NASA Technical Memorandum 86726

Digital Program for Calculating Static Pressure Position Error

J. Blair Johnson, Terry J. Larson, and Jules M. Ficke

(NASA-TM-86726) DIGITAL PROGRAM FOR CALCULATING STATIC PRESSURE POSITION ERROR (NASA) 28 p CSCL 01C

N87-16821

Unclas

G3/05 44004

February 1987

NASA
National Aeronautics and
Space Administration

NASA Technical Memorandum 86726

Digital Program for Calculating Static Pressure Position Error

J. Blair Johnson, Terry J. Larson, and Jules M. Ficke
Ames Research Center, Dryden Flight Research Facility, Edwards, California

1987

National Aeronautics and
Space Administration
Ames Research Center
Dryden Flight Research Facility
Edwards, California 93523-5000

SUMMARY

A computer program written to calculate the static pressure position error of airspeed systems contains five separate methods for determining position error, of which the user may select from one to five at a time. The program uses data from both the test aircraft and the ground-based radar to calculate the error. In addition, some of the methods require rawinsonde data or an atmospheric analysis, or both.

The program output lists the corrections to Mach number, altitude, and static pressure that are caused by position error. Reference values such as angle of attack, angle of sideslip, indicated Mach number, indicated pressure altitude, stagnation pressure, and total temperature are also listed.

INTRODUCTION

The measurement of true airspeed and altitude is extremely important when flight testing aircraft. To increase the accuracy of these measurements, the static pressure position error must be determined, then accounted for. Though many methods have been developed and used successfully, the data analysis can be tedious. To decrease the workload, a computer program (PERROR) was written that contains several methods (all incorporating radar measurements) for determining static pressure position error for a given air data system. The program accesses data from a file that contains ground-based radar and test aircraft information. Additionally, atmospheric analysis data are entered in table form if available. PERROR then calculates the position error for the system.

This report describes the methods that are used in the program, the inputs necessary for each method, and the limitations of the methods. In addition, the output from an actual test is presented.

NOMENCLATURE

ALPHANB	angle of attack, deg
BETANB	angle of sideslip, deg
CP	pressure coefficient, (PICT - P)/QCIT
DANG	difference between azimuth angle and direction of pressure gradient
DESCP	descent pressure method
DESCT	descent temperature method
DHP	calculated difference between true and indicated pressure altitude
DHPG	difference between geometric and pressure altitude as a function of horizontal position
DM	calculated difference between true and indicated Mach number

DPR	calculated difference between true and indicated static pressure
DR	distance from radar site to aircraft, ft
DZ	constant adjustment to radar altitude (ft) that allows the computed pressure altitude at the reference point to equal the calibrated pressure altitude; used if KK ≠ 0 and LL ≠ 0
DZEN	difference between geometric altitude and indicated pressure altitude as a function of elevation angle along a survey run
DZES	difference between geometric altitude and indicated pressure altitude at the maximum elevation angle along a survey run
DZH	difference between geometric and pressure altitude as a function of geometric altitude
DZHTABL	table of pairs of DZH as a function of altitude; used if KK ≠ 0 and LL ≠ 0
E	radar antenna elevation angle, deg
FLIGHT	flight number
G	pressure gradient obtained from atmospheric analysis, ft/n.mi.
GGHTBL	table of triples of G and GH as a function of altitude; used if KK ≠ 0 and ISURVEY = 0
GH	direction of the pressure gradient (direction of decreasing Z - HP at constant geometric altitude, measured in degrees from true north)
HP	true pressure altitude, ft
HPREF	reference pressure altitude, ft; used if MM ≠ 0
HPT	test or indicated pressure altitude, ft
h	geopotential altitude, ft
IETSV	end time (HR, MIN, SEC, MSEC) of the survey run; used if ISURVEY ≠ 0
IET	stop time (HR, MIN, SEC, MSEC)
IETAD	end time (HR, MIN, SEC, MSEC) of the acceleration-deceleration run associated with a survey run; used if ISURVEY ≠ 0
II	flag set to nonzero to calculate position error using radar-rawinsonde method
IST	start time (HR, MIN, SEC, MSEC)
ISTAD	start time (HR, MIN, SEC, MSEC) of the acceleration-deceleration run associated with a survey run; used if ISURVEY ≠ 0

ISTSV start time (HR, MIN, SEC, MSEC) of the survey run; used if ISURVEY ≠ 0

ISURVEY set to nonzero for a survey run (a survey run is used only when KK ≠ 0)

IUNITS set to 0 indicates standard English units for all inputs; set to 1 indicates pressure units are in lb/in^2 and other units are in standard English units; set to 2 indicates pressure is in N/m^2 and TT is in kelvins

KK flag set to nonzero to calculate position error using level acceleration-deceleration method

LEVD level acceleration-deceleration method

LL flag set to nonzero to calculate position error using descent pressure method

M Mach number

MI indicated Mach number

MM flag set to nonzero to calculate position error using descent temperature method

NDZH two times the number of pairs of entries in the DZH table; used if KK ≠ 0 or LL ≠ 0

NGGH three times the number of triples of entries in the GGH table; used if KK ≠ 0 and ISURVEY = 0

NN flag set to nonzero to calculate position error using total temperature method

OO flag set to 1 to compute PT from QCIT and PICT from the merged file; set to 2 to compute PT from VCT and HPT from the merged file; default of 0 will use PT as it comes from the merged file

P ambient static pressure, lb/ft^2

PICT indicated static pressure, lb/ft^2

PR ambient pressure determined from rawinsonde balloon measurements, lb/ft^2

PT indicated total pressure, lb/ft^2

QCIT indicated impact pressure, lb/ft^2

QQ flag set to nonzero to output pressure coefficient CP; default of 0 will output pressure altitude correction DHP

RRA radar parameter identification for azimuth angle, deg

RRH	radar file parameter identification for geometric altitude, ft above sea level
RRTA	radar file parameter identification for ambient temperature, °F
RRX	radar file parameter identification for x coordinate, centered at station, positive east, ft
RRY	radar file parameter identification for y coordinate, centered at station, positive north, ft
RUN	run number
T	ambient temperature, °F
TOT	total temperature method
TS	standard temperature, °F
TT	total or stagnation temperature, °F
VCT	calibrated airspeed, knots
Z	geometric altitude, ft
γ	specific heat ratio

Suffixes:

DP	output calculated using the descent pressure method
DT	output calculated using the descent temperature method
LD	output calculated using the level acceleration-deceleration method
R	output calculated using the radar-rawinsonde method
TT	output calculated using the total temperature method

DESCRIPTION OF METHODS

Level Acceleration-Deceleration Method

The level acceleration-deceleration method involves deriving true pressure altitude HP from geometric altitude Z and a calculated or measured difference in geometic altitude and pressure altitude, Z - HP. Ideally, the test aircraft flies a constant-altitude acceleration-deceleration run to cover the desired Mach range. However, if desired, this method can be used for a climbing or descending maneuver. Geometric altitude is obtained from ground-based radar.

An atmospheric analysis is conducted to determine values of Z - HP, pressure altitude gradient G, and ambient temperature T. When the value of Z - HP is subtracted from the radar altitude, true pressure altitude is obtained. The value Z - HP is referenced vertically, directly above the radar site. The pressure altitude gradient is the slope of the pressure altitude in the horizontal plane and is represented in vector form. The magnitude is in geopotential feet per nautical mile of horizontal distance and in the direction of decreasing Z - HP. The origin of this vector is at the radar site.

The atmospheric analysis is based on information received from rawinsonde soundings. NASA Ames Research Center, Dryden Flight Research Facility (Ames-Dryden) receives information from the Air Weather Service at the Air Force Flight Test Center, Edwards AFB. The Air Weather Service gathers data from rawinsonde soundings made twice a day at numerous locations. The analysis is made from constant-pressure-level maps from which pressure-height, temperature, and wind information can be extracted for the flight test runs. In addition, an Edwards rawinsonde sounding is generally scheduled by Ames-Dryden to provide data in the vicinity of the calibration flight, at approximately the time of the flight. Data from this sounding provide more points per unit of altitude than does the other sounding information.

When using the level acceleration-deceleration method, two options are available to compensate for the horizontal pressure gradient: the nonsurvey option, using an atmospheric analysis, and the survey option, using a constant-speed survey run made along the same path as the acceleration-deceleration. The option descriptions follow.

Nonsurvey option. — The nonsurvey option of the level acceleration-deceleration method uses both Z - HP and pressure-altitude gradient from the weather analysis. At each time interval (typically every second), a value of Z - HP as a function of vertical height DZH and a value of Z - HP as a function of horizontal position DHPG are calculated. The value of DZH is obtained directly from the weather analysis; DHPG is calculated by multiplying the distance to the aircraft from the radar site, DR, by the pressure gradient G and the cosine of the difference between the azimuth angle and direction of the pressure gradient, DANG:

$$DHPG = (DR)(G \cos DANG)$$

A schematic of this method is shown in figure 1(a). True pressure altitude is then determined by the equation

$$HP = Z - DZH + DHPG - DZ$$

where DZ is an adjustment. The variable DZ defaults to zero if there is no existing static-pressure position error correction. If a position error is firmly established for a specific flight condition, DZ can be used so that the calculated pressure altitude at the same flight condition is equal to the known pressure altitude. The previous equation can also be used for the "bootstrap" method where, again, the static pressure position error is known at a reference point, but values of DZH and DHPG are not used. In such a case, DZ is evaluated at the reference point. Finally, in all cases, DZ is applied as a constant throughout the entire test run.

Survey option. — With the survey option, DZH is used to provide a vertical component of Z - HP, as in the nonsurvey method. However, a different scheme is used to arrive at a horizontal component. During the flight test, a constant-Mach run is made at the same altitude and over the same path as the acceleration-deceleration run. From the survey run, a correction for pressure gradient error and radar error (such as, in elevation angle) can be obtained as a function of radar-antenna elevation angle. An initial value of Z - HPT = DZES is determined at the highest elevation angle, where radar errors are at a minimum. A table of Z - HPT values (DZEN) as a function of elevation angle is generated from the survey run. Then a correction for each point along the acceleration-deceleration can be determined from DZEN at the corresponding elevation angle. This method is shown schematically in figure 1(b). The following equation is used to calculate pressure altitude:

$$HP = Z - DZH - (DZEN - DZES) - DZ$$

Radar-Rawinsonde Method

The radar-rawinsonde method determines true Mach number directly. Adiabatic equations relating Mach number to stagnation and static pressure are used:

$$M = [5(PT/P)^{0.2857} - 5]^{0.5} \qquad M < 1.0$$

$$M = [(1.42851 - 0.357143y - 0.0625y^2 - 0.025y^3 - 0.012617y^4 - 0.00715y^5 - 0.004358y^6 - 0.0087725y^9)/y]^{0.5} \qquad M > 1.0$$

where

$$y = 1.839371/(PT/P)$$

The supersonic equation accounts for the total pressure loss across a normal shock wave. The approximation is given in appendix A. It is assumed in the equations that there is no position error in the measured stagnation pressure. The static pressure as a function of geometric altitude is obtained from the rawinsonde listing for the flight test.

Descent Pressure Method

The descent pressure method, which can also be used for climbs, is similar to the level acceleration-deceleration method in that it calculates true pressure altitude by subtracting Z - HP from radar altitude. However, with this method the test aircraft is flown in a descent or ascent, as opposed to level flight, and no corrections are made for horizontal variations in the atmosphere.

As in other methods, Z - HP is obtained from an atmospheric analysis, and DZ can be used to obtain the correct pressure altitude at a reference point in the run for which the static-pressure position error is known. As previously described, this correction is then applied throughout the rest of the run.

$$HP = Z - DZH - DZ$$

Descent Temperature Method

The descent temperature method, which can also be used for climbs, calculates true pressure altitude from stagnation temperature, indicated Mach number, and radar altitude. The difference between true and indicated pressure altitude then determines the correction for static-pressure position error.

Procedurally, the calculation begins at a reference point (either the first or last point of the run) for which the true pressure altitude HPREF is required.

The departure of the radar altitude from the reference altitude is divided into increments, each being converted to a pressure-altitude increment by the following method. First, the geometric altitude increments are converted to geopotential altitude by using the approximation

$$\Delta h \approx (20{,}825{,}000\ \Delta Z)/(20{,}825{,}000 + Z)\ \text{ft}$$

The resulting geopotential increment is converted to the associated pressure-altitude increment by the equation

$$\Delta HP = TS/T\ \Delta h$$

In this equation the temperatures are calculated by averaging the temperatures at the end points of the altitude increments. The standard temperature TS is calculated as a function of geopotential altitude. The ambient temperature T is determined from the approximation

$$T = TT/(1 + 0.2M^2)$$

Iteration is required in this equation since Mach number is used. Indicated Mach number is first used to obtain T, from which position error (and hence a new Mach number) is calculated. This new Mach number is used in the preceding equation for another iterative cycle. Convergence on true Mach number occurs rapidly.

After the geometric altitude increments have been converted to pressure-altitude increments, they are summed to determine position error correction for the test point:

$$HP = HPREF + \sum_{j=0}^{y} (\Delta HP)_j - HPT$$

where y is the number of altitude increments and the sum of the first two terms on the right-hand side of the equation is the true pressure altitude.

Total Temperature Method

The total temperature method solves for Mach number from the equation

$$TT = T(1 + 0.2M^2)$$

The specific heat ratio is assumed to be 1.4. The stagnation temperature TT is measured onboard the test aircraft, and the ambient temperature T is obtained from the atmospheric analysis, or directly from the rawinsonde.

The correction for Mach number determined by this method is for total pitot-static position error, not just the correction for the static-pressure position error.

METHOD LIMITATIONS

General Limitations

This program has several limitations. No data smoothing is done nor does the program seek and discard wild points. There are no provisions to compensate for pneumatic lag of the airspeed system. In addition, there are no corrections to the position error for real gas effects. One safeguard, however, is that if the value $PT < P$, calculations for that point are terminated.

Radar Data

The radar data used by Ames-Dryden are obtained from an FPS-16 radar system. Following a series of flight experiments, it was determined that the radar data may be questionable when the radar antenna elevation angle is below 7°. This is due in part to the large uncertainty of the refraction correction that exists at the low elevation angles. However, radar data from the lower elevation angles may be used if the survey option is exercised because the option will account for systematic radar errors. In addition, loss of beacon track will give unusable radar data.

Level Acceleration-Deceleration

When using the level acceleration-deceleration method with the nonsurvey option, the results are very dependent on the accuracy of the atmospheric analysis.

Temperature Descent Method

When exercising the temperature descent option, the altitude intervals should be less than 100 ft to adequately approximate the ambient temperature.

PROGRAM USE

General Description

To implement the program the user must first create a merged file of the necessary aircraft and radar data. All other necessary inputs are contained on a single data card that must be submitted with PERROR. The program then reads the data for the specified time interval, and unit conversions of inputs to the standard English system are made if required. If total pressure PT is not available directly, it is calculated from either impact pressure QCIT and static pressure PICT or velocity VCT and pressure altitude HPT. The output format for the flight conditions are then written.

At this point the corrections for position error are calculated, using any or all of the five methods, but usually no more than three. Each method uses common subroutines for calculating Mach number, pressure altitude (1962 Standard Atmosphere), and static pressure, for example. The main program calculates and writes the corrections for Mach number DM, static pressure DPR, and pressure altitude DHP caused by position error for each method used. After the calculations are performed for one time point using all methods selected, the program completes the calculations for the next time point. A listing of the program is presented in appendix B.

Program Inputs

A block diagram of the input sources to PERROR is shown in figure 2. In addition, table 1 lists the necessary input parameters for a particular method. First, a flight data base file and a radar data file are created. Rawinsonde data may be included when creating the radar data file. The data card required to run PERROR must contain the following quantities: FLIGHT, RUN, IUNITS, OO, and QQ. Zero is the default value for the last four quantities. The indices II, KK, LL, MM, and NN are used to designate which position error methods are to be used (refer to the NOMENCLATURE section).

The following is an example of a data card used when running PERROR (the DZH and GGH tables given in table 2 are included in the data card below).

```
 $PROG KK=1, FLIGHT=557, RUN=1, NDZH=32, NGGH=24,
 DZHTABL=2300.,175.,5000.,202.,7000.,240.,9000.,287.,11000.,
 340.,15000.,450.,20000.,650.,25000.,772.,3000.,895.,31000.,
 915.,35000.,915.,38000.,900.,40000.,850.,42000.,866.,44000.
 ,896.,46000.,920.,
 GGHTABL=5000.,0.,0.,11000.,0.5,30.,20000.,0.9,27.,25000.,
 1.1,35.,30000.,1.25,45.,35000.,1.2,40.,40000.,1.4,45.,
 46000.,1.9,45., $
6/7/8/9 card
```

Program Output

A sample page of output is shown in figure 3. For this particular case, the level acceleration-deceleration method with the nonsurvey option was used. For each time point, angle of attack, angle of sideslip, indicated Mach number, indicated pressure altitude, stagnation pressure, and total temperature are listed, in addition to the corrections to Mach number, altitude, and static pressure due to position error. The corrections are labeled DM__, DHP__, and DPR__, respectively. The last letter of each parameter specifies the method used to calculate the position error. If QQ = 0, appropriate headings for CP output are made. Any other integer for QQ results in appropriate headings for DHP output.

Conclusions

The computer program presented calculates static pressure position error of airspeed systems. The program contains five separate methods for determining position error, of which the user may select from one to all five at a time. All the methods require an input data file consisting of aircraft and radar parameters. In addition, most of the methods require rawindsonde data or an atmospheric analysis, or both.

National Aeronautics and Space Administration
Ames Research Center
Dryden Flight Research Facility
Edwards, California, Jan. 9, 1984

APPENDIX A — SUPERSONIC MACH NUMBER EQUATION

The following derivation for calculating supersonic Mach number (PT/P > 1.893) was developed by Raymond Jackson of NASA Ames Research Center, Dryden Flight Research Facility.

$$\frac{QCIT}{P} = \frac{1 + \gamma}{2} M^2 \left[\frac{(1 + \gamma)^2 M^2}{4\gamma M^2 - 2(\gamma - 1)}\right]^{1/(\gamma-1)} - 1 \quad \text{(eq. 3-26, ref. 1)}$$

$$1 + \frac{QCIT}{P} = \frac{1 + \gamma}{2} M^2 \left[\frac{\frac{\gamma + 1}{2} M^2}{\frac{2\gamma}{\gamma + 1} M^2 - \left(\frac{\gamma - 1}{\gamma + 1}\right)}\right]^{1/(\gamma-1)}$$

Let

$$K = \frac{\gamma - 1}{\gamma + 1}\left[\frac{1}{\gamma}\left(\frac{\gamma + 1}{2}\right)^2\right]^{\gamma/(\gamma-1)}$$

and substitute $\gamma = 1.4$:

$$K = \frac{1}{6}\left(\frac{5}{7}\right)^{7/2}\left(\frac{6}{5}\right)^7 = 0.18393711$$

Let

$$x = \frac{\gamma - 1}{2\gamma M^2} = \frac{1}{7M^2}$$

Then

$$\frac{K}{1 + \frac{QCIT}{P}} = x(1 - x)^{1/(\gamma-1)} = x(1 - x)^{5/2}$$

Binomial expansion yields

$$y = \frac{K}{1 + \frac{QCIT}{P}} \simeq x - \frac{5}{2}x^2 + \frac{15}{8}x^3 - \frac{5}{16}x^4 - \frac{5}{128}x^5 - \frac{3}{256}x^6 - \frac{5}{1004}x^7 \ldots$$

and reversion of series gives

$$x = y + 2.5y^2 + 10.625y^3 + 55y^4 + 316.05469y^5 + 1938y^6 + 12421.919y^7$$

Let

$$z = 10y = 10\left(\frac{0.1839371}{1 + \frac{QCIT}{P}}\right) = \frac{1.83971}{1 + \frac{QCIT}{P}}$$

Then

$$x = 0.1z + 0.025z^2 + 0.010625z^3 + 0.0055z^4$$
$$+ 0.0031605468z^5 + 0.001938z^6 + 0.0012421919z^7$$

$$\frac{1}{x} = \frac{10}{z} - 2.5 - 0.4375z - 0.175z^2 - 0.088320315z^3$$
$$- 0.05005z^4 - 0.030420265z^5$$

$$M^2 = \frac{1}{7x} = \frac{1.4285714}{z} - 0.35714286 - 0.0625z$$
$$- 0.025z^2 - 0.012617187z^3 - 0.00715z^4$$
$$- 0.004345752z^5 - 0.0087725z^8$$

The last term of this equation was added empirically so that the equation is exact at Mach 1. The effect of the empirical term is null at high Mach numbers.

$$M = [(1.428572 - 0.3571428z - 0.0625z^2 - 0.025z^3$$
$$- 0.01216172z^4 - 0.00715z^5 - 0.0043458z^6$$
$$- 0.0087725z^9)/z]^{1/2}$$

where $z = 1.839371(P/PT)$.

APPENDIX B — PERROR LISTING

The following is a complete listing of the PERROR program. It is written in FORTRAN 77 and is run on an ELXSI computer at NASA Ames-Dryden.

```
      PROGRAM PERROR
C
C   THIS PROGRAM CALCULATES POSITION ERRORS USING THE OPTIONS
C    OF SEVERAL DIFFERENT METHODS
C
      DIMENSION DZHTABL(40),GGHTABL(60)
      DIMENSION EMINMAX(10),EAPPROX(10),ZHPDIF(10),
     - ISTSV(4),IETSV(4),ISTAD(4),IETAD(4)
      INTEGER FLIGHT, RUN, OO, QQ
      DIMENSION IT(4)
      DATA LINE /6/
      NAMELIST /PROG/ II, KK, LL, MM, NN, OO, QQ, IUNTS, HPREF,
     *FLIGHT, RUN, DZ, NDZH, DZHTABL, NGGH, GGHTABL, ISURVEY,
     *ISTSV, IETSV, ISTAD, IETAD
      DATA II, KK, LL, MM, NN, OO, QQ, IUNTS, HPREF, FLIGHT, RUN,
     *DZ, ISURVEY /14*0/
      LINE=0
      OPEN (UNIT=1,FILE='perrorin')
      OPEN (UNIT=3,FILE='perrorout')
      OPEN (UNIT=4,FILE='frmerge',FORM='UNFORMATTED')
C
C  READ NAMELIST CARDS, SOME OF WHICH ARE OPTIONAL DEPENDING ON METHODS USED
C
      READ(1,PROG)
C
C  IF A SURVEY RUN IS BEING PROCESSED, READ FLIDAB-RADAR DATA DURING SURVEY,
C  CREATE DZEN TABLE, AND CALCULATE DZES.
      IF(ISURVEY.EQ.0) GO TO 7
      MSTSV = 3600000*ISTSV(1) + 60000*ISTSV(2) + 1000*ISTSV(3) +
     - ISTSV(4)
      METSV = 3600000*IETSV(1) + 60000*IETSV(2) + 1000*IETSV(3) +
     - IETSV(4)
      MSTAD = 3600000*ISTAD(1) + 60000*ISTAD(2) + 1000*ISTAD(3) +
     - ISTAD(4)
      METAD = 3600000*IETAD(1) + 60000*IETAD(2) + 1000*IETAD(3) +
     - IETAD(4)
      EMINMAX(1) = 1.0E10
      EMINMAX(10) = -1.0E10
   30 READ(4,END=31) TIME,Z,R,E,T,RRA,PR,
     - VCT,HPT,A,B,PT,PICT,QCIT,TT,FPA,FPH
      E=E/57.3
      RRA=RRA/57.3
      X=R*COS(E)*SIN(RRA)
      Y=R*COS(E)*COS(RRA)
      IDTTL=TIME*1000.
      IF(IDTTL.LT.MSTSV) GO TO 30
      IF(IDTTL.GT.METSV) GO TO 31
      IF(E.LT.EMINMAX(1)) EMINMAX(1)=E
      IF(E.GT.EMINMAX(10)) EMINMAX(10)=E
      GO TO 30
   31 DO 32 I=2,9
      EMINMAX(I) = EMINMAX(1) + (I-1)*((EMINMAX(10)-EMINMAX(1))/9.0)
   32 CONTINUE
      REWIND 4
      DO 33 I=1,10
      EAPPROX(I) = 1.0E10
      ZHPDIF(I) = 0.0
   33 CONTINUE
   34 READ(4,END=36) TIME,Z,R,E,T,RRA,PR,
     - VCT,HPT,A,B,PT,PICT,QCIT,TT,FPA,FPH
      E=E/57.3
```

```
      RRA=RRA/57.3
      X=R*COS(E)*SIN(RRA)
      Y=R*COS(E)*COS(RRA)
      IDTTL=TIME*1000.
      IF(IDTTL.LT.MSTSV) GO TO 34
      IF(IDTTL.GT.METSV) GO TO 36
      DO 35 I=1,10
      IF(ABS(E-EMINMAX(I)).GT.ABS(EAPPROX(I)-EMINMAX(I))) GO TO 35
      EAPPROX(I) = E
      ZHPDIF(I) = Z-HPT
   35 CONTINUE
      GO TO 34
   36 DZES = ZHPDIF(10)
      REWIND 4
      DO 37 I=1,10
      WRITE(3,101) EMINMAX(I),EAPPROX(I),ZHPDIF(I)
  101 FORMAT (1H ,3F12.5)
   37 CONTINUE

C  CALL SUBROUTINE TO WRITE TOP OF PAGE HEADINGS
C
    7 CALL HEAD(FLIGHT,RUN)
C   *****   READ FLIDAB AND RADAR DATA FROM MERGED DATA BASE.
   10 READ(4,END=99) TIME,Z,R,E,T,RRA,PR,
     -VCT,HPT,A,B,PT,PICT,QCIT,TT,FPA,FPH
      E=E/57.3
      RRA=RRA/57.3
      X=R*COS(E)*SIN(RRA)
      Y=R*COS(E)*COS(RRA)
      IDTTL=TIME*1000.
      IF(ISURVEY.NE.0 .AND. IDTTL.LT.MSTAD) GO TO 10
      IF(ISURVEY.NE.0 .AND. IDTTL.GT.METAD) GO TO 99
      IF(KK.EQ.0 .AND. LL.EQ.0) GO TO 6
C   *****   OBTAIN DZH FROM TABLE DZHTABL.
      IF(Z.LE.DZHTABL(1)) DZH=DZHTABL(2)
      IF(Z.GE.DZHTABL(NDZH-1)) DZH=DZHTABL(NDZH)
      IF(Z.LE.DZHTABL(1) .OR. Z.GE.DZHTABL(NDZH-1)) GO TO 3
      DO 1 I=3,NDZH,2
      IF(Z.LT.DZHTABL(I)) GO TO 2
    1 CONTINUE
    2 RATIO = (Z-DZHTABL(I-2))/(DZHTABL(I)-DZHTABL(I-2))
      DZH = DZHTABL(I-1) + RATIO*(DZHTABL(I+1)-DZHTABL(I-1))
      IF(KK.EQ.0) GO TO 6
      IF(ISURVEY.NE.0) GO TO 8
C   *****   OBTAIN G AND GH FROM TABLE GGHTABL.
    3 IF(Z.LE.GGHTABL(1)) G=GGHTABL(2)
      IF(Z.LE.GGHTABL(1)) GH=GGHTABL(3)
      IF(Z.GE.GGHTABL(NGGH-2)) G=GGHTABL(NGGH-1)
      IF(Z.GE.GGHTABL(NGGH-2)) GH=GGHTABL(NGGH)
      IF(Z.LE.GGHTABL(1) .OR. Z.GE.GGHTABL(NGGH-2)) THEN
      G=G/6080.0
      GO TO 6
      ENDIF
      DO 4 I=4,NGGH,3
      IF(Z.LT.GGHTABL(I)) GO TO 5
    4 CONTINUE
    5 RATIO = (Z-GGHTABL(I-3))/(GGHTABL(I)-GGHTABL(I-3))
      G = GGHTABL(I-2) + RATIO*(GGHTABL(I+1)-GGHTABL(I-2))
      G = G/6080.0
      GH = GGHTABL(I-1) + RATIO*(GGHTABL(I+2)-GGHTABL(I-1))
      GO TO 6
```

```
C   *****   OBTAIN DZEN FROM TABLES EAPPROX AND ZHPDIF.
    8 IF(E.LE.EAPPROX(1)) DZEN=ZHPDIF(1)
      IF(E.GE.EAPPROX(10)) DZEN=ZHPDIF(10)
      IF(E.LE.EAPPROX(1) .OR. E.GE.EAPPROX(10)) GO TO 6
      DO 9 I=2,10
      IF(E.LT.EAPPROX(I)) GO TO 11
    9 CONTINUE
   11 RATIO = (E-EAPPROX(I-1))/(EAPPROX(I)-EAPPROX(I-1))
      DZEN = ZHPDIF(I-1) + RATIO*(ZHPDIF(I)-ZHPDIF(I-1))

C  TEST LINE COUNT AND CALL HEAD IF NECESSARY
C
    6 LINE = LINE + 1
      IF (MOD(LINE,6) .EQ. 0) CALL HEAD(FLIGHT,RUN)
C
C  CALL SUBROUTINE UNITS TO MAKE UNIT CONVERSIONS TO ENGLISH UNITS IF NECESSARY
C
      IF(IUNTS.EQ.0) GO TO 15
      CALL UNITS (IUNTS,PT,PICT,QCIT,TT)
C
C  CALCULATE PT IF QCIT AND PICT OR VCT AND HPT ARE USED AS INPUTS
C
   15  IF(OO.EQ.0) GO TO 20
       IF(OO.EQ.2) GO TO 17
   16  PT=QCIT+PICT
       GO TO 20
   17  CALL QCC(VCT,QCIT)
       CALL PRES(HPT,PICT)
       GO TO 16
C
C  CALCULATIONS OF INDICATED MACH NO AND INDICATED PRESSURE ALTITUDE
C
   20 IF(PT.LT.PICT) GO TO 10
      PRATI=PT/PICT
      IF(OO.NE.0.AND.QQ.EQ.0) GO TO 21
      QCIT=PT-PICT
   21 CALL MACH(PRATI,AMI)
      CALL HPC (PICT,HPI)
C
C  WRITE HEADING FOR FLIGHT CONDITIONS, WRITE CONDITIONS AND HEADING FOR
C                      POSITION ERRORS
C
      WRITE (3,901)
      CALL  SECTOT(TIME,IT)
      WRITE(3,902) IT,A,B,AMI,HPI,PT,TT
      IF(QQ.NE.0) GO TO 23
      WRITE (3,903)
      GO TO 24
   23 WRITE (3,916)
C
C  POSITION ERROR CALCULATION USING RADAR-RAWINSONDE METHOD
C
   24 IF (II.EQ.0) GO TO 40
      P=PR
      PRAT=PT/PR
      CALL MACH (PRAT,AM)
      CALL PECALC(P,PICT,QCIT,HP,HPI,AM,AMI,DM,DPR,DHP,CP,QQ)
      IF(QQ.NE.0) GO TO 25
      WRITE (3,904)DM,DPR,DHP
      GO TO 40
   25 WRITE (3,910) DM,DPR,CP
```

```
C
C  POSITION ERROR CALCULATION USING LEVEL DECEL METHOD
C
   40 IF(KK.EQ.0) GO TO 50
      IF(ISURVEY.EQ.0) CALL LEVDNS(AM,G,GH,HP,P,PT,Z,DZH,X,Y,RRA,DZ)
      IF(ISURVEY.NE.0) CALL LEVDS(AM,HP,P,PT,Z,DZH,DZ,DZEN,DZES)
      CALL PECALC(P,PICT,QCIT,HP,HPI,AM,AMI,DM,DPR,DHP,CP,QQ)
      IF(QQ.NE.0) GO TO 41
      WRITE(3,906)DM,DPR,DHP
      GO TO 50
   41 WRITE (3,912) DM,DPR,CP
C
C  POSITION ERROR CALCULATION USING DESCENT PRESSURE METHOD
C
   50 IF(LL.EQ.0) GO TO 60
      CALL DESCP(DZ,Z,DZH,P,PT,AM,HP)
      CALL PECALC(P,PICT,QCIT,HP,HPI,AM,AMI,DM,DPR,DHP,CP,QQ)
      IF(QQ.NE.0) GO TO 51
      WRITE(3,907) DM,DPR,DHP
      GO TO 60
   51 WRITE(3,913) DM,DPR,CP
C
C  POSITION ERROR CALCULATION USING DESCENT TEMPERATURE METHOD
C
   60 IF(MM.EQ.0) GO TO 70
      CALL DESCT(Z,HPREF,AMI,TT,PT,PS,AMS,HPS)
      CALL PECALC(PS,PICT,QCIT,HPS,HPI,AMS,AMI,DM,DPR,DHP,CP,QQ)
      IF(QQ.NE.0) GO TO 61
      WRITE (3,908)DM,DPR,DHP
      GO TO 70
   61 WRITE (3,914) DM,DPR,CP
C
C  POSITITION ERROR CALCULATION USING TOTAL TEMPERATURE METHOD
C
   70 IF (NN.EQ.0) GO TO 10
      CALL TOT(TT,T,P,PT,AM)
      CALL PECALC(P,PICT,QCIT,HP,HPI,AM,AMI,DM,DPR,DHP,CP,QQ)
      IF(QQ.NE.0) GO TO 71
      WRITE (3,909)DM,DPR,DHP
      GO TO 98
   71 WRITE (3,915) DM,DPR,CP
   98 GO TO 10

   99 CLOSE(1)
      CLOSE(3)
      CLOSE(4)
      STOP
C
C  FORMAT STATEMENTS
C
  901 FORMAT(1X,/,T3,'TIME(HR,MIN,SEC)',T25,'ALPHA,DEG',T45,'BETA, DEG',
     *T63,'IND MACH NO',T79,'IND PRESSURE ALT',T99,'STAG PRESSURE',T121,
     *'TOT TEMP'/T85,'FEET',T104,'PSFA',T122,'DEG, R'/)
  902 FORMAT(1X,T7,3(I2,1X),I3,T28,F5.1,T47,F5.1,
     *T67,F5.3,T85,F6.0,T103,F6.1,T122,F6.1//)
  903 FORMAT (1X,T12,'DMR',T18,'DPRP',T25,'DHPR',T32,'DMLD',T38,'DPRLD',
     - T45,'DHPLD',T52,'DMDP',T58,'DPRDP',T65,'DHPDP',T72,'DMDT',
     - T78,'DPRDT',T85,'DHPDT',T92,'DMTT',T98,'DPRTT',T105,'DHPTT'/)
  904 FORMAT (1X,T10,F6.4,T16,F6.3,T23,F7.0)
  906 FORMAT ('+',T31,F6.4,T37,F6.3,T44,F7.0)
  907 FORMAT ('+',T51,F6.4,T57,F6.3,T64,F7.0)
```

```
  908 FORMAT ('+',T71,F6.4,T77,F6.3,T84,F7.0)
  909 FORMAT ('+',T91,F6.4,T97,F6.3,T104,F7.0)
  910 FORMAT (1X,T10,F6.4,T17,F6.3,T24,F6.4)
  912 FORMAT ('+',T30,F6.4,T37,F6.3,T44,F6.4)
  913 FORMAT ('+',T50,F6.4,T57,F6.3,T64,F6.4)
  914 FORMAT ('+',T70,F6.4,T79,F6.3,T84,F6.4)
  915 FORMAT ('+',T90,F6.4,T97,F6.3,T104,F6.4)
  916 FORMAT (1X,T12,'DMR',T18,'DPRR',T26,'CPR',T32,'DMLD',T38,'DPRLD',
     - T46,'CPLD',T52,'DMDP',T58,'DPRDP',T66,'CPDP',T72,'DMDT',
     - T78,'DPRDT',T86,'CPDT',T92,'DMTT',T98,'DPRTT',T106,'CPTT'/)
      END
C
      SUBROUTINE SECTOT(time,it)
C
C        THIS SUBROUTINE CONVERTS A SINGLE REAL TIME WORD REPRESENTING
C        TOTAL SECONDS TO A 4 WORD INTEGER ARRAY REPRESENTING
C        hr,min,sec,msec
C
      real min,msec
      integer  it(3)
C
      hr=time/3600.
      it(1)=hr
C
      min=time/60.
      it(2)=min-it(1)*60
C
      sec=time
      it(3)=sec-it(2)*60-it(1)*3600
C
      msec=time*1000.
      it(4)=msec-it(3)*1000-it(2)*6.e04-it(1)*3.6e06
C
      return
      end
C
C
      SUBROUTINE HEAD(FLIGHT,RUN)
C
C   THIS SUBROUTINE WRITES MAIN HEADINGS AT THE TOP OF EACH PAGE
C
      WRITE(3,100)FLIGHT,RUN
  100 FORMAT(1H1,T36,'STATIC PRESSURE POSITION ERROR CORRECTIONS FOR FLI
     *GHT ',I3,' RUN ',I2,///)
      RETURN
      END
C
      SUBROUTINE UNITS(IUNTS,PT,PICT,QCIT,TT)
C
C   THIS SUBROUTINE CONVERTS PARAMETERS IN METRIC UNITS TO ENGLISH UNITS
C
      IF(IUNTS.EQ.2) GO TO 5
      PT=144.*PT
      PICT=144.*PICT
      QCIT=144.*QCIT
      GO TO 10
    5 PT=.020885*PT
      PICT=.020885*PICT
      QCIT=.020885*QCIT
      TT = 1.8*(TT-459.67)+273.16
   10 RETURN
```

```
      END
C
      SUBROUTINE PECALC (P,PICT,QCIT,HP,HPI,AM,AMI,DM,DPR,DHP,CP,QQ)
C
C    THIS SUBROUTINE CALCULATES POSITION ERRORS COMMON TO ALL OF THE
C                       SUBROUTINES
      DPR= (P-PICT)/P
      DM=AM-AMI
      IF (QQ.NE.0) GO TO 5
      CALL HPC(P,HP)
      DHP=HP-HPI
      GO TO 10
    5 CP=(P-PICT)/QCIT
   10 RETURN
      END
C
      SUBROUTINE MACH (PRAT,AM)
C
C   THIS SUBROUTINE CALCULATES MACH NUMBER AS A FUNCTION OF PRESSURE RATIO
C
      IF (PRAT.GT.1.89293) GO TO 5
      AM=SQRT(5*((PRAT)**.2857-1.))
      GO TO 10
    5 Y=1.839371/PRAT
      AM=SQRT((1.42857-.357143*Y-.0625*(Y**2)-.025*(Y**3)-.012617*(Y**4)
     *-.00715*(Y**5)-.0043458*(Y**6)-.0087725*(Y**9))/Y)
   10 RETURN
      END
C
      SUBROUTINE PRATIO (AM,PRAT)
C
C    THIS SUBROUTINE CALCULATES PRESSURE RATIO AS A FUNCTION OF MACH NUMBER
C
      IF(AM.LT.1.) GO TO 5
      PRAT=1.2*AM**2*(((5.76*AM**2)/(5.6*AM**2-.8))**2.5)
      GO TO 10
    5 PRAT=(1.+.2*AM**2)**3.5
   10 RETURN
      END
C
      SUBROUTINE HPC(P,HP)
C
C    THIS SUBROURINE CALCULATES PRESSURE ALTITUDE AS A FUNCTION OF STATIC
C              PRESSURE
      IF(P.LT.114.35) GO TO 5
      IF(P.LT.472.68) GO TO 10
      HP=-145442.*((P/2116.22)**.190262-1.)
      GO TO 15
    5 HP=65616.8+710794.*((.008746*P)**(-.029271)-1.)
      GO TO 15
   10 HP=36089.2-20805.7*(ALOG(.0021155*P))
      GO TO 15
   15 RETURN
      END
C
      SUBROUTINE PRES (HP,PS)
C
C    THIS SUBROUTINE CALCULATES STATIC PRESSURE AS A FUNCTION OF PRESSURE
C              ALTITUDE
      IF(HP.LT.36089.2) GO TO 5
      IF(HP.LT.65616.8) GO TO 10
```

```
      PS=51.9754*((1 -.0000042204*(HP-65616.8))**(-11.3878),
      GO TO 15
    5 PS=2116.22*(1.-.0000068754*HP)**5.2561
      GO TO 15
   10 PS=472.68*(2.71828**(-.000048063*(HP-36089.24)))
   15 RETURN
      END
C
      SUBROUTINE QCC(VCT,QC)
C
C   THIS SUBROUTINE CALCULATES QC AS A FUNCTION OF VCT
C
      IF(VCT.GT.661.4) GO TO 5
      QC=2116.22*((.142857*(.0000032*VCT**2)+1.)**3.5-1)
      GO TO 10
    5 QC=.005805*VCT**2*(5.76/(5.6-349964./VCT**2))**2.5-2116.22
   10 RETURN
      END
C
      SUBROUTINE LEVDNS (AM,G,GH,HP,P,PT,Z,DZH,X,Y,RRA,DZ)
C
C
C   THIS SUBROUTINE CALCULATES TRUE MACH NUMBER AND STATIC PRESSURE
C    BY THE LEVEL DECELERATION METHOD FOR NON-SURVEY TEST.
C
      DANG = ABS(RRA-GH)/57.3
      DR = SQRT(X**2 + Y**2)
      DHPG = DR*G*COS(DANG)
      HP = Z-DZH+DHPG-DZ
      CALL PRES(HP,P)
      PRAT=PT/P
      CALL MACH(PRAT,AM)
      RETURN
      END
      SUBROUTINE LEVDS(AM,HP,P,PT,Z,DZH,DZ,DZEN,DZES)

C     THIS SUBROUTINE CALCULATES TRUE MACH NUMBER AND STATIC PRESSURE
C     BY THE LEVEL DECELERATION METHOD FOR SURVEY TEST.

      HP = Z-DZH-DZEN+DZES-DZ
      CALL PRES (HP,P)
      PRAT = PT/P
      CALL MACH (PRAT,AM)
      RETURN
      END
      SUBROUTINE DESCP(DZ,Z,DZH,P,PT,AM,HP)
C   THIS SUBROUTINE CALCULATES TRUE MACH NUMBER AND STATIC PRESSURE
C   BY THE DESCENT STATIC PRESSURE METHOD
C
      HP = Z-DZ-DZH
      CALL PRES(HP,P)
      PRAT = PT/P
      CALL MACH(PRAT,AM)
      RETURN
      END
      SUBROUTINE TOT(TT,T,P,PT,AM)
C
C   THIS SUBROUTINE CALCULATES MACH NUMBER FROM MEASUREMENTS OF TOTAL
C   TEMPERATURE AND AMBIENT TEMPERATURE
      AM=SQRT(5*(TT-T)/T)
C
```

```
      CALL PRATIO(AM,PRAT)
      P=PT/PRAT
      RETURN
      END
      SUBROUTINE DESCT(Z,HPREF,AMI,TT,PT,P,AM,HP)
C   THIS SUBROUTINE CALCULATES TRUE MACH NUMBER AND STATIC PRESSURE
C    BY THE DESCENT TEMPERATURE METHOD
C
C   STATEMENT FUNCTION FOR CALCULATING AMBIENT TEMPERATURE
C
      LOGICAL FIRST
      DATA FIRST/.TRUE./
      IF(.NOT.FIRST) GO TO 10
      FIRST = .FALSE.
      HP=HPREF
      CALL PRES(HP,P)
      PRAT=PT/P
      CALL MACH(PRAT,AM)
      GO TO 60
C
C   CALCULATION OF TRUE PRESSURE ALTITUDE USING ITERATION
C
   10 DZ=Z-ZJ
C
C   SET COUNTER FOR NUMBER OF ITERATIONS LIMITED TO 10
C
      N=0
      HPO=HP
      ZBAR = (Z+ZJ)/2.
      TTBAR=(TT+TTJ)/2.
      AMC=AMI
      DH=(20825000.*DZ)/(20825000.+ZBAR)
C   CALCULATION OF AMBIENT TEMPERATURE AS A FUNCTION OF ALTITUDE
C
C
      IF(ZBAR.GT.36089.) GO TO 20
      TS=518.7*(1.-.0000068756*ZBAR)
      GO TO 40
   20 IF(ZBAR.GT.65617.) GO TO 30
      TS=390.
      GO TO 40
   30 TS=390.*(1.+.0000014069*(ZBAR-65617.))
   40 TA=TTBAR/(1.0 + 0.2*AMC**2)
      DHP=TS*DH/TA
      HP=HPO+DHP
      CALL PRES(HP,P)
      PRAT=PT/P
      CALL MACH(PRAT,AM)
      DMI=AM-AMC
      IF(ABS(DMI).LT.0.0005) GO TO 60
      AMC=AM
      N=N+1
      IF(N.GT.10) GO TO 60
      GO TO 40
   60 ZJ=Z
      TTJ=TT
      RETURN
      END
```

REFERENCES

1. Gracey, William: Measurement of Aircraft Speed and Altitude. NASA RP-1046, 1980.

2. Larson, Terry J.; and Ehernberger, L.J.: Techniques Used For Determination of Static Source Position Error of a High Altitude Supersonic Airplane. NASA TM X-3152, 1974.

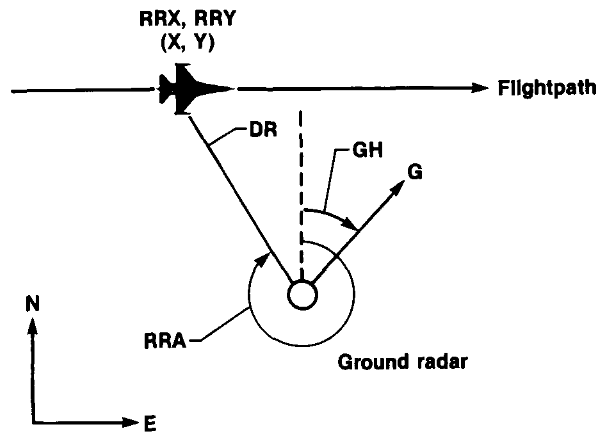
TABLE 1. — NECESSARY METHOD INPUTS TO PERROR

	LEVD		DESCP	DESCT	TOT	RADAR
	Survey (ISURVEY = 1)	Nonsurvey (ISURVEY = 0)				
$PROG card	KK = 1 ISTSV, IETSV ISTAD, IETAD DZHTABL DZ NDZH	KK = 1 DZHTABL DZ NDZH NGGH	LL = 1 DZH DZ NDZH	MM = 1 HPREF	NN = 1	II = 1
Merged file	P PT* Z	P, PT* GGHTABL Z, RRX, RRY RRA	P PT* Z	P PT* Z TT	P PT* TT	PR PT* Z
Calculated	DZEN DZES MI	DANG DR MI	MI	MI	MI T	MI

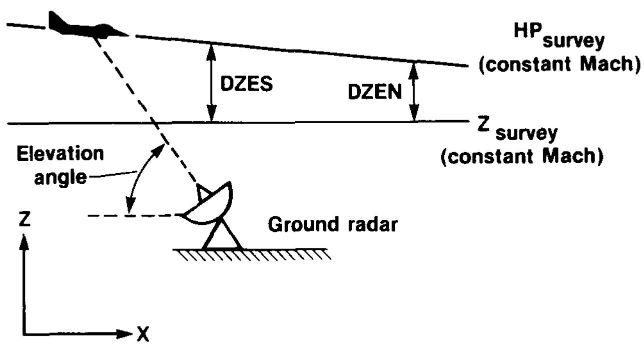
*QCIT and PICT or HPT and VCT may be substituted.

TABLE 2. — ATMOSPHERIC ANALYSIS RESULTS FOR F-14 FLIGHT 557

Z, 10^3 ft	DZH	G, ft/nmi	GH, deg true north
46	920	1.9	045
44	896	---	---
42	866	---	---
40	850	1.4	045
38	900	---	---
35	915	1.2	040
31	915	---	---
25	772	1.1	035
20	650	0.9	027
15	450	---	---
11	340	0.5	030
9	287	---	---
7	240	---	---
5	202	0.0	---
surface	175	---	---



(a) Nonsurvey method.



(b) Survey method.

Figure 1. Methods used to compensate for the horizontal pressure gradient when using level acceleration-deceleration.

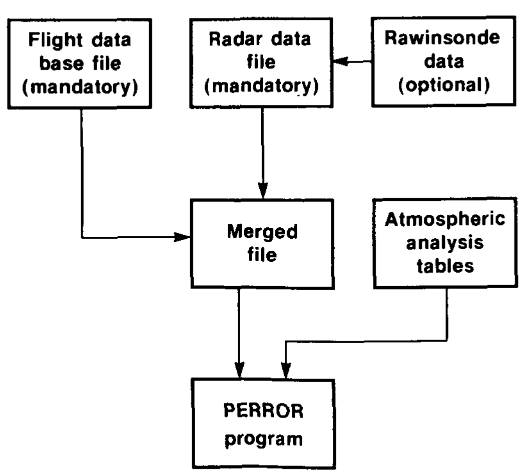


Figure 2. Block diagram of input sources to PERROR.

```
1                          STATIC PRESSURE POSITION ERROR CORRECTIONS FOR FLIGHT 178 RUN  1


 TIME(HR,MIN,SEC)     ALPHA,DEG          BETA, DEG        IND MACH NO      IND PRESSURE ALT     STAG PRESSURE        TOT TEMP
                                                                                 FEET               PSFA              DEG, R

     8 23 30   0           6.7              *****               .780            27851.             1035.3              -7.1

          DMR   DPRP   DHPR   DMLD  DPRLD  DHPLD  DMOP  DPROP  DHPOP  DMOT  DPROT  DHPOT  DMTT  DPRTT  DHPTT

+                             .0197 -.020   433.
 TIME(HR,MIN,SEC)     ALPHA,DEG          BETA, DEG        IND MACH NO      IND PRESSURE ALT     STAG PRESSURE        TOT TEMP
                                                                                 FEET               PSFA              DEG, R

     8 23 30 500           6.7              *****               .780            27851.             1035.3              -7.1

          DMR   DPRP   DHPR   DMLD  DPRLD  DHPLD  DMOP  DPROP  DHPOP  DMOT  DPROT  DHPOT  DMTT  DPRTT  DHPTT

+                             .0196 -.019   430.
 TIME(HR,MIN,SEC)     ALPHA,DEG          BETA, DEG        IND MACH NO      IND PRESSURE ALT     STAG PRESSURE        TOT TEMP
                                                                                 FEET               PSFA              DEG, R

     8 23 31   0           6.8              *****               .783            27862.             1037.6              -6.9

          DMR   DPRP   DHPR   DMLD  DPRLD  DHPLD  DMOP  DPROP  DHPOP  DMOT  DPROT  DHPOT  DMTT  DPRTT  DHPTT

+                             .0193 -.019   426.
 TIME(HR,MIN,SEC)     ALPHA,DEG          BETA, DEG        IND MACH NO      IND PRESSURE ALT     STAG PRESSURE        TOT TEMP
                                                                                 FEET               PSFA              DEG, R

     8 23 31 500           6.8              *****               .783            27862.             1037.6              -6.9

          DMR   DPRP   DHPR   DMLD  DPRLD  DHPLD  DMOP  DPROP  DHPOP  DMOT  DPROT  DHPOT  DMTT  DPRTT  DHPTT

+                             .0196 -.020   431.
 TIME(HR,MIN,SEC)     ALPHA,DEG          BETA, DEG        IND MACH NO      IND PRESSURE ALT     STAG PRESSURE        TOT TEMP
                                                                                 FEET               PSFA              DEG, R

     8 23 32   0           6.6              *****               .786            27873.             1039.6              -6.7

          DMR   DPRP   DHPR   DMLD  DPRLD  DHPLD  DMOP  DPROP  DHPOP  DMOT  DPROT  DHPOT  DMTT  DPRTT  DHPTT

+                             .0194 -.019   428.
```

Figure 3. Output format of PERROR program.

ORIGINAL PAGE IS
OF POOR QUALITY

1. Report No. NASA TM-86726	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle DIGITAL PROGRAM FOR CALCULATING STATIC PRESSURE POSITION ERROR		5. Report Date February 1987
		6. Performing Organization Code
7. Author(s) J. Blair Johnson, Terry J. Larson, and Jules M. Ficke		8. Performing Organization Report No. H-1284
9. Performing Organization Name and Address NASA Ames Research Center Dryden Flight Research Facility P.O. Box 273 Edwards, CA 93523-5000		10. Work Unit No. RTOP 505-31-21
		11. Contract or Grant No.
		13. Type of Report and Period Covered Technical Memorandum
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546		14. Sponsoring Agency Code
15. Supplementary Notes		

16. Abstract

A computer program written to calculate the static pressure position error of airspeed systems contains five separate methods for determining position error, of which the user may select from one to five at a time. The program uses data from both the test aircraft and the ground-based radar to calculate the error. In addition, some of the methods require rawinsonde data or an atmospheric analysis, or both.

The program output lists the corrections to Mach number, altitude, and static pressure that are due to position error. Reference values such as angle of attack, angle of sideslip, indicated Mach number, indicated pressure altitude, stagnation pressure, and total temperature are also listed.

17. Key Words (Suggested by Author(s)) Static pressure position error Computer code		18. Distribution Statement Unclassified — Unlimited Subject category 05	
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 26	22. Price* A03

*For sale by the National Technical Information Service, Springfield, Virginia 22161.