DOE/NASA/0042-79/3
NASA CR 159607
BCS 40262-1

SIMWEST: A SIMULATION MODEL FOR WIND AND PHOTOVOLTAIC ENERGY STORAGE SYSTEMS (CDC User's Manual)

Volume I

A. W. Warren
R. W. Edsinger
J. D. Burroughs
Energy Technology Applications Division
Boeing Computer Services Company
Seattle, Washington 98124

August 1979

Prepared for
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Lewis Research Center
Cleveland, Ohio 44135
Under Contract DEN3-42

For
U.S. DEPARTMENT OF ENERGY
Division of Energy Storage Systems
Washington, D.C. 20545
Under Interagency Agreement EX-76-A-31-1026

FOREWORD

This report documents the CDC version of the SIMWEST computer programs developed by Boeing Computer Services Company under NASA Contract DEN3-42, "An Expanded System Simulation Model for Solar Energy Storage". The SIMWEST codes were originally developed for simulation of wind energy storage systems. The current version of these codes also includes solar-photovoltaic energy systems modeling. This project was conducted under the sponsorship of the Division of Energy Storage Systems, DOE, under the direction of Dr. G. C. Chang, and was administered by the NASA-Lewis Research Center Thermal and Mechanical Storage Section with Mr. L. H. Gordon and Mr. R. H. Beach as Project Managers.

This report is in two volumes.

I. CDC User's Manual
II. CDC Program Descriptions

The Boeing principal investigator for this project was Dr. A. W. Warren. Major contributors in the development of SIMWEST were Dr. R. W. Edsinger, Dr. J. D. Burroughs, and Dr. Y. K. Chan.

TABLE OF CONTENTS

Page

CONTENTS (Continued)

CONTENTS (Continued)

CONTENTS (Continued)

LIST OF FIGURES

LIST OF FIGURES (Continued)

LIST OF FIGURES (Continued)

LIST OF TABLES

LIST OF REFERENCES

[1] R. W. Edsinger, Y. K. Chan, A. W. Warren, "A Simulation Model for Wind Energy Storage Systems," Volume I: Technical Report, NASA CR-135283, prepared by Boeing Computer Services Company, August 1977.

[2] A. W. Warren, R. W. Edsinger, J. D. Burroughs, "A Simulation Model for Wind Energy Storage Systems," Volume II: Operation Manual, NASA CR-135284, prepared by Boeing Computer Services Company, August 1977.

[3] SOLMET User's Manual, TD-9724, National Climatic Center, Ashville, North Carolina, December 1977.

[4] J. K. Linn, "Photovoltaic System Analysis Program--SOLCEL," Sandia Laboratories Report SAND77-1268, August 1977.

[5] AFFDL-TR-75-(D180-18575), "ECS Transient Analysis"; Vol. I - Component Mathematical Models; Vol. II - Component Transient Computer Programs; Vol. III - Integrated Transient Computer Program; and Vol. IV - Optimal Control and Design Synthesis; The Boeing Company, prepared under Air Force Contract F33615-74-C-3041, October 1975.

[6] A. W. Warren, "An Expanded System Simulation Model for Solar Energy Storage," Volume I: Technical Report, NASA CR-159601, prepared by Boeing Computer Services Company, August 1979.

1.0 INTRODUCTION

Energy storage systems for the utilization of intermittent power sources have received increased study over the past few years. The analysis of storage requirements for optimal utilization of solar-derived energy systems and the total cost of the resulting generator/storage system are often evaluated in such studies. The purpose of the SIMWEST (Simulation Model for Wind Energy Storage) program described in this document is to provide a tool for performing this needed analysis. It is a tool to aid in the design of a wind or solar-photovoltaic energy system for a given application and to allow the resulting system to be evaluated and verified through simulation.

SIMWEST consists of a library of system components and a precompiler program which allows these components to be put together in building block form. The present library contains components for five types of energy storage systems. They are pumped hydro, battery, thermal, flywheel, and pneumatic. The SIMWEST program version described in this document is for use on the CDC 6000 and CYBER series of computers.

The simulation program has proven to be efficient and versatile for performing parametric studies. It has a unique capability for simulating total wind/solar systems containing any one or combination of the above types of storage and at the same time has the flexibility and depth required to perform thorough and meaningful parameter studies.

1.1 SIMWEST OVERVIEW

SIMWEST consists of two basic programs, and a library of generation, storage, environmental, and load components. The first program, the Model Generation Program, is a precompiler which generates computer models (in FORTRAN) of complex energy generation/storage systems, from user specifications using SIMWEST library components. The second program utilizes the resulting computer model to perform cost and power utilization analysis. It handles input, output, integration of system dynamics, and iterates to obtain convergence of implicit variables. The combination of these two programs provides a powerful tool for analyzing alternate generation and storage system designs.

Figure 1.1-1 shows the general organization of the SIMWEST program. In addition to the two programs described above, there is a third which performs file maintenance. It is used to incorporate user supplied data for new subsystem models. Although the program is shown as a number of subprograms, it can be executed as a single batch program by supplying the model description cards and the control cards describing the desired analysis to be performed and the desired tabular and/or plotted output.

The SIMWEST model generation and simulation programs have a number of user oriented features which greatly enhance the value of the codes. Some of the more prominent features are shown in Table 1.1-1. These features and the supplemental components described in 1.2 enable the user to quickly build, debug, simulate and interpret alternative system designs.

1.2 SIMWEST LIBRARY

The SIMWEST library is listed in Table 1.2-1. It is made up of six types of components: environmental, generation, load, logical, storage and supplemental. The two character mnemonic names are used to identify components in the users model.

The degree of detail in the component models is based upon two design criteria. First, all models should contain sufficient detail to simulate all physical characteristics and constraints having significant impact on system cost effectiveness. Second, the models should be designed to minimize computer time and required user specification. It is assumed that a SIMWEST simulation might cover a time span of one year. Thus, from a computer run time and economic impact point of view a simulation step size of between 15 minutes and one hour was established as a design goal.

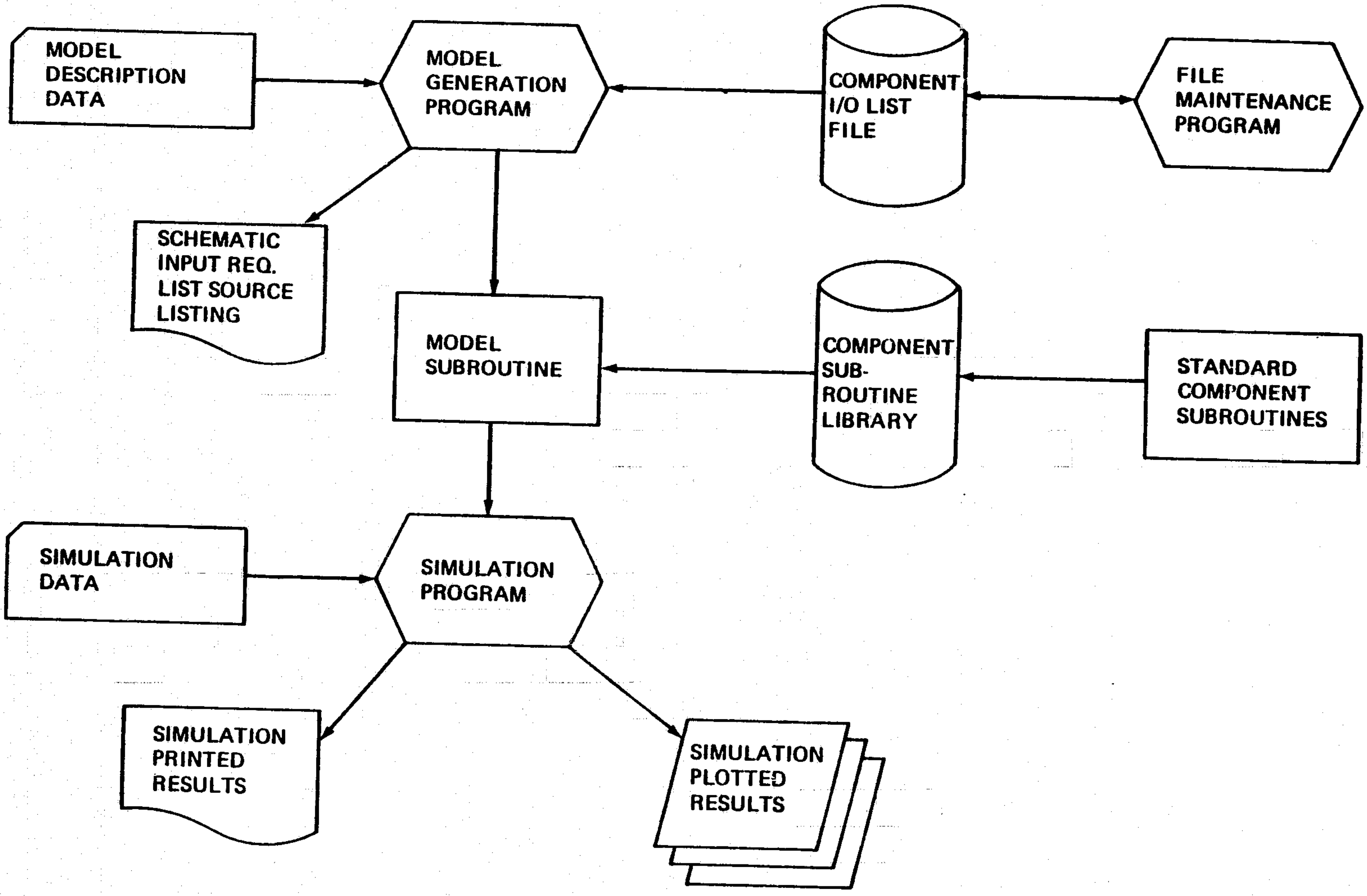

MODEL DESCRIPTION DATA
MODEL GENERATION PROGRAM
COMPONENT I/O LIST FILE
FILE MAINTENANCE PROGRAM
SCHEMATIC INPUT REQ. LIST SOURCE LISTING
MODEL SUBROUTINE
COMPONENT SUB-ROUTINE LIBRARY
STANDARD COMPONENT SUBROUTINES
SIMULATION DATA
SIMULATION PROGRAM
SIMULATION PRINTED RESULTS
SIMULATION PLOTTED RESULTS

Figure 1.1-1 SIMWEST Program Organization

Table 1.1-1 SIMWEST User Oriented Features

MODEL GENERATION PROGRAM

- Simplified Component Connections
- Availability of all Input Parameters for Connection
- Fortran Insertion Capability Between Components
- Line Printer Schematic of User's Model Provided
- Automated Naming of Parameters and Variables
- Built-in Diagnostic Capabilities

SIMULATION PROGRAM

- Free Field Data Inputs, Including Tables
- Diagnostics on Data Inputs
- Default Values Assigned to Unspecified Parameters
- Optional Levels of Line Printer and Diagnostic Output
- Multiple, Back-to-back Simulation Capability
- Printer Plotter Output of Time Histories and Crossplots

Table 1.2-1 SIMWEST Library Components

ENVIRONMENTAL	
WIND	WD
AMBIENT TEMP	TP
TMY WEATHER TAPE	ED
WIND POWER GENERATION	
TURBINE/GENERATOR	WP
WIND TURBINE	WT
FIXED RATIO TRANSMISSION	GR
AC GENERATOR	GE
SOLAR POWER GENERATION	
SOLAR ORIENTATION (TRACKING)	SO
FLAT PLATE COLLECTOR	FP
FOCUSING LENS COLLECTOR	FO
PHOTOVOLTAIC ARRAY	PV
UTILITY GENERATION	
UTILITY	UT
LOGIC	
POWER DIVIDER	PD
POWER ACCUMULATOR	PA
PRIORITY INTERRUPT	PI
SWITCHES	SW,SX SY,SZ
LOAD	
ELECTRICAL LOAD	LO
THERMAL LOAD	TL

BATTERY STORAGE	
INVERTER	IV
RECTIFIER	RE
BATTERY	BA
ADMITTANCE	AD
FLYWHEEL STORAGE	
AC MOTOR	MO
VARIABLE RATIO TRANSMISSION	TR
FLYWHEEL/CLUTCH	FL
HYDRO STORAGE	
HYDRO PUMP	PU
HYDRO TURBINE	HT
HYDRO STORAGE	HS
PNEUMATIC STORAGE	
COMPRESSOR	CO
TURBINE	TU
ADIABATIC HEAT EXCHANGER	HX,HY
BURNER	BN
PNEUMATIC STORAGE	CS
THERMAL STORAGE	
STORAGE VESSEL	TS
SUPPLEMENTAL	
SATURATION	SA
RANDOM NUMBER GENERATOR	RN
TEST FUNCTIONS	AF
TABLE LOOKUPS	FU,FV
TRANSFER FUNCTIONS	IT,LA,LL,TF
ARITHMETIC ELEMENTS	MA,MB,MC
COST MONITOR	CM
HISTOGRAM	HG
TIME CONVERSION	TI

As a result of the above design criteria, many physical components, such as the electrical components, were modeled mainly in terms of power flow and steady state response. This level of detail is consistent with a 15 minute time step and with the concept that important transients are on the time scale of demand curves or weather patterns, i.e., an hour or more, rather than on the time scale of electric motor transients of a few seconds. If short time transients were to be modeled, additional detail would be required in the component models which would greatly increase the user's task of specifying the model. Further, the simulation time step would have to be reduced and computer runs would be much costlier.

The environmental components listed in Table 1.2-1 simulate environmental conditions. In the present SIMWEST library a user can generate wind speed and ambient temperatures, or can use selected inputs from the recorded weather and insolation data on the Typical Meterological Year (TMY) tapes for one of 26 U.S. locations. These variables are generally used as inputs to physical components.

The generation components consist of wind generation, solar-photovoltaic and utility routines. The wind turbine-generation components are fairly simple models for computing the power output of a conventional, horizontal axis wind machine given basic machine parameters. The solar-photovoltaic components are somewhat more sophisticated, especially in the collector thermal analysis, and have a number of modeling options which a user may employ, e.g., active or passive cooling.

The storage components encompass such things as motors, generators, transmissions, flywheels, etc. These components model actual physical hardware which might be used in a wind or solar energy system. The selection of the particular SIMWEST library set of storage components was based on the requirement that it be capable of modeling the five types of energy storage systems mentioned previously: thermal, flywheel, battery, pumped hydro and pneumatic.

The load components in the SIMWEST library are used to simulate various types of power demand. They also monitor how well the system meets the simulated demand and compute the value of the energy delivered to the load. Like the environmental components, these components may be computed from actual measurement data or from randomly generated data based on user furnished load profiles.

The library's logical components are the power dividers, power accumulators, switches and priority interrupts. Although physical hardware or logic devices could be built to serve the function of the logical components, they are not meant to represent any particular existing hardware. Instead, they are idealized components that allow the user flexibility in modeling a wide variety of system and control logic for operational evaluation of energy storage systems. In practice, the control function might be performed by a control room operator using a predefined control strategy or by use of a process computer.

Finally, the supplemental components include such things as the tape read, the histogram and the cost monitor. These components serve to help the user run the simulation and analyze its results.

1.2.1 Storage Subsystems

Figures 1.2-1 through 1.2-5 give possible configurations of the five types of storage subsystems which can be modeled with the present SIMWEST library. For illustrative purposes the number of variables shown passed between components is limited. A description of the variables being passed is given in Table 1.2-2.

A total energy system will generally be made up of elements from a number of different subsystems (see Figure 1.2-6). In addition, the SIMWEST program can be used for models which include networks of storage subsystems of the same type or a network of wind or solar generators.

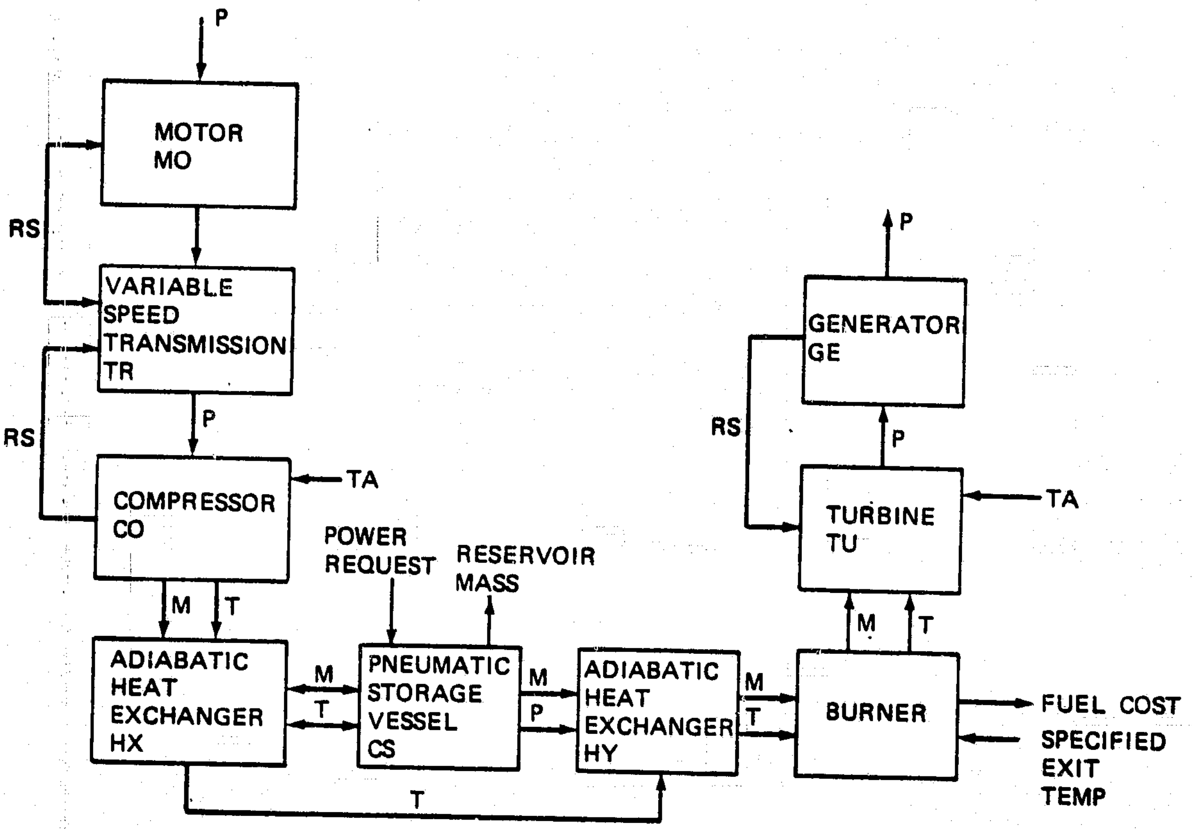


Figure 1.2-1 Pneumatic Storage Subsystem

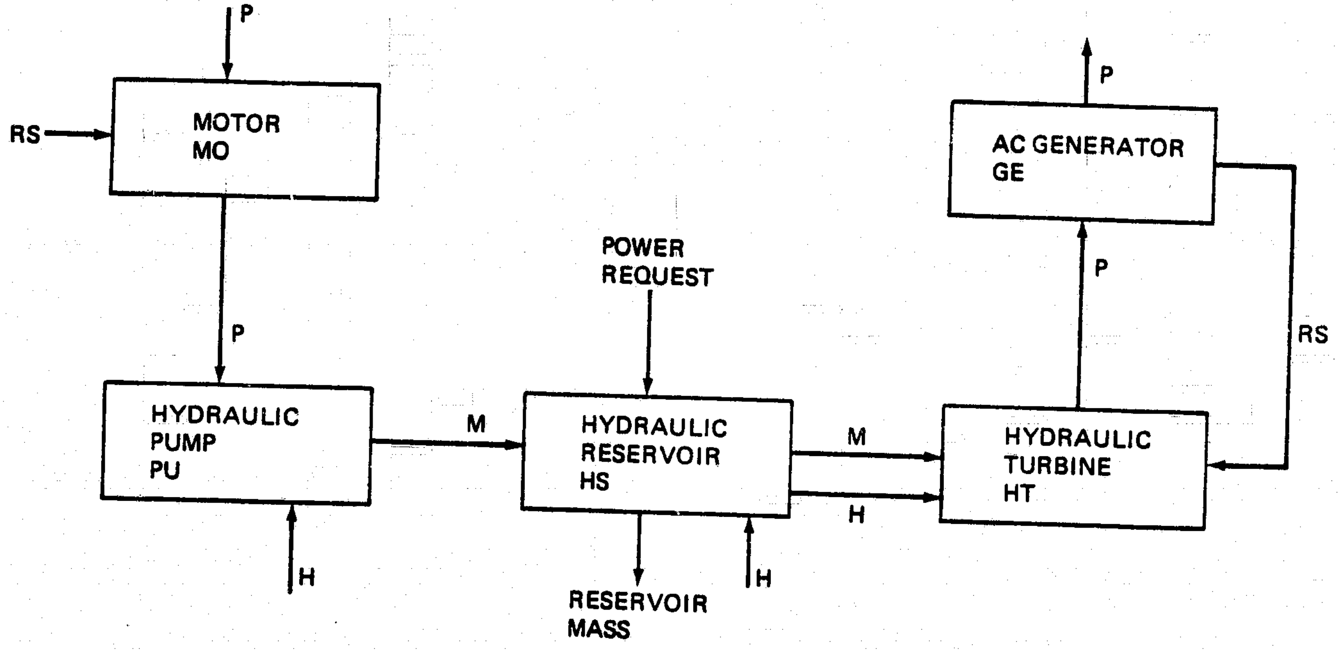


Figure 1.2-2 Pumped Hydro Storage Subsystem

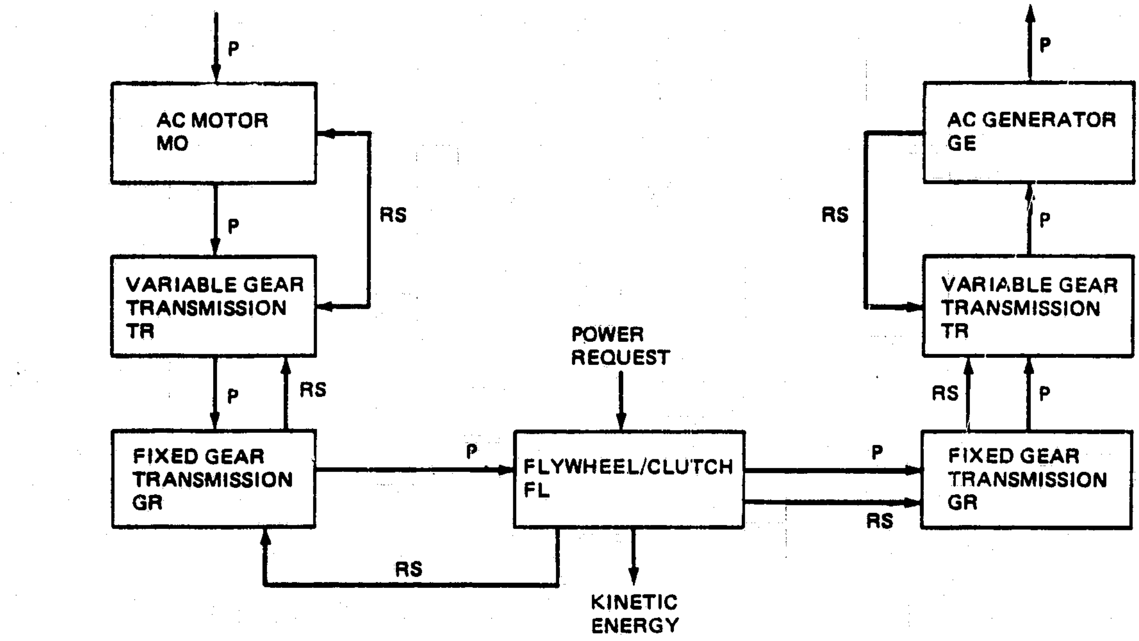


Figure 1.2-3 Flywheel Storage

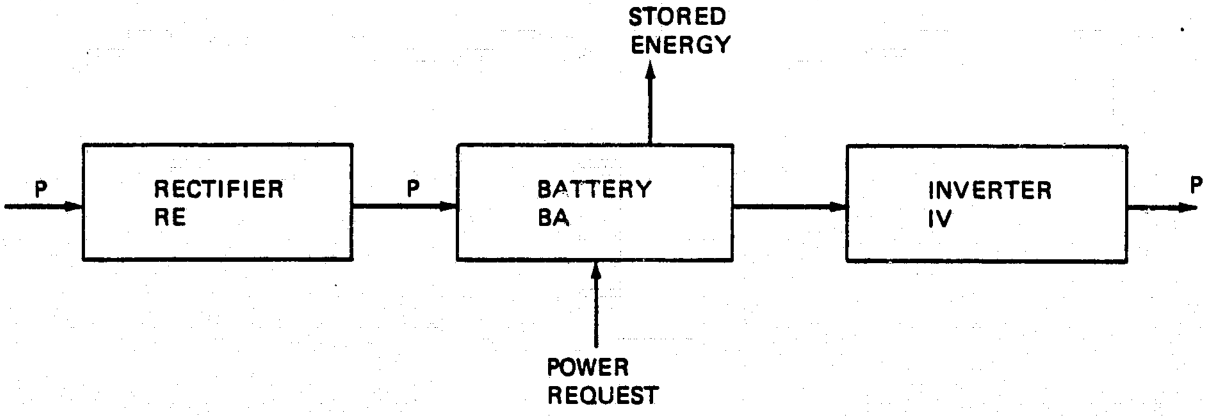


Figure 1.2-4 Battery Storage

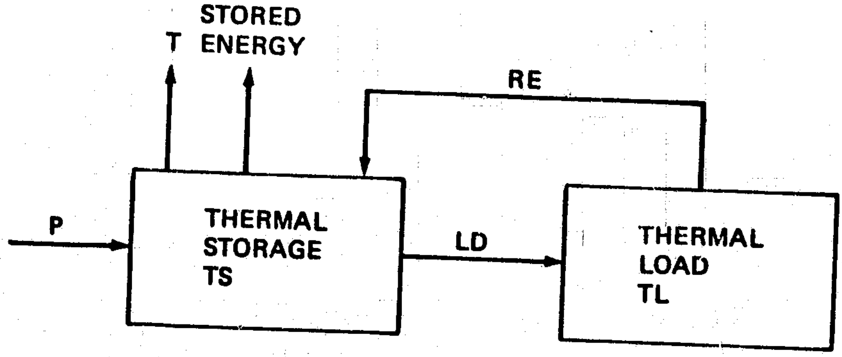


Figure 1.2-5 Thermal Storage

Table 1.2-2 Partial List of Component Inputs and Outputs

	SYMBOLS
P	POWER
RE	POWER REQUEST
MP	MAXIMUM POWER
RS	ROTOR SPEED
T	TEMPERATURE
TA	AMBIENT TEMPERATURE
M	MASS FLOW RATE
H	RESERVOIR HEIGHT
LD	THERMAL LOAD DELIVERED
WV	WIND VELOCITY
GR	GEAR RATIO
EF	EFFICIENCY
INT	INTERRUPT FLAG
PR	PRESSURE
PS	PRIORITY SEQUENCE
WY	WEEK OF YEAR
DW	DAY OF WEEK
TD	TIME OF DAY
SP	SURPLUS POWER

1.2.2 Logic Components

The capability for modeling complex system control logic is provided by the power divider, power accumulator and priority interrupt components. Both the divider and accumulator operate on a priority basis. The priority interrupt is used by other system components to change the priority setting of the divider and accumulator.

The power divider has one input power port and four output power ports (not all output ports need be used for a given simulation). The divider also has an input request associated with each of its output ports. A power request originates with a component which is directly or indirectly connected to an output port. The user specifies priorities of either 0, 1, 2, 3, or 4 to be associated with each of the output ports. If the input power exceeds that requested of the port with highest priority (priority 1) then the excess power goes to the port with the next priority. This process continues until either all power is distributed or all requests of non-zero priority ports are met. A port with zero (0) priority does not receive power. Such ports are included to model backup or switch operated components. In these situations, the connected component would change the zero priority setting of the power divider by use of a priority interrupt. Two or more ports may be assigned the same priority in which case the user may specify weights to be associated with each port. Then if there is not enough power available to satisfy all requests of equal priority, the power is divided between them in proportion to the user specified weights.

The power accumulator is similar to the divider except that instead of distributing power from a single input port among four output ports, it accumulates power from four input ports and sends it out through a single output port. The power accumulator accepts power requests from the downstream component and allocates requests to each of its input ports in order to service the downstream component.

An example illustration of the use of power dividers and power accumulators is given in Figure 1.2-6. It is seen that power from the turbine/generator is distributed with highest priority (priority 1) going to the power accumulator that services load 1. Since the power accumulator servicing load 1 has its priority 1 input port connected to the power divider, it will try first to satisfy load 1 from the turbine/generator and then from the utility. If the power divider satisfies load 1 and there is power left over, it will be used to satisfy the request from the battery. Finally, if the battery is full or if its charging rate is met, then the excess power goes to the flywheel. The battery also has a priority zero connection to the utility. If the battery remains in a discharge state for more than a specified amount of time, it can change the utility priority (from 0 to 1) to receive needed power.

Also in Figure 1.2-6, we see that load 2 prefers to draw power from the flywheel before turning to the battery. This configuration tends to keep the flywheel as discharged as possible, using it primarily as a means to absorb large influxes of power.

1.3 SIMWEST OUTPUT

There are three basic forms of SIMWEST output to facilitate the analysis of wind and solar energy storage systems; line printer plots, histograms of system variables and time sequenced output of variable values. Each SIMWEST library component is associated with a number of output variables. Prior to simulating a given system the user may select any of these outputs for plotting or tabular output. For example, he may want to plot the energy of pneumatic storage as a function of time and/or as a function of temperature. If the user wants a time sequenced listing of all variable values, he may specify the time step between printouts. The listing of all variables has proven to be a useful tool in understanding the performance of the system under consideration and a valuable aid in validating the system design.

SIMWEST also provides a special output which computes life cycle and levelized energy costs per kwh. This output is produced by the cost monitor component

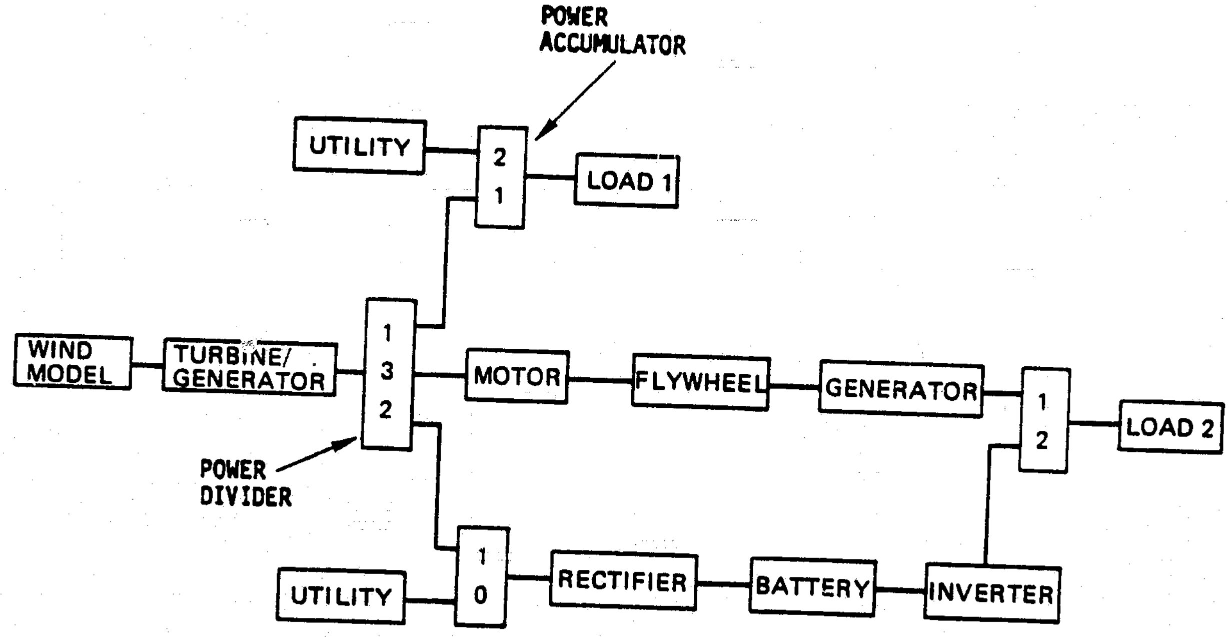


Figure 1.2-6 Example of Power Divider and Accumulator Use

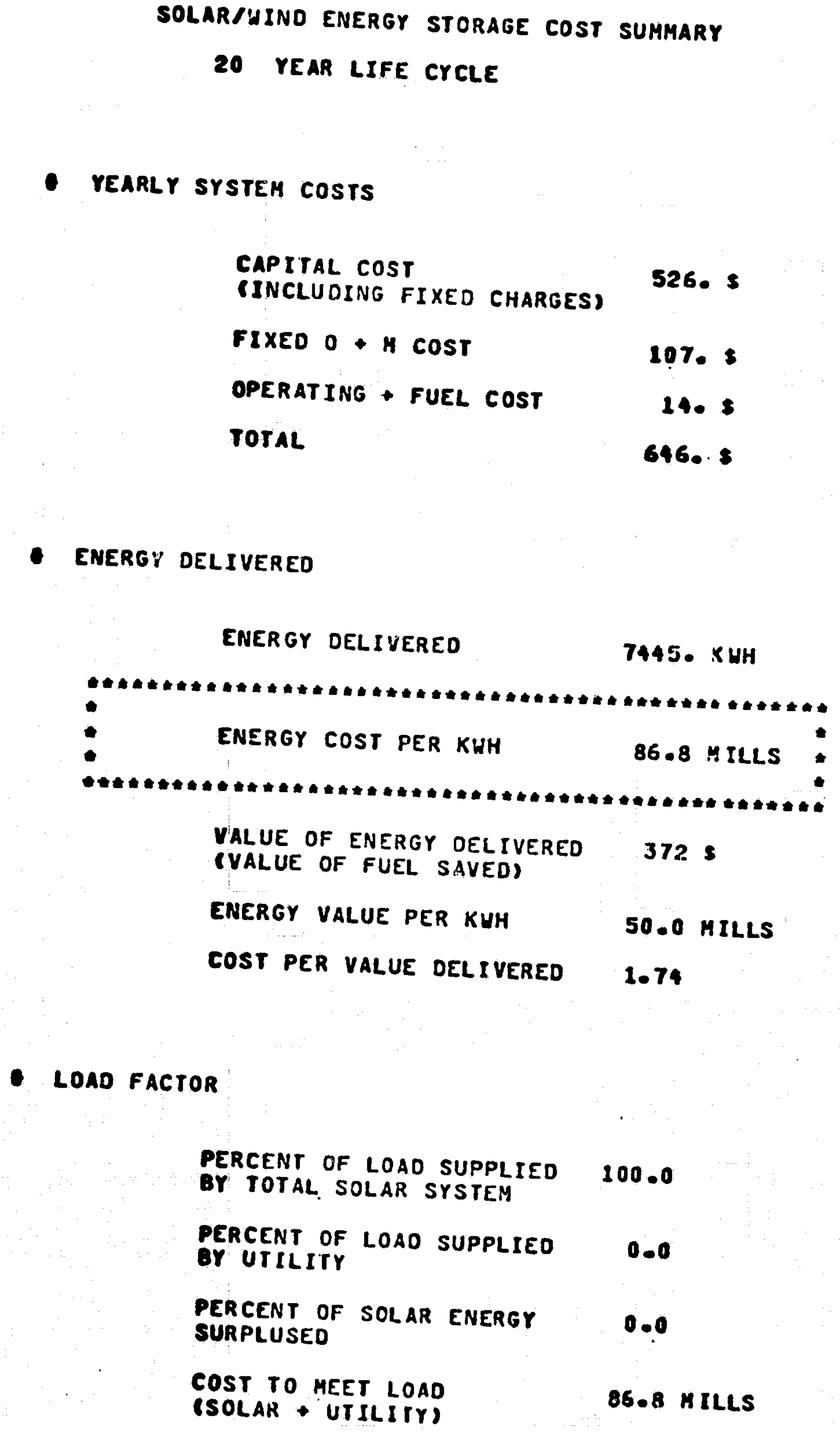
and is illustrated in Figure 1.3-1. The levelized energy costs are based on energy delivered to the loads during a simulation and forecasted to a full years' system operation. This output permits direct comparison of capital and energy costs for alternative system configurations, enabling a user to perform economic trade studies and system sizing.

1.4 TESTING

Reference [1] describes two simulation studies which were used to test the original SIMWEST program. Reference [6] describes the NASA-Lewis approved simulation studies for the expanded SIMWEST program. These studies provide an excellent test and illustration of the program's capability to model complex wind/solar energy systems.

Prior to performing the simulation studies and throughout its development the SIMWEST program was systematically tested. First components were grouped into simple systems and simulations were performed. During these simulations system parameters were driven so as to force the individual components through every normal program path and to assure that all component outputs assume a wide range of values. The number of components and the number of ways they can be connected makes it impossible to exercise every combination. However, the subsystem groupings that were used were representative of the expected program usage. Sections 8 and 9 describe some of the test cases for the wind and solar-photovoltaic generation components.

The test cases and simulation studies revealed that the code is reasonably efficient for system parameter studies. Even on very complex systems, such as represented by the NASA-Lewis test case, convergence of logic variables was quite rapid. Convergence generally took place in less than six iterations per simulation time step. As an example, the year simulations used in the NASA defined parameter study of Reference [1] took less than 420 CPU seconds on the CDC 6600. For comparison, the CPU time on the UNIVAC 1100/40 is approximately two to three times as great as that on the 6600, and CPU time on the Cyber 175 is a factor of two to three times smaller than that of the 6600.



```
               SOLAR/WIND ENERGY STORAGE COST SUMMARY
                    20  YEAR LIFE CYCLE


     ●  YEARLY SYSTEM COSTS

              CAPITAL COST                           526. $
              (INCLUDING FIXED CHARGES)

              FIXED O + M COST                       107. $

              OPERATING + FUEL COST                   14. $

              TOTAL                                  646. $


     ●  ENERGY DELIVERED

              ENERGY DELIVERED                      7445. KWH
   ****************************************************
   *                                                  *
   *          ENERGY COST PER KWH              86.8 MILLS *
   *                                                  *
   ****************************************************
              VALUE OF ENERGY DELIVERED              372 $
              (VALUE OF FUEL SAVED)

              ENERGY VALUE PER KWH                  50.0 MILLS

              COST PER VALUE DELIVERED              1.74


     ●  LOAD FACTOR

              PERCENT OF LOAD SUPPLIED             100.0
              BY TOTAL SOLAR SYSTEM

              PERCENT OF LOAD SUPPLIED               0.0
              BY UTILITY

              PERCENT OF SOLAR ENERGY                0.0
              SURPLUSED

              COST TO MEET LOAD                     86.8 MILLS
              (SOLAR + UTILITY)
```

Figure 1.3-1 Cost Monitor Output for Fresnel Lens Model

1.5 PROGRAM USAGE

During the testing it became clear that while the user need not be a SIMWEST expert or software specialist to make efficient use of the program, he should thoroughly think through and be familiar with the characteristics of the system he wants to simulate. Component models, if not carefully specified, may perform in unexpected ways. If the systems logic is not well thought out, the resulting system may be significantly out of balance and subsystems may not be fully utilized. The test case described in Reference [6] illustrates the process of sizing and logic adjustment to satisfy system performance objectives.

A number of useful procedures were developed during the simulation studies. First it was found that when simulating a complex system, it is best to separately develop and test subsystem portions of the model. This allows problems or unexpected results to be isolated and understood prior to the introduction of the more complex characteristics associated with the total system.

It was found during the simulations that the use of Fortran statements in the model definition is very useful for creating special input to system components and for defining special outputs to be plotted or statistics to be printed. For example, Fortran statements enable the user to generate and interpret trade study data by computing component input parameters from user specified system parameters. The use of Fortran statements is simple and should be encouraged early in SIMWEST applications.

Computer simulation costs may be minimized by appropriate tradeoffs between run time and simulation accuracy. Run time is most directly affected by the integration step size, the total simulation length, and the average number of iterations through the model at each time step. For long duration runs, an hour step size is usually acceptable. Models having smaller time constants than the step size may be approximated by implicit steady state conditions and solved by iteration through the model. If a model requires many iterations for convergence, then it may be useful to isolate the source of instability in order to modify or simplify that portion of the system model. It has been generally found in the simulation studies that use of a few seasonal weekly simulations is adequate to predict long term performance for system trade studies and design optimization. Four to six week-long simulations are recommended for this purpose.

When making a year simulation run, it is best to break it into twelve monthly simulations. Thus, measures of performance such as plots, histograms and performance statistics are available on a monthly basis. In addition to giving better visibility of the system performance, this helps limit the job core size. The twelve monthly simulations can be submitted as a single run with the results of a given month acting as initial conditions for the next month. The user only needs to submit new data cards for data which changes from one month to the next.

2.0 MODEL GENERATION

The Model Generation program design is based on the assumption that the system analyst will begin by constructing a schematic diagram of the system he wishes to analyze. This schematic will be comprised primarily of standard SIMWEST library components. Standard library components include wind models, AC induction motors, inverters, rectifiers, etc. If a particular system cannot be modeled with existing standard components, the analyst may construct his model by including appropriate FORTRAN statements in his system description.

All interconnections between standard components are accomplished by the Model Generation program. The analyst merely specifies each standard component in the schematic diagram and all of the components that provide inputs to that component. The Model Generation program then generates names and the proper interconnections between the specified components. This is accomplished by matching the input quantities required by each standard component to the output quantities of the specified input components.

After processing the complete system model description, the Model Generation program generates a schematic diagram of the model showing the interconnections between standard components and the quantities such as power, pressure, temperature, mass flow rates, etc., that pass through each interconnection. This schematic is produced on the lineprinter to provide a rapid graphic check on the program's interpretation of the model description.

In addition, the program produces a list of input data that will be required by each component to complete the model description. Both the scalar parameters and tabular data required for the analysis are included in this list. The program assumes that any quantity not supplied by another component will be supplied as a fixed parameter by the analyst. Thus requests for non-parameter items in the input data list will reveal any connection that was omitted from the system model description.

2.1 MODEL DESCRIPTION

The Model Generation program is a precompiler program which accepts model description instructions and from these instructions generates a FORTRAN model of a system. These instructions, referred to as "program commands," are made up of one or more words. In addition, the system model description contains numeric values, standard component names, and standard input and output quantity names.

The Model Generation commands may be best introduced with a simple example of their use to describe a wind turbine system. Figure 2.1-1 shows an analyst's schematic of a wind turbine model that has been constructed using standard components on a SIMWEST schematic form. The standard component names used in this sample are:

WD - Wind Model
WT - Wind Turbine
TI - Time Conversion
HG - Histogram Generator
GR - Fixed Ratio Transmission
LO - Electrical Load
GE - AC Induction Generator

The SIMWEST description of this model would be as follows:

Example 2.1

```
MODEL DESCRIPTION       WIND TURBINE TEST CASE
LOCATION=15        TI
LOCATION=11        WD       INPUTS=TI
LOCATION=31        WT       INPUTS=WD
LOCATION=35        GR       INPUTS=WT
```

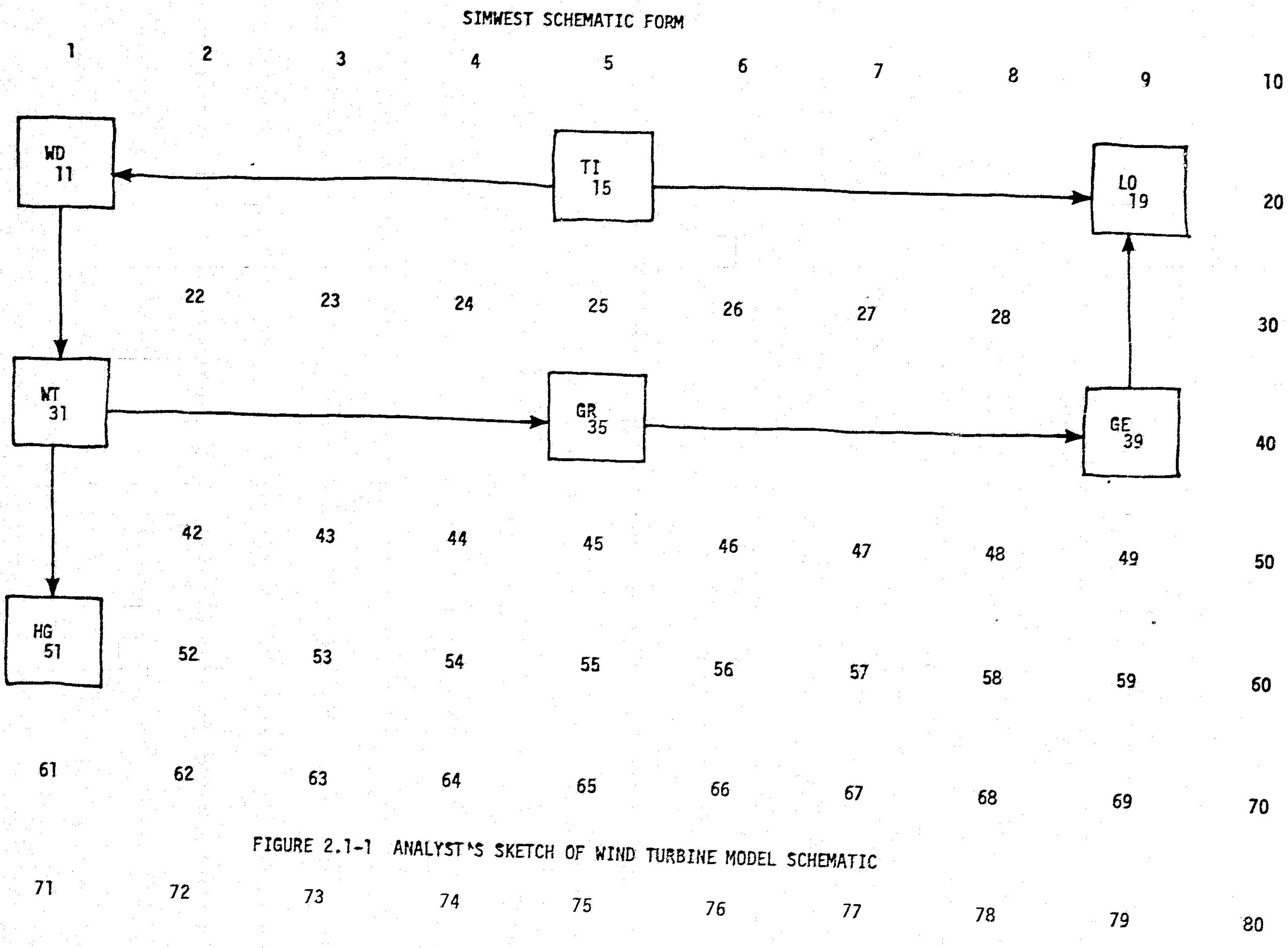


FIGURE 2.1-1 ANALYST'S SKETCH OF WIND TURBINE MODEL SCHEMATIC

Example 2.1 (Continued)

```
LOCATION=51     HG     INPUTS=WT(P=FIN)
LOCATION=39     GE     INPUTS=GR
LOCATION=19     LO     INPUTS=GE,TI
END OF MODEL
PRINT
```

The model description consists of a statement as to the location of each component in the schematic and a list of all components that provide inputs to that component. The location of the component in the schematic is used for a line printer drawn schematic of the model, such as shown in Figure 2.1-2. In the line printer schematic the connection variables such as powers (P2 WT, P2 GE, P2 GR) are shown on the various connecting lines.

2.1.1 Phrases and Delimiters

The system model description is interpreted by the Model Generation program as a series of "phrases", which can appear in a free field format in any position on a data card. Phrases must be separated by any one of the delimiter symbols shown in Table 2.1-1.

Table 2.1-1

Model Generation Program Language Delimiters

=	equal sign
,	comma
(	left parenthesis
)	right parenthesis
	three or more blanks

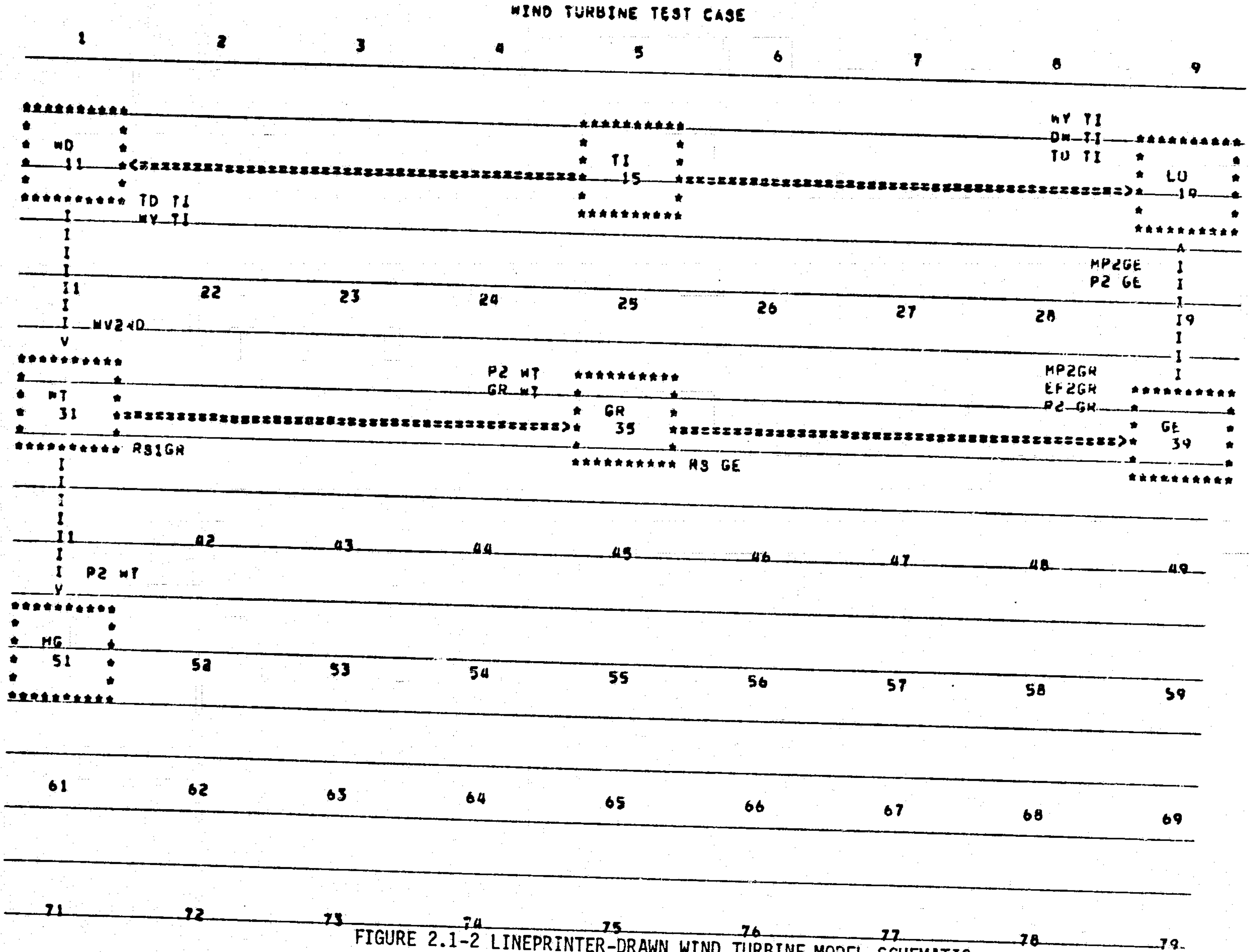


FIGURE 2.1-2 LINEPRINTER-DRAWN WIND TURBINE MODEL SCHEMATIC

2.1.2 Command Phrases

The Model Generation command phrases are described in this section in a logical sequence similar to that in which they appear in system model descriptions.

MODEL DESCRIPTION

The MODEL DESCRIPTION command phrase indicates the start of a new system model. This phrase may be followed, (on the same card), by a title of up to 60 characters. This title will be used to identify various program output schematics, lists and program listings. In Example 2.1, the title was "Wind Turbine Test Case."

LOCATION

The LOCATION command phrase indicates the start of the description of a new component in the system model. This command must be followed by a numeric value phrase that specifies the location of the new component on the model schematic. Thus in the example of Figure 2.1-1, the location number of the wind model WD was 11 and the wind turbine WT was 31, etc. To be a valid component location, the last two digits of this number must comprise a number between 1 and 80. The hundreds column is used to specify additional pages as needed for the schematic. Thus the numbers

1, 13, 51, 80

would be valid location numbers for components on the first page, (PAGE 0), of a system schematic. These same locations on the second page of the schematic, (PAGE 1), would be:

101, 113, 151, 180

The location number phrase is followed by the name of the component at that location. Component names are discussed in Section 2.2.

A LOCATION statement should be given only once for each component. That is, once a LOCATION statement is started for a component the complete description of all inputs to that component should be given.

INPUTS

The INPUTS command phrase indicates that the following phrases contain the names of the components that provide inputs to the component at the specified location. Thus in the example of Figure 2.1-1, the electric load at location 19 which receives inputs from generator GE and the time source TI was described as:

```
LOCATION=19     LO     INPUTS=GE,TI
```

In this example the command phrase INPUTS is followed by two component names. As many component names as are necessary to specify the inputs to a particular system component may be included in each component description.

For some system components there are multiple input and/or output ports. For example, a power divider has four input power ports. When specifying the connections between such components, it is advisable to specify which ports are to be connected. This is done by adding the port numbers to be connected after the name of the input component. Thus, the wind turbine to transmission connection could have been more explicitly described as:

```
LOCATION=35     GR     INPUTS=WT(2,1)
```

This says that port 2 of the wind turbine (WT) drives port 1 of the transmission (GR). Any quantities which have no port numbers are considered "universal ports" for input connections. Thus, the GR input of GR is connected up to GR WT, and the RS input of WT is connected up to RS1GR by the above command. If the port designations are omitted, as they were in Example 2.1, the connections will be made to the first available input port starting with the minimum port number. Once a connection has been made to an input port, those input quantities that are connected are no longer available for further connections. An exception is made when the physical quantities of both input and output are specified. This method of specifying connections is described in the following paragraphs.

For certain components, such as the arithmetic elements, the inputs to the component can be any physical quantity in the model. For these components, the input component names must be supplemented by the name of the particular output quantity that is to provide the input.

As an example, consider a component that represents a linear first order lag transfer function. If the transfer function component's input, FIN, was to be the rotor speed of the wind turbine WT in Example 2.1, then the statement

```
LOCATION=53      LA      INPUTS=WT(RS=FIN)
```

would indicate to the program that of the outputs of the wind turbine, the output rotor speed, RS, was to be used as the input, FIN, to the transfer function, LA.

To summarize, there are three levels of connection specification:

1. Default (only component names are specified)

   Connections are made between all unconnected inputs and outputs for the first ports for which a match of physical quantity names occurs.

2. Ports Specified

   Connections are made between matching physical quantities for all unconnected inputs and outputs of the specified ports.

3. Physical Quantities Specified

Connections are made between only those quantities specified. Previous connections can be overridden, providing the three character physical quantity name of the previously connected variable is used. For example, the phrase

```
LOCATION=19        LO       INPUTS=GE,GE(P,2=MP2)
```

will first replace the input parameter MP1LO by MP2GE and then override the connection MP2GE and substitute P2 GE as the LO input.

Note: The LIST STANDARD COMPONENTS command produces a listing of all input and output physical quantity names and port numbers. When specifying individual physical connections this listing may be used to differentiate physical quantity names and port numbers. For example, (P,2=...) denotes connection of a physical quantity P at port 2, whereas (P2=...) denotes connection of physical quantity P2 without regard to port number.

END OF MODEL

The END OF MODEL command phrase indicates that the model description has been completed and that the Model Generation program should proceed with the generation of the model subroutines.

PRINT

The PRINT command phrase causes the program to: (1) draw a schematic of the system model, as shown in Figure 2.1-2; (2) print a list of input requirements for the model; and (3) print a source listing of the FORTRAN subroutines that were generated for the model. The Model Generation program then terminates.

PUNCH

The PUNCH command phrase has the same effect as the PRINT command, but in addition a FORTRAN source deck of the system model is produced.

FORTRAN STATEMENTS

The FORTRAN STATEMENTS command phrase allows the system analyst to supplement the library components with FORTRAN statements. Using this feature, the analyst can introduce his own program logic, DO loops, etc., as necessary to model any system feature not obtainable with standard library components.

One of the common uses of the FORTRAN STATEMENTS command is to input large tables into the model. Two function subprograms TBLU1 and TBLU2 are provided for this use. They perform linear interpolation from one and two dimension tables, respectively. TBLU1 is in general called in the form

$$F = TBLU1(X,TAB(4),TAB(4+N),I,\pm N),$$

where F is the interpolated value at the desired point X, TAB is a one dimension table with dimension N, TAB(4) is the independent variable and TAB (4+N) is the dependent variable list, I = 0 for equal spaced data, I = 1 for unequal spaced data, and the dimension N is specified as the last variable if linear extrapolation is desired, and -N is specified if truncation is desired outside the table limits. Similarly, TBLU2 is in general called using the form

$$F = TBLU2(X,Y,TAB(4+M),TAB(4),TAB(4+M+N),IX,IY,\pm N,\pm M,N,M),$$

where X and Y are the values of the primary and secondary independent variables, N and M are the dimensions of the primary and secondary variable arrays, IX and IY are indicators for equal spaced or unequal spaced data as above, and the sign convention on N and M is positive for extrapolation, negative for truncation.

The FORTRAN STATEMENTS command would normally be used only when some portion of the system cannot be modeled with library components. When using this feature of the program, the analyst must include detail connections and naming of variables, that are normally accomplished by the Model Generation program. In return for these added tasks, the analyst gains a great deal of additional flexibility in forming details of his system model. Non-executable code such as common blocks must precede the first component definition and executable code should come after a component has been defined for the iteration logic to work properly.

ADD STATES
ADD VARIABLES
ADD PARAMETERS
ADD TABLES

The ADD COMMANDS may be used in conjunction with the FORTRAN STATEMENTS to add states, variables, parameters, and tables that occur within the FORTRAN statements, to the system model. Quantities that are not specified by one of these commands cannot be accessed or manipulated by the Analysis Program.

Before discussing these commands, a few definitions of terms are in order.

States: States are those quantities in the system model that are described by first order differential equations. The state variables are the result of integrating the set of first order differential equations that comprise the dynamic system model. The number of states equals the order of the system model. The states are dynamic, time varying quantities during most simulation studies. The initial values, (initial conditions), of the states must be input as part of the system model description. Derivatives of the state variables are stored in an array XDOT where XDOT(I) is the derivative of the Ith state variable stored in the model (EQMO).

Variables: Variables are all other dynamic time varying quantities in the system model that are not states. In general, variables are related to states by algebraic relationships.

Parameters: Parameters are constant scalar quantities in the system model. Parameters can be manipulated by the analyst to alter the system model. All parameter values* should be input as part of the system model description.

Tables: Tables are constant nonscalar quantities in the system model. Tables are used to represent algebraic functional relationships with one or two independent variables. All table values must be input as part of the system model description.

The format for the ADD commands is that the command is followed by one or more phrases that contain the names of the states, variables, parameters, or tables. In addition to each table name, a number, specifying the amount of storage to be allocated for that table must be given. This number is positive if the table is two dimensional and negative if one dimensional, with absolute value determined by the formula:

$$N = 3 + I + J + D$$

where

N = The total storage required by the table, in words.

* For certain components, default values are provided for some parameters.

I = The number of data points in the primary independent variable table.

J = The number of data points in the secondary independent variable table. (J=0 if there is only one independent variable.)

D = The number of data points in the dependent variable table. (D=I if there is only one independent variable. D=I*J if there are two independent variables.)

The following example from reference [1] illustrates the use of FORTRAN STATEMENTS:

Example 2.2

```
MODEL DESCRIPTION      PARAMETER STUDY
        .
        .
        .
ADD TABLES = WIND,802
LOCATION = 41    TI
FORTRAN STATEMENTS
C          READ WIND VELOCITY DATA
      WV1WD = TBLU2(TD TI,DY TI,WIND(35),WIND(4),
     1 WIND(59),0,0,24,-31,24,31)
LOCATION = 71    WD     INPUTS = TI
        .
        .
        .
```

In this model, Fortran is used to input wind velocity data. The wind table, denoted WIND, consists of up to 31 days of hourly wind speeds. Hence, as described previously, the total storage required is 3+24+31+24*31=802. The Fortran is inserted after time of day and day of the year are computed in TI. In this case, N=24, M=31, the data is equal spaced, and extrapolation is used to provide velocity data over each 24 hour period. The variable WV1WD is the name of the wind input to WD generated by the precompiler. Fortran insertion in the model ends when the LOCATION=71 ... command is read and a call to the subroutine WD is then generated.

Note: When interpolation of one dimension, equi-spaced data is desired, it is possible to reduce the table dimension and table input by using the following alternative procedure:

(1) Set the table dimension for the ADD TABLES command to 6+N, where N is the length of the dependent variable data.

(2) The call sequence for linear interpolation is changed to

```
F = TBLU1(X,TAB(4),TAB(6),0,±N),
```

where F is the interpolated value at the desired point X and TAB is the one dimension table name specified in the ADD TABLES command.

(3) The tabular data input to the simulation program as specified in 3.1-2 is modified such that only the first two values of the independent variable table are specified, i.e., the data cards for the table TAB may be input as follows:

Card 1	TABLE, TAB = N1
Card 2	X1, X2
Card 3	Y1, Y2, ... YN

where N1 = N/2 + 1 if N1 is even and N1 = (N+3)/2 if N is odd, X1, X2 are the first two values of the independent variable table, and Y1, ... N are the values of the dependent variable table.

LIST STANDARD COMPONENTS

The LIST STANDARD COMPONENTS command phrase causes the program to print a list of all standard components. For each standard component, lists of inputs, outputs, and tables for that component are provided. For each input, the physical quantity name and port number is given. For each output, the physical quantity name, port number, and the letter S, if the quantity is a state is given. For each table, the table name, the number of independent variables and the maximum amount of storage allowed is provided. This command is usually given as the first command of a model description and will result in a list of all standard component information as the first output from the Model Generation program.

2.2 NAMING CONVENTION

All standard components are given names consisting of two characters, the first of which is alphabetical. Thus we have WT for wind turbine, GE for generator, WD for wind model, etc. Where multiple components of the same type are required, the second character is used to distinguish between the different models of the same basic component type. A specific component in a model can be distinguished from other components of the same type by adding one more character to the standard component name. This character is usually numeric but can also be alphabetical or blank. Thus a given model can contain up to 37 different components of the same standard component type. For example, a model with ten different wind turbines might have these components designated as:

WT1,WT2,WT3,.....,WT8,WT9,WTA

2.2.1 Variable, Parameter, and Table Naming Conventions

All of the input, output, and tabular quantities required by each component in a system model must have unique FORTRAN names. These quantities are given names consisting of up to three characters that describe the physical quantity they represent.

Since a single component may have several inputs or outputs of the same physical quantity, a port number may be added to the second or third character of the physical quantity name to prevent duplication.

The physical quantities that are outputs of a given component are identified by catenating the three character name of that component to the three character name of the physical quantity. In this way, unique six character FORTRAN names are generated for all output quantities of the system model.

Input quantities to a component that are driven by another component carry the names of the component that drives them. Any inputs that are not driven by other model components are assumed to be parameters and are assigned the name of the component for which they are an input.

If a component should require tabular data as an input, unique table names are generated just as scalar input quantity names by adding the component name to the table name. A pictorial representation of the character assignment in component, variable, and table names is given in Figure 2.2-1.

2.3 MODEL SCHEMATIC

The Model Generation program produces an information flow or schematic diagram of the system being modeled. This schematic is crude but is inexpensive and does not have the flow delays associated with more elaborate plotting methods. Its purpose is to provide a means of rapidly locating errors in the model description.

In order to construct a schematic diagram in an efficient manner with a reasonable size program, it was necessary to establish some simple rules for symbol generation, component connection paths, and labeling. If these rules are kept in mind when laying-out a schematic for the system, the SIMWEST produced schematic will match that developed by the analyst. If the rules are violated, the SIMWEST schematic should still be correct, but may contain some unusual component connection paths and some labeling information may be overwritten.

INPUT/OUTPUT OR TABLE NAMES

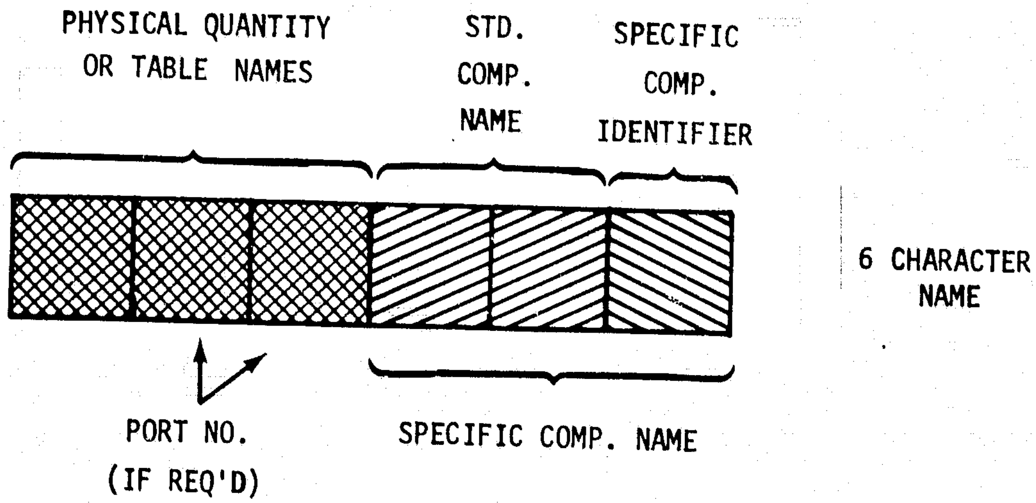


FIGURE 2.2-1 CHARACTER ASSIGNMENT INPUT/OUTPUT OR TABLE NAME

2.3.1 Standard Schematic Form

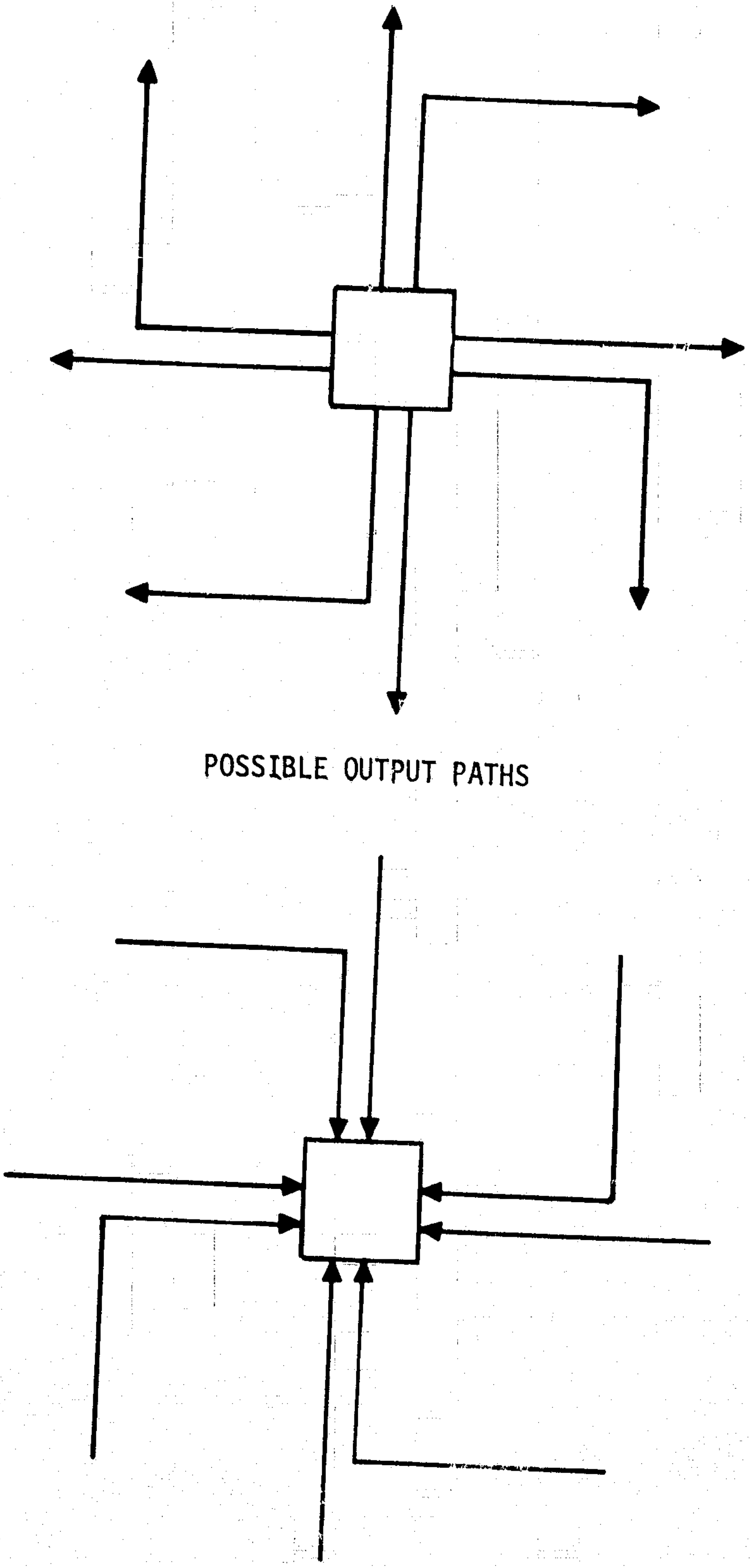
The SIMWEST schematic diagrams are produced on a standard 11" by 14" line-printer page with 80 component locations per page. A standard form containing only the location numbers can be obtained by executing the Model Generation program with the single program command, PRINT. This form can then be reproduced and the copies used as forms for drawing system model schematics.

2.3.2 Input Quantity Labeling

The names of the physical quantities that are input to one component from another component are listed adjacent to the downstream component symbol. These labels are placed near the connecting line that joins the two components. Since these names are composed of the physical quantity name and the name of the component that generates the information, the source of the input is evident from the name itself. Parameter and tabular inputs to a component are not shown on the schematic. These constant inputs are described in the Input Requirements List.

2.3.3 Component Connection Paths

In order to keep the requirements of the SIMWEST schematic subroutine small, it was necessary to limit the types of connecting paths between components to a few basic routes. These paths are shown in Figure 2.3-1. Connections between components on the same horizontal or vertical line are straightforward. However, connections between components that do not share a horizontal or vertical line require a two segment path. These paths have been arbitrarily chosen to follow a clockwise route. It is therefore advisable that components that are on diagonal locations be placed in a clockwise sequence. If counter-clockwise flow between components is necessary, it can be accommodated by placing the components on the same horizontal or vertical lines.



POSSIBLE OUTPUT PATHS

POSSIBLE INPUT PATHS

FIGURE 2.3-1 COMPONENT CONNECTION PATHS

The SIMWEST schematic subroutine makes no attempt to go around components that get in the way of a connection path. Such components are "run-over" by the connecting line.

2.3.4 Additional Pages

The SIMWEST schematic diagram may be broken into as many pages as are necessary. No attempt is made to draw connecting paths between components located on different pages. It is therefore advisable to minimize the number of connecting paths between pages. This can usually be done by grouping components with many interconnections on the same page and placing page boundaries between such groups of components.

2.3.5 Guidelines for Schematic Layout

The following guidelines may help in creating schematic layouts that can be duplicated by the SIMWEST program.

1. Try to place connected components on the same horizontal or vertical line.
2. Avoid placing components on adjacent location points.
3. Place diagonal components so that flow is clockwise.
4. Group components to minimize flow paths between pages.

2.4 WARNING MESSAGES

One or more of the following warning messages will occur if the program is unable to interpret a portion of the model description or encounters problems in assembling the system model. These messages will be preceded by: *** WARNING *** or *** NOTICE ***. The symbols xxx and zzz are used to indicate phrases from the model description that are included as part of the warning message. The following messages are listed in alphabetical order:

1. CAN'T IDENTIFY xxx AS A STANDARD COMPONENT

   xxx will contain the first two characters of the phrase which cannot be identified as a command or standard component. This message will often follow other warning messages as the program makes successive attempts to interpret the given phrase.

2. CAN'T IDENTIFY xxx AS A VALID INPUT COMPONENT TO zzz

   The component xxx cannot be found in the list of components for the current system model.

3. CAN'T LOCATE xxx AS AN INPUT COMPONENT TO LOCATION n

   This message indicates that the component xxx, which provides inputs to location n in the schematic, has not been assigned a location number. Check for a missing LOCATION statement or misspelling of the component name.

4. COMPONENT xxx DEFINITION WASN'T COMPLETED BEFORE STARTING THE DEFINITION OF COMPONENT zzz

   The command INPUTS was not given between the component names xxx and zzz. Check for proper spelling of INPUTS and a valid delimiter after the phrase xxx.

5. COMPONENT xxx HAS ALREADY BEEN DEFINED

   The component xxx was defined in a previous LOCATION statement.

6. LOCATION NO. xxx FOR COMPONENT zzz HAS LAST TWO DIGITS OUTSIDE THE ALLOWABLE RANGE 1 TO 80. NO SYMBOL WILL BE PLACED IN SCHEMATIC FOR THIS COMPONENT

This message will occur at the end of the model description for a component zzz which has an invalid location number. The system model may still be valid but the schematic will not contain this component.

7. NO xxx OUTPUTS MATCH UNSATISFIED zzz INPUTS

Check that it was intended to drive component zzz with component xxx or that the inputs to zzz have been previously satisfied by other component connections.

8. TABLE NAME xxx MUST BE FOLLOWED BY A NUMERIC DIMENSION RATHER THAN zzz

When using the ADD TABLES command, it is necessary to provide the maximum amount of storage to be allocated for the table as well as the table name. This storage value must be a numeric quantity.

9. THE FOLLOWING COMPONENTS FORM AN IMPICIT LOOP. xxx, zzz, ...

Implicit loops can often be corrected by inserting a component with a state variable as its output, e.g., a simple linear lag, LA. All models containing FORTRAN STATEMENTS will receive this warning.

10. THE SEQUENCE OF THE FOLLOWING COMPONENTS HAS BEEN ALTERED TO FORM AN EXPLICIT MODEL. xxx, zzz, ...

The model component sequence as given contained implicit equations. By altering the component sequence it was possible to form an explicit model.

11. xxx IS NOT A VALID INPUT QUANTITY OR PORT DESIGNATION FOR COMPONENT zzz

The phrase xxx cannot be located as one of the input quantities or input ports of the component zzz. No connections will occur. Check the list of standard components for the proper spelling or port designations for this component.

12. xxx IS NOT A VALID LOCATION NUMBER

The LOCATION command must be followed by a numeric location number.

13. xxx IS NOT A VALID PORT DESIGNATION FOR INPUT COMPONENT zzz. ERRONEOUS CONNECTIONS MAY OCCUR.

The phrase xxx cannot be located as a valid input port for the component zzz. Connections will be attempted using the upstream output port that was identified.

2.5 MODEL GENERATION LIMITATIONS

Certain limitations exist in the Model Generation program due to array dimensions within the program. For most applications these limits should not be encountered. However, if they should be encountered they can usually be extended at the expense of larger core requirements to execute the program. The following table describes these limitations:

Limitation Description	Maximum Value
Standard components in library	150
Components per model	200
States per model	200
Inputs per any standard component	50
Outputs per any standard component	50
Tables per any standard component	15
Ports per any standard component	10
Tables per model	100
Table dimension (words)	960

3.0 SIMULATION PROGRAM

Once a model has been generated as described in Section 2.0, the user must describe the simulation he wishes to perform. This involves specifying the various parameters detailing the model components and setting the model initial conditions. It involves defining input data tables and the type and quantities of printed output, both tabular and plotted. The user must also specify the number of iterations he wishes to perform at each time step and the maximum number of component diagnostics. This section describes the commands for specifying the simulation and gives some example output.

3.1 MODEL INPUT DATA

A dynamic system model requires that the values of model parameters, tables and initial conditions, be provided to complete the model description. Sections 3.1.1, 3.1.2 and 3.2 describe the methods used to specify parameter values, tables, and initial conditions.

3.1.1 Scalar Data

PARAMETER VALUES (Default values = .99999)

This program command allows the numeric values of parameters to be loaded into the system model. The PARAMETER VALUES command is followed by one or more parameter names followed by a numeric value. Each name and its value are separated by one of the standard delimiter symbols. This command is used to specify the values of all system model parameters at the beginning of an analysis. It may also be used at any point between analyses to modify the value of one or more model parameters. A default value of .99999 is provided by the Model Generation program for all parameters not so specified.

PRECEDING PAGE BLANK NOT FILMED

Example 3.1-1

```
PARAMETER VALUES = CYCLES = 6.01, TO  TI = 0, EW  WP = .2,
CR  CM = 15, LE  CM = 30, MDEHS = 4.E5, .....
```

3.1.2 Tabular Data

If tabular data is required by the system model, it should be loaded before any of the simulation commands described in Section 3.4 are issued. Tables may be modified between analyses by loading new values. The tables required by a SIMWEST generated model are specified in the Input Requirements List. These tables may have either one or two independent variables. All data items are in a free field format with each item separated by one of the standard delimiters: comma [,], equal sign [=], left or right parenthesis [()], or three or more consecutive blank spaces. The data items required for each table are placed on cards as follows:

Card 1	TABLE	table name	NX	NZ
Card 2*	Z table values			
Card 3*	X table values			
Card 4*	Y table values			

where: Table Name - The six character table name generated by the Model Generation program.

NX - The number of points in the primary independent variable table.

NZ** - The number of points in the secondary independent variable table.

Z table ** - Table of NZ secondary independent variable values.

X table - Table of NX independent table values.

Y table - 1 or NZ tables of NX dependent variable values.

* As many cards as required may be used. Each table must start with a new card and NZ, NX, and NX*NZ points must be given per table.

** These items are omitted for tables with one independent variable.

A copy of all tabular input data is printed as it is interpreted from data cards. The following example shows the data cards for a one and a two independent variable table.

Example 3.1-2

Card 1	TABLE, TABONE, 10
Card 2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10
Card 3	11, 12, 13, 14, 15, 16, 17, 18, 19, 110
Card 4	TABLE, TABTWO, 5, 4
Card 5	10.3, 20.4, 30.5, 40.6
Card 6	1, 2, 3, 4, 5
Card 7	11, 12, 13, 14, 15
Card 8	21, 22, 23, 24, 25
Card 9	31, 32, 33, 34, 35
Card 10	41, 42, 43, 44, 45

The printout of these tables would be:

```
                                 TABLE TABONE
                       PRIMARY INDEPENDENT VARIABLE TABLE
 1.000   2.000   3.000   4.000   5.000   6.000   7.000   8.000   9.000   10.00
                           DEPENDENT VARIABLE TABLE
 11.00   12.00   13.00   14.00   15.00   16.00   17.00   18.00   19.00   110.00
                                 TABLE TABTWO
                      SECONDARY INDEPENDENT VARIABLE TABLE
 10.30           20.40           30.50           40.60
                       PRIMARY INDEPENDENT VARIABLE TABLE
 1.000           2.000           3.000           4.000           5.000
                           DEPENDENT VARIABLE TABLE
 11.00           12.00           13.00           14.00           15.00
 21.00           22.00           23.00           24.00           25.00
 31.00           32.00           33.00           34.00           35.00
 41.00           42.00           43.00           44.00           45.00
```

3.2 INITIAL CONDITION, ERROR, AND INTEGRATION CONTROLS

```
INITIAL CONDITIONS (Default value = 0)
ERROR CONTROLS     (Default value = 0.1)
INT CONTROLS       (Default value = 1.0)
```

These program commands may be used to specify state variable initial condition values, integrator error controls, and integrator status, (either active (=1) or frozen (=0)). Default values are furnished by the simulation program. However, it is strongly recommended that values appropriate to the particular system model be furnished for the initial conditions and error controls.

Each of these commands is followed by phrases of the form of a state name followed by a numeric value.

Example 3.2-1:

```
INITIAL CONDITIONS = MA HS = 1.6E6, E  TS = 600, VDELO = 0, ....
ERROR CONTROLS = MA HS = 10, ....
INT CONTROLS = MA HS = 0, E  TS = 1, VDELO = 1, .....
```

```
ALL STATES (Default Condition)
NO STATES
```

These program commands may be used to activate or freeze all system integrators. These commands are normally used together with the INT CONTROLS command to specify the desired integrator configuration.

3.3 INITIAL CONDITION STORAGE COMMANDS

```
XIC-X
XIC-XIC1
XIC-XIC2
XIC-XIC3
XIC1-XIC
XIC2-XIC
XIC3-XIC
```

These program commands are used to transfer data from the current state vector, X, to the initial condition vector, XIC, and between the XIC vector and three auxiliary initial condition vectors XIC1, XIC2, XIC3.

Example 3.3-1

```
XIC1-XIC, XIC-X, XIC2-XIC
```

The three program commands shown above would take the current operating point (initial condition vector) and store it in vector XIC1; then transfer the current state, X, into XIC; and then store that value of XIC in XIC2.

3.4 SIMULATION COMMANDS

SIMULATE

This program command initiates simulation operation. Associated with this command are the program values:

			Default Values:
TINC	=	time increment, hours	0.1
TMAX	=	duration of the simulation run, hours	1.0
INT MODE	=	integration mode control	3
OUTRATE	=	output rate	1
PRATE	=	print rate	1
PRINT CONTROL	=	print control variable	0

These program commands specify the integration time increment, duration of simulation run, the integration mode, the simulation output rate, the printing rate, and the quantity of printing, at each point in time. These quantities should be specified before the first issuance of the SIMULATE command.

The Time increment, TINC, provides the integrator time step size, in hours, for the integrator. TINC also provides the report interval for which data will be available for printing or plotting. The default value for TINC is 0.1.

The duration of a simulation calculation in hours, is specified by the TMAX parameter. The default value of TMAX is 1.

The integration mode control, INT MODE, allows one of three different integration methods to be selected according to the values given below. The default value of INT MODE is 3.

Integration Method Selection

INT MODE	Method
1	Variable Step, Variable Order Gear
2	Variable Step, 4th Order Runge-Kutta
3	Fixed Step Euler (2nd order)

The error controls specified in Section 3.2 determines the system step except for INT MODE = 3.

The output rate parameter, OUTRATE, determines the sampling rate at which simulation data is added to plots. Thus, if OUTRATE is set equal to 10, data will be plotted every 10th time increment, TINC. The default value of OUTRATE is 1. OUTRATE should only be set to positive integer values.

The number of data samples plotted for a simulation analysis is thus given by:

$$\text{No. of Plotted Samples} = \frac{\text{TMAX}}{\text{TINC*OUTRATE}} + 1$$

For most simulation operation, the plotted output specified by the DISPLAY commands is the primary output and little printer output is used. However, for diagnosing problems in a simulation, the line printer options provided by the PRINT CONTROL parameter allow large amounts of detailed information about the simulated system to be obtained.

The value of the PRINT CONTROL parameter controls the quantity of data printed at each report interval as shown in Table 3.4-1. Options 1 through 4 give "snap-shots" of all states, rates, variables, and parameters of the system model at a particular point in time. Option 5 provides tabular lists of up to 10 specified quantities.* The default value for PRINT CONTROL is 0.

TABLE 3.4-1

Print Control Values

PRINT CONTROL	Resultant Lineprinter Output.
0	None (Default Condition)
1	All states, rates, and time
2	All states, rates, variables, and time
3	All states, rates, variables, and parameters at time = 0
4	All states, rates, variables, and parameters
5	Time and the quantities specified via PRINT VARIABLES command.

The PRATE parameter determines the sampling rate at which the simulation data specified by the PRINT CONTROL parameter is presented on the lineprinter. Thus if PRATE is set equal to 5, data will be printed on the lineprinter every 5th

* See the PRINT VARIABLES command description below.

time it is added to the output plots. The rate of output to the lineprinter can never be greater than that to the plots. The default value of PRATE is 1. PRATE should only be set to positive integer values.

The number of data samples printed for a simulation analysis is thus given by:

$$\text{No. of Printed Samples} = \frac{\text{TMAX}}{\text{TINC*OUTRATE*PRATE}} + 1$$

Example 3.4-1:

```
PRINT CONTROL = 2, TINC = .01, TMAX = 10.,
OUTRATE = 10, PRATE = 10, SIMULATE
```

In the example, the simulation would run for 10 hours. Plotted output would occur every .1 hour, (10* .01), and printed output would occur every 1. hour (10* 10* .01).

PRINT VARIABLES

This program command allows up to ten variables to be specified for printing under option 5 of the PRINT CONTROL. This command is followed by from one to ten state, rate, or variable names separated by delimiters. This command wipes out all previously stored PRINT VARIABLES names.

Example 3.4-2:

```
PRINT VARIABLES = MA HS, E  TS, VDELO
```

3.5 PLOT DESIGNATION COMMANDS

PRINTER PLOTS
PLOT OFF

The above program commands allow the plotted output to be turned on or off. The default condition is PLOT OFF. It is therefore necessary to include the PRINTER PLOTS command before requesting any analysis from which plots are desired. The PLOT OFF and PRINTER PLOTS commands can be issued between simulation requests if it is desired to omit the plotting of certain analysis results.

```
DISPLAY1
DISPLAY2
DISPLAY3
DISPLAY4
DISPLAY5
DISPLAY6
```

These program commands may be used to define the quantities to be displayed by lineprinter plots for simulation calculations. These commands must be issued before the simulation analysis is requested. From one to five plots may be specified per display. Each plot is specified by stating the dependent variable and the independent variable separated by the letters VS. If desired, the independent and dependent axis scale ranges can also be specified. The independent scale range is specified by the word XRANGE followed by the minimum and maximum values for this scale. The dependent scale similarly is specified by the word YRANGE. If scale ranges are not specified, values will be used that span the given data.

```
SI MANUAL SCALES
SI AUTO SCALES (Default Condition)
```

The SI MANUAL SCALES command allows the plotted output requested by the DISPLAY commands to be plotted on manual scales specified by the YRANGE and XRANGE commands. The SI AUTO SCALES command can be used to return plotting to the automatic scaling mode. Auto scales are selected so that they span each plotted quantity. The auto scale option is the default used until manual scales are requested. The PRINTER PLOTS command is also required to obtain plots.

Example 3.5-1:

```
SI MANUAL SCALES, PRINTER PLOTS
DISPLAY1
WV2WD, VS, TIME, YRANGE = 10,40
P1 PD, VS, TIME, YRANGE = 0,1000
P2 PD, VS, TIME, YRANGE = 0,1000
DISPLAY2
P2 IV, VS, TIME
RE2BA, VS, TIME
RE1LO, VS, TIME
DISPLAY3
P1 PD, VS, P2 PD, YRANGE =0,1000, XRANGE = 0,1000
```

TITLE

The TITLE command allows a title to be placed on all plotted output. Up to 74 characters may follow the delimiter after the TITLE command. The TITLE command may be changed before each analysis. Once defined, the title remains in effect until a new title is entered.

Example 3.5-2:

```
TITLE = BATTERY TEST MODEL
```

3.6 ITERATION AND DIAGNOSTIC CONTROL

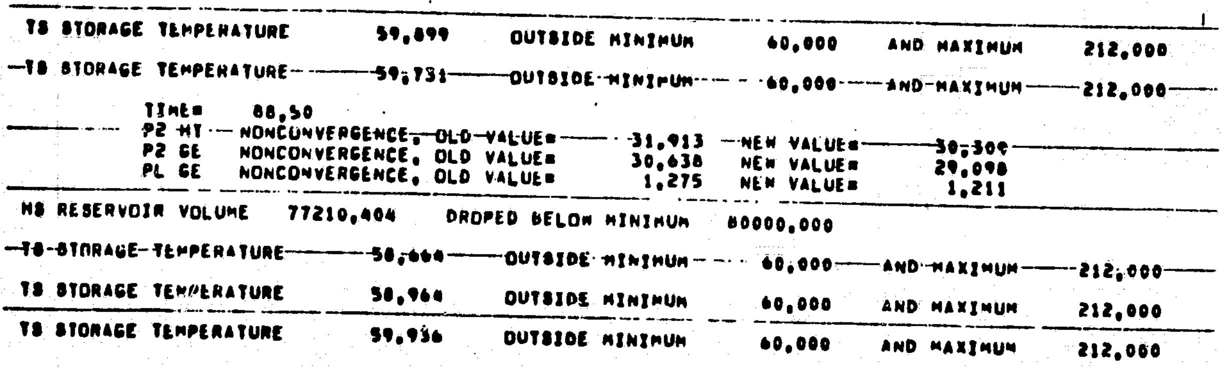
There are three built-in parameters in any SIMWEST model: CYCLES, DLINES and RESET. These parameters are specified similar to component parameters using the PARAMETER VALUES command.

CYCLES controls the number of iterations through the model to obtain steady state. If CYCLES < 1. then only one pass is made through the model. If CYCLES is a positive integer then the maximum number of iterations through the model is equal to CYCLES + 1. If cycles is positive, but not an integer, then the maximum number of iterations is equal to the smallest integer value exceeding cycles. A maximum of 20 iterations are permitted per time step. Most of the models tested require no more than six iterations per time step to attain steady state. A complex model with cascaded logic components may require more.

Each of the model output variables are monitored each pass for convergence. If all of the outputs are converged within 3% of their previous values, then one final pass is made through the model. Otherwise, all variables exceeding 5% of their previous value are printed out after the last iteration.

Since output statistics are only updated the last iteration, some of the variables printed indicating nonconvergence are statistics, and as such should be ignored.

DLINES controls the amount of convergence related printout as well as the amount of diagnostic printout from the library components. If DLINES > 0 then the total number of diagnostics is limited to DLINES. Figure 3.6 shows a typical section of diagnostic printout using DLINES > 0. If DLINES < 0 then only library component diagnostics are printed with no more than - DLINES of output. Typically, DLINES = 50 is sufficient to catch most simulation errors per run.



```
 TS STORAGE TEMPERATURE        59,899     OUTSIDE MINIMUM       60,000     AND MAXIMUM      212,000
 TS STORAGE TEMPERATURE        59,731     OUTSIDE MINIMUM       60,000     AND MAXIMUM      212,000
          TIME=    88,50
          P2 MT   NONCONVERGENCE, OLD VALUE=        31,913   NEW VALUE=        30,309
          P2 GE   NONCONVERGENCE, OLD VALUE=        30,638   NEW VALUE=        29,098
          PL GE   NONCONVERGENCE, OLD VALUE=         1,275   NEW VALUE=         1,211
 HS RESERVOIR VOLUME   77210,404    DROPED BELOW MINIMUM   80000,000
 TS STORAGE TEMPERATURE        58,664     OUTSIDE MINIMUM       60,000     AND MAXIMUM      212,000
 TS STORAGE TEMPERATURE        58,964     OUTSIDE MINIMUM       60,000     AND MAXIMUM      212,000
 TS STORAGE TEMPERATURE        59,936     OUTSIDE MINIMUM       60,000     AND MAXIMUM      212,000
```

FIGURE 3.6 TYPICAL DIAGNOSTIC OUTPUT

PRECEDING PAGE BLANK NOT FILMED

RESET controls the initialization value for the random number generators if several simulations are run back to back. If RESET >0 (Default) then the same random numbers are used for each simulation. If RESET ≤ 0 then the random numbers at the start of each simulation are obtained from the last value at the end of the previous simulation.

3.7 DEFINE COMMANDS

DEFINE STATES
DEFINE RATES
DEFINE PARAMETERS
DEFINE VARIABLES

These program commands may be used to replace model generated names by user defined alphanumeric names for system states, rates, parameters, and variables. (State variable derivatives, (Rates), are generated as R1, R2, ... for all models. R1, R2, ... refer to the rates of the first, second, ... states respectively.) If it is desired to replace these machine generated names with other names, the DEFINE command may be used to substitute any eight character name of the analyst's choosing. These names are associated with the corresponding numeric quantities located in the labeled commons /CX/, /CXDOT/, /CP/, and /CV/. The appropriate location for each quantity is printed out along with the quantity name prior to each simulation. Each of these commands is followed by phrases containing the location numeric followed by an alphanumeric name with one to eight characters the first of which must be alphabetic.

Example 3.7:

```
DEFINE STATES
1 = PRESSURE, 2 = STROKE, 5 = VELOCITY, 7 = ANGLE

DEFINE PARAMETERS
5 = MASS, 35 = DCT AREA
DEFINE VARIABLES, 1 = T OUTLET, 2 = LIQ H2O
```

Note that the program commands, numeric values and alphanumeric names must be separated by delimiters which are: comma [,], equals [=], left parentheses [(], right parenthesis [ )], or three or more consecutive spaces.

3.8 FUNCTION SCAN COMMANDS

SCAN1
SCAN2

These program commands initiate the calculation of general algebraic functions of one or two independent variables. Associated with these commands are the program names and values

1. DEPEN = dependent variable
2. INDEP1 = 1st independent variable
3. INDEP2 = 2nd independent variable
4. START1 = starting point of 1st independent variable
5. STOP1 = stopping point of 1st independent variable
6. START2 = starting point of 2nd independent variable
7. DELTA2 = increment of 2nd independent variable
8. CURVES2 = number of 2nd independent variable values

which specify the dependent and independent variables and scan ranges of these quantities. These quantities must be set to their desired values, before requesting the general algebraic function evaluation. If a single function is requested, i.e., SCAN1, only items 1, 2, 4 and 5 need be specified.

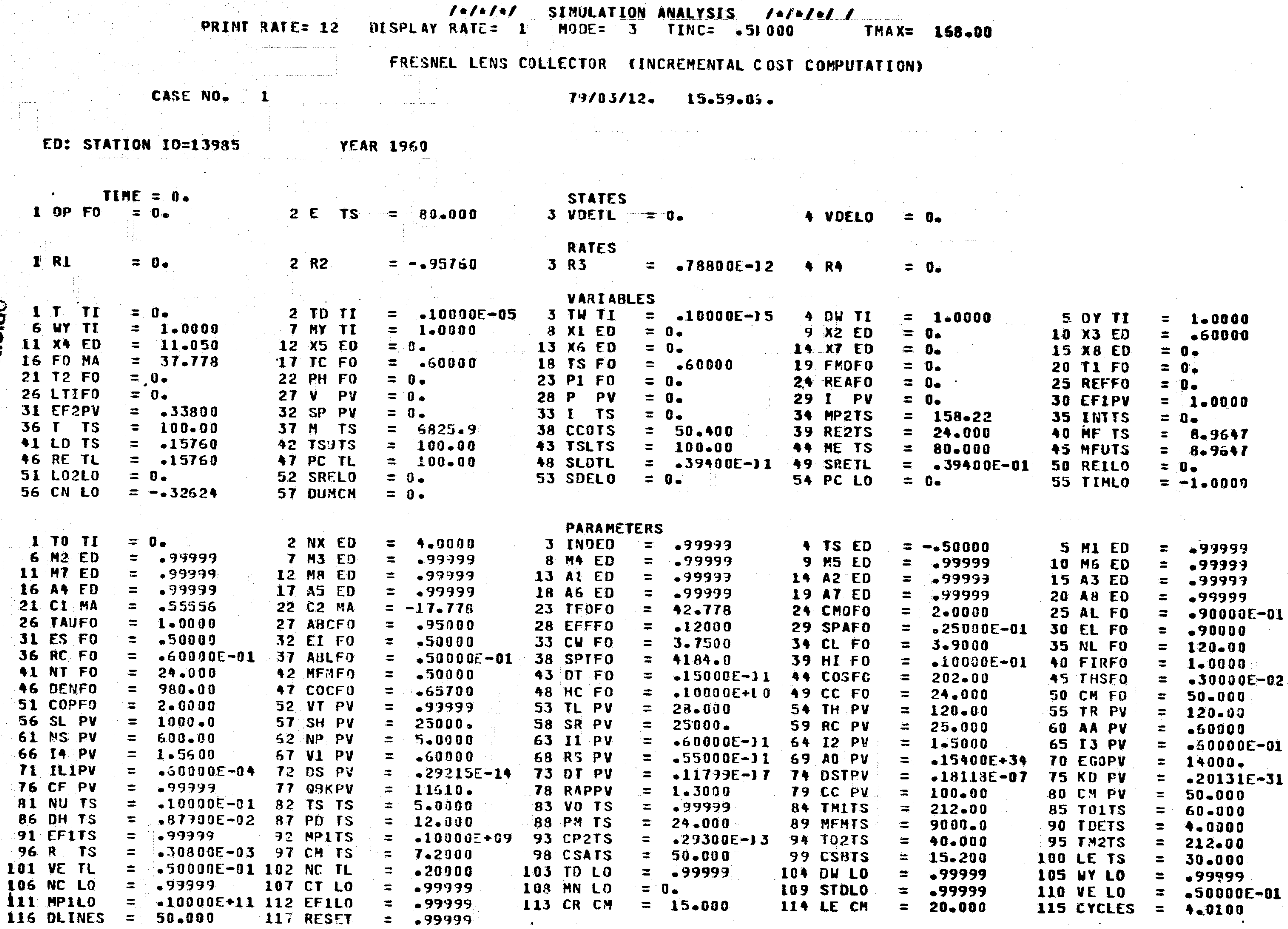
Example 3.8:

```
    DEPEN = I  PV, INDEP1 = ST1SO, INDEP2 = TC FP, START1 = 0
    STOP1 = 3000, START2 = 20, DELTA2 = 30, CURVES2 = 4
    SCAN2
```

In this example, the output current of a photovoltaic array, I PV, is calculated as a function of solar insolation, ST1SO, and cell temperature, TC FP (See Example 9.2).

3.9 EXAMPLE OUTPUT

Figure 3.9 shows a sample of the output print format generated using PRINT CONTROL = 3. This sample is taken from the Fresnel Lens Collector Model described in Section 9.3, which is a simple model. At each print time the output quantities are indexed by number and component name as they occur in the model. For example, first all the variables for component TI are printed, then all variables for component ED, etc. The parameter values at time = 0 show both the input values and the default parameters. After T = 0 only the states, rates, and output variables are printed. Since all the model connection variables and output variables are printed, this mode is especially valuable for program debugging and analysis at a fixed time. The printer plots, samples of which are shown in Sections 8 and 9 are useful for monitoring the time behavior of critical parameters such as energy in storage and percent of load delivered by storage.



```
                          /*/*/*/   SIMULATION ANALYSIS   /*/*/*/ /
           PRINT RATE= 12   DISPLAY RATE=  1   MODE=  3   TINC=  .51000        TMAX=  168.00

                                 FRESNEL LENS COLLECTOR  (INCREMENTAL COST COMPUTATION)

         CASE NO.   1                                   79/03/12.   15.59.05.

    ED: STATION ID=13985         YEAR 1960

        TIME = 0.                                       STATES
  1 OP FO    = 0.            2 E   TS    =  80.000     3 VDETL    = 0.         4 VDELO    = 0.

                                                        RATES
  1 R1       = 0.            2 R2        = -.95760     3 R3     =  .78800E-12  4 R4       = 0.

                                                        VARIABLES
  1 T  TI    = 0.            2 TD TI   =  .10000E-05   3 TW TI  =  .10000E-15  4 DW TI   =  1.0000      5 DY TI   =  1.0000
  6 WY TI    =  1.0000       7 MY TI   =  1.0000       8 X1 ED  = 0.           9 X2 ED   = 0.          10 X3 ED   =  .60000
 11 X4 ED    =  11.050      12 X5 ED   = 0.           13 X6 ED  = 0.          14 X7 ED   = 0.          15 X8 ED   = 0.
 16 FO MA    =  37.778      17 TC FO   =  .60000      18 TS FO  =  .60000     19 FMDFO   = 0.          20 T1 FO   = 0.
 21 T2 FO    = 0.           22 PH FO   = 0.           23 P1 FO  = 0.          24 REAFO   = 0.          25 REFFO   = 0.
 26 LTIFO    = 0.           27 V  PV   = 0.           28 P  PV  = 0.          29 I  PV   = 0.          30 EF1PV   =  1.0000
 31 EF2PV    =  .33800      32 SP PV   = 0.           33 I  TS  = 0.          34 MP2TS   =  158.22     35 INTTS   = 0.
 36 T  TS    =  100.00      37 M  TS   =  6825.9      38 CCOTS  =  50.400     39 RE2TS   =  24.000     40 MF TS   =  8.9647
 41 LD TS    =  .15760      42 TSUTS   =  100.00      43 TSLTS  =  100.00     44 ME TS   =  80.000     45 MFUTS   =  8.9647
 46 RE TL    =  .15760      47 PC TL   =  100.00      48 SLDTL  =  .39400E-11 49 SRETL   =  .39400E-01 50 RE1LO   = 0.
 51 LO2LO    = 0.           52 SRELO   = 0.           53 SDELO  = 0.          54 PC LO   = 0.          55 TIMLO   = -1.0000
 56 CN LO    = -.32624      57 DUMCM   = 0.

                                                        PARAMETERS
  1 TO TI    = 0.            2 NX ED   =  4.0000       3 INDED  =  .99999      4 TS ED   = -.50000      5 M1 ED   =  .99999
  6 M2 ED    =  .99999       7 M3 ED   =  .99999       8 M4 ED  =  .99999      9 M5 ED   =  .99999     10 M6 ED   =  .99999
 11 M7 ED    =  .99999      12 M8 ED   =  .99999      13 A1 ED  =  .99999     14 A2 ED   =  .99999     15 A3 ED   =  .99999
 16 A4 ED    =  .99999      17 A5 ED   =  .99999      18 A6 ED  =  .99999     19 A7 ED   =  .99999     20 A8 ED   =  .99999
 21 C1 MA    =  .55556      22 C2 MA   = -17.778      23 TFOFO  =  42.778     24 CMOFO   =  2.0000     25 AL FO   =  .90000E-01
 26 TAUFO    =  1.0000      27 ABCFO   =  .95000      28 EFFFO  =  .12000     29 SPAFO   =  .25000E-01 30 EL FO   =  .90000
 31 ES FO    =  .50000      32 EI FO   =  .50000      33 CW FO  =  3.7500     34 CL FO   =  3.9000     35 NL FO   =  120.00
 36 RC FO    =  .60000E-01  37 ABLFO   =  .50000E-01  38 SPTFO  =  4184.0     39 HI FO   =  .10000E-01 40 FIRFO   =  1.0000
 41 NT FO    =  24.000      42 MFMFO   =  .50000      43 DT FO  =  .15000E-11 44 COSFC   =  202.00     45 THSFO   =  .30000E-02
 46 DENFO    =  980.00      47 COCFO   =  .65700      48 HC FO  =  .10000E+10 49 CC FO   =  24.000     50 CM FO   =  50.000
 51 COPFO    =  2.0000      52 VT PV   =  .99999      53 TL PV  =  28.000     54 TH PV   =  120.00     55 TR PV   =  120.00
 56 SL PV    =  1000.0      57 SH PV   =  25000.      58 SR PV  =  25000.     59 RC PV   =  25.000     60 AA PV   =  .60000
 61 NS PV    =  600.00      62 NP PV   =  5.0000      63 I1 PV  =  .60000E-11 64 I2 PV   =  1.5000     65 I3 PV   =  .60000E-01
 66 I4 PV    =  1.5600      67 V1 PV   =  .60000      68 RS PV  =  .55000E-11 69 AO PV   =  .15400E+34 70 EGOPV   =  14000.
 71 IL1PV    =  .60000E-04  72 DS PV   =  .29215E-14  73 DT PV  =  .11799E-17 74 DSTPV   =  .18118E-07 75 KD PV   =  .20131E-31
 76 CF PV    =  .99999      77 QBKPV   =  11610.      78 RAPPV  =  1.3000     79 CC PV   =  100.00     80 CM PV   =  50.000
 81 NU TS    =  .10000E-01  82 TS TS   =  5.0000      83 VO TS  =  .99999     84 TM1TS   =  212.00     85 TO1TS   =  60.000
 86 DH TS    =  .87700E-02  87 PD TS   =  12.000      88 PM TS  =  24.000     89 MFMTS   =  9000.0     90 TDETS   =  4.0000
 91 EF1TS    =  .99999      92 MP1TS   =  .10000E+09  93 CP2TS  =  .29300E-13 94 TO2TS   =  40.000     95 TM2TS   =  212.00
 96 R  TS    =  .30800E-03  97 CM TS   =  7.2000      98 CSATS  =  50.000     99 CSBTS   =  15.200    100 LE TS   =  30.000
101 VE TL    =  .50000E-01 102 NC TL   =  .20000     103 TD LO  =  .99999    104 DW LO   =  .99999    105 WY LO   =  .99999
106 NC LO    =  .99999     107 CT LO   =  .99999     108 MN LO  = 0.         109 STDLO   =  .99999    110 VE LO   =  .50000E-01
111 MP1LO    =  .10000E+11 112 EF1LO   =  .99999     113 CR CM  =  15.000    114 LE CM   =  20.000    115 CYCLES  =  4.0100
116 DLINES   =  50.000     117 RESET   =  .99999
```

FIGURE 3.9 SAMPLE PRINTER OUTPUT

4.0 JOB CONTROL PROCEDURES

This chapter describes the job control procedures used on the BCS computer network MAINSTREAM-EKS to execute SIMWEST and to develop library components. The user at other CDC computer installations should consult his maintenance organization for extensions and variations to these procedures.

4.1 JOB ENVIRONMENT

The CDC version of SIMWEST was developed for a user with primary computer access via a communications terminal. The job control procedures described below are contained within the user's procedure file PROFIL stored as one of the user account permanent files. These procedures are for a CDC installation using the Network Operating System (NOS) and can be modified for a KRONOS operating system. They enable a user to easily edit files, compile Fortran source on-line, and to submit jobs in interactive or remote job entry modes. Job output may be directed to the terminal, stored in a user controlled file, or disposed to a remote line printer, depending on the procedure and user requirements. These procedures enable the user to minimize development time in constructing system models, and to minimize computer resources when performing system simulations.

4.2 SIMWEST PROGRAM EXECUTION

Four procedures have been developed for constructing SIMWEST system models and running simulations:

- EASYM - Interactive execution of the model generation program with output to user file EASYOUT

- EASY - Batch execution of the model generation and analysis programs with output to an RJE terminal printer

- EASYA - Batch execution of the model generation and analysis programs with output to user files EASYOUT, ANALOUT, and PLOTOUT

- EASYB - Similar to EASY, but includes the capability to input TMY tape environmental inputs (See Section 7.8)

The following is a summary of the usage of these procedures.

EASYM

This procedure is used in timeshare mode to generate the user's Fortran model and system schematic to verify the system model connections. The job command card is

```
CALL (,EASYM(DATAM=MODEL)
```

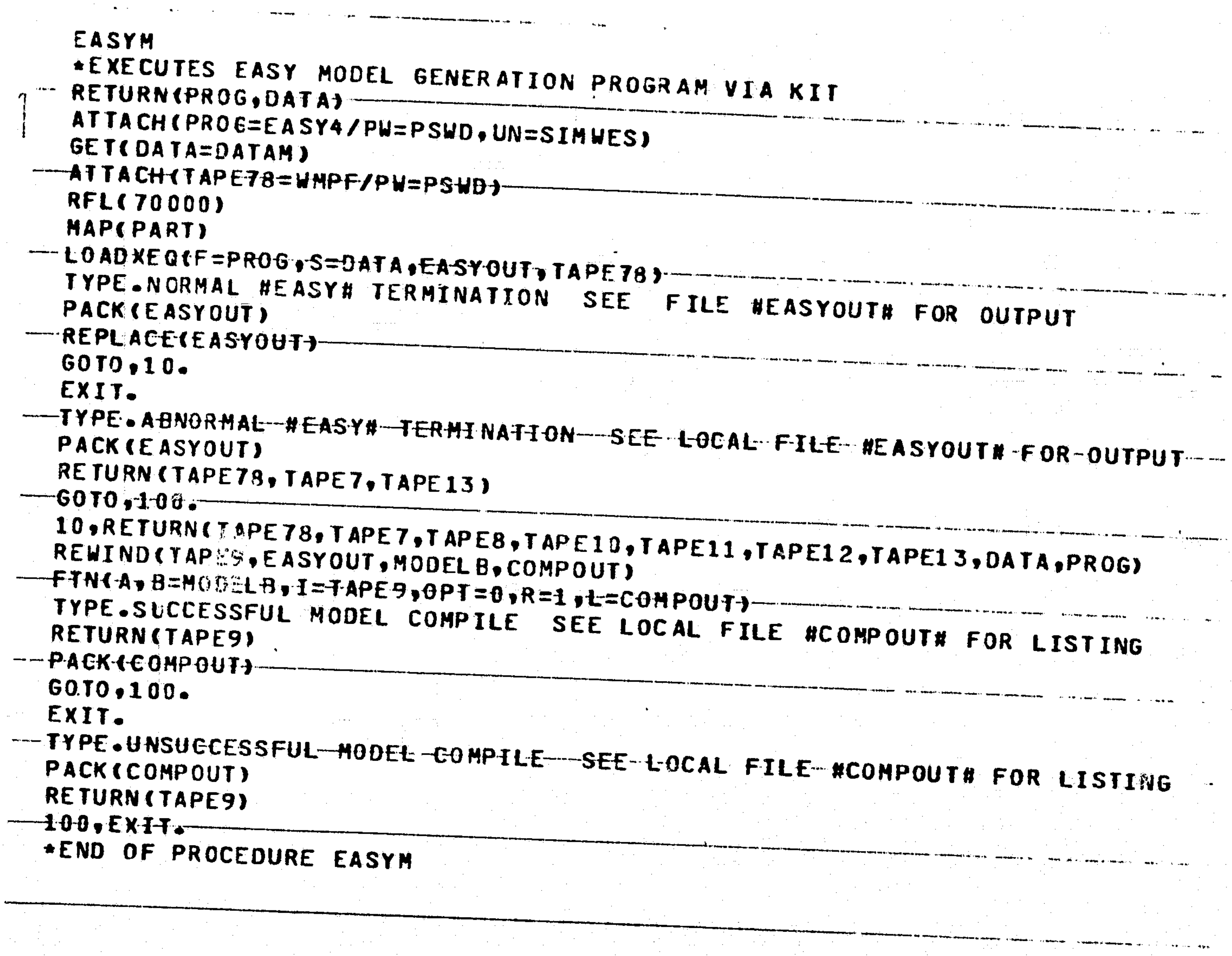
where MODEL is the user's model generation input file. The output file EASYOUT contains a readback of the input file with error diagnostics, the system model schematic, and the model data requirements list. A compilation listing of the Fortran model is also output to a local or temporary file COMPOUT. Figure 4.2-1 is a listing of the job control cards for EASYM.

EASY

This procedure is used to launch a SIMWEST batch job from a terminal. The job command card is

```
CALL (,EASY(DATAM=MODEL,DATAA=SIMUL)
```

where MODEL is the user's model generation input file and SIMUL is the user's input file for the simulation program. (It is not necessary to run the model generation program each time, but this is normally done since it is inexpensive



```
EASYM
*EXECUTES EASY MODEL GENERATION PROGRAM VIA KIT
RETURN(PROG,DATA)
ATTACH(PROG=EASY4/PW=PSWD,UN=SIMWES)
GET(DATA=DATAM)
ATTACH(TAPE78=WMPF/PW=PSWD)
RFL(70000)
MAP(PART)
LOADXEQ(F=PROG,S=DATA,EASYOUT,TAPE78)
TYPE.NORMAL #EASY# TERMINATION  SEE  FILE #EASYOUT# FOR OUTPUT
PACK(EASYOUT)
REPLACE(EASYOUT)
GOTO,10.
EXIT.
TYPE.ABNORMAL #EASY# TERMINATION  SEE LOCAL FILE #EASYOUT# FOR OUTPUT
PACK(EASYOUT)
RETURN(TAPE78,TAPE7,TAPE13)
GOTO,100.
10,RETURN(TAPE78,TAPE7,TAPE8,TAPE10,TAPE11,TAPE12,TAPE13,DATA,PROG)
REWIND(TAPE9,EASYOUT,MODELB,COMPOUT)
FTN(A,B=MODELB,I=TAPE9,OPT=0,R=1,L=COMPOUT)
TYPE.SUCCESSFUL MODEL COMPILE  SEE LOCAL FILE #COMPOUT# FOR LISTING
RETURN(TAPE9)
PACK(COMPOUT)
GOTO,100.
EXIT.
TYPE.UNSUCCESSFUL MODEL COMPILE  SEE LOCAL FILE #COMPOUT# FOR LISTING
PACK(COMPOUT)
RETURN(TAPE9)
100,EXIT.
*END OF PROCEDURE EASYM
```

FIGURE 4.2-1 JOB CONTROL PROCEDURE 'EASYM'

and provides a complete listing of the model associated with a simulation run.) Figure 4.2-2 gives a listing of the control cards for EASY. The job dayfile is also output to user file DAYFLE so that a user can verify whether a job ran successfully prior to receiving the lineprinter output.

The control cards shown in Figure 4.2-2 are for the general case in which the user maintains and updates his own component library files WMPF and WMCOMP. If the user is not developing new library routines, then this procedure can be simplified by replacing the control cards

```
ATTACH(WMCOMP/PW=PSWD)
REWIND(ULIB,MAPF)
LIBGEN(F=WMCOMP,P=ULIB)
RETURN(WMCOMP)
```

by the card

```
ATTACH(ULIB=SIMLIB/PW=PSWD).
```

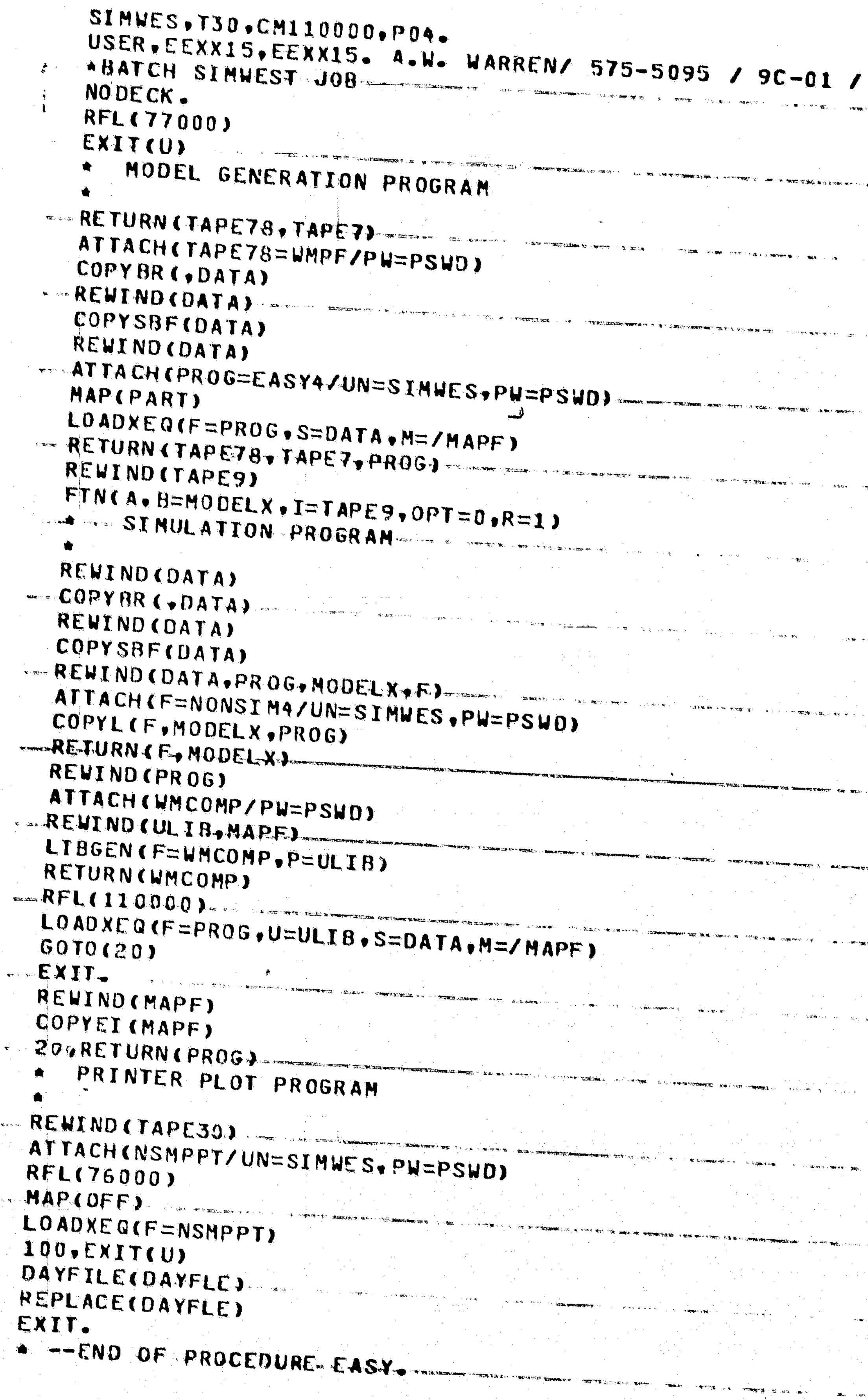
The permanent library file SIMLIB is easily generated by using the above four control cards plus

```
DEFINE(SIMLIB/PW=PSWD)
COPYEI(ULIB,SIMLIB)
RETURN(ULIB,SIMLIB).
```

EASYA

This procedure is used to launch a SIMWEST batch job from a terminal with output to user files for rapid inspection of output and subsequent printout as desired. The job command card is

```
CALL(,EASYA(DATAM=MODEL,DATAA=SIMUL)
```



```
SIMWES,T30,CM110000,P04.
USER,EEXX15,EEXX15. A.W. WARREN/ 575-5095 / 9C-01 /
*BATCH SIMWEST JOB
NODECK.
RFL(77000)
EXIT(U)
*  MODEL GENERATION PROGRAM
*
RETURN(TAPE78,TAPE7)
ATTACH(TAPE78=WMPF/PW=PSWD)
COPYBR(,DATA)
REWIND(DATA)
COPYSBF(DATA)
REWIND(DATA)
ATTACH(PROG=EASY4/UN=SIMWES,PW=PSWD)
MAP(PART)
LOADXEQ(F=PROG,S=DATA,M=/MAPF)
RETURN(TAPE78,TAPE7,PROG)
REWIND(TAPE9)
FTN(A,B=MODELX,I=TAPE9,OPT=0,R=1)
*  SIMULATION PROGRAM
*
REWIND(DATA)
COPYBR(,DATA)
REWIND(DATA)
COPYSBF(DATA)
REWIND(DATA,PROG,MODELX,F)
ATTACH(F=NONSIM4/UN=SIMWES,PW=PSWD)
COPYL(F,MODELX,PROG)
RETURN(F,MODELX)
REWIND(PROG)
ATTACH(WMCOMP/PW=PSWD)
REWIND(ULIB,MAPF)
LIBGEN(F=WMCOMP,P=ULIB)
RETURN(WMCOMP)
RFL(110000)
LOADXEQ(F=PROG,U=ULIB,S=DATA,M=/MAPF)
GOTO(20)
EXIT.
REWIND(MAPF)
COPYEI(MAPF)
20,RETURN(PROG)
*  PRINTER PLOT PROGRAM
*
REWIND(TAPE30)
ATTACH(NSMPPT/UN=SIMWES,PW=PSWD)
RFL(76000)
MAP(OFF)
LOADXEQ(F=NSMPPT)
100,EXIT(U)
DAYFILE(DAYFLE)
REPLACE(DAYFLE)
EXIT.
* --END OF PROCEDURE EASY.
```

FIGURE 4.2-2 JOB CONTROL PROCEDURE 'EASY'

where MODEL and SIMUL are the model generation and simulation input file names. The output of the model generation program and the model Fortran listings are contained in file EASYOUT. Similarly, ANALOUT contains the output of the simulation program, PLOTOUT contains the output of the printer plot program, and DAYFLE contains the job dayfile. The user is cautioned to print out results of interest before reexecuting this procedure since only the most recent program output is saved on the output files. Procedure LIST may be used for this purpose (see Section 4.3).

Figure 4.2-3 shows the control cards for EASYA. If the model generation or simulation program load and execute step aborts, a load map is also copied onto the respective output file for error traceback and debugging.

EASYB

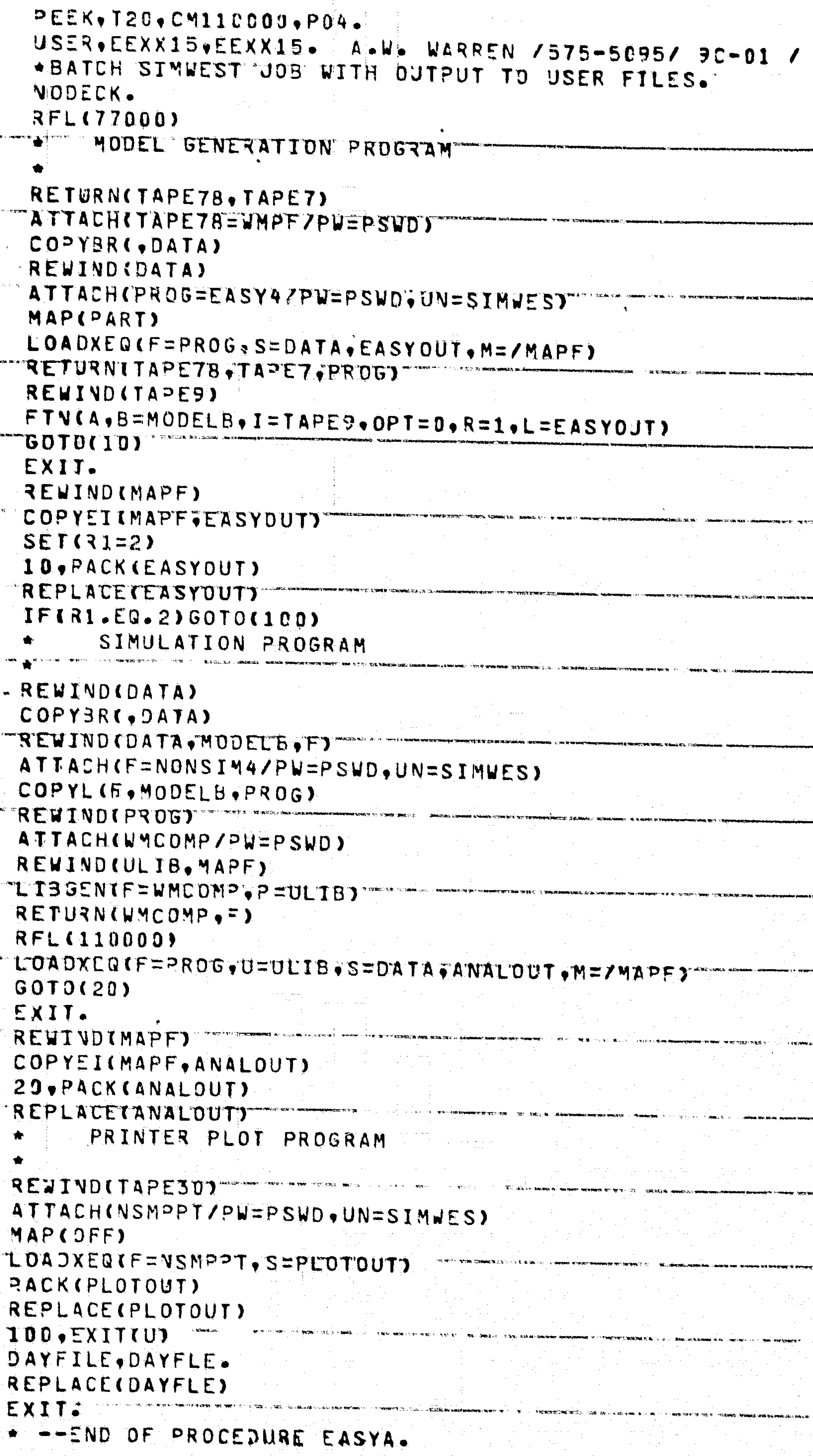
This procedure launches a SIMWEST batch job using the TMY tape for environmental inputs, i.e., whenever the user's system model includes an ED component. The TMY tape is blocked into logical records, each record containing environmental data for a 24 hour period. There are 26 stations on the tape with each station containing a standard meteorological year of 365 days. Thus the user must edit EASYB whenever a different station or data record length is required. A procedure TMYRD is used to select and copy that portion of the TMY tape to be input for subsequent analyses. See Figure 4.2-4. The command card to launch a SIMWEST job using EASYB is

```
CALL(,EASYB(DATAM=MODEL,DATAA=SIMUL)
```

where MODEL and SIMUL are the model generation and simulation input file names. Each EASYB job mounts the TMY tape and creates a local file TAPE1 for input to the simulation program. If many simulations are required using the same TMY input data, then TAPE1 can be saved and the TMYRD cards in EASYB replaced with

```
FILE(TAPE1,CM=YES,MBL=3168,FL=132,RB=24,RT=F,BT=K)
GET(TAPE1)
```

This eliminates creation of the input file TAPE1 for each separate SIMWEST job.



```
PEEK,T20,CM110000,P04.
USER,EEXX15,EEXX15.  A.W. WARREN /575-5095/ 9C-01 /
*BATCH SIMWEST JOB WITH OUTPUT TO USER FILES.
NODECK.
RFL(77000)
*    MODEL GENERATION PROGRAM
*
RETURN(TAPE78,TAPE7)
ATTACH(TAPE78=WMPF/PW=PSWD)
COPYBR(,DATA)
REWIND(DATA)
ATTACH(PROG=EASY4/PW=PSWD,UN=SIMWES)
MAP(PART)
LOADXEQ(F=PROG,S=DATA,EASYOUT,M=/MAPF)
RETURN(TAPE78,TAPE7,PROG)
REWIND(TAPE9)
FTN(A,B=MODELB,I=TAPE9,OPT=0,R=1,L=EASYOUT)
GOTO(10)
EXIT.
REWIND(MAPF)
COPYEI(MAPF,EASYOUT)
SET(R1=2)
10,PACK(EASYOUT)
REPLACE(EASYOUT)
IF(R1.EQ.2)GOTO(100)
*    SIMULATION PROGRAM
*
REWIND(DATA)
COPYBR(,DATA)
REWIND(DATA,MODELB,F)
ATTACH(F=NONSIM4/PW=PSWD,UN=SIMWES)
COPYL(F,MODELB,PROG)
REWIND(PROG)
ATTACH(WMCOMP/PW=PSWD)
REWIND(ULIB,MAPF)
LIBGEN(F=WMCOMP,P=ULIB)
RETURN(WMCOMP,F)
RFL(110000)
LOADXEQ(F=PROG,U=ULIB,S=DATA,ANALOUT,M=/MAPF)
GOTO(20)
EXIT.
REWIND(MAPF)
COPYEI(MAPF,ANALOUT)
20,PACK(ANALOUT)
REPLACE(ANALOUT)
*    PRINTER PLOT PROGRAM
*
REWIND(TAPE30)
ATTACH(NSMPPT/PW=PSWD,UN=SIMWES)
MAP(OFF)
LOADXEQ(F=NSMPPT,S=PLOTOUT)
PACK(PLOTOUT)
REPLACE(PLOTOUT)
100,EXIT(U)
DAYFILE,DAYFLE.
REPLACE(DAYFLE)
EXIT.
* --END OF PROCEDURE EASYA.
```

FIGURE 4.2-3 JOB CONTROL PROCEDURE 'EASYA'

```
TMYSIM,T20,CM11000C,P04.
USER,EEXX15,EEXX15.  A.W. WARREN /575-5095/ 9C-01 /
*BATCH SIMWEST JOB USING TMY TAPE INPUTS
NODECK.
RFL(77000)
*     MODEL GENERATION PROGRAM
*
RETURN(TAPE78,TAPE7)
ATTACH(TAPE78=WMPF/PW=PSWD)
COPYBR(,DATA)
REWIND(DATA)
COPYSBF(DATA)
REWIND(DATA)
ATTACH(PROG=EASY4/PW=PSWD,UN=SIMWES)
MAP(PART)
LOADXEQ(F=PROG,S=DATA,M=/MAPF)
RETURN(TAPE78,TAPE7,PROG)
REWIND(TAPE9)
FTN(A,B=MODELB,I=TAPE9,OPT=0,R=1)
*     PREPARE TMY TAPE INPUT FILE - TAPE1
*
RFL(30000)
GET(TMYRD/PW=PSWD,UN=SIMWES)
CALL(TMYRD(NSKIP=2920,NCOPY=9)
*     SIMULATION PROGRAM
*
REWIND(DATA)
COPYBR(,DATA)
REWIND(DATA)
COPYSBF(DATA)
REWIND(DATA,MODELB,F)
ATTACH(F=NONSIM4/PW=PSWD,UN=SIMWES)
COPYL(F,MODELB,PROG)
RETURN(F)
REWIND(PROG)
ATTACH(WMCOMP/PW=PSWD)
REWIND(ULIB,MAPF)
LIBGEN(F=WMCOMP,P=ULIB)
RETURN(WMCOMP)
RFL(110000)
LDSET(FILES=TAPE1)
LOADXEQ(F=PROG,U=ULIB,S=DATA,M=/MAPF)
GOTO(20)
EXIT.
REWIND(MAPF)
COPYEI(MAPF)
20,RETURN(PROG)
*     PRINTER PLOT PROGRAM
*
REWIND(TAPE30)
ATTACH(NSMPPT/PW=PSWD,UN=SIMWES)
MAP(OFF)
LOADXEG(F=NSMPPT)
100,EXIT(U)
DAYFILE,DAYFLE.
REPLACE(DAYFLE)
EXIT.
* --END OF PROCEDURE EASYB.
```

FIGURE 4.2-4 JOB CONTROL PROCEDURE 'EASYB'

The job command card to create input file TAPE1 is

```
CALL(TMYRD(NSKIP=N1,NCOPY=N2)
```

where N1 is the number of logical records to skip and N2 is the number of records to be copied from the TMY tape onto TAPE1. These parameters are specified using

```
N1 = 365 * (NSTATION -1) + DSTART -1
N2 = DEND - DSTART +1
```

where

NSTATION = station number of the data file as shown in Table 7.8
DSTART = first or start day of the desired data file
DEND = last or end day of the desired file.

For example, if a user wanted TMY inputs for April (DSTART = 91 and DEND = 120) at Albuquerque, New Mexico (NSTATION = 13), then N1 = 4470 and N2 = 30. Thus, procedure EASYB would be edited so that the TMYRD call statement reads

```
CALL(TMYRD(NSKIP=4470,NCOPY=30)
```

4.3 FILE MAINTENANCE AND LIBRARY UPDATES

This section describes frequently used procedures for modifying and developing SIMWEST library components. A terminal based user would first create his Fortran component subroutines and any associated subroutines as records within a user file of source routines. The procedures FORMOD and FORMODG are then used to compile these routines and merge the relocatables produced onto the component library WMCOMP. The FILOAD procedure is then called to update the component name list of input and output variables on WMPF. (See Section 6.0

for component coding conventions and preparation of input for the FILOAD program.) Usage of the procedures FORMOD, FORMODG, FILOAD, and LIST are described below.

FORMOD

This procedure is used in timeshare mode to compile a multi-record Fortran source file and merge the object code onto a specified file of relocatable records. The job command card is

```
CALL(,FORMOD(SOURCE=MYFILE,OLD=RELFLE)
```

where MYFILE is the permanent file name of the user's source code and RELFLE is the file name of the relocatable code.

Figure 4.3-1 shows the control cards for FORMOD. If the source code has fatal errors, diagnostics are printed out on the terminal printer. Otherwise the source code listings are disposed to a lineprinter. If no relocatable file is specified, or if RELFLE cannot be found, then the object code is copied onto permanent file 'OLD'.

FORMODG

This procedure is similar to FORMOD except it enables compilation of specified records from a multi-record source file. The job command card is

```
CALL(,FORMODG(SOURCE=MYFILE,OLD=RELFLE)
```

The terminal then prompts the user for the record numbers of the source code to be compiled, i.e., the terminal prints:

```
FORMOD
* COMPILES AND MERGES EXTENDED FORTRAN PROGRAMS
*
* ACCEPTS MULTIRECORD FILE OF SOURCE CODE --SOURCE--
* PLACES COMPILER LISTING ON --LIST--
* USES CDC EXTENDED COMPILER  FTN4.6
*
REWIND(LIST,TEXT)
GET(S1=SOURCE)
GOTO,5.
EXIT.
TYPE. CAN"T FIND #SOURCE#
GOTO,60.
5,REWIND(S1)
PACK(S1)
RFL(77000)
FTN(A,I=S1,B=TEXT,L=LIST,R=3,OPT=1)
TYPE.SUCCESSFUL FTN COMPILATION
SET(R2=0)     ZERO FOR NO FTN ERRORS
CALL(,LIST)
GOTO,10.
EXIT.
SET(R2=1)  INDICATE FORTRAN ERRORS
TYPE. FORTRAN ERRORS DETECTED BY EXTENDED COMPILER
RETURN(TEXT,S1)
REWIND(LIST)
ERRLIST,LIST.
TYPE.TO SEND LISTING TO LINEPRINTER% CALL(,LIST)
10,RETURN(S1)
IF(R2.EQ.1)GOTO,60.   SKIP MERGE IF ERRORS
*
*MERGES NEW TEXT RECORD  --TEXT-- WITH EXISTING MULTIRECORD
*  TEXT FILE CONTAINING A RECORD HAVING SAME NAME
*  OF SUBROUTINE OR PROGRAM.  TEXT FILE IS --OLD--
*
ATTACH(S1=OLD/PW=PSWD,M=W)
GOTO,30.
EXIT.
REWIND(TEXT)
DEFINE(S2=OLD/PW=PSWD)
TYPE.OLD FILE COULD NOT BE FOUND
TYPE.     BINARIES ARE PLACED ON PERMANENT FILE #OLD#
COPYEI(TEXT,S2)
GOTO,20.
EXIT.
TYPE. OLD FILE BUSY
TYPE. BINARIES ARE ON LOCAL FILE #TEXT#
GOTO(60)
30,REWIND(S1,OLD,TEXT)
COPYLM(S1,TEXT,OLD,,A)
REWIND(OLD,S1)
COPYEI(OLD,S1)
TYPE.SUCCESSFUL MERGE INTO #OLD# FILE
20,RETURN(S1,TEXT,OLD,S2)
60,RFL(20000)
*--END OF PROCEDURE FORMOD
```

FIGURE 4.3-1 JOB CONTROL PROCEDURE 'FORMOD'

```
INPUT RECORD NUMBERS
I>
```

The user enters the record numbers to be compiled followed by two carriage returns (CR), i.e.

```
I>    3,38   CR
I>           CR
```

This procedure will catalog the records to be compiled, compile the source code and merge the object code onto RELFLE, replacing old object code routines with those just created. Figure 4.3-2 shows the control cards for FORMODG.

FILOAD

This procedure is used in timeshare mode to create or modify input, output and table name and dimension data for SIMWEST library components. The job command card is

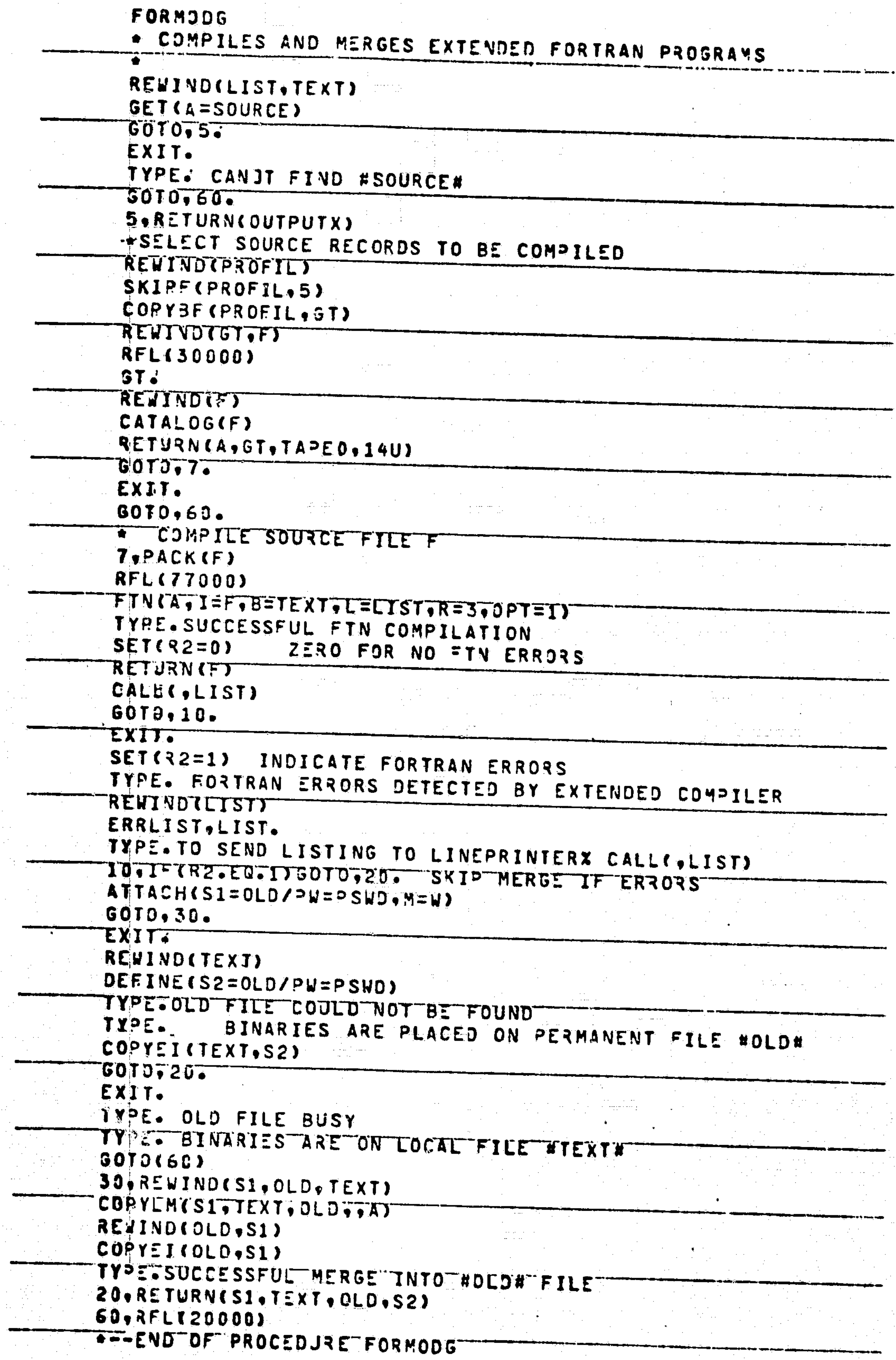
```
CALL(,FILOAD(DATA=NAMES)
```

where NAMES is the permanent file name of the input data for the FILOAD program. (See Section 6.2 for preparation of the input file.)

If the FILOAD program execution aborts, the user should check the format of the input data since exact spacing of the alphanumeric character inputs is required. Figure 4.3-3 shows the control cards for FILOAD.

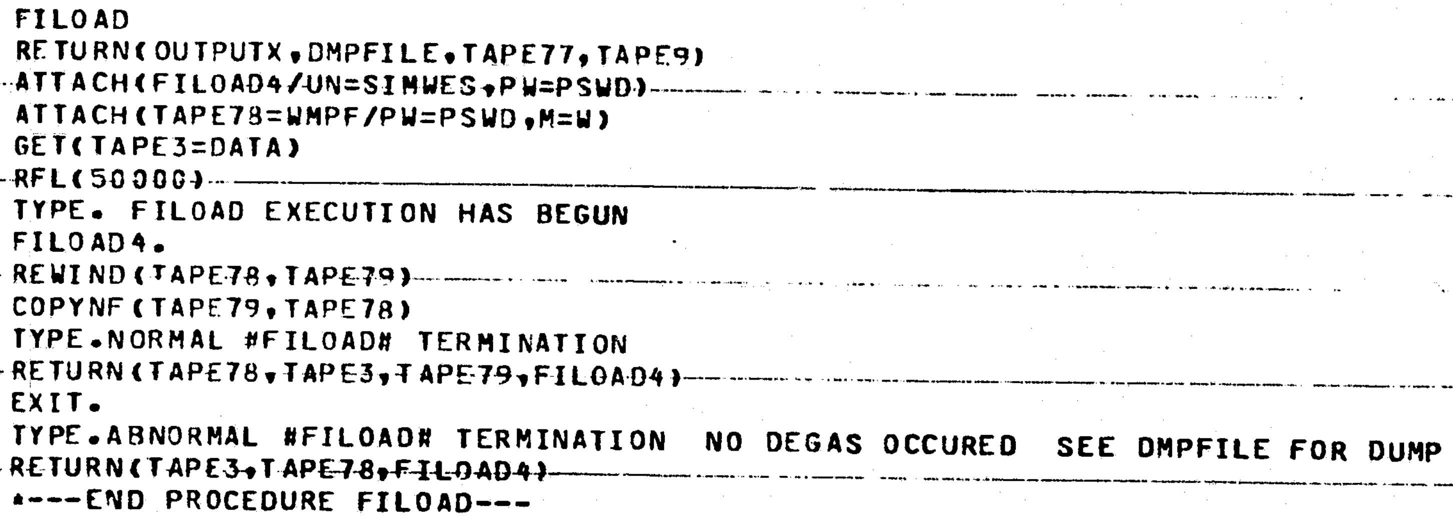
LIST

This procedure is used to dispose a local file to be printed. If the local file LOCAL has no printer control characters, then one uses the command cards:



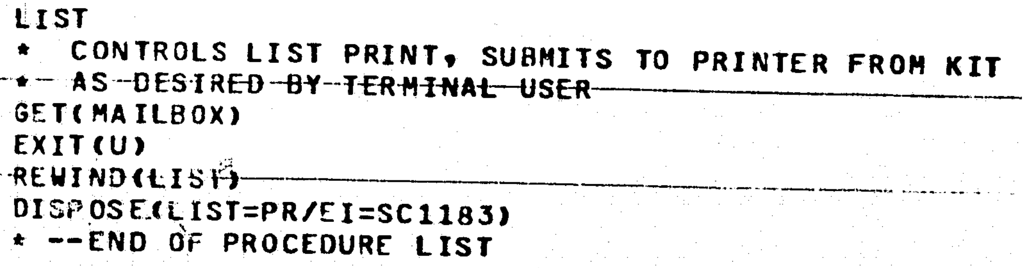
```
FORMODG
* COMPILES AND MERGES EXTENDED FORTRAN PROGRAMS
*
REWIND(LIST,TEXT)
GET(A=SOURCE)
GOTO,5.
EXIT.
TYPE. CAN]T FIND #SOURCE#
GOTO,60.
5,RETURN(OUTPUTX)
*SELECT SOURCE RECORDS TO BE COMPILED
REWIND(PROFIL)
SKIPF(PROFIL,5)
COPYBF(PROFIL,GT)
REWIND(GT,F)
RFL(30000)
GT.
REWIND(F)
CATALOG(F)
RETURN(A,GT,TAPE0,14U)
GOTO,7.
EXIT.
GOTO,60.
*  COMPILE SOURCE FILE F
7,PACK(F)
RFL(77000)
FTN(A,I=F,B=TEXT,L=LIST,R=3,OPT=1)
TYPE.SUCCESSFUL FTN COMPILATION
SET(R2=0)    ZERO FOR NO FTN ERRORS
RETURN(F)
CALL(,LIST)
GOTO,10.
EXIT.
SET(R2=1)  INDICATE FORTRAN ERRORS
TYPE. FORTRAN ERRORS DETECTED BY EXTENDED COMPILER
REWIND(LIST)
ERRLIST,LIST.
TYPE.TO SEND LISTING TO LINEPRINTERX CALL(,LIST)
10,IF(R2.EQ.1)GOTO,20.  SKIP MERGE IF ERRORS
ATTACH(S1=OLD/PW=PSWD,M=W)
GOTO,30.
EXIT.
REWIND(TEXT)
DEFINE(S2=OLD/PW=PSWD)
TYPE.OLD FILE COULD NOT BE FOUND
TYPE.    BINARIES ARE PLACED ON PERMANENT FILE #OLD#
COPYEI(TEXT,S2)
GOTO,20.
EXIT.
TYPE. OLD FILE BUSY
TYPE. BINARIES ARE ON LOCAL FILE #TEXT#
GOTO(60)
30,REWIND(S1,OLD,TEXT)
COPYLM(S1,TEXT,OLD,,A)
REWIND(OLD,S1)
COPYEI(OLD,S1)
TYPE.SUCCESSFUL MERGE INTO #OLD# FILE
20,RETURN(S1,TEXT,OLD,S2)
60,RFL(20000)
*--END OF PROCEDURE FORMODG
```

FIGURE 4.3-2 JOB CONTROL PROCEDURE 'FORMODG'



```
FILOAD
RETURN(OUTPUTX,DMPFILE,TAPE77,TAPE9)
ATTACH(FILOAD4/UN=SIMWES,PW=PSWD)
ATTACH(TAPE78=WMPF/PW=PSWD,M=W)
GET(TAPE3=DATA)
RFL(50000)
TYPE. FILOAD EXECUTION HAS BEGUN
FILOAD4.
REWIND(TAPE78,TAPE79)
COPYNF(TAPE79,TAPE78)
TYPE.NORMAL #FILOAD# TERMINATION
RETURN(TAPE78,TAPE3,TAPE79,FILOAD4)
EXIT.
TYPE.ABNORMAL #FILOAD# TERMINATION  NO DEGAS OCCURED  SEE DMPFILE FOR DUMP
RETURN(TAPE3,TAPE78,FILOAD4)
*---END PROCEDURE FILOAD---
```

FIGURE 4.3-3 JOB CONTROL PROCEDURE 'FILOAD'



```
LIST
*  CONTROLS LIST PRINT, SUBMITS TO PRINTER FROM KIT
*  AS DESIRED BY TERMINAL USER
GET(MAILBOX)
EXIT(U)
REWIND(LIST)
DISPOSE(LIST=PR/EI=SC1183)
* --END OF PROCEDURE LIST
```

FIGURE 4.3-4 JOB CONTROL PROCEDURE 'LIST'

```
COPYSBF(LOCAL,LIST)
CALL(,LIST)
```

To print files such as EASYOUT which contain printer control characters, it suffices to use the command

```
CALL(,LIST(LIST=EASYOUT)
```

Figure 4.3-4 shows the control cards for LIST. (See previous page.)

5.0 DIAGNOSTICS

Diagnostic messages are printed by both the model generation and the simulation program. In addition, individual library components generally have diagnostic printout asociated with them. The diagnostics associated with the model generation program are discussed in Section 2.0. Section 5.0 describes the diagnostics associated with the simulation program and the library components.

5.1 WARNING MESSAGES

One or more of the following warning messages will occur if the program encounters difficulty in interpreting analysis instructions or performing an analysis. These messages will be preceded by the flag:

*** WARNING ***

Message symbols xxx, zzz, or nnn are used to indicate phrases from the analysis description that are included as part of the warning message. The following messages are listed in alphabetical order:

1. A VALID PARAMETER NAME MUST PRECEDE THE NUMERIC VALUE nnn

   This message indicates that a valid parameter name was not identified preceding the numeric value nnn. Check for missing delimiters or misspelled parameter name.

2. xxx CAN'T BE SET EQUAL TO zzz. VALUE MUST BE NUMERIC.

   Check for missing numeric value or delimiters.

3. CAN'T IDENTIFY xxx AS A VALID PRINT VARIABLE

   Check spelling of xxx or for missing delimiters.

PRECEDING PAGE BLANK NOT FILMED

4. CAN'T IDENTIFY xxx VALUE WILL BE IGNORED

   This will result in not setting the quantity intended by xxx to its new value. Check for spelling of xxx or for missing delimiters.

5. CAN'T INTERPRET xxx

   The phrase xxx cannot be recognized as a valid program command, program name, or program value. Check spelling of xxx or for missing delimiters.

6. nnn EXCEEDS THE ALLOWABLE INDEX RANGE FOR xxx THIS QUANTITY WILL NOT BE DEFINED

   The number nnn was outside the allowable range of states, rates, variables, or parameters. Therefore, the name xxx cannot be assigned as a name for the nnnth state, rate, variable or parameter.

7. NON-ALPHA NAME ON THIS CARD --- xxx. WILL IGNORE THIS CARD.

   The table inputs routine expected an alphanumeric table name but encountered a numeric value on the data card printed. Check the sequence and number of tabular data cards to assure that they match those required by the model's tables and table input formats. See Section 3.1.2 for correct formats.

8. NON-NUMERIC DATA ON THIS CARD --- xxx. WILL READ NEXT TABLE

   The table input routine expected a numeric value but encountered an alphanumeric name on the data card printed. Check that the sequence and number of tabular data cards matches the model's tables and table input formats. See Section 3.1.2 for correct formats.

9. nnn PRIMARY and xxx SECONDARY INDEPENDENT VARIABLE POINTS EXCEEDS THE zzz WORD STORAGE LIMIT FOR THE FOLLOWING TABLE. SOME DATA WILL BE LOST.

   The maximum amount of data allowed for each table is given in the Input Requirements List produced by the Model Generation program. Check that given data falls within this limit or for data card errors.

5.2 DIAGNOSTIC MESSAGES FOR LIBRARY COMPONENTS

Diagnostic messages are produced by many components when a critical variable gets out of bounds during analysis. Adjustment of component parameters may be necessary.

In component alphabetical order, these diagnostic messages are:

AD: INPUT POWER xxxx TOO HIGH RELATIVE TO ADMITTANCE xxxx AND RATED VOLTAGE xxx
ADMITTANCE POWER LOSS xxxx EXCEEDS INPUT POWER xxxx

BA: POWER REQUEST xxxx EXCEEDS BATTERY CAPABILITY. CHECK VC, VO, AND RT.

BN: BN INLET AIR MASS FLOW RATE xxxx GREATER THAN MAXIMUM ALLOWABLE xxxx

CO: MAX ITERATIONS FOR COMPRESSOR EFFICIENCY. NP, XNP, RS = xxxx, xxxx, xxxx

CS: CS STORAGE TEMPERATURE xxxx GREATER THAN ALLOWABLE xxxx
CS MASS OF AIR IN STORAGE xxxx BELOW MINIMUM ALLOWABLE xxxx
CS MASS OF AIR IN STORAGE xxxx EXCEEDS MAXIMUM ALLOWABLE xxxx

ED: INPUT ERROR, DAY OF YEAR DY IS OUT OF RANGE
TAPE INPUT ERROR OR EOF

FL: FLYWHEEL POWER LOSS xxxx EXCEEDS CHARGING POWER xxxx
FLYWHEEL LOSS xxxx EXCEEDS DISCHARGING POWER xxxx
FLYWHEEL CLUTCH LOSS xxxx EXCEEDS DELIVERABLE POWER xxxx
FLYWHEEL KINETIC ENERGY xxxx EXCEEDS CAPACITY xxxx
FLYWHEEL KINETIC ENERGY xxxx FALLS BELOW MINIMUM REQUIREMENT xxxx

GE: GENERATOR OUTPUT EXCEEDS RATED POWER

HS: HS INLET MASS FLOW RATE xxxx OR OUTLET MASS FLOW RATE xxxx IS GREATER THAN MAXIMUM xxxx
HS RESERVOIR VOLUME xxxx EXCEEDED MAXIMUM ALLOWABLE xxxx
HS RESERVOIR VOLUME xxxx DROPPED BELOW MINIMUM xxxx

HT: HT TURBINE CHARACTERISTIC PARAMETER OUT OF RANGE
HT INLET MASS FLOW RATE xxxx GREATER THAN MAXIMUM DESIGN VALUE

HX: HX EXIT TEMPERATURE xxxx GREATER THAN MAXIMUM ALLOWABLE xxxx

IF: IV POWER LOSS xxxx EXCEEDS INPUT POWER xxxx CHECK RATED DC VOLTAGE VDC

MB: WARNING-DIVISOR IN MB EQUALS 0., HAS BEEN SET = 1.

MO: MOTOR INPUT POWER xxxx .GT. RATED INPUT POWER xxxx
MOTOR SLIP xxxx EXCEEDS RATED POWER SLIP xxxx
STATOR RESISTANCE xxxx OR DAMPING xxxx TOO HIGH FOR MOTOR

PV: WARNING: INSOLATION OR TEMPERATURE AT CELL EXCEED RANGE

RE: RE POWER LOSS xxxx EXCEEDS INPUT POWER xxxx
RE, AC INPUT POWER xxxx TOO LARGE IN RELATION TO TRANSFORMER REACTANCE xxxx AND RATED AC VOLTAGE xxxx

TR: TRANSMISSION POWER LOSS xxxx EXCEEDS INPUT xxxx
TRANSMISSION POWER LOSS xxxx EXCEEDS MAXIMUM INPUT POWER

TS: TS WORKING FLUID FLOW RATE xxxx GREATER THAN MAXIMUM ALLOWED xxxx
TS INPUT POWER xxxx GREATER THAN MAXIMUM ALLOWED CHARGE RATE xxxx
TS STORAGE TEMPERATURE xxxx OUTSIDE MINIMUM xxxx OR MAXIMUM xxxx

TU: TURBINE BACK PRESSURE xxxx GREATER THAN STORAGE VESSEL PRESSURE xxxx

6.0 CREATION OF NEW LIBRARY COMPONENTS

The addition of new standard components to the SIMWEST library involves two steps. The first is the design of the component. This design must conform to certain design conventions if the new component is to be compatible with existing components. Section 6.1 discusses these design conventions and the addition of the component subroutine to the SIMWEST library. The second step involves the addition of the new component's input and output description to the SIMWEST file WMPF. File WMPF is used by the precompiler to generate subroutine calling sequences for the library components. Section 6.2 discusses the use of the FILOAD program to accomplish this task.

6.1 LIBRARY COMPONENT CODING

6.1.1 Component Call Sequence

The items in the component subroutine call sequence must be arranged in the following order:

1. Tables
2. Output Quantities
3. Input Quantities

Tables or inputs may be absent from the subroutine call sequence. However those items that are present must follow the sequence given above.

Dummy argument names for the call sequence quantities that are used within each subroutine should be chosen to match the physical quantity names placed in the input, output, and table name lists. Exceptions to this policy may be made when integer names (names starting with I through N) must be avoided or when additional letters will clarify the name.

The subroutine name must contain only two characters and must not duplicate the name of an existing standard component.

Tables

The table arrays must be dimensioned within the component subroutine. They must be dimensioned with only one subscript; e.g., DIMENSION TABLE (1). When table data is passed to the component subroutine, the first word in the array contains the name of the table. The second word contains the number of values given for the primary independent variable. The third word contains the number of values given for the secondary independent variable. Both of these numbers are stored as REAL quantities and must be converted to INTEGER before they can be used as a subscript. This can be done by a statement such as:

```
NX = TABLE (2) - number of primary independent variables
NZ = TABLE (3) - number of secondary independent variables
```

If there is a secondary independent variable, the secondary independent variable array will begin with the fourth word in the array. Thus if this array is designated as z(1), z(2), ..., then:

```
z(1) = TABLE (4)
z(2) = TABLE (5)
z(3) = TABLE (6)
  .       .
  .       .
  .       .
```

The primary independent variable array begins with word NZ + 4. Thus, if this array is designated as X(1), X(2), ..., then:

```
X(1) = TABLE (NZ + 4)
X(2) = TABLE (NZ + 5)
  .         .
  .         .
  .         .
```

The dependent variable array begins with word NX + 4 if there is no secondary independent variable. Thus, if this array is designated as Y(1), Y(2), ..., then:

```
Y(1) = TABLE (NX + 4)
Y(2) = TABLE (NX + 5)
  .         .
  .         .
  .         .
```

If there is a secondary independent variable array and this array was designated Y (I,J), with 1≤ I≤ NX and 1≤ J≤ NZ, then Y(I,J) would be related to the table array as:

```
Y(I,J) = TABLE(NX+NZ+3+I+(J-1)*NX)
```

Normally the individual elements in the table are not used directly but are passed to a table look-up routine. In this case the starting address of the X, Y, and Z tables would be referred to as:

```
Z(1) = TABLE (4)            secondary independent variable table
X(1) = TABLE (NZ+4)         primary independent variable table
Y(1,1) = TABLE (NX+NZ+4)    dependent variable table
```

If more than one table is used by a component subroutine, the table names must appear in the same sequence in the table name list stored in WMPF as in the subroutine call sequence.

Example 6.1: Given a component, HA, that requires the tables TPH and TPC as inputs. The call sequence of this subroutine would appear as:

```
SUBROUTINE HA(TPH,TPC,...
```

Output Quantities

The term "output quantity" refers to information that is calculated and then "output" by a particular component subroutine. This is not to be confused with the "outlet quantities" of the component. The outlet quantities are associated with a particular component port as a result of assigning a positive direction of power or information flow through the component. Some outlet quantities may be calculated by the component subroutine and thus become output quantities of that component. While other outlet quantities may be furnished to the component subroutine and thus become input quantities to that subroutine.

Certain output quantities may be internal to the component and not associated with any port. In other cases the same output quantity may be associated with several ports. In such cases, no port designation is assigned to the output quantity. Such quantities are referred to as "universal port" quantities. As such, they are allowed to connect to any other similar physical quantity regardless of the input quantities port number. This is not the case for quantities with specified port numbers. Once a connection has been made between an input and output quantity with given port numbers, only connections of matching physical quantities with those port numbers occur. Manual override of this provision can be made by specifying particular physical quantity connections.

Three quantities are required for each state variable output. The first is the state variable, the second is the state variable derivative, (rate), and the third is an integer quantity, the integrator control variable.

Example: Given a component, HA, with the following outputs:

Physical Quantity	Port No.	
T	3	Outlet Ports
T	4	
P	1 (State Variable)	Inlet Ports
P	2 (State Variable)	

The call sequence arguments for these outputs would be:

```
SUBROUTINE HA(TPH,TPC,T3,T4,P1,P1DOT,IP1,P2,P2DOT,IP2,...
```

Input Quantities

The term "input quantity" refers to information that is provided to a particular component subroutine. This is not to be confused with the "inlet quantities" of the component. The inlet quantities are associated with a particular component port as a result of assigning a positive direction of power or information, through the component. Some inlet quantities may be calculated by the component subroutine and thus become output quantities of that component, while other inlet quantities may be furnished to the component subroutine and thus become input quantities to that subroutine.

The input quantities should be grouped together by port. That is, all inlet, (port one quantities), then all outlet, (port two quantities), etc. Port designations for two port components which have the same physical quantity on both inlet and outlet will be: port 1 for upstream or inlet port and port 2 for downstream or outlet port. It is important that the inlet port quantities be listed before any outlet port quantities. If a component has multiple inlet ports, the input quantities associated with each inlet port should be grouped together and listed before any outlet port quantities.

Certain input quantities may be internal to the component and not associated with any port. In other cases, the same input quantity may be associated with several ports. In such cases, no port designation is assigned to the input quantity. Such quantities are referred to as "universal port" quantities. As such, they are allowed to connect to any other similar physical quantity regardless of the output quantities port number. This is not the case for quantities with specified port numbers. Once a connection has been made between an input and output quantity with given port numbers, only connections of matching physical quantities with those port numbers occur. Manual override of this provision can be made by specifying particular physical quantity connections.

Example: Given the component HA described in the above example, with the following inputs:

Physical Quantity	Port No.	
T	1	Inlet Ports
T	2	
P	3	Outlet Ports
P	4	
AKH	(universal port quantity)	

The call sequence for these inputs would follow the output arguments, giving the complete call sequence:

```
SUBROUTINE HA(TPH,TPC,T3,T4,P1,P1DOT,IP1,P2,P2DOT,IP2,T1,T2,P3,P4,AKH)
```

The call sequence for standard component subroutines should follow the order shown in Table 6.1-1.

TABLE 6.1-1

COMPONENT SUBROUTINE
CALL SEQUENCE ORDER

1. Tables

2. Output Quantities
   - 2.1 All Outlet Port Quantities*
   - 2.2 All Inlet Port Quantities* (feedback variables)
   - 2.3 All Other Output Quantities

3. Input Quantities
   - 3.1 All Inlet Port Quantities*
   - 3.2 All Outlet Port Quantities* (feedback variables)
   - 3.3 All Other Input Quantities

* Group quantities with the same port number together. If multiple inlet or outlet ports exist, arrange port quantities in order of increasing port numbers.

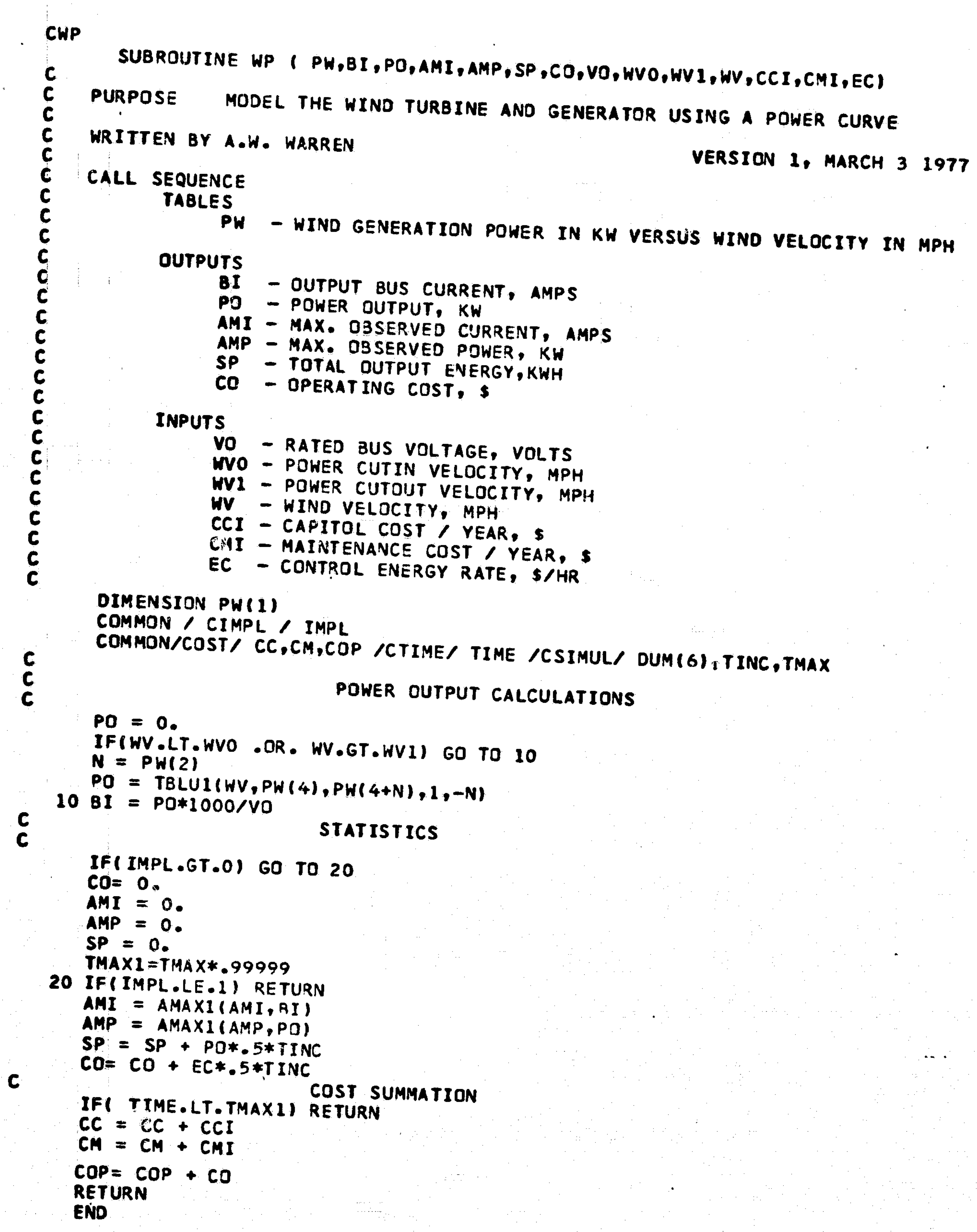
6.1.2 Additions and Modifications to Component Library

Section 4.3 describes the job control procedures to add a new component to the component library, compile the source code that describes the new component and add the relocatable binaries to the component library WMCOMP.

6.1.3 Coding Conventions

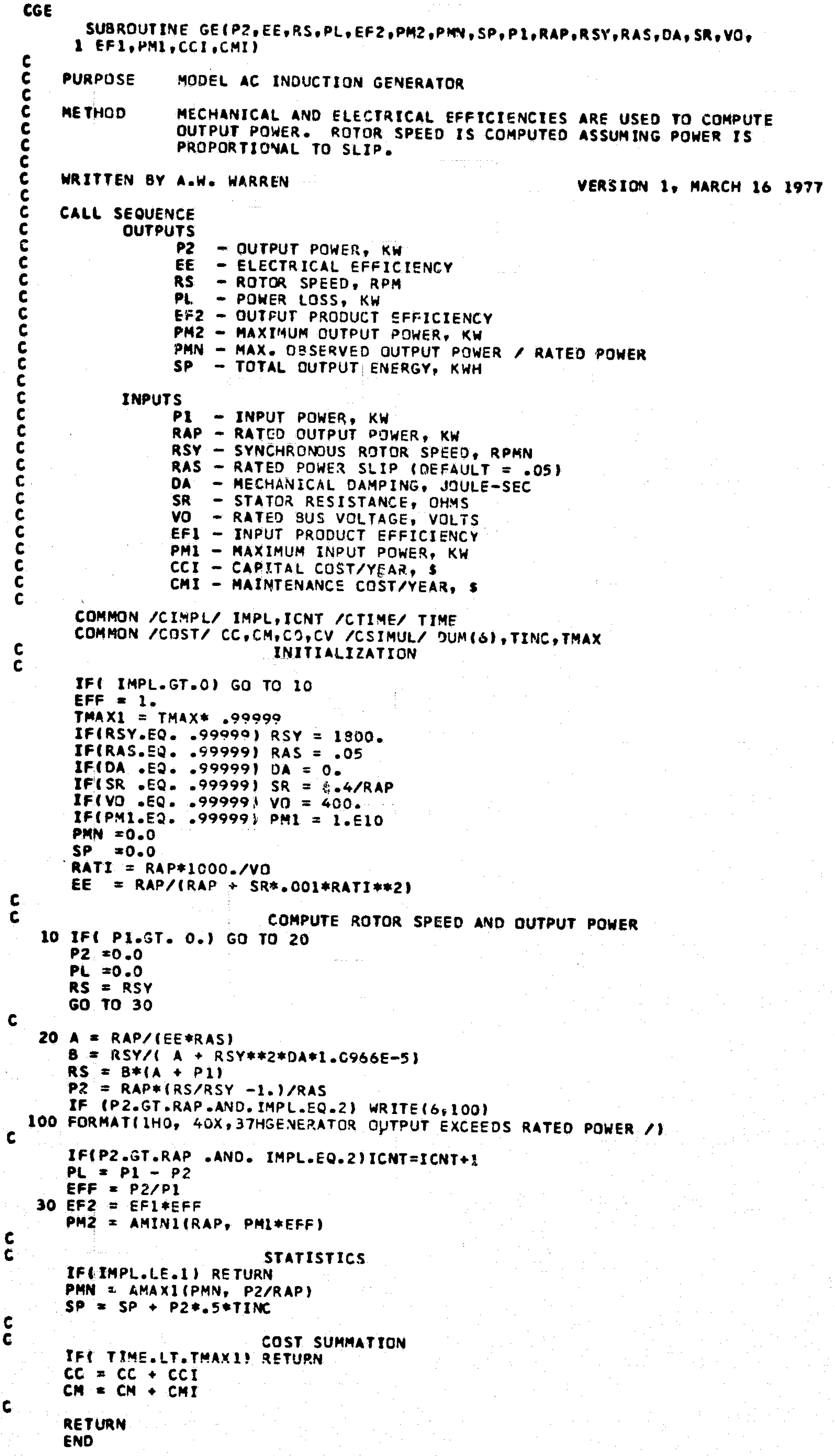
There are several coding rules which apply to any component coded. First of all, the calling sequence must be ordered so that it agrees with that constructed from the FILOAD program. Hence the calling sequence begins with table arrays, is followed by output variables, and then by input parameters. State variables require three sequential parameters in the calling sequence: the state variable, the state derivative, and an integer valued integration control. With the exception of the latter, all parameters in the calling sequence are real valued. In general, one cannot use any local variables or arrays to store information from call to call since there may be several components in the model which call a given subroutine. In other words, local variables can only be used for scratch calculations, unless the computed information is based on COMMON block inputs.

Most of the coding conventions and techniques used are illustrated in Figures 6.1-1 and 6.1-2. Figure 6.1-1 shows the code for the simple power curve component WP. Following the call sequence are a number of comment cards including the component purpose and calling sequence. The table PW is treated as a single dimension Fortran array. Power output is obtained from the table interpolation subroutine TBLU1. (Use of the table interpolation routines TBLU1 and TBLU2 is explained in Section 2.1). The rest of the code shows the conventions used to compute output statistics and add costs for the cost summary. IMPL is an integer variable which indicates the iteration control status:



```
CWP
      SUBROUTINE WP ( PW,BI,PO,AMI,AMP,SP,CO,VO,WVO,WV1,WV,CCI,CMI,EC)
C
C   PURPOSE    MODEL THE WIND TURBINE AND GENERATOR USING A POWER CURVE
C
C   WRITTEN BY A.W. WARREN
C                                                  VERSION 1, MARCH 3 1977
C   CALL SEQUENCE
C         TABLES
C               PW   - WIND GENERATION POWER IN KW VERSUS WIND VELOCITY IN MPH
C
C         OUTPUTS
C               BI   - OUTPUT BUS CURRENT, AMPS
C               PO   - POWER OUTPUT, KW
C               AMI  - MAX. OBSERVED CURRENT, AMPS
C               AMP  - MAX. OBSERVED POWER, KW
C               SP   - TOTAL OUTPUT ENERGY,KWH
C               CO   - OPERATING COST, $
C
C         INPUTS
C               VO   - RATED BUS VOLTAGE, VOLTS
C               WVO  - POWER CUTIN VELOCITY, MPH
C               WV1  - POWER CUTOUT VELOCITY, MPH
C               WV   - WIND VELOCITY, MPH
C               CCI  - CAPITOL COST / YEAR, $
C               CMI  - MAINTENANCE COST / YEAR, $
C               EC   - CONTROL ENERGY RATE, $/HR
C
      DIMENSION PW(1)
      COMMON / CIMPL / IMPL
      COMMON/COST/ CC,CM,COP /CTIME/ TIME /CSIMUL/ DUM(6),TINC,TMAX
C
C
C                             POWER OUTPUT CALCULATIONS
      PO = 0.
      IF(WV.LT.WVO .OR. WV.GT.WV1) GO TO 10
      N = PW(2)
      PO = TBLU1(WV,PW(4),PW(4+N),1,-N)
   10 BI = PO*1000/VO
C
C                             STATISTICS
      IF(IMPL.GT.0) GO TO 20
      CO= O.
      AMI = O.
      AMP = O.
      SP = O.
      TMAX1=TMAX*.99999
   20 IF(IMPL.LE.1) RETURN
      AMI = AMAX1(AMI,BI)
      AMP = AMAX1(AMP,PO)
      SP = SP + PO*.5*TINC
      CO= CO + EC*.5*TINC
C
C                             COST SUMMATION
      IF( TIME.LT.TMAX1) RETURN
      CC = CC + CCI
      CM = CM + CMI
      COP= COP + CO
      RETURN
      END
```

FIGURE 6.1-1 SAMPLE COMPONENT CODE - WP



```
CGE
      SUBROUTINE GE(P2,EE,RS,PL,EF2,PM2,PMN,SP,P1,RAP,RSY,RAS,DA,SR,VO,
     1 EF1,PM1,CCI,CMI)
C
C    PURPOSE     MODEL AC INDUCTION GENERATOR
C
C    METHOD      MECHANICAL AND ELECTRICAL EFFICIENCIES ARE USED TO COMPUTE
C                OUTPUT POWER.  ROTOR SPEED IS COMPUTED ASSUMING POWER IS
C                PROPORTIONAL TO SLIP.
C
C    WRITTEN BY A.W. WARREN                           VERSION 1, MARCH 16 1977
C
C    CALL SEQUENCE
C          OUTPUTS
C                P2  - OUTPUT POWER, KW
C                EE  - ELECTRICAL EFFICIENCY
C                RS  - ROTOR SPEED, RPM
C                PL  - POWER LOSS, KW
C                EF2 - OUTPUT PRODUCT EFFICIENCY
C                PM2 - MAXIMUM OUTPUT POWER, KW
C                PMN - MAX. OBSERVED OUTPUT POWER / RATED POWER
C                SP  - TOTAL OUTPUT ENERGY, KWH
C
C          INPUTS
C                P1  - INPUT POWER, KW
C                RAP - RATED OUTPUT POWER, KW
C                RSY - SYNCHRONOUS ROTOR SPEED, RPMN
C                RAS - RATED POWER SLIP (DEFAULT = .05)
C                DA  - MECHANICAL DAMPING, JOULE-SEC
C                SR  - STATOR RESISTANCE, OHMS
C                VO  - RATED BUS VOLTAGE, VOLTS
C                EF1 - INPUT PRODUCT EFFICIENCY
C                PM1 - MAXIMUM INPUT POWER, KW
C                CCI - CAPITAL COST/YEAR, $
C                CMI - MAINTENANCE COST/YEAR, $
C
      COMMON /CIMPL/ IMPL,ICNT /CTIME/ TIME
      COMMON /COST/ CC,CM,CO,CV /CSIMUL/ DUM(6),TINC,TMAX
C                           INITIALIZATION
C
      IF( IMPL.GT.0) GO TO 10
      EFF = 1.
      TMAX1 = TMAX* .99999
      IF(RSY.EQ. .99999) RSY = 1800.
      IF(RAS.EQ. .99999) RAS = .05
      IF(DA .EQ. .99999) DA = 0.
      IF(SR .EQ. .99999) SR = 4.4/RAP
      IF(VO .EQ. .99999) VO = 400.
      IF(PM1.EQ. .99999) PM1 = 1.E10
      PMN =0.0
      SP  =0.0
      RATI = RAP*1000./VO
      EE  = RAP/(RAP + SR*.001*RATI**2)
C
C                           COMPUTE ROTOR SPEED AND OUTPUT POWER
   10 IF( P1.GT. 0.) GO TO 20
      P2 =0.0
      PL =0.0
      RS = RSY
      GO TO 30
C
   20 A = RAP/(EE*RAS)
      B = RSY/( A + RSY**2*DA*1.0966E-5)
      RS = B*(A + P1)
      P2 = RAP*(RS/RSY -1.)/RAS
      IF (P2.GT.RAP.AND.IMPL.EQ.2) WRITE(6,100)
  100 FORMAT(1H0, 40X,37HGENERATOR OUTPUT EXCEEDS RATED POWER /)
C
      IF(P2.GT.RAP .AND. IMPL.EQ.2)ICNT=ICNT+1
      PL = P1 - P2
      EFF = P2/P1
   30 EF2 = EF1*EFF
      PM2 = AMIN1(RAP, PM1*EFF)
C
C                           STATISTICS
      IF(IMPL.LE.1) RETURN
      PMN = AMAX1(PMN, P2/RAP)
      SP = SP + P2*.5*TINC
C
C                           COST SUMMATION
      IF( TIME.LT.TMAX1) RETURN
      CC = CC + CCI
      CM = CM + CMI
C
      RETURN
      END
```

FIGURE 6.1-2 SAMPLE COMPONENT CODE - GE

IMPL = 0 the first time in a simulation that the model (EQMO) is called
= 1 if more iterations and hence subroutine calls are expected at a given time step
>1 the final iteration through the model

Hence when IMPL = 0, subroutine variables are initialized, default values are assigned, etc. The statistics are only updated at the final iteration when the model has presumably attained steady state values. Finally, the costs are added up when the simulation has reached the maximum time point. Capital costs, maintenance costs, and operating costs are stored in the first three locations of common block COST.

Figure 6.1-2 shows the code for the generator component GE. The program automatically assigns default parameters = .99999. Hence, when IMPL = 0 component dependent default values are assigned whenever the .99999 default is assumed. The code near Format statement 100 shows a typical diagnostic printout. The diagnostic is only printed if IMPL = 2 since we need only diagnose errors at the final iteration. Note that a counter ICNT is updated each time a diagnostic is printed. It is stored in the second location of common block CIMPL and is monitored to see if diagnostic print lines exceeds DLINES. If so, IMPL is set to 3 the final iteration, so that no further diagnostics are printed. The last convention observed here concerns the use of the maximum power and product efficiency variables denoted PM1, EF1, PM2, EF2. These variables are used to communicate information to the logic components PD and PA. The efficiency variable EFF is defined as the ratio of output power to input power except when P1 = 0. In this case the old EFF value is used, but in any case EFF = 0 must be avoided since this would communicate a zero efficiency to a logic device which would then generate an infinite request. Observe that EF2 and PM2 represent the joint efficiency and maximum power at the output port as a consequence of the rated generator power and computed input/output efficiency.

C - 2

Storage devices have in addition to the above, certain conventions to communicate with the logic components. An input parameter RE1 for port 1 request is used to initiate power discharge from storage. An output variable RE2 for port 2 request is used to communicate a maximum charge rate request and is usually computed by

$$RE2 = MIN\ (MP1,\ RAP)/EF1$$

where MP1 and EF1 are the input maximum power and input product efficiency, and RAP denotes the maximum storage charging rate. A priority interrupt INT should also be defined so that INT = 1. when storage is empty or at a minimum, INT = 0. if no interrupt is required, and INT = -1. at full storage capacity. The amount of storage is normally a state variable so that the code computes the state derivative at each time point and lets the integrator update the state at each time point.

6.2 FILOAD PROGRAM

In addition to merging the subroutine representing the new standard component into the component library, descriptions of the inputs, outputs, and tables required by the new component must be added to the permanent file WMPF. These lists are used by the Model Generation program to direct the connection of component inputs and outputs. The program FILOAD is provided to perform any of the following tasks:

1. Add new input, output, or table name lists.
2. Replace existing input, output or table name lists.
3. Remove all name lists for specified components.
4. Dump contents of WMPF file onto TAPE9 in input format.

6.2.1 FILOAD Program Commands

The FILOAD program will recognize the following commands.

LIST STANDARD COMPONENTS

The LIST STANDARD COMPONENTS command causes the program to print the input, output, and table lists for all components modified or added to the WMPF file. If this command is not given the program will merely give a message stating the name of the new components being added to the file.

PURGE

The PURGE command can be used to remove a component from the WMPF file. The PURGE command is folowed by the names of the components to be purged. The command and the component names must be separated by one of the standard delimiters; i.e., [ ] three or more blanks, [,] comma, [=] equal sign, [()] left or right parentheses.

Example 6.3: PURGE = CM, TB, OB

This command would remove all lists for the CM, TB, and OB components from the name list file.

SYMBOL

The SYMBOL command may be used to designate the type of symbol that is to appear for each standard component in the lineprinter drawn model schematic diagram. The SYMBOL command is followed by the names of the components each followed by a symbol number. The symbol numbers and their associated symbols are shown in Figure 6.2-1. The SYMBOL command, component names, and symbol numbers are separated by standard delimiters.

Example: SYMBOL, CO = 100, SH = 200, TU = 300, OC = 400

If a symbol number is not specified for a component, the default symbol of a square box will be used.

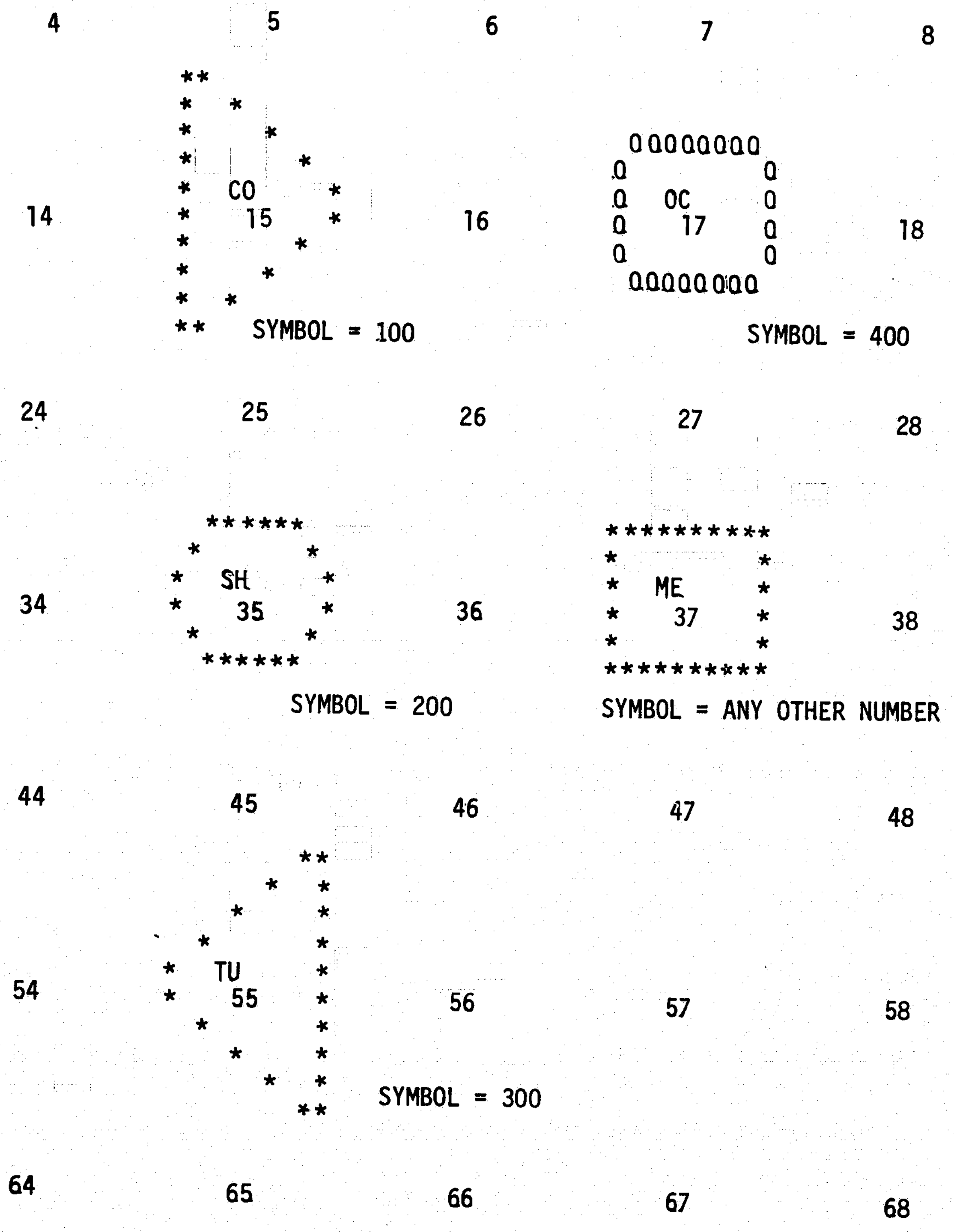


FIGURE 6.2-1 LIST OF STANDARD COMPONENT SYMBOLS

DUMP FILE

The DUMP FILE command causes the FILOAD program to dump the contents of the WMPF file onto TAPE9, in the input format of the FILOAD program. Thus for each standard component, a list of inputs, outputs, and tables will be produced. This data will be preceded by the command NEW FILE described below. This file may be edited to modify the input, output or tables description of any existing standard component or to derive a new standard component description from an existing one. The results of such an editing would then serve as input data to a subsequent run of the FILOAD program. Unless it is intended to purge the WMPF file and start anew, the NEW FILE command at the beginning of TAPE9 should be removed before the subsequent run of the FILOAD program.

NEW FILE

The NEW FILE command instructs the FILOAD program to construct a new WMPF file. This command must occur as the first card in a set of data describing a completely new WMPF file. Any previous components that may have existed on the WMPF file are purged by this command.

FILE NAME

This command is used to load the file name to be associated with the WMPF file. The current WMPF file name is SIMWEST II. This command is used as:

```
FILE NAME = SIMWEST II
```

6.2.2 Input Name Lists

Input name lists are identified by the letters INPT following the component name. Thus, the input name list for a component DC would be introduced with the phrase, DCINPT. This must be followed by a phrase that contains the number of names in the input name list.

The input names are contained on the following data cards, 8 names per card. The names must be left adjusted in fields, 10 characters wide. The names are placed in Columns 1 through 3 of each field. Column 9 of each field can be used to indicate a port number which can be attached to the name to distinguish it from other quantities of the same name that occur with the given component. Thus, to indicate that temperature, T, is an input to port 1, the input name list would be:

Column:	1	2	3	4	5	6	7	8	9	10
Item:	T								1	

This quantity would then be referred to as T1.

Example 6.4:

```
SWINPT = 3
IN......1.IN......2.CNT
```

(The dots are used here to indicate blank spaces and would not be included in an actual data card).

These two data cards would indicate that the component SW had 3 input quantity names. A quantity IN appears at port 1, and is to be referred to as IN1. A quantity IN appears at port 2, and is to be referred to as IN2. A third input quantity CNT has no port designation. Note that if a port number is to be attached to a quantity name, that name should contain no more than 2 characters.

The sequence of names in the input name list must match the sequence of input arguments in the component call sequence.

6.2.3 Output Name Lists

Output name lists are identified by the letters OUTP following the component name. Thus, the output name list for a component DC would be introduced with the phrase, DCOUTP. This must be followed by a phrase that contains the number of names in the output name list.

The output names are contained on the following data cards, 8 names per card. The names must be left adjusted in fields 10 characters wide. The names are placed in Columns 1 through 3 of each field. Column 9 of each field can be used to introduce a port number which can be attached to the name to distinguish it from other quantities of the same name that occur with the given component. If the output quantity is a state variable, this must be indicated by placing S in Column 10 of the field. Thus, if power P is a state variable output quantity at port 2, the output name list would be:

Column:	1	2	3	4	5	6	7	8	9	10
Item:	P								2	S

This quantity would then be referred to as P2.

Example 6.5:

```
TZOUTP = 3
X.......1SX.......2SOUT
```

(The dots are used here to indicate blank spaces, and would not be included on an actual data card).

These two data cards would indicate that the component TZ had 3 output quantity names. A quantity X appears at port 1. This is a state variable, and will be referred to as X1. A quantity X is also a state variable that appears at port 2. It will be referred to as X2. The quantity OUT is an output variable, not a state variable, and does not have a port number associated with it. Note, that if a port number is to be attached to a quantity name that name should contain no more than 2 characters. These two characters plus the port number will reach the maximum number of 3 characters in a quantity name.

The sequence of names in the output name list must match the sequence of output arguments in the component call sequence. However, whereas three arguments are provided for each state in the subroutine call sequence, only one name is included in the output name list.

6.2.4 Table Name Lists

Table name lists are identified by the letters TABS, following the component name. Thus, the table name list for a component CM would be introduced with the phrase CMTABS. This must be followed by a phrase containing the number of tables in the table name list. The table names are contained in the following cards, one table name per card. The name is located in the first 3 columns of the card. It must be accompanied by the maximum dimension that is to be provided for this table. This number must be given in columns 4 through 10 and should have a decimal point given. For single independent variable tables this number must be negative. For tables with two independent variables, this number must be positive.

Example:

```
CMTABS = 3
TAM       53.
TAB       43.
TCM      -27.
```

These four data cards would indicate that the component CM had 3 tables. The first two tables TAM and TAB have two independent variables each, as indicated by the positive dimension numbers. The table TCM has only one independent variable, as indicated by the negative dimension number. 53, 43, and 27 words of storage are to be provided for tables TAM, TAB, and TCM respectively. The maximum storage is related to the maximum number of primary, NX, and secondary, NZ, independent variables by:

MAX = 3 + NX+NZ+NX*NZ for tables with two independent variables
MAX = 3 + 2*NX for tables with one independent variable

7.0 LIBRARY COMPONENT DESCRIPTIONS

This section describes the mathematical algorithms and input/output structure of the SIMWEST library components. Each component writeup contains a brief textual description of the algorithms, a mathematical expression summarizing its function, a list of input and output variables, a description of the calculation sequence and logic used in the model, and the model code. A figure is provided which shows the nominal input and output connections, and the state variables of each component.

There are a number of features and conventions in the component descriptions which require some elaboration. These are briefly summarized below.

7a. INPUT/OUTPUT NAME LISTS

A potentially confusing factor is the way port numbers on input parameters and output variables are designated. On the model generation input cards the name of the physical quantity and the port number are separated by a comma. For example, the power variable with port designation 1 is denoted P,1. To emphasize the distinction between the physical quantities and port numbers they are listed separately in the name lists of the component writeups. For example, P 1 in the name list denotes the power variable (or parameter) with port designation 1 even though in other parts of the text it may simply be denoted P1.

Another convention in the name lists is that the alphabetic symbol 'O' is shown as Ø to distinguish this symbol from a zero. Elsewhere in the text symbols such as VØ may be referred to as VO.

7b. INPUT PARAMETER SPECIFICATION

All input parameters are associated with default values. Many of the parameters have default values denoted in the parameter description by the letter D.

For example, in the Battery component the default value for terminal resistance, RT, is D = .001 ohms. All input parameters for which a default value is not so specified have a default value of .99999. Default values are intended to enable users to put models together quickly by specifying a minimum of input data. Users need only specify detailed parameter values for those components of current interest. One must be careful using this approach since the operating characteristics and efficiency of a 10kw rated device may, for example, be quite different than for a 100kw device.

Any user-specified input parameter can be driven by one or two dimension table lookups using the FU and FV components. This enables the user to build more detailed models using time or other output variables to drive the tables. For example, if one needs to specify cost of peak load generation to the utility component as a function of peak load request, then one adds FU as an input to UT and specifies load request as an input connection to FU. The desired function table for FU is specified in the simulation input.

It may be noted that not all of the components have maintenance or operating cost inputs. Thus, whenever these costs are important, one can aggregate such costs and input lumped costs to the model. For example, the maintenance cost of the hydro storage system may include maintenance costs for the pump and turbine.

7c. COMPONENT LOGIC

In constructing SIMWEST components, we have adopted several conventions to aid communication with the logic components. All physical components distributing power are given two input parameters EF and MP (port 1) and two output variables EF and MP (port 2). The output EF is the product efficiency of all components in the distribution subsystem up to and including the given component, and MP is the maximum power deliverable at the output of the component. Each storage component has in addition a power request input denoted RE (port 1), a power request output denoted RE (port 2), and a priority interrupt flag denoted INT.

Figure 7.0 shows the logic and physical variable connections for power flow in and out of a hydro reservoir. Power flows from the power divider to the pump at a rate not to exceed the request RE from HS. The HS request is computed by dividing the input maximum power by the input (or pump) efficiency EF. Hence, the maximum power flowing to HS cannot exceed RE*EF = MP. Similarly, the input request to HS is computed by the PA component so as not to exceed the maximum input power MP divided by EF (turbine efficiency). Hence, the power that flows to PA cannot exceed RE*EF = input maximum power.

When the hydro reservoir is empty, the interrupt flag is turned on and the priority sequence is changed so that the reservoir is given access to power flowing into the divider.

7d. UNITS

Most of the SIMWEST components are coded in English units. However, SI or metric units were used to code the solar-photovoltaic components: ED, SO, FP, FO, and PV. This is generally not a problem since there are at most only a few interconnection variables between the solar-photovoltaic generation components and other SIMWEST components, and units conversions are easily handled using an MA arithmetic component. (See for example the Fresnel Lens Model, Section 9.3.)

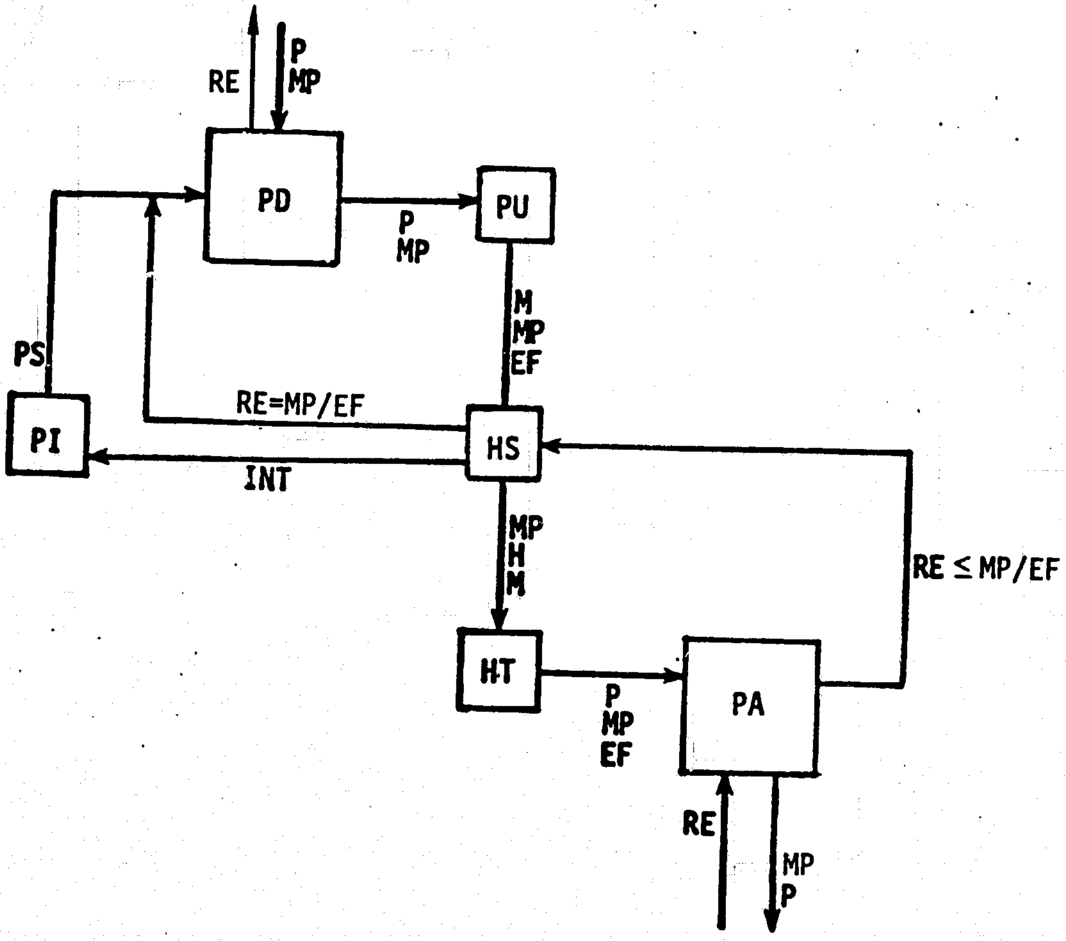
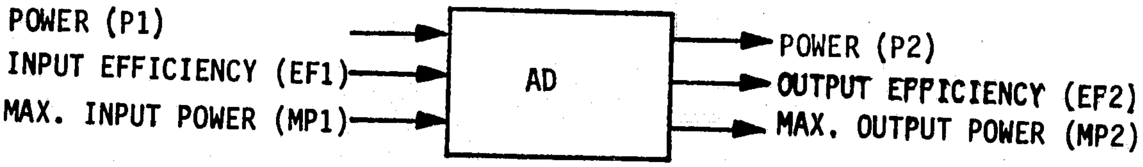


FIGURE 7.0 SAMPLE CONNECTIONS FOR LOGIC COMPONENTS

7.1 ADMITTANCE



The admittance model can be used to model transmission lines, transformers, capacitors or impedance power flows. A primary assumption is that the reactive parameters dominate the real parameters so that power transfer angle is solely based on reactive values, and power losses are based on the real admittance parameters and on power angle. The equation for power loss is based upon the following model:

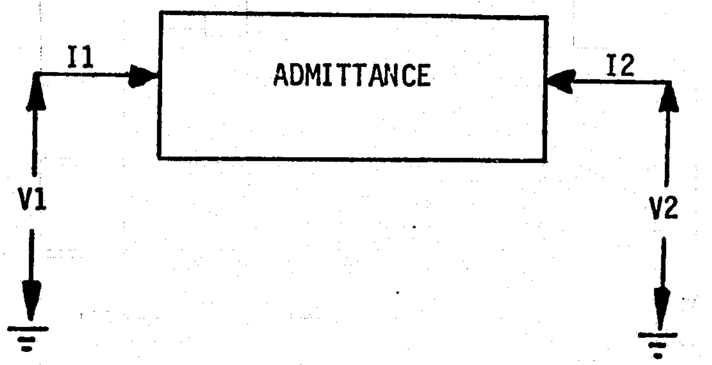


FIGURE 7.1 ADMITTANCE NETWORK MODEL

$$\begin{pmatrix} I1 \\ I2 \end{pmatrix} = \begin{pmatrix} G1 + jB1 & GM + jBM \\ GM + jBM & G2 + jB2 \end{pmatrix} \begin{pmatrix} V1 \\ V2 \end{pmatrix}$$

Where the reactive parameters B_1 and B_2 do not enter into the power loss calculations.

AD

Inputs

Parameter/Port		Description	Units
G1,GM,G2		Real admittance parameters *	mho
BM		Reactive admittance parameter *(≠0)	mho
VØ		Rated voltage magnitude	volts
P	1	Input power	kw
EF	1	Input product efficiency	-
MP	1	Maximum input power	kw
CC		Capital cost/year	$

Outputs

Variable/Port			
P	2	Output power	kw
PL		Power loss	kw
PA		Power angle	deg
EF	2	Output product efficiency	-
MP	2	Maximum output power	kw

* - See next page for User Input to Model Transmission lines, Transformers and Impedances.

Transmission Line Input:

$G1 = G2 = g*\ell$

$GM = -g*\ell$

$BM = 1/(\omega*L*\ell)$

where g = line conductance per unit length

ℓ = length of line

ω = frequency in radians/sec = 120π

L = line inductance per unit length

Transformer Input:

$G1 = G2 = GM = 0$

$BM = 1/X*h$

where X = reactance in ohms

h = turns ratio

(No power loss modeled with a transformer)

Impedance Input: (Includes capacitors and inductors)

$G1 = G2 = -GM = R/(R^2+X^2)$

$BM = X/(R^2+X^2)$

where R = resistance in ohms

X = reactance in ohms

$$= \begin{pmatrix} \omega L & \text{for an inductance } L \\ -\frac{1}{\omega C} & \text{for a capacitance } C \end{pmatrix}$$

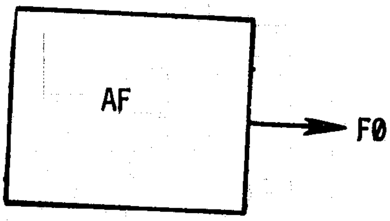
Calculation Sequence

```
If $P1 \leq 0$   $P2 = PL = PA = 0$ and Return

1)  Compute power angle
      If $P1*1000 > BM*VØ^2$, $\cos\theta = 0$ and write DIAGNOSTIC
      $\theta = -SIN^{-1}(P1*1000/BM*VØ^2)$
      $PA = \theta*180/\pi$
      $\cos\theta = \sqrt{1 - (P1*1000/BM*VØ^2)^2}$

2)  Compute power loss and output power
      $PL = VØ^2*(G1+G2+2*GM*\cos\theta)/1000$
      $P2 = P1 - PL$
      $EFF = P2/P1$
      If $P2 > 0$  go to 3)
      write DIAGNOSTIC
      $EFF = 1.$

3)  Efficiency and maximum output power
      $EF2 = EF1*EFF$
      $MP2 = MIN(MP1, |BM| *VØ^2/1000) * EFF$

4)  Compute costs
```

```
CAD
      SUBROUTINE AD(P2,PL,PA,EF2,MP2,  G1,GM,G2,BM,VO,P1,EF1,MP1,CC)
C
C                   PURPOSE   MODEL OF TRANSMISSION LINES,TRANSFORMERS,
C                             CAPACITORS, OR IMPEDANCE POWER LOSS
C
C                   METHOD    OUTPUT POWER AND POWER LOSS COMPUTED FROM
C                             INPUT POWER
C
C                   WRITTEN BY Y.K.CHAN      VERSION 1, JULY,1977
C
C   CALL SEQUENCE
C          OUTPUTS
C                 P2   -OUTPUT POWER,KW
C                 PL   -POWER LOSS,KW
C                 PA   -POWER ANGLE,DEG
C                 EF2  -OUTPUT PRODUCT EFFICIENCY
C                 MP2  -MAXIMUM OUTPUT POWER,KW
C          INPUTS
C                 G1,GM,G2  -REAL ADMITTANCE PARAMETERS,MHO
C                 BM   -REACTIVE ADMITTANCE PARAMETERS (.NE.O.),MHO
C                 VO   -RATED VOLTAGE MAGNITUDE,VOLTS
C                 P1   -INPUT POWER,KW
C                 EF1  -INPUT PRODUCT EFFICIENCY
C                 MP1  -MAXIMUM INPUT POWER,KW
C                 CC   -CAPITAL COST/YEAR,$
C
      COMMON /CIMPL/IMPL,ICNT/CTIME/TIME/CSIMUL/DUM(7),TMAX
     X       /COST/CCI
      REAL MP2,MP1
C
      P2=0.

      TMAX1=TMAX*.99999
      IF(P1.GT.0.)GO TO 10
      P2=0.
      PL=0.
      PA=0.
      MP2=MP1
      EF2=EF1
      GO TO 400
C
C                 COMPUTE POWER ANGLE
C
   10 RR=P1*1000./(BM*VO*VO)
      RR2=RR*RR
      IF(RR2.LE.1.)GO TO 100
      PA=-90.
      RRC=0.
      IF(IMPL.EQ.2)WRITE(6,108)P1,BM,VO
  108 FORMAT(1H0,13H INPUT POWER ,F12.3,33H TOO HIGH RELATIVE TO ADMITTA
     XNCE ,F12.3,19H AND RATED VOLTAGE ,F12.3)
      IF(IMPL.EQ.2)ICNT=ICNT+1
      GO TO 200
  100 THETA=-ASIN(RR)
      PA=THETA*180./3.14159
      RRC=SQRT(1.-RR2)
C
```

```
C                    COMPUTE POWER LOSS AND OUTPUT POWER
C
  200 PL=VO*VO*(G1+G2+2.*GM*RRC)/1000.
      P2=P1-PL
      EFF=P2/P1
      IF(P2.GE.0.)GO TO 300
      P2=0.
      EFF=1.
      IF(IMPL.NE.2)GO TO 300
      WRITE(6,308)PL,P1
  308 FORMAT(1H0,24H ADMITTANCE POWER LOSS ,F12.3,21H EXCEEDS INPUT POWE
     XR  ,F12.3)
      ICNT=ICNT+1
C
  300 EF2=EF1
      IF(P2.GT.0.)EF2=EF1*EFF
      IF(P2.GT.0.)MP2=AMIN1(MP1,ABS(BM)*VO*VO/1000.)*EFF
C
  400 IF(IMPL.LE.1)RETURN
      IF(TIME.LT.TMAX1)RETURN
      CCI=CCI+CC
C
      RETURN
      END
```

7.2 TEST FUNCTION GENERATOR



Inputs

Parameter/Port	Description
CØD	Specifies which analytic function is calculated. (See equations below for use of these inputs)
C1	
C2	
C3	
C4	
C5	

Outputs

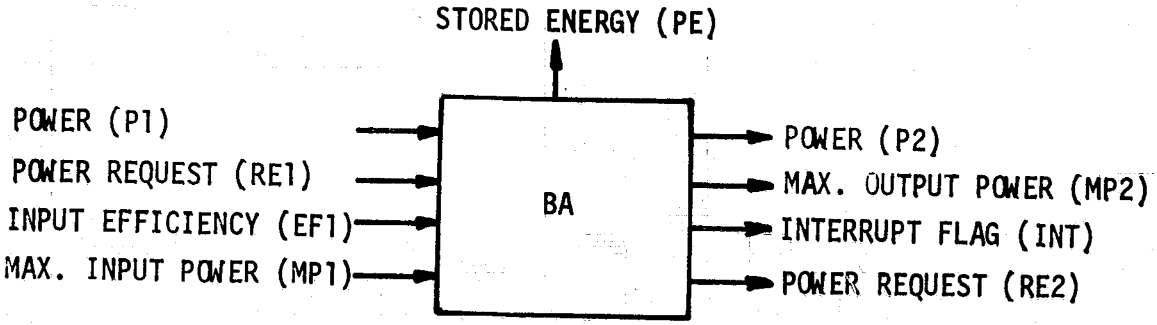
Variable/Port	
FØ	Output variable

Calculation Sequence

```
CØD = 1   FØ = C1 + C2*SIN(C3*T + C4)
      2   FØ = C1 + C2*COS(C3*T + C4)
      3   FØ = C1 + exp(-C5*T) * SIN(C3*T + C4)
      4   FØ = C1 + exp(-C5*T) * COS(C3*T + C4)
      5   FØ = C1 + C2*T
      6   FØ = C1 + C2*exp(-C3*T)
          where: T = TIME
```

```
CAF
      SUBROUTINE AF(FO,COD,C1,C2,C3,C4,C5)
C
C    PURPOSE - TO SIMULATE ANALYTICAL FUNCTIONS
C
C
C    METHOD  - SEE CODING
C
C
C    WRITTEN BY  -  ADAM LLOYD                    LATEST REVISION    FEB 76
C
C
C    LIMITATIONS -  NONE
C
C
C    INPUT/OUTPUT LIST
C
C    FO          OUTPUT VARIABLE                               ANY    OUTPUT VAR
C    COD         CODE IDENTIFYING ANALYTICAL FUNCTION          ---    INPUT  PARAM
C                CODE=    FO=
C                1        C1+C2*SIN(C3*TIME+C4)
C                2        C1+C2*COS(C3*TIME+C4)
C                3        C1+C2*EXP(-C5*TIME)*SIN(C3*TIME+C4)
C                4        C1+C2*EXP(-C5*TIME)*COS(C3*TIME+C4)
C                5        C1+C2*TIME
C                6        C1+C2*EXP(-C5*TIME)
C    C1          CONSTANT INPUTS FOR ABOVE EQNS                ---    INPUT  PARAM
C    C2          CONSTANT INPUTS FOR ABOVE EQNS                ---    INPUT  PARAM
C    C3          CONSTANT INPUTS FOR ABOVE EQNS                ---    INPUT  PARAM
C    C4          CONSTANT INPUTS FOR ABOVE EQNS                ---    INPUT  PARAM
C    C5          CONSTANT INPUTS FOR ABOVE EQNS                ---    INPUT  PARAM
      COMMON/CTIME/TIME
      NCODE=COD
      GO TO (10,20,30,40,50,60),NCODE
 10   FO=C1+C2*SIN(C3*TIME+C4)
      GO TO 100
 20   FO=C1+C2*COS(C3*TIME+C4)
      GO TO 100
 30   FO=C1+C2*EXP(-C5*TIME)*SIN(C3*TIME+C4)
      GO TO 100
 40   FO=C1+C2*EXP(-C5*TIME)*COS(C3*TIME+C4)
      GO TO 100
 50   FO=C1+C2*TIME
      GO TO 100
 60   FO=C1+C2*EXP(-C5*TIME)
 100  RETURN
      END
```

7.3 BATTERY



The battery model is based on the circuit diagram shown below. Current flow is determined by the output power request minus input power. Battery leakage is proportional to stored energy. Priority interrupt logic is activated when a minimum or maximum capacity level is attained.

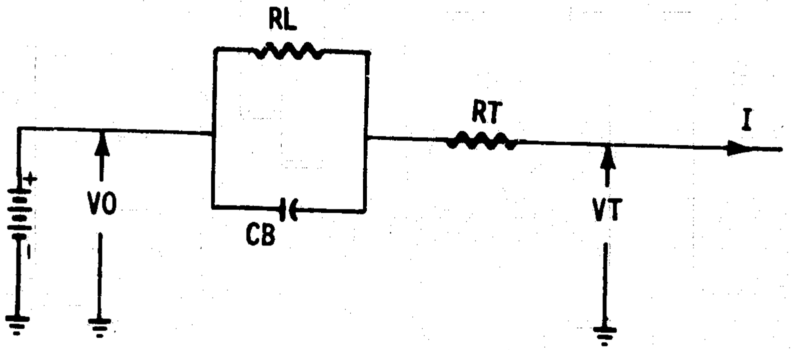


FIGURE 7.3 BATTERY CIRCUIT DIAGRAM

Basic Equations

The output power P2, stored energy PE, terminal current I, and capacitor voltage VC is computed using the following equations:

$$P2 = RE1$$

$$PE = (VC^2+2\ast V\O\ast VC)\ast CB/7.2X10^6$$

$$(P2-P1)\ast 1000 = (V\O+VC)I - I^2\ast RT$$

$$\dot{PE} = -(I+VC/RL)(VC+V\O)/1000$$

BA

Inputs

Parameter/Port		Description	Units
P	1	Input power	kw
VØ		Internal voltage	volts
RT*		Terminal resistance (D = 0.001)	ohms
CB[1]		Battery capacitance (D = $2.88x10^8$)	farads
RL[1]		Leakage resistance (D = 0.05)	ohms
RAP		Rated input power	kw
EF	1	Input product efficiency	-
MP	1	Maximum input power	kw
E1		Maximum energy storage	kwh
RE	1	Power request	kw
EDE		Energy deadband for priority resequencing	kwh
DT		Down time for priority resequencing	h
CC		Capital cost/year	$
CM		Maintenance cost/year	$

Outputs

Variable/Port			
P	2	Output power (=RE1)	kw
PE		Stored energy (state of charge)	kwh
I		Terminal current (+=out,-=in)	amps
VC		Capacitor voltage	volts
VT		Terminal voltage	volts
PL		Power loss	kw
TØ		Time when battery was discharged	h
MP	2	Maximum output power	kw
INT		Priority interrupt flag	-
RE	2	Maximum charging rate request	kw

D - Default values supplied
1 - Battery leakage time constant in hours = CB*RL/3600
* - RT may be used as an adjustment parameter for specifying battery efficiency at rated power.

Statistics

Variable/Port	Description	Units
MPE	Maximum stored energy	kwh
SPC	Sum of charging energy	kwh
SPD	Sum of discharging energy	kwh

Calculation Sequence

1) Compute VC

$$VC = \sqrt{7.2X10^6*PE/CB + V\emptyset^2} - V\emptyset$$

2) Solve for terminal current I

If $(P2-P1)*1000 \geq (VC+V\emptyset)^2/4*RT$, GO TO 2'

$$I = \frac{(VC+V\emptyset) - \sqrt{(VC+V\emptyset)^2 - 4*RT*(P2-P1)*1000}}{2*RT}$$

Go to 3)

2)' $I = (VC+V\emptyset)/2*RT$ and write DIAGNOSTIC

3) Compute VT

$$VT = VC+V\emptyset-I*RT$$

4) Potential energy balance and power loss

$$\dot{PE} = -(I+VC/RL)(VC+V\emptyset)/1000.$$

$$PL = (I^2*RT + VC^2/RL)/1000.$$

5) Maximum charging and discharging rates

$$RE2 = MIN(MP1,RAP,(E1-PE)/TINC)/EF1$$

$$MP2 = MIN(RAP,(VC+V\emptyset)^2/(4000*RT),(PE-EDE)/TINC)$$

where TINC = integration step size

Calculation Sequence Cont.

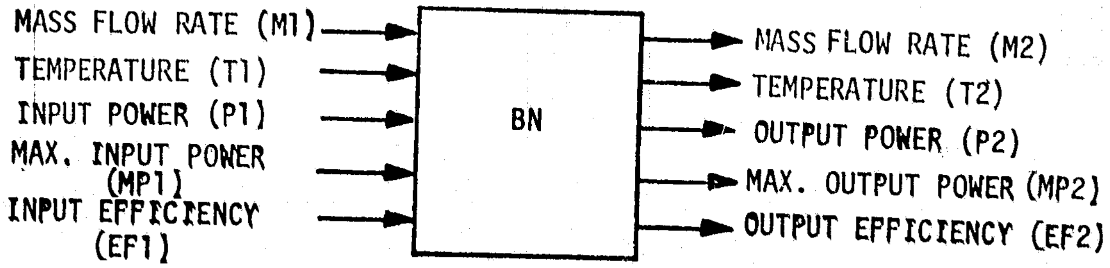
6) Priority interrupt logic

```
    If $PE \le EDE$ and $TØ = 10^6$, $TØ = TIME$
    If $PE \le EDE$ and $TIME - TØ \ge DT$, $INT = 1$ and go to 7)
    $TØ = 10^6$
    If $PE > 2*EDE$ and $INT = 1$, $INT = 0$
    If $PE \ge E1$, $INT = -1$
    If $PE < E1 - EDE$ and $INT = -1$, $INT = 0$
```

7) Compute Statistics and Costs

```
CBA
      SUBROUTINE BA(P2,PE,PED,IPE,I,VC,VT,PL,TO,MP2,INT,RE2,MPE,SPC,SPD,
     1        P1,VO,RT,CB,RL,RAP,EF1,MP1,E1,RE1,EDE,DT,CC,CM)
C
C                 PURPOSE   BATTERY MODEL
C
C                 METHOD    COMPUTE STORED ENERGY AND POWER OUTPUT AS
C                           FUNCTIONS OF POWER INPUT AND POWER REQUEST.
C                           A RESISTOR/CAPACITOR NETWORK IS USED TO
C                           MODEL BATTERY STORAGE.
C
C                 WRITTEN BY Y.K.CHAN                    VERSION 1, JUNE 3,1977
C
C    CALL SEQUENCE
C         OUTPUTS
C                 P2  -OUTPUT POWER, KW
C                 PE  -STORED ENERGY (STATE),KWH
C                 PED -STORED ENERGY DERIVATIVE
C                 IPE -INTEGRATOR CONTROL
C                 I   -TERMINAL CURRENT (+=OUT,-=IN),AMPS
C                 VC  -CAPACITOR VOLTAGE, VOLTS
C                 VT  -TERMINAL VOLTAGE, VOLTS
C                 PL  -POWER LOSS, KW
C                 TO  TIME WHEN BATTERY WAS DISCHARGED, HR
C                 MP2 -MAXIMUM OUTPUT POWER, KW
C                 INT -PRIORITY INTERRUPT FLAG
C                 RE2 -MAXIMUM CHARGING RATE REQUEST, KW
C         STATISTICS
C                 SPC -SUM OF CHARGING ENERGY, KWH
C                 MPE -MAXIMUM STORED ENERGY,KWH
C                 SPD -SUM OF DISCHARGING ENERGY, KHW
C         INPUTS
C                 P1  -INPUT POWER, KW
C                 VO  -INTERNAL VOLTAGE, VOLTS
C                 RT  -TERMINAL RESISTANCE, OHMS
C                 CB  -BATTERY CAPACITANCE, FARADS
C                 RL  -LEAKAGE RESISTANCE, OHMS
C                 RAP -RATED INPUT POWER, KW
C                 EF1 -INPUT PRODUCT EFFICIENCY
C                 MP1 -MAXIMUM INPUT POWER, KW
C                 E1  -MAXIMUM ENERGY STORAGE, KWH
C                 RE1 -POWER REQUEST, KW
C                 EDE -ENERGY DEADBAND FOR PRIORITY RESEQUENCING, KWH
C                 DT  -DOWNTIME FOR PRIORITY RESEQUENCING, HR
C                 CC  -CAPITAL COST/YEAR, $
C                 CM  -MAINTENANCE COST/YEAR, $
C
      COMMON /CIMPL/IMPL,ICNT/CTIME/TIME/CSIMUL/DUM(7),TMAX/COST/CCI,CMI
      REAL I,MP2,MPE,MP1,INT
      TINC1=DUM(7)*.5
C
      IF(IMPL.GT.0) GO TO 100
      IF(RT.EQ..99999)RT=.001
      IF(CB.EQ..99999)CB=2.88E6
      IF(RL.EQ..99999)RL=.05
      TO=1000000.
      INT=0
      TMAX1=TMAX*.99999
      TINC=DUM(7)

      MPE=0.
      SPC=0.
      SPD=0.
C
C                 CAPACITOR VOLTAGE
C
  100 VC=SQRT((PE*7.2E6/CB)+VO*VO)  -  VO
C
C                 TERMINAL CURRENT
C
      P2=RE1
      AA=(P2-P1)*4000.*RT
      B=VC+VO
      B2=B*B
      IF(AA.GT.B2) GO TO 200
      I=B-SQRT(B2-AA)
      I=I/(2.*RT)
      GO TO 300
C
  200 I=B/(2.*RT)
      IF(IMPL.EQ.2)WRITE(6,208)P2
  208 FORMAT(1H0,15H POWER REQUEST ,F12.3,50H EXCEEDS BATTERY CAPABILITY
     1. CHECK VC,VO, AND RT. )
      IF (IMPL.EQ.2)ICNT=ICNT+1
C
C                 TERMINAL VOLTAGE
C
  300 VT=VC+VO-I*RT
C
C                 POTENTIAL ENERGY BALANCE AND ENERGY LOSS
C
      IF(IPE.NE.0)PED=-(I+ VC/RL)*(VC+VO)/1000.
      PL=(I*I*RT+VC*VC/RL)/1000.
C
C                 MAXIMUM CHARGING AND DISCHARGING RATES
C
      AP1=AMAX1(0.,(E1-PE)/TINC)
      RE2=AMIN1(MP1,RAP,AP1)
      RE2=RE2/EF1
      AP2=AMAX1(0.,(PE-EDE)/TINC)
      MP2=AMIN1(RAP,B2/(4000.*RT),AP2)
C
C                 PRIORITY INTERRUPT
C
      C=E1-EDE
      ED2=EDE+EDE
      IF(PE.GT.EDE)GO TO 401
      IF(TO.GT.999999.)TO=TIME
      WAIT=TIME-TO
      IF (WAIT.GT.DT)INT=1
      GO TO 400
  401 TO=1000000
      IF(PE.LE.ED2)GO TO 400
      IF(PE.GT.E1)GO TO 403
      IF(PE.GT.C)GO TO 400
      INT=0
      GO TO 400
  403 INT=-1
  400 CONTINUE
C
      IF(IMPL.LE.1)RETURN
C
C                    STATISTICS
C
      MPE=AMAX1(MPE,PE)
      SPC=SPC+TINC1*P1
      SPD=SPD+TINC1*P2
C
      IF(TIME.LT.TMAX1)RETURN
      CCI=CCI+CC
      CMI=CMI+CM
C
      RETURN
      END
```

7.4 BURNER



The burner model computes the amount of fuel required to be burned in the inlet airstream to raise the air temperature from the given inlet temperature to the specified outlet temperature. The fuel mass flow rate when integrated over time allows calculation of the cost of burner fuel.

Basic Equation

The mass of fuel consumed, F, is computed from the equation:

$$\dot{F} = \frac{M1*CP*(T2-T1)}{NU*HF}$$

Inputs

Parameter/Port		Description	Units
M	1	Inlet air mass flow rate	lb/h
CP		Air heat capacity (D = $72X10^{-6}$)	kwh/lb-°F
T	1	Inlet air temperature	°F
T	3	Outlet air temperature (specified)	°F
NU		Combustor efficiency (D = 0.98)	-
HF		Fuel heating value (D = 5.56)	kwh/lb
CF		Specific fuel cost (D = 0.094)	$/lb
FDM		Maximum allowable fuel mass flow rate (D=17800)	lb/h
CB		Burner cost coefficient (D = 1.683)	$/lb/h
LE		Burner life expectancy	years
MDM		Maximum allowable air mass flow rate (D=27000)	lb/h
EF	1	Input product efficiency	-
MP	1	Maximum input power	kw
P	1	Input power	kw

Outputs

Variable/Port		Description	Units
F		Fuel mass consumed (state)	lb
EF	2	Output product efficiency	-
MP	2	Maximum output power	kw
T	2	Outlet air temperature	°F
FD		Fuel mass flow rate	lb/h
CCØ		Burner capital cost/year	$
CØ		Fuel cost	$
M	2	Outlet mass flow rate (= M1)	lb/h
P	2	Output power	kw

Statistics

FDU		Maximum fuel mass flow rate	lb/h

D - Default values supplied

The calculation sequence and default values are based on a burner sized using first principles to maintain the outlet temperature at 600°F assuming an inlet temperature of 120°F and a mass flowrate of $2.7\text{x}10^4$ lb/h. These conditions represent the extreme conditions expected and should satisfy all burner requirements. No. 6 fuel oil is assumed to be the fuel type. Cost and heating values were obtained from References 1 and 2. Cost estimates for the burner were estimated from the results of Reference 1.

Calculation Sequence

1) Capital Cost

$$CC\emptyset = CB*MDM/LE$$

2) Maximum air mass flow rate allowed

If $M1 = 0$ set $EFF = 1$, $MP2 = MP1$ and go to 3)

$$M1M = \min \left\{ \frac{NU*HF*FDM}{CP*(T3-T1)} , MDM \right\}$$

If $T1 > T3$, $M1M = MDM$

3) Efficiency and maximum discharge power

$$EFF = 1 + M1*CP*(T2-T1)/P1 \quad (\text{if } P1 > 0)$$

$$EF2 = EF1*EFF$$

$$MP2 = \min \{MP1*EFF, P1 * M1M/M1\} \quad (\text{if } M1 > 0)$$

$$P2 = P1*EFF$$

[1] "Preliminary Feasibility Evaluation of Compressed Air Storage Power Systems," United Technologies AER 74-00242, December 1976.

[2] Steam, Its Generation and Use, Babcock and Wilcox, New York, NY, 1972.

Calculation Sequence Cont.

4) Fuel mass flow rate

$$\dot{F} = \frac{M1*CP*(T2-T1)}{NU*HF}$$

$$T2 = MAX(T1, T3)$$

If M1 > M1M write DIAGNOSTIC

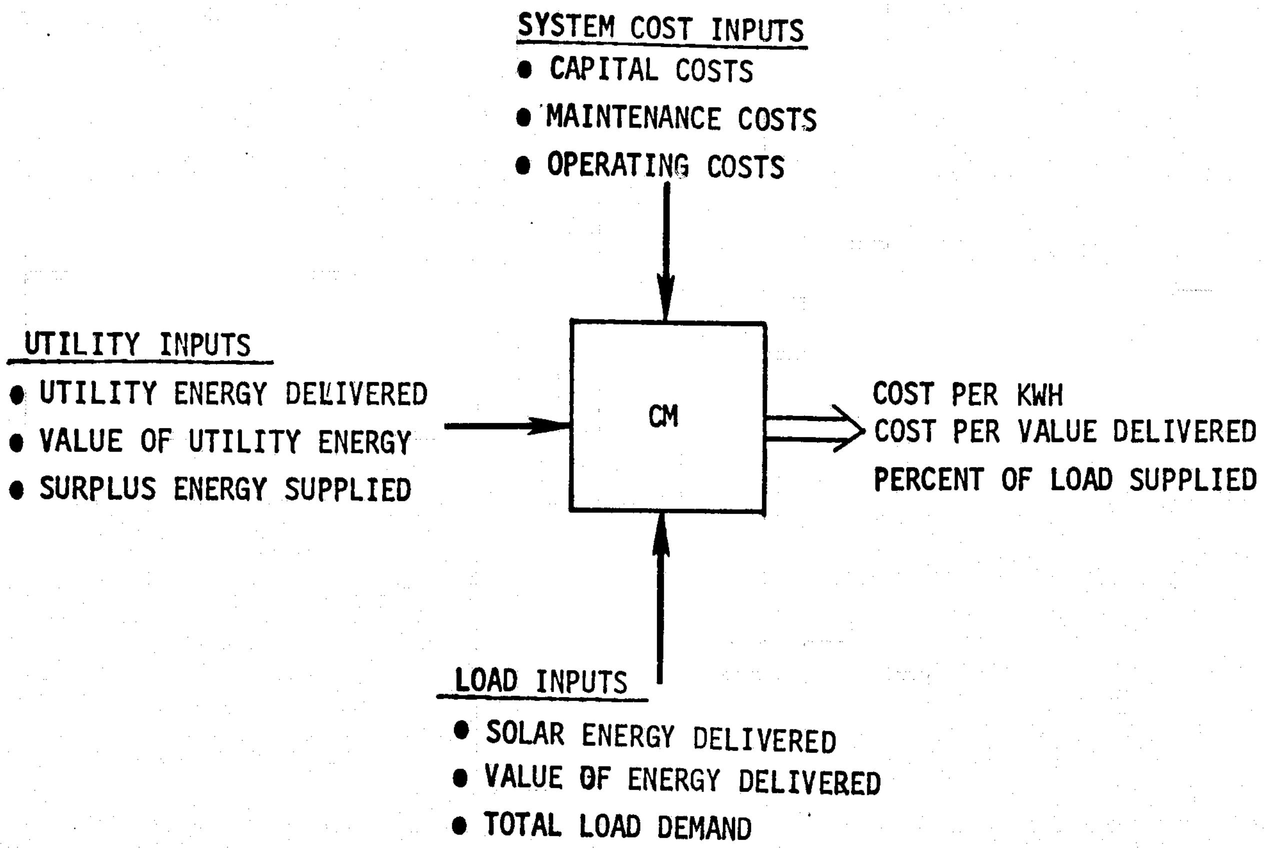
5) Compute Statistics and Costs

$$CØ = CF * F$$

```
CBN
      SUBROUTINE BN(F,DF,IF,EF2,MP2,T3,FD,CC,CO,M2,P2,FDU,M1,CP,T1,T2
     1             ,NU,HF,CF,FDM,CB,LE,MDM,EF1,MP1,P1)
C
C     PURPOSE     COMPUTE FUEL REQUIRED TO RAISE THE AIRSTREAM
C
C                 TEMPERATURE A GIVEN INCREMENT.
C
C     METHOD      INTEGRATE THE FUEL MASS FLOW RATE OVER TIME
C
C     WRITTEN BY F.O. MAHONY                      VERSION 1, MARCH 22 1977
C
C     CALL SEQUENCE
C
C            OUTPUTS
C                 F    - FUEL MASS CONSUMED SINCE TIME=0 (STATE), LB
C                 DF   - FUEL MASS DERIVATIVE
C                 IF   - STATUS INDICATOR
C                 EF2  - OUTPUT PRODUCT EFFICIENCY
C                 MP2  - MAXIMUM OUTPUT POWER, KW
C                 T3   - OUTLET AIR TEMPERATURE, DEG F
C                 FD   - FUEL MASS FLOW RATE, LB/HR
C                 CC   - BURNER CAPITAL COST/YEAR, $
C                 CO   - FUEL COST, $
C                 M2   - OUTLET MASS FLOW RATE, LB/HR
C                 P2   - OUTPUT POWER, KW
C                 FDU  - OBSERVED MAXIMUM FUEL MASS FLOW RATE, LB/HR
C            INPUTS
C                 M1   - INLET AIR MASS FLOW RATE, LB/HR
C                 CP   - AIR HEAT CAPACITY, KWH/LB-DEG F
C                 T1   - INLET AIR TEMPERATURE, DEG F
C                 T2   - OUTLET AIR TEMPERATURE, DEG F
C                 NU   - COMBUSTER EFFICIENCY
C                 HF   - FUEL HEATING VALUE, KWH/LB-DEG F
C                 CF   - SPECIFIC FUEL COST
C                 FDM  - MAXIMUM ALLOWABLE FUEL MASS FLOW RATE, LB/HR
C                 CB   - BURNER COST COEFFICIENT
C                 LE   - BURNER LIFE EXPECTANCY, YEARS
C                 MDM  - MAXIMUM ALLOWABLE AIR MASS FLOW RATE, LB/HR
C                 EF1  - INPUT PRODUCT EFFICIENCY
C                 MP1  - MAXIMUM INPUT POWER, KW
C                 P1   - INPUT POWER, KW
C
      COMMON/CIMPL/IMPL,ICNT /CTIME/TIME /CSIMUL/DUM(7),TMAX
      COMMON/COST/CCI,CMI,COP
      REAL MP2,M2,M1,NU,LE,MDM,MP1,M1M
C
      IF(IMPL.GT.0) GO TO 100
      TMAX1 =TMAX*.99999
      IF(CP .EQ. .99999) CP = 72.0E-6
      IF(NU .EQ. .99999) NU = 0.98
      IF(HF .EQ. .99999) HF = 5.56
      IF(CF .EQ. .99999) CF = 0.094
      IF(FDM.EQ. .99999) FDM=1.78E+4
      IF(CB .EQ. .99999) CB =1.683
      IF(MDM.EQ. .99999) MDM=2.7E+4
C
```

```
      FDU = 0.0
      CC =CB*MDM/LE
  100 EFF=1.0
      IF(M1.EQ.0.0)GO TO 200
      M1M=MDM
      IF(T1.GT.T2) GO TO 200
C
C                  MAXIMUM ALLOWABLE AIR FLOW RATE
C
      M1M=AMIN1(NU*HF*FDM/CP/(T2-T1),MDM)
C
      IF(M1.GT.M1M) GO TO 1000
  300 CONTINUE
C
C                  EFFICIENCY AND MAXIMUM DISCHARGE POWER
      IF(P1.EQ.0.0)GO TO 200
C
C
      EFF = 1.0+M1*CP*(T2-T1)/P1
C
  200 EF2 = EF1*EFF
      MP2=MP1
      IF(M1.GT.0.) MP2=AMIN1(MP1*EFF,P1*M1M/M1)
      P2=P1*EFF
C
C                  FUEL FLOW RATE
C
      IF(IF.NE.0) DF= M1*CP*(T2-T1)/NU/HF
      IF(T1.GT.T2) DF=0.0
      FD=DF
C
C                  COSTS
      CO = CF*F
      T3=AMAX1(T1,T2)
      M2=M1
C
C                  STATISTICS
      IF(IMPL.LE.1) RETURN
C
      FDU = AMAX1(FD,FDU)
C
      IF(TIME.LT.TMAX1) RETURN
      CCI= CCI + CC
      COP= COP + CO
      RETURN
C
 1000 IF(IMPL.EQ.2)WRITE(6,1010) M1,M1M
 1010 FORMAT(1H0,28HBN INLET AIR MASS FLOW RATE ,F12.3,
     1          36H    GREATER THAN MAXIMUM ALLOWABLE ,F12.3)
      IF(IMPL.EQ.2)ICNT=ICNT+1
      GO TO 300
      END
```

7.5 COST MONITOR[1]



This component sums the capital, operating and maintenance costs of all system components. The total yearly cost TC is then computed using a fixed charge rate factor which represents depreciation, cost of money, insurance and taxes.

The total energy delivered to the loads plus surplus energy is then summed and yearly energy delivered TED computed. Cost of operation in mills is

[1] This component must be placed last in the model generation input file, i.e., just prior to the END OF MODEL command.

then given by

$$\text{System cost/kwh} = TC * 1000./TED$$

Similarly, the value of energy delivered to the loads is summed minus the utility energy value and including the value of surplus energy, and factored to give yearly energy value delivered VED. Energy value in mills is given by

$$\text{Load value/kwh} = VED * 1000./TED.$$

Cost per value delivered is the ratio of the above two equations.

In addition to the above cost calculations, percent of total load supplied by storage PCW, percent of load supplied by utilities PCU, and percent of energy surplused to the utilities PCS is computed. The total cost in mills to meet the load is then given by

$$\text{Load cost/kwh} = (\text{system cost/kwh} * PCW + \text{utility cost/kwh} * PCU)/100.,$$

where

$$\text{Utility cost/kwh} = \text{value of utility energy} * 1000./\text{utility energy delivered}.$$

Inputs

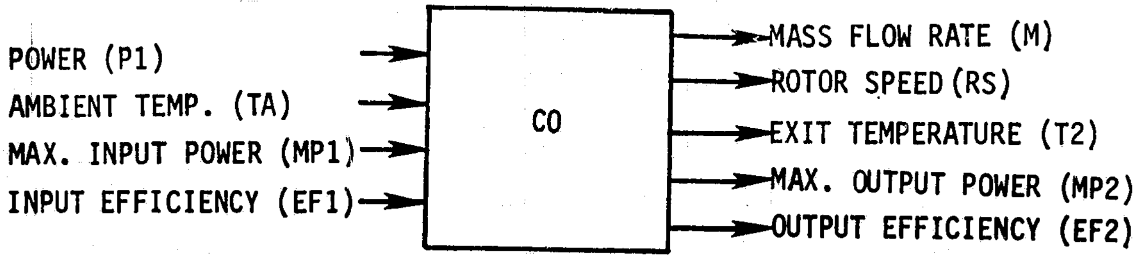
Parameter/Port	Description	Units
CR	Capital charge rate	%/year
LE	System life expectancy	years

Common Block Inputs	Description	Units
CC	Total yearly capital costs	$
CM	Total yearly maintenance costs	$
CO	Operating and fuel costs over TMAX	$
TMAX	Simulation time interval	hr
VDE	Value of energy delivered (including surplus)	$
TDE	Solar energy delivered (including surplus)	kwh
TLD	Total load demand	kwh
UTV	Value of utility energy	$
UTD	Utility energy supplied	kwh
SPD	Surplus energy supplied	kwh
Outputs[1]		
	Total yearly costs (TC)	$
	Yearly energy delivered (TED)	kwh
	Cost of energy per kwh	mills
	Yearly value delivered (VED)	$
	Cost per value delivered	-
	Percent of load supplied by	
	Storage (PCW)	-
	Utility (PCU)	-
	Surplus energy load factor (PCS)	-
	Total load cost per kwh	mills

[1] Printout only occurs when simulation is completed. Thus no output variable symbol is required.

```
      SUBROUTINE CM(DUMM,FCR,LE)
C
C     PURPOSE     SUMMARIZE WIND ENERGE STORAGE COSTS AND LEVELIZED
C                 ENERGY COSTS PER KWH.
C
C     WRITTEN BY A.W. WARREN                           VERSION 1, MAY 1977
C
C     INPUT PARAMETERS
C
C                 FCR  - FIXED CHARGE RATE FACTOR INCLUDING DEPRECIATION,
C                        MONEY COST, INSURANCE, AND TAXES,  PER YEAR
C                 LE   - SYSTEM LIFE EXPECTANCY , YEARS
C                 TMAX - SIMULATION TIME, HR
C                 CC   - TOTAL YEARLY CAPITAL COSTS, $
C                 CM   - TOTAL YEARLY MAINTENANCE COSTS, $
C                 CO   - TOTAL OPERATING AND FUEL COSTS OVER TMAX, $
C                 VDE  - VALUE OF ENERGY DELIVERED OVER TMAX, $
C                 TDE  - TOTAL ENERGY DELIVERED OVER TMAX, KWH
C                 TLD  - TOTAL LOAD DEMAND OVER TMAX, KWH
C                 UTV  - VALUE OF UTILITY ENERGY SUPPLIED LESS SURPLUS VALUE, $
C                 UTD  - TOTAL UTILITY ENERGY DELIVERED, KWH
C                 SPD  - TOTAL SURPLUS ENERGY SUPPLIED TO UTILITY, $
C
      COMMON /COST/ CC, CMA,CO,VDE,TDE,TLD,UTV,UTD ,SPD
      COMMON /CIMPL/IMPL  /CTIME/ TIME /CSIMUL/ DUM(7),TMAX
      REAL LE
C                                INITIALIZATION
C
      IF(IMPL.GT.0)GO TO 100
      DUMM=0.0
      CC  = 0 .
      CMA  = 0.
      CO  = 0.
      VDE= 0.
      TDE= 0.
      TLD= 0.
      UTV=0.
      UTD=0.
      SPD=0.
      TMAX1= TMAX*.99999
C
  100 IF(TIME.LT.TMAX1)RETURN
      IF(IMPL.LE.1)RETURN
C
C                                COST SUMMARY OUTPUT
C
      LLE  = LE
      WRITE(6,200)LLE
  200 FORMAT(1H1,35X,39H SOLAR/WIND ENERGY STORAGE COST SUMMARY //
     1 1H ,40X,I2,17H  YEAR LIFE CYCLE )
C
      COY  = CO*8760./TMAX
      CCY  = CC*LE*FCR*.01
      TOY  = COY  + CMA + CCY
      WRITE(6,300)CCY,CMA,COY,TOY
  300 FORMAT(//// 30X,22H0  YEARLY SYSTEM COSTS/ 1H+,29X,1H+/ 1H-,42X,
     1 12HCAPITAL COST,12X,F8.0,2H $ / 1H ,42X, 17H(INCLUDING FIXED ,
     2  8HCHARGES) / 1H0,42X,16HFIXED O + M COST, 8X,F6.0,2H $ /1H0 ,
     3 42X,21HOPERATING + FUEL COST, 3X,F8.0,2H $ / 1H0,42X,5HTOTAL,
     4 19X,F8.0,2H $ )
C
      EDE  = TDE * 8760./TMAX
      IVDE = VDE * 8760./TMAX
      TOYN = TOY*1000./ EDE
      VDEN = VDE*1000./ TDE
      CPV  = TOYN / VDEN
C
      WRITE(6,400)EDE,TOYN,IVDE,VDEN,CPV
  400 FORMAT(//// 30X,26H0  ENERGY DELIVERED        / 1H+,29X,1H+ / 1H-,
     1 42X,16HENERGY DELIVERED, 7X,F9.0,4H KWH / 1H0,33X,50(1H*) /
     1  1H ,33X,1H*,48X,1H* /
     2 1H ,33X,1H*, 8X,19HENERGY COST PER KWH, 7X,F6.1,9H MILLS  * /
     2  1H ,33X,1H*,48X,1H* / 1H ,
     3 33X,10(5H*****) / 1H0,42X,25HVALUE OF ENERGY DELIVERED,17,
     4 2H $ / 1H ,42X,22H(VALUE OF FUEL SAVED) / 1H0,42X,20HENERGY VALUE
     5 PER KWH, 6X,F6.1,6H MILLS / 1H0,42X,24HCOST PER VALUE DELIVERED,
     6 2X,F6.2)
C
      PCD= (TDE-SPD)*100./TLD
      PCU= UTD*100./TLD
      PCS= SPD*100./TLD
      CPKWH= (TOYN*(TDE-SPD) + UTV*1000.)/TLD
      WRITE(6,500)PCD ,PCU,PCS,CPKWH
  500 FORMAT(//// 30X,31H0  LOAD FACTOR                         / 1H+,29X,
     1  1H+ / 1H-,42X,
     1 26HPERCENT OF LOAD SUPPLIED  , F6.1, 2H  / 1H ,42X, 28HBY TOTAL S
     2OLAR SYSTEM        / 1H0,42X,24HPERCENT OF LOAD SUPPLIED,2X,F6.1 /
     2 1H ,41X,11H BY UTILITY  /
     3  1H0,42X,26HPERCENT OF SOLAR ENERGY    ,F6.1 /
     3 1H ,42X,9HSURPLUSED  /
     4  1H0,42X,23HCOST TO MEET LOAD       , 3X,F6.1,6H MILLS/
     5 1H ,42X,29H(SOLAR + UTILITY)               / 1H1)
C
      RETURN
      END
```

7.6 COMPRESSOR (PNEUMATIC)



The compressor model represents the off-design performance of a typical axial flow compressor. The compressor is assumed designed for a specified set of design operating conditions and performance requirements. The mass flow rate is assumed directly proportional to angular velocity and independent of the pressure ratio across the compressor. This is expected to hold for ±15% of the design mass flow rate. The polytropic efficiency of the compressor is assumed to be a weak function of the angular velocity. Initial calculations are made with the design polytropic efficiency, and refinements made after the off-design parameters are calculated.

Basic Equations

The expression for the angular velocity is

$$RS = P1*\frac{RSD}{MD}\,\frac{EFF}{CP*(TA+460)*[(PR2/PA\ )**A\ -1]}$$

where:

$$EFF = ((PR2/PA)**(A*NP) - 1.)/((PR2/PA)**A - 1.)$$

$$A = (GAM -1)/GAM*NP)$$

Inputs

Parameter/Port		Description	Units
P	1	Input power	kw
RSD		Design angular velocity (D = 3600)	rpm
MD		Design mass flow rate (D = 3000)	lb/h
CP		Heat capacity of air (D = $7.2X10^{-5}$)	kwh/lb°F
TA		Inlet air temperature (ambient) (D = 70)	°F
PR	2	Exit pressure (D = 147)	psi
PA		Inlet pressure (ambient) (D = 14.7)	psi
GAM		Heat capacity ratio (D = 1.4)	-
EF	1	Input product efficiency	-
MP	1	Maximum input power	kw
NPD		Design polytropic efficiency (D = 0.88)	-
PID		Design inlet pressure (ambient) (D = 14.7)	psi
PØD		Design outlet pressure (D = 147)	psi
TID		Design inlet air temp (ambient) (D = 70)	°F
CK		Compressor capacity cost coefficient[1] (D = 1.0)	-
FØ		Compressor exponent for cost calculations (D = 0.75)	-

Outputs

Variable/Port			
NP		Polytropic efficiency	-
EF	2	Output product efficiency	-
MP	2	Maximum output power	kw
TØ		Torque	ft-lb
M		Mass flow rate	lb/h
T	2	Exit temperature	°F
RS		Angular velocity	rpm
CCØ		Cost of compressor/year	$

[1] CK = capital cost (known unit)/(design point mass flow rate)^{FO}* LN (outlet/inlet pressure ratio)*(life expectancy of unit)

D – Default values supplied

Statistics	Description	Units
MT	Maximum temperature	^oF
MF	Maximum mass flow rate	lb/h

The calculation sequence and the default values are based on the assumption of an axial flow compressor, nominally rated at 125kw, and a pressure ratio of 10. The equations used relate first order effects among the various physical quantities and were derived from first principles originally to support the research work of Reference 1. Cost scaling was also developed in that reference based on cost estimates obtained from turbomachinery manufacturers.

Calculation Sequence

```
1)  Costs (First pass only)

        $CCØ = CK * (MD)**FØ * LN (\frac{PØD}{PID})$

    If $P1 > 0$  go to 2)
        $A = (GAM-1)/(GAM*NPD)$
        $RAT = (PØD/PID)**A$
        $EFF = 1$, $RS = 0$,  go to 3)
```

1. "Closed Cycle High Temperature Central Receiver Concept for Solar Electric Power," BEC/EPRI RP 377-1, June 1976.

Calculation Sequence Cont.

2) Angular velocity iteration

A = (GAM-1)/(GAM*NP) (Initially NP = NPD)

RAT = (PR2/PA)**A

EFF = (RAT**NP-1)/(RAT-1)

$$\frac{RS}{RSD} = \frac{P1}{MD}\,\frac{EFF}{CP*(TA+460) * (RAT - 1)}$$

Polytropic efficiency

$$NP = 1 - (1 - NPD)*[2.0- (\frac{PID}{PA}* \frac{(TA+460)}{(TID+460)}*\frac{RSD}{RS})**0.2]$$

Iterate until NP and RS are consistent

(If iteration doesn't converge, then write DIAGNOSTIC and exit)

3) Mass flow rate

M = MD*RS/RSD

4) Exit temperature

T2 = (TA+460)*RAT -460

5) Torque

If P1 ≤ 0, set TØ = 0 and go to 6)

TØ = P1*737.6/(RS*2π/60)

6) Efficiency and maximum power

EF2 = EF1*EFF

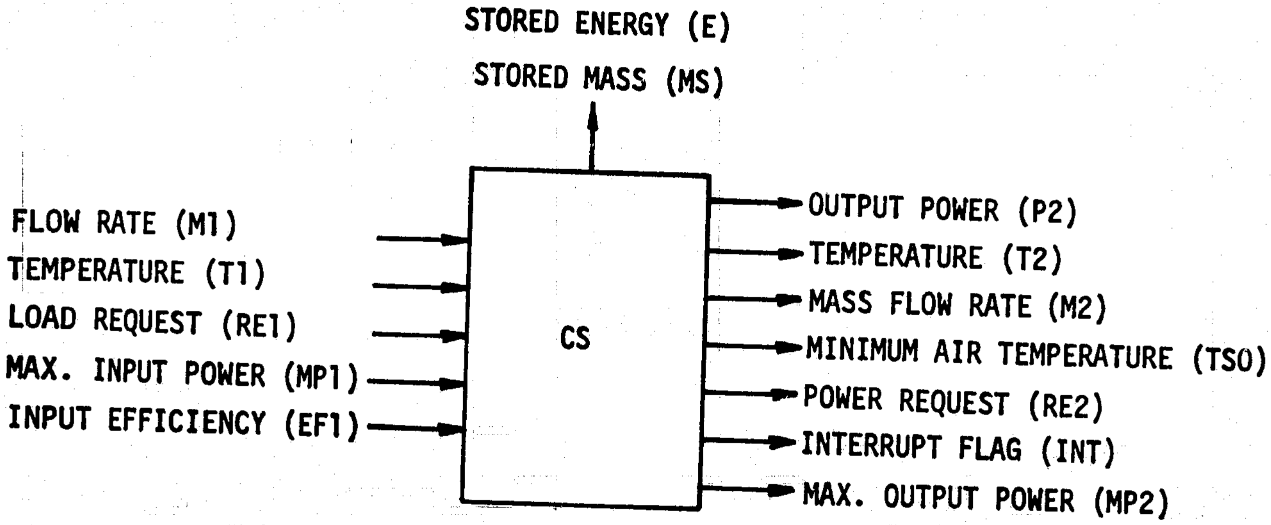
MP2 = MIN(MP1*EFF, 1.5*MD*CP*(T2-TA))

7) Compute Statistics and Costs

```
CCO
      SUBROUTINE CO(NP,EF2,MP2,TO,M,T2,RS,CC,MT,MF,P1,RSD,MD,CP,TA,PR2,
     1           PR1,GAM,EF1,MP1,NPD,PID,POD,TID,CK,FO)
C
C    PURPOSE     PERFORMANCE MODEL OF AXIAL FLOW COMPRESSOR
C
C    METHOD      COMPRESSOR IS SIZED FROM INPUT OPERATING REQUIREMENTS.
C
C                MASS FLOW IS ASSUMED PROPORTIONAL TO ANGULAR VELOCITY
C
C                AND INDEPENDENT OF PRESSURE RATIO.
C
C    WRITTEN BY F.O. MAHONY                       VERSION 1, MARCH 22 1977
C
C    CALL SEQUENCE
C          OUTPUTS
C                NP  - POLYTROPIC EFFICIENCY
C                EF2 - OUTPUT PRODUCT EFFICIENCY
C                MP2 - MAXIMUM OUTPUT POWER, KW
C                TO  - TORQUE, FT-LB
C                M   - MASS FLOW RATE, LB/HR
C                T2  - EXIT TEMPERATURE, DEG F
C                RS  - ANGULAR VELOCITY, RPM
C                CC  - COST OF COMPRESSOR PER YEAR, $/YEAR
C                MT  - MAXIMUM TEMPERATURE OBSERVED, DEG F
C                MF  - MAXIMUM MASS FLOW RATE, LB/HR
C
C          INPUTS
C                P1  - INPUT POWER, KW
C                RSD - DESIGN ANGULAR VELOCITY, RPM
C                MD  - DESIGN MASS FLOW RATE, LB/HR
C                CP  - HEAT CAPACITY OF AIR, KWH/LB-DEG F
C                TA  - INLET AIR TEMPERATURE (AMBIENT), DEG F
C                PR2 - EXIT PRESSURE, PSI
C                PR1 - INLET PRESSURE (AMBIENT), PSI
C                GAM - HEAT CAPACITY RATIO
C                EF1 - INPUT PRODUCT EFFICIENCY
C                MP1 - INPUT MAXIMUM DISCHARGE POWER, KW
C                NPD - DESIGN POLYTROPIC EFFICIENCY
C                PID - DESIGN INLET PRESSURE (AMBIENT),PSI
C                POD - DESIGN OUTLET PRESSURE (AMBIENT), PSI
C                TID - DESIGN INLET TEMPERATURE (AMBIENT), DEG F
C                CK  - COMPRESSOR CAPCITY COST COEFFICIENT
C                FO - COMPRESSOR EXPONENT FOR COST CALCULATION
C
      COMMON /CIMPL/ IMPL /CTIME/ TIME /CSIMUL/DUM(7),TMAX /COST/CCI
      REAL MD,MP1,NPD,NP,MP2,M,MT,MF
      DATA PI   /3.14159/
C                           INITIALIZATION
C
      IF(IMPL.GT.0) GO TO 100
      MT = 0.0
      MF = 0.0
C
      IF (RSD.EQ. .99999) RSD = 3600.0
      IF (MD .EQ. .99999)MD = 3.0E3
      IF(MP1.EQ. .99999) MP1= 1.E8
      IF (CP .EQ. .99999)CP = 72.0E-6
      IF (TA .EQ. .99999)TA = 70.0
      IF (PR2.EQ. .99999)PR2 = 147.0
      IF (PR1.EQ. .99999)PR1 = 14.7
      IF (GAM.EQ. .99999)GAM = 1.4
      IF (NPD.EQ. .99999)NPD = 0.88
      IF (PID.EQ. .99999)PID = 14.7
      IF (POD.EQ. .99999)POD = 147.0
      IF (TID.EQ. .99999)TID = 70.0
      IF (CK .EQ. .99999)CK = 1.0
      IF (FO .EQ. .99999)FO = 0.75
      NP = NPD
C
C                               COST
C
      CC = CK*MD**FO*ALOG(POD/PID)
      TMAX1 = TMAX*.99999
  100 CONTINUE
C
C                               SOLVE FOR POLYTROPIC EFFICIENCY
C                               AND ANGULAR VELOCITY
C
      ISP = 0
C
      EFF=1.0
      IF(P1.GT.0.0)GO TO 200
C
      RAT= (POD/PID)**((GAM-1.0)/(GAM*NPD))
      TO =0.0
      RS=0.0
      NP=NPD
      GO TO 300
C
  200 A = (GAM-1.0)/(GAM*NP)
      RAT= (PR2/PR1)**A
      EFF= (RAT**NP - 1.)/(RAT - 1.)
C
      RSNO = EFF*P1/MD*1.0/CP/(TA+460.0)/(RAT-1.0)
C
      XNP = NP
C
      NP = 1.0-(1.0-NPD)*(2.0-(PID/PR1*(TA+460.0)/(TID+460.0)/RSNO)
     1                  **0.2)
C
      IF(ISP.GT.10) GO TO 1000
C
      ISP = ISP+1
C
      IF(ABS((NP-XNP)/NP).GT.0.001) GO TO 200
      RS= RSD*RSNO
C
C                               MASS FLOW RATE
C
  300 M = MD*RS/RSD
C
C                               EXIT TEMPERATURE
C
      T2 = (TA+460.0)*RAT-460.0
      IF(P1.LE.0.0)GO TO 400
```

```
C
C                               TORQUE
C
      TO = P1*737.6/(RS*2.0*PI/60.0)
C
C                               EFFICIENCY AND MAXIMUM POWER
C
C
C
  400 EF2 = EF1*EFF
C
      MP2 = AMIN1(MP1*EFF,1.5*MD*CP*(T2-TA))
C                               STATISTICS AND COST SUMMATION
C
      IF (IMPL.LE.1) RETURN
      MT = AMAX1(MT, T2)
      MF = AMAX1(MF, M)
      IF(TIME.LT.TMAX1) RETURN
      CCI = CCI + CC
C
      RETURN
C
 1000 WRITE (6,1010) NP,XNP,RS
 1010 FORMAT (1H0,40HMAX ITERATIONS FOR COMPRESSOR EFFICIENCY,
     1             15H - NP,XNP,RS = ,3F12.6)
      STOP
      END
```

7.7 PNEUMATIC STORAGE VESSEL (CONSTANT PRESSURE)



The pneumatic storage vessel is based on a constant pressure underground cavern design as represented in Figure 7.7. A surface pressure-compensation pond via a water shaft is assumed to maintain the vessel pressure at a constant value. This model is assumed to be used in conjunction with a heat exchanger. The energy is calculated as a function of the stored gas mass, the inlet/storage air temperature, and a leakage function proportional to the stored energy.

Basic Equation

The rate of energy storage is computed from the equation

$$\dot{E} = M1*CP*(T1-TO) - NU*E, \text{ charging}$$

$$\dot{E} = -M2*CP*(T2-TO) - NU*E, \text{ discharging}$$

where M1 = mass flow rate during charge
M2 = mass flow rate during discharge

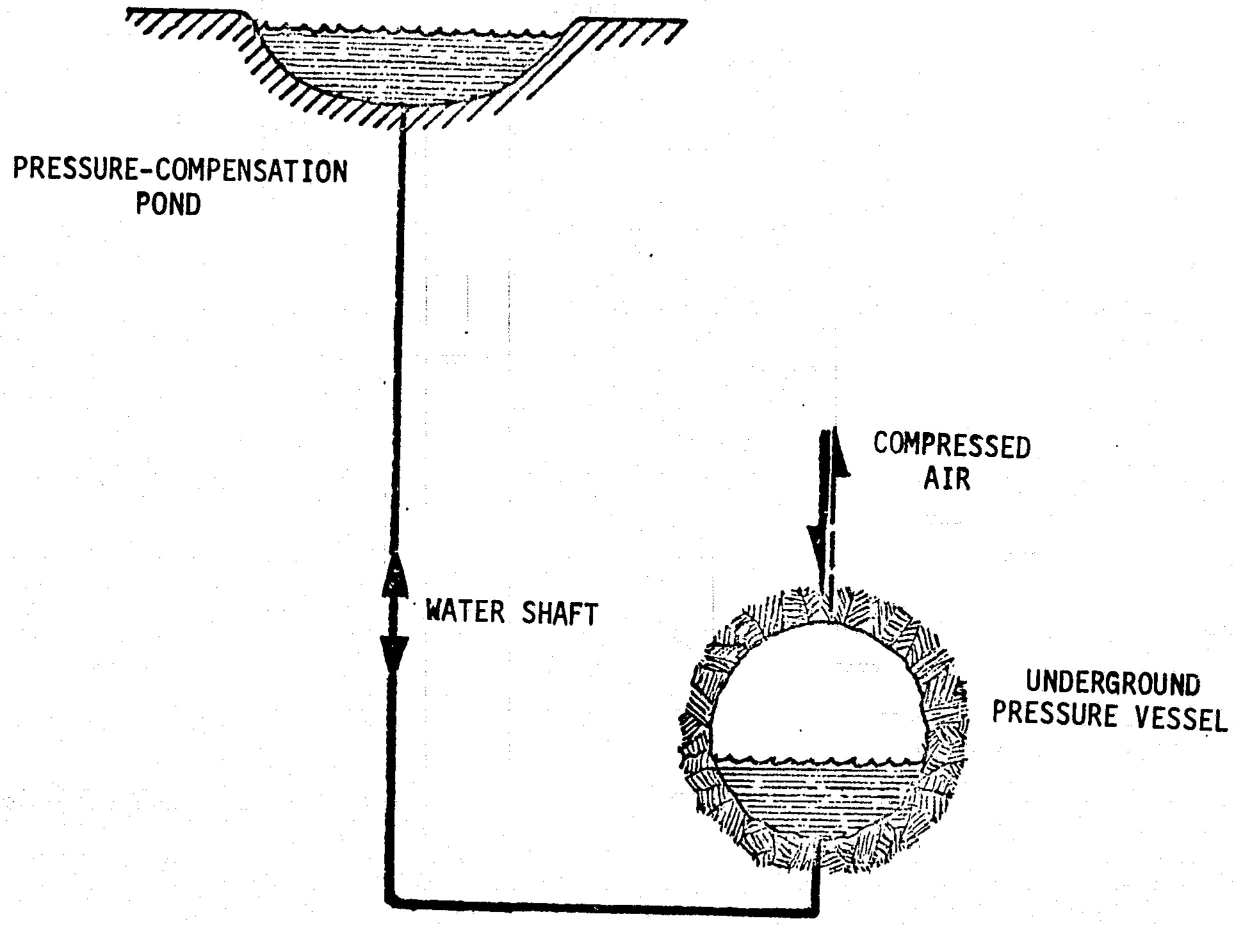


FIGURE 7.7 CONSTANT PRESSURE AIR STORAGE

Inputs

Parameter/Port		Description	Units
M	1	Inlet air mass flow rate	lb/h
CP		Air heat capacity (D = $7.2X10^{-5}$)	kwh/lb°F
T	1	Inlet air temperature	°F
TØ		Minimum air temperature (D = 60)	°F
NU		Leakage coefficient (D = 0.0008)	h^{-1}
R		Gas constant (D = $2.009X10^{-5}$)	kwh/lb
VM		Maximum storage capacity (D = $1.2X10^{6}$)	ft^3
PR	1	Vessel pressure (D = 147)	psi
LE		Life expectancy of vessel	years
CV		Vessel capacity cost (D = 0.22)	$\$/ft^3$
RE	1	Load request	kw
EF	1	Input product efficiency	-
MP	1	Input maximum power	kw
MDE		Mass threshold for priority resequencing	lb
MD		Maximum charge or discharge mass flow rate	lb
TM		Maximum allowable air temperature (D = 120)	°F
TEM		Maximum allowable inlet temperature	°F
CM		Maintenance cost/year	$

Outputs

Variable/Port			
E		Stored energy (state)	kwh
M	2	Outlet mass flowrate	lb/hr
T	2	Storage temperature	°F
V		Storage volume	ft^3
CCØ		Cost of vessel/year	$
MS		Mass of air in storage (state)	lb
MP	2	Maximum output power	kw
RE	2	Maximum charging rate	kw
INT		Interrupt priority flag	-
TSØ		Minimum air temperature (=TO)	°F

D - Default values supplied

Outputs Cont.

Variable/Port			
PR	2	Vessel pressure (=PR1)	psi
P	2	Output power (discharge)	kw
MDM		Maximum allowable mass flow rate (=MD)	

Statistics

EU	Maximum stored energy	kwh
VU	Maximum storage volume	ft^3

The pneumatic storage vessel calculation sequence and default values assume a 10atm cavern approximately 340 ft. below ground and sized for storage of 120kw for 24 hours. A maximum cavern wall temperature of 120°F is assumed. Cost estimates for the vessel were estimated from the results of Reference 1, with cost scaling by .05 to account for plant size differences.

[1] "Preliminary Feasibility Evaluation of Compressed Air Storage Power Systems," United Technologies AER 74-00242, December 1976.

Calculation Sequence

TINC = integration step size, hr

C1 = conversion constant = 5.43×10^{-5} $\frac{\text{kwh/ft}^3}{\text{psi}}$

1) Vessel Cost

$$\text{CCØ} = \text{CV*VM/LE}$$

2) Storage temperature

$$\text{T2} = \frac{\text{E}}{\text{CP*MS}} + \text{TØ}$$

3) Storage volume

$$\text{V} = \text{MS*}\frac{\text{R*(T2+460)}}{\text{PR1*C1}}$$

4) Maximum Mass and charging rate

$$\text{MSM} = \text{VM*}\frac{\text{(PR1*C1)}}{\text{R*(T2+460)}}$$

$$\text{MD1} = \text{MIN(MDM,(MSM-MS)/TINC)}$$

$$\text{RE2} = \text{min}\left\{\text{MP1, MD1*CP*(TEM-TØ)}\right\}\text{/EF1}$$

5) Mass flow out (discharge mode)

$$\text{M2} = \frac{\text{RE1}}{\text{CP*(T2-TØ)}}$$

$$\text{P2} = \text{RE1}$$

6) Maximum discharge rate

$$\text{MD} = \text{MIN(MDM,(MS-MDE)/TINC)}$$

$$\text{MP2} = \text{CP*(T2-TØ)*MD}$$

Calculation Sequence Cont.

7) Stored energy rate

$$\dot{E} = CP*(T1-T0)*M1 - NU*E - RE1$$

8) Stored mass rate

$$\dot{MS} = M1 - M2$$

9) Priority interrupt

If $MS \leq MDE$, $INT = 1$

If $MS > 2*MDE$ and $INT = 1$, $INT = 0$

If $MS \geq MSM$, $INT = -1$

If $MS < MSM-MDE$ and $INT = -1$, $INT = 0$

If $T2>TM$ write diagnostic and set $INT = -1$

If $MS < MDE$ or $MS > MSM$ write diagnostic

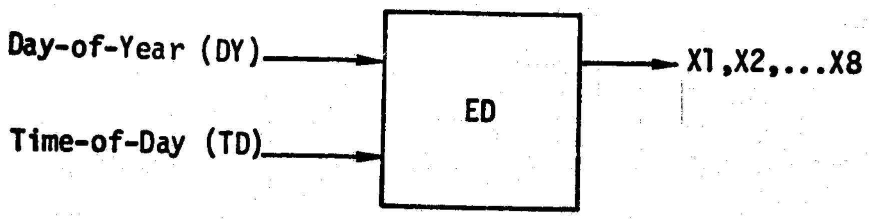
10) Compute Statistics and Costs

```
CCS
      SUBROUTINE CS( E,DE,IE,MS,DMS,IMS,M2,T2,V,CC,MP2,RE2,INT,TSO,PR2
     1 ,P,MDM,EU,VU,M1,CP,T,TO,R,VM,PR1,LE,NU,CV,RE1,EF1,MP1,MDE,MD,TM
     2 ,TEM,CM)
C
C   PURPOSE     PERFORMANCE MODEL OF CONSTANT PRESSURE STORAGE VESSEL
C
C   METHOD      ENERGY IN STORAGE COMPUTED AS A FUNCTION MASS AND
C
C               INLET TEMPERATURE.
C
C   WRITTEN BY F.O. MAHONY
C                                         VERSION 1, MARCH 23 1977
C  CALL SEQUENCE
C
C         OUTPUTS
C               E    - STORED ENERGY (STATE VARIABLE), KWH
C               DE   - STORED ENERGY DERIVATIVE, KW
C               IE   - STATUS INDICATOR FOR E
C               MS   - MASS OF AIR IN STORAGE (STATE VARIABLE), LB
C               DMS  - AIR FLOW RATE, LB/HR
C               IMS  - STATUS INDICATOR FOR MS
C               M2   - OUTLET MASS FLOWRATE, LB/HR
C               T2   - STORAGE TEMPERATURE, DEG F
C               V    - STORAGE VOLUME, FT**3
C               CC   - COST OF VESSEL/YEAR, $
C               MP2  - MAXIMUM OUTPUT POWER, KW
C               RE2  - MAXIMUM CHARGING RATE, KW
C               INT  - INTERRUPT PRIORITY FLAG
C               TSO  - MINIMUM AIR TEMPERATURE, DEG F
C               PR2  - VESSEL PRESSURE, PSI
C               P    - OUTPUT POWER (DISCHARGE), KW
C               MDM  - MAXIMUM ALLOWABLE MASS FLOW RATE, LB/HR
C               EU   - MAXIMUM STORED ENERGY, KWH
C               VU   - MAXIMUM STORAGE VOLUME, FT**3
C
C         INPUTS
C               M1   - INLET AIR MASS FLOW RATE, LB/HR
C               CP   - AIR HEAT CAPACITY, KWH/LB-DEG F
C               T    - INLET AIR TEMPERATURE, DEG F
C               TO   - MINIMUM AIR TEMPERATURE, DEG F
C               R    - GAS CONSTANT, KWH/LB-DEG R
C               VM   - MAXIMUM STORAGE CAPACITY, FT**3
C               PR1  - VESSEL PRESSURE, PSI
C               LE   - LIFE EXPECTANCY OF VESSEL, YEARS
C               NU   - LEAKAGE COEFFICIENT   , 1/HR
C               CV   - VESSEL CAPACITY COST   , $/FT**3
C               RE1  - LOAD REQUEST, KW
C               EF1  - INPUT PRODUCT EFFICIENCY
C               MP1  - INPUT MAXIMUM POWER, KW
C               MDE  - RESERVOIR THRESHOLD MASS FOR PRIORITY
C                      RESEQUENCING, LB
C               MD   - MAXIMUM CHARGE / DISCHARGE MASS FLOW RATE, LB/HR
C               TM   - MAXIMUM ALLOWABLE STORAGE TEMPERATURE, DEG F
C               TEM  - MAXIMUM ALLOWABLE INLET TEMPERATURE, DEG F
C               CM   - MAINTENANCE COST / YEAR , $
C
```

```
      COMMON /CIMPL/IMPL,ICNT/CTIME/ TIME /CSIMUL/DUM(7),TMAX
      COMMON /COST/ CCI,CMI
      REAL NU,MS,MP2,INT,MDM,M1,LE,MP1,MDE,MD,M2,MSM,MD1
C
C
      IF(IMPL.GT.0) GO TO 100
C
      IF(CP .EQ. .99999) CP = 72.0E-6
      IF(TM .EQ. .99999) TM= 120.0
      IF(TO .EQ. .99999) TO = 60.0
      IF(NU .EQ. .99999) NU = 0.0008
      IF(R  .EQ. .99999) R   = 2.009E-5
      IF(VM .EQ. .99999) VM = 1.2E+6
      IF(PR1.EQ. .99999) PR1 = 147.0
      IF(CV .EQ. .99999) CV  = 0.22
      RE1=0.0
C
      TMAX1 = TMAX*0.99999
      TINC=DUM(7)
      TSO=TO
C
      CC = CV*VM/LE
      C1 = 5.43E-5
C
      INT = 0.
      PR2=PR1
      EU= 0.
      VU= 0.
  100 CONTINUE
C
C
C
C                          STORAGE TEMPERATURE
C
      T2 = E/CP/MS+TO
C
C                          STORAGE VOLUME
C
      V = MS*R*(T2+460.0)/PR1/C1
C
C                          MAXIMUM MASS AND  CHARGING RATE
C
      MDM=MD
      MSM = VM*PR1*C1/R/(T2+460.0)
      MD1=AMIN1(MDM,AMAX1(0.,(MSM-MS)/TINC))
      RE2 = AMIN1(MP1,MD1*CP*(TEM-TO))/EF1
C
C                          MASS FLOW OUT (DISCHARGE)
C
      M2 = RE1/CP/(T2-TO)
      P = RE1
C
C                          MAXIMUM DISCHARGE RATE
      AMD=AMAX1(0.,(MS-MDE)/TINC)
      MDM=AMIN1(MDM,AMD)
      MP2 = CP*(T2-TO)*MDM
C
C                          STORED ENERGY RATE
```

```
C
      IF(IE.NE.0) DE=CP*(T-TO)*M1 - NU*E -RE1
C
C                      STORED MASS RATE
C
      IF(IMS.NE.0) DMS=M1-M2
C
C                                     PRIORITY INTERRUPT LOGIC
      IF(MS .LE. MDE) INT=1.
      IF(MS.GT. 2.*MDE .AND. INT.EQ.1.) INT=0.
      IF(MS .GE.MSM) INT = -1.
      IF(MS.LT. MSM-MDE .AND. INT.EQ.-1.) INT=0.
      IF(T2 .GT. TM) INT= -1.
C
C
      IF(IMPL.LE.1)RETURN
      IF(IMPL.GT.2)GO TO 200
      IF(T2.LT.TM)GO TO 10
      WRITE(6,1000)T2,TM
      ICNT=ICNT+1
   10 IF(MS.GT.MDE)GO TO 20
      WRITE(6,1010)MS,MDE
      ICNT=ICNT+1
   20 IF(MS.LT.MSM)GO TO 200
      WRITE(6,1020)MS,MSM
      ICNT=ICNT+1
C                      STATISTICS
C
  200 EU = AMAX1(EU,E)
      VU = AMAX1(VU,V)
C
      IF(TIME.LT.TMAX1) RETURN
      CCI = CCI+CC
      CMI=CMI+CM
C
      RETURN
 1000 FORMAT(1H0,23HCS STORAGE TEMPERATURE,F12.3,22HGREATER THAN ALLOWAB
     1LE,  F12.3)
 1010 FORMAT(1H0,25HCS MASS OF AIR IN STORAGE,F12.3,
     1 26H   BELOW MINIMUM ALLOWABLE,F12.3)
 1020 FORMAT(1H0,25HCS MASS OF AIR IN STORAGE,F12.3,
     1 28H   EXCEEDS MAXIMUM ALLOWABLE,F12.3)
      END
```

7.8 ENVIRONMENTAL DATA (TMY TAPE)



This component reads data values from the Typical Meteorological Year (TMY) tapes or data with a similar format structure such as the University of Wisconsin insolation and environmental data tape or the SOLMET tapes. Only one ED component is allowed per model. (Unit 1 is reserved for the input tape.) The file structure assumes hourly recorded data with one record or card image per hour of data. Twenty-four hourly records are read into core at a time and linear interpolation is used to obtain the output values at the current simulation time. The component TI is used to supply the time inputs DY and TD. Standard outputs with the TMY tape are direct and global solar insolation, dry bulb temperature, and wind speed. For non-standard outputs or non-TMY format tapes the user may specify the input format to read one to eight data variables. The following limitations apply in this case:

1) Time information is decoded in integer month (1-12), day (1-31), and hour (0-24) format.

2) Output variables are decoded in F or E format, even if recorded in integer format.

3) Where data is missing, fill in with 9's is assumed. The code checks for certain 9 fill values, namely 99., 999., 9999., and 99999. If any one of these values is read, then the corresponding data input is replaced with 0. or the previous value, depending on the sign of IND. (However, one must use FN.0 format N=2,3,4,5 for this option and a scale multiplier if necessary to obtain the desired exponent.)

Inputs/Port[1]	Description	Units
NX	Number of output variables (default = 4, max = 8)	-
IND	Indicator function: 0 = no read ±1 = standard format and units (default) ±2 = user-specified format and units IND>0 sets missing data = 0 IND<0 sets missing data = previous value	

[1] Also see page 65 in Section 4.2 for inputs to procedure TMYRD. This procedure creates the data input file TAPE1 from the multi-station TMY tape.

Inputs/Port (cont'd)	Description	Units
TS*	Time shift of data (default = -0.5)	hours
TD	Time of day (0-24)	hours
DY	Day of year (1-365)	-
M1	Units multiplier for X1 (default = 1)	-
.	.	
.	.	
.	.	
M8	Units multiplier for X8 (default = 1)	-
A1	Units addition factor for X1 (default = 0)	-
.	.	
.	.	
.	.	
A8	Units addition factor for X8 (default = 0)	-

*Compensation term since solar radiation data is an integrated total over the observation interval.

Outputs/Port	Description	Units
X1	1st output variable (IND = ±1: beam radiation in w/m^2	-
X2	2nd output variable (IND = ±1: global horizontal radiation in w/m^2)	-
X3	3rd output variable (IND = ±1: dry bulb temperature in °C)	-
X4	4th output variable (IND = ±1: wind speed in m/s)	-
.		
.		
.		
X8	8th output variable	-

Format Specification

A user-specified format may be input in order to select non-standard environmental outputs or to read a tape other than the TMY insolation tape. The following sequence of data cards is recommended for insertion in the model generation input following the MODEL DESCRIPTION command:

```
FORTRAN STATEMENTS
DIMENSION FMT(7)
COMMON/READER/N,FMT
DATA FMT/XXH(...)/,N/NN/
```

where the format specification contains XX-2 characters inserted after 'XXH(' and followed by ')', and NN = the number of characters per data record.

ED

The format specification must conform to the following rules:

1) The first two words read are station and year identifying information. These words must be either A format or nH format with up to six characters for station and two characters NN for year 19NN.

2) The next three words are two-digit integers containing month (1-12), day (1-31), and hour (0-24) information.

3) The next one to eight words specify the location of the output variables X1...X8 and must be given in F or E format.

NOTE: The tab or column spacing control T may be used to read data from files which are not ordered as in 1) to 3), e.g., (T71, A5, T1, A2,...).

For example, the standard TMY tape format specification (neglecting blanks) is

Station	Yr-Mo-Dy-Hr	Beam Rad.	Global Rad.	Temp	Wind
(A5,	A2,3I2,11X,	F4.0 ,26X,	F4.0,45X,	F4.1,7X,	F4.1)

and N = 132, XX = 46.

The general format for variables on the TMY tape is summarized in Figure 7.8.

								SOLAR RADIATION OBSERVATION									
	WBAN STN #	SOLAR TIME				LST TIME	ETR KJ/m^2	RADIATION VALUES KJ/m^2									
		YR	MO	DY	HRMN			DIRECT	DIFFUSE	NET	TILTED	GLOBAL			A	B	SUNSHINE MIN
												OBS	ENG COR	STD YR COR			
	XXXXX	XX	XX	XX	XXXX	XXXX	XXXX	1XXXX	1XXXX	1XXXX	1XXXX	1XXXX	1XXXX	1XXXX	1XXXX	1XXXX	XX
FIELD NUMBER	002	003				004	101	102	103	104	105	106	107	108	109	110	111

SURFACE METEOROLOGICAL OBSERVATION														
OBS TIME LST	CEILING dam	SKY COND	VSBY hm	WEATHER	PRESSURE kPa		TEMP °C		WIND				SNOW COVER
					SEA LEVEL	STA-TION	DRY BULB	DEW-PT.	DIR deg	SPD m/s	TOTAL	OPAQUE	
XX	XXXX	1XXXX	XXXX	XXXXXXXX	XXXXX	XXXXX	XXXX	XXXX	XXX	XXXX	XX	XX	X
201	202	203	204	205	206		207		208		209		210

TAPE FIELD NUMBER	RECORD POSITIONS	DESCRIPTION
002	01-05	WBAN STATION NUMBER
003	06-15	SOLAR TIME (YR,MO,DAY,HOUR,MINUTE)
004	16-19	LOCAL STANDARD TIME (HR AND MINUTE)
101	20-23	EXTRATERRESTRIAL RADIATION
102	24-28	DIRECT RADIATION
103	29-33	DIFFUSE RADIATION
104	34-38	NET RADIATION
105	39-43	GLOBAL RADIATION ON A TILTED SURFACE
106	44-48	GLOBAL RADIATION ON A HORIZONTAL SURFACE-OBSERVED DATA
107	49-53	GLOBAL RADIATION ON A HORIZONTAL SURFACE-ENGINEERING CORRECTED DATA
108	54-58	GLOBAL RADIATION ON A HORIZONTAL SURFACE-STANDARD YEAR CORRECTED DATA
109,110	59-68	ADDITIONAL RADIATION MEASUREMENTS
111	69-70	MINUTES OF SUNSHINE
201	71-72	TIME OF COLLATERAL SURFACE OBSERVATION (LST)
202	73-76	CEILING HEIGHT (DEKAMETERS)
203	77-81	SKY CONDITION
204	82-85	VISIBILITY (HECTOMETERS)
205	86-93	WEATHER
206	94-103	PRESSURE (KILOPASCALS)
207	104-111	TEMPERATURE (DEGREES CELSIUS TO TENTHS)
208	112-118	WIND (SPEED IN METERS PER SECOND TO TENTHS)
209	119-122	CLOUDS
210	123	SNOW COVER INDICATOR

FIGURE 7.8 TMY TAPE FORMAT

A complete description of the available data, and the meaning of the recorded outputs, is contained in the SOLMET user's manual [3]. The TMY tape was derived from SOLMET tapes of the 26 stations with rehabilitated solar radiation data, and has the same format as the SOLMET tapes except that tape deck number and detailed cloud data have been omitted. Table 7.8 shows the identity and location of the 26 stations on the TMY tape.

Calculation Sequence

If IND = 0 Return

1) INITIALIZATION (first pass only)

- Set defaults and initialize LTD = -1
- Read first data block and write out identification information. (Error exit to 6))
- Go to 4)

2) Table Interpolation for Output (DY = DYF)

- If DY > DYF go to 3)
- If DYF > DY go to 5)
- If LTD = TD return (LTD = last time C(I,J) was accessed)
- X(I) = TBLU1 (TD, TO, C(1,I),0,24)*M(I)+A(I) I = 1,...NX
- LTD = TD
- Return

3) Read One or More Data Blocks ($DY > DYF$)

- Read DY-DYF data blocks. (Error exit or EOF exit to 6))

4) Decode Using Specified Format

- Decode day-of-year (DYF) and time information (TO) and put output variables in array C(I,J) I=1,24 and J=1,NX. Check for missing data values in C(I,J).
- Go to 2)

5) Backspace the File ($DYF > DY$)

- Backspace and read first data block
- Decode day-of-year (DYF)
- Go to 4) if $DYF \leq DY$. Otherwise print diagnostic and stop.

6) Read Error or EOF Encountered

- Print diagnostic and stop.

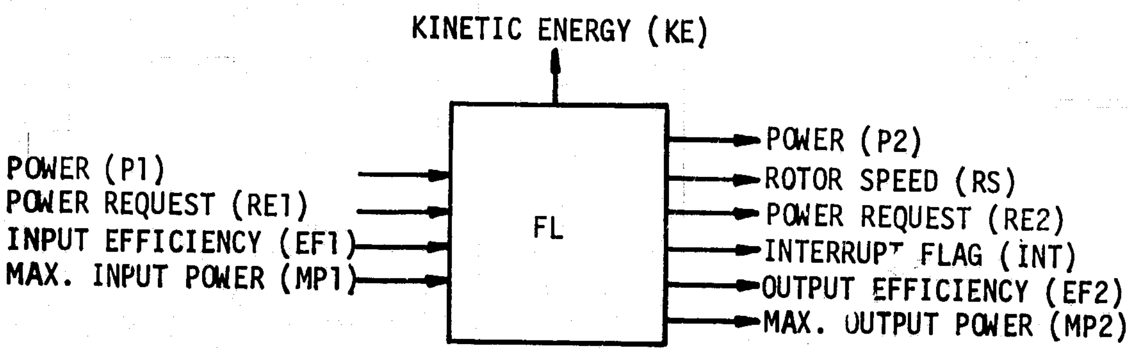
ED

TABLE 7.8 TMY TAPE STATIONS AND LOCATION

STATION NUMBER	WBAN IDENTIFIER	STATION	LATITUDE	LONGITUDE
1	3927	Fort Worth, Texas	32°50'	97°03'
2	3937	Lake Charles, Louisiana	30°07'	93°13'
3	3945	Columbia, Missouri	38°49'	92°13'
4	12832	Apalachicola, Florida	29°44'	84°59'
5	12839	Miami, Florida	25°48'	80°16'
6	12919	Brownsville, Texas	25°54'	97°26'
7	13880	Charleston, South Carolina	32°54'	80°02'
8	13897	Nashville, Tennessee	36°07'	86°41'
9	13985	Dodge City, Kansas	37°46'	99°58'
10	14607	Caribou, Maine	46°52'	68°01'
11	14837	Madison, Wisconsin	43°08'	89°20'
12	23044	El Paso, Texas	31°48'	106°24'
13	23050	Albuquerque, New Mexico	35°03'	106°37'
14	23154	Ely, Nevada	39°17'	114°51'
15	23183	Phoenix, Arizona	33°26'	112°01'
16	23273	Santa Maria	34°54'	120°27'
17	24011	Bismarck, North Dakota	46°46'	100°45'
18	24143	Great Falls, Montana	47°29'	111°22'
19	24225	Medford, Oregon	42°22'	122°52'
20	24233	Seattle-Tacoma, Washington	47°27'	122°18'
21	93193	Fresno, California	36°46'	119°43'
22	93729	Cape Hatteras, North Carolina	35°16'	75°33'
23	93734	Washington, D.C.	38°59'	77°28'
24	94701	Boston, Massachusetts	42°22'	71°03'
25	94728	New York, New York	40°47'	73°58'
26	94918	North Omaha, Nebraska	41°22'	96°01'

```
CED
      SUBROUTINE ED(X,X2,X3,X4,X5,X6,X7,X8,NX,IND,TS,TD,DY,
     1M1,M2,M3,M4,M5,M6,M7,M8,A1,A2,A3,A4,A5,A6,A7,A8)
C
C           PURPOSE   THIS COMPONENT READS THE TYPICAL METEOROLOGICAL
C                     YEAR TAPE WITH A STRUCTURE SIMILAR TO THE
C                     SOLMET DATA TAPE. USER MAY SPECIFY FORMAT FOR NON-
C                     STANDARD TAPES
C
C           WRITTEN BY Y.K.CHAN, 10-5-78, VERSION 1
C
C           METHOD    TWENTY FOUR HOURLY RECORDS ARE READ INTO CORE
C                     AT A TIME AND LINEAR INTERPOLATION IS USED TO
C                     OBTAIN THE OUTPUT AT CURRENT SIMULATION TIME.
C
C         CALL SEQUENCE
C              OUTPUTS
C                       X1,...,X8  -OUTPUT VARIABLES AT CURRENT TIME
C                       X1   -BEAM RADIATION IF IND=+-1, W/M2
C                       X2   -GLOBAL RADIATION IF IND=+-1, W/M2
C                       X3   -DRY BULB TEMPERATURE IF IND=+-1,  C
C                       X4   -WIND SPEED IF IND=+-1,M/S
C              INPUTS
C                       NX     -NUMBER OF OUTPUT VARIABLES(DEFAULT=4,MAX=8)
C                       IND    -INDICATOR FUNCTION
C                               0=NO READ
C                               +-1=STANDARD FORMAT AND UNITS(DEFAULT)
C                               +-2=USER SPECIFIED FORMAT AND UNITS
C                               >0,SETS MISSING DATA TO 0
C                               <0,SETS MISSING DATA TO PREVIOUS VALUE
C                       TS     -TIME SHIFT OF DATA(DEFAULT=-0.5)
C                               (COMPENSATION TERM SINCE SOLAR RADIATION
C                               DATA IS AN INTEGRATED TOTAL, USUALLY OVER 1 HOUR)
C                       TD     -CURRENT TIME OF DAY(0-24)
C                       DY     -CURRENT DAY OF YEAR(1-365)
C                       M1,...,M8  -UNITS MULTIPLIERS FOR X1,...,X8
C                               DEFAULT M1=...=M8=1
C                       A1,...,A8  -ADDITION FACTOR FOR X1,...,X8
C                               DEFAULT A1=...=A8=0
C
      DIMENSION X(8),A(8),FRMT(7),FMT(7),C(24,8),AA(336),IB(5),B(8),
     1 CL(8),TO(24),DM(12)
      COMMON /READER/N,FMT
      COMMON /CIMPL/IMPL,ICNT,ITEST
      REAL NX,IND,M1,M2,M3,M4,M5,M6,M7,M8,M(8),LTD
      DATA FRMT/70H(A5,A2,3I2,11X,F4.0,26X,F4.0,45X,F4.1,7X,F4.1)
     1                                                    /
      DATA DM/0.,31.,59.,90.,120.,151.,181.,212.,243.,
     1 273.,304.,334./
C
      IF(ABS(IND).LE..1)RETURN
C
C              INITIALIZATION
C
      IF(IMPL.GT.0)GO TO 100
      IF(NX.EQ..99999)NX=4
      IF(TS.EQ..99999)TS=-.5
      INX=NX+.1
      M(1)=M1
      M(2)=M2
      M(3)=M3
      M(4)=M4
      M(5)=M5
      M(6)=M6
      M(7)=M7
      M(8)=M8
      A(1)=A1
      A(2)=A2
      A(3)=A3
      A(4)=A4
      A(5)=A5
      A(6)=A6
      A(7)=A7
      A(8)=A8
      DO 11 I=1,INX
      IF(M(I).EQ..99999)M(I)=1.
   11 IF(A(I).EQ..99999)A(I)=0.
C
C                     SET DEFAULT TAPE RECORD FORMAT TO STANDARD
      IF(ABS(IND).GT.1.01)GO TO 2
      M(1)=1./3.6
      M(2)=1./3.6
      DO 3 I=1,7
    3 FMT(I)=FRMT(I)
      N=132
    2 CONTINUE
      LTD=-1.
      DO 10 J=1,INX
   10 CL(J)=0.
C
C                 READ FIRST DATA BLOCK
C
      IREWIN=0
      J13=N/10
      IF(N.GT.10*J13)J13=J13+1
      DO 20 J=1,24
      J1=J13*(J-1)+1
      J2=J1+J13-1
      BUFFER IN (1,0)(AA(J1),AA(J2))
      IF(UNIT(1))20,600,600
   20 CONTINUE
      DECODE(N,FMT,AA(1))IB,B
      WRITE(6,308)IB(1),IB(2)
  308 FORMAT(1H0,3X,*ED  STATION ID=*,A5,10X,*YEAR 19*,A2)
      GO TO 400
C
C
  100 CONTINUE
C
  200 CONTINUE
C
C                 INTERPOLATION FOR OUTPUT IF CURRENT DAY OF YEAR HAS
C                 BEEN LOCATED
C
      TD1=TD
      IF(DY.GT.(DYF+.1))GO TO 300
      IF(DY.LT.(DYF-.1))GO TO 500
   24 IF(LTD.EQ.TD)RETURN
      DO 201 I=1,INX
  201 X(I)=TBLU1(TD1,TO,C(1,I),0,24)*M(I)+A(I)
      LTD=TD
      RETURN
C
  300 CONTINUE
C
C             IF CURRENT DAY OF YEAR HAS NOT BEEN LOCATED, READ MORE TAPE
C
      ID=DY-DYF+.1
      IF(ID.EQ.1  .AND.  TD.LT. .00001) TD1=24.
      IF(TD1.EQ. 24.) GO TO 24
      DO 30 I=1,ID
      DO 301 J=1,24
      J1=J13*(J-1)+1
      J2=J1+J13-1
      BUFFER IN(1,0)(AA(J1),AA(J2))
      IF(UNIT(1))301,600,600
  301 CONTINUE
   30 CONTINUE
C
C             DECODE DATA AND TIME OF DAY
C
  400 CONTINUE
      DO 402 I=1,24
      I1=J13*(I-1)+1
      DECODE (N,FMT,AA(I1))IB,B
      DO 401 J=1,INX
      C(I,J)=B(J)
      CIJ=C(I,J)
      IF((CIJ.EQ.99.).OR.(CIJ.EQ.999.).OR.(CIJ.EQ.9999.).OR.
     1(CIJ.EQ.99999.))C(I,J)=CL(J)
      IF(IND.LT.0.)CL(J)=C(I,J)
  401 CONTINUE
      TO(I)=IB(5)+TS
  402 CONTINUE
      DYF=IB(4)+DM(IB(3))
      GO TO 200
C
  500 CONTINUE
C
C             IF DAY OF YEAR ON TAPE IS PAST CURRENT DAY OF YEAR,
C             REWIND TAPE.
      IF(IREWIN.GT.0)GO TO 507
      REWIND 1
      IREWIN=IREWIN+1
      DO 501 J=1,24
      J1=J13*(J-1)+1
      J2=J1+J13-1
      BUFFER IN(1,0)(AA(J1),AA(J2))
      IF(UNIT(1))501,600,600
  501 CONTINUE
      DECODE(N,FMT,AA(1))IB,B
      DYF=IB(4)+DM(IB(3))
      IF(DYF.LT.(DY+.1))GO TO 400
  507 WRITE(6,508)
  508 FORMAT(1H0,* INPUT ERROR, DAY OF YEAR DY IS OUT OF RANGE*)
      STOP
C
C                   IF ERROR IN READ, PRINT DIAGNOSTICS
  600 WRITE(6,608)
  608 FORMAT(1H0,*ED  TAPE INPUT ERROR OR EOF*)
      STOP
      END
```

7.9 FLYWHEEL/CLUTCH



The flywheel model is a first order differential equation for kinetic energy which is driven by input power when charging and by a load request when discharging. Power losses include clutch losses versus shaft speed and torque, windage losses, and friction losses due to bearing and seals. Shaft speed is determined analytically from kinetic energy. Priority interrupt logic is activated if minimum or maximum capacity levels are reached.

Basic Equations

$$KE = k*\omega^2$$

$$\dot{KE} = P_{IN} - P_{OUT} - C_1*\omega - C_2*\omega^{2.8},$$

where k, C_1, C_2 are flywheel constants

ω = rotor speed in rad/sec

P_{IN} = input power - clutch losses

P_{OUT} = output load request

Tables	Description	Units
CLØ	Clutch losses versus rotor speed (rpm) and torque (ft-lb), when engaged (Table dimension = 90)	kw
CL1	Clutch losses versus rotor speed (rpm) when disengaged (Table dimension = 17)	kw

Inputs

Parameter/Port		Description	Units
PR		Pressure in vacuum housing	psi
HM		Moment of inertia[1]	slug-ft^2
RF		Radius of flywheel	ft
SR		Shaft radius	ft
WT		Flywheel weight	lb
KF		Coefficient of friction	-
ZE		Width of flywheel at tip	ft
C2		Windage loss coefficient (analytic default)	-
P	1	Input power	kw
EF	1	Input product efficiency	-
MP	1	Input maximum charging rate	kw
RAP		Rated power, charge or discharge	kw
RE	1	Discharge load request	kw
EØ		Minimum allowable storage capacity	kwh
E1		Maximum allowable storage capacity	kwh
EDE		Energy deadband for priority resequencing	kwh
CM		Maintenance cost/year	$
CC		Capital cost/year	$

Outputs

Variable/Port	Description	Units
RS	Rotor speed	rpm
KE	Kinetic energy (state)	kwh

[1] Includes physical drive system.

Outputs Cont.

Variable/Port		Description	Units
TØ		Input torque (charging)	ft-lb
T1		Output torque (discharging)	ft-lb
P	2	Output power	kw
PLØ		Clutch losses (charging)	kw
PL1		Clutch losses (discharging)	kw
EF	2	Output efficiency	-
MP	2	Maximum output power	kw
INT		Priority interrupt flag	-
RE	2	Maximum charging power request	kw

Statistics

ME	Maximum stored energy	kwh
MPC	Maximum charge rate	kw
MPD	Maximum discharge rate	kw
SPC	Sum of charging energy	kwh
SPD	Sum of discharging energy	kwh

Calculation Sequence

1) Compute flywheel constants

$$k = \frac{1}{2} * HW * / 3.76616 * 10^{-7}$$

$$C_1 = KF * WT * SR * / 1.3558 * 10^{-3}$$

$$C_2 = C_o * PR^{0.8} * RF^{4.6} * (1 + 2.3 * ZE/RF) \quad \text{(DEFAULT)}$$

$$C_o = 1.0946 * 10^{-7}$$

If KE < E0 or KE > E1 write diagnostic

2) Compute rotor speed

$$\omega = \sqrt{KE/k}$$

$$RS = \omega * (60/2\pi)$$

$$P_{IN} = 0$$

3) Compute power losses and net power when charging

If P1 = 0, set TØ = PLØ = P_{IN} = 0 and go to 4)

$$T0 = P1 * 737.6/\omega$$

PLO = CLØ(RS,TØ)

P_{IN} = P1 − PLØ

If $P_{IN} < 0$, write diagnostic

4) Compute power losses and output power when discharging

If RE1 = 0, set T1 = PL1 = P2 = P_{OUT} = 0 and go to 5)

$$T1 = RE1 * 737.6/\omega$$

PL1 = CLØ(RS, −T1)

Calculation Sequence

4) Cont.

$P2 = RE1 - PL1$

$P_{OUT} = RE1$

If P2 < 0, set P2 = 0. and write diagnostic

5) Compute power losses when disengaged

If P1 > 0 or RE1 > 0, go to 6)

$P_{OUT} = CL1(RS)$

6) Flywheel kinetic energy rate

$$\dot{KE} = P_{IN} - P_{OUT} - C_1 * \omega - C_2 * \omega^{2.8}$$

7) Maximum Input (charging power)

$TM = MP1*737.6/\omega$

MPØ = MP1 - CLØ(RS,TM)

If MPØ ≤ 0, write diagnostic and go to 8)

EFØ = EF1*MPØ/MP1

RE2 = MIN(MPØ,RAP),(EI-KE)/TINC)/EFØ

8) Output efficiency and maximum power

RAP1 = MIN(RAP,(KE-EØ)/TINC), TINC = integration step

$TM = RAP1*737.6/\omega$

MP2 = RAP1 - CLØ(RS,-TM)

Calculation Sequence

8) Cont.

If $MP2 < 0$ write diagnostic

$EF2 = MP2/RAP1$

If $RE1 > 0$, $EF2 = P2/RE1$

9) Priority interrupt logic

If $KE \leq E\O$, $INT = 1$

If $KE > E\O + EDE$ and $INT=1$, $INT=0$

If $KE \geq E1$, $INT = -1$

If $KE < E1 - EDE$ and $INT= -1$, $INT=0$

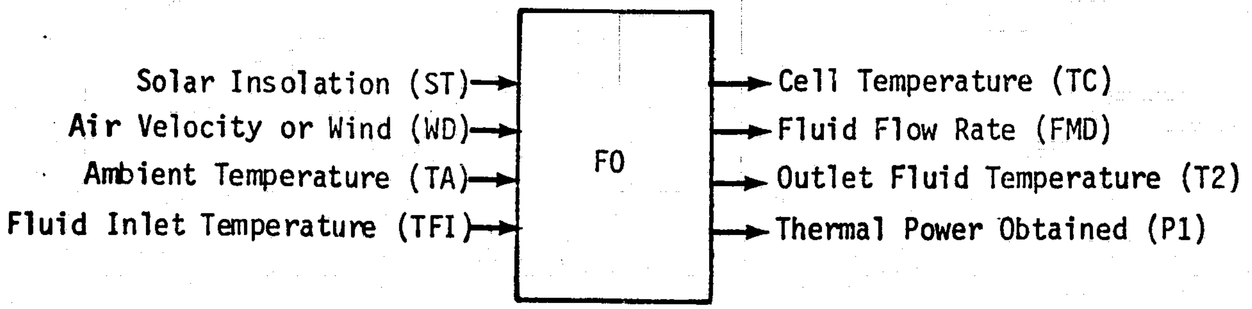
10) Compute Statistics and Costs

```
CFL
      SUBROUTINE FL(CL0,CL1,RS,KE,KED,IKE,T0,T1,P2,PL0,PL1,EF2,MP2,INT,
     1   RE2,ME,MPC,MPD,SPC,SPD,  PR,HM,RF,SR,WT,KF,ZE,C2,P1,EF1,MP1,RAP
     2   ,RE1,E0,E1,EDE,CM,CC)
C
C                 PURPOSE   MODEL OF FLYWHEEL CAPABLE OF ABSORBING POWER
C                           AND OF DELIVERING POWER ON REQUEST
C
C                 METHOD    OUTPUT POWER AND KINETIC ENERGY COMPUTED FROM
C                           POWER REQUEST AND INPUT POWER
C
C                 WRITTEN BY Y.K.CHAN             VERSION 1, JUNE 17, 1977
C
C   CALL SEQUENCE
C          TABLES
C                 CL0 -CLUTCH LOSSES VS ROTOR SPEED(RPM) AND TORQUE(FT-LB),KW
C                 CL1 -CLUTCH LOSSES VS ROTOR SPEED(RPM) WHEN DISENGAGED,KW
C          OUTPUTS
C                 RS  -ROTOR SPEED, RPM
C                 KE  -KINETIC ENERGY(STATE),KWH
C                 KED -KINETIC ENERGY INCREASE RATE,KW
C                 IKE -INTEGRATOR CONTROL
C                 T0  -INPUT TORQUE (CHARGING), FT-LB
C                 T1  -OUTPUT TORQUE (DISCHARGING),FT-LB
C                 P2  -OUTPUT POWER,KW
C                 PL0 -CLUTCH LOSSES (CHARGING),KW
C                 PL1 -CLUTCH LOSSES (DISCHARGING),KW
C                 EF2 -OUTPUT EFFICIENCY
C                 MP2 -MAXIMUM OUTPUT POWER,KW
C                 INT -PRIORITY INTERRUPT FLAG
C                 RE2 -MAXIMUM CHARGING POWER REQUEST
C          STATISTICS
C                 ME  MAXIMUM STORED ENERGY,KWH
C                 MPC -MAXIMUM CHARGE RATE,KW
C                 MPD -MAXIMUM DISCHARGE RATE,KW
C                 SPC -SUM OF CHARGING POWER,KWH
C                 SPD -SUM OF DISCHARGING POWER,KWH
C          INPUTS
C                 PR  -PRESSURE IN VACUUM HOUSING,PSI
C                 HM  -MOMENT OF INERTIA,SLUG-FT2
C                 RF  -RADIUS OF FLYWHEEL,FT
C                 SR  -SHAFT RADIUS,FT
C                 WT  -FLYWHEEL WEIGHT,LB
C                 KF  -COEFFICIENT OF FRICTION
C                 ZE  -WIDTH OF FLYWHEEL AT TIP,FT
C                 C2  -WINDAGE COEFFICIENT (ANALYTIC DEFAULT)
C                 P1  -INPUT POWER,KW
C                 EF1 -INPUT PRODUCT EFFICIENCY
C                 MP1 -INPUT MAXIMUM CHARGING RATE,KW
C                 RAP -RATED POWER, CHARGE OR DISCHARGE,KW
C                 RE1 -DISCHARGE LOAD REQUEST,KW
C                 E0  -MINIMUM ALLOWABLE STORAGE CAPACITY,KWH
C                 E1  -MAXIMUM ALLOWABLE STORAGE CAPACITY,KWH
C                 EDE -ENERGY DEADBAND FOR PRIORITY RESQUENCING,KWH
C                 CM  -MAINTENANCE COST/YEAR,$
```

```
C                   CC   -CAPITAL COST/YEAR,$
C
      COMMON /CIMPL/IMPL,ICNT/CTIME/TIME/CSIMUL/DUM(7),TMAX
     X         /COST/CC1,CMI
      REAL KE,KED,MP2,INT,ME,MPC,MPD,KF,MP1,MP0,MPA,MPB
      DIMENSION CL0(1),CL1(1)
C
      IF(IMPL.GT.0)GO TO 10
      TINC=DUM(7)*.5
      IF(C2.EQ..99999)C2=(1.0946E-7)*(PR**.8)*(RF**4.6+2.3*ZE*(RF**3.6))
      INT=0.
      TMAX1=TMAX*.99999
C
      ME=0.
      RE1=0.
      MPC=0.
      MPD=0.
      SPC=0.
      SPD=0.
   10 CONTINUE
      AK=.5*HM*3.76616E-7
      C1=KF*WT*SR*1.3558E-3
      IF(((KE.GT.E0).AND.(KE.LT.E1)).OR.(IMPL.NE.2))GO TO 20
      IF(KE.LE.E0)WRITE(6,108)KE,E0
      IF(KE.GE.E1)WRITE(6,109)KE,E1
  109 FORMAT(1H0,26H FLYWHEEL KINETIC ENERGY ,F12.3,
     X 18H EXCEEDS CAPACITY ,F12.3)
  108 FORMAT(1H0,24H FLYWHEEL KINETIC ENERGY,F12.3,
     X 33H FALLS BELOW MINIMUM REQUIREMENT ,F12.3)
      ICNT=ICNT+1
   20 CONTINUE
      NNRS=CL0(2)
      NNT=CL0(3)
      M4=NNT+4
      MN4=M4+NNRS
      NNNRS=CL1(2)
      NNNRS4=NNNRS+4
      T0=0.
      T1=0.
      P2=0.
      PL0=0.
      PL1=0.
      RE2=0.
C
C                   COMPUTE ROTOR SPEED
C
  100 OMEGA=1.E-6
      IF(KE.GT.0.)OMEGA=SQRT(KE/AK)
      RS=OMEGA*30./3.14159
      PIN=0.
C
C                   COMPUTE POWER LOSSES AND NET POWER WHEN CHARGING
C
      IF(P1.EQ.0.)GO TO 200
      T0=P1*737.6/OMEGA
      PL0=TBLU2(RS,T0,CL0(M4),CL0(4),CL0(MN4),1,1,-NNRS,-NNT,NNRS,NNT)
      PIN=P1-PL0
      POUT=0.
```

```
C
      IF(PIN.GE.0.)GO TO 200
      IF(IMPL.NE.2)GO TO 200
      WRITE(6,208)PLO,P1
  206 FORMAT(1H0,21H FLYWHEEL POWER LOSS ,F12.3,
     X24H EXCEEDS CHARGING POWER ,F12.3)
      ICNT=ICNT+1
C
C                    COMPUTE POWER LOSSES AND OUTPUT POWER WHEN DISCHARGING
C
  200 IF(RE1.EQ.0.)GO TO 300
       T1=RE1*737.6/OMEGA
      PL1=TBLU2(RS,-T1,CLO(M4),CLO(4),CLO(MN4),1,1,-NNRS,-NNT,NNRS,NNT)
      P2=RE1-PL1
      POUT=RE1
      IF(P2.GT.0..OR.IMPL.NE.2)GO TO 300
       WRITE(6,306)PL1,RE1
  308 FORMAT(1H0,16H  FLYWHEEL LOSS ,F12.3,
     X27H EXCEEDS DISCHARGING POWER ,F12.3)
      ICNT=ICNT+1
      P2=0.
C
C                    COMPUTE POWER LOSSES WHEN DISENGAGED
C


  300 IF(P1.GT.0.)GO TO 400
      IF(RE1.GT.0.)GO TO 400
      POUT=TBLU1(RS,CL1(4),CL1(NNNRS4),1,-NNNRS)
C
C                    FLYWHEEL KINETIC ENERGY BALANCE
C
  400 IF(IKE.NE.0)KED=PIN-POUT-C1*OMEGA-C2*(OMEGA**2.8)
C
C                    MAXIMUM CHARGING POWER REQUEST
C
      TM=MP1*737.6/OMEGA
      MPA=TBLU2(RS,TM,CLO(M4),CLO(4),CLO(MN4),1,1,-NNRS,-NNT,
     X    NNRS,NNT)
      MPO=MP1-MPA
      IF(MPO.GT.0.)GO TO 500
C     IF(IMPL.EQ.2)WRITE(6,508)MPA,MP1
C 508 FORMAT(1H0,22H FLYWHEEL CLUTCH LOSS  ,F12.3,
C    X 31H EXCEEDS MAXIMUM INPUT POWER  ,F12.3)
C     IF(IMPL.EQ.2)ICNT=ICNT+1
      GO TO 600
  500 EFO=EF1*MPO/MP1
      APC=AMAX1(0.,.5*(E1-KE)/TINC)
      RE2=AMIN1(MPO,RAP,APC)
      RE2=RE2/EFO
C
C                    OUTPUT EFFICIENCY AND MAXIMUM POWER
C
  600 RAPT=(KE-EO)/(TINC*2.)
      RAPT=AMIN1(RAPT,RAP)
      RAPT=AMAX1(RAPT,RAP/1000.)
      TM=RAPT*737.6/OMEGA
      MPB=TBLU2(RS,-TM,CLO(M4),CLO(4),CLO(MN4),1,1,-NNRS,-NNT,
     X   NNRS,NNT)
      MP2=RAPT-MPB
      IF(MP2.GT.0..OR.IMPL.NE.2)GO TO 700
  708 FORMAT(1H0,22H FLYWHEEL CLUTCH LOSS ,F12.3,
     X27H EXCEEDS DELIVERABLE POWER  ,F12.3)
      WRITE(6,708)MPB,RAPT
      ICNT=ICNT+1
  700 MP2=AMAX1(MP2,RAP/1000.)
      EF2=MP2/RAPT
      IF(RE1.GT.0..AND.P2.GT.0.)EF2=P2/RE1
C
C                   PRIORITY INTERRUPT
C
      EC1=E1-EDE
      EC0=E0+EDE
      IF((KE.GT.EC0).AND.(INT.EQ.1))INT=0
      IF((KE.LT.EC1).AND.(INT.EQ.-1))INT=0

      IF(KE.LE.E0)INT=1.
      IF(KE.GT.E1)INT=-1.
      IF((KE.GT.EC0).AND.(KE.LT.EC1))INT=0.
      IF(IMPL.LE.1)RETURN
C
C                   STATISTICS
C
      ME=AMAX1(ME,KE)
      MPC=AMAX1(MPC,KED)
      MPD=AMAX1(MPD,-KED)
      SPC=SPC+TINC*P1
      SPD=SPD+TINC*P2
C
      IF(TIME.LT.TMAX1)RETURN
      CCI=CCI+CC
      CMI=CMI+CM
C

      RETURN
      END
```

7.10 FRESNEL LENS SOLAR COLLECTOR


Solar Insolation (ST)
Air Velocity or Wind (WD)
Ambient Temperature (TA)
Fluid Inlet Temperature (TFI)
FO
Cell Temperature (TC)
Fluid Flow Rate (FMD)
Outlet Fluid Temperature (T2)
Thermal Power Obtained (P1)

The Fresnel lens collector model performs a thermal analysis for a concentrating photovoltaic array which tracks the sun. The array may be cooled passively or by forced air or fluid. Fins may be used on the back to increase convective heat transfer to the environment. Figures 7.10-1 and 7.10-2 show the physical construction of the array and the equivalent thermal network for the focusing collector. The purpose of the model is to compute the cell temperature TC, and the fluid pump rate FMD when fluid cooling is used. The analysis is based on a similar thermal model in SOLCEL [4].

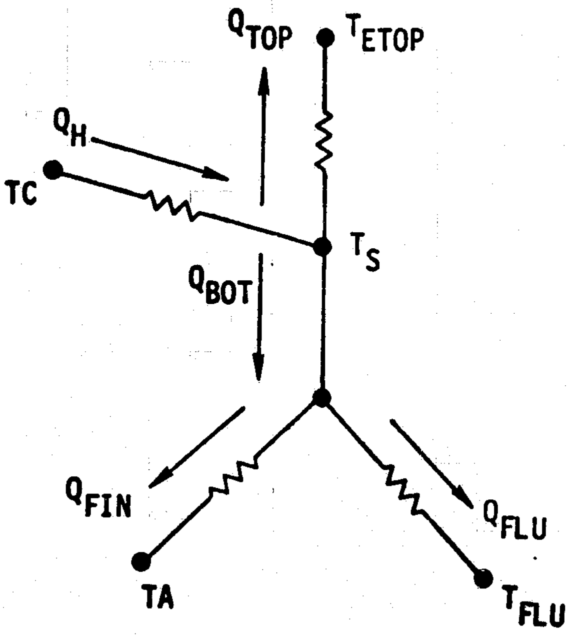


Figure 7.10-1 Equivalent Thermal Network for Fresnel Lens Collector

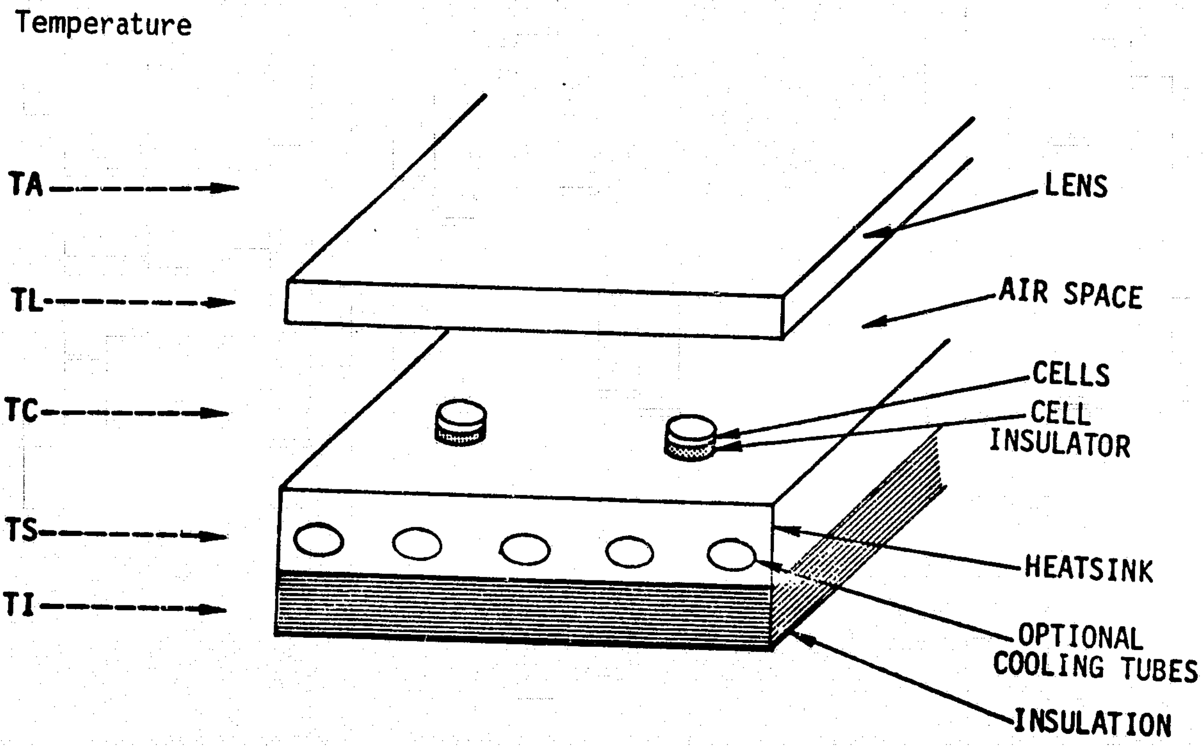


Figure 7.10-2 Fresnel Lens Thermal Model

FO

BASIC EQUATIONS

1) Energy absorbed by the collector per unit area

$$QH = ST*TAU*(ABC-EFF)$$

where

ST = direct beam solar insolation

TAU = lens transmittance

ABC = cell absorptance

EFF = nominal cell efficiency

2) Heat balance equations for the thermal network of 7.10-1:

$$Q_h = Q_{TOP} + Q_{BOT}$$

$$Q_{TOP} = H_{TOP}(TS-T_{ETOP}) = H_L(TS-TL)$$

$$Q_{BOT} = H_{BOT}(TS-T_{EBOT}) = Q_{FIN} + Q_{FLU}$$

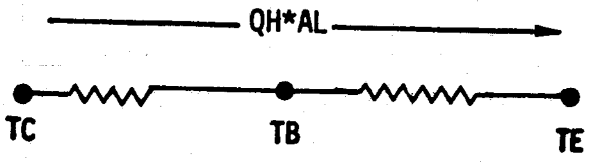
$$Q_{FIN} = H_{FIN}(TS-TA) = H_I(TS-TI)$$

$$Q_{FLU} = H_{FLU}(TS-T_{FLU})$$

3) The temperature variation in the insulating bond between the cell and the heat sink is given by a radial conduction equation for $r > a$:

$$r^2 \frac{\partial^2 T_B}{\partial r^2} + \frac{\partial T_B}{\partial r} - \alpha r^2 T_B = 0,$$

with $\frac{\partial T_B}{\partial r}$ specified at the cell radius r=a and at the equivalent lens radius r=b. This equation may be solved using modified Bessel functions to compute T_B at r=a given the overall heat transfer coefficient and equivalent temperature of the collector minus bonding. Thus the cell, bonding, and collector thermal diagram reduces to



where

AL = lens area = πb^2

$TE = (H_{TOP}*T_{ETOP}+H_{BOT}*T_{EBOT})/(H_{TOP}+H_{BOT})$

Input Specification Notes

Minimum input parameters to specify FO are

CMØ	=	Cooling mode option
TFØ	=	Outlet fluid temperature (CMØ=2)
NT	=	Number of cooling tubes (CMØ=2)
HI	=	Thermal conductivity/thickness of back insulation (CMØ=2)
AL	=	Area of lens
NL	=	Number of lenses
CL	=	Collector length
CW	=	Collector width
RC	=	Radius of solar cell
FIR	=	Cooling fin/collector area ratio (CMØ=0)

The user should check inputs for consistency with those used in the photovoltaic model PV. For example

FO collector area = $CL*CW \overset{?}{\geq} AL*NL \overset{?}{\geq}$ PV array area

FO concentration ratio = $AL/(\pi *RC^2) \overset{?}{=}$ PV concentration ratio

FO cell area = $\pi *RC^2 \overset{?}{\geq}$ PV array area/number of cells

Inputs/Port	Description	Units
ST	Direct beam solar insolation	w/m^2
WD	Air or wind velocity (default = 0.)	m/s
TA	Ambient temperature	^oC
TFI	Inlet fluid temperature	^oC
TFØ	Specified outlet fluid temperature	^oC
CMØ	Cooling mode (default = 0.) 0 = natural air cooling 1 = forced air cooling 2 = fluid cooling	-
AL	Lens area	m^2
TAU	Lens transmittance (default = 1.)	-
ABC	Cell absorptance (default = .95)	-
EFF	Nominal cell efficiency (default = .12)	-
SPA	Lens to heatsink space (default = .025)	m
EL	Emittance of lens (default = .9)	-
ES	Heatsink emittance (default = .5)	-
EI	Emittance of the back surface (default = .5)	-
CW	Collector width	m
CL	Collector length	m
NL	Number of lenses on collector	-
RC	Radius of solar cells (default = .025)	m
ABL	Absorptance of the lens (default = .05)	-
SPT	Specific heat of coolant (default = 4184)	j/kg-K
HI	Conductivity/thickness of the back insulation (default = 10^9 for no insulation)	w/m^2-K

Inputs/Port (cont'd)	Description	Units
FIR	Cooling fin to flat plate area ratio (default = 1 for no fin)	-
NT	Number of cooling tubes	-
MFM	Maximum fluid flow rate	kg/s
DT	Diameter of cooling tubes (default = .015)	m
CØS	Conductivity of heatsink (default = 202)	w/m-K
THS	Heatsink plate thickness (default = .003)	m
DEN	Coolant density (default = 980.)	kg/m^3
CØC	Conductivity of the coolant (default = .657)	w/m
HC	Conductivity/thickness of the cell insulator (default = 10^9 for no insulation)	w/m^2-K
CC	Capital cost per unit collector area per year	$\$/m^2$
CM	Maintenance cost per year	$
CØP	Cost of operating power	$/kwh

Outputs/Port		Description	Units
TC		Cell temperature	^{o}C
TS		Heatsink temperature	^{o}C
FMD		Fluid flow rate	kg/s
T	1	Inlet fluid temperature	^{o}C
T	2	Outlet fluid temperature	^{o}C
PH		Collector energy absorbed	kw
P	1	Thermal energy collected	kw

Outputs/Port (cont'd)	Description	Units
REA	Reynolds number (air cooling)	-
REF	Reynolds number (fluid cooling)	-
LTI	Last time at which the collector calculations were performed	hr
ØP	Operating Power used (state)	kwh

CALCULATION SEQUENCE

$RL = (AL/\pi)^{.5}$

1) Solar Power Absorbed by the Collector

$QH = ST*TAU(ABC-EFF)$

$PH = QH*AL*NL/1000.$

If $QH \leq 0.1$ set $TC = TA$, $FMD = P1 = \dot{Ø}P = 0$ and return

If $LTI = TIME$ and $|TFI - T1| < .1$, return

$LTI = TIME$

2) Convert TA,TFO,TFI to ^{o}K

3) Initial Temperature and Flow Rate Estimates

$TS = TA + QH/20$

$TL = (TS + TØ)*.5$

$TF = (TFI + TFO)*.5$

$TI = TL$

$FMD = IFLU = 0$

If $CMØ = 2$ and $TFØ > TFI$, $IFLU = 1$

If $IFLU = 1$,

CALCULATION SEQUENCE (cont'd)

RO = NT*SPT*(TFØ-TFI)/(AL*NL)

FMD = MIN(0.5*QH/RO,MFM)

o Iterate 4) to 8) three times:

4) HTOP Heat Transfer Coefficient and TETOP

$T_{SKY} = .0552*TA^{1.5}$

$\begin{pmatrix} HC1 \\ REA \end{pmatrix} = CNVC(TL,TA,WD,CL)$ Appendix (2)-(3)

$HR1 = RADC(TL,TSKY,EL,1.)*\frac{(TL-TSKY)}{(TL-TA)}$ Ibid,(8)

H1 = HC1 + HR1

TM = .5*(TL+TS)

$HC2 = (7.25x10^{-5}*TM+4.325x10^{-3})/SPA$

HR2 = RADC(TS,TL,ES,EL) Ibid,(8)

HL = HC2 + HR2

$HTOP = (1/H1 + 1/HL)^{-1}$

TETOP = TA+ST*(ABL+(1-TAU)*TAU*ABC)/H1

5) Fin Factor and HFIN Heat Transfer Coefficient

HC = CNVC(TI,TA,WD,CL) Ibid,(2)-(3)

HR = RADC(TI,TA,EI,1.) Ibid,(8)

FAC = 4.318 - 4.3375*EXP(-.26795*FIR) (First pass)

$HFIN = (1/HI + 1/(HC*FAC+HR))^{-1}$

6) HFLU Heat Transfer Coefficient to Fluid and REF

HFLU = 0.

CALCULATION SEQUENCE (cont'd)

If IFLU = 0 go to (7)

$\begin{pmatrix}HFLU\\REF\end{pmatrix}$ = FLUC(NT,DT,CW,CØS,THS, FMD,DEN,TF,CØC)   Ibid,(5)-(6)

7) HBOT Heat Transfer Coefficient and Temperature TEBOT

HBOT = HFIN + HFLU

TEBOT = (HFIN*TA+HFLU*TF)/HBOT

8) Temperature and Flow Rate Updates

H = HTOP + HBOT

TE = (HTOP*TETOP+HBOT*TEBOT)/H

TS = TE + QH/H

TL = TS - HTOP*(TS-TETOP)/HL

TI = TS - HFIN(TS-TA)/HI

QFLU = HFLU(TS-TF)

FMD = 0.

If QFLU > 0, FMD = QFLU/RO

If QFLU > MFM*RO,

FMD = MFM

RA = QFLU/MFM

TF = TFI+RA*AL*NL*.5/(SPT*NT)

9) Check for QFLU < 0

If QFLU < 0 set IFLU = 0 and repeat (4)-(8) once

10) Cell Temperature

ALPH = H/(CØS*THS)

X = SQRT(ALPH)*RC

CALCULATION SEQUENCE (cont'd)

$$Y = SQRT(ALPH)*RL$$

$$BETA = QH*AL/(2\pi *CØS*THS*X)$$

$$A = BETA*I1(Y)/(K1(X)*I1(Y)-K1(Y)*I1(X))$$

$$B = BETA*K1(Y)/(K1(X)*I1(Y)-K1(Y)*I1(X))$$

$$TB = A*K0(X)+B*I0(X)+TE$$

$$TC = TB+QH*AL/(\pi *RC^2*HC)$$

where I0,I1,K0,K1 are modified Bessel functions.

11) Output Calculation

$$T2 = 2*TF-TFI$$

Convert TC,TS,T1,T2,TA,TFI,TFØ to ^oC

$$P1 = QFLU*AL*NL/1000.$$

$$TKP = 5.E-4*CL*CW$$

$$\dot{Ø}P=TKP+\begin{cases} 0. & \text{if } CMØ = 0 \\ .0742*(CW*CL)^{.2835}*WD^{.567} & \text{if } CMØ = 1 \text{ and } WD>0 \\ 7.85x10^{-11}*FMD^{2.855}*DT^{(-4.702)}*NT*CL & \text{if } CMØ = 2 \text{ and } FMD>0 \end{cases}$$

REFERENCES FOR FO

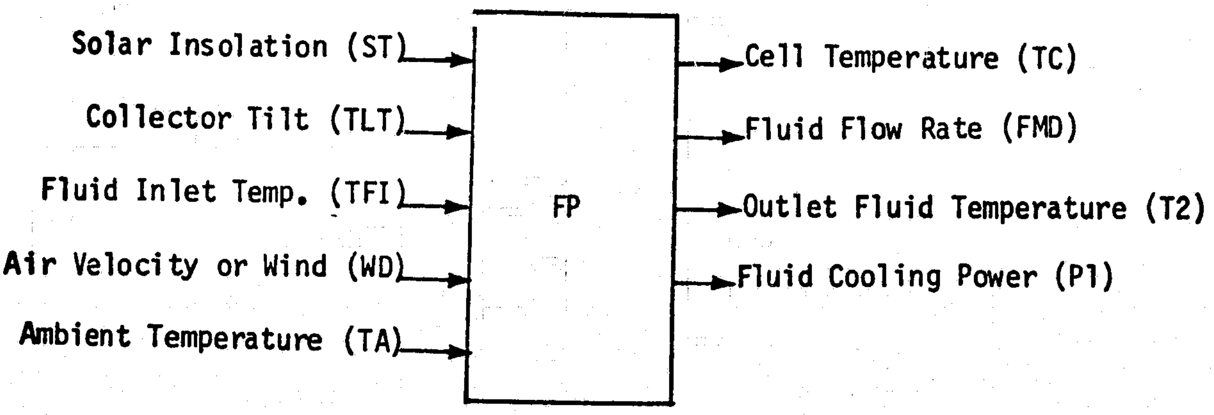
1. J. K. Linn, "Photovoltaic System Analysis Program-SOLCEL," Sandia Laboratories Report SAND77-1268, 1977.

2. E. L. Burgess and M. W. Edenburn, "One Kilowatt Photovoltaic Subsystem Using Fresnel Lens Concentrators," Paper 11.6, IEEE Photovoltaic Specialists Conference, Baton Rouge, November 1976.

```
CFC
      SUBROUTINE FO(TC,TS,FMD,T1,T2,PH,P1,REA,REF,LTI,OP,OPD,IOP,
     1 ST,WD,TA,TFI,
     1 TFO,CMO,AL,TAU,ABC,EFF,SPA,EL,ES,EI,CW,CL,NL,
     2 RC,ABL,SPT,HI,FIR,NT,MFM,DT,COS,THS,DEN,COC,
     3 HC,CC,CM,COP)
C
C       PURPOSE     THIS COMPONENT COMPUTES THE TEPERATURE OF
C                   THE SOLAR CELL IN THE FRESNEL LENS COLLECTOR,
C                   AND CALCULATES THE FLUID PUMP RATE WHEN FLUID
C                   COOLING IS USED
C
C     WRITTEN BY Y.K.CHAN, 10-19-76, VERSION 1
C
C       METHOD     THERMAL ANALYSIS BASED ON SOLCEL MODEL OF SANDIA
C
C       CALL SEQUENCE
C            OUTPUTS
C                TC    -CELL TEMPERATURE,C
C                TS    -HEATSINK TEMPERATURE,C
C                FMD   -FLUID FLOW RATE, KG/S
C                T1    -INLET FLUID TEMPERATURE,C
C                T2    -OUTLET FLUID TEMPERATURE,C
C                PH    -COLLECTOR ENERGY ABSORBED,KW
C                P1    -THERMAL POWER COLLECTED,KW
C                REA   -REYNOLDS NUMBER(AIR COOLING)
C                REF   -REYNOLDS NUMBER(FLUID COOLING)
C                LTI   -LAST TIME AT WHICH THE COLLECTOR
C                       CALCULATION WAS PERFORMED,HR
C                OP    -ACCUMULATIVE OPERATING ENERGY,(STATE),KWH
C                OPD   -OPERATING POWER,KW
C                IOP   -INTEGRATOR CONTROL
C            INPUTS
C                ST    -GLOBAL SOLAR INSOLATION,W/M2
C                WD    -AIR OR WIND VELOCITY,M/S,(DEFAULT=0)
C                TA    -AMBIENT TEMPERATURE,C
C                TFI   -SPECIFIED INLET FLUID TEMPERATURE,C
C                TFO   -SPECIFIED OUTLET FLUID TEMPERATURE,C
C                CMO   -COOLING MODE(DEFAULT=0)
C                       0=NATURAL AIR COOLING
C                       1=FORCED AIR COOLING
C                       2=FLUID COOLING
C                AL    -LENS AREA,M2,(DEFAULT=.09)
C                TAU   -LENS TRANSMITTANCE(DEFAULT=1.)
C                ABC   -CELL ABSORPTANCE(DEFAULT=.95)
C                EFF   -NOMIANAL CELL EFFICIENCY(DEFAULT=.12)
C                SPA   -LENS TO HEATSINK SPACE,M,(DEFAULT=.025)
C                EL    -EMITTANCE OF LENS(DEFAULT=.9)
C                ES    -HEATSINK EMITTANCE(DEFAULT=.5)
C                EI    -BACK SURFACE EMITTANCE(DEFAULT=.5)
C                CW    -COLLECTOR WIDTH,M
C                CL    -COLLECTOR LENGTH,M
C                NL    -NUMBER OF LENSES ON COLLECTOR
C                RC    -RADIUS OF SOLAR CELLS,M,(DEFAULT=.025)
C                ABL   -ABSORPTANCE OF THE LENS(DEFAULT=.05)
C                SPT   -SPECIFIC HEAT OF COOLANT,J/KG-K,(DEFAULT=4184)
C                HI    -CONDUCTIVITY/THICKNESS OF THE BACK INSULATION,
C                       W/M2-K,(DEFAULT=10**9 FOR NO INSULATION)
```

```
C                 FIR   -COOLING FIN TO FLAT PLATE AREA RATIO
C                        (DEFAULT=1 FOR NO FIN)
C                 NT    -NUMBER OF COOLING TUBES
C                 MFM   -MAXIMUM FLUID FLOW RATE,KG/S
C                 DT    -DIAMETER OF COOLING TUBES,M,(DEFAULT=.015)
C                 COS   -CONDUCTIVITY OF HEAT SINK,W/M-K,(DEFAULT=202)
C                 THS   -HEATSINK PLATE THICKNESS,M,(DEFAULT=.003)
C                 DEN   -COOLANT DENSITY,KG/M3,(DEFAULT=980)
C                 COC   -CONDUCTIVITY OF THE COOLANT,W/M-K,(DEFAULT=.657)
C                 HC    -CONDUCTIVITY/THICKNESS OF THE CELL INSULATOR,
C                        W/M2-K,(DEFAULT=10**9 FOR NO INSULATION)
C                 CC    -CAPITAL COST PER UNIT COLLECTOR AREA PER YEAR,$/M2
C                 CM    -MAINTENANCE COST PER YEAR
C                 COP   -COST OF OPERATING POWER,$/KWH
C
      COMMON /CIMPL/IMPL,ICNT,ITEST
      COMMON /CTIME/TIME /CSIMUL/DUM(7),TMAX
      COMMON /COST/CCAP,CMA,CPO
      REAL NL,NT,MFM,LTI
C
C                 INITIALIZATION
C
      IF(IMPL.GT.0)GO TO 100
      IF(WD.EQ..99999)WD=0.
      IF(CMO.EQ..99999)CMO=0.
      IF(AL.EQ..99999)AL=.09
      IF(TAU.EQ..99999)TAU=1.
      IF(ABC.EQ..99999)ABC=.95
      IF(EFF.EQ..99999)EFF=.12
      IF(SPA.EQ..99999)SPA=.025
      IF(EL.EQ..99999)EL=.9
      IF(ES.EQ..99999)ES=.5
      IF(EI.EQ..99999)EI=.5
      IF(RC.EQ..99999)RC=.025
      IF(ABL.EQ..99999)ABL=.05
      IF(SPT.EQ..99999)SPT=4184
      IF(HI.EQ..99999)HI=1.E9
      IF(FIR.EQ..99999)FIR=1.
      IF(DT.EQ..99999)DT=.015
      IF(COS.EQ..99999)COS=202
      IF(THS.EQ..99999)THS=.003
      IF(DEN.EQ..99999)DEN=980
      IF(COC.EQ..99999)COC=.657
      IF(HC.EQ..99999)HC=1.E9
      RL=SQRT(AL/3.1415926)
      FAC=4.318-4.3375*EXP(-.26795*FIR)
      TMAX1=TMAX*.99999
C
  100 CONTINUE
C
C          SOLAR POWER ABSORBED BY THE COLLECTOR
C
      QH=ST*TAU*(ABC-EFF)
      PH=QH*AL*NL/1000.
      IF(QH.GT..01 )GO TO 201
      TS=TA
      TC=TA
      OPD=0.
      FMD=0.
      P1=0.
      GO TO 920
  201 IF((LTI.EQ.TIME).AND.(ABS(TFI-T1).LT..1))GO TO 920
      LTI=TIME
C
C                CONVERT TA,TFO,TFI  FROM CENTIGRADE TO KELVIN
C
      TA=TA+273
      TFO=TFO+273
      TFI=TFI+273
C
C               INITIAL TEMPERATURE AND FLOW RATE ESTIMATES
C
      TS=TA+QH/20.
      TL=(TS+TA)*.5
      TF=(TFI+TFO)*.5
      TI=TL
      IFLU=0.
      FMD=0.
      IF((ABS(CMO-2.).LT..1).AND.(TFO.GT.TFI))IFLU=1
      IF(IFLU.NE.1)GO TO 301
      RO=NT*SPT*(TFO-TFI)/(AL*NL)
      FMD=MFM
      IF(RO.GT.0.)FMD=AMIN1(.5*QH/RO,MFM)
  301 CONTINUE
C
C               ITERATE HEAT COEFFICIENT CALCULATION THREE TIMES
C
      LOOP=0
  400 CONTINUE
C
C    HTOP, HEAT TRANSFER COEFFICIENT, AND TETOP,TOP EQUIVALENT TEMPERATURE
      TSKY=.0552*(TA**1.5)
      CALL CNVC(HC1,REA,TL,TA,WD,CL)
      CALL RADC(HR1,TL,TSKY,EL,1.)
      HR1=HR1*(TL-TSKY)/(TL-TA)
      H1=HC1+HR1
      TM=.5*(TL+TS)
      HC2=(7.25*1.E-5*TM+4.325E-3)/SPA
      CALL RADC(HR2,TS,TL,ES,EL)
      HL=HC2+HR2
      HTOP=1./(1./H1+1./HL)
      TETOP=TA+ST*(ABL+(1-TAU)*TAU*ABC)/H1
C
C               HEAT TRANSFER COEFFICIENT HFIN
      CALL CNVC(HC2,RE,TI,TA,WD,CL)
      CALL RADC(HR,TI,TA,EI,1.)
      HFIN=1./(1./HI+1./(HC2*FAC+HR))
C
C     FLUID HEAT TRANSFER COEFFICIENT HFLU AND REYNOLDS NUMBER REF
      HFLU=0.
      IF(IFLU.EQ.0)GO TO 700
      CALL FLUC(HFLU,REF,NT,DT,CW,COS,THS,FMD,DEN,TF,COC)
C
C     EQUIVALENT BOTTOM TEMPERATURE AND HEAT TRANSFER COEFFICIENT
  700 CONTINUE
      HBOT=HFIN+HFLU
      TEBOT=(HFIN*TA+HFLU*TF)/HBOT
C
C          UPDATE TEMPERATURE AND FLOW RATE
      H=HTOP+HBOT
      TE=(HTOP*TETOP+HBOT*TEBOT)/H
      TS=TE+QH/H
      TL=TS-HTOP*(TS-TETOP)/HL
      TI=TS-HFIN*(TS-TA)/HI
      QFLU=HFLU*(TS-TF)
C     WRITE(6,108)HFIN,HBOT,TEBOT,HTOP,TETOP,H,TE,TS,TL,TI,QFLU,RO
C 108 FORMAT(1H ,*FO  *,6E10.2,/,5X,6E10.2)
      FMD=0.
      IF(QFLU.LE.0.)GO TO 800
      IF(QFLU.GT.(MFM*RO))GO TO 799
      FMD=QFLU/RO
      GO TO 800
  799 FMD=MFM
      RA=QFLU/MFM
      TF=TFI+RA*AL*NL*.5/(SPT*NT)
  800 CONTINUE
C
      LOOP=LOOP+1
      IF(LOOP.LE.2)GO TO 400
C
C      CHECK FOR EFFECTIVE FLUID COOLING
C
      IF(QFLU.GE.0.)GO TO 900
      IFLU=0.
      GO TO 400
  900 CONTINUE
C
C                CELL TEMPERATURE
C
      ALPH=H/(COS*THS)
      X=SQRT(ALPH)*RC
      Y=SQRT(ALPH)*RL
      BETA=QH*AL/(2.*3.14159*COS*THS*X)
      CALL NATSI1(Y,BI1Y,1)
      CALL NATSK1(X,BK1X,1)
      CALL NATSI1(X,BI1X,1)
      CALL NATSK1(Y,BK1Y,1)
      CALL NATSK0(X,BK0X,1)
      CALL NATSI0(X,BI0X,1)
      A=BETA*BI1Y/(BK1X*BI1Y-BK1Y*BI1X)
      B=BETA*BK1Y/(BK1X*BI1Y-BK1Y*BI1X)
      TB=A*BK0X+B*BI0X+TE
      TC=TB+QH*AL/(3.14159*RC*RC*HC)
C
C      OUTPUT CALCULATION
C
      TC=TC-273
      TS=TS-273
      T1=TFI-273
      T2=2.*TF-TFI-273
      TA=TA-273
      TFI=TFI-273
      TFO=TFO-273
      P1=QFLU*AL*NL/1000.
      RE1=0.
      IF(ABS(CMO-1.).LE..1 .AND. WD.GT.0.)RE1=.0742*((CW*CL)**.2835)*
     1  WD**.567
      IF(FMD.LE.0.)GO TO 909
      IF(CMO.GT.1.1)RE1=7.85E-11*(FMD**2.855)*(DT**(-4.702))*NT*CL
  909 CONTINUE
      TKP=5.E-4*CL*CW
      IF(IOP.NE.0)OPD=TKP+RE1
  920 IF(TIME.LT.TMAX1)RETURN
      IF(IMPL.LT.2)RETURN
      CCAP=CCAP+CC*AL*NL
      CMA=CMA+CM
      CPO=CPO+COP*OP
      RETURN
      END
```

7.11 FLAT PLATE SOLAR COLLECTOR



The flat plate component performs a thermal analysis on a nonconcentrating photovoltaic array. Three types of cooling may be used:

- Front surface cooling using natural or forced air.
- Back surface cooling using natural or forced air with or without a finned back surface.
- Fluid cooling using tubes on the back and N glass covers (N = 0,1,2,3).

Figures 7.11-1 and 7.11-2 show the physical construction of the array and the equivalent thermal network for the flat plate component. The purpose of the analysis is to compute the cell temperature TC and the fluid pump rate FMD when fluid cooling is used. The analysis is based on the flat plate thermal model in SOLCEL [4], except that an empirical equation due to Klein is used to compute the top loss coefficient for 1 to 3 glass covers.

TEMPERATURES

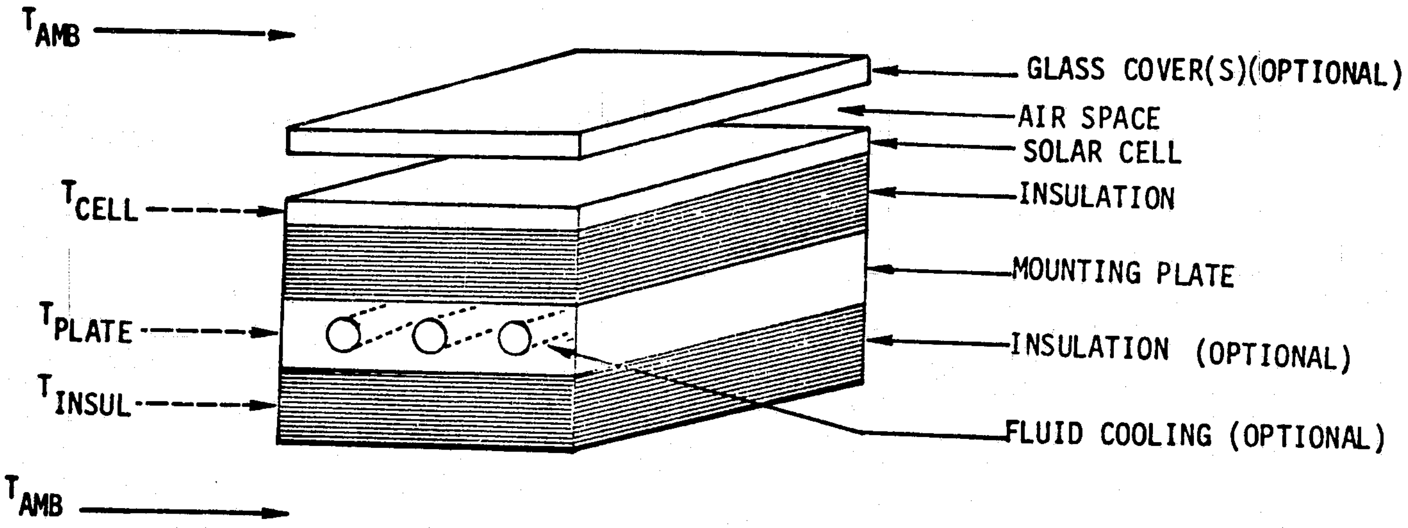


Figure 7.11-1 Physical Diagram of Flat Plate Collector

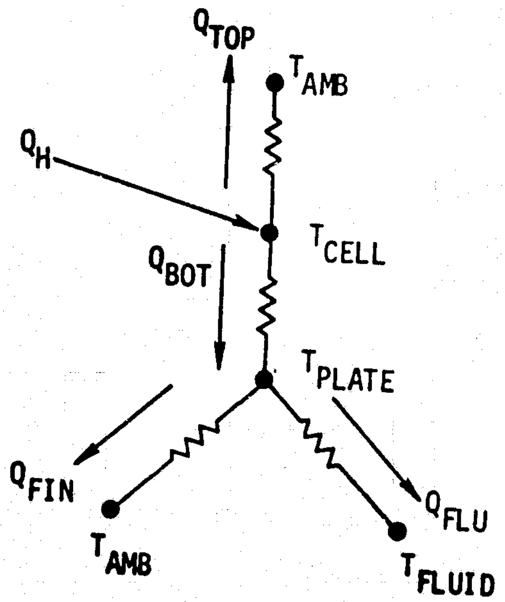


Figure 7.11-2 Equivalent Thermal Network for Flat Plate Collector

BASIC EQUATIONS

The basic thermal equations for the model are the heat balance equations for the network of Figure 7.11-2.

$$Q_H = ST{*}T_N(AB - EFF) = Q_{TOP} + Q_{BOT}$$

$$Q_{TOP} = H_{TOP}(T_{CELL} - T_{AMB})$$

$$Q_{BOT} = H_{BOT}(T_{CELL} - T_{EBOT}) = H_C\ (T_{CELL} - T_{PLATE})$$

$$= Q_{FIN} + Q_{FLU}$$

$$Q_{FIN} = H_{FIN}(T_{PLATE} - T_{AMB}) = H_I(T_{PLATE} - T_{INSUL})$$

$$Q_{FLU} = FMD{*}P(\text{TFØ} - TFI) = H_{FLU}(T_{PLATE} - T_{FLUID})$$

where H_{TOP}, H_{BOT}, H_C... denote heat transfer coefficients, and

T_N = transmittance of the N-covers

AB = collector cell absorptance

EFF = nominal cell efficiency

T_{EBOT} = equivalent bottom temp. (= T_{AMB} with no fluid cooling)

P = fluid specific heat/unit cell area * No. of cooling tubes

T_{FLUID} = average fluid temperature = (TFØ + TFI)/2.

Input Specification Notes

Minimum input parameters to specify FO are

CMØ = Cooling mode option
TFØ = Outlet fluid temperature (CMØ=2)
NG = Number of glass covers
HI = Conductivity/thickness of the back insulation
CW = Collector width
CL = Collector length
NT = Number of cooling tubes (CMØ=2)
FIR = Cooling fin/collector area ratio (CMØ=0)

The user should check the consistency of these inputs (e.g., collector area) with those of the tracking component SO and the photovoltaic component PV.

Inputs/Port	Description	Units
ST	Global solar insolation	w/m^2
TLT	Collector tilt	Deg
WD	Air or wind velocity (default = 0.)	m/s
TA	Ambient drybulb temperature	^{o}C
TFI	Inlet fluid temperature	^{o}C
TFØ	Specified outlet fluid temperature	^{o}C
MFM	Maximum fluid flow rate	kg/s
RE	Tracking power request	kw
CMØ	Cooling mode (default = 0.) 0 = natural air cooling 1 = forced air cooling 2 = fluid cooling	-
NG	Number of glass covers (default = 0.)	-
TN	Transmittance of the N-covers	-
AB	Collector cell absorptance (default = .9)	-
EFF	Nominal cell efficiency (default = .12)	-
EC	Emittance of cell (default = 0.5)	-
EG	Emittance of the glass covers (default = .9)	-
EP	Emittance of the back surface (default = .9)	-
CW	Collector width	m
CL	Collector length	m
SPT	Specific heat of coolant (default = 4184.)	j/kg-K
HI	Conductivity/thickness of the back insulation (default = 10^9 for no insulation)	w/m^{2}K

Inputs/Port (cont'd)	Description	Units
FIR	Cooling fin to flat plate area ratio (default = 1. for no fin)	-
NT	Number of cooling tubes (default = 1)	-
DT	Diameter of cooling tubes (default = .015)	m
CØP	Conductivity of mounting plate (default = 202.)	w/m-K
THP	Mounting plate thickness (default = .003)	m
DEN	Coolant density (default = 980.)	kg/m^3
CØC	Conductivity of the coolant (default = .657)	w/m-K
HC	Conductivity/thickness for cell insulation (default = 10^9 for no insulation)	w/m^2-K
CC	Capital cost per unit area per year	$\$/m^2$
CM	Maintenance cost per year	$
CPØ	Cost of operating power	$/kwh

Outputs/Port	Description	Units
TC	Cell temperature	°C
TP	Mounting plate temperature	°C
FMD	Fluid flow rate	kg/s
T1	Inlet fluid temperature	°C
T2	Outlet fluid temperature	°C
PH	Collector energy absorbed	kw
P1	Thermal energy collected	kw
ØP	Operating power used (state)	kwh
REA	Reynolds number (air cooling)	-
REF	Reynolds number (fluid cooling)	-
LTI	Last time at which the flat plate array calculations were performed (used internally)	hr

CALCULATION SEQUENCE

1) Solar power absorbed by the collector

$QH = ST*TN*(AB - EFF)$

$PH = QH*CL*CW/1000$

If $QH \leq 0.1$ set $TC = TA$, $FMD = P1 = \dot{O}P = 0$ and return

If $LTI = TIME$ and $|TFI - T1| < .1$, return

$LTI = TIME$

2) Convert TA, TFØ, TFI to ^{o}K

3) Initial temperature and flow rate estimates

$TC = TA + QH/20$

$TI = (TC + TA)*.5$

$TF = (TFI+TF\O)*.5$

$TP = TI$

$FMD = 0$

$IFLU = 0$

If $CM\O = 2$ and $TF\O > TFI$, $IFLU = 1$

If $IFLU = 1$,

$RO = NT*SPT*(TF\O - TFI)/CW*CL$

$FMD = MIN(MFM,0.8*QH/RO)$

- Iterate 4) to 8) three times:

4) HTOP heat transfer coefficient and REA

$TSKY = .0552*TA^{1.5}$

$\begin{pmatrix} HC1 \\ REA \end{pmatrix} = CNVC(TC, TA, WD, CL)$ See (2)-(3) in Appendix

If $NG = 0$,

CALCULATIONS (cont'd)

$HR1 = RADC(TC,TSKY,EC,1.)*\frac{(TC-TSKY)}{TC-TA)}$ Ibid,(8)

$HTOP = HC1 + HR1$

If $NG > 0$,

$HTOP = HTGLAS(N,TA,TC,HC1,EC,EG,TLT)$ Ibid,(7)

5) Fin factor FAC and HFIN heat transfer coefficient

$HC2 = CNVC(TI, TA, WD, CL)$ Ibid,(3)

$HR2 = RADC(TI, TA, EP, 1.)$ Ibid,(8)

$FAC = 4.318 - 4.3375*exp(- .26795*FIR)$ (first pass)

$HFIN = (1/HI + 1/(HC2*FAC + HR2))^{-1}$

6) HFLU heat transfer coefficient to fluid and REF

$HFLU = 0.$

If IFLU = 0 go to 7)

$\begin{pmatrix} HFLU \\ REF \end{pmatrix} = FLUC(NT,DT,DW,CØP,THP,FMD,DEN,TF,CØC)$ Ibid,(5)-(6)

7) HBOT heat transfer coefficient and equivalent temperature TEBOT

$HBOT = (1/HC + 1/(HFIN + HFLU))^{-1}$

$TEBOT = (HFIN*TA + HFLU*TF)/(HFIN + HFLU)$

8) Temperature and flow rate updates

$TC = (QH + HTOP*TA + HBOT*TEBOT)/(HTOP + HBOT)$

$TP = TC - HBOT*(TC-TEBOT)/HC$

$TI = TP - HFIN*(TP - TA)/HI$

$QFLU = HFLU*(TP - TF)$

CALCULATIONS (cont'd)

$$FMD = \begin{cases} 0. & \text{if } QFLU \leq 0. \\ QFLU/RO & \text{if } QFLU > 0. \end{cases}$$

If $QFLU > MFM*RO$,

$FMD = MFM$

$RA = QFLU/MFM$

$TF = TFI + RA*CL*CW*.5/(SPT*NT)$

9) Check for $QFLU < 0$

If $QFLU < 0$ set $IFLU = 0$ and repeat 4) to 8) once

10) Output calculations

$T2 = 2*TF - TFI$

Convert TC,TP,T1,T2,TA,TFI,TFØ to °C

$P1 = QFLU*CL*CW/1000$

If $CMØ = 0$

$\dot{Ø}P = RE$

If $CMØ = 1$ and $WD>0$

$\dot{Ø}P = RE+.0742*(CW*CL)^{.2835}*WD^{.567}$

If $CMØ = 2$ and $FMD>0$

$\dot{Ø}P = RE + 7.85 \times 10^{-11}*FMD^{2.855}*DT^{(-4.702)}*NT*CL$

REFERENCES FOR FP

1. S. A. Klein, M.S. Thesis, "The Effects of Thermal Capacitance Upon the Performance of Flat Plate Solar Collectors," University of Wisconsin, 1973.

2. J. A. Duffie and W. A. Beckman, Solar Energy Thermal Processes. Wiley, 1974.

3. F. Kreith, Principles of Heat Transfer, 3rd Edition, International Textbook Company, 1973.

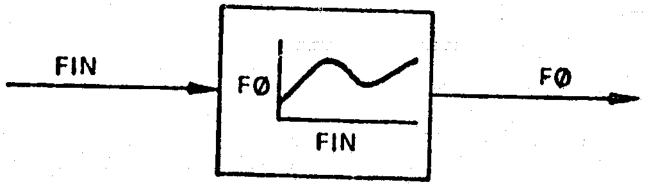
```
CFP
      SUBROUTINE FP(TC,TP,FMD,T1,T2,PH,P1,OP,OPD,IOP,REA,REF,LTI,
     1 ST,TLT,WD,TA,TFI,TFO,MFM,RE,CMO,NG,TN,AB,
     2 EFF,EC,EG,EP,CW,CL,SPT,HI,FIR,NT,DT,COP,THP,DEN,
     3 COC,HC,CC,CM,CPO)
C
C        PURPOSE     THIS COMPONENT PERFORMS A THERMAL ANALYSIS
C                    ON A NONCONCENTRATING PHOTOVOLTAIC ARRAY.
C                    THREE TYPES OF COOLING MAY BE USED
C                        FRONT SURFACE COOLING USING NATURAL OR FORCED AIR
C                        BACK SURFACE COOLING USING NATURAL OR FORCED AIR
C                             WITH OR WITHOUT FINS.
C                        FLUID COOLING USING TUBES ON THE BACK AND NG
C                             GLASS COVERS (NG=0,1,2,3).
C
C        WRITTEN BY Y.K.CHAN, 11-6-78, VERSION 1
C
C
C        METHOD     BASED ON THE FLAT PLATE THERMAL MODEL IN SOLCEL,
C                      EXCEPT THAT AN EMPIRICAL EQUATION DUE TO KLEIN IS USED
C                      TO COMPUTE THE TOP LOSS COEFFICIENT FOR 1 TO 3
C                      GLASS COVERS
C        CALLING SEQUENCE
C         OUTPUTS
C           ST    -GLOBAL SOLAR INSOLATION,W/M2
C           TC    -CELL TEMPERATURE,C
C           FMD   -FLUID FLOW RATE,KG/S
C           T1    -INLET FLUID TEMPERATURE,C
C           T2    -OUTLET FLUID TEMPERATURE,C
C           PH    -COLLECTOR ENERGY ABSORBED,KW
C           P1    -THERMAL ENERGY COLLECTED,KW
C           OP    -OPERATING POWER USED(STATE),KWH
C           REA   -REYNOLDS NUMBER(AIR COOLING)
C           REF   -REYNOLDS NUMBER(FLUID COOLING)
C           LTI   -LAST TIME AT WHICH THE FLAT PLATE ARRAY
C                  CALCULATIONS WERE PERFORMED(USED INTERNALLY)
C         INPUTS
C           TLT   -COLLECTOR TILT,DEGREES
C           WD    -AIR OR WIND VELOCITY,M/S,(DEFAULT=0.)
C           TA    -AMBIENT DRYBULB TEMPERATURE,C
C           TFI   -SPECIFIED INLET FLUID TEMPERATURE,C
C           TFO   -SPECIFIED OUTLET FLUID TEMPERATURE,C
C           MFM   -MAXIMUM FLUID FLOW RATE,KG/S
C           RE    -TRACKING POWER REQUEST,KW
C           CMO   -COOLING MODE(DEFAULT=0)
C                    0=NATURAL AIR COOLING
C                    1=FORCED AIR COOLING
C                    2=FLUID COOLING
C           NG    -NUMBER OF GLASS COVERS(DEFAULT=0)
C           TN    -TRANSMITTANCE OF THE NG GLASS COVERS
C           AB    -COLLECTOR CELL ABSORPTANCE(DEFAULT=.9)
C           EFF   -NOMINAL CELL EFFICIENCY(DEFAULT=.12)
C           EC    -EMITTANCE OF CELL(DEFAULT=.5)
C           EG    -EMITTANCE OF GLASS COVERS(DEFAULT=.9)
C           EP    -EMITTANCE OF THE BACK SURFACE(DEFAULT=.9)
C           CW    -COLLECTOR WIDTH,M
C           CL    -COLLECTOR LENGTH,M
C           SPT   -SPECIFIC HEAT OF COOLANT,J/KG-K,(DEFAULT=4184)
```

```
C          HI    -CONDUCTIVITY/THICKNESS OF THE BACK INSULATION,W/M2-K,
C                 (DEFAULT=1.E9 FOR NO INSULATION)
C          FIR   -COOLING FIN TO FLAT PLATE AREA RATIO(DEFAULT=1. FOR NO FIN
C          NT    -NUMBER OF COOLING TUBES(DEFAULT=1)
C          DT    -DIAMETER OF COOLING TUBES,M,(DEFAULT=.015)
C          COP   -CONDUCTIVITY OF MOUNTING PLATE,W/M-K,(DEFAULT=202)
C          THP   -MOUNTING PLATE THICKNESS,M,(DEFAULT=.003)
C          DEN   -COOLANT DENSITY,KG/M3,(DEFAULT=980)
C          COC   -CONDUCTIVITY OF COOLANT,W/M-K,(DEFAULT=.657
C          HC    -CONDUCTIVITY/THICKNESS FOR CELL INSULATION,W/M2-K,
C                 (DEFAULT=1.E9 FOR NO INSULATION)
C          CC    -CAPITAL COST PER UNIT AREA PER YEAR,$/M2
C          CM    -MAINTENANCE COST PER YEAR,$
C          CPO   -COST OF OPERATING POWER,$/KWH
C
      COMMON /CIMPL/IMPL
      COMMON /CTIME/TIME /CSIMUL/DUM(7),TMAX
      COMMON /COST/CCAP,CMA,CPOS
      REAL LTI,MFM,NG,NT
C
C           INITIALIZATION
C
      IF(IMPL.GT.0)GO TO 100
      IF(WD.EQ..99999)WD=0.
      IF(CMO.EQ..99999)CMO=0.
      IF(NG.EQ..99999)NG=0.
      IF(AB.EQ..99999)AB=.9
      IF(EFF.EQ..99999)EFF=.12
      IF(EC.EQ..99999)EC=.5
      IF(EG.EQ..99999)EG=.9
      IF(EP.EQ..99999)EP=.9
      IF(SPT.EQ..99999)SPT=4184
      IF(HI.EQ..99999)HI=1.E9
      IF(FIR.EQ..99999)FIR=1.
      IF(NT.EQ..99999)NT=1
      IF(DT.EQ..99999)DT=.015
      IF(COP.EQ..99999)COP=202.
       IF(THP.EQ..99999)THP=.003
      IF(DEN.EQ..99999)DEN=980
      IF(COC.EQ..99999)COC=.657
      IF(HC.EQ..99999)HC=1.E9
      TMAX1=TMAX*.99999
      FAC=4.318-4.3375*EXP(-.26795*FIR)
  100 CONTINUE
C
C        SOLAR POWER ABSORBED BY COLLECTOR
C
      QH=ST*TN*(AB-EFF)
      PH=QH*CL*CW/1000.
      IF(QH.GT.0.01)GO TO 201
      TP=TA
      OPD=0.
      TC=TA
      FMD=0.
      P1=0.
      GO TO 920
  201 IF((LTI.EQ.TIME).AND.(ABS(TFI-T1).LT..1))GO TO 920
      LTI=TIME
```

```
C
C          CONVERT TA,TFO,TFI TO KELVIN
C
      TA=TA+273
      TFO=TFO+273
      TFI=TFI+273
C
C          INITIAL TEMPERATURE AND FLOW RATE ESTIMATES
C
      TC=TA+QH/20.
      TI=(TC+TA)*.5
      TF=(TFI+TFO)*.5
      TP=TI
      FMD=0.
      IFLU=0
      IF((ABS(CMO-2.).LT..1).AND.(TFO.GT.TFI))IFLU=1
      IF(IFLU.NE.1)GO TO 301
      RO=NT*SPT*(TFO-TFI)/(CW*CL)
      FMD=MFM
      IF(RO.GT.0.)FMD=AMIN1(MFM,.8*QH/RO)
  301 CONTINUE
C
C          ITERATE HEAT TRANSFER COEFFICIENT CALCULATION THREE TIMES
C
      LOOP=0
  400 CONTINUE
C
C          HTOP, HEAT TRANSFER COEFFICIENT AND REA, REYNOLDS NUMBER
C
C
      TSKY=.0552*(TA**1.5)
      CALL CNVC(HC1,REA,TC,TA,WD,CL)
      IF(NG.GT.0.)GO TO 401
      CALL RADC(HR1,TC,TSKY,EC,1.)
      HR1=HR1*(TC-TSKY)/(TC-TA)
      HTOP=HC1+HR1
      GO TO 402
  401 HTOP=HTGLAS(NG,TA,TC,HC1,EC,EG,TLT)
  402 CONTINUE
C
C          HFIN  HEAT TRANSFER COEFFICIENT
C
      CALL CNVC(HC2,REN1,TI,TA,WD,CL)
      CALL RADC(HR2,TI,TA,EP,1.)
      HFIN=1./(1./HI+1./(HC2*FAC+HR2))
C
C     HFLU, HEAT TRANSFER COEFFICIENT TO FLUID AND REF,REYNOLDS NUMBER
C
      HFLU=0.
      IF(IFLU.EQ.0)GO TO 700
      CALL FLUC(HFLU,REF,NT,DT,CW,COP,THP,FMD,DEN,TF,COC)
C
C      EQUIVALENT BOTTOM TEMPERATURE TEBOT AND HEAT TRANSFER COEFFICIENT HBOT
C
  700 CONTINUE
      HBOT=1./(1./HC+1./(HFIN+HFLU))
      TEBOT=(HFIN*TA+HFLU*TF)/(HFIN+HFLU)
C
```

```
C     UPDATE TEMPERATURE AND FLOW RATE
C
      TC=(QH+HTOP*TA+HBOT*TEBOT)/(HTOP+HBOT)
      TP=TC-HBOT*(TC-TEBOT)/HC
      TI=TP-HFIN*(TP-TA)/HI
      QFLU=HFLU*(TP-TF)
      FMD=0.
      IF(QFLU.LE.0.)GO TO 800
      IF(QFLU.GT.(MFM*RO))GO TO 799
      FMD=QFLU/RO
      GO TO 800
  799 FMD=MFM
      RA=QFLU/MFM
      TF=TFI+RA*CL*CW*.5/(SPT*NT)
  800 CONTINUE
C
      LOOP=LOOP+1
      IF(LOOP.LE.2)GO TO 400
C
C       CHECK FOR EFFECTIVE FLUID COOLING
C
      IF(QFLU.GE.0.)GO TO 900
      IFLU=0
      GO TO 400
  900 CONTINUE
C
C       OUTPUT CALCULATION
C
      TC=TC-273.
      TP=TP-273.
      T1=TFI-273.
      T2=2.*TF-TFI-273.
      TA=TA-273.
      TFI=TFI-273.
      TFO=TFO-273.
      P1=QFLU*CL*CW/1000.
      RE1=0.
      IF(ABS(CMO-1.).LE..1 .AND. WD.GT.0.)RE1=.0742*((CW*CL)**.2835)*
     1  WD**.567
      IF(FMD.LE.0.)GO TO 909
      IF(ABS(CMO-2.).LE..1)RE1=7.85E-11*(FMD**2.855)*(DT**(-4.702))
     1   *NT*CL
  909 CONTINUE
      IF(IOP.NE.0)OPD=RE+RE1
  920 IF(TIME.LT.TMAX1)RETURN
      IF(IMPL.LT.2)RETURN
      CCAP=CCAP+CC*CL*CW
      CMA=CMA+CM
      CPOS=CPOS+CPO*OP
      RETURN
      END
```

7.12 ONE DIMENSION TABLE LOOKUP



Tables	Description
FTA	Tabular values of function

Inputs

Parameter/Port	
FIN	Input quantity
AN	ABS(AN) ≤ 0.5 for equispaced interpolation (AN < 0 prevents extrapolation)

Outputs

Variable/Port	
FØ	Output quantity

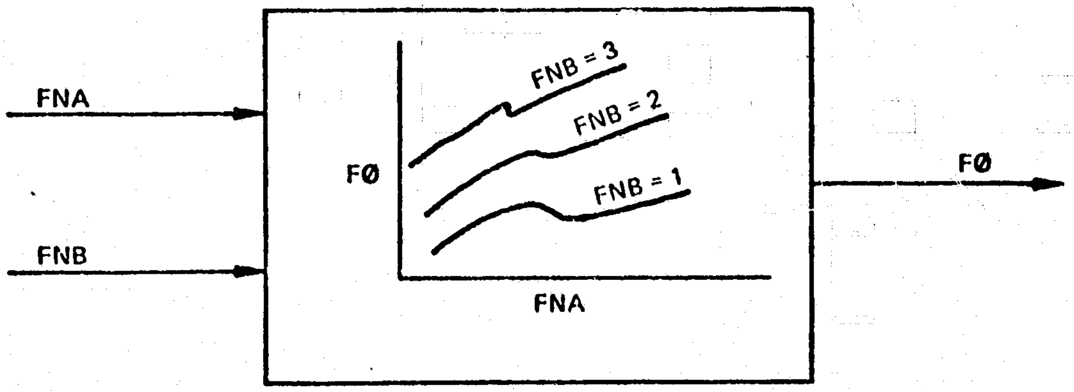
Calculation Sequence

$$FØ = FTA(FIN)$$

NOTE: A maximum of 18 points is allowed in the table.

```
CFU
      SUBROUTINE FU(FTA,FO,FIN,AN)
C
C    PURPOSE - TO CALCULATE OUTPUT FO AS AN ARBITRARY FUNCTION OF
C              INPUT FIN USING TABULAR INPUT FTA GIVING FO=F(FIN)
C
C    METHOD  - SELF EXPLANATORY
C
C
C    LIMITATIONS - MAXIMUM ARRAY SIZE IS 18
C
C
C    WRITTEN BY - ADAM LLOYD                    LATEST REVISION     APRIL  77
C
C
C    INPUT/OUTPUT LIST
C
C    FTA         TABULAR INPUT FO=F(FIN)        ANY              INPUT  TABLE
C    FO          OUTPUT                         ANY              OUTPUT VAR
C    FIN         INPUT                          ANY              INPUT  VAR
C    AN          SET ABS(AN).GT.0.5 FOR UNEQUAL SPACED TABLE DATA---INPUT
C                SET ABS(AN).LE.0.5 FOR EQUI-SPACED TABLE DATA
C                A NEGATIVE VALUE OF AN WILL
C                PREVENT EXTRAPOLATION BEYOND
C                TABLE LIMITS
C
      DIMENSION FTA(1)
      NA= SIGN(FTA(2),AN)
      NB=FTA(2)+4
      N=1
      IF(ABS(AN).LE.0.5) N=0
      FO=TBLU1(FIN,FTA(4),FTA(NB),N,NA)
      RETURN
      END
```

7.13 TWO DIMENSION TABLE LOOKUP



Tables	Description
FTA	Table of functional relationships (maximum number of table values = 144)

Inputs

Parameter/Port	
FNA	Input quantity (primary)
FNB	Input quantity (secondary)
AN	ABS(AN) ≤ 0.5 for equal spaced FNA data*
BN	ABS(BN) ≤ 0.5 for equal spaced FNB data*

Outputs

Variable/Port	
FØ	Output quantity

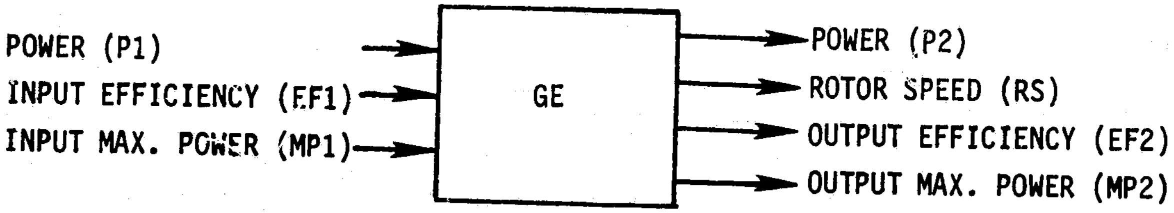
Calculation Sequence

FØ = FTA(FNA,FNB)

* A negative value for AN or BN prevents extrapolation beyond the table boundaries.

```
CFV
      SUBROUTINE FV(FTA,FO,FNA,FNB,AN,BN)
C
C     PURPOSE - TO CALCULATE OUTPUT FO AS AN ARBITRARY FUNCTION OF INPUT
C               VARIABLES FNA AND FNB. INPUT TABLE FTA IS USED GIVING
C               FO=F(FNA,FNB)
C
C     METHOD  - TWO DIMENSIONAL TABLE LOOKUP
C
C
C     LIMITATIONS - MAX ALLOWABLE SIZE OF TABULAR ARRAY IS 12X12.
C
C
C     WRITTEN BY  - GEORGE DULEBA
C                                                 LATEST REVISION   MAY 76
C
C     INPUT/OUTPUT LIST
C
C     FTA         TABULAR INPUT
C     FO          OUTPUT                          ---         INPUT TABLE
C     FNA         INPUT A                         ANY         OUTPUT VAR
C     FNB         INPUT B                         ANY         INPUT  VAR
C     AN          SET ABS(AN) .GT. 0.5 FOR UNEQUAL ANY        INPUT  VAR
C                                  SPACED FNA DATA-           INPUT PARM
C                    A NEGATIVE VALUE INDICATES THAT THE NEAREST END
C                    POINT IS TO BE USED UPON EXTRAPOLATION.
C     BN          SET ABS(BN) .GT. 0.5 FOR UNEQUAL SPACED FNB DATA- INPUT PARM
C                    A NEGATIVE VALUE INDICATES THAT THE NEAREST END
C                    POINT IS TO BE USED UPON EXTRAPOLATION.
C
      DIMENSION FTA(1)
      N1=FTA(3)+4
      N2=FTA(2)+FTA(3)+4
      N3=FTA(2)
      N4=FTA(3)
      N5= SIGN(FTA(2),AN)
      N6= SIGN(FTA(3),BN)
      NAN=1
      IF(ABS(AN).LE.0.5) NAN=0
      NBN=1
      IF(ABS(BN).LE.0.5) NBN=0
      FO=TBLU2(FNA,FNB,FTA(N1),FTA(4),FTA(N2),NAN,NBN,N5,N6,N3,N4)
      RETURN
      END
```

7.14 AC INDUCTION GENERATOR



The induction generator produces electrical power proportional to rotor slip, i.e., difference between rotor speed and synchronous speed. This relationship is used to compute rotor speed given input power and the generator parameters. Two power losses are modeled: a constant multiplicative term due to resistive heating and an additive term due to mechanical friction. Default parameters are based on a conventional squirrel-cage induction motor/generator machine. This component can also be used as a synchronous generator with RAS$\leq$.01.

Basic Equations

Output power P2 and rotor speed RS are computed from the following equations:

$$P2 = EE*(P1-C*RS^2)$$

$$\frac{P2}{RAP} = \frac{(RS/RSY-1)}{RAS}$$ (Power is proportional to slip)

where EE = electrical efficiency

Minimum input parameters to specify GE are

RAP = rated output power,

SR = stator resistance

Note: SR may be chosen to obtain a given efficiency EE using

$$SR = VØ^2(1/EE-1)/(RAP*1000)$$

Inputs

Parameter/Port		Description	Units
P	1	Input power	kw
RAP		Rated output power	kw
RSY		Synchronous rotor speed (D = 1800)	rpm
RAS		Rated power slip (D = 0.05)	-
DA		Mechanical damping (D = 0.0)	joule-sec
SR		Internal stator resistance (D = 6.4/RAP)	ohms
VØ		Rated bus voltage (D = 400)	volts
EF	1	Input product efficiency	-
MP	1	Maximum input discharge rate (D = 1x10^8)	kw
CC		Capital cost/year	$
CM		Maintenance cost/year	$

Outputs

Variable/Port			
P	2	Output power	kw
EE		Electrical efficiency	-
RS		Rotor speed	rpm
PL		Power loss	kw
EF	2	Output product efficiency	-
MP	2	Maximum output discharge rate	kw

Statistics

MPN	Maximum output power/rated power	kw
SP	Total output energy	kwh

D - Default values supplied.

Calculation Sequence

1) First pass only

$$EFF = 1$$

$$I_{RAT} = RAP*1000/V\varnothing$$

$$EE = \frac{RAP}{RAP+SR*I^{2}_{RAT}/1000}$$

2) If $P1 = 0$ set $P2 = 0$, $RS = RSY$ and go to 4)

Compute rotor speed ω in rad/sec using

$$\frac{EE(P1*1000 - \omega^{2}*DA\ )}{RAP*1000} = \frac{(\omega/\omega_{o} - 1)}{RAS}$$

with $\omega_{o} = RSY*(2\pi/60)$

3) Compute RS and output power

$$RS = \omega*(60/2\pi)$$

$$P2 = RAP(RS/RSY-1)/RAS$$

$P2 > RAP$ ⇨ DIAGNOSTIC

$$EFF = P2/P1$$

4) Compute loss, efficiency terms

$$PL = P1-P2$$

$$EF2 = EF1*EFF$$

5) Compute maximum output rate

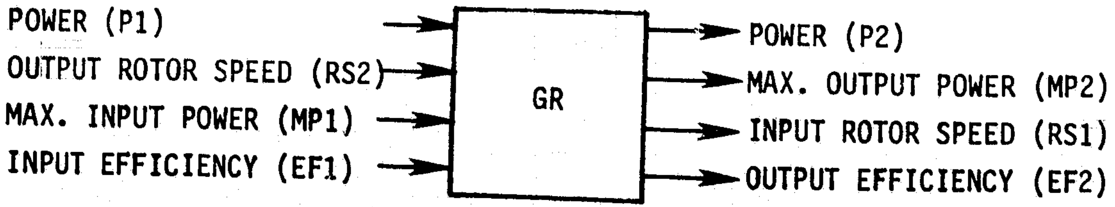
$$MP2 = MIN(RAP,MP1*EFF)$$

6) Compute Statistics and Costs

```
CGE
       SUBROUTINE GE(P2,EE,RS,PL,EF2,PM2,PMN,SP,P1,RAP,RSY,RAS,DA,SR,VO,
      1 EF1,PM1,CCI,CMI)
C
C     PURPOSE     MODEL AC INDUCTION GENERATOR
C
C     METHOD      MECHANICAL AND ELECTRICAL EFFICIENCIES ARE USED TO COMPUTE
C                 OUTPUT POWER.  ROTOR SPEED IS COMPUTED ASSUMING POWER IS
C                 PROPORTIONAL TO SLIP.
C
C     WRITTEN BY A.W. WARREN
C                                                   VERSION 1, MARCH 16 19
C     CALL SEQUENCE
C           OUTPUTS
C                 P2  - OUTPUT POWER, KW
C                 EE  - ELECTRICAL EFFICIENCY
C                 RS  - ROTOR SPEED, RPM
C                 PL  - POWER LOSS, KW
C                 EF2 - OUTPUT PRODUCT EFFICIENCY
C                 PM2 - MAXIMUM OUTPUT POWER, KW
C                 PMN - MAX. OBSERVED OUTPUT POWER / RATED POWER
C                 SP  - TOTAL OUTPUT ENERGY, KWH
C           INPUTS
C                 P1  - INPUT POWER, KW
C                 RAP - RATED OUTPUT POWER, KW
C                 RSY - SYNCHRONOUS ROTOR SPEED, RPMN
C                 RAS - RATED POWER SLIP (DEFAULT = .05)
C                 DA  - MECHANICAL DAMPING, JOULE-SEC
C                 SR  - STATOR RESISTANCE, OHMS
C                 VO  - RATED BUS VOLTAGE, VOLTS
C                 EF1 - INPUT PRODUCT EFFICIENCY
C                 PM1 - MAXIMUM INPUT POWER, KW
C                 CCI - CAPITAL COST/YEAR, $
C                 CMI - MAINTENANCE COST/YEAR, $
C
       COMMON /CIMPL/ IMPL,ICNT /CTIME/ TIME
       COMMON /COST/ CC,CM,CO,CV /CSIMUL/ DUM(6),TINC,TMAX
C
C                            INITIALIZATION
C
       IF( IMPL.GT.0) GO TO 10
       EFF = 1.
       TMAX1 = TMAX* .99999
       IF(RSY.EQ. .99999) RSY = 1800.
       IF(RAS.EQ. .99999) RAS = .05
       IF(DA .EQ. .99999) DA = 0.
       IF(SR .EQ. .99999) SR = 6.4/RAP
       IF(VO .EQ. .99999) VO = 400.
       IF(PM1.EQ. .99999) PM1 = 1.E10
       PMN =0.0
       SP  =0.0
       RATI = RAP*1000./VO
       EE   = RAP/(RAP + SR*.001*RATI**2)
C
C
C                            COMPUTE ROTOR SPEED AND OUTPUT POWER
C
 10 IF( P1.GT. 0.) GO TO 20
    P2 =0.0
```

```
      PL =0.0
      RS = RSY
      GO TO 30
C
   20 A = RAP/(EE*RAS)
      B = RSY/( A + RSY**2*DA*1.0966E-5)
      RS = B*(A + P1)
      P2 = RAP*(RS/RSY -1.)/RAS
      IF (P2.GT.RAP.AND.IMPL.EQ.2) WRITE(6,100)
  100 FORMAT(1H0, 40X,37HGENERATOR OUTPUT EXCEEDS RATED POWER /)
C
      IF(P2.GT.RAP .AND. IMPL.EQ.2)ICNT=ICNT+1
      PL = P1 - P2
      EFF = P2/P1
   30 EF2 = EF1*EFF
      PM2 = AMIN1(RAP, PM1*EFF)
C
C                                 STATISTICS
      IF(IMPL.LE.1) RETURN
      PMN = AMAX1(PMN, P2/RAP)
      SP = SP + P2*.5*TINC
C
C                                 COST SUMMATION
      IF( TIME.LT.TMAX1) RETURN
      CC = CC + CCI
      CM = CM + CMI
C
      RETURN
      END
```

7.15 FIXED RATIO TRANSMISSION



This component models a fixed gear ratio transmission. Power losses are modeled by a table lookup depending on input power. Rotor input speed is used as a feedback variable.

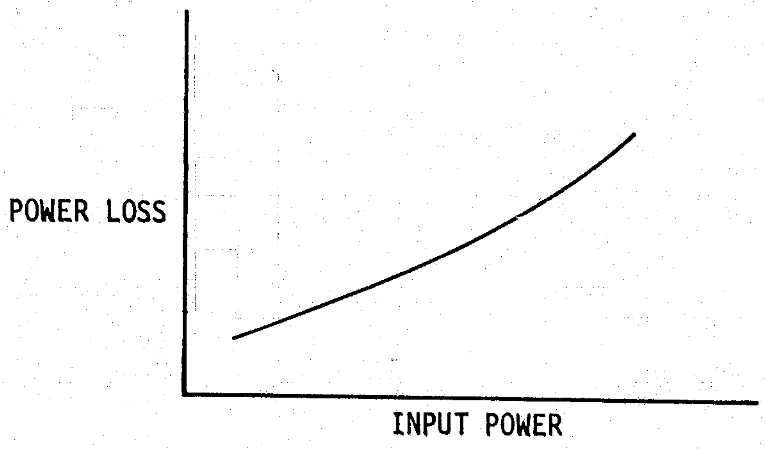


FIGURE 7.15: FIXED GEAR POWER LOSS

GR

Tables	Description	Units
PLØ	Power loss versus input power	kw

Inputs

Parameter/Port			
GR*		Gear ratio	-
RS	2	Output rotor speed	rpm
P	1	Input power	kw
EF	1	Input product efficiency	-
MP	1	Maximum input power (Default = $1x10^8$)	kw
CC		Capital cost/year	$
CM		Maintenance cost/year	$

Outputs

Variable/Port			
P	2	Output power	kw
TØ		Output torque	ft-lb
PL		Power loss	kw
EF	2	Output product efficiency	-
MP	2	Maximum output power	kw
RS	1	Input rotor speed	rpm

* A value for GR is supplied when connecting to the wind turbine component WT.

Calculation Sequence

1) MP2 = MP1 - PLØ(MP1) (First Pass Only)

If P1 ≤ 0, set PL = P2 = 0 and go to 2)

PL = PLØ(P1)

P2 = P1 - PL

RS1 = RS2/GR

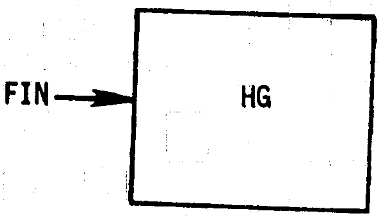
EF2 = EF1*P2/P1

2) TØ = P2*737.6/(RS2*2π/60)

3) Compute Costs

```
CGR
      SUBROUTINE GR(PLO,P2,TO,PL,EF2,PM2,RS1,GRA,RS2,P1,EF1,PM1,CCI,CMI)
C
C     PURPOSE   MODEL A FIXED GEAR RATIO TRANSMISSION
C
C     METHOD    POWER LOSSES ARE INPUT AS A FUNCTION OF INPUT POWER P1.
C
C     WRITTEN BY A.W. WARREN
C                                                 VERSION 1, MARCH 16 19
C
C     CALL SEQUENCE
C          TABLES
C               PLO - POWER LOSS IN KW VERSUS INPUT POWER IN KW
C
C          OUTPUTS
C               P2  - OUTPUT POWER, KW
C               TO  - OUTPUT TORQUE, FT-LB
C               PL  - POWER LOSS, KW
C               EF2 - OUTPUT PRODUCT EFFICIENCY
C               PM2 - MAXIMUM OUTPUT POWER, KW
C               RS1 - INPUT ROTOR SPEED, RPM
C
C          INPUTS
C               GRA - GEAR RATIO
C               RS2 - OUTPUT ROTOR SPEED, RPM
C               P1  - INPUT POWER, KW
C               EF1 - INPUT PRODUCT EFFICIENCY
C               PM1 - MAXIMUM INPUT POWER, KW
C               CCI - CAPITAL COST / YEAR, $
C               CMI - MAINTENANCE COST / YEAR, $
C
      DIMENSION PLO(1)
      COMMON /CIMPL/IMPL /CTIME/ TIME
      COMMON /COST/CC,CM,CO,CV /CSIMUL/ DUM(7),TMAX
C
C                             INITIALIZATION
C
      NP = PLO(2)
      IF( IMPL.GT.0) GO TO 10
      TMAX1 = .99999*TMAX
      EF2=1.
      RS2=1.
      IF(PM1.EQ. .99999) PM1=1.E10
      PM2 = PM1
      IF(PM1.LE. PLO(3+NP) ) PM2 = PM1-TBLU1(PM1,PLO(4),PLO(4+NP),1,-NP)
C
C                             POWER LOSS AND ROTOR SPEED CALUCATIONS
   10 PL=0.
      P2=0.
      IF(P1 .EQ. 0.) GO TO 20
      PL = TBLU1(P1,PLO(4),PLO(4+NP),1,-NP)
      P2 = P1 - PL
      EF2 = EF1*P2/ P1
      RS1 = RS2/GRA
   20 IF(RS2 .GT. 0.) TO = P2*7043./RS2
C
C                             COST SUMMATION
      IF(IMPL.LE.1) RETURN
      IF(TIME.LE.TMAX1) RETURN
      CC = CC + CCI
      CM = CM + CMI
      RETURN
      END
```

7.16 HISTOGRAM



The input quantity is monitored during a SIMULATE analysis. When time reaches TMAX a plotted histogram is produced with 16 intervals that span the range from FLO to FUP.

Inputs

Variable/Port	Description
FIN	Input quantity to be monitored
FUP	Upper limit for histogram
FLØ	Lower limit for histogram
F1,...F16[1]	Array containing histogram data
FA[1]	Measurement interval

Outputs

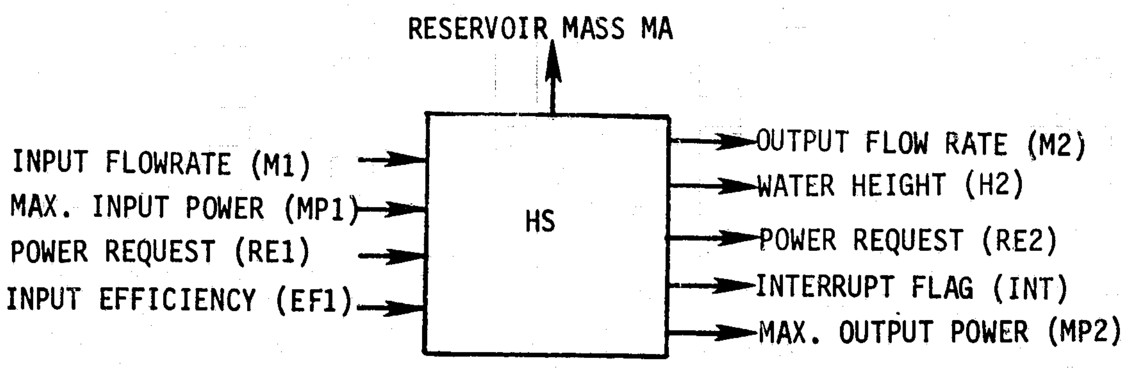
Variable/Port	
AV	Mean value (running sum during simulation)
SD	Standard deviation (running sum squared)
SAM	Number of samples

[1] These quantities do not require data input values.

```
CHG
      SUBROUTINE HG(SAMP,AV,SD,F1,F2,F3,F4,F5,F6,F7,F8,F9,F10,F11,
     1 F12,F13,F14,F15,F16,FA, FIN,FUP,FLO )
C       VERSION 2.          REVISED:  MARCH  1977
C   PURPOSE - DEVELOP A RUNNING HISTOGRAM OF AN INPUT SEQUENCE
C   CALL SEQUENCE
C                SAMP- OUTPUT NUMBER OF SAMPLES
C                AV  - OUTPUT AVERAGE (RUNNING SUM)
C                SD  - OUTPUT STANDARD DEVIATION (RUNNING SUM SQUARED)
C            F1-F16  -  ARRAY WITH NUMBER OF OCCURENCES IN EACH INTERVAL
C               FA   - OUTPUT CONTAINING MEASUREMENT INTERVAL
C               FUP  - INPUT SPECIFYING UPPER MEASUREMENT LIMIT
C               FLO  - INPUT SPECIFYING LOWER MEASUREMENT LIMIT
C               FIN  - INPUT MEASUREMENT
      DIMENSION F1(16),TD1(8),AX1(16)
      DIMENSION GRAPH(114,46)
      COMMON GRAPH
      COMMON/CTIME/TIME/CSIMUL/DUM(7),TMAX
      COMMON/CCVRLY/DUMM(3),CPUSEC /CIMPL/IMPL
      DATA BLANK,VERT,HORIZ,POINT/1H ,1HI,1H-,1H*/
      IF(IMPL.GT.0) GO TO 100
      DO 50 I=1,16
   50 F1(I)=0.
      FA=(FUP-FLO)/14.
      SD=0.
      AV =0.0
      SAMP=0.0
  100 CONTINUE
      IF(IMPL.LT.2) RETURN
      DO 200 I=1,16
      L=I
      FAX=FLO+(I-1)*FA
      IF(FIN.LE.FAX) GO TO 300
  200 CONTINUE
  300 F1(L)=F1(L)+1.
      SAMP=SAMP+1.
      AV=AV+FIN
      SD=SD+FIN*FIN
      IF(TIME.LT.TMAX*.99999)RETURN
      SAMP=0.
      DO 350 I=1,16
  350 SAMP=SAMP+F1(I)
      ISAMP=SAMP
      ISAMP=MAX0(1,ISAMP)
      AV=AV/ISAMP
      SD=SQRT(SD/ISAMP-AV*AV)
      XMAX=F1(1)
      DO 360 I=1,16
  360 IF(F1(I).GE.XMAX) XMAX=F1(I)
      IF(XMAX.EQ.0.) XMAX=10.
      HX=XMAX/44.
      DO 370 I=1,46
      GRAPH(1,I)=VERT
  370 GRAPH(114,I)=VERT
      DO 380 I=2,113
      GRAPH(I,1)=HORIZ
  380 GRAPH(I,46)=HORIZ
      DO 400 I=5,103,14
```

```
 400  GRAPH(I,46)=VERT
      DO 450 I=8,106,7
 450  GRAPH(I,1)=VERT
      DO 500 I=2,45
      DO 500 J=2,113
 500  GRAPH(J,I)=BLANK
      DO 600 IC=1,16
      J=IFIX(45.5-F1(IC)/HX)
      DO 600 J1=1,7
      J2=(IC-1)*7+J1+1
      DO 600 J3=J,45

 600  GRAPH(J2,J3)=POINT
      DO 700 I=1,16

 700  AX1(I)=F1(I)/ISAMP
      DO 800 I=1,8

 800  TD1(I)=FLO+(I-1)*2.*FA-FA/2.

      WRITE(6,900)(GRAPH(I,1),I=1,114)
 900  FORMAT(1H1,9X,114A1/)
      WRITE(6,1000)(AX1(I),I=1,16)
1000  FORMAT(1H+,9X,1HI,16F7.5,1HI/)
      WRITE(6,1100)
1100  FORMAT(1H+,9X,1HI,112X,1HI/)
      WRITE(6,1200)((GRAPH(I,J),I=1,114),J=2,46)
1200  FORMAT(1H+,9X,114A1/45(10X,114A1/))
      WRITE(6,1300)(TD1(I),I=1,8)
1300  FORMAT (1H+,9X,8(F13.5,1X)//)
      WRITE(6,1400) ISAMP,AV,SD
1400  FORMAT(1H+,10X,14HHISTOGRAM FOR ,I7,8H SAMPLES,
     19H    MEAN= ,F13.5,18H    STANDARD DEV.= ,F13.5/)
      RETURN
      END
```

7.17 HYDRO STORAGE VESSEL



The hydro storage vessel is modeled as an above ground reservoir with a large and constant surface area. The change in reservoir height between maximum and minimum levels is assumed small in comparison to the height of the water above the turbine. Hence, reservoir height is assumed constant. The reservoir has specified evaporation and leakage rates. Average input flow gained by rainfall is also specified. Energy storage is calculated based on the potential energy of the water in the reservoir relative to the turbine inlet.

Basic Equation

$$\dot{MA} = M1 - M2 - NE*AS - NL + MDR*AS/14052$$

Inputs

Parameter/Port		Description	Units
M	1	Input water mass flow rate	gal/h
NE		Evaporation coefficient (D = 0.03)	gal/ft^2-h
AS		Reservoir surface area	ft^2
NL		Leakage coefficient (D = 8.0)	gal/h
MDR		Rainfall rate	inches/year
MDM		Maximum allowable mass flow rate (D = 4X10^5)	gal/h
MM		Maximum allowable reservoir capacity (D=5X10^6)	gal
MØ		Minimum allowable reservoir capacity	gal
H	1	Reservoir height above turbine	ft
MDE		Reservoir deadband for priority resequence	gal
RE	1	Power request (discharge)	kw
CR		Reservoir cost coefficient (D = 0.025)	$/gal
EF	1	Input product efficiency	-
MP	1	Maximum input charging rate	kw
LE		Reservoir life expectancy	years
CM		Maintenance cost/year	$

Outputs

Variable/Port			
M	2	Outlet water mass flow rate	gal/hr
E		Energy stored	kwh
H	2	Reservoir height above turbine (=H1)	ft
MA		Reservoir mass (state)	gal
CCØ		Reservoir cost/year	$
MP	2	Maximum discharge rate allowable	kw
INT		Priority interrupt flag	-
RE	2	Maximum charging rate request	kw

D - Default values supplied

Statistics	Description	Units
MDU	Maximum mass flow rate	gal/hr
MU	Maximum reservoir mass	gal
ML	Minimum reservoir mass	gal

The calculation sequence and default values assume a pond sized for 120kw storage for 24 hours (5×10^6 gallons of water 200 ft. above turbine inlet). The evaporation coefficient NE assures the pond drops ½" in height per 10 hours. To obtain a more accurate value for this parameter requires knowledge of local conditions. The leakage coefficient NL is based on the assumption of a loss of 0.1% of the maximum reservoir capacity in the rated storage time of 24 hours. The reservoir cost estimates are based on the compensation reservoir given in Reference 1.

[1] "Preliminary Feasibility Evaluation of Compressed Air Storage Power Systems," United Technologies AER 74-00242, December 1976.

Calculation Sequence

C1 = conversion constant = 0.377×10^{-6} $\frac{\text{kwh}}{\text{ft-lb}}$

C2 = conversion constant = 8.3398 lb/gal

$A = C1*C2*H1$

1) Reservoir cost

$CC = CR*MM/LE$

2) Volume of water discharged

$M2 = RE1/A$

3) Reservoir water volume

$\dot{M}A = M1 - M2 - NE*AS-NL +(MDR*AS/14052.)$

4) Energy stored

$E = A*M$

5) Checks

$M1 > MDM$ or $M2 > MDM$ ⇨ DIAGNOSTIC

$M>MM$, ⇨ DIAGNOSTIC

$M<MØ$ ⇨ DIAGNOSTIC

6) Priority interrupt

If $M \le MØ$, INT = 1

IF $M>MØ+MDE$ and INT=1, INT = 0

If $M \ge MM$, INT = -1

If $M<MM-DME$ and INT = -1, INT = 0

Calculation Sequence Cont.

7) Maximum charging rate request

MD1 = MIN (MDM,(MM-M)/TINC)

RE2 = MIN (MP1,MD1*A)/EF1

Maximum discharge rate

MP2 = A * MIN (MDM,(M-MØ)/TINC)

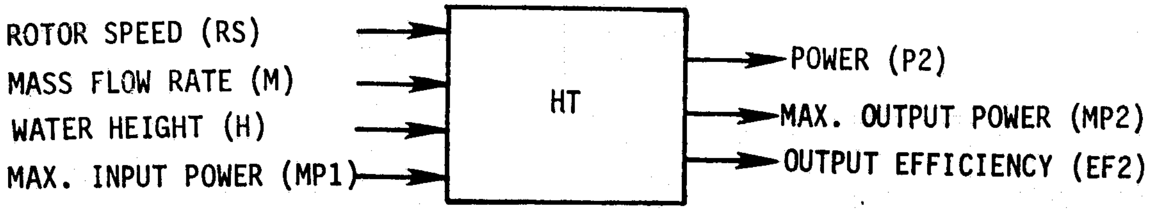
where TINC = integration step size in hrs

8) Compute Statistics and Costs

```
CHS
      SUBROUTINE HS(M,DM,IM,M2,E,H2,CC,MP2,INT,RE2,MDU,MU,ML,M1,NE
     1             ,AS,NL,MDR,MDM,MM,MO,H1,MDE,RE1,CR,EF1,MP1,LE,CM)
C
C    PURPOSE     PERFORMANCE OF A LARGE RESERVOIR AS AN ENERGY STORAGE
C
C                DEVICE.
C
C    METHOD      ENERGY IN STORAGE IS CALCULATED FROM THE POTENTIAL
C
C                BETWEEN THE RESERVOIR AND THE TURBINE INLET.
C
C    WRITTEN BY F. D. MAHONY                       VERSION 1, MARCH 30 1977
C
C    CALL SEQUENCE
C          OUTPUTS
C                M    - RESERVOIR MASS (STATE VARIABLE), GAL
C                DM   - RESERVOIR MASS FLOWRATE, GAL/HR
C                IM   - STATUS INDICATOR
C                M2   - OUTLET WATER MASS FLOW RATE, GAL/HR
C                E    - ENERGY STORED, KWH
C                H2   - RESERVOIR HEIGHT ABOVE TURBINE (=H1), FT
C                CC   - RESERVOIR COST/YEAR, $
C                MP2  - MAXIMUM DISCHARGE RATE ALLOWABLE, KW
C                INT  - PRIORITY INTERUPT FLAG
C                RE2  - MAXIMUM CHARGING RATE REQUEST, KW
C                MDU  - MAXIMUM MASS FLOW RATE, GAL/HR
C                MU   - MAXIMUM RESERVOIR MASS, GAL
C                ML   - MINIMUM RESERVOIR MASS, GAL
C
C          INPUTS
C                M1   - INPUT WATER MASS FLOW RATE, GAL/HR
C                NE   - EVAPORATION COEFFICIENT, GAL/FT**2-HR
C                AS   - RESERVOIR SURFACE AREA, FT**2
C                NL   - LEAKAGE COEFFICIENT
C                MDR  - RAINFALL RATE, INCHES/YEAR
C                MDM  - MAXIMUM ALLOWABLE MASS FLOW RATE, GAL/HR
C                MM   - MAXIMUM ALLOWABLE RESERVOIR CAPACITY, GAL
C                MO   - MINIMUM ALLOWABLE RESERVOIR CAPACITY, GAL
C                H1   - RESERVOIR HEIGHT ABOVE TURBINE, FT
C                MDE  - RESERVOIR DEADBAND FOR PRIORITY RESEQUENCE, GAL
C                RE1  - POWER REQUEST (DISCHARGE), KW
C                CR   - RESERVOIR COST COEFFICIENT
C                EF1  - INPUT PRODUCT EFFICIENCY
C                MP1  - MAXIMUM INPUT CHARGING RATE, KW
C                LE   - RESERVOIR LIFE EXPECTANCY, YEARS
C                CM   - MAINTENANCE COST/YEAR, $
C
      COMMON/CIMPL/IMPL,ICNT/CTIME/TIME/CSIMUL/DUM(7),TMAX
      COMMON/COST/CCI,CMI
      REAL M2,MP2,MDU,MU,ML,M1,NE,NL,MDR,MDM,MM,MO,MDE,MP1,LE,INT,M
      REAL MD1,MDM1
C
      IF(IMPL.GT.0)GO TO 100
C
      RE1=0.0
      H2=H1
      TMAX1=TMAX*0.99999
      TINC=DUM(7)
      C1 = 3.1441E-6
C
      INT=0.0
      MDU=0.0
      MU =0.0
      ML =1.0E10
      IF(NE .EQ. .99999)NE =0.03
      IF(NL .EQ. .99999)NL =8.0
      IF(MDM.EQ. .99999)MDM=4.0E5
      IF(MM .EQ. .99999)MM =5.0E6
      IF(CR .EQ. .99999)CR =0.025
C
C
C                          RESERVOIR COST
C
      CC =CR*MM/LE
C
C                          VOLUME OF WATER DISCHARGED
C
  100 A=C1*H1
      M2 =RE1/A
C
C                          RESERVOIR MASS FLOW RATE
C
      IF(IM.NE.0)DM=M1-M2-NE*AS-NL+MDR*AS/14052.0
C
C                          ENERGY STORED
C
      E =A*M
C
      MDM1=MDM/.9999
      IF(M1.LT.MDM1.AND.
     1   M2.LT.MDM1)GO TO 200
C
      IF(IMPL.EQ.2)WRITE(6,1010)M1,M2,MDM
      IF(IMPL.EQ.2) ICNT=ICNT+1
C
  200 IF(M .LT.MM+MDE)GO TO 300
C
      IF(IMPL.EQ.2)WRITE(6,1020)M,MM
      IF(IMPL.EQ.2)ICNT=ICNT+1
C
  300 IF(M .GT.MO)GO TO 400
C
      IF(IMPL.EQ.2)WRITE(6,1030)M,MO
      IF(IMPL.EQ.2) ICNT=ICNT+1
C
C                          PRIORITY INTERRUPT
C
  400 IF(M .LE.MO)INT=1.0
      IF(M .GT.(MO+MDE).AND.
     1  INT.EQ.1.0)INT=0.0
      IF(M .GT.MM)INT=-1.0
      IF(M .LT.(MM-MDE).AND.
     1    INT.EQ.-1.0)INT=0.0
C
C                          MAXIMUM CHARGE RATE REQUEST AND DISCHARGE RATE
```

```
C
      MD1=AMIN1(MDM,AMAX1(0.,(MM-M)/TINC))
      RE2=AMIN1(MP1,MD1*A)/EF1
      MP2=A*AMIN1(MDM,AMAX1(0.,(M-MO)/TINC))
C
      IF(IMPL.LE.1)RETURN
C
C                      STATISTICS
C
      MDU=AMAX1(DM,MDU)
      MU =AMAX1(M ,MU )
      ML =AMIN1(M ,ML )
C
      IF(TIME.LT.TMAX1)RETURN
C
      CCI=CCI+CC
      CMI=CMI+CM
C
      RETURN
C
 1010 FORMAT(1H0,23HHS INLET MASS FLOW RATE,F12.3, 5H  OR ,
     1           21HOULET MASS FLOW RATE,F12.3,
     2           26H   IS GREATER THAN MAXIMUM,F12.3)
 1020 FORMAT(1H0,19HHS RESERVOIR VOLUME,F12.3,
     1           30H    EXCEEDED MAXIMUM ALLOWABLE,F12.3)
 1030 FORMAT(1H0,19HHS RESERVOIR VOLUME,F12.3,
     1           24H    DROPED BELOW MINIMUM,F12.3)
C
      END
```

7.18 HYDRAULIC TURBINE



The hydraulic turbine model is based on a constant speed design and is typical of a reaction/Francis type turbine. The turbine is assumed to be designed to a specified operating point and output power.

For off design performance the pump efficiency is assumed to be functionally related to the first power of the mass flow rate. The equations are assumed to be valid over a specified range of values for the turbine parameter.

Basic Equations

$$P = EFF*M*C1*C2*H$$

$$EFF = 1-(1-EFD)*MD/M$$

where C1, C2 are conversion constants.

HT

Inputs

Parameter/Port		Description	Units
M		Inlet mass flow rate	gal/h
H		Height of reservoir above turbine inlet	ft
EFD		Design pt. turbine efficiency (D = 0.90)	-
MD		Design pt. mass flow rate (D = $2X10^5$)	gal/h
MM		Maximum mass flow rate (D = $3X10^5$)	gal/h
EF	1	Input product efficiency	-
MP	1	Input maximum discharge rate	kw
CK		Turbine capacity cost coefficient[1] (D = 0.011)	-
FØ		Turbine exponent for cost calculations (D = 0.5)	-
RS		Angular velocity	rpm
X		Turbine head exponent for cost calculations (D = 0.25)	-

Outputs

Variable/Port			
CCØ		Turbine cost/year	$
EFF		Turbine efficiency	-
P	2	Output power	kw
EF	2	Output product efficiency	-
MP	2	Output maximum discharge rate	kw
CP		Turbine characteristic parameter	-

Statistics

CPU	Maximum CP	-
CPL	Minimum CP	-
PU	Maximum output power	kw

D - Default values

[1] CK = Capital cost (known unit)/((MD*481.2)**FO*H**X*life expectancy)

The calculation sequence and default values assume a constant speed reaction type hydraulic turbine nominally rated for 120kw and located 200 ft. below the reservoir. The equations relating the various physical parameters are assumed to be valid for the indicated range of the characteristic turbine parameter, CP. The equations and cost estimates are based on the data given in Reference 1, and the cost estimates on data from Reference 2.

Calculation Sequence

$$C1 = 0.377\times10^{-6} \frac{\text{kwh}}{\text{ft-lb}}$$

$$C2 = 8.3398 \text{ lb/gal}$$

$$A = C1*C2*H$$

1) Costs

$$CC\emptyset = CK*(MD*481.2)**F\emptyset*H**X$$

2) Efficiency

If $M \leq 0$ set EFF = 1 and go to 3)

$$EFF = 1-(1-EFD)*MD/M$$

$$EFF = MAX(EFF, 0.6)$$

[1] L. Marks and T. Baumeister, "Mechanical Engineers Handbook", McGraw Hill, N.Y., 1958, Section 9, p. 207.

[2] Carson and Fogleman, "Comparison of Methods for Converting Existing Power Plants to Pumped Storage Facilities", International Engineering Company, Inc., 1974.

Calculation Sequence Cont.

3) Output Power

$$P2 = EFF*A*M$$

4) Product Efficiency

$$EF2 = EF1*EFF$$

$$EFM = MM - (1-EFD)*MD$$

5) Maximum Discharge Rate

$$MP2 = Min\{MP1*EFD, EFM*A\}$$

6) Turbine Characteristic Parameter

(If P2 ≤ 0 go to 7)

$$CP = RS*SQRT(P2*0.746)/H**1.25$$

If CP>100 write DIAGNOSTIC

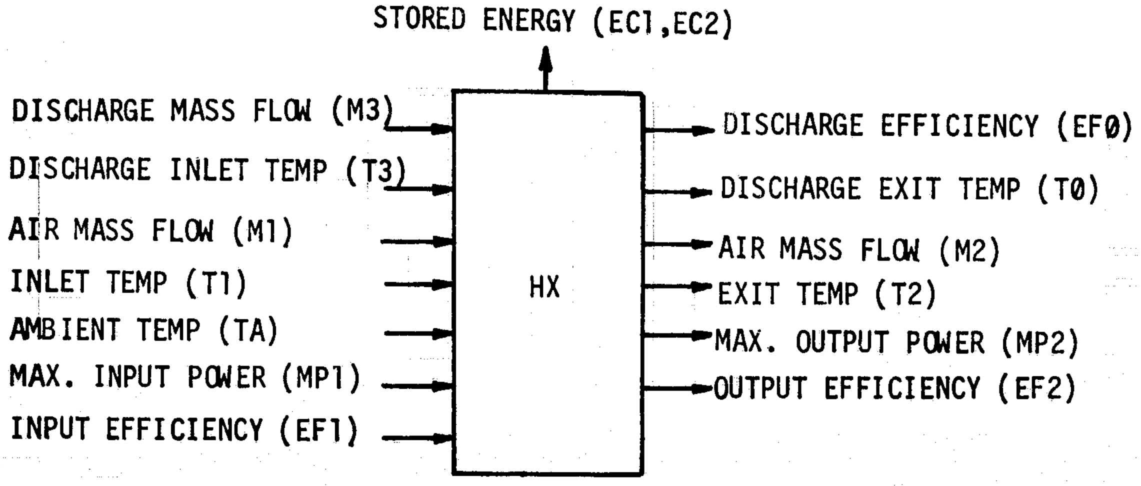
If M> MM write DIAGNOSTIC

7) Compute Statistics and Costs

```
CHT
      SUBROUTINE HT(CC,EFF,P,EF2,MP2,CP,CPU,CPL,PU,M,H,EFD,MD,MM
     1              ,EF1,MP1,CK,FO,RS,X)
C
C     PURPOSE    PERFORMANCE OF A HYDRAULIC TURBINE
C
C     METHOD     OFF DESIGN PERFORMANCE IS ASSUMED PROPORTIONAL TO
C
C                MASS FLOW RATE
C
C         WRITTEN BY F. G. MAHONY                   VERSION 1, MARCH 30 1977
C
C     CALL SEQUENCE
C          OUTPUTS
C                CC  - TURBINE COST/YEAR, $
C                EFF - TURBINE EFFICIENCY
C                P   - OUTPUT POWER, KW
C                EF2 - OUTPUT PRODUCT EFFICIENCY
C                MP2 - OUTPUT MAXIMUM DISCHARCE RATE, KW
C                CP  - TURBINE CHARACTERISTIC PARAMETER
C                CPU - MAXIMUM CP
C                CPL - MINIMUM CP
C                PU  - MAXIMUM OUTPUT POWER, KW
C
C          INPUTS
C                M   - INLET MASS FLOW RATE, GAL/HR
C                H   - HEIGHT OF RESERVOIR ABOVE TURBINE INLET, FT
C                EFD - DESIGN POINT TURBINE EFFICIENCY
C                MD  - DESIGN POINT MASS FLOW RATE, GAL/HR
C                MM  - MAXIMUM MASS FLOW RATE, GAL/HR
C                EF1 - INPUT PRODUCT EFFICIENCY
C                MP1 - INPUT MAXIMUM DISCHARGE RATE
C                CK  - TURBINE CAPACITY COST COEFFICIENT
C                FO  - TURBINE EXPONENT FOR COST CALCULATIONS
C                RS  - ANGULAR VELOCITY, RPM
C                X   - TURBINE HEAD EXPONENT FOR COST CALCULATIONS
C
      COMMON/CIMPL/IMPL,ICNT/CTIME/TIME /CSIMUL/DUM(7),TMAX /COST/ CCI
      REAL MP2,M,MD,MM,MP1
C
      IF(IMPL.GT.0)GO TO 100
C
      TMAX1=TMAX*0.99999
      RS =3600.0
C
      IF(EFD.EQ. .99999)EFD=0.9
      IF(MD .EQ. .99999)MD =2.0E5
      IF(MM .EQ. .99999)MM =3.0E5
      IF(CK .EQ. .99999)CK =0.011
      IF(FO .EQ. .99999)FO =0.5
      IF(X  .EQ. .99999)X  =0.25
C
      CPL=1.0E10
      CPU=0.0
      PU =0.0
C
      C1 = 3.1441E-6
      CC =CK*(MD*481.2)**FO*H**X
```

```
C
C                          EFFICIENCY
  100 EFF =1.0
C
      IF(M.LE.0.0)GO TO 400
C
      EFF=1.0-(1.0-EFD)*MD/M
      IF(EFF.LT.0.6) EFF=0.6
C
C                          OUTPUT POWER
C
  400 P   =EFF*M*H*C1
C
C                          PRODUCT EFFICIENCY
C
      EF2=EF1*EFF
C
C                          MAXIMUM DISCHARGE RATE
C
      EFM =MM*.9999-(1.0-EFD)*MD
C
      MP2=AMIN1(MP1*EFD,EFM*H*C1)
C
C                          TURBINE CHARACTERISTIC PARAMETER
C
      IF(P .LE. 0.0) GO TO 300
      CP =RS*SQRT(P*0.746)/H**1.25
C
      IF(CP.LT.100.0)GO TO 200
C
      IF(IMPL.EQ.2)WRITE(6,1010)CP
      IF(IMPL.EQ.2) ICNT=ICNT+1
C
  200 IF(M.LT.MM)GO TO 300
C
      IF(IMPL.EQ.2)WRITE(6,1020)M,MM
      IF(IMPL.EQ.2) ICNT=ICNT+1
C
  300 IF(IMPL.LE.1)RETURN
C
C                          STATISTICS
C
      CPU=AMAX1(CPU,CP)
      CPL=AMIN1(CPL,CP)
      PU =AMAX1(PU ,P )
C
      IF(TIME.LT.TMAX1)RETURN
C
C                          COST
C
      CCI=CCI+CC
C
      RETURN
C
 1010 FORMAT(1H0,48HHT TURBINE CHARACTERISTIC PARAMETER OUT OF RANGE,
     XF12.3)
 1020 FORMAT(1H0,23HHT INLET MASS FLOW RATE,F12.3
     1          ,37H    GREATER THAN MAXIMUM DESIGN VALUE,F12.3)
      END
```

7.19 ADIABATIC HEAT EXCHANGER



The purpose of the adiabatic heat exchanger is to recover a portion of the heat of compression from the high pressure, high temperature air exiting from the compressor. Figure 7.19-1 shows an adiabatic heat exchanger used in an underground, constant pressure compressed air energy storage system. The adiabatic heat exchanger operates in a manner similar to the high temperature thermal energy storage systems currently conceived for solar thermal power plants[1]. In the storage charging mode, high pressure, high temperature air enters the top of the heat exchanger and deposits a portion of its thermal energy in the storage media as either sensible heat or latent heat of fusion. The exiting high pressure air is stored in an appropriate vessel, e.g., underground cavern. In the discharge cycle (HY), high pressure, low temperature air enters the bottom of the heat exchanger, recovers thermal energy from the storage media and exits to the turbine.

The adiabatic heat exchanger model is based on a two cell storage model. Given the stored energy in both cells, a linear temperature profile is computed

[1] BEC/EPRI RP 788-1, "Advanced Thermal Energy Storage Systems," November 1976.

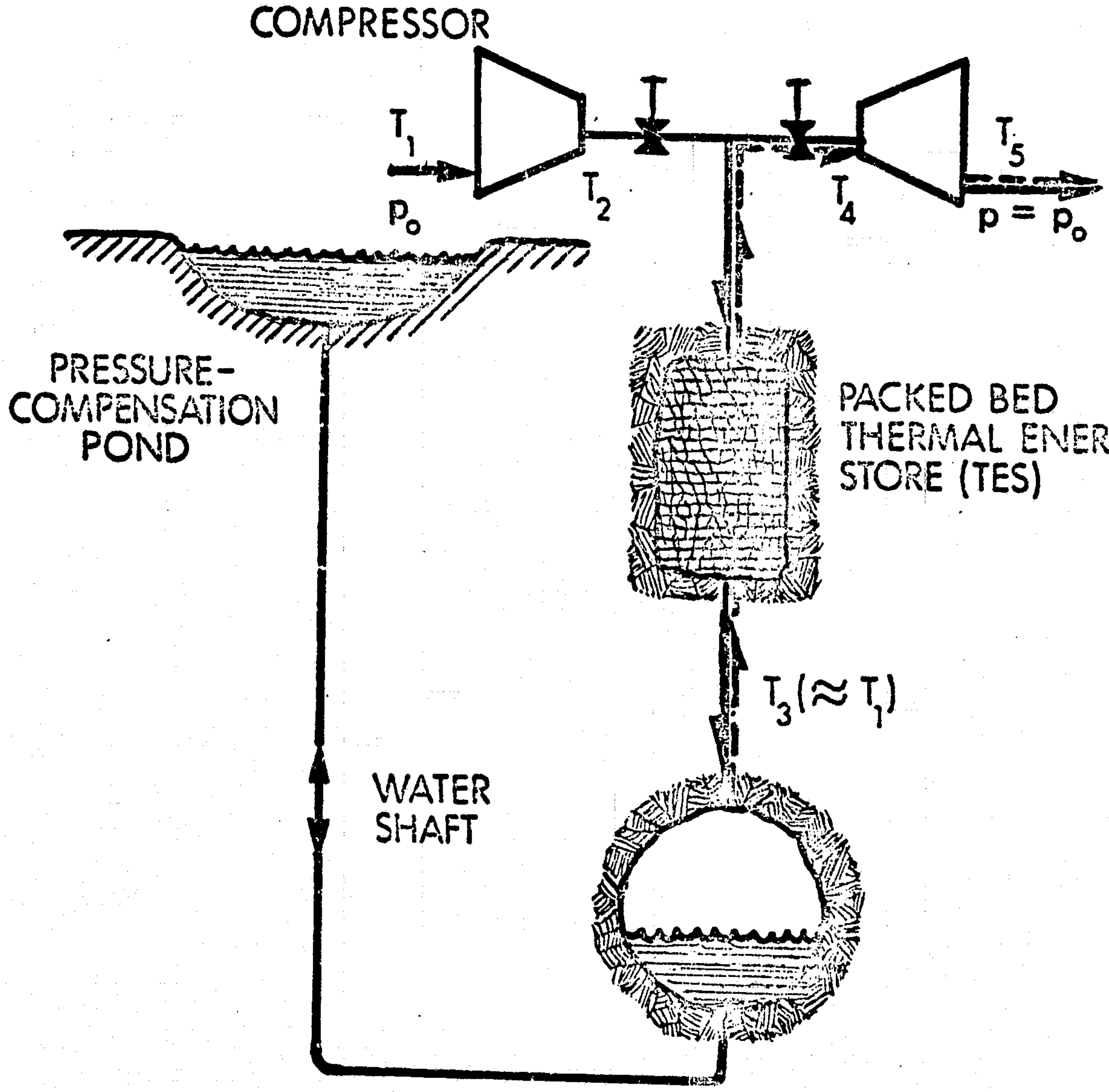


FIGURE 7.19-1 KOUTZ-GLENDENNING ADIABATIC COMPRESSED AIR STORAGE SCHEME (SINGLE-STAGE HEAT-OF-COMPRESSION STORAGE)

for the media mass. Based on a given inlet mass flow rate, the convective film coefficient, unit thermal conductance, and heat exchanger exit temperature are calculated.

The rate of energy deposited (or withdrawn) is calculated and integrated to yield the stored energy state. For a phase change media, the temperature profile is approximated in the following way: Average cell temperatures TS1 and TS2 are determined from the enthalpy diagram (Figure 7.19-2) using average cell entropy EC1/MA and EC2/MA, respectively. Then a linear temperature profile is constructed as shown in Figure 7.19-3.

Basic Equations

$$\dot{EC1} = PX - PY - NU * EC1 - BE * (EC1 - EC2)$$

$$\dot{EC2} = (P2 - PX) - (PØ - PY) - NU*EC2 + BE * (EC1 - EC2)$$

where

EC1, EC2 = storage power in cells 1 and 2, respectively

PX = charging power in cell 1

PY = discharging power in cell 1

P2 - PX = charging power in cell 2

PØ - PY = discharging power in cell 2

NU = storage media leakage constant

BE = storage media mixing constant

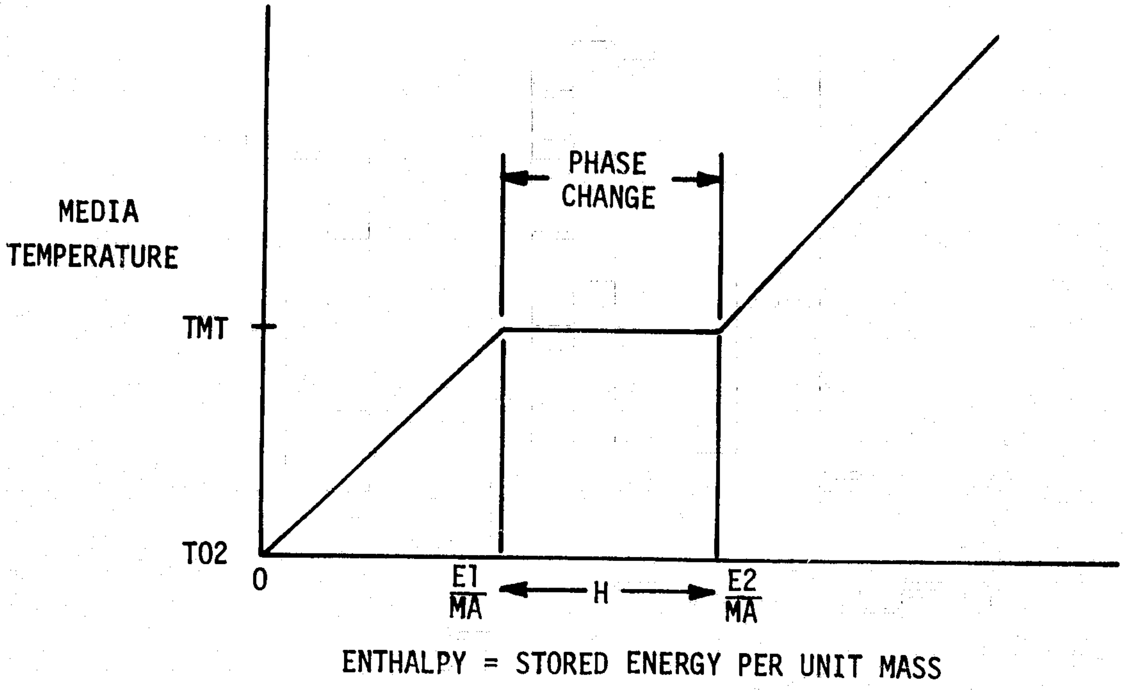


FIGURE 7.19-2: ENTHALPY-TEMPERATURE DIAGRAM FOR HX

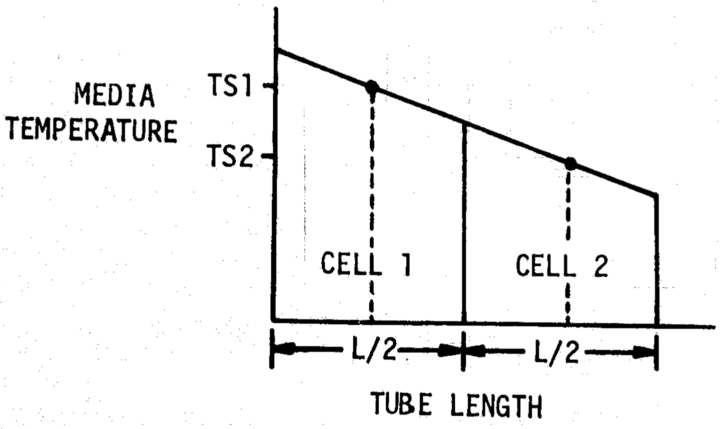


FIGURE 7.19-3: STORAGE TEMPERATURE VERSUS TUBE LENGTH

Inputs

Parameter/Port		Description	Units
NU		Storage energy loss coefficient (D = 0.002)	$(h)^{-1}$
ST		Rated storage time[1]	h
BE		Storage energy mixing coefficient (D = 0.0)	h^{-1}
TØ1		Minimum allowable storage temperature (D = 60)	°F
DTD		Media temperature swing[1] (D = 400)	°F
PD		Rated storage thermal power	kw
TEM		Maximum allowable exit temperature (D = 240)	°F
XD		Design point fraction of molten media mass (D = 0.8)	-
EF	1	Input product efficiency	-
MP	1	Maximum input charging rate	kw
CP1		Storage media heat capacity (D = $2.93X10^{-4}$)	kwh/lb °F
H		Storage media heat of fusion[2] (D = 0.0219)	kwh/lb
TMT		Storage media melt temperature [2] (D = 147)	°F
CPF		Air heat capacity (D = $7.6X10^{-5}$)	kwh/lb °F
KF		Air thermal conductivity (D = $1.03X10^{-4}$)	kw/ft °F
MU		Air viscosity (D = 0.055)	lb/ft-h
NT		Number of tubes (D = 200)	-
D		Tube diameter (D = 0.03)	ft
L		Tube length (D = 4)	ft
DEL		Tube half spacing (D = 0.085)	ft
K		Storage media thermal conductivity (D = 0.0078)	kw/ft-°F
T	1	Inlet air temperature	°F
M	1	Inlet mass flow rate	lb/h
CM		Storage device yearly maintenance cost (D = 0.6)	$/kw
CSA		Storage device capacity cost (D = 50)	$/kw
CSB		Storage device energy cost (D = 15.6)	$/kwh
LE		Unit life expectancy	years

D - Default values specified

[1] - Design point conditions

[2] - Used for phase change media, H = 0 for sensible heat

Inputs

Parameter/Port		Description	Units
M	3	Discharge cycle mass flow rate from storage	lb/hr
T	3	Discharge cycle temperature from storage	°F
TA		Ambient temperature	°F
TSØ		Storage vessel minimum temperature	°F

Outputs

Variable/Port		Description	Units
EC1		Stored energy (state) for cell 1 (hot side)	kwh
EC2		Stored energy (state) for cell 2 (cold side)	kwh
M	2	Outlet mass flow rate (=M1)	lb/hr
MP	2	Maximum discharge rate	kw
TS1,TS2		Average temperatures for cells 1 and 2	°F
T	2	Air exit temperature	°F
MA		Required storage media mass	lb
CCØ		Storage device capital cost/year	$
HF		Convective heat transfer coefficient	kwh/ft^2-°F
U		Unit thermal conductance	kwh/ft^2-°F
P	2	Charge rate into heat exchanger	kw
E1,E2		Energy stored at start and end of melt	kwh
PM		Maximum allowable charge rate	kw
EF	2	Output product efficiency	-
RT		Thermal resistance	°F/kw
PØ		Discharge power taken from heat exchanger	kw
TØ		Discharge cycle output temperature	°F
EFØ		Discharge cycle efficiency	-

Statistics

TSU	Maximum storage temperature	°F
TSL	Minimum storage temperature	°F
ME	Maximum stored energy	kwh
MT	Maximum exit temperature	°F

The default values assume use of paraffin wax as the phase change storage medium. (In reality, paraffin wax may not be applicable to temperatures as high as 600°F. The selection of a phase change medium involves careful consideration of a number of factors [see Reference 1]). The heat exchanger geometric parameters, i.e., tube number, diameter, etc., and heat exchanger cost estimates are based on the baseline phase change storage device developed in Reference 1, but scaled down to reflect expected mass flow rates and required media mass. Although these data were developed for a different application (50 MWe, 6 hour storage, average temperature = 786°C), they can be considered representative until detail design data is available.

[1] "Advanced Thermal Energy Storage," BEC/EPRI RP 788-1, July 1976.

Calculation Sequence

1) Initial Calculations

$$\text{MA} = \frac{\text{PD*ST*0.5}}{\text{XD*H+CP1*DTD}}$$

$$\text{CCØ} = \text{(CSA+CSB*ST)*PD/LE}$$

$$\text{E1} = \text{MA*CP1*(TMT-TØ1)}$$

$$\text{E2} = \text{MA*[H+CP1*(TMT-TØ1)]}$$

$$\text{T3} = \text{TSØ=TA}$$

$$\text{A} = \text{(D*DEL+DEL**2)/5.}$$

$$\text{RB(1)} = \text{D/2,}\quad \text{RB(I+1)} = \text{SQRT(RB(I)**2+A)} \qquad \text{I=1,5}$$

$$\text{RN(I)} = \text{SQRT((RB(I+1)**2+RB(I)**2)/2)}$$

$$\text{RT} = \frac{\text{D}}{\text{2*k}} \sum_{i=1}^{4} \text{LN}\left(\frac{\text{RN(I+1)}}{\text{RN(I)}}\right)$$

2) Storage Temperature (see Figure 7.19-2)

$$\text{TS} = \begin{cases} \text{TØ1} + \dfrac{E}{\text{MA*CP1}} & \text{if } E < E_1 \\ \text{TMT} & \text{if } E_1 \leq E \leq E_2 \\ \text{TØ1} + \dfrac{\left(\dfrac{E}{\text{MA}} - H\right)}{\text{CP1}} & \text{if } E > E_2 \end{cases}$$

where TS = TS1 and E = EC1 for storage cell 1 and similarly for cell 2.

3) HX Exit Temperature Calculations

$$\text{M2} = \text{M1}$$

$$\text{P2} = 0$$

$$\text{PX} = 0$$

3) Cont.

$$T2 = TS2 - (TS1-TS2)/2$$

$$\Delta T = TS1-TS2$$

If $M1 = 0$, GO TO 7)

4) Convective Heat Transfer Coefficient[1]

$$HF = \frac{KF}{D}\left[0.0215 * \left(\frac{M1}{NT} * \frac{4}{MU*PI*D}\right)^{0.8} * \left(\frac{CPF*MU}{KF}\right)^{0.6}\right]$$

5) Thermal Conductance

$$U = \left\{\frac{1}{HF} + RT\right\}^{-1}$$

$$UA = U*PI*D*L*NT/(CPF*M1*2)$$

6) Exit Temperature and Charge Rate (See Equation A2. in HX Appendix)

$$TX = T1 - \Delta T - (1. - EXP(-UA))*(T1 - TS1 - \Delta T/2 - \Delta T/UA)$$

$$T2 = TX - \Delta T - (1.-EXP(-UA))*(TX - (TS1+TS2)/2-\Delta T/UA)$$

$$P2 = M1*CPF*(T1-T2)$$

$$PX = M1*CPF*(T1-TX)$$

7) HY Exit Temperature Calculations

$$TØ = TS1 + \Delta T/2$$

$$PØ = 0.$$

$$PY = 0.$$

If $M3 = 0$ GO TO 11)

[1] Kays, W. M., Convective Heat and Mass Transfer, McGraw Hill, N.Y., 1966, p. 173.

8) Convective Heat Transfer Coefficient

$$\text{HFØ} = \frac{\text{KF}}{\text{D}}\left(.0215*\left(\frac{\text{M3}}{\text{NT}} * \frac{4}{\text{MU*PI*D}}\right)^{0.8} * \left(\frac{\text{CPF*MU}}{\text{KF}}\right)^{0.6}\right)$$

9) Thermal Conductance

$$\text{UØ} = \left(\frac{1}{\text{HFØ}} + \text{RT}\right)^{-1}$$

UA = UØ*PI*D*L*NT/(CPF*M3*2)

10) Exit Temperature and Discharge Rate (See Equation A3. in HX Appendix)

$$\text{TY} = \text{T3} + \Delta\text{T} - (1.-\text{EXP}(-\text{UA}))*(\text{T3} - \text{TS2} + \Delta\text{T}/2 + \Delta\text{T}/\text{UA})$$

$$\text{TØ} = \text{TY} + \Delta\text{T} - (1.-\text{EXP}(-\text{UA}))*(\text{TY} - (\text{TS1} + \text{TS2})/2 + \Delta\text{T}/\text{UA})$$

PØ = M3*CPF*(TØ-T3)

PY = M3*CPF*(TØ-TY)

11) Energy Deposited

$$\dot{\text{E}}\text{C1} = \text{PX} - \text{NU} * \text{EC1} - \text{PY} - \text{BE} * (\text{EC1} - \text{EC2})$$

$$\dot{\text{E}}\text{C2} = (\text{P2}-\text{PX}) - \text{NU} * \text{EC2} - (\text{PØ} - \text{PY}) + \text{BE}* (\text{EC1} - \text{EC2})$$

If T2 ≥ TEM, WRITE DIAGNOSTIC

12) Maximum Allowable Mass Flow Rate

MDM = PD/(CPF*DTD)

13) Maximum Allowable Charge Rate

PM = MDM * CPF * (T1-TA)

14) Charging and Discharging Efficiency

EFF = 1

If T2 > TSØ $\quad \text{EFF} = \dfrac{\text{T2 - TSØ}}{\text{T1 - TA}}$

EFØ = 1

If T3 > TSØ $\quad \text{EFØ} = \dfrac{\text{TØ - TA}}{\text{T3 - TSØ}}$

MP2 = MIN (MP1, PM) * EFF

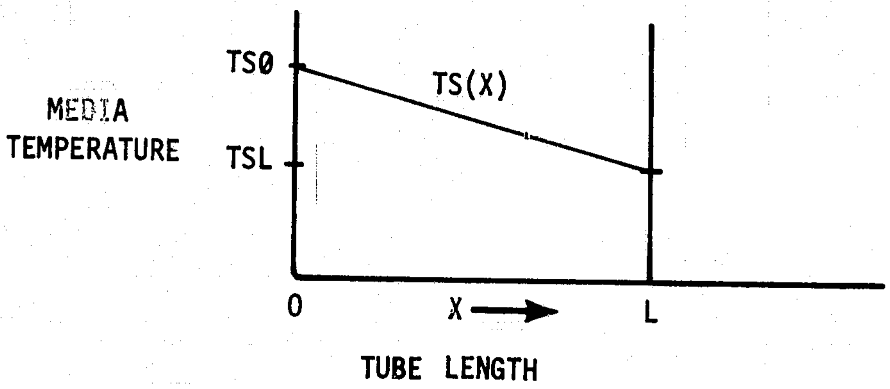
EF2 = EF1*EFF

15) Compute Statistics and Cost Summation

HX Appendix: Temperature Equations for a Media with Constant Gradient

Assumptions

1) Constant Gradient Media Temperature:



2) Working Fluid Differential Equation:

A1. $$\frac{\partial Tf}{\partial X} = \frac{UA}{L}(TS - Tf) \qquad 0 < x < L$$

Main Results: Exit temperature in the charging and discharging cycles are given by

A2. $$Tf(L) = Tf(0) + \Delta TS - (1-\exp(-UA))*\left(Tf(0) - TSØ + \frac{\Delta TS}{UA}\right)$$

A3. $$Tf(0) = Tf(L) - \Delta TS - (1-\exp(-UA))*\left(Tf(L) - TSL - \frac{\Delta TS}{US}\right)$$

where $\Delta TS = TSL - TSØ$.

Proof: Multiplying A1. by exp(UA·X/L) and recombining terms yields:

A4. $$\frac{\partial}{\partial X}\left(\exp(UA*X/L)*Tf\right) = \frac{UA}{L}*\exp(UA*X/L)*TS(X).$$

Integrating A4. and subsituting $TS(X) = TSØ + \frac{\Delta TS}{L} * X$ yields

A5.

$$Tf(X) = \exp(-UA*X/L)*Tf(0) + \frac{UA}{L}\int_{0}^{X} \exp(-UA(x-y)/L)*TS(y)dy$$

$$= \exp(-UA*X/L)Tf(0) + (1-\exp(-UA*X/L))*(TS\O - \Delta TS/UA)$$

$$+ \frac{\Delta TS}{L} * X$$

Recombining terms in A5. and letting X=L yields A2. Equation A3. follows from A2. by symmetry, i.e., substitute in A2:

Tf(0) for Tf(L)

Tf(L) for Tf(0)

TSL for TSØ

TSØ for TSL.

```
CHX
      SUBROUTINE HX(EC1,DE1,IE1,EC2,DE2,IE2,M2,MP2,TS1,TS2,T2,MA,
     1CC,HF,U,P,E1,E2,PM,EF2,PO,TO,EFO,R,TSU,TSL,ME,MT,NU,ST,BE,TO1,DTD
     2          ,PD,TEM,XD,EF1,MP1,CP1,H,TMT,CPF,KF,MU,NT,D,L,DEL,K,T1
     3               ,M1,CM,CSA,CSB,LE,M3,T3,TA,TSO)
C
C     PURPOSE   PERFORMANCE OF ADIABATIC HEAT EXCHANGER DURING CHARGE
C
C               CYCLE
C
C    METHOD     HEAT STORAGE MEDIA ASSUMED TO CONTAIN NO TEMPERATURE
C
C               GRADIENTS. ENERGY DEPOSITED IS A FUNCTION OF TEMPERATURE
C
C               AND THERMAL CONDUCTANCE
C
C    WRITTEN BF F. C. MAHONY                  VERSION 2, JUNE 1977
C
C    CALL SEQUENCE
C         OUTPUTS
C               EC1 - STORED ENERGY (STATE ) FOR STORAGE CELL 1, KWH
C               DE1 - ENERGY RATE FOR EC1, KW
C               IE1 - STATUS INDICATOR FOR EC1
C               EC2 - STORED ENERGY STATE FOR STORAGE CELL 2, KWH
C               DE2 - ENERGY RATE FOR EC2, KW
C               IE2 - STATUS INDICATOR FOR EC2
C               M2  - OUTLET MASS FLOW RATE, LB/HR
C               MP2 - MAXIMUM DISCHARGE RATE ALLOWABLE, KW
C               TS1 - STORAGE TEMPERATURE IN CELL 1, DEG F
C               TS2 - STORAGE TEMPERATURE IN CELL 2, DEG F
C               T2  - AIR EXIT TEMPERATURE, DEG F
C               MA  - REQUIRED STORAGE MEDIA MASS  LB
C               CC  - STORAGE DEVICE CAPITAL COST/YEAR, $
C               HF  - CONVECTIVE HEAT TRANSFER COEFFICIENT, KWH/FT2-F
C               U   - UNIT THERMAL CONDUCTANCE, KWH/FT2-F
C               P   - CHARGE RATE OF HEAT EXCHANGER
C               E1  - ENERGY STORED AT START OF MELT PHASE, KWH
C               E2  - ENERGY STORED AT END OF MELT PHASE, KWH
C               PM  - MAXIMUM ALLOWABLE CHARGE RATE, KW
C               EF2 - OUTPUT PRODUCT EFFICIENCY
C               PO  - DISCHARGE POWER TAKEN FROM HEAT EXCHANGER, KW
C               TO  - DISCHARGE CYCLE OUTPUT TEMPERATURE, DEG F
C               EFO - DISCHARGE CYCLE EFFICIENCY
C               R   - THERMAL RESISTANCE, DEG F/KW
C
C         STATISTICS
C               TSU - MAXIMUM STORAGE TEMPERATURE, DEG F
C               TSL - MINIMUM STORAGE TEMPERATURE, DEG F
C               ME  - MAXIMUM STORED ENERGY, KWH
C               MT  - MAXIMUM EXIT TEMPERATURE, DEG F
C
C         INPUTS
C               NU  - STORAGE ENERGY LOSS COEFFICIENT
C               ST  - RATED STORAGE TIME, HR
C               BE  - STORAGE ENERGY MIXING COEFFICIENT, 1/HR
C               TO1 - MINIMUM ALLOWABLE STORAGE TEMPERATURE, DEG F
C               DTD - MEDIA TEMPERATURE SWING, DEG F
C               PD  - RATED THERMAL STORAGE POWER, KW
```

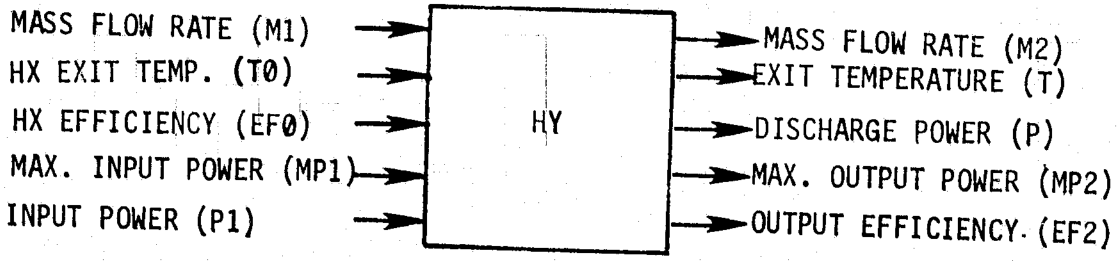
ORIGINAL PAGE IS OF POOR QUALITY

```
C                  TEM - MAXIMUM ALLOWABLE EXIT TEMPERATURE, DEG F
C                  XD  - DESIGN POINT FRACTION OF MOLTEN MEDIA MASS
C                  EF1 - INPUT PRODUCT EFFICIENCY
C                  MP1 - MAXIMUM INPUT CHARGING RATE
C                  CP1 - STORAGE MEDIA HEAT CAPACITY, KWH/LB-F
C                  H   - STORAGE MEDIA HEAT OF FUSION, KWH/LB
C                  TMT - STORAGE MEDIA MELT TEMPERATURE, DEG F
C                  CPF - AIR HEAT CAPACITY, KWH/LB-F
C                  KF  - AIR THERMAL CONDUCTIVITY, KWH/FT-F
C                  MU  - AIR VISCOSITY, LB/FT-HR
C                  NT  - NUMBER OF H/X TUBES
C                  D   - TUBE DIAMETER, FT
C                  L   - TUBE LENGTH, FT
C                  DEL - TUBE HALF SPACING, FT
C                  K   - STORAGE MEDIA THERMAL CONDUCTIVITY
C                  T1  - INLET AIR TEMPERATURE, DEG F
C                  M1  - INLET MASS FLOW RATE, LB/HR
C                  CM  - STORAGE DEVICE YEARLY MAINTENANCE COST $/KW
C                  CSA - STORAGE DEVICE CAPACITY COST, $/KW
C                  CSB - STORAGE DEVICE ENERGY COST, $/KWH
C                  LE  - UNIT LIFE EXPECTANCY, YEARS
C                  M3  - DISCHARGE CYCLE MASS FLOW RATE FROM CS, LB/HR
C                  T3  - DISCHARGE CYCLE TEMPERATURE FROM CS, DEG F
C                  TA  - AMBIENT TEMPERATURE, DEG F
C                  TSO - STORAGE VESSEL MINIMUM TEMPERATURE FROM CS, DEG F
C
      COMMON /CIMPL/IMPL,ICNT/CTIME/TIME /CSIMUL/DUM(7),TMAX
      COMMON /COST/ CCI,CMI
      REAL M3,NU,M2,MP2,MA,ME,MT,MP1,MU,NT,M1,LE,KF,K,L,MDM
      DIMENSION RB(6),RM(5)
      DATA PI/3.14159/
C
      IF(IMPL.GT.0)GO TO 100
C
      IF(NU .EQ. .99999)NU =0.002
      IF(BE .EQ. .99999)BE = 0.0
      IF(TO1.EQ. .99999)TO1=60.0
      IF(DTD.EQ. .99999)DTD=400.0
      IF(TEM.EQ. .99999)TEM=240.0
      IF(CP1.EQ. .99999)CP1=2.93E-4
      IF(H  .EQ. .99999)H  =2.188E-2
      IF(XD .EQ. .99999)XD =0.8
      IF(TMT.EQ. .99999)TMT=147.0
      IF(CPF.EQ. .99999)CPF=7.6E-5
      IF(KF .EQ. .99999)KF =1.03E-4
      IF(MU .EQ. .99999)MU =0.055
      IF(NT .EQ. .99999)NT =200.0
      IF(D  .EQ. .99999)D  =3.0E-2
      IF(L  .EQ. .99999)L  =4.0
      IF(DEL.EQ. .99999)DEL=8.5E-2
      IF(K  .EQ. .99999)K  =7.8E-3
      IF(CM .EQ. .99999)CM =0.6
      IF(CSA.EQ. .99999)CSA=50.0
      IF(CSB.EQ. .99999)CSB=15.6
C
      TSL=1.0E8
      PO=0.0
      PM= 0.0
      TSU=0.0
      ME=0 .0
      MT=0.0
      M3=0.0
      T3=TA
      TSO=TA
      MA =PD*0.5*ST/(XD*H+CP1*DTD)
      CC =(CSA+CSB*ST)*PD/LE
      CM= CM*PD
      E1 =MA*CP1*(TMT-T01)
      E2 =MA*(H+CP1*(TMT-T01))
      TMAX1=TMAX*0.99999
      A  =(D*DEL+DEL**2)/5.0
C
C                     COMPUTE THERMAL RESISTANCE OF MEDIA
C
      RB(1)=D/2.0
C
      DO 20 I=1,5
      RB(I+1)=SQRT(RB(I)**2+A)
   20 RN(I)=SQRT((RB(I+1)**2+RB(I)**2)/2.0)
C
      R=0.0
C
      DO 30 I=1,4
   30 R=R+ALOG(RN(I+1)/RN(I))
C
      R=R*D/2.0/K
C
C                     STORAGE TEMPERATURES
C
  100 TS1=TMT
C
      IF(EC1.LT.E1) TS1= T01+ EC1/(MA*CP1)
      IF(EC1.GT.E2) TS1= T01+ (EC1/MA - H)/CP1
      TS2=TMT
      IF(EC2.LT.E1)TS2= T01+ EC2/(MA*CP1)
      IF(EC2.GT.E2) TS2= T01+ (EC2/MA - H)/CP1
C
C
      DELT= TS1 - TS2
      TSH= TS1+ .5*DELT
      TSC= TS2 - .5*DELT
C
      T2= TSC
      M2=M1
      P =0.0
      PX=0.0
      U=1.0/R
C
      IF(M1.LE..001)GO TO 200
C
C                     CONVECTIVE HEAT TRANSFER COEFFICIENT
C
      HF =KF/D*(0.0215*(M1/NT*4.0/MU/PI/D)**0.8*(CPF*MU/KF)**0.6)
C
C                     UNIT THERMAL CONDUCTANCE
C
```

```
      U = 1.0/(1.0/HF+R)
      UA= U*PI*D*L*NT/(M1*CPF*2.)
      TEMP= DELT/UA
      UA= 1. - EXP(-UA)
C
C                        EXIT TEMPERATURE
C
      TX= T1 - DELT - UA*(T1-TSH-TEMP)
      T2= TX- DELT - UA*(TX-(TS1+TS2)*.5-TEMP)
C
C                        CHARGE RATE
C
      P =M1*CPF*(T1-T2)
      PX = M1*CPF*(T1-TX)
C
C                        HY EXIT TEMPERATURE CALCULATIONS
C
  200 TO = TSH
      PO =0.0
      PY=0.0
      IF(M3.EQ.0.0)GO TO 300
C
C                        CONVECTIVE HEAT TRANSFER COEFFICIENT
C
      HFO =KF/D*(0.0215*(M3/NT*4.0/MU/PI/D)**0.8*(CPF*MU/KF)**0.6)
C
C                        UNIT THERMAL CONDUCTANCE
C
      UO=1.0/(1.0/HFO+R)
      UA= UO*PI*D*L*NT/(M3*CPF*2.)
      TEMP= DELT/UA
      UA= EXP(-UA) - 1.
C
C                        EXIT TEMPERATURE AND DISCHARGE RATE
C
      TY= T3+ DELT+ UA*(T3-TSC+TEMP)
      TO= TY+ DELT + UA*(TY-(TS1+TS2)*.5+TEMP)
      PO =M3*CPF*(TO-T3)
      PY= M3*CPF*(TO-TY)
C
C
C                        ENERGY DEPOSITED
C
  300 IF(IE1.NE.0) DE1= PX- PY -NU*EC1 -BE*(EC1-EC2)
      IF(IE2.NE.0) DE2= P-PX - (PO-PY) -NU*EC2 +BE*(EC1-EC2)
C
      IF (T2.LT.TEM) GO TO 500
C
      IF(IMPL.EQ.2)WRITE(6,1010)T2,TEM
      IF(IMPL.EQ.2)ICNT=ICNT+1
C
C                        MAXIMUM ALLOWABLE CHARGE AND FLOW RATES
C
  500 MDM= PO/(CPF*DTD)
C
      PM =MDM*CPF*(T1-TA)
C
C                        CHARGING AND DISCHARGING EFFICIENCY
```

```
C
      EFF=1.0
C
      IF(T2.GE.TSO)EFF=(T2-TSO)/(T1-TA)
C
      EFO =1.0
C
      IF(T3.GT.TSO)EFO=(TO-TA)/(T3-TSO)
C
      MP2=AMIN1(MP1,PM)*EFF
C
      EF2=EF1*EFF
C
C
C                    STATISTICS
C
      IF(IMPL.LE.1)RETURN
C
      TSU =AMAX1(TSU,TS1)
      TSL =AMIN1(TSL,TS2)
      ME = AMAX1(ME, EC1+EC2)
      MT   =AMAX1(MT  ,T2)
C
      IF(TIME.LT.TMAX1)RETURN
C
      CCI  =CCI+CC
      CMI=CMI+CM
      CM= CM/PD
C
      RETURN
C
 1010 FORMAT(1H0,20HHX EXIT TEMPERATURE ,F12.3
     1          ,35H   GREATER THAN MAXIMUM ALLOWABLE  ,F12.3)
C
      END
```

7.20 ADIABATIC HEAT EXCHANGER - DISCHARGING CYCLE



HY is the discharge cycle complement to HX. All the calculations to obtain the exit temperature and heat exchange power deposited or withdrawn are done in HX. The results are then passed to HY for summary.

HY

Inputs

Parameter/Port		Description	Units
M	1	Air mass flow rate from storage	lb/hr
TØ		Exit temperature from HX	°F
P	1	Discharge power from storage	kw
EFØ		Discharge cycle efficiency from HX	-
MP	1	Maximum power from storage	kw

Outputs

Variable/Port		Description	Units
M	2	Exit air mass flow rate (=M1)	lb/hr
T		Exit temperature (=TØ)	°F
P	2	Discharge power	kw
MP	2	Maximum discharge power	kw
EF	2	Output product efficiency	-

Statistics

TL	Minimum exit temperature	°F
TU	Maximum exit temperature	°F
SP	Total energy discharged	kwh

Calculation Sequence

1) M2 = M1

   T = TØ

   MP2 = MP1*EFØ

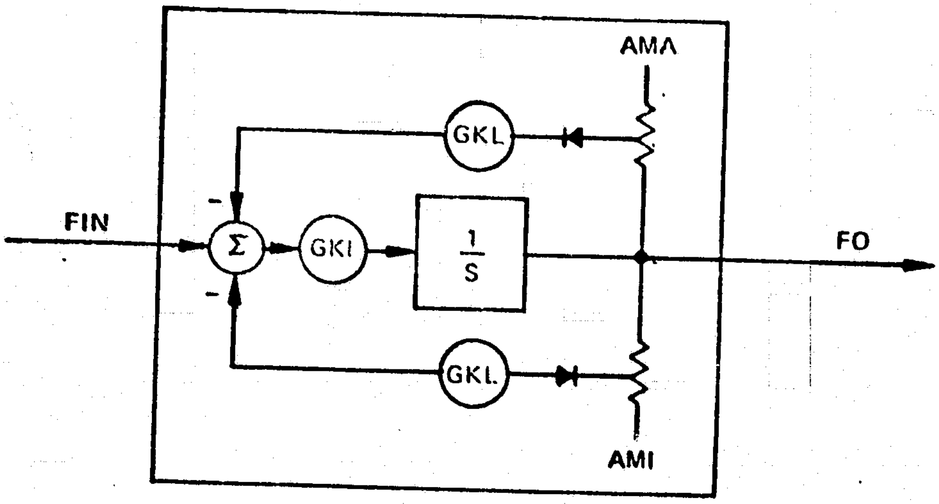
   EF2 = EFØ

   P2 = P1*EFØ

2) Compute Statistics

```
CHY
      SUBROUTINE HY(M2,T,P2,MP2,EF2,TL,TU,SP,M1,TO,P1,EFO,MP1)
C
C     PURPOSE   PERFORMANCE OF ADIABATIC HEAT EXCHANGER DURING DISCHARGE
C
C               CYCLE
C
C     METHOD    COMPUTE EXIT CONDITIONS USING HEAT EXCHANGER STATE
C
C               DETERMINED IN HX
C
C     WRITEN BY F. O. MAHONY                      VERSION 1, MARCH 27 1977
C
C     CALL SEQUENCE
C          OUTPUTS
C               M2  - EXIT AIR MASS FLOW RATE (=M1), LB/HR
C               T   - EXIT TEMPERATURE (=TO), DEG F
C               P2  - TOTAL DISCHARGE POWER, KW
C               MP2 - MAXIMUM DISCHARGE POWER, KW
C               EF2 - OUTPUT PRODUCT EFFICIENCY
C
C          STATISTICS
C               TL   - MINIMUM EXIT TEMPERATURE, DEG F
C               TU   - MAXIMUM EXIT TEMPERATURE, DEG F
C               SP   - TOTAL ENERGY DISCHARGED, KWH
C
C          INPUTS
C               M1   - AIR MASS FLOW RATE FROM STORAGE, LB/HR
C               TO   - EXIT TEMPERATURE FROM HX, DEG F
C               P1   - DISCHARGE POWER FROM STORAGE, KW
C               EFO  - DISCHARGE CYCLE EFFICIENCY
C               MP1  - MAXIMUM POWER FROM STORAGE, KW
C
      COMMON /CIMPL/IMPL /CSIMUL/DUM(6),TINC
C
      REAL M2,MP2,M1,MP1
C
      IF(IMPL.GT.0)GO TO 100
C
      TU =0.0
      SP =0.0
C
      TL =1.0E10
  100 M2 =M1
      T  =TO
      P2 =P1*EFO
      MP2=MP1*EFO
      EF2=EFO
C
      IF(IMPL.LE.1)RETURN
C
      TL =AMIN1(TL ,T )
      TU =AMAX1(TU ,T )
C
      SP =SP +P2*TINC/2.0
C
      RETURN
      END
```

7.21 INTEGRATOR WITH SATURATION



Inputs

Parameter/Port	Description
FIN	Input
GKI	Integration gain
GKL	Saturation limiter gain
AMA	Upper limit of output (Default = 10^{36})
AMI	Lower limit of output (Default = -10^{36})

Outputs

Variable/Port	
FØ	Output (state)

Calculation Sequence

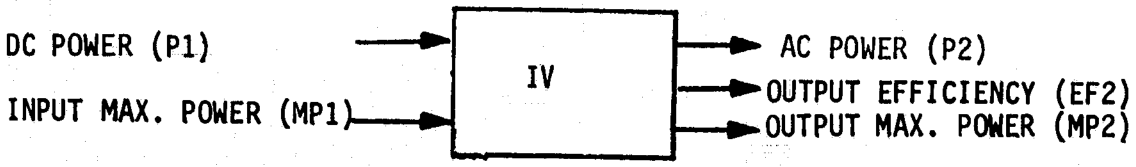
$\dot{F\emptyset} = GKI*[FIN-GKL*(F\emptyset-AMA)]$ if $F\emptyset > AMA$

$\dot{F\emptyset} = GKI * FIN$ if $AMI \leq F\emptyset \leq AMA$

$\dot{F\emptyset} = GKI*[FIN-GKL*(F\emptyset-AMI)]$ if $F\emptyset < AMI$

```
CIT
      SUBROUTINE IT(FO,FODOT,IFO,FIN,GKI,GKL,AMA,AMI)
C   VERSION 2.                                      REVISED: OCT 8 1976
C
C    PURPOSE - SIMULATION OF AN INTEGRATOR WITH SATURATION
C
C
C    METHOD   - SEE CODING
C
C
C    LIMITATIONS - EXCESSIVELY HIGH VALUES OF GKL MAY RESULT IN POOR
C                  STEADY STATE CONVERGENCE
C
C
C    WRITTEN BY  - ADAM LLOYD                    LATEST REVISION - NOV 75
C
C
C    INPUT/OUTPUT LIST
C
C    FO          INTEGRATOR OUTPUT               ANY         OUTPUT  STATE
C    FODOT       OUTPUT DERIVATIVE               ANY         OUTPUT  DERIV
C    IFO         INTEGRATOR CONTROL              ---         PROGRAM VAR
C    FIN         FUNCTION INPUT                  ANY         INPUT   VAR
C    GKI         INTEGRATOR GAIN                 ANY         INPUT   PARAM
C    GKL         DERIVATIVE LIMITER GAIN         ANY         INPUT   PARAM
C    AMA         UPPER LIMIT OF OUTPUT           ANY         INPUT   PARAM
C                WHERE DERIV. LIMITER STARTS
C    AMI         LOWER LIMIT OF OUTPUT           ANY         INPUT   PARAM
C                WHERE DERIV. LIMITER STARTS
      EPS=FIN
C  ------------- PROVIDE DEFAULTS THAT ELLIMINATE SATURATION
      IF(AMA.EQ..99999)AMA=1.E36
      IF(AMI.EQ..99999)AMI=-1.E36
      IF(FO.GT.AMA)EPS = FIN - GKL*(FO-AMA)
      IF(FO.LT.AMI)EPS = FIN - GKL*(FO-AMI)
      IF(IFO.NE.0)FODOT=GKI*EPS
      RETURN
      END
```

7.22 DC-AC INVERTER



This component models a solid state inverter/transformer. Power losses due to resistive heating and contact potential loss are modeled. A step-up transformer may also be needed to boost output voltage up to that of the bus. Default parameter values are based on rated power = 200 kw.

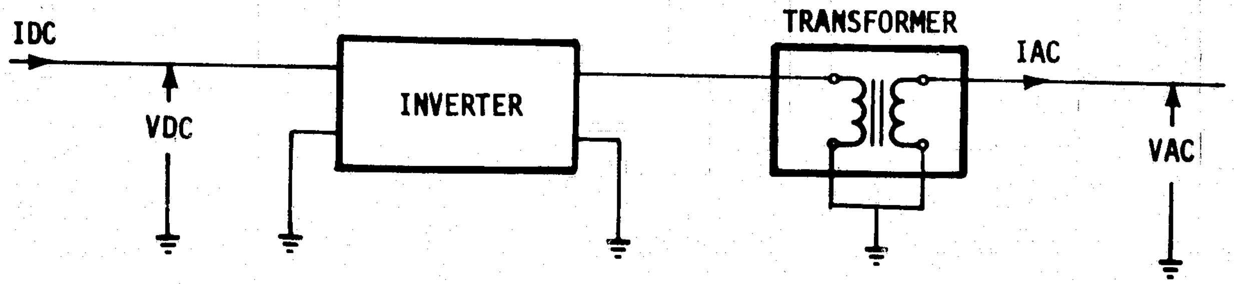


FIGURE 7.22 INVERTER FUNCTIONAL DIAGRAM

IV

Inputs*

Parameter/Port		Description	Units
P	1	DC input power	kw
RT		Transformer resistance (D = 0)	ohms
VDC		Rated DC voltage (D = 100)	volts
DI		Inverter contact potential (D = 0)	volts
RI		Inverter resistance (D = 0.005)	ohms
RAP		Rated input power	kw
EF	1	Input product efficiency	-
MP	1	Maximum input power	kw
CC		Inverter cost/year	$

Outputs

Variable/Port			
P	2	AC output power	kw
IDC		DC input current	amps
PL		Power loss	kw
EF	2	Output product efficiency	-
MP	2	Maximum output power	kw

* Minimum inputs to specify IV are:

RI = inverter resistance,
RAP = rated input power.

RI may be used as an adjustment parameter to obtain a specified efficiency at rated power.

D - Default values supplied.

Calculation Sequence

If $P1 \leq 0$, $P2 = IDC = PL = 0$, $EFF = 1$ and go to 3)

1) Input and output current

$IDC = P1*1000/VDC$

$IAC = \sqrt{6}*IDC/\pi$

2) Power loss and output power

$PL = (IDC*(DI+RI*IDC) + \sqrt{3}*RT*IAC^2)/1000$

$P2 = P1 - PL$

$EFF = P2/P1$

$P2 \leq 0$ ⇨ Diagnostic, $EFF = 1$

3) Efficiency and maximum power

$EF2 = EF1*EFF$

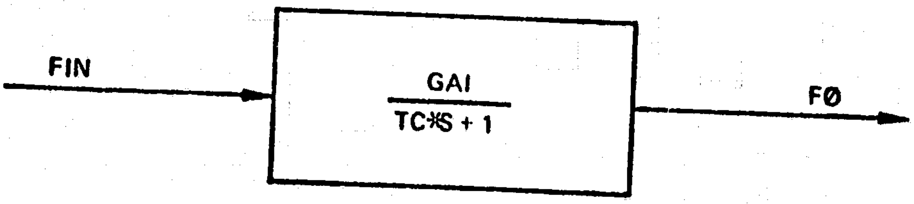
$MP2 = MIN(MP1,RAP)*EFF$

4) Compute Costs

```
CIV
      SUBROUTINE IV(P2,IDC,PL,EF2,MP2,P1,RT,VDC,DI,RI,RAP,EF1,MP1,CC)
C
C                PURPOSE    SOLID STATE INVERTER/TRANSFORMER MODEL
C
C                METHOD     COMPUTE AC POWER AS A FUNCTION OF
C                           INPUT DC POWER
C
C                WRITTEN BY  Y.K.CHAN      VERSION 1, JUNE 2, L977
C
C   CALL SEQUENCE
C                OUTPUTS
C                P2  -AC OUTPUT POWER, KW
C                IDC -DC INPUT CURRENT, AMPS
C                PL  -POWER LOSS, KW
C                EF2 -OUTPUT POWER EFFICIENCY
C                MP2 -MAXIMUM OUTPUT POWER, KW
C           INPUTS
C                P1  -DC INPUT POWER, KW
C                RT  -TRANSFORMER RESISTANCE, OHMS
C                VDC -RATED DC VOLTAGE, VOLTS
C                DI  -INVERTER CONTACT POTENTIAL, VOLTS
C                RI  -INVERTER RESISTANCE, OHMS
C                RAP -RATED OUTPUT POWER, KW
C                EF1 -INPUT PRODUCT EFFICIENCY
C                MP1 -MAXIMUM INPUT POWER, KW
C                CC  -INVERTER COST/YEAR
C
      COMMON /CIMPL/IMPL,ICNT/CTIME/TIME/CSIMUL/DUM(7),TMAX/COST/CCI
      REAL IDC,MP2,MP1,IAC
      DATA PI/3.14159/
C
      IF(IMPL.GT.0) GO TO 100
      IF(RT.EQ..99999)RT=0.
      IF(VDC.EQ..99999)VDC=100.
      IF(DI.EQ..99999)DI=0.
      IF(RI.EQ..99999)RI=.005
      TMAX1=TMAX*.99999
C
C                COMPUTE INPUT AND OUTPUT CURRENT
C
  100 IF(P1.GT.0.)GO TO 200
       P2=0.
       IDC=0.
       PL=0.
       EF2=EF1
       MP2=AMIN1(MP1,RAP)
       GO TO 400
  200 IDC=P1*1000./VDC
      IAC=SQRT(6.)*IDC/PI
C
C                POWER LOSS AND OUTPUT POWER
C
      PL=(IDC*(DI+RI*IDC)+SQRT(3.)*RT*IAC*IAC)/1000.
      P2=P1-PL
      EFF=P2/P1
C
C                EFFICIENCY AND MAXIMUM POWER
```

```
C
      EF2=EF1*EFF
      MP2=AMIN1(MP1,RAP)
      MP2=MP2*EFF
      IF(P2.GT.0.)GO TO 400
C
      EF2=EF1
      MP2=AMIN1(MP1,RAP)
      IF(IMPL.EQ.2)WRITE(6,208)PL,P1
  208 FORMAT(1H0,14HIV POWER LOSS ,F12.3,21H EXCEEDS INPUT POWER ,F12.3,
     128H CHECK RATED DC VOLTAGE VDC  )
      IF(IMPL.EQ.2)ICNT=ICNT+1
      P2=0.
C
  400 IF(IMPL.LE.1)RETURN
      IF(TIME.LT.TMAX1)RETURN
      CCI=CCI+CC
      RETURN
      END
```

7.23 FIRST ORDER LAG



Inputs

Parameter/Port	Description
FIN	Input quantity
GAI	Gain
TC	Time constant[1] (hours)
FØ	Output variable (state)

Calculation Sequence

$$\dot{\text{FØ}} = (\text{GAI} * \text{FIN} - \text{FØ})/\text{TC}$$

NOTE: d.c. gain = GAI; time constant = TC

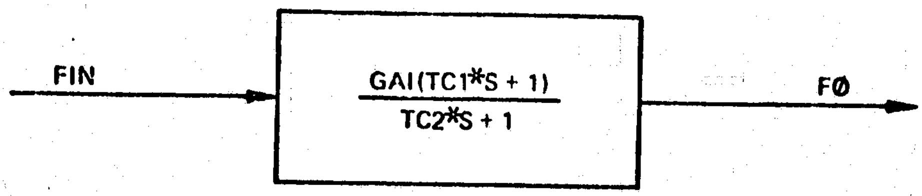
infinite frequency gain = 0

pole location = $\frac{-1}{\text{TC}}$ rad/sec.

[1] If TC = 0, then FØ = FIN*GAI

```
CLA
      SUBROUTINE LA(FO,FODOT,IFO,FIN,GAI,TC)
C
C    PURPOSE - TO SIMULATE FIRST ORDER LAG      FO       GAI
C                                              ---- = ----------
C                                              FIN    (1.+TC*S)
C
C
C    METHOD  - SEE CODING
C
C
C    WRITTEN BY  -   ADAM LLOYD         LATEST REVISION     NOV 75
C
C
C    INPUT/OUTPUT LIST
C
C    FO         TRANSFER FUNCTION OUTPUT              ANY      OUTPUT STATE
C    FODOT      TRANSFER FUNCTION OUTPUT DERIV.       ANY      OUTPUT STATE
C    IFO        INTEGERATOR CONTROL                   ---      PROGRAM VAR
C    FIN        TRANSFER FUNCTION INPUT               ANY      INPUT   VAR
C    GAI        TRANSFER FUNCTION GAIN                ---      INPUT   PARAM
C    TC         TIME CONSTANT                         SECS     INPUT   PARAM
      COMMON/CIO/IREAD,IWRITE,IDIAG
      IF(TC.NE.0.) GO TO 10
      FO= GAI*FIN
      RETURN
   10 IF(IFO.NE.0) FODOT=(GAI*FIN-FO)/TC
      RETURN
      END
```

7.24 LEAD LAG



Inputs

Parameter/Port	Description
FIN	Input quantity
TC1	Numerator time constant (hours)
TC2	Denominator time constant (hours)
GAI	Gain

Outputs

Variable/Port	
X1	Intermediate quantity (state)
FØ	Output quantity (variable)

Calculation Sequence

$$FØ = (X1 + FIN*TC1*GAI)/TC2$$

$$\dot{X1} = GAI*FIN-FØ$$

NOTE: d.c. gain = GAI

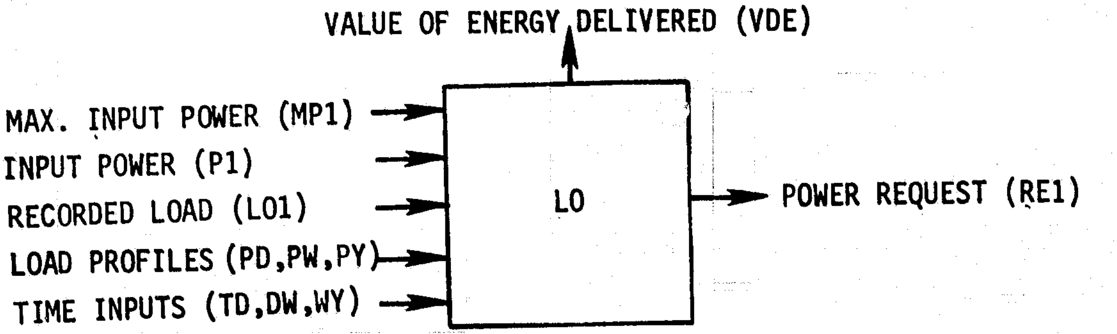
$$\text{infinite gain} = \frac{GAI*TC1}{TC2}$$

$$\text{zero location} = -\frac{1}{TC1}\text{, rad/sec}$$

$$\text{pole location} = -\frac{1}{TC2}\text{, rad/sec}$$

```
CLL
      SUBROUTINE LL(X1,X1DOT,IX1,FO,FIN,TC1,TC2,GAI)
C
C     PURPOSE - TO SIMULATE LEAD LAG TRANSFER FUNCTION
C
C                                   FO      GAI*(1.+TC1*S)
C                                 ----- = ---------------
C                                  FIN       (1.+TC2*S)
C
C
C     METHOD   - SELF EXPLANATORY
C
C
C     LIMITATIONS -    NONE
C
C
C     WRITTEN BY   -    ADAM LLOYD            LATEST REVISION  NOV 75
C
C
C     INPUT/OUTPUT LIST
C
C     X1          STATE VARIABLE                        ANY        OUTPUT STATE
C     X1DOT       STATE VARIABLE DERIVATIVE             ANY        OUTPUT STATE
C     IX1         INTEGRATOR CONTROL                    ----       PROGRAM VAR
C     FO          TRANSFER FUNCTION OUTPUT              ANY        OUTPUT  VAR
C     FIN         TRANSFER FUNCTION INPUT               ANY        INPUT   VAR
C     TC1         TIME CONSTANT (NUMERATOR)             SECS       INPUT   PARAM
C     TC2         TIME CONSTANT (DENOMINATOR)           SECS       INPUT   PARAM
C     GAI         TRANSFER FUNCTION GAIN                ----       INPUT   PARAM
      COMMON/CIO/IREAD,IWRITE,IDIAG
      FO=(X1+FIN*TC1*GAI)/TC2
      IF(IX1.NE.0)X1DOT= GAI*FIN-FO
      RETURN
      END
```

7.25 ELECTRICAL LOAD



This component represents electrical load either by a user-specified data file time history or by a set of random numbers with user-specified daily, weekly, and yearly average profiles and user-specified random variation. It also computes the value of the power delivered to the load by the system. This value delivered is determined from a user-specified value per kwh. This value may be input in tabular form as a function of time of day, time of year, or any other system parameter.

If the user selects to have the electrical load represented by random numbers, then the load (LO2) is generated from the following equation:

Basic Equation

$$\text{LØ2} = \left[PD(TD) + CN(t) \right] * PW(DW)*PY(WY)*NC$$

where

PD, PW, PY are the daily, weekly, and yearly profiles, respectively, and TD, DW, WY are the time of day, day of the week, and week of the year, respectively. NC is a normalizing constant.

CN is a colored noise term with user-specified correlation time, standard deviation and mean.

LO

Tables		Description	Units
PD		Daily profile (tabular with TD)	kw
PW		Weekly profile (tabular with DW)	arbitrary
PY		Yearly profile (tabular with WY)	arbitrary
Inputs			
Parameter/Port			
P	1	Power delivered	kw
MP	1	Maximum Input Power deliverable (D = $1X10^{10}$)	kw
NC		Normalizing constant	
VE		Value of Electrical Energy	$/kwh
LØ	1	Electrical load data file input	kw
TD		Time of day	-
DW		Day of week	-
WY		Week of year	-
CT		Correlation time of random noise	hr
MN,STD		Mean (D = 0) and std. deviation of random noise	kw
EF	1	Input Power Efficiency	-
Outputs			
Variable/Port			
RE	1	Power request	kw
VDE		Value of energy delivered (state)	$
LØ	2	Electrical load	kw
TIM		Last time a random sample was used	hr
CN		Colored noise sample	kw
Statistics			
SRE		Total energy requested	kwh
SDE		Total energy delivered	kwh
PC		Percentage of load met	-

D - Default values supplied

Calculation Sequence

1) Initialize CN(0) (first pass)

2) Check for data file input

If LØ1 = .99999 go to 3)

LØ2 = LØ1 and go to 5)

3) Generate colored noise CN

If TIM = TIME go to 5)

$$A = \begin{pmatrix} \exp(-\Delta/CT), & CT > 0, \ \Delta = \text{integration step size, hr} \\ 0. & CT=0 \end{pmatrix}$$

CN = A*CN+W,

Where W is white noise generated by RN with

Mean = MN * (1-A) and standard deviation = STD * $\sqrt{1-A^2}$

4) Compute LØ2

LØ2 = (PD(TD) + CN) * PW(DW) * PY (WY) * NC

TIM = TIME

5) Power request and value delivered

RE = MIN(MP,LØ2)/EF1

6) Statistics

$\dot{\text{V}}$DE = P1*VE

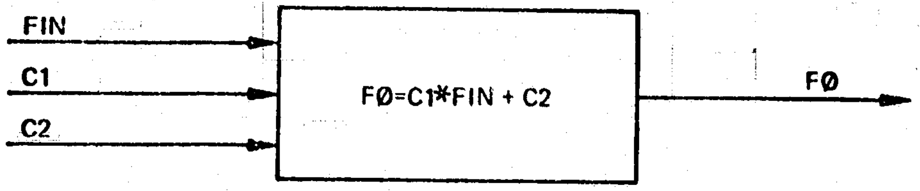
SRE = SRE + LØ2* Δ/2

SDE = SDE + P1* Δ/2

PC = 100.* SDE/SRE

```
CLO
      SUBROUTINE LO (PD,PW,PY,VDE,DVD,IVD,RE,LO2,SRE,SDE,PC,TIMO,XN,
     1 TD,DW,WY,XNC,CT,XMN,STD,VE,LO1,PMAX,PO,EF)
C
C     PURPOSE     GENERATE ELECTRICAL LOAD FROM DAILY, WEEKLY, YEARLY AND
C                 RANDOM PROFILE DATA AND EVALUATE PERFORMANCE STATISTICS
C
C     METHOD      COLORED NOISE IS ADDED TO A MEAN DAILY PROFILE AND MULTIPLIED
C                 BY WEEKLY AND YEARLY WEIGHTING FCNS. POWER REQUESTED IS EITHEF
C                 THE GENERATED LOAD OR THE MAX. POWER DELIVERABLE.
C
C     WRITTEN BY A.W.WARREN
C                                                   VERSION 1, MARCH 9 197
C     CALL SEQUENCE
C          TABLES
C                PD   - MEAN DAILY PROFILE, KW
C                PW   - MEAN WEEKLY PROFILE, -
C                PY   - MEAN YEARLY PROFILE, -
C
C          OUTPUTS
C                VDE  - VALUE OF ENERGY DELIVERED (STATE), $
C                DVD  - DERIVATIVE OF VDE
C                IVD  - INDICATOR FOR VDE
C                RE   - POWER REQUEST, KW
C                LO2  - ELECTRICAL LOAD DEMAND, KW
C                SRE  - SUM OF ENERGY DESIRED, KWH
C                SDE  - SUM OF ENERGY DELIVERED, KWH
C                PC   - CUMULATIVE PERCENT OF LOAD DELIVERED, -
C                TIMC- LAST TIME A RANDOM SAMPLE WAS USED, HR
C                XN   - COLORED NOISE SAMPLE, KW
C
C          INPUTS
C                TD   - TIME OF DAY, HR
C                DW   - DAY OF WEEK (1-7)
C                WY   - WEEK OF YEAR (1-52)
C                XNC  - NORMALIZING CONSTANT, -
C                CT   - CORRELATION TIME OF RANDOM NOISE, HR
C                XMN  - MEAN OF RANDOM NOISE, KW
C                STD  - STANDARD DEVIATION OF RANDOM NOISE, KW
C                VE   - VALUE OF ELECTRICAL ENERGY, $/KWH
C                LO1  - ELECTRICAL LOAD DATA FILE INPUT, XW
C                PMAX- MAX. INPUT POWER DELIVERABLE, KW
C                PO   - POWER DELIVERED TO LOAD, KW
C                EF   - INPUT POWER EFFICIENCY
C
      DIMENSION PD(1),PW(1),PY(1)
      REAL LO1,LO2
      COMMON /CIMPL/ IMPL /CSIMUL/ DUM(6),TINC,TMAX/CTIME/TIME
      COMMON /COST/CC,CM,CO,CV,CDE,CRE
      DATA AX/.99999/
C
C                          INITIALIZATION
C
      ND = PD(2)
      NW = PW(2)
      NY = PY(2)
      IF(IMPL.GT.0) GO TO 10
      IF(XMN.EQ. .99999)XMN=0.
      TMAX1 = TMAX*.99999
      TIMO=-1.
      SRE =0.0
      SDE =0.0
      PC  =0.0
      CALL RN(XN,AX,STD,XMN)
      IF(PMAX.EQ. .99999) PMAX = 1.E10
C
C                            CHECK FOR DATA FILE INPUT
C
   10 IF(LO1.EQ. .99999) GO TO 100
      LO2 = LO1
      GO TO 150
C
C                            GENERATE COLORED NOISE SAMPLE XN
C
  100 IF(  TIMO.EQ.TIME) GO TO 150
      A=0.
      IF(CT.GT.0.) A = EXP(-TINC/CT)
      WMN  = XMN*(1.-A)
      WSD  = STD*SQRT(1.-A*A)
      CALL RN(W,AX,WSD,WMN)
      XN = XN*A + W
C                            COMPUTE ELECTRICAL LOAD DEMAND
C
      DLO  = TBLU1(TD,PD(4),PD(ND+4),1,-ND)
      WLO  = TBLU1(DW,PW(4),PW(NW+4),1,-NW)
      YLO  = TBLU1(WY,PY(4),PY(NY+4),1,-NY)
      LO2 = (DLO+XN)*WLO * YLO*XNC
      TIMO = TIME
  150 RE  = AMIN1(PMAX,LO2)/EF
C                            PERFORMANCE STATISTICS
C
      IF(IMPL.LE.1) RETURN
      IF(IVD.NE. 0) DVD = PD*VE
      SRE  = SRE + LO2*0.5*TINC
      SDE  = SDE + PD*0.5*TINC
      IF(SRE.GT.0.) PC  = 100.*SDE/SRE
C
      IF(TIME.LT.TMAX1) RETURN
      CV = CV + VDE
      CDE= CDE + SDE- PD*0.5*TINC
      CRE= CRE + SRE- LO2*0.5*TINC
      RETURN
      END
```

7.26 MULTIPLY AND ADD



Inputs

Parameter/Port	Description
FIN	Input quantity
C1	Input quantity
C2	Input quantity

Outputs

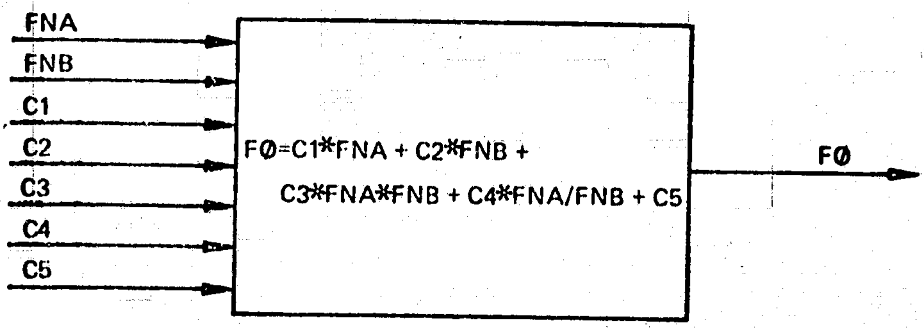
Variable/Port	
FØ	Output quantity

Calculation Sequence

$$FØ = C1*FIN + C2$$

```
CMA
      SUBROUTINE MA(FO,FIN,C1,C2)
C
C    PURPOSE - TO SIMULATE THE EQUATION    OUTPUT=C1*INPUT + C2
C
C
C    METHOD   - SEE CODING
C
C
C    WRITTEN BY  -   ADAM LLOYD           LATEST REVISION     NOV 75
C
C
C    LIMITATIONS - NONE
C
C
C    INPUT/OUTPUT LIST
C
C    FO          OUTPUT  VARIABLE              ANY          OUTPUT VAR
C    FIN         INPUT   VARIABLE              ANY          INPUT  VAR
C    C1          CONSTANT MULTIPLIER           ---          INPUT  PARAM
C    C2          CONSTANT ADDITION             ---          INPUT  PARAM
      FO=C1*FIN + C2
      RETURN
      END
```

7.27 MULTIPLY, DIVIDE, AND ADD



Inputs

Parameter/Port	Description
FNA	Input quantity
FNB	Input quantity
C1	Input quantity
C2	Input quantity
C3	Input quantity
C4	Input quantity
C5	Input quantity

Outputs

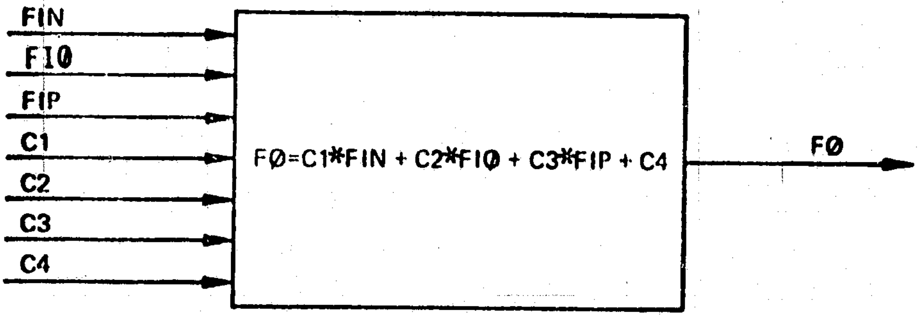
Variable/Port	
FØ	Output quantity

Calculation Sequence

FØ = C1*FNA+C2*FNB+C3*FNA*FNB+C4*FNA/FNB+C5

```
CMB
      SUBROUTINE MB(FO,FNA,FNB,C1,C2,C3,C4,C5)
C
C   PURPOSE - TO SIMULATE THE EQUATION   Y=C1*XA+C2*XB+C3*XA*XB+C4*XA/XB+C
C
C
C   WRITTEN BY - GEORGE DULEBA                    LATEST REVISION   MAY 76
C
C
C   LIMITATIONS - IF FNB=0 DURING DIVISION, FNB IS SET TO E-20.
C                 DIAGNOSTIC MESSAGE IS GIVEN.
C
C
C   INPUT/OUTPUT LIST
C
C   FO          OUTPUT VARIABLE                  ANY        OUTPUT \
C   FNA         INPUT VARIABLE A                 ANY        INPUT  \
C   FNB         INPUT VARIABLE B                 ANY        INPUT  \
C   C1          MULTIPLIER 1                     ANY        INPUT  \
C   C2          MULTIPLIER 2                     ANY        INPUT  \
C   C3          MULTIPLIER 3                     ANY        INPUT  \
C   C4          MULTIPLIER 4                     ANY        INPUT  \
C   C5          ADDITIVE VARIABLE                ANY        INPUT  \
C
C
      COMMON/ERMESS/IFATAL,IERR
      COMMON/CIO/IREAD,IWRITE,IDIAG
      FO= C1*FNA + C2*FNB + C3*FNA*FNB + C5
      IF(C4.EQ.0.99999) GO TO 30
      IF(FNB.EQ.0.) GO TO 10
      FO= FO + C4*FNA/FNB
      RETURN
 10   WRITE(IWRITE,20)
 20   FORMAT(/,30X, 53HWARNING- DIVISOR IN MB EQUALS O., HAS BEEN SET=1.
     2E-20)
      FO= FO + C4*FNA*1.E+20
 30   RETURN
      END
```

7.28 MULTIPLY AND ADD



Inputs

Parameter/Port	Description
FIN	Input quantity
FI0	Input quantity
FIP	Input quantity
C1	Input quantity
C2	Input quantity
C3	Input quantity
C4	Input quantity

Outputs

Variable/Port	
FØ	Output quantity

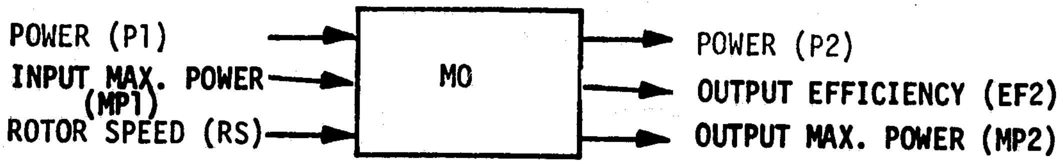
Calculation Sequence

FØ = C1*FIN + C2*FI0 + C3*FIP + C4

MC

```
CMC
      SUBROUTINE MC(FO,FIN,FIO,FIP,C1,C2,C3,C4)
C
C    PURPOSE - TO SIMULATE THE EQUATION  FO=C1*FIN+C2*FIO+C3*FIP+C4
C
C
C    METHOD  - SEE CODING
C
C
C    WRITTEN BY  -  ADAM LLOYD          LATEST REVISION   NOV 75
C
C
C    LIMITATIONS - NONE
C
C
C    INPUT/OUTPUT LIST
C
C    FO          OUTPUT VARIABLE              ANY           OUTPUT VAR
C    FIN         INPUT  VARIABLE              ANY           INPUT  VAR
C    FIO         INPUT  VARIABLE              ANY           INPUT  VAR
C    FIP         INPUT  VARIABLE              ANY           INPUT  VAR
C    C1          CONSTANT MULTIPLIER          ---           INPUT  PARAM
C    C2          CONSTANT MULTIPLIER          ---           INPUT  PARAM
C    C3          CONSTANT MULTIPLIER          ---           INPUT  PARAM
C    C4          CONSTANT ADDITION            ---           INPUT  PARAM
      FO=C1*FIN+C2*FIO+C3*FIP+C4
      RETURN
      END
```

7.29 AC INDUCTION MOTOR



The induction motor produces mechanical power and torque proportional to slip speed, i.e. power and torque approach zero as the rotor approaches synchronous speed. Two power losses are modeled: a constant multiplicative term due to resistive heating and an additive term due to mechanical friction. Default parameters are based on a conventional squirrel-cage induction motor/generator machine.

Basic Equations

$$P2 = EE*P1 + DA*RS^2*C$$

where

$P1, P2$ = input and output power

EE = electrical efficiency

DA = mechanical damping

C = conversion constant

MO

Inputs

Parameter/Port		Description	Units
P	1	Input power	kw
DA		Mechanical damping (D = 0)	joule-sec
RS		Rotor speed	rpm
RSY		Synchronous rotor speed (D = 1800)	rpm
SR		Stator resistance (D = 8/RAP)	ohms
VØ		Rated input voltage (D = 400)	volts
RAP		Rated input power	kw
RAS		Rated power slip (D = 0.05)	-
EF	1	Input product efficiency	-
MP	1	Maximum input power (D = 1x10^8)	kw
CC		Capital cost/year	$
CM		Maintenance cost/year	$

Outputs

Variable/Port			
P	2	Output mechanical power	kw
EE		Electrical efficiency	-
TØ		Mechanical torque	ft-lb
PL		Power loss	kw
EF	2	Output product efficiency	-
MP	2	Output maximum power	kw

Statistics

MT	Maximum torque	ft-lb
MPN	Maximum output power/rated power	-
SP	Output energy sum	kwh

D - Default values supplied.

Calculation Sequence

1) Compute electrical efficiency (first pass only)

$$I_{RAT} = RAP*1000/VO$$

$$EE = 1 - SR*I^{2}_{RAT}/RAP*1000$$

2) Diagnostics

$P1 > RAP$ ⇨ DIAGNOSTIC

$SLIP = 1 - RS/RSY > RAS$ ⇨ DIAGNOSTIC

3) Output power and power loss

$$\omega = RS*(2\pi/60)$$

$$P2 = EE*P1 - DA* \omega^{2}/1000$$

$$PL = P1 - P2$$

4) If $P2 > 0$ go to 5)

$P1 > 0$ ⇨ DIAGNOSTIC

$EF2 = EF1,\ MP2 = MIN(MP1,RAP)$

Go to 7)

5) Compute torque

$$\text{TØ} = P2*1000/\omega*k$$

$$k = 1.3558 \text{ joules/ft-lb}$$

Calculation Sequence Cont.

6) Efficiency and maximum output power

$$EF2 = EF1*(P2/P1)$$

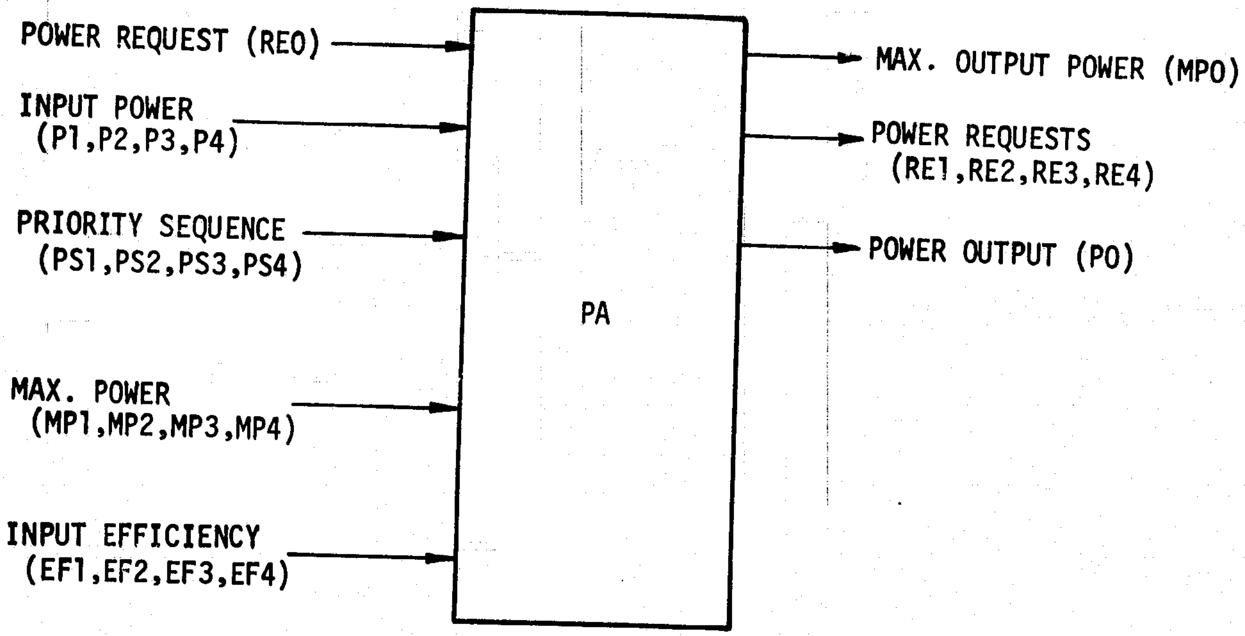
$$MP2 = MIN(MP1,RAP)*(P2/P1)$$

7) Compute Statistics and Costs

```
CMO
      SUBROUTINE MO(P2,EE,TO,PL,EF2,MP2,MT,MPN,SP,
     1              P1,DA,RS,RSY,SR,VO,RAP,RAS,EF1,MP1,CC,CM)
C
C               PURPOSE   AC INDUCTION MOTOR MODEL
C
C               METHOD    MECHANICAL POWER AND TORQUE CALCULATED
C                         FROM INPUT AC POWER AND ROTOR SPEED
C
C               WRITTEN BY Y.K.CHAN        VERSION 1, JUNE 13, 1977
C
C    CALL SEQUENCE
C          OUTPUTS
C               P2   -OUTPUT MECHANICAL POWER,KW
C               EE   -ELECTRICAL EFFICIENCY
C               TO   -MECHANICAL TORQUE,FT-LB
C               PL   -POWER LOSS,KW
C               EF2  -OUTPUT POWER EFFICIENCY
C               MP2  -OUTPUT MAXIMUM POWER,KW
C          STATISTICS
C               MT   -MAXIMUM TORQUE,FT-LB
C               MPN  -MAXIMUM OUTPUT POWER/RATED POWER
C               SP-OUTPUT POWER SUM
C          INPUTS
C               P1   -INPUT POWER,KW
C               DA   -MECHANICAL DAMPING,JOULE-SEC
C               RS   -ROTOR SPEED,RPM
C               RSY  -SYNCHRONOUS ROTOR SPEED,RPM
C               SR   -STATOR RESISTANCE,OHMS
C               VO   -RATED INPUT VOLTAGE,VOLTS
C               RAP  -RATED INPUT POWER,KW
C               RAS  -RATED PWER SLIP
C               EF1  -INPUT PRODUCT EFFICIENCY
C               MP1  -MAXIMUM INPUT POWE,KW
C               CC   -CAPITAL COST/YEAR,$
C               CM   -MAINTENANCE COST/YEAR,$
C
      COMMON /CIMPL/IMPL,ICWT/CTIME/TIME/CSIMUL/DUM(7),TMAX
     X       /COST/CCI,CMI,COP,VDE,TDE,TLD,UTV,UTD
      REAL MP2,MT,MPN,MP1
C
      IF(IMPL.GT.0)GO TO 100
      IF(DA.EQ..99999)DA=0.
      IF(RSY.EQ..99999)RSY=1800.
      IF(SR.EQ..99999)SR=8./RAP
      IF(VO.EQ..99999)VO=400.
      IF(MP1.EQ..99999)MP1=1.E8
      IF(RAS.EQ..99999)RAS=.05
      TMAX1=TMAX*.99999
      MT=0.
      MPN=0.
      SP=0.
      TINC=DUM(7)*.5
C
C              COMPUTE ELECTRICAL EFFICIENCY
C
      EE=1.-SR*RAP*1000./(VO*VO)
  100 IF(P1.LE.RAP*1.001)GO TO 200
```

```
      IF(IMPL.EQ.2)WRITE(6,208)P1,RAP
  208 FORMAT(1H0,18H MOTOR INPUT POWER,F12.3,23H .GT.RATED INPUT POWER ,
     1  F12.3)
      IF(IMPL.EQ.2)ICNT=ICNT+1
  200 SLIP=1.-(RS/RSY)
      IF(SLIP.LE.RAS)GO TO 300
      IF(IMPL.EQ.2)WRITE(6,308)SLIP,RAS
  308 FORMAT(1H0,11H MOTOR SLIP,F12.3,25H EXCEEDS RATED POWER SLIP,
     1  F12.3)
      IF(IMPL.EQ.2)ICNT=ICNT+1
C
C                   COMPUTE POWER AND POWER LOSS
C
  300 OMEGA=RS*3.14159/30.
      P2=EE*P1-DA*OMEGA*OMEGA/1000.
      PL=P1-P2
      TO=0.
      IF(P2.GT.0.)GO TO 400
      IF(P1.LE.0.)GO TO 409
      IF(IMPL.EQ.2)WRITE(6,408)SR,DA
  408 FORMAT(1H0,19H STATOR RESISTANCE  ,F12.3,12H OR DAMPING ,
     XF12.3,20H TOO HIGH FOR MOTOR   )
      IF(IMPL.EQ.2)ICNT=ICNT+1
C
C                   EFFICIENCY AND MAXIMUM OUTPUT POWER
C
  409 CONTINUE
      P2=0.
      EF2=EF1
      MP2=AMIN1(MP1,RAP)
      GO TO 500
  400 EF2=EF1*P2/P1
      MP2=AMIN1(MP1,RAP)*P2/P1
      IF(RS.NE.0.)TO=P2*737.6/OMEGA
C
  500 IF(IMPL.LE.1)RETURN
C
C                   STATISTICS
C
      MT=AMAX1(TO,MT)
      MPN=AMAX1(P2/RAP,MPN)
      SP=SP+P2*TINC
C
      IF(TIME.LT.TMAX1)RETURN
      CCI=CCI+CC
      CMI=CMI+CM
C
      RETURN
      END
```

7.30 POWER ACCUMULATOR


POWER REQUEST (REO)
INPUT POWER (P1,P2,P3,P4)
PRIORITY SEQUENCE (PS1,PS2,PS3,PS4)
MAX. POWER (MP1,MP2,MP3,MP4)
INPUT EFFICIENCY (EF1,EF2,EF3,EF4)
PA
MAX. OUTPUT POWER (MPO)
POWER REQUESTS (RE1,RE2,RE3,RE4)
POWER OUTPUT (PO)

This component sums power from four input ports and allocates power requests to each port's source of power generation. An input power request is allocated according to user-supplied weights within the ports of highest priority. If an input power request (load) exceeds the maximum power that can be delivered by the ports of highest priority, then the remaining load is allocated to the next priority ports. (See 1.2.2 and 7c for further discussion.)

Inputs[1]

Parameter/Port		Description	Units
RE	0	Load request	kw
EF	1,2,3,4	Input efficiency from port i	-
P	1,2,3,4	Input power from port i (default = 0.)	kw
PS	1,2,3,4	Priority sequence (default = 1,2,3,4)	-
F	1,2,3,4	Allocation weight (for equal priorities)	-
MP	1,2,3,4	Maximum power (default = 0.)	kw

Outputs

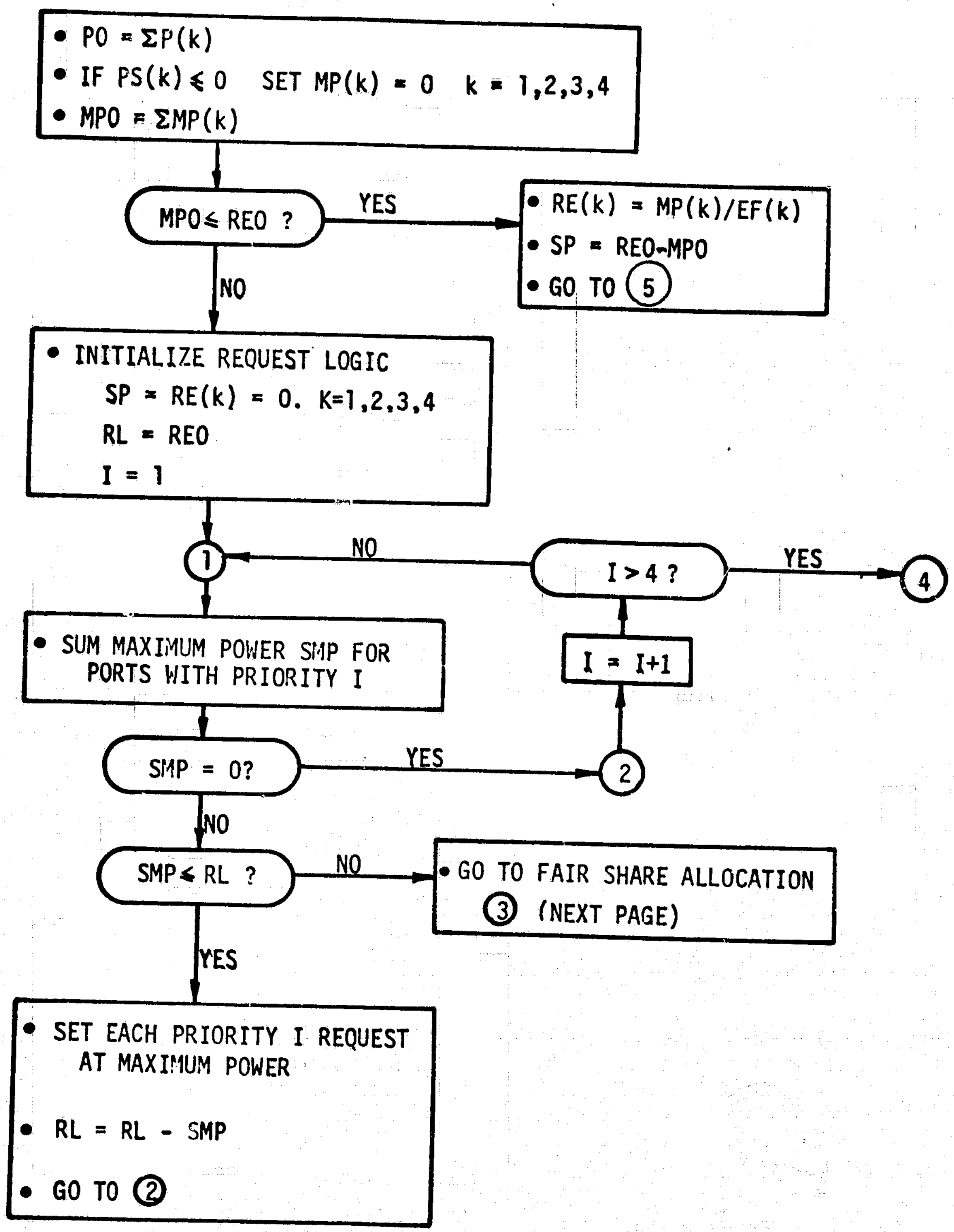
Variable/Port			
MP	0	Maximum deliverable power (Σ MP(i))	kw
RE	1,2,3,4	Power request for port i	kw
P	0	Power output	kw
SP		Supplemental power request to meet load (Power deficit = $RE_0 - \Sigma MP_i$)	kw

Statistics

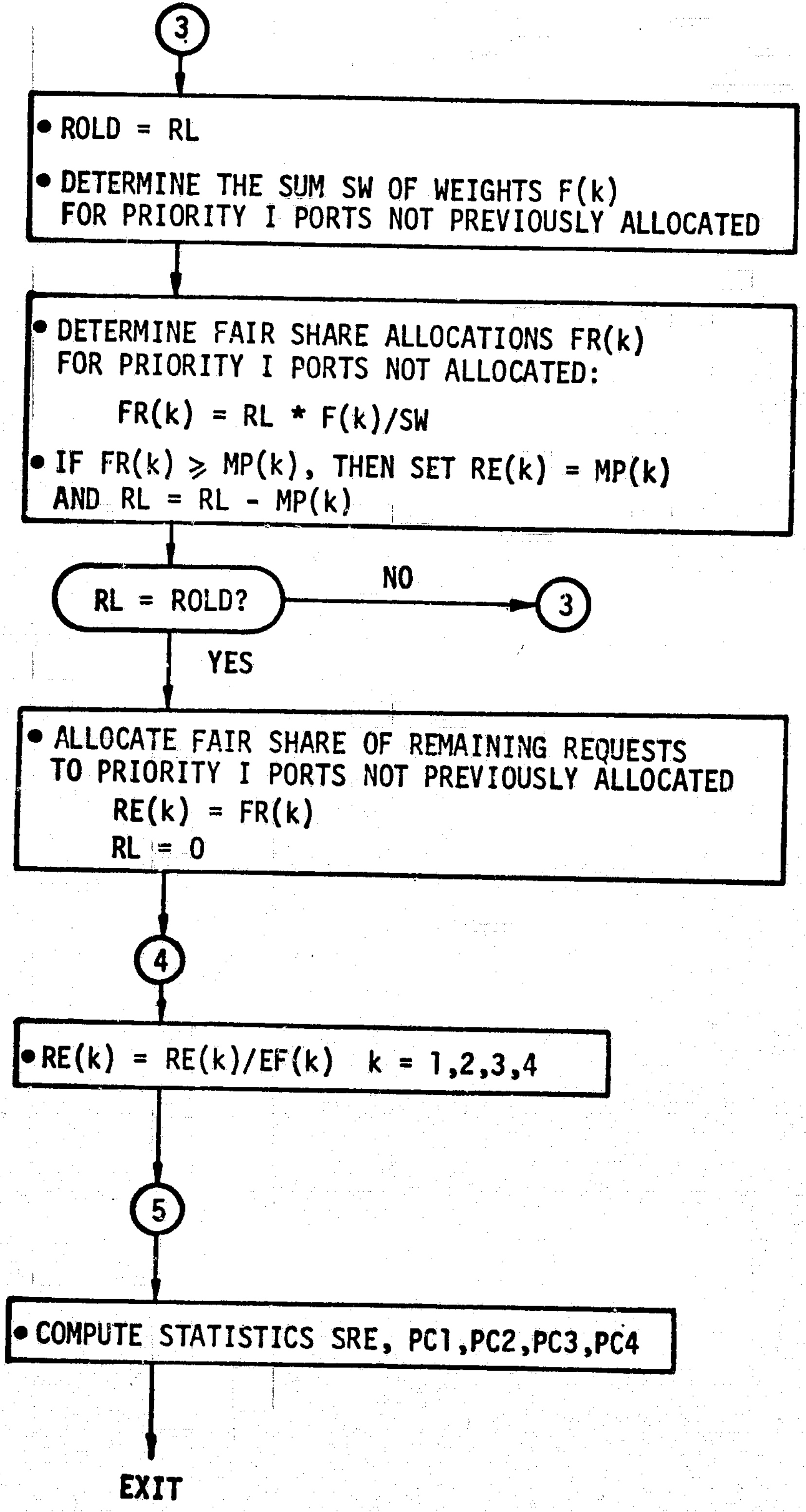
SRE		Sum of energy requested	kwh
PC	1,2,3,4	Percent of cumulative load request delivered by port i	%

[1] No capital costs assigned since this is an allocation component, not a physical device.

CALCULATION LOGIC


• PO = ΣP(k)
• IF PS(k) ≤ 0 SET MP(k) = 0 k = 1,2,3,4
• MPO = ΣMP(k)
MPO ≤ REO ?
YES
• RE(k) = MP(k)/EF(k)
• SP = REO-MPO
• GO TO 5
NO
• INITIALIZE REQUEST LOGIC
SP = RE(k) = 0. K=1,2,3,4
RL = REO
I = 1
1
NO
I > 4 ?
YES
4
I = I+1
• SUM MAXIMUM POWER SMP FOR PORTS WITH PRIORITY I
SMP = 0?
YES
2
NO
SMP ≤ RL ?
NO
• GO TO FAIR SHARE ALLOCATION 3 (NEXT PAGE)
YES
• SET EACH PRIORITY I REQUEST AT MAXIMUM POWER
• RL = RL - SMP
• GO TO 2

PA FAIR SHARE ALLOCATION


3
• ROLD = RL
• DETERMINE THE SUM SW OF WEIGHTS F(k) FOR PRIORITY I PORTS NOT PREVIOUSLY ALLOCATED
• DETERMINE FAIR SHARE ALLOCATIONS FR(k) FOR PRIORITY I PORTS NOT ALLOCATED:
FR(k) = RL * F(k)/SW
• IF FR(k) ≥ MP(k), THEN SET RE(k) = MP(k) AND RL = RL - MP(k)
RL = ROLD?
NO
3
YES
• ALLOCATE FAIR SHARE OF REMAINING REQUESTS TO PRIORITY I PORTS NOT PREVIOUSLY ALLOCATED
RE(k) = FR(k)
RL = 0
4
• RE(k) = RE(k)/EF(k) k = 1,2,3,4
5
• COMPUTE STATISTICS SRE, PC1,PC2,PC3,PC4
EXIT

```
CPA
      SUBROUTINE PA(MPO,
     1          R1, R2, R3, R4,
     2          PO,SP,
     3          SR,PC1,PC2,PC3,PC4,
     4          RO,
     4          EF1, EF2, EF3, EF4,
     5          P1, P2, P3, P4,
     6          PR1, PR2, PR3, PR4,
     7          W1, W2, W3, W4,
     8          MP1, MP2, MP3, MP4)
C
C      PURPOSE.   MODEL POWER ACCUMULATOR
C
C      METHOD.  PRIMARY REQUEST ALLOCATION RESULTING FROM PRIORITY
C      ASSIGNMENTS.  SECONDARY REQUEST ALLOCATION RESULTING
C      FROM WEIGHT ASSIGNMENTS.
C      THAT IS, REQUESTS ARE ALLOCATED ACCORDING TO:
C        *  PORT PRIORITY (HIGHEST PRIORITY = 1)
C        *  PORT WEIGHTS (IN CASE OF EQUAL PRIORTIES.   )
C
C      FORMAL ARGUMENT DEFINITION.
C      R1,...,  R4 :   POWER REQUESTS IN KW         (OUTPUTS)
C      MPO :           TOTAL MAXIMUM POWER          (OUTPUT)
C      SP   :          SURPLUS REQUEST              (OUTPUT)
C      PO   :          TOTAL LOAD IN KW             (OUTPUT)
C      SR   :          SUM OF ENERGY REQUESTED, KWH (OUTPUT)
C      PC1,...,PC4     PERCENT OF CUM LOAD DELIVERED (OUTPUT)
C      RO :            TOTAL POWER REQUESTED,KW     (INPUT)
C      P1,..., P4 :    INPUT POWER IN KW            (INPUTS)
C      PR1,..., PR4 :  PORT PRIORITIES              (INPUTS)
C      W1,..., W4 :    PORT WEIGHTS                 (INPUTS)
C      MP1, ..., MP4 :   MAXIMUM POWERS             (INPUTS)
C      EF1, ..., EF4 :   EFFICIENCIES               (INPUTS)
C       COMMON STORAGE
       COMMON/ CIMPL / IMPL
       COMMON / CSIMUL / DUM(6), TINC, TMAX
      REAL MPO,MP1,MP2,MP3,MP4
C
C      LOCAL VARIABLES
C
C
C      R(K) IS THE POWER REQUEST AT PORT K
      REAL R(4)
C
C      PR(K) IS THE PRIORITY ASSIGNED TO PORT K
      REAL PR(4)
C
C      W(K) IS THE WEIGHT ASSIGNED TO PORT K
      REAL W(4)
C
C      MP(K) IS MAXIMUM POWER TO BE ALLOCATED TO PORT K
      REAL MP(4)
C
C      SW(I) IS THE SUM OF THE WEIGHTS ASSIGNED TO PRIORITY-I PORTS
      REAL SW(4)
C
C      SMP(I) IS THE SUM OF THE MAXIMUM POWER AT PRIORITY-I PORTS
```

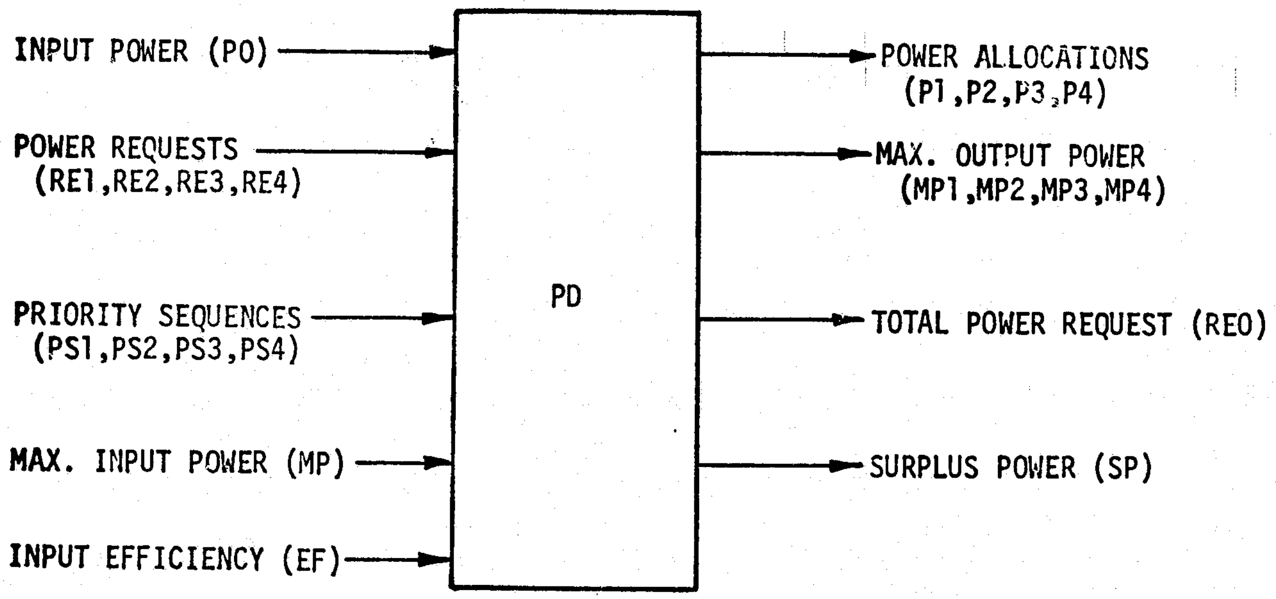
```
      REAL SMP(4)
C
C     FRU IS  FAIR SHARE  UNIT FOR PRIORITY-1 PORTS
C
C     FR(K) IS THE COMPUTED  FAIR SHARE  REQUEST FOR PORT K
      REAL FR(4)
C
C     LL IS THE LOAD LEFT AT EACH POINT IN THE ITERATION
      REAL LL,LOLD
C
C     IF IMPL IS ZERO, THEN ASSIGN DEFAULT VALUES
      IF (IMPL .GT. 0) GO TO 40
      RO = 0.0
      IF (PR1 .EQ. 0.99999) PR1 = 1.0
      IF (PR2 .EQ. 0.99999) PR2 = 2.0
      IF (PR3 .EQ. 0.99999) PR3 = 3.0
      IF (PR4 .EQ. 0.99999) PR4 = 4.0
      IF (MP1 .EQ. 0.99999) MP1 = 0
      IF (MP2 .EQ. 0.99999) MP2 = 0
      IF (MP3 .EQ. 0.99999) MP3 = 0
      IF (MP4 .EQ. 0.99999) MP4 = 0
      IF(P1  .EQ. .99999) P1=0.0
      IF(P2  .EQ. .99999) P2=0.0
      IF(P3  .EQ. .99999) P3= 0.0
      IF(P4  .EQ. .99999) P4=0.0
      SR=0.
      PC1=0.
      PC2=0.
      PC3=0.
      PC4=0.
      TINC1= 0.5*TINC
   40 CONTINUE
C
C     IF THE TOTAL MAXIMUM POWER IS .LE. TOTAL POWER
C     REQUESTED, THEN SUBMIT REQUESTS AT MAX-POWER, SET REQUEST
C     SURPLUS EQUAL TO THE DIFFERENCE, AND RETURN
      PO = P1 + P2 + P3 + P4
      IF(PR1.LE.0.0) MP1=0.
      IF(PR2.LE.0.0) MP2=0.
      IF(PR3.LE.0.0) MP3=0.
      IF(PR4.LE.0.0) MP4=0.
      MPO = MP1 + MP2 + MP3 + MP4
      IF (MPO .GT. RO) GO TO 80
      R1 = MP1/EF1
      R2 = MP2/EF2
      R3 = MP3/EF3
      R4 = MP4/EF4
      SP = RO - MPO
      GO TO 500
   80 CONTINUE
C
C     PROCEED WITH ALLOCATION ALGORITHM SINCE THE SUM OF
C     ALL MAXIMUM POWER INPUTS EXCEEDS THE TOTAL REQUEST RO
C
C     INITIALIZATION
      LL = RO
      R1 = 0.0
      R2 = 0.0
      R3 = 0.0
      R4 = 0.0
      SP = 0.0
C
C     IF THE TOTAL REQUEST (OR LOAD) IS ZERO, THEN RETURN
      IF (RO .LE. 0.0) GO TO 500
      R(1)=R1
      R(2)=R2
      R(3)=R3
      R(4)=R4
      PR(1) = PR1
      PR(2) = PR2
      PR(3) = PR3
      PR(4) = PR4
      W(1) = W1
      W(2) = W2
      W(3) = W3
      W(4) = W4
      MP(1) = MP1
      MP(2) = MP2
      MP(3) = MP3
      MP(4) = MP4
C
C
C     ITERATE ON PRIORITY I FOR I = 1, 2, 3, 4
C
      DO 1000 I = 1, 4
C
      XI = I
C     OBTAIN SUM OF MAXIMUM POWER FOR PORTS WITH PRIORITY I
      SMP(I) = 0.0
      DO 100 K = 1, 4
      IF (PR(K) .EQ. XI) SMP(I) = SMP(I) + MP(K)
  100 CONTINUE
C
C     IF NO PRIORITY-I MAXIMUM POWER EXISTS, THEN PROCEED WITH
C     THE NEXT HIGHER PRIORITY
      IF (SMP(I) .EQ. 0.0) GO TO 1000
C
C     IF THE SUM OF ALL PRIORITY-I MAXIMUM POWER .GT. LOAD
C     LEFT, THEN GO AROUND
      IF (SMP(I) .GT. LL) GO TO 400
C
C     THE SUM OF ALL PRIORITY-I MAXIMUM POWER .LE. LOAD
C     LEFT, SO SUBMIT EACH PRIORITY-I REQUEST
      DO 200 K = 1, 4
      IF (PR(K) .EQ. XI) R(K) = MP(K)
  200 CONTINUE
C
C     UPDATE LOAD LEFT
      LL = LL - SMP(I)
C
C     IF THE REMAINING LOAD IS ZERO, THEN EXIT THE ITERATION
      IF (LL .LE. 0.0) GO TO 2000
C
C     OTHERWISE, PROCEED WITH NEXT HIGHER PRIORITY
      GO TO 1000
C
```

PA

```
  400 CONTINUE
C
C     THE SUM OF THE PRIORITY-1 MAXIMUM POWER EXCEEDS THE
C     LOAD LEFT, SO COMPUTE AND SUBMIT FAIR SHARE REQUESTS
C      TO EACH PRIORITY-I PORT
C
  600 CONTINUE
C
C     SAVE LL FOR LATER REFERENCE
      LOLD = LL
C
C     DETERMINE FAIR SHARE UNITS FOR ALL PRIORITY-I
C     PORTS TO WHICH NO REQUEST HAS BEEN SUBMITTED
      SW(I) = 0.0
      DO 700 K = 1, 4
      IF (R(K) .NE. 0.0) GO TO 700
      IF (PR(K) .EQ. XI) SW(I) = SW(1) + W(K)
  700 CONTINUE
      FRU = 1.0 / SW(I)
C
C     FIRST, SUBMIT FAIR SHARE REQUESTS TO PORTS FOR WHICH THE
C     FAIR SHARE REQUEST EXCEEDS THE MAXIMUM POWER.  CONSIDER ONLY
C     PORTS TO WHICH NO REQUEST HAS BEEN SUBMITTED
      DO 800 K = 1, 4
      IF (R(K) .NE. 0.0) GO TO 800
      IF (PR(K) .NE. XI) GO TO 800
C
C     COMPUTE FAIR SHARE
      FR(K) = (W(K) * FRU) * LL
C
C     IF FAIR SHARE EXCEEDS MAXIMUM POWER, THEN SUBMIT REQUEST
      IF (FR(K) .GE. MP(K)) R(K) = MP(K)
C     - - - AND REDUCE LOAD LEFT TALLY
      IF (FR(K) .GE. MP(K)) LL = LL - MP(K)
  800 CONTINUE
C
C     IF LL .NE. LOLD, THEN LL WAS REDUCED DURING THE
C     PROCESSING IN THE  DO 800  LOOP ABOVE.  THIS CHANGES
C     THE FAIR SHARE COMPUTATION.  IT IS THEREFORE
C     NECESSARY TO GO BACK THROUGH THE  DO 800  LOOP IN
C     ORDER TO RECONSIDER ANY PORT WHICH MAY NOW
C     SATISFY THE REQUIREMENT THAT FR(K) .GE. MP(K).  ONLY
C     PRIORITY-I PORTS TO WHICH NO REQUEST HAS BEEN
C     MADE ARE ELIGIBLE FOR RECONSIDERATION
      IF (LL .LT. LOLD) GO TO 600
C
C     FINALLY, SUBMIT REQUESTS TO THOSE PORTS FOR WHICH THE FAIR SHARE
C     .LT. THAN THEIR MAXIMUM POWER.  CONSIDER ONLY
C     PRIORITY-I PORTS TO WHICH NO REQUEST HAS BEEN SUBMITTED
      DO 900 K = 1, 4
      IF (R(K) .NE. 0.0) GO TO 900
      IF (PR(K) .NE. XI) GO TO 900
      R(K) = FR(K)
  900 CONTINUE
      LL=0.0
      GO TO 2000
C
 1000 CONTINUE
```

```
C
 2000 CONTINUE
C
C     FINALLY, ASSIGN OUTPUTS TO NON-SUBSCRIPTED FORMAL PARAMETERS.
C     ALSO, MODIFY ALL REQUESTS ACCORDING TO THE INPUT EFFICIENCIES
      R1 = R(1) / EF1
      R2 = R(2) / EF2
      R3 = R(3) / EF3
      R4 = R(4) / EF4
      SP = LL
  500 IF(IMPL.LE.1) RETURN
      SRO= SR
      SR=SR+ RO*TINC1
      IF(SR.LE.0.) RETURN
      SRO=SRO/SR
      SRI= TINC1*100./SR
      PC1= PC1*SRO + P1*SRI
      PC2= PC2*SRO + P2*SRI
      PC3= PC3*SRO + P3*SRI
      PC4= PC4*SRO + P4*SRI
      RETURN
      END
```

7.31 POWER DIVIDER


INPUT POWER (PO)
POWER REQUESTS (RE1,RE2,RE3,RE4)
PRIORITY SEQUENCES (PS1,PS2,PS3,PS4)
MAX. INPUT POWER (MP)
INPUT EFFICIENCY (EF)
PD
POWER ALLOCATIONS (P1,P2,P3,P4)
MAX. OUTPUT POWER (MP1,MP2,MP3,MP4)
TOTAL POWER REQUEST (REO)
SURPLUS POWER (SP)

This component allocates power to four ports plus surplus based on priority and allocation weights for equal priority ports. Each port is assigned a priority sequence from 1 to 4, and a weighting $F_i > 0$, i=1,2,3,4 for proportional allocation among equal priority ports. If power available exceeds the power requested for the ports of highest priority, then the remaining power is allocated to ports having the next highest priority. If power available is less than the power requested for ports of equal priority, then power is allocated among them in proportion to their respective allocation weights.

The total power request is the sum of the port requests divided by input efficiency. The maximum power outputs MP1,...MP4 are necessary for direct connections to a power accumulator PA. These variables may be used as maximum power inputs to other components, although such connections are not required. (See 1.2.2 and 7c for further discussion.)

Inputs[1]

Parameter/Port		Description	Units
P	0	Input power	kw
RE	1,2,3,4	Power request of output ports	kw
PS	1,2,3,4	Priority sequence (default = 1,2,3,4)	kw
F	1,2,3,4	Allocation weight (for equal priorities)	-
MP		Maximum input power (default = PO)	kw
EF		Input efficiency	-

Outputs

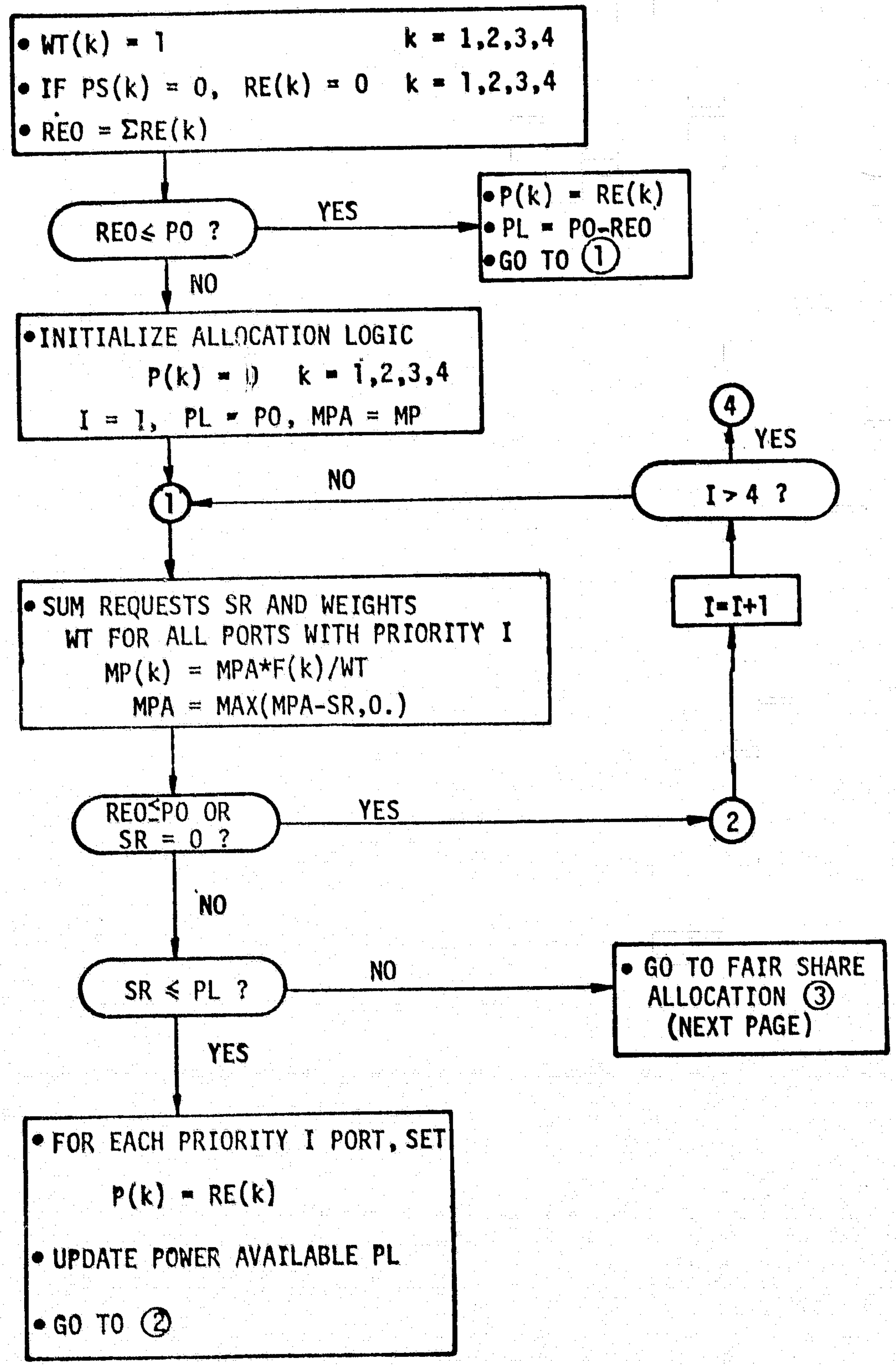
Variable/Port			
P	1,2,3,4	Output power for port i	kw
RE	0	Output power request	kw
MP	1,2,3,4	Output maximum power based on MP	kw

Statistics

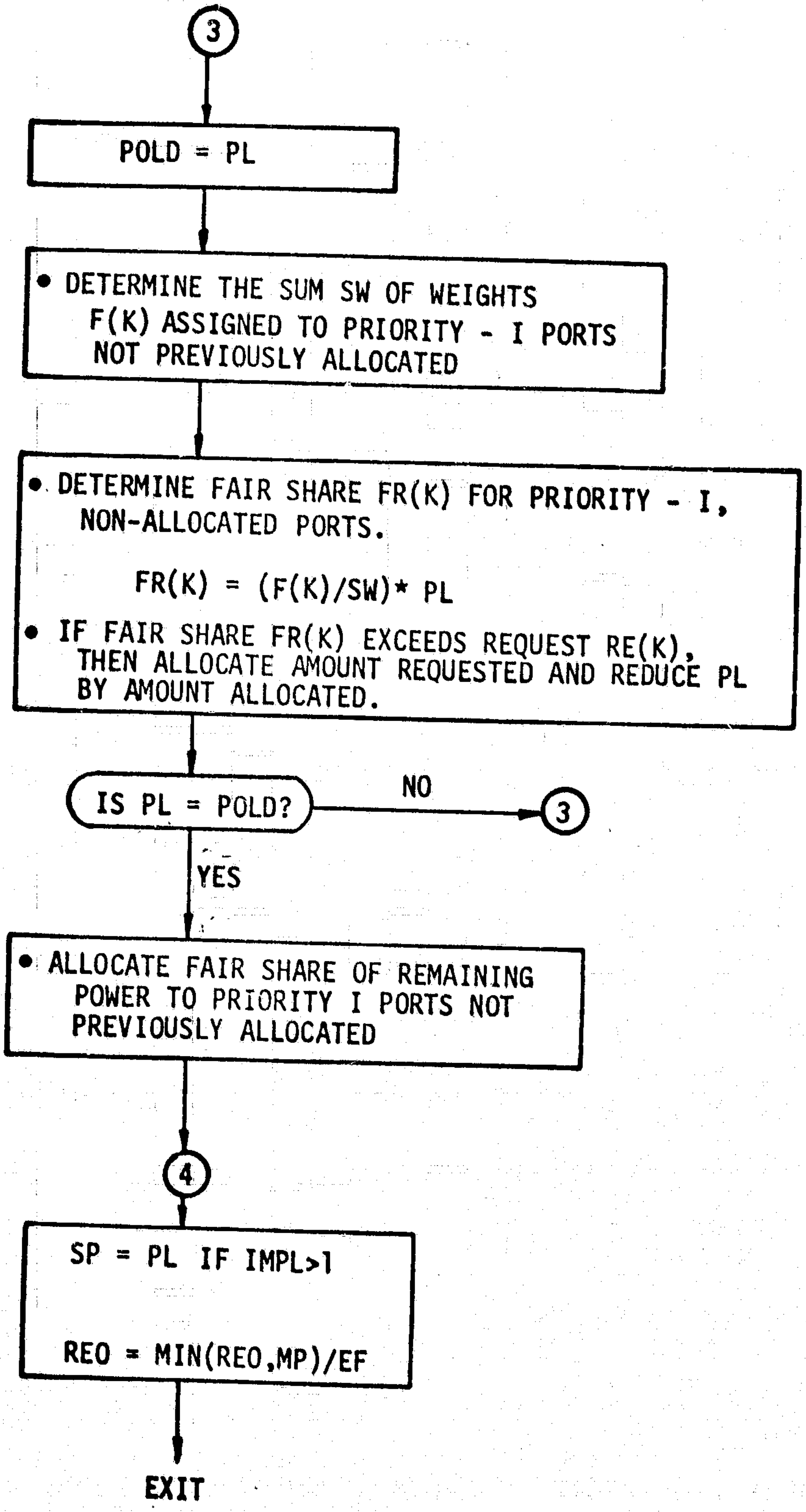
SP		Surplus power	kw

[1] No capital costs assigned since this is an allocation component, not a physical device.

CALCULATION LOGIC



PD FAIR SHARE ALLOCATION


3
POLD = PL
• DETERMINE THE SUM SW OF WEIGHTS F(K) ASSIGNED TO PRIORITY - I PORTS NOT PREVIOUSLY ALLOCATED
• DETERMINE FAIR SHARE FR(K) FOR PRIORITY - I, NON-ALLOCATED PORTS.
FR(K) = (F(K)/SW)* PL
• IF FAIR SHARE FR(K) EXCEEDS REQUEST RE(K), THEN ALLOCATE AMOUNT REQUESTED AND REDUCE PL BY AMOUNT ALLOCATED.
IS PL = POLD?
NO
3
YES
• ALLOCATE FAIR SHARE OF REMAINING POWER TO PRIORITY I PORTS NOT PREVIOUSLY ALLOCATED
4
SP = PL IF IMPL>1
REO = MIN(REO,MP)/EF
EXIT

```
CPD
      SUBROUTINE PD(
     1      P1, P2, P3, P4,
     2      RO,
     3      SP,PM1,PM2,PM3,PM4,
     4      PO,
     5      R1, R2, R3, R4,
     6      PR1, PR2, PR3, PR4,
     7      W1, W2, W3, W4, PM,EF)
C
C     PURPOSE.   MODEL POWER DIVIDER
C
C     METHOD.   PRIMARY FLOW ALLOGATION RESULTING FROM PRIORITY
C     ASSIGNMENTS.   SECONDARY FLOW ALLOCATION RESULTING
C     FROM WEIGHT ASSIGNMENTS.
C     THAT IS, TOTAL AVAILABLE POWER IS ALLOCATED
C     ACCORDING TO:
C       *  PORT REQUESTS
C       *  PORT PRIORITY (HIGHEST PRIORITY = 1)
C       *  PORT WEIGHTS (IN CASE OF EQUAL PRIORTIES)
C
C     ALLOCATION SCHEME.
C     IS SUM OF ALL REQUESTS .LT. POWER AVAILABLE PO
C     YES.
C       FULFILL EACH REQUEST
C       UPDATE POWER AVAILABLE
C       EXIT
C     NO.
C       IS SUM OF ALL PRIORITY-1 REQUESTS .LT. PO
C       YES.
C         FULFILL EACH PRIORITY-1 REQUEST
C         UPDATE POWER AVAILABLE (TO PL)
C         GO ON TO PRIORITY-2 REQUESTS
C       NO.
C         ALLOCATE  FAIR SHARE  TO EACH PRIORITY-1 PORT
C         EXIT.
C         IS SUM OF ALL PRIORITY-2 REQUESTS .LT. PL
C
C      AND SO ON AND SO FORTH
C
C     FORMAL ARGUMENT DEFINITION.
C     P1,...., P4 :   POWER ALLOCATIONS IN KW  (OUTPUTS)
C     RO :           TOTAL POWER REQUESTED    (OUTPUT)
C     SP   :          SURPLUS POWER            (OUTPUT)
C     PM1,....,PM4:  PORT MAXIMUM OUTPUT POWER IN KW      (OUTPUT)
C     PO   :          TOTAL POWER INPUT IN KW  (INPUT)
C     PM   :          MAXIMUM INPUT POWER IN KW  (INPUT)
C      EF  :          INPUT EFFICIENCY  (INPUT)
C     R1,...., R4 :   PORT REQUESTS IN KW      (INPUTS)
C     PR1,...., PR4 : PORT PRIORITIES          (INPUTS)
C     W1,...., W4 :   PORT WEIGHTS            (INPUTS)
C
C      COMMON STORAGE
       COMMON/ CIMPL / IMPL
C
C     LOCAL VARIABLES
C
C     P(K) IS THE POWER ALLOCATED TO PORT K
```

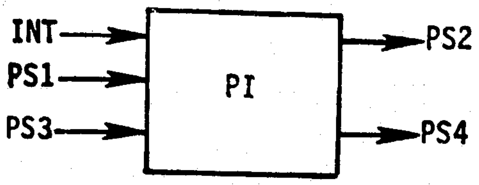
```
      REAL P(4)
C
C     R(K) IS THE POWER REQUEST AT PORT K
      REAL R(4)
C
C     PR(K) IS THE PRICRITY ASSIGNED TO PORT K
      REAL PR(4)
C
C     W(K) IS THE WEIGHT ASSIGNED TO PORT K
      REAL W(4)
C
C     SW(I) IS THE SUM OF THE WEIGHTS ASSIGNED TO PRIORITY-I PORTS
      REAL SW(4)
C
C     SR(I) IS THE SUM OF THE REQUESTS AT PRIORITY-I PORTS
      REAL SR(4)
C
C     FRU IS  FAIR SHARE  UNIT FOR PRIORITY-I PORTS
C
C     FR(K) IS THE COMPUTED  FAIR SHARE  ALLOCATION TO PORT K
      REAL FR(4)
C
C     PL IS THE POWER LEFT AT EACH POINT IN THE ITERATION
      REAL PL
C
C     IF IMPL IS ZERO, THEN ASSIGN DEFAULT VALUES
      IF (IMPL .GT. 0) GO TO 40
      R1 = 0.0
      R2 = 0.0
      R3 = 0.0
      R4 = 0.0
      IF (PR1 .EQ. 0.99999) PR1 = 1.0
      IF (PR2 .EQ. 0.99999) PR2 = 2.0
      IF (PR3 .EQ. 0.99999) PR3 = 3.0
      IF (PR4 .EQ. 0.99999) PR4 = 4.0
C
   40 CONTINUE
C
C     IF THE TOTAL POWER REQUESTED IS .LE. TOTAL POWER
C     INPUT, THEN SATISFY REQUESTS, SET POWER SURPLUS
C     EQUAL TO THE DIFFERENCE,
      IF(PR1.LE.0.0) R1=0.0
      IF(PR2.LE.0.0) R2=0.0
      IF(PR3.LE.0.0) R3=0.0
      IF(PR4.LE.0.0) R4=0.0
      RO = R1 + R2 + R3 + R4
      IF (RO .GT. PO) GO TO 80
      P1 = R1
      P2 = R2
      P3 = R3
      P4 = R4
      PL = PO - RO
      GO TO 60
   80 CONTINUE
C
C     PROCEED WITH ALLOCATION ALGORITHM SINCE THE SUM OF
C     ALL REQUESTS EXCEEDS THE TOTAL AVAILABLE POWER PO
C
```

```
C     INITIALIZATION
      PL = PO
      P1 = 0.0
      P2 = 0.0
      P3 = 0.0
      P4 = 0.0
C
   60 PMA= PM
      IF(PM.EQ. .99999) PMA=PO
      P(1) = P1
      P(2) = P2
      P(3) = P3
      P(4) = P4
      R(1) = R1
      R(2) = R2
      R(3) = R3
      R(4) = R4
      PR(1) = PR1
      PR(2) = PR2
      PR(3) = PR3
      PR(4) = PR4
      W(1) = W1
      W(2) = W2
      W(3) = W3
      W(4) = W4
C
C
C     ITERATE ON PRIORITY 1 FOR I = 1, 2, 3, 4
C
      DO 1000 I = 1, 4
C
      XI = I
C     OBTAIN SUM OF REQUESTS FROM PORTS WITH PRIORITY I
      SR(I) = 0.0
      WT=0.0
      DO 100 K = 1, 4
      IF (PR(K) .EQ. XI) SR(I) = SR(I) + R(K)
      IF(PR(K) .EQ. XI) WT= WT+ W(K)
  100 CONTINUE
C
      IF(PR1 .EQ. XI) PM1= PMA*W1/WT
      IF( PR2.EQ. XI) PM2= PMA*W2/WT
      IF(PR3 .EQ. XI) PM3= PMA*W3/WT
      IF(PR4 .EQ. XI) PM4= PMA*W4/WT
      PMA= AMAX1( PMA- SR(I),0.)
      IF(PL.LE.0.)GO TO 1000
C
C     IF NO PRIORITY-I REQUESTS EXIST, THEN PROCEED WITH
C     THE NEXT HIGHER PRIORITY
      IF (SR(I) .EQ. 0.0) GO TO 1000
      IF(RO.LE.PO) GO TO 1000
C
C     IF THE SUM OF ALL PRIORITY-I REQUESTS .GT. POWER
C     AVAILABLE, THEN GO AROUND
      IF (SR(I) .GT. PL) GO TO 400
C
C     THE SUM OF ALL PRIORITY-I REQUESTS .LE. POWER
C     AVAILABLE, SO FULFILL EACH PRIORITY-I REQUEST
```

```
      DO 200 K = 1, 4
      IF (PR(K) .EQ. XI) P(K) = R(K)
  200 CONTINUE
C
C     UPDATE POWER AVAILABLE
      PL = PL - SR(I)
C
      GO TO 1000
C
  400 CONTINUE
C
C     THE SUM OF THE PRIORITY-I REQUESTS EXCEEDS THE
C     POWER AVAILABLE, SO COMPUTE AND ALLOCATE FAIR
C     SHARE TO EACH PRIORITY-I PORT
C
  600 CONTINUE
C
C     SAVE PL FOR LATER REFERENCE
      POLD = PL
C
C     DETERMINE FAIR SHARE UNITS FOR ALL PRIORITY-I
C     PORTS FOR WHICH NO ALLOCATION HAS BEEN MADE
      SW(I) = 0.0
      DO 700 K = 1, 4
      IF (P(K) .NE. 0.0) GO TO 700
      IF (PR(K) .EQ. XI) SW(I) = SW(I) + W(K)
  700 CONTINUE
      FRU = 1.0 / SW(I)
C
C     FIRST, ALLOCATE FAIR SHARE TO PORTS FOR WHICH THE
C     FAIR SHARE EXCEEDS THE REQUEST.  CONSIDER ONLY PRIORITY-I
C     PORTS, AND CONSIDER ONLY PORTS TO WHICH NO ALLOCATION
C     HAS YET BEEN MADE
      DO 800 K = 1, 4
      IF (P(K) .NE. 0.0) GO TO 800
      IF (PR(K) .NE. XI) GO TO 600
C
C     COMPUTE FAIR SHARE
      FR(K) = (W(K) * FRU) * PL
C
C     IF FAIR SHARE EXCEEDS REQUEST, THEN FULFILL REQUEST
      IF (FR(K) .GE. R(K)) P(K) = R(K)
C     - - - AND REDUCE AVAILABLE POWER
      IF (FR(K) .GE. R(K)) PL = PL - P(K)
  800 CONTINUE
C
C     IF PL .NE. POLD, THEN PL WAS REDUCED DURING THE
C     PROCESSING IN THE  DO 800  LOOP ABOVE.  THIS CHANGES
C     THE FAIR SHARE COMPUTATION.  IT IS THEREFORE
C     NECESSARY TO GO BACK THROUGH THE  DO 800  LOOP IN
C     ORDER TO RECONSIDER ANY PORT WHICH MAY NOW
C     SATISFY THE REQUIREMENT THAT FR(K) .GE. R(K).  ONLY
C     PRIORITY-I PORTS FOR WHICH NO ALLOCATION HAS BEEN
C     MADE ARE ELIGIBLE FOR RECONSIDERATION
      IF (PL .NE. POLD) GO TO 600
C
C     FINALLY, ALLOCATE POWER TO THOSE PORTS REQUESTING
C     MORE THAN THEIR FAIR SHARE.  CONSIDER ONLY
```

```
C     PRIORITY-1 PORTS FOR WHICH NO ALLOCATION HAS BEEN MADE
      DO 900 K = 1, 4
      IF (P(K) .NE. 0.0) GO TO 900
      IF (PR(K) .NE. X1) GO TO 900
      P(K) = FR(K)
  900 CONTINUE
      PL = 0.0
C
 1000 CONTINUE
C
C
C     FINALLY, ASSIGN OUTPUTS TO NON-SUBSCRIPTED
C     FORMAL PARAMETERS
      P1 = P(1)
      P2 = P(2)
      P3 = P(3)
      P4 = P(4)
      IF(IMPL.GT.1) SP=PL
      RO=AMIN1(RO,PM)/EF
      RETURN
      END
```

7.32 PRIORITY INTERRUPT



This component is used by the storage components to change priority of the power requests when minimum or maximum capacity is approached.

Inputs

Parameter/Port		Description
PS	1	Input priority for PS2 output (0 to 4)
PS	3	Input priority for PS4 output (default=PS1)
PMX		Maximum priority for PS2 (default = 1)
INT		Interrupt flag (0,-1,1)

Outputs

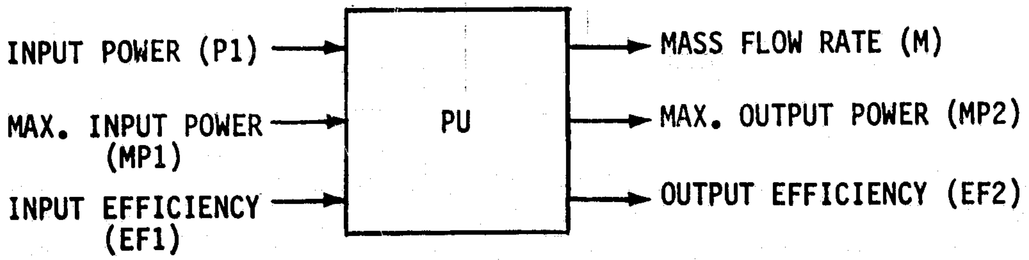
Variable/Port		
PS	2	Output priority for charge cycle
PS	4	Output priority for discharge cycle

Equations

$$
\begin{array}{ll}
PS2 = PS1 & \text{if } INT=0 \\
PS2 = PMX & \text{if } INT>0 \\
PS2 = 0 & \text{if } INT<0 \\
PS4 = PS3 & \text{if } INT\leq 0 \\
PS4 = 0 & \text{if } INT>0
\end{array}
$$

```
CPI
      SUBROUTINE PI(PS2,PS4,PS1,PS3,PMX,INT)
C
C    PURPOSE     CHANGE PRIORITY OF POWER ALLOCATION TO STORAGE COMPONENTS
C
C    WRITTEN BY A.W.WARREN                                  VERSION 1, APRIL 14 19
C
C    CALL SEQUENCE
C                 PS2 - OUTPUT PRIORITY (0 TO 4)
C                 PS4 - OUTPUT PRIORITY (COMPLEMENT TO PS2)
C                 PS1 - INPUT PRIORITY FOR PS2
C                 PS3 - INPUT PRIORITY FOR PS4
C                 PMX - MAXIMUM PRIORITY FOR PS2
C                 INT - INTERRUPT FLAG
C                         0= NO INTERRUPT
C                         1= INCREASE ALLOCATION PRIORITY
C                        -1= DECREASE ALLOCATION PRIORITY
C
      REAL INT
      COMMON /CIMPL/IMPL
      IF(IMPL.GT.0) GO TO 10
      IF(PS3.EQ. .99999) PS3=PS1
      IF(PMX.EQ..99999)PMX=1.
C
   10 PS2=PS1
      PS4=PS3
      IF(INT.GT.0.) PS2=PMX
      IF(INT.LT.0.) PS2=0.
      IF(INT.GT.0) PS4= 0.
      RETURN
      END
```

7.33 HYDRAULIC PUMP



The hydraulic pump model is based on a constant speed design. The pump is assumed to be designed to a nominal operating point and input power. For off-design performance the pump efficiency is assumed to be functionally related to the square root of the mass flow rate.

Basic Equations

The output mass flow rate is based on the equations

$$M = P1*EFF/(C1*C2*H1)$$

$$EFF = 1 - (1-EFD)*SQRT(MD/M)$$

where C1, C2 are conversion constants

PU

Inputs

Parameter/Port		Description	Units
P	1	Input power	kw
H	1	Height of water above inlet	ft
EFD		Pump efficiency at design pt. (D = 0.90)	-
MD		Mass flow rate at design pt. (D = $2X10^5$)	gal/h
EF	1	Input product efficiency	-
MP	1	Input maximum charging rate	kw
MM		Maximum allowable mass flow rate (D = $3X10^5$)	gal/h
CK		Pump capacity cost coefficient[1] (D = 0.011)	
FØ		Pump exponent for cost calculations (D = 0.5)	-
Y		Pumphead exponent for cost calculations (D=0.25)	-

Outputs

Variable/Port			
M		Output mass flow rate	gal/h
EFF		Pump efficiency	-
CCØ		Pump cost/year	$
EF	2	Output product efficiency	-
MP	2	Maximum output power	kw

Statistics

M2U	Maximum output mass flow rate	gal/h

D - default values

[1] CK = capital cost (known unit)/((MD*481.2)**FO*H1 **Y * expected life time)

The calculation sequence and default values assume a constant speed hydraulic pump nominally rated for 120KW and located 200 ft. below a reservoir. The equations relating the various physical quantities and the cost estimates are based on first principles and the data presented in Reference 1, and the cost estimates on Reference 2.

Calculation Sequence

$$C1 = 0.377*10^{-6} \quad \frac{\text{kwh}}{\text{ft-lb}}$$

$$C2 = 8.3398 \text{ lb/gal}$$

1) Costs (first pass only)

$$CC = CK*(MD*481.2)^{F0}*H1**Y$$

[1] L. Marks and T. Baumeister, "Mechanical Engineers Handbook", McGraw Hill, N.Y., 1958, Section 14, p. 19.

[2] Carson and Fogleman, "Comparison of Methods for Converting Existing Power Plants to Pumped Storage Facilities", International Engineering Company, Inc., 1974.

Calculation Sequence Cont.

2) Mass flow rate and pump efficiency

If $P1 \leq 0$, set $EFF = 1$, $M = 0$ and go to 3)

Solve the basic equations for M and EFF using:

$$X^3 - XA + B = 0$$

where

$$A = P1/(C1*C2*H1)$$

$$B = A*(1-EFD)* \sqrt{MD}$$

$$M = X^2$$

$$EFF = 1-(1-EFD)* \sqrt{MD}/X$$

3) Product efficiency and maximum charge rate

$$EF2 = EF1*EFF$$

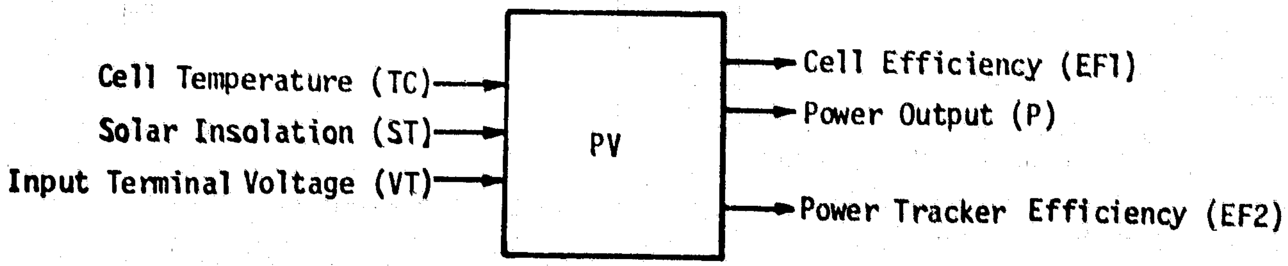
$$MP2 = MIN(MP1*EFF, \quad MM*C1*C2*H1)$$

4) Compute Statistics and Costs

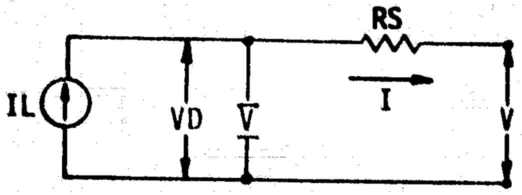
```
CPU
      SUBROUTINE PU(M,EFF,CC,EF2,MP2,M2U,P1,H1,EFD,MD,EF1,MP1,MM
     1              ,CK,FO,Y)
C
C     PURPOSE     PERFORMANCE OF HYDRAULIC PUMP
C
C     METHOD      COMPUTE PUMP FLOW RATES ASSUMING CONSTANT SPEED WITH
C
C                 EFFICIENCY A FUNCTION OF SQRT(FLOW RATE)
C
C     WRITTEN BF F. O. MAHONY                       VERSION 1, MARCH 29 1977
C
C     CALL SEQUENCE
C          OUTPUTS
C               M    - OUTPUT MASS FLOW RATE, GAL/HR
C               EFF- PUMP EFFICIENCY
C               CC   - PUMP COST/YEAR, $
C               EF2 - OUTPUT PRODUCT EFFICIENCY
C               MP2 - MAXIMUM OUTPUT CHARGE RATE, KW
C               M2U - MAXIMUM OUTPUT MASS FLOW RATE, GAL/HR
C
C          INPUTS
C               P1   - INPUT POWER, KW
C               H1   - HEIGHT OF WATER ABOVE INLET, FT
C               EFD - PUMP EFFICIENCY AT DESIGN POINT
C               MD   - MASS FLOW RATE AT DESIGN POINT, GAL/HR
C               EF1 - INPUT PRODUCT EFFICIENCY
C               MP1 - INPUT MAXIMUM CHARGING RATE, KW
C               MM   - MAXIMUM ALLOWABLE MASS FLOW RATE, GAL/HR
C               CK   - PUMP CAPACITY COST COEFFICIENT
C               FO   - PUMP EXPONENT FOR COST CALCULATIONS
C               Y    - PUMP HEAD EXPONENT FOR COST CALCULATIONS
C
      COMMON /CIMPL/IMPL /CTIME/TIME/CSIMUL/DUM(7),TMAX /COST/CCI
C
      REAL M,MP2,M2U,MD,MP1,MM
C
      IF(IMPL.GT.0)GO TO 100
C
      TMAX1=TMAX*.99999
C
      C1= 3.1441E-6
C
      IF(EFD.EQ. .99999)EFD=0.9
      IF(MD .EQ. .99999)MD =2.0E5
      IF(MP1.EQ. .99999)MP1=1.E8
      IF(MM .EQ. .99999)MM =3.0E5
      IF(CK .EQ. .99999)CK =0.011
      IF(FO .EQ. .99999)FO =0.5
      IF(Y   .EQ. .99999)Y  =0.25
      CC =CK*(MD*481.2)**FO*H1**Y
C
      M2U =0.0
  100 EFF= 1.0
      M= 0.0
      IF(P1 .LE. 0.0) GO TO 200
C
C                         SOLVE CUBIC EQUATION FOR M AND EFF
```

```
C
      A3= -P1/(C1*H1)
      A4 =-A3*(1.0-EFD)*SQRT(MD)
C
      CALL CUBIC(A3,A4,ANS)
      IF(ANS.LE.0.) GO TO 200
C
      M  =ANS**2
      EFF=1.0-(1.0-EFD)*SQRT(MD)/ANS
C
C                        PRODUCT EFFICIENCY AND CHARGE RATE
C
  200 EF2=EF1*EFF
      MP2=AMIN1(MP1*EFF,MM*H1*C1)
C
      IF(IMPL.LE.1)RETURN
C
C                        STATISTICS
C
      M2U=AMAX1(M2U,M  )
C
      IF(TIME.LT.TMAX1)RETURN
C
      CCI=CCI+CC
C
      RETURN
      END
```

7.34 SOLAR-PHOTOVOLTAIC ARRAY



The photovoltaic cell is modeled by the circuit below. Power is delivered at terminal voltage V and is dependent on the cell temperature and insolation. Default for V is the maximum power point. A square array of solar cells is assumed with both parallel and series connections.



BASIC SOLAR CELL MODEL

Basic Equations

Output current I as a function of terminal voltage V is given by the implicit relation

I = IL + IØ*(1-EXP((V+I*RS)*QBK/(T+273))) (1)

where

IL = light current (amps)

IØ = diode reverse saturation current (amps)

T = temperature (^{o}C)

RS = internal resistance (ohms)

QBK = device constant (default = electron charge/Boltzmann's constant)

The light current IL is computed by a bivariate expansion of insolation and cell temperature. It has been reported that this model fits observed solar cell characteristics within 5% at high temperatures and insolations and within less than 1% under more moderate conditions (ref. 2). The reverse saturation current IØ is given by

IØ(T) = KD*AO*((T+273)**3)EXP(-EGO/(T+273)) (2)

where

KD = a device constant

AO = a material constant

EGO = band gap at 0^{o}K/Boltzmann's constant

Tables	Description	Units
EFF	Efficiency of maximum power tracker versus fractional load (default table provided)	-
ØP	Optimum cell power versus insolation and temperature (computed table)	kw
ØV	Optimum cell voltage versus insolation and temperature (computed table)	volts

Inputs/Port	Description	Units
VT	Array terminal voltage (default = maximum power voltage)	volts
TC	Cell temperature	^oC
TL*	Low temperature value (default = 28)	^oC
TH*	High temperature value (default = 120)	^oC
TR	Temperature range (default = TH)	^oC
ST	Collector solar insolation	w/m^2
SL*	Low insolation value (default = 1000)	w/m^2
SH*	High insolation value (default = 25000)	w/m^2
SR	Insolation range (default = SH)	w/m^2
RC	Concentration ratio (default = 25)	-
AA	Total illuminated cell area (default = .00015*NS*NP)	m^2
NS	Number of cells in series (default = 300)	-
NP	Number of cells in parallel (default = 500)	-
I1*	Cell short circuit current at TL,SL (default = .06)	Amps
I2*	Cell short circuit current at TL,SH (default = 1.5)	Amps
I3*	Cell short circuit current at TH,SL (default = .06)	Amps

*These inputs may be ignored if IL1,DS,DT,DST,KD coefficients are supplied.

Inputs/Port (cont'd)	Description	Units
I4*	Cell short circuit current at TH,SH (default = 1.56)	Amps
V1*	Cell open circuit voltage at TL,SL (default = .6)	Volts
RS	Cell internal resistance (default = .055)	Ohms
AO	Material constant (default = 1.54E33 for silicon)	-
EGO	Band-gap at 0^oK normalized by Boltzmann's constant (default = 1.4E4 for silicon)	^oK
IL1	Coefficients in bivariate expansion for the light current IL. If not provided, they will be computed from the inputs I1,...,I4,	m^2V^{-1}
DS		m^2W^{-1}
DT		$1/^oC$
DST		m^2/w^oC
KD	Device constant, if not provided will be computed from I1,V1	-
CF	Lens radiation transmission coefficient	-
QBK	Device constant (default = 1.161E4)	$^oK/V$
RAP	Rated power of maximum power point tracker (default computed)	kw
CC	Capital cost/year/unit cell area	$\$/m^2$
CM	Maintenance cost/year	$

Note: Minimum input parameters to specify PV are cell area AA, number of cells in series NS and in parallel NP, concentration ratio RC, and rated power RAP. These parameters must be consistent with those for the collector model FO or FP.

* These inputs may be ignored if IL1,DS,DT,DST,KD coefficients are supplied.

Output/Port	Description	Units
V	Array terminal voltage	Volts
P	Array output power	kw
I	Array output current	Amps
EF1	Solar cell efficiency	-
EF2	Maximum power tracker efficiency	-
Statistics		
SP	Sum of energy delivered	kwh

Calculation Sequence

First Pass

1) Compute parameter KD (if not input)

$$KD = I1/\left[AO*((TL+273)**3)*EXP(-EGO/(TL+273))*(EXP(QBK*V1/(TL+273))-EXP(QBK*I1*RS/(TL+273)))\right]$$

2) Compute coefficients IL1,DS,DT,DST (if not input) in the light current bivariate expansion in temperature T and insolation S:

$$IL = IL1*S*(1+DS*(S-SL)+DT*(T-TL)+DST*(S-SL)*(T-TL)) \qquad (3)$$

Define

$$FIL(I,T) = I-I\emptyset(T)*(1-EXP(QBK*I*RS/(T+273))).$$

Then

$$IL1 = FIL\ (I1,TL)/SL$$

$$DS = (FIL(I2,TL) - IL1*SH)/(IL1*SH*(SH-SL))$$

$$DT = (FIL(I3,TH) - IL1*SL)/(IL1*SL*(TH-TL))$$

$$DST = (FIL(I4,TH) - IL1*SH-IL1*SH*DS*(SH-SL) - IL1*SH*DT*(TH-TL))/(IL1*SH*(SH-SL)*(TH-TL)).$$

3) If a terminal voltage VT is not input, calculate the optimal cell voltage V=ØV(S,T) with S ranging through 10 values equally spaced between 0 and SR, and with T ranging through 10 values equally spaced between 0 and TR, resulting in a 10 x 10 matrix ØV(S,T). The calculation is as follows: Given S and T, the open circuit voltage VOC is given by

VOC = (T+273)*ALOG(1+IL/IØ)/QBK,

where IL and IØ are computed from (2) and (3).

A binary search is performed in the range from 0 to VOC. For a value V in this range, Newton-Raphson iterations are used to solve for the terminal current I satisfying (1). The corresponding power P (in kw) is

P = I*V/1000 .

The iterative search process to maximize P is given by

(i) Take the initial interval [VL,VH] to be [0,VOC].

(ii) Compute a numerical derivative of P at the midpoint VM of [VL,VH]:

P' = (P(VM+1E-5)-P(VM))/1E-5

(iii) If P' ≥ 0, set VL = VM.

If P' < 0, set VH = VM.

(iv) IfVH-VL > 2E-5 and the number of iterations performed is 10, go to (ii). Otherwise P is maximized and

ØV(S,T) = VM

ØP(S,T) = P

The 10 x 10 matrices ØV(S,T) (optimal cell voltage) and ØP(S,T) (maximal cell power) are stored for use in subsequent passes.

Subsequent Passes

4) Compute insolation S at the cells

S = ST*RC*CF

5) If terminal voltage VT is not input, the cell terminal voltage V and power P are obtained by interpolation from the arrays ØV(S,T) and ØP(S,T). (A diagnostic is printed if S > SR or TC > TR).

6) If VT is used as an input voltage, then the cell voltage and power are determined using

V = VT/NS

I = IL(S,TC) + IØ(TC)*(1-EXP(QBK*(V+I*RS)/(TC+273)))

P = I*V/1000

7) Array outputs prior to maximum power tracker:

V = V*NS

P = P*NS*NP

I = P*1000/V

EF1 = P*1000/(S*AA) if S>0

EF2 = 1.

8) If the maximum power tracker is used,

EF2 = EFF(P/RAP)

P = P*EF2

REFERENCES FOR PV

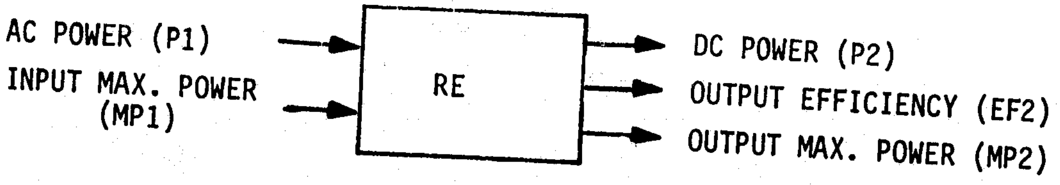
1. J. K. Linn, "Photovoltaic System Analysis Program-SOLCEL," Sandia Laboratories Report SAND77-1268, 1977.

2. L. H. Goldstein and G. R. Case, "PVSS-A Photovoltaic System Simulation Program," Sandia Laboratories, 1976.

```
CPV
      SUBROUTINE PV(EFF,OP,OV,V,P,I,EF1,EF2,SP,
     1VT,TC,TL,TH,TR,ST,SL,SH,SR,RC,AA,NS,NP,
     2I1,I2,I3,I4,V1,RS,AO,EGO,IL1,DS,DT,DST,KD,
     3CF,QBK,RAP,CC,CM)
C
C        PURPOSE    THIS COMPONENT COMPUTES THE POWER AND VOLTAGE
C                   OUTPUT OF A PHOTO-VOLTAIC CELL ARRAY GIVEN THE
C                   TEMPERATURE AND INSOLATION
C        WRITTEN BY Y.K.CHAN, 10-21-78, VERSION 1
C
C       METHOD      NEWTON RALPHSON METHOD IS USED TO CALCULATE CELL
C                   CURRENT AS FUNCTION OF INSOLATION, TEMPERATURE, AND
C                   TERMINAL VOLTAGE.  IF TERMINAL VOLTAGE IS NOT INPUT,
C                   POWER IS COMPUTED AT OPTIMAL VOLTAGE.  THIS IS DONE
C                   FOR A RANGE OF 10 VALUES OF TEMPERATURE AND 10
C                   VALUES OF INSOLATION IN THE FIRST PASS.
C                     AT SUBSEQUENT PASSES,
C                   INTERPOLATION IS USED.
C
C       CALL SEQUENCE
C            TABLES
C               EFF   -EFFICIENCY OF MAXIMUM POWER TRACKER
C                      VS FRACTIONAL LOAD (DEFAULT TABLE)
C               OP    -OPTIMAL POWER,KW, VS INSOLATION,W/M2, AND
C                      TEMPERATURE,C
C               OV    -OPTIMAL TERMINAL VOLTAGE,V, VS INSOLATION,W/M2, AND
C                      TEMPERATURE,C
C            OUTPUTS
C               V     -ARRAY TERMINAL VOLTAGE,VOLTS
C               P     -ARRAY OUTPUT POWER,KW
C               I     -ARRAY OUTPUT CURRENT,AMPS
C               EF1   -SOLAR CELL EFFICIENCY
C               EF2   -MAXIMUM POWER TRACKER EFFICIENCY
C            STATISTICS
C               SP    -SUM OF ENERGY DELIVERED,KWH
C            INPUTS
C               VT     -ARRAY TERMINAL VOLTAGE,VOLTS,(DEFAULT=MAXIMUM
C                     POWER VOLTAGE)
C               TC     -CELL TEMPERATURE,C
C               TL     -LOW TEMPERATURE VALUE,C,(DEFAULT=28)
C               TH     -HIGH TEMPERATURE VALUE,C,(DEFAULT=120)
C               TR     -TEMPERATURE RANGE,C,(DEFAULT=TH)
C               ST     -COLLECTOR SOLAR INSULATION,W/M2
C               SL     -LOW INSOLATION VALUE,W/M2,(DEFAULT=1000)
C               SH     -HIGH INSOLATION VALUE,W/M2,(DEFAULT=25000)
C               SR     -INSOLATION RANGE,W/M2,(DEFAULT=SH)
C               RC     -CONCENTRATION RATIO(DEFAULT=25)
C               AA     -TOTAL COLLECTOR CELL AREA,M2,(DEFAULT=2.5E-3)
C               NS     -NUMBER OF CELLS IN SERIES(DEFAULT=300)
C               NP     -NUMBER OF CELLS INPARALLEL(DEFAULT=500)
C               I1     -CELL SHORT CIRCUIT CURRENT AT TL,SL, AMPS
C                       (DEFAULT=.06)
C               I2     -CELL SHORT CIRCUIT CURRENT AT TL,SH, AMPS
C                       (DEFAULT=1.5)
C               I3     -CELL SHORT CIRCUIT CURRENT AT TH,SL, AMPS
C                      (DEFAULT=.06)
C               I4     -CELL SHORT CIRCUIT CURRENT AT TH,SH, AMPS
```

```
C                                   (DEFAULT=1.56)
C                         V1     -CELL OPEN CIRCUIT VLOTAGE AT TL,SL, VOLTS
C                                   (DEFAULT=.6)
C                         RS     -CELL INTERNAL RESISTANCE, OHMS,(DEFAULT=.055)
C                         AO     -MATERIAL CONSTANT(DEFAULT=1.54E33 FOR SILICON)
C                         EGO    -BAND GAP AT OK NORMALIZED BY BOLTZMANN S
C                                  CONSTANT(DEFAULT=1.4E4 FOR SILICON)
C                         IL1,DS,DT,DST
C                                -COEFFICIENTS IN BIVARIATE EXPANSION FOR THE
C                                  LIGHT CURRENT IL.  IF NOT PROVIDED, THEY WILL
C                                  BE COMPUTED FROM THE INPUTS I1,...,I4.
C                                  THE UNITS FOR IL1,DS,DT,DST ARE RESPECTIVELY
C                                  M2V-1,M2W-1,C-1,M2(WC)-1
C                         KD     -DEVICE CONSTANT. IF NOT PROVIDED, IT WILL BE
C                                  COMPUTED FROM I1,V1
C                         QBK    -DEVICE CONSTANT,K/V,(DEFAULT=ELECTRON CHARGE/
C                                   BOLTZMANN S CONSTANT=1.161E4)
C                         CF     -LENS RADIATION TRANSMITTANCE COEFFICIENT
C                         RAP     -RATED POWER OF MAXIMUM POWER POINT TRACKER,KW
C                                  (DEFAULT=LARGEST OPTIMAL POWER FOR THE RANGE
C                                  OF TC AND ST)
C                         CC     -CAPITAL COST/YEAR/UNIT CELL AREA, $/M2
C                         CM     -MAINTENANCE COST/YEAR, $
      REAL I,NS,NP,I1,I2,I3,I4,IL1,KD,IL,IO,IM,IME
      DIMENSION EFF(1),EFF1(14),OP(1),OV(1)
      COMMON /C1MPL/IMPL,ICNT,ITEST
      COMMON /CTIME/TIME /CSIMUL/DUM(7),TMAX
      COMMON /COST/CCAP,CMA,COP
      DATA EFF1/0.,.1,.2,.3,.4,.5,1.,.338,.44,.53,.61,.70,.75,.9/
      IL(S,T)=IL1*S*(1.+DS*(S-SL)+DT*(T-TL)+DST*(S-SL)*(T-TL))
      IO(T)=KD*AO*((T+273)**3)*EXP(-EGO/(T+273))
      FIL(I,T)=I-IO(T)*(1.-EXP(QBK*I*RS/(T+273)))
      IF(IMPL.GT.0)GO TO 100
      SP=0.
      TMAX1=TMAX*.99999
      TINC1=DUM(7)*0.5
C
C          INITIALIZATION
C
      IF(EFF(2).NE.1.99999)GO TO 11
      EFF(2)=7
      DO 12 II=4,17
   12 EFF(II)=EFF1(II-3)
   11 CONTINUE
      OP(2)=10.
      OP(3)=10.
      OV(2)=10.
      OV(3)=10.
      IF(TL.EQ..99999)TL=28
      IF(TH.EQ..99999)TH=120
      IF(TR.EQ..99999)TR=TH
      IF(SL.EQ..99999)SL=1000
      IF(SH.EQ..99999)SH=25000
      IF(SR.EQ..99999)SR=SH
      IF(RC.EQ..99999)RC=25
      IF(NS.EQ..99999)NS=300
      IF(NP.EQ..99999)NP=500
      IF(AA.EQ..99999)AA=1.5E-4*NS*NP
      IF(I1.EQ..99999)I1=.06
      IF(I2.EQ..99999)I2=1.5
      IF(I3.EQ..99999)I3=.06
      IF(I4.EQ..99999)I4=1.56
      IF(V1.EQ..99999)V1=.6
      IF(RS.EQ..99999)RS=.055
      IF(A0.EQ..99999)A0=1.54E33
      IF(EGO.EQ..99999)EGO=1.4E4
      IF(QBK.EQ..99999)QBK=1.161E4
C
      IF(KD.EQ..99999)KD=I1/(A0*((TL+273)**3)*EXP(-EGO/
     1 (TL+273))*(EXP(QBK*V1/(TL+273))-
     2 EXP(QBK*I1*RS/(TL+273))))
      IF(IL1.EQ..99999)IL1=FIL(I1,TL)/SL
      IF(DS.EQ..99999)DS=(FIL(I2,TL)-IL1*SH)/(IL1*SH*(SH-SL))
      IF(DT.EQ..99999)DT=(FIL(I3,TH)-IL1*SL)/(IL1*SL*(TH-TL))
      IF(DST.EQ..99999)DST=(FIL(I4,TH)-IL1*SH-
     1 IL1*SH*DS*(SH-SL)-IL1*SH*DT*(TH-TL))/
     2 (IL1*SH*(SH-SL)*(TH-TL))
C
C      CALCULATE OPTIMAL POWER OP AND CELL VOLTAGE
C
C      IF TERMINAL VOLTAGE IS NOT INPUT
C
      IF(VT.NE..99999)GO TO 100
      S=0.
      DO 33 J=1,10
      JO=J+3
      OP(JO)=(J-1)*TR/9.
   33 OV(JO)=OP(JO)
      DO 3 K=1,10
      T=0.
      DO 4 J=1,10
      AIL=IL(S,T)
      BIO=IO(T)
      VOC=(T+273)*ALOG(1.+AIL/BIO)/QBK
      VL=0.
      VH=VOC
C
C               BINARY SEARCH FOR MAX POWER POINT
C
      DO 5 M=1,10
      VM=( VL+VH)*.5
      VME=VM+1.E-5
      IM=AINR(AIL,BIO,QBK,VM,RS,T)
      PM=IM*VM
      IME=AINR(AIL,BIO,QBK,VME,RS,T)
      PME=IME*VME
      PMP=PME-PM
      IF(PMP.GE.0.)VL=VM
      IF(PMP.LT.0.)VH=VM
      IF((VH-VL).LE.2.E-5)GO TO 6
    5 CONTINUE
C
    6 CONTINUE
      IKJ=13+K+J*10
      OV(IKJ)=VM
      OP(IKJ)=PM/1000.
      T=T+TR/9.
    4 CONTINUE
      IKJO=13+K
      OP(IKJO)=S
      OV(IKJO)=S
      S=S+SR/9.
    3 CONTINUE
C
       IF(RAP.EQ..99999)RAP=OP(33)*NS*NP
C     WRITE(6,101)(OP(IK),IK=24,123)
C     WRITE(6,101)(OV(IK),IK=24,123)
C 101 FORMAT(1H0,3HPV  ,/,(5X,10E10.2))
C
  100 CONTINUE
C
C          COMPUTE INSOLATION AT THE CELLS
C
      S=ST*RC*CF
C
C          COMPUTE CELL VOLTAGE AND POWER
      IF(VT.NE..99999)GO TO 900
      IF(IMPL.NE.2)GO TO 809
      IF((S.GT.SR).OR.(TC.GT.TR))WRITE(6,808)
  808 FORMAT(1H0,62HPV  WARNING  INSOLATION OR TEMPERATURE AT CELL EXCEE
     1D RANGE       )
      IF((S.GT.SR).OR.(TC.GT.TR))ICNT=ICNT+1
  809 CONTINUE
      V=TBLU2(S,TC,OV(14),OV(4),OV(24),1,1,10,10,10,10)
      P=TBLU2(S,TC,OP(14),OP(4),OP(24),1,1,10,10,10,10)
      GO TO 901
  900 CONTINUE
      V=VT/NS
      AIL=IL(S,TC)
      BIO=IO(TC)
      I=AINR(AIL,BIO,QBK,V,RS,TC)
      P=I*V/1000.
  901 CONTINUE
C
C     COMPUTE ARRAY VOLTAGE AND POWER
C
      V=V*NS
      P=P*NS*NP
      I=0.
      IF(V.GT.0.)I=P*1000./V
      EF1=1.
      EF2=1.
      IF(S.GT.0.)EF1=P*1000/(S*AA)
      IF(VT.NE..99999)GO TO 904
      PRAT=P/RAP
      NEF=EFF(2)
      EF2=TBLU1(PRAT,EFF(4),EFF(4+NEF),1,-NEF)
      P=P*EF2
  904 CONTINUE
      IF(IMPL.LE.1)RETURN
      SP=SP+P*TINC1
      IF(TIME.LT.TMAX1)RETURN
      CCAP=CCAP+CC*AA
      CMA=CMA+CM
      RETURN
      END
```

7.35 AC-DC RECTIFIER



This component models a solid-state rectifier/transformer. Power losses due to resistive heating and contact potential loss are modeled. Default parameter values determining power losses are based on 200 kw rated power.

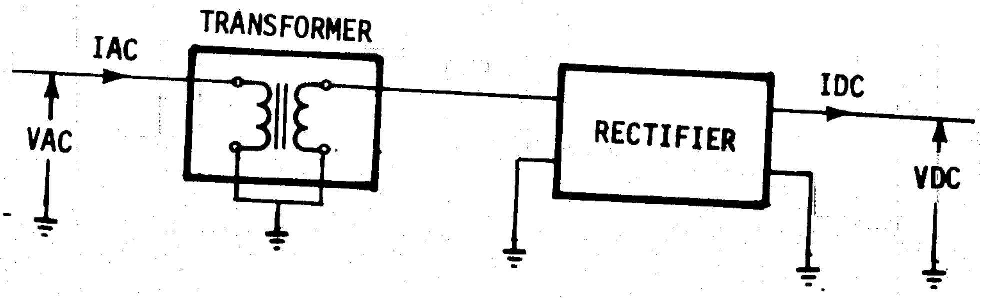


FIGURE 7.35: RECTIFIER FUNCTIONAL DIAGRAM

Inputs*

Parameter/Port		Description	Units
P	1	AC input power	kw
RT		Transformer resistance (D = 0)	ohms
XT		Transformer reactance (D = 0.03)	ohms
VAC		Rated AC voltage (D = 440)	volts
DR		Rectifier contact potential (D = 0)	volts
RR		Rectifier resistance (D = 0.02)	ohms
RAP		Rated input power	kw
EF	1	Input product efficiency	-
MP	1	Maximum input power (D = $1.\mathrm{x}10^8$)	kw
CC		Rectifier cost/year	$

Outputs

Variable/Port			
P	2	DC output power	kw
IAC		AC input current	amps
PL		Power loss	kw
EF	2	Output product efficiency	-
MP	2	Maximum output power	kw

* Minimum input parameters to specify RE are:

RR = rectifier resistance,
RAP = rated input power.

RR may be used as an adjustment parameter to obtain a specified efficiency at rated power.

D - Default values supplied.

Calculation Sequence

1) Compute transformer power angles

$$Y = SIN(\theta) = \sqrt{3}*XT*P1*1000/VAC^2$$

$ABS(Y) > 1$ ⇨ DIAGNOSTIC

2) Input and output current

If $P1 \leq 0$ set $P2 = IAC = PL = 0.$, $EFF = 1$ and go to 4)

$$IAC = VAC\ \sqrt{2-2COS(\theta)}\ /\ (\sqrt{3}*XT)$$

$$" = VAC\ \sqrt{2-2*\ \sqrt{1-Y^2}}\ /\ (\sqrt{3}*XT)$$

$$IDC = \pi*IAC\ /\ \sqrt{6}$$

3) Power loss and output power

$$PL = (\sqrt{3}*RT*IAC^2 + IDC*(DR+IDC*RR))/1000$$

$$P2 = P1 - PL$$

$$EFF = P2/P1$$

$P2 \leq 0$ ⇨ DIAGNOSTIC, $EFF = 1$

4) Efficiency and maximum power

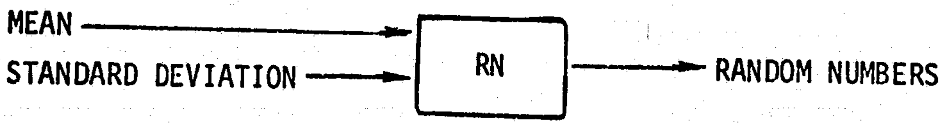
$$EF2 = EF1*EFF$$

$$MP2 = MIN(MP1,RAP)*EFF$$

5) Compute Costs

```
CRE
      SUBROUTINE RE(P2,IAC,PL,EF2,MP2,P1,RT,XT,VAC,DR,RR,RAP,EF1,MP1,CC)
C
C                 PURPOSE          SOLID STATE RECTIFIER/TRANSFORMER MODEL
C
C                 METHOD            COMPUTE OUTPUT DC POWER AS A FUNCTION
C                                   OF INPUT AC POWER
C
C                 WRITTEN BY Y.K.CHAN           VERSION 1, JUNE 1, 1977
C
C     CALL SEQUENCE
C         OUTPUTS
C                 P2   -DC OUTPUT POWER, KW
C                 IAC  -AC INPUT CURRENT, AMPS
C                 PL   -POWER LOSS, KW
C                 EF2  -OUTPUT PRODUCT EFFICIENCY
C                 MP2  -MAXIMUM OUTPUT POWER, KW
C         INPUTS
C                 P1   -AC INPUT POWER, KW
C                 RT   -TRANSFORMWR RESISTANCE, OHMS
C                 XT   -TRANSFORMER REACTANCE, OHMS
C                 VAC  -RATED AC VOLTAGE, VOLTS
C                 DR   -RECTIFIER CONTACT POTENTIAL, VOLTS
C                 RR   -RECTIFIER RESISTANCE, OHMS
C                 RAP  -RATED INPUT POWER, KW
C                 EF1  -INPUT PRODUCT EFFICIENCY
C                 MP1  -MAXIMUM INPUT POWER, KW
C                 CC   -RECTIFIER COST/YEAR, $
C
      COMMON /CIMPL/IMPL,ICNT/CTIME/TIME/CSIMUL/DUM(7),TMAX/COST/CC1
      REAL IAC,MP2,MP1,IDC
      DATA PI/3.14159/
      DATA ROOT3/1.73205/
C
      IF(IMPL.GT.0) GO TO 100
      IF(MP1.EQ..99999)MP1=1.E8
      IF(RT.EQ..99999) RT=0.
      IF(XT.EQ..99999) XT=.03
      IF(VAC.EQ..99999) VAC=440.
      IF(DR.EQ..99999) DR=0.
      IF(RR.EQ..99999) RR=.02
      TMAX1=TMAX*.99999
C
C                 COMPUTE TRANSFORMER POWER ANGLES
C
  100 Y=ROOT3*XT*P1*1000./(VAC*VAC)
      YY=Y*Y
      IF(YY.LE.1.)GO TO 200
      IF(IMPL.EQ.2)WRITE(6,108)P1,XT,VAC
  108 FORMAT(1H0,19HRE, AC INPUT POWER ,F12.3,49H TOO LARGE IN RELATION TO
     1TO TRANSFORMER REACTANCE ,F12.3,22H AND RATED AC VOLTAGE  ,F12.3)
      IF(IMPL.EQ.2)ICNT=ICNT+1
  200 YY=AMIN1(1.,YY)
C
C                 INPUT AND OUTPUT CURRENT
C
      IF(P1.GT.0.)GO TO 300
      P2=0.
      IAC=0.
      PL=0.
      EFF=1.
      GO TO 400
C
  300 IAC=VAC*SQRT(2.-2.*SQRT(1.-YY))/(ROOT3*XT)
      IDC=PI*IAC/SQRT(6.)
C
C                    POWER LOSS AND OUTPUT POWER
C
      PL=(ROOT3*RT*IAC*IAC+IDC*(DR+IDC*RR))/1000.
      P2=P1-PL
      EFF=P2/P1
      IF(P2.GT.0.) GO TO 400
      IF(IMPL.EQ.2)WRITE(6,308)PL,P1
  308 FORMAT(1H0,11HPOWER LOSS  ,F12.3,23HRE EXCEEDS INPUT POWER ,F12.3,
     156H CHECK RATED AC VOLTAGE VAC AND TRANSFORMER REACTANCE XT    )
      IF(IMPL.EQ.2)ICNT=ICNT+1
      P2=0.
      EFF=1.
C
C                    EFFICIENCY AND MAXIMUM POWER
C
  400 EF2=EF1*EFF
      MP2=AMIN1(MP1,RAP)
      MP2=MP2*EFF
      IF(IMPL.LE.1)RETURN
      IF(TIME.LT.TMAX1)RETURN
      CCI=CCI+CC
C
      RETURN
      END
```

7.36 RANDOM NUMBERS



This component generates an uncorrelated sequence of normally distributed random numbers with a specified mean and standard deviation.

Inputs

Parameter/Port	Description
MN	Mean value of sequence
SIG	Standard deviation of sequence
NST[1]	Start parameter. (Use any odd integer greater than 1). Default supplied.

Outputs

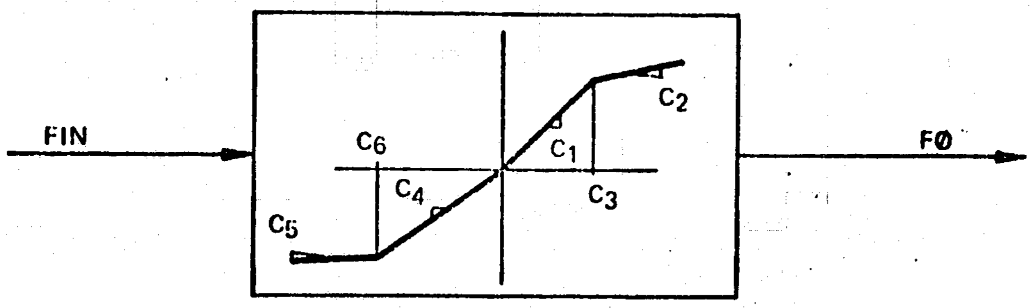
Variable/Port	
FØ	Random number output

[1] If RESET parameter > 0 then succeeding simulations use NST to start random sequence.

RN

```
CRN
      SUBROUTINE RN(U,AX,SIG,AMN)
C     VERSION 2.                        REVISED  MAY 1977
C     PURPOSE -  GENERATES A NORMALLY DISTRIBUTED RANDOM NUMBER
C     CALL SEQUENCE
C
C                    U - RANDOM NUMBER OUTPUT
C                    AX- A START PARAMETER WHICH CONTROLS THE BEGINNING POINT
C                        OF THE OUTPUT SEQUENCE.  AX SHOULD BE ANY ODD INTEGER
C                        GREATER THAN ONE.  THE DEFAULT VALUE OF AX IS 431469.
C                        AX IS UPDATED FOR NEW CALLS TO THE SUBROUTINE.
C                   SIG- THE DESIRED STANDARD DEVIATION OF THE SEQUENCE
C                   AMN- THE DESIRED MEAN OF THE SEQUENCE
C     DESIGNED BY          ROGER W. CALL                 SEPT 1976
      COMMON /CIMPL/IMPL,ICNT,ITEST
      DATA Y /253967./,AXO/0./
      IF(IMPL.GT.0)GO TO 5
      IF(AX.EQ..99999) AX=431469.
      IF(AXO.EQ.0.)AXO=AX
      IF(ITEST.EQ.1)AX=AXO
    5 X =AX
      SUM=0.
      DO 1 I=1,12
      X= AMOD(X*Y,16777216.)
      SUM= SUM+ X/16777215.
    1 AX= X
      U=(SUM-6.0)*SIG+AMN
      RETURN
      END
```

7.37 SATURATION FUNCTION



Inputs

Parameter/Port	Description
FIN	Input quantity
C1	Slope 0 < FIN < C3
C2	Slope FIN > C3
C3	Positive saturation intercept
C4	Slope 0 > FIN > C6
C5	Slope FIN < C6
C6	Negative saturation intercept

Outputs

Variable/Port	
FØ	Output quantity

Calculation Sequence

FØ = C1*C3 + C2*(FIN−C3) if FIN > C3

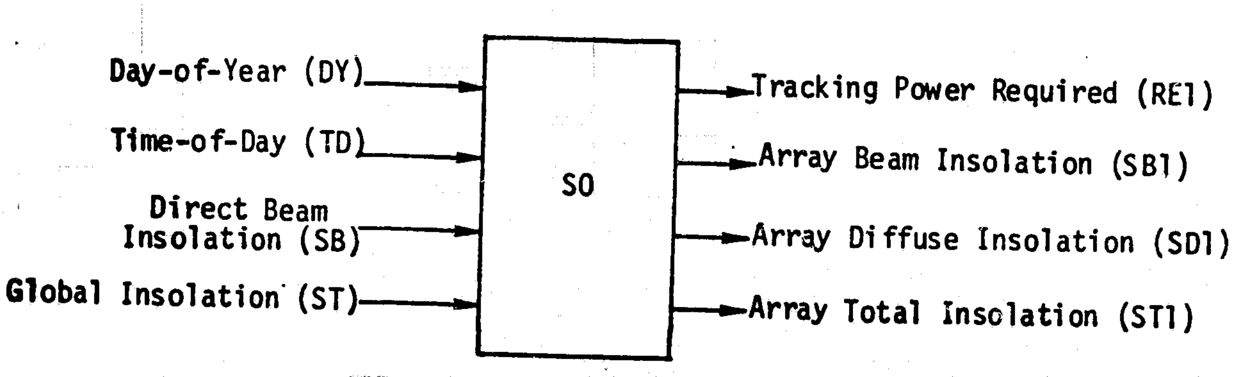
FØ = C1*FIN if 0 < FIN < C3

FØ ≠ C4*FIN if 0 > FIN > C6

FØ = C4*C6 + C5*(FIN−C6) if FIN < C6

```
CSA
      SUBROUTINE SA(FO,FIN,C1,C2,C3,C4,C5,C6)
C
C    PURPOSE - TO SIMULATE SATURATION
C
C
C    METHOD - SEE CODING. C3 AND C6 ARE VALUES OF THE INPUT AT WHICH
C             SATURATION OCCURS.  C3 IS GREATER THAN C6. THE ROUTINE
C             CAN SIMULATE A CHANGE OF SLOPE AT THE ORIGIN (C1.NE.C4)
C             PROVIDED C6 IS LESS THAN ZERO. SIMILARLY THE SLOPES
C             IN THE SATURATION REGION (C2 AND C5)CAN DIFFER.
C             THE SLOPES CAN BE POSITIVE OR NEGATIVE
C
C
C    WRITTEN BY  -  ADAM LLOYD                     LATEST REVISION - NOV 75
C
C    LIMITATIONS - USE OF ZERO SLOPES (C2=0 OR C5=0) IN THE SATURATION
C                  REGION SHOULD BE AVOIDED. IT IS DESIRABLE THAT THE
C                  SLOPE RATIOS C1/C2 AND C4/C5 SHOULD NOT EXCEED 100.
C                  EXCESSIVE SLOPE RATIOS MAY RESULT IN VERY SLOW
C                  CONVERGENCE
C
C
C    INPUT/OUTPUT LIST
C
C    FO          OUTPUT  VARIABLE                     ANY         OUTPUT  VAR
C    FIN         INPUT   VARIABLE                     ANY         INPUT   VAR
C    C1          SLOPE                 ) FIRST        ANY         INPUT   PARAM
C    C2          SATURATION SLOPE      ) SLOPE        ANY         INPUT   PARAM
C    C3          SATURATION INTERCEPT)                ANY         INPUT   PARAM
C    C4          SLOPE                 ) SECOND       ANY         INPUT   PARAM
C    C5          SATURATION SLOPE      ) SLOPE        ANY         INPUT   PARAM
C    C6          SATURATION INTERCEPT)                ANY         INPUT   PARAM
C
      IF(FIN.GT.C3)GO TO 10
      IF(FIN.LT.C6)GO TO 20
      IF(FIN.LT.0.)GO TO 30
      FO=C1*FIN
      GO TO 100
C   POSITIVE SATURATION
 10   FO=C1*C3+C2*(FIN-C3)
      GO TO 100
C   NEGATIVE SATURATION
 20   FO=C4*C6+C5*(FIN-C6)
      GO TO 100
C   NEGATIVE UNSATURATED
 30   FO=C4*FIN
 100  RETURN
      END
```

7.38 SOLAR ORIENTATION


Day-of-Year (DY)
Time-of-Day (TD)
Direct Beam Insolation (SB)
Global Insolation (ST)
SO
Tracking Power Required (RE1)
Array Beam Insolation (SB1)
Array Diffuse Insolation (SD1)
Array Total Insolation (ST1)

The Solar Orientation model computes flat plate collector insolation for five types of solar tracking:

- Tilted orientation, facing south
- Tracking about a horizontal EW axis
- Tracking about a horizontal NS axis
- Tilted, tracking about a vertical axis
- Two axis tracking

Array insolation is the sum of beam and diffuse components. The beam component is the product of normal incidence radiation and a geometry-dependent incidence factor. The diffuse component is approximated as the product of horizontal diffuse insolation times a geometry factor plus ground reflectance.

BASIC EQUATIONS

$$\begin{aligned} ST1 &= SB1 + SD1 + SR1 \\ &= SB*IF + SD*RD + ST*RR \\ SD &= ST - SB*SIN(EL) \\ RD &= .5*(1 + COS(TLT)) \\ RR &= .5*PR*(1 - COS(TLT)), \end{aligned}$$

where

IF = solar incidence factor (incidence angle cosine)

TLT = collector tilt angle from horizontal

PR = ground reflectance

Inputs/Port	Description	Units
LA	Collector latitude*	Deg
DY	Day-of-the-year (1-365)	-
TD	Time-of-day (0-24)	hr
MØ	Tracking mode 1 = fixed orientation and tilt (default) 2 = horizontal EW axis tracking 3 = horizontal NS axis tracking 4 = tilted, vertical axis tracking 5 = two axis tracking	-
TL	Collector tilt (MØ = 1, 4 inputs)	Deg
SB	Direct normal beam insolation	w/m^2
ST	Global insolation on a horizontal surface	w/m^2
PR	Ground reflectance (default = 0.2)	-

*For TMY stations, see Table 7.8 of the Environmental Data Component ED.

Inputs/Port (cont'd)		Description	Units
AA		Collector array area	m^2
SBT		Insolation threshold for tracking (default = 100.)	w/m^2

Outputs/Port		Description	Units
SE		SIN (Solar Elevation Angle)*	-
SA		SIN (Solar Azimuth Angle)*	-
IF		COS (Solar Incidence Angle)	-
RE	1	Tracking power required	kw
SB	1	Collector beam insolation	w/m^2
SD	1	Collector diffuse insolation	w/m^2
SR	1	Collector reflected insolation	w/m^2
ST	1	Collector total insolation	w/m^2
TLT		Collector tilt angle	Deg

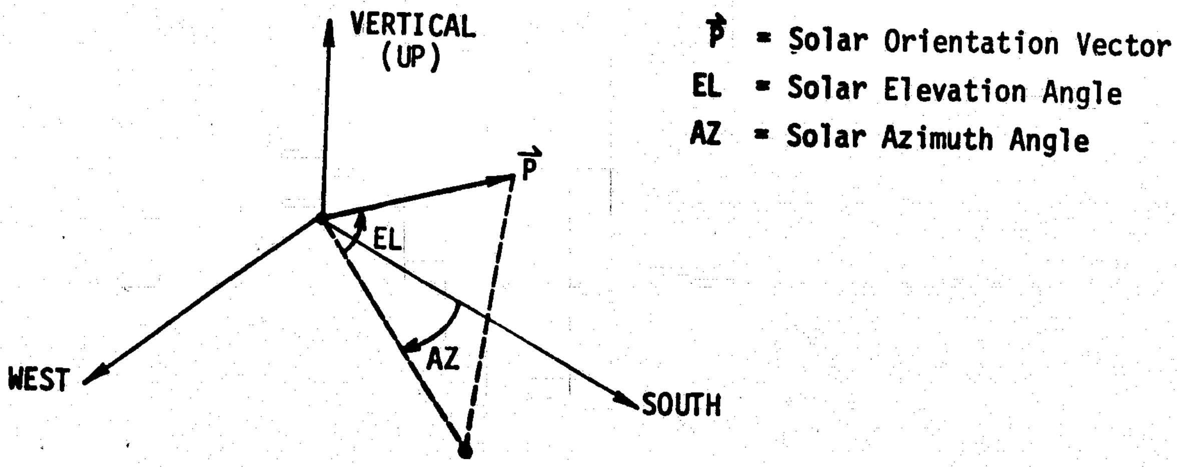

VERTICAL (UP)
$\vec{P}$
EL
AZ
WEST
SOUTH
$\vec{P}$ = Solar Orientation Vector
EL = Solar Elevation Angle
AZ = Solar Azimuth Angle

*FIGURE 7.38 SOLAR ORIENTATION ANGLES

CALCULATION SEQUENCE

```
            $RPD = \pi/180$
If $SB \leq 0$ and $MØ > 1$ return

1) Solar azimuth and elevation
            $W = 15*(12 - TD)*RPD$
            $\delta = 23.45*SIN(2\pi *(284 + DY)/365)*RPD$
            $LA' = LA*RPD$
            $SE = SIN\ \delta *SIN\ LA' + COS\ \delta *COS\ W*COS\ LA'$
            $CE = (1. - SE*SE)^{1/2}$
            $TAN(AZ) = COS\ \delta *SIN\ W/(COS\ W*SIN\ LA'*COS\ \delta - SIN\ \delta *COS\ LA')$
            $CA = 1/(1 + TAN^2(AZ))^{1/2}$
            $SA = TAN(AZ)*CA$

2) Horizontal diffuse insolation
            $SD = ST - SB*SE$

3) Array geometry and tracking power
            $RE1 = 0$
    If $MØ = 1$ then
            $TLT' = TL*RPD$
            $IF = SIN\ TLT'*CE*CA + COS\ TLT'*SE$
    If $MØ = 2$ then
            $IF = \sqrt{1. - (CE*SA)^2}$
            $TLT' = MIN(COS^{-1}(SE/IF),\ \pi/2)$
            $RE1 = 3.75\ E\text{-}4*AA$                    if $SB > SBT$
```

CALCULATIONS (contd)

If MØ = 3 then

$IF = \sqrt{1. - (CE*CA)^2}$

$TLT' = MIN(COS^{-1}(SE/IF),\ \pi/2)$

$RE1 = 3.75\ E\text{-}4*AA$ if $SB > SBT$

If MØ = 4 then

$TLT' = TL*RPD$

$IF = SIN\ TLT'*CE + COS\ TLT'*SE$

$RE1 = 3.75\ E\text{-}4*AA$ if $SB > SBT$

If MØ = 5 then

$IF = 1$

$TLT' = MIN(COS^{-1}(SE),\ \pi/2)$

$RE1 = 5.E\text{-}4*AA$ if $SB > SBT$

4) Insolation components

$SB1 = SB*IF$

$SD1 = SD*.5*(1 + COS(TLT'))$

$SR1 = ST*.5*PR*(1 - COS(TLT'))$

$ST1 = SB1 + SD1 + SR1$

5) Tilt

$TLT = TLT'/RPD$

REFERENCES FOR SO

1. J. K. Linn, "Photovoltaic System Analysis Program-SOLCEL," Sandia Laboratories Report SAND77-1268, August 1977.

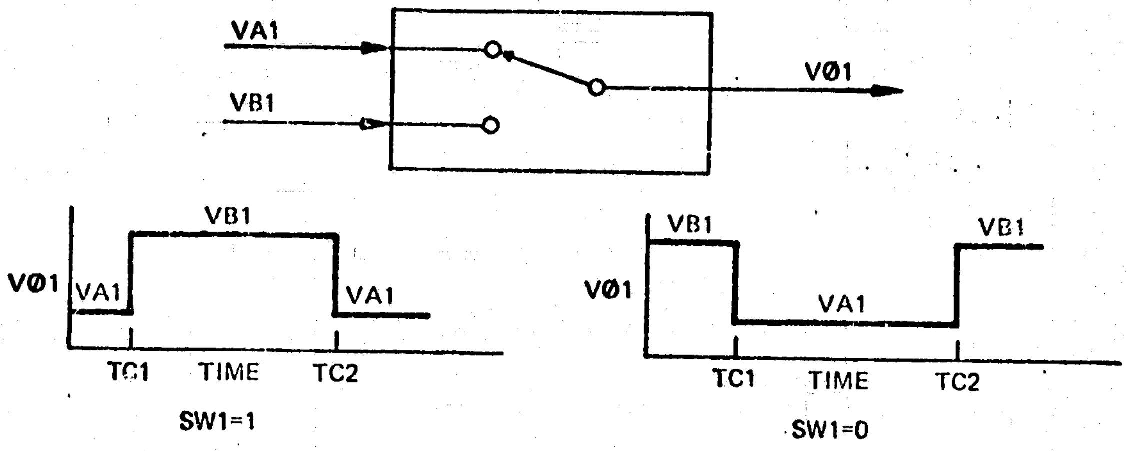
2. B. Y. Liu and R. C. Jordan, "The Interrelationship and Characteristic Distribution of Direct, Diffuse and Total Solar Radiation," Solar Energy, Vol. IV, July 1960, pp. 1-19.

3. J. A. Duffie and W. A. Beckman, *Solar Thermal Processes* (Chapter 2), Wiley, 1974.

```
CSO
      SUBROUTINE SO(SE,SA,IF,RE1,SB1,SD1,SR1,ST1,TLT,
     1  LA,DY,TD,MO,TL,SB,ST,PR,AA,SBT)
C
C        PURPOSE   THIS COMPONENT COMPUTES FLAT PLATE COLLECTOR
C                  INSOLATION FOR FIVE MODES OF SOLAR TRACKING
C                     TILTED ORIENTATION, FACING SOUTH
C                     TRACKING ABOUT A HORIZONTAL EW AXIS
C                     TRACKING ABOUT A HORIZONTAL NS AXIS
C                     TILTED, TRACKING ABOUT THE VERTICAL AXIS
C                     TWO AXIS TRACKING
C
C        WRITTEN BY Y.K.CHAN, 11-6-78, VERSION 1
C
C        METHOD   ARRAY INSOLATION IS SUM OF BEAM AND DIFFUSE
C                 COMPONENTS. THE BEAM COMPONENT IS THE PRODUCT OF
C                 NORMAL INCIDENCE INSOLATION AND A GEOMETRY DEPENDENT
C                 INCIDENCE FACTOR.  THE  DIFFUSE COMPONENT IS
C                 APPROXIMATED AS THE PRODUCT OF HORIZONTAL DIFFUSE
C                 INSOLATION TIMES A GEOMETRY FACTOR PLUS GROUND REFLECTANCE.
C
C        CALLING SEQUENCE
C          OUTPUTS
C            SE    -SINE OF SOLAR ELEVATION ANGLE
C            SA    -SINE OF SOLAR AZIMUTH ANGLE
C            IF    -COSINE OF SOLAR INCIDENCE ANGLE
C            RE1   -TRACKING POWER REQUIRED,KW
C            SB1   -COLLECTOR BEAM INSOLATION,W/M2
C            SD1   -COLLECTOR DIFFUSE INSOLATION,W/M2
C            SR1   -COLLECTOR REFLECTED INSOLATION,W/M2
C            ST1   -COLLECTOR TOTAL INSOLATION,W/M2
C            TLT   -COLLECTOR TILE ANGLE,DEGREES
C          INPUTS
C            LA    -COLLECTOR LATITUDE,DEGREES
C            DY    -DAY OF YEAR(1-365)
C            TD    -TIME OF DAY(0-24),HOUR
C            MO    -TRACKING MODE
C                      1=FIXED ORIENTATION AND TILT (DEFAULT)
C                      2=HORIZONTAL EW AXIS TRACKING
C                      3=HORIZONTAL NS AXIS TRACKING
C                      4=TILTED,VERTICAL AXIS TRACKING
C                      5=TWO AXIS TRACKIGN
C            TL    -COLLECTOR TILT (MO=1,4  INPUTS),DEGREES
C            SB    -DIRECT NORMAL BEAM INSOLATION,W/M2
C            ST    -GLOBAL INSOLATION ON A HORIZONTAL SURFACE,W/M2
C            PR    -GROUND REFLECTANCE (DEFAULT=0.2)
C            AA    -COLLECTOR ARRAY AREA,M2
C            SBT   -INSOLATION THRESHOLD FOR TRACKING,W/M2
C                   (DEFAULT=100)
C
      COMMON /CIMPL/IMPL
      REAL IF,LA,MO
      IF(IMPL.NE.0)GO TO 100
      IF(MO.EQ..99999)MO=1.
      IF(PR.EQ..99999)PR=.2
      IF(SBT.EQ..99999)SBT=100
      RPD=3.1415926/180.
  100 CONTINUE
```

```
      IF((SB.GT.0.).OR.(MO.LT.2.))GO TO 109
      SA=0.
      IF=0.
      RE1=0.
      SB1=0.
      SD1=0.
      SR1=0.
      ST1=0.
      RETURN
  109 CONTINUE
      RE1=0.
C
C       SOLAR AZIMUTH AND ELEVATION
C
      W=15.*(12.-TD)*RPD
      ADEL=23.45*SIN(.0172142*(284+DY))*RPD
      PLA=LA*RPD
      CLAP=COS(PLA)
      SADEL=SIN(ADEL)
      CADEL=COS(ADEL)
      SINPLA=SIN(PLA)
      COSW=COS(W)
      SE=SADEL*SINPLA+CADEL*COSW*CLAP
      CE=SQRT(1.-SE*SE)
      F=CADEL*COSW*SINPLA-SADEL*CLAP
      CA=0.
      SA=1.
      IF(ABS(F).LE.1.E-5)GO TO 200
      TAZ=CADEL*SIN(W)/F
      CA=1./SQRT(1.+TAZ*TAZ)
      SA=TAZ*CA
  200 CONTINUE
C
C       HORIZONTAL DIFFUSE INSOLATION
C
      SD=ST-SB*SE
C
C       ARRAY GEOMETRY AND TRACKING POWER
C
      IMO=MO+.1
      GO TO(301,302,303,304,305)IMO
  301 TLTP=TL*RPD
      IF=SIN(TLTP)*CE*CA+COS(TLTP)*SE
      GO TO 309
  302 IF=SQRT(1.-CE*CE*SA*SA)
      BIF=AMIN1(1.,SE/IF)
      TLTP=1.5708
      IF(BIF.GT.0.)TLTP=ACOS(BIF)
      IF(SB.GT.SBT)RE1=3.75E-4*AA
      GO TO 309
  303 IF=SQRT(1.-CE*CE*CA*CA)
      BIF=AMIN1(1.,SE/IF)
      TLTP=1.5708
      IF(BIF.GT.0.)TLTP=ACOS(BIF)
      IF(SB.GT.SBT)RE1=3.75E-4*AA
      GO TO 309
  304 TLTP=TL*RPD
      IF=SIN(TLTP)*CE+COS(TLTP)*SE
      IF(SB.GT.SBT)RE1=3.75E-4*AA
      GO TO 309
  305 IF=1.
      SE1=AMIN1(SE,1.)
      TLTP=1.5708
      IF(SE1.GT.0.)TLTP=ACOS(SE1)
      IF(SB.GT.SBT)RE1=5.E-4*AA
C
  309 CONTINUE
C
C              INSOLATION COMPONENTS
C
      SB1=SB*IF
      SD1=SD*.5*(1.+COS(TLTP))
      SR1=ST*.5*PR*(1.-COS(TLTP))
      ST1=SB1+SD1+SR1
C
C               TILT
C
      TLT=TLTP/RPD
      RETURN
      END
```

7.39 SINGLE POLE SWITCH



THE SWITCHING OPERATION MAY BE CONTROLLED BY EITHER TIME OR THE INPUT PARAMETER SW1. THE TIME DEPENDENCE MAY BE ELIMINATED BY SETTING TC1 = 10^{36}

Inputs

Parameter/Port	Description
VA1	Input to switch
VB1	Input to switch
SW1	Switch control parameter
TC1	Time for first switching (hours)
TC2	Time for second switching (hours)

Outputs

Variable/Port	
VØ1	Switch output

Calculation Sequence

If SW1 = 0 then

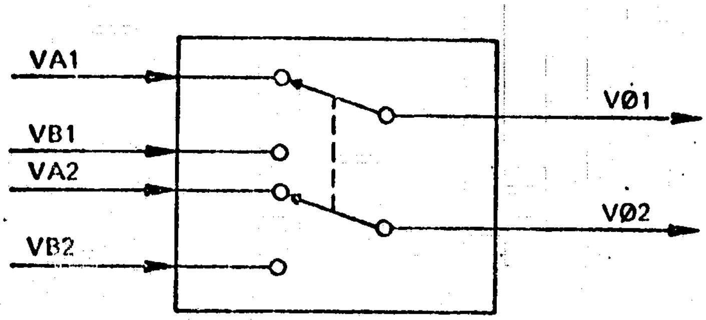
$$VØ1 = \left(\begin{array}{ll} VA1 & TC1 < TIME < TC2 \\ VB1 & otherwise \end{array} \right)$$

If SW1 = 1 then

$$VØ1 = \left(\begin{array}{ll} VB1 & TC1 < TIME < TC2 \\ VA1 & otherwise \end{array} \right)$$

```
CSW
      SUBROUTINE SW(VO1,VA1,VB1,SW1,TC1,TC2)
C
C     PURPOSE - TO PROVIDE SWITCH CONTROL FOR ONE VARIABLE
C
C
C     METHOD  - SEE CODING
C
C
C     WRITTEN BY  - ADAM LLOYD              LATEST REVISION     NOV 75
C
C
C     LIMITATIONS - NOT MORE THAN TWO SWITCHINGS AT TIMES TC1 AND TC2
C
C
C     INPUT/OUTPUT LIST
C
C     VO1        OUTPUT VARIABLE NO 1                 ANY     OUTPUT VAR
C     VA1        INPUT VARIABLE NO A1                 ANY     INPUT  VAR
C     VB1        INPUT VARIABLE NO B1                 ANY     INPUT  VAR
C     SW1        SWITCH CONTROL INITIAL VALUE         ----    INPUT  PARAM
C                    =1.   VO=VB
C                    =0.   VO=VA
C     TC1        TIME FOR FIRST SWITCH                SECS    INPUT  PARAM
C     TC2        TIME FOR SECOND SWITCH               SECS    INPUT  PARAM
C                    (TC2.GT.TC1)
      COMMON/CTIME/TIME
      COMMON/CIO/IREAD,IWRITE,IDIAG
      SX=SW1
      IF(TIME.GT.TC1.AND.TIME.LT.TC2)SX=ABS(SW1-1.)
      VO1=VA1
      IF(SX.GT.0.5)VO1=VB1
      RETURN
      END
```

7.40 TWO POLE SWITCH



SEE SW FOR SWITCH CONTROL LOGIC

Inputs

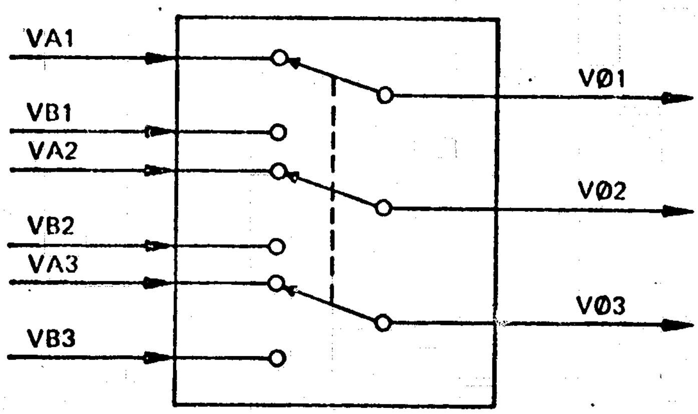
Parameter/Port	Description
VA1	Input to switch 1
VA2	Input to switch 2
VB1	Input to switch 1
VB2	Input to switch 2
SW1	Switch control parameter
TC1	Time for first switching (hours)
TC2	Time for second switching (hours)

Outputs

Variable/Port	
VØ1	Output from switch 1
VØ2	Output from switch 2

```
CSX
      SUBROUTINE SX(VO1,VO2,VA1,VA2,VB1,VB2,SW1,TC1,TC2)
C
C     PURPOSE - TO PROVIDE A SWITCH COMPONENT FOR TWO VARIABLES
C
C
C     METHOD   - SEE CODING
C
C
C     WRITTEN BY  - ADAM LLOYD                LATEST REVISION     NOV 75
C
C
C     LIMITATIONS - NOT MORE THAN TWO SWITCHINGS AT TIMES TC1 AND TC2
C
C
C     INPUT/OUTPUT LIST
C
C     VO1         OUTPUT VARIABLE NO 1             ANY        OUTPUT VAR
C     VO2         OUTPUT VARIABLE NO 2             ANY        OUTPUT VAR
C     VA1         INPUT VARIABLE NO A1             ANY        INPUT  VAR
C     VA2         INPUT VARIABLE NO A2             ANY        INPUT  VAR
C     VB1         INPUT VARIABLE NO B1             ANY        INPUT  VAR
C     VB2         INPUT VARIABLE NO B2             ANY        INPUT  VAR
C     SW1         SWITCH CONTROL INITIAL VALUE     ---        INPUT  PARAM
C                    =1.   VO=VB
C                    =0.   VO=VA
C     TC1         TIME FOR FIRST SWITCH            SECS       INPUT  PARAM
C     TC2         TIME FOR SECOND SWITCH           SECS       INPUT  PARAM
C                    (TC2.GT.TC1)
      COMMON/CTIME/TIME
      COMMON/CIO/IREAD,IWRITE,IDIAG
      SW=SW1
      IF(TIME.GT.TC1.AND.TIME.LT.TC2)SW=ABS(SW1-1.)
      VO1=VA1
      VO2=VA2
      IF(SW.GT.0.5)VO1=VB1
      IF(SW.GT.0.5)VO2=VB2
      RETURN
      END
```

7.41 THREE POLE SWITCH



SEE SW FOR SWITCH CONTROL LOGIC

Inputs

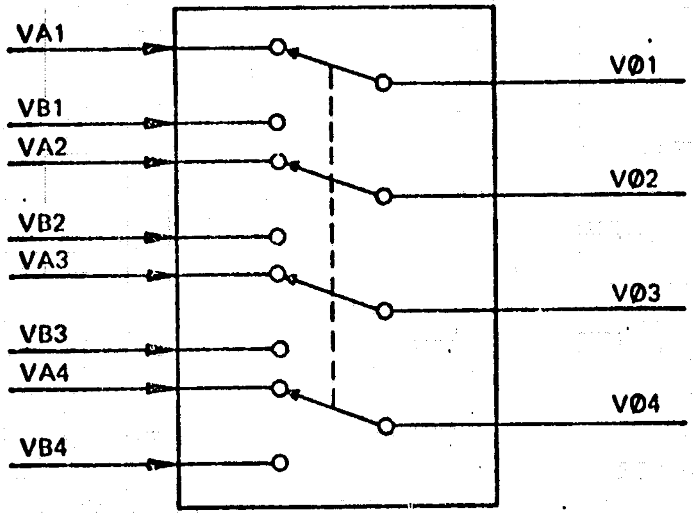
Parameter/Port	Description
VA1	Input to switch 1
VA2	Input to switch 2
VA3	Input to switch 3
VB1	Input to switch 1
VB2	Input to switch 2
VB3	Input to switch 3
SW1	Switch control parameter
TC1	Time for first switching (hours)
TC2	Time for second switching (hours)

Outputs

Variable/Port	
VØ1	Output from switch 1
VØ2	Output from switch 2
VØ3	Output from switch 3

```
CSY
      SUBROUTINE SY(VO1,VO2,VO3,VA1,VA2,VA3,VB1,VB2,VB3,SW1,TC1,TC2)
C
C    PURPOSE - TO PROVIDE A SWITCH COMPONENT FOR THREE VARIABLES
C
C
C    METHOD  - SEE CODING
C
C
C    WRITTEN BY  - ADAM LLOYD            LATEST REVISION     NOV 75
C
C
C    LIMITATIONS - NOT MORE THAN TWO SWITCHINGS AT TIMES TC1 AND TC2
C
C
C    INPUT/OUTPUT LIST
C
C    VO1        OUTPUT VARIABLE NO 1               ANY       OUTPUT VAR
C    VO2        OUTPUT VARIABLE NO 2               ANY       OUTPUT VAR
C    VO3        OUTPUT VARIABLE NO 3               ANY       OUTPUT VAR
C    VA1        INPUT VARIABLE NO A1               ANY       INPUT  VAR
C    VA2        INPUT VARIABLE NO A2               ANY       INPUT  VAR
C    VA3        INPUT VARIABLE NO A3               ANY       INPUT  VAR
C    VB1        INPUT VARIABLE NO B1               ANY       INPUT  VAR
C    VB2        INPUT VARIABLE NO B2               ANY       INPUT  VAR
C    VB3        INPUT VARIABLE NO B3               ANY       INPUT  VAR
C    SW1        SWITCH CONTROL INITIAL VALUE       ---       INPUT  PARAM
C                   =1.   VO=VB
C                   =0.   VO=VA
C    TC1        TIME FOR FIRST SWITCH              SECS      INPUT  PARAM
C    TC2        TIME FOR SECOND SWITCH             SECS      INPUT  PARAM
C                   (TC2.GT.TC1)
      COMMON/CTIME/TIME
      COMMON/CIO/IREAD,IWRITE,IDIAG
      SW=SW1
      VO1=VA1
      VO2=VA2
      VO3=VA3
      IF(TIME.GT.TC1.AND.TIME.LT.TC2)SW=ABS(SW1-1.)
      IF(SW.GT.0.5)VO1=VB1
      IF(SW.GT.0.5)VO2=VB2
      IF(SW.GT.0.5)VO3=VB3
      RETURN
      END
```

7.42 FOUR POLE SWITCH



SEE SW FOR SWITCH CONTROL LOGIC

Inputs

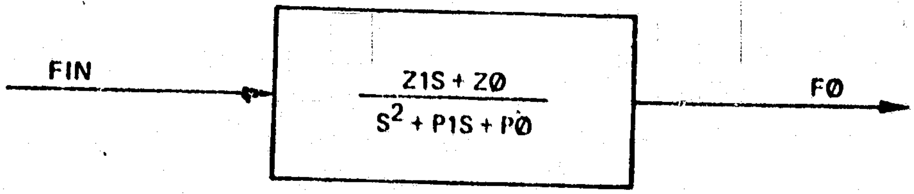
Parameter/Port	Description
VA1	Input to switch 1
VA2	Input to switch 2
VA3	Input to switch 3
VA4	Input to switch 4
VB1	Input to switch 1
VB2	Input to switch 2
VB3	Input to switch 3
VB4	Input to switch 4
SW1	Switch control parameter
TC1	Time for first switching (hours)
TC2	Time for second switching (hours)

Outputs

Variable/Port	
VØ1	Output from switch 1
VØ2	Output from switch 2
VØ3	Output from switch 3
VØ4	Output from switch 4

```
CSZ
      SUBROUTINE SZ(VO1,VO2,VO3,VO4,VA1,VA2,VA3,VA4,VB1,VB2,VB3,VB4,
     1 SW1,TC1,TC2)
C
C    PURPOSE - TO PROVIDE A SWITCH COMPONENT FOR FOUR VARIABLES
C
C
C    METHOD  - SEE CODING
C
C
C    WRITTEN BY  - ADAM LLOYD             LATEST REVISION     NOV 75
C
C
C    LIMITATIONS - NOT MORE THAN TWO SWITCHINGS AT TIMES TC1 AND TC2
C
C
C    INPUT/OUTPUT LIST
C
C    VO1         OUTPUT VARIABLE NO 1                ANY       OUTPUT VAR
C    VO2         OUTPUT VARIABLE NO 2                ANY       OUTPUT VAR
C    VO3         OUTPUT VARIABLE NO 3                ANY       OUTPUT VAR
C    VO4         OUTPUT VARIABLE NO 4                ANY       OUTPUT VAR
C    VA1         INPUT VARIABLE NO A1                ANY       INPUT  VAR
C    VA2         INPUT VARIABLE NO A2                ANY       INPUT  VAR
C    VA3         INPUT VARIABLE NO A3                ANY       INPUT  VAR
C    VA4         INPUT VARIABLE NO A4                ANY       INPUT  VAR
C    VB1         INPUT VARIABLE NO B1                ANY       INPUT  VAR
C    VB2         INPUT VARIABLE NO B2                ANY       INPUT  VAR
C    VB3         INPUT VARIABLE NO B3                ANY       INPUT  VAR
C    VB4         INPUT VARIABLE NO B4                ANY       INPUT  VAR
C    SW1         SWITCH CONTROL INITIAL VALUE        ---       INPUT  PARAM
C                    =1.   VO=VB
C                    =0.   VO=VA
C    TC1         TIME FOR FIRST SWITCH               SECS      INPUT  PARAM
C    TC2         TIME FOR SECOND SWITCH              SECS      INPUT  PARAM
C                    (TC2.GT.TC1)
      COMMON/CTIME/TIME
      COMMON/CIO/IREAD,IWRITE,IDIAG
      SW=SW1
      IF(TIME.GT.TC1.AND.TIME.LT.TC2)SW=ABS(SW1-1.)
      VO1=VA1
      VO2=VA2
      VO3=VA3
      VO4=VA4
      IF(SW.GT.0.5)VO1=VB1
      IF(SW.GT.0.5)VO2=VB2
      IF(SW.GT.0.5)VO3=VB3
      IF(SW.GT.0.5)VO4=VB4
      RETURN
      END
```

7.43 SECOND ORDER TRANSFER FUNCTION



Inputs

Parameter/Port	Description
FIN	Input quantity
ZØ	Numerator coefficient
Z1	Numerator coefficient
PØ	Denominator coefficient
P1	Denominator coefficient

Outputs

Variable/Port	
X1	Intermediate state
FØ	Output quantity (state)

Calculation Sequence

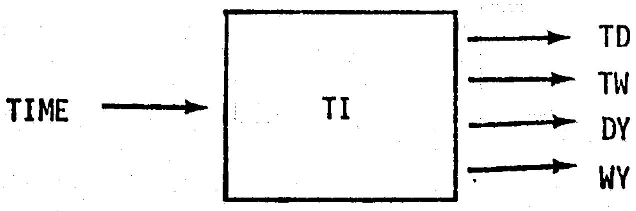
$$\dot{X1} = Z\emptyset{*}FIN - P\emptyset{*}F\emptyset$$

$$\dot{F\emptyset} = X1 + Z1{*}FIN - P1{*}F\emptyset$$

NOTE: d.c. gain = $\frac{Z\emptyset}{P\emptyset}$; infinite frequency gain = 0.

```
CTF
      SUBROUTINE TF(X1,X1DOT,IX1,FO,FODOT,IFO,FIN,ZO,Z1,PO,P1)
C
C   PURPOSE - TO SIMULATE A SECOND ORDER TRANSFER FUNCTION WITH
C             FIRST ORDER NUMERATOR
C
C                              FO       Z1*S + ZO
C                             ---- =  -------------
C                             FIN       2
C                                      S + P1*S+PO
C
C   METHOD  - SELF EXPLANATORY
C
C
C   LIMITATIONS  -  NONE
C
C
C   WRITTEN BY      ADAM LLOYD          LATEST REVISION  NOV 75
C
C
C   INPUT/OUTPUT LIST
C
C   X1          INTERMEDIATE STATE VARIABLE        ANY       OUTPUT STATE
C   X1DOT       STATE VARIABLE DERIVATIVE          ANY       OUTPUT STATE
C   IX1         INTEGRATOR CONTROL                 ---       PROGRAM VAR
C   FO          TRANSFER FUNCTION OUTPUT           ANY       OUTPUT STATE
C   FODOT       TRANSFER FUNCTION OUTPUT DERIV.    ANY       OUTPUT STATE
C   IFO         INTEGRATOR CONTROL                 ---       PROGRAM VAR
C   FIN         TRANSFER FUNCTION INPUT            ANY       INPUT  VAR
C   ZO          NUMERATOR COEFFICIENT              ANY       INPUT  VAR
C   Z1          NUMERATOR COEFFICIENT              ANY       INPUT  VAR
C   PO          DENOMINATOR COEFFICIENT            1/SEC2    INPUT  VAR
C   P1          DENOMINATOR COEFFICIENT            1/SEC     INPUT  VAR
      COMMON/CIO/IREAD,IWRITE,IDIAG
      IF(IX1.NE.0)X1DOT=ZO*FIN-PO*FO
      IF(IFO.NE.0)FODOT=X1+Z1*FIN-P1*FO
      RETURN
      END
```

7.44 TIME CONVERSION



Converts simulation running time in hours to time referenced to start of day and start of week, and computes number of days and weeks elapsed since start of year.

Inputs

Parameter/Port	Description	Units
TO	Initial time of simulation from start of year	hrs
TIME	Running time (input via common/CTIME)	hrs

Outputs

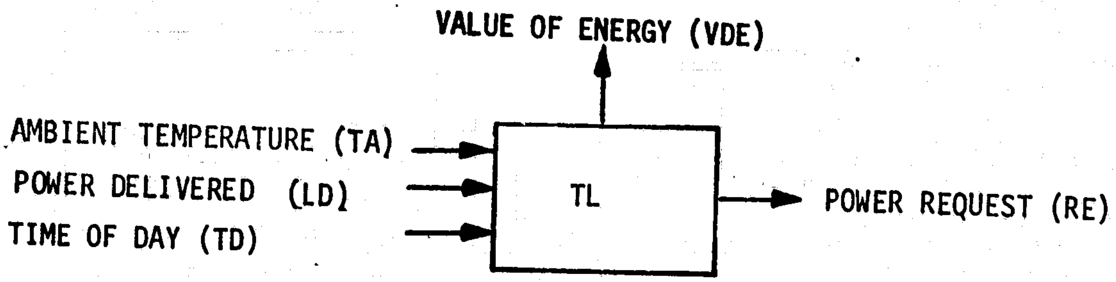
Variable/Port		
TW	Time since start of week	hrs
TD	Time since start of day	hrs
WY	Number of weeks	-
DY	Number of days	-
MY	Number of months (approx.)	-
T	Running time from start of year	hrs
DW	Day of week	-

Calculation Sequence

```
          T = AMØD(TO+TIME,8760)          TW = AMØD(T,168)
          WY = T/168+1                    TD = AMØD(T,24)
          DY = T/24+1                     DW = TW/24+1
          MY = T/730+1
```

```
CTI
      SUBROUTINE TI(T,TD,TW,DW,DY,WY,AMY,T0)
C
C   PURPOSE    CONVERT SIMULATION TIME TO DAILY, WEEKLY, MONTHLY UNITS
C
C   WRITTEN BY  A.W. WARREN                          VERSION 1, MARCH 3 197
C
C   CALL SEQUENCE
C             T    - SIMULATION TIME FROM START OF YEAR, HR          OUTPUT V
C             TD   - TIME OF DAY, HR                                 OUTPUT V
C             TW   - TIME SINCE START OF WEEK, HR                    OUTPUT V
C             DW   - DAY OF WEEK                                     OUTPUT V
C             DY   - DAY OF YEAR                                     OUTPUT V
C             WY   - WEEK OF YEAR                                    OUTPUT V
C             AMY  - MONTH OF YEAR (APPROX.)                         OUTPUT V
C             T0   - SIMULATION INITIAL TIME FROM START OF YEAR, HR  INPUT PA
C
      COMMON / CTIME / TIME
      DATA EPS/ .000001/
      T = AMOD(T0+TIME,8760.+EPS)
      TD = AMOD(T+EPS,24.)
      TW = AMOD(T+EPS,168.)
      ND = TW/24.+1.001
      DW = ND
      ND = T/24.+1.001
      DY = ND
      NW = T/168.+1.001
      WY = NW
      MY = T/730.+1.001
      AMY = MY
C
      RETURN
      END
```

7.45 THERMAL LOAD



Thermal load is computed as a user specified function of ambient temperature and time of day. The actual load delivered is either the load requested or the maximum discharge rate of the thermal storage chamber. The value of the thermal energy delivered and % of total load actually delivered are also computed.

Basic Equation

$$RE = TL\emptyset(TA)*TWT(TD)*NC$$

where

TLØ = Thermal load versus temperature table

TWT = Daily profile weighting function

NC = Normalizing constant

TL

Tables	Description	Units
TLØ	Thermal load versus ambient temperature	kw
TWT	Daily profile weighting function (tabular with time of day)	

Inputs

Parameter/Port		
TA	Ambient temperature	°F
LD	Power delivered	kw
TD	Time of day (0-24)	h
VE	Value of thermal energy	$/kwh
NC	Normalizing constant	

Outputs

Variable/Port		
RE	Load request	kw
VDE	Total value of energy delivered (state)	$

Statistics		
PC	Cumulative percent of load delivered	-
SLD	Total energy delivered	kwh
SRE	Total energy requested	kwh

Calculation Sequence

1) Compute load request

$$RE = TL\emptyset(TA)*TWT(TD)*NC$$

2) Value of energy dynamics

$$\dot{VDE} = LD*VE$$

3) Statistics

$$SLD = SLD+LD*\Delta/2$$

$$SRE = SRE+RE*\Delta/2$$

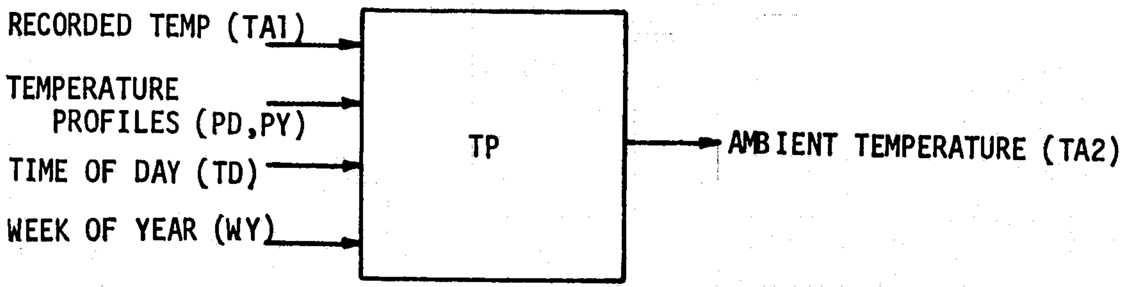
$$PC = 100.*SLD/SRE$$

where Δ = integration step size

```
CTL
      SUBROUTINE TL(TLO,TWT,VDE,DVD,IVD,RE,PC,SLD,SRE,TA,LD,TD,VE,NC)
C
C   PURPOSE    COMPUTE ENERGY RESPONSE FROM A THERMAL LOAD REQUEST
C
C   METHOD     ENERGY DELIVERED IS EQUAL TO THE LOAD REQUESTED OR
C
C              THE MAXIMUM DISCHARGE RATE.
C
C   WRITTEN BY F. O. MAHONY                   VERSION 1, APRIL 1 1977
C
C   CALL SEQUENCE
C        TABLES
C            TLO - THERMAL LOAD AS FUNCTION OF AMBIENT TEMPERATURE
C            TWT - DAILY PROFILE WEIGHTING FUNCTION VS TIME OF DAY
C
C        OUTPUTS
C            VDE - VALUE OF ENERGY DELIVERED (STATE), $
C            DVD - DERIVATIVE OF VDE
C            IVD - INDICATOR FOR VDE
C            RE  - LOAD REQUEST, KW
C            PC  - CUMULATIVE PERCENT OF LOAD DELIVERED
C            SLD - TOTAL ENERGY DELIVERED, KWH
C            SRE - TOTAL ENERGY REQUESTED, KWH
C
C        INPUTS
C            TA  - AMBIENT TEMPERATURE, DEG F
C            LD  - POWER DELIVERED, KW
C            TD  - TIME OF DAY, HR
C            VE  - VALUE OF THERMAL ENERGY, $/KWH
C            NC  - NORMALIZING CONSTANT FOR LOAD REQUEST
C
      DIMENSION TLO(3),TWT(5)
      COMMON/CIMPL/IMPL /CSIMUL/ DUM(6),TINC,TMAX/CTIME/TIME
      COMMON/COST /CC,CM,CO,CV,CLD,CRE
      REAL LD,NC
C
      ITL=TLO(2)
      ITW=TWT(2)
      IF(IMPL.GT.0)GO TO 100
C
C
      TMAX1=TMAX*0.99999
      TINC1=TINC*.5
C
      PC =0.0
      SLD=0.0
      SRE=0.0
C
C
C                  COMPUTE LOAD REQUEST
C
  100 TLD=TBLU1(TA,TLO(4),TLO(ITL+4),1,-ITL)
      TW=TBLU1(TD,TWT(4),TWT(ITW+4),1,-ITW)
C
      RE =TLD*TW*NC
C
C                  VALUE OF ENERGY
```

```
C
      IF(IVD.NE.0)DVD=LD*VE
C
      IF(IMPL.LE.1)RETURN
C
C                        PERFORMANCE STATISTICS
C
      SLD=SLD+LD*TINC1
      SRE=SRE+RE*TINC1
C
      IF(SRE.GT.0.0)PC=100.0*SLD/SRE
C
      IF(TIME.LT.TMAX1)RETURN
C
      CV=CV+VDE
      CLD=CLD+SLD-LD*TINC1
      CRE= CRE+ SRE-RE*TINC1
C
      RETURN
      END
```

7.46 AMBIENT TEMPERATURE



This component is very similar to the wind component. Ambient temperature is output either from user supplied time histories on storage files or by generating a set of random numbers with user specified random variations. If user supplied profiles are available, then the temperatures are generated from the following equation:

$$TA2 = [PD(TD) + CN(t)] * PY(WY)/MØ$$

where PD and PY are the user supplied daily and weekly profiles, TD and WY are the time of the day and week of the year, CN is a colored noise term and MO is the average value of PY:

$$MØ = \frac{1}{J} \sum_{j=1}^{J} PY(j)$$

Tables	Description	Units
PD	Daily profile versus TD	°F
PY	Yearly profile versus WY	arbitrary

Inputs

Parameter/Port			
TA	1	Ambient temperature data file	°F
TD		Time of day	hr
WY		Week of the year	-
CT		Correlation time of colored noise	hr
MN		Mean temperature of colored noise	°F
STD		Standard deviation of colored noise	°F

Outputs

Variable/Port			
CN		Colored noise sample	°F
TA	2	Ambient temperature	°F
AV		Mean of daily temperature	°F
MO		Mean of yearly profile	
TIM		Last time a random sample was generated	hr

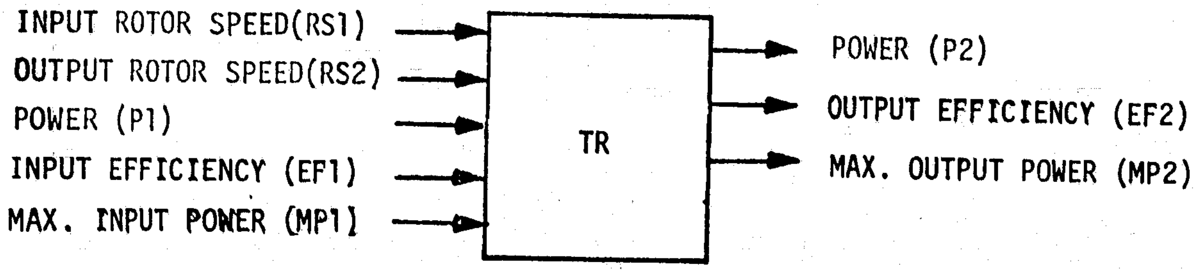
Calculation Sequence

```
1)  Initialization (first pass only)
    Compute AV, MO, and initial CN

        $AV = MN + \frac{1}{N} \sum_{j=1}^{N} PD(j)$

2)  Check for data file input
    If TA1 = .99999 go to 3)
        TA2 = TA1
        Return

3)  Generate colored noise sample CN
    If TIME = TIM  RETURN
        $A = \begin{cases} EXP(-TINC/CT) & CT>0 \\ 0. & CT=0 \end{cases}$
        where TINC = integration step size
        CN = CN * A + W
        Where W is white noise with mean = MN*(1-A) and
        standard deviation = STD* $\sqrt{1-A^2}$
        TIM = TIME

4)  Compute Temperature
        TA2 = (PD(TD) +CN) *PY (WY)/MO
```

```
CTP
      SUBROUTINE TP (PD,PY,TAO,AV,XM,TIMO,XN, TAI, TD,WY,CT,XMN,STD)
C
C   PURPOSE      GENERATE AMBIENT TEMPERATURE FROM DAILY, YEARLY AND RANDOM L
C
C   METHOD       COLORED NOISE WITH SPECIFIED PARMS IS ADDED TO A MEAN DAILY
C                PROFILE AND MULTIPLIED BY A YEARLY PROFILE.
C
C   WRITTEN BY  A.W. WARREN
C                                                    VERSION 1, MARCH 7 197
C   CALL SEQUENCE
C        TABLES
C                PD   - MEAN DAILY PROFILE, DEG.F
C                PY   - MEAN YEARLY PROFILE, DEG.F
C        OUTPUTS
C                TAO  - AMBIENT TEMPERATURE OUTPUT, DEG.F
C                AV   - MEAN DAILY TEMPERATURE, DEG.F
C                XM   - MEAN YEARLY TEMPERATURE,DEG.F
C                TIMO- LAST TIME COLORED NOISE WAS USED, HR
C                XN   - COLORED NOISE SAMPLE, DEG.F
C
C        INPUTS
C                TAI  - TEMPERATURE INPUT FROM DATA FILE,DEG.F
C                TD   - TIME OF DAY, HR
C                WY   - WEEK OF YEAR (1-52)
C                CT   - CORRELATION TIME FOR COLORED NOISE, HR
C                XMN  - MEAN TEMPERATURE OF COLORED NOISE, DEG.F
C                STD  - STANDARD DEVIATION OF COLORED NOISE, DEG.F
C
      DIMENSION PD(1),PY(1)
      COMMON/CIMPL/IMPL /CSIMUL/DUM(6),TINC /CTIME/TIME
      DATA AX /.99999/
C                              INITIALIZATION
C
      ND=PD(2)
      NY=PY(2)
      IF(IMPL.GT.0) GO TO 10
      TIMO=-1.
      CALL RN(XN,AX,STD,XMN)
C
      AV = 0.
      DO 20 I=1,ND
      L = 3+ND+I
   20 AV = AV + PD(L)
      AV = AV/ND +XMN
C
      XM=0.
      DO 30 I=1,NY
      L=3+NY+I
   30 XM=XM+PY(L)
      XM=XM/NY
C                              CHECK FOR DATA FILE INPUT
C
   10 IF(TAI.EQ. .99999) GO TO 100
      TAO = TAI
      GO TO 150
C                              GENERATE COLORED NOISE SAMPLE XN
C
```

```
  100 IF(  TIMO.EQ.TIME) GO TO 150
      A=0.
      IF(CT.GT.0.) A=EXP(-TINC/CT)
      WMN = XMN*(1.-A)
      WSD = STD*SQRT(1.-A*A)
      CALL RN(W,AX,WSD,WMN)
      XN = A*XN+W
C                              COMPUTE AMBIENT TEMPERATURE
C
      DTP = TBLU1(TD,PD(4),PD(4+ND),1,-ND)
      YTP = TBLU1(WY,PY(4),PY(4+NY),1,-NY)
      TAO = (DTP + XN)*YTP/ XM
      TIMO=TIME
C
  150 RETURN
      END
```

7.47 VARIABLE RATIO TRANSMISSION



This component models a transmission which couples a fixed speed rotor input (or output) to a variable speed rotor output (or input) component. Power losses are modeled as a table lookup depending on gear ratio and input power.

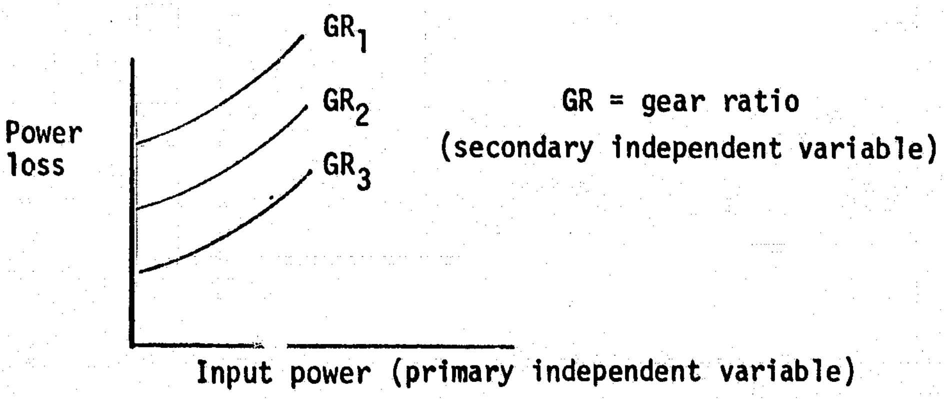


FIGURE 7.47 TRANSMISSION MODEL - LOOKUP TABLE

TR

Tables	Description	Units
PLØ	Power loss versus input power and gear ratio (TABLE DIMENSION = 66)	kw

Inputs

Parameter/Port		Description	Units
RS	1	Input rotor speed	rpm
RS	2	Output rotor speed	rpm
P	1	Input power	kw
EF	1	Input product efficiency	-
MP	1	Maximum input power	kw
CC		Capital cost/year	$
CM		Maintenance cost/year	$

Outputs

Variable/Port		Description	Units
P	2	Output power	kw
TØ		Output torque	ft-lb
PL		Power loss	kw
EF	2	Output product efficiency	-
MP	2	Maximum power output	kw

Calculation Sequence

If P1 ≤ 0 or RS1 ≤ 0 set P2 = TØ = PL = 0 and go to 4)

1) Determine gear ratio and power terms

   GR = RS2/RS1

   PL = PLØ(P1,GR)

   P2 = P1 - PL

2) Determine output torque

   TØ = P2*737.6/(RS2*(2π/60))

3) Efficiency and maximum power

   EF2 = EF1*(P2/P1)

   If P2 ≤ 0, set EF2 = EF1 and write Diagnostic

   MP2 = MP1 - PLØ(MP1,GR)

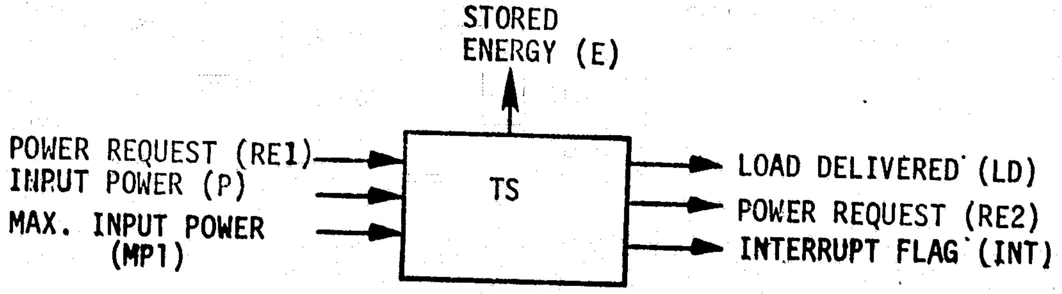
   MP2 ≤ 0 ⇨ DIAGNOSTIC

4) Compute Costs

```
CTR
      SUBROUTINE TR(PLO,P2,TO,PL,EF2,MP2,RS1,RS2,P1,EF1,MP1,CC,CM)
C
C                   PURPOSE   TRANSMISSION MODEL
C
C                   METHOD    OUTPUT POWER AND TORQUE COMPUTED FROM
C                             INPUT AND OUTPUT ROTOR SPEEDS. POWER
C                             LOSS MODELED BY TABLE LOOKUP DEPENDING
C                             ON GEAR RATIO AND INPUT POWER
C
C                   WRITTEN BY Y.K.CHAN              VERSION 1,JUNE 17,1977
C
C  CALL SEQUENCE
C       TABLES
C               PLO  -POWER LOSS VERSUS INPUT POWER AND GEAR RATIO ,KW
C       OUTPUTS
C               P2   -OUTPUT POWER,KW
C               TO   -OUTPUT TORQUE,FT-LB
C               PL   -POWER LOSS,KW
C               EF2  -OUTPUT PRODUCT EFFICIENCY
C               MP2  -MAXIMUM POWER OUTPUT,KW
C       INPUTS
C               RS1  -INPUT ROTOR SPEED,RPM
C               RS2  -OUTPUT ROTOR SPEED,RPM
C               P1   -INPUT POWER,KW
C               EF1  -INPUT PRODUCT EFFICIENCY
C               MP1  -MAXIMUM INPUT POWER,KW
C               CC   -CAPITOL COST/YEAR,$
C               CM   -MAINTENANCE COST/YEAR,$
C
      COMMON/CIMPL/IMPL,ICNT/CTIME/TIME/CSIMUL/DUM(7),TMAX
     X      /COST/CCI,CMI
      REAL MP2,MP1
      DIMENSION PLO(1)
C
      IF(IMPL.GT.0)GO TO 10
      TMAX1=TMAX*.99999
      RS2=RS1
   10 CONTINUE
      NNGR=PLO(3)
      NNP1=PLO(2)
      M4=NNGR+4
      MN4=NNP1+M4
C
C                 COMPUTE GEAR RATIO AND POWER TERMS
C
      P2=0.
      TO=0.
      PL=0.
      EF2=EF1
      MP2=MP1
  100 IF((RS1.LE.0.).OR.(P1.LE.0.))GO TO 400
      P2=P1
  200 IF(RS2.LE.0)GO TO 400
  300 GR=RS2/RS1
      PL=TBLU2(P1,GR,PLO(M4),PLO(4),PLO(MN4),1,1,-NNP1,-NNGR,NNP1,NNGR)
      P2=P1-PL
C
```

```
C                 OUTPUT TORQUE
C
      TO=P2*737.6*30./(RS2*3.14159)
C
C                 EFFICIENCY AND MAXIMUM POWER
C
      EF2=EF1*P2/P1
      IF(EF2.GT.0.)GO TO 409
      EF2=EF1
      IF(IMPL.EQ.2)WRITE(6,408)PL,P1
  408 FORMAT(1H0,25H TRANSMISSION POWER LOSS ,F12.3,
     X  21H EXCEEDS INPUT POWER  ,F12.3)
      IF(IMPL.EQ.2)ICNT=ICNT+1
  409 CONTINUE
      AMP2=TBLU2(MP1,GR,PLO(M4),PLO(4),PLO(MN4),1,1,-NNP1,-NNGR,
     XNNP1,NNGR)

       MP2=MP1-AMP2
      IF(MP2.GT.0.)GO TO 400
      IF(IMPL.EQ.2)WRITE(6,508)AMP2,MP1
  508 FORMAT(1H0,25H TRANSMISSION POWER LOSS ,F12.3,
     X 29H EXCEEDS MAXIMUM INPUT POWER  ,F12.3)
      IF(IMPL.EQ.2)ICNT=ICNT+1
C
  400 IF(IMPL.LE.1)RETURN
      IF(TIME.LT.TMAX1)RETURN
      CCI=CCI+CC
      CMI=CMI+CM
C
      RETURN
      END
```

7.48 THERMAL STORAGE CHAMBER



The thermal storage chamber is modeled by a "lumped" parameter approach. The entire storage media mass is characterized by a single temperature (no temperature gradient). The storage media is either a sensible heat or a phase change media. Energy is input via electrical resistance heaters and withdrawn by a heat exchanger. Energy is deposited in the media at a rate equal to the available electrical power up to a maximum charging power. The discharge heat exchanger fluid mass flow rate is adjusted to provide the desired heat load demand. The maximum mass flow rate condition determines the maximum thermal load. The maximum energy limit represents the point where the maximum media temperature is reached.

The model initially calculates the required storage media mass to provide the rated thermal energy storage (design point). Cost calculations are also made on the design point conditions. Initial checks on charge and discharge power and initial stored energy level are made. The storage temperature is determined based on the energy level.

Note: An example case discussing parameter specification for TS is provided on page 54, Reference [1].

Basic Equation

$$\dot{E} = P- LD-NU*E$$

Tables	Description	Units
HT	Media temperature versus enthalpy in KWH/LB[1]	°F

Inputs

Parameter/Port		Description	Units
P		Input power	kw
RE	1	Demand thermal load	kw
NU		Stored energy loss coefficient (D = 0.02)	$(h)^{-1}$
TS		Rated storage time[2]	h
VØ		Rated input voltage[2]	V
TM1		Maximum allowable storage temperature (D = 212)	°F
TØ1		Minimum allowable storage temperature (D = 60)	°F
DH		Design point enthalpy	kwh/lb
PD		Rated storage thermal power[2]	kw
PM		Maximum charge rate (D = 2*PD)	kw
MFM		Maximum working fluid mass flow rate (D = 9000)	lb/h
TDE		Temperature deadband for priority resequence (D = 4)	°F
EF	1	Input product efficiency	-
MP	1	Maximum input charging rate (D = $1.X10^{8}$)	kw
CP2		Working fluid heat capacity (D = $2.93X10^{-4}$)	kwh/lb-°F
TØ2		Working fluid return temperature (D = 40)	°F
TM2		Maximum allowable working fluid temperature (D = 212)	°F
R		Effective heat exchanger thermal resistance (D = $3.08X10^{-4}$)	°F/kw
CM		Storage device yearly maintenance cost (D = 0.6)	$/kw
CSA		Storage device capacity cost (D = 50)	$/kw
CSB		Storage device energy cost (D = 15.2)	$/kwh
LE		Unit life expectancy	years

D – Default values specified
1 – See Figure 7.48
2 – Design point conditions

Outputs

Variable/Port		Description	Units
E		Stored energy (state)	kwh
I		Input current	amps
MP	2	Maximum discharge rate allowable	kw
INT		Priority interrupt flag	-
T		Storage temperature	°F
M		Required storage media mass	lb
CCØ		Storage device capital cost/year	$
RE	2	Maximum charging rate request	kw
MF		Working fluid mass flow rate	lb/h
LD		Power Delivered	kw

Statistics

TSU	Maximum storage temperature	°F
TSL	Minimum storage temperature	°F
ME	Maximum stored energy	kwh
MFU	Maximum working fluid mass flow rate	lb/h

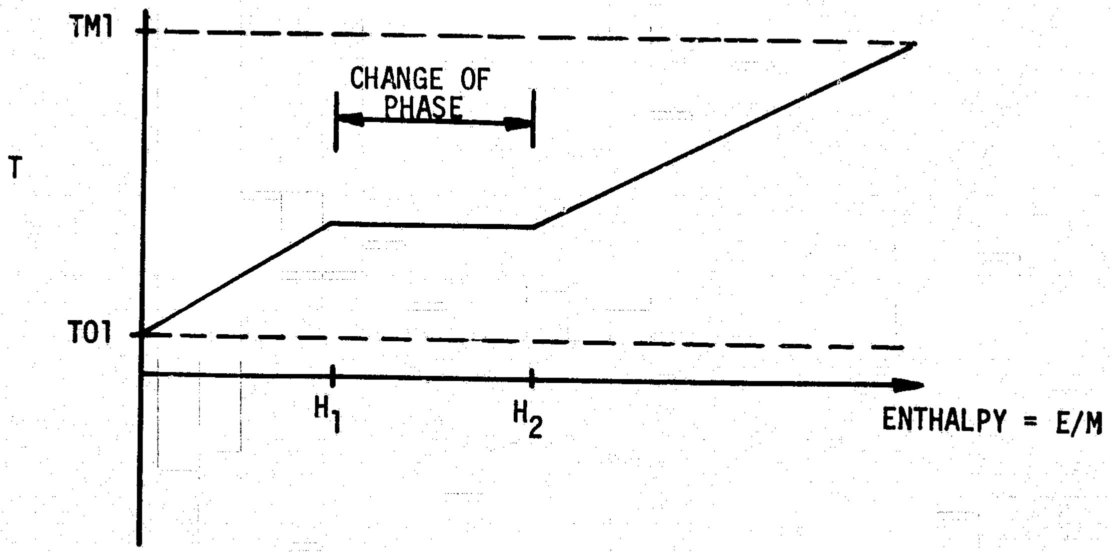


FIGURE 7.48: TEMPERATURE - ENTHALPY DIAGRAM

The calculation sequence and default values assume a thermal storage device sized to provide 10kw for 24 hours. A paraffin wax phase change storage medium is assumed. Water is assumed as the thermal transport fluid. Costs are assumed to be given by data for the phase change storage device given in Reference 1. The thermal resistance value, R, is assumed equal to that determined for the device of Reference 1. The value for the maximum charging rate, PM, reflects the acceptance of twice the design charge rate. The actual numbers which should be used will depend on specific design and performance requirements obtained from a desired application.

Calculation Sequence

1) Media mass, capital cost, maintenance cost (first pass)

$$M = \frac{PD*TS}{DH}$$

$$CC = (CSA+CSB*TS)*PD/LE$$

$$CM = CM*PD$$

2) Storage Temperature and Working Fluid Temperature

$$T = HT(E/M)$$

$$TF = \min\{TM2,\max\left[T02,T-RE1*R\right]\}$$

$$E2 = M*HT^{-1}(TM1)$$

[1] "Advanced Thermal Energy Storage," BEC/EPRI RP 788-1, July 1976.

Calculation Sequence Cont.

3) Discharge Rate and Thermal Load

$$E1 = M*HT^{-1}(T01)$$

$$MP2 = MFM*CP2*(TF-T02)$$

$$LD=MIN(RE1,MP2,(E-E1)/TINC)$$

$$MF = LD/(CP2*(TF-T02))$$

4) Diagnostic Checks

$$MF \leq MFM$$

$$P \leq PM$$

$$T01 \leq T \leq TM1$$

5) Current calculations

$$I = \frac{P*1000}{V0}$$

6) Energy dynamics

$$\dot{E} = P-LD-NU*E$$

7) Maximum Charging Rates

$$RE2 = min(PM,MP1,(E2-E)/TINC)/EF1$$

where TINC = integration step size

Calculation Sequence Cont.

8) Priority resequencing

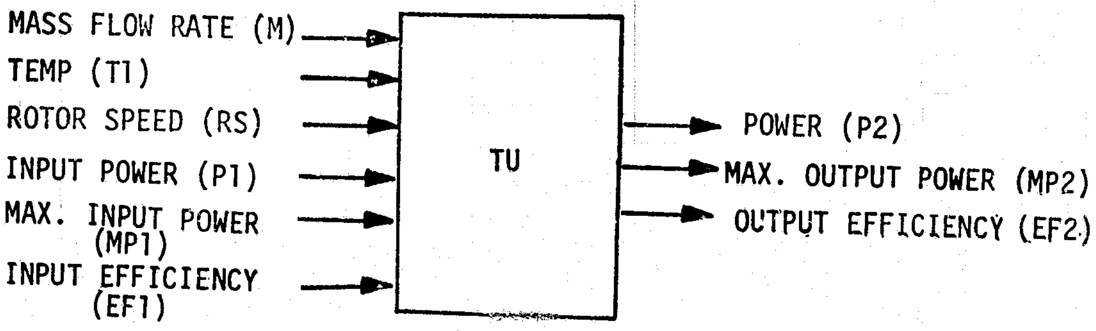
```
        if $T \leq TØ1$, $INT = 1$
        if $T \geq TØ1+TDE$ and $INT=1$,  $INT=0$
        if $T \geq TM1$, $INT=-1$
        if $T < TM1-TDE$ and $INT=-1$,  $INT=0$
```

9) Compute Statistics and Costs

```
CTS
      SUBROUTINE TS(HT,E,DE,IE,I,MP2,INT,T,M,CCO,RE,MF,LD
     1             ,TSU,TSL,ME,MFU,P,RE1,NU,TSO,VO,TM1,TO1,DH,PD,PM,
     2      MFM,TDE,EF1,MP1,CP2,TO2,TM2,R,CM,CSA,CSB
     3             ,LE)
C
C     PURPOSE    COMPUTE ENERGY CONTAINED IN A THERMAL STORAGE MEDIA
C
C     METHOD     A PHASE CHANGE OR SENSIBLE HEAT MEDIA IS MODELED AS
C
C                A SINGLE TEMPERATURE MASS WITH NO GRADIENTS.
C
C     WRITTEN BY F. O. MAHONY                 VERSION 2, JULY, 1977
C
C     CALL SEQUENCE
C
C           TABLE
C                      HT  - MEDIA TEMPERATURE VERSUS ENTHALPY IN KWH/LB, DEG F
C           OUTPUTS
C                E    - STORED ENERGY (STATE VARIABLE), KWH
C                DE   - POWER INTO STORAGE, KW
C                IE   - STATUS INDICATOR
C                I    - INPUT ELECTRIC CURRENT, KW
C                MP2  - MAXIMUM DISCHARGE RATE ALLOWABLE, KW
C                INT  - PRIORITY FLAG INTERRUPT
C                T    - STORAGE TEMPERATURE, DEG F
C                M    - REQUIRED STORAGE MEDIA MASS, LB
C                CCO  - STORAGE DEVICE CAPITAL COST/YEAR, $
C                RE   - MAXIMUM CHARGING RATE REQUEST, KW
C                MF   - WORKING FLUID MASS FLOW RATE, LB/HR
C                LD   - THERMAL LOAD DELIVERED, KW
C                TSU  - MAXIMUM STORAGE TEMPERATURE, DEG F
C                TSL  - MINIMUM STORAGE THEMPERATURE, DEG F
C                ME   - MAXIMUM STORED ENERGY, KWH
C                MFU  - MAXIMUM WORKING FLUID MASS FLOW RATE, LB/HR
C           INPUTS
C                P    - INPUT POWER, KW
C                RE1  - THERMAL DISCHARGE REQUEST, KW
C                NU   - STORAGE ENERGY LOSS COEFFICIENT, 1/HR
C                TSO  - RATED STORAGE TIME, HR
C                VO   - RATED INPUT VOLTAGE, VOLTS
C                TM1  - MAXIMUM ALLOWABLE STORAGE TEMPERATURE, DEG F
C                TO1  - MINIMUM ALLOWABLE STORAGE TEMPERATURE, DEG F
C                PD   - RATED STORAGE THERMAL POWER, KW
C                DH   - DESIGN POINT ENTHALPY, KWH/LB
C                PM   - MAXIMUM CHARGE RATE, KW
C                MFM  - MAXIMUM WORKING FLUID MASS FLOW RATE, LB/HR
C                TDE  - TEMPERATURE DEADBAND FOR PRIORITY RESEQUENCE, DEC F
C                EF1  - INPUT PRODUCT EFFICIENCY
C                MP1  - MAXIMUM INPUT CHARGING RATE, KW
C                CP2  - WORKING FLUID HEAT CAPACITY, KWH/LB-F
C                TO2  - WORKING FLUID RETURN TEMPERATURE, DEG F
C                TM2  - MAXIMUM ALLOWABLE WORKING FLUID TEMPERATURE, DEG F
C                R    - EFFECTIVE HEAT EXCHANGER THERMAL RESISTANCE, F/KW
C                CM   - STORAGE DEVICE YEARLY MAINTENANCE COST, $/KW
C                CSA  - STORAGE DEVICE CAPACITY COST, $/KW
C                CSB  - STORAGE DEVICE ENERGY COST, $/KWH
C                LE   - UNIT/LIFE FILE EXPECTANCY, YEARS
```

```
C
      COMMON/CIMPL/IMPL,ICN/CTIME/TIME /CSIMUL/DUM(7),TMAX /COST/CCI,CMI
      REAL NU,I,MP2,INT,MF,LD,ME,MFU,MFM,MP1,LE,M
      DIMENSION HT(1)
C
      IF(IMPL.GT.0)GO TO 100
      TMAX1=TMAX*.99999
      TINC=DUM(7)
C
C
      IF(NU.EQ. .99999)NU=0.02
      IF(TM1.EQ. .99999)TM1=212.0
      IF(TO1.EQ. .99999)TO1=60.0
      IF(PM. EQ. .99999) PM=2.0*PD
      IF(MFM.EQ. .99999)MFM=9000.0
      IF(TDE.EQ. .99999)TDE=4.0
      IF(CP2.EQ. .99999)CP2=2.93E-4
      IF(TO2.EQ. .99999)TO2=40.0
      IF(TM2.EQ. .99999)TM2=212.0
      IF(R  .EQ. .99999)R  =3.08E-4
      IF(CM .EQ. .99999)CM =0.6
      IF(CSA.EQ. .99999)CSA=50.0
      IF(CSB.EQ. .99999)CSB=15.2
      IF(MP1.EQ. .99999) MP1= 1.0E6
C
      INT=0.0
      RE1=0.0
C
      TSU=0.0
      ME =0.0
      MFU=0.0
      TSL=1.0E6
      CM= CM*PD
C
      M =PD*TSO/DH
      CCO=(CSA+CSB*TSO)*PD/LE
C
C                          COMPUTE STORAGE TEMPERATURE
C
  100 NH=HT(2)
      T=TBLU1(E/M,HT(4),HT(4+NH),1,NH)
      E1=M*TBLU1(TM1,HT(4+NH),HT(4),1,NH)
C
C                          WORKING FLUID TEMPERATURE
C
      TF =AMIN1(TM2,AMAX1(TO2,T-RE1*R))
C
C                          MAXIMUM DISCHARGE RATE AND THERMAL LOAD
C
      MP2=MFM*CP2*(TF-TO2)
      IF(INT.EQ.1.)MP2=0.
      LD =AMIN1(RE1,MP2)
C
C                          WORKING FLUID MASS FLOW RATE
C
      IF(LD.GT.0.0) MF =LD/CP2/(TF-TO2)
C
      IF(IMPL.LE.1)GO TO 200
      IF(IMPL.GT.2) GO TO 200
      PM1= PM/.9999
C
      IF(MF .GT.MFM)WRITE(6,1010)MF,MFM
      IF(P  .GT.PM1 )WRITE(6,1020)P ,PM
      IF(MF.GT.MFM .OR. P.GT.PM1)ICN=ICN+1
      IF(T  .LT.TO1.OR.
     1   T  .GT.TM1)WRITE(6,1030)T,TO1,TM1
      IF(T.LT.TO1 .OR. T.GT.TM1) ICN=ICN+1
C
C                       CURRENT CALCULATION
C
  200 I =P*1000.0/VO
C
C                       ENERGY STATE
C
C
      IF(IE.NE.O)DE=P-LD-NU*E
C
C                       MAXIMUM CHARGING RATE REQUEST
C
C
      A=AMAX1(E1-E,0.)/TINC
      RE =AMIN1(PM,MP1,A)/EF1
C
C                       PRIORITY RESEQUENCING
C
      IF(T.LE.TO1)INT=1.0
C
      IF(T.GE.(TO1+TDE).AND.
     1   INT.EQ.1.)INT=0.0
C
      IF(T.GE.TM1)INT=-1.0
C
      IF(T.LT.(TM1-TDE).AND.
     1   INT.EQ.-1.)INT=0.0
C
      IF(IMPL.LE.1)RETURN
C
      TSU=AMAX1(TSU,T)
      TSL=AMIN1(TSL,T )
      ME =AMAX1(ME ,E )
      MFU=AMAX1(MFU,MF)
      IF(TIME.LT.TMAX1)RETURN
C
C                       COST
C
      CMI=CMI+CM
      CCI=CCI+CCO
      CM = CM/PD
C
      RETURN
 1010 FORMAT(1H0,26HTS WORKING FLUID FLOW RATE,F12.3
     1          ,32H   GREATER THAN MAXIMUM ALLOWED,F12.3)
 1020 FORMAT(1H0,14HTS INPUT POWER,F12.3
     1          ,44H   GREATER THAN MAXIMUM ALLOWED CHARGE RATE,F12.3)
 1030 FORMAT(1H0, 23HTS STORAGE TEMPERATURE   ,F12.3
     1   ,20H      OUTSIDE MINIMUM ,F12.3
     2            ,15H    AND MAXIMUM,F12.3)
C
      END
```

7.49 TURBINE (PNEUMATIC)



The turbine model is based on a high pressure ratio, constant angular velocity design. The turbine is assumed to be designed to a set of of operating conditions defined in terms of user specified parameters. The polytropic efficiency is only weakly related to angular velocity. Initial calculations are made with the design polytropic efficiency, and refinements are then computed after off-design parameters are calculated.

Basic Equation

The equation for output power P2 is

$$P2 = M*CP*(T1-TA)$$

Inputs

Parameter/Port		Description	Units
M		Inlet mass flow rate	lb/h
CP		Air heat capacity (D = $7.2X10^{-5}$)	kwh/lb/°F
T	1	Input air temperature	°F
TA		Ambient air temperature	°F
MD		Design mass flow rate (D = 4800)	lb/h
TID		Design inlet air temperature (D = 600)	°F
PID		Design inlet pressure (D = 117.6)	psi
P2D		Design exit pressure (ambient) (D = 14.7)	psi
T2D		Design exit temperature (ambient) (D = 70)	°F
PS		Storage vessel pressure	psi
RS		Angular velocity	rpm
EF	1	Input product efficiency	-
MP	1	Maximum input power	kw
P	1	Input power	kw
CK		Capacity cost coefficient[1] (D = 0.015)	
FØ		Turbine mass flow exponent for capital cost (D = 0.75)	-
G		Turbine temperature exponent for capital cost (D = 0.5)	-
NPD		Design Polytropic Efficiency (D = 0.88)	

Outputs

Variable/Port			
P	2	Output power	kw
CCØ		Turbine cost/year	$
PR		Back pressure	psi
TØ		Torque	ft-lb

D - Default values supplied

[1] CK = Capital cost (known unit)/[(design point mass flow rate)^{FO}* (design point temperature + 460)^{G} * LN (inlet/outlet pressure ratio)*LE], where LE = life expectancy in years.

Outputs Cont.

Variable/Port		Description	Units
EF	2	Output product efficiency	-
MP	2	Maximum discharge power	kw

Statistics

MØP	Maximum power observed	kw

The calculation sequence and the default values are based on the assumption of a high pressure ratio, constant angular velocity turbine, rated at 150 kw and a pressure ratio of 8. The equations used relate first order effects among the various physical quantities and were derived from first principles originally in support of the research work of Reference 1. Cost scaling was also developed in that reference based on cost estimates from turbomachinery manufacturers.

[1] "Closed Cycle High Temperature Control Receiver Concept for Solar Electric Power," BEC/EPRI RP377-1, June 1976.

Calculation Sequence

1) Costs

CC = CK*(MD)**FØ*(TID+460)**G*LN(PID/P2D)

2) Back Pressure PR determined by

PR = (M/MD)*PID* $\sqrt{(T1+460)/(TID+460)}$

If PR > PS write DIAGNOSTIC

3) Efficiency

RAT = (PID/P2D)**(2/7)

EFF = (RAT-1.)/(RAT**(1/NPD)-1)

4) Power Out

P2 = M*CP*(T1-TA)*EFF

5) Torque

If RS = 0, set TO = 0 and go to 6)

TØ = P2*(737.6)/(RS*2π/60)

6) Efficiency and maximum power

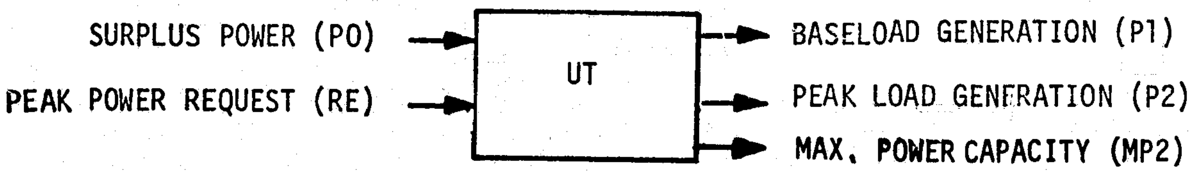
EF2 = EF1*EFF

MP2 = min(MP1*EFF,MD*CP*(T1-TA))

7) Compute Statistics and Costs

```
CTU
      SUBROUTINE TU(P2,CC,PR,TO,EF2,MP2,MOP,M,CP,T1,TA,MD,TID,PID,P2D,
     1            T2D,PS,RS,EF1,MP1,P1,CK,F,G,NPD)
C
C     PURPOSE     TURBINE PERFORMANCE MODEL
C
C     METHOD      COMPUTE TURBINE POWER OUTPUT FROM INPUT DESIGN
C
C                 CONDITIONS AS A FUNCTION OF INLET TEMPERATURE
C
C                 AND MASS FLOW RATE
C
C     WRITTEN BY F.O. MAHONY                    VERSION 1, MARCH 22 1977
C
C     CALL SEQUENCE
C          OUTPUTS
C                 P2   - OUTPUT POWER, KW
C                 CC   - TURBINE COST PER YEAR, $
C                 PR   - BACK PRESSURE, PSI
C                 TO   - TORQUE, FT-LB
C                 EF2  - OUTPUT PRODUCT EFFICIENCY
C                 MP2  - MAXIMUM DISCHARGE POWER, KW
C                 MOP  - MAXIMUM POWER OBSERVED,KW
C          INPUTS
C                 M    - INLET MASS FLOW RATE, LB/HR
C                 CP   - AIR HEAT CAPACITY, KWH/LB/DEG F
C                 T1   - INPUT AIR TEMPERATURE, DEG F
C                 TA   - AMBIENT AIR TEMPERATURE, DEG F
C                 MD   - DESIGN MASS FLOW RATE, LB/HR
C                 TID  - DESIGN INLET AIR TEMPERATURE, DEG F
C                 PID  - DESIGN INLET PRESSURE, PSI
C                 P2D  - DESIGN EXIT PRESSURE (AMBIENT), PSI
C                 T2D  - DESIGN EXIT TEMPERATURE (AMBIENT), PSI
C                 PS   - STORAGE VESSEL PRESSURE, PSI
C                 RS   - ANGULAR VELOCITY, RPM
C                 EF1  - INPUT PRODUCT EFFICIENCY
C                 MP1  - MAXIMUM INPUT POWER, KW
C                 P1   - INPUT POWER, KW
C                 CK   - CAPACITY COST COEFFICIENT
C                 F    - TURBINE MASS FLOW EXPONENT FOR CAPITAL COST
C                 G    - TURBINE TEMPERATURE EXPONENT FOR CAPITAL COST
C                 NPD  - DESIGN POLYTROPIC EFFICIENCY
C
      COMMON /CIMPL/IMPL,ICNT/CTIME/ TIME /CSIMUL/DUM(7),TMAX /COST/CCI
      REAL MP2,MOP, M,MD, MP1,NPD
      DATA PI /3.14159/
C
      IF(IMPL.GT.0) GO TO 100
      IF(CP .EQ. .99999) CP = 72.0E-6
      IF(TA .EQ. .99999) TA = 70.0
      IF(MD .EQ. .99999) MD = 4800.
      IF(TID.EQ. .99999) TID=600.0
      IF(PID.EQ. .99999) PID=117.6
      IF(P2D.EQ. .99999) P2D=14.7
      IF(T2D.EQ. .99999) T2D=70.0
      IF(CK .EQ. .99999) CK =0.015
      IF(F  .EQ. .99999) F  =0.75
      IF(G  .EQ. .99999) G  =0.5
      IF(NPD .EQ. .99999)NPD= .88
      MOP = 0.
      RS= AMAX1(0.0,AMIN1(RS, 4000.))
      TMAX1=.99999*TMAX
      CC = CK*MD**F*(TID+460.)**G*ALOG(PID/P2D)
C                        DETERMINE BACK PRESSURE
C
  100 RAT= (PID/P2D)**.2857
      EFF= (RAT-1.0)/(RAT**(1./NPD) - 1.0)
      PR = M/MD*PID*SQRT((T1 + 460.0)/(TID+460.))
C
      IF(PR.GT.PS) GO TO 1000
C
C                        POWER OUTPUT
C
C
  200 P2= M*CP*(T1-TA)*EFF
      TO = 0.
      IF(RS.EQ.0. .OR. P1.EQ. 0.) GO TO 300
C                        TORQUE
      TO = P2*737.6/(RS*2.0*P1/60.0)
C
C                        EFFICIENCY AND MAXIMUM POWER
C
  300 EF2 = EF1*EFF
      MP2 = AMIN1(MP1*EFF ,MD*CP*(T1-TA))
      IF(IMPL.LE. 1) RETURN
      MOP = AMAX1(MOP,P2)
      IF(TIME.LT.TMAX1) RETURN
      CCI = CCI + CC
C
      RETURN
C
 1000 IF(IMPL.EQ.2)WRITE(6,1010) PR,PS
 1010 FORMAT (1H0,21HTURBINE BACK PRESSURE,F12.3,
     1              39H   GREATER THAN STORAGE VESSEL PRESSURE ,F12.3)
      IF(IMPL.EQ.2)ICNT=ICNT+1
C
      GO TO 200
C
      END
```

7.50 UTILITY



The utility model has two power outputs corresponding to baseload and peak generation, with corresponding generation cost inputs. A surplus power input is also provided with cost credit depending on whether baseload or peak power is reduced. Total energy cost, total output power and total peak load requests are monitored.

Minimum input parameters to specify the utility are:

CB = cost of baseload generation ($/kwh),
CP = cost of peak load generation ($/kwh).

Note: Even if no baseload generation is assumed, CB may be needed to compute surplus power cost credits.

UT

Inputs

Parameter/Port		Description	Units
BS		Baseload generation (default = 0.)	kw
CB		Cost of baseload generation/kwh	$
MP	1	Maximum power capacity (default = 1×10^8)	kw
P	0	Surplus power returned to utility	kw
RE		Peak generation request	kw
CP		Cost of peak load generation/kwh	$
CC		Capital cost/year	$
CM		Maintenance cost/year	$

Outputs

Variable/Port			
P	1	Baseload generation (= BS)	kw
MP	2	Maximum power capacity (= MP1)	kw
P	2	Peak load generation	kw
CØ		Cost of energy used (state)	$

Statistics

SR	Sum of requested peak generation	kwh
SPØ	Sum of output energy	kwh
SP	Sum of surplus energy	kwh
VSP	Value of surplus energy	$

Calculation Sequence

1) Power outputs

If $BS > MP1$, write diagnostic

$P1 = BS,\ MP2 = MP1$

$P2 = MIN\ (MP1-BS, RE)$

2) Energy cost dynamics

$$C\varnothing = BS*CB + (P2-P\varnothing)*CX$$

$$CX = \left(\begin{array}{ll} CP & \text{if } P2-P\varnothing > 0 \\ 0 & \text{if } P2-P\varnothing < 0 \end{array}\right.$$

3) Statistics

$$SR = SR + RE * TINC$$

$$DEL = \left(\begin{array}{ll} 0 & \text{if } P2 > P\varnothing \\ (P\varnothing-P2)*TINC & \text{if } P\varnothing > P2 \end{array}\right.$$

$$SP\varnothing = SP\varnothing + (P1+P2-P\varnothing)*TINC + DEL$$

$$SP = SP + DEL$$

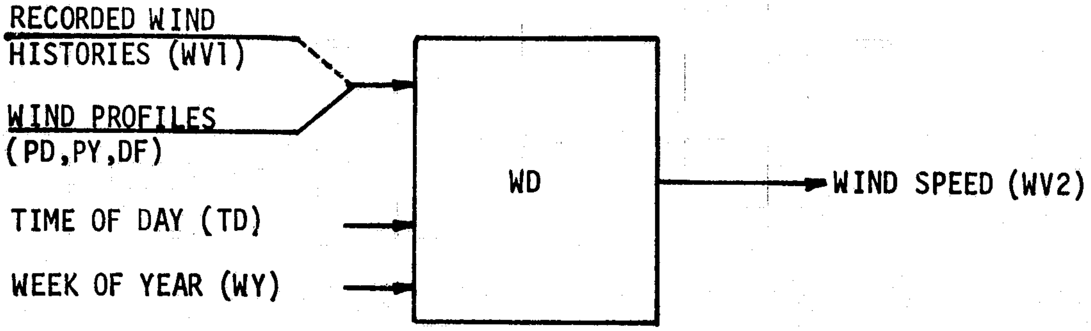
$$VSP = VSP + DEL * CB$$

Where TINC = integration step size/2

4) Compute Costs

```
CUT
      SUBROUTINE UT(P1,MP2,P2,CO,COD,ICO,SR,SPO,SP,VSP
     1              ,BS,CB,MP1,PO,RE,CP,CC,CM)
C
C               PURPOSE    MODEL OF UTILITY CAPABLE OF PRODUCING
C                          BASELOAD AND PEAKLOAD POWER, AND OF
C                          ABSORBING SURPLUS POWER
C
C               METHOD     COMPUTE PEAKLOAD GENERATION AND ENERGY COST
C
C               WRITTEN BY Y.K.CHAN      VERSION 1,JUNE 8,1977
C
C    CALL SEQUENCE
C          OUTPUT
C               P1   -BASELOAD GENERATION,KW
C               MP2  -MAXIMUM POWER CAPACITY,KW
C               P2   -PEAKLOAD GENERATION,KW
C               CO   -COST OF ENERGY USED (STATE), $
C               COD  -ENERGY COST RATE, $/HR
C               ICO  -INTEGRATOR CONTROL FOR CO
C          STATISTICS
C               SR   -SUM OF REQUESTED PEAK GENERATION,KW
C               SPO  -SUM OF OUTPUT ENERGY, KWH
C               SP   -SUM OF SURPLUS ENERGY, KWH
C               VSP  -VALUE OF SURPLUS ENERGY, $
C          INPUTS
C               BS   -BASELOAD GENERATION (DEFAULT=0.),KW
C               CB   -COST OF BASELOAD GENERATION/KWH, $
C               MP1  -MAXIMUM POWER CAPACITY,KW
C               PO   -SURPLUS POWER RETURNED TO UTILITY,KW
C               RE   -PEAK GENERATION REQUEST, KW
C               CP   -COST OF PEAKLOAD GENERATION/KWH, $
C               CC   -CAPITAL COST/YEAR, $
C               CM   -MAINTENANCE COST/YEAR, $
C
      COMMON /CIMPL/IMPL,ICNT/CTIME/TIME/CSIMUL/DUM(7),TMAX
     X  /COST/CCI,CMI,COP,VDE,TDE,TLD,UTV,UTD,SPD
      REAL MP2,MP1
C
      IF(IMPL.GT.0)GO TO 100
      IF(BS.EQ..99999)BS=0.
      IF(MP1.EQ..99999)MP1=1.E6
      TMAX1=TMAX*.99999
      SR=0.
      SP=0.
      SPO=0.
      VSP=0.
      RE=0.
      PO=0.
C
C               COMPUTE POWER OUTPUTS
C
      TINC1=DUM(7)*.5
      IF(BS.LE.MP1)GO TO 100
      IF(IMPL.EQ.2) WRITE(6,208)BS,MP1
  208 FORMAT(1H0,10H BASELOAD ,F12.3,32H EXCEEDS MAXIMUM POWER CAPACITY,
     1 F12.3)
      IF(IMPL.EQ.2)ICNT=ICNT+1
      BS=MP1
C
  100 P1=BS
      MP2=MP1
      P2=AMIN1(MP1-BS,RE)
C
C                COMPUTE ENERGY COST
C
      CX=0.
      IF(P2.GT.PO)CX=CP
      IF(ICO.NE.0)COD=BS*CB+(P2-PO)*CX
      IF(IMPL.LE.1)RETURN
C
C                STATISTICS
C
      SR=SR+RE*TINC1
      SPO=SPO+ (P1+P2-PO)*TINC1
      IF(P2.GT.PO) GO TO 200
      TERM=(PO-P2)*TINC1
      SPO= SPO+TERM
      SP= SP+ TERM
      VSP= VSP+ CB*TERM

C
  200 IF(TIME.LT.TMAX1)RETURN
      CCI=CCI+CC
      CMI=CMI+CM
      VDE=VDE-CO+VSP
      TDE=TDE-SPO+SP
      UTV=UTV+CO
      UTD=UTD+SPO
      SPD= SPD+SP
      RETURN
      END
```

7.51 WIND



This model computes wind speed either from user supplied time histories (data tape) or by generating a set of random numbers with user supplied daily and yearly average profiles and user specified random variation. If user supplied profiles are available then the wind speeds are generated from the following equation:

Basic Equation

$$WV = [PD(TD) + N(T)]*PY(WY)/M$$

where PD is the user supplied daily mean profile

TD is the time of the day

PY is the user supplied yearly profile

WY is the week of the year

N is white noise with user specified probability distribution

$$M = \frac{1}{J} \sum_{i=1}^{J} PY(i)$$

WD

Tables	Description	Units
PD	Daily profile versus TD (default = 0)	miles/hr
PY	Yearly profile versus WY	arbitrary
DF	Density function for white noise terms (tabular with speed WV)	arbitrary

Inputs

Parameter/Port			
WV	1	Wind speed data file input	miles/hr
TD		Time of day	hr
WY		Week of the year	-

Outputs

Variable/Port			
WV	2	Wind speed	miles/hr
M		Mean of yearly profile	
TIM		Last time a random sample was generated	hr

Statistics

Statistics		
MV	Maximum speed	miles/hr
AV	Average speed (expected daily wind)	miles/hr

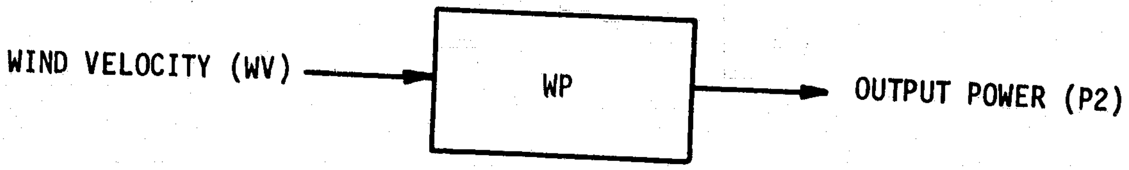
C-5

Calculation Sequence

```
1)  Compute distribution function and mean M (first pass only)

        $F(WV) = \frac{(\Sigma\ DF(V_i)\ :\ V_i \leq WV)}{\Sigma\ DF(V_i)}$

2)  Check for data file input
    If WV1 = .99999  go to 3)
        WV2 = WV1
        Go to 5)

3)  Generate white noise input N
    If TIME = TIM  go to 5)
        U = random noise sample, uniformly distributed [0,1]
        Interpolate to find $N = F^{-1}(U)$
        TIM = TIME

4)  Compute wind speed
        WV2 = [PD(TD) + N ]* PY(WY)/M

5)  Compute Statistics
```

```
CWD
      SUBROUTINE WD (PD,PY,WF,WVO,AMV,AV,XM,TIMO ,WVI,TD,WY)
C
C   PURPOSE    GENERATE WIND SPEED FROM DAILY, YEARLY, AND RANDOM PROFILE DA
C
C   METHOD     RANDOM NOISE WITH SPECIFIED DIST. IS ADDED TO MEAN DAILY PROF
C              AND MULTIPLIED BY A YEARLY PROFILE. INITIALLY THE DENSITY TABL
C              WF IS CONVERTED TO A DIST. FUNCTION.
C
C   WRITTEN BY A.W. WARREN                             VERSION 1, MARCH 4 197
C
C   CALL SEQUENCE
C         TABLES
C              PD   - MEAN DAILY WIND PROFILE, MPH
C              PY   - MEAN YEARLY WIND PROFILE
C              WF   - WIND FREQUENCY FUNCTION (NON-GUST, RANDOM COMPONENT), HI
C
C         OUTPUTS
C              WVO - WIND VELOCITY OUTPUT, MPH
C              AMV - MAX. OBSERVED WIND SPEED, MPH
C              AV   - MEAN DAILY WIND SPEED, MPH
C              XM   - MEAN YEARLY WIND, -
C              TIMO- LAST TIME A RANDOM SAMPLE WAS USED, HR
C
C         INPUTS
C              WVI - WIND VELOCITY INPUT FROM DATA FILE, MPH
C              TD   - TIME OF DAY, HR
C              WY   - WEEK OF YEAR (1-52)
C
C
      DIMENSION PD(1),PY(1),WF(1)
      COMMON/CIMPL/IMPL /CTIME/TIME
      DATA IX/1/
C                              INITIALIZATION
C                              COMPUTE MEAN DAILY WIND SPEED AND DIST. FCN
C
      NP=WF(2)
      ND=PD(2)
      NY=PY(2)
      IF(IMPL.GT.0) GO TO 10
      SUM=0.0
      AMN=0.0
      TIMO=-1.
      IF(WF(4+2*NP).EQ. 1.) GO TO 40
      DO 20 I=1,NP
      WF(I+2) = WF(I+3)
      L = 3+NP+I
      A=WF(L)
      WF(L)=SUM
      SUM=SUM+ A
   20 AMN = AMN + A*WF(2+I)
      AMN = AMN/SUM
      WF(3+NP)= WF(NP+2)*2. - WF(NP+1)
      WF(L+1)= 1.
C
      DO 30 I=1,NP
      L=3+NP+I
   30 WF(L) = WF(L)/SUM
   40 CONTINUE
C                                  DEFAULT TABLE FOR PD
C
      IF(PD(2).EQ. 1.99999) PD(4)=0.
      IF(PD(2).EQ. 1.99999) PD(5)=0.
      IF(PD(2).EQ. 1.99999) PD(2)=1.
C
      AV = 0.
      DO 25 I=1,ND
      L = 3+ND+I
   25 AV = AV+PD(L)
      AV = AV/ND + AMN
C
      XM=0.
      DO 15 I=1,NY
      L=3+NY+I
   15 XM=XM+PY(L)
      XM=XM/NY
      AMV =0.
C                                  CHECK FOR DATA FILE INPUT
C
   10 IF( WVI.EQ. .99999) GO TO 100
      WVO = WVI
      GO TO 150
C                                  GENERATE WHITE NOISE WITH DIST. WF
C
  100 IF(  TIME.EQ.TIMO) GO TO 150
      CALL UNIF(U,IX)
      NP1=NP+1
      WN = TBLU1(U,WF(4+NP),WF(3),1,-NP1)
C
C                                  GENERATE WIND SPEED USING DAILY AND YEARLY PROFILES
      DWV = TBLU1(TD,PD(4),PD(4+ND),1,-ND)
      YWV = TBLU1(WY,PY(4),PY(4+NY),1,-NY)
      WVO = (DWV + WN)* YWV / XM
      TIMO=TIME
C
C                                  MAX. OBSERVED WIND SPEED
  150 IF(IMPL.LE.1) RETURN
      AMV = AMAX1(AMV,WVO)
      RETURN
      END
```

7.52 TURBINE/GENERATOR



This component uses a power curve relationship with wind velocity to model the wind turbine and generator. It may be used in place of the more detailed wind turbine-transmission-generator components where a simplified analysis is desirable, or where a nonstandard wind generator model is desired. The model may be used for either A.C. or D.C. power generation.

Basic Equations

$$P2 = I = 0 \qquad \left(\begin{array}{l} WV < WV0 \\ WV > WV1 \end{array} \right.$$

$$P2 = V*I/1000 \qquad WV0 \leq WV \leq WV1$$

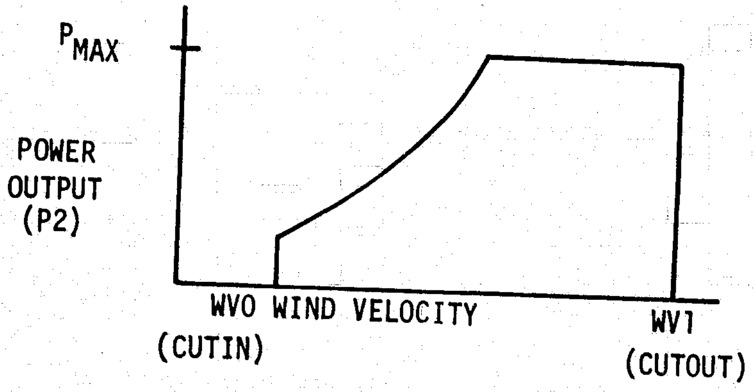


FIGURE 7.52: OUTPUT POWER VERSUS WIND VELOCITY

WP

Tables	Description	Units
PW	Wind generation power versus wind velocity [1]	kw

Inputs

Parameter/Port	Description	Units
V	Bus voltage (Rated)	volts
WV0	Power cutin velocity	mph
WV1	Power cutout velocity	mph
WV	Wind velocity	mph
CC	Capital cost/year	$
CM	Maintenance cost/year	$
EC	Control Energy Rate	$/hr

Outputs

Variable/Port		Description	Units
I		Bus current	amps
P	2	Real power output	kw

Statistics	Description	Units
MI	Maximum current	amps
MPØ	Maximum power	kw
SP	Total output energy	kwh
CØ	Total operating costs	$

[1] Output power including mechanical and electrical efficiencies

Calculation Sequence

1) Initialize statistics

2) Compute P2 and I

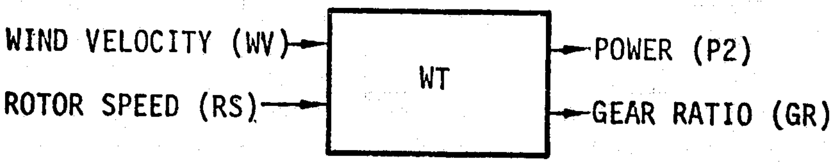
$$P2 = \left(\begin{array}{ll} PW(WV) & WV0 \leq WV \leq WV1 \\ 0 & \text{otherwise} \end{array} \right)$$

$$I = P2*1000/V$$

3) Compute Statistics and Costs

```
CWP
      SUBROUTINE WP ( PW,BI,PO,AMI,AMP,SP,CO,VO,WVO,WV1,WV,CCI,CMI,EC)
C
C   PURPOSE    MODEL THE WIND TURBINE AND GENERATOR USING A POWER CURVE
C
C   WRITTEN BY A.W. WARREN                               VERSION 1, MARCH 3 197
C
C   CALL SEQUENCE
C         TABLES
C              PW   - WIND GENERATION POWER IN KW VERSUS WIND VELOCITY IN MPH
C
C         OUTPUTS
C              BI   - OUTPUT BUS CURRENT, AMPS
C              PO   - POWER OUTPUT, KW
C              AMI  - MAX. OBSERVED CURRENT, AMPS
C              AMP  - MAX. OBSERVED POWER, KW
C              SP   - TOTAL OUTPUT ENERGY,KWH
C              CO   - OPERATING COST, $
C
C         INPUTS
C              VO   - RATED BUS VOLTAGE, VOLTS
C              WVO  - POWER CUTIN VELOCITY, MPH
C              WV1  - POWER CUTOUT VELOCITY, MPH
C              WV   - WIND VELOCITY, MPH
C              CCI  - CAPITOL COST / YEAR, $
C              CMI  - MAINTENANCE COST / YEAR, $
C              EC   - CONTROL ENERGY RATE, $/HR
C
      DIMENSION PW(1)
      COMMON / CIMPL / IMPL
      COMMON/COST/ CC,CM,COP /CTIME/ TIME /CSIMUL/ DUM(6),TINC,TMAX
C
C                          POWER OUTPUT CALCULATIONS
C
      PO = 0.
      IF(WV.LT.WVO .OR. WV.GT.WV1) GO TO 10
      N = PW(2)
      PO = TBLU1(WV,PW(4),PW(4+N),1,-N)
   10 BI = PO*1000/VO
C                          STATISTICS
C
      IF(IMPL.GT.0) GO TO 20
      CO= 0.
      AMI = 0.
      AMP = 0.
      SP = 0.
      TMAX1=TMAX*.99999
   20 IF(IMPL.LE.1) RETURN
      AMI = AMAX1(AMI,BI)
      AMP = AMAX1(AMP,PO)
      SP = SP + PO*.5*TINC
      CO= CO + EC*.5*TINC
C                          COST SUMMATION
      IF( TIME.LT.TMAX1) RETURN
      CC = CC + CCI
      CM = CM + CMI
      COP= COP + CO
      RETURN
      END
```

7.53 WIND TURBINE



This component models the wind turbine in terms of physical properties such as blade radius, power coefficient, and design tip speed ratio.[1] The step-up gear ratio is computed based on design rotor speed.

Basic Equations

Output power is given by

$$P2 = CP*1/2*AD*A*(WV*C)^3*k$$

where:

CP = effective power coefficient (tabular with WV)
$A = \pi*(BR)^2$
$C = 1.4667$ (mph to ft/sec. conversion)
$k = 1.3558 \times 10^{-3}$ (ft-lb to kw-sec. conversion)

Minimum input parameters to specify the wind turbine are:

VO = mean wind speed,
BR = blade radius,
CPM = maximum power coefficient at design speed VO.

[1] NASA CR 134937 "Design Study of Wind Turbines - 50kw to 3000 kw - For Electric Utility Applications", Kaman Aerospace Corporation, February 1976.

WT

Inputs

Parameter/Port	Description	Units
WV	Wind speed	mph
VØ	Mean wind speed (yearly)	mph
VR	Rated wind speed (default = 1.35 x VO)	mph
RS	Rotor speed	rpm
RSG	Generator shaft speed (design)(default = 1800)	rpm
BR	Blade radius	ft
EC	Cost to operate controls	$/h
AD	Air density (default = 0.0023)	slugs/ft^3
LAM[1]	Design tip speed ratio (default = 9.4)	-
CPM[2]	Maximum power coefficient at VO (default = 0.4)	-
CP	Effective power coefficient (default table versus VO/WV)	-
CC	Capital cost/year	$
CM	Maintenance cost/year	$

Outputs

Variable/Port		Description	Units
P	2	Output mechanical power	kw
TØ		Mechanical torque	ft-lb
CØ		Total operating cost	$
GR		Step-up gear ratio	-
RAP		Rated output power	kw

Statistics

	Description	Units
MT	Maximum torque	ft-lb
MPØ	Maximum power	kw
SP	Total energy delivered	kwh

[1] LAM may be computed using the design equation:

LAM = SQRT(8/(3* solidity constant * design lift coefficient))

[2] If default CP table not used then set CPM = CP(rated wind speed)

Calculation Sequence

1) First pass - Compute Gear Ratio and Rated Power

$RS = (LAM*V\emptyset*C/BR)*(60/2\pi)$

$GR = RSG/RS$

$RAP = .5*CP1*AD*A*(VR*C)^3$

where

$$CP1 = \begin{cases} CPM*F(V\emptyset/VR) & \text{if CP default used} \\ CPM & \text{otherwise} \end{cases}$$

2) Compute power coefficient CP

If WV = 0 set $P2=T\emptyset=0$ and go to 4)

If CP default used, then

$CP = CPM*F(V\emptyset/WV)$

where F is shown in Figure 7.53

3) Power and torque

$A = \pi*BR^2$

$P = .5*CP*AD*A*(WV*C)^3$ (C = 1.4667)

$TO = P/(RS*2\pi/60)$

$P2 = P*k$ ($k = 1.3558 \times 10^{-3}$)

4) Compute Statistics and Costs

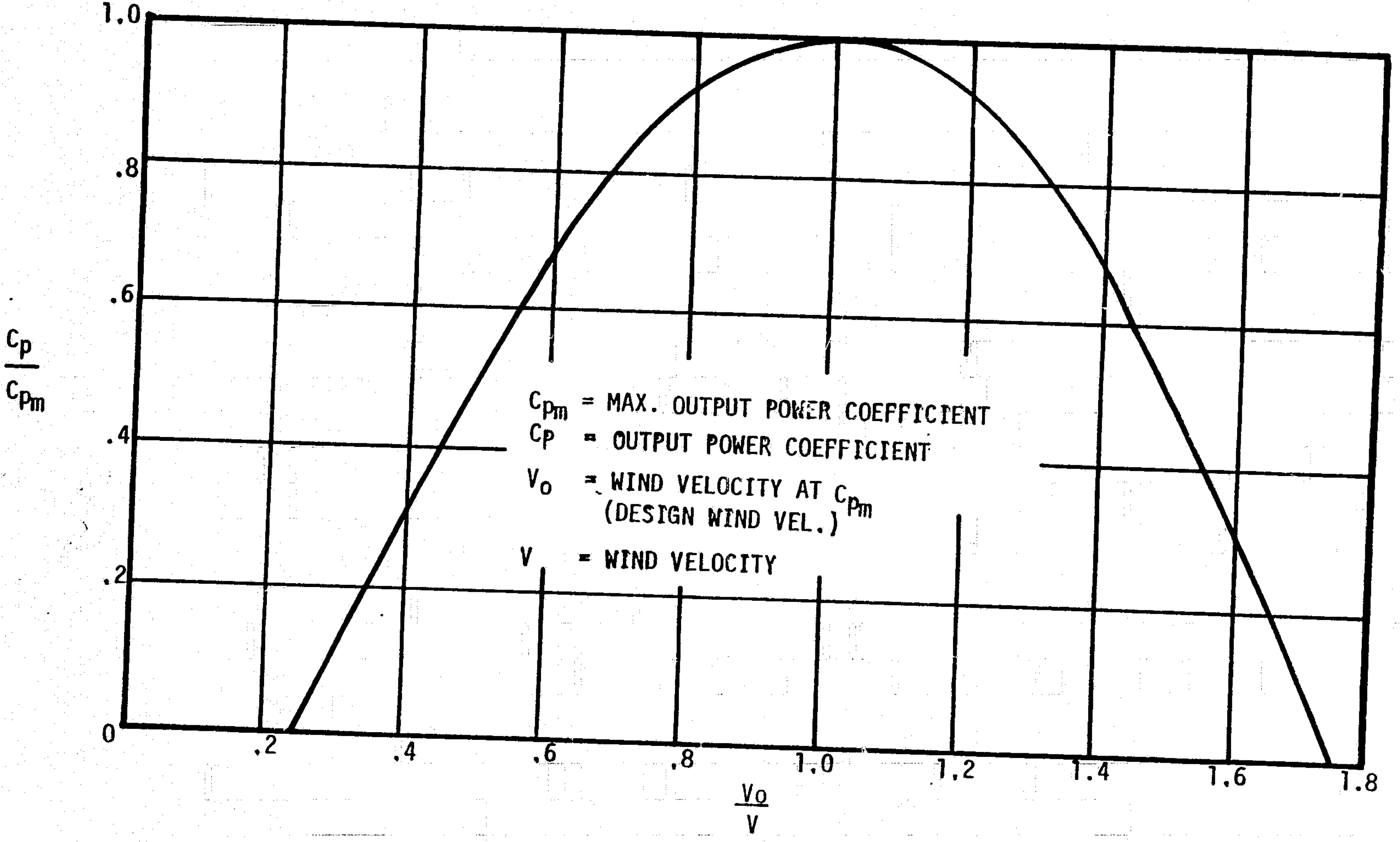


FIGURE 7.53 GENERALIZED MACHINE POWER OUTPUT PERFORMANCE

```
CWT
      SUBROUTINE WT (P2,TO,CO,GR,RAP,MT,MPO,SP,WV,VO,VR,RS, RSG,BR,EC,
     1                 AD,LAM,CPM, CP,CC,CM)
C
C   PURPOSE     MODEL WIND TURBINE POWER OUTPUT
C
C   METHOD      COMPUTE POWER COEFFICIENT AND ROTOR SPEED FROM PHYSICAL
C               DESIGN PARAMETERS.  RATED POWER COEFF. IS 3/4 OF CPM.
C
C   WRITTEN BY A. W. WARREN                      VERSION 2, APRIL 6 1977
C
C   CALL SEQUENCE
C         OUTPUTS
C               P2  - OUTPUT MECHANICAL POWER, KW
C               TO  - OUTPUT MECHANICAL TORQUE, FT-LB
C               CO  - OPERATING COST SUM, $
C               GR  - TURBINE/GENERATOR  GEAR RATIO
C               RAP - RATED OUTPUT POWER, KW
C               MT  - MAXIMUM TORQUE STATISTIC, FT-LB
C               MPO - MAXIMUM POWER STATISTIC, KW
C               SP  - TOTAL OUTPUT ENERGY DELIVERED, KWH
C
C         INPUTS
C               WV  - WIND SPEED, MPH
C               VO  - MEAN WIND SPEED (YEARLY), MPH
C               VR  - RATED WIND SPEED, MPH
C               RS  - ROTOR SPEED, RPM
C               RSG - GENERATOR SHAFT SPEED, RPM
C               BR  - BLADE RADIUS, FT
C               EC  - CONTROL ENERGY RATE, $/HR
C               AD  - AIR DENSITY, SLUGS/FT**3
C               LAM - DESIGN TIP SPEED RATIO
C               CPM - MAXIMUM POWER COEFFICIENT AT VO
C               CP  - EFFECTIVE POWER COEFFICIENT AT WV
C               CC  - CAPITAL COST PER YEAR
C               CM  - MAINTENANCE COST PER YEAR
C
      COMMON /CIMPL/IMPL /CTIME/ TIME /CSIMUL/ DUM(6),TINC,TMAX
      COMMON /COST/ CCI,CMI,COI
      REAL MT,MPO,LAM
      DIMENSION F(22)
      DATA F/.24,.4,.6,.68,.8,1.,1.2,1.31,1.4,1.6,1.74,0.,.31,.68,.8,
     1     .92,1.,.92,.8,.68,.3,0. /,C1,C2,PI/1.4667,.0013558,3.14159 /
C
C                         INITIALIZATION
C
      IF(IMPL.GT.0) GO TO 100
      TMAX1 = TMAX* .99999
      TINC2 = .5* TINC
C
      IF( VR .EQ. .99999) VR = 1.35* VO
      IF(RSG .EQ. .99999) RSG= 1800.
      IF( AD .EQ. .99999) AD = .0023
      IF(LAM .EQ. .99999) LAM= 9.4
      IF(CPM .EQ. .99999) CPM= 0.4
C
      RS = C1*LAM*VO/BR*(30./PI)
      GR = RSG /RS
      CP1=CPM
      IF(CP.EQ..99999)CP1= CPM*TBLU1(VO/VR,F(1),F(12),1,-11)
      RAP= .5*CP1*AD*PI*BR*BR*(VR*C1)**3*C2
      CO = 0.0
      SP = 0.0
      MPO= 0.0
      MT = 0.0
C
C                        POWER COEFFICIENT CALCULATION
C
  100 P2 = 0.0
      TO = 0.0
      IF( WV.EQ. 0.) GO TO 200
      CP1 = CP
      IF( CP1.EQ. .99999) CP1 = CPM*TBLU1(VO/WV,F(1),F(12),1,-11)
      IF(CP1.EQ. 0.) GO TO 200
C
C                        OUTPUT POWER AND TORQUE
C
      A = PI*BR**2
      P = .5*CP1*AD*A*(WV*C1)**3
      IF(WV.GT.VR)P= RAP/C2
      TO=P/(RS*PI/30.)
      P2 = P*C2
C
C                        STATISTICS AND COSTS
C
  200 IF(IMPL.LE.1) RETURN
C
      CO = CO + EC*TINC2
      MT = AMAX1( MT,TO)
      MPO= AMAX1(MPO,P2)
      SP = SP + P2*TINC2
C
      IF(TIME.LT.TMAX1)RETURN
      CCI = CCI + CC
      CMI = CMI + CM
      COI = COI + CO
C
      RETURN
      END
```

8.0 WIND GENERATION AND STORAGE EXAMPLES

This section gives four simple example simulations using the SIMWEST program. These examples exercise all storage components of the SIMWEST library and many of the model features. Each example contains the input data for model generation and analysis, selected printer output generated by the programs and a discussion of the results obtained. It is recommended that a user work through and understand the model connections for these examples before attempting to build more complex models.

8.1 BATTERY STORAGE MODEL

A simplified schematic of the battery storage model is shown in Figure 8.1-1. In this model, wind power supplemented by utility generation is supplied to a power divider, which delivers power to the load as a first priority, and battery storage as second priority. Similarly, if the load cannot be met from the wind or storage, then the utility is requested to supply peaking power to meet the load. This model exercises the logic components including the priority interrupt.

Figure 8.1-2 shows the model generation input data for the model. The components are generally defined in the order of power flow shown in Figure 8.1-1. Ordering the component definition in this way is recommended to avoid convergence problems in the iteration loop. Thus, it would be somewhat better for consistency to define UT after WP rather than after LO in the model. All three types of model connections are illustrated in this example. For example, WP has the general input connection WD, MAB has the specific input connection WP(P,2 = FIN), and PD has the port to port connection PA(1,1). The port connections are especially useful for connecting up the multiport logic components PA and PD. The connection PA(1,1), for example, connects an input request of PD to PA and a power and maximum power input of PA to PD. It may be observed that the utility is connected up to the surplus port of PD. Thus the baseload power sent to MAB in effect is reduced whenever the load and battery

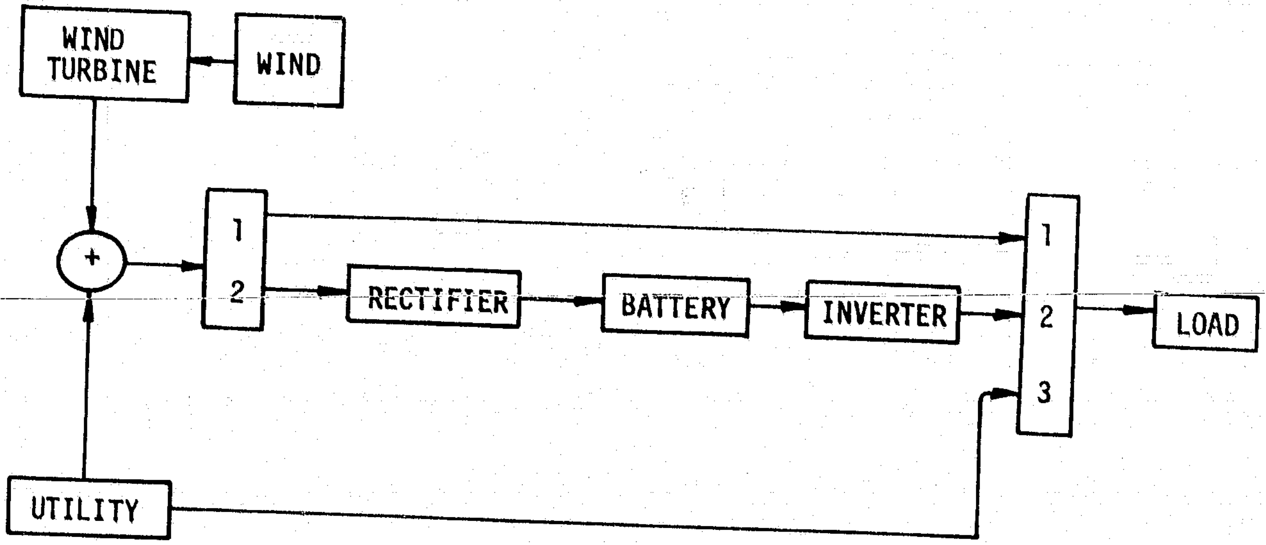


FIGURE 8.1-1: BATTERY STORAGE EXAMPLE

cannot absorb all the power generated. The last component defined is the cost monitor CM, which receives cost input data from other components through a common block rather than by model connections. Figure 8.1-3 shows the model schematic generated by the program. Most of the connection inputs are shown but occasionally a model connection will be overprinted. For example, the input RE1PA to PD is not shown in 8.1-3. In cases like this it is necessary to check the Fortran model (EQMO) in order to verify the model connections.

```
MODEL DESCRIPTION       BATTERY TEST CASE
LOCATION=74     TI
LOCATION=61     WD        INPUTS=TI
LOCATION=21     WP        INPUTS=WD
LOCATION=42     MAB       INPUTS=WP(P,2=FIN),UT(P,1=C2)
LOCATION=33     PD        INPUTS=MAB(FO=P),MAB(FO=MP),PA(1,1),PIB(2,2),
                          BA(RE=RE,2)
LOCATION=15     RE        INPUTS=PD(2,1)
LOCATION=17     BA        INPUTS=RE,PA(RE,2=RE)
LOCATION=45     PIB       INPUTS=BA
LOCATION=19     IV        INPUTS=BA
LOCATION=69     PA        INPUTS=IV(2,2),LO(1,0),PIB(4,2),UT(2,3)
LOCATION=76     LO        INPUTS=TI
LOCATION=62     UT        INPUTS=PD(SP=P,0)
LOCATION=1      CM
END OF MODEL
LIST OF STANDARD COMPONENTS
PRINT
```

FIGURE 8.1-2 BATTERY MODEL INPUT DATA

The input data for two simulations is shown in Figure 8.1-4. In the first simulation the battery is nearly full at time = 0 and the load is chosen larger on the average than the wind and utility power generated. In the second simulation the reverse is true, i.e. the load is less than that supplied by the

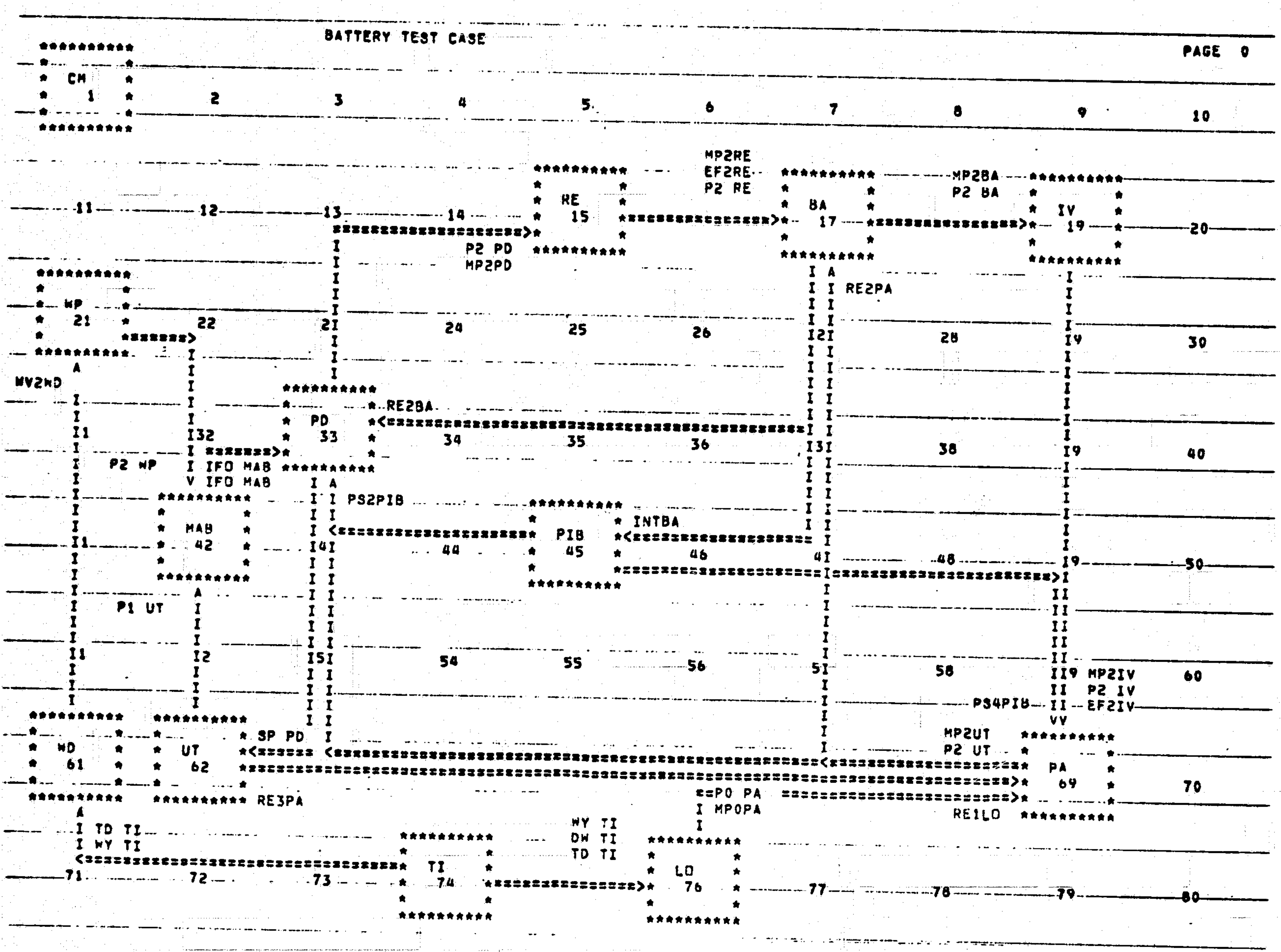


FIGURE 8.1-3 BATTERY MODEL SCHEMATIC

```
TITLE= BATTERY MODEL TEST
PARAMETER VALUES
CR CM=15.,LE CM=30.
BS UT=20.,CB UT=.016,CP UT=.03,CC UT=1000.,CM UT=1000.
C1 MAB=1.
CYCLES=4.01,TO TI=0,V  WP=400,WV0WP=8,WV1WP=60,DLINES=100.
CC WP=16000,CM WP=1200,PS1PIB=2.
EC WP=.2
NC LO=.005,CT LO=4,MN LO=0,STDLO=6,VE LO=.023
RAPBA=200.,E1 BA=2000.,EDEBA=100.
VO BA=100
DT BA=10.,CC BA=2000.,CM BA=100.
RAPRE=200.,CC RE=200.,RAPIV=200.,CC IV=0.
TABLE,PW WP=10
8,10,12,14,16,18,20,21.53,25,30
25.6,50.1,86.5,137.4,205.1,292.,400.6,500,782.8,800
TABLE,PY WD=13
0.,4.33,8.67,13.,17.33,21.67,26.,30.33,34.67,39.,43.33,47.67,52.
65,67,68.65,61,56,51,49,49,52.56,61,65
TABLE,PD WD=7
0.4,8,12,16,20,24
10,12,14.16,14,12,10
TABLE,DF WD=16
0.1,2,3,4,5,6,7,8,9,10,11,12,13,14,15
5,44,160,380,480.512,440.376,307,270,148,76,40,22,9,3
TABLE,PD LO=17
0,1.5,3,4.5,6,7.5,9,10.5,12
13.5,15,16.5,18,19.5,21,22.5,24
450,360,372,330,450,660,810,798,804
690,708,699,702,750,708,570,450
TABLE,PW LO=7
1,2,3,4,5,6,7
1,1,.9,.9,.9,.6,.5
TABLE,PY LO=6
0.10,20,30,40,52
226,194,180,174,194,226
INITIAL CONDITIONS, PE BA =1990.
PRINTER PLOTS,DISPLAY1
WV2WD,VS,TIME
P1 PD,VS,TIME
P2 PD,VS,TIME
PE BA,VS,TIME
DISPLAY2
P2 IV,VS,TIME
RE2BA,VS,TIME
RE1LO,VS,TIME
TINC=.25,TMAX=336.,PRATE=8,PRINT CONTROL=3,INT MODE=3,OUTRATE=8
SIMULATE
PARAMETER VALUES,BS UT=0.,NC LO=.005
E1 BA=1000.,EDEBA=200.
CC IV=1000.
INITIAL CONDITIONS,PE BA=250.
SIMULATE
```

FIGURE 8.1-4 INPUT DATA FOR BATTERY SIMULATION

wind system, and the battery storage is fairly low. Figures 8.1-5 to 8.1-8 show results from the first simulation. The cost monitor output is shown in Figure 8.1-5. The energy cost of the wind system is low because the wind profile delivers high energy winds during most of the simulation. The wind velocity shown in Figure 8.1-6 averages about 22 mph. Figure 8.1-7 shows the wind power output supplied directly to the load. The median power output is seen to exceed 450 kw and occasionally output reaches 800 kw = rated power. Figure 8.1-8 shows the usage of battery energy to meet the load during the week, and the increase in storage capacity during the weekends. Since the battery subsystem was limited to 180 kw maximum discharge, the utility was frequently called to meet peak loads. Thus about 10% of the load was satisfied by the utility backup.

```
                    WIND ENERGY STORAGE COST SUMMARY

                      30  YEAR LIFE CYCLE


      ●  YEARLY SYSTEM COSTS
                    CAPITAL COST                    86400. $
                    (INCLUDING FIXED CHARGES)

                    FIXED O + M COST                 2300. $

                    OPERATING + FUEL COST            1753. $

                    TOTAL                           90453. $


      ●  ENERGY DELIVERED
                    ENERGY DELIVERED            4682165. KWH
            *******************************************
            *                                          *
            *       ENERGY COST PER KWH      19.3 MILLS *
            *                                          *
            *******************************************

                    VALUE OF ENERGY DELIVERED 105440 $
                    (VALUE OF FUEL SAVED)

                    ENERGY VALUE PER KWH         22.5 MILLS

                    COST PER VALUE DELIVERED      .86


      ●  LOAD FACTOR
                    PERCENT OF LOAD SUPPLIED     90.4
                    BY TOTAL WID SYSTEM

                    PERCENT OF LOAD SUPPLIED      9.6
                    BY UTILITY

                    PERCENT OF WIND ENERGY        4.0
                    SURPLUSED

                    COST TO MEET LOAD            19.8 MILLS
                    (WIND + UTILITY)
```

FIGURE 8.1-5 COST MONITOR OUTPUT FOR BATTERY MODEL

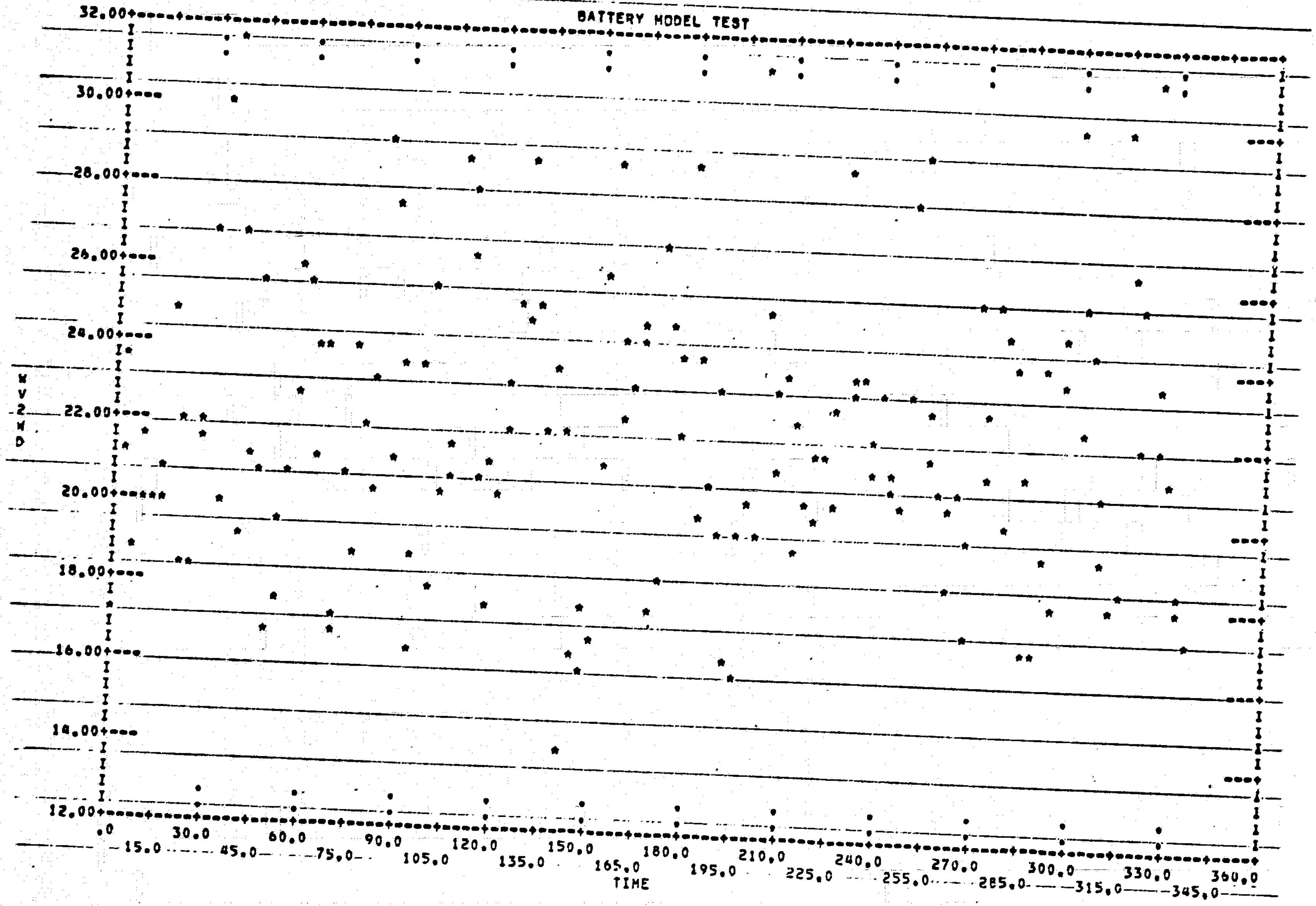


FIGURE 8.1-6 WIND PROFILE FOR BATTERY SIMULATION

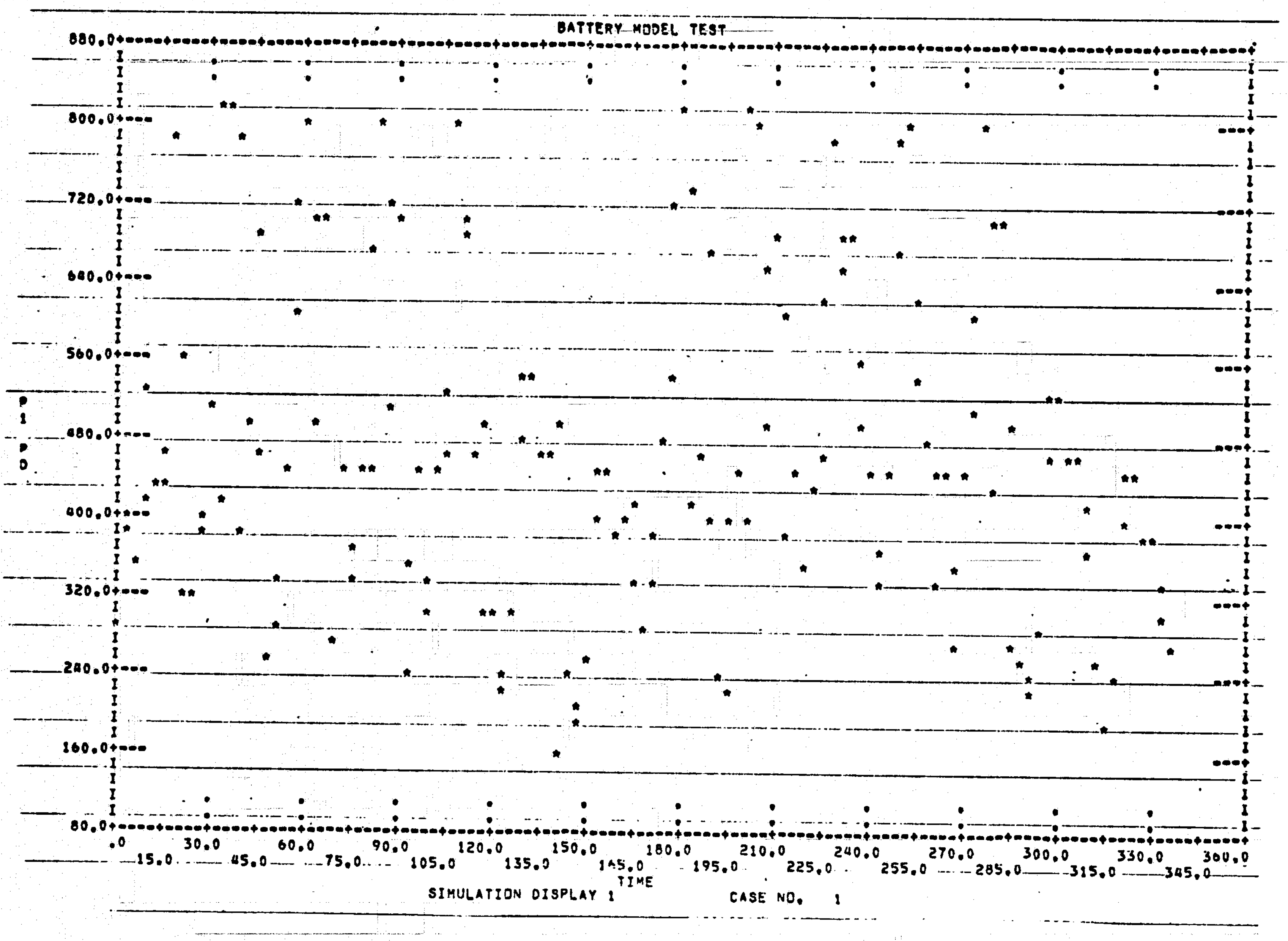


FIGURE 8.1-7 WIND POWER SUPPLIED TO LOAD

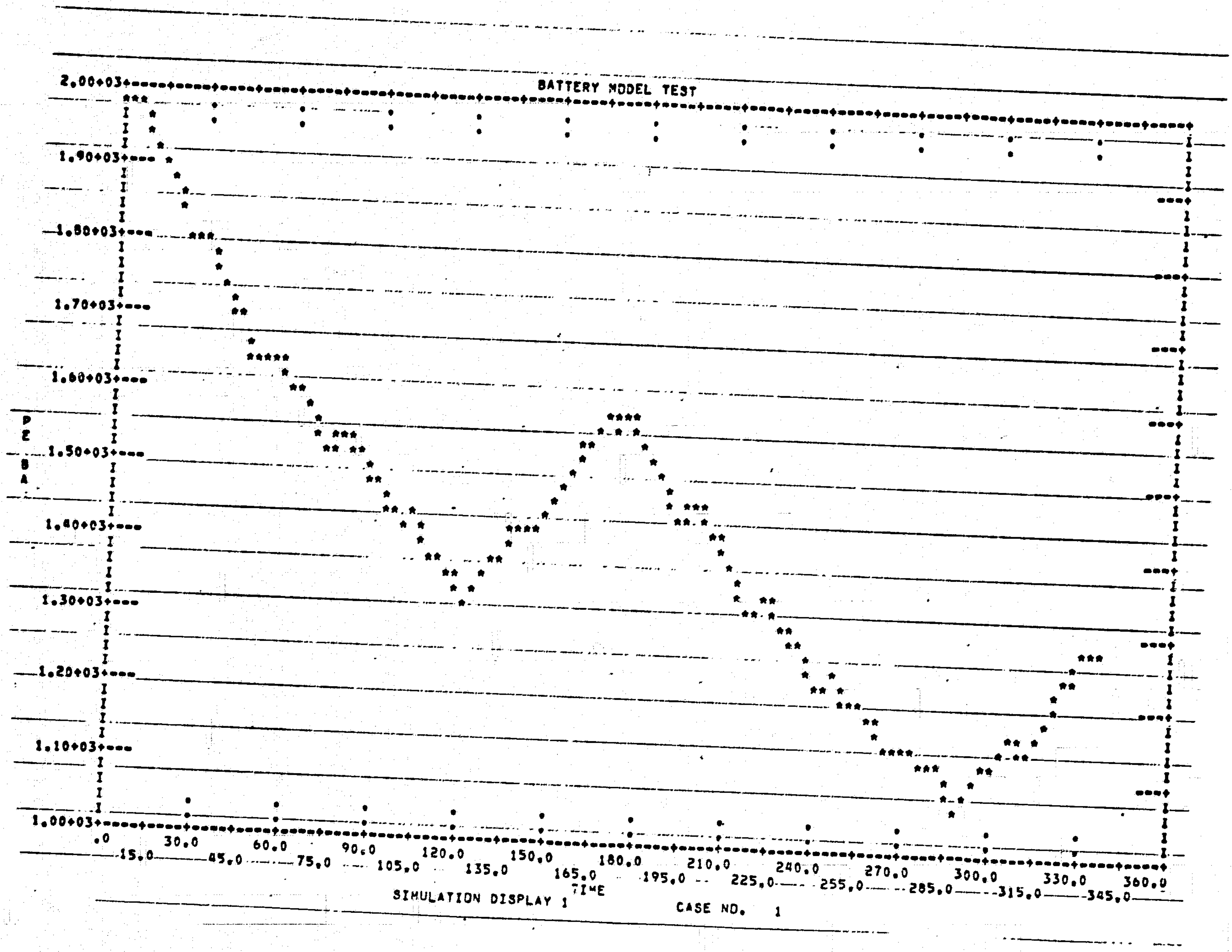


FIGURE 8.1-8 BATTERY POTENTIAL ENERGY STORAGE

8.2 FLYWHEEL STORAGE MODEL

Figure 8.2-1 shows a simplified schematic of the flywheel storage model. This model is very similar to that of 8.1 except that flywheel storage replaces battery storage, and a power line loss is included in the model. The input data for this model is shown in Figure 8.2-2. Observe that the components are defined in the order of information flow shown in 8.2-1. The admittance component AD is used to model transmission line power losses. The model schematic is shown in Figure 8.2-3.

```
MODEL DESCRIPTION         FLYWHEEL TEST CASE
LOCATION=74     TI
LOCATION=61     WD        INPUTS=TI
LOCATION=21     WP        INPUTS=WD
LOCATION=42     MAB       INPUTS=WP(P,2=FIN),UT(P,1=C2)
LOCATION=33     PD        INPUTS=MAB(FO=PP),PA(1,1),PIB(2,2),FL(RE=RE,2)
LOCATION=13     MO        INPUTS=PD(2,1)
LOCATION=4      TRI       INPUTS=MO(2,1),FL(RS=RS,2)
LOCATION=6      FL        INPUTS=TRI,PA(2,1)
LOCATION=8      TRO       INPUTS=FL,GE(RS=RS,2)
LOCATION=19     GE        INPUTS=TRO
LOCATION=69     PA        INPUTS=GE(2,2),LO(RE,1=RE,0),PIB(4,2),UT(2,3)
LOCATION=78     AD        INPUTS=PA
LOCATION=76     LO        INPUTS=TI,AD
LOCATION=62     UT        INPUTS=PD(SP=P,0)
LOCATION=1      CM
END OF MODEL
LIST STANDARD COMPONENTS
PRINT
```

FIGURE 8.2-2 FLYWHEEL MODEL INPUT DATA

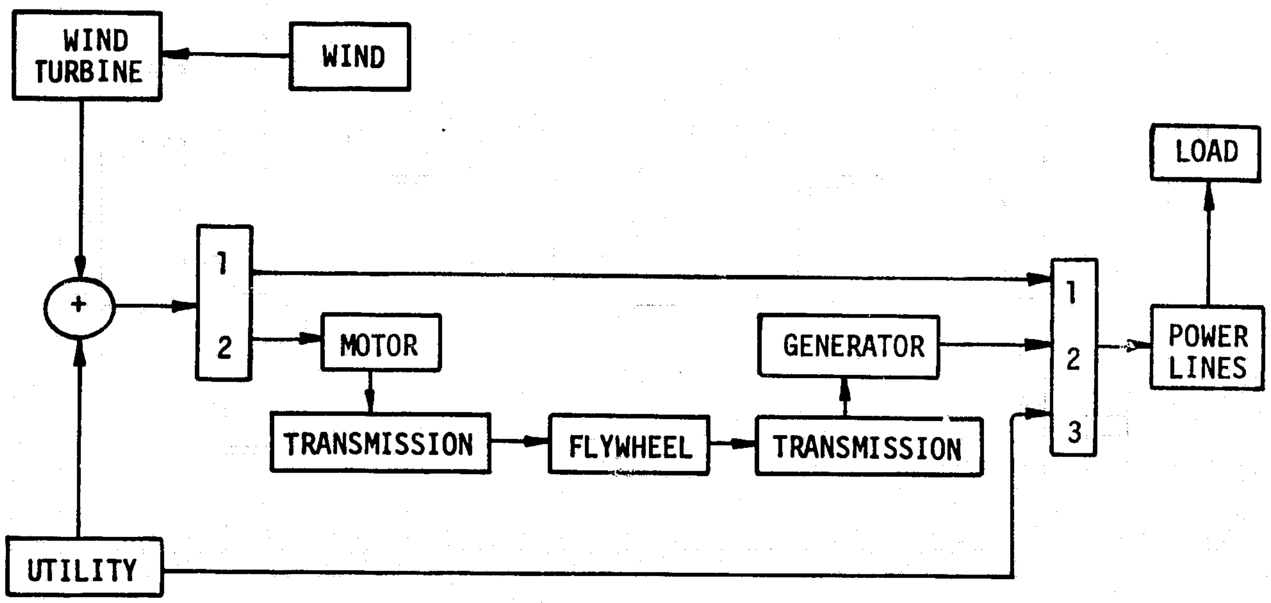


FIGURE 8.2-1: FLYWHEEL STORAGE EXAMPLE

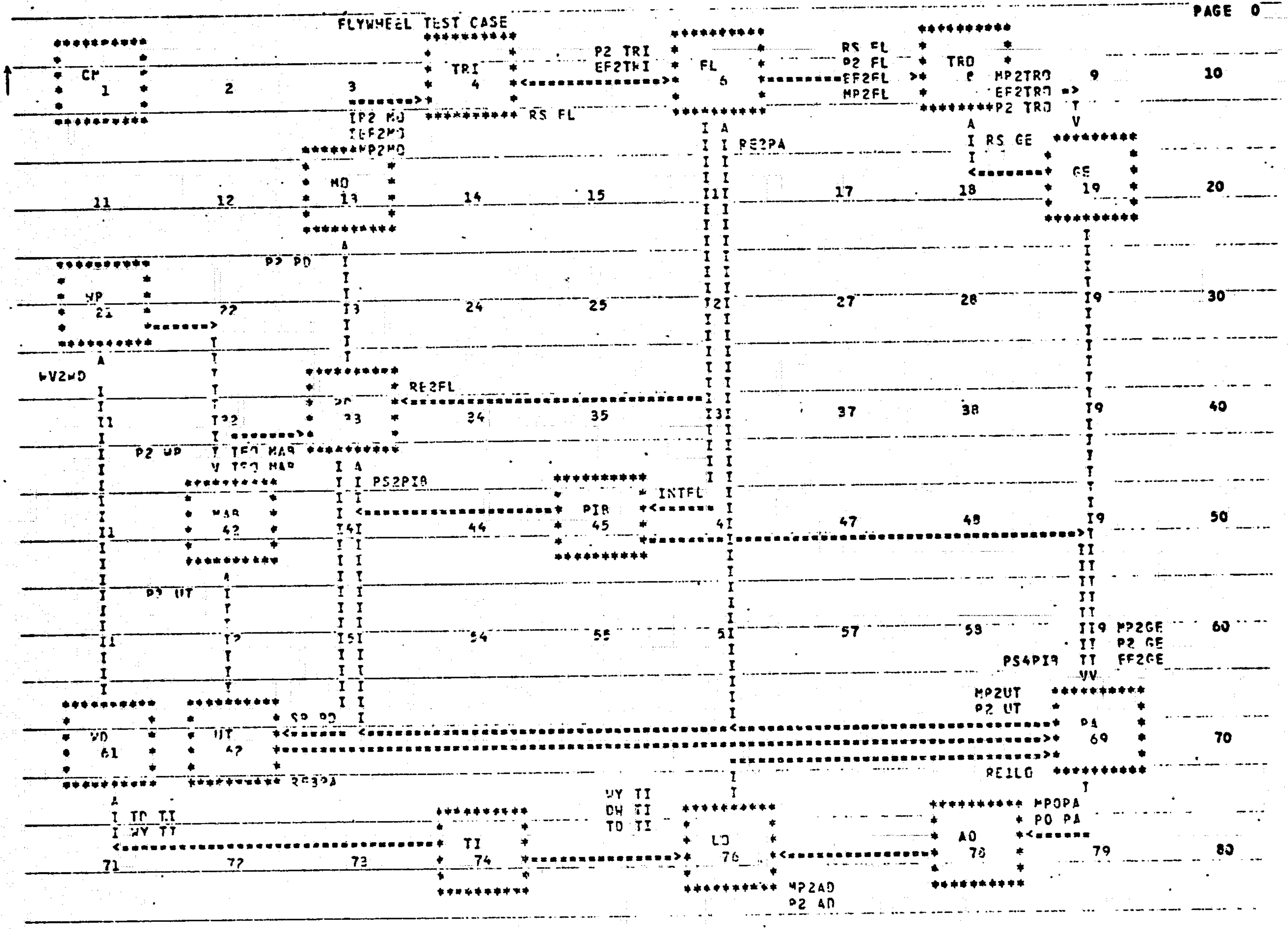


FIGURE 8.2-3 FLYWHEEL MODEL SCHEMATIC

The simulation input data shown in Figure 8.2-4 uses the same wind and load data as Example 8.1. However, the storage component is rated at 400 kw with one hour storage, simulating a system used for temporary storage and discharge during peak power generation and load demand periods. It may be noted that the transmission power loss table is input for both TRI and TRO. Figures 8.2-5 to 8.2-7 show results from the simulation. Charging power to the flywheel in excess of that needed for the load is shown in Figure 8.2-5. Even with average load demand exceeding wind generation, the flywheel is charged at rated power fairly often. The kinetic energy stored by the flywheel over a two week period is shown in Figure 8.2-6. During the week, energy is frequently withdrawn and storage is generally not much above the deadband (80 kwh), whereas during the weekend the reverse is true. Output from the cost monitor is shown in Figure 8.2-7. The capital costs may be low since nominal values were used for component costs. The utility supplied nearly 20% of the load in this case, since flywheel storage capacity is quite low.

```
TITLE= FLYWHEEL MODEL TEST
PARAMETER VALUES
VO AD=100,G1 AD=8.,G2 AD=8.,GM AD=-8.,BM AD=200
SR GE=.008,C2 FL=3.E-8
PR FL=.02,HM FL=3372,RF FL=3.5,SR FL=.4,WT FL=24000,KF FL=1,5E-5
ZE FL=.1,RAPFL=400,EO FL=40,E1 FL=400,EDEFL=20,CM FL=800,CC FL=300
RS MD=1750,RAPMD=1000,MP1MD=1.E8,CC MD=500,CM MD=0.
RS1TRI=1750,CC TRI=500,CM TRI=0,CC TRO=500,CM TRO=0.
RAPGE=1000,CC GE=1000,CM GE=100.
CR CM=15.,LE CM=30.
BS UT=20.,CB UT=.016,CP UT=.03,CC UT=1000.,CM UT=1000.
C1 MAB=1.
CYCLES=4.01,TO TI=0,V  WP=400,WVOWP=8,WV1WP=60,DLINES=100.
CC WP=16000,CM WP=1200,PS1PIB=2.
EC WP=.2
NC LO=.005,CT LO=4,MN LO=0,STDLO=6,VE LO=.023
TABLE,PLOTRI=5,4
0.5,1,1.5,1.72
0.400,900,1100,1300
0.16,18,18.5,20
0.10,11,11.5,12
0,10,10,10.5,11
0.6,6.5,7,10
TABLE,PLOTRO=5,4
.5,1,1.5,1.72
0.400,900,1100,1300
0,16,18,18.5,20
0.10,11,11.5,12
0.10,10,10.5,11
0,6,6.5,7,10
TABLE,CLOFL=3,3
-1000,  0,  1000
2000,4000,7000
2.8,     7.4,    15
 .9,      2.5,    5
2.6,     7.2,    15
TABLE,CL1FL=3
2000,4000,7000
 .8,      2.4,    4
TABLE,PW WP=10
8,10,12,14,16,18,20,21.53,25,30
25.5,50.1,86.5,137.4,205.1,292.,400,6,500,782,8,800
TAHLF,PY WD=13
0.,4.33,8.67,13.,17.33,21.67,26.,30.33,34.67,39.,43.33,47.67,52.
65,67,68,65,61,56,51,49,49,52,56,61,65
TABLE,PD WD=7
0,4,8,12,16,20,24
10,12,14,16,14,12,10
TABLE,DF WD=16
0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15
5,44,160,380,480,512,440,376,307,270,148,76,40,22,9,3
TABLE,PD LO=17
0.1.5,3,4.5,6,7.5,9,10.5,12
13.5,15,16.5,18,19.5,21,22.5,24
450,360,372,330,450,660,810,798,804
690,708,699,702,750,708,570,450
TABLE,PW LO=7
1,2,3,4,5,6,7
1,1,.9,.9,.9,.6,.5
TABLE,PY LO=6
0.10,20,30,40,52
226,194,180,174,194,226
INITIAL CONDITIONS, KE FL=300.
PRINTER PLOTS,DISPLAY1
WV2WD,VS,TIME
P1 PD,VS,TIME
P2 PD,VS,TIME
KE FL,VS,TIME
DISPLAY2
P2 GE,VS,TIME
RE2FL,VS,TIME
RE1LO,VS,TIME
TINC=.25,TMAX=336.,PRATE=8,PRINT CONTROL=3,INT MODE=3,OUTRATE=8
SIMULATE
```

ORIGINAL PAGE IS
OOR QUALITY

FIGURE 8.2-4 FLYWHEEL SIMULATION DATA

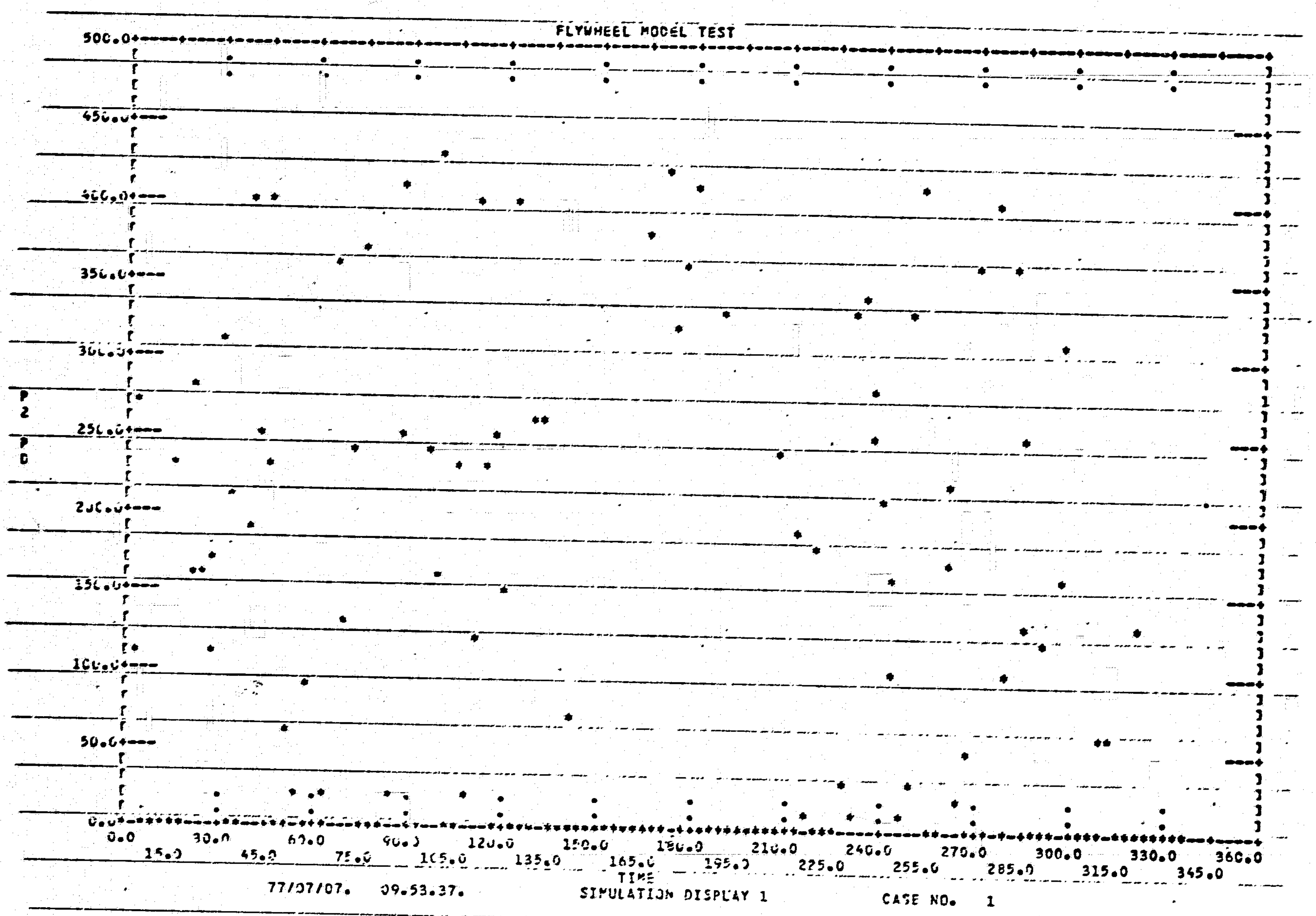


FIGURE 8.2-5 WIND POWER SUPPLIED TO FLYWHEEL STORAGE

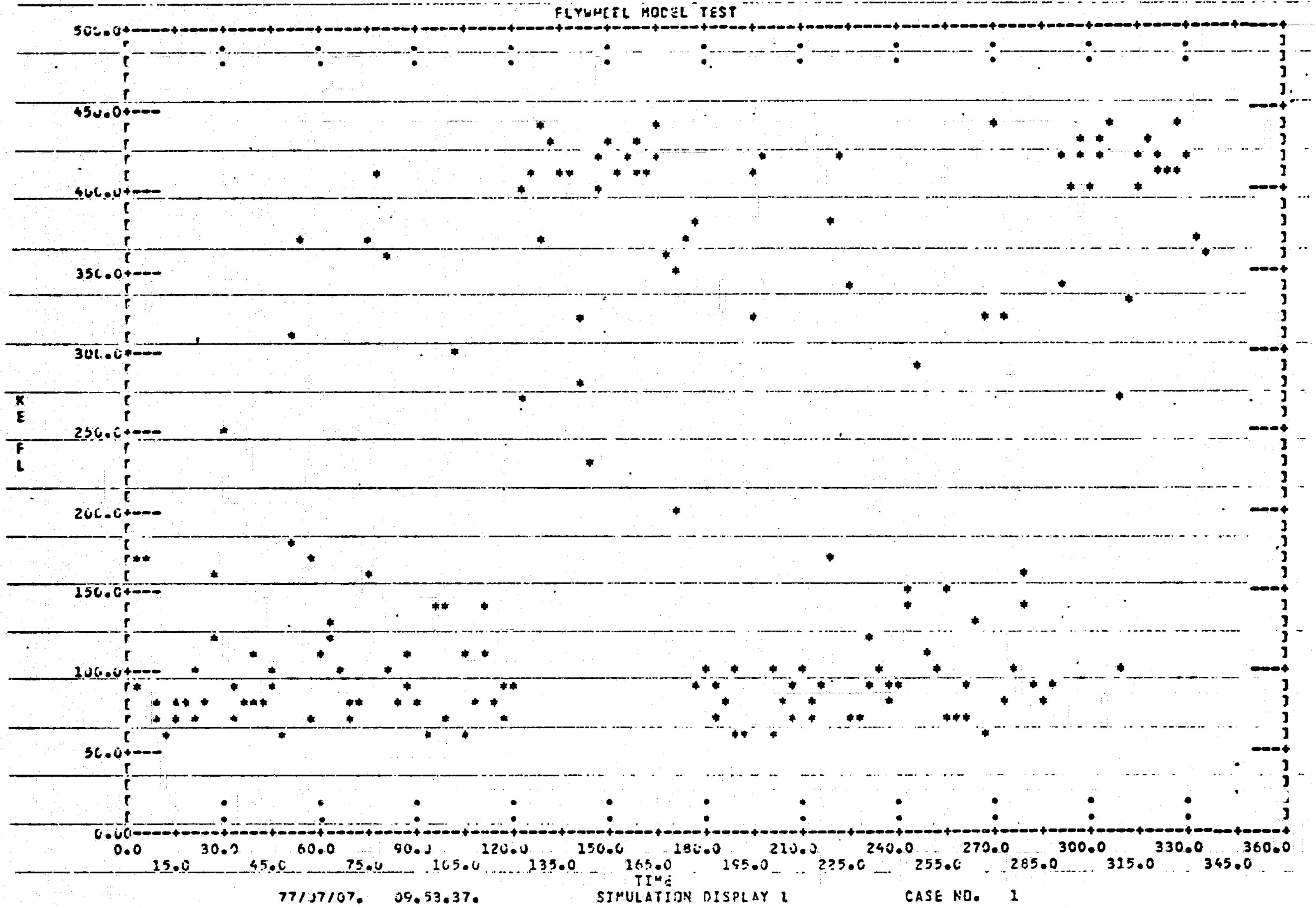
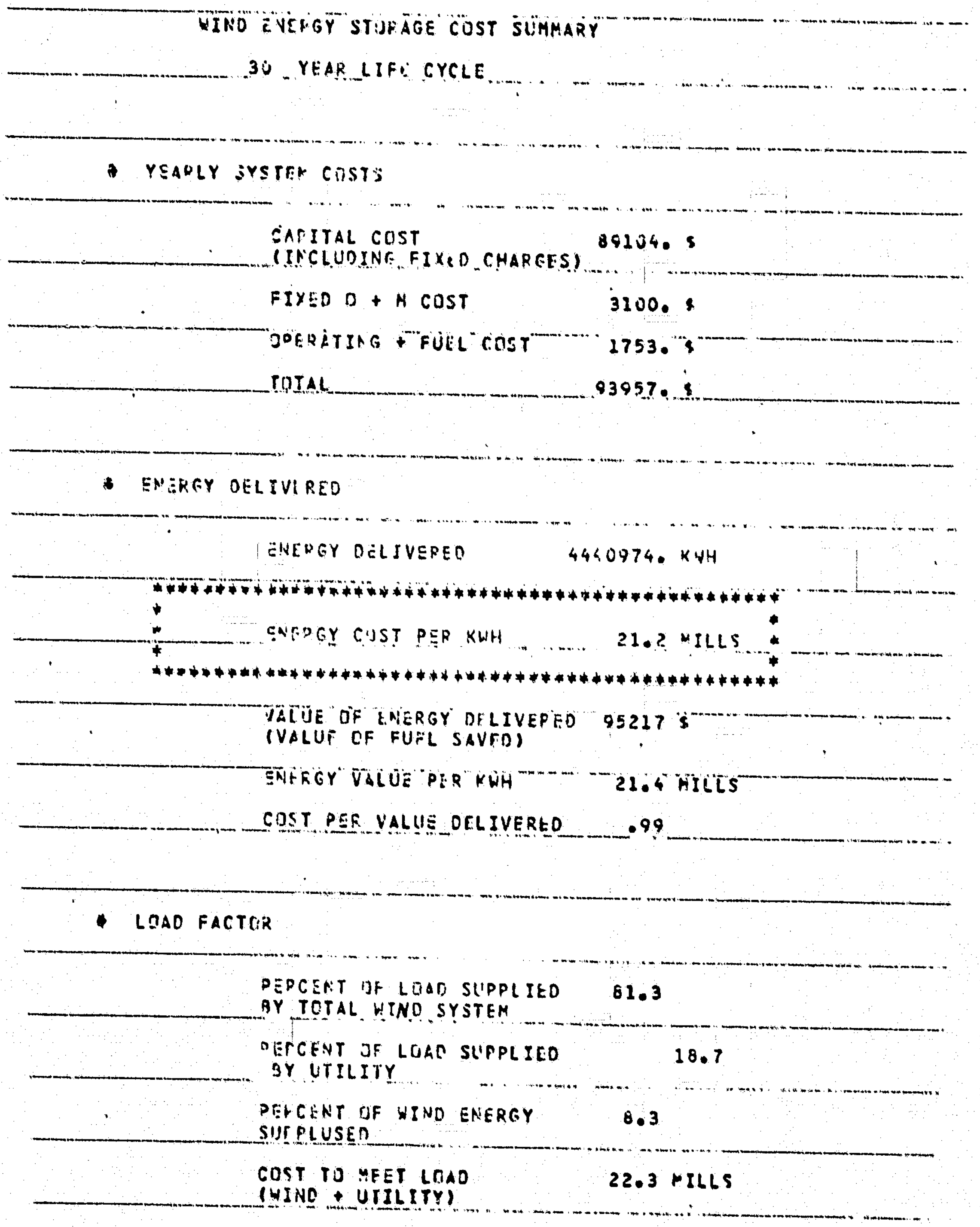


FIGURE 8.2-6 FLYWHEEL KINETIC ENERGY STORAGE



```
          WIND ENERGY STORAGE COST SUMMARY

                30  YEAR LIFE CYCLE

     *  YEARLY SYSTEM COSTS

                   CAPITAL COST                89104. $
                   (INCLUDING FIXED CHARGES)

                   FIXED O + M COST             3100. $

                   OPERATING + FUEL COST        1753. $

                   TOTAL                       93957. $

     *  ENERGY DELIVERED

                   ENERGY DELIVERED        4440974. KWH
         *****************************************************
         *
         *         ENERGY COST PER KWH          21.2 MILLS   *
         *                                                   *
         *****************************************************
                   VALUE OF ENERGY DELIVERED   95217 $
                   (VALUE OF FUEL SAVED)

                   ENERGY VALUE PER KWH         21.4 MILLS

                   COST PER VALUE DELIVERED      .99

     *  LOAD FACTOR

                   PERCENT OF LOAD SUPPLIED     81.3
                   BY TOTAL WIND SYSTEM

                   PERCENT OF LOAD SUPPLIED          18.7
                    BY UTILITY

                   PERCENT OF WIND ENERGY        8.3
                   SURPLUSED

                   COST TO MEET LOAD            22.3 MILLS
                   (WIND + UTILITY)
```

FIGURE 8.2-7 FLYWHEEL MODEL COST MONITOR OUTPUT

8.3 HYDRO AND THERMAL STORAGE MODEL

Figure 8.3-1 shows the basic model schematic for a model with both thermal and electrical loads. Wind power is supplied first to meet the electrical load, with excess power going into hydro and thermal storage. The thermal load is driven by an ambient temperature component. The electrical load energy value is supplied by a time dependent look-up table. Figure 8.3-2 shows the model input data. The components are ordered according to the flow of information in 8.3-1. Observe that the maximum power input of the power divider is connected up to the wind power output P. The model schematic is shown in Figure 8.3-3.

```
MODEL DESCRIPTION        HYDRO AND THERMAL TEST CASE
LOCATION=77      TI
LOCATION=51      WD      INPUTS=TI
LOCATION=21      WP      INPUTS=WD
LOCATION=33      PD      INPUTS=WP,WP(P=MP),PA(1,1),PIH(2,2),HS(RE=RE,2)
                         INPUTS=TS(2,3),PIT(2,3)
LOCATION=13      MO      INPUTS=PD(2,1)
LOCATION=15      PU      INPUTS=MO
LOCATION=17      HS      INPUTS=PU,PA(RE,2=RE)
LOCATION=45      PIH     INPUTS=HS
LOCATION=19      HT      INPUTS=HS
LOCATION=40      GE      INPUTS=HT
LOCATION=59      PA      INPUTS=GE(2,2),LO(1,0),PIH(4,2)
LOCATION=78      FU      INPUTS=TI(TD=FIN)
LOCATION=80      LO      INPUTS=TI,FU(FO=VE)
LOCATION=63      TS      INPUTS=TL
LOCATION=52      PIT     INPUTS=TS
LOCATION=67      TP      INPUTS=TI
LOCATION=65      TL      INPUTS=TI,TP
LOCATION=1       CM
END OF MODEL
PRINT
```

FIGURE 8.3-2 HYDRO AND THERMAL MODEL INPUT DATA

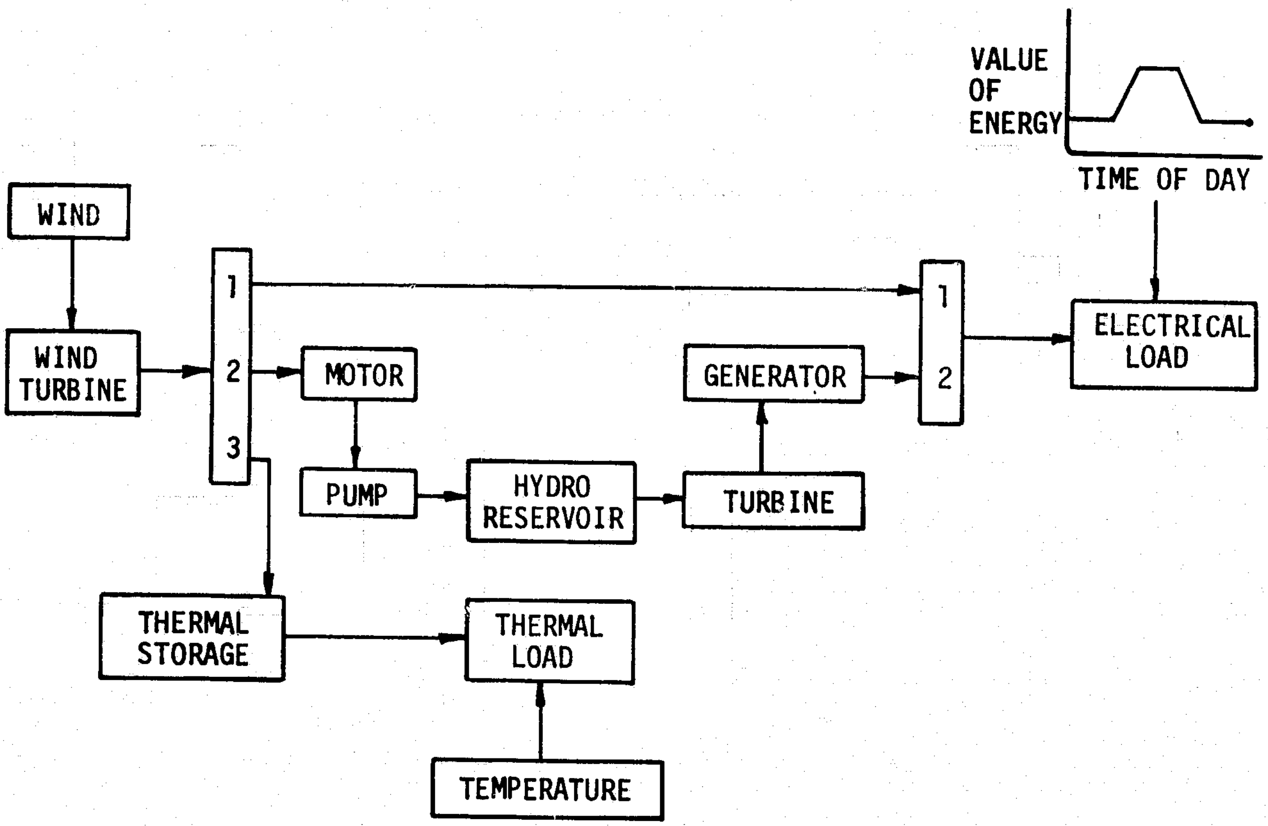


FIGURE 8.3-1: HYDRO AND THERMAL STORAGE EXAMPLE

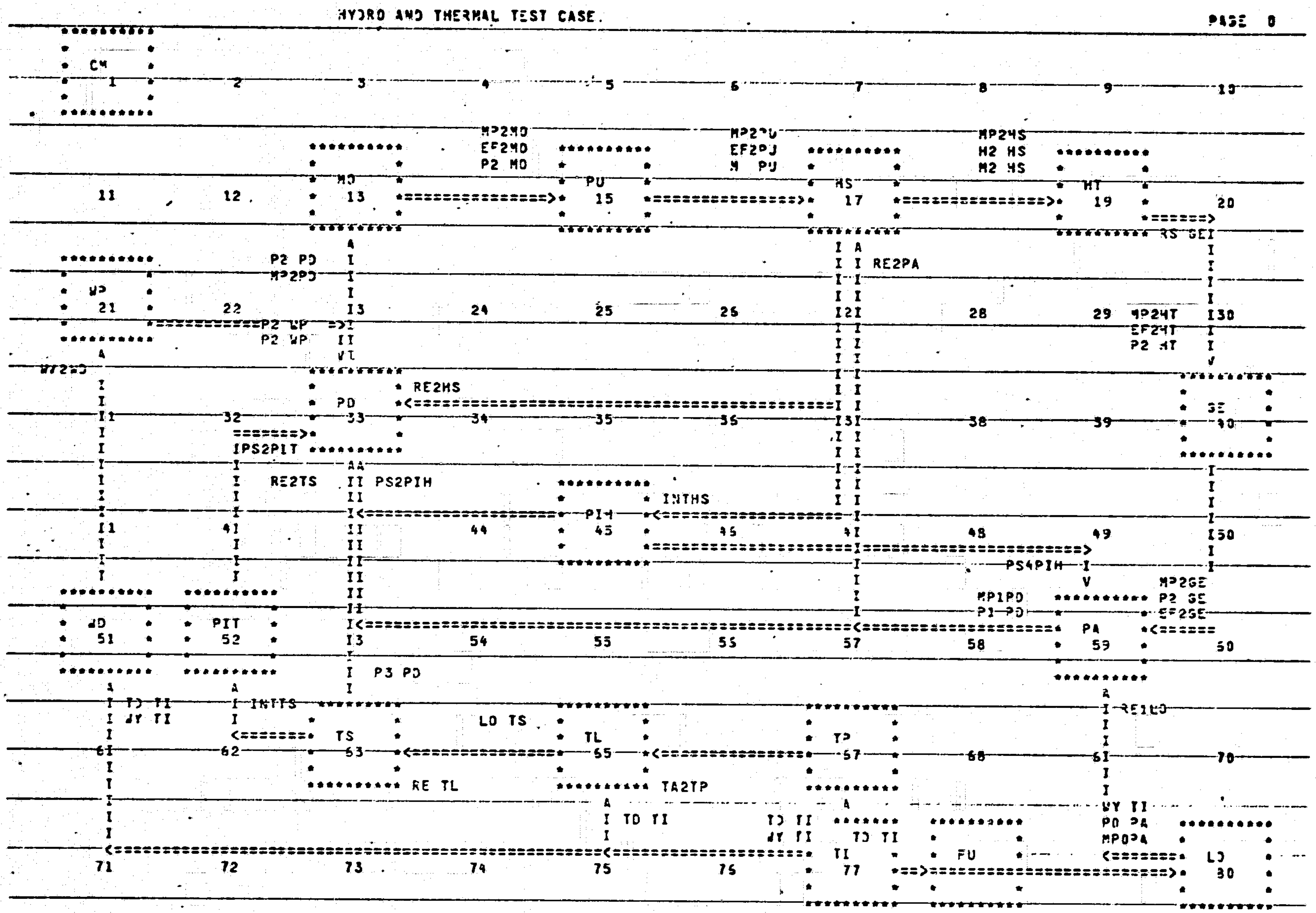


FIGURE 8.3-3 HYDRO AND THERMAL MODEL SCHEMATIC

The input data for a two week simulation with this model is shown in Figure 8.3-4. CYCLES is set to 6 in this model for sufficient iterations to attain steady state in the hydro storage subsystem. The hydro system has much larger capacity and supplies a bigger load than the thermal system in this run. Figures 8.3-5 to 8.3-9 show results of the simulation. Hydro energy storage is shown in 8.3-5. During the week most of the wind energy goes directly to the load except at night. The reservoir builds up to capacity during the weekends. The cumulative percent load delivered by wind and hydro storage is shown in 8.3-6, and averages about 91%. Similarly, thermal energy stored and percent thermal load delivered are shown in 8.3-7 and 8.3-8. The ambient temperature profile for a similar, one week simulation is shown in 8.3-9.

```
PARAMETER VALUES
CYCLES=5.01,TO TI=0,V  WP=400,WVOWP=8,WVIWP=50,DLINES=100.
CC WP=15000,CM WP=1200,PSIPIH=2.,EC WP=.2,CR CM=15,LE CM=30
H1 PU=200,AS HS=3600,MDRHS=80,MO HS=80000,CM HS=1000
H1 HS=200,MDEHS=400000,LE HS=30,F2 PD=.5,F3 PD=.5
RAPGE=200,RSYGE=3600,SR GE=.0333,CC GE=1000,CM GE=120
VC LO=.004,CT LO=4,MN LO=0,STDLO=6,AN FU=-1.
PSLPIT=2.,TS TS=10,VO TS=110,PD TS=100,LE TS=30,MFMTS=10000.
DH TS=.01465,TDETS=2,RS MO=1750,RAPMO=200,CC MO=500,CM MO=100
VE TL=.023,VC TL=40.,CT TP=12,MN TP=0,STDTP=5.
TABLE,PW WP=10
8,10,12,14,16,18,20,21.53,25,30
25.6,50.1,85.5,137.4,205.1,292.,400.6,500,782.8,800
TABLE,PY WD=13
0.,4.33,8.67,13.,17.33,21.67,26.,30.33,34.67,39.,43.33,47.67,52.
55,67,68,55,51,55,51,49,49,52,56,61,65
TABLE,PD WD=7
0,4,8,12,16,20,24
10,12,14,16,14,12,10
TABLE,DF WD=15
0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15
5,44,160,380,480,512,440,376,307,270,148,76,40,22,9,3
TABLE,HT TS=4
.00879,.025491,.047371,.064072
90,147,147,204
TABLE,PD LD=17
0,1.5,3,4.5,6,7.5,9,10.5,12
13.5,15,16.5,18,19.5,21,22.5,24
450,360,372,330,450,660,810,798,804
690,708,699,702,750,708,570,450
TABLE,PW LD=7
1,2,3,4,5,6,7
1,1.,.9,.9,.9,.6,.5
TABLE,PY LD=6
0,10,20,30,40,52
225,194,180,174,194,226
TABLE, FTAFU = 6
0,6,10,18,22,24
.019,.019,.028,.028,.019,.019
TABLE,TLDTL=4
0,32,60,100
4.,2.,1.5,1.
TABLE,TWTTL=4
0,6,18,24
.4,1.,1.,.4
TABLE,PD TP=9
0,3,6,9,12,15,18,21,24
45,45,48,55,62,64,56,48,46
TABLE,PY TP=5
0,13,26,39,52
40,50,75,65,40
INITIAL CONDITIONS, MA HS=1600000,E  TS=600
PRINTER PLOTS,DISPLAY1
WV2WD,VS,TIME
P1 PD,VS,TIME
M  PU,VS,TIME
DISPLAY2
E  HS,VS,TIME

H2 HS,VS,TIME
RE2HS,VS,TIME
DISPLAY3
PC LO,VS,TIME
P3 PD,VS,TIME
E  TS,VS,TIME
RELLO,VS,TIME
DISPLAY4
LD TS,VS,TIME
PC TL,VS,TIME
TA2TP,VS,TIME
FD FU,VS,TD TI
TINC=.50,TMAX=336.,PRATE=6,PRINT CONTROL=3,INT MODE=3,OUTRATE=4
TITLE = HYDRO AND THERMAL TEST
SIMULATE
```

FIGURE 8.3-4 HYDRO AND THERMAL SIMULATION DATA

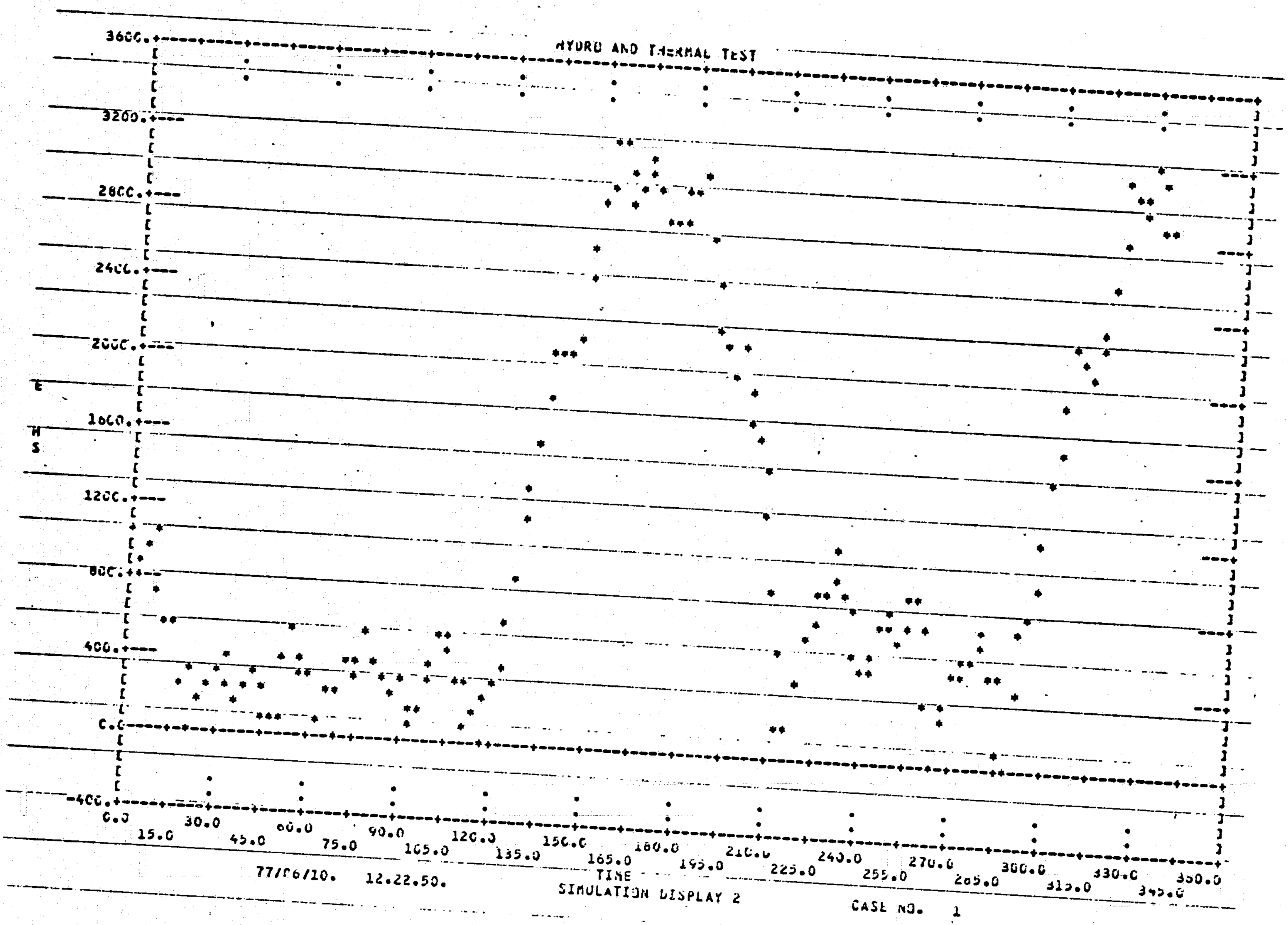


FIGURE 8.3-5 HYDRO RESERVOIR ENERGY STORAGE

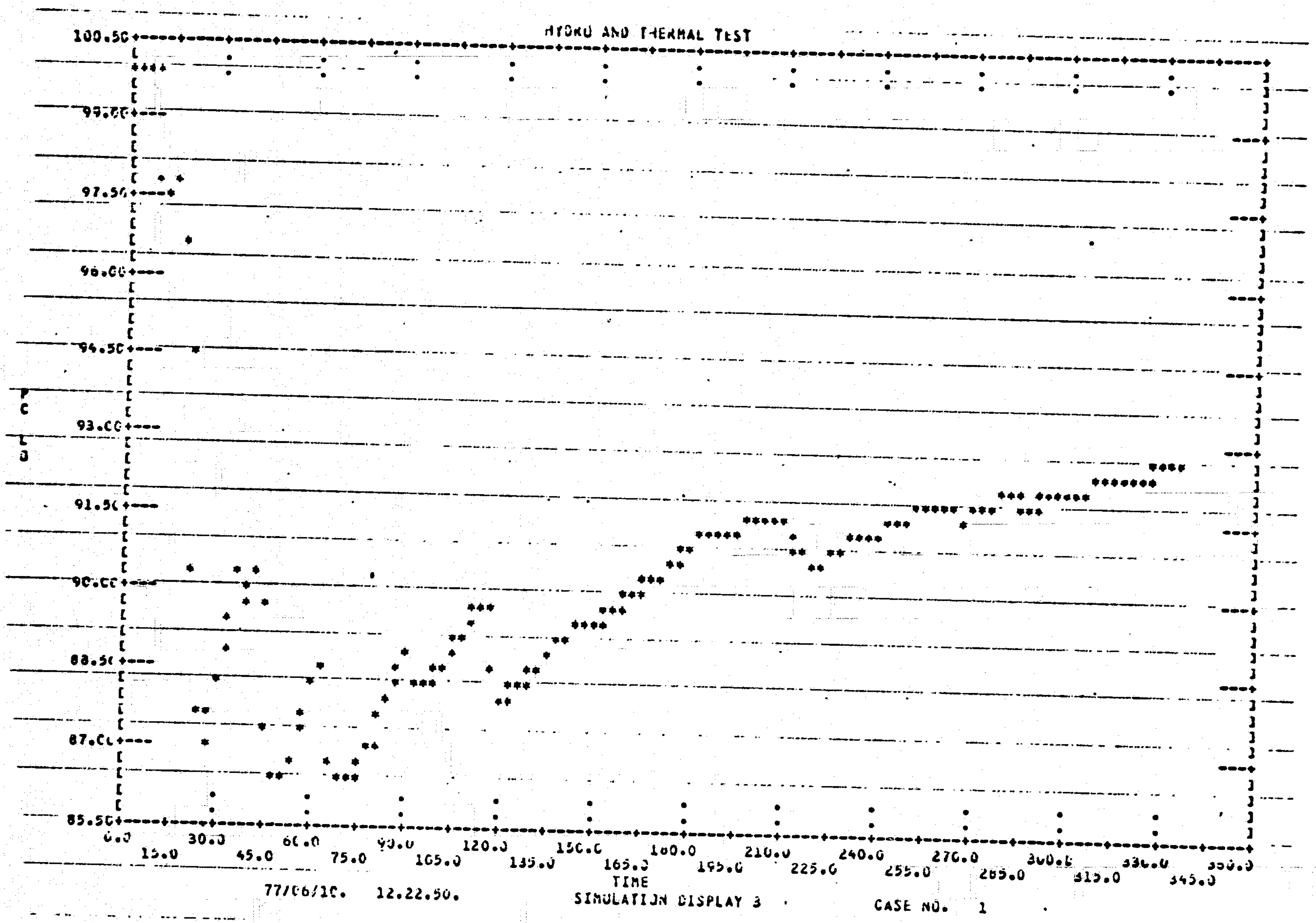


FIGURE 8.3-6 PERCENT CUMULATIVE LOAD DELIVERED

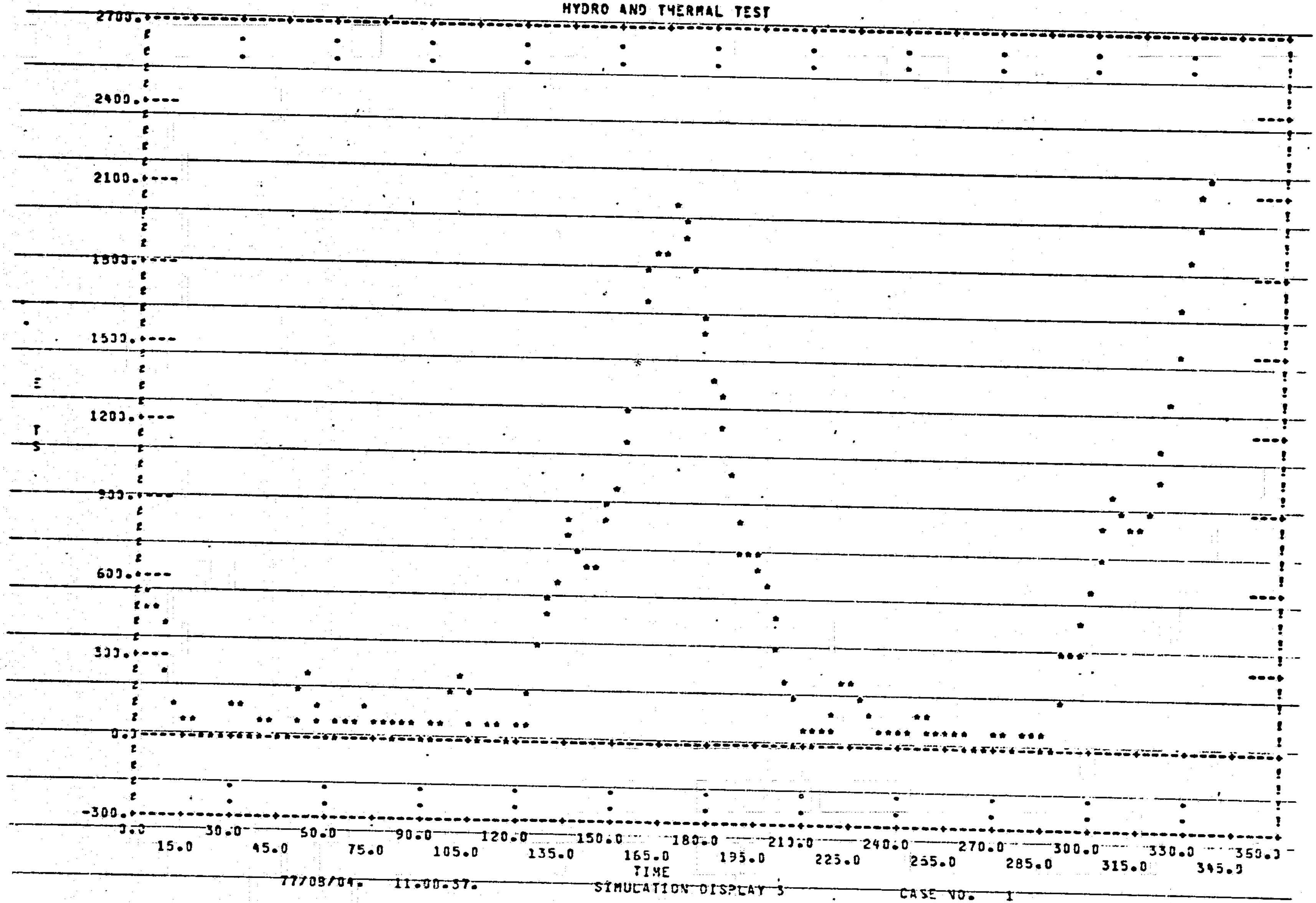


FIGURE 8.3-7 THERMAL ENERGY STORAGE

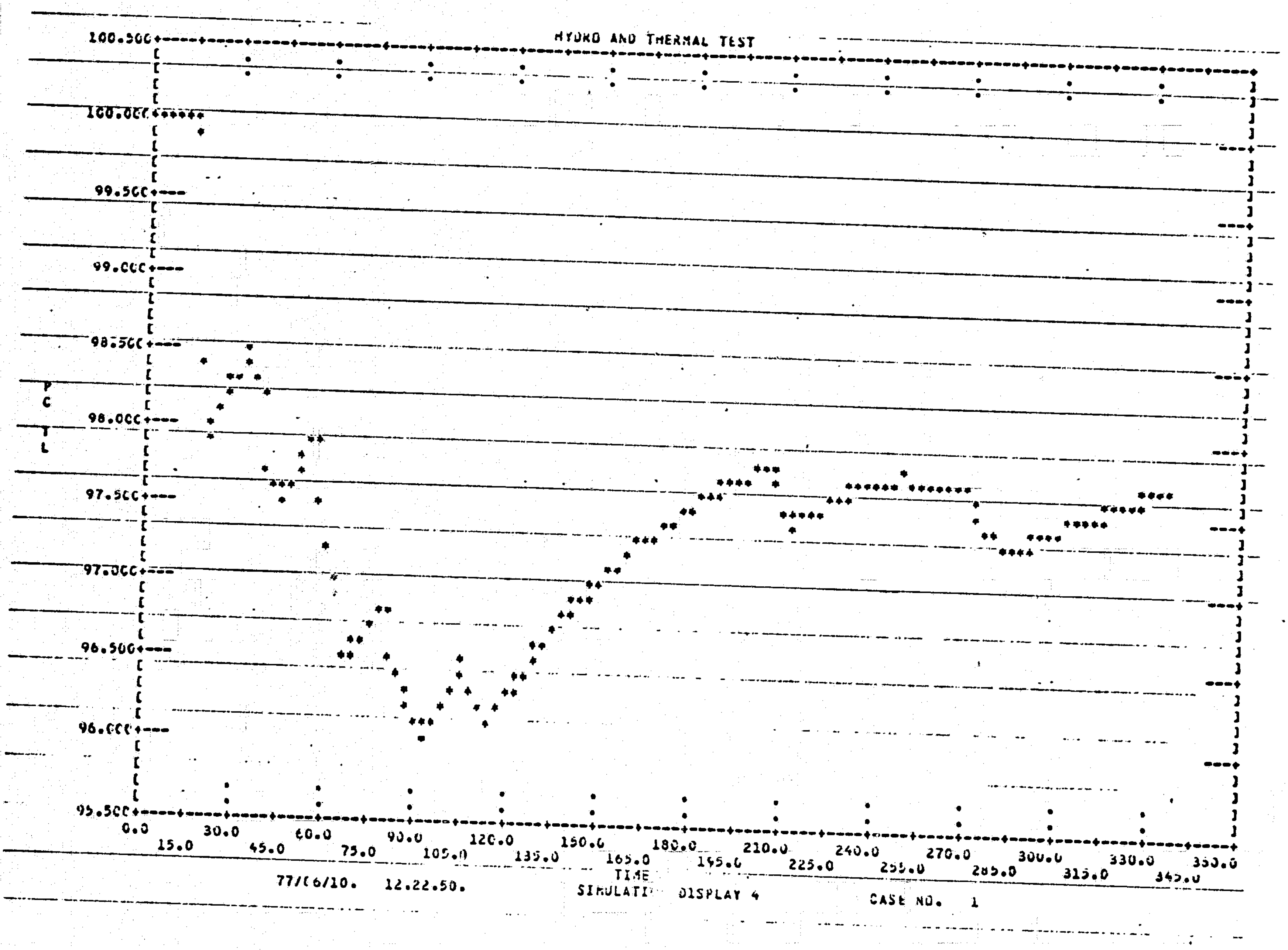


FIGURE 8.3-8 PERCENT CUMULATIVE THERMAL LOAD DELIVERED

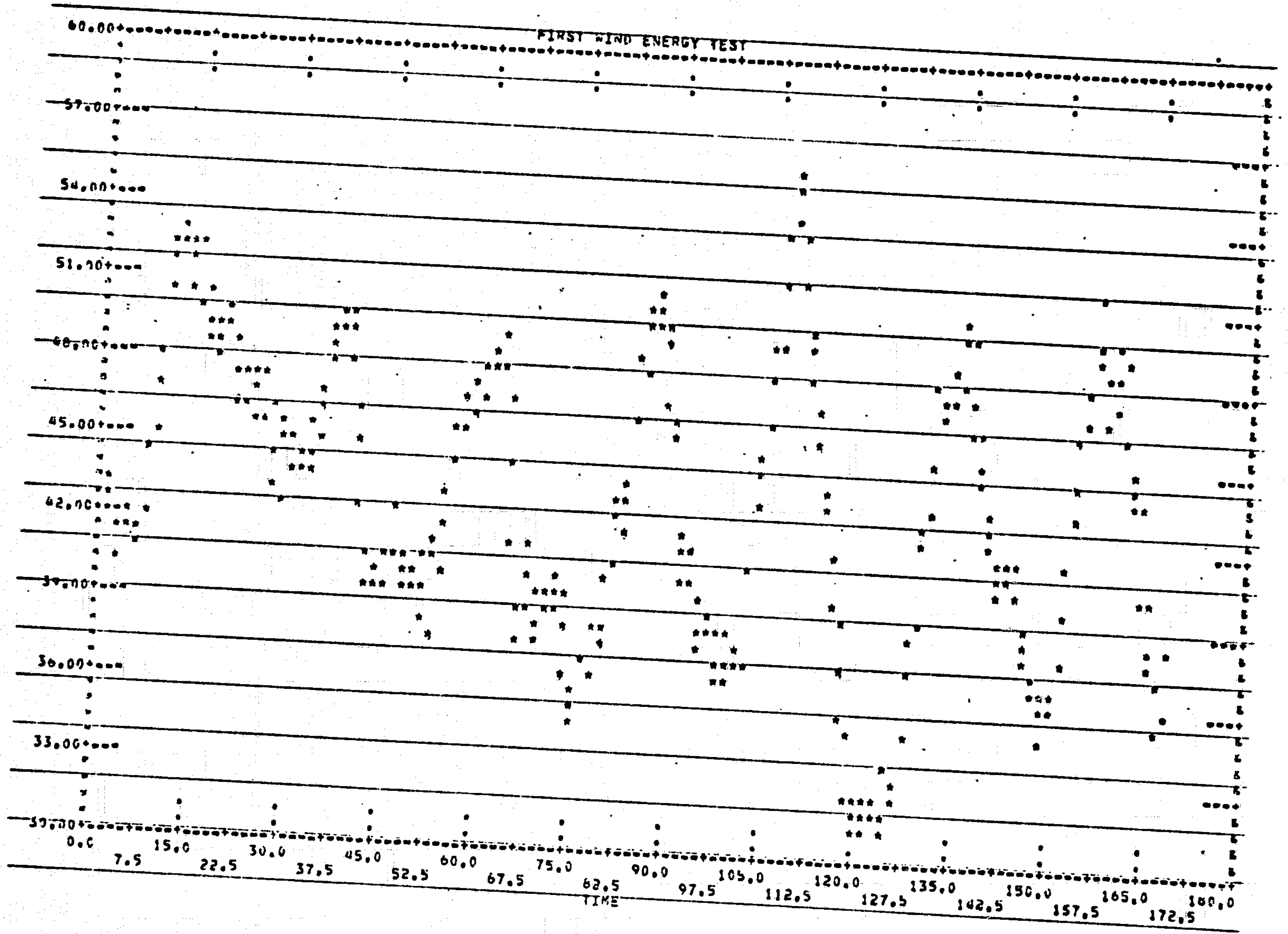


FIGURE 8.3-9 AMBIENT TEMPERATURE SIMULATION OVER ONE WEEK

8.4 PNEUMATIC STORAGE MODEL

Figure 8.4-1 shows the simplified schematic for the pneumatic storage model. For simplicity the motor and generator have been omitted from the pneumatic storage subsystem. A burner is used if needed to heat the exiting air to the turbine. The heat exchanger is a phase change medium. Figure 8.4-2 shows the input data for this model.

```
MODEL DESCRIPTION        PNEUMATIC STORAGE TEST CASE
LOCATION=1        TI
LOCATION=21       WD     INPUTS=TI
LOCATION=51       WP     INPUTS=WD
LOCATION=5        TP     INPUTS=TI
LOCATION=43       PD     INPUTS=WP,WP(P=MP),PA(1,1),PI(2,2),CS(RE=RE,2)
LOCATION=64       UT     INPUTS=PD(SP=P)
LOCATION=15       CO     INPUTS=PD(2,1),TP
LOCATION=17       HX     INPUTS=CO,TP,CS
LOCATION=47       CS     INPUTS=HX,PA(RE,2=RE)
LOCATION=36       PI     INPUTS=CS
LOCATION=49       HY     INPUTS=CS,HX
LOCATION=59       BN     INPUTS=HY
LOCATION=80       TU     INPUTS=BN,TP,CS(PR=PS)
LOCATION=76       PA     INPUTS=TU(2,2),LO(1,0),PI(4,2),UT(2,3)
LOCATION=72       LO     INPUTS=TI
LOCATION=71       CM
END OF MODEL
LIST STANDARD COMPONENTS
PRINT
```

FIGURE 8.4-2 PNEUMATIC STORAGE MODEL INPUT DATA

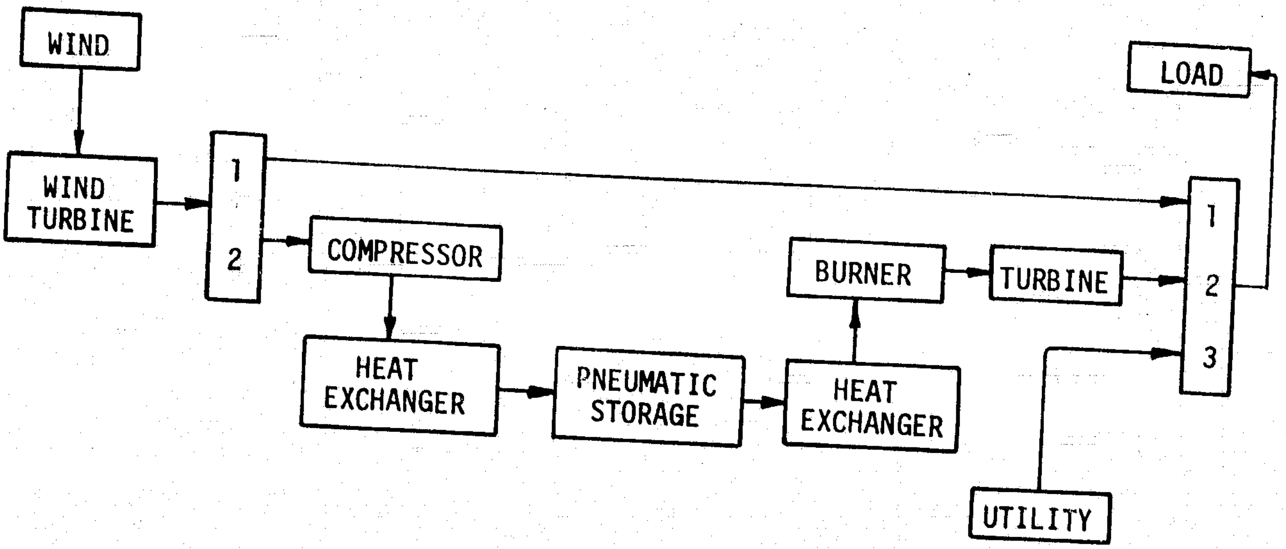
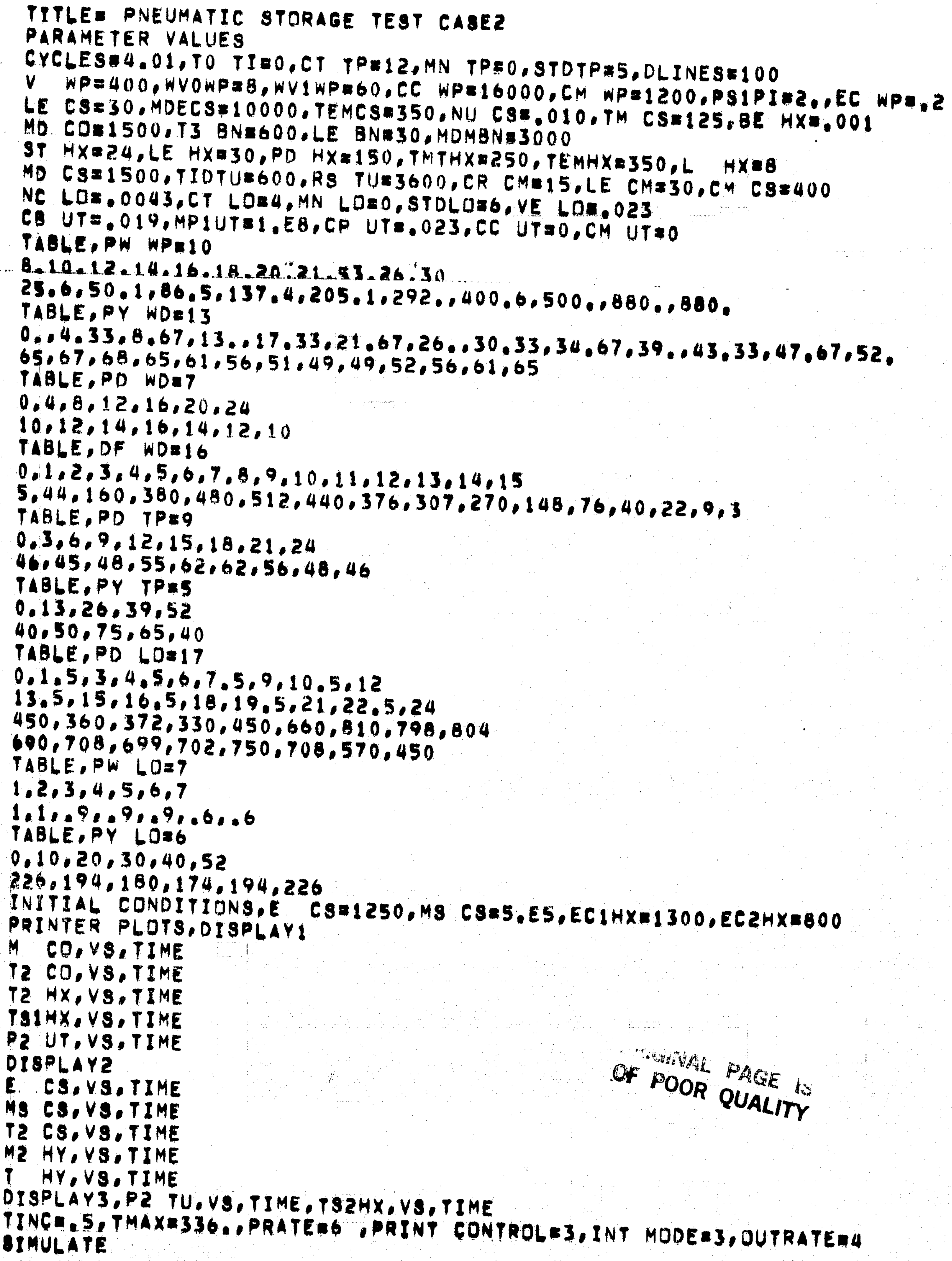


FIGURE 8.4-1: PNEUMATIC STORAGE EXAMPLE

The input data for a two week simulation is shown in 8.4-3. In order to keep the air entering the storage cavern from overheating, a fairly large leakage coefficient (NU = 0.01) is assumed. Hence the storage cavern loses about 2/3 of its heat energy every four days. The load constant NC LO can be adjusted to balance wind energy to the load so that weekly air mass flow in and out of the cavern is balanced. The initial values for the CS and HX states were chosen on the basis of an earlier one week simulation. Figures 8.4-4 to 8.4-8 show results of this simulation. Figure 8.4-4 shows the average temperature of the heat exchanger storage medium for the 'cool' cell. The initial temperature at the beginning of the simulation is a little too cool since the temperature rises to about 400° during the weekends. Phase change in this medium is indicated by the constant temperature intervals at 250°. Figure 8.4-5 shows the air temperature exiting from the heat exchanger into the cavern. During the week this temperature is generally held below 200° but may exceed 350° during the weekend. Figure 8.4-6 shows the air mass stored in the cavern. In this simulation wind power generation exceeded that for the load and thus there is a gradual buildup of air mass in the cavern. The temperature of the stored air mass is shown in Figure 8.4-7. There is about a 10° fluctuation in temperature each week in this case. The last figure, 8.4-8 shows the air temperature exiting from the heat exchanger to the burner. Neglecting the influence of the inital conditions, the average temperature is about 550° and thus a burner is probably not required for this system.



```
TITLE= PNEUMATIC STORAGE TEST CASE2
PARAMETER VALUES
CYCLES=4.01,TO TI=0,CT TP=12,MN TP=0,STDTP=5,DLINES=100
V  WP=400,WV0WP=8,WV1WP=60,CC WP=16000,CM WP=1200,PS1PI=2.,EC WP=.2
LE CS=30,MDECS=10000,TEMCS=350,NU CS=.010,TM CS=125,BE HX=.001
MD CO=1500,T3 BN=600,LE BN=30,MDMBN=3000
ST HX=24,LE HX=30,PD HX=150,TMTHX=250,TEMHX=350,L  HX=8
MD CS=1500,TIDTU=600,RS TU=3600,CR CM=15,LE CM=30,CM CS=400
NC LO=.0043,CT LO=4,MN LO=0,STDLO=6,VE LO=.023
CB UT=.019,MP1UT=1.E8,CP UT=.023,CC UT=0,CM UT=0
TABLE,PW WP=10
8,10,12,14,16,18,20,21,23,26,30
25.6,50.1,86.5,137.4,205.1,292.,400.6,500.,880.,880.
TABLE,PY WD=13
0.,4.33,8.67,13.,17.33,21.67,26.,30.33,34.67,39.,43.33,47.67,52.
65,67,68,65,61,56,51,49,49,52,56,61,65
TABLE,PD WD=7
0,4,8,12,16,20,24
10,12,14,16,14,12,10
TABLE,DF WD=16
0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15
5,44,160,380,480,512,440,376,307,270,148,76,40,22,9,3
TABLE,PD TP=9
0,3,6,9,12,15,18,21,24
46,45,48,55,62,62,56,48,46
TABLE,PY TP=5
0,13,26,39,52
40,50,75,65,40
TABLE,PD LO=17
0,1.5,3,4.5,6,7.5,9,10.5,12
13.5,15,16.5,18,19.5,21,22.5,24
450,360,372,330,450,660,810,798,804
690,708,699,702,750,708,570,450
TABLE,PW LO=7
1,2,3,4,5,6,7
1,1,.9,.9,.9,.6,.6
TABLE,PY LO=6
0,10,20,30,40,52
226,194,180,174,194,226
INITIAL CONDITIONS,E  CS=1250,MS CS=5.E5,EC1HX=1300,EC2HX=800
PRINTER PLOTS,DISPLAY1
M  CO,VS,TIME
T2 CO,VS,TIME
T2 HX,VS,TIME
TS1HX,VS,TIME
P2 UT,VS,TIME
DISPLAY2
E  CS,VS,TIME
MS CS,VS,TIME
T2 CS,VS,TIME
M2 HY,VS,TIME
T  HY,VS,TIME
DISPLAY3,P2 TU,VS,TIME,TS2HX,VS,TIME
TINC=.5,TMAX=336.,PRATE=6 ,PRINT CONTROL=3,INT MODE=3,OUTRATE=4
SIMULATE
```

FIGURE 8.4-3 PNEUMATIC STORAGE SIMULATION DATA

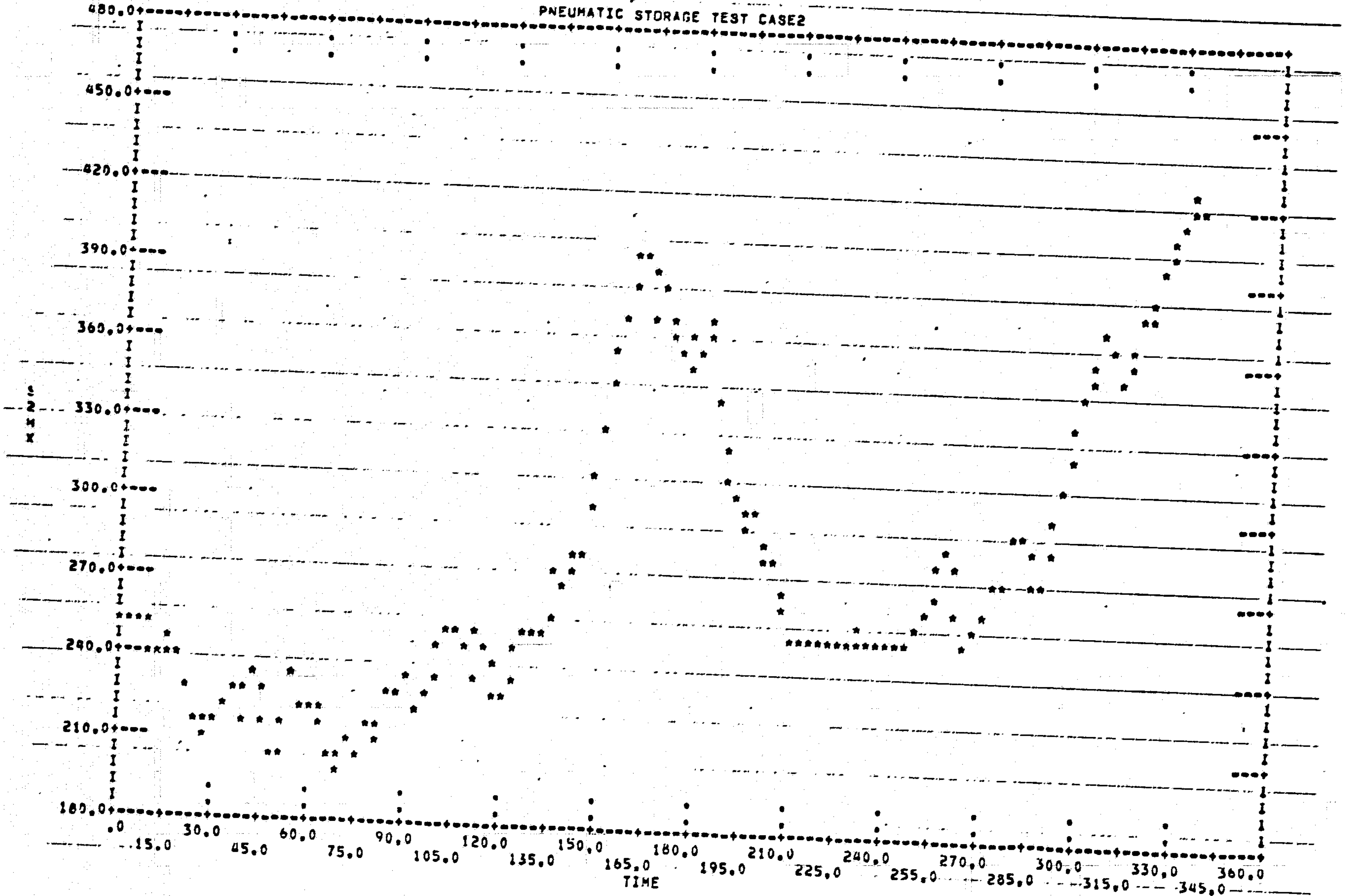


FIGURE 8.4-4 AVERAGE TEMPERATURE IN HEAT EXCHANGER CELL 2

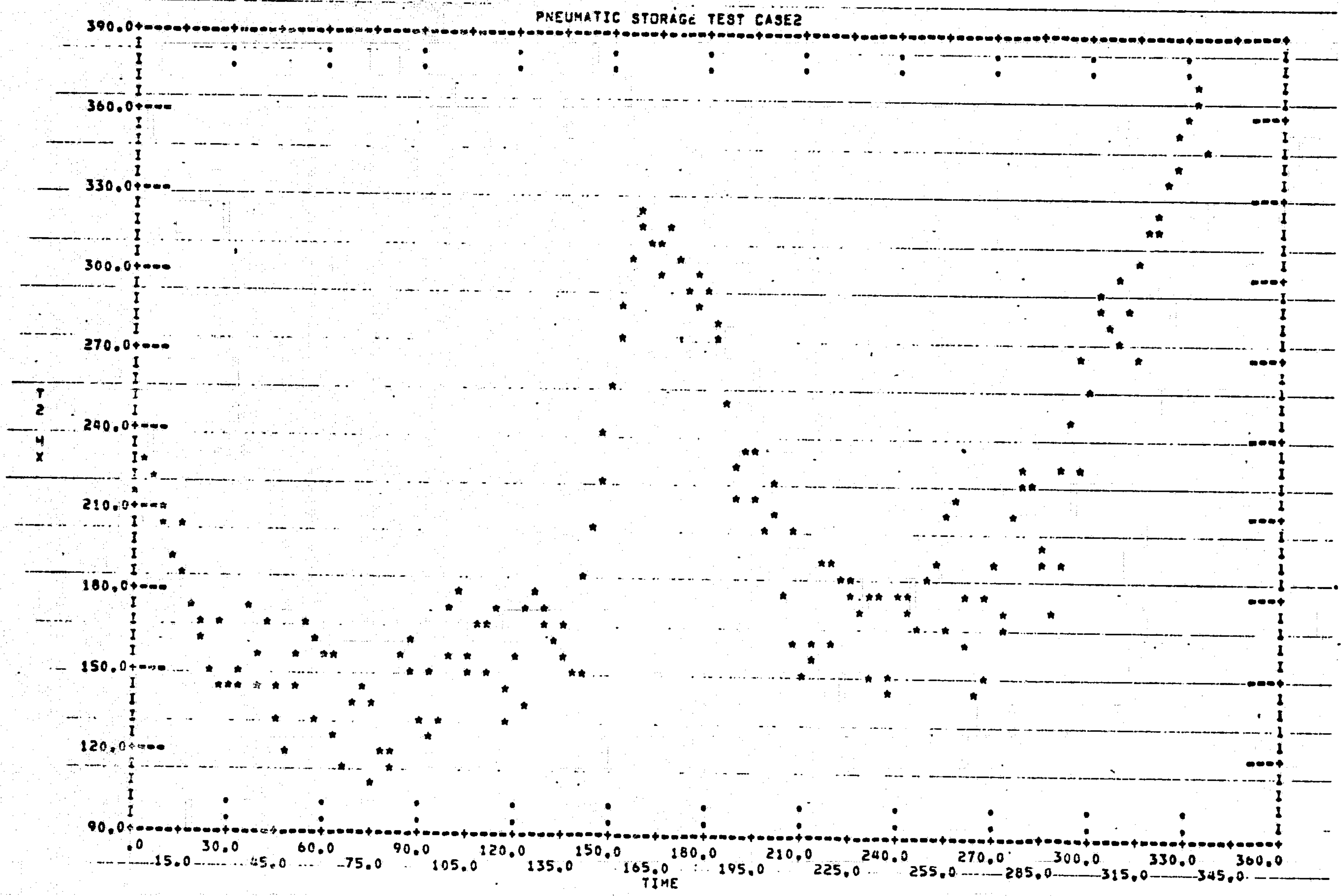


FIGURE 8.4-5 HEAT EXCHANGER OUTLET TEMPERATURE (CHARGING)

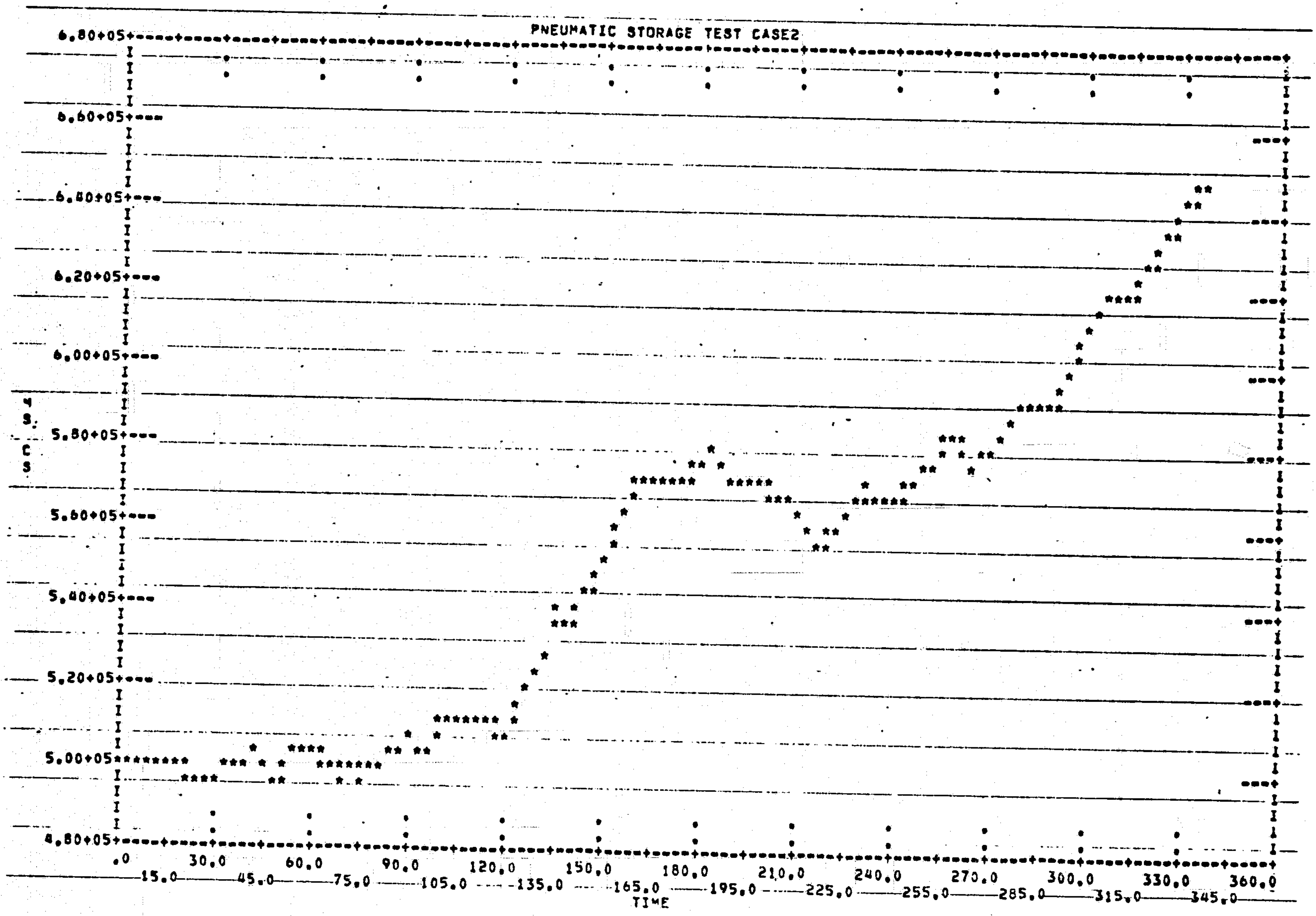


FIGURE 8.4-6 AIR MASS IN PNEUMATIC STORAGE

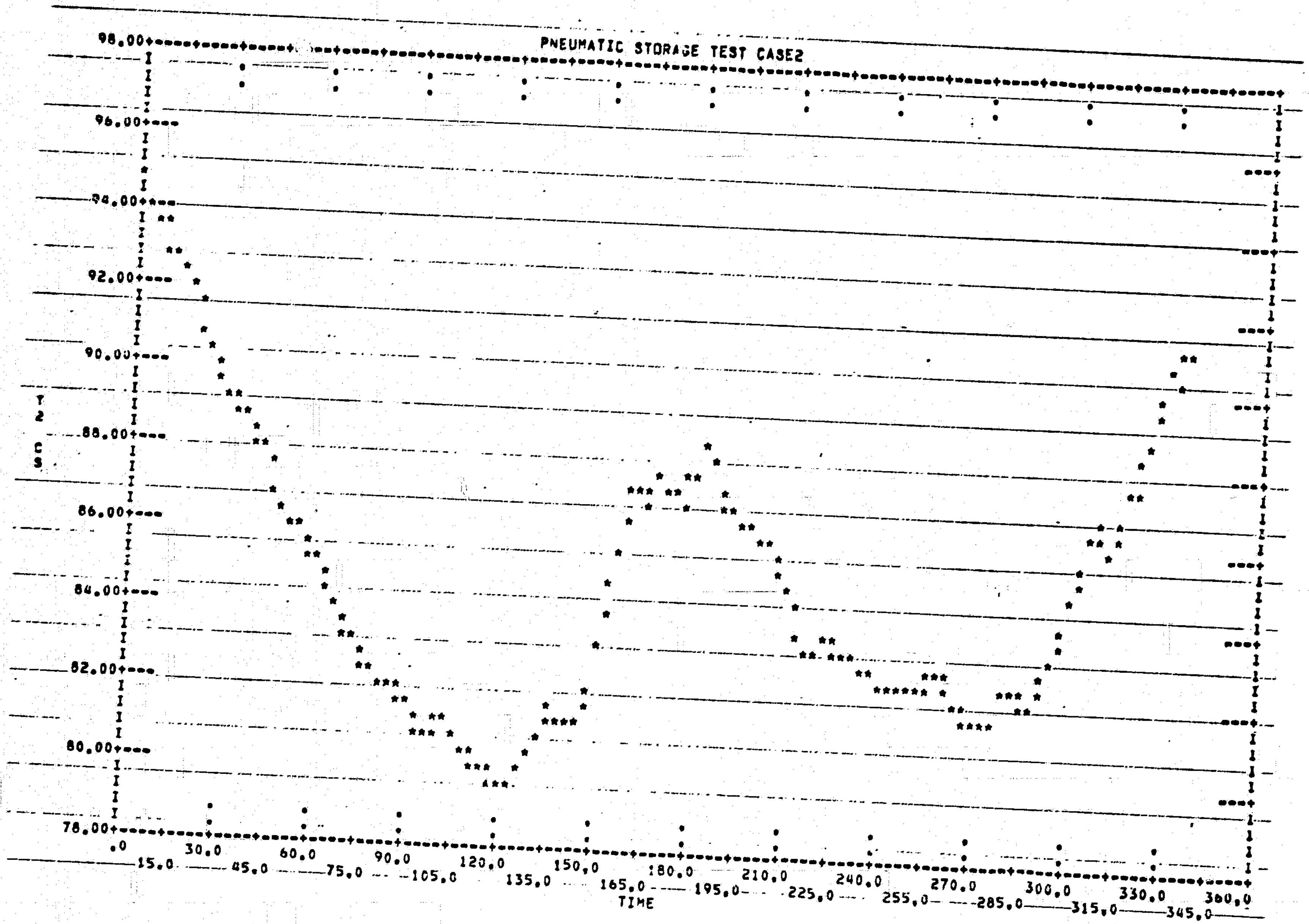


FIGURE 8.4-7 AIR MASS TEMPERATURE IN PNEUMATIC STORAGE VESSEL

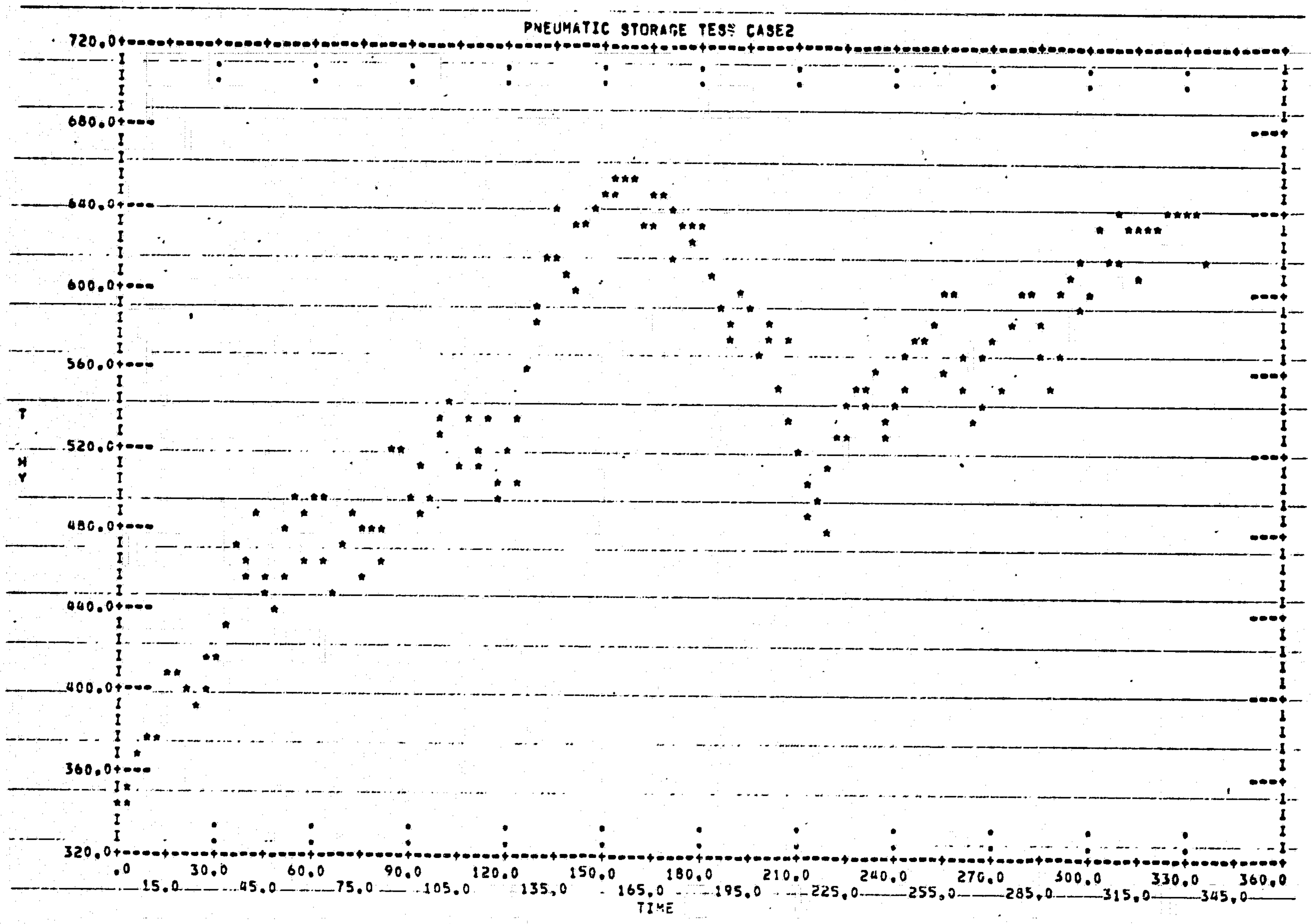


FIGURE 8.4-8 HEAT EXCHANGER OUTLET TEMPERATURE (DISCHARGING)

9.0 SOLAR PHOTOVOLTAIC EXAMPLES

The solar photovoltaic component models added to the SIMWEST library are briefly described and test case results illustrating their use are summarized in this section.

Table 9.0-1 summarizes the characteristics of the solar-photovoltaic components. The environmental data component is designed to read Typical Meteorological Year (TMY) data tapes containing hourly insolation and weather data at 26 U.S. locations. This component can also be used to read other hourly data tapes such as the SOLMET tapes by inputting a user specified format to the model generation program. The solar orientation or tracking component computes the sum of direct beam and global insolation on a flat plate array for fixed orientation and four different beam tracking options. The flat plate and focusing lens collector components provide detailed thermal analyses for determining average solar cell temperature. The collector models, and that of the solar array are based on similar models developed at Sandia Laboratories for the SOLCEL program (Reference [4]). The array component model is a simplified model based on scaling the characteristics of a single solar cell. Array voltage can either be user specified or determined by a maximum power tracker. It should be observed that the above components are coded in SI (metric) units, whereas most of the SIMWEST components are coded in English units. This is generally not a problem since there are at most only a few interconnection variables between the solar-photovoltaic generation components and other SIMWEST components, and these are easily converted using arithmetic components.

The TMY data tapes are currently the best environmental data sources available for simulating typical yearly solar energy system performance. These tapes were extracted from SOLMET data tapes containing rehabilitated hourly solar and meteorological observation data over a period of many years at each observation site. Each Typical Meteorological Year was created by statistical selection of a typical meteorological month for each calendar month in the long term data base and catenating the 12 months to form a TMY. All of the TMY data files are available for use by a SIMWEST user. He thus has access to a high quality environmental data base for solar energy simulations and system analyses.

TABLE 9.0-1 SOLAR-PHOTOVOLTAIC COMPONENTS

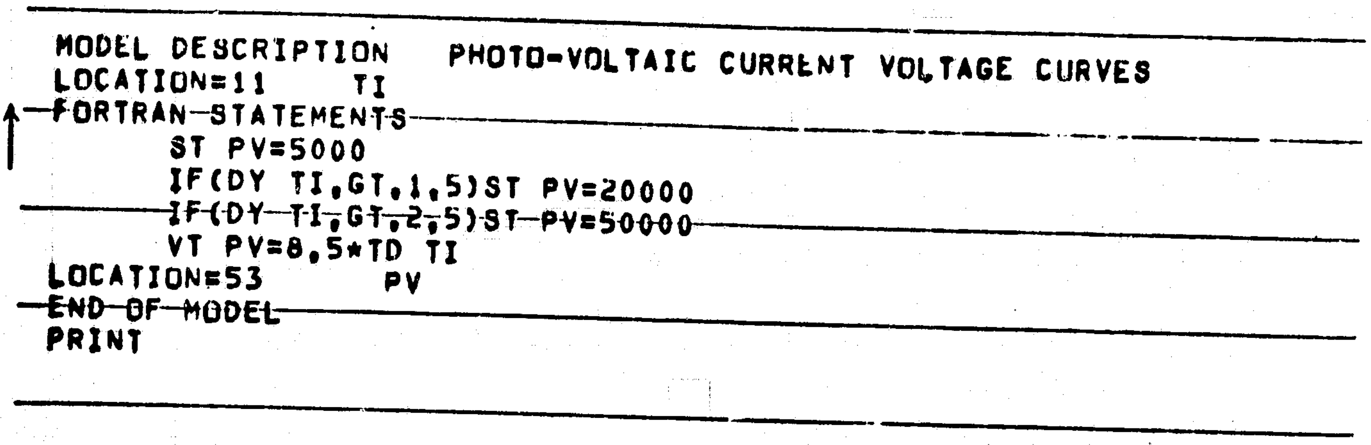
COMPONENT	SYMBOL	PURPOSE
● ENVIRONMENTAL DATA (TAPE)	ED	READ DOE SOLAR INSOLATION AND WEATHER DATA TYPICAL METEOROLOGICAL YEAR TAPE
● SOLAR ORIENTATION (TRACKING)	SO	SOLAR INSOLATION ON TILTED FLAT PLATE ARRAY (FIVE OPTIONS)
● FLAT PLATE COLLECTOR	FP	FLAT PLATE THERMAL MODEL WITH FLUID AND PASSIVE COOLING OPTIONS
● FOCUSING LENS COLLECTOR	FO	FRESNEL LENS THERMAL MODEL WITH FLUID AND PASSIVE COOLING OPTIONS
● PHOTOVOLTAIC ARRAY	PV	CONVERTS SOLAR INSOLATION TO D.C. ELECTRICAL POWER. MAXIMUM POWER TRACKER OR USER SPECIFIED VOLTAGE

9.1 PHOTOVOLTAIC MODEL TEST CASE

The input data for the photovoltaic model test case is shown in Figure 9.1-1. The purpose of this model is to obtain characteristic current voltage curves for the default solar array parameters. Fortran statements are used in the model generation data to let the terminal voltage range between 0 and 204 volts for solar insolation values of 5, 20, and 50 suns (1 sun = 1000 w/m^2). Cell temperature is specified at 25°C for the first simulation and 55°C for the second. Figure 9.1-2 shows the current voltage curves and Figure 9.1-3 shows power voltage cross plots at the lower cell temperature and for the three solar insolation levels. These curves verify the physical characteristics of the solar cell model. It may be noted in these figures that current and output power become negative when the specified voltage exceeds the array open circuit voltage. Individual cell characteristics may be obtained by dividing voltage by 300 (default number of cells in series) and by dividing current by 500 (default number of cells in parallel).

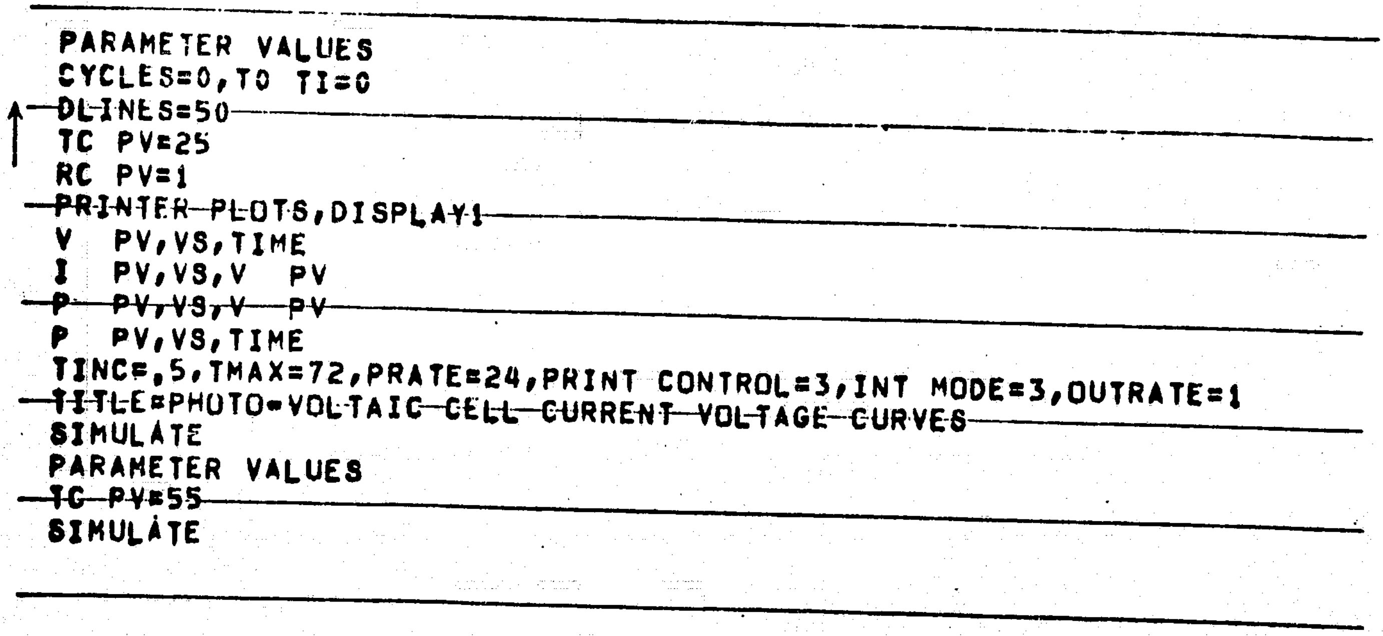
9.2 FLAT PLATE COLLECTOR MODEL

The input data for the flat plate model test case is shown in Figure 9.2-1. The purpose of this model is to illustrate water and wind cooling of the collector and to test the tracking options of the orientation component **SO**. There are six 1-1/2 day simulation runs. The first run uses water cooling (CMOFP=2), a single glass cover over the front plate and insolation on the back. The second run uses passive cooling (CMOFP=0), no plate insolation and fins on the back to cool the collector. In the first two runs, the collector is tilted and has a fixed, southward facing orientation (MO SO=1). The last four runs are similar to run 2 except different tracking options are utilized.



```
MODEL DESCRIPTION   PHOTO-VOLTAIC CURRENT VOLTAGE CURVES
LOCATION=11     TI
FORTRAN STATEMENTS
      ST PV=5000
      IF(DY TI,GT,1,5)ST PV=20000
      IF(DY TI,GT,2,5)ST PV=50000
      VT PV=8,5*TD TI
LOCATION=53       PV
END OF MODEL
PRINT
```

a) Model Generation Input Data



```
PARAMETER VALUES
CYCLES=0,TO TI=0
DLINES=50
TC PV=25
RC PV=1
PRINTER PLOTS,DISPLAY1
V  PV,VS,TIME
I  PV,VS,V  PV
P  PV,VS,V  PV
P  PV,VS,TIME
TINC=,5,TMAX=72,PRATE=24,PRINT CONTROL=3,INT MODE=3,OUTRATE=1
TITLE=PHOTO-VOLTAIC CELL CURRENT VOLTAGE CURVES
SIMULATE
PARAMETER VALUES
TC PV=55
SIMULATE
```

b) Simulation Program Input Data

FIGURE 9.1-1 PV TEST CASE INPUT DATA

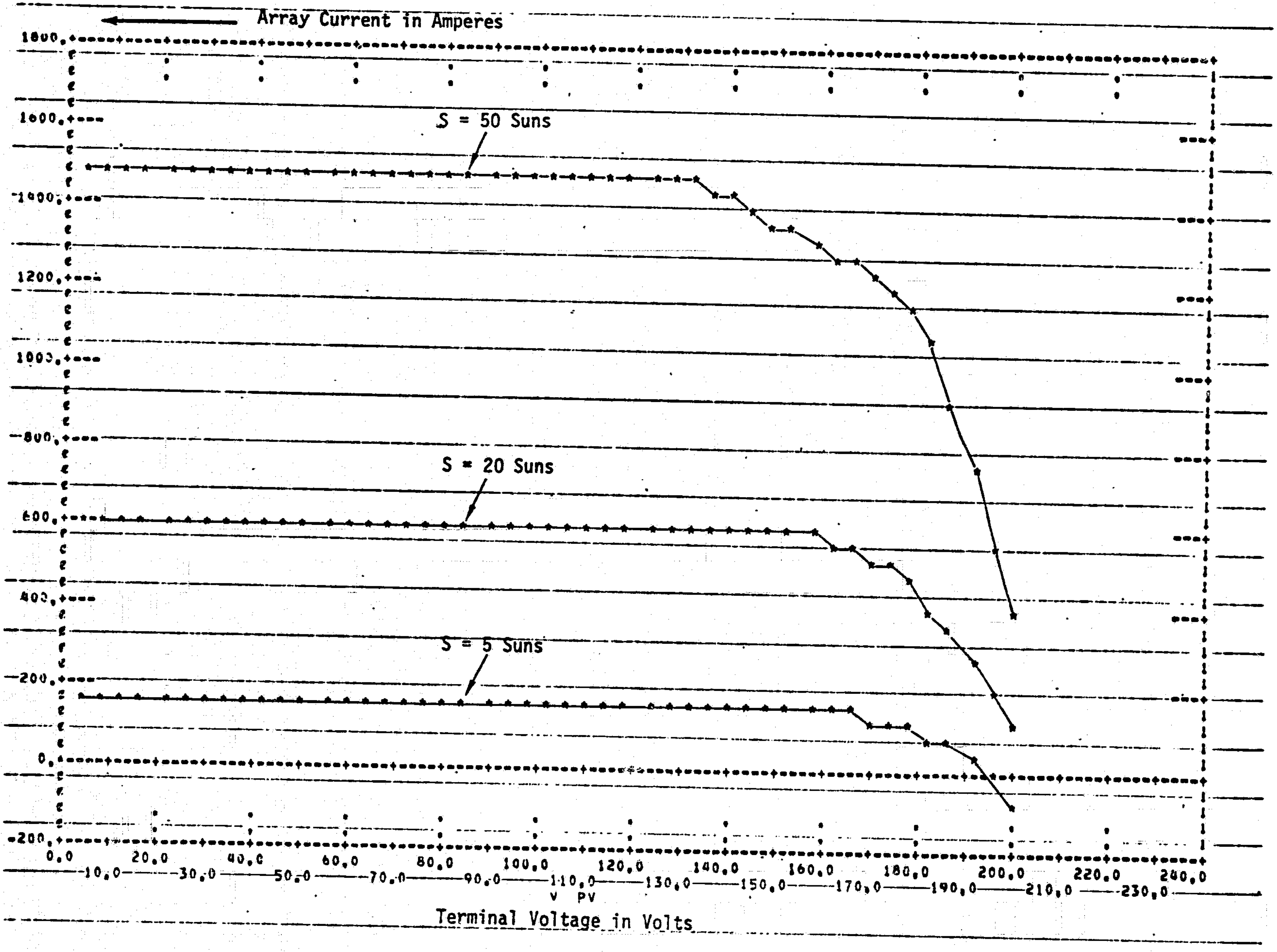


FIGURE 9.1-2 SOLAR ARRAY CHARACTERISTIC CURRENT - VOLTAGE CURVES

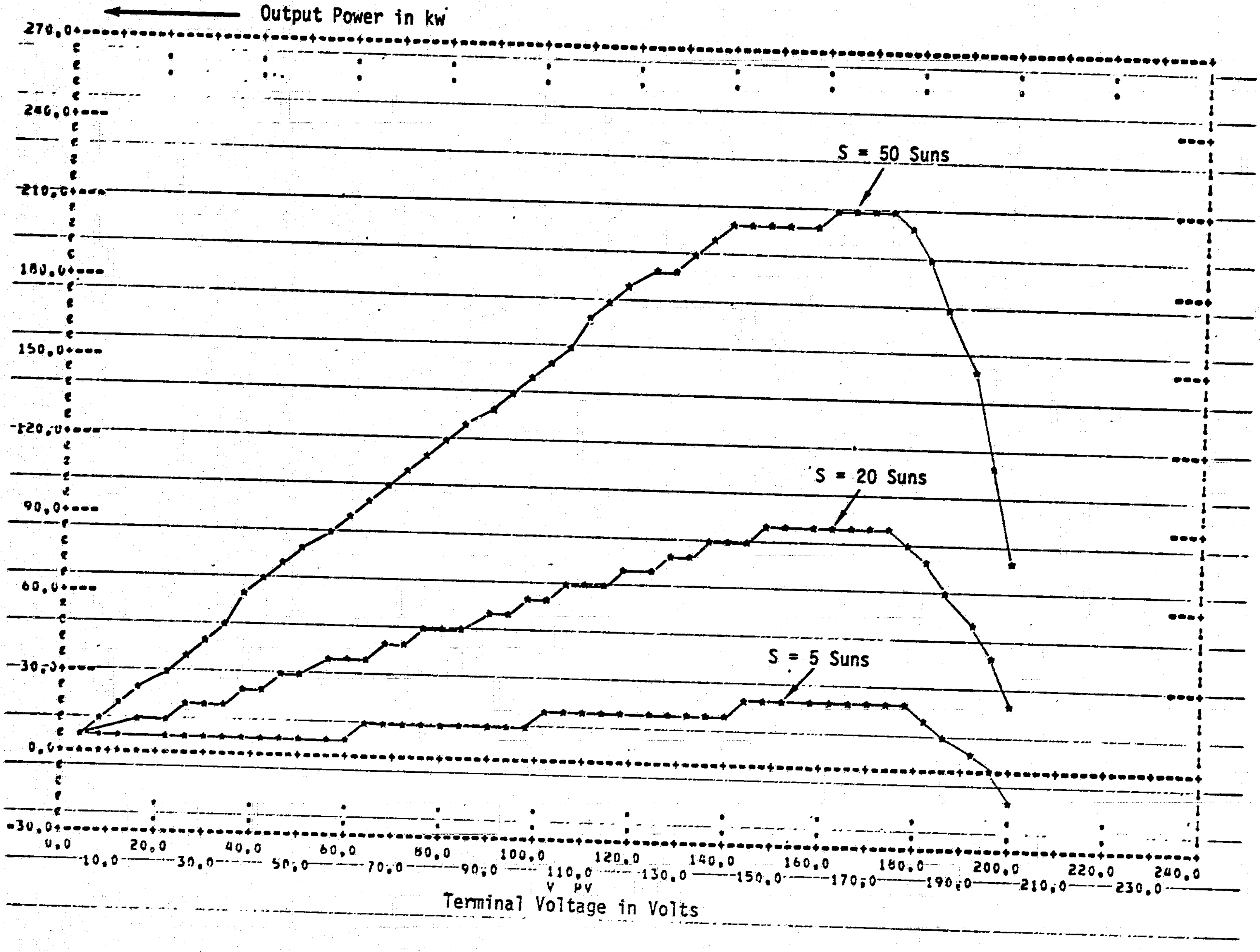
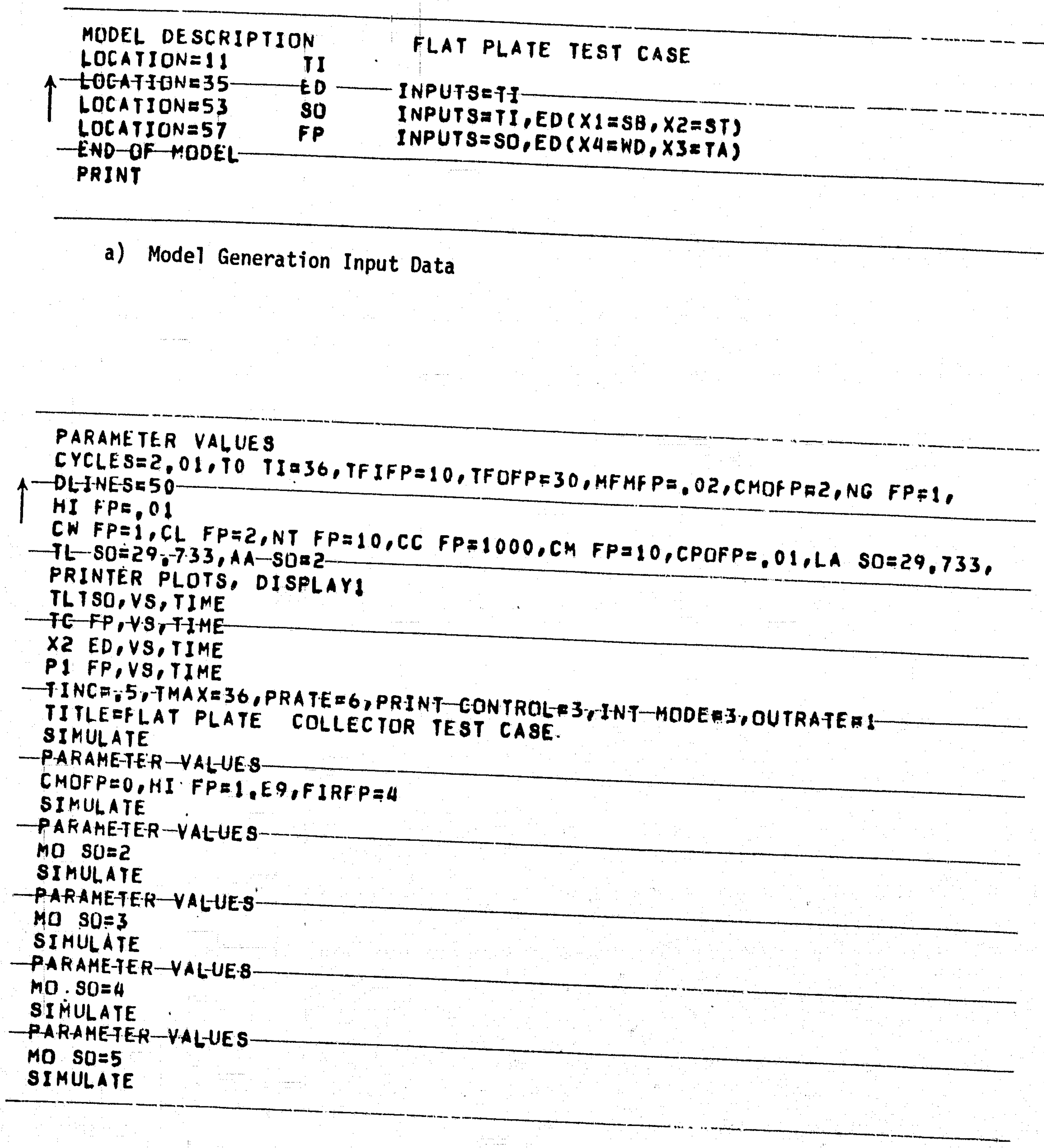


FIGURE 9.1-3 SOLAR ARRAY OUTPUT POWER VERSUS VOLTAGE



```
MODEL DESCRIPTION        FLAT PLATE TEST CASE
LOCATION=11      TI
LOCATION=35      ED      INPUTS=TI
LOCATION=53      SO      INPUTS=TI,ED(X1=SB,X2=ST)
LOCATION=57      FP      INPUTS=SO,ED(X4=WD,X3=TA)
END OF MODEL
PRINT
```

a) Model Generation Input Data

```
PARAMETER VALUES
CYCLES=2,01,TO TI=36,TFIFP=10,TFOFP=30,MFMFP=,02,CMOFP=2,NG FP=1,
DLINES=50
HI FP=,01
CW FP=1,CL FP=2,NT FP=10,CC FP=1000,CM FP=10,CPOFP=,01,LA SO=29,733,
TL SO=29,733,AA SO=2
PRINTER PLOTS, DISPLAY1
TLTSO,VS,TIME
TC FP,VS,TIME
X2 ED,VS,TIME
P1 FP,VS,TIME
TINC=,5,TMAX=36,PRATE=6,PRINT CONTROL=3,INT MODE=3,OUTRATE=1
TITLE=FLAT PLATE  COLLECTOR TEST CASE
SIMULATE
PARAMETER VALUES
CMOFP=0,HI FP=1,E9,FIRFP=4
SIMULATE
PARAMETER VALUES
MO SO=2
SIMULATE
PARAMETER VALUES
MO SO=3
SIMULATE
PARAMETER VALUES
MO SO=4
SIMULATE
PARAMETER VALUES
MO SO=5
SIMULATE
```

b) Simulation Program Input Data

FIGURE 9.2-1 FLAT PLATE COLLECTOR MODEL INPUT DATA

The model schematic produced by the model generation program is shown in Figure 9.2-2. The component **TI** is used to furnish time of day and day of year information to **SO** and to the TMY read component **ED. ED** supplies direct beam and global insolation to **SO**, and ambient temperature and wind speed to the collector component **FP**. Based on collector orientation, **SO** supplies solar insolation incident to the array, collector tilt angle, and tracking power to **FP**.

Typical results of the flat plate model runs are shown in Figures 9.2-3 through 9.2-5. Figure 9.2-3 shows the global horizontal insolation obtained from ED during the 36 hour simulation period. The data was for mid-winter and the daily peak levels are thus low to moderate. The array tilt angle daily pattern for horizontal E-W axis tracking is shown in Figure 9.2-4. At noon the array is oriented normal to the sun's incident rays and thus maximizes the insolation gathered during the mid-day peak. The tilt angle approaches 90° as the sun approaches the horizon, and remains fixed at 90° overnight. Comparison of the solar insolation peaks with the various tracking options showed that horizontal E-W axis tracking gave the best results of the single axis tracking systems, and was only slightly inferior to two-axis beam tracking. Solar cell temperature for this case is shown in Figure 9.2-5. The cell temperature is within a few degrees of ambient most of the day and rises in mid-day proportional to the solar insolation received. The results with water cooling are quite similar.

9.3 FRESNEL LENS COLLECTOR MODEL AND INCREMENTAL COSTS

The input data for the Fresnel Lens test case is shown in Figure 9.3-1. The purpose of this model is to illustrate a Fresnel Lens collector model with thermal fluid loops for collector cooling and for solar heating. Three week-long simulations are used to demonstrate incremental cost calculations for subsystem economic design. A variable speed pump is assumed for the collector fluid loop with the flow rate adjusted so that the outlet temperature is $5^{o}C$ greater than the inlet. The collector consists of a rectangular grid of 120

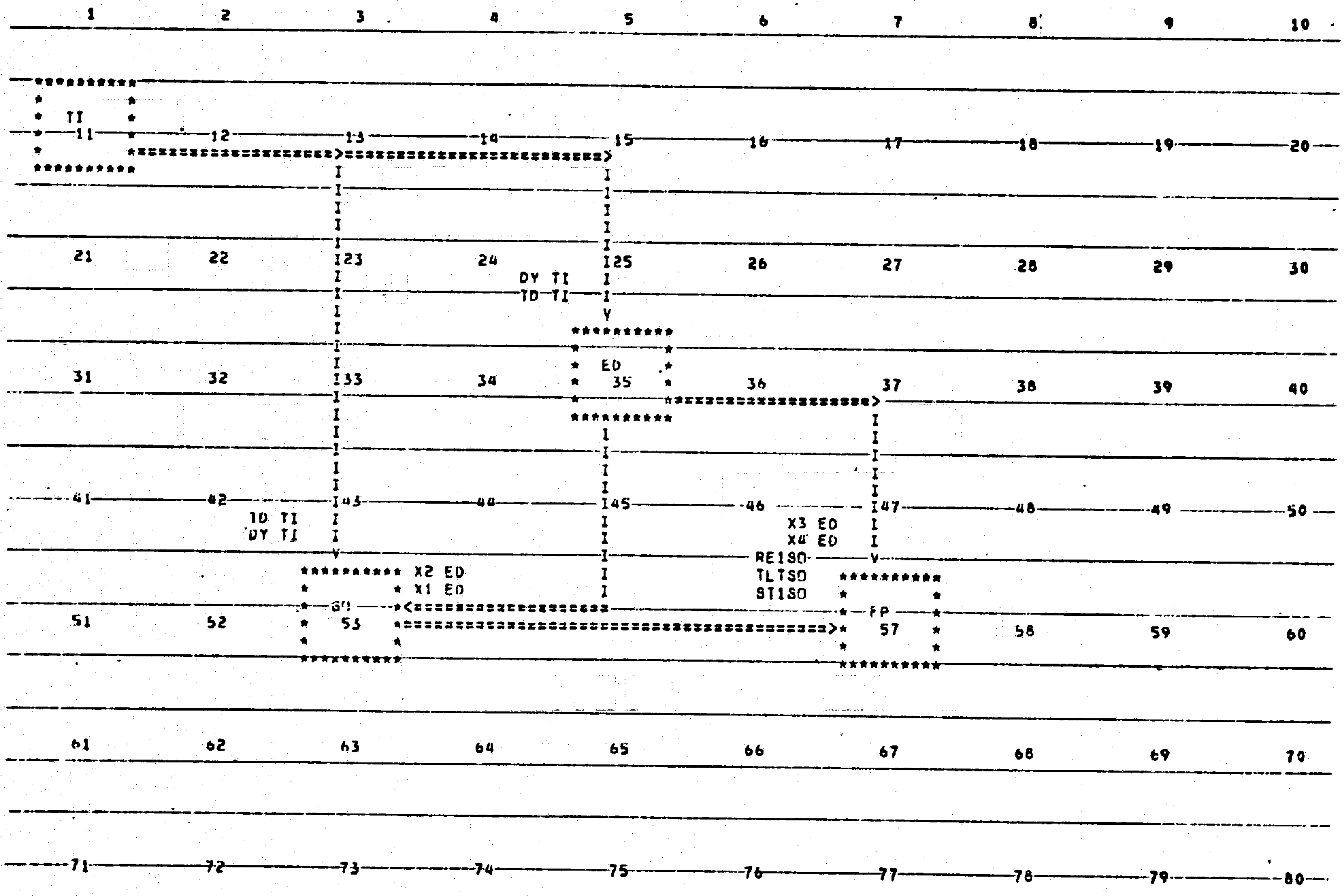


FIGURE 9.2-2 FLAT PLATE MODEL SCHEMATIC

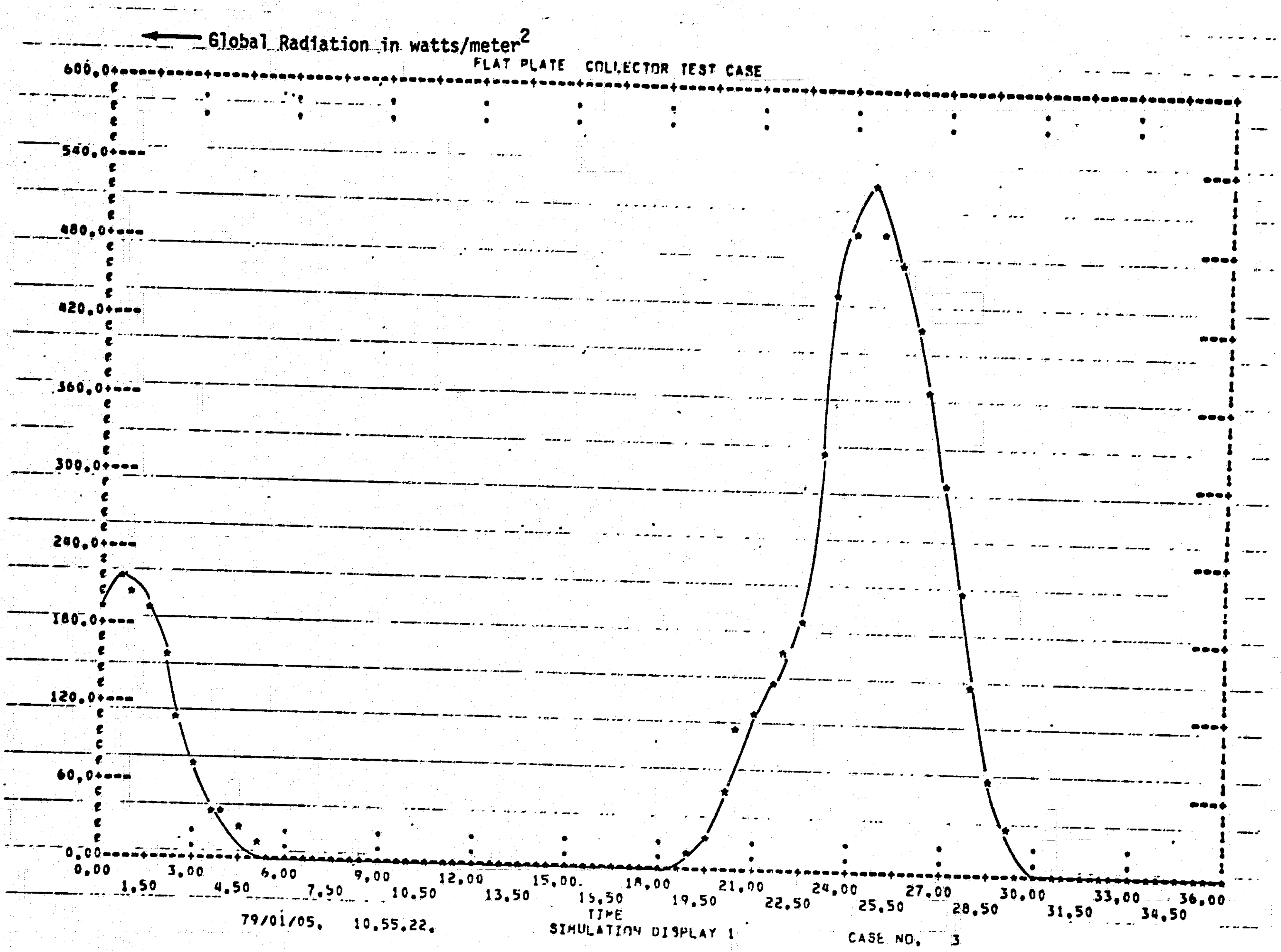


FIGURE 9.2-3 GLOBAL HORIZONTAL RADIATION VERSUS TIME

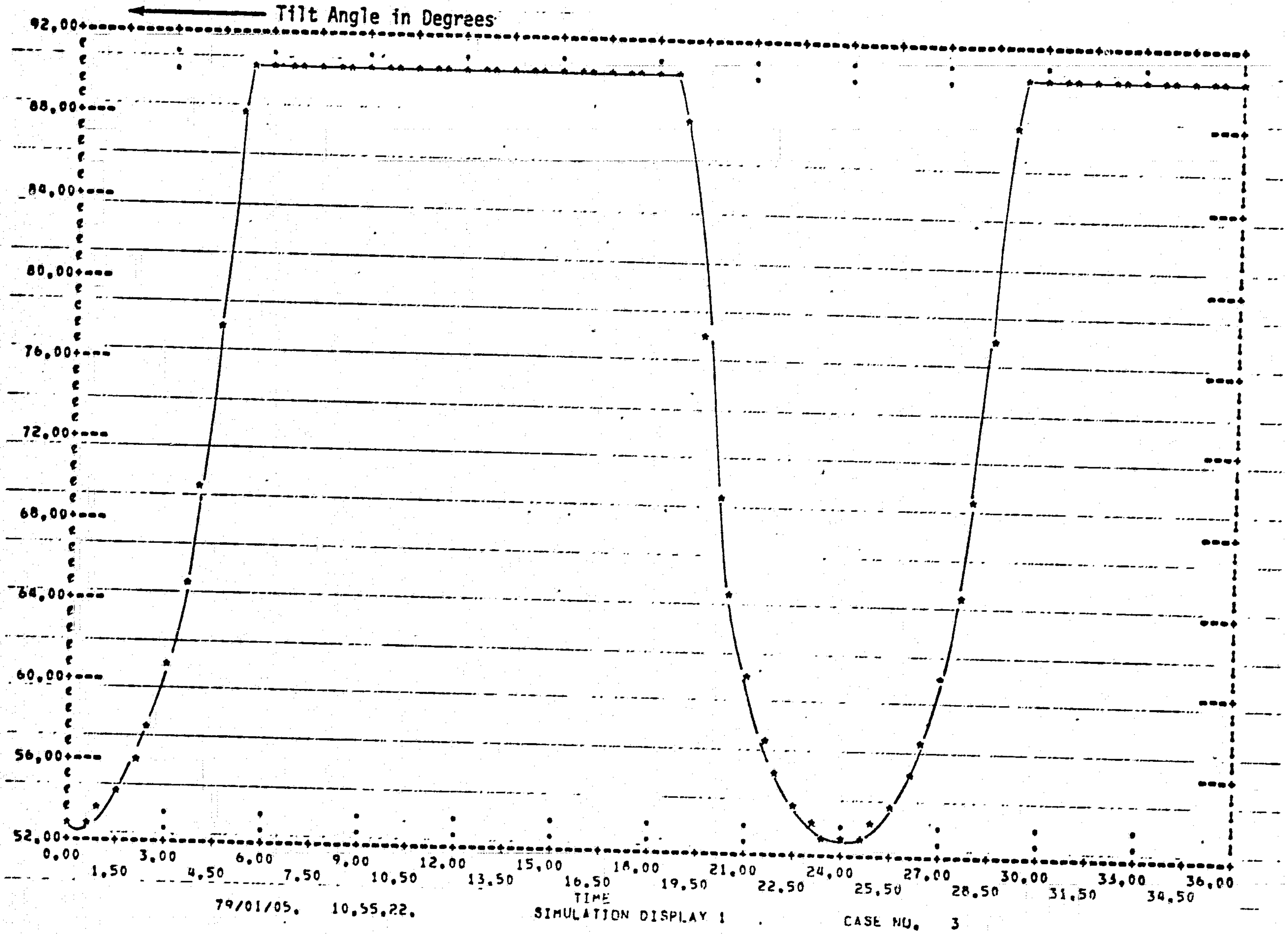


FIGURE 9.2-4 TILT ANGLE VERSUS TIME FOR HORIZONTAL E-W AXIS TRACKING

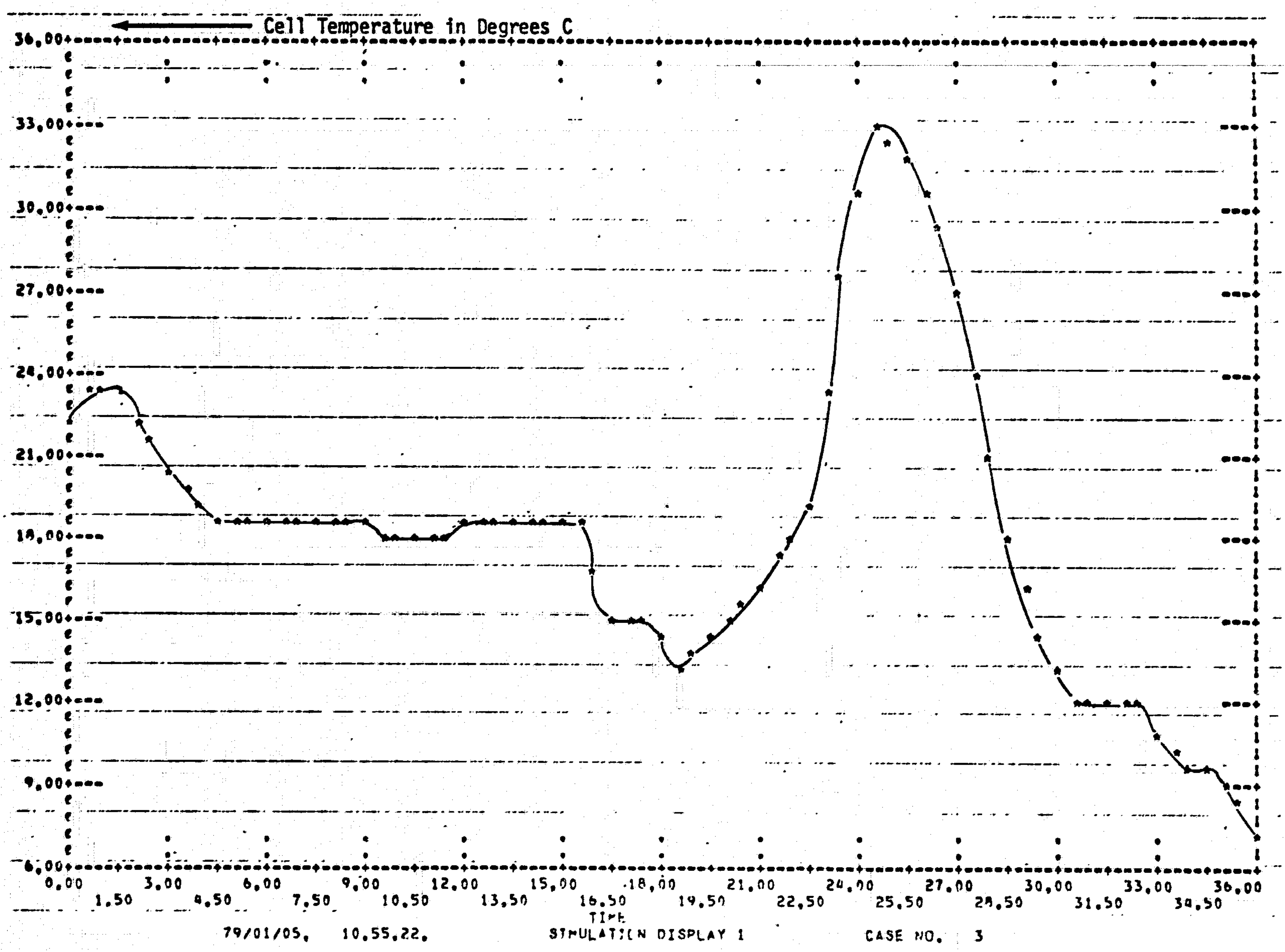


FIGURE 9.2-5 SOLAR CELL TEMPERATURE VERSUS TIME

```
MODEL DESCRIPTION     FRESNEL LENS COLLECTOR WITH THERMAL STORAGE AND LOAD
LOCATION=11     TI
LOCATION=71     ED     INPUTS=TI
LOCATION=45     MA     INPUTS=TS(T=FIN)
FORTRAN STATEMENTS
      TFOFO = FO MA+5.
LOCATION=33     FO     INPUTS=ED(X1=ST,X3=TA,X4=WD),MA(FO=TFI)
LOCATION=73     PV     INPUTS=ED(X1=ST),FO
LOCATION=47     TS     INPUTS=FO(P,1=P),TL
LOCATION=27     TL     INPUTS=TI,ED(X3=TA)
LOCATION=77     LO     INPUTS=PV(P=P,1,P=LO,1)
LOCATION=79     CM
END OF MODEL
PRINT
```

a) Model Generation Input Data

```
TITLE=FRESNEL LENS COLLECTOR  (INCREMENTAL COST COMPUTATION)
PARAMETER VALUES
CYCLES=4.01,TO TI=0,CMOFO=2,CW FO=3.75,CL FO=3.9, DLINES=50
NL FO=120,NT FO=24,MFMFO=0.5,CC FO=6.,CM FO=50,HI FO=.01,RC FO=.06
TS TS=5,DH TS=.00879,PD TS=12,LE TS=30,NU TS=.01,NC TL=0.2
C1 MA=.55556,C2 MA=-17.7778, COPFO=0.5
CC PV=100,CM PV=50,LE TS=30,CR CM=15,LE CM=20
AA PV=0.6,NS PV=600,NP PV=5,RAPPV=1.3
VE LO=.05,VE TL=.05
TABLE,HT TS=4
.00879,.025491,.047371,.064072
90,147,147,204
TABLE,TLOTL=4
-10,0,10,25
4,2,1.5,1
TABLE,TWTTL=4
0,6,18,24
.4,1,1,.4
PRINTER PLOTS,DISPLAY1
RE TL,VS,TIME
E  TS,VS,TIME
P1 FO,VS,TIME
FMDFO,VS,TIME
DISPLAY2
TC FO,VS,TIME
P. PV,VS,TIME
FO MA,VS,TIME
INITIAL CONDITIONS=E  TS=80
TINC=.5,TMAX=168,PRATE=12,PRINT CONTROL=3,INT MODE=3,OUTRATE=1
SIMULATE
PARAMETER VALUES, TS TS=5.5
SIMULATE
PARAMETER VALUES
TS TS=5.,NL FO=126,CW FO=3.94,AA PV=0.63,NS PV=630
SIMULATE
```

b) Simulation Program Input Data

FIGURE 9.3-1 FRESNEL LENS MODEL INPUT DATA

Fresnel lenses each of which focuses solar radiation on a 5 x 5 array of solar cells. Excess thermal energy is conducted to a heat sink surface and then dissipated by natural convection, radiation and heat exchange to the coolant fluid. The collector parameters are chosen for a lens concentration ratio of 25 and series connection of the output from each array. At maximum output the array collects about 10kw of solar radiation and produces about 1.7kw of electrical power. The user should be especially careful in specifying the input parameters to the collector and array components **FO** and **PV**, since inadvertant parameter errors can lead to physically inconsistent configurations, e.g., collector area smaller than the total lens area.

The model schematic produced by the model generation program is shown in Figure 9.3-2. The collector thermal loop is formed by the connections between the collector **FO** the thermal storage **TS** and the multiply and add component MA. The MA component is used to convert the thermal storage outlet temperature from degrees fahrenheit to degrees centigrade. The output temperature from MA is supplied as the inlet temperature to **FO.** The total thermal power gathered by the coolant fluid is computed in **FO** and supplied to **TS.** Similarly, the thermal load fluid loop is represented by a power request from the load component **TL** to **TS** and by thermal power delivered from **TS** to **TL.** The electrical output of the array is computed by **PV** and supplied to a load component **LO** which monitors the electrical energy collected.

Results of the first week simulation run are summarized in Figures 9.3-3 through 9.3-6. The weather was fairly constant during this run and solar insolation was fairly strong all week. Figure 9.3-3 shows that with water cooling cell temperature was held to less than 70^{o}C at peak insolation. In fact, about 60% of the solar energy incident on the array is exchanged to the coolant fluid during peak insolation. The electrical output of the array is shown in Figure 9.3-4. The fluid flow rate of the pump and thermal energy collected exhibit very similar daily patterns. The thermal load for this week is shown in Figure 9.3-5. This load is dependent on both time of day and

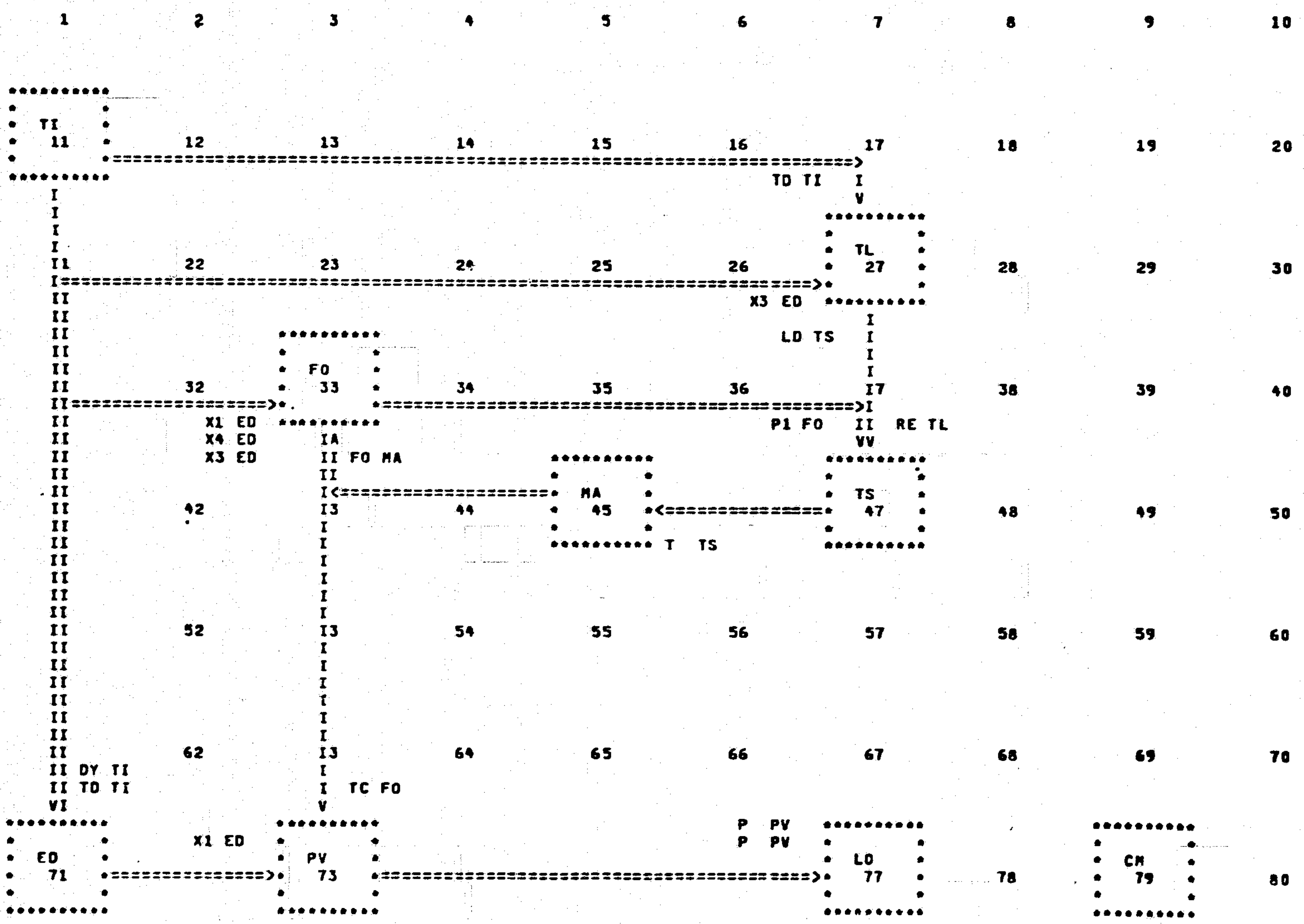


FIGURE 9.3-2 FRESNEL LENS MODEL SCHEMATIC

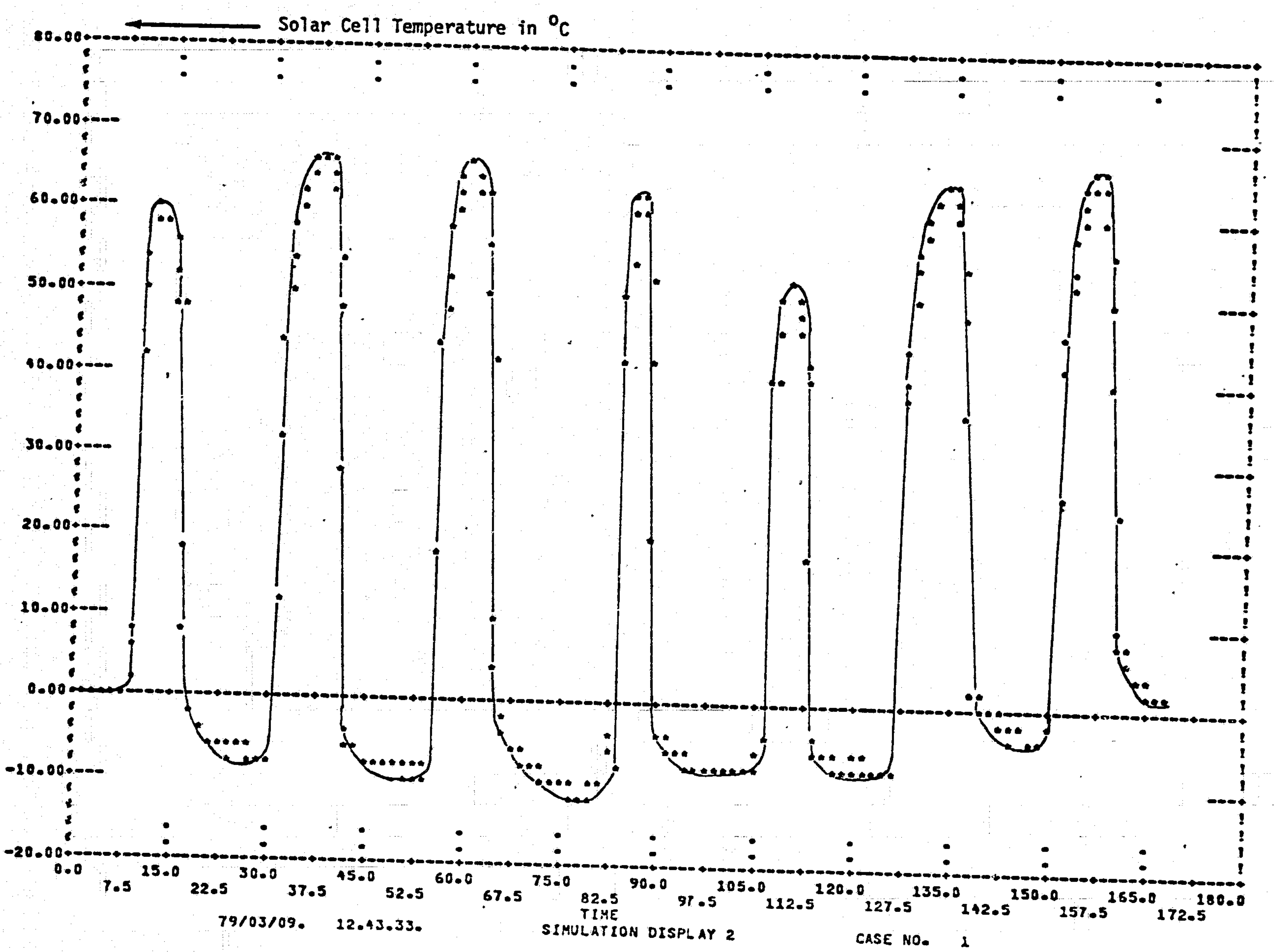


FIGURE 9.3-3 SOLAR CELL TEMPERATURE FOR ONE WEEK SIMULATION

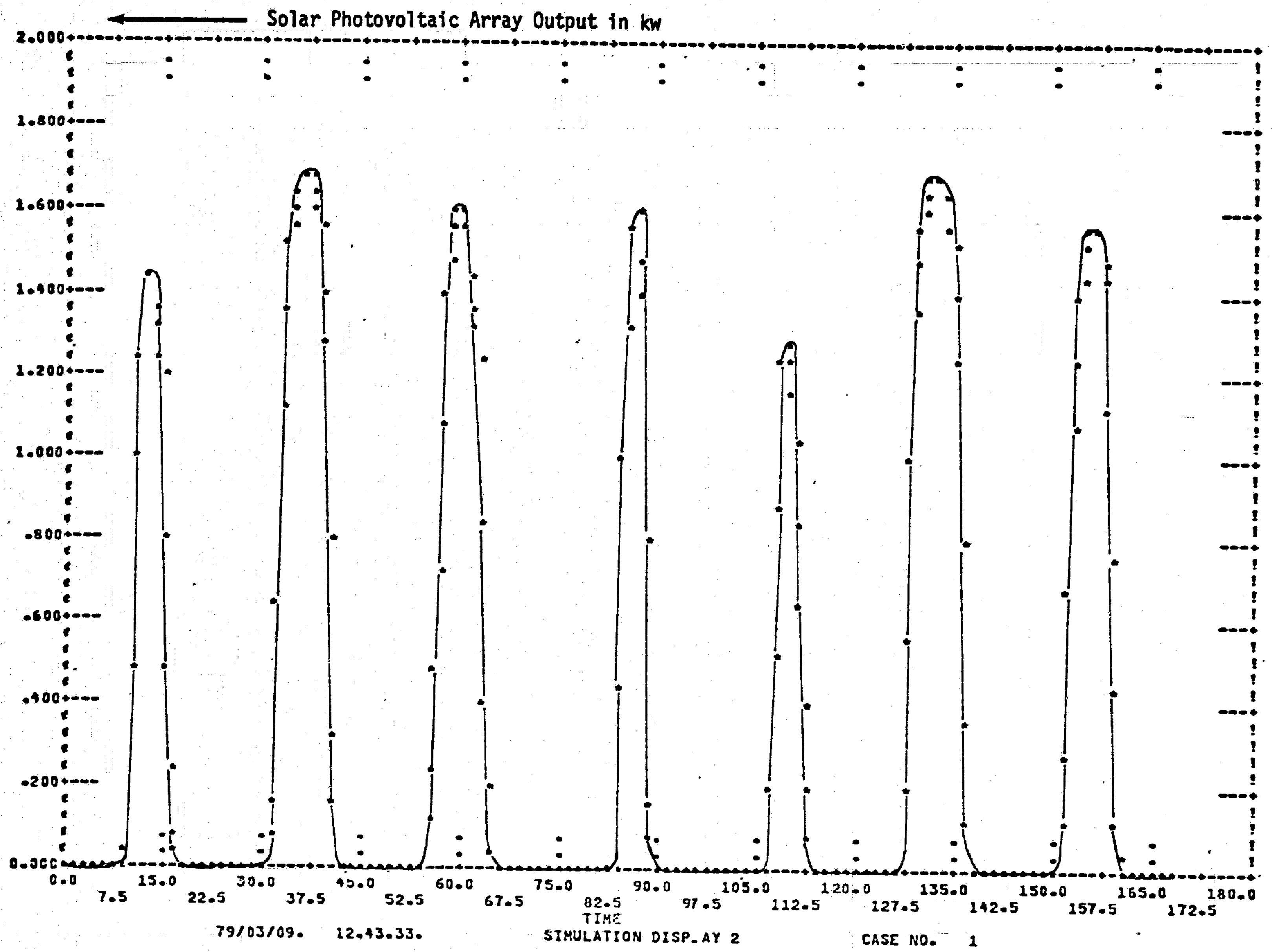


FIGURE 9.3-4 PHOTOVOLTAIC ARRAY OUTPUT FOR ONE WEEK SIMULATION

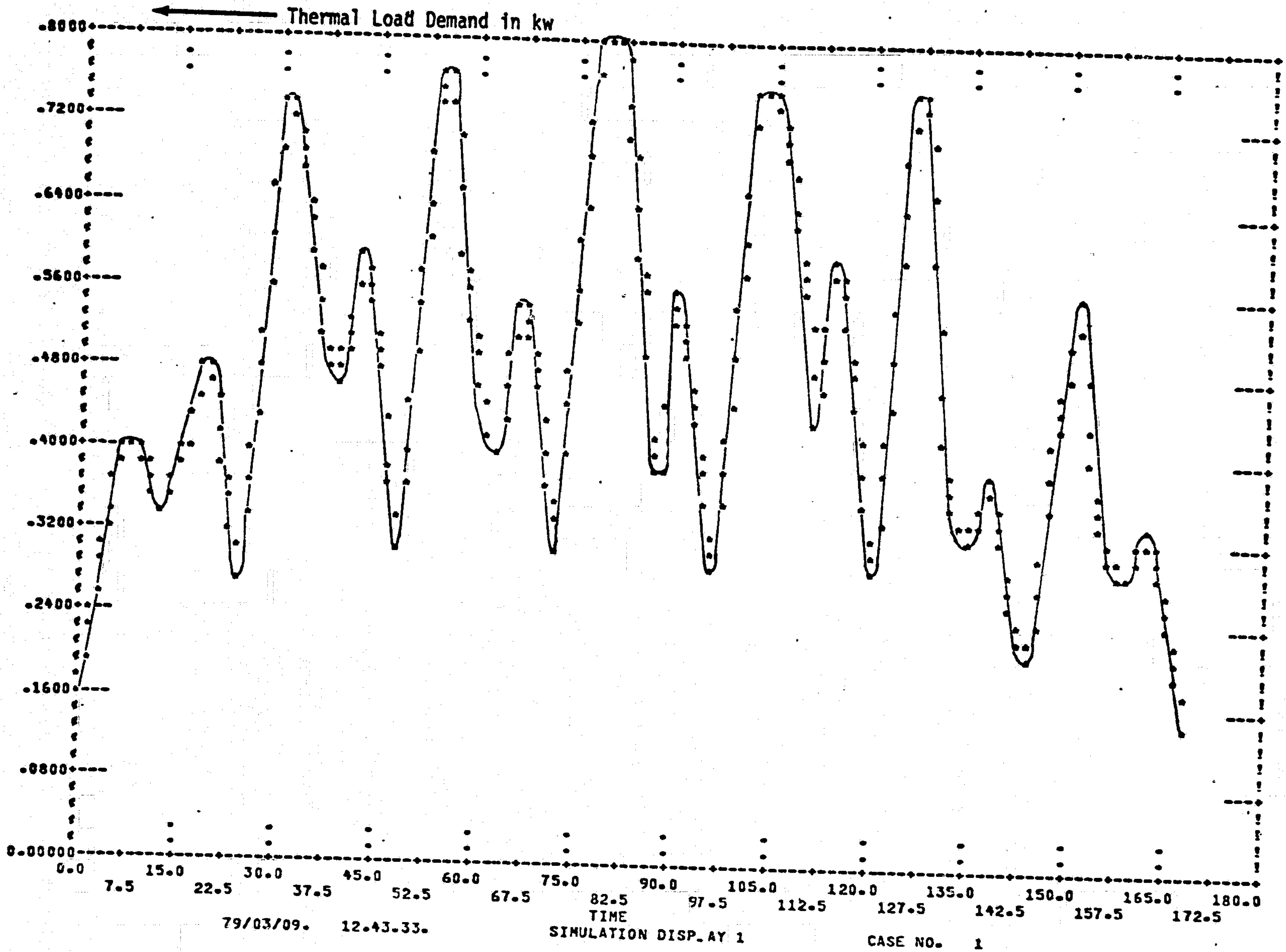


FIGURE 9.3-5 THERMAL LOAD DEMAND FOR ONE WEEK SIMULATION

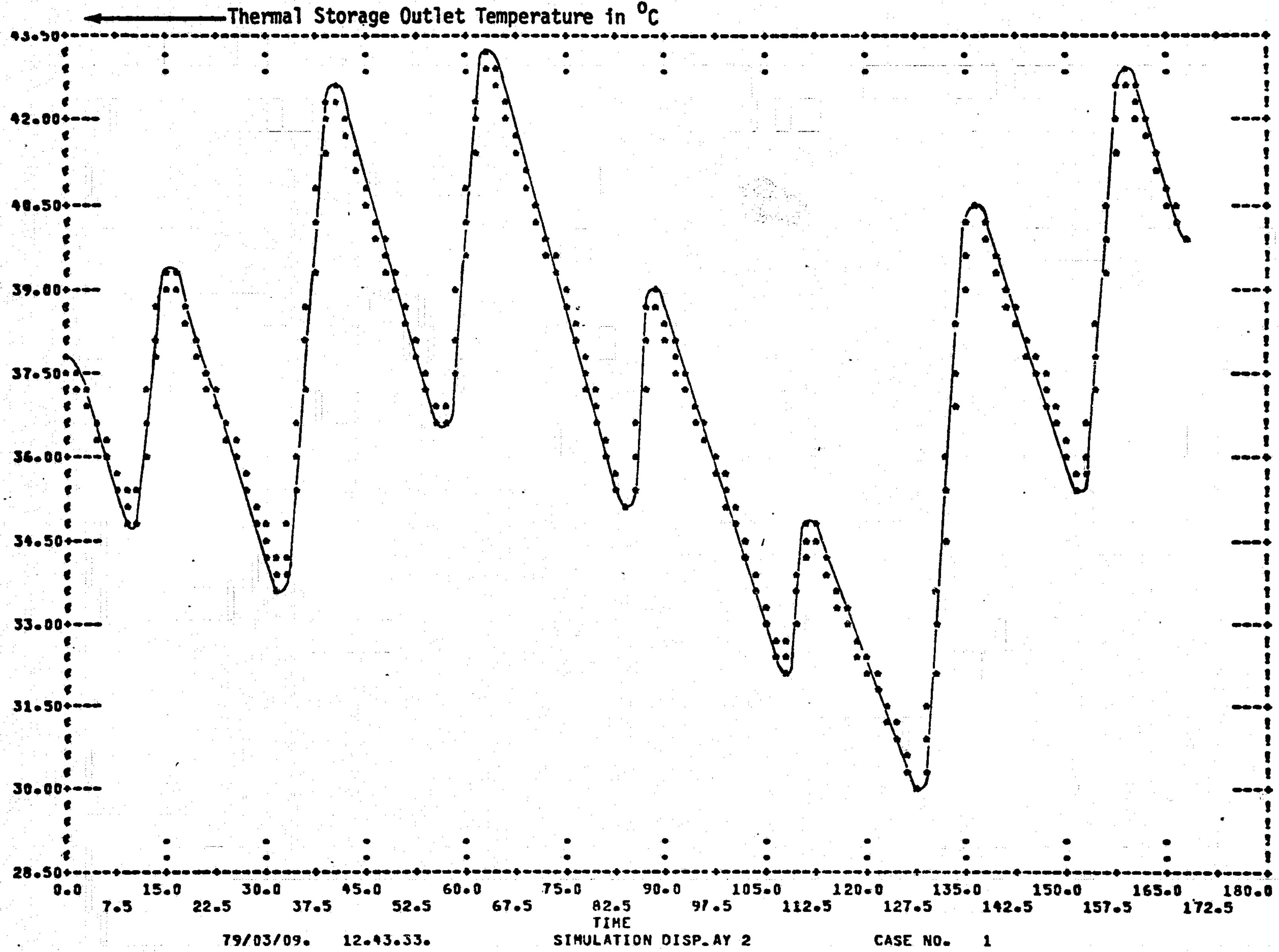


FIGURE 9.3-6 THERMAL STORAGE TEMPERATURE FOR ONE WEEK SIMULATION

ambient temperature which yields the complex load pattern shown. Figure 9.3-6 shows the temperature of the thermal storage vessel resulting from the collector and load thermal loops. The daily cycles are predominant with the periods of strong insolation providing sufficient energy to satisfy the load and compensate for thermal losses. Average load is fairly well matched to solar generation during the week since the temperature remains within a 15° channel and does not have an apparent trend away from this range.

One of the most important measures of performance for a solar energy system is the levelized cost of energy, i.e., the life cycle cost to produce one unit of usable energy including generation, storage, transmission and conversion subsystems. Energy cost may be used to size components and select most promising system alternatives, i.e., minimum energy cost is used as a selection or optimization principle. Although SIMWEST does not provide user optimization capability, optimal sizing of a few key parameters, such as the ratio of solar to utility generation and the size of storage relative to generation, is possible and may be accomplished quickly using the concept of incremental energy cost. The idea is to compute the incremental change in levelized energy cost per incremental change in capital cost, for the system parameters of interest. Given an initial system configuration and M sizing parameters to be selected, optimization proceeds as follows:

1) Perform M+1 back to back simulations to compute the cost and energy performance of the baseline configuration and M incremental configurations from the baseline.

2) Calculate the incremental energy costs for each parameter variation. Then select a new baseline configuration. Since the incremental costs are equal at the minimum cost point, increase or decrease the sizing parameters so as to equalize the new baseline incremental costs.

3) Go to 1) and continue adjusting subsystem parameters until either a performance limit is reached or until the incremental costs of the

remaining parameters are equalized. (If two incremental costs are unequal, one can always lower the system energy cost by increasing the subsystem with the smallest incremental cost at the expense of the other subsystem.)

This procedure is recommended as more efficient and economical than using a series of parametric trade studies for subsystem optimization.

The process of computing incremental costs is illustrated for the Fresnel Lens model. In the first simulation the baseline system performance and costs are computed. The second simulation differs from the first in that thermal storage capacity has been increased by 10%, and the third simulation differs from the first in that the solar collector and photovoltaic array area have been increased by 5%. Table 9.3-1 summarizes the incremental cost and simulation results for these runs. Column 1 shows the initial capital cost of the baseline system and the incremental capital costs for the thermal storage and solar array increases. (These costs are meant to be illustrative rather than representative.) Column 2 shows the results of a 20 year levelized cost analysis of the three systems, including maintenance and operating costs, e.g., the change in thermal storage increases costs by $9.10 per year. Column 3 shows the energy delivered to the loads in a year as estimated from the one week simulations. (Note: the change in storage capacity lowers the average coolant temperature, thus increasing output power.) Column 4 shows the levelized energy costs of the baseline system and of the increments in storage and generation. This column shows that the levelized energy cost will decrease as thermal storage or generation are increased, and that thermal storage is undersized relative to generation since a fixed $ increase in storage will lower the system energy cost more than the same $ increase in array area. Column 5 shows the % change in levelized energy cost given a 1% increase in capital investment. This column contains the same basic information as column 4 but provides a better quantitative measure of the economic value of increased storage capacity.

TABLE 9.3-1 INCREMENTAL COST CALCULATIONS

	CC	LC	ED	EC	NIC
Baseline	7392.	1272.	7829.	16.2	----
10% Inc. in Thermal	61.	9.10	110.5	8.2	-.84
5% Inc. in Solar	319.	47.90	365.0	13.1	-.21

NOMENCLATURE:

CC = Initial Capital Cost in $

LC = Levelized Total Cost/Yr. in $
= Capital Cost*Life Cycle*Charge Rate + Maintenance Cost + Operating Cost

ED = Useful Energy Delivered/Yr. in KWH
= Electrical Load + Thermal Load + Net Change in Thermal Storage

EC = Levelized Energy Cost in ¢/KWH
= LC*100/ED

NIC = Normalized Incremental Costs
= % Change in EC Per % Change in CC
$\cong (\Delta LC/LC - \Delta ED/ED)/(\Delta CC/CC)$

APPENDIX: UTILITY SUBROUTINES

This section provides a short description and source code for the utility subroutines called by the SIMWEST library components. These routines are also available to the user and may be called by FORTRAN statements in the user's manual. (See also page 26 of section 2.1.2 on the use of subroutines TBLU1 and TBLU2.)

- FUNCTION AINR

AINR computes the current of a photovoltaic cell given light current AIL, cell voltage V, and temperature T. Newton-Raphson iterations are used to solve the implicit equation (1) for current I:

$$I = AIL + BIO\ (1. - EXP((V+I*RS)*QBK/(T+273))) \qquad (1)$$

- SUBROUTINE CNVC

CNVC computes the convection coefficient HC and Reynolds number RE for air blown over a flat plate (ref. 1).

Inputs: T_A = air temperature in ^oK

T_P = plate temperature in ^oK

CL = length of plate in m

V = velocity of air in m/s

Equations:

$T_M = (T_A + T_P)/2$ (mean temp.)

$VI = 9.0 \times 10^{-8}*T_M - 1.115 \times 10^{-5}$ (viscosity)

$GR = 1.386 \times 10^{3} - 2.91*T_M$ (Grashof's no.)

$CO = 7.25 \times 10^{-5}*T_M + 4.325 \times 10^{-3}$ (conductivity)

$RE = V*CL/VI$ (2)

$$H_{FREE} = .116*CO*GR*|T_A - T_P|^{.333}$$

$$H_{WIND} = \begin{cases} .597*CO*REE^{.5}/CL & RE \leq 5 \times 10^5 \\ .032*CO*(RE^{.8} - 23000)/CL & \text{otherwise} \end{cases}$$

$$HC = H_{FREE} + H_{WIND} \qquad (3)$$

- SUBROUTINE CUBIC

CUBIC finds the roots of the cubic equation

$$x^3 + AAx + BB = 0 \qquad (4)$$

and selects the real root $\bar{x}$ with largest value.

- SUBROUTINE FLUC

FLUC computes the heat transfer coefficient HF from a collector plate into a fluid coolant. The empirical equations used are for water cooling (ref. 1).

Inputs: NT = number of cooling tubes

DT = diameter of cooling tubes in m

CW = collector width in m

COP = conductivity of mounting plate in w/m-K

THP = mounting plate thickness in m

FMD = coolant mass flow rate in kg/s

DEN = coolant density in kg/m^3

TF = mean coolant temperature in K

COC = coolant conductivity in w/m-K

Equations:

$$NT1 = NT/CW$$

$$HF1 = 12*NT1^{2}*COP*THP \qquad \text{(conduction coeff.)}$$

$$VI = (21.7*(TF - 256)^{-0.8} - .185) \times 10^{-6} \qquad \text{(fluid viscosity)}$$

$$PR = (.00518*TF - 1.25)**(-1.49) \qquad \text{(Prandtl no.)}$$

$$RE = 4.*FMD/(\pi*DT*NT*DEN*VI) \qquad \text{(Reynolds no.)} \qquad (5)$$

If $RE < 2100$,

$$HF2 = 4.36*COC*\pi*NT1$$

If $RE > 10000$

$$HF2 = .023*COC*RE^{.8}*PR^{.333}*\pi*NT1$$

If $2100 \leq RE < 10000$

$$X2 = 36.5*PR^{.33}$$

$$D2 = .0029*PR^{.33}$$

$$A = (4.36-X2)*1.6 \times 10^{-8} + D2*1.266 \times 10^{-4}$$

$$B = D2 - A*2.x10^{4}$$

$$C = X2 + A*10^{8} - D2*10^{4}$$

$$HF2 = (A*RE^{2}+B*RE+C)*COC*\pi*NT1$$

$$HF = (1/HF1 + 1/HF2)^{-1} \qquad (6)$$

- FUNCTION HTGLAS

HTGLAS computes the top surface heat loss coefficient H_t for a collector with 1 to 3 glass covers (ref. 2).

Inputs: N = number of glass covers (1,2,3)

T_A = ambient temperature in ^{o}K

T_C = mean cell temperature in ^{o}K

H_C = convection coefficient for air blowing over a heated flat plate in w/m^2-k

e_c, e_g = emittance of cell and glass covers

TLT = collector tilt from horizontal in degrees

Equations:

$$H_t = (N(T_C/C)/((T_C-T_A)/(N+f))^{0.33} + 1/H_C)^{-1}$$

$$+ \sigma(T_C^2+T_A^2)(T_C+T_A)/(A+(2N+f-1)/e_g-N) \qquad (7)$$

with $\sigma = 5.688 \times 10^{-8}\ w/m^2\text{-}K^4$

$C = 365.9\ (1.-.00883*TLT+.0001298*TLT^2)$

$f = (1.-.04*H_C+.0005*H_C^2)(1.+.091*N)$

$A = 1/(e_c+.05*N(1-e_c))$

- SUBROUTINE IMPLIC

IMPLIC controls the iteration logic which determines convergence of implicit variables in the user's system model, and prints convergence diagnostics. (See section 3.6 for a discussion of the iteration and diagnostic control logic.)

- SUBROUTINE RADC

RADC computes the infrared radiation coefficient HR between two bodies with surface temperatures T_1 and T_2. (See section 7.4 of Duffie and Beckman, ref. 3.)

Inputs: T_1,T_2 = surface temperatures in ^oK

e_1,e_2 = emittances for surfaces corresponding to T_1,T_2

$$H_R = 5.688 \times 10^{-8}\ (T_1^2+T_2^2)(T_1+T_2)/(e_1^{-1} + e_2^{-1} -1) \qquad (8)$$

- FUNCTIONS TBLU1, TBLU2

TBLU1 and TBLU2 perform one- and two-dimension linear interpolation. A binary search is used to locate the nearest grid points for unequally spaced data. See section 2.1.2 for subroutine usage within model generation FORTRAN statements.

- SUBROUTINE UNIF

UNIF generates uniformly distributed, pseudo-random number sequences in the range [0,1]. This routine may be used to obtain random number sequences with a specified distribution function. (See for example the coding for WD in section 7.47.)

REFERENCES

1. F. Kreith, Principles of Heat Transfer, 3rd Edition, International Textbook Co., 1973.
2. S. A. Klein, M. S. Thesis, "The effects of Thermal Capacitance Upon the Performance of Flat Plate Solar Collectors", University of Wisconsin, 1973.
3. J. A. Duffie and W. A. Beckman, Solar Energy Thermal Processes, Wiley, 1974.

```
CAINR
      FUNCTION AINR(AIL,BIO,QBK,V,RS,T)
C
C       NEWTON-RALPHSON TO COMPUTE PHOTO-VOLTAIC CELL CURRENT
C
      F(A)=A-AIL-B1O*(1.-EXP(QBK*(V+A*RS)/(T+273)))
      FP(A)=1.+BIO*EXP(QBK*(V+A*RS)/(T+273))*QBK*RS/(T+273)
      A=0.
      DO 1 J=1,10
      ANEW=A-F(A)/FP(A)
      IF((ANEW-A).LE..00001)GO TO 2
    1 A=ANEW
    2 AINR=ANEW
       RETURN
      END
```

```
CNVC
      SUBROUTINE CNVC(HC,RE,TP,TA,V,CL)
C
C           COMPUTES CONVECTION COEFFICIENT HC AND REYNOLDS
C           NUMBER RE FOR AIR BLOWING OVER A FLAT PLATE.
C           CALLED BY COMPONENT FO.
C           INPUTS    TA  -AIR TEMPERATURE,K
C                     TP  -PLATE TEMPERATURE,K
C                     V   -VELOCITY OF AIR,M/S
C                     CL  -LENGTH OF PLATE,M
C
      TM=(TA+TP)*.5
      VI=9.E-8*TM-1.115E-5
      GR=1386.-2.91*TM
      CO=7.25E-5*TM+4.325E-3
      RE=V*CL/VI
      HFREE=.116*CO*GR*((ABS(TA-TP))**(.333))
      HWIND=.597*CO*SQRT(RE)/CL
      IF(RE.GT.5.E5)HWIND=.032*CO*(RE**(.8)-23000.)/CL
      HC=HFREE+HWIND
      RETURN
      END
```

CUBIC

```
      SUBROUTINE CUBIC(AA,BB,ANS)
      TER=AA**3/27.
      TERM=BB**2/4.+TER
      IF(ABS(TERM).GT..0001)GO TO 10
C     ******************************************************************
C     THREE REAL ROOTS, TWO EQUAL
C     ******************************************************************
      AB=2.*CBRT(-BB/2.)
      ABB=-AB/2.
C     ******************************************************************
C     SELECT POSITIVE ROOT
C     ******************************************************************
      ANS=AMAX1(AB,ABB)
      RETURN
   10 IF(TERM.LT.0.)GO TO 20
C     ******************************************************************
C     ONE REAL ROOT, TWO CONJUGATE IMAGINARY ROOTS
C     ******************************************************************
      STERM=SQRT(TERM)
      AAA=CBRT(-BB/2.+STERM)
      BBB=CBRT(-BB/2.-STERM)
C     ******************************************************************
C     SELECT REAL ROOT
C     ******************************************************************
      ANS=AAA+BBB
      RETURN
C     ******************************************************************
C     THREE REAL, UNEQUAL ROOTS
C     ******************************************************************
   20 STER=SQRT(-TER)
      THETA=ACOS(-BB/2./STER)
      TE=2.*SQRT(-AA/3.)
      THETA3=THETA/3.
      X1=TE*COS(THETA3)
      X2=TE*COS(THETA3+2.09439)
      X3=TE*COS(THETA3+4.18879)
C     ******************************************************************
C     SELECT SMALLEST POSITIVE ROOT
C     ******************************************************************
      ANS=AMAX1(X1,X2,X3)
      RETURN
      END
```

```
CFLUC
      SUBROUTINE FLUC(HF,RE,NT,DT,CW,COS,THS,FMD,DEN,TF,COC)
C
C           COMPUTES HEAT TRANSFER COEFFICIENT HF TO FLUID
C           AND REYNOLDS NUMBER.
C           CALLED BY COMPONENT FO
C           INPUTS    NT  -NUMBER OF COOLING TUBES
C                     DT  -DIAMETER OF COOLING TUBES
C                     CW  -COLLECTOR WIDTH,M
C                     COS -CONDUCTIVITY OF MOUNTING PLATE,W/M-K
C                     THS -MOUNTING PLATE THICKNESS,M
C                     FMD -COOLANT MASS FLOW RATE,KG/S
C                     DEN -COOLANT DENSITY,KG/M3
C                     TF  -MEAN COOLANT TEMPERATURE,K
C                     COC -COOLANT CONDUCTIVITY,W/M-K
C
      REAL NT,NT1
C     WRITE(6,108 )FMD,DEN,TF,COC
C 108 FORMAT(1H0,5X,*FLUC INPUTS *,4F10.2)
      PR=(.00518*TF-1.25)**(-1.49)
      NT1=NT/CW
      HF1=12.*NT1*NT1*COS*THS
      VI=(21.7*(TF-256.)**(-.8)-.185)*1.E-6
      RE=4.*FMD/(3.1416*DT*NT*DEN*VI)
      IF(RE.GT.2100.)GO TO 1
      HF2=4.36*COC*3.1416*NT1
      GO TO 5
    1 IF(RE.GT.10000.)GO TO 2
      X2=36.5*(PR**(.33))
      D2=.0029*(PR**(.33))
      A=(4.36-X2)*1.6E-8+D2*1.266E-4
      B=D2-A*2.E4
      C=X2+A*1.E8-D2*1.E4
      HF2=(A*RE*RE+B*RE+C)*COC*3.1416*NT1
      GO TO 5
    2 CONTINUE
      HF2=.023*COC*(RE**(.8))*(PR**(.333))*3.1416*NT1
    5 CONTINUE
      HF=1./(1./HF1+1./HF2)
C     WRITE(6,109)HF,RE
C 109 FORMAT(1H0,5X,*FLUC OUTPUTS *,2F10.2)
      RETURN
      END
```

```
CHTGLAS
      FUNCTION HTGLAS(NG,TA,TC,HC1,EC,EG,TLT)
C
C        TOP HEAT LOSS COEFFICIENT HT FOR GLAS COVERS, CALLED BY FP
C
C
C        INPUTS
C            NG=NUMBER OF GLASS COVERS (1,2,3)
C            TA=AMBIENT TEMPERATURE,K
C            TC=MEAN CELL TEMPERATURE,K
C            HC1=CONVECTION COEFFICIENT FOR AIR BLOWING OVER
C                A HEATED FLAT PLATE, W/M2-K
C            EC,EG=EMITTANCE OF CELL AND GLASS COVERS
C            TLT=COLLECTOR TILT FROM HORIZONTAL IN DEGREES
C
      REAL NG
      SIGMA=5.688E-8
      C=365.9*(1.-.00883*TLT+.0001298*TLT*TLT)
      F=(1.-.04*HC1+.0005*HC1*HC1)*(1.+.091*NG)
      A=1./(EC+.05*NG*(1.-EC))
      G=NG*(TC/C)/(((TC-TA)/(NG+F))**0.33) + 1./HC1
      B=SIGMA*(TC*TC+TA*TA)*(TC+TA)/(A+(2.*NG+F-1.)/EG-NG)
      HTGLAS=1./G+B
      RETURN
      END
```

```
CIMPLIC
      SUBROUTINE IMPLIC(CYCLES,DLINES)
      COMMON/CIMPL/IMPL,ICNT /CORDER/ NOX,NOV /COLD/VOLD
      COMMON /CV/ V /CNAMEV/ NAMEV /CTIME/ TIME
      DIMENSION V(1),NAMEV(1),VOLD(1)
C ******      UNIVAC VERSION CODE ONLY
C     IF(CYCLES.LE.0.) GO TO 40
C ******
      IF(IMPL.GT.0)GO TO 10
      SP=0
      ITERS=CYCLES
      ITERS= MAXO(1,MINO(ITERS,20))
      ILINES= ABS(DLINES)
      ITNO= 0
      IMPL=1
      DO 5 I=1,NOV
    5 VOLD(I) = 0.
   10 CONTINUE
C ******         CDC VERSION CODE ONLY
      IF(CYCLES.GE.1.) GO TO 15
      IMPL=2
      IF(ICNT.GE.ILINES) IMPL=3
      RETURN
C ******
   15 IF(IMPL.GT.1) GO TO 20
      ITNO= ITNO+1
      IF(ITNO.GE.ITERS) IMPL=2
      ICON=1
      DO 30 I=1,NOV
      IF(ABS(V(I)).LT. 1.E-6) GO TO 30
      IF( ABS(VOLD(I)-V(I)) .GT. 0.03*ABS(V(I)) )ICON=0
      VOLD(I)= V(I)
   30 CONTINUE
      IF(ICON.EQ.1) IMPL=2
      IF(IMPL.EQ.2 .AND. ICNT.GE.ILINES) IMPL=3
      RETURN
C
C
   20 ITNO=0
      IF(IMPL.GT.2) GO TO 40
      IF(ICON.EQ.1) GO TO 40
      IF(DLINES.LT.0.) GO TO 40
      ICK=0
      DO 50 I=1,NOV
      IF( ABS(V(I)).LT.1.0E-6) GO TO 50
      IF( ABS(VOLD(I)-V(I)) .LT. 0.05*ABS(V(I)) )GO TO 50
      IF(ICK.EQ.0) WRITE (6,100) TIME
  100 FORMAT(1H0,10X,5HTIME=,F9.2)
      WRITE(6,200) NAMEV(I),VOLD(I),V(I)
  200 FORMAT(1H ,10X,A6,28H  NONCONVERGENCE. OLD VALUE=,F12.3,
     1  13H   NEW VALUE=,F12.3)
      ICK=1
   50 CONTINUE
      IF(ICK.EQ.1) ICNT=ICNT+1
   40 IMPL=4
      RETURN
      END
```

```
CRADC
      SUBROUTINE RADC(HR,T1,T2,E1,E2)
C
C         COMPUTES INFRARED RADIATION COEFFICIENT HR
C         CALLED BY COMPONENT FO
C          INPUTS   T1,T2  -SURFACE TEMPERATURES,K
C                  E1,E2  -CORRESPONDING SURFACE EMITTANCES
C
      HR=5.688E-8*(T1*T1+T2*T2)*(T1+T2)/(1./E1+1./E2-1.)
      RETURN
      END
```

```
CTBLU1
      FUNCTION TBLU1(X,XT,FT,NDX,NX)
C
C     PURPOSE      ONE DIMENSION LINEAR INTERPOLATION
C
C     CALL SEQUENCE
C
C
C               X   - VALUE OF INDEPENDENT VARIABLE
C               XT  - ARRAY OF LENGTH ABS(NX) CONTAINING X VALUES
C               FT  - ARRAY OF TABLE VALUES CORRESPONDING TO XT
C               NDX- INDICATOR FOR STEP SPACING
C                       IF NDX.EQ.0 THEN XT CONTAINS EQUAL SPACED DATA
C                       IF NDX.NE.0 THEN XT CONTAINS UNEQUAL SPACED DATA
C               NX  - ABS(NX) IS THE ARRAY LENGTH
C                       IF NX.LT.0 THEN TRUNCATE OUTSIDE TABLE RANGE
C                       IF NX.GE.0 THEN EXTRAPOLATE OUTSIDE TABLE RANGE
C     WRITTEN BY A.W.WARREN
C                                                       VERSION 1, APRIL 1977
      DIMENSION XT(1),FT(1)
      NA=IABS(NX)
      IF(NA.GT.1)GO TO 5
      TBLU1=FT(1)
      RETURN
    5 IF(NDX.NE.0) GO TO 100
C
C
C                                  EQUI-SPACED TABLE INTERPOLATION
      X0= XT(1)
      H= XT(2)-XT(1)
      XI= (X-X0)/H +1.
      I=XI
      IF(I.GT.0) GO TO 10
      TBLU1= FT(1)
      IF(NX.GE.0)TBLU1= FT(1) + (XI-1.)*(FT(2)-FT(1))
      RETURN
   10 IF(I.LT.NA) GO TO 20
      TBLU1=FT(NA)
      IF(NX.GE.0) TBLU1= FT(NA) + (XI-NA)*(FT(NA)-FT(NA-1))
      RETURN
   20 TBLU1= FT(I) + (XI-I)*(FT(I+1)-FT(I))
      RETURN
C
C
C                                  UNEQUAL SPACED TABLE INTERPOLATION
 100  IF(X.GE.XT(1)) GO TO 30
      TBLU1=FT(1)
      IF(NX.GE.0) TBLU1= FT(1) + (X-XT(1))*(FT(2)-FT(1))/(XT(2)-XT(1))
      RETURN
   30 IF(X.LT.XT(NA)) GO TO 40
      TBLU1= FT(NA)
      IF(NX.GE.0) TBLU1=FT(NA)+(X-XT(NA))*(FT(NA)-FT(NA-1))/(XT(NA)
     1    - XT(NA-1))
      RETURN
   40 I=1
      IGE= NA
   50 II=(IGE+I)/2
      IF(X.LT.XT(II)) GO TO 60
      I= II
      GO TO 70
   60 IGE= II
   70 IF(I+1.LT.IGE) GO TO 50
      TBLU1= FT(I) + (FT(I+1)-FT(I))*(X - XT(I))/(XT(I+1)-XT(I))
      RETURN
      END
```

```
CTBLU2
      FUNCTION TBLU2(X,Y,XT,YT,FT,IX,IY,NX,NY,MX,MY)
C
C   PURPOSE    TWO DIMENSION LINEAR INTERPOLATION
C
C   METHOD     BINARY SEARCH TO FIND NEAREST GRID POINTS.
C              TBLU1 IS USED TO REDUCE THE INTERPOLATION DIMENSION.
C
C   CALL SEQUENCE
C
C
C              X,Y  - POINT AT WHICH INTERPOLATION IS DESIRED
C              XT,YT- ARRAYS CONTAINING INDEPENDENT VARIABLE GRID POINTS
C              FT   - TWO DIEMSNION ARRAY OF VALUES SUCH THAT FT(I,J)
C                     CORRESPONDS TO XT(I),YT(J).
C              IX,IY- INDICATORS FOR GRID SPACING
C                        IF IX=0  THEN XT CONTAINS EQUAL SPACED VALUES
C                        IF IX.NE.0 THEN XT CONTAINS UNEQUAL SPACED VALUES
C              NX,NY- ABS(NX),ABS(NY) ARE THE ARRAY DIMENSIONS FOR XT,YT
C                        IF NX.LT.0 THEN TRUNCATE OUTSIDE XT RANGE
C                        IF NX.GT.0 THEN EXTRAPOLATE OUTSIDE XT RANGE
C                        LIKEWISE FOR NY AND YT VALUES.
C              MX,MY- DUMMY ARGUMENTS.SET EQUAL TO ABS(NX), ABS(NY).
C
C   WRITTEN BY A.W. WARREN
C                                                   VERSION 1, JUNE 1977
      DIMENSION XT(1),YT(1),FT(1)
      NA = IABS(NX)
      MX = NA
      NB = IABS(NY)
      MY = NB
      IF(NA.GT.1)GO TO 10
      TBLU2 =  TBLU1(Y,YT,FT,IY,NY)
      RETURN
   10 IF(NB.GT.1)GO TO 20
      TBLU2 =  TBLU1(X,XT,FT,IX,NX)
      RETURN
C
C                             Y OUTSIDE YT TABLE RANGE
   20 IF( Y.GT. YT(1))GO TO 100
      E = (Y-YT(1))/(YT(2)- YT(1))
      FF1 = TBLU1(X,XT,FT(1),IX,NX)
      TBLU2 =FF1
      IF(NY.GT.0)TBLU2 =FF1+ E*( TBLU1(X,XT,FT(NA+1),IX,NX) -FF1)
      RETURN
C
  100 IF( Y.LT. YT(NB))GO TO 200
      E = (YT(NB)-Y)/(YT(NB)-YT(NB-1))
      NB1 = NA*(NB-1)+1
      FF1 = TBLU1(X,XT,FT(NB1),IX,NX)
      TBLU2 = FF1
      IF(NY.GT.0)TBLU2 = FF1+ E*(TBLU1(X,XT,FT(NB1-NA),IX,NX) -FF1)
      RETURN
C
C
C                             YT GRID SEARCH AND INTERPOLATION
  200 IF(IY.NE.0)GO TO 240
      I = (Y - YT(1))/(YT(2)-YT(1)) + 1.
      GO TO 300
  240 I=1
      IGE = NB
  250 II = (IGE+I)/2
      IF(Y.LT. YT(II))GO TO 260
      I= II
      GO TO 270
  260 IGE = II
  270 IF(I+1 .LT. IGE)GO TO 250
C
  300 E = (Y-YT(I))/(YT(I+1)-YT(I))
      I1= NA *(I-1)+1
      FF1 = TBLU1(X,XT,FT(I1),IX,NX)
      TBLU2 = FF1 + E*(TBLU1(X,XT,FT(I1+NA),IX,NX) -FF1)
      RETURN
      END
```

```
CUNIF
      SUBROUTINE UNIF(U,IX)
      COMMON /CIMPL/ IMPL,ICNT,ITEST
      DATA Y/253967./
      IF(IMPL.EQ.0 .AND. ITEST.EQ.1) IX=431469
      IF (IX.EQ.1) IX = 431469
      X= AMOD( IX*Y,16777216.)
      U= X/16777215.
      IX=X
      RETURN
      END
```