FURTHER DEVELOPMENT OF THE
DYNAMIC GAS TEMPERATURE
MEASUREMENT SYSTEM

VOLUME II - COMPUTER PROGRAM
USER’S MANUAL

by
Dana R. Stocks
August 1986
Denny L. Elmore
Program Manager

Pratt & Whitney
Government Products Division
P.O. Box 109600
West Palm Beach, Florida 33410-9600

Prepared for
National Aeronautics and Space Administration
NASA-Lewis Research Center
Cleveland, Ohio 44135

DATE FOR GENERAL RELEASE FEBRUARY 1989
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>PHYSICAL MODEL</td>
<td>2</td>
</tr>
<tr>
<td>Probe Description</td>
<td>2</td>
</tr>
<tr>
<td>Data Acquisition and Processing Description</td>
<td>3</td>
</tr>
<tr>
<td>ANALYTICAL MODEL</td>
<td>4</td>
</tr>
<tr>
<td>Approach</td>
<td>4</td>
</tr>
<tr>
<td>Thermal Model</td>
<td>4</td>
</tr>
<tr>
<td>Overview of Compensation Procedure</td>
<td>9</td>
</tr>
<tr>
<td>DATA PREPROCESSING</td>
<td>15</td>
</tr>
<tr>
<td>Data Acquisition and Playback</td>
<td>15</td>
</tr>
<tr>
<td>Conversion from Digitized Data to Test Data Input File</td>
<td>16</td>
</tr>
<tr>
<td>DESCRIPTION OF THE COMPUTER PROGRAM</td>
<td>17</td>
</tr>
<tr>
<td>General Description</td>
<td>17</td>
</tr>
<tr>
<td>Functions Performed</td>
<td>19</td>
</tr>
<tr>
<td>Description of Subroutines</td>
<td>21</td>
</tr>
<tr>
<td>Subroutine Interaction</td>
<td>29</td>
</tr>
<tr>
<td>Common Blocks</td>
<td>29</td>
</tr>
<tr>
<td>INPUT/OUTPUT</td>
<td>34</td>
</tr>
<tr>
<td>Input Description</td>
<td>34</td>
</tr>
<tr>
<td>Output Description</td>
<td>37</td>
</tr>
<tr>
<td>PROGRAM PECULIARITIES AND RESTRICTIONS</td>
<td>38</td>
</tr>
<tr>
<td>Input Restrictions</td>
<td>38</td>
</tr>
<tr>
<td>Program Peculiarities</td>
<td>38</td>
</tr>
<tr>
<td>Restriction on Test Data</td>
<td>38</td>
</tr>
<tr>
<td>Accuracy of Results</td>
<td>41</td>
</tr>
<tr>
<td>PROGRAM LISTING</td>
<td>43</td>
</tr>
<tr>
<td>TEST CASES</td>
<td>103</td>
</tr>
<tr>
<td>Test Case 1</td>
<td>105</td>
</tr>
<tr>
<td>Test Case 2</td>
<td>109</td>
</tr>
<tr>
<td>Test Case 3</td>
<td>114</td>
</tr>
<tr>
<td>Test Case 4</td>
<td>117</td>
</tr>
<tr>
<td>Test Case 5</td>
<td>120</td>
</tr>
<tr>
<td>Test Case 6</td>
<td>122</td>
</tr>
<tr>
<td>Test Case 7</td>
<td>125</td>
</tr>
</tbody>
</table>
TABLE OF CONTENTS (Continued)

Program Execution on the TSS Operating System ........................................... 127

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set-Up</td>
<td>127</td>
</tr>
<tr>
<td>Digitizing Analog Data</td>
<td>127</td>
</tr>
<tr>
<td>Execution of GASTEMP</td>
<td>127</td>
</tr>
<tr>
<td>Creating MIG.NAME.DATA</td>
<td>128</td>
</tr>
<tr>
<td>Creating NAME.INPUT</td>
<td>128</td>
</tr>
<tr>
<td>TAPEINFO</td>
<td>128</td>
</tr>
<tr>
<td>Submitting to Background</td>
<td>129</td>
</tr>
<tr>
<td>Running GASTEMP as a Batch Job</td>
<td>129</td>
</tr>
<tr>
<td>Warning</td>
<td>129</td>
</tr>
<tr>
<td>Order of Execution</td>
<td>130</td>
</tr>
</tbody>
</table>

Appendix — Flowcharts ........................................... 132
INTRODUCTION

The Dynamic Gas Temperature Measurement System compensation software accepts digitized data from two different diameter, type B or K, thermocouples (T/C). The analysis method then determines the in situ value of an aerodynamic scaling parameter, \( \Gamma \), by comparing ratios of calculated dynamic response with ratios of measured dynamic response. The value of \( \Gamma \) identified determines an in situ heat transfer coefficient \( h_a \), and is used to compute a frequency response spectrum for one of the thermocouples.

Prior work in this area was performed under Contract NAS3-23154 and is reported in NASA Report NASA CR-168267, Volume II (Reference 1). The prior data analysis and compensation software was implemented on a digital computer based Hewlett Packard (HP) model 5451C Fourier Analyzer System. Under current efforts, this software has been rewritten and implemented for use on IBM mainframe computer systems which are more commonly available than the HP system.

Detailed discussions of the physical system, analytical model, and computer software are presented in this volume and in Volume I of this report under Task III activities. Computer program software restrictions and test cases are also presented. Compensated data are presented in either the time or frequency domain. Time domain data are presented as instantaneous temperature vs time (compensated or uncompensated) while frequency domain data may be presented in the forms shown in Table 1, below (compensated or uncompensated).

**TABLE 1. — DATA PRESENTATION FORMS**

<table>
<thead>
<tr>
<th>Function</th>
<th>Dimensions</th>
<th>Engineering Units</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Frequency Domain Presentations</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Power Spectral Density (PSD)</td>
<td>Mean Square/Hz</td>
<td>K^2/Hz</td>
</tr>
<tr>
<td>• Log Power Spectral Density</td>
<td>Mean Square/Hz</td>
<td>0 dB Ref 1 K^2/Hz</td>
</tr>
<tr>
<td>— 10 \times \log (PSD)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Linear Power Spectral Density</td>
<td>rms/\sqrt{Hz}</td>
<td>K/\sqrt{Hz}</td>
</tr>
<tr>
<td>— Positive Square Root of PSD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Narrowband Frequency Spectrum</td>
<td>rms</td>
<td>K</td>
</tr>
<tr>
<td>— Positive square root of auto-spectral density function with narrowband signal correction for Fast Fourier Transform (FFT) windowing function applied and no normalization to per unit bandwidth</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Time Domain Presentation</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Dynamic Temperature</td>
<td>Instantaneous</td>
<td>K</td>
</tr>
</tbody>
</table>
PHYSICAL MODEL

Probe Description

The dynamic temperature probe concept is shown in Figure 1. The probe employs two thermocouples of different wire diameters positioned in close proximity. The thermoelements are large enough in diameter that frequency response above a few Hz is limited by thermal inertia. When the thermocouples are exposed to the same instantaneous temperature and velocity in the gas stream, the difference in thermal responses will be governed by convective effects (proportional to wire diameter) and conductive effects (proportional to specific heat, thermal conductivity, and wire length). Many previous studies used thermoelements of sufficiently large ($\approx 100$) length-to-diameter ratio that conduction effects may be neglected, and compensations were based on first order convective time constants. The present sensor, however, is designed for engine hot-section applications, and the smaller length-to-diameter ratios required for structural adequacy necessitates inclusion of transient conduction effects in the compensation method.

![Dynamic Temperature Sensor Concept](https://via.placeholder.com/150)

**Figure 1. Dynamic Temperature Sensor Concept**

For engine tests, ISA type B thermocouples were chosen based on electromotive force (EMF) output, known fabrication characteristics, high melting point, and ready availability. Detailed structural analysis revealed that allowable yield stresses for platinum-rhodium alloys constrained length-to-diameter ratios to less than 10 for the support wires and less than 15 for the thermoelements. This probe design has demonstrated more than five hour durability in a high temperature (greater than 3000°F), atmospheric pressure laboratory combustor exhaust, and more than one hour in a high temperature (greater than 2000°F), high pressure (several atmospheres) gas turbine engine combustor exhaust.

A unique feature of each thermocouple is the beadless, butt-welded thermoelement. The beadless construction allows the sensor to be modelled as a cylinder in crossflow, which simplifies the model considerably.
Data Acquisition and Processing Description

During rig tests, data were collected on a frequency modulated (FM) magnetic tape recorder. The data were reproduced post-test in an off-line data processing center to produce a digital tape for input to a mainframe IBM computer. The data analysis routines were executed in the IBM computer. A detailed description and discussion of equipment used for the data acquisition, playback, and digitization is presented in the Task 1d section of Volume 1.
ANALYTICAL MODEL

Approach

Historically the problem of compensating wire thermocouples for frequency response rests on accurate determination of the in situ film heat transfer coefficient, \( h_s \). For wire thermocouples, three simplifications allow this to be done easily: (1) the thermocouple junction is fabricated without a bead, allowing the wire to be analyzed as a cylinder in crossflow, (2) two wires of different diameters may be co-located on the probe tip, and ratioing the different responses allows one to measure time constants \( \tau \), which are proportional to \( h_s \) for large \( L/D \) values; and (3) the thermocouple and support wires can be made long enough to eliminate conduction effects, facilitating the ratio analysis in (2). As shown in preceding reports such as Reference 1, however, probe durability requirements allowed only moderate \( L/D \) values, and initial calculations revealed that conduction effects should be included to meet accuracy goals.

The compensation approach used to include both conduction and convection effects involves ratioing signals of two different diameter beadless thermocouples. Heat transfer coefficient \( h_s \) is determined, however, by comparing finite-element conduction-convection calculations with experimental data. The calculations are done using \( h_s \) as a parameter, and require matching calculated thermocouple signal amplitude ratios at several discrete frequencies from \( \sim 6 \) to \( \sim 40 \) Hz. Agreement between calculated and experimentally observed signal amplitude ratios determines \( h_s \). For convenience, \( h_s \) is combined with other system parameters into an aerodynamic parameter, \( \Gamma \). The measured values of \( \Gamma \) obtained at specific frequencies are arithmetically averaged and used in the thermal model to compute the compensation frequency spectrum (gain and phase) for the smaller diameter thermocouple. The smaller diameter thermocouple should be selected for compensation since it has faster response and inherently better signal to noise characteristics at the higher frequency fluctuations. Compensation is performed digitally in the frequency domain by complex math division of the FFT spectrum of the thermocouple output by the compensation spectrum. The compensated time waveform is obtained by inverse Fourier transforming the compensated frequency spectrum.

Thermal Model

For the thermocouple probe modeled as a cylinder in crossflow, the basic thermal equation is:

\[
\frac{\partial T}{\partial t} = \frac{4h}{\rho_s C_{pw} D} (T_s - T) + a \frac{\partial^2 T}{\partial x^2} + \frac{4\alpha \varepsilon}{\rho_s C_{pw} D} (T_s^4 - T^4)
\]

(1)

Thermal analysis of the thermocouple model previously performed in Reference 1 showed that radiation effects could be neglected. It was also shown that neglecting conduction errors would typically introduce a 25 percent error (GAIN) in the compensated data. In the analysis that follows, only the radiation term from Equation 1 has been omitted.
The gas stream temperature can be expressed in terms of its mean and dynamic components as:

\[ T_s = T_0 + \sum_{n=1}^{\infty} a_n \sin (\omega_n t - \varphi_n) \]  

(2)

Substituting Equation 2 written for a single frequency for \( T_s \) in Equation 1 and normalizing, the transfer function between the temperature in the thermocouple wire and the gas stream temperature can be written as:

\[ \frac{\partial^2 \zeta}{\partial t^2} = \frac{4h}{\rho_c C_{pm} D} \left[ a_n \sin (\omega_n t - \varphi_n) - \zeta \right] + a \frac{\partial^3 \zeta}{\partial x^3} \]  

(3)

Where: \( \zeta = \frac{T_n - T_s}{T_s - T_0} = \frac{\theta_{in}}{a_n} \)  

(4)

The finite difference solution for Equation 3 is of the form:

\[ \frac{\partial \zeta}{\partial t} \approx \frac{\zeta_{n+1} - \zeta_n}{\Delta t} = \frac{\zeta_{n+1} - \zeta_n}{\Delta t} \]  

\[ \frac{\partial^2 \zeta}{\partial x^2} \approx \frac{\zeta_{n+1} - 2\zeta_n + \zeta_{n-1}}{\Delta^2} \]  

(5)

Application of the finite difference solution to Equation 3 for the 9 node model of the thermocouple in Figure 2 yields Equations 7 through 16.

\[ \zeta_0 = 0 \text{ (Assumed)} \]  

(7)

\[ \zeta_1 = \frac{C}{2A} [\zeta_0 + \zeta_4 - 2\zeta_3] + \zeta_1 \]  

(8)

\[ \zeta_2 = \frac{C}{2A} [\zeta_1 + \zeta_5 - 2\zeta_4] + \zeta_2 \]  

(9)

\[ \zeta_3 = \frac{1}{2A} [C(\zeta_2 + \zeta_4 - 2\zeta_3) + F(\sin C_s t - \zeta_0)] + \zeta_3 \]  

(10)

\[ \zeta_4 = \frac{1}{2A} [C(\zeta_3 + \zeta_5 - 2\zeta_4) + 2F(\sin C_s t - \zeta_4)] + \zeta_4 \]  

(11)

\[ \zeta_5 = \frac{1}{2A} [C(\zeta_4 + \zeta_6 - 2\zeta_5) + 2F(\sin C_s t - \zeta_5)] + \zeta_5 \]  

(12)

\[ \zeta_6 = \frac{1}{(A + B)} [C(\zeta_6 + \zeta_8 + 2\zeta_7) + E(\zeta_7 - \zeta_6) + (F + G)(\sin C_s t - \zeta_0)] + \zeta_6 \]  

(13)

\[ \zeta_7 = \frac{1}{2B} [E(\zeta_6 + \zeta_8 - 2\zeta_7) + 2G(\sin C_s t - \zeta_6)] + \zeta_7 \]  

(14)

\[ \zeta_8 = \frac{1}{2B} [E(\zeta_7 + \zeta_9 - 2\zeta_8) + 2G(\sin C_s t - \zeta_8)] + \zeta_8 \]  

(15)

\[ \zeta_9 = \frac{1}{B} [E(\zeta_8 - \zeta_9) + G(\sin C_s t - \zeta_9)] + \zeta_9 \]  

(16)
Figure 2. Finite Element Thermal Model — Used in Computer Compensation Program

- Definition of the dynamic temperature parameters is as follows:

\[ \theta_{1n} \] = Peak amplitude of smaller diameter thermocouple at frequency \( n \)

\[ \theta_1(f) = \theta_{1n} \text{ as a function of frequency} \]

\[ \theta_{2n} \] = Peak amplitude of larger diameter thermocouple at frequency \( n \)

\[ \theta_2(f) = \theta_{2n} \text{ as a function of frequency} \]

\[ a_n \] = Peak amplitude of the dynamic component of the gas stream temperature at frequency \( n \)

\[ a_n(f) = a_n \text{ as a function of frequency} \]

\[ \varphi_n \] = Phase shift of the gas temperature with respect to arbitrary time \( t_0 \) at frequency \( f_n \)
\( \eta_{1n} \) = Phase shift of smaller diameter thermocouple with respect to gas temperature at frequency \( f_n \)

\( \eta_1 (f) \) = \( \eta_{1n} \) as a function of frequency

\( \eta_{2n} \) = Phase shift of larger diameter thermocouple with respect to gas temperature at frequency \( f_n \)

\( \eta_2 (f) \) = \( \eta_{2n} \) as a function of frequency

\( j \) = Time index

\( x \) = Spacial coordinate along length of thermocouple

\( T_n \) = Instantaneous temperature of thermocouple wire at frequency \( n \)

\( T_{1...9n} \) = Instantaneous temperature of thermocouple wire at spacial location at frequency \( n \)

\( T_{x \text{ peak}} \) = Maximum peak in instantaneous temperature of thermocouple wire at spacial location \( x \)

\( T_g \) = Instantaneous gas stream temperature

\( T_{gn} \) = Instantaneous gas stream temperature at frequency \( n \)

\( h^+ \) = Convective film coefficient of thermocouple element

\( h^- \) = Convective film coefficient of thermocouple support wire

\( \sigma \) = Boltzmann constant

\( \alpha \) = Thermal diffusivity of the wire

\( \alpha = \frac{k_w}{\rho_w C_{pw}} \)

\( L \) = Length of larger diameter thermocouple support wire

\( \ell \) = One half of length of smaller diameter thermocouple wire

\( D \) = Diameter of larger diameter support wire

\( d \) = Diameter of smaller diameter thermocouple wire

\( \rho_w \) = Density of thermocouple wire

\( k_w \) = Thermal conductivity of thermocouple wire

\( C_{pw} \) = Specific heat of thermocouple wire

\( \rho_g \) = Density of gas stream
\( k_g \) = Thermal conductivity of the gas stream  
\( C_{pg} \) = Specific heat of the gas stream  
\( Pr_g \) = Prandtl number of gas stream  
\( U_g \) = Velocity of the gas stream  
\( \mu_g \) = Viscosity of the gas stream  
\( \gamma_g \) = Ratio of specific heats of gas stream  
\( P \) = Mean gas stream pressure  
\( M_n \) = Mach number  
\( f_1 \rightarrow f_x \) = Frequencies of \( f_n \) at which transfer functions will be evaluated  
\( F/A \) = Fuel air ratio  
\( A = \frac{D^2 \Delta}{8a(\Delta t)} \)  
\( B = \frac{d^2 \delta}{8a(\Delta t)} \)  
\( C = \frac{D^2}{4\Delta} \)  
\( C_n = 2\pi f_n \)  
\( \delta = \ell/3 \)  
\( E = \frac{d^2}{4\delta} \)  
\( F = \frac{h \cdot D\Delta}{2K_w} = \frac{\Gamma D^{1/2}}{2a} \)  
\( G = \frac{h \cdot d\delta}{2K_w} = \frac{\Gamma d^{1/2}\delta}{2a} \)  
\( \Delta \) = \( L/8 \) = Space step  
\( \Delta t \) = Time step  
\( \Gamma = \frac{0.48 k_g Pr_g^{1/3} U_g^{1/2}}{\left( \frac{\mu_g}{\rho_g} \right)^{1/2} \rho_w C_{pw}} \) = Aerodynamic parameter \( (17) \)  
\( H(f) \) = Measured transfer function (i.e., FFT frequency response function) of larger diameter thermocouple with respect to smaller diameter thermocouple  
\( G_{11}(f) \) = Measured FFT autospectral density function of smaller diameter thermocouple
\[ G_{12}(f) = \text{Measured FFT cross-spectral density function between small thermocouple and large diameter thermocouple} \]

\[ \gamma^2_{12}(f) = \text{Measured FFT ordinary coherence function between larger diameter thermocouple and smaller diameter thermocouple} \]

\[ S_2(f) = \text{Measured FFT spectrum of smaller diameter thermocouple} \]

**Overview of Compensation Procedure**

1. The theoretical transfer functions (frequency response function gain) between the 76 \( \mu \text{m} \) (3 mil) thermocouple and the gas stream \((\theta_{in}/a_n)\) and the 254 \( \mu \text{m} \) (10 mil) thermocouple and the gas stream \((\theta_{2n}/a_n)\) are computed from the thermal finite difference solution using Equations 7 through 16, for a range of values of the aerodynamic parameter \((\Gamma)\) at a number of discrete frequencies falling between the corner frequencies of the two thermocouples. These data are then used to compute the theoretical transfer function \((\theta_{2n}/\theta_{in})\) between the 250 \( \mu \text{m} \) (10 mil) thermocouple and the 76 \( \mu \text{m} \) (3 mil) thermocouple for the corresponding values of \(\Gamma\) and frequency. These curves will be used to determine the in situ value of \(\Gamma\) from the measured transfer function of \(\theta_{2n}/\theta_{in}\). The process is described in the following paragraphs.

   a. The following parameters are input or already stored in the computer. For type B thermocouple wire - \(L, f, D, d, \rho_w, k_w, C_{pw}, \) and \(\alpha_w\). For the gas stream - \(\rho_g, k_g, C_{pg}, \gamma_g, \mu_g, \) and \(Pr_g\).

   b. The average or mean conditions for the test data for the following variables are entered into the computer.

   \[ T = \text{Mean gas temperature} \]

   \[ P = \text{Mean gas pressure} \]

   \[ F/A = \text{Fuel air ratio} \]

   \[ f_i-f_x = \text{Frequencies of } f_n \text{ at which transfer functions will be evaluated} \]

   \[ Mn = \text{Mach number} \]

   c. The program computes an estimated value of \(\Gamma\) based on the estimated run conditions using Equation 17.

   d. The program then computes \(\zeta_{pg}\), the transfer function between the wire thermocouple and the gas stream, for the 76 \( \mu \text{m} \) (0.003 in.) and the 254 \( \mu \text{m} \) (0.010 in.) thermocouple via Equations 7 through 16 from 0.2 \(\Gamma\) to 1.8 \(\Gamma\) in steps of 0.1 \(\Gamma\) at frequencies \(f_1, \ldots, f_x\) which are user selected to fall in between estimated values of the corner frequencies of the two thermocouples (Figure 3). The equations are evaluated until steady state conditions are reached. The criteria for steady state is that the positive maximum peak of \(\theta_{2n}/a_n\) be within 0.1 percent of the absolute values of the negative
maximum peak within the same period. The computer code determines the sampling interval for each frequency evaluated to ensure mathematical stability of the finite element model and minimize computation time. The normalized ratio of the magnitude of the temperature fluctuation in the wire to the temperature fluctuation of the gas stream \((C_p - \theta_w/a_w)\) at frequency \(f_h\) is determined by locating the maximum peak amplitude (Figure 4) after the model has iterated to steady-state conditions. The phase shift \(\eta_0\) of the temperature fluctuation in the wire is determined by locating the time at which \(C_p\) crossed zero going positive at the beginning of the period in which the model reached steady-state conditions (Figure 4).

e. The data from (d) are then used to compute the theoretical transfer function \(\theta_{2u}/\theta_{1u}\) from 0.2 \(\Gamma\) to 1.8 \(\Gamma\) (Figure 5) at frequencies of \(f_1\) through \(f_e\).

2. Thermocouple test data are digitized into the Fourier system computer, typically 32 to 120 records each of the 76 \(\mu\)m thermocouple dynamic signal and the 254 \(\mu\)m thermocouple dynamic and dc signals. Each record contains 2048 samples of the data. These data are then converted from millivolts to temperature using National Bureau of Standards (NBS) calibration curve coefficients for type B thermocouples. The 254 \(\mu\)m dc channel is used as the mean for both dynamic data channels in converting the nonlinear thermocouple mv signals to linearized temperature. These data records are then saved for recall for additional processing or plotting.

3. An ensemble averaged FFT transfer function (frequency response) analysis is then performed on \(x\) number of time records of the dynamic data to yield the measured value of \(\theta_{2u}/\theta_{1u}\) (i.e., \(H(f)\)) as a function of frequency. The transfer function is computed as the FFT cross spectral density function between the 76 \(\mu\)m thermocouple and the 250 \(\mu\)m thermocouple divided by the FFT autospectral density function of the 76 \(\mu\)m signal:

\[
H(f) = \frac{G_{12}(f)}{G_{11}(f)}
\]

In conjunction with the computation of the measured transfer function, the coherence function \(\gamma_{12}^2(f)\) is computed and used to assess the quality of the measurement. For the 2048 time sample data record lengths used, 1024 line FFTs are produced. For the typical sampling rate of 4096 Hz (certain other sampling rates are permitted), the FFT analyses yield spectral information from dc to 2048 Hz in 2 Hz intervals. A standard Hewlett Packard windowing function (P301) is used prior to computation of the FFTs. This window is characterized by excellent spectral amplitude accuracy (less than \(\pm\) 0.1 percent). Side lobe suppression is greater than -70 dB at \(\pm\) 4 spectral lines and the effective noise bandwidth is 3.4 spectral lines.

4. Each measured value of \(H(f)\) at frequencies \(f_n = f_1 \rightarrow f_e\) are used in conjunction with the theoretical values of \(\theta_{2u}/\theta_{1u}\) vs \(\Gamma\) at the corresponding frequencies to determine a measured value of \(\Gamma\). (The program interpolates between the 0.1 \(\Gamma\) increments computed in (1) above.) The arithmetic average of \(\Gamma\) obtained for each frequency is taken as the in situ measured value.
5. Using the measured in situ average value of \( \Gamma \) obtained in (4), \( \zeta_{in} \), the normalized transfer function (gain \( \theta_{in}/a_n \) and phase \( \eta_{in} \)) of the 76 \( \mu \)m thermocouple with respect to the gas stream temperature is then computed at all frequencies from the first spectral line of the FFT spectrum to the Nyquist frequency of the FFT for each discrete frequency contained in the FFT (Figure 6). This is typically from 2 Hz to 2048 Hz in 2 Hz increments. This is the compensation spectrum \( \theta_1(f)/a_n(f)/\eta_1(f) \) which is then used to compensate the 76 \( \mu \)m thermocouple data as follows:

6. To compute the compensated ensemble averaged PSD, the measured ensemble averaged autospectral density function of the 76 \( \mu \)m (3 mil) thermocouple obtained in (3) above is divided by the autospectral density function of its compensation spectrum:

\[
\frac{G_{11}(f)}{(\Delta f)} \approx \frac{\theta_1(f) \ast \theta_1(f)}{\eta_1(f)} \rightarrow a_n^2(f)
\]

where: \( \ast = \) Complex conjugate multiplication.

Scaling factors for effective noise bandwidth and FFT symmetry are applied.

7. To compute the compensated instantaneous time waveform, an FFT spectrum \( S_n(f) \) is made on a specific user selected time record yielding amplitude and phase terms for each spectral component. This spectrum expressed in rectangular frequency coordinates is then divided (complex math) by the compensation spectrum. The compensated instantaneous spectrum is then inverse Fourier transformed to yield the compensated instantaneous time waveform. The software contains information on specific techniques employed to prevent time waveform distortions associated with the inverse Fourier transform. A threshold, in relative dB, is applied to the frequency spectrum of the data signal prior to division by the compensation spectrum to prevent errors where the signal to noise ratio is too low:

\[
\frac{S_n(f)/a_n^2(f)}{\theta_1(f)/\eta_1(f)} \approx \frac{\theta_1(f)/\eta_1(f) + \varphi(f)}{a_n^2(f)/\eta_1(f)} \rightarrow a_n(f)/\varphi(f)
\]

which is the compensated instantaneous frequency spectrum and

\[
F^{-1} a_n(f)/\varphi(f)
\]

which is the compensated time waveform where \( F^{-1} \) is the inverse Fourier Transform.
Figure 3. Theoretical Curves of $\zeta$ for 76 $\mu$m and 250 $\mu$m Thermocouples
Figure 4. Transfer Function of Thermocouple With Respect to Gas Temperature
Figure 5. Theoretical Curves of $\xi_q$ for the 250 $\mu$m Thermocouple Divided by $\xi_q$ of the 76 $\mu$m Thermocouple

Figure 6. Typical Compensation Spectrum for 76 $\mu$m (3 mil) Thermocouple Output
DATA PREPROCESSING

Data Acquisition and Playback

Test data are normally recorded on an FM magnetic tape recorder and then reproduced and digitized post-test off-line. Typically, an IBM compatible digital tape is prepared, and data are transferred to a file of the proper format (for input to the FORTRAN program) through the use of user supplied routines. For a system providing on-line digitization and input to a mainframe IBM computer, data would not have to be recorded/reproduced from FM magnetic tape.

The Dynamic Gas Temperature Measurement System does not incorporate a calibration signal analysis routine. The user must provide the means of measuring the rms and ‘dc offset’ values required for determining the necessary inputs of gain, output/input, and dc offset into the main program. Referring to Figure 7, these necessary user inputs are described as follows:

- **GAIN** — The gain of the thermocouple signal amplifier at each specific test point. Used to optimize signal level for recording.

- **OUTPUT/INPUT** — The overall gain of the record/playback system which is used in conjunction with GAIN to scale data to voltage for conversion to scaled temperature. Compute output/input for entry using the ac calibration event as follows:

\[
\frac{\text{OUTPUT}}{\text{INPUT}} = \frac{\text{rms volts (or counts) at conversion to data input file}}{\text{rms volts of input calibration signal}}
\]

(22)

- **DC OFFSET** — Zero offset of record/playback system. Compute as average of measured volts (or counts) at conversion to data input file from the shorted input calibration event.

**NOTE:** GAIN assumes offset in thermocouple signal amplifier is zero volts at all gain settings. If this is not the case, the user must algebraically add the setting to OFFSET determined from the calibration event.

**Figure 7. Elements of Data Acquisition/Playback**
Conversion from Digitized Data to Test Data Input File

Once the data have been digitized, they must be put into a form accessible by the program. The program accesses the data input file as unit 4. This file must be in the exact format described below.

The data are arranged in blocks with a data block consisting of 2048 data points (or the blocksize designated in the user input file) of a specific type of data. The data file is set up as one block of small thermocouple ac (dynamic), one block of large thermocouple ac (dynamic), and one block of dc (steady-state) data. This set-up is repeated for each record of data desired. Each individual line of a data block is in the format \((1X,6E13.0)\). For example, a data block with blocksize 2048 consists of 341 lines with six numbers (starting in column 2) in exponential format, and one line with only two numbers, completing the 2048 data points.
DESCRIPTION OF THE COMPUTER PROGRAM

General Description

The overall system logic is shown in the flow diagram of Figure 8. The names of the subroutines called to perform the various functions are enclosed in the flow diagram boxes. Detailed flow diagrams of the major subroutines are shown in the Appendix. The program operates in three major sections, (a) calculation of an estimated value of the aerodynamic parameter, gamma ($\Gamma_\nu$), and the estimated transfer functions, (b) evaluation of a measured value of gamma ($\Gamma_m$), and calculation of the compensation spectrum, and (c) calculation and plotting of averaged and/or instantaneous time and/or frequency domain data. On the TSS operating system, the program must be run twice, once for the calculations of the compensation spectrum, scaled data, and Fourier transforms, and again for the creation of plots. See "Program Execution on TSS Operating System Using the DISSPLA Graphics Package" section for details.
Figure 8. Program Logic Design
Functions Performed

Referring to Figure 8, the sequence of functional operation is in the following order:

1. Input thermocouple dimensions, gas stream properties, digitized data recording information, and all user options.

2. Generate the P301 windowing function.

3. Calculate thermocouple and gas stream parameters. Note: Thermocouple parameters include the estimated value of gamma ($\Gamma_e$).

4. Check user options to see if calculation of a new compensation spectrum is desired. If not, go to 12.

5. Check user options to see if calculation of a new gamma is desired. If not, go to 11.

6. Calculate the estimated thermocouple vs gas stream transfer functions for $0.2\Gamma_e$ to $1.8\Gamma_e$ at steps of $0.1\Gamma_e$ for both small and large diameter thermocouples.

7. Find the estimated transfer functions of the large diameter thermocouple to the small diameter thermocouple for $0.2\Gamma_e$ to $1.8\Gamma_e$.

8. Form the auto-power and cross-power spectra as follows:
   
   A) Read in 1 block of digitized test data desired in the ensemble averaging.

   B) Scale the data to temperature (Kelvin) and multiply by the P301 windowing function.

   C) Perform an FFT to convert the data to frequency domain.

   D) Self-conjugate and cross-conjugate multiply to form the auto-power and cross-power spectra respectively.

   E) Have all records that were desired been used? If not, return to A and repeat.

   F) Ensemble average the auto-power and cross-power spectra for the desired records.

9. Calculate the measured transfer function and the coherence function as functions of the auto-power and cross-power spectra.

10. Using the measured transfer function, interpolate with estimated transfer functions in order to find a measured gamma ($\Gamma_m$).

11. Calculate compensation spectrum (the transfer function of the desired thermocouple vs gas stream using the measured gamma).
12. Find the averaged frequency domain data as follows:

A) Access the frequency domain data calculated in 8C and stored on units 15 (small TC) and 16 (large TC).

B) Self-conjugate multiply to form the auto-power spectrum.

C) If data are to be uncompensated, skip to step F.

D) Perform self-conjugate multiplication on the compensation spectrum to put in power form.

E) Divide the auto-power spectrum by the power form of the compensation spectrum.

F) Scale the data according to Table 1.

G) Ensemble average desired records.

13. Calculate composite instantaneous time and frequency domain data as follows:

A) Access the scaled time domain temperature data found in step 8B, stored on units 13 and 14.

B) If data are to be uncompensated skip to step J.

C) Obtain a second data block and shift as described below in Description of Subroutines (CSFN).

D) Multiply data by the Hanning window.

E) Perform an FFT to the frequency domain.

F) Perform complex division by the compensation spectrum.

G) Subject data to an inverse Fourier transform back to the time domain.

H) De-Hann data by dividing by the Hanning Window.

I) Combine two data blocks into one as described in Description of Subroutines (CSFN) below.

J) Apply the P301 window to time domain data.

K) Perform an FFT to convert to frequency domain.

L) Self-conjugate multiply and scale according to Table 1.

Detailed flowcharts of the major subroutines are presented in the Appendix.
Description of Subroutines

The Dynamic Gas Temperature Measurement System program was written in such a way that all major functions are performed by distinct and separate subroutines. The function of the MAIN routine and all subroutines is described in this section.

MAIN

The primary function of the MAIN routine is to pass control of the program to the major subroutines. However, if the program is using a previously calculated compensation spectrum, the MAIN routine reads in this data from unit 12. The compensation spectrum will be written to unit 6 (if IBUG2 is turned on), and then converted to rectangular coordinates (it is stored as polar gain and phase) before execution continues.

CHECK

Because of the various user options available, required inputs change according to the IFLAGS entries. Subroutine CHECK checks that the user has input all needed information. If an error has occurred, control is passed to subroutine TERM which terminates execution with an appropriate message. See Input Description in the Input/Output Section for a description of variables.

CSFN

CSFN evaluates instantaneous compensated time and frequency spectra for plotting. This is done by reading in two records of scaled, digitized data from unit 13 or 14 (depending on the thermocouple used). Compensated time domain data are derived by applying the Hanning windowing function,

\[ H(x) = 1 - \cos \left( \frac{2 \pi (x-1)}{IBLSZ} \right) \]  \( \text{for } x = 1, IBLSZ \)  

followed by performing an FFT. The user supplied threshold level (see step 7 on page 11) is then applied before dividing by the compensation spectrum. Data are then subjected to an inverse Fourier transform followed by de-Hanning. Application of the Hanning Window causes some invalidities at the ends of the data blocks, therefore the two data blocks are shifted to provide one block of 'good' data. The shifting is performed as shown in Figure 9.

Frequency domain data are found by applying the P301 window to the time domain data, performing an FFT, scaling the data according to Table 1, and self-conjugate multiplying. If IBUG2 is turned on, these functions are written to the output file, unit 6.

Variables

- ALSS — Area line shape squared, needed for scaling of frequency data (input)
- COMP — Compensation spectrum (input)
- DATA — Time domain data (output to plotting routine)
- DATAF — Frequency domain data (output to plotting routine)
- GS — Array containing gas stream parameters, needed only for the plotting routine (input)
- WINDO — P301 windowing function (input).
Figure 9. Data Block Shifting

FFT

Subroutine FFT calculates finite complex Fourier transform or the inverse transform of a complex input array. Input to the routine is a complex array of size N. (Note: N must be an integer power of 2.) Complex data consist of a real part equal to the time domain data and an imaginary part of zero. The FFT is returned from the routine in the same complex array. The first element of the direct transform is the mean (zero frequency) value. The second through \((N/2 + 1)\) values are for positive frequencies, and the remaining \((N/2 - 2)\) values are for negative frequencies. Results from these negative frequencies are folded about the \((N/2 + 1)\) point. Thus, the frequency for the \((N/2 + 2)\) point is the negative of the frequency for the \(N/2\) point, and so on.

Variables

- **A** — Array of N complex values to which the transform is applied and put back into the array (input and output)
- **ISET** — Type of transform requested (input)
  - 1 = Direct transform
  - -1 = Inverse transform
- **N** — Number of data points or block size — must be an integer power of 2 (input).

GET

Subroutine GET accesses the Fourier transformed input data stored on unit 15 or unit 16, depending on the thermocouple for which the compensation spectrum was found, and self-conjugate multiplies to obtain the auto-power spectrum in \((\text{Peak deg})^2\).
**Variables**

WW — Array to contain the auto-power spectrum (output).

**GSPARM**

This subroutine calculates the gas stream parameters which follow:

- Density (RHO)
- Thermal conductivity (XK)
- Specific heat (CP)
- Specific heat ratio (GA)
- Viscosity (XMU)
- Sonic velocity (SONVL)
- Kinetic viscosity (G)
- Prandtl number (PR)
- Mean gas velocity (U)
- Aerodynamic parameter, \( \Gamma (GMA) \).

When the aerodynamic parameter is found, it is written to unit 6 if IDEBUG is turned on. The equations used for these calculations are listed in Section II.B of Volume II, Final Report (NAS3-23154, FR-17145)

**Variables**

GS — Array of gas stream parameters (output)
TS — Array of thermocouple parameters (input).

**INPUT**

Subroutine INPUT reads all of the user inputs from unit 5. For a description of the variables see Input Description in Input/Output Section.

**INTERP**

Subroutine INTERP uses the measured transfer function (see MEASUR) to interpolate within a family of estimated transfer functions of the two thermocouples (see TRANTC) at 0.2\( \Gamma_* \) to 1.8\( \Gamma_* \) to find a measured gamma, \( \Gamma_m \). The interpolation is performed at all user desired frequencies and the results are averaged to find the measured value of gamma. If the measured coherence function is not within the acceptable range of 0.8 to 1.005 at any frequency, that frequency point is not used. Failure to obtain at least one frequency with an acceptable coherence will result in an appropriate message being written to unit 6 and termination of program execution.

**Variables**

COHR — Coherence function (input)
GS — Array containing the gamma value that is updated in this routine (input and output)
RTRAN — Measured transfer function (input)
TRAN — Estimated transfer functions (input).
INTEST

This subroutine strictly reads in the digitized test data from unit 4.

Variables

- IFLAG — Flag to signal end of data (output)
- DATA3 — One record of small diameter thermocouple data (output)
- DATA10 — One record of large diameter thermocouple data (output)
- DATADC — One record of dc channel data (output).

MEASUR

Subroutine MEASUR calculates the measured transfer function of the two thermocouples, and the coherence function which is used to ensure adequate signal to noise ratio. The transfer function is computed as the FFT cross-spectral density function between the small and large diameter thermocouple divided by the FFT auto-spectral density function of the small diameter thermocouple. The coherence function is the self-conjugate multiplication of XY divided by the product of XX and YY.

Variables

- COHR — Coherence function (output)
- RTRAN — Measured transfer function array (output)
- XX — Small diameter thermocouple auto-power spectrum (input)
- XY — Cross-power spectrum of small to large diameter thermocouple (input)
- YY — Large diameter thermocouple auto-power spectrum (input).

OUT

Subroutine OUT prints the user inputs with a description of their meaning to unit 6. See Input Description in Input/Output Section for variable definitions.

PLT1

Subroutine PLT1 plots one input array on a page. This plot can either be the averaged frequency domain data (frequency vs temperature) or the composite instantaneous time waveform (time vs temperature). Which function is being plotted is determined by the value of ICODE, sent through the argument list. The information necessary for labeling the plot is transferred through the argument list and the two commons, INPUTS and PLOTTR. Examples of the type of plots generated can be found in the Test Cases Section, Test Case 2 Plot 1 and Test Case 3 Plot 2.

Variables

- ARRAY — The array to be plotted (input)
- ICODE — Code for function being plotted (input)
  1 = Power spectral density function
  2 = Composit instantaneous time waveform
- NUM — Number of record being plotted if ICODE=2 (input)
- TMEAN — The mean dc temperature for labeling plots (input).
PLT2

PLT2 generates two plots on the same page. These plots are either the compensation spectrum (frequency vs gain and phase), or the composite instantaneous spectra (time and frequency vs temperature). Which function is being plotted is determined by the value of ICODE, sent through the argument list. The information necessary for labeling the plots is transferred through the argument list and the two commons, INPUTS and PLOTTR. Examples of the plots generated can be found in the Test Cases Section, Test Case 1 Plot 1 and Test Case 2 Plot 2.

Variables

- ARR1 — The first array to be plotted (input)
- ARR2 — The second array to be plotted (input)
- GS — Array containing the value of gamma for which the compensation spectrum was found. Needed for plot labeling (input).
- ICODE — Code for function being plotted (input)
  1 = Compensation spectrum
  2 = Composite instantaneous spectra
- NUM — Number of the record being plotted if ICODE=2 (input)
- TMEAN — The mean dc temperature for labeling plots (input).

POWER

Subroutine POWER calculates auto-power spectrums of large and small diameter thermocouples, and the cross-power spectrum of small vs large diameter thermocouples. This is accomplished by taking digitized data for each record specified in IAVDAT (input variable), scaling to Kelvin, applying the P301 window, and performing an FFT.

The auto-power and cross-power spectrums are the self-conjugate multiplication and the conjugate multiplication of the data respectively. These are averaged together for specified records to arrive at the desired functions. The FFTs are written to units 15 and 16 (depending on the thermocouple) for future executions of the program with the same data.

Variables

- DATA3 — Small diameter digitized data (output)
- DATA10 — Large diameter digitized data (output)
- DATADC — DC channel digitized data (output)
- XX — Array containing the small wire auto-power spectrum (output)
- XY — Array containing the cross-power spectrum of small to large wire thermocouples (output)
- YY — Array containing the large wire auto-power spectrum (output).

PRNTIN

PRNTIN prints out (to unit 6) the user's input data in card image format.

Variables

- IIN — Unit number of the input file to be printed (input)
- IOUT — Unit number of the output file into which to write the data (input).
PSDFN

Subroutine PSDFN calculates the Power Spectral Density function (averaged frequency domain data) for plotting. This is accomplished by accessing FFTs for the desired records from units 15 and 16, and self-conjugate multiplying to form the auto-power spectrum. If the data are to be compensated, they are then divided by the power form of the compensation spectrum. To complete the computations, data are scaled according to Table 1 and plotted and/or printed depending on the user's options. Also performed in PSDFN is the calculation of ALSS, which is the sum of the squares of the P301 window at each channel, divided by the block size. ALSS is used in scaling frequency domain data.

Variables

ALSS — Area line shape squared (output)
COMP — Compensation spectrum (input)
WINDO — P301 windowing function (input).

SCALER

Subroutine SCALER takes digitized data and converts it to temperature (Kelvin). SCALER first removes the amplifier dc offset and scales the data to volts prior to conversion to temperature. Two records are ac data (large and small wire thermocouples) and the third is the dc channel. After removal of dc offset and scaling, the program adds the dc to the ac, converts it to temperature, and then removes the dc, leaving peak temperature. This is done to both ac channels. Scaled data are written onto units 13 and 14 for future access.

Variables

DATA3 — Data for the small wire thermocouple, contains digitized data as input and scaled data as output
DATA10 — Data for the large wire thermocouple, contains digitized data as input and scaled data as output
DATADC — Data for the dc channel, contains digitized data as input and scaled data as output.

SPCY

Subroutine SPCY determines the sampling frequency as a function of the input frequency.

Variables

A — Array which contains the sampling frequency (output)
FRQ — Frequency for which the sampling frequency is being found (input).

TCALC

Subroutine TCALC calculates temperature from digitized data for the routine SCALER. The equations used depend on the thermocouple material code, and involve coefficients stored in the array TCF.

The coefficients are from thermocouple curve equations which were derived from NBS curves, where the independent variable (millivolts) was normalized between -1 and +1. The reference junction is 32°F.
Variables

\(T\) — Contains scaled data as input and the derived temperature as output.

TCPARM

Subroutine TCPARM calculates the following thermocouple wire parameters:

- Density \((RHO)\)
- Thermal conductivity \((XK)\)
- Specific heat \((CP)\)
- Thermal diffusivity \((AL)\).

Equations for type B thermocouples can be found in Section II.B (4.1), Final Report Volume II (NAS3-23154 FR-17145). Equations used for type K thermocouples are as follows:

\[
RHO = 540.95 \text{ lb/ft}^3
\]

\[
CP = (0.0001129 \times T + 0.21454)/2.0 \text{ btu/lb}^\circ\text{F}
\]

\[
XK = (0.01547 \times T + 24.505)/7200.0 \text{ btu/ft-sec}^\circ\text{F}
\]

\[
AL = XK/(RHO \times CP) \text{ ft}^2/\text{sec}
\]

Variables

\(TC\) — Array containing the thermocouple parameters (output).

TERM

Subroutine TERM terminates the program due to an illegal user entry, or because a calculated gamma was not able to be found. Before terminating execution, TERM writes an appropriate message to output unit 6.

Variables

\(I\) — Code to determine which error has occurred (input).

TRFM

Subroutine TRFM performs the actual evaluation of the transfer function of the thermocouple wire and the gas stream using the finite difference method described in detail in Section III.C of Volume I, Final Report (Reference 1).

Variables

\(A\) — Array in which gain and phase are stored (output)

\(FRQ\) — Frequency (input)

\(TP\) — Parameters found by TRFP, needed to evaluate transfer function (input).
TRFP

Subroutine TRFP calculates the following parameters needed for evaluation of the transfer function.

- Delta
- Deltat
- Sigma
- CN
- A
- B
- C
- E
- F
- G

The equations for these parameters are located in Section II.B (4.3), Volume II, Final Report (NAS3-23154 FR-17145), and their description is in Section V.F, Volume I, Final Report (NAS3-23154 FR-17145).

Variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Array containing the sampling frequency (output)</td>
</tr>
<tr>
<td>FRQ</td>
<td>Frequency (input)</td>
</tr>
<tr>
<td>J</td>
<td>Code for which thermocouple is being evaluated (input)</td>
</tr>
<tr>
<td>TC</td>
<td>Array containing thermocouple parameters (input)</td>
</tr>
<tr>
<td>TP</td>
<td>Array containing above parameters (output).</td>
</tr>
</tbody>
</table>

TRANGS

Subroutine TRANGS evaluates the transfer function between the thermocouple and the gas stream. This routine is used twice, once for the estimated transfer functions, and once for the compensation spectrum. When finding the estimated transfer functions, the test gammas of 0.2 to 1.8, and the user requested channels (FREQ of input) are used for evaluation. For the compensation spectrum, the measured value of gamma is used, and a piecewise transform is performed (i.e., the transfer function is found at the first 50 channels, followed by every tenth, and filled in linearly). The compensation spectrum is plotted if requested, and then written to unit 12 for future use. If IBUG2 is turned on, it is also written to output file 6. The compensation spectrum is computed, stored, and plotted as polar gain and phase, but converted to rectangular coordinates before execution continues.

Variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMP</td>
<td>Compensation spectrum (output)</td>
</tr>
<tr>
<td>GS</td>
<td>Array which contains the aerodynamic parameter, gamma (input)</td>
</tr>
<tr>
<td>J</td>
<td>Code for thermocouple for which transfer function is desired (input)</td>
</tr>
<tr>
<td>1</td>
<td>Large diameter</td>
</tr>
<tr>
<td>2</td>
<td>Small diameter</td>
</tr>
<tr>
<td>NGAM</td>
<td>Number of gamma values for which transfer function is desired (input)</td>
</tr>
<tr>
<td>17</td>
<td>estimated transfer functions</td>
</tr>
<tr>
<td>1</td>
<td>compensation spectrum</td>
</tr>
<tr>
<td>TC</td>
<td>Array containing thermocouple parameters (input)</td>
</tr>
<tr>
<td>TRAN</td>
<td>Array into which the transfer function is placed (output).</td>
</tr>
</tbody>
</table>
TRANTC

TRANTC evaluates the estimated transfer functions between large and small wire thermocouples, for all test gammas and all desired frequencies.

\[
\begin{align*}
\text{Gain} &= \text{Gain large}/\text{Gain small} \\
\text{Phase} &= \text{Phase large}-\text{Phase small}
\end{align*}
\]

Variables

NGAM — Number of gamma values for which to evaluate the transfer function, always equal to 17 (input).
TRAN — Array containing transfer functions of large and small thermocouples vs gas stream, and the transfer function found in this routine (input and output).

WINDOW

Subroutine WINDOW applies the P301 windowing function to desired data.

Variables

WINDO — The P301 window (input)
DATA — The data to which the window is applied (input and output).

WINGEN

WINGEN generates the P301 windowing function and places it in the variable WINDO.

\[
P (x) = 0.9994484 + 2(-0.955728 \cos(2 \pi (x-1)/IBLSZ)) \\
+ 0.539289 \cos(4 \pi (x-1)/IBLSZ)) \\
- 0.091581 \cos(6 \pi (x-1)/IBLSZ))
\]

for \( x = 1, \text{IBLSZ} \)

Subroutine Interaction

The interaction between program subroutines is shown in Table 2.

Common Blocks

The Dynamic Gas Temperature Measurement System Program contains only three common blocks. Common /DATAS/ contains two arrays, both of which are coefficients needed for the finite difference method, initialized in BLOCKDATA and never altered. Common /INPUTS/ contains all user input data read in subroutine INPUT and needed throughout the program. The last common, /PLOTTR/, consists of only one array containing any data that the user wishes to have printed on all plots generated. Table 3 describes common block variables (type, dimension, units, and definition); Table 4 is a common block/subroutine cross reference; and Table 5 shows treatment of common block variables within the subroutines in which they are contained.
## TABLE 2. — SUBROUTINE INTERACTION

<table>
<thead>
<tr>
<th>Subroutine</th>
<th>Calls</th>
<th>Referenced by</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHECK</td>
<td>TERM</td>
<td>MAIN</td>
</tr>
<tr>
<td>CSFN</td>
<td>FFT</td>
<td>MAIN</td>
</tr>
<tr>
<td></td>
<td>PLT1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PLT2</td>
<td></td>
</tr>
<tr>
<td>FFT</td>
<td></td>
<td>CSFN</td>
</tr>
<tr>
<td>GET</td>
<td></td>
<td>PSDFN</td>
</tr>
<tr>
<td>GSPARM</td>
<td></td>
<td>MAIN</td>
</tr>
<tr>
<td>INPUT</td>
<td>OUT</td>
<td>MAIN</td>
</tr>
<tr>
<td></td>
<td>PRNTIN</td>
<td></td>
</tr>
<tr>
<td>INTERP</td>
<td>TERM</td>
<td>MAIN</td>
</tr>
<tr>
<td>INTEST</td>
<td></td>
<td>POWER</td>
</tr>
<tr>
<td>MAIN</td>
<td>CHECK</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CSFN</td>
<td></td>
</tr>
<tr>
<td></td>
<td>GSPARM</td>
<td></td>
</tr>
<tr>
<td></td>
<td>INPUT</td>
<td></td>
</tr>
<tr>
<td></td>
<td>INTERP</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MEASUR</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PLT2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>POWER</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PSDFN</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TCPARM</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TRANGS</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TRANTC</td>
<td></td>
</tr>
<tr>
<td></td>
<td>WINGEN</td>
<td></td>
</tr>
<tr>
<td>MEASUR</td>
<td></td>
<td>MAIN</td>
</tr>
<tr>
<td>OUT</td>
<td></td>
<td>INPUT</td>
</tr>
<tr>
<td>PLT1</td>
<td></td>
<td>CSFN</td>
</tr>
<tr>
<td></td>
<td>PSDFN</td>
<td></td>
</tr>
<tr>
<td>PLT2</td>
<td></td>
<td>CSFN</td>
</tr>
<tr>
<td></td>
<td>MAIN</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TRANGS</td>
<td></td>
</tr>
<tr>
<td>POWER</td>
<td>FFT</td>
<td>MAIN</td>
</tr>
<tr>
<td></td>
<td>INTEST</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SCALER</td>
<td></td>
</tr>
<tr>
<td></td>
<td>WINDOW</td>
<td></td>
</tr>
<tr>
<td>PRNTIN</td>
<td></td>
<td>INPUT</td>
</tr>
<tr>
<td>PSDFN</td>
<td>GET</td>
<td>MAIN</td>
</tr>
<tr>
<td></td>
<td>PLT1</td>
<td></td>
</tr>
<tr>
<td>SCALER</td>
<td>TCALC</td>
<td>POWER</td>
</tr>
<tr>
<td>SPCY</td>
<td></td>
<td>TRFM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TRFP</td>
</tr>
<tr>
<td>TCALC</td>
<td></td>
<td>SCALER</td>
</tr>
<tr>
<td>TCPARM</td>
<td></td>
<td>MAIN</td>
</tr>
<tr>
<td>TERM</td>
<td></td>
<td>CHECK</td>
</tr>
<tr>
<td></td>
<td>INTERP</td>
<td></td>
</tr>
<tr>
<td>TRANGS</td>
<td>PLT2</td>
<td>MAIN</td>
</tr>
<tr>
<td></td>
<td>TRFM</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TRFP</td>
<td></td>
</tr>
<tr>
<td>TRANTC</td>
<td></td>
<td>MAIN</td>
</tr>
<tr>
<td>TRFM</td>
<td>SPCY</td>
<td>TRANGS</td>
</tr>
<tr>
<td>TRFP</td>
<td>SPCY</td>
<td>TRANGS</td>
</tr>
<tr>
<td>WINDOW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WINGEN</td>
<td></td>
<td>MAIN</td>
</tr>
</tbody>
</table>
### TABLE 3. — COMMON BLOCK VARIABLES

<table>
<thead>
<tr>
<th>Block</th>
<th>Variables</th>
<th>Type</th>
<th>Dimension</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATAS</td>
<td>C</td>
<td>R</td>
<td>45</td>
<td>—</td>
<td>Array containing the coefficients of the equations for the thermocouple wire parameters.</td>
</tr>
<tr>
<td></td>
<td>TCF</td>
<td>R</td>
<td>(11,9)</td>
<td>—</td>
<td>Array containing the coefficients of the equations used to calculate temperature from digitized data.</td>
</tr>
<tr>
<td>INPUTS</td>
<td>IFLAGS</td>
<td>I</td>
<td>12</td>
<td>—</td>
<td>See Input Description section for definition of IFLAGS.</td>
</tr>
<tr>
<td></td>
<td>TCDATA</td>
<td>R</td>
<td>(4,2)</td>
<td>cm</td>
<td>Thermocouple dimensions</td>
</tr>
<tr>
<td></td>
<td>GAS</td>
<td>R</td>
<td>4</td>
<td>—</td>
<td>Fuel/air ratio</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>K</td>
<td>Mean gas temperature</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>N/m³</td>
<td>Mean gas pressure</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mach number</td>
</tr>
<tr>
<td></td>
<td>FREQ</td>
<td>R</td>
<td>4</td>
<td>Sec</td>
<td>Delta-T setting</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Hz</td>
<td>Starting frequency, ending frequency, and frequency increment for which to find the transfer function.</td>
</tr>
<tr>
<td></td>
<td>CHANL</td>
<td>R</td>
<td>9</td>
<td>Volta</td>
<td>Contains gain, record level, and offset for small, large, and dc thermocouples</td>
</tr>
<tr>
<td></td>
<td>IAVDAT</td>
<td>I</td>
<td>2</td>
<td>—</td>
<td>Starting record number and number of records to use in ensemble averaging</td>
</tr>
<tr>
<td></td>
<td>IBLSZ</td>
<td>I</td>
<td>—</td>
<td>—</td>
<td>Data block size</td>
</tr>
<tr>
<td></td>
<td>IREC</td>
<td>I</td>
<td>10</td>
<td>—</td>
<td>Records desired for plotting of instantaneous spectrum</td>
</tr>
<tr>
<td></td>
<td>TIME</td>
<td>R</td>
<td>2</td>
<td>Sec</td>
<td>Starting and ending time (with respect to data block) for partial time range plots</td>
</tr>
<tr>
<td></td>
<td>IBSZ</td>
<td>I</td>
<td>—</td>
<td>—</td>
<td>Half the block size plus one</td>
</tr>
<tr>
<td></td>
<td>GAMMA</td>
<td>R</td>
<td>—</td>
<td>—</td>
<td>User input value of gamma</td>
</tr>
<tr>
<td></td>
<td>NREC</td>
<td>I</td>
<td>10</td>
<td>—</td>
<td>Records for plotting of frequency domain data when only one record at a time is averaged</td>
</tr>
<tr>
<td></td>
<td>NRECS</td>
<td>I</td>
<td>2</td>
<td>—</td>
<td>Starting record and number records to use in averaging of frequency domain data</td>
</tr>
<tr>
<td></td>
<td>PLTFRQ</td>
<td>R</td>
<td>—</td>
<td>Hz</td>
<td>Frequency at which to end plots of frequency domain data</td>
</tr>
<tr>
<td></td>
<td>TIMTEM</td>
<td>R</td>
<td>—</td>
<td>K</td>
<td>Temperature to which the instantaneous time domain data is to be scaled</td>
</tr>
<tr>
<td></td>
<td>IDEBUG</td>
<td>I</td>
<td>—</td>
<td>—</td>
<td>Intermediate write option 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td>— no writes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>— write out the interpolated gammas</td>
</tr>
<tr>
<td></td>
<td>IBUG2</td>
<td>I</td>
<td>—</td>
<td>—</td>
<td>Intermediate write option 2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td>— no writes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>— print out all of the functions generated</td>
</tr>
<tr>
<td></td>
<td>ITHRSH</td>
<td>I</td>
<td>—</td>
<td>dB</td>
<td>Relative threshold level for instantaneous time domain plots</td>
</tr>
<tr>
<td></td>
<td>PLOTTR</td>
<td>A</td>
<td>(20,3)</td>
<td>—</td>
<td>User supplied headings for plots</td>
</tr>
</tbody>
</table>

7222C
<table>
<thead>
<tr>
<th>Common Block</th>
<th>Subroutines Containing the Common Block</th>
</tr>
</thead>
<tbody>
<tr>
<td>INPUTS</td>
<td>CHECK, CSFN, GET, GSPARM, INPUT, INTERP, INTTEST, MAIN, MEASUR, OUT, PLT1, PLT2, POWER, PSDFN, SCALER, TCALC, TCPARM, TRANGS, TRANTC, TRFP, WINDOW, WINGEN</td>
</tr>
<tr>
<td>DATAS</td>
<td>BLOKDATA, GSPARM, MAIN, POWER, SCALER, TCALC, TCPARM</td>
</tr>
<tr>
<td>PLOTTTR</td>
<td>INPUT, PLT1, PLT2</td>
</tr>
</tbody>
</table>
TABLE 5. — TREATMENT OF COMMON BLOCK VARIABLES WITHIN SUBROUTINES

<table>
<thead>
<tr>
<th>Common Block Variable</th>
<th>Subroutines and Usage Codes</th>
</tr>
</thead>
<tbody>
<tr>
<td>INPUTS</td>
<td>CHECK CSPN GET GSPARM INPUT INTERP INTENT MAIN MEASUR OUT PLT1 PLT2 POWER PPDFN SCALER TCLC TCPPARM TRFP TRANGS TRANVC WINDO WINGEN</td>
</tr>
<tr>
<td>IFLAGS</td>
<td>R R R R A U U U U R U R R U R U R U U U</td>
</tr>
<tr>
<td>TCADATA</td>
<td>U U U U A U U U A U R R R R U U U U R</td>
</tr>
<tr>
<td>GAS</td>
<td>U U U R A U U U U U R R R R U U U U R</td>
</tr>
<tr>
<td>FREQ</td>
<td>U R U U A R U U U R R R R U U U U R</td>
</tr>
<tr>
<td>CHANL</td>
<td>U U U U A U U U U R R R R U U U U R</td>
</tr>
<tr>
<td>IAVDAT</td>
<td>U U U U A U U U U R R R R U U U U R</td>
</tr>
<tr>
<td>IBLES</td>
<td>U R U U A U U U U R R R R U U U U R</td>
</tr>
<tr>
<td>IREC</td>
<td>R R U U A U U U U R R R R U U U U R</td>
</tr>
<tr>
<td>TIME</td>
<td>R U U U A U U U U R R R R U U U U R</td>
</tr>
<tr>
<td>IBZ</td>
<td>U R R U A U U U R R R R U U U U R</td>
</tr>
<tr>
<td>GAMMA</td>
<td>R U U U A U U U U R R R R U U U U R</td>
</tr>
<tr>
<td>NREC</td>
<td>R U U U A U U U U R R R R U U U U R</td>
</tr>
<tr>
<td>NRCS</td>
<td>R U U U A U U U U R R R R U U U U R</td>
</tr>
<tr>
<td>PLTFRQ</td>
<td>U U U U A U U U U R R R R U U U U R</td>
</tr>
<tr>
<td>TIMTRM</td>
<td>R U U U A U U U U R R R R U U U U R</td>
</tr>
<tr>
<td>IDSGUG</td>
<td>U U U U A U U U U R R R R U U U U R</td>
</tr>
<tr>
<td>IBUGS</td>
<td>U R U U A U U U R U R R U U U U R</td>
</tr>
<tr>
<td>ZTHRESH</td>
<td>U R U U A U U U U R U U U U R U</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DATAS</th>
<th>BLOCKDATA GSPARM MAIN POWER SCALER TCLC TCPPARM</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>A R U U U U U U R R</td>
</tr>
<tr>
<td>TCF</td>
<td>A U U U U U R</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FLOTTR</th>
<th>INPUT PLT1 PLT2</th>
</tr>
</thead>
<tbody>
<tr>
<td>FLOTTT</td>
<td>A R R</td>
</tr>
</tbody>
</table>

Usage Codes:
- A = Altered
- R = Referenced
- U = Unreferenced
The program requires two different types of files as input. The first of these files is test data in raw voltage. A description of how this file is created and its correct format is included in the Data Preprocessing section. The second input file is a user input file including information such as thermocouple wire dimensions, gas stream properties, test data recording information, and many user options.

Program output consists of five 'intermediate' output files, one 'final' output file, and graphical output. This output provides both graphic and tabulated data for the compensation spectrum, averaged frequency spectrum, and instantaneous time waveform and frequency spectrum. Table 1 lists specific frequency and time domain functions available for output. Plotting routines contain both CALCOMP and DISSPLA calls.

Input Description

The input description in Table 6 illustrates the required form for card input of the specifying parameters in the user input file. Lines 1 through 13 of the input file must always be present, while depending on the various user options, some of lines 14 through 23 may be absent as will be seen in the "Test Cases" section.
### TABLE 6. — USER INPUT DESCRIPTION

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Dimensions</th>
<th>Type</th>
<th>Input Format</th>
<th>Definition and Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Line 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TCDATA</td>
<td>(4,2)</td>
<td>R</td>
<td>8F10.0 cm</td>
<td>Thermocouple dimensions as below</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(1,1) Length of support wire for large diameter thermocouple</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(2,1) ¾ total length of the smaller wire for large thermocouple</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(3,1) Diameter of support wire for large diameter thermocouple</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(4,1) Diameter of smaller wire for large diameter thermocouple</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(1,2), (2,2), (3,2), and (4,2) Same as above for small diameter thermocouple</td>
</tr>
<tr>
<td>Line 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GAS(1)</td>
<td></td>
<td>R</td>
<td>3F10.0</td>
<td>Fuel/air ratio</td>
</tr>
<tr>
<td>GAS(2)</td>
<td></td>
<td></td>
<td></td>
<td>Mean gas temperature</td>
</tr>
<tr>
<td>GAS(4)</td>
<td></td>
<td></td>
<td></td>
<td>Mach number</td>
</tr>
<tr>
<td>Line 3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GAS(3)</td>
<td></td>
<td>R</td>
<td>E20.0 N/m²</td>
<td>Mean gas pressure</td>
</tr>
<tr>
<td>Line 4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FREQ(1)</td>
<td></td>
<td>R</td>
<td>E10.0 Sec</td>
<td>Delta-T setting</td>
</tr>
<tr>
<td>Line 5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FREQ(2)</td>
<td></td>
<td>R</td>
<td>3F10.0 Hz</td>
<td>Starting frequency, ending frequency, and frequency increment for which to find the transfer function</td>
</tr>
<tr>
<td>FREQ(3)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FREQ(4)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lines 6 and 7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CHANL</td>
<td>9</td>
<td>R</td>
<td>8F10.0 Volts</td>
<td>Contains the gain, output/input, and dc offset values for small, large, and dc thermocouples (i.e., CHANL(1) — CHANL(3) contain information for the small diameter thermocouple)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>F10.0</td>
<td></td>
</tr>
<tr>
<td>Line 8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IAVDAT</td>
<td>2</td>
<td>I</td>
<td>215</td>
<td>Starting record number and number of records to use in ensemble averaging</td>
</tr>
<tr>
<td>Line 9</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IBLSZ</td>
<td>—</td>
<td>I</td>
<td>15</td>
<td>Data block size*</td>
</tr>
<tr>
<td>Line 10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PLTFRQ</td>
<td>—</td>
<td>R</td>
<td>F10.0 Hz</td>
<td>Frequency at which to end plots of frequency domain data</td>
</tr>
<tr>
<td>Line 11</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IFLAGS</td>
<td>12</td>
<td>I</td>
<td>1215</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1) Gamma used for Comp spectrum</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1 — Calculate measured gamma</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2 — Use a user entered gamma</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3 — Use existing compensation spectrum</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2) Thermocouple material code</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0 — N/A : IFLAGS(1)=3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1 — PT / 6%RH</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2 — PT / 30%RH</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3 — CR / AL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3) Thermocouple to use for comp spectrum</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0 — N/A : IFLAGS(1)=3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1 — Small wire thermocouple</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2 — Large wire thermocouple</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4) Plot of compensation spectrum?</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1 — Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2 — No</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5) Plot of instantaneous data?</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1 — Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2 — No</td>
</tr>
</tbody>
</table>
### Table 6. — User Input Description (Continued)

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Dimensions</th>
<th>Type</th>
<th>Input Format</th>
<th>Units</th>
<th>Definition and Options</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6) Scaling technique to use</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1 — Regular power spectral density (PSD) K^2/Hz</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2 — Log PSD (10log(PSD)) dB — 0 dB ref 1 K^2/Hz</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3 — Linear PSD (√PSD) rms K/√Hz</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4 — Narrowband frequency rms K</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>7) Plot of averaged frequency domain data?</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1 — Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2 — No</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>8) Type averaging desired**</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0 — N/A: no plots or printouts</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1 — Use user specified number of records</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2 — Average only one at a time (user specifies the one)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>9) Plotting option 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0 — N/A: no plots or printouts</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1 — Plot compensated data</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2 — Plot uncompensated data</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10) Plotting option 2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0 — N/A: no plots or printouts</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1 — Plot time and frequency data</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2 — Plot time data only</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>11) Plotting option 3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0 — N/A: no plots or printouts</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1 — Plot full time range</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2 — Plot partial time range</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12) Temperature scaling flag</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0 — N/A: no instantaneous plots</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1 — Scale each record to its own maximum or minimum temperature</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2 — Scale all records to a user specified temperature</td>
</tr>
</tbody>
</table>

**Line 12**
IDEBUG — I 15 — Intermediate write option 1
0 — No writes
1 — Write out the interpolated gammas

**Line 13**
IBUG2 — I 15 — Intermediate write option 2***
0 — No writes
1 — Print out all of the functions generated

**Line 14** — Input if IFLAGS(1) = 2
GAMMA — R E20.0 SI User input value of gamma (m^1/2/sec)

**Line 15** — Input if IFLAGS(5) = 1 or IBUG2 = 1
IREC 10 I 1015 — Records desired for plotting or printout of the instantaneous spectrum

**Line 16** — Input if IFLAGS(11) = 2
TIME 2 R 2F10.0 Sec Starting and ending time (with respect to data block) for partial time range plots

**Line 17** — Input if IFLAGS(8) = 2
NREC 10 I 1015 — Records of frequency domain data to use when only one record at a time is averaged**
TABLE 6. — USER INPUT DESCRIPTION (Continued)

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Dimensions</th>
<th>Type</th>
<th>Format</th>
<th>Units</th>
<th>Definition and Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>NRECS</td>
<td>2</td>
<td>I</td>
<td>215</td>
<td></td>
<td>Starting record and number of records to use in averaging of frequency domain data</td>
</tr>
<tr>
<td>TIMTEM</td>
<td>R</td>
<td></td>
<td>F10.0</td>
<td>K</td>
<td>Temperature to which the instantaneous time domain data are to be scaled</td>
</tr>
<tr>
<td>ITHRSH</td>
<td>I</td>
<td></td>
<td>15</td>
<td>dB</td>
<td>Relative threshold level for instantaneous time domain plots</td>
</tr>
<tr>
<td>PLOTIT</td>
<td>(20,3)</td>
<td>A</td>
<td>3*20A4</td>
<td></td>
<td>User supplied headings for plots</td>
</tr>
</tbody>
</table>

*The block size must be an integer power of 2. Arrays are dimensioned such that the data block size must be less than or equal to 2048. This may be bumped up if necessary.

**Averaged' frequency domain data are available either averaged or instantaneous. If averaged, input desired information to NRECS. If instantaneous, put record numbers desired into NREC (maximum of 10).

***Note: A peculiarity of the program — If IBUG2 is turned on, at least one record must be requested for instantaneous data. A plot need not be requested (IFLAGS(5) can be set to 2), but a printout will be generated. Therefore, IFLAGS(9), IFLAGS(10), IFLAGS(11), and IFLAGS(12) must all be non-zero.

Output Description

Program output involves graphic and tabular form of any functions calculated, and files 'intermediate' output files saving time for any information desired after initial run of the program.

Upon initial execution of the program (IFLAGS(1) = 1), five unformatted files are created. These files contain the compensation spectrum (unit 12), scaled data for small and large thermocouples (units 13 and 14 respectively), and Fourier transformed data for small and large thermocouples (units 15 and 16). Once these five files exist, plots may be generated at minimal cost by executing the program with IFLAGS(1)=3.

The main output file (unit 6) will always contain a card copy of the user input file along with a 'summary' of the input. This 'summary' aids in locating incorrect inputs. Depending on the options in effect, the file might also contain step by step details of the interpolation process for calculating a measured gamma (IDEBUG=1), and tabular forms of all functions calculated (IBUG2=1).

Plots of the compensation spectrum, averaged frequency domain data, and instantaneous time and frequency domain data may be generated. These functions may be either compensated or uncompensated and the plot will denote this decision. Frequency domain graphs may be plotted to any desired frequency, and may be either averaged or instantaneous (both may be generated at the same time). Instantaneous plots may include either time and frequency on the same plot, or just the time domain. Other input options exist allowing several different presentations for the plots. See the Test Cases Section for examples of the types of plots that may be generated.
PROGRAM PECULIARITIES AND RESTRICTIONS

Input Restrictions

The Dynamic Gas Temperature Measurement System only models one element or leg of a thermocouple. Therefore, for TCDATA, the user should input the average of the measured dimensions of the two elements of each thermocouple.

The average of the properties of the two elements or legs comprising the thermocouple is used when finding the thermocouple parameters. (Example: the average of properties for PT/6% RH element and PT/30% RH element for type B thermocouples.) Thus, specification of 1 or 2 for the thermocouple material code (IFLAGS(2)) will produce identical results.

If IBUG2 is set to 1, at least one record of instantaneous data must be requested.

Program Peculiarities

If a partial time block is plotted for the Instantaneous Time Waveform, the rms value (printed on the plot) is only calculated over the displayed portion of the data block.

The absolute time of the Instantaneous Time Waveform is different for compensated and uncompensated data. For compensated data, the plot is shifted up in time by one quarter of a record. For example if Record 1 is plotted uncompensated, absolute time is 0 to 0.5 second (for delta T = 0.24415E-3 seconds or delta F = 2 Hz) while for compensated data the absolute time is 0.125 to 0.625 second. See CSFN in the Subroutines section for details of the shifting technique.

There are three different starting points for the Dynamic Gas Temperature Measurement System (IFLAGS(1) = 1, 2, or 3). The initial run for a test point must have IFLAGS(1) = 1. This setting allows the program to find a measured gamma, calculate the compensation spectrum, and create the five 'intermediate' files discussed in Output Description in the previous section. Once these files exist, the user may specify the gamma value for which to calculate the compensation spectrum by setting IFLAGS(1) = 2. For additional plots without any time consuming calculations, the program may be run with IFLAGS(1) = 3.

Restriction on Test Data

Measured data must contain dynamic temperature fluctuations at one or more frequencies lying generally between the corner frequencies of the large and small diameter thermocouples to ensure adequate sensitivity in the in situ measurement of gamma. The analysis technique is based on the use of thermocouples with differing frequency responses. At frequencies much greater than the thermocouple corner frequencies, the frequency response gain functions approach constant slopes (approximately 6 dB/octave). This results in approximately constant valued transfer functions (i.e., ratio of their frequency response functions) as a function of gamma.
Table 7 provides a list of various sizes of thermocouple elements with recommended usable frequency range (compensated) along with sampling frequency, anti-aliasing filter setting, and the spectral line separation of the frequency analysis (i.e., delta F). This table is intended for use as a guideline only, and is based on type B thermocouples operating at the following conditions:

Mean temperature — 1400K
Pressure — 1.02E + 6 N/m²
Mach number — 0.2
Fuel/air ratio — 0.02

TABLE 7. — SAMPLE USABLE COMPENSATED FREQUENCY RANGE AND DIGITIZING PARAMETERS FOR VARIOUS DIAMETER TYPE B THERMOCOUPLES

<table>
<thead>
<tr>
<th>Thermocouple Diameter (µm)</th>
<th>Fusable (Compensated) (Hz)</th>
<th>At</th>
<th>Anti-Aliasing Filter Setting (−3 dB) (Hz)</th>
<th>ΔF (2048 Block Size) (Hz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>25.4</td>
<td>6000</td>
<td>40E-6</td>
<td>7500</td>
<td>12.207</td>
</tr>
<tr>
<td>50.8</td>
<td>2000</td>
<td>125E-6</td>
<td>2500</td>
<td>3.906</td>
</tr>
<tr>
<td>76.2</td>
<td>1000</td>
<td>250E-6</td>
<td>1200</td>
<td>1.953</td>
</tr>
<tr>
<td>101.6</td>
<td>800</td>
<td>300E-6</td>
<td>1000</td>
<td>1.628</td>
</tr>
<tr>
<td>127.0</td>
<td>600</td>
<td>400E-6</td>
<td>750</td>
<td>1.221</td>
</tr>
<tr>
<td>152.4</td>
<td>450</td>
<td>500E-6</td>
<td>563</td>
<td>0.977</td>
</tr>
<tr>
<td>177.8</td>
<td>350</td>
<td>625E-6</td>
<td>438</td>
<td>0.781</td>
</tr>
<tr>
<td>203.2</td>
<td>300</td>
<td>800E-6</td>
<td>375</td>
<td>0.610</td>
</tr>
<tr>
<td>228.6</td>
<td>250</td>
<td>1E-3</td>
<td>312</td>
<td>0.488</td>
</tr>
<tr>
<td>254.0</td>
<td>200</td>
<td>1.25E-3</td>
<td>250</td>
<td>0.391</td>
</tr>
<tr>
<td>304.8</td>
<td>150</td>
<td>1.5E-3</td>
<td>188</td>
<td>0.326</td>
</tr>
<tr>
<td>355.6</td>
<td>125</td>
<td>2.0E-3</td>
<td>156</td>
<td>0.244</td>
</tr>
<tr>
<td>381.0</td>
<td>100</td>
<td>2.5E-3</td>
<td>125</td>
<td>0.195</td>
</tr>
<tr>
<td>406.4</td>
<td>100</td>
<td>2.5E-3</td>
<td>125</td>
<td>0.195</td>
</tr>
<tr>
<td>457.2</td>
<td>90</td>
<td>2.75E-3</td>
<td>113</td>
<td>0.178</td>
</tr>
<tr>
<td>508.0</td>
<td>80</td>
<td>3.0E-3</td>
<td>100</td>
<td>0.163</td>
</tr>
<tr>
<td>639.0</td>
<td>50</td>
<td>4.5E-3</td>
<td>63</td>
<td>0.109</td>
</tr>
<tr>
<td>762.0</td>
<td>40</td>
<td>6.0E-3</td>
<td>50</td>
<td>0.081</td>
</tr>
</tbody>
</table>

Table 7 was derived using a first order approximation to compute the corner frequency of the thermocouple for specific test conditions, then determining the frequency where the frequency response gain ratio is −35 dB. The usable frequency (Fusable) was then arbitrarily selected to be a convenient value less than the −35 dB frequency point. The sampling frequency was selected to be a convenient value at least four times greater and less than five times the usable frequency (Δt = 1/sampling frequency). The anti-aliasing filter was set to be 1.25 times Fusable (based on a minimum of 48 dB/octave roll-off rate for the anti-aliasing filter). The formulas for computing the first order corner frequency are:

\[
\tau = \frac{1.4p_w C_w D_w^{1.5}}{\sqrt{MP_s}} \left( \frac{T_w}{556} \right)^{0.18}
\]

Where:  
ρ_w = Density of thermocouple wire (grams/m³)  
C_w = Specific heat of thermocouple wire (Joules/kg K)  
D_w = Diameter of thermocouple element (m)  
T_w = Mean thermocouple wire temperature (K)  
M = Mach number  
P_s = Gas stream pressure (N/m²)
Then: \( F_c = \frac{1}{2\pi \tau} \)  
(25)

Where: \( F_c \) = Corner frequency (i.e., -3 dB gain in Hz).

For a first order system, the gain ratio (i.e., output/input) as a function of frequency is given by the expression:

\[
G(f) = \frac{1}{\sqrt{1 + (f/F_c)^2}}
\]

Where: \( f \) = Frequency (Hz)  
\( F_c \) = Corner frequency (Hz).

This expression can be manipulated to normalize \( f \) as a function of \( F_c \) to define \( f \) at any desired gain ratio (expressed in dB):

\[
f = F_c \sqrt{10^{(-\text{dB}/10)}} - 1
\]

(27)

Note: Observe sign of dB value. It will always be negative for thermocouples.

Thus, the frequency at which the thermocouple gain ratio is -35 dB (a practical limit of compensation) would be computed as:

\[
f(-35 \text{ dB}) = F_c \sqrt{10^{(-35/10)}} - 1 = F_c (56.225)
\]

(28)

The recommended \( F_{\text{usable}} \) is an arbitrarily convenient frequency less than \( f(-35 \text{ dB}) \).

Example: 76 \( \mu \)m diameter type B thermocouple operation at the following conditions:

\( T_o = 1400K \)  
\( M = 0.2 \)  
\( P_s = 1.02E + 6 \text{ N/m}^2 \)

Type B thermocouple properties at these conditions are:

\( C_w = 1.933E + 2 \text{ Joules/kg K} \)  
\( \rho_w = 2.009E + 4 \text{ kg/m}^3 \)

1) Compute \( \tau \):

\[
\tau = \frac{(1.4)(2.009E + 4)(1.933E + 2)(76E - 6)^{1.5}}{\sqrt{(0.2)(1.02E + 6)}} \left( \frac{1400}{556} \right)^{0.18}
\]

(29)

\( \tau = 6.754E-3 \text{ seconds} \)

2) Compute \( F_c \):

\[
F_c = \frac{1}{2\pi \tau} = \frac{1}{2\pi(6.754E-3)} = 23.6 \text{ Hz}
\]

(30)
3) Compute \( f(-35 \text{ dB}) \) and select \( F_{\text{usable}} \):

\[
f(-35 \text{ dB}) = (23.6)(56.225) = 1327 \text{ Hz}
\]

Select \( F_{\text{usable}} = 1000 \text{ Hz} \)

4) Compute sampling frequency:

\[
F_s = 4 \times 1000 = 4000 \text{ Hz}
\]

or

\[
\Delta t = 1/F_s = 1/4000 = 0.25 \text{ E \(-6\) seconds}
\]

5) Compute anti-aliasing filter setting:

\[
F_a = (1.25)(F_{\text{usable}}) = 1250 \text{ Hz}
\]

6) Compute FFT spectral line separation (\( \Delta F \))

\[
\Delta F = F_s/\text{block size} = 4000 \text{ Hz}/2048 = 1.953125 \text{ Hz}
\]

The selection of dual thermocouple combinations for the measurement of the aerodynamic parameter, gamma, has not been evaluated in depth other than the 76 \( \mu \text{m}/254 \mu \text{m} \) combination developed for specific application to meet the original Contract Statement of Work for 1 kHz compensated bandwidth in an F100 engine environment. It is imperative that the combination be selected judiciously to provide adequate sensitivity in the ratio of their amplitude response as a function of gamma.

**Accuracy of Results**

A rigorous analysis of all components contributing to the measurement error was beyond the scope of this contract. Table 8 depicts dynamic component errors (expressed as percent of reading) identified and evaluated in the original contract and reported in NASA CR-168267. Notable potential sources of errors that could not be (or were not) evaluated were: (1) dynamic velocity effects, (2) use of the average properties from both thermocouple legs for the single leg finite difference model, and (3) frequency dependency of the aerodynamic parameter gamma.

Results of comparisons made between the dual thermocouple and compensated resistance thermometer data on the rotating wheel and subscale combustor rig experiments (described in Volume I) showed typical dynamic component errors of about \(-22\%\) percent to \(+33\%\) percent (time domain 1 kHz bandwidth). The true errors would lie between these and those presented in Table 8. Note that these errors are based on percents of reading of very small dynamic temperature fluctuations. Inclusion of the mean gas path temperature into the error statements would reduce the errors to less than 4 percent of reading.
### TABLE 8. — OVERALL ERROR

<table>
<thead>
<tr>
<th></th>
<th>Error Due to Data</th>
<th>Error Due to FFT System SNR</th>
<th>Compensation Technique (%)</th>
<th>* (Error (diameters − %))</th>
<th>** (Error (H(f) − %))</th>
<th>Total Error (rss − %)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Averaged Frequency Spectrum</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1000K (1800°F) p-p at 200 Hz</td>
<td>&lt;0.1</td>
<td>1.1</td>
<td>4.9</td>
<td>3.3</td>
<td>6.0</td>
<td></td>
</tr>
<tr>
<td>1000K (1800°F) p-p at 1000 Hz</td>
<td>&lt;0.1</td>
<td>2.0</td>
<td>5.2</td>
<td>3.5</td>
<td>6.6</td>
<td></td>
</tr>
<tr>
<td>≈ 15K (27°F) p-p/√Hz 4 Hz to 200 Hz</td>
<td>0.7</td>
<td>1.2</td>
<td>4.9</td>
<td>3.3</td>
<td>6.1</td>
<td></td>
</tr>
<tr>
<td>≈ 15K (27°F) p-p/√Hz 200 Hz to 1000 Hz</td>
<td>3.8</td>
<td>1.9</td>
<td>5.2</td>
<td>3.5</td>
<td>7.6</td>
<td></td>
</tr>
<tr>
<td>≈ 6.7K (12°F) p-p/spectral line at 200 Hz</td>
<td>2.2</td>
<td>1.2</td>
<td>4.9</td>
<td>3.3</td>
<td>6.4</td>
<td></td>
</tr>
<tr>
<td>≈ 6.7K (12°F) p-p/spectral line at 1000 Hz</td>
<td>10.0</td>
<td>1.9</td>
<td>5.2</td>
<td>3.5</td>
<td>12.0</td>
<td></td>
</tr>
<tr>
<td><strong>Instantaneous Time Waveform</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1000K (1800°F) p-p at 200 Hz</td>
<td>&lt;0.1</td>
<td>1.6</td>
<td>4.9</td>
<td>3.3</td>
<td>6.1</td>
<td></td>
</tr>
<tr>
<td>1000K (1800°F) p-p at 1000 Hz</td>
<td>&lt;0.1</td>
<td>2.5</td>
<td>5.2</td>
<td>3.5</td>
<td>6.7</td>
<td></td>
</tr>
<tr>
<td>≈ 15K (27°F) p-p/√Hz 4 Hz to 200 Hz</td>
<td>0.7</td>
<td>4.4</td>
<td>4.9</td>
<td>3.3</td>
<td>7.4</td>
<td></td>
</tr>
<tr>
<td>≈ 15K (27°F) p-p/√Hz 200 Hz to 1000 Hz</td>
<td>3.8</td>
<td>10.6</td>
<td>5.2</td>
<td>3.5</td>
<td>12.9</td>
<td></td>
</tr>
<tr>
<td>≈ 6.7K (12°F) p-p/spectral line 4 Hz to 200 Hz</td>
<td>2.2</td>
<td>4.4</td>
<td>4.9</td>
<td>3.3</td>
<td>7.7</td>
<td></td>
</tr>
<tr>
<td>≈ 6.7K (12°F) p-p/spectral line 200 Hz to 1000 Hz</td>
<td>10.0</td>
<td>10.6</td>
<td>5.2</td>
<td>3.5</td>
<td>15.9</td>
<td></td>
</tr>
</tbody>
</table>

rms residue

rms input × 100%

* Error due to perturbation of measurement error in thermocouple diameters
** Error due to perturbation of measurement error in H(f)
PROGRAM LISTING

C
C******************************************************************************E249
C
C THIS IS THE MAIN PROGRAM FOR THE DYNAMIC GAS
C TEMPERATURE MEASUREMENT SYSTEM
C
C******************************************************************************E249
C
FOR EARLY DOMESTIC DISSEMINATION
C
BECAUSE OF ITS SIGNIFICANT EARLY COMMERCIAL POTENTIAL, THIS
INFORMATION, WHICH HAS BEEN DEVELOPED UNDER A U.S. GOVERNMENT
PROGRAM, IS BEING DISSEMINATED WITHIN THE UNITED STATES IN
ADVANCE OF GENERAL PUBLICATION. THIS INFORMATION MAY BE Duplic-
ATED AND USED BY THE RECIPIENT WITH THE EXPRESS LIMITATION
THAT IT NOT BE PUBLISHED. RELEASE OF THIS INFORMATION TO OTHER
DOMESTIC PARTIES BY THE RECIPIENT SHALL BE MADE SUBJECT TO THESE
LIMITATIONS.
C
FOREIGN RELEASE MAY BE MADE ONLY WITH PRIOR NASA APPROVAL
AND APPROPRIATE EXPORT LICENSES. THIS LEGEND SHALL BE MARKED ON
ANY REPRODUCTION OF THIS INFORMATION IN WHOLE OR IN PART.
C
******************************************************************************E249
C
CALLS - INPUT: INPUTS ALL THERMOCOUPLE AND GAS STREAM
INFORMATION, ALONG WITH THE USER OPTIONS.
C CHECK: CHECKS FOR ERRORS IN THE USER INPUT AND
TERMINATES EXECUTION IF ANY ARE FOUND.
C WINGEN: GENERATES THE P301 WINDOWING FUNCTION.
C TCPARM: CALCULATES THE THERMOCOUPLE PARAMETERS.
C GSPARM: CALCULATES THE GAS STREAM PARAMETERS.
C PLT2: PLOTS THE COMPENSATION SPECTRUM.
C TRANGS: GENERATES THE TRANSFER FUNCTION OF THE
THERMOCOUPLE TO THE GAS STREAM.
C TRANTC: GENERATES THE TRANSFER FUNCTION BETWEEN THE TWO THERMOCOUPLES.
C POWER: CALCULATES THE AUTO AND CROSS POWER SPECTRUMS.
C MEASUR: CALCULATES THE MEASURED TRANSFER FUNCTION AND THE COHERENCE FUNCTION.
C INTERP: INTERPOLATES BETWEEN THE ESTIMATED TRANSFER FUNCTIONS FOR A MEASURED VALUE OF GAMMA.
C PSDFN: GENERATES AND.Plots THE AVERAGED FREQUENCY DOMAIN DATA (POWER SPECTRAL DENSITY FUNCTIONS).
C CSFN: GENERATES AND PLOTS THE INSTANTANEOUS TIME AND FREQUENCY DOMAIN DATA.
C
FILES USED:
C 6 - WRITES THE COMPENSATION SPECTRUM TO THIS FILE
C IF THE USER SO DESIRES (IBUG2 - 1)
C 12 - READS THE COMPENSATION SPECTRUM FROM THIS FILE E249
C IF USER STARTS AT THIS POINT (IFLAGS(1) = 3) E249
C
C******************************************************************************E249
C
DIMENSION TC(4),GS(10),TRAN(17,3,1024,2),XX(1025),YY(1025),
* XY(1025,2),WINDO(2048),RTRAN(1025,2),COHR(1025),
* COMP(1024,2),IBUFF(1000),REA(1024),RIMA(1024),
* DATA3(2048),DATA10(2048),DATADC(2048)
COMMON /INPUTS/ IFLAGS(12),TCFAD(4,2),GAS(4),FREQ(4),CHANL(9),
* IAVDAT(2),ILSZ,IREC(10),TIME(2),IBSZ,GAMMA,NREC(10),NRECS(2),
* PLTFRQ,TIMTEM,DEBUG,IBUG2,ITHRSH
COMMON /DATAS/ C(45),TCF(11,9)
C******************************************************************************E249
C
C*** INPUT ALL USER EDITS NEEDED FOR RUNNING THE PROGRAM.
CALL INPUT E249
C
C*** CHECK USER INPUT FOR ERRORS
CALL CHECK E249
C
C*** GENERATE THE P301 WINDOWING FUNCTION
CALL WINGEN(WINDO)
C
C*** EVALUATE THE I/C AND GAS STREAM PARAMETERS NEEDED TO FIND
C*** THE TRANSFER FUNCTION
IF(IFLAGS(1).EQ.3) GO TO 5
CALL TCPARM(TC)
CALL GSPARM(TC,GS)
GO TO 50
C
C*** IF USER WISHES OLD COMPENSATION SPECTRUM TO BE USED, READ IN
C*** AND PLOT THE COMPENSATION SPECTRUM AND ALL RELATIVE DATA
5 CALL PLOISP(IBUFF,1000,0,4)
REWIND 12
ISZ = IBSZ - 1
READ(12) (COMP(J,1),J=1,ISZ), (COMP(J,2),J=1,ISZ),
* IFLACS(2),IFLACS(3),GMAMET,((TCDATA(I,J),I=1,4),J=1,2)
GS(10) = GMAMET * 5.9425
G = GS(10)
CALL TCPARM(TC)
CALL GSPARM(TC,GS)
GS(10) = G
DO 10 I = 1,ISZ
REA(I) = COMP(I,1)
RIMA(I) = COMP(I,2)
10 CONTINUE
IF(IFLAGS(4).EQ.1) CALL PLT2(REA,RIMA,1,GS,0,0.0)
IF(IBUG2.EQ.0) GO TO 30
WRITE(6,80)
ISTOP = ISZ/2
DO 20 I = 1,ISTOP
II = I + ISTOP
WRITE(6,90) I,REA(I),RIMA(I),II,REA(II),RIMA(II)
20 CONTINUE
30 WRITE(6,80) 44
CHANGE TO RECTANGULAR COORDINATES

DO 40 I = 1, ISZ
  GAIN = COMP(I,1)
  COMP(I,1) = GAIN * COS(COMP(I,2) * 3.1415927 / 180.)
  COMP(I,2) = GAIN * SIN(COMP(I,2) * 3.1415927 / 180.)
40 CONTINUE

C*** IF GAMMA IS BEING INPUT BY USER, JUMP DIRECTLY TO THE
  COMPENSATION SPECTRUM

IF(IFLAGS(1).EQ.2) GS(10) = GAMMA
IF(IFLAGS(1).EQ.2) GO TO 60

C*** FIND THE TRANSFER FUNCTIONS FOR THE T/C'S -VS- GAS STREAM.
  CALL TRANGS(1,17,CS,TC,TRAN,COMP)
  CALL TRANGS(2,17,CS,TC,TRAN,COMP)

C*** FIND THE TRANSFER FUNCTION BETWEEN THE TWO T/C'S.
  CALL TRANTC(17,TRAN)

C*** READ IN THE DIGITIZED TEST DATA AND FIND THE AUTO AND CROSS POWER
  SPECTRUMS NEEDED FOR THE MEASURED TRANSFER FUNCTION.
  CALL POWER(DATA3,DATA10,DATA1C,XX,YY,XY,WINDO)

C*** EVALUATE THE MEASURED TRANSFER FUNCTION.
  CALL MEASUR(XX,YY,XY,RTRAN,COHR)

C*** INTERPOLATE FOR A MEASURED VALUE OF GAMMA.
  CALL INTERP(TRAN,RTRAN,COHR,GS)

C*** INITIALIZE THE CALCOMP PLOTTER
  CALL PLOTSP(IBUFF,1000,0,4)

C*** FIND THE COMPENSATION SPECTRUM FOR THE USER SPECIFIED T/C
  IF(IFLAGS(3).EQ.1) CALL TRANGS(2,1,GS,TC,TRAN,COMP)
  IF(IFLAGS(3).EQ.2) CALL TRANGS(1,1,GS,TC,TRAN,COMP)

C*** EVALUATE THE POWER SPECTRAL DENSITY FUNCTION FOR PLOTTING
  CALL PSDF(N(COMP,WINDO,ALSS)

C*** EVALUATE THE COMPENSATION SPECTRA FOR PLOTTING
  CALL CSFN(COMP,WINDO,GS,ALSS)

C*** DONE! DISCONNECT PLOTTER
  CALL PLOT(0.,0.,999)
  WRITE(6,100)
  STOP

100 FORMAT('1','THE COMPENSATION SPECTRUM IN POLAR GAIN AND PHASE')
  FORMAT('2(2X,I4,3X,E13.7,3X,E13.7,6X))
  FORMAT('O',/,'EXECUTION OF THE PROGRAM HAS BEEN COMPLETED!!')
END
BLOCK DATA
COMMON /DATAS/ C(45),TCF(11,9)

C
DATA C /38.926, 1.8746E-3, 2.1226E-6, -2.7962E-10, 3.20708E-2, E249
* 4.86481E-6, -3.8201E-13, -1.0204E-13, 2.63368E-4, -2.4880E-8, E249
* 1.4692E-11, -1.6870E-15, 3.10239, 1.05268E-2, -1.81028E-6, E249
* 1.1490E-10, 3.92288E-2, 4.8327E-6, 3.34578E-9, -1.7809E-12, E249
* 2.25448E-16, 1.976413E-4, 3.21218E-8, -1.8888E-11, 2.9097E-15, E249
* 1.89988E-5, 1.40238E-2, 2.7857E-6, 2.47338E-1, -3.4000E-6, E249
* 1.3750, 1.9429E-6, 1.40418E-2, 2.14003E-5, 2.44138E-1, E249
* -3.4500E-6, 1.369, 1.9859E-5, 1.40628E-2, 2.7091E-6, E249
* 0.23937, 1.9859E-5, 1.40628E-2, 2.7091E-6, 2.44138E-1, E249
* -1.89988E-5, 1.40238E-2, 2.7857E-6, 2.47338E-1, -3.4000E-6, E249
* 1.3750, 1.9429E-6, 1.40418E-2, 2.14003E-5, 2.44138E-1, E249
* -3.4500E-6, 1.369, 1.9859E-5, 1.40628E-2, 2.7091E-6, E249
* 0.23937, 1.9859E-5, 1.40628E-2, 2.7091E-6, 2.44138E-1, E249
* -885.0557E-3, -110.0944E-2, -130.4843E-2, -150.4887E-2, E249
* -23.4314E+1, 92.665SE+1, 23.1347E+2, 163-3668, E249
* -14.7236E+1, 22.4518E+1, 80.7976E+1, 17.7489E+1, E249
* -40.5648E+1, 891.8105, 31215.7227E-2, -14.4194E-1, E249
* -40.4444E-1, 42.3674, -18.7673, -24.9707, E249
* -14.0391, -192.8691E-1, -561.0958E-1, -911.3308E-1, E249
* -101.6005, -299.8243E-1, 27.6855E-1, 0.0, E249
* 16.7773, 73.3276E-1, 53.3789E-1, 30.3267, E249
* 152.4649E-1, 371.1915E-1, 435.6711E-1, 677.5431E-1, E249
* 1035.7422E-2, 10+0.0, -669.6789E-2, 10+0.0, 2.6058594/ E249
END

C
C SUBROUTINE CHECK
C
C***************************************************************************
C SUBROUTINE 'CHECK' CHECKS THE USER INPUT FLAGS FOR ERRORS AND 
C TERMINATES THE PROGRAM IF ERRORS ARE FOUND
C***************************************************************************

C CALLED BY MAIN PROGRAM
C CALLS - TERM: TERMINATES THE PROGRAM DUE TO ERROR IN IFLAGS
C WRITES OUT AN APPROPRIATE MESSAGE

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

C COMMON /INPUTS/ IFLAGS(12),TCDATA(4,2),GAS(4),FREQ(4),CHANL(9), 
C IAVDAT(2),IBLSZ,IREC(10),TIME(2),IBSZ,GAMMA,NREC(10),NRECS(2),E249
C PLTFRQ,TIMTEM,IDEBUG,IBUG2,ITHRSH
C
C IF(IFLAGS(1).GT.3.OR.IFLAGS(1).LT.1) CALL TERM(1)

46
IF(IFLAGS(2).GT.3.OR.IFLAGS(2).LT.0) CALL TERM(2) 
IF(IFLAGS(3).GT.2.OR.IFLAGS(3).LT.0) CALL TERM(3) 
IF(IFLAGS(4).GT.2.OR.IFLAGS(4).LT.1) CALL TERM(4) 
IF(IFLAGS(5).GT.2.OR.IFLAGS(5).LT.1) CALL TERM(5) 
IF(IFLAGS(6).GT.4.OR.IFLAGS(6).LT.1) CALL TERM(6) 
IF(IFLAGS(7).GT.2.OR.IFLAGS(7).LT.1) CALL TERM(7) 
IF(IFLAGS(8).GT.2.OR.IFLAGS(8).LT.0) CALL TERM(8) 
IF(IFLAGS(9).GT.2.OR.IFLAGS(9).LT.0) CALL TERM(9) 
IF(IFLAGS(10).GT.2.OR.IFLAGS(10).LT.0) CALL TERM(10) 
IF(IFLAGS(11).GT.2.OR.IFLAGS(11).LT.0) CALL TERM(11) 
IF(IFLAGS(12).GT.2.OR.IFLAGS(12).LT.0) CALL TERM(12) 
IF(IFLAGS(1).EQ.2.AND.Gamma.EQ.0) CALL TERM(13) 
IF(IFLAGS(5).EQ.1.AND.IREC(1).EQ.0) CALL TERM(14) 
IF(IFLAGS(11).EQ.2.AND.TIME(1).EQ.0) CALL TERM(15) 
IF(TIME(2).LT.TIME(1)) CALL TERM(16) 
IF(IFLAGS(8).EQ.2.AND.NREC(1).EQ.0) CALL TERM(17) 
IF(IFLAGS(8).EQ.1.AND.NREC(2).EQ.0) CALL TERM(18) 
IF(IFLAGS(12).EQ.2.AND.TIMTEM.EQ.0) CALL TERM(19) 
RETURN 
END 

C SUBROUTINE CSFN(COMP,WINDO,GS,ALSS) 
C
C**************************************************************************
C CSFN EVALUATES THE COMPENSATED SPECTRUM FUNCTION FOR PLOTTING
C**************************************************************************
C
C IDENTIFICATION
C
C** COMP - COMPENSATION SPECTRUM
C -- INPUT
C
C** WINDO - P301 WINDOWING FUNCTION
C -- INPUT
C
C** GS - ARRAY CONTAINING THE GAS STREAM PARAMETERS, NEEDED IN THE
C PLOTTING ROUTINE
C -- INPUT
C
C** ALSS - AREA LINE SHAPE SquARED
C -- INPUT
C
C CALLED BY MAIN PROGRAM
C CALLS - PLT1: PLOTS THE INPUT ARRAY
C PLT2: PLOTS THE TWO INPUT ARRAYS ON ONE PAGE
C FFT: A ROUTINE THAT PERFORMS THE DIRECT AND
C THE INVERSE FOURIER TRANSFORMS
C
C FILES USED:
C 6 - WRITES THE FUNCTIONS EVALUATED TO THIS FILE
C IF THE USED SO DESIRES (IBUG2 = 1)
C 13 - READS THE SCALED DIGITIZED 3 MIL DATA FROM
C THIS FILE
C 14 - READS THE SCALED DIGITIZED 10 MIL DATA FROM
C THIS FILE
C
C*********************************************************************************************
C DIMENSION COMP(1024,2),HANN(2048),DATA(2048),GS(10),
* WINDO(2048),DATAF(1025),DATA2(2048),TEMP(2048),DUMIMG(1025),
* CT(1025),ST(1025),POLAR(1024),POLAR2(1024)
COMPLEX ARRAY(2048),CCOMP(1024),ARRAY(2048)
COMMON /INPUTS/ IFLAGS(12),TCDATA(4,2),CAS(4),FREQ(4),CHANL(9),
* T1AVDAT(2),IBLSZ,IREC(10),TIME(2),IBSZ,GAMMA,NREC(10),NRECS(2)
* PLTFRQ,TI4TME,IDEBUG,IBUG2,ITHRSH
C
C IF(IBUG2.EQ.0.AND.IL6AGS(5).EQ.2) RETURN
C
C*** SET UP THE HANNING WINDOWING FUNCTION
DO 10 I = 1,IBLSZ
    HANN(I) = 1.0 + COS(2.0*3.1415927*(I-1)/IBLSZ)
10 CONTINUE
ISIZ = IBSZ - 1
DO 15 I = 1,ISIZ
    II = I + ISIZ
    TEM = HANN(II)
    HANN(II) = HANN(I)
    HANN(I) = TEM
15 CONTINUE
C
C*** LOOP TO READ IN THE SCALED DIGITIZED TEST DATA
REWIND 13
REWIND 14
IEND = 0
IREAD = 0
DO 260 I = 1,400
    IF(IEND.EQ.1) GO TO 270
    IF(IREAD.EQ.1) GO TO 20
    IF(IFLAGS(3).EQ.1) READ(13,END=270) (DATA(J),J=1,IBLSZ),TMEAN
    IF(IFLAGS(3).EQ.1) READ(14,END=270) (DATA(J),J=1,IBLSZ),TMEAN
    IREAD = 0
    GO TO 40
20 DO 30 J = 1,IBLSZ
    DATA(J) = TEMP(J)
30 CONTINUE
    TMEAN = TMN2
    IREAD = 0
C
C*** CHECK TO SEE IF THIS RECORD IS IN THE DESIRED DATA
40 DO 50 J = 1,10
    IF(IREC(J).EQ.1) GO TO 60
50 CONTINUE
    GO TO 260
C
C*** RECORD IS DESIRED: FIND COMPOSITE INSTANTANEOUS TIME WAVEFORM
IF(IFLAGS(9).EQ.2) GO TO 180

C* COMPENSATED DATA DESIRED — TIME DOMAIN
C— READ IN ANOTHER DATA BLOCK FOR SHIFTING
IF(IFLAGS(3).EQ.1) READ(13,END=90) (DATA2(J),J=1,IBLSZ),TMN2
IF(IFLAGS(3).EQ.2) READ(14,END=90) (DATA2(J),J=1,IBLSZ),TMN2

IREAD = 1
DO 70 J = 1,IBLSZ
TEMP(J) = DATA2(J)
70 CONTINUE
ISZ = IBSZ - 1
DO 80 J = IBSZ,IBLSZ
DATA2(J) = DATA2(J-1024)
80 CONTINUE
DO 85 J = 1,ISZ
DATA2(J) = DATA(J+1024)
85 CONTINUE
GO TO 100
90 IEND = 1
DO 95 J = 1,IBLSZ
DATA2(J) = 0.0
95 CONTINUE
C— WINDOW THE DATA WITH THE HANNING WINDOW
DO 110 J = 1,IBLSZ
DATA(J) = DATA(J) * HANN(J)
DATA2(J) = DATA2(J) * HANN(J)
ARRAY(J) = CMPLX(DATA(J),0.0)
ARRAY2(J) = CMPLX(DATA2(J),0.0)
110 CONTINUE
C— PERFORM THE FORWARD TRANSFORM
CALL FFT(1,IBLSZ,ARRAY)
CALL FFT(1,IBLSZ,ARRAY2)
C— APPLY THE THRESHOLD LEVEL
ISZ = IBSZ - 1
PMAX = 0.0
PMAX2 = 0.0
DO 115 J = 2,ISZ
CT1 = REAL(ARRAY(J))
CT2 = REAL(ARRAY2(J))
ST1 = AIMAG(ARRAY(J))
ST2 = AIMAG(ARRAY2(J))
POLAR(J) = CT1**2 + ST1**2
POLAR2(J) = CT2**2 + ST2**2
IF(POLAR(J).GT.PMAX) PMAX = POLAR(J)
IF(POLAR2(J).GT.PMAX2) PMAX2 = POLAR2(J)
115 CONTINUE
X = PMAX * 10.0**(ITHRSH/10.0)
X2 = PMAX2 * 10.0**(ITHRSH/10.0)
C— DIVIDE BY THE COMPENSATION SPECTRUM
DO 120 J = 2,ISZ
CCOMP(J) = CMPLX(COMP(J,1),COMP(J,2))
IF(POLAR(J).GE.X) ARRAY(J) = ARRAY(J) / CCOMP(J)
120 CONTINUE
IF(POLAR(J).LT.X) ARRAY(J) = CMPLX(0.0,0.0)  E249
IF(POLAR2(J).GE.X2) ARRAY2(J) = ARRAY2(J) / CCOMP(J)  E249
IF(POLAR(J).LT.X2) ARRAY(J) = CMPLX(0.0,0.0)  E249
120 CONTINUE  E249
   DO 130 J = 2,ISZ  E249
      DUMIMG(J) = - COMP(J,2)  E249
      CCOMP(J) = CMPLX(COMP(J,1),DUMIMG(J))  E249
      JJ = IBLSZ * 2 - J  E249
      IF(POLAR(J).GE.X) ARRAY(JJ) = ARRAY(JJ) / CCOMP(J)  E249
      IF(POLAR(J).LT.X2) ARRAY(JJ) = CMPLX(0.0,0.0)  E249
      IF(POLAR2(J).GE.X2) ARRAY2(JJ) = ARRAY2(JJ) / CCOMP(J)  E249
      IF(POLAR2(J).LT.X2) ARRAY2(JJ) = CMPLX(0.0,0.0)  E249
130 CONTINUE  E249
C—— INVERSE FOURIER TRANSFORM THE DATA  E249
   CALL FFT(-1,IBLSZ,ARRAY)  E249
   CALL FFT(-1,IBLSZ,ARRAY2)  E249
C—— DE-HANNING THE DATA (IN CHANNELS THAT ARE TO BE USED)  E249
      IST = IBLSZ / 4  E249
      IEN = 3 * IST  E249
      DO 140 J = IST, IEN  E249
         DATA(J) = REAL(ARRAY(J)) / HANN(J)  E249
         DATA2(J) = REAL(ARRAY2(J)) / HANN(J)  E249
140 CONTINUE  E249
      IF(IEND.EQ.1) GO TO 160  E249
C—— SHIFTING THE DATA FOR 'GOOD' DATA BLOCK  E249
      DO 150 J = 1,ISZ  E249
         DATA(J) = DATA(J + ISZ/2)  E249
150 CONTINUE  E249
      DO 160 J = 1,ISZ  E249
         DATA(J+ISZ) = DATA2(J + ISZ/2)  E249
160 CONTINUE  E249
      GO TO 180  E249
165 CONTINUE  E249
180 IF(IFLAGS(10).EQ.2) GO TO 230  E249
C  E249
C*** FREQUENCY DOMAIN  E249
      DO 190 J = 1,IBLSZ  E249
         DUMM = DATA(J) * WINDO(J)  E249
         ARRAY(J) = CMPLX(DUMM,0.0)  E249
190 CONTINUE  E249
      CALL FFT(1,IBLSZ,ARRAY)  E249
      DO 200 J = 1,IBSZ  E249
         CT(J) = REAL(ARRAY(J)) * 2.0  E249
         ST(J) = AIMAG(ARRAY(J)) * 2.0  E249
200 CONTINUE  E249
C  E249
C*** PERFORM THE APPROPRIATE SCALING  E249
      DO 220 J = 2,IBSZ  E249
         DATAF(J) = (CT(J)**2 + ST(J)**2) / 2.0  E249
         IF(IFLAGS(6).NE.4) GO TO 210  E249
      50
DATAF(J) = SQRT(DATAF(J))
GO TO 220
210 DATAF(J) = DATAF(J) / (FREQ(1)*ALSS)
IF(IFLAGS(6).EQ.2) DATAF(J) = 10.0* ALOG10(DATAF(J))
IF(IFLAGS(6).EQ.3) DATAF(J) = SQRT(DATAF(J))
220 CONTINUE
C
C*** PLOT THE DATA
230 IF(IFLAGS(6).EQ.1.AND.IFLAGS(10).EQ.2) CALL PLT1(DATA,2,I,TMEAN)
   ! CALL PLT2(DATA,DATAF,2,GS,I,TMEAN)
IF(IBUG2.EQ.0) GO TO 260
WRITE(6,280) I
JSTOP = IBLSZ/4
DO 240 J = 1,JSTOP
   JJ = J + JSTOP
   JJJ = JJ + JSTOP
   JJJJ = JJJ + JSTOP
WRITE(6,290) J,DATA(J),JJ,DATA(JJ),JJJ,DATA(JJJ),JJJJ,DATA(JJJJ)
240 CONTINUE
IF(IFLAGS(10).EQ.2) GO TO 260
WRITE(6,300) I
JSTOP = IBLSZ/8
DO 260 J = 1,JSTOP
   JJ = J + JSTOP
   JJJ = JJ + JSTOP
   JJJJ = JJJ + JSTOP
WRITE(6,290) J,DATAF(J),JJ,DATAF(JJ),JJJ,DATAF(JJJ)
   ! JJJ,DATAF(JJJ),JJJJ,DATAF(JJJJ)
260 CONTINUE
260 CONTINUE
270 RETURN
280 FORMAT('I','THE INSTANTANEOUS TIME DATA FOR RECORD ','I4)
290 FORMAT(' ',4(2X,'I = ',I4,3X,E13.7,6X))
300 FORMAT('I','THE INSTANTANEOUS FREQUENCY DATA FOR RECORD ','I4)
END
C
C SUBROUTINE FFT(ISET,N,A)
C
C******************************************************************************
C SUBPROGRAM FFT CALCULATES FINITE COMPLEX FOURIER TRANSFORM OR
C THE INVERSE TRANSFORM OF A COMPLEX INPUT ARRAY
C******************************************************************************
C --- IDENTIFICATION ---
C
C** ISET - TYPE OF TRANSFORM
C    1 - DIRECT TRANSFORM
C    -1 - INVERSE TRANSFORM
C
C** N - NUMBER OF DATA POINTS — MUST BE AN INTEGER POWER OF TWO
C OR THE TRANSFORM WILL NOT BE COMPUTED
C INPUT

C** A - ARRAY OF N COMPLEX VALUES TO WHICH TO TRANSFORM IS APPLIED
C AND PUT BACK INTO THE ARRAY
C - INPUT AND OUTPUT
C
C CALLED FROM POWER SUBPROGRAM
CSFN SUBPROGRAM
C
C IDIOSYNCRACIES
C
C 1. THE FIRST ELEMENT OF THE DIRECT TRANSFORM IS THE MEAN
C (ZERO FREQUENCY) VALUE. THE SECOND THROUGH \((N/2 + 1)\) VALUES ARE FOR POSITIVE FREQUENCIES, AND THE REMAINING
C \((N/2 - 2)\) VALUES ARE FOR NEGATIVE FREQUENCIES.
C
C 2. FOR THE DIRECT TRANSFORM, THE RESULTS FOR NEGATIVE
C FREQUENCIES ARE FOLDED ABOUT THE \((N/2 + 1)\) POINT.
C THIS IS THE FREQUENCY FOR THE \((N/2 + 2)\) POINT, AND SO ON.
C
C 3. THE EQUIVALENCE IS USED ONLY FOR DATA SHIFTING WITHIN
C THIS PARTICULAR SUBROUTINE
C
C*********************************************************************************
C
COMPLEX A(2048),WN,WJ,T
DIMENSION W(2)
EQUIVALENCE (W(1),WN),(W(1,W(1)),(W2,W(2)))
C
C CHECK NUMBER OF DATA
M=IFIX(ALOG(FLOAT(N))/ALOG(FLOAT(2)) + .01 )
IF(2**M.EQ.N) GO TO 10
WRITE(6,110) N
RETURN
C
10 FN = FLOAT(N)
IF ( ISET ) 40,20,20
C
C DIRECT TRANSFORM IS REQUESTED
20 CONTINUE
W1 = COS(6.283185/FN)
W2 =-SIN(6.283185/FN)
DO 30 J=1,N
A(J) = A(J)/FN
30 CONTINUE
GO TO 50
C
C INVERSE TRANSFORM IS REQUESTED
40 CONTINUE
W1 = COS(6.283185/FN)
W2 =-SIN(6.283185/FN)
50 CONTINUE
C SHUFFLE INPUT DATA ACCORDING TO REVERSE BIT PATTERN OF ITS SUBSCRIPTS BEFORE TRANSFORMING

DO 70 J=2,N
  JJ = J-1
  JI = 1
  MML = N
  MML = MML/2
  IF ( MOD(JJ,2).EQ.1 ) JI = JI + MML
  JJ = JJ/2
  IF ( JJ.NE.0 ) GO TO 60
  IF ( JI.LE.J ) GO TO 70
  T = A(J)
  A(J) = A(JI)
  A(JI) = T
70 CONTINUE

C
LM1 = 1
MML = N/2
KLIM = N-1

C DO CASE FOR L=1
DO 80 KO=1,KLIM,2
  K1 = KO+1
  T = A(K1)
  A(K1) = A(KO) - T
  A(KO) = A(KO) + T
80 CONTINUE

C DO CASES FOR REMAINING L'S
DO 100 L=2,M
  LM1 = LM1*2
  LPO = LM1+2
  MML = MML/2
  KLIM = N - LPO +1
  JLIM = LM1 - 1

C DO CASE FOR J=0
DO 90 KO = 1,KLIM,LPO
  K1 = KO + LM1
  T = A(K1)
  A(K1) = A(KO) - T
  A(KO) = A(KO) + T
90 CONTINUE

C DO CASE FOR REMAINING J'S
DO 100 J=2,JLIM
  J1 = J+1
  KLIM = KLIM + 1
  WJ = WN**(J*MML)
  DO 100 KO = J1,KLIM,LPO
  K1 = KO + LM1
  T = A(K1)*WJ
  A(K1) = A(KO) - T
  A(KO) = A(KO) + T
100 CONTINUE

53
CONTINUE

FORMAT(//'******* ERROR IN FFT ** NUMBER OF DATA NOT A POWER OF', *, ' 2'//'**** N - ',I5,' **** TRANSFORM NOT CALCULATED ****'/)
RETURN
END

SUBROUTINE GET(WW)

C SUBROUTINE GET ACCESSES THE FOURIER TRANSFORMS DESIRED OFF OF DISK 16 OR 16 DEPENDING ON THE TC FOR WHICH THE COMPENSATION SPECTRUM WAS DESIRED. 'GET' ALSO PERFORMS THE APPROPRIATE SCALEING FOR THE DESIRED PSD FUNCTION AND FORMS THE AUTO POWER SPECTRUM.

C---------------------------------------------------------------------
C IDENTIFICATION
C---------------------------------------------------------------------
C WW - ARRAY INTO WHICH THE AUTO POWER SPECTRUM WILL BE PLACED
C OUTPUT
C CALLED BY PDSFN SUBPROGRAM
C FILES USED:
C 16 - READS THE FOURIER TRANSFORM OF THE 3 MIL THERMOCOUPLE FROM THIS FILE.
C 10 - READS THE FOURIER TRANSFORM OF THE 10 MIL THERMOCOUPLE FROM THIS FILE.

DIMENSION WW(1025),CT(1025),ST(1025)
COMMON /INPUTS/ IFLAGS(12),TCDATA(4,2),GAS(4),FREQ(4),CHANL(9),*
IADVAT(2),IBLSZ,IREC(10),TIME(2),IBSZ,GAMMA,NREC(10),NRECS(2),*
PLTFRQ,TIMTEM,DEBUG,IBUC2,ITHRSH

C*** LOOP TO READ ALL THE DATA OFF OF THE DISK
IF(IFLAGS(3).EQ.1) READ(16) (CT(J),J-1,IBSZ),(ST(J),J-1,IBSZ)
IF(IFLAGS(3).EQ.2) READ(16) (CT(J),J-1,IBSZ),(ST(J),J-1,IBSZ)

C*** PERFORM THE APPROPRIATE SCALING AND THE PSD FUNCTION
DO 10 J = 1,IBSZ
WW(J) = (CT(J)**2 + ST(J)**2) / 2.0
10 CONTINUE
RETURN
END

SUBROUTINE CSPARM(TC,GS)

GSPARM CALCULATES THE GAS STREAM PARAMETERS AND PUTS THEM INTO THE ARRAY "GS"
C** IDENTIFICATION **
C ** TC - ARRAY OF THERMOCOUPLE PARAMETERS, NEEDED TO FIND GAS STREAM
C ** GS - THE ARRAY OF GAS STREAM PARAMETERS CALCULATED, LISTED BELOW
C ** INPUT
C ** OUTPUT
C
C 1. DENSITY (RHO)
C 2. THERMAL CONDUCTIVITY (XK)
C 3. SPECIFIC HEAT (CP)
C 4. SPECIFIC HEAT RATIO (GA)
C 5. VISCOSITY (XMU)
C 6. SONIC VELOCITY (SONVL)
C 7. KINETIC VISCOSITY (G)
C 8. PRANDTL NUMBER (PR)
C 9. MEAN GAS VELOCITY (U)
C 10. AERODYNAMIC PARAMETER (GMA)
C
C CALLED BY MAIN PROGRAM
C
C FILES USED:
C 6 - WRITES THE ESTIMATED VALUE OF GAMMA TO THIS FILE
C
C ** IDIOSYNCRACIES **
C
C EQUATIONS USED TO CALCULATE PARAMETERS 2, 3, AND 4 ARE DEPENDENT ON THE FUEL TO AIR RATIO, GAS(1).
C
C** SETTING THE LIMITS AND CHECKING ON THE FUEL/AIR RATIO
C
C DIMENSION GS(10), TC(4)
C COMMON /INPUTS/ IFLAGS(12), TCDATA(4,2), GAS(4), FREQ(4), CHANL(9),
* IAVDAT(2), IBLSZ, IREC(10), TIME(2), IBSZ, GAMMA, NREC(10), NRECS(2),
* PLTFRQ, TIMTEM, IDEBUG, IBUG2, ITHRSH
C COMMON /DATAS/ C(45), TCF(11,9)
C
C T=GAS(2)
P=GAS(3)
XM=GAS(4)
FA=GAS(1)
C
C*** SETTING THE LIMITS AND CHECKING ON THE FUEL/AIR RATIO
C
C RLM1=.015
C RLM2=.025
C IF(FA.LE.RLM1) GO TO 10
C IF(FA.GT.RLM1.AND.FA.LE.RLM2) GO TO 20
C IF(FA.GT.RLM2) GO TO 30
C
C*** GS PARAMETERS 2,3,4 FOR FUEL/AIR < .015
10 \( XK = C(26) \times T + C(27) \)
   \( CP = C(28) \times T + C(29) \)
   \( GA = C(30) \times T + C(31) \)
   GO TO 40

C

C*** GS PARAMETERS 2,3,&4 FOR .015 < FUEL/AIR < .025

20 \( XK = C(32) \times T + C(33) \)
   \( CP = C(34) \times T + C(35) \)
   \( GA = C(36) \times T + C(37) \)
   GO TO 40

C

C*** GS PARAMETERS 2,3,&4 FOR FUEL/AIR > .025

30 \( XK = C(38) \times T + C(39) \)
   \( CP = C(40) \times T + C(41) \)
   \( GA = C(42) \times T + C(43) \)

C

C*** CALCULATING THE REST OF THE GAS STREAM PARAMETERS

40 \( XMU = C(44) \times T + C(46) \)
   \( RHO = 2.6983 \times (P/(T+460.)) \)
   \( SONVL = 41.454 \times (\sqrt{GA \times (T+460.)}) \)
   \( C = XMU/RHO \)
   \( PR = 3600. \times XMU \times CP \times XK \)
   \( U = XM \times SONVL \)
   \( GMA = 0.48 \times XK \times (PR^{**}0.3333) \times (\sqrt{U}) \)
   \( GMA = GMA/(\sqrt{G} \times TC(1) \times TC(3)) \)

C

C*** PLUGGING THE PARAMETERS INTO THE GS ARRAY

\( GS(1) = RHO \)
\( GS(2) = XK \)
\( GS(3) = CP \)
\( GS(4) = GA \)
\( GS(5) = XMU \)
\( GS(6) = SONVL \)
\( GS(7) = G \)
\( GS(8) = PR \)
\( GS(9) = U \)
\( GS(10) = GMA/3600. \)

C

GMAMET = GS(10) * .168279
IF(IFLAGS(1).EQ.1) WRITE(6,50) GMAMET
50 FORMAT('1', 'THE ESTIMATED GAMMA IS ', F13.7)
RETURN

C

SUBROUTINE INPUT

C

C**********************************************************************************************************E249

C THIS SUBROUTINE INPUTS ALL USER EDITS NEEDED FOR THE PROGRAM E249
C**********************************************************************************************************E249

C — IDENTIFICATION —

C

C

56
### 1) TCDATA - Data for the Large (COL1) and Small (COL2) Wire T/C
- (1) - Length of Support Wire (CM)
- (2) - Half the Total Length of Smaller Wire (CM)
- (3) - Diameter of the Support Wire (CM)
- (4) - Diameter of the Smaller Wire (CM)

### 2) GAS - Gas Stream Data
- (1) - Fuel to Air Ratio
- (2) - Mean Gas Temperature (K)
- (3) - Mean Gas Pressure (Pa - N/m²)
- (4) - Mach Number

### 3) FREQ - Frequency Information for Transfer Functions
- (1) - Delta-T Setting (Sec)
- (2) - Starting Frequency (Hz)
- (3) - Ending Frequency (Hz)
- (4) - Frequency Increment (Hz)

**Note:** The program converts Delta-T to a Delta-F setting and comes as close as possible to the frequencies specified.

### 4) CHANL - Channel Information
- (1) - Channel A (3MIL) Amplifier Gain (Volts)
- (2) - Voltage Ratio of FM Tape Input - TO - FM Tape Output
- (3) - Channel A DC Offset (Volts)
- (4) - (9) - Same as above for Channels B (10MIL) and C (DC)

### 5) IAVDAT - Data Required for Ensemble Averaging
- (1) - Starting Record for the Averaging (1, 2, Etc)
- (2) - Number of Records Desired in the Averaging

### 6) IBLSZ - Data Block Size

### 7) PLTFRQ - Frequency at Which to End Plots (Hz)
**Notes:** Plots start at zero frequency. If PLTFRQ is invalid, plots will cover the entire data block.

### 8) IFLAGS - Flags for Various User Options
- (1) - Gamma Used for Compensation Spectrum
  - 1 - Calculate Measured Gamma
  - 2 - Use a User Entered Estimated Gamma
  - 3 - Use a Previously Evaluated Compensation Spectrum
- (2) - The T/C Material Code
  - 0 - Not Applicable (IFLAGS(1) = 3)
  - 1 - PT / 6%RH
  - 2 - PT / 30%RH
  - 3 - CR / Al
- (3) - T/C Used for Compensation Spectrum
  - 0 - Not Applicable (IFLAGS(1) = 3)
  - 1 - 3 Mil
  - 2 - 10 Mil
- (4) - Plot of Compensation Spectrum Desired?
  - 1 - Yes
  - 2 - No
- (5) - Plot of Instantaneous Data Desired?
  - 1 - Yes
  - 2 - No
- (6) - Type of Frequency Domain Data Desired for Plots
  - 1 - Regular Power Spectral Density
2 - LOG POWER SPECTRAL DENSITY (10*LOG(PSD))
3 - LINEAR POWER SPECTRAL DENSITY (SQRT(PSD))
4 - NARROWBAND FREQUENCY SPECTRUM

(7) - PLOT OF AVERAGED FREQUENCY DOMAIN DATA DESIRED?
1 - YES
2 - NO

(8) - TYPE OF AVERAGING DESIRED FOR FREQUENCY DATA
0 - NOT APPLICABLE (NO PLOT OR PRINT-OUT DESIRED)
1 - AVERAGE IN USER SPECIFIED NUMBER OF RECORDS
2 - AVERAGE ONLY ONE AT A TIME

(9) - GENERAL PLOTTING FLAG
0 - NOT APPLICABLE (NO PLOTS OR PRINT-OUTS DESIRED)
1 - PLOT COMPENSATED DATA
2 - PLOT UNCOMPENSATED DATA

(10) - GENERAL PLOTTING FLAG
0 - NOT APPLICABLE (NO PLOTS OR PRINT-OUTS DESIRED)
1 - PLOT TIME AND FREQUENCY DATA
2 - PLOT TIME ONLY

(11) - GENERAL PLOTTING FLAG
0 - NOT APPLICABLE (NO PLOTS OR PRINT-OUTS DESIRED)
1 - PLOT FULL TIME RANGE
2 - PLOT PARTIAL TIME RANGE

(12) - PLOTTING FLAG FOR SCALING OF INSTANTANEOUS DATA
0 - NOT APPLICABLE (NO PLOTS OR PRINT-OUTS DESIRED)
1 - SCALE EACH RECORD TO IT'S OWN MAX TEMPERATURE
2 - SCALE ALL RECORDS TO A USER INPUT TEMPERATURE

9) IDEBUG - FLAG FOR INTERMEDIATE WRITES
0 - NO
1 - YES

10) IBUG2 - FLAG FOR WRITING ALL FUNCTIONS THAT ARE GENERATED (COMPENSATION SPECTRUM, TIME AND FREQUENCY DOMAIN DATA)
0 - NO
1 - YES

11) GAMMA - VALUE OF GAMMA TO USE IF ONE IS INPUT

12) IREC - RECORDS DESIRED FOR PLOTTING OF INSTANTANEOUS SPECTRUM
MAX OF 10 RECS, NEED NOT BE CONSECUTIVE.
NOTE: IF IBUG2 = 1, AT LEAST ONE RECORD IS REQUIRED

13) TIME - TIMES FOR PLOTTING IF PARTIAL TIME RANGE
(1) - STARTING TIME (WITH RESPECT TO THE DATA BLOCK)
(2) - ENDING TIME (WITH RESPECT TO THE DATA BLOCK)

14) NREC - RECORDS DESIRED FOR INSTANTANEOUS PLOTTING FREQUENCY DOMAIN DATA (IFLAGS(8) = 2)
MAX OF 10 RECORDS, NEED NOT BE CONSECUTIVE
NOTE: THE ONLY ALLOWABLE RECORDS ARE THE ONES USED FOR THE ENSEMBLE AVERAGING (IAVDAT)

15) NRECS - RECORDS FOR AVERAGING OF FREQUENCY DOMAIN DATA
(1) - STARTING RECORD NUMBER
(2) - NUMBER OF RECORDS IN AVERAGING
NOTE: THE ONLY ALLOWABLE RECORDS ARE THE ONES USED FOR THE ENSEMBLE AVERAGING (IAVDAT)

16) TIMTEM - MAX TEMPERATURE TO WHICH TO SCALE THE TIME DOMAIN DATA
NOTE: ZERO K IS IN CENTER OF AXIS AND PLOT IS SCALED TO POS AND NEG TIMTEM ON 4 INCH AXIS.

17) ITHRSH - RELATIVE THRESHOLD LEVEL NEEDED FOR THE INSTANTANEOUS
TIME WAVEFORM (ENTERED IN DB)
18) PLOTIT - ARRAY INTO WHICH THE USER ENTERS ANY GENERAL
       INFORMATION THAT HE/SHE WISHES TO HAVE PRINTED
       OUT ON THE PLOTS.
       CALLED BY MAIN PROGRAM
       CALLS : PRNTIN - PRINTS USER INPUT IN CARD IMAGE FORMAT
                OUT - PRINTS THE INPUT WITH DISCRIPTIONS
       FILES USED:
                5 - READS THE USER INPUTS FROM THIS FILE
------- IDIOSYNCRACIES -------
       NOTE: IN ORDER FOR THE USER INPUT GAMMA (IFLAGS(1) = 2), OR THE
       PREVIOUSLY EVALUATED COMPENSATION SPECTRUM (IFLAGS(1) = 3)
       OPTIONS TO BE USED, THIS PROGRAM MUST HAVE BEEN RUN BEFORE
       SO THAT VALUES OF THE FOURIER TRANSFORMS HAVE BEEN
       STORED ON DISK FILES 15(3MIL) AND 16(10MIL), THE TIME DOMAIN
       DIGITIZED DATA ON DISK FILES 13(3MIL) AND 14(10MIL), AND THE
       COMPENSATION SPECTRUM ON DISK FILE 12.

COMMON /INPUTS/ IFLAGS(12),TCDATA(4,2),GAS(4),FREQ(4),CHANL(9),
                 IAVDAT(2),IBLSZ,IREC(10),TIME(2),IBSZ,GAMMA,NREC(10),NRECS(2),
                 * PLTFRQ,TIMTEM,DEBUG,IBUC2,ITHRSH
       COMMON /PLOTTR/ PLOTIT(20,3)
       DATA BLNK/'  '

CALL PRNTIN(5,6)
READ (6,60) ((TCDATA(I,J),I=1,4),J=1,2)
READ (6,60) (GAS(I),I=1,2),GAS(4)
READ (5,90) GAS(3)
READ (5,70) FREQ(1)
READ (5,60) (FREQ(I),I=2,4)
READ (5,60) (CHANL(I),I=1,9)
READ (5,50) (IAVDAT(I),I=1,2)
READ (5,50) IBLSZ
READ (5,60) PLTFRQ
READ (5,50) (IFLAGS(I),I=1,12)
READ (6,60) DEBUG
READ (5,60) IBUC2

* CHECK TO SEE IF REST OF DATA IS REQUIRED
       IF(IFLAGS(1).EQ.2) READ (5,90) GAMMA
       IF(IFLAGS(5).EQ.1.OR.IBUC2.EQ.1) READ (5,50) (IREC(I),I=1,10)
       IF(IFLAGS(11).EQ.2) READ (5,60) (TIME(I),I=1,2)
       IF(IFLAGS(8).EQ.1) READ (5,50) (NRECS(I),I=1,2)
       IF(IFLAGS(8).EQ.2) READ (5,60) (NREC(I),I=1,10)
       IF(IFLAGS(12).EQ.2) READ (5,60) TIMTEM
       IF((IFLAGS(9).EQ.1.AND.IFLAGS(5).EQ.1).OR.(IFLAGS(9).EQ.1.AND.
       IBUC2.EQ.1)) READ(5,50) ITHRSH
       DO 10 I = 1,3
       DO 10 J = 1,20
PLOTIT(J,I) = BLNK
10 CONTINUE
DO 20 I = 1, 3
READ(5, 80, END=25) (PLOTIT(J,I), J=1, 20)
20 CONTINUE
25 CALL OUT
IF(IFLAGS(1).EQ.2) GAMMA = GAMMA * 5.9426
DO 30 I = 1, 4
TCDATA(I,1) = TCDATA(I,1)*.032808
TCDATA(I,2) = TCDATA(I,2)*.032808
30 CONTINUE
GAS(2) = 9.0 * (GAS(2)-233.15) / 5.0 - 40.0
GAS(3) = GAS(3) / 6894.8
FREQ(1) = 1.0 / (FREQ(1) * IBLSZ)
IBSZ = IBLSZ/2 + 1
40 RETURN
50 FORMAT (20I5)
60 FORMAT (8F10.0)
70 FORMAT (E10.0)
80 FORMAT (20A4)
80 FORMAT (E20.0)
END

C
C SUBROUTINE INTERP(TRAN,RTRAN,COHR,GS)
C
C******************************************************************************
C INTERP INTERPOLATES BETWEEN THE ESTIMATED TRANSFER FUNCTIONS OF
C THE 10MIL VS 3MIL T/C'S IN ORDER TO EVALUATE A MEASURED GAMMA
C******************************************************************************
C
C—— IDENTIFICATION ——
C
C** TRAN - ARRAY CONTAINING THE ESTIMATED TRANSFER FUNCTIONS
C — INPUT
C
C** RTRAN - ARRAY CONTAINING THE MEASURED TRANSFER FUNCTION
C — INPUT
C
C** COHR - ARRAY CONTAINING THE COHERENCE FUNCTION
C — INPUT
C
C** GS - ARRAY CONTAINING GAMMA VALUE THAT IS UPDATED IN THIS ROUTINE
C — INPUT AND OUTPUT
C
C CALLED BY MAIN PROGRAM
C CALLS - TERM: TERMINATES THE PROGRAM DUE TO NO CALCULATED GAMMA
C FILES USED:
C 6 - WRITES OUT THE GAINS OF THE ESTIMATED TRANSFER
C FUNCTIONS AT EACH FREQUENCY AND THE INTERPOLATED
C VALUE OF GAMMA IF IDEBUG = 1.
IDIOSYCRACIES

INTERP PERFORMS THE FOLLOWING DATA CHECKS:

1. DETERMINES IF THE MEASURED GAIN CROSSES THE THEORETICAL X-FER FUNCTION CURVE.

2. DETERMINES IF THE COHERENCE IS WITHIN SPECIFIED LIMITS OF .8 < Y**2 < 1.005.

3. DETERMINES IF A COMBINATION OF ABOVE ERRORS WOULD RESULT IN NOT HAVING A MEASURED VALUE OF GAMMA.

SET LIMITS FOR COHERENCE FUNCTION

Y2L = .8
Y2U = 1.005

FIND THE APPROPRIATE CHANNELS FOR MEASURED FUNCTION FREQUENCIES

IL = FREQ(2) / FREQ(1) + 1.49
IU = FREQ(3) / FREQ(1) + 1.49
IN = FREQ(4) / FREQ(1) + .05

LOOP THROUGH THE FREQUENCIES

DO 30 IC = IL, IU, IN

ACCESS GAIN FROM THE MEASURED AND ESTIMATED TRANSFER FUNCTIONS

GN = RTRAN(IC, 1)
IF(IDEBUG.EQ.1) WRITE(6, 70) IC, GN
DO 10 IR = 1, 16
IRT = IR + 1
PERGAM = FLOAT(IR + 1) / 10.0
PERGM1 = PERGAM + .1
V1 = TRAN(IR, 3, IC, 1)
V2 = TRAN(IRT, 3, IC, 1)
IF(IDEBUG.EQ.1) WRITE(6, 80) PERGM1, V1

COMPARE THE MEASURED GAIN TO THOSE OF THE ESTIMATED FUNCTIONS

IF(GN.LT.V1) GO TO 10
IF(GN.LE.V2) GO TO 20
10 CONTINUE
GO TO 30

CHECK THE COHERENCE FUNCTION AT THIS FREQUENCY

IF(IDEBUG.EQ.1) WRITE(6, 80) PERGM1, V2
Y2 = COHR(IC)
IF(Y2.GE.Y2L.AND.Y2.LE.Y2U) GO TO 25
IF(IDEBUG.EQ.1) WRITE(6,90) Y2
GO TO 30
C
C*** FIND THE FRACTION OF GAMMA WHERE THE MEASURED GAIN FALLS
25 R1 = .2 * (.1*(IR-1)
FRGMA = R1 + (GN-V1)*(.1)/(V2-V1)
AVGMA = AVGMA + FRGMA
AVC = AVC+1.
C
C*** IF DEBUG IS SET, WRITE OUT THE INTERPOLATED GAMMAS
26 IF(IDEBUG.EQ.0) GO TO 30
WRITE(6,40)
WRITE(6,45)
FRGMA = FRGMA * GS(10)
FGMAMT = FRGMA * .168279
WRITE(6,50) FGMAMT
C
C*** NEXT FREQUENCY
30 CONTINUE
IF(AVC.LE.O.0) CALL TERM(20)
C
C*** FIND AND PRINT THE AVERAGED GAMMA IF ONE WAS FOUND
AVGMA = AVGMA/AVC
GS(10) = GS(10)*AVGMA
GMAMET = GS(10) * .168279
IF(IDEBUG.EQ.1) WRITE(6,60) GMAMET
RETURN
40 FORMAT(0,T5,'INTERPOLATED GAMMA')
45 FORMAT('+',T4,'
50 FORMAT(0,' CHANNEL (',13,' ) MEASURED GAIN IS ',E13.7)
55 FORMAT(0,6X,'ESTIMATED GAIN FOR ',F4.2,' GAMMA IS ',E13.7)
60 FORMAT(0,6X, 'UNACCEPTABLE COHERENCE OF ',F10.6)
END
C
C
SUBROUTINE INTEST(DATA3,DATA10,DATADC,IFLAG)
C
C******************************************************************************E249
C INTEST READS IN THE DIGITIZED TEST DATA FOR THE 3MIL, 10MIL, *E249
C AND DC CHANNEL T/C. *E249
C******************************************************************************E249
C — IDENTIFICATION —
C
C** DATA3 - ARRAY FOR THE 3MIL TEST DATA
C — OUTPUT
C
C** DATA10 - ARRAY FOR THE 10MIL TEST DATA
C — OUTPUT
** DATADC - ARRAY FOR THE DC CHANNEL TEST DATA**

** IFLAG - FLAG TO SIGNAL THE END OF DATA**

** OUTPUT**

CALLED BY SUBPROGRAM POWER

FILES USED:

4 - READS THE DIGITIZED TEST DATA FROM THIS FILE.

** CONTENTS OF PROGRAM **

DIMENSION DATA3(2048),DATA10(2048),DATADC(2048)

COMMON /INPUTS/ IFLACS(12),TCDATA(4,2),GAS(4),FREQ(4),CHANL(9),
* IAVDAT(2),IBLSZ,IREC(10),TIME(2),IBSZ,GAMMA,NREC(10),NRECS(2),
* PLTFRQ,TIMTEM,IDEBUG,IBUG2,ITHRSH

READ (4,20,END=10) (DATA3(I),I=1,IBLSZ)
READ (4,20,END=10) (DATA10(I),I=1,IBLSZ)
READ (4,20,END=10) (DATADC(I),I=1,IBLSZ)
GO TO 30

10 IFLAG = 1
20 FORMAT (1X,6E13.0)
30 RETURN

** SUBROUTINE MEASUR(XX,YY,XY,RTRAN,COHR)**

MEASUR CALCULATES THE MEASURED TRANSFER FUNCTION OF THE TWO T/C
ALONG WITH THE COHERENCE FUNCTION TO MEASURE ACCURACY.

** CONTENTS OF SUBROUTINE **

** XX - ARRAY CONTAINING THE INPUT AUTO POWER SPECTRUM G 3-3**

** YY - ARRAY CONTAINING THE OUTPUT AUTO POWER SPECTRUM G 10-10**

** XY - ARRAY CONTAINING THE CROSS POWER SPECTRUM G 3-10**

** RTRAN - ARRAY TO CONTAIN THE MEASURED TRANSFER FUNCTION**

** COHR - ARRAY TO CONTAIN THE COHERENCE FUNCTION**

CALLED BY MAIN PROGRAM

63
DIMENSION XX(1025), YY(1025), XY(1025, 2), RTRAN(1025, 2), COHR(1025)

COMMON /INPUTS/ IFLACS(12), TCDATA(4, 2), GAS(4), FREQ(4), CHANL(9),
*   IAVDAT(2), IBLSZ, IREC(10), TIME(2), IBSZ, GAMMA, NREC(10), NRECS(2),
*   PLTFRQ, TIMTEM, IDEBUG, IBUG2, ITHRSH

DO 10 I = 1, IBSZ
  RTRAN(I, 1) = XY(I, 1) / XX(I)
  RTRAN(I, 2) = XY(I, 2) / XX(I)
  COHR(I) = (XY(I, 1)**2 + XY(I, 2)**2) / (XX(I) * YY(I))
  10 CONTINUE

DO 20 I = 1, IBSZ
  RTRAN(I, 1) = SQRT(RTRAN(I, 1)**2 + RTRAN(I, 2)**2)
  RTRAN(I, 2) = 0.0
  20 CONTINUE

RETURN
END

SUBROUTINE OUT

THIS SUBROUTINE PRINTS OUT ALL OF THE USER INPUTS

CALLED BY INPUT SUBROUTINE

FILES USED:

6 - WRITES OUT THE USER INPUT ALONG WITH
   DISCRIPTIONS OF THE INPUTS.

COMMON /INPUTS/ IFLACS(12), TCDATA(4, 2), GAS(4), FREQ(4), CHANL(9),
*   IAVDAT(2), IBLSZ, IREC(10), TIME(2), IBSZ, GAMMA, NREC(10), NRECS(2),
*   PLTFRQ, TIMTEM, IDEBUG, IBUG2, ITHRSH

WRITE(6, 65)
WRITE(6, 70) ((TCDATA(I, 1), TCDATA(I, 2)), I=1, 4)
WRITE(6, 80) GAS
WRITE(6, 90) FREQ
WRITE(6, 100) CHANL
WRITE(6, 110) IAVDAT
WRITE(6, 120) IBLSZ
WRITE(6, 130) PLTFRQ
WRITE(6, 140)
WRITE(6, 150) IFLAGS
WRITE(6, 160) IDEBUG
WRITE(6, 170) IBUG2
IF (IFLAGS(1).EQ.2) WRITE(6, 180) GAMMA
IF (IFLAGS(5).EQ.2.AND.IBUG2.EQ.0) GO TO 30
WRITE(6,190)
DO 10 I = 1,10
  IF(IREC(I).EQ.0) NUM = I-1
  IF(IREC(I).EQ.0) GO TO 20
10 CONTINUE
  NUM = 10
20 WRITE(6,200) (IREC(I),I-1,NUM)
30 IF(IFLAGS(11).EQ.2) WRITE(6,210) TIME
  IF(IFLAGS(8).EQ.1) WRITE(6,220) NRECS
  IF(IFLAGS(8).NE.2) GO TO 60
WRITE(6,230)
DO 40 I = 1,10
  IF(NREC(I).EQ.0) NUM = I-1
  IF(NREC(I).EQ.0) GO TO 50
40 CONTINUE
  NUM = 10
50 WRITE(6,200) (NREC(I),I=1,NUM)
60 IF(IFLAGS(12).EQ.2) WRITE(6,240) TIMTEM
  IF((IFLAGS(9).EQ.1.AND.IFLAGS(5).EQ.1).OR.
    (IFLAGS(9).EQ.1.AND.IBUC2.EQ.1)) WRITE(6,250) ITHRSH
RETURN
65 FORMAT('1',T15,'10 MIL',T62,'3 MIL')
70 FORMAT('F',T15,'_________',T62,'_________')
*E = 'F10.5, 'CM',4X)/,2(5X,'LENGTH OF SMALLER WIRE = ',F10.5E249
*CM',4X)/,2(5X,'DIAMETER OF SMALLER WIRE = ',F10.5, 'CM',4X)
80 FORMAT('0',T5,'FUEL TO AIR RATIO IS ',T35,F10.5, '/T5,
'MEAN GAS TEMPERATURE (K) IS ',T35,F10.5, '/T5,
'MEAN GAS PRESSURE (PA) IS ',T35,F10.5, '*E13.7,//,2(5X,'LENGTH OF SUPPORT WIRE= ',F10.5='CM',4X)
90 FORMAT('0',T5,'DELTA-T = ',E13.7, '/T5,'START FREQ = ',F10.4,6X, *
'END FREQ = ',F10.4,6X, 'FREQ INCREMENT = ',F10.4)
100 FORMAT('0',T13,'GAINT',T24,'INPUT/OUTPUT',T42,'OFFSET','/T13,
*GAIN',T24,'________',T42,'________','/T2,'3 MIL',T10,F10.4,
*T2,F10.5,T40,F10.5,T40,F10.6, '/T2,'10 MIL',T10,F10.4,T25,F10.5,T40,F10.6,
*T2,'DC',T10,F10.4,T25,F10.5,T40,F10.6)
110 FORMAT('0',T5,'THE ENSEMBLE AVERAGING STARTS WITH RECORD ',T35,
*USES ',T35,'RECORDS.')
120 FORMAT('0',T5,'WE HAVE A BLOCKSIZE OF ',T5)
130 FORMAT('0','ALL FREQUENCY DOMAIN PLOTS WILL END AS CLOSE TO ', 
*F10.3,'HZ AS POSSIBLE')
140 FORMAT('0',T6,'FLAGS',T23,'DESCRIPTION',T62,'VALUE')
150 FORMAT('0',T6,'FLAGS',T23,'DESCRIPTION',T62,'VALUE')
*WHERE TO BEGIN PROGRAM CALCULATIONS',T63,T2, '/T8,'2',T15,'T/C MAE249
*TERTIAL CODE',T63,T2, '/T8,'3',T15,'T/C USED FOR COMPENSATION SPECTRUM',E249
*UM',T63,T2, '/T8,'4',T15,'PLOT OF COMPENSATION SPECTRUM DESIRED?',E249
*T63,T2, '/T8,'5',T15,'PLOT OF INSTANTANEOUS DATA?',T63,T2, '/T8,'6',T15,
*TYPE OF SCALING DONE TO FREQUENCY DATA',T63,T2, '/T8,'7',T15,
*PLOT OF AVERAGED FREQUENCY DATA DESIRED?',T63,T2, '/T8,'8',T15,
*MORE OR LESS RECORDS?',T63,T2, '/T8,'9',T15,'COMPENSATEE249
*DATA?',T63,T2, '/T8,'10',T15,'PLOT TIME AND FREQUENCY DOMAINS?',E249
*T63,T2, '/T8,'11',T15,'PLOT FULL TIME RANGE?',T63,T2, '/T8,'12',T15,E249
*TEMPERATURE TO SCALE DATA ON PLOTS',T63,T2)
160 FORMAT('0',T5,'DEBUG IS SET TO ',T11)
170 FORMAT('0',T5,'IBUG2 IS SET TO ',T11)
180 FORMAT('O', 'USER INPUT GAMMA IS ', E13.7, ' M**1.6/SEC')
190 FORMAT('O', 'THE RECORDS OF INSTANTANEOUS FREQUENCY DOMAIN DATA PLOTTED ARE AS FOLLOWS:')
200 FORMAT(' ', 10(5X,113))
210 FORMAT('O', 'TIME DOMAIN DATA IS PLOTTED FROM TIME ', F10.7, ' TO TIME ', F10.7, ' (WITH RESPECT TO THE DATA BLOCK)')
220 FORMAT('O', 'THE AVERAGING OF THE FREQUENCY DOMAIN DATA STARTS WITH RECORD ', I3, ' AND USES ', I3, ' RECORDS.')
230 FORMAT('O', 'RECORDS PLOTTED OF AVERAGED FREQUENCY DOMAIN DATA ARE:')
240 FORMAT('O', 'THE TEMPERATURE TO WHICH THE INSTANTANEOUS TIME DOMAIN DATA IS SCALED IS ', F10.3, ' K')
250 FORMAT('O', 'THE THRESHOLD LEVEL USED FOR TIME DOMAIN DATA IS ', 15, ' DB')

END

C
C SUBROUTINE PLT1(ARRAY,ICODE,NUM,TMEAN)
C
C*********************************************************
C SUBROUTINE PLT1 WILL PLOT THE ONE INPUT ARRAY ALONG WITH SOME HEADERS
C ARRAY ALONG WITH SOME HEADERS
C*********************************************************
C
C IDENTIFICATION
C
C** ARRAY - THE ARRAY FOR WHICH PLOTTING IS DESIRED
C** ICODE - CODE OF THE FUNCTION TO BE PLOTTED SO THAT THE APPROPRIATE LABELING CAN BE DONE
C** NUM - RECORD NUMBER THAT IS BEING PLOTTED
C** TMEAN - THE MEAN DC TEMPERATURE (NEEDED TO PUT ON PLOTS)
C
C DIMENSION ARRAY(2048)
C
COMMON /INPUTS/ IFLAGS(12),TCDBTA(4,2),CAS(4),FREQ(4),CHARACTER(2),IAVDAT(2),IBLSZ,IREC(10),TIME(2),IBSZ,GAMMA,NREC(10),NRECS(2),PLTFRQ,TIMTEM,IDEBUG,IBUG2,1THRSH

C E249
COMMON /PLOTTR/ PLOTIT(20,3)

C*** FIND MAX AND MIN VALUES SO THE ARRAYS CAN BE WINDOWED PROPERLY
ISIZ = IBSZ - 1
N = PLTFRQ / FREQ(1) + 1.5
IF(N.GT.ISIZ) N = ISIZ
IF(ICODE.EQ.2.AND.IFLAGS(11).EQ.2) GO TO 20
ARRMAX = ARRAY(1)
ARRMIN = ARRAY(1)
DO 10 I = 1,IBLSZ
   IF(ARRAY(I).GT.ARRMAX) ARRMAX = ARRAY(I)
   IF(ARRAY(I).LT.ARRMIN) ARRMIN = ARRAY(I)
10 CONTINUE
   IF(ICODE.EQ.1.AND.I.EQ.N) GO TO 40
   CONTINUE
   GO TO 40
20 ICH1 = TIME(1)*IBLSZ*FREQ(1) + 1.49
ICH2 = TIME(2)*IBLSZ*FREQ(1) + 1.49
ARRMIN = ARRAY(ICH1)
ARRMAX = ARRAY(ICH1)
DO 30 I = ICH1,ICH2
   IF(ARRAY(I).LT.ARRMIN) ARRMIN = ARRAY(I)
   IF(ARRAY(I).GT.ARRMAX) ARRMAX = ARRAY(I)
30 CONTINUE

C*** FINDING THE LIMITS OF THE PLOTTING WINDOW
AMAX = ABS(ARRMIN)
IF(ARRMAX.GT.AMAX) AMAX = ARRMAX
RANGE = ARRMAX - ARRMIN
IF(ICODE.EQ.1) XINC = 400.0
   IF(ICODE.EQ.1) XINC = (N-1) * FREQ(1) / 5.0
   IF(ICODE.EQ.2) XINC = 1.0/(5.0*FREQ(1))
   IF(ICODE.EQ.2) XINC = (TIME(2)-TIME(1))/5.0
   IF(ICODE.EQ.2) XINC = AMAX / 2.0
   IF(ICODE.EQ.2) YINC = TIME(1) / 2.0
   IF(ICODE.EQ.1) YINC = ARRMAX / 4.0
   IF(ICODE.EQ.2) YINC = TIME(1) / XINC
   INTN = (ARRMAX + 9.9) / 10.0
   IAMAX = INTN * 10
   YINC = 10.0
   IF(RANGE.GT.40.) YINC = 20.0
   YSTRT = IAMAX - 4.0 * YINC

C*** START THE PLOTTING
IF(ICODE.EQ.1) CALL PLOT(0.5,2.0,-3)
IF(ICODE.EQ.2) CALL PLOT(0.0,4.0,-3)
   IF(ICODE.EQ.1) CALL AXIS(0.0,0.0,14HFREQUENCY (HZ),-14,5.5,0.,0.0,XINC)
   IF(ICODE.EQ.2.AND.IFLAGS(11).EQ.1) CALL AXIS(0.0,-2.,10TIME (SEC),-10,6.,0.,0.0*XINC)
   IF(ICODE.EQ.2.AND.IFLAGS(11).EQ.2) CALL AXIS(0.0,-2.,10TIME (SEC),-10,5.,0.,TIME(1),XINC)
   IF(ICODE.EQ.1.AND.IFLAGS(6).EQ.2) CALL AXIS(0.,0.,8HDECIBELS,8.,-400.0,XINC)
*4.,90.,YSTRT,YINC)
IF(ICODE.EQ.1.AND.IFLAGS(6).NE.2) CALL AXIS(0.0,0.0,11HTemperature)
*11,4.,90.,0.,YINC)
C
IF(ICODE.EQ.2.AND.IFLAGS(12).EQ.1) CALL AXIS(0.0,-2.,15HTemperature)
*E(K),15,4.,90.,-AMAX,YINC)
IF(ICODE.EQ.2.AND.IFLAGS(12).EQ.2) CALL AXIS(0.0,-2.,15HTemperature)
*E(K),15,4.,90.,-TIMTEM,YINC)
C
CALLS I WOULD EXPECT TO PREPARE PLOTTING WINDOW IN DISPLAY.
C
50 IF(ICODE.EQ.1) CALL PHYSOR (1.5,2.25)
C IF(ICODE.EQ.2) CALL PHYSOR (1.,2.25)
C CALL AREA2D(5.,4.)
C IF(ICODE.EQ.1) CALL XNAME("14HFREQUENCY (HZ)",14)
C IF(ICODE.EQ.2) CALL XNAME("10HTIME (SEC)",10)
C IF(ICODE.EQ.1.AND.IFLAGS(6).EQ.2) CALL YNAME("8HDECIBELS",8)
C IF(ICODE.EQ.1.AND.IFLAGS(6).NE.2) CALL YNAME("11HTEMPERATURE",11)
C IF(ICODE.EQ.2) CALL YNAME("15HTEMPERATURE (K)",15)
C
IF(ICODE.EQ.1.AND.IFLAGS(6).EQ.2)
C * CALL GRAF(O.,XINC,N*FREQ(1),YSTRT,YINC,TIMTEM)
C IF(ICODE.EQ.1.AND.IFLAGS(6).NE.2)
C * CALL GRAF(O.,XINC,N*FREQ(1),0.,YINC,ARRMAX)
C IF(ICODE.EQ.2.AND.IFLAGS(11).EQ.1.AND.IFLAGS(12).EQ.1)
C * CALL GRAF(O.,XINC,.5,-AMAX,YINC,AMAX)
C IF(ICODE.EQ.2.AND.IFLAGS(11).EQ.1.AND.IFLAGS(12).EQ.2)
C * CALL GRAF(O.,XINC,.5,-TIMTEM,YINC,TIMTEM)
C IF(ICODE.EQ.2.AND.IFLAGS(11).EQ.2.AND.IFLAGS(12).EQ.1)
C * CALL GRAF(TIME(1),XINC,TIME(2),-AMAX,YINC,AMAX)
C IF(ICODE.EQ.2.AND.IFLAGS(11).EQ.2.AND.IFLAGS(12).EQ.2)
C * CALL GRAF(TIME(1),XINC,TIME(2),-TIMTEM,YINC,TIMTEM)
C
PLOT FOR FULL TIME RANGE
RMS = 0.0
IPEN = 3
IF(ICODE.EQ.2.AND.IFLAGS(11).EQ.2) GO TO 70
DO 60 I = 1,IBLSZ
IF(ICODE.EQ.2) RMS = RMS + ARRAY(I)**2
IF(ICODE.EQ.1) XVAL = (I - 1.0)*FREQ(1) / XINC
IF(ICODE.EQ.2) XVAL = (I - 1.0) * (1.0 / (FREQ(1)*IBLSZ)) / XINC
IF(ICODE.EQ.1.AND.IFLAGS(6).EQ.2.AND.ARRAY(I).LT.YSTRT)
* ARRAY(I) = YSTRT
IF(ICODE.EQ.2.AND.IFLAGS(12).EQ.2.AND.ARRAY(I).LT.-TIMTEM)
* ARRAY(I) = -TIMTEM
IF(ICODE.EQ.2.AND.IFLAGS(12).EQ.2.AND.ARRAY(I).GT.TIMTEM)
* ARRAY(I) = TIMTEM
YVAL = ARRAY(I) / YINC
IF(ICODE.EQ.1.AND.IFLAGS(6).EQ.2) YVAL = (ARRAY(I) - YSTRT) / YINC
CALL PLOT(XVAL,YVAL,IPEN)
IF(ICODE.EQ.1.AND.I.EQ.N) GO TO 90
IPEN = 2
60 CONTINUE
RMS = SQRT(RMS/IBLSZ)
CALL NUMBER(5.07,-1.65,.07,RMS,0.,3)
CALL SYMBOL(5.77,-1.65,.07,6HK RMS,0.,5)
CALL NUMBER(5.07,-1.8,.07,TMEAN,0.,3)
CALL SYMBOL(5.77,-1.8,.07,6HK MEAN,0.,6)
GO TO 90

CALLS FOR DISSPLA TO PLOT THE FULL TIME RANGE PLOTS

C
RMS = 0.0
IF(ICODE.EQ.2.AND.IFLACS(ll).EQ.2) GO TO 70
DO 60 I = 1,IBLSZ
IF(ICODE.EQ.1) XARRY(1) = (I - 1) * FREQ(1)
IF(ICODE.EQ.2) XARRY(1) = (I - 1) * (1.0 / (FREQ(1)*IBLSZ))
IF(ICODE.EQ.2) RMS = RMS + ARRAY(I)**2
IF(ICODE.EQ.1.AND.IFLAGS(6).EQ.2.AND.ARRAY(I).LT.YSTRT) ARRAY(I) = YSTRT
IF(ICODE.EQ.2.AND.IFLACS(12).EQ.2.AND.ARRAY(I).LT.-TIMTEM) ARRAY(I) = -TIMTEM
IF~ICODE.EQ.2.AND.IFLAGS~l2~.EQ.2.AND.ARRAY~I~.GT.TIMTEM~ ARRAY(I) = TIMTEM
IF(ICODE.EQ.l.AND.1.EQ.N) GO TO 65
CONTINUE
IF(ICODE.EQ.1) CALL CURVE(XARRY,ARRAY,N,O)
IF(ICODE.EQ.2) CALL CURVE(XARRY,ARRAY,IBLSZ,O)
IF(ICODE.EQ.1) GO TO 90
RMS = SQRT(RMS/IBLSZ)
CALL HEIGHT(.07)
CALL REALNO(RMS,3,6.07,.35)
CALL MESSAG(9H K RMS,g,'ABUT','ABUT')
CALL REALNO(TMEAN,3,6.07,.2)
CALL MESSAC(9H K MEAN,g,'ABUT','ABUT')
CALL RESET('HEIGHT')
GO TO 90

C

PARTIAL TIME RANGE

DO 80 I = ICH1,ICH2
RMS = RMS + ARRAY(I)**2
XVAL = (I-ICH1) * (1.0 / (FREQ(1)*IBLSZ)) / XINC
IF(IFLAGS(12).EQ.2.AND.ARRAY(I).LT.-TIMTEM) ARRAY(I) = -TIMTEM
IF(IFLAGS(12).EQ.2.AND.ARRAY(I).GT.TIMTEM) ARRAY(I) = TIMTEM
YVAL = ARRAY(I) / YINC
CALL PLOT(XVAL,YVAL,IPEN)
IPEN = 2
CONTINUE
RMS = SQRT(RMS / (ICH2-ICH1+1))
CALL NUMBER(5.07,-1.65,.07,RMS,0.,3)
CALL SYMBOL(5.77,-1.65,.07,6HK RMS,0.,5)
CALL NUMBER(5.07,-1.8,.07,TMEAN,0.,3)
CALL SYMBOL(5.77,-1.8,.07,6HK MEAN,0.,6)

CALLS FOR DISSPLA TO PLOT THE PARTIAL TIME RANGE PLOTS
DO 80 I = ICH1, ICH2
II = I - ICH1 + 1
XARRY(II) = (I - 1) * (1.0 / (FREQ(1)*IBLSZ))
RMS = RMS + ARRAY(I)**2
IF(IFLAGS(12).EQ.2.AND.ARRAY(I).LT.-TIMTEM) ARRAY(I) = -TIMTEM
IF(IFLAGS(12).EQ.2.AND.ARRAY(I).GT.TIMTEM) ARRAY(I) = TIMTEM
CONTINUE
M = ICH2 - ICH1 + 1
CALL CURVE(XARRY, ARRAY, M, O)
RMS = SQRT(RMS/FLOAT(M))
CALL HEIGHT(.07)
CALL REALNO(RMS, 3, 5.07, .35)
CALL MESSAG(9H K RMS, 9, 'ABUT', 'ABUT')
CALL REALNO(TMEAN, 3, 5.07, .2)
CALL MESSAG(9H K MEAN, 9, 'ABUT', 'ABUT')
CALL RESET('HEIGHT')

*** HEADERS FOR THE PLOTS AND THE UNITS OF TEMPERATURE
90 IF(ICODE.EQ.1) CALL PLOT(-.5,-2.,-3)
   IF(ICODE.EQ.2) CALL PLOT(0.0,-4.,-3)
   IF(ICODE.EQ.2) GO TO 100
   CALL SYMBOL(0.,9.5,.2,30HAVERAGED FREQUENCY DOMAIN DATA,0.,30)
   IF(IFLAGS(6).EQ.1) CALL SYMBOL(.5,6.5,1.9HK**2 / HZ,0.,9)
   IF(IFLAGS(6).EQ.2) CALL SYMBOL(.5,6.5,1.21HO DB REF 1 K**2 / HZ,0.,21)
   * 0.,21)
   IF(IFLAGS(6).EQ.3) CALL SYMBOL(.5,6.5,1.16HRMS K / SQRT(HZ),0.,
   * 16)
   IF(IFLAGS(6).EQ.4) CALL SYMBOL(.5,6.5,1.5HRMS K,0.,5)
   GO TO 110
100 CALL SYMBOL(-.3,9.5,.2,36HCOMPOSIT INSTANTANEOUS TIME WAVEFORM,0.,36)

CALLS TO DISSPLA FOR HEADERS
90 CALL ENDCR(0)
   IF(ICODE.EQ.1) CALL OREL(-.5,-2.25)
   IF(ICODE.EQ.2) CALL OREL(0.0,-2.25)
   CALL AREA2D(7.,10.5)
   CALL HEIGHT(.2)
   IF(ICODE.EQ.2) GO TO 100
   CALL MESSAG(30HAVERAGED FREQUENCY DOMAIN DATA,30,0.,9.5)
   CALL RESET('HEIGHT')
   CALL HEIGHT(.1)
   IF(IFLAGS(6).EQ.1) CALL MESSAG(9HK**2 / HZ,9.,5.6.5)
   IF(IFLAGS(6).EQ.2) CALL MESSAG(21HO DB REF 1 K**2 / HZ,21,
   *.5,6.5)
   IF(IFLAGS(6).EQ.3) CALL MESSAG(16HRMS K / SQRT(HZ),16.,5.6.5)
   IF(IFLAGS(6).EQ.4) CALL MESSAG(5HRMS K,5.,5.6.5)
   CALL RESET('HEIGHT')
   GO TO 110
100 CALL MESSAG(36HCOMPOSIT INSTANTANEOUS TIME WAVEFORM,36,-.3,9.5)
   CALL RESET('HEIGHT')
C— TC PLOT IS DONE FOR E249
110 IF(IFLAGS(3).EQ.1) CALL SYMBOL(1.,9.0,14,9HSMALL T/C,0.,9) E249
IF(IFLAGS(3).EQ.2) CALL SYMBOL(1.,9.0,14,9HLARGE T/C,0.,9) E249
C— COMPENSATED?
   IF(IFLAGS(9).EQ.1) CALL SYMBOL(4.,9.0,14,16HCOMPENSATED DATA,0.,9) E249
   *16
   IF(IFLAGS(9).EQ.2) CALL SYMBOL(4.,9.0,14,19HUN-COMPENSATED DATA,0.,19) E249
C— INSTANTANEOUS DATA
   IF(ICODE.EQ.1) GO TO 120 E249
   CALL SYMBOL(0.5,8.6,.14,33HINSTANTANEOUS DATA, RECORD NUMBER,O., E249
   *33)
   RNUM = NUM + .05 E249
   CALL NUMBER(5.2,8.6,.14,RNUM,O.,O) E249
   GO TO 130 E249
C
C CALLS FOR DISSPLA E249
C— TC PLOT IS DONE FOR E249
C110 IF(IFLAGS(3).EQ.1) CALL MESSAG(9HSMALL T/C,9,1.,9.) E249
C IF(IFLAGS(3).EQ.2) CALL MESSAG(9HLARGE T/C,9,1.,9.) E249
C— COMPENSATED?
C IF(IFLAGS(9).EQ.1) CALL MESSAG(16HCOMPENSATED DATA,16,4.,9.) E249
C IF(IFLAGS(9).EQ.2) CALL MESSAG(19HUN-COMPENSATED DATA,19,4.,9.) E249
C— INSTANTANEOUS DATA
C IF(IFLAGS(8).EQ.1) GO TO 120 E249
C CALL MESSAG(33HINSTANTANEOUS DATA, RECORD NUMBER,33,.5,8.6) E249
C CALL INTNO(NUM,5.2,8.6) E249
C GO TO 130 E249
C
C— AVERAGED DATA
120 CALL SYMBOL(4.5,6.6,.07,20HSTARTING REC NUMBER,0.,20) E249
   CALL SYMBOL(4.5,6.1,.07,19HRECORDS IN AVERAGE,0.,19) E249
   IF(IFLAGS(8).EQ.1) STREC = NRECS(1) + .05 E249
   IF(IFLAGS(8).EQ.2) STREC = NUM + .05 E249
   IF(IFLAGS(8).EQ.1) RECNUM = NRECS(2) + .05 E249
   IF(IFLAGS(8).EQ.2) RECNUM = 1.0 E249
   CALL NUMBER(6.0,6.5,.07,STREC,O.,O) E249
   CALL NUMBER(6.0,6.1,.07,RECNUM,O.,O) E249
130 CALL SYMBOL(0.0,8.0,.07,PLOTIT(1,1),0.,80) E249
   CALL SYMBOL(0.0,7.8,.07,PLOTIT(1,2),0.,80) E249
   CALL SYMBOL(0.0,7.6,.07,PLOTIT(1,3),0.,80) E249
   CALL PLOT(0.,0.,-999) E249
C
C CALLS FOR DISSPLA E249
C
C— AVERAGED DATA
C120 CALL HEIGHT(.07) E249
C CALL MESSAG(20HSTARTING REC NUMBER,20,4.5,6.5) E249
C CALL MESSAG(19HRECORDS IN AVERAGE,19,4.5,6.1) E249
C CALL INTNO(NRECS(1),6.0,6.5) E249
C CALL INTNO(NRECS(2),6.0,6.1) E249
C CALL RESET('HEIGHT') E249
CALL HEIGHT(.07)
CALL MESSAG(PLOTIT(1,1),80,0.0,8.0)
CALL MESSAG(PLOTIT(1,2),80,0.0,7.8)
CALL MESSAG(PLOTIT(1,3),80,0.0,7.6)
CALL RESET('HEIGHT')
CALL ENDPL(0)

RETURN
END

SUBROUTINE PLT2(ARR1,ARR2,ICODE,GS,NUM,TMEAN)

--- IDENTIFICATION ---

* ARR1 - THE FIRST ARRAY TO BE PLOTTED
  — INPUT

* ARR2 - THE SECOND ARRAY TO BE PLOTTED
  — INPUT

* ICODE - CODE OF THE FUNCTION TO BE PLOTTED SO THAT THE APPROPRIATE
  LABELING CAN BE DONE
  1 = COMPENSATION SPECTRUM
  2 = COMPENSATED INSTANTANEOUS SPECTRUM
  — INPUT

* GS - ARRAY CONTAINING THE VALUE OF GAMMA FOR WHICH THE COMPENSATION
  SPECTRUM WAS FOUND
  — INPUT

* NUM - THE RECORD NUMBER BEING PLOTTED (INSTANTANEOUS DATA)
  — INPUT

* TMEAN - THE MEAN DC TEMPERATURE (NEEDED TO PUT ON PLOT)
  — INPUT

--- CALLED FROM TRANGS

--- CALLS - MANY CALCOMP PLOTTER ROUTINES

--- DIMENSION ARR1(2048), ARR2(1025), GS(10)
--- DIMENSION XARRY(2048)
--- COMMON /INPUTS/ IFLAGS(12), TCDATA(4,2), GAS(4), FREQ(4), CHANL(9),
   * IAVDAT(2), IBLSZ, IREC(10), TIME(2), IBSZ, GAMMA, NREC(10), NRECS(2),
   * PLTFRQ, TIMTEM, IDEBUG, IBUC2, ITHRSH
--- COMMON /PLOTTR/ PLOTIT(20,3)
YSTRT = 0.0
ISIZ = IBSZ - 1
N = PLTFRQ / FREQ(1) + 1.5
IF(N.GT.ISIZ) N = ISIZ

C
C*** CONVERSIONS FOR COMPENSATION SPECTRUM GAIN TO DB'S
IF(ICODE.EQ.2) GO TO 20
DO 10 I = 2, ISIZ
ARR1(I) = 20.0 * ALOG10(ARR1(I))
10 CONTINUE

C
C*** FIND MAX AND MIN VALUES OF THE ARRAYS FOR WINDOWING REASONS
C—— FOR FULL TIME RANGE
20 IF(ICODE.EQ.2.AND.IFLAGS(11).EQ.2) GO TO 40
IF(ICODE.EQ.1) NUMST = 2
IF(ICODE.EQ.1) IED'Q = ISIZ
IF(ICODE.EQ.2) NUMST = 1
IF(ICODE.EQ.2) IEDCH = IBSIZ
AR1MIN = ARR1(NUMST)
AR2MIN = ARR2(NUMST)
AR1MAX = ARR1(NUMST)
AR2MAX = ARR2(NUMST)
DO 30 I = 2, IEDCH
IF(ARR1(I).LT.AR1MIN) AR1MIN = ARR1(I)
IF(ARR1(I).GT.AR1MAX) AR1MAX = ARR1(I)
IF(I.GT.N) GO TO 30
IF(ARR2(I).LT.AR2MIN) AR2MIN = ARR2(I)
IF(ARR2(I).GT.AR2MAX) AR2MAX = ARR2(I)
IF(ICODE.EQ.1.AND.I.EQ.N) GO TO 70
30 CONTINUE
GO TO 70
C—— FOR PARTIAL TIME RANGE
40 ICH1 = TIME(1)*IBLSZ*FREQ(1) + 1.49
ICH2 = TIME(2)*IBLSZ*FREQ(1) + 1.49
AR1MIN = ARR1(ICH1)
AR2MIN = ARR2(2)
AR1MAX = ARR1(ICH1)
AR2MAX = ARR2(2)
DO 60 I = 2, IBSIZ
IF(I.LT.ICH1.OR.I.GT.ICH2) GO TO 60
IF(ARR1(I).LT.AR1MIN) AR1MIN = ARR1(I)
IF(ARR1(I).GT.AR1MAX) AR1MAX = ARR1(I)
IF(ICODE.EQ.1.AND.I.EQ.N) GO TO 70
60 CONTINUE
GO TO 70
C
C*** SETTING THE INCREMENTS FOR THE PLOTTED DOMAIN
70 AMAX = ABS(AR1MIN)
IF(AR1MAX.GT.AMAX) AMAX = AR1MAX
RANGE = AR2MAX - AR2MIN
C
IF(ICODE.EQ.1.AND.N.EQ.IBSZ) XINC1 = 400.0
IF (ICODE.EQ.1 .AND. N.NE.IBSZ) XINC1 = (N-1) * FREQ(1) / 5.0
IF (ICODE.EQ.2 .AND. IFLAGS(11).EQ.1) XINC1 = 1.0/(5.0+FREQ(1))
IF (ICODE.EQ.2 .AND. IFLAGS(11).EQ.2) XINC1 = (TIME(2)-TIME(1))/5.0

C

IF (N.EQ.IBSZ) XINC2 = 400.0
IF (N.NE.IBSZ) XINC2 = (N-1) * FREQ(1) / 5.0

C

IF (ICODE.EQ.1 .AND. AR1MIN.LT.-46.) YINC1 = 20.0
IF (ICODE.EQ.1 .AND. AR1MIN.GE.-45.) YINC1 = 10.0
IF (ICODE.EQ.2 .AND. IFLAGS(12).EQ.1) YINC1 = AMAX / 2.
IF (ICODE.EQ.2 .AND. IFLAGS(12).EQ.2) YINC1 = TIMTEM / 2.0

C

IF (ICODE.EQ.1) YINC2 = -25
IF (ICODE.EQ.2 .AND. IFLAGS(1).NE.2) YINC2 = AR2MAX / 4.0
IF (ICODE.EQ.1 .OR. IFLAGS(6).NE.2) GO TO 80

INTN = (AR2MAX + 9.9) / 10.0
IAMAX = INTN * 10
YINC2 = 10.0
IF (RANGE.GT.40.) YINC2 = 20.0
YSTRT = IAMAX - 4.0 * YINC2

C

C*** BEGINNING THE PLOTTING

80 IF (ICODE.EQ.1) CALL PLOT(0.,4.2,-3)
IF (ICODE.EQ.2) CALL PLOT(0.,5,-3)
IF (ICODE.EQ.2) CALL AXIS(0.0,0.0,14HFREQUENCY (HZ),-14,5.5,0.0,0.0,E249)
*XINC2)
IF (ICODE.EQ.1) CALL AXIS(0.0,0.0,14HFREQUENCY (HZ),14,5.5,0.0,0.0,E249)
*XINC2)
IF (ICODE.EQ.1) CALL AXIS(0.0,-4.0,11HPHASE (DEG),11,4.,90.,-100.0,E249)
*YINC2)
IF (ICODE.EQ.2 .AND. IFLAGS(6).NE.2) CALL AXIS(0.0,0.0,11HTEMPERATURE,E249)
*11,4.,90.,0.0,YINC2)
IF (ICODE.EQ.2 .AND. IFLAGS(6).EQ.2) CALL AXIS(0.0.,8HDECIBELS,8,
*4.,90.,YSTRT,YINC2)
CALL PLOT(0.,0.,3)
IPEN = 2
DO 90 I = 2,IBSZ
IF (ICODE.EQ.1 .AND. I.EQ.IBSZ) GO TO 90
IF (I.GT.N) GO TO 100
IF (ICODE.EQ.2 .AND. ARR2(I).LT.YSTRT) ARR2(I) = YSTRT
XVAL = FREQ(1) * (I-1.0) / XINC2
YVAL = ARR2(I) / YINC2
IF (ICODE.EQ.2 .AND. IFLAGS(6).EQ.2) YVAL = (ARR2(I) - YSTRT) / YINC2
CALL PLOT(XVAL,YVAL,IPEN)
90 CONTINUE

100 IF (ICODE.EQ.1) GO TO 110
IF (FLAGS(6).EQ.1) CALL SYMBOL(.6,4.1,.07,9HK**2 / HZ,0.,9)
IF (FLAGS(6).EQ.2) CALL SYMBOL(.5,4.1,.07,210DB REF 1K**2 / HZ)
*0.,21)
IF (FLAGS(6).EQ.3) CALL SYMBOL(.5,4.1,.07,16HRMS K / SQRT(HZ),
*0.,16)
IF (FLAGS(6).EQ.4) CALL SYMBOL(.5,4.1,.07,6HRMS K,0.,6)

C

C EXPECTED CALLS TO DISSPLA TO SET UP PLOTTING

74
C 80 IF (ICODE.EQ.1) CALL PHYSOR(1.,2)
C IF (ICODE.EQ.2) CALL PHYSOR(1.,3)
C CALL AREA2D(5.,4.)
C IF (ICODE.EQ.1) CALL CROSS
C IF (ICODE.EQ.1) CALL XNAME(14HFREQUENCY (HZ),-14)
C IF (ICODE.EQ.2) CALL XNAME(14HFREQUENCY (HZ),14)
C IF (ICODE.EQ.1) CALL YNAME(11PHASE (DEG),11)
C IF (ICODE.EQ.2.AND.IFLACS(6).EQ.2) CALL YNAME(8HDICBELES,8)
C IF (ICODE.EQ.2.AND.IFLACS(6).EQ.2) CALL YNAME(11HTATURATURE,11)
C
C XMX = N * FREQ(1)
C IF (ICODE.EQ.1) CALL GRAF(0.,XINC2,XMX,100.,YINC2,0.0)
C IF (ICODE.EQ.2.AND.IFLACS(6).EQ.2)
C * CALL GRAF(0.,XINC2,XMX,0.0,YINC2,2MAX)
C IF (ICODE.EQ.2.AND.IFLACS(6).EQ.2)
C * CALL GRAF(0.,XINC2,XMX,YSTRY,YINC2,2IAMAX)
C XARRY(1) = 0.0
C ARR2(1) = 0.0
C DO 90 I = 2,IBSZ
C IF (I.CT.N) GO TO 100
C IF (ICODE.EQ.1.AND.I.EQ.IBSZ)
C TO 90
C XARRY(1) = (I - 1.0) * FREQ(1)
C IF (ICODE.EQ.2.AND.IFLACS(11).EQ.1)
C YSTR = -40.
C IF (ICODE.EQ.2.AND.IFLACS(11).EQ.1)
C YSTR = -80.
C IF (ICODE.EQ.2.AND.IFLACS(11).EQ.1)
C YSTR = -100.
C CALL CURVE(XARRY,ARR2,YSTR,0.0)
C IF (ICODE.EQ.1) GO TO 110
C CALL HEIGHT(0.07)
C IF (IFLAGS(6).EQ.1) CALL MESSAG(9K**2/HZ,9,.5,4.1)
C IF (IFLAGS(6).EQ.2) CALL MESSAG(21HO DB REF 1 K**2/HZ,21,.5,4.1)
C *=.5,4.1)
C IF (IFLAGS(6).EQ.3) CALL MESSAG(16HRMS K/SQRT(HZ),16,.5,4.1)
C IF (IFLAGS(6).EQ.4) CALL MESSAG(5HRMS K,.5,.5,4.1)
C CALL RESET('HEIGHT')
C
C 110 IF (ICODE.EQ.1) CALL PLOT(0.,4.9,-3)
C IF (ICODE.EQ.2) CALL PLOT(0.,6.9,-3)
C IF (ICODE.EQ.1) CALL AXIS(0.0,0.0,14HFREQUENCY (HZ),14,.5,5,0.,
C *0.0,XINC1)
C IF (ICODE.EQ.2.AND.IFLACS(11).EQ.1) CALL AXIS(0.0,-2.,10HTIME (SEC))
C *=-10,5.0,0.0,XINC1)
C IF (ICODE.EQ.2.AND.IFLACS(11).EQ.1) CALL AXIS(0.0,-2.,10HTIME (SEC))
C *=-10,5.0,0.0,XINC1)
C IF (YINC1.EQ.10) YST = -40.
C IF (YINC1.EQ.20) YST = -80.
C IF (ICODE.EQ.1) CALL AXIS(0.0,-4.,9HGAIN (DB),9.4,.90.,YST,YINC1)
C IF (ICODE.EQ.2.AND.IFLACS(12).EQ.1) CALL AXIS(0.0,-2.,15HTATURATURE)
C *E (K),15,.4,.90.,-AMAX,YINC1)
C IF (ICODE.EQ.2.AND.IFLACS(12).EQ.2) CALL AXIS(0.0,-2.,15HTATURATURE)
C *E (K),15,.4,.90.,-TIMTEM,YINC1)
CALL ENDRG(0)

CALL OREL(0., 4.7)

CALL OREL(0., 4.9)

CALL AREASD(6., 4.)

IF(ICODE.EQ.1) CALL XNAME(14HFREQUENCY (HZ), -14)

IF(ICODE.EQ.2) CALL XNAME(10HTIME (SEC), 10)

IF(ICODE.EQ.1) CALL YNAME(9HGAIN (DB), 9)

IF(ICODE.EQ.2) CALL YNAME(16HTEMPERATURE (K), 15)

IF(YINC1.EQ.10) YST = -40.

IF(YINC1.EQ.20) YST = -80.

XMX = N * FREQ(1)

CALL GRAF(0.0, XINC1, XMX, YST, YINC1, 0.0)

IF(IFLAGS(11).EQ.1.AND.IFLAGS(12).EQ.1) CALL GRAF(O.O, XINC1, 0.6, -AMAX, YINC1, AMAX)

CALL GRAF(O.O, XINC1, 0.6, -TIMTEM, YINC1, TIMTEM)

CALL GRAF(TIME(1), XINC1, TIME(2), -AMAX, YINC1, AMAX)

CALL GRAF(TIME(1), XINC1, TIME(2), -TIMTEM, YINC1, TIMTEM)

RMS = 0.0

IF(IFLAGS(11).EQ.1.AND.IFLAGS(12).EQ.1) CALL PLOT(0., 0., 3)

IPEN = 2

DO 120 I = IST, IBLSZ

IF(ICODE.EQ.1.AND.I.GT.N) CALL SYMBOL(.1, -4., .1, 19HO DB REF UNITY)

**GAIN, 0., 19)

IF(ICODE.EQ.1.AND.I.GT.N) GO TO 160

IF(ICODE.EQ.2) RMS = RMS + ARR1(I)**2

IF(IFLAGS(12).EQ.2.AND. ARR1(I).GT.TIMTEM) ARR1(I) = TIMTEM

IF(IFLAGS(12).EQ.2.AND. ARR1(I).LT.-TIMTEM) ARR1(I) = -TIMTEM

IF(ICODE.EQ.1) XVAL = FREQ(1) * (I-1.0) / XINC1

IF(ICODE.EQ.2) XVAL = (I-1.0) * (1.0 / (FREQ(1)*IBLSZ)) / XINC1

YVAL = ARR1(I) / YINC1

CALL PLOT(XVAL, YVAL, IPEN)

CONTINUE

RMS = SQRT(RMS/IBLSZ)

CALL NUMBER(5.07, -1.66, .07, RMS, 0., 3)

CALL SYMBOL(6.77, -1.66, .07, RMS, 0., 5)

CALL NUMBER(5.07, -1.8, .07, TIME(1), 3)

CALL SYMBOL(6.77, -1.8, .07, TIME(2), 5)

GO TO 160

CALLS TO DISSPLA

CALLS TO DISSPLA

--- PLOT FOR FULL TIME RANGE ---

RMS = 0.0

IF(IFLAGS(11).EQ.1.AND.IFLAGS(12).EQ.1) CALL PLOT(0., 0., 3)

IPEN = 2

DO 120 I = IST, IBLSZ

IF(ICODE.EQ.1.AND.I.GT.N) CALL SYMBOL(.1, -4., .1, 19HO DB REF UNITY)

**GAIN, 0., 19)

IF(ICODE.EQ.1.AND.I.GT.N) GO TO 160

IF(IFLAGS(12).EQ.2.AND. ARR1(I).GT.TIMTEM) ARR1(I) = TIMTEM

IF(IFLAGS(12).EQ.2.AND. ARR1(I).LT.-TIMTEM) ARR1(I) = -TIMTEM

IF(ICODE.EQ.1) XVAL = FREQ(1) * (1-1.0) / XINC1

IF(ICODE.EQ.2) XVAL = (1-1.0) * (1.0 / (FREQ(1)*IBLSZ)) / XINC1

YVAL = ARR1(I) / YINC1

CALL PLOT(XVAL, YVAL, IPEN)

CONTINUE

RMS = SQRT(RMS/IBLSZ)

CALL NUMBER(5.07, -1.65, .07, RMS, 0., 3)

CALL SYMBOL(6.77, -1.65, .07, RMS, 0., 5)

CALL NUMBER(5.07, -1.8, .07, TIME(1), 3)

CALL SYMBOL(6.77, -1.8, .07, TIME(2), 5)

GO TO 160

--- PLOT FOR FULL TIME RANGE ---

RMS = 0.0
IF(IFLAGS(11).EQ.2.AND.ICODE.EQ.2) GO TO 130

IF(ICODE.EQ.2) IST = 1

IF(ICODE.EQ.1) IST = 2

IF(ICODE.EQ.1) XARRY(1) = 0.0

IF(ICODE.EQ.1) ARR1(I) = 0.0

DO 120 I = IST, IBLSZ

IF(ICODE.EQ.1.AND.I.GT.N) GO TO 125

IF(ICODE.EQ.2) RMS = RMS + ARR1(I)**2

IF(ICODE.EQ.1) XARRY(I) = (1-1.0) * (1.0 / (FREQ(I)*IBLSZ))

IF(ICODE.EQ.2.AND.IFLAGS.EQ.2.AND.ARR1.I.GT.TIMTEM) ARR1(I) = TIMTEM

IF(ICODE.EQ.2.AND.IFLAGS.EQ.2.AND.ARR1.I.LT.-TIMTEM) ARR1(I) = -TIMTEM

C120 CONTINUE

C125 IF(ICODE.EQ.1) CALL CURVE(XARRY,ARR1,N,O)

IF(ICODE.EQ.2) CALL CURVE(XARRY,ARR1,IBLSZ,0)

IF(ICODE.EQ.1) CALL MESSAG(1900 DB REF UNI GAIN,19,1,0.0)

IF(ICODE.EQ.1) GO TO 150

CALL HEIGHT(.07)

RMS = SQRT(RMS/IBLSZ)

CALL REALNO(RMS,3,.07,35)

CALL MESSAG(5HK RMS,5,.57,36)

CALL REALNO(TMEAN,3,.07,2)

CALL MESSAG(6HK MEAN,6,.57,2)

CALL RESET('HEIGHT')

GO TO 150

C--- PARTIAL TIME RANGE

130 IPEN = 3

DO 140 I = ICH1, ICH2

RMS = RMS + ARR1(I)**2

IF(IFLAGS(12).EQ.2.AND.ARR1(I).GT.TIMTEM) ARR1(I) = TIMTEM

IF(IFLAGS(12).EQ.2.AND.ARR1(I).LT.-TIMTEM) ARR1(I) = -TIMTEM

XVAL = (I-ICH1) * (1.0 / (FREQ(I)*IBLSZ)) / XINCI

YVAL = ARR1(I) / YINCI

CALL PLOT(XVAL,YVAL,IPEN)

IPEN = 2

140 CONTINUE

RMS = SQRT(RMS / (ICH2-ICH1+1))

CALL NUMBER(5.07,-1.65,.07,RMS,0.3)

CALL SYMBOL(5.77,-1.66,.07,5HK RMS,0.5)

CALL NUMBER(5.07,-1.8,.07,TMEAN,0.3)

CALL SYMBOL(5.77,-1.8,.07,6HK MEAN,0.6)

C--- DISSPLA CALLS

C--- PARTIAL TIME RANGE

C130 DO 140 I = ICH1, ICH2

RMS = RMS + ARR1(I)**2

XARRY(I) = (I-1.0) * (1.0 / (FREQ(I)*IBLSZ))

IF(IFLAGS(12).EQ.2.AND.ARR1(I).GT.TIMTEM) ARR1(I) = TIMTEM

IF(IFLAGS(12).EQ.2.AND.ARR1(I).LT.-TIMTEM) ARR1(I) = -TIMTEM

C140 CONTINUE
C M = ICH2 - ICH1 * 1  
C CALL CURVE(XARR,ARR1,M,0)  
C CALL HEIGHT(.07)  
C RMS = SQRT(RMS / M)  
C CALL REALNO(RMS,3.5,07,.35)  
C CALL MESSAG(9HK RMS,5.6,77,.35)  
C CALL REALNO(TMEAN,3.5,07,.2)  
C CALL MESSAG(9HK MEAN,6,5.77,.2)  
C CALL RESET('HEIGHT')  
C  
C*** HEADERS FOR THE PlOTS  
150 IF(ICODE.EQ.1) CALL PLOT(O.,-9.1,-3)  
IF(ICODE.EQ.2) CALL PLOT(O.,-7.4,-3)  
IF(ICODE.EQ.1) CALL SYMBOL(0.8,10.4,.1,21HCOMPENSATION SPECTRUM,  
* 0.,21)  
IF(ICODE.EQ.2) CALL SYMBOL(1.2,10.4,.1,18INSTANTANEOUS DATA,0.,  
*18)  
C TC PLOT IS FOR  
IF(IFLAGS(3).EQ.1) CALL SYMBOL(3.3,10.4,.07,9HSML T/C,0.,9)  
IF(IFLAGS(3).EQ.2) CALL SYMBOL(3.3,10.4,.07,9LARGE T/C,0.,9)  
C MATERIAL  
IF(IFLAGS(2).EQ.1) CALL SYMBOL(4.5,10.4,.07,10HPT / 6% RH,0.,10)  
IF(IFLAGS(2).EQ.2) CALL SYMBOL(4.5,10.4,.07,11HPT / 30% RH,0.,11)  
IF(IFLAGS(2).EQ.3) CALL SYMBOL(4.5,10.4,.07,7HCR / AL,0.,7)  
C  
C DISPLA CALLS  
C  
C*** HEADERS FOR THE PlOTS  
C150 CALL ENDGR(0)  
C IF(ICODE.EQ.1) CALL OREL(O.,-4.9)  
C IF(ICODE.EQ.2) CALL OREL(O.,-5.4)  
C CALL AREA2D(7.,10.6)  
C CALL HEIGHT(.1)  
C IF(ICODE.EQ.1) CALL MESSAG(21HCOMPENSATION SPECTRUM,21,1.2,10.4)  
C IF(ICODE.EQ.2) CALL MESSAG(18INSTANTANEOUS DATA,18,1.2,10.4)  
C CALL RESET('HEIGHT')  
C TC PLOT IS FOR  
C CALL HEIGHT(.07)  
C IF(IFLAGS(3).EQ.1) CALL MESSAG(9HSML T/C,9,3.3,10.4)  
C IF(IFLAGS(3).EQ.2) CALL MESSAG(9LARGE T/C,9,3.3,10.4)  
C MATERIAL  
C IF(IFLAGS(2).EQ.1) CALL MESSAG(10HPT / 6% RH,10,4.5,10.4)  
C IF(IFLAGS(2).EQ.2) CALL MESSAG(11HPT / 30% RH,11,4.5,10.4)  
C IF(IFLAGS(2).EQ.3) CALL MESSAG(7HCR / AL,7,4.5,10.4)  
C  
C GAMMA VALUE FOR COMPENSATION SPECTRUM  
IF(ICODE.EQ.2) GO TO 160  
CALL SYMBOL(0.0,10.25,.07,8HGAMMA = .0.,8)  
GMAMET = GS(10) * .168279  
CALL NUMBER(0.5,10.25,.07,GMAMET,0.)  
C TC DIMENSIONS FOR COMPENSATION SPECTRUM  
CALL SYMBOL(2.0,10.25,.07,3HL1 ,0.,3)  
CALL SYMBOL(3.1,10.25,.07,3HL2 ,0.,3)  
C
CALL SYMBOL(4.2, 10.25, .07, 3HD1, 0., 3)
CALL SYMBOL(5.3, 10.25, .07, 3HD2, 0., 3)
IF(IFLAGS(3).EQ.1) ICOL = 2
IF(IFLAGS(3).EQ.2) ICOL = 1
SULN = TCDATA(1, ICOL) / .032808 + .0000001
SMLN = TCDATA(2, ICOL) / .032808 + .0000001
SUDI = TCDATA(3, ICOL) / .032808 + .0000001
SMDI = TCDATA(4, ICOL) / .032808 + .0000001
CALL NUMBER(2.21, 10.25, .07, SULN, 0., 7)
CALL NUMBER(3.31, 10.25, .07, SMLN, 0., 7)
CALL NUMBER(4.41, 10.25, .07, SUDI, 0., 7)
CALL NUMBER(5.61, 10.25, .07, SMDI, 0., 7)
GO TO 170
C- COMPENSATED?
160 IF(IFLAGS(9).EQ.2) CALL SYMBOL(0., 10.25, .07, 19HUN-COMPENSATED DATA, 0., 19)
*0., 19)
IF(IFLAGS(9).EQ.1) CALL SYMBOL(0., 10.25, .07, 16HCOMPENSATED DATA, 0., 16)
INSTANTANEOUS DATA
CALL SYMBOL(3.6, 10.25, .07, 13HRECORD NUMBER, 0., 13)
RNUM = NUM + .05
CALL NUMBER(4.5, 10.25, .07, RNUM, 0., 0)
170 CALL SYMBOL(0.0, 9.95, .07, PLOTIT(1,1), 0., 80)
CALL SYMBOL(0.0, 9.75, .07, PLOTIT(1,2), 0., 80)
CALL SYMBOL(0.0, 9.55, .07, PLOTIT(1,3), 0., 80)
CALL PLOT(0., 0., -999)
C- DISPLA CALLS
C- GAMMA VALUE FOR COMPENSATION SPECTRUM
C IF(ICODE.EQ.2) GO TO 160
C CALL MESSAC(8HGAMMA = 8, 0.0, 10.25)
C GMAMET = GS(10) * .168279
C CALL REALNO(GMAMET, -5, 'ABUT', 'ABUT')
C- TC DIMENSIONS FOR COMPENSATION SPECTRUM
C CALL MESSAG(3HL1, 3, 2.0, 10.25)
C CALL MESSAG(3HL2, 3, 3.1, 10.25)
C CALL MESSAG(3HD1, 3, 4.2, 10.25)
C CALL MESSAG(3HD2, 3, 6.3, 10.25)
C IF(IFLAGS(3).EQ.1) ICOL = 2
C IF(IFLAGS(3).EQ.2) ICOL = 1
C SULN = TCDATA(1, ICOL) / .032808 + .0000001
C SMLN = TCDATA(2, ICOL) / .032808 + .0000001
C SUDI = TCDATA(3, ICOL) / .032808 + .0000001
C SMDI = TCDATA(4, ICOL) / .032808 + .0000001
C CALL REALNO(SULN, 7, 2.21, 10.25)
C CALL REALNO(SMLN, 7, 3.31, 10.25)
C CALL REALNO(SUDI, 7, 4.41, 10.25)
C CALL REALNO(SMDI, 7, 5.61, 10.25)
C GO TO 170
C- COMPENSATED?
C160 IF(IFLAGS(9).EQ.2) CALL MESSAG(19HUN-COMPENSATED DATA, 19, 0., 10.25)
C IF(IFLAGS(9).EQ.1) CALL MESSAG(16HCOMPENSATED DATA, 16, 0., 10.25)
C- INSTANTANEOUS DATA
C CALL MESSAG(13HRECORD NUMBER,13,3.5,10.25) E249
C CALL INTNO(NUM,4.5,10.26) E249
C170 CALL MESSAG(PLOITI(1,1),80.0.0,9.95) E249
C CALL MESSAG(PLOITI(1,2),80.0.0,9.76) E249
C CALL MESSAG(PLOITI(1,3),80.0.0,9.55) E249
C CALL RESET('HEIGHT') E249
C CALL RESET('CROSS') E249
C CALL ENDPL(0) E249
C
IF(ICODE.EQ.2) GO TO 190 E249
DO 180 I = 1,ISIZ E249
ARRI(I) = 10.0 ** (ARRI(I)/20.) E249
180 CONTINUE E249
190 RETURN E249
END E249
C
C SUBROUTINE POWER(DATA3,DATA10,DATADC,XX,YY,XY,WINDO) E249
C
C******************************************************************************E249
C POWER EVALUATES THE AUTO POWER SPECTRUMS OF THE 3MIL AND 10MIL DATA *E249
C AND THE CROSS POWER SPECTRUM OF THE 3MIL VS 10MIL *E249
C******************************************************************************E249
C
C --- IDENTIFICATION --- E249
C
C** DATA3 - ARRAY FOR THE 3MIL TEST DATA READ IN INTEST E249
C   - OUTPUT E249
C
C** DATA10 - ARRAY FOR THE 10MIL TEST DATA READ IN INTEST E249
C   - OUTPUT E249
C
C** DATADC - ARRAY FOR THE DC CHANNEL TEST DATA READ IN INTEST E249
C   - OUTPUT E249
C
C** XX - ARRAY CONTAINING THE INPUT AUTO POWER SPECTRUM G 3-3 E249
C   - OUTPUT E249
C
C** YY - ARRAY CONTAINING THE OUTPUT AUTO POWER SPECTRUM G 10-10 E249
C   - OUTPUT E249
C
C** XY - ARRAY CONTAINING THE CROSS POWER SPECTRUM G 3-10 E249
C   - OUTPUT E249
C
C** WINDO - ARRAY CONTAINING THE P301 WINDOW E249
C   - OUTPUT E249
C
C CALLED BY MAIN PROGRAM E249
C CALLS - INTEST: INPUTS THE DIGITIZED TEST DATA E249
C SCALER: SCALES THE DATA TO K E249
C WINDOW: APPLYS THE P301 TO INPUT DATA E249
C FFT: ROUTINE TO PERFORM FOURIER TRANSFORM E249
C OF A COMPLEX VALUED SEQUENCE E249

80
FILES USED:

15 - WRITES OUT THE FOURIES TRANSFORM OF THE 3 MIL THERMOCOUPLE TO THIS FILE.

16 - WRITES OUT THE FOURIES TRANSFORM OF THE 10 MIL THERMOCOUPLE TO THIS FILE.

 complex CDATA3(2048), CDAT10(2048)
 dimension XX(1025), YY(1025), XY(1025, 2), WINDO(2048), DATA3(2048),
 * DATA10(2048), DATADC(2048),
 * ST3(1025), CT3(1025), ST10(1025), CT10(1025)
 common /inputs/ IFLAGS(12), TCDATA(4, 2), GAS(4), FREQ(4), CHANL(9),
 * IAVDAT(2), IBSZ, IREC(10), TIME(2), IBSZ, GAMMA, NREC(10), NRECS(2),
 * PLTFRQ, TIMTEM, IDEBUG, IBUG2, ITHRSH
 common /datas/ C(45), ICF(11, 9)

*** INITIALIZE DATA

 rewind 13
 rewind 14
 rewind 15
 rewind 16
 KOUNT = 0
 IFLAG = 0
 do 10 J = 1, IBSZ
 XX(J) = 0.0
 YY(J) = 0.0
 XY(J, 1) = 0.0
 XY(J, 2) = 0.0
 10 continue

*** LOOP TO READ IN DIGITIZED TEST DATA (MAX OF 400 REC)

 do 60 I = 1, 400
 call INTEST(DATA3, DATA10, DATADC, IFLAG)

*** END OF DATA?

 if(IFLAG.EQ.1) go to 60
 if(I.LT.IAVDAT(1)) go to 50
 KOUNT = KOUNT + 1

*** CONVERT DATA FROM MV TO MV-PK AND THEN TO DEG-F

 call SCALER(DATA3, DATA10, DATADC)

*** APPLY P301 WINDOW TO TEST DATA

 call WINDOW(BINDO, DATA3)
 call WINDOW(WINDO, DATA10)

*** PERFORM FAST FOURIER TRANSFORMS ON DATA

 do 20 J = 1, IBSZ
 CDATA3(J) = CMPLX(DATA3(J), 0.0)
 CDAT10(J) = CMPLX(DATA10(J), 0.0)
 20 continue

 call FFT(1, IBSZ, CDATA3)
CALL FFT(1,IBLSZ,CDAT10)
DO 30 J = 1,IBSZ
CDATA3(J) = CDATA3(J)
CDAT10(J) = CDAT10(J)
CT3(J) = REAL(CDATA3(J)) * 2.0
ST3(J) = AIMAG(CDATA3(J)) * 2.0
CT10(J) = REAL(CDAT10(J)) * 2.0
ST10(J) = AIMAG(CDAT10(J)) * 2.0
30 CONTINUE
C
C*** KEEP RUNNING SUMS OF AUTO AND CROSS POWER SPECTRUMS
DO 40 J = 1,IBSZ
XX(J) = XX(J) + CT3(J)**2 + ST3(J)**2
YY(J) = YY(J) + CT10(J)**2 + ST10(J)**2
XY(J,1) = XY(J,1) + CT3(J)*CT10(J) + ST3(J)*ST10(J)
XY(J,2) = XY(J,2) + ST3(J)*CT10(J) - CT3(J)*ST10(J)
40 CONTINUE
C
C*** WRITE OUT FOURIER TRANSFORMS ON DISK FILES FOR TEMPORARY STORAGE
WRITE(16) (CT3(J),J=1,IBSZ),(ST3(J),J=1,IBSZ)
WRITE(16) (CT10(J),J=1,IBSZ),(ST10(J),J=1,IBSZ)
IF(KOUNT.EQ.IAVDAT(2)) GO TO 60
60 CONTINUE
C
C*** AVERAGE OUT THE POWER SPECTRUMS
DO 70 I = 1,IBSZ
XX(I) = XX(I) / KOUNT
YY(I) = YY(I) / KOUNT
XY(I,1) = XY(I,1) / KOUNT
XY(I,2) = XY(I,2) / KOUNT
70 CONTINUE
RETURN
END
C
C SUBROUTINE PRNTIN (IIN, IOUT)
C*****************************************************************************
C THIS ROUTINE PRINTS THE CARD IMAGE OF INPUT DATA SETS
C*****************************************************************************
C — IDENTIFICATION —
C
C** IIN — THE INPUT FILE TO BE PRINTED
C — INPUT
C
C** IOUT — THE OUTPUT FILE TO WHICH TO WRITE THE DATA
C — INPUT
C
C CALLED BY INPUT SUBROUTINE
C
C FILES USED:
C THE FILES DESCRIBED ABOVE, IN THIS CASE 5 AND 6
C
C************************************************************************** E249
C DIMENSION ARRAY(200) E249
DATA BLANK /4H / E249
C 10 KOUNT = 0 E249
50 DO 100 I=1,200 E249
100 ARRAY(I) = BLANK E249
IFLAG = 0 E249
KOUNT = KOUNT + 1 E249
READ (IIIN, 125 ,END=150) ARRAY E249
IFLAG = 1 E249
150 IF (KOUNT .GT. 1) GO TO 250 E249
WRITE (IOUT, 175) E249
WRITE (IOUT, 180) E249
WRITE (IOUT, 200) E249
250 IF (IFLAG .EQ. 0) GO TO 400 E249
WRITE (IOUT, 300) ARRAY E249
IF (KOUNT .NE. 4) GO TO 60 E249
WRITE (IOUT, 200) E249
WRITE (IOUT, 360) E249
GO TO 10 E249
400 WRITE (IOUT, 300) ARRAY E249
REWIND IIN E249
WRITE (IOUT, 350) E249
RETURN E249
125 FORMAT ( 20A4 ) E249
175 FORMAT (14HlINPUT LISTING ) E249
180 FORMAT ( 35X, 11HCARD COLUMN ) E249
200 FORMAT (/ 10X, 40H111111111222222222333333333333334444444444444444, E249
* 31H5555555555555555555555555555555555555555555555555555555555555, E249
* 50H123456789012345678901234567890123456789012345678901234567890, E249
* 20H123456789012345678901234567890 ) E249
300 FORMAT ( 1X, 20A4 ) E249
350 FORMAT (35X,11HCARD COLUMN ) E249
END E249
C C SUBROUTINE PSDFN(COMP,WINDO,ALSS) E249
C C************************************************************************** E249
C PSDFN CREATES THE POWER SPECTRAL DENSITY FUNCTION FOR PLOTTING * E249
C************************************************************************** E249
C IDENTIFICATION — — — — — — — — — — — — — — — — — — — — — — — — — — — E249
C COMP - THE COMPENSATION SPECTRUM E249
C — INPUT E249
C WINDO - P301 WINDOW, NEEDED TO FIND AREA OF THE LINE SHAPE SQUARED E249
C — INPUT E249
C ALSS - AREA LINE SHAPE SQUARED E249
C                 ** OUTPUT ** E249
C CALLED BY MAIN PROGRAM E249
C CALLS PLT1: THIS SUBPROGRAM PLOTS THE INPUT ARRAY E249
C GET : ACCESSES THE FFT DATA AND FORMS THE APPROPRIATE E249
C AUTO POWER SPECTRUM E249
C
C FILES USED: E249
C 6 - WRITES OUT ALL FUNCTIONS EVALUATED IF IBUG2 = 1 E249
C
C*****************************************************************************
C
C DIMENSION WW(1025),COMP(1024,2),PSD(1025),WINDO(2048),TEMP(1025) E249
C COMMON /INPUTS/ IFLAGS(12),TCDATA(4,2),GAS(4),FREQ(4),CHANL(9), E249
C                   IAVDAT(2),IBLSZ,IREC(10),TIME(2),IBSZ,GAMMA,NREC(10),NRECS(2), E249
C                   PLTFRQ,TIMTEM,IDEBUG,IBUG2,ITHRSH E249
C
C ISIZ = IBSZ - 1 E249
KOUNT = 0 E249

C*** FIND THE AREA OF THE LINE SHAPE SQUARED (ALSO INITIALIZE TEMP) E249
ALSS = 0.0 E249
DO 10 I = 1,IBLSZ E249
ALSS = ALSS + WINDO(I)**2 E249
IF(I.GT.IBSZ) GO TO 10 E249
TEMP(I) = 0.0 E249
PSD(I) = 0.0 E249
10 CONTINUE E249
ALSS = ALSS / IBLSZ E249
IF(IFLAGS(7).EQ.2) RETURN E249

C*** ACCESS THE FFT DATA AND FIND THE AUTO POWER SPECTRUM E249
REWIND 15 E249
REWIND 16 E249
ISTP = IAVDAT(2) E249
DO 110 I = 1,ISTP E249
CALL GET(WW) E249
NUM = IAVDAT(1) + (I-1) E249
IF(IFLAGS(8).EQ.1) GO TO 30 E249

C*** INSTANTANEOUS PLOTS: CHECK TO SEE IF THIS RECORD IS DESIRED E249
DO 20 J = 1,10 E249
IF(NUM.EQ.NREC(J)) GO TO 40 E249
20 CONTINUE E249
GO TO 110 E249

C*** PSD IS TO BE AVERAGED E249
30 IF(NUM.LT.NRECS(1)) GO TO 110 E249
KOUNT = KOUNT + 1 E249
IF(KOUNT.GT.NRECS(2)) GO TO 112 E249

C*** RECORD IS WANTED E249
40 IF(IFLAGS(8).EQ.1) GO TO 90 E249
IF(IFLAGS(0).EQ.2) GO TO 60

C

C* COMPENSATED
DO 60 J = 2, IISIZ
WW(J) = (WW(J)/(FREQ(1)*ALSS))/(COMP(J,1)**2 + COMP(J,2)**2)
IF(IFLAGS(G).EQ.1) TEMP(J) = WW(J)
IF(IFLAGS(G).EQ.2) TEMP(J) = 10. * ALOG10(WW(J))
IF(IFLAGS(G).EQ.3) TEMP(J) = SQRT(WW(J))
IF(IFLAGS(G).EQ.4) TEMP(J) = SQRT(WW(J)*FREQ(1)*ALSS)
60 CONTINUE

GO TO 80

C

C* UNCOMPENSATED
DO 70 J = 2, IBSZ
WW(J) = WW(J)/(FREQ(1)*ALSS)
IF(IFLAGS(G).EQ.1) TEMP(J) = WW(J)
IF(IFLAGS(G).EQ.2) TEMP(J) = 10. * ALOG10(WW(J))
IF(IFLAGS(G).EQ.3) TEMP(J) = SQRT(WW(J))
IF(IFLAGS(G).EQ.4) TEMP(J) = SQRT(WW(J)*FREQ(1)*ALSS)
70 CONTINUE

IF(IFLAGS(7).EQ.1) CALL PLT1(TMP,1,N,0.0)
IF(IBUG2.EQ.1) WRITE(6,180) N
ISTOP = IBSZ/4
DO 85 J = 1, ISTOP
JJ = J + ISTOP
JJJ = JJ + ISTOP
JJJJ = JJJ + ISTOP
IF(IBUG2.EQ.1) WRITE(6,200) J,TEMP(J), JJ,TEMP(JJ), JJJ,TEMP(JJJ), JJJJ,TEMP(JJJJ)
85 CONTINUE

GO TO 110

C

C* SUM IF AVERAGED
DO 100 J = 2, IBSZ
TEMP(J) = TEMP(J) + WW(J)
100 CONTINUE

110 CONTINUE

C

C*** LOOP IS FINISHED, IF DATA WAS INSTANTANEOUS YOU ARE DONE.
IF(IFLAGS(8).EQ.2) RETURN

DO 115 I = 2, IBSZ
TEMP(I) = TEMP(I)/NRECS(2)
115 CONTINUE

IF(IFLAGS(9).EQ.2) GO TO 140

C

C* COMPENSATED
DO 130 J = 2, IISIZ
TEMP(J) = (TEMP(J)/(FREQ(1)*ALSS))/(COMP(J,1)**2 + COMP(J,2)**2)
IF(IFLAGS(G).EQ.1) PSD(J) = TEMP(J)
IF(IFLAGS(G).EQ.2) PSD(J) = 10. * ALOG10(TEMP(J))
IF(IFLAGS(G).EQ.3) PSD(J) = SQRT(TEMP(J))
IF(IFLAGS(G).EQ.4) PSD(J) = SQRT(TEMP(J)*FREQ(1)*ALSS)
130 CONTINUE
GO TO 160

C
C* UNCOMPENSATED
140 DO 150 J = 2,IBSZ
   TEMP(J) = TEMP(J) / (FREQ(1)*ALSS)
   IF(I_FLAGS(G).EQ.1) PSD(J) = TEMP(J)
   IF(I_FLAGS(G).EQ.2) PSD(J) = 10. * ALOG10(TEMP(J))
   IF(I_FLAGS(G).EQ.3) PSD(J) = SQRT(TEMP(J))
   IF(I_FLAGS(G).EQ.4) PSD(J) = SQRT(TEMP(J)*FREQ(1)*ALSS)
150 CONTINUE

160 IF(I_FLAGS(7).EQ.1) CALL PLT1(PSD,1,0,0,0)
   IF(IBUG2.EQ.1) WRITE(6,190) NRECS(2)
   ISTOP = IBSZ/4
   DO 176 J = 1,ISTOP
     JJ = J + ISTOP
     JJJ = JJ + ISTOP
     JJJJ = JJJ + ISTOP
     IF(IBUG2.EQ.1) WRITE(6,200) J,PSD(J),JJ,PSD(JJ),JJJ,PSD(JJJ),JJJJ,PSD(JJJJ)
176 CONTINUE
RETURN

C
C 180 FORMAT('1', 'THE FREQUENCY DOMAIN DATA FOR RECORD NUMBER ',I3)
190 FORMAT('1', 'THE FREQUENCY DOMAIN DATA FOR USING ',I3,2X,'RECORDS IN THE AVERAGING')
200 FORMAT( ' ',4(2X,'I4,3X,E13.7,6X))
END

C
C SUBROUTINE SCALER(DATA3,DATADC)
C
C**************************************************************************************************************************
C SCALER CONVERTS THE DATA TO DEG FAHRENHEIT
C**************************************************************************************************************************
C
C — IDENTIFICATION —
C
C** DATA3 - TEST DATA FOR 3MIL TC READ IN INTEST, INPUT TO SCALE, AND CONTAINING THE SCALED DATA AS OUTPUT FROM THE ROUTINE
C** DATA10 - TEST DATA FOR 10MIL TC READ IN INTEST, INPUT TO SCALE, AND CONTAINING THE SCALED DATA AS OUTPUT FROM THE ROUTINE
C** DATADC - TEST DATA FOR DC CHANNEL READ IN INTEST, INPUT TO SCALE, AND CONTAINING THE SCALED DATA AS OUTPUT FROM THE ROUTINE
C
C CALLED BY POWER SUBPROGRAM
C CALLS - TCALC: ACTUALLY CALCULATES THE TEMPERATURE
C
C FILES USED:
C 13 - WRITES OUT THE SCALED DIGITIZED 3 MIL DATA TO THIS FILE.
C
C 14 - WRITES OUT THE SCALED DIGITIZED 10 MIL DATA TO THIS FILE.
C
C — IDIOSYNCRACIES —
C
C SCALER REMOVES AMPLIFIER DC OFFSET AND SCALES THE DATA PRIOR TO
C LINEARIZATION. TWO RECORDS ARE AC DATA (LARGE & SMALL WIRE T/C)
C AND THE THIRD IS THE DC CHANNEL. AFTER REMOVAL OF DC OFFSET AND
C SCALING, THE PROGRAM ADDS THE DC TO THE AC, CONVERTS IT TO TEMP-
C ERATURE AND THEN REMOVES THE DC, LEAVING PEAK TEMPERATURE. THIS
C IS DONE TO BOTH AC CHANNELS.
C THE SCALED DATA IS WRITTEN ONTO DISK FILE 17 FOR FUTURE ACCESS.
C
C******************************************************************************************************************************
C
C DIMENSION DATA3(2048),DATA10(2048),DATADC(2048)
COMMON /INPUTS/ IFLACS(l2),TCDATA(4,2),GAS(4),FREQ(4),CHANL(9),
* IAVDAT(2),IBLSZ,IREC(10),TIME(2),IBSZ,GAMMA,NREC(10),NRECS(2),
* PLTFREQ,TIMTEM, IDEBUG, IBUG2, ITHRSH
COMMON /DATAS/ CF45),TCF(11,9)
C
C*** SCALE THE DATA
TMEAN = 0.0
DO 10 I=1,IBLSZ
D3M = (DATA3(I)-CHANL(3)) * CHANL(2) * 1000. / CHANL(1)
D1OM = (DATA10(I)-CHANL(6)) * CHANL(5) * 1000. / CHANL(4)
DDC = (DATADC(I)-CHANL(9)) * CHANL(8) * 1000. / CHANL(7)
C
C***CALCULATE DC TEMPERATURE
T = DDC
CALL TCALC(T)
TDC = T
TMEAN = TMEAN + TDC
C
C*** ADD THE DC TO THE AC'S
D3MDDC = D3M + DDC
D1OMDC = D1OM + DDC
C
C*** CALCULATE TEMPERATURE
T = D3MDDC
CALL TCALC(T)
T3MDDC = T
T = D1OMDC
CALL TCALC(T)
T10MDC = T
C
C*** REMOVE THE DC
T3M=T3MDDC-TDC
T10M=T10MDC-TDC
C
C*** PEAK TEMPERATURE
DATA3(I) = T3M * .5./9.
DATA10(I) = T10M * .5./9.
DATA D(I) = TDC
CONTINUE
TMEAN = TMEAN / IBLSZ
TMEAN = 6.0*(TMEAN+40.)/9.0 + 233.16
WRITE(13) (DATA3(I),I=1,IBLSZ),TMEAN
WRITE(14) (DATA10(I),I=1,IBLSZ),TMEAN
RETURN
END

SUBROUTINE SPCY(FRQ,A)

"SPCY" IS A SUBPROGRAM THAT DETERMINES THE SAMPLING FREQUENCY AS A FUNCTION OF THE INPUT FREQUENCY.

IDENTIFICATION
FRQ - FREQUENCY FOUND IN TRANGS, NEEDED HERE FOR COMPUTATIONS
A - ARRAY WHICH CONTAINS THE SAMPLING FREQUENCY
CALLED BY TRFP SUBPROGRAM
TRFM SUBPROGRAM

DIMENSION A(2)

F = FRQ +0.00001
P=1./F
T1=P/(.0005 * 4)
I1 = T1
T1 = I1
A(1)=(T1+1.)*4.
IF(A(1).LT.128.0) A(1) = 128.0
RETURN
END

SUBROUTINE TCALC(T)

"TCALC" IS A SUBPROGRAM THAT CALCULATES TEMPERATURE FROM THE COEFFICIENTS IN THE ARRAY "TCF" AND SPECIFIED THE TC CODE FLAG

IDENTIFICATION
C** I - VARIABLE CONTAINING THE SCALED DATA TO BE INPUT TO Tcalc.
C WILL CONTAIN THE TEMPERATURE DERIVED IN THIS ROUTINE.
C — INPUT AND OUTPUT
C CALLED BY SCALER SUBPROGRAM
C
C********************************************************************E249
C
COMMON /INPUTS/ IFLACS(12),TCDATA(4,2),GAS(4),FREQ(4),CHANL(9),
* IAVIDAT2,IPLS,IREC(10),TIME(2),IBSZ,GAMMA,NREC(10),NRECS(2),
* PLTFQ,TIMTEM,IBU2I2,ITHRSH
COMMON /DATAS/ C(46),TCF(11,9)
C
C*** DIFFERENT INCREMENTS FOR THE DIFFERENT MATERIAL CODES
IF(IFLAGS(2).EQ.3) GO TO 10
IF(IFLAGS(2).EQ.2.OR.IFLACS(2).EQ.1) GO TO 20
10 IRL=1
IRH=IRL+2
GO TO 30
20 IRL=7
IRH=IRL+3
30 DO 40 IR=IRL,IRH
IF(TCF(IR,1).GT.T) GO TO 60
40 CONTINUE
XN=IRL
XR=IRH
C
C*** CALCULATE TEMPERATURE
50 XN=TCF(IR,2)*T+TCF(IR,3)
DF=TCF(IR,4)+TCF(IR,5)*XN+TCF(IR,6)*XN**2+TCF(IR,7)*XN**3
* + TCF(IR,8)*XN**4+TCF(IR,9)*XN**5
T=DF
RETURN
END
C
C SUBROUTINE TCPARM(TC)
C
C********************************************************************E249
C TCPARM CALCULATES THE THERMOCOUPLE WIRE PARAMETERS AND
C PUTS THEM INTO THE ARRAY "TC"
C ********************************************************************E249
C — IDENTIFICATION —
C
C** TC IS THE ARRAY OF THERMOCOUPLE PARAMETERS CALCULATED, LISTED BELOW
C — OUTPUT
C
C 1. DENSITY (RHO)
C 2. THERMAL CONDUCTIVITY (XK)
C 3. SPECIFIC HEAT (CP)
C 4. THERMAL DIFFUSITY (AL)
C
C CALLED BY MAIN PROGRAM
C -- IDIOSYNCRACIES --
C
C THE VALUE OF "IFLAGS(2)" DETERMINES WHICH SET OF EQUATIONS WILL BE USED
C
C 1. WHEN IFLAGS(2) IS 1, THE EQUATIONS FOR PT/6%RH TC'S WILL BE USED
C
C 2. WHEN IFLAGS(2) IS 2, THE EQUATIONS FOR PT/30%RH TC'S WILL BE USED
C
C 3. WHEN IFLACS(2) IS 3, THE EQUATIONS FOR CR/AL TC'S WILL BE USED
C
C******************************************************************************
C
DIMENSION TC(4)
COMMON /INPUTS/ IFLACS(12),TCDATA(4,2),FREQ(4),CHANL(9),
* IAVDAT(2),IBLSZ,IREC(10),TIME(2),IBSZ,GAMMA,NREC(10),NRECS(2),
* PLTFRQ,TIMTEM,IDEBUG,IBUC2,ITHRSH
COMMON /DATAS/ C(45),TCF(1119)
C
T=CAS
C
*** CHECKING ON T/C MATERIAL CODE
IF(IFLAGS(2).EQ.1 .OR. IFLAGS(2).EQ.2) GO TO 10
IF(IFLAGS(2).EQ.3) GO TO 30
C
C*** EQUATIONS FOR MATERIAL PT / 6% RH -OR- PT /30% RH
10 RH01=1278.7
XK1=C(1)+C(2)*T+C(3)*T**2+C(4)*T**3
CP1=C(5)+C(6)*T+C(7)*T**2+C(8)*T**3
AL1=C(9)+C(10)*T+C(11)*T**2+C(12)*T**3
C
RH02=1092.1
XK2=C(13)+C(14)*T+C(15)*T**2+C(16)*T**3
CP2=C(17)+C(18)*T+C(19)*T**2+C(20)*T**3+C(21)*T**4
AL2=C(22)+C(23)*T+C(24)*T**2+C(25)*T**3
C
RHO = (RH01 * RH02) / 2.0
XK = (XK1 + XK2) / 2.0
CP = (CPI + CP2) / 2.0
AL = (AL1 + AL2) / 2.0
GO TO 40
C
C*** EQUATIONS FOR MATERIAL CU / AL
30 RHO = 640.95
XK = (0.01547 * T + 24.505) / 2.0
CP = (.0001129 * T + .21454) / 2.0
AL = XK / (RHO * CP * 3600.)
C
C*** EVALUATING THE PARAMETERS
40 TC(1)=RHO
TC(2)=XK/3600.
TC(3)=CP
TC(4)=AL
RETURN
END

C
C SUBROUTINE TERM(I)
C
C******************************************************************************
C 'TERM' Terminates the program due to an illegal user entry, or *
C because a calculated gamma was not able to be found *
C******************************************************************************
C
C — IDENTIFICATION —
C
C** I - Code passed to determine which error caused termination of
C the program
C
C CALLED FROM CHECK SUBPROGRAM
C INTERP SUBPROGRAM
C
C FILES USED:
C 6 - Writes out appropriate reasons for any premature
C termination of the programs execution.
C
C******************************************************************************
C
C 11 = I - 15
IF(I.LT.20) WRITE(6,210)
GO TO (10,20,30,40,50,60,70,80,90,100,110,120,130,140,150),I
GO TO (160,170,180,190,200),II
10 WRITE(6,220)
WRITE(6,420)
STOP
20 WRITE(6,230)
WRITE(6,420)
STOP
30 WRITE(6,240)
WRITE(6,420)
STOP
40 WRITE(6,250)
WRITE(6,420)
STOP
50 WRITE(6,260)
WRITE(6,420)
STOP
60 WRITE(6,270)
WRITE(6,420)
STOP
70 WRITE(6,280)
WRITE(6,420)
STOP
80 WRITE(6,290)
WRITE(6,420)
STOP
90 WRITE(6,300)
WRITE(6,420)
STOP
100 WRITE(6,310)
WRITE(6,420)
STOP
110 WRITE(6,320)
WRITE(6,420)
STOP
120 WRITE(6,330)
WRITE(6,420)
STOP
130 WRITE(6,340)
WRITE(6,420)
STOP
140 WRITE(6,350)
WRITE(6,420)
STOP
150 WRITE(6,360)
WRITE(6,420)
STOP
160 WRITE(6,370)
WRITE(6,420)
STOP
170 WRITE(6,380)
WRITE(6,420)
STOP
180 WRITE(6,390)
WRITE(6,420)
STOP
190 WRITE(6,400)
WRITE(6,420)
STOP
200 WRITE(6,410)
STOP
210 FORMAT(' ', 'EXECUTION TERMINATED DUE TO ILLEGAL USER INPUT.')
220 FORMAT(' ', '** INVALID SELECTION FOR IFLAGS(1)**')
230 FORMAT(' ', '** INVALID SELECTION FOR IFLAGS(2)**')
240 FORMAT(' ', '** INVALID SELECTION FOR IFLAGS(3)**')
250 FORMAT(' ', '** INVALID SELECTION FOR IFLAGS(4)**')
260 FORMAT(' ', '** INVALID SELECTION FOR IFLAGS(5)**')
270 FORMAT(' ', '** INVALID SELECTION FOR IFLAGS(6)**')
280 FORMAT(' ', '** INVALID SELECTION FOR IFLAGS(7)**')
290 FORMAT(' ', '** INVALID SELECTION FOR IFLAGS(8)**')
300 FORMAT(' ', '** INVALID SELECTION FOR IFLAGS(9)**')
310 FORMAT(' ', '** INVALID SELECTION FOR IFLAGS(10)**')
320 FORMAT(' ', '** INVALID SELECTION FOR IFLAGS(11)**')
330 FORMAT(' ', '** INVALID SELECTION FOR IFLAGS(12)**')
340 FORMAT(' ', 'USER IMPLIED A PRE-DETERMINED VALUE OF GAMMA WAS TO BE USED', 'USER IMPLIED HE WANTED PLOTS OF THE INSTANTANEOUS COMPUTED SPECTRA', 'USER IMPLIED HE WANTED A PARTIAL TIME RANGE TO BE PLOTTED')
350 FORMAT(' ', 'USER IMPLIED HE WANTED PLOTS OF THE INSTANTANEOUS COMPUTED SPECTRA', 'USER IMPLIED HE WANTED A PARTIAL TIME RANGE TO BE PLOTTED')
360 FORMAT(' ', 'USER IMPLIED HE WANTED A PARTIAL TIME RANGE TO BE PLOTTED')
**"TIME"**

370 FORMAT( ' ', 'FOR PARTIAL TIME RANGE, ENDING TIME MUST BE GREATER THAN STARTING TIME')

380 FORMAT( ' ', 'USER IMPLIED HE WANTED INSTANTANEOUS PLOTS OF THE FREQUENCY DOMAINS, AND NO RECORDS WERE INPUT TO THE ARRAY "NREC"')

390 FORMAT( ' ', 'USER IMPLIED HE WANTED AVERAGED PLOTS OF THE FREQUENCY DOMAINS, AND NO RECORDS WERE INPUT TO THE ARRAY "NRECS"')

400 FORMAT( ' ', 'USER IMPLIED HE WANTED ALL INSTANTANEOUS PLOTS SCALED TO THE SAME TEMPERATURE (IFLAGS(12)=2), BUT NO TEMPERATURE WAS ENTERED TO TIMTEM')

410 FORMAT( ' ', 'EXECUTION TERMINATED - NO CALCULATED VALUE OF GAMMA WAS FOUND')

420 FORMAT( ' ', 'PLEASE CHECK USER INPUTS AND TRY AGAIN')

END

C

SUBROUTINE TRANCS(J,NGAM,GS,TC,TRAN,COMP)

C TRANCS EVALUATES THE TRANSFER FUNCTIONS OF THE T/C -VS- CS FOR NCAM VALUES OF GAMMA AND STORES THEM IN TRAN(NGAM,J,LOC,TYPE)

C - IDENTIFICATION -

C** J - T/C FOR WHICH THE TRANSFER FUNCTION IS DESIRED: 1=LARGE, 2=SMALL

C** NGAM - NUMBER OF GAMMA VALUES FOR WHICH THE TRANSFER FUNCTION IS DERIVED. NOTE: NGAM SHOULD BE 17 IF USING THE THEORETICAL VALUE OF GAMMA, OR 1 IF USING THE MEASURED VALUE OF GAMMA.

C** GS - ARRAY CONTAINING THE AERODYNAMIC PARAMETER, GAMMA.

C** TC - ARRAY CONTAINING THE THERMOCOUPLE PARAMETERS REQUIRED IN TRFP

C** TRAN - ARRAY INTO WHICH THE TRANSFER FUNCTION IS PUT

C** COMP - ARRAY INTO WHICH THE COMPENSATION SPECTRUM WILL BE PLACED

C CALLED BY MAIN PROGRAM

C CALLS - TRFP: THIS SUBPROGRAM CALCULATES THE COEFFICIENTS FOR THE TRANSFER PROGRAM "TRFM"

C TRFM: THIS SUBPROGRAM EVALUATES THE TRANSFER FUNCTION.

C PLT2: THIS SUBPROGRAM PLOTS THE INPUT ARRAYS
C FILES USED:
6 - WRITES THE COMPENSATION SPECTRUM TO THIS FILE
IF THE USER SO DESIRES (IBUG2 = 1)
12 - WRITES THE COMPENSATION SPECTRUM TO THIS FILE
C
C******************************************************************************
C
DIMENSION TRAN(17,3,1024,2),A(2),GS(10),COMP(1024,2),
* REA(1024),RIMA(1024),TP(10),TC(4)
COMMON /INPUTS/ IFLAGS(12),TCDATA(4,2),GAS(4),FREQ(4),CHAN(9),
* IAVDAT(2),IBLSZ,IREC(10),TIME(2),IBSZ,GAMMA,NREC(10),NRECS(2),
* PLIFRO,TIMTEM,IDEBUG,IBUG2,ITHRSH
C
A(1) = GS(10)
C
C** SETTING A TEMPORARY VARIABLE FOR THE APPROPRIATE DELTA-F
T-FREQ(1)+.00001
C
C*** SETTING THE START CHANNEL, END CHANNEL, AND STEPPING INCREMENT FOR
C*** THE PIECEWISE TRANSFORM IN THE COMPENSATION SPECTRUM
IF(NGAM.EQ.17) GO TO 10
ISTCH = 2
IEDCH = IBSZ - 1
ISTEP = 1
GO TO 20
C
C*** SETTING THE START CHANNEL, END CHANNEL, AND STEPPING INCREMENT FOR
C*** THE USER SPECIFIED FREQUENCIES

10 CHN = FREQ(2) / FREQ(1)
ISTCH = CHN +1.49
CHN = FREQ(3) / FREQ(1)
IEDCH = CHN +1.49
CHN = FREQ(4) / FREQ(1)
ISTEP = CHN +.05
C
20 DELGMA=.2
IC1-ISTCH
C
C*** LOOP FOR EACH 'TEST' GAMMA USED FOR AN ESTIMATED TRANSFER FUNCTION
DO 110 IC-1,NCAM
OLDPHS=0.
IF(NGAM.EQ.1) GO TO 30
TGMA=GS(10)*DELCMA
GO TO 40
30 TGMA=GS(10)
C
C*** LOOPING THROUGH THE CORRECT FREQUENCIES
40 DO 100 IC-ISTCH,IEDCH,ISTEP
IF(NGAM.EQ.17) GO TO 60
C
C** CHECKING ON CHANNELS FOR PIECEWISE TRANSFORM
IF(IC.GE.IEDCH) GO TO 60
IF(IC.GE.51) GO TO 60
IC1-IC1+1
GO TO 60
50 IF(IC.LT.IC1) GO TO 100
   IC1=IC1+10
C
C*** CALLING ROUTINES TO EVALUATE THE TRANSFER FUNCTION
60 A(2)=TCMA
   FRQ=T*(IC-1)
   CALL TRFP(J,TC,FRQ,A,TP)
   CALL TRFM(TP,FRQ,A)
   IF(OLDPHS.GT.A(2)) GO TO 70
   A(2)=OLDPHS
   GO TO 80
70 OLDPHS=A(2)
C
C*** STORING GAIN AND PHASE AS A TRANSFER FUNCTION OR AS COMPENSATION
C*** SPECTRUM DEPENDING ON GAMMA (TEST GAMMAS OR MEASURED GAMMA)
80 IF(NGAM.EQ.1) GO TO 90
   TRAN(IG,J,IC,1) = A(1)
   TRAN(IG,J,IC,2) = A(2)
   GO TO 100
90 COMP(IC,1) = A(1)
   COMP(IC,2) = A(2)
100 CONTINUE
C—— NEXT FREQUENCY
   DELGMA = DELGMA +.1
110 CONTINUE
C—— NEXT GAMMA
   IF(NGAM.EQ.17) RETURN
C
C*** FILL IN THE COMPENSATION SPECTRUM LINEARLY BETWEEN EVALUATED POINTS
   INT = IEDCH / 10
   ISTP = INT*10 - 9
   DO 130 I = 61,ISTP,10
      IX1 = I + 1
      IX2 = I + 9
      DO 120 JJ = IX1,IX2
         COMP(JJ,1) = (COMP(I+10,1) - COMP(I,1)) * (JJ-I) / 10. + COMP(I,1)
         COMP(JJ,2) = (COMP(I+10,2) - COMP(I,2)) * (JJ-I) / 10. + COMP(I,2)
120 CONTINUE
130 CONTINUE
   ISZ = IBSZ - 1
   ISTP = ISTP + 10
   I1 = ISTP + 1
   I2 = IBSZ - 2
   DO 140 JJ = I1,I2
      COMP(JJ,1) = (COMP(ISZ,1)-COMP(ISTP,1))*(JJ-ISTP) / (ISZ-ISTP)
      COMP(JJ,2) = (COMP(ISZ,2)-COMP(ISTP,2))*(JJ-ISTP) / (ISZ-ISTP)
140 CONTINUE
C
C*** STORE VARIOUS DATA ON DISK FILES FOR FUTURE ACCESS
   REWIND 12
   GMAMET = GS(10) * .168279
WRITE(12) (COMP(JJ,1),JJ=1,ISZ),(COMP(JJ,2),JJ=1,ISZ),
* IFLAGS(2),IFLAGS(3),GMAMET,((TCDATA(I,JJ),I=1,4),JJ=1,2)E249
C
C*** PLUG COMPENSATION SPECTRUM INTO ARRAYS FOR PLOTTING
ISIZ - IBSZ - 1
DO 160 I = 1,ISIZ
REA(I) = COMP(I,1)
RIMA(I) = COMP(I,2)
160 CONTINUE
IF(IFLAGS(4).EQ.1) CALL PLT2(REA,RIMA,1,GS,0,0.0)
IF(IBUG2.EQ.0) GO TO 170
WRITE(6,190)
ISTOP = ISZ/2
DO 160 I = 1,ISTOP
II = I + ISTOP
WRITE(6,200) I,REA(I),RIMA(I),II,REA(II),RIMA(II)
160 CONTINUE
C
C CHANGE TO RECTANGULAR COORDINATES
170 DO 180 I = 1,IBSZ
GAIN = COMP(I,1)
COMP(I,1) = GAIN * COS(COMP(I,2) * 3.1415927 / 180.)
COMP(I,2) = GAIN * SIN(COMP(I,2) * 3.1415927 / 180.)
180 CONTINUE
RETURN
190 FORMAT('1',THE COMPENSATION SPECTRUM IN POLAR GAIN AND PHASE')
200 FORMAT('2(2X,'I = 'I4,3X,E13.7,3X,E13.7,6X))
C
C SUBROUTINE TRANTC(NGAM,TRAN)
C
C**************************************************************************************
C TRANTC EVALUATES THE TRANSFER FUNCTION BETWEEN *E249
C THE LARGE AND SMALL T/C *E249
C**************************************************************************************
C
C — IDENTIFICATION —
C
C
C** NGAM - NUMBER OF GAMMA VALUES TRANSFER FUNCTION TO BE EVALUATED FOR E249
C — INPUT E249
C
C** TRAN - ARRAY CONTAINING THE TRANSFER FUNCTIONS OF THE LARGE AND E249
C SMALL T/C -VS- GAS STREAM (FOR Input), ALONG WITH THE E249
C OUTPUT OF THIS SUBROUTINE E249
C — INPUT AND OUTPUT E249
C
C CALLED BY MAIN PROGRAM E249
C
C**************************************************************************************
C
COMMON /INPUTS/ IFLAGS(12), TCDATA(4,2), GAS(4), FREQ(4), CHANL(9),
* IAVDAT(2), IBLSZ, IREC(10), TIME(2), IBSZ, GAMMA, NREC(10), NRECS(2),
* PLTFRQ, TIMTEM, IDEBUG, IBUG2, ITHRSH E249
DIMENSION TRAN(17,3,1024,2)

C

C*** START AND STOP CHANNELS AND THE STEPPING INCREMENT
IST=FREQ(2)/FREQ(1) + 1.49
IED=FREQ(3)/FREQ(1) + 1.49
INC=FREQ(4)/FREQ(1) + .05

C

C*** LOOPING THROUGH THE 'TEST' GAMMAS AND THE CHANNELS
DO 40 IC=1,NGAM
DO 30 IC=IST,IED,INC
G3=TRAN(IG,2,IC,1)
P3=TRAN(IG,2,IC,2)
G10=TRAN(IG,1,IC,1)
P10=TRAN(IG,1,IC,2)

C

C*** EVALUATING 10MIL VS 3MIL TRANSFER FUNCTION
TF10T3=G10/G3
PH10T3=P10-P3

C

C*** CHECKING AND CORRECTING FOR QUADRANT OF PHASE
T=ABS(PH10T3)
IF(180.0.GE.T) GO TO 20
IF(PH10T3.GE.0.0) GO TO 10
PH10T3=PH10T3+360.
GO TO 20
10 PH10T3=PH10T3-360.

C

20 TRAN(IG,3,IC,1)=TF10T3
TRAN(IG,3,IC,2)=PH10T3
30 CONTINUE
40 CONTINUE

C

RETURN
END

C

SUBROUTINE TRFM(TP,FRQ,A)

C

C "TRFM" IS A SUBPROGRAM THAT EVALUATES THE TRANSFER FUNCTION BETWEEN THE THERMOCOUPLE WIRE AND THE GAS STREAM.

C** TP - PARAMETERS FOUND BY TRFP NEEDED TO EVALUATE TRANSFER FUNCTION
C — INPUT
C
C** FRQ - FREQUENCY FOUND IN TRANGS, NEEDED IN SPCY
C — INPUT
C
C** A - ARRAY INTO WHICH THE GAIN AND PHASE ARE STORED
C — OUTPUT
C CALLED BY TRANGS SUBPROGRAM
C CALLS - SPCY: DETERMINES THE SAMPLING FREQUENCY AS A FUNCTION OF THE ANALYSIS FREQUENCY

C*******************************************************************************/
C
C DIMENSION Z(10),ZP(10),TP(10),A(2)
C
C*** INITIALIZING VARIABLES
CALL SPCY(FRQ,A)
XN2=A(1)
N2=XN2
LAP=0
P1=0.
P2=0.
P3=0.
ZC1=0.
ZC2=0.
PKPOS=0.
PKNEG=0.
DO 10 I=1,10
Z(I)=0.
ZP(I)=0.
10 CONTINUE
DELTAT=TP(2)
T=0.0
CN=TP(4)
A1=TP(5)
B=TP(6)
C=TP(7)
E=TP(8)
F=TP(9)
G=TP(10)

C
C*** FINITE DIFFERENCE METHOD FOR TRANSFER FUNCTION (UNTILL CONVERGENCE)
DO 70 I=1,32000
T1=Z(1)
T2-Z(2)
T3-Z(3)
T4-Z(4)
T5-Z(5)
T6-Z(6)
T7-Z(7)
T8-Z(8)
T9-Z(9)
TO-Z(10)
TR-SIN(CN*T)
ZP(1)=-.5*(C*(T0*T2-2*T1))/(A1)+T1
ZP(2)=-.5*(C*(T1*T3-2*T2))/(A1)+T2
ZP(3)=-.5*(C*(T2*T4-2*T3)+F*(TR-T3))/(A1)+T3
ZP(4)=-.5*(C*(T3*T5-2*T4)+(2*F)*(TR-T4))/(A1)+T4
ZP(5)=-.5*(C*(T4*T6-2*T5)+(2*F)*(TR-T5))/(A1)+T5
ZP(6)=(C*(T5-T6)+E*(T7-T6)+(F+G)*(TR-T6))/(A1+B)+T6

98
\[ \begin{align*}
ZP(7) &= (0.5) \times (E \times (T6 + T8 - 2 \times T7) + (2 \times G) \times (TR - T7)) / (B) + T7 \\
ZP(8) &= (0.5) \times (E \times (T7 + T9 - 2 \times T8) + (2 \times G) \times (TR - T8)) / (B) + T8 \\
ZP(9) &= (E \times (T8 - T9) + G \times (TR - T9)) / (B) + T9 \\
P1 &= P2 \\
P2 &= ZP(9) \\
P3 &= ZP(9) \\
T &= T + \text{DELTAT} \\
LAP &= LAP + 1 \\
\text{IF}(N2 > \text{LAP}) \text{ GO TO } 20 \\
T &= 0 \\
LAP &= 0
\end{align*} \]

**C**

**C*** RESETTING THE VARIABLES

20 DO 30 IP = 1, 9
\[ Z(IP) = ZP(IP) \]
30 CONTINUE

\[ \begin{align*}
\text{IF}(P2 > 0.0) \text{ GO TO } 40 \\
\text{IF}(P3 < 0.0) \text{ GO TO } 40 \\
ZC1 &= P2 \\
ZC2 &= P3 \\
XIC &= I \\
XIC &= XIC - 2.
\end{align*} \]

40 IF(ABS(P1) < GT.ABS(P2)) GO TO 70
\[ \begin{align*}
\text{IF}(ABS(P2) \leq \text{ABS}(P3)) \text{ GO TO } 70 \\
\text{IF}(P2 > 0.0) \text{ GO TO } 50 \\
PKNeg &= P2 \\
\text{GO TO } 60 \\
PKPOS &= P2
\end{align*} \]

50 PKPOS = P2

**C**

**C*** CHECKING ON CONVERGENCE

60 PKDIF = PKPOS - ABS(PKNEG)
\[ \begin{align*}
\text{IF}(PKNEG = 0.0) \text{ PKQT = PKDIF} \\
\text{IF}(PKNEG \neq 0.0) \text{ PKQT = PKDIF / ABS(PKNEG)} \\
\text{IF}(PKQT < 0.001) \text{ GO TO } 80 \\
XXX &= 200.0 \times XN2 \\
\text{IF}(XIC < XXX) \text{ GO TO } 70 \\
PKPOS &= (PKPOS - PKNEG) / 2. \\
\text{GO TO } 80
\end{align*} \]

70 CONTINUE

80 A(1) = PKPOS
\[ \begin{align*}
XNC &= XIC / XN2 \\
NC &= XNC \\
YNC &= NC \\
CHR &= YNC \times XN2 \\
FRC &= XIC - CHR \\
ZC &= ZC2 - ZC1 \\
\text{IF}(ZC = 0.0) \text{ GO TO } 90
\end{align*} \]

C

\[ \begin{align*}
\text{FRC} &= \text{FRC} - ZC1 / (ZC2 - ZC1) \\
\text{PHSLAG} &= (\text{FRC} / XN2) \times (-360.) \\
A(2) &= \text{PHSLAG} \\
\text{RETURN}
\end{align*} \]
C
C SUBROUTINE TRFP(J,TC,FRQ,A,TP)
C
C**********************************************************************E249
C "TRFP" CALCULATES THE PARAMETERS REQUIRED BY THE
C TRANSFER FUNCTION PROGRAM "TRFM".
C**********************************************************************E249
C
C IDENTIFICATION
C
C** J - CODE FOR WHICH T/C IS BEING EVALUATED
C INPUT
C
C** TC - THE ARRAY HOLDING THE T/C PARAMETERS
C INPUT
C
C** FRQ - FREQUENCY FOUND IN TRANGS, NEEDED IN SPCY
C INPUT
C
C** A - ARRAY INTO WHICH THE SAMPLING FREQUENCY FOUND BY
C SUBROUTINE SPCY WILL BE PUT
C OUTPUT
C
C** TP - ARRAY CONTAINING THE PARAMETERS EVALUATED BY TRFP
C OUTPUT
C
C 1. DELTA
C 2. DELTAT
C 3. SIGMA
C 4. CN
C 5. A1
C 6. B
C 7. C
C 8. E
C 9. F
C 10. G
C
C REFERENCE "DYNAMIC GAS TEMPERATURE MEASURING SYSTEM - SYSTEM
C DESIGN AND TEST PLAN (FR-16381)" FOR DEFINITION OF ABOVE TERMS.
C
C CALLED BY TRANGS SUBPROGRAM
C CALLS - SPCY: DETERMINES THE SAMPLING FREQUENCY AS A FUNCTION
C OF THE ANALYSIS FREQUENCY
C
C**********************************************************************E249
C
C DIMENSION A(2),TC(4),TP(10)
C COMMON /INPUTS/ IFLAGS(12),TCDATA(4,2),GAS(4),FREQ(4),CHANL(9),
C IADVAT(2),TBLSZ,IREC(10),TIME(2),IBSZ,GAMMA,NREC(10),NRECS(2)
** PLTFRQ, TIMTEM, IDBUG, IBUG2, ITHRSH**

C

C*** FIND THE SAMPLING FREQUENCY
CALL SPCY(FRQ,A)
XN2=A(1)
C
C*** CALCULATE THE TRANSFER FUNCTION PARAMETERS
DELTA = TCDATA(1,J) / 3.0
DELTAT = 1./(XN2*FRQ)
SIGMA = TCDATA(2,J) / 3.0
CN = 2.*3.1415*FRQ
A1 = (TCDATA(3,J)**2 * DELTA) / (8.*TC(4)*DELTAT)
B = (TCDATA(4,J)**2 * SIGMA) / (8.*TC(4)*DELTAT)
C = (TCDATA(3,J)**2) / (4.*DELTA)
E = (TCDATA(4,J)**2) / (4.*SIGMA)
F = (A(2) * SQRT(TCDATA(3,J)) * DELTA) / (2.*TC(4))
G = (A(2) * SQRT(TCDATA(4,J)) * SIGMA) / (2.*TC(4))
C
C*** STORE THE PARAMETERS INTO THE ARRAY TP
TP(1) = DELTA
TP(2) = DELTAT
TP(3) = SIGMA
TP(4) = CN
TP(5) = A1
TP(6) = B
TP(7) = C
TP(8) = E
TP(9) = F
TP(10) = G
RETURN
END
C
C
SUBROUTINE WINDOW(WIND0, DATA)
C
C *********** WINDOW APPLIES THE P301 WINDOW TO THE INPUT DATA BLOCK ***********
C
C *********** WINDOW APPLIES THE P301 WINDOW TO THE INPUT DATA BLOCK ***********

C
C — IDENTIFICATION —
C
C
C** WIND0 - THE P301 WINDOW FOUND IN WINGEN
C — INPUT
C
C** DATA - ARRAY TO WHICH THE WINDOW IS TO BE APPLIED
C — INPUT AND OUTPUT
C
C CALLED BY POWER SUBPROGRAM
C
C
C*********** WINDOW APPLIES THE P301 WINDOW TO THE INPUT DATA BLOCK ***********

C
C DIMENSION WIND0(2048), DATA(2048)
COMMON /INPUTS/ IFLAGS(12), TCDATA(4,2), GAS(4), FREQ(4), CHANL(9),

101
* IAVDAT(2),IBLSZ,IREC(10),TIME(2),IBSZ,GAMMA,NREC(10),NRECS(2), E249
* PLTFRQ,TIMTEM,IDEBUG,IBUG2,ITHRSH
C
DO 10 I = 1,IBLSZ
DATA(I) = DATA(I) * WINDO(I)
10 CONTINUE
RETURN
END
C
C  SUBROUTINE WINGEN(WINDO)
C
C*******************************************************************************
C WINGEN GENERATES THE P301 WINDOWING FUNCTION
C*******************************************************************************
C IDENTIFICATION

C WINDO - ARRAY TO CONTAIN THE WINDOWING FUNCTION

C OUTPUT

C CALLED BY POWER SUBPROGRAM

C*******************************************************************************
C
DIMENSION WINDO(2048)
COMMON /INPUTS/ IFLAGS(12),TCDATA(4,2),GAS(4),FREQ(4),CHANL(9),
* IAVDAT(2),IBLSZ,IREC(10),TIME(2),IBSZ,GAMMA,NREC(10),NRECS(2),
* PLTFRQ,TIMTEM,IDEBUG,IBUG2,ITHRSH
C
TWOPI = 2.0 * 3.1415927
DO 10 I = 1,IBLSZ
WINDO(I) = 0.9994484 + 2.0 * (0.955728*COS(TWOPI*(I-1)/IBLSZ) +
* 0.639289*COS(2.0*TWOPI*(I-1)/IBLSZ) +
* 0.091581*COS(3.0*TWOPI*(I-1)/IBLSZ))
10 CONTINUE
ISIZ = IBSZ - 1
DO 20 I = 1,ISIZ
II = I + ISIZ
TEMP = WINDO(II)
WINDO(II) = WINDO(I)
WINDO(I) = TEMP
20 CONTINUE
RETURN
END
C*******************************************************************************
TEST CASES

The program was developed and tested on an IBM 3090 computer with a VM/CMS operating system. The CMS environment facilitated interactive execution and testing of the program. The program was written in the IBM System/370 FORTRAN IV language.

Seven test cases were run to show the available user options. A sample EXEC with comments as to the statements functions is shown in Figure 10.
&TRACE off
GLOBAL TXTLIB FGHLIB CMSLIB PWALIB SUBLIB
EXEC CPU
&READ VARS &MINUTES &SECONDS &HUNDRED
CLEAR
EXEC PACKSCAN ENCOOS STACK LINK
&READ VARS &AVD &MOD
FI 4 DISK E249SIN DATA &MOD DSN E100827 $E249 SIN DATA
FI 5 DISK E249SIN INPUT A (LRECL 80 BLKSIZE 80)
FI 6 DISK E249SIN OUTPUT A (LRECL 133 BLKSIZE 133)
FI 12 DISK E249COM3 SIN A
FI 13 DISK E249DATS SIN A
FI 14 DISK E249DATL SIN A
FI 15 DISK E249FFTS SIN A
FI 16 DISK E249FFTL SIN A
FI PLOTPARM DISK VECINR DATA 0
FI PLOTLOG TERMINAL
FI VECTR1 DISK VECTR1 DATA 0
FI VECTR2 DISK VECTR2 DATA 0
&TYPE EXECUTING ...........
E249
STATE VECTR1 DATA A
*
&IF &RETCODE NE 0 &GOTO -CONT
EXEC USE SYNCSORT
&STACK 1 4 CH A
SSORT VECTR2 DATA A SECTR2 DATA A (OUTPUT REP
FILEDEF PLOTLOG TERMINAL
FILEDEF RJVECTR PRINTER (LRECL 132 BLOCK 132 RECFM FB)
FILEDEF SECTR2 DISK SECTR2 DATA A
FILEDEF VECTR1 DISK VECTR1 DATA A
EXEC ROUTE PRINT LOCAL SYSOUT-6
CP SPOOL PRT CONT
LOADM VTPlot
START
CP SPOOL PRT CLOSE
CP SPOOL PRT NOCONT
ERASE VECTR1 DATA A
ERASE VECTR2 DATA A
ERASE SECTR2 DATA A
**** ACCOUNTING INFORMATION ************
EXEC CPU &MINUTES &SECONDS &HUNDRED
&READ VARS &MINUTES &SECONDS &CRU &COST
EXEC ACNT2 183 E249 &CRU &COST
&TYPE VIRTUAL CPU USED : &MINUTES MINUTE(S) &SECONDS SECOND(S)
&TYPE ESTIMATED CRU : &CRU
&TYPE ESTIMATED COST : $ &COST

************
EXEC USE SYNCSORT (DET)
*REL &MODE (DET)
&EXIT

Figure 10. Sample EXEC Program
Test Case 1

The input file that was used to run the first test case is shown in Figure 11. Please note that lines 14 through 20 are not necessary.

```
.3048   .140589 .0508  .025999 .20828  .0635  .0381  .007787
  .02  1266.48  .355
1.0895784E+06
.24414E-03
8.0   8.0   2.0
2500.0 2.0  .125  1500.0  0.5  -.2  50.0  1.0
  -.4
1  3
2048
2000.0
1 1 1 1 2 4 2 0 0 0 0 0
1 0
CHECK-OUT TEST CASE B
8 HZ SIN WAVE USING PT / 6% RH
6/6/96
```

Figure 11. Input for Test Case 1
Since this case is a first time run (IFLAGS(1)=1), a measured gamma and compensation spectrum will be calculated. IDEBUG has been set to one so a printout of the interpolation for a measured gamma will be produced. Printed output generated from this test case is shown in Figure 12. The only plot generated by this execution is of the compensation spectrum and is shown in Figure 13.

**Figure 12. Output Generated by Test Case 1**
Figure 12. Output Generated by Test Case 1 (Continued)
Figure 13. Test Case 1 Plot 1
Test Case 2

The second test case uses the compensation spectrum, scaled data, and FFT files generated in Test Case 1. This greatly reduces the execution time. This case requests an instantaneous, compensated plot of Record 1 using a threshold level of \(-80\) dB. The averaged frequency domain plot is generated using Records 1 through 3, and scaling in K^2/Hz. The input to Test Case 2 is shown in Figure 14, noting that input lines 14, 16, 17, and 19 are not necessary.

```
.3048  .140589  .0508  .025999  .20828  .0635  .0381  .007787
  .02 1266.48  .355
1.0893784E+06
.24414E-03
  8.0  8.0  2.0
  2500.0  2.0  .125 1500.0  0.5  -.2  50.0  1.0
  .4
  1 3
2048
2000.0
  3 0 0 2 1 1 1 1 1 1 1 1
  0
  1
  1
  3

CHECK-OUT TEST CASE B
COMPENSATED WITH THRESHOLD OF -8008
8 HZ SIN WAVE USING PT / 6% RH
6/6/86
```

Figure 14. Input for Test Case 2
Printed output for Test Case 2 is a card copy of the input along with a summary of the input variables shown in Figure 15. When DEBUG and IBUG2 are both set to zero, this is the only output generated. Figures 16 and 17 show the plots generated. Note that the frequency domain plots could be 'spread out' so as to see the data better by changing the value of PLTFRQ input on line 10.

Figure 15. Output Generated by Test Case 2
<table>
<thead>
<tr>
<th>FLAG</th>
<th>DESCRIPTION</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>WHERE TO BEGIN PROGRAM CALCULATIONS</td>
<td>5</td>
</tr>
<tr>
<td>1</td>
<td>T/C MATERIAL CODE</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>T/C USED FOR COMPENSATION SPECTRUM</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>PLOT OF COMPENSATION SPECTRUM DESIRED?</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>PLOT OF INSTANTANEOUS DATA DESIRED?</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>TYPE OF SCALING DONE TO FREQUENCY DATA</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>PLOT OF AVERAGED FREQUENCY DATA DESIRED?</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>AVERAGE ONE OR MANY RECORDS?</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>COMPENSATED DATA?</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>PLOT TIME AND FREQUENCY DOMAINS?</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>PLOT FULL TIME RANGE?</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>TEMPERATURE TO SCALE DATA ON PLOTS</td>
<td>1</td>
</tr>
</tbody>
</table>

OIBURG IS SET TO 0
OIBURG IS SET TO 0
EXECUTION OF THE PROGRAM HAS BEEN COMPLETED!!

Figure 15. Output Generated by Test Case 2 (Continued)
Figure 16. Test Case 2 Plot 1
Figure 17. Test Case 2 Plot 2
Test Case 3

Test Case 3 demonstrates how partial time may be requested for instantaneous time domain plots. A compensated plot of Record 1 for time 0.125 second to 0.375 second is requested using a threshold of -80 dB. The instantaneous frequency domain plot is no longer needed so IFLAGS(10) has been set to 2. The averaged frequency domain is now scaled in decibels. Figure 18 shows input for Test Case 3, while Figures 19 and 20 show plots generated by this input. Please note that lines 14, 17, and 19 are not needed in the input file.

Figure 18. Input for Test Case 3
AVERAGED FREQUENCY DOMAIN DATA

SMALL T/C COMPENSATED DATA

CHECK-OUT TEST CASE 3 COMPENSATED WITH THRESHOLD OF -8000 dB
8 Hz SIN WAVE USING PT / 6% RH
6/6/86

STARTING REC NUMBER 1.
RECORDS IN AVERAGE 3.

Figure 19. Test Case 3 Plot 1
Figure 20. Test Case 3 Plot 2
Test Case 4

Test Case 4, whose input is shown in Figure 21, again requests a compensated, instantaneous plot of Record 1. This run now has the time domain scaled to a user input temperature of 200K. The averaged frequency domain is now scaled linearly (rms K/√Hz). The plots generated are shown in Figures 22 and 23. Please note that lines 14, 16, and 17 of the input description are not necessary.

Figure 21. Input for Test Case 4
AVERAGED FREQUENCY DOMAIN DATA

SMALL T/C COMPENSATED DATA

CHECK-OUT TEST CASE B COMPENSATED WITH THRESHOLD OF -80DB
8 Hz SIN WAVE USING PT / 6% RH INSTANTANEOUS SCALED TO 200K
6/6/86

STARTING REC NUMBER 1.
RECORDS IN AVERAGE 3.

Figure 22. Test Case 4 Plot 1
COMPOSITE INSTANTANEOUS TIME WAVEFORM

SMALL T/C COMPENSATED DATA

INSTANTANEOUS DATA, RECORD NUMBER 1.

CHECK-OUT TEST CASE B COMPENSATED WITH THRESHOLD OF -80DB
8 HZ SIN WAVE USING PT / 6% RH INSTANTANEOUS SCALED TO 200K
6/6/86

Figure 23. Test Case 4 Plot 2
Test Case 5

Test Case 5 requests only a plot of the averaged frequency domain data, scaled in narrowband (rms K). The plot is compensated, but since no instantaneous data are requested, a threshold level should not be input. Figure 24 shows input for Test Case 5 (lines 14, 15, 16, 17, and 19 are not needed) while Figure 25 is the plot generated by this input.

```
.3048  .140589  .0508  .025999  .20828  .0635  .0381  .0077787
.02  1266.48  .355
1.0893784E+06
.24414E-03
0.0  8.0  2.0
2500.0  2.0  .125  1500.0  0.5  -.2  50.0  1.0
-.4
1  3
2048
2000.0
1  0  0  2  2  4  1  1  1  1
0
0
1  3
CHECK-OUT TEST CASE B
8 HZ SIN WAVE USING PT / 6% RH
6/6/86
```

Figure 24. Input for Test Case 5
AVERAGED FREQUENCY DOMAIN DATA

SMALL T/C

COMPENSATED DATA

CHECK-OUT TEST CASE 8
8 Hz. SIN WAVE USING PT / 6% RH
6/6/86

STARTING REC NUMBER 1.
RECORDS IN AVERAGE 3.

Figure 25. Test Case 5 Plot 1
Test Case 6

Test Case 6 requests uncompensated plots of instantaneous data for Record 1 and averaged frequency domain using Records 1 through 3. Frequency domain is again scaled in rms K. Test Case 6 input is shown in Figure 26. Please note lines 14, 16, 17, and 19 are not present. Figures 27 and 28 show plots generated.

```
.3048  .140589  .0508  .025999  .20828  .0635  .0381  .007787
  .02  1266.48  .355
1.0893784E+06
.24414E-03
  8.0  8.0  2.0
2500.0  2.0  .125  1500.0  0.5  -.2  50.0  1.0
  -.4
  1  3
2048
2000.0
  3  0  0  2  1  4  1  1  2  1  1  1
  0
  0
  1
  1  3
CHECK-OUT TEST CASE B
8 HZ SIN WAVE USING PT / 6% RH
6/6/86
```

*Figure 26. Input for Test Case 6*
AVERAGED FREQUENCY DOMAIN DATA

SMALL T/C  UN-COMPENSATED DATA

CHECK-OUT TEST CASE B
8 Hz SIN WAVE USING PT / 6% RH
6/6/86

STARTING REC NUMBER 1.
RECORDS IN AVERAGE 3.

Figure 27. Test Case 6 Plot 1
Figure 28. Test Case 6 Plot 2
Test Case 7

Test Case 7 illustrates the effect of the data block shifting (described in Figure 9) on the last record of data. The compensated time waveform for Record 3 was requested. Since only 3 records were generated for this test data, there was not another record to merge with Record 3. In this case the first and last quarter of the record is set to zero. This input to Test Case 7 is shown in Figure 29, and Figure 30 shows the plot generated. Please note that lines 14, 16, 17, 18, and 19 of the input description are not needed.

```
.3048 .140589 .0508 .025999 .20828 .0635 .0381 .007787
.02 1264.48 .355
1.0893784E+06
.24414E-03
8.0 8.0 2.0
2500.0 2.0 .125 1500.0 0.5 -.2 50.0 1.0
-.4
1 3
2048
3 0 0 2 1 4 2 0 1 2 1 1
0
0
3
-80
CHECK-OUT TEST CASE B
RECORD 3 (LAST RECORD)
THRESHOLD LEVEL OF -80DB

Figure 29. Input for Test Case 7
```
COMPOSIT INSTANTANEOUS TIME WAVEFORM

SMALL T/C COMPENSATED DATA
INSTANTANEOUS DATA, RECORD NUMBER 3.

CHECK-OUT TEST CASE B
8 HZ SIN WAVE USING PT / 6% RH
THRESHOLD LEVEL OF -80DB
6/9/86

Figure 30. Test Case 7 Plot 1
PROGRAM EXECUTION ON THE TSS OPERATING SYSTEM USING THE DISSPLA GRAPHICS PACKAGE

Set-Up

The first time user of GASTEMP must issue the following commands in order to access the object code for GASTEMP. These commands need only be issued once. From then on, the user has access to the required code.

SHARE D1100LI$,UUDALIB,D1100LI$
SHARE GPMLIB1$,UUDALIB,GPMLIB1$
SHARE DISSPLA,XXISSCO,DISSPLA
SHARE MIG.GASLIB,1OPRATT,MIG.GASLIB
SHARE DDUM,1OPRATT,USERLIB
PROCTRAN DDUM,GASTEMP,TAPERED,TAPEINFO

Digitizing Analog Data

Only three channels of data should exist on the digitized tape. These are 1) small diameter thermocouple ac, 2) large diameter thermocouple ac, and 3) dc reading of the large or small diameter thermocouple. If more than these three channels exist on the digital tape, an error will occur while reading the tape.

Four items of information concerning digitizing analog data will be needed in the user input file. These items are:

1) Digitizing rate (delta T)
2) Preston amplifier gain
3) Output/input ratio
4) DC offset.

Items 1 and 2 are information the user should give the technician digitizing the data. If the user does not know items 3 and 4, they can be found by using TAPEINFO if the required information was included on the analog tape and digitized along with the real data. (See TAPEINFO below.)

Execution of GASTEMP

The execution of GASTEMP is a two part process. The first run sets up the compensation spectrum, scaled data, and FFT's necessary for all plots to be generated. This run takes a large amount of connect time and the user may wish to run it as a batch job. Due to system restrictions, no plots should be requested for a first time run if it is submitted as a batch job. Two files must exist on the user's ID before GASTEMP may be executed for the first time. These files are: 1) digitized data in volts, and 2) user input file. Digitized data is a migrated dataset called MIG.NAME.DATA where NAME is user supplied to specify this particular test case. The user input file must be named NAME.INPUT and should not be a migrated dataset.

To create plots, GASTEMP should be run interactively on a Tektronics or a Solanar terminal. If hardcopies are desired, make sure the terminal is hooked up to a hardcopy device. Required input for these runs consists of the user input file, NAME.INPUT, along with migrated files created on the first time run containing compensation spectrum, scaled data, and FFTs.
The following command will initiate execution of the program: GASTEMP NAME = name, STATUS = status. For a first time run use STATUS = new, and once the generated files exist, STATUS = old may be used. Files that are created within the first execution of GASTEMP, and used for successive runs to generate plots, are named as follows:

- MIG.NAME.COMP — compensation spectrum (computed only if a measured gamma is found)
- MIG.NAME.DATS — scaled data for small diameter thermocouples
- MIG.NAME.DATL — scaled data for large diameter thermocouples
- MIG.NAME.FFTS — FFT for small diameter thermocouples
- MIG.NAME.FFTL — FFT for large diameter thermocouples.

These files are created as temporary datasets and migrated in order to save storage.

Creating MIG.NAME.DATA

A procedure exists that will create MIG.NAME.DATA for the user. The user must know the tape number that contains the digitized data, the digital reading number of the desired event, and a unique name that describes this particular test case. This name should be simple, consisting of one to eight alphanumeric characters beginning with an alphabetic. The command that begins execution of this procedure is TAPEREAD VOL= tape#, RDG= rdg#, NAME= name. The program will prompt the user for the number of records to be processed beginning with the first data point of RDG. The amount of connect time required for this procedure varies according to if the tape is already mounted, and the number of records the user wishes to process. If the user wants to submit this job to background, it may be done after entering the desired number of records. (See Submitting to Background.) The program reads in the desired amount of data, converts this data to volts, and writes it to a temporary file, NAME.DATA. This file is then migrated under the name MIG.NAME.DATA.

Creating NAME.INPUT

NAME.INPUT is the user input describing the options for a particular run. Two sample input files exist on IOPRATT called SAMPLE.INNEW and SAMPLE.INOLD. These files may be copied to the user’s ID and edited for his/her own use. These files will help the user to fit data in the correct columns and to pick appropriate options for a particular type of run. SAMPLE.INNEW is an input file for a first time run to be submitted to batch, and requesting no plots. SAMPLE.INOLD is a sample of an input file for plot generating runs, showing several of the available options.

See Description of the Computer Program for a complete input description.

TAPEINFO

TAPEINFO is a routine that will give the user information to calculate two input variables, output/input ratio, and dc offset. The command is TAPEINFO VOL= tape#, RDG= rdg#. The user is prompted for block size (see Input Description) and the variable for which information is sought.

For the output/input ratio, a reading of known input is included on the FM analog tape and digitized along with the raw data. The program calculates the rms voltage of the digitized reading and prints it on the terminal screen for the user. For example, an input of 400 mv rms is included on the analog tape and digitized. The output value from the program is 2.9446 volts rms. The output/input ratio is then calculated by dividing 0.4 volt rms into 2.9446 volts rms.
For the dc offset calculation, an input of zero volts is digitized and TAPEINFO finds the average over an entire data block. This output value is the dc offset and should be close to zero.

**Submitting to Background**

A job that is begun interactively can be submitted to finish running background by issuing the command BACK DSNAME. DSNAME is the name of a file on the user's ID which will tell the system how to finish the job. For TAPERead (and GASTEMP if the user desires), the only commands necessary are 'go' and 'logoff'. A sample file is stored on IOPRATT as SAMPLE.GOBACK. Once the user has issued the command BACK, if he/she interactively uses any of the files needed by the background job, an error will occur. To be safe, the user should logoff after submitting the job to background.

**Running GASTEMP as a Batch Job**

When GASTEMP is run with a status of 'new', the user may want to run it as a batch job due to the time required, or submit it to background once execution has begun (see above section). If a total batch job is desired, the user enters the command EXECUTE DSNAME. In this case, DSNAME is a file telling the machine which commands to execute, beginning with 'logon' and ending with 'logoff'. Since, in this case, the procdef GASTEMP controls all desired actions, the only other command needed in DSNAME is GASTEMP NAME = name, STATUS = new. A sample dataset for submitting to batch exists on IOPRATT under the name SAMPLE.BATCH.

**Warning**

The user should be aware of the correlation between the procdef parameter STATUS, and the input variable, IFLAGS(1). The input variable IFLAGS(1) determines order of program execution. The choices are:

1. The program calculates a measured value of gamma and uses that value to find the compensation spectrum (long run with IFLAGS(1) = 1).
2. The user inputs the value of gamma to be used in the calculation of the compensation spectrum (some time saved with IFLAGS(1) = 2).
3. The compensation spectrum, scaled data, and FFTs already exist in migrated files and are used to create all plots (short run with IFLAGS(1) = 3).

The procdef parameter, STATUS, controls only the allocation of datasets that contain or will contain the compensation spectrum, scaled data, and FFTs. With a status of 'new', the program assumes the files do not exist. They are created with execution of the program and migrated upon termination of the run. If a status of 'old' is specified, the program assumes the files already exist as migrated datasets. The files are restored to temporary storage and used to create plots. Upon termination of the program, the files are erased from temporary storage but still exist as migrated datasets.

The proper way to combine these two parameters is as follows: A status of 'new' should be used when the files are being created and no migrated files exist under the NAME specified (except for the raw data, MIG.NAME.DATA). IFLAGS(1) should be 1 in order to use a status of 'new'. When a status of 'old' is specified, scaled data and FFTs should exist as migrated datasets. The compensation spectrum may or may not exist depending on whether a measured gamma was found when the case was run as 'new'. IFLAGS(1) should be either two or three in order to use a status of 'old'.
If a status of 'new' is specified with an IFLAGS(1) value of three, the program will terminate with a message stating the end of the record was encountered while trying to read the compensation spectrum (file 12, dsname=MIG.NAME.COMP). The reason for this is that the procdef did not restore the migrated datasets because of the status of 'new', and the program tries to read restored files which are empty. If a status of 'old' is specified with an IFLAGS(1) value of 1, the following things may happen.

- If the files do not exist as migrated datasets, messages will arise which state the files could not be restored.
- The files created within the program are erased upon its termination and all the data calculated is lost. This will cause no permanent harm, but will waste time.

There is no way for the procdef to check if an error occurred within the execution of GASTEMP. Therefore if a status of 'new' is specified, the compensation spectrum, scaled data, and FFT files will still be migrated even if an error occurred within execution of the program. If the program is then rerun with a status of 'new', the program will execute properly but an error may occur when trying to migrate the files, since they already exist (even though they are wrong). The files will not be migrated and will exist in temporary storage under the names NAME.COMP, NAME.DATL, etc. The user may manually correct this by erasing the migrated files MIG.NAME____, renaming NAME____ to MIG.NAME____, and migrating these files. The user may also prevent this from happening by making a habit of looking at the program output, NAME.OUTPUT. This file is automatically printed for the user and stored as a permanent dataset. If the output shows an error has occurred, and the program is to be rerun, make sure all datasets that were created and migrated with the original run are erased before starting over.

**Order of Execution**

The following order of execution is recommended for use with the program GASTEMP and related programs.

**TAPEREAD VOL=tape#, RDG=rdg#, NAME=name**

Execution of this program is required before any data can be processed. Execution must begin as interactive, but may be submitted to background after entering the number of records to be processed.

**TAPEINFO VOL=tape#, RDG=rdg#**

Execution of this program is optional. The program's only use is to supply the user with information as to the values of input variables, output/input ratio, and dc offset. If the user knows these values, this program need not be run.

**NAME.INPUT**

Creation of the input dataset should be accomplished at this time. Please make use of the sample datasets that exist on IOPRATT.

**GASTEMP NAME=name, STATUS=new**

The first run of GASTEMP will create the data necessary for subsequent runs of the program. This run will take a large amount of time. The user can handle this in one of two ways.
First, the user may begin execution interactively with the above statement and then submit to background once it looks as though it is working. (See Submitting to Background, above.) Secondly, the user may submit the job as total batch. (See Running GASTEMP as a Batch Job, above.)

GASTEMP NAME=name,STATUS=old

The plotting runs of GASTEMP should be run interactively on a graphics terminal such as the Tektronics or the Solanar. The plotting package, DISSPLA, is used, and at this time, DISSPLA is restricted to interactive runs on the above terminals.

If a hardcopy of a plot is desired, the run must be on a Tektronics that is hooked to a hardcopy device. Note: To continue execution once a plot has been displayed on the screen, the user must hit 'break' followed by a 'return'. Watch the computing newsletter to keep abreast of DISSPLA changes.
APPENDIX
SUBROUTINE FLOWCHARTS

The following subroutine flowcharts are shown in Figures 31 through 34 in this appendix:

- POWER
- INTERP
- PSDFN
- CFSN
Enter

INTEST
Read One Record of Small, Large, and DC Data

Is Record Desired?

No

SCALER
Scale Data to Temperature K and Store in Files 13 and 14

WINDOW
Apply the P301 Window

FFT
Perform FFTs on Data and Store in Files 15 and 16

Self- and Cross-Conjugate Multiply To Find the Auto- and Cross-Power Spectrums

Yes

All Records Done?

Average the Auto- and Cross-Power Spectrums

Exit

Figure 31. Subroutine POWER
Access Gain from Measured Transfer Function and Estimated Transfer Functions \((0.2\Gamma \rightarrow 1.8\Gamma)\) at \(F_X\)

Interpolate Between the 0.2\(\Gamma\) to 1.8\(\Gamma\) Transfer Functions to Find a Measured Value of Gamma

---

**Figure 32. Subroutine INTERP**

---

134
Enter

GET
Access the FFTs for the Appropriate Thermocouple

Scale FFT According to User's Choice

Self-Conjugate Multiply to Find the Auto-Power Spectrum

Compensated Data?

- No
  - Divide by the Appropriate Form of the Comp Spec
  - Sum the Desired Number of Records
- Yes

Done?

- No
  - Plot?
    - No
      - Exit
    - Yes
      - Plot the Averaged Frequency Domain Data
- Yes

Figure 33. Subroutine PSDFN
Figure 34. Subroutine CSFN
1. Report No. CR-179513 - vol. 2

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle
   FURTHER DEVELOPMENT OF THE DYNAMIC GAS TEMPERATURE MEASUREMENT SYSTEM

5. Report Date
   August 1986

6. Performing Organization Code

7. Author(s)
   Dana R. Stocks

   P&W/GPD FR-19381

9. Performing Organization Name and Address
   Pratt & Whitney
   Government Products Division
   P.O. Box 109600, West Palm Beach, FL 33410-9600

10. Work Unit No.

11. Contract or Grant No.
    NAS3-24228

12. Sponsoring Agency Name and Address
    National Aeronautics and Space Administration
    NASA-Lewis Research Center
    Cleveland, Ohio 44135

13. Type of Report and Period Covered
    Draft Final Report


15. Supplementary Notes
    Program Manager, Denny L. Elmore, Pratt & Whitney
    Project Manager, G. C. Fralick, Instrumentation and Fluid Mechanics Division, NASA-Lewis
    Research Center, Cleveland, Ohio 44135

16. Abstract
   The Dynamic Gas Temperature Measurement System compensation software accepts digitized data from two different diameter thermocouples and computes a compensated frequency response spectrum for one of the thermocouples. Detailed discussions of the physical system, analytical model, and computer software are presented in this volume and in Volume I of this report under Task III. Computer program software restrictions and test cases are also presented. Compensated and uncompensated data may be presented in either the time or frequency domain. Time domain data are presented as instantaneous temperature vs time. Frequency domain data may be presented in several forms such as power spectral density vs frequency.

17. Key Words (Suggested by Author(s))

18. Date for general release
    Feb. 1987

19. Security Classif. (of this report)
    Unclassified

20. Security Classif. (of this page)
    Unclassified

21. No. of Pages
    136

22. Price*