Linearizing Thermistors for Use as Temperature Sensors

by: D. Steinmeyer
D. R. Steinmeyer

Approved by: [signature]

0. Summary:

Conventional negative temperature coefficient thermistors can be made to give a linear emf readout over a limited range if used in the right kind of circuit. A computer program is described which, given the thermistor data in tabular form and the desired temperature range, calculates optimum circuit valves which minimize the linearization error.

1. The Problem:

Thermistors can be useful devices for measuring temperature. However, their resistance approximates an exponential inverse relationship (Fig. 1, curve A):

$$R_t = R_o \exp \beta (1/T_t - 1/T_o) \quad (1a)$$

$$= R_\infty \exp (-\beta/T_t) \quad (1b)$$

or, more appropriate to measuring temperature,

$$T_t = \beta/\ln (R_\infty/R_t) \quad (1c)$$

where:

R_t = thermistor resistance at T_t

R_o = thermistor resistance at T_o, usually room temperature,

298.16K (+25°C),

R_∞ = thermistor resistance at $T_t = \infty$, and

β = beta, a parameter which is a property of the thermistor material.

Since temperature measurement is defined in terms of the thermal expansion of materials, people have come expect their temperature readouts to be linear with respect to same (Figure 1 curve B). Consequently, thermistors are poorly human engineered.

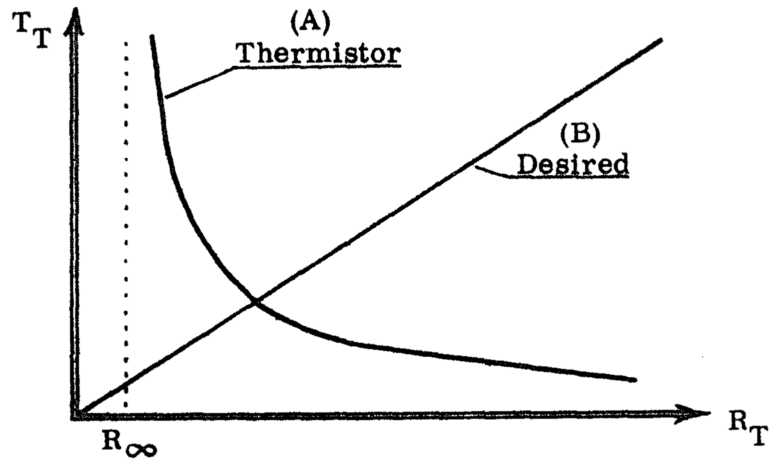

Figure 1 -- Thermistor response.

What is needed is a circuit whose output is the reciprocal log of one of its resistors. It can be shown that the total paralled resistance R_p of a combination such as R_t and R_2 can behave as a log function

$$R_p/R_2 = L + M \ln (R_t/R_2) \tag{2}$$

to a specified accuracy over a sufficiently limited range of R_t.[1] Furthermore, placing this paralled combination in the top leg of an emf divider causes the output emf to rise as the temperature rises. The circuit of Fig. 2 is the result.

[1] IEEE Transactions on Computers 1969 April, pp. 379 to 381 Donald J. Steinmeyer, Logarithm Function Generated by Parallel Resistors".

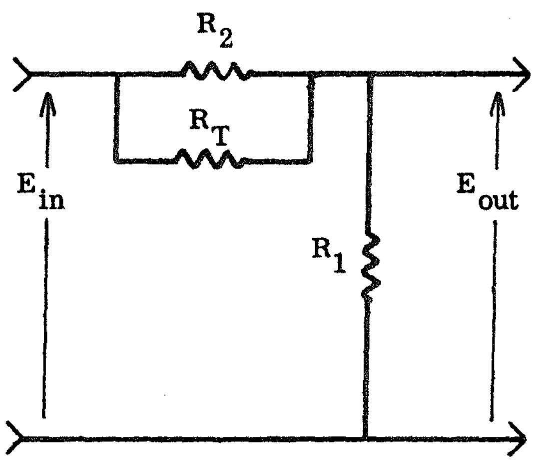

Figure 2 -- The basic thermistor emf divider.

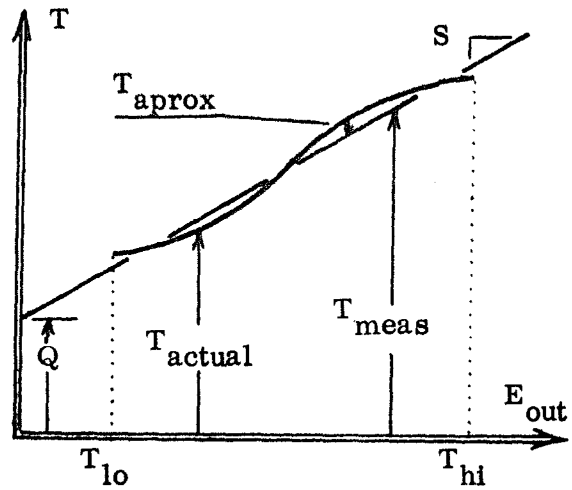

Figure 3 -- Response of the Figure 2 circuit.

If, now, R_t is made a thermistor, the circuit output E_{out} will approximate a linearly increasing function of temperature, as shown in Figure 3. Two problems, then, exist:

(1) selecting R_1 and R_2 to minimize the $T_{approx.}$ error, and (2) find the equation of the best fit straight line.

Stating the problem mathematically, it is desired to have an equation of the form

$$T_{actual} \approx Q + S\, E_{out} \tag{3}$$

where the error implied by the "≈" is as small as possible. More explicitly,

$$T_{actual} = T_{measured} - T_{approx.} - T_{self\ ht.}, \tag{4}$$

where:

T_{actual} = true temperature being measured,

$T_{measured} = Q + S\, E_{out}$, the temperature indicated by the thermistor circuit,

$T_{approx.}$ = temperature error due to the attempted linearization, and

$T_{self\ ht.}$ = temperature error due to I^2R_t in the thermistor.

2. The Solution:

A computer program, THMSTI by name, has been written which seeks to minimize these errors. The program begins with the thermistor data file; a table of T_t vs. R_t for a specified thermistor which was previously saved on the computer as file D24W.DAT. Since β varies somewhat with temperature, precision thermistors must be characterized by such a table rather than R_∞ and β.

The program will ask you for temperature range and dissapation constant. The latter is used to determine self heating error. Specification sheets usually give it for free air and well-stirred bath. The actual value lies somewhat between, depends on the thermistor mounting and environment, and requires a good solid engineering guess.

The program begins by calculating the eqn (1b) parameters and some other stuff to give you an insite into what's happening. Then it asks you to select a value of E_{in}. The program finishes by giving:

1) the best R_1 and R_2,

2) the coefficients of eqn. (3),

3) E_{out}, $T_{approx.}$ and $T_{self\ ht.}$ at each temperature in the thermistor data file, and finally

4) Sensitivity factors. These assume a + 1.0 part per thousand (+0.1%) change in each parameter and calculate the resultant temperature error.

Finally, the program recycles, permitting a re-run with different temperature range, dissipation constant, and/or E_{in}, but using the same thermistor.

3. Units:

Thermistors are sensitive to absolute temperature only. Therefore, THMSTI is written to use SI units exclusively. However, as a concession to the obsolescent user, the terminal inputs and most of the terminal outputs can be changed to any desired temperature scale by altering only one line, #864, in the program. Typical forms of line 864 would be:

```
864 &      'CELS', 'IUS ', '    ', 1.0, 273.6/

864 &      'REAU', 'MUR ', '    ', 1.25, 218.528/

864 &      'RANK', 'INE ', '    ', 0.55555, 0./

864 &      'FAHR', 'ENHE', 'IT  ', 0.55555, 459.658/
```

Without using any of these lines, 'KELV', 'IN ', '   ', 1. , 0./ is assumed.

4. Thermistor Data Input:

The thermistor data file D24W.DAT must be a list giving line number, temperature, and resistance in a specified format. Furthermore, a heading is needed to the file which gives manufacturer, part number, and entry count. Finally, regardless of the choice of line 864, D24W.DAT must be in Kelvin. The program THMSTH takes care of all thest chores.

Before running THMSTH, it is necessary to:

(1) type in the thermistor data (line no., temperature, resistance) and save it under the name D24Z.DAT and then

(2) insert one of the lines #864 corresponding to the temperature units used in D24Z.DAT.

Upon RUNning, THMSTH:

(1) asks you to enter a 2 alphanumeric digit code identifying the thermistor,

(2) asks you, in 40 digits, to identify the thermistor manufacturer and part number,

(3) uses the above information to generate a heading which is used to identify the:

(4) thermistor data which is converted into Kelvin and written in file D24W.DAT.

This completes the work of THMSTH. D24W.DAT must have line numbers to permit possible editing and use in THMSTI. This is simply done with the 2 commands.

SEQUENCE D24W.DAT

REPLACE

5. Sample Run:

In the following example, data inputed by the operator, either via keyboard or punch tape, is underlined. The thermistor under consideration is a Fenwal GB25 PM2A; the T vs. R data, copied from Bendix SCD 233 5661-1, was entered, mistakes and all, as:

```
NEW D24Z.DAT
READY

TAPE
READY

10   , -40, 10975.
20   , -20., 5844.
30   , 0., 3315.
40   , 20., 1935.
50   , 40., 1170.
60   , 60., 722.
80   , 80., 469.
100  , 10,
100  , 100., 310.
120  , 120., 210.
140  , 240
140  , 140., 145.
160  , 160., 103.4
180  , 280
180  , 180., 74.8
200  , 200., 55.5
220  , 220., 41.3
230  , 240., 31.4
240  , 260., 24.3
250  , 280., 19.0
260  ., 300., 15.1

KEY

READY

REPLACE
READY
```

```
OLD D24Z.DAT
READY

LIST

D24Z.DAT        12:35        03-JUL-72


00010  , -40, 10975.
00020  , -20., 5844.
00030  , 0., 3315.
00040  , 20., 1935.
00050  , 40., 1170.
00060  , 60., 729.
00080  , 80., 469.
00100  , 100., 310.
00120  , 120., 210.
00140  , 140., 145.
00160  , 160., 103.4
00180  , 180., 74.8
00200  , 200., 55.5
00220  , 220., 41.3
00230  , 240., 31.4
00240  , 260., 24.3
00250  , 280., 19.0
00260  , 300., 15.1


READY
```

Figure 4 -- D24Z.DAT data file as accepted by the computor

```
COPY D24W.DAT TO TTY:


000000000000000000000000000000000000000000
C   D24W B1 .DAT                        -72 JUL  3-
C   THRMISTR:   T(K), R:   BENDIX 233 5661 -1
,     18.00 ,   1.800E+01
,    233.14 ,   1.097E+04
,    244.25 ,   5.844E+03
,    255.36 ,   3.315E+03
,    266.47 ,   1.935E+03
,    277.58 ,   1.170E+03
,    288.70 ,   7.290E+02
,    299.81 ,   4.690E+02
,    310.92 ,   3.100E+02
,    322.03 ,   2.100E+02
,    333.14 ,   1.450E+02
,    344.25 ,   1.034E+02
,    355.36 ,   7.480E+01
,    366.47 ,   5.550E+01
,    377.58 ,   4.130E+01
,    388.69 ,   3.140E+01
,    399.81 ,   2.430E+01
,    410.92 ,   1.900E+01
,    422.03 ,   1.510E+01
,      0.00 ,   8.765E+21
READY
```

Figure 5 -- D24W.DAT as generated by THMSTH.

```
LIST

D24W.DAT        12:37        03-JUL-72


00100000000000000000000000000000000000000000000000000000000
00110C   D24W B1 .DAT                        -72 JUL  3-
00120C   THRMISTR:   T(K), R:   BENDIX 233 5661 -1
00130,     18.00 ,   1.800E+01
00140,    233.14 ,   1.097E+04
00150,    244.25 ,   5.844E+03
00160,    255.36 ,   3.315E+03
00170,    266.47 ,   1.935E+03
00180,    277.58 ,   1.170E+03
00190,    288.70 ,   7.290E+02
00200,    299.81 ,   4.690E+02
00210,    310.92 ,   3.100E+02
00220,    322.03 ,   2.100E+02
00230,    333.14 ,   1.450E+02
00240,    344.25 ,   1.034E+02
00250,    355.36 ,   7.480E+01
00260,    366.47 ,   5.550E+01
00270,    377.58 ,   4.130E+01
00280,    388.69 ,   3.140E+01
00290,    399.81 ,   2.430E+01
00300,    410.92 ,   1.900E+01
00310,    422.03 ,   1.510E+01
00320,      0.00 ,   8.765E+21


READY
```

Figure 6 -- D24W.DAT after SEQuencing.

Fig. 4 shows the form in which the data was accepted and stored by the computer with the duplicate line numbers, and therefore the errors, eliminated.

Next, THMSTH is invoked, keeping in mind that D24Z. DAT uses fahrenheit temperatures.

```
OLD THMSTH
READY

00864&        'FAHR','ENHE','IT  ', 0.55555, 459.658/

RUN

THMSTH.F4       12:31        03-JUL-72

FORTRAN:  005CUR.TMP
LOADING

LOADER 2K CORE
EXECUTION

GIVE  1) MATL. CODE (2 DIGITS)
      2) MFR. & P. NR. (40 DIGITS)
81, BENDIX 233 5661 -1


25  PAGE-SEC.
READY
```

The temperatures have been converted to Kelvin (see Fig. 5), a heading added, and the format made respectable. Next, line numbers are added:

```
SEQUENCE D24W.DAT
READY

REPLACE
READY
```

Fig. 6 gives the final line numbered form of D24W. DAT. At this point, D24W. DAT can be listed, a tape punched, and the data file saved for a future use with THMSTI.

THMSTI processing is next. In this example the ALSEP temperature range of -30 to +70°C will be used. Therefore, the Celsius version of line 864 is chosen.

```
OLD THMSTI
READY

00864&      'CELS','IUS ','     ', 1.0, 273.16/

RUN

THMSTI.F4      13:31      03-JUL-72

FORTRAN:  013CUR.TMP
LOADING

LOADER 3K CORE
EXECUTION

GIVE, IN CELSIUS        & WATTS/CELSIUS
TEMPR. RANGE T(LO), T(HI), AND DIS. CONST.
-30, 70, .001


THERMISTOR CHARACTERISTICS (LINEARIZED)
--------------------------

  FOR THE TYPE BENDIX 233 5661 -1  THERMISTOR
  WITH A DISAPATION CONST. OF  0.00100 WATTS/KELVIN
  OVER THE TEMPR. RANGE OF  244.2 TO  333.1 KELVIN

    R(THERMSTR)  =  5.924E-03 * EXP( 3383.7 KELVIN / T ) OHMS


RESULTS OF BEST LINEARIZATION
----------------------------

  70 VALUES OF RESISTANCE HAVE BEEN TRYED
    FROM R(2) = 5.52E+02
    TO          1.10E+03 OHMS.
    CURVE TYPE   1
  3 ITERATIONS WERE NEEDED TO GET A BEST
    MAX. EROR OF 2.62E+00 CELSIUS
  TO HOLD SELF HT. EROR WITHIN 2.62E-01 CELSIUS
    E(IN) = 2.31E+00 VOLTS OR LES.
    WHAT VALUE DO YOU SELECT FOR E(IN)?  (VOLTS)
5


POINT BY POINT PERFORMANCE (CELSIUS      )
--------------------------

  IF:  E(IN) = 5.00E+00 VOLTS
       R(1) = 1.638E+03 OHMS
       R(2) = 9.740E+02 OHMS

  THEN:
    T(ACTUAL)  =  (     62.934)*E(OUT) + (   -234.784)
                  - T(APROX.) - T(SELF HT.)

    WHERE:
          AT         E(OUT) IS      AND THE EROR TERMS ARE:
      T(ACTUAL),     (VOLTS)       T(APROX.)   T(SELF HT.)
      ---------     ---------     ---------    ---------
       -28.910      3.343E+00       2.5564      0.4876
       -17.800      3.466E+00      -1.4003      0.7478
        -6.690      3.633E+00      -2.6145      1.0380
         4.420      3.832E+00      -1.6305      1.2827
        15.535      4.045E+00       0.4941      1.4119
        15.540      4.045E+00       0.4971      1.4119
        26.650      4.247E+00       2.2680      1.3985
        37.760      4.423E+00       2.6198      1.2712
        48.870      4.565E+00       0.9932      1.0837
        59.980      4.676E+00      -2.6069      0.8825
```

```
SENSITIVITY ERORS  ("PPK" = PARTS PER THOUSAND)
-----------------

          A +1. PPK     WIL PRODUCE AN ADITIONAL
          CHANGE IN      EROR IN MEASURED TEMPR.
          ---------         OF (CELSIUS      )
           THRMSTOR     LO END    MIDLE     HI END
           CONSTS.:     ------    ------    ------
             R(INF)    -0.0101   -0.0291   -0.0182
              BETA     -1.1281   -2.3379   -1.0975
           R(T)        -0.0101   -0.0291   -0.0182
           R(1)         0.0710    0.0517    0.0211
           R(2)        -0.0603   -0.0218   -0.0027
           E(IN)        0.0421    0.0509    0.0589


↑   ↑↑↑ ↑  ↑ ↑
YOU HAVE WON A FREE GAME!  TRY AGAIN.
GIVE, IN CELSIUS       & WATTS/CELSIUS
TEMPR. RANGE T(LO), T(HI), AND DIS. CONST.
.X
.STOP

112  PAGE-SEC.
READY
```

Since no further trials were wanted at this time, the "break" key was pushed and then STOP was typed. This completed the example.

6. Availability:

THMSTH and THMSTI were written for the Cyphernet time share system. At this writing, however, they are not stored on the system. If you want to use them, ask me or Eric Granholm for the punch tapes.

The development of these 2 programs represents 8 months worth of lunch hour effort. They have been tried innumerable times and are believed to be thoroughly debugged. In the process, a large number of thermistors have been tabulated as D24W.DAT data files. These files, also available from Granholm or me on punch tape are:

Manufacturer	Part No.	File Name	Range (K)	Increment(K)
Yellow Springs	44 002	D24W02.DAT	190-415	20-5
" "	44 003	D24W03.DAT	190-415	20-5
" "	44 005	D24W05.DAT	190-424	20-5
" "	44 006	D24W06.DAT	190-425	20-5
" "	44 011	D24W11.DAT	230-425	20-5
Fenwal	GB25 PM2A			
(Bendix)	233 5661-1)	D24WB1.DAT	230-425	11
Fenwal	GB42 M/P M62A			
(Bendix)	233 5661-2/3)	D24WB2.DAT	230-500	11
Fenwal	curve A	D24WFA.DAT	210-575	20-10
"	curve D	D24WFD.DAT	210-575	20-10
"	4K series	D24WFV.DAT	215-450	20-3
Veco	isocurve	"	"	"
Fenwal	UUA & EA	D24WUA.DAT	190-425	20-5
"	UUT	D24WUT.DAT	230-425	20-5

7.0 Program Architecture:

Using the programs, while somewhat involved, is easy. Understanding how the programs do what they do is difficult. Describing this in writing is impossible. So here goes. Parenthesis enclose line numbers.

THMSTH.F4
-p.1/1

```
LIST

THMSTH.F4       14:31       03-JUL-72

00000C00000C00000C00000C00000C00000C00000
00001C             DJ24H.F4   STEINMEYER   PR72/1/5W RV6/7T
00002C             "D24Z##.DAT"  TOO "D24W##.DAT";  T(?) TOO T(K)
00009C
00860         DATA UTMPR1, UTMPR2, UTMPR3, UTMPR4, UTMPR5/
00864&         'KELV','IN  ','     ', 1., 0./
01010         DIMENSION T(200), R(200), A(10), DI(2), DO(3)
01020         CALL OPEN (5, 'D24Z.DAT!', 'I', KI)
01030C
01040C
01050C             REED FROM INPUT FYL
01060     20  DO 29 N1=2,200
01070     60  READ (5,69) IT, TT, RT
01080     40  IF (EOFTST(5))  GO TO 49
01090     69  FORMAT (I, F, F)
01100         T(N1) = (TT+UTMPR5) * UTMPR4
01110         R(N1) = RT
01120     29  CONTINUE
01130C
01140     49  CONTINUE
01150         CALL CLOSE (5)
01160C
01170C             ASK FOR TYTL, ETS.
01180         CALL OPEN (5, 'TTY:!', 'OI', KI)
01190     10  WRITE (5,19)
01200     19  FORMAT (32H GIVE  1) MATL. CODE (2 DIGITS)  /
01210&               36H       2) MFR. & P. NR. (40 DIGITS)  /)
01220     30  READ (5,39) AH, (A(N),N=1,10)
01230     39  FORMAT (A2, 1X, 10A4)
01240         CALL CLOSE (5)
01250C
01260C             SET *(1) VALEWS
01270         T(N1) = 0.0
01280         R(N1) = 8.76543E21
01290         T(1) = N1-2
01300         R(1) = N1-2
01310C
01320C             FYND DAYT
01330         CALL OPEN (7, 'DAYT!', 'OI', KD)
01340         CALL DATE (DI)
01350     90  WRITE (7,99) DI(1), DI(2)
01360     99  FORMAT (2A5)
01370         REWIND 7
01380    100  READ (7,109) (DO(K), K=3,1,-1)
01390    109  FORMAT (3A3)
01400C
01410C             RYT TOO OWTPUT FYL
01420         CALL OPEN (6, 'D24W.DAT!', 'O', KO)
01430     80  WRITE (6,89) AH, (DO(K),K=1,3), (A(K),K=1,10)
01440     89  FORMAT ( 10(6H000000)/
01450&        8HC  D24W , A2, 21H .DAT                  , 3(A3, 1X)/
01460&        23HC  THRMISTR:  T(K), R: , 10A4)
01470     50  WRITE (6,59) (T(K), R(K), K=1,N1)
01480     59  FORMAT (2H, , 0P1F7.2, 3H , , 1P1E10.3, 2H  )
01490         STOP
99999         END


READY
LEN
1996  CHARACTERS
READY
```

Figure 7 -- THMSTH program file text.

7a. D24Z.DAT: (see Fig. 4)

... is little more than the raw thermistor data transcribed from the specification sheet. Each data point takes a line. Each line consists of:

a line number,

temperature (any units), and

resistance (ohms),

with commas as delimiters. Of course, as indicated in the example, D24Z.DAT need not be as neat as Fig. 4 when first typed in.

7b. THMST H (see Fig. 7)

... is also relatively simple. It begins by reading in the values stored in D24Z.DAT (1020-1090), ignores the line numbers, converts the temperatures to Kelvin (1100) using the constants of line 864, and writes temperature and resistance in an Array (1100-1110). The number of data points is calculated (1290-1300) and placed in the first slot of the Array, and an artificial value "8.76543E21" is placed in the last (1280).

The program asks for the material code, manufacturer, and part number, storing the last 2 in another Array (1180-1240). It asks the computer to furnish the current date, which is written in the scratch file DAYT, (1330-1360) then read in reverse order into a 3rd Array to give a year-month-day sequence (1370-1390).

Finally, the output is written to file D24W.DAT, with all elements arranged in an artistic and cybernetically soothing manner. The only thing that will be missing is the line numbers.

7c. D24W.DAT (see Fig. 6)

... is not the form generated by THMST H; line numbers have been added. This is the form in which the files are punched on tape.

The body of the file consists of:

line number,

temperature (Kelvin), and

resistance (ohms),

THMSTI.F4
- p. 1/6

```
LIST

THMSTI.F4      14:09      03-JUL-72

C00000C00000C00000C00000C00000C
C         DJ24I.F4   STEINMEYER   PR.72/1/20TH RV7/3M
C         THRMISTR AZ AY TEMPR. SENSR
C
          DATA UTMPR1, UTMPR2, UTMPR3, UTMPR4, UTMPR5/
&          'KELV','IN  ','    ', 1.0, 0.0/
          DIMENSION T(100), RT(100), A(100), AN(10)
&         , TE(100), TEPRE(100)
          REAL MA, MAOELD, MAPRE, MK, MT, LN
     419  CONTINUE
          TERPRE = 9.99999E23
          TER = TERPRE
          NPOST = 0
          ADJAK = 7.
          ADJBAF = 0.95
          ADJMAF = 0.95
C
C               INPUT TEMPR. LIMITS, ETC., & CHYNJ EWNITS
     280  TYPE 289, UTMPR1,UTMPR2,UTMPR3, UTMPR1,UTMPR2,UTMPR3
     289  FORMAT (' GIVE, IN ', 3A4, ' & WATTS/', 3A4
&          / ' TEMPR. RANGE T(LO), T(HI), AND DIS. CONST.'/)
          ACCEPT, TLO, THI, DIS
          TLO = (TLO+UTMPR5) * UTMPR4
          THI = (THI+UTMPR5) * UTMPR4
          DIS = DIS/UTMPR4
C
C               INPUT THRMISTR DATA, SET TEMPR. RAYNJ
          CALL OPEN (5, 'D24W.DATI', 'I', KI)
     260  READ (5,269) (AN(N), N=1,5)
     269  FORMAT (// 29X, 5A4)
     270  READ (5,279) NLIST, DUM
     279  FORMAT (7X, F8.0, 2X, E11.0)
          NA = 0
      20  DO 29 N = 1,NLIST
     271  READ (5,279) TTH, RTH
      40  IF (TTH .GE. 9.876543E21)  GO TO 49
      50  IF (TTH .LT. TLO) GO TO 59
      60  IF (TTH .GT. THI) GO TO 69
          NA = NA + 1
          T(NA) = TTH
          RT(NA) = RTH
      59  CONTINUE
      69  CONTINUE
      29  CONTINUE
      49  CONTINUE
          NHI = NA
C
```

Figure 8 -- THMSTI program text (page 1 of 6).

THMSTI.F4 -p. 2/6

```
C           FYND MIDL VALEWZ
      TMD = (T(1)+T(NHI)) / 2.
  460 DO 469 N=1,NHI
  470 IF (T(N) .GE. TMD)  GO TO 479
  469 CONTINUE
  479 CONTINUE
      NMD = N
      RTMD = RT(N-1) + (TMD-T(N-1)) * (RT(N)-RT(N-1))
&      / (T(N)-T(N-1))
      NHI = NHI + 1
  480 DO 489 N=NHI,NMD,-1
      RT(N) = RT(N-1)
      T(N) = T(N-1)
  489 CONTINUE
      RT(NMD) = RTMD
      T(NMD) = TMD
C
C           KALK, PRINT THRMISTR PARAMETRS
      LN = ALOG( RT(NHI)/RT(1) )
      BETA = T(1) * T(NHI) / (T(1)-T(NHI)) * LN
      RINF = RT(NMD) * EXP(-BETA/T(NMD) )
      CALL OPEN (6,  'TTY:!', 'O', KO)
   70 WRITE (6,79) (AN(N),N=1,10), DIS, T(1), T(NHI), RINF, BETA
   79 FORMAT (/ 40H THERMISTOR CHARACTERISTICS (LINEARIZED)
&      / 27H ---------------------------
&      // 15H   FOR THE TYPE , 10A4, 11H THERMISTOR
&      /31H   WITH A DISAPATION CONST. OF , F8.5, 7H WATTS/
&      6HKELVIN/ 28H   OVER THE TEMPR. RANGE OF  , F6.1, 4H TO
&      , F6.1, 7H KELVIN
&      // 19H     R(THRMSTR)   = , 1P1E10.3, 7H * EXP( , 0P1F7.1,
&      18H KELVIN / T ) OHMS)
C
C           LOG APROX.
      AK = RT(1) / RT(NHI) * ADJAK
      X = ALOG(AK)
      AL = 0.49825 + X*(2.94725E-3 + X*(-1.28465E-3))
      AM = 0.25004 + X*(2.86052E-4 + X*(-4.53302E-3 +
&           X*4.01613E-4))
C
C           SET INISHAL R2
      R2 = RT(NMD) * 0.75
      R2FRST = R2 * 1.01
C
C           STEP R2
  100 DO 109 K=1,500
      R2P = R2
      R2 = R2 * 1.01
      ADJBA = ADJBAF
      ADJMA = ADJMAF
C
C           SAYV REEZULTS UV BEST R2
  380 IF (TER .GT. TERPRE) GO TO 389
      R2PRE = R2P
      R1PRE = R1
      MAPRE = MA
      BAPRE = BA
      NOPPRE = NOP
      KRVPRE = KRV
  340 DO 349 I=1,NHI
      TEPRE(I) = TE(I)
  349 CONTINUE
      TERPRE = TER
      NPOST = 0
  590 GO TO 599
C
```

Figure 8 (cont.) -- THMSTI (p 2/6).

THMSTI.F4
- p. 3/6

```
  389 CONTINUE
      NPOST = NPOST + 1
  390 IF (NPOST .GE. 11) GO TO 399
C
  599 CONTINUE
C
C             KALK. SKT. GAYN
      R1 = R2 * ( AM * ALOG( R2/RINF ) - AL )
  110 DO 119 N=1,NHI
      RP = R2 * RT(N) / (R2+RT(N))
      A(N) = R1 / (R1+RP)
  119 CONTINUE
C
C             KALK. STRAYT LYN APROX.
      MA = (T(NHI) - T(1)) / (A(NHI) - A(1))
      BA = T(1) - MA*A(1)
C
C             ADJ. TOO MINIMYZ EROR
      ALIMTE = 0.01
      ALIMTH = 0.1
      NOP = 0
  339 CONTINUE
      TELMX = 0.
      TELMN = 0.
      TEHMX = 0.
      TEHMN = 0.
      NOP = NOP + 1
C
C             KALK. EROR @ EECH POINT
  120 DO 129 N=1,NHI
      TMEZH = MA * A(N) + BA
      TACTL = T(N)
      TER = TMEZH - TACTL
      TE(N) = TER
  129 CONTINUE
C
C             FYND MAX. LOE RAYNJ ERORS
  310 DO 319 I=1,NMD
  500 IF (TELMX .GT. TE(I))  GO TO 509
      TELMX = TE(I)
      NLMX = I
  509 CONTINUE
C
  510 IF (TELMN .LT. TE(I))  GO TO 519
      TELMN = TE(I)
      NLMN = I
  519 CONTINUE
C
  319 CONTINUE
C
C             FYND MAX. HY RAYNJ ERORS
  320 DO 329 I=NMD,NHI
  520 IF (TEHMX .GT. TE(I))  GO TO 529
      TEHMX = TE(I)
      NHMX = I
  529 CONTINUE
C
  530 IF (TEHMN .LT. TE(I))  GO TO 539
      TEHMN = TE(I)
      NHMN = I
  539 CONTINUE
C
  329 CONTINUE
C
```

Figure 8 (cont.) -- THMSTI (p 3/6).

THMSTI.F4 -p. 4/6

```
C               KLASIFY BY KRV SHAYP
      540 IF (TEHMX-TEHMN .LT. TELMX-TELMN)  GO TO 549
          N3 = NHI
      550 IF (TE(NHI) .GT. TE(NHI-1))  GO TO 559
          N1 = NLMN
          N2 = NHMX
          KRV = 1
      560 GO TO 569
C
      559 CONTINUE
          N1 = NLMX
          N2 = NHMN
          KRV = 2
      561 GO TO 569
C
      549 CONTINUE
          N3 = 1
      570 IF (TE(1) .LT. TE(2))  GO TO 579
          N1 = NHMX
          N2 = NLMN
          KRV = 3
      562 GO TO 569
C
      579 CONTINUE
          N1 = NHMN
          N2 = NLMX
          KRV = 4
      569 CONTINUE
C
C               TEST FOR LEEST MAX. EROR
          TELIM = ALIMTE * AMIN1(ABS(TE(N1)), ABS(TE(N2)))
      130 IF ((ABS(TE(N1)+TE(N2)) .LE. TELIM) .AND.
&              (ABS(TE(N3)+TE(N2)) .LE. TELIM))  GO TO 139
C
C               PRFORM ADJUSTMENT
          BA = BA - (TE(N2)+TE(N1)) /2. * ADJBA
          ADJBA = ADJBA * 0.9999
          MAOELD = MA
          MA = MA - (TE(N3)-TE(N1)) / (A(N3)-A(N1)) * ADJMA
          BA = BA + (MAOELD-MA) * A(NMD)
          ADJMA = ADJMA * 0.9999
      330 GO TO 339
      139 CONTINUE
          TER = AMAX1( ABS(TE(N1)), ABS(TE(N2)) )
      109 CONTINUE
C
C               KALK. SELF HEETING EFEKT
      399 CONTINUE
          R2LAST = R2
          THER = ALIMTH * TER
          EI = 1.0E+20
          R1 = R1PRE
          R2 = R2PRE
      150 DO 159 N=1,NHI
          R = RT(N)
          RDIS = ( R1*R + R1*R2 + R*R2 )/R2
          E = RDIS * SQRT(THER*DIS/R)
      160 IF (E .GE. EI)  GO TO 169
          EI = E
      169 CONTINUE
      159 CONTINUE
C
```

Figure 8 (cont.) -- THMSTI (p 4/6).

Again delimited by commas. The heading has 4 lines:

1) a row of zeros to help in starting the tape punch,

2) file identification & date,

3) manufacturer and part number, and

4) a count of the qty. of data points.

The last line in the file contains the 8.765E21 value which can be used by the reading program as an end of file delimiter.

7d. <u>THMSTI</u> (Fig. 8)

... is a real biggie! It begins by asking for temperature range and dissipation constant (1130-1160), and converting them to Kelvin (1170-1190). D24W.DAT is then read (1280-1290) with only those values lying in the specified range being stored (1300-1350). In the process, D24W.DAT's heading is also read and held for later printout (1230-1260).

The middle values TMD & RMD of the range are found (1430-1470) using linear interpolation if needed (1490-1500). The result is jammed into the middle of the Array (1510-1570).

Using the LO and HI Array values, thermistor parameters are calculated (1600-1620) and printed (1630-1720).

Here, everything begins to happen at once. R_2 is incremented in 10pp K steps (1860-1880) beginning at 0.75 of the thermistor's mid range value (1820). Then R_1 is calculated (2140) using constants obtained from a polynomial form (1750-1790) of the log approximation (ref. 1). Next, this R_1 and R_2 are used to calculate the circuit gain at each of the thermistor points (2150-2180).

There follows a lenghty procedure of finding the best straight line approximation to these gains. Essentially, initial values of slope MA & intercept BA are assumed (2210-2220). The errors are calculated (2360-2410), a corrective strategy is decided upon depending on the curve shape (2440-3030), corrections are made (3060-3110), and the process repeated (3120, 2280).

When a comparison of errors shows that a best fit is obtained (3010-3030, 3140), the process is repeated from the top with the next value of R_2 (3150, 1860). But before the calculation is repeated, the error obtained

THMSTI.F4
-p.5/6

```
C         PRINT OWT POUT
      MAPRE = MAPRE / UTMPR4
      BAPRE = BAPRE/UTMPR4 - UTMPR5
      TERPRE = TERPRE / UTMPR4
      TEHT = ALIMTH * TERPRE
  300 WRITE (6,309) K, R2FRST,R2LAST, KRVPRE, NOPPRE, TERPRE
&      ,UTMPR1,UTMPR2,UTMPR3, TEHT, UTMPR1,UTMPR2,UTMPR3, EI
  309 FORMAT (// 30H RESULTS OF BEST LINEARIZATION / 1H , 29(1H-)
&      // I5, 37H VALUES OF RESISTANCE HAVE BEEN TRYED
&      / 16H      FROM R(2) = , 1P1E9.2
&      / 7H     TO ,9X, E9.2, 6H OHMS.
&      /17H     CURVE TYPE  , I2
&      / I4, 37H ITERATIONS WERE NEEDED TO GET A BEST
&      / 17H      MAX. EROR OF  , E9.2, 1X, 3A4
&      / 31H    TO HOLD SELF HT. EROR WITHIN , E9.2, 1X, 3A4
&      / 12H      E(IN) = , E9.2, 14H VOLTS OR LES.
&      / 51H      WHAT VALUE DO YOU SELECT FOR E(IN)?  (VOLTS)  /)
C
C         KALK, PRINT SPESIFIK REEZULTS
      ACCEPT, EI
      MA = MAPRE/EI
  370 WRITE (6,379) UTMPR1,UTMPR2,UTMPR3, EI, R1PRE,R2PRE
&      , MA, BAPRE
  379 FORMAT (/ 29H POINT BY POINT PERFORMANCE ( , 3A4, 1H)
&      / 27H ---------------------------
&      // 15H    IF:  E(IN) =, 1P1E9.2, 6H VOLTS
&      / 14H         R(1) =, E10.3, 5H OHMS
&      / 14H         R(2) =, E10.3, 5H OHMS
&      // 8H    THEN: / 20H     T(ACTUAL)  =  (, 0P1F10.3
&      ,12H)*E(OUT) + ( , F10.3, 1H)
&      / 18X, 25H- T(APROX.) - T(SELF HT.)
&      // 11H      WHERE:
&      / 28H          AT      E(OUT) IS
&      , 26H   AND THE EROR TERMS ARE:
&      / 28H        T(ACTUAL),    (VOLTS)
&      , 26H   T(APROX.)  T(SELF HT.)   / 4H    , 4(3X,9(1H-)))
C
C         KALK. SELF HT., PRINT ERORS
      R1 = R1PRE
      R2 = R2PRE
  350 DO 359 N=1,NHI
      R = RT(N)
      NLMN = I
      RDIS = ( R1*R + R1*R2 + R*R2 )/R2
      TEHT = R * EI/RDIS * EI/RDIS / DIS / UTMPR4
      EO = A(N) * EI
      TO = T(N) / UTMPR4 - UTMPR5
      TEPRE(N) = TEPRE(N) / UTMPR4
  360 WRITE (6,369) TO, EO, TEPRE(N), TEHT
  369 FORMAT (1H , 5X, F9.3, 3X, 1P1E10.3, 2(3X, 0P1F9.4))
  359 CONTINUE
C
C         KALK. SENSITIVITEE ERORS
  420 DO 429 K=1,3
      IF (K .EQ. 1) L = 1
      IF (K .EQ. 2)  L = NMD
      IF (K .EQ. 3) L = NHI
      DENOM = ( R1*R2 + R2*RT(L) + RT(L)*R1 )
      DTEO = MA
      DAR1 = R2 * RT(L) * A(L) / DENOM
      DAR2 = -R1 * R2 * (RT(L)/DENOM)**2
      DART = -R1 * RT(L) * (R2/DENOM)**2
      DABETA = DART * (BETA/T(L))**2
```

Figure 8 (cont.) -- THMSTI (p 5/6).

THMSTI.F4
-p.6/6

```
      AN(10) = '  R('
      AN(11) = 'INF)'
      AN(11+K) = DTEO * EI * DART * 0.001
      AN(20) = '  BE'
      AN(21) = 'TA  '
      AN(21+K) = DTEO * E * DABETA * 0.001
      AN(30) = 'R(T)'
      AN(31) = '    '
      AN(31+K) = DTEO * EI * DART * 0.001
      AN(40) = 'R(1)'
      AN(41) = '    '
      AN(41+K) = DTEO * EI * DAR1 * 0.001
      AN(50) = 'R(2)'
      AN(51) = '    '
      AN(51+K) = DTEO * EI * DAR2 * 0.001
      AN(60) = 'E(IN'
      AN(61) = ')   '
      AN(61+K) = DTEO * A(L) * 0.001
  429 CONTINUE
C
C          PRINT SENSITIVITEE ERORS
  430 WRITE (6,439) UTMPR1,UTMPR2,UTMPR3
  439 FORMAT (// 20H SENSITIVITY ERORS
&      , 28H("PPK" = PARTS PER THOUSAND)  / 1H , 17(1H-)
&      // 12(1H ), 37HA +1. PPK    WIL PRODUCE AN ADITIONAL
&      /  12(1H ), 37HCHANGE IN     EROR IN MEASURED TEMPR.
&      / 12(1H ), 9(1H-), 8X, 4HOF (, 3A4, 1H)
&      /  12(1H ), 37H THRMSTOR    LO END   MIDLE    HI END
&      /  12(1H ), 37H CONSTS.:    ------   ------   ------ )
  440 DO 449 K=1,6
  450 WRITE (6,459) (AN(N),N=(10*K),(10*K+4))
  459 FORMAT (13(1H ) , 2A4, 1X, 3(2X, F7.4))
  449 CONTINUE
C
C          REE SYKL
  400 WRITE (6,409)
  409 FORMAT (//// 15H            /
&      38H YOU HAVE WON A FREE GAME!  TRY AGAIN.  )
  410 GO TO 419


READY
LEN
12045  CHARACTERS
READY
REPLACE
READY
```

Figure 8 (cont.) -- THMSTI (p 6/6).

with the last R_2 is compared with that from the previous best R_2 (1930). If it is less, then the significant parameters of the last R_2 calculations are saved (1940-2030).

At first, the R_2's chosen cause the linearity error to decrease on successive tries. Eventually, an R_2 is reached where the error begins to increase. When this happens, 10 more values of R_2 (2080-2090) are tried to make sure that the minimum error R_2 was indeed a true minimum and not a round off freak.

At this point, the optimum circuit has been found, so the bulk of the computation is finished. Those steps which remain are straight through operations.

Next, the effect of self heating is calculated. At each thermistor data point, the E_{in} which gives a self heat error of 1/10 the best linearity is calculated (3240-3270), and the lowest value saved (3280-3290).

The results of the calculations so far are then printed out (3380-3480), giving

1) the R_2s tried,

2) qty. of iterations needed for best value,

3) worst linearity error, and

4) max. E_{in} to hold self heating error at 1/10 linearity error.

the program asks for your choice of E_{in} (3490-3520), and then continues on its merry way.

Next printed are the circuit parameters and the transfer equation (3540-3630). Then, for each thermistor point, E_{out}, $T_{approx.}$, and $T_{self\ ht.}$ are calculated (3730-3800) and printed (3810). Finally, the sensitivity factors are calculated (3860-4140) and printed (4180-4280).

The recycle option (4310-4340) branches back to the top (1040), completing the program.

THE END

This page added for protection in filing and handling.