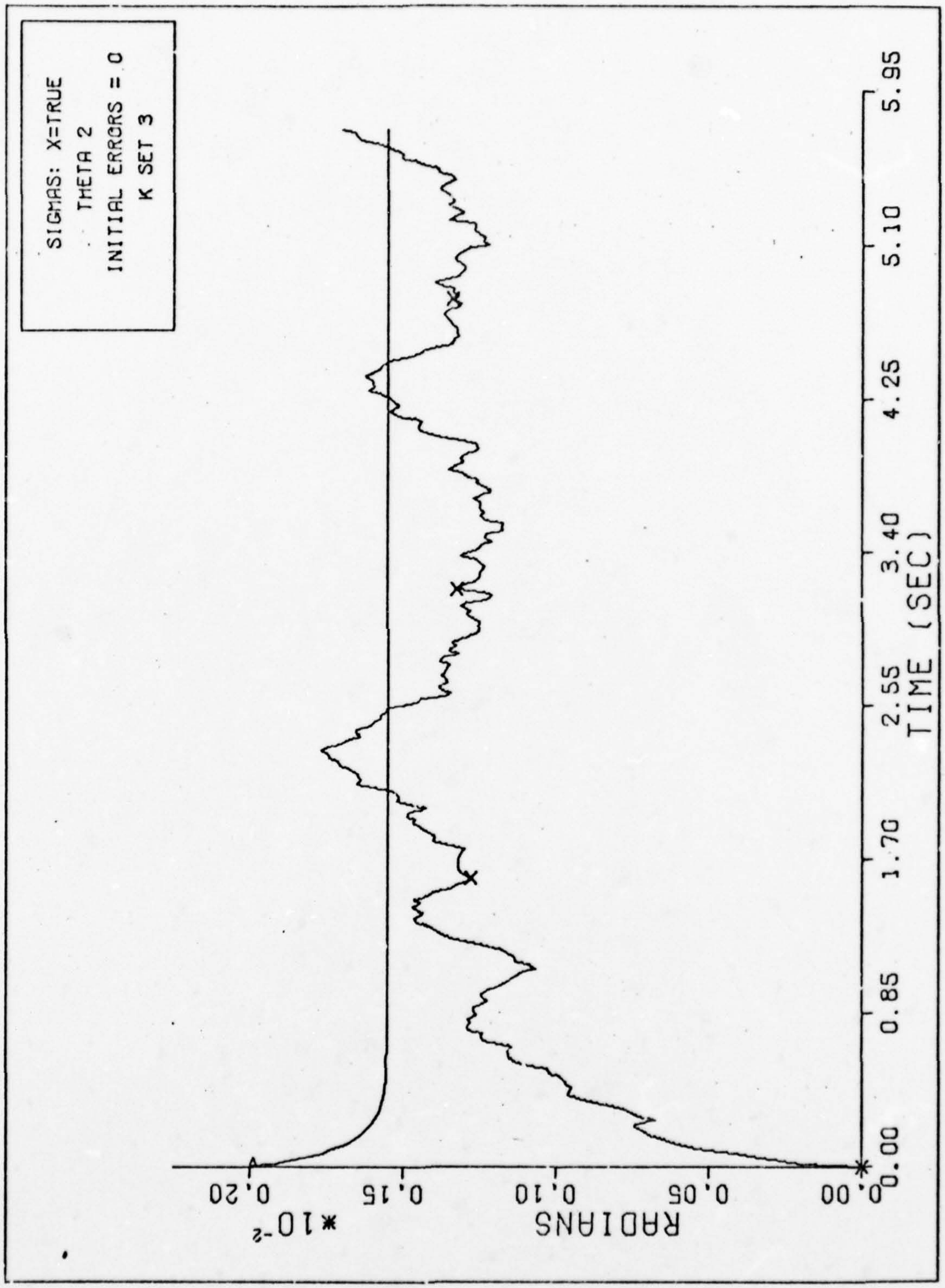



Fig. E-13 THETA 2 FILTER & TRUE SIGMAS, 11 STATE FILTER

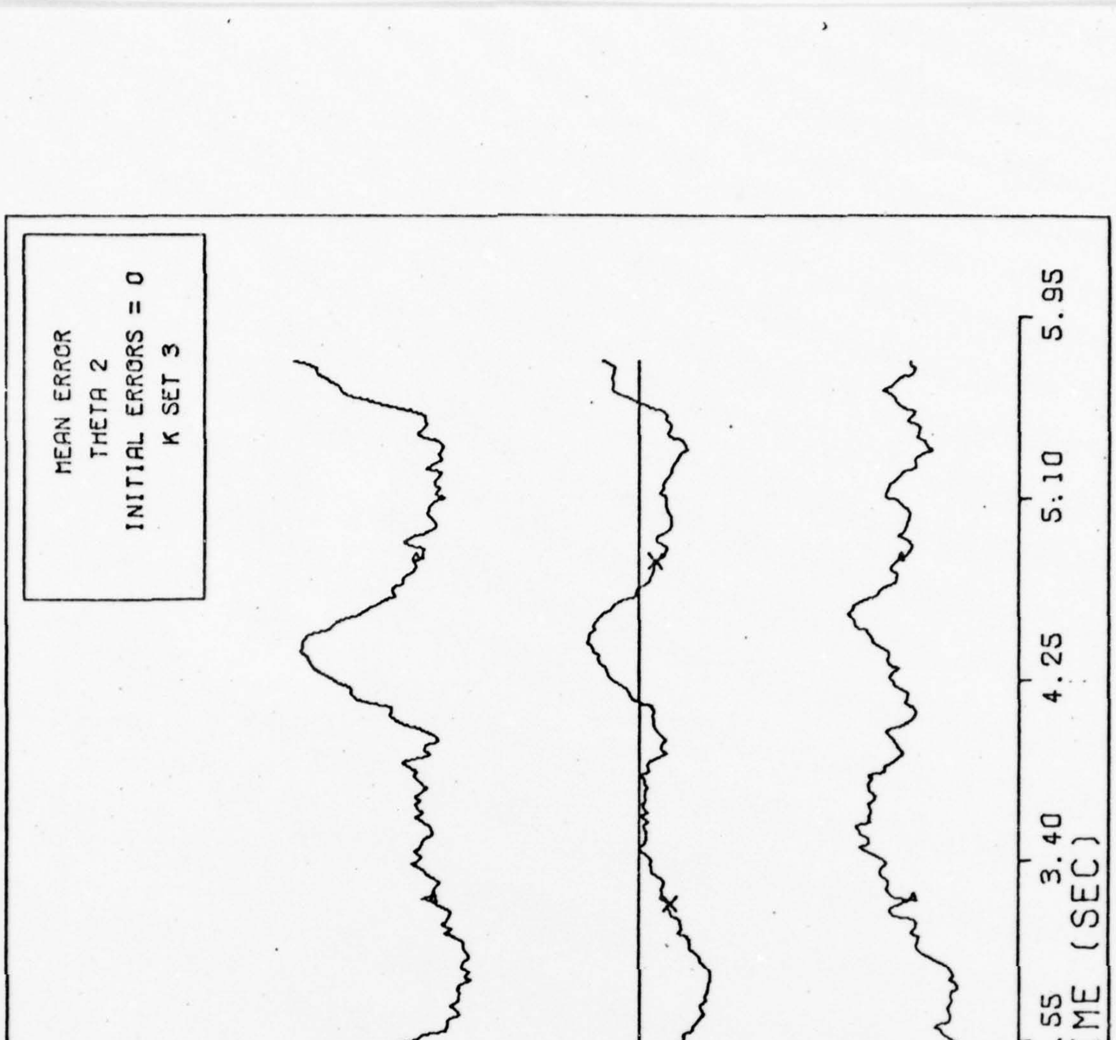



Fig. E-14 THETA 2 MEAN ERROR, 11 STATE FILTER

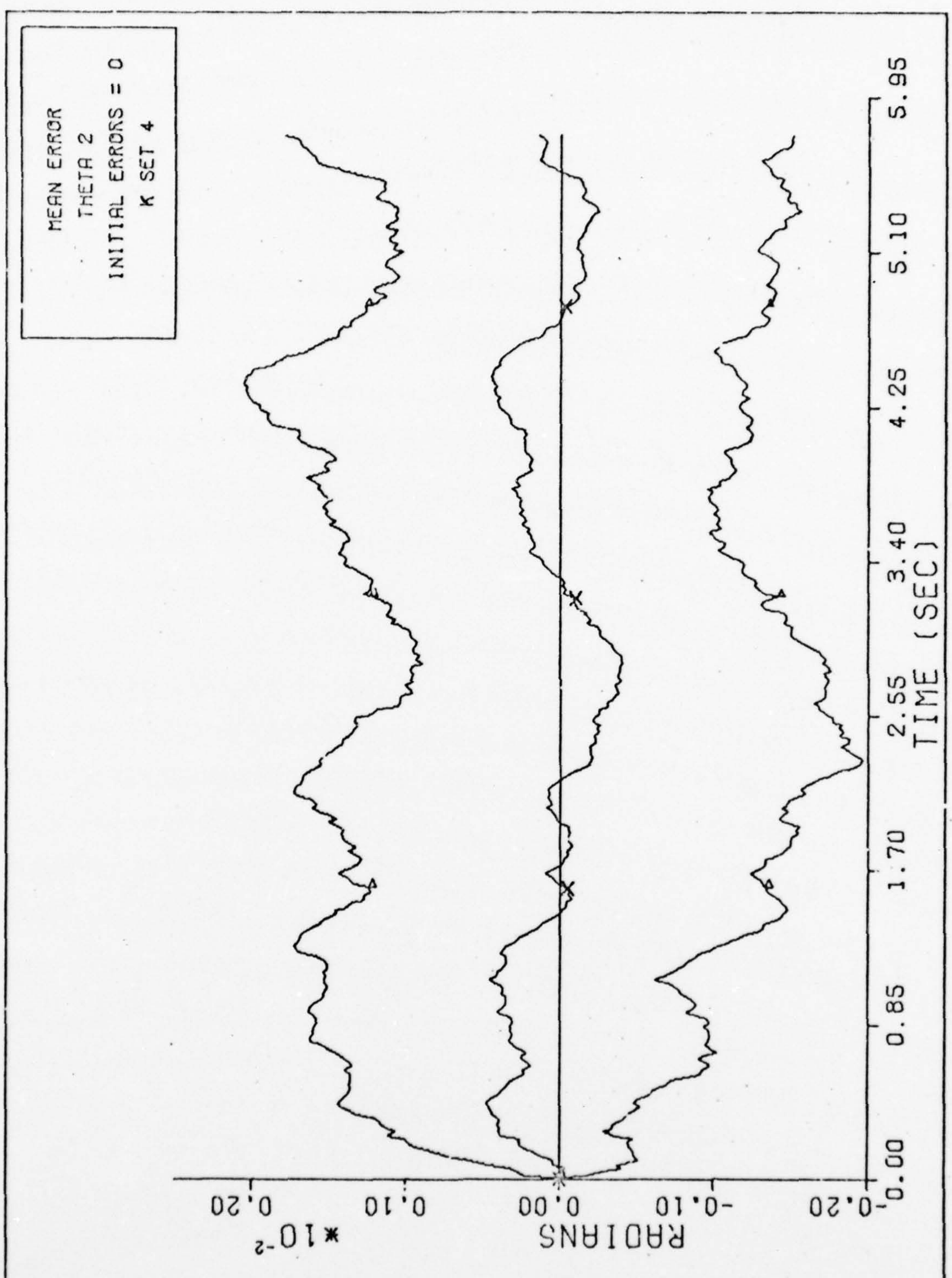



Fig. E-15 THETA 2 MEAN ERROR, 11 STATE FILTER

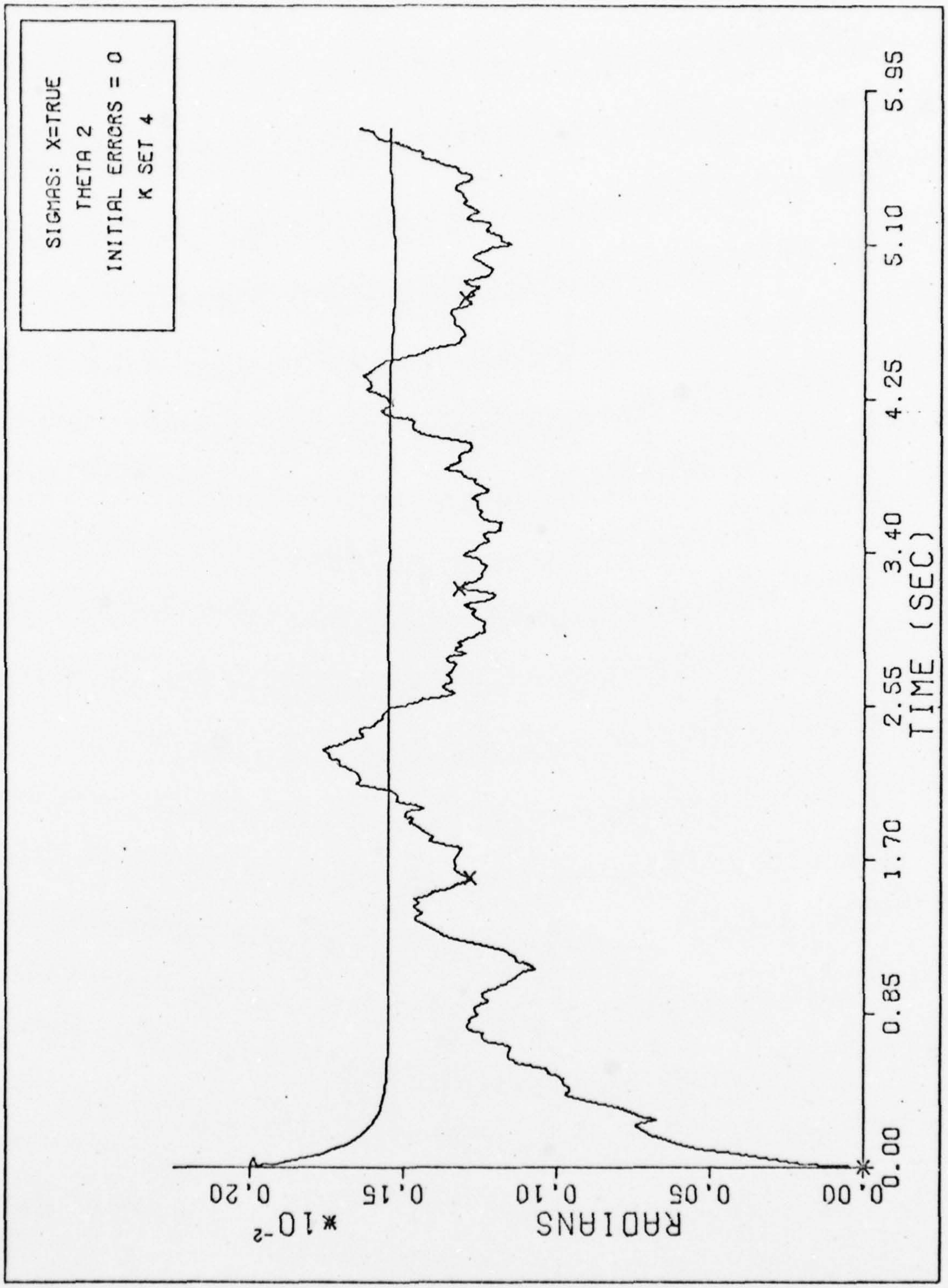



Fig. E-16 THETA 2 FILTER & TRUE SIGMAS, 11 STATE FILTER

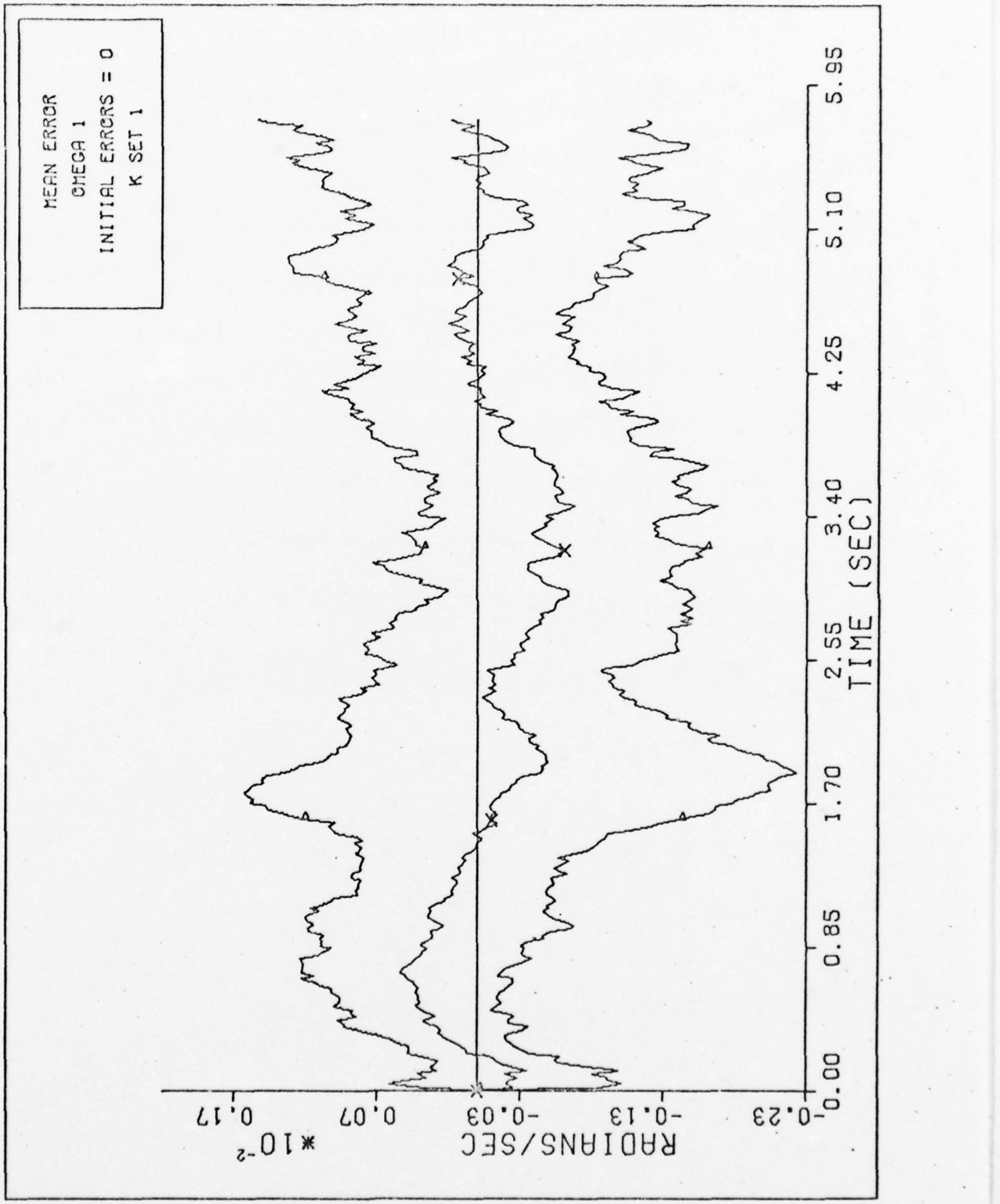



Fig. E-17 OMEGA 1 MEAN ERROR, 11 STATE FILTER

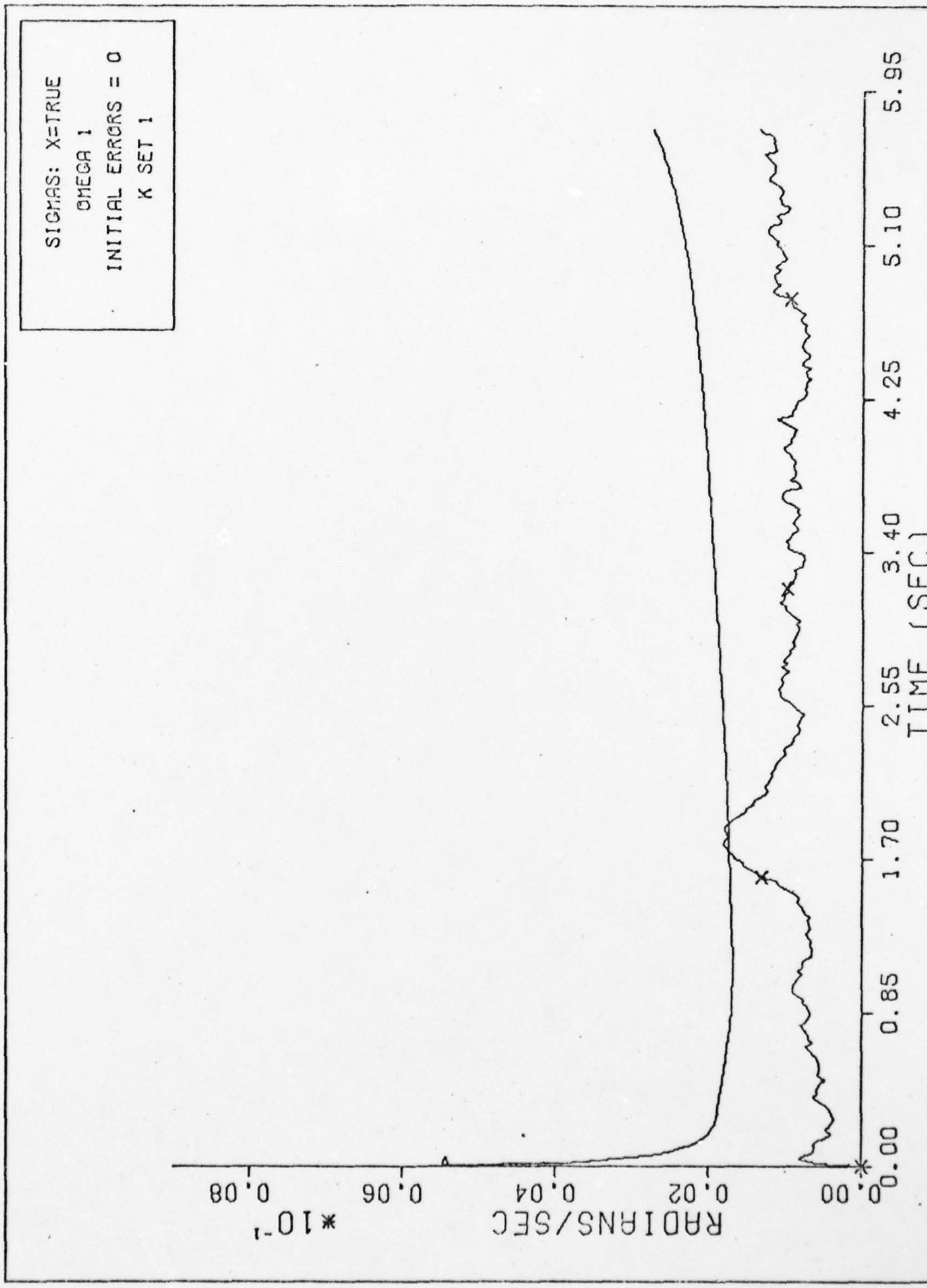



Fig. E-18 OMEGA 1 FILTER & TRUE SIGMAS, 11 STATE FILTER

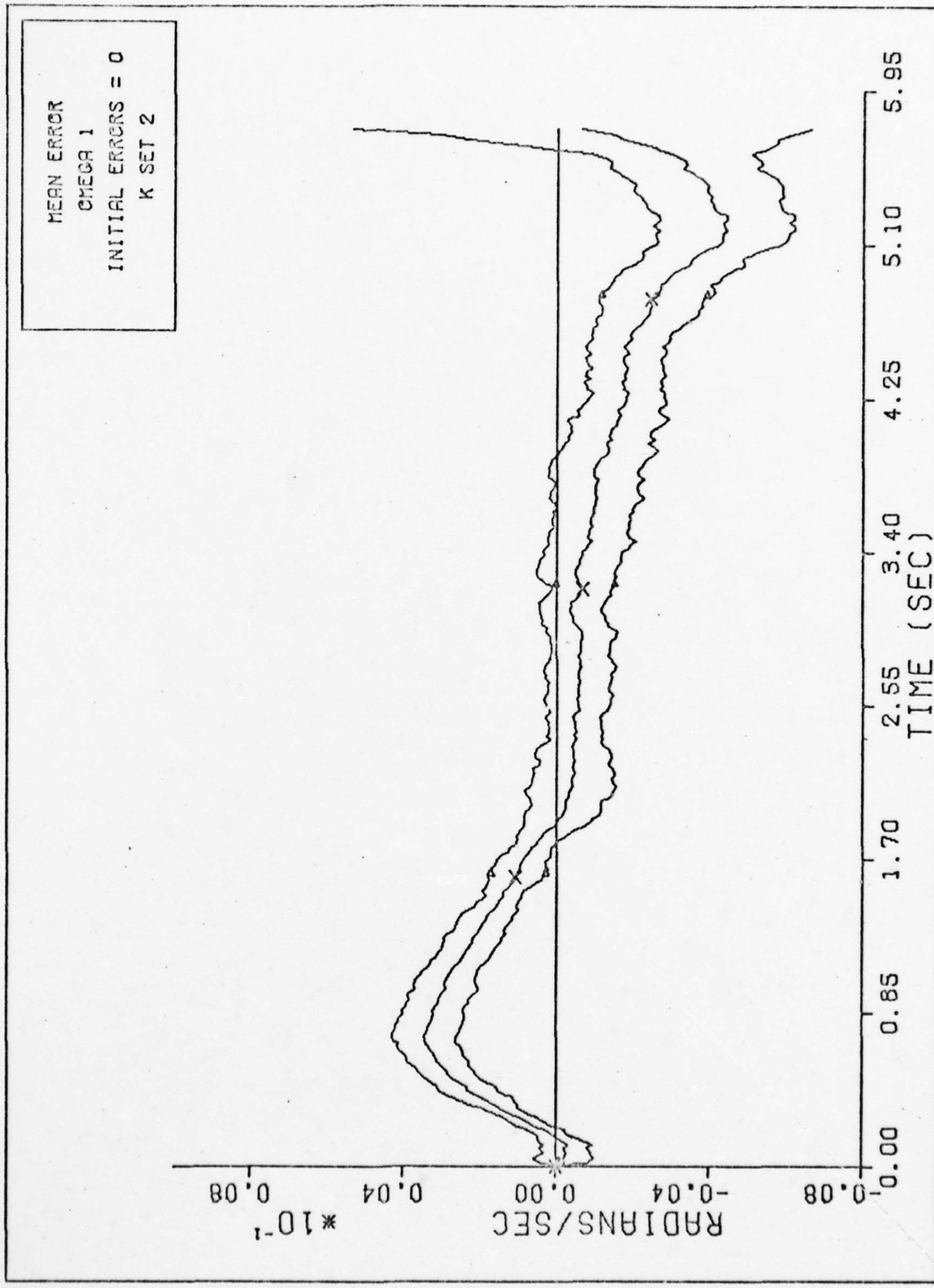



Fig. E-19 OMEGA 1 MEAN ERROR, 11 STATE FILTER

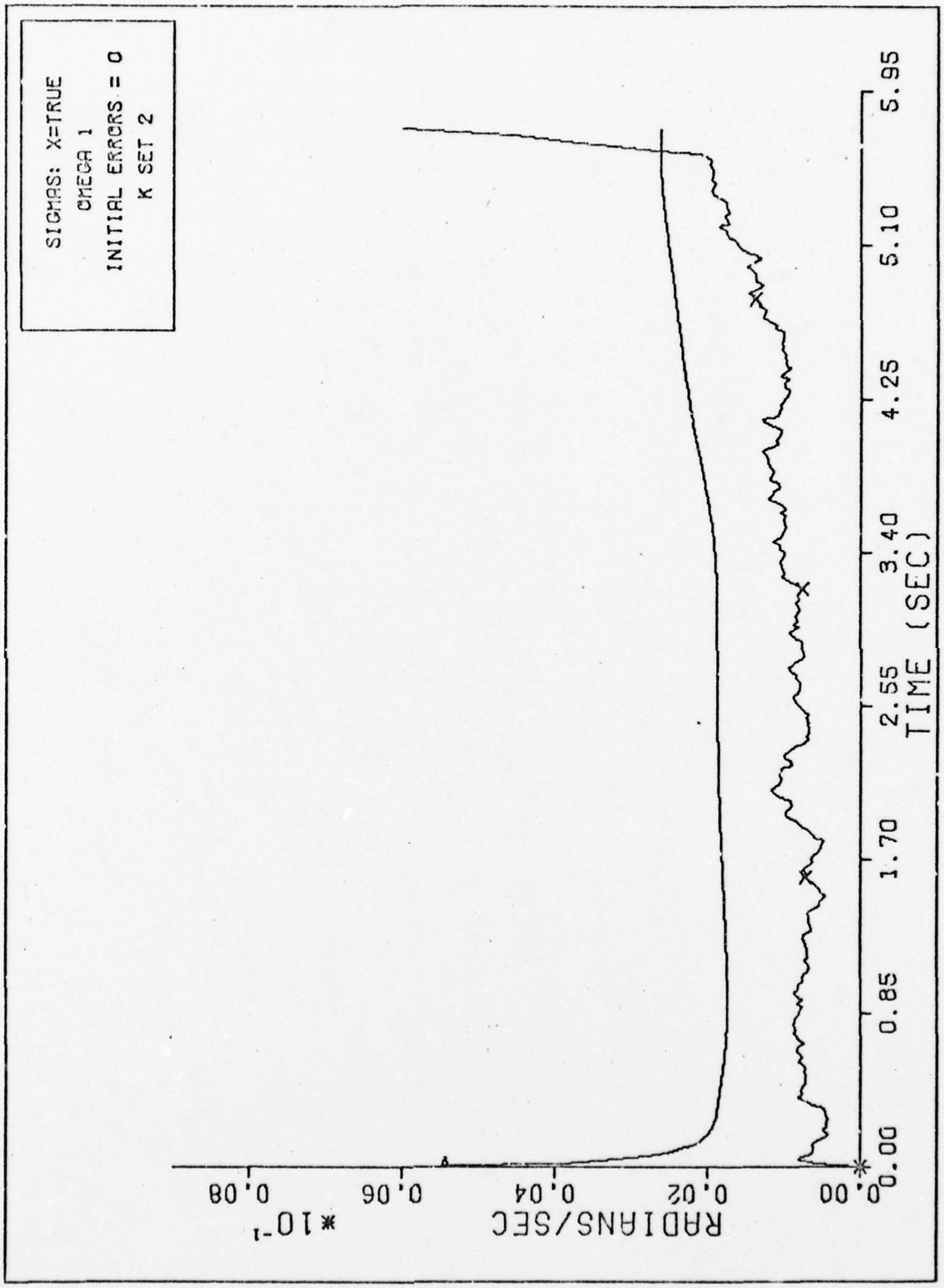



Fig. E-20 OMEGA 1 FILTER & TRUE SIGMAS, 11 STATE FILTER

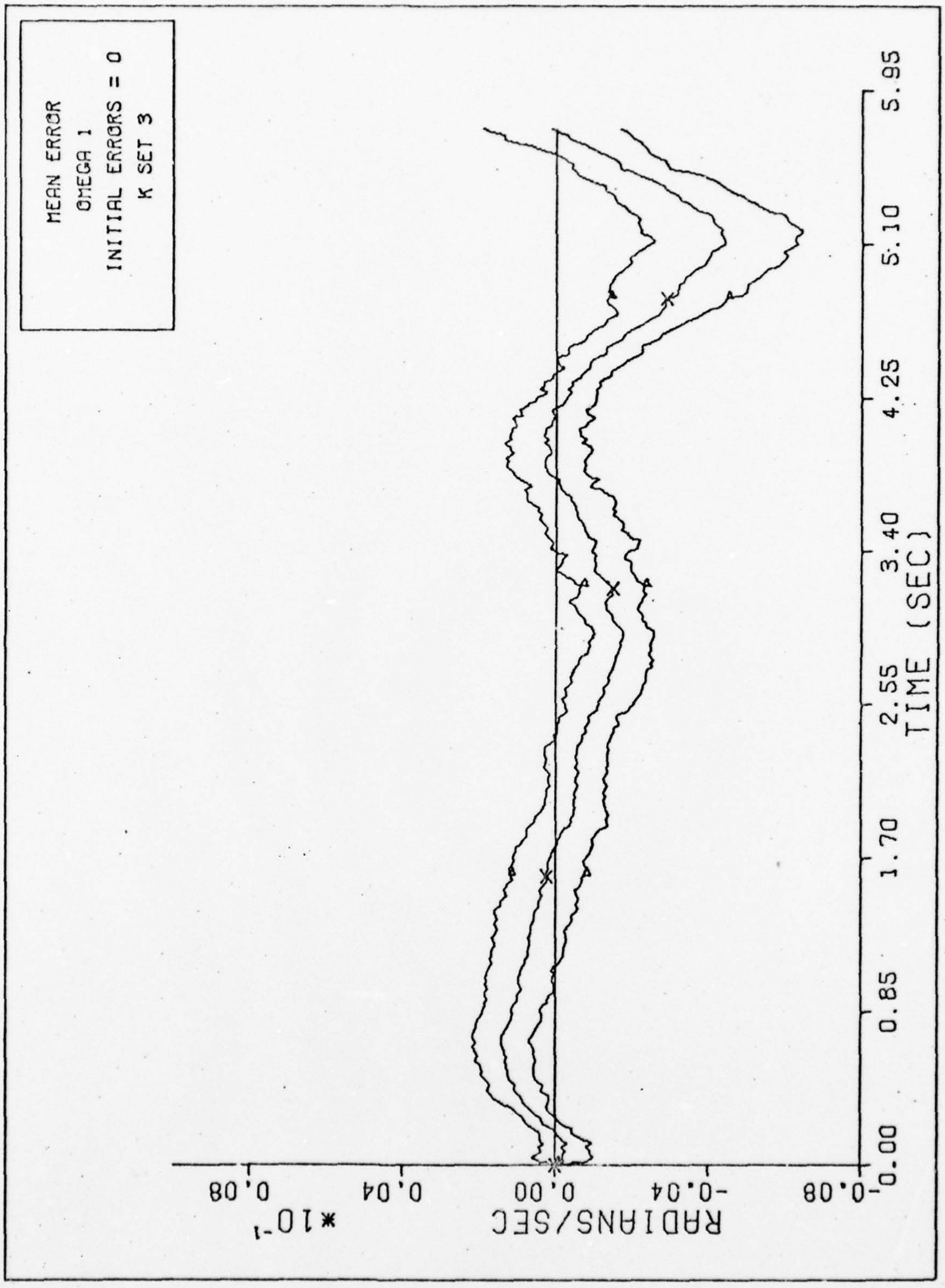



Fig. E-21 OMEGA 1 MEAN ERROR, 11 STATE FILTER

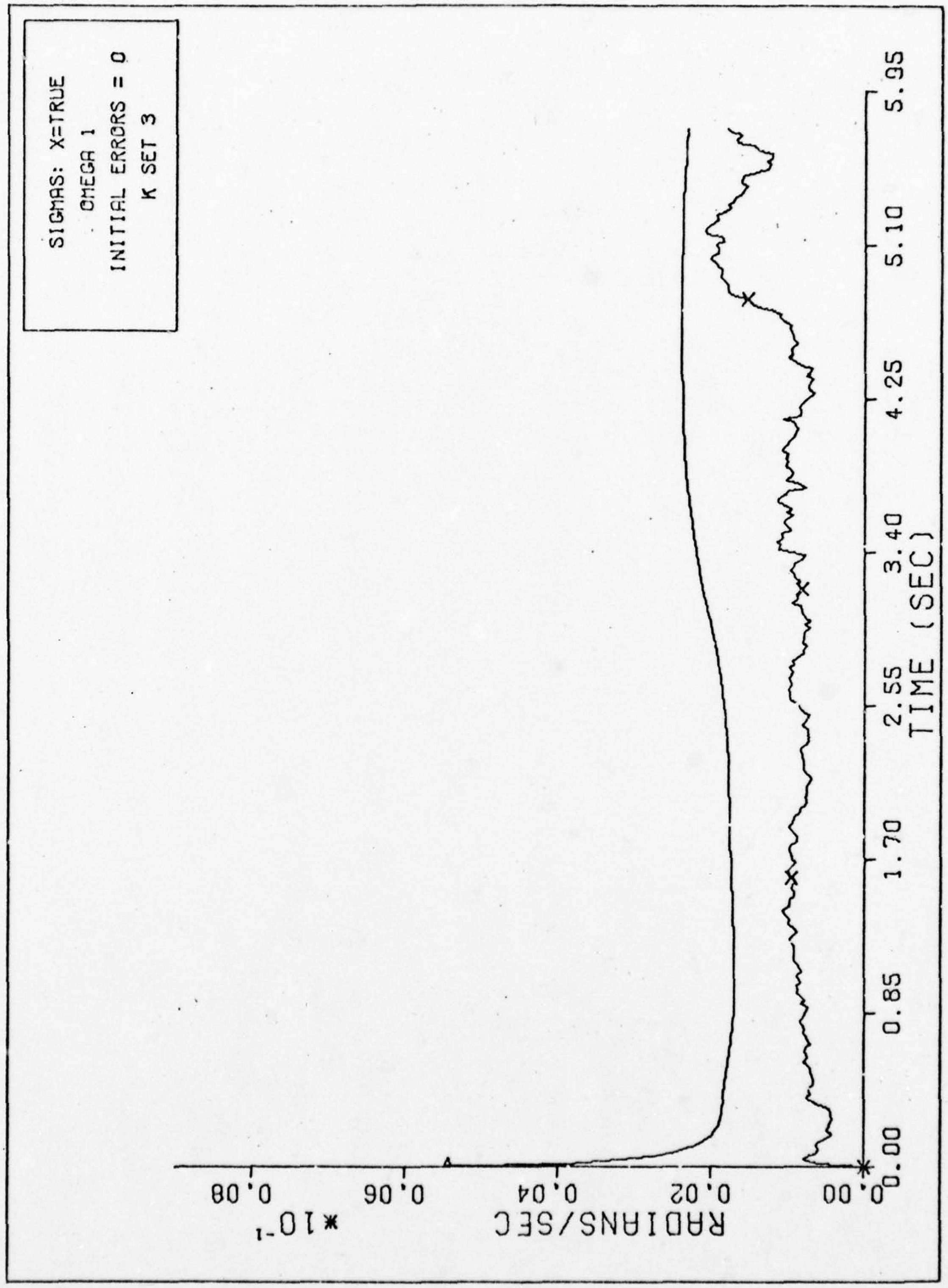



Fig. E-22 OMEGA 1 FILTER & TRUE SIGMAS, 11 STATE FILTER

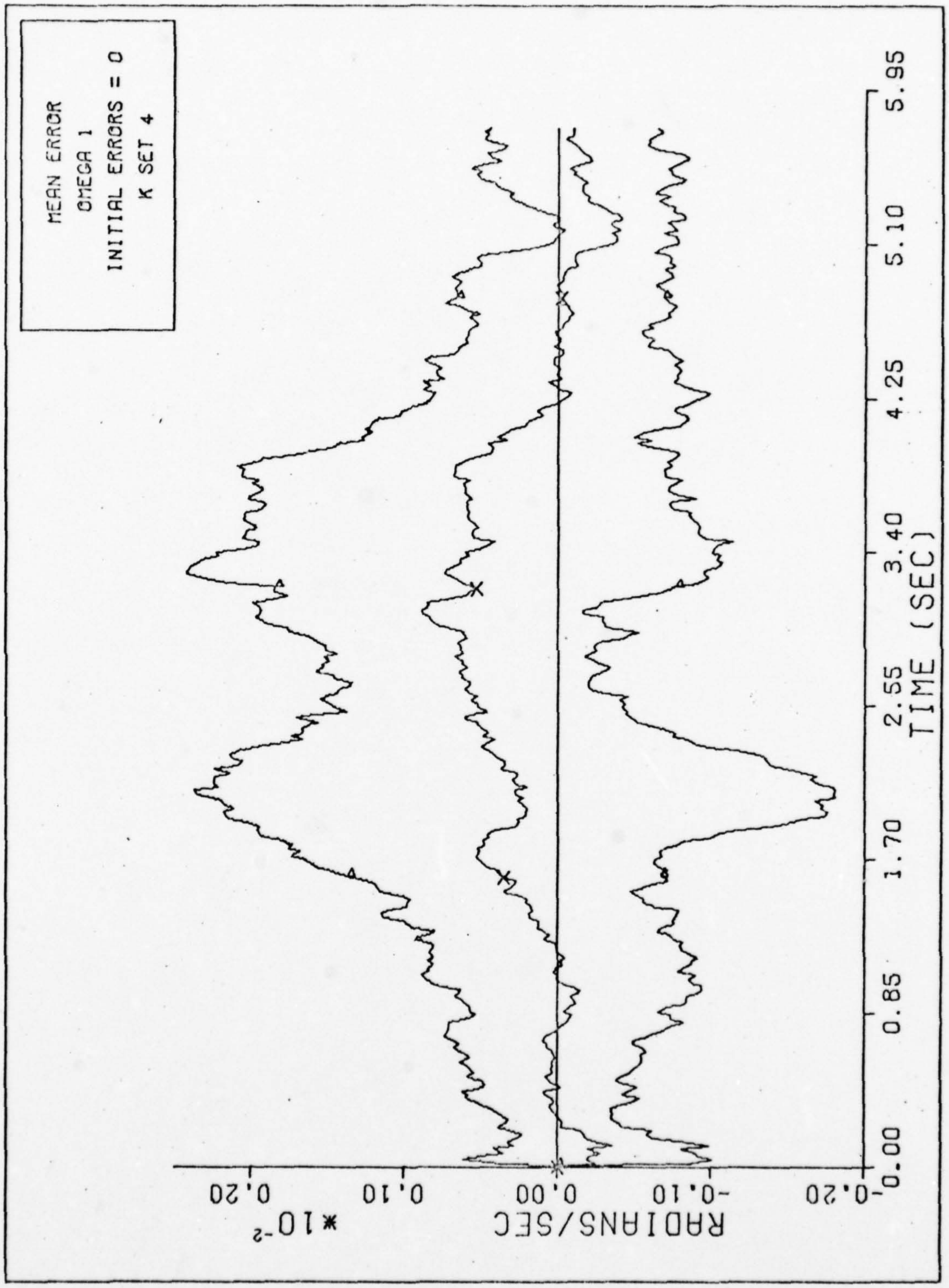



Fig. E-23 OMEGA 1 MEAN ERROR, 11 STATE FILTER

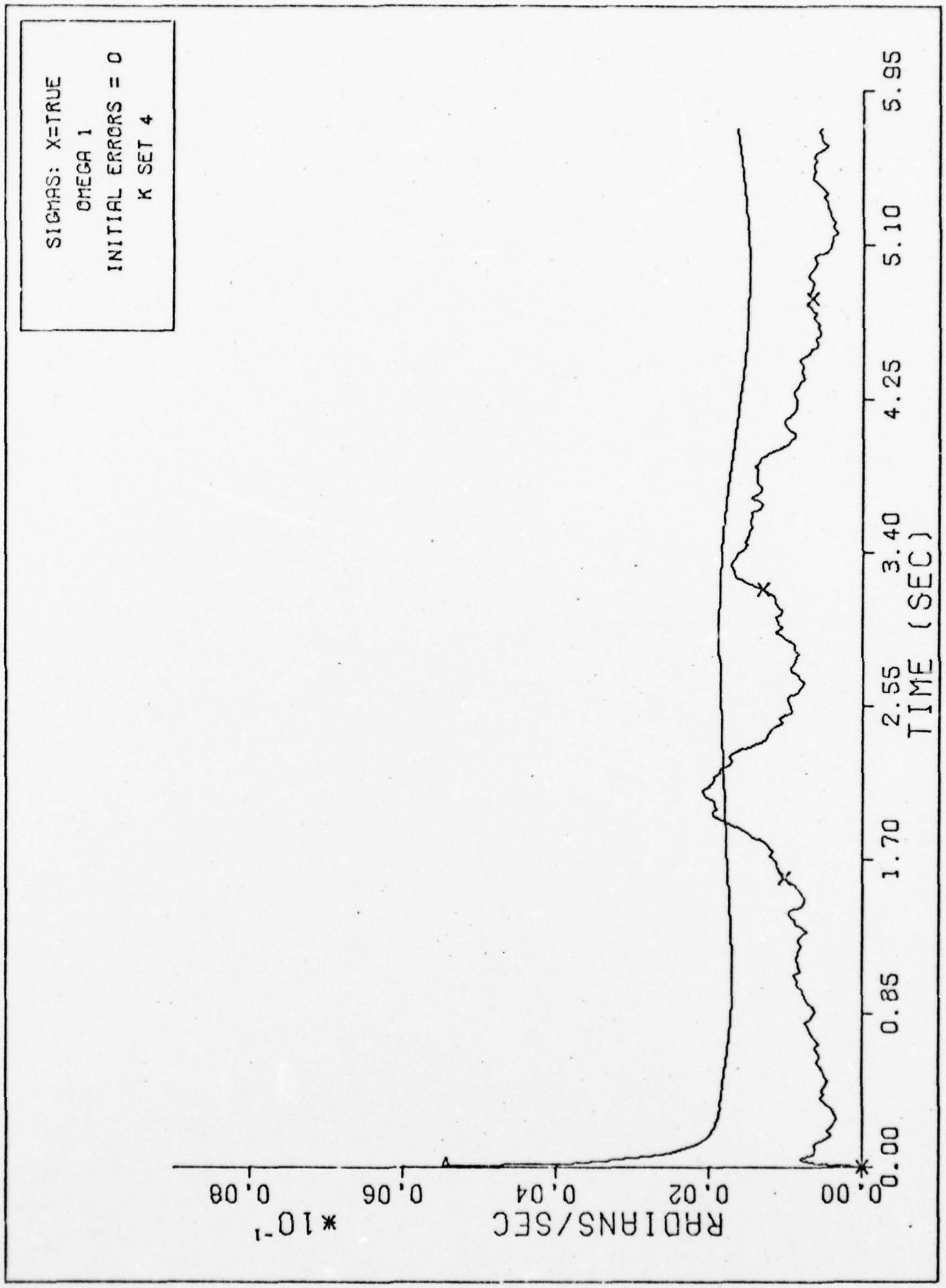



Fig. E-24 OMEGA 1 FILTER & TRUE SIGMAS, 11 STATE FILTER

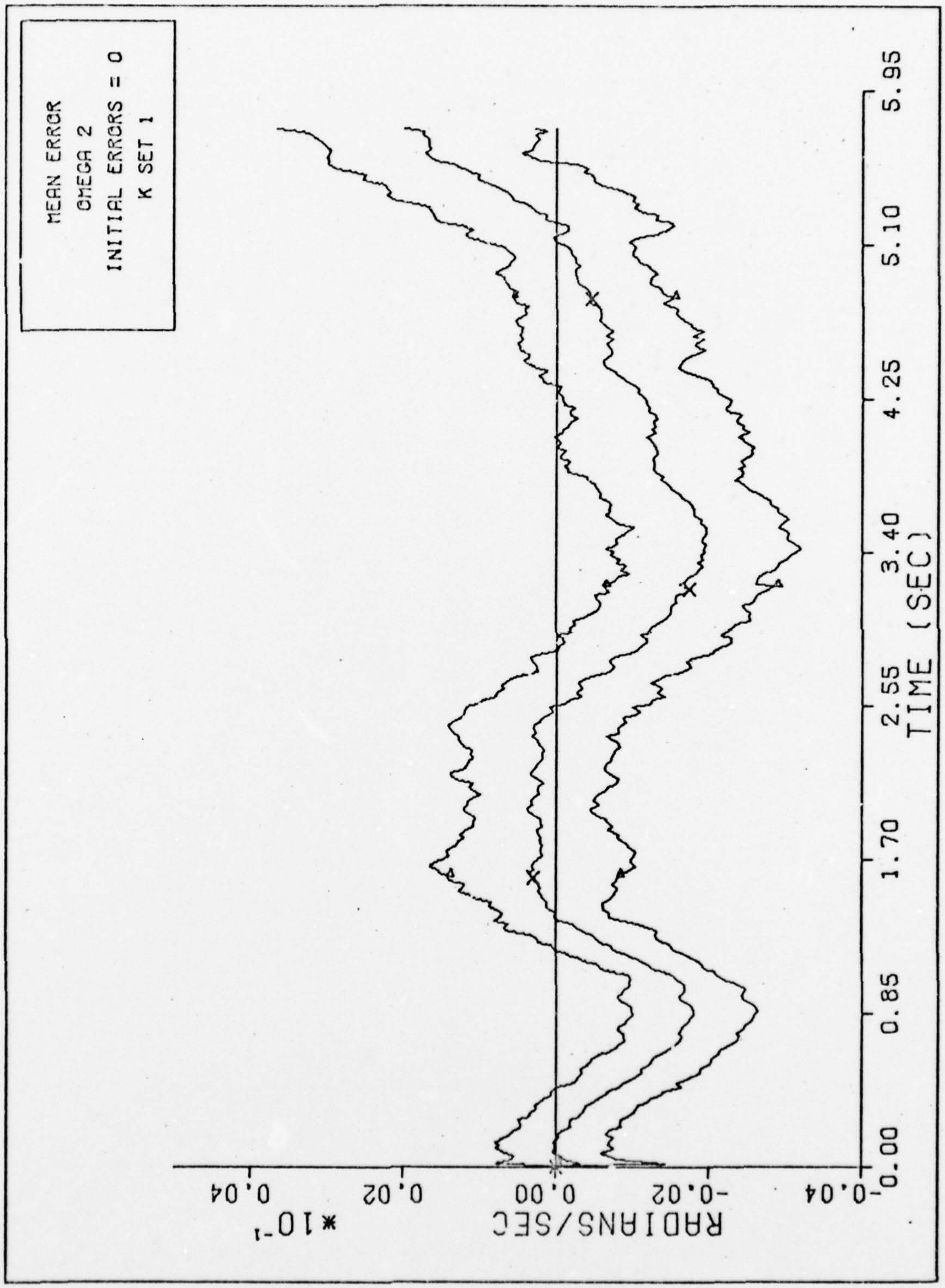



Fig. E-25 OMEGA 2 MEAN ERROR, 11 STATE FILTER

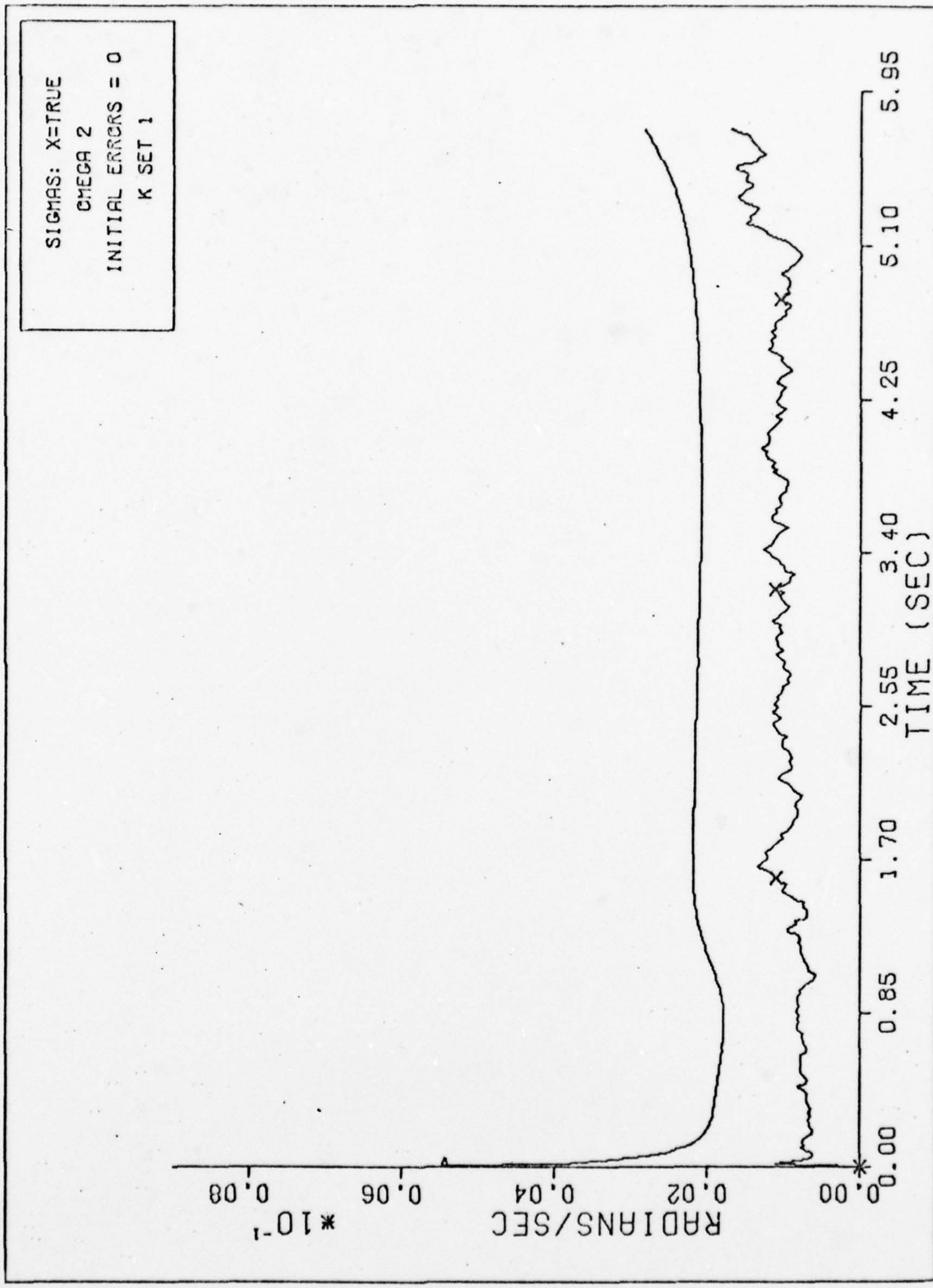



Fig. E-26 OMEGA 2 FILTER & TRUE SIGMAS, 11 STATE FILTER

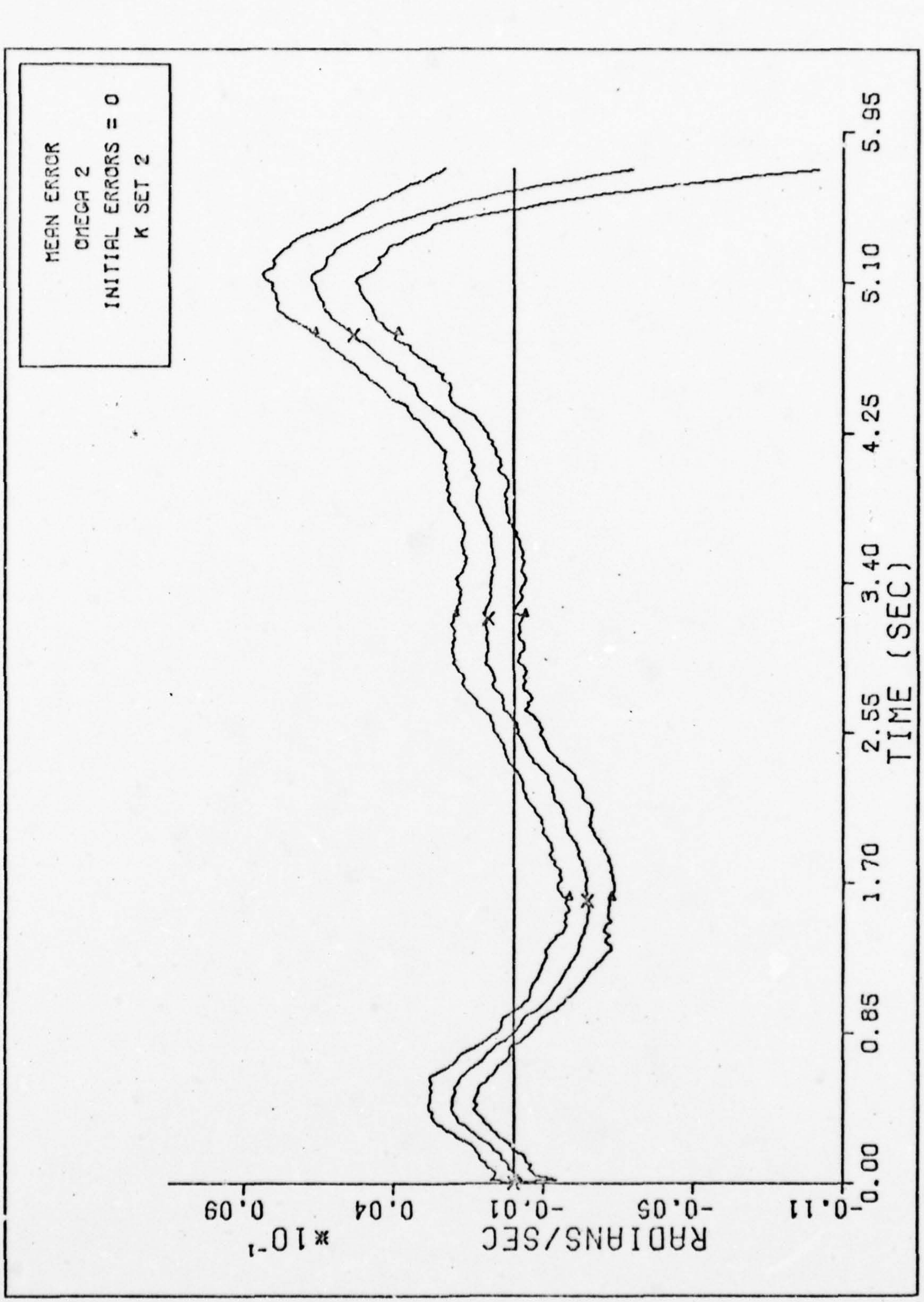



Fig. E-27 OMEGA 2 MEAN ERROR, 11 STATE FILTER

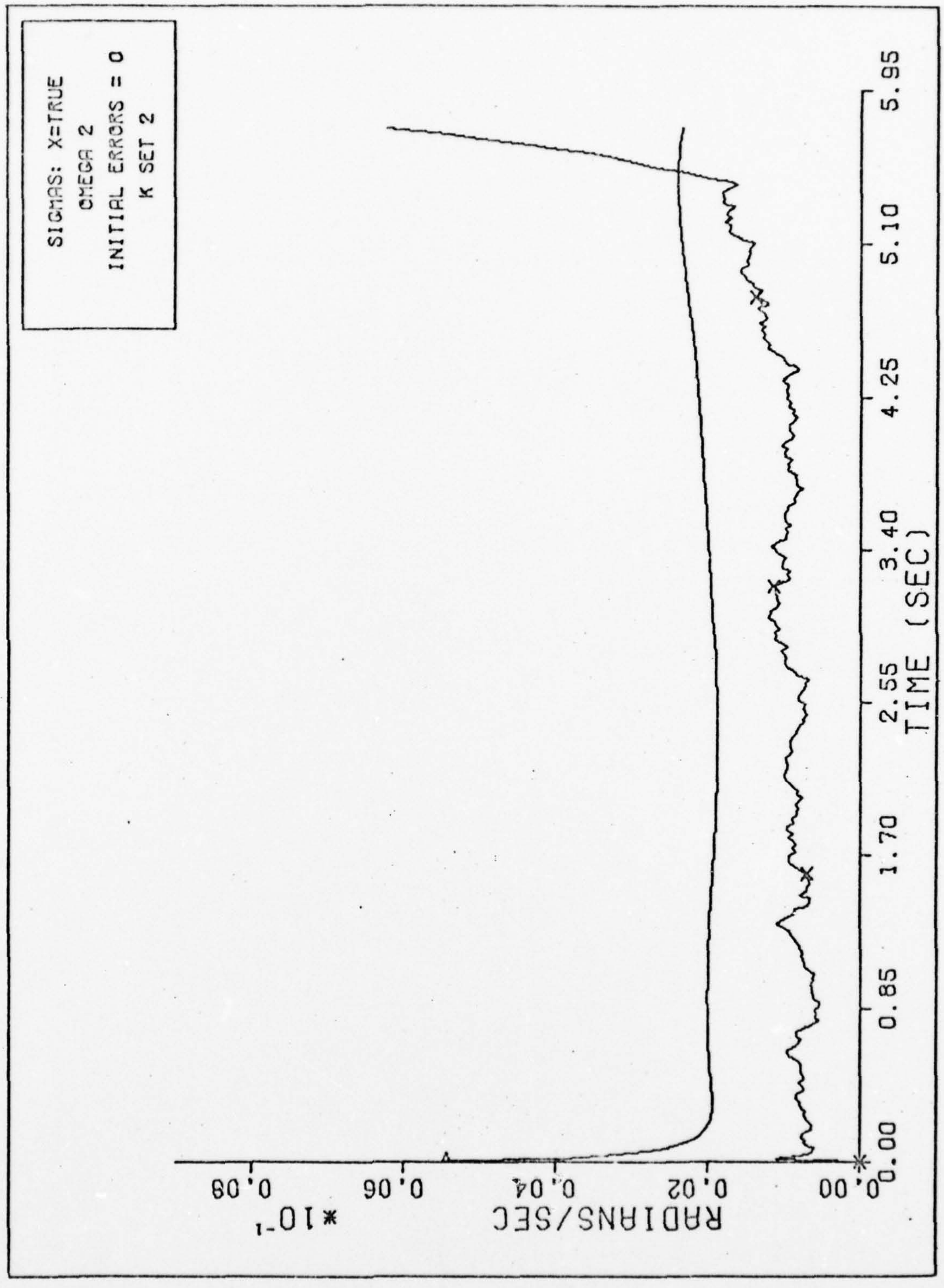



Fig. E-28 OMEGA 2 FILTER & TRUE SIGMAS, 11 STATE FILTER

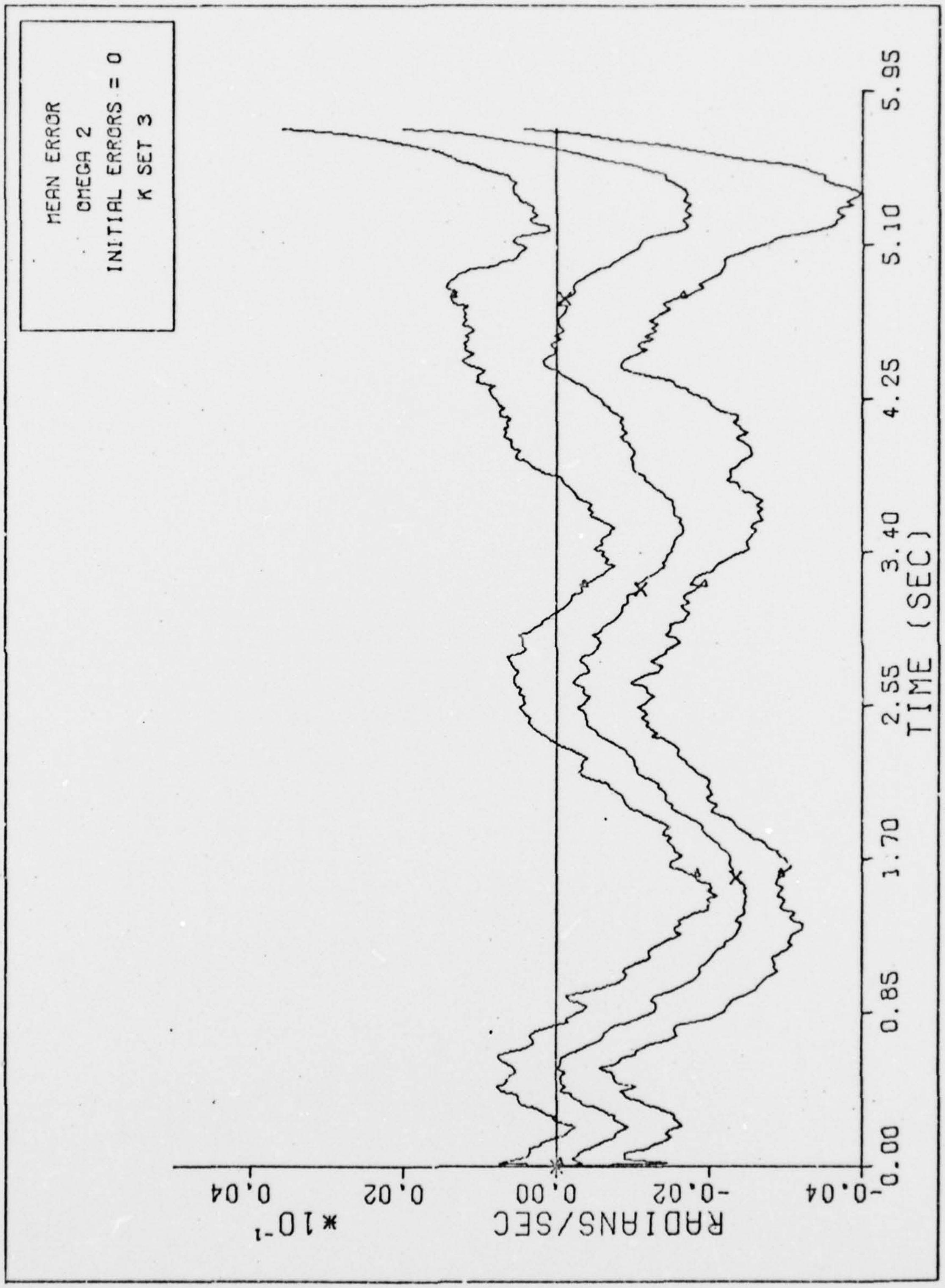



Fig. E-29 OMEGA 2 MEAN ERROR, 11 STATE FILTER

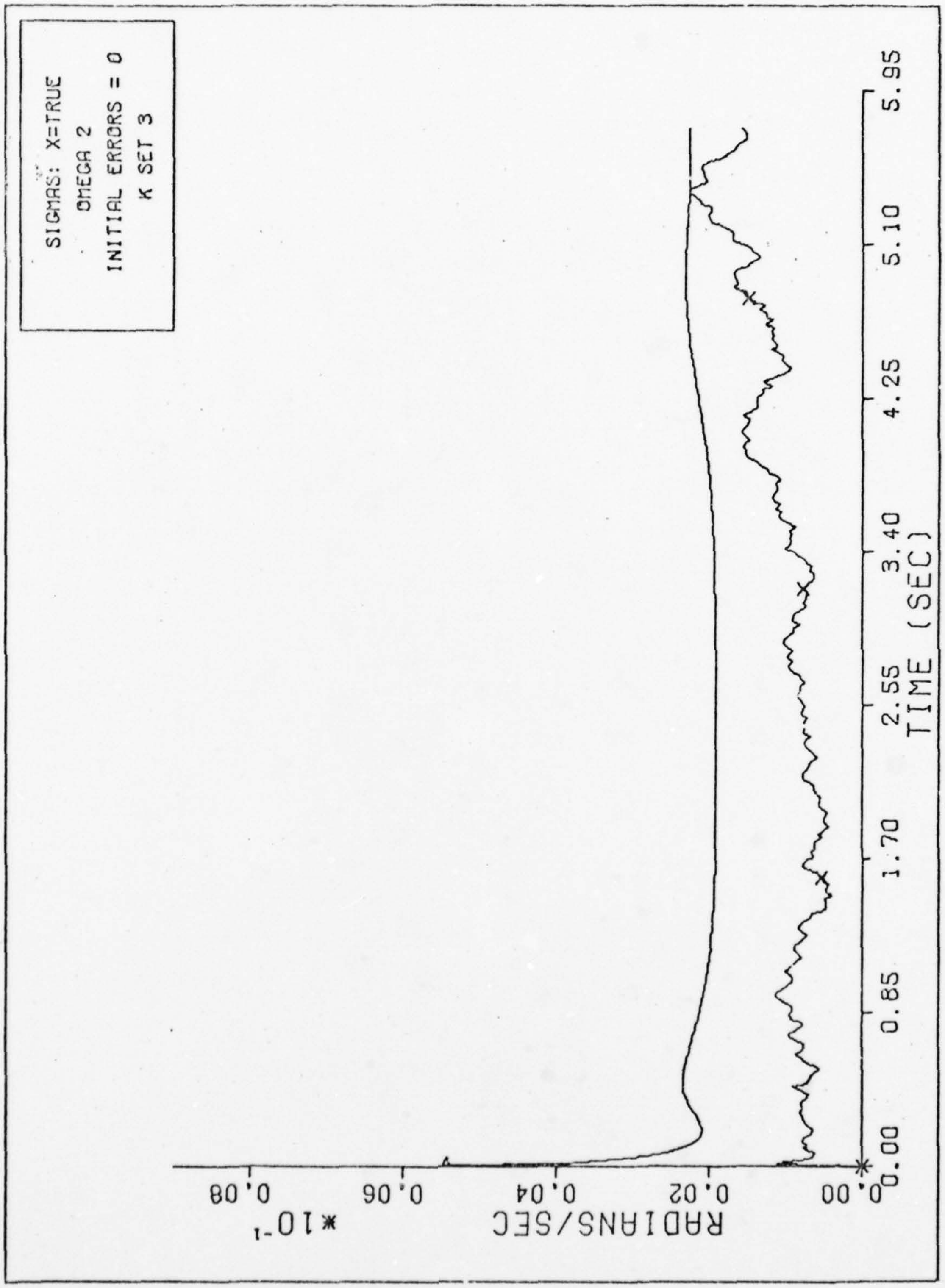



Fig. E-30 OMEGA 2 FILTER & TRUE SIGMAS, 11 STATE FILTER

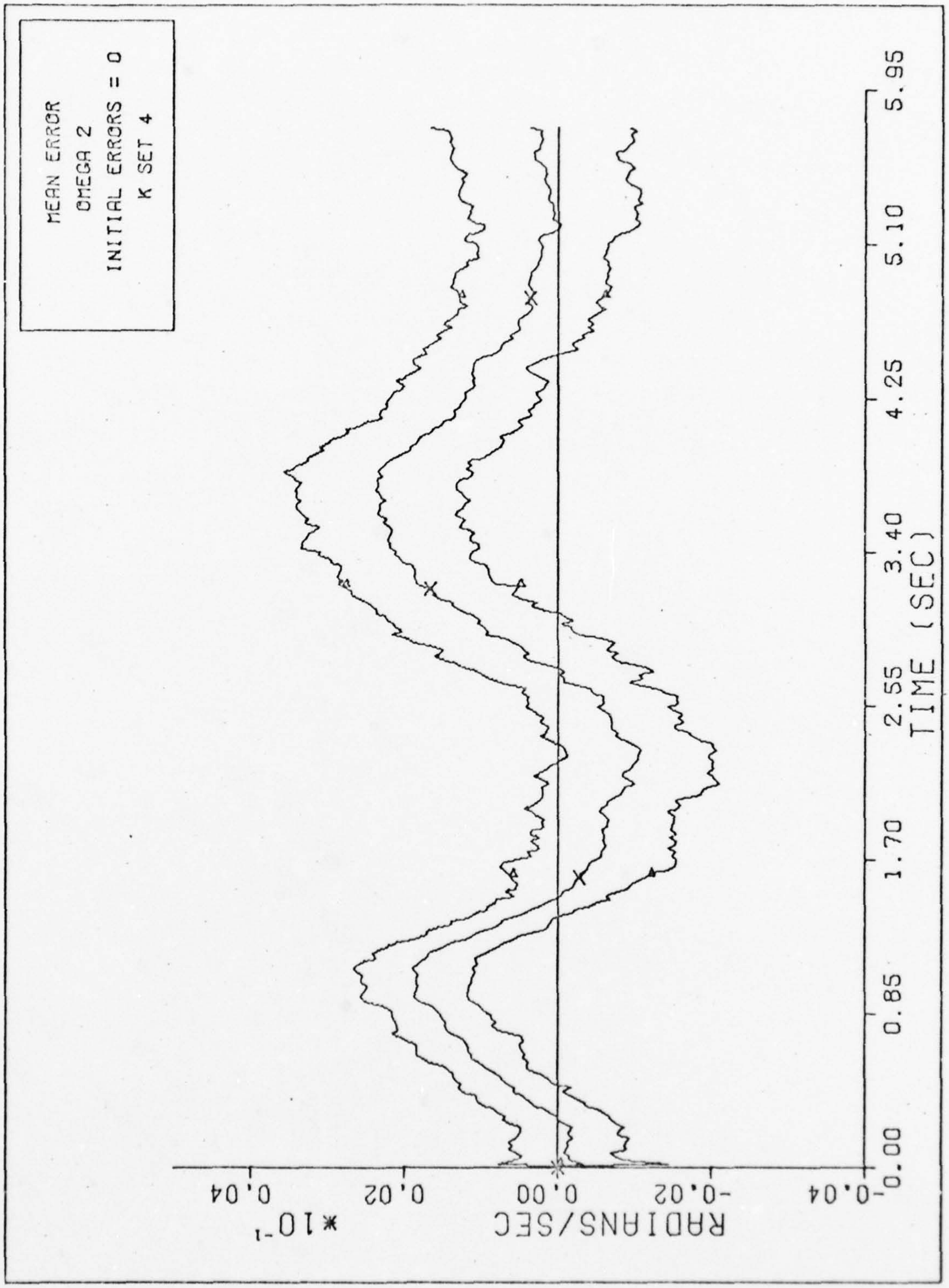



Fig. E-31 OMEGA 2 MEAN ERROR, 11 STATE FILTER

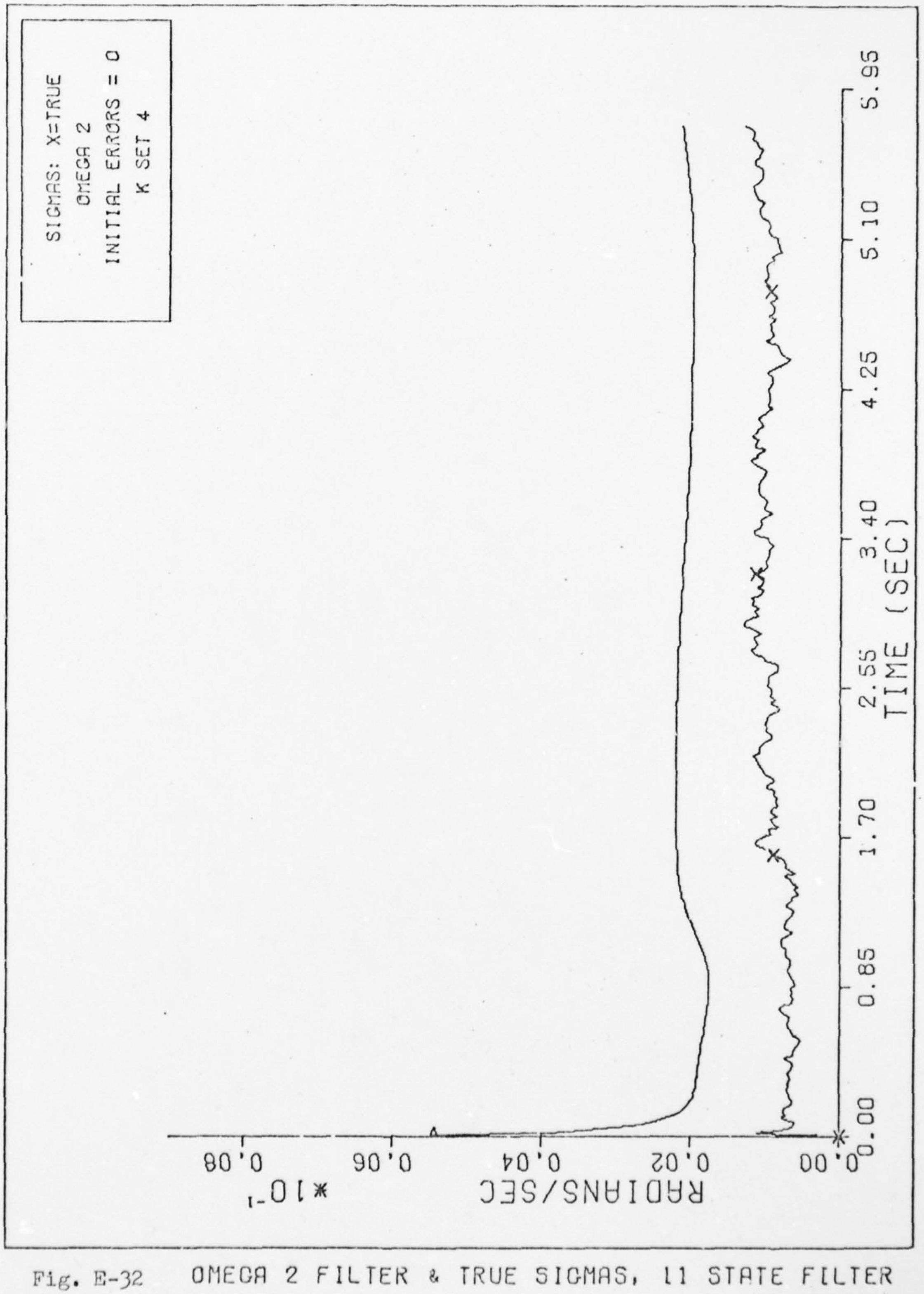



Fig. E-32 OMEGA 2 FILTER & TRUE SIGMAS, 11 STATE FILTER

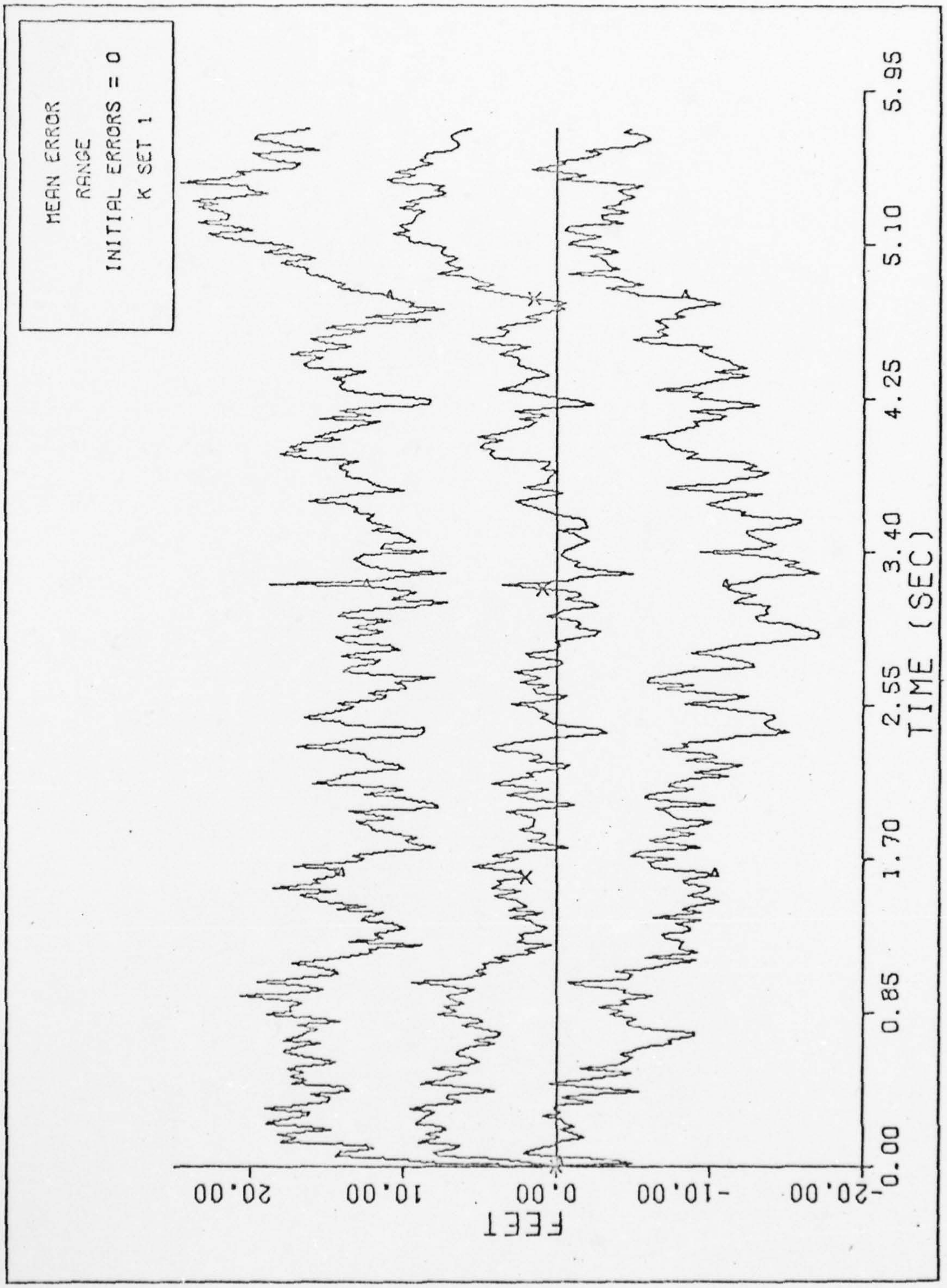



Fig. E-33 RANGE MEAN ERROR, 11 STATE FILTER

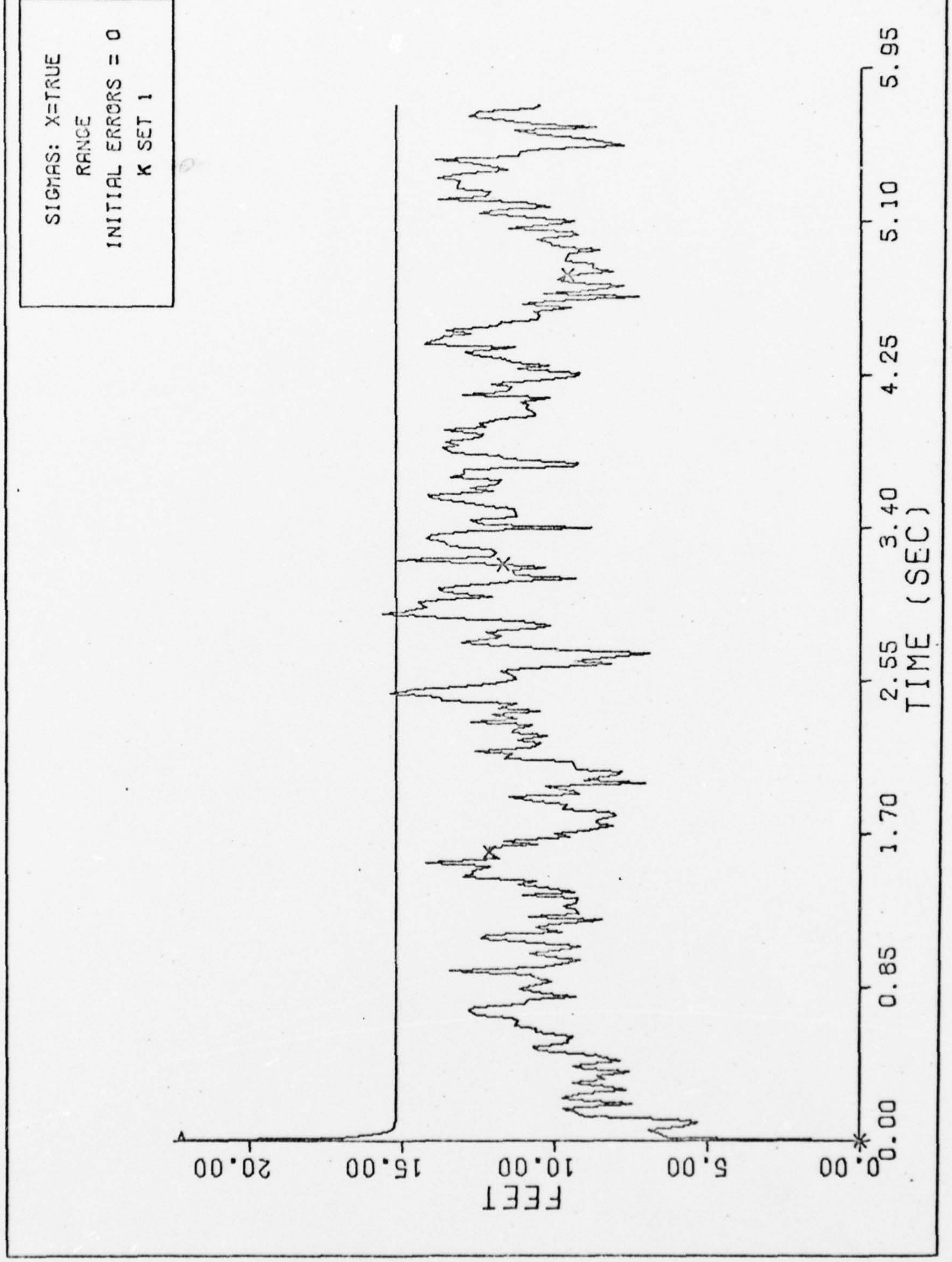



Fig. E-34 RANGE FILTER & TRUE SIGMAS, 11 STATE FILTER

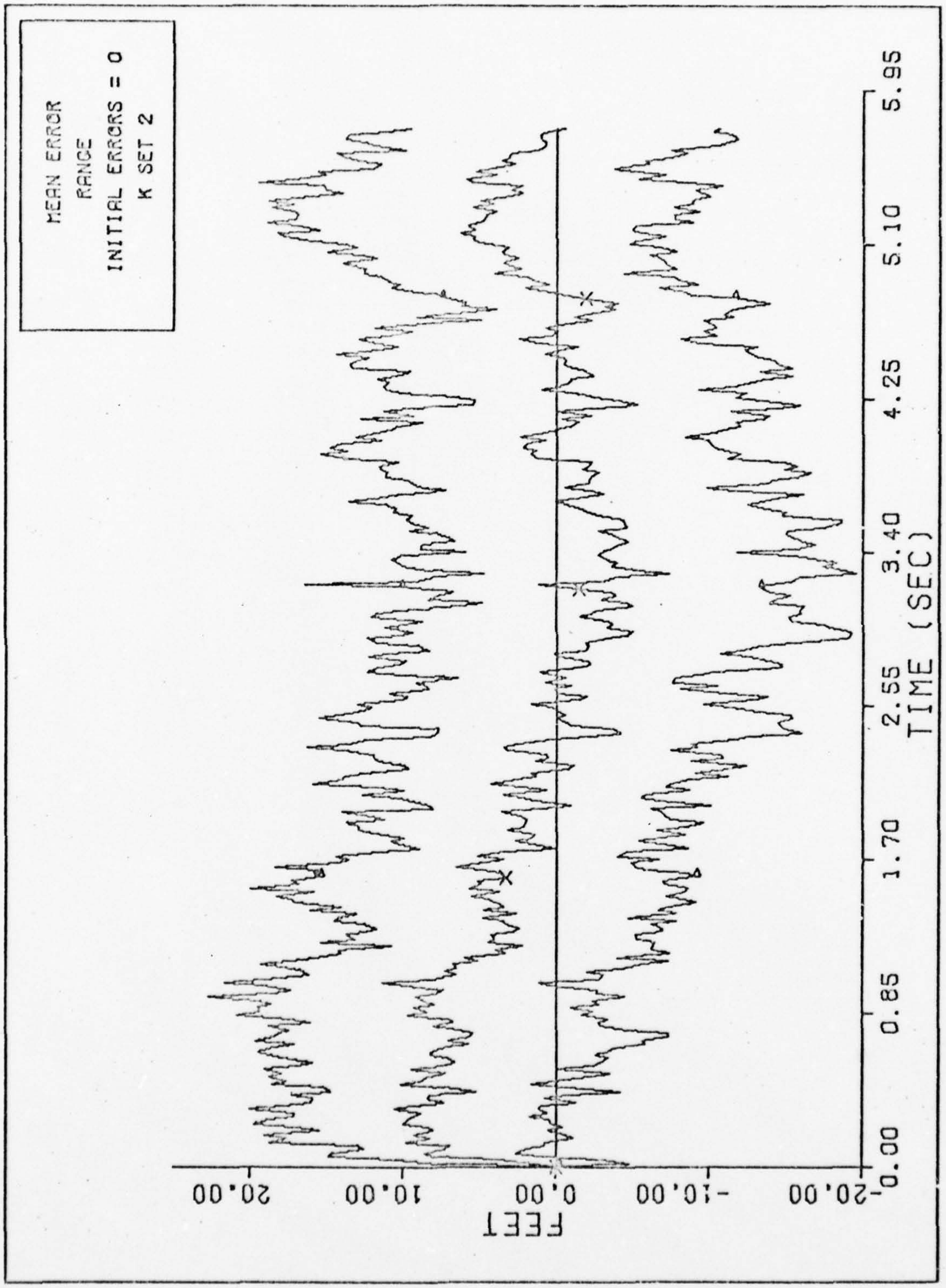



Fig. E-35 RANGE MEAN ERROR, 11 STATE FILTER

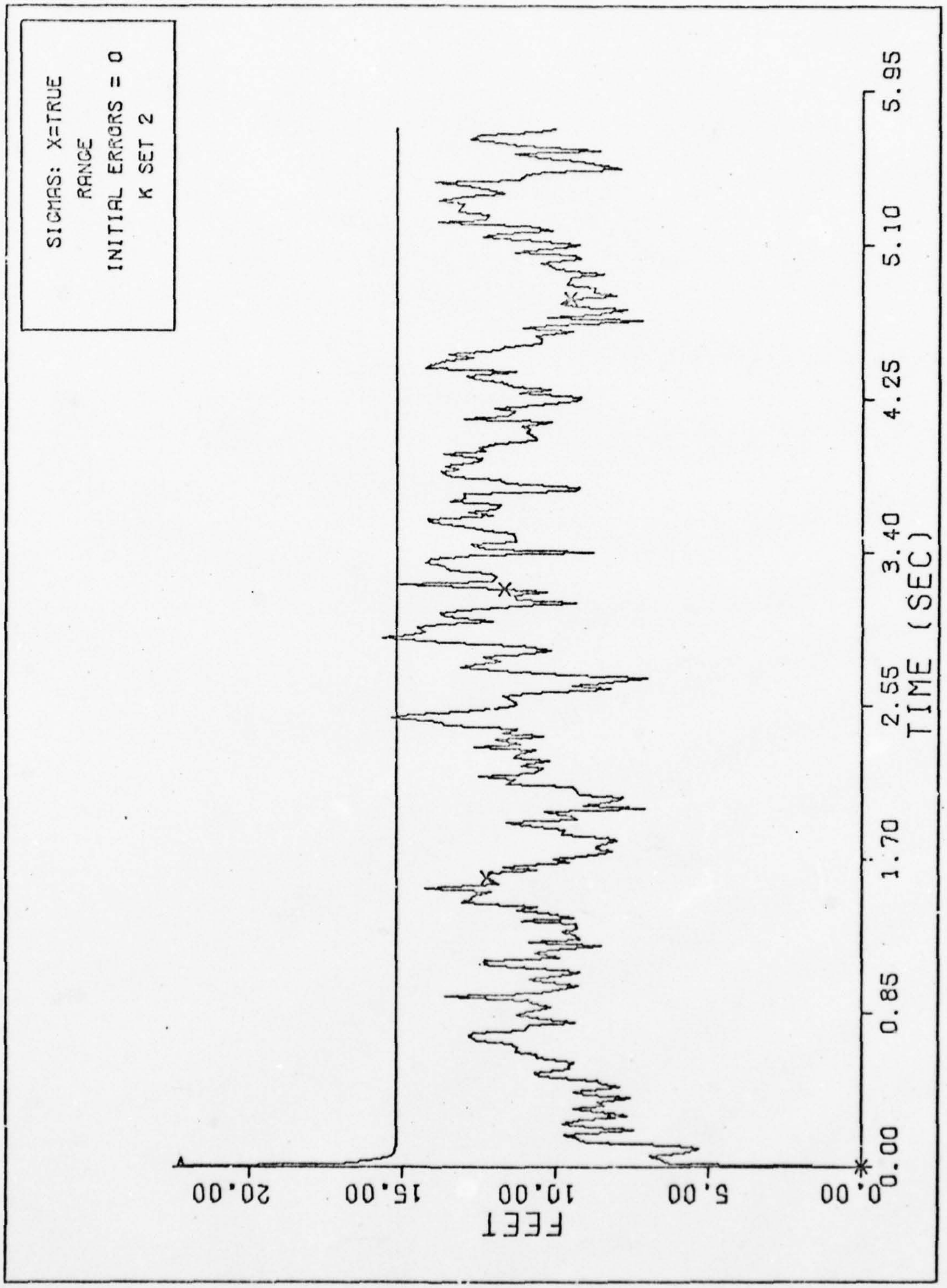



Fig. E-36 RANGE FILTER & TRUE SIGMAS, 11 STATE FILTER

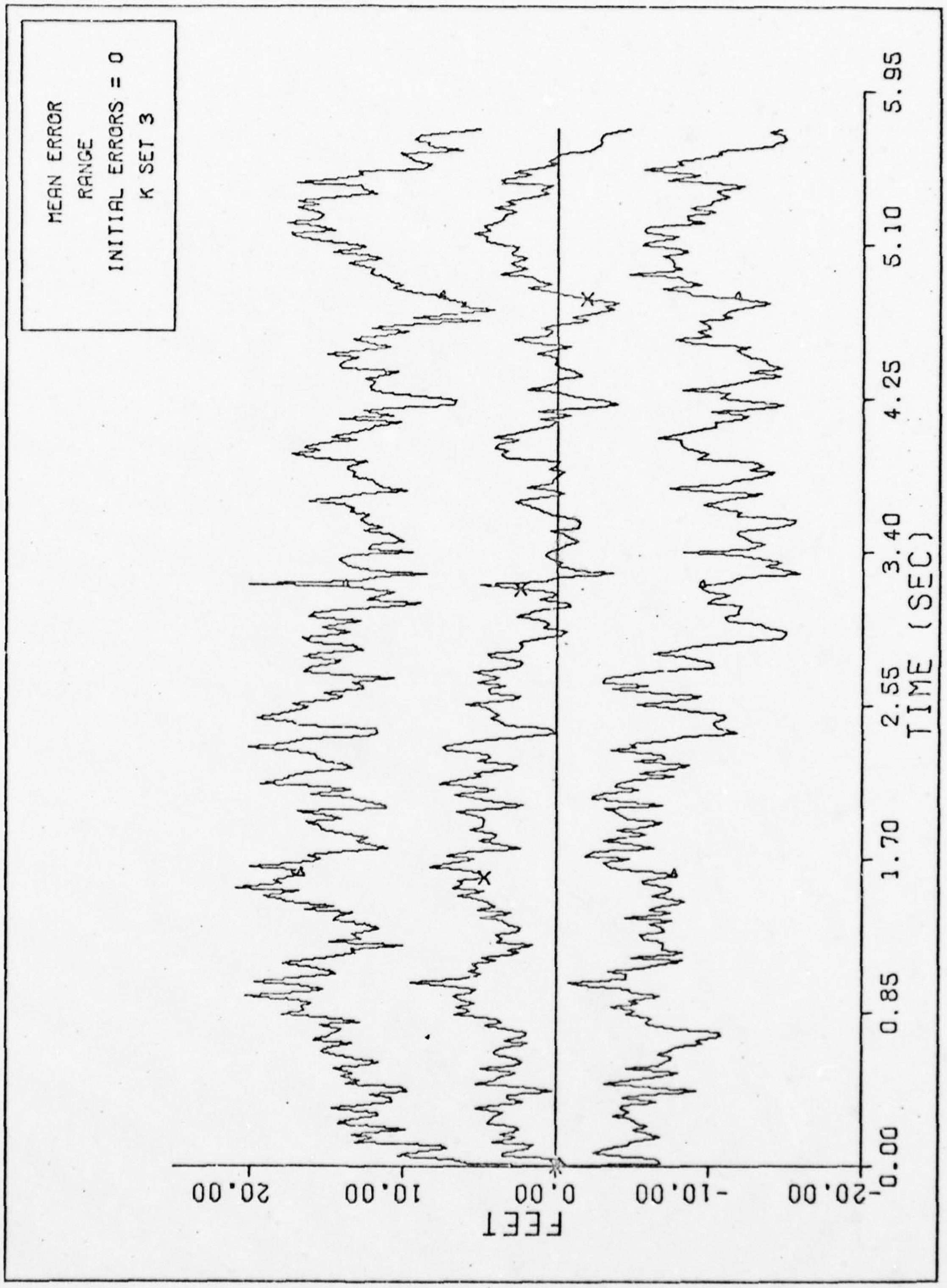



Fig. E-37 RANGE MEAN ERROR, 11 STATE FILTER

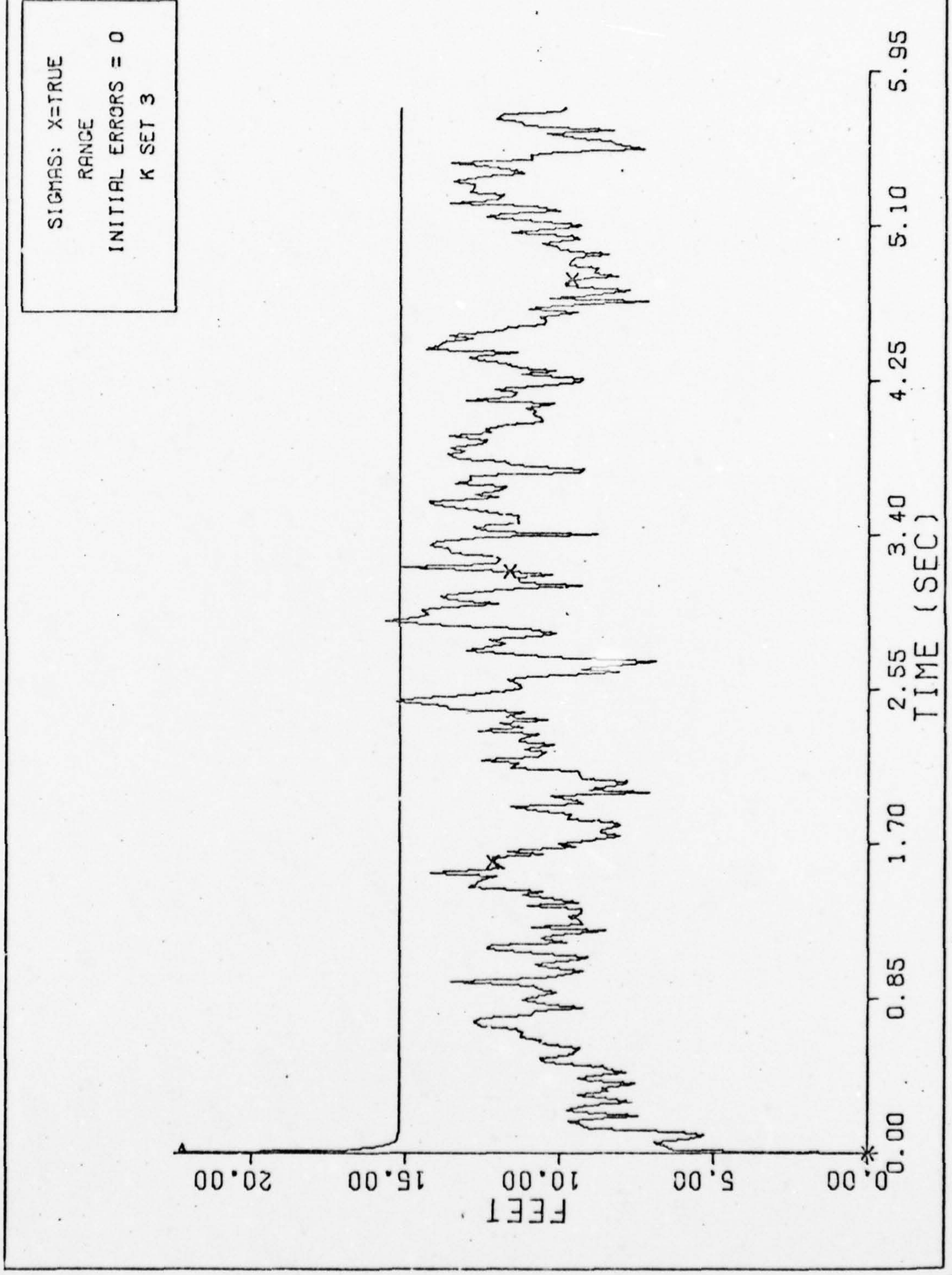



Fig. E-38 RANGE FILTER & TRUE SIGMAS, 11 STATE FILTER

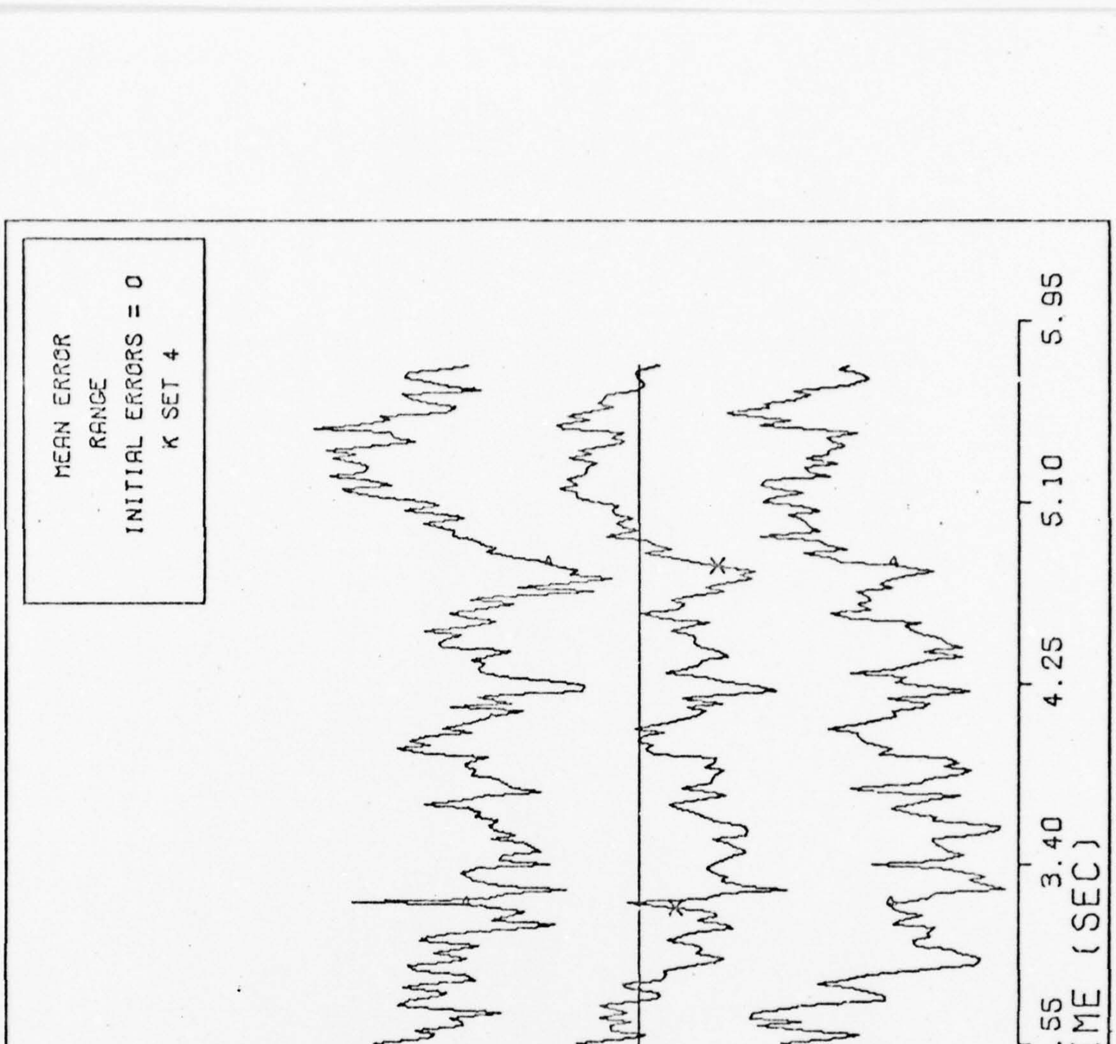



Fig. E-39 RANGE MEAN ERROR, 11 STATE FILTER

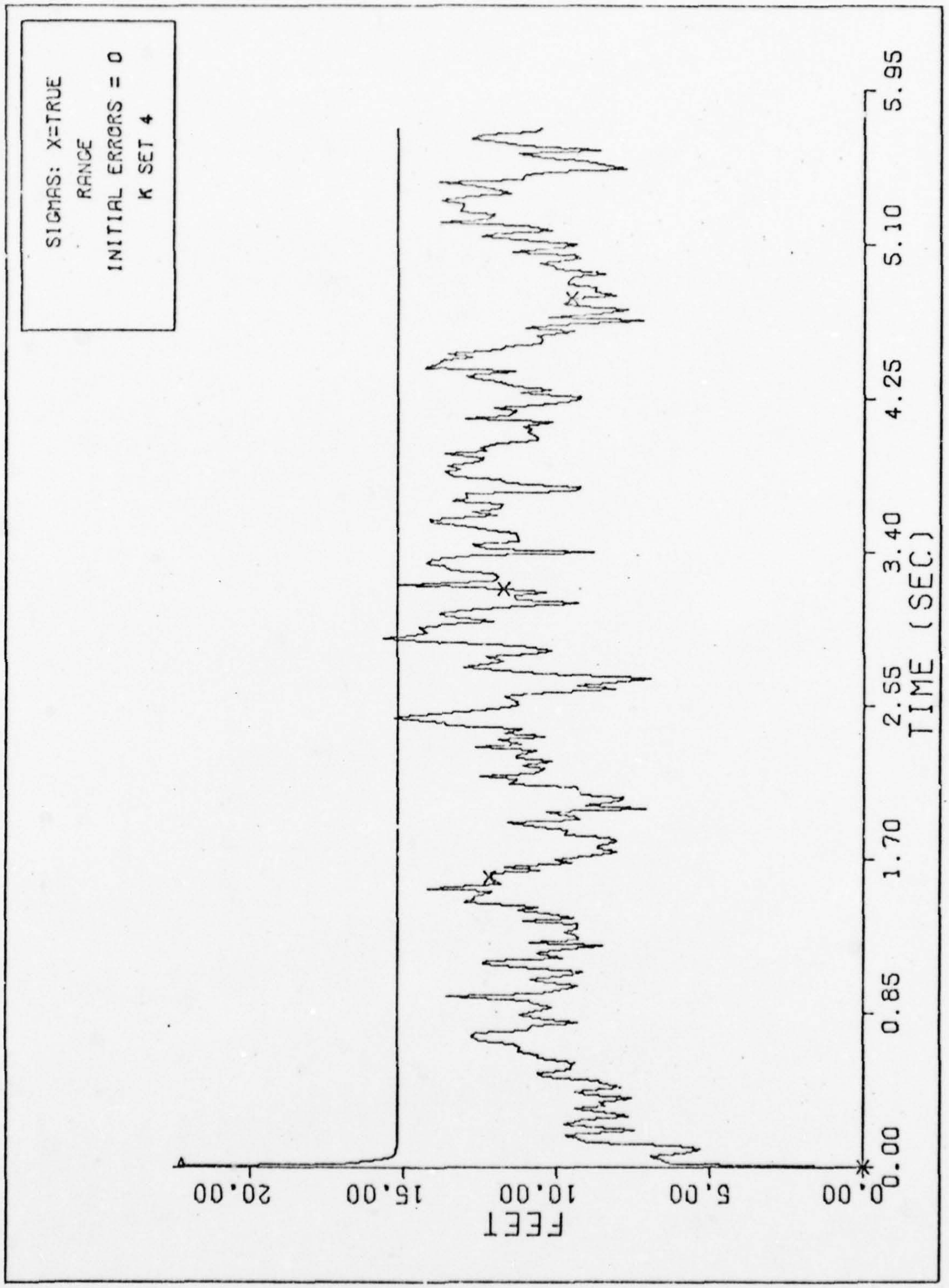



Fig. E-40 RANGE FILTER & TRUE SIGMAS, 11 STATE FILTER

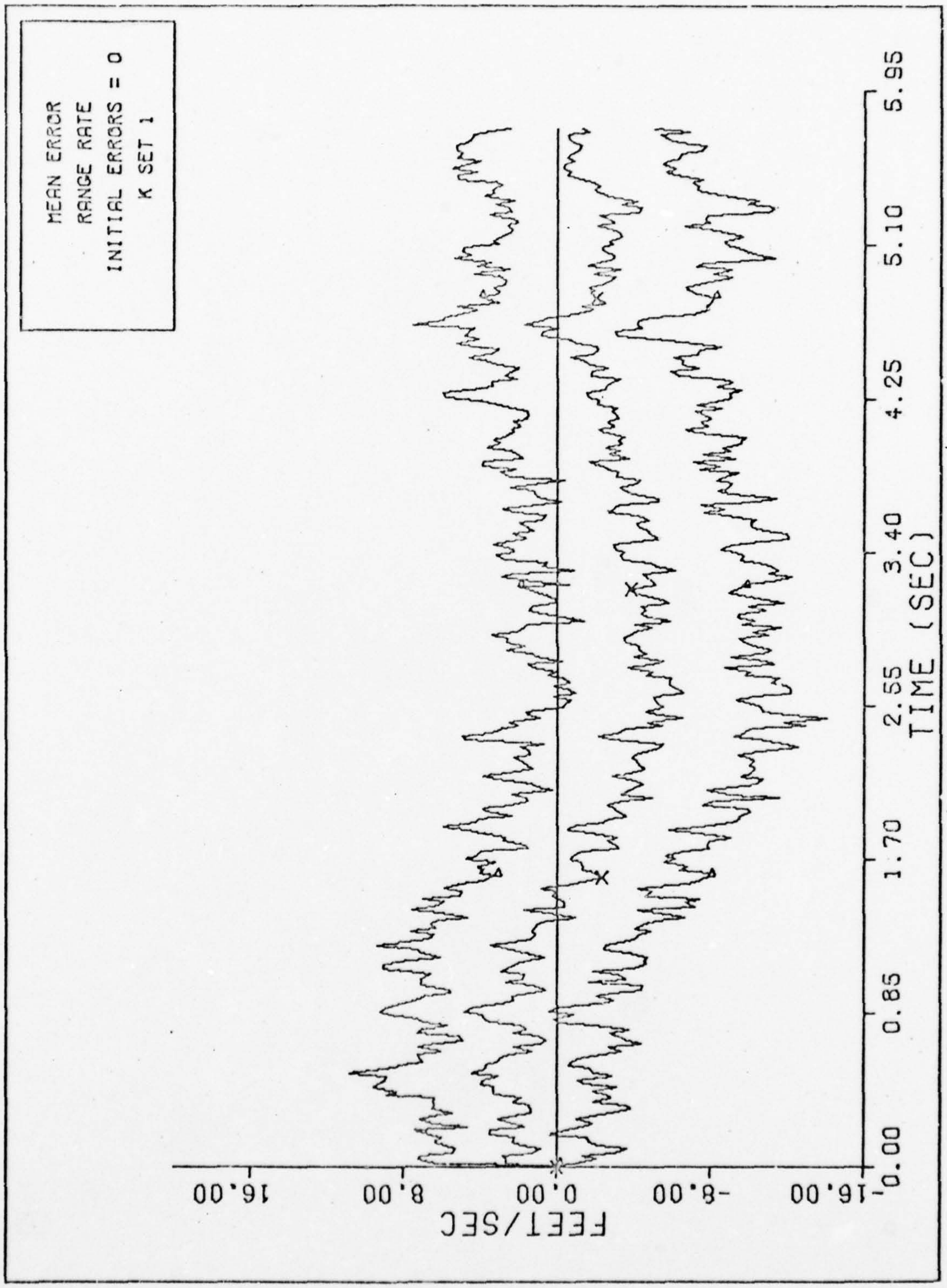



Fig. E-41 RANGE RATE MEAN ERROR, 11 STATE FILTER

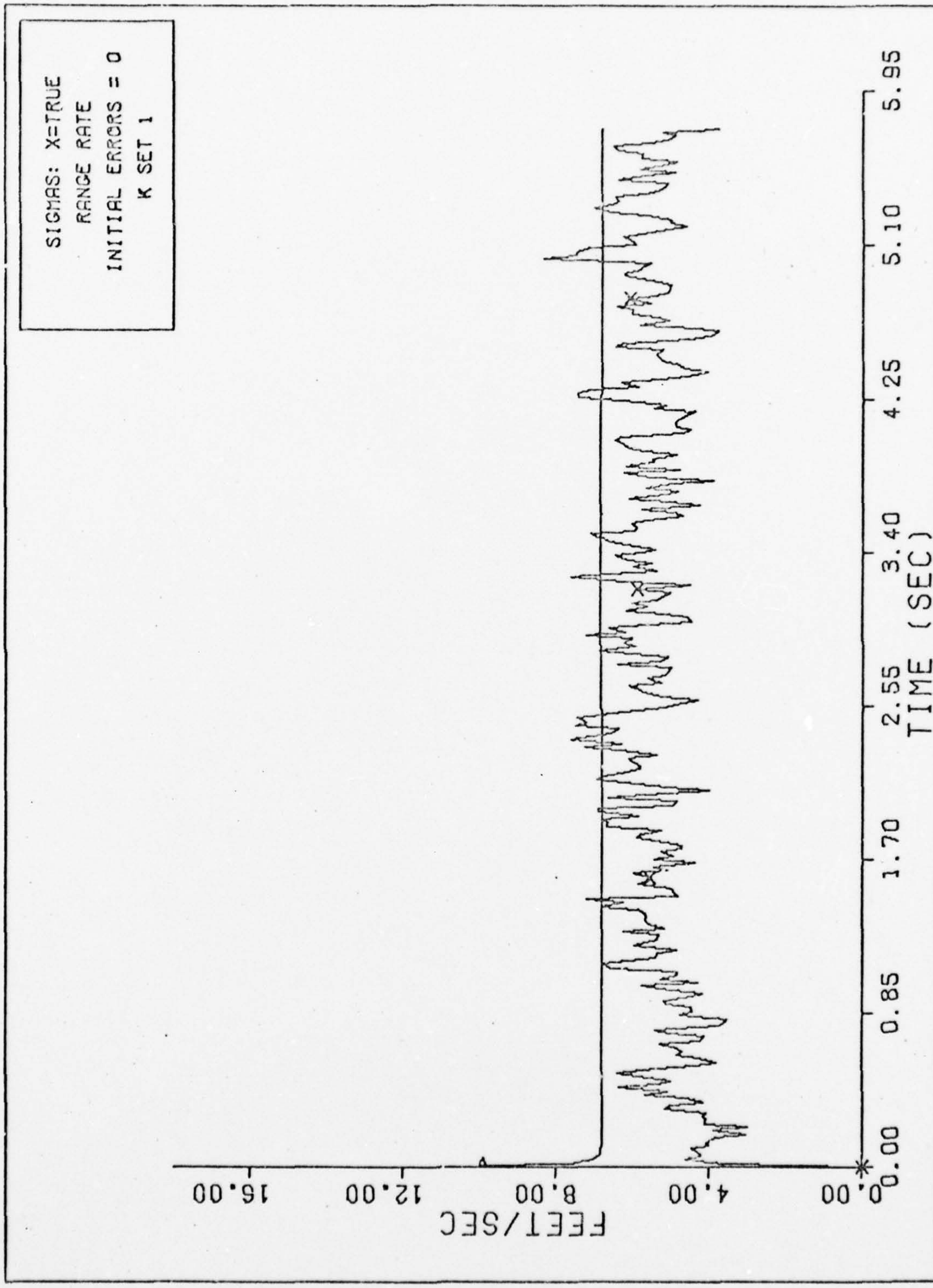



Fig. E-42 RANGE RATE FILTER & TRUE SIGMAS, 11 STATE FILTER

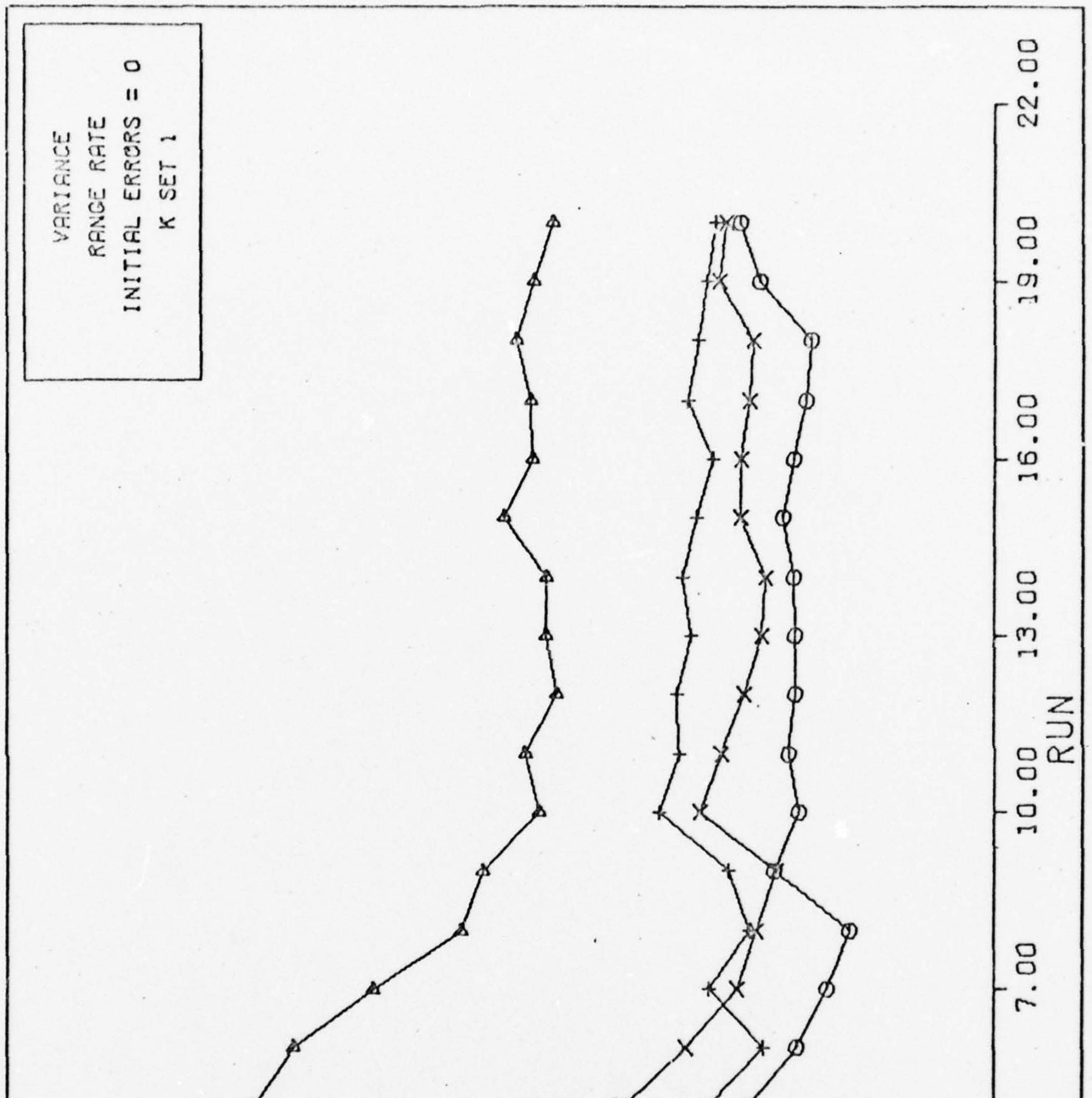



Fig. E-43 VARIANCE CONVERGENCE

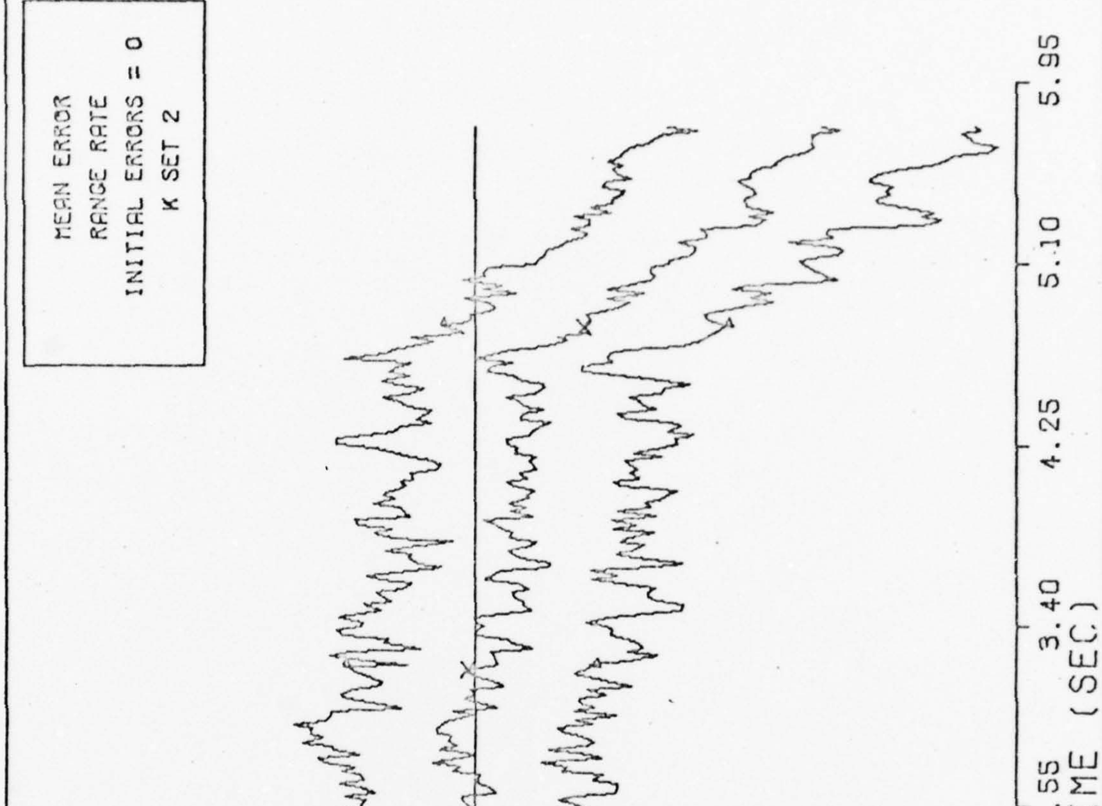



Fig. E-44 RANGE RATE MEAN ERROR, 11 STATE FILTER

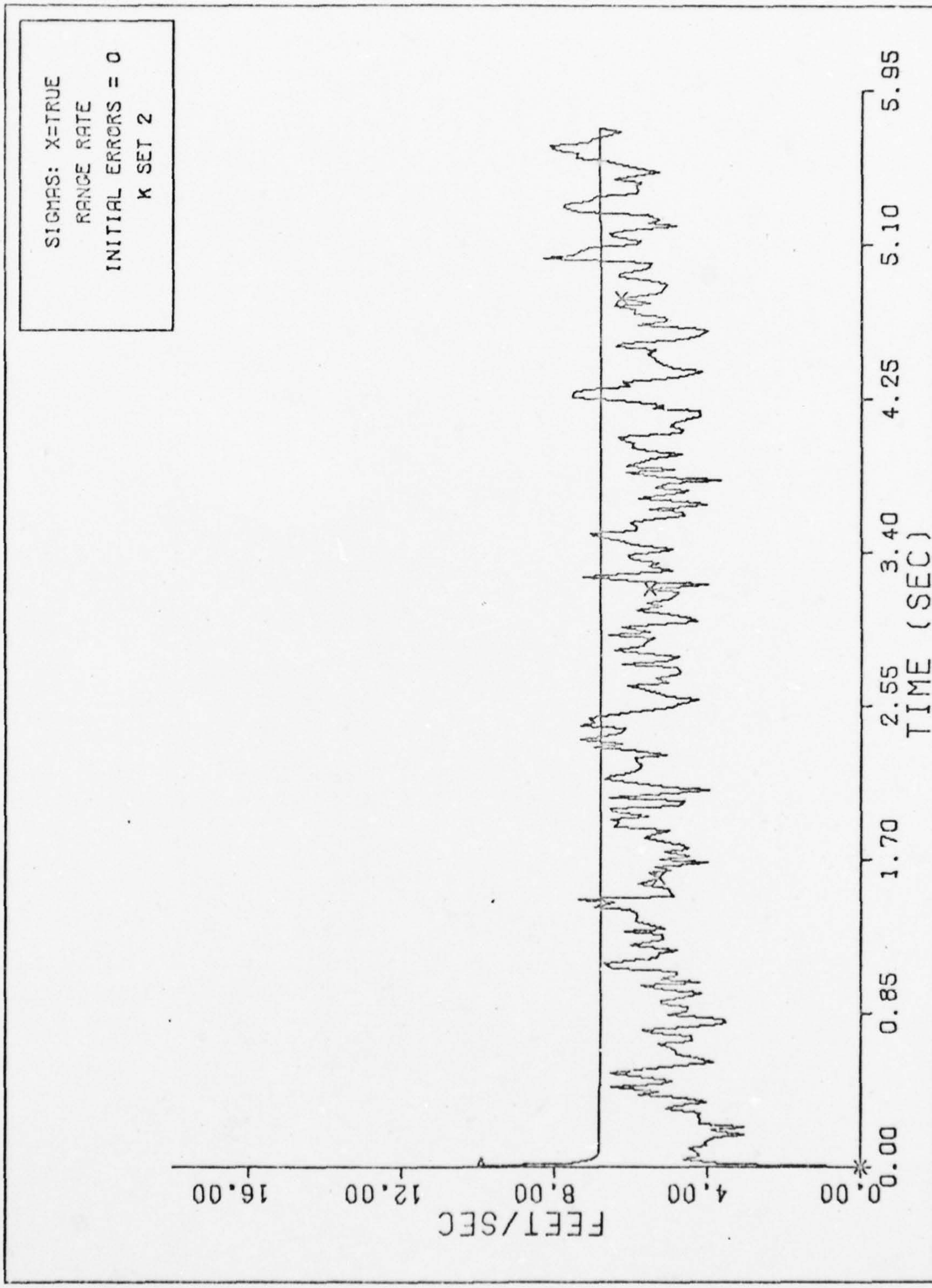



Fig. E-45 RANGE RATE FILTER & TRUE SIGMAS, 11 STATE FILTER

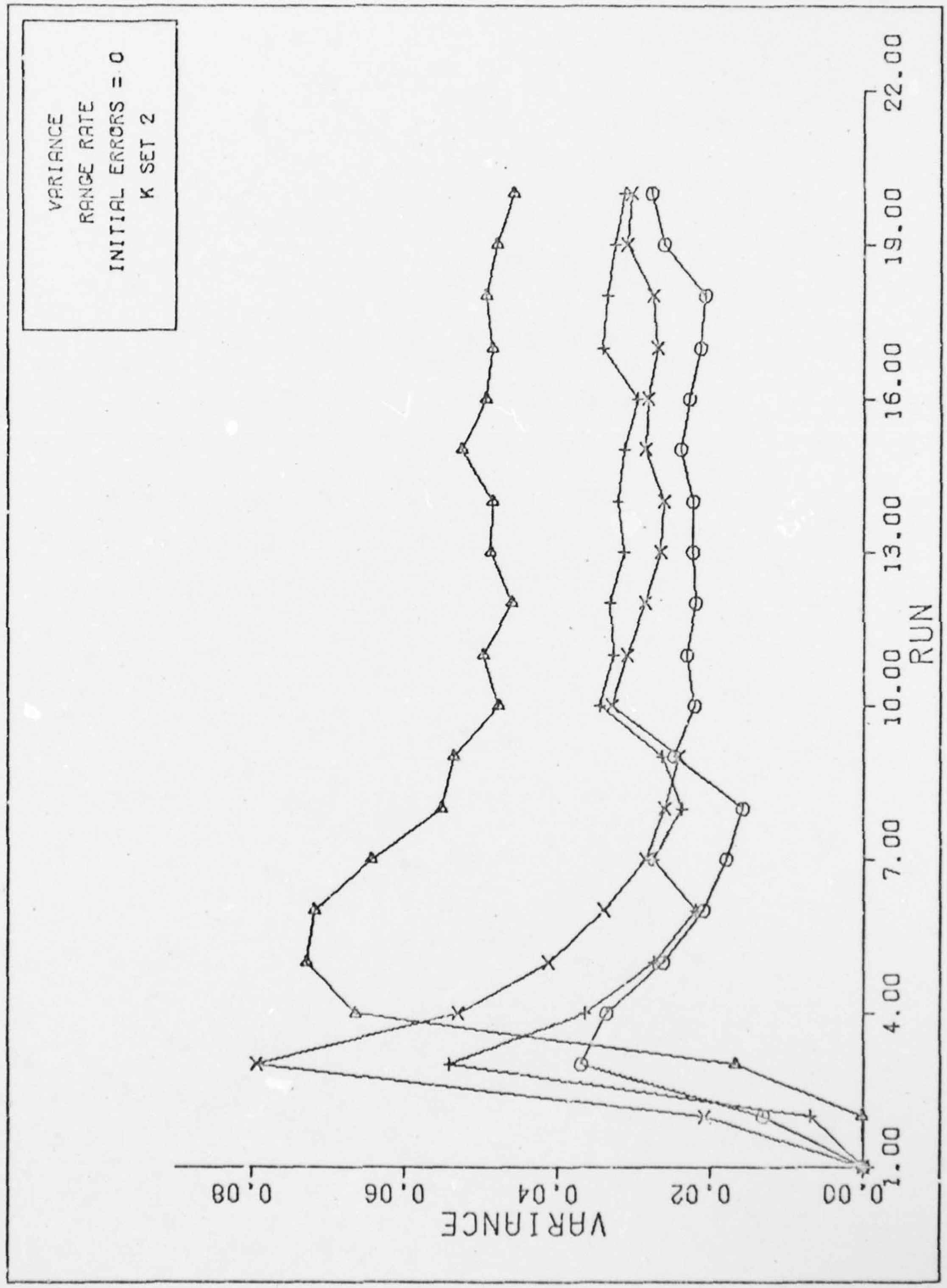



Fig. E-46

VARIANCE CONVERGENCE

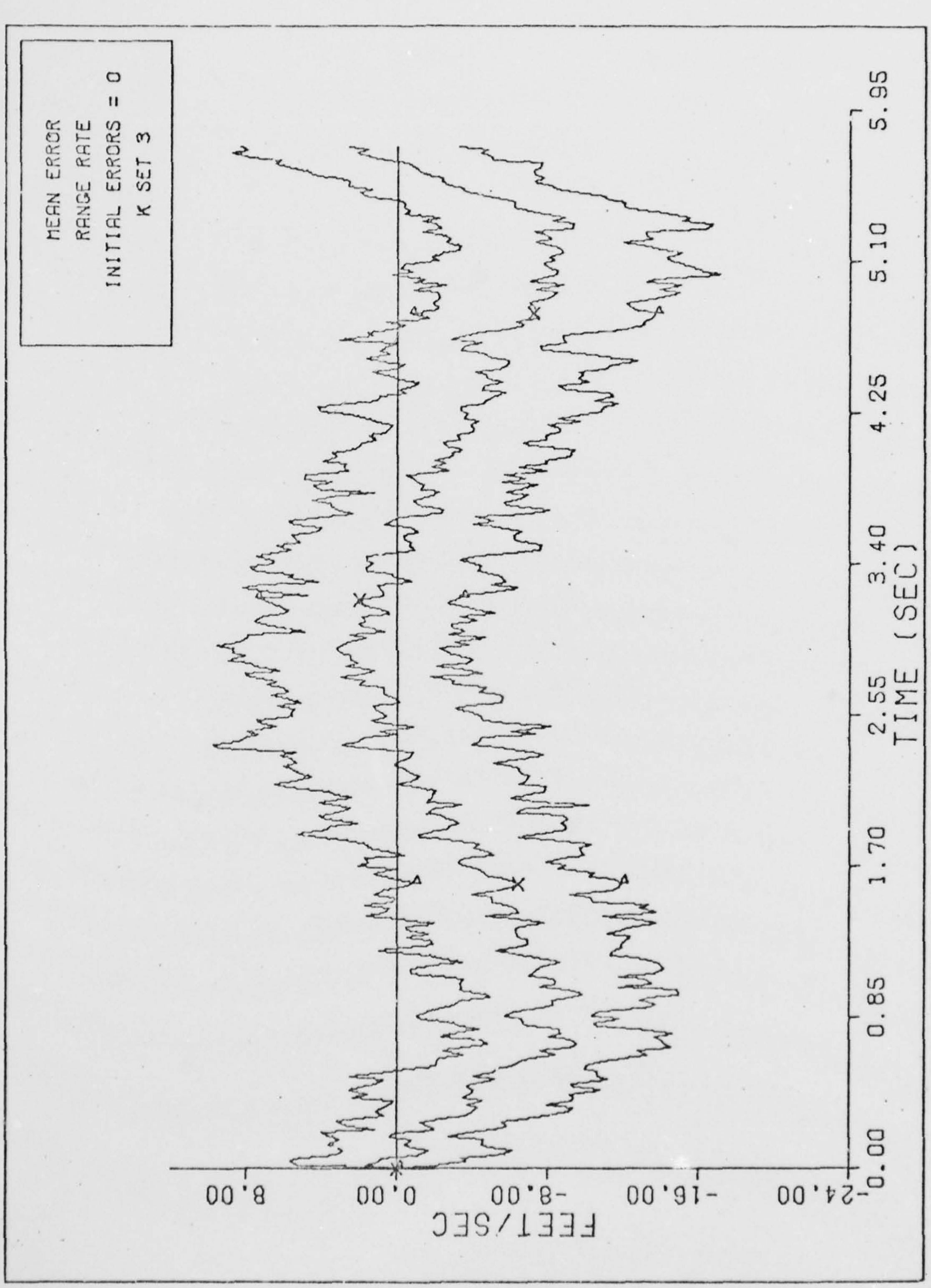



Fig. E-47 RANGE RATE MEAN ERROR, 11 STATE FILTER

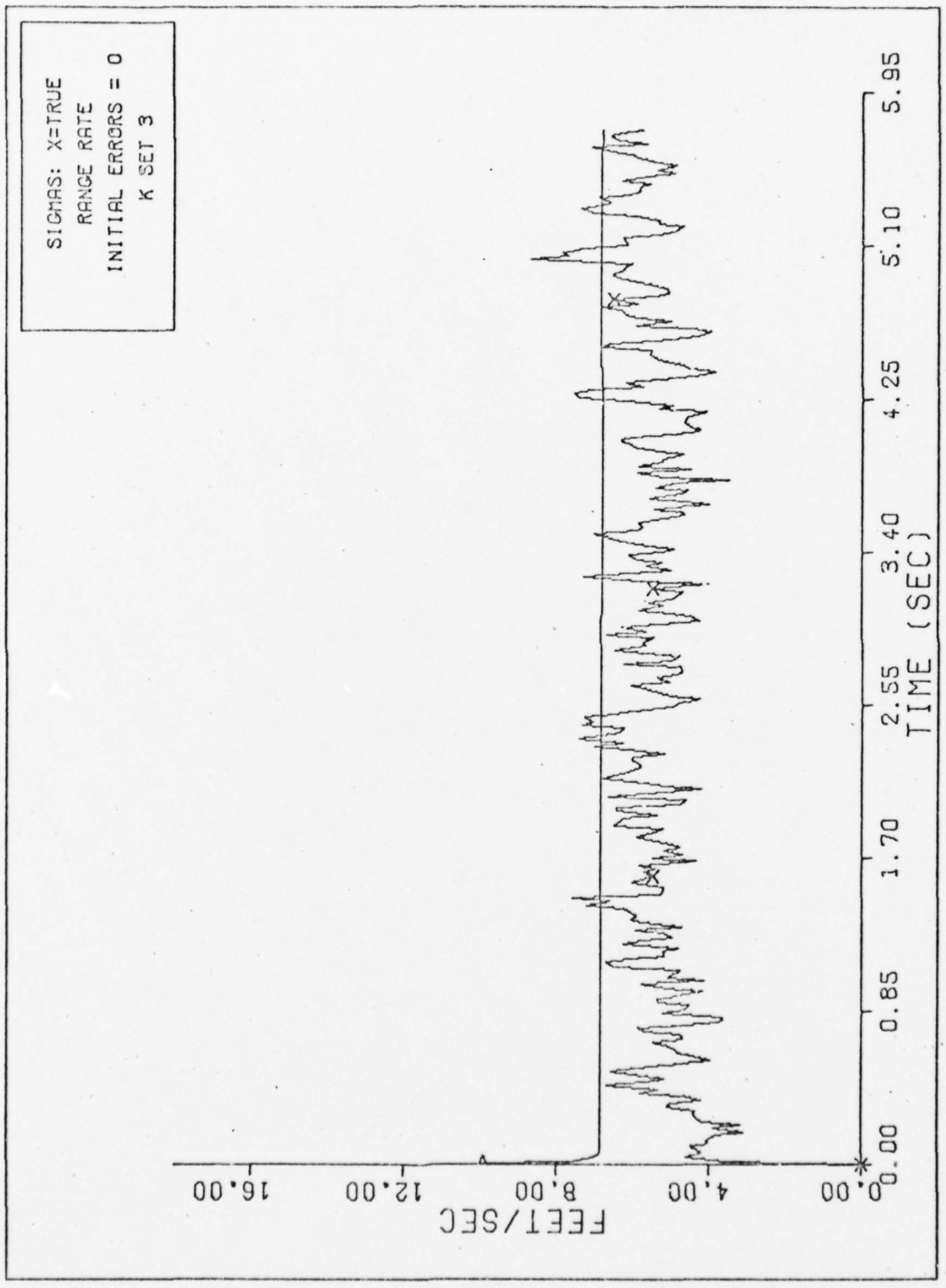



Fig. E-48 RANGE RATE FILTER & TRUE SIGMAS, 11 STATE FILTER

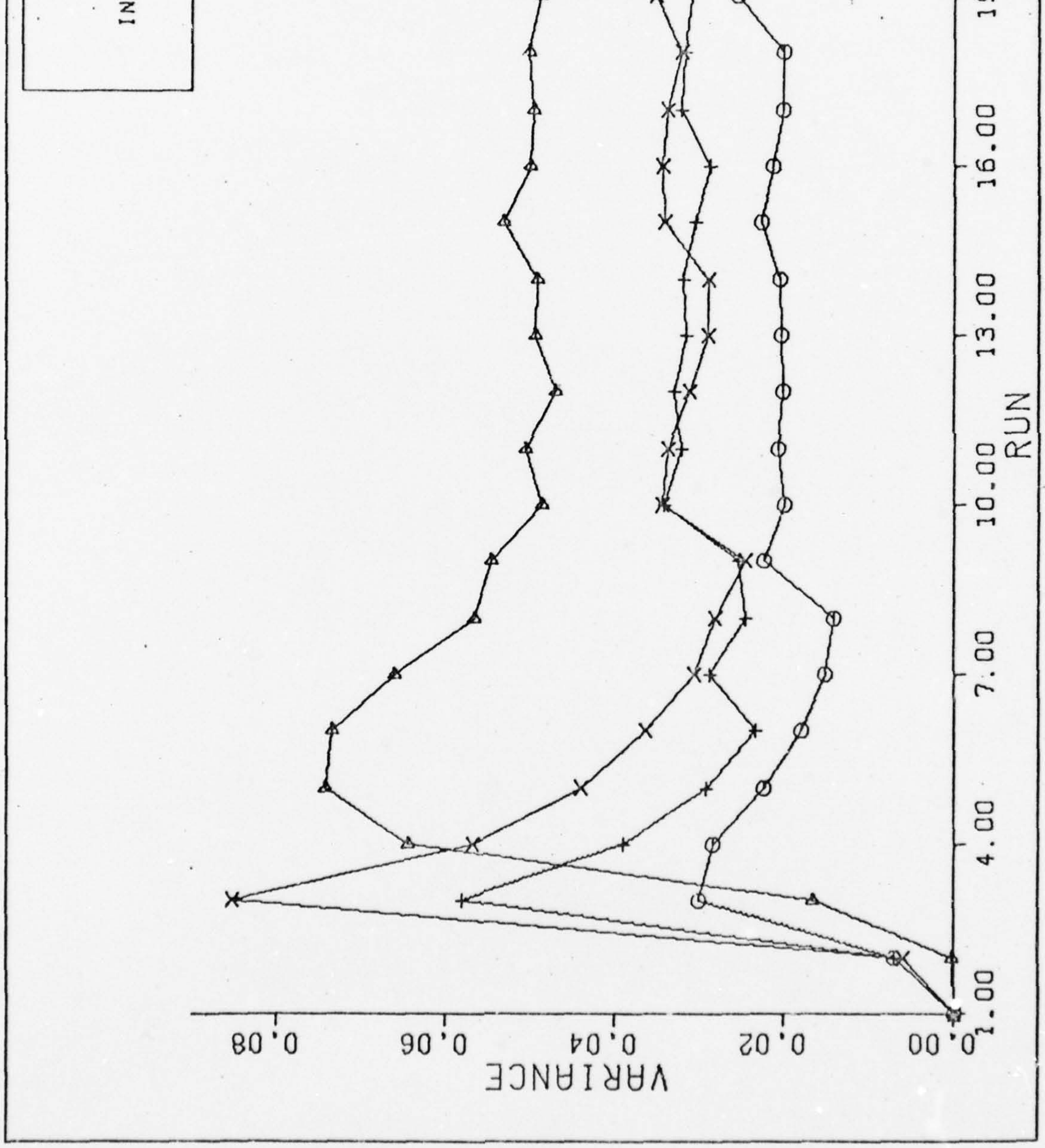



Fig. E-49 VARIANCE CONVERGENCE

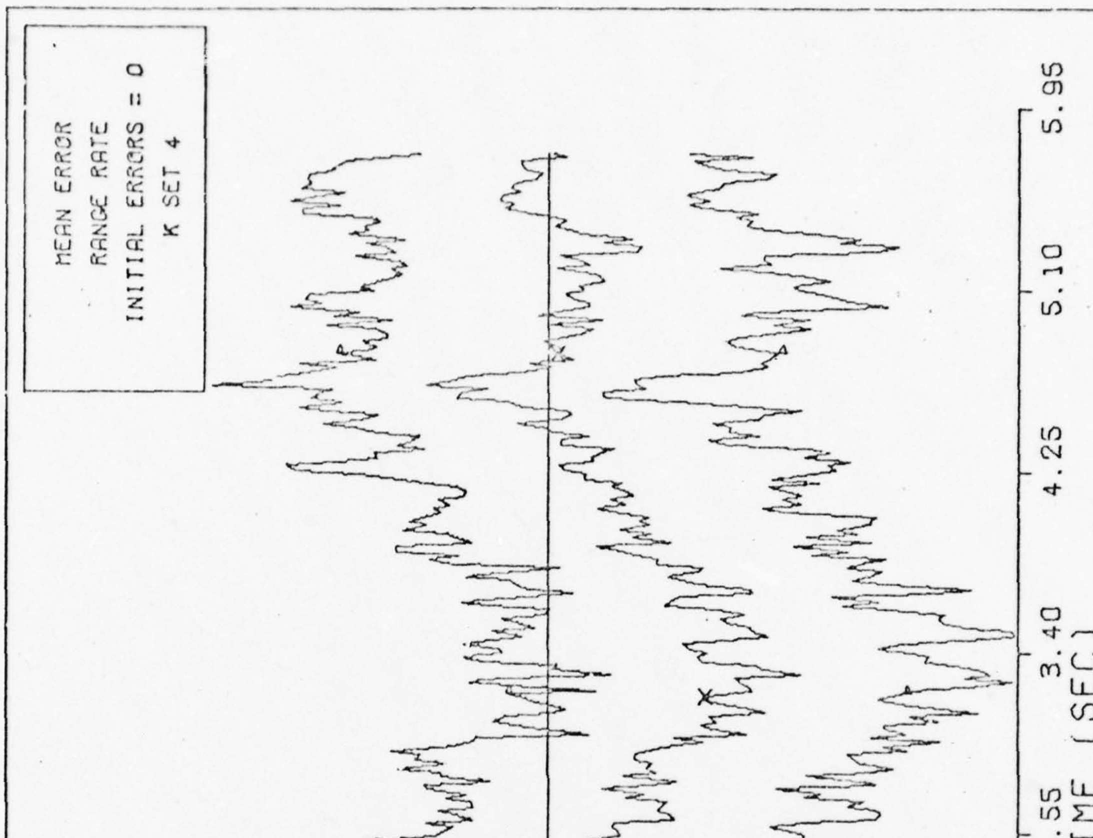



Fig. E-50 RANGE RATE MEAN ERROR, 11 STATE FILTER

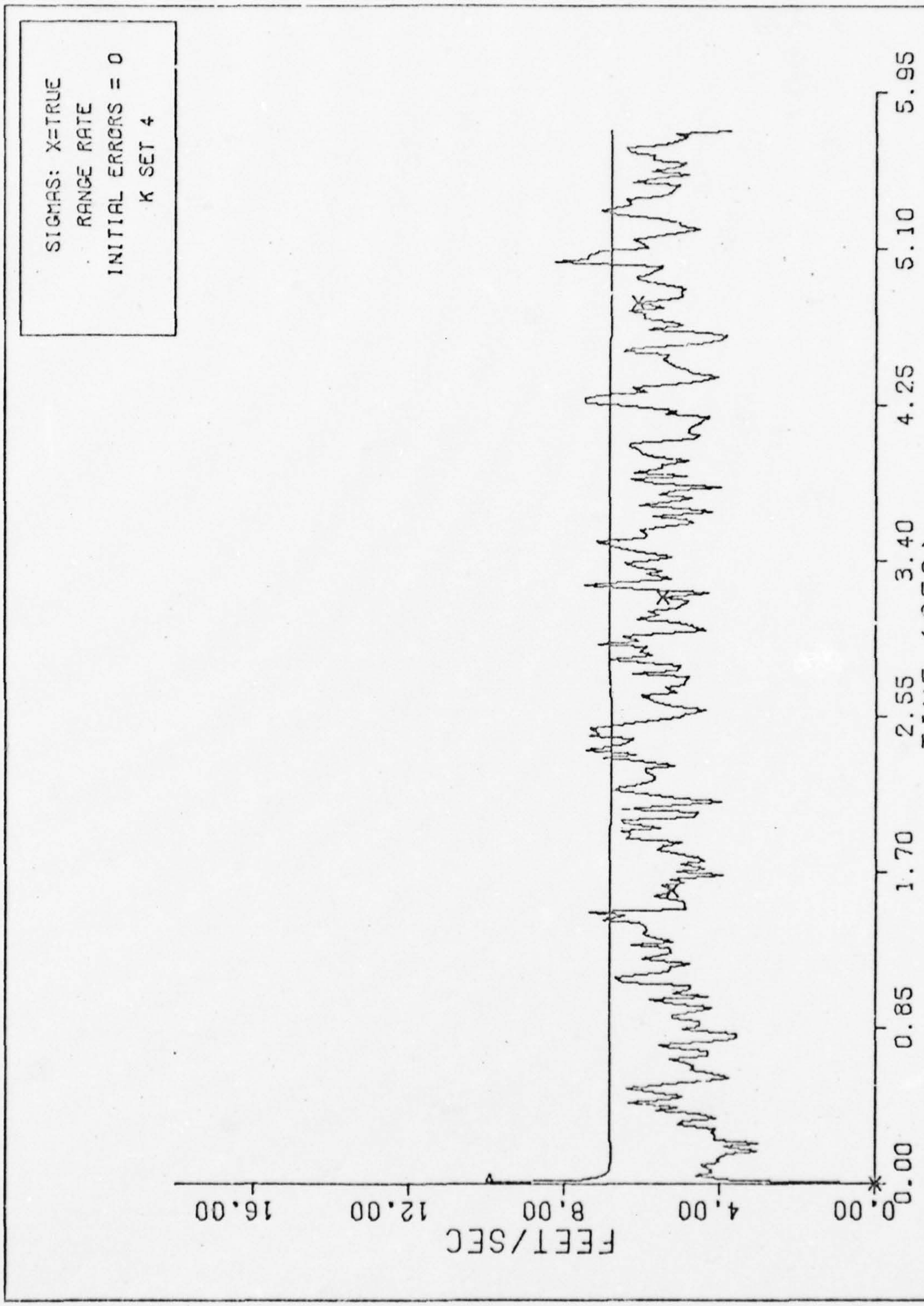



Fig. E-51 RANGE RATE FILTER & TRUE SIGMAS, 11 STATE FILTER

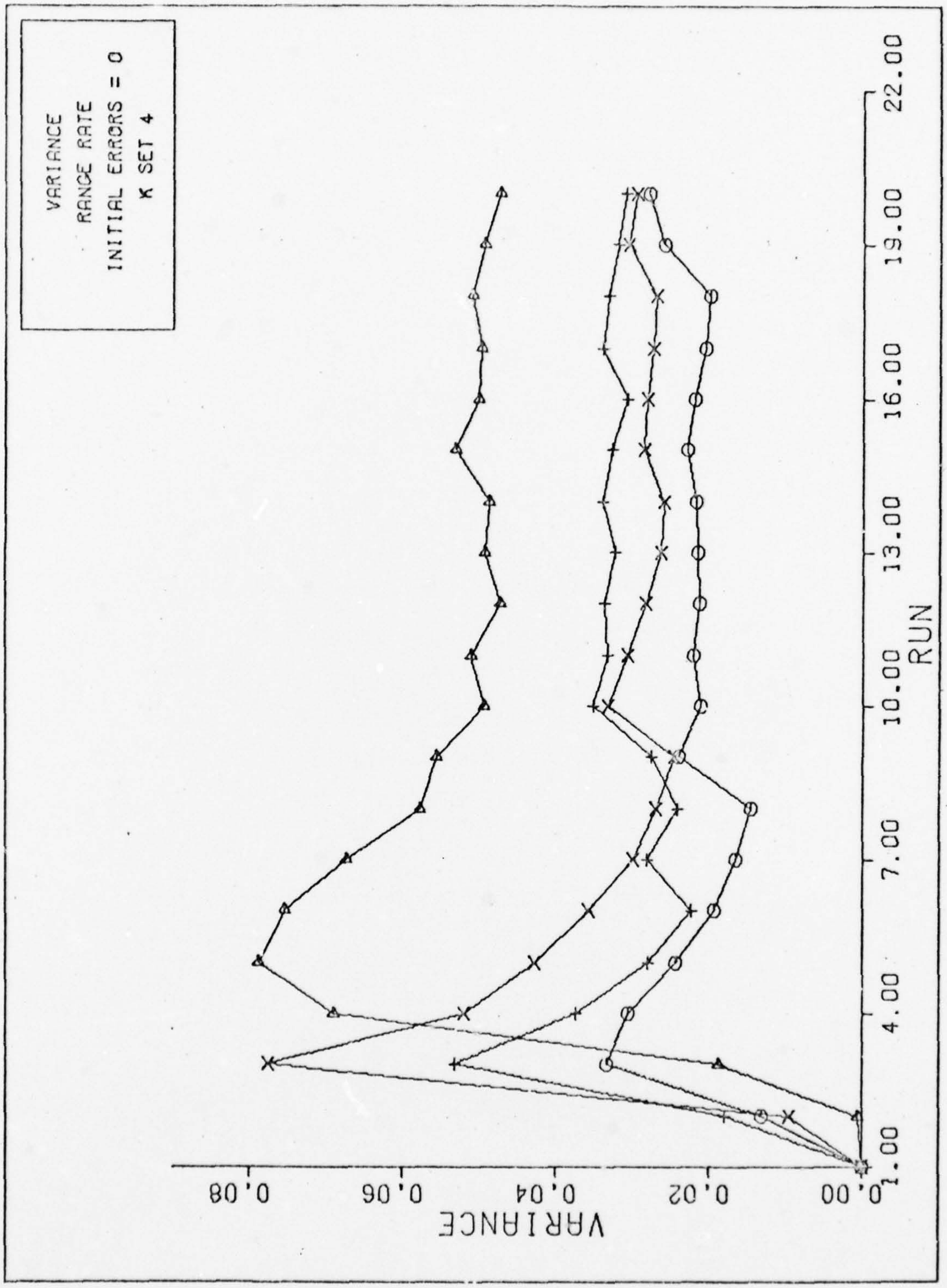



Fig. E-52 VARIANCE CONVERGENCE

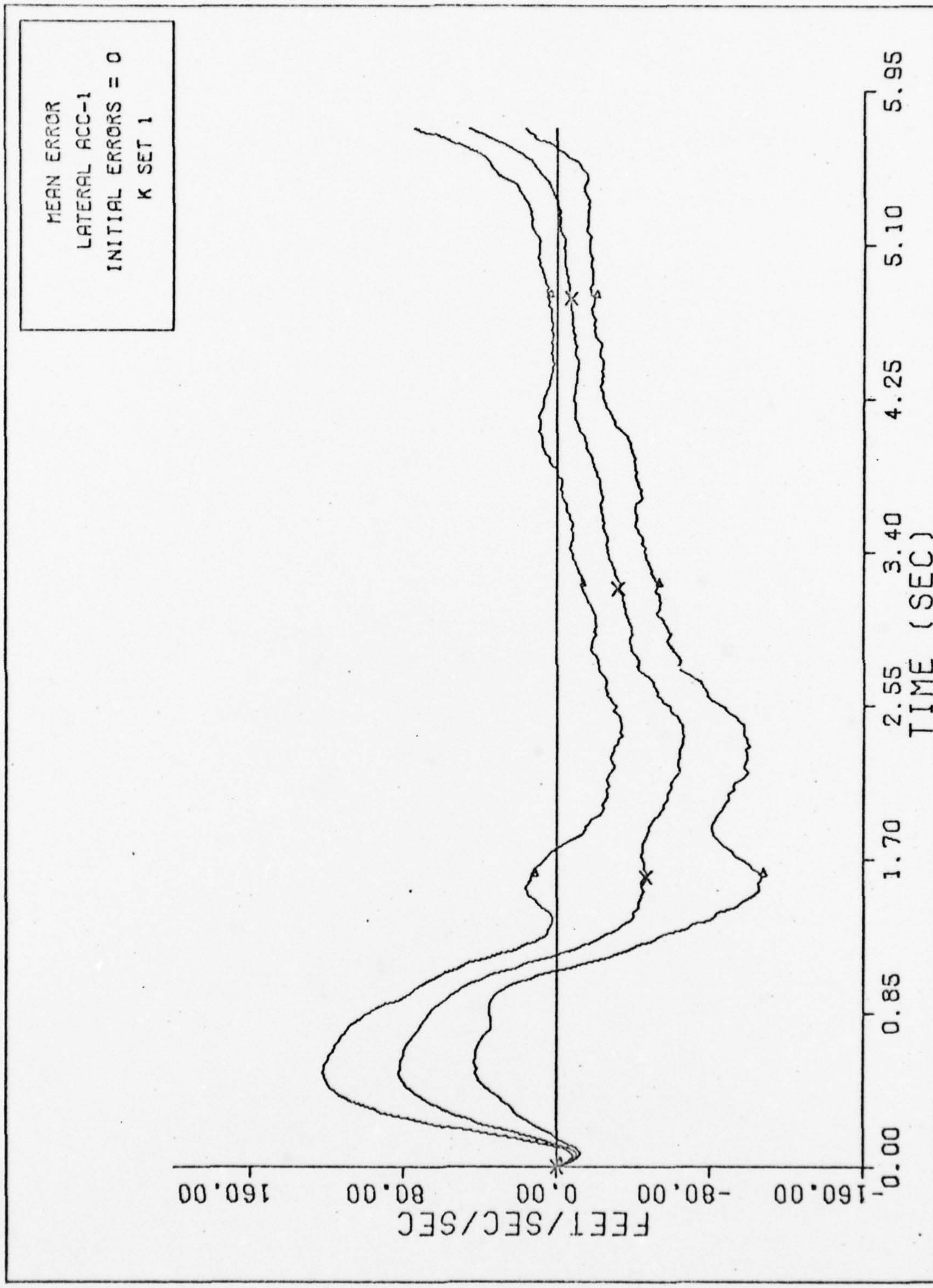



Fig. E-53 LATERAL ACC-1 MEAN ERROR, 11 STATE FILTER

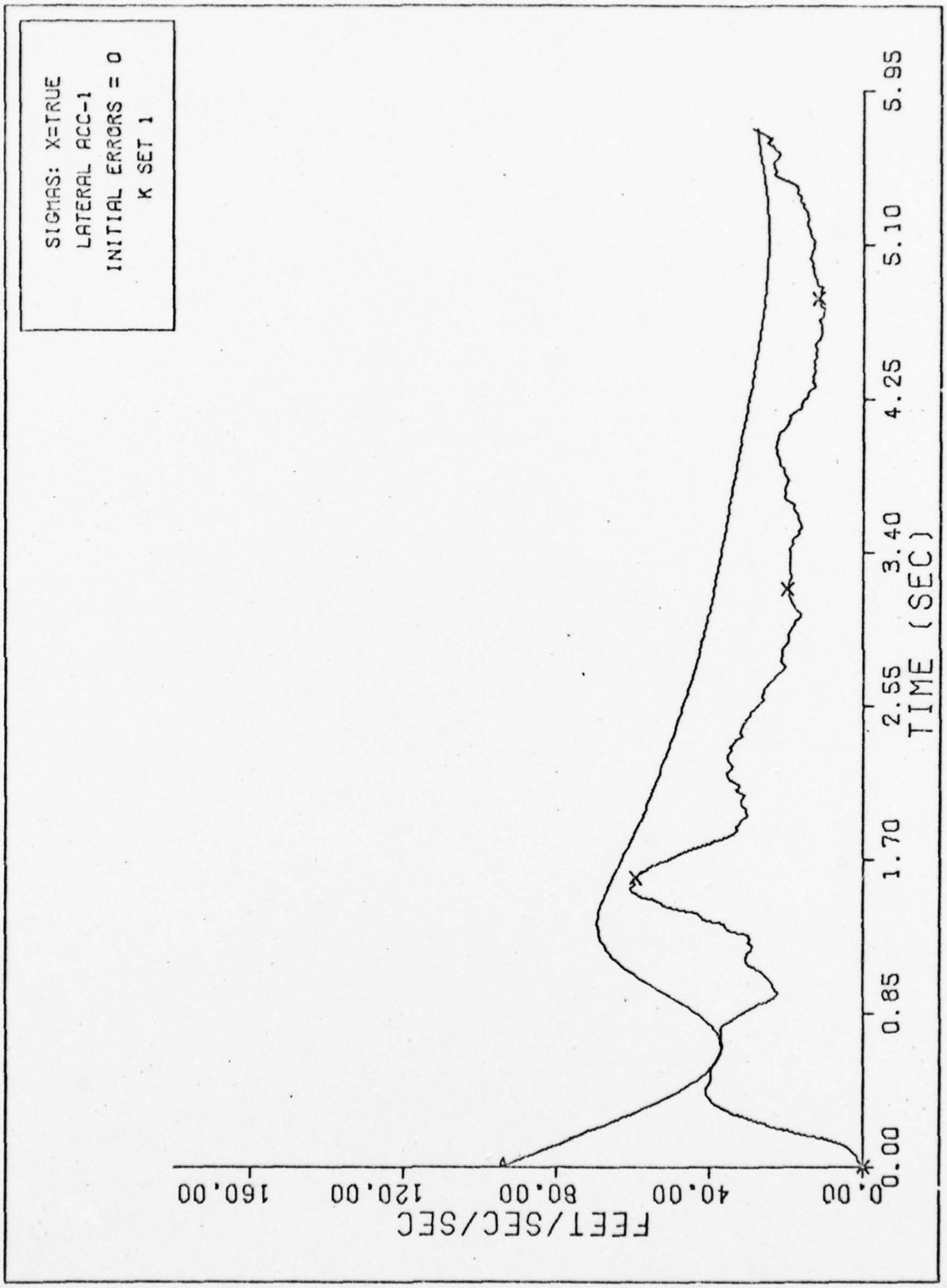



Fig. E-54 LAT ACC-1 FILTER & TRUE SIGMAS, 11 STATE FILTER

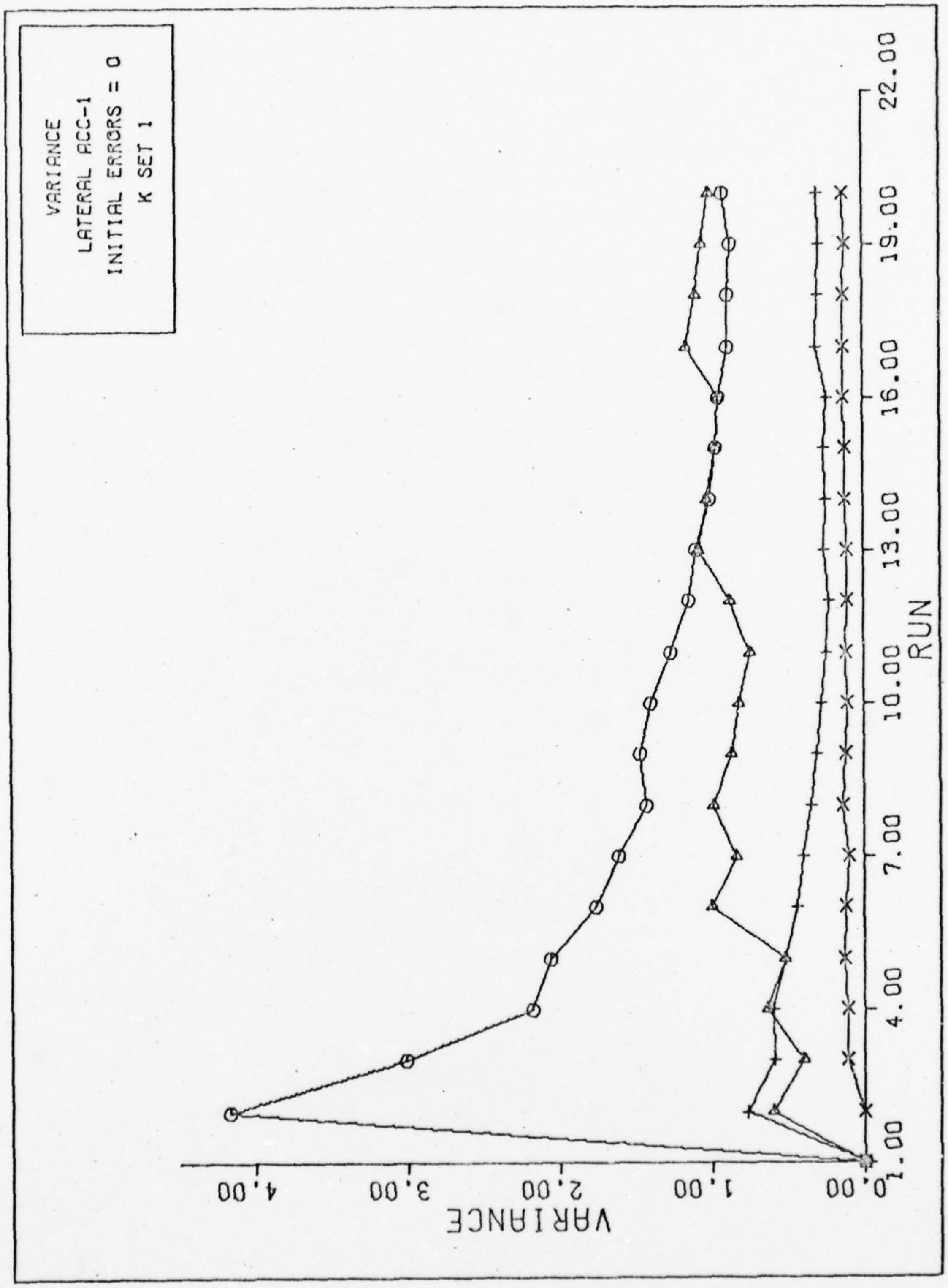



Fig. E-55 VARIANCE CONVERGENCE

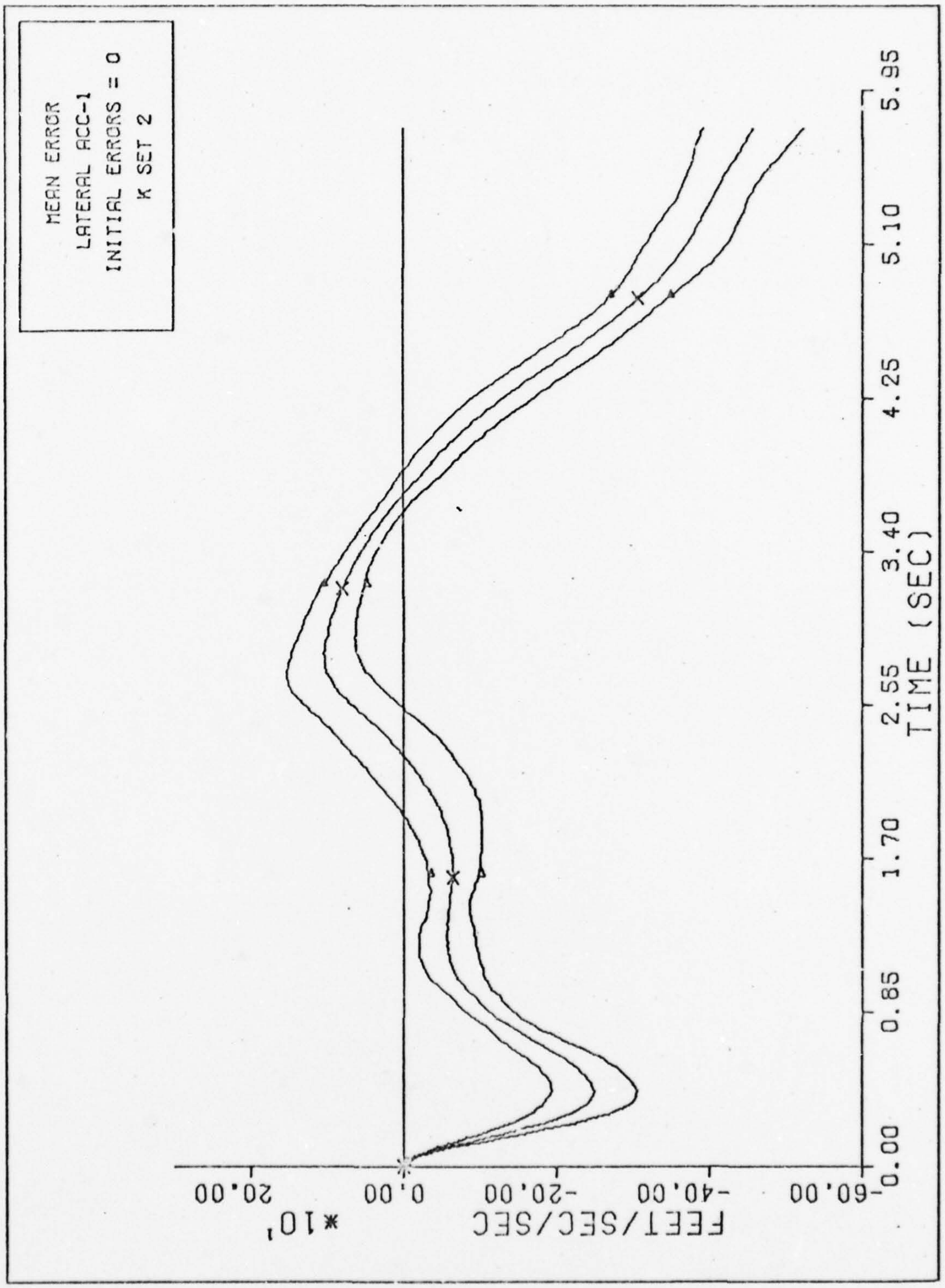



Fig. E-56 LATERAL ACC-1 MEAN ERROR, 11 STATE FILTER

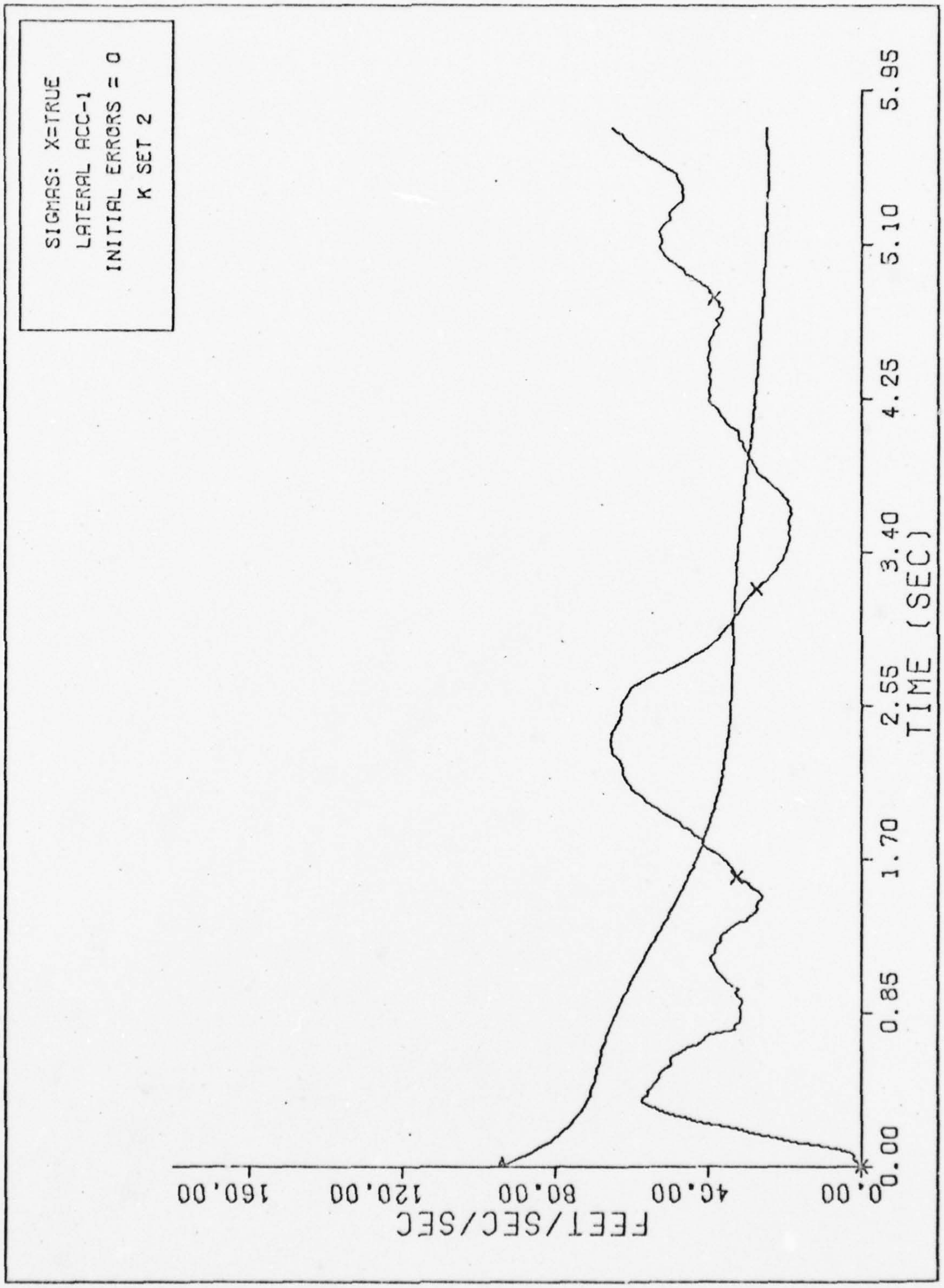



Fig. E-57 LAT ACC-1 FILTER & TRUE SIGMAS, 11 STATE FILTER

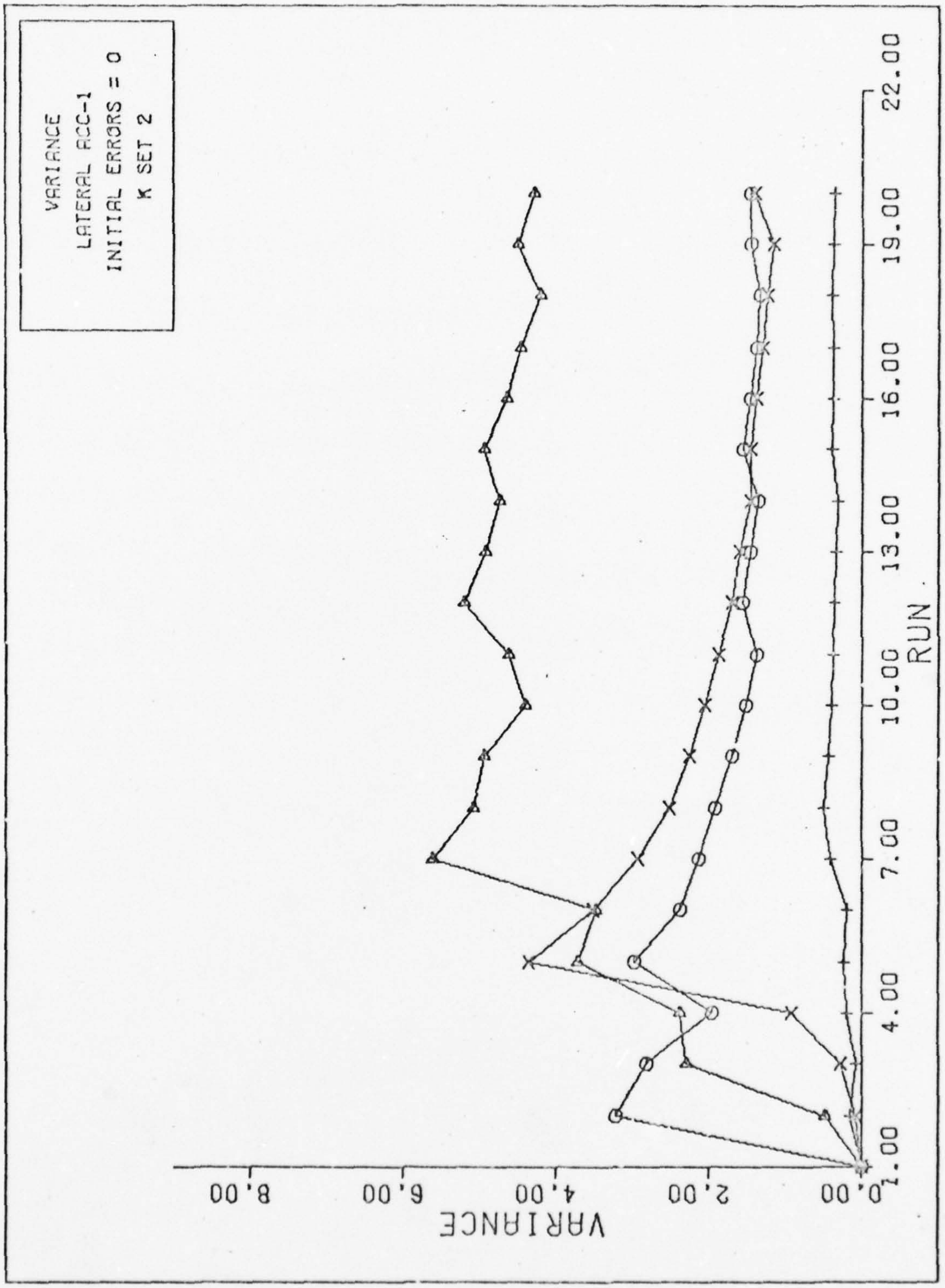



Fig. E-58 VARIANCE CONVERGENCE

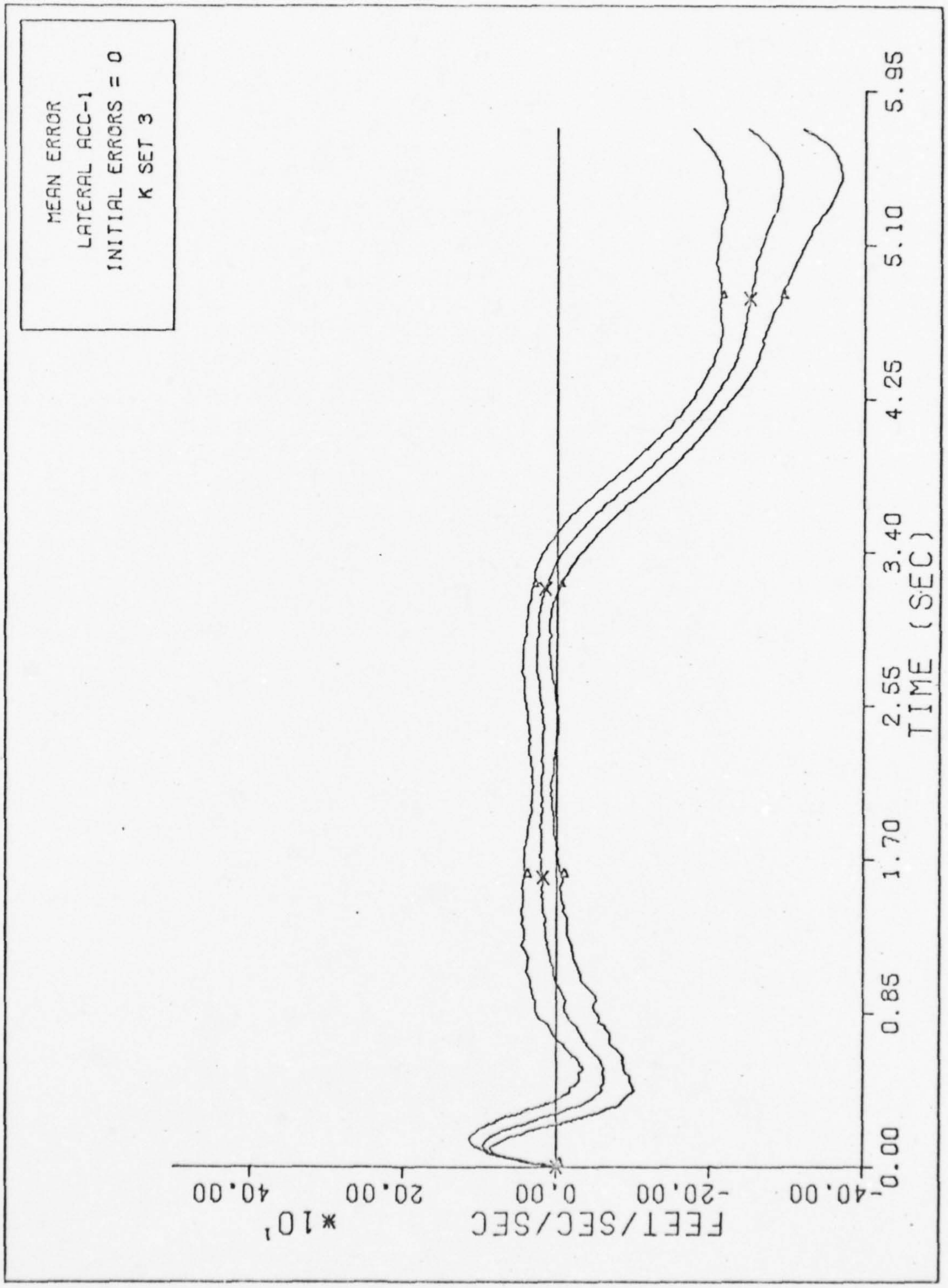



Fig. E-59 LATERAL ACC-1 MEAN ERROR, 11 STATE FILTER

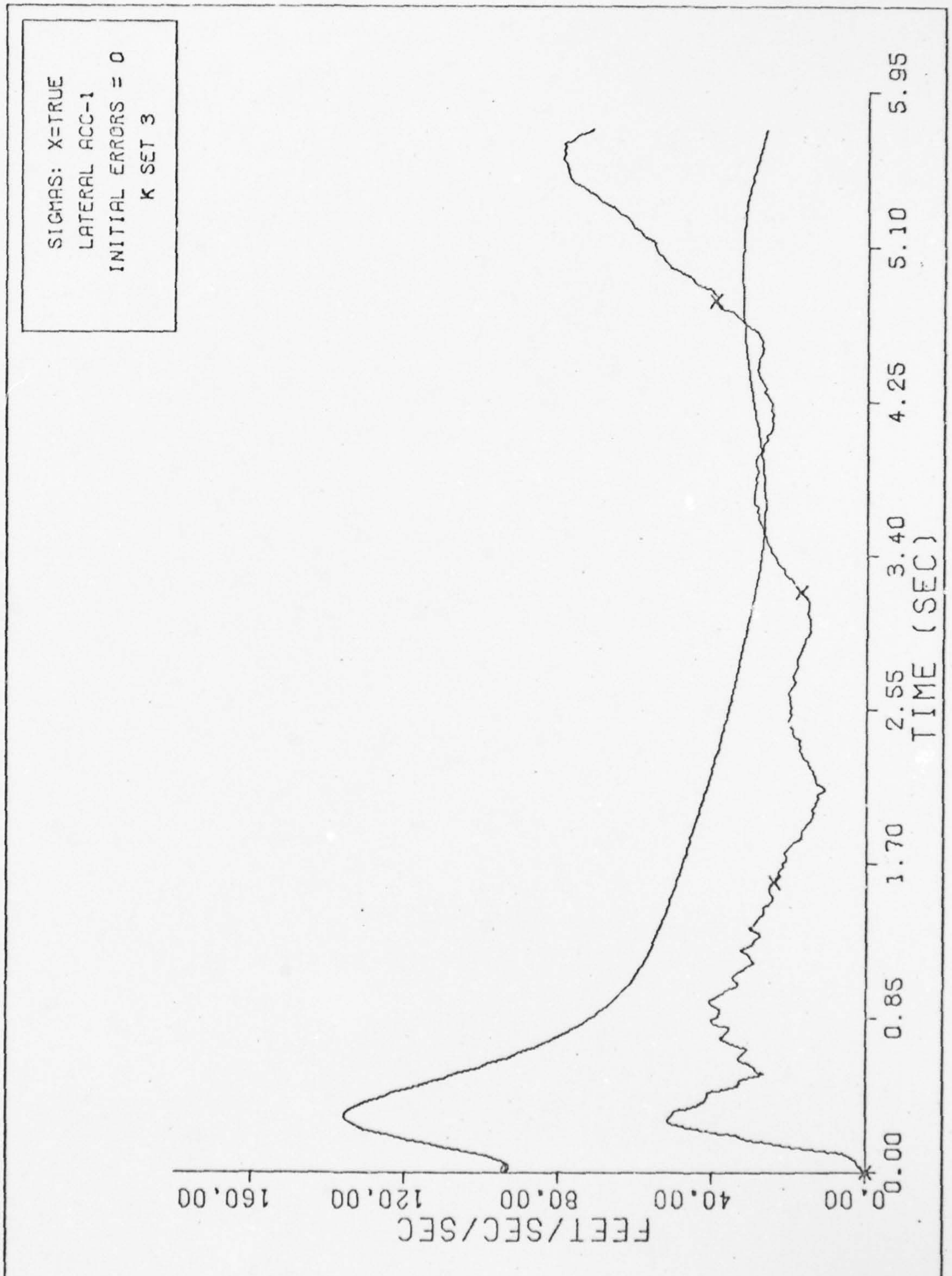



Fig. E-60 LAT ACC-1 FILTER & TRUE SIGMAS, 11 STATE FILTER

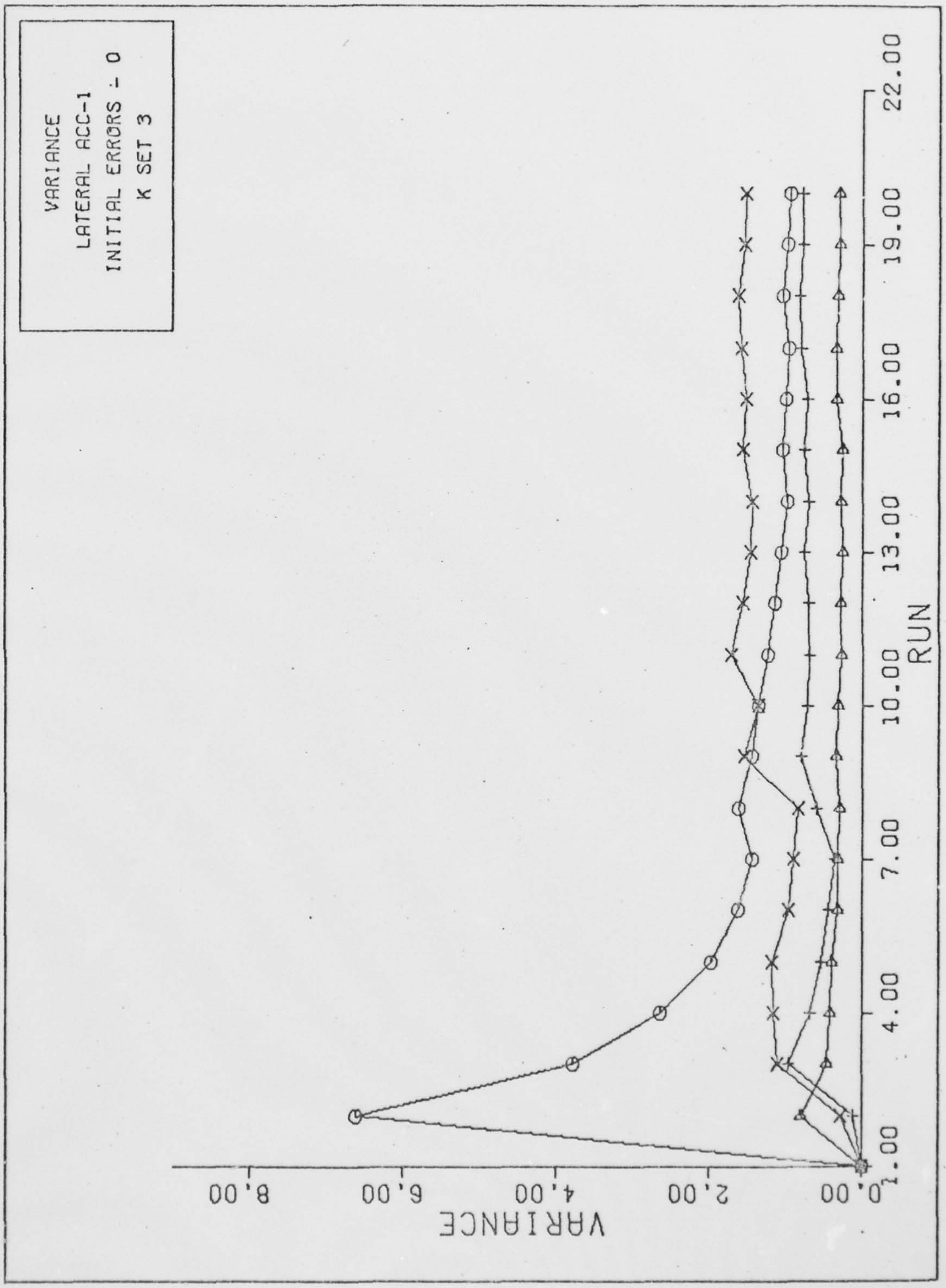



Fig. E-61 VARIANCE CONVERGENCE

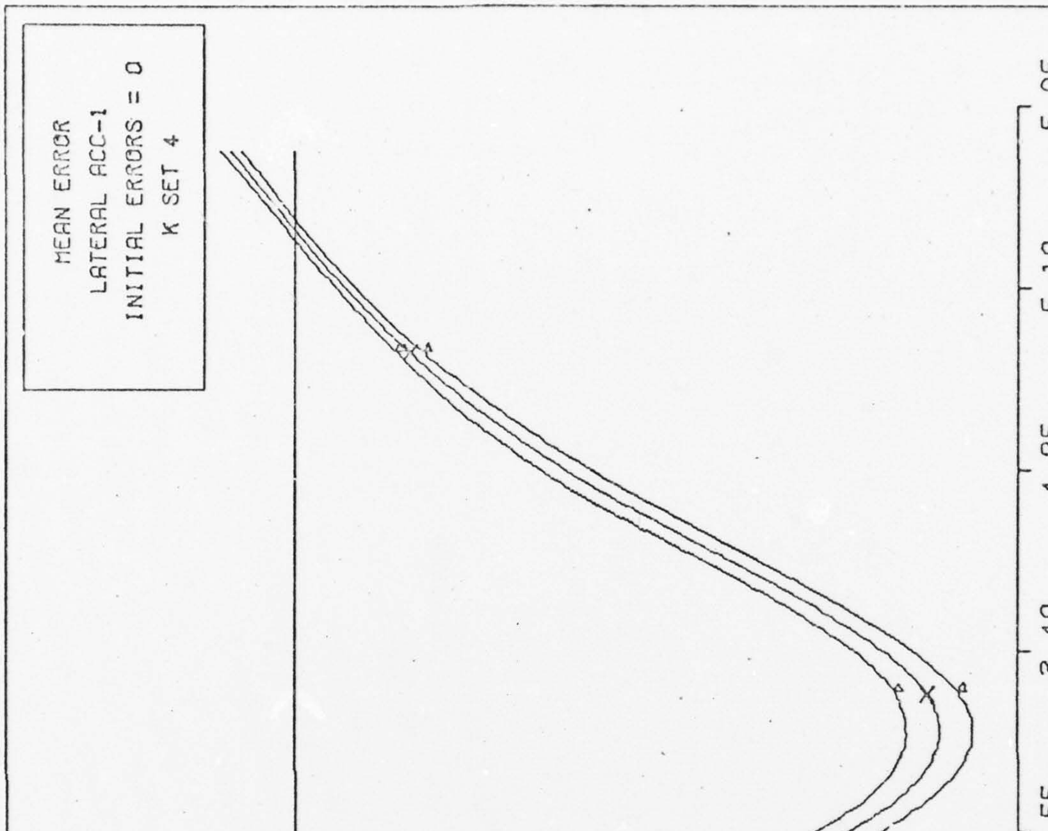



Fig. E-62 LATERAL ACC-1 MEAN ERROR, 11 STATE FILTER

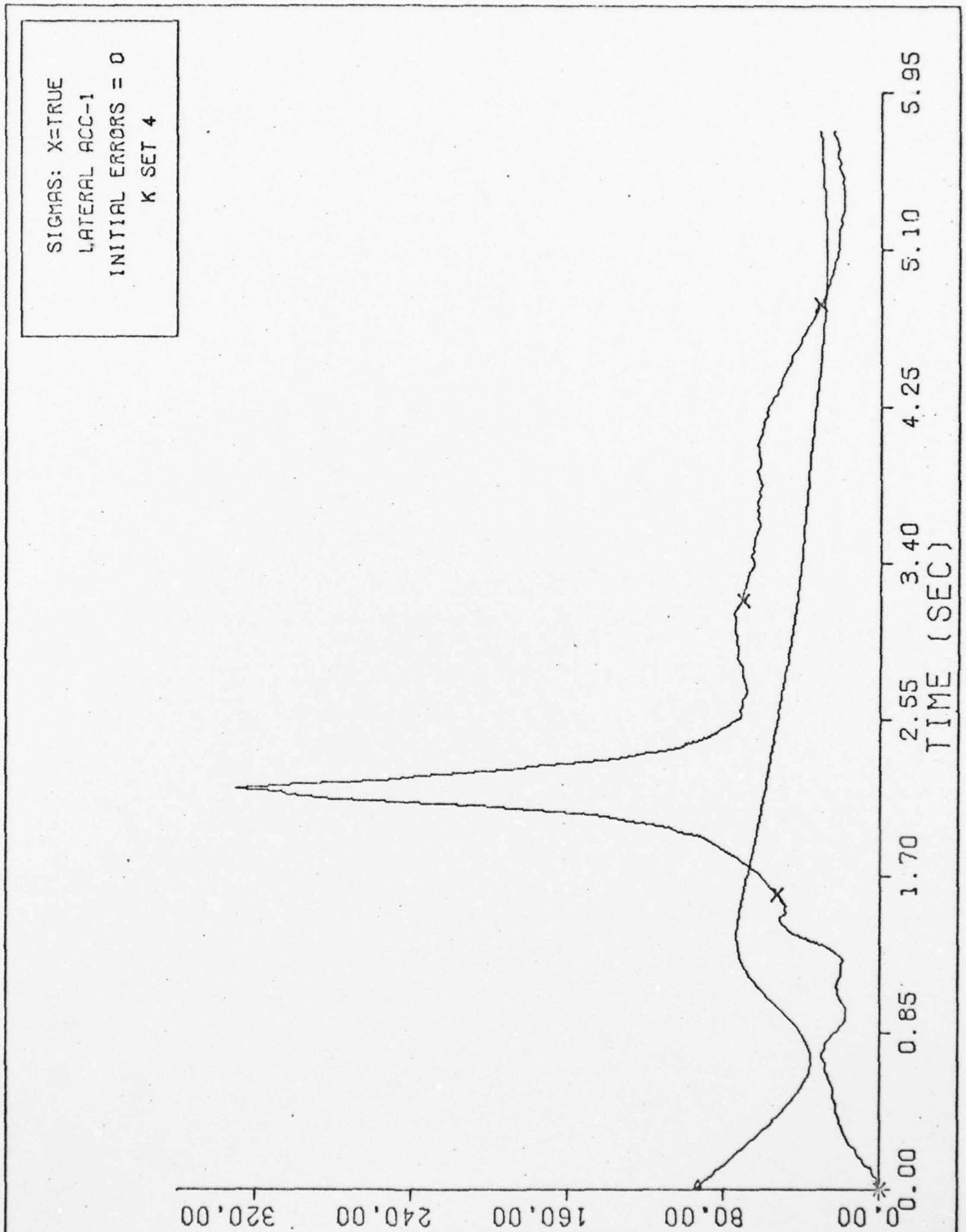



Fig. E-63 LAT ACC-1 FILTER & TRUE SIGMAS, 11 STATE FILTER

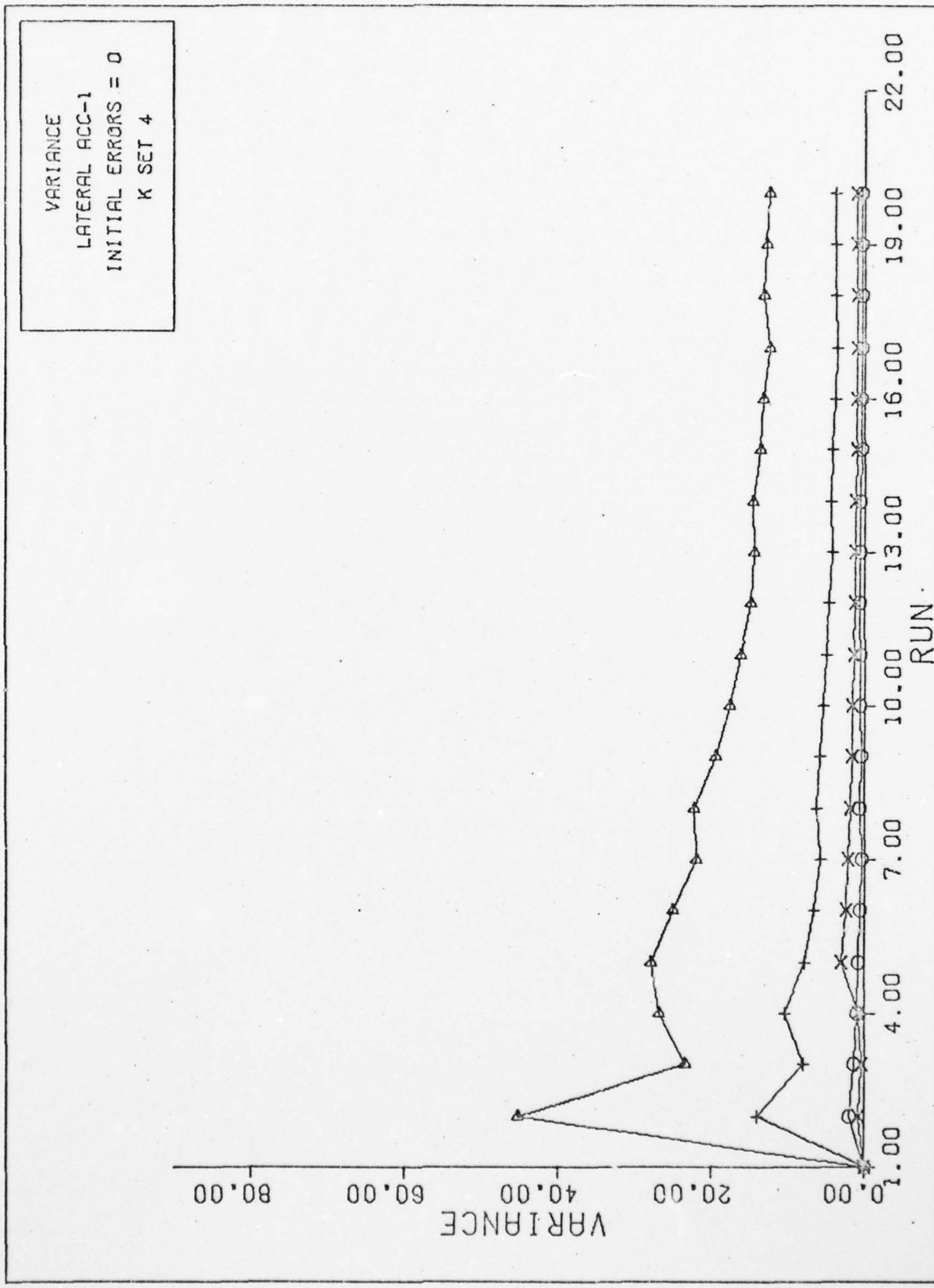



Fig. E-64

VARIANCE CONVERGENCE

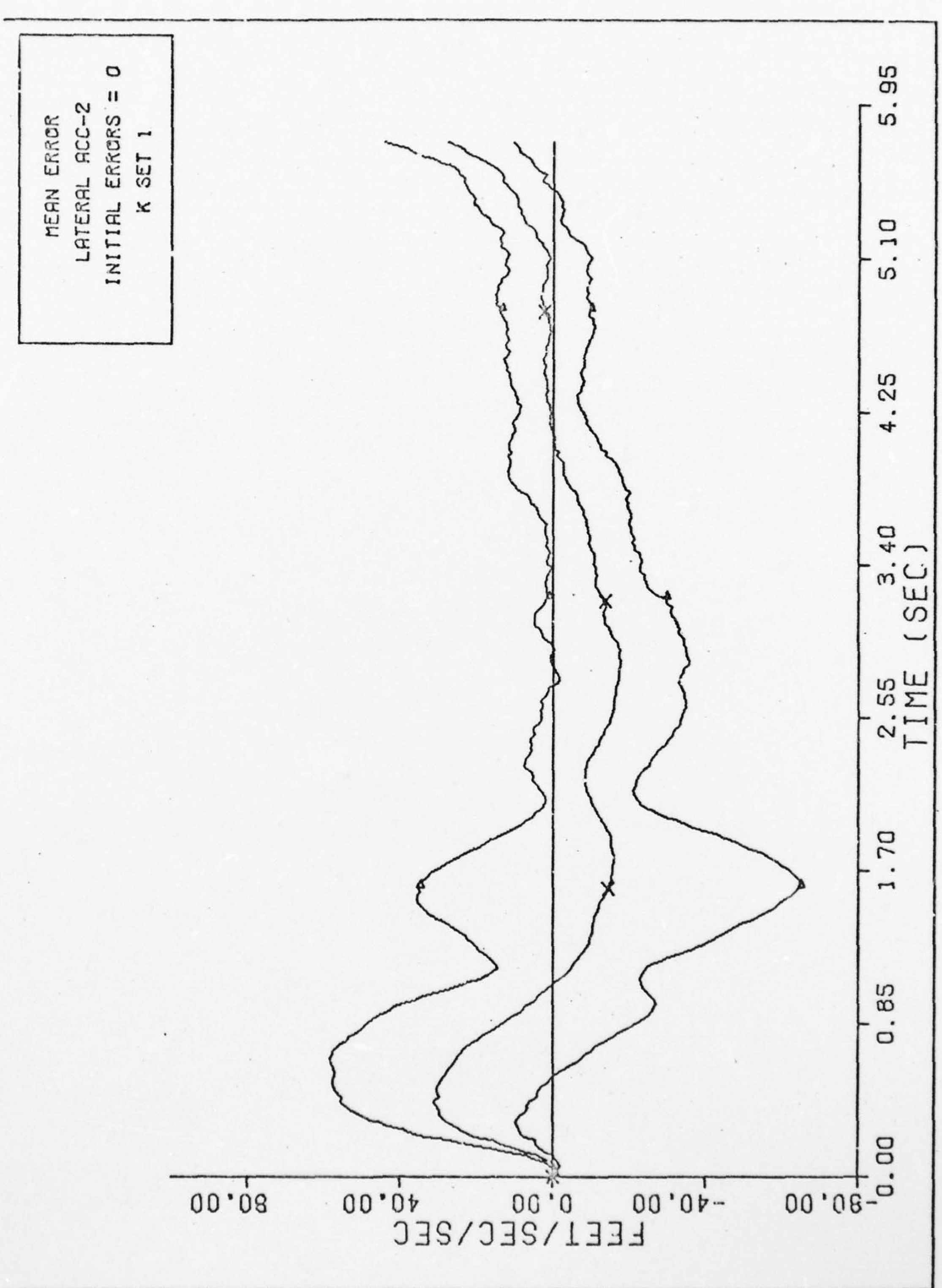



Fig. E-65 LATERAL ACC-2 MEAN ERROR, 11 STATE FILTER

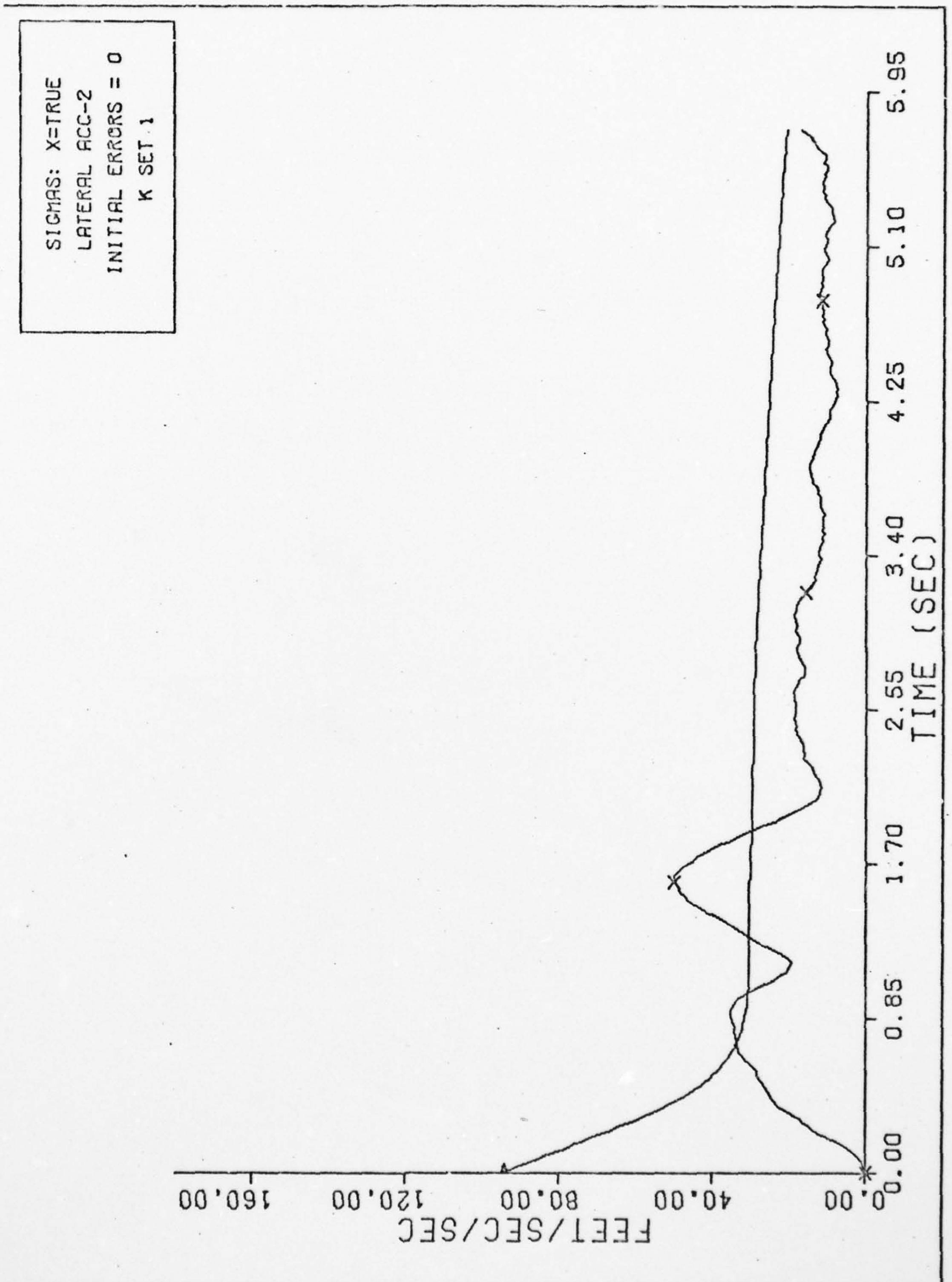



Fig. E-66 LAT ACC-2 FILTER & TRUE SIGMAS, 11 STATE FILTER

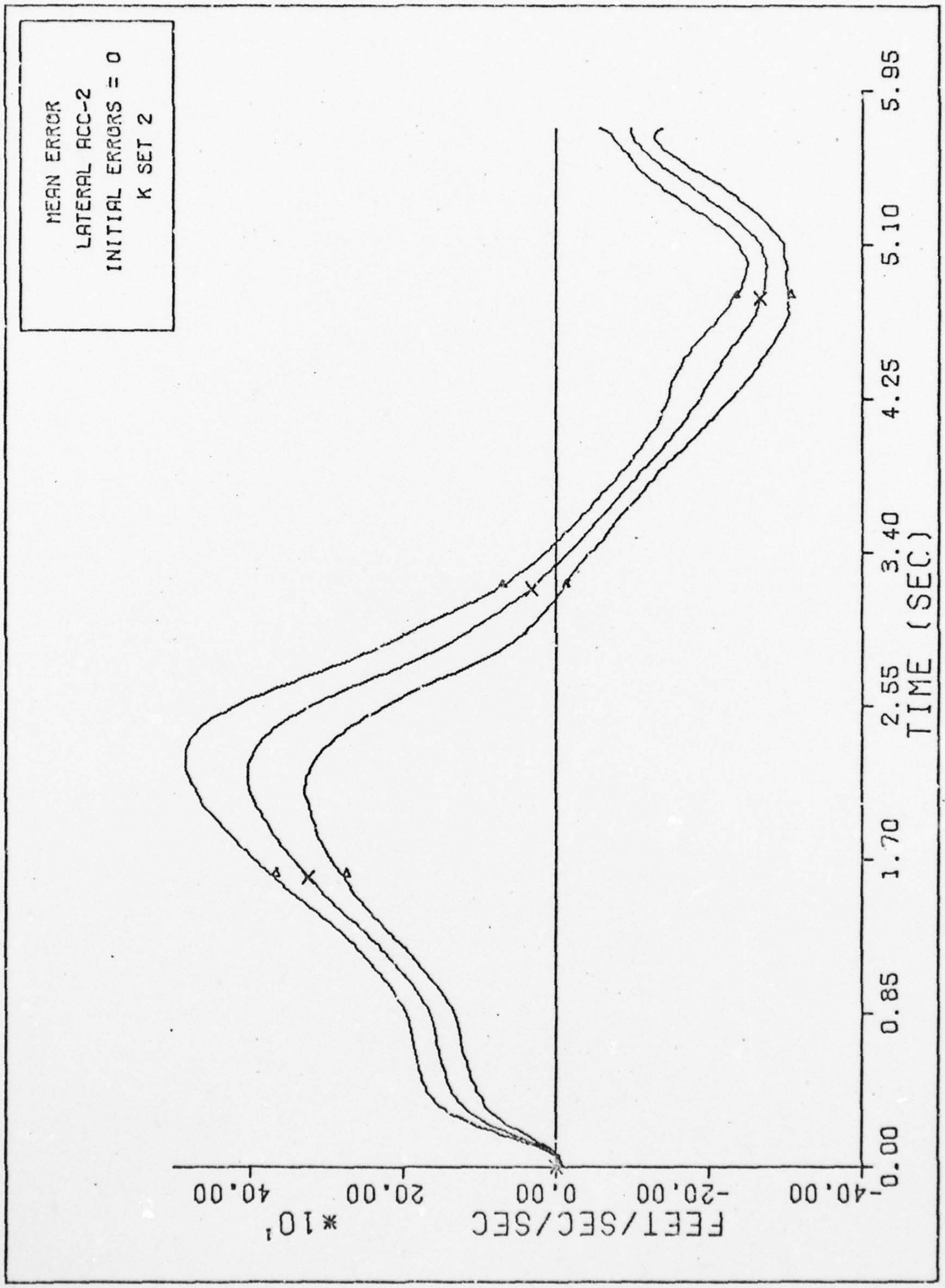



Fig. E-67 LATERAL ACC-2 MEAN ERROR, 11 STATE FILTER

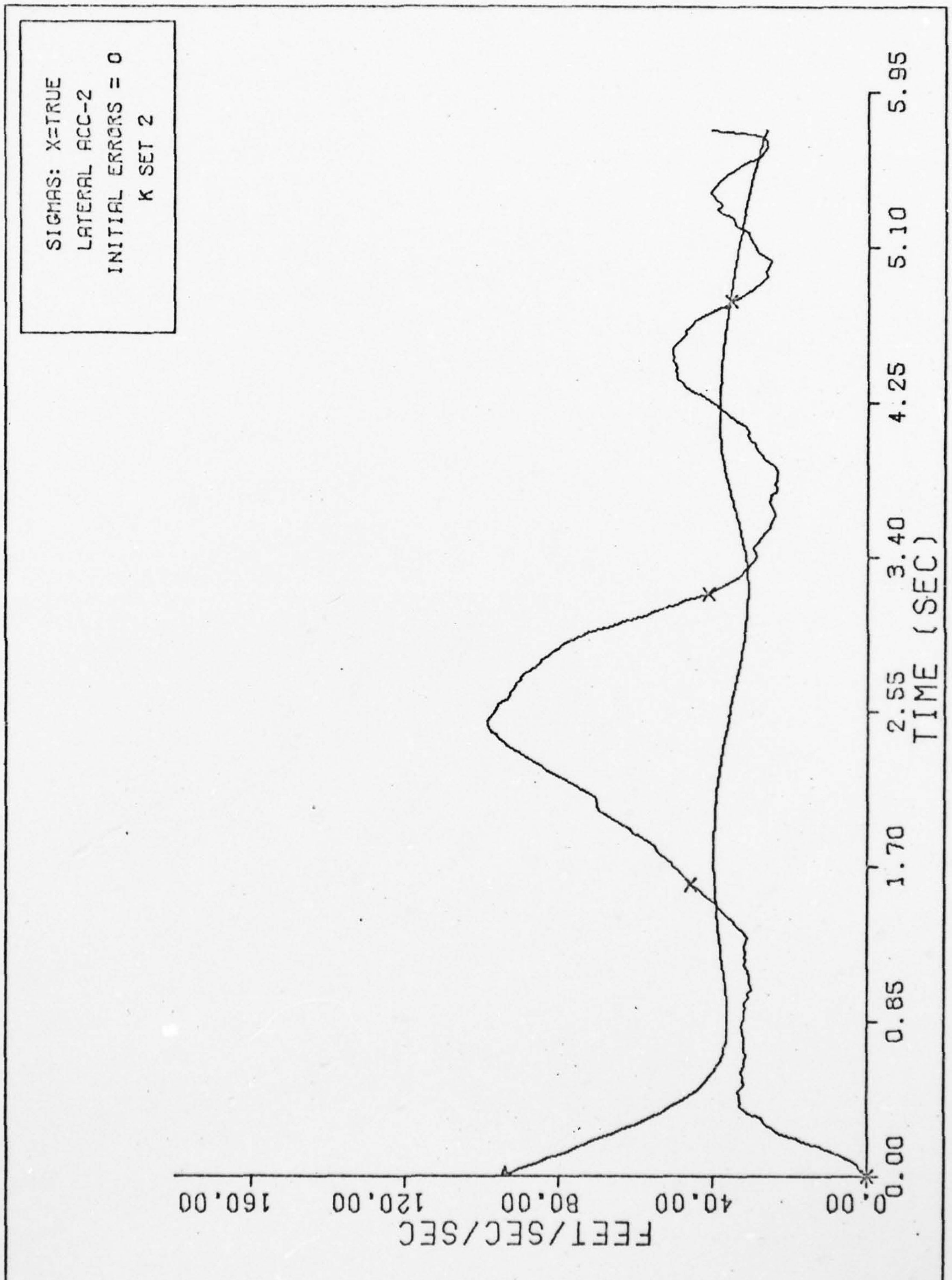



Fig. E-68 LAT ACC-2 FILTER & TRUE SIGMAS, 11 STATE FILTER

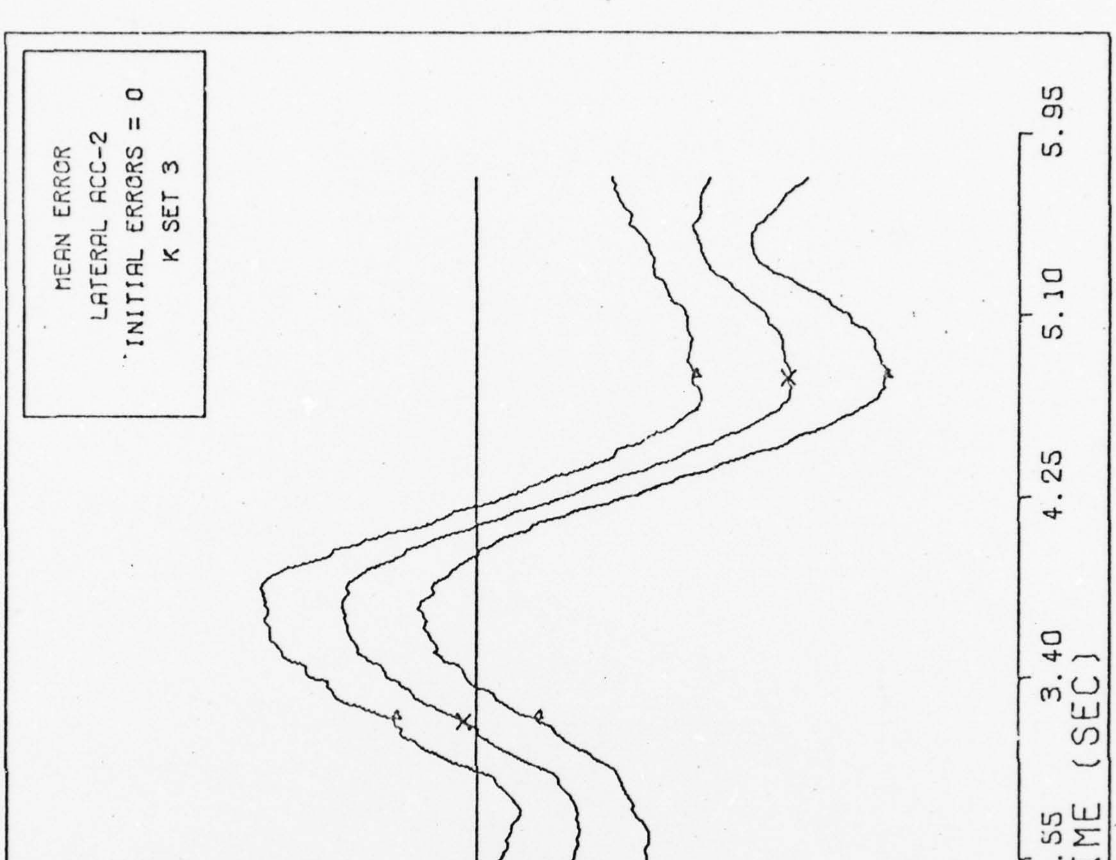



Fig. E-69 LATERAL ACC-2 MEAN ERROR, 11 STATE FILTER

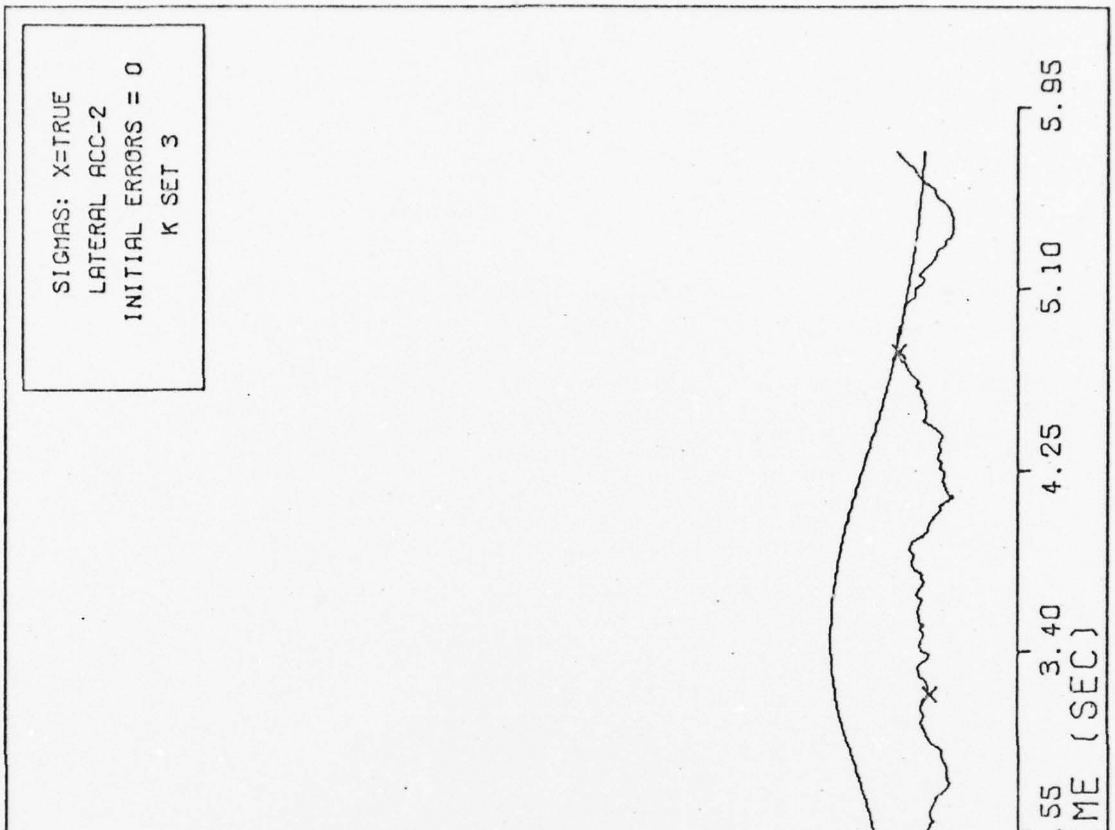



Fig. E-70 LAT ACC-2 FILTER & TRUE SIGMAS, 11 STATE FILTER

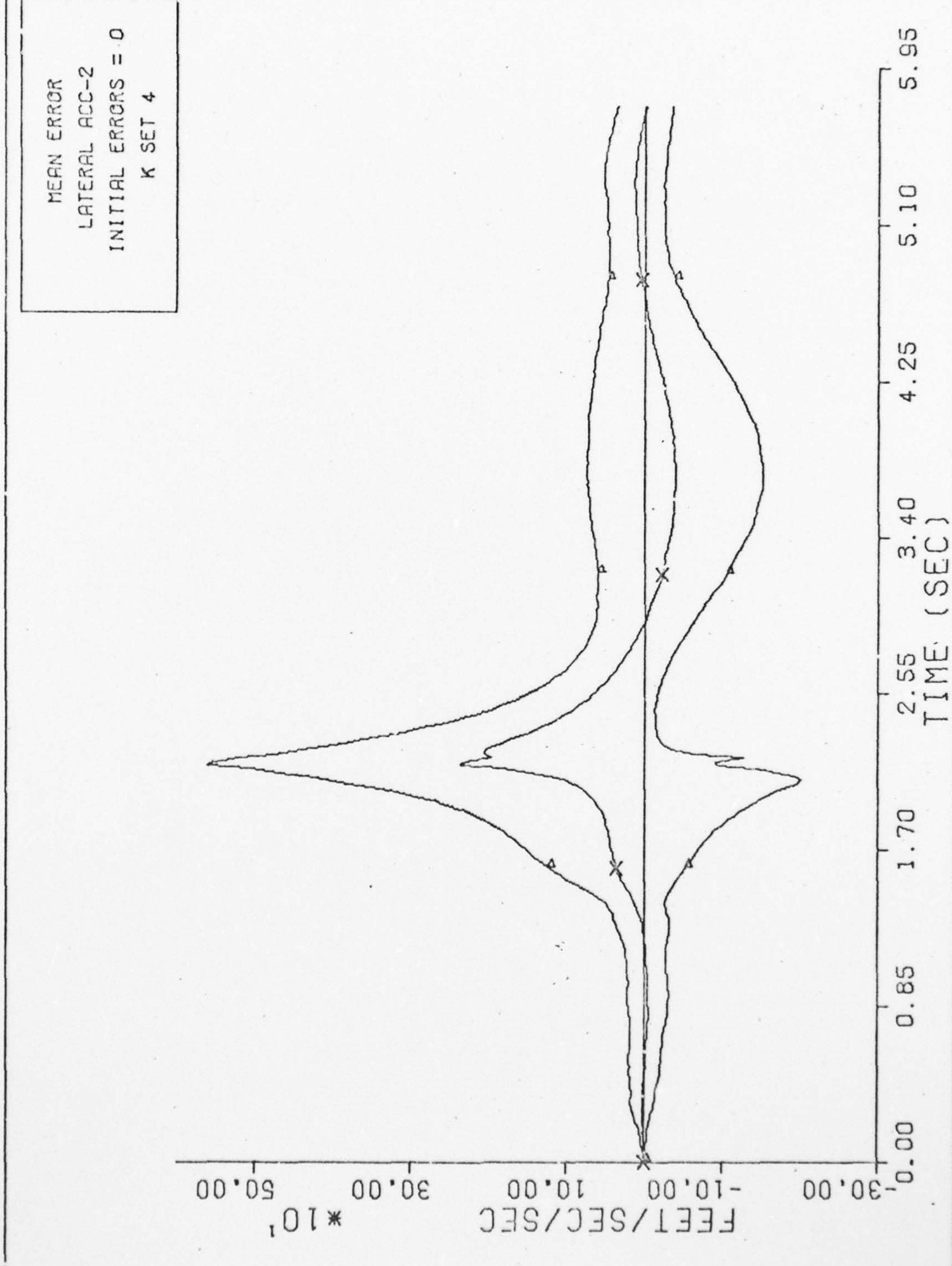



Fig. E-71 LATERAL ACC-2 MEAN ERROR, 11 STATE FILTER

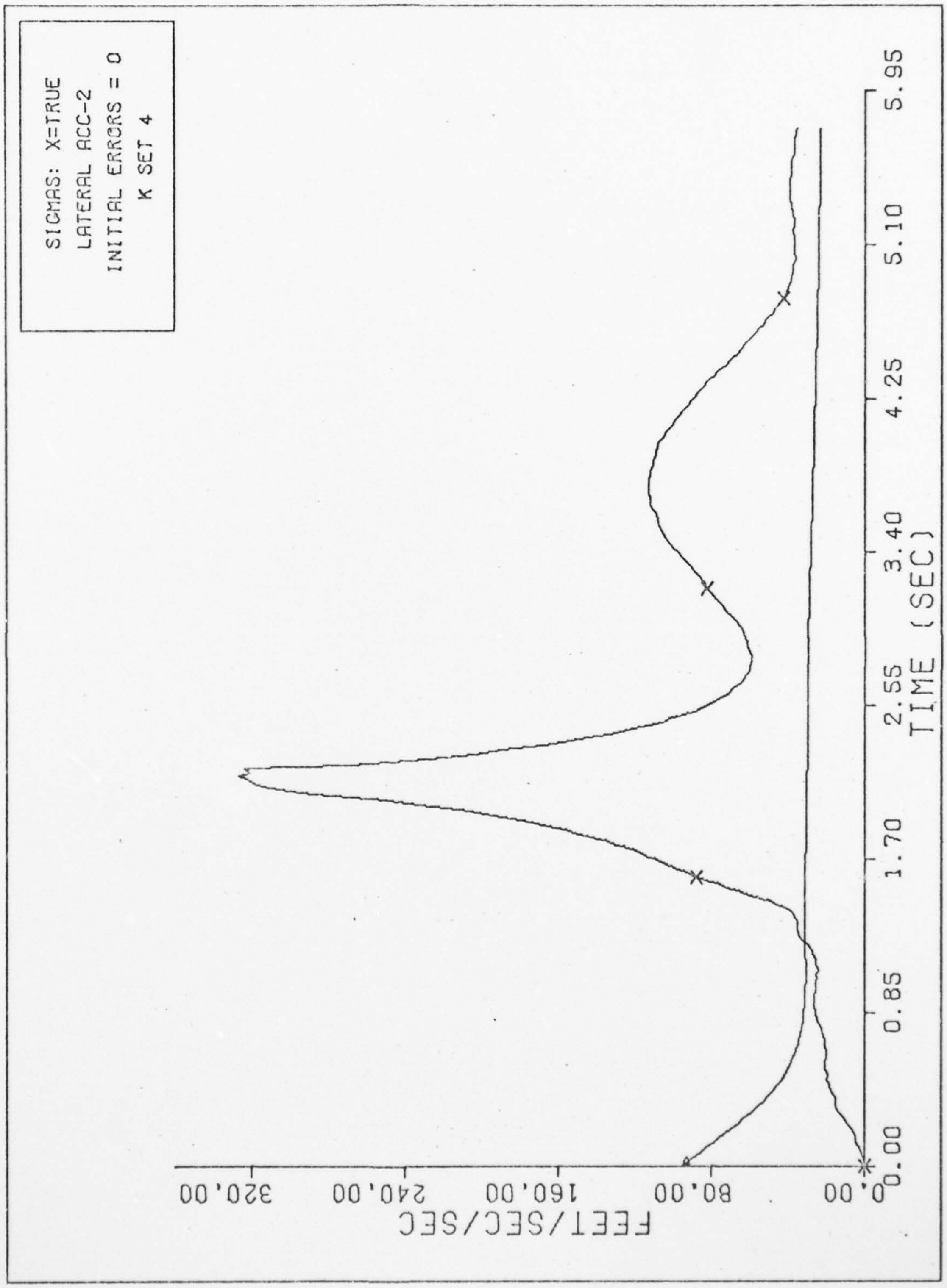



Fig. E-72 LAT ACC-2 FILTER & TRUE SIGMAS, 11 STATE FILTER

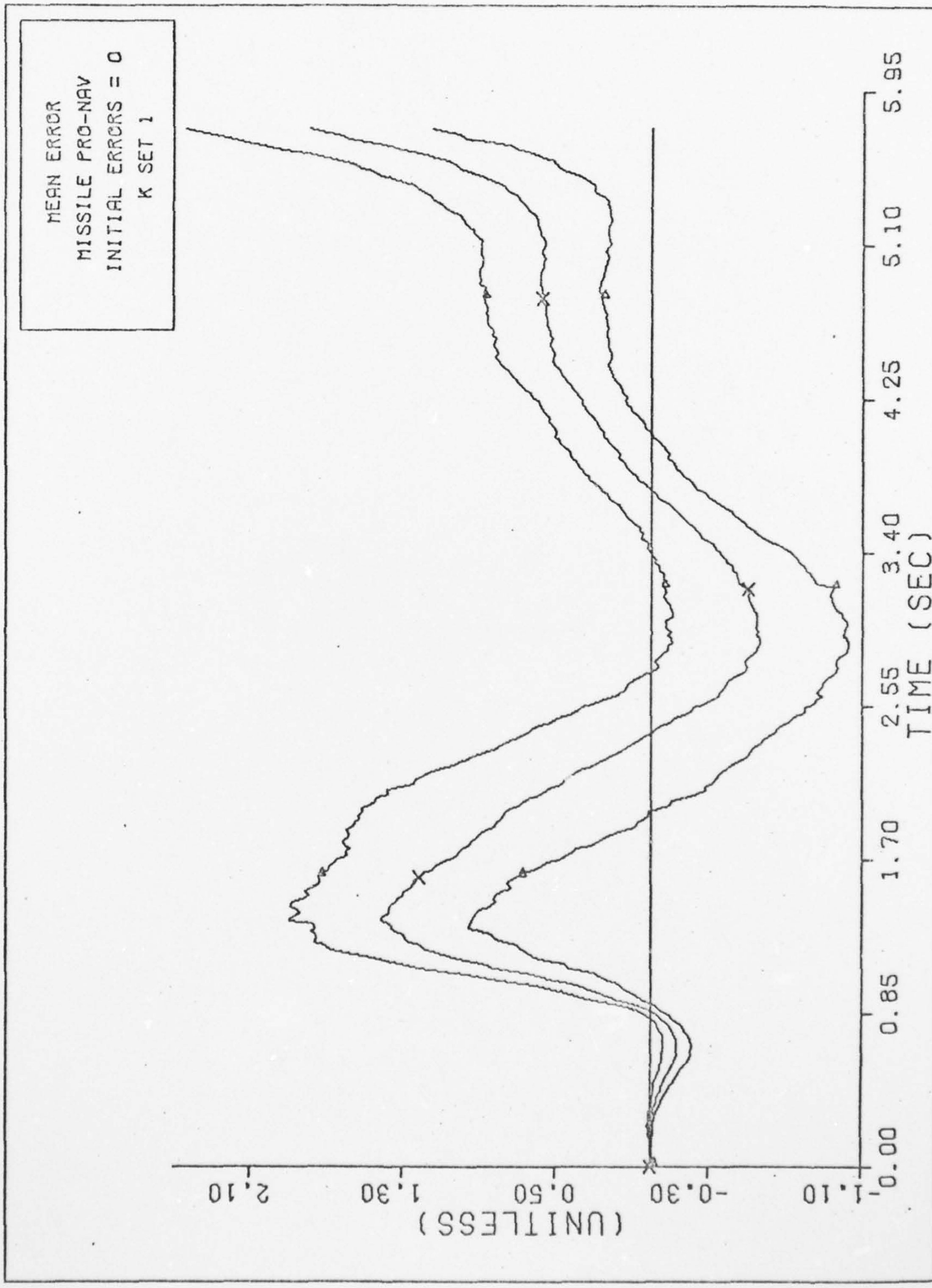



Fig. E-73 MISSILE PRO-NAV MEAN ERROR, 11 STATE FILTER

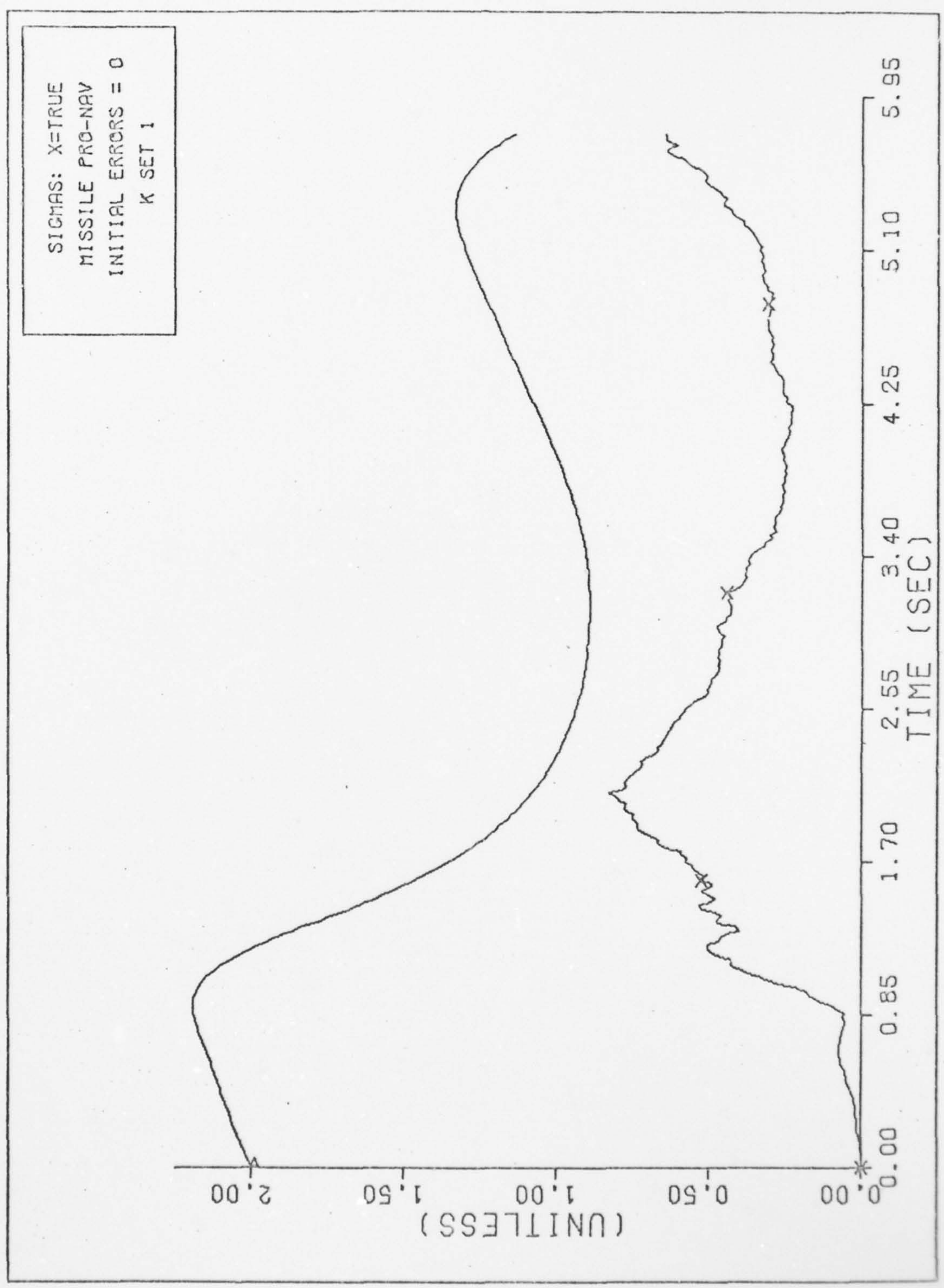
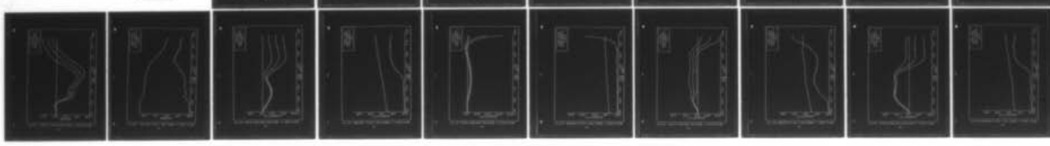


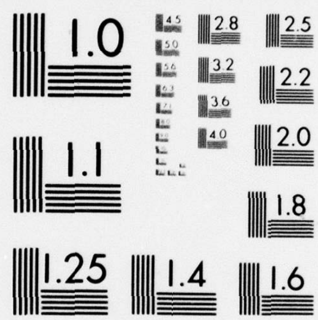

Fig. E-74 PRO-NAV FILTER & TRUE SIGMAS, 11 STATE FILTER

AD-A064 760 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OHIO SCH--ETC F/G 15/3.1
A PRACTICAL THREE DIMENSIONAL, 11 STATE EXTENDED KALMAN FILTER --ETC(U)
DEC 78 C W HLAVATY
UNCLASSIFIED AFIT/GGC/EE/78-7-VOL-2 NL

3 OF 3
AD
A064760



END
DATE
FILMED
4-79
DDC




MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

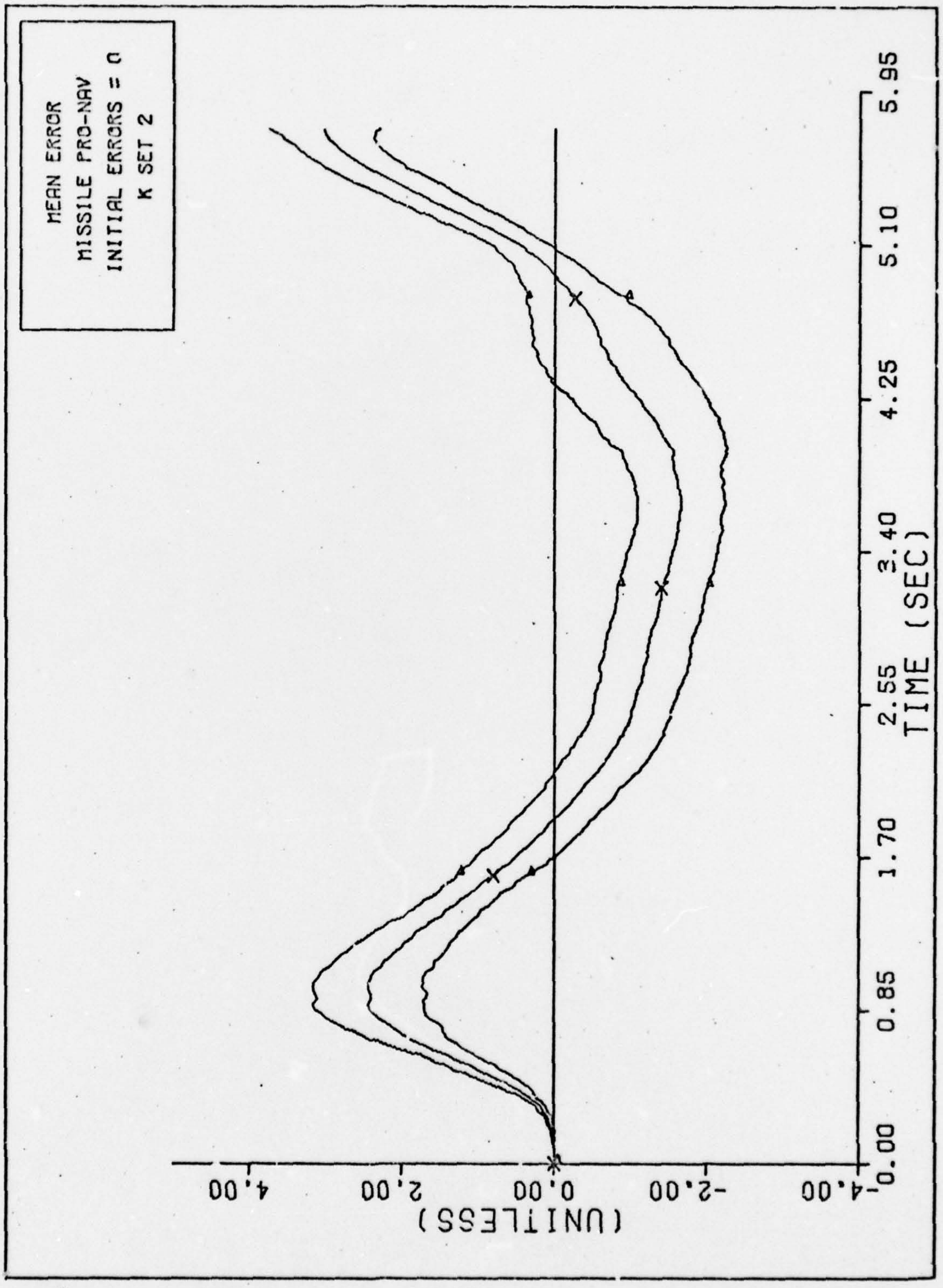



Fig. E-75 MISSILE PRO-NAV MEAN ERROR, 11 STATE FILTER

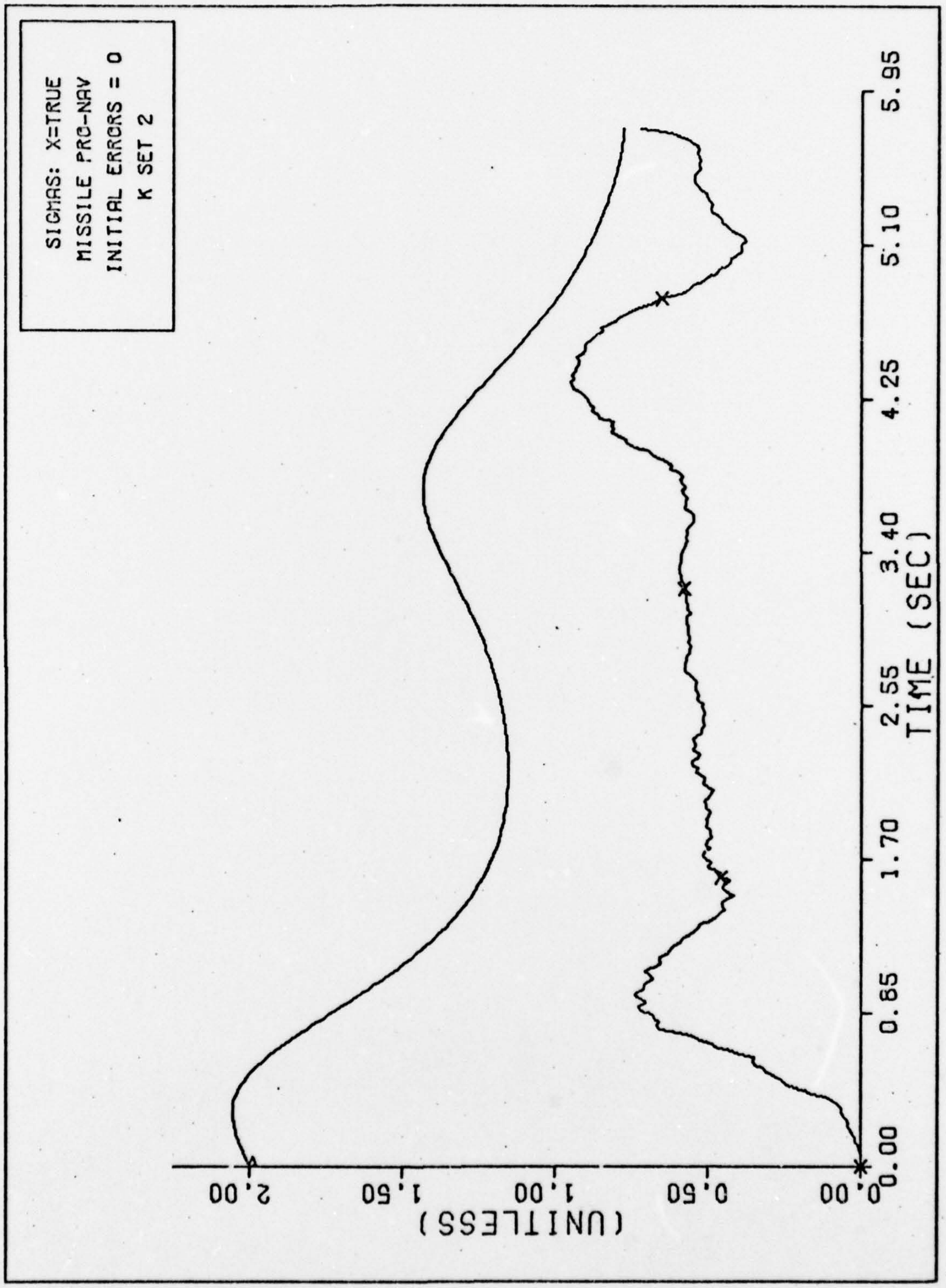



Fig. E-76 PRO-NAV FILTER & TRUE SIGMAS, 11 STATE FILTER

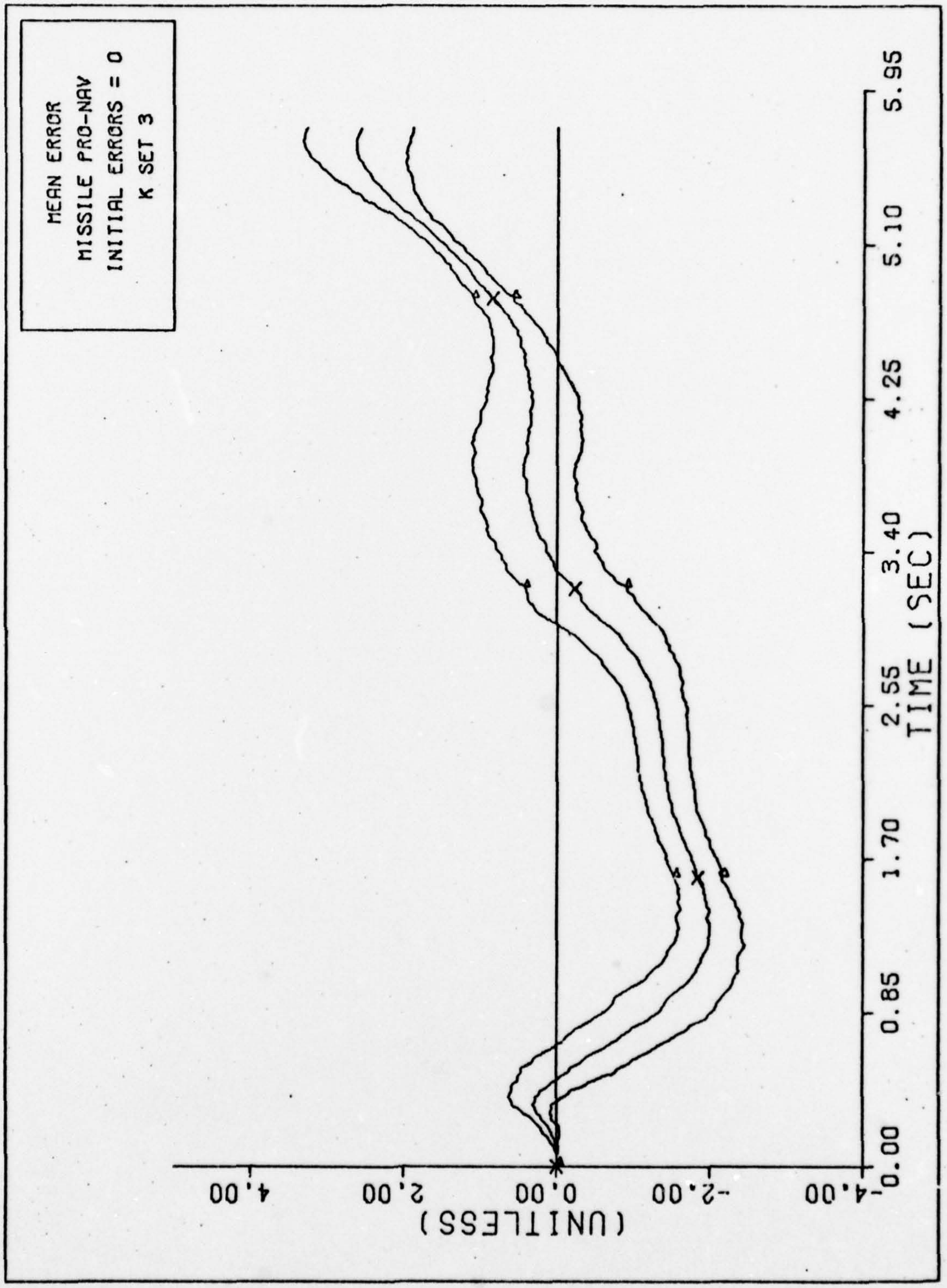



Fig. E-77 MISSILE PRO-NAV MEAN ERROR, 11 STATE FILTER

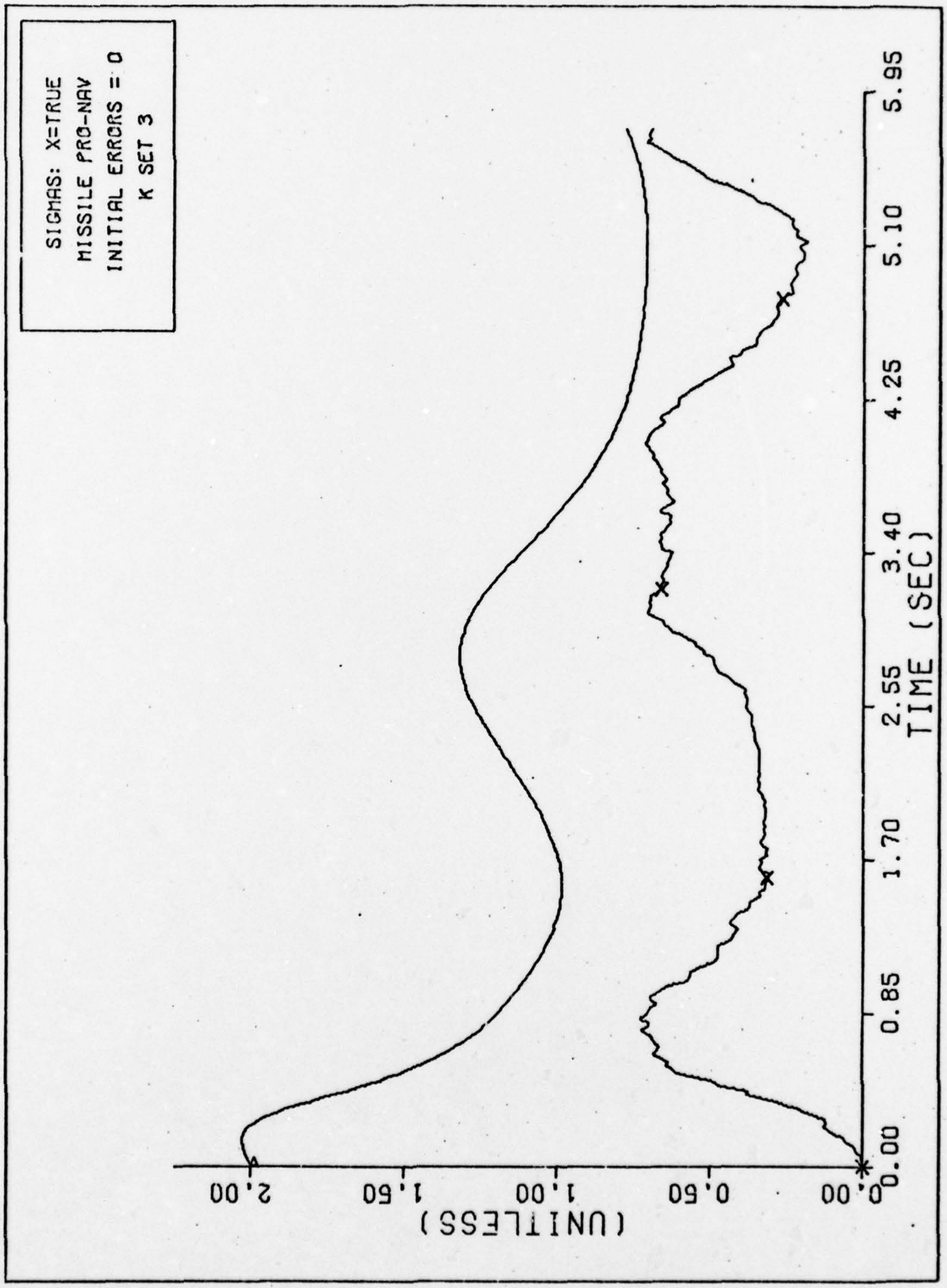



Fig. E-78 PRO-NAV FILTER & TRUE SIGMAS, 11 STATE FILTER

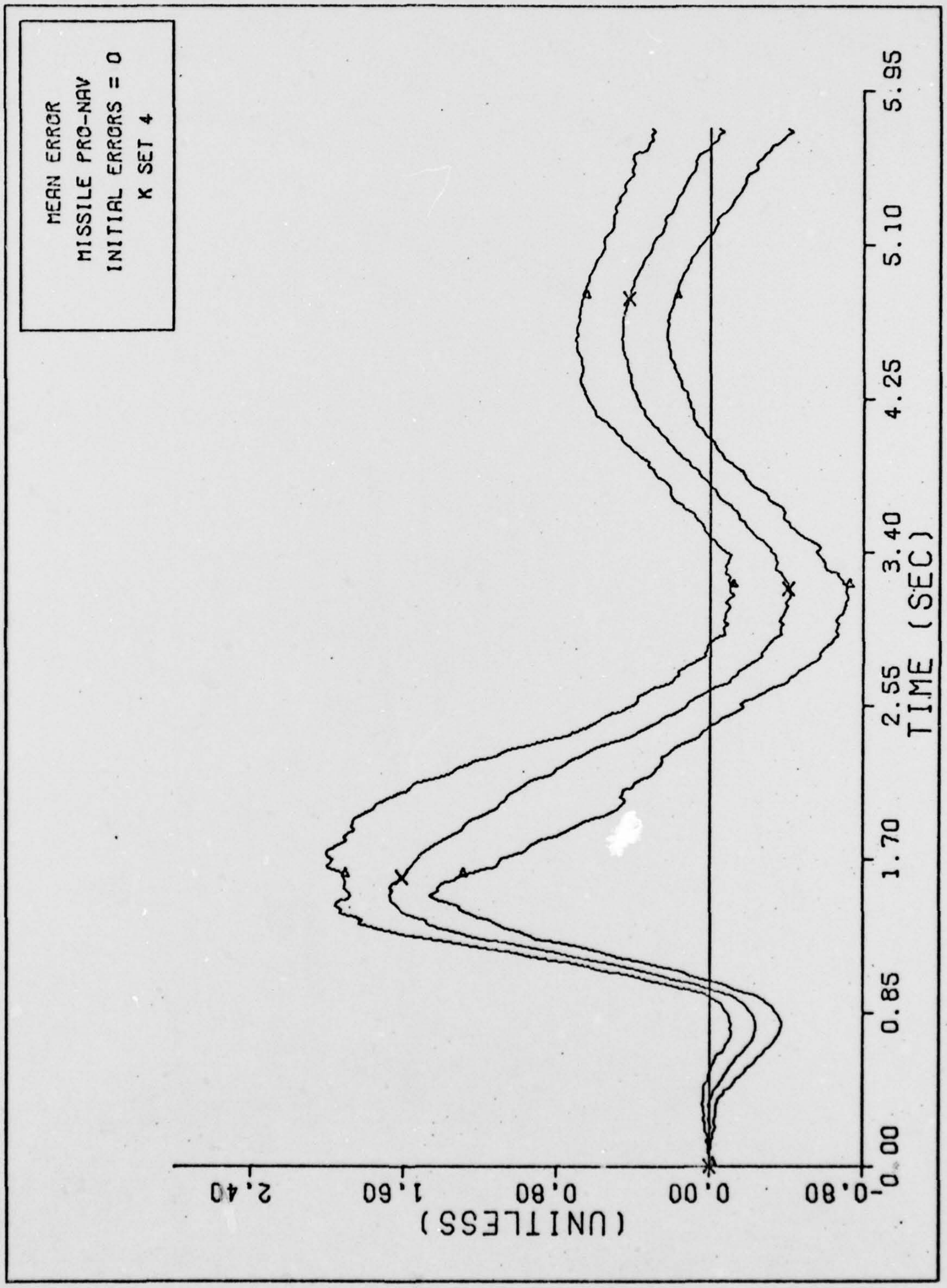



Fig. E-79 MISSILE PRO-NAV MEAN ERROR, 11 STATE FILTER

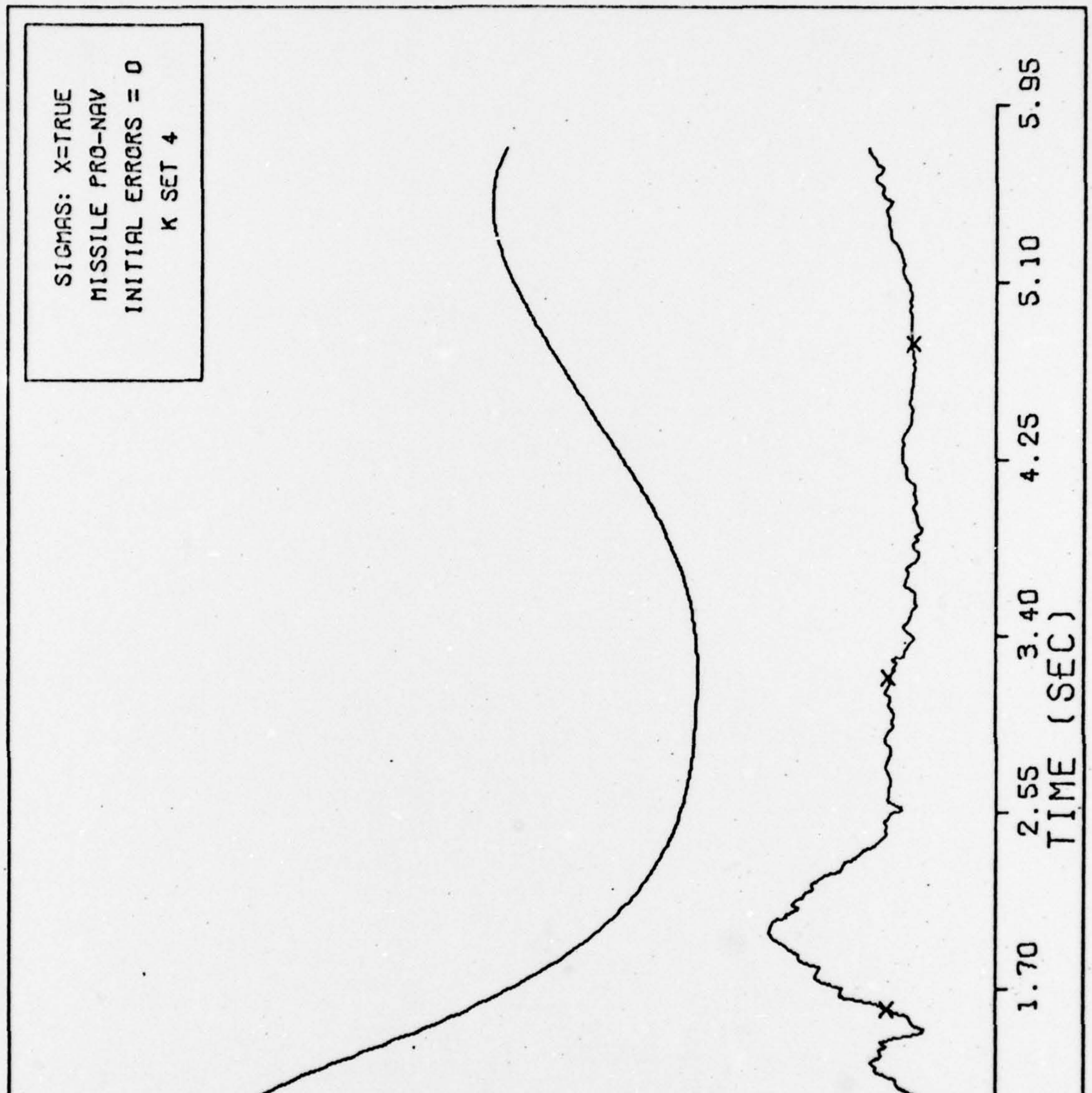



Fig. E-80 PRO-NAV FILTER & TRUE SIGMAS, 11 STATE FILTER

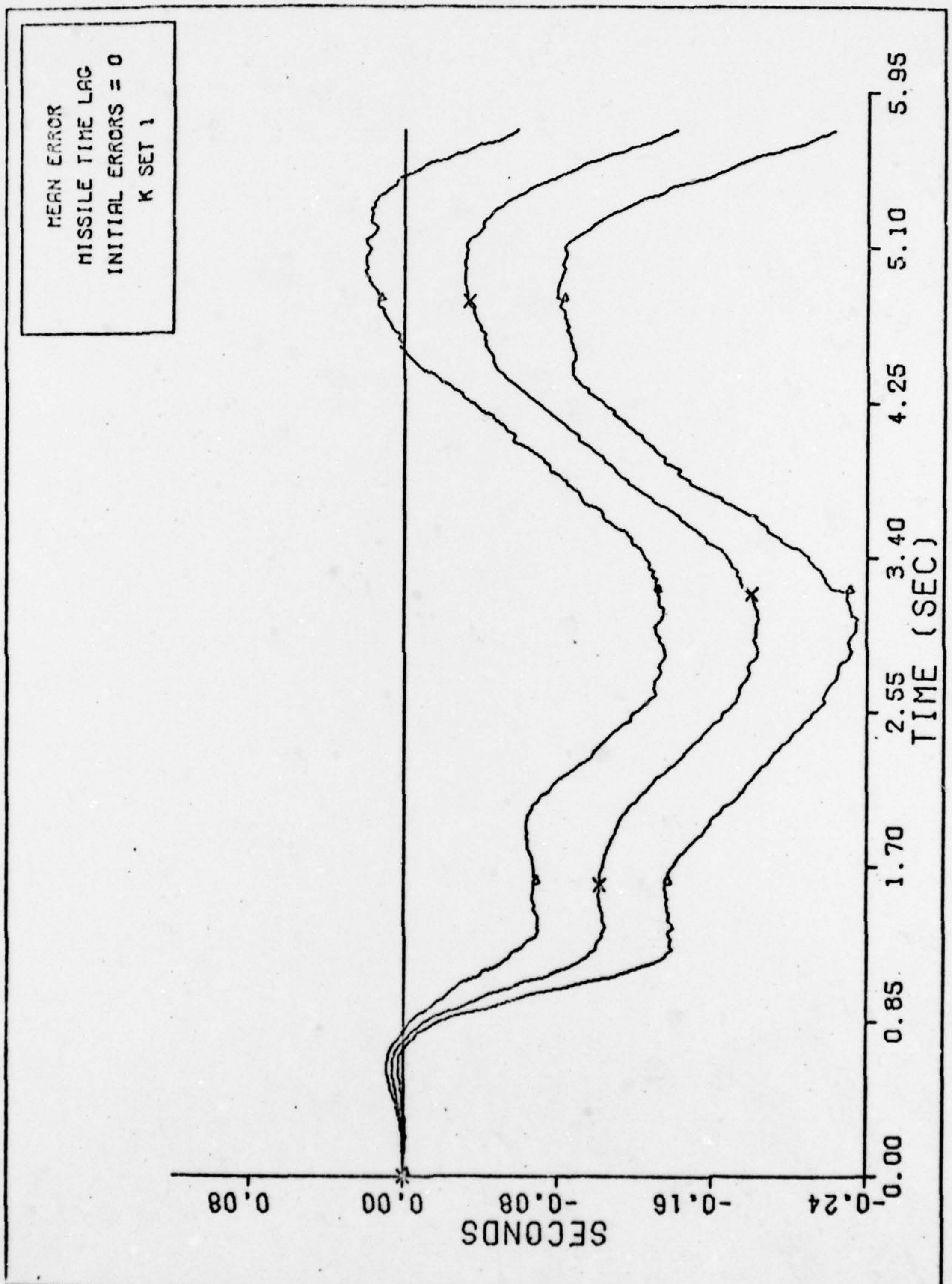



Fig. E-81 MISSILE TIME LAG MEAN ERROR, 11 STATE FILTER

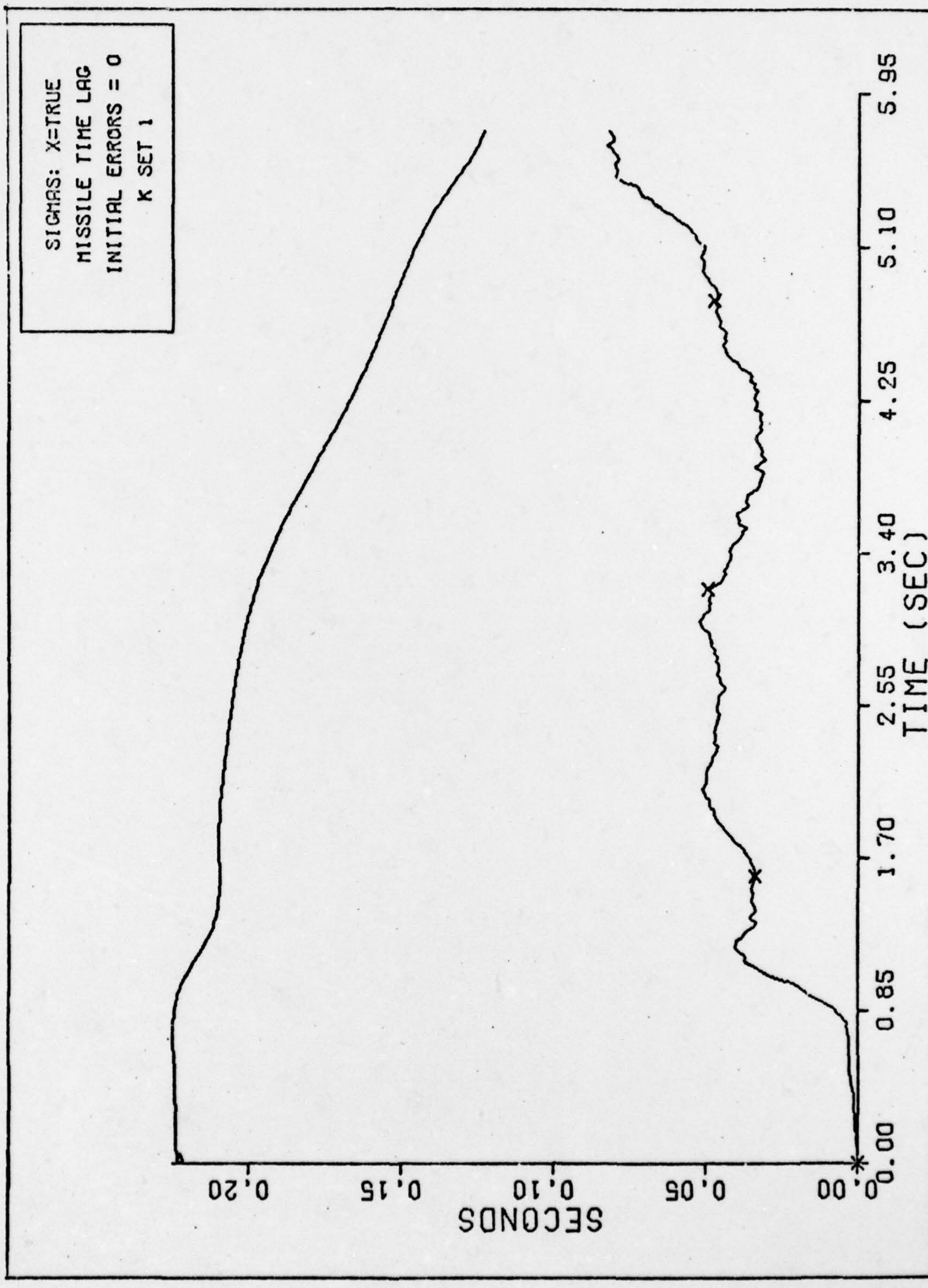



Fig. E-82 TIME LAG FILTER & TRUE SIGMAS, 11 STATE FILTER

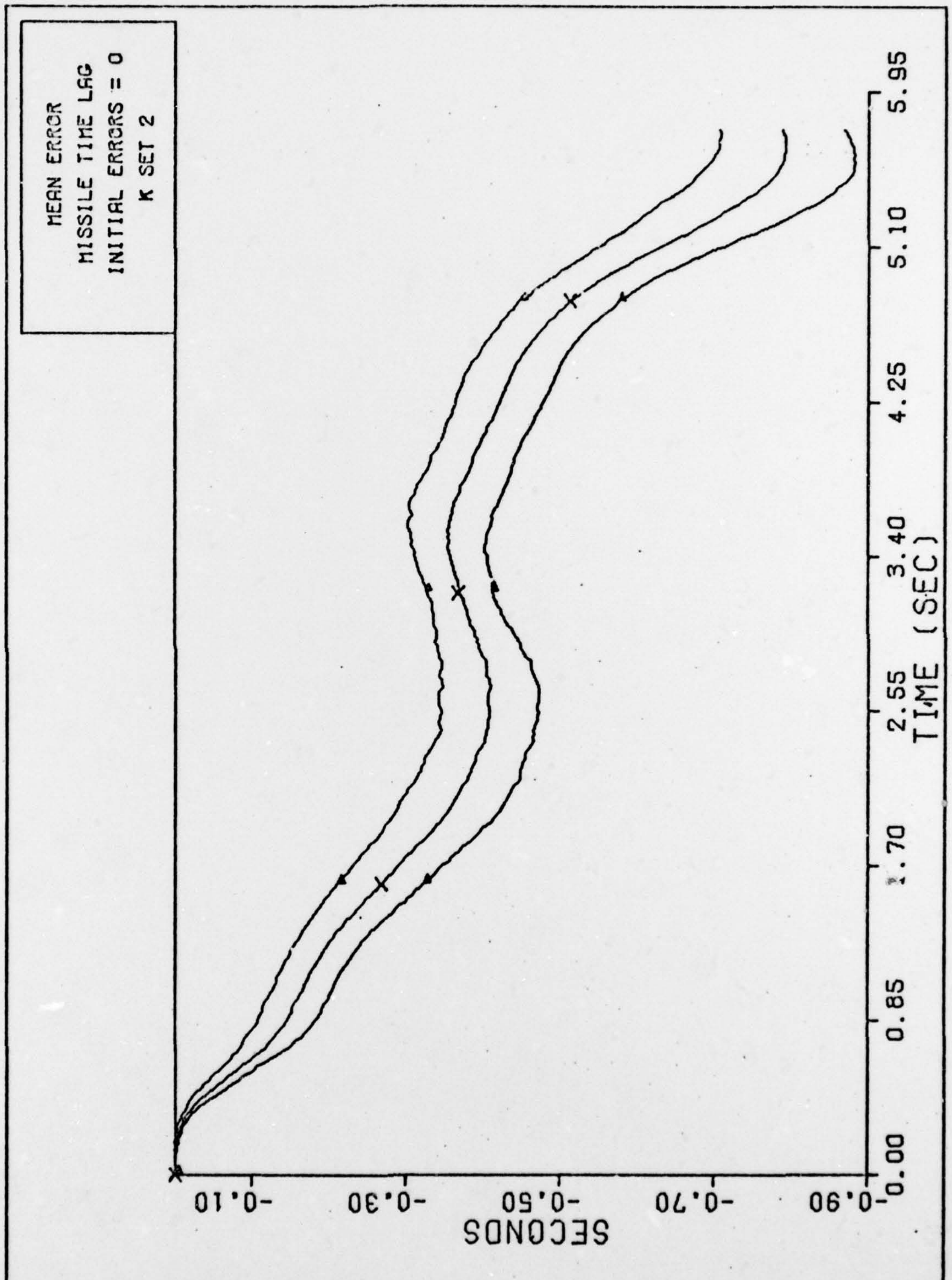



Fig. E-83 MISSILE TIME LAG MEAN ERROR, 11 STATE FILTER

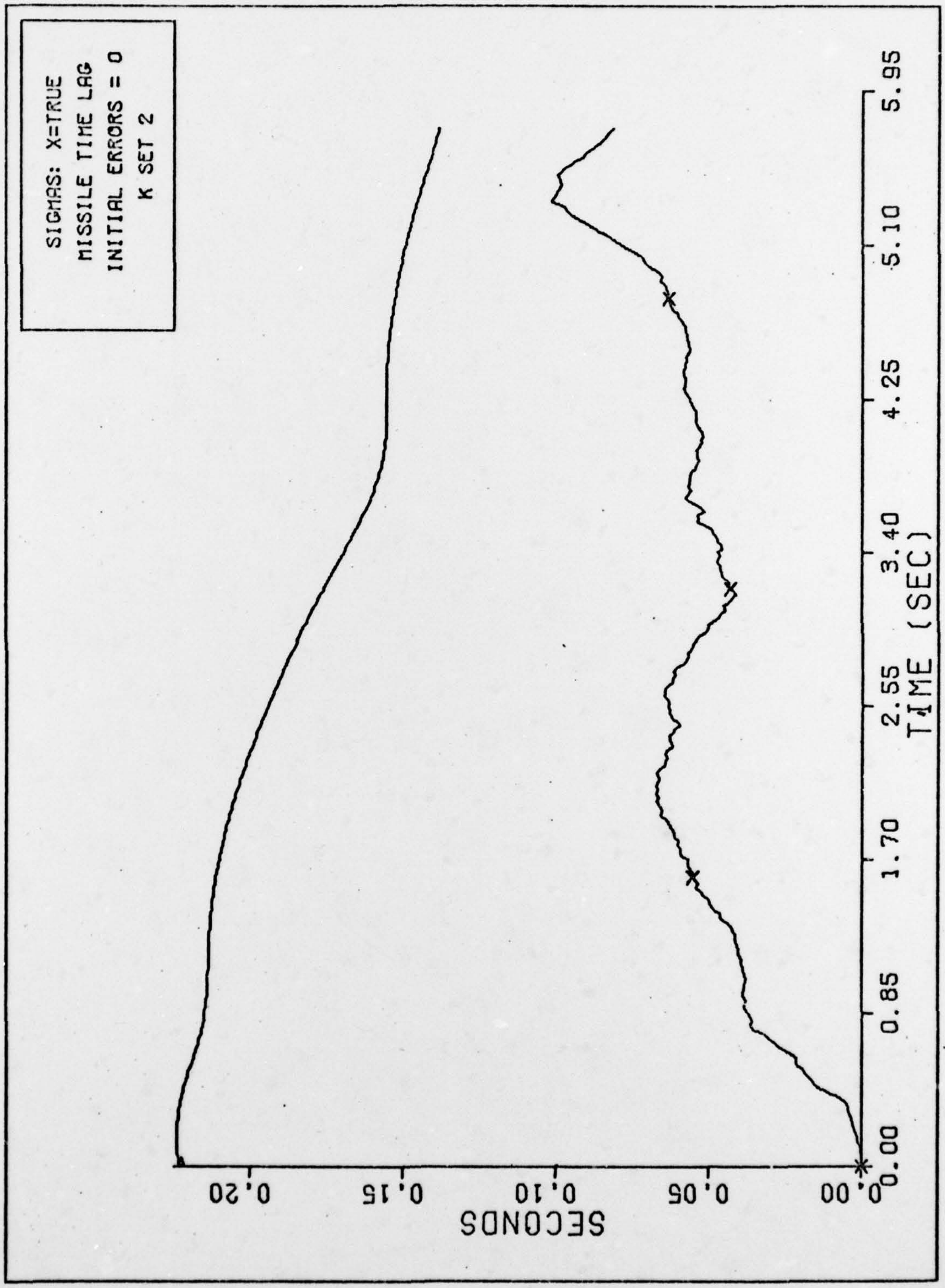



Fig. E-84 TIME LAG FILTER & TRUE SIGMAS, 11 STATE FILTER

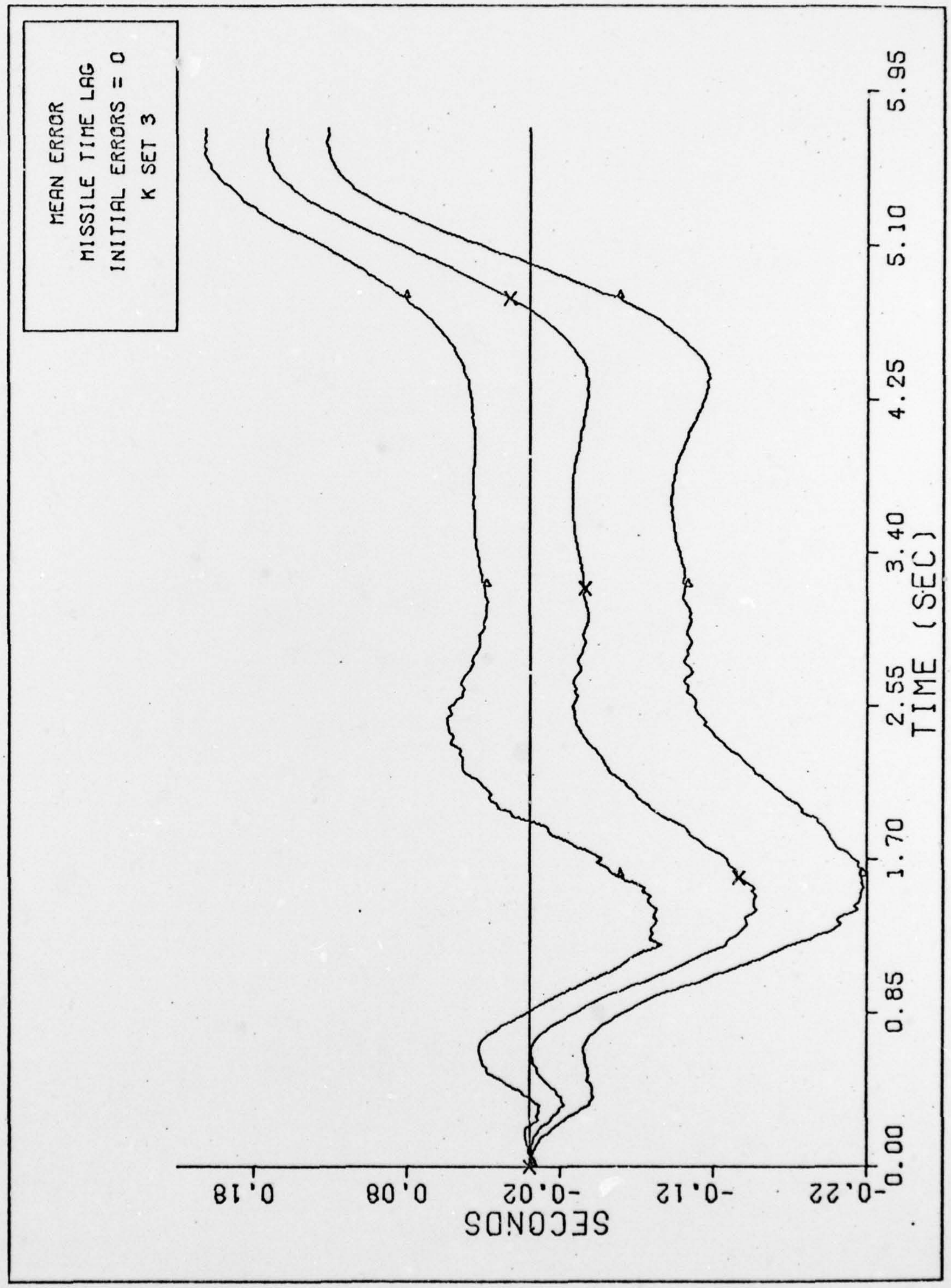



Fig. E-85 MISSILE TIME LAG MEAN ERROR, 11 STATE FILTER

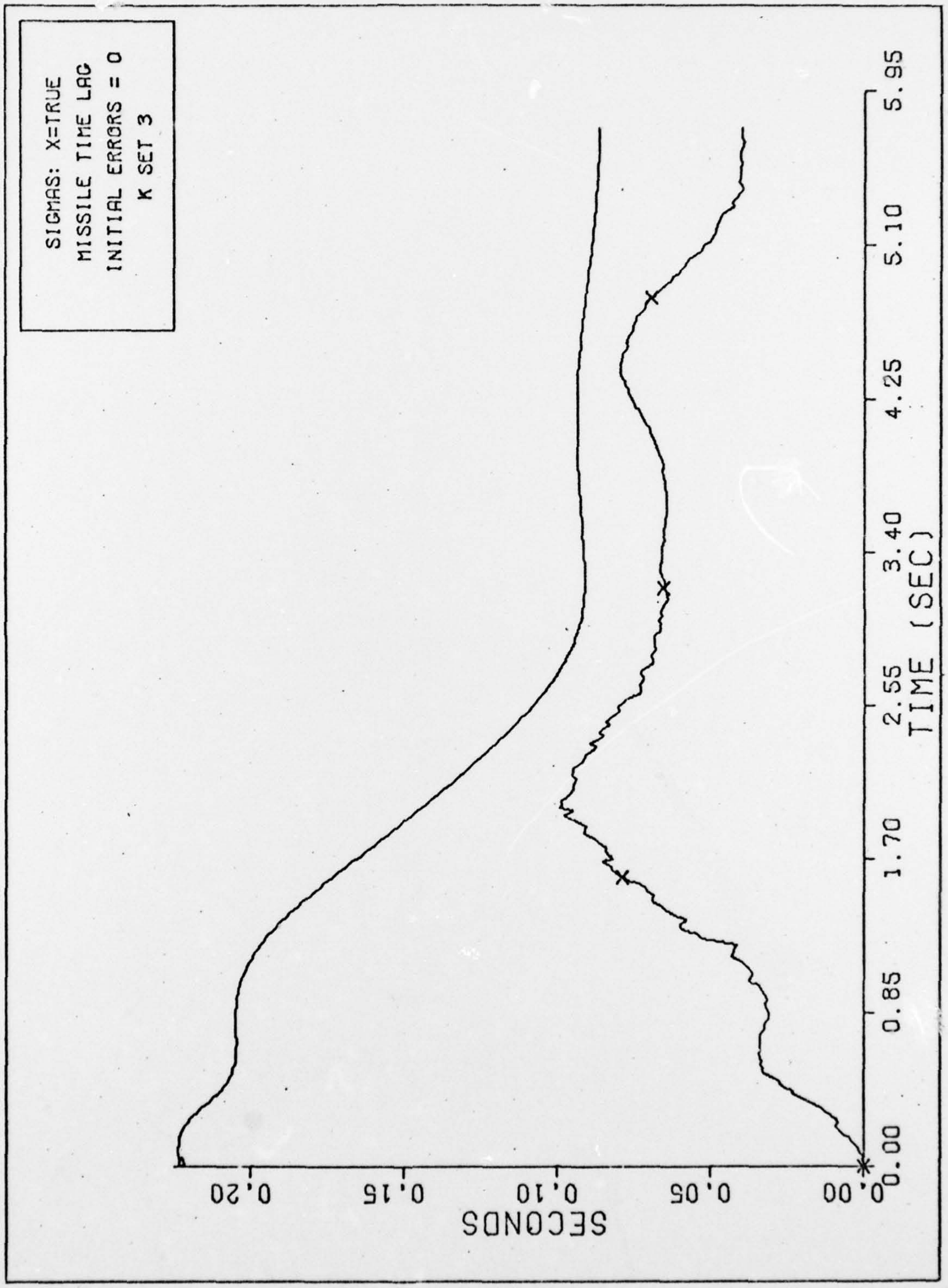



Fig. E-86 TIME LAG FILTER & TRUE SIGMAS, 11 STATE FILTER

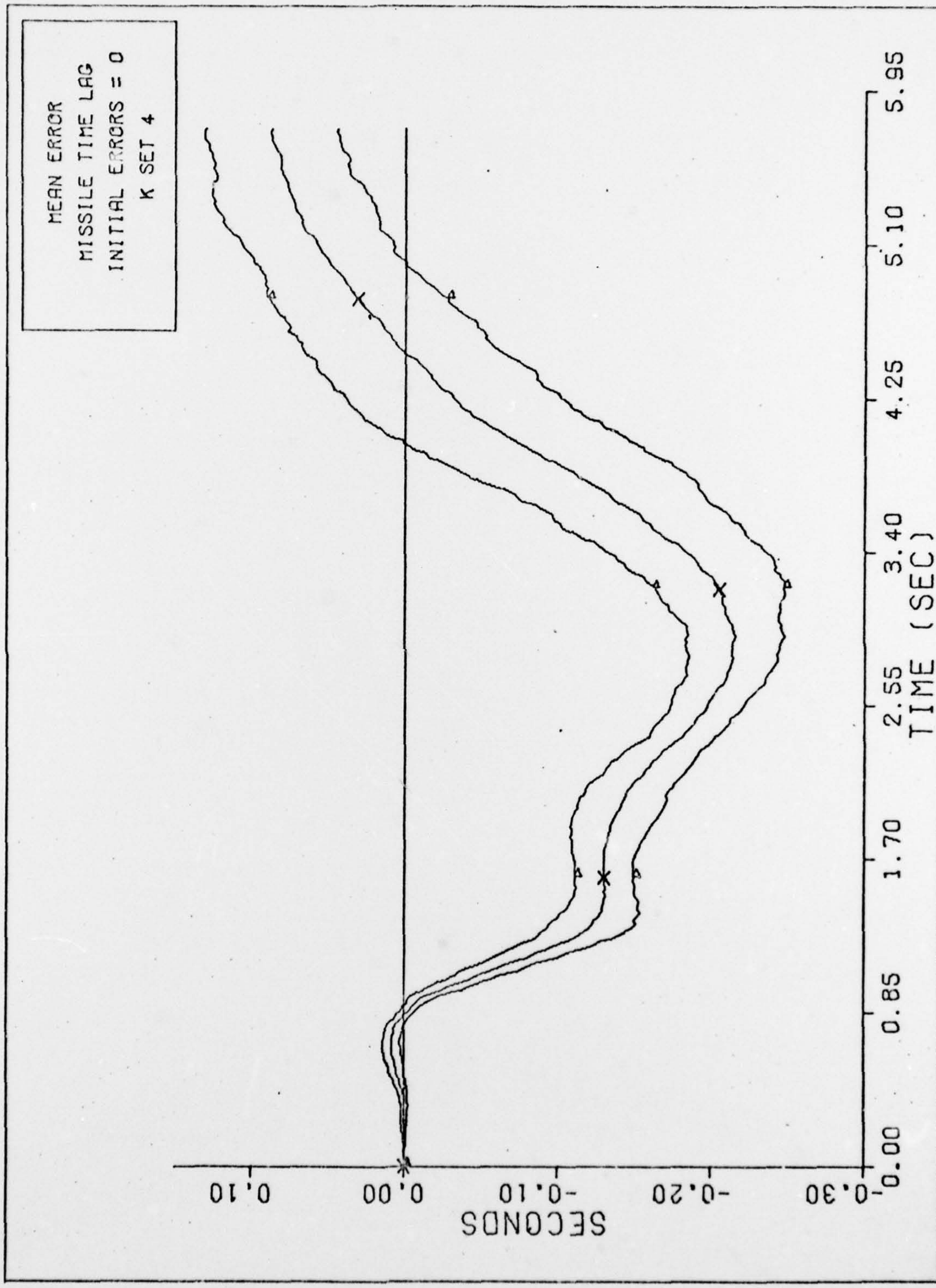



Fig. E-87 MISSILE TIME LAG MEAN ERROR, 11 STATE FILTER

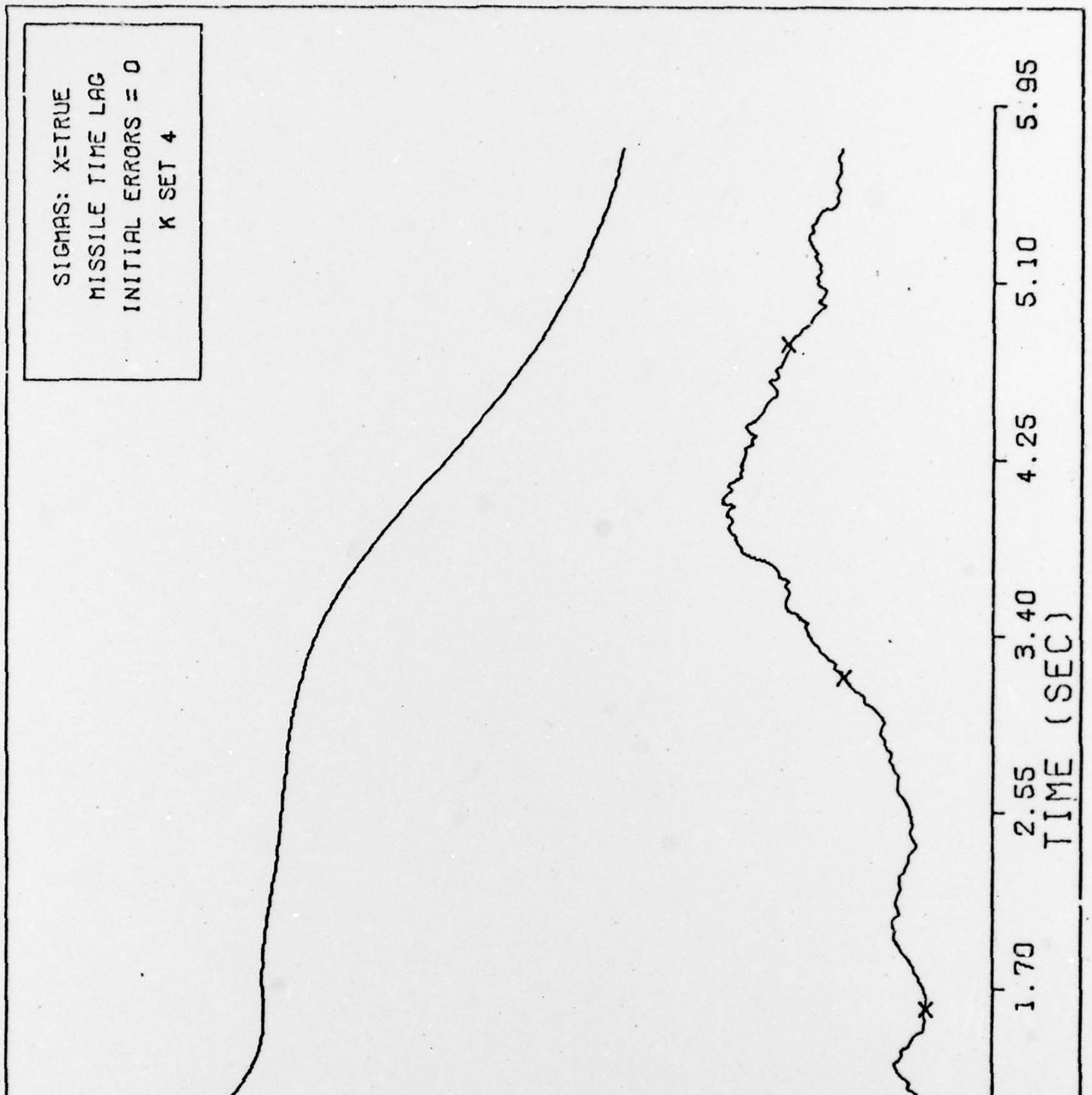



Fig. E-88 TIME LAG FILTER & TRUE SIGMAS, 11 STATE FILTER

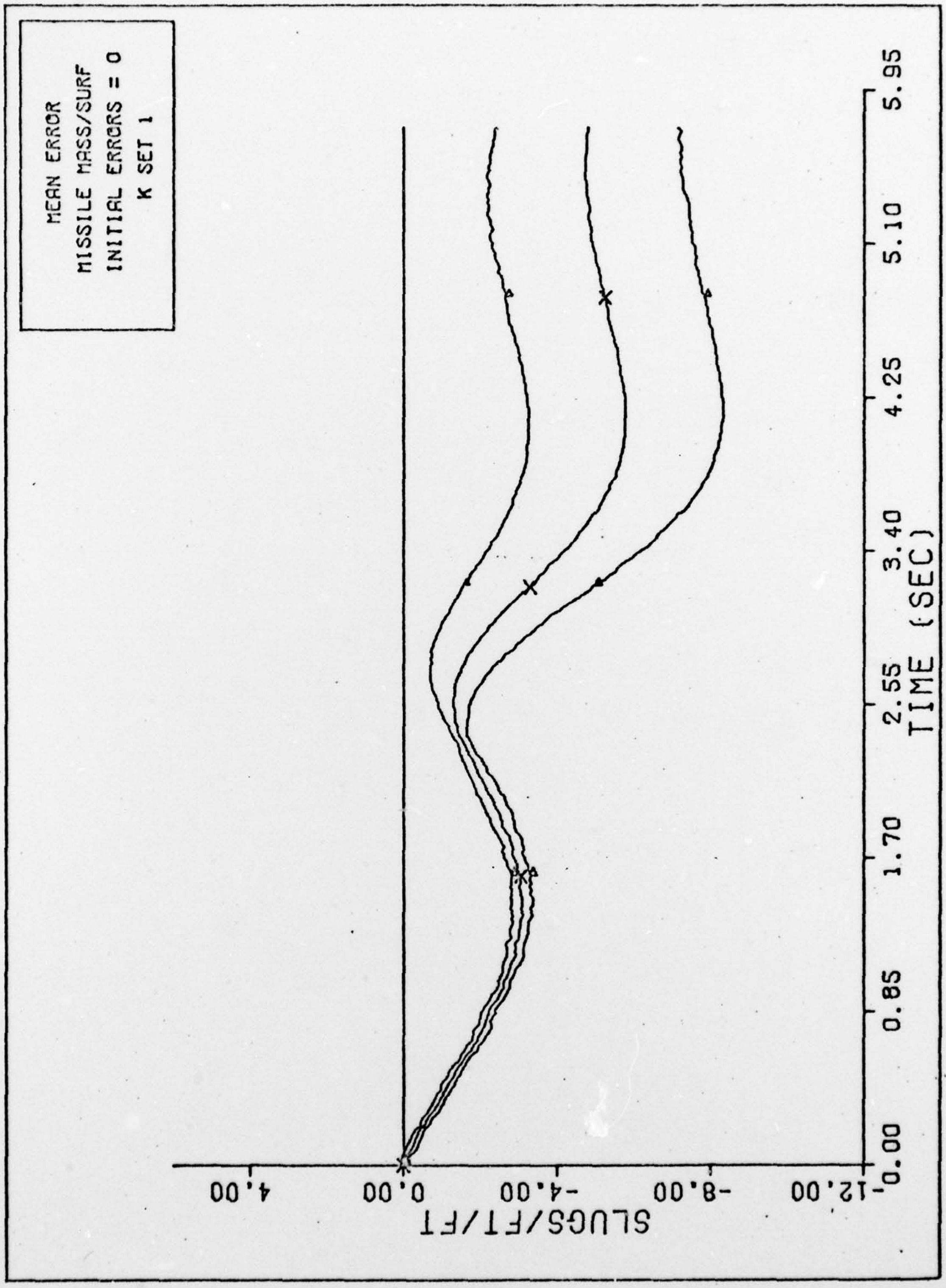



Fig. E-89 MISSILE MASS/SURF MEAN ERROR, 11 STATE FILTER

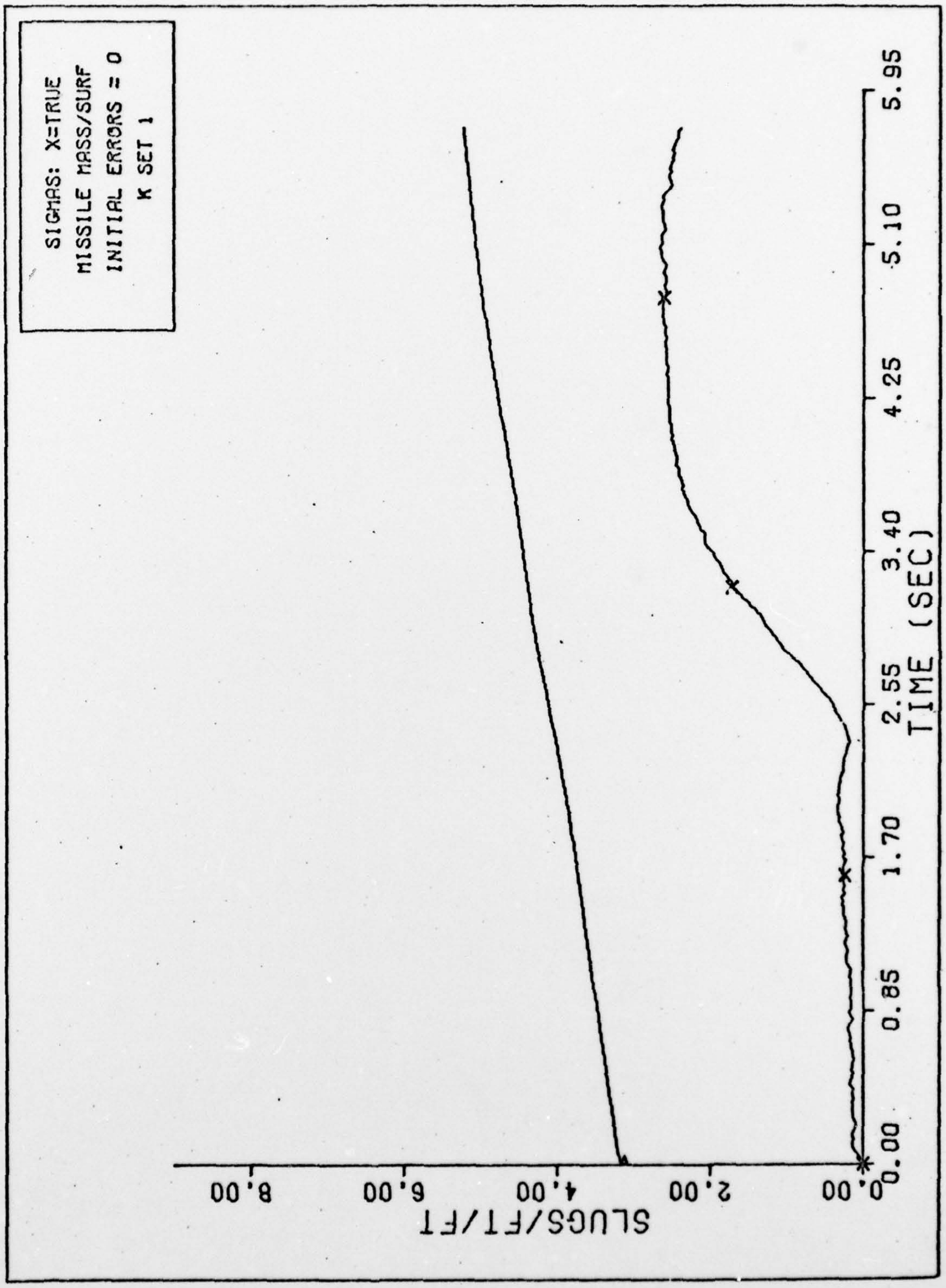



Fig. E-90 MASS/SURF FILTER & TRUE SIGMAS, 11 STATE FILTER

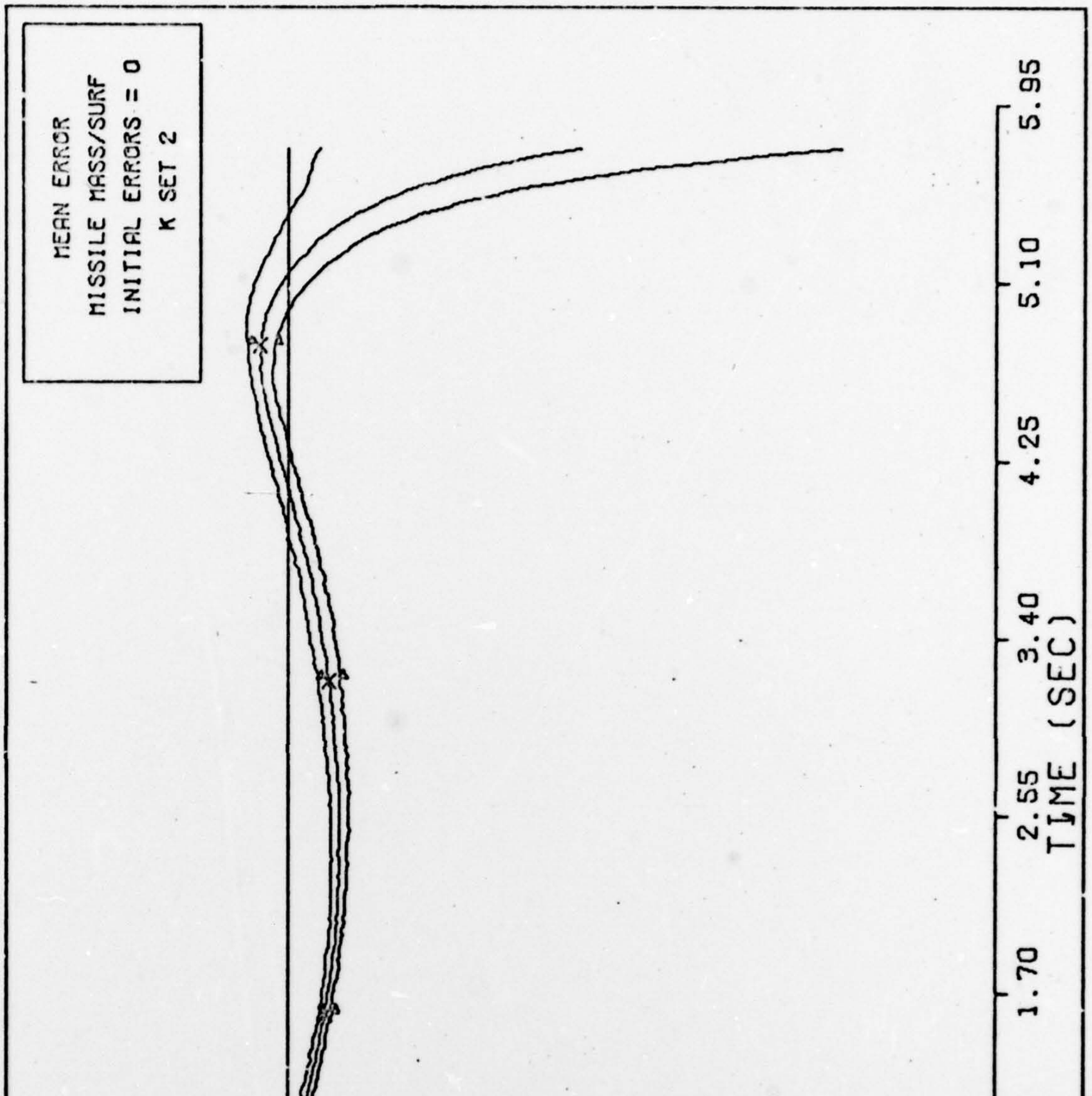



Fig. E-91 MISSILE MASS/SURF MEAN ERROR, 11 STATE FILTER

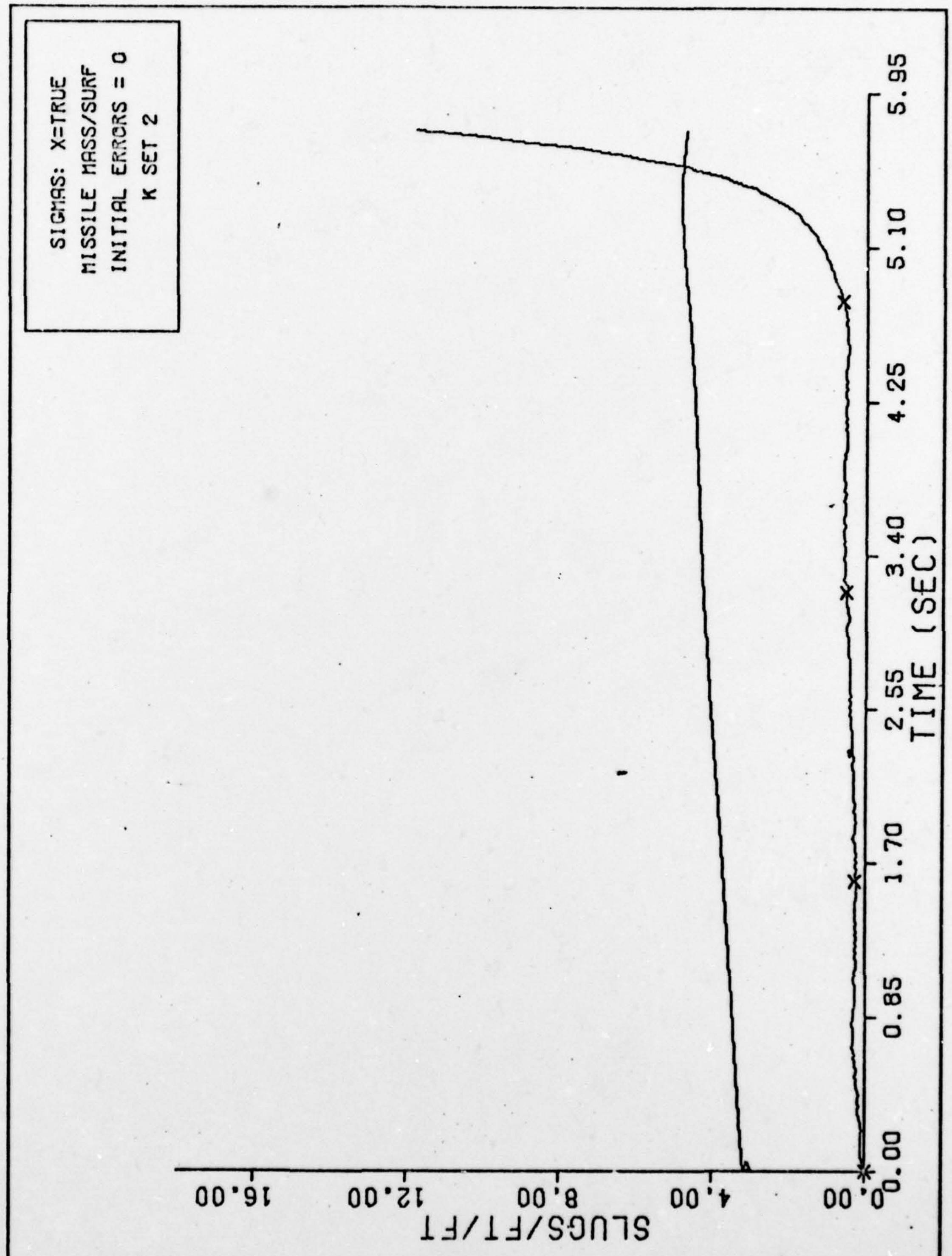



Fig. E-92 MASS/SURF FILTER & TRUE SIGMAS, 11 STATE FILTER

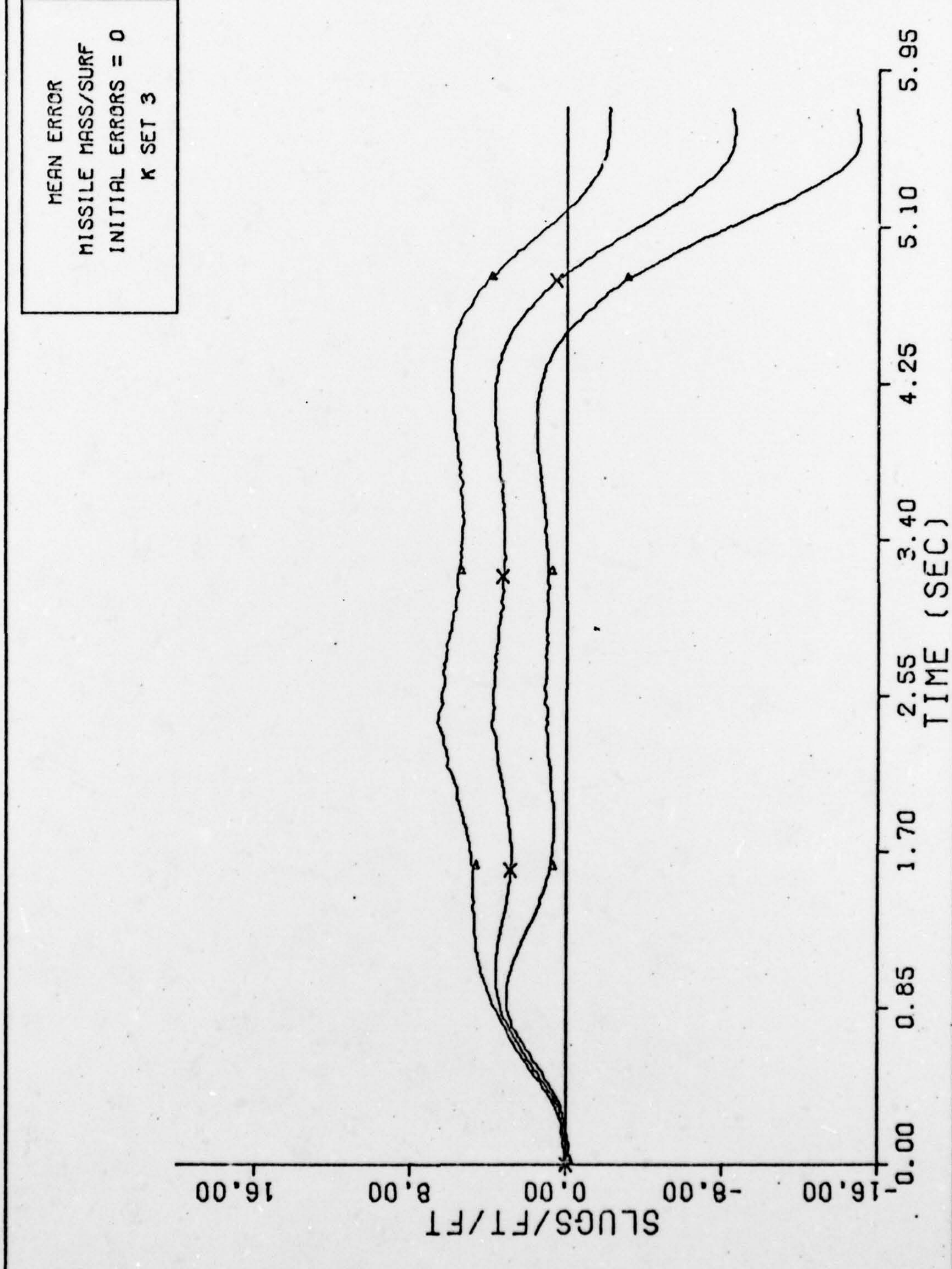



Fig. E-93 MISSILE MASS/SURF MEAN ERROR, 11 STATE FILTER

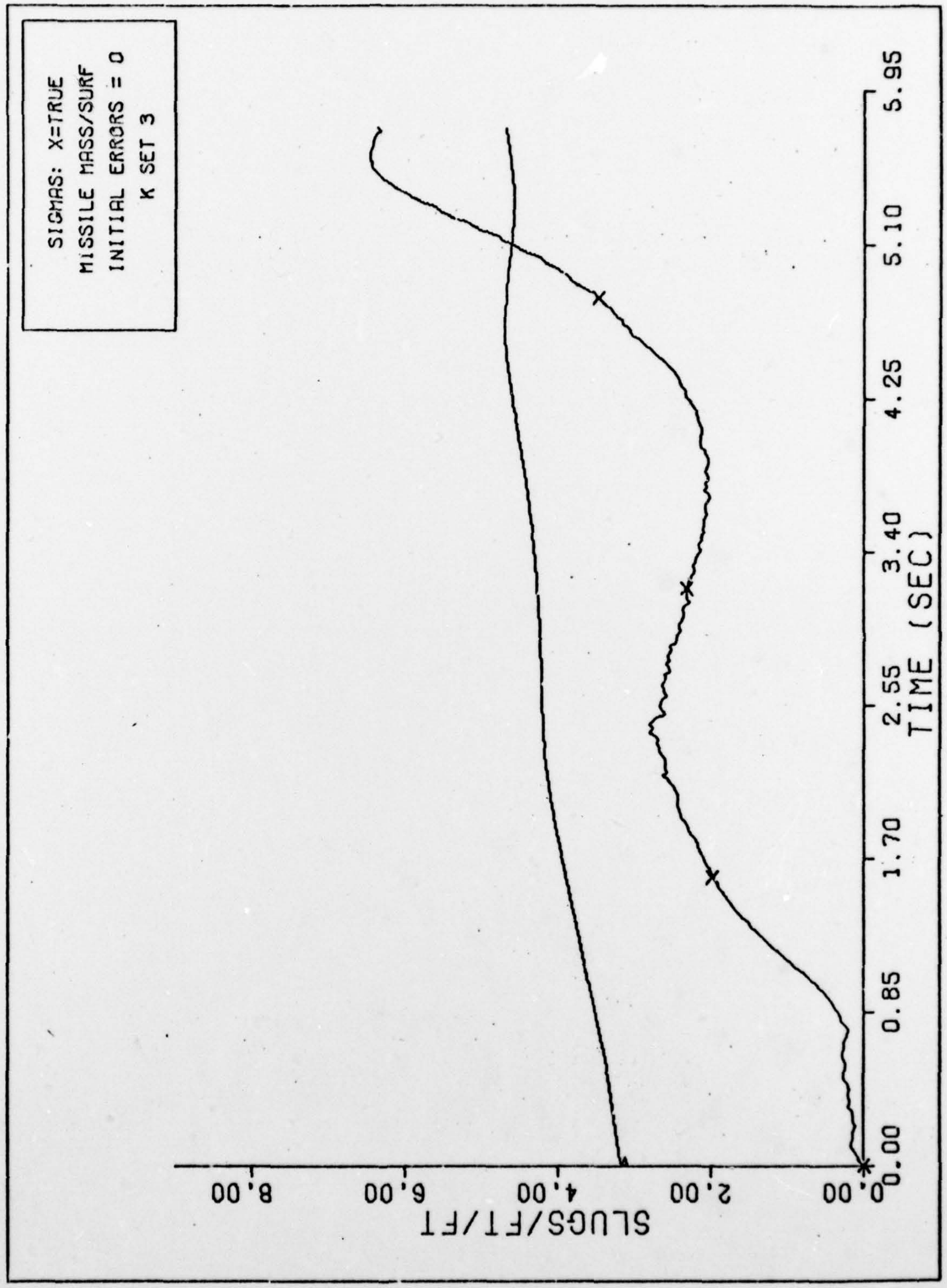



Fig. E-94 MASS/SURF FILTER & TRUE SIGMAS. 11 STATE FILTER

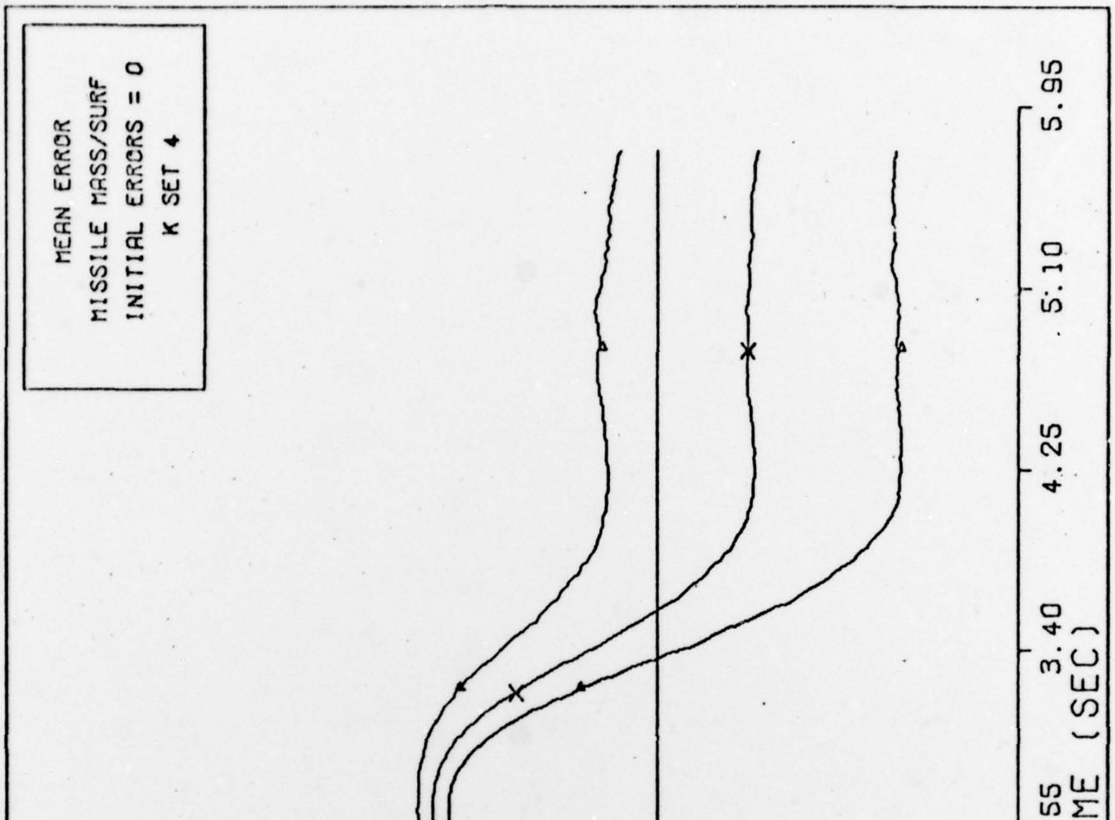



Fig. E-95 MISSILE MASS/SURF MEAN ERROR, 11 STATE FILTER

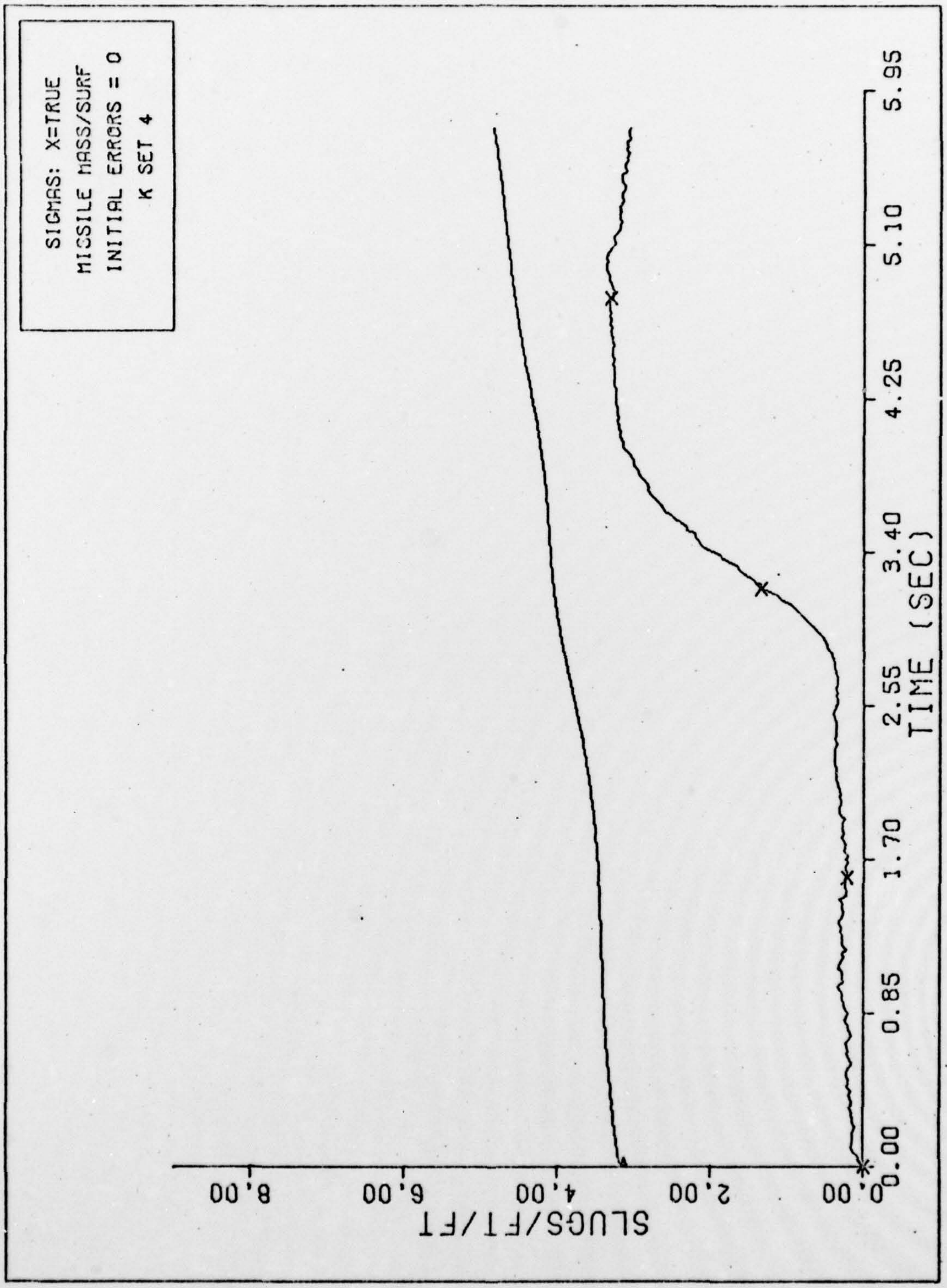



Fig. E-96 MASS/SURF FILTER & TRUE SIGMAS, 11 STATE FILTER