THE OPTIMISED DESIGN OF HVAC SYSTEMS.

by

Jonathan Andrew Wright B.Tech.

A Doctoral Thesis submitted in partial
fulfilment of the award of
Doctor of Philosophy
of the Loughborough University of Technology
September 1986.

SYNOPSIS.

The workable design of HVAC (Heating Ventilating and Air Conditioning) systems is based upon sizing the components individually to meet a peak duty of a nominal operating point. Growing economic pressure demands more cost effective and efficient designs, but the appraisal of alternative solutions is limited by short design and construction times. The design of HVAC systems can benefit from the application of numerical optimisation methods as these allow the rapid appraisal of alternative schemes and the sizing of the components simultaneously for criteria such as minimum first cost, operating cost, life-cycle cost or primary energy consumption.

Optimisation problems can be categorised according to the characteristics of the functions used to appraise the solutions and those of the constraints on the problem. This thesis discusses the formulation of HVAC system design problems in this context and describes the development of an optimisation procedure which is based upon a data base of manufactured components and operating parameters such as controller setpoints, mass flow rates and temperatures. The thesis describes several objective functions used in the appraisal of solutions and describes the use of constraint functions in restricting the solution to a practicable design.

HVAC system optimised design problems can be solved using direct search methods. The implementation of three direct search algorithms is described and the limitations of each discussed. Conclusions are drawn and the characteristics of HVAC system optimised design problems used to make recommendations for the future development of an idealized algorithm.

The thesis describes the development and structure of the optimised design program and its integration with an existing suite of simulation programs. The application of the program to the design of example heat recovery systems is given and the potential use of the software in other applications described together with proposals for the development of the procedure as a design tool.

ACKNOWLEDGEMENTS.

The author would like to thank Dr. V.I. Hanby for his advice and encouragement: Dr. M.A.P. Murray for his collaboration in program development and Dr. P.R. Deering for the development of a chiller model data base. Thanks also to Mr. L. Mustoe and to Mr. L.A. Wright for proof reading this thesis and for their useful advice.

The author would also like to acknowledge Woods of Colchester Ltd. and Dunham Bush Ltd. for providing valuable component performance and cost data.

CONTENTS

Chapter 1. COMPUTERS AND OPTIMUM DESIGN.

The use of computers in the thermal analysis of buildings and of heating, ventilating and air-conditioning (HVAC) systems has grown from a need to improve the efficiency of the design process, this itself being influenced by changes in technology and by growing economic pressure.

Initially computer software was developed for the thermal analysis of buildings, these programs used to predict the energy demand within each zone, thus allowing rapid appraisal of architectural changes and the selection of equipment based on the resulting peak loads. Programs of this type assume idealised control of the installed plant, the system maintaining constant conditions in the occupied zone.

A natural development from the idealised control approach is to full 'system simulation' in which the performance and operating point of the plant is predicted under varying load conditions. Several system simulation programs of varied sophistication are available, their applications ranging from simple energy accounting to a detailed analysis of the system variables for each component in the system. The advantage of using system simulation programs is that the performance of different designs can be analysed where this would be impossible using manual calculation methods.

With design and construction times at a premium and increasing demand for more effective designs, there is vast scope for the development of software to aid the design of building services systems.

1.1 Computers in Building Services Design.

Justification for using computers in the design of building services is well established (Wix, 1985: Baxter, 1985: Wright, 1985). The most influential reason is that computers perform numerical and data retrieval tasks much faster than humans, which allows appraisal of alternative designs and a higher level of accuracy to be employed. Exploitation of this power is expected to reach a level where computers will produce integrated building designs from a minimum of human input.

Wix (1985) suggest three categories of software type (figure 1.1):

Category 1 software can be regarded as the implementation of manual calculations. Each program of this type is totally separate and requires comprehensive input from the users as stored data is kept at a minimum or may be non-exisitent. Users of this type of software require a number of programs to cover the range of applications and may find them of limited use due to the time spent entering data. Category 1 software can be run on virtually any type of computer from a simple micro to a large mainframe.

Category 2 software is defined as data base software in which data entry by the users is rationalised by the existence of a data base. Several application routines can access the common data minimising the need for repetitive input. The data bases are either fixed or generated using category 1 software. An example of this type of software is the room model in which the room data provides a common base for a variety of applications such as heat gains, daylight calculations and acoustics. Thermal analysis of buildings and system simulation software fall into this category with room, system and transport medium data bases. The sophisticated file handling and data retrieval of category 2 software requires a more powerful computer, such as the larger 16 bit micros with hard disk facilities.

Category 3 software is comprised of the whole building model and incorporates the amalgamation of design and draughting packages. Definition of the building and HVAC system configuration is by graphics software thereby reducing the amount of data entered and therefore the effort required in specifying the building and system configurations. This is followed by true integrated design in which the relationships between building and system parameters are evaluated at all stages of the design. The individual design and draughting packages in the amalgamated software could originate from previously developed category 2 software. Category 3 software requires computers capable of complex graphics and long numerical calculations and therefore can only be implemented on the larger mini and mainframe computers.

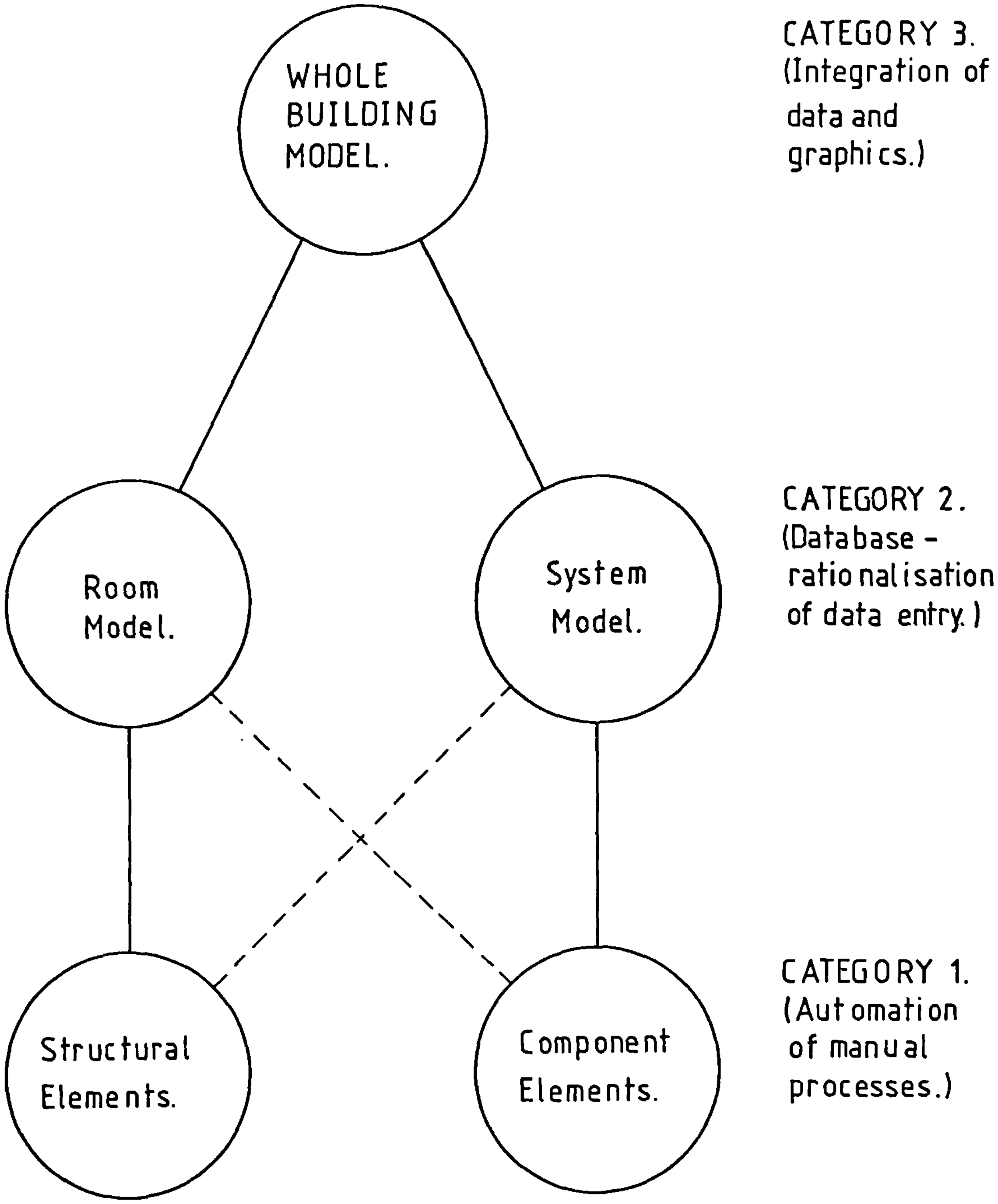


figure 1-1, Hierarchy of Software Types. (after Wix, 1985)

It is likely that the introduction of 'Expert systems' software will enhance category 3 software by providing specialised knowledge on the application of different schemes and in the control of the design process by generating alternative proposals and appraising solutions. An advantageous characteristic of expert systems is that they can make both quantitative and qualitative appraisals of design solutions. For example, in the design of an HVAC system, the expert system software could assess the suitability of a particular control strategy based on its cost effectiveness, probability of failure and ease of use. Some of this information is in the form of results obtained from other software, such as the existing system simulation packages.

System simulation software assesses the performance of HVAC systems for a fixed size of component. No attempt is made to assess the affect of a change in size of component on system performance. Before the 'expert' can select the 'optimum' scheme the quantitative parameters in the appraisal must represent the best that can be obtained from each scheme. This can only be achieved by varying the size of components until the 'optimum' system performance is obtained and the components are at their 'optimum' size. Selection of the optimum size of components represents a gap in existing software. Its development is necessary before true category 3 software can be developed and as an individual package would provide a useful tool in improving the design process.

1.2 Workable or Optimum Design ?

Figure 1.2 illustrates the steps in the building process, those most relevant to developing an optimum design strategy are the preliminary and detailed design stages. At the preliminary design stage one scheme is selected from a range of alternatives, the selection often based on intuition and the engineers experience. Detailed design begins with an assessment of the plants operating point, based on the peak loads and continues with the sizing of the individual components.

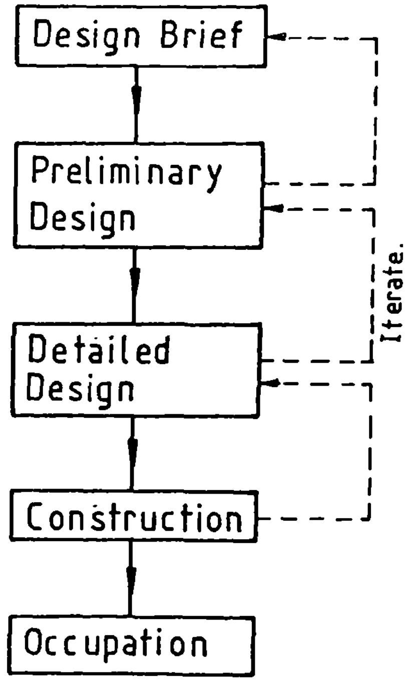


figure 1·2, Simplified Building Process.

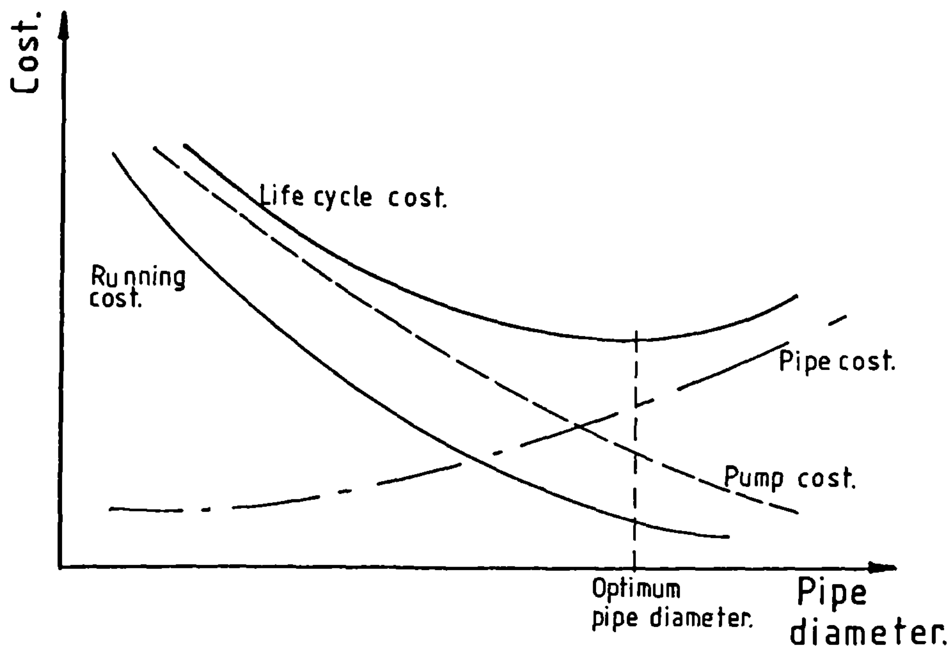


figure 1·3, Life Cycle Cost of a Pump Scheme.

The introduction of system simulation software has improved the design process by allowing the performance of several schemes to be evaluated, giving the engineer more information on which to base his selection. However, the existing design process leads to the production of a 'workable' system as opposed to an 'optimum' system.

Stoecker (1971) defines a 'workable' system as one which:

'1. Meets the requirements of the purpose of the system (such as providing the required amount of power, heating, cooling, or fluid flow, or surrounding a space with a specified environment).
2. Will have satisfactory life and maintenance costs.
3. Abides by all constraints, such as size, weight, temperatures, pressure, material properties, noise, pollution, etc.

In summary, a workable system performs the assigned task within imposed constraints.'

What then is an 'optimum' system design ? This is best illustrated by example. Suppose that a pump and pipework is installed in a large office block to pump water from a basement tank to a tank on the roof. The approach in producing a workable system might be:

1. Allow a nominal water velocity of 1.5 m/s.
2. Size the pipe diameter from the required volume flow rate and water velocity.
3. Calculate the head loss in the system.
4. Size the pump from the head loss and volume flow rate.

In order to produce an optimum design there is a need to specify some criterion to optimise. Often this is a life-cycle cost consisting of first cost, pumping cost and maintenance cost. In the optimum design approach the water velocity is not fixed but allowed to float free. Since it is the components we are sizing it is more convenient to continue our discussion in terms of pipe diameter rather than water velocity in the pipe.

As the pipe diameter increases so its first cost increases, but due to the lower head loss, the running and first cost of the pump are reduced. Taking the life-cycle cost as the sum of the individual costs there is a size of pipe diameter which gives the minimum life-cycle cost, figure 1.3.

The principal differences in producing an optimum design as opposed to a workable design are:

1. Design of a workable system necessitates the fixing of design parameters such as velocities and temperatures. These values are arbritrary and originate from what is regarded as 'good working practice'. The optimum design approach allows as many parameters as is possible to float free during the design process.
2. In the optimum design approach the final values of the design parameters are obtained by varying their value until a minimum value of an 'objective function' is reached, thus ensuring they have the 'best' and not arbritrary value.

To summarise, an optimum system is the 'best' of all workable systems. The advantages of using the optimum design approach are obvious, but the time taken with manual calculations are prohibitive and therefore this approach requires the development of computer software to perform the calculations.

Use of computer software to find the optimum size HVAC systems will improve the efficiency of the design process by combining in part the preliminary and detailed design stages. The new procedure would be:

1. Identify alternative schemes and if building thermal analysis software is unavailable, calculate the zone loads.
2. Find the optimum size of the components in each scheme.
3. Use the results of the optimisation in the selection of the 'best' scheme.

No further calculation is necessary as the operating point and performance of the plant is calculated during the optimisation.

Clearly there is a need to improve the efficiency and cost effectiveness of HVAC systems designs. Development of an optimised design procedure would not only help meet this need but would improve the effectiveness of the design process itself. This thesis describes the development and structure of an optimised design procedure for the optimum selection of HVAC system components.

Chapter 2. FORMULATION OF AN OPTIMISATION PROBLEM.

The are three elements in the formulation of an optimisation problem:

1. The problem variables.
2. The objective function, which is an expression giving a measure of how close the solution is to the optimum.
3. The problem constraints.

The solution of optimisation problems is by an algorithmic search for the minimum or maximum value of the objective function. The type of search is dependent upon the characteristics of the particular problem as described by the variables and objective and constraint functions.

2.1 The Problem Variables.

Formulation of an optimisation problem begins with the identification of the problem variables. It is the value of these that the optimisation algorithm varies until a combination is found which gives an optimum value of objective function. Examples of problem variables in HVAC system design are fan diameter and condenser water flow temperature.

All methods of optimisation demand that the problem variables are independent but can be continuous or discrete. The problem variables are denoted by:

$$x_1, x_2, \ldots\ldots x_n$$

or in vector form: $\underline{X}$

2.2 The Objective Function.

The objective function is a single measure of 'goodness' the value of which is dependent on the value of the problem variables. Optimising the objective function involves finding the values of the problem variables which gives the minimum or maximum value of the objective function. In HVAC design this is normally a minimum value, for example minimum first cost of the system.

An objective function of 'n' variables is denoted by:

$$F(\underline{X})$$

2.3 The Constraints.

Most engineering design problems have constraints, for instance the water velocity in a pipe lies between zero and an upper limit after which erosion of the pipe begins. Similarly the optimisation of an objective function is subject to constraints. Three types of constraints can be identified, simple bounds, linear constraints and non-linear constraints.

Variables which have a restriction on their value are said to be simply bounded, which is a special form of linear constraint. A general linear constraint is defined as a constraint function which is linear in more than one variable. The various types of linear constraint are summarised below and are presented in a form which represents the mathematical statement of a linearly constrained problem:

equality constraints:	$g_i(\underline{X}) = b_i$:	$i=1,2,\ldots,m_1$
inequality constraints:	$g_i(\underline{X}) \leq b_i$:	$i=m_1+1,\ldots,m_2$
	$g_i(\underline{X}) \geq b_i$:	$i=m_2+1,\ldots,m_3$
range constraints:	$lb_j \leq g_i(\underline{X}) \leq ub_j$:	$i=m_3+1,\ldots,m_4$
		$j=1,\ldots,m_4-m_3$

Each g_i is a linear function and b_i, lb_j and ub_j are constant scalars. The addition of non-linear constraint functions would include the statements:

equality constraints:	$c_i(\underline{X}) = 0$:	$i=m_4+1,\ldots,m_5$
inequality constraints:	$c_i(\underline{X}) \geq 0$:	$i=m_5+1,\ldots,m_6$
range constraints:	$lbn_j \leq c_i(\underline{X}) \leq ubn_j$:	$i=m_6+1,\ldots,m_7$
		$j=1,2,\ldots,m_7-m_6$.

where each c_i is a non-linear constraint function and lbn_j and ubn_j are scalars. Note that inequality constraints of the form $c_i(\underline{X})\leq 0$ are not included as this is equal to $-c_i(\underline{X})\geq 0$.

2.4 The Character of Optimisation Problems.

Figure 2.1 illustrates a two dimensional problem in which the function values decrease towards the centre contour. Point $\underline{X}_g$ is a local unconstrained minimum and as this is the smallest of all function values it is termed the global minimum. Point $\underline{X}_s$ is termed a saddle point as it is a minimum along $\underline{AB}$ but a maximum along $\underline{CD}$.

The hatched side of the constraint $g(\underline{X}) \geq 0$ represents the infeasible region, $g(\underline{X}) < 0$ and as the global optimum lies outside this the solution is unaffected by the constraint. Introducing the non-linear constraint $c(\underline{X}) \geq 0$, places the global optimum in the infeasible region giving a new solution point $\underline{X}_L$, which is termed a local constrained optimum. Therefore the effect of the constraints is to reduce the region in which the solution can lie. This in some cases leads to a local optimum solution as opposed to the global optimum.

2.5 Classification of Optimisation Problems.

The sequence of operations performed by most optimisation algorithms is: find a point which satisfies all the constraints, optimise the value of the objective function and finally confirm the optimallity of the solution. The optimisation of the objective function has itself two processes, an assessment of the direction in which to move the value of the problem variables and by how much to move them. Such optimisation algorithms can be classified as direct search methods or derivative methods. Direct search methods are heuristic in character basing their search strategy on a comparison of objective function values, whereas derivative methods are mathematical in character, using the first and sometimes second derivatives of the objective functions to establish a search direction.

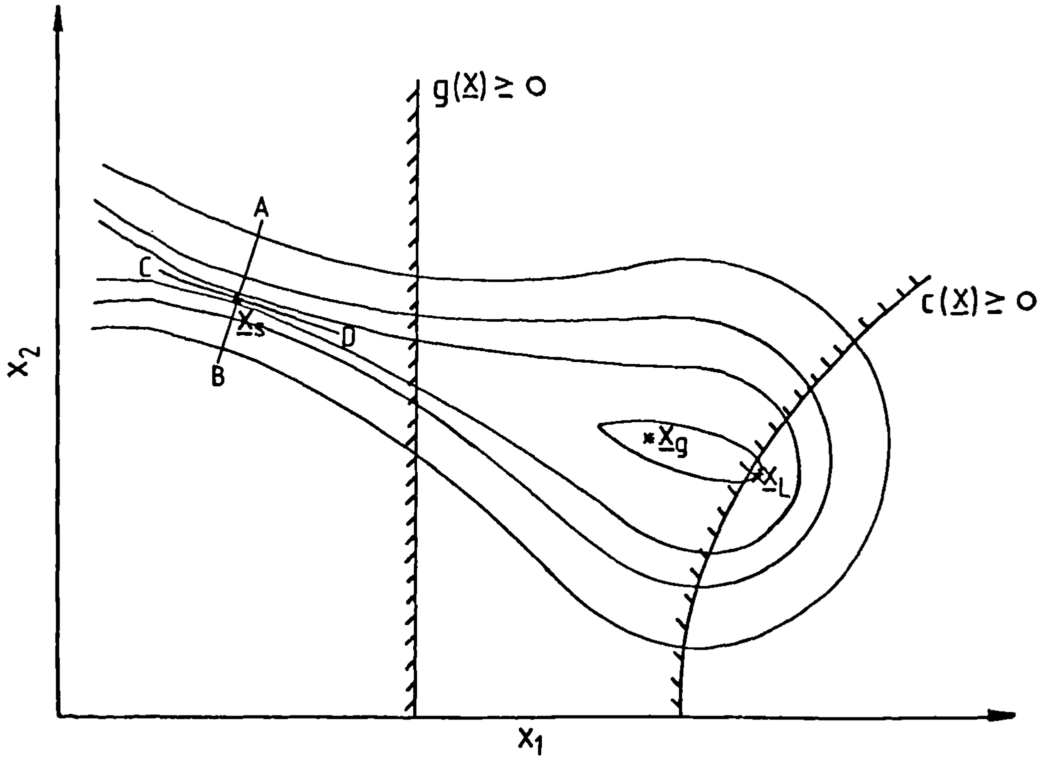


figure 2·1, An Optimisation Problem in Two Dimensions. (after NAG).

Properties of $F(\underline{X})$	Properties of $c(\underline{X})$
Function of a Single Variable. Linear Function. Sum of Squares of a Linear Function. Quadratic Function. Sum of Squares of a Nonlinear Function. Smooth Nonlinear Function. Sparse Nonlinear Function. Non-smooth Nonlinear Function.	No Constraints. Simple Bounds. Linear Function Sparse Linear Function. Smooth Nonlinear Function. Sparse Nonlinear Function. Non-smooth Nonlinear Function.

table 2·1, Properties of Objective and Constraint Functions. (after Gill, 1981).

Due to the differences in individual problems, their solution by a single all purpose algorithm would prove cumbersome and inefficient. It is desirable to identify the characteristics of the problem which allow it to be solved more easily. The most notable differences in optimisation problems are in the mathematical characteristics of the objective and constraint functions. For example, the objective function may be smooth in some cases and discontinuous in others, the objective function may be calculated from a simple relationship or require a complex series of calculations. Table 2.1 gives a reasonable classification scheme the development of which has been based on balancing improvements in efficiency against the complexities of providing a larger selection of solution algorithms (Gill, 1981). An example classification might be, a linear objective function with linear constraints.

Another important feature of optimisation problems which affects the choice of solution method is the 'size' of the problem. This affects both storage requirements and the time taken in computation. The importance of problem size is related to the availability of large computers: obviously the more powerful the machine that is available, the less the significance the problem size.

Choice of solution method is also influenced by the availability of information. For instance, the first and second derivatives of the objective function may be obtained analytically or by numerical methods. Here there is a need to balance the effort in calculating function values against that of the operation of the solution method.

Finally, there may be any number of special requirements which influence the choice of solution method, not least of these is the accuracy required.

Chapter 3. OPTIMISED DESIGN OF HVAC SYSTEMS.

Selection of the optimum HVAC system is based on both qualitative and quantitative parameters. To ensure a true comparison of systems, the quantitative parameters must represent the optimum design/selection of the system components. The procedure for the optimised design of HVAC systems has three elements (figure 3.1):

1. The 'expert', whether the designer or expert system software identifies the possible system types based on an outline of the application.
2. For each system the size of components is optimised for a given objective function.
3. The objective function values (eg: life-cycle cost) of each system are used as the quantitative parameters in the assessment of each systems performance, thus enabling the selection of the optimum system.

The complexity of selecting HVAC components is illustrated in figure 3.2. The 'size' of each component is specified by one or more variables, some of which may also be associated with the adjacent components. For example, the size of an axial flow fan is represented by two variables, the fan diameter and running speed. To allow its installation the diameter of the fan must match that of the adjoining duct work and therefore in the optimised design process these dimensions are represented by a single variable, the diameter.

The values of certain fluid variables also affect the optimum selection of components. For instance the choice of condenser water flow temperature will influence the selection of the chiller and cooling tower. It is therefore important to identify the fluid variables which influence design solutions and include them as design variables in the problem specification.

The task of component selection is further complicated by a component data base which consists of several product ranges. Each product range has two sources: firstly the component could be supplied by one of several manufacturers and secondly within each manufacturers range there will be geometric and variable differences which effectively divides each range into different products.

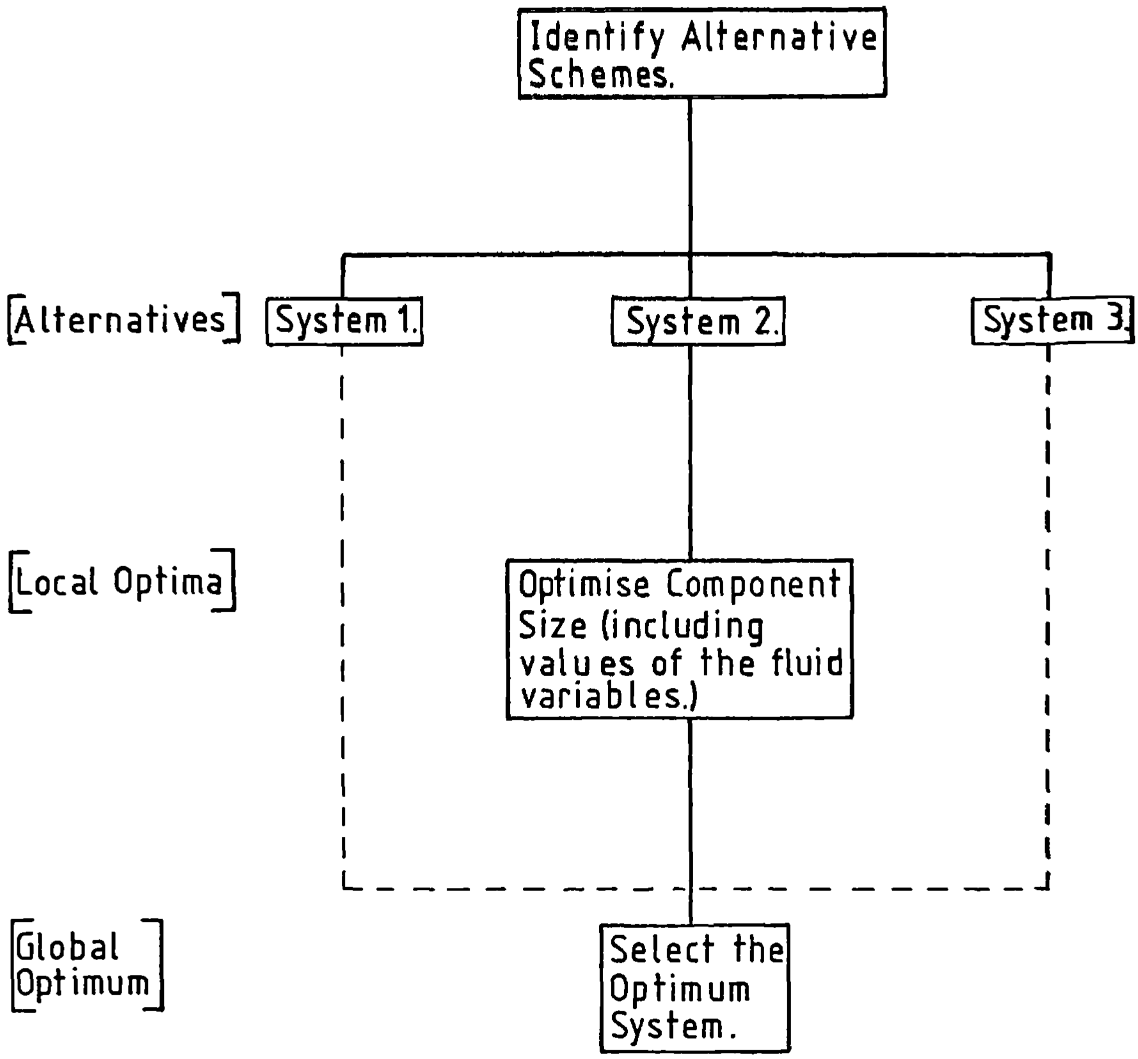


figure 3·1, The Optimised Design of HVAC Systems.

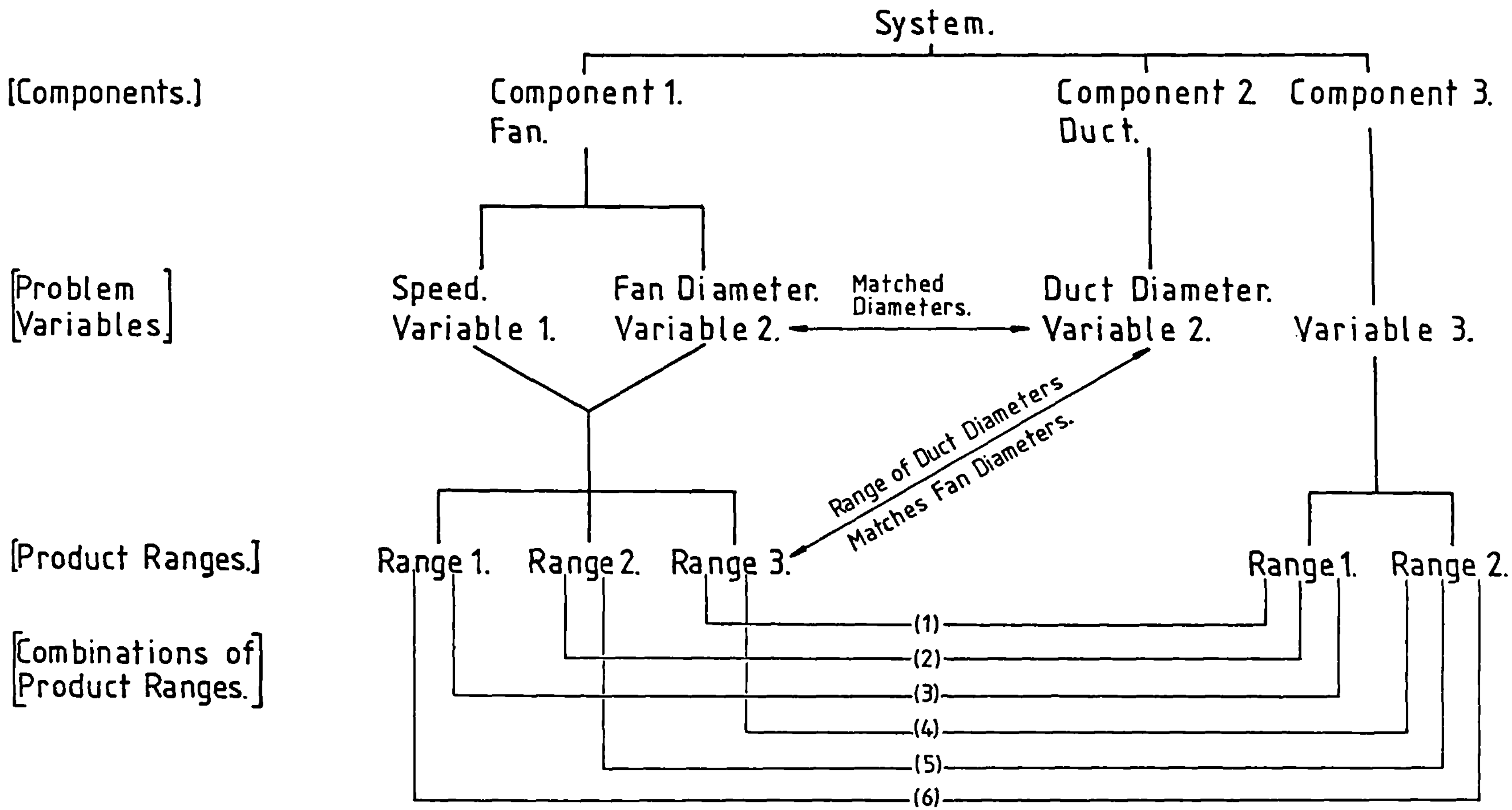


figure 3·2, The Relationships in Optimised Component Selection.

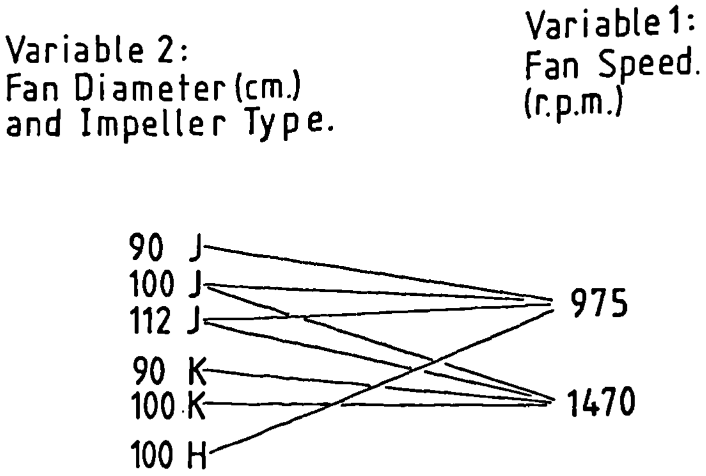
Suppose that an extract system requires an axial flow fan of diameter between 0.9 m. and 1.12 m., then a range of products for a given manufacturer may be similar to figure 3.3. Codes J, K and H represent different impeller geometries which gives three geometrical product ranges. To ensure complete independence of variables a further distinction is made for fan speed. The J range is subdivided for the 90J fan as this has been designed to run only at 975 rpm. where 100J and 112J fans can run at speeds of either 975 rpm. or 1470 rpm. This gives a total of four product ranges.

For a given combination of product ranges there will be a local optimum choice of component sizes. Changing the product range for one component can influence the optimum size of other components in the system. This indicates that there are two levels of optimisation in sizing the system, finding the optimum combination of product ranges and for that combination, finding the optimum size of components.

Numerical optimisation methods require numerically identified problem variables so that they can assess the direction and amount of change in value of the variables throughout the search. Therefore if the optimum combination of product ranges is to be found by numerical methods, each product range must be numerically and uniquely identified. It is imposssible to assign meaningful numerical values to product ranges when they are distinguished merely by supply from different manufacturers. This limits the choice of search technique to an exhaustive search of all possible combinations of product ranges.

The process of finding the optimum size of components can be summarised in three steps (figure 3.4):

1. Identify all combinations of product ranges.
2. Find the optimum size of components for each combination of product ranges, these representing local optimum solutions.
3. The overall optimum solution is then taken as the local optimum with the lowest objective function value.



Range.	Fan Diameter and Impeller Type.	Speed.
1	90 J	975
2	100J 112 J	975 or 1470
3	90 K 100K	1470
4	100H	975 or 1470

figure 3·3, Axial Fan Product Ranges.

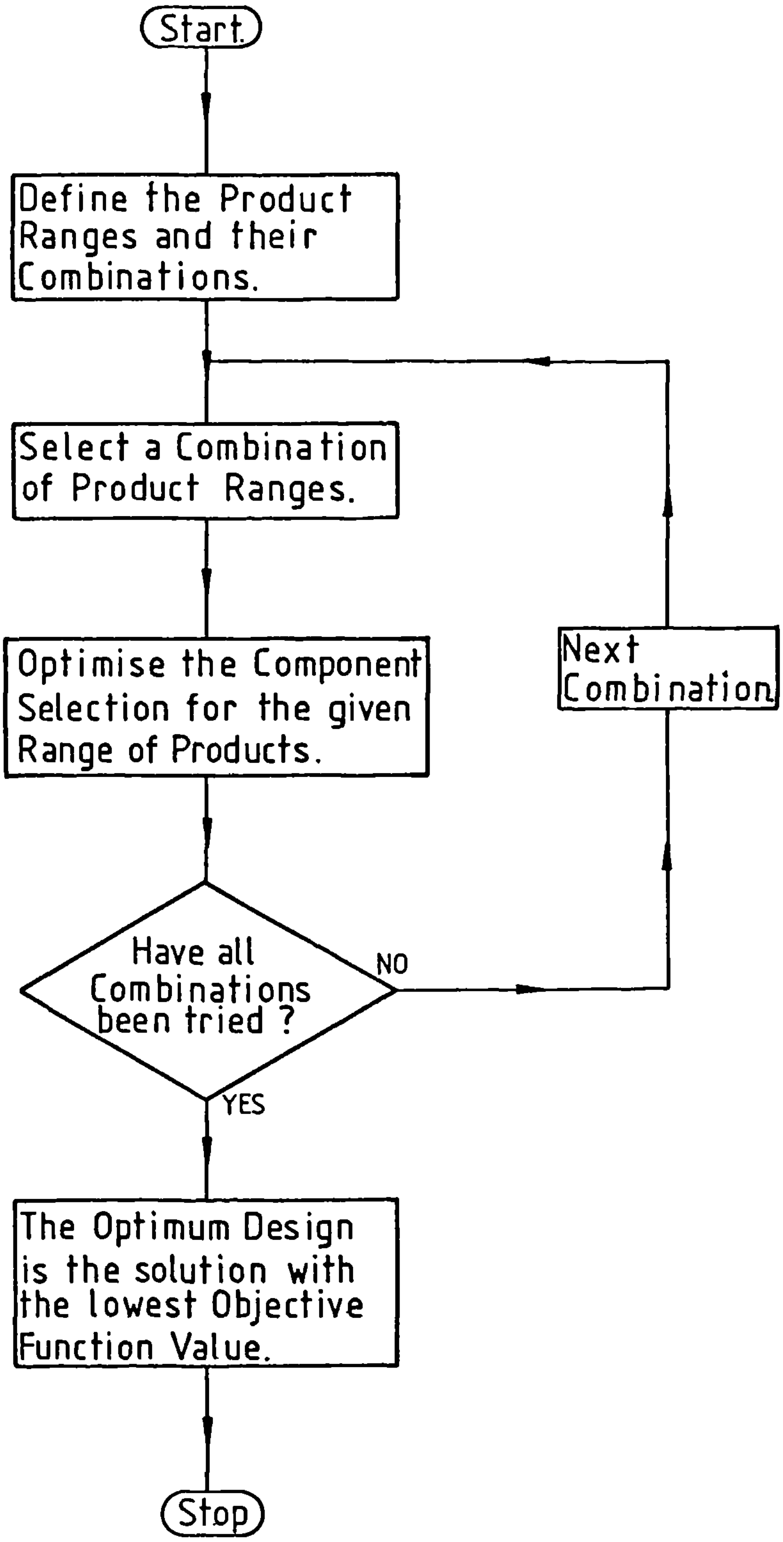


figure 3·4, The Process of Optimised Component Selection.

The exhaustive search of product ranges is easily developed and therefore the area of most interest is in finding the optimum size of components for a given range of products. Mathematically, this is the more complicated task and is fundamental to the whole process of optimised design. This therefore is the area of research described in this thesis.

3.1 The Problem Variables in HVAC Design.

The problem or design variables are the parameters normally used to describe the selection of HVAC components. These represent the physical size and operating point of the component or may be associated with the capacity of the component. For example, the parameters used to specify the selection of centrifugal fans are impeller diameter and running speed, the impeller diameter representing the physical size of the fan and running speed its operating point. Conversely the manufacturers catalogue numbers used to identify the selection of package chillers are more often related to the peak duty of the chiller than its physical dimensions. Such catalogue numbers can form suitable design variables with which to size package components as they are an indication of both the physical size and operating point of the component. A final group of problem variables are the fluid property variables which affect the choice of components and therefore the optimum solution. In practice these variables generally appear as the set points of the equipment controls.

It is important when defining the problem variables within the design procedure, to ensure that each of the matching dimensions of adjoining components forms a single problem variable, thus guaranteeing that the optimum solution will be one which allows the components to be physically connected.

3.1.1 Mathematical Characteristics.

The most important characteristic of the problem variables is that the majority are discrete and cannot be approximated as continuous: which severely restricts the choice of optimisation algorithm. The discrete nature of problem variables arises due to the way in which products are manufactured.

A range of fans is manufactured in fixed diameters and the height interval of a heating coil is restricted by the spacing between the water tubes. Any continuous variables that do occur are usually associated with the transport media and control settings.

3.2 The Constraints in HVAC Design.

The most important constraint is that the optimum solution must be one in which all the components selected are correctly sized and operating within their design limits. Although this is obvious its implications in optimised design are not. If the undersizing of components is to be included as a mathematical constraint, then the severity of undersizing must have a numerical significance. For example, if a fan is unable to meet the required pressure rise then there must be a numerical relationship between the 'degree' of undersizing and a change in fan size. Identifying undersized components requires a sophisticated system simulation technique which can assess the operating point of the plant and can provide numerical data which can be used to formulate a component undersizing constraint function.

Other sources of design constraint are:

1. Codes of Practice.
2. Restrictions on configuration.
3. Physical restrictions forming simple bounds on the variables.

Apart from British Standards Codes of Practice, several organisations have their own Code of Practice. Such codes set limits on the design parameters, the limits dictated by what is regarded as good working practice. As optimised design becomes more established the nature of the codes will change. For instance, to prevent moisture carry-over the face velocity of a cooling coil is often limited to a maximum value of 2.5 m/s. Obviously in the optimised design process the true constraint is on moisture carry-over and not face velocity. Therefore it is likely that as optimised design becomes common practice, moisture carry-over will replace face velocity in codes of practice.

Configuration constraints are those related to the construction of the components. For example, in designing the fan section of an air-handling unit (AHU) the relationship between box size and fan size is restricted, as to allow easy assembly there must be a certain amount of space between the fan and the sides of the box.

The final source of problem constraint is one which restricts the range of component sizes to those avalable and limits the fluid variable values to appropriate physical conditions. This type of constraint is represented by applying simple bounds to the problem variables.

It should be noted that not all the constraints described for each component will be required in every design, especially those related to Codes of Practice. Therefore the ability to specify an appropriate set of constraints for each design is a necessary feature of optimised design software.

3.2.1 Mathematical Characteristics.

The most significant characteristic of the problem constraints is that the majority are non-linear functions. This severely restricts the choice and development of an optimisation algorithm. Most constraints encountered in HVAC design problems fall into one of four categories:

1. Simple bounds.
2. Smooth linear functions.
3. Smooth non-linear functions.
4. Sparse non-linear functions.

3.3 Objective Functions in HVAC Design.

The objective functions implemented in this research have been chosen for their usefulness as quantitative measures in the comparison of system designs. Not all comparators used by designers have been included, but the range is considered comprehensive enough to prove the effectiveness of the optimisation algorithms. The inclusion of other objective functions at a later date should involve no more than writing subroutines which return the value of the objective function.

The objective functions implemented in this research are:

1. Net energy consumption of the system.
2. Primary energy consumption of the system.
3. Capital cost of the system.
4. Annual operating cost of the system.
5. Net present value of the system.
6. Payback period of the system.

3.3.1 System Energy Consumption.

An account of the system energy consumption is best achieved by primary energy modelling. To clarify this point a few definitions are required.

Primary (gross) energy is defined as (BRE, 1976): 'The (higher) calorific value of the of the raw fuel, eg: oil, coal, natural gas, nuclear and hydro-electricity, which is input into the UK economy'.

Net energy is the energy content of the fuel as received by the consumer. The difference between this and primary energy is termed the 'overhead'.

Useful energy is the energy required to perform a given task. The ratio of useful energy to net energy represents the efficiency of the component.

A primary energy ratio is the ratio of primary to net energy. Typical primary energy ratios are given in table 3.1.

System simulation techniques allow the calculation of the net energy consumption of the components and therefore the system. Yet this does not reflect the gross energy used by the system as this is dependent upon the overall efficiency of converting primary energy to useful energy.

Electricity.	3·82
Coal.	1·03
Natural Gas.	1·07
Oil.	1·09

table 3·1, Primary Energy Ratios. (B.R.E. 1976).

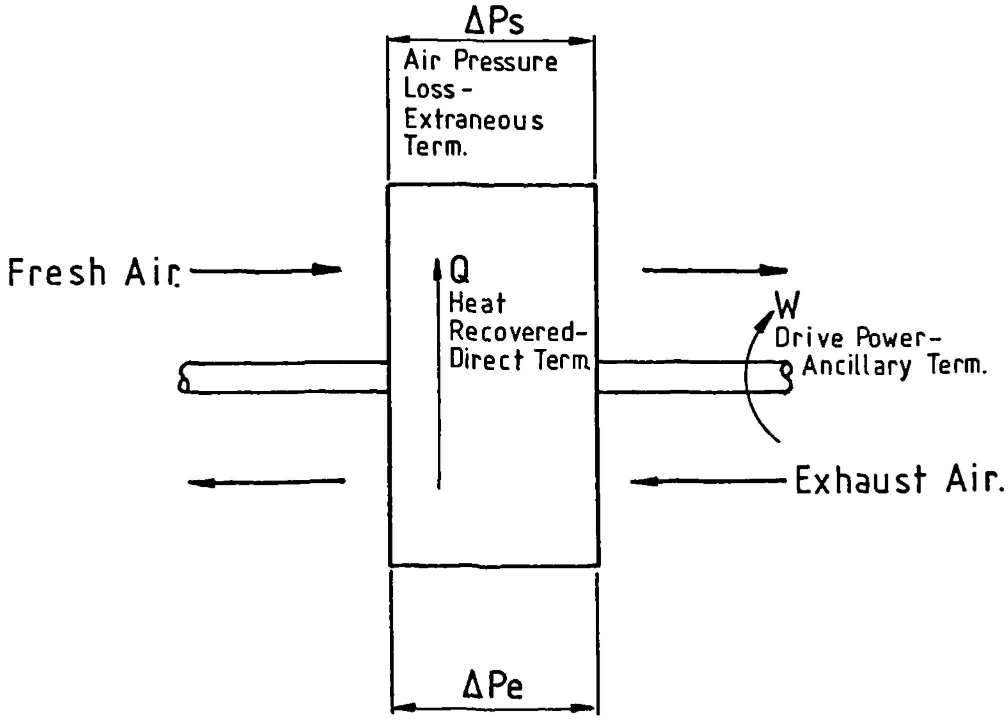


figure 3·5, Thermal Wheel Energy Terms.

In a comparison of a direct fired gas heater with an electric heater battery, the electric heater will have the best performance if net energy consumption is used as a comparitor. This is due to the electric heater battery having the higher efficiency in converting net to useful energy. The converse is true of converting primary energy to net energy, gas being the more efficient. Therefore if primary energy consumption is used as the comparator, the comparison may be more balanced or even in favour of the gas fired heater.

Both net and primary energy consumption of the system have been included as objective functions. The net system energy consumption has been included as for the moment this is the most common and simplest energy accounting procedure.

Modelling of the system energy consumption has two elements. Identification of which component energy terms to include in the model and whether the value of each term adds to or offsets the system energy consumption.

Three categories of energy term can be associated with each component and arise due to the simplistic nature of some component models together with a desire to optimise the design of subsystems. The three categories are direct, ancillary and extraneous. These are described in relation to the heat recovery wheel illustrated in figure 3.5.

A direct energy term is one for which the net energy can be calculated without reference to the performance other components. In the case of the thermal wheel this is the heat recovered.

An ancillary term occurs due to the simplicity of the component models. In practice the thermal wheel is constructed from two components, the wheel and its drive motor, each of which provide separate energy terms. In a sophisticated model the motor would appear as a separate component. Yet this would give a disproportionate increase in accuracy compared to the increase in calculation time. It is therefore more likely that the motor would be modelled as integral with the wheel, the net energy consumption of the motor forming another thermal wheel energy term. Thus an ancillary energy term is one which in practice forms a direct energy term of a separate component.

Extraneous energy terms are those which include the performance of other components in net energy calculations. The energy required to meet the air pressure loss across the thermal wheel can only be expressed as net energy if the efficiencies of the supply and extract fans are known.

The ability to specify whether to add or subtract the value of the energy terms in the system energy model is necessary when modelling subsystems and in particular heat recovery equipment. For example, the convention might be that energy used is added and energy recovered is subtracted. The same convention is also useful in formulating economic models, ie: the cost of energy used is added in the model and the cost of energy saved subtracted.

Consider the energy modelling of two problems, one in which the thermal wheel is part of a complete heating and ventilating system and the second, a subsystem consisting of just the thermal wheel. The formulation of the energy model for the subsystem might be:

Energy model =	Wheel drive +	Air pressure loss –	Energy recovered
(Energy used by the system.)	(Electrical energy.)	(Electrical energy.)	(Thermal energy, eg: Gas or Oil.)

To express the air pressure loss term as net energy requires a knowledge of the supply and extract fan efficiencies. As the fans are not part of the problem definition it is likely that this term will remain expressed as useful energy. Similarly, in order to express the energy recovered by the wheel as primary energy, an assesment of the efficiency of an alternative heat supply device is required.

In the heating and ventilating system the energy recovered by the thermal wheel is represented by a reduction in boiler duty and as the supply and extract fans are part of the problem definition the only thermal wheel energy term to appear in the system energy model is the wheel drive energy. Thus the system energy model could be expressed as:

Energy model	=	Boiler duty	+	Fan drive	+	Wheel drive
(Energy used by the system.)		(Thermal energy.)		(Electrical energy.)		(Electrical energy.)

To summarise, if net and primary energy consumption are to be included as objective functions in the optimisation procedure, the following criteria are required of the optimised design software:

1. Each component should be allowed to have any number of associated energy terms and there must be a means of defining which terms are included in the system model.
2. Those terms included in the system model must be linked with a fuel type and for extraneous terms be associated with the efficiency of the relevant energy using component. Each value of energy term should be allowed to be added or subtracted in the system model.

3.3.2 Capital Cost.

The capital cost of a component consists of:

1. The price of the component.
2. Delivery cost.
3. Installation cost.
4. The cost of additional building work.

Of these the predominant cost is the price of the component with delivery and installation cost increasing the total capital cost by a smaller percentage. Additional building costs are most significant in a comparison of entirely different schemes as any additional building work which occurs due to a change in size of the component is likely to be insignificant.

The component prices are normally presented by the manufacturer in a tabular form of discrete prices against the size of component. Some price lists can be curve fitted to reduce the amount of data handled, whilst the price of others can only be calculated using a complicated algorithm. It is impossible to define a general rule for the presentation of cost data but for data storage the facilities of most use are the ability to store curve fit coefficients and groups of discrete data.

Although the mathematical characteristics of objective functions are discussed in section 3.3.5, the discontinuous nature of the capital cost function warrants a more detailed explanation and is therefore described here. The discontinuous nature of cost functions is related to:

1. Allocation of manufacturing time.
2. Changes in manufacturing technique with size of component.
3. Allocation of materials.

The time allowed for manufacturing operations is rarely allocated as continuously proportional to the size of component, it is more common to allocate a 'time slot'. For example, the minimum time taken in grinding a 20 mm. diameter shaft may be longer than a 10 mm. shaft, yet it is possible that both minimum times are close enough to fall into the same time slot and hence are allocated the same grinding time. The minimum grinding time for a 30 mm. shaft may force it into the time slot above that of the 10 mm. and 20 mm. shafts, thus producing a discontinuity in the cost function.

Due to the physical limitations of manufacturing machinery, it is impossible to increase the size of component perpetually without changes in manufacturing technique. A change in manufacturing technique requires a change in cost structure which in turn leads to discontinuity in the cost function.

To reduce costs manufacturers often produce a range of finished components from the same 'rough' components. For instance, a range of axial fan blades can be produced from the same rough casting by machining the casting to different lengths. There is a point when this operation impairs the performance of the fan and therefore a new size of casting is required. This itself requires a different casting die, the cost of which will be reflected in the price of the component and subsequent discontinuity in the cost function.

3.3.3 Operating Costs.

The major contributors to the operating cost of HVAC systems are:

1. Energy costs.
2. Maintenance costs.
3. Labour costs.
4. Water costs.
5. Insurance.

Fuel tariff structures are often based upon the peak demand on the supply, type of consumer, season and region, with electricity charges in particular dependent upon the class of consumer, size of consumers load and pattern of demand (CIBS, 1977). The most reliable procedure for estimating the peak demand and annual energy consumption is by hourly integration of the calculated consumption.

Maintenance costs include the cost of repairs, cleaning, lubricating, adjusting, painting, inspecting and testing. Two categories of maintenance cost exist, the direct cost which covers labour and materials and the on-cost which includes supervision, sick leave, holiday pay, national insurance contribution and support services such as workshops. Milbank (1971) suggests that the on-costs are approximately 40% of the direct charge for maintenance.

Estimates of maintenance cost are often related to the floor area of the building, yet this is of little use in a component based optimised design procedure as the size of component is unrelated to the maintenance charge. The most appropriate method avalable for use in optimised design is that suggested by Milbank (1971). This relates the annual maintenance charges for a group of components to an appropriate parameter such as the rated capacity of the boiler plant group (table 3.2). An adaptation of this approach for use with a component based design procedure is discussed in section 5.5.

In addition to maintenance, the efficient running of HVAC systems is dependent upon skillful monitoring and adjustment by trained personnel. The cost of training personnel and their remuneration is related to the complexity of the system and is therefore a factor which should be considered when comparing different schemes.

Water costs are often estimated from the rateable value of the building and as such the only factor which is of relevance in optimised design is the cost of water treatment in controlling scale forming salts, corrosion and organic growth. This is of more importance to the comparison of different schemes than to the optimum component selection as the cost of treatment will be fixed for a given scheme.

A final factor to consider in formulating the system operating cost is the cost of insurance to cover breakdowns. The premium paid for such policies is often dependent upon the duty of the installed plant and therefore can influence the choice of components.

3.3.4 Life-cycle Costs.

Life-cycle cost analysis accounts for both capital and operating cost of the plant. Several life-cycle or economic evaluation techniques exist, the most common of which is simple payback period. This is the time taken for the investment to repay the initial expenditure and therefore is most applicable to evaluating heat recovery schemes. A more realistic calculation which includes the interest paid on the outstanding capital borrowed, is the discount payback period.

Group	Plant	Elements		
		Parameter	Unit	Coefficient (£/Unit)
Distribution	Fans, pumps, pipes, ducts, filters, valves, controls and heating or air-conditioning terminal units	Shaft power	kW	35
Lighting	Lighting, small power and main electrical distribution	Connected load	kW	10
Lifts	—	No. of passengers × floors	—	3·5
Boiler Plant	Boilers, burners, fuel storage, chimneys	Rated capacity	kW	0·85
Refrigeration Plant	Machines and cooling towers	Rated capacity	kW	1·2
Plumbing and Drainage	Baths, sinks, showers and WC's etc.	Number	—	7·7

table 3·2, Coefficients for Annual Maintenance Charge, (after C.I.B.S., 1977)

Equipment Item	Median Years	Equipment Item	Median Years
Air conditioners		Coils	
Window unit	10	DX, water, or steam.	20
Residential single or split package	15	Electric	15
Commercial through-the-wall	15	Heat exchangers	
Water-cooled package	15	Shell-and-tube	24
Computer room	15	Reciprocating compressors	20
Heat pumps		Package chillers	
Residential air-to-air	b	Reciprocating	20
Commercial air-to-air	15	Centrifugal	23
Commercial water-to-air	19	Absorption	23
Roof-top air conditioners		Cooling towers	
Single-zone	15	Galvanized metal	20
Multizone	15	Wood	20
Boilers, hot water (steam)		Ceramic	34
Steel water-tube	24 (30)	Air-cooled condensers	20
Steel fire-tube	25 (25)	Evaporative condensers	20
Cast iron	35 (30)	Insulation	
Electric	15	Molded	20
Burners	21	Blanket	24
Furnaces		Pumps	
Gas- or oil-fired	18	Base-mounted	20
Unit heaters		Pipe-mounted	10
Gas or electric	13	Sump and well	10
Hot water or steam	20	Condensate	15
Radiant heaters		Reciprocating engines	20
Electric	10	Steam turbines.	30
Hot water or steam	25	Electric motors	18
Air terminals		Motor starters	17
Diffusers, grilles, and registers	27	Electric transformers	30
Induction and fan-coil units	20	Controls	
VAV and double-duct boxes	20	Pneumatic	20
Air washers	17	Electric	16
Duct work	30	Electronic	15
Dampers	20	Valve actuators	
Fans		Hydraulic	15
Centrifugal	25	Pneumatic	20
Axial	20	Self-contained	10
Propeller	15		
Ventilating roof-mounted	20		

table 3·3, Equipment Service Life, (after ASHRAE, 1984).

The interest rate used in economic calculations is the interest paid on borrowed capital, or for self financed projects the interest available from investment. The lowest interest rate for low risk projects is the prevailing minimum lending rate, higher risk projects perhaps warranting higher rates.

Although discount payback period calculations include interest charges, the effect of cash flows after the payback period are ignored. A superior method of analysis which solves this problem is the Net Present Value (NPV) calculation. This is the total value of the project over the life of the building expressed in terms of prices at the outset of the project.

The inclusion of replacement costs in NPV calculations requires an estimate of the expected life of the component. Equipment life is highly variable, those values given by ASHRAE (1984) (table 3.3), allow for 'diverse equipment applications, the preventive maintenance given, the environment, technical advancements of new equipment and personal opinions'.

NPV is the most realistic economic comparator for use in HVAC system design: however, discounted payback period is in common use and is applicable to heat recovery applications and therefore it has also been included as an objective function.

3.3.5 Mathematical Characteristics.

All the objective functions implemented in this reseach, with the exception of discount payback period, have in general optimum solutions which tend towards the bounds of the problem variables. Solutions for system energy consumption and operating cost tend towards the largest size of components as these are inclined to be the most efficient components. Conversely, solutions for capital cost tend towards the smallest components since the larger the component the more expensive it is. Although the net present value function is a combination of capital and operating cost, solutions for this tend towards the largest components as the operating cost is inclined to be the dominant element. Discounted payback period is the only objective function which may have solutions in the mid-range of the problem variables.

Some objective functions such as capital cost may prove to be linear, but the discounted payback period is invariably non-linear: this and the discontinuous nature of the capital cost function lead to a general description of the objective functions as non-smooth non-linear functions.

The characteristic of solutions to tend towards the variable bounds suggests that the optimisation algorithm could be relatively simplistic in its methodology. Yet introducing constraints and discrete problem variables increases the complexity of the problem and therefore that required of a solution algorithm. An example problem in two dimensions is illustrated in figure 3.6. Problem variables x_1 and x_2 are discrete, the discrete increments illustrated by the lines on the grid. The objective function $F(\underline{X})$ is linear, discontinuous in variable x_2 and has a global optimum, $\underline{X}_g$ at the minimum value of each variable. Finding such an optimum would prove a simple optimisation problem, although the discrete variables and discontinuous objective function would restrict the choice of solution algorithm. ıntroducing the non-linear constraint, $c(\underline{X})\geq 0$ increases the complexity of the problem by restricting the solution to the local optimum $\underline{X}_L$. On encountering the constraint the solution algorithm would have to have the ability to follow the contraint towards the optimum, $\underline{X}_L$.

3.4 System Definition and System Simulation.

The system definition and simulation techniques are central to the optimised design software. If the optimised design software is to form part of future high level software, definition of the system configuration must be such as to allow integration of the technique with graphics software. The system simulation procedure must predict the system operating point such that this can be used to calculate the energy consumption of the system, identify undersized components and calculate the value of other constraint functions.

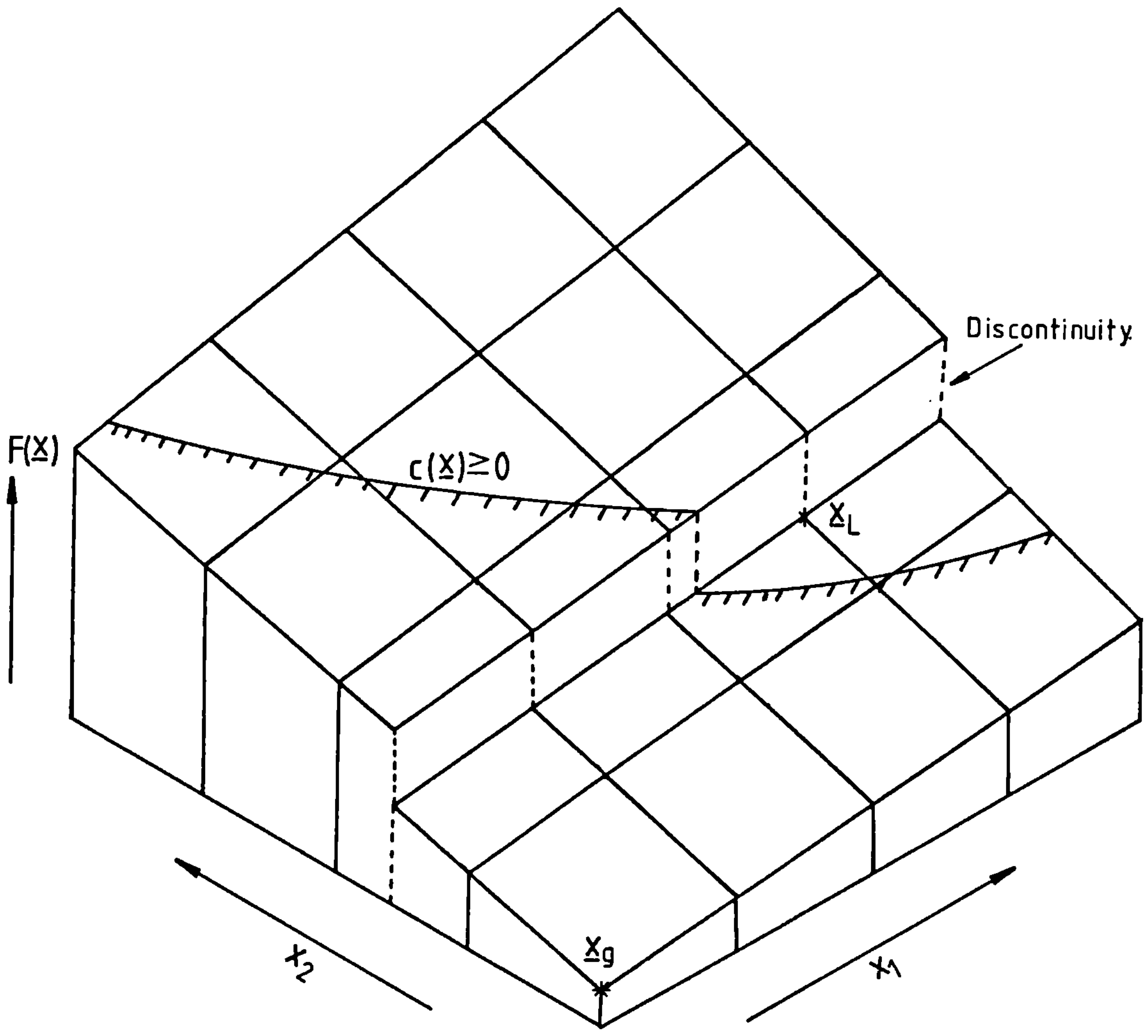


figure 3·6, Example Problem.

Several menu based system definition and simulation techniques exist in which the system configuration is chosen from a menu of systems and control strategies. This approach is of limited use in optimised design as the system definition is fixed and does not allow for hybrid systems or innovative design. A more flexible approach is the component based method, in which the system definition is related to the engineers schematic diagram and is built up from a menu of components. This close relationship with the engineers schematic diagram should enable integration with graphics routines which facilitate rapid system definition.

A steady state simulation procedure is of sufficient accuracy for most HVAC design applications as the time constants of the HVAC system are significantly less than those of the building (ASHRAE 1975). The disadvantages are that the dynamic response of the system 'start up' is not modelled and the stability of control schemes is impossible to establish.

In choosing a simulation method for use with optimised design software, it is important to consider the availability of component models and to establish the extent to which they represent the component performance as predicted by the manufacturer. This is essential if the optimum size and manufacturer of the components are to be identified. The most common component performance models are steady state input/output form using either manufacturers published data or the laws of heat transfer and fluid dynamics.

To summarise, a system simulation procedure for use in the optimised design of HVAC systems should have the following attributes:

1. A component based system definition which is related to an engineer schematic diagram.
2. A steady state simulation which employs lumped parameter input/output models.
3. A simulation procedure which can be used to calculate the system energy consumption, calculate the value of the constraint functions and can be used to indicate the undersizing of components.

A simulation procedure which includes all of these features is SPATS (Simulation of the Performance of Air-conditioning and Thermal Systems), developed by Murray (1984) at Loughborough University of Technology.

3.5 Proposals for an Optimised Design Procedure.

The objectives of this research are to develop a component based procedure for the definition and solution of an optimised design problem in HVAC systems. Before a design procedure and solution algorithm can be formulated, the principal design parameters and fundamental characteristics of the objective and constraint functions must be identified. It should be emphasized that it is the characteristics of the problem that are important in developing a procedure and therefore to avoid obscuring development due to over complication, simplifications have been made throughout the problem definition, although care has been taken to ensure that the characteristics of the problem are not inhibited and that the procedure forms a basis for future development.

It has been the intention from the beginning of this research to employ the techniques of system definition and simulation described by Murray (1984). The two reasons for this are, firstly it is the most appropriate method available and secondly this research is part of the continuing development of HVAC design software at Loughborough University and as such must be integrated within the existing software framework.

It is desirable that the form of software should be such as to allow the characteristics of the problem to be fully investigated and that it can be developed into a user friendly package. Maximum flexibility in problem definition can be achieved through a modular approach which allows the investigation and control of individual elements of the problem. By combining the modules to perform tasks to a predefined default, such software is easily developed into a more practicable package and is therefore the approach adopted in the development of the optimised design software.

The proposals for the investigation and development of the optimised design procedure are:

1. **System and Problem Variable Definition.** - It is proposed that the method of system definition described by Murray (1984) is used to define the system configuration. Definition of the problem variables requires a method of identifying the system variables of the configuration definition which are included in the optimised design problem. Each variable must be defined as discrete or continuous and the matching dimensions of adjoining components must form a single design variable.

2. **Component Models.** - The simulation procedure defined by Murray (1984) uses the most commonly available steady state input/output form of component performance model. This is the most appropriate format for use in optimised design as the component performance can be derived from the manufacturers test data. The optimised design procedure requires the development of the component models to include steady state energy terms, component constraint functions, capital and maintenance cost models.

3. **System Simulation and Component Undersizing.** - Murray (1984) implemented an algorithm for the solution of the equations which describe the performance of the components in HVAC systems. Although robust the algorithm is far from ideal for use in an optimised design procedure as it is slow to converge to a solution and therefore the development of a simultaneous solution algorithm is subject to continuing research at Loughborough University.

   Identification of undersized components is related to the formulation of the component performance equations, the definition of the component performance envelope and the bounds on the controller variables. These should be used to formulate a procedure for identifying undersized components which can be used with a variety of simulation solution algorithms. The inclusion of component undersizing as a mathematical constraint requires the development of a procedure which assigns numerical significance to the severity of undersizing.

4. **Objective and Constraint Functions.** - The procedure should include the definition of objective functions for net energy consumption, primary energy consumption, capital cost, operating cost, net present value and discounted payback period. These have been selected to provide a range of objective function characteristics with which to develop an optimistaion algorithm. The procedure should allow the definition of subsequent objective functions without major changes in the software.

Modelling of the system energy consumption requires a method of definition which identifies the energy terms of each component included in the system model. Each term included, must be associated with a type of fuel and should be allowed to be added to or subtracted from the system energy consumption. Similarly, the procedure should include a method of defining the constraint functions of each component to be included in a given design problem.

5. **Optimisation Algorithm.** - An algorithm is required for the simultaneous optimisation of the 'size' of components in an HVAC system. It is desirable that the software structure should enable several different optimisation algorithms to be implemented without any change to the general program. Each optimisation problem can have any combination of discrete and continuous variables. The objective and constraint functions are in general non-smooth non-linear functions although some configuration constraints may be sparse non-linear function. First and second derivatives of the objective and constraint functions are unobtainable. The dependence of the optimised design procedure on the system simulation to find the operating point of the plant, results in a long calculation time for the value of the objective and constraint functions.

Chapter 4. GENERALISED PROBLEM DEFINITION.

Definition of an optimised design problem is in two parts, definition of the system configuration and definition of the optimised design parameters and criteria.

Flexibility in the use of software is often sacrificed for the sake of improved speed. This results in so called 'black box' software and consequently little or no understanding of the processes performed. Black box software is undesirable in a design environment as the loss of flexibility inhibits innovative design.

The method of system definition described here is the work of Murray (1984) and reflects the same flexibilty in system definition as is available in manual design methods. This approach has been maintained in developing a method of defining the optimisation problem, with flexibility being of prime importance.

Each system configuration definition is written to a data file for subsequent use in the simulation, optimisation and for interactive redefinition. Similarly, the definition of the optimised design problem, including the problem variables, constraints, energy model and general design parameters are held on a further data file and may be recalled for redefinition or use in the optimisation.

4.1 Generalised System Definition.

Definition of the system configuration is by a network of 'nodes' and 'arcs', the nodes representing the components and the arcs the 'connection' of system variables between components. This concept is closely related to the engineers schematic diagram in which the nodes are the components, but the arcs represent the connecting pipe or ductwork.

The method of describing a system configuration by network techniques is best explained with the aid of an example. Figure 4.1 is a schematic diagram for a sub-system consisting of a run-around heat recovery unit, supply and extract fans and a heating coil proportionally controlled by the action of a diverting valve. The hydrodynamic characteristics of the system are not modelled in this example.

Each component has several associated variables which must be uniquely identified within the system. Where the same variable appears at more than one component it is assigned the same arc-variable number. For example, the air temperature leaving the run-around coil (node [3]), is assigned the same arc-variable number as the air temperature entering the heating coil (node [5]), 17 = ta-out = ta-in. The arc-variables can be numbered in an arbitrary but consistent manner. No physical meaning is attached to the arc-variables, they are simply information flow lines. This allows the definition of variables other than thermofluid variables and is particularly useful in modelling controller signals and actuator inputs.

4.1.1 Exogenous Variables.

Some arc-variables are external to the simulation. In a full system definition these are the 'driving' variables of the system, such as the weather parameters and zone conditions. The number of exogenous variables equals the number of arc-variables minus the number of system equations, this ensures the simulation problem is for 'n' unknowns in 'n' equations.

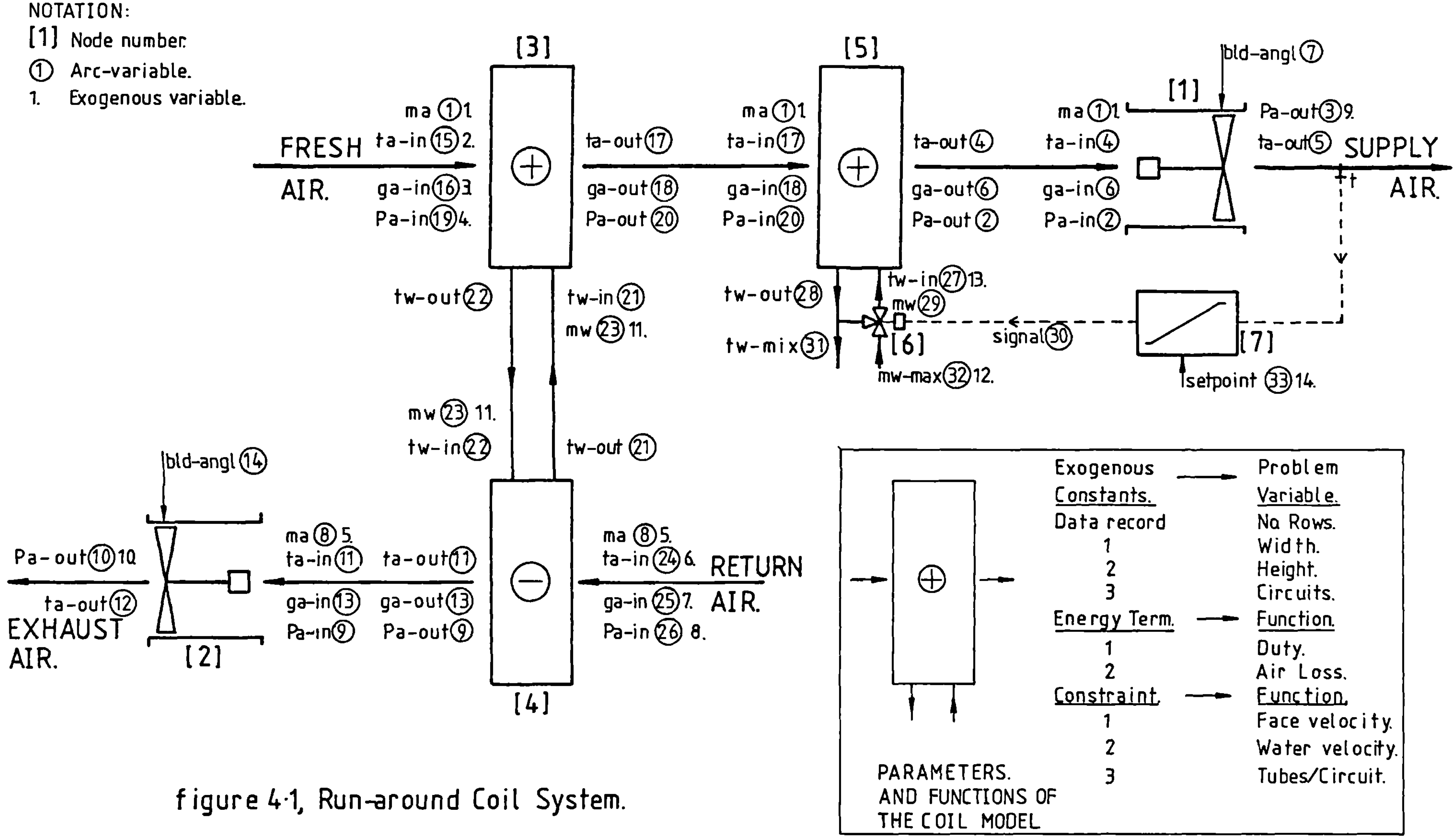


figure 4·1, Run-around Coil System.

4.1.2 Component Constants.

Component performance models consist of the describing equations and a set of data constants. The data constants can take any of three forms, constants, polynomial curvefit coefficients or exogenous constants. The constants and polynomial coefficients are stored in structured data files. Exogenous constants are a means of reducing the amount of data stored: normally they are constants which would be fixed in a manual design process and are often related to the selection of the component. For example, the coil model has a separate data file for every coil row, the remaining coil geometries of height, width and water circuits are specified as exogenous constants.

4.1.3 Network Definition.

The system illustrated in figure 4.1 has seven nodes. Each node is sequentially selected from a menu of components and indexed in the two dimensional array NET, by each node forming a row in the array (table 4.1). The first column of NET gives the node type, this is used to call the component initialisation routine which defines the number of variables, exogenous constants, data constants, polynomials and equations associated with that node. The additional information contained in the initialisation routine which relates to the optimisation problem definition is described in section 4.2.

An index of the arc-variables associated with each component is held in the array NET. The initial estimates of the value of the arc-variables and their upper and lower bounds are read from a selected data record and held in arrays ARCVAR, UB and LB respectively. Exogenous constants are numbered sequentialy in NET and their values held in array EXCON. Indexes for data and polynomial constants are held in CONST and NETCOE respectively. The final column of NET is an index of the number of residual performance equations for each node. Exogenous variables are indexed in EXVAR. Additional character arrays of VNAME, NAMEXC and FNAME, hold the variable exogenous constant and data record names.

Node:	Node Type.	Number of Arc-variables.	Index of Arc-variables.	Number of Exogenous Constants.	Index of Exogenous Constants.	Number of Data Constants.	Index of Data Constants.	Index of Polynomials.	Number of Residual Equations.
1	2	7	1 2 3 4 5 6 7	1	1	1	1	1 2	2
2	2	7	8 9 10 11 12 13 14	1	2	1	2	3 4	2
3	6	10	15 16 17 18 19 20 1 21 22 23	3	3 4 5	5	3 4 5 6 7	5 6 7 8	4
⋮									

Array: NET.

Array: FNAME.
90KG569
90KG569
5/8+04

Arc-Variable:	Array: VNAME.	Array: ARCVAR.	Array: LB.	Array: UB.
1	ma	0·7200e+1	0·5000e+1	0·1000e+2
2	Pa-in	-0·2000e+3	-0·5000e+3	0·0000e+0
3	Pa-out	0·5000e+3	0·4000e+3	0·6000e+3
⋮				

at Node:	Array: NAMEXC.	Array: EXCON.
1	speed	1470·0
2	speed	1470·0
3	width	1·50
3	height	1·75
⋮		

[0·9000e+0, 0·9000e+0, 0·5800e-4 - - - - - - - - -]

Array: CONST.

table 4·1, Network Arrays.

4.2 Definition of the Optimisation Problem.

Definition of the optimised design parameters is in four parts, identification of the problem variables, definition of the design constraints, definition of a system energy model and assignment of general design data. A suite of directories has been developed which enables a flexible and interactive definition of these parameters and which relates the problem variables to the system parameters of the simulation procedure. As the formulation of the objective function is fixed within the software, no complex directory is required for its definition, the choice of objective function requiring only identification of the appropriate subroutine.

4.2.1 Definition of the Problem Variables.

The system parameters of the configuration definition which appear as optimised design variables are the parameters related to component size, the controller settings and the fluid variables which would normally remain fixed in the simulation. The size of component is specified by the exogenous constants and a data set held in data record, whereas the controller settings and fluid variables are represented by the exogenous variables. Definition of the optimised design problem variables is concern with identifying these parameters within the optimisation procedure.

It is possible to develop an optimisation procedure which operates directly on the system parameters, yet this is undesirable in a research environment as development of optimisation software is more transparent if the optimisation problem variables remain distinct from the simulation parameters. This also has the advantage that parallel development of the simulation software can continue with any changes in software affecting only the 'interfacing' subroutines of the simulation and optimisation software.

A characteristic of component based software is that the most flexible format of index array is one in which the rows of arrays are formed by the nodes in the network, as in the array NET. However optimisation algorithms act on the problem variables and not the nodes, hence the most efficient arrangement here is to have a variable directory which relates the problem variables to their assigned nodes. A dual function of such a directory is to indentify the form of the system parameters associated with each node ie: exogenous constants, exogenous variables or a data file.

VARDIR is a two dimensional array forming a directory of problem variables, the index for each problem variable taking a row in the array. Initially the array is formed node by node in the simulation - optimisation interfacing subroutine <setup>. Variables associated with data files are defined first, followed by exogenous constants and lastly exogenous variables. Subsequent definition of adjacent components, discrete data and exclusion of unwanted exogenous variables is performed interactively.

The use of the array VARDIR in defining variable types can be clarified by example. Table 4.2 is an example definition for the run-around coil system illustrated in figure 4.1. The adjoining dimensions of the adjacent coils at nodes [3] and [5] must match when installed and therefore the width and height of both coils form only one variable each.

The number of nodes to which each variable is assigned is indexed in column two of VARDIR. The next 'n' pairs of numbers in each row defines the nodes themselves and related variables. The first number of a pair is the node number and the second a variable index. Exogenous variables are distinguished from other variable types by a non-zero value in the final column of VARDIR, this being the arc-variable number of the exogenous variable. A zero value indicates that the problem variable is either associated with an exogenous constant or a data file. Problem variables assigned to data files are recognised by a zero variable index, a non-zero value being an index of an exogenous constant within the component model. The position of the variable name within the array DVARNM is indexed for all variable types by the value of the variable index +1.

Problem Variable:	Number of Discrete Values.	Number of Assigned Nodes.	Index of Nodes and System Variables.				Exogenous Variable Index.
			Node.	Variable. Index.	Node.	Variable. Index.	
1	3	1	1	0	- - - - - -	- - - - - -	0
2	1	1	1	1			0
⋮	⋮						
5	5	1	3	0			0
6	5	2	3	1	5	1	0
⋮	⋮		Matching Dimension				
18	0	2	3	4	3	4	23
⋮	⋮						

Array: VARDIR.

Array:STPVAR.

90	100	112	- - - - -	
1470				
⋮				
2	4	6	8	10
1·0	1·25	1·50	1·75	2·0
⋮				

Node:					
1	90KG569	100KG569	112KG569	- - - - -	- - - - -
⋮	⋮				
3	5/8t02	5/8t04	5/8t06	5/8t08	5/8t10
⋮	⋮				

Array:COMFIL.

axialfan
⋮
h/c-coil
⋮

Array: COMNAM.

fan-dia	speed	- - - - - -
⋮		
no. rows	width	height
⋮		

Array:DVARNM.

table 4·2, Problem Variable Arrays.

For example:

A variable associated with a data record: The second column of VARDIR indicates that problem variable 5 is associated with a single node, which is node [3] as indexed in column three. As the exogenous variable index, held in the final column of VARDIR is zero, the variable is associated with either a data file or exogenous constant. In this case, as the variable index held in column four of VARDIR is zero, the variable is associated with the component data file of node [3]. The position of the variable name in array DVARNM is assertained by adding 1 to the value of the variable index, hence at node [3] the problem variable is the 'no. rows'.

An exogenous constant: These design problem variables are distinguished from those associated with data records by a non-zero value for the variable index. Problem variable 6 is defined at two nodes, node [3] and node [5]. A zero exogenous variable index in the final column of VARDIR and non-zero variable index values in columns four and six, indicates that the variable is associated with exogenous constants at both nodes [3] and [5]. The exogenous constants are the first exogenous constants defined in the component models as the variable indexes are 1. Again the location of the variable name in DVARNM is indexed by adding 1 to the variable index.

An exogenous variable: Problem variable 18 is identified as an exogenous variable as the index of the last column of VARDIR is non-zero, a value of 23 indicating that this exogenous variable is the arc-variable no. 23. Columns three and five of VARDIR indicate that this is associated with two nodes, node [3] and node [4]. The name of the variable is indexed in the same fashion as other variable types by adding 1 to variable index values.

4.2.2 Definition of the Constraints.

The upper and lower bounds of the problem variables are kept in the arrays DVARUB and DVARLB respectively and are assigned values during the interactive definition of discrete data. Values of the problem variables themselves are held in the array DVAR.

Table 4.3 illustrates the constraint formulation for the run-around coil example of figure 4.1. Obviously all components are required to be correctly sized, but the only components whose selection is additionally restrained are the heating and cooling coils, each coil allowing up to three additional constraints of, a restriction on coil face velocity, a restriction on the water velocity and a configuration constraint which ensures there are sufficient tubes to form the required number of water circuits. All three constraints have been assigned to the example problem definition for the coil at node [5], but only the configuration and water velocity constraints have been assigned for nodes [3] and [4].

To facilitate easy handling of the constraints within the optimisation algorithms, the values of the constraints are held sequentially in the array DCON and their corresponding upper and lower bounds in the arrays DCONUB and DCONLB. The constraints assigned to the problem and their position within the array DCON are defined sequentiallly node by node and indexed in the directory CONDIR, each node forming a row in the directory.

The first column of CONDIR defines the number of constraints assigned to the problem for each node. The next 'n' pairs of numbers in each row index the particular constraint function within the component model and its position within the array DCON. For example, in table 4.3 the coil at node [4] has two of the possible three constraint functions assigned to the example problem. From column two of CONDIR the first of these is the second constraint function of the coil model and from column three is the third constraint in the problem definition and hence has a value which is held in position three of DCON.

The names of the constraint functions are not held permanently in a separate array, but to save on storage space are recalled from the component initialisation routines as required and held temporarily in the array CONNAM. The position of the names in CONNAM are indexed via the node index of CONDIR, hence in the example the second constraint of node [3] has the name 'watervel'.

Problem Variable:	Array: DVAR.	Array: DVARLB.	Array: DVARUB.
1	100	90	112
2	1470	1470	1470
⋮	⋮	⋮	⋮
5	6	2	10
6	1·75	1·25	2·0
⋮	⋮	⋮	⋮
18	4·0	2·0	6·0
⋮	⋮	⋮	⋮

Number of Designated Constraint Functions.

Index of Constriants. at Node. in Problem.

Node:							
1	0	- -	- -	- -	- -		
⋮	⋮						
3	2	2	1	3	2		
4	2	2	3	3	4		
5	3	1	5	2	6	3	8
⋮	⋮						

Array: CONDIR.

Array: CONNAM.
face vel
watervel
circuits
⋮

at node 3.

Constraint:	Array: DCON.	Array: DCONLB.	Array: DCONUB.
1	1·5	0·0	1·8
2	0·6	0·0	1·0
3	1·2	0·0	1·8
4	0·5	0·0	1·0
5	2·5	0·0	2·5
⋮	⋮	⋮	⋮

table 4·3, Constraint Arrays.

4.2.3 Definition of the System Energy Model.

A directory of energy terms has three functions, to define the terms active at each node, assign a fuel type to each term and specify whether to add or subtract the value of the term in the system energy model.

Table 4.4 is an example of an energy model definition for the system illustrated in figure 4.1. The example definition is for the run-around coils only, with all other components in the system excluded from the model. The active terms are defined node by node by each node forming a row in the array ENGDIR. The first column of the array gives the number of terms active at each node and the next 'n' columns the terms themselves.

For instance, two energy terms are associated with the coil model, the first of which is the coil duty and the second the energy related to the air pressure loss across the coil. In the example definition, column one of ENGDIR indicates that both terms are assigned to the problem for node [3] but only one is assigned at node [4]. From the second column of ENGDIR the term at node [4] is the second energy term of the coil model, the air pressure loss term.

The fuel and addition/subtraction terms are held in array ENGDRC, each node forming a row in the array. The associated terms in ENGDIR and ENGDRC are related by the terms holding corresponding positions in their respective arrays. For example, the single term assigned for node [4] has a related addition/subtraction/fuel term of (-e) as held in column one of ENGDRC. The first character of a pair of ENGDRC terms defines whether to add or subtract the value of the term in the system energy model, the second character defining the fuel type associated with the term.

As for the constraint function names, the names of the energy terms are not held permanently in an array but are recalled from the component initialisation routines using the array ENGNAM as they are required. The positions of the names in the array correspond to the position of the term in the component model, hence in the example the second energy term of node [3] has the name 'airloss'.

Node:	Number of Active Functions.	Index of Functions.	
1	0	- - -	- - -
2	0		
3	2	1	2
4	1	2	
5	0		
⋮	⋮		

Array:ENGDIR.

Node:	+/- Term.	Fuel Type.	+/- Term.	Fuel Type.
1	- - -	- - -	- - -	- - -
2	- - -	- - -	- - -	- - -
3	(+	o)	(-	e)
4	(-	e)		
5		⋮		
⋮		⋮		

Array:ENGDRC.

Array:
ENGNAM

duty
airloss
⋮

at node 3.

table 4·4, Energy Model Arrays.

4.2.4 General Terms in the Optimisation Problem.

Table 4.5 illustrates the definition of the general design data. Array FUELS holds the fuel prices in pounds/MJ, whilst the array PRIRAT holds the equivalent primary energy ratios. These arrays form part of the data base of default values, which are held in a structured data file and can be recalled for updating or for use in a particular design problem. The other default values in the data base are the building life BLDLIF, interest rate INTRST, and the number of hours assigned to each time step in the simulation TIMPRD.

The service life of each component is read from the component initialisation routine and held in the array SRVLIF. The logical array CSTFIL is also formulated by calling the initialisation routines, this array indicates the existence of a capital cost model for each component (logical value=.TRUE.) and is necessary as during development of an optimisation procedure the less significant components will not have capital cost models.

The array PBNODE indexes the components which are included in the payback period calculations (index value = 1). This ensures correct sizing of all components as the performance of the complete system can be simulated without requiring every component to be part of the payback calculation. This can be useful in evaluating the payback period of heat recovery equipment which form part of a larger system.

The component capital cost data which is read from the component capital cost data files, is held in the array SYSCST and is arranged in a format which allows two rows of data per component.

	FUELS	PRIRAT
Electricity:	1·06	3·82
Gas:	0·31	1·07
Oil:	0·51	1·09
Coal:	0·22	1·03

Array: FUELS. Array: PRIRAT.

BLDLIF = 30 : INTRST = 10 : TIMPRD = 1

Node:	SRVLIF	PBNODE	CSTFIL
1	20	1	.true.
2	20	1	.true.
⋮	⋮	⋮	⋮
6	15	0	.false.
⋮	⋮	⋮	⋮

Array: SRVLIF. Array: PBNODE. Array: CSTFIL.

	Array: SYSCST.
1	1470, 3, 5, 14, 1989 - - - - -
2	Two Rows of Polynomials and Data Constants per Node.
3	⋮
⋮	⋮

table 4·5, General Design Data.

Chapter 5. COMPONENT MODELS.

Component models for use in optimised design software have three elements, the component performance model, the cost model and the constraint models. The format of the component performance model is influenced by the simulation methodology and its definition of the system operating point. The format of the cost and constraint models are less rigorous but as for the performance models they must be uniform in accuracy and sophistication as the accuracy of the solutions is dictated by the accuracy of the least accurate and sophisticated model in the system.

5.1 Component Performance Models.

The purpose of component performance models is to represent the performance of the component as predicted by the manufacturer and to allow the development of the component energy model. The type of performance model used by SPATS is the steady state lumped parameter input/output model. This is the most widely available format and lends itself to the application of manufacturers published data.

It is rare that the dynamic performance of components is tested by manufacturers. Most performance tests are for steady state conditions and may be for only part of the component operating range. Care must be exercised in developing component models from published data as the data represents the performance of the component under specific test conditions. The performance of the component can not be guaranteed when it is installed in systems not represented by the test installation and when its operation is for conditions other than the test conditions. A further limiting factor in developing performance models is that unless the tests are performed to recognised standards there may be no indication of the accuracy of the published data and hence accuracy of the model.

Fortunately growing pressure is forcing manufacturers to comply with standards of manufacturing quality and component testing and to present test data in a uniform format. This is most evident in the air moving section of the industry with the recent introduction of the BSI quality assurance systems, BS 5750 (1979) which assess the ability of manufacturers to produce and test their products. Test standards are also moving towards specifying tests which reflect the performance of components in different installations, as in the new fan performance test standard BS 848 (1980).

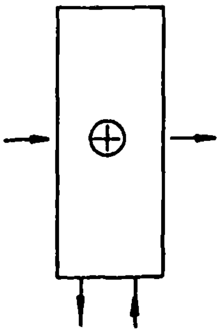
Manufacturers data lends itself to modelling the performance of package equipment whose performance modelled from first principles would prove inefficient with complicated calculation procedures and a large number of describing equations. Conversely, the performance of simple components such as diverting tees is best modelled using the established laws of thermo-fluid dynamics. This also applies to more complex components for which only peak load data is available or for which the manufacturers data is over simplified with a consequent loss of system variables.

Modelling of empirical data in this research has been by the least squares polynomial curve fitting technique. The order of polynomial is not fixed by the component algorithm but may vary to give the best fit for each size and make of component. The suite of curve fitting routines developed during this research allow data to be entered via the keyboard or by digitising curves on a graphics tablet. The data is curve fitted and the resulting coefficients stored on a data file for subsequent loading into component data files. To ensure a good representation of the original data, the order of polynomial is optimised for each size of component and the fit checked visually with the aid of graphics routines for the presence of spurious data and a poor fit (appendix A.).

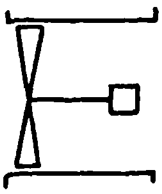
Development of a library of component algorithms and performance data is a major task and has required the effort of several researchers at Loughborough. A menu of component performance models available on SPATS is given in table 5.1 and details of the algorithms for the components used in the examples of chapter 10. are given in appendix B.

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-valvet	49 - contstep
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 -

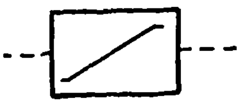
<u>For example:</u>



6 - h/c-coil = heating or cooling coil.



2 - axialfan = axial flow fan.



37 - pcontrol = proportional controller.

table 5·1, SPATS Component Menu.

5.1.1 The Format of Describing Equations.

It is usual to formulate steady state input-output models explicitly for the outlet conditions. Consider the simple heat exchanger illustrated in figure 5.1. The outlet conditions of the component can be expressed as (figure 5.2):

$$t_{1out} = t_{1in} - (W_{min} / W_1).\varepsilon.(t_{1in} - t_{2in}) \tag{5.1}$$

$$t_{2out} = t_{2in} - (W_{min} / W_2).\varepsilon.(t_{1in} - t_{2in}) \tag{5.2}$$

where ε is the effectiveness of the heat exchanger and W the capacity rate of the fluids. Explicit equations can be solved sequentially but the solution of implicit equations requires an iterative approach or simultaneous solution.

The formulation of equations developed by Murray (1984) and used in SPATS allows the solution of both implicit and explicit equations. Both types of equation are cast in a residual form:

$$F_1 = 0 = W_1(t_{1in} - t_{1out}) - W_{min}.\varepsilon.(t_{1in} - t_{2in}) \tag{5.3}$$

$$F_2 = 0 = W_2(t_{2in} - t_{2out}) - W_{min}.\varepsilon.(t_{1in} - t_{2in}) \tag{5.4}$$

The residual equations become zero at the operating point of the exchanger, which is specified by the capacity rates and inlet conditions of the fluids. At values of outlet conditions other than the operating point the residual equations F_1 and F_2 have finite values which are used by the solution algorithm in a simultaneous search for the outlet conditions.

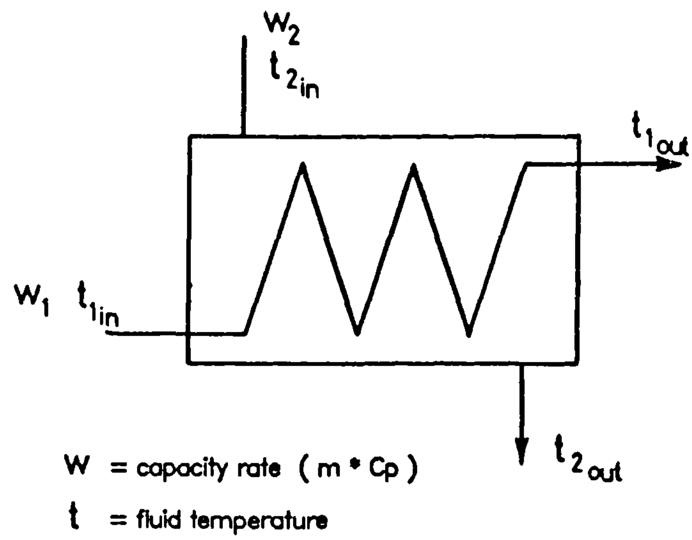


figure 5·1, Heat Exchanger.
(after Murray, 1984)

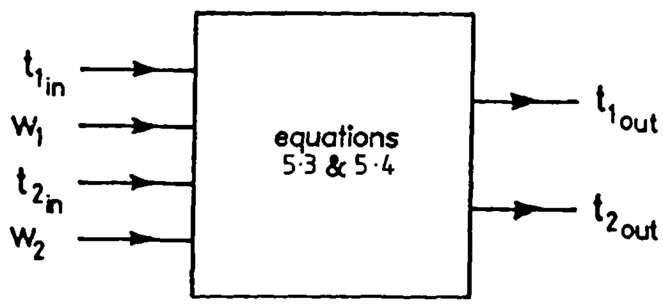


figure 5·2, Input-Output Model.
(after Murray, 1984)

5.2 Modelling Ancillary Equipment.

Items of plant are often assembled from main and ancillary components such as a fan and its drive motor. The performance of the ancillary component may not affect the simulated operation of the main component and therefore it is wasteful of calculation time to include the ancillary component in the simulation. For example, including a fan drive motor in the performance simulation of a fan and duct system will not affect the solution obtained for the pressure at outlet to the fan.

However ancillary equipment such as drive motors become important in energy and cost modelling and in ensuring the item of plant is not undersized. Ancillary equipment can be specified as separate components in the simulation or form part of the main component model, for example the fan motor can form a separate motor model or be part of the fan model. Both approaches increase the number of variables and complexity of the problem without significantly changing its characteritics. Therefore to allow the problem to be more transparent, ancillary equipment has often been excluded from the examples used in this research. These approximations do not inhibit development of an optimisation algorithm as in general the characteristics of the objective and constraint functions are unchanged.

5.3 Component Energy Models.

Several energy terms can be associated with each component. Each can fall into one of the three categories of direct, ancillary and extraneous. Direct energy terms by definition are expressed as net energy, but ancillary and extraneous terms can only be converted from useful to net energy when the efficiencies of other components in the system are known, (section 3.3.1).

Conversion of ancillary terms from useful to net energy requires the efficiency of the ancillary component and as a consequence the inclusion of the ancillary component in the problem definition. For example, in the fan model the impeller power is expressed as useful energy, to convert this to net energy requires the efficiency of the fan drive motor and therefore its inclusion in the simulation.

Consequently, excluding ancillary components from the simulation leaves the ancillary energy terms expressed as useful and not net energy. This however, does not adversely affect the characteristics of the problem, since the efficiency of most ancillary items such as drive motors, remain relatively constant. Therefore throughout this research ancillary components have been excluded from the problem definition as this does not affect the characteristics of the problem but in reducing the number of parameters yields a more transparent and manageable research problem.

A similar argument applies to extraneous energy terms used in the modelling of subsystems, in which the efficiency of components not included in the problem definition are required to convert values of useful energy to net energy. For example, if a supply fan is excluded from a problem definition, the energy required to overcome the pressure drop across the components in the supply duct, can only be expressed as useful energy if an estimation of of the fan efficiency is included in the analysis. This is best done by including in the problem definition a pseudo-component which is used only to define the efficiency. However, modelling the extraneous terms as useful energy greatly reduces the complexity of the problem without affecting its characteristics and therefore for the purposes of this research extraneous energy terms have been expressed as useful energy.

The complexity of the component energy model is dependent upon whether it is more appropriate to formulate it from the component performance model or from a fundamental thermo-fluid relationship. For example, the impeller power, described by a polynomial curve fit of the normalised power curve, forms part of the fan performance model. As there is no simpler thermo-fluid relationship by which to calculate the fan power, the fan energy model is formulated by interpreting the curve fit of normalised power and converting this to the appropriate units. Conversly, the algorithm which is used to calculate the coil operating point is highly complicated, yet the coil duty is easily derived from the air mass flow rate and difference in enthalpy across the coil.

The value of the energy terms over a given time period is calculated within the component subroutine and subsequent to the system simulation for that period. The value of each term is expressed in MJ of net energy or where simplified useful energy. The integration of the terms and their use in calculating energy costs and primary energy consumption is described in chapter 7.

5.4 Component Capital Cost Models.

Development of rigorous component cost functions has proved difficult due to the lack of available data. In a competitive market, manufacturers are reluctant to release cost data even when the information is used purely for research and therefore the component cost functions developed in this research cannot be considered as generally applicable to a wide range of manufacturers data. It is the authors experience that the presentation of cost data by different manufacturers is as diverse as the presentation of performance data. This adds to the difficulty of developing generalised capital cost functions and suggests that, as for some performance models, data preparation programs will be required to convert the data into an appropriate format. Although rigorous cost functions have not been defined, a generalised format of data storage has been developed.

5.4.1 Structure of Capital Cost Models.

Manufacturers normally present their cost data in a tabular form of discrete prices against the size of component. Some price lists can be curve fitted to reduce the amount of data handled whilst others are represented by complicated algorithms. This leads to two types of cost data, polynomial curve fit coefficients and sets of data constants. Both types are stored on structured data files which are analogous to the performance data files, ie: for every performance data file there is a cost data file, both addressed by the same record name. This presupposes that cost functions can be defined which allow the cost data to be arranged in this format. This however is not unreasonable as the size of component is defined by problem variables which are associated with the component exogenous constants and performance data files.

For example, the coil performance model has a data file for every number of coil rows, the remaining geometry being described by the exogenous constants of width, height and water circuits. Thus the coil cost model must be developed such that the cost functions are formulated from the coil width height and water circuits and in order to allow a cost data file per coil row, the coefficients of the cost functions must vary in relation to the coil depth.

Future software development may allow alternative and more efficient methods of formulation such as defining one data file for each range of components or price list. However the extra work involved in developing alternative strategies is not justified here as defining a cost data file for each performance data file has proved flexible enough for the purposes of this research.

The cost function values, expressed in thousands of pounds are calculated after the simulation thus ensuring the components are correctly sized before the costs are evaluated.

5.4.2 Modelling Ancillary Equipment Cost.

The cost of ancillary equipment such as drive motors is generally included within the price of the main component. Yet to allow the modelling of individual items, separate prices for the main and ancillary plant are required. However as in this research ancillary components are not modelled as separate items, the final price of the main and ancillary items can be calculated using a combined price list.

A problem which arises when ancillary components are not modelled separately is that their selection and hence price often depends upon the peak duty of the main component. For example, selection of a fan drive motor depends upon the peak duty of the fan. The simplified approach adopted in this research is to base the price of the component on its duty at the final time period in the simulation. This can be justified in that the characteristics of the objective function are unchanged and that the extra programming required to record the peak duty of the motor is not warranted as future development work should allow the separate modelling and simulation of ancillary equipment.

5.5 Maintenance Costs.

A maintenance cost calculation procedure which can be used to approximate component maintenance costs is that developed by Milbank (1971). This procedure estimates the direct maintenance charge for a group of components by multiplying a 'plant parameter' by a maintenance cost coefficient (table 3.2). In formulating the groups, Milbank investigated the effect of specifying smaller groups of components and found no significant change in the coefficients. It is on this premise that the coefficients are used in the research to approximate component maintenance costs.

The groups which are of main interest in this research are the major energy using groups. Here the plant parameter is often related to the duty of the main energy using component in the group. In a full system design these parameters can be used to estimate maintenance cost in the normal way, but when modelling subsystems in which the parameter related component is not included, an alternative strategy is necessary. The method of system energy modelling developed in this research lends itself to this problem as when the main energy using components are not included in the system, extraneous energy terms which are associated with the group plant parameters are automatically included in the problem definition and therefore can be used as plant parameters with which to calculate the individual component maintenance costs. For example, the maintenance cost for air distribution systems is based on the fan shaft power. In the design of a subsystem which does not include the fans, the extraneous energy terms relating to the air pressure drop across the components and hence fan shaft power are included in the system energy model and can therefore be used as parameters to estimate the maintenance cost of each component. To summarise, in a full system definition, maintenance costs can be calculated from the group plant parameters. Variation in maintenance charge with the size of individual components is reflected by a corresponding change in the value of the group plant parameter. In subsystems design the extraneous energy terms relating to the group plant parameters are used to approximate the maintenance cost of individual components.

To facilitate this approach two forms of component maintenance cost models have been derived. The first is for the components to which the group parameter is related. Here the maintenance charge for the whole group is calculated by multiplying the group plant parameter by the relevant cost coefficient. The second form of model occurs for all components other than the group plant parameter component. Here the individual component maintenance charge is calculated by multiplying the extraneous energy term related to the group plant parameter by the cost coefficient. To ensure this is only performed when the extraneous energy term is defined within the problem, the energy term directory ENGDIR is interrogated before the calculation are performed. Although this approach is far from ideal as it cannot be applied to all design problems, in the absence of more precise maintenance cost data it is the most applicable method available. Values of maintenance cost, calculated in the component subroutines are expressed in thousands of pounds per annum.

5.6 Constraint Models.

The form of constraints encountered in HVAC optimised design problems are: equality constraints, inequality constraints and range constraints. The majority of these are smooth non-linear functions, although certain configuration constraints are sparse non-linear functions.

Example equality constraint: Often water flow and return connections of heating coils are specified to be on the same side of the coil. This requires an even number of water tubes per circuit giving the sparse equality constraint:

fractional part of:[tubes / (2 circuits)] = 0

Example inequality constraint: The limiting values of fluid velocities can be expressed as smooth non-linear inequality constraints:

water velocity $\leq$ 1.8 m/s

Example range constraint: The number of water tubes in a coil must always be greater than or equal to the number of water circuits. This is represented by the range constraint:

$0 \leq$ (circuits-tubes) / (1-tubes) ≤ 1

The form of constraint definition adopted for this research is not critical as the constraints have been used as simple checks on feasibility. Therefore the format of constraint function adopted in this research is the range constraint:

$lbn_i \leq c_i(\underline{X}) \leq ubn_i$: $\quad i=1,2,...,m.$

Together with simple bounds on the variables this has allowed the definition of most constraints, although not always in the most rigorous fashion. However research suggests that the constraint functions must be used in a more sophisticated manner and therefore in future developments the constraints will require a more rigorous definition.

5.7 Complete Component Definition.

The complete description of a component model takes the form of six subroutines and two data files. These can be grouped into those associated purely with the optimisation and the more generally applicable simulation routines, initially developed by Murray (1984).

5.7.1 Simulation Subroutines and Data file.

The simulation group consists of an initialisation routine, an executive routine, a results routine and the performance data, file. The initialisation routine contains general information on the component models, such as the number of functions and variable names. This subroutine has been modified in this research to include information on the optimised design parameters, such as the number of energy terms, constraints and component service life. Executive subroutines return values of the residual equations which are called during the simulation solution.

Results routines are used to interpret the results of the simulation and present them in a form familiar to engineers, for example the heating coil routine converts temperature and mass flow rate into the coil duty expressed in KW.

5.7.2 Optimisation Subroutines and Data file.

The three component subroutines in this group return values for the energy functions, capital and maintence cost functions and constraint functions. Each component has an associated data file containing coefficients and constants for use with the capital cost functions.

Chapter 6. SOLUTION OF SYSTEM EQUATIONS.

Murray (1984) applied several available optimisation algorithms to the solution of the system performance equations. The procedure is similar for all algorithms and starts from an initial estimate of the solution from which successive approximations aimed at minimising the absolute value of the component residual equations can be generated.

Of those algorithms implemented it is the derivative methods which have proved to be the most efficient in solving the system performance equations. The most robust in solving a variety of problems is the Generalised Reduced Gradient Method. However this is far from ideal for use in an optimised design procedure, as it is slow to converge to a solution which results in a prohibitive calculation time. A faster but less robust algorithm is that based upon a Newton-Raphson iteration. Unfortunately the initial version of this is very unreliable and is only suitable for solving very simple problems. A scaled variable version of the algorithm has been implemented as part of this research in an attempt to develop a fast and reliable simulation solution algorithm. Although more robust than the unscaled version this algorithm is still unreliable and only suitable for solving problems consisting of a few simple components.

Both the Generalised Reduced Gradient and scaled variable Newton-Raphson algorithms are available for the solution of the system equations within the optimised design procedure.

6.1 Scaling of Variables.

Poorly scaled variables can cause the optimisation algorithm to fail to find a feasible point, or to be slow in converging to a solution. Ideally scaled variables will produce the same unit change in the objective function at the minimum for a unit change in each variable. Often this is impracticable and the best that can be achieved is to ensure that the variables are all of the same magnitude in the region of interest.

Scaling of variables is also important in calculating the derivatives of the objective functions as if the variables are badly scaled it is dificult to select a set of differencing intervals which produce a realistic change in objective function with each variable.

The method of scaling used by Murray (1984) in implementing the Generalised Reduced Gradient algorithm, is one in which the variables are transformed to be in the range -1 to +1 by the expression:

$$y_i = \frac{2 * x_i}{ub_i - lb_i} - \frac{lb_i + ub_i}{ub_i - lb_i} \qquad (6.1)$$

where lb_i and ub_i are the lower and upper bounds on the variable x_i and y_i is the transformed variable. Obviously care must be taken in selecting the bounds on the variables as crude limits which are wrong by several orders of magnitude can cause poor performance of the optimisation algorithm.

6.2 The Generalised Reduced Gradient Method.

The version of a Generalised Reduced Gradient Algorithm, used in the solution of the system equations is the GRG2 algorithm by Lasdon et al. (1978 and 1982). GRG2 solves nonlinear problems subject to equality or range constraints and simple bounds on the variables. The solution operates in two phases, if the initial estimate of the solution does not satisfy all the constraints a phase 1 optimisation is started. The objective function during phase 1 is the sum of the violations of all the constraints. This phase is terminated at either a feasible point or with a message that the problem is infeasible. In the context of HVAC system simulation, the phase 1 optimisation is used to find a feasible operating point by minimising the sum of the residual equations formed as equality constraints.

This can be expressed as:

$$\text{minimise} \qquad F(\underline{Xs}) = \sum_{i=1}^{n} f_i(\underline{Xs})$$

$$\text{subject to} \qquad f_i(\underline{Xs}) = 0 \qquad i=1,n$$

$$\text{and} \qquad lb_i \leq xs_i \leq ub_i \qquad i=1,n$$

where $\underline{Xs}$ is a vector of the 'n' HVAC system arc-variables and each f_i is a nonlinear constraint formed from the system residual equations.

Starting from an initial feasible point, phase 2 optimises the true objective function. Murray (1984) used this to optimise the value of exogenous variables, such as control settings, for an objective function formed from a simplified energy model. This has been modified for use in the optimised design procedure to include the net energy, primary energy and operating cost objective functions, but is however of little practical use as a different solution and therefore control setting is obtained for each time period in the simulation.

A more detailed discussion of the GRG2 algorithm is not appropriate to this thesis but can be found in Murray (1984) and by reference to Lasdon et al. (1978 and 1982).

6.3 The Scaled Variable Newton-Raphson Algorithm.

The generalised Newton-Raphson solution procedure uses a linear approximation to the function based upon the Taylor series expansion about the solution.

The solution procedure iterates between the expressions:

$$\underline{Xs}_{k+1} = \underline{Xs}_k - \underline{P}_k$$

$$\underline{P}_k \quad = \underline{J}_k^{-1} * \underline{f}(\underline{Xs}_k)$$

where $\underline{f}(\underline{Xs}_k)$ is a vector of residual equations evaluated at $\underline{Xs}_k$ and $\underline{J}_k$ is the Jacobian matrix of first partial differentials , $\partial f_i(\underline{Xs}) \,/\, \partial xs_j$ i,j=1,n. The solution of the equations $\underline{J}_k * \underline{P}_k = \underline{f}\ (\underline{Xs}_k)$ for the direction vector $\underline{P}_k$ and the criteria for convergence are described by Murray (1984).

Because the Newton-Raphson algorithm does not optimise subject to constraints or bounds on the variables, it is difficult to interpret the reasons for its failure as there is no indication of which variables have infeasible values or which variables remain unsolved. Unbounded variables can also lead to failure of the solution by allowing infeasible points to be generated during the search. This prompted development of a scaled variable procedure using the scaling method described in section 6.1. A simplified flow chart of the Newton-Raphson procedure with scaled variables is illustrated in figure 6.1.

The generation of infeasible points by the search can cause numerical problems in calculating the value of the residuals within the component executive routines. This is avoided by resetting the variables to their nearest bound and evaluating the residuals at this modified point. This is mathematically unsound as it alters the defined search direction $\underline{P}_k$, but has been found to work in practice.

As the procedure is based upon a linear approximation of the function at the solution, a good estimate of the solution is required as an initial guess. Providing this is obtained the algorithm works well for simple systems and appears to be more robust than the unscaled version. The primary reason for failure is ill-conditioning of the Jacobian matrix, which is often caused by numerical 'hunting' across the throttling range of a controller.

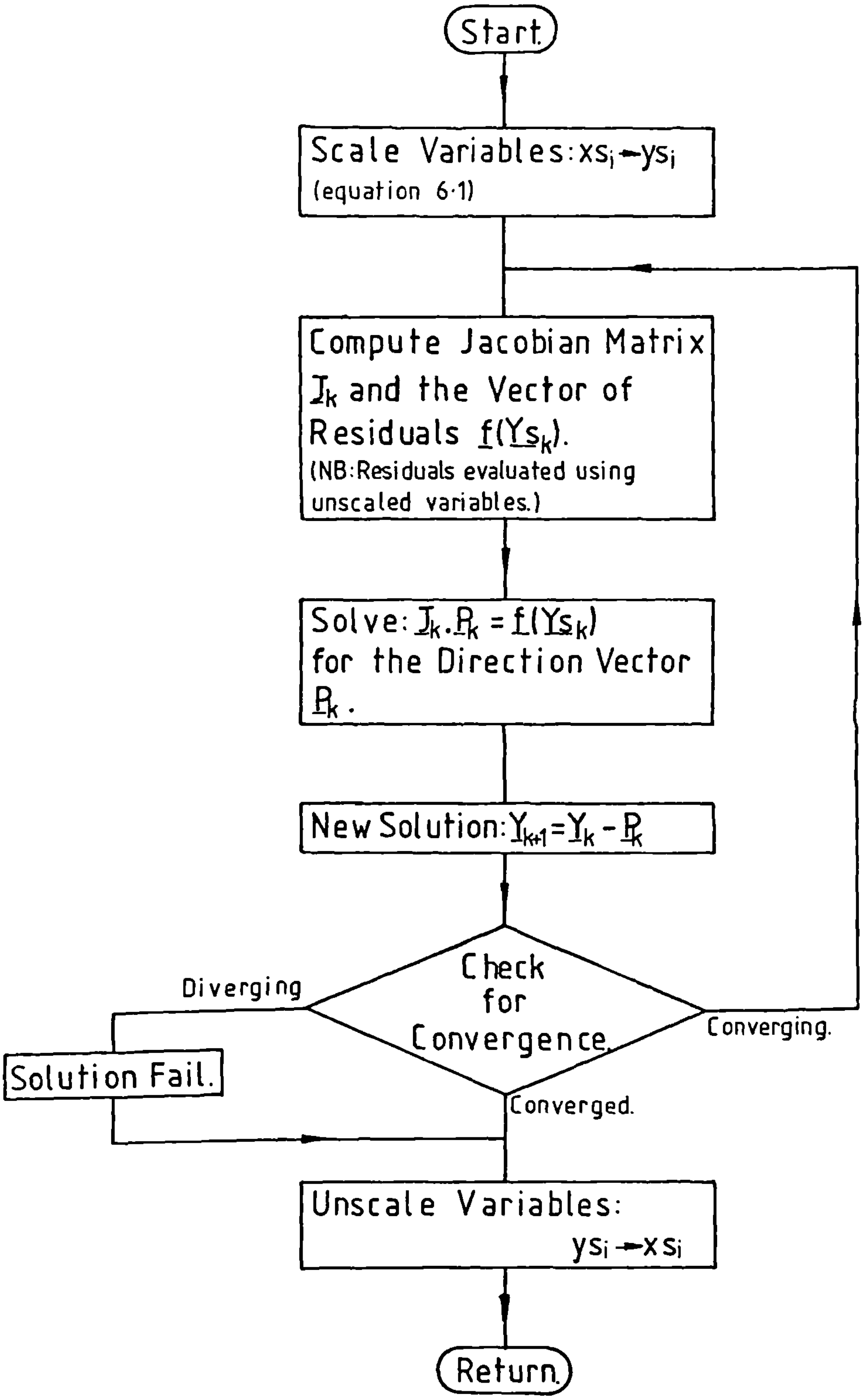


figure 6·1, Newton-Raphson Algorithm.

6.4 Component Performance Envelopes.

Extrapolation of component performance beyond the known performance characteristic is often precarious and meaningless. Where component models employ polynomial curvefits the behaviour of the curvefit outside the fitted data region can lead to spurious operating characteristics which in turn can mislead the solution algorithm and result in its failure. It is therefore important to ensure that the search is restrained to lie within the known region of component performance.

Development of bounds constrained solution algorithms has helped with this problem but still allows the search points to be generated beyond the performance envelopes. For example consider the fan performance curves illustrated in figure 6.2: part of the performance envelope of the volume-pressure characteristic can be defined by bounds on the fan blade angle, but complete definition requires two additional constraint functions for static pressure expressed as a function of volume flow rate.

In the context of developing an optimised design procedure, solution of the system equations by a constrained optimisation algorithm is important in recognising undersized components. The constraint functions which remain active on failure of such an algorithm are likely to be those of the undersized components and can therefore be used to identify these components and allow corrective action to be taken.

Rather than develop a non-linearly constrained optimisation algorithm which suits the characteristics of the simulation problem, Murray (1984) suggests that the constraints could be checked after completion of a search by an unconstrained algorithm. This however is likely to prove unreliable when the algorithm has failed to find a solution, as there would be no guarantee that the active constraints related to the undersized components. Further a constrained solution algorithm may prove more robust than an unconstrained one as the active constraints could be used during the search to 'direct' the solution away from the constraints towards a feasible solution.

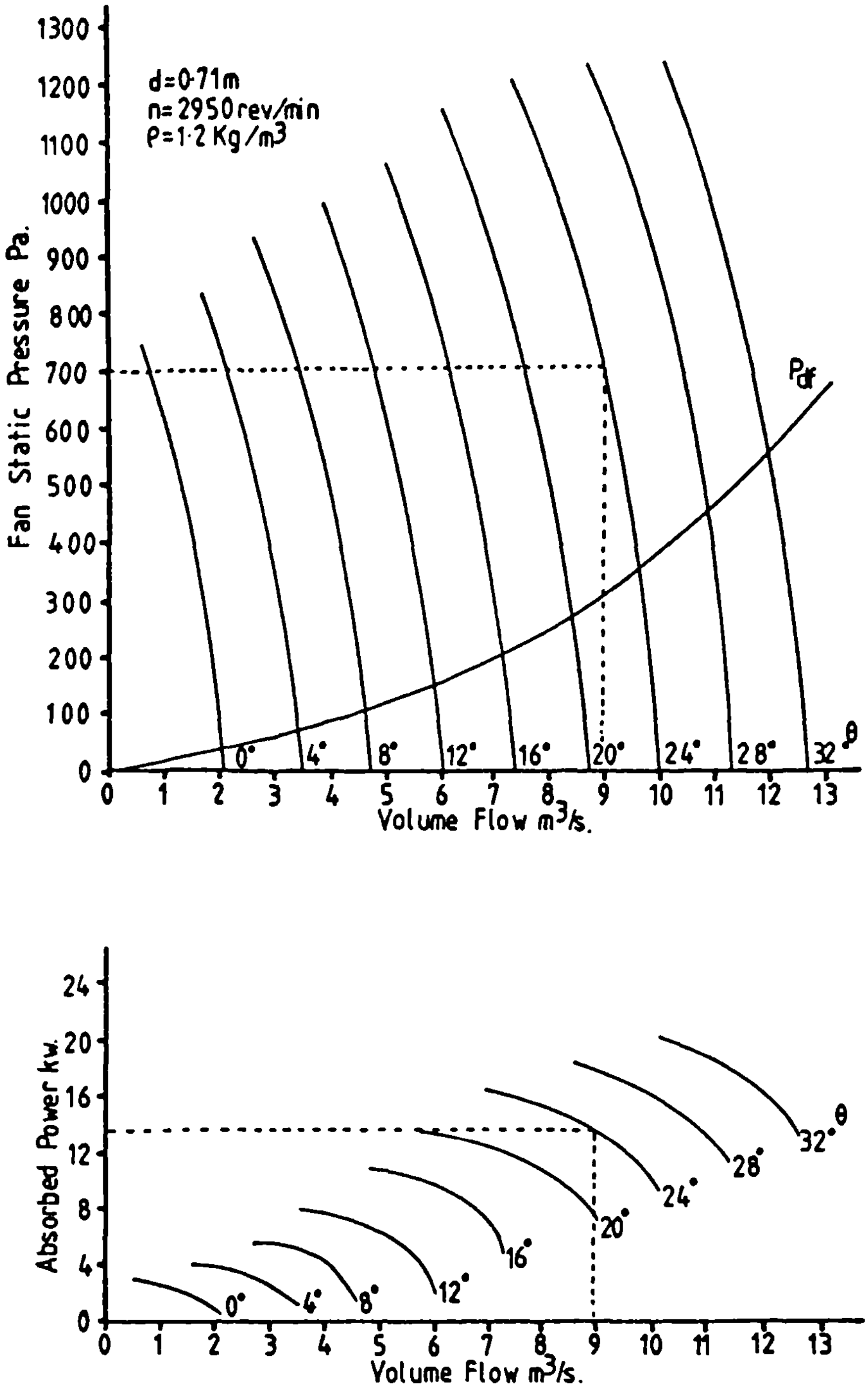


figure 6·2, Fan Performance Curves.

Therefore it is desirable that future research should aim at developing a non-linearly constrained optimisation algorithm for the solution of the system equations.

Chapter 7. FORMULATION OF OBJECTIVE AND CONSTRAINT FUNCTIONS.

The major elements of the HVAC system optimised design problem are described in chapter 3. Those parameters used in this research to formulate the objective and constraint functions have been chosen to reproduce the characteristics of the optimisation problem without over complication and loss of clarity. Once a solution procedure has been developed, future research can concentrate on improving the integrity of the function formulations. Possibly the most important parameter in assessing the validity of the objective and constraint function formulations, is the representation of varying climatic and zone loads.

7.1 The Variation in Load on the System.

HVAC systems must be capable of operating over a range of climatic and zone loads. In this research the variation in load over a period of time is represented by changes in the value of the exogenous variables. The SPATS system simulation procedure allows the definition of a load profile of values of exogenous variables for up to twenty-five time periods. The profile is stored on a data file and is recalled for use in the system simulation, the simulation repeated for each time period in the profile.

7.2 Formulation of the Energy Objective Functions.

Three types of energy term can be associated with any one component, direct, ancillary and extraneous. Direct energy terms by definition are expressed as net energy and ancillary and extraneous terms as useful energy. In this research both system net and primary energy models are formulated using the extraneous and ancillary terms expressed as useful energy and which is justified in that the characteristics of the objective functions are unchanged but the problem definition is less complex and therefore more transparent, (section 5.3).

Formulation of the system energy model is in two parts, selection of the energy terms to include in the model and for those terms included, definition of whether to add or subtract their value in the model. The convention adopted for the addition/subtraction of energy terms in the system model is:

Energy terms relating to energy **used** are **added.**
Energy terms relating to energy **saved** are **subtracted.**

To save confusion, the absolute value of each energy term is taken before its value is used in the system model. This is useful when dealing with components whose energy terms are considered as negative in a thermodynamic sense but are added in the system energy model. For example, in modelling a cooling coil the calculated coil duty is generally considered negative, yet if the duty is related to energy use it is added in the system model. Although this method is convenient, care is required to ensure that poorly controlled heat recovery devices do not change from recovering heat to using heat as this would not be recognised by the model. For instance, if a run-around coil is poorly controlled there may be certain conditions at which the coil ceases to be useful and starts to impose an extra load on the system. This change would not be recognised by the energy model as the absolute value of the energy term is used.

7.2.1 Primary Energy Modelling.

Formulation of the system primary energy model differs only from that of the net energy model in that the component energy terms are multiplied by the primary energy ratios before inclusion in the system model. The fuel types and corresponding primary energy ratios, used in this research are given in table 3.1.

7.2.2 Integration of Energy Consumption.

The formulation of an exogenous variable load profile provides a convenient method of integrating energy consumption over a period of time. The technique is commonly known as the 'bin' method in which each 'bin' in a series represents the load on the system for a specified interval of time and the energy consumption over the complete series estimated by integrating the calculated energy consumption from each time interval. Each bin of load data, in this research, is represented by a time period in the load profile of exogenous variables.

Selection of an appropriate time interval is dependent upon the availability of climatic and zone load data and the accuracy of the calculation required. The smaller the time interval the greater the accuracy but the longer the calculation time. For example, the boiler energy usage shown in figure 7.1 is best approximated by integrating bins (1) to (8) rather than employing the single bin (9). However eight bins will require eight calls to the simulation solution algorithm for every evaluation of the objective function, thus increasing the calculation time by a factor of eight.

The number of time intervals used in this research has been influenced by the computation time of the simulation solution algorithms. Of those available the fastest is the scaled variable version of the Newton-Raphson algorithm. A maximum number of time intervals for use with this algorithm would be in the order of 24. This would allow a load profile constructed from two typical days, one for the heating season and the other the cooling season. However, the simulation of systems of more than a few components requires the more robust GRG2 solution algorithm. This is much slower than the Newton-Raphson procedure which inhibits the number of time intervals used and therefore care has been taken to ensure that the systems used in the development of the optimised design solution algorithms, maintain their problem characteristics when their performance is simulated with a small number of time intervals.

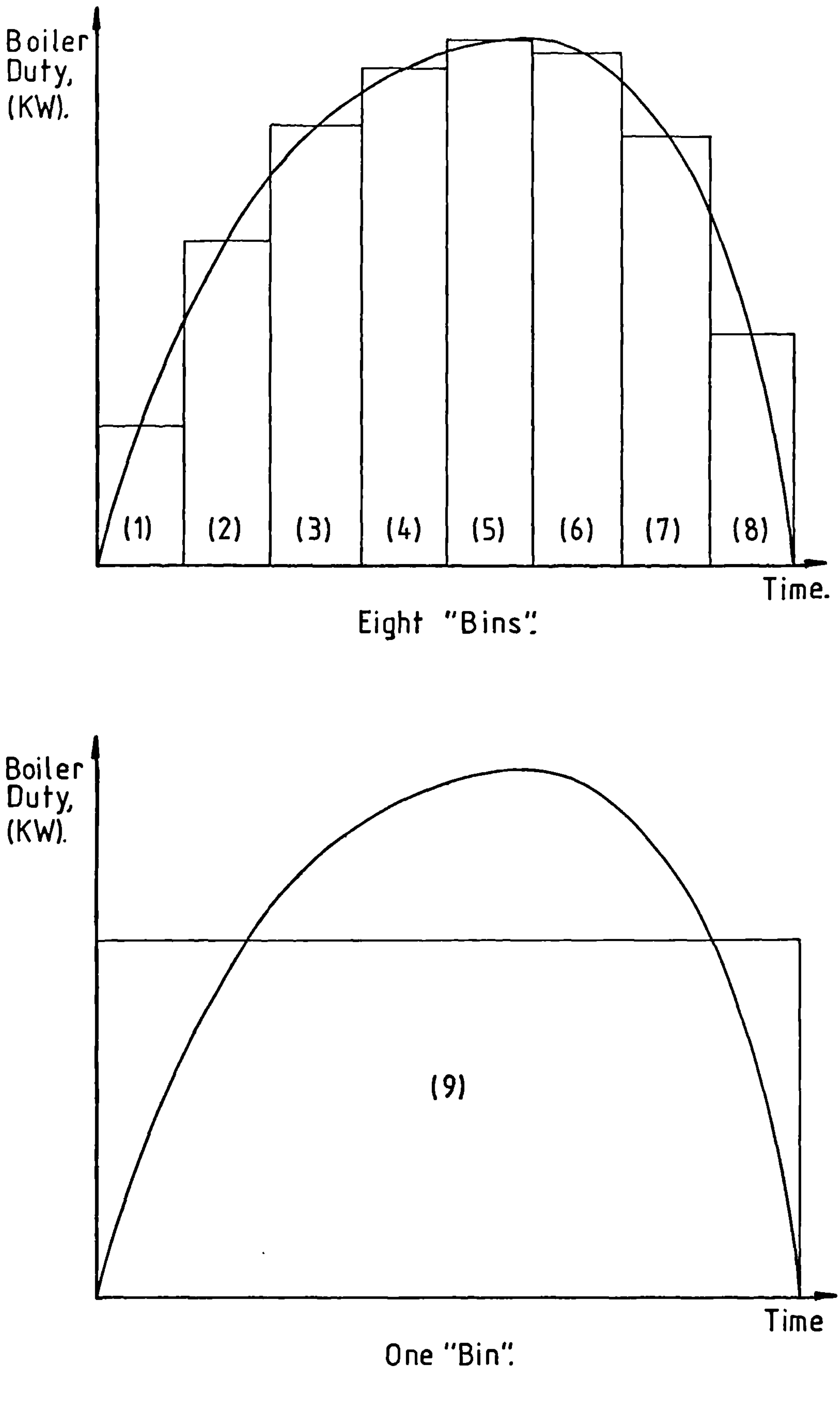


figure 7·1 , Example "Bin" Method.

7.3 Formulation of Economic Objective Functions.

The objective functions included under the title of 'economic' functions are the system annual operating cost, the first cost of the system and the true economic comparitors of net present value and discounted payback period. The sign convention adopted for use in the economic calculations is:

Costs incurred are added.
Capital saved (or revenue) is **subtracted.**

7.3.1 System Annual Operating Cost.

The most influential contributor to the characteristic behaviour of the operating cost with a change in component size is the system energy cost. Calculation of this requires an assessment of fuel tariffs which are often based on the energy consumption and peak demand of the system. However the inclusion of a complex tariff structure in the design procedure is beyond the scope of this research and therefore fuel prices have been restricted to a single value based on the gross calorific value of the fuel. Those fuel types included in this research are coal, fuel oil (35 second), gas and electricity (peak).

Formulation of the system energy cost model is based upon the system energy model, the sign conventions in the energy and cost models being compatible. For example, energy recovered is subtracted in the energy model as is the cost of the energy recovered in the economic models. Formulation of the energy cost model is similar to that of the primary energy model in that the value of each energy term is multiplied by the appropriate fuel price before being added or subtracted in the system model. Addition or subtraction of values is dictated by the sign associated with each energy term in the definition of the system energy model. The fuel type assigned to each energy term in the primary energy model is used to associate the correct fuel price with each energy term. The factors which affect the formulation of the net and primary energy objective functions, also apply to the formulation of the energy cost objective function since these are distinguished only by the multiplication of energy terms by the fuel prices.

The remaining factor included in the formulation of the system operating cost is the annual direct maintenance charge for the system. On-costs are not included in this formulation but could be calculated at 40% of the direct cost (Milbank 1971). The variation in system performance over a load profile complicates the calculation of the direct maintenance charge as this is often dependent upon the peak duty of the component. The procedure adopted for this calculation is to compare the maintenance charge at each time period with the highest maintenance charge encountered up to that time period and to retain the largest of the two values for comparison in the next time period.

Figure 7.1 illustrates that the selection of suitable time intervals can affect the accuracy of the maintenance charge calculations which are based upon the peak duty of a component. A load profile of eight bins includes the peak duty on the boiler at time period (5), however the approximation of the system performance by a single time period (9) does not include the peak duty expected of the boiler and hence the maintenance charge calculation is only approximate.

Labour, water and insurance costs are not easily determined and as they do not represent a major contribution to the characteristic behaviour of the operating cost, they have not been included in the formulation used in this research.

7.3.2 System Capital Cost.

The predominant element in influencing the characteristics of the system capital cost function is the price of the component, with delivery and installation costs increasing the total capital cost by a fixed percentage. The cost of additional building work caused by a change in the system design is more applicable to the comparison of different schemes than to the optimum sizing of components. Therefore in this research the system capital cost is formulated from the sum of the component prices.

7.3.3 System Net Present Value.

The net present value (NPV) of the system is the total value of the project over the life of the building expressed in terms of prices at the beginning of the project. NPV calculations include the capital and operating costs of the plant and therefore the accuracy of these functions affects that of the NPV calculations.

Component replacement and system operating costs are discounted to represent prices at the beginning of the project by use of the series and single value present worth factors:

Single value present worth factor (pwfsng):

$$pwfsng = \frac{1}{(1+i)^n} \qquad (7.1)$$

Uniform series present worth factor (pwfsrs):

$$pwfsrs = \frac{i*(1+i)^n - 1}{i*(1+i)^n} \qquad (7.2)$$

where 'i' is the rate of interest. Component operating cost is discounted by multiplying its value by the uniform series present worth factor in which 'n' is taken as the life of the building. Similarly, component replacement cost is discounted by multiplying its value by the single value present worth factor in which 'n' is the number of years from the beginning of the project.

The affect of inflation on the operating cost, component replacement cost and interest rate is not included in the NPV formulation used in this research. This approximation allows for a simpler NPV model as the replacement and operating costs are assumed to be uniform throughout the life of the project.

The NPV is formulated from the sum of the discounted system operating cost, capital and discounted replacement costs of each component. The number of times each component is replaced during the life of the building is calculated by comparing the component service life (table 3.3), with the estimated life of the building.

7.3.4 System Discounted Payback Period.

The discounted payback period can be defined as the time taken from the beginning of the project for the present worth of the system to become zero. The present worth of the system is the total value of the project to date, expressed in terms of prices at the beginning of the project. Discount payback period calculations are inferior to the net present value calculations in that no account is taken of the cash flow after the payback period. However, payback period is often used for the economic assessment of heat recovery systems. The parameters included in the formulation of the discount payback period calculations are the system operating cost, capital cost, interest rate and the single value present worth factor, each of which has the same influence on the discounted payback period as on the net present value calculation.

Discount payback period is formulated in this research by calculating the present worth for each year of the project until the present worth is less than zero. The payback period itself is estimated by a linear interpolation between the first negative and last positive present worth values, to determine the point at which the present worth is zero (table 7.1 and figure 7.2). The present worth in each year is calculated by subtracting the discount operating cost from the present worth of the previous year, the initial present worth taken as the capital cost of the system. Note that because of the energy/cost term sign conventions, the calculated operating cost for heat recovery systems will be negative, hence in practice the discount operating cost is added to the previous present worth. Discounted operating cost is calculated by multiplying the operating cost by the single value present worth factor.

Year.	Capital Cost, (Pounds.).	Operating Cost, (Pounds.).	Present Worth Factor. (Equation 7·1).	Discount Operating Cost, (Pounds.).	Present Worth, (Pounds.).
0	3000				3000
1		-1000	0·9091	- 909·1	2090·9
2		-1000	0·8264	- 826·4	1264·5
3		-1000	0·7513	- 751·3	513·2
4		-1000	0·6830	- 683·0	-169·8

Interpolate between years 3 and 4 for a zero Present Worth and the Payback Period:-

Payback Period = 3 + 513·2 / (513·2 + 169·8)
= 3·75 Years.

table 7·1, Discount Payback Period Example.

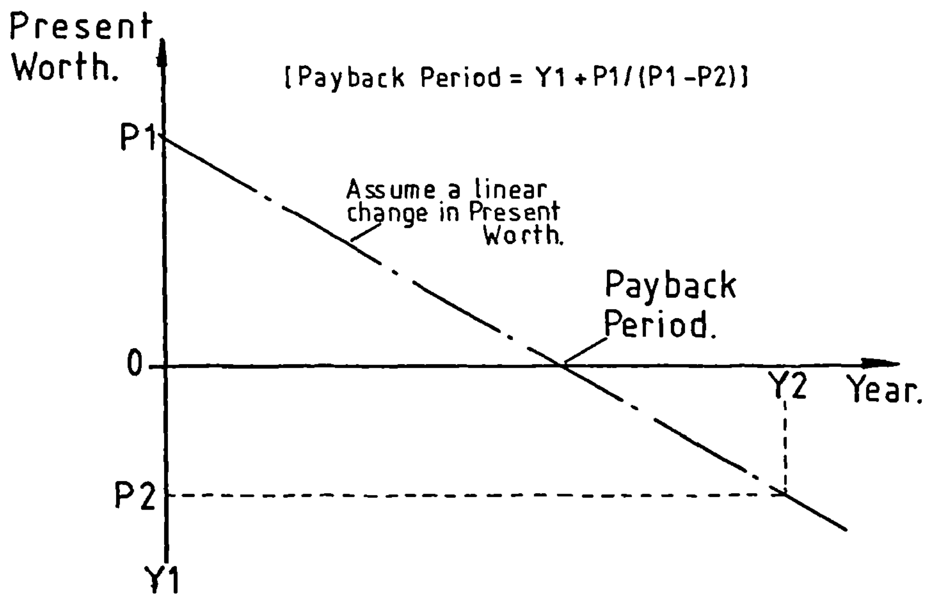


figure 7·2, Interpolation of Payback Period.

As the payback period can apply to specific components in the system ie: the heat recovery components, it is convenient to have an index of components to be included in the calculation. This allows the performance simulation of the complete system and evaluation of all the constraints without including every component in the formulation of the objective function.

7.4 Formulation of the Constraint Functions.

The simulation of system performance over several time periods affects the formulation of all constraint functions except those which are not associated with the fluid properties, ie: the configuration constraints. The change in system performance over a load profile gives a different value of constraint function for each time period in the load profile. Clearly there is a choice of formulating a constraint *function for each time period* or to use a single constraint value representative of the constraints behaviour over the range of load conditions. The latter approach has the advantage of reducing the number of active constraints and therefore calculation time, but it can be difficult to ensure the value chosen is representative of all load conditions (section 8.6.5).

The validity of any constraint value will, as for the objective function formulation depend on the degree to which the load profile represents the real load conditions. If the complete range of conditions that the system is expected to perform under are not included in the load profile, then when these conditions are ecountered in practice some of the constraints may be violated. For example, if the complete range of conditions imposed on a variable air volume (VAV) system are not included in the load profile, then when the selected system is installed, it may be found that when it is operating at its maximum volume flow rate the cooling coil face velocity is higher than the maximum value specified.

Selection of realistic load data is of even more importance in formulating an undersizing constraint function as many components have to cope with maximum and minimum loads. For instance, the fan of a VAV system must be capable of operating at both the maximum and minimum volume flow rates. Further, extreme load conditions perhaps not normally used in the selection of components can prove invaluable if included in a load profile as they improve the reliability of the undersized component constraint. For example, the selection of a cooling coil is normally based upon the peak cooling load, but if the extreme humidity conditions often encountered early in the morning are not included in the load profile, then the selected cooling coil may not be able to cope with the dehumidification load imposed under these extreme conditions.

The complexity of forming rigorous constraint functions representative of a range of operating conditions and the problems of forming a component undersizing constraint, have led to the simple rejection of infeasible points in the optimised design solution algorithms. However this has proved to be unreliable and future optimisation algorithms will require a more rigorous constraint formulation (chapter 8).

7.5 Component Undersizing as a Constraint Function.

If the undersizing of components is to be included as a mathematical constraint in future optimisation algorithms the severity of undersizing must have a numerical significance. The undersizing of components is indicated by failure of the simulation solution algorithm, hence the successful formulation of the constraint is dependent upon being able to interpret the simulation problem parameters on failure of the solution algorithm. In an attempt to identify possible numerical indicators the behaviour of the following parameters on failure of the simulation solution algorithm has been investigated (appendix C):

1. The sum of the component residual equations.
2. The largest value of the unsolved residual equations.
3. The number of unsolved residual equations.
4. The arcvariable values which are on their bounds.

Results suggest that these parameters could be used to formulate a constraint function for the undersizing of components and to identify the components actually undersized. However, experience has shown that the currently available simulation solution algorithms exhibit the same characteristics when failure of the solution is due to the instability of the algorithm as when failure occurs due to the undersizing of components. Therefore, before a mathematical constraint which represents component undersizing can be considered reliable, a more robust simulation solution algorithm must be developed.

7.5.1 Formulation of an Undersizing Constraint Function.

Research (appendix C) has shown that the largest value of all the unsolved residual equations *increases with an increase in severity* of the undersizing, whilst the number of unsolved equations is unaffected. Formulating an undersized components constraint function from the largest value of the unsolved residual equations could prove unreliable when more than one component in the system is undersized as the constraint function would only be related to the single component. A further complication is that there is no guarantee that the residual equations that remain unsolved are related to the components which are undersized, since the order of solving the equations is often dependent upon the scaling of the system variables and equations (Murray 1984).

A more reliable approach would be to employ the sum of the residual equation values as this is not related to a single component. The sum of the residual equation values was found to increase with the severity of undersizing, ie: the less likely a component is to meet an imposed load the greater the sum of the residuals on failure of the algorithm. This is as might be expected since the largest value of the unsolved residual equations increases with an increase in the severity of undersizing. Further, although not investigated, it is likely that the number of unsolved equations would be proportional to the number of undersized components thus an increase in the number of undersized components would be reflected by an increase in the value of the sum of the residuals.

The sum of the residuals could be formulated as an equality constraint of the form $c_i(\underline{X})=0$, since this would be zero in a system in which all components are correctly sized:

$$c_i(\underline{X}) = \sum_{j=1}^{n} f_j(\underline{Xs})$$

where $\underline{Xs}$ is a vector of the 'n' HVAC system arc-variables and each f_j is a non-linear constraint formed from the system residual equations. Before this is adopted as a constraint function, further research is required to establish its reliability when several components in the system are undersized.

7.5.2 Identification of the Undersized Components.

Research (appendix C) indicated that on failure of the simulation solution algorithm the signal value of the proportional controller controlling the undersized component was on its bounds, suggesting that the component was operating at its maximum capacity. This characteristic could be used to improve the speed of the optimised design algorithm by identifying the undersized components and thus the problem variables which influence the value of the undersized constraint function. Further research is required to ensure that this characteristic is reliable when several components in the system are undersized and that it is unaffected by the scaling of variables and the order of solving equations.

Chapter 8. SOLUTION OF THE OPTIMISED DESIGN PROBLEM.

Three phases can be identified in the structure of algorithms for the solution of constrained non-linear optimisation problems: validation of the initial estimate of the solution as a feasible point, minimisation of the objective function value and, for some problems, the additional validation of the solution. The research described in this thesis concentrates on the development of an algorithmic search method to minimise the objective function value. Suggestions are also given for the future development of algorithms which determine an initial feasible point and establish the validity of the solution.

The majority of numerical methods for solving constrained non-linear optimisation problems are iterative in character. Starting from an initial feasible estimate of the solution they proceed by generating a sequence of new estimates, each of which represents an improvement over the previous one. Of those techniques available, it is the direct search methods which lend themselves to the development of an algorithm for the solution of HVAC system optimised design problems.

The optimised design software has been constructed to enable the development and implementation of several algorithms. Development has been assisted by use of examples which highlight the salient characteristics of the problem and solution algorithms.

8.1 Why Direct Search Methods ?

Direct search methods are heuristic in character basing their search strategy on a comparison of objective function values. Gradient based methods are generally more efficient and faster to converge to a solution than direct methods, because unlike direct methods, gradient based techniques are mathematical in character basing their search strategy on the derivatives of the objective functions.

The most influential reason for adopting direct search methods to solve HVAC optimised design problems, is the behaviour of derivative techniques when used with discrete variables and discontinuous objective functions. As the partial derivatives of the objective functions are unavailable, the implementation of gradient based methods would require the calculation of derivatives by numerical techniques. These estimates are frequently plagued by numerical difficulties which affect the value of the estimates and convergence criteria. Numerical rounding errors can occur when the differencing interval is either too large or too small. Too large an interval can cross the minimum resulting in a change in sign of the gradient and subsequent failure of the algorithm. Too small an interval can result in a gradient value which is dictated by the numerical round-off procedure of the computer. These problems are compounded by the use of discrete variables as the available differencing interval is dictated by the difference in discrete values. Therefore, gradient based methods are best avoided when used with discrete variables as they invariably prove unstable (English).

A common approach to improving the stability of these methods is to form pseudo-continuous variables from the discrete variables. The resultant optimisation procedure is, optimise the problem using pseudo an true continuous variables, fix the pseudo-variables at the discrete values nearest to the solution and re-optimise to find the optimum value of the true continuous variables. The disadvantages of adopting this approach are that since the number of discrete variables is far in excess of the continuous variables, the formulation of pseudo-continuous variables would prove cumbersome, inefficient and require major restructuring of some component models. Further, optimising the problem twice increases the number of calls to the objective function which with its long calculation time would result in an excessive overall solution time.

The solution time for optimisation problems which have discrete variables decreases with the number of discrete values per variable, ie: the less the number of options the faster the solution. The number of discrete values for each variable is typically less than ten, this and the disadvantages of using derivative methods with discrete variables leads to the conclusion that direct search methods are more applicable than derivative methods to the solution of HVAC system optimised design problems. A final point in favour of direct search methods is that because they tend to repeat identical arithmetic operations with simple logic for convergence on the optimum, it is easier to gain a greater understanding of the characteristics of the optimisation problem than it would be through a more mathematical and complex approach.

8.2 Development of a Direct Search Algorithm.

Of those direct search algorithms available none have been developed specifically for use with discrete variables and non-linear constraints. The adaptation of existing algorithms to cater for discrete variables is likely to prove difficult and result in unreliable algorithms. The success of an algorithm formulated specifically for solving HVAC system optimised design problems relies upon its ability to include and use the characteristics of the problem. This has been considered in the development of a solution algorithm.

8.2.1 Selection of a Direct Search Technique.

The lack of a mathematical basis to direct search methods makes it difficult to assess the validity of the solution. Associated with this is the difficulty in establishing sound convergence criteria which can result in either a prolonged search or one which ends prematurely. A search method, which when used with discrete variables does not suffer from these problems, is the grid or exhaustive search method. An exhaustive search is one in which the objective function is evaluated at each point on an 'n' dimensional grid of discrete values and the solution taken as the point with the lowest objective function value. Continuous variables are included by assigning to them a set of discrete values, the grid size of which is reduced during the search until the required level of accuracy is obtained.

Obviously this technique is very inefficient as every combination of discrete values is explored in the search for the optimum. This represents the worst case and as such can be used to gauge the efficiency of other algorithms. Despite its inefficiency the exhaustive search is useful in establishing the validity of solutions obtained from more efficient algorithms and therefore has been included in the suite of solution algorithms developed in this research.

Akin to the exhaustive search in its simplicity is a search in which a series of trial points are generated at random. The point with the lowest objective function value is taken as the solution or can be used to define a reduced search region. Although random search techniques have proved effective for solving some optimisation problems, they are not suited to the solution of HVAC design problems as they usually require a high number of objective function calls in order to establish the validity of the solution.

The characteristic of the optimum to tend towards the bounds of the variables suggests that the speed of solution would benefit from an algorithm which once it had established the direction of the optimum converged upon it at an increasingly rapid rate. Several of the more common direct search methods establish a search direction by making trial probes along each axial direction. There are two alternative probe strategies in general use. The first probes along an axis in a direction which reduces the objective function value. When no further improvement is found, the search is switched to another axis and the process repeated. The search continues until no improvement is found in any direction, at which point either the solution has been found or the search has failed on a ridge (Dixon, 1972). Davis, Swann and Campey (reviewed by Dixon, 1972) improved upon the efficiency of this technique by accelerating the search towards the optimum after each of the 'n' axial probes and reduced the tendancy of the basic technique to fail on a ridge by rotating the axis to lie in the direction of the solution. However, search techniques which use this method of probing are likely to fail when the constraints are used to simply reject infeasible points. This is illustrated with reference to figure 8.1.

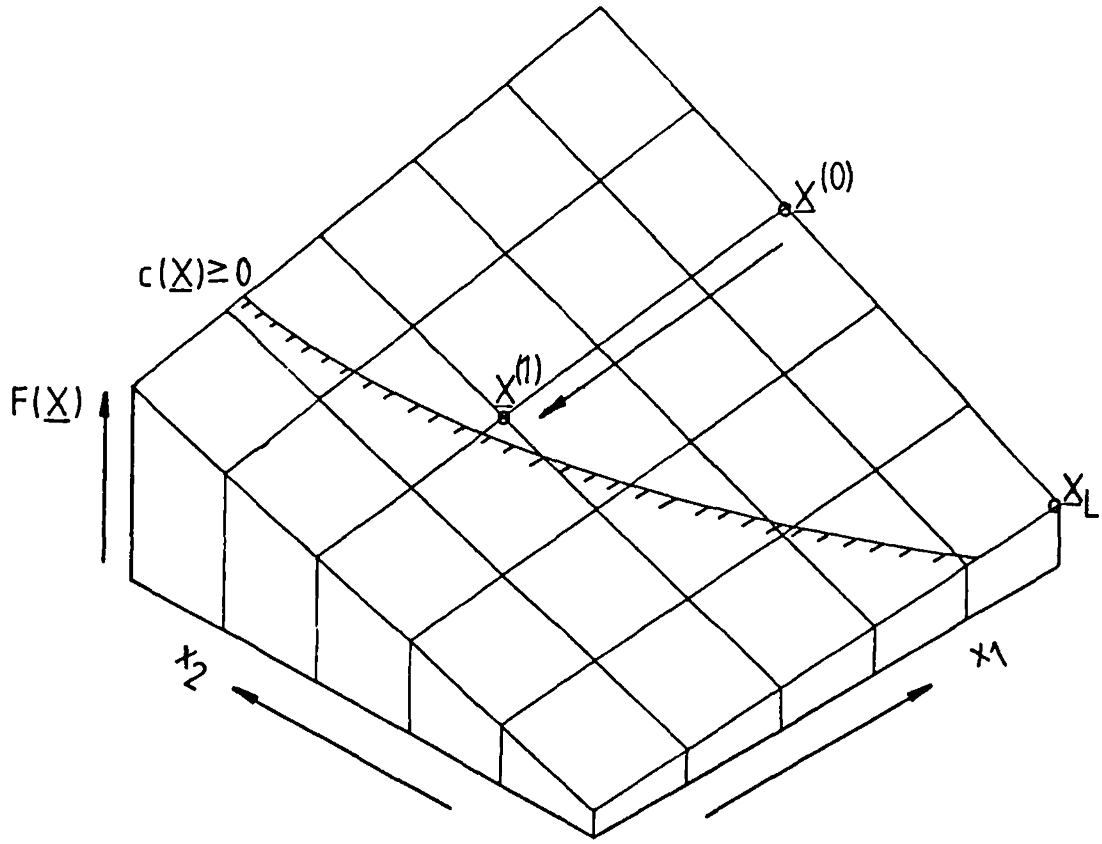


figure 8·1, Failure of an Axial Probe.

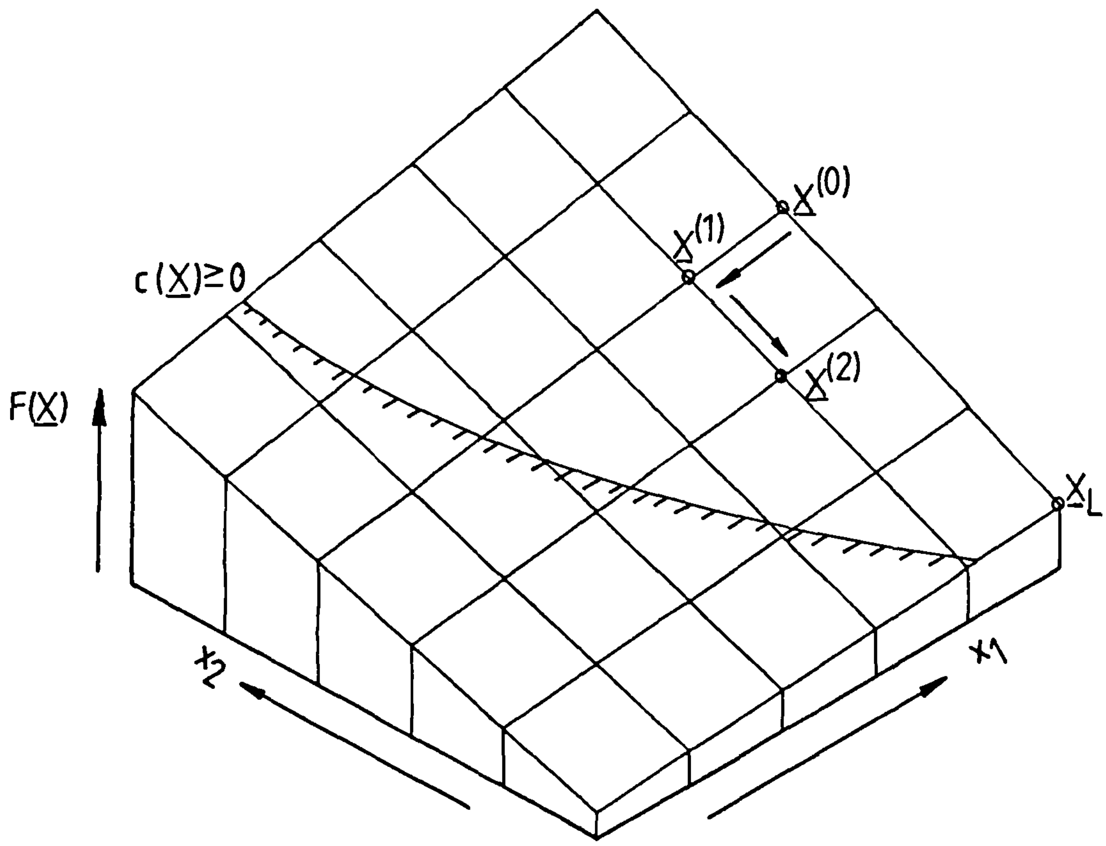


figure 8·2, Fixed Step Axial Probe.

Starting from the initial guess $\underline{X}^{(0)}$ trial probes are made along the axis x_1 until the constraint $c(\underline{X})\geq 0$ is ecountered: any further increase is rejected fixing the search position at point $\underline{X}^{(1)}$. Switching to axis x_2 produces no improvement in the point $\underline{X}^{(1)}$ as a move in x_2 towards the optimum $\underline{X}_L$ is rejected due to the proximity of the constraint, hence the search is terminated and has failed after probing only one axis.

A probing technique which is less likely to fail is one in which the length of step used in each probe is predefined. A shortened step length can allow more information to be obtained about the direction of the optimum as often more than one axis is probed before a constraint is encountered. The search illustrated in figure 8.2 probes along axis x_1 as far as position $\underline{X}^{(1)}$. Since this is remote from the constraint $c(\underline{X})\geq 0$, the axis x_2 can be probed enabling the search to progress. A search technique which employs this method of probing is that due to Hooke and Jeeves (1960), in which a set of 'n' axial probes resulting in an improved objective function value is followed by an accelerated 'pattern' move in the direction established by the exploratory probes. If this results in a further reduction in the objective function value, the accelerated point is retained and the direction of the optimum is checked by another set of exploratory probes and the process repeated. Failure to improve the objective function value after a set of exploratory moves or a pattern move results in an attempt to relocate the direction of the optimum by exploring around the last feasible point. Failure here is followed by a reduction in the probe step length and the repetition of the exploratory moves. This procedure continues until a new search direction is located or the step length falls below a predefined minimum. The caution of repeated exploratory moves lends itself to the solution of HVAC design problems as although the direction of the optimum is well defined, too rapid a progress can result in difficulties when solutions are rejected through the violation of constraints.

Simple rejection of infeasible points has proved to be an unreliable constraint handling technique when used with this search method, however as a more rigourous method of constraint formulation is developed, the Hooke and Jeeves pattern search should prove to be a useful solution algorithm. Its simplicity in repeating identical arithmetic operations not only makes it easy to implement but also allows a greater understanding of the behaviour of the constraint and objective functions than might be obtained through a more complex algorithm.

It is for these reasons that the Hooke and Jeeves pattern search has been used in this research as the main solution technique. The greater understanding of the problem gained through its use has led to suggestions for improvements in the solution algorithm and the selection of alternative solution methods to be implemented as part of future research.

8.2.2 Constrained Optimisation by Direct Search.

Simple bounds on the variables can be successfully incorporated with search methods which probe along the co-ordinate axis by resetting variables onto their bounds when they have been violated by a search move (Swann, 1978). This allows the search to progress along the bound and as subsequent exploratory probes are made normal to the bound, the search can leave it if the bound becomes inactive.

Search moves which violate non-linear constraints can be simply rejected by assigning the violated point a very large objective function value, thus ensuring that the search point is rejected when compared with other solution points. However, in practice this has been found to be of little use as the close proximity of the constraint often requires a major reduction in step length before the search can progress and normally results in premature termination of the search before a sufficiently small step length is found (Swann, 1978).

Swann (1978), reports on a number of proposals which have been made for extending the basic Hooke and Jeeves pattern search to deal with constraints by using derivatives to direct the search along the constraint towards the optimum. However, the difficulties which arise in forming the derivatives with discrete variables suggests such methods are impracticable here. Of the non-derivative constraint handling techniques, the penalty and barrier transformation methods have proved to be the most popular and are successful when used with direct search methods. As with all the more complicated constraint handling techniques, these require reliable formulation of the constraint functions. The difficulties in forming reliable component undersizing constraint functions leaves no alternative but to use these constraints to simply reject infeasible points. In this research all constraint functions have been used simplyto reject infeasible points. Inevitably this has proved unreliable, but its simplicity has allowed a clearer understanding of the problem and lead to subsequent suggestions for the future development of more robust techniques.

8.3 Development of an Exhaustive Search Algorithm.

An exhaustive search is one in which the objective function is evaluated at each point on an 'n' dimensional grid of discrete values and the solution taken as the point which lies inside the feasible region and has the lowest objective function value. Continuous variables are included by defining a set of discrete values with a suitably small interval between values. Exhaustive search techniques cannot fail to find the correct solution when used with discrete variables and are unaffected by the simplicity of the constraint formulation as every point on the grid is evaluated and those which violate constraints rejected. The relatively small number of continuous variables in the HVAC system design problem does not affect the reliability of the exhaustive search method, provided that the interval specified between the discrete values assigned to the continuous variables, is small enough not to cross the optimum.

Each point on the 'n' dimensional grid is explored by sequentially varying the value of each variable. It is normal to vary the value of the first variable of the set most rapidly and the last variable least rapidly. For example, in the three dimensional grid illustrated in figure 8.3, variables x_1 and x_3 have two discrete values of 0.5 and 0.6 and variable x_2 three values of 0.5, 0.6 and 0.7. The search begins from the lowest value of each variable, $\underline{X}$=(0.5,0.5,0.5) and continues by evaluating each point along the first co-ordinate direction x_1, until its maximum value is reached, $\underline{X}$=(0.6,0.5,0.5). This is followed by incrementing the next variable in the set, x_2, by one step and repeating the search along the first co-ordinate direction. When the second variable has reached its maximum value the third variable is incremented and the previous combination of the first two variables repeated. The process of incrementing and repeating previous combinations continues until all the variables are at their maximum values. The complete sequence is given in table 8.1 and is illustrated in figure 8.3 with position (1) marking the beginning of the search and position (12) its completion.

If the exhaustive search is to be of use in solving optimised design problems it must be flexible enough to accept any number of variables each with a different number of discrete values. Although this has proved difficult an algorithm has been developed which uses two 'markers' and two index arrays in its formulation. The first marker, i_r indexes the variable incremented as part of a repeated combination and the second marker, i_f indexes the variable incremented for the first time. For instance, in table 8.1 at combination (8), i_r=1 as the first variable, x_1 is currently being incremented for the third time and i_f=2 since x_2 is the variable being incremented for the first time. Two arrays represent the current and final search positions. Array P represents the search position by indexing the active values for each variable. For example, at combination (5) in table 8.1 the first discrete value is active for variables x_1 and x_3 hence P(1) and P(3)=1 and the third discrete value is active for variable x_2, giving P(2)=3. Similarly the final search position is represented in the array M by indexing the maximum number of discrete values for each variable. A generalised flow chart of the algorithm is illustrated in figure 8.4.

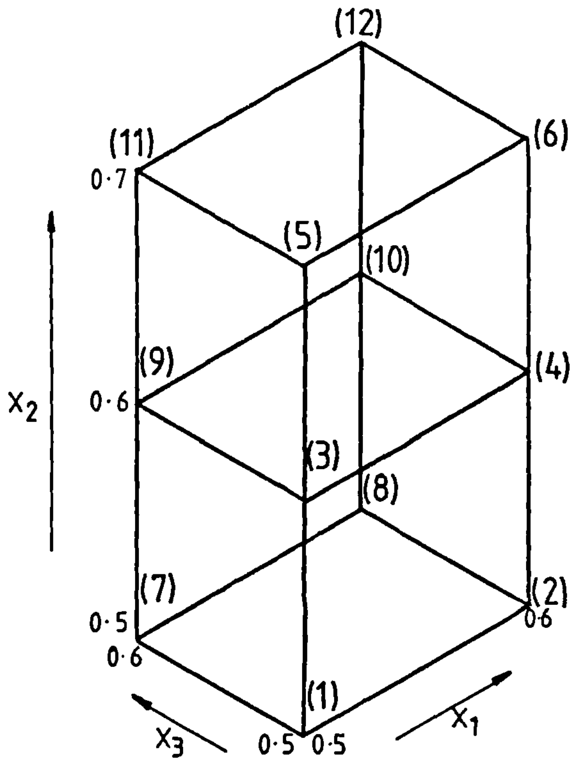


figure 8·3, Exhaust Search Grid.

Combi-nation.	Variable Value. X_1	X_2	X_3	Value Index. P(1)	P(2)	P(3)	Search Marker i_r	i_f
(1)	0·5	0·5	0·5	1	1	1	1	1
(2)	0·6	0·5	0·5	2	1	1	1	1
(3)	0·5	0·6	0·5	1	2	1	1	2
(4)	0·6	0·6	0·5	2	2	1	1	2
(5)	0·5	0·7	0·5	1	3	1	2	2
(6)	0·6	0·7	0·5	2	3	1	1	2
(7)	0·5	0·5	0·6	1	1	2	1	3
(8)	0·6	0·5	0·6	2	1	2	1	3
(9)	0·5	0·6	0·6	1	2	2	2	3
(10)	0·6	0·6	0·6	2	2	2	1	3
(11)	0·5	0·7	0·6	1	3	2	2	3
(12)	0·6	0·7	0·6	2	3	2	1	3

table 8·1, Exhaustive Search Example.

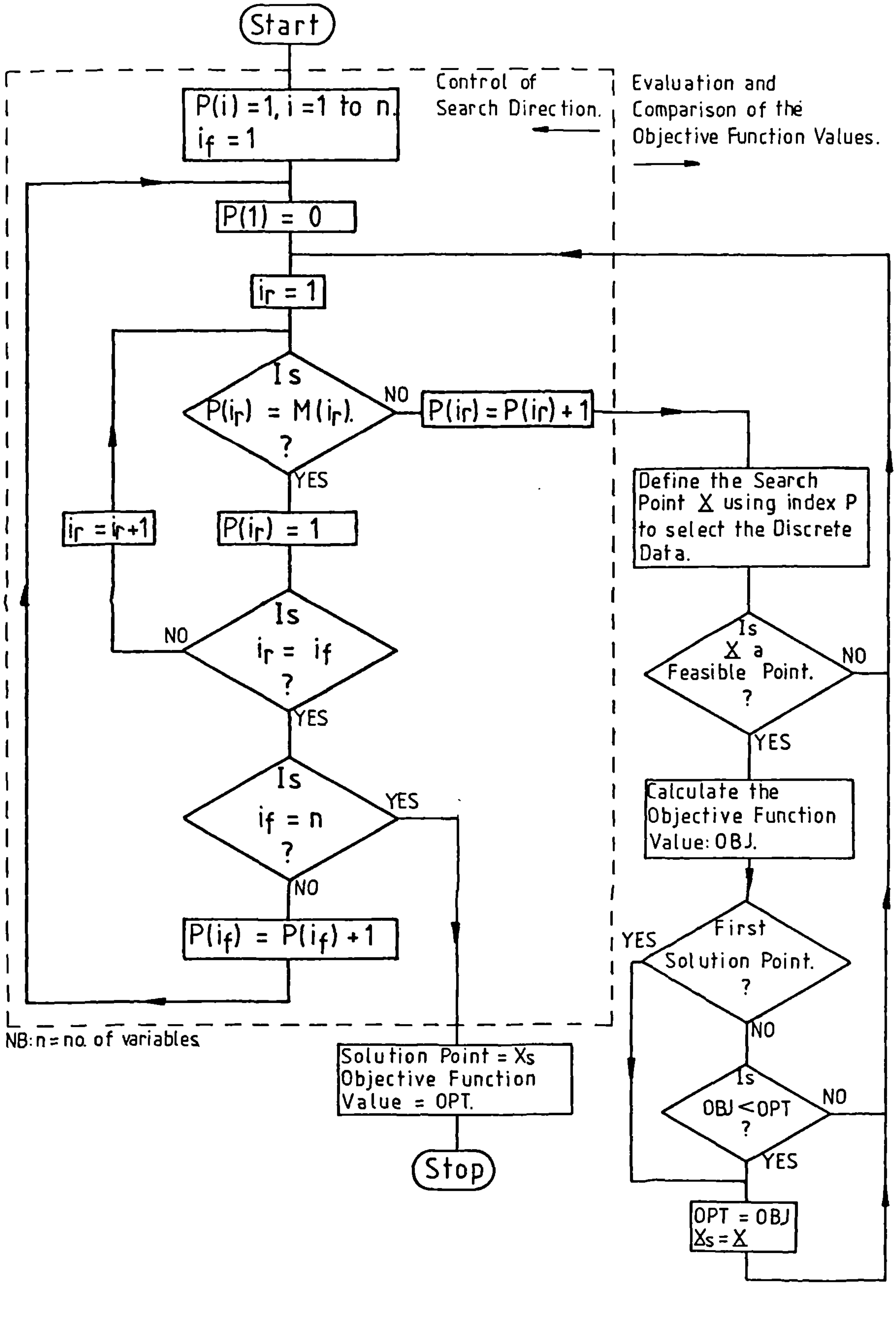


figure 8·4, Generalised Exhaustive Search Algorithm.

8.3.1 Use of the Exhaustive Search Algorithm and its Limitations.

It is inappropriate in this research to gauge the efficiency of optimised design search algorithms by the time taken to find a solution since the greatest influence on this is the efficiency of the performance simulation solution algorithm. A better measure of efficiency is the number of times the objective function is evaluated during the search. Because the exhaustive search algorithm evaluates the objective function at every combination of discrete values, the number of functions calls used by it is a measure of the complexity of the design problem and as such can be used to assess the variation in performance of other algorithms in solving different design problems.

The number of solution point is equal to the product of the number of discrete values assigned to each variable. This renders the exhaustive search method unusable for all except the simplest of problems. For example, in the design of a heating coil there may be 5 choices of coil depth and 10 choices of width and of height, giving 5 x 10 x 10 = 500 combinations. Changing the problem to the design of a run-around coil system with each coil having the same choice of depth, width and height increases the number of solution points from 500 to 500 x 500 = 250,000.

The high number of discrete values required to represent continuous variables greatly reduces the efficiency of this algorithm. A technique which helps reduce the number of times the objective function is evaluated, is to define a low number of discrete values for the continuous variables, ie: a coarse grid size and use the solution from this to define a reduced search region with a finer grid size. The sequence of repeating the search and reducing the search region is repeated until the difference in discrete values falls below a predefined minimum. This is the most common technique for dealing with continuous variables in an exhaustive search and is reliable provided the objective function is unimodal.

The exhaustive search has proved to be of use in checking the solution obtained from other algorithms. For most problems the number of solution points prohibits the use of an exhaustive search over the complete variable space and therefore solutions can be checked by either exhaustively searching the region immediately around the solution or by identifying the variables which appear to be incorrect at the solution, fixing all other variables at their solution values and exhaustively searching the suspect variables. These techniques have prove successful in identifying the incorrect solutions obtained from other algorithms.

8.4 Development of a Pattern Search Algorithm.

The development of a pattern search algorithm has been based upon the technique described by Hooke and Jeeves (1960). This procedure is characterised by two operations, exploratory moves and pattern moves. Exploratory moves attempt to locate the direction of the optimum by examining the local behaviour of the objective function. Pattern moves utilize this information and make an accelerated step towards the optimum. Both types of move are made relative to a set of co-ordinates, $(x_1, x_2, \ldots, x_n)$, termed base points. Exploratory moves are made relative to a temporary base point, $\underline{T}$, whilst pattern moves are made relative to a base point representing the current solution, $\underline{S}$.

Exploratory moves probe along each axial direction in turn. A co-ordinate is increased by a fixed step, k_i, and the value of the objective function compared with that at the temporary base. If the function value is lower, the co-ordinate is retained to form a new temporary base. Where the increased co-ordinate produces a higher objective function value, the original co-ordinate is reduced by the same step length and the comparison repeated. Failure to improve the objective function value leaves the temporary base unchanged.

When each co-ordinate direction has been explored, the pattern search compares the function values at the temporary and solution base point. If the temporary base point has the lower value an accelerated pattern move is made from the solution base towards and beyond the temporary base. This is in two stages, firstly a new temporary base $\underline{T}^{(j+1)}$ is created at a distance equal to the increment between the two base points and in the same direction as the existing temporary base $\underline{T}^{(j)}$ from the solution base:

$$\underline{T}^{(j+1)} = 2.\underline{T} - \underline{S}^{(j)}) \qquad (8.1)$$

The second stage of a pattern move is to set the solution point to the original temporary base:

$$\underline{S} = \underline{T}^{(j)} \qquad (8.2)$$

The search begins from a given feasible point which is taken as the initial temporary and solution base. Exploratory moves are made relative to this and the point arrived at used to make a pattern move. The procedure continues to alternate between exploratory and pattern moves until the point reached from a set of exploratory moves has a higher or equal objective function value than the current solution base. When this occurs the temporary base is set to the solution base and the search restarted with a set of exploratory moves. Failure here to locate a new search direction results in a reduction in the probe step length and a repeat of the exploratory moves. The search continues in this manner until a new search direction is found or the step length falls below a predefined minimum and resulting in convergence of the solution.

This unconstrained version of the algorithm is illustrated in figures 8.5 and 8.6. The notation is for single base points $\underline{T}$ and $\underline{S}$ which are 'overwritten' in each pattern move. This change the format of equations (8.1) and (8.2) to that of equations (8.3) and (8.4) respectively:

$$T = 2. \underline{T} - \underline{S} \qquad (8.3)$$

$$S = (\underline{T} - S) / 2 + \underline{S} \qquad (8.4)$$

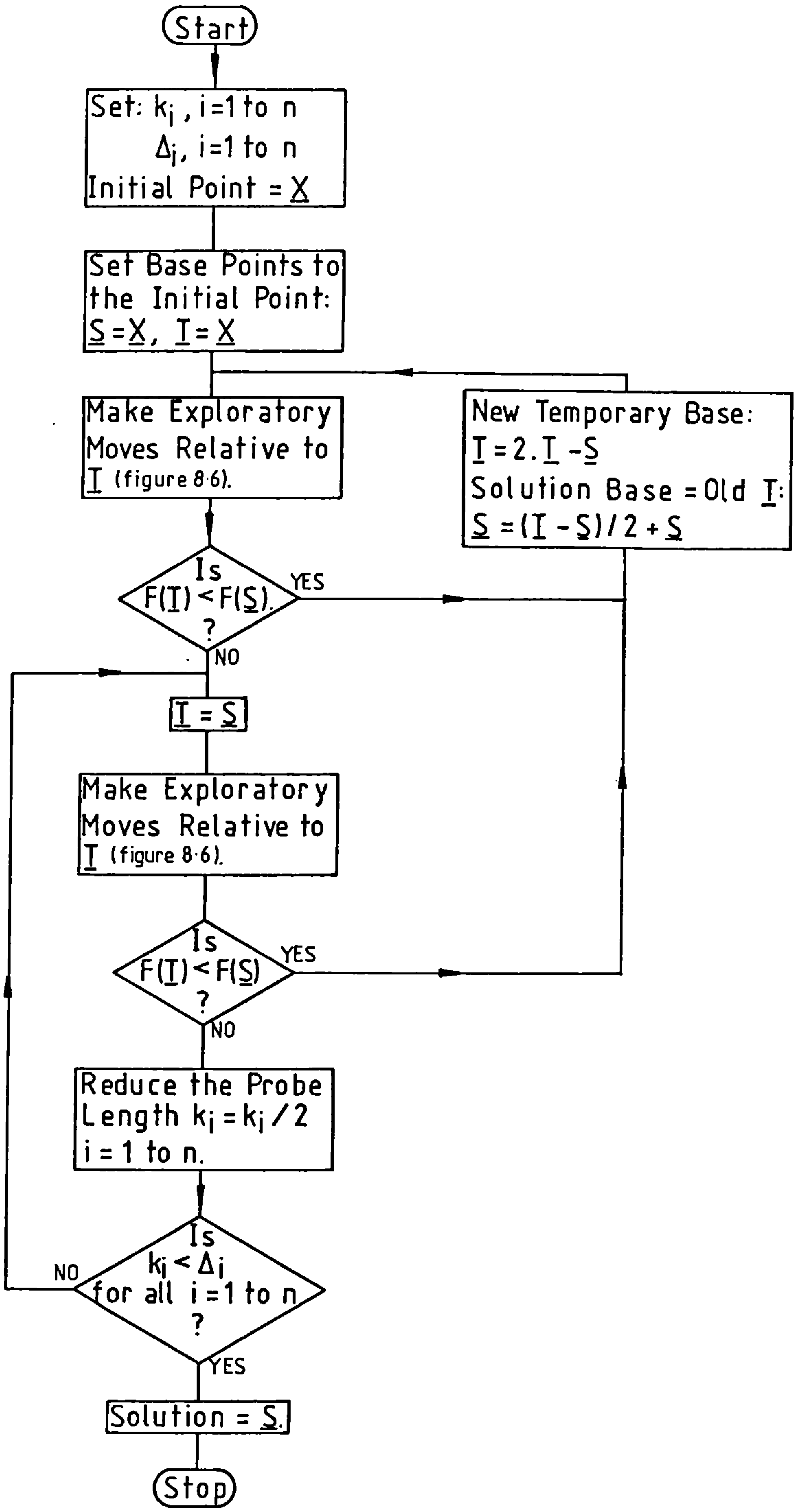


figure 8·5, Unconstrained Pattern Search: Pattern Moves.

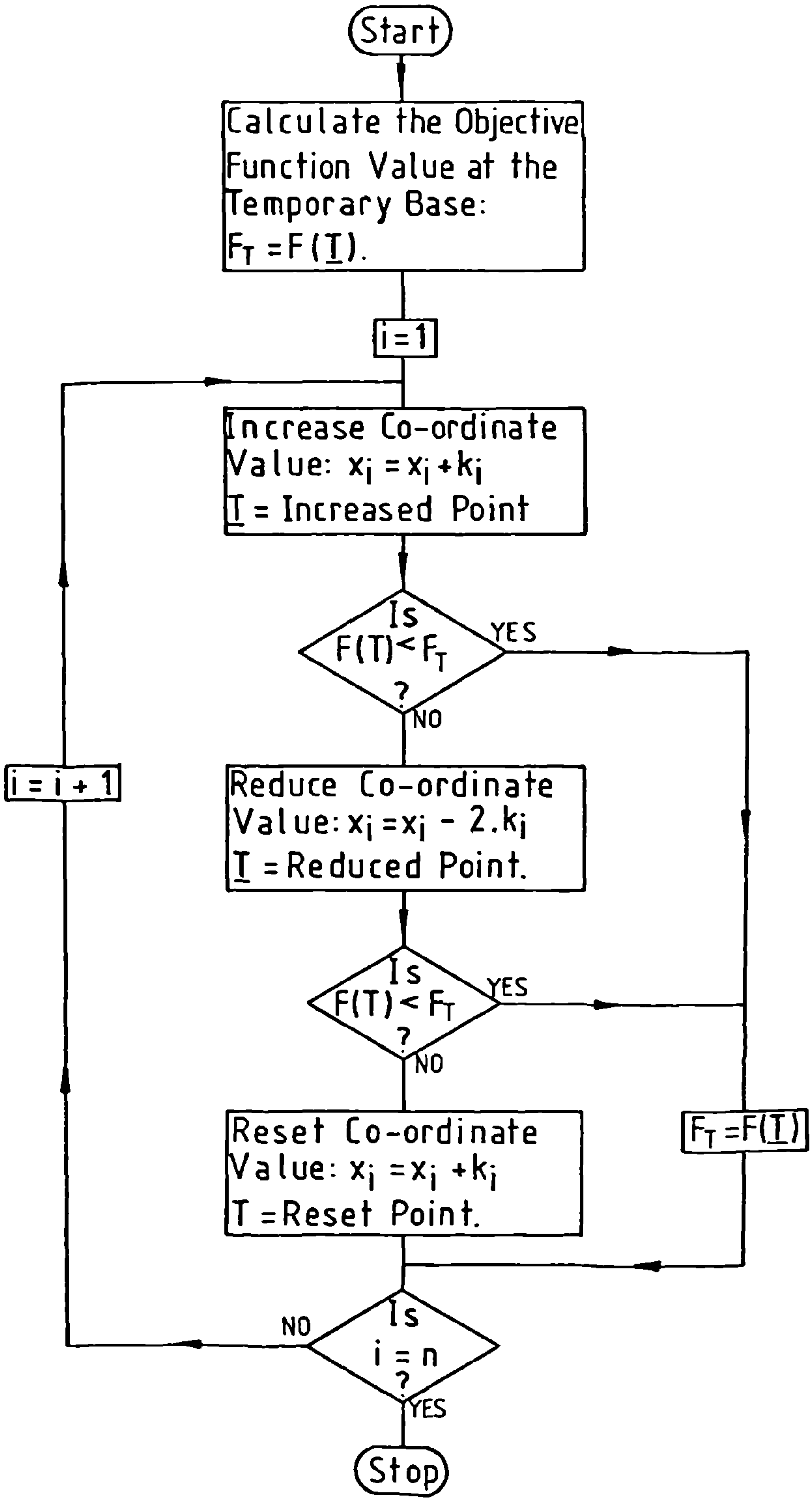


figure 8·6, Unconstrained Pattern Search: Exploratory Moves.

8.4.1 The Constrained Pattern Search.

The constrained pattern search initially implemented, simply rejects search moves which violate constraint functions. Variables whose bounds are violated during the search are reset onto their nearest bound. It is characteristic of the problem for the solution to lie on the bounds of the variables and constraint functions. As the solution nears the optimum the active bounds and constraint functions cause a repetition of exploratory and pattern moves rendering the basic pattern search very inefficient. This occurs in several ways:

If a set of exploratory moves results in a position with the active variables on their bounds, further progress of the search beyond the bounds will be prohibited. Therefore a subsequent pattern move is unable to locate a new temporary base which results in a common temporary and solution base. As this is not recognised by the algorithm, failure of the ensuing exploratory moves to locate a new search direction is followed by a search move which attempts to locate a new search direction by repeating the exploratory moves about the same point.

As the solution converges certain variables will reach their optimum values, which are often on bounds or constraints, before other variables. Subsequent exploratory moves reduce the efficiency of the algorithm as they produce no change in the value of these variables.

Often, a pattern move produces a temporary base which is in the infeasible region. If the constraint violation is caused by the value of a single variable the search can perform several unnecessary exploratory probes of other variables before the offending variable is identified and the search moved back into the feasible region. The worst case occurs when the temporary base is located so far into the infeasible region that it remains there throughout the whole of the exploratory move.

Having identified the inefficiency of the basic algorithm, an improved algorithm has been developed which incorporates the following refinements:

1. If the temporary base cannot be moved to a new location during a pattern move, because the variables are on their bounds, the search is advanced to that part of a pattern move which tries to locate a new search direction by setting the temporary base equal to the solution base. This eliminates the characteristic of the basic search algorithm to repeat this procedure.

2. If during a set of exploratory moves a probe in either direction produces no change in the value of a variable, ie: the variable appears to be at its optimum value, the speed of convergence can be improved by reducing the probe step size as part of the exploratory move. As the step length of discrete variables is selected to give an increment of one discrete value, discrete variables which appear to be at their optimum value are fixed at that value by assigning them a zero step length. Continuous variable step sizes can be halved, which although it does not fix the variable allows the search to probe closer to the supposed optimum.

   Fixing variables permanently is inadvisable because the search can change direction or move away from a constraint allowing a variable to change value. Therefore discrete variables remain fixed only so long as the search is progressing towards the solution and are released if the search is trying to relocate the direction of the optimum by exploring around the last feasible solution. Likewise continuous variable step lengths are only reduced in exploratory moves when the search is progressing towards the optimum.

3. If a pattern move produces a temporary base which lies in the infeasible region the point is rejected and the search continued with the temporary base set equal to the solution base. This reduces the tendency of the search to flounder in the infeasible region.

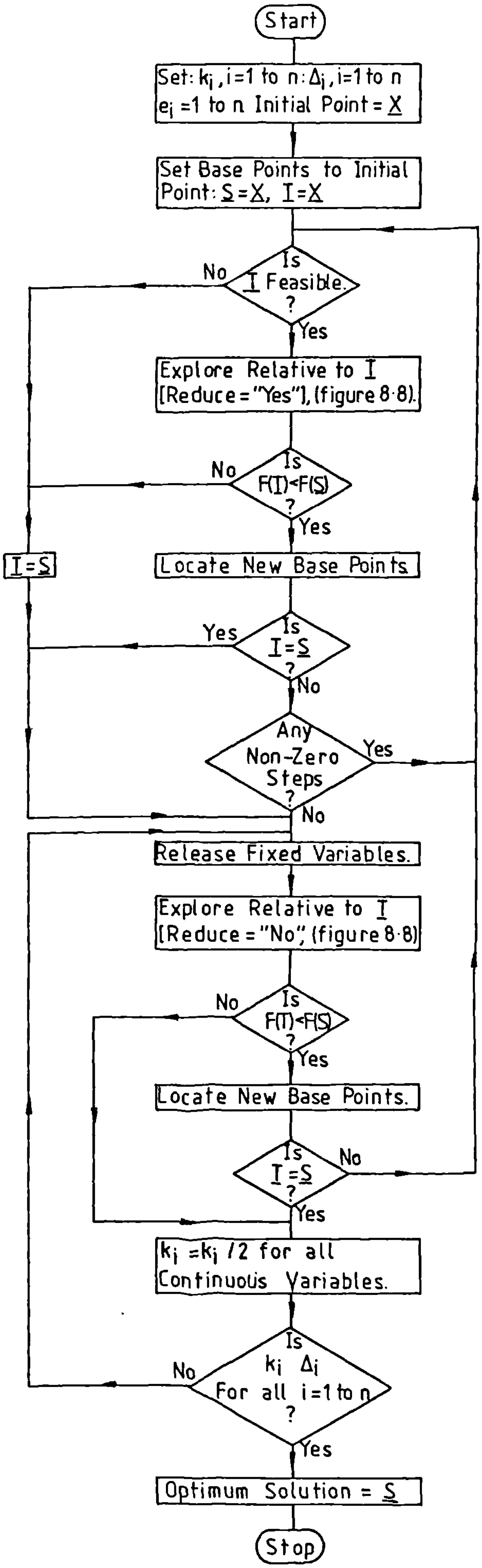
4. As the direction of the optimum is normally well defined the search can benefit if the intial exploratory probe is made towards the solution. The basic search adds an increment to the variable, compares the function values and then subtracts the increment if no improvement is found. If the direction of the optimum is thought to be in that of a negative increment, time could be saved by probing that direction first as this would eliminate the positive increment. Therefore a direction vector has been included to define the first probe direction. Initially this is in the positive direction and is subsequently dictated by the progress of the search.

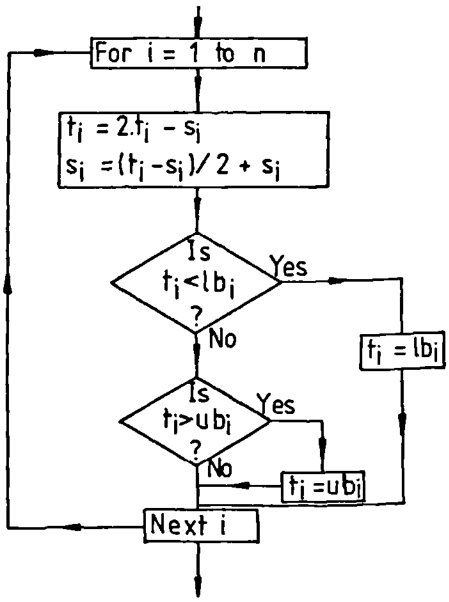
These improvements are illustrated in the flow charts of figures 8.7 and 8.8, in which, e, is the direction vector and t_i and s_i are the individual variable values at the temporary and solution bases.

8.4.2 Selection of an Initial Guess and Probe Step Length.

The solution time of the Hooke and Jeeves Pattern Search is influenced by the choice of the exploratory probe step lengths, k_i. the accuracy required, Δ_i and the closeness of the initial guess to the solution. Of these the most influential in the application of the algorithm to HVAC design is the selection of the initial guess.

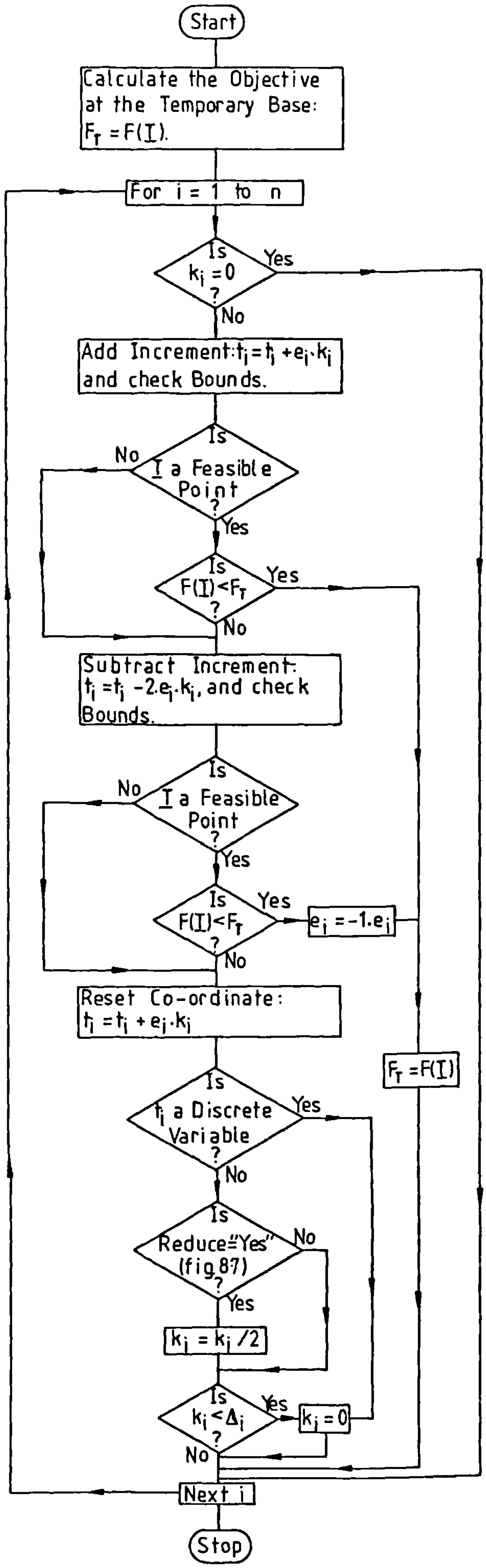
Since the solution of HVAC problems tend towards the bounds of the variables considerable time can be saved by specifying an initial guess which is close to the appropriate variable bounds. The problem that arises in using the current algorithm is that the initial guess must lie in the feasible region and a guess on the variable bounds is likely to lie outside of this. For the purposes of this research an initial guess has been chosen which is in the middle of the range of variable values. This has two advantages, firstly the initial guess is remote from the constraints allowing the search to find the direction of the optimum before they are encountered and secondly a mid-range guess should give an indication of the average performance of the algorithm, assuming that this lies between the best and worst guesses.

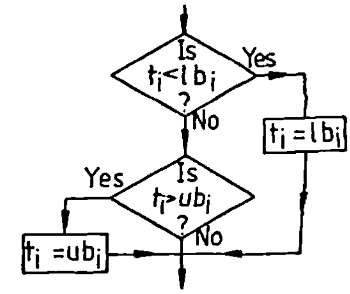




Location of New Base Points, (with Variable Bounds).

figure 8·7, Constrained Pattern Search: Pattern Moves.





Check on Bounds Violation.

figure 8·8, Constrained Pattern Search: Exploratory Moves.

It might be assumed that as the direction of the optimum is well defined, choosing a large probe step length would lead to a significantly faster solution time. Yet, it is a characteristic of the Hooke and Jeeves pattern search that a pattern move advances the temporary base by twice the distance advanced in the previous pattern move. This acceleration characteristic, together with the typically small number of values assigned to discrete variables, suggests that the solution time is little affected by a small initial step length. Further, a small initial step length can be an advantage in that the search is less likely to encounter the constraints before it has had a chance to locate the direction of the optimum. The step lengths chosen in this research are equal to an increment of one value for discrete variables and initially, a tenth of the variable range for continuous variables. This allows the improved search algorithm to extract the discrete variables from the active set when they appear to be at their optimum and when used with a mid-range initial guess, generally allows the search to locate the direction of the optimum before the constraints are encountered.

The minimum step length Δ_i, for discrete variables is dictated by the increment between discrete values. The minimum step length for continuous variables has been chosen to be a hundredth of the range of possible values. As the initial step length is a tenth of the range of values, each step length will be bisected four times before the search stops. The number of bisections could be reduced, improving the solution time, if in future algorithms a different minimum step length is defined to meet the required accuracy for each variable. For example, suppose a chiller water flow temperature is defined to be in the range $5^{\circ} - 12^{\circ}$ C then the initial step length would be 0.7° C and would have a final value of less than 0.07° C before the search stops. Obviously this is far beyond the accuracy required and a more realistic minimum step length might be 0.5° C, requiring only one bisection of the initial step length.

8.4.3 The Efficiency and Limitations of the Pattern Search Algorithm.

The efficiency and limitations of the modified pattern search have been assessed by applying the algorithm to a variety of problems. Of these the most informative has been the optimised design of a swimming pool heat recovery system in which two schemes were considered, a run-around coil system and a package chiller heat recovery system. The run-around coil system is comprised of an uncontrolled run-around coil which recovers waste heat from the exhaust air of the swimming pool hall and transfers it to the colder fresh air intake. Any additional heat requirement is supplied via a heating coil which is proportionally controlled by the action of a three port diverting valve. Conversely in the package chiller system, the total heat requirement is supplied solely by the package chiller which recovers waste heat from the exhaust air via a cooling coil connected to the evaporator. Heat is supplied from the condenser via a heating coil which is proportionally controlled by the action of a three port diverting valve. Operation of the chiller is controlled by varying the speed of the centrifugal compressor proportionally to the condenser water flow temperature. The problem definition includes discrete and continuous variables and a range of objective and non-linear constraint functions. A more detailed description of the example systems and their problem definitions is given in chapter 10.

The modified pattern search is robust, fast to find a solution and is suited to the characteristics of the objective functions associated with HVAC systems design problems. An important feature of the search method is that it is unaffected by spurious constraint functions introduced by the instability of the simulation solution algorithm. It is difficult to ensure the stability of the GRG2 simulation solution algorithm for all possible solution points, consequently spurious failure of the algorithm introduces false component undersizing constraints into the solution process. Providing these are sufficiently sparse and do not occur at the solution, the pattern search can 'bypass' the spurious points and eventually find the optimum solution. This suggests that the ability of the pattern search algorithm to find an optimum solution is unaffected by the occurence of sparse non-linear contraints.

Since the exhaustive search algorithm evaluates the object function for every combination of discrete data it is the least efficient search method and as such might be used to assess the efficiency of other algorithms. Yet even for the example problems which represent a low level of system complexity, it is impracticable to solve them using the exhaustive search as the number of objective function calls which would be required by the algorithm would lead to an excessive computation time. This and the large differential in the number of objective function evaluations required by the exhaustive and pattern searches implies that it is meaningless to quantitatively assess the efficiency of the pattern search based on the performance of the exhaustive search. However, as the exhaustive search evaluates every solution point, the number of function calls used by it is indicative of the complexity of the problem.

Table 8.2 contains the number of objective function calls used by the pattern search to find solutions to the example problems. The number of objective function calls which would be required by an exhaustive search have been included to indicate the relative complexity of the run-around coil and chiller problems. The tabulated values for the exhaustive search assume that the continuous variables would be assigned discrete values equal to the initial probe step length of the pattern search, ie: ten discrete values each. The difference in function calls used by the algorithms illustrates the vast improvement in efficiency which can be obtained by implementing an 'intelligent' search which seeks to identify the direction of the optimum and converge upon it at an increasingly rapid rate.

The number of function evaluations used by the pattern search to solve the package chiller problem is in general greater than that for the run-around coil system. This is predictable as the package chiller system has a similar number of constraint functions but an increased number of possible solution points. The increase in efficiency of the modified over the basic pattern search is in the range 20 - 30 %.

Objective Function.	Run-around Coil System.		Package Chiller System.
	Basic Pattern Search.	Modified Pattern Search.	Modified Pattern Search.
System Net Energy Consumption.	130	90	152
Primary Energy Consumption.	-	78	152
Capital Cost.	125	91	83
Operating Cost.	-	65	162
Net Present Value.	-	78	182
Payback Period.	129	104	165
Function Evaluations in an Exhaustive Search. (All Objective Functions).	632×10^6.		6328×10^6.

table 8·2, Number of Objective Function Evaluations.

Although this has not been investigated for each objective function and example problem, the three comparisons chosen have solutions for opposite extremes and mid-range component sizes. The net system energy consumption solution tends to large component sizes, where the capital cost function tends to the smaller sizes. The payback period objective function has a solution which reflects both those of energy consumption and capital cost.

A limitation recognised at an early stage in their development, is that as the solution is approached it is characteristic of both the basic and modified pattern search algorithms to repeat evaluations of previously searched points. Therefore, in order to reduce the calculation time, the most recently searched points are held temporarily in an array for recall should the search require them. The number of points which can be held in the array is equal to the allowable maximum number of design variables defined within the software. This ensures that points for at least half the variables are held in the array when each variable has been searched in the direction of both a positive and negative increment. The values contained in table 8.2 are exclusive of any function values which have been recalled from this array and as such are the function evaluations which require the solution of the system performance equations by the simulation algorithm.

The major limitation of both the basic and modified pattern search is that they tend to converge on a false solution when the optimum lies on a constraint function. This is most evident in the solutions obtained for the capital cost objective functions in which the size of components is reduced until a further reduction is limited by a constraint function. A case in point is in the solution obtained for the coil sizes of the run-around coil system example. Figure 8.9 illustrates a surface plot of the capital cost objective function against the width and height of the supply side coils, these dimensions being the same for both the supplementary heating coil and supply side coil of the run-around coils. The objective function values are in pounds sterling and are given as the increase over the value at the optimum.

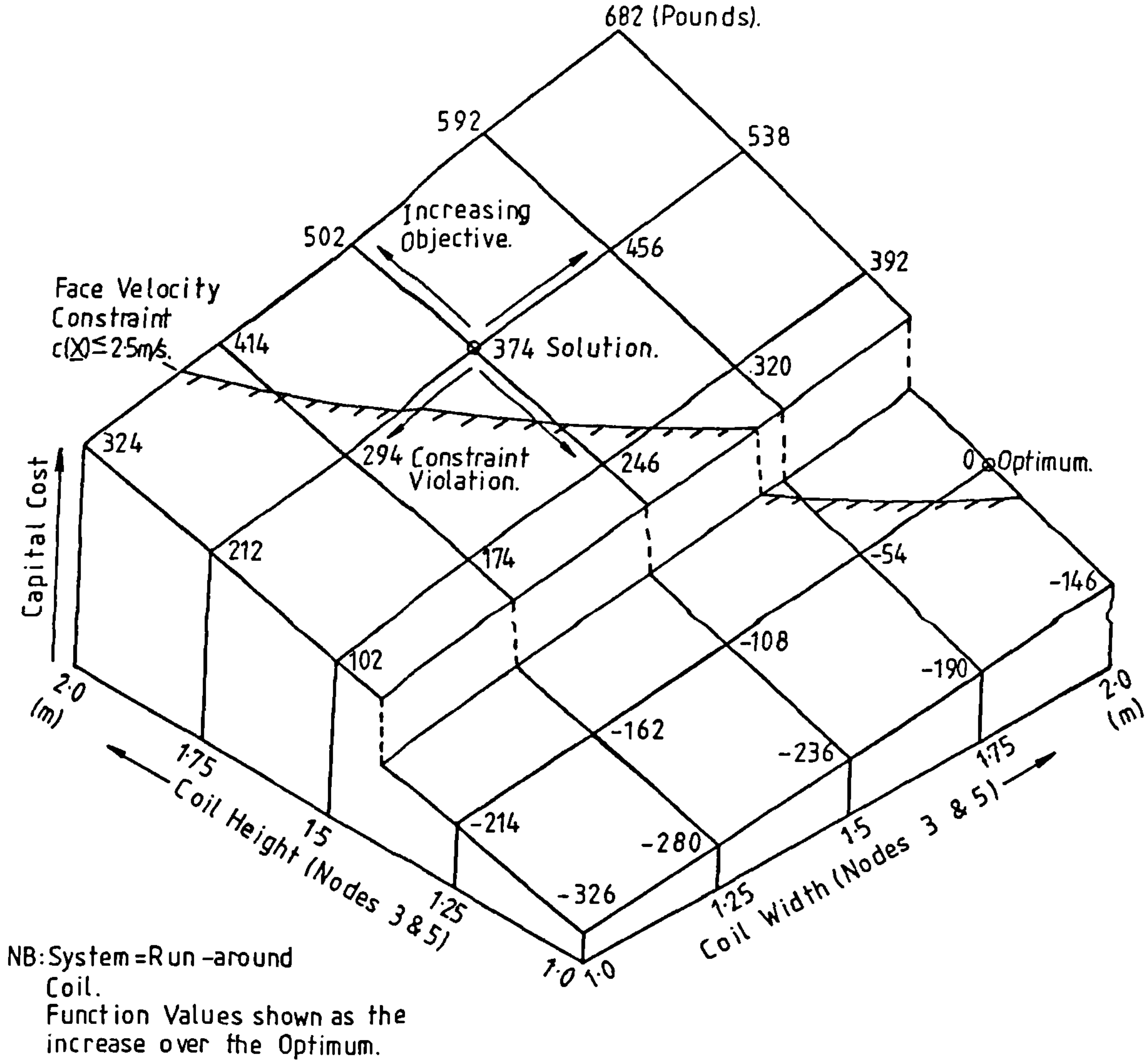


figure 8·9, Failure of the Pattern Search Algorithm.

Once the pattern search encountered the constraint function it failed to progress towards the optimum as on probing along the variable axis, an increase in dimensions produced an increase in the objective function value whereas a decrease in dimensions violated the constraint function thus leading the search with its simple logic to believe that the solution had been found. It is important that this limitation is considered in the future development of the pattern search as the characteristic of the solutions to lie on or close to the constraint functions will render an otherwise robust algorithm unreliable.

8.5 Characteristics of the Objective and Constraint Functions.

In developing an optimisation algorithm it is not only important to confirm the general characteristics of the objective and constraint functions but also to consider any characteristics which are likely to cause numerical instability of the solution algorithm. The characteristic of the HVAC design problem objective and constraint functions have been derived from the swimming pool heat recovery application (chapter 10) used to assess the performance of the pattern search algorithm.

8.5.1 Function Characteristics.

In general, the objective functions can be described as non-linear and discontinuous. Solutions tend to lie on the bounds of the variables or on the constraint functions. Figure 8.10 is a surface plot of net energy consumption for the package chiller heat recovery system, against chiller size and the number of heating coil rows connected to its condenser. The objective function values are in GJ per annum and are given as the increase in value over the optimum solution. This example illustrates the general characteristics of the objective function and the tendancy of the solution to lie on a constraint. It is less usual for solutions to lie in the conventional 'valley bottom' minima associated with the chiller size in this example. Figure 8.9, which is a surface plot of capital cost, illustrates the discontinuous character of this objective function.

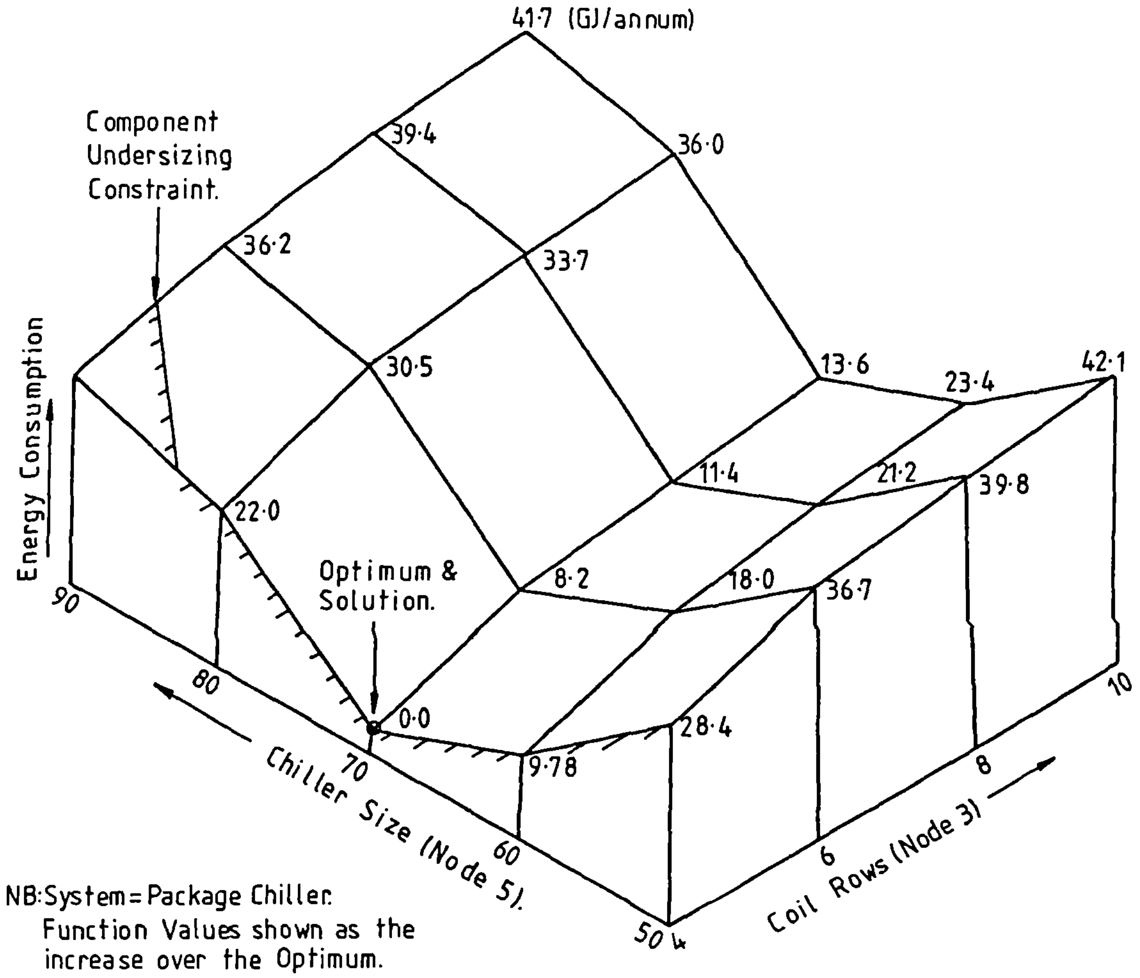


figure 8·10, General Characteristics of the Objective Function and of the Optimum Solutions.

The direction of the optimum is normally well defined and it would be unusual for a search to change direction unless a constraint function was encounterd. Exceptions to this have been noted and one in particular in optimising the design of the example run-around coil system for minimum payback period. Initially, the height of the coils tended towards the largest dimensions suggesting their value was most influenced by the operating cost element of the payback period. As the solution was approached the emphasis changed to capital cost causing a change in search direction with the coil heights tending towards a smaller dimension.

The majority of constraint functions are smooth non-linear functions although sparse non-linear functions can occur in some HVAC design problems.

8.5.2 Numerical Problems.

It is common for the optimum to be independent of certain variables their value only affecting the constraint functions. This is most notable for the capital cost objective function in which the exogenous fluid variables, such as water mass flow rates are rarely parameters in the capital cost models, but are important only in ensuring the correct sizing of components. The value of such variables remains unchanged during a search until a constraint function is encountered at which point their value is varied such as to allow the search to move along the constraint towards the optimum. It is not envisaged that this will be a major cause of numerical instability as most direct search algorithms can be adapted to operate on a subset of variables until a constraint is encountered and further variables become active.

A similar problem which is more significant arises when the optimum value of a variable is only marginally influenced by the objective function value. Numerical instability can occur for energy related objective functions when either a change in value of a variable or combination of variables produces a small change in the value of the objective function.

The cause of this instability is that when small changes in energy related objective functions occur, the value of the change is more influenced by the accuracy of the system operating point found by the system simulation algorithm, than by the actual change in component performance.

Provided that a feasible system has been specified, the simulation solution algorithm reduces the sum of the component residual performance equations until it is less than a predefined value which has been derived to achieve a suitable level of accuracy. As this does not necessitate a zero sum of residuals, it is likely that the residuals themselves will, although small, be non-zero. The simulation solution algorithm will find equally accurate solutions for different values of problem variable, but because it solves a different set of conditions the final value of the residuals is different in each case. Although these differences are small they can combine to produce errors in an otherwise stable objective function.

This behaviour is exhibited by the net system energy consumption objective function of the run-around coil example. Figure 8.11 illustrates the unstable nature of the objective function in relation to the supply fan diameter and depth of additional heating coil. Because the supply fan is positioned down stream of the heating coil the temperature rise across the fan offsets the coil duty. Increasing the coil depth results in a lower coil duty as the increased air pressure drop increases the fan power and the temperature rise across the fan. All other components in the system are unaffected and should have a constant duty regardless of the coil depth. Table 8.3 contains values for the variation in duty of the supply and extract fan of the heating coil against the coil depth. These changes correspond to a supply fan diameter of 0.9m in figure 8.11. Reducing the coil depth from 6 to 4 rows reduces the duty of the supply fan which in turn results in an increase in coil duty, the duty of the extract fan remaining constant. It would be expected that a further reduction in coil rows from 4 to 2 would lead to the same characteristic, yet in this case an erroneous change in the duty of the extract fan occurs. This marginal change in operating point found by the simulation solution algorithm influences the objective function because both the change in system operating point and the true change in objective function are small.

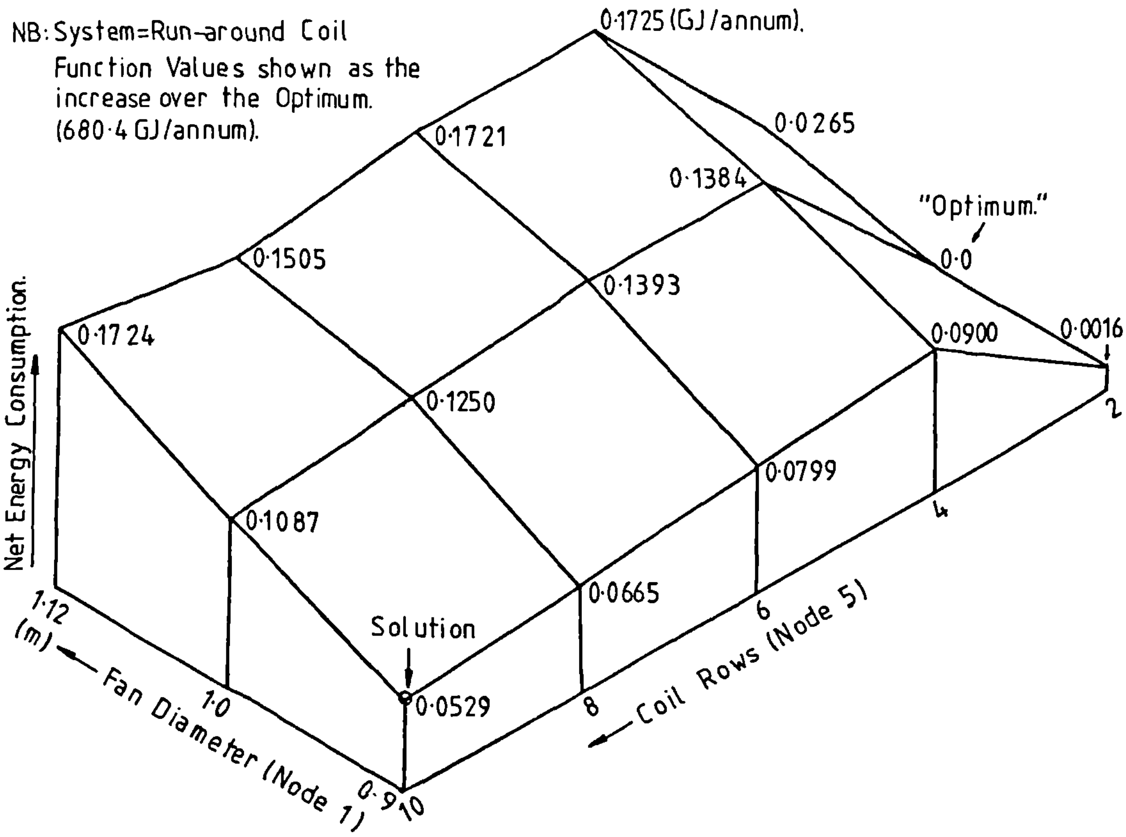


figure 8·11, Numerical Instability of the Objective Function.

Component.	Component Duties (KW).		
Supply Fan (Node 1).	5·3724	5·4515	5·5316
Extract Fan (Node 2).	5·5573	5·5541	5·5541
Heating Coil (Node 5).	32·8206	32·7504	32·6697
Coil Rows (Node 5).	2	4	6

table 8·3, Unstable Change in Component Duty.

Numerical instability of this type is often corrected by scaling of the variables and objective function. Before such measures are considered it is prudent to examine the integrity of the objective function as in this case the instability is an inherent characteristic of its formulation. Two factors affect the stability of the objective function in this case, the accuracy of the system operating point found by the simulation solution algorithm and the change in component performance which occurs for an increment in value of problem variable.

The significance of erroneous changes in system operating can be assessed by comparing them with the accuracy of the component models, as in a component based simulation it is the accuracy of the component models which dictates the accuracy of the solutions. Inspection of table 8.3 indicates that the erroneous change in system operating point which occurs when reducing the coil depth from 4 to 2 rows, produces an erroneous change in extract fan power of 0.06%. This is insignificant when compared to the +10% to which fan power is measured (BS. 848, 1980) and therefore implies that the solution found by the simulation algorithm is of sufficient accuracy.

Assessing the significance of true changes in system operating point for an increment in value of a problem variable, can prove more difficult. A small change in fan power can appear insignificant in itself but when this change is related to say running costs and integrated over the life of the building it becomes more meaningful. A 1.5% change in supply fan power for an increment in coil rows, table 8.3 produces a 0.0015% change in the net system energy consumption (figure 8.11). Yet the same change in fan power when applied to the primary energy consumption results in a more meaningful and stable change of 0.3% (figure 8.12). Therefore the significance of such changes can only be assessed in relation to their affect on each objective function.

As mentioned previously, it is common to improve the numerical stability of optimisation problems by scaling the problem variables and objective function. Scaling of the variables attempts to ensure that unit increment in value of any variable produces the same change in value of objective function.

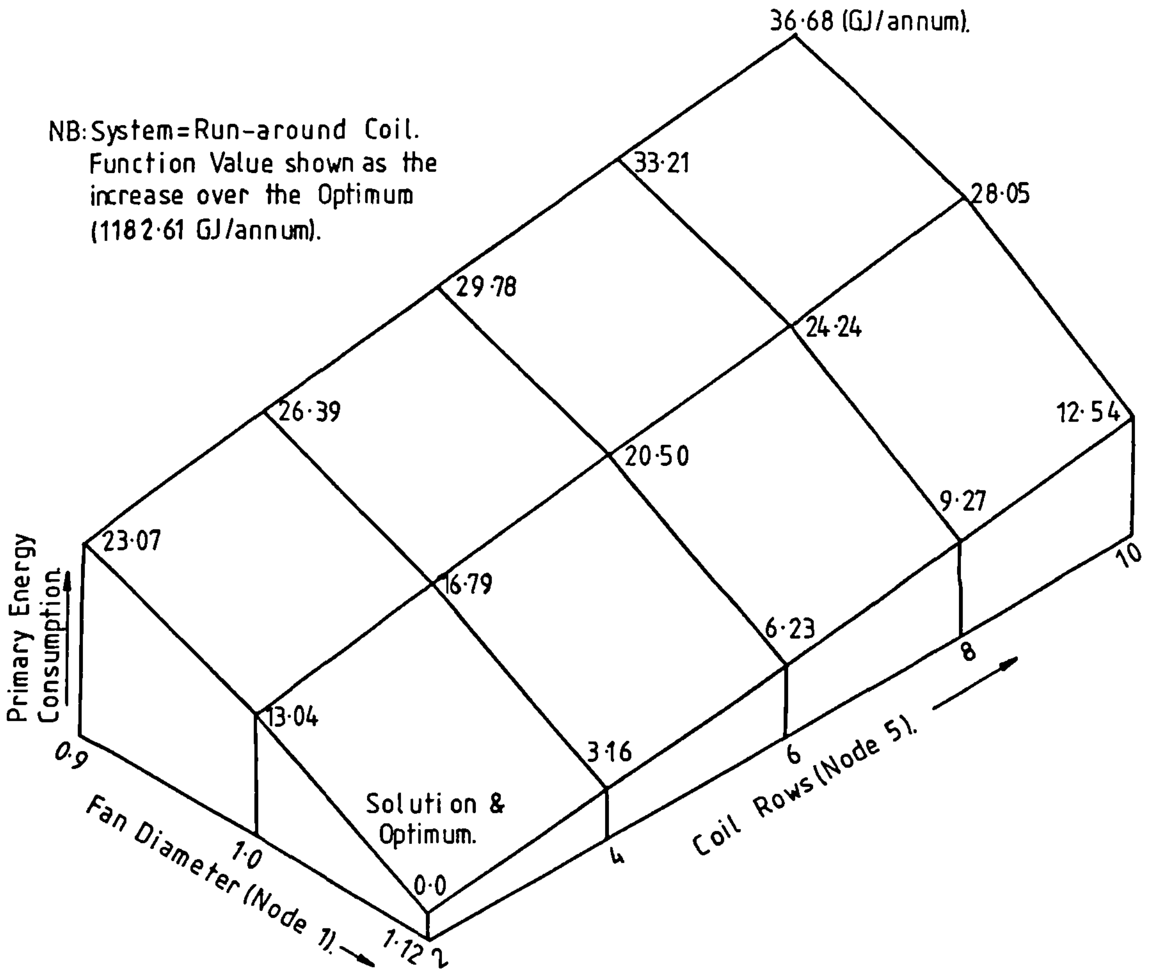


figure 8·12, Stable Energy Objective Function.

This is important in derivative search methods as it helps establish convergence and the selection of differencing intervals. Variable scaling might improve the efficiency of direct search methods by ensuring all variables converge on the solution at an equally rapid rate, yet the effectiveness of this in HVAC design would be impaired as the increment in variable values is restricted for discrete variables. Further, variable scaling will not improve the numerical stability of energy related objective functions as this is not concern with the differences in objective function value for increments in each variable value, but is related to the errors that are introduced when the change in objective function value is small. Similarly therefore, scaling of the objective function by adding a constant to it or multiplying it by a positve constant, will not eliminate error inherent in its formulation.

8.6 Development of an Idealised Solution Algorithm.

An ideal solution algorithm is one which matches or is tailored to the characteristics of the optimisation problem. The cautious nature of the pattern search algorithm in repeating exploratory and accelerated pattern moves is well suited to the characteristics of HVAC design objective functions, as although the direction of the optimum is generally well defined, this can change as the solution is approached. A major failing of the pattern search algorithm is its inability to adapt to the characteristics of solutions to lie on constraints. Once a constraint is encountered it is impossible for the current pattern search algorithm to move along the constraint and converge on the optimum and therefore it is in the area of constraint handling that development of an algorithm is required. Other associated areas of development which are required if the algorithm is to be used in a real design environment are in finding an initial feasible solution, confirmation of the optimum solution and improved numerical stability of the objective function.

8.6.1 Improvements in Constraint Handling.

Several of the constraint handling techniques developed for use with direct search methods employ the derivatives of the constraint and objective functions. These however are inappropriate for solving HVAC optimised design problems as the discrete nature of the problem variables limits the available differencing interval which leads to problems of stability.

Of the non-derivative methods of handling non-linear constraints, perhaps the most widely used and successful have been the penalty function transformation methods. These techniques transform the constrained problem into an unconstrained one by imposing a penalty on the objective function in the region of the constraint. Penalties may be imposed as the search nears the constraint or not until the constraint is violated, although the latter technique is inappropriate for solving HVAC optimised design problems as when a component undersizing constraint is violated the objective function, if related to energy consumption is unobtainable.

One of the earlier internal penalty functions was developed by Rosenbrock (1960) and imposes a penalty function only within the narrow region of the constraint. This however is likely to prove difficult to use with both discrete and continuous variables as the closeness of a discrete variable to a constraint and therefore the region of penalty, is limited by the interval between values of the discrete variables.

A more applicable approach is in the created response surface technique developed by Carroll (1961). This imposes a penalty on the objective function over the whole variable space and reduces the weighting of the penalty in a sequence of unconstrained optimisations.

The modified objective function is of the form:

$$F^*(\underline{X},r) = F(\underline{X}) + r \sum_{i=1}^{m} W_i \;/\; c_i(\underline{X}) \quad , \; r>0, \; W_i>0$$

where W_i is the weight of the various constraints one against the other and r determines the affect of the constraints compared with the original objective function. As a constraint is approached the reciprical of the constraint will tend to infinity and so it is hoped that the search will not cross the constraint boundary, $c_i(\underline{X})=0$.

Two problems are envisaged in implementing this technique: firstly the complexity of design problems which can be solved by this technique will be restricted, as due to the repetitive optimisation required to reduce the weighting, r, the number of objective function evaluations will be high. Secondly, it is likely that the fixed increment between values of the variables will lead to moves in the search which violated the variable bounds and cause the search to fail. Swann (1978) devised procedures for use with the pattern search which reduce the affect of both of these problems. As the probe step lengths are a measure of the accuracy of the current solution and progress of the search, Swann suggests that the weighting, r, may be reduced at the same time as the probe step lengths, thus limiting the need for repetition of the optimisation. As for violation of constraints, Swann adopted the policy that exploratory moves which violate the constraint are rejected but when a pattern move violated a constraint, the search was allowed to perform an exploratory move about the infeasible point in the hope that a feasible point which reduced the objective function would be found.

It is likely that combining the created response surface constraint handling technique with the pattern search will provide a useful algorithm for solving HVAC optimised design problems. The limiting factor on its implementation is in the formulation of component undersizing constraints as the technique requires meaningful constraint formulations within the feasible region. An optimisation procedure which does not suffer from this problem and requires only simple checks for feasibility is the 'complex' method devised by Box (1965).

The 'complex' method is a variation on the simplex method which explores the variable space with a regular simplex of n+1 mutually equidistant points, n being the number of variables. The simplex method operates by replacing the vertex with the highest objective function value, with a point reflected about the centriod of the other vertices, thereby creating a new simplex. The basic simplex method has been modified to incorporate expansion and contraction moves which enable the simplex to adapt to the local geometry of the objective function. Constraints have been included by assigning a large positive value to vertices which violate constraints thus ensuring they are rejected. In practice, Box (1965) found that this simplex procedure tends to flatten itself against the constraint before the optimum is reached. He therefore developed a new constrained procedure, using $q > n+1$ vertices, termed a 'complex', the extra $(q-n-1)$ vertices aimed at preventing the complex losing dimensions when constraints are encountered.

Construction of the complex begins from a supplied initial feasible point $\underline{X}^{(0)}$. The remaining q-1 vertices are generated one at a time such that:

$$x_i^{(j)} = lb_i + r_i\,(ub_i - lb_i) \quad , \; i=1,2.....n, \; j=1,2.....q-1$$

where r_i is a pseudo-random deviate rectangularly distributed over the interval (0,1). All generated vartices lie within the bounds of the variables but may violate constraint functions. Such points are moved back towards the centroid of the remaining vertices until they become feasible. The search proceeds in a similar fashion to the simplex method with the worst vertex reflected about the centroid of the remaining vertices. However, unlike the simplex method when a constraint is violated the search adopts the method of moving the vertex towards the centroid of the remaining vertices until it becomes feasible. Convergence is assumed when five successive moves yield no improvement in the objective function. A problem which may arise in applying the complex method to HVAC optimised design problems is that the discrete variables may restrict the movement of vertices which violate constraints and as a consequence cause the search to fail. However the effect of this problem may be reduced by ensuring a sufficiently high number of vertices.

Both the complex and response surface-pattern search methods should be implemented as part of future research. The response surface approach has the advantage that it is easily adapted for use with the existing pattern search algorithm and is likely to perform well with discrete variables. Its disadvantages are that it requires rigorous formulation of the constraints within the feasible region and may demand a prohibitive number of objective function evaluations. Conversely, the complex method does not require a rigorous constraint formulation, but may fail due to limitations imposed by discrete variables. Implementing both procedures and comparing their relative performance will help delineate future algorithm development.

8.6.2 Improving Numerical Stability.

Numerical instability of energy related objective functions occurs when changes in component performance are so small that the corresponding change in the objective function is more influenced by the accuracy of the system operating point than by the actual change in component performance. Research is required to determine the smallest change in value of the objective functions which remain unaffected by erroneous changes in the operating point. Such changes are a measure of the obtainable accuracy of the solution and as such can be used in convergence criteria and to improve numerical stability by ensuring that an increment in value of the variables is sufficiently large that errors are not introduced, or where this is not possible within the defined variable space, that the variables which do not produce a sufficiently large change in objective function value are removed from the 'active' variable set.

8.6.3 Finding an Initial Feasible Point.

Both the complex and pattern search algorithms require a given initial feasible solution from which to begin their search. The initial feasible point, in this research has been found by inspection. If however the optimised design procedure is to become a useful design tool, an initial feasible solution must be found automatically.

A widely used and reliable technique of finding an initial feasible point is to reduce the value of the sum of the violated constraint functions, $c_i(\underline{X}) \leq 0$, to zero using an optimisation algorithm, ie: reduce $F(\underline{X})$ to zero where $F(\underline{X})$ is given by:

$$F(\underline{X}) = \sum c_i(\underline{X}) \quad \text{for all } c_i > 0$$

The existing pattern search or any future implemented algorithm should prove reliable in finding a solution. Failure to find a solution does not however in itself disprove the existance of a feasible point, since this can only be interpreted from the final value of the variables and constraints still active at the point of failure. This method of finding an initial feasible point relies on robust constraint formulations: should this prove difficult to implement a procedure which operates on a random search should provide an alternative method.

Although the initial point may be within the feasible region, the search may fail to start if the feasible point lies in the flat region of the payback period objective function. The formulation of the payback period objective function is such that the maximum value obtainable is equated to the life of the building. This produces a flat region in the objective function in which the search may flounder. This problem did not arise during this research, but it may be prudent to reformulate the payback period objective function to allow an unlimited value.

8.6.4 Confirmation of an Optimum Solution.

Unlike derivative methods which employ rigorous mathematical tests, the convergence criteria for direct search methods are usually based upon a law of diminishing returns. For example, convergence in the complex method is assumed when five successive evaluations of the objective function yield no change in the optimum. Such simple strategies can lead to premature convergence and false solutions. However, in HVAC optimised design the shape of the objective functions and direction of the optimum are so well defined that a more sophisticated check for convergence is unlikely to be necessary once a robust search algorithm has been developed.

Should, this however not prove to be the case it is common to check the solutions of direct search methods by restarting the search from a different feasible point, the assumption being that the search should find the same solution. As this technique reduces the efficiency of the optimisation procedure and is expensive on computer time, it may prove more economic to implement a random search procedure to check for a lower objective function value.

8.6.5 Formulation of the Constraint Functions and Development of a Simulation Solution Algorithm.

Successful implementation of future optimisation algorithms relies heavily upon the rigorous formulation of the constraint functions. Three aspects in particular require further research:

1. Formulation of the fluid related constraints with respect to the change in load on the system.
2. Formulation of the undersizing constraint function within the feasible region.
3. Formulation of the undersizing constraint function in the infeasible region.

The variation in system performance induced by changing load conditions produces different constraint function values for each time period in the load profile. The simplest approach to constraint formulation here is to provide constraint values for each time period in the profile, although this would require considerable calculation time and would lead to an unwieldy and possibly unmanageable optimisation problem. Research therefore is required to develop a means of constraint formulation in which the number of constraint values required to represent the behaviour of the constraint over the load profile, is limited to a manageable number. Several possibilities exist in this respect the most reliable of which is likely to be to use the worst value of the constraint evaluated over the profile. Consideration should also be given to the calculation time as simulating the system performance for each time period in the profile is costly especially when the solution is already known to be infeasible.

Implementation of the Created Response Surface technique requires rigorous constraint formulations within the feasible region. All of the existing constraint functions, except the component undersizing constraint are valid in this respect. Constraints representing the 'closeness' of a components performance to its limits can be formulated from the component performance envelopes. For example, the limits of an axial flow fans performance could be expressed by a maximum and minimum blade angle β, a maximum fan static pressure P_{max} expressed as a function of volume flow rate V and finally a minimum fan static pressure of zero (figure 6.2). This would provide four non-linear constraints of the form $c_i(\underline{X}) \geq 0$, where the four functions are:

$$c_1(\underline{X}) = \beta_{max} - \beta$$
$$c_2(\underline{X}) = \beta - \beta_{min}$$
$$c_3(\underline{X}) = P_{max} - P \quad (\text{where } P_{max} = f(V))$$
$$c_4(\underline{X}) = P - P_{min} \quad (\text{where } P_{min} = 0)$$

This formulation is only valid within the performance envelopes as extrapolation of component performance beyond the known and measured performance is unreliable and meaningless.

None of the optimisation algorithms suggested for future implementation require rigorous constraint formulations outside the feasible region. Yet this is prerequisite of developing a procedure to find an initial feasible solution. The approach suggested above for the formulation of an undersizing constraint within the feasible region is not valid here as the occurence of undersized components is indicated by failure of the simulation solution algorithm. Formulation of component undersizing constraints within the infeasible region is therefore restricted to utilising the charcteristics of the simulation parameters on failure of the algorithm. The characteristics of the parameters and suggestions as to how they may be used to formulate constraints is discussed in section 7.5 and appendix C.

Successful implementation of the component undersizing constraints relies upon future development of the simulation procedure and in particular the availability of unique characteristics associated with failure due to undersizing as apposed to instability. Improved stability of the simulation procedure is also required to ensure spurious failures do not mislead the optimisation algorithm and result in false solutions. Suggestions as to how stability might be improved through the development of a constrained simulation procedure are discussed in section 6.4. Use of the optimisation procedure in the design of large systems is restricted by the poor solution time of the simulation procedure. The applicability of the optimisation algorithm to the design of full systems can only be fully validated by using it to design such systems and therefore the future development of the optimisation algorithms depends upon the improved computational speed of the simulation procedure.

Chapter 9. PROGRAMMING AND SOFTWARE DEVELOPMENT.

A modular software structure has been implemented in this research because it allows the characteristics of individual elements in the optimisation problem to be investigated and it gives the greatest flexibility in developing a new procedure, new developments accommodated by changes to individual modules or the addition of further modules. An added advantage of a modular structure is that a rationalised more 'intelligent' version of the software can be developed at a later date simply by combining modules in an orderly fashion to perform predefined tasks.

The most important criterion affecting the software framework in this early research, is that it should allow various optimisation solution algorithms to be implemented and evaluated without major changes in the software. Again, a modular approach lends itself to this requirement as each new solution algorithm can be accommodated by implementing a new module or several new modules.

Integration of the optimised design software with the existing simulation software has been programmed such that parallel development of each procedure can continue with any changes affecting only the interfacing software. Retaining some distinction between elements is also important for the commercial development of the software as not every customer would want to purchase both the simulation and optimised design packages.

In order to distinguish the optimised design procedure and software developed in this research from that by other researchers, the suite of optimised design programs is collectively referred to as ODESSY, (Optimised DESign SYstems). The current procedure is accessed through the Loughborough University simulation software SPATS (Murray 1984), although after future rationalisation ODESSY will appear as a separate software package. The existing version of ODESSY operates at two levels: the first enables control of the design procedure through access to the design functions and the second allows specific operation of those functions (figure 9.1).

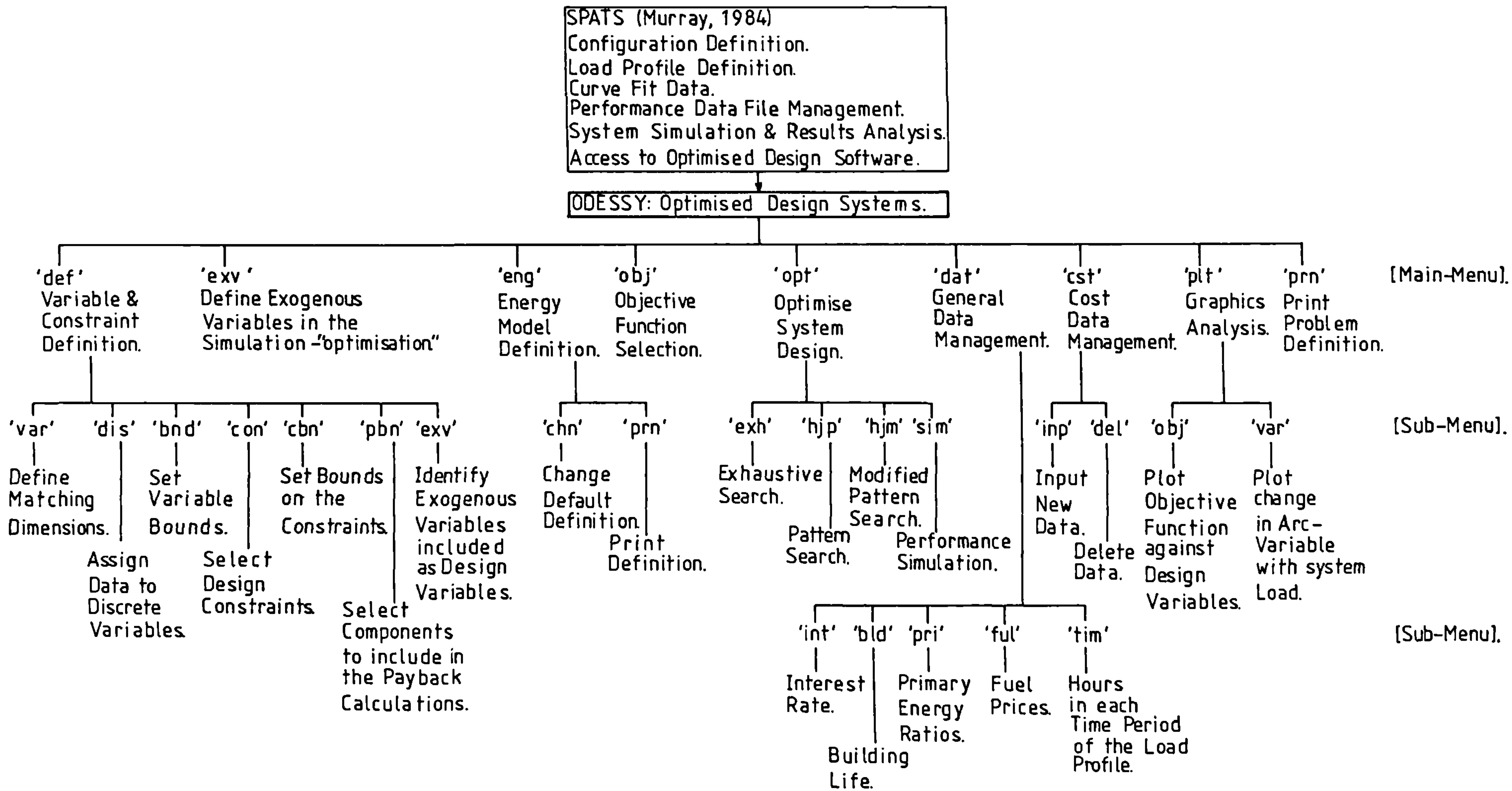


figure 9·1, Structure of the Optimised Design Software.

The following description of the major segments of ODESSY is not intended to be employed as a user manual or to be a complete description. but has been written to illustrate the general approach to formulating the software and the features of the major functions. Commands are referenced by enclosing them in inverted commas, 'command' and subroutines in brackets, <subroutine>.

9.1 Machine Implementation and Program Language.

The scale of HVAC system design problems and the complex numerical procedures required to solve the system performance equations, limits the implementation of the simultion and optimised design software to mainframe computers, although a single component or small sub-system procedure could be implemented on a smaller machine. The current version of the software has been implemented on a Honeywell mainframe computer with a Multics operating system. The major feature of this system is that program segments can be linked dynamically during run time allowing each individual user to develop his own component models and program segments without compiling the complete program each time a change in software is made. This an advantage in the parallel development of the simulation and optimised design software as each procedure is easily developed by different researchers.

Apart from machine dependent commands, the software has been written in standard FORTRAN77 (ANSI, 1978), this language allowing a modular program structure and being suitable for numerical problem solving. Extensive use has been made of common blocks reducing storage requirements in transfering data between segments. Parameter statements have been used to define the size of data arrays and therefore the size of problem manageable by the software (appendix D.)

In most cases, data input is checked against the variable type expected, ie: integer, character or real: any error is used to direct the program control to a re-input request. Data files have been used throughout the program to hold component performance data, problem definitions and general design data. Each file is structured in format and can be expanded automatically to accommodate additional data. Most files are labelled in two parts, the first part identifies the particular component or system to which the file is attached and the second part the data contained in the file (table 9.1). Several different load profiles can be defined within SPATS and therefore a further identifier number is attached to these files. Each user of the software has a set of data files the contents being tailored to his particular needs.

9.2 Integration of the Simulation and Optimisations Software.

Three elements of the simulation and optimised design software overlap: definition of the problem variables, changing the size of components within the problem and simulation of the system performance. The first, definition of the problem variables is performed initially in an interfacing subroutine which associates the system variables of the configuration definition with the optimised design problem variables. Changing the size of the components specified within a problem and running the system simulation are common functions of both the simulation and optimised design software, the only difference being that the optimised design software performs the tasks automatically where as in the simulation software, control is retained by the user.

9.2.1 Problem Initialisation and Default Definition.

Each new problem definition begins with a description of the system configuration and load profile. These tasks are performed under the control of SPATS and each system definition and profile labelled with an appropriate system name. Control can then be passed from SPATS to ODESSY whereupon an interfacing subroutine <setup> is called. This subroutine checks for the existence of a configuration definition, previous design problem definition and can initialise a default problem definition.

File Name:	Source:	Description:
BASEDATA	ODESSY	General Optimed Design Data.
Component.CST	ODESSY	Component Cost Data.
Component.DIR	SPATS	Component Performance Data.
System.CAL	ODESSY	Optimisation Search Points.
System.DES	ODESSY	Optimised Design Problem Definition.
System.NET	SPATS	Configuration-Simulation Problem Definition.
System.No.PRO	SPATS	Load Profile.
System.No.RES	SPATS	Results from the System Simulation.

table 9·1, Data Files.

Subroutine <setup> begins, checking for a previous problem definition by searching for an optimised design data file with the specified system name. If previously defined the definition is read from the data file and retained by the problem arrays and directories. Control is then passed to the first command level of ODESSY.

If the optimisation problem is being defined for the first time, <setup> checks that the system configuration is defined and if it is not, returns control to SPATS to allow its definition. Once the configuration is defined, <setup> intialises a default problem definition which is formulated by interpreting the system configuration and the default parameters defined in the component intialisation subroutines.

The majority of the default definition is formulated node by node for each component in the system. The size of each component as described by a data record and exogenous constants, is defined first in the directory and arrays VARDIR, DVARNM and CONNAM (section 4.2). Each component intitialisation subroutine contains default energy parameters and a description of the number and type of constraints. These are used in the formulation of the energy model directories ENGDIR and ENGDRC (section 4.2.3), the default model being formulated for a complete system which therefore contains only direct energy terms. All component design constraints are included in the default problem and are defined within the formulation of the constraint directory, CONDIR (section 4.2.2). The node by node assignment of component parameters is followed by extending the problem variables definition to include all exogenous variables. Default values are assigned to the general design data (section 4.2.4) and a default objective function of primary energy consumption selected. The completed problem definition is stored on a structured data file which is identified by the system name appended with the characters .DES.

9.2.2 Changing System Variable Values.

At each new solution point the optimisation algorithm requires a change in value of the problem variables. Subroutines to change the component data records and values of the exogenous variables and constants, are an integral part of SPATS. However, these have been designed to operate under user control and therefore demand input from the user which is supplied automatically in the optimisation procedure. The most elegant and efficient means of developing a single set of subroutines which operate under control from the user or optimisation procedure would be to arrange the software such that input commands were only initialised when required under user control. During the initial programming of the optimisation procedure, implementation of this approach would have been disruptive to the development of SPATS. This together with the initially volatile nature of the design procedure led to a duplication of subroutines, those used by ODESSY being stripped of their input commands, the input data supplied automatically by the optimisation procedure.

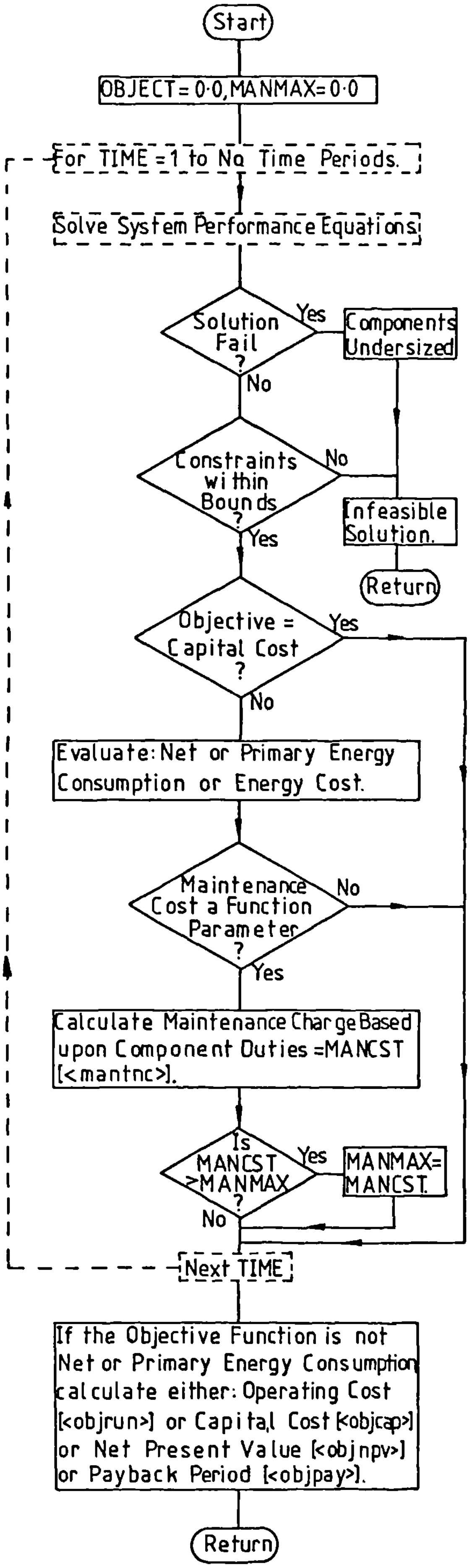
A further difference in operation arises in the subroutines which change component data records. Changing a specified component data record in SPATS results in a resetting of the bounds and initial guess of the arc-variables associated with that component. This has been implemented since each component size has a different operating range and therefore bounds on its variables. However, during operation of the software, greater stability of the simulation solution algorithm, over a range of component sizes, has been achieved by fixing the arc-variable bounds and initial guess once stability of the initial point has been found. Hence, in ODESSY changing a component data record leaves the arc-variable bounds and guess unchanged.

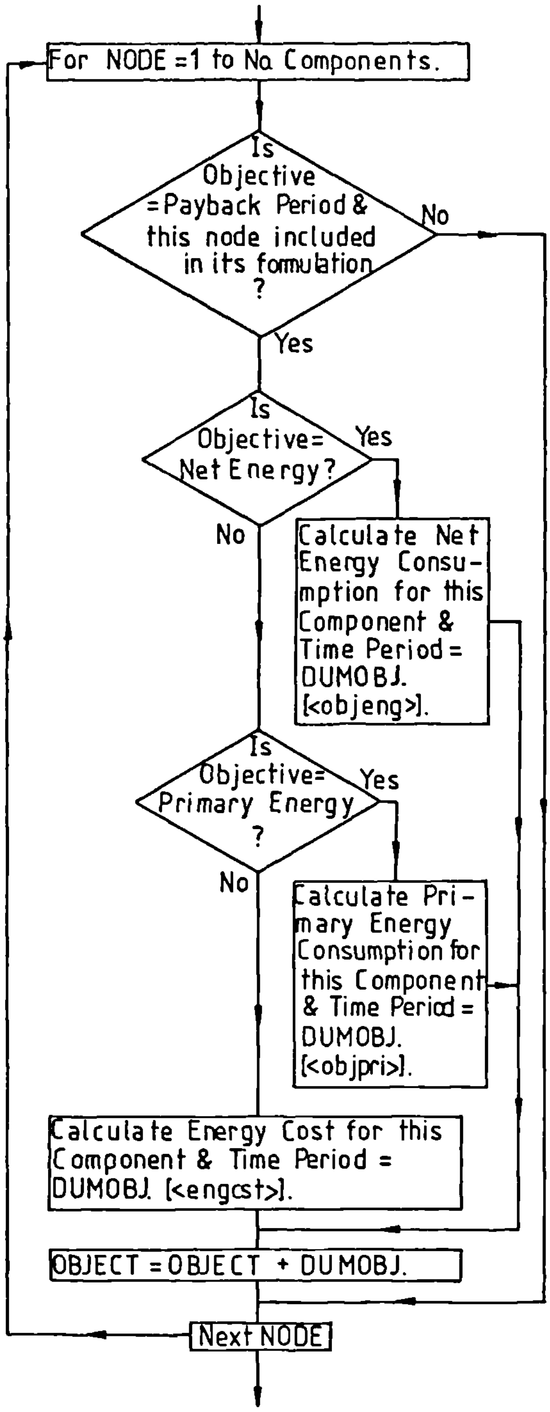
9.2.3 Performance Simulation, Objective Functions and Constraints.

Each evaluation of a new solution point by the optimisation algorithm not only leads to a change in value of the system variables but also requires simulation of the system performance, the results of which are used in evaluating the constraint and objective functions. Again the simulation subroutines are an integral part of SPATS but differ in operation for the optimisation procedure in that they must be structured to allow evaluation of the load profile dependent objective and constraint functions and must operate automatically with data supplied by the optimisation procedure and not the user. The generalised simulation solution algorithms are interfaced with SPATS by a subroutine which scales the variables and initialises the load profile. These subroutines have been duplicated and modified within the optimised design software ODESSY, as modification within SPATS would have proved disruptive at the time of programming.

A generalised algorithm which determines the system operating point and evaluates the constraint and objective functions is illustrated in figure 9.2. The operations enclosed by broken lines are those of the original SPATS subroutines whilst those of solid lines are the additional operations required by the optimised design procedure. This algorithm has been implemented for each of the simulation solution algorithms, but in future rationalisation will form a single subroutine in which any specified simulation solution algorithm can be called.

In an effort to economise on calculation time the constraint functions are evaluated before the objective functions and control returned to the optimisation algorithm if the constraints are violated. This approach is possible since the constraints are used simply to reject infeasible solution points, but future solution algorithms will require amore rigourous constraint definition and therefore formulation. Violation of component undersizing constraints is recognised by failure of the simulation solution algorithm and the violation of other constraints by evaluating the constraints within the component models and checking their values against the constraint bounds.





Evaluation of System Energy Functions.

figure 9·2, Evaluation of the Constraint and Objective Functions.

It is important to note that exogenous variables defined as optimisation problem variables have values which vary only in the optimisation and remain unchanged by the operation of the load profile. The implementation of additional objective functions is easily incorporated into the procedure by calling the appropriate calculation subroutine at the end of the general evaluation algorithm (figure 9.2: section 9.4).

9.3 Problem Definition.

The default problem definition initialised in the interface subroutine <setup> is somewhat incomplete: all exogenous variables are defined as problem variables where clearly some must remain as exogenous load variables, the matching dimensions of adjacent components, bounds on the variables and component product ranges are all undefined. Although some of these parameters could be given default definitions, most are specific to each design problem and therefore should be defined separately. Software for the definition of these parameters is accessed through the main command level of ODESSY (figure 9.1). The option 'exv' which allows definition of the exogenous variables to be optimised in the GRG2 simulation is largely redundant as this optimisation is of limited use (section 6.2).

9.3.1 System Operating Variables.

All exogenous variables are defined as problem variables in the default definition, but some must be variables in the load profile. Those to be retained as design problem variables and therefore whose value is not changed during simulation over the profile, are defined using the ODESSY sub-menu option 'exv'. Executing this command accesses the subroutine <setdex> which first recalls the existing problem definition from the data file and then displays all exogenous variables. The exogenous variables to be retained as design problem variables are allocated by the user and defined using the variable directory VARDIR.

9.3.2 Matching Adjacent Dimensions.

Software which allows dimensions of adjacent components to be formulated as single problem variables are accessed through the ODESSY sub-menu command 'var'. Three subroutines are employed in the matching of dimensions: <setvar>, <assvar> and <edvar>. The first, <setvar>, acts as a general control routine enabling the adjacent components with matching dimensions to be identified. Control is passed from <setvar> to either <assvar>, which allows the adjacent dimensions to be matched, or, when the existing definition contains matched dimensions to <edvar> which allows the matched dimensions to be separated (figure 9.3).

Subroutine <assvar> operates by redefining the relationships between system parameters and problem variables described in the variable directory VARDIR (section 4.2.1). The subroutine sorts the directory to ensure the redefined problem has a sequentially numbered set of variables. Any discrete data and variable bounds previously assigned to the dimensions before matching are removed as these are not necessarily the same for all the unmatched variables.

Subroutine <edvar> allows the matched dimensions assigned to a particular problem variable to be separated into individual variables. Each variable defined by the user for separation is checked to ensure it is assigned to more than one dimension and that it is not a system exogenous variable. The individual dimensions are displayed and where more than two dimensions are assigned the user can specify which dimensions are to be separated. The subroutine separates the dimensions by creating a new problem variable described by a row in the variable directory VARDIR. New variables are inserted into the directory in a position which corresponds to the initial sequential problem definition. Discrete data values and variable bounds of the matched dimensions are retained by the separate dimensions.

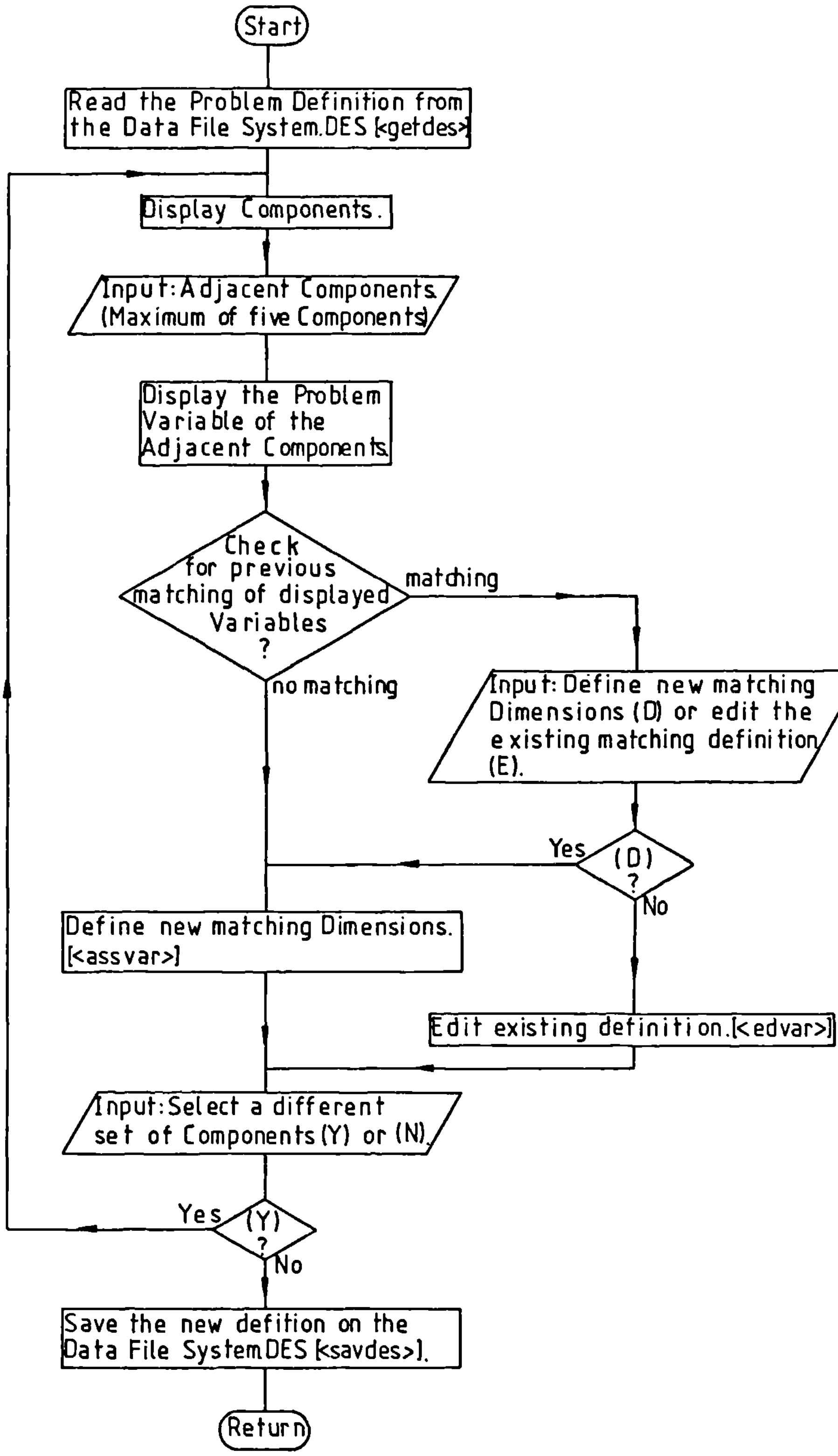


figure 9·3, Adjoining Dimensions Definition Algorithm (<setvar>).

9.3.3 Product Range Definition.

Component product ranges are described by the bounds on the continuous variables and more commonly by the values assigned to the discrete problem variables. Software for the definition of discrete data values is accessed through the ODESSY sub-menu option 'dis'. Execution of this command accesses the subroutine <setstp> and as most problem variables are discrete, this subroutine gives the option of a sequential specification for all variables or, alternatively, specification of data for one variable only. Each variable for which new discrete data is to be assigned is displayed with its matching dimensions and previously assigned discrete data. New discrete values and any corresponding data record names are supplied by the user. Data can be supplied randomly as the subroutine checks the input to ensure that it is stored as a rising series of numbers. Discrete values are held in the array STPVAR and the corresponding data record names in the character array COMFIL (section 4.2.1).

9.3.4 Bounds and Constraint Functions.

Design problems are restricted to realistic solutions by the application of variable bounds and constraint functions. Software for the definition of these parameters is accessed through the ODESSY sub-menu commands 'con', 'cbn' and 'bnd' (figure 9.1).

The command 'con' accesses the subroutine <setcon> which allows definition of the contraint functions to be included in the design problem. Every constraint function is included in the initial default definition and therefore <setcon> gives the option to reset the constraints for all or selected components. Each component specified is considered in turn, giving the user the option to delete any of the constraints in the existing definition or to include any previously excluded. The directory CONDIR and the arrays CONLB and CONUB are then sorted accordingly to ensure a sequential set of constraint functions. Subroutine <setcbd> allows the subsequent definition of bounds on the constraint functions and again the option is given to define the bounds for all or selected constraints.

Variable bounds are initialised by the subroutine <setbnd>. When called, <setbnd> automatically sets the bounds of the discrete variables to comply with the range of discrete data. The subroutine then allows the bounds of selected variables to be defined by the user: narrower bounds than those automatically specified are allowable if required. Bounds specified for each variable are checked to ensure the lower bound has a smaller value than the upper bound and for discrete variables, to ensure that the bounds specified lie within the range of discrete data.

9.3.5 Energy Model and Payback Period Components Definition.

The default system energy model, comprised of direct energy terms, can be redefined by calling the subroutine <seteng>. This subroutine displays the active energy terms of each component and allows the user to specify the components for which terms are to be changed. Each energy term of the specified components is then either excluded from the system model or redefined by associating it with a fuel type and an addition/subtraction term. The new definition for each component is described in the energy model directories ENGDIR and ENGDRC (section 4.2.3).

Not all components in the system are necessarily included in payback period calculations: those to be included can be specified by calling the subroutine <paynod>. Definition of the payback period components is held in the array PBNODE (section 4.2.4).

9.3.6 General Design Data.

The general design data includes the interest on borrowed capital, building life, primary energy ratios, fuel tariffs and time period associated with each interval in the load profile. As each of these is subject to change, the default values used in the initial problem definition are stored on a data file and can be updated by calling the subroutine <basdat>. Similarly, <basdat> allows the default values initially assigned to a design problem to be modified to suit a particular application. The general design data is held in the arrays FUELS, PRIRAT and in the variables BLDLIF, INTRST and TIMPRD (section 4.2.4).

Two additional general arrays are formulated during the default problem definition, these are SRVLIF, containing values of equipment service life and the logical array CSTFIL, which indicates the components for which a cost model has been developed. The default values used in their formulation are read from the component initialisation subroutines and as such can only be redefined through editing and recompiling the appropriate component subroutine.

9.3.7 Cost Data Management.

The constants and coefficients of the capital cost models are held in the array SYSCST, each component in the system allowed up to two rows of data in the array. This format gives flexibility in model development as for example, one row may contain polynomial curve fit coefficients and the second may contain constants relating perhaps to a different element of the cost model. SYSCST is formulated when the initial feasible point is defined in the solution procedure, the cost data being read from structured data files. Capital cost values are computed in the component subroutines by accessing the data directly from SYSCST, thus reducing the need for repetitive access to data files. SYSCST is updated each time a new data file variable value is changed by the solution procedure.

Any number of data records can be held in the component cost data files which have been structured to accommodate new cost data by automatic expansion. Management of this data is performed through two subroutines <inpcst> and <delcst>. Subroutine <inpcst> allows new data to be stored in the cost data files. Data constants can be input from the keyboard and polynomial coefficients automatically transfered from the curve fitting procedure. Each new set of data is assigned a record name which matches the corresponding component performance data record name defined under SPATS. Subroutine <delcst> can be used to delete redundant cost data from the data file and in doing so automatically reduces the file size.

9.4 Objective Function Implementation.

An important requirement of the software structure is that additional objective functions can be implemented with the minimum of programming. The software structure developed in this research accommodates additional objective functions with no more programming than a subroutine to calculate the function value and simple program statements to ensure that the function can be identified and that the calculation routine is called in the appropriate section of the solution procedure. A further factor considered in developing the objective function evaluation software is the depedence of some objective functions on the format of the load profile.

The evaluation of the objective function values in relation to the constraint values and simulation of system performance is illustrated in figure 9.2. This algorithm has been structured to ensure only data required in evaluating the specified objective function is calculated. Values for net and primary energy consumption and energy costs are calculated for each time period and integrated over the load profile, the final function values held by the variable OBJECT. Where annual energy cost is a parameter in another objective function, OBJECT is used to pass the integrated cost into the specified objective function calculation subroutine. Maintenance costs are dependent upon the component duties and therefore are calculated using the duty at each time period and the maximum value over the profile retained for use as a parameter in the objective functions.

9.4.1 Energy Models and Cost.

Energy models and costs are formulated in two parts, integration of values over the load profile and summation of values for the components is performed in the solution point evaluation algorithm (figure 9.2). Values for individual components are evaluated in separate subroutines which interpret the energy model definition and add, subtract or exclude the individual energy terms accordingly. Figure 9.4 illustrates the model interpretation algorithm for the evaluation of net energy consumption of a component (subroutine <objeng>). Values for individual energy terms are calculated in the component subroutine and absolute values returned in the vector ENGFUN as MJ for the specified time interval.

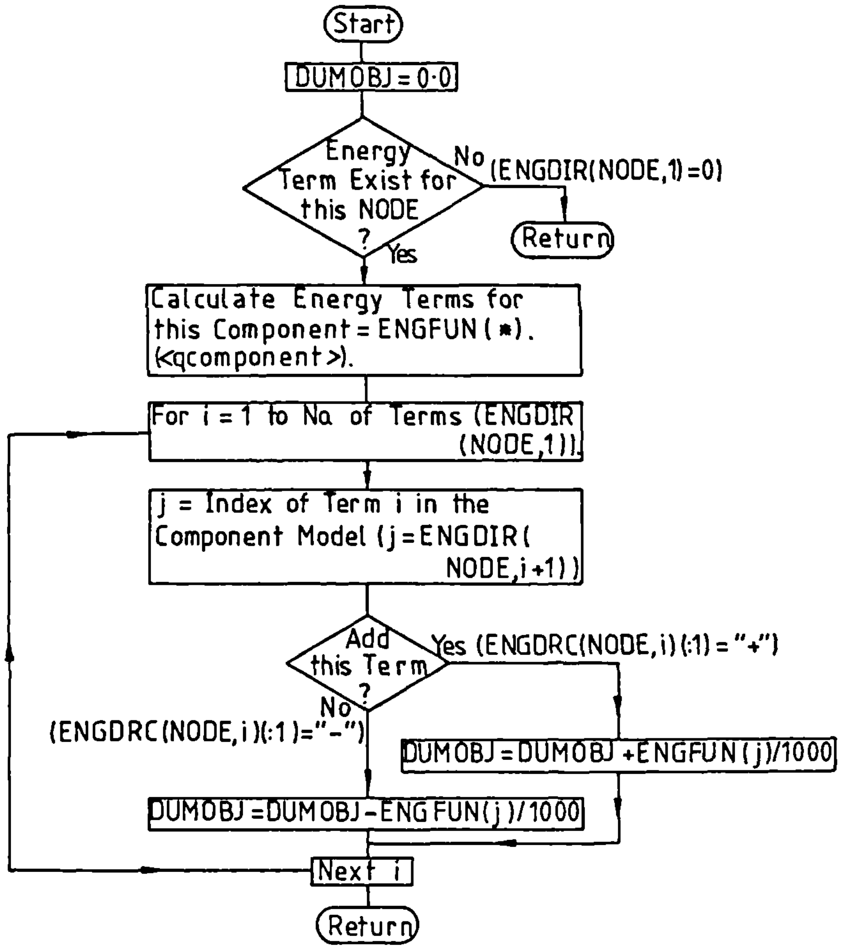


figure 9·4, Net Energy Model Algorithm.
(<objeng>)

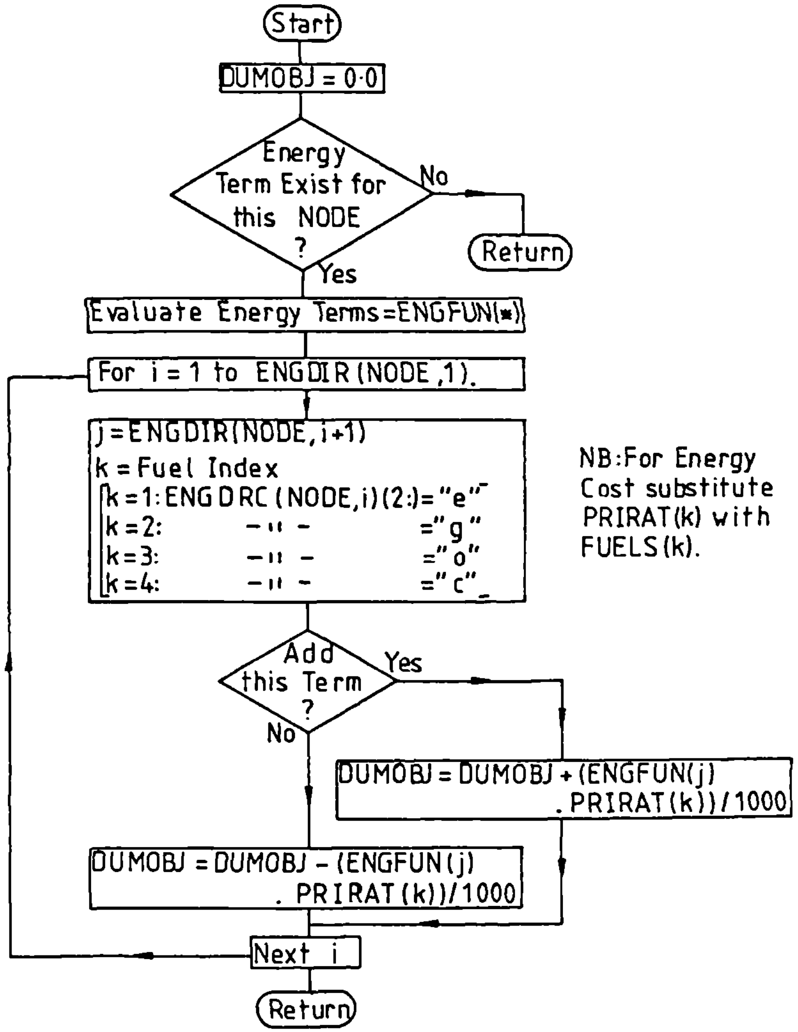


figure 9·5, Primary Energy and Energy Cost Algorithm.
(<objpri> & <engcst>).

Subroutine <objeng> then interprets the system energy model described in the arrays ENGDIR and ENGDRC and adds or subtracts the appropriate energy terms, converted to GJ, from the component overall energy parameter DUMOBJ. The final value of DUMOBJ is returned to the solution point evaluation subroutine for summation with other component values and integration over the load profile.

The algorithms for the primary energy model (subroutine <objpri>) and energy costs (subroutine <engcst>) differ only from that of the net energy model algorithm in that they have additional program statements which identify the appropriate primary energy ratio and fuel tariffs held in the vectors PRIRAT and FUELS (figure 9.5). The individual energy terms are multiplied by the appropriate value before their addition or subtraction in the model. Primary energy values are expressed as GJ and energy costs in thousands of pounds.

9.4.2 Capital and Operating Cost.

Operating cost is the algebraic sum of energy and maintenance costs, both of which are calculated over the load profile. the final value of energy cost assigned to the variable OBJECT and that of maintenance cost to MANMAX. Summation of the two is performed in the subroutine <objrun> and the operating cost overwritten to the objective function variable OBJECT.

The maintenance charge for the system at each time period is evaluated in the subroutine <mantnc>. Maintenance charges for each component are calculated in the component subroutines and summed for each component in the system. Similarly, the capital cost for each component is calculated in the component subroutines and summed for system cost in the subroutine <objcap>. Maintenance and capital costs are returned from the component subroutines by the vector COST, COST(1) the value of capital cost and COST(2) the maintenance charge. The variable CSTTYP can be used to economise on calculation time by specifying the calculation of either capital or maintenance cost in the component subroutine, CSTTYP=1 for capital cost only and CSTTYP=2 for maintenance cost only. Further economy is employed by ensuring that the component maintenance costs are only evaluated when appropriate to the system model (section 7.3.1). All costs including energy and operating cost are expressed in thousands of pounds.

9.4.3 Net Present Value and Discount Payback Period.

Two 'true' economic comparitors have been included as objective functions, the net present value of the system and the system discount payback period. Both techniques require present worth or discount factors in their formulation. Two subroutines have been implemented in this respect, <pwfsrs> which returns the series present worth factor for a given period and interest rate and <pwfsng> which returns the single value present worth factor for a specified year and interest rate.

The discount payback period calculation procedure described in chapter 7 has been implemented in the subroutine <objpay> (figure 9.6). This subroutine is called subsequent to the simulation/energy consumption calculation procedures and the energy cost passed into <objpay> by the objective function variable OBJECT. Similarly, the formulation of the net present value calculation described in chapter 7 has been implemented in the subroutine <objnpv> and includes an estimate of component replacement costs throughout the life of the building (figure 9.7).

9.5 Component Models.

Each component model is described by six subroutines each identified by the components generic name and a prescript defining the subroutine function:

<icomponent> - Initialisation subroutine.
<ecomponent> - Executive subroutine.
<rcomponent> - Results interpretation subroutine.

<qcomponent> - Energy term subroutine.
<ccomponent> - Cost term subroutine.
<bcomponent> - Constraint function subroutine.

For example, subroutine <icoil> is the heating/cooling coil component initialisation subroutine.

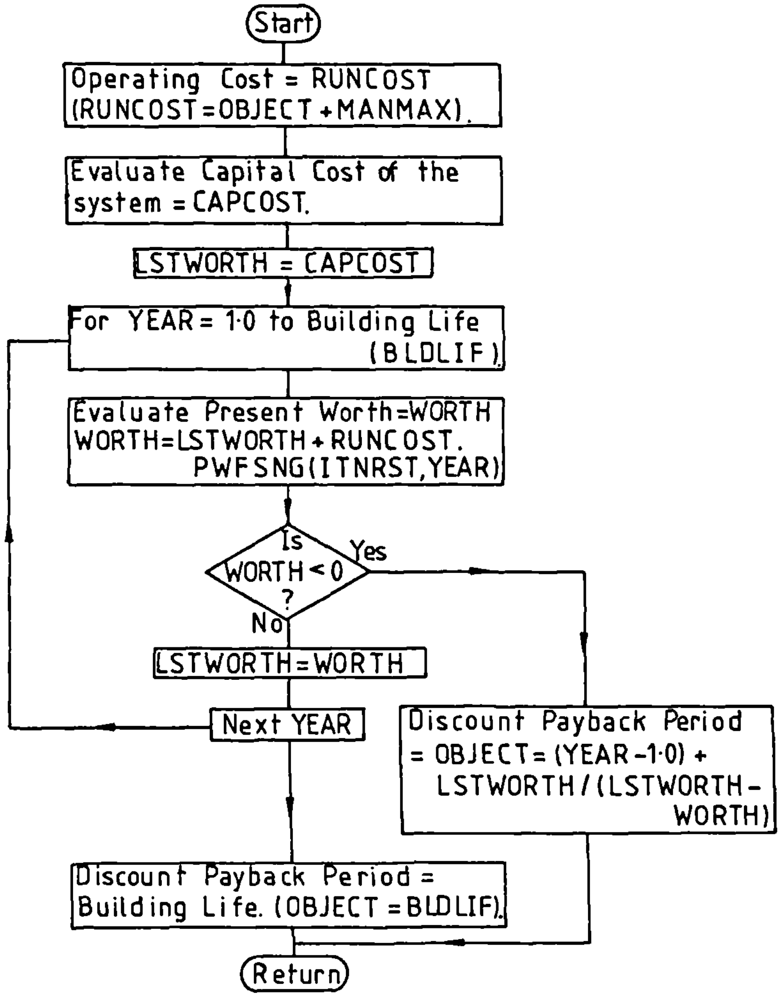


figure 9·6, Discount Payback Period Algorithm.
(<objpay>)

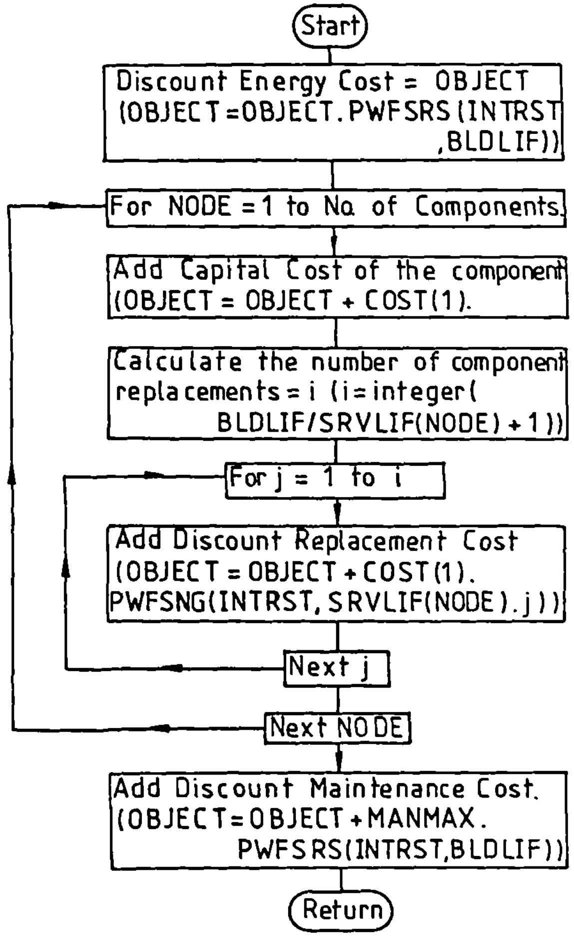


figure 9·7, Net Present Value Algorithm.
(<objnpv >)

The format of the first three subroutine in this list was developed as part of the simulation procedure SPATS, the final three only being specific to this research. The initialisation subroutine contains information regarding the identity of the component, ie: number of describing equations, number of polynomial curve fits, number of exogenous constants, variable names etc. The initialisation subroutine is called during the initial problem definition and if required, during subsequent redefinition. The parameters added to this subroutine which relate to the optimisation procedure are:

1. Name of the variable to be attached to the data file.
2. Number of constraint functions and their names.
3. Number of energy terms and their names.
4. A variable which identifies the existence of a component cost model (CSTFIL, appendix D.).

Component executive subroutines contain the describing equations of the component performance written in residual form. The executive subroutines return values of the residuals when called by the simulation solution algorithm. Executive subroutines evaluate the residuals at a particular solution point by using the network definition array NET to identify the arc-variables which are associated with the component. Once identified the variables can be passed into the equations and the residuals evaluated. The algorithm for identifying the component arc-variable values by interpreting the network definition also appears in the subroutines, <rcomponent>, <qcomponent>, <ccomponent> and <bcomponent> since each of these will contain equations whose values are dependent upon the arc-variable values.

Results subroutines convert the arc-variable values at a given solution point and present them in a form which is more recognisable to a practising engineer. For instance, the subroutine <rcoil> interprets the air pressure, temperature and moisture content at the coil inlet and outlet and presents them as the coil duty, sensible heat ratio and air pressure loss. The energy function component subroutines operate in a similar fashion to the results subroutines, but go one step further in multiplying the energy terms such as coil duty by the time interval in the load profile, TIMPRD, which converts values from power in KW to energy in MJ.

Parameters such as air pressure drop across the coil would also be converted to an energy term for use in sub-system design. Values of energy terms are returned by the vector ENGFUN for integration over the load profile.

Both the capital and maintenance costs are evaluated in the component subroutine <ccomponent>. Capital cost is a function of the component size and possibly its operating point. Therefore the component exogenous constant values and any relevant arc-variable values are derived from the network definition and used to evaluate the capital cost, by either locating the value in a table or from a polynomial curve fit. Data for the table and the polynomial coefficients are held in the array SYSCST. Maintenance costs are generally a function of an energy term, which is evaluated in the same manner as the terms in the energy function subroutine <qcomponent>. The annual maintenance charge can then be estimated by multiplying the energy term by a maintenance coefficient. Costs are expressed in thousands of pounds and are returned from the subroutine by the vector COST.

Constraint functions, evaluated in the component subroutines <bcomponent>, can be a function of the system operating point, size and configuration of the component. Consequently the network definition is interogated and the exogenous constant and arc-variable values related to the component, identified and used to evaluate the constraints. Constraint function values are returned to the optimisation procedure, for checking against their bounds, by the vector DUMCON.

9.6 Solution Algorithm Implementation.

The software structure has been developed to promote the implementation of several optimisation algorithms. Evaluation of objective and constraint functions can be controlled through a single subroutine <calobj>, which facilitates ease of access to these parameters. Both discrete and continuous variables are represented by the vector DESVAR which allows for compact programming, although in the current algorithm implementation, discrete and continuous variables are handled separately because during the early stages of research, maintaining individual identities assists in investigating the characteristics of the algorithm.

9.6.1 Solution Point Evaluation.

Two factors prompted the implementation of a procedure which controls the evaluation of the constraint and objective functions (subroutine <calobj>). The computation time required to solve the example design problems is in excess of the maximum processing time of three hours allocated by the Loughborough University computer centre (section 10.2.6). Therefore for optimum solutions to be obtained several 'runs' are necessary, the intermediate solution points from each run stored on a data file which allows the search to access these more rapidly on subsequent runs. Eventually, evaluation of only a few new search point is required enabling the search to be completed within the maximum computer time available. A second factor influencing the development of this subroutine is that the evaluation of the objective and constraint functions can be easily incorporated into each optimistaion subroutine by simply calling <calobj>.

The parameters stored on the data file by <calobj> are the problem variable values, the objective function type and value and a variable indicating the feasibility of the solution point. On calling the subroutine it searches the stored solution points: if a match of objective function type and problem variable values is found, the objective function value and feasibility parameter are passed back to the optimisation algorithm. If the point has not previously been evaluated <calobj> calls the specified simulation/solution point evaluation subroutine and stores the resulting values before passing them back to the optimisation subroutine. Data files containing the solution points are labelled with the system name appended with the characters .CAL.

9.6.2 The Exhaustive Search.

The exhaustive search algorithm described in chapter 8 has been implemented in subroutine <exhast>. Discrete values used to form points on the exhaustive search grid are held in the array STPVAR. Normally, only values for the true discrete variables are stored in the array, but use of the exhaustive search requires discrete values to be assigned to the continuous variables also. The most efficient approach would be to automate this process, the user specifying only the accuracy required for the continuous variables at the solution.

This approach has not been implemented in this research as the main use of this algorithm has been to investigate the validity of solutions obtained from other search methods and as such it is more appropriate for the user to specify the discrete interval.

9.6.3 The Pattern Search.

The structure of both the standard and improved constraint handling version of the pattern search are similar. The equivalent and common subroutines of these algorithms are:

Standard search:	Improved search:	
<pattrn>	<patrnm>	- Pattern moves and control.
<search>	<serchm>	- Exploratory moves.
<fndfes>	<fndfsm>	- Initial feasible point.

Common subroutines:

<stpset> - Initial probe length definition.
<savobj> - Save solution point temporarily.
<getobj> - Retrieve temporary solution point.

The main solution algorithms have been implemented in two subroutines: the first, <pattrn> (or <patrnm>) performs the pattern moves, assesses convergence of the solution and acts as overall control. The second element of the algorithm is the exploratory moves, these are performed in the separate subroutine of <search> (or <serchm>). Segregation of the pattern and exploratory procedures ensures coding does not become unwieldy and difficult to follow.

Evaluation of the initial feasible point is performed in the subroutine <fndfes> (or <fndfsm>). This at present is restricted to asking the user for a initial guess and checking its validity. The user may input the variable values or ask for a default guess which is based on the type of objective function. For example, a default guess for the capital cost objective function is the minimum value of the problem variables as invariably the cheapest components are the smallest. Further automation of this procedure will be possible as research progresses and constraint handling improves.

Once a feasible solution is established the initial probe step lengths are assigned by calling the subroutine <stpset>. The probe lengths assigned are the interval between values for discrete variables and a tenth of the range of values for continuous variables. These intervals are fixed but could be defined by the user in future implementations.

It is characteristic of the pattern search algorithm to repeat evaluations of previously searched points. Therefore during the early development of these algorithms two subroutines, <getobj> and <savobj> were implemented to initialise and access a temporary data base of previously searched solution points. The data base is held by a group of arrays, the maximum number of points held at one time equal to the maximum number of design variables manageable by the software.

9.7 Solution and Characteristics Analysis.

Several procedures have been implemented for the analysis of results and problem characteristics. SPATS has the ability to record the results from the simulation for each time period in the profile. These results are available as a table of arc-variable values and as the component performance evaluated by the component results subroutines. Similarly results from the optimisation are available through the ODESSY subroutine <prnsol>, which lists the optimum size of the components and system operating variable values node by node for all components in the system and includes a summary of the design parameters such as fuel costs, energy model terms and constraint values (table 9.2).

Progress of the solutions can be monitored for both the SPATS simulation and the ODESSY optimisation. Monitoring of the ODESSY solutions is via the subroutine <monsol>, which lists the variable values, objective function value and constraint violations at each of the trial points. Graphics representation of the objective functions is available as a two dimensional surface plot or as a function of a single variable. The subroutine <optplt> provides general graphics control whilst the subroutines <surplt> and <graf> execute NAG graphics routines for surface and graph plots (NAG).

```
***********************************************************************
OPTIMAL COMPONENT SELECTION.

System name:pool-run

-----------------------------------------------------------------------
Node: 1 Component:axialfan
                        Solution.
Design variables:
Variable: 1 :fan-dia = 0.9000D+02 lb: 0.9000D+02 ub: 0.1120D+03
Variable: 2 :speed   = 0.1470D+04 lb: 0.1470D+04 ub: 0.1470D+04

Service life=20.0 years.
-----------------------------------------------------------------------

-----------------------------------------------------------------------
Node: 5 Component:h/c-coil

Design variables:
Variable: 6 :width    = 0.2000D+01 lb: 0.1000D+01 ub: 0.2000D+01
Variable: 7 :height   = 0.1250D+01 lb: 0.1000D+01 ub: 0.2500D+01
Variable:13 :no. rows= 0.2000D+01 lb: 0.2000D+01 ub: 0.1000D+02
Variable:14 :wat-circ= 0.3000D+02 lb: 0.1000D+02 ub: 0.3000D+02

Constraints:
Constraint: 7 :facevel = 0.2432D+01 lb: 0.0000D+00 ub: 0.2500D+01
Constraint: 8 :watervel= 0.3529D-01 lb: 0.0000D+00 ub: 0.1800D+01
Constraint: 9 :circuits= 0.4082D+00 lb: 0.0000D+00 ub: 0.1000D+01

Energy terms:
Energy term:duty     Energy type:      o Objective term:      +
Energy term:airloss  Energy type:      e Objective term:      +

Service life=20.0 years.

This node is included in pay back calcs.
-----------------------------------------------------------------------

-----------------------------------------------------------------------
Number of time periods=  1
Time per period= 4320.0000 hours
Interest rate: 10.0
Building life:30.0
Fuel prices (Pence/MJ):
Electricity (peak):1.06
Gas              :0.31
Oil (35 sec.)    :0.51
Coal             :0.22
-----------------------------------------------------------------------

Objective function:pay back =    1.2622
***********************************************************************
```

table 9·2, Example Table of Results.

The subroutine <simnet>, accessed through the ODESSY sub-menu command 'sim', enables the simulation to be run from within ODESSY for a specified optimisation solution point. This is useful in determining whether failed simulation solutions are due to instability of the solution procedure or caused by component undersizing.

9.8 Application Methodology and Future Development.

A modular software structure lends itself to an investigation of the characteristics of individual elements of the design problem. Yet, the extreme modularity of the current research software can, to the uninitiated, lead to confusion and difficulty in problem solving. A description of the application methodolgy not only makes clear the use of individual program elements, but more importantly can indicate areas of program development which once implemented will provide a more efficient and 'user friendly' design tool.

The generalised application methodology illustrated in figure 9.8 assumes that a comprehensive data base of components and associated product ranges are available. Although shown as a continuous process the methodology can be divided into three procedures: the problem definition, operations (1) to (5), establishing the feasibility of an initial estimate to the solution, operation (6) and finally the optimisation including an assessment of the validity of the solution, operations (7) to (10).

9.8.1 Problem Definition.

A detailed description of the elements of the problem definition, represented by operation (1) to (5) of figure 9.8, is given earlier in this chapter. Improvements to the efficiency and usability of this software is largely development work requiring little research input. The most obvious improvement is to provide access to all problem definition elements through the same commnd menu, instead of two split between SPATS and ODESSY.

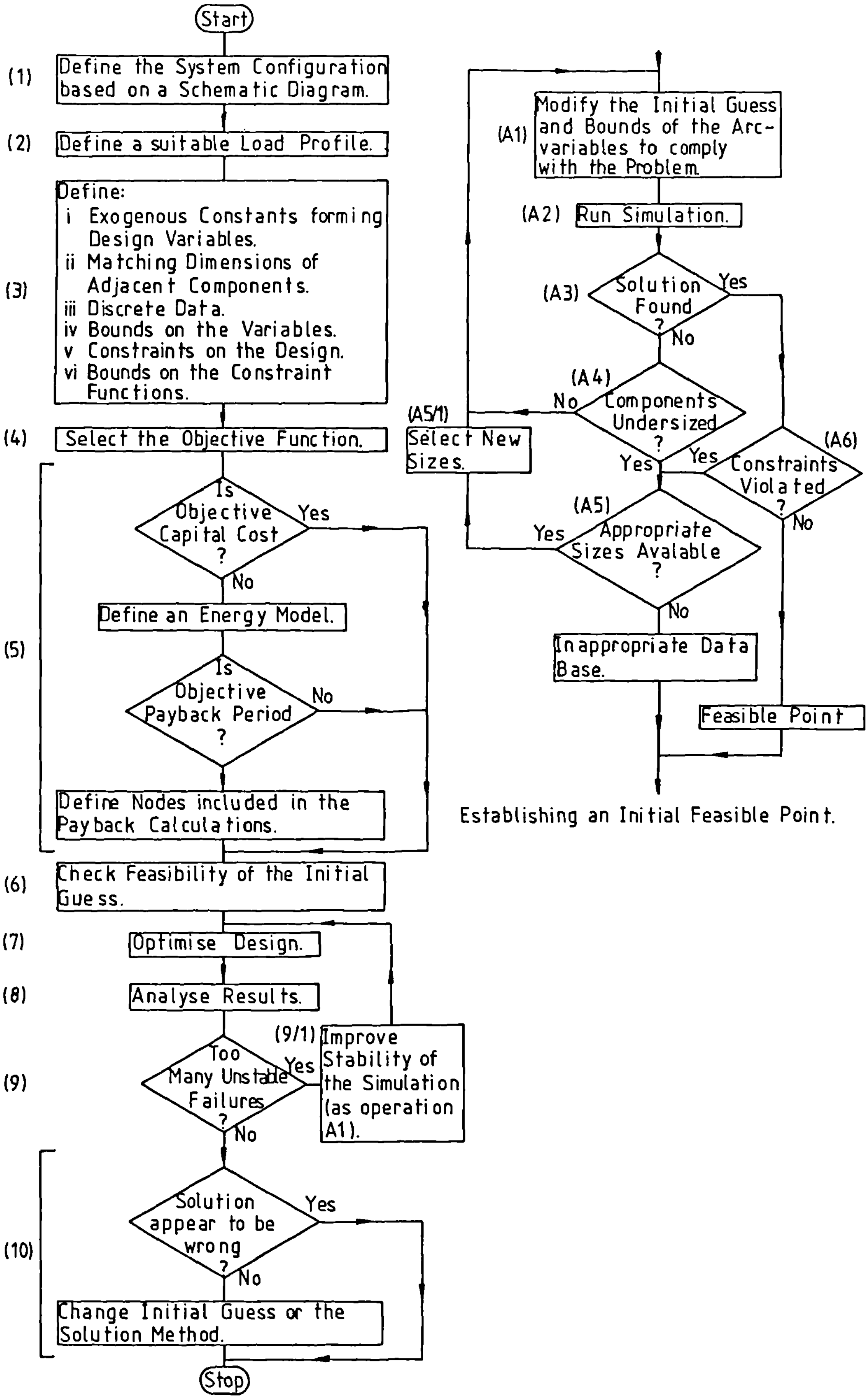


figure 9·8, Optimised Design Software Application Methodology.

Many features of the problem definition will become more automated with future development. Interfacing the procedure with graphics software will enable the system configuration to be defined more rapidly, the user graphically specifying the position of components within the system. The matching dimensions of adjacent components could also be automated as part of this process. Constraints could be assigned to a particular problem based on a standard specification and product ranges could be selected on an estimate of the plant loads. A user of the current software has to remember to define an energy model for those objective functions which require one and to define the components included in the payback period calculations (operation (5)). Obviously this operation can be improved if only by reminding the user to define these elements.

9.8.2 Establishing an Initial Feasible Point.

Establishing an initial feasible solution poses two problems, ensuring that the simulation procedure is robust and checking that the initial solution satisfies all the constraints. As in the optimisation procedure, the simulation procedure must be provided with initial values and bounds for the system variables. Both the Newton-Raphson and GRG2 solution procedures can be very sensitive to these parameters with what appears to be only small changes in value leading to instability and failure. The 'trial and error' process which is used to establish initial values of these variables is represented by operations (A1) to (A4) of figure 9.8. If the simulation fails an examination of the results can indicate whether the components of the initial guess are undersized and if so an appropriate size can be specified and the robustness of the simulation re-checked. For example, if on failure of the simulation the results indicated that a component was operating at full capacity the component could possibly be undersized. If however all components appear to be adequately sized the user must persist in changing system variable values and bounds until the simulation finds a solution.

Once a stable simulation is established, the initial solution of the optimised design problem must be checked against the constraints and if found to be infeasible the component sizes and operating variables re-selected. Automation of this process requires extensive research, (section 8.6) and is essential if the software is to be used in a design environment.

9.8.3 Design Optimisation.

The instability of the simulation solution algorithms and tendancy for the pattern search to fail when constraints are encountered, limits the reliability of design solutions. Results must be analysed to ensure that any unstable failures of the simulation algorithm which occured during the search for the optimum, have not led to a false solution. Solutions which lie on or close to constraints should be checked to ensure that their occurrence has not led to failure of the pattern search (operations (7) to (10), figure 9.8). Future development of the solution algorithms will dispense with these manual checks and allow the software to be used in a real design environment.

9.9 Validation of the Optimised Design Software.

The suite of programs implemented in this research have two distinct elements, the plant simulation procedure and the optimisation procedure. Future implementation is likely to include integration of these with a third element, a building performance simulation procedure. Although each of these procedures can be validated individually, it is important on integration of the software to ensure that the interfacing parameters are such that the integrity of the software is maintained. For example, when integrating the simulation and optimisation procedures it is important to ensure that results from the simulation can be used to formulate realistic and robust component undersizing constraints.

An important validation exercise for building thermal models is being conducted by Bowman et al. (1986) and aims to produce a package of tests to validate dynamic thermal models. It is invisaged that the package will contain: the data necessary to implement the tests, guidance on any model modifications necessary to implement the tests, a list of algorithms exercised by the tests and the answers which the models should provide with a statement about the accuracy to be expected. In developing the package, Bowman identifies three approaches to validation, analytical verification, intermodel comparisons and empirical validation. Of these the most important is empirical validation, the comparison of predicted and measured building response. Unfortunately this approach is plagued by all the problems of experimentation, not least of which in this case are the problems of measuring and modelling the effect occupance have on the building operation. Conclusions from this work are wide ranging and suggest a series of tests should be devised which proceed sequentially from simple to more complex situations, although for validation against larger buildings this is likely to be restricted to the more measurable and quantifiable parameters.

The most useful exercise for the validation of the system simulation procedures is that conducted under the auspice of the Annex 10 of the International Energy Agency. The exercise aim is to validate and compare various simulation procedures and their component models by each simulating the performance of a Variable Air Volume system in the Collins Building, Glasgow. Part of this exercise is described by Murray (1984) in the application of the SPATS simulation procedure. Validation of component based procedures such as SPATS, is largely concern with the verification of individual component models. Component models developed from manufacturers data can be validated by comparing the predicted performance against measured performance, although care should be taken to ensure that the conditions under which the performance was measured are representative of the installation modelled in the comparison. Component models developed from first principles are somewhat more difficult to validate and require instigation of extensive empirical validation procedures.

Validation of the optimisation procedure can, as for component based simulation procedures, be conducted largely by the validation of the individual elements. Once a robust solution procedure is established, validity of the optimum solutions will depend upon the validity of component models, accuracy to which design criteria such as fuel prices and interest rates have been modelled and to what extent the constraint functions represent the true design and operating limits of the system. Absolute validity of the optimum solutions will, due to the stochastic nature of some design parameters, always be in doubt. Fluctuations in the economic parameters and climatic conditions are impossible to predict to a high level of accuracy. This therefore suggests that the optimum solutions will never be finite and should be supported with the risk of being wrong. The risk should be evaluated automatically as part of the optimisation procedure and as such would provide a measure of the validity of each solution. The simplest probabilistic procedures to implement initially are those of sensitivity and risk analysis, the use of which in improving design reliabilty is discussed in section 10.3.2.

Chapter 10. THE APPLICATION OF OPTIMISED DESIGN SOFTWARE.

Future optimisation software will be integrated with other design and draughting packages to form a user friendly software tool capable of integrated building design. The existing suite has its main application in the optimised design of small sub-systems and has proved useful as a tool for model development. Continuing development of the software will lead to improvements in the design procedure and therefore reliability of design solutions. In particular, design reliability will be improved by analysing the sensitivity of solutions to changes in design criteria such as interest rates, fuel costs and climatic conditions.

10.1 Component Model Development.

The successful application of the software relies upon a comprehensive data base of component models: consequently included in the suite are programs which assist the development of such models. A comprehensive method of least squares curve fitting (appendix A) has been developed to enable the curve fitting of performance and cost data. Polynomial coefficients obtained from the curve fit are automatically transfered into designated component performance and cost data files for use with the component algorithms.

Development of the component energy models relies upon the development of the performance models. Murray (1984) included in SPATS a facility to compare various component performance algorithms through the use of 'test nodes'. These allow the comparison of algorithms for the components operating separately or as part of a system.

As the presentation of cost data varies in format between manufacturers, cost model development is restricted to the formulation of an algorithm which can be used with the majority of component. No special facilities have been developed in this respect, but future software development could include an aid to the comparison of the error between the original manufacturers data and output from the model.

10.2 A Heat Recovery Application.

It is beyond the scope of this thesis to demonstrate all applications of the software, the most that can be achieved is to illustrate the typical characteristics of an optimised design problem and its solutions. The system configurations chosen as examples are simplistic to ensure that the characteristic of the problem are easily understood and are not obscured by complicated relationships.

Two systems have been chosen in relation to heat recovery in swimming pools, a run-around coil system and a package chiller heat recovery system. Both systems recover heat from the high temperature exhaust air of the pool hall and transfer it to the colder fresh air. In practice this type of system would undoubtedly benifit from computer analysis and design as the choice of heat recovery systems is limitless and the sizing of items of plant can be critical if their installation is to be justified.

Both examples exhibit many of the characteristics of an optimisation problem and have been used extensively in the development of an optimisation algorithm (chapter 8). The characteristics of the example problems analysed here are aimed at providing an insight to the advantages of using optimised design software.

10.2.1 Design Conditions and Parameters.

The design conditions for swimming pools are well established. Ventilation rates are maintained to prevent condensation occuring on the coldest surfaces of the pool hall, the level of humidity at which this occurs typically ranges from 75% in summer to 55% in winter. A recommended volume flow rate which prevents condensation under the worst conditions throughout the year is 0.015 m^3/s. per square metre of wetted area, (Burgess, 1982). Taking an example wetted area of 400 m^2 this gives a volume flow rate of 6.0 m^3/s (approximately 7.2 kg/s). Pool hall air temperatures are maintained at 28 ^oC, which ignoring fabric losses, has been taken as the supply air temperature, ± 0.25 ^oC allowed for a variation under control.

It is assumed that the plant would operate for 12 hours per day, 360 days per year and that during the remaining 12 hour period each day, that the pool would be covered to reduce heat loss.

Correct definition of a load profile (external weather data), is crucial if a meaningful analysis of the system energy consumption is to be performed and if the probability of the plant failing to operate correctly once installed is to be minimised. Use of a realistic load profile in the examples is limited by the availability of computer processing time. Solutions for the examples require a high number of calls to the GRG2 simulation solution algorithm which with its long calculation time has proved prohibitive. Average values for external temperature of 10.5 ^{o}C and relative humidity of 78%, have been selected to give a reasonable representation of the system energy consumption, but obviously as extreme conditions are not modelled the reliability of the system to perform within its design conditions would in reality be doubtful.

Apart from the correct sizing of components, the additional constraints imposed on each design relate to the heating/cooling coils and are that to reduce the risk of noise problems and moisture carryover, the coil face velocity should not exceed 2.5 m/s. Secondly to prevent erosion of the pipe work the water velocity per circuit in each coil should not exceed 1.8 m/s and finally a configuration constraint to ensure that sufficient tubes are available to form the required number of water circuits. These and the other design parameters are summarised in tables 10.1 (a) to (d).

Designs have been optimised for all available objective functions: that is system net energy consumption, primary energy consumption, operating cost, net present value and payback period.

Parameter	Climatic Conditions	Supply Air.	Exhaust Air.
Temperature. (°C)	10·5	28 ± 0·25	28
Moisture Content. (Kg/Kg)	0·0062	0·0062	0·0132
Air Mass Flow Rate. (Kg/s)	-	7·2	7·2

table 10·1 (a), Design Conditions.

Fuel Type.	Primary Energy Ratio. (BRE, 1976)	Fuel Tariff. (Pence/MJ-Gross) (NIFES, 1985)
Electricity (Peak)	3·82	1·06
Fuel Oil (35 second)	1·09	0·51

table 10·1 (b), Fuel Tariffs and Primary Energy Ratios.

System Operating Hours per Annum.	4320 Hours
Interest Charged on Borrowed Capital.	10 %
Life of the Building.	30 Years.
Service Life of: Fans Coils and Chillers. (ASHRAE, 1984).	20 Years.

table 10·1 (c), General Design Data.

0 ≤	Coil Face Velocity (m/s).	≤	2·5
0 ≤	Water Velocity per Circuit (m/s).	≤	1·8
0 ≤	(Circuits - Tubes) / (1 - Tubes).	≤	1

table 10·1 (d), Design Constraints.

tables 10·1, Design Conditions and Constraints.

10.2.2. The Run-around Coil System.

Figure 10.1 is a schematic diagram of the run-around coil system which denotes the system variables and indicates that for simplicity the hydrodynamic characteristics of the system are not modelled. An uncontrolled run-around coil recovers waste heat from the exhaust air of the pool hall and transfers it to the colder fresh air. The temperature of the supply air is maintained by an additional heating coil which is proportionally controlled by the action of a three port diverting valve. Ventilation is provided by axial flow fans running at a speed of 1470 rpm.

The discrete problem variables which represent the size of the components are the diameters of the axial flow fans and the width, height, number of rows and water circuits of the coils. A single problem variable assigned to each of the matching dimensions of the adjacent heating coils, ensures that in practice the solutions would allow the components to be connected. Fluid variables defined as continuous problem variables are the water mass flow rate of the run-around coils and the maximum water mass flow rate available to the additional heating coil (exogenous variable 12). These have been chosen as problem variables as they are the fluid variables most likely to affect the optimum solutions.

In a more sophisticated model, rather than assign arbitrary ventilation rates, the optimised design approach would be to extend the system to include a condensation model of the pool hall, define the air mass flow rates as continuous variables and assign a constraint which specified that no condensation should occur. This would allow optimum values of air mass flow rates to be found whilst ensuring that condensation did not occur. The problem variables and their associated product ranges are summarised in table 10.2.

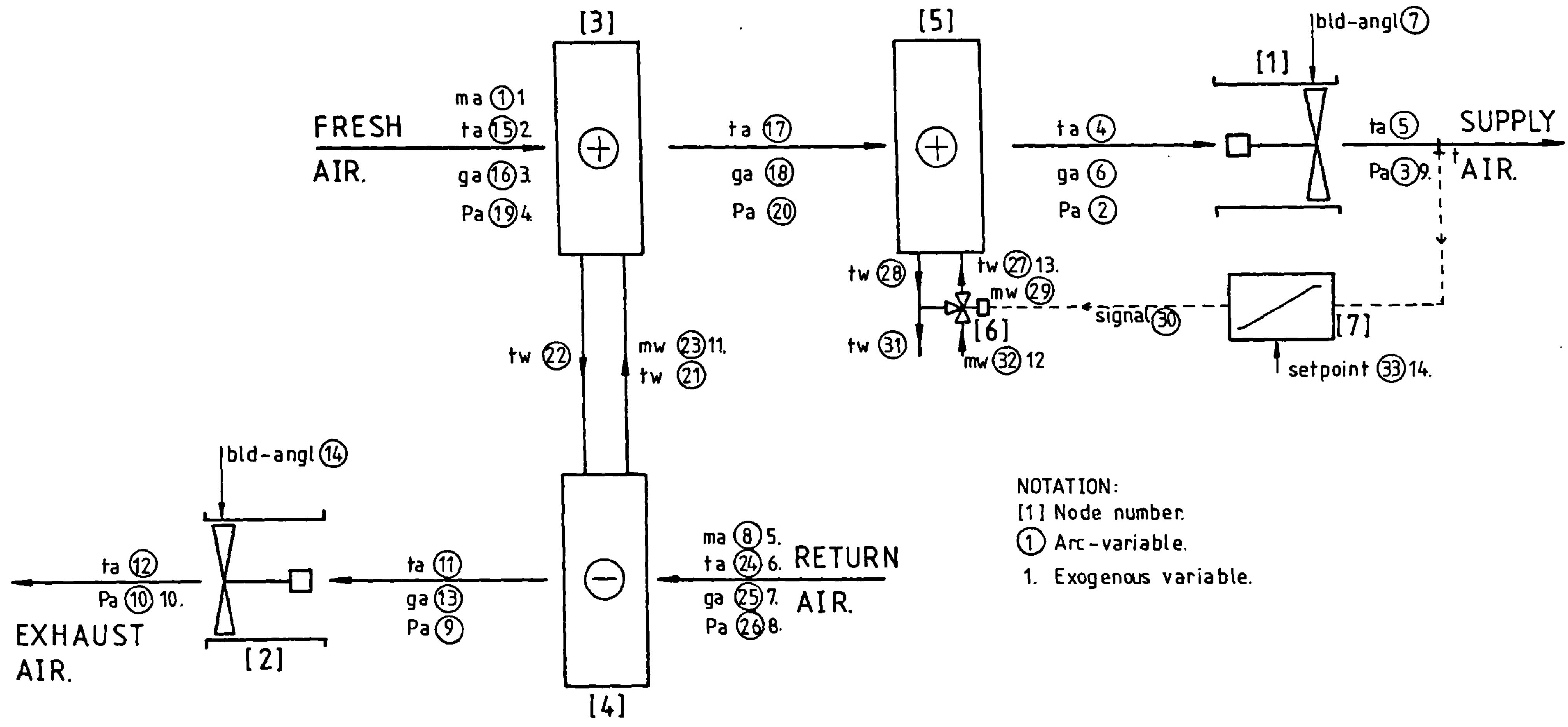


figure 10·1, Run-around Coil System.

Fans.	Discrete Data.				
Supply - Diameter (m).	0·9	1·0+	1·12		
Extract - Diameter (m).	0·9	1·0+	1·12		
Run-around Coils.	Discrete Data.				
Supply - No. Rows.	2	4	6+	8	10
- Width (m). *	1·0	1·25	1·5	1·75+	2·0
- Height (m). **	1·0	1·25	1·5	1·75+	2·0
- Water Circuits.	10	20+	30		
Extract - No. Rows.	2	4	6+	8	10
Width (m).	1·0	1·25	1·5	1·75+	2·0
Height (m).	1·0	1·25	1·5	1·75+	2·0
Water Circuits.	10	20+	30		
Heating Coil	Discrete Data.				
Supply - No. Rows.	2	4	6+	8	10
- Width (m). *	1·0	1·25	1·5	1·75+	2·0
- Height (m). **	1·0	1·25	1·5	1·75+	2·0
- Water Circuits.	10	20+	30		
Water Mass Flow Rates.	Variable Bounds.				
Run-around Coils (Kg/s)	2·0 —— 6·0 (4)+				
Heating Coil (Kg/s)	2·0 —— 6·0 (4)+				

*, ** Matching Dimensions.
\+ Initial Guess.

table 10·2, Run-around Coil System Problem Variables and Product Ranges.

An energy model which represents the total energy consumption of the system is defined in table 10.3 and assumes that additional heat is supplied to the heating coil from an oil fired boiler. This energy model is used with all objective functions requiring energy modelling except the payback period calculations as these have been restricted to the plant concerned with supplying heat to the pool hall, ie:the run-around coils and supply heating coil. Supply and extract fans are excluded from the payback period objective function calculation but are included in the system simulation to ensure the correct sizing of components. The energy model defined for use with the payback period calculations (table 10.3) indicates that excluding fans from the energy model leads to the inclusion of the coil model energy terms which relate to the fan performance, ie: the energy lost due to the air pressure drop across the coils.

10.2.3 Solutions for the Run-around Coil System.

It is difficult in a manual design process to consider the complex relationships between components when they are sized on an individual basis. Often the best that is achieved is to consider the general characteristics of the components. For example, the larger sizes of the fan within a given range are the most efficient. The larger the run-around coils the more heat they will recover and with large face areas the air pressure loss and hence fan power is reduced. Conversely it is invariably the smaller components which prove the cheapest. The advantage of optimising the size of the components simultaneously as a system is that the operating relationships between the components are considered during the optimisation which leads to a better combination of component sizes being selected.

Results for the optimised design of the run-around coil system, obtained from the constrained pattern search, are summarised in table 10.4. Although some of these are not optimum solutions (chapter 8), they are sufficiently close to the optimum to exhibit the characteristics of the true solutions.

Model.	Supply Fan.	Extract Fan.	Run-around Coils.				Heating Coil.	
			Supply.		Extract.			
	Impeller Power.	Impeller Power.	Duty.	Air Loss.	Duty.	Air Loss.	Duty.	Air Loss.
Net Energy Model.	e +	e +	/	/	/	/	o +	/
Payback Period Model.	/	/	o −	e +	/	e +	o +	e +

e: Fuel Type = Electricity.
o: Fuel Type = Oil.
±: Add or subtract value in the model. (/ Term excluded).

table 10·3, Run-around Coil System Energy Models.

Objective Function.	Fan Diameter.		Run-around Coil, Supply side.				Run-around Coil, Exhaust side.				Heating Coil.		MassFlow Rate.		Objective Function Value.
	Supply	Extract.	Rows.	Width *	Height *	Circuits	Rows	Width	Height	Circuits.	Rows	Circuits	Run-around	Heat'g Coil.	
Net Energy	0·9	1·12	10	2·0	2·0	10	10	2·0	2·0	10	10	30	2·0	2·0	680·5 GJ/annum.
Primary Energy.	1·12	1·12	10	2·0	2·0	10	10	2·0	2·0	10	2	30	2·0	2·0	1,183GJ/annum.
Capital Cost	0·9	0·9	2	1·5	1·75	20	2	1·5	1·75	20	2	20	4·0	4·0	7,419pounds.
Operating Cost.	1·12	1·12	10	2·0	2·0	10	10	2·0	2·0	10	2	30	2·0	2·0	4,722 pounds/annum.
N.P.V.	1·12	1·12	10	2·0	2·0	10	10	2·0	2·0	10	2	30	2·0	2·0	57,196pounds.
Payback Period.	0·9	1·0	10	2·0	1·25	10	10	2·0	1·5	10	2	30	2·0	2·0	1·3 years.

* Dimensions match those of the heating coil.
Dimensional units as table. 10·2

table 10·4, Run-around Coil System Results.

The solution for minimum net energy consumption indicates several anomalies: the smallest supply fan and a heating coil of maximum rows have been selected where it would have been expected for the solution to tend to the opposite extreme of the product ranges. Closer inspection reveals that although a smaller supply fan is less efficient than a larger fan, the reduced efficiency results in a higher temperature rise across the fan which reduces the load on the heating coil. Likewise a larger number of heating coil rows increases the fan power and therefore the temperature rise across it. This effect is clearly artificial as the net energy consumption calculation takes no account of the fuels being used by the system. The primary energy consumption calculation is more realistic in this respect as when primary energy efficiency is considered the increase in fan power outweighs any decrease in heating coil load and results in a larger more efficient supply fan and a smaller heating coil with a lower air pressure loss.

The solution for operating cost is identical to that for primary energy consumption confirming that primary energy modelling is a more realistic indication of energy usage and cost than net energy modelling. An anomaly which appears in the operating cost and most other solutions, is that the lowest value of water mass flow rate in the run-around coils has been selected, where it might be expected that a higher mass flow rate would lead to a reduction in the thermal resistance and greater heat recovery. In this case the lower mass flow rate produces a higher water temperature difference with greater heat recovery. Extending the lower limit of the water mass flow rate might produce an 'optimum' value at which point a further reduction in mass flow increased the thermal resistance to an extent that less heat would be recovered. For the majority of objective functions the maximum water mass flow rate to the additional heating coil has a solution value which is on the the lower bound of the variable. This occurs as the controller throttling range effectively allows a range of heating duties. The optimum solution for an energy related objective function will tend towards a mass flow rate which produces the lower coil duty. In the case of the operating cost objective function the supply air temperature at the solution is near the upper limit of the throttling range, suggesting that a lower mass flow rate would be found if its lower bound was extended.

For all objective functions except capital cost, the solutions for the coil water circuits have been influenced by the system energy consumption. The water circuits of the run-around coils have been kept as low as possible which maintains the water velocity per circuit, thus reducing the thermal resistance. Conversely, the coil circuits of the additional heating coil have been increased to a maximum, thus lowering the coil output and energy consumption.

The capital cost objective function is of particular importance in developing an optimisation algorithm since certain problem variables do not directly affect capital cost and remain unchanged from their initial values, unless the solution lies near a constraint function which is affected by the inactive variables. For example, the water mass flow rate to the heating coil remained unchanged by the optimisation as it does not directly affect the capital cost of the coil. Yet the search was only able to select a small number of coil rows because the maximum water mass flow rate was high enough to maintain the required heat transfer with. Had this not been the case, as a two row coil proved to be undersized, the water mass flow rate would indirectly influence the capital cost of the system as its value would be increased to enable a smaller coil to be selected. Those variables which only indirectly affect capital cost and whose values have remained at the initial guess to the solution are: the water mass flow rates and water circuits of all the coils. The width and height of the coils in the capital cost solution have tended towards their lower bounds as the smaller the coil the cheaper it is. Yet they have been prevented from obtaining the smallest values of 1.0m as a further reduction in size from the solution, results in a violation of the face velocity constraint. The limiting factor here in terms of the size of coil can be is approximated by its face area and therefore for a given face area several solutions of varying configuration lie on the constraint ie: a wide short coil or a tall thin coil. The solutions obtained for the capital cost function are erroneous as the configuration with the lowest capital cost has a larger width and smaller height (figure 8.9). Capital cost solutions for the number of coil rows and fan diameters lie on their lower bounds suggesting that if these were extended smaller components would be selected, the limiting factor being that all components meet the required duties.

It might be expected that the solutions obtained for the net present value objective function would be influenced by the combination of capital and operating costs. However the predominant factor here has proved to be the operating cost with the same size of components selected for the net present value objective function as for operating cost. Conversely, the payback period solution reflects that of both the operating and capital cost solutions. As might be expected the water mass flow rates and coil circuits selected for minimum payback period, have the same values as for operating costs solutions as these are the variables which have no direct affect on capital cost. The width and height of each coil is influenced more by the capital cost of the component, yet the coil rows have been selected for maximum heat recovery and minimum operating cost. Although the fans do not appear in the formulation of the payback period objective function, the smallest size of suppply fan has been selected as the efficiency of this fan indirectly affects the energy consumption of the system. The size of the extract fan has remained at the initial guess to the solution as this fan has no interaction with the other components in the system, apart from ensuring that it is large enough to meet its operating load.

10.2.4 The Package Chiller Heat Recovery System.

Much of the application methodology and solution characteristics described for the run-around coil system also apply to the package chiller system. To avoid repetition, description of the package chiller problem and its solution is limited to the additional characteristics associated with the system. As for the run-around coil system the hydrodynamic characteristics of the package chiller heat recovery system have not been modelled.

Figure 10.2 is a schematic diagram of the package chiller heat recovery system and its associated system variables. The temperature of the supply air is maintained by a heating coil connected to a package chiller which recovers low grade heat from the exhaust air of the pool hall via a cooling coil. Proportional control of the heating coil is achieved by the action of a three port diverting valve. The chiller compressor is of the centrifugal type which enables continuous proportional control. Ventilation is provided by axial flow fans running at 1470 rpm.

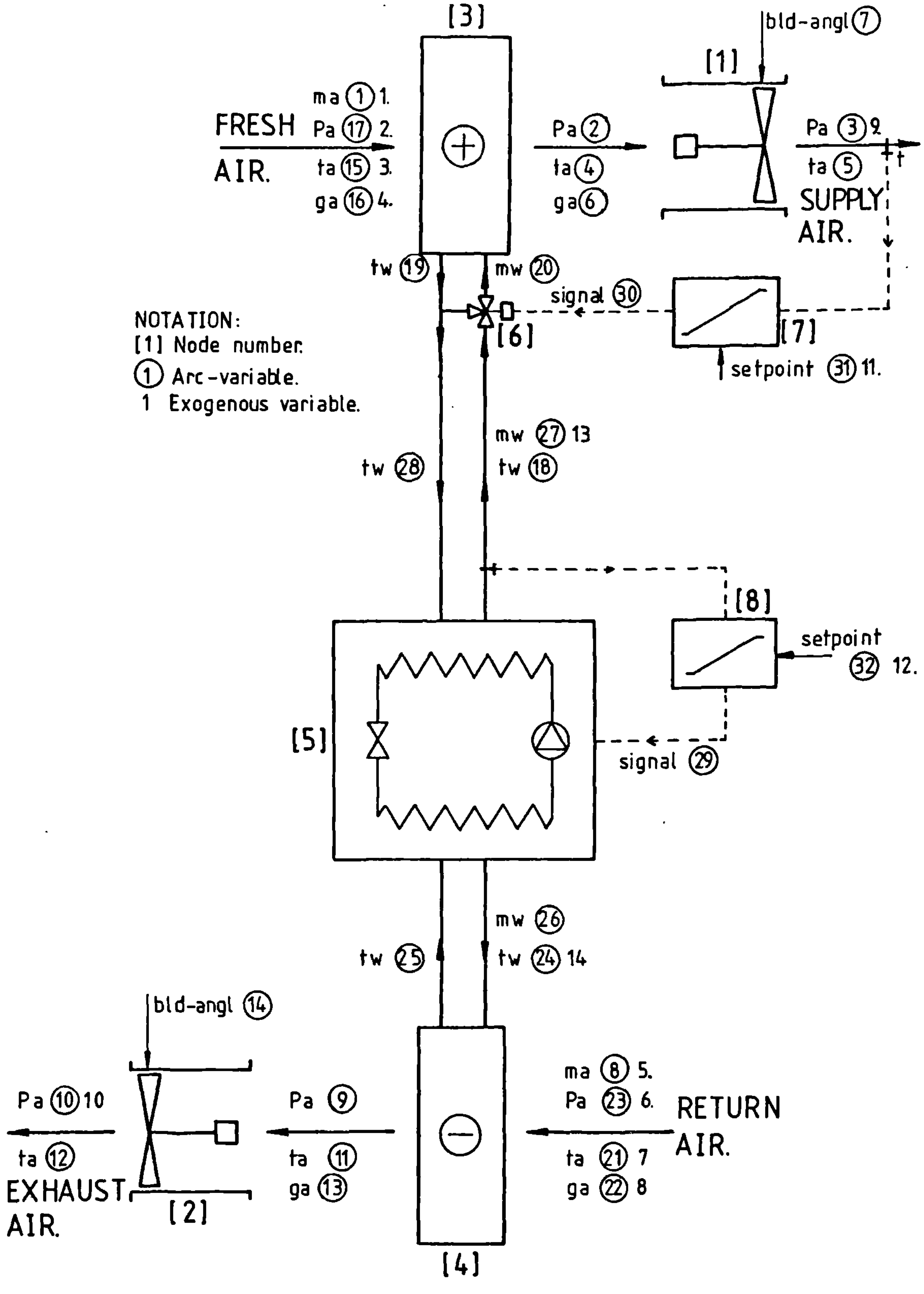


figure 10·2, Package Chiller Heat Recovery System.

The discrete problem variables which represent the size of the components are the diameters of the supply and extract fans, width, height, number of rows and water circuits of the coils and a variable related to the chiller catalogue number. The fluid variables defined as continuous problem variables are the condenser water mass flow rate, the flow temperature of the condenser water and the set point of the chiller proportional controller. The package chiller problem variables and their associated product ranges are summarised in table 10.5.

An energy model for use with all energy related objective functions except the payback period is given in table (10.6). As for the run-around coil system, the payback period calculations have been restricted to the plant concern with the direct supply of heat to the pool hall, ie: the chiller and coils. The payback period energy model (table 10.6) illustrates the advantages of implementing a flexible method of definition as this allows the heat input to be related to an alternative method of heat supply to that of the chiller, ie: it is assumed that the chiller heat recovery system is an alternative scheme to that of supplying heat from a conventional oil fired boiler and therefore that the true cost saving is in terms of the fuel that would be supplied to the boiler in the abscence of the chiller system. This also ensures compatibility the chiller and run-around coil system models since the additional heat to that recovered by the run-around coils is supplied by an oil fired boiler. The inefficiency of the chiller is represented in the payback period energy model by offsetting the heat supplied with the chiller compresser power.

10.2.5 Solutions for the Package Chiller System.

The optimised design solutions obtained using the constrained pattern search are summarised in table 10.7 and as for the run-around coil system, although some results are erroneous they are of sufficient accuracy to exhibit the characteristics of the true solutions.

Fans.	Discrete Data.
Supply - Diameter (m).	0·9 1·0+ 1·12
Extract - Diameter (m).	0·9 1·0+ 1·12
Run-around Coils.	**Discrete Data.**
Supply - No. Rows.	2 4 6+ 8 10
- Width (m).	1·0 1·25 1·5 1·75+ 2·0
- Height (m).	1·0 1·25 1·5 1·75+ 2·0
- Water Circuits.	10 20+ 30
Extract - No. Rows.	2 4 6+ 8 10
- Width (m).	1·0 1·25 1·5 1·75+ 2·0
- Height (m).	1·0 1·25 1·5 1·75+ 2·0
- Water Circuits.	10 20+ 30
Package Chiller.	**Discrete Data.**
Catalogue Number.	50 60 70+ 80 90
Fluid Variables.	**Variable Bounds.**
Condenser Water Mass Flow Rate (Kg/s).	2·0 —— 6·0 (4·0)+
Evaporator Water Flow Temperature. (°C).	5·0 —— 9·0 (7·0)+
Condenser Water Flow Temperature (°C)	29·5 —— 44·5 (40·0)+

+Initial Guess.

table 10·5, Package Chiller System Problem Variables and Product Ranges.

Model.	Supply Fan.	Extract Fan.	Coils.				Chiller.
			Supply.		Extract.		
	Impeller Power.	Impeller Power.	Duty.	Air Loss.	Duty.	Air Loss	Compressor Power.
Net Energy Model.	e +	e +	/	/	/	/	e +
Pay back Period Model.	/	/	o +	e +	/	e +	e +

e: Fuel Type = Electricity.
o: Fuel Type = Oil.
±: Add or sutract value in model. (/ Term excluded).

table 10·6, Package Chiller System Energy Models.

Objective Function.	Fan Diameter.		Heating (supply) Coil.				Cooling (extract) Coil				Chil'r Size	Mass Flow	Fluid Temperature.		Objective Function Value.
	Supply.	Ext-ract.	Rows.	Width.	Height.	Cir-cuits.	Rows.	Width.	Height.	Cir-cuits.	Cata-log'No.	Cond-ens'g.	Cond-ens'g.	Evap-orat'g.	
Net Energy	1·12	1·12	4	1·5	1·75	30	2	2·0	2·0	20	70	2·65	38·5	9·0	721·5GJ/annum.
Primary Energy.	1·12	1·12	4	1·5	1·75	30	2	2·0	2·0	20	70	2·65	38·5	9·0	2,755 GJ/annum.
Capital Cost.	0·9	0·9	4	1·5	1·75	20	2	1·5	1·75	20	50	4·0	40·0	7·0	18,330 pounds.
Operating Cost.	1·12	1·12	4	1·5	1·75	30	2	2·0	2·0	20	70	2·63	38·5	9·0	8,204 pounds/annum.
N.P.V.	1·12	1·12	4	1·5	1·75	30	2	2·0	1·5	20	60	2·8	37·0	8·3	99,730 pounds.
Payback Period.	1·12	1·12	4	2·0	1·25	10	2	2·0	1·5	30	60	6·0	36·3	7·85	5·1 years.

Dimensional units as table 10·5.

table 10·7, Package Chiller System Results.

The solutions for the net and primary energy consumption and the operating cost objective functions are identical. This is predictable as each term in the energy model is associated with the same fuel type, eliminating any difference in weighting of the energy terms between the net energy consumption and primary consumption or operating cost. The characteristics of these solutions are influenced by the proportional control of the heating coil and the fluid property variables assigned to the problem.

The performance of the chiller is related directly to the evaporating and condensing water flow temperature, with a maximum chiller efficiency obtained for the lowest condenser water flow temperature and the highest evaporator water flow temperature. Therefore it has been assumed that both the evaporator and condenser water flow temperatures should be included as problem variables. However, in hindsight this has produced a less than flexible problem in the case of the cooling coil selection, the size of the coil having no interaction with the chiller performance and size as the determining factor, the evaporator flow temperature is fixed by the optimisation algorithm. Greater flexibility could be achieved by defining the evaporator water mass flow rate rather than temperature as the problem variable. This, with the chiller evaporating load dictated by the condensing load, would have allowed the evaporating temperature to change with coil size or water mass flow rate and result in greater interaction between the coil and chiller.

The consequences of a poor problem definition are reflected in the results obtained for the net energy consumption objective function. Lack of coupling between the cooling coil and chiller has resulted in a coil size which is influenced only by the energy consumption of the extract fan and produces a coil of least air resistance, ie: one with the smallest number of rows and largest width and height. The solution for the number of water cicuits is unchanged from the initial guess as the poor interaction reduces the affect the circuits have on chiller performance. As would be expected a high evaporating water flow temperature has been selected as this gives maximum chiller efficiency. Both the solution for coil rows and evaporator water flow temperature are on their bounds which suggests that if these bounds were extended a smaller coil and higher temperature would be selected.

The predominant factor in determining the energy consumption of the chiller is the condensing load imposed by the heating coil and which is a variable due to the range of conditions allowed under proportional control. Therefore the optimum size of heating coil for minimum chiller energy consumption is one which just produces the lowest temperature allowed by the proportional controller when the coil is operating under its greatest load. This is reflected in the results as the number of heating coil rows has tended towards its lowest bound, but has been prevented from reaching it because a smaller coil from that of the solution is unable to maintain the supply temperature. Similarly, the condenser water flow rate and set point temperature of the chiller have been reduced from the intial guess until a further reduction results again in too low a supply temperature. The major restriction on a reduction in the width and height of the heating coil is the face velocity constraint, although obviously a further reduction in size from the solution would also produce a coil size unable to maintain the required conditions. The net result of this reduction in coil size and value of fluid variables is that the system is just able to maintain the lowest feasible supply temperature of 27.75 ^{o}C with the diverting valve fully open and diverting less than 1% of the flow. The minimum energy consumption solution for chiller size lies at what is normally considered to be a true minimum, ie: an increase or decrease in size of chiller leads to an increase in energy consumption (figure 8.10). Predictably the most efficient fans have proved to be the largest with both supply and extract fans having solutions which lie on their upper bounds.

The solution for capital cost reflects that the cheapest components are the smallest and that the problem variables which do not affect capital cost and have remained at their initial values are the coil water circuits, condenser water mass flow rate, chiller set point temperature and evaporator water flow temperature. The solutions for the net present value and payback period objective functions are influenced by both the capital and operating cost of the system, with as for the run-around coil system, the width and height of the coil influenced more by the capital cost element. Even though both fans are excluded from the formulation of the payback period objective function, their sizes have changed from the initial guess.

The size of supply fan has changed as the heat input to the supply air by the fan indirectly affects the load on the chiller. However, the exhaust fan has no influence directly or indirectly over the formulation of operating costs and has changed size due to inaccuracies in the operating point found by the GRG2 simulation algorithm (section 8.5.2).

10.2.6 Computational Speed.

The most contributory factor in limiting the use of the software to the design of small sub-system is the computational speed of the solution procedure and in particular that of the GRG2 solution algorithm. Although design solution times for the example systems using the pattern search algorithms, range between three and nine hours, the majority of this is occupied in finding the system operating point, any computation directly related to the optimisation algorithm taking but a few minutes. Each evaluation of the system operating point by the GRG2 algorithm averaged three minutes computation time and had the load profile been more realistic would have been longer by a multiple of the increase in time periods. For example, 24 time periods instead of one would result in a computation time for each simulation of 24 x 3 = 72 minutes and with a minimum of 65 evaluations of system performance required by the optimisation (table 8.2), the lowest design computation time would be 78 hours (65 x 72 minutes). Obviously if the software is to be applied practicably, considerable improvement is required in these computation times both through the development of improved solution algorithms and the implementation of the software on a more powerful computer system.

10.3 The Development of a Design Tool.

The immediate application of the optimised design software is to aid the selection of HVAC systems by providing quantitative information, such as operating cost with which to compare alternative schemes. By optimising the size of the components and system operating point, the software ensures that the quantitative criteria used in the comparison of systems. represents the best solution obtainable for each scheme.

Development of the software is required if the solutions are to represent true optimum designs. Many of the simplifications made in formulating the objective and constraint functions in this research will have to be resolved if finite rather than characteristic solutions are to be found. These modifications range from simple programming, such as linking realistic fuel tariffs to the time periods of the load profile, to fundamental research such as in improving the integrity component models. Most important is the development of robust and efficient simulation and optimisation solution algorithms, as without these the future development of the software as a design tool is limited by its inability to provide a reliable solution quickly.

The most obvious criteria for the comparison of systems are represented by the objective functions, yet the software can be developed to provide less direct information by establishing the sensitivity of solutions to changes in design criteria and by identifying the inefficient operation of items of plant within the system.

10.3.1 Indentifying Poor Scheme Designs.

Poor scheme designs can be recognised by identifying components whose operation throughout the load profile is negligible. Optimum solutions which lie on the bounds of the variables suggest that either the bound of the variables are not wide enough or that if the bounds represent the range of available products that the system configuration may benefit from a redesign. This is of particular relevance to solutions with variables on their lower bounds as where solutions tend towards the smallest components the size of component should be reduced until either it is just large enough to meet the imposed load or to comply with some other constraint.

Using the lower bounds to identify a poor system configuration is in itself unreliable as it is feasible that the optimum size of component is the smallest component. Therefore the operation of components which lie on the lower bounds must be investigated by analysing the fluid variables related to the component. For example, if under the greatest load the mass flow rate through a heating coil is very small then the coil would only just be operating and it would be fair to suggest that it could be dispensed with.

This technique can complement the design process in that although a comparison of different scheme designs based on the objective function values identifies the inefficient schemes, this technique goes a step further and identifies the part of the systems operating poorly and hence which would benefit from redesign.

10.3.2 Design Reliability - Sensitivity and Risk Analysis.

A further indirect source of information useful to the design process, is sensitivity analysis. It is well known, for example, that when using the net present value calculation as an economic comparator that the apparent advantage of adopting one scheme over another can change significantly with a fluctuation in interest rates and often leads to a change in ranking of schemes. Sensitivity analysis can help establish the reliability of solutions by assessing the degree of change in design data, such as fuel prices and interest rates, which produce an unacceptable change in solutions. For example, if a 1% change in fuel prices produces a solution with a 2% change in the optimum size of component it could be said that the operation of the system is very sensitive to changes in fuel prices and may become uneconomical to run in the future.

Results from this type of analysis could be used in addition to the objective function values when comparing systems, as although one system may have a lower objective function value, it may be so sensitive to changes in design data that the system with a higher initial objective function value is selected.

A similar approach to that of identifying poor system configurations by locating the components which are only just operating, is to use the sensitivity analysis to locate the components which are most sensitive to design criteria. This may lead to changes, not necessarily in system configuration, but in component specification, for example from an oil fired boiler to a gas fired boiler.

A form of sensitivity analysis already occurs in the optimisation process as the change in objective function value for an increment in value of the variables is calculated in order to assess the direction of the optimum. This information could be used to recognise poor problem definitions by identifying the variables which have little affect on the object function. This would not include the variables which normally have no affect on the objective function values, such as the fluid temperatures which do not directly affect the capital cost of components. An example is illustrated by the package chiller system (section 10.2.5) where definition of inappropriate problem variables resulted in the size of cooling coil having little interaction with other components and negligible affect on the objective function value. Using this information to identify a poor problem definition would require some skill but the chance of a poor definition becomes less as the system model becomes more sophisticated and where the full system is modelled there is no risk since the only variables available for inclusion in the optimised design process, apart from the size of components, are the controller set points.

The changes in objective function values about the solutions, for an increment in variable values could assist the designer if a compromise in the solutions obtained for different objective functions was required. Although this is unlikely, indicating the effect each variable has on the objective functions can enhance the understanding of the characteristic behaviour of each solution.

A superior technique to sensitivity analysis is that of risk analysis. It could be argued that the finite calculation of the objective function values is unlikely to be reliable even with the development of sophisticated models, since small changes in design data can invalidate the solutions. This is of less importance when solutions are used in the comparison of systems, especially when supported by a sensitivity analysis. However where finite values are required for predicting say, operating costs of the building, the solutions should be supported by an assessment of the risk of the solution becoming invalid. This will become a critical part of the future design methodology as economic constraints are continually demanding more efficient and cost effective designs. Current design methods tend to 'over' design systems in an attempt to ensure an arbritrary and often unknown factor of safety. As economic pressure increases, forcing tighter design limits, it will become increasingly important to assess the likely probability of designs failing. Risk analysis uses the variation in design data and probabilities of the variations occuring, to determine the likely deviations in solutions. Successful implementation of such a procedure will require fundamental research into the variation in design data. For instance, research would be required to establish the probability of occupants changing and by how much, the controller set points. However sufficient data is available on the fluctuations in interest rates, fuel prices and variation in climatic conditions, to justify the development and implementation of the statistical procedures required for risk analysis.

The most beneficial application of risk analysis is not simply to support solutions with the risk of failure, but eventually to develop design constraints which ensure the solutions obtained are for a predefined rather than arbitrary risk of failure. This would enable clients or designers to stipulate predefined design limits, for example a design brief might stipulate that not only should the system be designed for minimum operating costs, but that once installed the predicted costs should not vary from the actual costs by more than 5% and that the risk of the system being unable to meet the imposed loads should be less than 10%. This approach will dramatically increase the computational time of solutions and at present is likely to prove prohibitive, however future developments in computer hardware and operating systems will eventually make this a feasible and valuable design approach.

10.3.3 Energy Consumption and Automatic Controls.

The importance of modelling a realistic control action is emphasized by the solutions obtained for the example systems, since where energy consumption is a prime consideration the components are often sized such that the controlled conditions are just maintained under the greatest loads. A recent CIBS (1985) Applications Manual on automatic controls identifies several points for consideration when designing HVAC systems for minimum energy consumption. Many of these points are an integral part of the optimised design process.

It is difficult in a manual design technique to match the efficiency characteristics of the plant with the action of the controller. Using the optimised design software to minimise energy consumption, by definition ensures that the plant is sized to give the highest efficiency over the range of control. This range of control could be extended in future software development to include the start up of the system ensuring that the capacity and efficiency characteristic of the plant would be matched to the control system for both start up and normal running.

The controller throttling range has a marked affect on energy consumption. An increase in room temperature of 1°C above normal can increase energy consumption by up to 10%. Selection of a suitable throttling range is initially linked to comfort criteria, but during detailed design may be restricted to ensure the control action is stable with no hunting across the throttling range and that where the system allows, there is no overlap of heating and cooling. Such criteria could be handled by the optimisation software by defining the controller throttling range as a continuous variable and formulating a constraint function which, say, defines the percentage of occupants satisfied with the comfort conditions. This would allow the search to find the optimum throttling range for minimum energy consumption whilst ensuring comfort conditions are maintained. Throttling ranges which produce unstable control conditions would be rejected as this is unlikely to maintain the controlled condition and would produce a high energy consumption, as would an overlap of heating and cooling.

Obviously modelling of realistic controller action is severely restricted by the steady state simulation procedure as this excludes the ability to model the system start up and controller stability. Although the immediate development of the optimised design software can benefit most from developing the steady state simulation procedure, eventually a dynamic simulation procedure will be required to allow the modelling of controller action and system response. However this will present little problem as the modular structure of the optimised design software will facilitate its integration with an existing dynamic simulation technique.

Chapter 11. CONCLUSIONS AND FUTURE DEVELOPMENT.

The conclusions of this research are that it is feasible to develop an optimisation procedure for the optimum design of HVAC systems, although the practicable application of the procedure is at present restricted by poor computational speed and the integrity of constraint and component models.

11.1 The Optimisation Parameters.

The optimisation procedure developed can usefully improve the effectiveness of the design process and reliability of designs by providing quantitative criteria with which to compare system performance and by allowing components to be selected simultaneously as a system at an early stage in the design process. The procedure implemented has been structured to comply with the three parameters of numerical optimisation problems: the problem variables, the design constraints and the objective functions. Additional consideration has been given for the integration of the design procedure with the Loughborough University system simulation procedure, SPATS, which is used to define the system configuration and simulate its steady state performance.

The problem variables of HVAC system optimisation problems are the dimensional and operating variables of the components used in their selection and the operating variables of the system such as the controller set points. The physical connection of adjacent components is ensured as the matching dimensions can be defined to be the same variable. Each variable can be defined as discrete or continuous and suitable product ranges can be selected from a comprehensive data base, although in the solution process this is restricted to one range per component. Future development of the procedure should allow several product ranges to be assigned to a component and the optimisation procedure extended to exhaustively search all combinations of product ranges for the one with the best solution.

The most important constraint imposed on any HVAC system design is that the system and therefore components in the system should continue to operate under all load conditions. Formulation of a component undersizing constraint function has proved to be difficult and relies heavily upon the interpretation of the results from the system simulation. The future application of the optimised design procedure is dependent on the successful development of a component undersizing constraint function and therefore this should be a major element of future research. Other design constraints which can be defined within a design problem are limitations on fluid parameters such as fluid velocities, physical limitations on component size represented by bounds on the variables and limitations on component configurations.

The objective functions of HVAC optimised design problems are the criteria used in comparing the performance of different schemes. Those implemented in this research are common criteria but have been selected in particular to provide a wide range of objective function characteristics with which to develop a solution algorithm. The objective functions available are: system net energy consumption, primary energy consumption, capital cost, operating cost, net present value and payback period.

Evaluation of all objective functions, except capital cost, requires an assessment of the system energy consumption and therefore a procedure has been implemented to formulate a system energy model from the energy terms of the individual components. Three categories of term are identified: direct, ancillary and extraneous. Direct terms represent the net energy consumption of the component whereas ancillary and extraneous terms can only be expressed as net energy when the performance of components other than the referencing component is known. Formulation of a system model using these terms enables sub-systems to be designed and simplified component models to be implemented. Development of more sophisticated component models will in the future eliminate the use of ancillary terms. Other parameters included in the definition of the system model are the fuels used and an indicator which defines whether the value of an energy term is added or subtracted in the model.

11.2 The Solution Procedure.

HVAC system optimised design problems can be solved by numerical optimisation methods the characteristics of which, if maximum efficiency is to be achieved, must match those of the problem variables and constraint and objective functions. The most important characteristics of these parameters in this respect are: most problem variables are discrete and cannot be approximated as continuous, the constraint functions are non-linear and the objective functions non-linear and discontinuous. Two categories of optimisation method exist, derivative methods which employ the derivatives of the objective and constraint functions in a search for the optimum and direct search methods which base their search strategy on a simple comparison of the function values at series of trial points. The choice of technique for solving HVAC system design problems is restricted to direct search methods as the discontinuous nature of the problem variables severely affects the stability of the numerical techniques used to calculate the derivatives which are required by mathematical based solution methods.

Each evaluation of a constraint or objective function by the solution procedure, requires simulation of the systems performance. Solution of the system performance equations is by a generalised reduced gradient method. This however is slow to converge on a solution which results in a prohibitive computation time restricting design problems to that of simple systems consisting of a few components. The algorithm also lacks stability in simulating performance over a wide range of component sizes and system operating conditions. The practicable application of the optimisation software therefore depends upon the improved computational speed and stability of the simulation solution algorithm.

Development of this procedure for the solution of HVAC system optimised design problems, with particular reference to improved constraint handling and computational speed, is the central element of a research project starting in January 1987 at the Universities of Liverpool and Loughborough and which is funded by the Science and Engineering Research Council of the U.K.

11.3 Component Model and Data Base Development.

Component performance model development is the subject of many research projects. Exchange of information and of algorithms through organisations such as the International Energy Agency, will lead to improved integrity and wider applicability of the models. Development of a comprehensive product data base is however restricted by manufacturers reluctance to release information and present it in a standard format. Fortunately, growing pressure is forcing manufacturers not only to comply with standards of component testing and data presentation but also to comply with standards of manufacturing quality. This is most evident in the air moving section of the industry with the introduction of the BS 5750 (1979).

Although development of component performance models is progressing, very little attention is given to the development of component maintenance and capital cost models. Development of component maintenance cost models requires extensive research and is a project area yet to be initiated. Development of component cost models which are applicable to equipment supplied from several manufacturers can only be achieved through the release of more information by manufacturers and in particular the standardisation of methods of presenting data.

11.4 Development of a Design Tool.

Only the thermo-fluid performance of the system has been considered in the design of HVAC systems. True optimum design however can only be achieved when the acoustic and thermo-fluid properties of the system are considered simultaneously. Extension of the component models to include their acoustic performance and the addition of acoustic design constraints will not only broaden the range of application of the software but will also improve the reliability of the design solutions.

The integrity of solutions can be further improved by the development of a probabilistic design procedure as the stochastic nature of design parameters such as climatic conditions and fuel prices, prevents the finite evaluation of solutions. Implementation of a sensitivity analysis procedure will help establish the reliability of solutions by assessing the degree of change in design data which produces an unacceptable change in the solution. A somewhat superior technique is that of risk analysis which attempts to quantify the risk of solutions becoming invalid and therefore its implementation would improve the integrity of the software by including in its formulation a self validation procedure.

REFERENCES.

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

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

ASHRAE, 1984 'Life Cycle Costing.', ASHRAE Handbook, Systems Volume, 1984, Chp. 42, pp. 42.1-42.8, ASHRAE, New York, 1984.

Baxter, 1985 Baxter. A.J., 'Cost Effective Design by Computer - The Software Future.', Computers in Building Services Design. Hanby. V.I. (editor). CICC Publications, 1985.

Bowman, 1986 Bowman. N. and Lomas. K., 'Does Dynamic Simulation Work?', Building Services, Vol.8, No.3, 1986.

Box, 1965 Box. M.J., 'A new method of constrained optimization and a comparison with other methods', The computer Journal, No.8, 1965.

B.R.E., 1976 'Energy consumption and conservation in buildings', Building Research Establishment Digest, 191, July 1976, Garston, Watford, WD2 7JR.

B.S. 848., 1980 B.S. 848, Part 1, 1980, 'Fans for general purposes, Part 1 Methods of testing performance', British Standards Institution, 2 Park Street, London, W1A 2BS.

B.S. 5750., 1979 B.S. 5750, Part 1, 1979. 'Quality Systems, Part 1, Specification for design, manufacture and installation.', British Standards Institution, 2 Park Street, London, W1A 2BS.

Burgess, 1982 Burgess. C.K. and Pearson. J., 'Cost Effective Energy Conservation for Indoor Swimming Pools', (report MRS E 401), British Gas Corporation, Midlands Research Station, Wharf Lane, Solihull, West Midlands, B91 2JW., 1982.

Carroll, 1961 Carroll. C.W., 'The Created Response Surface Technique for Optimising Nonlinear Restrained Systems', Operational Research, No.9, 1961.

CIBS, 1977 'Owning and Operating Costs', CIBS Guide, B18, 1977, CIBSE, Delta House, 222 Balham High Road, London, SW12 9BS.

CIBS, 1985 'Automatic Controls, Applications Manual', pp. 8-9., 1985, CIBSE, Delta House, 222 Balham High Road, London, SW12 9BS.

Dixon, 1972 Dixon. L.C.W., 'Nonlinear Optimisation', The English Universities Press Ltd., 1972.

English	English. H., Private Communication (ref: The use of gradient based search methods with discrete variables.), Dept. Engineering Mathematics, University of Technology, Loughborough.
Gill, 1981	Gill. P.E., Murray. W. and Wright. M.H., 'Practical Optimization', Academic Press, 1981.
Hooke and Jeeves, 1960	Hooke. R. and Jeeves. T.A. 'Direct Search Solution of Numerical and Statistical Problems', J. of Assn. of Computing Machinery, 8, pp. 212-229., 1960.
Lasdon, 1978	Lasdon. L.S. and 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 Warren. 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.
Milbank, 1971	Milbank. N.O., Dowdall. J.P. and Slater. A., 'Investigation of maintenance and energy costs for services in office buildings', Building Research Station Current Paper. 38/71, 1971, Building Research Station, Garston, Watford, WG2 7JR.
Murray, 1984	Murray. M.A.P., 'Component Based Performance Simulation of HVAC Systems', Doctoral Thesis, University of Technology, Loughborough, 1984.
NAG	Numerical Algorithms Group, Mayfield House, 256 Banbury Road, Oxford, OX2 7DE. (Ref:E04-Minimizing or Maximizing a Function and Graphics Package.).
NIFES, 1985	National Fuel Efficiency Service Ltd., NIFES House, Sinderland Road, Broadheath, Altrincham, Cheshire, WA14 5HQ.
Rosenbrock, 1960	Rosenbrock. H.H, 'An Automatic Method for finding the Greatest or Least Value of a Function'. The Computer Journal, No.3, pp. 175-184, 1960-1961.
Stark, 1970	Stark. P.A.. 'Introduction to Numerical Methods.'. Tne Macmillan Company, pp. 319-327, 1970.
Stoecker, 1971	Stoecker. W.F., 'Design of Thermal Systems', McGraw-Hill Book Company, 1971.
Swann, 1978	Swann. W.H., 'Constrained Optimization by Direct Search', a chapter in 'Numerical Methods for Constrained Optimization', (P.E. Gill and W. Murray, editors), pp. 191-217, Academic Press, London and New York, 1978.
Wright, 1984	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.

Wright, 1985 Wright. R.N., 'Computers in buildings, building and building research', BSER and T, 6, No.1. pp. 21-27, 1985

Wix, 1985 Wix. J., 'Computers in Building Services Design', Computers in Building Services Design. Hanby. V.I. (editor), CICC Publications, 1985.

Appendix A. THE APPLICATION OF THE LEAST SQUARES CURVE FITTING METHOD TO THE MODELLING OF HVAC COMPONENT PERFORMANCE.

An increasing amount of research activity is taking place in the area of HVAC system simulation which in turn has focussed attention on the formulation of individual component models. Two approaches to HVAC component model development can be distinguished, those developed from fundamental principles and those developed from manufacturers' performance test data. A significant benchmark in the evaluation of the latter type of model was the publication of a curve fitting technique and several component models by ASHRAE (1975). The application of these techniques to the modelling of equipment supplied by different manufacturers is limited as the shape of the curves for which the fits are derived may vary from one manufacturer to the next. In order to maximise the applicability of component models and to promote the development of comprehensive data bases comprised of several manufacturers data, a method of curve fitting is required which automatically adapts to the particular performance characteristics of plant supplied from each manufacturer.

A commonly used technique which lends itself to the modelling of HVAC components performance is the polynomial form of the least squares method. Formulating this curve fitting procedure such that the order of polynomial is a variable enables the curve fit to adapt to the characterisitics of individual components. An added advantage of employing the least squares technique is that it can be formulated to fit a function of several variables, which is often required when modelling components such as pumps and fans.

Although the flexibility of the least squares method of curve fitting is suited to modelling HVAC component performance, it is by no means always evident which order of polynomial gives the best fit to the data. Practicable use therefore of this method of curve fitting, requires the development of a procedure which can automatically assess the accuracy achieved from a particular order of polynomial and make a decision as to which order, if necessary, will improve the accuracy.

The Least Squares Method-Three Dimensional Polynomial Form.

The general form of the least squares curve fit is given by the expression:

$$f(x) = a_0 g_0(x) + a_1 g_1(x) + \ldots . + a_m g_m(x) \tag{1}$$

where the function g_i is chosen in such a way that no g_i can be expressed as a linear combination of any other g_i. The least squares approach is that the coefficients a_i should be chosen such that the sum of the squares of the deviation of the fit $f(x_i)$ from the n+1 data points $Y(x_i)$ should be minimised. A set of simultaneous equations which satisfy this criterion and which can be solved for the coefficients a_i, are those derived by Stark (1970):

$$\begin{array}{cc} & \longrightarrow k \\ j \big\downarrow & \begin{bmatrix} \alpha_{01}, & \alpha_{11}, & \alpha_{21}, & \ldots . & \alpha_{m1} \\ \cdot & & & & \\ \cdot & & & & \\ \cdot & & & & \\ \alpha_{0m}, & \ldots & \ldots & \ldots & \alpha_{mm} \end{bmatrix} \end{array} \begin{bmatrix} a_0 \\ a_1 \\ \cdot \\ \cdot \\ a_m \end{bmatrix} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \cdot \\ \cdot \\ \beta_m \end{bmatrix} \tag{2}$$

$$\text{where } \alpha_{kj} = \sum_{i=0}^{n} g_k(x_i) g_j(x_i) \tag{3}$$

$$\text{and } \beta_k = \sum_{i=0}^{n} Y(x_i) g_j(x_i) \tag{4}$$

Any conventional solution method can be used to solve this set of equations for the polynomial coefficients a_i, although a simple Gaussian elimination technique has proved to be of sufficient accuracy.

The functions $g_i(x)$ are of a general form and can be substituted for any linearly dependent functions. The most suitable form for modelling HVAC component performance is the polynomial form for a fit as a function of two dimensions: two dimensions is chosen as many component models can be developed in this format and increasing the number of dimensions further, increases the complexity of the fit making it more difficult to assess its behaviour and uniqueness. The polynomial form is particularly suited to HVAC component modelling as often component performance can be modelled using simple quadratic or cubic polynomials. If the two independent variables are x and z and the corresponding powers of fit Px_i and Pz_i. then for a fit as a function of two dimensions the polynomial form of the functions g_i is:

$$g_i(x,z) = x^{(Px_i)}.z^{(Pz_i)} \tag{5}$$

In order to implement this procedure two algorithms are required, one to formulate the simultaneous equations (2) and another to return values of the curve fit according to the least squares expansion, equation (1). Both must be capable of handling any order of polynomial as this must be a variable if the procedure is to be flexible enough to curve fit a variety of components performance. Fundamental to the development of these algorithms is the relationship which associates a particular coefficient in the expansion with the powers to which the corresponding independent variables are raised:

$$k = Px_k + Pz_k(Pxo + 1) \tag{6}$$

where k is the coefficient subscript, Px_k and Pz_k the powers of x and z and Pxo the maximum power (termed the order), to which x is raised. For example, if x is raised to an order of 2 and z to 1 then the polynomial expansion would be:

$$f(x,z) = a_0 + a_1x + a_2x^2 + a_3z + a_4zx + a_5zx^2$$

where the relationship between the powers and the coefficients are:

Px:	0	1	2	0	1	2
pz:	0	0	0	1	1	1
k:	0	1	2	3	4	5

An algorithm for the formulation of the α and β matrices which is based upon the relationship of equation (6), is represented by the flow diagram of figure A1. Use has been made of the symmetrical nature of the α matrix as values are computed for the upper triangle only and simply duplicated in the lower triangle. An algorithm for the calculation of the dependent variable $f(x,z)$ for a given set of coefficients and order of polynomials is illustrated by the flow diagram in figure A2. It is useful to note that both algorithms can be used for data which is a function of one variable by setting the order of z to zero.

Search for the Optimum Order of Polynomial.

Practical use of the curve fitting procedure requires the development of an automatic method of finding the order of polynomial that gives the best fit to the data. as this is by no means always evident and often requires an extensive investigation. Automation of this process can be achieved through the application of an optimal search technique which searches for the order of polynomial that gives a specified level of accuracy.

Use of a detailed statistical regression analysis to assess the accuracy achieved by a particular order of polynomial is unnecessary as in general the empirical data to be modelled will have had its statistical significance assessed before its publication. A suitable criterion with which to assess the accuracy of the curve fit is the worst error which occurs between the curve fit and any one of the data points. In order to allow this criterion to be used with data values of differing magnitude, it is convenient to normalise the deviation by dividing it by the range of data values (figure A3):

$$\frac{|f(x_i.z_i) - Y(x_i.z_i)|max}{Y(x_i,z_i)max - Y(x_i,z_i)min} \qquad (7)$$

In choosing a value of normalised deviation to represent a suitable level of accuracy, it is important to consider the benefit obtained in improved accuracy against the possible increase in number of polynomial coefficients and therefore data storage requirements and computational time.

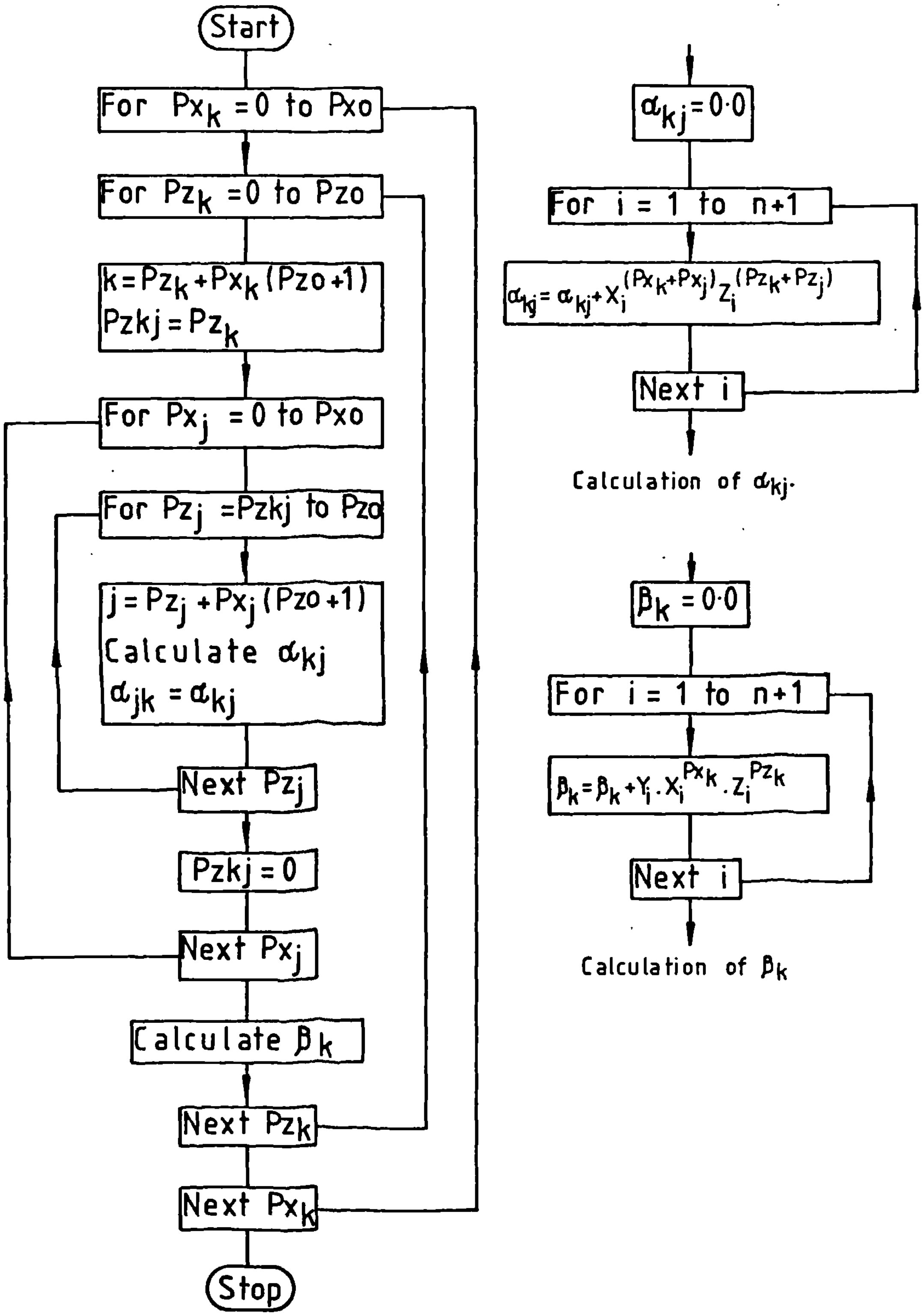


figure A1, Formulation of the α and β Matrices.

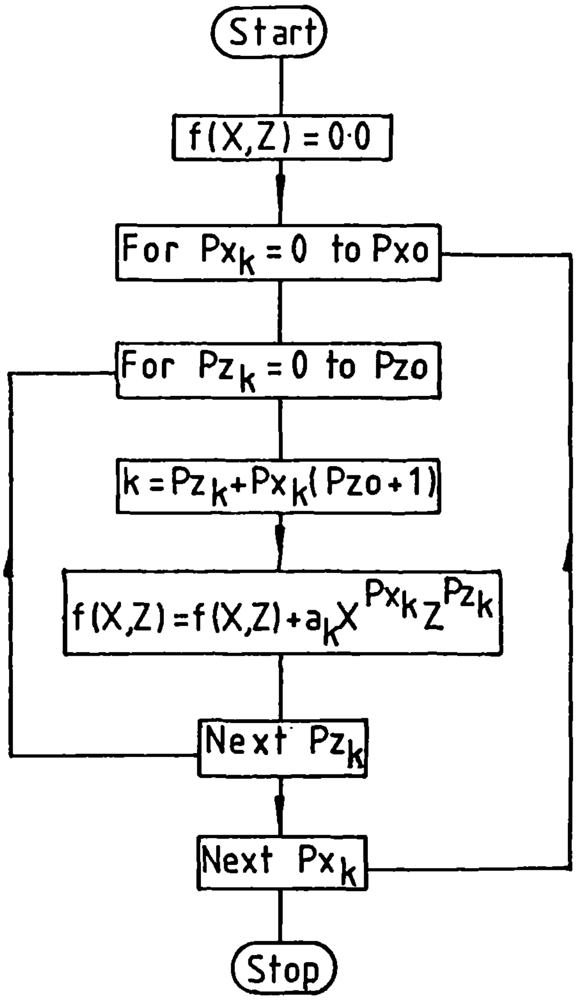


figure A2, Calculation of f(X,Z).

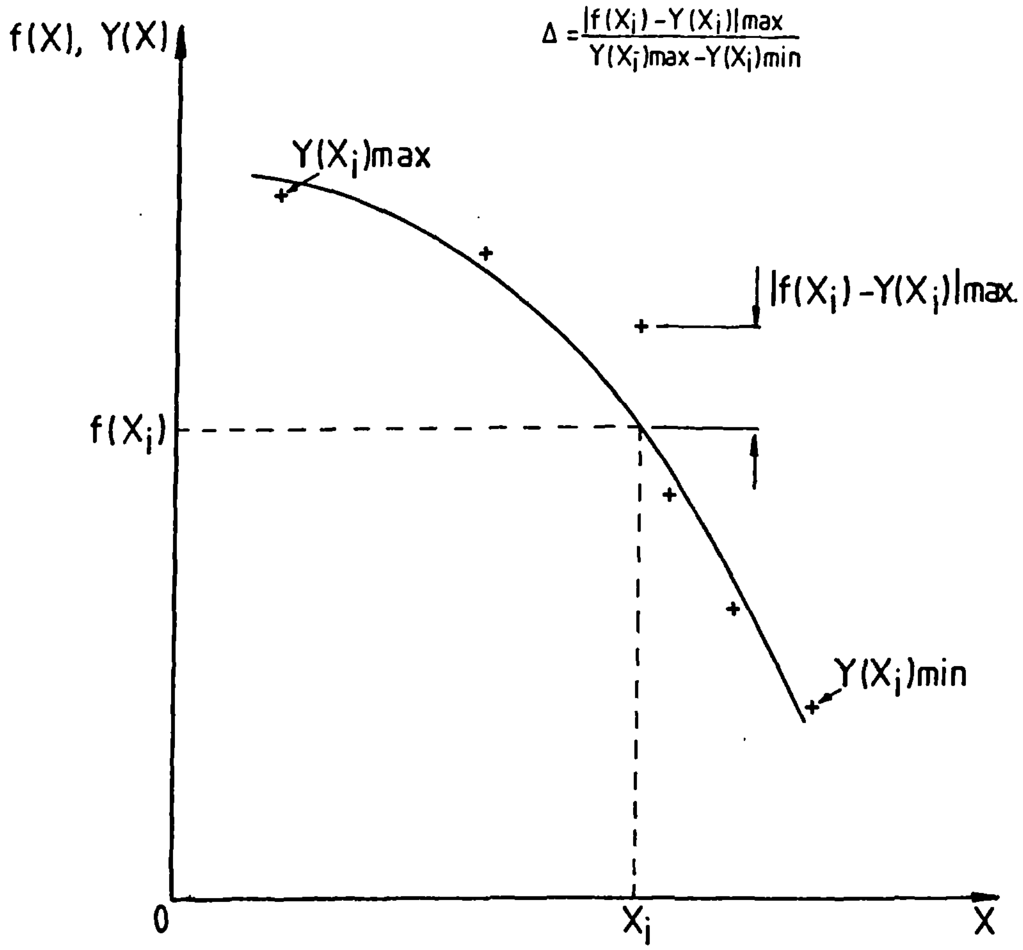


figure A3, Normalised Deviation for a Function of One Variable.

A suitable level of accuracy for modelling HVAC component performance which in general leads to an acceptable number of polynomial coefficients is a normalised deviation of 2.5%.

In developing a search method which varies the order of polynomial to find the specified level of accuracy, it is important to consider the characteristic behaviour of the normalised deviation with respect to the order of polynomial. Generally, as the order of the polynomial is increased the normalised deviation will decrease. The rate at which the deviation changes depends on the order of polynomial and shape of the curves, but often reaches a point where an increase in order of polynomial produces only a marginal improvement in accuracy, as the error is then due to scatter in the data points. A change in normalised deviation of 15% or less for a unit increase in order of polynomial can be taken as an indication that increasing the order of polynomial further, will not produce a significant increase in accuracy. However. exceptions to this characteristic do exist and normally occur when there has been little improvement in the accuracy since an increase in order of the polynomial began.

A search method which lends itself to the characteristic behaviour of the normalised deviation is the multivariate search (Stoecker, 1971). The multivariate search method is one which probes along one co-ordinate axis, (ie:order of x) until no further improvement in accuracy is achieved: the search is then changed to the next co-ordinate axis, (ie:order of z) and the procedure repeated. The process of changing search direction continues until a search along any co-ordinate direction produces no improvement in accuracy and the resulting search point taken as the optimum solution. For the purposes of curve fitting HVAC component performance data, the criteria which dictate a change in search direction are:

1. If an increase in order of polynomial produces an increase in the normalised deviation.

2. If there has been a change in the normalised deviation of 15% or greater, but further increasing the order of polynomial produces a change of less that 15%.

3. If the change in normalised deviation has been less than 15% for the previous 5 increases in order of polynomial.

Although any of these can lead to a change in search direction and eventual convergence of the search, if at any time during the search a value of the normalised deviation of 2.5% or less is found, this can be taken as the solution point and the search abandoned. A final check on the solution point can be made to ensure that it cannot be improved, as there are occasions when a reduction in order of the polynomial produces a normalised deviation which is larger than the solution found by the search, but which is still less than 2.5% and has less coefficients.

A flow chart representing the multivariate search algorithm is illustrated in figure A4. A useful addition to this is a temporary data base of solution points created as the search progresses. This can be used to reduce the computational time as it is characteristic of the multivariate search to re-evaluate previously searched points. An example of the search is illustrated by the normalised deviation surface of figure A5, (and the data which was curve fitted, in figure A11). The progress of the search is shown by the solid arrows, the broken arrows illustrating search moves which were rejected. An order of polynomial of 5, for the variable x was found to be the solution by the multivariate search, but on checking the solution an order of 4 was found to comply with the limit of a 2.5% normalised deviation and was therefore accepted as the optimum solution as a lower order of polynomial requires less polynomial coefficients.

Application Methodology.

Two factors must be considered when applying the curve fitting procedure to modelling HVAC component performance: firstly the effect the characteristics of the data have on the fitting procedure and secondly any effect the behaviour of the curve fit may have on a system simulation procedure.

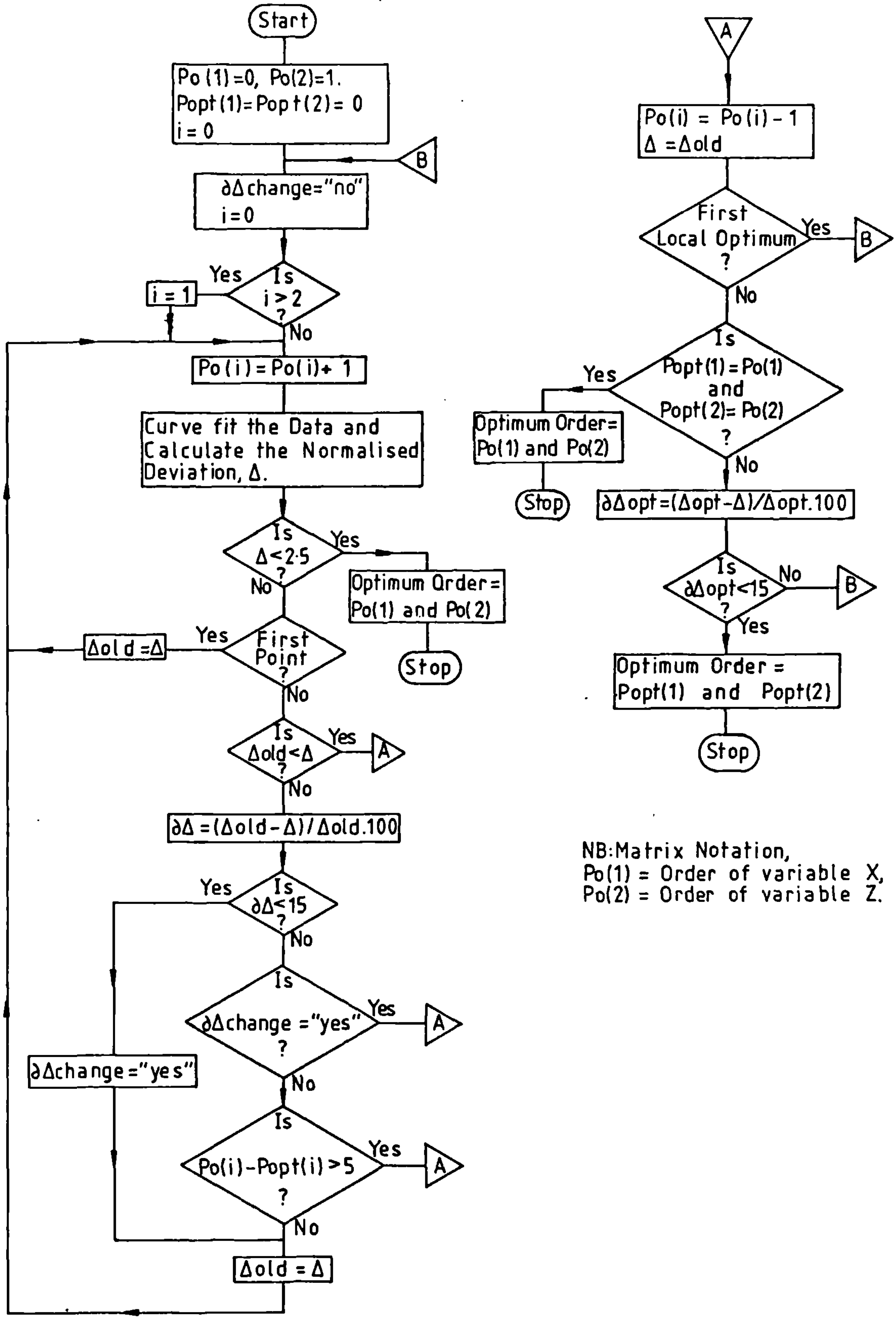


figure A4, Multivariate Search.

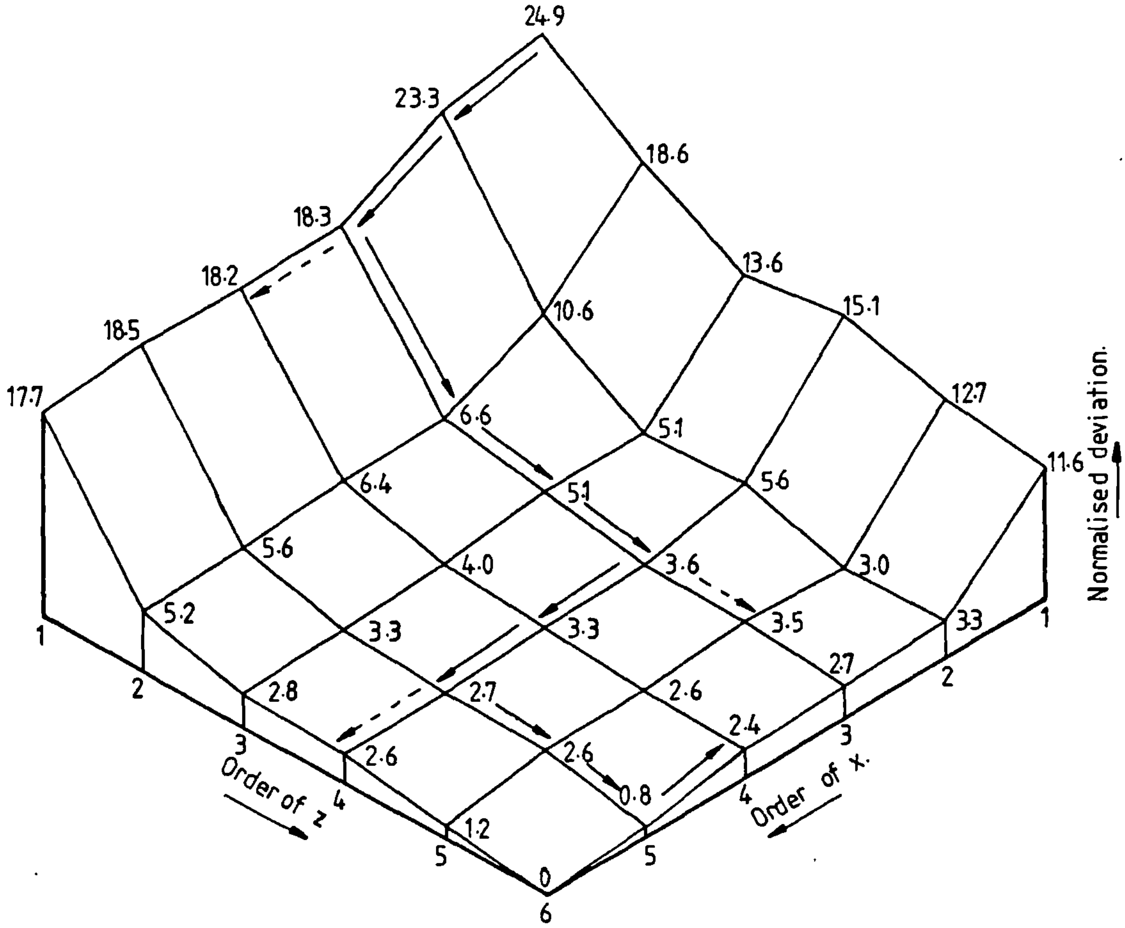


figure A5, Normalised Deviation Surface.

Although in general HVAC components performance data is 'smooth' as any spurious points will have been eliminated before publication, occasions do arise when the data to be modelled has a degree of 'scatter'. This normally occurs when the data has been transformed from its original state. For instance, in the modelling of centrifugal fan performance it is convenient to transform the published fan performance data to a non-dimensional form, as this reduces the number of variables in the curve fit (Wright, 1984). The occurrence of scatter in the transformed data points can be attributed to poor component performance measurement and/or incorrect assumptions in transforming the data.

Scatter in the data points influences the solution found by the multivariate search as in minimising the normalised deviation an order of polynomial will be selected which produces a curve fit with least deviation from any of the data points, including any spurious points. For example, figure A6 illustrates a set of data with two spurious points (1) and (2). The general trend of the data is represented by the curve (a) yet the curve selected by the multivariate search would be that of (b) as this has the least deviation from any of the data points. An example of transformed performance data for a centrifugal fan is illustrated in figure A7 and figure A8. Figure A7 illustrates the curve selected by the multivariate search, a higher order of polynomial rejected as this has a larger normalised deviation. Increasing the order of polynomial manually produces a curve which, although having a higher normalised deviation gives a better fit to the data (figure A8). Clearly, if the search for an optimum order of polynomial is to be used extensively with scattered data then the procedure would benefit from a different criterion for which to search. A suggested approach is to use the sum of the squares of the deviations as the few spurious points would then be 'outweighed' by the remaining points.

The sophistication of the system simulation procedure in which the curve fit is to be used influences the integrity required of the curve fit. Extrapolation of component performance beyond the known and measured performance is precarious and meaningless, yet some HVAC system simulation techniques may in their solution process, look at points which are outside the measured performance of the component and the region of data curve fitted.

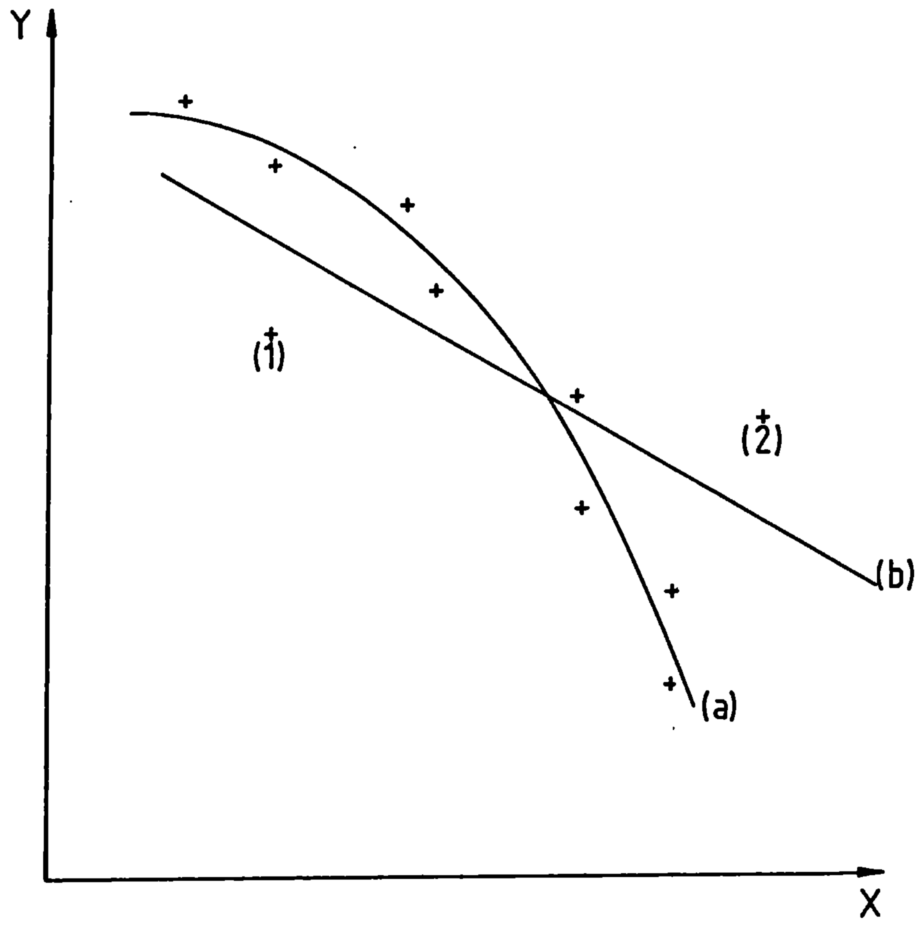


figure A6, Scattered Data Points.

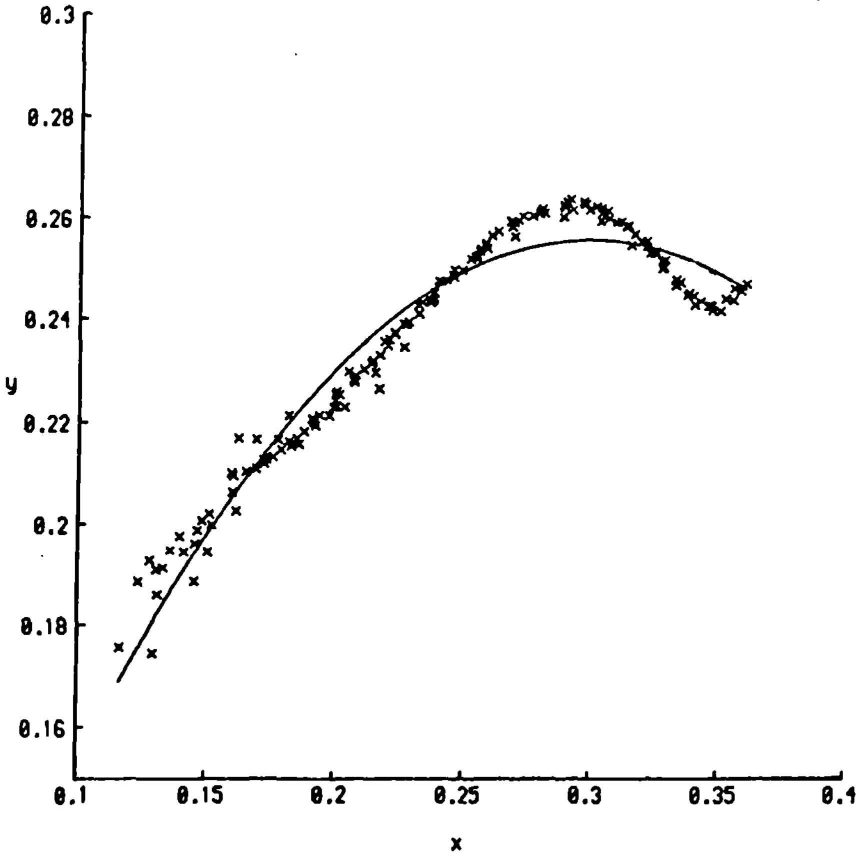


figure A7, Scatter in Transformed Data.

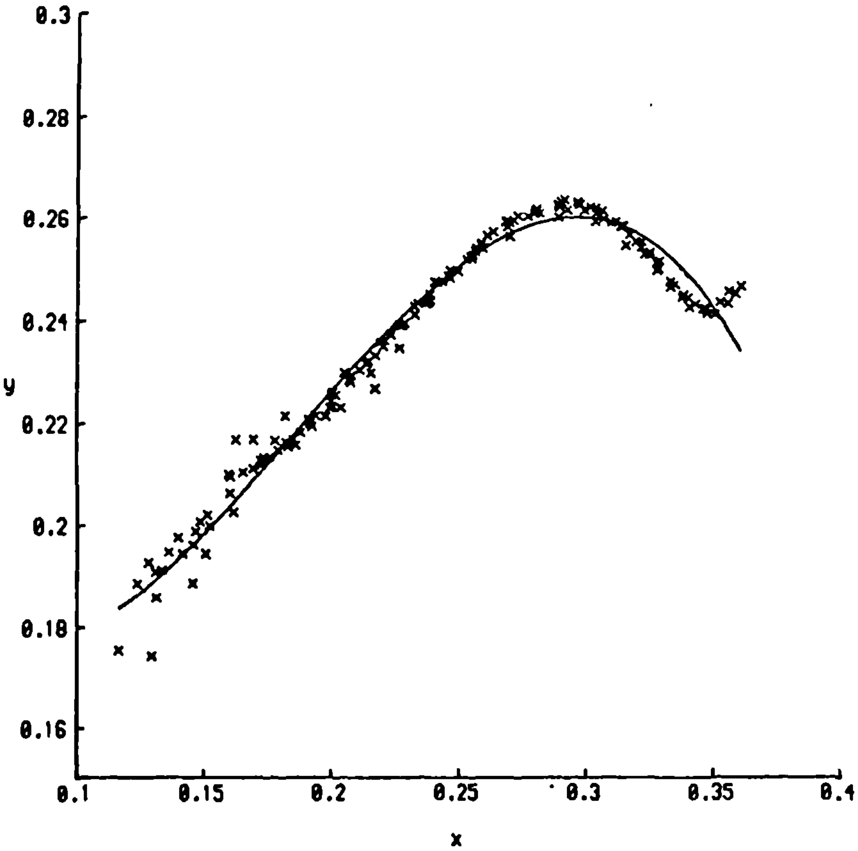


figure A8, Scatter in Transformed Data.

For example, figure A9 illustrates the characteristic behaviour of the least squares curvefit when curves are extended beyond the original data, the extended curves represented by the broken lines. If the system simulation procedure is unable to accommodate constraint functions which restrict the value of variable x in relation to the variable z, then two solutions are possible at the point (X1,Y1), a value of z=2 from the original data and a value of z=4 from the extrapolated data. In such cases it is wise to ensure that the curve fit produces unique solutions within the data region. This can be achieved by adding data points to the original data such that the extrapolated curves are drawn away from the original data region. The curve fit will then be as far as is possible, unique within simple bounds on the variables, (Xmax, Xmin and Ymax. Ymin, figure A10).

This approach is best implemented through the application of graphics software which enables the curves to be drawn and data points to be added. The user of the software can then add data points, curve fit the new data set and reassess the uniqueness of the fit very rapidly. As part of this procedure, it is an advantage to retain the ability to manually specify the order of polynomial. as intuition on behalf of the user can often lead to a quicker solution. The curve fit of fan performance data illustrated in figure A11, indicates the problem of producing a unique curvefit. The broken lines are the curves extrapolated for values of the variable x which lie outside the original data region and cross this from both above and below. The advantage of adding data points outside the original data region and curve fitting the new data set is illustrated in figure A12.

Interpolation of component performance using the least squares curve fitting technique can on occasions also produce some unexpected characteristics, as for the z=4.5 curve in figure A10. It is therefore prudent to plot a few intermediate curves to check their behaviour. If this proves to be unacceptable, lowering the order of polynomial often reduces the tendency for the curves to deviate from the expected trend.

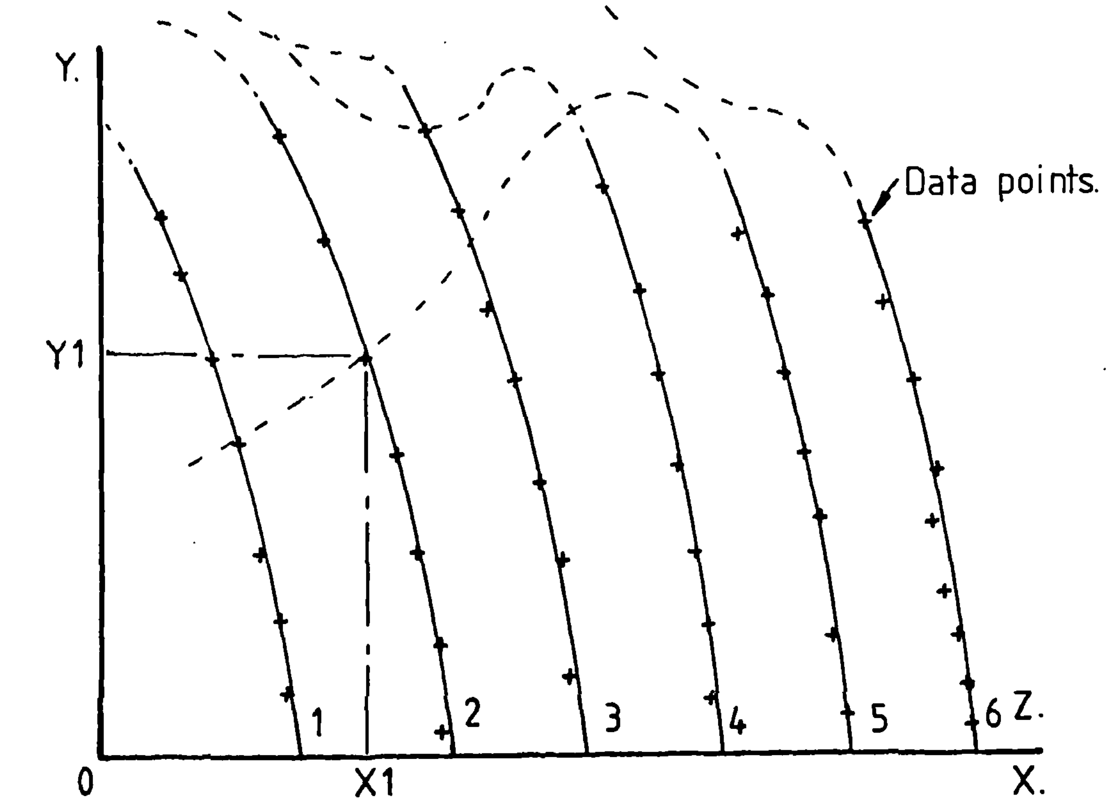


figure A9, Extrapolation of Curves.

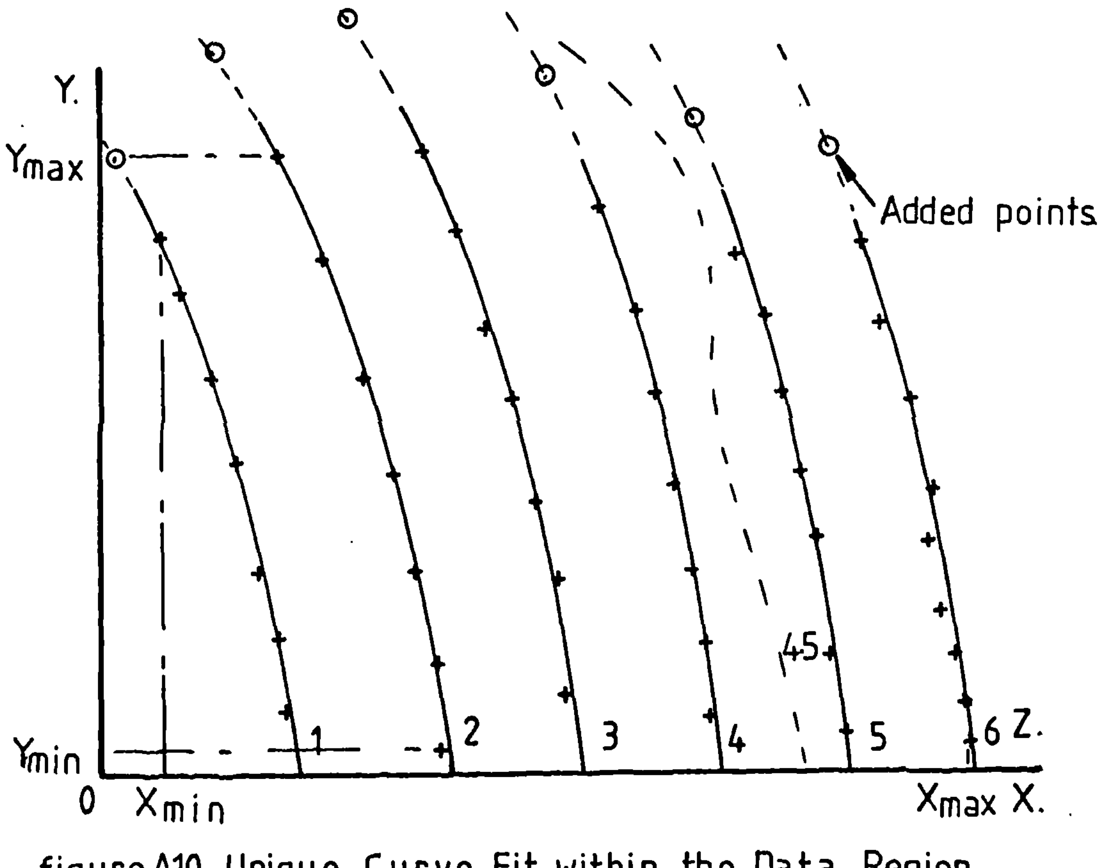


figure A10, Unique Curve Fit within the Data Region.

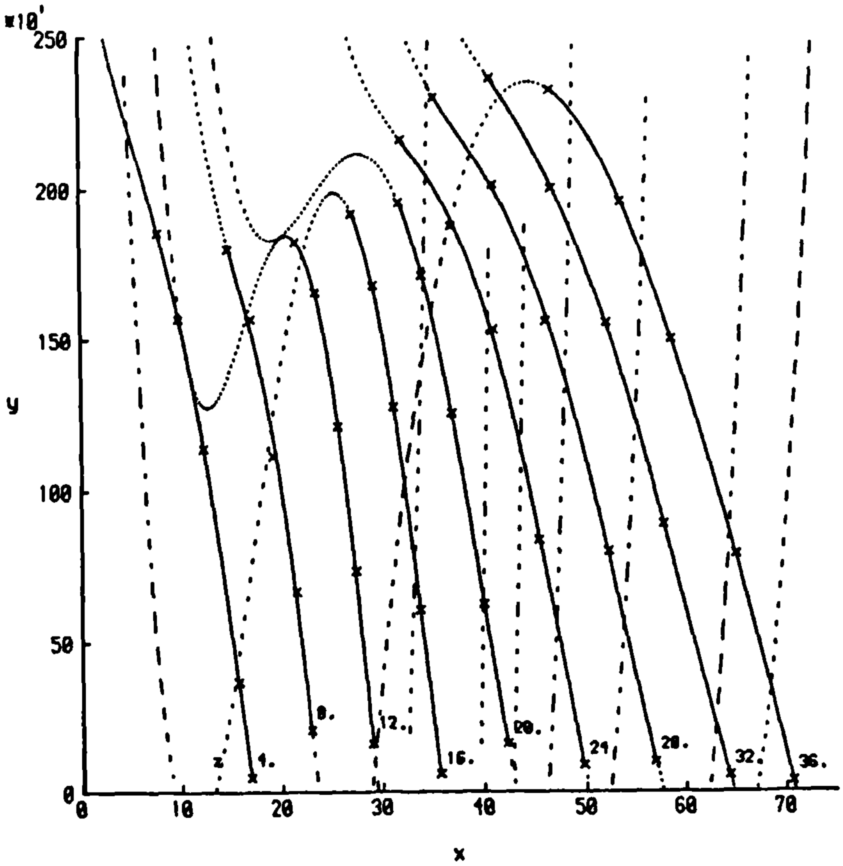


figure A11, Extrapolated Fan Performance.

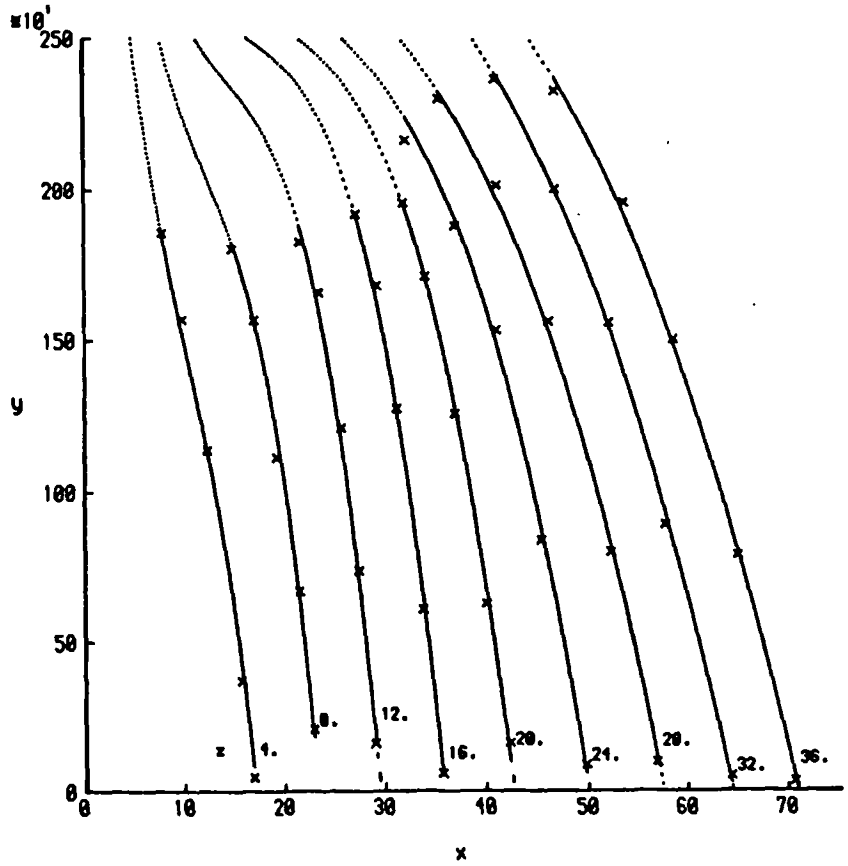


figure A12, Unique Curve Fit of Fan Performance.

A final factor which must be considered in applying the least squares curve fitting technique is the magnitude of the numbers in the curve fit. If the number of data points is large and the order of polynomial high then the numbers in the α and β matrices (equation 2) become too great and can result in precision overflow. This can be avoided if the variables are transformed to lie in the range -1 to 1. A means of transforming the variables which retains the precision of the original data is that suggested by Gill (1981):

$$Xs = \frac{2.X}{ub - lb} - \frac{lb + ub}{ub - lb} \qquad (8)$$

where Xs is the transformed variable and where lb is the lower bound and ub the upper bound of the variable x, which is to be transformed. Using expression (8), the variables x_i, z_i and Y_i at each data point can be transformed before curve fitting. The resulting polynomial coefficients then relate to the transformed variables and therefore when using the curve fit both the independent variables x and z must be transformed and the resulting value of the fit , f(x,z) transformed back to a meaningful order of magnitude.

Discussion and Conclusions.

The polynomial form of least squares curve fit lends itself to the modelling of HVAC component performance as the procedure can be formulated to fit a function of several variables and more importantly, a variable order of polynomial enables the fit to adapt to the characteristics of individual components. Curve fits of the type of response curve experienced in control schemes, could however benefit from the introduction of logarithmic and trigonometric functions in addition to the polynomial form of fit, as this would reduce the number of coefficients required to fit such curves.

Although the shape of HVAC component performance curves are usually uncomplicated, it is by no means always evident which order of polynomial gives the best fit to the data. The adaptation of the multivariate search method enables the order of polynomial to be automatically searched for the best fit to the data.

Although robust in its use, the search can find an order of polynomial which gives a poor fit where there is significant scatter in the data points. In such cases the search could possibly be improved by changing the search criterion from a normalised deviation to the sum of the squares of the errors between the fit and data, although this is yet to be investigated.

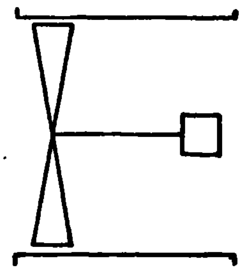
The importance of obtaining a unique curve fit is dependent not only upon the simulation methodology, but also the system simulated. A system configuration may be such that the arrangement of the components forces the solution to lie in the correct region of fit. For example, the inclusion of controls in the simulation of a variable air volume system, should ensure that the solution lies in the correct region of the fan characteristic. Where the system and solution procedure are simplistic, the curve fit can be modified to be unique within the variable bounds by adding data points outside the original data region before the data is curve fitted. Use of computer graphics has proved a valuable tool in this respect, as it allows a visual representation and understanding of the accuracy and behaviour of the fit.

Appendix B. <u>COMPONENT MODELS.</u>

The component models summarised in this appendix are only those which appear in examples throughout this thesis. Development of a data base of component models at Loughborough University is a task performed by several researchers. Authors of the performance models are referenced for each component. All cost, energy and constraint models have been developed additionally as part of this research and are attributed to the author of this thesis. The component models listed are:

B1 - Axial Fan.
B2 - Heating/Cooling Coil.
B3 - Centrifugal Chiller.
B4 - Duct Fitting.
B5 - Diverting Valve.
B6 - Controllers.

B1 - Axial Fan. (Author:J.A. Wright (1984)).



System Variables.

ma air mass flow rate.

Pi total pressure at inlet.

Po total pressure at outlet.

ti air temperature at inlet.

to air temperature at outlet.

ga air moisture content at inlet.

β fan blade angle.

Constants.

d impeller diameter.

n fan speed.

T hours in each time period of the load profile.

α maintenance charge coefficient.

Internally Computed Variables.

Ptf fan total pressure. $[(\gamma.\rho.(\pi.d.n)^2)/2]$.

Qr fan absorbed power. $[(\lambda.\pi^4.d^5.n^3.\rho)/8]$.

Cp specific heat capacity of air.

γ normalised fan total pressure.

λ normalised fan absorbed power.

ρ air density at inlet.

Performance Model.

Describing equations:

$$Ptf = (Po - Pi)$$

$$Qr = ma \cdot Cp \cdot (to - ti)$$

Energy Model.

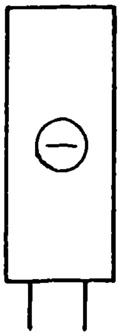
Direct term = $Qr \cdot T$ [absorbed power].

Cost Model.

Capital cost = function of d and β.

Maintenance cost = $Qr \cdot \alpha$

B2 - Heating/Cooling Coil. (Author: V.I. Hanby).



System Variables.

ma	air mass flow rate.
ta1	air on dry bulb temperature.
ta2	air off dry bulb temperature.
ga1	air on moisture content.
ga2	air off moisture content.
P1	air on total pressure.
P2	air off total pressure.
mw	water mass flow rate.
tw1	water return temperature.
tw2	water flow temperature.

Constants.

nrows	number of coil rows.
ncirc	number of water circuits.
width	coil width.
height	coil height.
ai	internal face area of the coil tubes.
T	hours in each time period of the load profile.
α	maintenance charge coefficient.
faaa	ratio of face area/air side surface area.
flfa	free flow area/ air side surface area.
C1,C2	Colburn factor constants.
f1,f2	friction factor constants.
rmet	coil metal thermal resistance.

Internally Computed Variables.

Cmin	minimum fluid capacity rate.
Cw	water side capacity rate.
eff	coil effectiveness.
f	friction factor.
G	mass velocity of the air.
h1	entering air enthalpy.
h2	exit air enthalpy.
shr	sensible heat ratio.
v	specific volume of air.
vw	specific volume of water.
fa	coil face area.
ntubes	number of water tubes.

Performance Model.

Describing equations.

$$ma \cdot (h2 - h1) = Cmin \cdot eff \cdot (tw1 - ta1)$$
$$ma \cdot (h2 - h1) = Cw \cdot (tw1 - tw2)$$
$$ga1 - ga2 = (1 - shr)\ (ta1 - ta2) / (2400 \cdot shr)$$
$$P1 - P2 = (G^2 \cdot v \cdot f) / (2 \cdot f1fa)$$

Energy Model.

Direct term = ma . (h2 - h1) . T [duty].

Extraneous term = (ma . v) (P2 - P1) . T [air loss].

Cost Model.

Capital cost = function of nrows, width and height.

Maintenance cost = (ma . v) (P2 -P1) . α

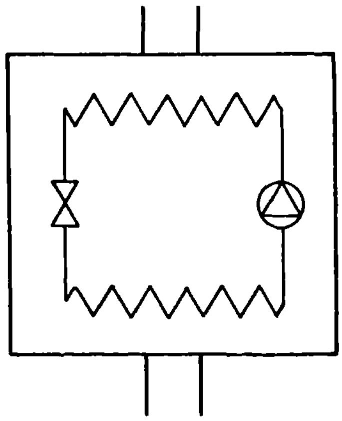
Constraint Model.

Face velocity = (ma . v) / fa

Circuits configuration = (ncirc - ntubes) / (1 - ntubes)

Water velocity = (mw . vw) / (ai . ncirc)

B3 – Centrifugal Chiller, (Author: P.R. Deering).



System Variables,

mwe	evaporator water mass flow rate.
te1	evaporator water flow temperature.
te2	evaporator water return temperature.
mwc	condenser water mass flow rate.
tc1	condenser water flow temperature.
tc2	condenser water return temperature.
x	control signal.

Constants,

c1	part load compressor power factor.
Qr	rated capacity of the chiller.
T	hours in each time period of the load profile.
α	maintenance charge coefficient.

Internally Computed Variables.

Qc	chiller cooling capacity (evaporating load).
W	compressor power.
Cp	specific heat capacity of water.

Performance Model.

Describing equations:

$$(x \, . \, Qc) = mwe \, . \, Cp \, . \, (te2 - te1)$$

$$x \, . \, (Qc + (c1 \, . \, W)) = mwc \, . \, Cp \, . \, (tc1 - tc2)$$

Energy Model.

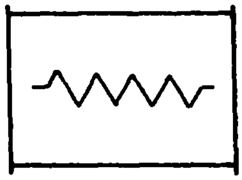
Direct term = W . T	[compressor power].
Extraneous term = Qc . t	[condensing load].
Extraneous term = (Qc - W) . T	[evaporating load].

Cost Model.

Capital cost = based on package component price list.

Maintenance cost = Qr . α

B4 - Duct Fitting. (Author: V.I. Hanby).



System Variables.

ma	air mass flow rate.
Pi	total pressure at inlet.
Po	total pressure at outlet.
ti	air temperature at inlet.
ga	moisture content at inlet.

Constants.

d	duct diameter.
K	pressure loss coefficient.

Internally Computed Variables.

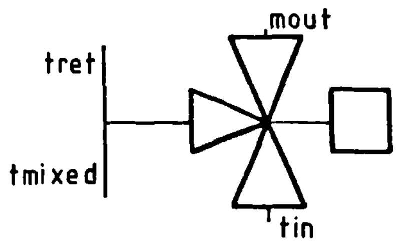
V	mean flow velocity in the fitting.
ρ	air density at inlet.

Performance Model.

Describing equation:

$$K \cdot \rho / 2 \cdot V^2 = (Po - Pi)$$

B5 - Diverting Valve. (Author: M.A.P. Murray).



System Variables.

x	control signal.
mout	output mass flow rate.
tin	water temperature at input to the valve.
tret	return water temperature.
tmixed	mixed water temperature.
mmax	maximum water mass flow rate.

Performance Model.

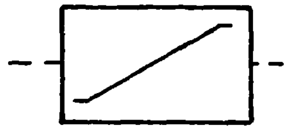
Describing equations:

$$mout = (1 - x) \cdot mmax$$

$$mmax \cdot tmixed = mout \cdot tret + mmax \cdot x \cdot tin$$

B6 - Controllers. (Author: V.I. Hanby).

Proportional controller.



System Variables.

cv controlled variable.

sp set point.

x output signal.

Constants.

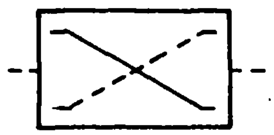
tr throttling range (symmetrically placed about the set point).

Performance Model.

Describing equation:

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

Signal Inverter.



System Variables.

x1 input signal.

x2 output signal.

Constants.

c offset.

m gradient.

Performance Model.

Describing equations:

If x1 = 0 then x2 = c

If x1 = 1 then x2 = m + c

else x2 = m . x + c

Appendix C. AN EXAMPLE OF COMPONENT UNDERSIZING.

The purpose of this exercise is to identify possible numerical indicators of component undersizing from which a constraint function could be formed. The occurence of undersized components in the system is indicated by the simulation algorithm failing to find a solution. Therefore, it is the values of the describing parameters of the simulation problem which have been investigated on failure of the solution algorithm. The parameters investigated are:

1. The sum of the component residual equations.
2. The largest value of the unsolved residual equations.
3. The number of unsolved residual equations.
4. The arc-variable values which are on their bounds.

The example system used to investigate these parameters, consists of a variable blade angle axial flow fan controlled by the action of a proportional controller, and signal invertor: system pressure is represented by a fitting attached to the axial fan figure C1. The parameter values have been compared for three sizes of fan the largest of which (size 112), is the only fan of the three capable of meeting the imposed load. The smallest of the three fans (size 90), is the worst selection and is least likely to meet the imposed load. Performance envelopes of the fans are represented by bounds on the fan blade angle and the proportional controller signal (table C1). Undersizing of the two smallest fans has been assured by selecting a flow rate and system pressure (represented by the controller set point), which lie outside the limits of the fans performance (table C3). To allow easier interpretation of the results, a single time period in the profile of exogenous variables has been used. System performance has been simulated using the GRG2 algorithm and scaled variable form of the Newton-Raphson algorithm. Although the largest of the fans is correctly sized, the value of the system parameters on completion of the simulation for this fan have been included in the results for comparison with the undersized component results. The formulation of a constraint function from the system parameters is discussed in section 7.5.

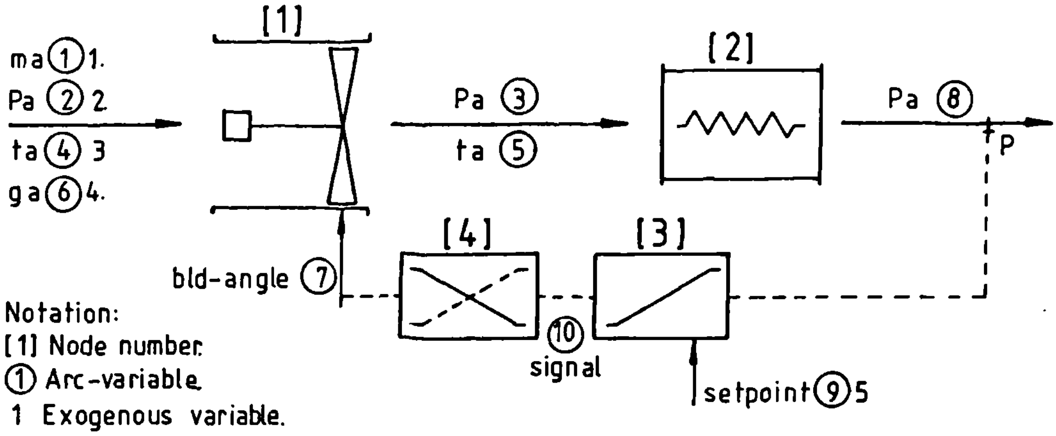


figure C1, VAV System.

Arc-variable	Initial Guess.	Lower Bound.	Upper Bound.
Pa ③	800	600	1200
ta ⑤	20	0	60
bld-angle ⑦	16	0	32
Pa ⑧	600	550	650
signal ⑩	0·5	0·0	1·0

table C1, Arc-variable Initial Guess and Bounds.

Exogenous Variable.	Value.
ma 1.	21·5
Pa 2.	0
ta 3.	5
ga 4.	0·002
setpoint 5.	600

table C2, Exogenous Variables.

Variation in the Sum of the Residuals.

The sum of the residual equation values is proportional to the degree of undersizing, table C3, ie: the less likely the fan is to meet the imposed load then the larger the sum of the residuals. The sum of the residual values should be zero when all the components in the system are correctly sized and the performance simulation has found a solution. However, in practice this is dependent upon the rules used by the solution algorithm to assess the convergence of the solution and therefore the sum of the residuals is often non-zero, as for fan size 112, table C3. If a zero value proves to be important for the development of a constraint function then this could be achieved by setting the residual values to zero when a solution is found.

Variation in the Unsolved Residual Equations.

Using the notation described in appendix B. the system residual equations are:

$(Po - Pi) - Ptf$	$= 0$	: 1.	Axial Fan.
$ma.Cp(to - ti) - Qr$	$= 0$	: 2.	
$K.\rho/2.V^2 - (Po - Pi)$	$= 0$	: 3.	Fitting.
$((cv - sp + tr/2) / tr) - x$	$= 0$	: 4.	Proportional Controller.
if $x1 = 0$: $c - x2$	$= 0$	: 5.	Signal Inverter.
if $x1 = 1$: $(m + c) - x2$	$= 0$	:	
else : $(m.x + c) - x2$	$= 0$	:	

The unsolved equation values, in table C4, are enclosed in brackets [] and the largest of these marked by an asterisk. Although the equation with the largest value differs between solution algorithms, its value for both is proportional to the degree of undersizing. The order in which the residual equations are solved is related to scaling (Murray, 1984), yet in this example there is a relationship between the equation having the largest residual value and the undersizing of the component. The equation with the largest value on failure of the GRG2 algorithm is the fan pressure residual, (equation 1.), which would suggest the fan is not capable of maintaining the pressure required.

Fan Size.	Sum of the Residuals.	
	GRG2	Newton-Raphson.
112	$1{\cdot}4\times10^{-5}$	$2{\cdot}7\times10^{-15}$
100	$3{\cdot}7\times10^{-2}$	$3{\cdot}9\times10^{1}$
90	$4{\cdot}7\times10^{-1}$	$1{\cdot}81\times10^{2}$

table C3, Sum of the Residuals.

Solution Algorithm.	Fan Size.	Residual Equations.				
		1	2	3	4	5
GRG2	112	$4{\cdot}8\times10^{-6}$	$9{\cdot}7\times10^{-6}$	$9{\cdot}7\times10^{-8}$	$-5{\cdot}4\times10^{-17}$	0·0
	100	$[-3{\cdot}7\times10^{-2}]^{*}$	$9{\cdot}1\times10^{-8}$	$1{\cdot}2\times10^{-15}$	$8{\cdot}3\times10^{-16}$	0·0
	90	$[-4{\cdot}7\times10^{-2}]^{*}$	$-3{\cdot}7\times10^{-17}$	$-1{\cdot}0\times10^{-8}$	$8{\cdot}3\times10^{-15}$	$-1{\cdot}3\times10^{-7}$
Newton-Raphson.	112	$5{\cdot}7\times10^{-11}$	$1{\cdot}1\times10^{-10}$	$-1{\cdot}0\times10^{-10}$	$-1{\cdot}5\times10^{-15}$	0·0
	100	$[-3{\cdot}7\times10^{-2}]$	$[-1{\cdot}0\times10^{-1}]$	$[-8{\cdot}3\times10^{-2}]$	$[3{\cdot}8\times10^{1}]^{*}$	0·0
	90	$[-4{\cdot}7\times10^{-1}]$	$[-6{\cdot}1\times10^{-1}]$	$[-2{\cdot}4\times10^{-1}]$	$[1{\cdot}8\times10^{2}]^{*}$	0·0

table C4, Residual Equation Values.

Solution Algorithm.	Fan Size.	Arc-variables.				
		3	5	7	8	10
GRG2.	100	1160·4	6·09	31·99	580·0	0·0
	90	1158·5	5·19	32·0	580·0	0·0
Newton-Raphson.	100	1159·2	6·19	32·0	578·4	0·0
	90	1152·8	5·81	32·0	572·8	0·0
GRG2 & NR	112	1173·3	6·16	21·8	592·7	0·32

table C5, Arc-variable Values.

Similarly, the equation with the largest value on failure of the Newton-Raphson algorithm is the proportional controller residual (equation 4), which suggests that the fan is unable to maintain the controlled variable, of duct pressure.

The number of unsolved equations varies between solution algorithms but in both is unaffected by the degree of under sizing.

The Arc-variables at their Bounds.

It might be expected that the arc-variable values which lie on their bounds on failure of the solution algorithm, would be those representing the limits of the component performance, ie: those which form part of the performance envelope. In the example. the arc-variable values which lie on their bounds, on failure of both the GRG2 and Newton-Raphson algorithms are the fan blade angle (arc-variable 7), which represents part of the fan performance envelope and the proportional controller signal (arc-variable 10). This suggests that the fan is operating at its maximum capacity as the solution has driven the controller signal and hence blade angle , to their limiting values. This characteristic could be used to identify the undersized components in the system, however its reliability could be influenced by the scaling of variables and order of solving equations and therefore would require further research to assess the reliability of this characteristic.

Appendix D. VARIABLES AND DIRECTORIES.

This appendix lists the main variables and directories used in implementing the optimised design procedure, together with the parameters which specify the size of problem manageable by the program. A detailed description of the function of the main variables and directories is given in chapter 4.

Variable types are indicated by DP for double precision, I for integer, L for logical and C for character, C*8 representing a character variable 8 characters in length.

Design Variables.

Variable:	Type:	Function:
COMFIL(maxnod,maxstp)	C*8	contains component record names for discrete variables.
COMNAM(maxnod)	C*8	array of component names.
DESVAR(maxdvr)	DP	vector of design variable names.
DVARNM(maxnod,3*mxdvr)	C*8	array of design variable names.
NDVAR	I	number of design variables.
STPVAR(maxdvr,maxstp)	DP	array of discrete data values.
VARDIR(maxdvr,maxdir)	I	directory of problem variables.

Bounds and Constraints.

Variable:	Type:	Function:
CONDIR(maxnod,mdrcon)	I	directory of constraint functions.
CONLB(mxdcon)	DP	vector of lower bounds on the constraint functions.
CONUB(mxdcon)	DP	vector of upper bounds on the constraints.
CONNAM(mxcon)	C*8	array of constraint names read from the component initialisation subroutine.
DCON(mxdcon)	DP	vector of constraint values.
DUMCON(mxcon)	DP	vector of constraint values used temporarily to return the function values from the component subroutines.
DVARLB(maxdvr)	DP	lower bounds on the design variables.
DVARUB(maxdvr)	DP	upper bounds on the design variables.
NCON	I	number of constraint functions defined in a given problem.
NCONI	I	number of constraint functions for a specified component.

Energy Models.

Variable:	Type:	Function:
ENGDIR(maxnod,mxeng)	I	directory of energy terms.
ENGDRC(maxnod,maxeng)	C*2	array of energy term fuel types and system model parameters.
ENGFUN(maxeng)	DP	vector of energy term values returned from the component subroutines.
ENGTYP(maxeng)	C*2	array of default fuel types and model parameters returned from the component initialisation subroutine.
NENGI	I	number of energy terms for a specified component.

Component Costs.

Variable:	Type:	Function:
COST(maxcst)	DP	vector of component costs returned from the component subroutine, COST(1) = capital cost, COST(2) = maintenance cost.
CSTCON(mcst,mcoe)	DP	array of cost data read from component data files and overwritten to SYSCST.
CSTFIL(maxnod/3)	L	array indicating the existence of a cost model for each component, .true. = model exists, .false. = no modedl.

CSTREC	C*8	name used to identify individual data records.
CSTTYP	I	indicator of the type of cost calculation required within the component subroutine, CSTTYP = 1 for capital cost, CSTTYP = 2 for maintenance cost, CSTTYP = 3 for both maintenance and capital cost.
SYSCST(msycst,mcoe)	DP	array of cost data for each node in the system.

<u>General Design Data.</u>

Variable:	Type:	Function:
BLDLIF	DP	value of estimated building life
FUELS(4)	DP	vector of fuel tariffs.
INTRST	DP	interest on borrowed capital.
PRIRAT(4)	DP	vector of primary energy ratios.
SRVLIF(maxnod/3)	DP	vector of the estimated service life of the components.
TIMPRD	DP	time assigned to one division in the load profile.

Dimensioning Parameters.

Parameter:	Value:	Function:
maxcst	3	maximum number of calculated cost values per component.
maxdir	2*mxdvar+3	variable directory row length.
maxdvr	40	maximum number of design variables.
maxeng	3	maximum number of energy terms per component.
maxnod	30	maximum number of components.
maxstp	20	maximum number of discrete values per variable.
mcoe	50	maximum number of cost data ceofficients in one data set.
mcst	2	maximum nuber of cost data sets per component.
mdrcon	2*mxcon+1	constraint directory row length.
msycst	20	maximum number of cost data sets held in the cost data array SYSCST (strictly maxnod*mcst).
mxcon	4	maximum number of constraints for each component.
mxdcon	30	maximum number of constraints assigned to a given problem (strictly maxnod*mxcon).

mxdvr	5	maximum number of matching dimensions.
mxeng	maxeng+1	energy term directory row length.