

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
12 June 2003 (12.06.2003)

PCT

(10) International Publication Number
WO 03/048689 A2

- (51) International Patent Classification⁷: **G01D**
- (21) International Application Number: PCT/US02/37784
- (22) International Filing Date:
25 November 2002 (25.11.2002)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
60/334,386 30 November 2001 (30.11.2001) US
10/301,485 21 November 2002 (21.11.2002) US
- (63) Related by continuation (CON) or continuation-in-part (CIP) to earlier applications:
US 60/334,386 (CON)
Filed on 30 November 2001 (30.11.2001)
US 10/301,485 (CON)
Filed on 21 November 2002 (21.11.2002)
- (71) Applicant (for all designated States except US): **BAE SYSTEMS INFORMATION AND ELECTRONIC SYSTEMS INTEGRATION INC.** [US/US]; 65 Spit Book Road, NHQ 1-719, Nashua, NH 03060-6909 (US).
- (72) Inventor; and
- (75) Inventor/Applicant (for US only): **SCHORR, David** [US/US]; 3810 Frodo Cove, Austin, TX 78739 (US).
- (74) Agent: **BARRETT, Michael, C.**; Fulbright & Jaworski, L.L.P., Suite 2400, 600 Congress Avenue, Austin, TX 78701 (US).
- (81) Designated States (national): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SC, SD, SE, SG, SI, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZM, ZW.
- (84) Designated States (regional): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, SK, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).
- Published:**
— without international search report and to be republished upon receipt of that report
- For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.



WO 03/048689 A2

(54) Title: METHODS AND APPARATUSES FOR RECONSTRUCTING ANGLE INFORMATION

(57) Abstract: Methods and apparatuses for determining angular information from a target. A plurality of sensors on an image plane are used to obtain measured amplitude data. An azimuth and elevation may be estimated using this measured amplitude data. A residual error between the measured amplitude data and calibrated amplitude data may then be determined. The angular information corresponding to the azimuth and elevation at which the residual error is minimized may then be outputted as representing or corresponding to the angular information of the target.

DESCRIPTION**METHODS AND APPARATUSES FOR RECONSTRUCTING ANGLE INFORMATION**

5

BACKGROUND OF THE INVENTION

This application claims priority to U.S. Provisional Patent Application Serial No. 60/334,386 entitled, "Methods And Apparatuses for Reconstructing Angle Information" by David Schorr. That application is incorporated herein by reference in its entirety.

10

1. Field of the Invention

The present invention relates generally to the fields of position sensors and the analysis of positional data. More particularly, it concerns methods and apparatuses for efficiently reconstructing angle information by analyzing data from a set of independent sensors. Even more particularly, it concerns the reconstruction of angle information by comparing a data set from independent sensors with a calibration data set and minimizing an error function between those sets of data to reconstruct angle information.

15

2. Description of Related Art

20

The ability to gather data from one or more sensors and translate that data into meaningful positional information is important in a wide variety of disciplines. For instance, tracking objects is essential for commercial aircraft controllers and an assortment of military applications.

25

Over the years, sundry techniques aimed at determining positional information, such as angular information, from sensor readings have been developed. Although each has shown at least a degree of success in its respective application, room for significant improvement remains.

30

U.S. Patent No. 5,724,047, which is hereby incorporated by reference, involves a precision direction finding system for making precision angle of arrival estimates for a signal received through two antenna elements separated in space. Phase interferometry is used to determine a precise angle of arrival, with multiple ambiguities due to the periodic nature of the

phase difference related to geometric angle. The interferometric ambiguities are resolved using the time difference of arrival (TDOA) of the signal at the two antenna elements. TDOA is measured using leading edge envelope detection for simple pulsed signals, and predetection correlation for phase and frequency modulated signals. Although such phase interferometry determinations may exhibit certain advantages over other conventional methods, shortcomings remain due to, at least in part, the relatively complex nature of the computations required to implement the methodology. Further, reliance upon two antenna elements, rather than a plurality of independent sensors, may inhibit at least some flexibility of such a system.

U.S. Patent No. 6,104,345, which is hereby incorporated by reference, involves a method for the direction of arrival (DOA) tracking of at least one source, along a single azimuth axis or along both azimuth and elevation axes. The method includes the steps of selecting all the high peaks from a DOA function as potential track points, converting the potential track points into a plurality of tracks and selecting a true track from the plurality of tracks. This methodology also suffers from some shortcomings; for example, it may be desirable to avoid the particular computational complexity involved with selecting a true track from the plurality of potential track points.

U.S. Patent No. 4,562,439, which is hereby incorporated by reference, involves an imaging radar seeker for producing two-dimensional images of a target, which can be mounted on a missile or other moving body, such as an automobile. A computer directs the seeker to operate sequentially in searching, tracking, and imaging modes. In the searching mode, a combination of circumferential rotation of an antenna of the seeker and frequency scanning of electromagnetic energy fed to the antenna enables the seeker to search for its target over a conical field-of-view or a wider, peripheral belt field-of-view. In the imaging mode, circumferential rotation of an antenna is stopped, and the tilt angle of a linear array of the antenna is stepped or continuously moved to compensate for radial movement of a radiated beam caused by frequency stepping imparted by a frequency synthesizer. This keeps the beam fixed in space and centered on the target. Inverse synthetic aperture imaging is used to create a two-dimensional image of the target wherein the first dimension (range) is obtained by performing inverse Fourier transforms on the echo signals, and the second orthogonal dimension (cross-range or doppler frequency) is obtained by performing Fourier transforms. The array can be a

linear array of E-plane stacked linear waveguide antenna elements operating in either the traveling wave mode or the standing wave mode. Such methodology too exhibits shortcomings at least due to its complexity, lack of flexibility, and its inability to more efficiently perform angle reconstruction determinations.

5

U.S. Patent No. 5,818,393, which is hereby incorporated by reference, involves a fixed body wide field-of-view conformal antenna array suitable for broadband precision direction finding on missile platforms. The array is configured as multiple sub-arrays of spiral antennas that cover particular regions within the desired field-of-view of the entire array. A lower cost, more reliable and more accurate direction finding solution for missile needs is provided, primarily by the elimination of conventional radomes and antenna gimbal structures. The array can be configured to include multi-mode sensors. Although useful in this elimination of structures, this methodology exhibits room for significant improvement given its inability to more efficiently reconstruct angular information.

10

15

U.S. Patent No. 6,313,794, which is hereby incorporated by reference, uses feedback from RF carrier frequency measurements to disassociate the emitter angle-of-arrival component in the ambiguous phase measurement from the initially unknown phase measurement integer ambiguities. It then resolves the ambiguities and obtains the correct emitter angle-of-arrival (AOA). This is accomplished by converting the actual interferometer baselines on which the unassociated pulse phase measurements were made at different emitter frequencies to a baseline set for a single-frequency equivalent interferometer array. This methodology, as with the methodologies described above, suffers from issues relating to complexity, flexibility, and the inability to more quickly and efficiently reconstruct high-resolution angular information.

20

25

U.S. Patent No. 5,657,251, which is hereby incorporated by reference, involves a computer-implemented process for processing incoming target data from a focal plane or scanning radar to accomplish multiple Target Tracking. Inputs are pixel plane coordinates and intensity of target blips. The Intelligent Target Tracking Processor (ITTP) employs an optimal target tracking algorithm. An optimal observation-to-track assignment exists when all target blips in a new frame of target data are matched up with nearby tracks, such that the sum of all the distances from each target blip to its assigned track is minimized. An expert system is used to

30

control overall processing flow and provide efficient allocation of computing resources. Target blips without near neighbors are allowed to go directly to a real track table of established tracks, if their coordinates match-up with projected tracking gates. Otherwise, target blips are tested sequentially against two-frame, three-frame, and four- or higher-frame discriminants, to reject blips not belonging to established tracks. The ITTP can partition the pixel plane into "bite size partitions", each with a manageable number of target blips, which it handles sequentially. The ITTP is designed to handle hundreds or thousands of targets as a stand-alone processor as is required in space object tracking or military scenarios. The expert system maintains an optimum balance between correlating on existing tracks and discriminating against impossible tracks. A total of 26 different metric and radiometric target tracking discriminants are employed. The ITTP is a dynamically and optimally configured set of general purpose parallel processors. Although useful for trajectory tracking for applications such as air-traffic controlling, this methodology nevertheless includes shortcomings; for instance, it does not allow for the reconstruction of high quality, high resolution angle information using a plurality of independent sensors or detectors on an image plane.

Problems enumerated above are not intended to be exhaustive, but rather are among many that tend to impair the effectiveness of previously known techniques concerning positional tracking and the determination of angle information. Other noteworthy problems may also exist; however, those presented above should be sufficient to demonstrate that methodology appearing in the art have not been altogether satisfactory. In particular, existing techniques do not adequately allow for the determination of angle information with improved resolution from independent sensors or detectors on an image plane.

SUMMARY OF THE INVENTION

Shortcomings listed above are reduced or eliminated by the techniques disclosed herein. These techniques are applicable to a vast number of applications, including general targeting and tracking applications.

In one respect, the invention is a method for determining angular information from a target. Measured amplitude data is obtained from a plurality of sensors. An azimuth and

elevation are estimated using the measured amplitude data. Calibrated amplitude data corresponding to the azimuth and elevation is obtained. A residual error between the measured amplitude data and the calibrated amplitude data is determined. Aspects of this process are performed iteratively until the residual error is minimized. The angular information, which corresponds to the azimuth and elevation at which the residual error was minimized, is then output.

In other respects, the invention is a computer program, comprising computer or machine readable program elements translatable for implementing the method described above. Further the invention is an apparatus for performing such a method.

In another respect, the invention is a method for determining angular information from a target. A plurality of sensors on an image plane are obtained. The target or a point target is defocused on the image plane such that one or more of the sensors respond to a target pulse. Measured amplitude data from the plurality of sensors is obtained. An azimuth and elevation are estimated using the measured amplitude data. Calibrated amplitude data corresponding to the azimuth and elevation is obtained. A residual error between the measured amplitude data and the calibrated amplitude data is determined. Aspects of this process are performed iteratively until the residual error is minimized. The angular information, which corresponds to the azimuth and elevation at which the residual error was minimized, is then output.

In other respects, the invention is a computer program, comprising computer or machine readable program elements translatable for implementing the method described above. Further the invention is an apparatus for performing such a method.

In another respect, the invention is a system for determining angular information from a target. The system includes a plurality of sensors and a computer. The computer is configured to: (a) acquire measured amplitude data from the plurality of sensors; (b) estimate an azimuth and elevation using the measured amplitude data; (c) access calibrated amplitude data corresponding to the azimuth and elevation; (d) determine a residual error between the measured amplitude data and the calibrated amplitude data; and (e) output the angular information as corresponding to the azimuth and elevation at which the residual error is minimized.

Other features and associated advantages will become apparent with reference to the following detailed description of specific embodiments in connection with the accompanying drawings.

5

BRIEF DESCRIPTION OF THE DRAWINGS

The following drawings form part of the present specification and are included to further demonstrate certain aspects of the present invention. The invention may be better understood by reference to one or more of these drawings in combination with the detailed description of specific embodiments presented herein.

10

FIG. 1 is a flowchart showing techniques for reconstructing angle information in accordance with embodiments of the present disclosure.

15

FIGS. 2A and **2B** show exemplary data involving angle information in accordance with embodiments of the present disclosure.

DESCRIPTION OF ILLUSTRATIVE EMBODIMENTS

20

The present disclosure describes apparatuses and methods for efficiently determining high-quality, high-resolution angle information from several independent sensors on an image plane. Applications for this technology are vast. For instance, one may use the techniques disclosed herein in a variety of military applications. Similarly, one may find applications in aircraft traffic controlling, general target seeking and/or tracking, and any other application where angular information is required. With the benefit of the present disclosure, those having skill in the art will comprehend that the techniques disclosed herein may be modified and applied to a number of additional, different applications. The present disclosure and claims attached hereto cover all such modifications that fall within the scope and spirit of this disclosure.

25

30

In one embodiment, the techniques for determining angle information may take the form of what the inventor has coined an “angle reconstruction algorithm.” The angle reconstruction

algorithm is one method for improving the resolution and quality of angle information from relative amplitude measurement information given a set of independent sensors or detectors on an image plane. These sensors may be part of a CCD, thermal imaging array, a quadrant detector, or any other device suitable for gathering data relevant to angular information. The angle reconstruction process may involve producing a “match” between predicted or calibrated relative amplitude responses and measured relative amplitude responses. In one embodiment, this match may involve minimizing a residual error function, although those having skill in the art will understand that other mathematical techniques may be employed to determine a correspondence between the predicted or calibrated amplitudes and the measured responses.

In one embodiment, a LASER seeker field-of-view (FOV) may be broken up into a plurality of distinct pixels. In one embodiment, seven pixels may be used, although more or fewer pixels may be utilized depending upon preference and/or the specific application in question. A point source may be defocused on the image plane such that a plurality of sensors respond to a given target pulse. In the embodiment using seven pixels, for instance, a point source may be defocused such that at least 3 of 7 independent detectors respond to a target pulse.

The resulting amplitudes from each of the detectors (7 in the embodiment discussed above) for a given received pulse may then be compared with expected amplitudes from either predictions or calibration data for each location in the FOV. The location in the FOV that produces a match or correspondence, which may take the form of exhibiting the least residual error between the predicted response magnitudes and the actual response amplitude measurements, is then determined to be the target’s location in the field of view. Hence, angle (and position) information may be efficiently obtained.

As will be understood by those having skill in the art, one may determine an appropriate correspondence between measured amplitudes and prediction or calibration sets of data in many ways, including the determination of a minimum residual error. In one embodiment, a rather robust and computationally-intensive approach is to accomplish a grid search of a region of interest. The residual between the amplitude response prediction and the actual measured amplitude response may be compared for all grid points. The grid point resulting in the lowest

residual is assumed to be the best estimate of the targets location in the FOV. Refinement of the grid search interval may be accomplished until a particular desired level of accuracy is achieved.

5 Other computationally-efficient methods of finding the location in the FOV with the lowest residual may be readily implemented according to programming and mathematical methods known in the art. Different methods may, of course, amount to a significant reduction in the computational effort required to find a target location in the field of view.

10 Angle reconstruction techniques of this disclosure applied to an imaging array or other sensors provide for angle resolution much greater than would be derived based on the estimated angular extent of a pixel on the image plane. Intentional defocusing of the target (when angular extent is less than 1 pixel) or point target allows for relative amplitude data to be collected on multiple adjacent pixels. The relative amplitude data from each pixel can then be utilized to determine the power centroid of an extended target or a point source in the FOV.

15

These techniques allow very noisy non-linear raw sensor data to be interpreted into angle of arrival (AOA) with the aid of a database and an appropriate calibration set of data or data forming the basis of a prediction set of data. Noise corruption of the measured signal levels often prevents a perfect match to angle of arrival to be made; however, extremely good estimates of angle of arrival may be readily made with an angle reconstruction algorithm as described herein. No loss of bandwidth need be realized since previous data does not have to be utilized in determining the angle of arrival output.

20

Apparatuses suitable for implementing aspects of the methods described herein may include equipment known in the art such as sensors, detectors, seekers, etc. Additionally, the apparatuses may include one or more computers to analyze the sensor data, in accordance with the techniques described herein via corresponding software. The one or more computers may be networked and may be in contact with one or more local or remote databases or storage networks that store calibration or prediction data sets relating to the sensors or arrays.

25
30

The following examples are included to demonstrate specific embodiments of this disclosure. It should be appreciated by those of skill in the art that the techniques disclosed in

the example which follow represent techniques discovered by the inventor to function well in the practice of the invention, and thus can be considered to constitute specific modes for its practice. However, those of skill in the art should, in light of the present disclosure, appreciate that many changes can be made in the specific embodiments which are disclosed and still obtain a like or similar result without departing from the spirit and scope of the invention.

EXAMPLE 1

Turning to **FIG. 1**, there is shown a flowchart illustrating one embodiment of a suitable angle reconstruction algorithm in which 7 detectors are used. As the flowchart shows, amplitudes from the seven detectors are first measured and then normalized, if necessary. This information is used to form an initial estimate of both azimuth (AZ) and elevation (El) information. Using a database or another appropriate storage device or network, one may then extract amplitude information corresponding to the initially-estimated AZ and El from calibration or prediction data. This amplitude information may be normalized, if necessary. With the measured and calibration amplitude information, both corresponding to the initially-estimated AZ and El, one may then determine a residual error function (or determine a correspondence between the data sets using an alternative method).

In **FIG. 1**, the amplitude residual is determined by considering the difference between the measured amplitude and the amplitude from the calibration set, for each and every detector (detectors A through G). Specifically, one may sum the squares of the differences of those amplitudes, as shown, to arrive at the residual value. In **FIG. 1**, A_{meas} means the measured amplitude from detector A, while A_{est} means the amplitude at the estimated AZ and El from the calibration set.

If this residual value is not the minimum value, the initial estimate of AZ and El is updated, as shown. With this new AZ and El estimate, amplitude information may again be extracted from the calibration data set — this time, the calibration information will correspond to newly-estimated AZ and El. Again, normalization of the calibration amplitude information may be required. Once again, a residual error function may be obtained by considering the

differences between the measured and calibration amplitudes. And, once again, if the residual is not minimized, the estimate of AZ and EI may be updated, thereby repeating the process.

5 However, if the residual is minimized, one may output the AZ and EI that accountable for the minimization. It is this AZ and EI that represents the angle information of the target or object being studied.

FIGS. 2A and 2B show exemplary amplitude data from seven detectors.

10 This example and the disclosure herein shows that, in general, target location and angular information may be triangulated by comparison to a look up table (which may take the form of the calibration data set or prediction data discussed above). The look up table may be several dimensions, including 3. The look up table may be built-up via calibration steps. Specifically, one may position a target sequentially at various, known locations throughout a grid. At each
15 location, one may take sensor readings from a plurality of sensors to obtain relative light amplitudes. Taking multiple readings of this sort builds up a library of amplitudes as a function of target locations. This library may then be used to compare with measured amplitudes in the fashion described herein (for example, by determining a residual error function) to quickly determine the angular and positional information of a target based on measured amplitude data
20 from those same sensors.

EXAMPLE 2

25 Shown in this example is source code, written in FORTRAN, that is suitable for carrying out steps described herein. This source code is exemplary only and does not limit the scope of the claims appended hereto. It simply represents one specific embodiment for carrying out aspects of this disclosure and is included for the convenience of the reader in this regard. Those having skill in the art, with the benefit of this disclosure, will recognize that a wide variety of computational techniques and different types of corresponding source code and software may be
30 used in implementing the concepts described herein.

```

Program AngleRecon.dof
C*****
5 C This code synthesises ideal detector amplitudes from AZ EL seeker angle
C and corrupts the amplitudes with predicted noise.
C The corrupted detector amplitudes are then utilized to reconstruct the
C original seeker angles through an angle reconstruction algorithm.
10 C The effects of the noise on the detector amplitudes in studied.

C Need to check optical gain profile in data statements and optical efficiency.
C*****
    USE MSIMSL
    USE MSFLIB
15 C    USE PORTLIB
    IMPLICIT DOUBLE PRECISION (A-H,O-Z)

    Parameter (NDET = 7)
    Dimension Signal(NDET)
    Dimension AZ(1000), EL(1000)
    Double Precision SNR, SNREST
20
C-----
C*****
25 C All of these files are not necessarily used every time. These are
C All of these files are output files.

    OPEN(UNIT=7, FILE='Amplitude_Noise.xls')
    Write(7,*) 'TargetRange AZtruth ELtruth SNR AZnoise ELnoise'

30    OPEN(UNIT=8, FILE='CalMatrix_20X20.txt')

    OPEN(UNIT=9, FILE='Errormap.xls')

    OPEN(UNIT=10, FILE='Anglemeas.xls')
35    OPEN(UNIT=11, FILE='Angleres.xls')

C-----
40 C*****
c    Nuber of Monte Carlo noise cases for shotgun plot and noise statistics

    Nmontenoise = 100
45 C*****

    Do 6000 Iseekercase=1,3,1
50
    If(Iseekercase.EQ.1) Then
        Aztruth = 0.0
        Eltruth = 0.0
        OPEN(UNIT=12, FILE='Anglestatsz0-0.xls')
55        Write(12,*) 'TargetRange,AZtruth,ELtruth,Nmontenoise,AZmean,
        &ELmean,AZrms@0/0,ELrms@0/0'

        ElseIf(Iseekercase.EQ.2) Then
60        Aztruth = 5.00
        Eltruth = 0.00
        OPEN(UNIT=12, FILE='Anglestatsz5-0.xls')
        Write(12,*) 'TargetRange,AZtruth,ELtruth,Nmontenoise,AZmean,
        &ELmean,AZrms@5/0,ELrms@5/0'

```

```

    ElseIf(Iseekercase.EQ.3) Then
      Aztruth = 10.0
      Eltruth = 0.0
5      OPEN(UNIT=12, FILE='Anglestatsz10-0.xls')
        Write(12,*) 'TargetRange,AZtruth,ELtruth,Nmontenoise,AZmean,
&ELmean,AZrms@10/0,ELrms@10/0'

10      ElseIf(Iseekercase.EQ.4) Then
        Aztruth = 4.00
        Eltruth = 7.00
      ElseIf(Iseekercase.EQ.5) Then
15      Aztruth = 8.50
        Eltruth = -10.75
      ElseIf(Iseekercase.EQ.6) Then
        Aztruth = 0.50
        Eltruth = -13.00
20      ElseIf(Iseekercase.EQ.7) Then
        Aztruth = -12.75
        Eltruth = 12.75
      ElseIf(Iseekercase.EQ.8) Then
        Aztruth = 16.9
        Eltruth = 10.25
25      ElseIf(Iseekercase.EQ.9) Then
        Aztruth = -2.25
        Eltruth = -5.00
      ElseIf(Iseekercase.EQ.10) Then
        Aztruth = -10.75
30      Eltruth = -3.00
      ElseIf(Iseekercase.EQ.11) Then
        Aztruth = 13.25
        Eltruth = 11.75
      Endif
35
c      AZtruth=float(Iseekercase)/10.0
c      ELtruth=0.0

40
c*****
c***** This is the MonteCarlo Noise Loop *****
  Do 1000 Irange=100,6000, 100

45
    Do 2000 Inoise=1, Nmontenoise

      TargetRange = dmax1(Float(Irange),50.0)
      DesignatorRange = 6000.0

50
      Call DetectorAmps(TargetRange, DesignatorRange,
&      AZtruth, ELtruth, SNREST, Signal)
      Call SeekerNoise(Signal, AZnoise, ELnoise)

c      Write(*,1500)TargetRange,AZtruth,ELtruth,SNREST,AZnoise,ELnoise
c      Write(12,1500)TargetRange,AZtruth,ELtruth,SNREST,AZnoise,ELnoise
55      Write(7,1500)TargetRange,AZtruth,ELtruth,SNREST,AZnoise,ELnoise
1500      Format(9F10.3)

c Save results to array for post processing statistics
  AZ(Inoise)=AZnoise
60      EL(Inoise)=ELnoise

2000 Continue
c***** End of Monte Carlo Noise Loop *****
c*****Calculate Summary Statistics Data *****
65

```

```

    Call Anglestats(Inoise-1,AZ, EL, AZmean, ELmean, AZrms, ELrms)
    Write(*,1500) TargetRange,AZtruth,ELtruth,
    &                                     dFloat(Inoise-1),AZmean,ELmean,AZrms,ELrms
5      Write(12,1500) TargetRange,AZtruth,ELtruth,
    &                                     dFloat(Inoise-1),AZmean,ELmean,AZrms,ELrms

1000 Continue
c***** End of AZ,EL case study *****
c*****
10      close (12)
6000 Continue
    Stop
    END
c*****
15      C END of trash code used to exercize the subroutine for seeker noise.
c*****
c*****
    Subroutine AngleStats(N,AZ, EL, AZmean, ELmean, AZrms, ELrms)

20      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
    Dimension AZ(1000), EL(1000)

c calculate mean of n data points
25      AZsum=0.0
    ELsum=0.0
    AZres=0.0
    ELres=0.0

30      Do 1000 i=1,N
    AZsum=AZsum + AZ(i)
    ELsum=ELsum + EL(i)
1000 continue
    AZmean=AZsum/n
35      ELmean=ELsum/n

    Do 2000 i=1,N
    AZres=AZres + (AZ(i) - AZmean)**2
    ELres=ELres + (EL(i) - ELmean)**2
40      2000 continue
    AZrms=SQRT(AZres/Dfloat(N-1))
    ELrms=SQRT(ELres/Dfloat(N-1))

    return
45      end

c*****
c*****
50      Subroutine DetectorAmps(TargetRange, DesignatorRange,
    &                               AZtruth, ELtruth, SNREST, Signal)

c*****
55      C This code generates ideal detector signal levels, then corrupts those signals
    C with the anticipated noise levels from the APD and
    C assumed internal SNR limits.
    C
    C As a note of caution, no approximation to scintillation is included
60      C although this is felt to be a dominat influence at
    C intermediate ranges (i.e. from 500 meters to 3 km)
    C
    C Inputs:
65      C TargetRange = range from Seeket to Target in meters
    C DesignatorRange = range from Designator to target in meters

```

```

C      AZtruth      = true AZ line of sight
C      ELtruth      = true EL line of sight

C  Outputs:
5  C      SNREST      = predicted signal to noise ratio
C      Signal      = predicted normalized detector signal levels
C                      corrupted with noise
C
C*****
10  C-----
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C-----
C Parameters
C-----
15  C
      Parameter (NAZ = 81)
      Parameter (NEL = 81)
      Parameter (NDET = 7)
C-----
20  C-----
      Dimension Signal(NDET)
      Dimension OPT_GAIN(NDET)
25  C      Double Precision TotAng(NDET)
      Double Precision Theta
      Double Precision whatimeisit

      Logical Firstcall

30  C      Dimension Anglens(3,NDET)

      COMMON / Gainlens / Nangles, Angles(200), OptTrans(200)
      COMMON / Caldata / AZtbl(NAZ), ELtbl(NEL), Tgain(NAZ,NEL,NDET)
35  C
C CONSTANTS

      Parameter (PI=3.1415925)
40  C      Parameter (RTD = 180.0d0/PI)
      Parameter (DTR = PI/180.0d0)

C Optical Gain vs Total Angle for individual Lens (30 data points)
45  C
      Data Nangles / 30 /

      Data Angles / 0.0, 0.5, 1.0, 1.5, 2.0,
&      2.5, 3.0, 3.5, 4.0, 4.5,
&      5.0, 5.5, 6.0, 6.5, 7.0,
&      7.5, 8.0, 8.5, 9.0, 9.5,
50  C      &      10.0,10.5,11.0,11.5,12.0,
&      12.5,13.0,13.5,14.0,180.0, 170*0 /

c This data is from Zemax
55  C      Data OptTrans / 0.146, 0.147, 0.154, 0.166, 0.168,
&      0.170, 0.156, 0.150, 0.141, 0.137,
&      0.128, 0.120, 0.113, 0.107, 0.102,
&      0.090, 0.087, 0.077, 0.072, 0.061,
&      0.055, 0.046, 0.039, 0.030, 0.025,
60  C      &      0.016, 0.011, 0.005, 0.000, 0.000, 170*0 /

c This data is a slice from measure calibration used for outdoor testing at TA-3
c      Data OptTrans / 0.150, 0.146, 0.141, 0.140, 0.133,
65  C      &      0.123, 0.115, 0.107, 0.100, 0.090,

```

```

C      &          0.082, 0.073, 0.065, 0.057, 0.049,
C      &          0.041, 0.035, 0.028, 0.023, 0.019,
C      &          0.015, 0.011, 0.009, 0.007, 0.005,
C      &          0.003, 0.002, 0.001, 0.001, 0.000, 170*0 /

```

5

Data Firstcall /.True./

10

```

C-----
C*****
C-----
C-----
C READ OPTICAL gain versus angle data.
C This is the only required input file.
C-----

```

15

```

If (Firstcall) then
Firstcall=.False.

```

20

```

C Point of each of 7 lenses
C *****Lens Pattern*****
C      4   3

```

25

```

C      5   1   2
C      6   7
C*****

```

30

```

Anglens(1,1) = 0.0
Anglens(2,1) = 0.0

```

```

Anglens(1,2) = 14.00
Anglens(2,2) = 0.0

```

35

```

Anglens(1,3) = 7.0
Anglens(2,3) = 12.12

```

40

```

Anglens(1,4) = -7.00
Anglens(2,4) = 12.12

```

```

Anglens(1,5) = -14.00
Anglens(2,5) = 0.00

```

45

```

Anglens(1,6) = -7.00
Anglens(2,6) = -12.12

```

```

Anglens(1,7) = 7.0
Anglens(2,7) = -12.12

```

50

```

C*****
C*****
C Generate Analytical Calibration tables from
C Analytical Optical Gain Profiles on 0.5 degree intervals
C This will be replaced with measured calibration data when available.

```

55

```

Do 1000 IEL= NEL,1,-1
Do 2000 IAZ= NAZ,1,-1

```

60

```

AZ = (dfloat(IAZ)*0.5d0-20.5d0)
EL = (dfloat(IEL)*0.5d0-20.5d0)

```

65

```

Aztbl(IAZ) = AZ
Eltbl(IEL) = EL

```

```

C Determine total angle in radians to each lens
  Do 3000 Idet = 1,7

5      TotAng(Idet)=sqrt((Anglens(1,Idet)-AZ)**2+(Anglens(2,Idet)-EL)**2)
      Tgain(IAZ,IEL,Idet) = OPGain3(TotAng(Idet))/OPGain3(0.0)

3000  Continue

10
C Write out calibration file for MatLab Plot utilities.
C   Write(8,1600) AZ, EL, 1.00 * Tgain(IAZ,IEL,1), Tgain(IAZ,IEL,2) ,
C   &                                     Tgain(IAZ,IEL,3), Tgain(IAZ,IEL,4) ,
C   &                                     Tgain(IAZ,IEL,5), Tgain(IAZ,IEL,6) ,
15  C   &                                     Tgain(IAZ,IEL,7)

2000  Continue
1000  Continue

20  1600  Format(17(F10.4,','),F10.4)
      Write(*,*) ' Calibration Table Generation Complete.'

C-----
C-----
25  C Aperture of individual Optic in cm^2
      APERTURE = 0.78/10000.0

C Total FOV of individual lens in radians
C   (assumed half power full width)
30  TLFOV = 14.0*DTR

C Fraction of Photons going thru aperture than hit detector
C Includes numerical aperture effects but excludes spot size vs ferrul size.
C Optical Gain Profile updated to include overall efficiency.
35  Opt_eff= 1.00

C APD Gain
      Gain = 20

40  C Temperature (in Kelvin)
      Temp = 300

C LASER Pulse Width (seconds)
      Plwid = 20e-9
45  C BW of filter on amplifier(Hz)
      BW = 6.0e6
      BWnoise = BW*PI
      Tau=1/(2*pi*BW);
50  BWfactor=(1-exp(-Plwid/Tau))

C Power of the laser(Watts)
      POWER_LASER = 5.0e6

55  C Atmospheric extinction coefficient (km)
C   EXT_COEF = -0.09135
      Visibility = 23.5d0
      EXT_COEF=- (10.0d0**(-0.136d0+(1.16d0*log10(3.912d0/Visibility))))

60  C Target Reflectivity
      REFLTAR = 0.1

C Reflectivity of background (i.e. terrain)
65  REFLBACK = 0.1

```

```

C Number of detectors per aperture
  Num_det = 1

5  C Number of apertures in the reciever
  NUM_APERT_RECV = 3
  NUM_APERT_BACK = 4

C INCIDENT SPECTRAL FLUX WATTS/(M^2 * MICRON) 1.06 microns
10  SPEC_FLUX = 500.

C OPTICAL FILTER BANDWIDTH IN MICRONS
  FIL_BW = .010

15  C Range in meters
  C   Range = 6000

  Endif
C End of calculations done for first subroutine call only
C-----
20  C-----
C*****
C Note reference to optical gain at 0 radians offset.
C Note assumes designator fixed at 6km while target range is variable.

25  EXTINCT = EXP( EXT_COEF*(TargetRange+DesignatorRange)/1000)
  POWER = POWER_LASER*EXTINCT*REFLTAR*APERTURE*
  &      FLOAT(NUM_APERT_RECV)*OPT_EFF*OPGAIN3(0.0)/
  &      (3.14159*TargetRange**2*FLOAT(NUM_DET))

30  BACKPOW = (TLFOV)**2*SPEC_FLUX*FIL_BW*APERTURE*NUM_APERT_BACK*
  &      REFLBACK*OPT_EFF*OPGAIN3(0.0)/(NUM_DET*8.)

  TotalPower = POWER + BACKPOW
  Call DETECT(TotalPower,GAIN,TEMP,BWnoise,ZNEP)

35  SNRlimit=1000.0
  zneplim=sqrt(znep**2+(1/SNRlimit*BWfactor*POWER)**2)
  SNR = power*BWfactor/zneplim

40  C*****
  C-----
  C-----
C Get ideal "truth amplitudes" from optical model for truth Az,El location
C before adding noise to ideal amplitudes.
45  C Determine total angle in DEGREES to each lens.

  TotAng(1)=sqrt((Anglens(1,1)-AZtruth)**2
  &      +(Anglens(2,1)-ELtruth)**2)
50  TotAng(2)=sqrt((Anglens(1,2)-AZtruth)**2
  &      +(Anglens(2,2)-ELtruth)**2)
  TotAng(3)=sqrt((Anglens(1,3)-AZtruth)**2
  &      +(Anglens(2,3)-ELtruth)**2)
  TotAng(4)=sqrt((Anglens(1,4)-AZtruth)**2
  &      +(Anglens(2,4)-ELtruth)**2)
55  TotAng(5)=sqrt((Anglens(1,5)-AZtruth)**2
  &      +(Anglens(2,5)-ELtruth)**2)
  TotAng(6)=sqrt((Anglens(1,6)-AZtruth)**2
  &      +(Anglens(2,6)-ELtruth)**2)
  TotAng(7)=sqrt((Anglens(1,7)-AZtruth)**2
60  &      +(Anglens(2,7)-ELtruth)**2)

C Determine Optical Gain for each lens
  Opt_gain(1) = OPGain3(TotAng(1))/OPGain3(0.0)
65  Opt_gain(2) = OPGain3(TotAng(2))/OPGain3(0.0)
  Opt_gain(3) = OPGain3(TotAng(3))/OPGain3(0.0)

```

```

Opt_gain(4) = OPGain3(TotAng(4))/OPGain3(0.0)
Opt_gain(5) = OPGain3(TotAng(5))/OPGain3(0.0)
Opt_gain(6) = OPGain3(TotAng(6))/OPGain3(0.0)
Opt_gain(7) = OPGain3(TotAng(7))/OPGain3(0.0)

5
C Add noise to signal
Signal(1) = power*OPT_GAIN(1) + znep*dnorin(zrand())
Signal(2) = power*OPT_GAIN(2) + znep*dnorin(zrand())
Signal(3) = power*OPT_GAIN(3) + znep*dnorin(zrand())
10 Signal(4) = power*OPT_GAIN(4) + znep*dnorin(zrand())
Signal(5) = power*OPT_GAIN(5) + znep*dnorin(zrand())
Signal(6) = power*OPT_GAIN(6) + znep*dnorin(zrand())
Signal(7) = power*OPT_GAIN(7) + znep*dnorin(zrand())

15
C Normalize all measured amplitudes to maximum
Signalmax = MAX(Signal(1),Signal(2),Signal(3),
& Signal(4),Signal(5),Signal(6),
& Signal(7))
20 Signal(1) = Signal(1)/Signalmax
Signal(2) = Signal(2)/Signalmax
Signal(3) = Signal(3)/Signalmax
Signal(4) = Signal(4)/Signalmax
Signal(5) = Signal(5)/Signalmax
25 Signal(6) = Signal(6)/Signalmax
Signal(7) = Signal(7)/Signalmax

C Estimated SNR from measured pulse amplitude
C SNRLimit=100.0
30 C znep=sqrt(znep**2+(1/SNRLimit*BWFactor*POWER)**2)
C SNR = power*BWFactor/znep

znepmeas=sqrt(znep**2+(1/SNRLimit*BWFactor*Signalmax)**2)
35 SNREST = (Signalmax * BWFactor) / znepmeas

Return
End

40
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
45 C*****
C*****

Subroutine SeekerNoise(Signal, AZ, EL)

C*****
50 C The seeker angle is then recostructed with the corrupted signals to approximate the
C anticipated seeker angle noise.
C Seeker angles are reconstructed from detector amplitudes
C utilizing a coarse grid search followed
C by a gradient search technique to minimize the residual.
55 C*****

IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C-----
60 C Parameters
C-----

Parameter (NAZ = 81)
Parameter (NEL = 81)
Parameter (NDET = 7)
65

```

```

C-----
      Dimension Signal(NDET)
      Dimension OPT_GAIN(NDET)
5      Double Precision TotAng(NDET)
      Double Precision Theta
      Double Precision whatimeisit

      Logical Firstcall

10      Dimension Anglens(3,NDET)

      COMMON / Gainlens / Nangles, Angles(200), OptTrans(200)
      COMMON / Caldata / AZtbl(NAZ), ELtbl(NEL), Tgain(NAZ,NEL,NDET)
15 C-----
C*****
C*****
C*****
C Angle reconstruction from amplitudes for a given pulse begins here.
20 C Do coarse grid search to determine starting point for gradient search.
C*****
C*****
C Initial center of search pattern and pattern density
25      BestResidual = 9999.0
      Azcenter = 0.0
      Elcenter = 0.0
      DelAz = 1.0
      DelEl = 1.0

30      9000 Continue

      Do 7000 IEL= -15,15,2
      Do 8000 IAZ= -15,15,2

35      AZest = (dfloat(IAZ)*DelAz+Azcenter)
      If(Azest.gt. 15.0d0) Azest = 15.0
      If(Azest.lt.-15.0d0) Azest = -15.0

      ELest = (dfloat(IEL)*DelEl+ElCenter)
40      If(Elest.gt. 15.0d0) ELest = 15.0
      If(Elest.lt.-15.0d0) ELest = -15.0

      C Determine residual for point in grid.
      Call DetResidual(AZest,ELest,Signal,Residual)
45

      C Save Az and El for search grid point having lowest residual
      C for further grid refinement
      If(Residual.LT.BestResidual) Then
          BestResidual=Residual
50          Az=Azest
          El=Elest
      endif
      8000 Continue
      7000 Continue

55

C*****
C*****
C Gradient Search Algorithm starts here.
C*****
60 C*****
      Residual = 9999.0
      DelAZ = 0.5
      DelEL = 0.5

65      Do While (DelAZ.GT.0.01.OR.RESest.LT.Residual)

```

```

C Determine Base AZ,EL Residual
  Call DetResidual (AZ,EL,Signal,Residual)

5  C Determine Local Residual Slope with AZ
    AZPDAZ=AZ + DelAZ
    Call DetResidual (AZPDAZ,EL,Signal,ResDAZ)
    AZReslope=(ResDAZ-Residual)/DelAZ

10 C Determine Local Residual Slope with EL
    ELPDEL=EL+DeleEL
    Call DetResidual (AZ,ELPDEL,Signal,ResDEL)
    ELReslope=(ResDEL-Residual)/DeleEL

15 C Determine direction weighting Factors (sin and cos)
    Denom=max(SQRT(AZReslope**2+ELReslope**2),1.0d-6)
    AZfactor = -AZReslope/Denom
    ELfactor = -ELReslope/Denom
20  If (Denom.LE.1d-6) then
    AZfactor=1.0
    ELfactor=1.0
    Endif

25 C Determine New AZ, EL estimate and calculate residual
    AZest=AZ+AZfactor*DelAZ
    ELest=EL+ELfactor*DeleEL
    Call DetResidual (AZest,ELest,Signal,REsEst)

30 C Accept new AZ, EL estimate if Residual improved
C Otherwise assume step size to big (stepped over best solution)
    Itry=Itry+1
    If (REsEst.GT.Residual.or.Itry.gt.1000) Then
35  DelAZ=DelAZ/5.0
    DeleEL=DeleEL/5.0
    Itry=0
    Else
40  AZ=AZest
    EL=ELest
    Endif

C-----
45 C TheEnd of loop converging on angle estimate
C Reduce search deltas and continue.
C This is an intermediate iteration result.
C-----

50 C Write(10,1500) AZtruth, ELtruth, AZ, EL, REsEst
1500 Format(9F12.6)

    Enddo

55 C*****
C*****
C TheEnd of Angle Reconstruction Algorithm
C*****
C*****
C-----
60 C TheEnd of loop for particular AZ, EL Truth Data Case Study
C-----
C-----
C-----
65 5000 Return
    End

```

```

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
5  CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

C*****
Subroutine DetResidual(AZest,Elest,Signal,Res)
10  IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C-----
C Parameters
C-----

15  Parameter (NAZ = 81)
    Parameter (NEL = 81)
    Parameter (NDET = 7)

C-----

20  Logical ISEARCH, IWARN

    Dimension Signal(NDET), Sigest(NDET)

25  COMMON / Gainlens / Nangles, Angles(200), OptTrans(200)
    COMMON / Caldata / AZtbl(NAZ), ELtbl(NEL), Tgain(NAZ,NEL,NDET)
C-----
C Determine residual for all points in search grid
C Retrieve amplitudes for search grid Angles from Calibration DataBase
30
    ISEARCH = .TRUE.
    Call Intrp2(Aztbl, Eltbl, Tgain(1,1,1), NAZ,NEL,NAZ,NEL,
    &           AZest,Elest,Sigest(1),ISEARCH,IWARN)
    ISEARCH = .FALSE.
35  Call Intrp2(Aztbl, Eltbl, Tgain(1,1,2), NAZ,NEL,NAZ,NEL,
    &           AZest,Elest,Sigest(2),ISEARCH,IWARN)
    Call Intrp2(Aztbl, Eltbl, Tgain(1,1,3), NAZ,NEL,NAZ,NEL,
    &           AZest,Elest,Sigest(3),ISEARCH,IWARN)
40  Call Intrp2(Aztbl, Eltbl, Tgain(1,1,4), NAZ,NEL,NAZ,NEL,
    &           AZest,Elest,Sigest(4),ISEARCH,IWARN)
    Call Intrp2(Aztbl, Eltbl, Tgain(1,1,5), NAZ,NEL,NAZ,NEL,
    &           AZest,Elest,Sigest(5),ISEARCH,IWARN)
    Call Intrp2(Aztbl, Eltbl, Tgain(1,1,6), NAZ,NEL,NAZ,NEL,
    &           AZest,Elest,Sigest(6),ISEARCH,IWARN)
45  Call Intrp2(Aztbl, Eltbl, Tgain(1,1,7), NAZ,NEL,NAZ,NEL,
    &           AZest,Elest,Sigest(7),ISEARCH,IWARN)

C Normalize retrieved amplitudes from caibration database

50  Sigestmax = DMAX1(Sigest(1),Sigest(2),Sigest(3),Sigest(4),
    &               Sigest(5),Sigest(6),Sigest(7))
    Sigest(1) = Sigest(1)/Sigestmax
    Sigest(2) = Sigest(2)/Sigestmax
55  Sigest(3) = Sigest(3)/Sigestmax
    Sigest(4) = Sigest(4)/Sigestmax
    Sigest(5) = Sigest(5)/Sigestmax
    Sigest(6) = Sigest(6)/Sigestmax
    Sigest(7) = Sigest(7)/Sigestmax

60  C Calculate Residual Component for Edge of FOV

    FOVest=SQRT(AZest**2+ELeSt**2)
    FOVres=0.0
65  IF(FOVest.GT.14.0) FOVres = (FOVest-14.0)/14.0

```

```

C Calculate residual for each point in search grid
C   Res = Signal(1)*(Signal(1)-Sigest(1))**2 +
C   &   Signal(2)*(Signal(2)-Sigest(2))**2 +
5 C   &   Signal(3)*(Signal(3)-Sigest(3))**2 +
C   &   Signal(4)*(Signal(4)-Sigest(4))**2 +
C   &   Signal(5)*(Signal(5)-Sigest(5))**2 +
C   &   Signal(6)*(Signal(6)-Sigest(6))**2 +
C   &   Signal(7)*(Signal(7)-Sigest(7))**2 +
10 C   &   FOVres**2

C Calculate residual for each point in search grid
C   Res = (Signal(1)-Sigest(1))**2 +
15 C   &   (Signal(2)-Sigest(2))**2 +
C   &   (Signal(3)-Sigest(3))**2 +
C   &   (Signal(4)-Sigest(4))**2 +
C   &   (Signal(5)-Sigest(5))**2 +
C   &   (Signal(6)-Sigest(6))**2 +
20 C   &   (Signal(7)-Sigest(7))**2 +
C   &   FOVres**2

      Return
      END
25 CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
C-----

      SUBROUTINE DETECT (BACKPOW, GAIN, TEMP, BW, ZNEP)
30      IMPLICIT DOUBLE PRECISION (A-H, O-Z)

C-----
C
C   THIS SUBROUTINE COMPUTES THE NOISE EQUIVALENT POWER,
35 C   INCLUDING BACKGROUND/BACKSCATTER EFFECTS
C
C   INPUTS -
C
40 C   BACKPOW = BACKSCATTER POWER, WATTS
C   GAIN = APD GAIN
C   TEMP = TEMPERATURE OF AMP AND DETECTOR, KELVIN
C   BW = Effective Noise BANDWIDTH OF FILTER ON AMP, HZ
C
45 C   OUTPUTS -
C
C   ZNEP = NOISE EQUIVALENT POWER, WATTS
C
C   VARIABLE DEFINITIONS -
50 C
C   PI = PI
C   ZIDS = DARK SURFACE CURRENT OF APD AT 25C, AMPS
C   ZIDB = DARK, UNITY GAIN, BULK CURRENT OF APD AT 25C, AMPS
C   RES = FEEDBACK RESISTOR, OHMS
55 C   OP1 = AMP CURRENT NOISE, AMPS/RHZ
C   ENOP1 = AMP VOLTAGE NOISE, VOLTS/RHZ
C   CT = TOTAL CAPACITANCE AT INPUT TO AMP (INCLUDES APD CAP), FARADS
C   ZKEFF = EFFECTIVE IONIZATION RATIO IN APD
C   WAVELENGTH = LIGHT WAVELENGTH, MICRONS
60 C   QE = QUANTUM EFFICIENCY
C   QSELECT = CHARGE ON AN ELECTRON, COULOMBS
C   RNOISE = FEEDBACK RESISTOR NOISE, AMPS
C   ZINOP1 = AMPLIFIER CURRENT NOISE, AMPS
C   ZINENOP1 = AMPLIFIER VOLTAGE NOISE CONVERTED TO CURRENT, AMPS
65 C   FSNR = EXCESS NOISE FACTOR, DIMENSIONLESS

```

```

C          RESP = DETECTOR RESPSONSIVITY, AMPS/WATT
C          ZIDSTC = ZIDS COMPENSATED FOR TEMPERATURE, AMPS
C          ZIDBTC = ZIDB COMPENSATED FOR TEMPERATURE, AMPS
5          C          TOTALN = TOTAL NOISE, AMPS
C          C          THRESH = DETECTION THRESHOLD
C
C-----
10          C          **OPEN FILES*****
C          OPEN(UNIT=12, FILE='OUTPUT.DAT')
C
15          C          **SENSOR CHARACTERISTICS*****
C          ZIDS = 150.E-9
C          ZIDB = 2.0E-12
C          RES = 20.E+3
20          C          OP1 = 1833.0E-15
C          ENOP1 = 1.3e-9
C          CT = 15.0E-12
C          ZKEFF = 0.02
C          WAVEL = 1.064
25          C          QE = 0.36
C-----
C          COMPUTE THE NOISE AND THE THRESHOLD
C-----
30          C          PI=3.141592654
C          QELECT = 1.6E-19
C          RNOISE = SQRT(4.*1.38E-23*TEMP*BW/RES)
C          ZINOP1 = OP1*SQRT(BW)
C          ZINENOP1 = (ENOP1*2.*PI*CT)*SQRT(0.3333*BW*BW*BW)
35          C          FSNR = ZKEFF*GAIN + (2.-1./GAIN)*(1 - ZKEFF)
C          RESP = GAIN*QE*WAVEL/1.24
C          ZIDSTC = ZIDS/(2**((298 - TEMP)/10))
C          ZIDBTC = ZIDB*GAIN/(2**((298 - TEMP)/10))
C          ZIBG = BACKPOW*RESP
40          C          ZIBG NOISE = SQRT(BW*2.*QELECT*ZIBG*GAIN*FSNR)
C          ZINDET = SQRT(BW*2.*QELECT*((ZIBG + ZIDBTC)*
c          C          GAIN*FSNR + ZIDSTC))
C          ZINDETBG = SQRT(BW*2.*QELECT*ZIBG*GAIN*FSNR)
C          ZINDETDB = SQRT(BW*2.*QELECT*ZIDBTC*GAIN*FSNR)
45          C          ZINDETDS = SQRT(BW*2.*QELECT*ZIDSTC)
C          ZINDET = SQRT(ZINDETBG**2 + ZINDETDB**2 + ZINDETDS**2)
C          TOTALN = SQRT(RNOISE**2 + ZINOP1**2 + ZINENOP1**2 + ZINDET**2)
C          ZNEP = TOTALN/RESP
50          C          RETURN
C          END
C*****
55          C*****
C          FUNCTION ZRAND()
C          IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C          ZRAND = RAND()
60          C          IF(ZRAND .EQ. 1) THEN
C              ZRAND = 0.999999999999D0
C          ELSE IF(ZRAND .EQ. 0) THEN
C              ZRAND = 1.E-20
65          C          ENDIF

```

```

END FUNCTION

5 C*****
C*****
  Double Precision Function OPgain3(Theta)
  IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C-----
10 C THIS SUBROUTINE RETURNS OPTICAL GAIN BASED ON INTERPOLATING IN TABULAR DATA
C-----
C DECLARATIONS

  LOGICAL ISEARCH, IWARN

15 COMMON / Gainlens / Nangles, Angles(200), OptTrans(200)

C-----
C-----

20 ISEARCH = .TRUE.
  IWARN = .False.
  CALL INTRP1(Angles,OPTTRANS,200,NANGLES,THETA,OPgain3
    & ,ISEARCH,IWARN)

25 IF(IWARN) THEN
  PRINT*, 'SUBROUTINE OPTICAL DATA OUT OF BOUNDS!', THETA
  WRITE(7,*)'SUBROUTINE OPTICAL DATA OUT OF BOUNDS!', THETA
ENDIF

30 END FUNCTION

C*****
C*****
35 SUBROUTINE SKIPL(IUNIT, NLINES)
  IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C-----
C THIS ROUTINE IS USED TO SKIP LINES IN INPUT FILES
C-----SKIPL-----
40 DO 10 I=1, NLINES
  READ(IUNIT,*)
  10 CONTINUE
  RETURN
  END

45 C*****
C*****
  SUBROUTINE INTRP1(XA, FA, MAXX, NX, X, F, ISEARCH, IWARN)
  IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C*****
50 C**
C** Single interpolation subroutine
C**
C** ARGUMENT DEFINITIONS:
C**
55 C** XA - Array of first independent variable (input)
C** FA - Array of dependent variable (input)
C** X - Value of first independent variable (input)
C** F - Value of dependent variable (output)
C** MAXX - Dimension of XA and first dimension of FA (input)
C** NX - Actual number of values in XA array (input)
60 C** ISEARCH - Logical flag indicating whether it is necessary to search
C** the XA array for the appropriate interpolation
C** range.
C** IWARN - Logical flag indicating whether to warn of X outside
65 C** of range.
C**

```

```

C*****
LOGICAL ISEARCH, IWARN
DIMENSION XA(MAXX), FA(MAXX)
5 SAVE KX1, KX2, FACTX
DATA KX1, KX2 /2*1/

IF (ISEARCH) THEN
10 CALL SEARCH(XA, MAXX, NX, X, KX1, KX2, FACTX, IWARN)
ENDIF

F = FA(KX1) + (FA(KX2) - FA(KX1))*FACTX

15 RETURN
END
C*****
C*****

20 SUBROUTINE INTRP2(XA, YA, FA, MAXX, MAXY, NX, NY,
1 X, Y, F, ISEARCH, IWARN)
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C*****
C**
25 C** Double interpolation subroutine
C**
C** ARGUMENT DEFINITIONS:
C**
C** XA - Array of first independent variable (input)
30 C** YA - Array of second independent variable (input)
C** FA - Array of dependent variable (input)
C** X - Value of first independent variable (input)
C** Y - Value of second independent variable (input)
35 C** F - Value of dependent variable (output)
C** MAXX - Dimension of XA and first dimension of FA (input)
C** MAXY - Dimension of YA and second dimension of FA (input)
C** NX - Actual number of values in XA array (input)
C** NY - Actual number of values in YA array (input)
40 C** ISEARCH - Logical flag indicating whether it is necessary to search
the XA and YA arrays for the appropriate interpolation
range.
C** IWARN - Logical flag indicating whether to warn of X or Y outside
of range.
C**
45 C*****
LOGICAL ISEARCH, IWARN
DIMENSION XA(MAXX), YA(MAXY), FA(MAXX, MAXY)
SAVE KX1, KX2, KY1, KY2, FACTX, FACTY
DATA KX1, KX2, KY1, KY2 /4*1/

50 IF (ISEARCH) THEN
CALL SEARCH(XA, MAXX, NX, X, KX1, KX2, FACTX, IWARN)
CALL SEARCH(YA, MAXY, NY, Y, KY1, KY2, FACTY, IWARN)
ENDIF

55 TEMPX1 = FA(KX1,KY1) + (FA(KX2,KY1) - FA(KX1,KY1))*FACTX
TEMPX2 = FA(KX1,KY2) + (FA(KX2,KY2) - FA(KX1,KY2))*FACTX

F = TEMPX1 + (TEMPX2-TEMPX1)*FACTY

60 RETURN
END
C*****
C*****
65 SUBROUTINE SEARCH(XA, MAXX, NX, X, K1, K2, FACTOR, IWARN)

```

IMPLICIT DOUBLE PRECISION (A-H,O-Z)

C*****

C**
 C** This routine is called by INTRP1, INTRP2, or INTRP3 to search through
 5 C** a vector table, XA, to find the subscripts K1 and K2 such that
 C**
 C** XA(K1) <= X <= XA(K2).
 C**

C** In general,
 10 C**
 C** K2 = K1 + 1
 C**
 C** although the subscripts may be equal if X lies on or outside either end
 C** of the vector.
 15 C** Note that the XA vector must be arranged in increasing order.
 C**

C*****

C**

C** ARGUMENT DEFINITIONS:

20 C**
 C** IWARN - Logical flag indicating whether to warn of X or Y outside
 C** of range.
 C** FACTOR - Ratio (X - XA(K1))/(XA(K2) - XA(K1)) to be used for
 25 C** linear interpolation by the calling routine. If X < XA(1)
 C** or X > XA(NX), then FACTOR = 0 is returned.
 C** K1 - Largest index of XA array having value less than or equal
 C** to X (output)
 C** K2 - Smallest index of XA array having value greater than or equal
 30 C** to X (output)
 C** MAXX - Dimension of XA (input)
 C** NX - Actual number of values in XA array (input)
 C** XA - Vector array of independent variable (input)
 C** X - Value of independent variable (input)
 C**

35 C*****

LOGICAL IWARN
 DIMENSION XA(MAXX)

40 C WRITE(*,*) ' ENTER K2 = ',K2
 C-----
 C-- Check for X outside of range of XA

45 IF (X.LT.XA(1)) THEN
 K1=1
 K2=1
 FACTOR=0.
 IWARN = .TRUE.
 IF (IWARN) THEN
 50 WRITE(*,*) 'X .LT. XA(1) IN SEARCH; X=',X
 WRITE(7,*) 'X .LT. XA(1) IN SEARCH; X=',X
 Read(*,*)
 ENDIF
 GOTO 900
 55 ELSE IF (X.GT.XA(NX)) THEN
 K1=NX
 K2=NX
 FACTOR=0.
 IWARN = .TRUE.
 IF (IWARN) THEN
 60 WRITE(*,*) 'X .GT. XA(NX) IN SEARCH; X=',X
 WRITE(7,*) 'X .GT. XA(NX) IN SEARCH; X=',X
 Read(*,*)
 ENDIF
 65 GOTO 900

ENDIF

C-----

5 C-- Find indices of XA for which $XA(K1) \leq X \leq XA(K2)$

```

    ICOUNT = 0
10  CONTINUE
    IF ( X .LT. XA(K1) ) THEN
10   K1 = K1 - 1
        ICOUNT = ICOUNT + 1
        GOTO 10
    ENDIF

```

```

15   K2 = MINO((K1 + 1), NX)
20  CONTINUE
    C  WRITE(*,*) ' K2 = ',K2
    IF ( X .GT. XA(K2) ) THEN
        K2 = K2 + 1
        ICOUNT = ICOUNT + 1
20   GOTO 20
    ENDIF

```

```

    K1 = MAXO((K2 - 1), 1)
25   FACTOR = (X - XA(K1))/(XA(K2) - XA(K1))

```

900 CONTINUE

```

30  C  WRITE(*,*) '          RETURN K2 = ',K2
    RETURN
    END

```

C*****
C*****

```

35  SUBROUTINE UNIT(A,AU)
    IMPLICIT DOUBLE PRECISION (A-H,O-Z)

```

C*****

```

40  C
    C  FUNCTION: COMPUTE A UNIT VECTOR
    C

```

C*****

```

45  DIMENSION A      ( 3)
    DIMENSION AU     ( 3)

```

```

    C*  AMAG=VECMAG(A)
        DO 10 I=1,3
            AU(I)=A(I)/AMAG

```

```

50  10 CONTINUE
    RETURN
    END

```

C*****
C*****

```

55  SUBROUTINE VECSUB(A,B,R)
    IMPLICIT DOUBLE PRECISION (A-H,O-Z)

```

C*****

```

60  C
    C  FUNCTION: SUBTRACT TWO VECTORS
    C

```

C*****

```

65  DIMENSION A      ( 3)
    DIMENSION B      ( 3)
    DIMENSION R      ( 3)

```

```

      DO 10 I=1,3
        R(I)=A(I)-B(I)
5     10 CONTINUE
      RETURN
      END
C*****

10     SUBROUTINE SCAVEC(S,V,SV,N)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C*****
      C
      C   SCAVEC MULTIPLIES A SCALAR TIMES A VECTOR
15     C
C*****
      DIMENSION V(N),SV(N)

      DO 10 I=1,N
20     SV(I)=S*V(I)
      10 CONTINUE
      RETURN
      END
25     C*****
      C*****

      DOUBLE PRECISION FUNCTION VECMAG(A)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
30     C*****
      C
      C   FUNCTION: COMPUTE A VECTOR MAGNITUDE
      C
35     C*****
      DIMENSION A(3)

      R=0.0D0
      DO 10 I=1,3
40     R=R+ A(I)*A(I)
      10 CONTINUE
      VECMAG = DSQRT(R)
      RETURN
      END
45     C*****
      C*****
      C*****
      C*****
      C*****
      C*****

```

CLAIMS

1. A method for determining angular information from a target, comprising:
 - (a) obtaining measured amplitude data from a plurality of sensors;
 - 5 (b) estimating an azimuth and elevation using the measured amplitude data;
 - (c) obtaining calibrated amplitude data corresponding to the azimuth and elevation;
 - (d) determining a residual error between the measured amplitude data and the calibrated amplitude data;
 - (e) repeating steps (b) through (d) until the residual error is minimized; and
 - 10 (f) outputting the angular information as corresponding to the azimuth and elevation at which the residual error was minimized.

2. The method of claim 1, wherein obtaining measured amplitude data comprises obtaining measured amplitude data from seven sensors.
- 15 3. The method of claim 1, wherein one or more of the sensors comprise a CCD, thermal imaging array, or a quadrant detector.

4. The method of claim 1, wherein obtaining measured amplitude data comprises defocusing of the target.
- 20 5. The method of claim 1, further comprising determining a power centroid of the target using the measured amplitude data.

- 25 6. The method of claim 1, wherein determining the residual error comprises summing differences between the measured amplitude data and the calibrated amplitude data.

7. A computer program, comprising computer or machine readable program elements translatable for implementing the method of claim 1.
- 30 8. An apparatus for performing the method of claim 1.

9. A method for determining angular information from a target, comprising:

- (a) obtaining a plurality of sensors on an image plane;
- (b) defocusing the target or a point target on the image plane such that one or more of the sensors respond to a target pulse;
- 5 (c) obtaining measured amplitude data from the plurality of sensors;
- (d) estimating an azimuth and elevation using the measured amplitude data;
- (e) obtaining calibrated amplitude data corresponding to the azimuth and elevation;
- (f) determining a residual error between the measured amplitude data and the calibrated amplitude data;
- 10 (g) repeating steps (d) through (f) until the residual error is minimized; and
- (h) outputting the angular information as corresponding to the azimuth and elevation at which the residual error was minimized.

10. The method of claim 9, wherein obtaining measured amplitude data comprises obtaining
15 measured amplitude data from seven sensors.

11. The method of claim 9, wherein one or more of the sensors comprise a CCD, thermal imaging array, or a quadrant detector.

20 12. The method of claim 9, further comprising determining a power centroid of the target using the measured amplitude data.

13. The method of claim 9, wherein determining the residual error comprises summing differences between the measured amplitude data and the calibrated amplitude data.

25

14. A computer program, comprising computer or machine readable program elements translatable for implementing the method of claim 9.

15. An apparatus for performing the method of claim 9.

30

16. A system for determining angular information from a target, comprising:
a plurality of sensors on an image plane; and

a computer configured to:

acquire measured amplitude data from the plurality of sensors;

estimate an azimuth and elevation using the measured amplitude data;

access calibrated amplitude data corresponding to the azimuth and elevation;

5

determine a residual error between the measured amplitude data and the calibrated
amplitude data; and

output the angular information as corresponding to the azimuth and elevation at
which the residual error is minimized.

10

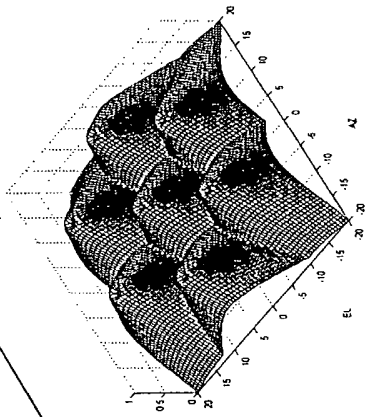
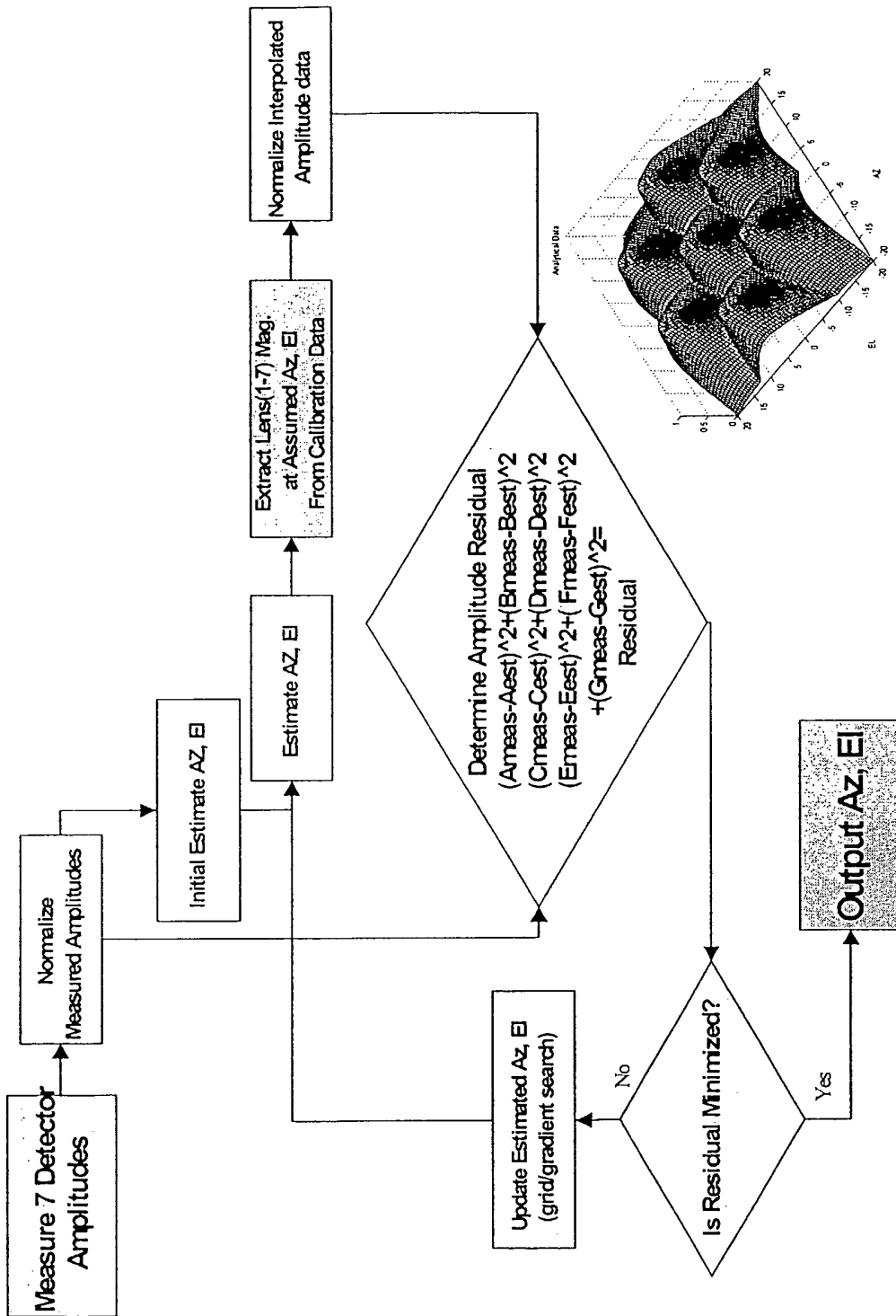


FIG. 1

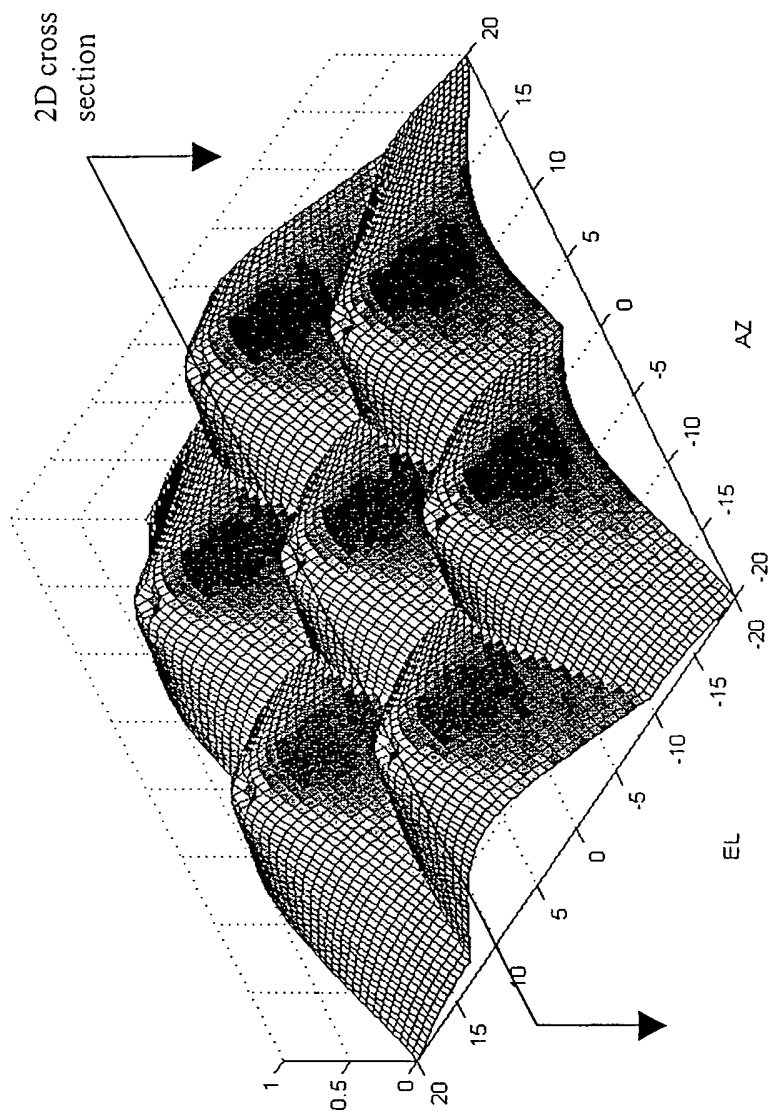


FIG. 2A

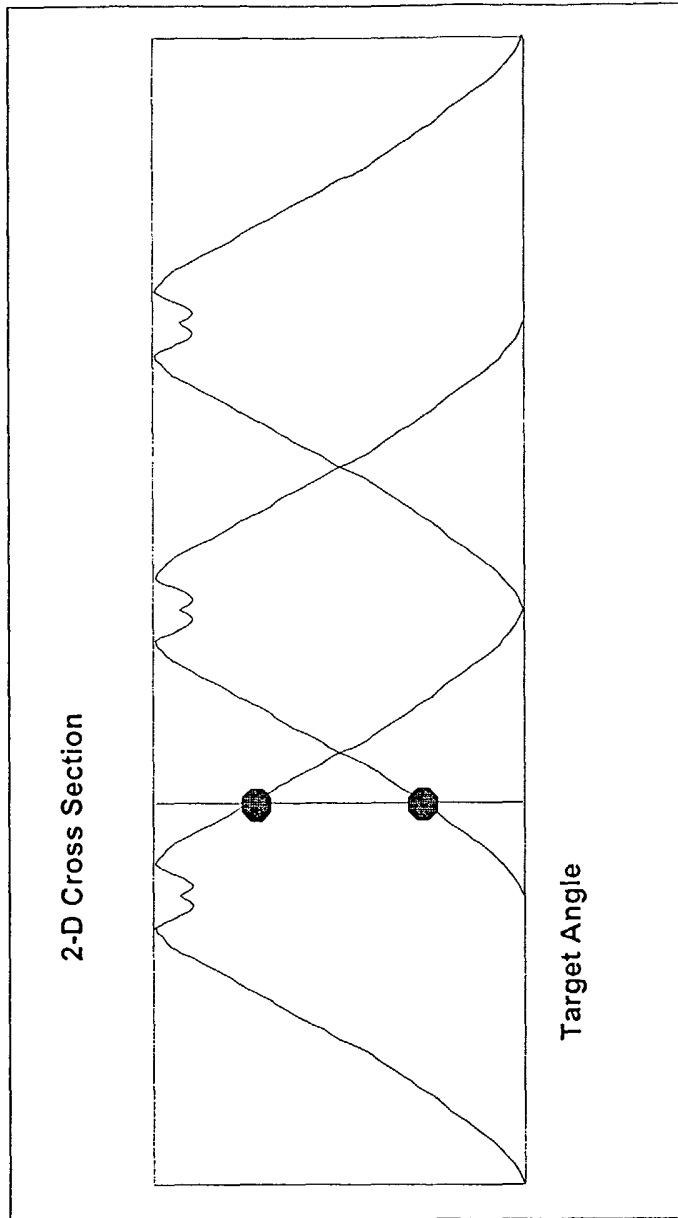


FIG. 2B