AD-A089 579 MARK RESOURCES INC MARINA DEL REY CA F/G 14/2
EXTENDED TARGET SIMULATION PROGRAM. REVISION II.(U)
JUN 78 R L MITCHELL DAAK40-78-C-0031
UNCLASSIFIED MRI-149-24-REV-2 NL

1 of 1
AD A
089579

END
DATE
FILMED
10-80
DTIC

RFSS

EXTENDED TARGET SIMULATION PROGRAM - REVISION II

TECH NOTE 105-046

20 JUN 78

DAAK40-78-C-0031

PREPARED FOR: RF SYSTEMS BRANCH (DRDMI-TDR)
SYSTEMS SIMULATION DIRECTORATE
TECHNOLOGY LABORATORY
US ARMY MIRADCOM
REDSTONE ARSENAL, AL 35809

PREPARED BY: DR. R. L. MITCHELL
MARK RESOURCES, INC.
4676 ADMIRALTY WAY
SUITE 303
MARINA DEL REY, CA 90291

MARK RESOURCES INCORPORATED

EXTENDED TARGET SIMULATION PROGRAM

REVISION II

by R. L. Mitchell

20 June 78

MRI — 149 — 24

On 16 June, discussions were held at MIRADCOM with Dwight McPherson (Boeing) and Robert Smith (NWC/China Lake). As a result of these discussions, two revisions have been made to the extended target simulation program:

(1) The target aspect angle is measured from the roll axis, not in the wing plane. This change was implemented in ETGEO by redefining a new angle (ANGL), and in the argument list in SCTAMP.

(2) In order to introduce some "randomness" from run to run, a bias angle (ST) is added to the target aspect angle in SCTAMP. It was suggested by R. Smith that this angle be a random variable, uniform over ±2°, that is chosen at the beginning of each run.

The changes made since 5 May 78 (the version described in MRI Report 149-21) are indicated by a double $$ beginning in column 73. Only MAIN, ETGEO, and SCTAMP are affected.

4676 ADMIRALTY WAY / SUITE 303 / MARINA DEL REY, CALIFORNIA 90291 / (213) 822-4955

```
      PROGRAM MAIN (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C
C THIS IS A SAMPLE MAIN PROGRAM FOR TEST PURPOSES ONLY.
C
C THIS EXTENDED-TARGET SIMULATION PACKAGE HAS BEEN PREPARED BY
C RL MITCHELL OF MARK RESOURCES, INC (213-822-4955), UNDER CONTRACT TO
C MIRADCOM.  IT IS WRITTEN WITH THE INTENTION THAT IT WILL BE MADE PART
C OF A REAL-TIME SIMULATION PROGRAM, ALTHOUGH SOME OF THE CODE IS NOT
C WRITTEN IN COMPLETELY OPTIMUM FORM (IT IS MORE EASILY UNDERSTOOD THIS
C WAY, AND THE REVISIONS ARE EASILY MADE).
C
C ALL ARRAYS IN COMMON SHOULD BE DIMENSIONED IN THE MAIN PROGRAM.
C
C SEE THE SUBROUTINES FOR A DEFINITION OF THE VARIABLES.
C
C RULES FOR DIMENSIONING ARRAYS.....
C
C       X,Y,Z................NSCAT
C       AMP,DA,DB,DC,DAD....NSCAT (MAYBE SMALLER)
C       TWARAY...............4*NARAY
C       VR,VI,DAZ,DEL.......NTAP
C
C THE CHANGES MADE ON 5 MAY 78 ARE INDICATED WITH A $ SIGN IN COLUMN 73
C THE CHANGES MADE SINCE 5 MAY 78 ARE INDICATED WITH A PAIR OF $$ SIGNS
C
      COMMON /T1/ XO,YO,ZO,XOD,YOD,ZOD,PSI,THETA,PHI,BP,BQ,BR           $
      COMMON /T2/ CPSI,SPSI,CTHETA,STHETA,CPHI,SPHI
      COMMON /T3/ XM,YM,ZM,XMD,YMD,ZMD
      COMMON /T4/ NSCAT,ST,AMPMIN,X(20),Y(20),Z(20)
      COMMON /T5/ NTAP,DRTAP,DRGATE,XL,PTGDSQ
      COMMON /T6/ NARAY,TWARAY(404)
      COMMON /T7/ NS,RO,R1,ROD,AMP(20),DA(20),DB(20),DC(20),DAD(20)
      COMMON /T8/ PEFF,VR(8),VI(8),DAZ(8),DEL(8)
C
C DEFINE VARIABLES.....
C
      DATA LR,LW/5,6/
      DATA NTAP/8/,DRTAP/30./,DRGATE/60./,XL/.02/,AMPMIN/1.E-10/,
     1     PTGDSQ/1./
      DATA DTIME/1./
      DATA NARAY/101/
      DATA ST/0./                                                      $$

      READ  (LR,100) XO,YO,ZO
      READ  (LR,100) XOD,YOD,ZOD
      READ  (LR,100) PSI,THETA,PHI
      READ  (LR,100) BP,BQ,BR                                           $
      READ  (LR,100) XM,YM,ZM
      READ  (LR,100) XMD,YMD,ZMD
      READ  (LR,101) NSCAT
```

```
      READ  (LR,100) (X(K),Y(K),Z(K),K=1,NSCAT)

      WRITE (LW,200) XO,YO,ZO
      WRITE (LW,201) XOD,YOD,ZOD
      WRITE (LW,202) PSI,THETA,PHI
      WRITE (LW,203) BP,BQ,BR
      WRITE (LW,204) XM,YM,ZM
      WRITE (LW,205) XMD,YMD,ZMD
      WRITE (LW,206) NSCAT
      WRITE (LW,207) (X(K),Y(K),Z(K),K=1,NSCAT)
C
C SUBROUTINES TO BE CALLED FROM MAIN OR DRIVER PROGRAM.....
C
      CALL DATAIN

      CALL TAPSET

      CALL ETQEO

      CALL ETQDM(DTIME)

      WRITE (LW,208) RO,R1,ROD
      WRITE (LW,209) (AMP(K),DA(K),DB(K),DC(K),DAD(K),K=1,NS)
      WRITE (LW,210) DTIME
      WRITE (LW,211) (VR(I),VI(I),DAZ(I),DEL(I),I=1,NTAP)
      WRITE (LW,212) PEFF

      STOP

  100 FORMAT(3F10.1)
  101 FORMAT(I5)

  200 FORMAT(//14H XO,YO,ZO...../(20X3F12.6))
  201 FORMAT(//17H XOD,YOD,ZOD...../(20X3F12.6))
  202 FORMAT(//19H PSI,THETA,PHI...../(20X3F12.6))
  203 FORMAT(//14H BP,BQ,BR...../(20X3F12.6))
  204 FORMAT(//14H XM,YM,ZM...../(20X3F12.6))
  205 FORMAT(//17H XMD,YMD,ZMD...../(20X3F12.6))
  206 FORMAT(//11H NSCAT...../20XI12)
  207 FORMAT(//11H X,Y,Z...../(20X3F12.6))
  208 FORMAT(//15H RO,R1,ROD...../(20X3F12.6))
  209 FORMAT(//22H AMP,DA,DB,DC,DAD...../(20X5F12.6))
  210 FORMAT(//11H DTIME...../20XF12.6)
  211 FORMAT(//19H VR,VI,DAZ,DEL...../(20X4F12.6))
  212 FORMAT(//10H PEFF...../20XE12.5)
      END
```

```
      SUBROUTINE ETGEO
C
C TRANSFORMATION TO RADAR SPACE FOR N-POINT SCATTERER MODEL
C
C IN THIS SUBROUTINE WE BEGIN WITH THE MODEL OF AN EXTENDED TARGET AND
C THE ENGAGEMENT GEOMETRY IN ORDER TO COMPUTE THE AMPLITUDE AND RADAR
C COORDINATES FOR EACH SCATTERER IN THE MODEL.
C
C THE MODEL IMPLEMENTED IS THE SO-CALLED MEDIUM-RANGE MODEL (SEE MRI
C REPORT 132-44).
C
C ASSUMPTIONS AND LIMITATIONS.....
C
C           1. ALL SCATTERERS ASSUMED TO BE ILLUMINATED BY SAME TRANSMIT
C              ANTENNA GAIN.
C           2. TARGET ASSUMED TO BE WITHIN LINEAR REGION OF MONOPULSE
C              RECEIVE BEAM.
C           3. THE DOPPLER SHIFT OF THE TARGET CG IS IMPLEMENTED BY MEANS
C              OF A FINELY-CONTROLLABLE DELAY LINE (THE LASER DEVICE),
C              PLUS THE USE OF THE FREQUENCY SYNTHESIZER.
C           4. ONLY ONE PHYSICAL TARGET IS SIMULATED PER CALL.
C
C ALL COMMUNICATION TO AND FROM THIS SUBROUTINE IS THRU COMMON.
C
C ON INPUT.....
C
C       /T1/    XO,YO,ZO          = TARGET CG IN INERTIAL COORDINATES
C               XOD,YOD,ZOD       = TARGET CG RATE IN INERTIAL COORDINATES
C               PSI,THETA,PHI     = YAW, PITCH, ROLL ANGLES
C               BP,BQ,BR          = YAW, PITCH, ROLL ANGLE BODY RATES
C
C       /T3/    XM,YM,ZM          = MISSILE CG IN INERTIAL COORDINATES
C               XMD,YMD,ZMD       = MISSILE CG RATE IN INERTIAL COORDINATES
C
C       /T4/    NSCAT = NUMBER OF SCATTERERS IN TARGET MODEL
C               ST    = APPROXIMATE PHYSICAL SIZE OF TARGET
C               AMPMIN = AMPLITUDE THRESHOLD FOR SCATTERERS
C               X,Y,Z = ARRAYS CONTAINING SCATTERER LOCATIONS IN TARGET
C                       COORDINATES
C
C       /T5/    NTAP    = NUMBER OF TAPS IN TAPPED DELAY LINE
C               DRTAP   = SPACING BETWEEN TAPS (2-WAY RANGE)
C
C ON OUTPUT.....
C
C       /T2/    CPSI,SPSI,...ETC = SINES AND COSINES OF TARGET ANGLES
C
C       /T7/    NS      = NUMBER OF SCATTERERS VISIBLE
C               RO      = RANGE TO TARGET CG
C               R1      = RANGE TO FIRST TAP
```

```
C                 ROD    = RANGE RATE OF TARGET CG
C                 AMP(J) = AMPLITUDE OF J-TH SCATTERER
C                 DA(J)  = INCREMENTAL A-VECTOR OF J-TH SCATTERER
C                 DB(J)  = INCREMENTAL B-VECTOR OF J-TH SCATTERER
C                 DC(J)  = INCREMENTAL C-VECTOR OF J-TH SCATTERER
C                 DAD(J) = INCREMENTAL A-VECTOR RATE OF J-TH SCATTERER
C
C THE TARGET CG AND MISSILE CG COORDINATES ARE IN AN INERTIAL COORDINATE
C SYSTEM REFERENCED TO THE GROUND (XY-PLANE PARALLEL TO GROUND, Z- DOWN)
C
C THE ABC-VECTORS ARE DEFINED AS.....
C
C       A - FROM THE TARGET TO THE MISSILE
C       B - PARALLEL TO THE GROUND, TO THE LEFT AS VIEWED FROM MISSILE
C       C - PERPENDICULAR TO A AND B IN RIGHT-HAND COORDINATE SYSTEM
C
C THE TARGET COORDINATES ARE.....
C
C       X - TARGET LONGITUDINAL AXIS, POSITIVE IN DIRECTION OF NOSE
C       Y - IN DIRECTION OF RIGHT WING
C       Z - DOWN
C
C THE BODY RATES ARE DEFINED AS.....                                  $
C                                                                     $
C      BP - CW ROTATION RATE ABOUT TARGET X-AXIS                      $
C      BQ - CW ROTATION RATE ABOUT TARGET Y-AXIS                      $
C      RQ - CW ROTATION RATE ABOUT TARGET Z-AXIS                      $
C                                                                     $
C THE DIRECTION OF ROTATION IS DEFINED LOOKING OUT FROM THE COORDINATE $
C ORIGIN.                                                             $
C                                                                     $
C SEE SUBROUTINE XFORM FOR A DEFINITION OF THE YAW, PITCH, AND ROLL
C ANGLES.
C
C THE RFSS CHAMBER COORDINATES ARE ASSUMED TO BE PARALLEL TO THE ABC-
C VECTORS.  RANGE IS IN -A DIRECTION, RIGHT AZIMUTH IN -B DIRECTION, AND
C UP ELEVATION IN -C DIRECTION.
C
C ALL DISTANCES (INCLUDING WAVELENGTH) MUST BE IN THE SAME UNITS.  ALL
C ANGLES MUST BE IN RADIANS.
C
      DIMENSION A(3),B(3),C(3),WA(3),WB(3),WC(3)
      COMMON /T1/ XO,YO,ZO,XOD,YOD,ZOD,PSI,THETA,PHI,BP,BQ,BR         $
      COMMON /T2/ CPSI,SPSI,CTHETA,STHETA,CPHI,SPHI
      COMMON /T3/ XM,YM,ZM,XMD,YMD,ZMD
      COMMON /T4/ NSCAT,ST,AMPMIN,X(20),Y(20),Z(20)
      COMMON /T5/ NTAP,DRTAP
      COMMON /T7/ NS,RO,R1,ROD,AMP(20),DA(20),DB(20),DC(20),DAD(20)
C
C COMPUTE SINES AND COSINES OF ANGLES
```

```
C
      CALL SINCOS(PSI,SPSI,CPSI)
      CALL SINCOS(THETA,STHETA,CTHETA)
      CALL SINCOS(PHI,SPHI,CPHI)
C
C COMPUTE RANGE TO TARGET CG AND A-VECTOR
C
      A(1)=XM-XO
      A(2)=YM-YO
      A(3)=ZM-ZO
      RO=SQRT(A(1)**2+A(2)**2+A(3)**2)
      A(1)=A(1)/RO
      A(2)=A(2)/RO
      A(3)=A(3)/RO
C
C COMPUTE RANGE TO FIRST TAP
C
      R1=RO-.5*(NTAP-1)*DRTAP
C
C COMPUTE RANGE RATE OF TARGET CG
C
      ROD=A(1)*(XOD-XMD)+A(2)*(YOD-YMD)+A(3)*(ZOD-ZMD)
C
C COMPUTE B- AND C-VECTORS
C
      RHO=SQRT(A(1)**2+A(2)**2)
      B(1)=-A(2)/RHO
      B(2)= A(1)/RHO
      B(3)=0.
      C(1)=-A(3)*B(2)
      C(2)= A(3)*B(1)
      C(3)=RHO
C
C TRANSFORM A-, B-, AND C-VECTORS TO TARGET COORDINATES
C
      CALL XFORM(A,WA)
      CALL XFORM(B,WB)
      CALL XFORM(C,WC)
C
C COMPUTE TARGET ASPECT ANGLE (ALPHA=AZIMUTH, BETA=ELEVATION,          $$
C                                ANGL=ANGLE TO ROLL AXIS)              $$
C
C     ALPHA=ATAN2(WA(2),WA(1))                                         $$
C     SBETA=-WA(3)                                                     $$
C     BETA=ATAN2(SBETA,SQRT(1.-SBETA**2))
      ANGL=ATAN2(SQRT(1.-WA(1)**2),WA(1))                              $$
C
C LOOP OVER SCATTERERS
C
      L=1
```

```
      DO 20 K=1,NSCAT
      SAMP=SCTAMP(K,ANGL)                                                    $$
      IF(SAMP.LE.AMPMIN) GO TO 20
      AMP(L)=SAMP
C
C COMPUTE INCREMENTAL A,B,C COORDINATE
C
      DA(L)=X(K)*WA(1)+Y(K)*WA(2)+Z(K)*WA(3)
      DB(L)=X(K)*WB(1)+Y(K)*WB(2)+Z(K)*WB(3)
      DC(L)=X(K)*WC(1)+Y(K)*WC(2)+Z(K)*WC(3)
C
C COMPUTE INCREMENTAL A-VECTOR RATE (SMALL ANGLES ARE ASSUMED)
C
      XKD= Z(K)*BQ-Y(K)*BR                                                   $
      YKD=-Z(K)*BP+X(K)*BR                                                   $
      ZKD= Y(K)*BP-X(K)*BQ                                                   $
      DAD(L)=XKD*WA(1)+YKD*WA(2)+ZKD*WA(3)
      L=L+1
   20 CONTINUE
      NS=L-1
      RETURN
      END
```

```
      SUBROUTINE ETGDM(DTIME)
C
C GLINT AND DOPPLER MODULATION FOR N-POINT SCATTER MODEL
C
C IN THIS SUBROUTINE WE COMPUTE THE GLINT OFFSETS AND MODULATION SIGNALS
C APPLIED TO EACH TAP OF THE TAPPED-DELAY LINE.  IT IS TO BE CALLED
C AFTER ETGEO TRANSFORMS COORDINATES TO RADAR SPACE.  IT WILL USUALLY
C BE CALLED MORE FREQUENTLY THAN ETGEO.  IT IS ALSO THE BEST SUBROUTINE
C TO PLACE IN THE AP120B.
C
C EXCEPT FOR TIME, ALL COMMUNICATION TO AND FROM THIS SUBROUTINE IS THRU
C COMMON.
C
C ON INPUT.....
C
C               DTIME  = TIME SINCE LAST UPDATE IN TARGEO
C
C       /T5/    NTAP   = NUMBER OF TAPS IN TAPPED DELAY LINE
C               DRTAP  = SPACING BETWEEN TAPS (2-WAY RANGE)
C               XL     = WAVELENGTH
C               PTGDSQ = PRODUCT OF TRANSMIT POWER, GAIN, AND SQUARE OF
C                        RFSS CHAMBER LENGTH
C
C       /T7/    NS     = NUMBER OF SCATTERERS VISIBLE
C               RO     = RANGE TO TARGET CG
C               R1     = RANGE TO FIRST TAP
C               AMP(J) = AMPLITUDE OF J-TH SCATTERER
C               DA(J)  = INCREMENTAL A-VECTOR OF J-TH SCATTERER
C               DB(J)  = INCREMENTAL B-VECTOR OF J-TH SCATTERER
C               DAD(J) = INCREMENTAL A-VECTOR RATE OF J-TH SCATTERER
C
C ON OUTPUT.....
C
C       /T8/    PEFF   = EFFECTIVE RADIATED POWER AT RFSS ARRAY
C               VR(I)  = IN-PHASE   MODULATION SIGNAL TO I-TH TAP
C               VI(I)  = QUADRATURE MODULATION SIGNAL TO I-TH TAP
C               DAZ(I) = GLINT OFFSET (AZIMUTH)   FOR I-TH TAP
C               DEL(I) = GLINT OFFSET (ELEVATION) FOR I-TH TAP
C
C THE PARAMETER PMIN IS JUST SOME SMALL NUMBER TO PREVENT DIVIDE BY ZERO
C
C ARRAYS VBR,VBI,VCR,VCI MUST BE DIMENSIONED AS LARGE AS NTAP
C
      DIMENSION VBR(8),VBI(8),VCR(8),VCI(8)
      DIMENSION SS(4),CC(4)
      DIMENSION TW(4)
      COMMON /T5/ NTAP,DRTAP,DRGATE,XL,PTGDSQ
      COMMON /T7/ NS,RO,R1,ROD,AMP(20),DA(20),DB(20),DC(20),DAD(20)
      COMMON /T8/ PEFF,VR(8),VI(8),DAZ(8),DEL(8)
      DATA PMIN/1.E-10/
```

```
      DATA FOURPI/12.5663706/
C
C ZERO ARRAYS
C
      CALL XMIT(-NTAP,O.,VR)
      CALL XMIT(-NTAP,O.,VI)
      CALL XMIT(-NTAP,O.,VBR)
      CALL XMIT(-NTAP,O.,VBI)
      CALL XMIT(-NTAP,O.,VCR)
      CALL XMIT(-NTAP,O.,VCI)
      CALL XMIT(-NTAP,O.,DAZ)
      CALL XMIT(-NTAP,O.,DEL)
C
C LOOP OVER NS SCATTERERS
C
      DO 40 J=1,NS
C
C COMPUTE TAP NUMBER OF FIRST TAP (ITAP) AND FRACTION (P)
C
      R=RO-(DA(J)+DAD(J)*DTIME)
      P=(R-R1)/DRTAP+100.
      ITAP=P
      P=P-ITAP
      ITAP=ITAP-100
C
C COMPUTE RANGE DIFFERENCE FROM TAP NUMBER ITAP
C
      DR=(P+1.)*DRTAP
C
C FIND TAP WEIGHTS
C
      CALL TAPWTS(P,TW)
C
C COMPUTE PHASE ON FOUR TAPS
C
      DO 20 I=1,4
      CALL SINCOS(-FOURPI*DR/XL,S,C)
      SS(I)=S*AMP(J)*TW(I)
      CC(I)=C*AMP(J)*TW(I)
      DR=DR-DRTAP
   20 CONTINUE
C
C LOOP OVER UP TO FOUR TAPS AND INCREMENT ARRAYS
C
      IF(ITAP.GT.NTAP) GO TO 40
      IF(ITAP.LT.-2)   GO TO 40
      I1=MAXO(ITAP,1)
      I2=MINO(ITAP+3,NTAP)
      II=I1-ITAP
      DO 30 I=I1,I2
```

```
      II=II+1
      VR (I)=VR (I)+CC(II)
      VI (I)=VI (I)+SS(II)
      VBR(I)=VBR(I)+CC(II)*DB(J)
      VBI(I)=VBI(I)+SS(II)*DB(J)
      VCR(I)=VCR(I)+CC(II)*DC(J)
      VCI(I)=VCI(I)+SS(II)*DC(J)
   30 CONTINUE
   40 CONTINUE
C
C COMPUTE GLINT OFFSETS FOR EACH TAP AND PEAK POWER
C
      PEAK=0.
      DO 50 I=1,NTAP
      POW=VR(I)**2+VI(I)**2
      IF(POW.GT.PEAK) PEAK=POW
      IF(POW.LT.PMIN) GO TO 50
      DAZ(I)=-(VBR(I)*VR(I)+VBI(I)*VI(I))/(RO*POW)
      DEL(I)=-(VCR(I)*VR(I)+VCI(I)*VI(I))/(RO*POW)
   50 CONTINUE
C
C NORMALIZE AMPLITUDE
C
      ANORM=SQRT(PEAK)
      DO 60 I=1,NTAP
      VR(I)=VR(I)/ANORM
      VI(I)=VI(I)/ANORM
   60 CONTINUE
C
C COMPUTE EFFECTIVE RF POWER
C
      PEFF=PEAK*PTGDSQ/(FOURPI*RO**4)

      RETURN
      END
```

```
      SUBROUTINE XFORM(A,W)
C
C IN THIS SUBROUTINE WE TRANSFORM A VECTOR (A) IN INERTIAL COORDINATES
C TO A VECTOR (W) IN TARGET COORDINATES.  THE COORDINATE ROTATIONS, IN
C THE ORDER OF APPLICATION, ARE.....
C
C       PSI   = CW ROTATION OF Z-AXIS (YAW)
C       THETA = CW ROTATION OF Y-AXIS (PITCH)
C       PHI   = CW ROTATION OF X-AXIS (ROLL)
C
C THE DIRECTION OF ROTATION IS DEFINED LOOKING INTO THE COORDINATE
C ORIGIN.  IN THIS SUBROUTINE THE SINES AND COSINES OF THE ANGLES ARE
C INPUT THRU COMMON /T2/.
C
      DIMENSION A(3),W(3)
      COMMON /T2/ CPSI,SPSI,CTHETA,STHETA,CPHI,SPHI
      UX=A(1)*CPSI-A(2)*SPSI
      UY=A(1)*SPSI+A(2)*CPSI
      UZ=A(3)
      VX= UX*CTHETA+UZ*STHETA
      VY= UY
      VZ=-UX*STHETA+UZ*CTHETA
      W(1)=VX
      W(2)=VY*CPHI-VZ*SPHI
      W(3)=VY*SPHI+VZ*CPHI
      RETURN
      END
```

```
      SUBROUTINE TAPWTS(P,TW)
C
C IN THIS SUBROUTINE FOUR TAP WEIGHTS ARE RETURNED IN ARRAY TW ACCORDING
C TO THE FRACTION P.  THE WEIGHTS ARE EXTRACTED FROM A PRECOMPUTED TABLE
C (SEE SUBROUTINE TAPSET).
C
C ARRAY TWARAY IS USED AS IF IT WERE DIMENSIONED (4,NARAY).
C
      DIMENSION TW(4)
      COMMON /T6/ NARAY,TWARAY(1)
      DATA LW/6/
      INDEX=(NARAY-1)*P+1.5
      CALL XMIT(4,TWARAY(4*INDEX-3),TW)
C                                            )6.21F5X02/.....WT,P H01(TAMROF (
C                                                     WT,P )002,WL( ETIRW
      RETURN
      END
```

```
      SUBROUTINE TAPSET
C
C IN THIS SUBROUTINE THE TAP WEIGHT TABLE IS COMPUTED.  IT IS A
C COMPANION SUBROUTINE TO TAPWTS, AND IT IS TO BE CALLED AS AN INITIAL-
C IZATION STEP PRIOR TO THE BEGINNING OF THE SIMULATED MISSION.
C
C       /T5/   DRTAP  = SPACING BETWEEN TAPS (2-WAY RANGE)
C              DRGATE = SPACING BETWEEN RECEIVER GATES (2-WAY RANGE)
C
C ARRAY TWARAY MUST BE DIMENSIONED AS LARGE AS 4*NARAY.
C
      DIMENSION A(4,4),X(4)
      COMMON /T5/ NTAP,DRTAP,DRGATE
      COMMON /T6/ NARAY,TWARAY(1)
      D=DRTAP/DRGATE
      L=1
      DO 30 K=1,NARAY
      P=(K-1)/FLOAT(NARAY-1)
      DO 10 J=1,4
      X(J)=CHI(D*(P+2-J))
   10 CONTINUE
      DO 20 I=1,4
      DO 20 J=1,4
      A(I,J)=CHI(D*(I-J))
   20 CONTINUE
      CALL SIMQ(A,X,4,IERR)
      IF(IERR.GT.0) STOP
      CALL XMIT(4,X,TWARAY(L))
      L=L+4
   30 CONTINUE
      RETURN
      END
```

```
      FUNCTION CHI(P)
C
C RANGE GATE RESPONSE.  THE ARGUMENT P IS THE RANGE MISMATCH NORMALIZED
C TO THE RECEIVER GATE SPACING.  INTERPOLATION IS USED ON THE SAMPLES
C STORED IN THE A-ARRAY, WHERE THE SPACING IS 0.1 UNIT.
C
C THE RESIDUAL ERROR IN THE INTERPOLATION IS LESS THAN .0003
C
C P MUST BE LESS THAN 1.5 IN MAGNITUDE.
C
C THE SAMPLES ARE OF THE RESPONSE DERIVED IN MRI REPORT 149-4.
C
      DIMENSION A(18)
      DATA A/1.00000, .98104, .92193, .81903, .67431, .50112, .32385,
     1        .17071, .06308, .00731,-.00651, .00182, .01262, .01458,
     2        .00713,-.00313,-.00898,-.00762 /
      H=10.*ABS(P)
      IF(H.GT.15.) STOP 55
      I=H
      H=H-I
      IP1=I+1
      IP2=I+2
      IP3=I+3
      IF(I.LE.0) I=2
      CHI=-.166667*H*(H-1.)*(H-2.)*A(I)+.5*(H**2-1.)*(H-2.)*A(IP1)
     1    -.5*H*(H+1.)*(H-2.)*A(IP2)+.166667*H*(H**2-1.)*A(IP3)
      RETURN
      END
```

```
      SUBROUTINE SIMQ(A,B,N,IERR)
C
C SOLVES SET OF N SIMULTANEOUS EQUATIONS.....
C
C                    A * X = B          SUM (A(I,J)*X(J)) = B(I)
C
C WHERE ARRAY A IS 2-DIMENSIONAL.  ARRAY X IS RETURNED IN ARRAY B, AND
C ARRAY A IS DESTROYED.  COMPUTATION IS VALID IF IERR=0
C
      DIMENSION A(1),B(1)
      IERR = 0
      IF (N.GT.0)   GO TO 10
      IERR = 1
      RETURN
C
C         FORWARD SOLUTION
C
   10 TOL = 0.0
      KS = 0
      JJ = -N
      DO 65 J = 1,N
      JY = J + 1
      JJ = JJ + N + 1
      BIGA = 0.
      IT = JJ - J
      DO 30 I = J,N
C
C         SEARCH FOR MAXIMUM COEFFICIENT IN COLUMN
C
      IJ = IT + I
      IF (ABS(BIGA)-ABS(A(IJ)))   20, 30, 30
   20 BIGA = A(IJ)
      IMAX = I
   30 CONTINUE
C
C         TEST FOR PIVOT LESS THAN TOLERANCE (SINGULAR MATRIX)
C
      IF (ABS(BIGA)-TOL)   35, 35, 40
   35 IERR = 2
      RETURN
C
C         INTERCHANGE ROWS IF NECESSARY
C
   40 I1 = J + N*(J-2)
      IT = IMAX - J
      DO 50 K = J,N
      I1 = I1 + N
      I2 = I1 + IT
      SAVE = A(I1)
      A(I1) = A(I2)
```

```
      A(I2) = SAVE
C
C          DIVIDE EQUATION BY LEADING COEFFICIENT
C
   50 A(I1) = A(I1)/BIGA
      SAVE = B(IMAX)
      B(IMAX) = B(J)
      B(J) = SAVE/BIGA
C
C          ELIMINATE NEXT VARIABLE
C
      IF (J-N)  55, 70, 55
   55 IQS = N*(J-1)
      DO 65 IX = JY,N
      IXJ = IQS + IX
      IT = J - IX
      DO 60 JX = JY,N
      IXJX = N*(JX-1) + IX
      JJX = IXJX + IT
   60 A(IXJX) = A(IXJX) - (A(IXJ)*A(JJX))
   65 B(IX) = B(IX) - (B(J)*A(IXJ))
C
C          BA SOLUTION
C
   70 NY = N - 1
      IT = N*N
      DO 80 J = 1,NY
      IA = IT - J
      IB = N - J
      IC = N
      DO 80 K = 1,J
      B(IB) = B(IB) - A(IA)*B(IC)
      IA = IA - N
   80 IC = IC - 1
      RETURN
      END
```

```
      SUBROUTINE XMIT(N,A,B)
C
C IN THIS SUBROUTINE WE EITHER TRANSMIT ARRAY A TO ARRAY B (IF N.GT.0)
C OR WE TRANSMIT THE CONSTANT A TO ARRAY B (IF N.LT.0).  IN EITHER CASE
C THE ARRAY LENGTH IS IABS(N).
C
C THIS SUBROUTINE SHOULD BE WRITTEN IN ASSEMBLY LANGUAGE
C
      DIMENSION A(1),B(1)
      IF(N) 10,20,25
   10 NN=-N
      AA=A(1)
      DO 15 K=1,NN
      B(K)=AA
   15 CONTINUE
   20 RETURN
   25 DO 30 K=1,N
      B(K)=A(K)
   30 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SINCOS(ARG,S,C)
C
C THIS SUBROUTINE SHOULD BE WRITTEN IN ASSEMBLY LANGUAGE, USING THE
C TABLE-LOOKUP METHOD DESCRIBED BY MITCHELL (RADAR SIGNAL SIMULATION).
C
      S=SIN(ARG)
      C=COS(ARG)
      RETURN
      END
```

```
      SUBROUTINE ETGD1(DTIME)
C
C GLINT AND DOPPLER MODULATION FOR N-POINT SCATTER MODEL
C
C NO RANGE EXTENSION
C
C SUBROUTINE REPLACES ETGDM
C
C ON INPUT.....
C
C               DTIME  = TIME SINCE LAST UPDATE IN TARGEO
C
C       /T5/    XL     = WAVELENGTH
C               PTGDSQ = PRODUCT OF TRANSMIT POWER, GAIN, AND SQUARE OF
C                        RFSS CHAMBER LENGTH
C
C       /T7/    NS     = NUMBER OF SCATTERERS VISIBLE
C               AMP(J) = AMPLITUDE OF J-TH SCATTERER
C               DA(J)  = INCREMENTAL A-VECTOR OF J-TH SCATTERER
C               DB(J)  = INCREMENTAL B-VECTOR OF J-TH SCATTERER
C               DC(J)  = INCREMENTAL C-VECTOR OF J-TH SCATTERER
C               DAD(J) = INCREMENTAL A-VECTOR RATE OF J-TH SCATTERER
C
C ON OUTPUT.....
C
C       /T8/    PEFF   = EFFECTIVE RADIATED POWER AT RFSS ARRAY
C
C       /T9/    VR,VI        = DOPPLER MODULATION SIGNAL
C               DR,DAZ,DEL = RANGE,AZIMUTH, AND ELEVATION GLINT OFFSETS
C
      COMMON /T5/ NTAP,DRTAP,DRGATE,XL,PTGDSQ
      COMMON /T7/ NS,RO,R1,ROD,AMP(20),DA(20),DB(20),DC(20),DAD(20)
      COMMON /T8/ PEFF
      COMMON /T9/ VR,VI,DR,DAZ,DEL
      DATA FOURPI/12.5663706/
C
C ZERO ACCUMULATORS
C
      VR=0.
      VI=0.
      VAR=0.
      VAI=0.
      VBR=0.
      VBI=0.
      VCR=0.
      VCI=0.
C
C LOOP OVER NS SCATTERERS
C
      DO 40 J=1,NS
```

```
      CALL SINCOS(FOURPI*(DA(J)+DAD(J)*DTIME)/XL,S,C)
      C=C*AMP(J)
      S=S*AMP(J)
      VR =VR +C
      VI =VI +S
      VAR=VAR+C*DA(J)
      VAI=VAI+S*DA(J)
      VBR=VBR+C*DB(J)
      VBI=VBI+S*DB(J)
      VCR=VCR+C*DC(J)
      VCI=VCI+S*DC(J)
   40 CONTINUE

      POW=VR**2+VI**2
      AMPL=SQRT(POW)
C
C COMPUTE GLINT OFFSETS
C
      DR =-(VAR*VR+VAI*VI)/POW
      DAZ=-(VBR*VR+VBI*VI)/(RO*POW)
      DEL=-(VCR*VR+VCI*VI)/(RO*POW)
C
C COMPUTE EFFECTIVE RF POWER
C
      PEFF=POW*PTGDSQ/(FOURPI*RO**4)
C                                                                    $
C NORMALIZE                                                          $
C                                                                    $
      VR=VR/AMPL                                                     $
      VI=VI/AMPL                                                     $

      RETURN
      END
```

```
      SUBROUTINE DATAIN
C
C READS TARGET SCATTERING DATA SUPPLIED BY M. MUMFORD (SEE SCTAMP).
C
      DIMENSION IA(1),AA(4),XX(4),YY(4),ZZ(4)
      COMMON /DP/ P(100),IP(100)
      COMMON /DQ/ Q(918)
      COMMON /T4/ NSCAT
      DATA LR,LW/5,6/
      NSCAT=10
      M=1
      DO 20 I=1,NSCAT
      PRINT 99,I
      L=10*(I-1)
   10 L=L+1
      READ  (LR,100) IA,P(L),AA,XX,YY,ZZ
      WRITE (LW,100) IA,P(L),AA,XX,YY,ZZ
      IP(L)=M
      IA=IA-2
      CALL XMIT(17,IA,Q(M))
      M=M+17
      IF(P(L).LT.180.) GO TO 10
   20 CONTINUE
      RETURN
   99 FORMAT(/29H TARGET DATA FOR SCATTERER NOI3//)
  100 FORMAT(1XI1,12XF8.3,4E14.8/(22X4E14.8))
      END
```

```
      FUNCTION SCTAMP(K,ANGL)                                          $$
C
C IN THIS SUBROUTINE WE COMPUTE THE AMPLITUDE (SQRT(RCS)) OF THE K-TH
C SCATTERER AS VIEWED FROM THE TARGET ASPECT.....
C
C     ANGL = ANGLE FROM ROLL AXIS MEASURED FROM NOSE (RAD)             $$
C
C IN ADDITION IN COMMON /T4/.....                                     $$
C                                                                     $$
C       ST = BIAS THAT IS ADDED TO ANGL (RAD)                         $$
C                                                                     $$
C THIS SUBROUTINE ACCESSES TARGET DATA SUPPLIED BY MIKE MUMFORD AT NWC/
C CHINA LAKE IN THE FORMAT DEFINED BY A COMPUTER PROGRAM WRITTEN 5/11/78
C BY E. HUTTON X3219.
C
      DIMENSION IA(1),AA(4),XX(4),YY(4),ZZ(4)
      COMMON /T4/ NSCAT,ST,AMPMIN,X(20),Y(20),Z(20)
      COMMON /DP/ P(100),IP(100)
      COMMON /DQ/ Q(918)

      ANG=ABS(ANGL+ST)*57.2957795                                     $$
      IF(ANG.GT.180.) ANG=180.                                        $$
      I1=10*(K-1)+1
      I2=I1+9
      DO 20 I=I1,I2
      IF(ANG.LT.P(I)) GO TO 25
   20 CONTINUE
   25 M=IP(I)
      CALL XMIT(17,Q(M),IA)
      IF(IA) 30,31,32
   30 SCTAMP=AA(1)+ANG*(AA(2)+ANG*(AA(3)+ANG*AA(4)))
      GO TO 35
   31 SCTAMP=EXP(AA(1)+ANG*AA(2))
      GO TO 35
   32 SCTAMP=EXP(AA(1)+ANG*AA(2))-EXP(AA(3)+ANG*AA(4))
   35 SCTAMP=.09004*SCTAMP                                            $$
      IF(SCTAMP.LT.AMPMIN) RETURN                                     $$
      X(K)=XX(1)+ANG*(XX(2)+ANG*(XX(3)+ANG*XX(4)))
      Y(K)=YY(1)+ANG*(YY(2)+ANG*(YY(3)+ANG*YY(4)))
      Z(K)=ZZ(1)+ANG*(ZZ(2)+ANG*(ZZ(3)+ANG*ZZ(4)))
      Y(K)=-Y(K)                                                      $$
      Z(K)=-Z(K)                                                      $$
      RETURN
      END
```