ACD
33090/8
Copy 2

DOT/FAA/CT-ACD33090/08

SOFTWARE FOR THE CREATION AND ANALYSIS OF MEASURED FLIGHTPATHS FROM LASER TRACKER DATA

Linda Pasquale

May 1990

SEP 17 Rec'd
TECHNICAL CENTER LIBRARY
ATLANTIC CITY, N.J. 08405

Engineering, Research, and Development Service
Concepts Analysis Division
FAA Technical Center
Atlantic City International Airport, N.J. 08405

TABLE OF CONTENTS

INTRODUCTION

VIEWLAS.V10

LASROT.V10

LASDIF.V10

PLTLASD.V10

APPENDIX A

APPENDIX B

APPENDIX C

REFERENCES

INTRODUCTION

The Microwave Landing System (MLS) Mathematical Model includes a measured flightpath option which enables the model to read flightpath data (X,Y,Z coordinates) from a second input file (separate from the primary formatted input file). A measured flightpath, created from MLS airborne data or tracking system data, represents the path actually flown by an aircraft with greater accuracy and detail than a flightpath calculated mathematically. A measured flightpath is used in modeling scenarios for which actual signal data is available. Output from modeling with a measured flightpath can be compared directly to the "real world" signal data for purposes of evaluating the accuracy of model predictions. This is the procedure used for MLS mathematical model validation studies [1].

At the Federal Aviation Administration (FAA) Technical Center, one of the systems used for tracking aircraft is a mobile laser tracking system. This system generates a tape of time, azimuth, elevation, range, and flag data in binary format. The tape created at the laser site is further processed on a Honeywell computer by a program called LASER. The result is a binary tape in Honeywell format consisting of time and X, Y, Z coordinates of the aircraft relative to the laser [2]. The tape is converted to a VAX disk file by the program EDEL, written by ACD-330 personnel. It reads the data on the Honeywell tape, converts it to VAX format, and writes it to a binary disk file. The disk file consists of time (in seconds) and X, Y, Z coordinates (in feet relative to the laser) for each flightpath point. All flights on the laser tape are included in the disk file. Creation and analysis of measured flightpaths for the MLS math model begins with this VAX disk file. The programs written to accomplish this task are:

1. VIEWLAS - creates an ASCII dump of the binary data in the file produced by EDEL to allow inspection of the "raw" laser data

2. LASROT - converts the X,Y,Z data for a specific run from the laser tracker coordinate system (origin at laser boresite, receiver at laser retro) to the model coordinate system (origin at runway stop end, receiver at MLS antenna). The output from LASROT is a measured flightpath readable by the MLS math model.

3. LASDIF - creates a file containing an angle difference and the distance from runway stop end at which the difference occurred. The "difference" is between the MLS angle and the laser angle for the system being analyzed (AZ or EL).

4. PLTLASD - plots the data from the file created by LASDIF. This "differential error" plot is used for comparison with error plots produced by the MLS math model.

VIEWLAS.V10

The disk file of laser data (time, X, Y, Z) created by the program EDEL is in binary format and is therefore unreadable with a text processor. At times, it is desirable to be able to view the "raw" laser data directly (for example, to observe the effect on the laser of a known timed event such as the departure of an aircraft from a nearby runway). The program VIEWLAS creates an ASCII dump of the binary EDEL file.

VIEWLAS queries the user for the name of the input file containing binary laser data. It also asks for the name of an output file to which the ASCII data is written. If no output file is named, the ASCII data is written to the display screen. Since the input data is binary, it is read unformatted. The format of the output is 4(1X,F12.4).

In addition, VIEWLAS creates a file (TIMES.DAT) containing the start and stop time of each "run". Run is in quotes here because VIEWLAS has no direct way of distinguishing one run from another. Instead, it reports any gap in time greater than one second as the start of a new run. The laser tracking system is supposed to produce coordinate triplets every .1 second. However, this data rate is not always maintained. By comparing the data in TIMES.DAT with the times recorded in the flight logs, the analyst can determine where unacceptably large gaps in laser data occur during an individual run.

LASROT.V10

Once an acceptable span of laser data has been chosen, it is converted to a measured flightpath by the program LASROT. The steps taken by LASROT are:
1. read every other point in the laser file between the specified start and stop times to achieve the .2 second data rate required by the MLS math model.
2. translate and rotate each laser point to refer to the stop end of the runway (the origin used by the MLS math model).
3. adjust the receiver coordinates to refer to the MLS antenna rather than the laser retro.

To accomplish its task, LASROT requires an input file (*.INF) containing the following information:
1. the name of the file produced by EDEL containing "raw" laser data, specifically time in seconds and X, Y, Z coordinates in feet relative to the laser. (The file name is 80 characters.)
2. the name of an output file to which the corrected coordinates are written in a format readable by the MLS math model (80 characters).
3. a title written to the output file to identify its contents (80 characters).
4. a start time and stop time (in seconds) for the run (complete or partial) being processed (both real*4).
5. the latitude of the airport, used for calculating the radius of the earth which is required to determine angles of rotation for X and Y axes (real*4).
6. X, Y, Z axis translations. The translations (in feet) are defined as the coordinates of the laser tracker in the model coordinate system (three real*4).
7. the angle for rotation around the Z axis (in degrees - real*4). (See appendix A for the method of determining this angle.)

Once LASROT has read the data in the input file, it calculates the radius of the earth at the airport latitude using the following formula:

$$\text{radius} = (J * K) / \text{sqrt}\,[(J * \sin(\text{lat}))^2 + (K * \cos(\text{lat}))^2]$$

where:

J =	20925874.69	is the earth radius at the equator in feet
K =	20854934.39	is the earth radius at the poles in feet
lat		is the latitude of the airport in degrees

This radius is used to calculate the angles of rotation around the X and Y axes using the formula for the arc of a circle as follows:

$$S = R * A$$

where:

S is the length of the arc along the earth surface. The Y translation is used for this in the case of the X axis rotation, X translation for Y axis rotation. For an explanation of signs, see appendix B.
R is the radius of the earth at the airport latitude.
A is the angle being determined, in radians.

The coordinate conversion is accomplished by rotating around the three axes using the angles determined by the method described above and translating by adding the translation values read in from the input file. The three rotation matrices [3] are used with the following assumptions:

1. Both the source (laser) coordinate system and the target (model) coordinate system are right handed systems. In the airport environment, the positive X axis of the laser coordinate system points due east. The positive X axis of the model coordinate system runs along runway centerline from stop end to threshold.

2. If the angle of rotation is positive, the point is rotated around the axis in a counter-clockwise direction looking from the positive axis toward the origin. If negative, the rotation is clockwise.

3. The order of rotation is Z axis, X axis, Y axis. This rotation order is used to be consistent with the practice of other projects in ACD-330.

When the laser coordinates have been rotated and translated to the model origin, they represent the position of the laser retro relative to the stop end of the runway. However, the MLS math model assumes that the position of the aircraft refers to the MLS antenna. Therefore, before the measured flightpath is complete, each point must be corrected to refer to the MLS antenna rather than the laser retro.

The correction algorithm makes no assumptions about the geometry of the aircraft or the positions of the laser retro and MLS antenna. This information is hard coded into LASROT for each aircraft. The dimension corrections needed for N-91, a Convair 580 used to collect airborne data, are:

X = 10.21 feet
Y = 0.0 feet
Z = -4.465 feet

Other dimensions for other aircraft or receiver positions can be substituted as required. The algorithm used to determine the correction factors is found in subroutine ACCOR and is discussed in detail in appendix C. When the correction factors for each coordinate are added to the rotated/translated coordinate, the result is the position of the MLS antenna relative to the stop end of the runway. These data, along with the time of each

flightpath point, are written to the output file using the format F12.3. The output file is a measured flightpath file which can be used directly by the MLS math model because it is consistent with the following assumptions:

1. All coordinates are in feet relative to the stop end of the runway.
2. All coordinates refer to the MLS antenna on the aircraft.
3. The coordinates are presented at a data rate of .2 second (assuming a laser data rate of .1 second without gaps).

LASDIF.V10

LASDIF creates a file of distance/difference data pairs. The "difference" is that between the AZ (or EL) angle as measured by the MLS and the angle as measured by the laser tracking system. The MLS angle is read from an input file. The angle at the laser is determined from the laser coordinate data. The "distance" is the point on the flightpath at which the difference occurs relative to the azimuth transmitter. It is determined from the laser coordinate data using the same method as LASROT. The difference data is filtered by a path following error (PFE) filter algorithm. Plots of these data (made by PLTLASD) are comparable to error plots from the MLS math model and can be used to evaluate the accuracy of the model.

LASDIF requires a control file (*.LCF) which provides the following information to the program:
1. the name of the file (created by EDEL) containing laser data (80 characters).
2. the name of the file (created by REDUCE [4]) containing MLS/RNAV data (80 characters).
3. the name of the output file to which the final results are written (80 characters).
4. a title written to the output file for inclusion on later plots (80 characters).
5. the flightpath type - orbit, approach, or radial (1 character). This determines whether a distance or angle (orbit) value is paired with the difference.
6. the start time and stop time of the run (in seconds - real*4). The start time must be higher than the lowest time of both the laser and MLS data to ensure that data from both systems is available for the difference calculation.
7. the latitude of the airport (in degrees - real*4). This is used in the same way as in LASROT.
8. translations for each of the three axes for converting from the laser to model coordinate systems (in feet - three real*4).
9. angles for Z axis rotation for converting (1) from the laser to model coodinate systems and (2) from the laser to transmitter coordinate systems (in degrees - two real*4).
10. coordinates of the azimuth transmitter, in feet relative to the stop end of the runway (three real*4). These data are used in the calculation of the angle at the transmitter.
11. coordinates of the elevation transmitter (as above).
12. an integer indicating which MLS receiver system is to be processed (integer: 1-AZ system 1, 2-AZ2, 3-EL1, 4-EL2).

LASDIF queries the user for the name of the control file containing the above information. All numeric data are read in free format.

Two subroutines determine the constants and trigonometric functions that will be required by the program. The values are returned through COMMONs. Subroutine CALCON initializes a number of constants such as that to convert feet to nautical miles,

degrees to radians, and radians to degrees. In addition, it calculates the earth radius using the same method as LASROT. It uses this value to determine the angles of rotation around the X and Y axes for converting from laser to model coordinates and for converting from laser to transmitter (system) coordinates (all angles in radians). The former is used to determine the distance (or azimuth angle). The latter is used to determine the angle at the transmitter for the difference. The subroutine also determines the translation required for the laser-to-system conversion. These translation values are defined as the coordinates of the laser tracker (in feet) relative to the transmitter.

Subroutine CALTRG uses the angles returned by CALCON to set up two sets of rotation matrices: one for the laser-to-model conversion required for the distance (or AZ angle) calculation; one for the laser-to-system conversion required to determine the angle at the laser for the difference calculation. The rotation matrices are used in the same way, and with the same assumptions, as in LASROT.

Once the necessary constants and functions have been determined, LASDIF completes the remaining tasks in four loops. The first loop is the "laser loop". This section of the program reads the laser coordinates from the laser input file, translates and rotates them to the model origin, and corrects them to refer to the MLS antenna using the same methods as LASROT. These coordinates are then converted to a distance value (for approaches and radials) or an azimuth angle (for orbits) as follows:

$$\text{distance} = \text{sqrt}\ [(x - azx)^2 + (y - azy)^2]$$

$$\text{angle} = -\ \text{arctangent}\ [(y - azy) / (x - azx)]$$

where:

x, y are the coordinates of the flightpath point in the model coordinate system

azx, azy are the coordinates of the azimuth transmitter in the model coordinate system

The distance value is converted from feet to nautical miles. The azimuth angle value is converted to radians. It is negative to conform with MLS practice. That is, an angle rotating from the X axis to the Y axis (counter-clockwise as viewed from the positive Z axis) is considered positive by mathematical convention (and functions such as the arctangent function) but negative by MLS conventions. The distance (or angle) values thus calculated provide the X axis range for plots of these data produced by PLTLASD (discussed in the next section). They match the X axis values of the model error plots so that the two plots can be directly compared.

Within the laser loop, the "raw" laser coordinates are also

converted to an angle value relative to the system for which the difference is being calculated (as indicated in the control file). This is accomplished as follows:

1. The laser coordinates are rotated and translated to the transmitter coordinate system assuming a right handed system with the origin at the transmitter boresite and the positive X axis parallel to runway centerline.
2. The rotated coordinates are corrected for the offset on the aircraft between the laser retro and the MLS antenna (see appendix C).
3. The corrected coordinates are converted to an angle as follows:

AZ angle = -arctangent (y / x) (MLS convention)

EL angle = arctangent [z / sqrt(x^2 + y^2)]

where:
x, y, z are the rotated, translated, corrected flightpath point coordinates relative to the transmitter

These angle values, in degrees, are tested against validity limits (+/- 60° for azimuth, 0 to 20° for elevation) and are discarded if they do not fall within these limits. At the end of the laser loop, arrays hold a distance (or AZ angle) value and an angle at the transmitter for each laser time reading. As with LASROT, every other laser point is read to achieve a data rate of .2 second.

Next, LASDIF executes an "MLS loop". In this loop, data is read from the MLS data file named in the LASDIF control file. Specifically, the loop reads a time value (in the format HH:MM:SS.SSS, all integers), a flag value (integer), a data value (real*4), and a record indentifier (integer) which indicates the type of data value that has been read (AZ angle, EL angle, RTT angle, DME distance). If the identifier does not equal the system identifier read in from the control file, the record is discarded. Otherwise it is processed as follows:

1. The time value is converted to seconds.
2. The angle value is tested against validity limits (same as for laser angle).
3. If the angle data is valid, angle and time data are both stored. Otherwise, they are discarded.

Since the laser data have been reduced to five readings per second and the MLS systems read data thirteen times (AZ) or thirty-nine times (EL) per second, the MLS data must also be reduced. This is done by the "interpolation loop". For each time reading in the laser time array, an MLS angle value is interpolated used the MLS data and time arrays. The INTERP subroutine that accomplishes this is described in detail in

Pasquale [4].

At the end of the "interpolation loop", the program has a distance in nautical miles (or AZ angle in degrees), a laser angle in degrees, and a system angle in degrees for each time in the laser time array. In the final processing stage, the "output loop", the difference is calculated (MLS angle minus laser angle, to conform to MLS math model practice). The difference is filtered by a PFE algorithm, described in detail in Pasquale [4], and the distance and difference are written to the output file using the format (1X,F8.3,1X,F10.7). A title for the run and the flightpath type are also written to the output file for use by PLTLASD. The data in the output file, when plotted, will produce a PFE filtered difference (error) value against the distance from azimuth antenna (or AZ angle) at which the difference occurred. This plot is directly comparable to the PFE error plot produced by the MLS math model.

PLTLASD.V10

As a final stage in the analysis of "real world" laser data, the differential error is plotted by PLTLASD. The program requires two input files which are identified interactively by the user:

1. the file produced by LASDIF containing the difference and distance (or angle) values.
2. a title file containing information for plot identification, specifically:
   a. an airport name (40 characters).
   b. a runway ID (5 characters).
   c. the ID number of the laser tape (10 characters).
   d. the date of the run (8 characters as mm/dd/yy).
   e. the run number (2 characters).

Once all identifiers have been initialized, PLTLASD reads the difference data and plots it using graphics software developed by personnel in the MLS math modeling program. The plot axis values and limits correspond to those used by the MLS math model and are depended on angle system and flightpath type. Specifically, the error is plotted along the Y axis from -0.3° to 0.1°. Approaches and radials both use an X axis distance measured in nautical miles from the azimuth antenna (from 1.0 nmi in increments of 1 mile for approaches and 1.5 miles for radials) while orbit flightpaths are plotted against the azimuth angle in degrees (from -45° to +45°).

APPENDIX C - METHOD OF DETERMINING ANGLE OF ROTATION FOR Z AXIS

The angle of rotation around the Z axis (for any calculation) is read in to the program from the input control file. This angle is determined using the formula for an angle between two vectors:

angle = arcosine [(A • B) / (Al * Bl)]

where:

A and B are directional vectors defining the positive X axis of the source and target coordinate systems respectively

A • B represents the dot product of these two vectors

Al * Bl represents the product of the lengths of the vectors

An example will serve to illustrate this method.

Problem: determine the angle of Z axis rotation for converting from the laser coordinate system to the model coordinate system on runway 13 at the FAA Technical Center

1. The positive X axis of the laser system points due east and can therefore be desribed by the directional vector A = (1,0)

2. The positive X axis of the model system runs along centerline from stop end of runway 13 (threshold of runway 31) to the threshold. This is defined as a directional vector by subtracting the coordinates of the stop end from the coordinates of the threshold. Both sets of coordinates are defined relative to the laser so that the two directional vectors are defined with respect to a common origin. Thus:

B = (-5862.73,299.88) - (2970.56,-4388.69)
= (-8833.29,4688.57)

3. The dot product of the vectors is

$\Sigma A_i B_i$ for i=1,N

A • B = (1 * -8833.29) + (0 * 4688.57) = -8833.29

4. The length of a vector is the square root of the sum of its elements squared.

Al = sqrt [(1 * 1) + (0 * 0)] = 1

Bl = sqrt [$-8833.29^2 + 299.88^2$] = 10000.48503

5. The product of these lengths is 10000.48503.

6. The cosine of the angle between these systems is the dot product divided by the product of the lengths or

$$\cos(\text{theta}) = -8833.29 / 10000.48503 = -.883286158$$

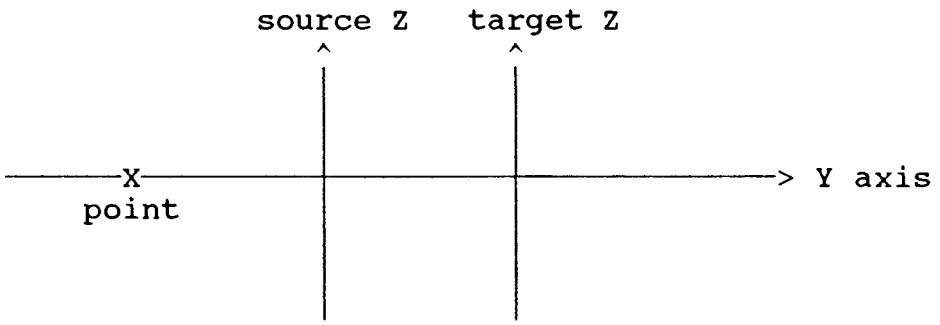
7. Theta is the arc cosine of -.883286158 which is 152.0413476°. If the X axis of the laser system must be rotated +152° to become the X axis of the model system, a point on the X axis (laser system) must be rotated -152° to have the correct relationship to the X axis (model system). This is the angle used for Z axis rotations in this scenario.

APPENDIX B - THE SIGN OF THE ANGLE OF ROTATION FOR THE X (Y) AXIS

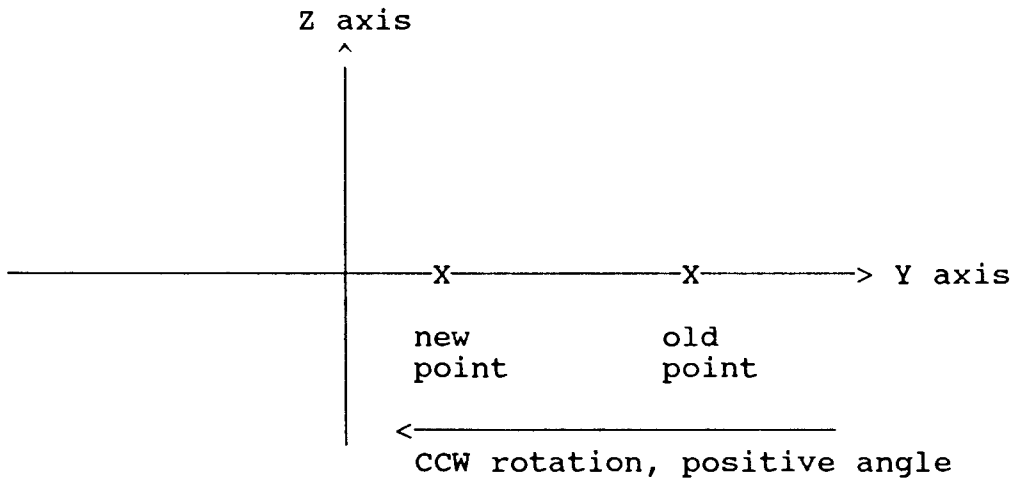
Both LASROT and LASDIF calculate the angles of rotation around the X and Y axes (for all transformations) using the formula for the length of an arc: arc = angle * earth radius. The "arc" is defined as the translation along the required axis. The translation is defined as the coordinate (X or Y as appropriate) of the source origin relative to the target origin. This formula will yield the correct magnitude for the angle of rotation but does not, necessarily, give a sign for the angle consistent with the assumptions of the rotation matrices (discussed above in the section on LASROT). For rotation around the X axis, the sign of the angle is opposite the sign of the Y translation. Therefore, the formula must be adjusted accordingly. On the other hand, the sign of the Y axis angle is the same as that of the X translation so no adjustment is required. This problem is illustrated below.

<u>X axis rotation</u>

Assume the translation is negative (X axis points toward the viewer). This means that the source origin has a negative Y coordinate in the target system.



This means that the Y coordinate of the point must become more negative with respect to a static origin. This is a counter-clockwise rotation around the X axis, i.e., a positive angle.



The opposite case applies if the translation is positive. The point will have to move to a more positive Y position which defines a clockwise rotation or a negative angle.

Y axis rotation

In the case of a Y axis rotation, the rotation angle and the translation have the same sign. Therefore, the formula can be used "as is".

Assume that the translation is negative (Y axis points towards the viewer).

```
                    target Z    source Z
                        ^           ^
                        |           |
                        |           |
                        |           |
X axis <----------X-----+-----------+-----------------------
                point   |           |
                        |           |
                        |           |
                        |           |
```

This means that the X coordinate of the point will become more negative with respect to a static origin. This is a clockwise rotation around the Y axis, i.e., a negative angle.

```
                                Z axis
                                  ^
                                  |
                                  |
                                  |
              old      new        |
             point    point       |
X axis <-------X--------X---------+-----------------------
                                  |
                                  |
        ------------------->      |
CW rotation, negative angle       |
                                  |
                                  |
```

APPENDIX C - METHOD OF CORRECTION FOR AIRCRAFT RECEIVER POSITION

This algorithm was developed with the help of Terry Frink.

When the coordinates produced by the laser tracking system have been rotated and translated, they represent the position of the aircraft with respect to the target origin. Specifically, the coordinates refer to the position of the laser retro on the aircraft. However, the MLS math model assumes that the position of the aircraft refers to the MLS receiver. Therefore, an additional adjustment must be made to the aircraft coordinates to refer to the MLS receiver. Subroutine ACCOR returns three values, DELTAX, DELTAY, and DELTAZ, which are added to the X, Y, Z coordinates respectively to accomplish this adjustment. It calculates the delta values by:

1. defining an aircraft coordinate system (right-handed) as three unit vectors relative to the target origin.

2. combining these three directions with the magnitude of the offset required in each direction. These magnitudes are hard-coded in the array OFFS for the specific aircraft being used to collect airborne data.

DEFINING THE AIRCRAFT COORDINATE SYSTEM

The aircraft coordinate system is defined with the following restrictions:

1. The aircraft is traveling in the direction of its nose. The algorithm does not allow for yaw.

2. The pitch of the aircraft is assumed to lie within, but not including, ± 90° of horizontal.

3. The algorithm does not allow for roll.

Within these restrictions, an aircraft coordinate system is defined relative to the target origin as follows:

1. The positive X axis points toward the nose of the aircraft, that is in the direction of travel.

2. The positive Z axis is orthogonal to the X axis and points "up", that is from belly to roof.

3. The positive Y axis is normal to the X-Z plane (parallel to the wings) and points to the left wing (from pilot's point of view) to be consistent with a right-handed system.

DEFINITION OF THE X AXIS - NU

NU is a unit vector that defines the direction of the X axis of the aircraft system relative to the target origin. It is calculated from the vector U that defines the direction of movement of the aircraft. That is:

$U = P2 - P1$ where P2 is the current aircraft position
P1 the previous aircraft position

$NU = U / |U|$

$|U| = \text{sqrt}[\ (x2-x1)^2 + (y2-y1)^2 + (z2-z1)^2\ ]$

So:

$NUx = (x2 - x1) / \text{sqrt}[\ (x2-x1)^2 + (y2-y1)^2 + (z2-z1)^2\ ]$

$NUy = (y2 - y1) / \text{sqrt}[\ (x2-x1)^2 + (y2-y1)^2 + (z2-z1)^2\ ]$

$NUz = (z2 - z1) / \text{sqrt}[\ (x2-x1)^2 + (y2-y1)^2 + (z2-z1)^2\ ]$

DEFINITION OF THE Z AXIS - NW

NW is a unit vector that defines the direction of the Z axis of the aircraft system relative to the target origin. The definition of a unit vector, the definition of the Z axis, and the restrictions outlined above allow the determination of several useful constraints on the vector NW:

1. The restriction against aircraft roll implies that NW (and NU) lies in the same plane as the "up" vector (0,0,1).

2. NW is orthogonal to NU. That is, $NU \cdot NW = 0$.

3. NW is a unit vector. Therefore, $|NW| = 1$.

4. The restriction of aircraft pitch to ± 90° (not inclusive) and the definition of the positive Z axis as going from belly to roof means that the third component of the NW vector will be positive. That is, $NWz > 0$.

The NW vector is determined by the following procedure.

Applying constraint 1, NW can be defined as a linear combination of NU and (0,0,1). That is, there exist two real numbers, a and b, such that

$NW = (a * NU) + (b * (0,0,1))$ equation 1

The reals, a and b, must be expressed in known terms. So, using equation 1, the second constraint can be expressed as

$$NU \cdot [(a * NU) + (b * (0,0,1))] = 0$$

$$NU \cdot [(a * NU) + (0,0,b)] = 0$$

$$[a * (NU \cdot NU)] + [NU \cdot (0,0,b)] = 0$$

$$(a * |NU|^2) + (NUz * b) = 0$$

But since NU is a unit vector with a length of 1,

$$a + (b * NUz) = 0$$

$$a = -b * NUz \qquad \text{equation 2}$$

Substituting equation 2 into equation 1,

$$NW = [(-b * NUz) * NU] + [b * (0,0,1)]$$

$$NW = b * [(0,0,1) - (NUz * NU)]$$

$$NW = b * [(0,0,1) - (NUz*NUx,\ NUz*NUy,\ NUz*NUz)]$$

$$NW = b * (-NUz*NUx,\ -NUz*NUy,\ 1-NUz*NUz) \qquad \text{equation 3}$$

Using constraint 3,

$$|NW| = |b * (-NUz*NUx,\ -NUz*NUy,\ 1-NUz*NUz)| = 1$$

So:

$$|b| * |(-NUz*NUx,\ -NUz*NUy,\ 1-NUz*NUz)| = 1$$

$$|b| * \text{sqrt}(\ NUz^2*NUx^2 + NUz^2*NUy^2 + (1-NUz^2)^2\ ) = 1$$

$$|b| * \text{sqrt}(\ NUz^2*NUx^2 + NUz^2*NUy^2 + 1 - 2NUz^2 + NUz^4\ ) = 1$$

$$|b| * \text{sqrt}(\ NUz^2 * (NUx^2 + NUy^2 + NUz^2) + 1 - 2NUz^2) = 1$$

$$|b| * \text{sqrt}(\ NUz^2 * (1) + 1 - 2NUz^2) = 1$$

$$|b| * \text{sqrt}(\ NUz^2 + 1 - 2NUz^2\ ) = 1$$

$$|b| * \text{sqrt}(\ 1 - NUz^2\ ) = 1$$

$$|b| = 1 / \text{sqrt}(\ 1 - NUz^2\ )$$

The real, b, can now be expressed in known terms:

$$b = \pm\ 1 / \text{sqrt}(\ 1 - NUz^2\ ) \qquad \text{equation 4}$$

The question is, which value for b is the one that expresses the linear combination of equation 1? Investigation of the component NUz will shed light on this question.

NUz is the third component of the unit vector NU. Since NU is a

unit vector,

$$-1 \leq NUz \leq +1$$

In fact, NUz cannot be equal to -1 or +1 as this would violate the restriction (#2 above) of the pitch of the aircraft to within ± 90° not inclusive (since the resulting NU vector would be (0,0,-1) or (0,0,1)). Therefore,

$$-1 < NUz < +1$$

Exploring the properties of this inequality:

$$0 \leq NUz^2 < +1$$

$$-1 < -NUz^2 \leq 0 \qquad \text{negating}$$

$$(-1+1) < (-NUz^2+1) \leq (0+1) \qquad \text{adding}$$

$$0 < (1-NUz^2) \leq +1$$

$$0 < (1-NUz^2) \;/\; sqrt(1-NUz^2) \qquad \text{dividing; equation 5}$$

This last relationship will be useful in evaluating the use of b in equation 3. That is:

$$NW = \pm\, 1 \;/\; sqrt(1-NUz^2) * (-NUz*NUx,\ -NUz*NUy,\ 1-NUz*NUz)$$

Specifically, the Z component of NW is

$$NWz = \pm\, 1 \;/\; sqrt(1-NUz^2) * (1-NUz^2)$$

By constraint #4 in the definition of the Z axis, NWz > 0. Therefore, the positive solution for b (equation 4) is the one that correctly expresses the linear combination of equation 1 since equation 5 demonstrates that the positive solution is consistent with all other requirements. That is

$$0 < 1 \;/\; sqrt(1-NUz^2) * (1-NUz^2)$$

but

$$0 > -1 \;/\; sqrt(1-NUz^2) * (1-NUz^2)$$

So, using the postive solution for b, NW can be determined as follows:

$$NWx = (-NUz*NUx) \;/\; sqrt(1-NUz^2)$$

$$NWy = (-NUz*NUy) \;/\; sqrt(1-NUz^2)$$

$$NWz = (1-NUz^2) \;/\; sqrt(1-NUz^2) = sqrt(1-NUz^2)$$

DEFINITION OF THE Y AXIS - NV

NV is a unit vector that defines the direction of the Y axis of the aircraft system relative to the target origin. Since NV is a normal to the plane defined by NU and NW, it can be calculated as the cross product of NU and NW. That is:

$$NVx = (NUy * NWz) - (NUz * NWy)$$

$$NVy = (NUz * NWx) - (NUx * NWz)$$

$$NVz = (NUx * NWy) - (NUy * NWx)$$

DETERMINATION OF THE DELTA FACTORS

The correction factors to be added to the coordinates of the current flightpath point can now be determined as a combination of the three direction vectors NU, NV, NW and the three offset magnitudes hard-coded in the array OFFS. That is:

$$(DELTAX, DELTAY, DELTAZ)$$

$$= (OFFS(1), OFFS(2), OFFS(3)) \cdot \begin{vmatrix} NUx & NUy & NUz \\ NVx & NVy & NVz \\ NWx & NWy & NWz \end{vmatrix}$$

So:

$$DELTAX = OFFS(1)*NUx + OFFS(2)*NVx + OFFS(3)*NWx$$

$$DELTAY = OFFS(1)*NUy + OFFS(2)*NVy + OFFS(3)*NWy$$

$$DELTAZ = OFFS(1)*NUz + OFFS(2)*NVz + OFFS(3)*NWz$$

These are the values returned by subroutine ACCOR.

REFERENCES

[1] Linda Pasquale and Jesse D. Jones, Approach to Validation of the MLS Mathematical Model", Concepts Analysis Division Report, ACD-330-90-04

[2] K. T. Culbertson, "A General Description of the NAFEC Laser Software Program/LASER/LASER.0", January 31, 1980, prepared for FAA by Lockheed Electronics Company under contract DOT-FA74NA-1027

[3] J. D. Foley and A. Van Dam, Fundamentals of Interactive Computer Graphics, Reading, Massachusetts: Addison-Wesley Pub. Co., 1982. See chapter 7, "Geometric Transformations".

[4] Linda Pasquale, "Data Reduction and Graphics Software for Analysis of Airborne MLS Data", Concepts Analysis Division report, ACD-330-90-02.