ENGINEERING NOTE

TUNING THE PID COEFFICIENTS

Sun Electronic Systems, Inc.

1900 Shepherd Dr., Titusville, FL 32780
Tel: (407) 383-9400 Fax: (407) 383-9412

PURPOSE

The purpose of this engineering note is to provide the user of the TC01 and the TC10 digital temperature controllers with the necessary information to adjust or “tune” the control loop coefficients. By tuning the controller coefficients the user can optimize the overall control system response to a wide range of processes.

DEFINING FEEDBACK CONTROL

Control systems can be classified as “open loop” or “closed loop”. Figure 1 shows the block diagram of an open loop type of control system. In this type of system the input is processed by the controller and then drives the process, producing the output. The problem with the open loop system is that the output varies as the controller or process transfer function changes. For example if the controller were a simple amplifier and its gain changed by 10% then the output would also change by 10%, assuming that the process is linear. The open loop system, however, does have a couple of advantages. First is simplicity and second is stability.

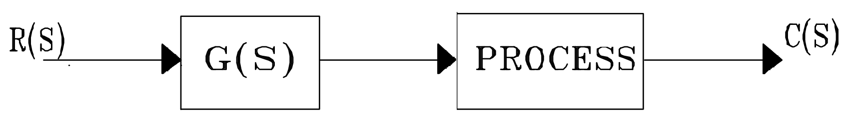


FIGURE 1

Figure 2 gives the block diagram for a closed loop negative feedback control system. The prime difference between the open and the closed loop systems is the generation and utilization of the error signal E(S). The primary reason for using the closed loop control system is to reduce the sensitivity of the system to parameter variations. In order to illustrate the effects of parameter variations, let us consider a change in the transfer function G(S) such that we have G(S)+delta G(S). Then in the open loop case, the change in the transform of the output is

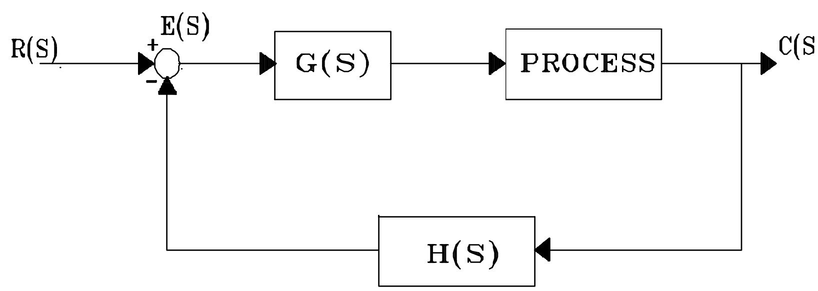


FIGURE 2

$$C(S) = G(S)R(S)$$

In the closed loop system we have

$$C(S) = G(S)R(S) / [\ 1+G(S)H(S)\ ]$$

or $$C(S)/R(S) = G(S) / [\ 1+G(S)H(S)\ ]$$

Taking partial derivative with respect to G(S)

$$\frac{\delta[C(S) / R(S)]}{\delta [G(S)]} = \frac{[1+G(S)H(S)]G(S) - G(S)G(S)H(S)}{[1+G(S)H(S)]^{**}2}$$

$$\frac{\delta [C(S) / R(S)]}{\delta[G(S)]} = \frac{[G(S)[1+G(S)H(S) - G(S)H(S)]}{[1+G(S)H(S)]^{**}2}$$

$$\frac{\delta[C(S) / R(S)]}{\delta[G(S)]} = \frac{G(S)}{[1+G(S)H(S)]^{**}2}$$

or the change in the transfer function of the closed loop system is reduced by the factor [1+G(S)H(S)]**2 which is usually much greater than one over the range of complex frequencies of interest.

In a closed loop temperature control system, as shown in figure 3, a temperature sensor measures the chamber temperature. The feedback signal from the sensor is applied to the error detector which compares the desired temperature signal, or set point, with the feedback signal from the sensor. If the desired and the actual temperatures are not the same, the error detector drives the controller to increase or decrease the heat/cool output as appropriate. A closed loop system thus operates to reduce the error signal to zero so that the process temperature and the set temperature are equal.

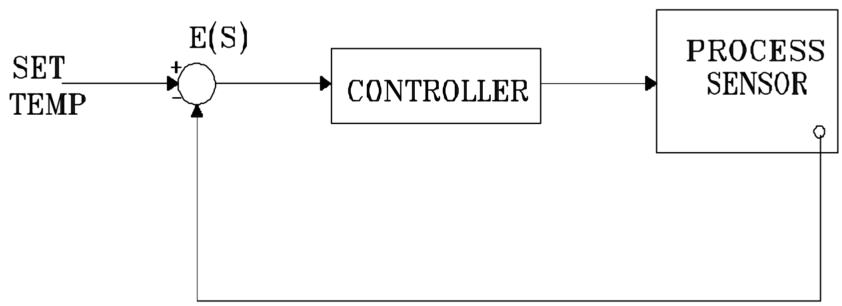


FIGURE 3

CONTROL SYSTEM RESPONSE

The manner in which the system responds to a new set point, or an external disturbance, is referred to as the control systems response. The two most common disturbances used to test control systems response are a step input (A/S) or a ramp input (A/S**2). Depending upon the application, a variety of system responses can be designed into a controller by changing the foward loop coefficient. Microprocessor driven digital control systems offer the flexibility of setting the controller response through software commands, providing the capability of tailoring the system precisely to a given application.

Proportional response is the most basic controller response characteristic and is a component of most control systems. In a proportional only system, the corrective action is always proportionate to the difference between the desired output and the set point. In other words, as the error becomes smaller the corrective action reduces. Figure 4 curve A shows the response of a proportional only control system to a step input. Note that the proportional response results in an offset or continuous error since the error signal approaches zero as the error approaches zero. The offset error can be reduced by increasing the proportional gain, however, there is an upper limit to the loop gain that will still provide a stable system.

In many systems an offset error cannot be tolerated. One way to eliminate the error is to modify the controller response so it also responds to how long the error is present. Such a response is an integral response and combined with proportional results in an overall system response as illustrated in figure 4 curve B. As can be seen the integral will remove the offset error.

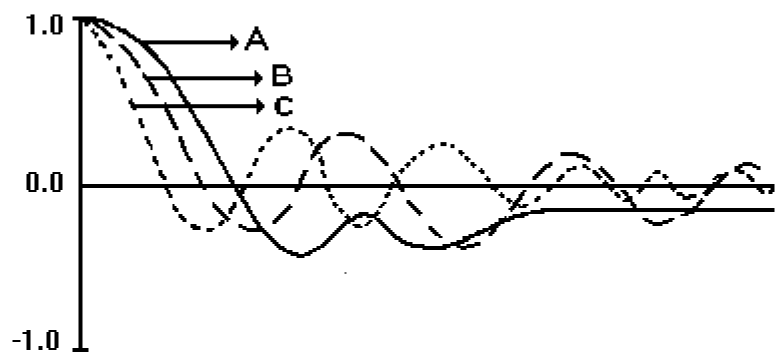


FIGURE 4

A further reduction in settling time and overshoot, or increased stability, can be obtained by designing the controller to respond to how fast the error is changing. A response to how fast the error is changing is called a rate or derivative response.

The result of combining the proportional, integral, and derivative responses is illustrated in figure 4 curve C. This form of controller response is called a PID response. As shown, a properly designed PID response provides the shortest settling time and minimum overshoot of the three responses shown. Digital controllers use the PID control algorithm, and allow the user to set the P,I, and D coefficients. Thus the user can tune the PID coefficients for an optimum control system response.

The question now is " How do I pick the PID coefficients? " To address the question we will next present a simple iterative approach and then provide a mathematical approach.

ITERATIVE PID SELECTION

Since mathematically modeling the thermal control system requires that the thermal coefficients of the process and sensor be determined, which in many circumstances is an undesirable use of time, the following iterative approach is presented first.

Iterative PID Selection for TC01

Some temperature processes provide different amounts of energy during heating and cooling. For the TC01, a compromise must be reached when tuning the PID coefficients for heating and cooling. The proportional, integral and derivative coefficients consist of three PID base values and three PID weighting coefficients. It is the three weighting coefficients that may be modified via the PID change command. The user changeable weighting coefficient is 2 raised to the power of a number [$2^{(number)}$]. For example, the actual proportional coefficient(Kp) equals the base fixed value times the user changeable weighting coefficient, i.e. $Kp=(K_{fixed})(2^{Pnumber})$

1. Start by finding a proportional coefficient that provides a stable system. The integral and derivative coefficients should be disabled by entering weighting coefficients of -9 . Note that there will be a steady state error that should be ignored for the present.

example bus command: PID = -1, -9, -9

proportional weighting= 2**(-1) = .5

integral weighting= 2**(-9) = .001953 ? 0.0

derivative weighting= 2**(-9) = .001953 ? 0.0

2. Next bring up the integral weighting until the steady state error is reduced to an acceptable value in a reasonable amount of time. If, by increasing the integral weighting, the system becomes unstable then reduce the proportional weighting slightly.

example bus command: PID = -1, -3, -9

proportional weighting= 2**(-1) = .5

integral weighting= 2**(-3) = .125

derivative weighting= 2**(-9) = .001953 ? 0.0

3. Finally bring up the derivative weighting to reduce any overshoot and to increase stability.

example bus command: PID=-1,-3,-1

proportional weighting= 2**(-1) = .5

integral weighting= 2**(-3) = .125

derivative weighting= 2**(-1) = .5

Iterative PID Selection for TC10, PC1000, TC02, PC100, PC100-2 and PC10

The TC10 has two sets of PID coefficients, one for heating and one for cooling. The proportional, integral and derivative coefficients are entered as simple floating point numbers and may be modified via the PID change command. The following example shows how to select the PID coefficients for heating. A similar approach can be followed for selecting the PID coefficients for cooling.

The heater element is full on when P*ABS(error)>1 and the cooling system is full on when P*ABS(error)<-1; where,
P = Proportional coefficient for heat or cool
error = Difference between the set temperature and the process temperature

1. Start by finding a proportional coefficient that provides a stable system. The integral and derivative coefficients should be disabled by entering 0. Note that there will be a steady state error that should be ignored for the present.

example bus command: PIDH = 0.25, 0.0, 0.0

proportional coefficient = 0.25

integral coefficient = 0.0

derivative coefficient = 0.0

2. Next bring up the integral coefficient until the steady state error is reduced to an acceptable value in a reasonable amount of time. If by increasing the integral coefficient the system becomes unstable then reduce the proportional coefficient slightly.

example bus command: PIDH = 0.25, 0.001, 0.0

proportional coefficient = 0.25

integral coefficient = 0.001

derivative coefficient = 0.0

3. Finally bring up the derivative coefficient to reduce any overshoot and to increase stability.

example bus command: PIDH = 0.25, 0.001, 0.10

proportional coefficient = 0.25

integral coefficient = 0.001

derivative coefficient = 0.10

In the above iterative approach the step input is performed by instructing the controller to go to a temperature that is a significantly different than its current temperature, with the Rate set to a high value. The time response is obtained by monitoring the process temperature. The optimization can be performed manually or with a remote bus controller by plotting the process temperature versus time after the temperature step is given.

LINEAR CONTINUOUS MODEL

Figure 5 shows the block diagram of a continuous data, linear PID control system acting on an error signal E(S). The proportional block simply multiplies the error signal by a constant KP, the integral block multiplies the integral of E(S) by the integral coefficient KI, and the derivative block multiplies the time derivative of the error signal by the derivative coefficients KD. The KH block depicts the function of conversion from the PID output to heat flow rate into the temperature chamber. The Gp1(S) block represents the transfer function of the chamber while the Gp2(S) block represents the transfer function of the temperature sensing probe.

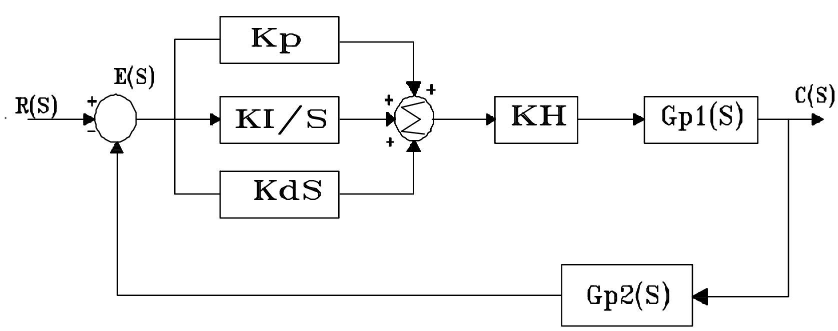


FIGURE 5

Figure 6 shows the diagram of a typical temperature chamber with the simplified equivalent circuit where:

$$Ct = Q / Tc$$

or $\quad qc(t) = Ct\,(d\,Tc) / dt$

also $\quad Rt = (Tc - Ta) / qo(t)$

or $\quad qo(t) = (Tc - Ta) / R$

differential equation for the chamber is:

$$qi(t) = qc(t) + qo(t)$$

or $\quad qi(t) = [Ct\,(d\,Tc) / dt] + [(Tc - Ta) / Rt]$

and its Laplace transform is:

$$Qi(S) = Ct(S)Tc(S)S + [Tc(S) - Ta(S)] / Rt(S)$$

or $\quad Qi(S) = [CtTcRtS + Tc - Ta] / Rt$

or $\quad Qi(S)Rt = Tc(CtRtS + 1) - Ta$

therefore,

$$Tc = [Qi(S)Rt + Ta] / [CtRtS+1]$$

where $\quad Qi(S) = (Kp + Ki/S + KdS)Kh$

the Laplace transfer function of chamber is:

$$G(S) = Tc = \frac{[KpKhRtS + KiKhRt + KdKhRtS^2 + TaS]}{[S(CtRtS+1)]}$$

Figure 7 shows the equivalent circuit for the temperature sensing probe where:

$$Tp =(1/Cp)\ \ [q(t)dt]$$

and $\quad q(t) = [Tc - Tp] / \int Rp$

and its Laplace transform is:

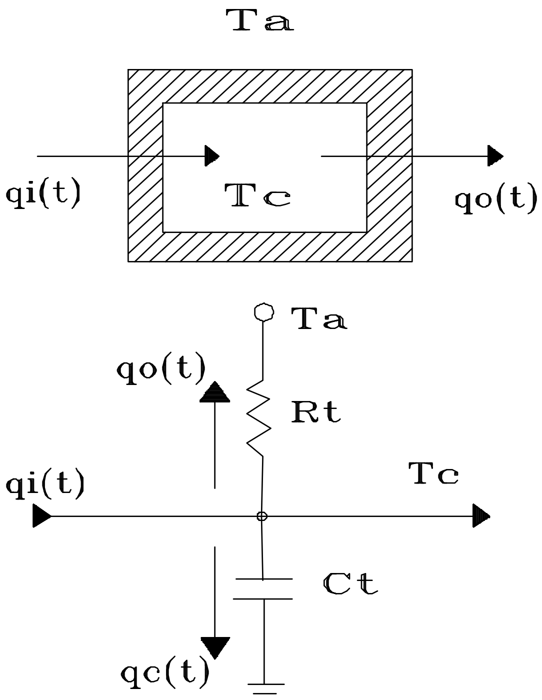
$$Tp(s) = [Tc - Tp]/[CpRpS]$$

where $\quad CpRp = tp$

or $\quad Tp =Tc / [RpCpS+1]$

the Laplace transfer function of probe is:

$$H(S) = Tp / Tc = 1 / [tpS+1]$$



Tc = Chamber Temperature

Ta = Ambient Temperature

qi(t) = Input Heat Flow Rate

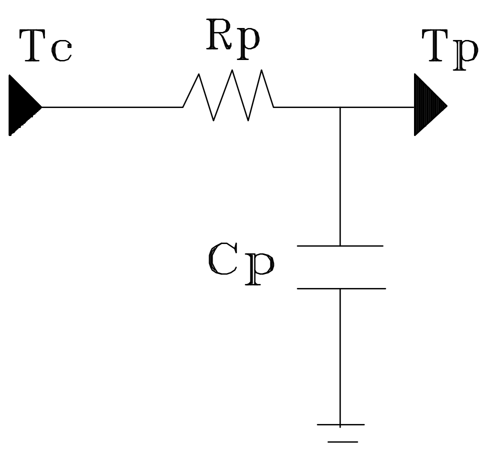
qo(t) = Heat Loss Flow Rate

qc(t) = Heat Flow Rate

Rt = Thermal Resistance of Chamber to Ambient

Ct = Thermal Capacitance of Chamber

FIGURE 6



Tc = Actual Chamber Temperature

Tp = Indicated Chamber Temperature

Rp = Thermal Resistance of Probe

Cp = Thermal Capacitance of Probe

FIGURE 7

Now we will construct the closed loop error equation. The error signal for a closed loop system is:

$$E(S) = R(S)/[1+G(S)H(S)]$$

where $R(S) = 1/S$ for step input

and $Tc = CtRt$

so

$$E(S) = \frac{(1/S)}{\dfrac{S(CtRtS+1)(tpS+1) + KpKhRtS+KiKhRt+KdKhRtS^2 + TaS}{S(CtRtS+1)(tpS+1)}}$$

$$= \frac{(CtRtS+1)(tpS+1)}{tpCtRtS^3+tpS^2+RtCtS^2+S+KpKhRtS+KdKhRtS^2+KiKhRt+TaS}$$

$$= \frac{tpCtRtS^2+CtRtS+tpS+1}{S^3(tpCtRt)+S^2(tp+CtRt+KdKhRt)+S(1+KpKhRt+Ta)+KiKhRt}$$

$$E(S) = \frac{S^2(Tctp) + S(Tc+tp) + 1}{S^3(Tctp)+S^2(tp+Tc+KdKhRt)+S(1+KpKhRt+Ta)+KiKhRt}$$

dividing by TcTp gives,

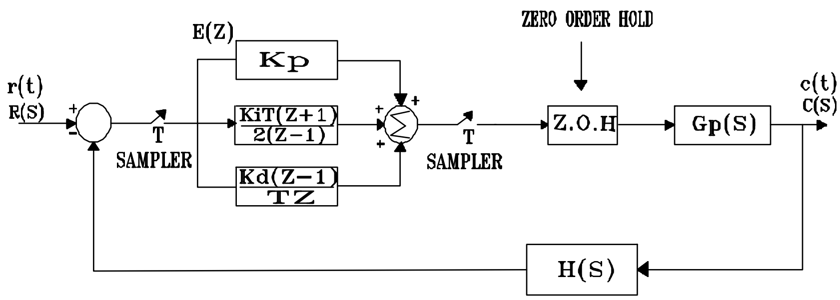
$$E(S) = \frac{S^2 + S\left({}^{1}/_{tp} + {}^{1}/_{Tc} \right) + {}^{1}/_{Tctp}}{S^3+S^2\left({}^{1}/_{Tc} + {}^{1}/_{tp} + {}^{KdKhRt}/_{Tctp}\right)+S\left({}^{(1 + KpKhRt + Ta)}/_{Tctp} \right)+ {}^{KiKhRt}/_{Tctp}}$$

The time response of the control system maybe calculated by finding the roots of the denominator of E(S), performing partial fraction expansion, and then transforming back to the time domain. A Fortran program, called CHPID, that does exactly this is included at the end of this note. To illustrate the effect of the PID coefficients on the time response, figure 8 shows the response of a proportional only system. Note the steady state error. Figure 9 gives the time response with the same proportional coefficient and the addition of the integral coefficient. Note that the steady state error has been reduced to zero. Finally in figure 10 and 11 the derivative coefficient is added to reduce overshoot in the time response.

At this point it is necessary to point out that the model presented assumes a linear, continuous, time invariant control system. However, the PID control in the TC01 and TC10 is implemented digitally. Also, there is usually a fixed limit to the amount of energy that can be supplied to the temperature chamber, which causes limiting to occur in the control loop. Therefore the continuous linear constraints of the model are not realized. The accuracy of this model is dependant upon the rate of change of the process temperature relative to the sampling rate of the controller and to the amount of limiting that occurs. Of the two contributors to inaccuracy of this model, since most temperature processes are slow, the limiting of energy to the process is usually the most influential. With these limitations in mind, this model does show the effect of the PID coefficients on the time response of the control system.

NONLINEAR DISCRETE TIME MODEL

To accurately model the sampled data control of the TC01 and TC10 we will add a sampler before the PID blocks and a sampler and zero order hold after the PID section. Figure 12 shows the block diagram of the improved model. Note also that we have replaced the Laplace transform I and D blocks with Z transform approximations. In TC01 and TC10 it is these approximations to integration and differentiation that are actually used. At this point we now have a model that accounts for the digital implementation of the PID control section. The only remaining problem is to model the nonlinearity of the system.

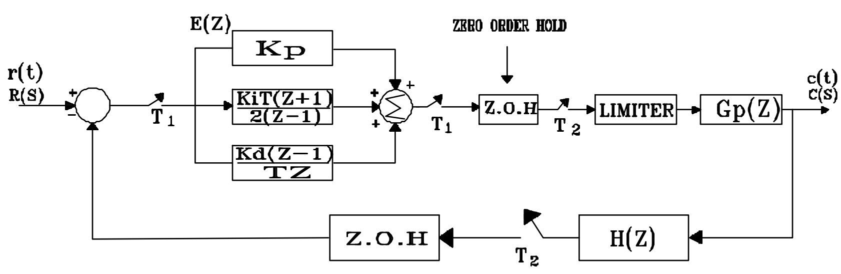


$$\frac{KiT(Z+1)}{2(Z-1)} = \text{POLYGONAL INTEGRATION APPROXIMATION TO } KI/S$$

$$\frac{Kd(Z-1)}{TZ} = Z\left[\frac{de(t)}{dt}\right] * Kd$$

FIGURE 12

Since Laplace and Z transforms require a linear system, we will now analyze the system via a simulation program. Figure 13 provides the final model. The major difference between the models in figure 12 and 13 is the replacement of the Gp(S) and H(S) in figure 12 with a new HpHs(Z) equivalent model and the addition of two samplers, a zero order hold, and limiter. The new GpHs(Z), samplers, and zero order hold are implemented in the simulation program via numerical approximation to the chamber and probe differential equations. As shown in the figure there are two sampling times used.



WHERE,

$T1 = nT2$

FIGURE 13

Sample time T1 is the sampling rate while sample time T2 is selected sufficiently faster than T1 to allow the use of straight foward numerical methods. A Fortran source "CHSIM" listing is included at the end of this note. Once the chamber's thermal capacitance "CT" and thermal resistance to ambient "RT", the sensing probe's actual time constant "TP" and the maximum heat flow rate "KH" are determined for the process of interest, the simulation program can be used to pick the required PID coefficients. The simulation program is also useful for determining the sensitivity of the control system to changes in the thermal coefficients.

THERMAL COEFFICIENTS

In this section we will show a quick method of determining the approximate thermal capacitance and the thermal resistance of a chamber. A basic sample program on page 16 can be used to determine an approximate thermal capacitance of a chamber. The two following examples were actual tests run on a 0.7 cu ft chamber, and by using the appropriate equations we calculated the thermal coefficients of this particular chamber.

Figure 14 shows the simplified equivalent circuit of the temperature chamber.

Example 1 : Thermal Capacitance (Ct)

To = initial chamber temp = 24.3 ºC

T = final chamber temp = 37.32 ºC

delta temp. = (T - To)

t = test time in seconds = 60 s

qc = heat flow rate in cal/s

Qc = heat flow in cal

Kh = heater gain constant

voltage applied = 115 V

current = 12 A

power = (Voltage)(Current)

= (115 V)(12 A)= 1.380 KW

1 watt = 1 joule/s = .239 cal/s

252 cal = 1054.37 Joules = 1 btu

q = (.239)(1380) = 329.82 cal/s

Kh = 329.82 cal/s = 1380 J/s = 1.31 btu/s

Q = (329.82)(60) = 19789.20 cal

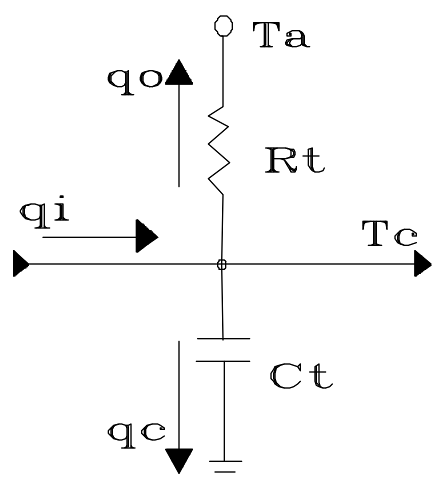


FIGURE 14

Referring to figure 14,

Ct = Total heat flow / delta temp. = Q / (T - To)

= 19789.20 / (37.32 - 24.3)

or Ct = 1519.91 cal/ºC = 6359.30 J/ºC = 6.03 btu/ºC

Example II : Thermal Resistance (Rt)

In this case, continuous power was supplied to the chamber and the chamber was allowed to reach thermal equilibrium.

Ta = ambient temp = 25 ºC

Tc = chamber temp at equilibrium = 150 ºC

qo = heat flow rate in cal/s

voltage applied = 60 V

current = 3 A

power = (60 V)(3 A) = 180 W

1 watt = 1 Joule/s = .239 cal/s

252 cal = 1054.37 joules = 1 btu

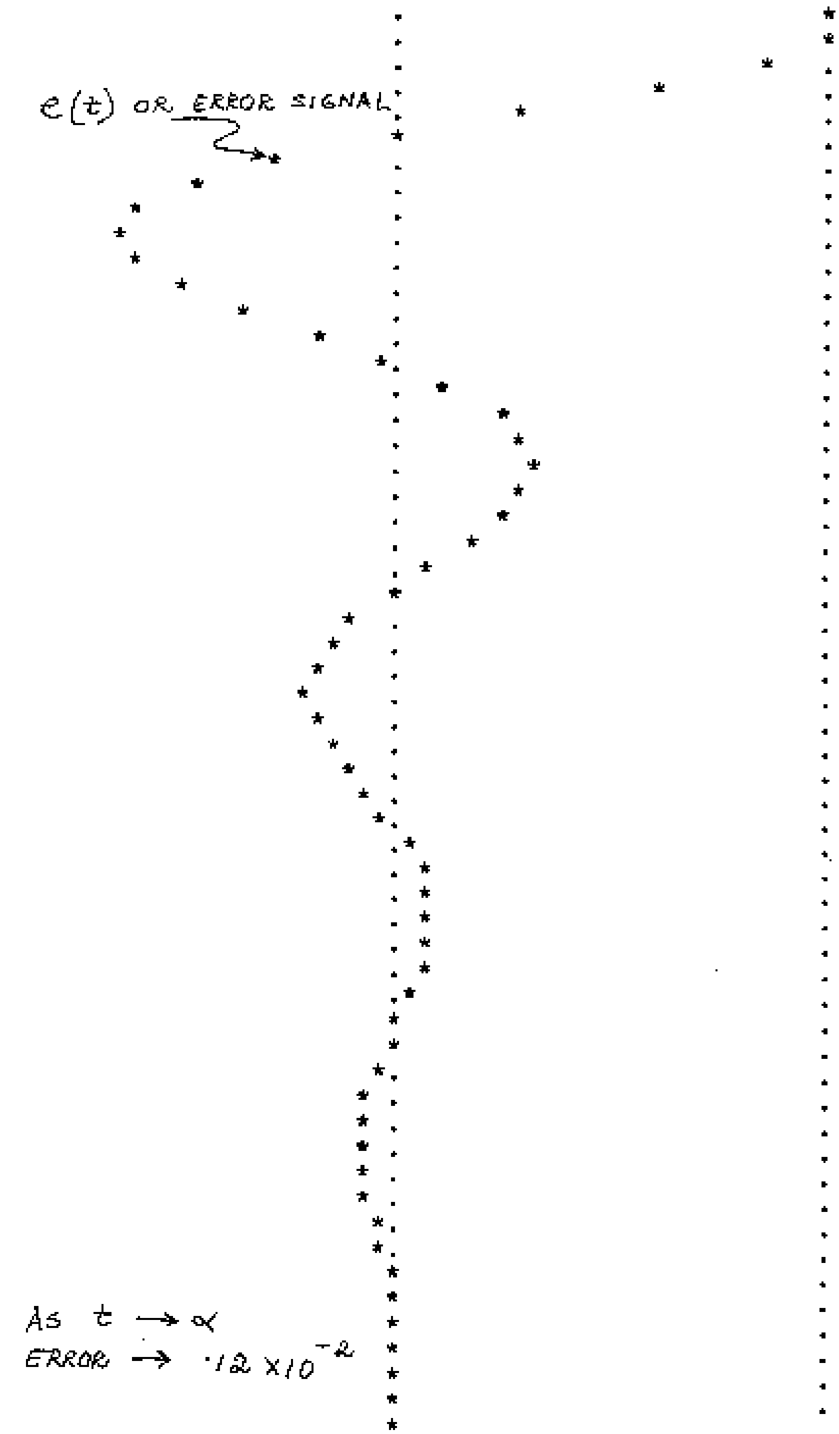
qo = (.239)(180) = 43.02 cal/s

Referring to figure 14,

Rt = (Tc - Ta) / q =(150 - 25) / 43.02

or Rt = 2.70 ºC s/cal = 0.645 ºC s/J = 680 ºC s/btu

```
PID=   Ø.25ØØE+ØØ  Ø.ØØØØE+ØØ  Ø.ØØØØE+ØØ
TIME STEP=  Ø.4ØØØØØE+Ø1 PLOT BOUNDS=+/- -Ø.1ØØØØØE+Ø1
RT,CT,TC,TP=  Ø.6732ØØE+Ø3  Ø.4Ø4ØØØE+Ø1  Ø.271973E+Ø4  Ø.3ØØØØØE+Ø2
```

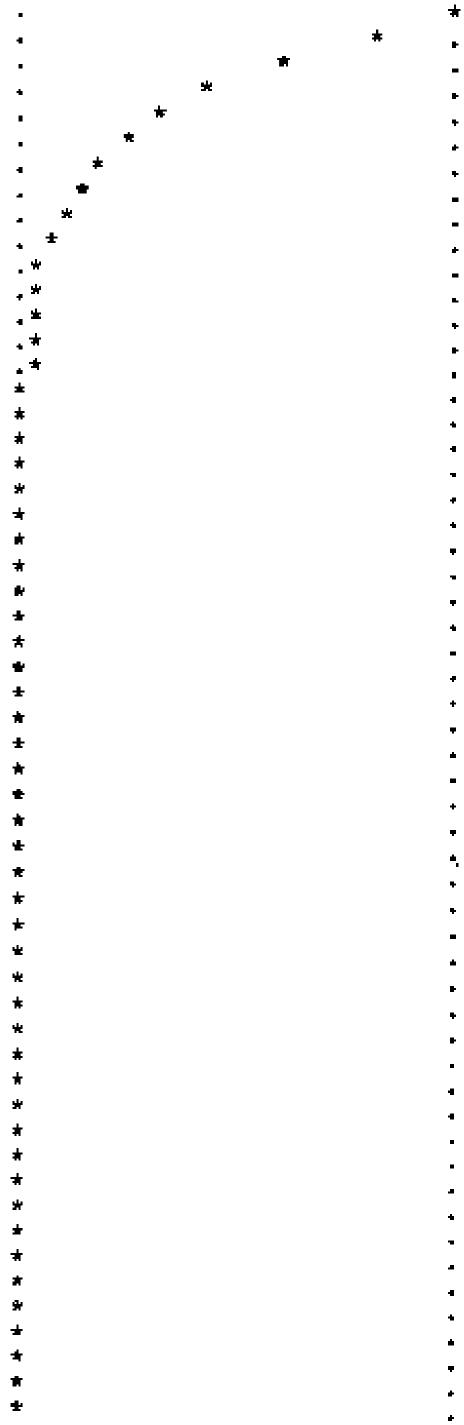


Ø.ØØ.	-1.ØØØØ
4.ØØ.	-Ø.94Ø3
8.ØØ.	-Ø.7784
12.ØØ.	-Ø.5462
16.ØØ.	-Ø.2795
2Ø.ØØ.	-Ø.Ø14Ø
24.ØØ.	Ø.2195
28.ØØ.	Ø.3977
32.ØØ.	Ø.5Ø67
36.ØØ.	Ø.5426
4Ø.ØØ.	Ø.51Ø4
44.ØØ.	Ø.4224
48.ØØ.	Ø.2959
52.ØØ.	Ø.15Ø6
56.ØØ.	Ø.ØØ59
6Ø.ØØ.	-Ø.1215
64.ØØ.	-Ø.2188
68.ØØ.	-Ø.27Ø4
72.ØØ.	-Ø.2981
76.ØØ.	-Ø.28Ø7
8Ø.ØØ.	-Ø.2329
84.ØØ.	-Ø.164Ø
88.ØØ.	-Ø.Ø849
92.ØØ.	-Ø.ØØ59
96.ØØ.	Ø.Ø635
1ØØ.ØØ.	Ø.1166
1Ø4.ØØ.	Ø.1492
1Ø8.ØØ.	Ø.16Ø1
112.ØØ.	Ø.15Ø7
116.ØØ.	Ø.1247
12Ø.ØØ.	Ø.Ø872
124.ØØ.	Ø.Ø44Ø
128.ØØ.	Ø.ØØ1Ø
132.ØØ.	-Ø.Ø369
136.ØØ.	-Ø.Ø659
14Ø.ØØ.	-Ø.Ø837
144.ØØ.	-Ø.Ø897
148.ØØ.	-Ø.Ø846
152.ØØ.	-Ø.Ø7Ø4
156.ØØ.	-Ø.Ø5ØØ
16Ø.ØØ.	-Ø.Ø265
164.ØØ.	-Ø.ØØ31
168.ØØ.	Ø.Ø176
172.ØØ.	Ø.Ø334
176.ØØ.	Ø.Ø432
18Ø.ØØ.	Ø.Ø465
184.ØØ.	Ø.Ø437
188.ØØ.	Ø.Ø36Ø
192.ØØ.	Ø.Ø249
196.ØØ.	Ø.Ø121
2ØØ.ØØ.	-Ø.ØØØ7
2Ø4.ØØ.	-Ø.Ø12Ø
2Ø8.ØØ.	-Ø.Ø2Ø6
212.ØØ.	-Ø.Ø259
216.ØØ.	-Ø.Ø277
22Ø.ØØ.	-Ø.Ø262
224.ØØ.	-Ø.Ø221

FIGURE 8

Pg 12

```
PID=    0.2500E+00   0.1000E-03   0.1000E+02
TIME STEP=   0.100000E+01 PLOT BOUNDS=+/- -0.100000E+01
RT,CT,TC,TP=   0.673200E+03  0.404000E+01  0.271973E+04   0.300000E+02
```



0.00.	-1.0000
1.00.	-0.7308
2.00.	-0.5360
3.00.	-0.3950
4.00.	-0.2929
5.00.	-0.2189
6.00.	-0.1653
7.00.	-0.1263
8.00.	-0.0980
9.00.	-0.0774
10.00.	-0.0624
11.00.	-0.0513
12.00.	-0.0432
13.00.	-0.0372
14.00.	-0.0327
15.00.	-0.0293
16.00.	-0.0267
17.00.	-0.0247
18.00.	-0.0231
19.00.	-0.0219
20.00.	-0.0208
21.00.	-0.0200
22.00.	-0.0192
23.00.	-0.0186
24.00.	-0.0180
25.00.	-0.0175
26.00.	-0.0170
27.00.	-0.0165
28.00.	-0.0161
29.00.	-0.0157
30.00.	-0.0153
31.00.	-0.0149
32.00.	-0.0145
33.00.	-0.0142
34.00.	-0.0138
35.00.	-0.0135
36.00.	-0.0132
37.00.	-0.0129
38.00.	-0.0126
39.00.	-0.0123
40.00.	-0.0120
41.00.	-0.0117
42.00.	-0.0114
43.00.	-0.0111
44.00.	-0.0109
45.00.	-0.0106
46.00.	-0.0103
47.00.	-0.0101
48.00.	-0.0098
49.00.	-0.0096
50.00.	-0.0094
51.00.	-0.0092
52.00.	-0.0089
53.00.	-0.0087
54.00.	-0.0085
55.00.	-0.0083
56.00.	-0.0081

As $t \to \infty$

ERROR $\to 0.0$

FIGURE 11

Pg 15

PID= 0.2500E+00 0.1000E-03 0.5000E+01
TIME STEP= 0.100000E+01 PLOT BOUNDS=+/- -0.100000E+01
RT,CT,TC,TP= 0.673200E+03 0.404000E+01 0.271973E+04 0.300000E+02

0.00.	-1.0000
1.00.	-0.8531
2.00.	-0.7256
3.00.	-0.6151
4.00.	-0.5194
5.00.	-0.4367
6.00.	-0.3653
7.00.	-0.3036
8.00.	-0.2505
9.00.	-0.2049
10.00.	-0.1657
11.00.	-0.1322
12.00.	-0.1035
13.00.	-0.0791
14.00.	-0.0584
15.00.	-0.0408
16.00.	-0.0260
17.00.	-0.0136
18.00.	-0.0033
19.00.	0.0053
20.00.	0.0123
21.00.	0.0181
22.00.	0.0227
23.00.	0.0263
24.00.	0.0292
25.00.	0.0313
26.00.	0.0328
27.00.	0.0339
28.00.	0.0345
29.00.	0.0348
30.00.	0.0347
31.00.	0.0345
32.00.	0.0340
33.00.	0.0334
34.00.	0.0326
35.00.	0.0318
36.00.	0.0308
37.00.	0.0299
38.00.	0.0288
39.00.	0.0278
40.00.	0.0267
41.00.	0.0256
42.00.	0.0246
43.00.	0.0235
44.00.	0.0225
45.00.	0.0215
46.00.	0.0205
47.00.	0.0195
48.00.	0.0186
49.00.	0.0177
50.00.	0.0168
51.00.	0.0160
52.00.	0.0152
53.00.	0.0144
54.00.	0.0137
55.00.	0.0130
56.00.	0.0123

As $t \to \infty$
ERROR $\to$ Ø.Ø

FIGURE 10

```
PID=    0.2500E+00  0.1000E-03  0.0000E+00
TIME STEP=  0.400000E+01 PLOT BOUNDS=+/- -0.100000E+01
RT,CT,TC,TP=  0.673200E+03  0.404000E+01  0.271973E+04  0.300000E+02
```

0.00.	-1.0000
4.00.	-0.9403
8.00.	-0.7782
12.00.	-0.5454
16.00.	-0.2780
20.00.	-0.0114
24.00.	0.2233
28.00.	0.4026
32.00.	0.5126
36.00.	0.5490
40.00.	0.5168
44.00.	0.4282
48.00.	0.3007
52.00.	0.1540
56.00.	0.0076
60.00.	-0.1214
64.00.	-0.2201
68.00.	-0.2808
72.00.	-0.3011
76.00.	-0.2837
80.00.	-0.2353
84.00.	-0.1655
88.00.	-0.0850
92.00.	-0.0046
96.00.	0.0663
100.00.	0.1206
104.00.	0.1541
108.00.	0.1655
112.00.	0.1561
116.00.	0.1296
120.00.	0.0914
124.00.	0.0472
128.00.	0.0031
132.00.	-0.0359
136.00.	-0.0658
140.00.	-0.0843
144.00.	-0.0906
148.00.	-0.0855
152.00.	-0.0711
156.00.	-0.0502
160.00.	-0.0259
164.00.	-0.0017
168.00.	0.0197
172.00.	0.0362
176.00.	0.0464
180.00.	0.0499
184.00.	0.0472
188.00.	0.0393
192.00.	0.0278
196.00.	0.0145
200.00.	0.0012
204.00.	-0.0105
208.00.	-0.0196
212.00.	-0.0252
216.00.	-0.0272
220.00.	-0.0257
224.00.	-0.0214

As $t \to \infty$

ERROR $\to \emptyset.\emptyset$

FIGURE 9

Pg 13

```
1 REM *** PROGRAM(RS232) IN BASIC TO CALCULATE THERMAL CAPACITANCE ***
5 DIM T$(10),D$
10 REM** MAKE SURE DEFAULTS IN SDEF MENU ARE: NO ECHO, BAUD RATE=2400**
20 OPEN "COM1:2400,N,8,1,CS,DS,CD" AS #1
25 INPUT "TEMP STEP?", TSTEP
30 ON TIMER(10) GOSUB 610
40 PRINT #1, "TEMP?"
50 INPUT #1, OLDCT
55 PRINT "OLD CHAM TEMP=", OLDCT
60 TIMER ON
70 IW=0
80 REM WAITING
90 IF IW=0 GOTO 80
100 PRINT #1, "TEMP?"
110 INPUT #1, CT
115 PRINT "CHAMB TEMP=", CT
120 TIMER ON
130 IW=0
140 REM WAITING
150 IF IW=0 GOTO 140
160 DELCT=ABS(OLDCT-CT)
170 OLDCT=CT
180 IF DELCT>.25 GOTO 100
190 SCT=CT
200 PRINT "STARTING CHAMBER TEMP=", SCT
210 PRINT #1, "TIME?"
220 INPUT #1, T$
221 H=VAL(LEFT$(T$,2))
222 M=VAL(MID$(T$,4,2))
223 S=VAL(MID$(T$,7,2))
230 PRINT "START TIME OF DAY=", H;":";M;":";S
235 STIME=(H*60*60)+(M*60)+S
236 PRINT "START TIME IN SECONDS=", STIME
240 PRINT #1, "RATE=200"
245 LINE INPUT #1, D$
250 TT=SCT+TSTEP+20
260 PRINT #1, "SET=";TT
265 LINE INPUT #1, D$
270 PRINT #1, "HON"
275 LINE INPUT #1, D$
280 ON TIMER(1) GOSUB 610
290 PRINT #1, "TEMP?"
300 INPUT #1, TC
305 PRINT "CURRENT CHAMBER TEMP=",TC
310 IF TC >= TT-20 GOTO 365
320 TIMER ON
330 IW=0
340 REM WAITING
350 IF IW=0 GOTO 340
360 GOTO 290
365 PRINT #1, "HOFF"
366 LINE INPUT #1, D$
370 PRINT #1, "TIME?"
380 INPUT #1, T$
381 H=VAL(LEFT$(T$,2))
382 M=VAL(MID$(T$,4,2))
383 S=VAL(MID$(T$,7,2))
```

```
390 PRINT "END TIME OF DAY=", H;":";M;":";S
395 ETIME=(H*60*60)+(M*60)+S
396 PRINT "END TIME IN SECONDS=", ETIME
410 OLDCT=TC
420 CNT=0
430 TIMER ON
440 IW=0
450 REM WAITING
460 IF IW=0 GOTO 450
470 PRINT #1,"TEMP?"
480 INPUT #1, TC
490 IF TC>OLDCT THEN OLDCT=TC
500 CNT=CNT+1
510 IF CNT<60 GOTO 430
520 PRINT  "ENDING CHMBER TEMP=", OLDCT
530 DELTEMP=ABS(OLDCT-SCT)
540 TIME=ETIME-STIME
545 REM**THE POWER SUPPLIED(IN WATTS) SHOULD BE COMPATIBLE TO YOUR TEST SYSTEM**
550 WATTS=1744
555 REM **1 BTU= 1054.368**
560 BTU=(WATTS*TIME)/1054.368
570 CT=BTU/DELTEMP
580 PRINT "DELTEMP=",DELTEMP
590 PRINT "TOTAL HEAT FLOW(BTU)=", BTU
600 PRINT "THERMAL CAPACITANCE=BTU/DELTEMP=", CT
605 STOP
610 IW=1
620 RETURN
630 END
```

```
      PROGRAM CHPID
      DIMENSION X(10),COEF(10),RTR(10),RTI(10)
      CHARACTER BUF(64),IRESP
      REAL KH,KP,KI,KD
      COMPLEX R1,R2,R3,ANS,ADS,BNS,BDS,CNS,CDS,AS,BS,CS
      COMPLEX RESP1S,RESP2S,RESP3S,RESPS,S0,S1,S2,S3
C
C     CALCULATES THE TIME ERROR RESPONSE
C     NOTE:  REQUIRES SUBROUTINE ROOT
C
C     WHERE E(S)=[1/(1+G(S)H(s))]*R(S)
C
C       AND R(S)=1/S FOR STEP INPUT
C
C     G.G.C.  1985  SUN ELECTRONIC SYSTEMS INC.
C
C       RT=THERMAL RESISTANCE
C       CT=THERMAL CAPACITANCE
C       TP=PROBE TIME CONSTANT
C       KH=HEATER GAIN CONSTANT
C       KP,KI,KD=PROPORTIONAL,INTEGRAL AND DERIVATIVE COEFFICIENTS (PID)
C     TC=CHAMBER TEMPERATURE
C       TAMB=AMBIENT TEMPERATURE
C
2       IDEV=6
        M=3
        RT=673.2
        T=2.0
        CT=4.04
        TP=30
        KH=1.76
        WRITE(6,3)
3     FORMAT(' USE DEFAULT RT,CT,TP,KH VALUES [Y/N]')
        READ(5,50)IRESP
        IF (IRESP.NE.'Y') THEN•
        WRITE(6,4)
4       FORMAT(' ENTER Rt,Ct,TP,KH')
        READ(5,5)RT
        READ(5,5)CT
        READ(5,5)TP
        READ(5,5)KH
        END IF
        TC=RT*CT
6       WRITE(6,7)
7     FORMAT(' ENTER PID COEFICIENTS')
        READ(5,5)KP
        READ(5,5)KI
        READ(5,5)KD
5       FORMAT(F14.6)
        TAMB=0.0
        OPEN(UNIT=9,FILE='LPT1')
C
C       CALCULATE DENOMINATOR COEFICIENTS
C       WHERE X(N) REPRESENTS S**(N-1) COEFICIENT
```

```
      X(4)=1.0
     X(3)=1.0/TP+1.0/TC+(KD*KH*RT)/(TP*TC)
     X(2)=1.0/(TC*TP)+(KP*KH*RT)/(TP*TC)+TAMB/(TP*TC)
      X(1)=(KI*KH*RT)/(TP*TC)•
      DO 20 I=0,3
      TEMP=X(I+1)
      WRITE(6,8)I,TEMP
8     FORMAT(' S**',I5,' TERM=',E14.6)
20    CONTINUE
C
C     FIND ROOTS OF DENOMINATOR
C     RTR=REAL ROOTS
C     RTI=IMAGINARY ROOTS
C
     CALL ROOT(X,COEF,M,RTR,RTI,IR)
      WRITE(6,10)IR
10   FORMAT(' ROOT ERROR STATUS=',I8)
      WRITE(6,12)
12    FORMAT('      REAL       IMAJ')
      DO 13 I=1,M
      WRITE(6,14)RTR(I),RTI(I)
14   FORMAT(2X,E12.4,2X,E12.4)
13    CONTINUE
C
C     THE ROOTS ARE NEGATIVE IF STABLE BUT FOR (S+A) A IS POSITIVE
C
     R1=-CMPLX(RTR(1),RTI(1))
     R2=-CMPLX(RTR(2),RTI(2))
     R3=-CMPLX(RTR(3),RTI(3))
C
C     CHECK THAT THE ROOTS ARE CORRECT
C     S0,S1,S2,S3 REPRESENT DENOMINATOR COEFFICIENTS
C
      S3=1.0
      S2=R1+R2+R3
      S1=R1*R2+R2*R3+R1*R3
      S0=R1*R2*R3
      WRITE(6,15)S0,S1,S2,S3
15   FORMAT(' ROOT CHECK=',4E14.6)
C
C     PERFORM PARTIAL FRACTION EXPANSION SUCH THAT
C     N(S)/D(S)=AS/(S+R1) + BS/(S+R2) + CS/(S+R3)
C     I.E. FIND AS,BS,CS
C     NOTE: MULTIPLE ROOTS AT ZERO ARE NOT ALLOWED OR EXPECTED
C
C     N(S)=S*S+S(1.0/TP+1.0/TC)+1.0/(TP*TC)
C
      XX=1.0/TP+1.0/TC
      YY=1.0/(TP*TC)
      ANS=R1*R1-R1*XX+YY
      BNS=R2*R2-R2*XX+YY
      CNS=R3*R3-R3*XX+YY
      ADS=1.0*(R2-R1)*(R3-R1)
      BDS=1.0*(R1-R2)*(R3-R2)
```

```
      CDS=1.0*(R1-R3)*(R2-R3)
      WRITE(6,16)ANS,ADS,BNS,BDS,CNS,CDS
16    FORMAT(2E12.4,2E12.4/2E12.4,2E12.4/2E12.4,2E12.4)
      AS=-ANS/ADS
      BS=-BNS/BDS
      CS=-CNS/CDS
      WRITE(6,18)AS,BS,CS
18    FORMAT(' AS=',2E14.2/' BS=',2E14.2/' CS=',2E14.2)
100   CONTINUE
C
C     CALCULATE THE TIME RESPONSE WHERE
C     E(t)=AS*e**(-R1*t)+BS*e**(-R2*t)+CS*e**(-R3*t)
C
C     AND PLOT TIME RESPONSE
C
      WRITE(6,30)
30    FORMAT(' PLOT ON LINE PRINTER[Y/N]')
      READ(5,50)IRESP
      IDEV=6
      IF(IRESP.EQ.'Y')IDEV=9
      WRITE(6,25)
25    FORMAT(' ENTER TIME STEP , PLOT LIMIT')
      READ(5,26)STEP
      READ(5,26)PLIMIT
26    FORMAT(E12.4)
      WRITE(IDEV,27)KP,KI,KD
27    FORMAT(' PID= ',3E12.4)
      PM=PLIMIT/32.0
      WRITE(IDEV,28)STEP,PLIMIT
28    FORMAT(' TIME STEP=',E14.6,' PLOT BOUNDS=+/-',E14.6)
      WRITE(IDEV,34)RT,CT,TP,KH
34    FORMAT(' RT,CT,TP,KH=',4E14.6)
      DO 40 TIME=0.0,400*STEP,STEP
      RESP1S=AS*CEXP(-R1*TIME)
      RESP2S=BS*CEXP(-R2*TIME)
      RESP3S=CS*CEXP(-R3*TIME)
      RESPS=RESP1S+RESP2S+RESP3S
      RRESPS=REAL(RESPS)
      DO 35 I=1,64
35    BUF(I)=' '
      S0=RRESPS/PM+32
      I=INT(S0)
      IF(I.LT.1)I=1
      IF(I.GT.64)I=64
      BUF(1)='.'
      BUF(32)='.'
      BUF(64)='.'
      BUF(I)='*'
      WRITE(IDEV,36)TIME,(BUF(I),I=1,64),RRESPS
36    FORMAT(F8.2,64A1,F8.4)
40    CONTINUE
      WRITE(6,45)
45    FORMAT(' PLOT SAME PID AGAIN [Y/N]')
      READ(5,50)IRESP
```

```
      IF(IRESP.EQ.'Y')GOTO 100
      WRITE(6,46)
46    FORMAT(' RUN AGAIN NEW PIDs ? [Y/N]')
      READ(5,50)IRESP
      IF(IRESP.EQ.'Y')GOTO 6
      WRITE(6,47)
47    FORMAT(' RUN AGAIN ? [Y/N]')
      READ(5,50)IRESP
      IF(IRESP.EQ.'Y')GOTO 2
50    FORMAT(1A1)
C
C
      STOP
      END
```

```
C         SUBROUTINE ROOT (XCOF, COF, M, ROOTR, TOORI, IER)
C         ERROR STATUS IN IER
C         0 = NO ERROR
C         1 = ORDER LESS THAN 1
C         2 = ORDER GT 10
C         3 = CANNOT FIND ROOTS
C         4 = COEF OF HIGHEST TERM = 0
C
                    DIMENSION XCOF(10), COF(10), ROOTR(10), ROOTI(10)
                    DOUBLE PRECISION X0, Y0, X, Y, XPR, YPR, UX, UY, V, YT, XT, U
                    DOUBLE PRECISION XT2, YT2, SUMSQ, DX, DY, TEMP, ALPHA
                    N = M
                    IFIT = 0
                    IER = 0
                    IF (XCOF (N+1))10, 25, 10
10                  IF (N)15, 15, 32
15                  IER = 1
20                  RETURN
25                  IER = 4
                    GOTO 20
30                  IER = 2
                    GOTO 20
32                  IF (N-10)35, 35, 30
35                  NX = N
                    NXX = N+1
                    N2 = 1
                    KJ1 = N+1
                    DO 40 L = 1, KJ1
                    MT = KJ1-L+1
40                  COF (MT) = XCOF (L)
45                  X0 = 0.00500101
                    Y0 = 0.01000101
                    IN = 0
50                  X = X0
                    X0 = -10.0*Y0
                    Y0 = -10.0*X
                    X = X0
                    Y = Y0
                    IN = IN+1
                    GOTO 59
55                  IFIT = 1
                    XPR = X
                    YPR = Y
59                  ICT = 0
60                  UX = 0.0
                    UY = 0.0
                    V = 0.0
                    YT = 0.0
                    XT = 1.0
                    U = COF (N+1)
                    IF (U)65, 130, 65
65                  DO 70 I=1, N
                    L = N-I+1
                    TEMP = COF (L)
```

```
          XT2 = X*XT-Y*YT
          YT2 = X*YT+Y*XT
          U = U+TEMP*XT2
          V = V+TEMP*YT2
          FI = I
          UX = UX+FI*XT*TEMP
          UY = UY-FI*YT*TEMP
          XT = XT2
70        YT = YT2
          SUMSQ = UX*UX+UY*UY
          IF (SUMSQ)75, 110, 75
75        DX = (V*UY-U*UX)/SUMSQ
          X = X+DX
          DY = -(U*UY+V*UX)/SUMSQ
          Y = Y+DY
78        IF (DABS(DY)+DABS(DX)-(1.0D-05))100, 80, 80
80        ICT = ICT+1
          IF (ICT-500)60, 85, 85
85        IF (IFIT)100, 90, 100
90        IF (IN-5)50, 95, 95
95        IER = 3
          GOTO 20
100       DO 105 L=1, NXX
          MT = KJ1-L+1
          TEMP = XCOF(MT)
          XCOF(MT) = COF (L)
105       COF (L) = TEMP
          ITEMP = N
          N = NX
          NX = ITEMP
          IF (IFIT)120, 55, 120
110       IF (IFIT)115, 50, 115
115       X = XPR
          Y = YPR
120       IFIT = 0
122       IF (DABS(Y)-(1.0D-4)*DABS(X))135, 125, 125
125       ALPHA = X+X
          SUMSQ = X*X+Y*Y
          N = N-2
          GOTO 140
130       X = 0.0
          NX = NX-1
          NXX = NXX - 1
135       Y = 0.0
          SUMSQ = 0.0
          ALPHA = X
          N = N-1
140       COF(2) = COF(2)+ALPHA*COF(1)
          NE = N
          IF (NE.LT.2)NE = 2
145       DO 150 L=2, NE
150       COF (L+1) = COF(L+1)+ALPHA*COF(L)-SUMSQ*COF(L-1)
155       ROOTI (N2) = Y
```

```
      ROOTR (N2) = X
      N2 = N2 +1
      IF (SUMSQ)160, 165, 160
C
C
160   Y = -Y
      SUMSQ = 0.0
      GOTO 155
165   IF (N)20, 20, 45
      END
```

```
              PROGRAM CHISM
              CHARACTER IRESP, BUF (64)
              REAL KP, KI, KD, KH, NEWTMP
              OPEN (UNIT = 9, FILE = 'LPT1')
C
C             CALCULATES TIME RESPONSE OF NONLINEAR-SAMPLED DATA CONTROL SYSTEM
C
C             G.G.C.  SUN ELECTRONIC SYSTEMS, INC.
C
100           CONTINUE
              IDEV = 6
              RT = 673.2
              T = 2.0
              CT = 4.04
              TP = 1.0
              KH = 3.82
              WRITE (6, 3)
3             FORMAT ('USE DEFAULT Rt, Ct, Tp, KH VALUES [Y,N]')
              READ (5, 8)IRESP
              IF (IRESP.NE.'Y') THEN
              WRITE (6, 16)
16            FORMAT ('  ENTER Rt, Ct, Tp, Kh')
              READ (5, 4)RT
              READ (5, 4)CT
              READ (5, 4)TP
              READ (5, 4)KH
              END IF
              XKTMAX = 2.147E12
              WRITE (6, 15)
15            FORMAT ('  ENTER TAMB, NEWTMP')
              READ (5, 4)TAMB
              READ (5, 4) NEWTMP
4             FORMAT (E14.6)
              WRITE (6, 10)
10            FORMAT ('  ENTER PID COEFFICIENTS')
              READ (5, 4) KP
              READ (5, 4) KI
              READ (5, 4) KD
              WRITE (6, 11)
11            FORMAT ('  PLOT ON LP [Y,N]')
              READ (5, 8) IRESP
              IF (IRESP.EQ.'Y') IDEV = 9
              XKT = 0.0
              QINT = 0.0
              Q = 0.0
              ODTMP = 0.0
              CTL = 0.0
              TCHAM = TAMB
              TPINT = 0.0
              TPROBE = TAMB
              SETTMP = TAMB
              IDCNT = 0
              STEP = ABS (NEWTMP - TAMB)
              PM = STEP / 32.0
```

```
                K = NINT (15*STEP)
                K = 1000
                WRITE (IDEV, 12)RT, CT, TP, KH, NEWTMP, TAMB
12              FORMAT('  Rt, Ct, Tp, KH, NEWTMP, TAMB = ', 6E12.1)
                WRITE (IDEV, 13)KP, KI, KD
13              FORMAT ('  PID=', 3E12.4
C
C       ASSUMES SAMPLE RATE T = 2.0 SEC
C       KP SINCE IN TC10 1 = FULL ON
C       KI SINCE IN TC10 1 = FULL ON AND NO SCALE FACTOR
C       KD FOR SAM REASONS AS KP
C
                DO 2 J = -10, K
                IF (J.GE.0) SETTMP = NEWTMP
                DELTMP = SETTMP - TPROBE
                ADTMP = ABS (DELTMP)
                ADDTMP = ABS (DELTMP-ODTMP)
                IF (ADTMP.GT.10.0)XKT = 0.0
                CTL = (KP*DELTMP)
                XKT = XKT+T*((ODTMP+DELTMP)/2)
                IF (XKT.GT.XKTMAX) XKT = XKTMAX
                IF (XKT.LT.-XKTMAX) XKT = -XKTMAX
                CTL = CTL+(KI*XKT)+(KD*(DELTMP-ODTMP)/T)
                IF ((J.GE.0) .AND. (ADDTMP.LT.1.E-4)) THEN
                IDCNT = IDCNT+1
                ELSE
                IDCNT = 0
                END IF
                IF (IDCNT.GT.20) GOTO 2
                DDTMP = DELTMP - ODTMP
                ODTMP = DELTMP
C       LIMIT CTL OUTPUT
                IF (CTL.GT.1.0)CTL = 1.0
                IF (CTL.LT.-1.0) CTL = -1.0
                Q = CTL *KH
C
C       ASSUMES SAMPLE TIME T = 0.1 SEC FOR PROCESS MODEL
C
                DO 1 I=1, 20
                QINT = QINT+(Q*0.1)-(TCHAM-TAMB)*(0.1/RT)
                TCHAM = (QINT/CT)+TAMB
                TPINT = TPINT +((TCHAM-TPROBE)*0.1)
                TPROBE = (TPINT/TP)+TAMB
C               WRITE (9,30)QINT, TCHAM, TPINT, TPROBE
C       30      FORMAT ('  QINT, TCHAM, TPINT, TPROBE = ', 4E12.1)
1               CONTINUE
C
C               WRITE (9,9) CTL, Q
C       9       FORMAT ('  CTL, Q = ', 2E12.4)
                DO 5 I = 1, 64
5               BUF (I) = ' '
                I1 = NINT (DELTMP/PM+32)
                I2 = NINT (CTL*32.0+32)
                I3 = NINT (TCHAM/10.0+32)
                I4 = NINT (TPROBE/10.0+32)
                I5 = NINT (KI*XKT832.0+32)
                I6 = NINT (KC*DDTMP*32.0+32)
                IF (I1.LT.1)I1 = 1
                IF (I1.GT.64) I1 = 64
```

```
      IF (I2.LT.1) I2 = 1
      IF (I2.GT.64)I2 = 64
      IF (I3.LT.1) I3 = 1
      IF (I3.GT.64) I3 = 64
      IF (I4.LT.1) I4 = 1
      IF (I4.GT.64) I4 = 64
      IF (I5.LT.1) I5 = 1
      IF (I5.GT.64) I5 = 64
      IF (I6.LT.1) I6 = 1
      IF (I6.GT.64) I6 = 64
      BUF (1) = ' . '
      BUF (32) = ' . '
      BUF (64) = ' . '
      BUF (I6) = 'D'
      BUF (I5) = 'I'
      BUF (I4) = 'R'
      BUF (I3) = 'T'
      BUF (I2) = 'Q'
      BUF (I1) = '*'
      ITIME = 2*J
C     WRITE (IDEV, 1111) I1, I2, I3, I4, I5, I6
C 1111 FORMAT (6I2)
      WRITE (IDEV, 6) ITIME, (BUF(I), I = 1, 64), DELTMP
6     FORMAT (I8, 64A1, F8.4)
C
2     CONTINUE
C
C
      WRITE (6, 7)
7     FORMAT ('  RUN AGAIN [Y,N]')
      READ (5, 8) IRESP
8     FORMAT (1A1)
      IF (IRESP.EQ.'Y') GOTO 100
      STOP
      END
```