63-3-4

RTD-TDR-63-9, Vol III

FINAL TECHNICAL REPORT
EVALUATION AND MODIFICATION OF EXISTING PROTOTYPE DYNAMIC CALIBRATION SYSTEM FOR PRESSURE-MEASURING TRANSDUCERS
VOLUME III COMPUTER PROGRAMS

TECHNICAL DOCUMENTARY REPORT NO RTD-TDR-63-9, VOL III
March 1963

6593d Test Group (Development)
Air Force Flight Test Center
Edwards Air Force Base, California

Project No 3850 BPSN 3850, Task No 38506

(Prepared under Contract No AF 04(611) 8199
by Houston Engineering Research Corporation
Houston, Texas)

SECTION 1 TRANSFER FUNCTION APPROXIMATION PROGRAM

SCOPE

The Transfer Function Approximation Program (TFAP) is an IBM 7090 FORTRAN program which will compute the approximate transfer function for any linear pressure transducer system from the system's time response to a step input. Input data to the program contains information which specifies test conditions, information which defines the data range and density for the approximation, and information which specifies the amplitude of the time response at each of a number of equally spaced sample points in the time domain. Primary output data from the program contains the amplitude and phase of the approximate transfer function for the transducer system at equally spaced points in the frequency domain.

With minor modifications, the program is adaptable to any problem which requires the computation of a linear system's transfer function from its time response to a known input. The program has been tested with time response data for an analytically known two degree-of-freedom system excited by a step input. The results of these tests indicate that the accuracy of the approximate transfer function is determined by the data sampling frequency and the data range. For sampling frequencies greater than twenty times the highest frequency of interest and a data range which includes the time response out to a point at which the time response has reached 99% of its final value, the approximate transfer function was found to be within .5% of its known value.

Experimental results indicate that the time response data can be truncated at a time-point at which the time response has not yet reached 99% of its final value and reasonably accurate results can still be expected. If data is truncated, the experimental results indicate that the maximum error can be expected to occur at frequencies in the neighborhood of the system's natural frequencies. In general, the maximum error will occur in the amplitude characteristic of the transfer function if the input data is truncated.

The program is written to anticipate data truncation and, (if the user so elects) will seek a point for truncation which satisfies the following criteria:

(1) The point is on the positive going edge of an oscillation,

(2) The point is that point for which the amplitude of the time response is near the average (d-c) value of the time response data,

(3) The total number of data points is odd.

The first criterion assures that an integral number of cycles of oscillation exist in the data so that the computed average value of the time response data is based on an equal number of positive and negative oscillations about the average value. The second criterion is required by the assumptions made in the mathematical relations which define the approximation procedure, i.e., the last data point corresponds to the final value of the time response with all oscillations dampened out. The third criterion is imposed by Simpson's Rule which requires an even number of intervals, i.e., an odd number of data points, in the sampled time response. If the user elects to specify the final value of the transient response, it is assumed that the first two criteria have been satisfied. The program will always satisfy the third criterion by discarding the last data point if the given number of data points is an even number.

The program removes zero-offset in the time response data prior to the computation of the approximate transfer function. All input data is echo-checked by the program both before and after the final value has been determined and the zero-offset has been removed.

Program running time can be estimated from the relation

$$\text{Running Time} = (\text{NBRPTI})\,(\text{NBRPTO})\,(1.7)\,(10^{-5})\text{ minutes}$$

where NBRPTI is the number of time response data points used and NBRPTO is the number of frequency response data points to be computed. For 1501 time response data points and 201 frequency response points, running time is about 5 minutes.

The program is dimensioned for 2000 input data points and an unlimited number of output points and utilizes approximately 6000 words of storage including data storage.

PREPARATION OF INPUT DATA

Input data for the program contains three output identification cards, one test data card, one input/output parameter card, and the time response data cards. The detailed formats for these cards are described in the Input Card Format section of this manual. The procedure for data preparation starts with the FSADC transient waveform digitizing operation. Preliminary editing of the time response data is necessary. The objectives of this editing are (1) to remove FSADC output cards which correspond to the oscilloscope trace prior to the arrival of the step pressure function at the transducer and (2) to correct erroneous data points which correspond to transparency imperfections or random errors.

If the transparency is treated by applying Kodak Opaque paint to transparency imperfections and to that portion of the transient which corresponds to the base line, then the digitizing process will result in few if any erroneous data points. It will still be necessary to remove that portion of the data which corresponds to the transparency base line, however all such data points will digitize as 0000 for the amplitude. Since the FSADC prints a paper tape as it punches digitized data point cards, the operator can readily determine the x-coordinate of the point at which the step function is sensed. All data point cards prior to that point must be removed from the card deck produced by the FSADC. The total number of cards removed should be carefully ccunted so that the total number of input data cards to the program can be specified. The FSADC system always digitizes a point at $x = 000$ and $x = 1000$, however the latter will appear as 000 on the corresponding data card due to FSADC limitations. The total number of data points will be 101, 201, 501, or 1001, depending upon FSADC switch settings during the digitizing phase.

Assuming that the transparency was treated as previously specified, the data editing procedure is:

(1) Remove and count those data point cards which correspond to the base line of the waveform.

(2) Determine the number of cards which remain by subtracting the number of cards removed from the number of points digitized. For example, if 24 cards were removed and the number of points digitized was 501, the number of cards which remain would be 477.

(3) Scan the printed tape for erroneous data points. The tape will contain an asterisk (*) beside the values corresponding to such points and the corresponding data card will be blank in column 11. (Valid data cards will contain a 9-punch in column 11.) Since such erroneous points arise because either the FSADC sensed two or more traces when it scanned the x-position or the FSADC failed to sense any trace, the user must use judgment in correcting such points. Frequently, no correction will be necessary if the FSADC sensed the second trace at a point which was above the actual waveform trace. By examining the values digitized adjacent to the erroneous point, the user can decide whether or not the point does require correction. If correction is necessary, linear interpolation (using the amplitude values at adjacent points) should be applied in estimating the correct value. Cards which require correction should be removed from the data deck and marked with the corrected value. Preferably, the corrected data card should be keypunched immediately and the card reinserted in its proper position in the data deck.

(4) The edited data deck should be listed prior to its use as input data to the program and this listing checked for missing data points, card handling errors, and keypunch errors. Upon verification of this deck, the user should mark the deck with such information as the number of cards in the deck and any identification which is considered to be adequate for uniquely identifying

the deck. The first and last cards of the deck should be marked "1st Card" and "Last Card". This deck will be referred to as the "FSADC Deck".

Upon completing the preceding steps, the user should prepare the following header cards:

(1) OID-1

(2) OID-2

(3) OID-3

(4) STTD-1

(5) DD-1

The format and content of these header cards are defined in the Input Data Card Format section. It is important that the user observe the units of each quantity specified on the header cards, position each quantity within the card columns (field) assigned to that quantity, and keypunch the decimal point with each quantity. The location of a quantity within its field is arbitrary if the decimal point is punched. The only quantity which must not have a decimal point punched is NBRPTI, i.e., the number of points supplied in the FSADC Deck. The value specified for NBRPTI must be positioned in columns 7-10 of the DD-1 card so that the assumed decimal point position is to the right of column 10.

INPUT DATA CARD FORMAT

Input data cards to the Transfer Function Approximation Program are:

(1) Output Identification Cards — (OID-1, OID-2, OID-3)

(2) Shock Tube Test Data Card — (STTD-1)

(3) Digitized Data Card — (DD-1)

(4) Flying Spot Analog-to-Digital Converter Cards — (FSADC)

Detailed formats for these cards are specified in the Input Data Card Format Layout tables. A brief description of the purpose of each of the above card types follows:

(1) OID-1, OID-2, OID-3 These cards provide the means for identifying each output page with any arbitrary identification. Such information as transducer manufacturer, transducer serial number, date, remarks, etc., may be entered on these cards.

(2) STTD-1 This card specifies information relative to the shock tube test conditions and transient waveform recording parameters. The program utilizes this information in computing shock wave velocity, Mach number, pressure step size, and data sampling rate.

(3) DD-1 This card specifies information relative to the digitizing phase parameters and range of transfer function approximation.

(4) FSADC These cards are the output cards from the FSADC system that remain after preliminary editing by the user. Each card represents a digitized data point on the transient waveform and contains the transducer system identification in addition to the x and y coordinates of the point.

OUTPUT DATA CARD FORMAT

Output cards from the Transfer Function Approximation Program are:

(1) Approximate Transfer Function Header Card — (ATFH)

(2) Approximate Transfer Function Cards — (ATF)

Detailed formats for these cards are specified in the Output Data Card Format Layout tables. A brief description of the purpose of each of the above card types follows:

(1) ATFH This card contains transducer identification, the number of ATF cards, the bandwidth covered by the ATF cards, the frequency increment between sampled points, and the allowable time interval for approximating input time functions.

(2) ATF Each card defines the approximate transfer function for the transducer at a discrete frequency in the band width specified on the ATFH card.

The above cards are utilized as input data to the Input Time Function Approximation Program. Their contents also appear in the output listings generated by the Transfer Function Approximation Program.

PROGRAM OPERATING PROCEDURES

The order of input data cards for the Transfer Function Approximation Program is (for each transducer):

(1) OID-1

(2) OID-2

(3) OID-3

(4) STTD-1

(5) DD-1

(6) FSADC Deck

OUTPUT DATA CARD FORMAT LAYOUT

ABBREVIATED CARD NAME	CARD COLUMNS**	PROGRAM VARIABLE NAME	FORTRAN FORMAT	DEFINITION
ATFH	1-8	IDOUT(1) IDOUT(2)	2I4	Transducer identification as contained in columns 1-8 of FSADC DECK
"	9-20	NBRPTO	I12	Number of points for which the transfer function is computed
"	21-30	BEFREQ	F10.0	Lowest frequency for which transfer function is computed
"	31-40	FIFREQ	F10.0	Highest frequency for which transfer function is computed
"	41-50	FREQIN	F10.0	Frequency increment used in computing transfer function
"	51-60	SAMP2	F10.8	Allowable time interval for computing input time function
ATF	1-8	IDOUT(1) IDOUT(2)	2I4	Transducer identification as contained in columns 1-8 of FSADC DECK
"	9-20	FREQ	F12.0	Frequency value
"	21-35	AMPLIT	F15.5	Amplitude
"	36-50	PHASED	F15.5	Phase — Degrees
"	51-65	THETA	F15.5	Theta — Degrees

** Unlisted columns are blank

The above card deck is referred to as a Transducer Data Deck. An unlimited number of these decks can be processed with a single run on the IBM 7090 by stacking the decks in the desired order of processing. The first deck must always be preceded by a card with an asterisk (*) punched in column 1 and the word, DATA, punched in columns 7-10. The last deck must be followed by an end-of-file card, i.e., a card with a 7 and 8 punch in column 1 and the end-of-file card must be followed by a card with an asterisk in column 1 and the words, END TAPE, in columns 7-14.

The user should consult with the computing installation staff for the card formats they have established for the FMS (Fortran Monitor System) date and identification cards. These two cards must precede the TFAP Binary Deck (i.e., the TFAP program deck in column-binary form). The user must assemble the preceding cards and decks into the following order for a computer run:

(1) FMS Date Card

(2) FMS Identification Card

(3) TFAP Binary Deck

(4) * Data Card

(5) Transducer No 1 Data Deck

(6) Transducer No 2 Data Deck

(7) Transducer No N Data Deck

(8) End-of-File Card

(9) * End Tape Card

The TFAP program reads all input from logical tape unit No 5, writes all output for listing purposes on logical tape unit No 6, and writes all output for card punching on logical tape unit No 11. All input and output is in BCD form. The program will write an End File mark behind each output on logical tape unit No 11. The total number of such End File marks is equal to the total number of Transducer Data Decks submitted by the user for the computer run. The user must specify the preceding information to the computing installation staff.

The TFAP program makes use of the on-line printer during processing. When called upon to compute the estimated final value of the transducer time response data, the program will print the results of each value estimated during the iteration process until the estimation criteria has been satisfied. Generally, no more than 15 to 20 lines of output should occur during the estimating phase, however the user's experience with the program will determine what is reasonable. The Program Organization section should be consulted for further details on the estimating process. Following a successful estimation, the program will print on-line an estimated running time figure for the Transducer Data Deck currently being processed. The user should advise the computing installation staff of the approximate value to expect for each transducer since an error in card-handling or data preparation will usually result in either a ridiculous time estimate or excessive processing time by the program. This information will tend to reduce the possibility of unproductive computer runs due to card-handling and/or key-punching errors.

Upon completion of a transfer function approximation for each Transducer Data Deck, the program will print on-line that it has completed the approximation. This print should occur within the time period which began with the printing of the estimated running time.

PROGRAM ORGANIZATION

It is assumed that the user is familiar with the Mathematical Analysis section of this report. The basis for the computational procedures utilized in the Transfer Function Approximation Program are developed and discussed in that section in Volume I.

The block diagram for the program is presented in this section. The reader should refer to the Program Listing section for the program details which represent each block. The following conventions are established as an aid to associating the block diagram with the corresponding FORTRAN statements:

(1) Where a block is numbered, that number corresponds to the FORTRAN statement number which begins the sequence of operations defined in the block. The block number will be outside the block at the upper left hand corner.

(2) Input and output blocks will contain the corresponding FORTRAN format statement number in the lower left hand corner of the block. The number of the logical tape unit being used as the input or output unit will appear in the lower right hand corner of the block.

(3) Each FORTRAN statement card is numbered in columns 75 through 80. The beginning and ending card number for a sequence of operations represented in the block diagram will appear outside the corresponding block at the upper and lower right hand corners of the block, respectively.

The major computational processes in the program are:

(1) Final Value Estimation

(2) Transfer Function Approximation

The program always checks to see if the number of raw time response data points is an odd number. If not, the last data point is discarded and the total input data point count is reduced by one. The program then checks the value given by the user on the DD-1 card for the final value. If the given value is zero, the program assumes that the final value must be estimated. Prior to beginning the estimation, the program checks for a positive slope on the portion of the time response defined by the last two time response points. If the slope is not positive, the program discards the last two data points and reduces the total input data point count by two. The slope-check is repeated and points are discarded by two's until the program determines that the last two points do define a positive slope. The program then determines the average value of the remaining raw time response data by numerically integrating the data and dividing by the time range defined by the data. The average value is compared to the last time response data point. If the average value differs from the last time response data point by an amount which is less than the difference between the last data point and the second from the last data point, the program assumes that the final value of the time response data is the computed average value. If this difference criteria is not satisfied, the program discards the last two data points and returns to the section which checks for positive slope. The program will print on-line (1) the value of the second from last data point, (2) the computed average value, and (3) the value of the last data point, prior to returning to the slope-check section. Since all numerical integration performed in the program is based on Simpson's Rule, data points must be discarded by two's to preserve the "even number of intervals" requirement (i.e., an odd number of points are required when Simpson's Rule is used). This method of estimating the final value resulted in an estimated final value of .9991 versus an actual final value of .9998 or about .07% error for the known two-degree-of-freedom test case.

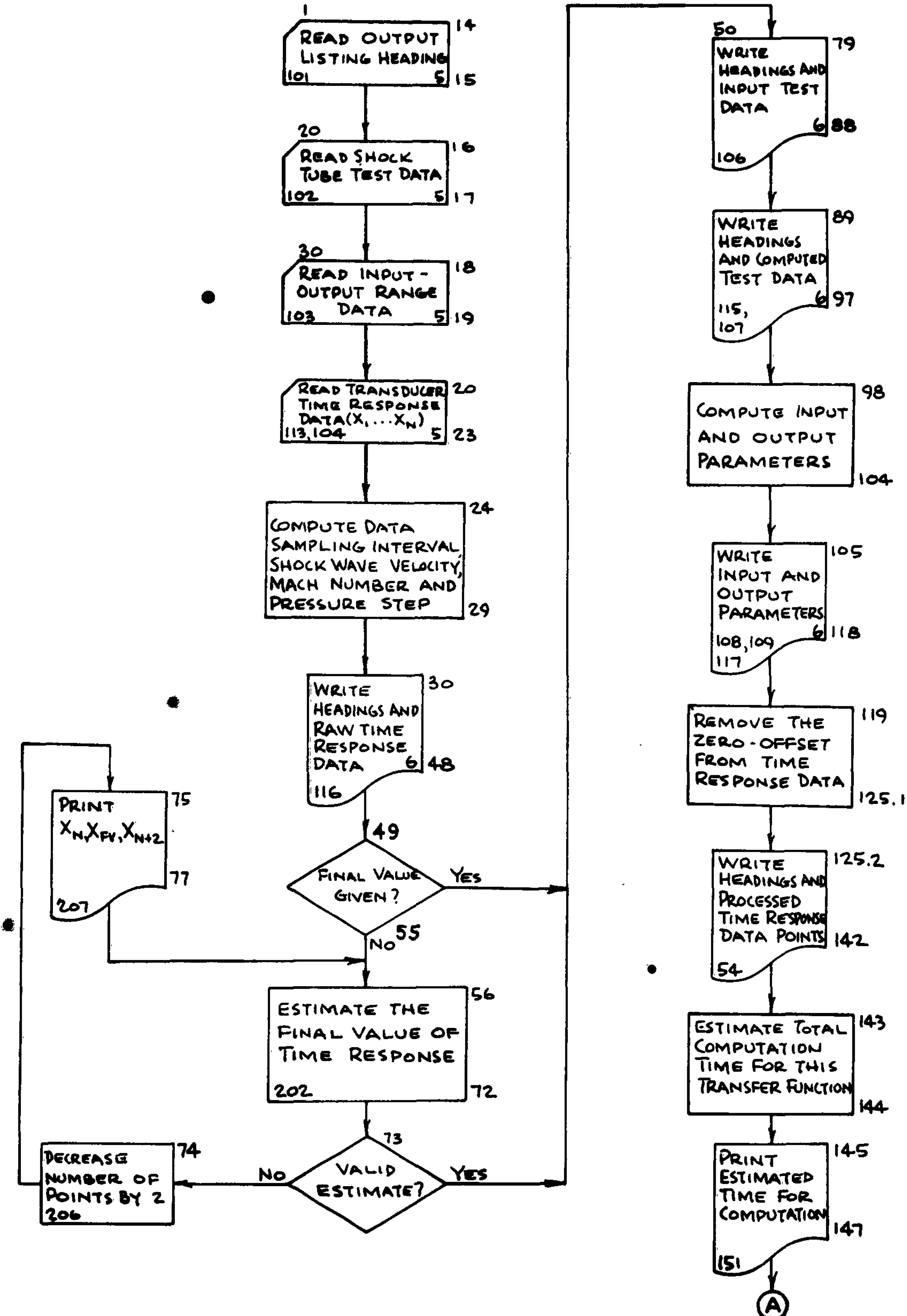
The section of the program which computes the approximate transfer function is based on the relations defined in Equations (44) through (50) of the Mathematical Analysis section of the report. The integrals are evaluated numerically by applying Simpson's Rule. It should be noted that the phase angle for the approximate transfer function is determined in the program from the relation

$$\phi = \tan^{-1} Y/X - 270^{\circ}$$

and not the relation defined in Equation (50). Experimental results with the two degree-of-freedom system indicate that a phase error will occur if the $\theta/2$ angle is included, and that this phase error is approximately equal to the angle, $\theta/2$. It appears that the source of this error is in the approximation made for the Fourier Transform of a step function. The program also includes a provision for completing the angle, $\tan^{-1}Y/X$, when it exceeds 270°. This provision forces the phase angle to always be developed as a negative angle.

A major part of the program is concerned with the details of the housekeeping that are necessary for obtaining the output formats and data. It should be noted that the program echo-checks the raw time response data and follows this with the processed time response data, i.e., the time response data with zero offset removed.

TRANSFER FUNCTION APPROXIMATION PROGRAM BLOCK DIAGRAM



The preparation and editing of time response data is a function of the system being used to generate and digitize this data. If the FSADC is being used, the preparation and editing procedures are the same as those described in the TFAP write-up. Regardless of the system being used to obtain this data, the card format for time response data points must be identical to the FSADC card format which is also described in the TFAP write-up.

The user must prepare the three output identification cards and the input/output parameter card in accordance with the formats defined in the Input Data Card Format section.

INPUT DATA CARD FORMAT

Input data cards to ITFAP are:

(1) Output Identification Cards — (OID-1, OID-2, OID-3)

(2) Approximate Transfer Function Header Card — (ATFH)

(3) Approximate Transfer Function Cards — (ATF)

(4) Input/Output Parameter Card — (IOP)

(5) Time Response Data Cards — (TRD)

The detailed card format for the OID cards is described in the TFAP write-up as are the ATFH and ATF cards. The TRD cards must have the same format as the FSADC cards described in the TFAP write-up, however the contents of columns 1-10 of these cards should contain time response data run identification. The IOP card's detailed format is specified in the Input Data Card Layout table. A brief description of the information contained on this card follows. It specifies the time response data run identification, the number of time response data points, the highest frequency component assumed to exist in the input time function, the time response data sampling interval, the time increment to be used between successive input time function points, the time interval for approximation of the input time function, and the conversion constant for converting time response amplitude values to pressure.

PROGRAM OPERATING PROCEDURES

The order of input data for ITFAP is:

(1) OID-1

(2) OID-2

(3) OID-3

(4) ATFH

INPUT DATA CARD FORMAT LAYOUT

ABBREVIATED CARD NAME	CARD COLUMNS**	PROGRAM VARIABLE NAME	FORTRAN FORMAT	DEFINITION
OID-1 OID-2 OID-3	SEE TFAP WRITE-UP — INPUT DATA CARD FORMAT LAYOUT TABLES			
ATFH	SEE TFAP WRITE-UP — OUTPUT DATA CARD FORMAT LAYOUT TABLES			
ATF	" " " " " " " " " "			
OIP	1-10	IDRUN(1) IDRUN(2)	2I5	Identification for Time Response Data Run
"	16-20	NBRPTI	I5	Number of Time Response Data Cards supplied as input to ITFAP
"	21-30	FREQLM	F10.0	Highest Frequency Component assumed to exist in the input time function, in cps.
"	31-40	DELTI	F10.0	Time Increment between successive time response data points, in microseconds.
"	41-50	DELTO	F10.0	Time Increment to be used in computing successive input time function data points, in microseconds.
"	51-60	TIME	F10.0	Time Interval over which the input time function is to be computed, in seconds.
"	61-70	CONVER	F10.5	Conversion constant to be used in converting a time response value to pressure, in psi/unit.
TRD	SEE TFAP WRITE-UP — INPUT DATA CARD FORMAT LAYOUT TABLES, FSADC CARDS			

** Unlisted columns are blank

(5) ATF Deck

(6) IOP

(7) TRD Deck

Any number of input data decks can be processed with a single run on the IBM 7090 by assembling each deck in the above order and stacking the assembled decks in the desired order of processing. The first such deck must be preceded by an *Data card and the last such deck must be followed by an End-of-File and End Tape card. These cards are described in the TFAP write-up.

The user should consult with the computing installation staff for the card formats they have established for the FMS (FORTRAN Monitor System) date and identification cards. These two cards must precede the ITFAP Binary Deck, i.e., the ITFAP program in column-binary form. The user must assemble the preceding cards and decks into the following order for a computer run;

(1) FMS Date Card

(2) FMS Identification Card

(3) ITFAP Binary Deck

(4) * Data Card

(5) Data Decks

(6) End-of-File Card

(7) * End Tape Card

The ITFAP program reads all input data from logical tape unit No 5 and writes all output for listing purposes on logical tape unit No 6. All input and output is in BCD form.

ITFAP makes use of the on-line printer during processing. The program prints an estimate of the running time for each data deck prior to processing the data deck. The user should advise the computing installation staff of the approximate value to expect for each data deck since an error in card handling or data preparation will usually result in either a ridiculous time estimate or excessive processing time. The program prints an on-line estimate as processing of each data deck is completed and this comment should appear within the time which has lapsed since the running time estimate was printed.

PROGRAM ORGANIZATION

It is assumed that the user is familiar with the Mathematical Analysis section of this report and the TFAP write-up. The basis for the computational procedures utilized in the approximation of input time functions are developed and discussed in the former and the block diagram conventions are established in the latter.

The major computational processes in the program are:

(1) Input Function Spectrum Computation

(2) Input Time Function Approximation

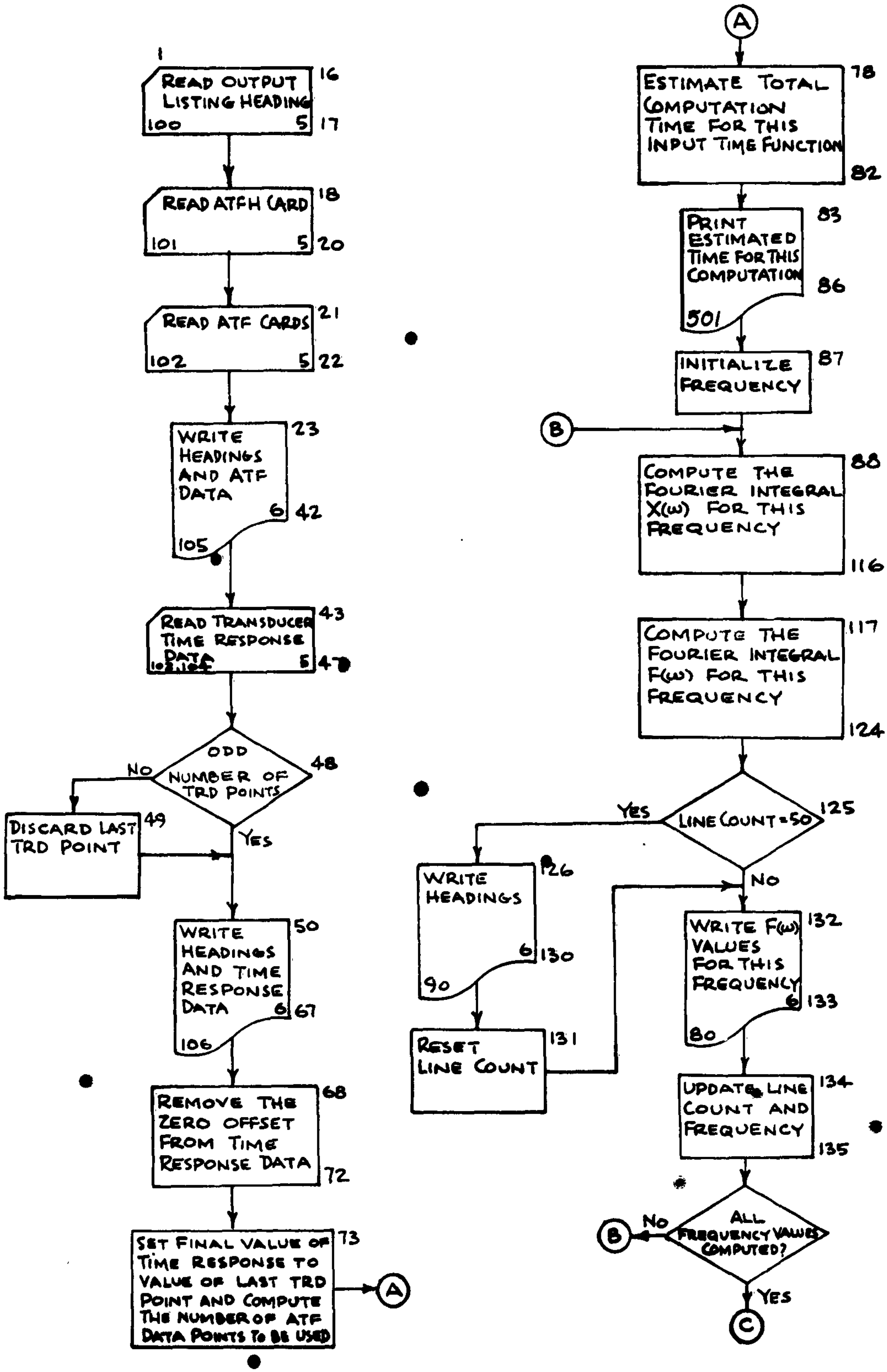
The program always checks to see if the number of time response data points is an odd number and will discard the last data point if not. This requirement is due to the method of numerical integration utilized in the program (Simpson's Rule). After echo-checking the input data, the program removes the x and y zero offset in the time response data and estimates the running time. The latter will appear in an on-line comment. This estimate is conservative and if the running time exceeds this figure, it is reasonable to assume that a malfunction has occurred due to incorrect data preparation.

The program computes the frequency spectrum for the input time function by computing the frequency spectrum for the time response data over the range specified by the user, and multiplying this spectrum by the inverse transfer function. Since the transfer function is known at discrete points only, the time response spectrum is computed at the same points only. Usually, the magnitude of the high frequency components of the input time function will be small relative to the lower frequency components. If so, computation time can be reduced by the user through specifying the limiting frequency at the lowest feasible value. The user may view this part of the program as a low-pass filter whose bandwidth is equal to the difference between FREQLM (Specified on IOP header card in ITFAP data) and BEFREQ (originally specified on the DD-1 header card in TFAP data). It is interesting to note that the program can be readily modified to act as a high-pass, band-pass, or notched filter with sharp roll-off characteristics. The program evaluates the frequency spectrum for the time response data by numerical computation of the values defined in Equations (44) and (45), over the bandwidth specified in the input data and in increments defined by the approximate transfer function data. As each value of $X(j\omega)$ is computed, the program divides its amplitude by the amplitude of the approximate transfer function at that frequency to obtain the amplitude of $F(j\omega)$. The phase of $F(j\omega)$ is computed by computing the difference between the phase of $X(j\omega)$ and the phase of the approximate transfer function. These two results are used to resolve $F(j\omega)$ into its real and imaginary parts. The program stores these two parts for use during the input time function approximation phase.

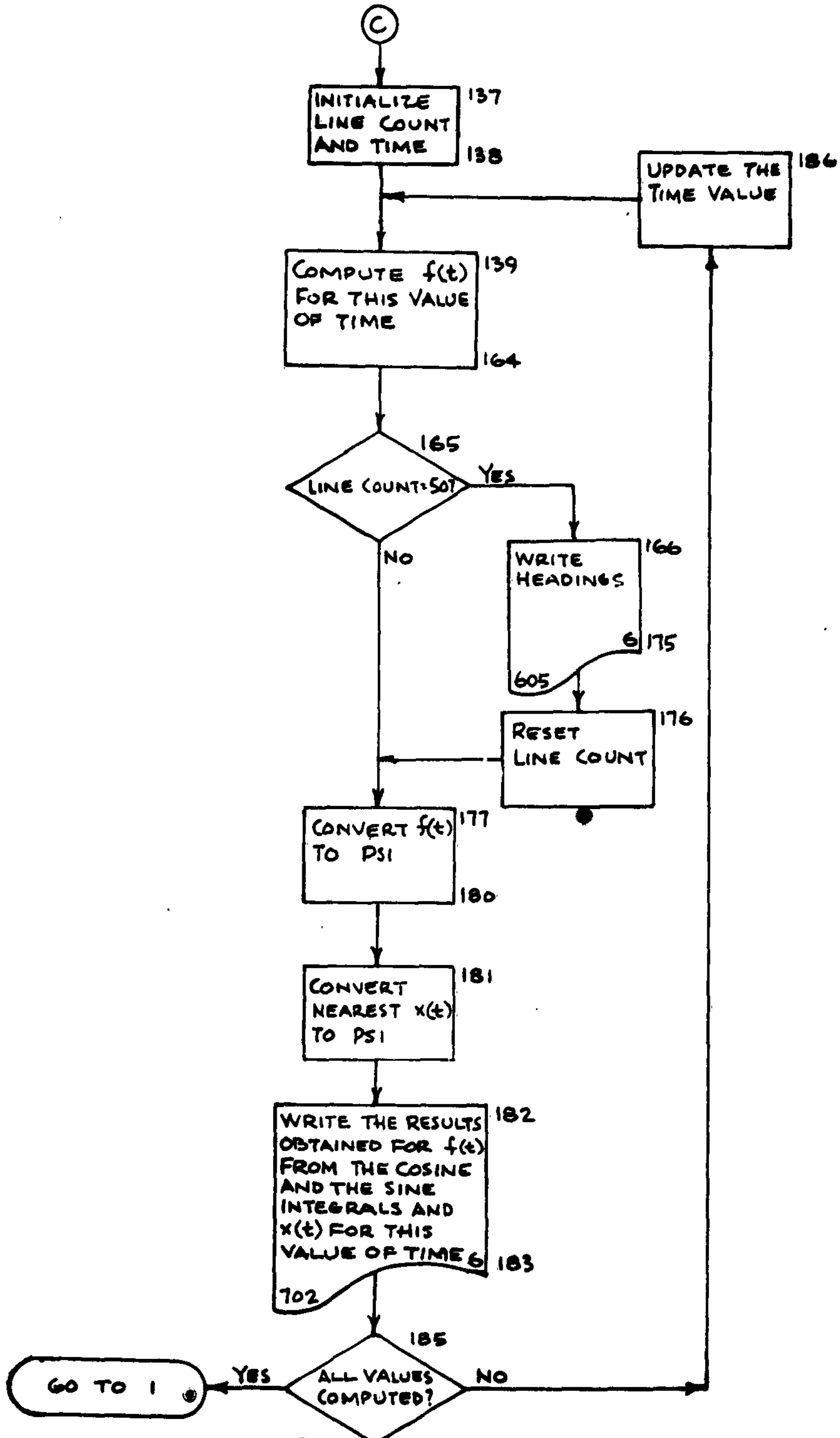
Upon completing the computation of the spectrum of $F(j\omega)$, the program proceeds to the section which computes the approximate input time beginning at $t = 0$. The computation is based on the relations given in Equations (61) and (62). Both equations are evaluated by numerical integration and the results from the two equations will be approximately equivalent if $F(j\omega)$ is Fourier transformable. In order to aid in the interpretation of results, each equation's result will appear in the output listing as will the nearest value of time response. Equation (61) is referred to as the "Cosine Integral" and Equation (62) is referred to as the "Sine Integral".

The user should note that the zero offset is removed from the time response data prior to its use in the computation, hence the conversion constant which is specified by the user should be given on a pressure per unit amplitude basis.

INPUT TIME FUNCTION APPROXIMATION PROGRAM
BLOCK DIAGRAM



INPUT TIME FUNCTION APPROXIMATION PROGRAM (CONTINUED)
BLOCK DIAGRAM



APPENDIX

```
*     HERCO PROJECT P-106-9- 2    AHM - 2-21-63   TFAP-02
*     XEQ
*     FORTRAN
C              TRANSFER FUNCTION APPROXIMATION PROGRAM
C                             PROGRAMMED FOR
C     6593D  TEST GROUP(DEVELOPMENT) - AIR FORCE FLIGHT TEST CENTER -
C                      EDWARDS AIR FORCE BASE, CALIFORNIA
C                             PROGRAMMED BY
C                   HOUSTON ENGINEERING RESEARCH CORPORATION
C                                 UNDER
C                   CONTRACT NUMBER AF 04(611) 8199
      DIMENSION XT(2000), HEDING(36),XCORD(2000),IDOUT(2),XARRAY(300),
     1YARRAY(300)
    1 READ INPUT TAPE 5,101,(HEDING(I),I=1,36)
  101 FORMAT(12A6/12A6/12A6)
   20 READ INPUT TAPE 5,102,P1,PA,TEMP,DIST,TIMINT,GAMMA,R,VERSEN,HORSWP
  102 FORMAT(9F8.0)
   30 READ INPUT TAPE 5,103,NBRPTI,XTFIN,      BEFREQ,FIFREQ,FREQIN
  103 FORMAT(I10,4F10.0)
      READ INPUT TAPE 5,113,IDOUT(1),IDOUT(2),XCORD(1),XT(1)
  113 FORMAT(2I4,9X,F3.0,F10.0)
   40 READ INPUT TAPE 5,104,(XCORD(I),XT(I),I=2,NBRPTI)
  104 FORMAT(15X,F5.0,F10.0)
      DELX = XCORD(2) - XCORD(1)
C              VELOCITY - MACH NBR - PRESSURE STEP COMPUTATION
      VEL = DIST*1.E6/TIMINT
      FMACH = VEL/SQRTF(32.2*GAMMA*R*(TEMP+460.))
      FM2 = FMACH*FMACH
      PSTEP = (P1+PA)*(2.3333333*(FM2-1.)*(4.*FM2+2.))/(FM2+5.)
      IBASE = 0
   23 IF(IBASE - NBRPTI) 24,28,28
   24 NPAGE = NBRPTI - IBASE
      IF(NPAGE - 300) 25,26,26
   25 LIM2 = NPAGE
      GO TO 27
   26 LIM2 = 300
   27 CALL HEDOUT(IBASE,HEDING)
      DO 188 I = 1, LIM2
      K = I + IBASE
      XARRAY(I) = XCORD(K)
  188 YARRAY(I) = XT(K)
      WRITE OUTPUT TAPE 6,116,IDOUT(1),IDOUT(2),(XARRAY(I),YARRAY(I),I=1
     1,LIM2)
  116 FORMAT(1H0,49HRAW INPUT TIME RESPONSE DATA FOR TRANSDUCER I.D. ,
     12I4,/6(2F10.0))
      IBASE = IBASE + LIM2
      GO TO 23
   28 CALL HEDOUT(1,HEDING)
C           FINAL VALUE ESTIMATION
```

```
      CXTFIN=XTFIN
      NBRPT=NBRPTI
      IF(NBRPT  - (NBRPT /2)*2)301,300,301
  300 NBRPT  = NBRPT  -1
  301 CONTINUE
      IF(XTFIN)50,200,50
  200 PRINT 208,IDOUT(1),IDOUT(2)
  208 FORMAT(1H1,42HCOMPUTED FINAL VALUES FOR TRANSDUCER I.D. ,2I4/1H0,
     110X,10HXT(LAST-2),7X,13HAVERAGE VALUE,12X,8HXT(LAST))
  202 BASE=NBRPT-1
      LIM=BASE
      DIF1=XT(NBRPT)-XT(LIM-1)
      IF(DIF1)203,201,201
  203 NBRPT=NBRPT-2
      GO TO 202
  201 SUMOD=0.
      SUMEV=0.
  204 DO 205 I=2, LIM, 2
      SUMOD=SUMOD+XT(I)
  205 SUMEV=SUMEV+XT(I+1)
      SUMEV=SUMEV-XT(NBRPT)
      CXTFIN=(XT(1)+XT(NBRPT)+4.*SUMOD+2.*SUMEV)/(3.*BASE)
      DIF2=ABSF(XT(NBRPT) - CXTFIN)
      IF(DIF1-DIF2)206, 50, 50
  206 NBRPT=NBRPT-2
      PRINT 207,XT(NBRPT),CXTFIN,XT(NBRPT+2)
  207 FORMAT(1H , 3F20.5)
      GO TO 202
   50 WRITE OUTPUT TAPE 6, 444
  444 FORMAT(1H0/1H0/1H0/1H0/1H0/1H0/)
             ECHO CHECK AND PRELIMINARY OUTPUT
      WRITE OUTPUT TAPE 6,106,P1,PA,TEMP,DIST,TIMINT,R,GAMMA,VERSEN,HORS
     1WP
  106 FORMAT(1H0,48X,23HECHO CHECK OF TEST DATA/1H0,23HTEST SECTION PRES
     1SURE =,F10.3,5H PSIG,21X,22HATMOSPHERIC PRESSURE =,F10.3,5H PSIA/1
     2H0,13HTEMPERATURE =,F10.3,6H DEG-F,30X,33HDISTANCE BETWEEN VELOCIT
     3Y GAGES =,F10.3,5H FEET/1H0,15HTIME INTERVAL =,F10.3,13H MICROSECO
     4NDS,21X,23HSPECIFIC GAS CONSTANT =,F10.3,17H FT-LBS/LBM-DEG-R/1H0,
     521HSPECIFIC HEAT RATIO =,F10.3,27X,23H VERTICAL SENSITIVITY =,F6.0
     6,14H MILLIVOLTS/CM/1H0,18HHORIZONTAL SWEEP =,F10.0,16H MICROSECOND
     7S/CM)
      WRITE OUTPUT TAPE 6,115
  115 FORMAT(1H0///50X,18HCOMPUTED TEST DATA)
      CONVER = 5.E2/(CXTFIN-XT(1))*PSTEP/VERSEN
      WRITE OUTPUT TAPE 6,107,VEL,FMACH,PSTEP,CONVER
  107 FORMAT(1H0,21HSHOCK WAVE VELOCITY =,F10.3,7H FT/SEC,21X,13HMACH NU
     1MBER =,F10.3,/1H0,25HREFLECTED PRESSURE STEP =,F10.3,4H PSI,20X,
     224HTRANSDUCER SENSITIVITY =,F10.3,14H PSI/MILLIVOLT/1H0///50X,
     323HINPUT/OUTPUT PARAMETERS)
```

```
      DELTI = DELX*HORSWP*1.E-8
      NBRPTO=(FIFREQ-BEFREQ)/FREQIN + 1.
      IF(NBRPTO-(NBRPTO/2)*2)401,400,401
  400 NBRPTO=NBRPTO+1
  401 CONTINUE
      FFN = NBRPTO - 1
      FIFREQ =FFN*FREQIN + BEFREQ
      WRITE OUTPUT TAPE 6,108,NBRPTI,NBRPT,DELTI ,BEFREQ,FIFREQ,FREQIN
  108 FORMAT(1H0,38HNUMBER OF INPUT DATA POINTS SUPPLIED =,I4,18X,34HNUM
     1BER OF INPUT DATA POINTS USED =,I4,/1H0,24HDATA SAMPLING INTERVAL
     2=,F11.8,13H SECONDS AND , 25HTRANSFER FUNCTION RANGE =,F6.0,2H -,F
     38.0,4H CPS,12H IN STEPS OF,F8.0,4H CPS)
      SAMP1=.5/DELTI
      SAMP2=.5/FREQIN
      WRITE OUTPUT TAPE 6,109,SAMP1,SAMP2
  109 FORMAT(1H0,34HSAMPLING THEOREM FREQUENCY LIMIT =,F10.0,8H CPS AND,
     138H INPUT TIME FUNCTION INTERVAL LIMIT = ,F10.8, 8H SECONDS)
      WRITE OUTPUT TAPE 6, 117, XT(NBRPTI),CXTFIN,XCORD(1),XT(1)
  117 FORMAT(1H0,22HRAW DATA FINAL VALUE =,F10.0,23H AND FINAL VALUE USE
     1D =,  F10.0,3X,15HZERO-OFFSET X =,F10.0,1X,15HZERO-OFFSET Y =,
     2F10.0)
      NBRPTI=NBRPT
      XCBASE = XCORD(1)
      XTBASE = XT(1)
      CXTFIN = CXTFIN - XTBASE
      DO 2 I=1,NBRPTI
      XCORD(I) = XCORD(I) - XCBASE
    2 XT(I) = XT(I) - XTBASE
      RANGE = 1.E8/(XCORD(NBRPTI)*HORSWP)
      WRITE OUTPUT TAPE 6,119,RANGE
  119 FORMAT(1H0,65HTHE TRANSFER FUNCTION MAY BE INACCURATE FOR FREQUENC
     1IES LESS THAN,F10.0,40H CPS IF TIME RESPONSE DATA WAS TRUNCATED)
      IBASE = 0
   63 IF(IBASE - NBRPTI)64,68,68
   64 NPAGE = NBRPTI - IBASE
      IF(NPAGE-300) 65,66,66
   65 LIM2 = NPAGE
      GO TO 67
   66 LIM2 = 300
   67 CALL HEDOUT(1,HEDING)
      DO 199 I = 1,LIM2
      K = I + IBASE
      XARRAY(I) = XCORD(K)
  199 YARRAY(I) = XT(K)
      WRITE OUTPUT TAPE 6,54,(XARRAY(I),YARRAY(I),I=1,LIM2)
   54 FORMAT(1H0,78HINPUT DATA USED IN APPROXIMATION OF TRANSFER FUNCTIO
     1N WITH T=0 OFFSET REMOVED /6(2F10.0))
      IBASE = IBASE + LIM2
      GO TO 63
   68 ESTIME = NBRPTO*NBRPTI
```

```
      ESTIME = ESTIME * 1.6E-5
      PRINT 151, ESTIME,IDOUT(1),IDOUT(2)
  151 FORMAT(1H1, 47HTHE NEXT ON LINE PRINT-OUT SHOULD OCCUR WITHIN ,F4.
     10,              24H MINUTES FOR TRANSDUCER ,2I4,/1H1)
      WRITE OUTPUT TAPE 11, 152,IDOUT(1),IDOUT(2),NBRPTO,BEFREQ,FIFREQ,
     1 FREQIN,SAMP2
  152 FORMAT(2I4,I12,3F10.0,F15.8)
      TRANSFER FUNCTION APPROXIMATION
      INIT = 50
      FREQ = BEFREQ
      DO 49 J=1,NBRPTO
      ANGLE = 6.2831853    *FREQ*DELTI
      SUMO1 = 0.
      SUMO2 = 0.
      SUME1 = 0.
      SUME2 = 0.
      LIM = NBRPTI - 1
      DO 3 I = 2,LIM,2
      FI = I-1
      TANG1 = FI * ANGLE
      SUMO1 = SUMO1 + XT(I)*COSF(TANG1)
      SUMO2 = SUMO2 + XT(I)*SINF(TANG1)
   12 FI = I
      TANG1 = FI*ANGLE
      SUME1 = SUME1 + XT(I+1)*COSF(TANG1)
    3 SUME2 = SUME2 + XT(I+1)*SINF(TANG1)
      SUME1 = SUME1 - XT(NBRPTI)*COSF(TANG1)
      SUME2 = SUME2 - XT(NBRPTI)*SINF(TANG1)
   10 FN = NBRPTI - 1
      TANG1 = FN*ANGLE
      YOPYN1 = XT(1)              + XT(NBRPTI)*COSF(TANG1)
      YOPYN2 =                      XT(NBRPTI)*SINF(TANG1)
      SUM1 = .33333333*(YOPYN1 + 4.*SUMO1 + 2.*SUME1)
      SUM2 = .33333333*(YOPYN2 + 4.*SUMO2 + 2.*SUME2)
      TANG1 = ANGLE*(FN+.5)
      FACT=1./SINF(.5*ANGLE)
      X = SUM1 - .5*      CXTFIN *SINF(TANG1)*FACT
      Y = -SUM2 - .5*      CXTFIN *COSF(TANG1)*FACT
      AMP = SQRTF(X*X+Y*Y)
      AMPLIT = 2./      CXTFIN *SINF(ANGLE*.5)*AMP
      PHASE = ARCTAN(Y,X)
      IF(PHASE - 4.7123889) 16,16,15
   15 PHASE = PHASE - 6.2831853
   16 PHASER = PHASE - 4.7123889
      PHASED = PHASER * 57.29578
      IF(INIT-50)502,501,501
  501 CALL HEDOUT(1,HEDING)
      WRITE OUTPUT TAPE 6,110,PSTEP
  110 FORMAT(1H0,22HUNIT PRESSURE STEP IS ,F4.0,4H PSI/1H ,13HFREQUENCY-
     1CPS,5X,9HAMPLITUDE,5X,9HPHASE-DEG,5X,9HPHASE-RAD,5X,9HTHETA-DEG,9X
```

```
     2,11HX-COMPONENT,9X,11HY-COMPONENT/1H )
      INIT=0
  502 THETA=ANGLE*57.29578
      WRITE OUTPUT TAPE6,111,FREQ,AMPLIT,PHASED,PHASER,THETA,X,Y
  111 FORMAT(1H ,F13.0,4F14.3,2E20.3)
      WRITE OUTPUT TAPE11,112,IDOUT(1),IDOUT(2),FREQ,AMPLIT,PHASED,THETA
  112 FORMAT(2I4,F12.0,3F15.5)
      INIT = INIT + 1
   49 FREQ = FREQ + FREQIN
      PRINT 114,IDOUT(1),IDOUT(2)
  114 FORMAT(34HOTRANSFER FUNCTION FOR TRANSDUCER ,2I4,19H HAS BEEN COMP
     1UTED.)
      END FILE 11
      GO TO 1
      END
*     FORTRAN
      SUBROUTINE HEDOUT( M,HEDING)
      DIMENSION HEDING(36)
      IF( M ) 802,801,802
  801 IPAGE = M
  802 IPAGE = IPAGE + 1
      WRITE OUTPUT TAPE 6, 803,IPAGE,(HEDING(I),I=1,36)
  803 FORMAT(1H1,57X,5HPAGE ,I2,/24X,12A6/24X,12A6/24X,12A6)
      RETURN
      END
*     FORTRAN
      FUNCTION ARCTAN(Y,X)
      IF (X) 1,2,3
    1 IF(Y)11,12,13
   11 R= 3.1415927
      GO TO 5
   12 R = 3.1415927
      GO TO 4
   13 R = -3.1415927
      GO TO 5
    2 IF(Y) 21,22,23
   21 R = 4.7123889
      GO TO 4
   22 R = 0.
      GO TO 4
   23 R = 1.5707963
      GO TO 4
    3 IF (Y) 31,32,33
   31 R = -6.2831854
      GO TO 5
   32 R = 0.
      GO TO 4
   33 R = 0.
      GO TO 5
    4 ARCTAN = R
```

```
      GO TO 6
    5 U = ABSF(Y/X)
      Q = (U-1.)/(U+1.)
      Q2 = Q*Q
      ALPHA = .78539815+Q*(.99999612+Q2*(-.33317376+Q2*(.19807869+Q2*
     1(-.13233510+Q2*(.07962632+Q2*(-.03360627+Q2*.00681241))))))
      ARCTAN = ABSF(R+ALPHA)
    6 CONTINUE
      RETURN
      END
      DATA
```

```
*     HERCO PROJECT P-106-9- 4    AHM - 2-20-63     ITFAP-04
*     XEQ
*     FORTRAN
C                 INPUT TIME FUNCTION APPROXIMATION PROGRAM
C                            PROGRAMMED FOR
C     6593D  TEST GROUP(DEVELOPMENT) - AIR FORCE FLIGHT TEST CENTER -
C                  EDWARDS AIR FORCE BASE, CALIFORNIA
C                            PROGRAMMED BY
C                 HOUSTON ENGINEERING RESEARCH CORPORATION
C                                UNDER
C                     CONTRACT NUMBER AF 04(611) 8199
      DIMENSION IDRUN(2),IDOUT(2),XCORD(5000),XT(5000),HEDING(36),
     1  HW(5000),QHW(5000),FWR(5000),FWI(5000),W(5000),XARRAY(300),
     2YARRAY(300),WARRAY(300)
      EQUIVALENCE(HW,FWR),(QHW,FWI),(XCORD,W)
    1 READ INPUT TAPE 5,100,(HEDING(I),I=1,36)
  100 FORMAT(12A6/12A6/12A6)
      READ INPUT TAPE 5,101,IDOUT(1),IDOUT(2),NBRPTO,BEFREQ,FIFREQ,
     1FREQIN,SAMPLM
  101 FORMAT(2I4,5X,I7,3F10.0,F15.8)
      READ INPUT TAPE 5,102,(W(I),HW(I),QHW(I),I=1,NBRPTO)
  102 FORMAT(10X,F10.0,2F15.5)
   86 IBASE = 0
   23 IF(IBASE-NBRPTO)24,11,11
   24 NPAGE = NBRPTO-IBASE
      IF(NPAGE-200)25,26,26
   25 LIM2=NPAGE
      GO TO 27
   26 LIM2 = 200
   27 CALL HEDOUT(IBASE,HEDING)
      DO 199 I = 1,LIM2
      K = I + IBASE
      WARRAY(I) = W(K)
      XARRAY(I) = HW(K)
  199 YARRAY(I) = QHW(K)
      WRITE OUTPUT TAPE 6,105,IDOUT(1),IDOUT(2),       BEFREQ,FIFREQ,
     1FREQIN,(WARRAY(I),XARRAY(I),YARRAY(I),I=1,LIM2)
  105 FORMAT(1H0, 33HTRANSFER FUNCTION FOR TRANSDUCER ,2I4,6X,F10.0,
     13H - ,F10.0, 10H CPS RANGE,7X,F10.0,15H CPS INCREMENTS/1H /
     24(F8.0,4HCPS=,F7.3,3H L ,F8.3))
      IBASE = IBASE + LIM2
      GO TO 23
   11 READ INPUT TAPE 5,103,IDRUN(1),IDRUN(2),NBRPTI,FREQLM,DELTI,DELTO,
     1TIME,CONAMP
  103 FORMAT(2I5,5X,I5,4F10.0,F10.5)
      READ INPUT TAPE 5, 104,(XCORD(I),XT(I),I=1,NBRPTI)
  104 FORMAT(14X,F6.0,F10.0)
      IF(NBRPTI -(NBRPTI/2)*2)28,85,28
   85 NBRPTI = NBRPTI-1
   28 IBASE = 0
```

```
 29 IF(IBASE-NBRPTI)31,35,35
 31 NPAGE = NBRPTI - IBASE
    IF(NPAGE - 300)32,33,33
 32 LIM2 = NPAGE
    GO TO 34
 33 LIM2 = 300
 34 CALL HEDOUT (1,HEDING)
    DO 188 I = 1, LIM2
    K = I + IBASE
    XARRAY(I) = XCORD(K)
188 YARRAY(I) = XT(K)
    WRITE OUTPUT TAPE 6,106,IDRUN(1),IDRUN(2),NBRPTI,DELTI,
   1(XARRAY(I),YARRAY(I),I=1,LIM2)
106 FORMAT(1H0,25HTIME RESPONSE DATA I.D. =,2I5,I12  ,11H POINTS AT ,
   1F10.3, 22H MICROSECOND INTERVALS / 6(2F10.0))
    IBASE = IBASE + LIM2
    GO TO  29
 35 XCBASE = XCORD(1)
    XTBASE = XT(1)
    DO 2 I = 1,NBRPTI
    XCORD(I) = XCORD(I) - XCBASE
  2 XT(I) = XT(I) - XTBASE
    XTFIN = XT(NBRPTI)
    NBRW = (FREQLM - BEFREQ)/FREQIN
    IF(NBRW-(NBRW/2)*2) 4,3,4
  3 NBRW = NBRW + 1
  4 FNBRW = NBRW
    FIFREQ = FNBRW*FREQIN
    INIT = 50
    FN = NBRPTI*NBRW
    ESTIME = FN*1.7E-5 + TIME/DELTO*FNBRW*1.6E-5
    NBRT = TIME/DELTO*1.E6+ 1.
    PRINT 501,IDOUT(1),IDOUT(2),IDRUN(1),IDRUN(2),ESTIME
501 FORMAT(1H0, 49HINPUT TIME FUNCTION APPROXIMATION FOR TRANSDUCER ,
   12I4, 24H AND TIME RESPONSE I.D. ,2I5,/1H , 38HNEXT ON LINE PRINT S
   1HOULD OCCUR WITHIN,F10.0, 8H MINUTES/1H1)
    FREQ = BEFREQ
    DO 49 J=1,NBRW
    ANGLE = 6.2831853E-6*FREQ*DELTI
    SUMO1 = 0.
    SUMO2 = 0.
    SUME1 = 0.
    SUME2 = 0.
    LIM = NBRPTI - 1
    DO13I = 2,LIM,2
    FI = I-1
    TANG1 = FI * ANGLE
    SUMO1 = SUMO1 + XT(I)*COSF(TANG1)
    SUMO2 = SUMO2 + XT(I)*SINF(TANG1)
 12 FI = I
```

```
      TANG1 = FI*ANGLE
      SUME1 = SUME1 + XT(I+1)*COSF(TANG1)
  133 SUME2 = SUME2 + XT(I+1)*SINF(TANG1)
      SUME1 = SUME1 - XT(NBRPTI)*COSF(TANG1)
      SUME2 = SUME2 - XT(NBRPTI)*SINF(TANG1)
   10 FN = NBRPTI - 1
      TANG1 = FN*ANGLE
      YOPYN1 = XT(1)               + XT(NBRPTI)*COSF(TANG1)
      YOPYN2 =                       XT(NBRPTI)*SINF(TANG1)
      SUM1 = .33333333*(YOPYN1 + 4.*SUMO1 + 2.*SUME1)
      SUM2 = .33333333*(YOPYN2 + 4.*SUMO2 + 2.*SUME2)
      TANG1 = ANGLE*(FN+.5)
      FACT=1./SINF(.5*ANGLE)
      X =(SUM1 - .5*      XTFIN *SINF(TANG1)*FACT)*1.E-6
      Y =(-SUM2 - .5*      XTFIN *COSF(TANG1)*FACT)*1.E-6
      AMP = SQRTF(X*X+Y*Y)
      FMAG = AMP/HW(J)
      FANG = ARCTAN(Y,X) - QHW(J)/57.2958
      IF(FANG-6.2831853)901,901,900
  900 FANG = FANG - 6.2831853
  901 CONTINUE
      FWR(J) = FMAG* COSF(FANG)
      FWI(J) = FMAG*SINF(FANG)
      QFANG = FANG*57.2958
      IF(INIT-50) 46,45,46
   45 CALL HEDOUT(1,HEDING)
      WRITE OUTPUT TAPE 6,90,IDOUT(1),IDOUT(2),IDRUN(1),IDRUN(2)
   90 FORMAT(1H ,36HINTERMEDIATE RESULTS FOR TRANSDUCER ,2I4, 24H AND TI
     1ME RESPONSE I.D. ,2I5/8X,4HFREQ,9X,4HMAGN,8X,5HANGLE,9X,4HREAL,
     29X,4HIMAG,12X,1HX,12X,1HY,9X,4HSUM1,9X,4HSUM2/1H )
      INIT = 0
   46 WRITE OUTPUT TAPE 6,80,FREQ,FMAG,QFANG,FWR(J),FWI(J),X,Y,SUM1,SUM2
   80 FORMAT(1H ,F11.0,8E13.4)
      INIT = INIT + 1
   49 FREQ = FREQ + FREQIN
C           INPUT TIME FUNCTION COMPUTATION
      INIT=50
      T=0.
  601 SUMO1=0.
      SUMO2=0.
      SUME1=0.
      SUME2=0.
      W=BEFREQ*6.2831853
      LIM = NBRW - 1
      DO 602 I=2,LIM,2
      FI1=I-1
      FI2=I
      ARG1=(FREQIN*6.2831853*FI1+W)*T
      ARG2=(FREQIN*6.2831853*FI2+W)*T
      SUMO1=SUMO1+FWR(I)*COSF(ARG1)
```

```
      SUME1=SUME1+FWR(I+1)*COSF(ARG2)
      SUMO2=SUMO2+FWI(I)*SINF(ARG1)
  602 SUME2=SUME2+FWI(I+1)*SINF(ARG2)
      T1=BEFREQ*6.2831853*T
      T2=FIFREQ*6.2831853*T
      T3=FWR(NBRW  )*COSF(T2)
      T4=FWI(NBRW  )*SINF(T2)
      SUME1=SUME1-T3
      SUME2=SUME2-T4
      YOPYN1=FWR(1)*COSF(T1)+T3
      YOPYN2=FWI(1)*SINF(T1)+T4
      DEL=FREQIN*(1.3333333   )*DELTI
      SUM1=DEL*(YOPYN1+4.*SUMO1+2.*SUME1)
      SUM2=-DEL*(YOPYN2+4.*SUMO2+2.*SUME2)
      IF(INIT-50)606,603,603
  603 CALL HEDOUT(1,HEDING)
      WRITE OUTPUT TAPE 6,605,IDOUT(1),IDOUT(2),IDRUN(1),IDRUN(2),FREQLM
     1,NBRT,DELTO,CONAMP,SAMPLM
  605 FORMAT(1H0,16X,44HCOMPUTED INPUT TIME FUNCTION FOR TRANSDUCER ,2I4
     1,24H AND TIME RESPONSE I.D. ,2I5/1H ,19HCUT-OFF FREQUENCY =,F10.0,
     2 4H CPS, I10,17H OUTPUT POINTS AT,F10.3 ,22H MICROSECOND INTERVAL
     3S/20H CONVERSION FACTOR =,F10.5,10X,39HALLOWABLE APPROXIMATION TIM
     4E INTERVAL =,F10.8,8H SECONDS/
     513X,17HTIME-MILLISECONDS,15X,15HCOSINE INTEGRAL,17X,13HSINE-INTEGR
     6AL,17X,13HTIME RESPONSE/1H )
      INIT=0
  606 TYME=T*1.E3
      K = T/DELTI*1.E6 + 1.
      SUM1 = SUM1*CONAMP
      SUM2 = SUM2*CONAMP
      TIRESP = XT(K)*CONAMP
      WRITE OUTPUT TAPE 6,702,TYME,SUM1,SUM2,TIRESP
  702 FORMAT(F30.3,3F30.5)
      INIT=INIT+1
      IF(T-TIME)610,611,611
  610 T=T+DELTO*1.E-6
      GO TO 601
  611 PRINT 600,IDOUT(1),IDOUT(2),IDRUN(1),IDRUN(2)
  600 FORMAT(1H , 49HINPUT TIME FUNCTION APPROXIMATION FOR TRANSDUCER ,
     12I4, 19H AND TIME RESPONSE ,2I5,10H COMPLETED/1H1)
      GO TO 1
      END
      FORTRAN
      SUBROUTINE HEDOUT( M,HEDING)
      DIMENSION HEDING(36)
      IF( M ) 802,801,802
  801 IPAGE = M
  802 IPAGE = IPAGE + 1
      WRITE OUTPUT TAPE 6, 803,IPAGE,(HEDING(I),I=1,36)
  803 FORMAT(1H1,57X,5HPAGE ,I2,/24X,12A6/24X,12A6/24X,12A6)
```

```
      RETURN
      END
*     FORTRAN
      FUNCTION ARCTAN(Y,X)
      IF (X) 1,2,3
    1 IF(Y)11,12,13
   11 R= 3.1415927
      GO TO 5
   12 R = 3.1415927
      GO TO 4
   13 R = -3.1415927
      GO TO 5
    2 IF(Y) 21,22,23
   21 R = 4.7123889
      GO TO 4
   22 R = 0.
      GO TO 4
   23 R = 1.5707963
      GO TO 4
    3 IF (Y) 31,32,33
   31 R = -6.2831854
      GO TO 5
   32 R = 0.
      GO TO 4
   33 R = 0.
      GO TO 5
    4 ARCTAN = R
      GO TO 6
    5 U = ABSF(Y/X)
      Q = (U-1.)/(U+1.)
      Q2 = Q*Q
      ALPHA = .78539815+Q*(.99999612+Q2*(-.33317376+Q2*(.19807869+Q2*
     1(-.13233510+Q2*(.07962632+Q2*(-.03360627+Q2*.00681241))))))
      ARCTAN = ABSF(R+ALPHA)
    6 CONTINUE
      RETURN
      END
*     DATA
```

Research & Technology Division, Edwards AF Base, Calif. Rpt. No. RTD-TDR-63-9. EVALUATION AND MODIFICATION OF EXISTING PROTOTYPE DYNAMIC CALIBRATION SYSTEMS FOR PRESSURE-MEASURING TRANSDUCERS (U). Final Report, Vol III - Computer Program Write-Up March 63, 33 p incl illus, tables.

Unclassified Report

This report presents an IBM 7090 FORTRAN program applicable to existing dynamic prototype calibration system for pressure measuring transducers at EAFB.

1 Instrumentation
I Project 3850
Task 38506
II Contract No
AF 04(611)-8199
III Houston Engineering Research Corporation
IV J L Schweppe
J L Williams
A H McMorris
W R Busby
V In ASTIA collection

Research & Technology Division, Edwards AF Base, Calif. Rpt. No. RTD-TDR-63-9. EVALUATION AND MODIFICATION OF EXISTING PROTOTYPE DYNAMIC CALIBRATION SYSTEMS FOR PRESSURE-MEASURING TRANSDUCERS (U). Final Report, Vol III - Computer Program Write-Up March 63, 33 p incl illus, tables.

Unclassified Report

This report presents an IBM 7090 FORTRAN program applicable to existing dynamic prototype calibration system for pressure measuring transducers at EAFB.

1 Instrumentation
I Project 3850
Task 38506
II Contract No
AF 04(611)-8199
III Houston Engineering Research Corporation
IV J L Schweppe
J L Williams
A H McMorris
W R Busby
V In ASTIA collection

Research & Technology Division, Edwards AF Base, Calif. Rpt. No. RTD-TDR-63-9. EVALUATION AND MODIFICATION OF EXISTING PROTOTYPE DYNAMIC CALIBRATION SYSTEMS FOR PRESSURE-MEASURING TRANSDUCERS (U). Final Report, Vol III - Computer Program Write-Up March 63, 33 p incl illus, tables.

Unclassified Report

This report presents an IBM 7090 FORTRAN program applicable to existing dynamic prototype calibration system for pressure measuring transducers at EAFB.

1 Instrumentation
I Project 3850
Task 38506
II Contract No
AF 04(611)-8199
III Houston Engineering Research Corporation
IV J L Schweppe
J L Williams
A H McMorris
W R Busby
V In ASTIA collection

Research & Technology Division, Edwards AF Base, Calif. Rpt. No. RTD-TDR-63-9. EVALUATION AND MODIFICATION OF EXISTING PROTOTYPE DYNAMIC CALIBRATION SYSTEMS FOR PRESSURE-MEASURING TRANSDUCERS (U). Final Report, Vol III - Computer Program Write-Up March 63, 33 p incl illus, tables.

Unclassified Report

This report presents an IBM 7090 FORTRAN program applicable to existing dynamic prototype calibration system for pressure measuring transducers at EAFB.

1 Instrumentation
I Project 3850
Task 38506
II Contract No
AF 04(611)-8199
III Houston Engineering Research Corporation
IV J L Schweppe
J L Williams
A H McMorris
W R Busby
V In ASTIA collection