To the Graduate Council:

I am submitting herewith a thesis written by Robert Wesley Wysor entitled "Coordinated Control of Relay Feedback Control Systems for Electrical Power Cost Reduction." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Electrical Engineering.

J M Googe

Joseph M. Googe, Major Professor

We have read this thesis and recommend its acceptance:

J M Bailey

Donald W. Bouldin

Accepted for the Council:

C W Minkel

Vice Provost and Dean of the Graduate School

STATEMENT OF PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a Master's degree at the University of Tennessee, Knoxville, I agree that the Library shall make it available to borrowers under rules of the Library. Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgement of the source is made.

Permission for extensive quotation from or reproduction of this thesis may be granted by my major professor, or in his absence, by the Head of Interlibrary Services when, in the opinion of either, the proposed use of the material is for scholarly purposes. Any copying or use of the material in this thesis for financial gain shall not be allowed without my written permission.

Signature Robert Wesley Hyror

Date July 12, 1988

COORDINATED CONTROL OF RELAY FEEDBACK CONTROL SYSTEMS FOR ELECTRICAL POWER COST REDUCTION

A Thesis

Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

R. Wesley Wysor

August 1988

ACKNOWLEDGMENTS

The author wishes to thank the members of his Committee, Drs. J. M. Bailey, D. W. Bouldin, and J. M. Googe, for their help and support in this project. Special thanks goes to Major Advisor Dr. J. M. Googe for his patience, support, and invaluable advice throughout the entire M.S.E.E. Program.

A special thanks is extended to his parents for their understanding and support and especially for their contribution in the area of computational resources. The author also wishes to thank Karen Mahan for her assistance in preparing the figures and also for her encouragement in the completion of this study.

Acknowledgment and appreciation are extended to Shirley Hendrix for her editorial assistance and to Kim Gaddis for her help in getting this thesis in final form.

Special appreciation is also extended to Jim Jansen and the members of the Real Time Computing Systems Group in the Instrumentation and Controls Division at the Oak Ridge National Laboratory for their support and for providing the computational resources needed for the large simulations.

ABSTRACT

A study was undertaken to determine if coordinated control could be used to reduce the peak electrical demand for a group of cycling controllers. In this study, a bank of six aluminum furnaces was modeled and studied with a computer simulation. In this simulation, random disturbances in weight were used to simulate adding and removing aluminum from the furnaces. Three coordinated control algorithms were developed and simulated. These algorithms included a fixed and floating maximum number of units on at a time and a scheme in which adjacent units shared a duty cycle. Electrical cost savings from these algorithms was from 15 to 30%, depending on operating conditions.

Two proposed methods of implementing coordinated control are a digital controller and a simple system of relays and switches. The simpler implementation provides the quicker return on investment and an easy way to test the coordinated control concept. The digital control system is more expensive but offers the advantage of being a development system in which different algorithm variations can be implemented and tested. Success in this application should provide an incentive for considering coordinated control in other applications.

TABLE OF CONTENTS

CHAPTER PAGE

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES (Continued)

INTRODUCTION

This study concerns minimizing the cost of electrical power resulting from the operation of a group of relay (or bang-bang) controllers. The proposed method involves coordinating the control of previously independent systems while maintaining their capability to keep their individual control parameters within given ranges.

The idea for this thesis topic was based on a practical problem noted at a local casting company. The company operates six large electric furnaces which are used to melt aluminum for specialty castings. The temperature of these furnaces is regulated by bang-bang controllers which supply power to three-phase heating elements. The contacts in these controllers open and close as the furnace temperature reaches the upper and lower limits of a specified temperature range.

The cost of commercial power is based on the total power usage as well as the peak power demand for a given period. This demand charge can account for as much as half of the monthly power cost. If cycling loads can be coordinated to minimize the peak power demand, a potential savings of up to 30% can be realized in electrical energy cost. It is important, however, that the cycling loads still operate so that they maintain the control parameters within the required ranges.

Preliminary study has shown that it may not be feasible to use this approach in the case of the local casting company during some operating conditions. For example, when large parts are being cast, the aluminum usage rate is such that the furnaces must be operated continuously (all furnaces on) to maintain a sufficient supply of molten aluminum. During these conditions, a coordinated system needs to be bypassed. Consequently, coordinated control of the furnaces during the remainder of this billing period provides no potential energy savings, since the peak demand already has been reached. Assuming a change in operating conditions, coordinated control can then be resumed to reduce the electrical cost for the next billing period.

These furnaces provide a good model for a type of operation that might apply to a variety of applications. They operate independently with bang-bang controllers which cycle within a differential range to maintain a nominal furnace temperature. Disturbances to them are random and may differ from furnace to furnace. If an algorithm can be found to optimize the control of these furnaces, it is likely that it could be generalized for other applications involving a group of cycling controllers.

The first objective in the research of this problem was to construct a mathematical model of the six furnaces used at the local casting company. Because these furnaces provided a source of data for cycling controllers, they were used to verify the mathematical model. The model was used to provide a computer simulation tool to observe the effects of the coordinated control algorithm.

Simulation was accomplished in two steps. The furnace dynamics were first established using the True Basic language on a Macintosh Plus personal computer. A more complex simulation of six independent control systems operating simultaneously was accomplished through the use of a software package known as the Advanced Continuous Simulation Language (ACSL). The ACSL simulation was also used in the development of the control algorithm.

Once it was established that the model operation was similar to the operation of the actual furnaces, the algorithm for coordinated control was developed. This involved a study of the function that determines the cost of commercial electrical power for the local casting company. An attempt was made to minimize this cost by intelligent control of the furnaces. The algorithm was based on the input from the individual furnace thermostats from all six furnaces.

After the algorithm was written, it was tested by using the ACSL simulation model. The ACSL software package provided the necessary flexibility to enter the electrical power cost function directly into the simulation. With this information available, a simulation was performed with all six furnaces operating simultaneously and responding to different random disturbances. The cost of electrical power was then available at the end of each simulation run. These data made it possible to assess the effects of changes and adjustments to the algorithm.

The implementation of the algorithm for the local casting company was then considered. Although it is beyond the scope of this study to produce a detailed hardware design, several implementation alternatives are discussed. This thesis includes generalized block diagrams and a discussion of the system requirements.

CHAPTER I

PROBLEM STATEMENT

Before proceeding with this study it was necessary to outline a purpose to provide a measure of success of the result. It was also necessary to define the assumptions used to ensure that results were measured in their proper context. In addition, the limitations of the study provided the necessary boundaries to focus the effort on the desired purpose. This chapter deals with these topics and precedes the discussion of the study itself.

Purpose of Study

There are many applications in which a number of cycling electrical loads operate independently of each other. Many of these applications are such that electrical billing is based on the peak amount of power demanded during that period as well as the amount of power consumed. When these conditions exist, there is a potential for reducing the cost of operating these electrical loads. This study examines a specific example of this type of operation.

There were three main points to consider in initiating this study. The first was that it had to be determined if it was possible to coordinate the cycling of these loads without limiting their operational capability. This coordination amounted to reducing peak electrical demand by cycling some of the loads on when other loads were off. The second consideration was to determine if this coordination resulted in a significant savings in electrical cost. Depending on the potential savings, the third consideration was to determine if this savings justified the cost of implementing a coordinated control system. In other words, it was necessary to determine how long it would take for the electrical cost savings to pay for the cost of implementing the control system.

In a broader sense, this study serves as an example for other applications in which loads are cycled and for which peak demand is billed. This example can be compared with other applications to determine whether a coordinated control system might be worth considering. This study also serves as an example for developing and implementing a coordinated control system.

Assumptions

One assumption of this study is that the furnaces can be modeled with a second-order system. This assumption is based on there being two primary heat storage elements involved, the crucible containing the molten aluminum and the molten aluminum itself. It is further assumed that cycling times of the furnaces are a reasonable measure for determining the accuracy of the model. Although it is understood that this model may not give an exact representation of the furnace dynamics, it is primarily the cycling times that are of interest.

This example assumes the coordinated system is an addition to the existing independent control systems. It also assumes that, in the event of a failure, deactivating the coordinated control system returns the system to a safe state. That is, the proposed control system does not add any additional safety features or interlocks.

Limitations

The furnace model is limited to operation in which the aluminum is in the molten state. Since the properties of the heat transfer and storage elements change with temperature and state, this model will not be as accurate for furnaces close to room temperature, but it can be used to predict the general response of the furnace with the understanding that there may be some distortion with respect to the timing of that response.

The furnaces in this example use mechanical contacts to turn heating elements on and off, thus placing some limitation on the control system concerning how often the heating elements are energized. If the contacts are cycled several times a second, the additional wear and tear can cause them to fail prematurely. Ideally, the contacts are not cycled any more often than with an independent control system (applications for which electronic switching is used do not have this restriction).

To maximize the ratio of cost savings to cost of implementation, every effort has been made to keep the coordinated control system as simple as possible. For this reason, the coordinated system uses the thermostats in the individual furnace controllers. It senses whether the thermostat switches are open or closed. This is equivalent to sensing when the individual controllers would cycle on or off under independent control. The control system uses these signals to determine when to cycle the furnaces. This configuration offers the advantage that it can be generalized to other applications. For example, the controller could receive a signal from a pressure switch instead of a thermostat switch.

While the furnaces used in this example are capable of being operated at full or half power, this study was limited to furnaces operating at full power only. Although provisions were made for changing the power level of the heating element in the simulation, it was felt that this would lead to an unnecessary level of complexity. In addition, operation at a single power level provides a result that may be generalized to a wider variety of applications.

CHAPTER II

MODEL DEVELOPMENT

An important part of developing an algorithm for peak electrical demand reduction was the development of a simple model of the furnaces used in the example problem. The model is basically a set of mathematical equations used to predict the response of the furnace under a given set of conditions. Since one of the main purposes of this study is to present a method that can be extended to related applications, it is not extremely critical that the model provide a duplicate response to any one of the furnaces at the local casting company. It is, however, desirable that the model provide a response that is reasonably close to that represented by the test data obtained from the operating furnaces.

Before the furnace model equations were written, a careful look was taken at the operation of the system. For the purpose of this model, data were collected at the foundry operated by the local casting company where six furnaces are used. A model was constructed for each of the two furnace sizes used at the foundry.

Furnace Description

The furnaces were manufactured by the Naber company in West Germany and are distributed by Bickley Furnaces Incorporated in Philadelphia. Electrical schematics obtained for the furnaces, though somewhat difficult to interpret because they were annotated in German, did not present too much of a problem because of the relative simplicity of the control device. Most of the symbols and specifications on the schematics became recognizable after careful examination.

One of the most important parts of developing model equations for the furnaces involved obtaining a simplified representation of the complex heat transfer mechanisms. Unfortunately, detailed drawings of the furnaces were not available to show the physical arrangement of the surfaces and heat flow paths involved. Most of the required information, however, was obtained when one of the furnaces was disassembled for maintenance, thus providing the opportunity for measurements and sketches of the interior configuration. A simplified cross-sectional view of one of the furnaces is shown in Figure 2-1.

With the exception of capacity and physical dimensions, all six furnaces are essentially similar. Four of the furnaces have a capacity of 500 lb and two have a capacity of 800 lb. Each furnace is heated by 48 coils of nickel-chromium wire which are mounted in a chamber surrounding a carbon bonded silicon carbide crucible. Three-phase power is supplied to the heating unit through two solenoid operated relays in the furnace controller. The coils of the heating unit are divided into two delta connected groups, one of which is connected to each three-phase relay in the controller. The furnace controller has a power level selection switch to allow one or both of the groups of heating coils to cycle on and off as a setpoint is reached on a thermostat. This allows the user to select either full power or half power, depending on the expected operating load. The scope of this project includes control of the furnace in the normal operating range.

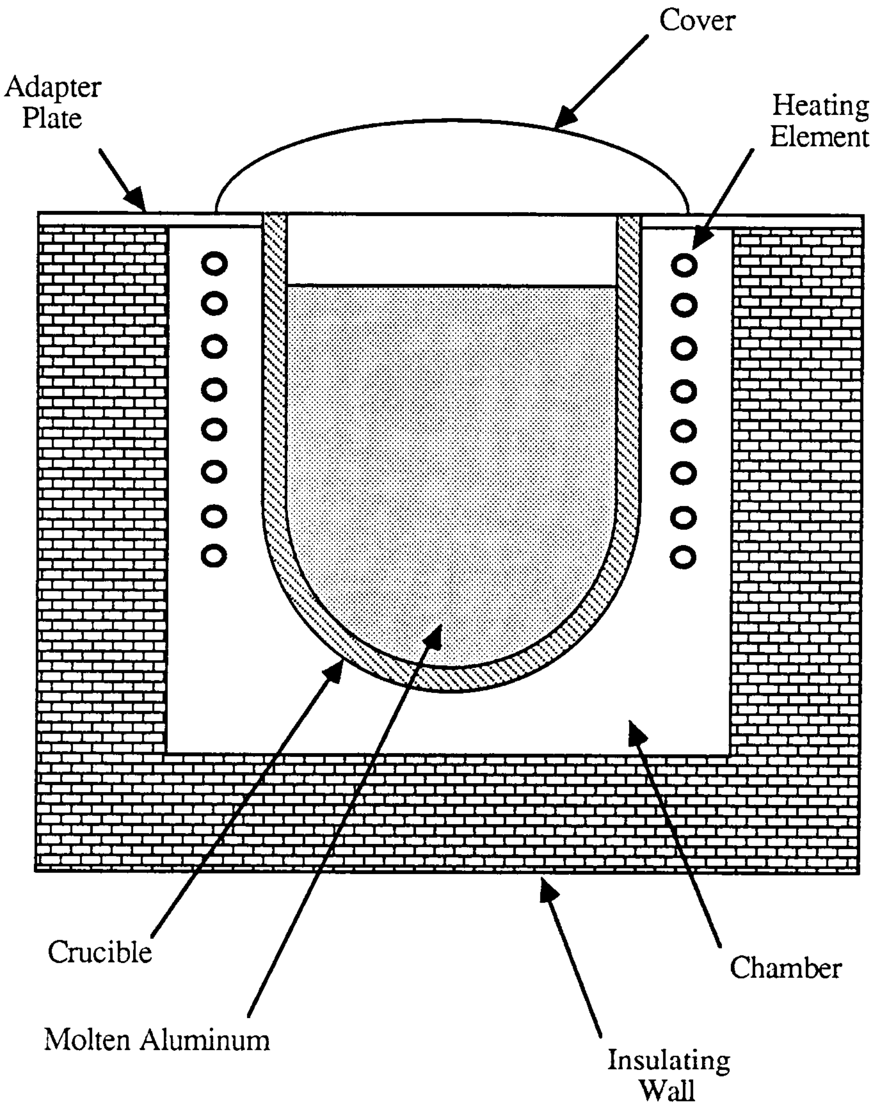


Figure 2-1. Simplified Furnace Cross Section

Although control of furnace heatup might also be desirable, no attempt was made to model and optimize this phase of operation. Modeling a furnace startup from room temperature is a complex problem because aluminum undergoes a phase change. Other furnace materials also operate over a wide range of temperatures. For these reasons, material properties cannot be assumed to be as constant as they are in the normal operating range.

When the heating element is energized, heat energy enters the surrounding chamber because of resistance heating in the nickel chromium coils. The temperature in the chamber continues to rise until the user-selected upper setpoint is reached on the control thermostat. The control relay(s) then opens and removes the electrical supply from the coils. At this point the coils begin to cool, exponentially approaching the temperature of the chamber. Meanwhile, the heat from the chamber flows out to its surroundings, including the crucible containing molten aluminum, the surrounding air via the steel adapter plate, and the surrounding air via the firebrick insulating wall. The design of the furnace is such that most of the heat flow is into the molten aluminum.

Although a more complex model might more exactly represent the complex heat transfer mechanisms in the furnace system, a simple model can produce a response that closely parallels the response of the actual system. The second-order model used in this case offers two advantages: (1) there are a wide variety of analytical tools available for examining the response of second-order systems, and (2) the developed algorithm can easily be applied to other systems since second-order approximations are often used in practice.

Formulating the Model

A useful tool in the analysis of temperature control systems is the electrical circuit analogy, which compares the flow of heat to the flow of electrical current through a current-carrying conductor. One-dimensional heat flow by conduction can be represented by the relationship

$$q = \frac{kA(T_1 - T_2)}{L} \quad , \tag{2-1}$$

where

k = thermal conductivity,

A = cross-sectional area (perpendicular to heat flow),

T_1 = temperature at one end of heat flow path,

T_2 = temperature at other end of heat flow path, and

L = length of heat flow path.

Taking the partial derivative with respect to T_1 and T_2, respectively, yields the equations

$$\frac{\delta q}{\delta T_1} = \frac{kA}{L} \quad , \tag{2-2}$$

and

$$\frac{q}{\delta T_2} = \frac{-kA}{L} \quad . \tag{2-3}$$

When these equations are compared to Ohm's law, written in the form

$$\frac{I}{E} = \frac{1}{R} \quad , \qquad (2\text{-}4)$$

it can be seen that, when heat flow is replaced by current flow and temperature is represented by voltage, electrical resistance can be replaced by an equivalent thermal resistance represented by

$$R = \frac{L}{kA} \quad . \qquad (2\text{-}5)$$

The net heat flow into a material can be integrated to obtain a quantity that is proportional to the temperature of that material. The proportional constant that relates these two quantities is thermal capacitance (C), which can be seen in the equation

$$T = \frac{1}{C} \int q \; dt \quad . \qquad (2\text{-}6)$$

Thermal capacitance in this case is given by the equation

$$C = c_p V p \quad , \qquad (2\text{-}7)$$

where

c_p = specific heat,

V = volume, and

p = density.

Thus, the elements of thermal resistance and capacitance can be interconnected in a diagram similar to the elements of electrical resistance and capacitance (Buckley, pp. 217-218). With this arrangement, a difference of potential on the circuit diagram corresponds to a temperature difference in the physical system. In modeling the furnace system, ambient temperature was used as a reference. This corresponds to setting T_2 to zero in the previous equations. Other temperatures were measured relative to this reference. This approach was used with the previous description of the furnace operation to establish a starting point for the furnace model.

The development of an electrical circuit analogy to model the furnace was an iterative process. The behavior of the model in numerous computer simulations was studied and compared to experimental furnace data in this development process. The following discussion concentrates primarily on the final form of the furnace model. The circuit analogy for this model is illustrated in Figure 2-2.

The heating element acts as the current source for the electrical analogy. Heat generated by electrical resistance in the element flows toward lower temperature materials via the path of least resistance. Although the actual heat transfer process is a combination of radiation, conduction, and convection, it is assumed that all the heat energy from the heating element flows by conduction into the chamber area between the crucible and the outer insulating walls.

It is also assumed that there are two paths of heat flow from the chamber, one through the crucible wall to the molten aluminum, and the other through the outer insulating walls and floor and surrounding ambient air. Since the heating element cycles on and off in response to the electrical control system, a switch is placed in series with the voltage source in the electrical circuit analogy.

The first main path for heat flow from the chamber area is through the crucible wall into the molten aluminum. In modeling this path, it is assumed that there are two significant heat storage devices, the crucible which transfers heat to the aluminum and provides a

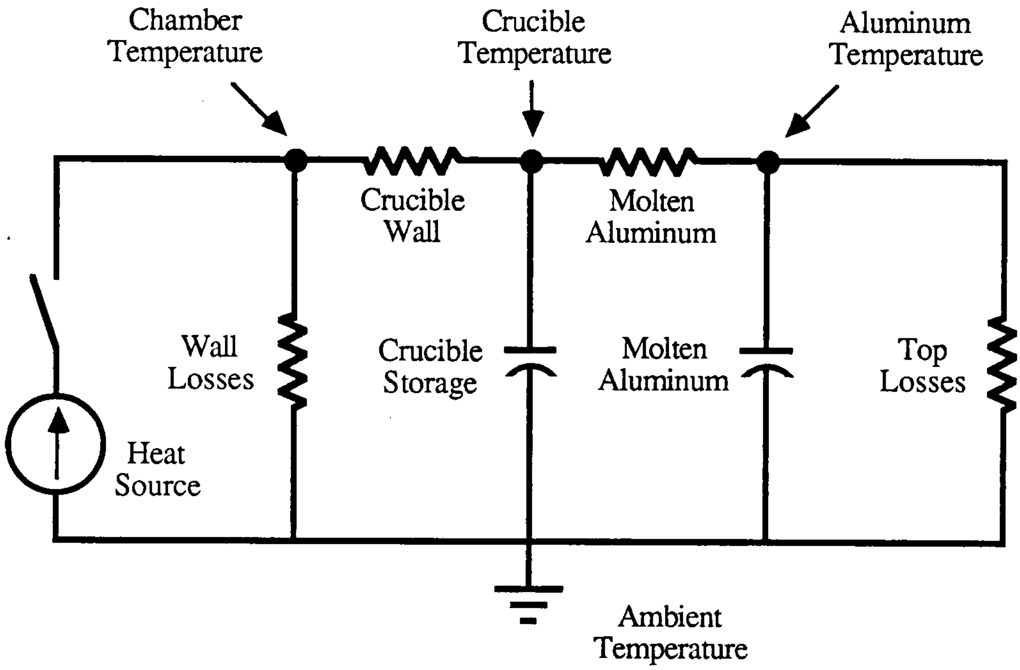


Figure 2-2. Simplified Furnace Schematic

heat-resistant structure to contain the molten aluminum, and the molten aluminum itself. Both heat storage devices are represented as capacitors in the schematic diagram.

En route to the storage devices, the heat energy encounters some resistance as it passes through the crucible wall and the molten aluminum. For the purpose of this simple model, the complex heat transfer at the material boundaries is not considered. In other words, it is assumed that there is no resistance to heat flow at these boundaries. Placement of the resistors on the circuit diagram is such that the heat energy encounters the resistance of the crucible wall before being stored in the crucible, then encounters the resistance of the aluminum before being stored in the molten aluminum.

A dome-shaped lid over the crucible acts to reduce the amount of heat escaping from the molten aluminum. As ambient air circulates over this lid, however, a small amount of heat is transferred away from the furnace. These top losses are represented by a resistor in parallel with the heat storage in the molten aluminum. Additional heat losses due to leakage around the lid and the periodic opening and closing of the lid are neglected for this model.

The second main path of heat flow from the chamber area is through the outer side walls of the furnace. The hot wall surface generates a natural convective heat flow around the furnace. This heat flow path is represented by a resistor in parallel with the chamber area. The flow of heat through the bottom of the furnace was neglected in the initial analysis.

Based on the assumptions stated previously, an electrical circuit analogy is thus established to describe the response of the furnace as a second-order system. Note that this analogy has been developed without reference to the quantification of the elements of the system. There is some implicit relative quantification, however, in the assumption that some effects are negligible. At this point, the circuit analogy in Figure 2-2 could be used to model a variety of furnaces with the same general configuration.

The next step in developing the model was to assign quantities to the elements in the circuit to represent the furnaces used in this study.

Estimation of Furnace Parameters

One of the most difficult parts of modeling the furnace system was the assignment of representative quantities to the elements in the electrical circuit analogy. Thermal capacitance was calculated by multiplying the specific heat of a material by its weight (Harrison, p. 27). Two methods were used to calculate thermal resistance. For the crucible wall and molten aluminum, this calculation was based on heat flow by conduction. For the furnace wall and top, however, the effects of convection and radiation could not be neglected.

Thermal resistance due to conduction, convection, and radiation was estimated by consulting a reference on industrial furnace design (Trinks, Vol. 2, p. 81). This reference includes a graph that presents the approximate heat loss per unit area from vertical furnace walls as a function of the temperature difference between wall surface and surrounding ambient air. The total heat loss is then found by multiplying this number by the total area of the wall. With the heat loss known, the thermal resistance can then be found by dividing the temperature difference across this resistance by the rate of heat loss (Harrison, p. 27).

Although the above method for estimating thermal resistance is best suited for vertical walls, it was also used for estimating the heat loss from the furnace top. Since the graph is based on the air circulation patterns around vertical furnace walls, it is likely the indicated heat loss is higher than would actually occur for a horizontal surface. The resulting error is offset to some extent by neglecting the heat losses associated with the lid. As stated earlier, these losses are a result of leakage around the edges and escaping heat during the opening and closing of the lid.

The following discussion describes the calculations used to quantify the elements of the furnace model, starting with the heating element. Calculations for only the 500-lb furnaces are shown; values for the 800-lb furnaces were obtained in a similar manner. The resulting values for both furnace sizes are shown in Table 2-1.

The heat source can be set to cycle either with full or half power. At full power, the heating element supplies

$$U = 46 \text{ kW} \times 0.9478 \frac{\text{Btu}}{\text{kW-s}} = 43.6 \frac{\text{Btu}}{\text{s}} \quad . \qquad (2\text{-}8)$$

At half power, the heating element output is reduced by a factor of 2 to

$$U = 23 \text{ kW} \times 0.9478 \frac{\text{Btu}}{\text{kW-s}} = 21.8 \frac{\text{Btu}}{\text{s}} \quad . \qquad (2\text{-}9)$$

Table 2-1. Calculated furnace element values.

Element	Units	500-lb Furnace	800-lb Furnace
Aluminum capacitance (*CA*)	Btu/°F	0.2727 W	0.2727 W
Aluminum resistance (*RA*)	Btu/°F	3994.0/W	8987.0/W
Crucible capacitance (*CC*)	BTU/°F	71.81	104.5
Crucible resistance (*RC*)	°F-s/Btu	5.03	3.55
Heating element (*U*)			
Full power	Btu/s	43.6	43.6
Half power	°F-s/Btu	21.8	21.8
Top resistance (*RT*)	°F-s/Btu	757.0	573.3
Wall resistance (*RW*)	°F-s/Btu	603.9	525.5

Generally, a furnace is set to operate at full power during startup and normal operation. Half power is typically used when the consumption of aluminum is low and when a furnace is in the standby mode.

To calculate wall losses, temperatures were measured on the walls of five of the furnaces at the casting company by placing a bimetallic thermometer in contact with exposed wall surface. The average of these wall temperatures was approximately 166°F; the average ambient temperature in the vicinity of the furnaces was approximately 110°F. Using the graphical method outlined previously (Trinks, Vol. 2, p. 81), a surface temperature 56°F higher than ambient corresponds to a heat loss of 170 Btu/ft^{2}h.

Measurement of many of the physical dimensions was made when one of the furnaces at the casting company was disassembled for maintenance. These dimensions are indicated in a simplified drawing of the furnace in Figure 2-3. From these measurements, the total surface area for four furnace walls was found to be:

$$A = 47 \text{ in.} \times 36 \text{ in.} \times 4 \text{ walls} = 6768 \text{ in.}^2 \quad . \qquad (2\text{-}10)$$

Total heat loss from the walls was then found by multiplying this area by the heat loss per unit area:

$$q = 170 \frac{\text{Btu}}{\text{ft}^2\text{h}} \times 6768 \text{ in.}^2 \times \frac{1 \text{ ft}^2}{144 \text{ in.}^2} \times \frac{1 \text{ h}}{3600 \text{ s}} = 2.219 \frac{\text{Btu}}{\text{s}} \quad . \qquad (2\text{-}11)$$

In a similar manner, the heat loss for the top was found by first measuring the surface temperature on several of the furnaces. Although this was a somewhat difficult measurement to obtain, the temperature was found to be in the vicinity of 250°F. Using the graphical method, a temperature 140°F higher than ambient corresponds to a heat loss of 400 Btu/ft^{2}h. Combining the calculation steps used above, the top loss was found to be

$$q = 400 \frac{\text{Btu}}{\text{ft}^2\text{h}} \times 2209 \text{ in.}^2 \times \frac{1 \text{ ft}^2}{144 \text{ in.}^2} \times \frac{1 \text{ h}}{3600 \text{ s}} = 1.704 \frac{\text{Btu}}{\text{s}} \quad . \qquad (2\text{-}12)$$

With the approximate loss from the furnace top and walls known, a value was then obtained to represent the thermal resistance at those locations. The temperature of the molten aluminum is approximately 1400°F, or 1290°F above ambient during normal operation. With a heat loss of 1.704 Btu/s, the thermal resistance of the top (R_T) was calculated to be

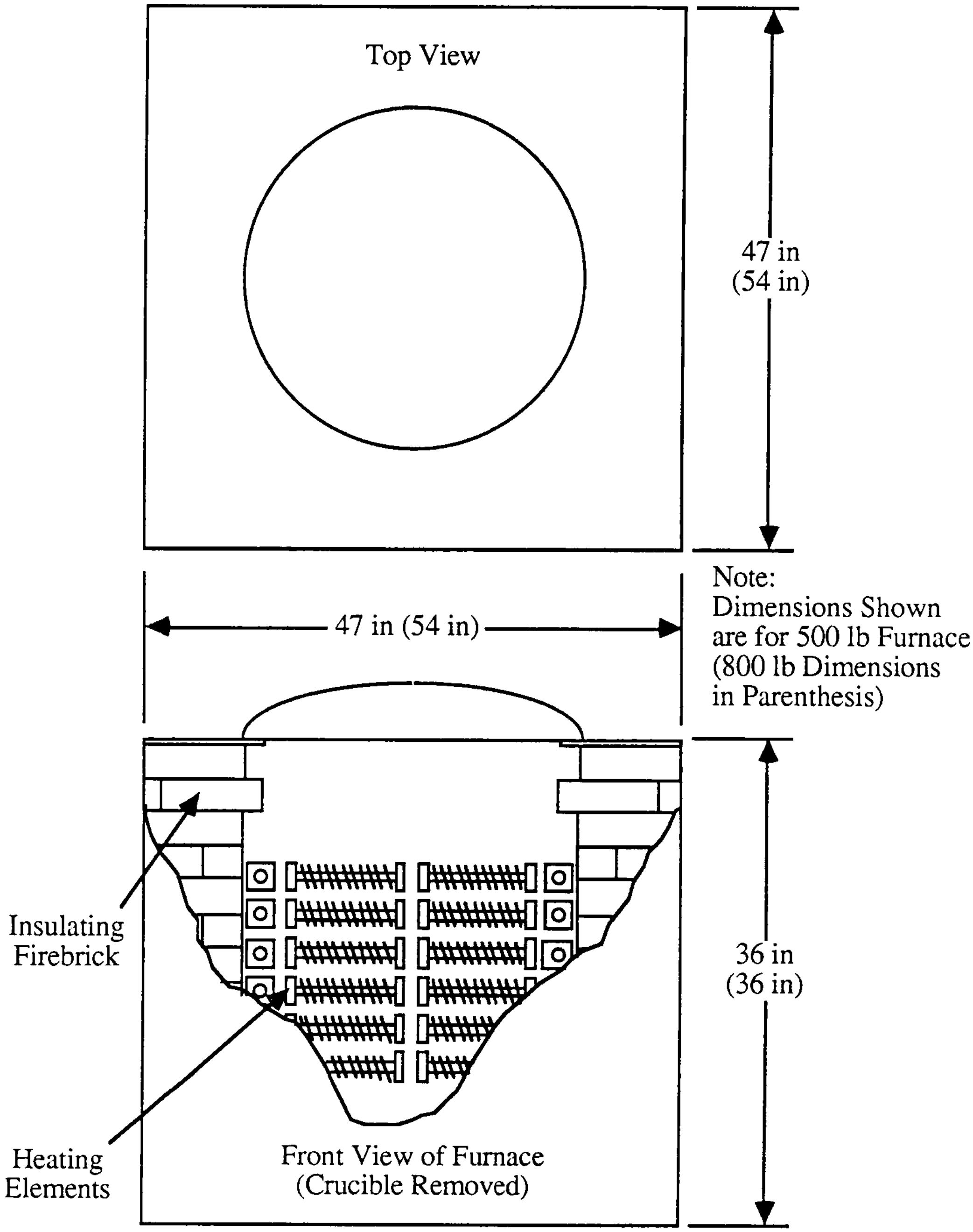


Figure 2-3. Furnace Dimensions

$$R_T = \frac{1290°F}{1.704 \frac{Btu}{s}} = 757.0 \frac{°F\text{-}s}{Btu} \quad . \tag{2-13}$$

The chamber is designed to operate 50°F higher at approximately 1450°F, or 1340°F above ambient. With a heat loss of 2.219 Btu/s, the thermal resistance of the walls (R_W) was calculated to be

$$R_W = \frac{1340°F}{2.219 \frac{Btu}{s}} = 603.9 \frac{°F\text{-}s}{Btu} \quad . \tag{2-14}$$

In calculating the thermal resistance and capacitance of the crucible, measurements were made on a crucible that was removed for replacement. These measurements are shown on a diagram of the crucible in Figure 2-4. To simplify the calculation of surface area, however, the crucible was approximated by a 24-in.-long cylinder with an outer diameter of 24 in. The surface area was calculated to be

$$A = \pi \times 24 \text{ in.} \times 24 \text{ in.} = 1809.6 \text{ in.}^2 \quad . \tag{2-15}$$

The crucible volume was estimated by assuming the crucible was constructed from the above cylinder with a uniform thickness of 2 in. and no bottom. The omission of the bottom was intended to offset the fact that the actual crucible has a diameter of 12 in. at the bottom. This estimated volume is:

$$V = \pi \times [(12 \text{ in.})^2 - (10 \text{ in.})^2] \times 24 \text{ in.} = 3317.5 \text{ in.}^3 \quad . \tag{2-16}$$

Using the equation for thermal resistance due to conduction and the thermal conductivity of silicon carbide for the crucible (Cone, p. 65), the crucible resistance (R_C) was found to be

$$R_C = \frac{2 \text{ in.}}{1809.6 \text{ in.}^2 \times 0.0002199 \frac{Btu}{in.\text{-}s\text{-}°F}} = 5.03 \frac{°F\text{-}s}{Btu} \quad . \tag{2-17}$$

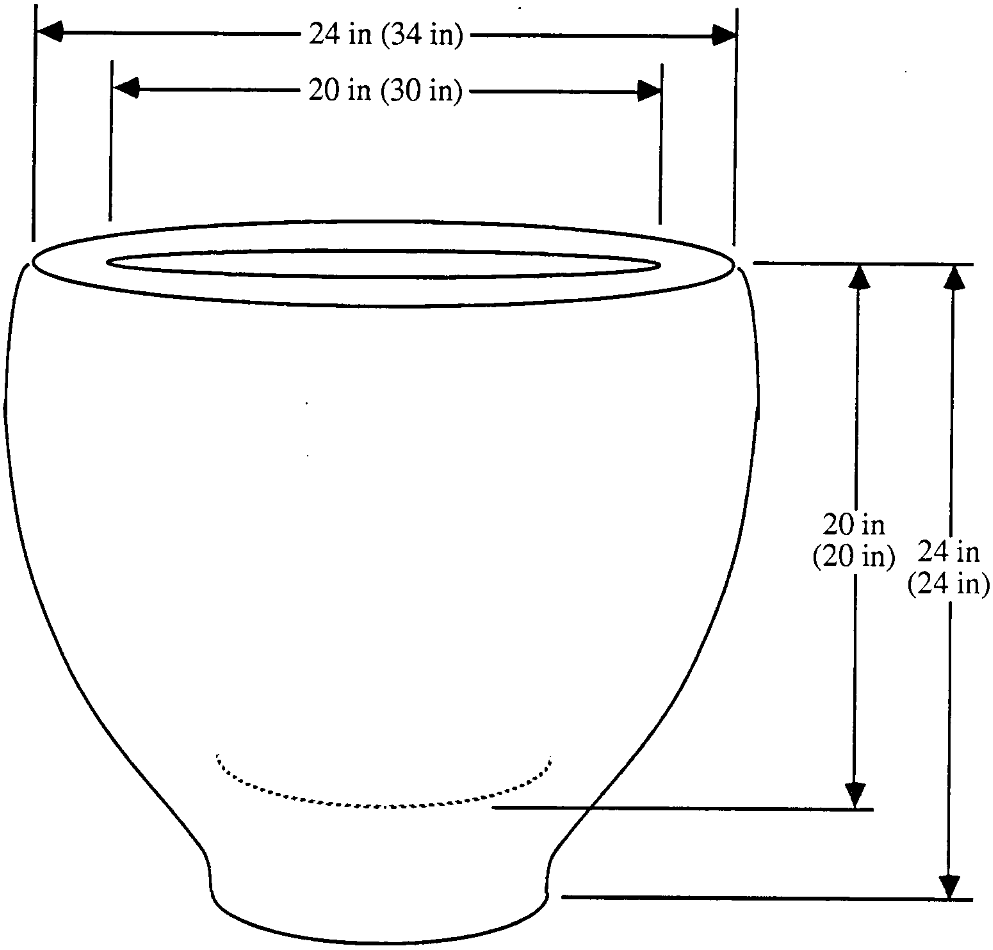


Figure 2-4. Crucible Dimensions

The capacitance of the crucible (C_C) was obtained by multiplying the specific heat of silicon carbide (Trinks, Vol. I, p. 95) by the weight of the crucible. The weight of the crucible was estimated by multiplying the volume by the density of silicon carbide (Trinks, Vol. 1, p. 94). This resulted in a crucible capacitance of

$$C_C = 0.2625 \frac{\text{Btu}}{\text{lb-°F}} \times 0.08247 \frac{\text{lb}}{\text{in.}^3} \times 3317.5 \text{ in.}^3 = 71.81 \frac{\text{Btu}}{\text{°F}} \quad . \quad (2\text{-}18)$$

Since the amount of aluminum in the crucible of the furnace varies, the thermal resistance of the aluminum was calculated as a function of weight. Using the cylindrical approximation for the crucible shape, the thermal resistance of the aluminum (R_A) is:

$$R_A = \frac{r}{Ak} \quad , \quad (2\text{-}19)$$

where r is the radius of the cylinder, A is the outside area of the cylinder of aluminum, and k is the thermal conductivity of aluminum (Hatch, p. 8). As the amount of aluminum changes, the area of the cylinder of aluminum changes. The area of this cylinder is

$$A = 2\pi r l \quad , \quad (2\text{-}20)$$

where l is the length of the cylinder. In addition, the weight of this cylinder is

$$W = p \times \text{volume} = p \times \pi r^2 l \quad , \quad (2\text{-}21)$$

where W is the weight and p is the density of the aluminum (Hatch, p. 8). Solving this equation for l and inserting in the equation the area of the aluminum cylinder results in the area as a function of weight. The resulting equation is:

$$A = 2\frac{W}{rp} \quad . \qquad (2\text{-}22)$$

This equation was substituted in the equation for aluminum resistance to make it a function of weight in the equation,

$$R_A = \frac{r^2 p}{2kW} \quad . \qquad (2\text{-}23)$$

Inserting the appropriate values, the thermal resistance is calculated using

$$R_A = \frac{(100 \text{ in.})^2 \times 0.1 \dfrac{\text{lb}}{\text{in.}^3}}{2W \times 0.001252 \dfrac{\text{Btu}}{\text{in.-s-°F}}} \quad , \qquad (2\text{-}24)$$

which simplifies to

$$R_A = \frac{3994.0}{W} \frac{\text{°F-s}}{\text{Btu}} \quad . \qquad (2\text{-}25)$$

The thermal capacitance of the aluminum (C_A) was calculated by multiplying the weight of the aluminum, which is a variable, by the specific heat of silicon carbide (Cone, p. 64). This results in a capacitance of

$$C_A = 0.2727 \times W \frac{\text{Btu}}{\text{°F}} \quad , \qquad (2\text{-}26)$$

where W is the weight of the aluminum.

The results of the calculations for all the furnace model parameters are summarized in Table 2-1 on p. 19. The same procedure was used to calculate model element values for the 800-lb furnaces. Although not shown, the results of these calculations are included in Table 2-1.

Development of Model Equations

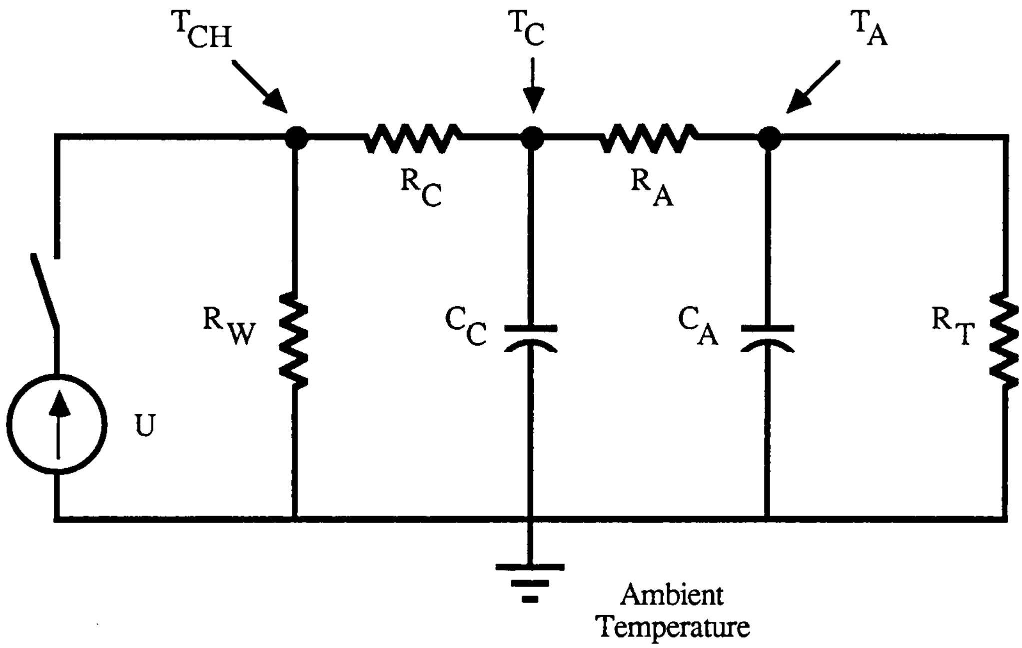
With values assigned to the elements of the furnace model, standard circuit techniques were used to reduce the model to a set of second-order equations. Because of the configuration of the electrical circuit analogy for the model, the nodal analysis technique was used (Hayt, pp. 67-77). The schematic representation of the furnace model is repeated in Figure 2-5 with the variable name assignments used in the derivation of the second-order equations.

The nodal analysis technique is based on summing the currents leaving a given node. For a thermal resistance, this current is simply the temperature difference divided by the resistance. For a thermal capacitance, however, the current leaving a node is equal to the temperature difference divided by the impedance due to capacitance. This impedance (Z_C) is equal to

$$Z_C = \frac{1}{Cs} \quad , \tag{2-27}$$

where C is the thermal capacitance and s is a differential operator corresponding to the Laplace transfer of the first derivative. In other words, s in the frequency domain is equivalent to d/dt in the time domain. Similarly, $1/s$ in the frequency domain is equivalent to an integrator in the time domain. The transformation to the frequency domain is preferred for circuit analysis because it simplifies the manipulation of nonlinear elements such as capacitance.

The three nodes used were the temperatures of the chamber, crucible, and aluminum above ambient. An equation was written for each of these nodes based on setting the sum of voltages leaving the node



C_A	Aluminum Capacitance	R_W	Wall Loss Resistance
C_C	Crucible Capacitance	T_A	Aluminum Temperature
R_A	Aluminum Resistance	T_C	Crucible Temperature
R_C	Crucible Resistance	T_{CH}	Chamber Temperature
R_T	Top Loss Resistance	U	Heat Source

Figure 2-5. Schematic Variable Assignments

equal to zero. These equations were combined to obtain an equation for the aluminum temperature as a function of heat source input.

Starting with the chamber temperature and summing currents,

$$-U + \frac{T_{CH}}{R_W} + \frac{T_{CH} - T_C}{R_C} = 0 \quad . \tag{2-28}$$

Similarly, for the crucible temperature,

$$\frac{T_C - T_{CH}}{R_C} + \frac{T_C}{\frac{1}{C_C s}} + \frac{T_C - T_A}{R_A} = 0 \quad . \tag{2-29}$$

And, finally, for the aluminum temperature,

$$\frac{T_A - T_C}{R_A} + \frac{T_A}{\frac{1}{C_A s}} + \frac{T_A}{R_T} = 0 \quad . \tag{2-30}$$

Rearranging Equation (2-29) and inserting it into Equation (2-28) results in

$$-U + \frac{T_{CH}}{R_W} + T_C C_C s + \frac{T_C - T_A}{R_A} = 0 \quad . \tag{2-31}$$

Next, Equation (2-28) is solved for T_{CH} to yield

$$T_{CH} = R_W \left(\frac{R_C U + T_C}{R_C + R_W} \right) \quad . \tag{2-32}$$

Equation (2-32) is then inserted into Equation (2-31) to remove the dependence on T_{CH}:

$$-U + \frac{R_C U + T_C}{R_C + R_W} + T_C C_C s + \frac{T_C - T_A}{R_A} = 0 \quad . \tag{2-33}$$

Multiplying this equation by $(R_C + R_W)$ and rearranging terms yields

$$-R_W U + \left[1 + \frac{R_C + R_W}{R_A} + (R_C + R_W) C_C s\right] T_C + \left[\frac{R_C + R_W}{R_A}\right] T_A = 0 \quad . \qquad (2\text{-}34)$$

Next, Equation (2-30) is multiplied by $(R_A R_T)$, and terms are rearranged to obtain an equation for T_C as a function of T_A:

$$T_C = \left[1 + \frac{R_A}{R_T} + R_A C_A s\right] T_A = 0 \quad . \qquad (2\text{-}35)$$

Substituting Equation (2-35) into Equation (2-34) gives:

$$-R_W U + \left[1 + \frac{R_C + R_W}{R_A} + (R_C + R_W) C_C s\right] \left[1 + \frac{R_A}{R_T} + R_A C_A s\right] T_A - \left[\frac{R_C + R_W}{R_A}\right] T_A = 0 \quad . \qquad (2\text{-}36)$$

This equation is then solved for T_A/U to obtain a second-order transfer function for the furnace model. This equation is of the form

$$\frac{T_A}{U} = \frac{N}{s^2 + D_A s + D_B} \quad , \qquad (2\text{-}37)$$

where

$$N = \frac{R_W}{R_A C_A (R_C + R_W) C_C} \quad , \qquad (2\text{-}38)$$

$$D_A = \frac{1}{(R_C + R_W) C_C} + \frac{1}{R_A C_C} + \frac{1}{R_A C_A} + \frac{1}{R_T C_A} \quad , \qquad (2\text{-}39)$$

and

$$D_B = \frac{1}{R_A C_A C_C (R_C + R_W) + \dfrac{1}{R_T C_A C_C (R_C + R_W)}} + \frac{1}{R_A R_T C_A C_C} \quad . \qquad (2\text{-}40)$$

Equations (2-37) through (2-40) were used in the initial modeling of the furnace system. At this stage of development, the aluminum resistance R_A and capacitance C_A were the only parameters that were a function of weight. All others were fixed and to be considered constant in the model.

Computer Simulation of Furnace Model

Once the model equations were established, a computer was needed to study the response of the system. As stated earlier, the development of this model was an iterative process. The final modeling was accomplished on a VAX computer using the Advanced Continuous Simulation Language (ACSL) software. However, since there is some cost involved with using a VAX, the initial modeling was done on a Macintosh personal computer using the True Basic software.

Along with developing and troubleshooting the furnace model, equations had to be written to simulate the operation of the thermostat. Because the furnaces used as the basis of this study had a deadband of approximately 10°F, this was set as the requirement for the thermostat of the model.

The thermostat model was started by sensing the chamber temperature, T_{CH}, to correspond with the operation of the actual furnaces. This temperature was compared to a setpoint to determine the value to be assigned to the heat source, U. With the temperature below the setpoint, the heat source was assigned a value corresponding to either full or half power (depending on the desired setting). When the temperature was higher than the setpoint, the heat source was assigned a value of zero. This arrangement provided the correct operation of the setpoint but did not include the deadband.

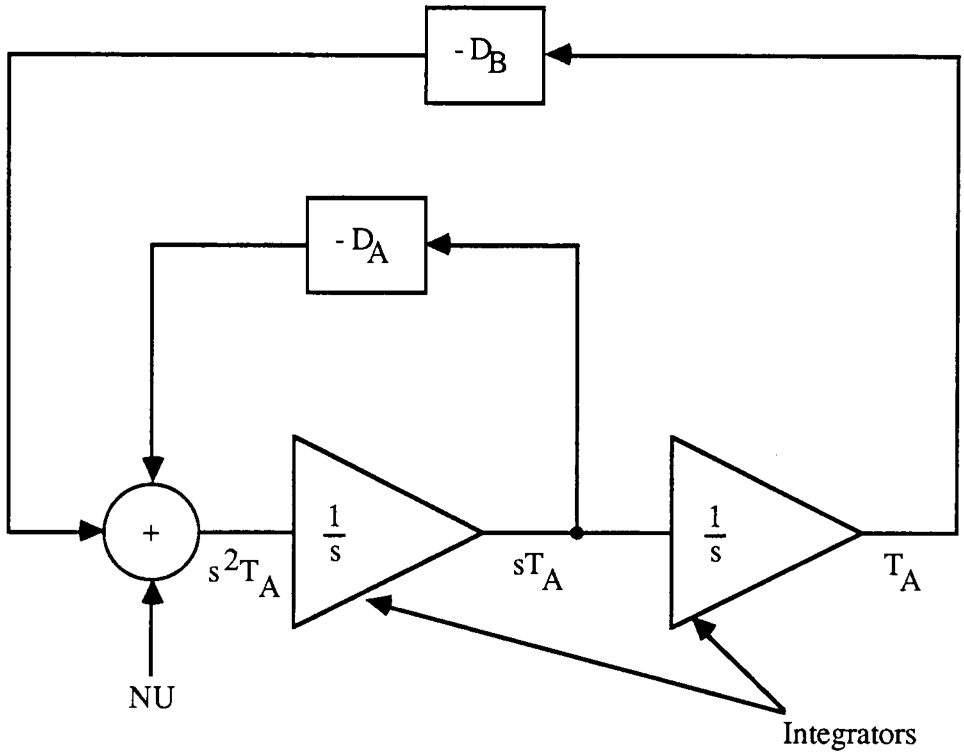
The thermostat deadband was added by dividing the setpoint into two parts, one for rising temperature and one for falling temperature. The setpoint for increasing temperature was 10°F higher than the setpoint for decreasing temperature. As the temperature increased past the upper setpoint, the heat source was assigned a value of zero and the temperature began to fall. The falling temperature caused the thermostat to switch to the lower setpoint to provide the required deadband, and the process continued in a similar manner. The thermostat model sensed temperature change by comparing the temperature at a given time to the temperature sensed on the previous time increment in the computer simulation.

A simple technique was used to convert the second-order model equation to the statements required for the computer simulation. First, Equation (2-37) was rearranged by cross-multiplying and solving for the highest derivative, s^2 in this case. This resulted in

$$s^2T_A = NU - sT_AD_A - T_AD_B \quad . \qquad (2\text{-}41)$$

The simulation was then constructed by assuming that the highest derivative was available and integrating twice. The result of the first integration is sT_A and that of the second is T_A. These results, along with the heat source input, were multiplied by the appropriate constants and summed to obtain s^2T_A, which was assumed initially. This procedure is shown graphically in Figure 2-6.

Two different methods were used to perform the integration shown in the furnace model. For the BASIC program, the integration was accomplished by summing the area under the curve of the quantity to be integrated. In this process, summation was made by multiplying the



$$s^2T_A = NU - sT_A D_A - T_A D_B$$

Figure 2-6. Furnace Model Process

time increment by the integrand and adding to the previous sum. The ACSL program uses a fourth-order Runge-Kutta procedure for integration. Although a full explanation of this process is beyond the scope of this discussion, it is based on using a weighted average of four different slopes of the line representing the function being integrated (Beyer, pp. 468-470). Of the two methods of integration, the one used by the ACSL program results in the smaller error.

Program listings for the computer simulation of the furnace model are included in Appendix A. Many of the program lines include comment statements which are intended to improve the readability and understanding of the program. The BASIC program includes a significant amount of complexity because a graphical interface had to be written to assist in the evaluation of the simulation data. As a matter of fact, the BASIC program actually performs the simulation twice, once to determine the bounds for plotting and once to actually display the simulation result. In addition, the BASIC program performs an algorithm to ensure that grid lines for plotting occur at numbers divisible by 10. The ACSL program does not require this additional complexity because of built-in plotting and scaling functions.

For a number of reasons, the ACSL program executes much quicker than the BASIC program. For this reason, the BASIC program was used to model only one furnace at a time. Once the desired response was obtained, the model was then transferred to the ACSL program where all six furnaces were simulated simultaneously.

In actual practice, the operation of the furnaces is such that the weight of aluminum fluctuates as molten aluminum is used and replenished. The model was made more realistic by making the weight fluctuate with time. To be precise, the model would have to include the effect of the phase change of the room-temperature aluminum used to restock the furnaces, but this adds a complexity beyond the scope of this model.

Weight fluctuation was accomplished by using a random number generator in the ACSL program. At regular intervals in the simulation, three random numbers were generated. The first was used to select one of the furnaces. The weight of the selected furnace was then changed by an amount proportional to one of the other two random numbers, depending on the size of the furnace.

Verification of Model

On the first attempt to run the simulation, a problem was encountered with thermostat placement. With the original configuration of the model, the thermostat cycled off and on hundreds of times per second, instead of once per minute as expected, because the thermostat sensed chamber temperature. The assumption was made that all energy from the heating element was transferred directly into the chamber. When placed in the chamber, the thermostat was reading the temperature of the heating element. Since the time lag for the heating element was disregarded for the purpose of the model, this placement caused the thermostat to turn off every time the heating element was energized and turn on every time the heating element was deenergized.

In the actual furnace, the temperature sensor is located in the lower part of the chamber in a protective enclosure. If a more exact model of the furnace were required, the dynamics of the thermostat would need to be modeled as a first- or second-order system. For this simplified model, however, a compromise was made to eliminate further complexity. The thermostat in the model was moved to the wall between the crucible and the aluminum to provide a more realistic response.

With the thermostat placement problem resolved, the model performed better but was still somewhat different from the actual furnaces. With the thermostat on the model set at 10°F, the cycling time of the model was slower than measured cycling time. This was corrected by reducing the resistance representing the wall losses (R_W). This corresponds to increasing wall losses attributed to an air vent in the furnace wall, which was neglected in the model formulation.

Inspection of the graphical output from the simulation reveals that the crucible temperature changes abruptly when the thermostat cycles on and off. The actual chamber temperature, however, is much slower to change when the thermostat cycles. This difference can be explained by the fact that the model represents the furnace by concentrated elements with a limited number of nodes, while the furnace is actually made up of an infinite number of nodes distributed in three dimensions.

An example of the graphical output from the ACSL program for modeling six furnaces is shown in Figure 2-7. In this figure, the crucible temperature response for each of the six furnaces is shown. Crucible temperature is shown because it is the temperature used by the thermostats in the model. It can be seen that the cycling times for furnaces 1 and 6 are noticeably longer than for furnaces 2, 3, 4, and 5

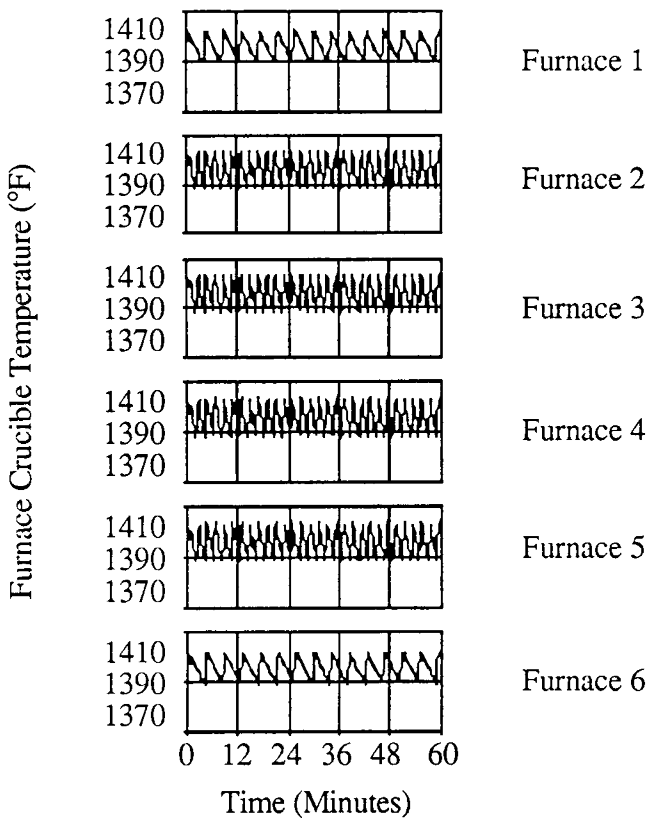


Figure 2-7. Six Furnace Simulation

because furnaces 1 and 6, which are 800-lb furnaces, are larger and slower than the 500-lb furnaces. Cycling times shown are within a few seconds of the actual times measured on the furnaces.

The length of this particular simulation is 3600 seconds (1 hour). Although the simulation could have been run longer, it becomes difficult to display the results as time increases. In addition, the simulation of six furnace systems simultaneously is computationally intensive and can result in a significant computer operating expense.

Initially, the simulation did not show a significant change in cycling time when the random weight disturbances were introduced. To make the cycling time a stronger function of weight, the resistance due to top loss (R_T) was made a function of weight. This change was partially justified by the fact that actual weight disturbances are accompanied by the effect of adding cooler material to the furnaces. In addition, variations in the cycling times of the furnaces were needed to provide a more thorough test of the coordinated control system.

A simulation showing the response to random weight disturbances is shown in Figure 2-8. In this figure, the individual furnace weights are shown along with the crucible temperature response for a 1-hour simulation run. It can be seen that random weight disturbances resulted in discontinuities in the crucible temperature. This effect is believed to be caused by introducing instantaneous changes to coefficients in the second-order equation and by top losses that were made a stronger function of weight. The discontinuities in the crucible temperature were considered allowable, since the furnace

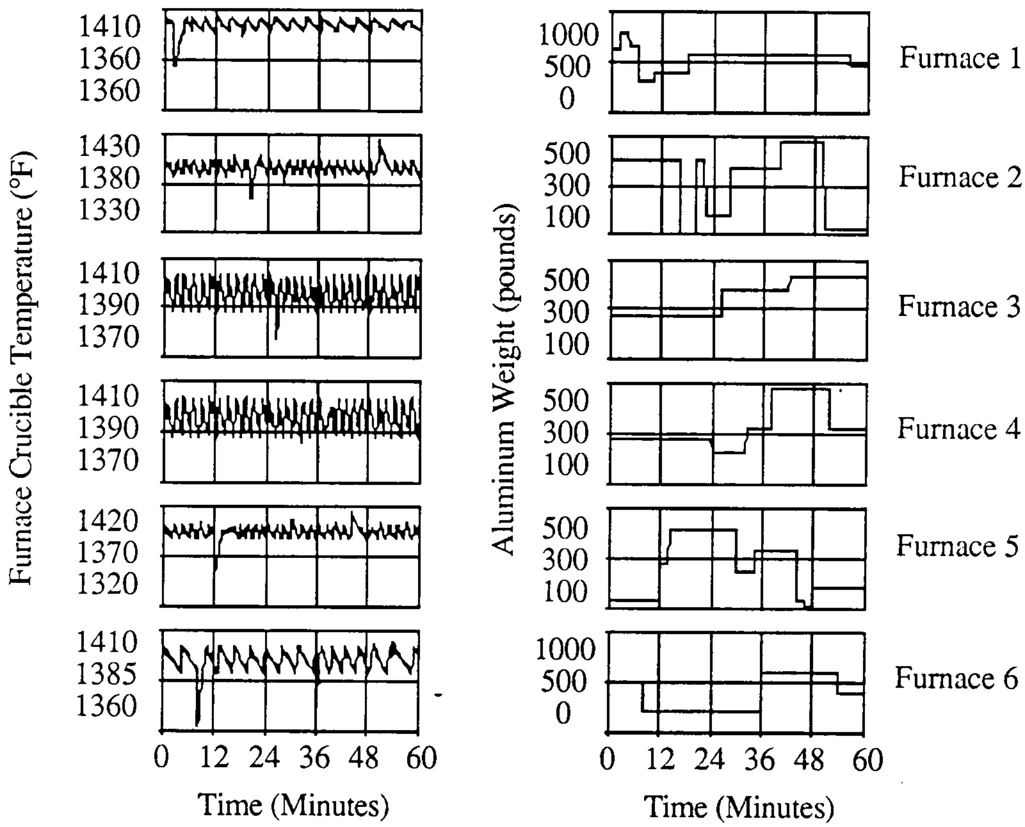


Figure 2-8. Six-Furnace Simulation with Random Weights

control is based on cycling intervals and not on continuous temperature functions.

These changes to the model caused the final version of the model to be somewhat different from the theoretical model. Some differences were to be expected, however, since the model formulation was based on approximations and simplifications. The values used for the furnace elements in the final form of the model are shown in Table 2-2.

Table 2-2. Revised furnace element values.

Element	Units	500-lb Furnace	800-lb Furnace
Aluminum capacitance (*CA*)	Btu/°F	0.2727 W	0.2727 W
Aluminum resistance (*RA*)	Btu/°F	3994.0/W	8987.0/W
Crucible capacitance (*CC*)	Btu/°F	71.81	104.5
Crucible resistance (*RC*)	°F-s/Btu	5.03	3.55
Heating element (*U*)			
Full power	Btu/s	43.6	43.6
Half power	Btu/s	21.8	21.8
Top resistance (*RT*)	°F-s/Btu	200.0 + (W-100.0)	200.0 + (W-100.0)
Wall resistance (*RW*)	°F-s/Btu	200.0	300.0

CHAPTER III

ALGORITHM DEVELOPMENT

In developing the algorithm for the coordinated control system, there were a number of factors to be considered. This chapter presents a discussion of these factors before describing the algorithms used. One of the most important factors is the cost of electrical power, since it is a basis for judging the effectiveness of the coordinated system.

Electrical Power Cost Function

As stated earlier, the cost of commercial electrical power is computed on the basis of total power consumption as well as peak demand for a given month. In this study, the cost of electricity at the casting company was used as a basis for determining cost savings. Cost savings in other applications may vary, depending on the cost of electricity in the local area. Power is supplied to the casting company by the Sevier County Electric System, Sevier County, Tennessee. Charges for electrical power to the casting company are shown in Table 3-1.

As a figure of merit for measuring the effectiveness of a control algorithm, an equation was obtained to determine a cost function for electrical power. There are three parts to the electric power cost function: the fixed customer charge, the charge for meter usage, and the charge for peak demand. The fixed customer charge is a constant unaffected by the amount or rate of power usage.

Table 3-1. Commercial electric rates.

Customer charge	\$ 8.80
Meter usage	
First 15,000 kWh	\$ 0.06278/kWh
Additional kilowatt-hours	\$ 0.03034/kWh
Demand charge	
First 50 kW of demand	\$ 0.00 (no charge)
Next 950 kW of demand	\$ 9.50/kW
Next 1,500 kW of demand	\$ 9.92/kW
Over 2,500 kW of demand	\$ 10.34/kW

Source: Sevier County Electric System electric rates, January 2, 1988.

As shown in Table 3-1, the cost per kilowatt-hour is different for meter usage in excess of 15,000 kWh. To account for this in the meter usage part of the cost function, the unit step function was used. This function is defined as follows:

$$u(t) = 0 \text{ for } t < 0 \quad , \tag{3-1a}$$

$$u(t) = 1 \text{ for } t \geq 0 \quad , \tag{3-1b}$$

Using this function, the cost for meter usage (C_M) is calculated with

$$\begin{aligned} C_M &= (0.06278)d \\ &+ [0.03034(d - 15{,}000) - 0.06278(d - 15{,}000)]\ u(15{,}000 - d) \quad , \end{aligned} \tag{3-2}$$

where d is the number of kilowatt-hours used. The unit step function causes the second term to take effect when the meter usage exceeds 15,000 kWh. In actual practice, the number of kilowatt-hours used is read directly from the electric meter. In the simulation program, however, this number was obtained by integrating total power over time.

In practice, the cost of meter usage calculated above is applied to the amount of power consumed in a 1-month period. The simulation used in this example, however, ran only for a fraction of that period, so this cost was normalized to a 1-month period to obtain a normalized meter cost (C_{MN}). This cost was calculated using

$$C_{MN} = \frac{(2,592,000)}{t} C_M \quad , \tag{3-3}$$

where 2,592,000 is the number of seconds in a month and t is the length of the simulation in seconds.

The power demand is measured at the monthly reading of the electric meter. A needle on the meter provides an indication of power demand at any given time. A second needle moves with the demand needle as long as demand is increasing. When demand decreases, however, the second needle remains in place. Hence, the second needle gives an indication of the peak power demand between meter readings. After the meter is read, this peak demand indicator is reset. Because the meter scale is not calibrated in kilowatts, a multiplier is applied to the number read from the demand meter to obtain the power demand in kilowatts.

Although there are rate changes for higher levels of demand, the power demand for the casting company has never been more than 500 kW, so the lowest demand rate is all that is needed in the cost function. The cost for power demand (C_D) was computed as follows:

$$C_D = 9.50 \times (\text{peak demand in kW} - 50) \quad . \tag{3-4}$$

If the demand were such that different rates were applicable, a unit step function would be used to account for the rate changes.

The resulting cost function which is used as a figure of merit for comparison of alternative algorithms, will be referred to as the effective monthly cost (*EMC*). It was calculated using

$$EMC = \$8.80 + C_{MN} + C_D \quad . \tag{3-5}$$

It should be noted that small differences in the cost of meter usage during a simulation run will be multiplied by the number of hours in a month and may thus become substantial.

Handling Priorities

Since reducing demand results in reducing the number of furnaces on at one time, some form of priority scheme had to be used. The most obvious scheme is to use a first-come, first-served procedure. In this application, there may be times during which the cycling of one furnace is more important than the others. While it might be desirable to be able to program or select certain furnaces for priority, this adds additional complexity to the control system and makes it more difficult to operate. Hence, the decision was made to use the first-come, first-served procedure to determine priority for receiving power. Depending on the implementation of the furnace control system, it might be possible to reconfigure the system for some other priority scheme if it is found necessary or desirable.

Hardware Considerations

It was necessary to give some consideration to the hardware used before the algorithm could be developed completely. Originally, a simple digital state machine was considered because of its low cost, simplicity, and ease of implementation. Based on state variables such as thermostat requests and total power, the machine makes transitions between logical states to distribute power based on the developed algorithm. Although conceptually simple, the state machine is not flexible as far as changes to the algorithm are concerned. To make changes, one needs access to equipment to program devices such as programmable read-only memories (PROMs).

A convenient approach to implementation is to use an IBM-compatible personal computer (PC). The large numbers of these devices available from different manufacturers have made their prices very reasonable. The PC has the advantage of being programmable in BASIC, which makes changes to the algorithm simple. In addition, there is a real-time clock that can be accessed with BASIC to monitor furnace cycling times. Expansion devices are available to connect the PC to the independent furnace controllers to provide the necessary control functions.

Ideally, this control system is implemented as a development system in which the algorithm can be tested and adjusted on the actual system. The PC approach is well suited for this type of system since control system adjustments can be made by changing a BASIC program. For these reasons, the capabilities of the PC implementation were kept in mind as the algorithm was being developed. It is possible, however, that the system could be implemented with other combinations of hardware and software.

Sampling Rate

In any control system that uses a digital computer to monitor inputs and control outputs, some consideration must be given to the sampling rate of the input data. Based on the assumptions used in this design study, sampling amounts to determining if a thermostatic switch is on or off for each of the furnaces. This is the equivalent of sampling a square wave with a frequency approximately equal to the furnace cycle frequency.

To obtain an accurate representation of the input function, the Nyquist criterion states that the sample frequency should be at least twice the highest frequency in that function (Defatta, p. 13). The worst case for this example is a square wave with a period equal to the "on" cycle time of the thermostat for the 500-lb furnaces. This period is generally around 20 or 30 seconds, which would be equivalent to a square wave with a minimum frequency of 3 Hz. Consequently, the minimum sampling frequency for this example will be twice that frequency, or 6 Hz. This requirement does not place a serious limitation on the hardware used for the control system and can easily be met with the PC implementation.

Failure Modes

One important consideration in the development of a control algorithm is behavior of the coordinated control system in the event of a failure. In the example used in this study, a potential exists for a tremendous loss if the system is improperly controlled. The replacement cost for six of the furnaces used in this application is on the order of $100,000, not including the cost of expensive aluminum alloys that could be damaged by a control system failure.

The effect of not allowing a furnace to cycle on is an uncontrolled cooldown. The molten metal in a furnace solidifies and expands as the furnace cools down. This expansion destroys the furnace crucible, which is made of a brittle ceramic material. This failure can then cause additional damage to the heating elements and furnace structure which surround the crucible.

A failure of the control system to allow a furnace to cycle off will cause the furnace to overheat, resulting in damage to the molten metal as well as to the furnace itself. Although it is beyond the scope of this study to add additional high temperature protection, the system should not keep the furnace from cycling off when the thermostat reaches its upper limit.

Bypass Mode

There are several cases in which a coordinated control system needs to be bypassed. One is a failure of the system itself. As mentioned previously, it is assumed that the furnace is put in a safe state when control is returned to the individual control systems. Another case is during maintenance or special testing in which an operator cycles the furnace locally by operating the individual furnace controller. In either case, the furnace controller should operate as though it were not attached to the coordinated control system.

To improve the reliability with which the coordinated system returns control to the individual controller, interface between the two systems should be minimized. This requirement is based on the assumption that fewer connections provide fewer places for a failure to occur. With these requirements established, it was determined that the control system could be implemented with only two interface points. One is a sensing point to indicate the position of the thermostatic switch, the other a device to keep the local controller from cycling on.

The sensing point can be a high impedance device which can be left connected in the bypass mode with no effect on the local controller. The actual control interface can be placed in series with the thermostatic switch and opened as necessary to interrupt the "on" cycle of the local controller. In order that the device fail in the safe condition, the control interface should have contacts that are normally closed. With this arrangement, a loss of control signal from the coordinated control system allows the contacts to close and return control to the local system.

Self-Checking

In establishing the bypass mode, it was determined that the system could be made to fail in a safe condition if normally closed contacts were used to remove the coordinated control system from the local controller. There is a condition, however, in which this does not offer sufficient protection for the local controller: the one in which the program that executes the control algorithm fails during execution. A failure of this type can be caused by an error in the control program or any of a number of failures in the digital device executing the program.

To provide protection from this type of failure, a self-checking feature is used to monitor the control program and remove the coordinated control system in the event of failure. A simple implementation of this feature is a timer that resets each time the control algorithm completes a loop. A failure of this timer to reset indicates a failure of the algorithm to complete a loop. After a predetermined time limit, the individual control systems are isolated from the coordinated system. The timer is located outside the hardware that executes the algorithm to provide independent self-checking.

Algorithms Considered

With these considerations in mind, several algorithms were developed. Among these was one which allows only a fixed number of furnaces on at a time. A variation of this algorithm is one that attempts periodically to adjust the fixed number to the minimum number that will maintain temperature. A third algorithm sets the maximum number of furnaces at three, but offers the simplest implementation. The following sections describe these algorithms.

Fixed Maximum Algorithm

The Fixed Maximum Algorithm is perhaps the most obvious solution to limiting the peak power demand. This algorithm services each furnace in sequence and makes a decision based on the condition of the furnace and whether or not the maximum number of furnaces has been reached. A block diagram of this algorithm is shown in Figure 3-1.

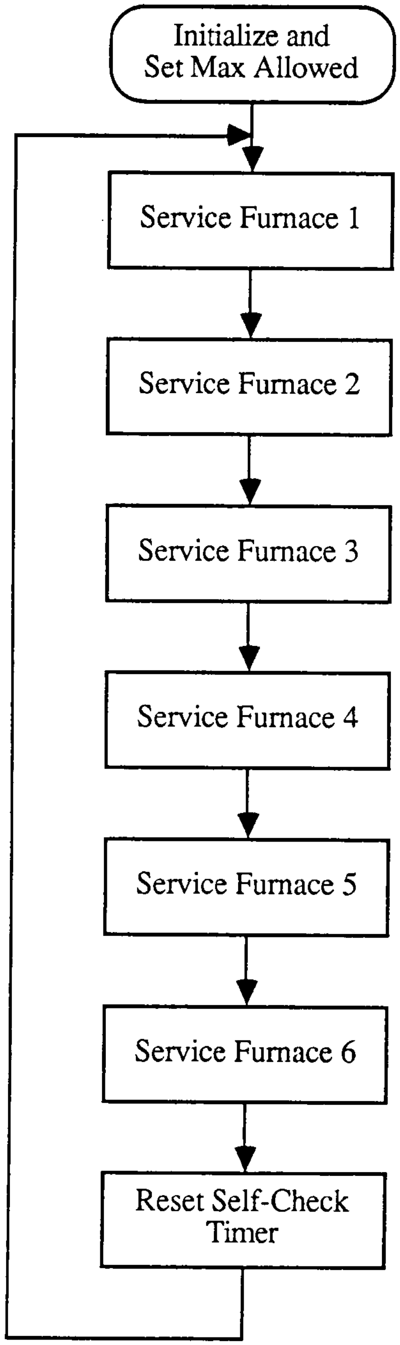


Figure 3-1. Fixed Maximum Algorithm

After being initialized and having set the maximum allowed, the algorithm operates in a continuous loop. After servicing the furnaces in sequence, an external timer is reset before repeating the loop. In the event the self-check timer is not reset, the coordinated control system is removed from the individual controllers after a nominal time delay.

Figure 3-2 shows a detailed diagram of the decision process that takes place as each furnace is serviced. If the furnace is requesting power, the algorithm checks to see if the maximum number of furnaces has been granted power. If so, no action is taken and the next furnace is serviced. If not, the algorithm checks to see if the furnace has been granted power. If it has not, the furnace is granted power, and the number of furnaces on is incremented. Otherwise, the algorithm moves to the next furnace.

If the furnace is not requesting power, the algorithm checks to see if the furnace has been granted power. If it has, the power grant is released, and the number of furnaces on is decremented. If it has not been granted power, no action is taken, and the algorithm proceeds to service the next furnace.

Floating Maximum Algorithm

The next algorithm is a variation on the previous algorithm except that it automatically resets the maximum number of furnaces allowed on by integrating the difference between the current maximum allowable (P_{MAX}) and the instantaneous total power (P_T) over time. The average power available (P_{AV}) is calculated using

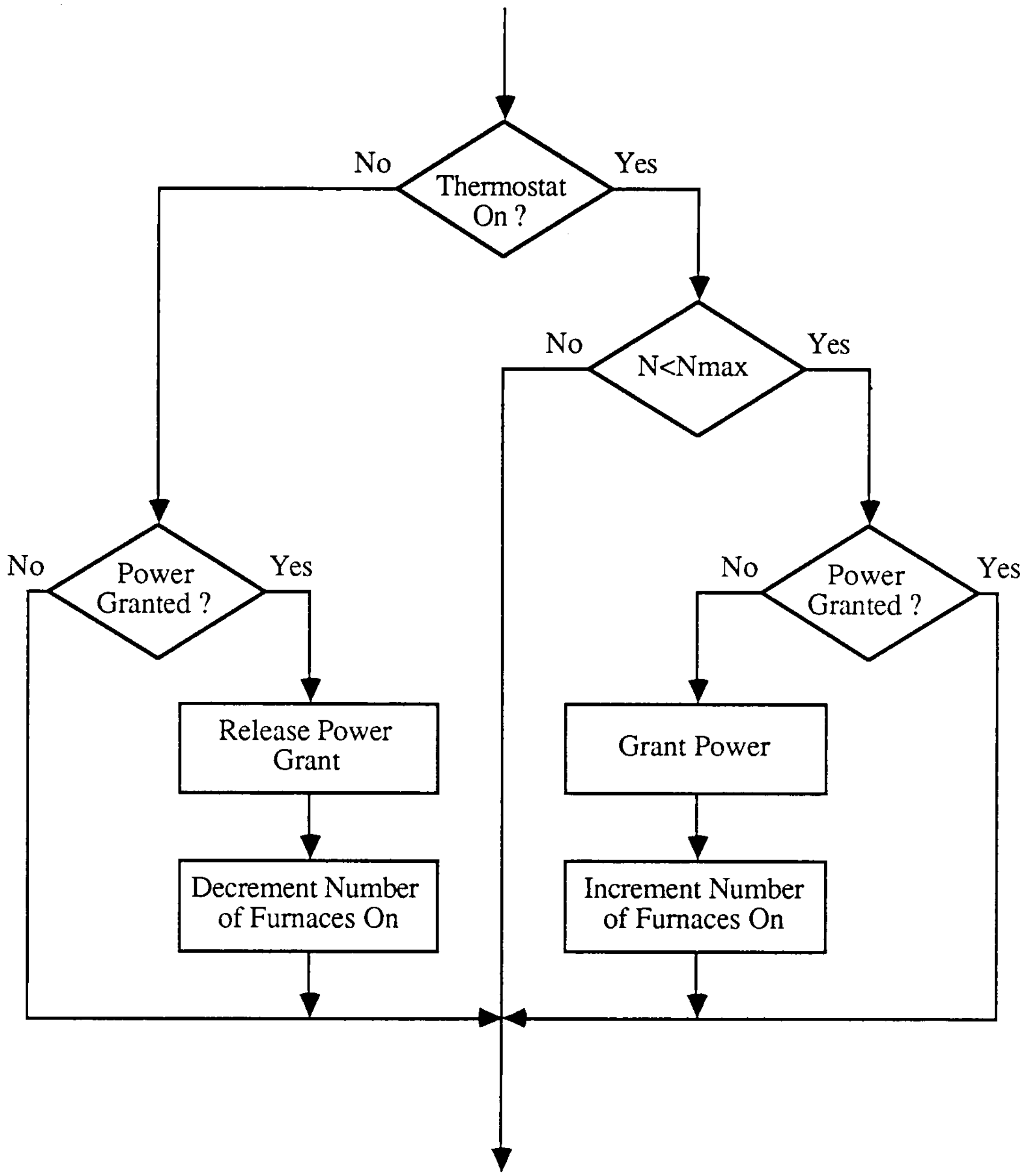


Figure 3-2. Fixed Maximum Detail

$$P_{AV} = \frac{1}{t}\int_0^t (P_{MAX} - P_T)dt \quad . \tag{3-6}$$

If this average is significantly more than the power required for one furnace, the maximum number of furnaces allowed is incremented. A block diagram of this algorithm is shown in Figure 3-3. Servicing of the individual furnaces is similar to that of the Fixed Maximum, as shown in Figure 3-2. Periodically during the operation of the control system, P_{AV} is calculated to determine if P_{MAX} should be changed. This calculation was performed every 10 minutes for this study, although it should be experimentally adjusted in a practical implementation.

If the power available is greater than that needed for two furnaces (P_2), P_{MAX} is reduced so that P_{AV} is less than or equal to P_2. If the power available is less than the amount needed for one furnace (P_1), P_{MAX} is increased so that P_{AV} is greater than or equal to P_1. The algorithm then continues through the self-check timer and repeats the loop.

The reasoning behind the Floating Maximum is that it always keeps the system set to a conservative maximum number of furnaces. Although the power cost is based on the peak maximum for an entire month, this algorithm ensures that this maximum is always set as low as possible. The same result could be obtained with the Fixed Maximum Algorithm, but it would require constant monitoring and adjustment of the system.

Buddy System Algorithm

A very simple and inexpensive algorithm is to pair the furnaces, effectively setting a fixed maximum of three furnaces. This is more limiting than the Fixed Maximum Algorithm, however, since each furnace

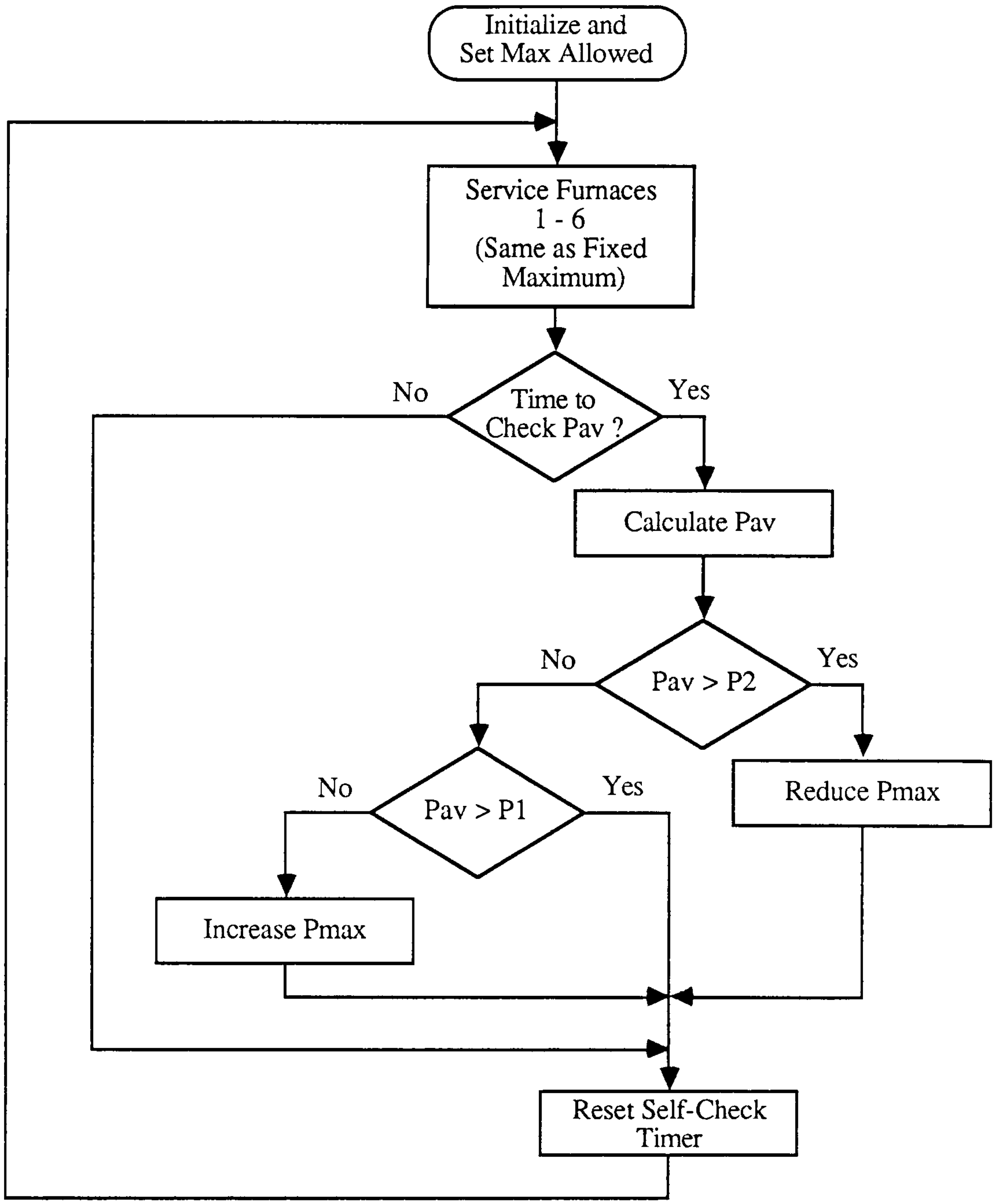


Figure 3-3. Floating Maximum Algorithm

is in contention with one specific furnace for power. With the Fixed Maximum Algorithm, the furnaces contend for power as a group, thus increasing their opportunities to receive power, since the only requirement is for any three other furnaces to be off.

In this example, the furnaces were paired in order: Furnace 1 with Furnace 2, Furnace 3 with Furnace 4, and Furnace 5 with Furnace 6. Since Furnaces 1 and 6 are larger and have longer cycling times than the others, Furnaces 2 and 5 have a lower probability of receiving power than Furnaces 3 and 4. This disadvantage is offset to some extent by ease of implementation.

A flowchart of this algorithm is shown in Figure 3-4. If the thermostat for the local thermostat is on, the state of the other furnace is checked, and if it is on, the loop is repeated. If the other furnace is off, however, the local furnace cycles on and locks out the other furnace. If the thermostat for the local furnace is off, the algorithm checks the state of the local furnace. If the local furnace is on, it is turned off and the other furnace is enabled. If it is off, the loop is repeated in a similar manner.

Breaking Up Synchronized Furnaces

Originally, it was thought that there might be a tendency for the furnaces to begin to cycle together. If two or more furnaces began to cycle together, there would be a tendency for some of the furnaces always to be denied power, which would require an addition to the algorithm to break up the synchronization.

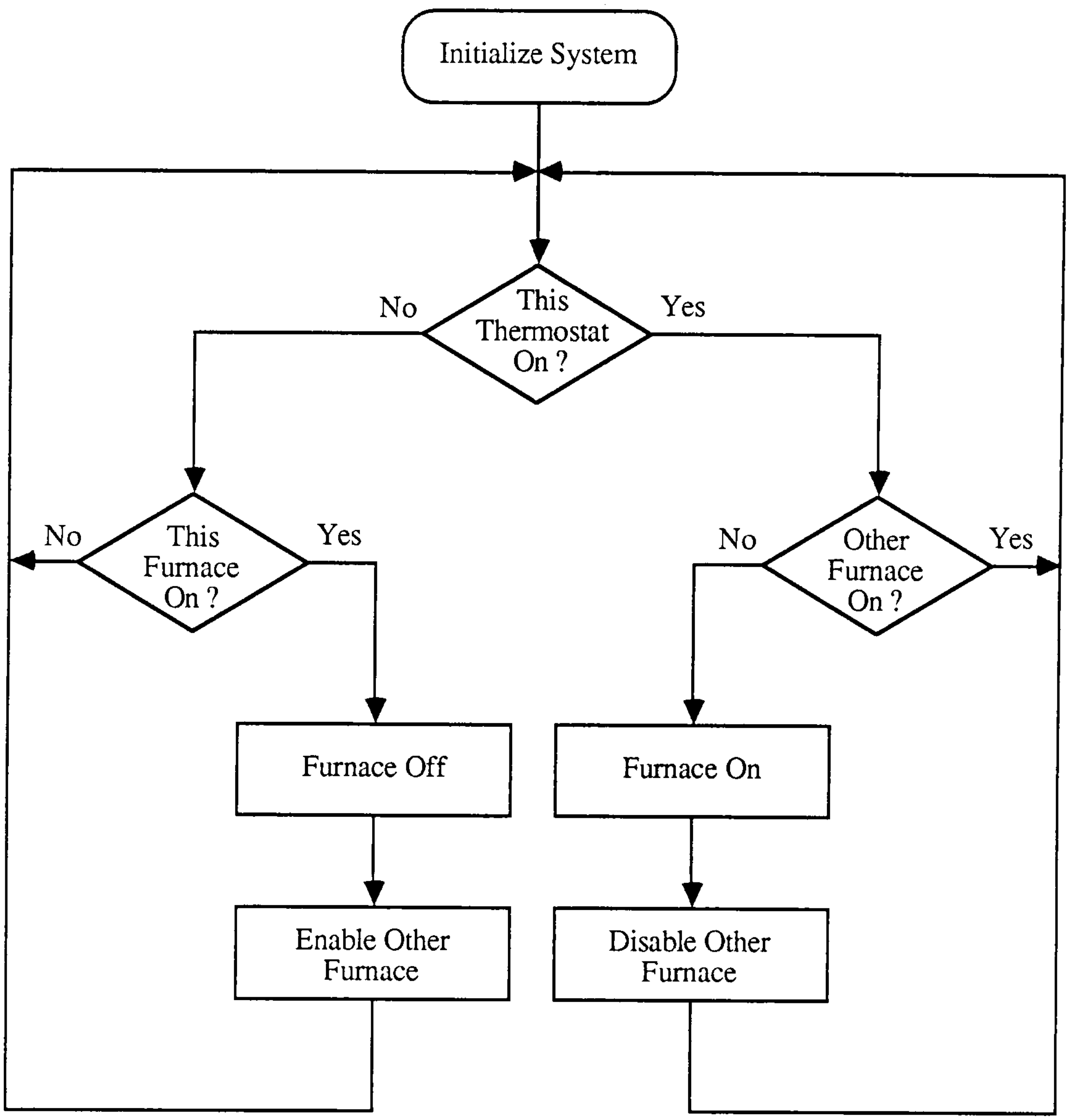


Figure 3-4. Buddy System Algorithm

After studying the simulation results, however, it was found that this problem took care of itself for all the algorithms tested. When two or more furnaces were cycling together, the first furnaces to request power received a power grant. Furnaces denied power by the coordinated control system continued to cool as they waited for a power grant. The furnaces then cycled on when power was granted, but their cycling period was offset from the furnace(s) with which they were previously synchronized. Hence, no special accommodations needed to be made in the algorithms to keep the furnaces from cycling.

CHAPTER IV

RESULTS

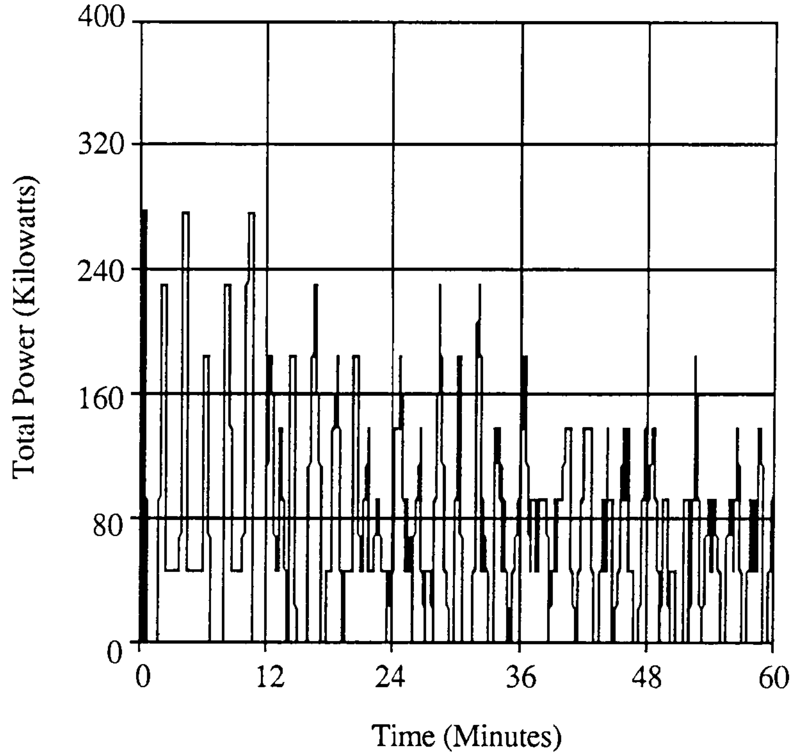
To assess the potential savings in the cost of electrical power for this example, the ACSL simulation program was modified to include the algorithms developed in the previous chapter. This chapter discusses the results of simulation of these algorithms. The actual programs used for this simulation are included in the appendix.

To provide a comparison between the algorithms, the equations for calculating the cost of electrical power developed in Chapter III were included in the simulation programs, causing the simulation program to keep a running total power cost which could be interrogated at the end of the run. With these data available, a direct comparison of algorithm effectiveness could be made.

Independent Control

Baseline data for this comparison were obtained by running the simulation with the independent controllers. All simulations had the same random weight disturbance applied to the furnaces for comparison purposes. The graphical outputs used for this comparison were the total power and individual crucible temperatures as a function of time.

Figure 4-1 shows the total power consumed with the six furnace controllers operating independently. Step changes can be seen in the plot as individual furnaces cycle on and off. A close inspection reveals that there are peaks of six different levels of total power, from 46 kW to 276 kW. Dividing the total power by 46 kW indicates the



Meter Cost: $ 3430.90

Demand Cost: $ 2147.07

Effective Monthly Cost: $ 5586.77

Figure 4-1. Total Power Consumption with Independent Control number of furnaces on at any given time. Note that there are some intermediate steps on the plot which were caused by bit-mapping of a limited resolution graphics image.

Also shown in Figure 4-1 is the cost of electric power as calculated by the simulation. As indicated in the previous chapter, the meter usage cost assumes that consumption is essentially similar for the remainder of the month. It is interesting to note that the calculated monthly cost is within the range of actual monthly billings obtained from the utility company. With independent furnace control, the demand cost accounts for nearly 40% of the total electric bill.

The crucible temperature response plots of six furnaces operating independently are shown in Figure 4-2. As mentioned previously, discontinuities can be seen in the response due to random disturbances in weight. These response plots are important in evaluating the effectiveness of the coordinated control system. The temperature response of the coordinated control system should not differ significantly from that of the independent furnace controllers.

It should be pointed out that the temperature of the aluminum does not fluctuate as much as that of the crucible due to the thermal capacitance of the large mass of aluminum. For this reason, some amount of deviation from the response of the independent furnaces can be tolerated. The maximum amount of deviation allowed depends on operational limitations of the application, and is best set by the operator of the furnaces. At any rate, the crucible temperature should cycle in the same general range as the coordinated system. A cooling trend in crucible temperature is not acceptable.

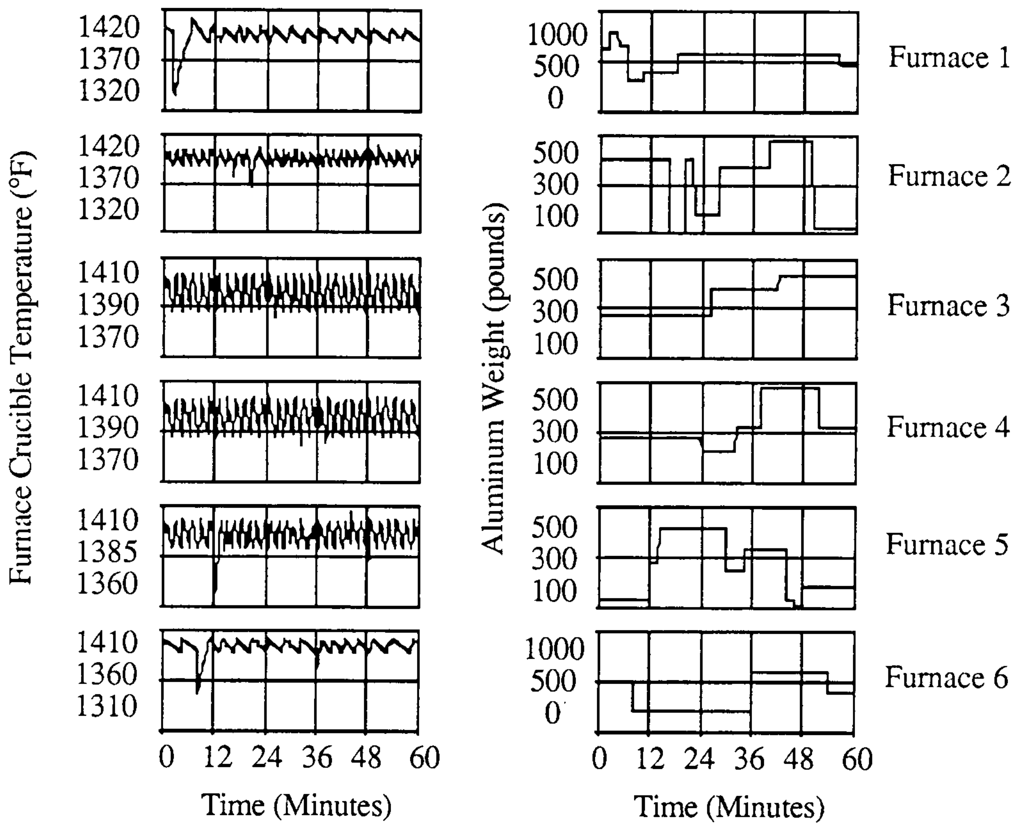


Figure 4-2. Crucible Temperatures with Independent Control

Fixed Maximum Algorithm

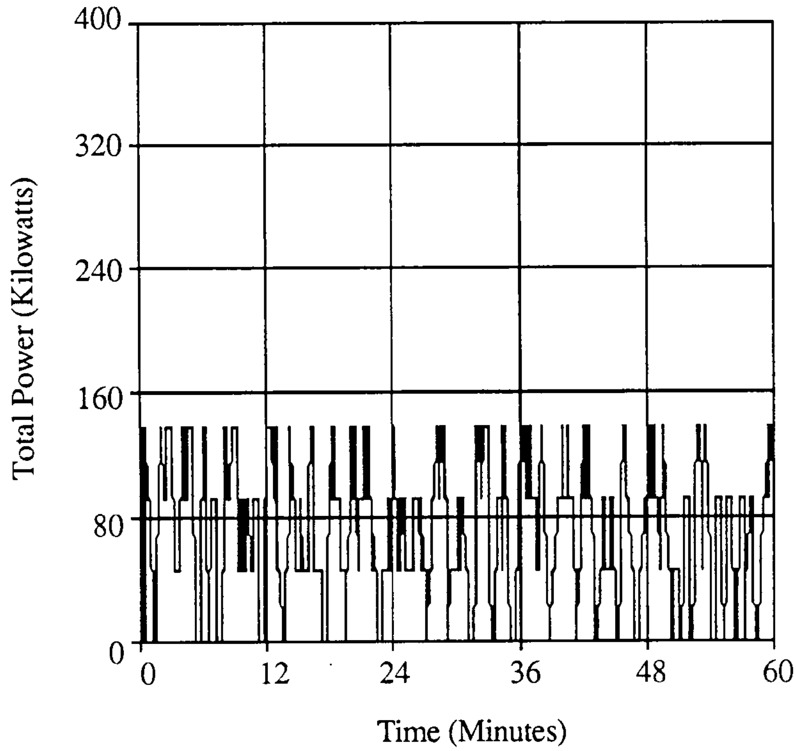
For the purpose of this comparison, the Fixed Maximum Algorithm was set to allow a maximum of three furnaces to come on at a time. Examples of simulation runs for setting this maximum from one to six are shown in Appendix B. The response plots from these simulations are

in ... in this chapter to allow easy comparison.
... with a three-furnace maximum is shown in
..., it can be seen that the peak power curve
... independent thermostats because the power
... limit is now distributed below that limit.
... ith Figure 4-1 shows graphically the
... ntrol system. The coordinated system
... vailable below the three-furnace limit to

... number of furnaces on limited to three,
... iderably lessened. Although the
... at of the independent controllers,
... 60%. It should be pointed out that
... han they appear because any
... e multiplied by the number of hours
i

Wysor, R. Wesley
Coordinated control of relay feedback control systems for electric
Thesis 88. W986
Tsypkin, Ia. Z.
Relay Control Systems
Tj 218.5.T7513

The crucible temperature response for this algorithm is shown in Figure 4-4. It can be seen that this response is very similar to that of the independent controllers. A careful comparison of these response curves reveals that there are places where the temperature dips below the normal cycling point. This is because the fourth furnace is



Meter Cost: $ 3257.62

Demand Cost: $ 836.04

Effective Monthly Cost: $ 4102.45

Figure 4-3. Total Power Consumption with a Three-Furnace Maximum

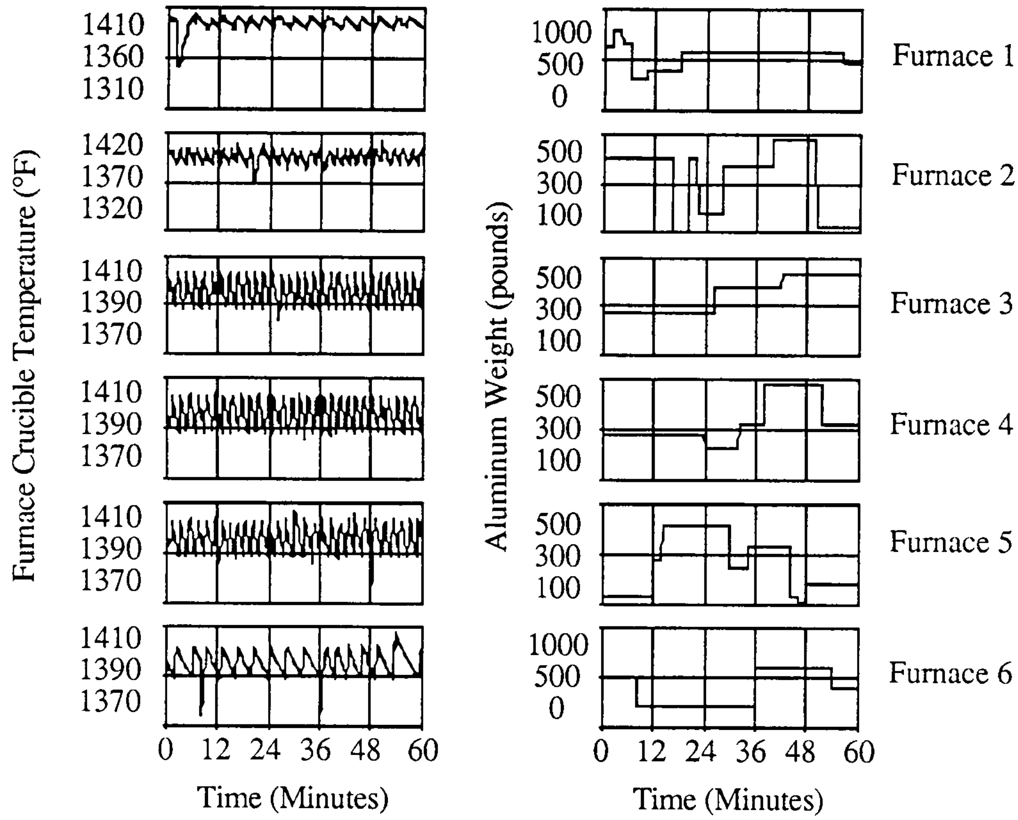


Figure 4-4. Crucible Temperatures with a Three-Furnace Maximum

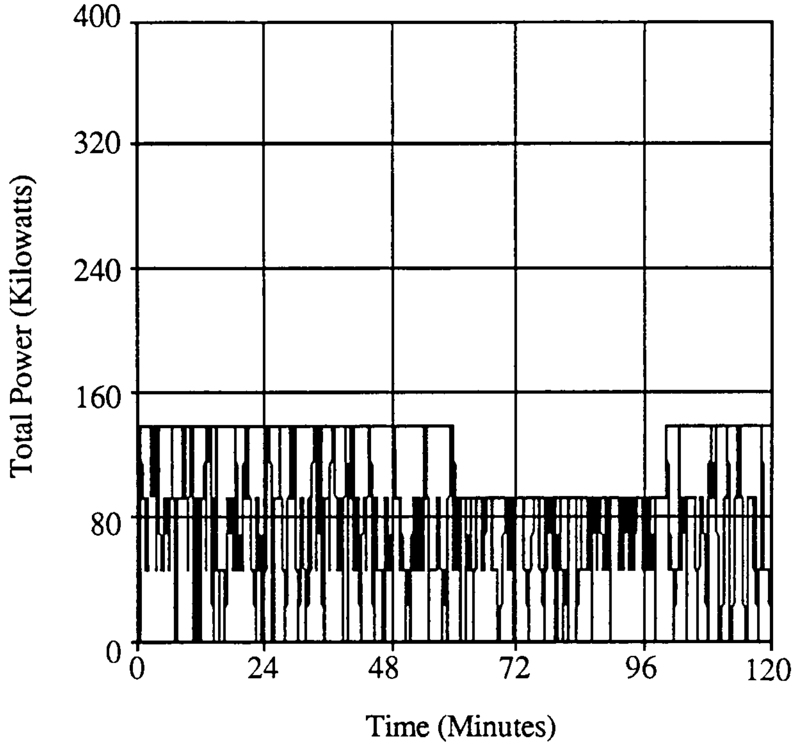
prevented from cycling on when three furnaces are already on. These dips are different from random weight changes in that they are not sharp discontinuities.

Floating Maximum Algorithm

In the simulation described here, the Floating Maximum Algorithm was initialized with a three-furnace maximum. Then, every 10 minutes the average power available was recalculated. If it was less than the power required for one furnace, the maximum number of furnaces was incremented. If it was greater than the power required for two furnaces, the maximum number of furnaces was decremented by one. Otherwise, the maximum number of furnaces was not changed.

Figure 4-5 shows the total power response for this simulation. The curve for the maximum power allowed is plotted with the same scale as for the fixed maximum algorithm. The graph shows that this maximum defines the envelope in which the furnaces are allowed to cycle. This simulation was run for twice as long as the others to show the response of the floating maximum over a longer period. In this example, the maximum number changes twice in a 2-hour period. It can be seen that simulation runs of longer duration become difficult to evaluate because of reduced graphical resolution at this scale.

The cost of electrical power is similar to that of the Fixed Maximum Algorithm with a three-furnace maximum. This is to be expected, since the maximum number of furnaces never exceeds three furnaces over the simulation period. Cost might be reduced by tuning and adjusting the algorithm on the actual system. The real advantage



Meter Cost:	$ 3127.66
Demand Cost:	$ 836.04
Effective Monthly Cost:	$ 3972.50

Figure 4-5. Total Power Consumption with the Floating Maximum Algorithm

of this algorithm, however, is that it keeps trying to reduce the maximum number of furnaces on at one time without any operator action. This means that during periods of reduced activity, power consumption is automatically reduced.

The crucible temperature response for this algorithm is shown in Figure 4-6. Although this response was available for the entire simulation, it is shown only for the first hour because of poor resolution in the plot for the entire run. The second half of the run was essentially similar to the first half except that there were no large discontinuities like the ones in the initial part of Furnaces 1 and 6. It is believed that these have to do with the nature of the weight changes interacting with the initialization of the model, since they occurred only during the early part of the simulations.

Buddy System Algorithm

The operation of the Buddy System Algorithm is similar to that of a fixed maximum of three furnaces except it is more limiting. For this reason, it was expected to produce a similar response. Peak power consumption for this algorithm, which pairs the furnaces in groups of two, is shown in Figure 4-7.

Inspection of the peak power curve for this algorithm reveals three distinct levels of power as the furnace pairs share power. In comparing this response with that of the Fixed Maximum Algorithm, it appears that there is more time during which only two furnaces are on. This provides graphical evidence that the Buddy System Algorithm is more limiting.

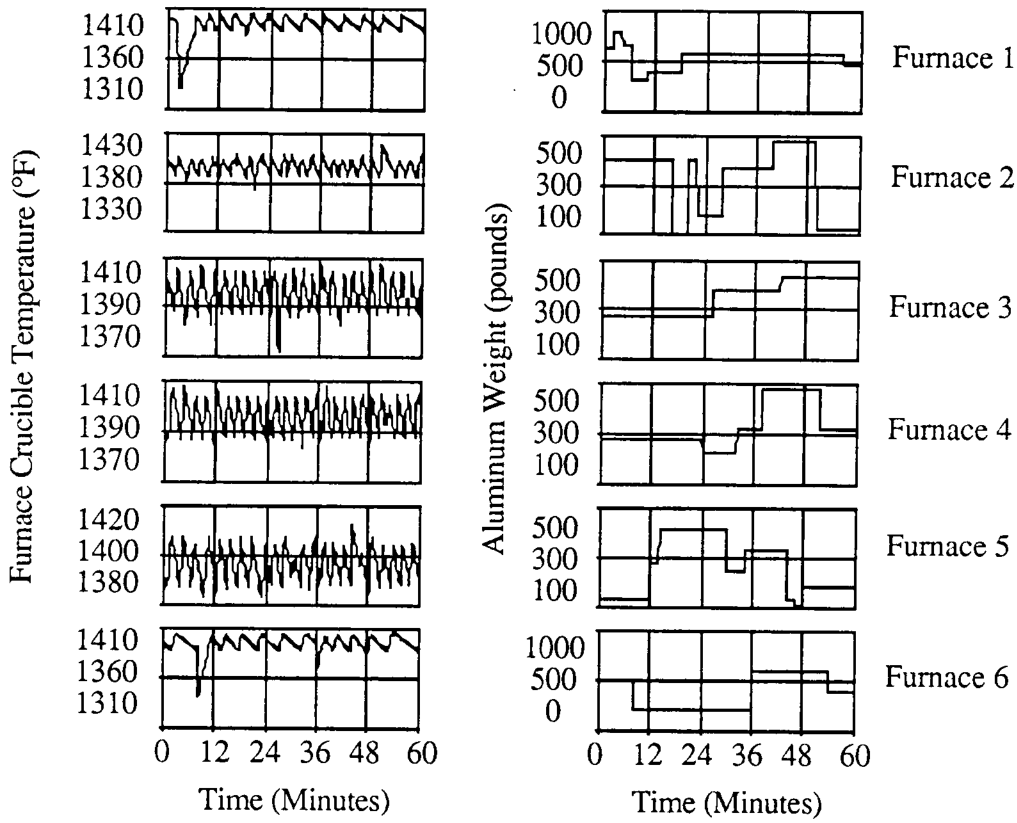
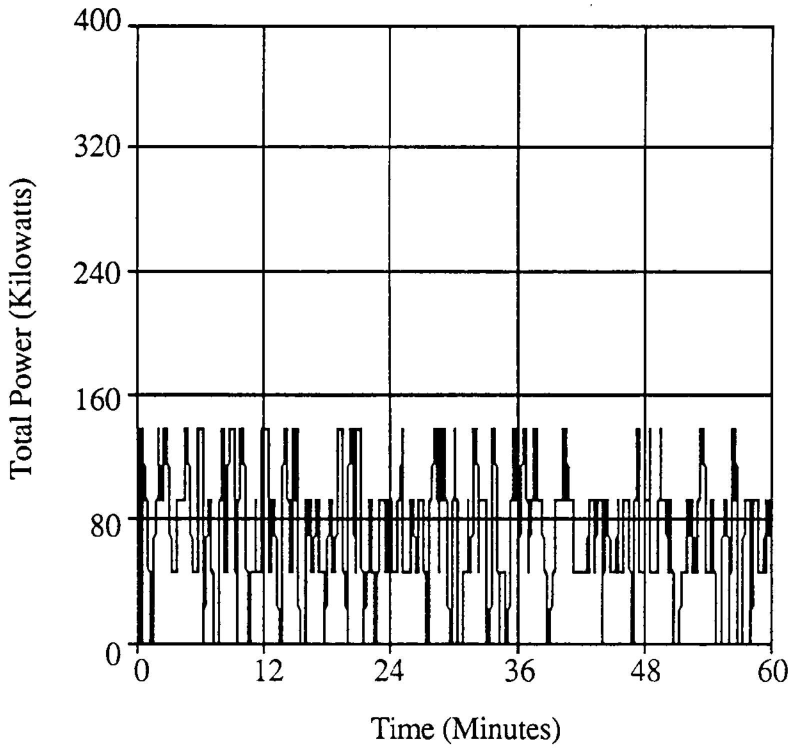


Figure 4-6. Crucible Temperatures with the Floating Maximum Algorithm



Meter Cost: $ 3243.18

Demand Cost: $ 836.04

Effective Monthly Cost: $ 4088.01

Figure 4-7. Total Power Consumption with the Buddy System

In Figure 4-8, the crucible temperature responses for this algorithm are shown. Because of the limiting nature of this algorithm, the crucible temperatures have a tendency to fall below the normal cycling range. Over several cycles, however, the control system is able to maintain the temperature within the required range. As stated previously, slight fluctuations outside the normal cycling range have minimal effect on the temperature of the aluminum.

Cost Comparison

The costs computed during the algorithm simulations are shown in Table 4-1. At first glance, it appears that the Floating Maximum Algorithm is the most attractive. It provides a savings of nearly 30% over independent control for continuous operation. There are some other factors, however, which must be considered in making the comparison.

One factor, as mentioned previously, is that the meter cost has been normalized for a full month. This means that small differences from the 1-hour simulation are multiplied by 720. When the coordinated control algorithms are compared on that basis, the cost differences are not significant.

Another consideration is the current mode of operation at the aluminum foundry. The furnaces are generally turned off for approximately 36 hours each weekend. This amounts to approximately 6.5 days over a 1-month period. When the meter cost from the simulation is adjusted for this mode of operation, the effective monthly cost for independent control is reduced to $4842.78.

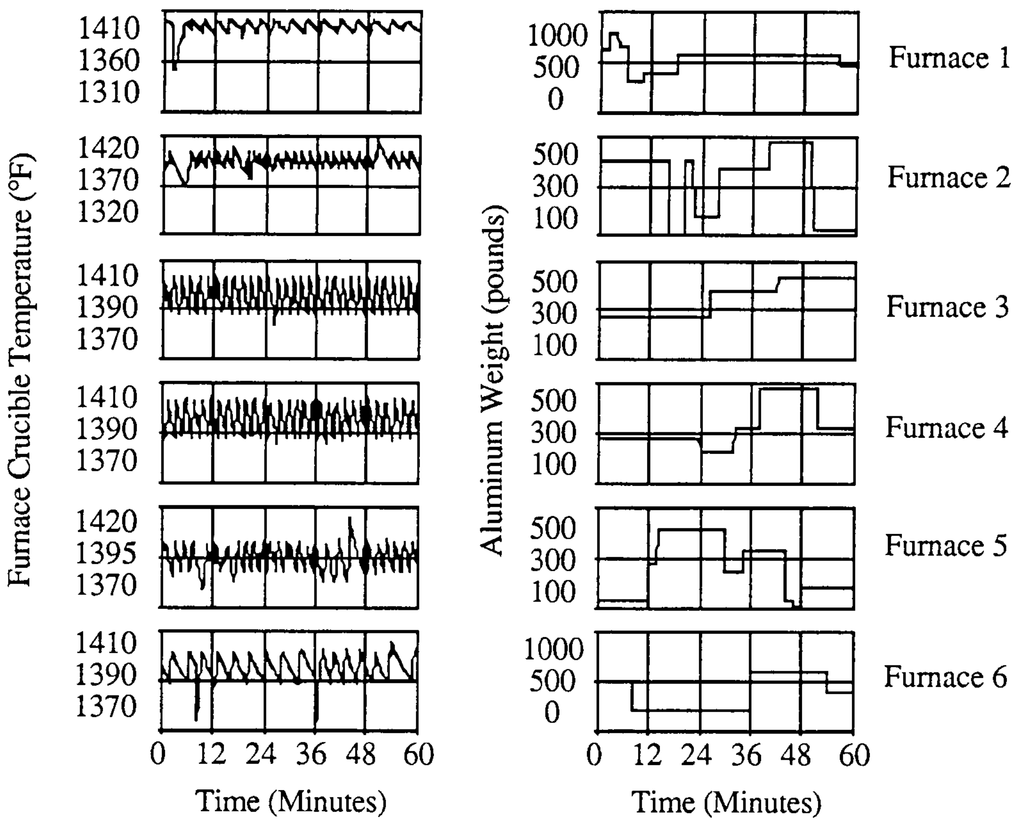


Figure 4-8. Crucible Temperatures with the Buddy System

Table 4-1. Algorithm cost comparison.

Algorithm	Meter Cost	Demand Cost	Effective Monthly Cost
Independent control	$3430.90	$2147.07	$5586.77
Three-furnace maximum	3257.62	836.04	4102.45
Floating Maximum	3127.66	836.04	3972.50
Buddy System	3243.18	836.04	4088.01

This reduces the savings of the coordinated control system to between 15 and 18%.

Startup Simulation

As stated previously, the coordinated control system algorithms developed in this study were not designed for starting the furnaces from ambient temperature. Since the capability was available, however, a startup simulation was run for each of the algorithms and the results are shown in Appendix B. Because the model used is limited to temperatures above the melting point of aluminum, actual furnace response time may differ from these simulations. The shape and behavior of the simulation responses, however, should be similar to that of the actual furnaces.

The simulation of the independent control systems is shown first for comparison purposes. With independent control, the crucible temperatures stabilize in just under 2 hours. The heating elements for all the furnaces are on continuously for most of this time.

With the maximum number of furnaces fixed at three, the crucible temperatures require over 4 hours to stabilize. An undesirable effect of this algorithm is that the first furnaces to reach operating temperature are cooled a significant amount when the power grant is taken and held by the remaining furnaces. After reaching operating temperature, however, the algorithm operation is acceptable.

The Floating Maximum Algorithm requires just over 2 hours for the crucible temperatures to stabilize. This algorithm starts with a maximum of three furnaces and increases to six after 30 minutes. Although this provides fast startup, it drives the demand cost to maximum for the month. After the temperatures stabilize, however, the algorithm acts to reduce the number of furnaces to limit the peak demand.

The Buddy System Algorithm has a response similar to that of the Fixed Maximum. It requires over 4 hours for the crucible temperatures to stabilize. Although it limits the demand to that of three furnaces, it also allows cooling of the first furnaces to reach operating temperature.

CHAPTER V

IMPLEMENTATION

In previous chapters, several coordinated control algorithms were developed and tested with a simulation model. This chapter outlines the hardware implementation for these algorithms. First, a digital control system is described which could be used to implement all three algorithms. Then a simple hardware implementation of the Buddy System Algorithm is described. Although it is beyond the scope of this study to produce a detailed design, this discussion gives an idea of the resources required for a coordinated control system.

Digital Control System Implementation

One possible implementation of the algorithms considered is a digital control system based on a personal computer. Although implementation was developed with an IBM-compatible system, it is not limited to such a system. This system was selected because of widespread availability of hardware, software, and interface boards. A block diagram of this system is shown in Figure 5-1.

The digital control system consists of a personal computer with a digital input/output interface board, a digital voltage sensing interface, a digital power controller interface, and a self-check timer with associated connecting logic. The personal computer is equipped with a minimal operating system, the BASIC computer language, and at least one floppy disk drive. It is desirable to have a fixed disk and

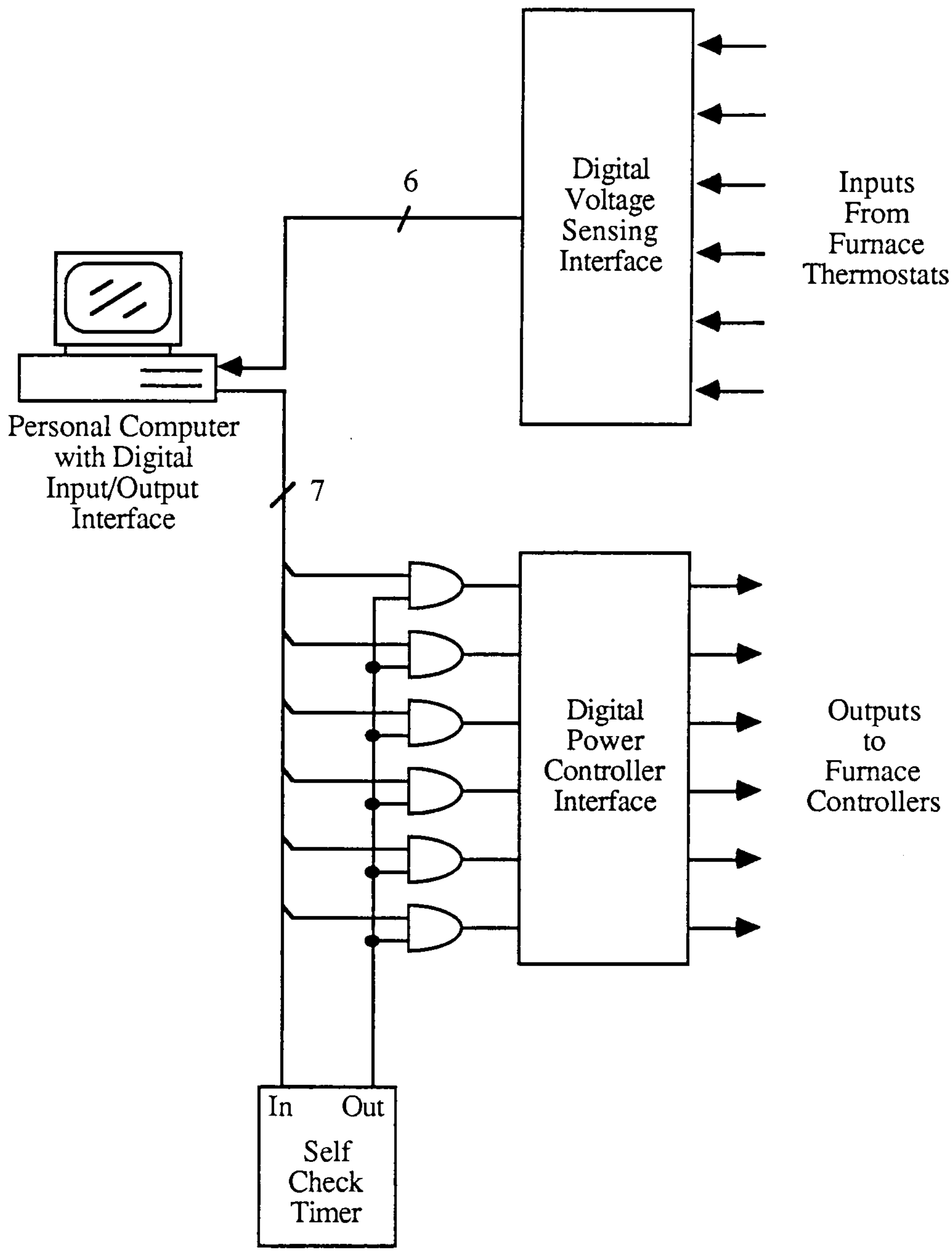


Figure 5-1. Digital Control System

printer to assist in control system development, but these are not required for a minimal system.

The digital input/output interface, the voltage sensing interface, and the power controller interface are available commercially. The use of commercial off-the-shelf devices is preferred for this system for several reasons. One is that the hardware has already been developed, tested, and field proven. This speeds system development time and improves reliability. In addition, replacement parts are readily available and do not have to be made to order.

The digital input/output interface board has one 8-bit input channel and one 8-bit output channel. As stated earlier, the sampling rate must be at least 6 samples/second for each furnace. Since the information on the state of all six furnaces can be contained in one 8-bit input, this input only needs to be sampled at least 6 times/second. Similarly, a single 8-bit output contains the information necessary to coordinate the furnaces. Assuming the output operates at the same rate as the input, this system requires only 6 inputs and 6 outputs/second. This is a very modest rate and allows the use of an inexpensive personal computer and interface board.

The digital voltage sensing interface connects to a point in each furnace controller which goes from a low to high voltage when the thermostatic switch closes. Since the furnace controller operates at 220 VAC, the sensing device must have at least this rating. This interface combines the inputs from all six furnaces into one 6-bit word. Depending on the type of device used, this may be an 8-bit interface with 2 bits unused.

The digital power controller interface takes 6 inputs and coordinates the control of the furnaces by interrupting the power to a relay in the individual furnace controllers. These contacts normally are closed, so this system drops out on a loss of power. When a logical one is placed on the input of the power controller, a set of contacts opens to interrupt power to the individual controller relay. A logical zero causes the contacts to close to allow the local controller to cycle normally.

Logical AND gates are used on the inputs of the power controller to implement the self-check feature. One of the digital outputs from the personal computer resets the self-check timer every time the control program completes a loop. Failure of the timer to be reset allows the output from the timer to be a logical zero. This prevents the power controller from interrupting power in the individual controllers and removes coordinated control from the system.

Although a device that can function as the self-check timer may be available commercially, it is shown in more detail here to describe the desired operation. A diagram of the self-check timer logic is shown in Figure 5-2. In this simple implementation, a timer supplies a clock pulse to a counter and a D flip-flop. The frequency of the clock pulse and the bit used on the counter determine the timing interval for the self-check. This interval should be conservatively longer than the longest loop time for the control program. When the input is set to a logical one, the counter is reset. This sets all the outputs to a logical zero, which gets inverted on the bit used for timing. This signal is input to a logical AND gate which is output to a D flip-flop.

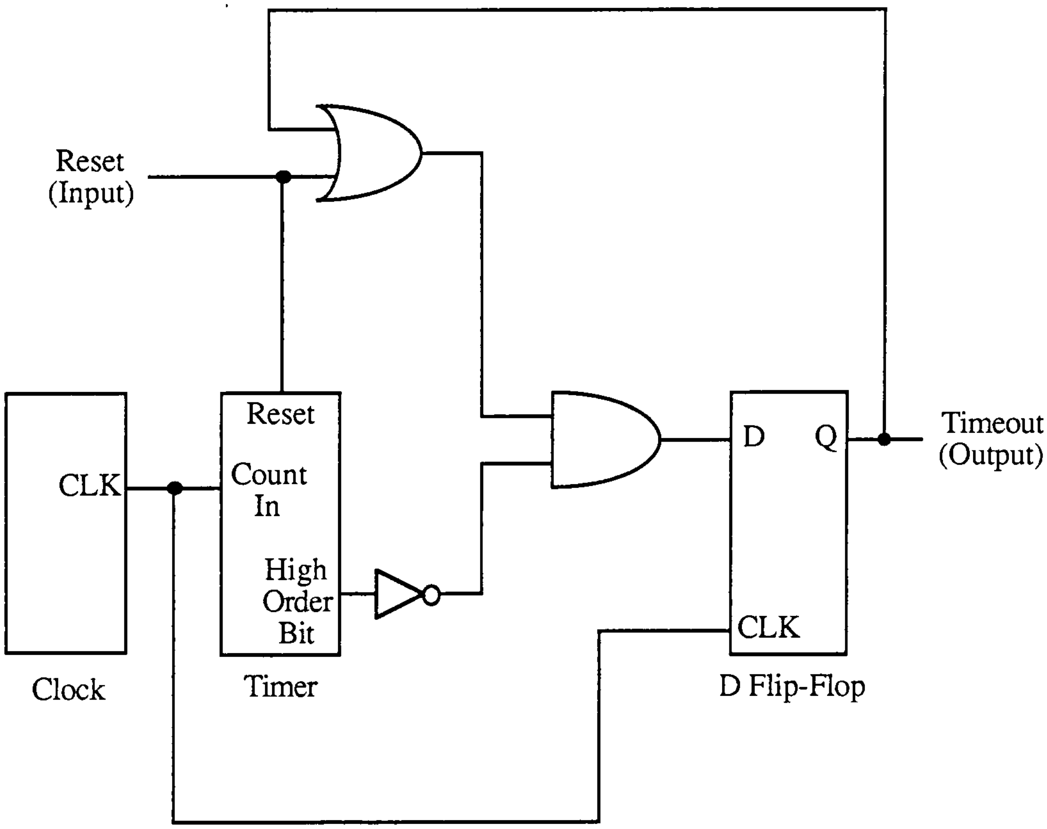


Figure 5-2. Self-Check Timer Detail

The output of the flip-flop then moves to a logical one and is sustained by feedback through a logical OR gate. After the reset input is removed, the timer continues to count. The output of the D flip-flop remains high with each input clock pulse until the counter reaches a state that sets the high order bit to a logical one. This causes a D flip-flop transition to a logical zero, which is sustained by feedback. During normal operation, however, the timer will be reset before the counter reaches this state. This maintains a logical one on the output as long as the reset pulse is received at the required intervals.

Although this device is shown as a collection of logical devices, in practice it might be implemented with a programmable array logic (PAL) device. With the exception of the timer, the entire self-check timer can be implemented in a single PAL. In addition, the logical AND gates on the input of the power controller can be supplied by the PAL. This reduces the number of integrated circuits and makes the device more compact. It also makes the device more reliable, as digital system faults are often found to be caused by faulty mechanical connections.

Buddy System Implementation

The Buddy System Algorithm is attractive in that it is easy to implement with a minimum investment. This algorithm can be implemented with some rewiring in the existing individual controller and possibly six additional relays. A simplified diagram of this algorithm implemented in one of the furnace controllers is shown in Figure 5-3.

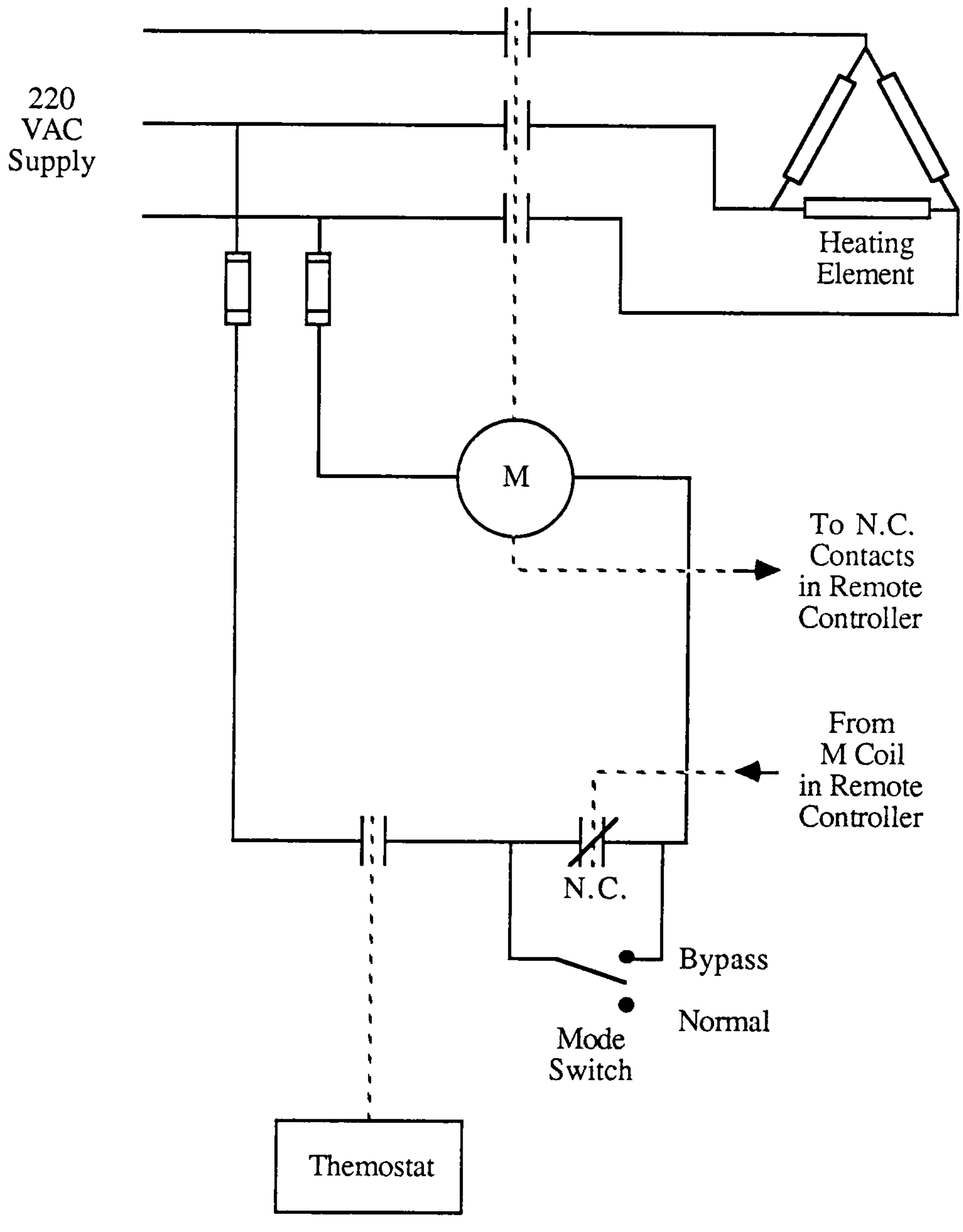


Figure 5-3. Buddy System Implementation

The relay that operates the main contacts in the local controller is connected to a set of normally closed contacts in the remote controller. When the main contacts in this controller are closed, the remote controller is disabled by opening the normally closed contacts. Similarly, the relay that operates the main contacts in the remote controller operates a set of normally closed contacts in the local controller. This disables the local controller when the main contacts are closed in the remote controller.

If the main relay in the furnace controller has a set of normally closed contacts that are not in use, they can be used to implement this algorithm. Otherwise, an additional relay needs to be connected such that it is energized when the main contacts are closed. Normally closed contacts on this relay are used to implement this algorithm in the remote controller. A bypass switch is included to return the furnace controller to normal (individual) operation.

CHAPTER VI

SUMMARY AND CONCLUSIONS

This study began with a practical problem in which a local foundry was paying more for peak electrical demand than seemed necessary. Coordinating the control of these furnaces so that not all the furnaces come on at once seemed possible, since the furnaces were generally off more than twice as long as they were on. The object of this study was to determine if a coordinated control system could be designed to fit this application.

One of the most involved parts of this study was the development of a model for the furnaces under consideration. This was accomplished by studying the construction of the furnaces and determining the materials and physical dimensions. Some assumptions were then made concerning how the furnaces operated, and a second-order model was constructed. This model included the major elements of the furnace and was presented using the electrical circuit analogy.

Using the furnace material properties, values were assigned to the furnace elements and the model tested using a computer simulation. The results of the initial simulations were studied and compared to the actual furnaces. Some adjustments were made to the furnace parameters so that operation of the model was similar to that of the actual furnaces. The simulation program was then enhanced to include six furnaces with independent thermostatic controllers. In addition, random disturbances in weight were added to simulate adding and removing aluminum from the furnaces during normal operation. The simulation program was used to develop the coordinated control algorithms.

Before the algorithms were developed, the computation of electrical utility cost was added to the simulation program. From this computation, the effective monthly electrical cost was computed from its three component parts, the fixed cost, the meter cost, and the demand cost. This cost represented the projected cost of furnace operation based on a 1- or 2-hour simulation.

Three algorithms were developed in this study: the Fixed Maximum, the Floating Maximum, and the Buddy System Algorithms. For each of these algorithms, total power and crucible temperature were plotted as a function of time. Comparison of these algorithms was based on their ability to maintain temperature and effective monthly cost.

The Buddy System Algorithm is essentially a restricted version of the three-furnace maximum. For a consistent comparison, the Fixed Maximum Algorithm was set to a three-furnace maximum and the Floating Maximum was initialized with a maximum of three furnaces. Simulations of the Fixed Maximum Algorithm for one to six furnaces are shown in Appendix A. In addition, a simulation of a startup of all six furnaces is shown in that appendix. The development of coordinated control during startup is left as an area for future study and development.

All three of the algorithms maintained temperature in a satisfactory manner. Although slight variations outside the normal cycling range were noted, the large thermal capacitance of the molten aluminum made these variations negligible. The electrical costs for all the algorithms were essentially similar, considering that small differences in meter cost are multiplied by number of hours in a month.

When the furnaces are operated continuously, the coordinated control algorithms offer a potential savings of between 25 and 30%. A more realistic comparison, however, would be to include weekends, when the furnaces are shut down. In this case, the coordinated control algorithms offer a potential savings of from 15 to 18%, excluding the savings in labor cost for starting the individual controllers on Sunday afternoons.

After simulating the algorithms, implementation was considered. All three algorithms can be implemented with a digital control system. Although this implementation has an initial expense of around $2,000, it provides the most flexibility. This digital control system implementation offers the advantage of being a development system in which different algorithms can be adjusted and tested.

The Buddy System Algorithm offers the advantage of being inexpensive and effective. For a few hundred dollars, it offers a savings of over $700/month. In other words, this algorithm would pay for itself with the first utility bill. Although not as flexible as the digital control system, implementation requires only the addition of relays and switches. It would probably be the most suitable for the operating environment in the foundry. In addition, the Buddy System provides for an inexpensive operational test of coordinated control on the actual furnaces.

In conclusion, this study has shown that coordinated control is an effective way to reduce the cost of electrical power for large cycling loads. In the foundry example, a savings of 15 to 30% is possible, depending on operating conditions. This example provides a case study for the design of a coordinated control system. The positive results of this study should provide an incentive for using coordinated control in other applications.

BIBLIOGRAPHY

BIBLIOGRAPHY

Beyer, William E., ed. *CRC Standard Mathematical Tables.* Twenty-seventh edition. Boca Raton, FL: CRC Press, Inc., 1984.

Buckley, Page S. *Techniques of Process Control.* New York: John Wiley & Sons, Inc., 1964.

Cone, Carroll. *Energy Management for Industrial Furnaces.* New York: John Wiley and Sons, 1980.

D'Azzo, John J. and Constantine H. Houpis. *Linear Control System Analysis and Design.* New York: McGraw-Hill, Inc., 1975.

DeFatta, David J., Joseph G. Lucas, and William S. Hodgkiss. *Digital Signal Processing: A System Design Approach.* New York: John Wiley & Sons, 1988.

Harrison, Howard L. and John G. Bollinger. *Introduction to Automatic Controls.* Second Edition. New York: Harper & Row, 1969.

Hatch, John E., ed. *Aluminum: Properties and Physical Metallurgy.* Metals Park, Ohio: American Society for Metals, 1984.

Hayt, William H., Jr. and Jack E. Kemmerly. *Engineering Circuit Analysis.* New York: McGraw Hill, Inc., 1978.

Kreith, Frank. *Principals of Heat Transfer.* Third Edition. New York: Harper & Row, 1973.

Millar, William J. *Electric Furnace Design, Manufacture, and Application in Germany.* New York: Mapleton House, 1947.

Otto, Charles A., ed. *Electric Furnaces.* London: George Newnes Ltd., 1958.

Paschkis, Victor and John Persson. *Industrial Electric Furnaces and Appliances.* Second Edition. New York: Interscience Publishers, Inc., 1960.

Pring, John N. *The Electric Furnace.* London: Longmans, Green and Co., 1921.

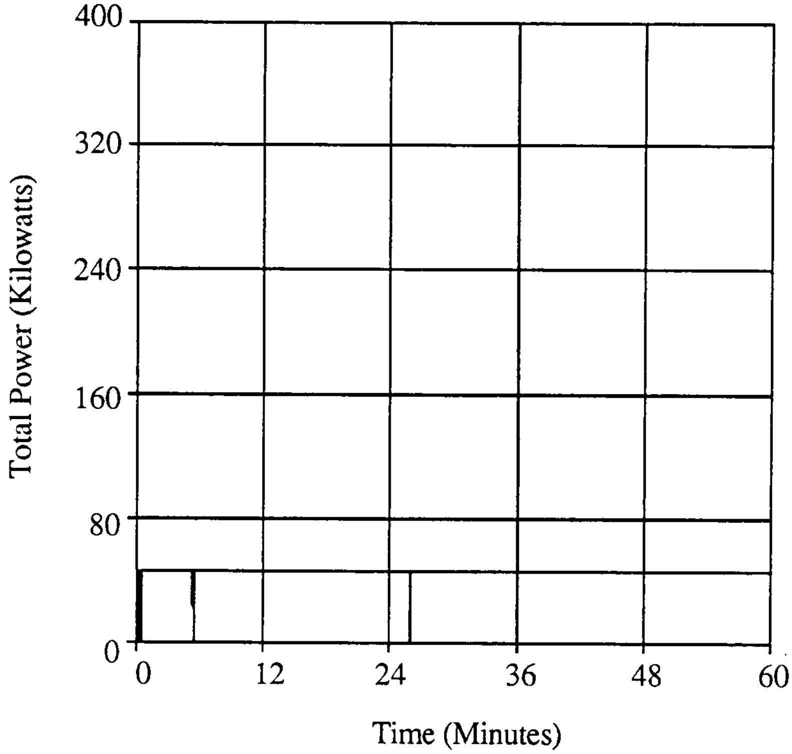
Trinks, Willibald. *Industrial Furnaces. Volume 1.* Third Edition. New York: John Wiley & Sons, Inc., 1934.

Trinks, Willibald. *Industrial Furnaces. Volume 2.* Second Edition. New York: John Wiley & Sons, Inc., 1942.

APPENDIXES

APPENDIX A

ADDITIONAL SIMULATION RESULTS



Meter Cost:	$ 2064.89
Demand Cost:	$ 37.99
Effective Monthly Cost:	$ 2035.70

Figure A-1. Total Power Consumption with a One-Furnace Maximum

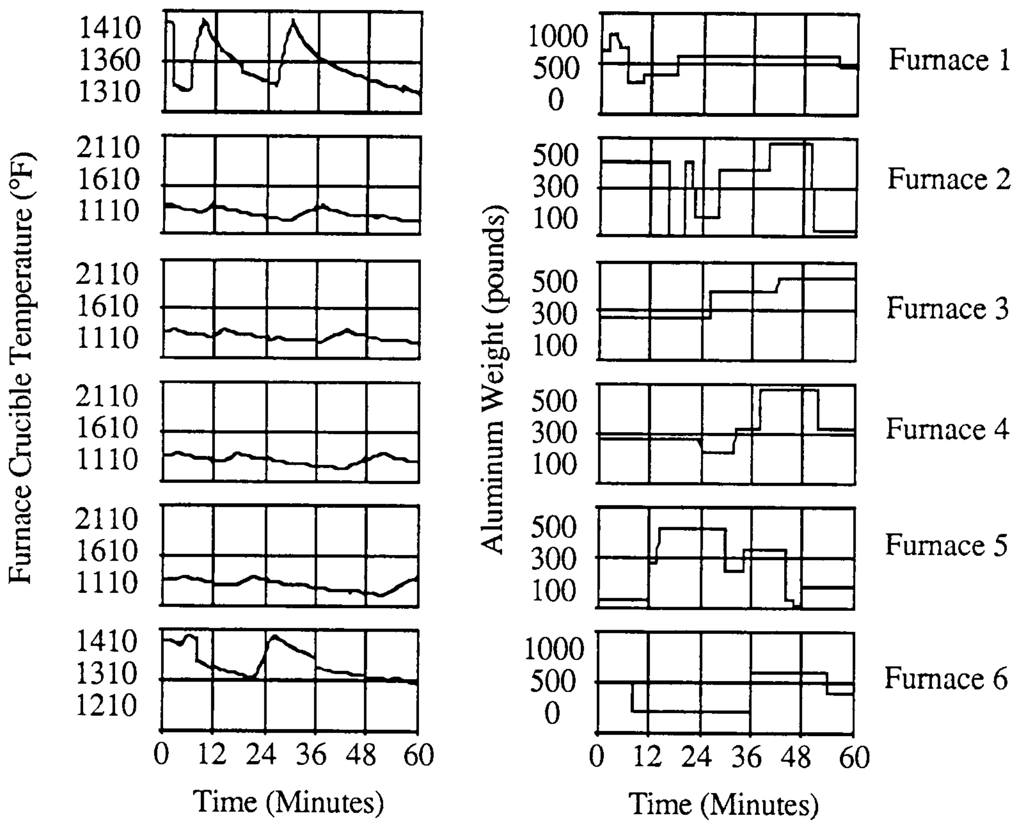
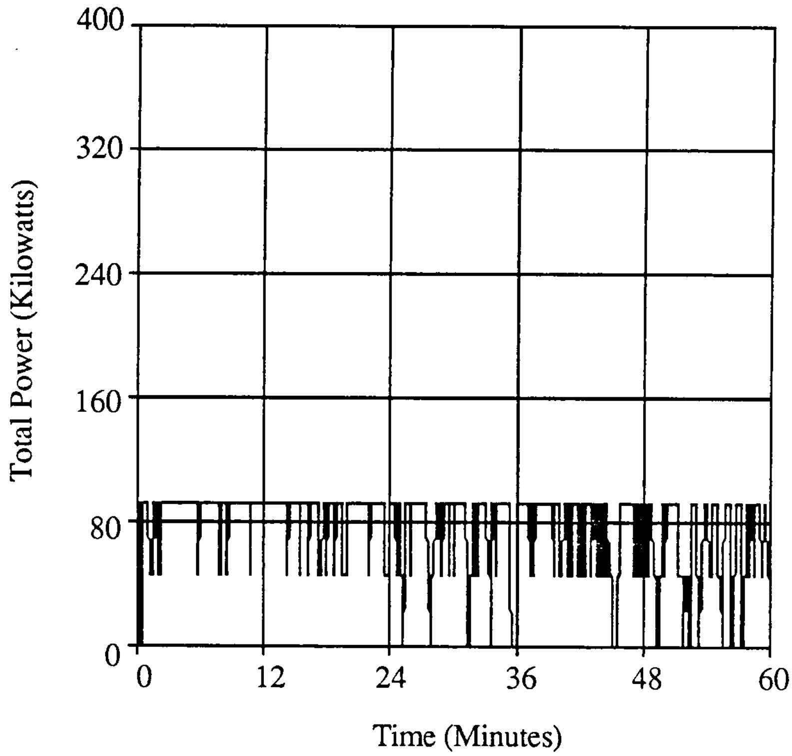


Figure A-2. Crucible Temperatures with a One-Furnace Maximum



Meter Cost:	$ 3462.66
Demand Cost:	$ 399.02
Effective Monthly Cost:	$ 3870.49

Figure A-3. Total Power Consumption with a Two-Furnace Maximum

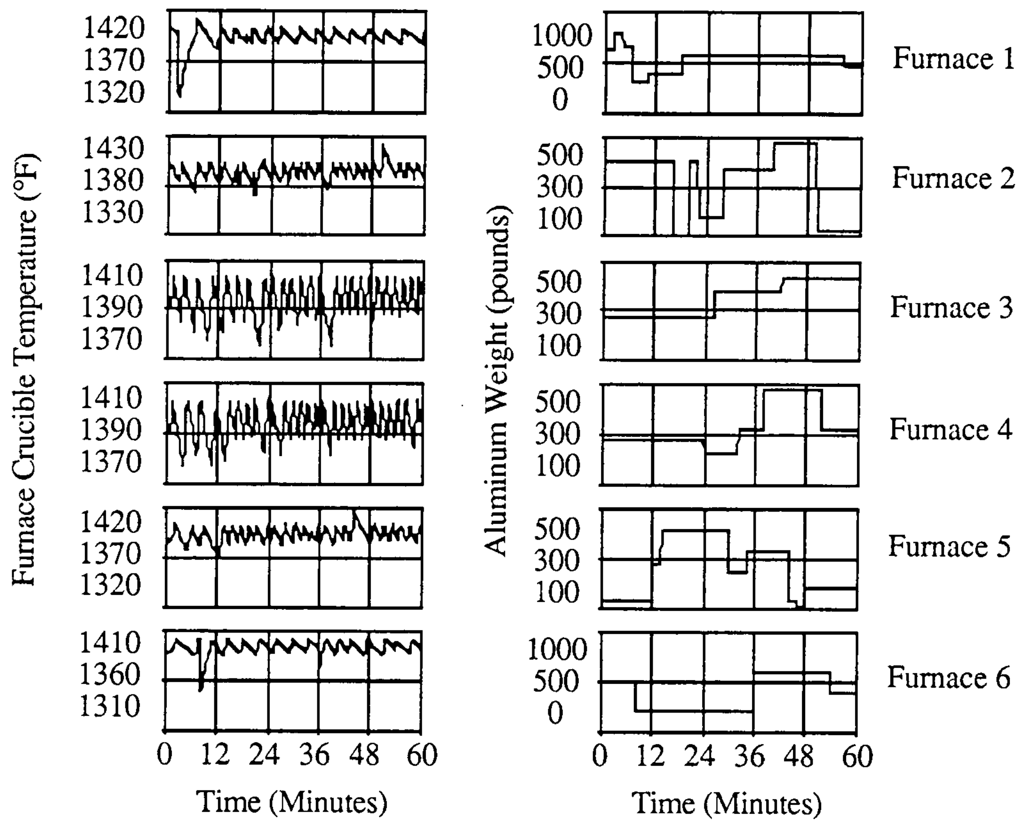
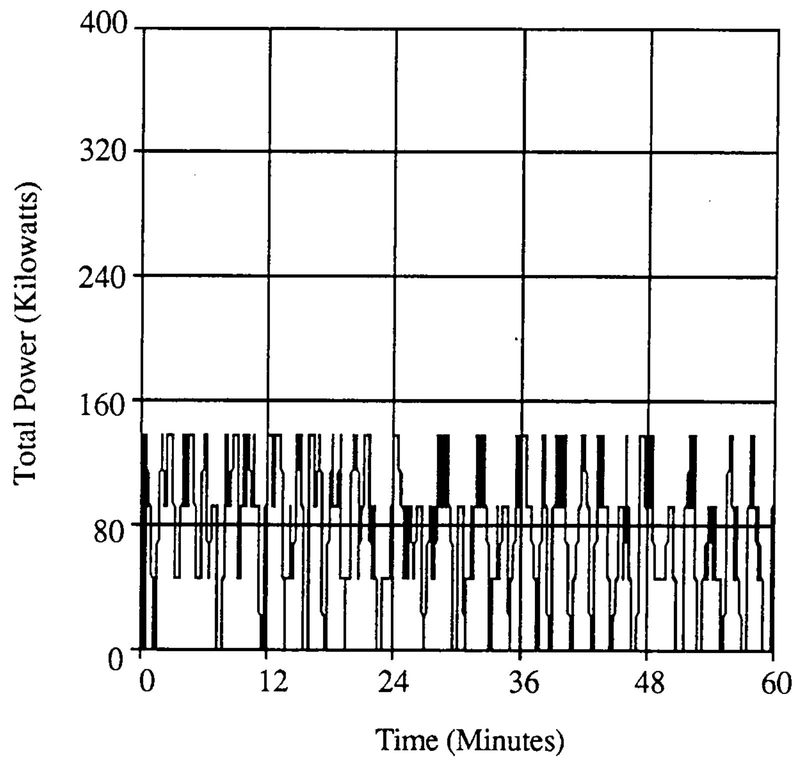


Figure A-4. Crucible Temperatures with a Two-Furnace Maximum



Meter Cost:	$ 3430.90
Demand Cost:	$ 836.04
Effective Monthly Cost:	$ 4275.73

Figure A-5. Total Power Consumption with a Three-Furnace Maximum

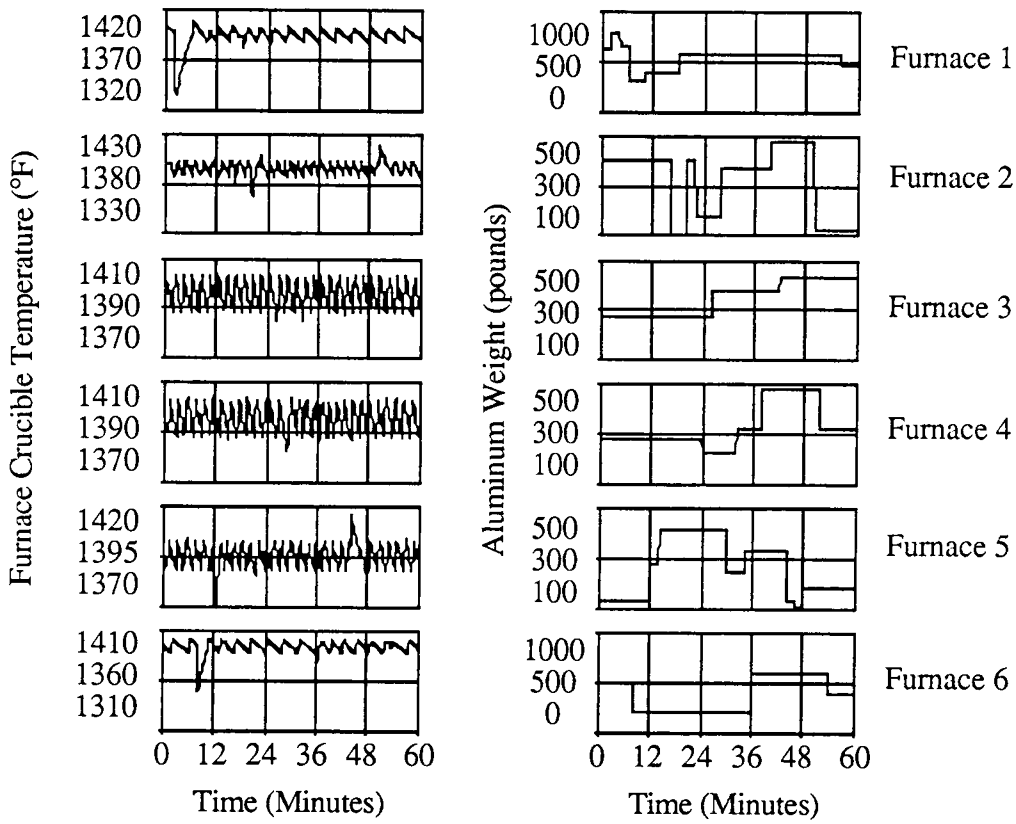
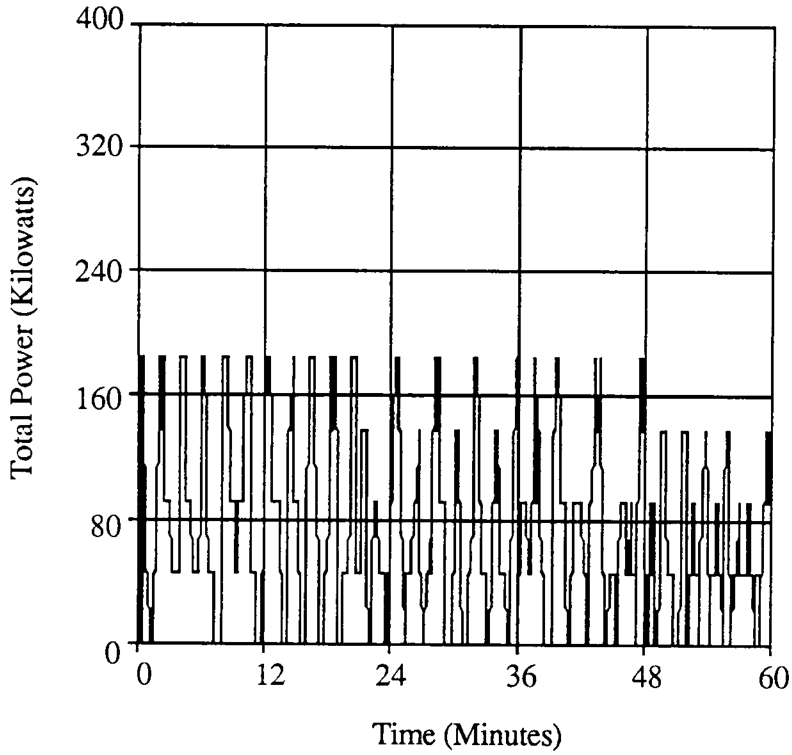


Figure A-6. Crucible Temperatures with a Three-Furnace Maximum



Meter Cost:	$ 3456.89
Demand Cost:	$ 1273.05
Effective Monthly Cost:	$ 4738.74

Figure A-7. Total Power Consumption with a Four-Furnace Maximum

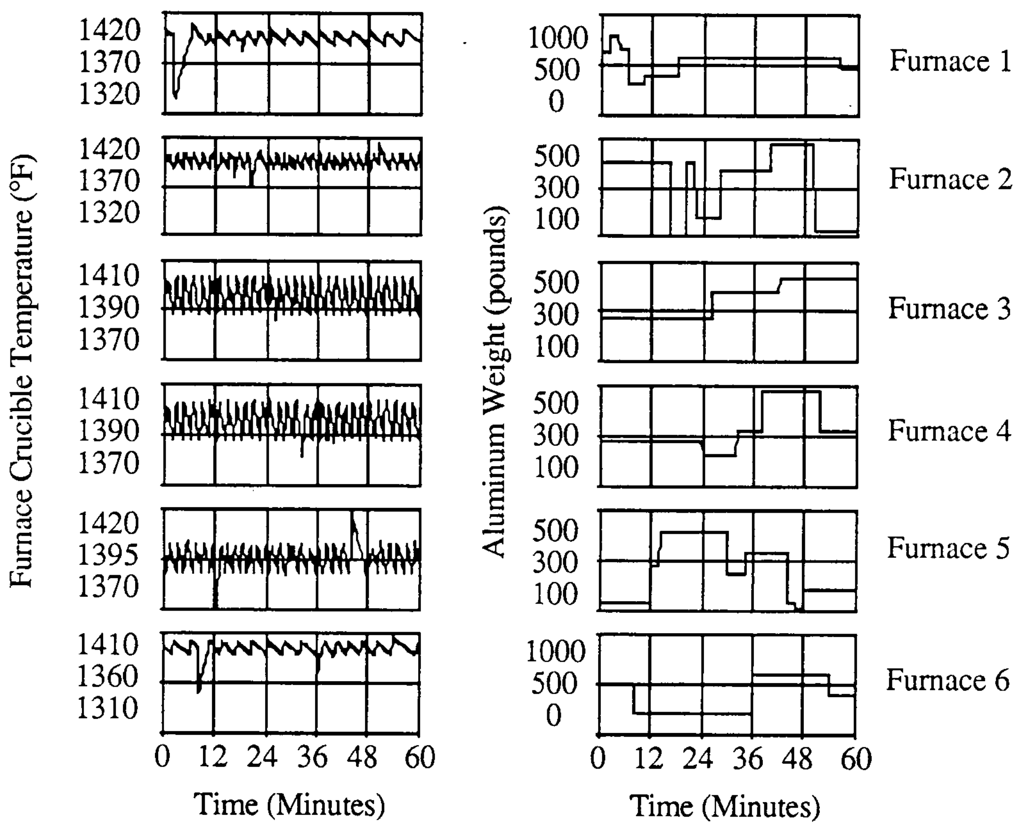
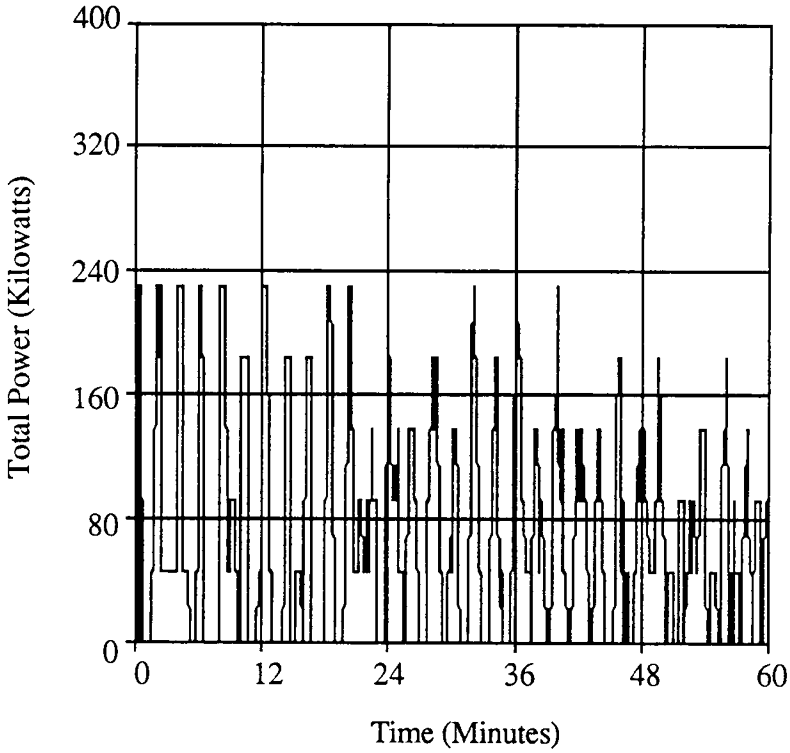


Figure A-8. Crucible Temperatures with a Four-Furnace Maximum



Meter Cost:	$ 3237.40
Demand Cost:	$ 1710.06
Effective Monthly Cost:	$ 4956.26

Figure A-9. Total Power Consumption with a Five-Furnace Maximum

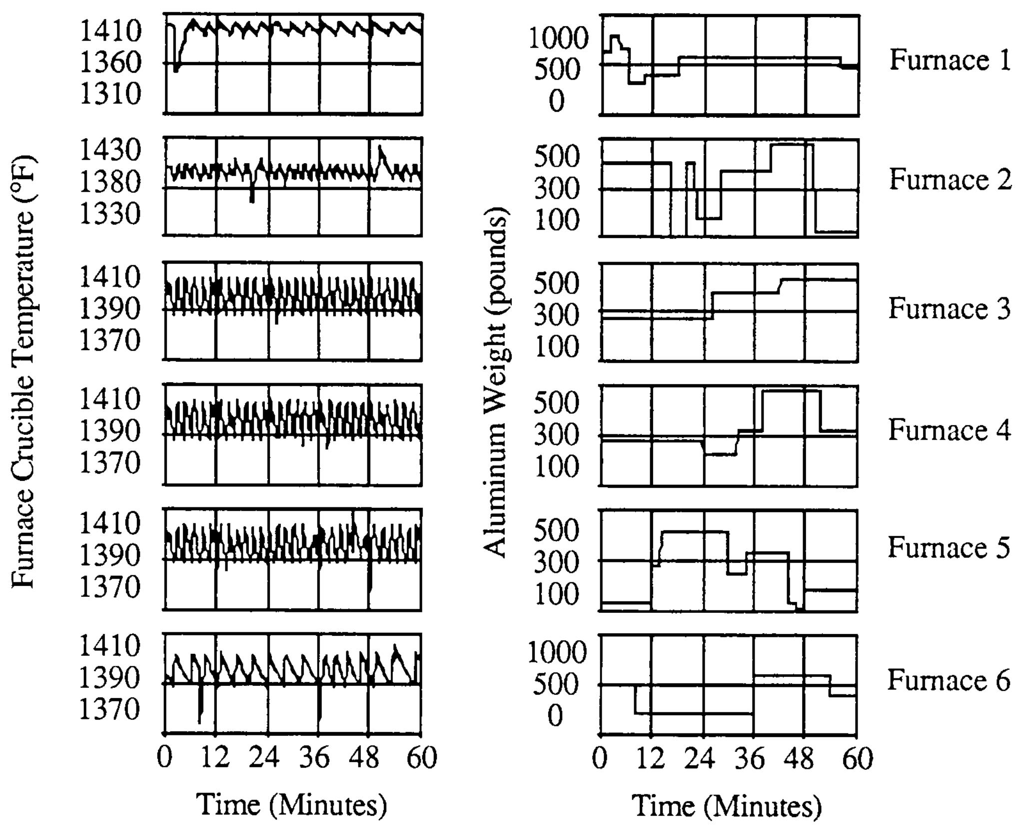
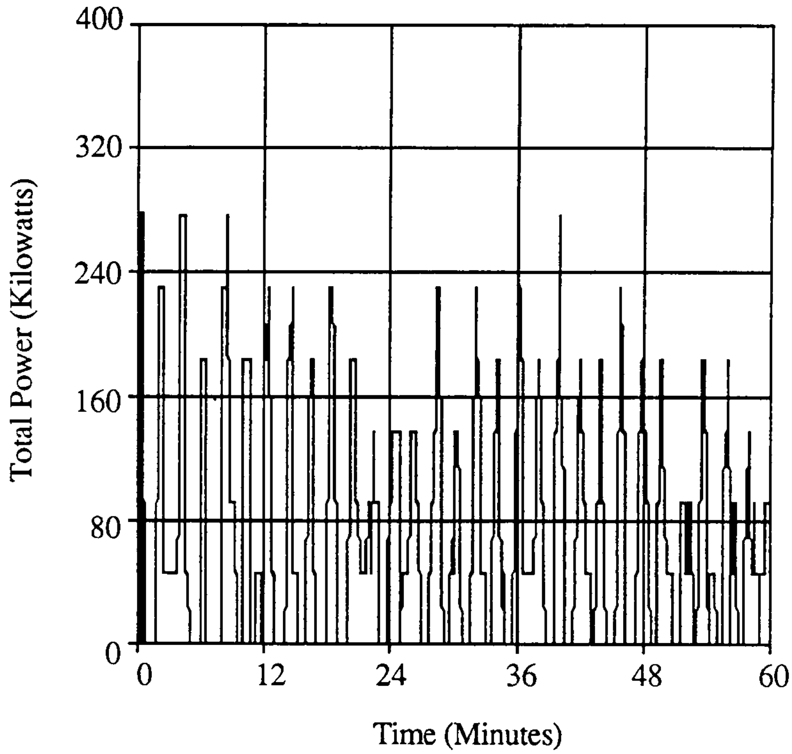


Figure A-10. Crucible Temperatures with a Five-Furnace Maximum



Meter Cost:	$ 3228.74
Demand Cost:	$ 2147.07
Effective Monthly Cost:	$ 5384.61

Figure A-11. Total Power Consumption with a Six-Furnace Maximum

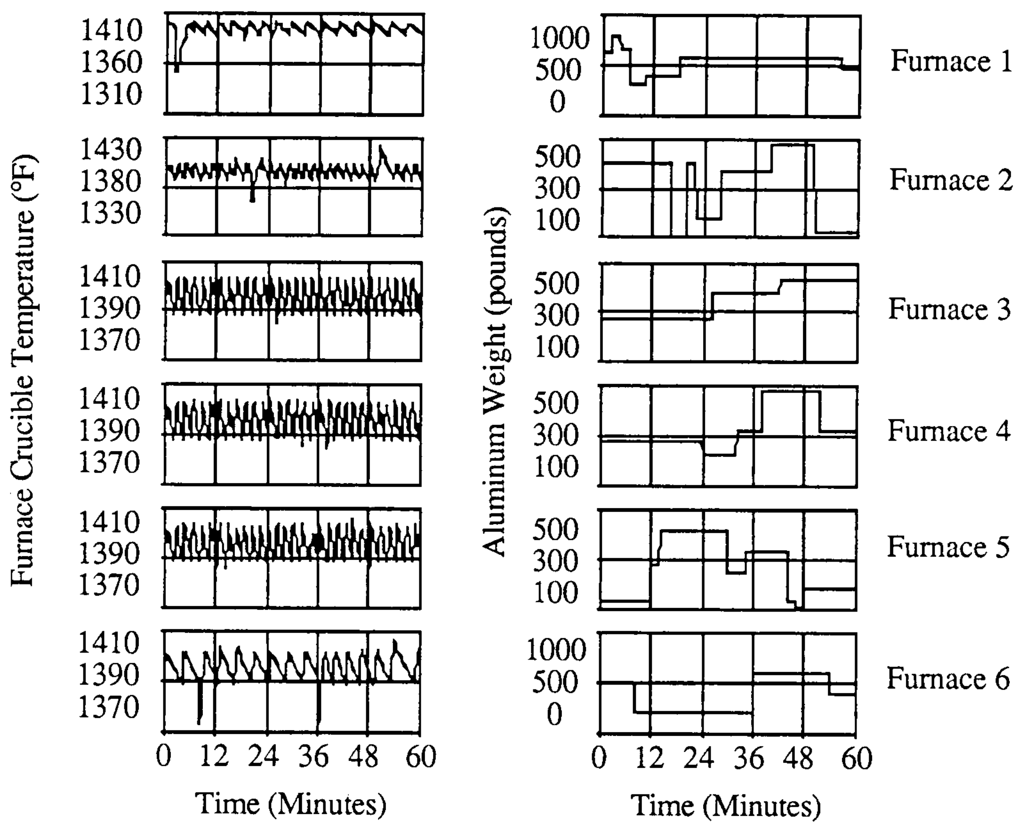
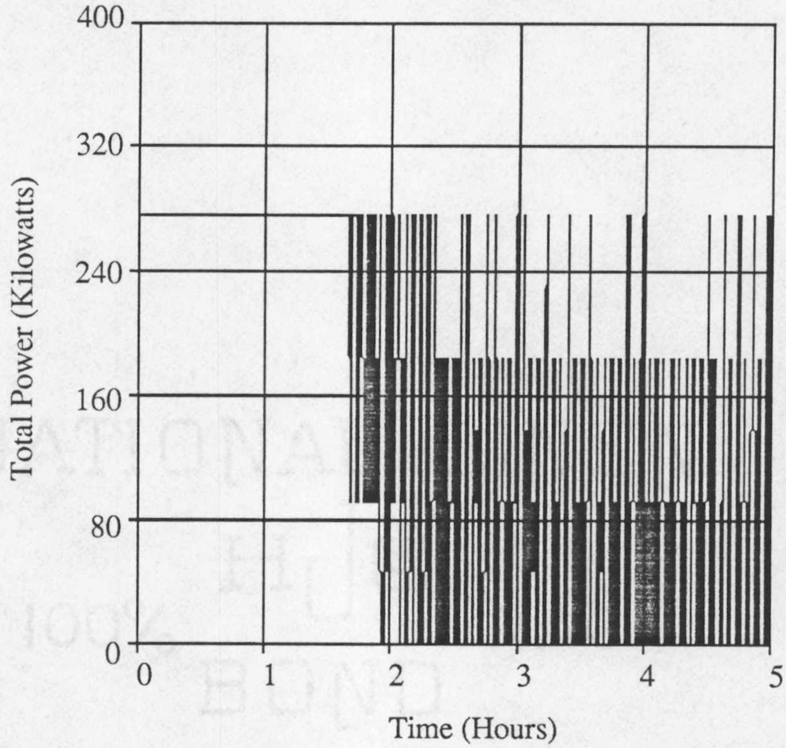


Figure A-12. Crucible Temperatures with a Six-Furnace Maximum



Meter Cost: $ 509.44

Demand Cost: $ 2147.07

Figure A-13. Total Power Consumption for Furnace Startup with Independent Control

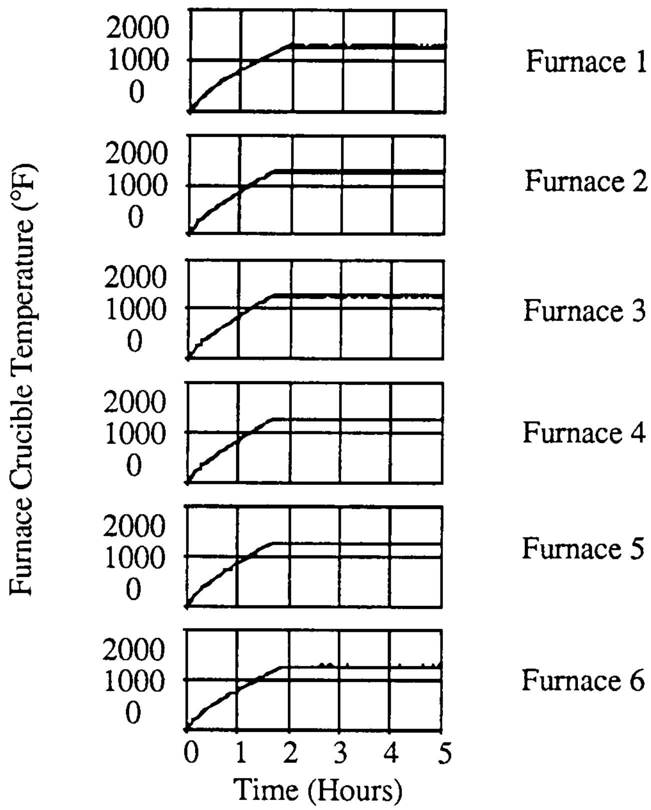
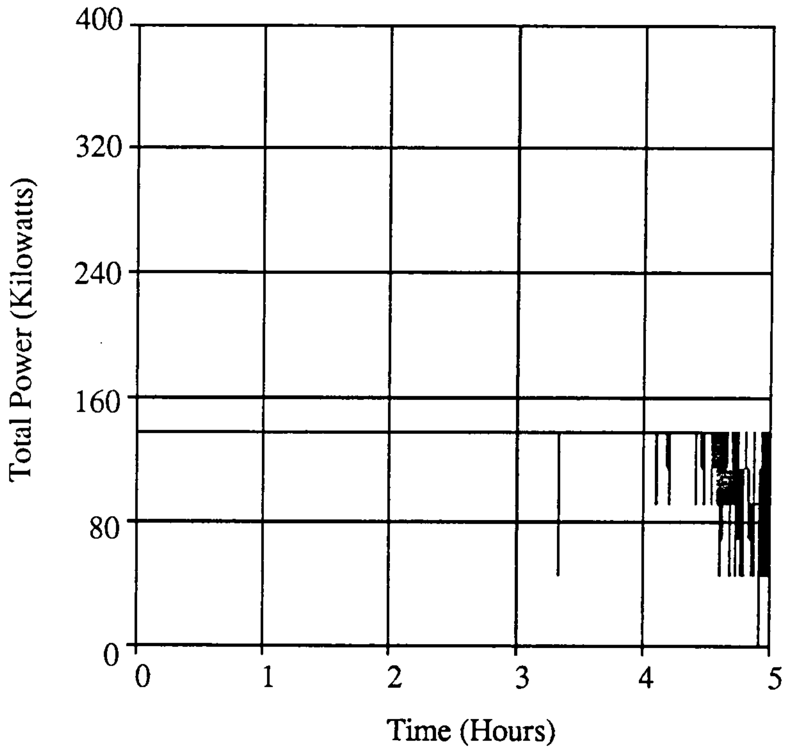


Figure A-14. Crucible Temperatures for Furnace Startup with Independent Control



Meter Cost: $ 506.94

Demand Cost: $ 836.04

Figure A-15. Total Power Consumption for Furnace Startup with a Three-Furnace Maximum

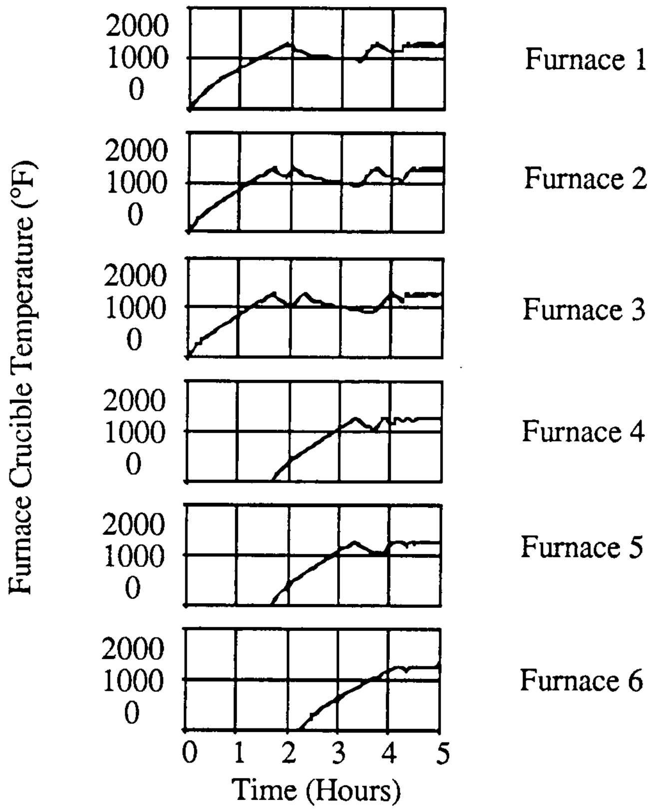
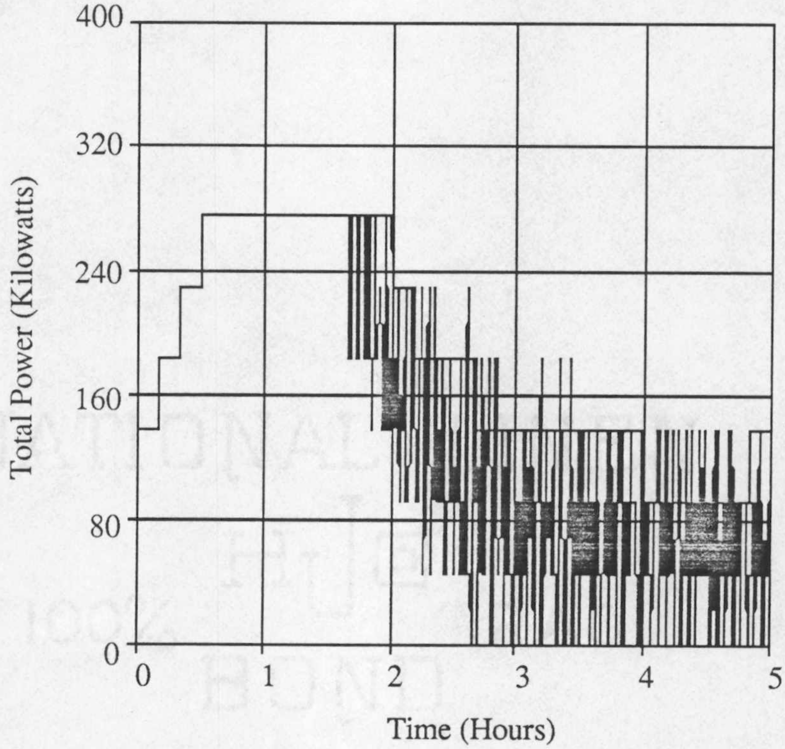


Figure A-16. Crucible Temperatures for Furnace Startup with a Three-Furnace Maximum



Meter Cost: $ 509.11

Demand Cost: $ 2147.07

Figure A-17. Total Power Consumption for Furnace Startup with the Floating Maximum Algorithm

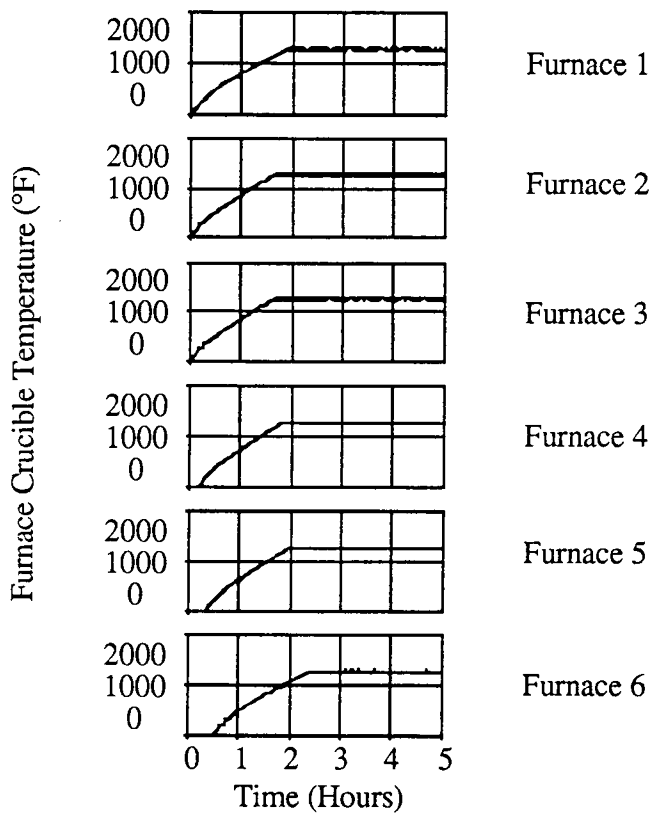
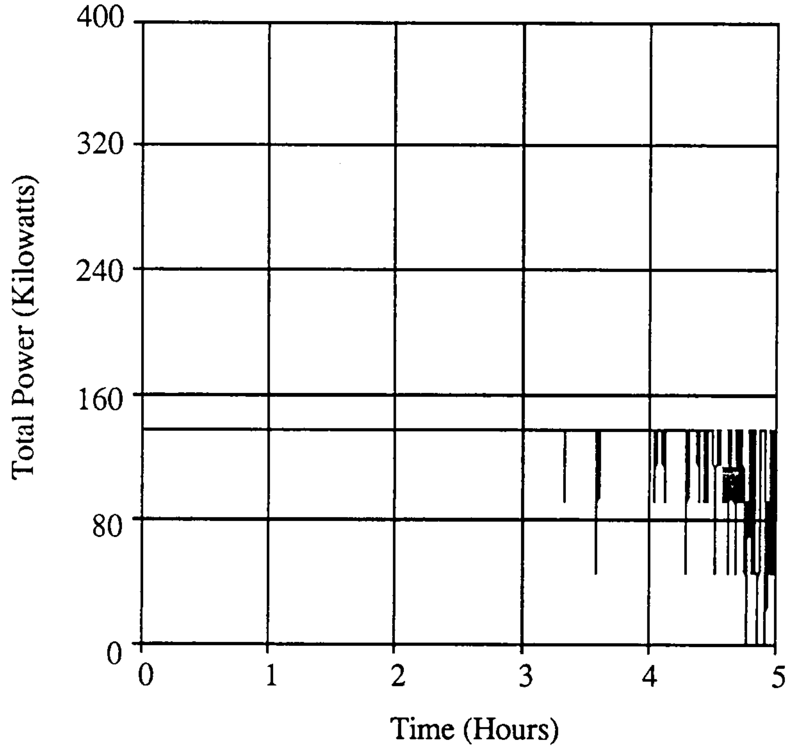


Figure A-18. Crucible Temperatures for Furnace Startup with the Floating Maximum Algorithm



Meter Cost: $ 506.84

Demand Cost: $ 836.04

Figure A-19. Total Power Consumption for Furnace Startup with the Buddy System

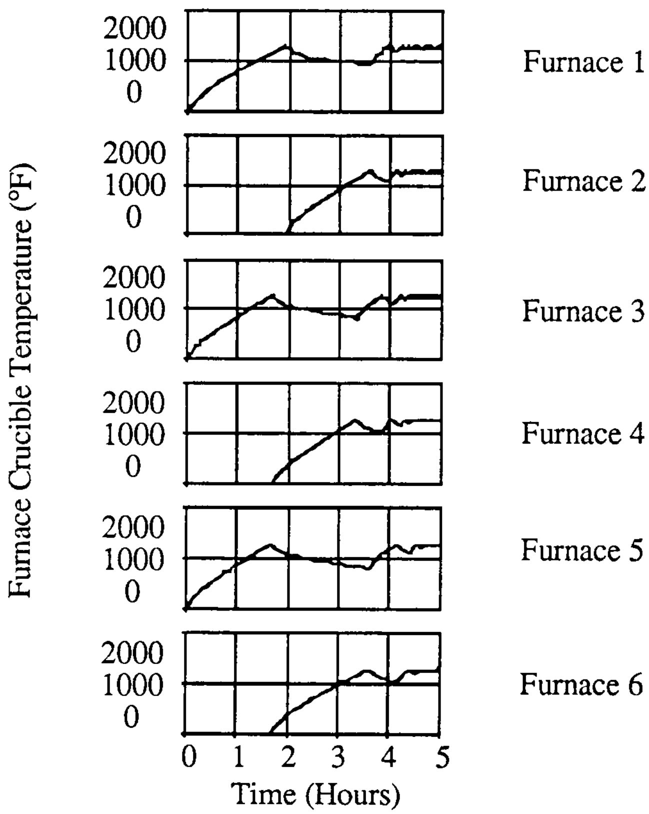


Figure A-20. Crucible Temperatures for Furnace Startup with the Buddy System

APPENDIX B

SIMULATION PROGRAMS

```
! Furnace Simualtion Using True Basic on Macintosh
!
! This program simulates the operation of the furnace
! in the normal operating range
!
! This program does a control system simulation using
! a digital simulation of an analog simulation technique.
! The integration is accomplished by keeping a running
! sum of the integrand multiplied by the delta ("d")

! Initial Conditions
PRINT
INPUT PROMPT "   Enter WEIGHT of Aluminum  ":W
LET RA=1997.0/W                          ! VARIABLE RA
LET RC=5.026                             ! WAS   5.026
LET RT=757.0                             ! WAS 757.0
LET RW=100.0                             ! WAS 603.9
LET CA=0.2727*W                          ! VARIABLE CA
LET CC=71.81                             ! WAS  71.81
LET NUM=RW/(RA*CA*(RC+RW)*CC)
LET D1=1/((RC+RW)*CC)+1/(RA*CC)+1/(RA*CA)+1/(RT*CA)
LET D2=1/(RA*CA*CC*(RC+RW))+1/(RT*CA*CC*(RC+RW))
LET D2=D2+1/(RA*RT*CA*CC)
PRINT
PRINT "   NUM = ";NUM
PRINT "   D1 = ";D1
PRINT "   D2 = ";D2

PRINT
INPUT PROMPT "   Enter stop time in seconds  ":Tstp
! tstp = simulation stop time
print
INPUT PROMPT "   Enter temperature differential in degrees  ":delta
LET HISET=1290
LET LOSET = HISET - delta
print
INPUT PROMPT "   Enter temperature display range in degrees ":max
LET u=0                                  ! u = input to the system
LET HEAT = 43.6
LET d=Tstp/5000                          ! d = integration increment
LET range=Tstp/d                         ! range = screen width of plot

!  Note: the screen is divided into increments based on the
!  integration increment.

!  This is the part that draws and labels the axes

LET ofh = .18*range                      ! ofh = horizontal offset
LET wide = 1.05*range+ofh                ! wide = width...
```

```
LET high = 1.2*max                      ! high=height of screen
CLEAR                                   ! clear screen
SET window 0, wide, HISET-high/2, HISET+high/2
!
LET ofv = HISET-.9*high/2               ! calc. vertical offset
PLOT ofh,ofv; ofh+range,ofv             ! draw horixontal axis
PLOT ofh,ofv; ofh,ofv+max               ! draw vertical axis
FOR i=ofh to range+ofh step range/10     ! draw numbers and
    LET j=(i-ofh)*d                     !    tick marks on
    LET s$=str$(j)                      !    the horizontal axis
    PLOT TEXT, at i-.01*wide,HISET-.49*high: s$
    PLOT i,ofv; i,max+ofv
NEXT i

FOR i=ofv  to max+ofv+.001*max step max/10 ! draw numbers and
    LET j=i-ofv+HISET-max/2             !    tick marks on
    LET s$=str$(j)                      !    the vertical axis
    PLOT TEXT, at .005*range,i-.01*high: s$
    PLOT ofh,i; ofh+range,i
NEXT i

!  Do actual calculation and plot result

LET TC=HISET-delta/2                    ! Initialize TC
LET sT=0                                ! Initialize dT/dt
LET TC1 = 1 + RA/RT
LET TC2 = RA*CA
LET T = TC/TC1
FOR n=0 to range
    LET ssT=-D1*sT-D2*T+NUM*u
    LET sT=sT+d*ssT
    LET T=T+d*sT
    LET TC = TC1*T+TC2*ST
    PLOT n+ofh,TC
    PLOT n+ofh,T+20
    IF TC < TOLD THEN LET SET = LOSET    ! Thermostat
    IF TC >= TOLD THEN LET SET = HISET   ! Equations

    IF TC < SET THEN LET U = HEAT
    IF TC >= SET THEN LET U = 0.0

    LET TOLD = TC
NEXT n
PLOT TEXT, at .3*wide,1425+.45*high: "DONE! (Press any key to
continue)"
END
```

```
! This program does a simulation of a furnace startup
! from room temperature in True Basic on Macintosh
!
! It calculates the maximum value automatically
!
! This program does a control system simulation using
! a digital simulation of an analog simulation technique.
! The integration is accomplished by keeping a running
! sum of the integrand multiplied by the delta ("d")

! Subroutine to find maximum output value
SUB find_max
    PLOT TEXT, at .4*range, high/2: "Calculating Graph Sizes"
    LET max=0
    FOR n=0 to range
        LET SSt=-3.05015E-4*st-2.54473E-9*t+3.5667E-7*u
        LET St=St+d*sst
        LET t=t+d*st
        IF max<t then LET max = t
    NEXT n
END SUB

! Initial Conditions
PRINT
PRINT          "   Enter stop time in seconds  "
INPUT PROMPT " (then press ""Return""):       ":Tstp
!LET tstp=.5                        ! tstp = simulation stop time
LET u=43.6                          ! u = input to the system
LET T=0                             ! t = initail temperature
LET sT=0                            ! st = dT/dt
LET d=Tstp/1000                     ! d = integration increment
LET range=Tstp/d                    ! range = screen width of plot

! Note: the screen is divided into increments based on the
! integration increment.

! This is the part that draws and labels the axes

LET ofh = .18*range                 ! ofh = horizontal offset
LET wide = 1.05*range+ofh           ! wide = width...
LET high = 260                      ! high = height...
SET window 0, wide, 0, high         !   ...of screen window
CALL find_max                       ! Find max output value
LET e=log10(max)                    ! This routine rounds
LET e=int(e+0.5)                    ! off the values on
LET max=max/(10^e)                  ! the y-axis
LET max=int(max+1)
LET max=max*10^e
```

```
LET high = 1.2*max                     ! high=height of screen
CLEAR                                  ! clear screen
SET window 0, wide, 0, high            ! reset window for max
LET ofv = 0.1*high                     ! calc. vertical offset
PLOT ofh,ofv; ofh+range,ofv            ! draw horixontal axis
PLOT ofh,ofv; ofh,ofv+max              ! draw vertical axis
FOR i=ofh to range+ofh step range/10     ! draw numbers and
    LET j=(i-ofh)*d                    ! tick marks on
    LET s$=str$(j)                     ! the horizontal axis
    PLOT TEXT, at i-.01*wide,ofv/2: s$
    PLOT i,ofv; i,max+ofv
NEXT i
LET tick=0.004*range                   ! scale vertical ticks
FOR i=ofv to max+ofv+.001 step max/10  ! draw numbers and
    LET j=i-ofv                        ! tick marks on
    LET s$=str$(j)                     ! the vertical axis
    PLOT TEXT, at .005*range,i-.01*high: s$
    PLOT ofh,i; ofh+range,i
NEXT i

! Do actual calculation and plot result

LET T=0                               ! Re-initialize T
LET sT=0                              ! Re-initialize dT/dt
FOR n=0 to range
    LET ssT=-3.05015E-4*sT-2.54473E-9*T+3.5667E-7*u
    LET sT=sT+d*ssT
    LET T=T+d*sT
    !    LET FM=SSU-X*SU*(1-U*U)+U
    PLOT n+ofh,T+ofv
NEXT n
PLOT TEXT, at .3*wide,.95*high: "DONE! (Press any key to continue)"
END
```

```
PROGRAM FURNACE 1.0

" This program uses a second order approximation to model "
" a group of furnaces. "

CONSTANT  CCB = 104.5          $" Crucible Capacitance "
CONSTANT  CCS = 71.81          $" Crucible Capacitance "
CONSTANT  DELTA = 10.0         $" Thermostat Deadband "
CONSTANT  HEAT1 = 43.6         $" Heat element (BTU/sec) "
CONSTANT  HEAT2 = 43.6         $" Heat element (BTU/sec) "
CONSTANT  HEAT3 = 43.6         $" Heat element (BTU/sec) "
CONSTANT  HEAT4 = 43.6         $" Heat element (BTU/sec) "
CONSTANT  HEAT5 = 43.6         $" Heat element (BTU/sec) "
CONSTANT  HEAT6 = 43.6         $" Heat element (BTU/sec) "
CONSTANT  HISET1 = 1290.0      $" High setpoint (above 110 F) "
CONSTANT  HISET2 = 1290.0      $" High setpoint (above 110 F) "
CONSTANT  HISET3 = 1290.0      $" High setpoint (above 110 F) "
CONSTANT  HISET4 = 1290.0      $" High setpoint (above 110 F) "
CONSTANT  HISET5 = 1290.0      $" High setpoint (above 110 F) "
CONSTANT  HISET6 = 1290.0      $" High setpoint (above 110 F) "
CONSTANT  RCB = 3.05           $" Big Crucible Resistance "
CONSTANT  RCS = 5.026          $" Small Crucible Resistance "
CONSTANT  RSEED = 111          $" Seed for Random Number "
CONSTANT  RT1 = 350.0          $" Top-Loss Resistance "
CONSTANT  RT2 = 757.0          $" Top-Loss Resistance "
CONSTANT  RT3 = 757.0          $" Top-Loss Resistance "
CONSTANT  RT4 = 757.0          $" Top-Loss Resistance "
CONSTANT  RT5 = 757.0          $" Top-Loss Resistance "
CONSTANT  RT6 = 350.0          $" Top-Loss Resistance "
CONSTANT  RWB = 525.5          $" Wall-Loss Resistance "
CONSTANT  RWS = 120.0          $" Wall-Loss Resistance "
CONSTANT  TSTP = 3600.0        $" Simulation Stop Time "
CONSTANT  TIMER = 120.0        $" Disturbance Rate "
CONSTANT  TZ = 0.0             $" Simulation Start Time "
CONSTANT  W1 = 600.0           $" Default Weight (lbs) "
CONSTANT  W2 = 400.0           $" Default Weight (lbs) "
CONSTANT  W3 = 400.0           $" Default Weight (lbs) "
CONSTANT  W4 = 400.0           $" Default Weight (lbs) "
CONSTANT  W5 = 400.0           $" Default Weight (lbs) "
CONSTANT  W6 = 600.0           $" Default Weight (lbs) "

INITIAL

CINTERVAL  CINT = 5.0          $" Communication Interval "
PT = 0.0                       $" Initialize Total Power "
UNIFI(RSEED)                   $" Initialization of Random Number "
RDIFF = 0.0                    $" Initialize RDIFF "
RTIME = 0.0                    $" Initialize RTIME "
KWHUSE = 0.0                   $" Initialize Kilowatt-hours used "
CM = 0.0                       $" Initialize KWH cost "
```

```
CMN = 0.0                          $" Initialize Normalized KWH cost "
CD = 0.0                           $" Initailize Demand cost "
EMC = 0.0                          $" Initialize Effective Monthly cost "
PEAK = 0.0                         $" Initialize Peak Power "
U1 = 0.0                           $" Initial heat input "
U2 = 0.0                           $" Initial heat input "
U3 = 0.0                           $" Initial heat input "
U4 = 0.0                           $" Initial heat input "
U5 = 0.0                           $" Initial heat input "
U6 = 0.0                           $" Initial heat input "
LOSET1 = HISET1 - DELTA            $" Low Setpoint "
LOSET2 = HISET2 - DELTA            $" Low Setpoint "
LOSET3 = HISET3 - DELTA            $" Low Setpoint "
LOSET4 = HISET4 - DELTA            $" Low Setpoint "
LOSET5 = HISET5 - DELTA            $" Low Setpoint "
LOSET6 = HISET6 - DELTA            $" Low Setpoint "
TC1 = HISET1 - DELTA/2             $" Thermocouple Initialize "
TC2 = HISET2 - DELTA/2             $" Thermocouple Initialize "
TC3 = HISET3 - DELTA/2             $" Thermocouple Initialize "
TC4 = HISET4 - DELTA/2             $" Thermocouple Initialize "
TC5 = HISET5 - DELTA/2             $" Thermocouple Initialize "
TC6 = HISET6 - DELTA/2             $" Thermocouple Initialize "
TOLD1 = 0.0                        $" Initialize Memory "
TOLD2 = 0.0                        $" Initialize Memory "
TOLD3 = 0.0                        $" Initialize Memory "
TOLD4 = 0.0                        $" Initialize Memory "
TOLD5 = 0.0                        $" Initialize Memory "
TOLD6 = 0.0                        $" Initialize Memory "
CA1 = 0.2727*W1                    $" Aluminum Capacitance "
CA2 = 0.2727*W2                    $" Aluminum Capacitance "
CA3 = 0.2727*W3                    $" Aluminum Capacitance "
CA4 = 0.2727*W4                    $" Aluminum Capacitance "
CA5 = 0.2727*W5                    $" Aluminum Capacitance "
CA6 = 0.2727*W6                    $" Aluminum Capacitance "
RA1 = 8987.0/W1                    $" Initialize Aluminum Res. "
RA2 = 3994.0/W2                    $" Initialize Aluminum Res. "
RA3 = 3994.0/W3                    $" Initialize Aluminum Res. "
RA4 = 3994.0/W4                    $" Initialize Aluminum Res. "
RA5 = 3994.0/W5                    $" Initialize Aluminum Res. "
RA6 = 8987.0/W6                    $" Initialize Aluminum Res. "
TA01 = TC1/(1 + RA1/RT1)           $" Initialize Aluminum Temp "
TA02 = TC2/(1 + RA2/RT2)           $" Initialize Aluminum Temp "
TA03 = TC3/(1 + RA3/RT3)           $" Initialize Aluminum Temp "
TA04 = TC4/(1 + RA4/RT4)           $" Initialize Aluminum Temp "
TA05 = TC5/(1 + RA5/RT5)           $" Initialize Aluminum Temp "
TA06 = TC6/(1 + RA6/RT6)           $" Initialize Aluminum Temp "

" Numerator and Denominator Coefficients: "

NUM1 = RWB/(RA1*CA1*(RCB+RWB)*CCB)
```

```
DA1 = 1/((RCB+RWB)*CCB) + 1/(RA1*CCB) ...
      + 1/(RA1*CA1) + 1/(RT1*CA1)
DB1 = 1/(RA1*CA1*CCB*(RCB+RWB)) ...
      + 1/(RT1*CA1*CCB*(RCB+RWB)) ...
      + 1/(RA1*RT1*CA1*CCB)
NUM2 = RWS/(RA2*CA2*(RCS+RWS)*CCS)
DA2 = 1/((RCS+RWS)*CCS) + 1/(RA2*CCS) ...
      + 1/(RA2*CA2) + 1/(RT2*CA2)
DB2 = 1/(RA2*CA2*CCS*(RCS+RWS)) ...
      + 1/(RT2*CA2*CCS*(RCS+RWS)) ...
      + 1/(RA2*RT2*CA2*CCS)
NUM3 = RWS/(RA3*CA3*(RCS+RWS)*CCS)
DA3 = 1/((RCS+RWS)*CCS) + 1/(RA3*CCS) ...
      + 1/(RA3*CA3) + 1/(RT3*CA3)
DB3 = 1/(RA3*CA3*CCS*(RCS+RWS)) ...
      + 1/(RT3*CA3*CCS*(RCS+RWS)) ...
      + 1/(RA3*RT3*CA3*CCS)
NUM4 = RWS/(RA4*CA4*(RCS+RWS)*CCS)
DA4 = 1/((RCS+RWS)*CCS) + 1/(RA4*CCS) ...
      + 1/(RA4*CA4) + 1/(RT4*CA4)
DB4 = 1/(RA4*CA4*CCS*(RCS+RWS)) ...
      + 1/(RT4*CA4*CCS*(RCS+RWS)) ...
      + 1/(RA4*RT4*CA4*CCS)
NUM5 = RWS/(RA5*CA5*(RCS+RWS)*CCS)
DA5 = 1/((RCS+RWS)*CCS) + 1/(RA5*CCS) ...
      + 1/(RA5*CA5) + 1/(RT5*CA5)
DB5 = 1/(RA5*CA5*CCS*(RCS+RWS)) ...
      + 1/(RT5*CA5*CCS*(RCS+RWS)) ...
      + 1/(RA5*RT5*CA5*CCS)
NUM6 = RWB/(RA6*CA6*(RCB+RWB)*CCB)
DA6 = 1/((RCB+RWB)*CCB) + 1/(RA6*CCB) ...
      + 1/(RA6*CA6) + 1/(RT6*CA6)
DB6 = 1/(RA6*CA6*CCB*(RCB+RWB)) ...
      + 1/(RT6*CA6*CCB*(RCB+RWB)) ...
      + 1/(RA6*RT6*CA6*CCB)

END                                         $" Of Initial Section "

DYNAMIC

SST1 = NUM1*U1 - DA1*ST1 - DB1*TA1          $" Second Derivative "
SST2 = NUM2*U2 - DA2*ST2 - DB2*TA2          $" Second Derivative "
SST3 = NUM3*U3 - DA3*ST3 - DB3*TA3          $" Second Derivative "
SST4 = NUM4*U4 - DA4*ST4 - DB4*TA4          $" Second Derivative "
SST5 = NUM5*U5 - DA5*ST5 - DB5*TA5          $" Second Derivative "
SST6 = NUM6*U6 - DA6*ST6 - DB6*TA6          $" Second Derivative "

DERIVATIVE
ST1 = INTEG(SST1,0.0)                       $" First Derivative "
ST2 = INTEG(SST2,0.0)                       $" First Derivative "
```

```
ST3 = INTEG(SST3,0.0)                    $" First Derivative "
ST4 = INTEG(SST4,0.0)                    $" First Derivative "
ST5 = INTEG(SST5,0.0)                    $" First Derivative "
ST6 = INTEG(SST6,0.0)                    $" First Derivative "
TA1 = INTEG(ST1,TA01)                    $" Aluminum Temperature "
TA2 = INTEG(ST2,TA02)                    $" Aluminum Temperature "
TA3 = INTEG(ST3,TA03)                    $" Aluminum Temperature "
TA4 = INTEG(ST4,TA04)                    $" Aluminum Temperature "
TA5 = INTEG(ST5,TA05)                    $" Aluminum Temperature "
TA6 = INTEG(ST6,TA06)                    $" Aluminum Temperature "
KWHUSE = INTEG(PT,0.0)/3600.0            $" Kilowatt Hours "
END                                      $" Of Derivative Section "

TC1 = (1 + RA1/RT1)*TA1 + (RA1*CA1)*ST1 $" Crubicle Temperature "
TC2 = (1 + RA2/RT2)*TA2 + (RA2*CA2)*ST2 $" Crubicle Temperature "
TC3 = (1 + RA3/RT3)*TA3 + (RA3*CA3)*ST3 $" Crubicle Temperature "
TC4 = (1 + RA4/RT4)*TA4 + (RA4*CA4)*ST4 $" Crubicle Temperature "
TC5 = (1 + RA5/RT5)*TA5 + (RA5*CA5)*ST5 $" Crubicle Temperature "
TC6 = (1 + RA6/RT6)*TA6 + (RA6*CA6)*ST6 $" Crubicle Temperature "

PROCEDURAL     $" This Section for 6 Independent Thermostats "

  IF (TC1 .LT. TOLD1) SET1 = LOSET1     $" Falling Temperature "
  IF (TC1 .GE. TOLD1) SET1 = HISET1     $" Rising Temperature "
  IF (TC1 .LT. SET1) U1 = HEAT1         $" Heat On "
  IF (TC1 .GE. SET1) U1 = 0.0           $" Heat Off "
  TOLD1 = TC1
  IF (TC2 .LT. TOLD2) SET2 = LOSET2     $" Falling Temperature "
  IF (TC2 .GE. TOLD2) SET2 = HISET2     $" Rising Temperature "
  IF (TC2 .LT. SET2) U2 = HEAT2         $" Heat On "
  IF (TC2 .GE. SET2) U2 = 0.0           $" Heat Off "
  TOLD2 = TC2
  IF (TC3 .LT. TOLD3) SET3 = LOSET3     $" Falling Temperature "
  IF (TC3 .GE. TOLD3) SET3 = HISET3     $" Rising Temperature "
  IF (TC3 .LT. SET3) U3 = HEAT3         $" Heat On "
  IF (TC3 .GE. SET3) U3 = 0.0           $" Heat Off "
  TOLD3 = TC3
  IF (TC4 .LT. TOLD4) SET4 = LOSET4     $" Falling Temperature "
  IF (TC4 .GE. TOLD4) SET4 = HISET4     $" Rising Temperature "
  IF (TC4 .LT. SET4) U4 = HEAT4         $" Heat On "
  IF (TC4 .GE. SET4) U4 = 0.0           $" Heat Off "
  TOLD4 = TC4
  IF (TC5 .LT. TOLD5) SET5 = LOSET5     $" Falling Temperature "
  IF (TC5 .GE. TOLD5) SET5 = HISET5     $" Rising Temperature "
  IF (TC5 .LT. SET5) U5 = HEAT5         $" Heat On "
  IF (TC5 .GE. SET5) U5 = 0.0           $" Heat Off "
  TOLD5 = TC5
  IF (TC6 .LT. TOLD6) SET6 = LOSET6     $" Falling Temperature "
  IF (TC6 .GE. TOLD6) SET6 = HISET6     $" Rising Temperature "
  IF (TC6 .LT. SET6) U6 = HEAT6         $" Heat On "
```

```
        IF (TC6 .GE. SET6) U6 = 0.0            $" Heat Off "
        TOLD6 = TC6

      PT = (U1+U2+U3+U4+U5+U6)/0.9478          $" Total Power "

" Calculate Effective Monthly Power Cost "
      IF(PT.GT.PEAK) PEAK = PT                 $" Calculate peak demand "
      CM = KWHUSE*0.06278 + ( (KWHUSE-15000.0)*0.03034 ...
           - (KWHUSE-15000.0)*0.06278 )*STEP(15000.0-KWHUSE)
      IF(T.GT.0.0) CMN = (2592000.0/t)*CM
      CD = 9.50*(peak-50.0)
      EMC = 8.80 + CMN + CD                    $" Effect Monthly Cost "

      RDIFF = T - RTIME                        $" Elapsed Time Counter "
      IF(RDIFF .LT. TIMER) GO TO 700           $" Adj Every (TIMER) Sec "
      RAND = UNIF(0.0,6.0)                     $" Random # Selects Furn "
      RANDB = UNIF(200.0,800.0)                $" Random Number, 200-800 "
      RANDS = UNIF(100.0,500.0)                $" Random Number, 100-500 "
      IF (RAND .LT. 1.0) GO TO 100
      IF ((RAND .GE. 1.0).AND.(RAND .LT. 2.0)) GO TO 200
      IF ((RAND .GE. 2.0).AND.(RAND .LT. 3.0)) GO TO 300
      IF ((RAND .GE. 3.0).AND.(RAND .LT. 4.0)) GO TO 400
      IF ((RAND .GE. 4.0).AND.(RAND .LT. 5.0)) GO TO 500
      IF ((RAND .GE. 5.0).AND.(RAND .LT. 6.0)) GO TO 600
      100.. W1 = RANDB                         $" Random Weight, 200-800 "
       RA1 = 1997.0/W1                         $" Aluminum Resistance "
       CA1 = 0.2727*W1                         $" Aluminum Capacitance "
       NUM1 = RWB/(RA1*CA1*(RCB+RWB)*CCB)      $" Numerator "
       DA1 = 1/((RCB+RWB)*CCB) + 1/(RA1*CCB) ...
             + 1/(RA1*CA1) + 1/(RT1*CA1)
       DB1 = 1/(RA1*CA1*CCB*(RCB+RWB)) ...
             + 1/(RT1*CA1*CCB*(RCB+RWB)) ...
             + 1/(RA1*RT1*CA1*CCB)
      RTIME = T                                $" Reset Elapsed Time "
      GO TO 700
      200.. W2 = RANDS                         $" Random Weight, 100-500 "
       RA2 = 1997.0/W2                         $" Aluminum Resistance "
       CA2 = 0.2727*W2                         $" Aluminum Capacitance "
       NUM2 = RWS/(RA2*CA2*(RCS+RWS)*CCS)      $" Numerator "
       DA2 = 1/((RCS+RWS)*CCS) + 1/(RA2*CCS) ...
             + 1/(RA2*CA2) + 1/(RT2*CA2)
       DB2 = 1/(RA2*CA2*CCS*(RCS+RWS)) ...
             + 1/(RT2*CA2*CCS*(RCS+RWS)) ...
             + 1/(RA2*RT2*CA2*CCS)
      RTIME = T                                $" Reset Elapsed Time "
      GO TO 700
      300.. W3 = RANDS                         $" Random Weight, 100-500 "
       RA3 = 1997.0/W3                         $" Aluminum Resistance "
       CA3 = 0.2727*W3                         $" Aluminum Capacitance "
       NUM3 = RWS/(RA3*CA3*(RCS+RWS)*CCS)      $" Numerator "
```

```
 DA3 = 1/((RCS+RWS)*CCS) + 1/(RA3*CCS) ...
       + 1/(RA3*CA3) + 1/(RT3*CA3)
 DB3 = 1/(RA3*CA3*CCS*(RCS+RWS)) ...
       + 1/(RT3*CA3*CCS*(RCS+RWS)) ...
       + 1/(RA3*RT3*CA3*CCS)
RTIME = T                               $" Reset Elapsed Time "
GO TO 700
400.. W4 = RANDS                        $" Random Weight, 100-500 "
 RA4 = 1997.0/W4                        $" Aluminum Resistance "
 CA4 = 0.2727*W4                        $" Aluminum Capacitance "
 NUM4 = RWS/(RA4*CA4*(RCS+RWS)*CCS)     $" Numerator "
 DA4 = 1/((RCS+RWS)*CCS) + 1/(RA4*CCS) ...
       + 1/(RA4*CA4) + 1/(RT4*CA4)
 DB4 = 1/(RA4*CA4*CCS*(RCS+RWS)) ...
       + 1/(RT4*CA4*CCS*(RCS+RWS)) ...
       + 1/(RA4*RT4*CA4*CCS)
RTIME = T                               $" Reset Elapsed Time "
GO TO 700
500.. W5 = RANDS                        $" Random Weight, 100-500 "
 RA5 = 1997.0/W5                        $" Aluminum Resistance "
 CA5 = 0.2727*W5                        $" Aluminum Capacitance "
 NUM5 = RWS/(RA5*CA5*(RCS+RWS)*CCS)     $" Numerator "
 DA5 = 1/((RCS+RWS)*CCS) + 1/(RA5*CCS) ...
       + 1/(RA5*CA5) + 1/(RT5*CA5)
 DB5 = 1/(RA5*CA5*CCS*(RCS+RWS)) ...
       + 1/(RT5*CA5*CCS*(RCS+RWS)) ...
       + 1/(RA5*RT5*CA5*CCS)
RTIME = T                               $" Reset Elapsed Time "
GO TO 700
600.. W6 = RANDB                        $" Random Weight, 200-800 "
 RA6 = 1997.0/W6                        $" Aluminum Resistance "
 CA6 = 0.2727*W6                        $" Aluminum Capacitance "
 NUM6 = RWB/(RA6*CA6*(RCB+RWB)*CCB)     $" Numerator "
 DA6 = 1/((RCB+RWB)*CCB) + 1/(RA6*CCB) ...
       + 1/(RA6*CA6) + 1/(RT6*CA6)
 DB6 = 1/(RA6*CA6*CCB*(RCB+RWB)) ...
       + 1/(RT6*CA6*CCB*(RCB+RWB)) ...
       + 1/(RA6*RT6*CA6*CCB)
RTIME = T                               $" Reset Elapsed Time "
700.. CONTINUE                          $" End of Conditional "
   END                                  $" End of Procedural "

TERMT(T.GE.TSTP)                        $" Stop Instruction "
END                                     $" Of Dynamic "

END                                     $" End of Program "
```

```
PROGRAM CONTROL1 1.0

" This program uses a second order approximation to model "
" a group of furnaces. "
" Then it sets a maximum number of furnaces to be on at a "
" time.  This variable is NMAX. "

CONSTANT  CCB = 104.5          $" Crucible Capacitance "
CONSTANT  CCS = 71.81          $" Crucible Capacitance "
CONSTANT  DELTA = 10.0         $" Thermostat Deadband "
CONSTANT  HEAT1 = 43.6         $" Heat element (BTU/sec) "
CONSTANT  HEAT2 = 43.6         $" Heat element (BTU/sec) "
CONSTANT  HEAT3 = 43.6         $" Heat element (BTU/sec) "
CONSTANT  HEAT4 = 43.6         $" Heat element (BTU/sec) "
CONSTANT  HEAT5 = 43.6         $" Heat element (BTU/sec) "
CONSTANT  HEAT6 = 43.6         $" Heat element (BTU/sec) "
CONSTANT  HISET1 = 1290.0      $" High setpoint (above 110 F) "
CONSTANT  HISET2 = 1290.0      $" High setpoint (above 110 F) "
CONSTANT  HISET3 = 1290.0      $" High setpoint (above 110 F) "
CONSTANT  HISET4 = 1290.0      $" High setpoint (above 110 F) "
CONSTANT  HISET5 = 1290.0      $" High setpoint (above 110 F) "
CONSTANT  HISET6 = 1290.0      $" High setpoint (above 110 F) "
CONSTANT  NMAX = 3.0           $" Maximum furnaces on "
CONSTANT  PMAX = 0.0           $" Maximum allowable power "
CONSTANT  RCB = 3.05           $" Big Crucible Resistance "
CONSTANT  RCS = 5.026          $" Small Crucible Resistance "
CONSTANT  RSEED = 99           $" Seed for Random Number "
CONSTANT  RT1 = 350.0          $" Top-Loss Resistance "
CONSTANT  RT2 = 757.0          $" Top-Loss Resistance "
CONSTANT  RT3 = 757.0          $" Top-Loss Resistance "
CONSTANT  RT4 = 757.0          $" Top-Loss Resistance "
CONSTANT  RT5 = 757.0          $" Top-Loss Resistance "
CONSTANT  RT6 = 350.0          $" Top-Loss Resistance "
CONSTANT  RWB = 525.5          $" Wall-Loss Resistance "
CONSTANT  RWS = 120.0          $" Wall-Loss Resistance "
CONSTANT  TSTP = 3600.0        $" Simulation Stop Time "
CONSTANT  TIMER = 120.0        $" Disturbance Rate "
CONSTANT  TZ = 0.0             $" Simulation Start Time "
CONSTANT  W1 = 600.0           $" Default Weight (lbs) "
CONSTANT  W2 = 400.0           $" Default Weight (lbs) "
CONSTANT  W3 = 400.0           $" Default Weight (lbs) "
CONSTANT  W4 = 400.0           $" Default Weight (lbs) "
CONSTANT  W5 = 400.0           $" Default Weight (lbs) "
CONSTANT  W6 = 600.0           $" Default Weight (lbs) "

INITIAL

CINTERVAL  CINT = 5.0          $" Communication Interval "
PT = 0.0                       $" Initialize Total Power "
     KWHUSE = 0.0              $" Initialize Kilowatt-hours used "
     CM = 0.0                  $" Initialize KWH cost "
```

```
     CMN = 0.0                    $" Initialize Normalized KWH cost "
     CD = 0.0                     $" Initailize Demand cost "
     EMC = 0.0                    $" Initialize Effect. Monthly cost "
     PEAK = 0.0                   $" Initialize Peak Power "
UNIFI(RSEED)                      $" Initialization of Random Number "
RDIFF = 0.0                       $" Initialize RDIFF "
RTIME = 0.0                       $" Initialize RTIME "
U1 = 0.0                          $" Initial heat input "
U2 = 0.0                          $" Initial heat input "
U3 = 0.0                          $" Initial heat input "
U4 = 0.0                          $" Initial heat input "
U5 = 0.0                          $" Initial heat input "
U6 = 0.0                          $" Initial heat input "
LOSET1 = HISET1 - DELTA           $" Low Setpoint "
LOSET2 = HISET2 - DELTA           $" Low Setpoint "
LOSET3 = HISET3 - DELTA           $" Low Setpoint "
LOSET4 = HISET4 - DELTA           $" Low Setpoint "
LOSET5 = HISET5 - DELTA           $" Low Setpoint "
LOSET6 = HISET6 - DELTA           $" Low Setpoint "
TC1 = HISET1 - DELTA/2            $" Thermocouple Initialize "
TC2 = HISET2 - DELTA/2            $" Thermocouple Initialize "
TC3 = HISET3 - DELTA/2            $" Thermocouple Initialize "
TC4 = HISET4 - DELTA/2            $" Thermocouple Initialize "
TC5 = HISET5 - DELTA/2            $" Thermocouple Initialize "
TC6 = HISET6 - DELTA/2            $" Thermocouple Initialize "
TOLD1 = 0.0                       $" Initialize Memory "
TOLD2 = 0.0                       $" Initialize Memory "
TOLD3 = 0.0                       $" Initialize Memory "
TOLD4 = 0.0                       $" Initialize Memory "
TOLD5 = 0.0                       $" Initialize Memory "
TOLD6 = 0.0                       $" Initialize Memory "
CA1 = 0.2727*W1                   $" Aluminum Capacitance "
CA2 = 0.2727*W2                   $" Aluminum Capacitance "
CA3 = 0.2727*W3                   $" Aluminum Capacitance "
CA4 = 0.2727*W4                   $" Aluminum Capacitance "
CA5 = 0.2727*W5                   $" Aluminum Capacitance "
CA6 = 0.2727*W6                   $" Aluminum Capacitance "
RA1 = 8987.0/W1                   $" Initialize Aluminum Res. "
RA2 = 3994.0/W2                   $" Initialize Aluminum Res. "
RA3 = 3994.0/W3                   $" Initialize Aluminum Res. "
RA4 = 3994.0/W4                   $" Initialize Aluminum Res. "
RA5 = 3994.0/W5                   $" Initialize Aluminum Res. "
RA6 = 8987.0/W6                   $" Initialize Aluminum Res. "
TA01 = TC1/(1 + RA1/RT1)          $" Initialize Aluminum Temp "
TA02 = TC2/(1 + RA2/RT2)          $" Initialize Aluminum Temp "
TA03 = TC3/(1 + RA3/RT3)          $" Initialize Aluminum Temp "
TA04 = TC4/(1 + RA4/RT4)          $" Initialize Aluminum Temp "
TA05 = TC5/(1 + RA5/RT5)          $" Initialize Aluminum Temp "
TA06 = TC6/(1 + RA6/RT6)          $" Initialize Aluminum Temp "
```

```
" Numerator and Denominator Coefficients: "

NUM1 = RWB/(RA1*CA1*(RCB+RWB)*CCB)
DA1  = 1/((RCB+RWB)*CCB) + 1/(RA1*CCB) ...
       + 1/(RA1*CA1) + 1/(RT1*CA1)
DB1  = 1/(RA1*CA1*CCB*(RCB+RWB)) ...
       + 1/(RT1*CA1*CCB*(RCB+RWB)) ...
       + 1/(RA1*RT1*CA1*CCB)
NUM2 = RWS/(RA2*CA2*(RCS+RWS)*CCS)
DA2  = 1/((RCS+RWS)*CCS) + 1/(RA2*CCS) ...
       + 1/(RA2*CA2) + 1/(RT2*CA2)
DB2  = 1/(RA2*CA2*CCS*(RCS+RWS)) ...
       + 1/(RT2*CA2*CCS*(RCS+RWS)) ...
       + 1/(RA2*RT2*CA2*CCS)
NUM3 = RWS/(RA3*CA3*(RCS+RWS)*CCS)
DA3  = 1/((RCS+RWS)*CCS) + 1/(RA3*CCS) ...
       + 1/(RA3*CA3) + 1/(RT3*CA3)
DB3  = 1/(RA3*CA3*CCS*(RCS+RWS)) ...
       + 1/(RT3*CA3*CCS*(RCS+RWS)) ...
       + 1/(RA3*RT3*CA3*CCS)
NUM4 = RWS/(RA4*CA4*(RCS+RWS)*CCS)
DA4  = 1/((RCS+RWS)*CCS) + 1/(RA4*CCS) ...
       + 1/(RA4*CA4) + 1/(RT4*CA4)
DB4  = 1/(RA4*CA4*CCS*(RCS+RWS)) ...
       + 1/(RT4*CA4*CCS*(RCS+RWS)) ...
       + 1/(RA4*RT4*CA4*CCS)
NUM5 = RWS/(RA5*CA5*(RCS+RWS)*CCS)
DA5  = 1/((RCS+RWS)*CCS) + 1/(RA5*CCS) ...
       + 1/(RA5*CA5) + 1/(RT5*CA5)
DB5  = 1/(RA5*CA5*CCS*(RCS+RWS)) ...
       + 1/(RT5*CA5*CCS*(RCS+RWS)) ...
       + 1/(RA5*RT5*CA5*CCS)
NUM6 = RWB/(RA6*CA6*(RCB+RWB)*CCB)
DA6  = 1/((RCB+RWB)*CCB) + 1/(RA6*CCB) ...
       + 1/(RA6*CA6) + 1/(RT6*CA6)
DB6  = 1/(RA6*CA6*CCB*(RCB+RWB)) ...
       + 1/(RT6*CA6*CCB*(RCB+RWB)) ...
       + 1/(RA6*RT6*CA6*CCB)

END                                        $" Of Initial Section "

DYNAMIC

SST1 = NUM1*U1 - DA1*ST1 - DB1*TA1         $" Second Derivative "
SST2 = NUM2*U2 - DA2*ST2 - DB2*TA2         $" Second Derivative "
SST3 = NUM3*U3 - DA3*ST3 - DB3*TA3         $" Second Derivative "
SST4 = NUM4*U4 - DA4*ST4 - DB4*TA4         $" Second Derivative "
SST5 = NUM5*U5 - DA5*ST5 - DB5*TA5         $" Second Derivative "
SST6 = NUM6*U6 - DA6*ST6 - DB6*TA6         $" Second Derivative "
```

```
DERIVATIVE

ST1 = INTEG(SST1,0.0)                        $" First Derivative "
ST2 = INTEG(SST2,0.0)                        $" First Derivative "
ST3 = INTEG(SST3,0.0)                        $" First Derivative "
ST4 = INTEG(SST4,0.0)                        $" First Derivative "
ST5 = INTEG(SST5,0.0)                        $" First Derivative "
ST6 = INTEG(SST6,0.0)                        $" First Derivative "
TA1 = INTEG(ST1,TA01)                        $" Aluminum Temperature "
TA2 = INTEG(ST2,TA02)                        $" Aluminum Temperature "
TA3 = INTEG(ST3,TA03)                        $" Aluminum Temperature "
TA4 = INTEG(ST4,TA04)                        $" Aluminum Temperature "
TA5 = INTEG(ST5,TA05)                        $" Aluminum Temperature "
TA6 = INTEG(ST6,TA06)                        $" Aluminum Temperature "
KWHUSE = INTEG(PT,0.0)/3600.0                $" Kilowatt Hours "

END                                          $" Of Derivative Section "

TC1 = (1 + RA1/RT1)*TA1 + (RA1*CA1)*ST1 $" Crucible Temperature "
TC2 = (1 + RA2/RT2)*TA2 + (RA2*CA2)*ST2 $" Crucible Temperature "
TC3 = (1 + RA3/RT3)*TA3 + (RA3*CA3)*ST3 $" Crucible Temperature "
TC4 = (1 + RA4/RT4)*TA4 + (RA4*CA4)*ST4 $" Crucible Temperature "
TC5 = (1 + RA5/RT5)*TA5 + (RA5*CA5)*ST5 $" Crucible Temperature "
TC6 = (1 + RA6/RT6)*TA6 + (RA6*CA6)*ST6 $" Crucible Temperature "

PROCEDURAL     $" This Section for 6 Independent Thermostats "

pmax=nmax*43.6/0.9478

  IF (TC1 .LT. TOLD1) SET1 = LOSET1       $" Falling Temperature "
  IF (TC1 .GE. TOLD1) SET1 = HISET1       $" Rising Temperature "
  IF ((TC1.LT.SET1).and.(pt.lt.pmax)) U1 = HEAT1     $" Heat On "
  IF (TC1 .GE. SET1) U1 = 0.0             $" Heat Off "
PT = (U1+U2+U3+U4+U5+U6)/0.9478           $" Total Power in KW "
  TOLD1 = TC1
  IF (TC2 .LT. TOLD2) SET2 = LOSET2       $" Falling Temperature "
  IF (TC2 .GE. TOLD2) SET2 = HISET2       $" Rising Temperature "
  IF ((TC2.LT.SET2).and.(pt.lt.pmax))  U2 = HEAT2    $" Heat On "
  IF (TC2 .GE. SET2) U2 = 0.0             $" Heat Off "
PT = (U1+U2+U3+U4+U5+U6)/0.9478           $" Total Power in KW "
  TOLD2 = TC2
  IF (TC3 .LT. TOLD3) SET3 = LOSET3       $" Falling Temperature "
  IF (TC3 .GE. TOLD3) SET3 = HISET3       $" Rising Temperature "
  IF ((TC3.LT.SET3).and.(pt.lt.pmax))  U3 = HEAT3    $" Heat On "
  IF (TC3 .GE. SET3) U3 = 0.0             $" Heat Off "
PT = (U1+U2+U3+U4+U5+U6)/0.9478           $" Total Power in KW "
  TOLD3 = TC3
  IF (TC4 .LT. TOLD4) SET4 = LOSET4       $" Falling Temperature "
  IF (TC4 .GE. TOLD4) SET4 = HISET4       $" Rising Temperature "
  IF ((TC4.LT.SET4).and.(pt.lt.pmax))  U4 = HEAT4    $" Heat On "
  IF (TC4 .GE. SET4) U4 = 0.0             $" Heat Off "
```

```
PT = (U1+U2+U3+U4+U5+U6)/0.9478           $" Total Power in KW "
  TOLD4 = TC4
  IF (TC5 .LT. TOLD5) SET5 = LOSET5       $" Falling Temperature "
  IF (TC5 .GE. TOLD5) SET5 = HISET5       $" Rising Temperature "
  IF ((TC5.LT.SET5).and.(pt.lt.pmax))  U5 = HEAT5   $" Heat On "
  IF (TC5 .GE. SET5) U5 = 0.0             $" Heat Off "
PT = (U1+U2+U3+U4+U5+U6)/0.9478           $" Total Power in KW "
  TOLD5 = TC5
  IF (TC6 .LT. TOLD6) SET6 = LOSET6       $" Falling Temperature "
  IF (TC6 .GE. TOLD6) SET6 = HISET6       $" Rising Temperature "
  IF ((TC6.LT.SET6).and.(pt.lt.pmax))  U6 = HEAT6   $" Heat On "
  IF (TC6 .GE. SET6) U6 = 0.0             $" Heat Off "
  TOLD6 = TC6
PT = (U1+U2+U3+U4+U5+U6)/0.9478           $" Total Power in KW "


" Calculate Effective Monthly Power Cost "

IF(PT.GT.PEAK) PEAK = PT                  $" Calculate peak demand "
CM = KWHUSE*0.06278 + ( (KWHUSE-15000.0)*0.03034 ...
     - (KWHUSE-15000.0)*0.06278 )*STEP(15000.0-KWHUSE)
IF(T.GT.0.0) CMN = (2592000.0/t)*CM
CD = 9.50*(peak-50.0)
EMC = 8.80 + CMN + CD                     $" Effect. Monthly Cost "

RDIFF = T - RTIME                         $" Elapsed Time Counter "
IF(RDIFF .LT. TIMER) GO TO 700            $" Adj Every (TIMER) Sec "
RAND = UNIF(0.0,6.0)                      $" Random # Selects Furn. "
RANDB = UNIF(200.0,800.0)                 $" Random #, 200 to 800 "
RANDS = UNIF(100.0,500.0)                 $" Random #, 100 to 500 "
IF (RAND .LT. 1.0) GO TO 100
IF ((RAND .GE. 1.0).AND.(RAND .LT. 2.0)) GO TO 200
IF ((RAND .GE. 2.0).AND.(RAND .LT. 3.0)) GO TO 300
IF ((RAND .GE. 3.0).AND.(RAND .LT. 4.0)) GO TO 400
IF ((RAND .GE. 4.0).AND.(RAND .LT. 5.0)) GO TO 500
IF ((RAND .GE. 5.0).AND.(RAND .LT. 6.0)) GO TO 600
100.. W1 = RANDB                          $" Rand Weight, 200-800 "
 RA1 = 1997.0/W1                          $" Aluminum Resistance "
 CA1 = 0.2727*W1                          $" Aluminum Capacitance "
 NUM1 = RWB/(RA1*CA1*(RCB+RWB)*CCB)       $" Numerator "
 DA1 = 1/((RCB+RWB)*CCB) + 1/(RA1*CCB) ...
     + 1/(RA1*CA1) + 1/(RT1*CA1)
 DB1 = 1/(RA1*CA1*CCB*(RCB+RWB)) ...
     + 1/(RT1*CA1*CCB*(RCB+RWB)) ...
     + 1/(RA1*RT1*CA1*CCB)
RTIME = T                                 $" Reset Elapsed Time "
GO TO 700
```

```
200.. W2 = RANDS                              $" Rand Weight, 100-500 "
 RA2 = 1997.0/W2                              $" Aluminum Resistance "
 CA2 = 0.2727*W2                              $" Aluminum Capacitance "
 NUM2 = RWS/(RA2*CA2*(RCS+RWS)*CCS)           $" Numerator "
 DA2 = 1/((RCS+RWS)*CCS) + 1/(RA2*CCS) ...
       + 1/(RA2*CA2) + 1/(RT2*CA2)
 DB2 = 1/(RA2*CA2*CCS*(RCS+RWS)) ...
       + 1/(RT2*CA2*CCS*(RCS+RWS)) ...
       + 1/(RA2*RT2*CA2*CCS)
RTIME = T                                     $" Reset Elapsed Time "
GO TO 700
300.. W3 = RANDS                              $" Rand Weight, 100-500 "
 RA3 = 1997.0/W3                              $" Aluminum Resistance "
 CA3 = 0.2727*W3                              $" Aluminum Capacitance "
 NUM3 = RWS/(RA3*CA3*(RCS+RWS)*CCS)           $" Numerator "
 DA3 = 1/((RCS+RWS)*CCS) + 1/(RA3*CCS) ...
       + 1/(RA3*CA3) + 1/(RT3*CA3)
 DB3 = 1/(RA3*CA3*CCS*(RCS+RWS)) ...
       + 1/(RT3*CA3*CCS*(RCS+RWS)) ...
       + 1/(RA3*RT3*CA3*CCS)
RTIME = T                                     $" Reset Elapsed Time "
GO TO 700
400.. W4 = RANDS                              $" Rand Weight, 100-500 "
 RA4 = 1997.0/W4                              $" Aluminum Resistance "
 CA4 = 0.2727*W4                              $" Aluminum Capacitance "
 NUM4 = RWS/(RA4*CA4*(RCS+RWS)*CCS)           $" Numerator "
 DA4 = 1/((RCS+RWS)*CCS) + 1/(RA4*CCS) ...
       + 1/(RA4*CA4) + 1/(RT4*CA4)
 DB4 = 1/(RA4*CA4*CCS*(RCS+RWS)) ...
       + 1/(RT4*CA4*CCS*(RCS+RWS)) ...
       + 1/(RA4*RT4*CA4*CCS)
RTIME = T                                     $" Reset Elapsed Time "
GO TO 700
500.. W5 = RANDS                              $" Rand Weight, 100-500 "
 RA5 = 1997.0/W5                              $" Aluminum Resistance "
 CA5 = 0.2727*W5                              $" Aluminum Capacitance "
 NUM5 = RWS/(RA5*CA5*(RCS+RWS)*CCS)           $" Numerator "
 DA5 = 1/((RCS+RWS)*CCS) + 1/(RA5*CCS) ...
       + 1/(RA5*CA5) + 1/(RT5*CA5)
 DB5 = 1/(RA5*CA5*CCS*(RCS+RWS)) ...
       + 1/(RT5*CA5*CCS*(RCS+RWS)) ...
       + 1/(RA5*RT5*CA5*CCS)
RTIME = T                                     $" Reset Elapsed Time "
GO TO 700
600.. W6 = RANDB                              $" Rand Weight, 200-800 "
 RA6 = 1997.0/W6                              $" Aluminum Resistance "
 CA6 = 0.2727*W6                              $" Aluminum Capacitance "
 NUM6 = RWB/(RA6*CA6*(RCB+RWB)*CCB)           $" Numerator "
 DA6 = 1/((RCB+RWB)*CCB) + 1/(RA6*CCB) ...
       + 1/(RA6*CA6) + 1/(RT6*CA6)
```

```
 DB6 = 1/(RA6*CA6*CCB*(RCB+RWB)) ...
       + 1/(RT6*CA6*CCB*(RCB+RWB)) ...
       + 1/(RA6*RT6*CA6*CCB)
RTIME = T                                     $" Reset Elapsed Time "
700.. CONTINUE                                $" End of Conditional "
END                                           $" End of Procedural "

TERMT(T.GE.TSTP)                              $" Stop Instruction "
END                                           $" Of Dynamic "

END                                           $" End of Program "
```

```
PROGRAM CONTROL2 1.0

" This program uses a second order approximation to model "
" a group of furnaces. "
" Then it adjusts the maximum number of furnaces periodically "
" using the Floating Maximum Algorithm. "

CONSTANT  CCB = 104.5          $" Crucible Capacitance "
CONSTANT  CCS = 71.81          $" Crucible Capacitance "
CONSTANT  DELTA = 10.0         $" Thermostat Deadband "
CONSTANT  HEAT1 = 43.6         $" Heat element (BTU/sec) "
CONSTANT  HEAT2 = 43.6         $" Heat element (BTU/sec) "
CONSTANT  HEAT3 = 43.6         $" Heat element (BTU/sec) "
CONSTANT  HEAT4 = 43.6         $" Heat element (BTU/sec) "
CONSTANT  HEAT5 = 43.6         $" Heat element (BTU/sec) "
CONSTANT  HEAT6 = 43.6         $" Heat element (BTU/sec) "
CONSTANT  HISET1 = 1290.0      $" High setpoint (above 110 F) "
CONSTANT  HISET2 = 1290.0      $" High setpoint (above 110 F) "
CONSTANT  HISET3 = 1290.0      $" High setpoint (above 110 F) "
CONSTANT  HISET4 = 1290.0      $" High setpoint (above 110 F) "
CONSTANT  HISET5 = 1290.0      $" High setpoint (above 110 F) "
CONSTANT  HISET6 = 1290.0      $" High setpoint (above 110 F) "
CONSTANT  NMAX = 3.0           $" Maximum furnaces on "
CONSTANT  PTCALC = 600.0       $" Check PMAX every 10 min "
CONSTANT  RCB = 3.05           $" Big Crucible Resistance "
CONSTANT  RCS = 5.026          $" Small Crucible Resistance "
CONSTANT  RSEED = 99           $" Seed for Random Number "
CONSTANT  RT1 = 350.0          $" Top-Loss Resistance "
CONSTANT  RT2 = 757.0          $" Top-Loss Resistance "
CONSTANT  RT3 = 757.0          $" Top-Loss Resistance "
CONSTANT  RT4 = 757.0          $" Top-Loss Resistance "
CONSTANT  RT5 = 757.0          $" Top-Loss Resistance "
CONSTANT  RT6 = 350.0          $" Top-Loss Resistance "
CONSTANT  RWB = 525.5          $" Wall-Loss Resistance "
CONSTANT  RWS = 120.0          $" Wall-Loss Resistance "
CONSTANT  TSTP = 3600.0        $" Simulation Stop Time "
CONSTANT  TIMER = 120.0        $" Disturbance Rate "
CONSTANT  TZ = 0.0             $" Simulation Start Time "
CONSTANT  W1 = 600.0           $" Default Weight (lbs) "
CONSTANT  W2 = 400.0           $" Default Weight (lbs) "
CONSTANT  W3 = 400.0           $" Default Weight (lbs) "
CONSTANT  W4 = 400.0           $" Default Weight (lbs) "
CONSTANT  W5 = 400.0           $" Default Weight (lbs) "
CONSTANT  W6 = 600.0           $" Default Weight (lbs) "

INITIAL

CINTERVAL  CINT = 5.0          $" Communication Interval "
PT = 0.0                       $" Initialize Total Power "
PAV = 0.0                      $" Initialize Power Available "
```

```
PMAX=NMAX*43.6/0.9478          $" Initialize PMAX "
PTIME = 0.0                    $" Initialize PAV Timer Memory "
PTEMP = 0.0                    $" Temporary comparison variable "
PDIFF = 0.0                    $" Initialize PAV Timer "
PINT = 0.0                     $" Initialize PMAX - PT "
NLO = 0.5                      $" Minimum Furnace Power Available "
NHI = 1.5                      $" Maximum Furnace Power Available "
   KWHUSE = 0.0                $" Initialize Kilowatt-hours used "
   CM = 0.0                    $" Initialize KWH cost "
   CMN = 0.0                   $" Initialize Normalized KWH cost "
   CD = 0.0                    $" Initailize Demand cost "
   EMC = 0.0                   $" Initialize Effect Monthly cost "
   PEAK = 0.0                  $" Initialize Peak Power "
UNIFI(RSEED)                   $" Initialization of Random Number "
RDIFF = 0.0                    $" Initialize RDIFF "
RTIME = 0.0                    $" Initialize RTIME "
U1 = 0.0                       $" Initial heat input "
U2 = 0.0                       $" Initial heat input "
U3 = 0.0                       $" Initial heat input "
U4 = 0.0                       $" Initial heat input "
U5 = 0.0                       $" Initial heat input "
U6 = 0.0                       $" Initial heat input "
LOSET1 = HISET1 - DELTA        $" Low Setpoint "
LOSET2 = HISET2 - DELTA        $" Low Setpoint "
LOSET3 = HISET3 - DELTA        $" Low Setpoint "
LOSET4 = HISET4 - DELTA        $" Low Setpoint "
LOSET5 = HISET5 - DELTA        $" Low Setpoint "
LOSET6 = HISET6 - DELTA        $" Low Setpoint "
TC1 = HISET1 - DELTA/2         $" Thermocouple Initialize "
TC2 = HISET2 - DELTA/2         $" Thermocouple Initialize "
TC3 = HISET3 - DELTA/2         $" Thermocouple Initialize "
TC4 = HISET4 - DELTA/2         $" Thermocouple Initialize "
TC5 = HISET5 - DELTA/2         $" Thermocouple Initialize "
TC6 = HISET6 - DELTA/2         $" Thermocouple Initialize "
TOLD1 = 0.0                    $" Initialize Memory "
TOLD2 = 0.0                    $" Initialize Memory "
TOLD3 = 0.0                    $" Initialize Memory "
TOLD4 = 0.0                    $" Initialize Memory "
TOLD5 = 0.0                    $" Initialize Memory "
TOLD6 = 0.0                    $" Initialize Memory "
CA1 = 0.2727*W1                $" Aluminum Capacitance "
CA2 = 0.2727*W2                $" Aluminum Capacitance "
CA3 = 0.2727*W3                $" Aluminum Capacitance "
CA4 = 0.2727*W4                $" Aluminum Capacitance "
CA5 = 0.2727*W5                $" Aluminum Capacitance "
CA6 = 0.2727*W6                $" Aluminum Capacitance "
RA1 = 8987.0/W1                $" Initialize Aluminum Res. "
RA2 = 3994.0/W2                $" Initialize Aluminum Res. "
RA3 = 3994.0/W3                $" Initialize Aluminum Res. "
RA4 = 3994.0/W4                $" Initialize Aluminum Res. "
```

```
RA5 = 3994.0/W5               $" Initialize Aluminum Res. "
RA6 = 8987.0/W6               $" Initialize Aluminum Res. "
TA01 = TC1/(1 + RA1/RT1)      $" Initialize Aluminum Temp "
TA02 = TC2/(1 + RA2/RT2)      $" Initialize Aluminum Temp "
TA03 = TC3/(1 + RA3/RT3)      $" Initialize Aluminum Temp "
TA04 = TC4/(1 + RA4/RT4)      $" Initialize Aluminum Temp "
TA05 = TC5/(1 + RA5/RT5)      $" Initialize Aluminum Temp "
TA06 = TC6/(1 + RA6/RT6)      $" Initialize Aluminum Temp "

" Numerator and Denominator Coefficients: "

NUM1 = RWB/(RA1*CA1*(RCB+RWB)*CCB)
DA1 = 1/((RCB+RWB)*CCB) + 1/(RA1*CCB) ...
      + 1/(RA1*CA1) + 1/(RT1*CA1)
DB1 = 1/(RA1*CA1*CCB*(RCB+RWB)) ...
      + 1/(RT1*CA1*CCB*(RCB+RWB)) ...
      + 1/(RA1*RT1*CA1*CCB)
NUM2 = RWS/(RA2*CA2*(RCS+RWS)*CCS)
DA2 = 1/((RCS+RWS)*CCS) + 1/(RA2*CCS) ...
      + 1/(RA2*CA2) + 1/(RT2*CA2)
DB2 = 1/(RA2*CA2*CCS*(RCS+RWS)) ...
      + 1/(RT2*CA2*CCS*(RCS+RWS)) ...
      + 1/(RA2*RT2*CA2*CCS)
NUM3 = RWS/(RA3*CA3*(RCS+RWS)*CCS)
DA3 = 1/((RCS+RWS)*CCS) + 1/(RA3*CCS) ...
      + 1/(RA3*CA3) + 1/(RT3*CA3)
DB3 = 1/(RA3*CA3*CCS*(RCS+RWS)) ...
      + 1/(RT3*CA3*CCS*(RCS+RWS)) ...
      + 1/(RA3*RT3*CA3*CCS)
NUM4 = RWS/(RA4*CA4*(RCS+RWS)*CCS)
DA4 = 1/((RCS+RWS)*CCS) + 1/(RA4*CCS) ...
      + 1/(RA4*CA4) + 1/(RT4*CA4)
DB4 = 1/(RA4*CA4*CCS*(RCS+RWS)) ...
      + 1/(RT4*CA4*CCS*(RCS+RWS)) ...
      + 1/(RA4*RT4*CA4*CCS)
NUM5 = RWS/(RA5*CA5*(RCS+RWS)*CCS)
DA5 = 1/((RCS+RWS)*CCS) + 1/(RA5*CCS) ...
      + 1/(RA5*CA5) + 1/(RT5*CA5)
DB5 = 1/(RA5*CA5*CCS*(RCS+RWS)) ...
      + 1/(RT5*CA5*CCS*(RCS+RWS)) ...
      + 1/(RA5*RT5*CA5*CCS)
NUM6 = RWB/(RA6*CA6*(RCB+RWB)*CCB)
DA6 = 1/((RCB+RWB)*CCB) + 1/(RA6*CCB) ...
      + 1/(RA6*CA6) + 1/(RT6*CA6)
DB6 = 1/(RA6*CA6*CCB*(RCB+RWB)) ...
      + 1/(RT6*CA6*CCB*(RCB+RWB)) ...
      + 1/(RA6*RT6*CA6*CCB)

END                           $" Of Initial Section "
```

```
DYNAMIC

SST1 = NUM1*U1 - DA1*ST1 - DB1*TA1      $" Second Derivative "
SST2 = NUM2*U2 - DA2*ST2 - DB2*TA2      $" Second Derivative "
SST3 = NUM3*U3 - DA3*ST3 - DB3*TA3      $" Second Derivative "
SST4 = NUM4*U4 - DA4*ST4 - DB4*TA4      $" Second Derivative "
SST5 = NUM5*U5 - DA5*ST5 - DB5*TA5      $" Second Derivative "
SST6 = NUM6*U6 - DA6*ST6 - DB6*TA6      $" Second Derivative "

DERIVATIVE

ST1 = INTEG(SST1,0.0)                   $" First Derivative "
ST2 = INTEG(SST2,0.0)                   $" First Derivative "
ST3 = INTEG(SST3,0.0)                   $" First Derivative "
ST4 = INTEG(SST4,0.0)                   $" First Derivative "
ST5 = INTEG(SST5,0.0)                   $" First Derivative "
ST6 = INTEG(SST6,0.0)                   $" First Derivative "
TA1 = INTEG(ST1,TA01)                   $" Aluminum Temperature "
TA2 = INTEG(ST2,TA02)                   $" Aluminum Temperature "
TA3 = INTEG(ST3,TA03)                   $" Aluminum Temperature "
TA4 = INTEG(ST4,TA04)                   $" Aluminum Temperature "
TA5 = INTEG(ST5,TA05)                   $" Aluminum Temperature "
TA6 = INTEG(ST6,TA06)                   $" Aluminum Temperature "
KWHUSE = INTEG(PT,0.0)/3600.0           $" Kilowatt Hours "
PAV = INTEG(PINT,0.0)                   $" Power Available "
END                                     $" Of Derivative Section "

TC1 = (1 + RA1/RT1)*TA1 + (RA1*CA1)*ST1 $" Crucible Temperature "
TC2 = (1 + RA2/RT2)*TA2 + (RA2*CA2)*ST2 $" Crucible Temperature "
TC3 = (1 + RA3/RT3)*TA3 + (RA3*CA3)*ST3 $" Crucible Temperature "
TC4 = (1 + RA4/RT4)*TA4 + (RA4*CA4)*ST4 $" Crucible Temperature "
TC5 = (1 + RA5/RT5)*TA5 + (RA5*CA5)*ST5 $" Crucible Temperature "
TC6 = (1 + RA6/RT6)*TA6 + (RA6*CA6)*ST6 $" Crucible Temperature "

PROCEDURAL     $" This Section for 6 Independent Thermostats "

  IF (TC1 .LT. TOLD1) SET1 = LOSET1     $" Falling Temperature "
  IF (TC1 .GE. TOLD1) SET1 = HISET1     $" Rising Temperature "
  IF ((TC1.LT.SET1).and.(pt.lt.pmax)) U1 = HEAT1    $" Heat On "
  IF (TC1 .GE. SET1) U1 = 0.0           $" Heat Off "
PT = (U1+U2+U3+U4+U5+U6)/0.9478         $" Total Power in KW "
  TOLD1 = TC1
  IF (TC2 .LT. TOLD2) SET2 = LOSET2     $" Falling Temperature "
  IF (TC2 .GE. TOLD2) SET2 = HISET2     $" Rising Temperature "
  IF ((TC2.LT.SET2).and.(pt.lt.pmax))  U2 = HEAT2   $" Heat On "
  IF (TC2 .GE. SET2) U2 = 0.0           $" Heat Off "
PT = (U1+U2+U3+U4+U5+U6)/0.9478         $" Total Power in KW "
  TOLD2 = TC2
  IF (TC3 .LT. TOLD3) SET3 = LOSET3     $" Falling Temperature "
  IF (TC3 .GE. TOLD3) SET3 = HISET3     $" Rising Temperature "
  IF ((TC3.LT.SET3).and.(pt.lt.pmax))  U3 = HEAT3   $" Heat On "
  IF (TC3 .GE. SET3) U3 = 0.0           $" Heat Off "
```

```
      PT = (U1+U2+U3+U4+U5+U6)/0.9478          $" Total Power in KW "
        TOLD3 = TC3
        IF (TC4 .LT. TOLD4) SET4 = LOSET4      $" Falling Temperature "
        IF (TC4 .GE. TOLD4) SET4 = HISET4      $" Rising Temperature "
        IF ((TC4.LT.SET4).and.(pt.lt.pmax))  U4 = HEAT4    $" Heat On "
        IF (TC4 .GE. SET4) U4 = 0.0            $" Heat Off "
      PT = (U1+U2+U3+U4+U5+U6)/0.9478          $" Total Power in KW "
        TOLD4 = TC4
        IF (TC5 .LT. TOLD5) SET5 = LOSET5      $" Falling Temperature "
        IF (TC5 .GE. TOLD5) SET5 = HISET5      $" Rising Temperature "
        IF ((TC5.LT.SET5).and.(pt.lt.pmax))  U5 = HEAT5    $" Heat On "
        IF (TC5 .GE. SET5) U5 = 0.0            $" Heat Off "
      PT = (U1+U2+U3+U4+U5+U6)/0.9478          $" Total Power in KW "
        TOLD5 = TC5
        IF (TC6 .LT. TOLD6) SET6 = LOSET6      $" Falling Temperature "
        IF (TC6 .GE. TOLD6) SET6 = HISET6      $" Rising Temperature "
        IF ((TC6.LT.SET6).and.(pt.lt.pmax))  U6 = HEAT6    $" Heat On "
        IF (TC6 .GE. SET6) U6 = 0.0            $" Heat Off "
        TOLD6 = TC6
      PT = (U1+U2+U3+U4+U5+U6)/0.9478          $" Total Power in KW "

" Calculate Power Available and Adjust PMAX every PTCALC seconds "
      PINT = PMAX - PT                         $" Calculate PAV integrand "
      PDIFF = T - PTIME                        $" Update Timer "
      IF(PDIFF .LT. PTCALC) GO TO 50           $" Chk time to recalculate "
      PAV = (1.0/PDIFF) * PAV                  $" Scale PAV elapsed time "
      PTEMP = NLO*(43.6/0.9478)
      IF(PAV .LT. PTEMP) PMAX = PMAX + (43.6/0.9478)
      PTEMP = NHI*(43.6/0.9478)
      IF(PAV .GT. PTEMP) PMAX = PMAX - (43.6/0.9478)
      PAV = 0.0                                $" Reset Power Available "
      PTIME = T                                $" Reset Timer "
      50.. CONTINUE
" Calculate Effective Monthly Power Cost "
      IF(PT.GT.PEAK) PEAK = PT                 $" Calculate Peak Demand "
      CM = KWHUSE*0.06278 + ( (KWHUSE-15000.0)*0.03034 ...
           - (KWHUSE-15000.0)*0.06278 )*STEP(15000.0-KWHUSE)
      IF(T.GT.0.0) CMN = (2592000.0/t)*CM
      CD = 9.50*(peak-50.0)
      EMC = 8.80 + CMN + CD                    $" Effect. Monthly Cost "

      RDIFF = T - RTIME                        $" Elapsed Time Counter "
      IF(RDIFF .LT. TIMER) GO TO 700           $" Adj Every (TIMER) Sec "
      RAND = UNIF(0.0,6.0)                     $" Random # Selects Furn. "
      RANDB = UNIF(200.0,800.0)                $" Random Number, 200-800 "
      RANDS = UNIF(100.0,500.0)                $" Random Number, 100-500 "
      IF (RAND .LT. 1.0) GO TO 100
      IF ((RAND .GE. 1.0).AND.(RAND .LT. 2.0)) GO TO 200
      IF ((RAND .GE. 2.0).AND.(RAND .LT. 3.0)) GO TO 300
```

```
IF ((RAND .GE. 3.0).AND.(RAND .LT. 4.0)) GO TO 400
IF ((RAND .GE. 4.0).AND.(RAND .LT. 5.0)) GO TO 500
IF ((RAND .GE. 5.0).AND.(RAND .LT. 6.0)) GO TO 600
100.. W1 = RANDB                            $" Random Weight, 200-800 "
 RA1 = 1997.0/W1                            $" Aluminum Resistance "
 CA1 = 0.2727*W1                            $" Aluminum Capacitance "
 NUM1 = RWB/(RA1*CA1*(RCB+RWB)*CCB)         $" Numerator "
 DA1 = 1/((RCB+RWB)*CCB) + 1/(RA1*CCB) ...
       + 1/(RA1*CA1) + 1/(RT1*CA1)
 DB1 = 1/(RA1*CA1*CCB*(RCB+RWB)) ...
       + 1/(RT1*CA1*CCB*(RCB+RWB)) ...
       + 1/(RA1*RT1*CA1*CCB)
RTIME = T                                   $" Reset Elapsed Time "
GO TO 700
200.. W2 = RANDS                            $" Random Weight, 100-500 "
 RA2 = 1997.0/W2                            $" Aluminum Resistance "
 CA2 = 0.2727*W2                            $" Aluminum Capacitance "
 NUM2 = RWS/(RA2*CA2*(RCS+RWS)*CCS)         $" Numerator "
 DA2 = 1/((RCS+RWS)*CCS) + 1/(RA2*CCS) ...
       + 1/(RA2*CA2) + 1/(RT2*CA2)
 DB2 = 1/(RA2*CA2*CCS*(RCS+RWS)) ...
       + 1/(RT2*CA2*CCS*(RCS+RWS)) ...
       + 1/(RA2*RT2*CA2*CCS)
RTIME = T                                   $" Reset Elapsed Time "
GO TO 700
300.. W3 = RANDS                            $" Random Weight, 100-500 "
 RA3 = 1997.0/W3                            $" Aluminum Resistance "
 CA3 = 0.2727*W3                            $" Aluminum Capacitance "
 NUM3 = RWS/(RA3*CA3*(RCS+RWS)*CCS)         $" Numerator "
 DA3 = 1/((RCS+RWS)*CCS) + 1/(RA3*CCS) ...
       + 1/(RA3*CA3) + 1/(RT3*CA3)
 DB3 = 1/(RA3*CA3*CCS*(RCS+RWS)) ...
       + 1/(RT3*CA3*CCS*(RCS+RWS)) ...
       + 1/(RA3*RT3*CA3*CCS)
RTIME = T                                   $" Reset Elapsed Time "
GO TO 700
400.. W4 = RANDS                            $" Random Weight, 100-500 "
 RA4 = 1997.0/W4                            $" Aluminum Resistance "
 CA4 = 0.2727*W4                            $" Aluminum Capacitance "
 NUM4 = RWS/(RA4*CA4*(RCS+RWS)*CCS)         $" Numerator "
 DA4 = 1/((RCS+RWS)*CCS) + 1/(RA4*CCS) ...
       + 1/(RA4*CA4) + 1/(RT4*CA4)
 DB4 = 1/(RA4*CA4*CCS*(RCS+RWS)) ...
       + 1/(RT4*CA4*CCS*(RCS+RWS)) ...
       + 1/(RA4*RT4*CA4*CCS)
RTIME = T                                   $" Reset Elapsed Time "
GO TO 700
500.. W5 = RANDS                            $" Random Weight, 100-500 "
 RA5 = 1997.0/W5                            $" Aluminum Resistance "
 CA5 = 0.2727*W5                            $" Aluminum Capacitance "
```

```
 NUM5 = RWS/(RA5*CA5*(RCS+RWS)*CCS)      $" Numerator "
 DA5 = 1/((RCS+RWS)*CCS) + 1/(RA5*CCS) ...
       + 1/(RA5*CA5) + 1/(RT5*CA5)
 DB5 = 1/(RA5*CA5*CCS*(RCS+RWS)) ...
       + 1/(RT5*CA5*CCS*(RCS+RWS)) ...
       + 1/(RA5*RT5*CA5*CCS)
RTIME = T                                $" Reset Elapsed Time "
GO TO 700
600.. W6 = RANDB                         $" Random Weight, 200-800 "
 RA6 = 1997.0/W6                         $" Aluminum Resistance "
 CA6 = 0.2727*W6                         $" Aluminum Capacitance "
 NUM6 = RWB/(RA6*CA6*(RCB+RWB)*CCB)      $" Numerator "
 DA6 = 1/((RCB+RWB)*CCB) + 1/(RA6*CCB) ...
       + 1/(RA6*CA6) + 1/(RT6*CA6)
 DB6 = 1/(RA6*CA6*CCB*(RCB+RWB)) ...
       + 1/(RT6*CA6*CCB*(RCB+RWB)) ...
       + 1/(RA6*RT6*CA6*CCB)
RTIME = T                                $" Reset Elapsed Time "
700.. CONTINUE                           $" End of Conditional "
   END                                   $" End of Procedural "

TERMT(T.GE.TSTP)                         $" Stop Instruction "
END                                      $" Of Dynamic "

END                                      $" End of Program "
```

```
PROGRAM CONTROL3 1.0

" This program uses a second order approximation to model "
" a group of furnaces.                                   "
" Then it adjusts the maximum number of furnaces periodically "
" using the Buddy System Algorithm.                      "

CONSTANT  CCB = 104.5          $" Crucible Capacitance "
CONSTANT  CCS = 71.81          $" Crucible Capacitance "
CONSTANT  DELTA = 10.0         $" Thermostat Deadband "
CONSTANT  HEAT1 = 43.6         $" Heat element (BTU/sec) "
CONSTANT  HEAT2 = 43.6         $" Heat element (BTU/sec) "
CONSTANT  HEAT3 = 43.6         $" Heat element (BTU/sec) "
CONSTANT  HEAT4 = 43.6         $" Heat element (BTU/sec) "
CONSTANT  HEAT5 = 43.6         $" Heat element (BTU/sec) "
CONSTANT  HEAT6 = 43.6         $" Heat element (BTU/sec) "
CONSTANT  HISET1 = 1290.0      $" High setpoint (above 110 F) "
CONSTANT  HISET2 = 1290.0      $" High setpoint (above 110 F) "
CONSTANT  HISET3 = 1290.0      $" High setpoint (above 110 F) "
CONSTANT  HISET4 = 1290.0      $" High setpoint (above 110 F) "
CONSTANT  HISET5 = 1290.0      $" High setpoint (above 110 F) "
CONSTANT  HISET6 = 1290.0      $" High setpoint (above 110 F) "
CONSTANT  NMAX = 3.0           $" Maximum furnaces on "
CONSTANT  PTCALC = 600.0       $" Check PMAX every 10 min "
CONSTANT  RCB = 3.05           $" Big Crucible Resistance "
CONSTANT  RCS = 5.026          $" Small Crucible Resistance "
CONSTANT  RSEED = 99           $" Seed for Random Number "
CONSTANT  RT1 = 350.0          $" Top-Loss Resistance "
CONSTANT  RT2 = 757.0          $" Top-Loss Resistance "
CONSTANT  RT3 = 757.0          $" Top-Loss Resistance "
CONSTANT  RT4 = 757.0          $" Top-Loss Resistance "
CONSTANT  RT5 = 757.0          $" Top-Loss Resistance "
CONSTANT  RT6 = 350.0          $" Top-Loss Resistance "
CONSTANT  RWB = 525.5          $" Wall-Loss Resistance "
CONSTANT  RWS = 120.0          $" Wall-Loss Resistance "
CONSTANT  TSTP = 3600.0        $" Simulation Stop Time "
CONSTANT  TIMER = 120.0        $" Disturbance Rate "
CONSTANT  TZ = 0.0             $" Simulation Start Time "
CONSTANT  W1 = 600.0           $" Default Weight (lbs) "
CONSTANT  W2 = 400.0           $" Default Weight (lbs) "
CONSTANT  W3 = 400.0           $" Default Weight (lbs) "
CONSTANT  W4 = 400.0           $" Default Weight (lbs) "
CONSTANT  W5 = 400.0           $" Default Weight (lbs) "
CONSTANT  W6 = 600.0           $" Default Weight (lbs) "

INITIAL

CINTERVAL  CINT = 5.0          $" Communication Interval "
PT = 0.0                       $" Initialize Total Power "
PAV = 0.0                      $" Initialize Power Available "
PMAX=NMAX*43.6/0.9478          $" Initialize PMAX "
PTIME = 0.0                    $" Initialize PAV Timer Memory "
```

```
PTEMP = 0.0                      $" Temporary comparison variable "
PDIFF = 0.0                      $" Initialize PAV Timer "
PINT = 0.0                       $" Initialize PMAX - PT "
NLO = 0.5                        $" Minimum Furnace Power Available "
NHI = 1.5                        $" Maximum Furnace Power Available "
   KWHUSE = 0.0                  $" Initialize Kilowatt-hours used "
   CM = 0.0                      $" Initialize KWH cost "
   CMN = 0.0                     $" Initialize Normalized KWH cost "
   CD = 0.0                      $" Initailize Demand cost "
   EMC = 0.0                     $" Initialize Effect Monthly cost "
   PEAK = 0.0                    $" Initialize Peak Power "
UNIFI(RSEED)                     $" Initialization of Random # "
RDIFF = 0.0                      $" Initialize RDIFF "
RTIME = 0.0                      $" Initialize RTIME "
U1 = 0.0                         $" Initial heat input "
U2 = 0.0                         $" Initial heat input "
U3 = 0.0                         $" Initial heat input "
U4 = 0.0                         $" Initial heat input "
U5 = 0.0                         $" Initial heat input "
U6 = 0.0                         $" Initial heat input "
LOSET1 = HISET1 - DELTA          $" Low Setpoint "
LOSET2 = HISET2 - DELTA          $" Low Setpoint "
LOSET3 = HISET3 - DELTA          $" Low Setpoint "
LOSET4 = HISET4 - DELTA          $" Low Setpoint "
LOSET5 = HISET5 - DELTA          $" Low Setpoint "
LOSET6 = HISET6 - DELTA          $" Low Setpoint "
TC1 = HISET1 - DELTA/2           $" Thermocouple Initialize "
TC2 = HISET2 - DELTA/2           $" Thermocouple Initialize "
TC3 = HISET3 - DELTA/2           $" Thermocouple Initialize "
TC4 = HISET4 - DELTA/2           $" Thermocouple Initialize "
TC5 = HISET5 - DELTA/2           $" Thermocouple Initialize "
TC6 = HISET6 - DELTA/2           $" Thermocouple Initialize "
TOLD1 = 0.0                      $" Initialize Memory "
TOLD2 = 0.0                      $" Initialize Memory "
TOLD3 = 0.0                      $" Initialize Memory "
TOLD4 = 0.0                      $" Initialize Memory "
TOLD5 = 0.0                      $" Initialize Memory "
TOLD6 = 0.0                      $" Initialize Memory "
CA1 = 0.2727*W1                  $" Aluminum Capacitance "
CA2 = 0.2727*W2                  $" Aluminum Capacitance "
CA3 = 0.2727*W3                  $" Aluminum Capacitance "
CA4 = 0.2727*W4                  $" Aluminum Capacitance "
CA5 = 0.2727*W5                  $" Aluminum Capacitance "
CA6 = 0.2727*W6                  $" Aluminum Capacitance "
RA1 = 8987.0/W1                  $" Initialize Aluminum Res. "
RA2 = 3994.0/W2                  $" Initialize Aluminum Res. "
RA3 = 3994.0/W3                  $" Initialize Aluminum Res. "
RA4 = 3994.0/W4                  $" Initialize Aluminum Res. "
RA5 = 3994.0/W5                  $" Initialize Aluminum Res. "
RA6 = 8987.0/W6                  $" Initialize Aluminum Res. "
TA01 = TC1/(1 + RA1/RT1)         $" Initialize Aluminum Temp "
TA02 = TC2/(1 + RA2/RT2)         $" Initialize Aluminum Temp "
```

```
TA03 = TC3/(1 + RA3/RT3)        $" Initialize Aluminum Temp "
TA04 = TC4/(1 + RA4/RT4)        $" Initialize Aluminum Temp "
TA05 = TC5/(1 + RA5/RT5)        $" Initialize Aluminum Temp "
TA06 = TC6/(1 + RA6/RT6)        $" Initialize Aluminum Temp "

" Numerator and Denominator Coefficients: "

NUM1 = RWB/(RA1*CA1*(RCB+RWB)*CCB)
DA1 = 1/((RCB+RWB)*CCB) + 1/(RA1*CCB) ...
      + 1/(RA1*CA1) + 1/(RT1*CA1)
DB1 = 1/(RA1*CA1*CCB*(RCB+RWB)) ...
      + 1/(RT1*CA1*CCB*(RCB+RWB)) ...
      + 1/(RA1*RT1*CA1*CCB)
NUM2 = RWS/(RA2*CA2*(RCS+RWS)*CCS)
DA2 = 1/((RCS+RWS)*CCS) + 1/(RA2*CCS) ...
      + 1/(RA2*CA2) + 1/(RT2*CA2)
DB2 = 1/(RA2*CA2*CCS*(RCS+RWS)) ...
      + 1/(RT2*CA2*CCS*(RCS+RWS)) ...
      + 1/(RA2*RT2*CA2*CCS)
NUM3 = RWS/(RA3*CA3*(RCS+RWS)*CCS)
DA3 = 1/((RCS+RWS)*CCS) + 1/(RA3*CCS) ...
      + 1/(RA3*CA3) + 1/(RT3*CA3)
DB3 = 1/(RA3*CA3*CCS*(RCS+RWS)) ...
      + 1/(RT3*CA3*CCS*(RCS+RWS)) ...
      + 1/(RA3*RT3*CA3*CCS)
NUM4 = RWS/(RA4*CA4*(RCS+RWS)*CCS)
DA4 = 1/((RCS+RWS)*CCS) + 1/(RA4*CCS) ...
      + 1/(RA4*CA4) + 1/(RT4*CA4)
DB4 = 1/(RA4*CA4*CCS*(RCS+RWS)) ...
      + 1/(RT4*CA4*CCS*(RCS+RWS)) ...
      + 1/(RA4*RT4*CA4*CCS)
NUM5 = RWS/(RA5*CA5*(RCS+RWS)*CCS)
DA5 = 1/((RCS+RWS)*CCS) + 1/(RA5*CCS) ...
      + 1/(RA5*CA5) + 1/(RT5*CA5)
DB5 = 1/(RA5*CA5*CCS*(RCS+RWS)) ...
      + 1/(RT5*CA5*CCS*(RCS+RWS)) ...
      + 1/(RA5*RT5*CA5*CCS)
NUM6 = RWB/(RA6*CA6*(RCB+RWB)*CCB)
DA6 = 1/((RCB+RWB)*CCB) + 1/(RA6*CCB) ...
      + 1/(RA6*CA6) + 1/(RT6*CA6)
DB6 = 1/(RA6*CA6*CCB*(RCB+RWB)) ...
      + 1/(RT6*CA6*CCB*(RCB+RWB)) ...
      + 1/(RA6*RT6*CA6*CCB)

END                                       $" Of Initial Section "

DYNAMIC

SST1 = NUM1*U1 - DA1*ST1 - DB1*TA1      $" Second Derivative "
SST2 = NUM2*U2 - DA2*ST2 - DB2*TA2      $" Second Derivative "
```

```
SST3 = NUM3*U3 - DA3*ST3 - DB3*TA3      $" Second Derivative "
SST4 = NUM4*U4 - DA4*ST4 - DB4*TA4      $" Second Derivative "
SST5 = NUM5*U5 - DA5*ST5 - DB5*TA5      $" Second Derivative "
SST6 = NUM6*U6 - DA6*ST6 - DB6*TA6      $" Second Derivative "

DERIVATIVE
ST1 = INTEG(SST1,0.0)                   $" First Derivative "
ST2 = INTEG(SST2,0.0)                   $" First Derivative "
ST3 = INTEG(SST3,0.0)                   $" First Derivative "
ST4 = INTEG(SST4,0.0)                   $" First Derivative "
ST5 = INTEG(SST5,0.0)                   $" First Derivative "
ST6 = INTEG(SST6,0.0)                   $" First Derivative "
TA1 = INTEG(ST1,TA01)                   $" Aluminum Temperature "
TA2 = INTEG(ST2,TA02)                   $" Aluminum Temperature "
TA3 = INTEG(ST3,TA03)                   $" Aluminum Temperature "
TA4 = INTEG(ST4,TA04)                   $" Aluminum Temperature "
TA5 = INTEG(ST5,TA05)                   $" Aluminum Temperature "
TA6 = INTEG(ST6,TA06)                   $" Aluminum Temperature "
KWHUSE = INTEG(PT,0.0)/3600.0           $" Kilowatt Hours "
PAV = INTEG(PINT,0.0)                   $" Power Available "
END                                     $" Of Derivative Section "

TC1 = (1 + RA1/RT1)*TA1 + (RA1*CA1)*ST1 $" Crubicle Temperature "
TC2 = (1 + RA2/RT2)*TA2 + (RA2*CA2)*ST2 $" Crubicle Temperature "
TC3 = (1 + RA3/RT3)*TA3 + (RA3*CA3)*ST3 $" Crubicle Temperature "
TC4 = (1 + RA4/RT4)*TA4 + (RA4*CA4)*ST4 $" Crubicle Temperature "
TC5 = (1 + RA5/RT5)*TA5 + (RA5*CA5)*ST5 $" Crubicle Temperature "
TC6 = (1 + RA6/RT6)*TA6 + (RA6*CA6)*ST6 $" Crubicle Temperature "

   PROCEDURAL  $" This Section for 6 Independent Thermostats "

IF (TC1 .LT. TOLD1) SET1 = LOSET1       $" Falling Temperature "
IF (TC1 .GE. TOLD1) SET1 = HISET1       $" Rising Temperature "
IF ((TC1.LT.SET1).and.(u2.eq.0.0))  U1 = HEAT1    $" Heat On "
IF (TC1 .GE. SET1) U1 = 0.0             $" Heat Off "
TOLD1 = TC1
IF (TC2 .LT. TOLD2) SET2 = LOSET2       $" Falling Temperature "
IF (TC2 .GE. TOLD2) SET2 = HISET2       $" Rising Temperature "
IF ((TC2.LT.SET2).and.(u1.eq.0.0))  U2 = HEAT2    $" Heat On "
IF (TC2 .GE. SET2) U2 = 0.0             $" Heat Off "
TOLD2 = TC2
IF (TC3 .LT. TOLD3) SET3 = LOSET3       $" Falling Temperature "
IF (TC3 .GE. TOLD3) SET3 = HISET3       $" Rising Temperature "
IF ((TC3.LT.SET3).and.(u4.eq.0.0))  U3 = HEAT3    $" Heat On "
IF (TC3 .GE. SET3) U3 = 0.0             $" Heat Off "
TOLD3 = TC3
IF (TC4 .LT. TOLD4) SET4 = LOSET4       $" Falling Temperature "
IF (TC4 .GE. TOLD4) SET4 = HISET4       $" Rising Temperature "
IF ((TC4.LT.SET4).and.(u3.eq.0.0))  U4 = HEAT4    $" Heat On "
IF (TC4 .GE. SET4) U4 = 0.0             $" Heat Off "
TOLD4 = TC4
```

```
    IF (TC5 .LT. TOLD5) SET5 = LOSET5        $" Falling Temperature "
    IF (TC5 .GE. TOLD5) SET5 = HISET5        $" Rising Temperature "
    IF ((TC5.LT.SET5).and.(u6.eq.0.0))  U5 = HEAT5     $" Heat On "
    IF (TC5 .GE. SET5) U5 = 0.0              $" Heat Off "
    TOLD5 = TC5
    IF (TC6 .LT. TOLD6) SET6 = LOSET6        $" Falling Temperature "
    IF (TC6 .GE. TOLD6) SET6 = HISET6        $" Rising Temperature "
    IF ((TC6.LT.SET6).and.(u5.eq.0.0))  U6 = HEAT6     $" Heat On "
    IF (TC6 .GE. SET6) U6 = 0.0              $" Heat Off "
    TOLD6 = TC6
    pt = (u1+u2+u3+u4+u5+u6)/0.9478          $" Total Power in KW "

" Calculate Effective Monthly Power Cost "
    IF(PT.GT.PEAK) PEAK = PT                 $" Calculate peak demand "
    CM = KWHUSE*0.06278 + ( (KWHUSE-15000.0)*0.03034 ...
         - (KWHUSE-15000.0)*0.06278 )*STEP(15000.0-KWHUSE)
    IF(T.GT.0.0) CMN = (2592000.0/t)*CM
    CD = 9.50*(peak-50.0)
    EMC = 8.80 + CMN + CD                    $" Effect Monthly Cost "

    RDIFF = T - RTIME                        $" Elapsed Time Counter "
    IF(RDIFF .LT. TIMER) GO TO 700           $" Adj Every (TIMER) Sec "
    RAND = UNIF(0.0,6.0)                     $" Random # Selects Furn "
    RANDB = UNIF(200.0,800.0)                $" Random Number, 200-800 "
    RANDS = UNIF(100.0,500.0)                $" Random Number, 100-500 "
    IF (RAND .LT. 1.0) GO TO 100
    IF ((RAND .GE. 1.0).AND.(RAND .LT. 2.0)) GO TO 200
    IF ((RAND .GE. 2.0).AND.(RAND .LT. 3.0)) GO TO 300
    IF ((RAND .GE. 3.0).AND.(RAND .LT. 4.0)) GO TO 400
    IF ((RAND .GE. 4.0).AND.(RAND .LT. 5.0)) GO TO 500
    IF ((RAND .GE. 5.0).AND.(RAND .LT. 6.0)) GO TO 600
    100.. W1 = RANDB                         $" Random Weight, 200-800 "
     RA1 = 1997.0/W1                         $" Aluminum Resistance "
     CA1 = 0.2727*W1                         $" Aluminum Capacitance "
     NUM1 = RWB/(RA1*CA1*(RCB+RWB)*CCB)      $" Numerator "
     DA1 = 1/((RCB+RWB)*CCB) + 1/(RA1*CCB) ...
           + 1/(RA1*CA1) + 1/(RT1*CA1)
     DB1 = 1/(RA1*CA1*CCB*(RCB+RWB)) ...
           + 1/(RT1*CA1*CCB*(RCB+RWB)) ...
           + 1/(RA1*RT1*CA1*CCB)
    RTIME = T                                $" Reset Elapsed Time "
    GO TO 700
    200.. W2 = RANDS                         $" Random Weight, 100-500 "
     RA2 = 1997.0/W2                         $" Aluminum Resistance "
     CA2 = 0.2727*W2                         $" Aluminum Capacitance "
     NUM2 = RWS/(RA2*CA2*(RCS+RWS)*CCS)      $" Numerator "
     DA2 = 1/((RCS+RWS)*CCS) + 1/(RA2*CCS) ...
           + 1/(RA2*CA2) + 1/(RT2*CA2)
     DB2 = 1/(RA2*CA2*CCS*(RCS+RWS)) ...
           + 1/(RT2*CA2*CCS*(RCS+RWS)) ...
           + 1/(RA2*RT2*CA2*CCS)
```

```
RTIME = T                                   $" Reset Elapsed Time "
GO TO 700
300.. W3 = RANDS                            $" Random Weight, 100-500 "
 RA3 = 1997.0/W3                            $" Aluminum Resistance "
 CA3 = 0.2727*W3                            $" Aluminum Capacitance "
 NUM3 = RWS/(RA3*CA3*(RCS+RWS)*CCS)         $" Numerator "
 DA3 = 1/((RCS+RWS)*CCS) + 1/(RA3*CCS) ...
       + 1/(RA3*CA3) + 1/(RT3*CA3)
 DB3 = 1/(RA3*CA3*CCS*(RCS+RWS)) ...
       + 1/(RT3*CA3*CCS*(RCS+RWS)) ...
       + 1/(RA3*RT3*CA3*CCS)
RTIME = T                                   $" Reset Elapsed Time "
GO TO 700
400.. W4 = RANDS                            $" Random Weight, 100-500 "
 RA4 = 1997.0/W4                            $" Aluminum Resistance "
 CA4 = 0.2727*W4                            $" Aluminum Capacitance "
 NUM4 = RWS/(RA4*CA4*(RCS+RWS)*CCS)         $" Numerator "
 DA4 = 1/((RCS+RWS)*CCS) + 1/(RA4*CCS) ...
       + 1/(RA4*CA4) + 1/(RT4*CA4)
 DB4 = 1/(RA4*CA4*CCS*(RCS+RWS)) ...
       + 1/(RT4*CA4*CCS*(RCS+RWS)) ...
       + 1/(RA4*RT4*CA4*CCS)
RTIME = T                                   $" Reset Elapsed Time "
GO TO 700
500.. W5 = RANDS                            $" Random Weight, 100-500 "
 RA5 = 1997.0/W5                            $" Aluminum Resistance "
 CA5 = 0.2727*W5                            $" Aluminum Capacitance "
 NUM5 = RWS/(RA5*CA5*(RCS+RWS)*CCS)         $" Numerator "
 DA5 = 1/((RCS+RWS)*CCS) + 1/(RA5*CCS) ...
       + 1/(RA5*CA5) + 1/(RT5*CA5)
 DB5 = 1/(RA5*CA5*CCS*(RCS+RWS)) ...
       + 1/(RT5*CA5*CCS*(RCS+RWS)) ...
       + 1/(RA5*RT5*CA5*CCS)
RTIME = T                                   $" Reset Elapsed Time "
GO TO 700
600.. W6 = RANDB                            $" Random Weight, 200-800 "
 RA6 = 1997.0/W6                            $" Aluminum Resistance "
 CA6 = 0.2727*W6                            $" Aluminum Capacitance "
 NUM6 = RWB/(RA6*CA6*(RCB+RWB)*CCB)         $" Numerator "
 DA6 = 1/((RCB+RWB)*CCB) + 1/(RA6*CCB) ...
       + 1/(RA6*CA6) + 1/(RT6*CA6)
 DB6 = 1/(RA6*CA6*CCB*(RCB+RWB)) ...
       + 1/(RT6*CA6*CCB*(RCB+RWB)) ...
       + 1/(RA6*RT6*CA6*CCB)
RTIME = T                                   $" Reset Elapsed Time "
700.. CONTINUE                              $" End of Conditional "
   END                                      $" End of Procedural "

TERMT(T.GE.TSTP)                            $" Stop Instruction "
END                                         $" Of Dynamic "

END                                         $" End of Program "
```

VITA

Robert Wesley Wysor was born in Wilmington, Delaware, on September 25, 1955. He graduated from Lee High School in Huntsville, Alabama, in May of 1973. He then began six years of active duty in the United States Navy Nuclear Power Program. After spending the last four years aboard the USS Virginia (CGN 38), he received an honorable discharge as a First Class Electrician's Mate in June 1979.

He then entered the University of Tennessee in September 1979 where he studied Mechanical Engineering. After working two summers for IBM in Manassas, Virginia, he received a B.S.M.E. degree with High Honors in June of 1983. He worked for Pratt & Whitney in West Palm Beach, Florida, for over two years as a test engineer.

In September 1985, he returned to The University of Tennessee to begin work on a Master's Degree in Electrical Engineering. He began part-time work in August 1987 as a research assistant at the Oak Ridge National Laboratory in the Instrument and Controls Division. In December of that year, he was hired as a development engineer in that same division.