Component Based Performance Simulation of HVAC Systems

by

Michael Andrew Peter Murray B.Tech.

A Doctoral Thesis Submitted in partial

fulfilment of the requirements

for the award of

Doctor of Philosophy

of the Loughborough University of Technology

December 1984.

TEXT CUT OFF IN ORIGINAL

Synopsis

The design process of HVAC (Heating, Ventilation and Air Conditioning) systems is based upon selecting suitable components and matching their performance at an arbitrary design point, usually determined by an analysis of the peak environmental loads on a building. The part load operation of systems and plant is rarely investigated due to the complexity of the analysis and the pressure of limited design time. System simulation techniques have been developed to analyse the performance of specific commonly used systems: however these 'fixed menu' simulations do not permit appraisal of hybrid and innovative design proposals.

The thesis describes research into the development of a component based simulation technique in which any system may be represented by a network of components and their interconnecting variables. The generalised network formulation described is based upon the engineer's schematic diagram and gives the designer the same flexibility in simulation as is available in design. The formulation of suitable component algorithms using readily available performance data is discussed, the models developed being of a 'lumped parameter' steady state form.

The system component equations are solved simultaneously for a particular operating point using a gradient based non-linear optimisation algorithm. The application of several optimisation algorithms to the solution of HVAC systems is described and the limitations of these methods are discussed. Conclusions are drawn and recommendations are made for the required attributes of an optimisation algorithm to suit the particular characteristics of HVAC systems.

The structure of the simulation program developed is given and the application of the component based simulation procedure to several systems is described. The potential for the use of the simulation technique as a design tool is discussed and recommendations for further work are made.

ACKNOWLEDGEMENTS

I should like to thank Dr. Vic Hanby for his help, advice and friendly encouragement throughout the research work: other members of the Department of Civil Engineering for their interest, and the staff of the Computer Centre for their helpful advice.

Special thanks to Jonathan A. Wright for his friendship and for the use of his polynomial curve fitting routines and fan models. My thanks also to Judith Murray for proof reading the text and to Ann Gormley for proof reading and checking the mathematical notation.

The author gratefully acknowledges the S.E.R.C. for financial support of the work and my parents and family for their unfailing support and encouragemnt over the past eight years.

Contents

Chapter 1. A BASIS FOR SYSTEM SIMULATION

Interest in computer analysis of the thermal performance of buildings and heating ventilating and air-conditioning (HVAC) systems has grown out of the need and desire to improve the effectiveness of the design process. This is influenced by technological changes in materials and equipment and by economic changes, such as the relative cost of different energy sources.

Prior to the 1973 oil crisis the main emphasis was on designs for both buildings and systems which could be economically built and installed to maintain a satisfactory indoor environment. Methods of analysis for buildings have been developed to predict the energy demand of each zone of interest which enable the effects of architectural decisions on this demand to be studied. Peak loads may be identified from an analysis of the thermal performance of the building fabric together with any process loads for use in initial plant selection and thermal systems design. Most building energy analysis procedures include implicit assumptions about idealised plant characteristics which maintain constant environmental conditions in the space; however various procedures have been developed, such as the 'admittance' method (Loudon 1968), which can be used to predict the resulting environmental conditions when either no plant is installed in a zone or the installed plant capacity cannot meet the calculated thermal load.

Further advances in the thermal analysis of buildings include the assessment of the actual energy consumption of buildings and HVAC systems, taking into account the inefficiencies of individual plant components and the various thermal distribution systems. Since 1973 particular attention has been paid to the part load operation of HVAC systems, where the performance of plant may vary significantly from that at peak load design conditions. Over one hundred energy analysis procedures for HVAC system simulation have been reported, many of which are of a proprietorial nature (Kusuda 1981). Detailed information on these programs is limited by commercial interests but,

although the analytical rigour of the analysis varies considerably between them, many of the programs in use are based upon the same fundamental principles.

The implementation of system simulation as defined by ASHRAE :

'...predicting the operating quantities within a system (pressure, temperature, energy and fluid flow rates) at the condition where all energy and material balances, all equations of state of working substances and all performance characteristics of individual components are satisfied.'

has been interpreted in several different ways (ASHRAE 1975). The resulting procedures may be grouped broadly into two distinct approaches, dynamic analysis and quasi-steady state analysis, the choice between the methods being dependant upon the intended use of the study.

1.1 Dynamic Analysis

Dynamic analysis is the study of system performance subject to changes in the operating variables with respect to time and has been applied to the thermal analysis of buildings and the study of closed loop control of HVAC systems.

1.1.1 Energy Simulation

A procedure for the dynamic analysis of the thermal performance of buildings, called ESP, has been developed in the U.K. by ABACUS (Architecture and Building Aids Computer Unit - Strathclyde), to predict the effects of climate, construction, operation and plant characteristics on the energy requirement of buildings (Sussock 1978)

The individual elements of a building are represented by nodal points connected to adjacent elements to form a network scheme. Dynamic energy balance equations are formulated for each node incorporating energy transfer to and from external sources (boundary conditions) and energy transfers between nodes.

The recent expansion of ESP. to include simulation of HVAC systems is based upon the premise that there is little difference between simulation of building energy flows between building elements (eg: air to surface, surface to surface) and energy flows between plant items, or elemental nodes of the same component (ABACUS 1984).

The simulation equations must be derived independently for each characteristic nodal type not already considered in the E.S.P. system. Energy simulation equations are derived from explicit and implicit finite difference energy balance equations and must be in a general form relating energy transfers between the component and all other components to which it could feasibly be connected. In the ESP simulation procedure individual constructional elements of a building, items of HVAC plant or sub-systems, are defined as a network scheme of arcs and nodes, using terminology borrowed from mathematical network analysis. The 'nodes' represent individual components or elements and the 'arcs' represent energy exchanges between nodes. Interactive definition of these nodal networks is complicated by legality checks between different node types.

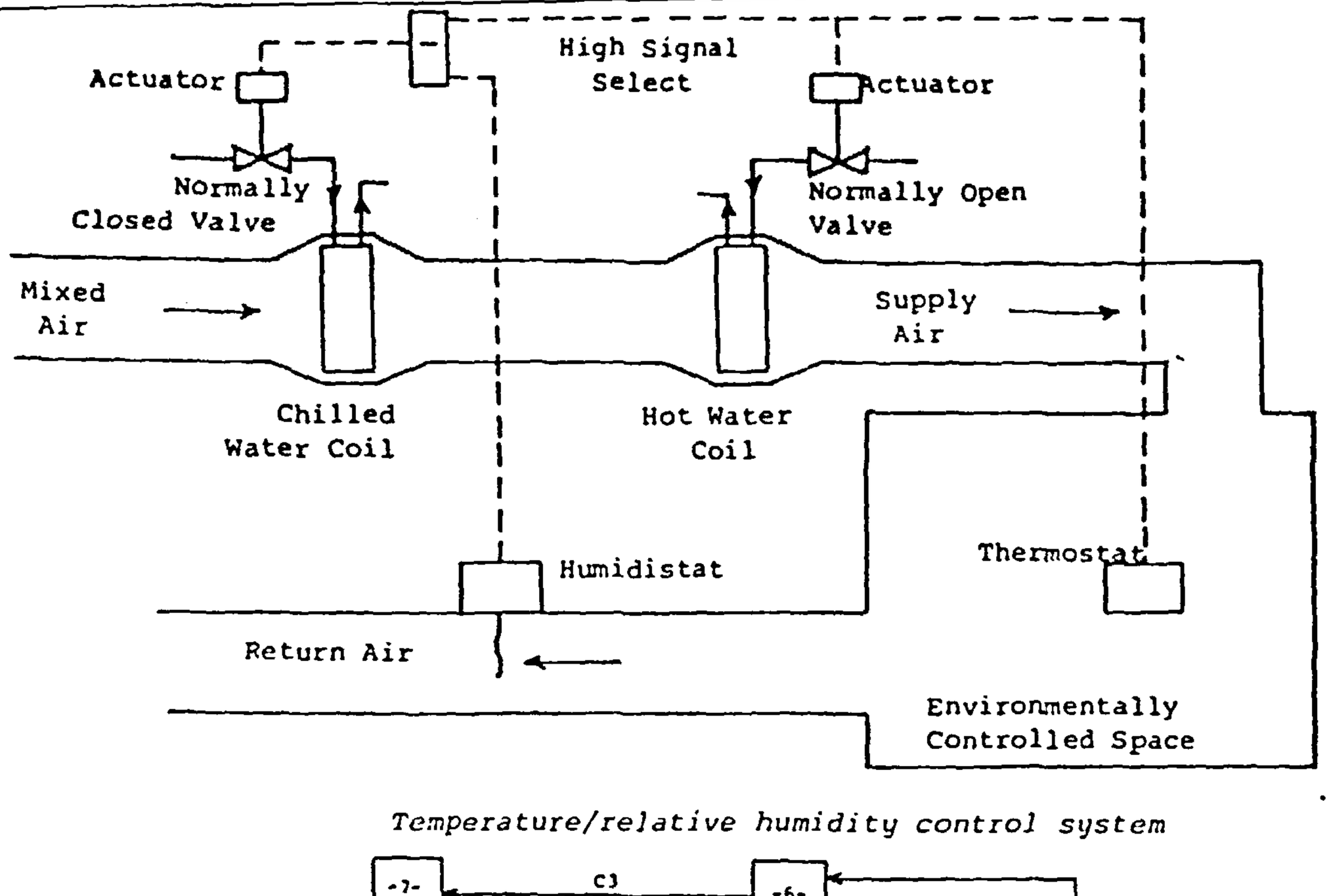
Individual components may be modelled either as a single node or by expansion to a multi-node representation, depending upon the complexity of the analysis appropriate to that particular type of component. For a simple air-conditioning system, equation sets may be developed for both energy balances and mass balances between nodes. In recirculating systems however, the mass balance would require the actual or desired diversion ratios of the two fluid loops to be specified: alternatively a system of flow simulation equations could be developed to take into account the flow characteristics of the fluid loop components. In a complex, multi fluid loop system further equation sets would need to be explicity developed, and solved, for each variable type of interest. This could be visualised as several networks, one for each variable of interest, overlaid one on top of another, each of which would have to be solved to form a complete performance simulation of a system.

The dynamic characteristics of buildings, having time constants in the order of one half to several hours, are markedly different from the dynamic characteristics of HVAC plant and systems, which have time constants measured in the order of minutes. The implicit finite difference methodology developed in ESP for the analysis of building envelopes may be appropriate for the study of the dynamic interaction between buildings and plant, but its application as a generalised system simulation design tool is limited by the explicit formulation of nodal schemes to represent various HVAC sub-systems and the assumptions about the dynamic characteristics of plant inherent in the method.

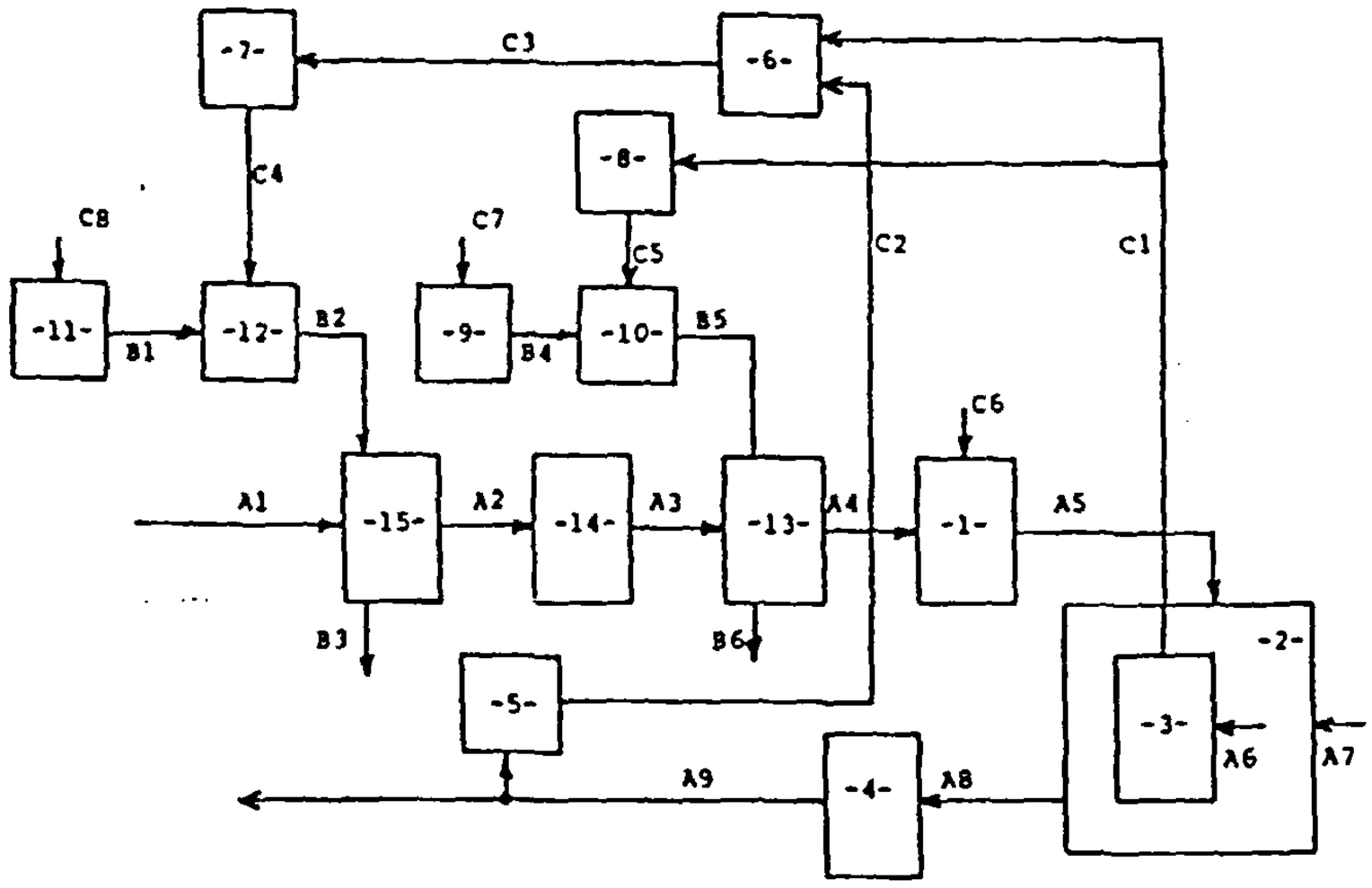
1.1.2 Simulation of Closed Loop Control

Closed loop control of HVAC systems has been simulated by several studies; however one of the insurmountable difficulties is the lack of dynamic performance data for HVAC components. In a detailed study (Marchant 1979), a finite difference model of the major elements of a domestic gas-fired boiler was formulated and the simulated performance compared against a parametric laboratory study. The model was progressively refined to obtain a reasonable correlation between the simulated and the actual performance: however the results show that the accuracy of the model is highly dependent on the thermal capacity of the various elements and the conclusion is drawn that it would be difficult to obtain the neccesary data to formulate a generally applicable model.

In another study of the dynamic performance of HVAC systems (Miller 1982) the individual components are represented by first order lumped parameter models. These are based upon derived and empirical data for transfer functions, time delays etc., and use a simulation language to develop component algorithms. Any system may be simulated by defining a network of nodes and arcs in a similar manner to the method used in ESP. The 'nodes' represent individual components which are connected together via 'arcs' carrying a vector of 'state' variables, such as the 'air state' in figure (1.1) which comprises



Temperature/relative humidity control system



Simulation representation of control system

Elements In Calculation Sequence

1. Supply air volume control.
2. Environmentally controlled space (room).
3. Thermostat.
4. Return air duct.
5. Humidistat.
6. High signal select.
7. Chilled water valve actuator.
8. Hot water valve actuator.
9. Hot water supply.
10. Hot water valve (normally open).
11. Chilled water supply.
12. Chilled water valve (normally closed).
13. Hot water coil.
14. Supply duct.
15. Chilled water coil.

A - Air states.
B - Thermal media states.
C - Control signal states.

figure 1.1 Simulation Representation of Temperature and Relative Humidity Control (after Miller 1982)

six parameters; enthalpy, humidity ratio, temperature, density, mass flow rate, and volume flow rate.

The network must be defined for a system in a specific order for the sequential solution of the equations which describe its peformance, for example, in figure (1.1), the relationship determining room temperature is solved before that determining the output signal of the thermostat. The computational interval is chosen to be approximately one tenth of the fastest dynamic characteristic of any of the elements. The results of the simulation have been used to study the interaction of various control parameters but the method depends so heavily upon the assumed dynamic characteristics of the elements that it would be difficult to formulate a generalised system model and the procedure is unsuitable for the study of either long term or full building characteristics.

The difference between dynamic closed loop control and steering control is highlighted in a study which uses a novel 'chaining and transition' process to simulate time interval performance (Borreson 1981). The term 'steering control' is applied to the quasi-steady state operation of HVAC systems in response to slowly changing external stimuli, such as air temperature. The closed loop control of a specific HVAC system is studied by developing empirical transition functions for each element in the system. During the transition process all the inputs to the element are assumed constant and the transition functions are used to predict the value of the state variables for one time step in the future. The chaining process consists of calculating the new reference values and controller outputs and using these to calculate new inputs to each element. The simplification of the analysis by the decoupling into transition and chaining processes enables the simulation to be performed on a micro-computer. The application of the procedure is limited by the empirical assumptions inherent in the development of the transition functions.

1.1.3 State Space Vector Analysis

A proprietory modelling program, GEMS (Generalised Modeling and Simulation), is based upon a state-space vector analysis of systems (Benton et al 1982). A system model described by differential equations may be cast in the form :-

differential equation for inputs $\underline{u}$:

$$\underline{\dot{x}} = [\ A\ (\underline{x},\underline{u},t)\ ]\underline{x} + [\ B\ (\underline{x},\underline{u},t)\ ]\underline{u}$$

algebraic equation for outputs $\underline{r}$:

$$\underline{r} = [\ C\ (\underline{x},\underline{u},t)\ ]\underline{x} + [\ D\ (\underline{x},\underline{u},t)\ ]\underline{u}$$

where $\underline{x}$ vector of state variables.

$\underline{\dot{x}}$ time derivative of $\underline{x}$

$\underline{u}$ vector of inputs

$\underline{r}$ vector of outputs

t time

and A, B, C, and D is the matrix quadruple of constant co-efficients. The modelling procedure depends on being able to automatically combine sub-systems cast in the state-space form into an overall system to obtain the matrix quadruple which describes the combined system.

Individual sub-systems may be modelled in several ways depending on the characteristics of the elements:

i) building fabric and structural elements may be defined as resistance-capacitance networks which can then be automatically cast into state-space form.

ii) linear systems may be modelled directly as simultaneous first order differential equations specified in state-space form in FORTRAN subroutines.

iii) systems incorporating transform functions may be specified in FORTRAN subroutines, casting the describing equations in Laplace transform form.

iv) non-linear elements may be modelled in state space form by deriving lumped parameter models based upon either a Taylor series expansion of the differential modelling equations, or from continuity equations for each elemental node within a component.

Because the state-space models of components are well ordered they may be easily combined to form a total system state-space model with interconnections between sub-systems defined completely in terms of the input and output vectors $\underline{u}$ and $\underline{r}$.

The generalised method of simulation is :-

1) calculate $\underline{\dot{x}}$ at time t as a function of $\underline{x}$, $\underline{u}$ and t

2) integrate $\underline{\dot{x}}$ from time t to time t + Δt

3) calculate output vector $\underline{r}$ for time t + Δt as a function of $\underline{x}$, $\underline{u}$ and t

The principle advantage of the state-space method is that individual sub-systems may be modelled in a form most appropriate to their performance characteristics rather than all elements in a system being constrained to one particular mode of representation. A multi-rate simulation for a complete system may be carried out with a short time step for those elements which have strong non-linear dynamic characteristics the output of which may be averaged over time for input to elements with a longer time step. The application of this state-space analysis to the dynamic performance of various HVAC systems has been reported but due to the proprietorial nature of the research details of the algorithms used are not given.

1.2 Steady State Analysis

Steady state simulation methods are based upon the assumption that, because the time constants of HVAC systems are considerably less than those of the driving functions of the building model, then a steady state analysis of HVAC systems is adequate (ASHRAE 1975).

Most steady state procedures use lumped parameter input/output models of system components in which the performance may be described in terms of the system variables; either by using experimental performance data or by deriving operational relationships from the fundamental priciples of thermodynamics and fluid flow. The resulting component model, for example a cooling coil, could be a polynomial representation of cooling capacity as a function of inlet conditions obtained from experimental observations, or could be a complex row by row analysis of heat and mass transfer across finned tube. The choice depends upon striking a balance between computational efficiency and the analytical rigour required for the intended simulation.

1.2.1 Sequential Simulation

ASHRAE recommendations for both simulation procedures and component representation have formed the basis of many of the available system simulation programs (ASHRAE 1975). The simulation process is generally based on sequential solution of the system equations with an input/output formulation of the component models. However, whilst this approach works for open loop systems, problems arise with recirculating loops where the value of one of the input variables to a component is dependant upon the output of some future component calculation. For the recirculating system shown in figure (1.2) a sequential solution method would start at the air inlet and proceed around the system in the direction of the air flow. A problem would immediately arise in calculating the outlet conditions from the mixing box since the psychrometric state of the return air is, as yet, unknown.

A common approach to this problem is to set up an iterative loop in which an estimate of the state of the return air is used to allow the calculation to proceede. The resultant return air state is then compared with the initial estimate which may be adjusted and the calculation loop repeated until the estimated and calculated values are within a specified tolerance. This approach becomes more

complicated where the system to be simulated has two or more distinct fluid loops, in which case each fluid loop must be solved iteratively before 'jumping' to another loop. In figure (1.2), the full system would be represented by three fluid loops: air, chilled water, and hot water. The initial solution for the air loop would be used to calculate the operating points of the cooling and heating coils, and an iterative procedure would then be used to calculate the actual water inlet conditions in each circuit in turn. These conditions would then be put back into the air loop and the procedure would iterate between the loops until a feasible operating point was obtained which satisfied all the system equations simulataneously. Although this solution method is widely used it can become unwieldy for multi-fluid loop systems and there is no guarantee that the iterative process will converge.

The DOE-2 program, developed by the U.S. Department of Energy, is based upon the ASHRAE proposals and consists of four main programs : LOADS, SYSTEMS, PLANT, and ECONOMICS (NESC 1981). The LOADS program computes the transient response of the building fabric to produce hourly thermal loads in each space. These thermal loads are then used by the SYSTEMS program, together with the characteristics of secondary systems to calculate the loads on the central plant. The secondary systems which may be specified are drawn from a menu of approximately twenty-five options together with several control schemes and operating schedules. The energy load data is then used by the PLANT program to simulate the performance of the central plant, which may be selected from a menu of available options which includes conventional heating and cooling equipment together with co-generation and solar systems. The ECONOMICS program provides a life cycle cost analysis to estimate the relative costs of the various options.

One of the problems of these simulations is the limited number of systems which may be analysed. One response is to expand the menu of available systems, such as NASA's NECAP program which, together with

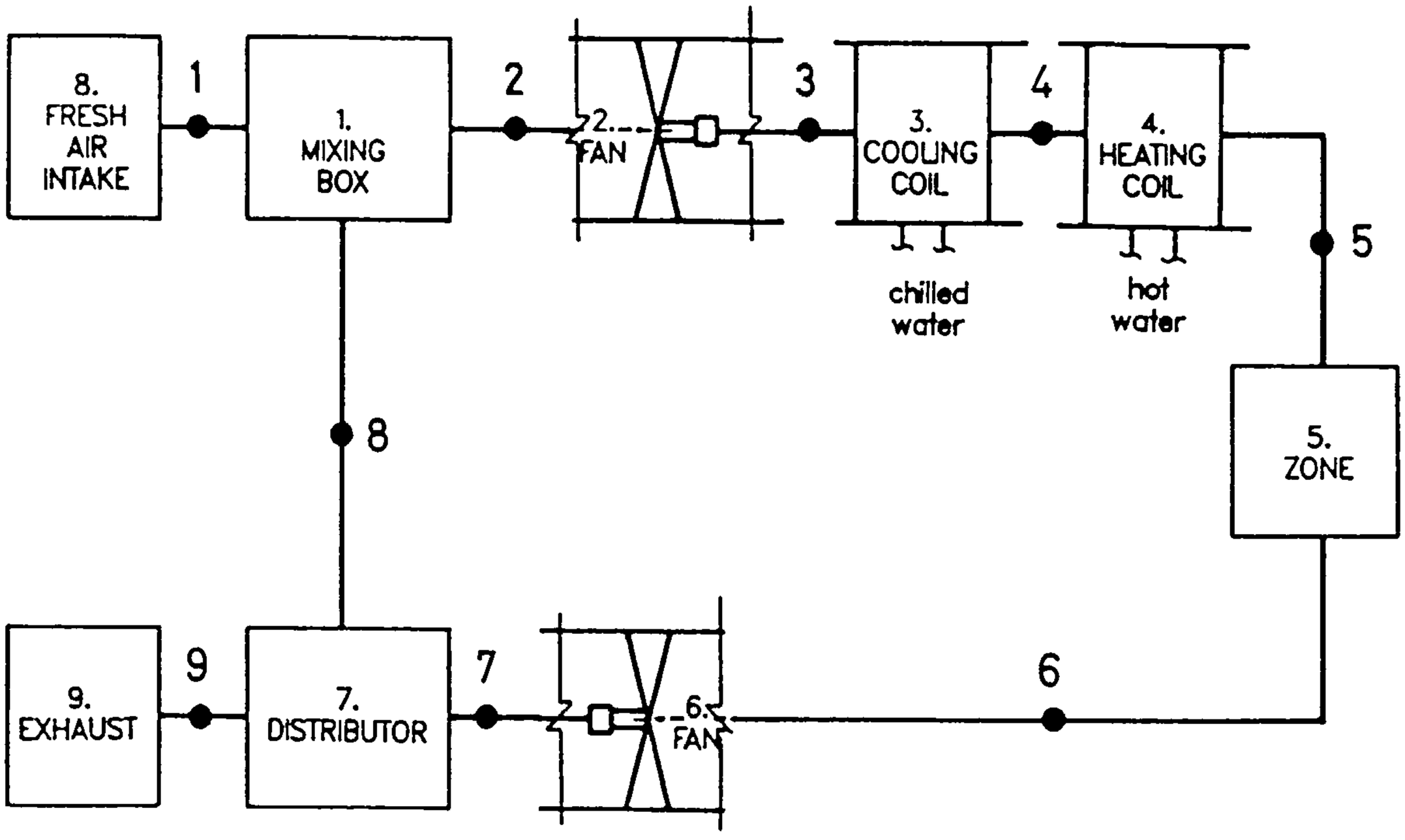


figure 1.2 Single Zone Recirculating System

between nodes 1 and 2	mixing box
between nodes 2 and 3	fan
between nodes 3 and 4	cooling coil
between nodes 4 and 5	heating coil
between nodes 5 and 6	zone
between nodes 6 and 7	fan
between nodes 7,8 and 9	distributor

Table (1.1) Single Zone System Definition for figure (1.2)
(after Quick 1982).

a multitude of control strategies enables over 3 x 10^{10} systems to be analysed (M^{c}Nally 1976). Most simulation programs have far fewer options than this but an experienced user may be able to adapt the fixed systems available to approximate the specific proposals for a particular application. However with increasing demand for energy efficient system designs, innovative and hybrid systems may be difficult to analyse. A further dissadvantage of expanding the menu of systems available is that, since each system schematic must be programmed explicitly, as the menu of systems expands so do the computer facilities required.

A quasi-dynamic approach to system simulation has been developed in the U.K. where a system schematic may be built up from individual components by describing the points in the system where the components are joined (Quick 1982). This approach is known as component based simulation. The system is defined as a network of nodes and arcs but, unlike the network schemes described in section 1.1, the 'nodes' have no physical significance but represent the 'state' of the interconnecting variables whilst the components are defined on the 'arcs' between them (figure 1.2 and table 1.1). The data describing the performance of each component, such as fan characteristics and cooler coil contact factors is read from files

A control scheme may be imposed on the system network by utilising 'perfect' control, eg: temperature at node 4 controlled at 16 °C, or by scheduling a nodal value based upon a sensed condition at another node, eg: flow at node 3 controlled by a proportional controller based upon comfort temperature at node 6. The system flows are determined from controllers which use the value of the control variable from the previous time step (Irving 1982)

The component models are cast in an input/output format and the system is solved sequentially for each time-step. To reduce the need for iteration where recirculating loops occur, the program uses the value of the recirculating variable from the previous time step as an estimate to allow the calculation to proceed. Because the simulation

time step is short, typically five to ten minutes, the error introduced by using an estimate of the recirculating variables from the previous period is assumed to be negligible.

Although this component based procedure overcomes the limitations of fixed menu simulations, it is difficult to program a generalised sequential solution methodology which could be used to simulate systems with more than one fluid loop.

1.2.2 Simultaneous Simulation

By solving the component equations for a system simultaneously a generalised simulation procedure can be developed which will be capable of solving multiple fluid loop systems. Simultaneous solution methods start from an initial estimate of the solution point and use some form of 'hill-climbing' search algorithm to minimise the error between the estimated point and the actual operating point of the sytem.

One component based approach uses a sophisticated gradient based optimisation algorithm to find values for the system which satisfy a set of constraints representing the characteristic equations of the system components (Silverman 1981). The system is represented by a network of nodes and arcs similar to the scheme described in section 1.1: the nodes corresponding to individual components of plant whilst the arcs represent 'state' vectors containing variables such as fresh air fraction, mass flow rate, and enthalpy.

The definition of the system shown in figure (1.2) in this manner is shown in tables (1.2) to (1.4). Each node is given a number and a node type (NODTYP) which identifies which constraints that particular node invokes. Each arc is specifically identified (in NARTAB) as going 'from' one node 'to' another and the corresponding state variables for the fluid are indexed.

Node	NODTYP	Description	Constraint types invoked
1	2	Mixing Box	2,3,4,5
2	3	Fan	17
3	4	Cooling Coil	6,16,19
4	5	Heating Coil	18
5	6	Zone	10,11,12,13,14,15
6	3	Fan	17
7	7	Distributor	7
8	1	Fresh Air Intake	none
9	8	Exhaust Air Outlet	none

Table (1.2) Node Types (after Silverman 1981)

arc	from	to	r	m	h	w
1	8	1	-1	1	-1	-1
2	1	2	2	3	4	5
3	2	3	2	3	6	5
4	3	4	2	3	7	8
5	4	5	2	3	9	8
6	5	6	10	11	12	13
7	6	7	10	11	14	13
8	7	1	10	15	14	13
9	7	9	10	16	14	13

r - fresh air ratio
m - mass flow rate
h - enthalpy
w - moisture content

Table (1.3) Array NARTAB (after Silverman 1981)

Type	Description
2	collector mass balance
3	collector enthalpy balance
4	collector moisture balance
5	collector fresh air balance
6	cooling coil not allowed to heat
7	distributor mass balance
10	zone temperature lower bound
11	zone temperature upper bound
12	zone R.H. upper bound
13	fresh air fraction lower bound
14	zone latent load = Δg
15	zone sensible load = Δh
16	coil latent heat removal
17	fan energy conservation
18	heating coil not allowed to cool
19	cooling coil maximum capacity

Table (1.4) Constraints (after Silverman 1981)

Certain arc-variables are determined by external factors, such as weather data, occupancy data etc, and are defined as 'exogenous' in EXO. This expression simply means that these variables are external to the solution process and by altering the values of exogenous variables the load imposed on the system may be changed.

The unique identification of the state variables is done by placing a -1 in the appropriate column of NARTAB where the variable is exogenous and a -2 where a state variable is not used. The remaining spaces in NARTAB are numbered with a sequence of positive integers to uniquely define each state variable, eg: mass flowrate in arcs 2,3,4, and 5 is the same and is given the identity number 3.

The exogenous variables in NARTAB are defined in EXO and the remaining variables represent the operating point of the system. The solution algorithm seeks to find values for these variables which satisfy all of the constraints. The simulation may be critically constrained where the number of arcvariables is exactly equal to the number of constraints and the system will have a unique solution. More usually there are more arc-variables than constraints so that there are many feasible solutions: in this case either conventional controls schemes may be imposed on the system to effectively make it critically constrained, or the solution algorithm may be used to optimise an objective function, such as total energy consumption, to yield an optimum operating point.

This network approach to component based simulation is limited in several ways. Despite the generalised form of the solution algorithm, only single fluid networks may be represented due to the limitation of defining a system as a series of arcs carring a limited number of 'state' vectors. Component models are of the input/output form but are defined as invoking a number of constraints which represent mass balances, energy balances etc. Inequality constraints are used to maintain controlled conditions within upper and lower limits (constraint types 10,11 and 12 table (1.3)) but the simulation of

conventional controls is reported to be difficult.

A recent paper (Sowell 1984) details the expansion and refinement of the work by Silverman to include analysis of controls. However this is at the expense of one of the fundamental objectives of the earlier research - to produce a generalised network simulation. In the latest work a mainframe computer technique is used to generate a sequential algorithm for the solution of HVAC networks. The HVAC system is defined in much the same way as in Silverman, but instead of using the gradient based solution method the latter work uses 'graph theory' to determine a specific order for the solution of the system equations, including iterative loops if neccesary. The network solution algorithm generated on the mainframe is 'down loaded' onto a desk-top micro-computer for use as a design tool. Although the analysis is still claimed to to be component based much of the advantage of network concepts as a design tool is lost in the automatic generation of specific sequential solution algorithms.

1.3 Proposals for a Component Based Simulation Procedure.

The research presented in this thesis outlines the development of a system simulation procedure for use as a design tool, which attempts to overcome some of the limitations of the simulation methods detailed above.

i) **System Definition** - The method of defining a system must be completely generalised and must reflect the same flexibility in simulation as is generally available in design. The method must be user friendly without restrictions to variable state vectors or a limited number of fluid loops. Control equipment should be included within the general system definition rather than being imposed upon the simulation externally.

ii) **Component Models** - The procedure should use the most commonly available type of component model which is the steady state input/output form using either manufacturers' experimental

performance data or the accepted laws of heat transfer and fluid dynamics. As new component algorithms are formulated, published and validated, they must be capable of being easily incorporated into the simulation.

iii) **Solution Techniques** - An algorithm is required for the simultaneous solution of the equations which describe the peformance of the components in a HVAC system. Techniques of optimisation can be applied to reduce an objective function formed from the describing equations, which may be non-linear and possibly discontinuous. The objective function may be formulated in different ways to enable the application of several optimisation methods to be studied in order to detemine the characteristics of an appropriate algorithm.

Chapter 2. GENERALISED SYSTEM DESCRIPTION

Component based HVAC system simulation requires a method by which any HVAC system or sub-system may be modelled. A criticism of many computer aided design tools is that the data input and checking is such a complex and tedious task that the advantages of computer techniques are lost and it may be as well to recourse to manual methods. As part of this study an inter-active, user friendly method of specifying a system for simulation has been developed in which the engineer or designer can have complete control over the system definition process. A transparent programming technique, in which the simulation is as flexible as the design process and where the engineer can have a feel for the models developed, is more acceptable than 'black box' techniques where the inherent assumptions within the system model are not readily apparent.

The system definition is cast in the familiar terms of the engineering schematic diagram which is readily understood and accepted by experienced design engineers. This process is inter-related to the form of the component models, which is discussed in chapter 3, and has the flexibility to incorpororate a wide range of component algorithms of differing levels of sophistication.

2.1 Network Formulation

In a HVAC design the individual environmental plant components are connected to other components by pipes, ducts and control systems to form a complete system. An engineer's schematic diagram may be represented by a network of 'nodes' and 'arcs' ; the nodes representing components such as fans and boilers, and the arcs representing the inter-connecting pipes and ducts. The concept of a network analysis of HVAC systems has been interpreted in a number of different ways by other researchers (Quick 1982, Miller 1982, Silverman 1981).

The process of defining a system as a network may best be explained by reference to a schematic diagram of a hypothetical system (figure 2.1). In this system one zone is heated by a ducted fan and coil system, the other by a hydronic radiator system. The water is heated by the boiler maintaining a water flow temperature through the action of the proportional controller and the modulating valve. The air zone temperature is regulated through the action of the proportional controller and the three-port diverting valve whilst the 'wet' zone air temperature is maintained by the action of the thermostatic radiator valve responding to the proportional controller. The pressure characteristics of the hydronic circuit are not included within this model.

Each component type has a certain number of system variables associated with it, referred to as arc-variables, which must be uniquely identified within the system. Where a system variable is associated with more than one node it is given one identifying arc-variable number, for example in figure (2.1) water flow temperature from boiler 20 = tw-input to radiator = tw-supply to diverting valve. The identification and numbering of arc-variables in a network may be in any arbitrary but consistent manner; a description of the interactive process of network definition is given in chapter 5.

Unlike other network descriptions of HVAC systems, the 'arcs' of the network have no specific physical meaning but are merely information flow lines. This concept of defining the inter-connection between components in terms of individual arc-variables means that the system definition is not limited to a number of 'state' variable vectors, nor is it limited to certain fluid loops.

Although the arc-variable definition of a system is based upon the inter-connection of 'real' variables, such as mass flow rate, temperature etc. it is possible to define pseudo variables which do not occur naturally but represent calculated quantities which can be used in the network definition. Examples of pseudo variables are

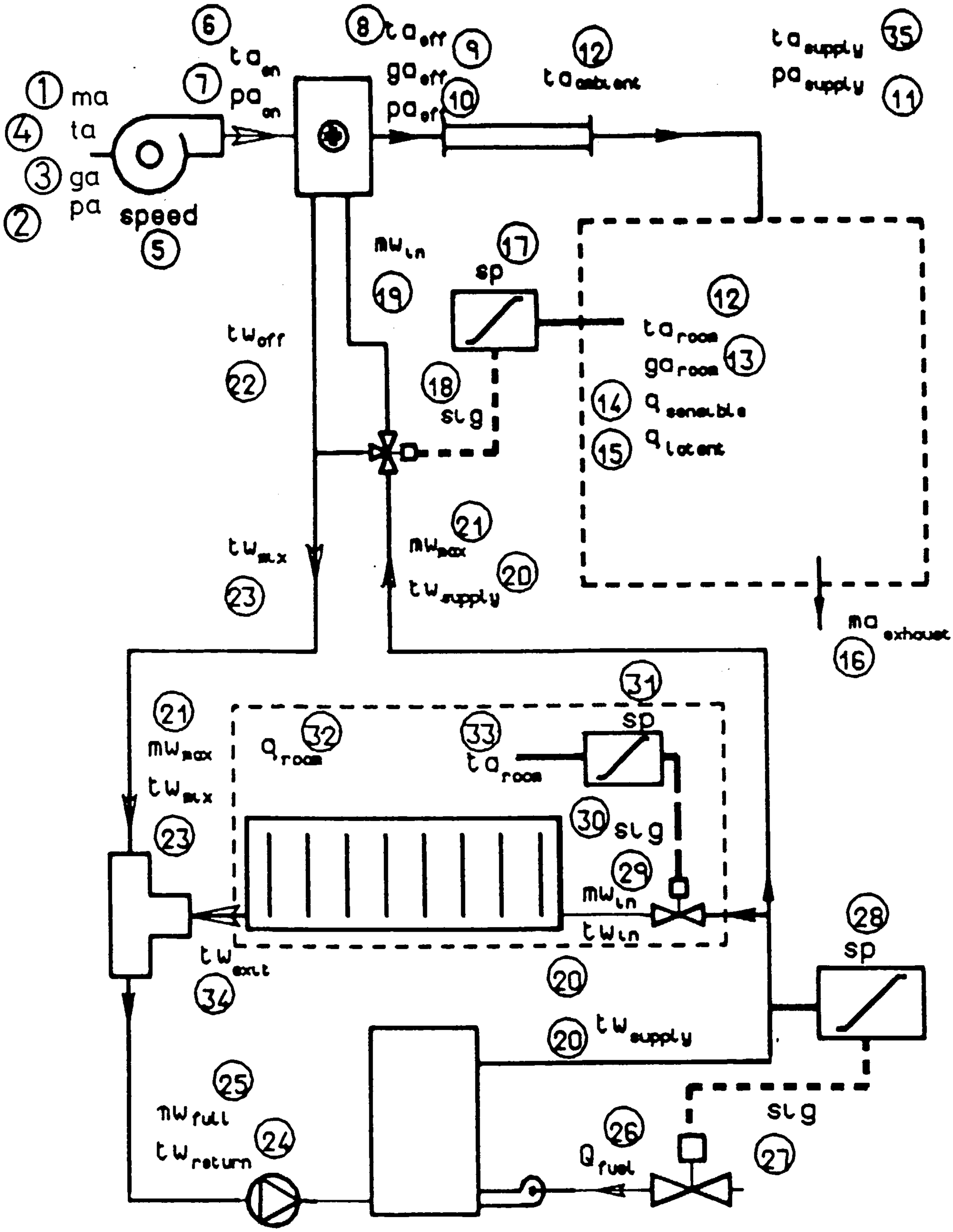


figure 2.1 Simulation Network for TWOZONE System

heat transfer rates, controller signals or actuator inputs : values which would necessarily form part of a manual analysis but are not physicaly measurable quantities. This enables certain component models to use other system variables within a calculation sequence without there being a direct physical connection between the components. For example the characteristics of dampers and valves are dependent upon the authority of the device within the circuit in which it is used, where authority is the ratio of pressure drop across the device to the pressure drop across the rest of the circuit. Although there is no physical connection between them, the appropriate pressure arc-variables could be input to the valve model to enable the valve authority to be internally calculated.

2.2 Exogenous Variables

Some of the arc-variables which define the network will not be variables within the simulation but will be determined externally, such as weather parameters. These variables are referred to as 'exogenous', meaning external to the simulation, and are of special significance in the solution process. Generally the number of exogenous variables in a network should equal the number of arc-variables less the total number of equations in the system: this ensures that the system is represented by 'n' equations in exactly 'n' unknowns.

2.3 Component Constants

The performance of a particular component is defined by an algorithm comprising a generalised description of that node type which requires a number of data constants to form a complete mathematical model for that particular item of equipment. The formulation of component models is discussed fully in chapter 3, but the component data can be in any of three forms; constants, polynomial co-efficients or exogenous constants.

Constants include such data as maximum capacity rates, or the physical size of equipment such as the number of rows in a coil or the height of a radiator. Polynomial co-efficients are derived from curve-fitting manufacturers' or experimental data. The constants and polynomial co-efficients for a particular item of equipment are held in a structured data file which may be read during system definition. 'Exogenous constants' is a term used to describe those attributes of a component which would be fixed during system design, such as coil face area or the length of a radiator. The use of exogenous constants is a convienient way of reducing the number of data files required to specify a complete range of components, for example the radiator model described in Appendix A-4 has only one data file for each combination of radiator height and number of panels, the length of each radiator in a system is defined as an exogenous constant which is fixed during system definition.

2.4 Network Definition

The system shown in figure (2.1) has thirteen components each represented by a node in the network. The nodes are numbered sequentially, in any order, and a node type for each is selected from a menu of available component models. This node by node system definition is indexed by each node forming a row in a two dimensional array 'NET' (table 2.1). The node type is entered in the first column of array NET and is used to call an initialisation routine which defines the number of variables, exogenous constants, data constants, polynomials and equations associated with that node.

The arc-variables associated with the component are indexed in NET and the initial estimates, upper and lower bounds are read from a selected data record and held in arrays 'ARCVAR', 'UB', and 'LB' respectively. The exogenous constants for a network are numbered sequentially and the values are kept in array 'EXCON'. The numerical constants for each node are automatically assigned an indexing number in NET and the values are held in array 'CONST'. An index is also kept in 'NET' of any polynomials used in the network and the

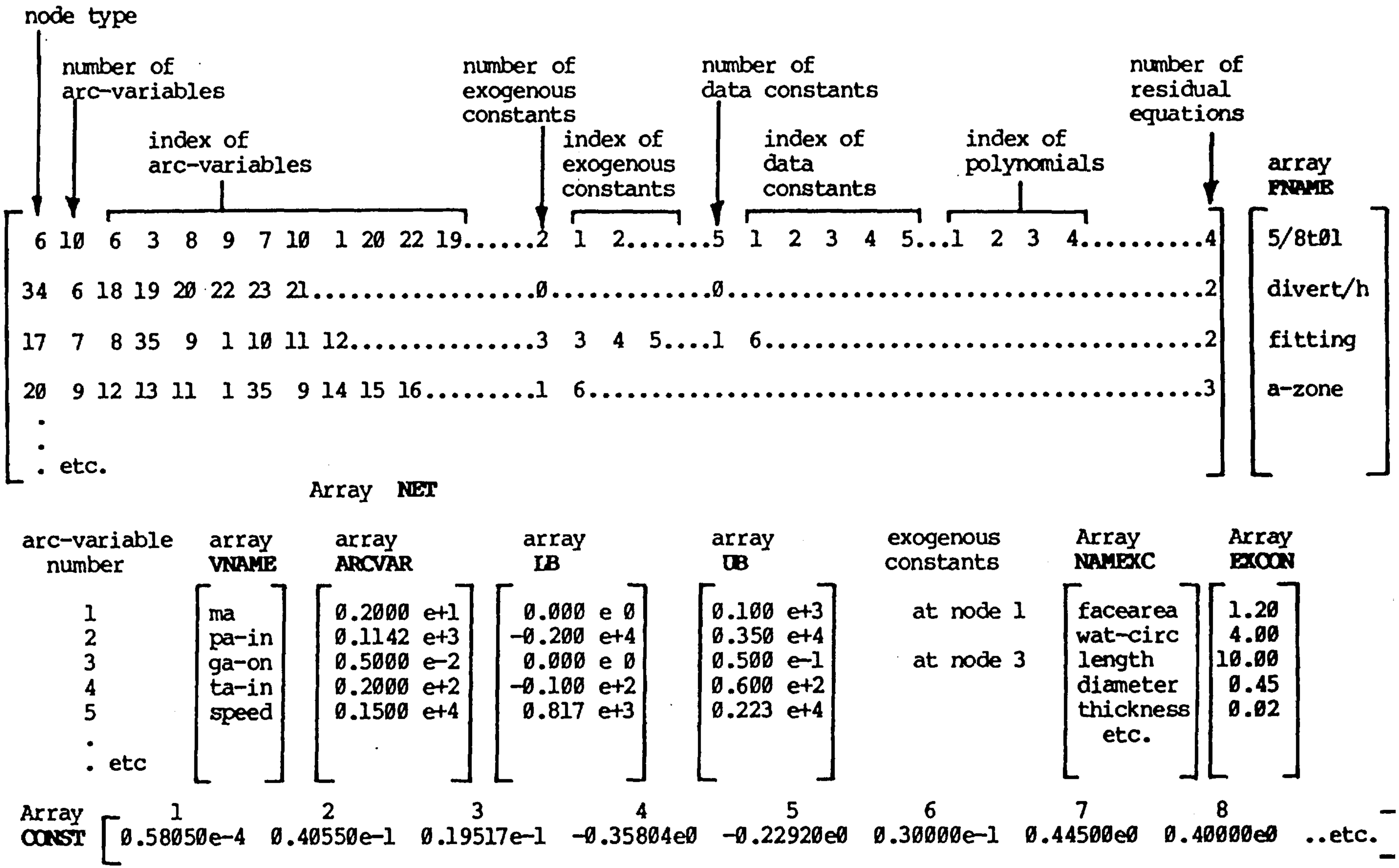


Table 2.1 Network Arrays for **TWO-ZONE System.**

polynomial co-efficients are read from the data record and stored in array 'NETCOE'. The number of equations associated with each node is entered as the last element in each row of array NET and an index of which arc-variables are declared exogenous is kept in array 'EXVAR'. Additional arrays are used to hold names for each arc-variable, 'VNAME', for each exogenous constant, 'NAMEXC' and a list of data record names for each node in 'FNAME'.

The system definition, as represented by the various network arrays is written to a file for use in the simulation procedure. The network definition may be recalled and modified under program control as described in chapter 5.

Chapter 3. COMPONENT MODELS

The generalised technique of describing a HVAC system as a network of components and their interconnecting variables, and the methods of determining the operating point of these systems, both influence the formulation of suitable component models. The aim of a component model is to describe the performance of a piece of plant in terms of the system variables, either using manufacturers' experimental data or by deriving operational characteristics from the fundamental priciples of thermodynamics and fluid flow. Since a major objective of this research is to develop a methodology for a simulation design tool, the form of the component models must be such that performance data from a wide variety of sources, presented in different ways, can be transcribed into compatible component models. The level of computational sophistication of the models must be uniform since it is pointless to develop a dynamic finite difference algorithm for one component for use in conjunction with a steady state model of another component.

3.1 Limitations of Available Data

Several attempts to develop accurate dynamic models of manufactured plant have foundered because of the lack of information on the dynamic performance of equipment. Data published by manufacturers is usually for specific test conditions under steady state full load operation; part load steady state data is scarce and dynamic performance tests are rarely carried out comprehensively across the whole of the range of a manufactured item.

There are two common approaches to developing dynamic models of components : finite difference techniques and differential equations.

i) Using a multi-node representation of the component, dynamic finite difference energy balance equations may be developed to describe the energy exchanges between the nodes (ABACUS 1984, Marchant 1979). However as the results of the study by Marchant indicate (section 1.1.2) it would be extremely difficult to extend this approach to generalised models of a range of equipment.

ii) Lumped parameter differential equations may be developed to describe the response of components with respect to time. In a first order model, such as those used in TRYNSYS , the component is assumed to have a uniform property, such as temperature, and differential equations are developed to describe the change in this property with respect to time (TRNSYS 1983). The differential equations must then be integrated to simulate the performance of the component.

The transient response of components may be modelled using transfer functions and time delays, possibly using one of a number of simulation languages available. These languages are an extension to FORTRAN which include specific functions for time delays, transfer functions, integration etc., but require extensive programming expertise and dynamic performance data to develop component simulation routines. These simulations give a valuable insight into closed loop control, and in particular they have been used to study control instability, but the results of such work can only yield a set of generalised guidlines for use in the design of similar systems.

In terms of system simulation procedures the likely conclusion of an I.E.A. (International Energy Agency) Annex 10 simulation exercise which studied the comparative performance of several boiler models, is that steady state models of components may be most appropriate for energy calculations (Hanby 1984)

Several of the steady state component models outlined in Appendix A are based upon currently available manufacturers performance data which is generally published to assist design engineers in selecting and specifying equipment: since most design calculations are for peak load conditions, part load data is rarely available. Where suitable experimental data is not generally available the components are modelled by developing algorithms based upon the commonly accepted principles of thermo-fluid dynamics.

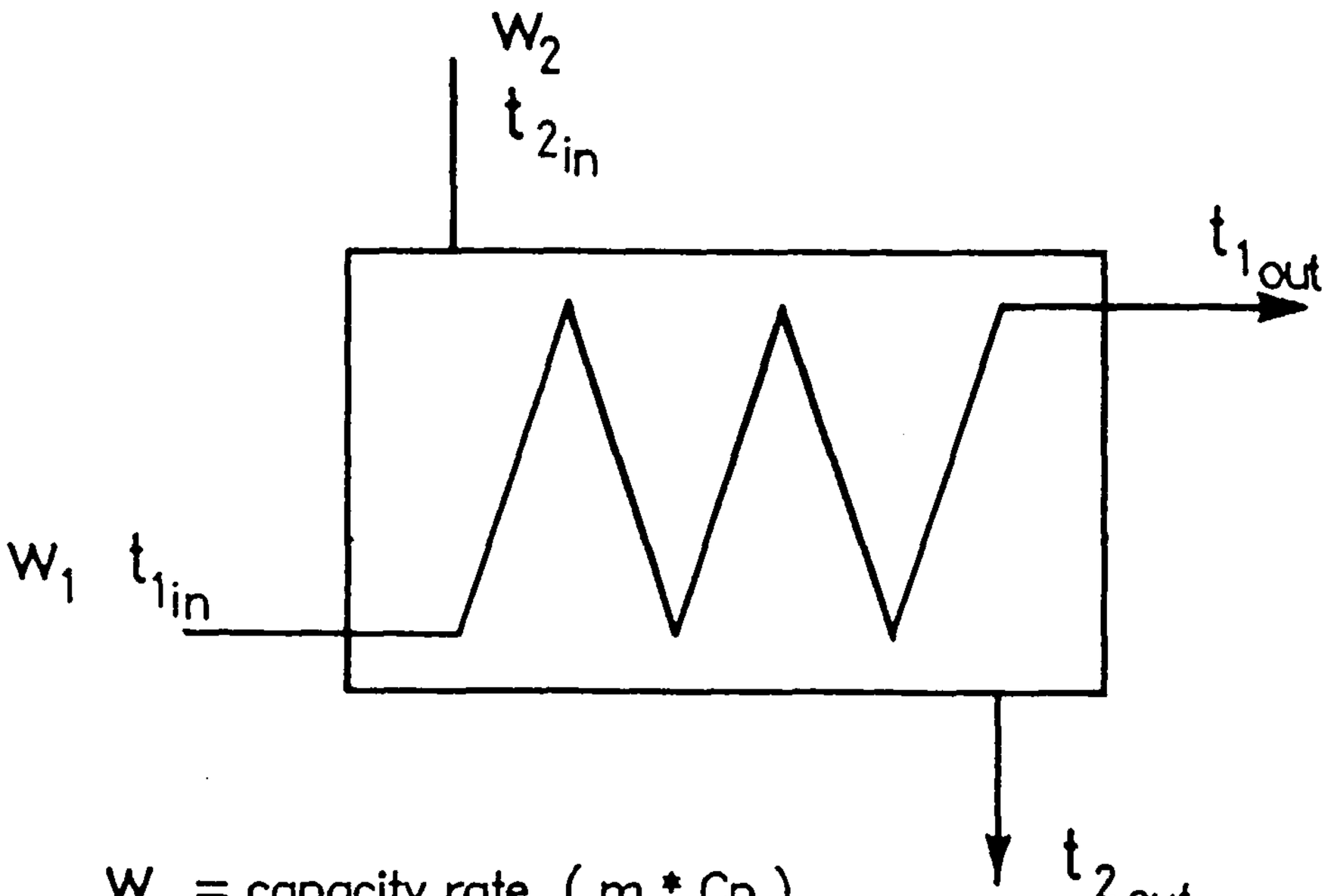
Care must be excercised in interpreting manufacturer's performance data to ensure that the conditions of the test procedure are similar to the conditions likely to be encountered in a HVAC installation. The characteristics of a boiler, for example, are often developed from a bench test in which the boiler return is held at a constant 50°C , however most systems are designed for a 10-20°C differential between flow and return temperatures with the flow temperature maintained at 80-90°C. Thus the operating conditions in the simulation are likely to be markedly different than those in a performance test.

The development of a suitable component algorithm using published data is most appropriate where performance test are carried out to an agreed standard and the results are published in the same format by different manufacturers. If system simulation becomes widely used it should be possible to persuade manufactures to publish appropriate performance equations or polynomial co-efficients, and to develop standardised procedures and reporting formats for comprehensive performance tests.

3.2 Algorithmic Structure

Steady state component algorithms usually contain component equations which describe the performance of the component in terms of the system variables. These describing equations can be in an explicit or implicit form. Explicit equations may be used in a deterministic sequential solution algorithm to determine the performance of each component in a system. Solution of implicit equations generally requires iteration within a sequential algorithm, or alternatively the equations may be solved simultaneously.

Most steady state component models are of an explicit 'input/output' form in which the describing equations use the input variables to determine values for the output variables. A simple heat exchanger is shown in figure (3.1), where w_i is the fluid capacity rate ($m_i * cp_i$) and t_i is the fluid temperature.



W = capacity rate (m * Cp)

t = fluid temperature

figure 3.1 Simple Heat Exchanger

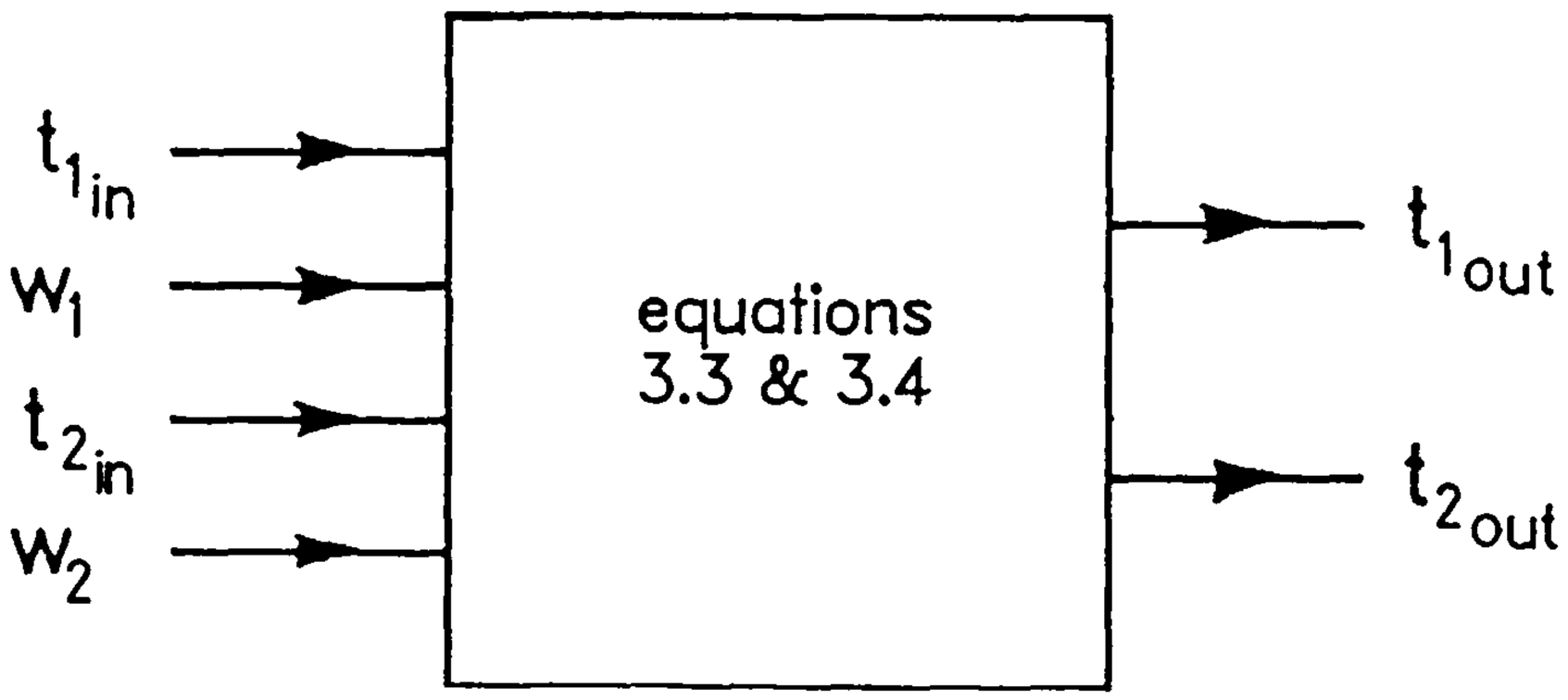


figure 3.2 Input – Output Model for Heat Exchanger

The performance of the component may be described by two equations:

$$w_{min}\ \varepsilon\ (t^1_{in} - t^2_{in}) = w^1\ (t^1_{in} - t^1_{out}) \qquad (3.1)$$

$$w_{min}\ \varepsilon\ (t^1_{in} - t^2_{in}) = w^2\ (t^2_{in} - t^2_{out}) \qquad (3.2)$$

where ε is the effectivness of the heat exchanger. In an input /output model these can be re-arranged to explicitly determine the output values for each fluid :-

$$t^1_{out} = t^1_{in} - w_{min}/w^1\ \varepsilon\ (t^1_{in} - t^2_{in}) \qquad (3.3)$$

$$t^2_{out} = t^2_{in} - w_{min}/w^2\ \varepsilon\ (t^1_{in} - t^2_{in}) \qquad (3.4)$$

Thus the output values t^1_{out} and t^2_{out} are explicitly determined from the input values t^1_{in}, w^1 and t^2_{in}, w^2 and the component model would be of the form shown in figure (3.2). The component modeling procedures recommended by ASHRAE and many of the published algorithms are of this format (ASHRAE 1975).

Implicit equations and explicit equations may be solved simultaneously by casting them in a constraint or residual form. The equations (3.1) and (3.2) may be re-arranged as

$$F^1 = 0 = w^1\ (t^1_{in} - t^1_{out}) - w_{min}\ \varepsilon\ (t^1_{in} - t^2_{in}) \qquad (3.5)$$

$$F^2 = 0 = w^2\ (t^2_{in} - t^2_{out}) - w_{min}\ \varepsilon\ (t^1_{in} - t^2_{in}) \qquad (3.6)$$

At the operating point of the exchanger defined by w^1, t^1_{in}, w^2 and t^2_{in}, the equations (3.5) and (3.6) would be satisfied exactly and the residuals F^1 and F^2 would be zero. For any other values of the two outlet temperatures, the residuals would take some finite value. The solution algorithms, discussed in chapter 4, use the values of the residuals as an estimate of the error between the current point and the actual operating point of the system.

3.3 Component Algorithms

The development of a library of component algorithms and the assimilation of appropriate performance data is a major task and has required the collaboration of other researchers at Loughborough. Since this thesis is concerned with the development of the system simulation methodology, recourse has been made where possible to

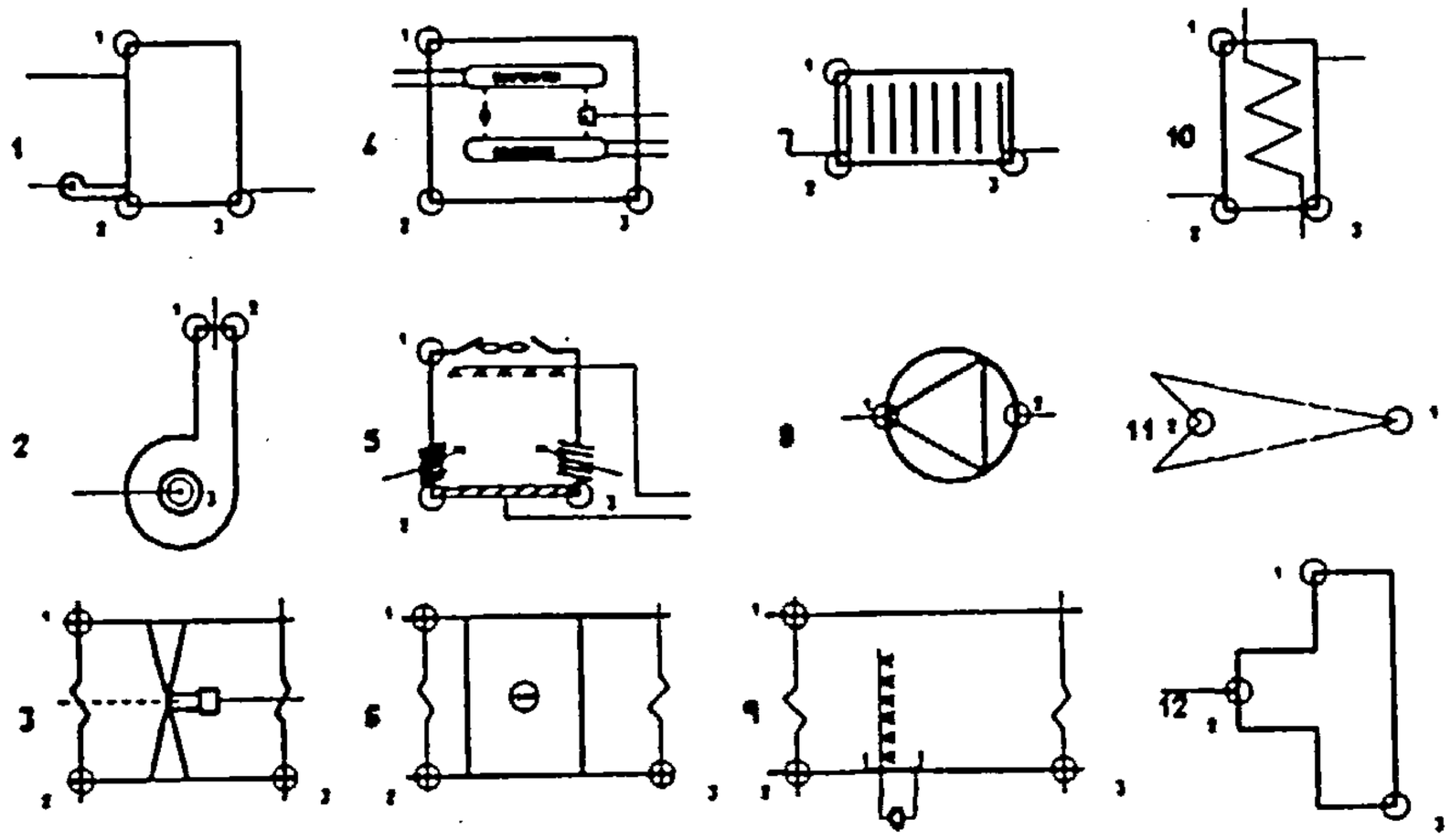
published algorithms. Where published algorithms are not available and where an appropriate algorithm was not under development elsewhere the aim has been to adopt a simple algorithm which nevertheless exhibits similar properties to those expected of a more rigorously developed algorithm. Figure (3.3) shows the currently available component types and details of each of the algorithms currently in use are given in Appendix A.

A fundamental approach to developing an algorithm for a particular component is appropriate where the relationship between the system variables which describes the performance of the component is simple, for example mass and energy balances across mixing tees. Fundamental models of components in which the relationships between the system variables are not simplistic, such as for heating and cooling coils, have been developed because this approach permits the many variables in equipment design to be taken into account.

Complex equipment such as boilers and chillers may best be modelled from experimental data: fundamental relationships could be developed for these components but this would require many more system variables and residual equations, furthermore it would be difficult to obtain the comprehensive constructional data from all manufacturers which would be required to develop a generally applicable model.

In these cases, where the component algorithms are based on published experimental data, extensive use has been made, where appropriate, of a comprehensive polynomial curve fitting program (Wright 1984). This program allows data to be entered either via the keyboard from tables of published performance data or by using a digitiser on published performance curves. Polynomials in one or two variables may be fitted to the data and the optimum powers of fit for each dimension is determined automatically. The resulting co-efficients are written to a data file which may be read directly into a component data record. The program also offers graphics options to plot the

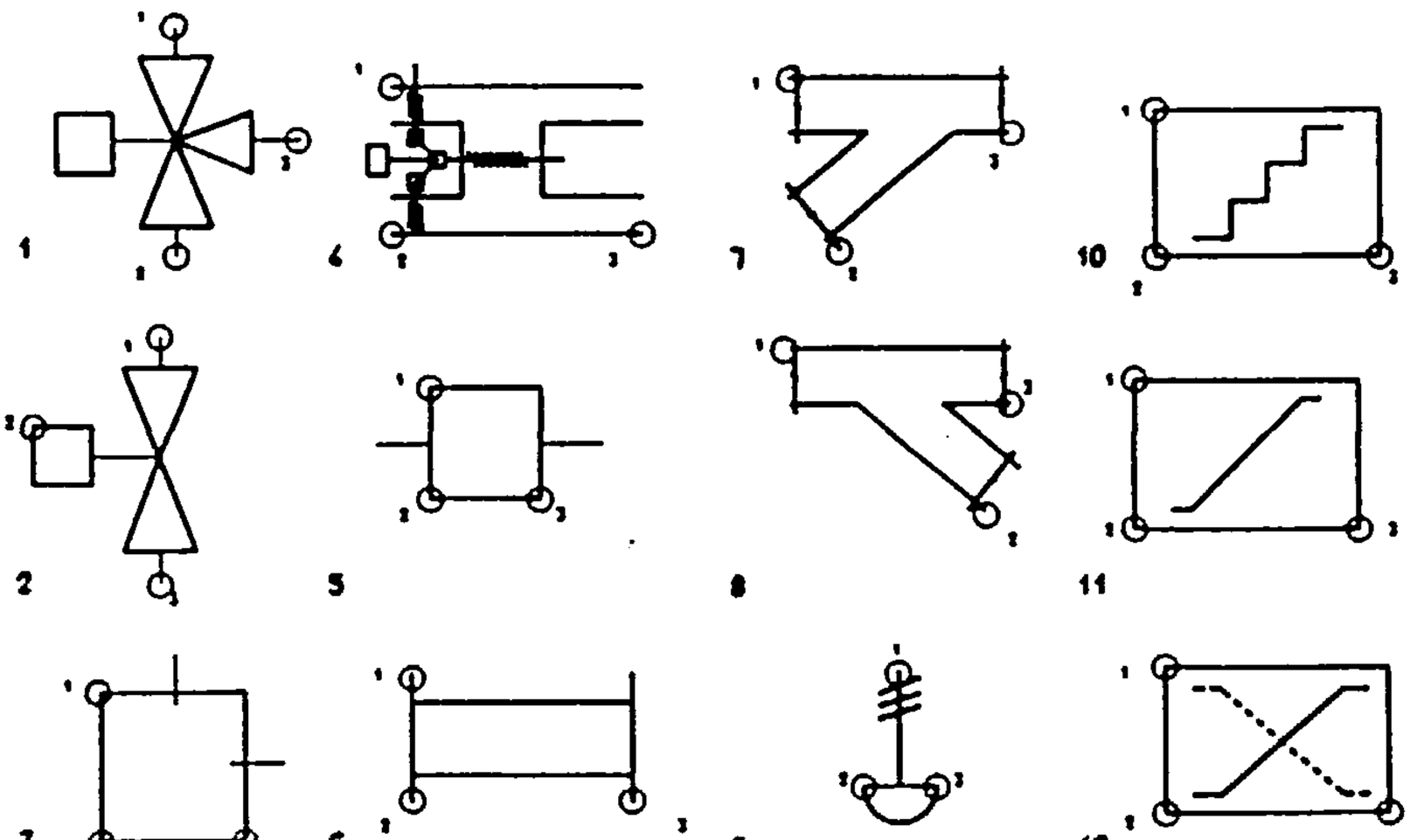
MAIN PLANT	FITTINGS	CONTROLS	TEST NODES
1 - boilerst	16 - mix-tees	31 - mixvalve	46 -
2 - axialfan	17 - duct-ins	32 - modvalve	47 -
3 - cent-fan	18 - ventecon	33 -	48 -
4 - wchiller	19 - roomzone	34 - d-valve	49 -
5 - clgtower	20 - air-zone	35 - hmdrctrl	50 - room-rad
6 - h/c-coil	21 - conv-wye	36 - stepcont	51 -
7 - radiator	22 - div--wye	37 - pcontrol	52 -
8 - compresr	23 - duct-siz	38 - siginvtr	53 -
9 - heatexch	24 - duct-sim	39 -	54 -
10 - humidifr	25 - ftgs-sim	40 -	55 -
11 -	26 -	41 -	56 -
12 -	27 -	42 -	57 -
13 -	28 -	43 -	58 -
14 -	29 -	44 -	59 -
15 -	30 -	45 -	60 -



SYMBOLS SHEET 1

KEY:

1-BOILER 4-CHILLER 7-RADIATOR 10-HEATEXCH
2-CENT-FAN 5-CLGTOWER 8-COMPRESR 11-ARROW
3-AXIALFAN 6-COIL 9-HUMIDIFR 12-MIX-TEE



SYMBOLS SHEET 2

KEY:

1-MIXVALVE 4-ECONISER 7-CONV-VYE 10-STEPCONT
2-MODVALVE 5-FITTING 8-DIV-VYE 11-PROPCONT
3-AIR-ZONE 6-DUCT 9-DAMPER 12-SIGPROC

figure 3.3 SPATS Component Menu

resulting polynomials against the original data as a visual check on the validity of the curve fit.

One advantage of this curve fitting routine is that the exact degree of the polynomial is not fixed by the component algorithm but may be determined independently for each item of equipment to be modelled. However the polynomials must be used with care since the fit is only valid within the range of the original data. Although the curve fitting routines currently determine the best polynomial representation of performance data, the basic method could be extended if neccesary to include spline curve and exponential characteristics.

The structure of the data format for a particular component algorithm is based upon the most commonly available published performance data. Where the method of data presentation is markedly different for a particular manufacturer the data can usually be interpreted into a more compatible and usable form.

Whichever approach to component algorithms is used, whether fundamental or empirical, a complete component model comprises four parts; an initialisation subroutine, a data record, an executive subroutine and a results subroutine. An outline of these component routines and details of the stucture of the data files are given in chapter 5.

3.4 Limitations of Component Models

The primary aim in developing the component models for this study has been to have a suitable component library available for use in the development of the methodology for the component based simulation of HVAC systems. The fundamental models have been developed to give similar characteristics to those expected of a more rigorous analysis, although unless otherwise indicated they have not been published or validated.

One of the difficulties inherent in digitising published performance curves is that there is usually no indication of the error term in

the manufacturers' original polynomial representation of test data. Further, these errors will be magnified if an assumption is taken, based upon limited data, that a family of curves exists for components in the same range. A more reliable approach would be to formulate a polynomial representation of the actual test data, if this could be made generally available. Since a polynomial curve is only applicable within the range of the original data it is important to limit the range over which the polynomial is used by placing suitable upper and lower bounds on the variables.

In a steady state procedure the approach to simulation of controller action must concentrate upon the open loop 'management control' under the assumption that the control scheme ultimately adopted would give an adequate transient response. A good control system using proportional and integral action would have minimal offset of the controlled variable from the set point value, hence over a steady state or quasi-steady state time interval the control action will be essentialy proportional. In contrast to this the most difficult form of control to simulate is on/off control where the transient response of the components has a major effect on its long term steady state performance. There are ways of taking this into account in a steady state model, such as developing a component model which uses the ratio of 'on' time to 'off' time over the steady state time period to factorise the component consumption or output. If a component model were to be based upon experimental performance data then the transient effect of on/off operation could be taken into account in developing the performance test procedure.

Because of the difficulties of simulating on/off control, and because it is rarely used in large scale HVAC systems, this problem has been excluded from the development work reported in this thesis. It is recognised however that due account would need to be taken of the effects of this, especially when trying to simulate the performance of systems at very low loads where even well designed systems may be subject to control cycling.

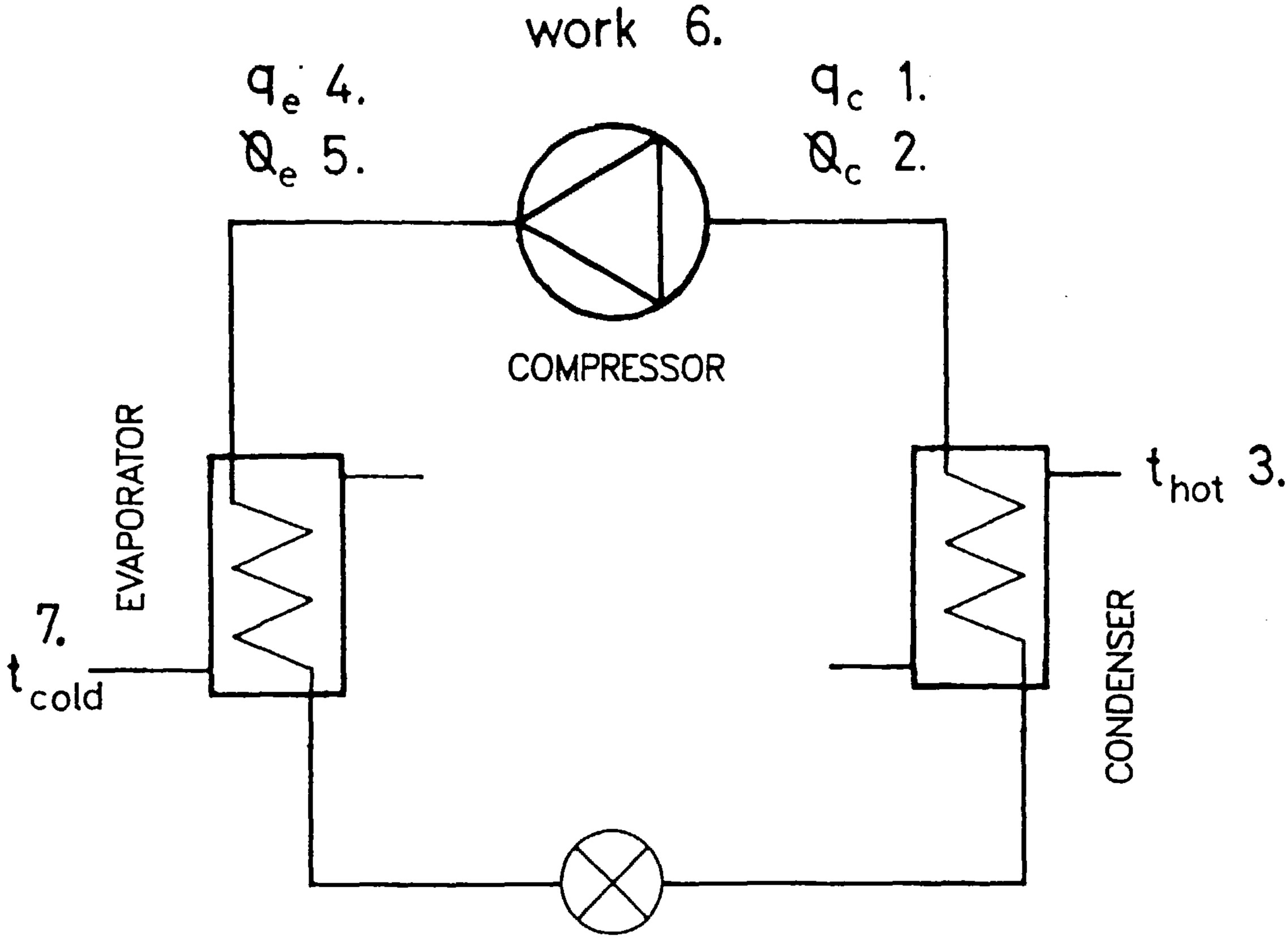
Chapter 4. SOLUTION OF NETWORK EQUATIONS

In order to develop a technique which will solve all the system equations simultaneously the problem is formulated in such a way as to make use of one of several available 'optimisation' algorithms. The methodology is much the same for all of these algorithms, the requirement being to minimise an objective function formed from the residuals between the true operating point and an initial, estimated operating point. These minimisation algorithms start from an initial estimate of the solution and generate a sequence of points designed to converge to a minimum. The differences between the methods are primarily concerned with the generation of these points and can be categorised into two groups, direct search methods and derivative methods.

Unfortunately there is no general optimisation algorithm which is applicable to all problems. The performance of a particular algorithm is dependant upon the characteristics of the objective function to which it is applied. The application of several optimisation algorithms to the simple HEATPUMP system shown in figure (4.1) has been studied to determine which type of algorithm may be best suited to the problems presented by HVAC systems (Murray 1982). A full discussion of the priciples and application of optimisation is given in Gill et al, along with details of most of the commonly available algorithms (Gill 1981).

4.1 Direct Search Methods.

Direct search methods compare values of the objective function at succesive points to determine the direction of a successful move to a lower point. These methods have the advantage that they do not require calculation of the derivatives of the function and can be effective where the function has discontinuities in the derivatives.



Evaporator $\theta_e = T_{cold} - Q_e/(EFF_e * UA_e)$

Condenser $\theta_c = T_{hot} + Q_c/(EFF_c * UA_c)$

Compressor:

refigeration $Q_e = \int_1(\theta_e, \theta_c)$

work input $W = \int_2(\theta_e, \theta_c)$

heat rejection $Q_c = Q_e + W$

EFF is the effectiveness of the coil

UA is the overall heat transfer co-efficient (W/°C)

functions $\int_1$ and $\int_2$ are 2nd order polynomials obtained by curve fitting manufacturer's performance data.

θ_e is the evaporating temperature

θ_c is the condensing temperature

t_{hot} and t_{cold} are fluid temperatures

figure 4.1 Simulation Network for HEATPUMP System

4.1.1 Multi-variable Search

Several of the direct search methods assume that the minimum value of a function is known to lie in some interval within which the function is unimodal (ie has only one minimum). The object of each method is to reduce this 'interval of uncertainty' to within acceptable limits. The procedures depend upon placing successive observations within the interval of uncertainty such that the interval is reduced according to the relative values of the objective function at each point.

The 'golden section' search technique, which is a variant of the Fibonacci search (Gill 1981), was applied to the multi-variable problem of determining the operating point for the HEATPUMP system in figure (4.1). The objective function, the sum of all the residual equations, is minimised with repect to each variable in turn, substituting the 'optimal' value for each variable in the successive searches. Thus the search takes place parallel to each co-ordinate direction, ignoring any interaction between the variables, and terminates when the objective function can no longer be reduced. The method works well provided that the initial estimate for the arc-variables is good: with even slight perturbations in the initial estimates the method can converge to false optima.

Variations on the method have been proposed which take account of the variable interaction but the method is generally slow to converge to the operating point and gives no guarantee as to the nature of the minimum obtained.

4.1.2 Simplex Method

A regular simplex in n-dimensional space consists of n+1 mutually equidistant points: in 2-dimensions the simplex is an equilateral triangle,in 3-dimensional space it is a tetrahedron ,etc. The basic simplex method generates the n+1 vertices starting at the initial guess to the solution, examines the objective function values at each vertex and reflects the vertex with the highest value through the centroid of the simplex to form a new point. If the objective function at this new vertex has a lower objective function value than

the previous one, it forms a new simplex and the process is repeated. If the value at the new vertex is higher than the original, then the original vertex is retained and the next highest valued vertex reflected. When no further progress can be made the sides of the simplex are reduced and the process repeated until the required accuracy in the solution is obtained.

A simplex algorithm, available as a NAG routine (NAG 1), which uses an irregular simplex to improve the rate of convergence to a minimum was applied to the HEATPUMP problem in figure (4.1). The method was slow to converge, requiring many function evaluations and, although it was less dependent upon the accuracy of the initial estimate than the multi-variate algorithm, it occasionally failed to converge to the minimum with sufficient accuracy. This is due to excessive rounding and truncation errors in comparing the objective value at each vertex when the sides of the simplex have been reduced. The method gives no indication as to the nature of the minimum found and although it works well with discontinuous, non-differentiable functions, the number of function evaluations required will increase exponentially as the number of system variables increases.

4.2 Derivative Methods

Derivative methods use first and possibly second derivatives to form a local approximation to the objective function in order to select a direction for a succesful move to a lower point. A comparison of the performance of several available derivative optimisation procedures, applied to the HEATPUMP system in figure (4.1), is summarised in table (4.1).

Most of these optimisation algorithms can be expressed in matrix and vector form using the following nomenclature:-

$\underline{x}$ is the vector of variables x_i in the n - dimensional space representing a particular point.

$\underline{f}(\underline{x})$ is the vector of residual equations $f_i(\underline{x})$ evaluated at the point $\underline{x}$.

	NR-Crout		NR-Gauss		Quasi-Newton		Least Squares		GRG-2	
t_{cold} / t_{hot}	time	nf	time	nf	time	nf	time	nf	time	nf
-10/35	.946	18	.381	37	4.213	742	1.013	51	3.934	65
5/35	.935	18	.115	11	1.604	250	.924	40	3.333	63
25/35	.935	18	.154	15	.982	654	.994	43	5.216	100
-10/45	.932	18	1.31	126	3.104	512	1.024	52	3.243	61
5/45	1.000	18	.336	31	1.568	261	.952	42	3.4	63
25/45	.968	18	.857	82	1.962	345	.991	45	3.769	72
-10/55	.926	18	*	*	6.368	1069	1.243	59	4.208	83
5/55	.991	18	.669	64	2.637	422	.949	40	3.321	63
25/5	.999	18	*	*	2.43	362	1.201	54	5.078	95
Average	.959	18	.546	52	2.763	513	1.032	47	3.945	73

'time' is the processor time taken relative to the NR-Crout solution at $t_{cold} = 5°C$, $t_{hot} = 45°C$.

'nf' is the number of residual function evaluations required including those for the numerical estimation of partial derivatives.

'*' indicates that the algorithm failed to find a solution at this point.

Table (4.1) Comparative Performance of Optimisation Algorithms for solution of HEATPUMP equations.

$F(\underline{x})$ is the objective function evaluated at point $\underline{x}$. This is u the Euclidean norm of the vector $\underline{f}(\underline{x})$:

$$F(\underline{x}) = \{ \sum_{i=1}^{n} f_i(\underline{x})^2 \}^{1/2}$$

$\underline{j}$ is the Jacobian gradient vector of first partial differentials

$\partial F(\underline{x}) / \partial x_j \quad j=1,n$

$\underline{J}$ is the Jacobian gradient matrix of first partial differentials

$\partial f_i(\underline{x}) / \partial x_j \quad i,j=1,n$

$\underline{H}$ is the Hessian curvature matrix of second partial differentials

$\partial^2 F(\underline{x}) / \partial x_i \partial x_j \quad i,j=1,n$

the suffix 'k' refers to the value of a parameter evaluated at the k^{th} iteration.

4.2.1. Newton - Raphson Algorithm.

A generalised Newton-Raphson solution procedure was developed which, starting from an initial point, uses a linear approximation to the function to make successive corrections to the system variables to obtain a minimum of $F(\underline{x})$. The iterative procedure is :-

$\underline{x}_{k+1} = \underline{x}_k - \underline{p}_k$

where $\underline{p}_k = \underline{J}_k^{-1} * \underline{f}(\underline{x}_k)$.

The solution of the equation $\underline{J}_k * \underline{p}_k = \underline{f}(\underline{x}_k)$ for the direction vector $\underline{p}_k$ can be obtained using standard matrix methods of either Gaussian elimination with partial pivoting, or a NAG routine using Crout's decomposition (NAG 2). The formulation of the Newton Raphson solution algorithm and both the Gauss and Crout procedures are described in Appendix B.

The Newton Raphson algorithm has been used extensively in the development of the simulation procedure, because it is a simple, first order method which will work well with numerical estimates of the partial derivatives. However since the method is based upon a first order approximation to the function at the solution, it requires a good initial estimate of the solution point to ensure convergence to the minimum. When it works, it works well, usually taking approximately n+1 iterations to solve an n - dimensional

problem, an almost quadratic convergence rate. Because the variables are unbounded the algorithm can generate infeasible points which cause the solution to fail. There are two prime modes of failure, ill-conditioning and derivation of a singular Jacobian. Ill-conditioning may be caused by the calculation of bogus residuals resulting from infeasible variable values (eg: negative mass flow rates) or by numerical 'hunting' across the throttling range of a controller. A singular matrix is one which has no inverse and indicates that the system equations are badly defined, or that the system equations are not independent.

Because the Newton - Raphson procedure is an unconstrained optimisation algorithm it is difficult to decipher the reasons for failure since there is no indication of which equations remain unsolved or which variables have infeasible values.

4.2.2. Quasi-Newton Algorithm.

Quasi-Newton methods of optimisation are a variation of the second-order Newton methods which seek to retain the rapid, almost quadratic termination whilst avoiding the direct computation of the Hessian matrix of second partial derivatives. A second order 'Conjugate Direction' algorithm was tried but was subject to repeated failure due to the inaccuracies inherent in estimating the Hessian numerically. The quasi-Newton method makes an estimate of the curvature of $F(\underline{x})$ by approximating the Hessian matrix from first order information only. A linear search of the form :-

$$\underline{x}_{k+1} = \underline{x}_k - \alpha_k * \underline{p}_k$$

in the direction $\underline{p}_k$ with repect to α, is applied, not necessarily exact, to ensure that $F(\underline{x})_{k+1}$ is a better approximation to the solution than $F(\underline{x})_k$. There are many proposals for obtaining the search direction $\underline{p}_k$ (Gill 1981) but the NAG routine used employs a Cholesky factorisation of the Hessian into a lower triangular matrix $\underline{L}$ and diagonal matrix $\underline{D}$ to determine the search direction $\underline{p}_k$ from :-

$$\underline{L}_k * \underline{D}_k * \underline{L}_k^T * \underline{p}_k = -\underline{j}_k$$

where $\underline{L}^T$ denotes the transpose of $\underline{L}$. At each iteration a

recurrence relationship for the approximation to the Hessian is applied to the Cholesky factorisation and a finite difference approximation to the Jacobian vector $\underline{j}_k$ is obtained.

The NAG quasi-Newton routine (NAG 3) permits bounds to be declared for any or all of the variables which has the effect of improving the convergence of the algorithm by reducing the size of the feasible region. If a variable hits a bound during a linear search its value is fixed at that bound and the search continues in the remaining variables, however the method can converge to false minima with one or more variables fixed at their bounds.

The application of the quasi-Newton algorithm to the heat-pump system of figure (4.1) required a non-dimensionalised form of the residual equations. This is to avoid bias in the computation of the Jacobian vector $\underline{j}$ and the objective function $F(\underline{x})$ when the individual residuals $f_i(\underline{x})$ are poorly scaled (ie: vary widely in their numerical values). The non-dimensionalised form of equation (3.5) would be :-

$$F^1 = 0 = w^1 \, (t^1_{in} - t^1_{out}) \; / \; w_{min} \, \varepsilon \, (t^1_{in} - t^2_{in}) - 1 \qquad (4.1)$$

This form of residual equations is however unsuitable for methods which use the Jacobian matrix $\underline{J}$ since the division of the terms in equation (3.1) to form equation (4.1) introduces excessive rounding and truncation error in the calculation of the individual terms in $\underline{J}$. The problems of scaling are discussed further in section (4.3).

The performance of the quasi-Newton algorithm in solving the heat-pump network, summarised in table (4.1) and detailed in an earlier report (Murray 1982), shows that the algorithm is succeptible to poor scaling and shows a poor ultimate rate of convergence, possibly due to problems of approximating the Hessian from a numerical estimate of derivatives which tend to zero at the solution.

4.2.3. Non-linear Least Squares Algorithm.

Least squares algorithms are applicable to problems which can be expressed in the form :-

$$F(\underline{x}) = {}^1/_2 \sum_{i=1}^{n} f_i(\underline{x})^2 \ \}$$

The Jacobian and Hessian matices of these problems have a special structure in that the Hessian may be expressed as a special combination of first and second order information :-

$$\underline{H}(\underline{x}) = 2 * \underline{J}^T * \underline{J} + 2 * \sum_{i=1}^{n} \{ f_i(\underline{x}) * \underline{h}_i \ \}$$

where $\underline{h}_i$ is the Hessian of $f_i(\underline{x})$. In problems where the residuals are 'small' the first order terms are assumed to dominate and the Hessian of $F(\underline{x})$ is approximated as :-

$$\underline{H}(\underline{x}) = 2 * \underline{J}^T * \underline{J}$$

This gives a reasonable estimate of the Hessian, and hence the curvature of the objective $F(\underline{x})$ near the optimum, without explicit calulation of the second partial derivatives.

The NAG least squares algorithm (NAG 4) performs a line search in α ($0 \leq \alpha \leq 1$), similar to that for the quasi-Newton method :-

$$\underline{x}_{k+1} = \underline{x}_k - \alpha_k * \underline{p}_k$$

to find an approximate minimum of $F(\underline{x}_k + \alpha_k \underline{p}_k)$. The direction vector $\underline{p}_k$ depends upon the previous reduction in the sum of squares: if steady progress has been made then $\underline{p}_k$ is the Gauss-Newton first order approximation to the Newton direction, otherwise further function evaluations are made to form a second order approximation to the Newton direction. Generally where $f_i(\underline{x}) = 0$ at the solution the Gauss-Newton approximation approaches the Newton direction and the least squares algorithm converges quadratically.

In the application of the least squares algorithm to the heat-pump network the algorithm converged to machine accuracy within a few iterations although this was at the expense of a large number of function evaluations compared with the Newton-Raphson/Crout algorithm of section (4.2.2).

4.2.4. Generalised Reduced Gradient Algorithm.

A version of the generalised reduced gradient algorithm, GRG2 (Lasdon 1978), has been applied to the solution of HVAC systems by Silverman (Silverman 1981). GRG2 solves non-linear problems expressed in the form :-

minimise $g_k(\underline{x})$

subject to $lb(n+i) \leq g_i(\underline{x}) \leq ub(n+i)$ $\quad i=1,m \; i \neq k$

and $lb(i) \leq x_i \leq ub(i)$ $\quad i=1,n$

where $\underline{g}$ is a vector of m constraint functions

$\underline{x}$ is a vector of n system variables

and $\underline{lb}$ and $\underline{ub}$ are vectors of lower and upper bounds respectively.

The algorithm can be used to minimise an objective function subject to equality constraints, or range constraints, and simple bounds on the variables. In the context of the simulation of HVAC systems the residual equations described in section (3.2) are treated as equality constraints and the algorithm minimises a 'phase 1' objective, formed as the sum of all the constraint violations, to find a feasible operating point. Using the terminology defined in section (4.2) the problem can be expressed as :-

minimise $F(\underline{x}) = \sum_{i=1}^{n} f_i(\underline{x})$ $\quad i=1,n$

subject to $f_i(\underline{x}) = 0$ $\quad i=1,n$

and $lb(i) \leq x_i \leq ub(i)$ $\quad i=1,n$

Reduced gradient methods seek to use similar ideas to linearly constrained optimisation methods by defining a 'working' or 'active' set of constraints to develop a reduced problem in the sub-space of points, $\tilde{\underline{x}}$, which will continue to satisfy the active constraints. A feasible search direction is then found and a one dimensional search in α is used to find an approximate minimum of $F(\tilde{\underline{x}} + \alpha \underline{p})$

In a linearly constrained problem the process of maintaining feasibility can be accounted for in the construction of a feasible search direction. However in the non-linear case there is no 'a priori' procedure for ensuring a feasible search direction $\underline{p}$, since it is difficult to ensure that a search direction follows a curved constraint boundary. The GRG2 algorithm uses an iterative 'Newton' method to determine a feasible search direction.

The GRG2 algorithm may be used to perform true optimisation of an objective function with respect to a extended set of system variables. The problem is then of the form :-

$$\begin{aligned} &\text{minimise} && g(\underline{x},\underline{y}) && \\ &\text{subject to} && f_i(\underline{x}) = 0 && i=1,n \\ & && lb(i) \leq x_i \leq ub(i) && i=1,n \\ &\text{and} && lb(n+i) \leq y_i \leq ub(n+i) && i=1,m \end{aligned}$$

where $g(\underline{x},\underline{y})$ is an objective function comprising, say, the total energy consumption of the HVAC system and $\underline{y}$ is a set of 'm' of the exogenous variables. The algorithm performs a phase 1 minimisation to find a feasible operating point and then procedes to minimise the energy consumption, $g(\underline{x},\underline{y})$, with respect to exogenous variables such as controller set points etc.

A copy of the GRG2 program was obtained and implemented on the Honeywell MULTICS computer at Loughborough. The algorithm is comprehensive and robust: when it fails to find a feasible operating point for a system there is usually sufficient information about the progress of the optimisation to be able to diagnose the likely cause of failure. A general outline of the GRG2 algorithm is given in appendix C and examples of its use are given in chapter 6.

Although the GRG2 algorithm is generally more robust than the Newton-Raphson method this is at the expense of an approximately fourfold increase in the computational effort required to find a solution (table 4.1). The algorithm does not converge rapidly to a feasible point, sometimes taking many iterations to satisfy a single remaining

constraint. This is because generalised reduced gradient algorithms tend to approach a solution tangentially to a constraint and GRG2 in particular can generate infeasible points even though the variables may be bounded.

4.3 Numerical Prolems

There are numerical problems inherent in the formulation of the optimisation problem for typical HVAC systems which cause the algorithm either to fail to find a feasible operating point for the system or to show such poor convergence towards a feasible point as to be impractical for general application. These numerical difficulties include scaling, estimation of numerical derivatives and generation of infeasible points.

4.3.1 Scaling and Multi-dimensional Geometry.

'Scaling' is blamed for many numerical problems without any clear understanding of the way in which scaling of variables, constraints or the objective function can affect the performance of optimisation algorithms. In most physical problems the ideas of Euclidean geometry, such as the concept of area or length, cannot be expanded directly to n-dimensional non-Euclidean problems. The definition of distance in Euclidean geometry is given by Pythagoras as :-

$$r = \{ x^2 + y^2 \}^{1/2}$$

but this cannot be applied to non-Euclidean space since 'square degrees centigrade' cannot be added to 'square minutes'. This problem is usually overcome by plotting the functions on a graph to scales which transform the problem into one to which the rules of Euclidean geometry can be applied.

Most derivative optimisation problems are based upon forming an estimate of the function at or near the minimum where the contours of the objective function are almost ellipsoidal (or the n^{th} dimensional analogue). Scaling techniques seek to transform the original problem into a function which has this ellipsoidal form.

Variables should be scaled so that a unit change in one variable at the minimum produces the same change in the objective function as that produced by a unit change in another variable. This ideal scaling is frequently impossible with realistic problems but the aim should be to ensure that the variables are all of the same magnitude in the region of interest. Optimisation algorithms require some definition of 'big' and 'small' in order to assess convergence criteria, clearly if the variables remain unscaled, for example power in Watts (500 - 15000) and fluid flow rates in kg/s (0.5 - 10.0) there would be bias in the computation of the Euclidean norm of the residuals $\underline{f}(\underline{x})$ for the objective function $F(\underline{x})$.

Scaling the variables also affects the computation of derivatives since if the variables are badly scaled it is difficult to select a set of differencing intervals which will produce a reasonable change in the objective $F(\underline{x})$ with respect to each variable. This is also related to the scaling of the residual constraints with repect to each other, since if the residuals differ widely in their numerical values so will the partial differentials, giving a distorted estimate of the surface of the objective function.

Initialy a scaling technique was used whereby each variable was transformed by a linear relationship :-

$x_i = D * y_i$

where y_i are the transformed variables and D is a constant diagonal matrix whose elements D_{ij}, i=j are typical values of x_i, in this case the initial estimate of x_i. However this transformation has the disadvantage that some of the numerical accuracy may be lost by truncation and cancellation errors, especialy in the computation of numerical derivatives, and if the initial estimate is not a 'good' one, the advantages of scaling may be lost nearer the solution point.

Since for most of the variables in a HVAC system a reasonable range of values is known, the variables may be scaled such that the transformed variables are in the range -1 to +1 for all values of x_i within upper and lower bounds UB_i and LB_i :-

$$y_i = \frac{2 * x_i}{UB_i - LB_i} - \frac{LB_i + UB_i}{UB_i - LB_i}$$

This transformation gives full precision representation of the scaled variables (Gill 1981 p274) and has been used in the application of the GRG2 algorithm.

Although this transformation works well where the bounds on a variable are reasonable, the solution algorithm can perform poorly when either of the bounds are merely a crude limit, possibly wrong by several orders of magnitude. In an analysis of several HVAC networks, notably containing coils, the performance of the GRG2 algorithm could be improved by reducing the upper bound on the water mass flowrate from a crude estimate of 20 kg/s empirically to 4 kg/s. However care must be taken in specifying bounds for variables since reasonable bounds may be related to the value of the exogenous constants specified during system definition, for example the upper bound on the water mass flowrate for a coil is related to the number of circuits in the coil (Appendix A-3).

There is no generalised transformation suitable for scaling the non-linear residuals. The normalised form of the residuals used in the quasi-Newton algorithm (section 4.2.2) is a form of scaling. Whilst this produces adequate scaling for the derivation of the Jacobian vector $\underline{j}$, considerable cancellation and truncation errors are introduced into the computation of the individual residuals $f_i(\underline{x})$ which cause numerical problems in the derivation of the Jacobian matrix J. The GRG2 algorithm uses the numerical values of the residual constraints as initial estimates of the slack variables (see Appendix C) and hence requires all the residuals to be of a similar order of magnitude (Lasdon 1982).

Some of the residual equations in Appendix A have been scaled empirically by looking at their values at a typical initial point and adding a simple linear transformation to the executive routine to ensure that the numerical value of the residual at a typical starting point is within the range -100 to +100 in line with recommendations by Lasdon. The effects of different scaling factors on the performance of the GRG2 algorithm can be checked during component model development. Generally this has resulted in a multiplicative factor of 100 for mass balances and 10 for enthalpy balances (see Appendix A), however no firm guidlines can be developed for other residuals.

4.3.2. Numerical Differentiation.

It is unlikely that all the component residuals for HVAC components will be algebraicly differentiable in a staightforward manner. The first derivatives of certain functions could be determined explicitly but for the more complex component residuals recourse must be made to finite difference approximations to the derivatives. A forward difference approximation to the Jacobian matrix may be formed using the formula :

$$\partial f_i(\underline{x})/\partial x_i \simeq \{f_i(\underline{x} + h_i) - f_i(\underline{x})\} / h_i \qquad i=1,n$$

which requires 'n' extra function evaluations for every estimate of the Jacobian. The difficulty is in selecting suitable differencing intervals h_i : if h_i is too large the estimate will be inaccurate, whilst if h_i is too small the residual functions will have so many figures in common that the cancellation error in computing the differences will dominate (Gill 1981 p339). There is a NAG service routine (E04HBF) which may be used to select intervals which balance the truncation and cancellation errors for a particular function, given a typical point. These 'optimum' intervals need only be determined once, usually at the initial point, since there is little gain in accuracy in recomputing the intervals during the optimisation search. This routine was used to obtain intervals for the least-squares algorithm (section 4.2.3) which determines the finite

difference approximations internally.

The difference intervals used to calculate the numerical derivatives for the Newton-Raphson routine are based upon the NAG auxilliary routine E04FCX :-

$h_i = \varepsilon^{1/2} * (1.0 + x_i)$

where ε is the machine constant for the computer such that $1.0 + \varepsilon > 1.0$ to machine accuracy. This gives difference intervals which are simple to calculate and which alter as the search progresses. The approximation to $J(\underline{x})$ thus formed is good, provided that $\underline{x}$ is well scaled. If $\underline{x}$ is badly scaled then, as x_i tends to zero so h_i tends to $\varepsilon^{1/2}$, hence for 'small' values of x_i (such as mass flowrate in kg/s, say 0.2kg/s) h_i is dominated by ε, whereas for 'large' values of x_i (such as mass flowrate in g/s, say 200g/s) then h_i reflects the value of x_i more closely.

Forward difference approximations to $J(\underline{x})$ or to $\underline{j}(\underline{x})$ tend to become unreliable near the minimum even for a well scaled problem with good differencing intervals because the gradient of the function tends to zero at the minimum. In this case a better approximation to the Jacobian could be obtained using central differences and a larger interval $\tilde{h}_i \simeq h_i^{2/3}$ (Gill 1978 p131). However this requires 2n function evaluations for each Jacobian estimate, so the forward difference method could be used as far as possible, switching to central differences only when the errors in the estimation of the Jacobian become too large. This does not cause problems with the Newton-Raphson algorithms because, due to the quadratic convergence of the method, the gradient Jacobian is not evaluated 'near' the minimum.

A further problem in using forward difference approximations to the Jacobian occurs when the residual functions have discontinuities, such as the equations for a cooling coil at the transition from dry to wet cooling (Appendix A-3) or as in the characteristics of some controllers (Appendix A-12). A study of the partial differentials for a cooling coil over the transition between wet and dry cooling

revealed that problems with numerical differences will occur only when the differencing intervals are extremely small. If discontinuous functions cause problems in the development of component algorithms it would be possible to form both backward and forward finite difference approximations either side of the discontinuity.

4.3.3. Feasibility of Iterative Points.

Generation of infeasible points, for which the objective or residual functions are undefined within the optimisation algorithms can cause numerical problems in the component executive algorithms. The most common restriction on feasibility is non-negative bounds on physical variables such as mass flow rates, since negative mass flowrates are meaningless in the thermodynamic sense. Another restriction on feasibility with certain components is the range over which a polynomial curve-fit can be used. If the polynomial is evaluated for values of the variables outside of the range of the original data then the result is undefined, frequently producing spurious values for the residuals.

The application of both the unconstrained Newton-Raphson algorithm and the bounds constrained GRG algorithm result in the generation of infeasible points (see Appendix C for an explanation of the GRG procedure). To avoid numerical problems with the executive routines variables are reset to their bounds and the residual functions are evaluated at this modified point. Although this is mathematically undefined in that it alters the computed search direction $\underline{p}_k$, it has been found to work in practice. An alternative strategy for the GRG2 algorithm is to set the objective function to some large arbitrary value which will cause the Newton search sub-problem to terminate at the latest feasible point and a new search direction to be evaluated (Lasdon 1982). In the application of GRG2 to HVAC networks the former strategy has been found to be most effective.

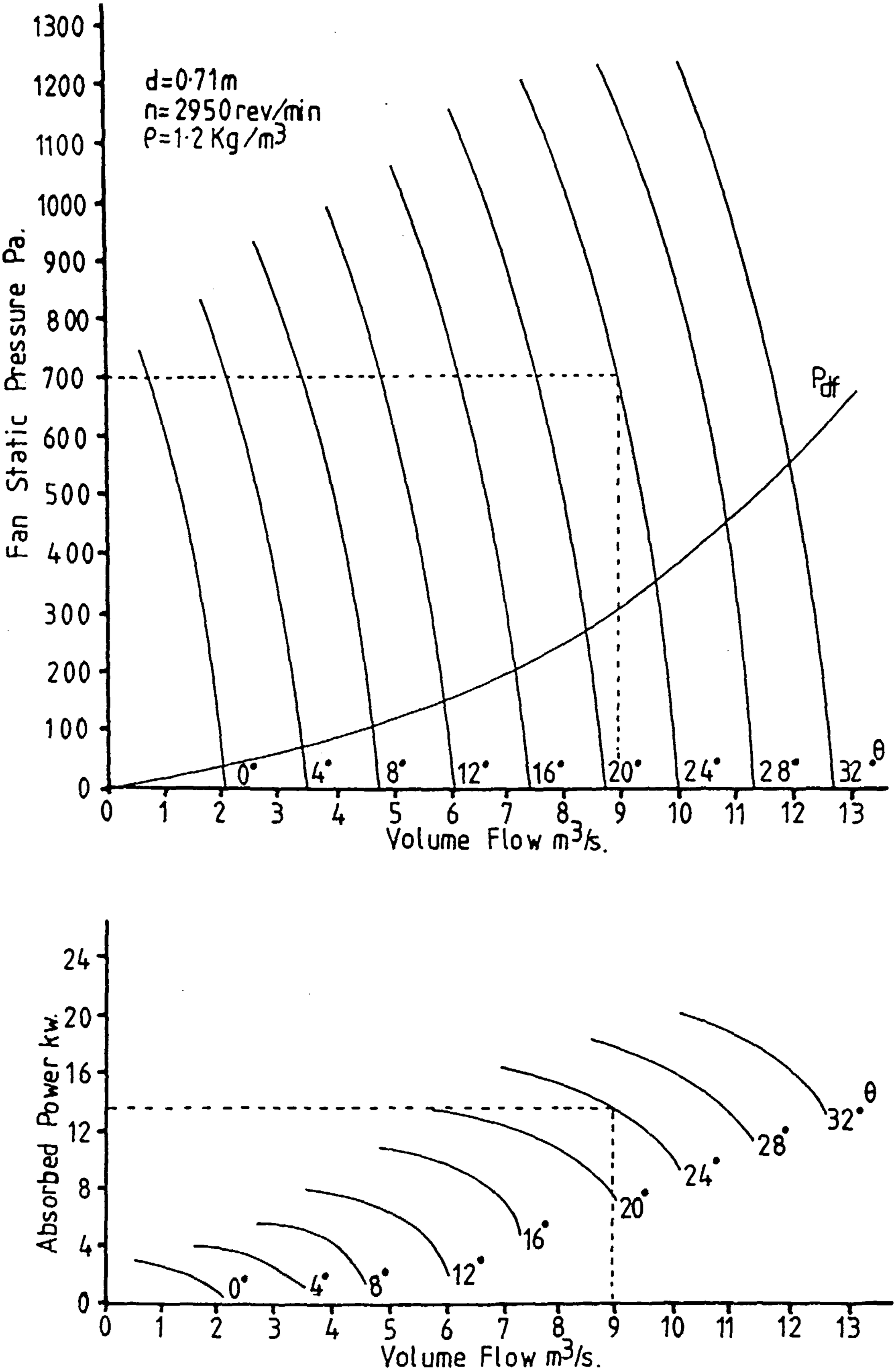


figure 4.2 Fan Performance Curves (Wright 1984)

The use of a polynomial in one variable may be restricted to the feasible region by placing bounds on the variables, but for a two dimensional curve-fit the bounds on the polynomial may be far more complex. A typical performance envelope for a fan is shown in figure (4.2) where the feasible region is defined by an upper and lower bound on the speed and two non-linear constraints which could be expressed as polynomials in total pressure and flowrate.

4.4. An 'ideal' Solution Algorithm

Each of the algorithms which have been applied to the problem of solving the residual equations for HVAC systems are designed for general application to problems which are cast in a particular form. Although the individual algorithms may have specific attributes which are advantageous in this application, no one algorithm can be considered ideal for these problems.

Each iteration in the optimisation algorithms generally consists of two sub-problems, computation of a suitable search direction $\underline{p}_k$ and derivation of a suitable step length α.

The direction of search is usually deterministic, ie: independent of the particular value of the objective function, being fully defined by the particular algorithm applied. The algorithm must be chosen to suit the particular type of problem at hand, such as linear or non-linear functions, or whether first and possibly second derivatives are available. Specific algorithms can be adopted to take advantage of, or to overcome, specific characteristics of the objective such as a sum of terms formulation or sparse Jacobian or Hessian matrices.

The selection of a suitable step length α_k is highly dependent upon the nature of the objective function at the point k and is usually an adaptive algorithm which can take account of the progress of the search or special features of the objective to ensure that the step $\alpha_k \underline{p}_k$ leads to an improvement in the objective (ie: a lower point). Since the step length must also be chosen to ensure feasibility in constrained problems, these algorithms are considerably more complex

and take longer to execute than their unconstrained equivalents. Algorithms for linearly constrained problems are widely available but the extension of these methods to cope with non-linear constraints is a non-trivial task, often necessitating a completely different formulation of the basic algorithm.

The general optimisation problem of solving a set of residual functions to find a feasible operating point of a HVAC network may be stated as :-

$$\text{minimise} \qquad F(\underline{x}) = \sum_{i=1}^{n} f_i(\underline{x}) \quad \text{or} \quad \left(\sum_{i=1}^{n} \{f_i(\underline{x})^2\} \right)^{1/2}$$

subject to :

$$\text{inequality constraints} \qquad c_i(\underline{x}) \geq 0 \qquad i=1,m$$

$$\text{bounds} \qquad lb_i \leq x_i \leq ub_i \qquad i=1,n$$

The number of constraints will generally be either zero or small compared to the number of residual functions and system variables. In general the operating point of a system will be expected to lie well within the region defined by non-linear constraints so that there would be no active constraints at the solution. If a non-linear constraint were active at the solution then one or more of the components would be operating at the limit of the feasible region and the performance of the system might benefit from a reselection of components.

Since there is not a non-linearly constrained non-linear optimisation algorithm available which can take advantage of the 'sum of terms' formulation of the residual equations, the choice must be between sacrificing the special form of the objective function to include a few non-linear constraints which ensure the feasibility of the polynomials, or retaining the advantages of the sum of terms formulation, optimising subject to simple linear bounds on the variables and then checking whether the operating point is within the feasible region.

Chapter 5. PROGRAMMING AND SOFTWARE DEVELOPMENT

The component based simulation procedure outlined in the preceding chapters has been implemented as a suite of computer programs collectively referred to as 'SPATS' - Simulation of the Performance of Air-conditioning and Thermal Systems. The primary objectives of this study, ie: to develop a simulation methodology for use as a HVAC system design tool, have been used to produce a software framework for the development of both component models and the application of techniques of optimal search to the solution of HVAC system equations.

The initial development of the program was on a micro-computer (Intertec Superbrain, 8-bit) in FORTRAN, however the limit of the available memory (approximately 40 kBytes of FORTRAN routines) restricts the program to ten component algorithms. For further development work the suite of programs was transfered to a mainframe computer. The current implementation of SPATS is on a Honeywell MULTICS mainframe but the program could ultimately be implemented on one of the larger 16-bit micro-computers now available.

As far as is practicable the programs have been written in standard ansi77 Fortran (ANSI 1978) with the exception of an interface routine to the GRG2 program since this program is written in a variant of FORTRAN IV which is not directly compatible with FORTRAN77 (Lasdon 1982). In the following description of the major program segments commands appear in inverted commas and **'bold'** type, FORTRAN variable names appear **bold** and subroutine names are enclosed in brackets eg: ⟨option⟩.

All data input from the keyboard is checked against the variable type expected, ie: integer, real or character, and file management routines request confirmation before overwritting or deleting files.

Extensive use has been made of the 'parameter' statement to define variables which are used in array declarations. These parameters define the size of the problems which can be analysed using 'SPATS',

such as the maximum number of nodes in a network or the maximum number of variables associated with a component: a list of the main parameters is given in table (5.1). The routines which read or write data arrays to files, such as ⟨fwrite⟩ for component data records and ⟨filnet⟩ for network arrays, are written in a generalised form in terms of the parameters used to declare the size of each array. The size of a data file will thus vary when the parameters are changed, hence the need for subroutine ⟨newdat⟩ to up-date component data records automatically (section 5.2.9).

Data transfer between subroutines is almost entirely by means of common blocks, 'local' variables being confined to integer counters for 'do-loops' etc, and some local constants. Variables which are 'local' to a subroutine are given a unique address in computer memory even though variables of the same name may be declared in other routines. Clearly where the same variables are used temporarily in many subroutines, such as variables (x) and constants (c) in executive and results subroutines, considerable savings in memory requirements may be made by declaring these variables as 'common' even though they are used as local variables. Parameter lists are used for subroutines which perform generalised computational tasks, such as NAG routines and the Newton-Raphson solution routine ⟨crout⟩, since these routines are independent of the particular application to HVAC simulation.

5.1 'SPATS' as a Multi-User Development Tool

Several specific features of the MULTICS have been used to develop a multi-user version of SPATS, notably the ability to link program segments dynamically at run time. This has enabled individual users to develop component models or utility routines which may then be referenced by other users of SPATS through a system of 'search rules'. The location of the program ⟨spats⟩ in the MULTICS file store defines the working directory, which must contain all the relevant data files and to which any files created within SPATS are written.

maxvar	50	maximum No. of arcvariables in a network
maxnod	30	maximum No. of nodes in a network
maxexo	20	maximum No. of exogenous constants in a network
maxcon	200	maximum No. of component constants in a network
mvar	16	maximum No. of variables for a component
mex	9	maximum No. of exogenous constants for a component
mcon	30	maximum No. of constants for a component
mnp	5	maximum No. of polynomials for a component
mcoe	50	maximum No. of co-efficients for a polynomial
maxnet =	5	+ mvar + mex + mcon dimensional parameter for network index array net.

Table (5.1) Parameters used in SPATS

The overall structure of SPATS is given in figure (5.1) and the interrelationship between the subroutines is shown in figure (5.2): a list and a brief description of all the subroutines referenced by SPATS is given in Appendix D. The program totals some ten thousand lines of FORTRAN text, including comments but excluding NAG library routines, MULTICS operating system routines, the GRG2 routines and routines developed by other users (italics in figure 5.2).

5.2 Description of the Major Elements in SPATS

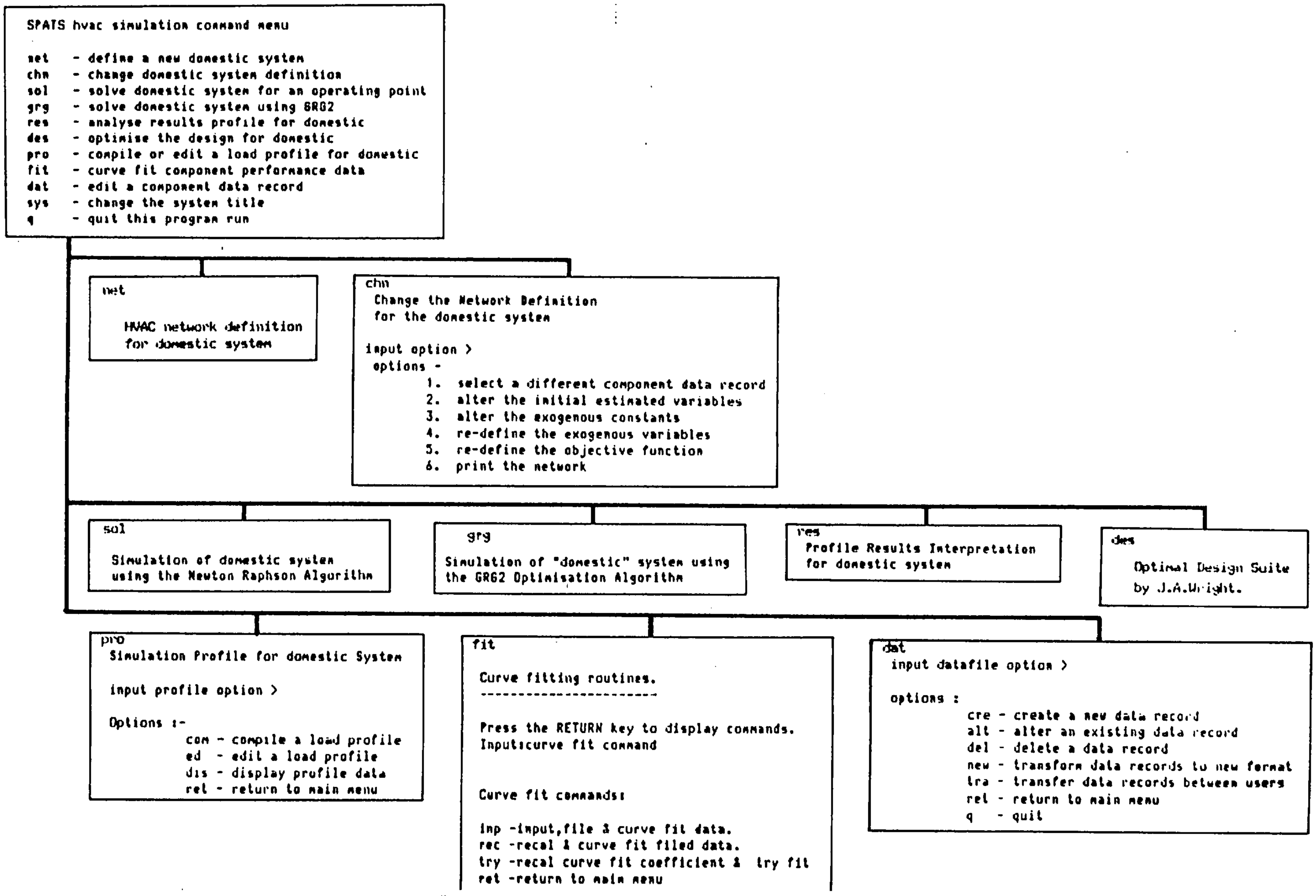
The SPATS suite of programs are interactive and are accessed by running the program ⟨spats⟩ (figures 5.1 and 5.2). Subroutine ⟨version⟩ sets the date and prompts input for a reference name and a system title which is used within the program in the command menus and to create various file names. In ⟨setnode⟩ the MULTICS operating system uses the search rules to locate all the component initialisation subroutines within the MULTICS file store and sets up the menu of available nodes. If a particular routine is not available a dummy routine is called to set a flag so that that component type cannot be referenced by any of the SPATS routines. Program control then passes to ⟨option⟩ which is an interface to all the other subroutines in the suite (figure 5.2).

Upon completion of an option command, or if a programmed error condition occurs, control normally reverts to ⟨option⟩. Programmed error conditions are those which may be forseen, such as read/write errors in files or when specific files cannot be found in the current directory.

5.2.1 Network Definition

The option command **'net'** calls the subroutine ⟨netwrk⟩ for the definition of a new HVAC system network. A simplified flow diagram for ⟨netwrk⟩ is shown in figure (5.3) and a flow diagram for subroutine ⟨iarray⟩ which inserts the component data into the network arrays is shown in figure (5.4). The program prompts the user through the network definition node by node: should a mistake be made, such as specifying the wrong data file or defining the

figure 5.1 Structure of the SPATS Suite of Programs



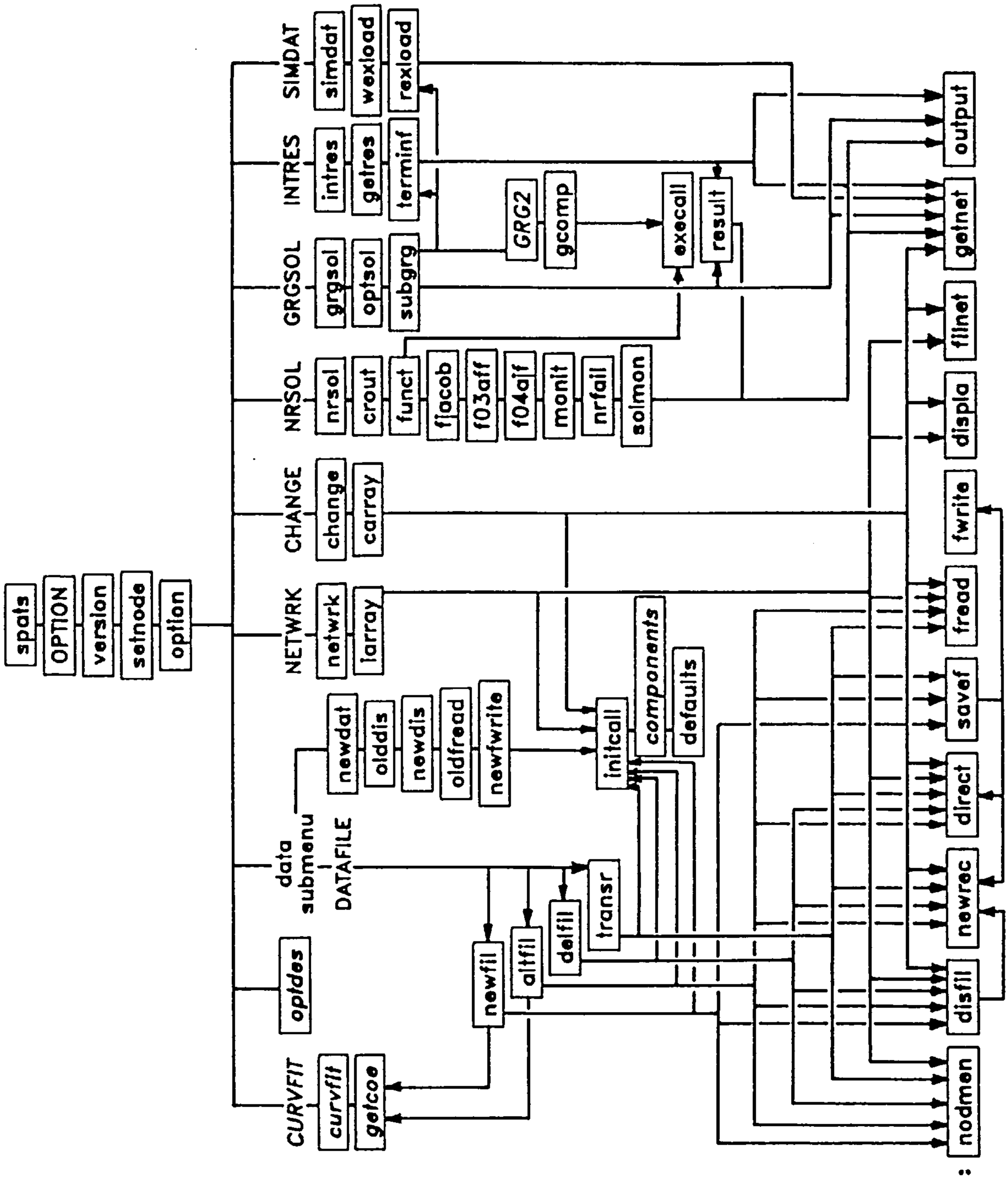


figure 5.2 Subroutine Structure of SPATS

exogenous constants or variables wrongly, these can be rectified by using the option command **'chn'** (figure 5.1).

In subroutine ⟨iarray⟩ (figure 5.4) the chosen data record for each node is read from the directory file and the data is assigned to the relevant network arrays. If the name of an exogenous constant, as defined in the component initialisation routine, begins with the four characters 'mult' then the set of variables associated with that node may be extended to include extra variables, such as multiple heat sources in a zone (Appendix A-10)

Each of the variables associated with a node must be identified by an arc-variable. If the arc-variable is newly defined in the system definition then the initial estimate, together with lower and upper bounds for that variable are assigned to the relevant network arrays. If the arcvariable number has been previously assigned at another node in the system definition, then a compatibility check is made between the two variables by comparing the first two characters of the component variable name with the first two characters of the system arc-variable name. If the characters do not match then the user is given the opportunity to override the compatibility check. Thus all air temperature names begin with the characters 'ta...' , all water flow rate names begin with 'mw...' etc., however the ability to override the check enables specific variable types, such as air temperature, to be assigned to general variables such as controller inputs.

If a particular node type has a direct consumption of energy this may be added to, subtracted from or excluded from the overall system objective function. This system objective function may be minimised with respect to one or more of the exogenous variables by one of the solution algorithms (section 5.2.4).

All the data necessary to completely define a system network is written to a file called '*.net' where '*' is the system title.

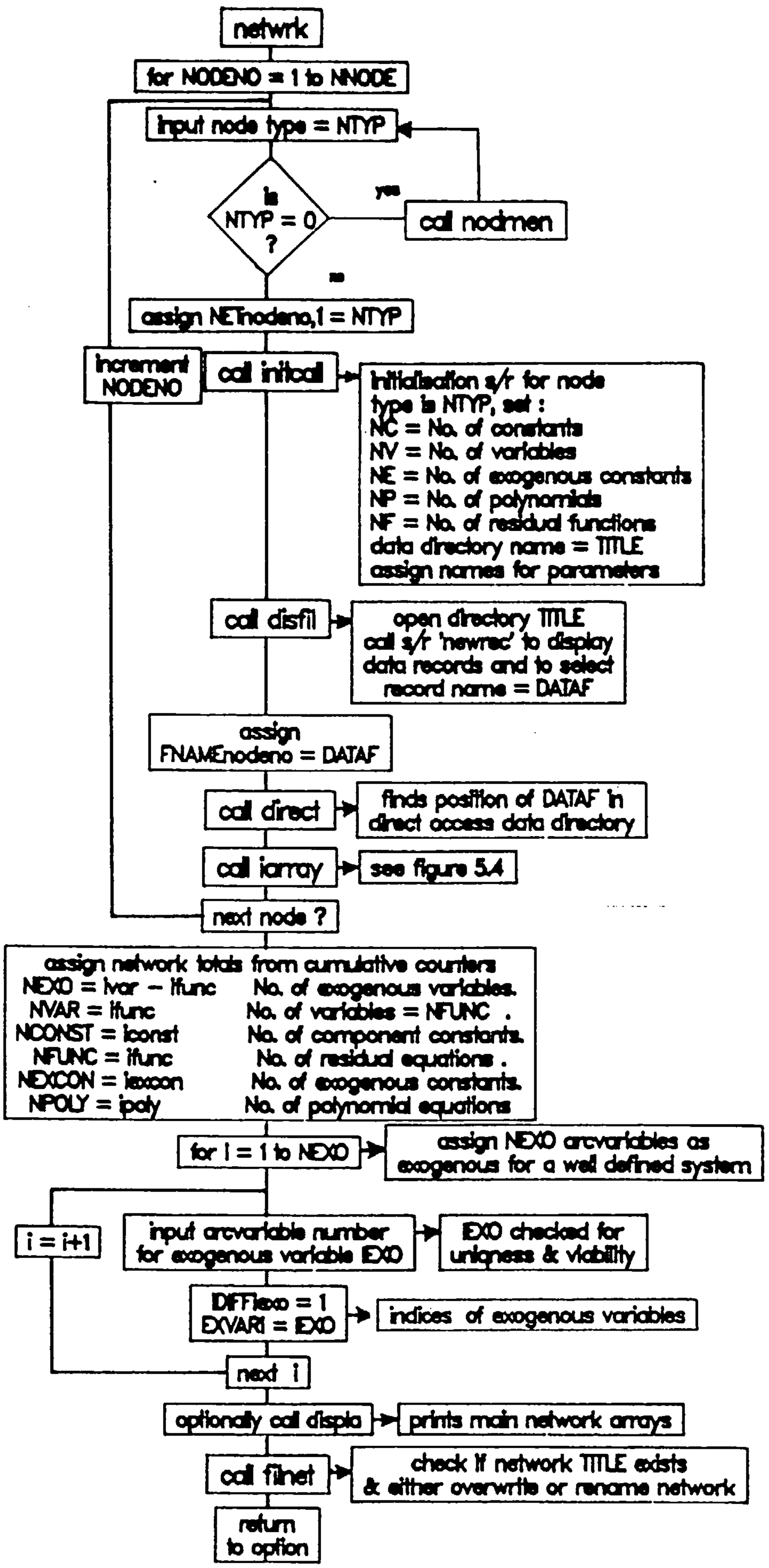


figure 5.3 Flow Diagram for 'netwrk' Subroutine

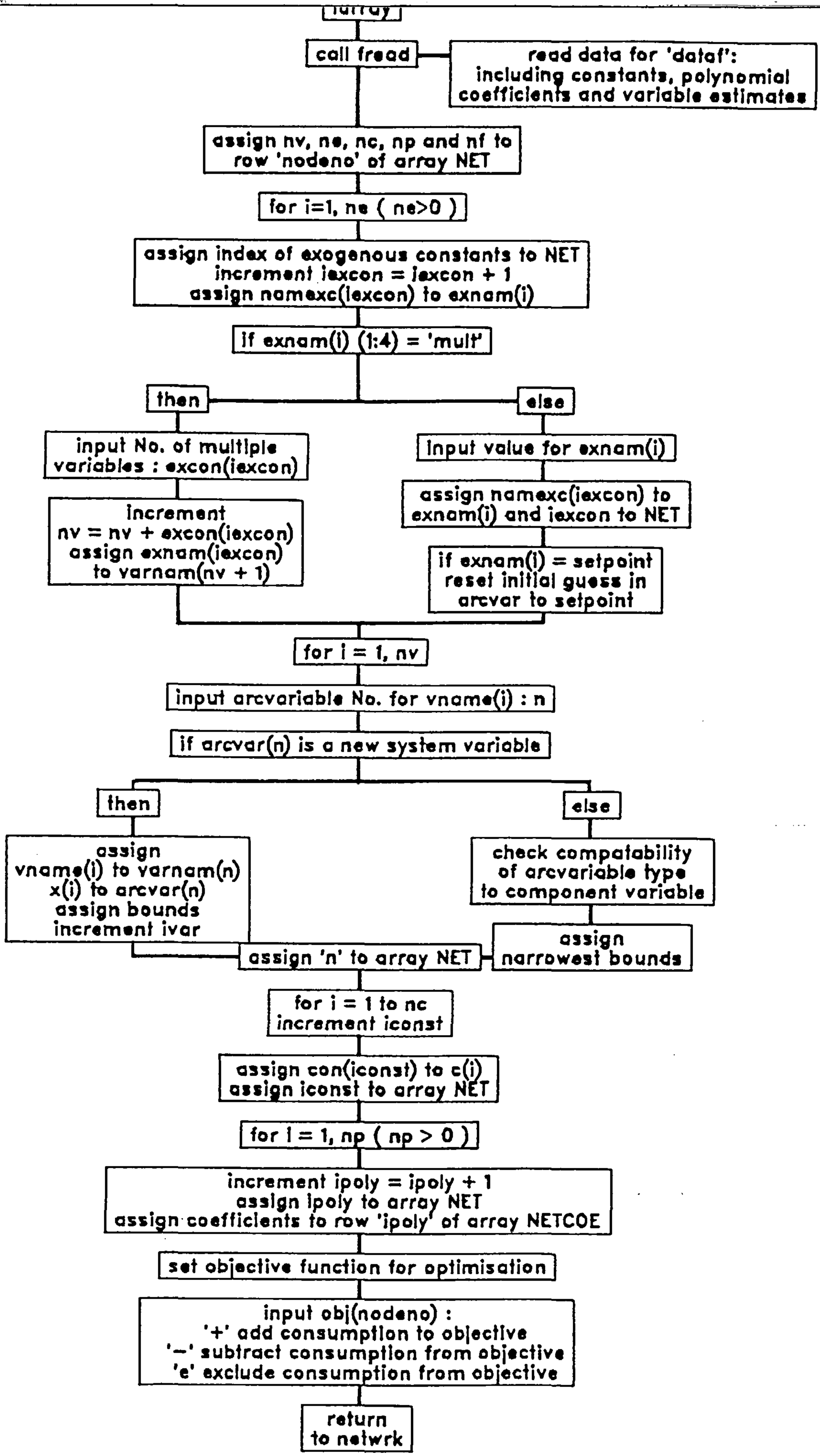


figure 5.4 Flow Diagram for 'iarray' Subroutine

5.2.2 Changing a Network Definition

Option command 'chn' calls subroutine <change> to allow the network definition to be altered under program control. The list of options within change is shown in figure (5.1):

1) a different component data record may be specified for a particular component, allowing data for a different size component or data for a different manufacturer to be inserted into the network arrays.
2) the initial estimate for any of the arc-variables may be altered, allowing the search algorithms to start from a different point. The bounds on a variable may also be altered to ensure good scaling of the variables within the optimisation routines.
3) the value of an exogenous constant may be changed to permit a component to be re-specified, for example the face area or water circuits of a coil may be altered.
4) an alternative set of exogenous variables may be specified. Because all the system equations are solved in an implicit form any combination of arc-variables may be declared exogenous, providing that there is a system of 'n' equations in exactly 'n' unknown variables.
5) the system objective function may be redefined by adding, subtracting or excluding the direct consumption of a component.
6) the principle indexing arrays and data record names for a network are displayed at the terminal as a visual check on system definition

Upon completion of any of these options, program control passes to subroutine <option>.

5.2.3 Newton Raphson Solution

Option command 'sol' calls the subroutine <nrsol> (figure 5.5), to solve all the system equations to give an operating point for the system. The operating point of a HVAC system may be completely defined by specifying values for each of the exogenous variables:

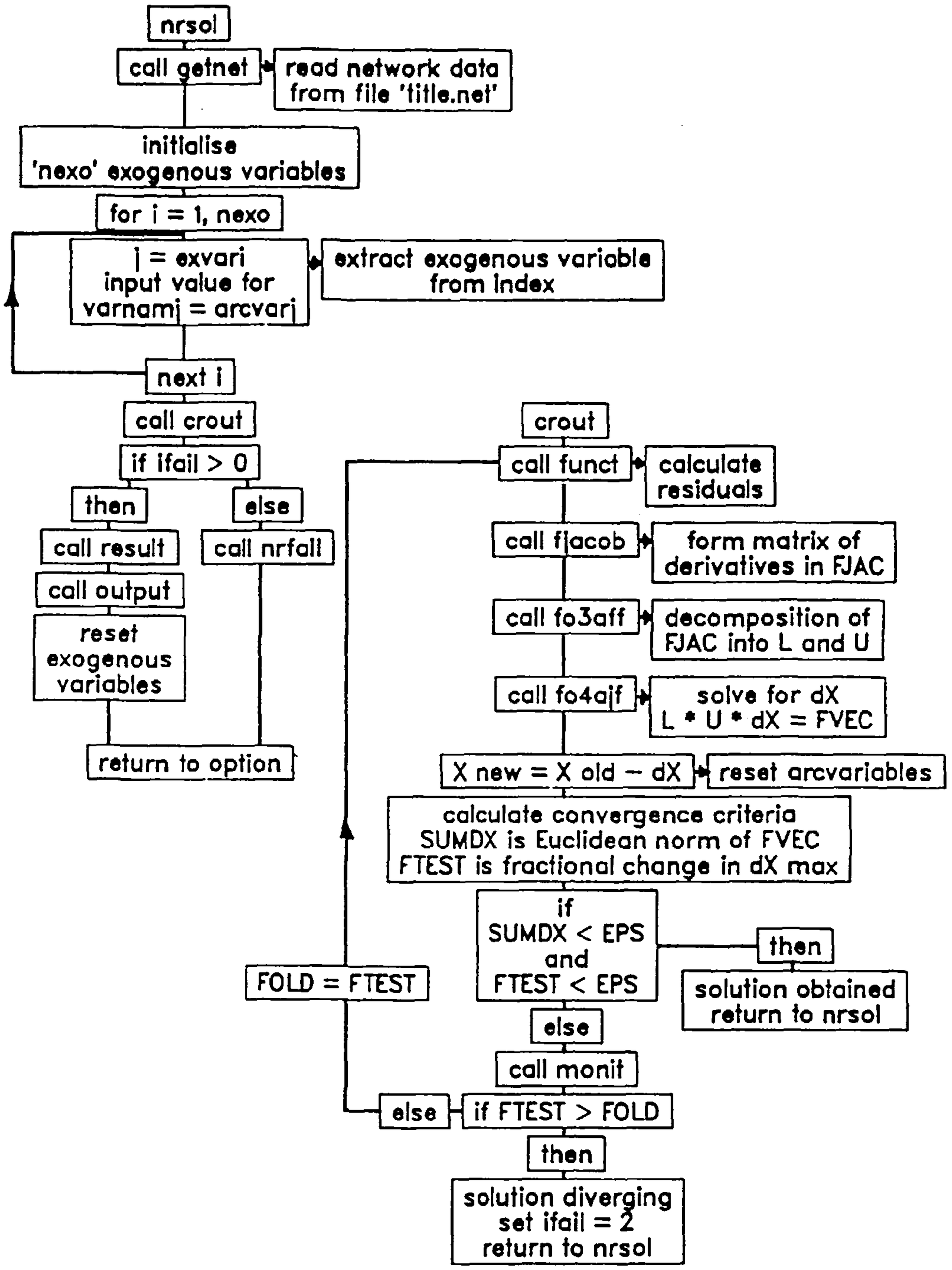


figure 5.5 Flow Diagram for 'nrsol' & 'crout' Subroutines

thus by changing the value of an exogenous variable the load on a system may be altered.

Subroutine ⟨nrsol⟩ determines an operating point for a system by prompting of a value for each exogenous variable in turn and then calling subroutine ⟨crout⟩ to solve the residual equations for that operating point. The subroutine ⟨crout⟩ (figure 5.5) is an implementation of the Newton Raphson algorithm outlined in Appendix B. If on return from ⟨crout⟩ the routine has failed to find a feasible operating point for the system then subroutine ⟨nrfail⟩ is called to analyse the progress of the solution algorithm as an aid in the diagnosis of the cause of failure. If all the equations have been solved, giving a feasible operating point for the system, then subroutine ⟨result⟩ is called to interpret the operating point for each node in the system in turn. The subroutine ⟨output⟩ may be called to print out all the system variable values and the exogenous variables may be reset for another operating point. Control passes to ⟨option⟩ upon completion of ⟨nrsol⟩.

5.2.4 GRG2 Optimisation

Option **'grg'** calls subroutine ⟨grgsol⟩ which uses the GRG2 algorithm, described in Appendix C, to find a feasible operating point for the system and optionally to minimise the system objective with respect to some of the exogenous variables. A flow diagram for ⟨grgsol⟩ is shown in figure (5.6).

If the objective function, at present the total energy consumed, is to be minimised then the set of system variables is extended by adding a number of exogenous arc-variables for which upper and lower bounds for the optimistaion must be specified. The GRG2 algorithm then finds an operating point of the system by using a 'phase 1' objective formed from the constraint violations (ie: residual function values) to find a feasible operating point where all the constraints are satified, and then minimising the system objective function with respect to the optimisation variables whilst maintaining feasibility.

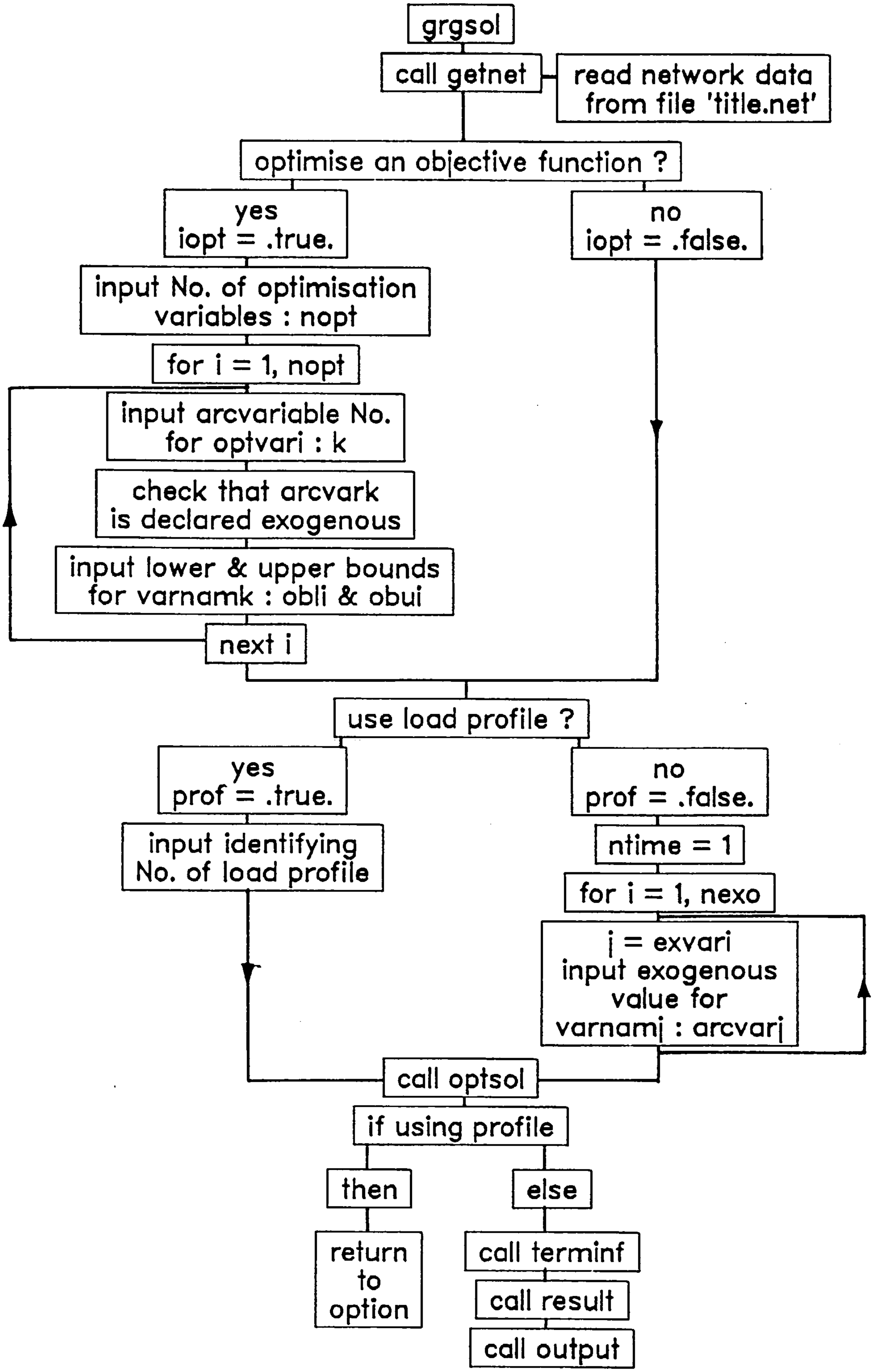


figure 5.6 Flow Diagram for 'grgsol' Subroutine

The performance of a HVAC system over a number of different time periods or load cases may be simulated by specifying that the values for the exogenous variables should be read from a previously defined load profile (section 5.2.5), in which case a profile number must be specified. If the simulation is for a single load case only then the values for each exogenous variable must be specified to define a single operating point.

The solution algorithm GRG2 is called via ⟨optsol⟩ (figure 5.7) which uses the variable bounds to scale the variables as described in section (4.3.1). Subroutine ⟨optsol⟩ calls an 'easy to use' subroutine interface with GRG2 called ⟨subgrg⟩ which sets up default parameters for the particular application of GRG2 to the simulation of HVAC systems. GRG2 calls subroutine ⟨gcomp⟩ which computes the vector of residuals and the system objective function by de-scaling the variables and calling ⟨execall⟩ which then calls the appropriate executive routine for each node type. If a load profile is in use then the results for each time period are written to a file called '*.res' where '*' is a concatenation of the system title and the profile number.

On exit from ⟨optsol⟩ control returns directly to ⟨option⟩ if a load profile has been used or calls subroutine ⟨terminf⟩ to print information on the mode of termination of GRG2. If a feasible point has been found ⟨result⟩ and ⟨output⟩ are called and the exogenous variables may be reset to solve the system for another operating point. If GRG2 has not found a feasible point a brief message outlining the form of failure is printed and program control returns to ⟨option⟩.

5.2.5 Simulation Load Profiles

Option command **'pro'** calls subroutine ⟨simdat⟩ which allows a load profile of exogenous variables for up to twenty five time periods to be specified (figure 5.1). Compilation of a new profile requires values for each exogenous variable for each time period to be entered, however once the data for the first time period has been

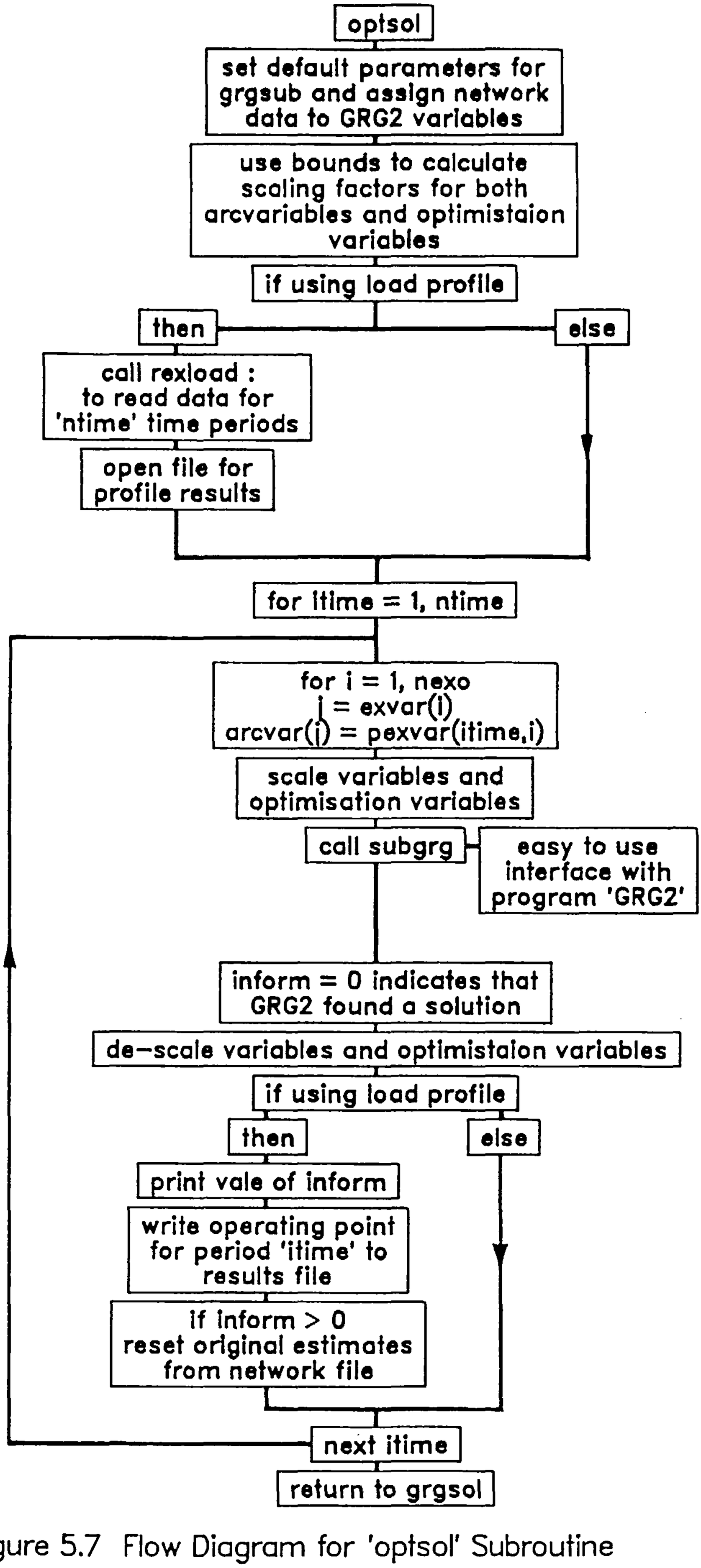


figure 5.7 Flow Diagram for 'optsol' Subroutine

completed this may be used to automatically generate data for subsequent time periods. The exogenous variables which change with time period may then be altered by using the profile option 'ed'. The profile data is saved on a file called '*.sim' where '*' is a concatenation of the system title and the profile number.

5.2.6 Interpretation of Profile Results

Option command 'res' calls subroutine <intres> to interpret the results of a profile simulation. The results for either a single time period or all the time periods in the profile may be interpreted by calling <result> for each time period of interest and optionally printing the operating points.

5.2.7 Optimised Design of HVAC Systems

Option command 'des' calls a suite of programs written by J.A.Wright which are being developed to permit optimisation of equipment selection and system design with repect to some objective function. The results of this study are expected to be published in December 1986.

5.2.8 Curve Fitting of Performance Data

Option command 'fit' calls a suite of subroutines by J.A.Wright which produce the option menu shown in figure (5.1) for fitting polynomial curves to manufacturers' data (Wright 1984). Performance data may be input either in tabular form or by digitising published performance curves. The routines find the optimum powers of a polynomial fit for the data and store the coefficients on a file called '*.coedat' where '*' is a name for the component. Graphics routines available within the curve fitting routines allow the user to plot both the original data and the fitted polynomial as a visual check on the efficacy of a particular fit.

5.2.9 Component Data Record Management

Option command 'dat' calls up a sub-menu of data file options shown in figure (5.1). Component data records are written in a directory file called '*.dir' where '*' is the component name specified in the

initialistaion routine and which appears in the node menu. The data files are direct access, unformatted. The first five logical records containing a directory of the data records available in the file. The position of a record name in this directory identifies the logical record number at which the data for that component record begins. Data records may be created, altered or deleted by using the data options **'cre'**, **'alt'** and **'del'** respectively.

Data option **'new'** allows any existing data files to be transformed to a new format when the program parameters, such as the maximum number of data constants, are altered during program development or transfer to another machine.

Data option **'tra'** enables individual data records to be transferred between the directory files of users of the multi-user version of SPATS. This is particularly useful where a component data record includes many constants or several sets of polynomial co-efficients, since the resulting data record may be transferred for use by other users without having to transfer original coefficient files or manually re-typing data.

5.3 Component Models

The library of component models available within the current version of SPATS is shown in figure (5.8). The first three columns of the node menu, 'MAIN PLANT', 'FITTINGS' and 'CONTROLS' indicate component models which have been developed and tested. The subroutines for these models are referenced explicitly within the SPATS subroutines and are generally available within the multi-user implementation of SPATS. The fourth column of the node menu is headed 'TEST NODES' and contains names for those component models currently under development.

The current version of SPATS has provision to reference up to fifteen 'TEST NODES' by explicit reference to subroutines called ‹itest1›, ‹itest2›, ..., ‹etest1›, ‹etest2›, ..., ‹rtest1›, ‹rtest2›, ..., for initialisation, executive and results routines respectively. This

MAIN PLANT	FITTINGS	CONTROLS	TEST NODES
1 - boilers_	16 - mix-tees	31 - mixvalve	46 - multizon
2 - axialfan	17 - duct-ins	32 - modvalve	47 - q-rads__
3 - cent-fan	18 - ventecon	33 -	48 - zone-s/l
4 - wchiller	19 - roomzone	34 - d-valve_	49 -
5 - clgtower	20 - air-zone	35 - hmdrctrl	50 - ecomiser
6 - h/c-coil	21 - conv-wye	36 - stepcont	51 -
7 - radiator	22 - div--wye	37 - pcontrol	52 -
8 - compresr	23 - duct-siz	38 - siginvtr	53 -
9 - heatexch	24 - duct-sim	39 -	54 -
10 - humidifr	25 - ftgs-sim	40 -	55 -
11 -	26 -	41 -	56 -
12 -	27 -	42 -	57 -
13 -	28 -	43 -	58 -
14 -	29 -	44 -	59 -
15 -	30 -	45 -	60 -

figure 5.8 SPATS Component Menu

enables different component models to be developed and tested without having to alter the source code within the SPATS subroutines since the ⟨test⟩ subroutines are found and linked into SPATS automatically using the MULTICS search rules (section 5.1). Thus different users of SPATS can develop, test and validate different models which may then be included explicitly in future versions of SPATS.

Each component model requires three subroutines: an initialisation subroutine, an executive subroutine and a results subroutine. The form of each of these subroutines is described below illustrated by reference to the boiler model (Appendix A-1).

Since the FORTRAN text for parameter declarations, common blocks and type declarations is identical for each type of subroutine these are referenced by a '%include ....' statement. The FORTRAN text to extract the data for a particular node from the network arrays is identical for all executive and results routines (figure 5.12) and is referenced by the '%include dataex' statement. A list of common blocks and common block variables is given in Appendix E.

5.3.1 Initialisation Subroutines.

The initialisation subroutine for a component is used in both the system definition subroutines and in the component data handling routines. The FORTRAN text for subroutine ⟨iboiler⟩ is shown in figure (5.9). Names for each of the variables, exogenous constants and data constants associated with a component are assigned to the local variables **name**, **namex** and **namcon** respectively using 'data' statements (NB: the boiler model does not require any exogenous constants). These variables are then assigned to variables in common **vname**, **exnam** and **cname**. Names for polynomials may be added after the names for the constants in **namcon** (eg: in subroutine ⟨iboiler⟩ **namcon**(3) = 'temp-eff.' a polynomial relating efficiency to inlet temperature and fraction of maximum rated load). These polynomial names are used only as prompts in the data file subroutines.

```
c     ***************************************************************
      subroutine iboiler
c     initialisation routine for boiler
c
      %include icommon
c
c
      data name/'tw-inlet','tw-flow ','mw-flow ','fuelrate',
     c12*'          '/
      data namcon/'maxRload','eff. mrl','temp-eff.',27*'          '/
c
c     assign data names to common block variables
c
      do 10 i=1,mvar
10    vname(i)=name(i)
      do 20 i=1,mcon
20    cname(i)=namcon(i)
c
c     name for node menu and directory file.
      file='boilers_'
c
c     specify parameters for component
      nv=4                                (No. of system variables)
      nc=2                                (No. of component constants)
      ne=0                                (No. of exogenous constants)
      np=1                                (No. of polynomials)
      nf=1                                (No. of residual functions)
c
c     indicate that component has a direct consumption
      objchar='+'
c
      return
      end
c
c     ***************************************************************
```

figure 5.9 Initialisation Routine for Boiler Model

The name for a component model assigned to **file** is used by subroutine ⟨setnode⟩ to form the menu of nodes available, and by the data file subroutines to form a name for the directory file of component data records: it must be eight characters, lower case and must contain no blank characters.

If the component has a direct consumption of energy this is indicated by assigning a character to **objchar**, such as '+'. The variable **objchar** is set to 'e' by default prior to calling any initialisation subroutine and unless this is changed by the initialisation subroutine the network definition subroutines do not expect a consumption for that node type.

5.3.2 Executive Subroutines

An executive subroutine for a component model is used to calculate values for the residual equations which define the performance of the component. The FORTRAN text for the subroutine ⟨eboiler⟩ is shown in figure (5.10). Local constants may be defined, such as the thermodynamic properties of fluids although these properties can be determined as functions of temperature, pressure etc. in more rigorous component models.

Data for a particular node in a system network is extracted from the network arrays in 'dataex' (figure 5.12), in which the arc-variables for the component are assigned to array **x**, the constants to array **c**, and the exogenous variables to array **e**. Co-efficients for each polynomial are assigned to a row of array **coefs**. If intermediate variables are required in the computation of the residual equations, such as Δt, Q and γ in ⟨eboiler⟩ (figure 5.10), these are assigned to the unused portions of arrays **x**, **c** and **e** to save declaring extra local variables in each executive subroutine.

The computational sequence to calculate the residual function(s) for a component may take any form, from simple mass and energy balances for a mixing tee (Appendix A-8), to the more complex conditional branching and iterative loops required to define the performance of a

```
c     ***************************************************************
      subroutine eboiler (n,fvec)
c
c     executive routine for temperature dependent boiler model
c
      %include ecommon
c
c     set constants
      cpw=4.81                              (kJ/kg °C)
      np=1                                  (one polynomial in model)
c
c     extract component data from network arrays
      %include dataex
c
c     variables :     1 - tw inlet °C
c                     2 - tw outlet °C
c                     3 - mass flow rate of water  kg/s
c                     4 - fuel input rate kW
c
c     constants :     1 - L, the maximum rated load (mrl) kW
c                     2 - η, efficiency at mrl (fraction)
c
c     polynomial :    1 - normalised η vs fraction of full load
c                         and tw inlet
c
c     temporary variabes :
      x(7) = x(2) - x(1)                Δt  temperature difference
      x(8) = x(3) * cpw * x(7)          Q   boiler output
      x(9) = x(8) / c(1)                γ   fraction of full load = Q/L
c
c     check range of variable for polynomial
      if( x(9) .gt. 1.0) x(9)=1.0
      if( x(9) .lt. 0.05) x(9)=0.05
c
c     calculate polynomial
      x(10) = polyxz( coefs,mcoe,x(9),x(1),1)        η - efficiency
c
c     increment function counter and calculate residual function
      ifunc=ifunc+1
      fvec(ifunc) = (x(8)/(c(2)*x(10))) - x(4)
c
c     assign direct consumption of node to objective function
      objfun(nodeno)=x(4)
c
      return
      end
c
c     ***************************************************************
```

figure 5.10 Executive Routine for Boiler Model

cooling coil (Appendix A-3). Care must be exercised in the development of a component algorithm to ensure that the model portrays the performance of the component at all feasible operating conditions, not just at the operating conditions normally associated with peak load design methods.

The range of values for the arguments to a polynomial function may be checked either directly within the executive routine, as in ⟨eboiler⟩, or by assigning suitable bounds to the variables in the network definition. Polynomials are evaluated by calling the functions ⟨polyxz⟩ for polynomials in two variables or ⟨polyx⟩ for polynomials in one variable. The arguments are : **coefs** - array of polynomial co-efficients, **mcoe** - dimension parameter for the maximum number of co-efficients for any one polynomial, either one or two independent variables (eg: γ - x(9) and t-inlet - x(1) for ⟨eboiler⟩), and n an index for the n^{th} polynomial in the executive routine.

The index **ifunc** is incremented for each residual function and the residual equation is assigned to the function vector **fvec**. If the component has a direct consumption this is assigned to the node by node objective function array **objfun** from which the system objective function may be compiled using the definition of the objective for the system network.

5.3.3 Results Subroutines

Once an operating point for a system has been established a results routine for each node may be called to calculate quantities of interest which do not appear as variables within the simulation, such as the sensible heat ratio of a coil (Appendix A-3) or the efficiency of a fan (Appendix A-2). The FORTRAN text for the subroutine ⟨rboiler⟩ is shown in figure (5.11). Results subroutines are generally similar to the executive subroutines except that the residual function assignment is replaced by statements to print out the component data record used and the results of interest. The results subroutine may also be used to print warning messages where a

```
c     ***************************************************************
      subroutine rboiler
c
c     results routine for temperature dependent boiler model
c
      %include ecommon
c
c     set constants
      cpw=4.81                              (kJ/kg °C)
      np=1                                  (one polynomial in model)
c
c     extract component data from network arrays
      %include dataex
c
c     variables :     1 - tw inlet °C
c                     2 - tw outlet °C
c                     3 - mass flow rate of water  kg/s
c
c     constants :     1 - L, the maximum rated load (mrl) kW
c                     2 - η, efficiency at mrl (fraction)
c
c     polynomial :    1 - normalised η vs fraction of full load
c                         and tw inlet
c
c     temporary variabes :
      x(7) = x(2) - x(1)                Δt  temperature difference
      x(8) = x(3) * cpw * x(7)          Q   boiler output
      x(9) = x(8) / c(1)                γ   fraction of full load = Q/L
c
c     check range of variable for polynomial
      x(12)=x(9)
      if( x(9) .gt. 1.0) x(12)=1.0
      if( x(9) .lt. 0.05) x(12)=0.05
c
c     calculate polynomial
      x(10) = polyxz( coefs,mcoe,x(12),x(1),1)        η - efficiency
c
c     express fractions as %
      x(9)=x(9) * 100
      x(10)=x(10) * c(2) * 100
c
c     print out results of interest
      write(*,130)nodeno,fname(nodeno),x(8),x(9),x(10)
130   format(1h ,2x,i2,10x,a8,10x,'boiler output = ',f10.4,' kW'/
     c1h ,32x,'%   full load = ',f10.4,' %'/
     c1h ,28x,'boiler efficiency = ',f10.4,' %'/)
c
c
      return
      end
c
c     ***************************************************************
```

figure 5.11 Result Routine for Boiler Model

```
c     indecies of positions in array 'net'
      iv = 2
      ie = 3 + mvar
      ic = 4 + mvar + mex
      if = 5 + mvar + mex + mcon
c
c     read values for component parameters
      nv = net(nodeno,iv)
      ne = net(nodeno,ie)
      nc = net(nodeno,ic)
      nf = net(nodeno,if)
c
c     extract data from network arrays
c
c     arcvariables
      do 100 j = iv + 1, iv + nv
      i = net(nodeno,j)
      k = j - iv
      x(k) = arcvar(i)
100   continue
c
c     exogenous constant
      if(ne.ne.0) then
        do 110 j = ie + 1, ie + ne
        i = net(nodeno,j)
        k = j - ie
        e(k) = excon(i)
110     continue
      endif
c
c     constants
      if(nc.ne.0)then
        do 120 j = ic + 1, ic + nc
        i=net(nodeno,j)
        k=j-ic
        c(k)=con(i)
120     continue
      endif
c
c     polynomials
      if(np.ne.0)then
        do 150 i=1,np
        j = net(nodeno, ic + nc + i)
           do 140 k=1, mcoe
           coefs(i,k) = netcoe(j,k)
140        continue
150     continue
      endif
c
c     end of data extraction
```

figure 5.12 FORTRAN text for 'dataex'

component is operating outside of its normal range, such as when the throttling range of a controller is exceeded.

5.4 Validation of SPATS.

Validation procedures for building energy performance simulation techniques have attempted to compare the predicted energy consumption with the measured consumption over a suitable period, however these efforts are complicated by the difficulty of simulating random fluctuations in occupancy and of measuring and storing large amounts of data (Kusuda 1981). Field validation exercises, where a building is instrumented and the operating conditions compared with the computer simulation, have been succesful on a small scale (Arumi-noe 1979) but these methods would be difficult to apply to a large scale HVAC system due to the complexity of the task and the difficulty of isolating the simulation parameters from other operational effects.

The efforts of the International Energy Agency - Annex 10 Program (I.E.A. 1984) may prove most useful in verification of simulation procedures. An exercise to simulate a multi-storey residential hydronic heating system enabled a comparison of the measured performance with the predicted performance (La Chaumiere exercise). The exercises allow the significance of basic simulation hypotheses to be checked (ie : dynamic vs steady state models) and a comparative validation of several component models to be made.

Further I.E.A. Annex 10 exercises planned include the simulation of part of a V.A.V. system in the Collins Building in Glasgow : part of this exercise is used as an example of the application of SPATS in chapter 6. The procedure used in these exercises is to simulate the performance of the system over a few specific days since, once a procedure is validated to produce results for a few representative measured days then it may be reasonably assumed that the fundamental simulation methodology is sound.

The simulation procedure used in SPATS is component based, therefore the problem of validation is largely reduced to the validation of individual component models. Where the component models are based on published performance data validation consists of ensuring that the model can reproduce the original data: however a model developed from fundamental principles can only be validated by a parametric study of a fully instrumented laboratory experiment or by cross validation against other validated models provided care is taken to use the same test data and assumptions. A proposal by Kusuda is to adopt a standard set of systems and building performance data, for which an accurate engineering analysis may be performed manually, and against which the results of any new simulation methodology may be compared (Kusuda 1981).

Chapter 6. APPLICATIONS OF SPATS

In addition to the primary objective of developing a methodology for a component based system simulation procedure, the SPATS suite of programs has been used extensively in the development of HVAC component algorithms and as a basis for further research into the optimal selection of HVAC components.

6.1 Component Algorithm Development

New component algorithms have been developed using SPATS by defining the algorithms as 'TEST NODES' (section 5.3). A network may be defined which consists of a single node and the performance of the component algorithm may be tested with respect to any combination of the system variables declared exogenous. This is in contrast to an input/output formulation for a component algorithm in which the component performance is described by explicit equations which require rearrangement to analyse the performance with respect to a different sub-set of system variables. One possible application of this implicit formulation of component algorithms is to develop a selection program in which performance data from different manufacturers, presented in different ways, could be compiled into compatible data files to facilitate easy comparison of different items of equipment. Analysis of the performance of a single component with respect to any of the associated variables would be useful in predicting the behaviour of, say, a coil at conditions other than those used for initial selection or design.

The SPATS framework is also suitable for comparative analysis of different algorithms for the same component type. The different component models may be specified as test routines and either analysed as individual networks or defined in similar system networks for an analysis as part of a larger HVAC system.

6.2 System Simulation

System simulation may best be used as a design tool in the early stages of a project as an aid in evaluating alternative proposals for the HVAC systems, rather than as a means for evaluating the likely

running costs of an installation once the design has been finalised. The techniques of simulation are appropriate for a comparative analysis between alternatives to estimate the relative costs or the relative savings of a design proposal. Finite prediction of the energy use of a system is inappropriate because of the difficulty of simulating actual usage patterns and operating data. Small changes in the way a building is operated, such as altering thermostats or keeping doors open, can invalidate basic assumptions in the simulation. Integration of techniques of simulation into the design process is likely to require fundamental changes in approach: there is no point in performing detailed simulation of alternatives at an early stage in the design process to find that the client has nothing left in the budget during the later stages of the design to act upon alternative proposals (Nall 1983).

The two examples which follow illustate the application of SPATS to the analysis of two hypothetical systems, a multi-zone hydronic system and a multi-fluid loop single zone air system.

6.2.1 Simulation of a Domestic System

A network for the simulation of a domestic hydronic heating system is shown in figure (6.1). Zone one has a combined heat transfer coefficient (U-value multiplied by area) of 0.2 W/°C and is served by two radiators in series, the output of which is controlled by the action of a thermostatic radiator valve (TRV) with a proportional controller. Two radiators in parallel, controlled by separate TRV's, serve zone two which has a heat transfer coefficient of 0.31W/°C. The radiator serving zone 3 is uncontrolled to allow for full variation of water mass flow rate as the valves in zones one and two close whilst maintaining a constant mass flow rate through the boiler. The hydaulic characteristics of the system are not modelled in this simulation, hence zone three is equivalent to a towel rail or hot water cylinder in a practical system which ensures a complete circuit irrespective of the action of the valves. The output of the boiler is controlled by the action of a modulating valve which throttles the

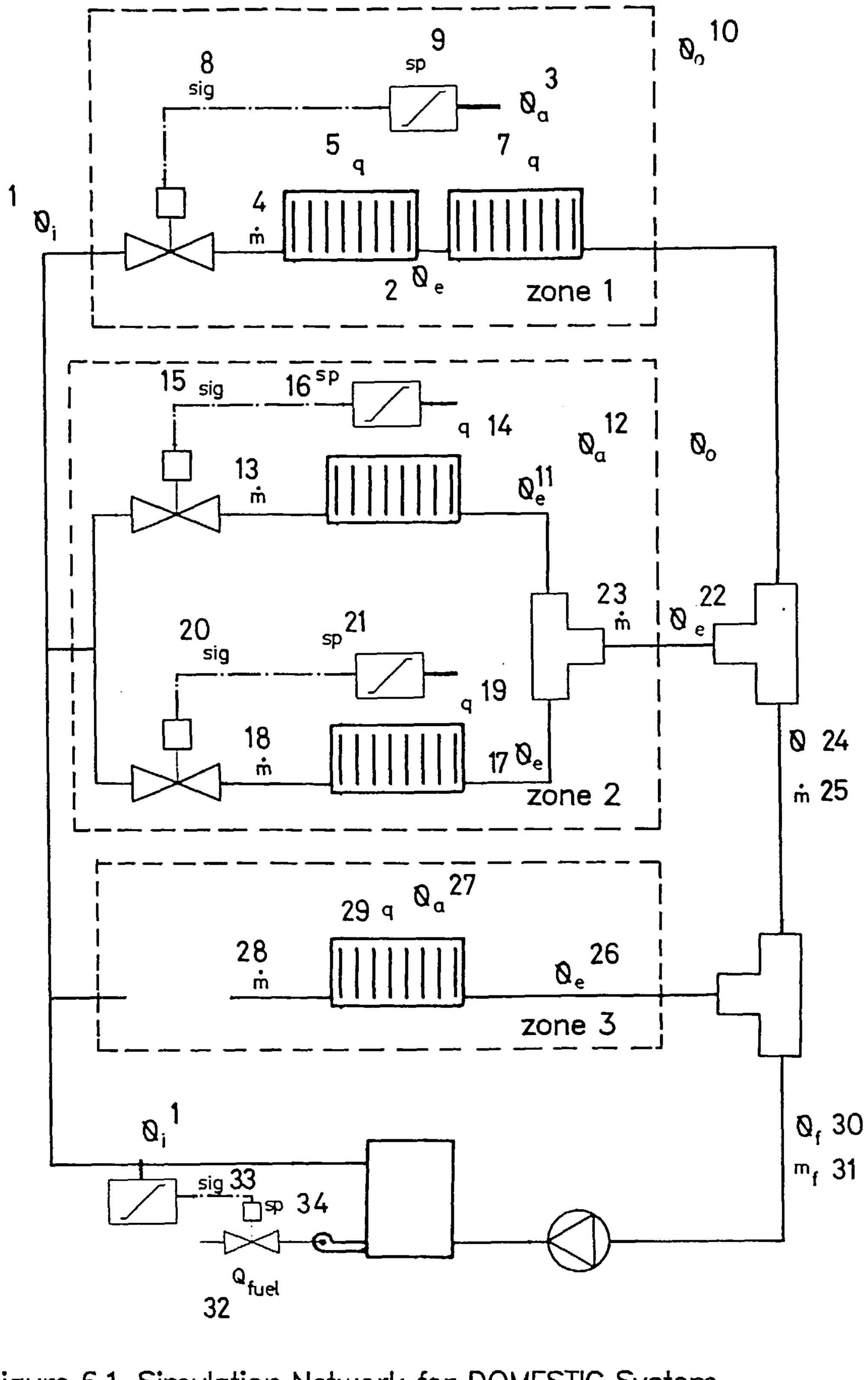


figure 6.1 Simulation Network for DOMESTIC System

fuel input rate to maintain a set water flow temperature within a proportional band.

The system network comprises twenty components which are represented by twenty eight equations in thirty four variables. To specify a fully defined system six variables are declared exogenous: the set points for the four proportional controllers, the mass flow rate of water (provided exogenously by the pump which is not modelled) and the external environmental temperature. The full flow mass flowrate is calculated from the maximum flow rate through each branch, ie: three TRV's at 0.2 kg/s max, plus the rated flowrate for the uncontrolled radiator in zone 3, ie : 0.05, giving a total of 0.65 kg/s.

The simulation profile for the system varies the outside temperature (arc-variable 10) from -5 °C through to 20 °C. The resulting room temperature profile is shown for each room in figure (6.2): note that the rise in room temperature in zone three would seem severe, however in practice either the zone infiltration rate would decrease the resultant temperature or the boiler would be turned off at low loads rather than being modulated down to less than twenty percent full load as in this simulation.

A typical operating point for the system at t_{ao} = -3 °C is shown in figure (6.4) : note that the set point temperature for zone two is not maintained at this condition. The output of the radiators could be increased in several ways, such as selecting a larger radiator or a larger TRV. This operating point represents a poorly designed system since the Δt across the radiators is approximately 5°C compared with a design Δt of 10 °C, indicating that the radiators are undersized for this duty. The operating point for the same system but with the radiator lengths increased to 2.5m and 3.0m respectively is shown in figure (6.5).

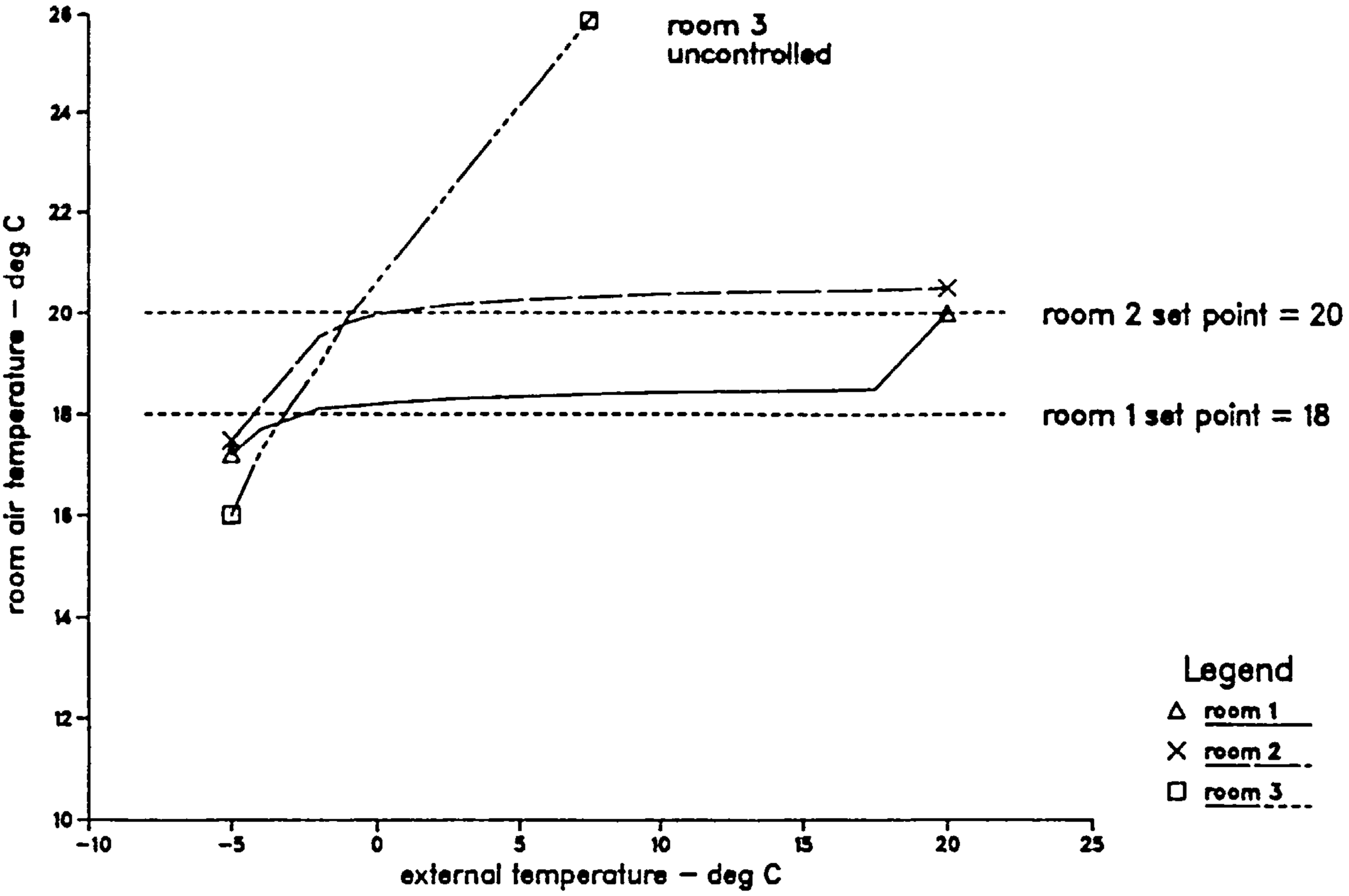


figure 6.2 Temperature Profile for DOMESTIC System

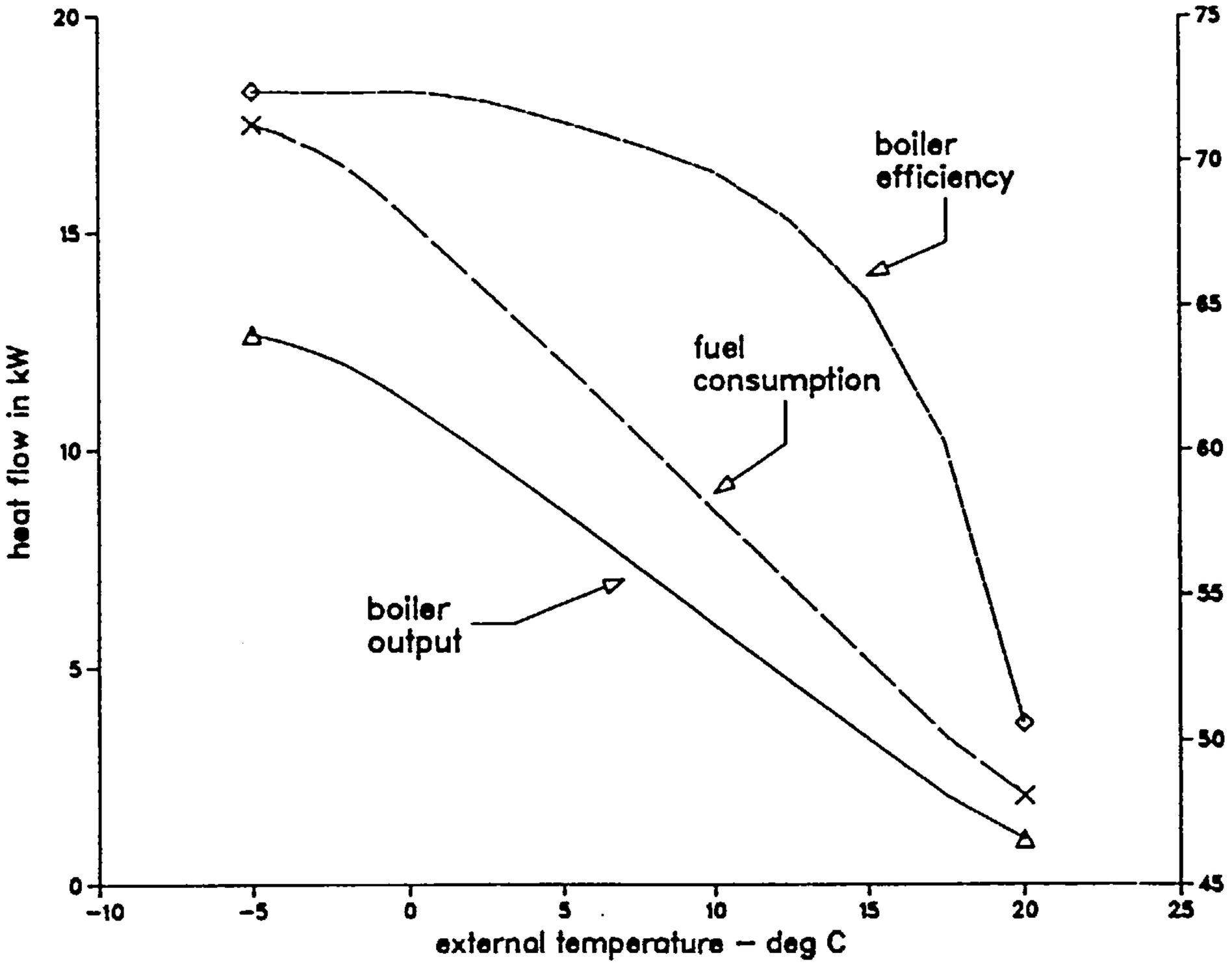


figure 6.3 Boiler Performance for DOMESTIC System

```
period 2  - minus 3
Solution obtained from GRG2
results for domestic simulation    dated : 05/12/84  by : Mike

node no.    data record

   1          mys117          radiator output =      1.9878 kw

   2          mys117          radiator output =      2.1963 kw

   3          trv-200         flow input rate =      0.1167

   5          hydronic             total load =      4.1833 kW
                                     room dbt =  17.9165 deg C
                                  outside air =  -3.0000deg C
   6          mys217          radiator output =      3.8657 kw

   7          trv-200         flow input rate =      0.2000


   8          propcont
              WARNING  variable outside range -      18.8466
   9          mys217          radiator output =      3.7867 kw

  10          trv-200         flow input rate =      0.2000


  11          propcont
              WARNING  variable outside range -      18.8466
  13          hydronic             total load =       6.7724 kW
                                     room dbt =  18.8466 deg C
                                  outside air =  -3.0000deg C
  15          mys212          radiator output =       1.2699 kw

  16          hydronic             total load =       1.2699 kW
                                     room dbt =  18.1644 deg C
                                  outside air =  -3.0000deg C
  18          tboil-15           boiler output =  12.2256 kw
                                 %   full load =  81.5041 %
                               boiler efficiency =  72.3448 %

  19          mod-20          flow input rate =      16.8993
```

```
Arcvariables :-
No.  1     tw-flow  =    79.6558
No.  2     tw-retrn =    75.5813
No.  3     dbt-room =    17.9165
No.  4     mw-flow  =     0.1167
No.  5     q-output =     1.9878
No.  6     tw-retrn =    71.0787
No.  7     q-output =     2.1963
No.  8     signal   =     0.4165
No.  9     setpoint =    18.0000
No. 10     ta-outsd =    -3.0000
No. 11     tw-retrn =    75.9879
No. 12     dbt-room =    18.8466
No. 13     mw-flow  =     0.2000
No. 14     q-output =     3.8657
No. 15     signal   =    -0.6534
No. 16     setpoint =    20.0000
No. 17     tw-retrn =    75.2212
No. 18     mw-flow  =     0.2000
No. 19     q-output =     3.7867
No. 20     signal   =    -0.6534
No. 21     setpoint =    20.0000
No. 22     tw-full  =    75.6045
No. 23     mw-full  =     0.4000
No. 24     tw-full  =    74.5824
No. 25     mw-full  =     0.5167
No. 26     tw-retrn =    77.3761
No. 27     dbt-room =    18.1644
No. 28     mw-flow  =     0.1333
No. 29     q-output =     1.2699
No. 30     tw-full  =    75.1554
No. 31     mw-full  =     0.6500
No. 32     fuelrate =    16.8993
No. 33     signal   =     0.1550
No. 34     setpoint =    80.0000
```

figure 6.4 Operating Point for the DOMESTIC System

```
Solution obtained from GRG2
results for domstc-1 simulation    dated : 11/12/84  by : mapm

node no.    data record

   1          mys117          radiator output =      1.9864 kw

   2          mys117          radiator output =      2.1965 kw

   3          trv-200         flow input rate =      0.1171

   5          hydronic             total load =      4.1829 kW
                                     room dbt =  17.9143 deg C
                                  outside air =  -3.0000deg C
   6          mys217          radiator output =      3.3848 kw

   7          trv-200         flow input rate =      0.0825

   9          mys217          radiator output =      3.7723 kw

  10          trv-200         flow input rate =      0.0825


  13          hydronic             total load =       7.1571 kW
                                     room dbt =  20.0875 deg C
                                  outside air =  -3.0000deg C
  15          mys212          radiator output =       1.2909 kw

  16          hydronic             total load =       1.2909 kW
                                     room dbt =  18.5154 deg C
                                  outside air =  -3.0000deg C
  18          tboil-15           boiler output =  12.6309 kw
                                 %   full load =  84.2061 %
                               boiler efficiency =  72.3878 %

  19          mod-20          flow input rate =      17.4490
```

```
Arcvariables :-
No.  1     tw-flow  =    79.6276
No.  2     tw-retrn =    75.5706
No.  3     dbt-room =    17.9143
No.  4     mw-flow  =     0.1171
No.  5     q-output =     1.9864
No.  6     tw-retrn =    71.0846
No.  7     q-output =     2.1965
No.  8     signal   =     0.4143
No.  9     setpoint =    18.0000
No. 10     ta-outsd =    -3.0000
No. 11     tw-retrn =    69.8124
No. 12     dbt-room =    20.0875
No. 13     mw-flow  =     0.0825
No. 14     q-output =     3.3848
No. 15     signal   =     0.5875
No. 16     setpoint =    20.0000
No. 17     tw-retrn =    68.6886
No. 18     mw-flow  =     0.0825
No. 19     q-output =     3.7723
No. 20     signal   =     0.5875
No. 21     setpoint =    20.0000
No. 22     tw-full  =    69.2505
No. 23     mw-full  =     0.1650
No. 24     tw-full  =    70.0120
No. 25     mw-full  =     0.2821
No. 26     tw-retrn =    78.7880
No. 27     dbt-room =    18.5154
No. 28     mw-flow  =     0.3679
No. 29     q-output =     1.2909
No. 30     tw-full  =    74.9787
No. 31     mw-full  =     0.6500
No. 32     fuelrate =    17.4490
No. 33     signal   =     0.1276
No. 34     setpoint =    80.0000
```

figure 6.5 Revised Operating Point for the Domestic System
(long radiators specified at nodes 6 & 9

The boiler output, fuel consumption and efficiency over the simulation profile are shown in figure (6.5). Note the decrease in efficiency as the load decreases. In a practical system a further thermostat might be used to detect a fall in the temperature rise across the boiler : when this Δt falls below a preset value, say 2°C, the circulating pump would be switched off. A simulation to model these control options at low loads would require the hydronic characteristics of the system to be included within the simulation and a wider range of controller models to be available.

The GRG2 solution algorithm was used to optimise the boiler set-point temperature with respect to the fuel consumption. Since the boiler model is temperature dependent (Appendix A-1) and the water pump is not included within the simulation, the optimisation phase invariably sets the boiler set point to its lower limit, even though the zone set points may not be maintained at this condition. This indicates that an objective function comprising the fuel consumption alone is insufficient and a more realistic objective would include both the additional 'cost' of pumping more water around the system at lower temperatures and a penalty term to compensate for 'loads not met' conditions. Study of the optimal control of HVAC systems, such as chiller control, are a useful application of SPATS, however lack of time prevented a detailed study of this aspect of system simulation within this investigation.

6.2.2 Collins Building Simulation Exercise

The Collins Building simulation exercise within I.E.A. Annex 10 includes modelling the performance of the variable air volume (V.A.V.) cooling system installed in the third floor penthouse suite of the Collins Publishing building in Glasgow. This building and its sytems was the subject on an earlier study (Annex 4) to measure the thermal performance of the building against the calculated performance predicted by various building thermal load analysis programs.

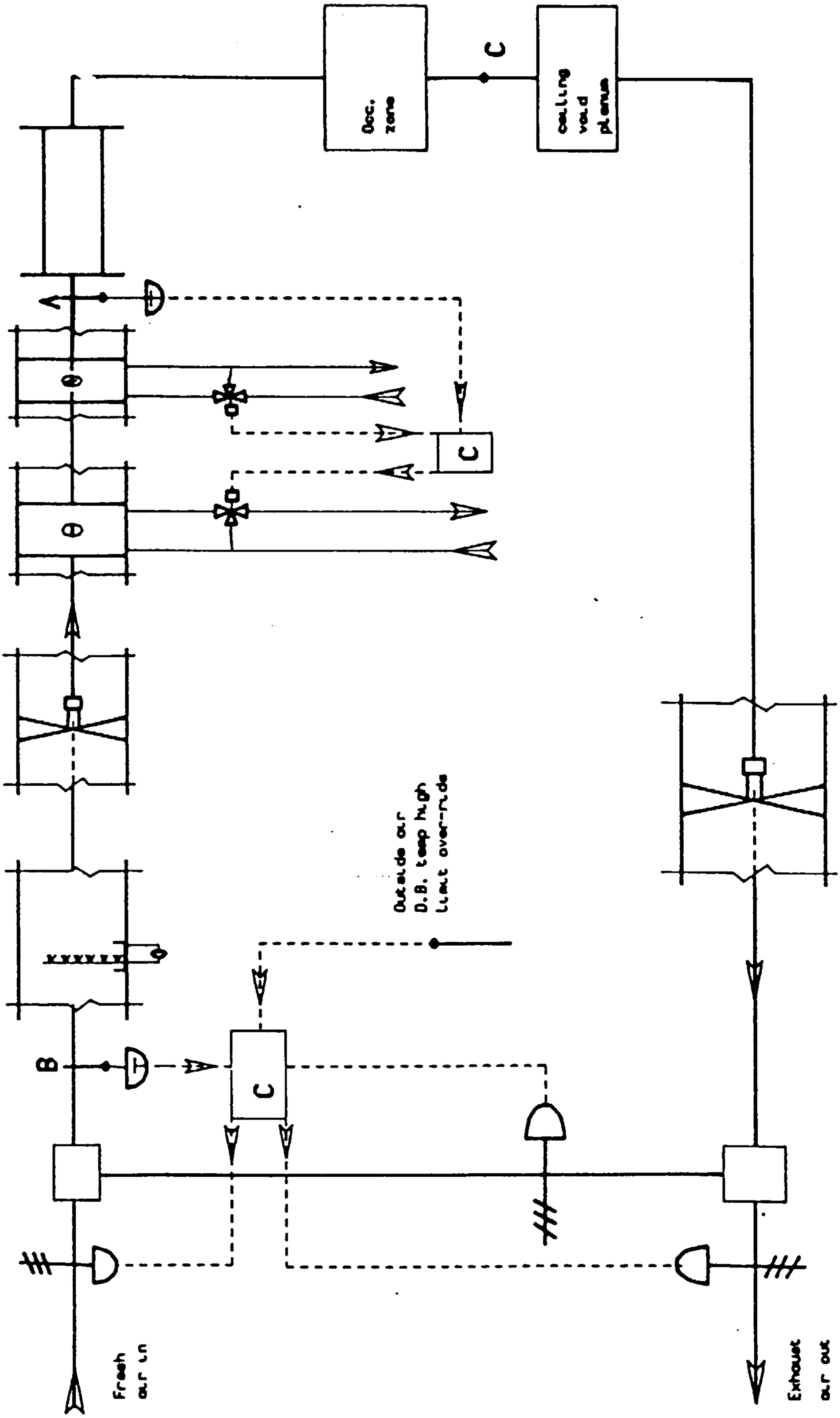


figure 6.6 Collins Building Exercise : System Configuration 1

The current exercise includes two system configurations, the network for configuration 1 is shown in figure (6.6). The supply system comprises :- modulating damper, mixing box, spray type humidifier, axial fan, cooling coil, heating coil and supply duct. System configuration 2 uses the same components in a different order, simulation of this configuration is not included here. Details of the systems, their controls and component performance data is given (I.E.A. 1984) along with confirmation of the assumptions to be made and measured weather data for one summer and one winter day (figure 6.7).

The performance of the systems is to be modelled under several simplifying hypotheses :-

i) 'ideal' control of air mass flow rate with respect to zone temperature within a proportional band of 6.2 m^3/s at 22.1°C and 1.24 m^3/s at 20.1°C. This avoids the added complexity of modelling the V.A.V. box for which there is as yet no agreed component algorithm within Annex 10 participants. The volumetric flow rates are arbitrarily converted to mass flow rates at a constant density of 1.2 kg/m^3.

ii) 'ideal' control of the zone conditions. Exercises a) and b) are based on constant zone conditions : 21°C 55% R.H. , and in exercises c) and d) the zone conditions are specified as varying sinusoidally between 20°C and 23°C, 50% and 70% R.H. over twenty four hours with peaks at midday. This eliminates the need to interface with a building zone model, so isolating the plant simulation from any building simulation, for the purposes of these exercises. The occupied zone, ceiling void and return air duct are specifically excluded from the simulation exercise.

iii) the ceiling void plenum is assumed to be at a constant condition of 22°C and 45% R.H.

iv) the fan blade pitch angle is adjusted to maintain 1000 Pa in the duct downstream of the heating coil.

v) proportional control of the supply air temperature, assuming no dead band between heating and cooling, as follows :

heating coil set point = 10.5°C range = $\pm$0.5°C

cooling coil set point = 11.5°C range = $\pm$0.5°C

vi) an enthalpy cycle comprising fresh air, exhaust and recirculation dampers which are used to maintain a mixed air temperature of 10°C $\pm$0.5°C. The minimum fresh air fraction is 0.1 which overrides the enthalpy control when the outside air exceeds 16°C.

vii) the humidifier is on only when the outside air is less than 14°C and the zone R.H. is less than 60%, hence for this exercise the humidifier is controlled exogenously.

viii) the hydronic systems are excluded from the simulation exercise. It is assumed that the chilled water is maintained at 4.4°C and the hot water at 40.6°C.

The results for a part of the exercise are summarised here for configuration 1 exercise 1a) and 1b) for the summer day (September 6th). In exercise 1a) constant zone conditions are assumed with full fresh air, whilst in exercise 1b) constant zone conditions are assumed with the fresh air fraction being controlled though the economy cycle. The SPATS system networks for exercises 1a) and 1b) are shown in figure (6.9). The water mass flow rate to each coil is modulated by a proportional controller and a signal processor (ie: a two way valve) for simplicity since the performance of the hydronic systems is excluded from these exercises.

For full fresh outside air the network for system 1a) has an equivalent air terminal pressure drop to replace the mixing box of exercise 1b). Since the zone conditions for these exercises are fixed the performance of the economiser can be analysed separately to

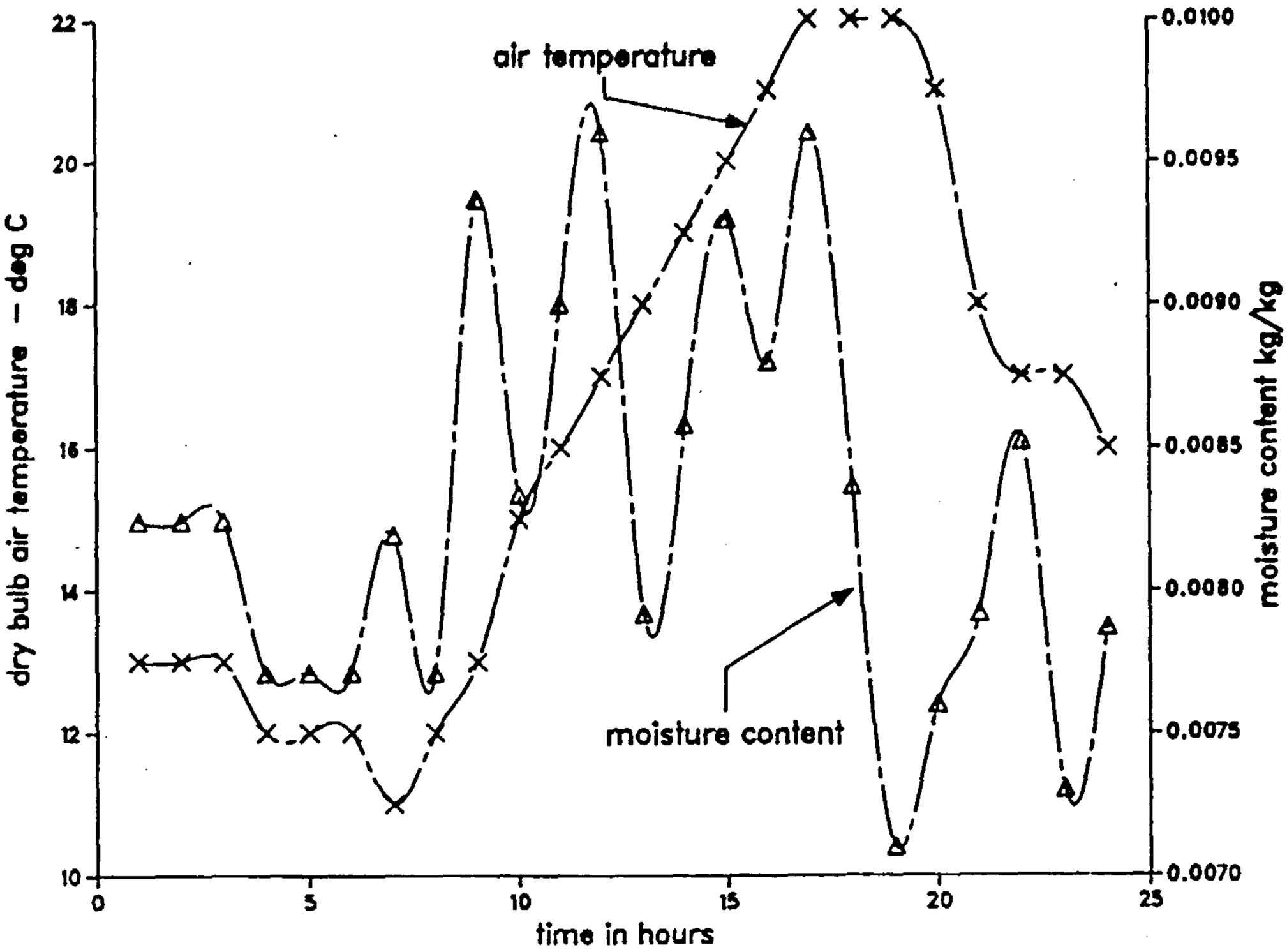


figure 6.7 Weather Data for the Collins Exercise

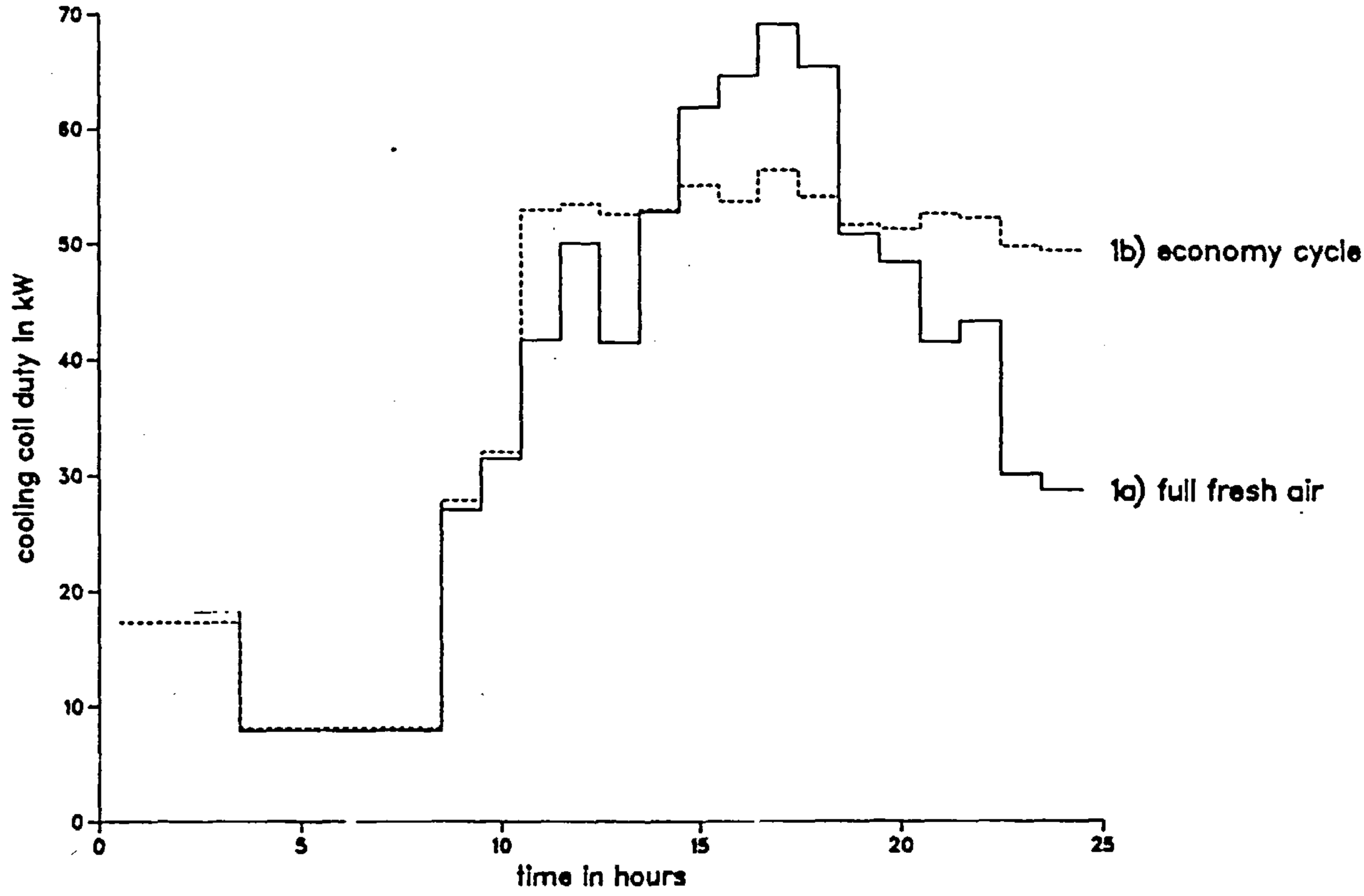


figure 6.8 Comparison of Coil Duties for Collins Exercise 1 a&b

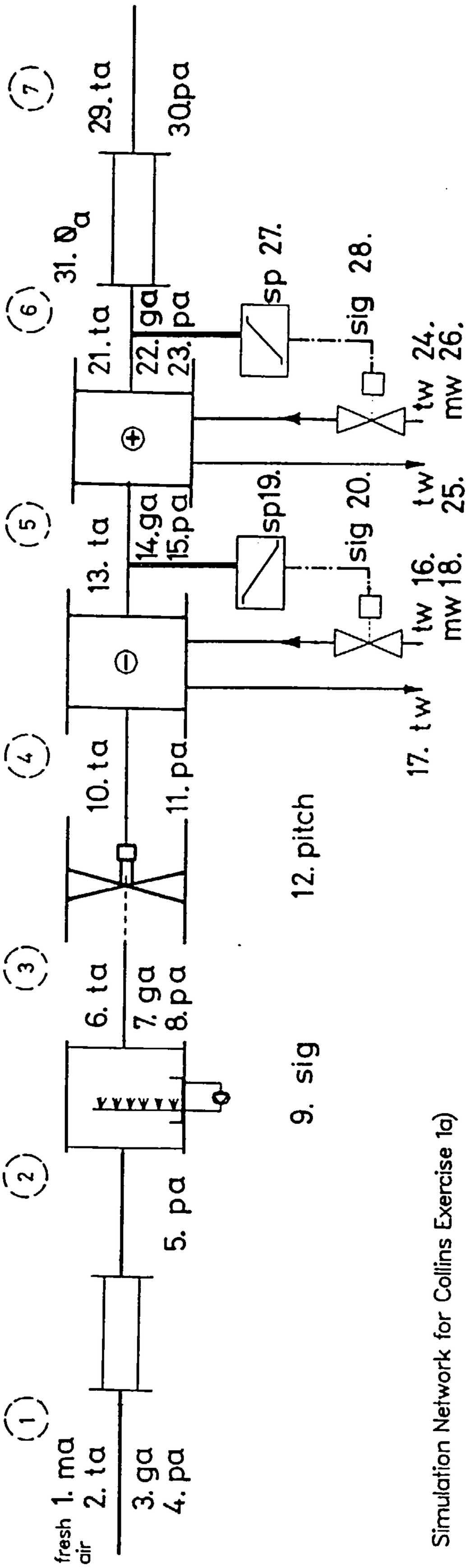


figure 6.9 Simulation Network for Collins Exercise 1a) & 1d)

give exogenous values for the arc-variables ta 2 ga 3 and pa 4

The temperature profiles at various points around the systems are given in figures (6.10) and (6.11) and a comparison of the coil duties for each cycle is given in figure (6.8). Note the sudden rise in air temperature 'on-coil' (system point 2) when the 'economy' cycle comes in as the outside air temperature rises to 16 °C at 1100 hrs. From the comparison of coil duties in figure (6.8) it is apparent that the 'economy' cycle actually increases the refrigeration load for much of the time (ie: between 1100 and 1300 hrs and between 1900 and 2400 hrs) when more 'free-cooling' could be utilised. However the economy cycle does enable the air flow temperature to be maintained within the set point for the whole cycle (figure 6.11) whereas without the economiser the set point 'off-coil' is not maintained at the high air enthalpy which occurs at 1700 hrs. in exercise 1a) (figure 6.12).

Preliminary interpretation of these results would indicate that the set point for the economiser to return to minimum fresh air is too low. Determination of the most appropriate set point would require the simulation to include a model of the the zone but a value in line with the zone set point (21°C) minus the expected temperature rise across the fan (say 1.5°C) and the heat pickup in the distribution ductwork (say 2°C), ie: 18.5°C would be more appropriate.

Although the heat pickup in the supply air seems excessive (approximately 8.7 kW) most of the ductwork runs uninsulated through the ceiling void.

This description serves as an example of the application of SPATS to part of the Annex 10 simulation exercises: a presentation of the full results of the exercises for configuration 1 a) - d) and configuration 2 a) - d) for a summer and a winter day, ie: 16 daily 24 hour profiles, would be inappropriate here.

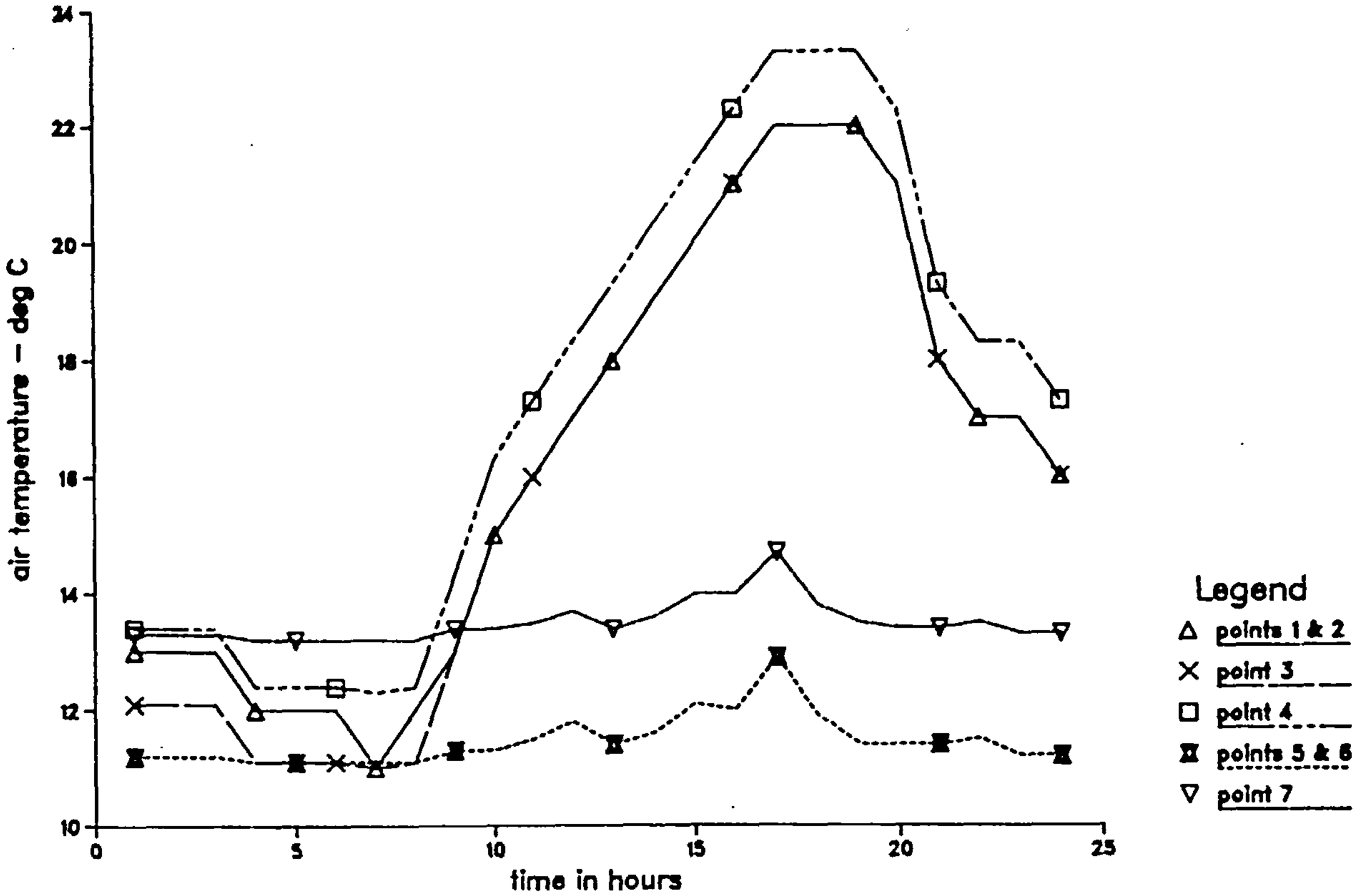


figure 6.10 Temperature Profiles for Collins Exercise 1a

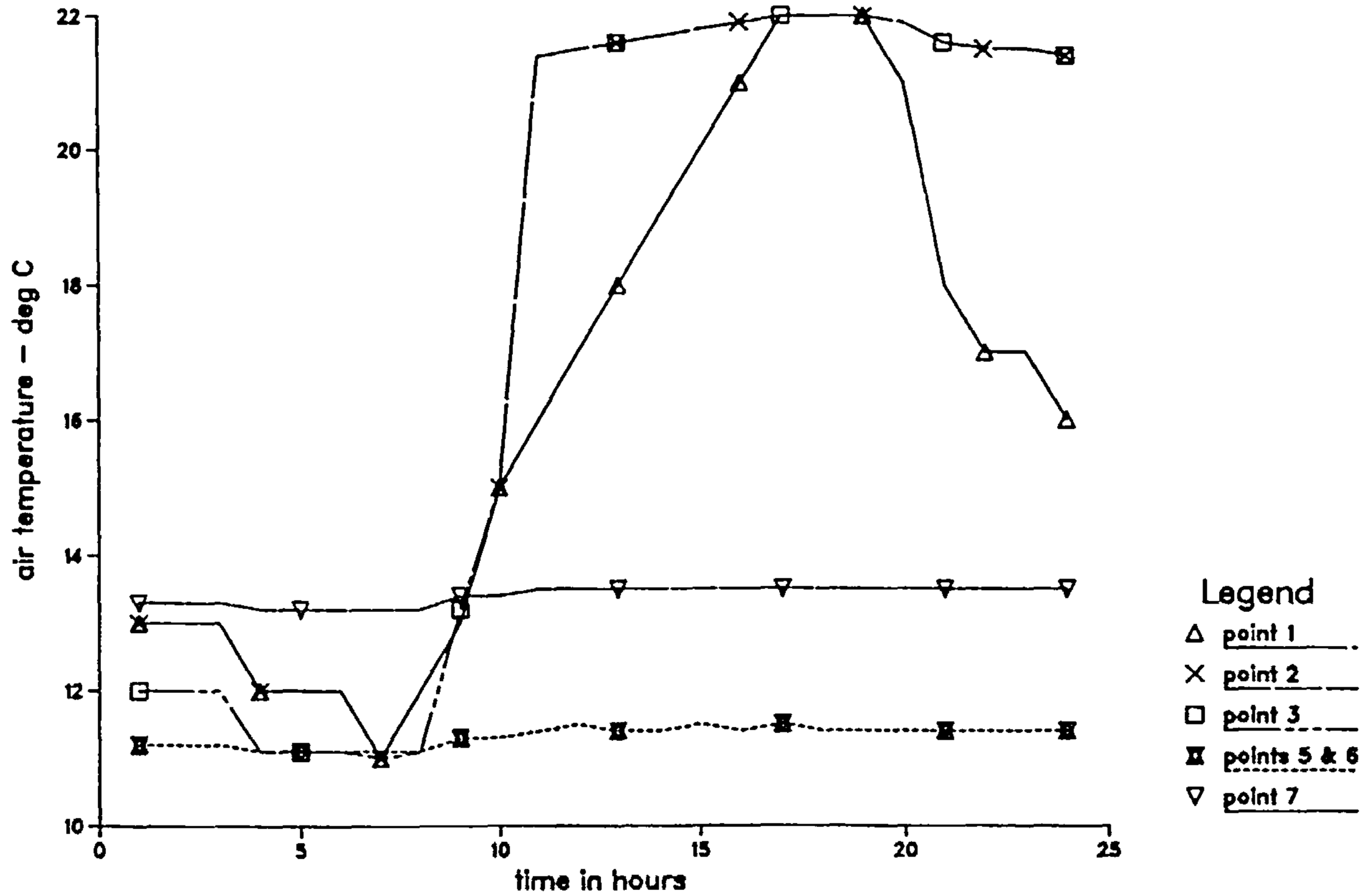


figure 6.11 Temperature Profiles for Collins Exercise 1b

```
Period 1  - 17-00
Solution obtained from GRG2
results for collins1 simulation    dated : 25/10/84  by : peter

node no.    data record


   1          fitting                    pressure drop =  25.892 N/m**2
   2          hum1
                        dry bulb off              =    22.0000 'C
                        moisture content off      =  0.008347 kg/kg
                        static pressure off       =  -66.1266 n/m2
                        pressure difference           =    40.2347 Pa
                        wet bulb temp off         =    15.6250 'C

   3          collins              Absorbed power =    5.5160 KW
                                 Fan total pressure =  1107.6097 Pa
                               Fan total efficiency =    71.0259 %

   4          5/8t04                        coil duty =   -65.3840 kW
                                  sensible heat ratio =    0.7415
                                        pressure drop =    29.5857 N/m**2

   7          5/8t02                        coil duty =    0.0000 kW
                                  sensible heat ratio =    1.0000
                                        pressure drop =    11.8974 N/m**2

   8          cont1
              WARNING - variable outside range -    11.8460

  10          duct22                pressure drop =   133.7463 N/m**2
                                          heat loss =    -8.3603 kW
```

```
Arcvariables :-
No.  1     ma         =     4.1730     No. 16     twat-on   =     4.4000
No.  2     tair       =    22.0000     No. 17     twat-off  =     8.8869
No.  3     ga         =     0.0084     No. 18     mw        =     3.4687
No.  4     pa-inlet   =     0.0000     No. 19     setpoint  =    11.5000
No.  5     pa-out     =   -25.8920     No. 20     signal    =     0.8460
No.  6     db-off     =    22.0000     No. 21     tair-off  =    11.8460
No.  7     ga-off     =     0.0083     No. 22     ga-off    =     0.0068
No.  8     pa-off     =   -66.1266     No. 23     pa-off    =  1000.0000
No.  9     signal     =     0.0000     No. 24     twat-on   =    40.6000
No. 10     ta-out     =    23.3218     No. 25     twat-off  =    10.9784
No. 11     pa-out     =  1041.4831     No. 26     mw        =     0.0000
No. 12     bldtangl   =    23.6059     No. 27     setpoint  =    10.5000
No. 13     tair-off   =    11.8460     No. 28     signal    =     1.8460
No. 14     ga-off     =     0.0068     No. 29     tair-out  =    13.8262
No. 15     pa-off     =  1011.8974     No. 30     pair-out  =   866.2537
                                       No. 31     tambient  =    22.0000
```

figure 6.12 Operating Point of Collins Exercise 1a) at 1700hrs Sep 6th

6.3 Use of SPATS in System Fault Diagnosis

The solution algorithms used by spats do not always find a feasible operating point of a system, however experience of using SPATS has developed an insight to the possible causes of failure. The most common cause of failure is an ill-defined system of equations, either because the component equations are badly formulated or because an infeasible system has been specified. The version of the Newton Raphson algorithm currently in use does not use scaled values of the system variables and fails to find an operating point when the numerical values of the different variables are markedy dissimilar.

The imposition of bounds on the variables, which are automatically specified from a component data record, can be problematical where a component data record may be used for a wide range of systems since a variable bound, and hence the scaling factor, may be appropriate for one network but not for another which has much larger (or smaller) components. As a compromise the current version of SPATS accepts the narrowest bounds for each variable from any of the data records to which it is associated, on the premise that if the solution algorithm fails with a variable set at its bound it is easier to spot and reset the bound or alter the size of the component, than it is to diagnose the cause of the solution algorithm failing due to poor scaling of one of the variables.

The full potential of the use of SPATS to diagnose poor system specification or poorly defined component models cannot be realised until a more suitable solution algorithm is developed to suit the problems encountered in HVAC system simulation. Using the present solution algorithms can make it difficult to differentiate between numerical problems inherent in the algorithms and problems arising solely from the system definition.

Chapter 7. CONCLUSIONS AND FURTHER DEVELOPMENT

The primary conclusions of the research presented in this thesis are: that it is feasible to devlop a component based steady state simulation procedure for HVAC systems, and that the practical application of this procedure will require the development of an effective optimisation algorithm for the solution of the specific non-linear and possibly discontinuous problems arising in the analysis of HVAC systems and components.

7.1 Simulation Methodology

The simulation methodology developed in this thesis includes the simultaneous solution of all the component performance relationships using optimisation techniques. A system may be described as a network of components connected together by system variables or arc-variables, which yields a set of equations in a number of variables. This method of specifying and analysing a HVAC system overcomes several of the problems reported by other researchers in this field.

Simultaneous solution of system equations avoids the problems of setting up stable generalized iterative procedures for the sequential analysis of systems. Sequential techniques are difficult to apply to either multi-fluid-loop systems or to systems which include recirculation and feedback. Where a system has more than one fluid loop a solution technique is required which iterates within each fluid loop independently and then iterates between the different loops to obtain a solution. This is both difficult to program in a generalized, way suitable for component based system specification, and is expensive in terms of computer resources. Convergence of iterative procedures cannot always be guaranteed, hence proposals have been made for refined sequential methods which avoid the need for iteration (Stoecker 1976). One of these methods is to use the values of recirculating variables from the previous time period, as used by Irving (Irving 1982), which assumes that, provided the time step is sufficiently short, the error introduced is negligible in comparison to other assumptions.

The procedure developed by Sowell (Sowell 1984) of reducing the essentially simultaneous problem to a sequential one, represents a fundamental departure from the objective of a versatile component based simulation. Generalised mathematical mapping theory is used to develop a deterministic sequential solution algorithm for a particular combination of components. Although there is flexibility in system definition, a separate solution algorithm must be generated, albeit automatically within a computer program, for each system.

Specification of control systems for steady state simulation techniques have been reported as problematical due to the discontinuous nature of control functions, although the concept of 'ideal' or 'perfect' control can bypass these problems. The technique developed in this thesis, of solving a set of equations in a number of system variables, enables the variables to represent any value within a system, whether a physical variable, such as mass flow rate, or a 'pseudo' variable, such as controller output or heat transfer. The variables are related only via the component equations, and not as a fluid or control 'state' (Miller 1982, Silverman 1981), which means that control equipment can be specified as an integral part of the network definition of a system, in the same way as any other component.

7.2 Solution Algorithms

Experience with applying some of the most commonly available optimisation algorithms suggests that the analysis of HVAC systems requires the development of a specific optimisation method to suite the particular form of the problems presented by HVAC systems. A solution method which makes use of the 'sum of terms' form of the residual equations, such as Least Squares or Newton Raphson, combined with linear bounds on the system variables may be most appropriate. Development of an 'ideal' non-linearly constrained non-linear solution algorithm as outlined in section (4.4) would be a major undertaking.

The reliance on the GRG2 algorithm to solve the residual equations for all but the simplest systems can be problematical. As the complexity of the systems increases so does the number of constraints to be satisfied at each iteration and the solution time can be appreciable even on a powerful mainframe computer. This computation time could affect the application of the method as a design tool, although is not envisaged as being used primarily for hour by hour, year-round computation but rather to solve for a few specific periods to analyse typical off peak operation of a system.

Although the GRG2 algorithm is a simultaneous method, the solution is approached via a sequence of satisfied and violated constraints, the order of which sequence is highly dependent upon the relative scaling of both variables and residual functions. It is apparent that in certain cases the algorithm fails to find a feasible operating point for a system because the order of satisfying the constraints is such that once satisfied a constraint may not re-enter the working set to enable other constraints to be solved and for the solution process to proceed in another direction.

Immediate progress may be made however on two techniques, the Newton Raphson algorithm and the Least Squares algorithm. The Newton Raphson technique requires the imposition of appropriate bounds on the variables within the basic algorithm and as a suitable means of scaling the variables. The Least Squares technique also requires simple bounds on the variables which could be approached rigorously by either developing an entirely new computer program based on a published algorithm or altering existing codes to incorporate bounds on the linear search: alternatively pseudo bounds could be placed on the variables within the calculation of the residual functions. This latter approach undoubtably alters the shape of the residual function 'surface' being minimised and changes the calculated search direction but may work in practice.

7.3 Component Model Development

The development and refinement of a library of component models is a major task in two phases:

i) component algorithms must be developed based upon a rigorous analysis of the performance of components and of the different techniques of predicting that performance: this will depend upon the cooperation and exchange of ideas between research groups both within Europe, through I.E.A. Annex 10, and worldwide through such forums as ASHRAE algorithm publications (Low 1982).

ii) To enable the library of algorithms to be used effectively in system simulation a data base of manufacturers' performance data must be compiled. Through the development of system simulation and its acceptance as an aid in the system design process manufacturers must be persuaded to publish more of the data which they already have and to carry out testing procedures to an agreed standard, publishing the results in a common format.

7.4 Further Applications

Besides the application to analysis of HVAC systems the basic methodology outlined here could be applied to the analysis of any system of components which can be related in a network with describing equations written in residual form.

An application related to the thermodynamic performance of HVAC systems is the analysis of sound transmission and generation in ducted systems, where the ability to solve all the acoustical relationships for a system simultaneously would have a fundamental advantage over current sequential techniques.

The residual form of the component equations and their implicit solution by using optimisation techniques could be used to develop an equipment selection program. By cataloguing manufacturers performance data and specifying any combination of the system variables as exogenous it would be possible to compare equipment from different manufacturers and to check the performance of equipment at other than design conditions. Since the equations for most

individual components can be solved using the Newton Raphson technique this program could be implemented on a micro-computer.

A further application which has already been taken up and is currently under development is in the analysis of fluid flow in water treatment works (Mortimer 1984). For this application an early micro-computer version of SPATS has been altered and translated to FORTRAN77 to run on a 16-bit micro-computer. The number of different system variables is limited (ie: water flow rate and static pressure head or depth) and the residual equations for the components are less complex than many of those which appear in HVAC system analysis although they do have non-linear properties. From preliminary investigations it appears that the residual equations may be solved using the Newton Raphson algorithm.

REFERENCES

ABACUS 1984 — ABACUS - Summary Interim Report for Grant GR/C/2283, 'Major Extensions to the ESP System Oct. 82 - Sep. 84', Dept. of Architecture and Building Science, University of Strathclyde.

ANSI 1978 — 'American National Standard Programming Language FORTRAN, X3.9 - 1978', American National Standards Institute.

Arumi-noé 1979 — Arumi-noé, F., and Northrup. D. 'A Field Validation of the Thermal Performance of a Passively Heated Building as Simulated by the DEROB System', Energy and Buildings, 2(1979), pp 65-75.

ASHRAE 1975 — Stoecker. W. (editor), 'Procedures for Simulating the Performance of Components and Systems for Energy Calculations', ASHRAE. New York. 1975.

Borreson 1981 — Borreson. B.A., 'HVAC Control Process Simulation', ASHRAE Trans., V87 Pt 2, 1981, pp 871-882.

Benton 1982 — Benton. R., MacArthur. J.W., Makesh. J.K. and Cockroft. J.P., 'Generalised Modeling and Simulation Software Tools for Building Systems', ASHRAE Trans., V88 Pt 2, 1982, pp 839-856.

Gill 1981 — Gill. P.E., Murray. W., Wright. M.H., 'Practical Optimisation', Academic Press, London 1981.

Hanby 1984 — Hanby. V.I., Private Communication (ref: I.E.A. Annex 10 'Le Chaumier' Exercise), Dept. Civil Engineering, University of Technology, Loughborough.

Irving 1982 — Irving. S.J., and Quick. J.P., 'Simultaneous Solution of Room Response and Plant Performance', System Simulation in Buildings, Proceedings of the International Conference, Liege, Belgium, Dec. 1982.

Kusuda 1981 — Kusuda. T., 'Standards Criteria for HVAC Systems and Equipment Performance Simulation Procedures', ASHRAE J., Oct. 1981, pp 25-28.

Lasdon 1978 — Lasdon. L.S., Warren. A.D., Jain. A., and Ratner. M., 'Design and Testing of a Generalised Reduced Gradient Code for Non-linear Programming', ACM Trans. on Mathematical Software, V4/1, March 1978, pp 31-50.

Lasdon 1982 — Lasdon. L.S., and Waren. A.D., 'GRG2 User's Guide', Dept. of General Business, School of Business Administration and Dept. of Mechanical Engineering, University of Texas at Austin, Austin, Texas 78712, U.S.A., May 1982.

Loudon 1968 — Loudon. A.G., 'Summertime Temperatures in Buildings', IHVE/BRS Symposium, Feb. 1968, B.R.S. Current Paper 47/68.

Low 1982 — Low. D.W., and Sowell. E.F., 'A Plan for Establishing Computational Algorithms as ASHRAE Literature', ASHRAE Trans., V88 Pt 2, 1982.

M^{c}Nally 1976 — M^{c}Nally. J.T., 'A Commentary on the State of the Art', Heating Piping Air-conditioning, April 1976, p 73.

Marchant 1979 Marchant. S.J., 'Dynamic Modelling of Boilers', Technical Memorandum No. 68, Polytechnic of the South Bank. Institute of Environmental Science and Technology, Oct. 1979.

Miller 1982 Miller. D.E., 'A Simulation to Study HVAC Process Dynamics', ASHRAE Trans., V88 Pt 2, 1982, pp 809-825.

Mortimer 1984 Mortimer. G.H., Private Communiction (ref: 'The Computerised Hydraulic Analysis of Water Reclamation Works' for the Severn Trent Water Authority), Dept. of Civil Engineering, University of Technology, Loughborough.

Murray 1982 Murray. M.A.P., 'Component Based Simulation of HVAC Systems - First Interim Research Report', Dept. Civil Engineering, University of Technology, Loughborough, July 1982.

Nall 1983 Nall. D.H., and Crawley. D.B., 'Energy Simulation in the Building Design Process', ASHRAE J., V25 No. 11, Nov. 1983, pp 28-32.

NAG Numerical Algorithms Group, Mayfield House, 256 Banbury Road, Oxford OX2 7DE.
1 - E04CCF Simplex Algorithm.
2 - F03AFF and F04AJF Crout's Decomposition.
3 - E04JBF Quasi-Newton Algorithm.
4 - E04FCF Least Squares Algorithm.

NESC 1981 National Energy Software Centre, 'NESC Abstract No. 782 - DOE2', U.S.A..

Quick 1982 Quick. J.P., Irving. S.J., 'Computer Simulation for Predicting Building Energy Use', C.I.B. Symposium. March 1982 (Dublin).

Silverman 1981 Silverman. G.J., Jurovics. S.A., Low. D.W., and Sowell. E.F., 'Modeling and Optimisation of HVAC Systems Using Network Concepts', ASHRAE Trans. V87 Pt 2, 1981.

Sowell 1984 Sowell. E.F., Taghavi. K., Levy. H., and Low. D.W., 'Generation of Building Energy System Models', ASHRAE Trans., V90 Pt 1, 1984.

Stoecker 1976 Stoecker. W.F., Fenske. R.P., and Witkin. C.C., 'The Accuracy - Complexity Compromise in Simulating Systems for Energy Calculations', ASHRAE Research Report No. 2290 RP. 131, 1976.

Sussock 1978 Sussock. H., 'Simulation - almost as good as the real thing', Energy Manager, June 1978, pp 19-21.

TRNSYS 1983 Solar Energy Laboratory, University of Wisconsin, Wisconsin, U.S.A..

Wright 1984 a) Wright. J.A., 'Modelling of HVAC Components by the Least Squares Curve Fitting Method', Dept. Civil Engineering, University of Technology, Loughborough, May 1984.

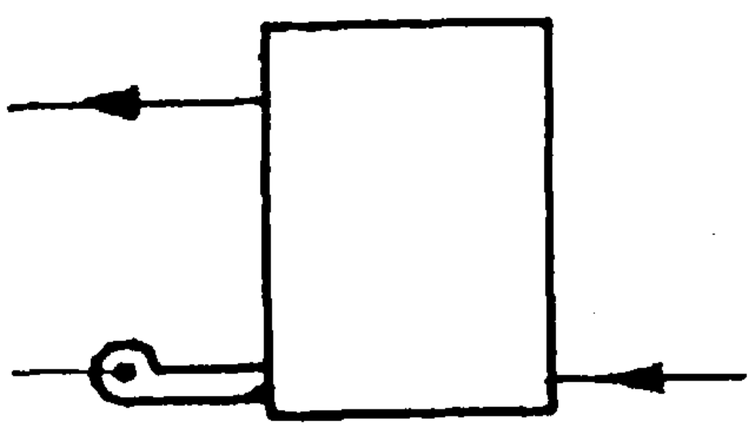
b) Wright. J.A., and Hanby. V.I., 'HVAC Component Specification - Fans', IEA Annex 10 Document AN10 840906 02, L.P.B. University of Liege, Belgium.

APPENDIX A

Component Models

The form of the component models used in the examples are given in this appendix together with an acknowledgement to the author of each model which appears at the bottom right hand corner of each section. All the models used have been developed within the Department of Civil Engineering, University of Technology, Loughborough.

A-1 BOILERS



Describing Equation

(i) $Q_{load} / (\eta \cdot \eta_{max}) - Q_{fuel}$

System Variables

t_i	water inlet temperature
t_e	water exit temperature
m_w	water mass flow rate
Q_{fuel}	fuel input rate

Local Variables

Q_{load}	=	$m_w \cdot Cp \cdot (t_e - t_i)$	(boiler load)
γ	=	Q_{load} / L	(% of maximum load)
η	=	polyxz(γ , t_i , 1)	(normalised efficiency)

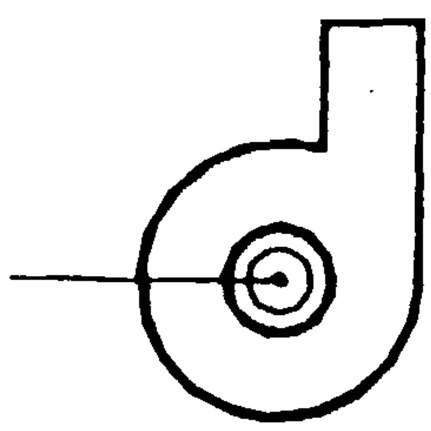
Constants

L	maximum rated load
η_{max}	% eficiency at maximum rated load

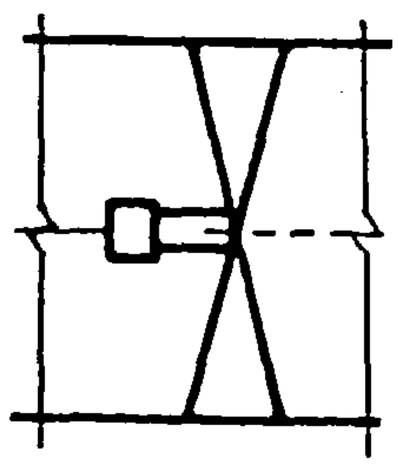
polynomial coefficients for η

(M.A.P.Murray)

A-2 FANS (Wright 1984 b.)



centrifugal



axial

Describing Equations

(i) $P_{tf} = (P_o - P_i)$

(ii) $Q_r = M \cdot C_p (T_o - T_i)$

Internal Equations

(i) $\omega = V_i / (\pi^2 . d^3 . n/4)$

(ii) $\tau_t = \int_1 = \text{polyxz}(\omega , \beta , 1)$

(iii) $\lambda = \int_2 = \text{polyxz}(\omega , \beta , 2)$

(iv) $P_{tf} = [\tau_t . \rho_i . (\pi . d . n)^2] / 2$

(v) $Q_r = [\lambda . \pi^4 . d^5 . n^3 . \rho_i] / 8$

System Variables

M	air mass flow rate
P_i	total pressure at inlet to fan
P_o	total pressur at outlet
T_i	air temperature at inlet
T_o	air temperature at outlet
g_i	air moisture content
β	fan blade angle (NB : axial fans only)
n	fan speed (NB : exogenous constant for axial fans)

Local Variables

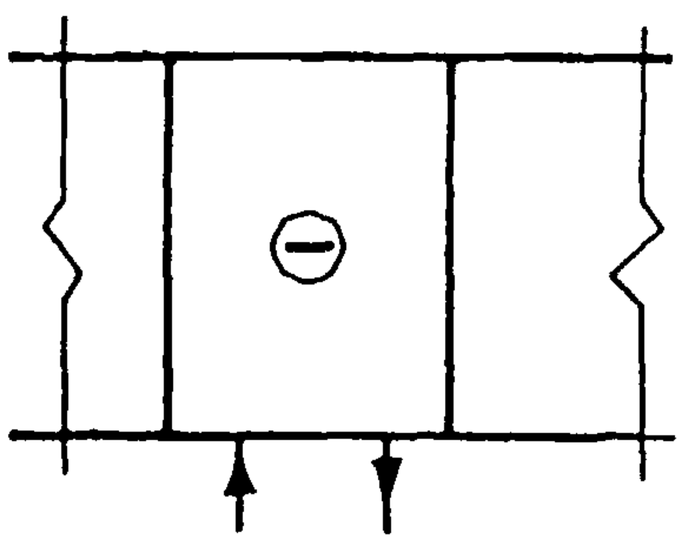
V_i	volume flow rate at inlet
ω	normalised volume flowrate
τ_t	normalised total pressure
λ	normalised absorbed power
P_{tf}	fan total pressure
ρ	air density at inlet
Q_r	fan absorbed power
C_p	specific heat

Constants

d	fan diameter
polynomial coeficients for τ_t and λ	

(J.A.Wright)

A-3 HEATING/COOLING COIL



Describing Equations

(i) $ma(h_2 - h_1) = c_{min}.eff.(tw_1 - ta_1)$ (rate equation)

(ii) $ma(h_2 - h_1) = cw(tw_1 - tw_2)$ (enthalpy balance)

(iii) $g_1 - g_2 = (1 - shr)(t_1 - t_2)/(2400.shr)$ (moisture balance)

(iv) $p_1 - p_2 = (g^2.v.f)/(2f1fa)$ (pressure drop)

System Variables

ma	air mass flow rate
ta_1	air on dry bulb temperature
ta_2	air off dry bulb temperature
g_1	air on moisture content
g_2	air off moisture content
p_1	air on pressure
p_2	air off pressure
mw	water mass flow rate
tw_1	water on temperature
tw_2	water off temperature

Local Variables

c_{min}	minimum fluid capacity rate
cw	water side capacity rate
eff	coil effectiveness (internally computed)
f	friction factor (internally computed)
g	mass velocity of air
h_1	entering air enthalpy
h_2	exit air enthalpy
shr	sensible heat ratio (internally computed)
v	specific volume of air

Exogenous Constants

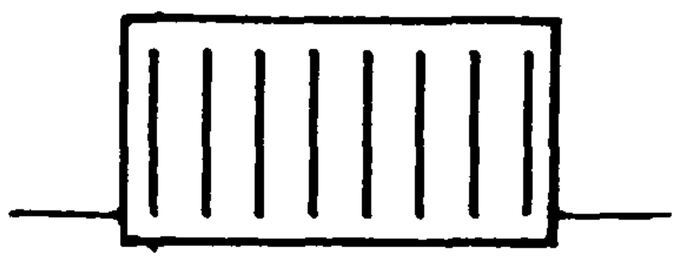
fa	face area of coil
ncirc	number of water circuits

Constants

r_{met}	coil metal thermal resistance
faaa	face area/air side area
f1fa	free flow area/air side area
c_2	Colburn factor constant
f_2	friction factor constant

(V.I.Hanby)

A-4 RADIATORS



Describing Equations

(i) $q = m.C_p.(t_i - t_e)$

(ii) $q = \int.CAPR.h^{.82}.LMTD(t_i, t_e, t_a)^{1.3}$

Local Equations

(i) $m_f = m / (m_r . 1)$ (fraction rated flow)

(ii) $\int = 1 - \exp(-2.5 . m_f)$ (flow factor for output at reduced load)

System Variables

t_i inlet water temperature

t_e outlet water temperature

t_a room air temperature

m water mass flow rate

q radiator heat output

Local Variables

LMTD log mean temperature difference

Exogenous Constant

l radiator length

Constants

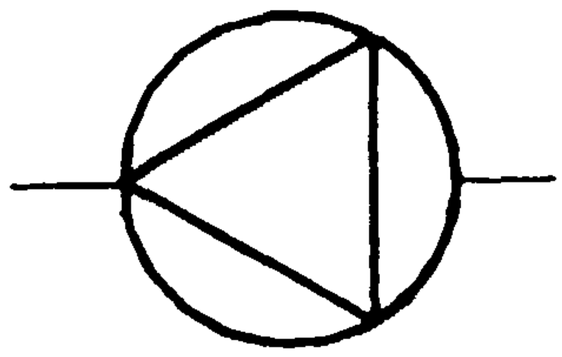
h radiator height

CAPR capacity rate of radiator (per metre run)

m_r rated flow per metre at 80/70 flow and return $t_a = 18^oC$

(M.A.P.Murray)

A-5 COMPRESSORS



Describing Equations

(i) $Q_e = \int_1(t_e , t_c)$

(ii) $W = \int_2(t_e , t_c)$

(iii) $Q_c = Q_e + W$

System Variables

Q_e refrigeration effect

Q_c heat rejection

W compressor work input

t_e evaporating temperature

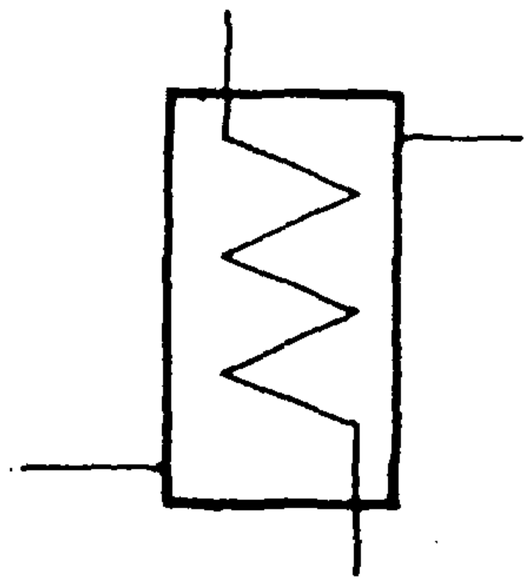
T_c condensing temperature

Constants

polynomial coefficients for Q_e and W

(M.A.P.Murray)

A-6 HEAT EXCHANGER



Describing Equation

$$Q = \varepsilon \cdot UA \cdot (t_r - t_f)$$

System Variables

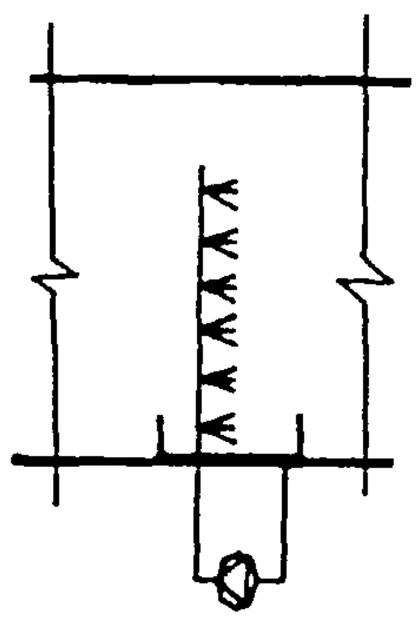
Q	heat transfer rate
t_r	refigerant temperature
t_f	fluid temperature

Constants

ε	exchanger effectiveness
UA	heat transfer coefficient

(M.A.P.Murray)

A-7 HUMIDIFIER



Describing Equations

(i) $eff = x.(g_2 - g_1)/(g_3 - g_1)$ (moisture balance)

(ii) $eff = x.(h_2 - h_1)/(h_3 - h_1)$ (enthalpy balance)

(iii) $p_2 - p_1 = k_c.Q^2$ (pressure drop)

System Variables

ma	air mass flow rate
ta_1	entering air dry bulb temperature
g_1	entering air moisture content
p_1	entering air pressure
ta_2	exit air dry bulb temperature
g_2	exit air moisture content
p_2	exit air pressure
g_3	air moisture content at saturation
x	on/off control signal

Local Variables

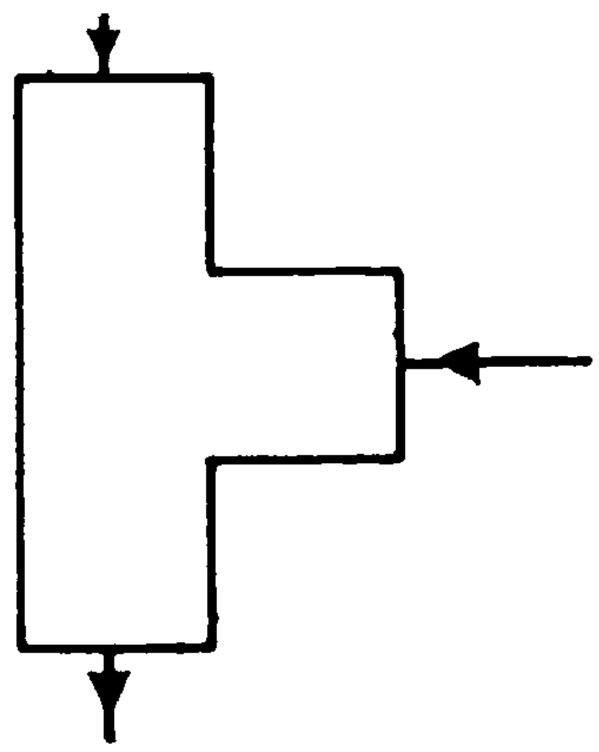
h_1	entering air enthalpy
h_2	exit air enthalpy
h_3	air enthalpy at saturation
Q	air volume flow rate

Constants

eff	effectiveness
k_c	pressure loss coefficient

(P.R.Deering)

A-8 MIXING TEE



Describing Equations

$$m_f \cdot t_f = m_2 \cdot t_2 + m_3 \cdot t_3 \quad (* \text{ 10 scaling factor})$$

$$m_f = m_2 + m_3 \quad (* \text{ 1000 scaling factor})$$

System Variables

m_f mass flowrate in full flow leg

t_f water temperature in full flow leg

m_2 mass flow rate in leg 2

t_2 water temperature in leg 2

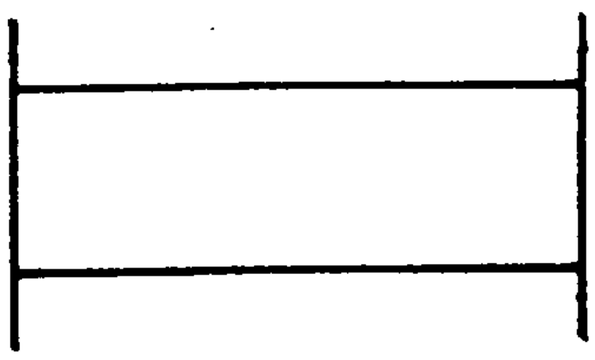
m_3 mass flow rate in leg 3

t_3 water temperature in leg 3

No Constants

(M.A.P.Murray)

A-9 INSULATED ROUND DUCT



Describing Equations

(i) $t_2 = t_1 + (t_{amb} - t_1)(1 - \exp(-UA/C_a))$ (heat transfer)

$$\frac{1}{\sqrt{f}} = -4 \log_{10}\left[\frac{k_s}{3.7d} + \frac{1.255}{N_{Re}\sqrt{f}}\right]$$ (solved locally by iteration)

(ii) $p_1 - p_2 = (2fv^2 l)/d$ (pressure loss)

System Variables

m_a	air mass flow rate
g_a	air moisture content
t_1	entering air temperature
p_1	entering air pressure
t_2	exit air temperature
p_2	exit air pressure
t_{amb}	ambient air temperature

Local Variables

A	surface area of inside of duct
C_a	air thermal capacity rate
f	friction factor
N_{Re}	Reynolds number of air flow in duct
U	overall heat transfer coefficient (locally computed)
v	air velocity in duct
	density of air

Exogenous Constants

d	duct diameter
l	length of duct
x	insulation thickness

Constants

k	thermal conductivity of insulation
k_s	absolute roughness of duct wall
h_{so}	exterior heat transfer coefficient

(V.I.Hanby)

A-10 ZONE

Describing Equations

(i) $UA \cdot (t_{ao} - t_{room}) = \sum q_i \ (i=1,n)$

System Variables

t_{ao} outside air temperature

t_{room} air temperature in zone

+ n multiple 'heat inputs' (eg: radiator output)

Exogenous Constants

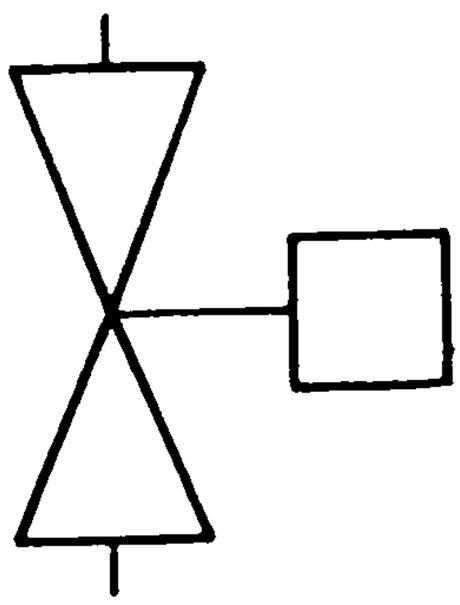
UA overall heat transfer coefficient for zone

n number of multiple heat inputs

(M.A.P.Murray)

A-11 CONTROL VALVES

a) Two way Modulating Valve



Describing Equation

$$m = (Q_0 - Q_1) \cdot x + Q_0$$

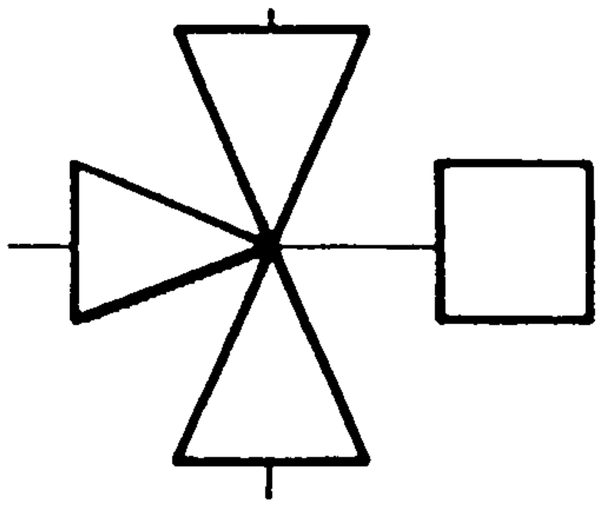
System Variables

m	flow rate
x	controller signal

Constants

Q_0	flow rate at x=0 (Q_{max} for heating, Q_{min} for cooling)
Q_1	flow rate at x=1 (Q_{min} for heating, Q_{max} for cooling)

b) Three way Diverting Valve



Describing Equation

(i) $$m_{in} = (1 - x) \cdot m_{max} \quad (* 1000 \text{ scaling factor})$$

(ii) $$m_{max} \cdot t_{mixed} = m_{in} \cdot t_{ret} + m_{max} \cdot x \cdot t_i \quad (* 10 \text{ scaling factor})$$

System Variables

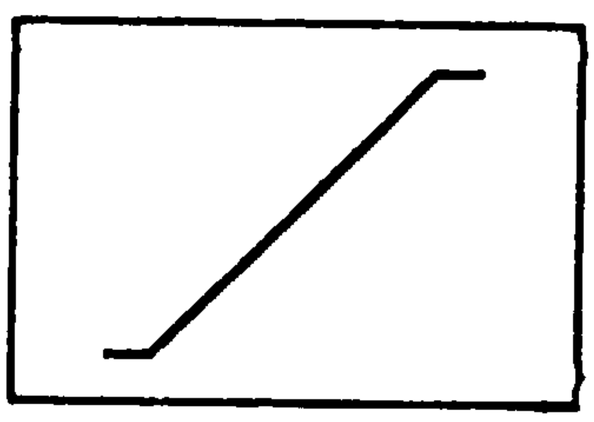
x	controller signal
m_{in}	input mass flow rate
t_i	supply water temperature
t_{ret}	return water temperature
t_{mixed}	mixed water temperature
m_{max}	maximum mass flow rate

No Constants

(M.A.P.Murray)

A-12 CONTROLS

a) Proportional Controller



Describing Equation

$$x = (cv - sp + tr/2)/tr$$

System Variables

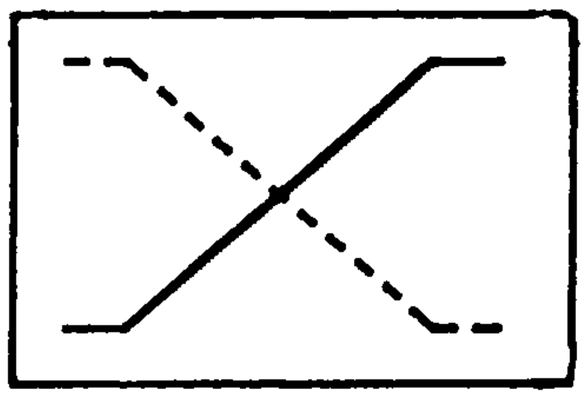
cv	controlled variable
sp	set point
x	output signal

Exogenous Constant

tr	throttling range (symmetrically placed about set point)

(V.I.Hanby)

b) Signal Inverter



Describing Equations

(i) if x_1 0 $x_2 = c$

(ii) if x_1 1 $x_2 = m + c$

(iii) else $x_2 = mx + c$

System Variables

x_1	input signal
x_2	output signal

Exogenous Constants

c	offset
m	gradient

(V.I.Hanby)

APPENDIX B

Development of a Newton-Raphson Minimisation Algorithm

The Newton-Raphson algorithm uses an approximation to the function based upon the Taylor series expansion about a trial point $\underline{x}_k$:-

$$f(\underline{x}_{k+1}) = f(\underline{x}_k) + J_k * (\underline{x}_{k+1} - \underline{x}_k) + {}^{1}/_{2} (\underline{x}_{k+1} - \underline{x}_k)^2 * H_k * (\underline{x}_{k+1} - \underline{x}_k)$$

This can be used to develop an iterative scheme using both the Jacobian and the Hessian matrices to predict a new value of $\underline{x}$. The method takes advantage of the fact that the contours near a local minimum are nearly elliptical (or the n^{th} dimensional analogue) and this quadratic property ensures rapid convergence on the minimum.

However when $\underline{x}$ is near the minimum the higher order terms in the Taylor expansion become negligible and a first-order Newton-Raphson method can be developed (Gill 1981 p140) :-

$$f(\underline{x}_{k+1}) = f(\underline{x}_k) + J_k * (\underline{x}_{k+1} - \underline{x}_k)$$

but at the minimum $f(\underline{x}) = 0$ hence :-

$$f(\underline{x}_k) + J_k * (\underline{x}_{k+1} - \underline{x}_k) = 0$$

which gives the iterative scheme :-

$$\underline{x}_{k+1} = \underline{x}_k - J_k^{-1} * f(\underline{x}_k)$$

the method is thus reduced to obtaining correction terms $\Delta\underline{x}$ by solving $J_k * \Delta\underline{x} = f(\underline{x}_k)$.

Since the method is iterative the loss in accuracy in abandoning the higher order terms of the Taylor expansion is insignificant although the advantage of quadratic termination is lost and a reduction in the speed of convergence would be expected over the second order method. However the simplification of the method by not requiring computation of the Hessian, which is subject to numerical limitations, far outweighs this loss of speed.

The solution of $J_k * \Delta\underline{x} = f(\underline{x}_k)$ may be obtained using standard matrix methods for the solution of simultaneous equations. Two methods were used for comparative analysis :- Gaussian elimination with partial

pivoting and Crout's decomposition with iterative reduction of round-off and truncation errors.

Gaussian elimination operates by successively eliminating variables from equations by performing row operations and subtracting a multiple of one row from another, until one of the variables is explicitly obtained which is then used in back substitution to obtain the remaining variables (Gill 1981 p33). Partial pivoting is a technique used to minimise the round-off and truncation error. The Gauss elimination process was used as a subroutine written using 'double precision' giving approximately fourteen figure accuracy in computation.

Convergence criteria are difficult to establish since the algorithm tends to oscillate towards the solution. A useful indication of the magnitude of the correction factors $\Delta\underline{x}$ is formed :-

$$PDX = \Delta x_i \; / \; x_i \; * \; 100 \qquad \text{(ie \% change in } x_i\text{)}$$

$$\text{and} \quad SUMDX = SQRT \; \{ (\sum_{i=1}^{n} (PDX)^2) \; / \; n \}$$

In practice sufficient accuracy can be assured by allowing the iterative process to continue until SUMDX $\leq$ XTOL with XTOL = 0.05 giving accuracy to about three significant figures.

The Crout decomposition makes use of the special features of triangular matrices by forming lower and upper triangular matrices L and U such that

$$A = L * U$$

the approximate solution of $A * \underline{x} = \underline{b}$ is then found by forward and backward substitution in

$$L * \underline{y} = \underline{b} \text{ and } U * \underline{x} = \underline{y}$$

The accuracy of the solution is enhanced by forming an iterative process: calculate the residual vector $\underline{r} = \underline{b} - A * \underline{x}$ and obtain a correction $\underline{\delta}$ to the solution by solving $L * U * \underline{\delta} = \underline{r}$, $\underline{x}$ is replaced iteratively by $\underline{x} + \underline{\delta}$ until machine accuracy is obtained.

This Crout decomposition and iterative error reduction algorithm is available as a NAG routine (F04AJF) which uses a special 'extended precision' for the accumulation of inner products thereby reducing truncation error and enhancing accuracy. The convergence criteria used is the same as that for the Gaussian elimination although the algorithm does not suffer from the problem of oscillating towards a solution.

APPENDIX C

Generalised Reduced Gradient Algorithm

Reduced gradient methods of optimisation subject to non-linear constraints use the same ideas as for linearly constrained methods to reduce the objective function whilst maintaining feasibility of a subset of active constraints. The extension of linearly constrained methods to non-linear problems is non-trivial since there is no procedure for ensuring that the sequence of points generated during a search stays on the active constraints.

In general GRG algorithms are not unlike other algorithms in that a search direction $\underline{p}_k$ is formed and a value of α is sought such that $F(\underline{x}_k + \alpha\underline{p}_k)$ is an improved point. However there is no guarantee that $\underline{x}_{k+1} = \underline{x}_k + \alpha\underline{p}_k$ will still be feasible so α is determined from an adaptive sub-problem. A full discussion of reduced gradient type methods is given in (Gill 1981 p220) and details of the GRG2 algorithm in particular are given in (Lasdon 1978).

The GRG2 algorithm solves non-linear problems of the form :-

minimise	$g_{obj}(\underline{x})$		objective function
subject to	$g_i(\underline{x}) = 0$	$i=1,\ neq$	equality constraints
and	$0 \leq g_i(\underline{x}) \leq ub(n+i)$	$i=neq+1,\ m$	range constraints
and	$lb_i \leq x_i \leq ub_i$	$i=1,n$	variable bounds

where **neq** is the number of equality constraints, (**m**-**neq**) is the number of range constraints and **n** is the number of 'natural' variables.

The problem is converted to a pure equality form by extending the set of natural variables to include **m** slack variables, one for each constraint. The problem is then of the form :-

minimise	$g_{obj}(\underline{x})$		
subject to	$g_i(\underline{x}) - x_{n+i} = 0$	$i=1,\ m$	
and	$lb_i \leq x_i \leq ub_i$	$i=1,\ n+m$	

where the slack variable bounds are given by :

equality constraints: $lb_i = ub_i = 0$ i= n+1, n+neq

range constraints: $lb_i = 0$ i= n+neq+1, n+m

At a feasible point $\tilde{\underline{x}}$ where **nb** of the constraints are binding, GRG2 uses the binding constraint equations to solve for **nb** of the natural variables in terms of the (**n**-**nb**) natural variables and the **nb** slack variables associated with the binding constraints. These **nb** natural variables are called <u>basic</u> variables and the (**n**-**nb**) natural variables together with the **nb** slack variables (ie: **n** variables altogether) are called the non-basic variables. The binding constraints can then be expressed as :

$g(\underline{y}, \underline{z}) = 0$

where $\tilde{\underline{x}} = (\underline{y}, \underline{z})$

$\underline{y}$ is the vector of **nb** basic variables

$\underline{z}$ is the vector of **n** non-basic variables

The basic variables must be chosen so that the **nb** * **nb** basis matrix:-

$B = \partial g_i / \partial y_i$

is non-singular at $\tilde{\underline{x}}$ (ie: the equations are independant). The binding constraints are solved for $\underline{y}$ in terms of $\underline{z}$ to give a function $f(\underline{z})$ for all $(\underline{y}, \underline{z})$ near to $\tilde{\underline{x}}$. Thus the objective is reduced to a function of the non-basic variables only :-

$g_{obj}(f(\underline{z}), \underline{z}) = F(\underline{z})$

and the original problem is reduced, near $\tilde{\underline{x}}$ to a simpler form :-

minimise $F(\underline{z})$

subject to $\underline{l} \leq \underline{z} \leq \underline{u}$

where $\underline{l}$ and $\underline{u}$ are bound vectors for $\underline{z}$.

This is the reduced objective and $\Delta F(\underline{z})$ is the reduced gradient which is used to form a search direction $\underline{p}$ and a one dimensional Newton type search in α is used to find a minimum of $F(\underline{z} + \alpha\underline{p})$ for $\alpha > 0$. Initially a conservative step size is taken which is reduced if Newton does not converge at the first iteration. If Newton fails to converge after several iterations the search is terminated at the best point and a new search direction calculated.

The Newton search may continue until an objective is found which is larger than the previous value, in which case the value of α at an approximate minimum is found by passing a quadratic through the three points bracketing the minimum. The reduced problem remains unchanged and a new search direction is calculated from this point.

At the minimum determined by Newton some previously non-binding constraint or basic variable bound may be violated. A new value of α is then determined such that at least one new constraint or variable is at its bound and all the active constraints are satisfied. If the objective is reduced at this point the new constraint is added to the active set and the solution of a new reduced problem commences.

If the search direction is infeasible, either because it is not a descent direction or because any value of α will cause a currently satisfied constraint to be violated, the problem is said to be degenerate. The Kuhn - Tucker conditions for optimality (Gill 1981 p77-82) are checked and if satisfied the algorithm terminates at the optimum. If the conditions are not satisfied the algorithm attempts to form another basis matrix by dropping a basic variable from the basis, which is then known as a superbasic variable, and admitting a new variable to the basis. If this is not possible the algorithm terminates at the current point with a suitable message.

APPENDIX D

List of subroutines used by SPATS

Internal subroutines to 'hvacprog'

sub-routine	source segment	operation:
altfil	DATAFILE	enables a component data record to be edited.
carray	CHANGE	inserts new data record into network arrays.
change	CHANGE	enables a network definition to be altered.
crout	NRSOL	solution subroutine for Newton-Raphson algorithm with Crout's decomposition.
delfil	DATAFILE	enables a component data record to be deleted.
direct	HVACLIB	reads a component directory file for the record position.
disfil	HVACLIB	opens a data directory file or writes a new directory
displa	HVACLIB	displays the network indexing arrays.
execall	EXECALL	calls the executive subroutines to calculate residuals.
filnet	HVACLIB	writes network data arrays to file 'title'.net.
fjacob	NRSOL	forms Jacobian matrix of partial differentials of residuals with respect to the system variables (base on NAG routine E04FCX)
fread	HVACLIB	reads a data record from a directory file.
funct	NRSOL	forms residual vector FVEC to Newton Raphson algorithm.
fwrite	HVACLIB	writes a data record to a directory file.
gcomp	GRGSOL	forms vector of constraints for GRG2 algorithm.
getnet	HVACLIB	reads network array data from file 'title'.net.
getres	INTRES	reads results for profile simulation from file.
grgsol	GRGSOL	solution routine using GRG2 algorithm.
iarray	NETWRK	inserts component data into network arrays.
initcall	INITCALL	calls initialisation subroutines for network and data record routines.
intres	INTRES	interprets results of profile simulation.
monit	NRSOL	monits the progress of 'nrsol'.
netwrk	NETWRK	network definition of HVAC system.
newdat	NEWDAT	transforms data records to new format when program parameters have been altered.
newdis	NEWDAT	new version of subroutine 'disfil'.
newfil	DATAFILE	compilation of a new component data record.
newfwrite	NEWDAT	new version of subroutine 'fwrite'.

newrec	HVACLIB	displays data records available in directory file and prompts selection of record name.
nodmen	HVACLIB	displays the node menu as defined at start of run by subroutine 'setnode'
nrfail	NRSOL	permits analysis of progress of solution algorithm 'nrsol' if a feasible point is not found
nrsol	NRSOL	solution subroutine to call 'crout' for the Newton - Raphson algorithm.
olddis	NEWDAT	old version of subroutine disfil.
oldfread	NEWDAT	old version of subroitine fread.
option	OPTION	displays menu of option commands available within SPATS. Program control returns to the option subroutine upon completion of a command and upon any programmed error condition.
optsol	GRGSOL	solution subroutine to call the GRG2 routines.
output	HVACLIB	prints the values of arcvariables at a soltion point
result	RESULT	calls the component result routines to interpret the performance of each component at the operating point of the system.
rexload	SIMDAT	reads a profile of exogenous variables from file.
savef	HVACLIB	writes a component data record to the directory file.
setnode	OPTION	sets up the node menu for currently available initialisation subroutines.
simdat	SIMDAT	gives options to compile or edit a load profile of exogenous variables.
solmon	NRSOL	analyses the progress of the Newton - Raphson algorithm by lookin at the output of subroutine 'monit' for each iteration.
terminf	INTRES	prints information on the way the GRG2 algorithm terminated for a specific run.
transr	DATAFILE	allows transfer of individual component data records between the component directory files of different users of the multi-user version of SPATS.
version	OPTION	sets up a run of SPATS by referencing the date, user reference name and system title. Gives the current version number for the multi-user environment.
wexload	SIMDAT	write a profile of exogenous variables to a file.

External subroutines to 'hvacprog'

1. Component algorithms compiled into segment 'components' which comprise initialisation, executive and results subroutines for component models which are tried, tested and generally available within the multi-user environment.

2. Default initialisation subroutines relating to each node type number (1 - 60) which ensure that these node types cannot be referenced by the network definition and data record routines unless a component model has been specifically defined and the initialisation routine has been referenced through the MULTICS search rules for that node type.

3. Subroutines developed by J.A.Wright:-

subroutine curvfit - gives access to a suite of programs for curve fitting, graphical display and filing component performance data (Wright 1984).

subroutine getcoe - reads a file of polynomial coefficients produced by the 'curvfit' suite directly into a data record.

subroutine optdes - gives access to a suite of programs for optimisation of the design of HVAC systems.

function polyxz(x,z,n) - calculates the value of the n^{th} polynomial in a component executive routine for variables x and z.

4. NAG (Numerical Algorithms Group) Subroutines.

subroutine f03aff - Crout decomposition of matrix A into a lower triangular matrix L and an upper triangular matrix U.

subroutine f04ajf - solution of simulataneous equations using a previous factorisation $LU\underline{x} = \underline{b}$.

5. Generalised Reduced Gradient Algorithm.

The GRG2 program (Lasdon 1978) is accessed via an 'easy to use' subroutine interface 'subgrg' which sets the default parameters for the particular application of GRG2 to the solution of HVAC simulation networks.

6. Machine Specific Routines.

Although the entire suite of routines in SPATS is written in FORTRAN77 to ease transfer of the program between different computers, reference is made to several machine dependent routines which are written in PL1 on the MULTICS :-

subroutine com_issue	issues a command to the MULTICS operating system. The principle use is to list the files of a specific format available in the current working directory, eg: *.net for all network files or *.res for all results files.
subroutine ioa_$nnl	MULTICS subroutine for a prompt without a line feed character - there is no FORTRAN77 command to supress the line feed after a line of text.
subroutine qmill	MULTICS subroutine to determine the elapsed prcessor time (CPU time) since 'login' : used to determine the relative computation time for different solution algorithms.
subroutine timp	subroutine to read the MULTICS clock for current time and date.

APPENDIX E

List of common block variable names used in SPATS

Common Block	Variable Name	Type	Use
cdataf	dataf	c*8	component data record name.
	buff	c*8(16)	array buffer for record names.
	fdate	c*8	date on which data record was created.
	feng	c*8	user name who created data record.
cinit	file	c*8	filename for component record directory file
	vname	c*8(mvar)	names for component arcvariables
	exnam	c*8(mvar)	names for component exogenous constants
	cname	c*8(mcon)	names for component constants & polynomials
cnetwk	varnam	c*8(maxvar)	network arc variable names
	fname	c*8(maxnod)	record names for each node in network
	namexc	c*8(maxnod)	network exogenous constant names
cprofil	timnam	c*8(maxtim)	names for time periods in load profile
	num	c*2	load profile identifier
datfil	nfiles	i*4	number of component data records in file.
	lr	i*4	logical record No. for direct access files.
	pos	i*4	position of record in directory buffer.
	iflag	l	logical variable for file handling routines.
	funit	i*4	file input/output unit number
init	nc	i*4	number of constants for component.
	nv	i*4	number of variables for component.
	ne	i*4	number of exogenous constants for component.
	nf	i*4	number of residual functions for component.
	np	i*4	number of polynomials for component.
kounts	ivar	i*4	arcvariable index for network.
	iconst	i*4	constant index for network.
	ifunc	i*4	residual function index for network.
	iexo	i*4	exogenous variable index for network.
	iexcon	i*4	exogenous constant index for network.
	ipoly	i*4	polynomial index for netwrok.
logsol	iflag	l	logical variable for file handling.
	iopt	l	if .true. GRG2 optimises objective.
	prof	l	if .true. use load profile.
	iprint	i*4	print level for GRG2.
	inf	i*4	termination index for GRG2.
mdata	mfsum	r*8(30)	monit array for fsum
	msumdx	r*8(30)	monit array for sumdx
	mxmax	r*8(30)	monit array for xmax
	mdxmax	r*8(30)	monit array for dxmax

	mx	r*8(30)	monit array for x
	mdelta	r*8(30)	monit array for the dx_i
	ml β nl	i*4	monit equivalents of m & n
	itl	i*4	monit iteration counter.

netwk	arcvar	r*8(maxvar)	arc variable array
	lb	r*8(maxvar)	lower bounds array
	ub	r*8(maxvar)	upper bounds array
	net	i*4(maxnod,maxnet)	network array
	idiff	i*4(maxvar)	index of exogenous variables in arcvar
	exvar	i*4(maxexo)	index of exogenous arcvariables
	con	r*8(maxcon)	network constants
	excon	r*8(maxnod)	network exogenous constants
	netcoe	r*8(maxnod,mcoe)	network polynomial coefficients
	nodeno	i*4	current node number

node	x	r*8(mvar)	component arcvariables
	bn	r*8(2,mvar)	bounds for component arcvariables
	c	r*8(mcon)	component constants
	e	r*8(mex)	component exogenous constants
	coefs	r*8(mnp,mcoe)	component polynomial coefficients

optdat	iopvar	i*4(maxexo)	index of optimisation exo-constants
	obu	r*8(maxexo)	upper bounds for optimisation variables
	obl	r*8(maxexo)	lower bounds for optimisation variables
	objfun	r*8(maxnod)	energy consumption for each node
	nopt	i*4	number of optimisation variables

profile	ntime	i*4	No. of time intervals for load profile
	nload	i*4	No. of exogenous variables for profile
	pexvar	r*4(maxtim,maxexo)	profile exogenous variables

resdat	resvar	r*8(maxtim,maxvar)	operating point for time intervals
	inform	i*4(maxtim)	termination mode for each time interval

solve	it	i*4	number of iterations of NR solution.
	narc	i*4	total number of arcvariables.
	k	i*4	total number of residual vector calculations.
	fsum	r*8	Euclidean norm of residual vector.
	sumdx	r*8	summation of all dx_i in NR iteration.
	maxdx	r*8	maximum delta x_i in NR iteration.

titles	ftitle	c*8	job title.
	date	c*8	date in dd/mm/yy format from MULTICS.
	enginr	c*12	user identification.

totals	nnode	i*4	number of nodes in network.
	nvar	i*4	number of true variables (narc-nexo) .
	nexo	i*4	number of exogenous variables in network.
	nconst	i*4	number of data constants for network.
	nfunc	i*4	number of residual functions in network.
	nexcon	i*4	number of exogenous constants.
	npoly	i*4	number of polynomials.