GAS-FIRED RADIANT HEATERS

CHRISTOPHER DAVID ZIESLER

Doctor of Philosophy

THE UNIVERSITY OF ASTON IN BIRMINGHAM

February 1991

THESIS SUMMARY

The University of Aston in Birmingham

Christopher David Ziesler

Gas-Fired Radiant Heaters

Doctor of Philosphy

1991

This dissertation covers four areas of particular interest for the successful application of radiant heating in industrial environments. In it the author tackles the problem of how to predict the thermal comfort produced both by single heaters and also systems of heaters; proposes a method for modelling the mechanisms by which heaters interact with the buildings in which they are installed, in the static and dynamic cases; explores techniques for measuring the radiation produced by heaters; and presents experiments concerned with finding the temperatures and power balances prevailing during normal operation.

It is contended that, whilst the generally accepted guides for sizing and operating space heating plant are a good first approximation, there are intrinsic subtleties arising from the fact that the primary mode of heat transfer in this instance is radiative. These nuances are concerned with how best to maximise the heat transfer from the heat source to the heated object; the placement of heaters within a system; and an assessment of the various techniques and strategies involved in controlling a radiant heating system.

The conclusions reached are that: if sized and controlled correctly radiant heating systems offer considerable operational advantages over other types of space heating systems in certain applications, in terms of both economy and controllability. The efficacy of radiant heating systems is affected primarily by the control strategy implemented; secondarily, by the structure of the building into which it is installed; and only marginally by all other factors.

KEYWORDS

Radiant Heating

Thermal Comfort

Mathematical Modelling

Monte Carlo

For Kaja, Thomas, and Samuel.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank Dr John Maund and Mr Jeff Pearson, my Supervisors at Aston and British Gas respectively. They have seen this project through from start to finish; and without their unstinting support, advice, and encouragement this dissertation would never have been completed. I would also very much like to thank Dr Eric Smith for giving so freely of his time in the latter stages of the thesis preparation; it would have been an immeasurably poorer work without his keen criticism.

I must express my gratitude to two friends with whom have I communicated, more often than not, via the electronic mail system in the last four years, to Dr Malcolm Howe, of Oxford University, for his algorithms, knowledge of Fortran, patience, and expertise in all matters physical; and to Dave Stops, of Aston's Computing Service, for his continual help, and also his laconic style and wit. I would also like to express my appreciation to all my co-postgraduates at Aston, for their friendship, help, and unending cups of coffee.

Finally, especial thanks are due to British Gas whose funding made this work possible, and specifically to all the staff of the Midlands Research Station with whom I have come into contact during my Industrial Scholarship, and with whom I now work.

LIST OF CONTENTS

LIST OF FIGURES

Chapter 2

Chapter 3

Chapter 4

Chapter 5

Chapter 6

Appendix B:

LIST OF TABLES

Chapter 2

Chapter 3

Chapter 4

Chapter 5

Chapter 6

1 An Introduction to Radiant Heating

1.1 Background

The value of the industrial and commercial comfort heating market in Great Britain has been estimated to be 450 million GJ p.a., or 4000 million therms (British Gas, School of Fuel Management, 1983). This implies that at today's prices some £1,470 million a year is expended in an attempt to keep the occupants of industrial and commercial premises comfortable.

An estimate of the annual market for all new heating units in the United Kingdom, according to a BSRIA survey of suppliers (BSRIA, 1989), is £57.9 million. Of this total the share of radiant tube heaters is £8.8 million (15%), radiant plaques is £3.1 million (5.4%), and electric heaters is £4.1 million (7.1%); these sales represent 24600, 14700, and 30000 units respectively. The overall tendency is towards increased use of radiant heaters in preference to warm-air. This is reflected by the fact that 60% of new all-radiant systems displace another type of heating. This should be compared with only 30% of warm-air systems being "displacement" systems.

It is difficult to assess the environmental impact of such quantities with anything but a broad-brush picture. If a typical gas-fired heater produces 66 ppm of NO_x (wet) at 10% excess air in the flue gas, a figure not unreasonable for such devices, then this implies that radiant heaters alone produce 9000 tonnes of NO_x p.a. Likewise the combustion of 215 million GJ equivalent of methane releases 10 million tonnes of carbon dioxide into the atmosphere each year.

It is against the backdrop of these sorts of figures that the

work described in this thesis, which was aimed at improving heater and system efficiencies, has been undertaken. It is undoubtedly the case that all countries will have to reduce their carbon dioxide, NO_x, and SO_x emissions. The only uncertainty, in this country, is whether this will be achieved by direct legislation or the less direct "carbon tax". In either case the need for improved efficiency is unquestionably one of the major design requirements of any innovative heating system.

There is another related problem concerning contemporary trends in building design. There is a current tendency towards reducing energy consumption by reducing heat loss through the fabric and the ventilation heat loss by the application of more stringent building standards. This has the direct effect of reducing the plant size required to heat these "low-energy factories". This trend has a direct bearing on the design of radiant heating systems. When a smaller number of heater units are used to cover a given area the spread of radiation produced from an individual heater must clearly be different from a system with a higher unit density. A design tool to assist in assessing novel speculative designs was, in the light of this development, an obvious necessity.

1.2 Varieties of Radiant Heater

The broadest definition of a radiant heater is any heater which transfers a significant proportion of its heat output to its environment by means of radiation. Many heaters which would be classed as radiant heaters by this definition also have, in addition, a substantial convective component to their heat output. The actual model names and specific manufacturers are not named, rather a coding scheme is employed whereby a heater under discussion is denoted as being either: an IRH - Industrial Radiant Heater, ie a tube heater; a PLAQ - ceramic plaque heater; or a QLL - Quartz Linear Lamp. The specific model of heater is then given by the numerical suffix, eg IRH_1 is a different model to IRH_2 and so on. The three categories of interest to this dissertation are:

1) Gas-fired Tubular Radiant (IRH) Heaters

These consist of: a burner unit; a length of tube of cylindrical cross-section, which may be either straight or have a U-bend, which serves a dual purpose as a combustion chamber and emitter; a fan attached to the other end of the tube which extracts the products of combustion and expels them from the tube; and some form of metal reflector cowling which is placed over the top of the tube to direct the radiation to where it is required and also to prevent excessive heat loss by convection from the hot tube. The temperature of the tube varies greatly from the burner to the exhaust but an average value for a tube might be 400 °C. The rated inputs vary from around 13kW up to 60kW.

2) Gas-fired Plaque (PLAQ) Heaters

These consist of: a perforated ceramic plaque through which the combustion mixture passes before burning on the surface of the plaque itself; a wire mesh, or set of cylindrical bars, which sit adjacent to the surface of the plaque and serve to retain the flame; and a metal reflector which helps to direct the radiation in a similar manner to that used with tubular heaters. This type of heater has a typical plaque temperature of 800 °C, and they vary in size from 5-45kW input.

3) Quartz Linear Lamp (QLL) Heaters

A more recent addition to the radiant heating market are the electrically powered Quartz Linear Lamps. All the designs for QLL heaters are based around a heating lamp, which is a tungsten filament mounted in an inert atmosphere of argon plus a halogen. These "lamps" were originally developed for the "Halogen-Hob" electric cooker but have subsequently been put to other uses. The designs usually consist of the tungsten filament lamp surrounded by a filter which is intended to limit the amount of visible light emitted. This is then mounted in a metal parabolic reflector to focus the radiation and mounted either singly or in multiple units. The temperature of the tungsten when operating is 2400K. The lamps are usually either rated as 1.5 or 2.5 kW input, and therefore the range of heater sizes is determined by how many lamps are mounted together. Three-lamp heaters giving 4.5kW input are common, whereas heaters with an input over 10kW are rare.

1.3 The Objectives and Structure of the Dissertation

The aims of the work can be considered to be encompassed by four main subject areas.

1) To develop a computer-based model capable of predicting the performance of a single heater and assisting in the improvement of the design of reflector profile.

2) The investigation of the thermal interactions between a system of heaters and the building in which the system is installed, which should ultimately be expressed in a computer-executable form.

3) The construction and calibration of a radiometer specifically designed for measuring the irradiance produced by large, ie tubular, radiant heaters over the area of occupation beneath the heater. This should then provide evidence concerning the reliability of the mathematical model.

4) The gathering of reliable physical data from a variety of heaters. In particular this should include accurate information concerning the working emitter temperatures of heaters under realistic operating conditions.

The five chapters present the work performed in a logical manner, dealing in turn with: the Monte Carlo model; room simulations; the control of radiant systems; radiometry; and thermometry.

The appendices contain all the listings quoted in the text, plus some other items, which were deemed to be important but to which reference was not essential within the main body of the text. The system of footnoting is that a bold superscript number in the text refers to the end of the chapter in which the footnote appears.

1.4 Nomenclature

The nomenclature used in this dissertation has followed the guidelines laid down by the British Standard Institute (BSI, 1971) and the paper by D.A.McIntyre (McIntyre, 1989). A brief precis of the relevant terms is given here:

Radiant Quantities:

Radiant Energy: Energy which is emitted, transferred or received in the form of radiation (J).

Radiant Flux: Power which is emitted, transferred, or received in the form of radiation (W).

Radiant Efficiency: Ratio of the radiant flux emitted to the input power.

Irradiance: Flux received by a plane surface normalized to unit area (W/m^2). It is specified at a point on a surface. It is also referred to occasionally as the flux-density. It will depend upon the orientation of the test surface with respect to the emitter.

Spherical Irradiance: Flux incident upon a test sphere of unit area (W/m^2). Although this measure has the same units as irradiance it is independent of orientation due to spherical symmetry.

Emissivity: A measure of how well a body radiates compared to an ideal "black-body" radiator. In general it will depend upon the emitter temperature, the wavelength of the radiation, and the angle at which the radiation is emitted. In this dissertation all emissivities used assume that the emitters to which they refer are "diffuse, gray bodies", ie that the value of the emissivity does not depend on either the wavelength or the angle of emission, but may depend on temperature.

Subsidiary Quantities:

Black Globe Temperature: The equilibrium temperature of a black globe thermometer suspended in the test plane (K or °C).

Mean Radiant Temperature: The temperature of a black enclosure, surrounding the test point, which would have the same radiative interchange with the test point as does the actual surroundings (K or °C).

Dry Bulb Resultant Temperature: An index of thermal comfort used by Chartered Institute of Building Service Engineers (CIBSE) for assessing the subjective warmth of the thermal environment. It is defined to be:

$$T_{res} = 1/2.\ T_{air} + 1/2.\ T_{mrt} \qquad (1.1)$$

and can be seen to be a weighted average of the air and mean radiant temperatures (CIBSE, 1986).

Environmental Temperature: A temperature used by CIBSE for assisting in the assessment of the heat load of a building (CIBSE, 1986):

$$T_{env} = 1/3.\ T_{air} + 2/3.\ T_{mrt} \qquad (1.2)$$

2 The Monte Carlo Model

There are numerous ways to solve the integro-differential equations governing radiative heat transfer. The first, analytic solution, is applicable only in simple cases where there is a high degree of geometrical symmetry which has the effect of reducing the dimension of the integrals to manageable proportions. Of other methods, numerical integration and more specifically the Monte Carlo technique, are more widely applicable. The Monte Carlo method may be regarded as a specialized form of numerical integration.

The Monte Carlo method was originated by Ulam and Metropolis, to deal with the complex integrals evolving from the description of the passage of neutrons through matter (Ulam and Metropolis, 1959), although since then it has found applications as diverse as stochastic ray-tracing and Queuing Theory. Its underlying *modus operandi* has been nicely summarised as follows:

> The Monte Carlo Method is a method of solving various problems in computational mathematics by constructing for each problem a random process with parameters equal to the required quantities of the problem. The unknowns are determined approximately by carrying out observations on the random process, and by computing its statistical characteristics which are approximately equal to the required parameters.
> (Shreider, 1964)

The advantages of Monte Carlo are that it has an intuitive appeal stemming from the close correspondence of the physical analogy and the model, is inherently extensible, and offers shorter computation times than numerical integration, at least for medium accuracy problems. Amongst its disadvantages are that it is dependent upon the availability of a good random number generator, and also that it converges only as the square root of the number of trials, so that for highly accurate work it becomes slower than numerical integration. It

can also be difficult to check the results for self-consistency. Ward and Tucker concluded from their comparison of numerical integration and the Monte Carlo method:

> The Monte Carlo technique is a convenient method for determining direct exchange areas in enclosures. However, unless a very large sample beam size is used, some of the exchange areas calculated by this technique will be considerably less accurate than those obtained by numerical integration with a fine grid size.
>
> (Ward and Tucker, 1987)

2.1 The British Gas Model

A British Gas paper outlines a model, based on the Monte Carlo technique, for calculating the total flux output from a tubular radiant heater. The following brief description is taken from the paper:

> A Monte Carlo simulation method is used. The model simulates random rays emanating from the tube. The starting point of each ray is chosen at random. The initial direction of the ray is also chosen at random but a non-uniform probability density function is used to account for the cosΘ term for the ray intensity *(Butler actually meant flux, CDZ)*.
> The initial intensity *(flux)* of the simulated ray is determined by the tube temperature at the starting point. The simulated ray is reflected specularly if and when it hits either the reflector or the U-tube. For the reflector this is exact. In the case of the U-tube this is an approximation since the reflection is in fact diffuse. However, the error introduced should be small due to the high emissivity of the U-tube and the random nature of the rays.
> If the ray leaves the heater, possibly after one or more reflections, in a downward direction, it contributes to the heat radiated downwards from the heater. If it leaves in an upwards direction it does not contribute.
> ... To calculate the radiation propagated in a specific direction or falling on a given surface it would be necessary to alter the model. A test would be included to check whether a ray leaving the heater falls into the required category. The smaller the surface the greater the number of rays that would need to be simulated to achieve convergence. (Butler, 1982)

The algorithms provided by this paper include: the generation of the ray from the source, i.e. the tube; the intersection of the ray with either of the tubes, the semi-torus or the reflectors; and the summing of the downward flux to produce a total flux output from the unit.

Fig.2.1 shows how the temperatures are assigned to various points on the tube, whilst Fig.2.2 shows a cross section of the heater used to produce the results in Table 2.1.

Several points should be made at this stage: most importantly, that this is a hybrid model in the sense that a crucial part of its

input, i.e. the tube's temperature field, is derived from experiment. No attempt has been made to derive these temperatures theoretically. To a large extent then, the accuracy of the model is dependent upon the validity of the temperatures supplied.

In its original state, the only figure of merit provided by the program was the total flux output. When the temperatures were taken from a real heater then this figure, along with the known calorific input, was used to find the efficiency of the heater as defined by the British Standard for plaque heaters (British Standards Institute, 1962).

Also, the original model did not provide any means of assessing:

1) the uniformity of the radiation produced by the unit, and as a corollary;

2) how a system of heaters might be designed to even out the radiation over a larger zone than would be heated by an individual unit;

3) how the heater might interact with its environment as opposed to radiating into free space;

4) and lastly how the radiation produced by the heater might be related to important comfort indices such as mean radiant temperature.

These were the considerations that prompted further development of the model into a more widely applicable tool for predicting the effect of radiant heaters on the thermal environment.

Fig.2.1: Determination of Tube Temperature

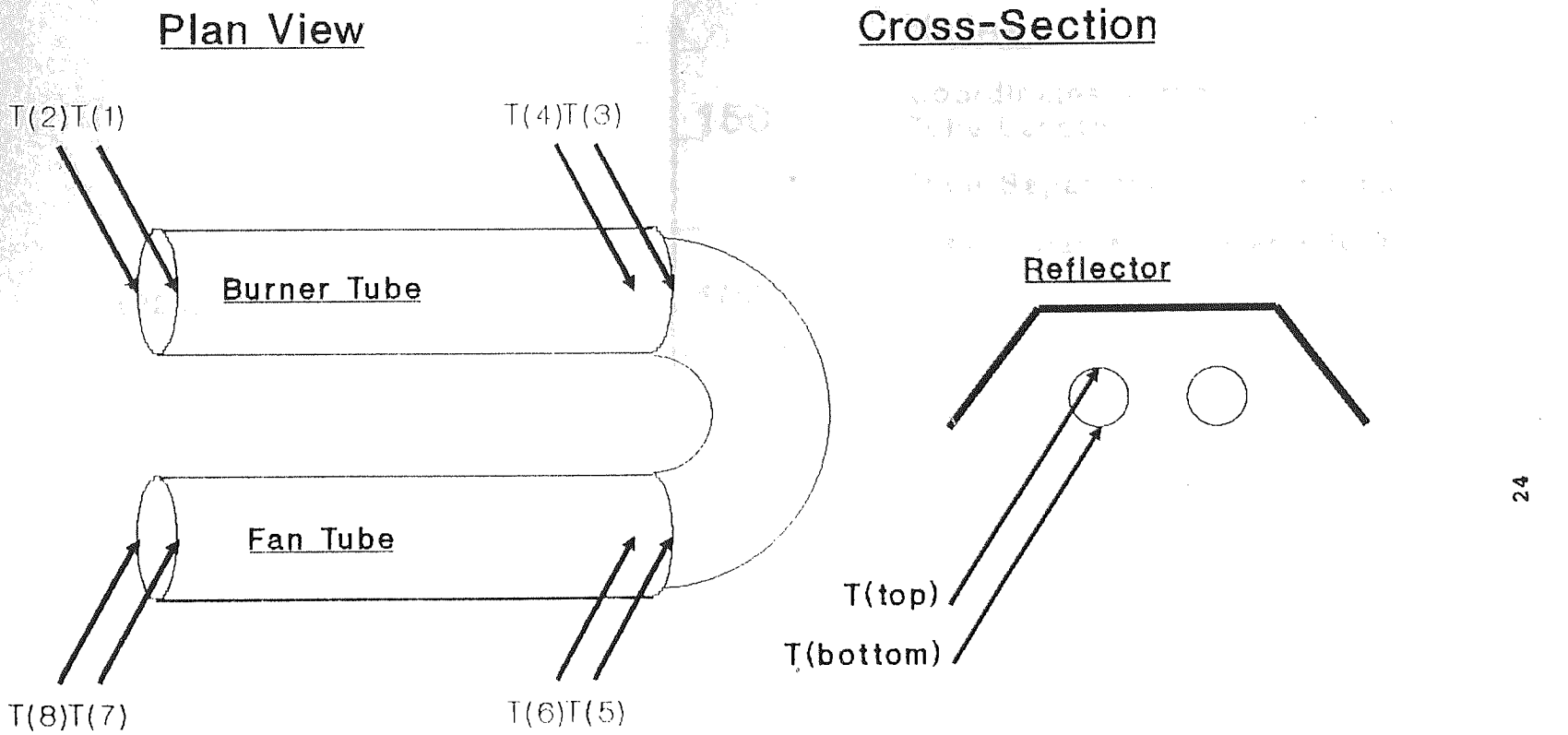
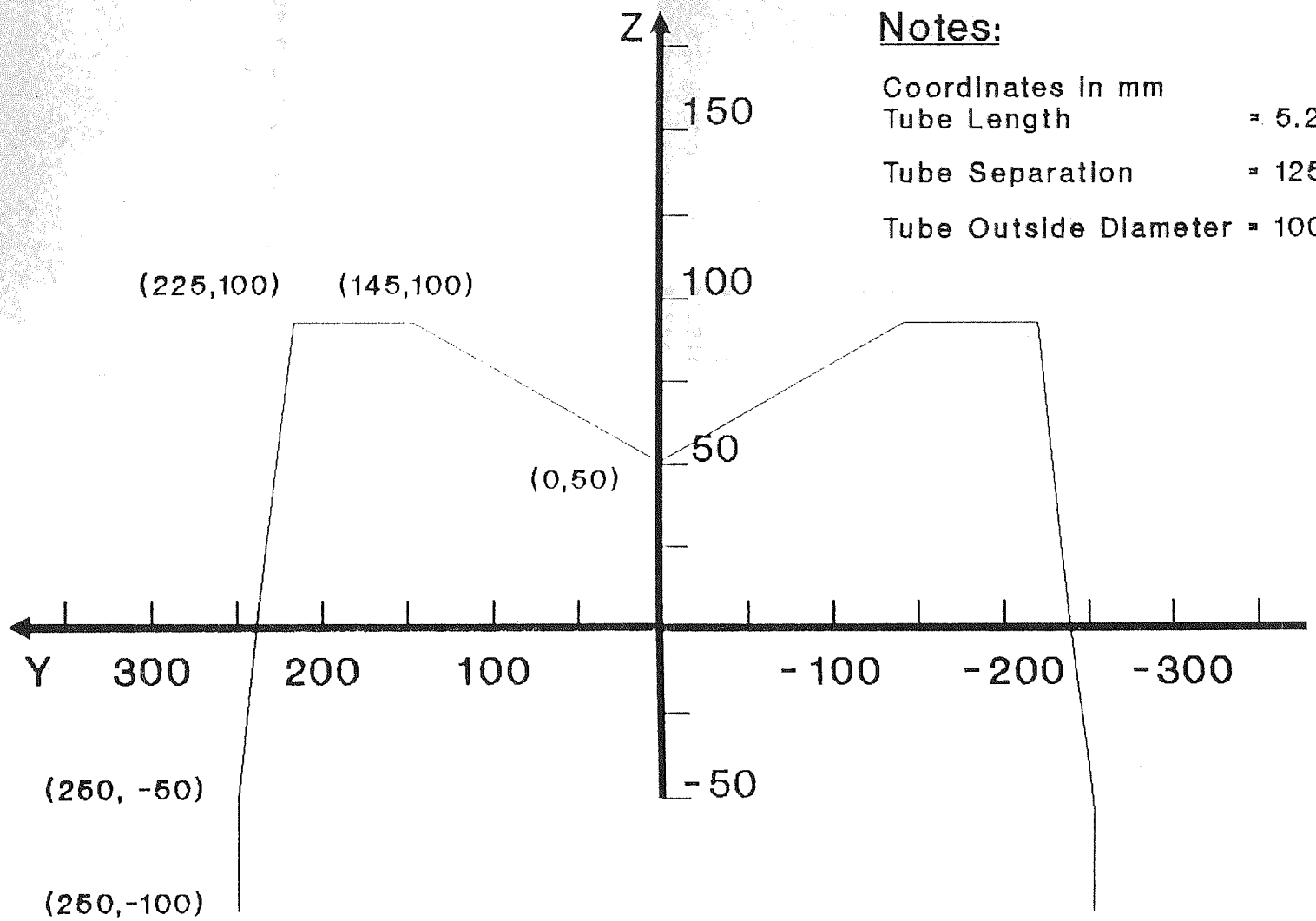


Figure 2.2: Profile of IRH_1



Notes:

Coordinates in mm
Tube Length = 5.20m
Tube Separation = 125mm
Tube Outside Diameter = 100mm

Table 2.1 Sample Output of Model in Original Form

Results of Original Model:

IRH model: IRH_8.

Number of reflector surfaces = 3
Number of rays computed = 3000

Temperature profile of tube

Position	1	2	3	4	5	6	7	8
Temperature (K)	750.0	720.0	590.0	560.0	590.0	560.0	510.0	480.0

Percentage of Run (%)	Total Flux (W)
4	13627.5
8	13929.7
12	14146.1
16	14121.0
20	14194.8
24	14309.4
28	14186.0
32	14111.3
36	14178.0
40	14184.4
44	14254.3
48	14246.6
52	14167.4
56	14201.8
60	14285.7
64	14286.1
68	14268.3
72	14288.9
76	14321.4
80	14287.9
84	14324.6
88	14329.3
92	14346.9
96	14371.7
100	14347.5
Input:	16131.3

OBSTACLE No.		Number of Hits
(U-tube)	-3	72
(Tube 2)	-2	35
(Tube 1)	-1	40
(No hit)	0	1129
(Surf 1)	1	493
(Surf 2)	2	1085
(Surf 3)	3	471

2.2 Testing the Butler Model.

A detailed account of the testing of the Butler model comprises a large section of the author's first year report (Ziesler, 1988). A summary is given below.

2.2.1 Testing the Pseudo-Random Number Generator

The most readily available Pseudo-Random Number Generator was that provided in the NAG (numerical algorithms) Library. It was felt that this was likely to be a better algorithm than the equivalent Fortran intrinsic function. In order to evaluate it as thoroughly as possible, several tests derived from "Empirical Tests on a Pseudo-Random Number Generator" were applied to the NAG routine (Smith, Green and Klein, 1959). The tests rely on detecting the correct recurrence frequencies of a given digit in a given position of a pseudo-random number. The results from one of these tests, the first order chi-squared, are shown in Table 2.2 and demonstrate that the PRNG is satisfactory at least to this level.

Table 2.2: Chi-Squared Test on the NAG PRNG

Position	1st	2nd	3rd	4th	5th
Digit					
0	102	99	88	96	89
1	99	103	115	108	101
2	105	107	102	112	105
3	79	103	106	118	102
4	115	102	92	103	88
5	102	108	90	76	100
6	90	92	103	97	99
7	91	106	104	81	94
8	110	82	99	105	105
9	107	98	101	104	117
Chi-Square	10.	5.6	6.0	15.	6.5
Prob(x)	.33	.78	.74	.80	.69

Test carried out on 10 sets of 1000 pseudo-random numbers.

2.2.2 Considerations of Speed and Accuracy

The time constraint placed upon the program was that it should be possible to simulate a large number of different reflector geometries or temperature profiles in a matter of hours rather than days. In practice this meant that the program had to run to completion in two or three minutes. Various test data files were created, with differing numbers of rays and reflectors, for the program to process.

Table 2.3: Summary of Tests for Speed and Accuracy

Results from original model operating on data files LNUM.IPT;1-6 and HNUM.IPT;1-6.

Filename	L1	L4	H1	H4	L2	L5	H2	H5	L3	L6	H3	H6
Number of Surfaces	1	1	1	1	3	3	3	3	4	4	4	4
NRAYS	300	1000	3000	10000	300	1000	3000	10000	300	1000	3000	10000
flux out (kW)	10.0	10.4	10.4	10.5	14.4	14.2	14.3	14.3	13.9	14.0	14.0	14.0
Efficency (%)	62.6	65.6	65.8	65.8	90.6	89.5	90.0	89.9	87.7	88.4	88.4	87.9
run time (s)	8.9	40.2	101.	393.	13.0	50.5	111.	437.	17.1	56.9	130.	510.
time/NRAY (ms)	29.7	40.2	33.2	39.3	43.2	50.5	37.0	43.7	57.1	56.9	43.2	51.0
std dev (W)	61.1	38.2	58.2	15.7	72.8	32.5	25.6	13.6	72.3	32.7	10.7	11.4
accuracy (%)	0.61	0.37	0.44	0.15	0.50	0.23	0.23	0.09	0.51	0.23	0.08	0.08

The table demonstrates two features of the model: that each additional reflector adds about 4-8ms per ray to the calculation time, and that the accuracy of the answer was inversely proportional to the square root of the number of rays used. The results of this test showed that a self-consistent accuracy of better than 0.25% could be obtained using 10,000 or more rays, in a time of two minutes. This, then, was the minimum number of rays adopted for subsequent runs.

2.2.3 Further Tests

Two further tests were carried out on the model in its original form, both aimed at increasing the certainty that the specified algorithms were behaving as expected.

The first test involved keeping a history of all of the intersections of specific rays which had been ejected in known and well-defined directions and whose trajectories had been calculated beforehand. This test proved that all of the intersection algorithms were working correctly.

The other test involved calculating the expected total flux from the tube, using a summation over 90,000 area elements and using this nominal maximum value to ensure that the program was not giving any spuriously large outputs. This test was carried out to check the flux calculating algorithm. Like the previous check, this test gave acceptable results.

2.3 Improvements to the Butler Model

Having thoroughly tested the simulation and run a large number of test cases which provided useful, if limited, information, the next stage of development was to improve on the shortcomings of the original model. Three major improvements which were achieved during the initial stages were:

1) to allow the use of more than 8 temperatures;

2) to generate irradiance plots as a function of position under the heater;

3) and also to relate the spatial output to comfort indices.

These improvements were all vital if one of the original aims of the project, the provision of better information on system design, was to be tackled successfully.

2.3.1 Temperature Determination

In the algorithm used by the original model the temperature of the point of ejection of a ray was calculated using a linear interpolation between eight points on the tube (see Fig. 2.1). Even a cursory inspection of the temperature profiles in Appendix 2 will show that this assignation of temperature using only eight points is not satisfactory.

The first advance in this area was to allow for any number of temperatures to be used with the proviso that they were taken only from the bottom and top of the tube. This allowed the use of more accurate temperature data, particularly that of Maund (1979).

It will be shown in later sections how this method has been further improved upon, and also how a more sophisticated means of

gathering temperature data has been found.

2.3.2 Spatial Information

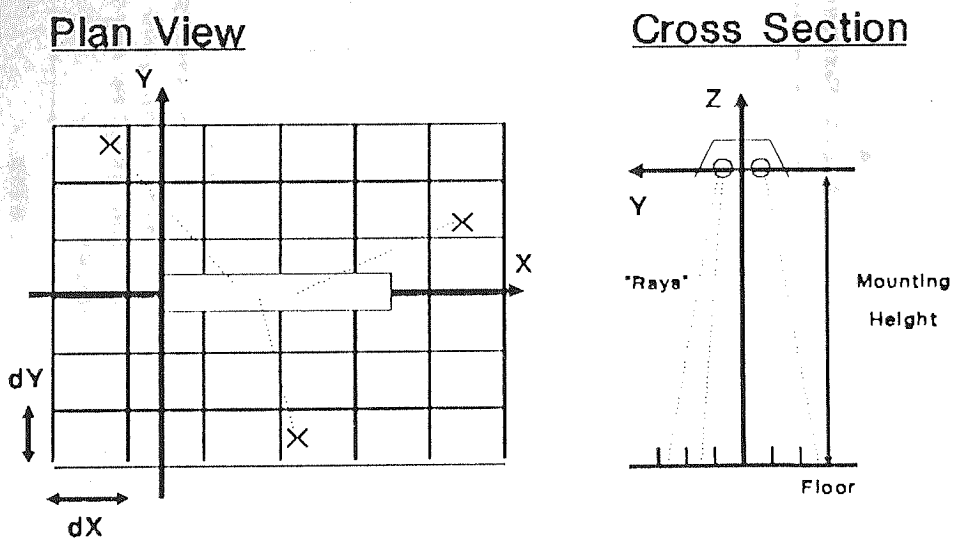
The model produced only one figure of merit for a heater, the efficiency. For purposes of systems design it was therefore important to find a means of relating not only the total heat output of a heater, but also the detailed spatial distribution of this output to the input parameters.

The flux densities over a zone were calculated by using the ray-plane intersection algorithm to find where the ray hit the detector plane (see Fig. 2.3). Then the coordinates of the intersection were used to increment array elements to keep track of the flux-density pattern. These arrays were in turn written to files for subsequent display in graphical form. The UNIRAS suite of graphics programs was used to produce all of the contour and isometric plots that follow.

In order to extend the model's generality, having successfully modelled tubular radiant heaters, it was necessary to reconfigure the program to enable it to simulate both gas-fired plaque heaters, and also electrically-powered Quartz Linear Lamps (QLL). These latter radiant heaters are of especial interest to British Gas insofar as they are marketed in direct competition to the gas-fired heaters.

There were several changes that had to be made to various modules to achieve the above objective. The relation between the three programs and their constituent modules are shown in Fig.2.4; whilst the geometries of the plaque and QLL heaters are shown in Fig.2.3. Full listings of the three programs are given in Listings 2.1, 2.2, and 2.3 in Appendix 1.

Fig.2.3: Visualization of Detector Algorithm



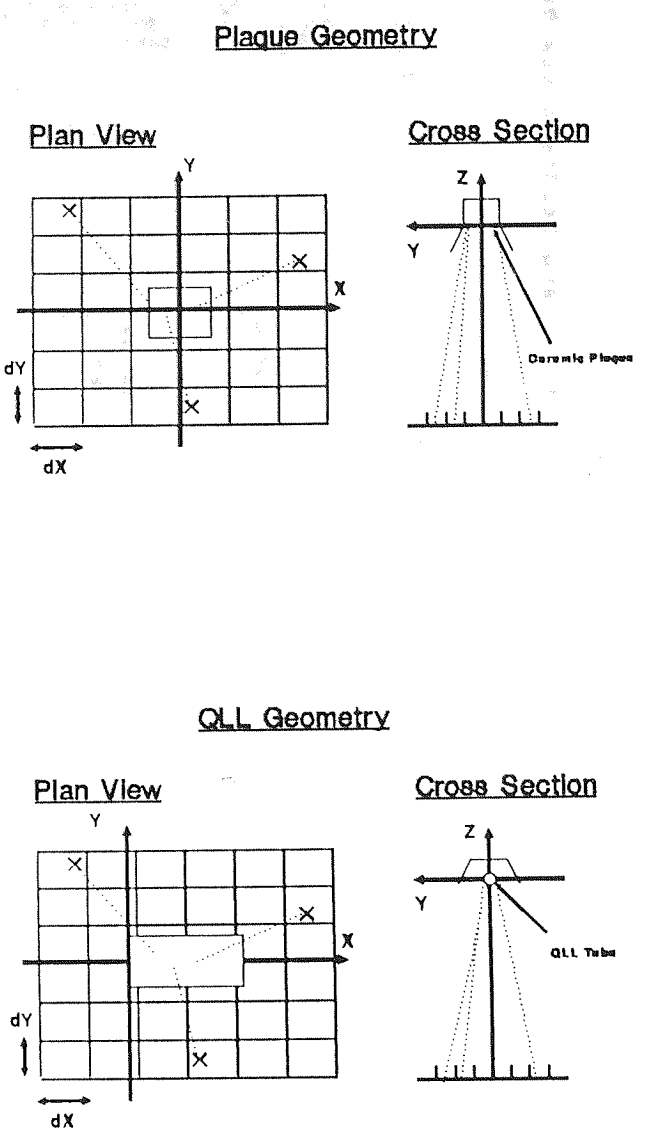
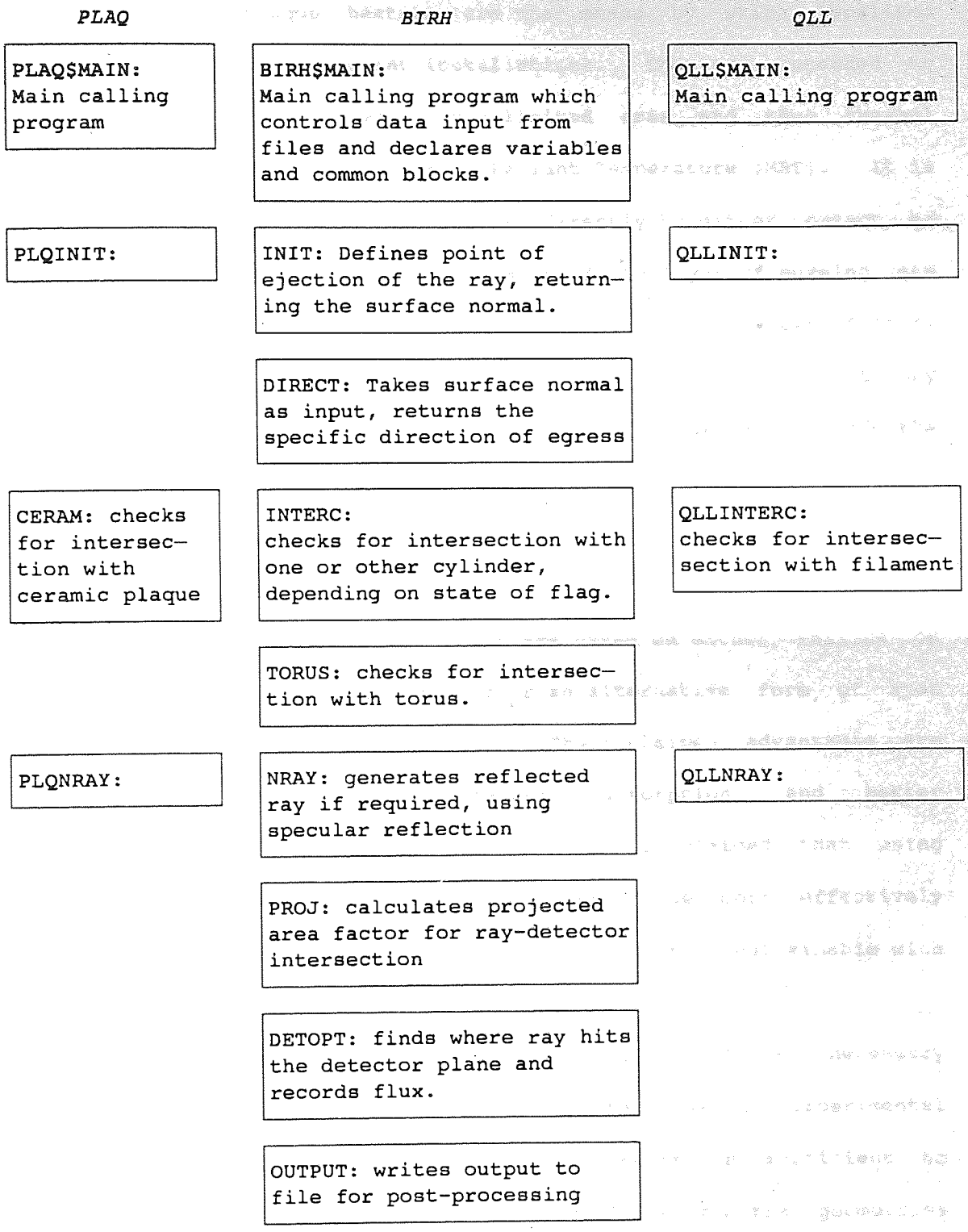


Figure 2.4: Inter-relation between Modules and Programs



NOTE: Full listings to all of these subroutines and programs are given in Appendix 1.

2.3.3 The Plaque Model

Ceramic plaque heaters are the means by which localised heating is achieved in radiant installations. They are intended to provide high flux-densities over a limited area and thus thermal comfort by directly raising the Mean Radiant Temperature (MRT). It is difficult to measure plaque temperatures directly by either contact or non-contact methods, principally because of the layer of burning gas and combustion products on the plaque. This layer can falsify emissivity estimates if non-contact methods are employed, whilst any attempt to measure the temperature directly has to take account of the cooling of the plaque by the probe.

2.3.4 The QLL Model

These heaters, all of which are based on either Philips or Thorn Quartz Linear Filaments, offer an alternative form of spot heating to gas-fired plaque heaters. Their claimed advantages are higher efficiencies, lower atmospheric absorption, and better controllability. In addition to this it is claimed that using parabolic reflectors enables the radiation to be more effectively focussed, thus achieving a long-throw characteristic unattainable with plaque heaters.

These claims will be assessed after all of the necessary mathematical tools have been expounded and the relevant experimental data has been discussed. For the time being it is sufficient to demonstrate the general characteristics that arise from the geometries and operating temperatures applicable to QLLs, and also to demonstrate that an accurate way of predicting flux-density outputs exists.

2.3.5 Comparison of Three Heater Types

Contour plots of spherical irradiance over a nominal occupation zone beneath the heater are displayed in Fig.2.5, 2.6, and 2.7 for the three types of heater which are of principal interest; the gas-fired tubular heater, the gas-fired plaque heater, and the electrically powered QLL heater. Various points arise from an examination of these plots concerning the characteristic output from the modelled heaters. The criteria of interest are; maximum flux-density, distribution of flux-density over the zone, and total output or efficiency. It is worth bearing in mind in the following discussion the wide difference between the input powers to the heaters being modelled.

Table 2.4: Input-Output Characteristics of Three Modelled Heaters

Heater	IRH_1	PLAQ_1	QLL_1
Q_{in} (kW)	30.	10.	1.5
Q_{out} (kW)	12.8	7.8	1.4
Q_{out}/Q_{in} (%)	42.	78.	93.
Q_{max} (Wm^{-2})	334.	278.	100.
Q_{max}/Q_{in} (%)	1.1	2.8	6.7

Note: The input powers are nominal manufacturers values based on Gross Calorific Values, not experimental results.

The ratio of Q_{max} to Q_{in} is the value of the highest irradiance produced divided by the input power, ie a normalization of the maximum irradiance, and it is indicative of the extent to which the beam is concentrated by the reflector. These ratios support the

earlier assertion that the plaque and QLL heaters are designed to produce spot rather than wide-area heating, and that, of the two, the QLL has a significantly more focussed beam. Further evidence for the efficacy of the design comes from examining the distribution of flux over the zone; again the QLL heater has by far the most concentrated beam, although it is surprising to note that the persistence of flux-density along the x-axis is quite pronounced compared to those of both the tubular and the plaque heater.

These plots are of great use in deciding how to mount a number of heaters together in a system, in such a way as to minimise the areas that are not covered. Such plots can also be used to assess changes in mounting height, and also to perform sensitivity studies on small changes to reflector profile. In the latter case it is necessary to only make small changes because large changes may well invalidate the temperature fields that have been fed into the model. The temperatures over the surface of the tube are themselves dependent on the rate of convective heat transfer from the tube, and as yet no work has been performed to assess numerically how the convective heat transfer is affected by altering the reflector shape.

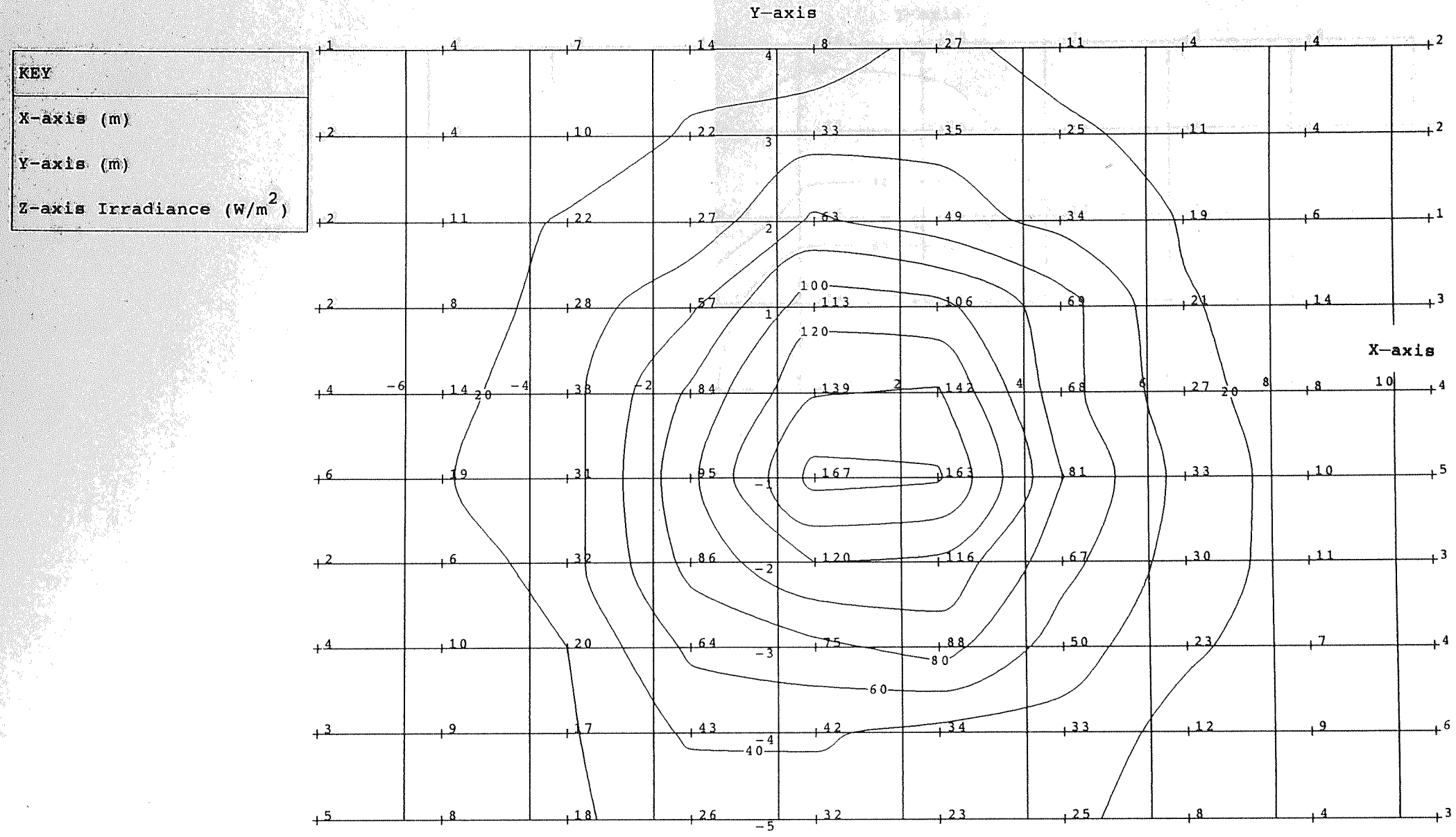


Fig.2.5:IRH_1 Irradiance Plot

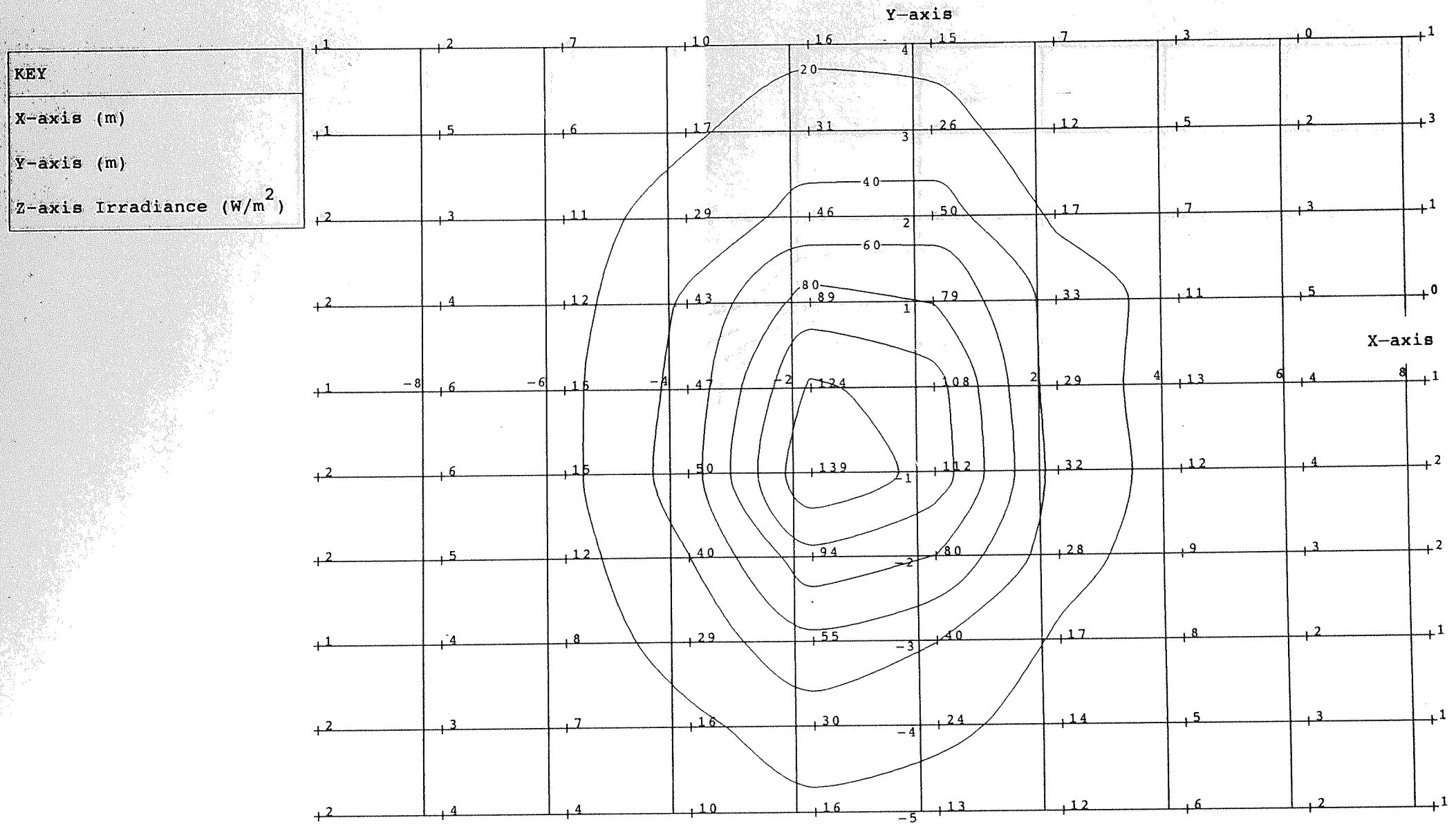


Fig.2.6:PLQ_1 Irradiance Plot

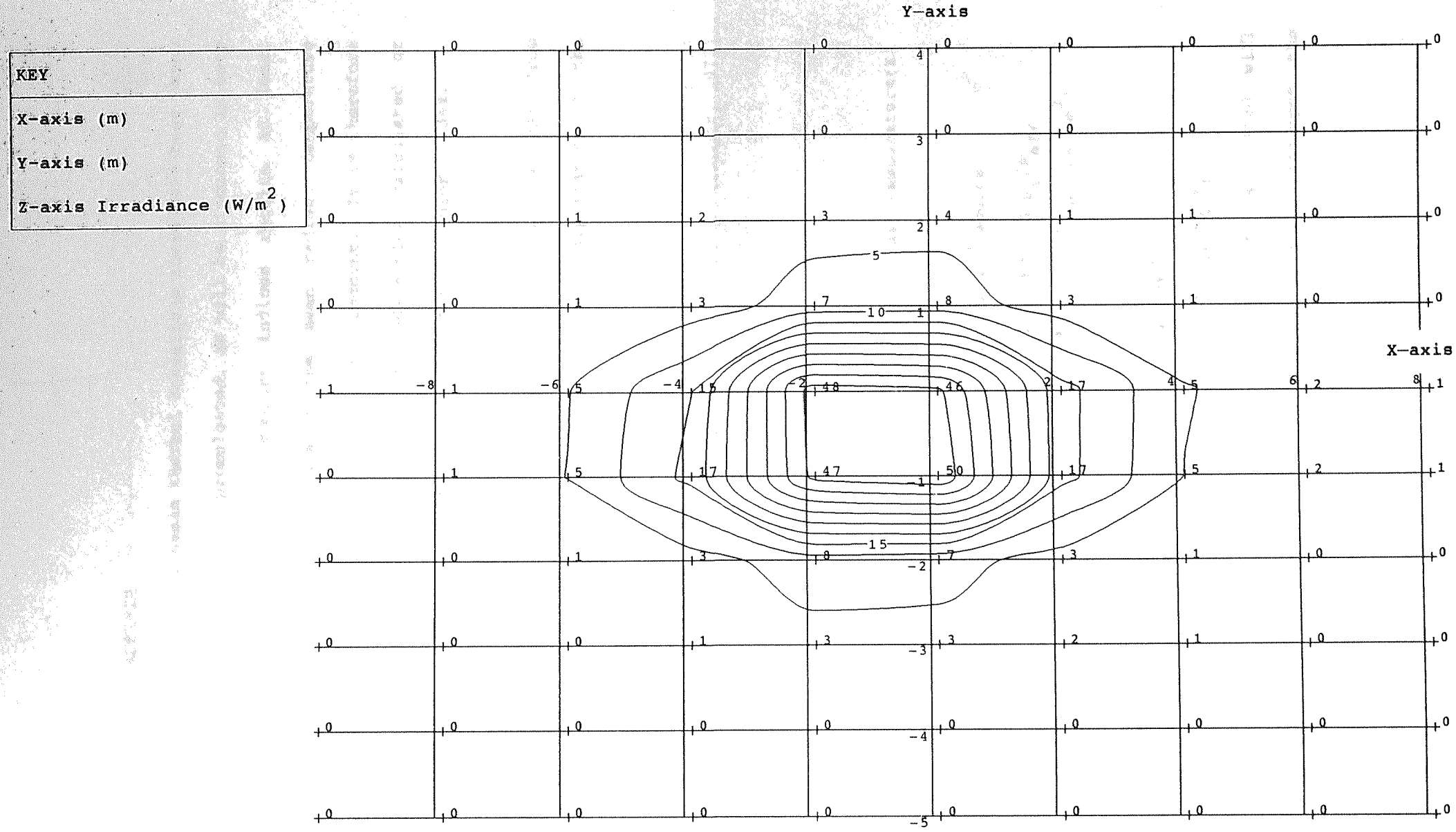


Fig.2.7:QLL_1 Irradiance Plot

2.4 Comfort Parameters

In the study of human thermal sensations there have been many different measures of comfort promulgated, as will be discussed in more detail in section 2.4.2. All comfort indices require at least knowledge of the air temperature and the mean radiant temperature (MRT); some indices take further factors into account. It is therefore important to be able to relate the flux-densities calculated or measured to the MRT as a step towards calculating a comfort index.

2.4.1 Mean Radiant Temperatures

The most straightforward way to calculate the MRT from flux-density measurements or calculations is by the application of the following relationship:

$$T_{mrt}^4 = T_{umrt}^4 + 1/(\sigma\epsilon).\sum_{i=1}^{N}(f_i.\alpha_i.q_i) \qquad (2.1)$$

(Fanger, 1972)

where:

N = number of sources

T_{mrt} = MRT (K)

T_{umrt} = unirradiated temperature - usually taken as air temperature(K)

ϵ = emissivity of subject

α_i = absorptivity of subject at wavelength of i^{th} source

f_i = projected area factor in direction of source: $A.p_i/A_{eff}$

Ap_i = area projected onto plane perpendicular to i^{th} source (m^2)

A_{eff} = total radiating area of subject (m^2)

q_i = incident flux-density from i^{th} source (Wm^{-2})

σ = Stephan-Boltzmann constant = 5.67×10^{-8} ($Wm^{-2}K^{-4}$)

It is important to note that the MRT thus calculated will refer to a specific geometry because the value of f_i depends upon the

geometry of the shape being irradiated. Unless an explicit statement to the contrary is made, it is conventional to assume that MRT always refers to a sphere, in which case:

$$f_i(\Theta,\Phi) = 1/4 \qquad (2.2)$$

where Θ and Φ are the azimuthal and altitudinal angles of incidence of an impingent radiant source. For any subject other than a sphere, f_i will vary with both Θ and Φ. The exact geometry of the human form has been quantified using a planimetric technique (Fanger, 1972), but for the purposes of this exposition the black bulb approximation will be accepted as sufficiently accurate.

The series of figures Fig.2.8, 2.9, and 2.10 show the predicted MRT, calculated using the method outlined above, for the three different types of heater. They are all shown with an ambient temperature of 0°C. These contour plots follow a similar pattern with respect to the distribution of temperatures as the irradiance plots, but the actual values produced will clearly be of significance when the ramifications of these figures for the thermal comfort provided are explored in the next section.

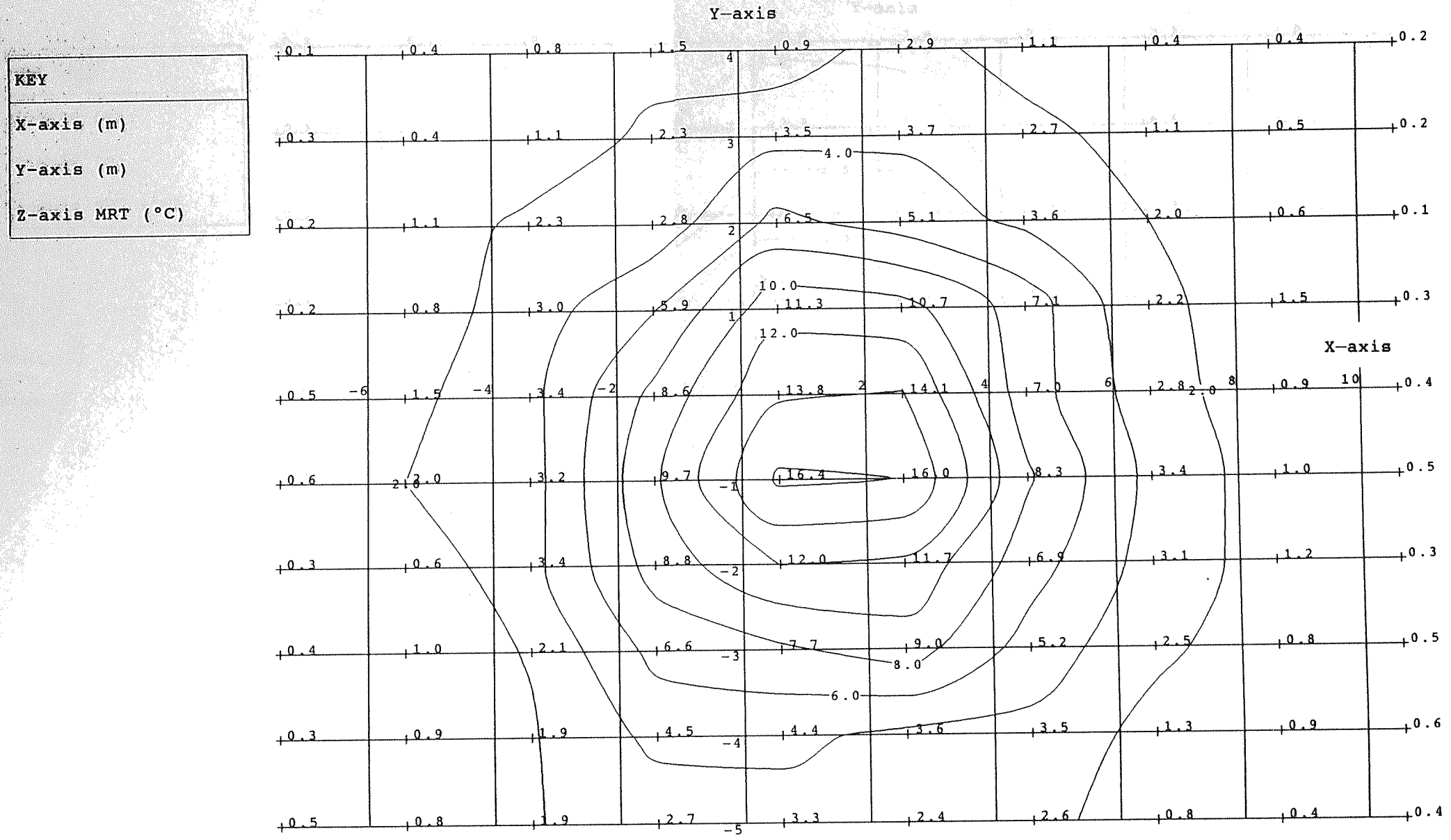


Fig.2.8:IRH_1 Black Bulb MRT Plot

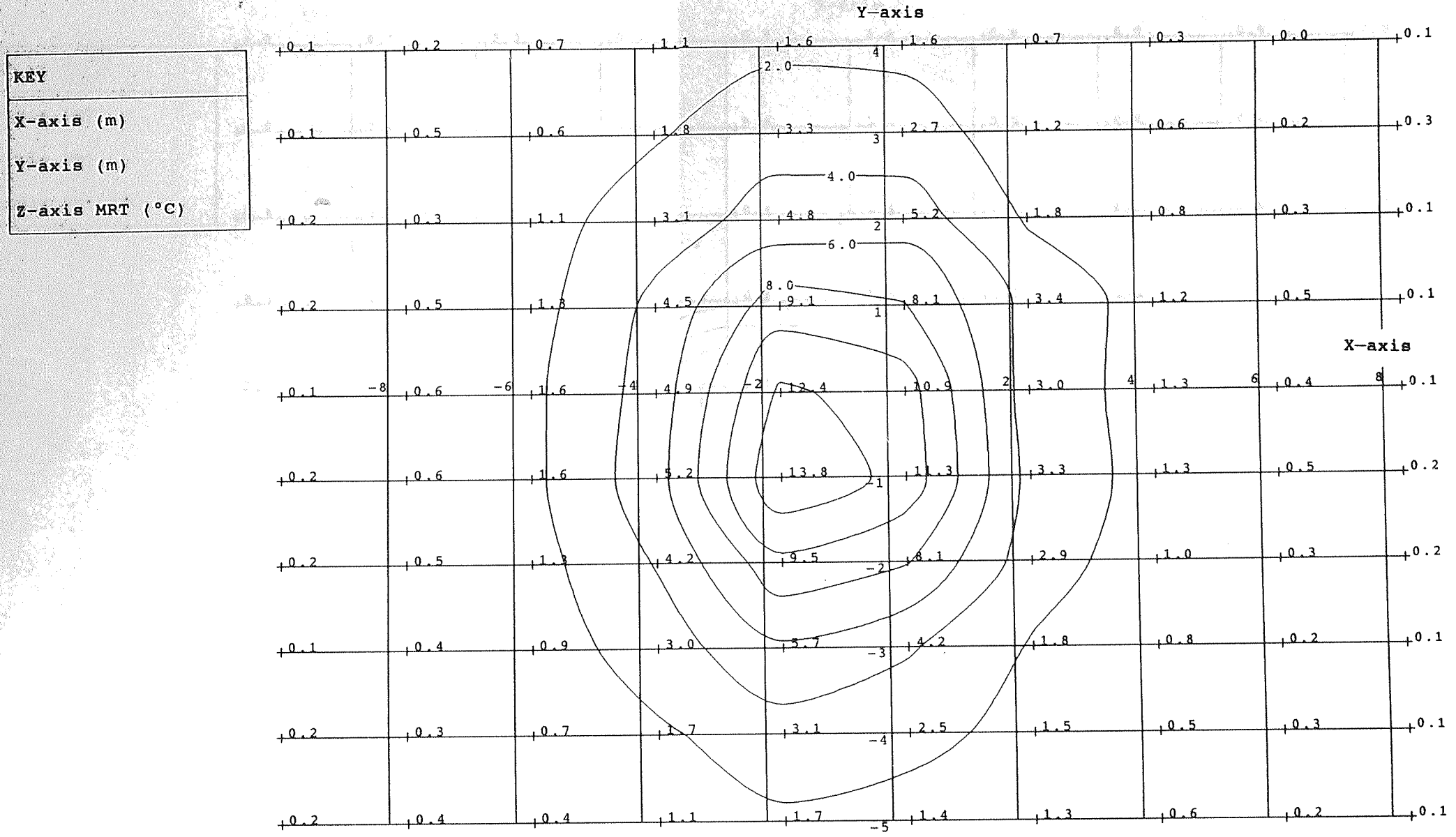


Fig.2.9:PLAQ_1 Black Bulb MRT Plot

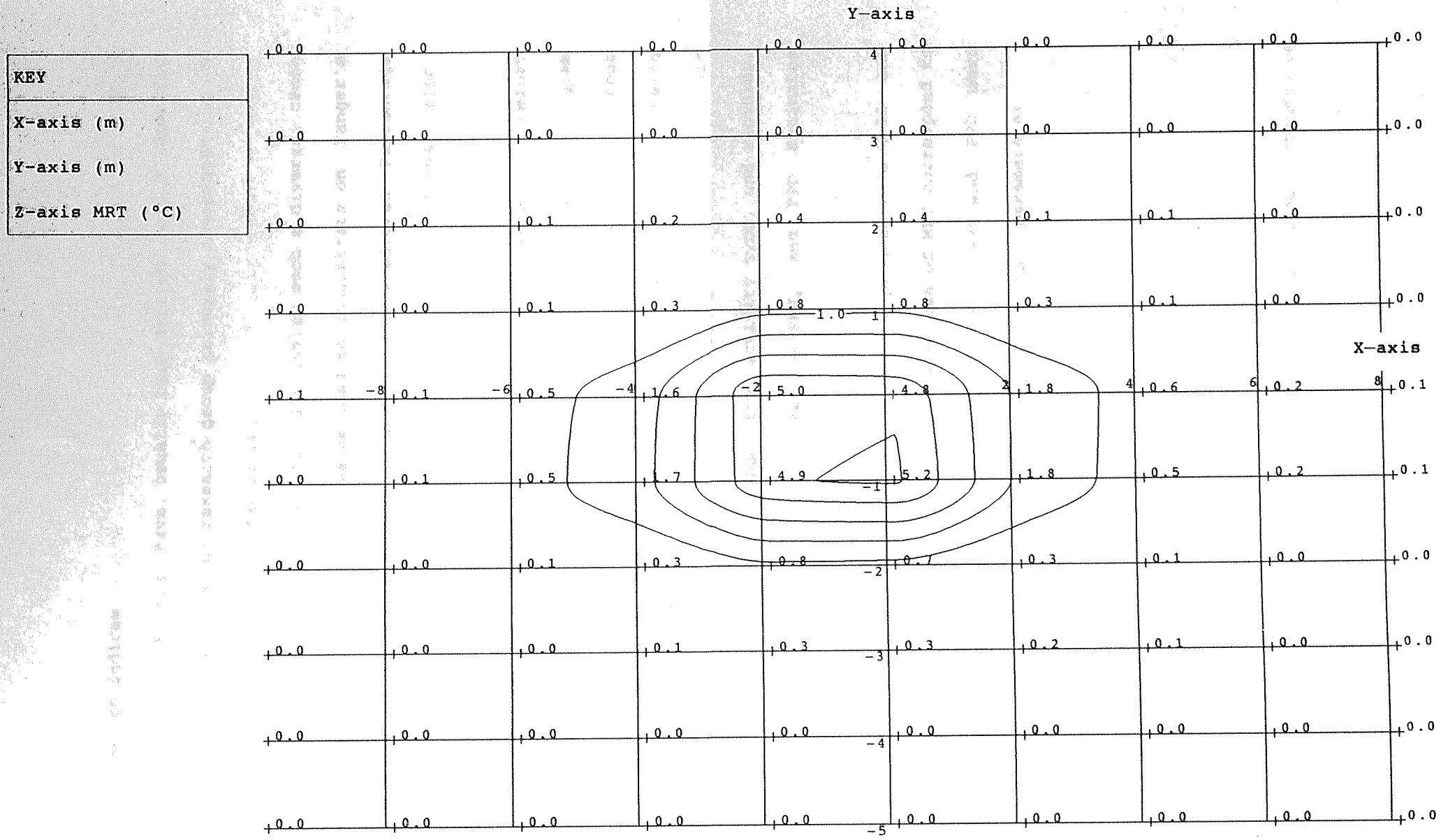


Fig.2.10:QLL_1 Black Bulb MRT Plot

2.4.2 Comfort Indices

Comfort indices have usually been derived for a specific purpose by a researcher or research group primarily interested in a particular aspect of thermal sensation.

Rather than attempt to critically review such a diverse canon of technique and practice, it was decided to concentrate on Fanger's Predicted Mean Vote (PMV) and its comeasure, the Predicted Percentage of Dissatisfied (PPD) as defined in the ISO document 7730-1984 (ISO, 1984).

In order to provide a flexible method of producing comfort estimates from the output of the model, a further short Fortran program was written called *COMFORT.FOR* (Listing 2.4). This takes the output file from the Monte Carlo model and first calculates the MRT, using Eqn.2.1, and then prompts the user for the appropriate values of air temperature, air speed, relative humidity, activity type, and clothing. From this information it then calculates the PMV, and PPD estimates across the occupied zone.

Plots produced from the output of this program (Fig.2.11 and Fig.2.12) give a good indication of how the values of MRT correspond to comfort index values. These particular values of PMV and PPD were derived with the following values of the other comfort parameters:

T_{air} = 15 °C;

V_{air} = 0.2 m/s;

Activity = MEDIUM = 116 W/m^2;

Clothing = LIGHT = 0.078 $°C.m^2/W$

Relative Humidity = 50 %

The values of PPD show clearly that radiant heaters are not intended to

be used as individual units, but rather as components of a larger system. This is borne out by Fig.2.13 and Fig.2.14 which show how several heaters may be mounted together to provide an even spread of MRT across a whole zone. These values were found using a simple program *COMB.FOR* which allows the values of flux into the occupied zone calculated by the Monte Carlo model to be combined. This combination is simply a scalar addition and does not attempt to extrapolate the modelled values before combining them.

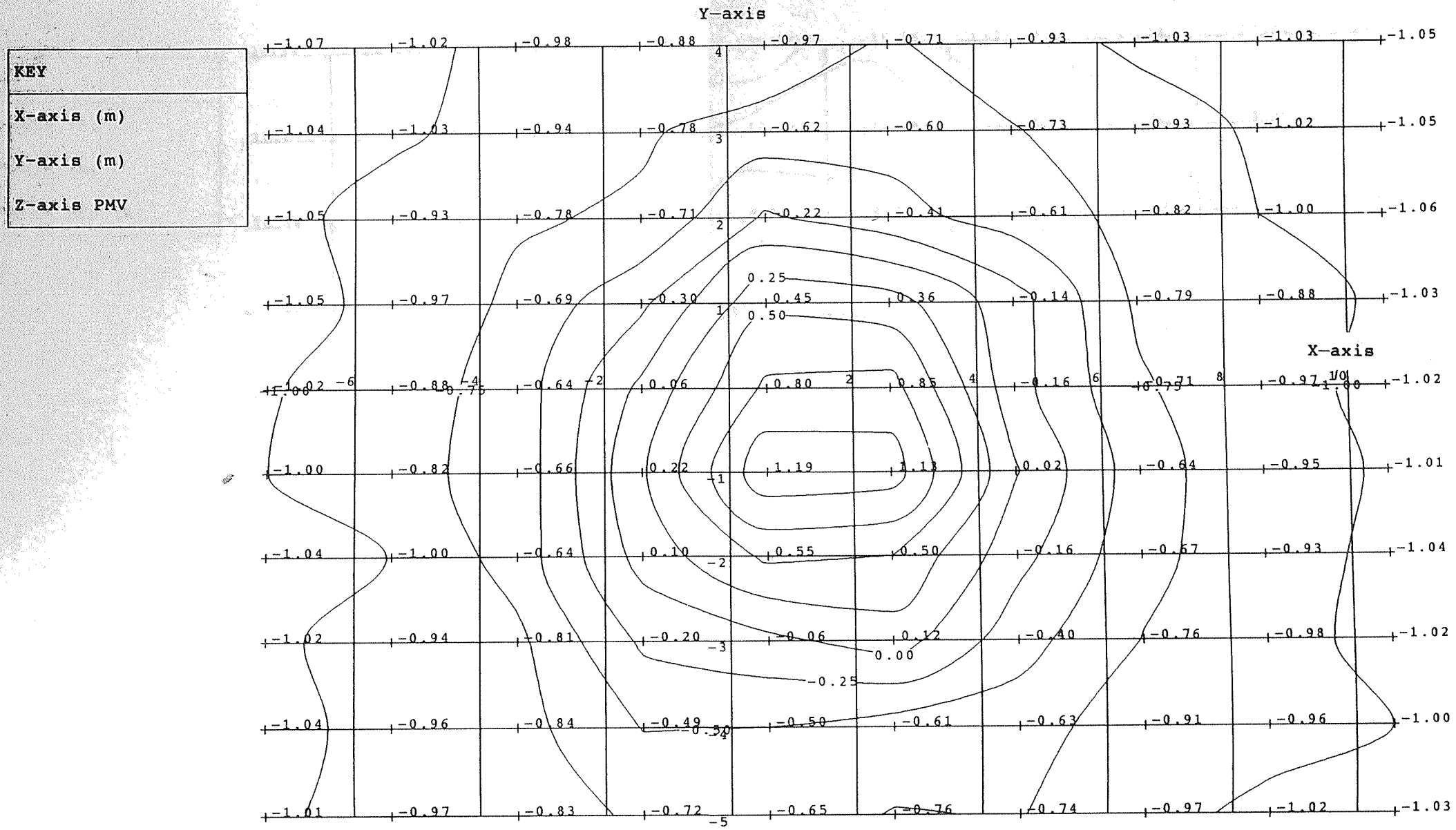


Fig.2.11:IRH_1 PMV Plot

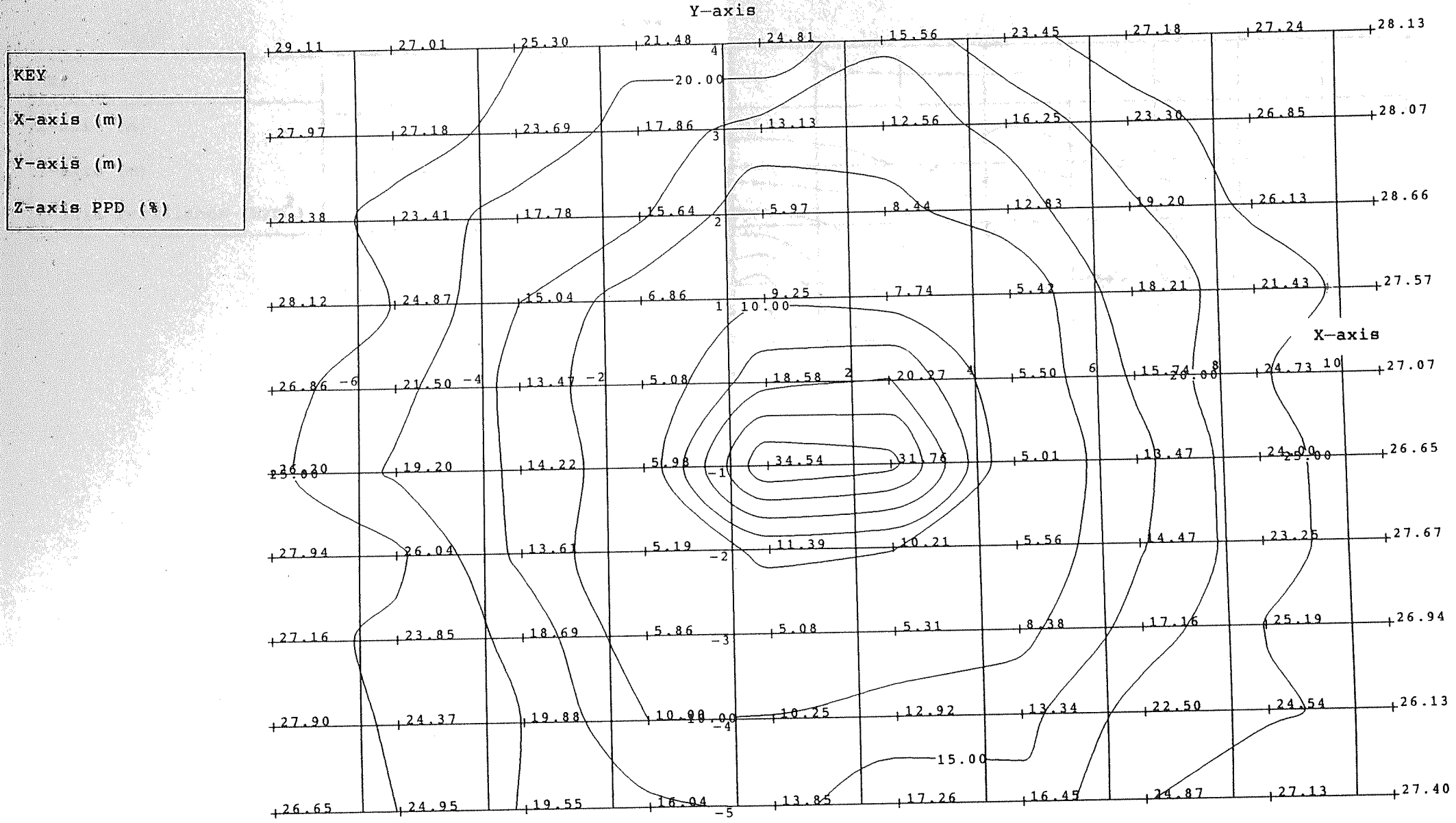


Fig.2.12:IRH_1 PPD Plot

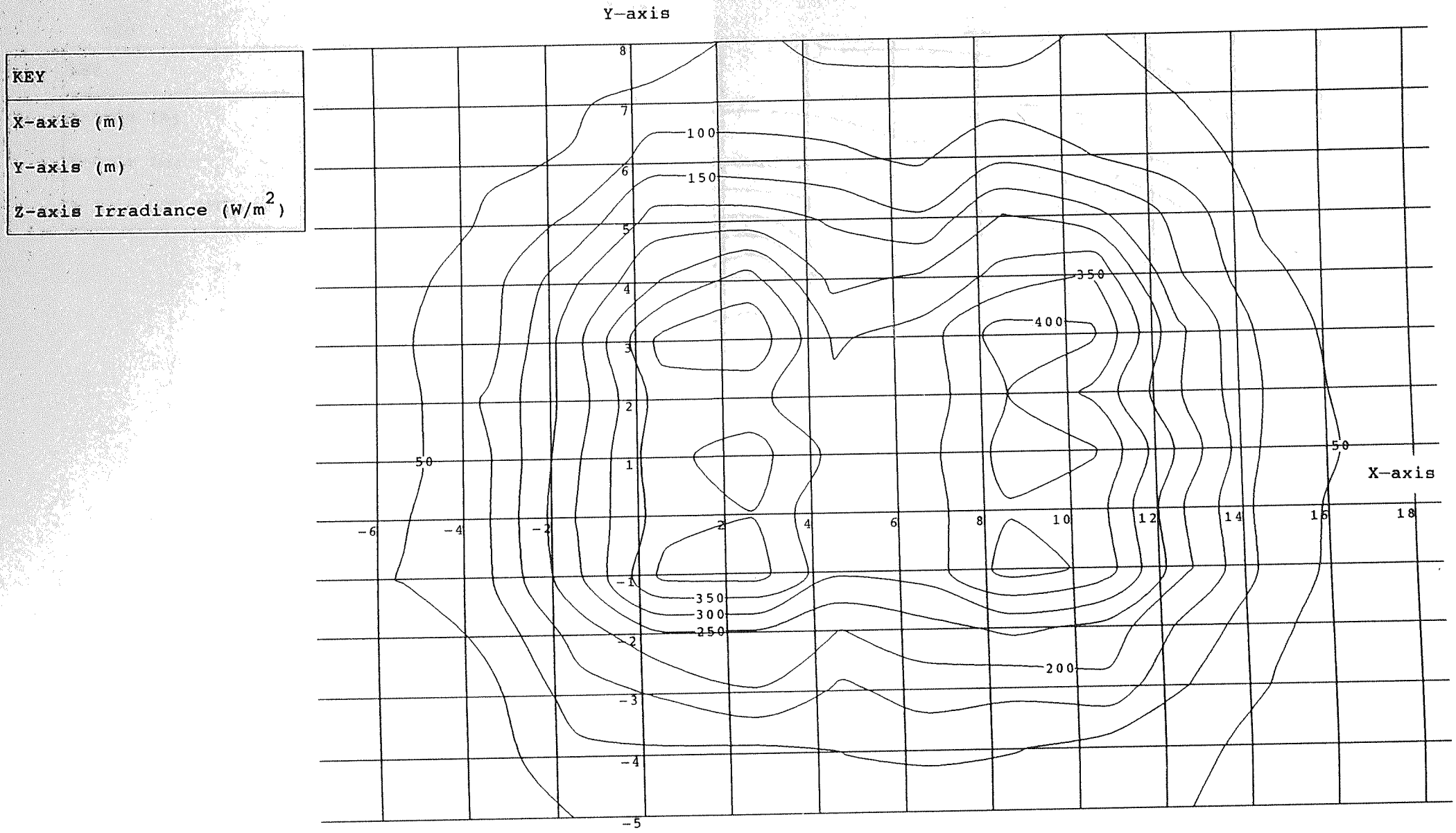


Fig.2.13:4xIRH_1 Irradiance Plot

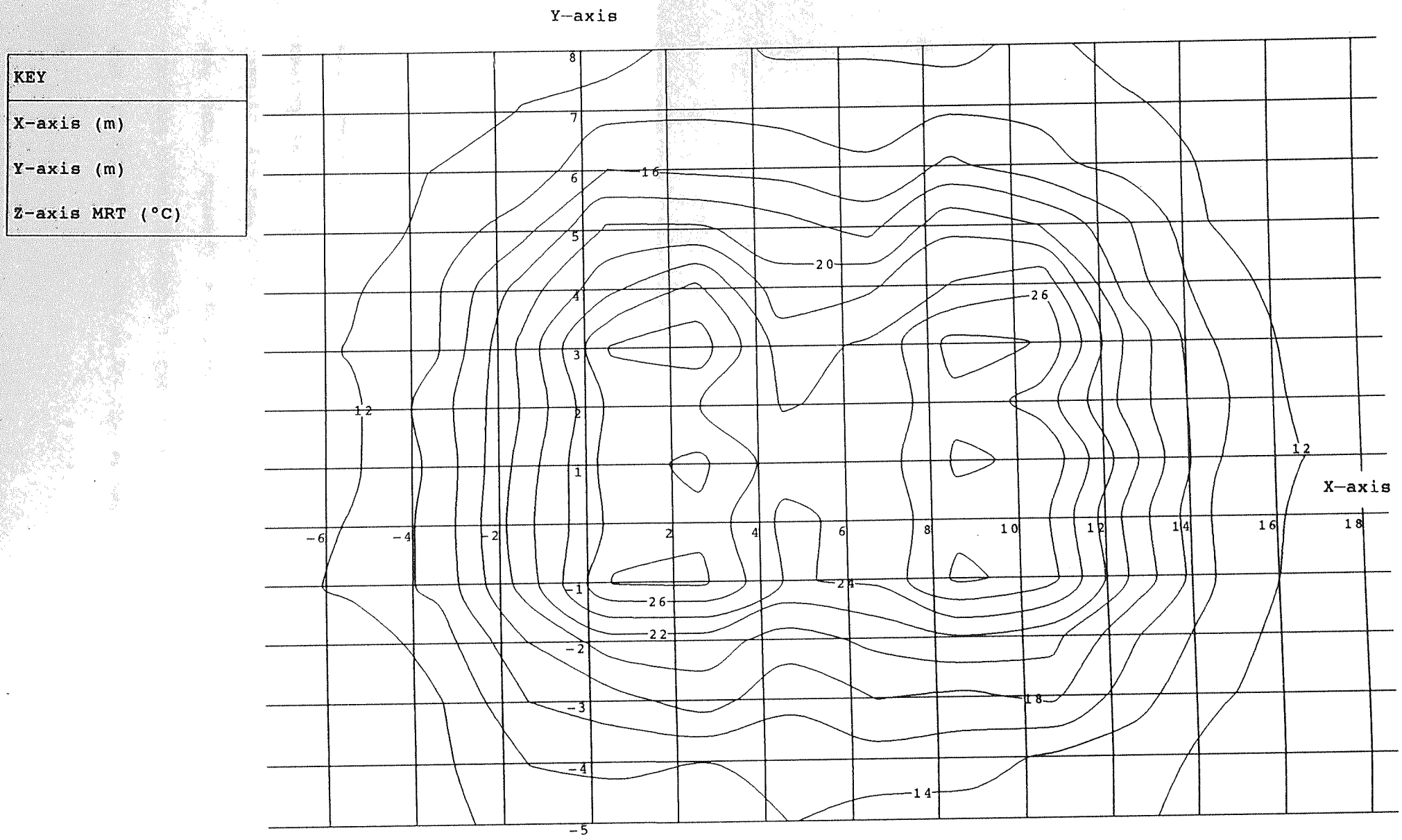


Fig.2.14:4xIRH_1 Black Bulb MRT Plot

2.5 Summary

This section provides an outline of the installation of the British Gas model onto the Aston VAX as well as its subsequent testing. It explains how the program was developed into a powerful tool for predicting not only the flux-density distribution immediately beneath an individual heater, but also the MRT and the comfort indices derived from it. Lastly, the extension of the model to deal with other types of radiant heaters is described.

3 ROOM SIMULATIONS

Having successfully implemented the Monte Carlo model, the next logical development was to find a way in which heaters could be coupled with room characteristics. The purpose of this was to provide an insight into how a system interacts with its environment and a better understanding of how radiant systems should be sized and installed.

This work is presented in three sections. The first deals with heat flows under steady state conditions. The second looks at the complications arising from transient behaviour. The last section describes how radiant systems may be controlled to maintain thermally comfortable environments whilst also minimizing fuel consumption.

A standard method for computing temperatures and heat fluxes within rooms, either in steady-state or during transient conditions, is by the application of the electrical analogy of heat flow. One of the principal attractions of this method is that once the various thermal characteristics of a room have been cast into their electrical counterparts, the solution of the complex networks that arise is facilitated by the application of standard techniques for solving electrical networks (Bode, 1957). Furthermore, if the systems are described using transfer functions or state-space variables then the considerable body of knowledge subsumed under the title of "Control Theory" may also be brought to bear on the problem (Franklin, 1986) (Elgerd, 1967). These techniques allow the dynamic behaviour of systems to be fully investigated. There is also the additional benefit that a great deal of the experience gained in assessing the performance of other systems, for example electromechanical control systems, is

also directly applicable to heating systems.

All of the work carried out in the following three sections was carried out using a matrix manipulation package "CTRL-C" on the Aston mainframe (System Control Technology, 1987). This package is a development of another package called "MATLAB" which is available for PCs. The text contains the formal mathematical representations of any matrices used, whereas any listings referred to are in CTRL-C format. The conventions used are as follows: vectors are designated with an underscore, eg $\underline{x}$; matrices are shown in bold, eg $\mathbf{Q}$; subscripts refer to the elements of a matrix, eg Q_{ij} is the element occupying the i^{th} row and j^{th} column of the matrix $\mathbf{Q}$; the transpose of a matrix or vector is represented by a superscript τ, ie $\underline{y}^{\tau}$; matrix inversion is shown by $\mathbf{Q}^{-1}$; differentiation with respect to time is denoted by use of primes, eg y'' is the second time derivative.

3.1 Steady State Heat Flows

3.1.1. The Electrical Analogy

The basis of the analogy between flow of electrical charge and heat is that the equations governing transport phenomena have the same general form: Flow = (Conductance) . (Driving Force)

This common basis allows the application of knowledge gained in one area to be made in an analogous field (Ziesler and Maund, 1989) (Whitman and Ziesler, 1989).

As with any analogy there are pitfalls, the most serious being the extension of the analogy to situations in which it is no longer applicable. It is arguable, for example, to what extent linearised heat transfer coefficients for radiative and convective heat transfer fall within the domain of valid analogy. Table 3.1 displays various electrical quantities and their counterparts for the three modes of heat transfer.

Table 3.1: Analogous Quantities

Electrical	Conduction	Convection	Radiation
Voltage (V)	temperature (K)	temperature (K)	temperature (K)
Current (A)	heat flow (W)	heat flow (W)	heat flow (W)
Resistance (Ω)	thermal resistance (K/W)	1/(HTC) (K/W)	radiative resistance (K/W)
Conductance (mho)	thermal conductance (W/K)	HTC (W/K)	radiative conductance (W/K)

NOTE: HTC = Heat Transfer Coefficient

It should also be mentioned that it is more usual to encounter the thermal transmittance or U-value ($Wm^{-2}K^{-1}$) (this is the conductance, normalized to unit area) of which there is no commonplace electrical equivalent. These transmittances are calculated by different methods for the three modes. For conduction G^n could be

taken directly from the steady state equation of heat flow across a slab:

$$q_w = G^n.A(T_{si} - T_{se}) \qquad (3.1)$$

where q_w = heat transmitted through the slab (W);

G^n = normalised thermal conductance ($Wm^{-2}K^{-1}$);

A = cross-sectional area of slab (m^2);

T_{si} = Temperature of internal face (°C);

T_{se} = Temperature of external face (°C);

it is customary, however, to tabulate transmittances which include an estimate of the surface thin-film Heat Transfer Coefficient (HTC) for the walls known as U-values. It has been necessary, for the modelling work which follows, to try to arrive at an estimate of the value of G^n by using the following formula:

$$(G^n)^{-1} = U^{-1} - h_a^{-1} \qquad (3.2)$$

where U is the transmittance found from standard tables and h_a is the convective HTC. The values of h_a used were the conventional ones of 3.0 $Wm^{-2}K^{-1}$ (vertical surfaces), 4.3 $Wm^{-2}K^{-1}$(horizontal lower surface), and 1.5 $Wm^{-2}K^{-1}$ (horizontal upper surface) (CIBSE, 1986). The problem of choosing values of radiative conductances proved to be complicated. The usual method of calculating radiative interchanges within a network uses the analogies: voltage => rate of energy emission per unit area (Wm^{-2}); current => rate of heat flow (W); resistance => radiation resistance (m^{-2})(Gray and Muller, 1974). The problem that occurs when using this scheme is that it becomes difficult to integrate the radiative mode of heat transfer with the two other modes. The alternative scheme, whereby the analogy is made using the equation:

$$Q_{ij} = G_{ij}.(T_i - T_j) \qquad (3.3)$$

where Q_{ij} = heat gain at j^{th} from the i^{th} surface (W)

$G_{ij} = Hr_{ij}.A_j$ = radiative conductance (W/K)

T_j = temperature of the j^{th} surface (K)

T_i = temperature of the i^{th} surface (K)

Hr_{ij} = linearized radiative HTC[1] ($W/m^2/K$)

A_j = area of j^{th} surface (m^2)

necessitates the use of a conductance which depends on the cube of the temperature. This is only acceptable provided the temperature range over which the approximation is used is small. In this case the radiating surface deviates from the nominal 20 °C by ±5 °C, this results in an acceptably small error in the values of HTC used.

3.1.2 Methods of Solving Thermal Networks

The two most powerful methods of solving electrical networks are the "current mesh" and "voltage node" techniques (Edminster, 1983). These methods may be directly employed to solve thermal networks by direct use of the analogy.

The basis of the Current Mesh approach is the division of the network into smaller loops called meshes, to which Kirchoff's Voltage Law (KVL) may be applied in a straightforward manner to yield a matrix equation relating the mesh currents to the node voltages:

$$\mathbf{R}\cdot\underline{I} = \underline{V} \qquad (3.4)$$

where $\mathbf{R}$ = (n,n) mesh resistance matrix;

$\underline{I}$ = (n,1) current vector;

$\underline{V}$ = (n,1) voltage vector.

The Voltage Node method is complementary, in that it involves

the application of Kirchoff's Current Law (KCL), as opposed to KVL, at each successive principal node[2], rather than at each current loop. One principal node is selected as a reference, and this technique yields a relation between the node voltages and the branch currents:

$$\mathbf{G}\cdot\underline{V} = \underline{I} \qquad (3.5)$$

where $\mathbf{G}$ = (n,n) conductance matrix.

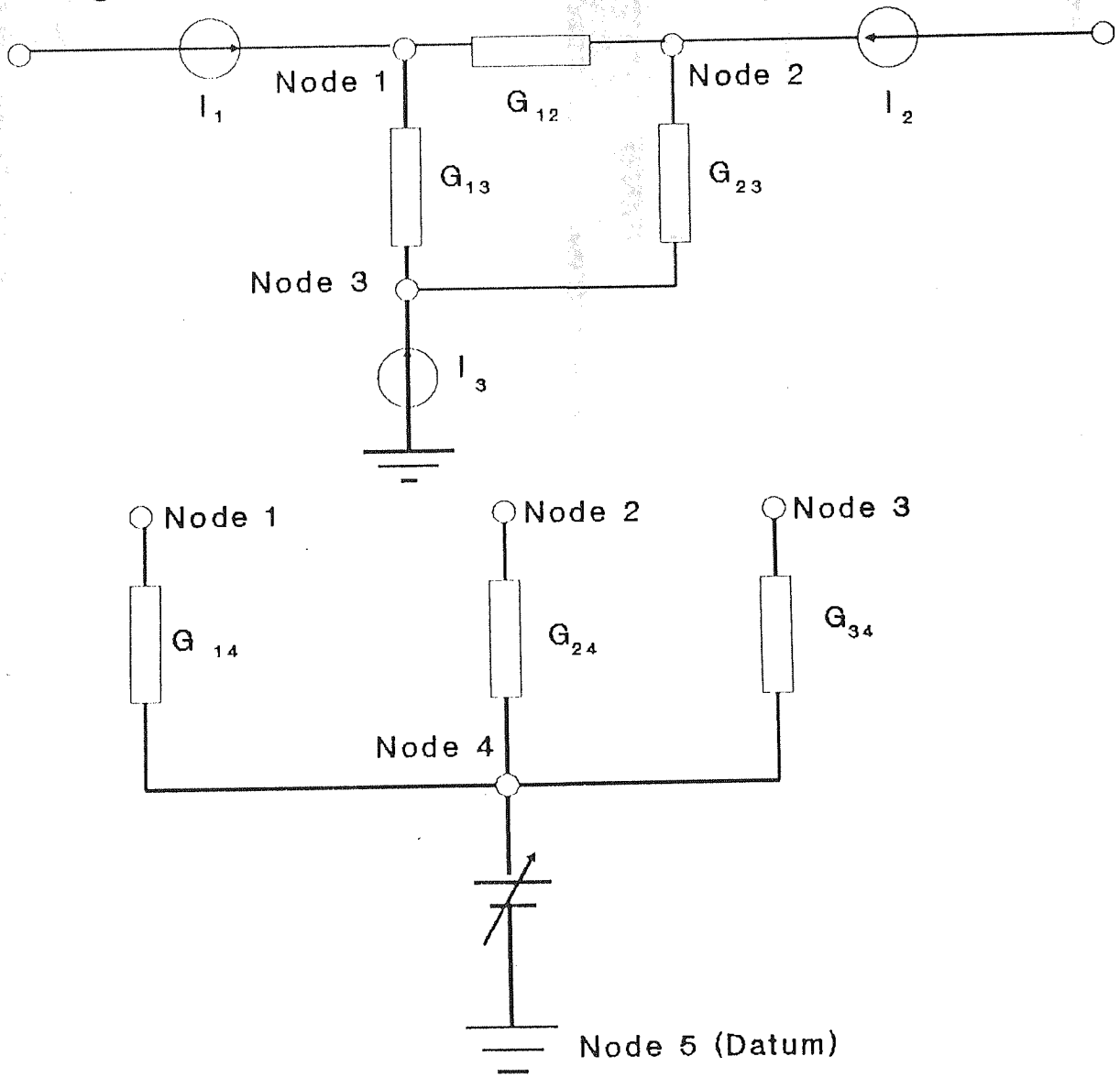
These equations may be solved by the inversion of the resistance or conductance matrices. To illustrate the procedure a conductance matrix will be derived for a two walled room which will demonstrate the essential points of the technique. Fig.3.1 shows the equivalent electrical circuit along with typical values. The final conductance matrix[3] is:

$$\begin{vmatrix} I_1+V_4G_{14} \\ I_2+V_4G_{24} \\ I_3+V_4G_{34} \end{vmatrix} = \begin{vmatrix} \Sigma G_{1j} & -G_{12} & -G_{13} \\ -G_{21} & \Sigma G_{2j} & -G_{23} \\ -G_{31} & -G_{32} & \Sigma G_{2j} \end{vmatrix} \cdot \begin{vmatrix} V_1 \\ V_2 \\ V_3 \end{vmatrix} \qquad (3.6)$$

In order to complete the analogy the currents that are being sourced into the circuit are taken to be the heat flowing into the various nodes in question. For the surfaces this means the heat absorbed from radiant heaters; for the air-node it means the sum of convective heat from the reflectors plus the sensible heat from the flue (provided of course that the system is not flued externally).

Having specified all the necessary values, it only remains to post-multiply the inverted conductance matrix by the current vector to yield temperature values for all of the surfaces and the air point. Conversely, if the desired temperatures are known, then the heat inputs required to produce them are readily found. The next section shows a fully worked example of the use of this technique.

Fig.3.1: Circuit for 2-Surface Room



3.1.3 Application of the Voltage Node Method to a Medium Sized Room

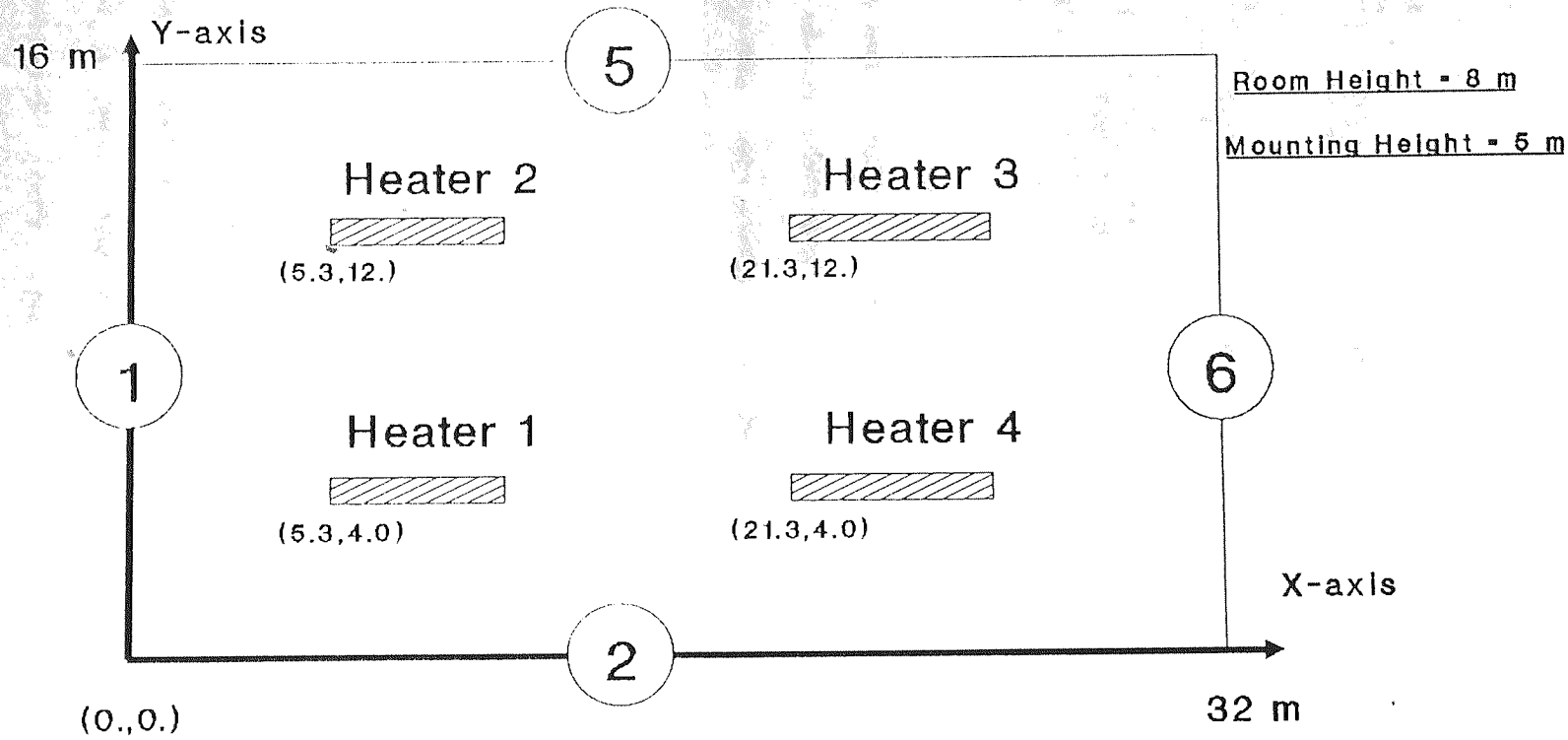
Fig.3.2 shows a notional medium sized room in which four radiant heaters are mounted. In contradistinction to the 5-node network that was examined in the last section, this room is represented by 9-nodes: six walls, the internal and external air points, and the reference ground. Because the external air-node is specified and the ground-node is a reference, this configuration leads to a (7,7) conductance matrix, which can be solved in exactly the same way as the (3,3) matrix in the simpler example.

$$\begin{vmatrix} I_1+V_4G_{18} \\ I_2+V_4G_{28} \\ I_3+V_4G_{38} \\ I_4+V_4G_{48} \\ I_5+V_4G_{58} \\ I_6+V_4G_{68} \\ I_7+V_4G_{78} \end{vmatrix} = \begin{vmatrix} \Sigma G_{1j} & -G_{12} & -G_{13} & -G_{14} & -G_{15} & -G_{16} & -G_{17} \\ -G_{21} & \Sigma G_{2j} & -G_{23} & -G_{24} & -G_{25} & -G_{26} & -G_{27} \\ -G_{31} & -G_{32} & \Sigma G_{3j} & -G_{34} & -G_{35} & -G_{36} & -G_{37} \\ -G_{41} & -G_{42} & -G_{43} & \Sigma G_{4j} & -G_{45} & -G_{46} & -G_{47} \\ -G_{51} & -G_{52} & -G_{53} & -G_{54} & \Sigma G_{5j} & -G_{56} & -G_{57} \\ -G_{61} & -G_{62} & -G_{63} & -G_{64} & -G_{65} & \Sigma G_{6j} & -G_{67} \\ -G_{71} & -G_{72} & -G_{73} & -G_{74} & -G_{75} & -G_{76} & \Sigma G_{7j} \end{vmatrix} \cdot \begin{vmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \\ V_7 \end{vmatrix} \quad (3.7)$$

Table 3.2: Values of Components for Thermal Circuit

Component	Description	Value
V_i	temperature of the i^{th} node	
$I_i (0<i<7)$	heat flux into the i^{th} node	
G_{ij}	conductance between i and j^{th} node	
$0<i\&j<7$	radiative conductance surface i=>j	$Hr_{ij}A_j$
$G_{i7} = G_{7j}$	convective conductance surface i=>air	Ha_iA_i
$G_{i8} = G_{8j}$	conductive conductance i=>external	$(U_i^{-1}-Ha_i^{-1})^{-1}.A_i$
G_{78}	ventilation conductance air=>external	$c_a.rho_a.N.V/3600$
Hr	radiative HTC ($W/m^2/K$)	
Ha	convective HTC ($W/m^2/K$)	
A	area (m^2)	
U	thermal transmittance ($W/m^2/K$)	
c_a	specific heat capacity of air (J/kg/K)	

Fig.3.2: Geometry of Medium Sized Room



Notes;

Nodes 1 & 6 : Surfaces of End Walls	Nodes 8 & 13: Bodies of End Walls
Nodes 2 & 5: Surfaces of Side Walls	Nodes 9 & 12: Bodies of Side Walls
Node 3: Surface of Floor	Node 10: Body of Floor
Node 4: Surface of Ceiling	Node 11: Body of Ceiling
Node 7: Air Point	Node 14: Datum

Coordinates of Heaters refer to the Burner Position (m)

The values of the heat flows to the various nodes were derived from a modification to the Monte Carlo model. The alteration to the geometry of the model took the form of placing the modelled unit in an enclosure whose dimensions and emissivity had been previously specified. Then, rather than tallying the irradiances over a grid below the heater, the program *TRIP.FOR* (Listing 3.1) was used to keep an account of the heat fluxes into the various surfaces involved. The procedure was then repeated for all of the other heaters present to obtain a grand total heat flux into the pertinent nodes.

The calculation scheme embodied in the CTRL-C procedure *ITV.CTR* (Listing 3.2) is that the parameters are provided by first specifying the geometry of the problem, then calculating the requisite heat fluxes with the modified model, and lastly supplying the remaining parameters from the building definition. There are 7 variables to be supplied before the solution can be computed. These are: the heat input to the 7 nodes, the U-values of the walls, the temperatures and the emissivities of the surfaces, the air change rate (ACR), the room dimensions, and the external temperature.

Table 3.3: Monte Carlo Output (kW)

Node	1	2	3	4	5	6	7	SUM
4.00 m	End	Side	Floor	Ceiling	Side	End	Air	
IRH_1	2.08	7.32	44.68	2.07	4.34	2.46	7.34	70.29
IRH_6	1.60	5.73	38.54	1.90	3.69	2.14	5.37	58.97
5.00 m	End	Side	Floor	Ceiling	Side	End	Air	
IRH_1	2.65	8.96	41.15	2.48	5.31	2.41	7.34	70.30
IRH_6	2.04	7.16	35.43	2.27	4.65	2.05	5.37	58.97
6.00 m	End	Side	Floor	Ceiling	Side	End	Air	
IRH_1	3.09	10.44	37.91	3.01	6.19	2.34	7.34	70.32
IRH_6	2.43	8.35	32.64	2.73	5.47	1.99	5.37	58.98

Note: These figures are for 4 units installed into the MEDIUM-SIZED room at 5.00 m above the floor.

Table 3.3 gives the summed flux from two types of heaters as a function of mounting height as an example of modelled values that are supplied to *ITV.CTR*.

It is interesting to observe that IRH_1 delivers a much larger proportion of its available input energy directly to the floor than IRH_6 at all mounting heights. This can only be due to a reflector design which maintains a narrower beam. As far as the thermal network goes this is a good strategy to adopt, firstly, because the floor has the lowest conductance and will therefore maintain its temperature better than any of the other surfaces; secondly, the floor has the largest surface-to-air conductance which means that heating the floor offers a good way of indirectly raising the air temperature.

The variation of flux with mounting height is also quite straightforward to explain. As the height is increased the beam is cast over a wider area implying that the floor will receive less direct radiation and the walls proportionally more.

Table 3.4 outlines the various parameters whose effects have been studied using the method described above. These are a representative selection of values which would be encountered in a wide range of installations. For sake of simplicity the measure which has been applied to make a comparison between the different combinations of parameters is the environmental temperature (T_e) which is defined as being:

$$T_e = 1/3.T_a + 2/3.T_{mrt} \qquad (3.8)$$

where T_e = environmental temperature (°C)

T_a = air temperature (°C)

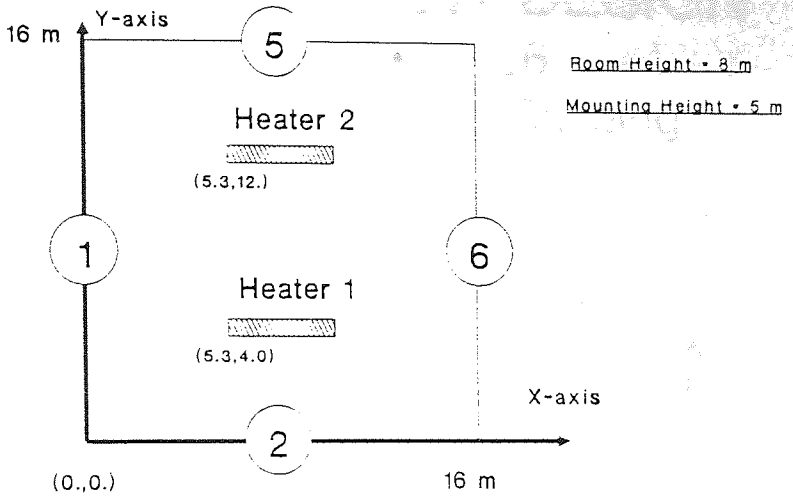
T_{mrt} = mean radiant temperature (°C)

Table 3.4: Sizing Model Parameters

Parameter	Quantity varied	Range of Values
Installation	Heater type	IRH_1 (30 kW i/p) irh_6 (36 kW i/p)
	Mounting Height	3.50 m to 6.50 m with 5.00 m DEFAULT
	Spacing	EQUAL SPACING[4] -1m -2m
Building	Physical dimensions (See Fig.3.3)	Small (16x16x 8 m) MEDIUM (32x16x 8 m) Large (72x32x 8 m)
	Air Change Rate (ACR)	Low (0.5 /hr) MEDIUM (1.0 /hr) High (2.0 /hr)
	U-value	Low (0.5 $Wm^{-2}K^{-1}$) (0.75$Wm^{-2}K^{-1}$) MEDIUM (1.0 $Wm^{-2}K^{-1}$) (1.5 $Wm^{-2}K^{-1}$) High (2.0 $Wm^{-2}K^{-1}$)
External conditions	External air temperatures	

The series of graphs Figs.3.4-8 chart the effects of the parameters presented in Table 3.4 by varying the value of the parameter under examination against a default parameter-set which consists of the capitalized values in the table. It is then possible to separate the parameters into those which are of primary importance, and those which have a secondary effect. This, albeit arbitrary, demarcation divides those parameters which have a $\delta T_e > 1°C$ from those with $\delta T_e < 1°C$: the measure δT_e has been introduced to try to approximately quantify the relative sizes of the effects of different parameters, and is the temperature variation produced when a given parameter is varied over the whole of its normal range. Table 3.5 draws together these designations and gives a ranking in terms of the size of the effect.

Fig.3.3: Geometry of Small Sized Room



Notes:

Nodes 1 & 6 : Surfaces of End Walls	Nodes 8 & 13: Bodies of End Walls
Nodes 2 & 5: Surfaces of Side Walls	Nodes 9 & 12: Bodies of Side Walls
Node 3: Surface of Floor	Node 10: Body of Floor
Node 4: Surface of Ceiling	Node 11: Body of Ceiling
Node 7: Air Point	Node 14: Datum

Coordinates of Heaters refer to the Burner Position (m)

Geometry of Large Sized Room

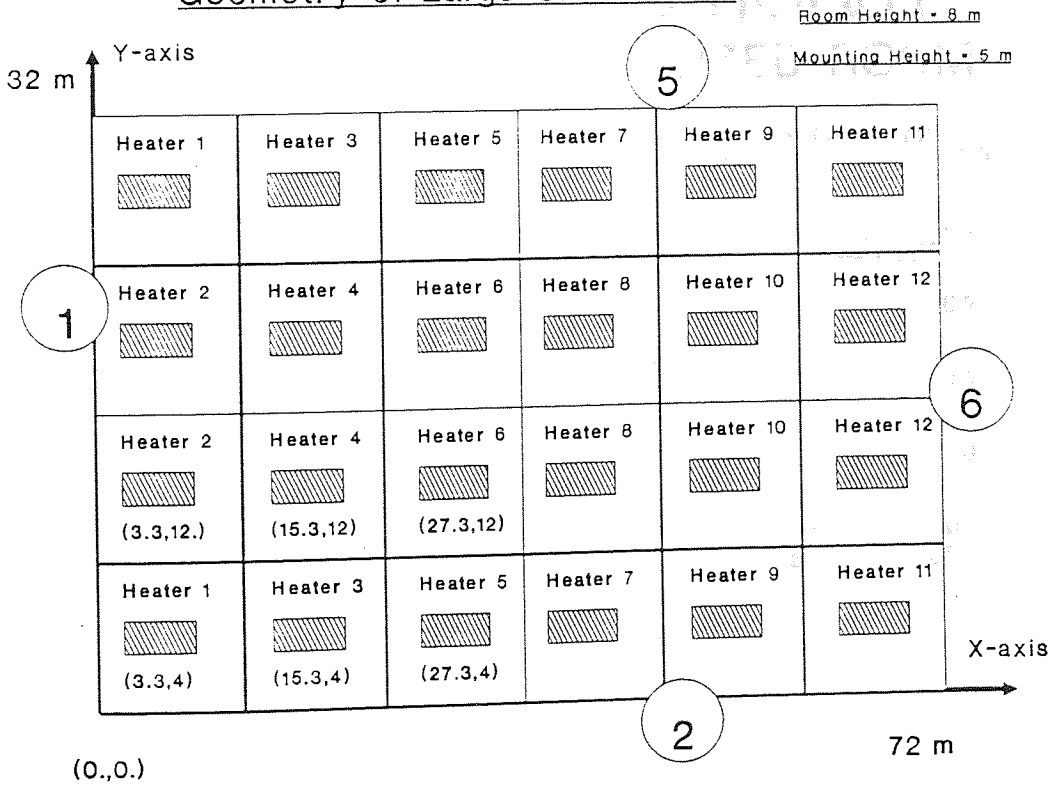
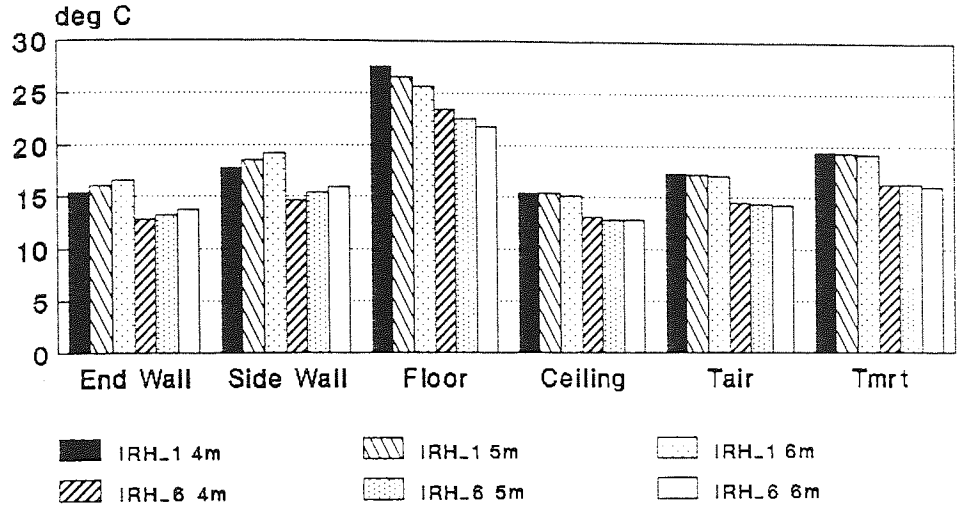


Fig.3.4: HEATER DESIGN

IRH_1 vs IRH_6

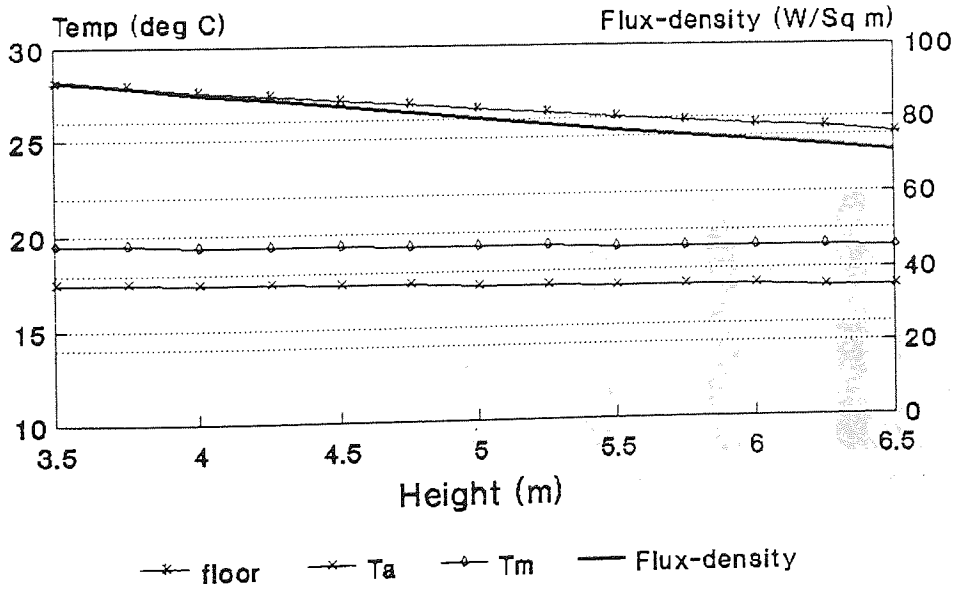
In a Medium Sized Building



T_ext = 0 deg C

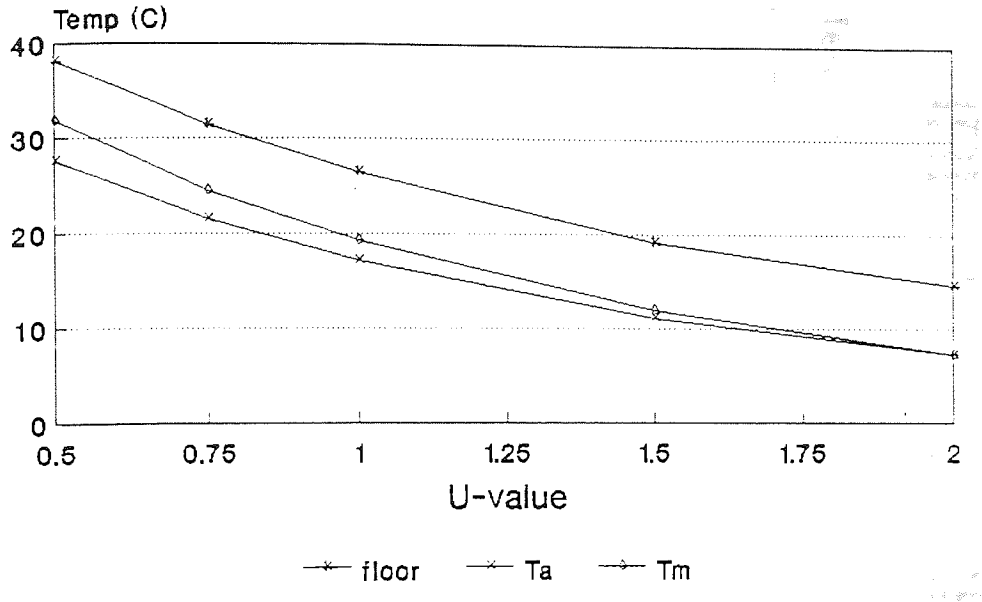
FIG.3.5: MOUNTING HEIGHT

IRH_6 Units in a MEDIUM-SIZED ROOM



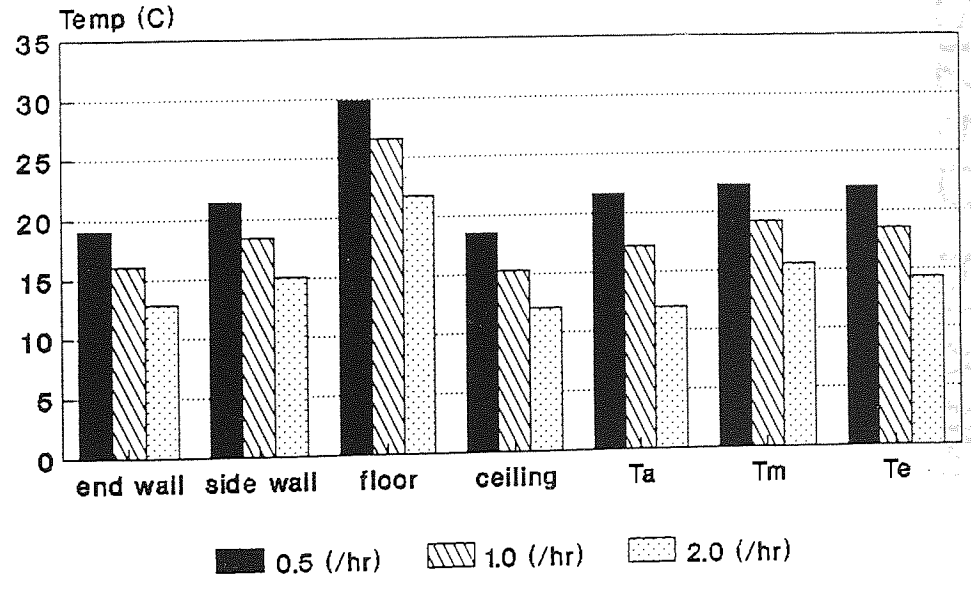
T_ext = 0 deg C

FIG.3.6: U-VALUES
IRH_6 in a MEDIUM-SIZED ROOM



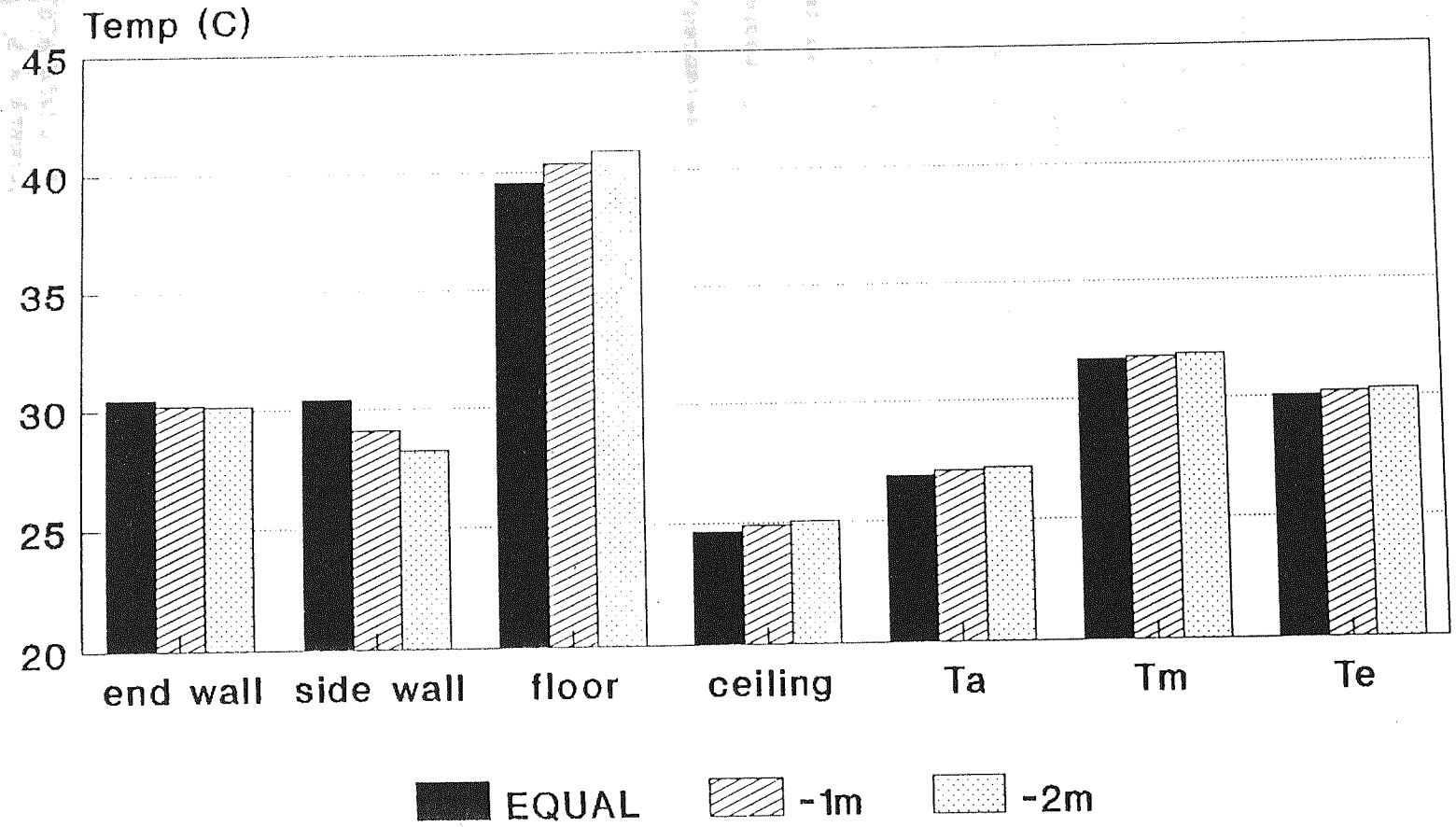
T_ext = 0 deg C

FIG.3.7: EFFECT OF ACR
IRH_6 in MEDIUM-SIZED ROOM



T_ext = 0 C

FIG.3.8: HEATER SPACING
IRH_6 in a MEDIUM-SIZED ROOM



T_ext = 0 C

Table 3.5: Relative Importance of Various Sizing Parameters

Parameter	Size of Effect δT_e (°C)	Rank Order	Significance Primary = P Secondary = S	Figure
Heater	3.	3	P	3.4
Height	0.5	4	S	3.5
U-value	23.	1	P	3.6
Spacing	0.2	5	S	3.7
ACR	8.	2	P	3.8

There are several important conclusions to be drawn from this modelling work. The most pertinent point as far as this dissertation is concerned is that individual heater design has a significant effect upon system performance. This means that the difference between a well-designed heater and a poorly-designed one can be equivalent to increasing the U-value by ≈0.2 $Wm^{-2}K^{-1}$, or the ACR by ≈0.7 hr^{-1}.

Once the particular heaters have been selected, however, the actual installation, involving the placement of heaters within the building and the selection of a mounting height, has relatively little effect on the heat balances and resulting temperatures. This is extremely important because these two factors have a great deal of effect on how thermally comfortable the occupants are. Removing the energy-efficiency constraint from the installation allows the heaters to be mounted at such heights and using such spacings as to provide even flux-density fields at comfortable intensities. The values of mounting height and spacings should ideally be calculated using the full Monte Carlo model discussed in section 2.3, although most manufacturers will have their own rules-of-thumb.

Lastly, and perhaps of greatest significance, considering trends towards low-energy factories, is the fact that the radiant

heaters perform best in buildings with low U-values - Cf Fig.3.6. This is a consequence of the direct coupling of the heat source to the walls. By comparison radiant heating systems are relatively insensitive to changes in ACR - Fig.3.7 - because the air is warmed only as a secondary effect, mostly by natural convection from the floor.

3.2 Transient Heat Flow and Dynamic Room Response

In taking up the challenge of producing an accurate picture of transient behaviour, it is necessary to make more approximations when modelling system behaviour. The most obvious problem confronting the environmental engineer in this field of enquiry is the inconstancy of the weather. Along with the lack of precise knowledge of building materials and constructions, the influence of the weather typifies the essential differences between the analysis and control of thermal versus electromechanical systems. No matter how elegant the analysis involved in specifying a heating system, it will always be ultimately the commissioning of the system which is the most important single determinate to the success or otherwise of the installation.

3.2.1 State Variable Analysis

When dealing with electrical networks in the frequency domain there are two complementary approaches to the task. The first uses complex numbers to represent the amplitude and phase information required to properly specify the relationship between the driving force and the driven quantity: this is known as the method of admittances. The admittance of an electrical impedance is generally a complex number because the impedance will contain inductive and capacitive elements, ie the circuit will have components capable of storing energy. Likewise in thermal networks, although there is no counterpart to the electrical inductance, the fabric and air are capable of storing heat, leading to complex thermal admittances. If this approach were adopted then equations of a similar nature to Eqn. 3.7 would result, except that the conductances, G_{ij}, would be replaced by complex admittances,

Y_{ij}.

The alternative approach is to use the state variable description of the system: in general any n^{th} order Linear Time Invariant system whose behaviour can be described by a differential equation, of the form

$$f^{n}(x_1,..,x_n)+b_{n-1}f^{n-1}(x_1,..,x_n)+\ldots\ b_1f^{1}(x_1,..,x_n)+b_0f(x_1,..,x_n) = f(t) \quad (3.9)$$

can be re-cast as a set of first order matrix equations:

$$\underline{x}' = \mathbf{A}\cdot\underline{x} + \mathbf{B}\cdot\underline{u} \quad (3.10)$$

$$\underline{y} = \mathbf{C}\cdot\underline{x} + \mathbf{D}\cdot\underline{u} \quad (3.11)$$

where:

$\underline{x}$	= (ns,1) column vector of state variables,
ns	= number of state variables,
$\underline{u}$	= (nc,1) column vector of control variables,
nc	= number of control variables,
$\underline{y}$	= (no,1) column vector of output variables,
no	= number of output variables,
A	= (ns,ns) system matrix,
B	= (ns,nc) input matrix,
C	= (no,ns) output matrix,
D	= (no,nc) direct transmission matrix.

(System Control Technology, 1987; Elgerd, 1967)

There are several advantages to using this representation of the system. Firstly, the effort involved in setting up the simulation is principally concerned with drawing a representative circuit diagram and defining the space-state matrices: this procedure can be mechanized to a large degree by adopting consistent methods of labelling components. Having defined the matrices, packages such as CTRL-C, MATLAB or subroutine libraries such as NAG routines can be applied speedily and easily to yield solutions to a wide variety of cases. The further advantage of couching the description in these terms is that in so doing it is possible to build aggregate systems out of a collection of component systems. This is of especial use when the

characteristics of conditioning plant are known in terms of a transfer function, in either continuous or discrete form. The transfer function can be derived either from theoretical considerations or from experimental measurements of the dynamic response of the plant (Porch and Mo, 1990). The transfer function and space-state descriptions are interconvertible by application of the identity:

$$T(s) = \underline{y}(s)/\underline{u}(s) = \mathbf{C}(sI-A)^{-1}\mathbf{B} + \mathbf{D}$$

$$T(s) = \underline{n}(s)/\underline{d}(s) = \frac{n(1)s^{p-1} + n(2)s^{p-2} +\ldots n(p-1)s + n(p)}{d(1)s^{r-1} + d(2)s^{r-2} +\ldots d(r-1)s + d(r)} \qquad (3.12)$$

where:

$T(s)$ = the Frequency Transfer Function of a system or sub-system;

s = the Laplacian variable = jw;

j = $\sqrt{-1}$;

w = angular frequency (rad/sec);

$\underline{y}$ = column vector of outputs;

$\underline{u}$ = column vector of controls;

$\underline{n}$ = vector containing polynomial coefficients of the numerator of the Frequency Transfer Function (FTF);

$\underline{d}$ = vector containing polynomial coefficients of the denominator of the FTF.

Similar matrix manipulations for aggregating sub-systems in series, parallel, and feedback configurations are also well established.

Another benefit of viewing a thermal system in this way is that it allows performance specifications to be made and calculated. Whilst the margins will never be as closely defined as for electromechanical systems, it is of use to quantify such indicators of

system performance as:

1) Delay Time (TD)

2) Rise Time (TR)

3) Percentage Overshoot (PO)

4) Settling Time (TS)

5) Final Static Value of Error (FVE)

(see Fig.3.9) (Elgerd, 1967).

It is now necessary to demonstrate that it is possible to write the system equations in the state space form. Again a simple, two-walled room will be used to demonstrate the principles before moving on to a real system. Fig.3.10 shows the same room as in Fig.3.1, with the addition of the appropriate capacitive elements. These are intended to model the ability of the building fabric and the enclosed air to store heat[5].

As mentioned earlier there are two ways of solving networks such as the one shown in Fig.3.10: the current mesh technique and nodal analysis. In considering which method to use it is pertinent to assess how many unknowns are to be found in each case. These can be found from the formulae:

$$U_c = B - N + 1 \qquad (3.13)$$

$$U_v = N - 1 \qquad (3.14)$$

where U_c = number of unknowns using the current mesh method,

U_v = number of unknowns using the voltage node analysis,

B = number of branches in the network,

N = number of nodes.

In this case B = 17, N = 6 and consequently $U_c = 12$, $U_v = 5$. It is thus obvious that nodal analysis will be more appropriate in this case.

Fig.3.9: Performance Specifications

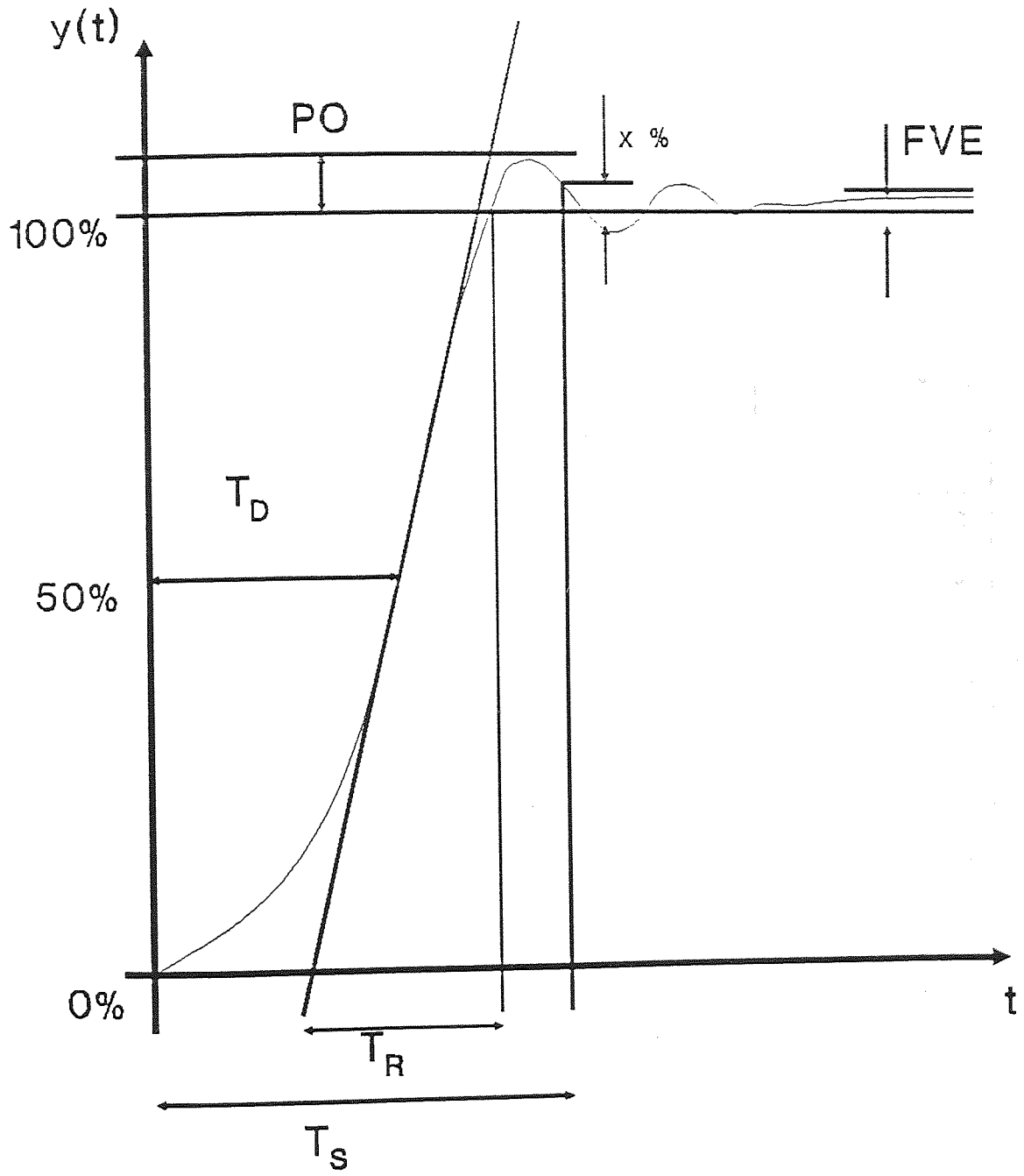
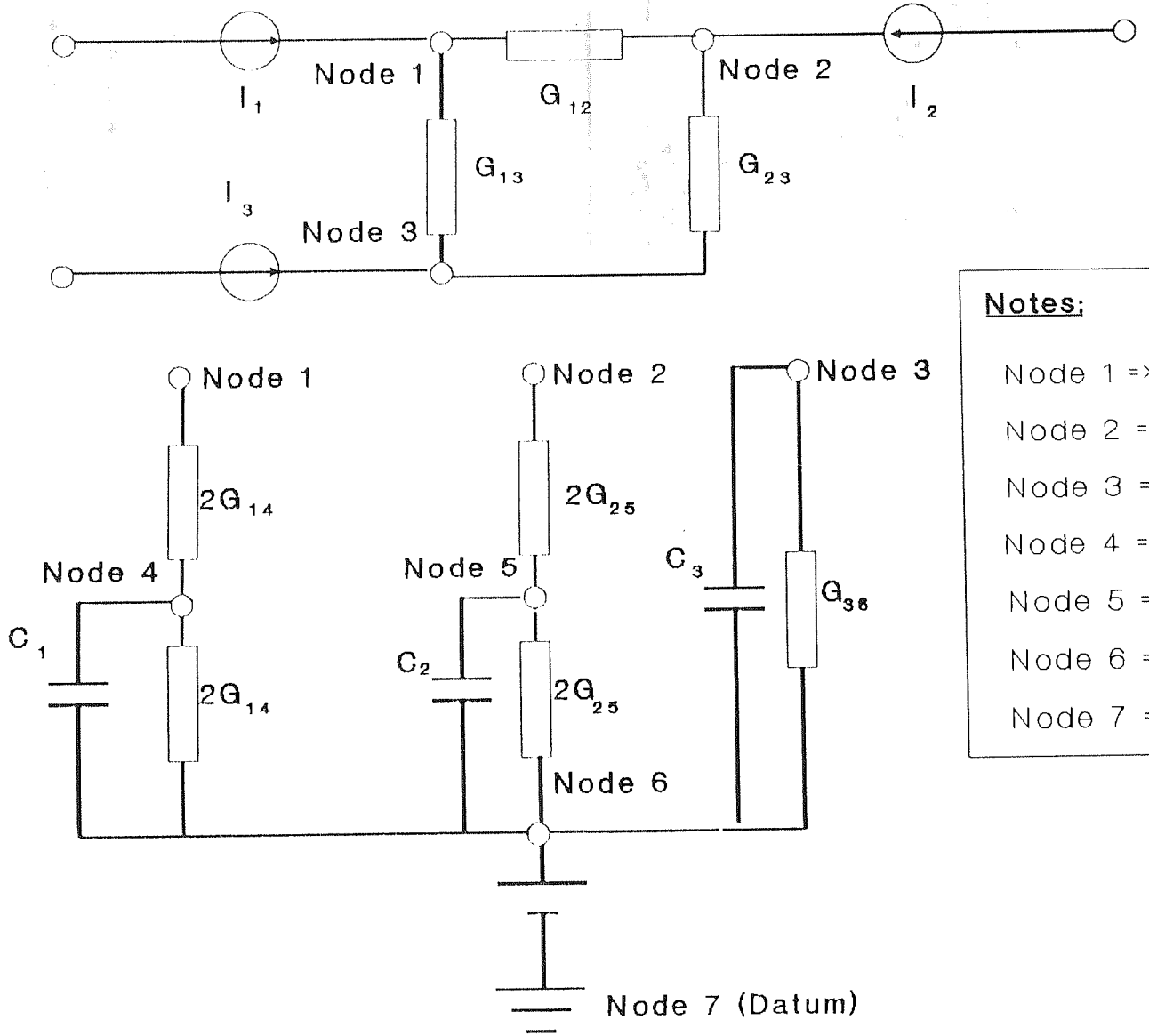


Fig.3.10: Circuit for 2-Surface Room



Notes:

Node 1 => Surface 1

Node 2 => Surface 2

Node 3 => Air Point

Node 4 => Body 1

Node 5 => Body 2

Node 6 => External Air

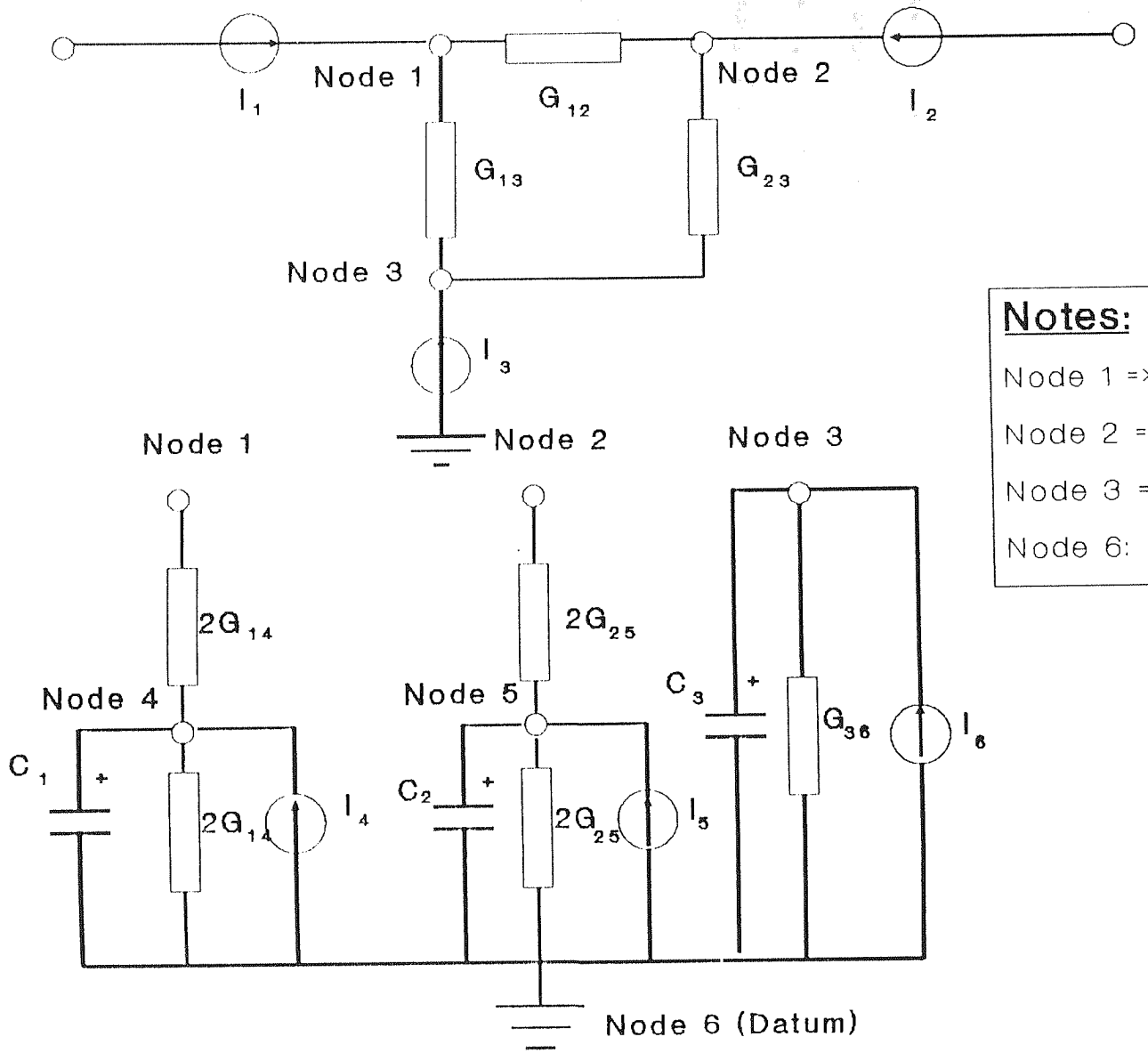
Node 7 => Datum

When using nodal analysis it is generally the case that the currents are known and the voltages are then found in terms of the currents. With the network as presented in Fig.3.10 this presents a problem with the independent voltage generator used to represent the external air temperature. In fact, it becomes impossible to write down the KCL nodal equations in a form which can be solved. The solution to this problem is to firstly push the voltage generator "through" node-4[6] and then replace the voltage generator in series with a resistance, R, with the equivalent current source, ie an independent current generator producing V(t)/R amps, where V(t) is the desired time-dependent voltage input, in parallel with the resistance R (Chan, 1969). This leads to the network shown in Fig.3.11, which turns out to be more amenable to nodal analysis.

Table 3.6: Components for Simple Thermal Circuit with Heat Storage

Component	Description	Value
G_{12}	radiative conductance surface 1=>2	$Hr_{21}.A_1$
G_{13}	convective conductance surface 1=>air	$Ha_1.A_1$
G_{23}	convective conductance surface 2=>air	$Ha_2.A_2$
$G_{14}=G_{46}$	conductive conductance 1=>external	$2.(U_1-Ha_1).A_1$
$G_{25}=G_{56}$	conductive conductance 2=>external	$2.(U_2-Ha_2).A_2$
G_{36}	ventilation conductance air=>external	$c_a.rho_a.N.v/3600$
C_1	thermal capacitance of wall 1	$rho_1.c_1.\delta x_1.A_1$
C_2	thermal capacitance of wall 2	$rho_1.c_2.\delta x_2.A_2$
C_3	thermal capacitance of air	$c_a v$
Hr	radiative HTC	$(W/m^2/K)$
Ha	convective HTC	$(W/m^2/K)$
A	area	(m^2)
U	thermal transmittance	$(W/m^2/K)$
c	specific heat capacity	$(J/kg/K)$
rho	density	(kg/m^3)
δx	thickness	(m)
v	volume	(m^3)

Fig.3.11: Modified 2-Surface Room



Notes:

Node 1 => Surface 1

Node 2 => Surface 2

Node 3 => Air Point

Node 6: Datum Point

Writing the KCL equations for each of the nodes in turn, but excluding the reference node:

NOTE: Summation performed from j=1 to j=6.

Node 1: $I_1 = \Sigma G_{1j}V_{1j} = G_{12}V_{12} + G_{13}V_{13} + G_{14}V_{14}$

Node 2: $I_2 = \Sigma G_{2j}V_{2j}$

Node 3: $C_3V'_3 + I_3 + I_6 = \Sigma G_{3j}V_{3j}$

Node 4: $I_4 + C_1V'_4 = \Sigma G_{4j}V_{4j}$

Node 5: $I_5 + C_2V'_5 = \Sigma G_{5j}V_{5j}$

Where G_{ij} is the symmetric conductance matrix.

$$\begin{vmatrix} 0 & G_{12} & G_{13} & G_{14} & 0 & 0 \\ G_{21} & 0 & G_{13} & 0 & G_{25} & 0 \\ G_{31} & G_{32} & 0 & 0 & 0 & G_{36} \\ G_{41} & 0 & 0 & 0 & 0 & G_{46} \\ 0 & G_{52} & 0 & 0 & 0 & G_{56} \\ 0 & 0 & G_{63} & G_{64} & G_{65} & 0 \end{vmatrix}$$

In order to express the state variables V_3,V_4, and V_5 in the form of Eqn. 3.10 and 3.11 it is necessary to eliminate the output voltages V_1 and V_2 using the equations for nodes 1 and 2. If the equations are combined into two groups gathering nodes 1 and 2 and nodes 3,4 and 5 separately, then the two matrix equations will be of the form

$$\begin{vmatrix} I_1 \\ I_2 \end{vmatrix} = \mathbf{E}\cdot\begin{vmatrix} V_1 \\ V_2 \end{vmatrix} + \mathbf{F}\cdot\begin{vmatrix} V_3 \\ V_4 \\ V_5 \end{vmatrix} \qquad (3.15)$$

$$\begin{vmatrix} V'_3 \\ V'_4 \\ V'_5 \end{vmatrix} = \mathbf{J}\cdot\begin{vmatrix} V_1 \\ V_2 \end{vmatrix} + \mathbf{K}\cdot\begin{vmatrix} V_3 \\ V_4 \\ V_5 \end{vmatrix} + \mathbf{L}\cdot\begin{vmatrix} I_1 \\ I_2 \\ I_3 \\ I_4 \\ I_5 \\ I_6 \end{vmatrix} \qquad (3.16)$$

Solving 3.15 yields

$$\begin{vmatrix} V_1 \\ V_2 \end{vmatrix} = \mathbf{E}^{-1} \cdot \left| \begin{vmatrix} I_1 \\ I_2 \end{vmatrix} - \mathbf{F} \begin{vmatrix} V_3 \\ V_4 \\ V_5 \end{vmatrix} \right| \tag{3.17}$$

which incidentally very nearly corresponds to Eqn. 3.11, the output equations, with a minor addition. Eqn. 3.17 becomes

$$\begin{vmatrix} V_1 \\ V_2 \\ V_3 \end{vmatrix} = \mathbf{C} \cdot \begin{vmatrix} V_3 \\ V_4 \\ V_5 \end{vmatrix} + \mathbf{D} \cdot \begin{vmatrix} I_1 \\ I_2 \\ I_3 \\ I_4 \\ I_5 \\ I_6 \end{vmatrix} \tag{3.18}$$

where $$\mathbf{C} = \begin{vmatrix} -\mathbf{E}^{-1} \cdot \mathbf{F} \\ 1 \quad 0 \quad 0 \end{vmatrix} \quad \text{(output matrix)} \tag{3.19}$$

$$\mathbf{D} = \begin{vmatrix} \mathbf{E}^{-1} & 0 & 0 & 0 & 0 \\ & 0 & 0 & 0 & 0 \\ 0 \quad 0 & 0 & 0 & 0 & 0 \end{vmatrix} \tag{3.20}$$

Substituting Eqn. 3.17 into Eqn. 3.16 gives us

$$\underline{x}' = \mathbf{A} \cdot \underline{x} + \mathbf{B} \cdot \underline{u}$$

where $$\mathbf{A} = \mathbf{K} - \mathbf{J} \cdot \mathbf{E}^{-1} \cdot \mathbf{F} \tag{3.21}$$

$$\mathbf{B} = [1\ 0\ 0\ ] \phi (\mathbf{J} \cdot \mathbf{E}^{-1}) + \mathbf{L} \tag{3.22}$$

where ϕ represents the Kronecker or direct tensor product.

The above example illustrates a general approach to the solution of thermal networks which can be summarised as follows:

1) Determine the electrical equivalent of the thermal network under scrutiny.

2) Determine the values of the conductances and capacitances involved.

3) Write out the KCL equations at all of the nodes bar the datum node.

4) Gather the equations into the matrix form of Eqn. 3.15 and 3.16.

5) Compute the system, input, output and direct transmission matrices from Eqn. 3.19, 3.20, 3.21 and 3.22. Clearly the more complex the problem, the more nodes and consequently the larger the matrices

involved. But the essential point is that the full problem is in principle no more difficult to solve than the simple example.

3.2.2 State-Space Description of a Medium Sized Room

Fig.3.12 shows the details of the full 14-node model of the room thermal network.

Table 3.7: Components for Full Thermal Circuit with Heat Storage

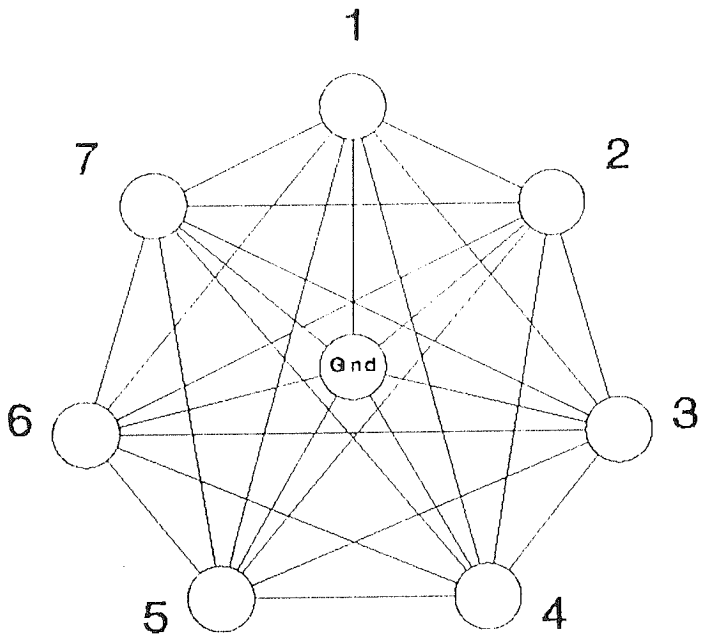
Component	Description	Value
V_i	temperature of the i^{th} node	
I_i^i $(0<i<7)$	heat flux into the i^{th} node	
I_{i+7}^i $(0<i<7)$	equivalent flux to give V_{ext}	
G_{ij}	conductance between i and j^{th} node	
$0<i\&j<7$	radiative conductance surface i=>j	$Hr_{ji}A_i$
$j = 7$	convective conductance surface 1=>air	Ha_iA_i
$j=14$ & $i<>7$	conductive conductance 1=>external	$2(U_i-Ha_i)A_i$
$j=14$ & $i=7$	ventilation conductance air=>external	$c_a.rho_a.N.V/3600$
C_i	thermal capacitance of body i	
$0<i<7$	thermal capacitance of wall i	$rho_i.c_i.\delta x_i.A_i$
$i = 7$	thermal capacitance of air	c_aV

Listing 3.3 shows the CTRL-C procedure (*ROOM_7_N.CTR*) which computes the necessary state space matrices to give the relevant physical information and parameters.

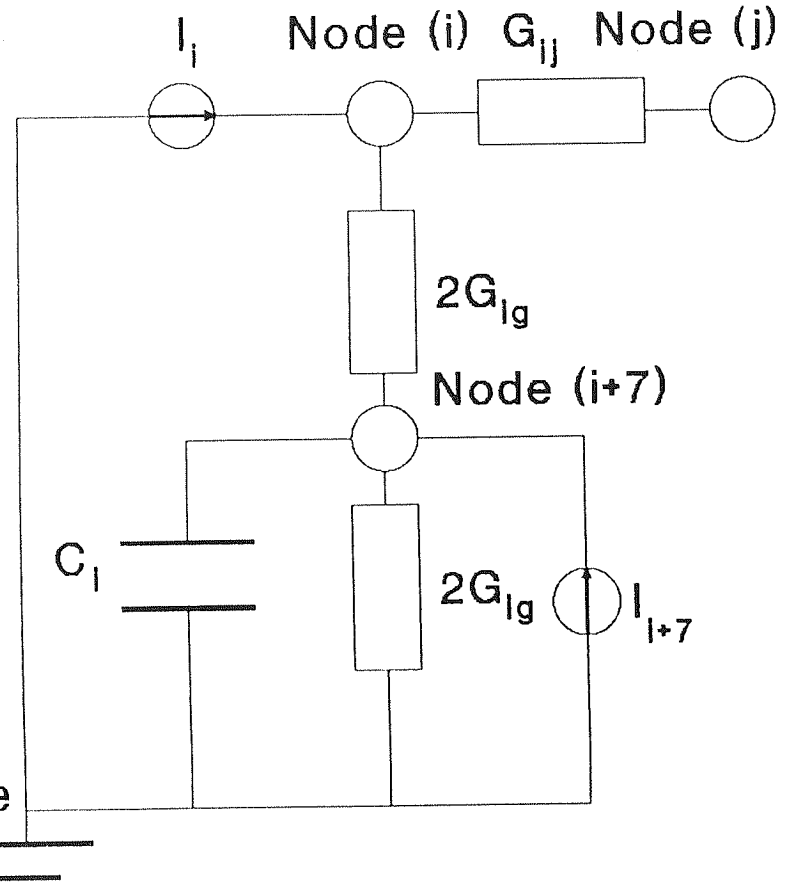
As a demonstration of how powerful state-space analysis can be, two characteristics of the thermal network which are of great importance when considering the control of thermal systems, the Nyquist diagram and the frequency response, will be produced directly from the network matrices using two CTRL-C library procedures, *nyqu* and *bode*.

Fig.3.12: 6-Surface Thermal Network

Network Graph



Components at ith Node



The frequency response is obtained by allowing the vector $\underline{s}$ to vary along the imaginary (jw) axis for the frequency transfer function T(s) defined in Eqn. 3.12. The resulting log-log plot of magnitude against frequency shows how the outputs vary as a function of the frequency of the inputs. For a system with n_c control inputs and n_o outputs this represents (n_c x n_o) different curves. As thermal systems only have "capacitance" but no "inductance" the curves are straightforward; usually they can be characterised by a specific frequency at which the magnitude response hsa been reduced by 3 dB (known as the half-power frequency) and a well-defined roll-off rate, usually quoted in dB/octave. Both of these features can be seen clearly in Fig.3.13. If the graph is studied closely it can be seen that there are two observable characteristic frequencies, one at $w \approx 3x10^{-6}$ rad/s and a much more pronounced one at $w \approx 1x10^{-3}$ rad/s: these correspond to the room and heater time constants respectively. The other important point to note is that the air temperature behaves very differently to the surface temperatures. This is due to its indirect coupling to the heat source, which leads to a lower gain and larger phase shift at "high" ($>1x10^{-3}$ rad/s) frequencies.

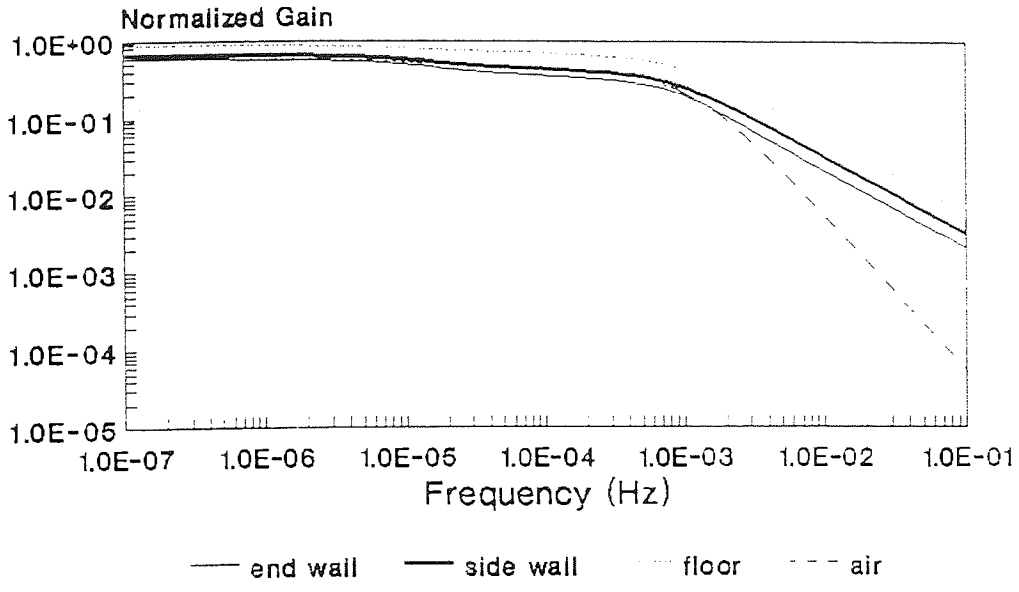
The Nyquist plot is closely related to the frequency response, and in terms of the frequency transfer function, T(s), it is the Argand diagram of W where

$$\begin{aligned} Re(W) &= real(T(jw)) \\ Im(W) &= imag(T(jw)) \end{aligned} \qquad (3.23)$$

Fig.3.14 shows the Nyquist plot for the 6-surface room for the heater input to the controller output. The importance of the Nyquist plot derives from the Nyquist Stability Criterion:

Fig.3.13 : Bode Plot

IRH_1 input to all room outputs



T_ext = 0 C

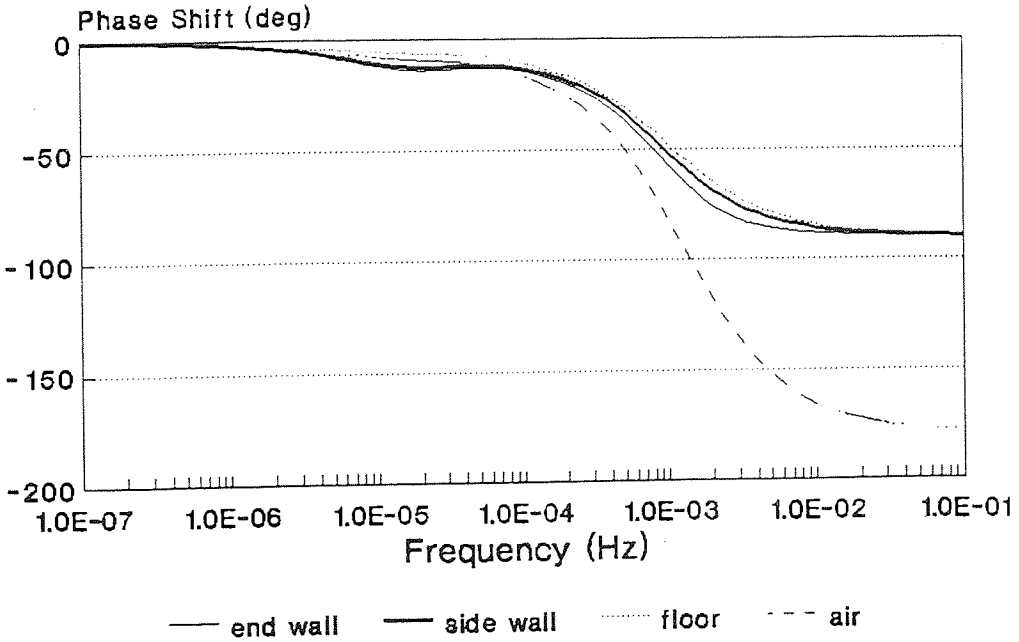
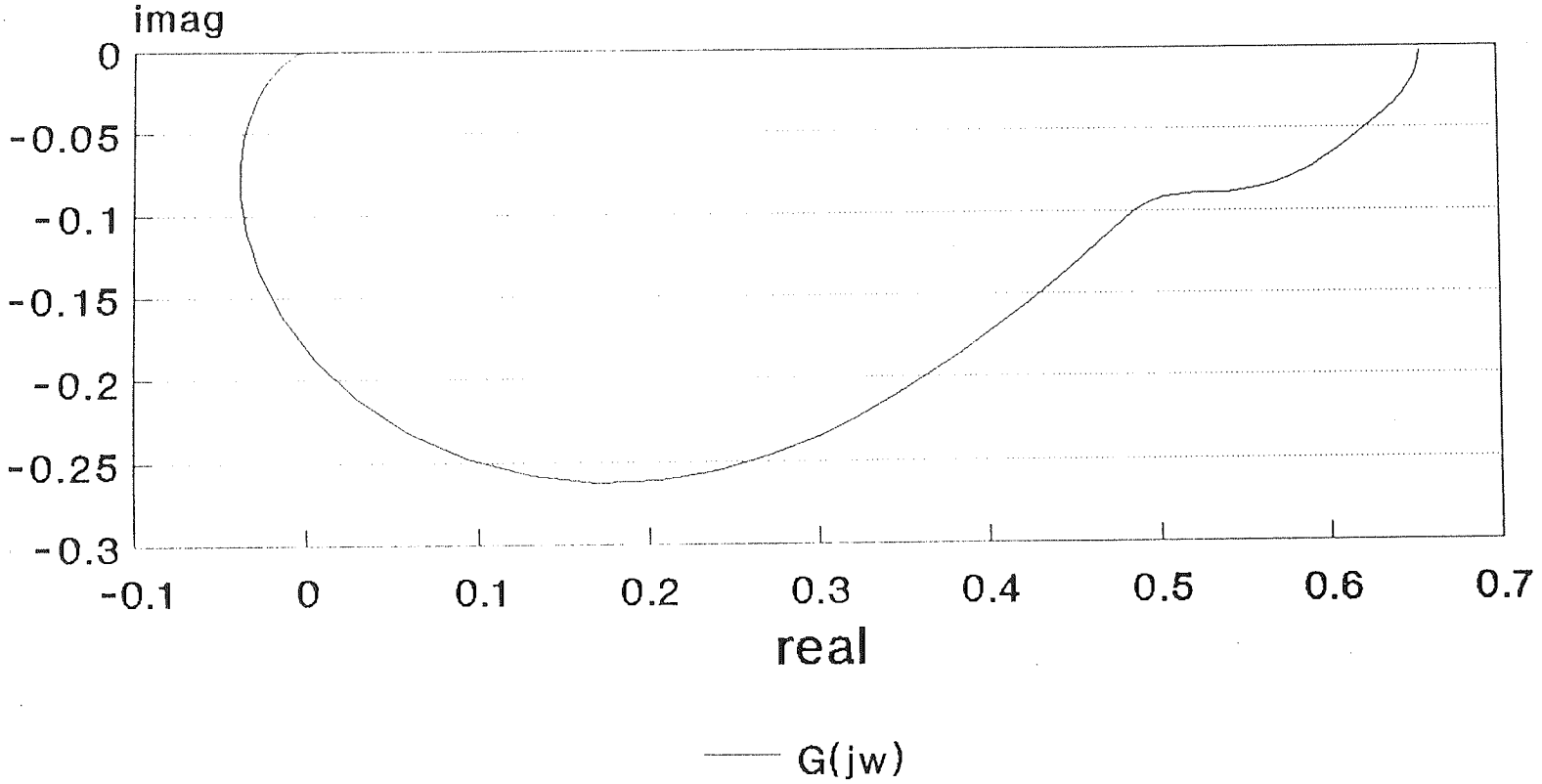


Fig.3.14: Nyquist Plot

IRH_6 in Medium Room

Heater Input to Controller Output



T_ext = 0 C

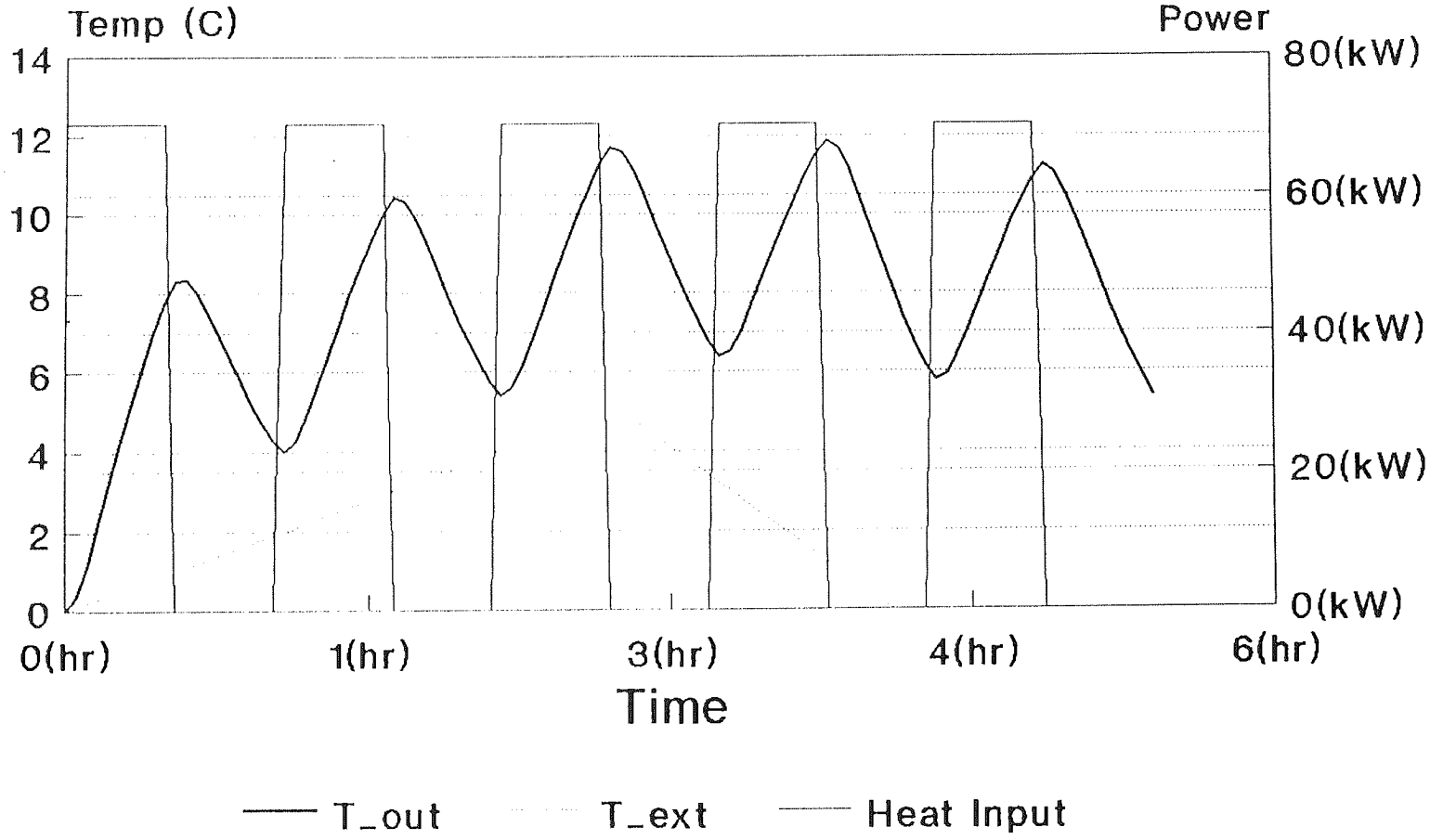
> If the region R, consisting of the entire right half of the s-plane, is mapped onto the T(s) plane, and if the resulting region R' includes the point T(s) = -1, then the corresponding system is unstable. If the point T(s) = -1 is not included in R', the system is stable. (Elgerd, 1967)

This criterion is especially important when a control system for a specific heating system is being designed to condition a particular thermal system. If the Nyquist plot of the whole system can be drawn then it is possible to see whether the system is unconditionally or conditionally stable. If only the latter is the case then it is also possible to predict the phase and gain margins that pertain. Clearly in this instance, with no control system yet specified, the room is unconditionally stable.

Lastly, one of the most useful features of this particular representation is the ability to simulate the time response of the thermal network to arbitrary time-varying inputs. This is achieved in CTRL-C by either the Euler method of rectangular integration or the Adam's predictor/corrector scheme of Shampine and Gordon (Shampine, 1975). This method will be used extensively in the next section to emulate on/off controllers which are otherwise extremely difficult to model. By way of example, Fig.3.15 shows the thermal network's response to the displayed inputs: this demonstrates in the time-domain some of the features of the frequency-domain analysis, namely the characteristic time constants associated with the various system components, and the phase shifting of the high-frequency components of the square-wave and the triangular-wave which form the system inputs.

Fig.3.15: Arbitrary Input

IRH_6 in Medium Room



3.3 Conclusion

A method for deriving the state-space matrices of a heating system consistently and straightforwardly has been expounded. The mathematical and physical principles upon which this method is founded are rigorous and robust. The uncertainties in the technique derive from the aptness of the analogy and the exactitude of the linearised heat transfer coefficients. It has been shown that within the regime in which this simulation is operating the errors are acceptably small.

The method provides a sound basis for exploring the behaviour of thermal systems to time-varying energy inputs and, furthermore, makes it possible to objectively assess the performance of alternative control systems.

Footnotes to Chapter 3

1: The use of a linearized HTC for radiation heat transfer can lead to significant errors if used with insufficient care. The major stipulation for its correct use is that the temperatures of the surfaces which are exchanging radiant energy should not be widely different. A subsidiary point is that when calculating the flux interchange it is necessary to take account of all of the radiation interchanges not just the first exchange. If only the first exchange is taken into account then the corresponding value of the HTC would be found to be:

$$Hr_{21} = \epsilon_1 . \sigma . \Theta_1^{\ 3} . f_{12} \qquad (3.R)$$

In the case of two surfaces which have a general orientation it can be shown (Clarke, 1985) that by summing a geometric series:

$$q_{(2)1,2} = \frac{\epsilon_1 . \epsilon_2 . A . \Theta_1^{\ 4} . f_{1,2}}{1 - (1-\epsilon_1).(1-\epsilon_2).f_{1,2}.f_{2,1}} \qquad (3.S)$$

Furthermore, in the case of n interacting surfaces, the linearized HTC, taking account of all participating surfaces, is shown to be:

$$Hr = \frac{\epsilon_1 . \epsilon_2 . \sigma . f_{1,2} . (\Theta_2^{\ 2} + \Theta_1^{\ 2}) . (\Theta_2 + \Theta_1)}{[1 - (1-\epsilon_1).(1-\epsilon_2).f_{1,2}.A_1/A_2]} + \qquad (3.T)$$

$$\epsilon_1 . \epsilon_2 . \sigma . A . (\Theta_2^{\ 2} + \Theta_1^{\ 2}) . (\Theta_2 + \Theta_1) \sum_{i=1}^{N} \frac{(1-\epsilon_1).f_{1i}.f_{2i}}{A_i.[1 - (1-\epsilon_1).(1-\epsilon_2).(1-\epsilon_i).f_{1i}.f_{i2}.f_{21}]}$$

It is this formulation which is used to evaluate the required HTCs in the CTRL-C function *RADHTC.CTR* (Listing 3.4). It is worth mentioning that in order to check the validity of this approach the procedure *ITV.CTR* was written in such a way as to enable the output temperatures from the first solution to be fed back into the procedure as initial guesses. The following table shows the results of three such iterations:

Table 3.8: Solution Accuracy as a Function of the Number of Iterations

Node	Initial Guess	1st iteration	2nd iteration	3rd iteration
end wall	10.0000	16.1448	16.1615	16.1610
side wall	10.0000	18.5201	18.5090	18.5079
floor	10.0000	26.5920	26.0805	26.0898
ceiling	10.0000	15.3575	15.4969	15.4950
air	0.0000	17.2267	17.1305	17.1322

It can be seen that the first iteration gives good enough results for all but the most demanding of requirements. The use of just one iteration was adopted for all subsequent calculations.

2: The pertinent terms from Network Theory are:

Node: place where two or more components are joined.
Principal Node: Node with three or more components.
Branch: Segment joining two nodes.
Mesh: Interconnection of branches.
Circuit or Loop: Sequence of connected branches within a mesh whose final node is the same as its first node.
Independent voltage (or current) generator: one whose output voltage(or current) is independent of the circuit connected across its terminals.

3: Σ signifies the sum over the appropriate dummy index.

4: The designations for the spacings of "-1 m" and "-2 m" signify that all of the heaters were displaced towards the centre of the room by the amount quoted. This results in a progressively more "bunched" arrangement, with the heaters being clustered closer together than for "EQUAL" spacing.

5: The representation of heat storage within the electrical analogy of thermal circuits is a contentious issue. At least two major references Fisk (1981) and Clarke (1985) incline to the use of the electrical "T-section" with complex impedances which appear to originate from the solution to the subsidiary equations arising from a specific Laplace Transform (Carslaw and Jaeger, 1959). Both have good reason for choosing this method; in both instances this reduces to a large extent to the fact that it is mathematically convenient for their expositions. There are two comments that should be made about this approach: by deciding to use complex impedances, the original directness of the analogy is sacrificed since there is no passive electrical component which has the properties cited; further, there is some doubt as to whether this approach is any more accurate than the alternative - which is the to use a T-section comprising of two resistors and a capacitor to ground - insofar as the latter has been used extensively in analogue computing with satisfactory results (Paschkiss, 1944). Furthermore, at least one author (Gebhart, 1971) cites this more direct analogy with approval. Thus it is proposed to use this alternative analogy here.

6: In topological terms this is known as a "Blakesley transformation" or alternatively as the application of the "ϵ-shift operator" (Chan, 1969). Physically speaking an equivalent circuit replacement of one independent voltage generator by n parallel independent voltage generators is being made (Chirlian, 1969)

4 Control of Radiant Heating Systems

In this section an introduction is given to the use of state variable analysis to provide open-loop responses of systems which include plant, room, and controller dynamics. A step forward towards a pseudo-closed loop system is then taken by implementing a switching algorithm in software which emulates the on/off controllers commonly used to control radiant heaters. The aim of this is to produce time-domain characteristics of systems to illustrate how different controllers and plant affect temperatures and fuel consumption.

The full closed-loop responses cannot be derived in the same format as has been previously employed because of the essentially non-linear characteristics of a simple on/off switch. It is, however, possible to investigate how controller characteristics affect system response and also the effect of switching strategies in multi-heater systems using this approach. These two areas alone are of sufficient interest to warrant the attempt to model system behaviour in this way.

4.1 Connecting Plant, Room and Controller Sub-Systems

In section 3.2.1 it was suggested that one of the attractions of using state-variable analysis was that larger systems could be easily constructed out of smaller building blocks. Fig.4.1 shows some of the common interconnections. It is pertinent to see how these connections affect the state space matrices.

Firstly for a series connection, the component system equations are:

$$\underline{x}_1' = \mathbf{A}_1\cdot\underline{x}_1 + \mathbf{B}_1\cdot\underline{u}_1; \qquad \underline{y}_1 = \mathbf{C}_1\cdot\underline{x}_1 + \mathbf{D}_1\cdot\underline{u}_1 \tag{4.1}$$

$$\underline{x}_2' = \mathbf{A}_2\cdot\underline{x}_2 + \mathbf{B}_2\cdot\underline{u}_2; \qquad \underline{y}_2 = \mathbf{C}_2\cdot\underline{x}_2 + \mathbf{D}_2\cdot\underline{u}_2 \tag{4.2}$$

As in the diagram, the linkage is made by routing the outputs from S_1 to the inputs of S_2; clearly these must be numerically equal to one another. The resulting matrix equation is:

$$\underline{x}' = \mathbf{A}\cdot\underline{x} + \mathbf{B}\cdot\underline{u} \;\; ; \; \underline{y} = \mathbf{C}\cdot\underline{x} + \mathbf{D}\cdot\underline{u} \tag{4.3}$$

where $\mathbf{A} = \begin{vmatrix} \mathbf{A}_1 & 0 \\ \mathbf{B}_2\cdot\mathbf{C}_1 & \mathbf{A}_2 \end{vmatrix}$

$\mathbf{B} = \begin{vmatrix} \mathbf{B}_1 \\ \mathbf{B}_2\cdot\mathbf{D}_1 \end{vmatrix}$

$\mathbf{C} = \begin{vmatrix} \mathbf{D}_2\cdot\mathbf{C}_1 & \mathbf{C}_2 \end{vmatrix}$

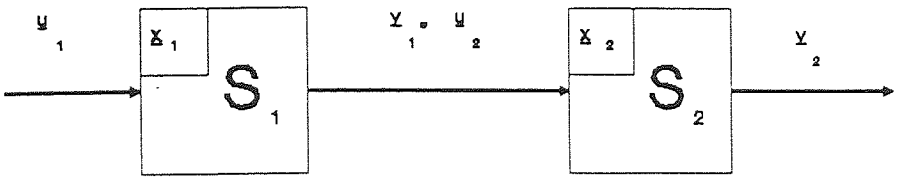
$\mathbf{D} = \begin{vmatrix} \mathbf{D}_2\cdot\mathbf{D}_1 \end{vmatrix}$

$\underline{x} = \begin{vmatrix} \underline{x}_1^{\tau} : \underline{x}_2^{\tau} \end{vmatrix}^{\tau}$

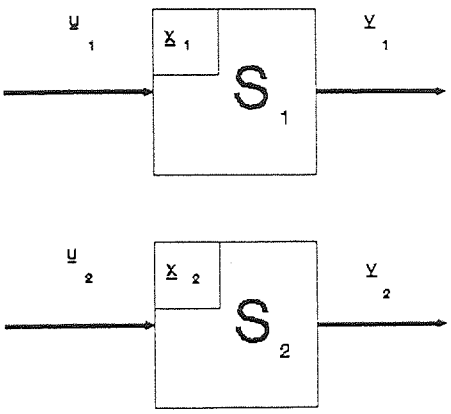
For the parallel connection there are three possibilities to consider: uncoupled inputs and outputs, coupled inputs with uncoupled outputs , and uncoupled inputs with coupled outputs. The various values of the

Fig 4.1: Common Interconnections

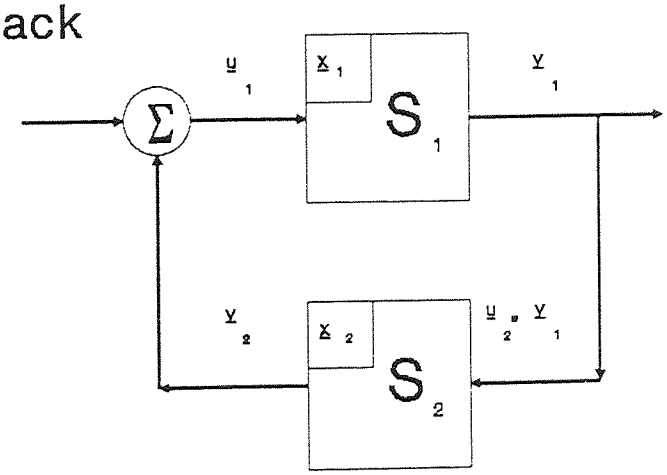
a) Series



b) Parallel



c) Feedback



matrices and vectors thus formed are summarized below:

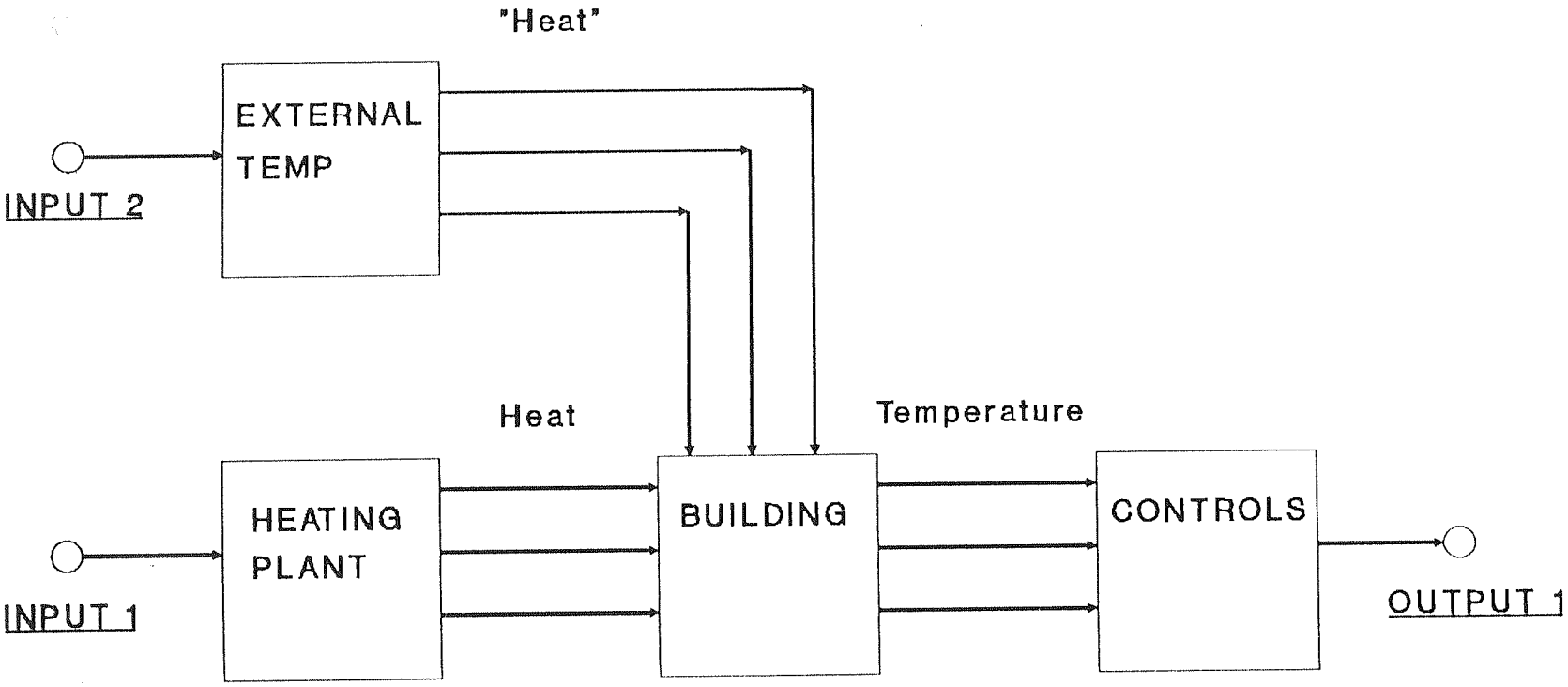
Table 4.1 Space State Matrices for Parallel Interconnections

Matrix	$\underline{u}_1 \neq \underline{u}_2; \underline{y}_1 \neq \underline{y}_2$	$\underline{u}_1 = \underline{u}_2; \underline{y}_1 \neq \underline{y}_2$	$\underline{u}_1 \neq \underline{u}_2; \underline{y}_1 = \underline{y}_2$
A	$\begin{vmatrix} A_1 & : & 0 \\ 0 & : & A_2 \end{vmatrix}$	$\begin{vmatrix} A_1 & : & 0 \\ 0 & : & A_2 \end{vmatrix}$	$\begin{vmatrix} A_1 & : & 0 \\ 0 & : & A_2 \end{vmatrix}$
B	$\begin{vmatrix} B_1 & : & 0 \\ 0 & : & B_2 \end{vmatrix}$	$\begin{vmatrix} B_1 \\ B_2 \end{vmatrix}$	$\begin{vmatrix} B_1 & : & 0 \\ 0 & : & B_2 \end{vmatrix}$
C	$\begin{vmatrix} C_1 & : & 0 \\ 0 & : & C_2 \end{vmatrix}$	$\begin{vmatrix} C_1 & : & 0 \\ 0 & : & C_2 \end{vmatrix}$	$\begin{vmatrix} C_1 \\ C_2 \end{vmatrix}$
D	$\begin{vmatrix} D_1 & : & 0 \\ 0 & : & D_2 \end{vmatrix}$	$\begin{vmatrix} D_1 & : & 0 \\ 0 & : & D_2 \end{vmatrix}$	$\begin{vmatrix} D_1 & : & 0 \\ 0 & : & D_2 \end{vmatrix}$
$\underline{x}$	$\begin{vmatrix} \underline{x}_1 \\ \underline{x}_2 \end{vmatrix}$	$\begin{vmatrix} \underline{x}_1 \\ \underline{x}_2 \end{vmatrix}$	$\begin{vmatrix} \underline{x}_1 \\ \underline{x}_2 \end{vmatrix}$

Using these simple matrix manipulations large systems may be constructed by the aggregation of simpler sub-systems. The principles of constructing larger systems will be demonstrated using the simplified 2-surface room before citing the results obtained from the full 6-surface model. The block diagram of the complete system including plant, room and controller is shown in Fig.4.2.

The dynamics of the heater may be approximated by the transfer function

Fig.4.2: Control System (5 node case)



$$Q_{htr}(s) = \frac{Q_{out}}{(1 + \tau.s)} \qquad (4.4)$$

where Q_{htr} = Instantaneous power of the heater (W)

s = Laplacian operator (Hz)

Q_{out} = Steady state output of the heater (W)

τ = Time constant of heater (s)

This is an exponential lag of time constant τ with a DC-gain of Q_{out}^{1}. Obviously this only provides a single-input-single-output system whereas a single-input-multiple-output is required. This can be achieved by using the configuration in Fig.4.3 to give the output into each of the appropriate nodes of the room. The resulting equations are:

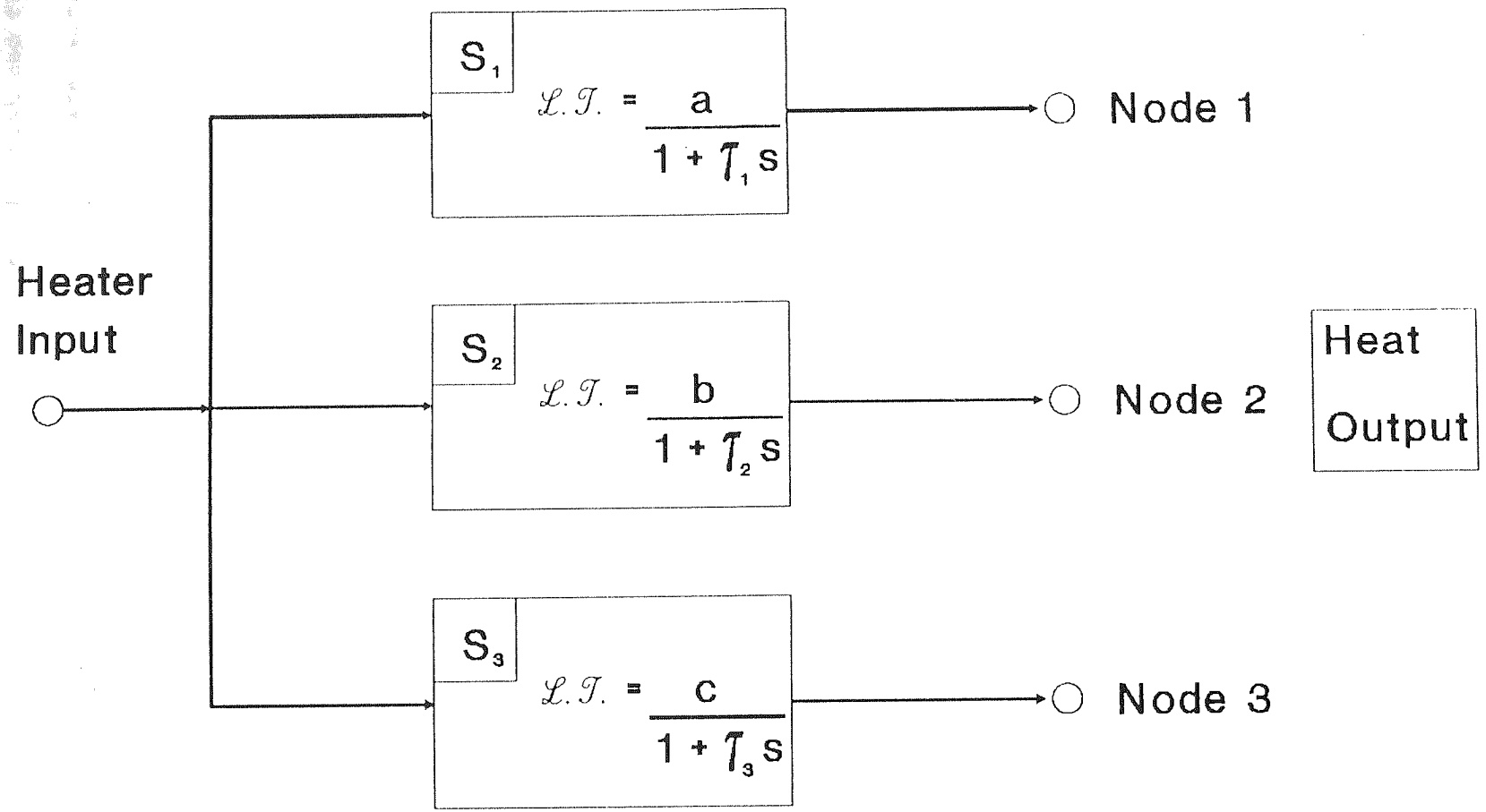
$$\begin{vmatrix} x_1' \\ x_2' \\ x_3' \end{vmatrix} = \begin{vmatrix} -\tau_1^{-1} & 0 & 0 \\ 0 & -\tau_2^{-1} & 0 \\ 0 & 0 & -\tau_3^{-1} \end{vmatrix} \cdot \begin{vmatrix} x_1 \\ x_2 \\ x_3 \end{vmatrix} + \begin{vmatrix} a \\ b \\ c \end{vmatrix} \cdot u \qquad (4.5)$$

$$\begin{vmatrix} y_1 \\ y_2 \\ y_3 \end{vmatrix} = \begin{vmatrix} \tau_1^{-1} & 0 & 0 \\ 0 & \tau_2^{-1} & 0 \\ 0 & 0 & \tau_3^{-1} \end{vmatrix} \cdot \begin{vmatrix} x_1 \\ x_2 \\ x_3 \end{vmatrix} + \begin{vmatrix} 0 \\ 0 \\ 0 \end{vmatrix} \cdot u$$

In this instance, the control matrix represents the weighting of the outputs of the heater into the various nodes and it is the values of the constants a, b, and c that were determined by the Monte Carlo model. The same method that was used in Section 3.1.3 to determine the heat flow into the nodes of the resistive network can be used to find a,b, and c, the only difference being that it is necessary to normalize the coefficients.

This is the situation, then, for one heater. If several

Fig.4.3: Model of Heater Dynamics



heaters are required the procedure is essentially the same, that is subsequent heaters are just added in parallel with the first. The only practical effect, assuming that identical units are being modelled, is to alter the values of the weighting vector, because heaters in different positions would contribute by varying amounts to the different nodes. Incidentally, although the values of the time constants in Eqn.4.5 are different, in all practical cases they would be set to the same value.

Next, Fig.4.4 shows the external temperature simulator which is necessary because of the way the external conditions are represented in the room model. For the un-normalized case:

$$\mathbf{D} = \begin{vmatrix} G_1 \\ G_2 \\ G_3 \end{vmatrix} \qquad \text{while } \mathbf{A} = \mathbf{B} = \mathbf{C} = \mathbf{0} \tag{4.6}$$

In order to obtain a correctly normalized value of the direct transmission matrix it is necessary to compute:

$$\mathbf{D}_n = \mathbf{N}_y \backslash \mathbf{D} \cdot \mathbf{N}_u \tag{4.7}$$

where $$\mathbf{N}_y = |T_{max}| \quad \text{(Output normalization matrix)} \tag{4.8}$$

$$\mathbf{N}_u = \begin{vmatrix} Q_{max} & 0 & 0 \\ 0 & Q_{max} & 0 \\ 0 & 0 & Q_{max} \end{vmatrix} \quad \text{(Control normalization matrix)}$$

and T_{max} is the maximum temperature, whilst Q_{max} is the maximum power.

Lastly, Fig.4.5 shows how the controller is implemented in the 2-surface room. In this case the values of the weighting determine to what extent the controller is affected by the various temperatures of the nodes. This is a particularly useful feature, since if the approximation:

Fig.4.4: Simulation of External Conditions

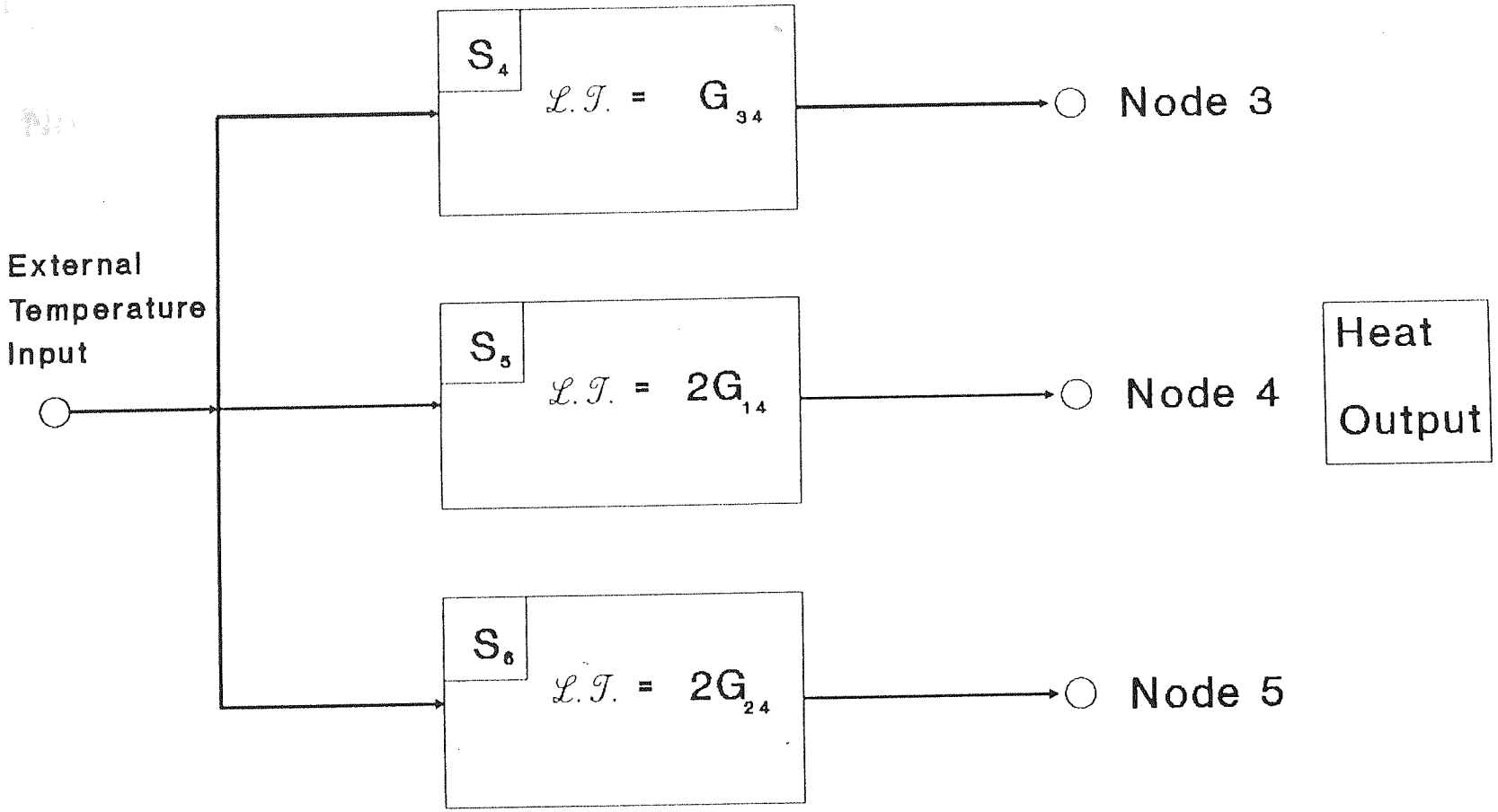
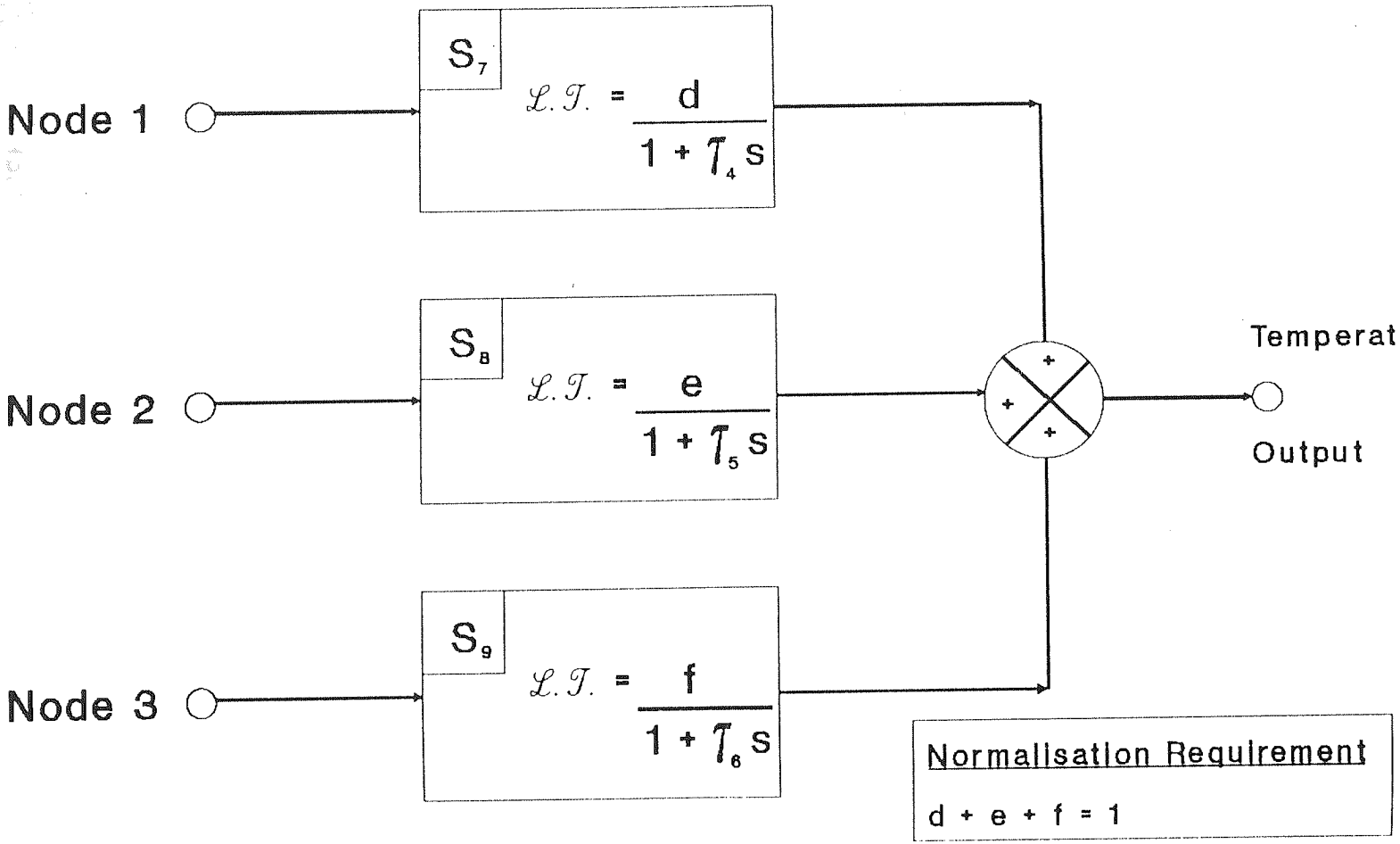


Fig.4.5: Model of Controller Dynamics



$$T_{mrt} = \Sigma A_j . T_j / \Sigma A_j \qquad (4.9)$$

where T_{mrt} = Mean radiant temperature (°C)

A_j = Area of j^{th} surface (m^2)

T_j = Temperature of j^{th} surface (°C) (CIBSE,1986)

is used, it is possible to evaluate different methods of controlling radiant heating systems, principally by changing the weighting factor between T_{mrt} and the internal air temperature (T_{ai}).

The state space matrices for the controller are:

$$\begin{vmatrix} x_1' \\ x_2' \\ x_3' \end{vmatrix} = \begin{vmatrix} -\tau^{-1} & 0 & 0 \\ 0 & -\tau^{-1} & 0 \\ 0 & 0 & -\tau^{-1} \end{vmatrix} \cdot \begin{vmatrix} x_1 \\ x_2 \\ x_3 \end{vmatrix} + \begin{vmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{vmatrix} \cdot u \qquad (4.10)$$

$$|y| = \begin{vmatrix} \tau^{-1} & \tau^{-1} & \tau^{-1} \end{vmatrix} \cdot \begin{vmatrix} x_1 \\ x_2 \\ x_3 \end{vmatrix} + \begin{vmatrix} 0 & 0 & 0 \end{vmatrix} \cdot u$$

where τ = controller time constant (s)

a = fractional contribution from surface 1

b = fractional contribution form surface 2

c = fractional contribution from air

4.2 The Open-Loop Response

Having derived the necessary state-space matrices for the relevant system components for the 2-surface room in the last section, it is now necessary to specify the interconnections for the 6-surface room which are shown in schematic form in Fig.4.6. This is exactly the same form as Fig.4.2, the only increase in complexity being due to the large number of state variables. There are 28 in the full simulation as opposed to just 12 in the 2-surface room. Listing 4.1 shows the annotated CTRL-C procedure used to set up the matrices.

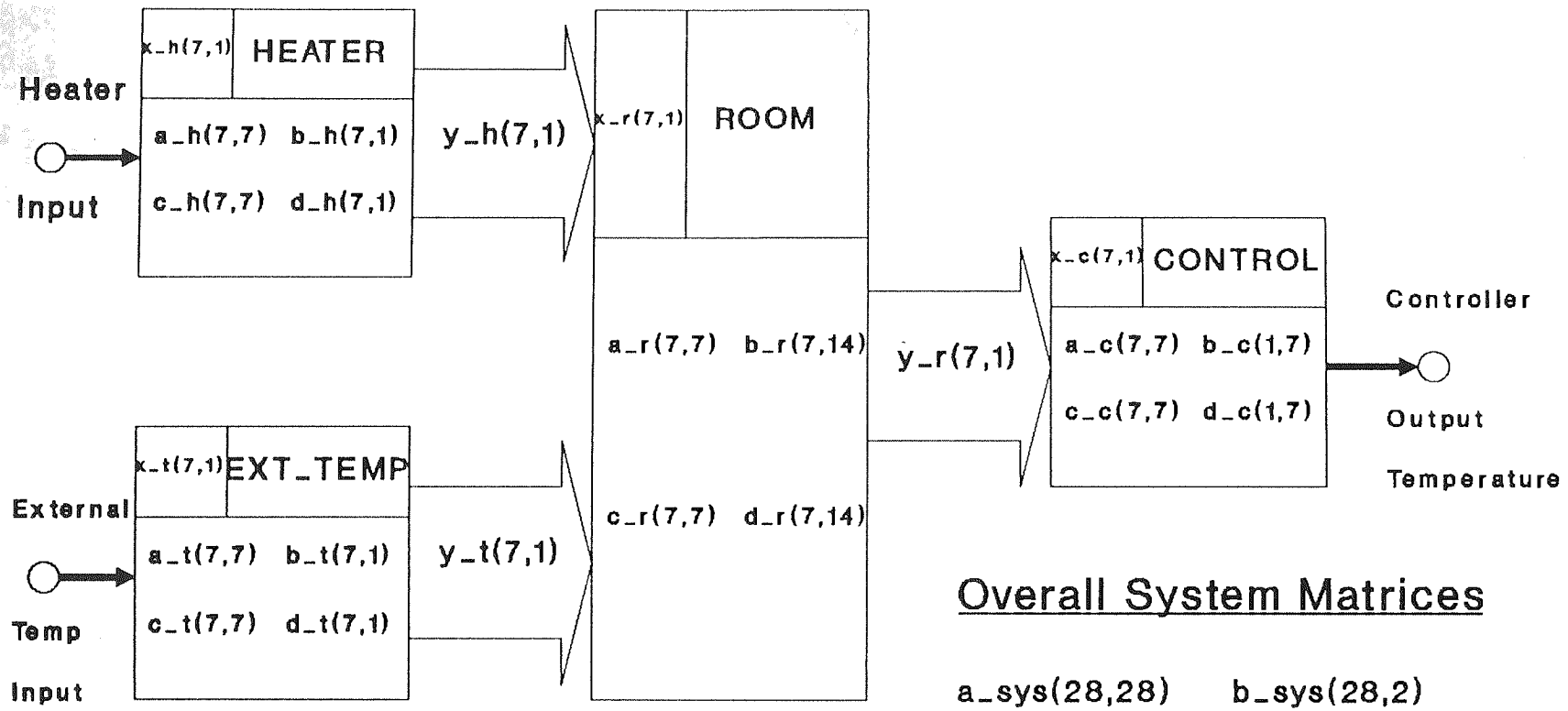
4.2.1 Time-Domain Response

The controller will be of importance in the next section when the closed-loop response of the system is studied. For the time being it can be ignored and attention focussed on the response of the outputs from the room, ie the temperatures of the surfaces and the air, to the two inputs.

Before looking at the transfer characteristics of the system in detail it is important to check that the description of the system satisfies two requirements. Firstly, the principle of the conservation of energy must not be contravened. Secondly, for long time-periods the temperatures produced agree with those predicted by the static model.

The results of a typical simulation are shown in Table 4.2: the heat flux matrix Q displays several noteworthy features. Firstly, it is skew-symmetric, i.e. $Q_{ij} = -Q_{ji}$, which ensures compliance with the conservation of energy. Secondly, the sum of any row is the heat flowing into that node, while the sum of a column is the flow of heat out of the node. Thirdly, it can be seen that the sum of power into nodes 1 to 7 is the same as the input from the heating system,

Fig.4.6: Schematic Equivalent to ROOM_7_N.CTR



Overall System Matrices

a_sys(28,28) b_sys(28,2)

c_sys(1,28) d_sys(1,2)

Table 4.2: The Heat Flux and the Conductance Matrices

Model parameters:

Heater:	time constant	= 600	sec	(MEDIUM)
	output	= 10.000	kW	(MEDIUM)
Building:	thermal weight	= 761.4	MJ/K	(MEDIUM)
	size	= 30x10x8	m	(MEDIUM)
	air change rate	= 1.0	/hr	
	modus operandi	= FULL ON		
External Conditions:	external temperature	= 0 °C (throughout)		

Model Conditions:

Initial conditions:	nodal temperatures	= all 0 °C
Heater mounting height:		= 5.00 m
Heater type:		= IRH_1
Number of heaters:		= 4
Number of controls:		= N/A
Period of simulation:		= 1 week

Heat Flux Matrix, Q (W)

NODE	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	0.	-61.	-114.	4.	-38.	-28.	-545.	3434.	0.	0.	0.	0.	0.	0.
2	61.	0.	-695.	290.	179.	115.	948.	0.	8065.	0.	0.	0.	0.	0.
3	114.	695.	0.	11313.	3159.	1561.	18751.	0.	0.	5564.	0.	0.	0.	0.
4	-4.	-290.	-11313.	0.	-780.	-328.	-3966.	0.	0.	0.	19161.	0.	0.	0.
5	38.	-179.	-3159.	780.	0.	23.	186.	0.	0.	0.	0.	7619.	0.	0.
6	28.	-115.	-1561.	328.	-23.	0.	-3.	0.	0.	0.	0.	0.	3753.	0.
7	545.	-948.	-18751.	3966.	-186.	3.	0.	0.	0.	0.	0.	0.	0.	22709.
8	-3434.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	3428.
9	0.	-8065.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	8053.
10	0.	0.	-5564.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	1679.
11	0.	0.	0.	-19161.	0.	0.	0.	0.	0.	0.	0.	0.	0.	19153.
12	0.	0.	0.	0.	-7619.	0.	0.	0.	0.	0.	0.	0.	0.	7604.
13	0.	0.	0.	0.	0.	-3753.	0.	0.	0.	0.	0.	0.	0.	3746.
14	0.	0.	0.	0.	0.	0.	-22709.	-3428.	-8053.	-1679.	-19153.	-7604.	-3746.	0.
SUM	-2652.	-8964.	-41156.	-2479.	-5309.	-2408.	-7338.	6.	12.	3885.	8.	15.	7.	66372.

Conductance Matrix, G (W/K)

NODE	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	0.	9.	7.	7.	9.	9.	240.	282.	0.	0.	0.	0.	0.	0.
2	9.	0.	198.	198.	286.	78.	720.	0.	846.	0.	0.	0.	0.	0.
3	7.	198.	0.	506.	309.	97.	1290.	0.	0.	169.	0.	0.	0.	0.
4	7.	198.	506.	0.	309.	97.	1290.	0.	0.	0.	1514.	0.	0.	0.
5	9.	286.	309.	309.	0.	78.	720.	0.	0.	0.	0.	846.	0.	0.
6	9.	78.	97.	97.	78.	0.	240.	0.	0.	0.	0.	0.	282.	0.
7	240.	720.	1290.	1290.	720.	240.	0.	0.	0.	0.	0.	0.	0.	800.
8	282.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	282.
9	0.	846.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	846.
10	0.	0.	169.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	169.
11	0.	0.	0.	1514.	0.	0.	0.	0.	0.	0.	0.	0.	0.	1514.
12	0.	0.	0.	0.	846.	0.	0.	0.	0.	0.	0.	0.	0.	846.
13	0.	0.	0.	0.	0.	282.	0.	0.	0.	0.	0.	0.	0.	282.
14	0.	0.	0.	0.	0.	0.	800.	282.	846.	169.	1514.	846.	282.	0.

whilst the sum of the nodes 8 to 13 is the power being absorbed into the fabric. In this case an appreciable amount of power (3.6 kW) is still being absorbed even after a week of continuous operation. The power absorbed by the fabric diminishes asymptotically to zero, its values at various times being:

Q_{in} = 70.308 kW

Q_{out}(1 week) = 66.372 kW; δQ = 3936 W

Q_{out}(2 weeks) = 68.151 kW; δQ = 2157 W

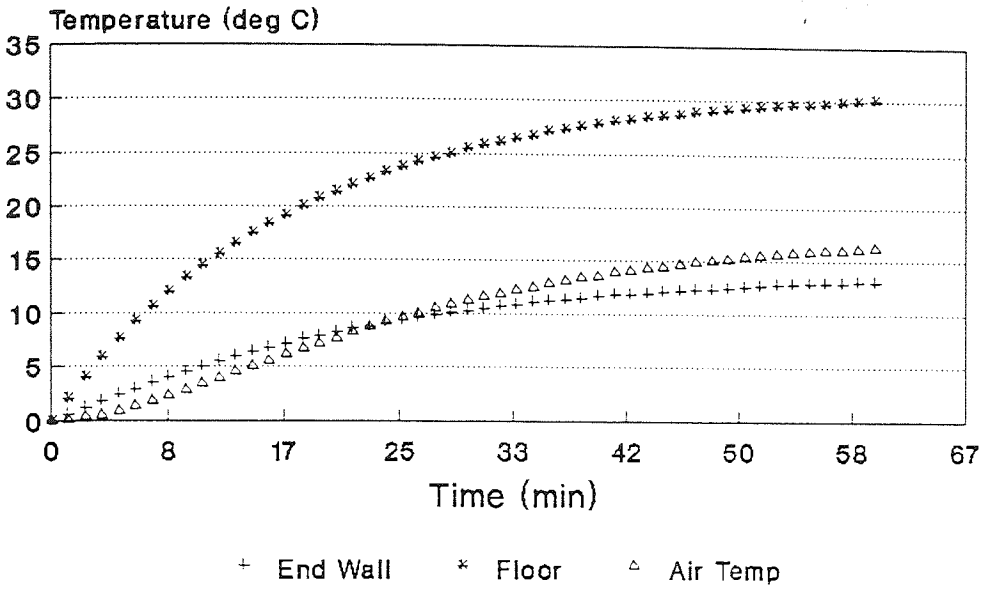
Q_{out}(4 weeks) = 69.660 kW; δQ = 648 W

Q_{out}(8 weeks) = 70.249 kW; δQ = 59 W

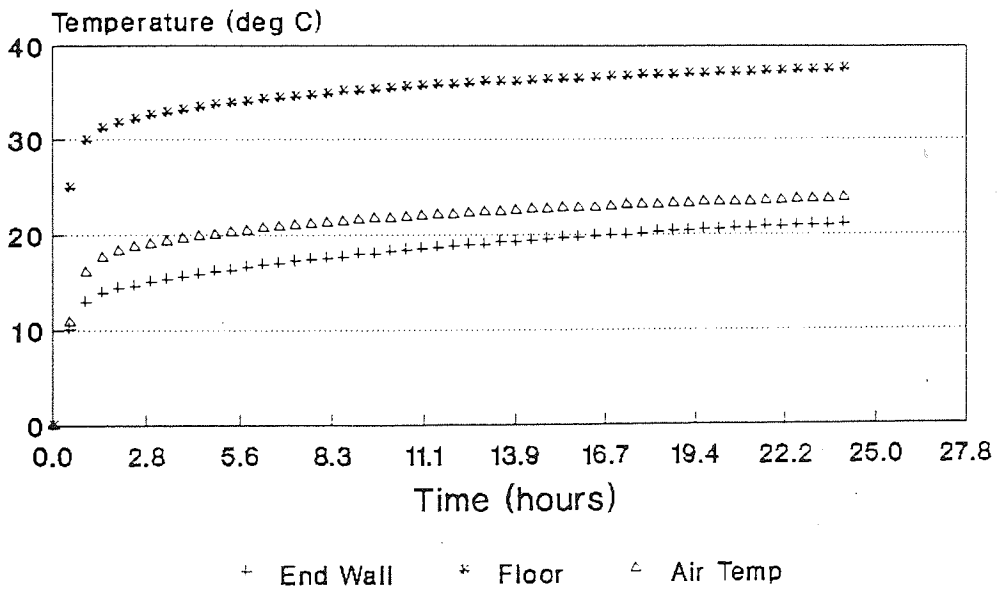
The response of the model to a step change applied at the heaters' input represents what happens when the heating system is turned on; this is depicted in Fig.4.7 for two different time-scales - an hour, and a day. The functional form of the step response is best explained in terms of the interplay between several different time constants along with consideration of the conductance matrix. The values of the time constants - which are equal to the thermal capacitance divided by the conductance - determine how quickly a node will heat up, whilst the conductances are an indication of how strongly the nodes are coupled to each other. Thus, looking at the step response over an hour, it is apparent that the floor surface temperature rises rapidly because it is directly coupled to the heaters with no capacitive loading. By contrast the air, which is also receiving an appreciable heat input, warms up more slowly because of its intrinsic time constant of about an hour. An interesting point is that the shape of the air curve differs from the surface curves when the time interval is less than 20 minutes due to the indirect heating

Fig 4.7: System Step Response

Period = 1 hour



Period = 1 Day



floor, which accelerates the rate of heating but only after the floor itself has begun to warm up.

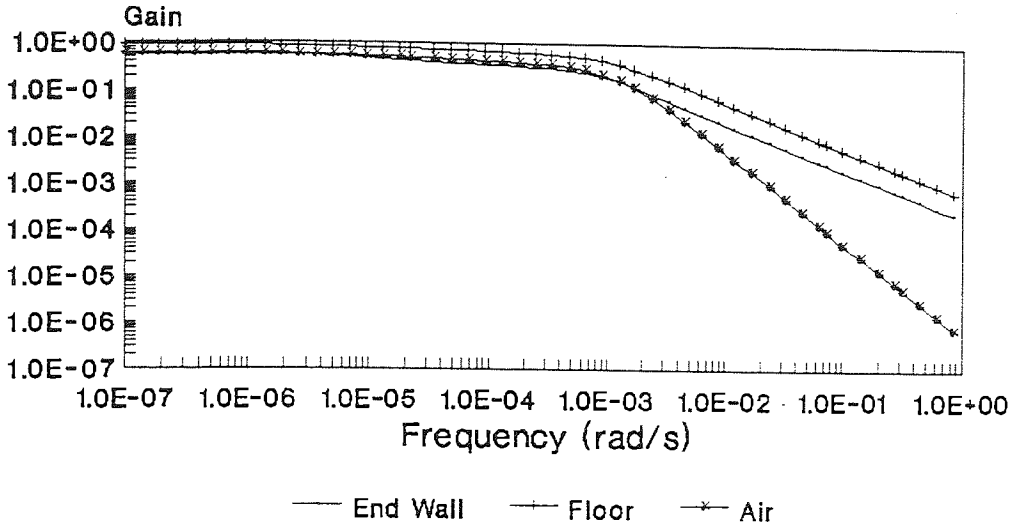
Moving on to the longer time-scale, it is noticeable how the change over from the dependence of the temperature rise on the heater time constant (≈ 10 minutes) to a dependence on the longer time constant of the building structure (≈ 1 day) occurs at around an hour. Up until that point the type of heater chosen will make a difference to the system response; afterwards it is the importance of the nature of the building which predominates.

4.2.2 Frequency-Domain Response

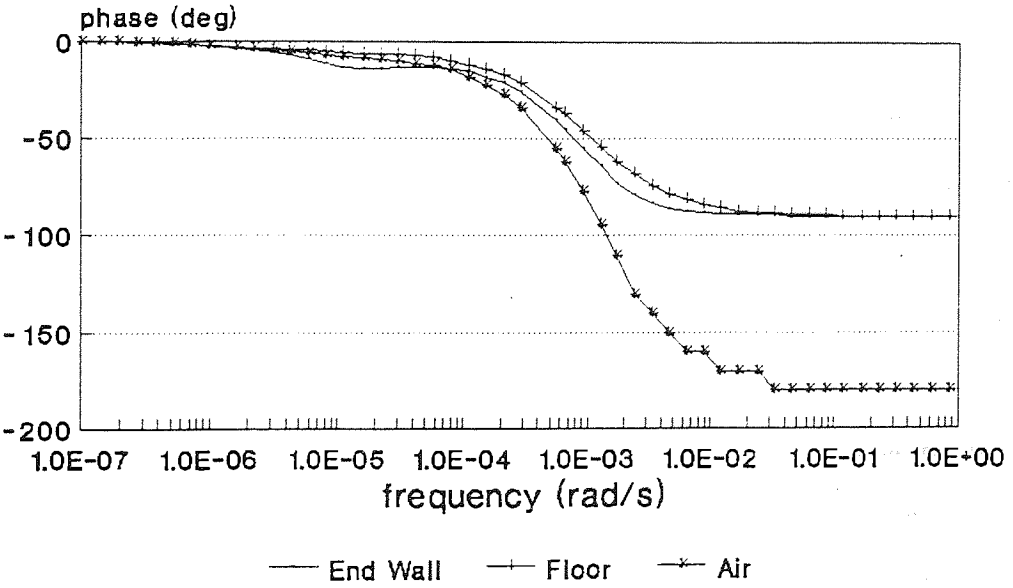
When dealing with the time-domain the graphs that are produced by plotting the temperature against time are self-explanatory. In the frequency domain things are not always so straightforward. Historically it has proved necessary to develop certain methods for plotting the characteristics of interest in order to show clearly their import. The Bode plot is a good example of just such a development. It is a means of conveying both the gain and phase information of a system's response to an wide range of frequencies. The top half of the plot is a log-log graph of gain against frequency, whilst the bottom half is a log-linear plot of phase against frequency. The advantage of this particular portrayal of the information is that it can display a frequency dependence over a number of decades in a compact form, which is also physically relevant. Furthermore important information such as cut-off frequencies and roll-off rates can be read by eye from the graphs as they can easily be distinguished in this particular form.

Fig.4.8 shows the bode plots of these outputs to the first input, i.e. the input to the heating system. The three curves

Fig.4.8: Bode Plot
Room Temperature Response to Heater Input



Phase



displayed are representative of the seven outputs of the room.

On close examination the curves for the surfaces display two time-constants, one at $w \approx 10^{-5}$ rad/s (τ = 1 week) which is associated with the building structure, and the other at $w \approx 10^{-3}$ rad/s (τ = 1.75 hr) which is associated with the time constant of the air (see footnote 4.1 for a list of intrinsic time constants).

Fig.4.9 displays Bode plots in response to the second input. This input simulates the effect of changing weather conditions. The main differences between the response to the heating system and that to the external conditions are:

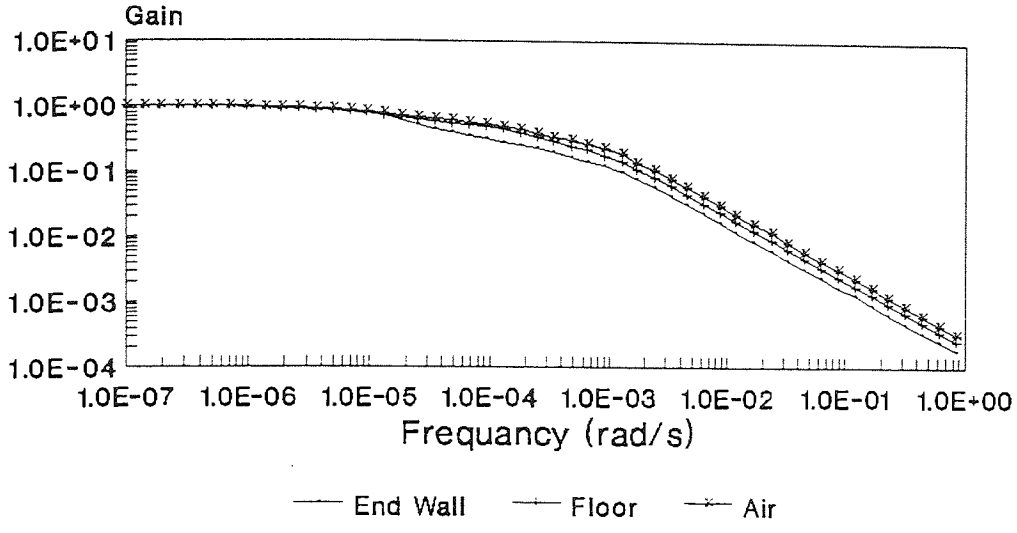
1) the gains are identical with respect to input 2 whereas the air temperature has a much lower gain at higher frequencies with respect to the heating, due to the indirect heating effect of secondary convection from the floor.

2) the gains with respect to the external conditions roll-off at a the same rate as those produced at the surfaces by the heating system ≈ 3 dB/octave, but slower than the internal air temperature with respect to the heating ≈ 5 dB/octave.

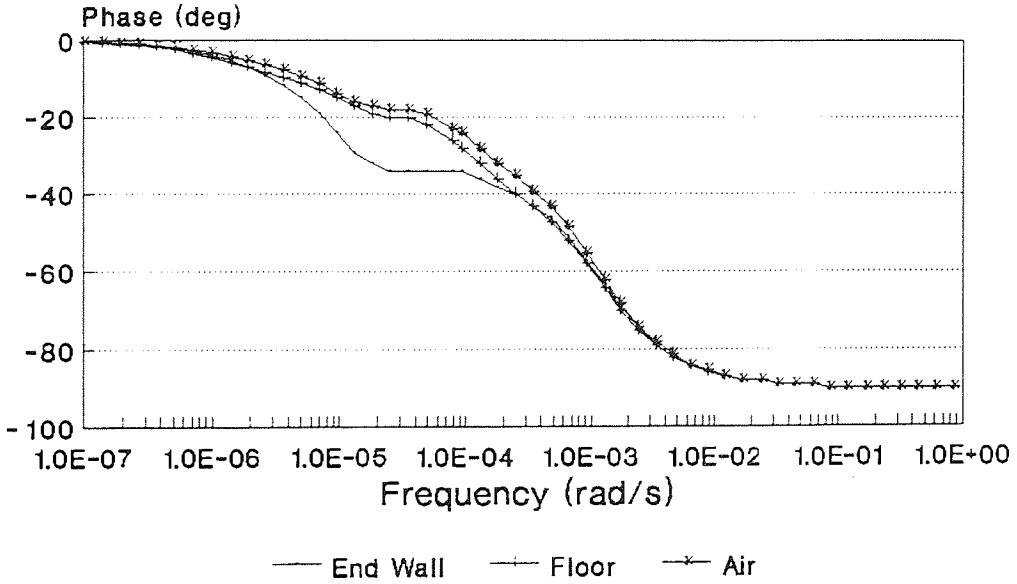
3) for the heating system input the phase of the temperature fluctuations for the surfaces change from being in-phase at low frequencies to lagging by 90° at high frequencies, whereas the air temperatures are 180° out of phase at high frequencies. This contrasts with the phase relations with the external temperature variations where none of the temperatures ever go beyond 90° out of phase with the driving force. This 180° phase shift is a possible source of instability if controllers based on air temperatures are employed.

These curves, along with the Nyquist plot, represent

Fig.4.9: Bode Plot
Room Temperature Response to External Temperature



Phase



fundamental design tools which provide indispensable information to the environmental engineer who is attempting to design a system that will conform to the performance specification laid down, without displaying any unwanted control-loop instabilities.

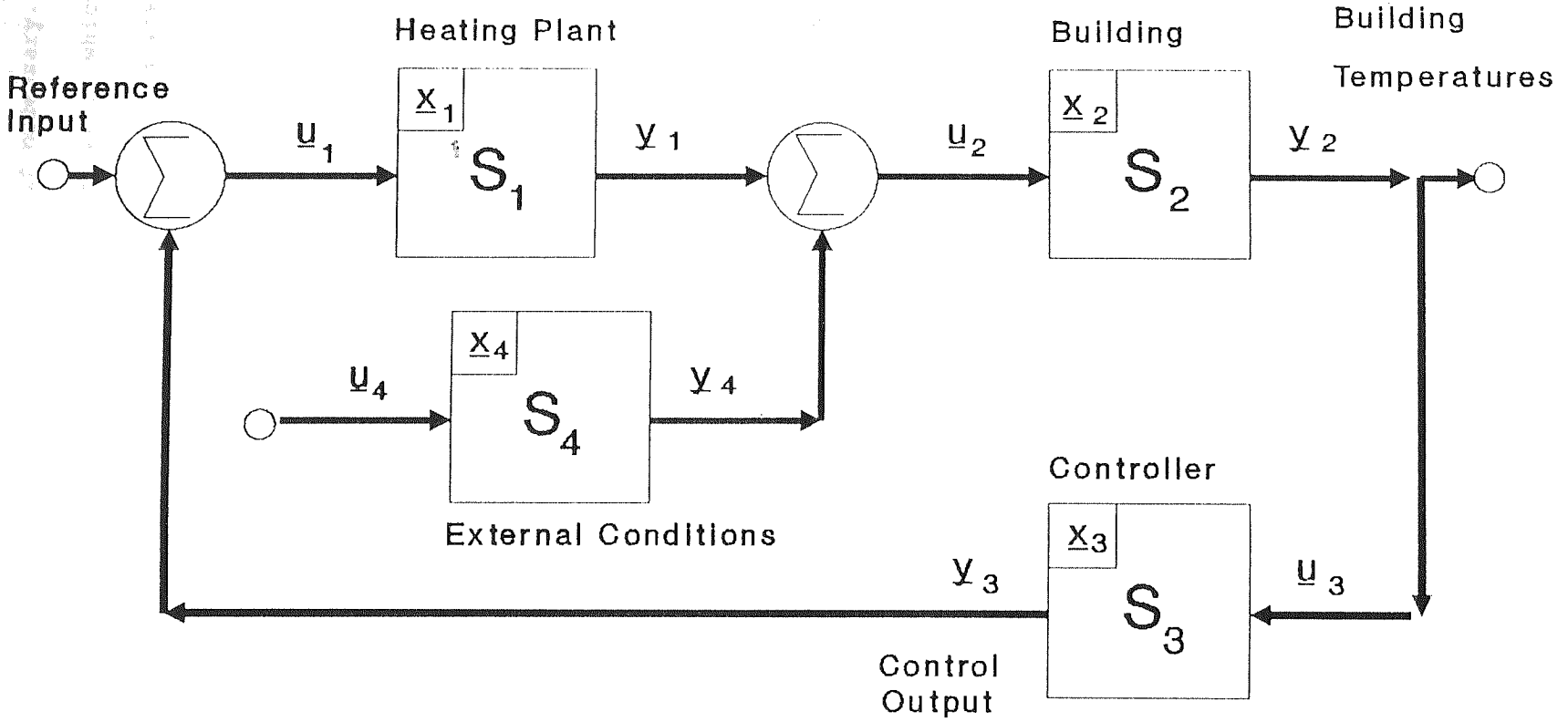
4.3 Pseudo Closed-Loop Response

Unfortunately none of the methods normally employed in designing a control system, eg the Linear Quadratic Regulator algorithm or the Root Locus method, are of much help in the case of radiant heating systems. This is because all currently available radiant heating systems are controlled using the so-called "bang-bang" principle whereby the system input is either fully-on or fully-off. None of them have the facility to reduce or increase the input. Fig.4.10 demonstrates the circuit arrangement employed universally in radiant heating systems.

The effect of introducing a non-linear device, such as a relay, into the control circuit is two-fold: from a mathematical point of view it makes further analysis very difficult, whilst from a practical point of view it immediately precludes the possibility of optimal control in the precise sense of the word. The best that can be achieved in the circumstances is to produce a limit cycle of specified period and amplitude for a minimum fuel cost.

In order to investigate this method of control it is first necessary to arrive at a method of simulating the behaviour of a relay with hysteresis in software. Listing 4.2 contains the CTRL-C prescription for simulating a relay, a procedure called *PCL.CTR* (Pseudo-Closed Loop). This procedure takes the space-state matrices, a time row vector, an upper and lower switching temperature and an initial state vector as input parameters, and returns the three matrices **Y**, **X**, and **U** which are the histories of the output, input, and control vectors respectively. The heart of the procedure is a call to the library procedure *simu*, which computes the system response to the

Fig.4.10: Control System for Radiant Heating

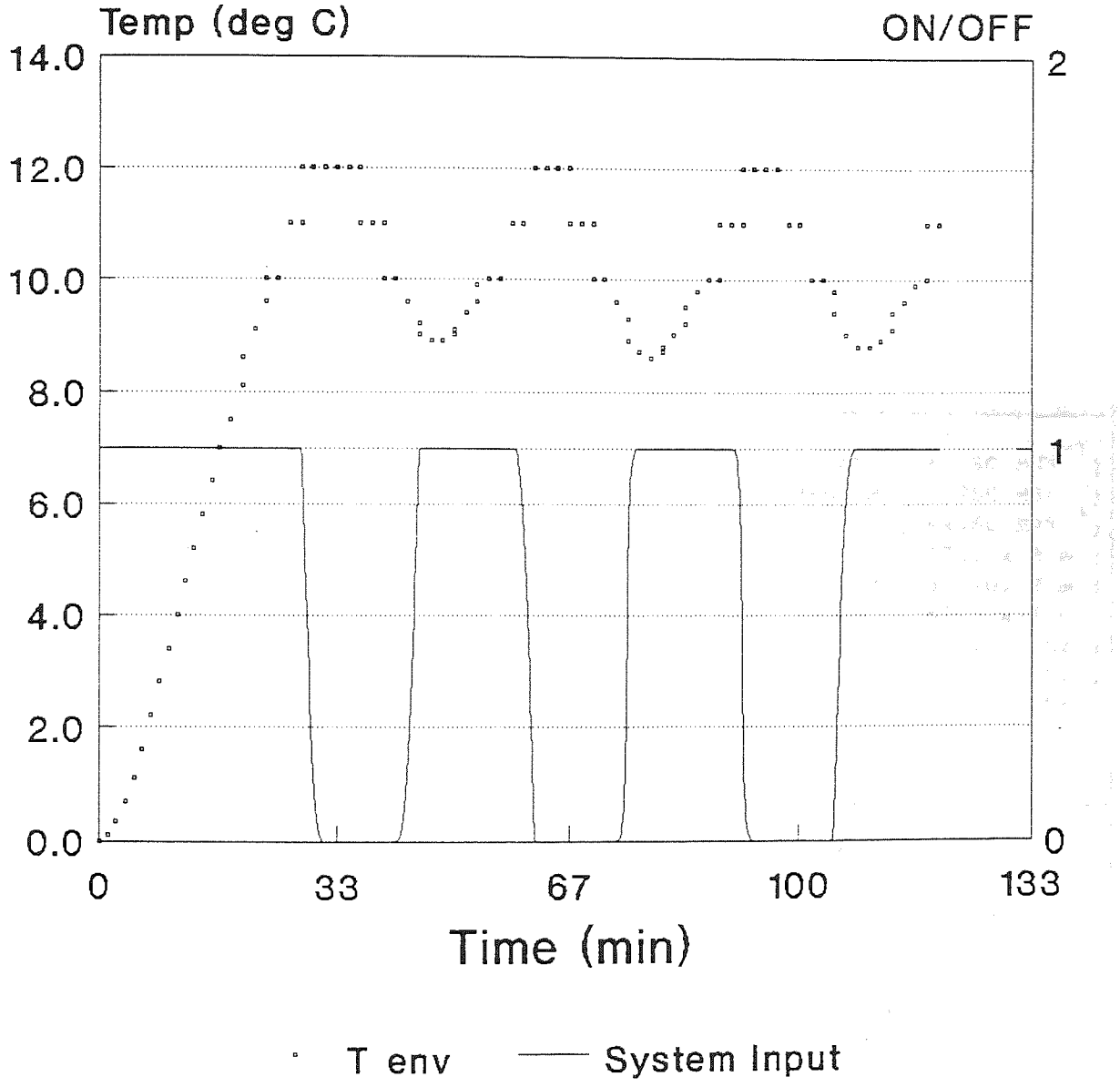


given input for one time step; *pcl* then checks the value of the output produced and uses it to switch the input if necessary.

There are two advantages to the way in which *pcl* has been implemented. Firstly, because the state vectors are recorded it is possible to call *pcl* using large time steps in the start-up phase when there will be no switching while the system comes up to the set temperature and, once switching has commenced, it is then possible to change the time-step to give a more detailed history of the switching phase. Secondly, because the switching is described by a logical function, it is possible to use any combination of state-vectors or output-vectors to decide when to switch; alternatively, if the system has multiple inputs, as would be the case if the heaters were wired up in such a way to switch on at different temperatures, then more complex algorithms can be employed.

The output from *pcl* displays some important features (Fig.4.11). The delay in response of the heating system due to the intrinsic constants of both the controller and the heaters is quite evident. It is noticeable how much this affects system performance at switch-on where there is a large overshoot of the lower temperature bound before the heating system manages to raise the temperature. The duty-cycle is heavily biased towards the heating side of the cycle under these conditions with the heaters being on about 75% of the time.

Fig.4.11:Controller Output History from Cold Start for Medium Room with 4xIRH_1



4.4 Investigation of the System Parameters

The two criteria by which any heating system must ultimately be judged are the degree of control over the environment that it achieves, and the cost, in terms of the fuel used, to provide this control. All heating control systems embody a compromise between control and economy. The solutions produced are, however, compromises that are designed to maintain the best possible balance between these two conflicting requirements.

Table 4.3: Model Parameters

Parameter	Quantity varied	Range of Values
Type of heater	Time constant	Short (≈ 6 min)
		MEDIUM (≈ 10 min)
		Long (≈ 20 min)
	Output per heater	Low (≈ 5 kW)
		Medium (≈ 10 kW)
		HIGH (≈ 15 kW)
Building	Thermal weight * (Cf. Table 4.4)	Light (≈ 250 MJK^{-1})
		MEDIUM (≈ 750 MJK^{-1})
		Heavy (≈4250 MJK^{-1})
	Physical dimensions	Small (15x10x 8 m)
		MEDIUM (30x10x 8 m)
		Large (60x30x16 m)
	Air change rate	Low (0.5 /hr)
		MEDIUM (1.0 /hr)
		High (2.0 /hr)
	Mode of operation	Intermittent
		CONTINUOUS
Control law	Strategy[2] *	BANG-BANG
		Multi-stage
		Full-state feedback
	Feedback variable *	Air temperature
		ENVIRONMENTAL TEMP
		Mean Radiant Temp
Controller properties	Time constant[3] *	Fast (100 s)
		MEDIUM (250 s)
		Slow (500 s)
External conditions	External air temperature	

This section will adumbrate the effect of several of the principal system parameters upon these performance specifications. Table 4.3 shows the full list of possible variants, with those which are to be examined being marked with an asterisk. The method of exposition will be to choose a default parameter-set which is representative of a typical system and then to alter the parameter of interest whilst maintaining all the other parameters at their default values - these are capitalized in the table.

4.4.1 Method

A systematic approach to generating results is vital if any significant comparison is to be attempted. The procedure adopted to produce the results that follow was:

1) define all necessary values of the parameters;

2) establish the space-state description of the specific case being investigated;

3) call *pc1* with a time-period long enough to allow the system to commence switching, and for the room dynamics to settle down with T_{ext} = 0 °C;

4) record the state-vector at the end of the initialization phase and then use it as the starting point for the second call to *pc1* using much shorter time-intervals;

5) compute the performance-index summations and plot the time-histories of the variables of interest.

Several points need to be stressed. Firstly, the initial state of the thermal capacitances has a very important effect on the subsequent behaviour of the system. Secondly, the length of time

Table 4.4: Thermal Properties of Construction Materials (Clarke, 1985)

Walls: Nodes 1,2,5,6: Area = 640 m^2: H_a = 3.0 W m^{-2} K^{-1}.

Type	Material	Conductivity	Density	Specific heat	Emissivity	Conductivity	Thickness	U-value	Thermal weight
		$Wm^{-1}K^{-1}$	kgm^{-3}	$Jkg^{-1}K^{-1}$		$Wm^{-2}K^{-1}$	m	$Wm^{-2}K^{-1}$	$MJm^{-3}K^{-1}$
Light	Insulating brick	0.27	700	840	0.90	1.71	0.15	1.12	0.588
Medium	Breeze block	0.44	1500	650	0.90	1.71	0.25	1.11	0.975
Heavy	Paviour	0.96	2000	840	0.93	1.71	0.55	1.10	1.68

Ceiling: Node 4: Area = 300 m^2: H_a = 4.3 W m^{-2} K^{-1}.

Material	Conductivity	Density	Specific heat	Emissivity	Conductivity	Thickness	U-value	Thermal weight
	$Wm^{-1}K^{-1}$	kgm^{-3}	$Jkg^{-1}K^{-1}$		$Wm^{-2}K^{-1}$	m	$Wm^{-2}K^{-1}$	$MJm^{-3}K^{-1}$
Roof insulating board	0.19	960	950	0.90	2.55	0.075	1.59	0.912

Floor: Node 3: Area = 300 m^2: H_a = 4.3 W m^{-2} K^{-1}.

Type	Material	Conductivity	Density	Specific heat	Emissivity	Conductivity	Thickness	U-value	Thermal weight
		$Wm^{-1}K^{-1}$	kgm^{-3}	$Jkg^{-1}K^{-1}$		$Wm^{-2}K^{-1}$	m	$Wm^{-2}K^{-1}$	$MJm^{-3}K^{-1}$
Light	Aerated concrete	0.16	500	840	0.90	0.28	0.60	0.251	0.420
Medium	Light mix concrete	0.38	1200	653	0.90	0.28	1.35	0.264	0.784
Heavy	Heavy mix concrete	1.40	2100	653	0.90	0.28	5.00	0.263	1.37

required for each system to stabilize will depend on the intrinsic time constants. Thirdly, it is safe to say that once the system has been in the switching mode for several hours it will have stabilized to an acceptable degree. Finally, this method of using two calls to *pcl* results in a large saving in computation time.

4.4.2 The Quadratic Performance Function

One further concept needs to be introduced before the performances of the various simulations are compared, and that is the idea of a performance index or function. This is defined to be:

$$J = 1/2 \; \Sigma \; [\underline{y}_t^{\,T} \; \underline{u}_t^{\,T}] \; \begin{vmatrix} \mathbf{Y} & \mathbf{N} \\ \mathbf{N}' & \mathbf{R} \end{vmatrix} \; \begin{vmatrix} \underline{y}_t \\ \underline{u}_t \end{vmatrix} \qquad (4.12)$$

where: $\underline{y}_t$ = row-vector output time-history,

$\underline{u}_t$ = row-vector control time-history,

$\mathbf{Y}$ = symmetric weighting matrix of output errors,

$\mathbf{R}$ = symmetric weighting matrix of control effort,

$\mathbf{N}$ = matrix of weighting between states and controls.

This is known as the quadratic performance function (QPF) and it occupies a special place in control theory. The use of the QPF for a system using full-state proportional control allows an optimal solution to be designed by solving the algebraic Ricatti equation to find the state-feedback coefficients. Although the control systems currently used with practical radiant heating systems are non-linear on/off relay controls, it is still worth using the quadratic performance function for two reasons. Firstly, it allows a direct comparison to be made with the optimal solution, despite the fact that this can not, at present, be practically realized. Furthermore,

work carried out on modulating radiant burners (Koussouris, 1984) may mean that in the future some sort of proportional feedback control system may be available: this would allow a closer approach to optimal control.

4.4.3 Simulation Results

The results of the calculations of the QPF for each of the parametric variations are shown in Table 4.5. This is an integration over time of the deviation from the reference which also incorporates the control effort. In this case it has been equally weighted between the temperature error and the normalised system input.

Table 4.5: Performance Function Values (°C)

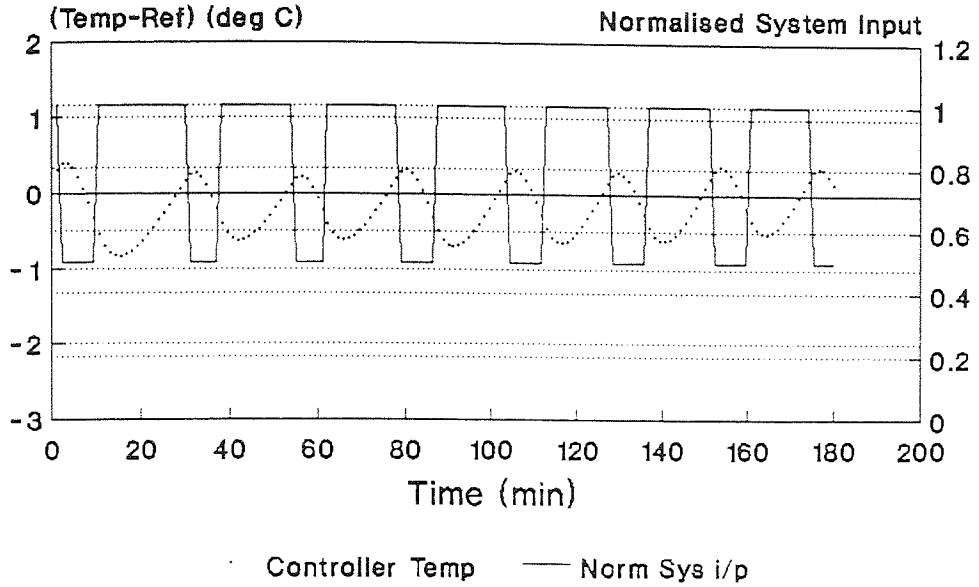
Parameter / QPF	Thermal Weight	Control Law	Feedback Variable	τ_{ctr}
J_1	2.09 (Light)	2.39 (Single)	2.31 (Air)	2.80 (Short)
J_2	2.39 (Medium)	1.24 (Double)	2.39 (T_e)	2.39 (Medium)
J_3	2.96 (Heavy)	1.02 (Quad)	2.24 (MRT)	2.49 (Long)

Note: The values of the parameters are given in parentheses.

On examining Table 4.4 it is apparent that the factor which has the most significant effect on J is the strategy of multi-stage switching (Fig.4.12). Introducing the dual switching capability reduces J by 48%, whilst the quadruple switching reduces J by 57%. The thermal weight, which is next in order of precedence, affects J by ≈ ±20 %, with lighter weight proving easier to control (Fig.4.13). The feedback variable makes only a small (≈± 7 %) difference: in particular using the air temperature is not as effective as using the MRT. This is because the surface temperature of the walls changes faster than the air temperature, as was noted earlier, and thus changes in MRT provide

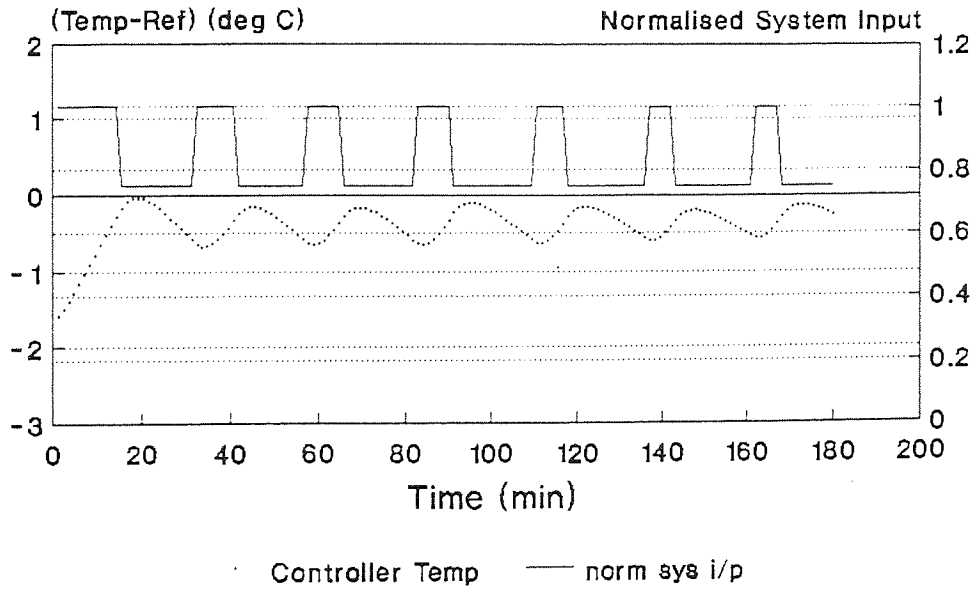
Fig.4.12: Switching Strategy

Dual Switching: QPF = 1.24 K



T_ext = 0 deg C

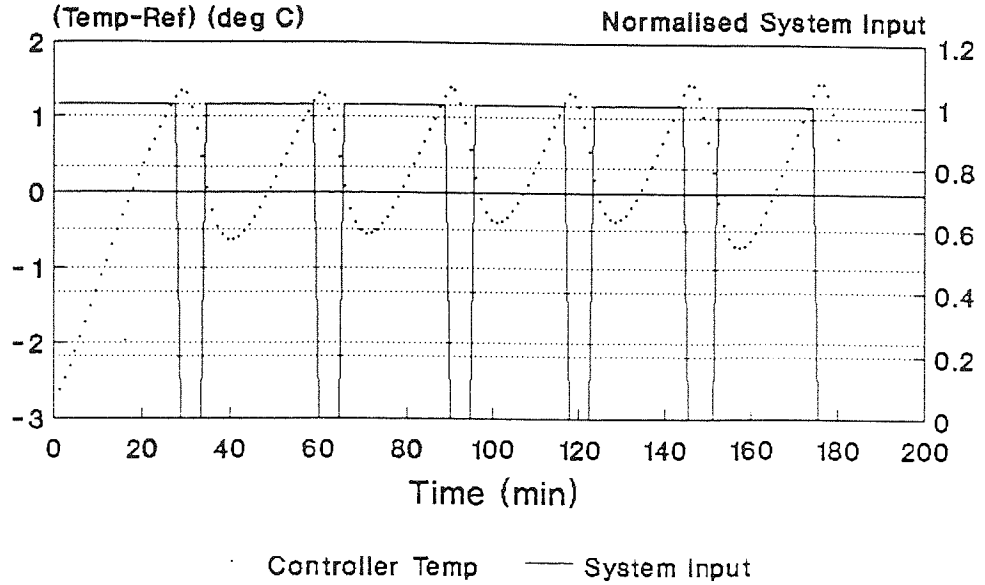
Quad Switching: QPF = 1.02 K



T_ext = 0 deg C

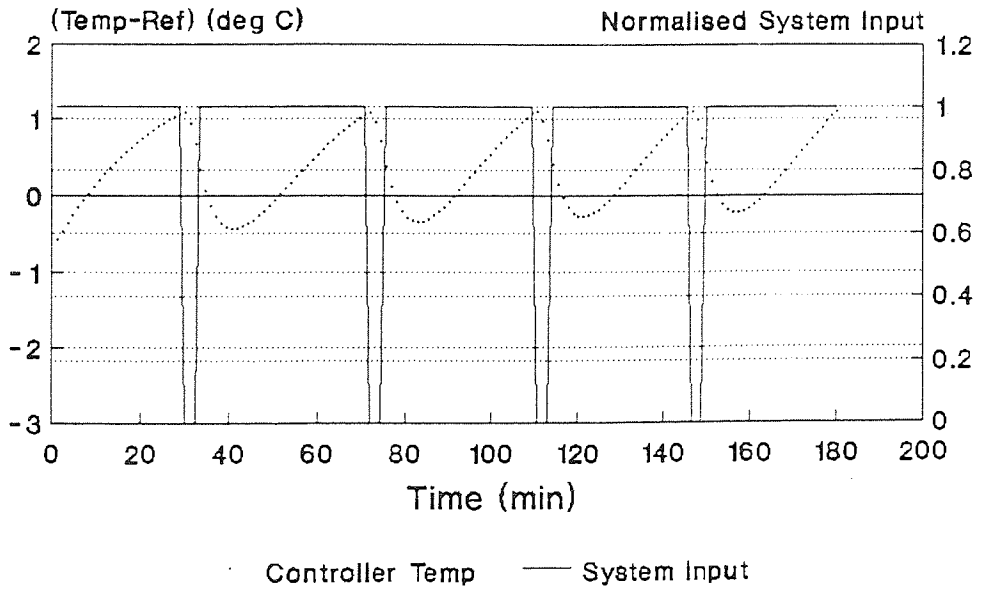
Fig.4.13: Feedback Response

Light thermal weight: QPF = 2.09



T_ext = 0 deg C

Heavy Thermal Weight: QPF = 2.96 K



T_ext = 0 deg C

a faster feedback route (Fig.4.14). Lastly, the figures for the time constant of the controller appear to contain an anomaly because the value of J for the short and long time constants are higher than for the medium constant, whereas a monotonic increase or decrease would be expected (Fig.4.15). This can only be explained by remembering that to achieve a low value of J requires a balance of both precise control and small control effort. It appears that in the case of the short time constant there is good control but poor economy, whilst the long constant provides good economy but poor control, whereas the medium time constant provides the best overall balance. It would be worthwhile to perform a parametric study to find out upon what the best controller time constant for a given system depended.

4.4.4 Conclusions

1) The most important factor affecting system performance is the choice of control law.

2) The choice of feedback variable has only a marginal effect.

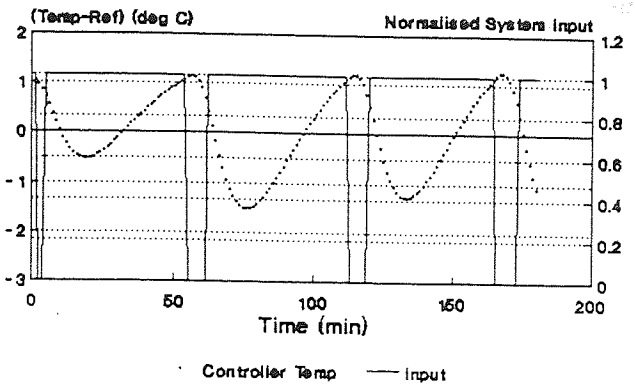
3) The effect of the thermal weight of the building is also of considerable importance in designing the heating and control systems, and should be taken into account when the performance specifications are being written.

4) From a comparison of the graphs and the QPF values it can be seen that the QPF proves to be a reliable indicator of well and badly designed systems.

5) the application of network and control theory to the problem of controlling radiant heating systems has proved to be a fruitful method of modelling the overall processes behaviour.

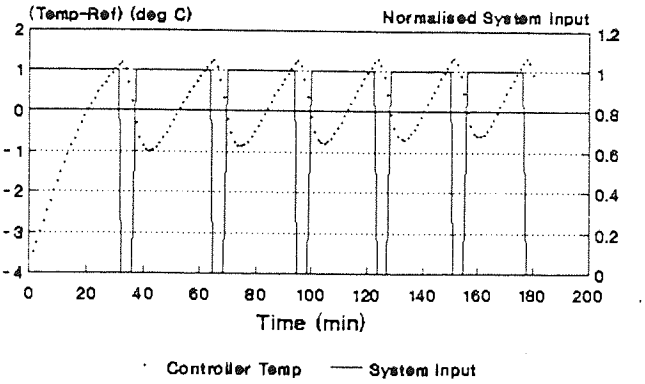
Fig.4.14: Feedback Response

Air Temperature: QPF = 2.31 K



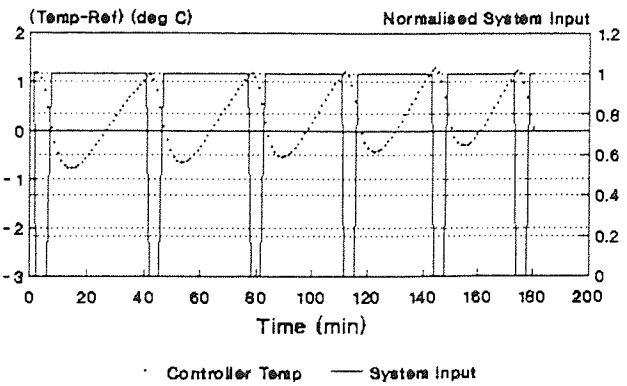
T_ext = 0 deg C

Mean Radiant Temperature: QPF = 2.24 K



T_ext = 0 deg C

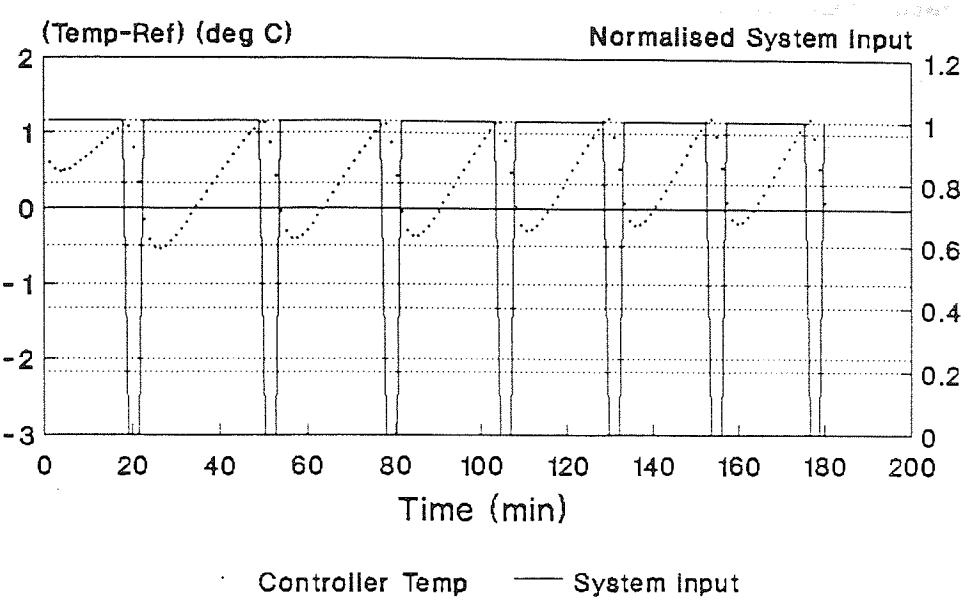
Environmental Temperature: QPF = 2.39 K



T_ext = 0 deg C

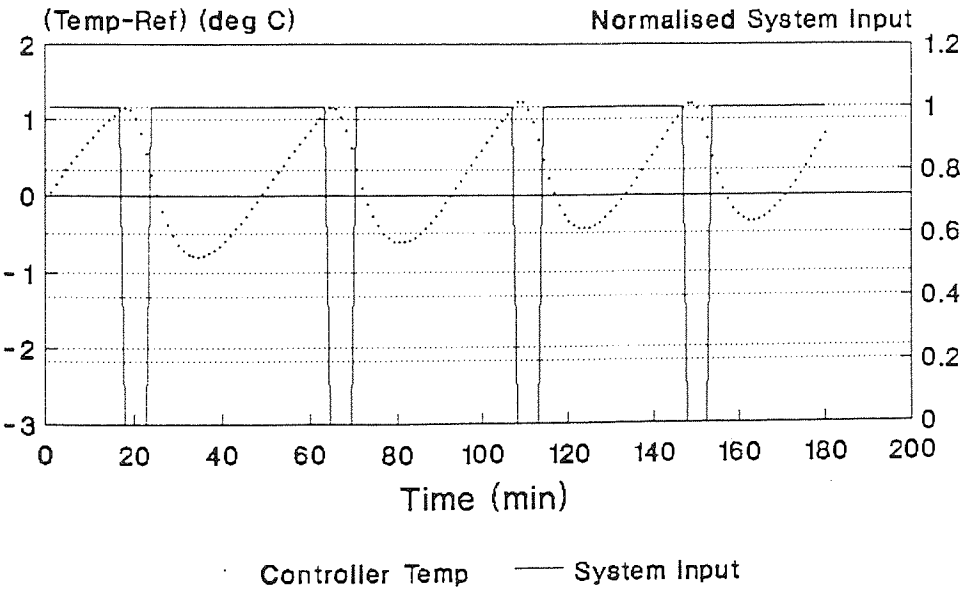
Fig.4.15: Feedback Response

Fast controller: QPF = 2.80 K



T_ext = 0 deg C

Slow Controller: QPF = 2.49 K



T_ext = 0 deg C

4.5 Summary

This section has dealt with the theoretical aspects of controlling a radiant heating system. The electrical analogy of heat flow within a building, developed in chapter 3, has been used in conjunction with the state-space description of the relevant plant and controller dynamics to simulate a complete system. A definition of a particular measure of control system performance, the quadratic performance function, has been provided. This has then been used to differentiate between the effects of the various model parameters. This has led, finally, to the important conclusion that it is the control law, or algorithm, which has the major effect on system performance.

Footnotes to Chapter 4:

1: In Section 6.1 it is shown that the time constants of radiant heaters for the heating phase and cooling phase are different by up to a factor of two. What then is the justification for using a first order model with a single time constant? There are two reasons. Firstly, modelling a sub-system with two time constants which are applicable at different times is difficult. This is because there is an implicit discontinuity where the component is "switched" from one representation to the other. This non-linearity is impossible to model with an essentially linear modelling package. The choices are to either choose a completely different method of modelling the system, or to try to find an *ad hoc* method of getting around the problem, as has been done with the controller, or alternatively to accept a compromise in the description of the system. The path chosen here is to accept the compromise.

Secondly, having accepted a compromise, it should be noted that the relative order of magnitude of the time constants is:

Component	time constant (s)	
floor	370×10^3	(≈ days)
ceiling	24×10^3	(≈ hours)
air	3.6×10^3	(1 hour)
heating	0.5×10^3	(≈ 10 min)
controller	0.3×10^3	(≈ 5 min)

Bearing these values in mind it is not unreasonable to take a single mean figure for a heater to reflect a heater's dynamic behaviour.

2: The three forms of control cited in Table 4.4 have the following meanings:

Bang-bang: The system inputs are either fully on or fully off

Multi-Stage: The system can be switched on in a number of stages which will depend upon how it is wired up, the two examples used in this exposition are DUAL, ie two stages, and QUAD, ie four stages. These stages are activated at different temperatures as an attempt to mimic proportional control.

Full-State Proportional Feedback: This method of controls assumes that full information of the system state, ie the vector $\underline{x}$, is available to the controller. This can then be used in conjunction with a performance index, and a suitable design method to generate an optimal control law for the system. In this discussion this technique is largely of academic interest as it is inconceivable that all of this information would be available.

3: The controller time constants are taken from Fisk (1981) as being a representative sample of the type of controllers employed in radiant heating control systems.

5 Radiometry

When seeking to measure the output of radiant heaters there are two complementary approaches that may be adopted. The first involves the measurement of the MRT directly with a suitable device, most usually a black globe thermometer; the second is the measurement of the irradiance. This latter technique is most often performed with a thermopile. The results of both these methods are easily interconvertible by the formula quoted in Eqn. 2.1. The advantages of measuring the irradiance are that the time constant of a thermopile is orders of magnitude smaller than that of a black globe, and also that the values measured may be compared directly to those calculated by the Monte Carlo model. The disadvantage of using a thermopile, particularly with tubular radiant heaters, is that commercially available radiometers do not have a sufficiently large viewing angle to "see" all of the heater at the measuring distances typically used. This necessitates the construction and calibration of special "wide-angle" radiometers.

In this chapter the construction, calibration, and use of two wide-angle radiometers will be outlined. The use of the Land Radiometer for taking measurements of plaques and Quartz Linear Lamps (QLL) heaters will also be covered. The account of the Maund Radiometer is given first, followed by details of the improvements which resulted in the automated version. Two examples of their use for measuring the irradiance output of heaters are then given.

5.1 The Maund Radiometer

The original purpose of the Maund Radiometer was to take accurate measurements of tube heaters that would allow an assessment of the design under test (Maund, 1987). It was subsequently used to test the predictions made by the Monte Carlo model. In order to achieve both of these objectives it was necessary to ensure that the radiometers were calibrated to a sufficiently high degree of accuracy.

5.1.1 Construction and Calibration

The construction of the radiometer is shown in Fig.5.1. It is a very simple device consisting of a blackened type-K thermocouple and a hand-operated polished aluminium shutter. The output of the thermocouple is amplified by the integral chopper-amplifier in a Comark electronic thermometer and then fed to the 0-1V input of a chart recorder. The trace on the chart record is used to find the gradient of the thermocouple output at the instant when the shutter is removed. This gradient is proportional to the incident irradiance. The calibration is concerned with establishing the value of the constant of proportionality.

The calibration of four Maund Radiometers was performed using a plaque heater as a convenient source of radiation. Six positions were marked on the floor, calculated in such a way as to provide irradiances in the range of 50 to 400 Wm^{-2}, see Fig.5.2. The Land Radiometer was used as the secondary standard of comparison against which the radiometers were compared. The method adopted of gathering the data was as follows:

1) Turn on the heater and allow to heat up for 15 minutes.

Fig.5.1: Original Aston Radiometer

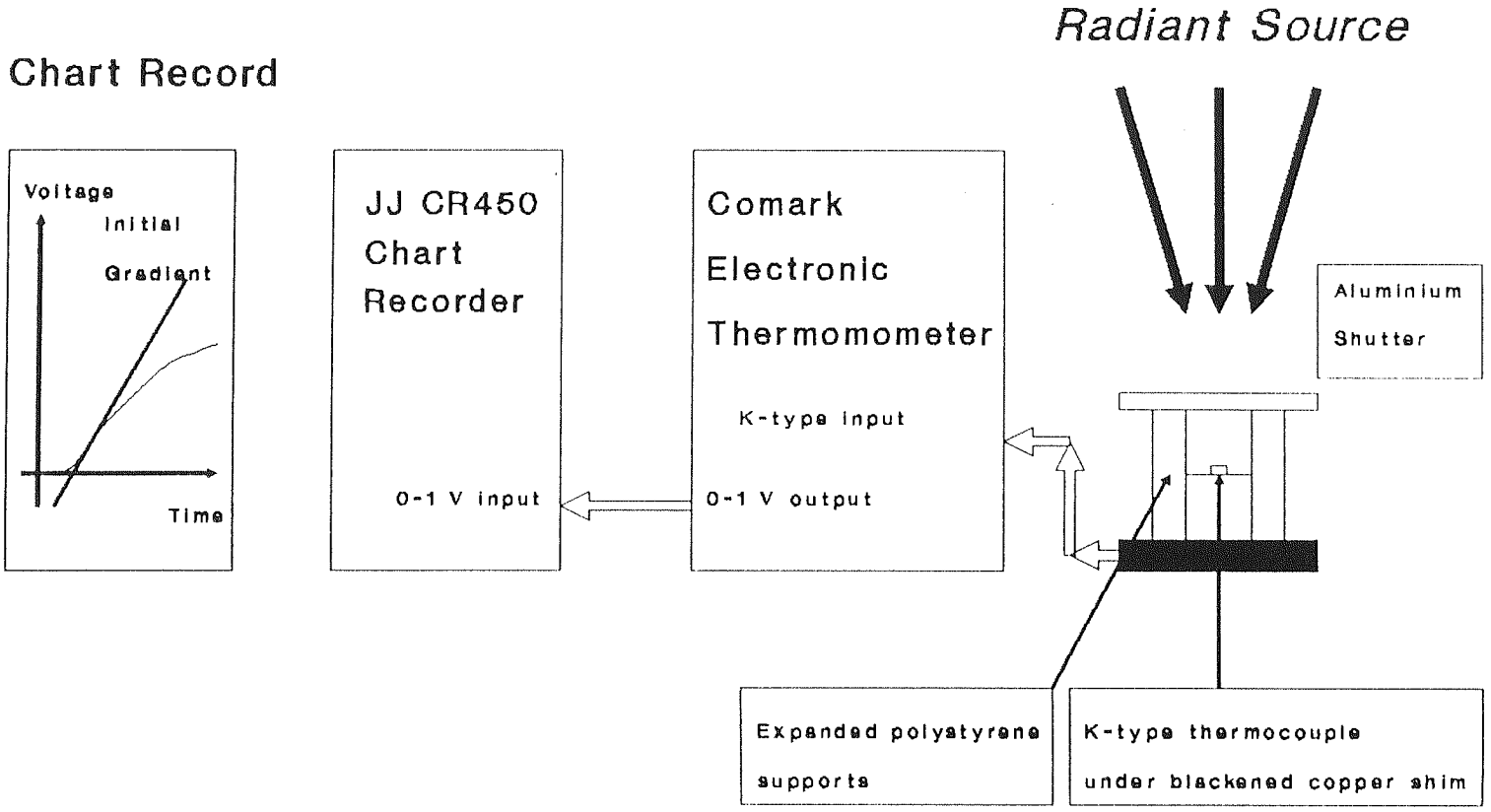
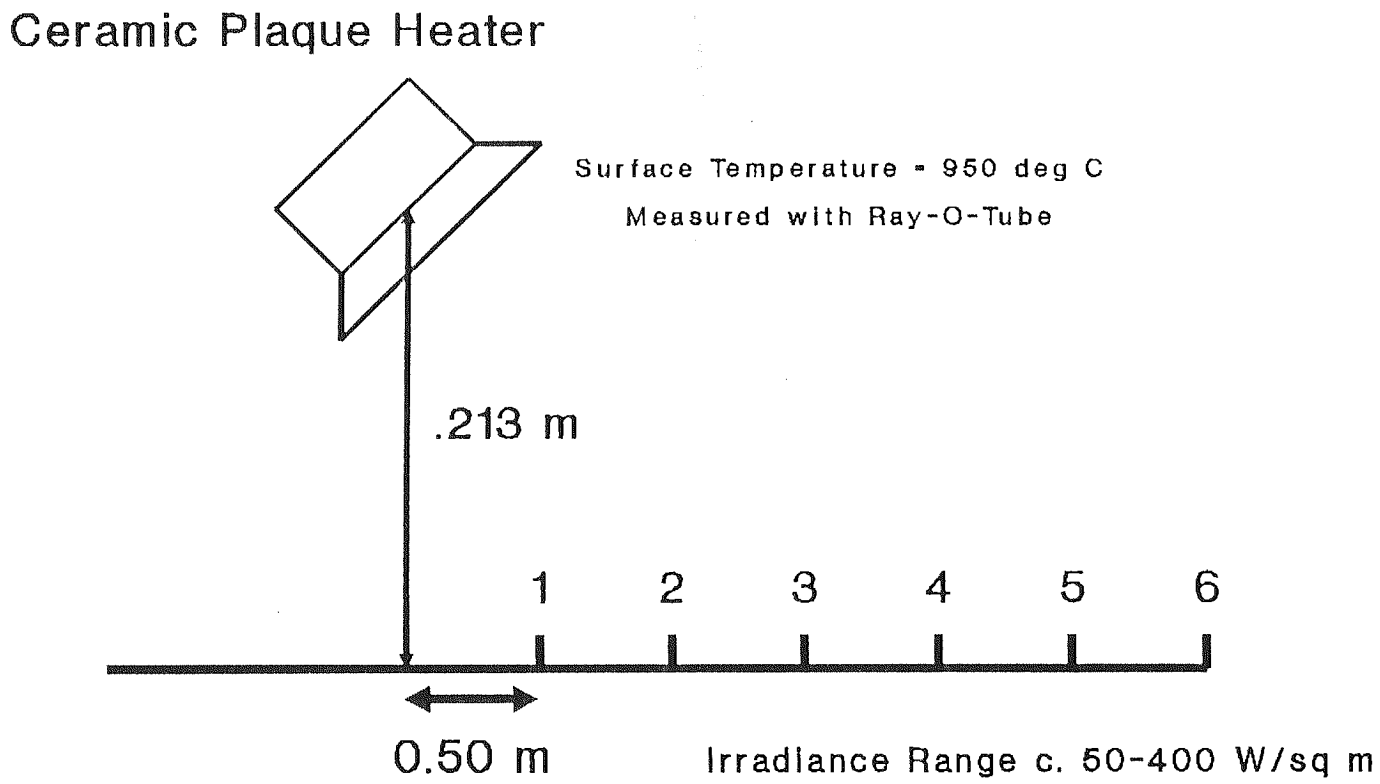


Fig.5.2: Calibration Arrangement



2) Place the four numbered radiometers on spots 1 to 4.

3) Place the Land next to the first radiometer and take a reading of the voltage produced by the Land.

4) Start the chart recorder and then remove the aluminium cover from above the thermocouple.

5) Replace the aluminium cover and stop the recorder.

6) Take another Land reading.

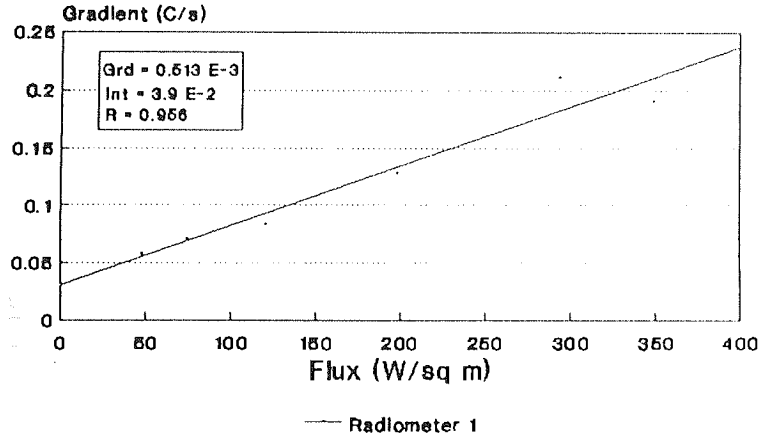
7) Move to the next radiometer and repeat process.

8) When all four radiometers have been read move them all to the next position and repeat procedure until all the radiometers have been measured in all positions 3 times in order to reduce the standard deviations.

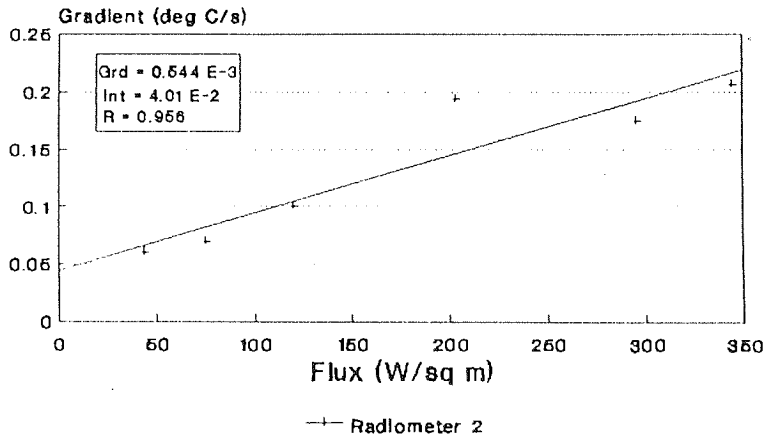
This procedure resulted in 6x4x3 = 72 chart records from the Aston radiometers and 144 Land readings. This data had then to be processed. Firstly the chart records were digitised by hand and then a Fortran program *CALIBRATE.FOR* was used to find the best-fit gradients: this was achieved using a least squares fit through the origin. Then a linear regression was performed on these gradients against the averaged Land readings. These Land readings had previously been corrected for ambient and source temperatures according to the method prescribed by the Land data sheet. Fig.5.3 shows the linear regression that resulted for all four radiometers, whilst the results are summarised in Table 5.1.

Fig.5.3: Calibration Graphs

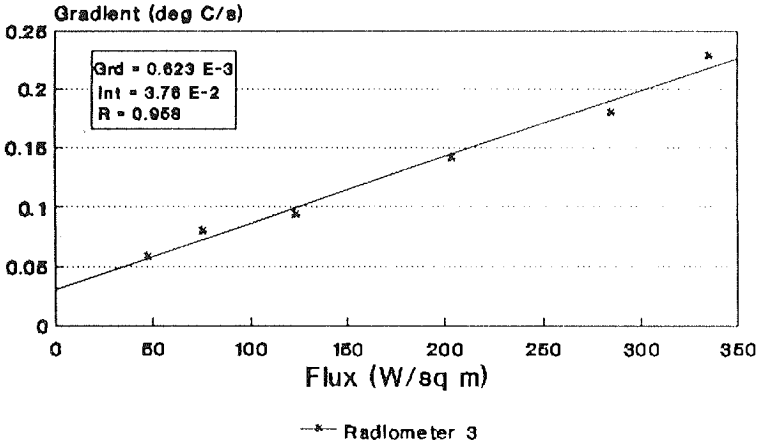
Radiometer 1



Radiometer 2



Radiometer 3



Radiometer 4

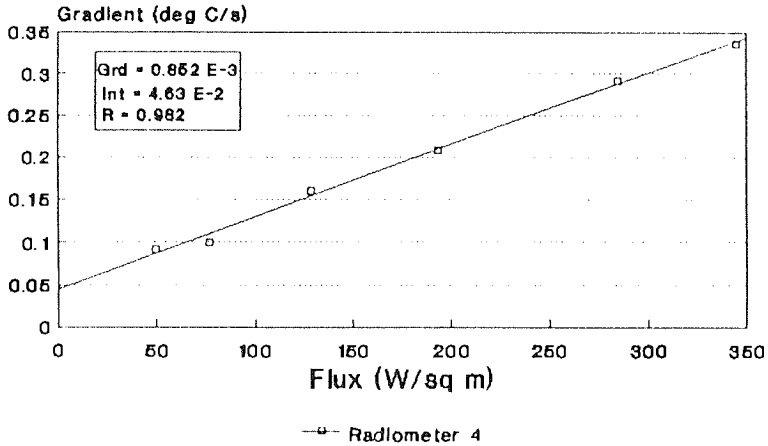


Table 5.1: Summary of Calibration of Maund Radiometers

Radm. Number	Gradient $°Cs^{-1}/(Wm^{-2})$	Error on Grd $°Cs^{-1}/(Wm^{-2})$	Cross Moment Correlation	Error (%)
1	0.51	0.08	0.956	±5
2	0.54	0.08	0.956	±5
3	0.62	0.09	0.958	±5
4	0.85	0.05	0.982	±2

It can be seen that the accuracy and consistency produced by this method of correlation are satisfactory.

5.1.2 Measuring the Irradiance Output of the Unit IRH_7

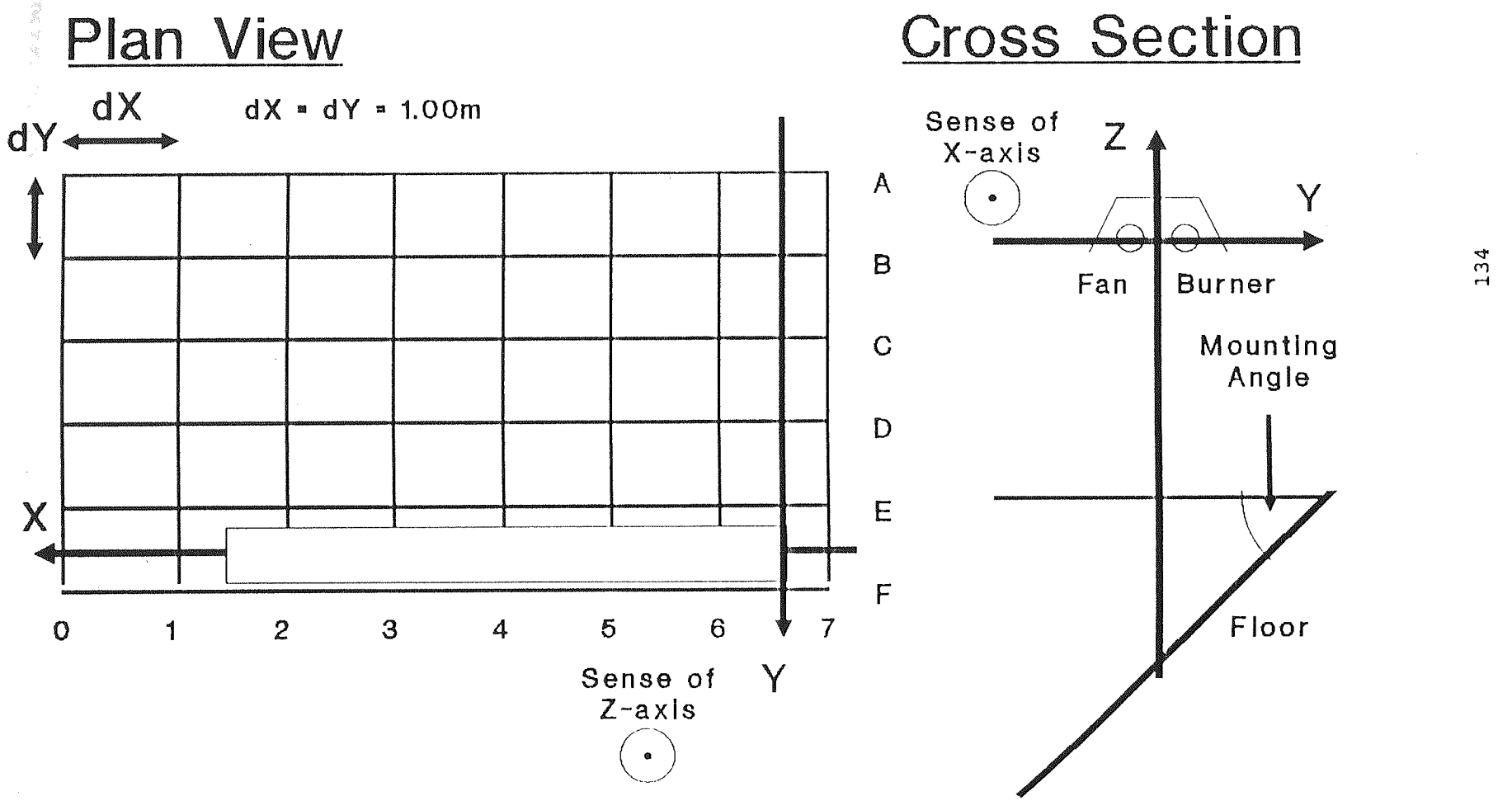
Before the measurements of irradiance were made various other relevant readings were taken so that the heater could be modelled accurately. The geometry of the heater is shown in Fig.5.4 along with an indication of the grid over which the readings were taken. The tube characteristics are summarised in Table 5.2.

Table 5.2 Heater Characteristics of IRH_7

Tube Length	: 4.55 m
Tube Outside Diameter	: 76.2 mm (3 ")
Tube Inside Diameter	: 68.3 mm (2 11/16 ")
Tube Thickness	: 7.9 mm (5/16 ")
Tube Material	: mild steel
Tube Separation	: 194 mm
Reflector Length	: 4.76 m
Reflector Material	: stainless steel
Mounting Height	: 1.975 m (to bottom of reflector)
Mounting Angle	: 41.5 ° (to the horizontal)

The minimum information required by the model is the heater geometry and the profile of temperatures along the tube. In order to calculate a heat and mass balance additional information is necessary, namely the temperature of the reflectors, the temperature and flow rate of the flue gas, the oxygen concentration in the flue, and if possible

Fig.5.4: Experimental Geometry for IRH_7



the flow rate and temperature of the air spilling out from underneath the reflector. The list of equipment used to measure these variables is given in Table 5.3.

Table 5.3 Equipment Inventory

Variable	Instrument	Type
Air speed	Hot-wire anemometer	Airflow TA3000
Surface Temperature (Contact)	K-type thermocouple plus electronic thermometer	Comark 2001
Surface Temperature (Optical)	Optical pyrometer	Ray-O-Tube 0076
Humidity	Sling psychrometer	Casella
Combustion Analysis	Oxygen monitor	Neotronics FEM
Irradiance	Maund Radiometer plus chart recorder	JJ CR450

The temperatures of the tube and the reflector, shown in Fig.5.5, were used both in the heat balance and in the model to calculate the predicted irradiance.

Table 5.4: Miscellaneous Experimental Results

Measured:

Input

Gas Consumption : 50.0s/ft^3, 51.0s/ft^3, 50.5s/ft^3 : mean 50.5s/ft^3
: 71.3ft^3/hr
: 21.6 kW

Atmospheric Absorption
Humidity:
Dry Bulb : 19.0 °C
Wet Bulb : 16.0 °C Implies Relative Humidity: 73%

Convection from Reflector

Temperature (°C)			Power (W)			
surface a	surface b	surface c	Qa	Qb	Qc	Total (W)
76.0	111.0	92.0	62.2	135.5	120.0	317.67
80.0	92.0	86.0	68.1	99.2	109.7	277.07
57.0	73.0	72.0	35.8	66.0	86.6	188.41
43.0	68.0	66.0	19.0	57.9	77.1	153.98
39.0	66.0	64.0	14.8	54.7	74.0	143.43
30.0	53.0	68.0	4.8	26.7	61.0	92.51
			204.7	440.0	528.4	1173.06

Convection from Tubes

The convective heat loss from the tube was estimated by measuring the flow rate of air through the gap between IRH_7 and a heater mounted next to it, the size of the gap, and the temperature of the air.

Flow rates (m/s): 0.33, 0.46, 0.42, 0.34, 0.35, 0.27, 0.24: mean 0.34
Temperature (°C): 40. Heat Content : 7.5 kW

Stack Losses and Flue Gas Analysis (45 minutes after switch-on)

Flue Temperature : 204 °C, 204°C : mean 204 °C
Oxygen in flue : 12.1%, 12.3% : mean 12.2% dry
Exhaust Flow Rate: 4.78, 5.00, 5.08, 5.19, 5.13, 5.07: mean 5.04 m/s
through area of 0.00421 m^2 (60.3x69.8 mm) : 21.2 dm^3/s
Excess Air : for 12.2% O_2 in flue : 132 %
Stack Losses : O_2 = 12.2%, Flue Temp = 204 °C : 25.0 % gross
: 5.184 kW
Latent Heat : H_2O = 0.839 g/s : 1.894 kW
Sensible Heat : by subtraction : 3.290 kW

Table 5.5: Mass Balance

Input	gmole/hr	kg/hr	g/s
Methane	84.	1.34	0.373
Oxygen	390.	12.5	3.47
Nitrogen	1460.	49.7	13.8
Output	gmole/hr	kg/hr	g/s
Carbon Dioxide	84.	3.70	1.04
Water Vapour	168.	3.02	0.84
Nitrogen	1460.	49.7	13.8
Oxygen	222.	7.09	1.97

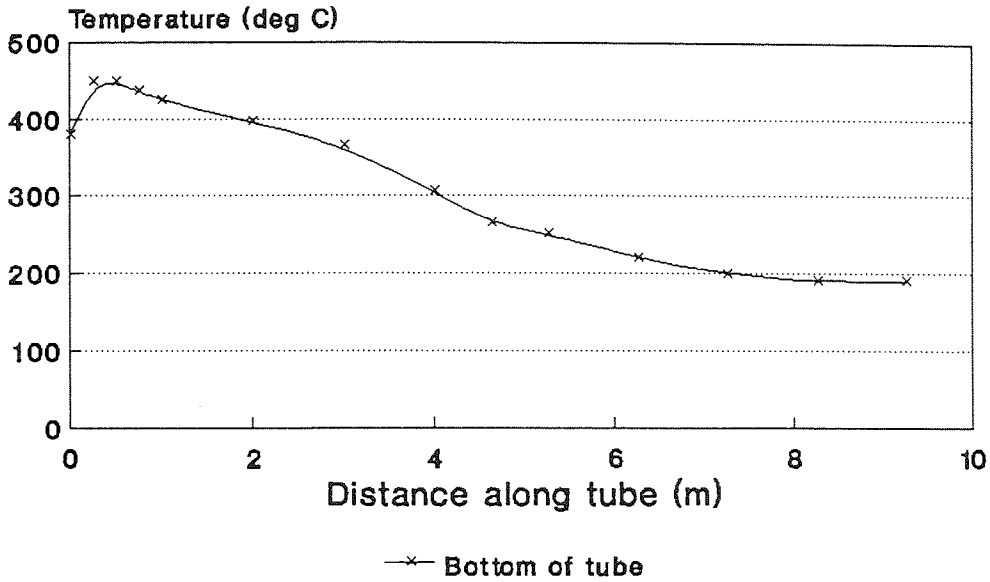
Table 5.6: Power Balance

Component	Power (W)	Power (%)
Input	21600	100.0
Radiation to Zone	7106	32.9
Atmospheric Absorption	604	2.8
Convection from Reflector	1173	5.4
Convection from Tube	7488	34.7
Flue Sensible	3290	15.2
Flue Latent	1894	8.8
Radiation from Reflector	107	0.5
Total	21662	100.3

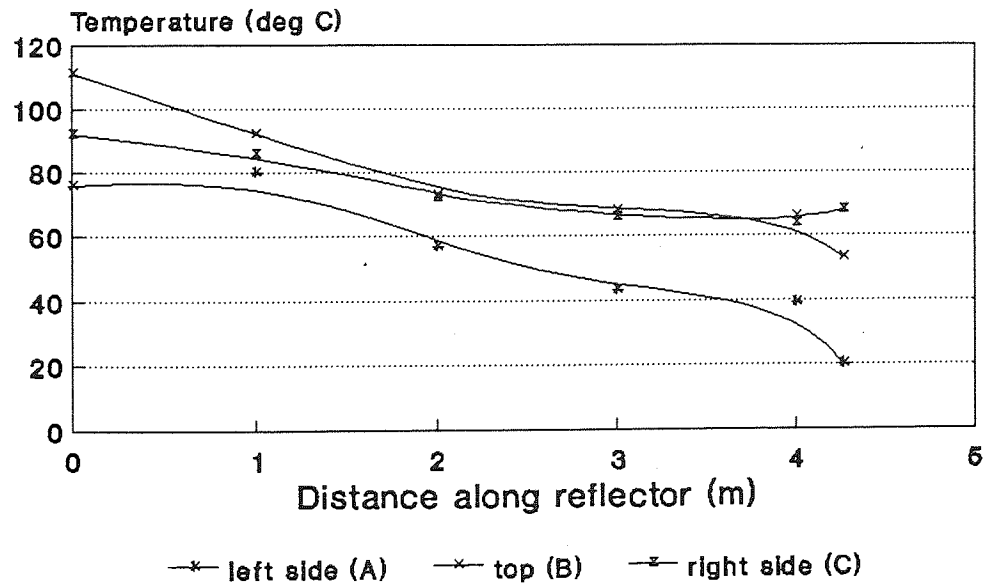
NOTE: The atmospheric absorption was calculated using a multi-window variable-transmittance model of the atmosphere (McIntyre, 1987).

Fig.5.5: IRH_7 Temperatures

Tube Temperatures



Reflector Temperatures



The proportion of radiant power directed onto surfaces, 33%, is typical for tube heaters, if a little on the low side. This may be explained by recalling that the heater was mounted at an angle of 41.5° to the horizontal which will have increased the convective loss considerably.

Having said which, the "fixed" losses, ie the flue loss, the convective and radiant loss from the reflectors, and the atmospheric absorption, will all be substantially the same whatever mounting arrangement is employed. They total 14.5 kW (67%), and this represents the figure which must be reduced if the efficiency of the heater is to be improved.

The technique used to measure the radiant power to surfaces is clearly of importance. The method was as follows:

1) Mark grid with spacings dX = 1.00 m; dY = 1.00 m on the floor under the heater so as to maximise the number of possible readings allowing for obstructions.

2) Place radiometers on first four positions.

3) Take reading from radiometer 1.

4) Move radiometer 1 to position 5.

5) Read next radiometer before moving it to subsequent position.

The use of this procedure ensured that each radiometer had about 2 minutes between readings to equilibriate at the ambient temperature. The gradients were all recorded on a chart recorder and digitized by hand in the same way as was used when the radiometers were calibrated. The irradiances were then calculated from the calibration coefficients produced by the calibration and shown in Fig.5.3.

A comparison of the measured figures and those calculated using the model are shown in Fig.5.6, 5.7, and 5.8. It should be noted that the modelled results were calculated for a mounting angle of 0° as opposed to the measured results taken from a heater mounted at 41.5°. It can immediately be observed that, whilst the modelled results are broadly similar to the experimental results, they are consistently lower across all of the zone other than the central region. The summed irradiances bear out this impression, with the sum of the modelled irradiances being 5.2 kW which is only 80% of the total measured value of 7.2 kW. Even allowing for the ±5% statistical uncertainty in the measurements there is still a sizable discrepancy. A possible explanation for this is arrived at by consideration of the information used in the model.

There are two sets of data which contain uncertainties. The first set is the emissivity of the tube and the reflector. The values of these constants were taken to be 0.9 and 0.1 respectively, because these are typically representative values for matt black paint and polished stainless steel. Clearly any errors in these values will feed through directly to the model. They are unlikely to be incorrect by more than ±5%.

The second set is the tube temperatures, especially the temperatures of the top of the tube. These were assumed to be 10°C higher than those taken along the bottom of the tube. This was because when the modelling was carried out it was known that the top of the tube would be hotter than the bottom but no figures were available. It is quite likely, as will be seen in the next section, that they had considerably higher values than those assumed. There is,

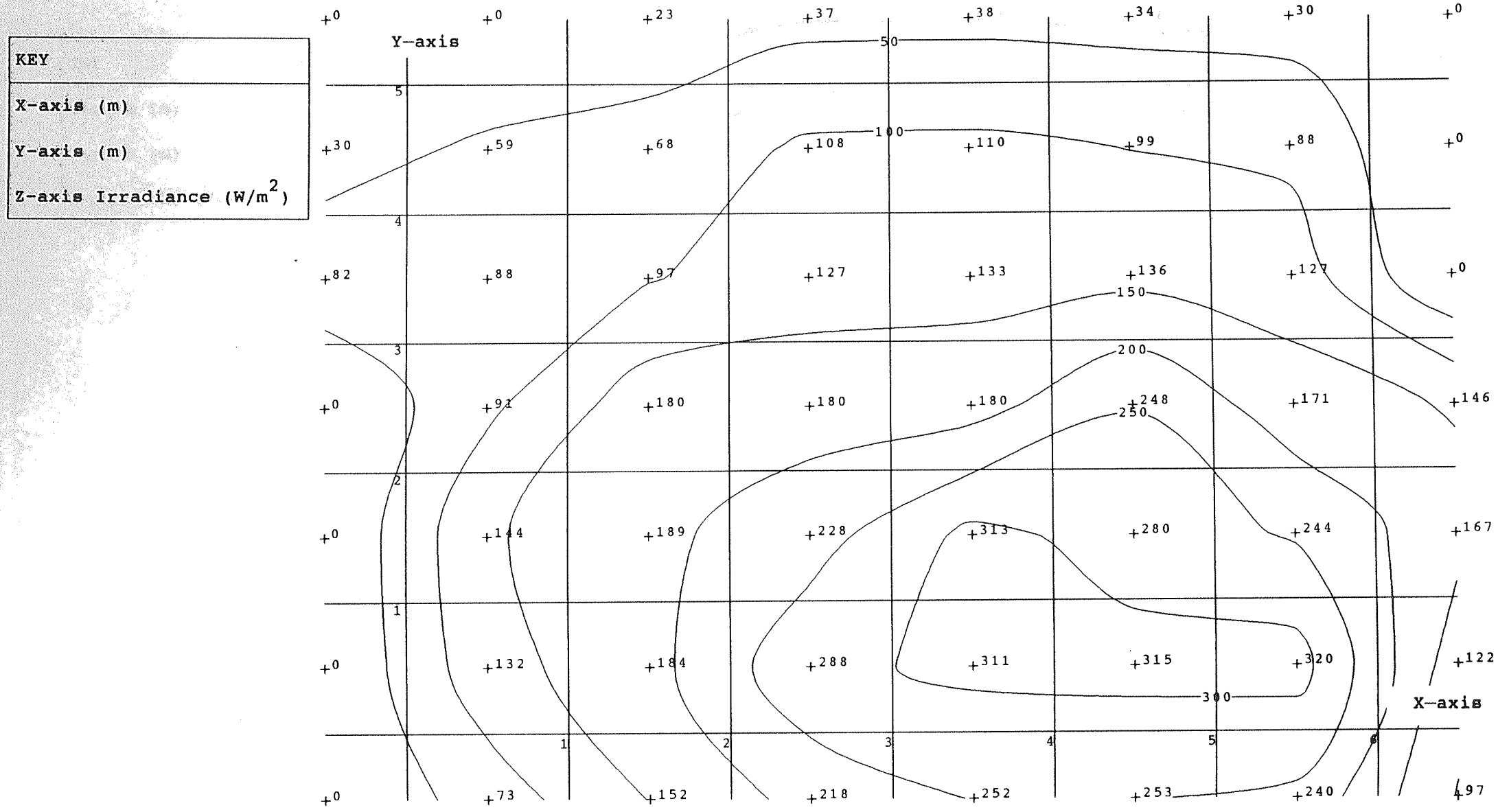


Fig.5.6:IRH_7 Irradiance Plot

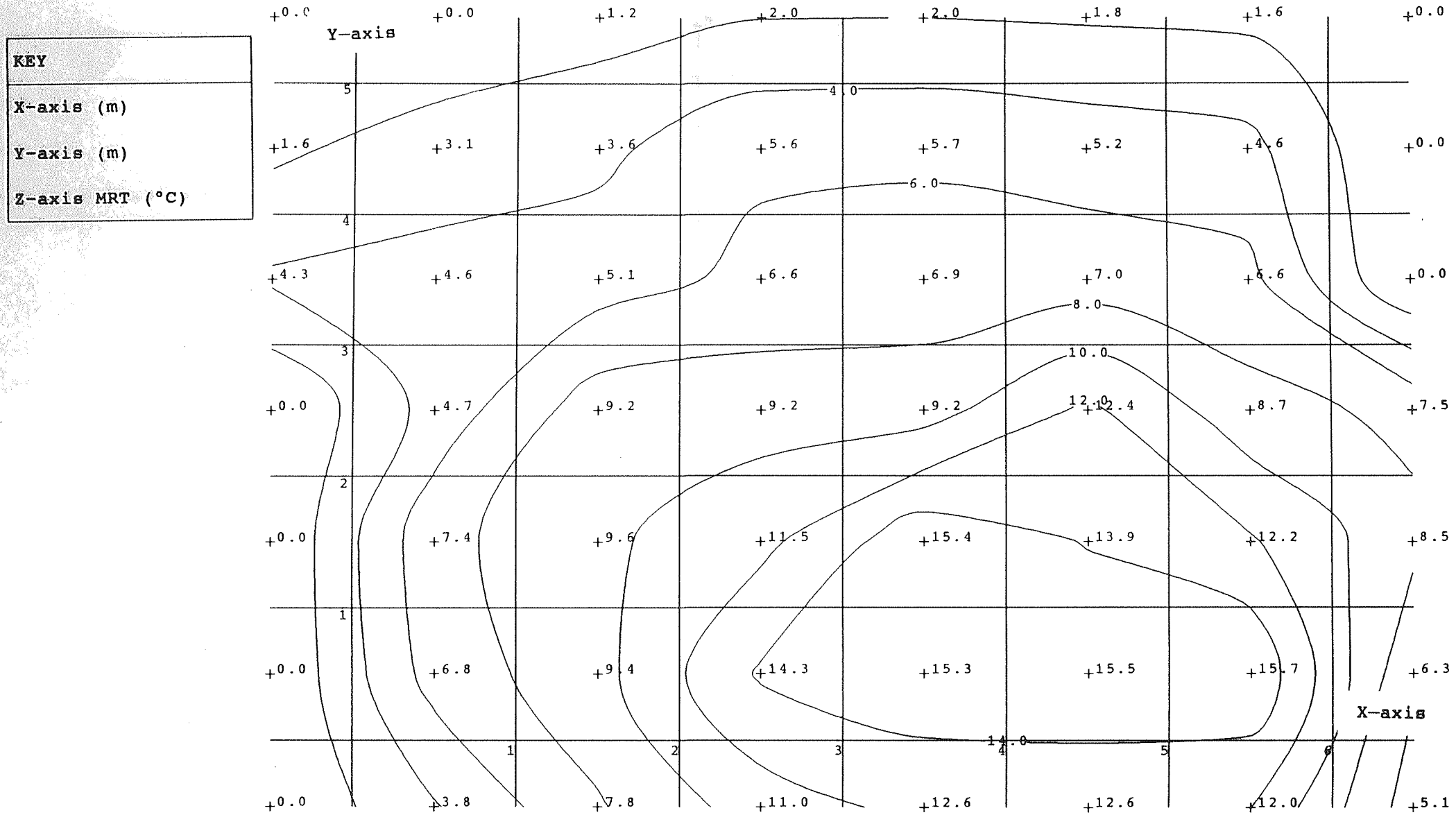


Fig.5.7:IRH_7 Black Bulb MRT Plot

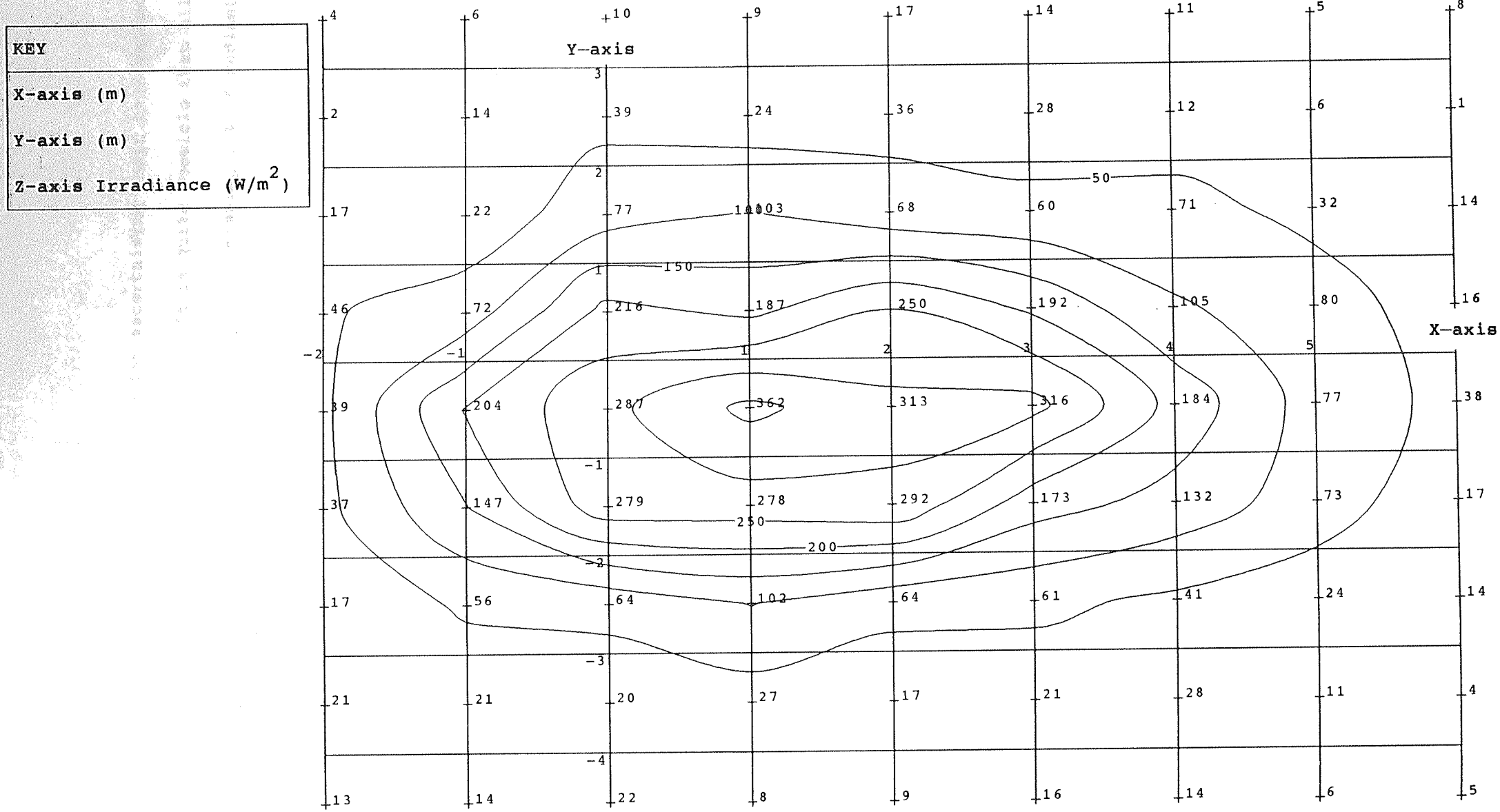


Fig.5.8:IRH_7_Modelled Irradiance Plot

unfortunately, no *a postiori* method for ascertaining their real values from the irradiance measurements taken. It is quite possible that all these sources of error add up to account for the missing 20% of radiant energy.

5.2 The Automated Radiometer

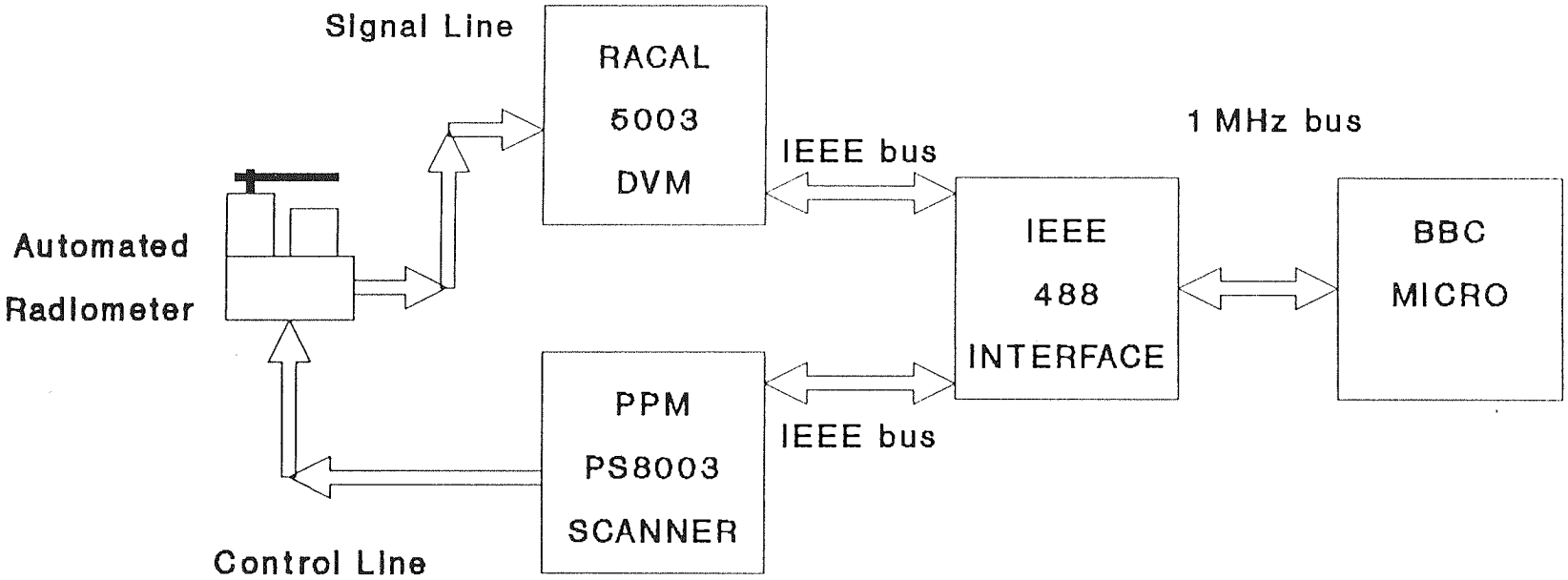
The drawbacks of the Maund Radiometer became apparent very quickly. It required two people to operate; it had a quick response but a slow recovery due to the length of time for which the thermocouple had to be exposed to record an adequate trace; it was susceptible to draughts; and, above all, it took an unconscionable effort to calculate the irradiance from the chart records. It was in order to address some of these problems that the Automated Radiometer was designed and built.

5.2.1 Construction

The essential difference between the Automated Radiometer and its predecessor was the addition of a rotary solenoid which allowed the shutter to be opened and closed remotely. This feature coupled with an integral thermocouple amplifier which improved the signal-to-noise ratio of the transmitted voltage allowed the radiometer to be coupled to a BBC microcomputer in such a way as to automate the whole process of measuring an irradiance, from opening the shutter to calculating the irradiance from the calibration coefficients.

The block diagram of the major system components, Fig.5.9, shows how the set of radiometers is controlled and monitored by a program running on the BBC. This program, the "Gather And Measure Program" (GAMP), was written in BBC Basic, and uses the '6502' assembler library supplied with the IEEE 488 interface to address the hardware. It has two main sections: the CALIBRATE section, and the MEASURE section - the latter sub-program using the information gathered by the former.

Fig.5.9: Schematic of Remote Control of Radiometer



The mechanical construction of the radiometer is shown in Fig.5.10. It is a very simple device which consists essentially of an aluminium plate mounted on a rotary solenoid which acts as a shutter either admitting or excluding the radiation depending on whether or not the solenoid is energized.

The electrical circuitry, shown in Fig.5.11, is based around a single chip, the thermocouple amplifier (AD595), which has internal cold-junction compensation and runs from a single sided 5 V supply (RS Components Ltd., 1990). This amplifier provides an output of 10mV/°C over the range of interest. It is pretrimmed for type K thermocouples, has a maximum gain error of 1.5 %, a 3dB bandwidth of 15kW, and is specifically designed for use in such an application.

Due to problems with the switching transient and mains frequency (50 Hz) noise it was decided to implement a delay and a digital transversal filter in software. The algorithm for the filter is demonstrated in Fig.5.12 (Lynn, 1982). The filter was tuned by the introduction of a variable count delay loop to have a sampling frequency = 32.1 Hz. This results in a first minimum at 10.1 Hz, and a subsequent notch at 50.5 Hz. The software filter improved the noise rejection of the device considerably.

5.2.2 Calibration

In order to provide a more controllable source of radiation than was used in the previous calibration, a small electrical oven was modified by the addition of a mask to the door. The mask had a circular hole cut in it. The oven was thermostatically controlled and provided a convenient source of radiation which varied with an amplitude of 7% over a period of 500 seconds. The steepest gradient

Fig.5.10: Mechanical Details of Automated Radiometer

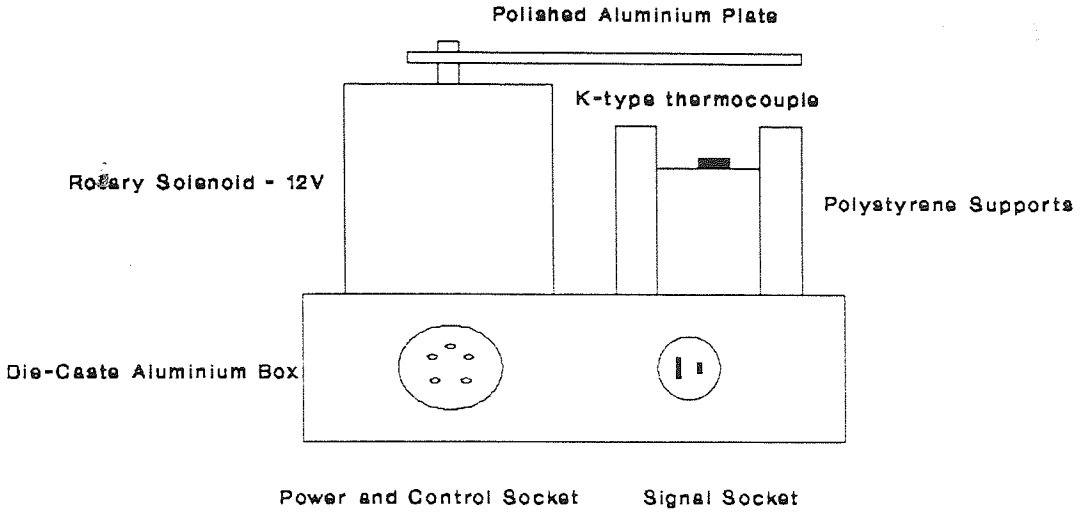


Fig.5.11: Electrical Connections to AD595

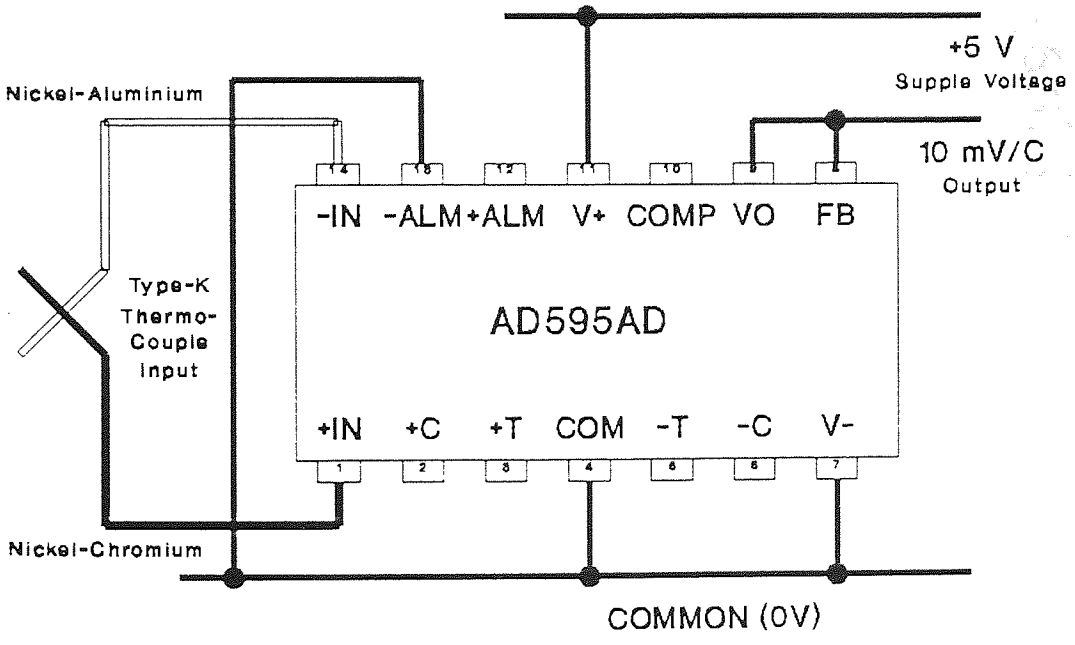
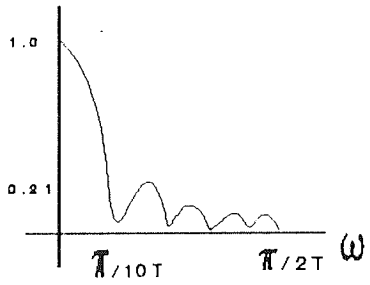


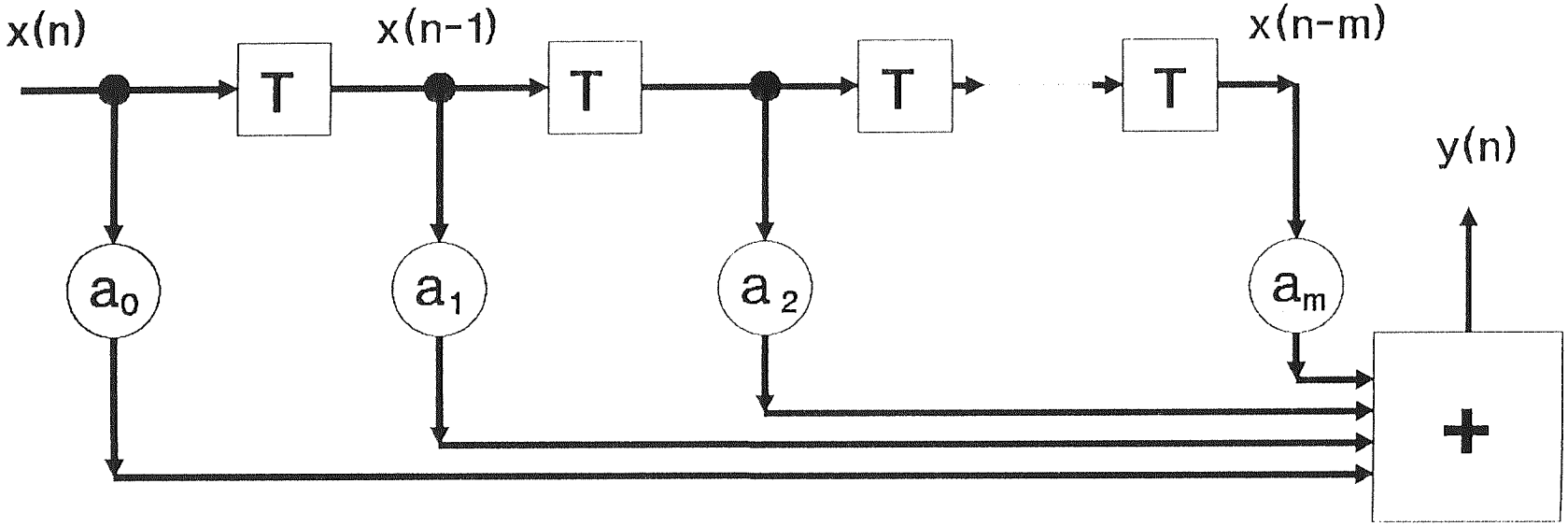
Fig. 5.12: Digital Transversal Filter

Algorithm

$$y(n) = \sum_{i=0}^{m} a_i x(n-i)$$

Frequency Response





measured was 75 mW m^{-2}/s. This would have resulted in a change in irradiance over a typical 3 second sampling period of ≈ 0.25%. It was concluded that the oven provided a satisfactorily stable source of radiation.

An optical bench was used to mount the radiometers under test in front of the furnace, see Fig.5.13. The gradients produced by the automated radiometers were compared with the corrected output from the Land Radiometer at ten different distances. These distances were calculated according to an inverse square law to provide a linear ordinate spacing.

A linear regression was then performed which resulted in the coefficients given in Table 5.7.

Table 5.7 Linear Regression Coefficients

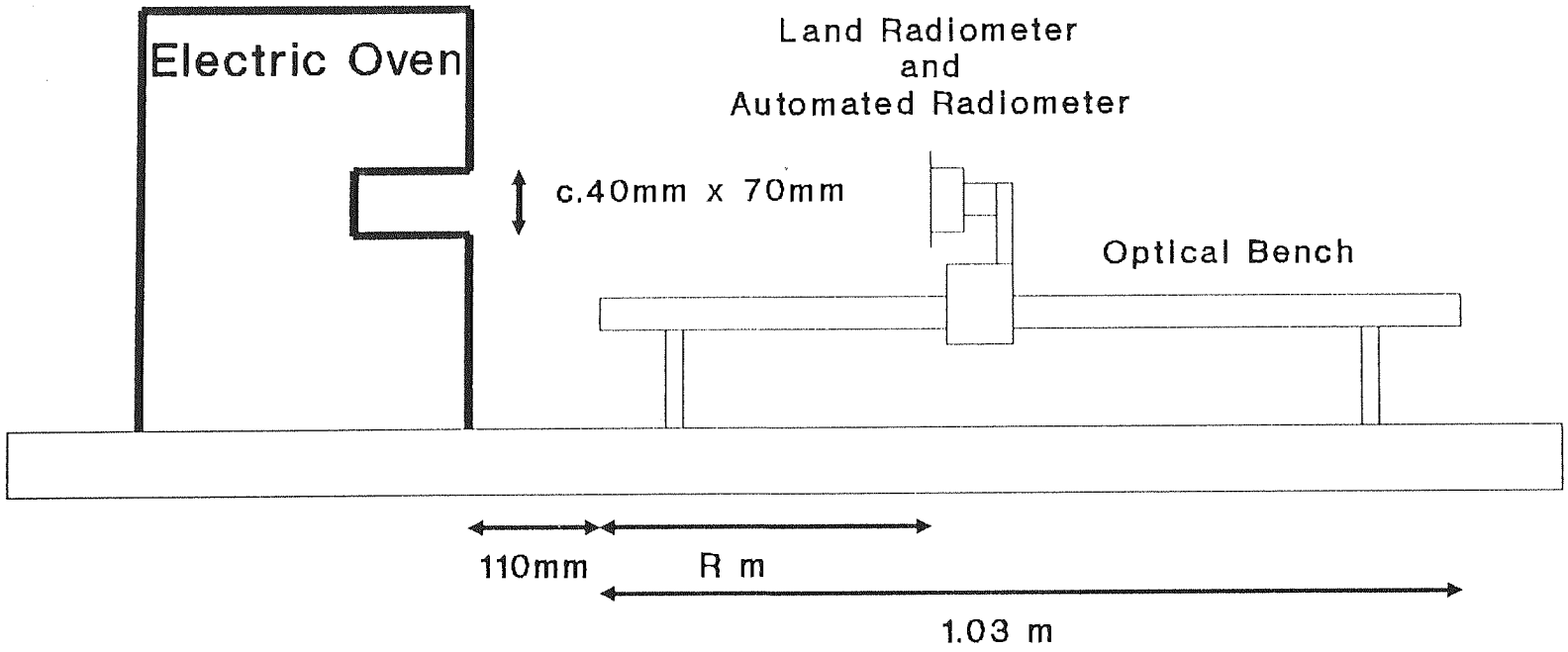
Radiometer	Gradient Wm^{-2}/mVs^{-1}	Intercept Wm^{-2}	R-coefficient	Error (%)
A	61.49	30.85	0.993	0.7
B	66.01	21.50	0.992	0.8
C	67.15	43.74	0.991	0.9
E	76.12	20.41	0.993	0.7
F	65.57	35.73	0.971	2.9
G	61.47	18.39	0.982	1.8
H	71.57	3.12	0.994	0.6
I	64.21	11.02	0.997	0.3

The reduction in the errors from ≈ ±5% to ≈ ±0.5 % can immediately be seen. This resulted from the precautions outlined above, plus careful screening and earthing of signal cables.

5.2.3 Measuring the Irradiance of the Unit IRH_2

The procedure for measuring the irradiance of the IRH_2 was identical in form to that used to measure the IRH_7, with the exception

Fig.5.13: Experimental Setup for Calibration



that the four automated radiometers, A, B, C, and E, were used in place of the original radiometers. The arrangement of the apparatus for the experiment is shown in Fig.5.14, whilst the power balance appears in Table 5.8.

Table 5.8: Power Balance for IRH_2

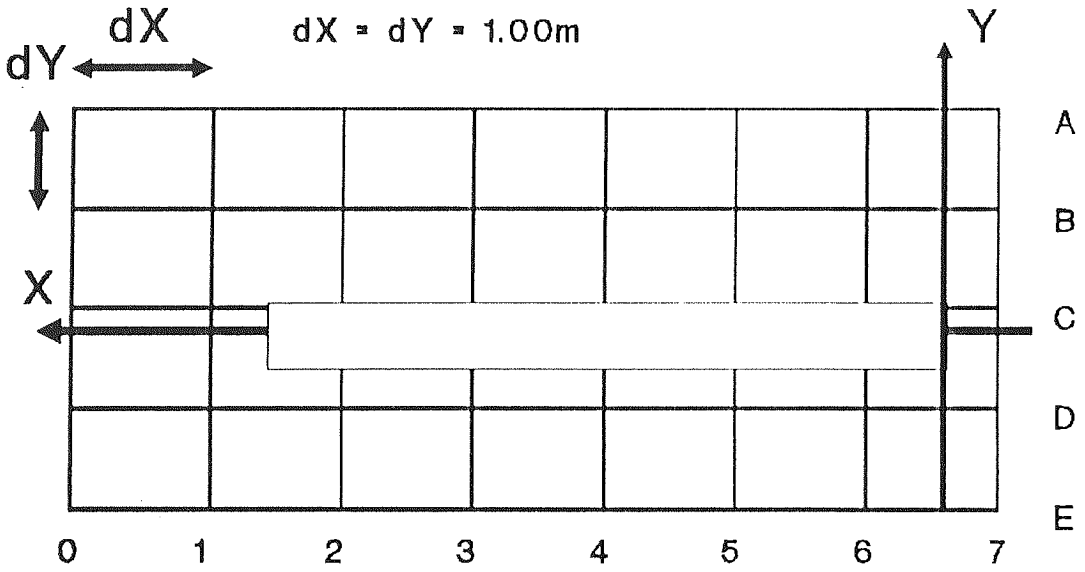
	IRH_7	
	Power (kW)	Power (%)
Input Power	30000	100.0
Radiation to Zone	8403	28.0
Atmospheric Absorption	714	2.4
Convection from Reflector	2882	9.6
Convection from Tube	9000	30.0
Flue Loss	7500	25.0
Radiation from Reflector	380	1.3
Total	28880	96.3

It should be noted that the power balance is nowhere near as complete a description as for IRH_7. This is because readings were not taken of the gas input, the oxygen content of the flue, or the convection from the tubes. In the table the nominal manufacturer's rating is taken for the gas input and the values for the stack loss and the convection from the tubes were assumed to be consistent with those obtained from the work performed on IRH_7. The agreement between the model and the measurements was so good that it was decided to use the model to extrapolate the readings beyond the area that had been available in the Pilot Plant for measurement. This only makes a difference of a couple of percentage points but is justified insofar as it is a good way of ensuring that the "missing" irradiance is as small as possible.

The measured and modelled irradiance plots are shown in Fig.5.15, 5.16, and 5.17. The temperatures used in the model were extracted from results measured on equipment described in the section concerned with thermometry. Table 5.9 summarises the data that was

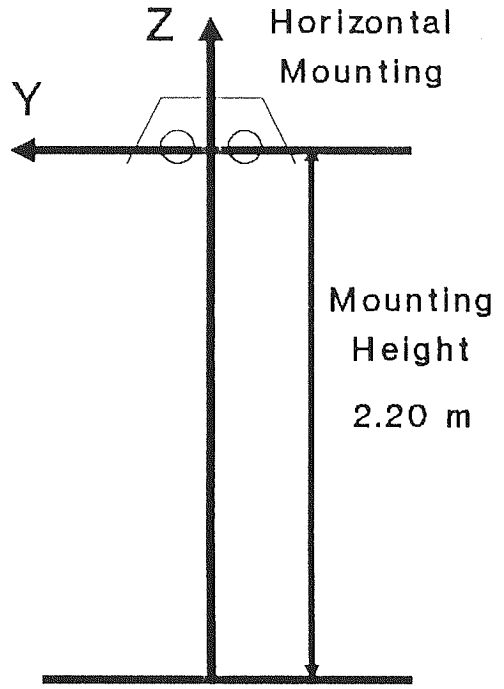
Fig.5.14: Geometry of IRH_2

Plan View



A0 is at (6.34,2.67,-2.20)

Cross Section



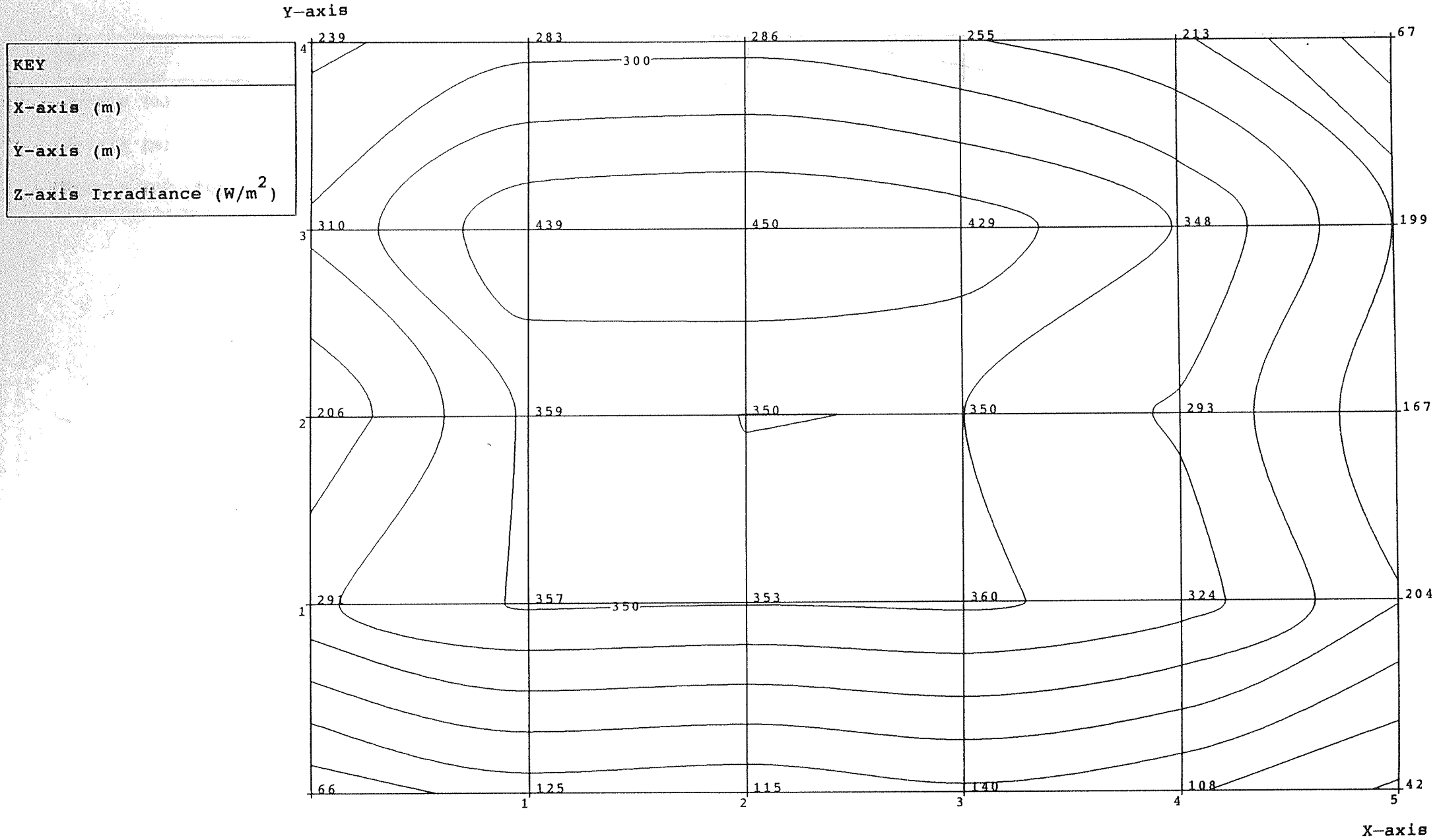


Fig.5.15:IRH_2_Measured Irradiance Plot

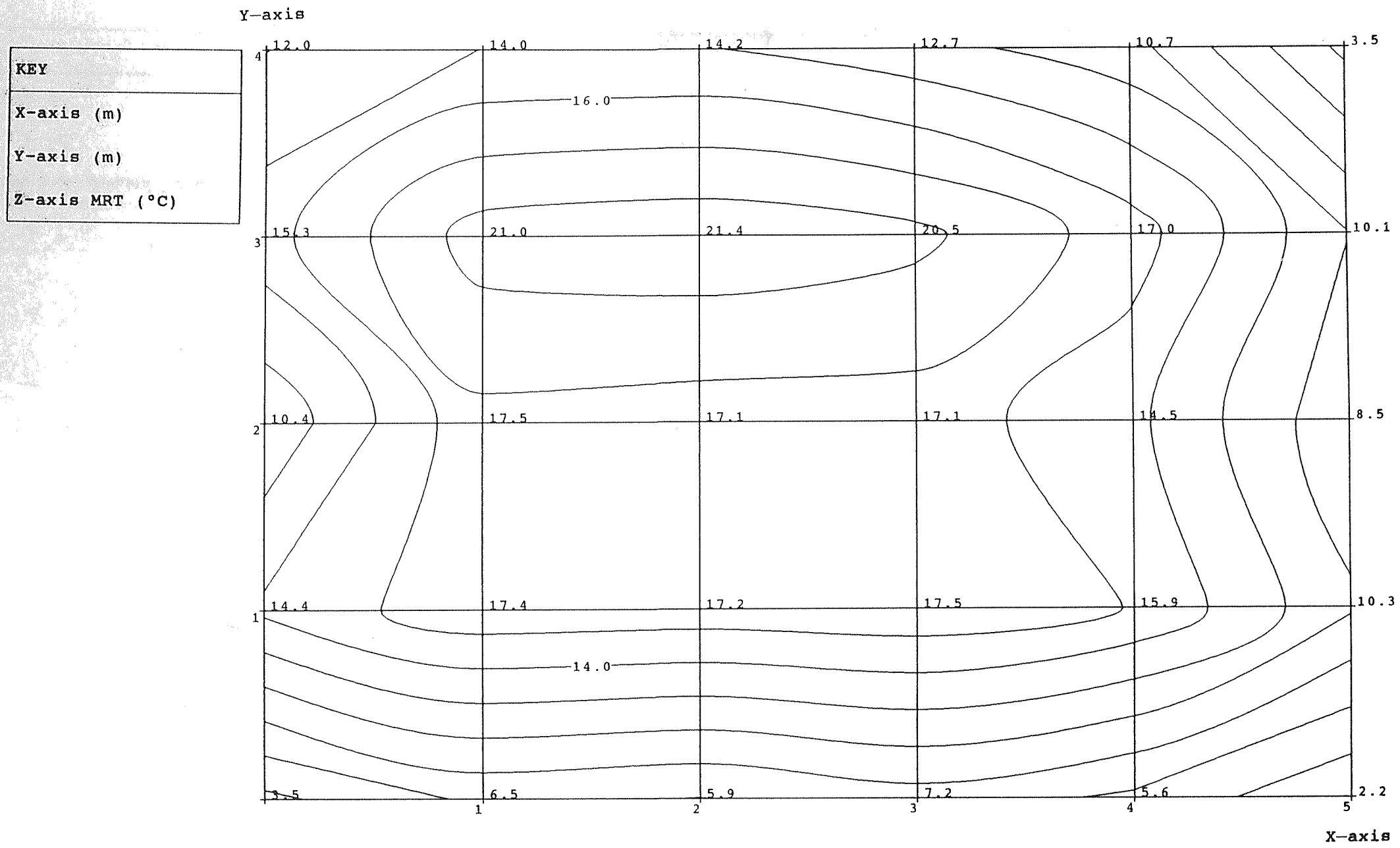


Fig.5.16:IRH_2 Black Bulb MRT Plot

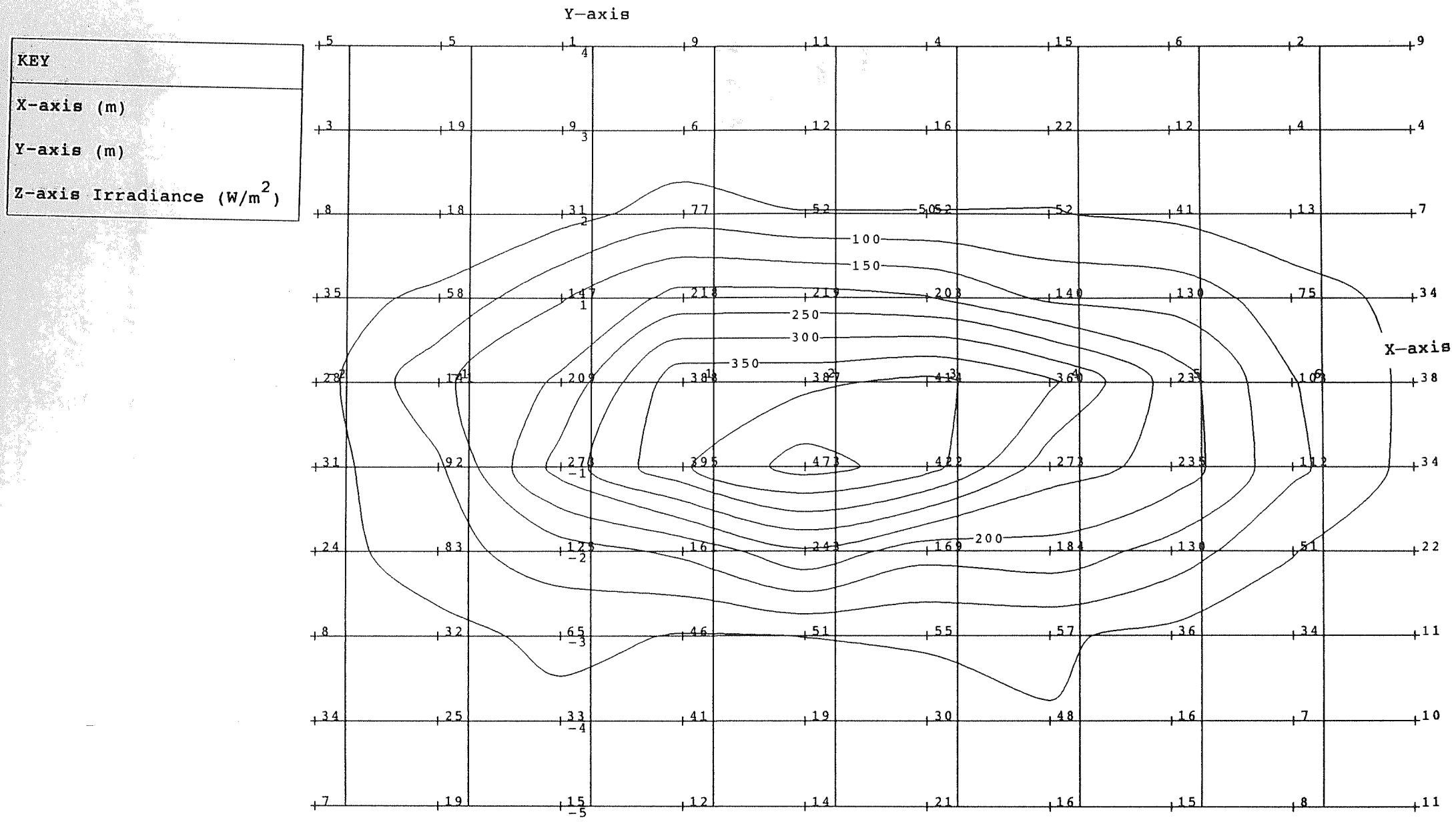


Fig.5.17:IRH_2_Modelled Irradiance Plot

used in the model, and also to evaluate some of the elements of the power balance.

Table 5.9: Temperature Data for IRH_2

Position	Tube 1	2	3	4	Reflector	Flue Gas
B2	414.	319.	486.	505.	133.	205.
B3	419.	315.	370.	442.	123.	204.
B4	358.	237.	382.	393.	117.	204.
B5	348.	294.	278.	354.	120.	207.
F1	192.	172.	195.	205.	90.	207.
F2	194.	204.	222.	238.	-	207.
F3	253.	246.	226.	261.	108.	206.
F4	267.	246.	251.	292.	112.	208.
F5	288.	277.	282.	333.	109.	205.

NOTE: Figures 6.1 and 6.2 show the positions of the measuring points on the tube.

It is evident that whilst this heater has a higher absolute output, its efficiency is much lower. This is simply due to poor reflector design as can be demonstrated by altering the design to a more "open" design, heater IRH_8 (Fig.5.18). A direct comparison between the predicted radiant power to surfaces of the two heaters, IRH_2 and IRH_8, is made in Table 5.10. The calculation of these values has assumed the same tube temperatures for both heaters, which is not an unreasonable assumption considering the minor change made to the geometry.

Table 5.10: Comparison of IRH_2 and IRH_8 Radiation to Surfaces

Heater	IRH_7 Power (kW)	IRH_7 Power (%)	IRH_8 Power (kW)	IRH_8 Power (%)
Input Power	30.0	100.0	30.0	100.0
Radiation to Zone	8.40	28.0	10.5	35.0

Fig.5.18: Reflector IRH_8

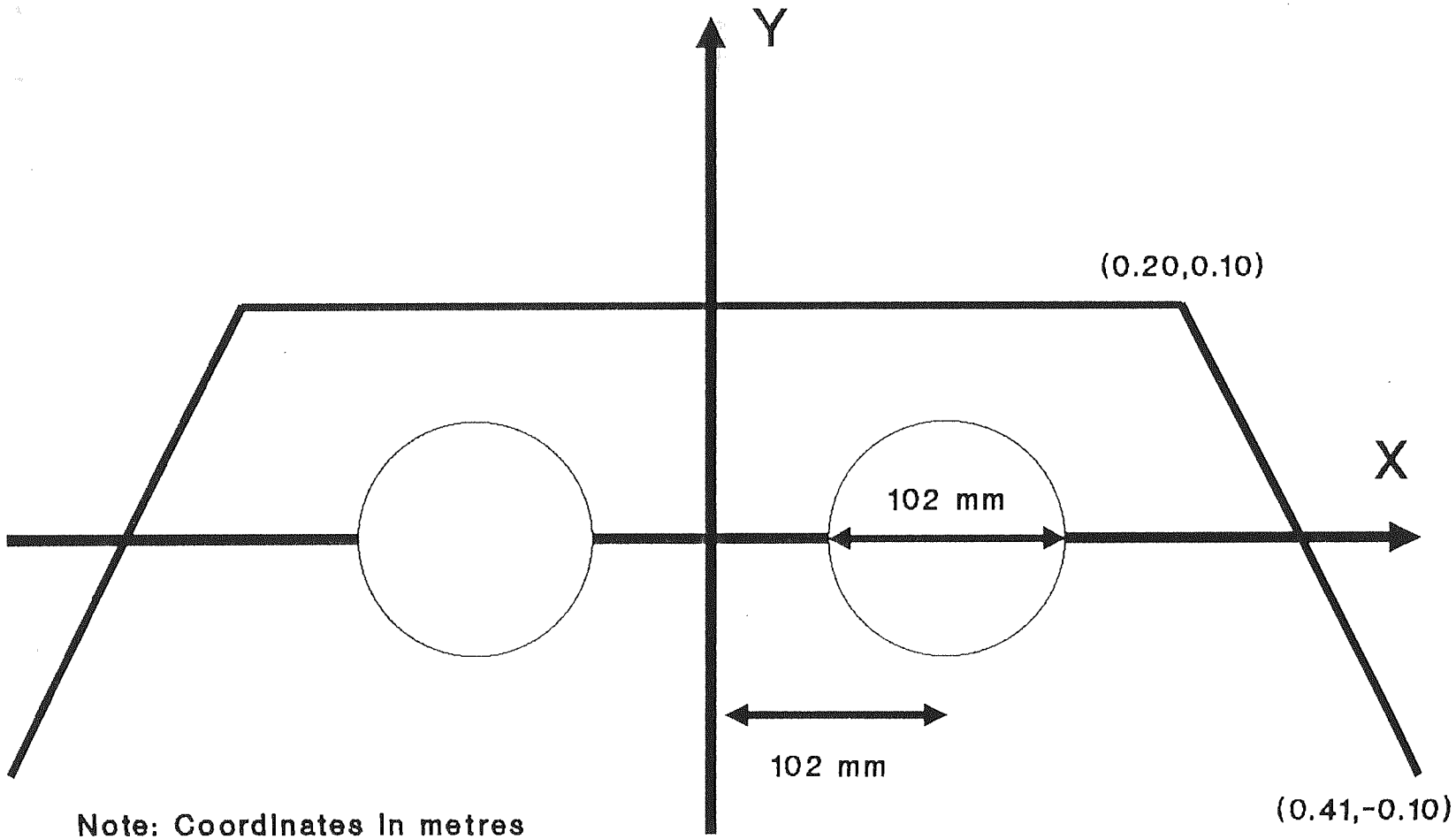


Table 5.11 shows a comparison between the three heater designs IRH_7, IRH_2, and IRH_8 in which the actual modelled output is given as a proportion of the maximum possible output based on a simple Stephan-Boltzmann evaluation of the output power from the tubes.

Table 5.11 Comparison of Outputs with Maximum Available Power

Heater	Modelled Output (W)	Maximum Output (W)	Proportion (%)
IRH_7	5842	12945	45.13
IRH_2	8403	21614	38.88
IRH_8	10488	21614	48.52

It is clear from this table and the power balances that the design of the reflector plays a significant part in the quest to improve overall efficiency. In the instance cited a difference of 2kW in the output of the heater is effected simply by changing the reflector shape.

5.3 The Land Radiometer

The measurements in the previous two sections were taken with wide-angle radiometers that were constructed and calibrated at Aston University. Despite all developments to date these devices are not as reliable, accurate, or easy to use as the Land Radiometer against which they were calibrated. The specifications of the Land given by the manufacturer are shown below.

Table 5.12: Land Radiometer - Type RAD/P

Sensitivity (Open Circuit Output)	S_{1000} =	5.0 μV $W^{-1}m^2$ (at 1000 K)
Calibration Accuracy		± 3 %
Angle of Complete Vision		± 25 °
Cut-off Angle		± 40 °
Ambient Temperature Compensation		0 - 70 °C
Measuring Range		0 - 10000 Wm^{-2}

Corrections

1) Source at T (°C) $S_T = (a + bT + cT)S_{1000}$

Where

a = 0.9076; b = 1.216e-4 $°C^{-1}$; c = -2.971e-8 $°C^{-2}$

2) Output voltage E (V) $S_E = S_{1000}(1 - 0.0014E)$

In the investigation of PLAQ_1 the readings were all taken manually and the corrections were calculated on a hand-held calculator. For the QLL heater experiments, which are not documented but are included in the summary of results, the PET data-logger was used to gather and calculate the results at the same time. The difference in procedures allowed more results to be taken in the second instance with a commensurate improvement in accuracy.

5.3.1 The Performance of PLAQ_1

The method of investigating the heating capabilities of the PLAQ_1 unit were essentially similar to those employed with IRH_7. The most notable exception, as mentioned above, was the use of the Land

Radiometer because of the geometry of the apparatus.

The procedure followed was:

1) Note down the physical dimensions of the apparatus and the grid used, cf. Fig.5.19.

2) Measure the input to the device by recording the rate of gas flow into the heater using a gas meter.

3) Perform a flue gas analysis to ascertain the percentage oxygen in the flue gas and measure its temperature, so that the stack losses can be computed.

4) Measure the irradiance over the "occupied" zone using the Land Radiometer.

5) Measure the reflector and baffle temperatures in order to estimate the convective and radiative heat losses from the casing.

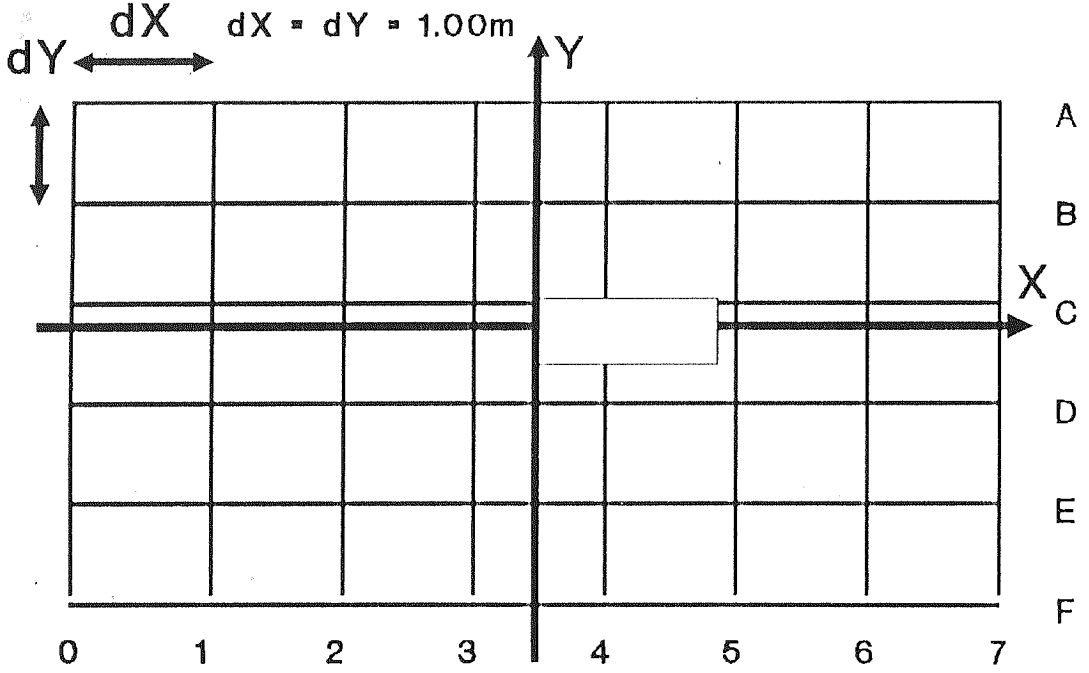
The flue gas analysis proved to be unreliable until a hardboard "chimney" was fitted to the heater to funnel the exhaust gases; the results then became more consistent. The results given below are with the chimney fitted. The irradiance was in fact measured twice, once with the chimney fitted and once without. There proved to be no substantial alteration to the performance of the heater.

Table 5.13: Power Balance for PLAQ_1

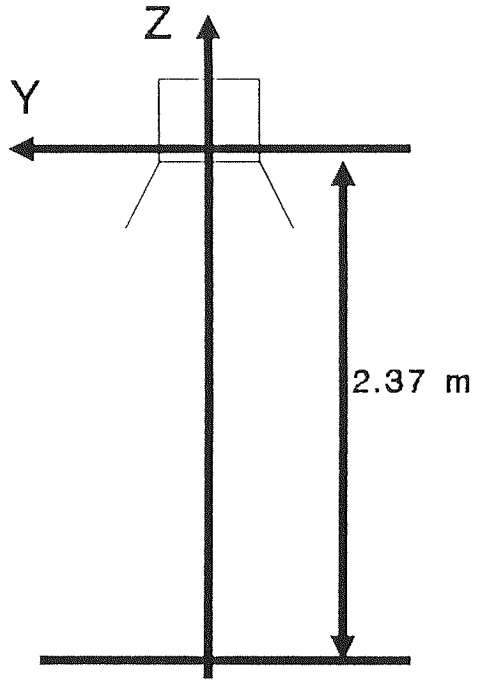
	W	%
Input	16707	100.0
Radiation to zone	7099	42.5
Atmospheric Absorption	560	3.3
Casing Convection	271	1.6
Casing Radiation	40	0.2
Stack Losses	9857	59.0
Total	17828	106.7

Fig.5.19: Geometry of PLAQ_1

Plan View



Cross Section



The most certain of these figures is the radiation to the zone which has an accuracy of ±2 percentage points, whereas the stack losses are only accurate to ±5 percentage points at best because the heater was operating at such a high level of excess air (≈750%) that the graphs used for finding stack losses were at the edge of their range. Thus, the difference between the input and output figures of nearly 7 percentage points is within the range of experimental error.

Even allowing for these inaccuracies, the power balance clearly shows that a much higher proportion of the available energy is being directed to where it is useful, ie the occupied zone. This trend of higher temperature leading to greater efficiency will be brought to its logical conclusion in the section dealing with QLL heaters.

The energy distribution across an occupied zone and the values of MRT derived from the measurements are shown in Fig.5.20. This reveals a much more compact pattern than was evident for the tubular heaters, with much steeper gradients across the zone.

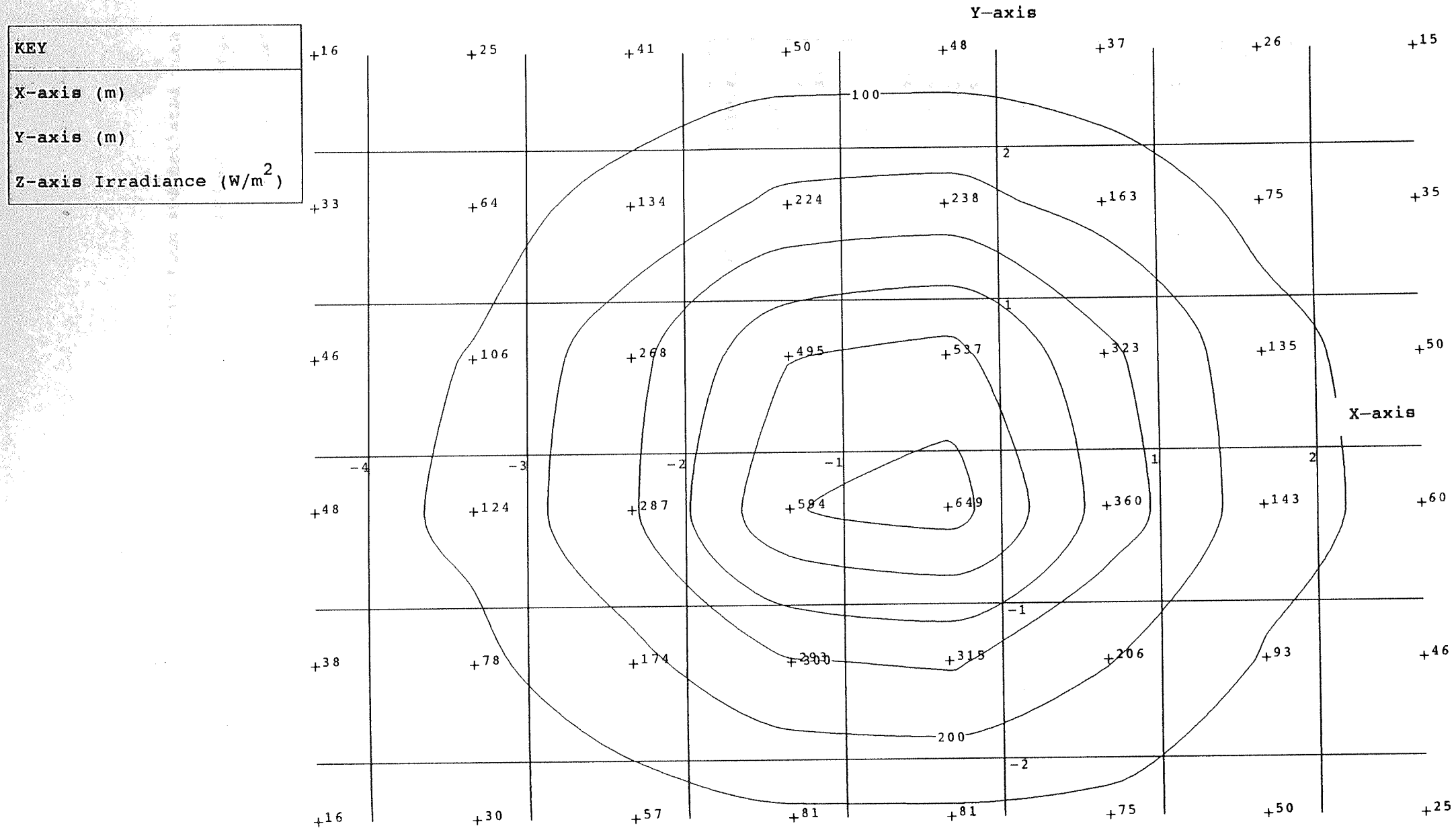


Fig.5.20:PLAQ_1 Irradiance Plot

5.4 Summary of Results

In this section three sources of information are collated with the aim of arriving at some general conclusions about radiant heating. The sources are the experiments outlined above (plus others performed on QLLs which have not been fully documented), an extensive series of tests of plaque and QLL heaters carried out at the British Gas Midlands' Research Station in 1988 (Millward, 1988), and lastly a report on four tube heaters and a plaque heater carried out at Aston University in 1979 (Maund, 1979). A comprehensive summary of the results from these three sources, Table 5.14, is included below.

Table 5.14: Summary of Experimental Results

Heater	Experimenter	Emitter Temperature (K)	Mounting Height (m)	Input Power (kW)	Radiant Output Power (kW)	(%)	Maximum Irradiance (W/m^2)	(%)
PLAQ_1	JKM & CDZ	≈1000	2.44	16.95	6.95	(41.0)	723	(10.4)
PLAQ_2	SM		2.44	7.32	3.10	(42.4)	207	(6.7)
PLAQ_3	SM		2.44	5.39	3.04	(56.5)	217	(7.1)
PLAQ_4	SM		2.44	11.48	6.82	(59.4)	500	(7.3)
QLL_1	CDZ	≈2700	2.13	1.5	1.11	(74.2)	208	(18.7)
QLL_2	CDZ		2.13	4.5	3.48	(77.3)	324	(9.3)
QLL_3	SM		2.44	4.44	3.67	(82.8)	472	(12.9)
QLL_4	SM		2.44	4.53	3.45	(76.2)	532	(15.4)
QLL_5	SM		2.44	4.58	3.53	(76.9)	390	(11.0)
QLL_6	SM		2.44	4.02	3.46	(86.1)	349	(10.9)
IRH_1	JKM & CDZ	≈ 600	2.20	30.	14.95	(49.5)	575	(3.8)
IRH_2	CDZ		2.20	30.	8.4	(28.0)	473	(5.6)
IRH_3	JKM		2.20	17.6	6.0	(34.0)	300	(5.0)
IRH_4	JKM		2.20	19.1	5.8	(30.3)	≈300	(≈5.)
IRH_5	JKM		2.20	14.7	5.2	(35.2)	≈250	(≈5.)
IRH_6	JKM		2.20	31.3	10.6	(33.9)	460	(4.3)
IRH_7	JKM & CDZ		2.13	21.6	7.1	(32.9)	315	(4.4)

Note: The two parenthesized sets of figures quoted, the percentage of Output Power and the percentage Maximum Irradiance, are taken as a proportion of the Input Power and Output Power respectively. The latter quantity is a measure of the focussing of the heater's output.

This summary clearly demonstrates two important points:

1) As the temperature of the emitter rises so does the efficiency of the heater unit. The efficiencies of tubes fall within the range, 28 to 36%; the plaques, 40 to 60%; and the QLLs, 75 to 87%. This is primarily due to the interplay between the functional dependencies of the heat transfer coefficients on temperature. The radiation HTC depends on T^4, whereas the convective HTC depends on $T^{4/3}$. Thus the overall balance between radiation and convection will tend towards radiation as the temperature increases.

2) The ratio of the maximum irradiance to the total output is an indication of how focussed or diffuse the radiation is. A low percentage implies that the radiation is spread out over the zone, whereas a high proportion implies the opposite. This point was briefly covered in section 2.3.5 in relation to the output from the model. These experimental figures confirm the view that QLL heaters are inapplicable to wide-area heating.

5.5 Conclusion

It must be borne in mind that the figures presented in this chapter only apply to steady state conditions, a point which is of most relevance to the tube heater. Also, the modelled results are for free-field heaters, ie heaters in total isolation, whereas the measured results were taken within the confines of a laboratory. The experimental work performed has resulted in the following conclusions:

1) The Monte Carlo model is an essentially accurate representation of the radiation transfer processes taking place in a radiant heater, and thus it produces results which agree in substantial measure with those found by experiment.

2) The efficiencies of the heaters depend critically upon the balance of radiative and convective heat transfer of the emitter and the reflector.

3) Whilst the efficiencies of compact, hot emitters are higher than those of large, relatively cooler emitters, the thermal comfort produced by the latter is undoubtedly superior.

It should also be recalled that none of the preceding comments attempt to account for the relative prices of the fuels used to power the heaters. Clearly in any full analysis of the heating needs of a site this factor may well outweigh other considerations.

6 Thermometry

The model which has been expounded in chapters 2 and 3 of this dissertation is a semi-empirical one. It does not seek to predict the behaviour of heater units starting entirely from fundamental principles, but relies on the availability of experimental data concerning radiant source temperatures. The accuracy of these predictions depends on several factors. Firstly, it is affected by the reliability of the data, which can be ensured by sound experimental technique. Secondly, it depends on the resolution, both spatial and temporal, of the measurements. Lastly, the algorithm chosen to interpolate between data points can in some circumstances prove to be important: this is especially true when the resolution of the experimental measurements is low.

Both the experiments described below were performed in an attempt to improve on the existing state of affairs regarding the first two of these factors. Using both contact and optical methods of temperature measurement improves the reliability of the results as cross-checks can be made. The temperature calipers, that are described in the first section on contact methods, generated low-resolution data which was highly reliable as it did not depend upon knowledge of the surface emissivity or upon making corrections for absorption by an intervening medium. The thermographic camera generated vast amounts of high-resolution data which, if used with care and in conjunction with other temperature measurements, can give detailed knowledge of the temperatures of the tubes.

Another feature of these experiments was that both provided the opportunity to look at the thermal behaviour of the tubes as a function of time. This is something which has not been attempted before. All previous investigations of radiant heaters have focussed on the steady state behaviour. When it is remembered for how little of the duty-cycle a real heating system is at steady state, it becomes clear that previous approaches have been inadequate.

The results produced were then used as a basis for calculating the irradiances of the heaters as a function of time. They also served as a basis for the description of a heater as a conceptual "system" with its own intrinsic transfer function in Laplace Transform space. This, in turn, led to the concomitant space-state description of the heaters used in chapters 3 and 4.

6.1 The Contact Method

6.1.1 Equipment and Method

Fig.6.1 shows the positions on the tubes where the calipers were attached during the nine experimental runs, whilst Fig.6.2 gives an indication of the mechanical construction of the calipers along with a block diagram of the electrical equipment used to record and process the readings. Each set of data was gathered in a separate run lasting for about 70 minutes. This was because there was only one pair of calipers available. Every effort was therefore made to ensure that the experimental sequence was the same for each run so that the results could be collated as a whole.

The experimental sequence was as follows:

1) Start the datalogger and record the time.

2) Wait for 30 seconds and then switch on the power to the heater. The heater took 10 seconds to fire, this being the time taken for the automatic controller to run through its test sequence.

3) Measure the gas pressure with a water-filled manometer, and take the wet and dry-bulb thermometer readings.

4) Allow the heater to warm up for 24 minutes with the data-logger recording the temperatures every 30 seconds.

5) Switch off the power to the unit 15 seconds after start of a data-reading cycle to allow for all the channels to be read, and note the stop time (the gas supply to the unit is stopped instantaneously).

6) Allow the unit to cool for 45 minutes whilst still logging, and then for a further 90 minutes without monitoring before

Fig.6.1: Tube Sections for Calipers

Plan View

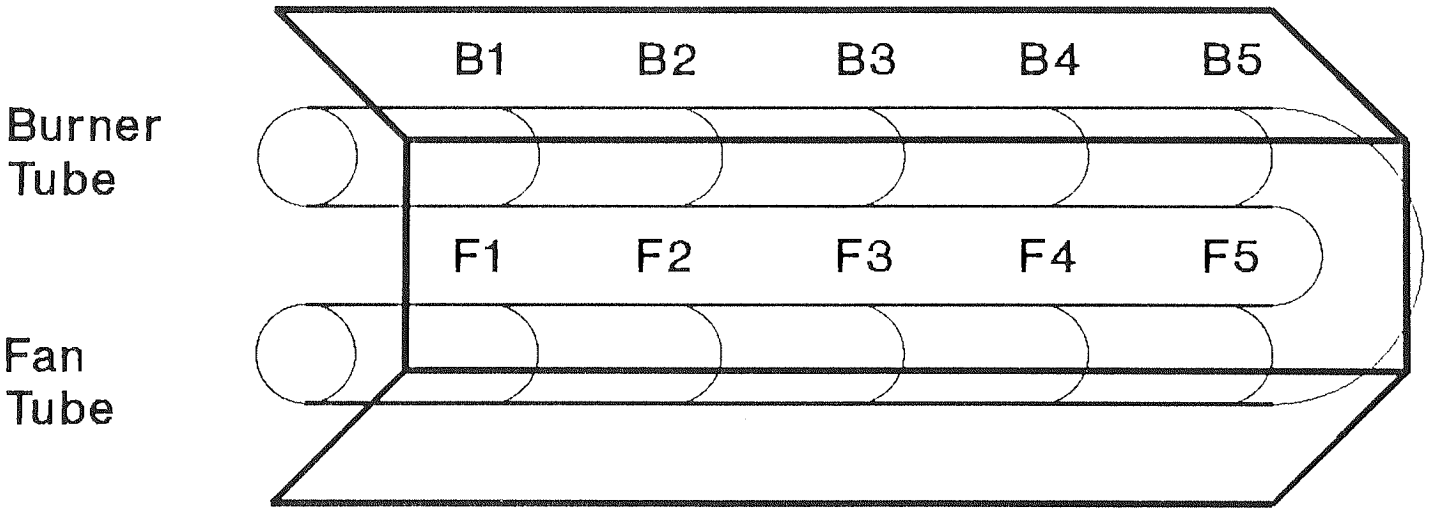
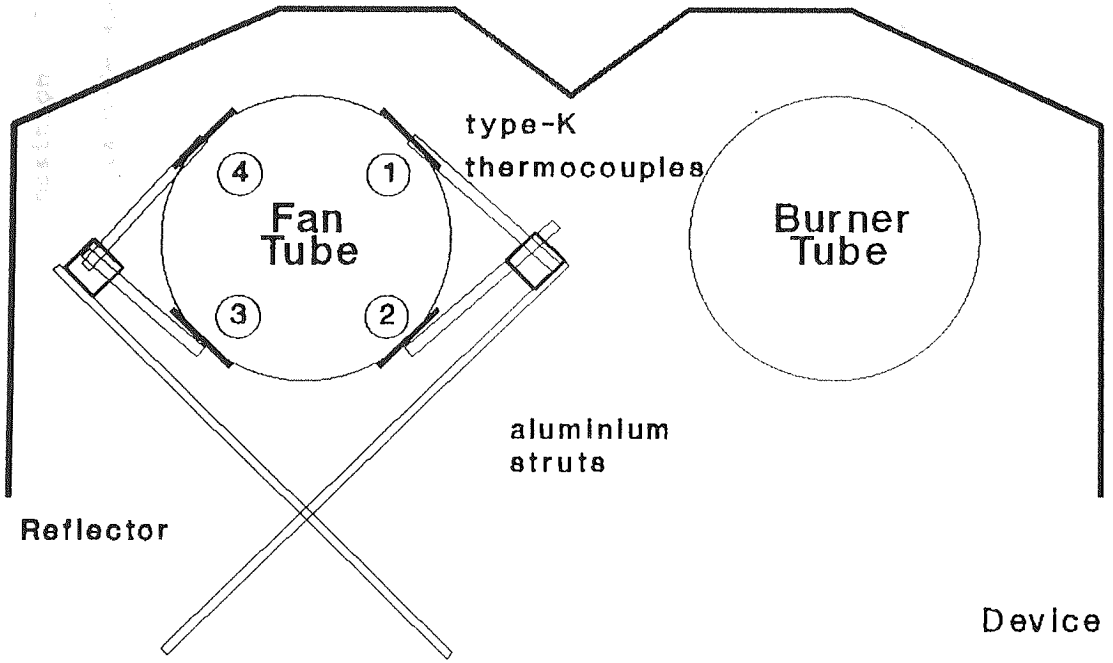


Fig.6.2: Temperature Calipers

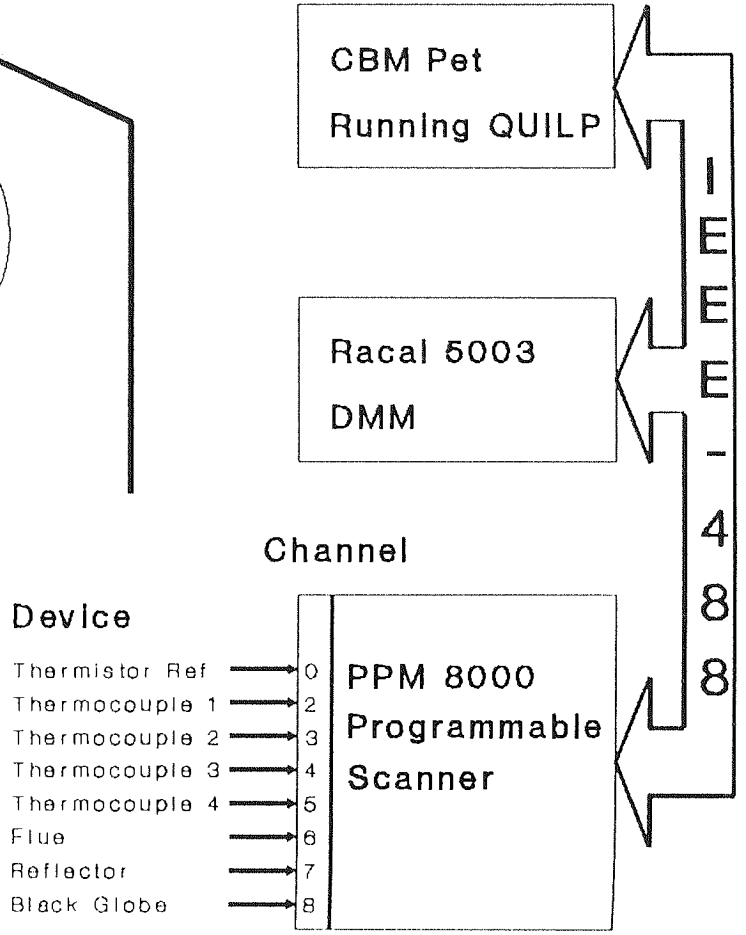
Cross-Section



Channels 2-5 to
Programmable Scanner

Note: Looking towards Burner

Electrical



starting the sequence again for the next position.

Table 6.1 summarizes the data files made and the conditions under which they were recorded.

Table 6.1: Summary of Experimental Conditions

		Heater On				Heater Off				Stop
Filename	Gas Press " wg	Tdry °C	Twet °C	RH %	Time	Tdry °C	Twet °C	RH %	Time	Time
IRH_F1	7.7	-	-	-	12:30	-	-	-	12:54	13:43
IRH_F2	7.8	-	-	-	10:19	-	-	-	10:44	11:28
IRH_F3	7.7	23.5	-	-	07:33	28.0	22.5	62	07:59	08:45
IRH_F4	7.7	25.0	19.6	60	14:41	31.0	23.5	53	15:09	15:55
IRH_F5	7.4	23.0	19.8	76	08:34	29.2	23.2	60	08:59	09:43
IRH_B5	7.8	24.5	20.0	66	12:04	31.0	23.5	53	12:31	13:15
IRH_B4	7.4	23.0	-	-	08:00	28.7	22.2	57	08:25	09:10
IRH_B3	7.5	25.5	21.0	73	14:03	32.5	24.3	51	14:41	15:33
IRH_B2	7.8	24.0	20.5	73	07:55	-	-	-	08:20	09:05

Two other points are worth making: firstly the temperatures were calculated by adding the thermocouple temperature (found using factors in Table 6.2) to the internal reference temperature of the PPM scanner which is generated by a thermistor and compensation circuit which produces a voltage of 10mV/°C. Secondly, the caliper was always orientated in such a manner as to maintain a consistent orientation of the thermocouples to the tube. Thus, looking towards the burner end, for the measurements of the fan tube thermocouple number 1 was in the top right position, and for measurements of the burner tube thermocouple number 4 was in the top right position.

Table 6.2: Thermocouple Constants for Type-K Thermocouple

A quadratic fit was used

$$T = a + b.T + c.T^2 \qquad (6.1)$$

where the values of the constants were:

Temperature Range	a	b	c
0 - 400 °C		2.4689 e-2	-1.4594 e-8
0 - 1370 °C		2.3041 e-2	2.5075 e-8
400 - 1370 °C	5.8998 e1	1.9745 e-2	7.5045 e-8

(Biodata Ltd.)

The quoted accuracy of type-K thermocouples is ±2.5°C or 0.0075T (where T is the measured temperature in °C) whichever is the larger. (RS Components Ltd., 1990)

6.1.2 Results

As stated above this experiment had two objectives: firstly, to provide information which could be used in the mathematical model to show the output of a heater unit as a function of time; secondly, to try to discover if there was a reasonably simple method of describing the heater's dynamic performance. It was to this latter end that the subsequent analysis was performed.

Neglecting momentarily the complexities of the divers heat transfer mechanisms and concentrating instead on the functional form of the heating and cooling curves, Newton's law of cooling suggests that the data might well fit a curve of the form

$$T(t) = T(\infty) - (T(\infty) - T(0)).A_h.\exp\left(-\frac{t}{\tau h}\right) \qquad (6.2)$$

when the unit is heating up, and

$$T(t) = T(\infty) + (T(0) - T(\infty)).A_c.\exp\left(-\frac{t}{\tau c}\right) \qquad (6.3)$$

when the unit is cooling down, where:

t = time (s)

$T(t)$ = instantaneous temperature (°C)

$T(0)$ = initial temperature (°C)

$T(\infty)$ = final temperature (°C)

A_h = heating normalisation constant

A_c = cooling normalisation constant

τh = heating time constant (s)

τc = cooling time constant (s)

These equations represent temperature differences decreasing exponentially with time constants τh and τc seconds. In order to test whether the data could be described in this way with sufficient accuracy, a short Fortran program, *LOGLSF.FOR*, was written. Using the NAG routine GO2CAF (a least squares linear regression) this program manipulated the data into the forms:

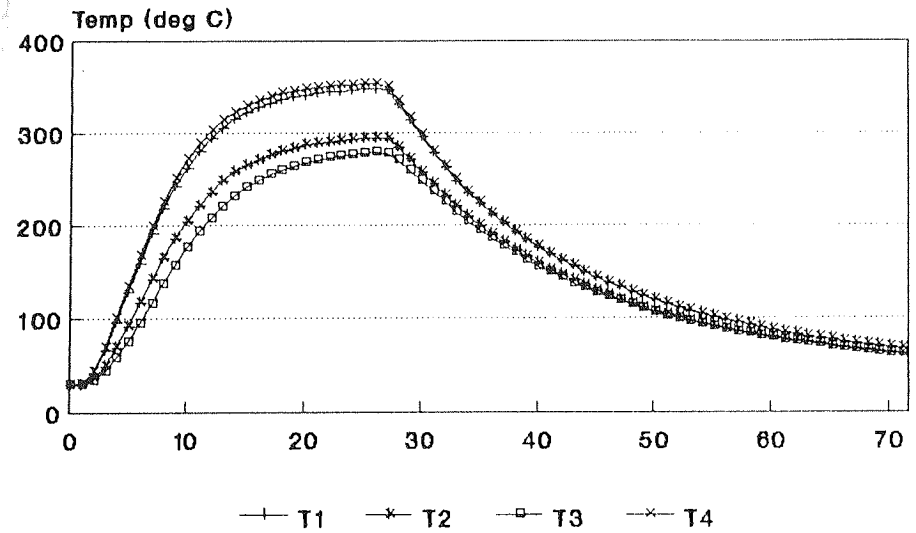
$$\log_e \left| \frac{T(\infty) - T(t)}{T(\infty) - T(0)} \right| = \frac{-t}{\tau h} + \log_e(A_h) \qquad (6.4)$$

$$\log_e \left| \frac{T(t) - T(0)}{T(\infty) - T(0)} \right| = \frac{-t}{\tau c} + \log_e(A_c) \qquad (6.5)$$

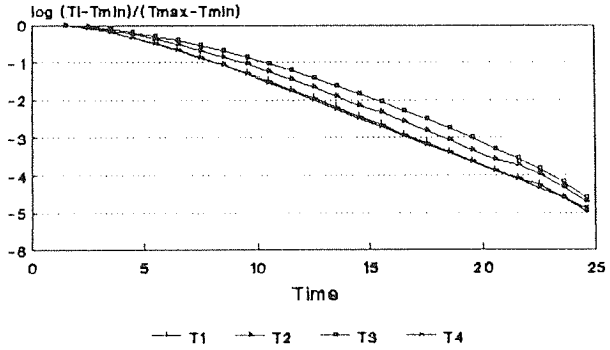
Values of τ_h, τ_c, and A_h, A_c were then calculated from the intercepts and gradients of the fitted linear plots.

Fig.6.3 shows several curves of both the unprocessed and semi-logarithmic plots after processing the data in the manner described above. Table 6.3 summarises the results from the nine data files and for all the relevant thermocouples.

Fig.6.3: IRH_2 Temperatures
Burner Section 5



IRH_2 Heating
Burner Section 5



IRH_2 Cooling
Burner Section 5

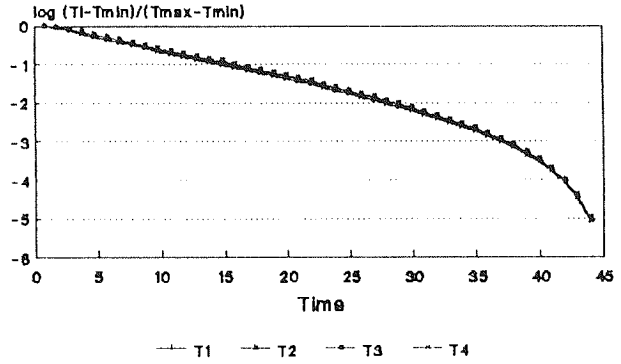


Table 6.3: Summary of Experimental Results for IRH_2

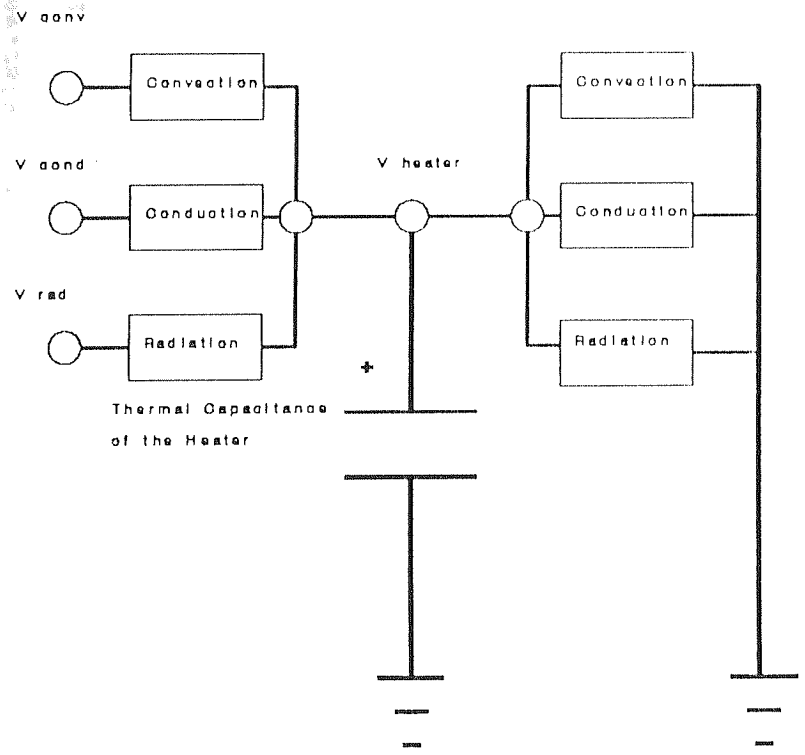
Mode	Section	τ1 (s)	A1	τ2 (s)	A2	τ3 (s)	A3	τ4 (s)	A4	Mean R
HEATING	B2	258	1.20	314	1.58	244	1.40	247	1.40	-0.98
	B3	387	0.77	389	1.03	372	1.07	365	1.07	-0.98
	B4	271	1.80	334	1.94	251	1.61	267	1.61	-0.99
	B5	274	1.97	290	2.11	303	2.26	273	2.26	-0.99
	F1	372	1.90	387	1.92	373	2.01	349	2.01	-0.95
	F2	376	1.99	358	2.00	332	2.05	320	2.05	-0.95
	F3	310	2.01	305	2.02	302	1.98	328	1.98	-0.98
	F4	318	1.98	326	1.98	317	2.01	297	2.01	-0.98
	F5	292	2.10	271	2.03	284	1.98	264	1.98	-0.98
COOLING	B2	547	1.61	578	1.68	562	1.32	559	1.32	-0.94
	B3	628	1.39	648	1.50	644	1.42	631	1.42	-0.95
	B4	571	1.60	596	1.85	568	1.44	568	1.44	-0.94
	B5	583	1.61	582	1.69	584	1.74	579	1.74	-0.94
	F1	662	2.10	1528	1.14	649	1.96	679	1.96	-0.94
	F2	618	2.06	625	1.94	616	1.89	619	1.89	-0.92
	F3	629	1.74	632	1.72	625	1.84	626	1.84	-0.93
	F4	611	1.81	619	1.79	623	1.77	614	1.77	-0.93
	F5	575	1.79	574	1.75	580	1.74	576	1.74	-0.93

Note: R is the Pearson Cross-Moment Correlation Statistic

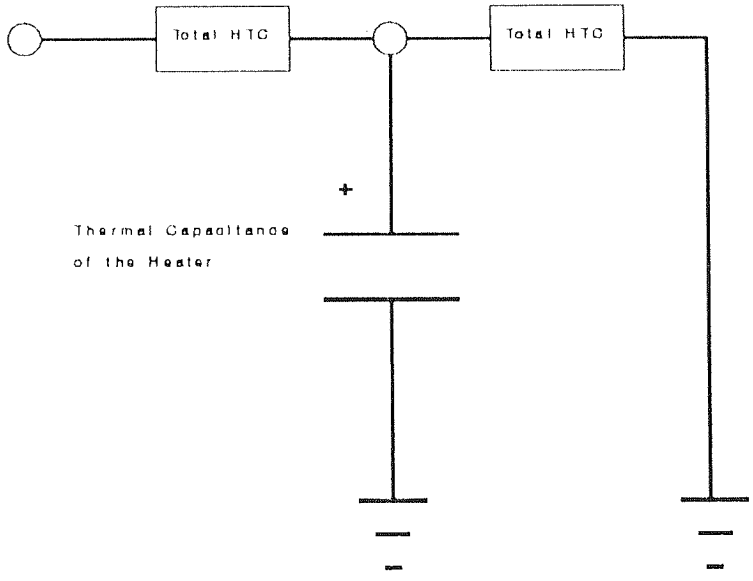
This table shows various interesting features. The most noticeable fact is that $2\tau_h \approx \tau_c$, where τ_h is the time constant for heating, and τ_c that for cooling. This can best be explained in physical terms by reference to another electrical analogy[1]. Fig.6.4 shows the electrical equivalent circuit consisting of a single capacitor, representing the thermal storage capacity of the metal, and six resistors, representing the three modes of heat transfer for the input and output respectively. These six resistors can be combined into just two resistors effectively giving one value of time constant for the input and one for the output. Clearly it is the ratio of these

Fig.6.4: Electrical Analogy of Heater Unit

Full Circuit



Combined Circuit



two "resistances" that will determine the relative sizes of the time constants. By solving the simplified first order differential equation for the heat flow it can be shown that:

$$\tau_c/\tau_h = 1 + R_c/R_h = 1 + G_h/G_c \qquad (6.6)$$

where R_h = Resistance to heating (K/W)

R_c = Resistance to cooling (K/W)

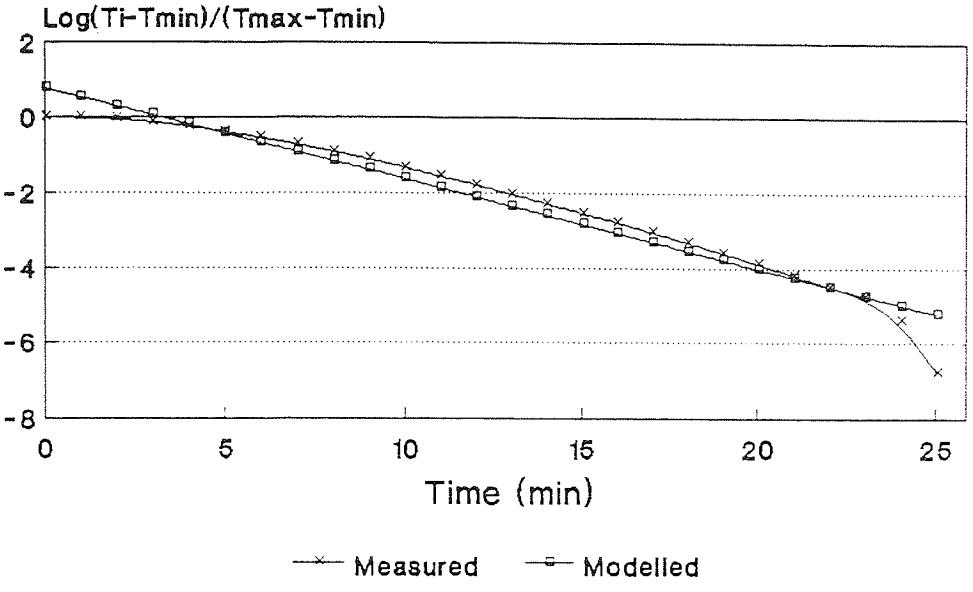
$G_h = 1/R_h$ = Heating conductance (W/K)

$G_c = 1/R_c$ = Cooling conductance (W/K)

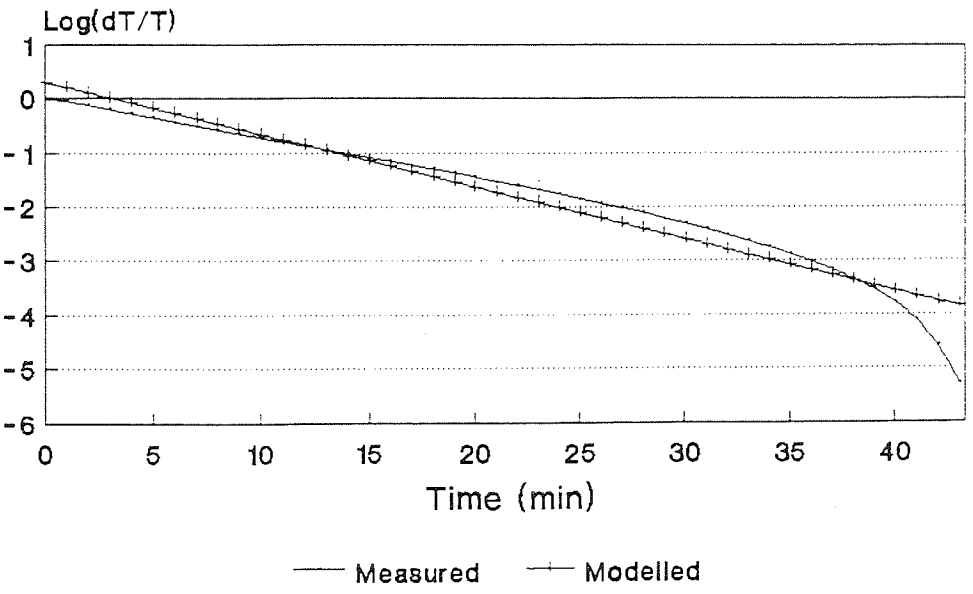
The above approximate equality implies that $R_c \approx R_h$, over this temperature range. Later on it will be shown that this is not always the case. It is on these values of time constants that the heater transfer functions were based in the dynamic room modelling presented in section 4.2 and 4.3.

Another important aspect of the modelling is the value of the Pearson cross-moment correlation (R). This is consistently above 95% for the heating part of the cycle, and above 90% for the cooling. A plot of the measured against the fitted function is shown in Fig.6.5. for Section B5 thermocouple 1. This plot shows that the gradient is only a least-squares-fit-average over the temperature range, and that the gradient of the experimental curve does in fact vary appreciably over the temperature range.

Fig 6.5: IRH_2
Section B5 Heating



Section B5 Cooling



6.2 The Optical Method

Remote sensing by means of infra-red imaging techniques provides an effective and efficient method of gathering data from the temperature fields produced on the surfaces of radiant tubes. One drawback is the dependence of the calculated temperatures on the value chosen for the emissivity of the surface. There is, however, a method for measuring emissivities[2] using images of surfaces of known temperature which make this problem less significant.

The aim of this experiment was to back up the findings of the temperature caliper experiment and to improve upon the spatial resolution of those readings.

6.2.1 Equipment

The AGA Thermovision 782 thermal imaging system and supporting hardware were loaned for the duration of the experiment from the Midlands' Research Station. A block diagram of the components of the system, which includes a brief statement of the function of each unit, is shown in Fig.6.6, whilst a schematic of the camera itself is shown in Fig.6.7. The salient features of the system are:

1) the wavelength range detected is 2-6 μm;

2) the detector is cooled with liquid nitrogen;

3) it has the ability to work over a variety of thermal levels and ranges;

4) it records data onto a video cassette recorder which allows the possibility of off-line processing of data as well as real-time image-capture;

Fig.6.6: Agema Thermovision System

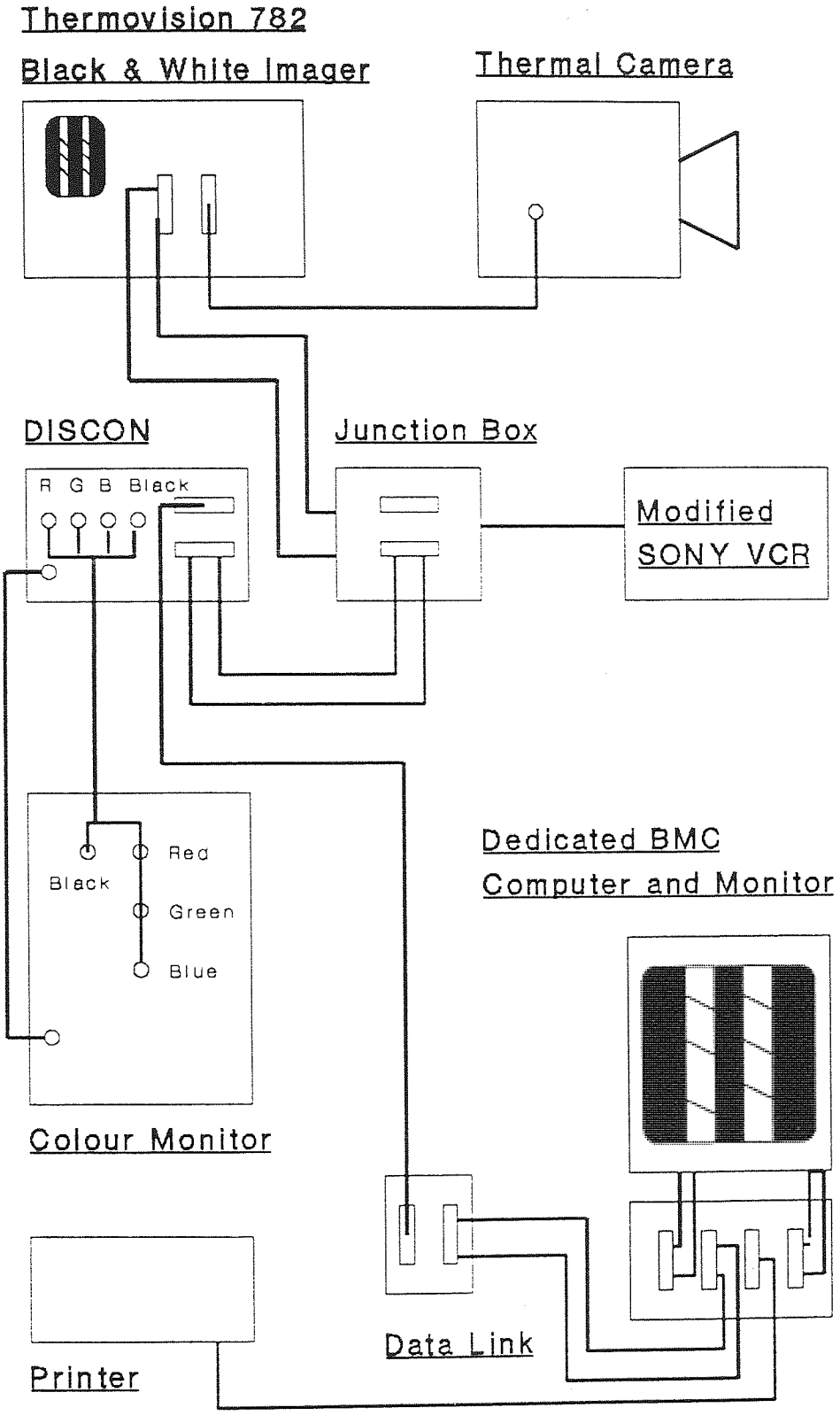
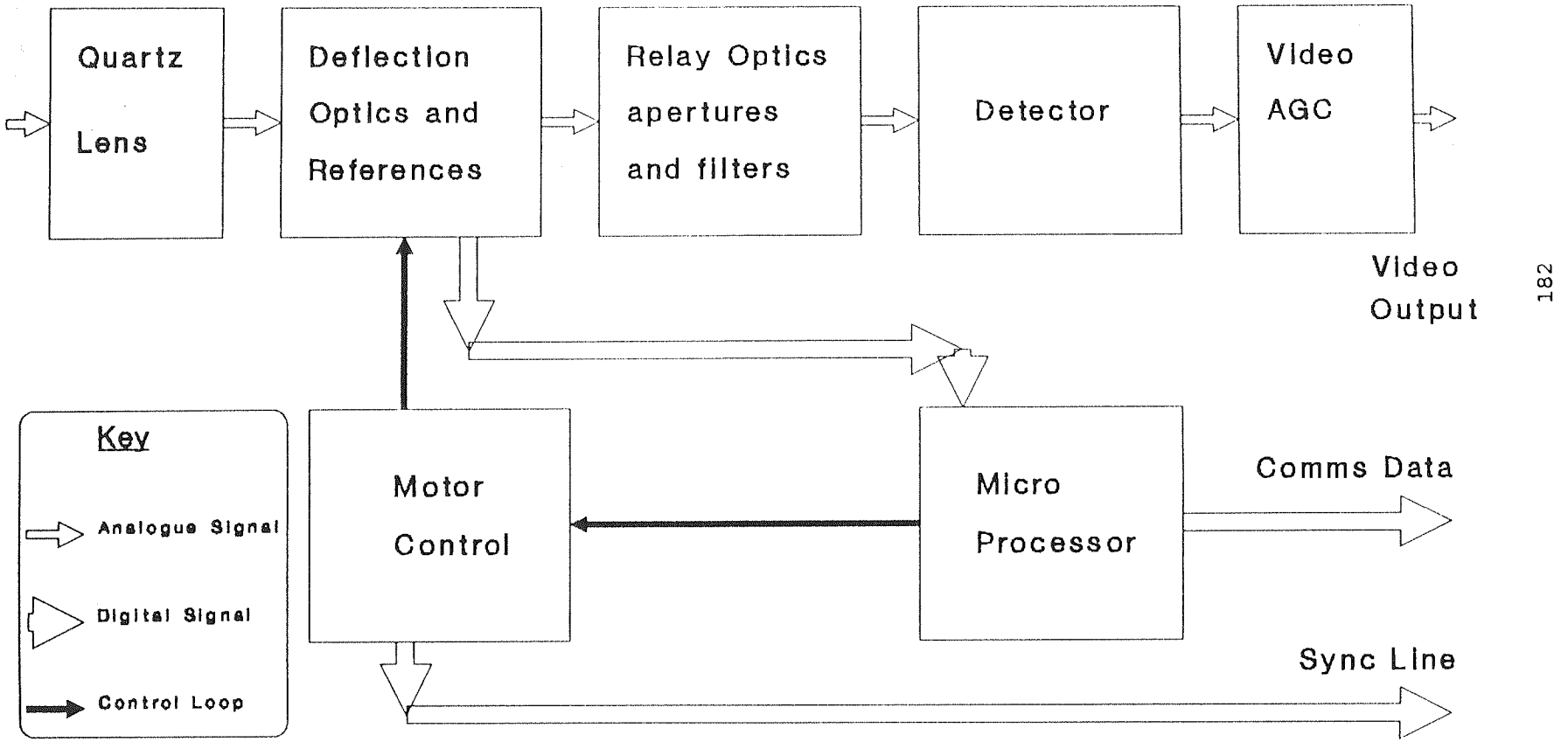


Fig.6.7: Thermal Imaging Camera



5) it is linked with a dedicated image processing computer which captures the desired images to disk;

6) the disk files produced have a resolution of 64 by 128 pixels, with 256 gray-scales; these files were unfortunately incompatible with any major operating system and had to be specially converted to MS-DOS format with a utility program supplied by AGEMA.

7) A consequence of point six is that the spatial resolution in the experiments described below was calculated to be ≈ 15mm; it is in general a function of the object's distance and the lens used.

8) the lenses were made of quartz and had the optical properties shown in Table 6.4.

Table 6.4: Infra-Red Lens' Properties

Lens	distance(m)	0.3	0.5	0.8	1.2	2.	3.	5.	10.
20°	width	-	0.13	0.22	0.37	0.68	1.0	1.7	3.5
	+	-	0.05	0.16	0.45	1.9	12.1	∞	∞
	-	-	0.04	0.11	0.25	0.6	1.3	2.8	7.1
40°	width	0.17	0.34	0.59	0.82	1.6	2.4	4.0	8.2
	+	0.05	0.29	1.9	∞	∞	∞	∞	∞
	-	0.03	0.12	0.6	0.6	1.3	2.2	4.1	9.0

Lens	Minimum Focus	Focal Length	Geometric Resolution
	(m)	(mm)	(mRad)
20°	0.5	38	3.9
40°	0.3	19	7.8

Note: "width" is the width of the scanned field; "+" and "-" indicate the limits of the focus, in metres, at a given distance.

6.2.2 Method

Three heaters were imaged using the AGEMA system. Two of these, IRH_6 and IRH_7, were installed in an engineering workshop, at a

height of about 5m; the other, IRH_1, was mounted on a test rig in a pilot plant laboratory at about 2m above the floor.

Having set up the equipment according to the operating instructions (MRS, 1989) and having positioned the camera relative to the heater as shown in Fig.6.8, the infra-red images of the heaters during a heating and cooling cycle were recorded by the VCR for later analysis. As the heaters were all too long to fit into the field of view, the thermographs of all of the heaters were taken in sections. The convention used to specify the sections was that the U-tube end was called "A" and then each subsequent letter was used to denote the next area moving in the direction of the burner. The recording of the images for all of the heaters followed the same protocol which was:

1) Position the heater beneath the section to be imaged.

2) With the heater running and in steady-state adjust the focus, aperture, and thermal level to give a clear image of the tubes with as high a contrast as possible. The thermal range was always set to as large a value as possible, usually 1000 isothermal units: this was so as to keep an image on the screen for as long as possible as the tube cooled.

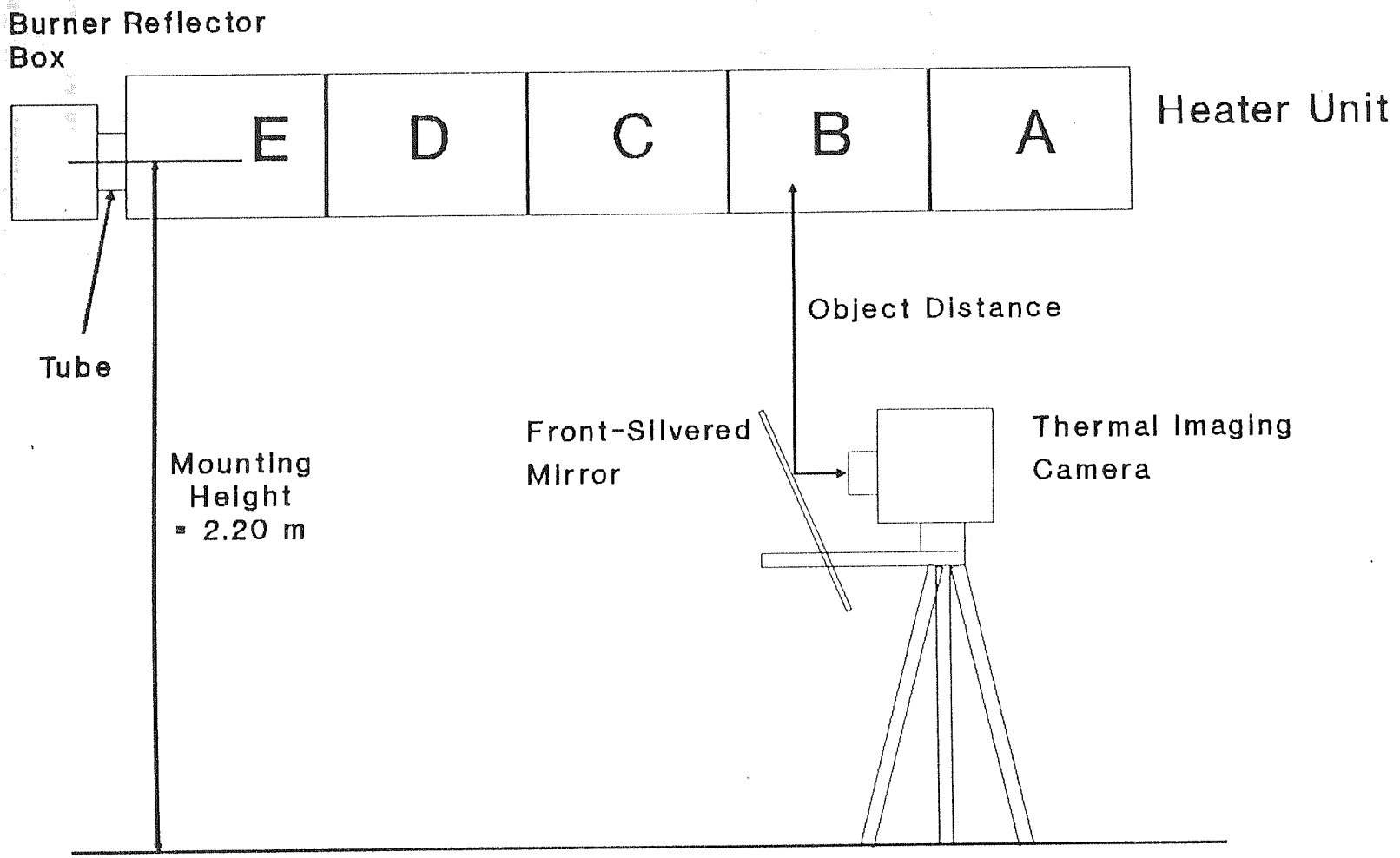
3) Note the wet and dry bulb thermometer readings.

4) Start the VCR, noting the time and VCR counter reading.

5) After 15 seconds, switch off the electrical power to the heater.

6) Allow the heater to cool for 6:00 minutes, at the end of which time switch the power to the heater back on.

Fig.6.8: Configuration of Thermal Imaging Equipment



7) Let the heater warm up for 13:00 minutes, at the end of which time stop the VCR, again noting the VCR counter reading, and allow the heater to heat up towards steady state for the start of the next reading-cycle.

8) Ensure that the heater is at the same temperature as at the start of the cycle before recording at the next position. This heating typically requires about 10 minutes.

9) Move the equipment along the tube to the next measuring position.

The only variation was that the length of time that IRH_7 was allowed to cool down was longer as it had a longer time constant. Appendix 3 contains the full details of the tapes that were recorded and the files that were subsequently captured from the recordings, whereas Table 6.5 is a summary of the essential operating information.

The reason for the two imaging runs with IRH_1 and IRH_6 is that at the burner end of the heaters the temperature differential between the tubes was too large to be able to record accurate information for both tubes at the same time. Thus one run was taken with the aperture, level, and range set to capture the profile of the hot tube, and the second to capture that of the cool tube.

Table 6.5: Thermography Details

Subject	Section	Lens (°)	Aperture	Distance (m)	Filter	Relative Humidity (%)	Ambient Temperature (°C)
IRH_6	A	20	f2.5	3.38	NOF	-	-
IRH_6	B hot	20	f2.5	3.38	NOF	-	-
IRH_6	B cool	20	f1.8	3.38	NOF	-	-
IRH_6	C hot	20	f5.1	3.38	NOF	-	20.1
IRH_6	C cool	20	f1.8	3.38	NOF	-	-
IRH_6	D hot	20	f5.1	3.38	NOF	-	20.2
IRH_6	D cool	20	f2.5	3.38	NOF	-	20.7
IRH_1	A	40	f5.1	1.42	NOF	47.7	18.6
IRH_1	B	40	f3.6	1.42	NOF	37.5	20.2
IRH_1	C	40	f5.1	1.42	NOF	52.8	17.1
IRH_1	D hot	40	f7.2	1.42	NOF	50.9	17.8
IRH_1	D cool	40	f2.5	1.42	NOF	47.7	18.1
IRH_1	E hot	40	f10.0	1.42	NOF	50.1	17.6
IRH_1	E cool	40	f2.5	1.42	NOF	46.9	18.6
IRH_7	A	20	f7.2	3.38	NOF	-	16.7
IRH_7	B	20	f5.1	3.38	NOF	-	-
IRH_7	C	20	f2.5	3.38	NOF	-	17.9
IRH_7	D	20	f1.8	3.38	NOF	-	16.6

From each of the above runs 38 images were captured to disk at 30 second intervals for IRH_1 and IRH_6, whilst for IRH_7 only 23 files, taken one every minute, were saved to disk. This resulted in 650 files in TTX format which were then converted using the AGEMA utility program *AIMS* into MS-DOS format binary files. These were then transferred using the Public Domain communication package Kermit to the Aston Vax for further processing. This package has a "binary" communications mode which transfers files exactly as they are without trying to interpret their contents in any way, which was essential for this task.

6.2.3 Analysis of Results

A Fortran program (*CONVERT.FOR*) had been developed at the Midlands Research Station which reads in the binary data and converts the pixel values to temperatures. This was achieved using an algorithm from the manufacturer's manual, in conjunction with information concerning the lens and atmospheric absorption. This was all encapsulated into a subroutine called *DEGREE*. The program was used originally in an experiment to measure convective heat transfer from a heated plate and consequently required modification to be of use. Two programs resulted from *CONVERT*: the first, *IMAGE.FOR* (Listing 6.1), uses the UNIRAS graphics library to produce colour images of the tubes with the colours corresponding to the isothermal regions; the second, *TIME.FOR* (Listing 6.2), produced values of the mean temperature (T_m), the fourth root mean temperature (T_{m4}), the maximum temperature (T_{max}), and $\sigma.\epsilon.T_{m4}^4$. T_{m4} is defined to be:

$$T_{m4} = (\Sigma\ T_{ij}^4/N)^{0.25} \qquad (6.7)$$

where:

T_{ij} = the temperature of the i,j th pixel (K)

N = number of pixels

Σ = two dimensional summation over all pixels

It becomes clear why this quantity is of interest if a radiating surface divided into N equal areas is considered. The total power radiated is:

$$P = \sigma.\epsilon.A.(T_{m4}^4 - T_a^4) \qquad (6.8)$$

where σ = Stefan-Boltzmann constant ($Wm^{-2}K^{-4}$)

ϵ = emissivity

A = total surface area (m^{-2})

T_{m4} = fourth-root mean temperature (K)

T_a = surrounding M.R.T. (K)

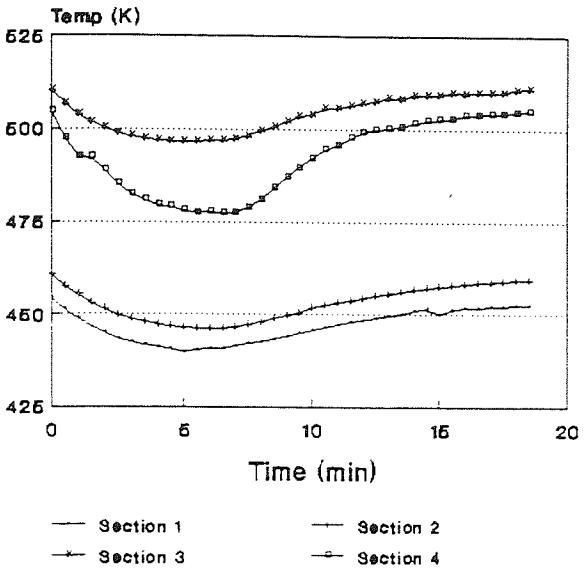
It is also evident why the program computes $\sigma.\epsilon.A.T_{m4}^4$ which has the same dimensions as irradiance (Wm^{-2}) and is a normalised value of the power produced by the source. These values are plotted for individual sections, Fig.6.9, and with the individual data agglomerated into values for the whole tube, as in Fig.6.10 where the values are plotted against time for the three heaters tested.

There are two important aspects of these plots. Firstly, the time constants for heating and cooling can be calculated from the values of T_{m4}, and secondly, that the time-dependence of the total power output of the heater can be estimated.

Table 6.6: Time Constants from Thermography

Heater	T max (K)	T min (K)	Cooling A	Cooling τ (s)	Heating A	Heating τ (s)
IRH_1 Section 1	574	534	0.97	206.	0.95	230.
Section 2	548	500	1.04	226.	0.95	232.
Section 3	573	530	0.99	219.	0.99	241.
Section 4	662	632	0.88	267.	0.85	215.
Section 5	624	550	0.87	312.	0.73	169.
IRH_6 Section 1	454	433	0.93	267.	0.82	266.
Section 2	461	440	0.86	286.	0.86	245.
Section 3	510	487	0.85	371.	0.75	235.
Section 4	505	465	0.90	271.	0.95	209.
IRH_7 Section 3	464	451	0.95	333.	1.00	494.
Section 4	429	413	0.93	402.	0.92	374.

Fig.6.9: IRH_6
Sectional Temperatures from Thermography



Sectional Irradiances

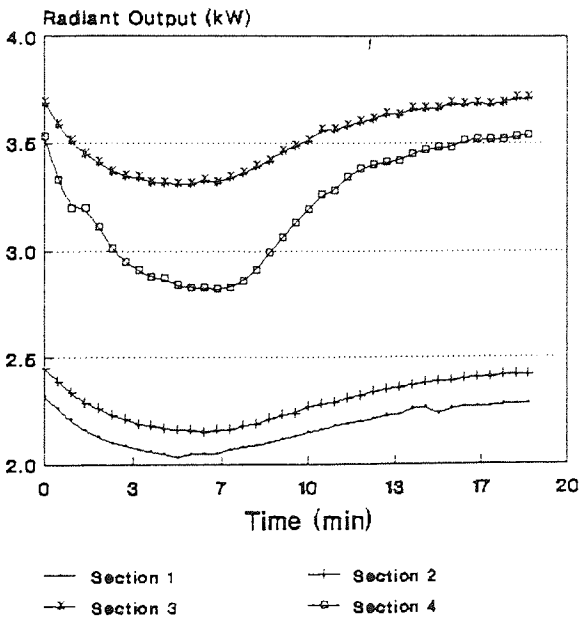
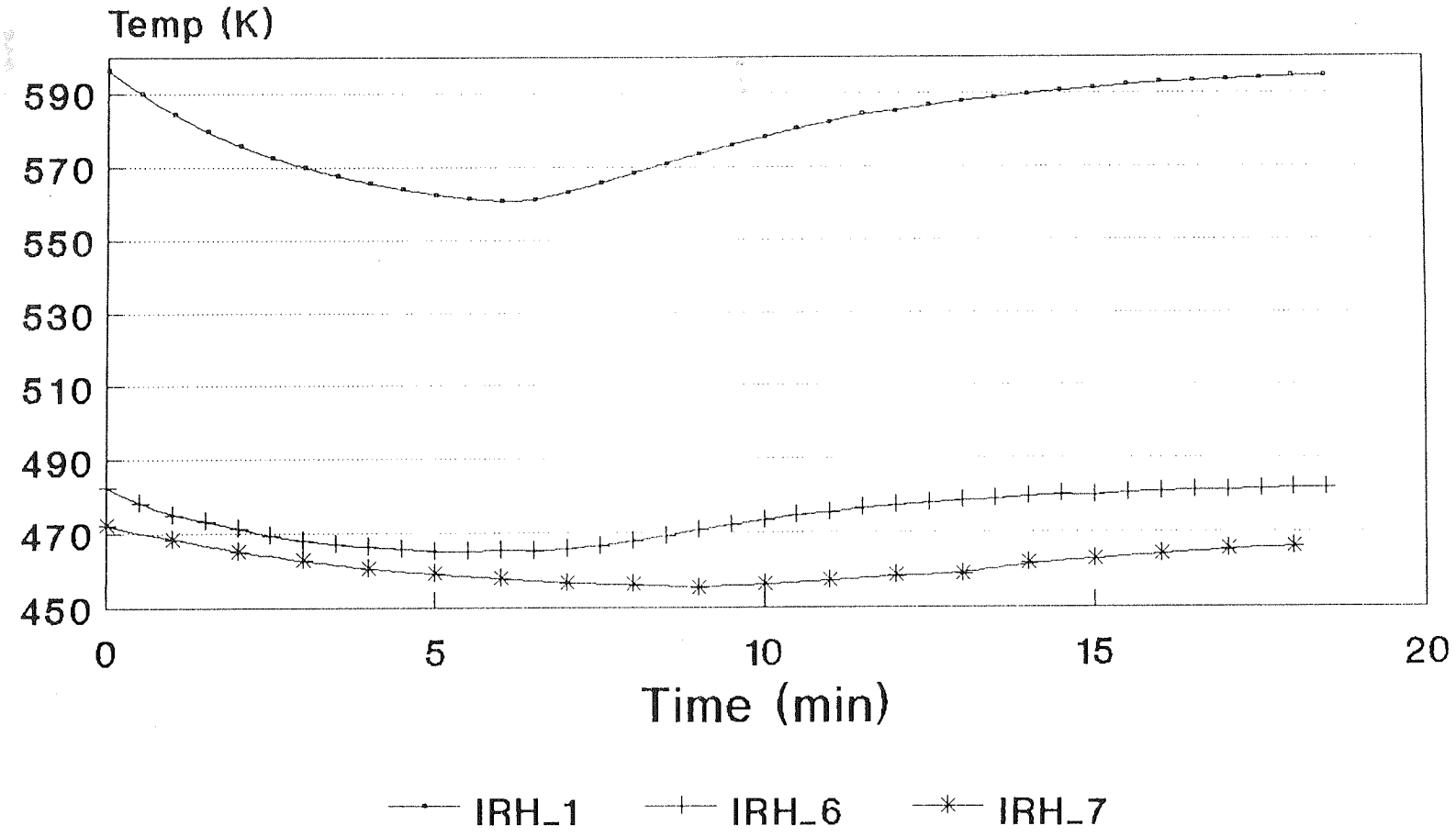


Fig.6.10: Comparison of Heaters

Mean 4th Root Temperature vs Time

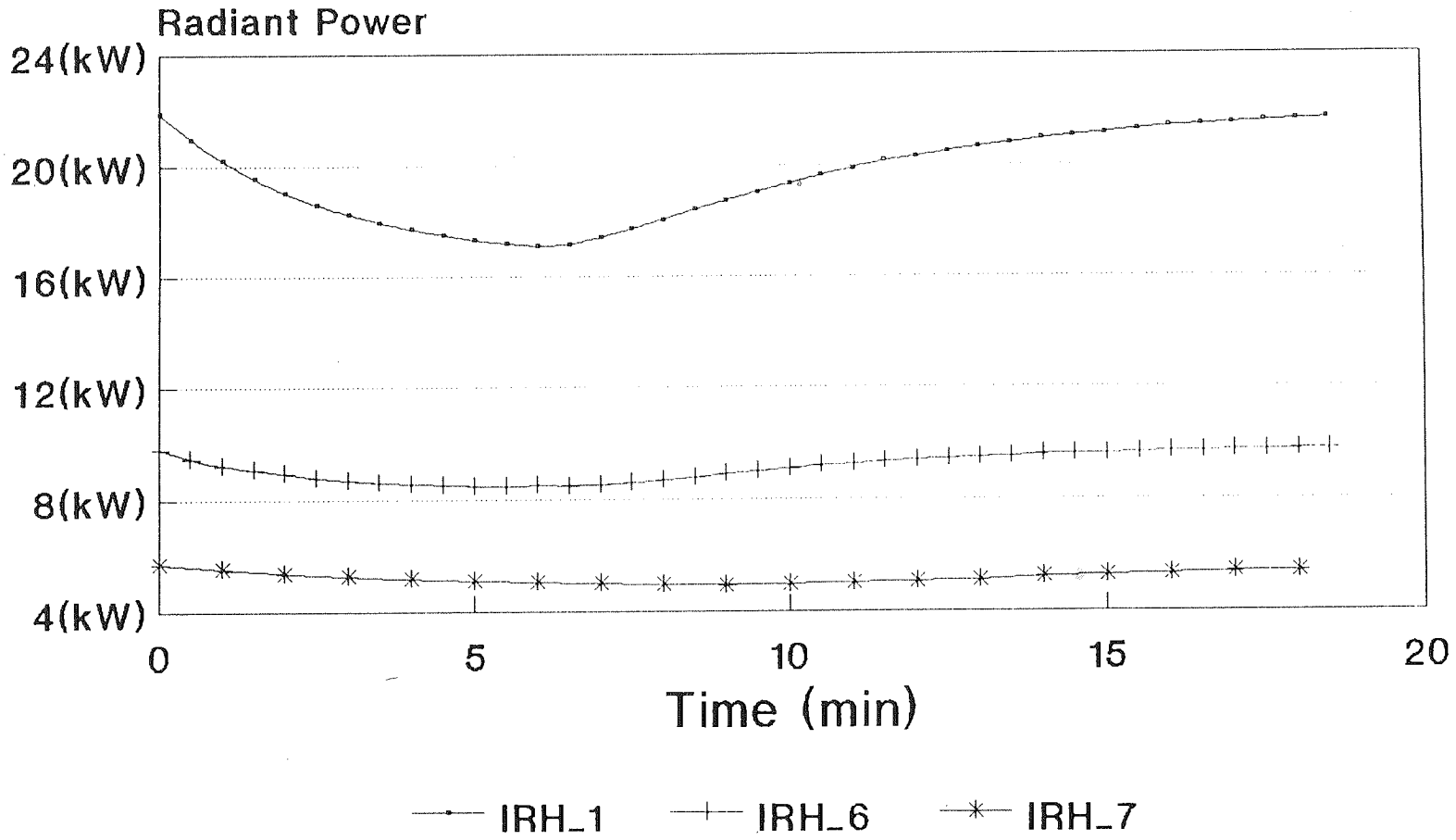


There are two striking features in the above table, firstly that the time constants for the heating part of the cycle are in good agreement with the values derived using the contact method, and secondly that $\tau_c \approx \tau_h$. At first sight this appears to be in contradiction to the findings of section 6.1.2. In fact the issue arises because the optical analysis is only covering part of the temperature operating range of the heater. The values of the HTC that determine the time constants are a function of temperature. In general they will be higher for the cooling process at higher temperatures; this in turn will lead to shorter time constants if only the higher range of temperatures is sampled. The figures in table 6.3 are calculated across the whole range of temperatures, and therefore represent good estimates of τ_{avg}, whilst the values obtained from the thermographic images are the average values of only the top 200 K of the operating range and will therefore have time constants different from the average in the manner described above. The values used in the dynamic modelling were those generated from the contact method, as they represent the best average figure over the whole range.

The graphs of total power versus time, calculated by using the Stefan-Boltzmann relation on the 40,000 imaged segments of each of the three tubes, are shown in Fig.6.11. It demonstrates clearly that the most effective heater with the smallest time constant is IRH_1, followed by IRH_6, and then IRH_7.

Fig.6.11: Comparison of Output

Radiant Power vs Time



6.3 Conclusions

1) The time-dependent thermal behaviour of a radiant heater can satisfactorily be summarised by two constants, τ and A, for most modelling work. If greater accuracy is required then four constants, two each for the heating regime and two for the cooling, should be used.

2) An accurate assessment of the experimental results shows that in fact both τ_c and τ_h are temperature dependent, and the selection of average values over the operating range of interest would yield better results.

3) The time constants generally depend upon the construction of the individual heaters, specifically the type of materials and tube thickness, and on the reflector shape and thermal characteristics.

4) The fourth-root mean-temperature proves to be a reliable indicator of radiant behaviour, both static and dynamic.

Footnotes to Chapter 6

1: This simple model can not be modelled using the package (CTRL-C) chosen, for the reason that in the circuit shown in Fig.6.4 in order to arrive at the correct behaviour when the voltage source is switched off it must have infinite resistance to prevent the charge from returning whence it came. In the CTRL-C implementation when a circuit of this sort is modelled a voltage source has zero resistance when off, thus rendering this simple circuit untenable.

2: There was a facility on the Thermovision 782 black and white monitor to display two isotherms on the screen which could be adjusted to any thermal level desired. The technique for measuring emissivities was to use this facility to surround the hottest area being imaged and then to record the IU (isothermal units) of this isotherm. The emissivities used in this experiment were calculated from this value and from a contact measurement, performed in this case with a thermocouple.

7 Final Perspectives

7.1 Overview

The author has sought to explore four main areas concerning the use of radiant heating to provide thermal comfort in industrial applications. These areas are described below.

1) The Monte Carlo approach to the numerical integration of the flux produced by a radiant heater. This approach based upon the principles of stochastic ray-tracing was successfully applied to the problem of predicting both the total flux produced by the heater and its distribution across the zone of occupation. It resulted in a suite of Fortran programs which predicted key quantities of interest to assessments of thermal comfort, namely the Mean Radiant Temperature, the Predicted Mean Vote, and the Predicted Percentage of Dissatisfied.

2) The application of the electrical analogy of heat flow firstly to static and subsequently to transient heat flows produced by radiant heating systems. By taking the results of the Monte Carlo modelling along with some of the experimental work as the starting point, this analogy was used to make predictions concerning both the sizing and control of radiant systems.

3) The design and calibration of a system for measuring the irradiance produced by an extended radiant source, which involved a novel approach to radiometry. This area of work also included the measurement of smaller, point-like sources with a conventional radiometer.

4) The use of thermocouples and thermographic imaging equipment to produce information about the temperature distribution

produced along and around the tube. This study was performed both at steady state and whilst the tube was heating and also cooling.

7.2 Conclusions

Having carried out the above research programme, the important conclusions from each of these areas can be summarised as follows:

7.2.1 Monte Carlo Modelling

1) This proved to be an efficient and potentially accurate method of predicting system performance. It also proved extremely flexible to new demands made on the program which arose as the answers to the original questions were found. It is an easily comprehensible approach and almost certainly the best method available for attacking the particular problem of modelling industrial radiant heaters.

2) As computer hardware improves and especially as the software techniques for exploiting parallel processing mature over the next five years, ray-tracing will become an extremely attractive option for engineers wishing to simulate radiation heat transfer.

7.2.2 The Electrical Analogy

1) This offers a very fruitful *modus operandi* when a designer is confronted with highly involved heat interchange problems. Also, the state space matrix formulation enabled the problem to be greatly simplified and for the solutions to be produced almost mechanically.

2) The results of the sizing simulation showed that whilst the CIBSE Guide is acceptable as a first approximation, its lack of detail and direction on matters such as mounting height and placement of heaters is best calculated using the electrical analogy in conjunction with the Monte Carlo model.

3) In the section on the control of radiant systems, some

shortcomings of conventional practice were also highlighted. It was found that the most important determinant of system performance is the control law applied to the system. Of no less significance is the fact that the only way to fully assess the performance of any control system is by computing a weighted sum of both the error and control effort components to produce a performance index.

7.2.3 Radiometry

1) Radiometry provides an appropriate and accurate method of measuring the irradiance produced by radiant heaters.

2) Radiometry has decisive advantages over each of the two competing methods: firstly, thermopiles and thermocouples have shorter intrinsic time constants than black globe thermometers which allows non-steady state readings to be taken; secondly, radiometers provide detailed information of irradiances over the "occupied-zone" beneath the heater allowing assessments of thermal comfort to be made, whereas the calorimetric technique produces only one figure of merit, the efficiency of the heater.

3) There was substantial agreement between the results of the radiometric experiments and the Monte Carlo model, within the bounds of experimental error.

4) The efficiencies of the various heaters measured were found to be critically dependent on the balance between the radiative and convective heat transfer attained by a given heater.

7.2.4 Thermometry

1) The time-dependent thermal behaviour of radiant heaters depends primarily upon the thermal weight of the materials from which they are constructed and, to a lesser extent, the manner in which they are constructed.

2) A light-weight heater has a time constant ≈ 600 s, whilst a heavy-weight will have a time constant ≈ 1200 s.

3) The time constants for heating and cooling are different by about a factor of two, heating being the slower process.

4) Analysis of the results shows that the representaion of the heaters as a system with one time constant is an acceptable approximation in the first instance.

7.3 Future Work

7.3.1 Modelling

The most apparent shortcomings of the suite of programs produced are the lack of modularity and the difficulty of use. In order to address these problems the whole suite could be re-written in a form more amenable to addition of new detector and reflector geometries. Such a scheme is outlined in a monograph on ray-tracing which also contains many useful algorithms and ideas for improving the usability of the programs (Glassner, 1989). The particular approach advocated is known as Object Oriented Programming (OOP) and its implementation in this case would facilitate modifications. There is now a wide range of OOP languages available commercially; Zortech C++, Turbo C++, Turbo Pascal v5.0 to name three that run under MS-DOS. This raises another issue: as it is currently written the model will run only on a VAX mainframe, whilst, due to the ever-improving cost-performance ratio, the most commonly available equipment is the PC or PC-compatible. A version of the program to run on a PC would make it more widely accessible, as well as creating the possibility of a more usable graphical interface.

Concerning the issue of speed, the recent advent of cheap parallel processing hardware in the form of add-on transputer boards for PCs has introduced the possibility of performances on a par with, or maybe even surpassing, mainframes. Ray-tracing is a particularly propitious application for parallel processing since it has the potential for two possible parallelizations. The first is due to the fact that the program deals with vector algebra in three dimensions:

this means that if each of the three dimensions is assigned a processor the computation will proceed much more quickly. The second possibility is that as each ray is independent of all the other rays, processing time could be reduced indefinitely by addition of more processors, the limit only occurring when there is one processor computing the trajectory of a single ray.

There are a number of possible refinements that could be made to the physical modelling. Firstly, it should be noted that the reflection of the rays from both the tube and the reflectors is calculated as a specular reflection, whereas a more physically realistic assumption would be to consider a partly specular, partly diffuse reflection. Secondly, atmospheric absorption is currently ignored. A simple calculation based on a window-model of absorption suggests that for a tubular heater the absorption ≈ 8-10%. The addition of a subroutine to calculate absorption would be reasonably straightforward and it would have the benefit of providing a good estimate of the differences in atmospheric attenuation between gas-fired heaters and QLLs. At the moment the QLL heaters are modelled by assuming the filament to be a cylinder with an operating temperature ≈ 2700 K; a better model would be to consider it as two concentric cylinders, the inner one at 2700K and the outer transparent one at 600K. This would simulate the effect of the glass sheath which contributes ≈ 20% of the radiant output of a QLL.

The effects of convective heat transfer have been by-passed in the modelling procedure by basing the temperature field of the tube on experimental values. It would be a significant advance if work were to

be carried out on the effect of reflector geometry on the convective heat transfer coefficients. This could be carried out using a commercial computational fluid dynamics package, for example PHOENIX which is available on the Aston VAX.

Lastly, there is currently only a planar geometry available for use as a detector. Having the flexibility to chose between two or three alternative geometries, hemispherical or bath-shaped for example, would make results from the model easier to compare with experimental data. This would be particularly true of the plaque and QLL heaters for which there is a British Standard of measurement based on a hemisphere (British Standards Institute, 1962). Likewise with the reflector, the only surfaces described by the model are rectangular sheets. For gas-fired heaters this is sufficient to represent nearly all commercial heaters, but a lot of QLL heaters are designed with a parabolic reflector. This could be modelled by implementing a quartic surface intersection algorithm (Hanrahan, 1989) to yield a more accurate representation of the heater.

7.3.2 Instrumental Work

As it stands the Automated Aston Radiometer is a great improvement upon the original version. There are several problems associated with it which give scope for improvements to its design. These problems fall into two categories: the ease with which the radiometer can be employed, and its accuracy.

One major problem is the number of cables which must be run to each radiometer, which makes the use of more the two or three at a time very cumbersome. Two improvements could be made to the device which

would render it much easier to use. Firstly, the power cables could be replaced by *in situ* batteries which would mean that only the signal line would have to be trailed to the data-logger. To take the process of reduction of cabling to its logical conclusion a telemetric link could be designed, using either infra-red, ultrasonic, or radio communications. There are standard integrated circuits available which are used in the remote control of domestic appliances that could be used for this purpose.

Another important issue which lies somewhere between the two problem areas outlined above is signal noise and signal compatibility. The device is currently extremely unintelligent; it merely sends back a voltage leaving the logging computer to do all the computation necessary. This method means that small voltages, ≈mV, are being carried over 10 m runs of cable often through electromagnetically noisy environments. The solutions suggested above could also incorporate some on-board analogue to digital conversion and even some numerical preprocessing, so that the radiometer could send data in a digital, rather than the more corruptible analogue, form. For the sake of enhanced compatibility it would be extremely useful if the radiometer could conform to the IEEE-488, or similar, communications protocol. This would enable the use of a wide range of commercial hardware and software running on IBM-compatible machines to access the radiometer.

Concerning the issue of accuracy, the simple calculation of irradiance from a measurement of an initial gradient displayed a disappointing lack of consistency. A potentially more accurate solution to the problem is to use a method of detection which responds

to a change in irradiance, for example a pyroelectric detector, in conjunction with a radiation modulator, a rotating toothed wheel for example, and a FET amplifier. This would produce an AC output proportional to the irradiance impinging on the device. Initial experiments with a prototype showed promising results with sensitivities of ≈ 20 mV/W m^{-2}, at 14 Hz, being recorded. This device would not suffer the same zero-drift problems as the Aston Radiometer, and also the amplification and signal processing involved with an AC, in contrast with a DC, signal is much more straightforward.

7.3.3 Experimental Work

1) A full analysis of the results of the thermal imaging work should be undertaken. The library of images contains a wealth of untapped information about radiant heaters. The analysis should include work on: the dependence of temperature on radial position; the effect of the U-bend on heat-transfer coefficients within the tube; a study of the mutual tubular heating, an effect which is very apparent on the thermographs; the effect of reflector geometry on radiation heat transfer. Other effects such as the use of "turbulators", helical corrugated metal strips placed within the tubes, the determination of reflector temperature fields, or the effect of recuperating or range rating the heaters could be undertaken in a new thermographic study.

2) Monitoring of installations of a variety of sizes should be undertaken to assess the validity of both the Monte Carlo model and the electrical analogue.

3) A data-base of complete physical characteristics of as wide

a range of heaters as possible should be compiled. As well as the measurements discussed in the section concerned with measuring the output of IRH_7 (5.1.2) it would also be instructive to measure the levels of NO_x and SO_x produced, especially from ceramic heaters.

APPENDIX A: Annotated Program Listings

Chapter 2

The program modules shown in Fig.2.4 were collected into two libraries. The file BIRH.TLB contained the source code modules, and the file ZIRH.OLB contained the object module. The source code below represents the latest version of the program which is different in some respects to that installed on the MRS VAX mainframe. The principal difference between the versions being that the MRS version has two data-editors written specifically to enable ease of manuipulation of the data files.

```
C     ********* ********* ********* ********* ********* ********* ******
C      *****PROGRAM TO MODEL IRH UNITS*****
C      *****C. W.BUTLER 1982        *****
C     ***** Alteration version 1.1 ******
C     ***** This version is one step closer to the MRS edition of the
C     ***** because of the following differences from v1.0:
C     ***** 1) Implicit None
C     ***** 2) Two data i/p files
C     ***** 3) Machine dependent constant definitions
C     ***** 4) More consistent use of common blocks for removing
C     *****   redundant parameter passing
C     ***** CDZ 29/5/90
C     ********* ********* ********* ********* ********* ********* ******

      IMPLICIT NONE

C     ********* ********* ********* ********* ********* ********* ******
C     *** Variables from Tube data file
C     ********* ********* ********* ********* ********* ********* ******

      INTEGER NSURF, NT
      REAL*8 SURF(40,3,3), XD(0:100), TT(0:100), TB(0:100)
      REAL*8 X, Y, RAD, EPS1, EPS2

C     ********* ********* ********* ********* ********* ********* ******
C     *** Variables from Detector data file
C     ********* ********* ********* ********* ********* ********* ******

      INTEGER NRAYS, NX, NY
      REAL*8 XLO, XHI, YLO, YHI, DETLEN, DETWID
      REAL*8 AXIS(3), FRONT(3)
      CHARACTER*6 DETTYP

C     ********* ********* ********* ********* ********* ********* ******
C     *** File name variables
C     ********* ********* ********* ********* ********* ********* ******

      CHARACTER*40 FDET, FTUB, FOUT
```

```
C     ********* ********* ********* ********* ********* ********* ******
C     *** Variables from Main body of Program
C     ********* ********* ********* ********* ********* ********* ******

      INTEGER HIT(0:100, 0:100)
      INTEGER NS, IC, NPER, I, J, K, KEYB, VDU

      REAL*4 IRRAD(0:100, 0:100)
      REAL*8 RAY(3), POINT(3), ANORM(3)
      REAL*8 EX(3), EY(3), EZ(3)

      REAL*8 EMISS, TOTFLX, DUM, FLUX, PROP, TEMP, DIST, FP
      REAL*8 PI, SIGMA0

C     ********* ********* ********* ********* ********* ********* ******
C     *** Common Blocks
C     ********* ********* ********* ********* ********* ********* ******

      COMMON /tube/ X, Y, RAD, EPS1, EPS2
      COMMON /reflect/  NSURF, SURF
      COMMON /temp/ NT, XD, TT, TB
      COMMON /detect/ NX, NY, XLO, YLO, DETLEN, DETWID
      COMMON /unitv/ EX, EY, EZ
      COMMON /array/ IRRAD, HIT
      COMMON /const/ PI, SIGMA0

C     ********* ********* ********* ********* ********* ********* ******
C     *** Set up unit vectors
C     ********* ********* ********* ********* ********* ********* ******

      DATA (EX(I),EY(I),EZ(I),I=1,3)/1.0,3*0.0,1.0,3*0.0,1.0/

C     ********* ********* ********* ********* ********* ********* ******
C     *** Define Constants
C     ********* ********* ********* ********* ********* ********* ******

      SIGMA0 = 5.729D-8
      PI     = 4. * DATAN(1.D0)
      KEYB   = 5
      VDU    = 6

C     ********* ********* ********* ********* ********* ********* ******
C     *** Initialize count variable
C     ********* ********* ********* ********* ********* ********* ******

      NPER   = 0

C     ********* ********* ********* ********* ********* ********* ******
C     *** Read Data in from specified files
C     ********* ********* ********* ********* ********* ********* ******
```

```
      WRITE (VDU,*) 'Enter detector file name (with extension)'
      READ (KEYB, '(A)') FDET
      FDET = '[ZIESLERCD.FORT]'//FDET

      WRITE (VDU,*) 'Enter tube file name (with extension)'
      READ (KEYB, '(A)') FTUB
      FTUB = '[ZIESLERCD.FORT]'//FTUB

      WRITE (VDU,*) 'Enter Output file name (with extension)'
      READ (KEYB, '(A)') FOUT
      FOUT = '[ZIESLERCD.FORT]'//FOUT

      OPEN (UNIT=10, FILE=FTUB, STATUS='OLD')
      READ (10,*) NSURF
      READ (10,*) X, Y, RAD, EPS1, EPS2
      READ (10,*) (((SURF(I,J,K),K=1,3),J=1,3),I=1,NSURF)
      READ (10,*) NT
      READ (10,*) (XD(I),I=1,NT)
      READ (10,*) (TT(I),I=1,NT)
      READ (10,*) (TB(I),I=1,NT)
      CLOSE (UNIT=10)

      NSURF = NSURF + 1
      OPEN (UNIT=10, FILE=FDET, STATUS='OLD')
      READ (10, '(A)') DETTYP
      READ (10, *) NRAYS
      READ (10, *) ((SURF(NSURF, J, K), K=1,3), J=1,3)
      READ (10, *) NX, NY
      IF (DETTYP.NE.'PLANAR') THEN
         READ (10,*) (AXIS(I), I=1,3)
         READ (10,*) (FRONT(I), I=1,3)
      ENDIF
      CLOSE (UNIT=10)

C     ********* ********* ********* ********* ********* ********* ******
C     *** Initialize variables
C     ********* ********* ********* ********* ********* ********* ******
      TOTFLX = 0.0D0
      DUM    = 0.0D0
      IC     = 1

      XLO    = SURF(NSURF, 1, 1)
      YLO    = SURF(NSURF, 1, 2)
      XHI    = XLO
      YHI    = YLO

      DO 111 I= 2,3
         IF (SURF(NSURF,I,1).LT.XLO) XLO=SURF(NSURF,I,1)
         IF (SURF(NSURF,I,1).GT.XHI) XHI=SURF(NSURF,I,1)
         IF (SURF(NSURF,I,2).LT.YLO) YLO=SURF(NSURF,I,2)
         IF (SURF(NSURF,I,2).GT.YHI) YHI=SURF(NSURF,I,2)
 111  CONTINUE
```

```
      DETLEN = XHI - XLO
      DETWID = YHI - YLO

C     ********* ********* ********* ********* ********* ********* ******
C     *** Perform ray tracing over NRAYS
C     ********* ********* ********* ********* ********* ********* ******

      DO 1000 I= 1, NRAYS

C     ********* ********* ********* ********* ********* ********* ******
C     *** Define surface normal vector and calculate exit temperature
C     ********* ********* ********* ********* ********* ********* ******

        CALL INIT(POINT,ANORM,TEMP)

C     ********* ********* ********* ********* ********* ********* ******
C     *** Determine initial direction of ray, and calculate the flux
C     ***  which it is carrying
C     ********* ********* ********* ********* ********* ********* ******

        CALL DIRECT(ANORM, RAY)
        FLUX = SIGMA0*(TEMP**4)*EPS1*2.*PI*RAD*(2.*X+PI*Y)/NRAYS

C     ********* ********* ********* ********* ********* ********* ******
C     *** Define the initial strength of the ray
C     ********* ********* ********* ********* ********* ********* ******

        PROP = 1.
 65     NS = -9
        DIST = 1.E29

C     ********* ********* ********* ********* ********* ********* ******
C     *** Check for intersection with first cylinder
C     ********* ********* ********* ********* ********* ********* ******

        CALL INTERC(RAY, POINT, DIST, NS, 1)

C     ********* ********* ********* ********* ********* ********* ******
C     *** Check for intersection with second cylinder
C     ********* ********* ********* ********* ********* ********* ******

        CALL INTERC(RAY, POINT, DIST, NS, 2)

C     ********* ********* ********* ********* ********* ********* ******
C     *** Check for intersection with torus
C     ********* ********* ********* ********* ********* ********* ******

        CALL TORUS(RAY, POINT, DIST, NS)
```

```
C     ********* ********* ********* ********* ********* ********* ******
C     *** Check for intersection with any reflector or detector
C     ********* ********* ********* ********* ********* ********* ******

 666    CALL REFL(RAY, POINT, DIST, NS)

C     ********* ********* ********* ********* ********* ********* ******
C     *** No hit or low intensity ray
C     ********* ********* ********* ********* ********* ********* ******

        IF ((NS.EQ.-9).OR.(NS.EQ.NSURF)) GOTO 800
        IF (PROP.LT..001) GOTO 810

C     ********* ********* ********* ********* ********* ********* ******
C     *** Find intensity, starting position, and direction of reflected
C     ***  ray
C     ********* ********* ********* ********* ********* ********* ******

        IF (NS.LT.0) THEN
           EMISS=EPS1
        ELSEIF ((NS.GT.0).AND.(NS.LT.NSURF)) THEN
           EMISS=EPS2
        ENDIF
        PROP= PROP*(1-EMISS)
        CALL NRAY(RAY, POINT, ANORM, DIST, NS)
        GOTO 65

C     ********* ********* ********* ********* ********* ********* ******
C     *** Log the downward flux and also the grid position on the
C     ***  detector at which the ray intersected
C     ********* ********* ********* ********* ********* ********* ******

 800    IF (DETTYP.NE.'PLANAR') THEN
           CALL PROJ(AXIS, FRONT, RAY, FP)
           FLUX = FLUX*FP/1000.
        ENDIF

        IF (RAY(3).LT.0.) THEN
             TOTFLX = TOTFLX + PROP*FLUX
        ENDIF

        IF (NS.EQ.NSURF) THEN
             CALL DETOPT(POINT, RAY, DIST, PROP, FLUX)
        ENDIF

 810    IF (MOD(I,NRAYS/25).EQ.0) THEN
           NPER = NPER + 4
           WRITE (6,945) 'The run is ', NPER, ' % complete'
 945       FORMAT (1X,A11,I3,A11)
        ENDIF

 1000 CONTINUE
```

```
      CALL OUTPUT(FOUT)

      STOP
      END

C     ********* ********* ********* ********* ********* ********* ******
C     *** Miscellaneous Functions
C     ********* ********* ********* ********* ********* ********* ******


      FUNCTION FUNCT(R1,R2,PAR2)
      IMPLICIT NONE

      REAL*8 R1, R2, PAR2
      REAL*8 FUNCT

      REAL*8 PI
      REAL*8 PA, EST1, EST2, CHI, DUM

      PI    = 4.*DATAN(1.0D0)
      PA    =PAR2 - R1/R2
      EST1  = 0.
      EST2  = 2*PI
    1 DUM   = .5*(EST1+EST2)
      CHI   = DUM - R1/R2*COS(DUM)
      IF (CHI.GE.PA) EST2=DUM
      IF (CHI.LT.PA) EST1=DUM
      IF (ABS(EST1-EST2).GE..00001) GOTO 1
      FUNCT = EST1

      RETURN
      END

C     ********* ********* ********* ********* ********* ********* ******
C     *** Functions MOD
C     *** Takes a modulus of a 3-vector
C     ********* ********* ********* ********* ********* ********* ******

      FUNCTION XMOD(A)

      IMPLICIT NONE
      REAL*8 A(3)
      REAL XMOD

      XMOD = SQRT(A(1)*A(1) + A(2)*A(2) + A(3)*A(3))

      RETURN
      END
```

Subroutines from the BIRH.TLB

```
C     ********* ********* ********* ********* ********* ********* ******
C     *** SUBROUTINE INIT : Calculates the point of emission of the ray
C     *** , the surface normal, and also the colour temperature.
C     ********* ********* ********* ********* ********* ********* ******

        SUBROUTINE INIT(POINT,ANORM,TEMP)

      IMPLICIT NONE

        REAL*8 POINT(3),ANORM(3)
        REAL*8 TEMP

C     ********* ********* ********* ********* ********* ********* ******
C     *** Variables from Tube CBlk
C     ********* ********* ********* ********* ********* ********* ******

      REAL*8 X, Y, RAD, EPS1, EPS2

C     ********* ********* ********* ********* ********* ********* ******
C     *** Variables from Temp CBlk
C     ********* ********* ********* ********* ********* ********* ******

      INTEGER NT
      REAL*8 XD(0:100), TT(0:100), TB(0:100)

C     ********* ********* ********* ********* ********* ********* ******
C     *** Variables from Const CBlk
C     ********* ********* ********* ********* ********* ********* ******

      REAL*8 PI, SIGMA0

C     ********* ********* ********* ********* ********* ********* ******
C     *** Local Variables
C     ********* ********* ********* ********* ********* ********* ******

        INTEGER J, K

        REAL*8 V(3), W(3)
        REAL*8 PAR1, PAR2, DUM2, TLENGTH, PHI, THETA, TTOP, TBOT
        REAL*8 G05CAF, FUNCT

C     ********* ********* ********* ********* ********* ********* ******
C     *** Common Blocks
C     ********* ********* ********* ********* ********* ********* ******

      COMMON /tube/ X, Y, RAD, EPS1, EPS2
      COMMON /temp/ NT, XD, TT, TB
      COMMON /const/ PI, SIGMA0
```

```
      IF (ABS(RAD).LT.1E-6) THEN
            POINT(1) = X*G05CAF(1.)
            PAR1     = POINT(1)
            POINT(2) = Y*G05CAF(1.)
            DUM2     = POINT(2)/Y
            POINT(3) = 0.
            ANORM(1) = 0.
            ANORM(2) = 0.
            ANORM(3) = -1.
      ELSE
            TLENGTH = 2.*X + PI*Y
            PAR1 = G05CAF(1.) * TLENGTH
            PAR2 = 2.*PI * G05CAF(1.)
            IF (PAR2.LE.PI) THEN
               DUM2 = PAR2/PI
            ELSE
               DUM2 = 2.- PAR2/PI
            ENDIF

            IF (PAR1.LT.X) THEN
               V(1) = PAR1
               V(2) = Y
               V(3) = 0.
               ANORM(1) = 0.
               ANORM(2) = DSIN(PAR2)
               ANORM(3) = DCOS(PAR2)
            ELSEIF (PAR1.GT.(TLENGTH-X)) THEN
               V(1) = TLENGTH - PAR1
               V(2) = -Y
               V(3) = 0.
               ANORM(1) = 0.
               ANORM(2) = -DSIN(PAR2)
               ANORM(3) = DCOS(PAR2)
            ELSE
               PHI = (PAR1-X)/Y
               V(1)= X+Y*DSIN(PHI)
               V(2)= Y*DCOS(PHI)
               V(3)= 0.
               THETA = FUNCT(RAD,Y,PAR2)
               ANORM(1) = DSIN(THETA)*DSIN(PHI)
               ANORM(2) = DSIN(THETA)*DCOS(PHI)
               ANORM(3) = DCOS(THETA)
            ENDIF
            CALL SCAL(W, RAD, ANORM)
            CALL ADD(POINT, V, W)
      ENDIF

            K = 1
11          K = K + 1
            IF (PAR1.GT.(XD(K))) GOTO 11
            J = K - 1
```

```
      TTOP = TT(J)+(PAR1-XD(J))/(XD(K)-XD(J))*(TT(K)-TT(J))
      TBOT = TB(J)+(PAR1-XD(J))/(XD(K)-XD(J))*(TB(K)-TB(J))
      TEMP = TBOT + DUM2*(TTOP-TBOT)

RETURN
END
```

```
C     ********* ********* ********* ********* ********* ********* ******
C     *** SUBROUTINE DIRECT: Finds the initial direction of the ray
C     ********* ********* ********* ********* ********* ********* ******

      SUBROUTINE DIRECT(ANORM, RAY)

      IMPLICIT NONE

      REAL*8 ANORM(3), RAY(3)

C     ********* ********* ********* ********* ********* ********* ******
C     *** Variables from Unitv CBlk
C     ********* ********* ********* ********* ********* ********* ******

      REAL*8 EX(3), EY(3), EZ(3)

C     ********* ********* ********* ********* ********* ********* ******
C     *** Variables from Const CBlk
C     ********* ********* ********* ********* ********* ********* ******

      REAL*8 PI, SIGMA0

C     ********* ********* ********* ********* ********* ********* ******
C     *** Local Variables
C     ********* ********* ********* ********* ********* ********* ******

      REAL*8 V1(3),V2(3),V3(3),V4(3)
      REAL*8 UX(3),UY(3)
      REAL*8 DUM, PAR3, PAR4
      REAL*8 XMOD, G05CAF

C     ********* ********* ********* ********* ********* ********* ******
C     *** Common Blocks
C     ********* ********* ********* ********* ********* ********* ******

      COMMON /unitv/ EX, EY, EZ
      COMMON /const/ PI, SIGMA0


      CALL CROSS(V1, EX, ANORM)
      DUM = XMOD(V1)
      IF (DUM.GT.0.1) GOTO 60

        CALL CROSS(V1, EY, ANORM)
        DUM = 1./XMOD(V1)

 60   CALL SCAL(UX, DUM, V1)
      CALL CROSS(UY, ANORM, UX)
      PAR3 = 2.*PI*G05CAF(1.)
      PAR4 = G05CAF(1.)
      PAR4 = DSQRT(PAR4)
      PAR4 = DASIN(PAR4)
```

```
DUM  = DCOS(PAR4)

CALL SCAL(V1, DUM, ANORM)
DUM = DSIN(PAR3)*DSIN(PAR4)
CALL SCAL(V2, DUM, UX)
DUM = DCOS(PAR3)*DSIN(PAR4)
CALL SCAL(V3, DUM, UY)
CALL ADD(V4,V1,V2)
CALL ADD(RAY, V3, V4)

RETURN
END
```

```
C     ********* ********* ********* ********* ********* ********* ******
C     *** SUBROUTINE INTERC: Checks to see if ray intersects with either
C     *** cylinder depending on the value of N
C     ********* ********* ********* ********* ********* ********* ******

      SUBROUTINE INTERC(RAY, POINT, DIST, NS, N)

      IMPLICIT NONE

      INTEGER N, NS
      REAL*8 RAY(3),POINT(3)
      REAL*8 DIST

C     ********* ********* ********* ********* ********* ********* ******
C     *** Variables from Tube CBlk
C     ********* ********* ********* ********* ********* ********* ******

      REAL*8 X, Y, RAD, EPS1, EPS2

C     ********* ********* ********* ********* ********* ********* ******
C     *** Local Variables
C     ********* ********* ********* ********* ********* ********* ******

      REAL*8 DUM, DUMA, DUMB, DUMC, DUMD, DUM1, DUM2
      REAL*8 ALAM

C     ********* ********* ********* ********* ********* ********* ******
C     *** Common Blocks
C     ********* ********* ********* ********* ********* ********* ******

      COMMON /tube/ X, Y, RAD, EPS1, EPS2

      DUMA = RAY(2)
      DUMB = POINT(2) + (-1.)**N*Y
      DUMC = RAY(3)
      DUMD = POINT(3)
      DUM1 = DUMA*DUMB + DUMC*DUMD
      DUM2 = DUMA*DUMA + DUMC*DUMC
      DUM  = DUM1*DUM1 -DUM2*(DUMB*DUMB+DUMD*DUMD - RAD*RAD)

      IF (DUM.LE.0) GOTO 70
        ALAM = -(DUM1 + DSQRT(DUM))/DUM2
      IF (ALAM.LT.1.E-20) GOTO 70
        DUM = POINT(1) + ALAM*RAY(1)
      IF (DUM.LT.0..OR.DUM.GT.X) GOTO 70
      IF (N.EQ.2.AND.ALAM.GT.DIST) GOTO 70
        NS = -N
        DIST = ALAM

 70   RETURN
      END
```

```
C     ********* ********* ********* ********* ********* ********* ******
C     *** SUBROUTINE TORUS: Returns NS=3 if ray hits torus
C     ********* ********* ********* ********* ********* ********* ******

      SUBROUTINE TORUS(RAY,POINT,DIST,NS)
      IMPLICIT NONE

      INTEGER NS
      REAL*8 RAY(3),POINT(3)
      REAL*8 DIST

C     ********* ********* ********* ********* ********* ********* ******
C     *** Variables from Tube CBlk
C     ********* ********* ********* ********* ********* ********* ******

      REAL*8 X, Y, RAD, EPS1, EPS2

C     ********* ********* ********* ********* ********* ********* ******
C     *** Local Variables
C     ********* ********* ********* ********* ********* ********* ******

      INTEGER J, NI
      REAL*8 V1(3),AA(5),XX(5),YY(5)
      REAL*8 TOL, DUM, DUMA, DUMB, DUMC, DUMD, DUME, DUMF, ALAM
      REAL*8 C02AEF, X02AAF, DOT

C     ********* ********* ********* ********* ********* ********* ******
C     *** Common Blocks
C     ********* ********* ********* ********* ********* ********* ******

      COMMON /tube/ X, Y, RAD, EPS1, EPS2


              TOL  = X02AAF(DUM)
              DUMA = DOT(RAY,RAY)
              V1(1)= POINT(1)-X
              V1(2)= POINT(2)
              V1(3)= POINT(3)
              DUMB = 2*DOT(V1,RAY)
              DUMC = DOT(V1,V1) - Y*Y - RAD*RAD
              DUMD = 4*Y*Y*RAY(3)*RAY(3)
              DUME = 8*Y*Y*RAY(3)*POINT(3)
              DUMF = 4*Y*Y*(POINT(3)*POINT(3) - RAD*RAD)
              AA(1)= DUMA*DUMA
              AA(2)= 2*DUMA*DUMB
              AA(3)= 2*DUMA*DUMC + DUMB*DUMB + DUMD
              AA(4)= 2*DUMB*DUMC + DUME
              AA(5)= DUMC*DUMC + DUMF
              IF (AA(5).LE.TOL) AA(5)=0.0
              J=0
              NI = 5
```

```
            CALL C02AEF(AA, NI, XX, YY,TOL,J)
            IF (J.NE.0)  WRITE(6,*) 'ROOT FAILURE J= ', J

            DO 85 J=1,4
                    DUM = ABS(YY(J))
                    IF (DUM.GT.1.E-25) GOTO 85
                ALAM = XX(J)

                    IF (ALAM.LT.1.E-25) GOTO 85
                    IF (ALAM.GE.DIST)   GOTO 85
                    DUM = POINT(1) + ALAM*RAY(1)
                    IF (DUM.LT.X)          GOTO 85
                    NS = -3
                    DIST = ALAM
85          CONTINUE
      RETURN
      END
```

```
C     ********* ********* ********* ********* ********* ********* ******
C     *** SUBROUTINE REFL: returns NS= reflector number if ray hits
C     ********* ********* ********* ********* ********* ********* ******

        SUBROUTINE REFL(RAY,POINT,DIST,NS)

        IMPLICIT NONE

        INTEGER NS
        REAL*8 RAY(3), POINT(3)
        REAL*8 DIST

C     ********* ********* ********* ********* ********* ********* ******
C     *** Variables from Reflect CBlk
C     ********* ********* ********* ********* ********* ********* ******

      INTEGER NSURF
      REAL*8 SURF(40, 3, 3)

C     ********* ********* ********* ********* ********* ********* ******
C     *** Local Variables
C     ********* ********* ********* ********* ********* ********* ******

      INTEGER J,K
      REAL*8 V(3),V1(3),V2(3),V3(3),V4(3),W(3)
      REAL*8 DUM, ALAM
      REAL*8 DOT

C     ********* ********* ********* ********* ********* ********* ******
C     *** Common Blocks
C     ********* ********* ********* ********* ********* ********* ******

      COMMON /reflect/ NSURF, SURF

              DO 100 J=1, NSURF
                      DO 90 K=1,3
                              DUM = SURF(J, 2, K)
                              V1(K)= POINT(K) - DUM
                              V2(K)= SURF(J, 1, K) - DUM
                              V3(K)= SURF(J, 3, K) - DUM
 90                   CONTINUE
                      CALL CROSS(V4, V2, V3)

                      IF (ABS(DOT(RAY,V4)).LT.1.E-20) THEN
                         ALAM = 1.E30
                      ELSE
                         ALAM = -DOT(V1, V4)/DOT(RAY, V4)
                      END IF

                      IF (ALAM.LT.1.E-15) GOTO 100
                      IF (ALAM.GT.DIST)   GOTO 100
```

```
                    CALL SCAL(V, ALAM, RAY)
                    CALL ADD(W, V, V1)
                    DUM = DOT(W, V2)/DOT(V2, V2)
                    IF (DUM.LT.0. .OR. DUM.GT.1.) GOTO 100
                DUM = DOT(W,V3)/DOT(V3,V3)
                IF (DUM.LT.0. .OR. DUM.GT.1.) GOTO 100
                    NS = J
                    DIST = ALAM
100         CONTINUE
        RETURN
        END
```

```
C     ********* ********* ********* ********* ********* ********* ******
C     *** SUBROUTINE NRAY: Determines properties of reflected rays
C     ********* ********* ********* ********* ********* ********* ******

      SUBROUTINE NRAY(RAY,POINT,ANORM,DIST,NS)

      IMPLICIT NONE

      INTEGER NS
      REAL*8 RAY(3),POINT(3),ANORM(3)
      REAL*8 DIST

C     ********* ********* ********* ********* ********* ********* ******
C     *** Variables from Tube CBlk
C     ********* ********* ********* ********* ********* ********* ******

      REAL*8 X, Y, RAD, EPS1, EPS2

C     ********* ********* ********* ********* ********* ********* ******
C     *** Variables from Reflect CBlk
C     ********* ********* ********* ********* ********* ********* ******

      INTEGER NSURF
      REAL*8 SURF(40, 3, 3)

C     ********* ********* ********* ********* ********* ********* ******
C     *** Local Variables
C     ********* ********* ********* ********* ********* ********* ******

      INTEGER K
      REAL*8 V(3),W(3),V1(3),V2(3),V3(3)
      REAL*8 DUM, DUMA
      REAL*8 XMOD, DOT

C     ********* ********* ********* ********* ********* ********* ******
C     *** Common Blocks
C     ********* ********* ********* ********* ********* ********* ******

      COMMON /tube/ X, Y, RAD, EPS1, EPS2
      COMMON /reflect/ NSURF, SURF


      CALL SCAL(V, DIST, RAY)
      CALL ADD(POINT, V, POINT)

      IF (NS.EQ.-3) GOTO 110
      IF (NS.GT.0)  GOTO 150

      V(1) = POINT(1)
      V(2) = Y
      IF (NS.EQ.-2) V(2)=-Y
      V(3) = 0.
```

```
      CALL SUB(W, POINT, V)
      DUM = 1./RAD
      CALL SCAL(ANORM, DUM, W)
      GOTO 200

110   V1(1) = POINT(1) - X
      V1(2) = POINT(2)
      V1(3) = 0.
      DUM   = Y/XMOD(V1)
      CALL SCAL(V, DUM, V1)
      V(1)  = V(1) + X
      CALL SUB(W, POINT, V)
      DUM   = 1./RAD
      CALL SCAL(ANORM, DUM, W)
      GOTO 200

150   DO 160 K=1,3
              V2(K) = SURF(NS,1,K) - SURF(NS,2,K)
              V3(K) = SURF(NS,3,K) - SURF(NS,2,K)
160   CONTINUE

      CALL CROSS(V1, V2, V3)
      DUM  = DOT(RAY, V1)
      DUMA = 1./XMOD(V1)
      IF (DUM.GT.0) DUMA = -DUMA
      CALL SCAL(ANORM, DUMA, V1)

200   DUM  = -2*DOT(ANORM,RAY)
      CALL SCAL(V1, DUM, ANORM)
      CALL ADD(RAY, RAY, V1)

      RETURN
      END
```

```
C     ************** CDZIESLER OCT 1987 **********************
C     ** Calculates the value of the projected area factor  **
C     ** (Fanger p172) for a given geometry of person and   **
C     ** incident ray, defined by the 3 vectors A, F, R     **
C     ** respectively.  The data is derived from the figures**
C     ** on p172/4 of Thermal Comfort, and the interpolation**
C     ** is performed using a NAG routine E02CAF/CBF which  **
C     ** employs a Chebyshev polynomial series.             **
C     ************** CDZIESLER OCT 1987 **********************

      SUBROUTINE PROJ(AXIS ,FRONT, RAY, FP)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)       !unit vectors defining
      REAL*8 AXIS(3), FRONT(3), RAY(3)          !persons geometry
      REAL*8 FP
      REAL*8 RP(2), C(3), RAYMAG, ALPHA, BETA   !arrays used internally
      REAL*8 A(40), FF(1), WORK(1000)           !and by NAG routine
      REAL*8 X(1), YR, NA, NW

      DATA RAD/57.29577951/
      DATA NA/40/, NW/1000/

      DATA (A(I), I=1,6)/   8.7067D+02,-6.4804D+00, 6.1451D+01,
     *              5.0873D-01,-2.5456D+01,-1.3770D-01/
      DATA (A(I), I=7,12)/ -2.5756D+02,-2.0173D+00,-2.6909D+01,
     *               -2.4494D+00, 1.2329D+01, 1.8822D+00/
      DATA (A(I), I=13,18)/-3.4598D+01, 2.5018D+00,-3.9575D+00,
     *               7.9977D-01, 6.5442D-01,-1.0883D+00/
      DATA (A(I), I=19,24)/ 2.0653D+01, 1.5901D+00, 1.5092D+00,
     *              1.3115D+00,-9.5786D-01,-5.8725D-01/
      DATA (A(I), I=25,30)/-4.2419D+00, 3.2255D-01,-1.1754D+00,
     *               2.3957D-01, 7.3973D-01, 1.3547D-01/
      DATA (A(I), I=31,36)/ 2.1774D+00, 8.3727D-01,-2.1639D-01,
     *              -1.5234D-01,-5.2449D-02,-2.8470D-01/

      CALL CROSS(C, AXIS, FRONT)
      RAYMAG = DSQRT(DOT(RAY, RAY))
      RAYMAG = 1./RAYMAG
      CALL SCAL(RAY, RAYMAG, RAY)

      RP(1) = DOT(C, RAY)
      RP(2) = DOT(FRONT, RAY)

      DUM   = DSQRT(RP(1)*RP(1) + RP(2)*RP(2))
      ALPHA = DACOS(-RP(2)/DUM)
      BETA  = DACOS(DUM)

      YR    = RAD*ALPHA
      X(1)  = RAD*BETA
      IFAIL = 1
```

```
CALL E02CBF(1,1,5,5,X(1),0.,90.,YR,0.,180.,FF,A,NA,WORK,NW,IFAIL)
IF (IFAIL.NE.0) STOP
FP    = FF(1)
RETURN
END
```

```
C     ********* ********* ********* ********* ********* ********* ******
C     *** SUBROUTINE DETOPT: records grid position of ray hit
C     ********* ********* ********* ********* ********* ********* ******

      SUBROUTINE DETOPT(POINT,RAY,DIST,PROP,FLUX)

      IMPLICIT NONE

C     ********* ********* ********* ********* ********* ********* ******
C     *** Parameters
C     ********* ********* ********* ********* ********* ********* ******

      REAL*8 POINT(3), RAY(3)
      REAL*8 DIST, PROP, FLUX

C     ********* ********* ********* ********* ********* ********* ******
C     *** Variables from Detect CBlk
C     ********* ********* ********* ********* ********* ********* ******

      INTEGER NX, NY
      REAL*8 XLO, YLO, DETLEN, DETWID

C     ********* ********* ********* ********* ********* ********* ******
C     *** Variables from Array CBlk
C     ********* ********* ********* ********* ********* ********* ******

      INTEGER HIT(0:100, 0:100)
      REAL*4 IRRAD(0:100, 0:100)

C     ********* ********* ********* ********* ********* ********* ******
C     *** Local Variables
C     ********* ********* ********* ********* ********* ********* ******

      INTEGER I,J
      REAL*8 V(3)

C     ********* ********* ********* ********* ********* ********* ******
C     *** Common Blocks
C     ********* ********* ********* ********* ********* ********* ******

      COMMON /detect/ NX, NY, XLO, YLO, DETLEN, DETWID
      COMMON /array/ IRRAD, HIT

      CALL SCAL(V,DIST,RAY)
      CALL ADD(POINT,POINT,V)

      V(1) = POINT(1) - XLO
      V(2) = POINT(2) - YLO
      I    = INT(V(1)*NX/DETLEN)
      J    = INT(V(2)*NY/DETWID)
      IRRAD(I,J) = IRRAD(I,J) + SNGL(PROP*FLUX)
      HIT(I,J)= HIT(I,J) + 1
```

```
RETURN
END
```

```
C     ********* ********* ********* ********* ********* ********* ******
C     *** SUBROUTINE OUTPUT: Write o/p to file FOUT
C     ********* ********* ********* ********* ********* ********* ******

      SUBROUTINE OUTPUT(FOUT)

      IMPLICIT NONE

C     ********* ********* ********* ********* ********* ********* ******
C     *** Parameters
C     ********* ********* ********* ********* ********* ********* ******

      CHARACTER*40 FOUT

C     ********* ********* ********* ********* ********* ********* ******
C     *** Variables from Detector CBlk
C     ********* ********* ********* ********* ********* ********* ******

      INTEGER NX, NY
      REAL*8 XLO, YLO, DETLEN, DETWID

C     ********* ********* ********* ********* ********* ********* ******
C     *** Variables from Array CBlk
C     ********* ********* ********* ********* ********* ********* ******

      INTEGER HIT(0:100, 0:100)
      REAL*4 IRRAD(0:100, 0:100)

C     ********* ********* ********* ********* ********* ********* ******
C     *** Local Variables
C     ********* ********* ********* ********* ********* ********* ******

      INTEGER I, J, TH
      REAL*8 AREA, WEIGHT, TF

C     ********* ********* ********* ********* ********* ********* ******
C     *** Common Blocks
C     ********* ********* ********* ********* ********* ********* ******
      COMMON /detect/ NX, NY, XLO, YLO, DETLEN, DETWID
      COMMON /array/ IRRAD, HIT

      OPEN (UNIT=10, FILE=FOUT, STATUS='NEW')
      AREA = DETLEN*DETWID
      WEIGHT = (NX*NY)/AREA

      TF = 0
      DO I=0,NX-1
        DO J=0,NY-1
          TF = TF + IRRAD(I,J)
          TH = TH + HIT(I,J)
        ENDDO
      ENDDO
```

```
      WRITE (10,*) 'Number of x and y nodes'
      WRITE (10,*) NX, NY
      WRITE (10,*) 'Coords of bottom left corner '
      WRITE (10,'(X,2F8.2)') XLO, YLO
      WRITE (10,*) 'Coords of top right corner '
      WRITE (10,'(X,2F8.2)') XLO+DETLEN, YLO+DETWID
      WRITE (10, *) 'Flux density W/Sq m'
      WRITE (10,9998) ((IRRAD(I,J)*WEIGHT, I=0,NX-1), J=0,NY-1)
      WRITE (10,*) 'Flux W'
      WRITE (10,9998) ((IRRAD(I,J), I=0,NX-1),J=0,NY-1)
      WRITE (10,*) 'Hits'
      WRITE (10,9997) ((HIT(I,J), I=0,NX-1),J=0,NY-1)
      WRITE (10,*) 'Total FLUX OUTPUT = '
      WRITE (10, '(X,F10.1)')TF
      WRITE (10,*) 'Total no. of hits '
      WRITE (10,*) TH

9998 FORMAT (<NX>F6.1)
9997 FORMAT (<NX>I5)
      CLOSE (UNIT=10)
      RETURN
      END
```

```
C     ********* ********* ********* ********* ********* ********* ******
C     *** Collection of Vector Algebra Subroutines for ZIRH
C     ********* ********* ********* ********* ********* ********* ******

C     ********* ********* ********* ********* ********* ********* ******
C     *** FUNCTION DOT: returns dot product of two 3-vectors
C     ********* ********* ********* ********* ********* ********* ******
      FUNCTION DOT(A,B)
              IMPLICIT NONE
              REAL*8 DOT
              REAL*8 A(3), B(3)
              DOT = A(1)*B(1) + A(2)*B(2) + A(3)*B(3)
              RETURN
      END

C     ********* ********* ********* ********* ********* ********* ******
C     *** SUBROUTINE ADD: returns vector sum of two 3-vectors
C     ********* ********* ********* ********* ********* ********* ******
      SUBROUTINE ADD(A,B,C)
      IMPLICIT NONE
              REAL*8 A(3), B(3), C(3)
              A(1) = B(1) + C(1)
              A(2) = B(2) + C(2)
              A(3) = B(3) + C(3)
              RETURN
      END

C     ********* ********* ********* ********* ********* ********* ******
C     *** SUBROUTINE SUB: returns vector subtraction of two 3-vectors
C     ********* ********* ********* ********* ********* ********* ******
      SUBROUTINE SUB(A, B, C)
      IMPLICIT NONE
              REAL*8 A(3), B(3), C(3)
              A(1) = B(1) - C(1)
              A(2) = B(2) - C(2)
              A(3) = B(3) - C(3)
              RETURN
      END

C     ********* ********* ********* ********* ********* ********* ******
C     *** SUBROUTINE SCAL: returns 3-vector multiplied by a scalar
C     ********* ********* ********* ********* ********* ********* ******
      SUBROUTINE SCAL(A, X, B)
      IMPLICIT NONE
              REAL*8 A(3), B(3)
              REAL*8 X
              A(1) = X*B(1)
              A(2) = X*B(2)
              A(3) = X*B(3)
              RETURN
      END
```

```
C     ********* ********* ********* ********* ********* ********* ******
C     *** SUBROUTINE CROSS: returns cross product of two 3-vectors
C     ********* ********* ********* ********* ********* ********* ******
      SUBROUTINE CROSS(A, B, C)
      IMPLICIT NONE
              REAL*8 A(3), B(3), C(3)
              A(1) = B(2)*C(3) - B(3)*C(2)
              A(2) = B(3)*C(1) - B(1)*C(3)
              A(3) = B(1)*C(2) - B(2)*C(1)
              RETURN
      END
```

Listing 2.2: PLAQ.FOR

```
C     ********* ********* ********* ********* ********* ********* ******
C     ***** PROGRAM PLAQ: a variation on the MC model which calculates
C     *****  fluxes, and flux-densities for ceramic plaque heaters.
C     ***** LINK with ZIRH/LIB + NAG11/LIB
C     *****
C     ***** Version 1.0                                                ******
C     ***** This version has been derived from the BIRH v1.1
C     ***** CDZ 30/5/90
C     ********* ********* ********* ********* ********* ********* ******

      IMPLICIT NONE

C     ********* ********* ********* ********* ********* ********* ******
C     *** Variables from Plaque data file
C     ********* ********* ********* ********* ********* ********* ******

      INTEGER NSURF, NTX, NTY
      REAL*8 SURF(40,3,3),PQTEMP(0:99,0:99)
      REAL*8 X, Y, EPS1, EPS2

C     ********* ********* ********* ********* ********* ********* ******
C     *** Variables from Detector data file
C     ********* ********* ********* ********* ********* ********* ******

      INTEGER NRAYS, NX, NY
      REAL*8 XLO, XHI, YLO, YHI, DETLEN, DETWID
      CHARACTER*6 DETTYP

C     ********* ********* ********* ********* ********* ********* ******
C     *** File name variables
C     ********* ********* ********* ********* ********* ********* ******

      CHARACTER*40 FDET, FPLQ, FOUT

C     ********* ********* ********* ********* ********* ********* ******
C     *** Variables from Main body of Program
C     ********* ********* ********* ********* ********* ********* ******

      INTEGER HIT(0:100, 0:100)
      INTEGER NS, IC, NPER, I, J, K, KEYB, VDU

      REAL*4 IRRAD(0:100, 0:100)
      REAL*8 RAY(3), POINT(3), ANORM(3)
      REAL*8 EX(3), EY(3), EZ(3)

      REAL*8 PLQ(3,3)
      REAL*8 EMISS, TOTFLX, DUM, FLUX, PROP, TEMP, DIST
      REAL*8 RADFAC, PI, SIGMA0
```

```
C     ********* ********* ********* ********* ********* ********* ******
C     *** Common Blocks
C     ********* ********* ********* ********* ********* ********* ******

      COMMON /plaque/ X, Y, EPS1, EPS2, PLQ
      COMMON /reflect/  NSURF, SURF
      COMMON /temp/ NTX, NTY, PQTEMP
      COMMON /detect/ NX, NY, XLO, YLO, DETLEN, DETWID
      COMMON /unitv/ EX, EY, EZ
      COMMON /array/ IRRAD, HIT
      COMMON /const/ PI, SIGMA0

C     ********* ********* ********* ********* ********* ********* ******
C     *** Set up unit vectors
C     ********* ********* ********* ********* ********* ********* ******

      DATA (EX(I),EY(I),EZ(I),I=1,3)/1.0,3*0.0,1.0,3*0.0,1.0/

C     ********* ********* ********* ********* ********* ********* ******
C     *** Define ConstaNTs
C     ********* ********* ********* ********* ********* ********* ******

      SIGMA0 = 5.729D-8
      PI     = 4. * DATAN(1.D0)
      KEYB   = 5
      VDU    = 6

C     ********* ********* ********* ********* ********* ********* ******
C     *** Initialize count variable
C     ********* ********* ********* ********* ********* ********* ******

      NPER   = 0

C     ********* ********* ********* ********* ********* ********* ******
C     *** Read Data in from specified files
C     ********* ********* ********* ********* ********* ********* ******

      WRITE (VDU,*) 'ENTer detector file name (with extension)'
      READ (KEYB, '(A)') FDET
      FDET = '[ZIESLERCD.FORT]'//FDET

      WRITE (VDU,*) 'Enter Plaque file name (with extension)'
      READ (KEYB, '(A)') FPLQ
      FPLQ = '[ZIESLERCD.FORT]'//FPLQ

      WRITE (VDU,*) 'Enter Output file name (with extension)'
      READ (KEYB, '(A)') FOUT
      FOUT = '[ZIESLERCD.FORT]'//FOUT
```

```
      OPEN (UNIT=10, FILE=FPLQ, STATUS='OLD')
      READ (10,*) NSURF
      READ (10,*) X, Y, EPS1, EPS2
      READ (10,*) (((SURF(I,J,K),K=1,3),J=1,3),I=1,NSURF)
      READ (10,*) NTX, NTY
      READ (10,*) ((PQTEMP(I,J),J=0,NTY-1),I=0,NTX-1)
      CLOSE (UNIT=10)

      NSURF = NSURF + 1
      OPEN (UNIT=10, FILE=FDET, STATUS='OLD')
      READ (10, '(A)') DETTYP
      READ (10, *) NRAYS
      READ (10, *) ((SURF(NSURF, J, K), K=1,3), J=1,3)
      READ (10, *) NX, NY
      CLOSE (UNIT=10)

C     ********* ********* ********* ********* ********* ********* ******
C     *** Initialize variables
C     ********* ********* ********* ********* ********* ********* ******

      TOTFLX = 0.0D0
      DUM    = 0.0D0
      IC     = 1

      XLO    = SURF(NSURF, 1, 1)
      YLO    = SURF(NSURF, 1, 2)
      XHI    = XLO
      YHI    = YLO

      DO 111 I= 2,3
         IF (SURF(NSURF,I,1).LT.XLO) XLO=SURF(NSURF,I,1)
         IF (SURF(NSURF,I,1).GT.XHI) XHI=SURF(NSURF,I,1)
         IF (SURF(NSURF,I,2).LT.YLO) YLO=SURF(NSURF,I,2)
         IF (SURF(NSURF,I,2).GT.YHI) YHI=SURF(NSURF,I,2)
 111  CONTINUE

      DETLEN = XHI - XLO
      DETWID = YHI - YLO

      RADFAC = SIGMA0*EPS1*X*Y/NRAYS

      CALL CERAMINIT(PLQ)

C     ********* ********* ********* ********* ********* ********* ******
C     *** Perform ray tracing over NRAYS
C     ********* ********* ********* ********* ********* ********* ******

      DO 1000 I= 1, NRAYS
```

```
C     ********* ********* ********* ********* ********* ********* ******
C     *** Define surface normal vector and calculate exit temperature
C     ********* ********* ********* ********* ********* ********* ******

        CALL PLQINIT(POINT,ANORM,TEMP)

C     ********* ********* ********* ********* ********* ********* ******
C     *** Determine initial direction of ray, and calculate the flux
C     ***  which it is carrying
C     ********* ********* ********* ********* ********* ********* ******

        CALL DIRECT(ANORM, RAY)
        FLUX = RADFAC*TEMP*TEMP*TEMP*TEMP

C     ********* ********* ********* ********* ********* ********* ******
C     *** Define the initial strength of the ray
C     ********* ********* ********* ********* ********* ********* ******

        PROP = 1.
 65     NS = -9
        DIST = 1.E29


C     ********* ********* ********* ********* ********* ********* ******
C     *** Check for iNTersection with plaque
C     ********* ********* ********* ********* ********* ********* ******

        CALL CERAM(RAY, POINT, DIST, NS)

C     ********* ********* ********* ********* ********* ********* ******
C     *** Check for iNTersection with any reflector or detector
C     ********* ********* ********* ********* ********* ********* ******

        CALL REFL(RAY, POINT, DIST, NS)

C     ********* ********* ********* ********* ********* ********* ******
C     *** No hit or low intensity ray
C     ********* ********* ********* ********* ********* ********* ******

        IF ((NS.EQ.-9).OR.(NS.EQ.NSURF)) GOTO 800
        IF (PROP.LT..001) GOTO 810

C     ********* ********* ********* ********* ********* ********* ******
C     *** Find intensity, starting position, and direction of reflected
C     ***  ray
C     ********* ********* ********* ********* ********* ********* ******

        IF (NS.LT.0) THEN
           EMISS=EPS1
        ELSEIF ((NS.GT.0).AND.(NS.LT.NSURF)) THEN
           EMISS=EPS2
        ENDIF
```

```
        PROP= PROP*(1-EMISS)
        CALL PLQNRAY(RAY, POINT, ANORM, DIST, NS)
        GOTO 65

C     ********* ********* ********* ********* ********* ********* ******
C     *** Log the downward flux and also the grid position on the
C     ***  detector at which the ray iNTersected
C     ********* ********* ********* ********* ********* ********* ******

 800    IF (RAY(3).LT.0.) THEN
           TOTFLX = TOTFLX + PROP*FLUX
        ENDIF

        IF (NS.EQ.NSURF) THEN
           CALL DETOPT(POINT, RAY, DIST, PROP, FLUX)
        ENDIF

 810    IF (MOD(I,NRAYS/25).EQ.0) THEN
          NPER = NPER + 4
          WRITE (6,945) 'The run is ', NPER, ' % complete'
 945      FORMAT (1X,A11,I3,A11)
        ENDIF

 1000 CONTINUE

      CALL OUTPUT(FOUT)

      STOP
      END
```

Source Code Module PLQCER.FOR

```
C     ********* ********* ********* ********* ********* ********* ******
C     *** SUBROUTINE CERAM: returns NS= -1 if ray hits plaque
C     ********* ********* ********* ********* ********* ********* ******

        SUBROUTINE CERAM(RAY,POINT,DIST,NS)

        IMPLICIT NONE

        INTEGER NS
        REAL*8 RAY(3), POINT(3)
        REAL*8 DIST

C     ********* ********* ********* ********* ********* ********* ******
C     *** Variables from Plaque CBlk
C     ********* ********* ********* ********* ********* ********* ******

      REAL*8 PLQ(3,3)
      REAL*8 X, Y, EPS1, EPS2

C     ********* ********* ********* ********* ********* ********* ******
C     *** Local Variables
C     ********* ********* ********* ********* ********* ********* ******

      INTEGER J,K
      REAL*8 V(3),V1(3),V2(3),V3(3),V4(3),W(3)
      REAL*8 DUM, ALAM
      REAL*8 DOT

C     ********* ********* ********* ********* ********* ********* ******
C     *** Common Blocks
C     ********* ********* ********* ********* ********* ********* ******

      COMMON /plaque/ X, Y, EPS1, EPS2, PLQ

      DO 90 K=1,3
              DUM   = PLQ(2, K)
              V1(K) = POINT(K) - DUM
              V2(K) = PLQ(1, K) - DUM
              V3(K) = PLQ(3, K) - DUM
 90   CONTINUE
      CALL CROSS(V4, V2, V3)

      IF (ABS(DOT(RAY,V4)).LT.1.E-20) THEN
         ALAM = 1.E30
      ELSE
         ALAM = -DOT(V1, V4)/DOT(RAY, V4)
      END IF

      IF (ALAM.LT.1.E-15) GOTO 100
      IF (ALAM.GT.DIST)   GOTO 100
```

```
      CALL SCAL(V, ALAM, RAY)
      CALL ADD(W, V, V1)
      DUM = DOT(W, V2)/DOT(V2, V2)
      IF (DUM.LT.0. .OR. DUM.GT.1.) GOTO 100
      DUM = DOT(W,V3)/DOT(V3,V3)
      IF (DUM.LT.0. .OR. DUM.GT.1.) GOTO 100
      NS = -1
      DIST = ALAM

  100 RETURN
      END

C     ********* ********* ********* ********* ********* ********* ******
C     *** FUNCTION CERAMINIT: initializes PLQ array
C     ********* ********* ********* ********* ********* ********* ******

      FUNCTION CERAMINIT(DUM)

      IMPLICIT NONE

      INTEGER CERAMINIT
      INTEGER DUM

C     ********* ********* ********* ********* ********* ********* ******
C     *** Variables from Plaque CBlk
C     ********* ********* ********* ********* ********* ********* ******

      REAL*8 PLQ(3,3)
      REAL*8 X, Y, EPS1, EPS2

C     ********* ********* ********* ********* ********* ********* ******
C     *** Common Blocks
C     ********* ********* ********* ********* ********* ********* ******

      COMMON /plaque/ X, Y, EPS1, EPS2, PLQ

      PLQ(1,1) = X/2.
      PLQ(1,2) = Y/2.
      PLQ(1,3) = 0.
      PLQ(2,1) = -X/2.
      PLQ(2,2) = Y/2
      PLQ(2,3) = 0.
      PLQ(3,1) = -X/2.
      PLQ(3,2) = -Y/2.
      PLQ(3,3) = 0.

      CERAMINIT = DUM
      RETURN
      END
```

Source Code Module: PLQINIT.FOR

```
C     ********* ********* ********* ********* ********* ********* ******
C     *** SUBROUTINE PLQINIT : Calculates the point of emission of the
C     *** ray, the surface normal, and also the colour temperature.
C     *** SPECIFIC to Program PLAQUE.
C     ********* ********* ********* ********* ********* ********* ******

        SUBROUTINE PLQINIT(POINT,ANORM,TEMP)

      IMPLICIT NONE

        REAL*8 POINT(3),ANORM(3)
        REAL*8 TEMP

C     ********* ********* ********* ********* ********* ********* ******
C     *** Variables from Plaque CBlk
C     ********* ********* ********* ********* ********* ********* ******

      REAL*8 X, Y, EPS1, EPS2, PLQ(3,3)

C     ********* ********* ********* ********* ********* ********* ******
C     *** Variables from Temp CBlk
C     ********* ********* ********* ********* ********* ********* ******

      INTEGER NTX, NTY
      REAL*8 PLQTEMP(0:99,0:99)

C     ********* ********* ********* ********* ********* ********* ******
C     *** Variables from Const CBlk
C     ********* ********* ********* ********* ********* ********* ******

      REAL*8 PI, SIGMA0

C     ********* ********* ********* ********* ********* ********* ******
C     *** Local Variables
C     ********* ********* ********* ********* ********* ********* ******

        INTEGER I, J

        REAL*8 PAR1, PAR2, TX1, TX2
        REAL*8 G05CAF, FUNCT

C     ********* ********* ********* ********* ********* ********* ******
C     *** Common Blocks
C     ********* ********* ********* ********* ********* ********* ******

      COMMON /plaque/ X, Y, EPS1, EPS2, PLQ
      COMMON /temp/ NTX, NTY, PLQTEMP
      COMMON /const/ PI, SIGMA0
```

```
 PAR1      = G05CAF(1.)
 POINT(1) = PAR1*X
 PAR2      = G05CAF(1.)
 POINT(2) = PAR2*Y
 POINT(3) = 0.
 ANORM(1) = 0.
 ANORM(2) = 0.
 ANORM(3) = -1.

 I     = INT(PAR1*(NTX-1))
 J     = INT(PAR2*(NTY-1))

 TX1   = PLQTEMP(I,J) +
$            (PAR1*(NTX-1)-I)*(PLQTEMP(I+1,J)-PLQTEMP(I,J))
 TX2   = PLQTEMP(I,J+1) +
$            (PAR1*(NTX-1)-I)*(PLQTEMP(I+1,J+1)-PLQTEMP(I,J+1))

 TEMP = TX1 + (PAR2*(NTY-1)-J)*(TX2-TX1)

 RETURN
 END
```

Source Code Module: PLQNRAY.FOR

```
C     ********* ********* ********* ********* ********* ********* ******
C     *** SUBROUTINE PLQNRAY: Determines properties of reflected rays
C     ********* ********* ********* ********* ********* ********* ******

        SUBROUTINE PLQNRAY(RAY,POINT,ANORM,DIST,NS)

        IMPLICIT NONE

        INTEGER NS
        REAL*8 RAY(3),POINT(3),ANORM(3)
        REAL*8 DIST

C     ********* ********* ********* ********* ********* ********* ******
C     *** Variables from Plaque CBlk
C     ********* ********* ********* ********* ********* ********* ******

      REAL*8 X, Y, RAD, EPS1, EPS2, PLQ(3,3)

C     ********* ********* ********* ********* ********* ********* ******
C     *** Variables from Reflect CBlk
C     ********* ********* ********* ********* ********* ********* ******

      INTEGER NSURF
      REAL*8 SURF(40, 3, 3)

C     ********* ********* ********* ********* ********* ********* ******
C     *** Local Variables
C     ********* ********* ********* ********* ********* ********* ******

      INTEGER K
      REAL*8 V(3),V1(3),V2(3),V3(3)
      REAL*8 DUM, DUMA
      REAL*8 XMOD, DOT

C     ********* ********* ********* ********* ********* ********* ******
C     *** Common Blocks
C     ********* ********* ********* ********* ********* ********* ******

      COMMON /plaque/ X, Y, EPS1, EPS2, PLQ
      COMMON /reflect/ NSURF, SURF


      CALL SCAL(V, DIST, RAY)
      CALL ADD(POINT, V, POINT)

      IF (NS.EQ.-1) THEN
         DO K=1,3
              V2(K) = PLQ(1,K) - PLQ(2,K)
              V3(K) = PLQ(3,K) - PLQ(2,K)
         ENDDO
```

```
ELSEIF (NS.GT.0) THEN
   DO K=1,3
        V2(K) = SURF(NS,1,K) - SURF(NS,2,K)
        V3(K) = SURF(NS,3,K) - SURF(NS,2,K)
   ENDDO
ENDIF

CALL CROSS(V1, V2, V3)
DUM  = DOT(RAY, V1)
DUMA = 1./XMOD(V1)
IF (DUM.GT.0) DUMA = -DUMA
CALL SCAL(ANORM, DUMA, V1)

DUM  = -2*DOT(ANORM,RAY)
CALL SCAL(V1, DUM, ANORM)
CALL ADD(RAY, RAY, V1)

RETURN
END
```

Listing 2.3: QLL.FOR

```
C     ********* ********* ********* ********* ********* ********* ******
C     ***** PROGRAM TO MODEL QLL HEATERS
C     ***** based on the model by W.BUTLER 1982: BIRH.FOR          ******
C     ***** Version 1.0                                            ******
C     ***** CDZ 31/5/90
C     ********* ********* ********* ********* ********* ********* ******

      IMPLICIT NONE

C     ********* ********* ********* ********* ********* ********* ******
C     *** Variables from .QLL data file
C     ********* ********* ********* ********* ********* ********* ******

      INTEGER NSURF, NTX, NTTHETA
      REAL*8 SURF(40,3,3), QLLTEMP(0:99,0:99)
      REAL*8 THETA0
      REAL*8 X, RAD, EPS1, EPS2

C     ********* ********* ********* ********* ********* ********* ******
C     *** Variables from Detector data file
C     ********* ********* ********* ********* ********* ********* ******

      INTEGER NRAYS, NX, NY
      REAL*8 XLO, XHI, YLO, YHI, DETLEN, DETWID
      CHARACTER*6 DETTYP

C     ********* ********* ********* ********* ********* ********* ******
C     *** File name variables
C     ********* ********* ********* ********* ********* ********* ******

      CHARACTER*40 FDET, FQLL, FOUT

C     ********* ********* ********* ********* ********* ********* ******
C     *** Variables from Main body of Program
C     ********* ********* ********* ********* ********* ********* ******

      INTEGER HIT(0:100, 0:100)
      INTEGER NS, IC, NPER, I, J, K, KEYB, VDU

      REAL*4 IRRAD(0:100, 0:100)
      REAL*8 RAY(3), POINT(3), ANORM(3)
      REAL*8 EX(3), EY(3), EZ(3)

      REAL*8 EMISS, TOTFLX, DUM, FLUX, PROP, TEMP, DIST
      REAL*8 PI, SIGMA0, RADFAC

C     ********* ********* ********* ********* ********* ********* ******
C     *** Common Blocks
C     ********* ********* ********* ********* ********* ********* ******
```

```
      COMMON /qll/ X, RAD, EPS1, EPS2
      COMMON /reflect/  NSURF, SURF
      COMMON /temp/ NTX, NTTHETA, THETA0, QLLTEMP
      COMMON /detect/ NX, NY, XLO, YLO, DETLEN, DETWID
      COMMON /unitv/ EX, EY, EZ
      COMMON /array/ IRRAD, HIT
      COMMON /const/ PI, SIGMA0

C     ********* ********* ********* ********* ********* ********* ******
C     *** Set up unit vectors
C     ********* ********* ********* ********* ********* ********* ******

      DATA (EX(I),EY(I),EZ(I),I=1,3)/1.0,3*0.0,1.0,3*0.0,1.0/

C     ********* ********* ********* ********* ********* ********* ******
C     *** Define Constants
C     ********* ********* ********* ********* ********* ********* ******

      SIGMA0 = 5.729D-8
      PI     = 4. * DATAN(1.D0)
      KEYB   = 5
      VDU    = 6

C     ********* ********* ********* ********* ********* ********* ******
C     *** Initialize count variable
C     ********* ********* ********* ********* ********* ********* ******

      NPER   = 0

C     ********* ********* ********* ********* ********* ********* ******
C     *** Read Data in from specified files
C     ********* ********* ********* ********* ********* ********* ******

      WRITE (VDU,*) 'Enter detector file name (with extension)'
      READ (KEYB, '(A)') FDET
      FDET = '[ZIESLERCD.FORT]'//FDET

      WRITE (VDU,*) 'Enter QLL file name (with extension)'
      READ (KEYB, '(A)') FQLL
      FQLL = '[ZIESLERCD.FORT]'//FQLL

      WRITE (VDU,*) 'Enter Output file name (with extension)'
      READ (KEYB, '(A)') FOUT
      FOUT = '[ZIESLERCD.FORT]'//FOUT

      OPEN (UNIT=10, FILE=FQLL, STATUS='OLD')
      READ (10,*) NSURF
      READ (10,*) X, RAD, EPS1, EPS2
      READ (10,*) ((SURF(I,J,K),K=1,3),J=1,3),I=1,NSURF)
      READ (10,*) NTX, NTTHETA, THETA0
      READ (10,*) ((QLLTEMP(I,J),J=0,NTTHETA-1),I=0,NTX-1)
      CLOSE (UNIT=10)
```

```
      NSURF = NSURF + 1
      OPEN (UNIT=10, FILE=FDET, STATUS='OLD')
      READ (10, '(A)') DETTYP
      READ (10, *) NRAYS
      READ (10, *) ((SURF(NSURF, J, K), K=1,3), J=1,3)
      READ (10, *) NX, NY
      CLOSE (UNIT=10)

C     ********* ********* ********* ********* ********* ********* ******
C     *** Initialize variables
C     ********* ********* ********* ********* ********* ********* ******

      TOTFLX = 0.0D0
      DUM    = 0.0D0
      IC     = 1

      XLO    = SURF(NSURF, 1, 1)
      YLO    = SURF(NSURF, 1, 2)
      XHI    = XLO
      YHI    = YLO

      DO 111 I= 2,3
         IF (SURF(NSURF,I,1).LT.XLO) XLO=SURF(NSURF,I,1)
         IF (SURF(NSURF,I,1).GT.XHI) XHI=SURF(NSURF,I,1)
         IF (SURF(NSURF,I,2).LT.YLO) YLO=SURF(NSURF,I,2)
         IF (SURF(NSURF,I,2).GT.YHI) YHI=SURF(NSURF,I,2)
 111  CONTINUE

      DETLEN = XHI - XLO
      DETWID = YHI - YLO

      RADFAC = SIGMA0*EPS1*2*PI*RAD*X/NRAYS

C     ********* ********* ********* ********* ********* ********* ******
C     *** Perform ray tracing over NRAYS
C     ********* ********* ********* ********* ********* ********* ******

      DO 1000 I= 1, NRAYS

C     ********* ********* ********* ********* ********* ********* ******
C     *** Define surface normal vector and calculate exit temperature
C     ***  and calculate the flux carried
C     ********* ********* ********* ********* ********* ********* ******

        CALL QLLINIT(POINT,ANORM,TEMP)
        FLUX = RADFAC*TEMP*TEMP*TEMP*TEMP
```

```
C     ********* ********* ********* ********* ********* ********* ******
C     *** Determine initial direction of ray
C     ********* ********* ********* ********* ********* ********* ******

        CALL DIRECT(ANORM, RAY)

C     ********* ********* ********* ********* ********* ********* ******
C     *** Define the initial strength of the ray
C     ********* ********* ********* ********* ********* ********* ******

        PROP = 1.
 65     NS = -9
        DIST = 1.E29

C     ********* ********* ********* ********* ********* ********* ******
C     *** Check for intersection with filament
C     ********* ********* ********* ********* ********* ********* ******

        CALL QLLINTERC(RAY, POINT, DIST, NS)

C     ********* ********* ********* ********* ********* ********* ******
C     *** Check for intersection with any reflector or detector
C     ********* ********* ********* ********* ********* ********* ******

 666    CALL REFL(RAY, POINT, DIST, NS)

C     ********* ********* ********* ********* ********* ********* ******
C     *** No hit or low intensity ray
C     ********* ********* ********* ********* ********* ********* ******

        IF ((NS.EQ.-9).OR.(NS.EQ.NSURF)) GOTO 800
        IF (PROP.LT..001) GOTO 810

C     ********* ********* ********* ********* ********* ********* ******
C     *** Find intensity, starting position, and direction of reflected
C     ***  ray
C     ********* ********* ********* ********* ********* ********* ******

        IF (NS.LT.0) THEN
           EMISS=EPS1
        ELSEIF ((NS.GT.0).AND.(NS.LT.NSURF)) THEN
           EMISS=EPS2
        ENDIF
        PROP= PROP*(1-EMISS)
        CALL QLLNRAY(RAY, POINT, ANORM, DIST, NS)
        GOTO 65

C     ********* ********* ********* ********* ********* ********* ******
C     *** Log the downward flux and also the grid position on the
C     ***  detector at which the ray intersected
C     ********* ********* ********* ********* ********* ********* ******
```

```
800     IF (RAY(3).LT.0.) THEN
            TOTFLX = TOTFLX + PROP*FLUX
        ENDIF

        IF (NS.EQ.NSURF) THEN
            CALL DETOPT(POINT, RAY, DIST, PROP, FLUX)
        ENDIF

810     IF (MOD(I,NRAYS/25).EQ.0) THEN
           NPER = NPER + 4
           WRITE (6,945) 'The run is ', NPER, ' % complete'
945        FORMAT (1X,A11,I3,A11)
        ENDIF

1000 CONTINUE

     CALL OUTPUT(FOUT)

     STOP
     END
```

Program Modules for QLL.FOR

```
C     ********* ********* ********* ********* ********* ********* ******
C     *** SUBROUTINE QLLINIT : Calculates the point of emission of the
C     *** ray, the surface normal, and also the colour temperature.
C     *** SPECIFIC to Program QLL.FOR
C     ********* ********* ********* ********* ********* ********* ******

        SUBROUTINE QLLINIT(POINT,ANORM,TEMP)

      IMPLICIT NONE

C     ********* ********* ********* ********* ********* ********* ******
C     *** Parameters
C     ********* ********* ********* ********* ********* ********* ******
        REAL*8 POINT(3),ANORM(3)
        REAL*8 TEMP

C     ********* ********* ********* ********* ********* ********* ******
C     *** Variables from QLL CBlk
C     ********* ********* ********* ********* ********* ********* ******
      REAL*8 X, RAD, EPS1, EPS2

C     ********* ********* ********* ********* ********* ********* ******
C     *** Variables from Temp CBlk
C     ********* ********* ********* ********* ********* ********* ******
      INTEGER NTX, NTTHETA
      REAL*8 THETA0, QLLTEMP(0:99,0:99)

C     ********* ********* ********* ********* ********* ********* ******
C     *** Variables from Const CBlk
C     ********* ********* ********* ********* ********* ********* ******

      REAL*8 PI, SIGMA0

C     ********* ********* ********* ********* ********* ********* ******
C     *** Local Variables
C     ********* ********* ********* ********* ********* ********* ******

        INTEGER I, J, K

        REAL*8 V(3), W(3)
        REAL*8 PAR1, PAR2, DUM1, DUM2, THETA
        REAL*8 TX1, TX2
        REAL*8 G05CAF, FUNCT

C     ********* ********* ********* ********* ********* ********* ******
C     *** Common Blocks
C     ********* ********* ********* ********* ********* ********* ******

      COMMON /qll/ X, RAD, EPS1, EPS2
      COMMON /temp/ NTX, NTTHETA, THETA0, QLLTEMP
```

```
      COMMON /const/ PI, SIGMA0

      PAR1 = G05CAF(1.)
      DUM1 = PAR1*X
      PAR2 = G05CAF(1.)
      DUM2 = 2.*PI*PAR2

      V(1) = DUM1
      V(2) = 0.
      V(3) = 0.
      ANORM(1) = 0.
      ANORM(2) = DSIN(DUM2 + THETA0)
      ANORM(3) = DCOS(DUM2 + THETA0)
      CALL SCAL(W, RAD, ANORM)
      CALL ADD(POINT, V, W)


      I    = INT(PAR1*(NTX-1))
      J    = INT(PAR2*NTTHETA)

      TX1  = QLLTEMP(I,J) +
     $               (PAR1*(NTX-1)-I)*(QLLTEMP(I+1,J)-QLLTEMP(I,J))
      IF ((J+1).NE.NTTHETA) THEN
         TX2  = QLLTEMP(I,J+1) +
     $               (PAR1*(NTX-1)-I)*(QLLTEMP(I+1,J+1)-QLLTEMP(I,J+1))
      ELSE
         TX2  = QLLTEMP(I, 0 ) +
     $               (PAR1*(NTX-1)-I)*(QLLTEMP(I+1, 0 )-QLLTEMP(I, 0 ))
      ENDIF
      TEMP = TX1 + (PAR2*NTTHETA-J)*(TX2-TX1)

      RETURN
      END
```

```
C     ********* ********* ********* ********* ********* ********* ******
C     *** SUBROUTINE QLLINTERC: Checks to see if ray intersects with
C     *** cylinder representing tungsten filament, returns -1 if YES
C     ********* ********* ********* ********* ********* ********* ******

      SUBROUTINE QLLINTERC(RAY, POINT, DIST, NS)

      IMPLICIT NONE

C     ********* ********* ********* ********* ********* ********* ******
C     *** Parameters
C     ********* ********* ********* ********* ********* ********* ******
      INTEGER NS
      REAL*8 RAY(3),POINT(3)
      REAL*8 DIST

C     ********* ********* ********* ********* ********* ********* ******
C     *** Variables from QLL CBlk
C     ********* ********* ********* ********* ********* ********* ******

      REAL*8 X, RAD, EPS1, EPS2

C     ********* ********* ********* ********* ********* ********* ******
C     *** Local Variables
C     ********* ********* ********* ********* ********* ********* ******

      REAL*8 DUM, DUMA, DUMB, DUMC, DUMD, DUM1, DUM2
      REAL*8 ALAM

C     ********* ********* ********* ********* ********* ********* ******
C     *** Common Blocks
C     ********* ********* ********* ********* ********* ********* ******

      COMMON /qll/ X, RAD, EPS1, EPS2

      DUMA = RAY(2)
      DUMB = POINT(2)
      DUMC = RAY(3)
      DUMD = POINT(3)
      DUM1 = DUMA*DUMB + DUMC*DUMD
      DUM2 = DUMA*DUMA + DUMC*DUMC
      DUM  = DUM1*DUM1 -DUM2*(DUMB*DUMB+DUMD*DUMD - RAD*RAD)

      IF (DUM.LE.0) GOTO 70
        ALAM = -(DUM1 + DSQRT(DUM))/DUM2
      IF (ALAM.LT.1.E-20) GOTO 70
        DUM = POINT(1) + ALAM*RAY(1)
      IF (DUM.LT.0..OR.DUM.GT.X) GOTO 70
        NS = -1
        DIST = ALAM

 70   RETURN
```

```
END
```

```
C     ********* ********* ********* ********* ********* ********* ******
C     *** SUBROUTINE QLLNRAY: Determines properties of reflected rays
C     ***  Specific to QLL.FOR
C     ********* ********* ********* ********* ********* ********* ******

        SUBROUTINE QLLNRAY(RAY,POINT,ANORM,DIST,NS)

        IMPLICIT NONE

        INTEGER NS
        REAL*8 RAY(3),POINT(3),ANORM(3)
        REAL*8 DIST

C     ********* ********* ********* ********* ********* ********* ******
C     *** Variables from QLL CBlk
C     ********* ********* ********* ********* ********* ********* ******

      REAL*8 X, RAD, EPS1, EPS2

C     ********* ********* ********* ********* ********* ********* ******
C     *** Variables from Reflect CBlk
C     ********* ********* ********* ********* ********* ********* ******

      INTEGER NSURF
      REAL*8 SURF(40, 3, 3)

C     ********* ********* ********* ********* ********* ********* ******
C     *** Local Variables
C     ********* ********* ********* ********* ********* ********* ******

      INTEGER K
      REAL*8 V(3),W(3),V1(3),V2(3),V3(3)
      REAL*8 DUM, DUMA
      REAL*8 XMOD, DOT

C     ********* ********* ********* ********* ********* ********* ******
C     *** Common Blocks
C     ********* ********* ********* ********* ********* ********* ******

      COMMON /qll/ X, RAD, EPS1, EPS2
      COMMON /reflect/ NSURF, SURF


      CALL SCAL(V, DIST, RAY)
      CALL ADD(POINT, V, POINT)

      IF (NS.GT.0)  GOTO 150

      V(1) = POINT(1)
      V(2) = 0.
      V(3) = 0.
```

```
      CALL SUB(W, POINT, V)
      DUM = 1./RAD
      CALL SCAL(ANORM, DUM, W)
      GOTO 200

150   DO 160 K=1,3
            V2(K) = SURF(NS,1,K) - SURF(NS,2,K)
            V3(K) = SURF(NS,3,K) - SURF(NS,2,K)
160   CONTINUE

      CALL CROSS(V1, V2, V3)
      DUM  = DOT(RAY, V1)
      DUMA = 1./XMOD(V1)
      IF (DUM.GT.0) DUMA = -DUMA
      CALL SCAL(ANORM, DUMA, V1)

200   DUM  = -2*DOT(ANORM,RAY)
      CALL SCAL(V1, DUM, ANORM)
      CALL ADD(RAY, RAY, V1)

      RETURN
      END
```

Listing 2.4: COMFORT.FOR

```
C     ********* ********* ********* ********* ********* ********* ******
C  Program to take data from files containing info re: air temperature
C   mean radiant temp, air velocity across an occupied zone, and prompt
C   for values of relatiev humidity, clothing and activity, these are
C   the combined to give PMV and PPD values over the zone.  These values
C   are o/p to two files with extensions .PMV and .PPD respectively.
C   The o/p files are intended to act as i/p to UNIRAS UNIMAP as
C   irregular data.    AUTHOR: CD Ziesler  23/11/89
C     ********* ********* ********* ********* ********* ********* ******
      PROGRAM COMFORT
C
      IMPLICIT NONE
C
      INTEGER Nx, Ny
      INTEGER I, J, L
      INTEGER IACT, ICLO
      INTEGER Iwant
C
      INTEGER MaxX, MaxY
      PARAMETER (MaxX=20, MaxY=20)
C
      REAL*4 FluxD(MaxX, MaxY), Tm(MaxX, MaxY)
      REAL*4 PPD(MaxX, MaxY), PMV(MaxX, MaxY)
C
      REAL*4 FPMV, FPPD
C
      REAL*4 RH, M, Icl
      REAL*4 Ta, Taa, Va

      REAL*4 Xlo, Xhi, Ylo, Yhi, Gx, Gy
C
C     ********* ********* ********* ********* ********* ********* ******
C     *** CONSTANTS
C     ********* ********* ********* ********* ********* ********* ******
      REAL*4 TOCa, SIGMA
      DATA SIGMA/5.67e-8/, TOCa/273.16/

      character file*80, line*80
C
C     ********* ********* ********* ********* ********* ********* ******
C     Prompt for name of data file
C     ********* ********* ********* ********* ********* ********* ******
C
      write (6,*) 'Enter the MC results file (with extension)'
      read (5,'(a)') file
      open (1,file=file,status='old',readonly)
C
C
```

```
C     ********* ********* ********* ********* ********* ********* ******
C     read in values of tmrt from data files
C     ********* ********* ********* ********* ********* ********* ******
C
      READ (1,'(q,a)') l,line
      READ (1,*) Nx, Ny
      READ (1,'(q,a)') l,line
      READ (1,*) Xlo, Ylo
      READ (1,'(q,a)') l,line
      READ (1,*) Xhi, Yhi
      READ (1,'(q,a)') l,line
      READ (1,*) ((FluxD(I,J),I=1,Nx),J=1,Ny)
      CLOSE (UNIT=1)
C
C     ********* ********* ********* ********* ********* ********* ******
C     *** Calculate grid spacing
C     ********* ********* ********* ********* ********* ********* ******
      Gx = (Xhi-Xlo)/Nx
      Gy = (Yhi-Ylo)/Ny
C
C     ********* ********* ********* ********* ********* ********* ******
C     prompt for rel humidity, clothing, activity
C     ********* ********* ********* ********* ********* ********* ******
C
      WRITE (5,*) 'Enter Air Temperature (deg C) :'
      READ (6,*) Ta
      WRITE (5,*) 'Enter Air Speed (m/s)         :'
      READ (6,*) Va
      WRITE (5,*) 'Enter relative humidity (%)   :'
      READ (6,*) RH

C     ********* ********* ********* ********* ********* ********* ******
C     *** Convert flux-densities to MRTs ***
C     ********* ********* ********* ********* ********* ********* ******
      Taa= Ta + TOCa

      DO 1000 I=1, NX
         DO 1000 J=1, NY
            Tm(I,J) = SQRT(SQRT(FluxD(I,J)/(SIGMA*4)+
     $           Taa*Taa*Taa*Taa)) - TOCa
 1000 CONTINUE

C     ********* ********* ********* ********* ********* ********* ******
C     *** Prompt for comfort variables
C     ********* ********* ********* ********* ********* ********* ******

 10   WRITE (6,*) 'Select Activity; 1) SEDENTARY (58 W/Sq m)'
      WRITE (6,*) '                 2) MEDIUM   (116 W/Sq m)'
      WRITE (6,*) '                 3) HEAVY    (174 W/Sq m)'
      READ (5,*) IACT
      IF (IACT.EQ.1) M =  58.
      IF (IACT.EQ.2) M = 116.
```

```
      IF (IACT.EQ.3) M = 174.
      WRITE (6,*)
      WRITE (6,*) 'Select Clothing; 1) NUDE    (0 mC.Sq m/W)'
      WRITE (6,*) '                 2) LIGHT  (78 mC.Sq m/W)'
      WRITE (6,*) '                 3) MEDIUM(155 mC.Sq m/W)'
      WRITE (6,*) '                 4) HEAVY (233 mC.Sq m/W)'
      READ (5,*) ICLO
      IF (ICLO.EQ.1) Icl =   0.D-3
      IF (ICLO.EQ.2) Icl =  78.D-3
      IF (ICLO.EQ.3) Icl = 155.D-3
      IF (ICLO.EQ.4) Icl = 233.D-3
C
C     ********* ********* ********* ********* ********* ********* ******
C     calculate pmv, ppd across zone
C     ********* ********* ********* ********* ********* ********* ******
C
      DO J=1, Ny
         DO I=1, Nx
            PMV(I,J) = FPMV(Ta, Tm(I,J),Va, RH, Icl, M)
            PPD(I,J) = FPPD(PMV(I,J))
          ENDDO
      ENDDO
C
C     ********* ********* ********* ********* ********* ********* ******
C     prompt for output file name
C     ********* ********* ********* ********* ********* ********* ******
C
      write (6,*) 'Enter the PMV filename (with extension)'
      read (5,'(a)') file
      open (1,file=file,status='new')

 20   write (6,*) 'Which format 1) Unimap i/p'
      write (6,*) '             2) Tabular'
      read (5,*) Iwant
      if (Iwant.EQ.1) then
C
C     ********* ********* ********* ********* ********* ********* ******
C     write unimap format file to default directory
C     ********* ********* ********* ********* ********* ********* ******
C
        DO J= 1,Ny
           DO I= 1,Nx
              IF (ABS(PMV(I,J).LE.3.)) THEN
                WRITE (1,999) (Xlo+(I-1)*Gx), (Ylo+(J-1)*Gy), PMV(I,J),
     $                                                        PPD(I,J)
              ENDIF
            ENDDO
        ENDDO
C
      elseif(Iwant.EQ.2) then
```

```
C
C     ********* ********* ********* ********* ********* ********* ******
C     write tabular format file to default directory
C     ********* ********* ********* ********* ********* ********* ******
C
        WRITE (1,*) 'Number of x and y nodes'
        WRITE (1,*) NX, NY
        WRITE (1,*) 'Coords of bottom left corner '
        WRITE (1,'(X,2F8.2)') Xlo, Ylo
        WRITE (1,*) 'Coords of top right corner '
        WRITE (1,'(X,2F8.2)') Xhi, Yhi
        WRITE (1,*) 'Air temp, Air speed, RH, Act, Clothing'
        WRITE (1,'(X,3F8.2, 2I5)') Ta, Va, RH, IACT, ICLO
        WRITE (1, *) 'Predicted Mean Vote'
        WRITE (1,9999) ((PMV(I,J), I=1,NX), J=1,NY)
        WRITE (1, *) 'Predicted percentage of Dissatisfied'
        WRITE (1,9998) ((PPD(I,J), I=1,NX), J=1,NY)
C
      else
        goto 20
      endif

      CLOSE (UNIT=1)


C
 999  FORMAT (x,4F8.2)
 9999 FORMAT (x,<NX>F8.2)
 9998 FORMAT (x,<NX>F6.1)
 9997 FORMAT (x,<NX>I5)
C
C     ********* ********* ********* ********* ********* ********* ******
C     another set of clothing, activity levels
C     ********* ********* ********* ********* ********* ********* ******
C
      WRITE (5,*) 'Do you want another set of clo/act levels? (1=yes) '
      READ (6,*) Iwant
      IF (Iwant.EQ.1) GOTO 10
      STOP
      END
C
C     ********* ********* ********* ********* ********* ********* ******
C     Function PMV: calculates the pmv from the six comfort paramters
C     ********* ********* ********* ********* ********* ********* ******
```

```
C
C*************************************************************************
C*************************************************************************
C*** SUBROUTINE version of the program which calculates the values of ***
C*** the PMV and PPD indices based on Fangers Eqn.                      ***
C*** These calculations are only valid if the values of the variables ***
C*** are between M   = 46 to 232 W/Sq m      [0.8 to 4 met]           ***
C***             Icl =  0 to 0.310 sq m.C/W  [0 to 2 clo]             ***
C***             Ta  = 10 to 30 C                                     ***
C***             Tr  = 10 to 40 C                                     ***
C***             Va  =  0 to 1 m/s                                    ***
C*** Reference ISO 7730-1984(E) Annex D                               ***
C*************************************************************************
C*************************************************************************
C***              **                              **                  ***
C*** AUTHOR CDZ   ** WRITTEN  iv88                **   LAST MOD 26v88 ***
C***              **                              **                  ***
C*************************************************************************
C*************************************************************************
C
      FUNCTION FPMV(Ta, Tm, Va, RH, Icl, M)
C
      IMPLICIT NONE
      REAL*4 Ta, Tm, Va, RH, Icl, M, W
      REAL*4 FPMV, PMV
C
      INTEGER Noi
C
      REAL*4 Eps, Pa, Mw
      REAL*4 Tak, Tmk, Tcl, Tcla
      REAL*4 Fcl, Fcic, P1, P2, P3, P4, PM1, PM2, PM3, PM4
      REAL*4 Xn, Xf, Hcn, Hc, Hcf
C
      Eps = 1.5e-4
C
      Mw  = M - W
C
C*** Compute the corresponding Fcl value ***
      Fcl = 1.05 + 0.645*Icl
      IF (Icl.LT.0.078) Fcl = Fcl - 0.05 + 0.645*Icl
      Fcic = Fcl * Icl
      P2  = Fcic * 3.96
      P3  = Fcic * 100
      Tak = Ta + 273.16
      P1  = Fcic * Tak
C*** Calculate Partial Press of water vapour from a linear regression ***
C***  done on data from CRC p D91-93                                    ***
      Pa  = RH*10**(9.257-2312/Tak)
      Tmk = Tm + 273.16
      P4  = 308.7 - 0.028*Mw + P2*(Tmk/100)**4
C
```

```
C*** First guess for surface temperature ***
      Tcla = Tak + (35.5-Ta)/(3.5*(6.45*Icl + 0.1))
      Xn   = Tcla/100
      Xf   = Xn
C
      Hcf  = 12.1*SQRT(Va)
      Noi  = 0
C
C*** Compute surface temperature of clothing by iterations ***
 100          Xf   = (Xf + Xn)/2
              Hcn  = 2.38*ABS(100*Xf - Tak)**0.25
              Hc   = MAX1(Hcf,Hcn)
              Xn   = (P4 + P1*Hc - P2*Xf**4)/(100 + P3*Hc)
              Noi  = Noi + 1
              IF (Noi.GT.150) THEN
                 PMV = 9999.999
                 GOTO 200
              ENDIF
C
              IF (ABS(Xn-Xf).GT.Eps) GOTO 100
      Tcl  = 100*Xn - 273.1
C
C*** Compute PMV ***
      PM1 = 3.96*Fcl*(Xn**4 - (Tmk/100)**4)
      PM2 = Fcl*Hc*(Tcl - Ta)
      PM3 = 0.303*EXP(-0.036*M) + 0.028
      PM4 = 0.0
      IF (Mw.GT.58.15) PM4 = 0.42*(Mw-58.15)
      PMV = PM3*(Mw - 3.05d-3*(5733-6.99*Mw-Pa) - PM4 -
     $1.7d-5*M*(5867-Pa) - 1.4d-3*M*(34-Ta) - PM1 - PM2)
C
 200  IF (ABS(PMV).LE.3) THEN
         FPMV = PMV
      ELSE
         FPMV = 9999.999
      ENDIF
C
      RETURN
      END
```

```
C
C     ********* ********* ********* ********* ********* ********* ******
C     Function PPD: calculates the ppd from the pmv
C     ********* ********* ********* ********* ********* ********* ******
      FUNCTION FPPD(PMV)
C
      IMPLICIT NONE
      REAL*4 FPPD, PMV
      IF (ABS(PMV).LE.3) THEN
          FPPD = 100 - 95*EXP(-0.03353*PMV**4 - 0.2179*PMV*PMV)
      ELSE
          FPPD = 100.
      ENDIF
C
      RETURN
      END

C
C     ********* ********* ********* ********* ********* ********* ******
C     SubR: OUTPUT, Outputs data to file in tabular form
C     ********* ********* ********* ********* ********* ********* ******
C
```

Chapter 3

Listing 3.1: TRIP.FOR

```
C     ********* ********* ********* ********* ********* ********* ******
C      *****PROGRAM TO MODEL IRH UNITS installed into rooms        ******
C      *****Based on the program by  W.BUTLER 1982                 ******
C     ***** Alteration version 1.1                                 ******
C     ***** This version is one step closer to the MRS edition of the
C     ***** because of the following differences from v1.0:
C     ***** 1) Implicit None
C     ***** 2) Two data i/p files
C     ***** 3) Machine dependent constant definitions
C     ***** 4) More consistent use of common blocks for removing
C     *****   redundant parameter passing
C     ***** This is a hacked together version of TRIP using the    ******
C     ***** BIRH v1.1 code to ensure that it runs OK
C     ***** CDZ 28/8/90
C     ********* ********* ********* ********* ********* ********* ******
      PROGRAM TRIP

      IMPLICIT NONE

C     ********* ********* ********* ********* ********* ********* ******
C     *** Variables from Tube data file
C     ********* ********* ********* ********* ********* ********* ******

      INTEGER NSURF, NT
      REAL*8 SURF(40,3,3), XD(0:100), TT(0:100), TB(0:100)
      REAL*8 X, Y, RAD, EPS1, EPS2

C     ********* ********* ********* ********* ********* ********* ******
C     *** Variables from keyboard
C     ********* ********* ********* ********* ********* ********* ******

      INTEGER NRAYS, NHEAT
      REAL*8 HEAT(40,3), ROOM(3)
      REAL*8 EPSR, MHT

C     ********* ********* ********* ********* ********* ********* ******
C     *** File name variables
C     ********* ********* ********* ********* ********* ********* ******

      CHARACTER*40 FROOM, FTUB, FOUT

C     ********* ********* ********* ********* ********* ********* ******
C     *** Variables from Main body of Program
C     ********* ********* ********* ********* ********* ********* ******

      INTEGER NS, IC, NPER, I, J, K, IT, JT, KT, IHEAT, KEYB, VDU

      REAL*8 Q(40,-3:40), TOTFLX(40)
```

```
      REAL*8 RAY(3), POINT(3), ANORM(3)
      REAL*8 EX(3), EY(3), EZ(3)

      REAL*8 EMISS, DUM, FLUX, PROP, TEMP, DIST
      REAL*8 PI, SIGMA0

C     ********* ********* ********* ********* ********* ********* ******
C     *** Common Blocks
C     ********* ********* ********* ********* ********* ********* ******

      COMMON /tube/ X, Y, RAD, EPS1, EPS2
      COMMON /reflect/  NSURF, SURF
      COMMON /temp/ NT, XD, TT, TB
      COMMON /unitv/ EX, EY, EZ
      COMMON /const/ PI, SIGMA0

C     ********* ********* ********* ********* ********* ********* ******
C     *** Set up unit vectors
C     ********* ********* ********* ********* ********* ********* ******

      DATA (EX(I),EY(I),EZ(I),I=1,3)/1.0,3*0.0,1.0,3*0.0,1.0/

C     ********* ********* ********* ********* ********* ********* ******
C     *** Define Constants
C     ********* ********* ********* ********* ********* ********* ******

      SIGMA0 = 5.729D-8
      PI     = 4. * DATAN(1.D0)
      KEYB   = 5
      VDU    = 6

C     ********* ********* ********* ********* ********* ********* ******
C     *** Initialize count variable
C     ********* ********* ********* ********* ********* ********* ******

      NPER   = 0

C     ********* ********* ********* ********* ********* ********* ******
C     *** Read Data in from specified files
C     ********* ********* ********* ********* ********* ********* ******

      WRITE (VDU,*) 'Enter tube file name (with extension)'
      READ (KEYB, '(A)') FTUB
      FTUB = '[ZIESLERCD.FORT]'//FTUB

      WRITE (VDU,*) 'Enter room file name (with extension)'
      READ (KEYB, '(A)') FROOM
      FROOM = '[ZIESLERCD.FORT]'//FROOM

      WRITE (VDU,*) 'Enter Output file name (with extension)'
      READ (KEYB, '(A)') FOUT
      FOUT = '[ZIESLERCD.FORT]'//FOUT
```

```
      WRITE (VDU,*) 'Enter mounting height (m)'
      READ (KEYB, *) MHT


C      ***      READ Data ROOM FILE ***

      OPEN (UNIT=10, FILE=FROOM, STATUS='OLD')
      READ (10,*) NRAYS,NHEAT
      READ (10,*) ROOM(1),ROOM(2),ROOM(3)
      READ (10,*) EPSR
      READ (10,*) ((HEAT(I,J),J=1,2),I=1,NHEAT)
      CLOSE (UNIT=10)

      DO I= 1,NHEAT
         HEAT(I,3)= MHT
      ENDDO

      OPEN (UNIT=10, FILE=FTUB, STATUS='OLD')
      READ (10,*) NSURF
      READ (10,*) X, Y, RAD, EPS1, EPS2
      READ (10,*) (((SURF(I,J,K),K=1,3),J=1,3),I=1,NSURF)
      READ (10,*) NT
      READ (10,*) (XD(I),I=1,NT)
      READ (10,*) (TT(I),I=1,NT)
      READ (10,*) (TB(I),I=1,NT)
      CLOSE (UNIT=11)


C      ****Add the number of walls (6) to the # of surfaces ****
      NSURF = NSURF + 6

C      ****Loop over Nheat heaters (Assume that non-active  ****
C      **** heaters are transparent)                        ****
      DO 111 IHEAT = 1,NHEAT

C      ****Calculate wall coords wrt RC ****
      CALL WALL(SURF, NSURF, ROOM)

C      ****Transform to current heater coords****

      DO 222  IT= NSURF-5, NSURF
      DO 222 JT= 1,3
      DO 222 KT= 1,3
         SURF(IT,JT,KT) =  SURF(IT,JT,KT) - HEAT(IHEAT,KT)
 222  CONTINUE

C     ********* ********* ********* ********* ********* ********* ******
C     *** Perform ray tracing over NRAYS
C     ********* ********* ********* ********* ********* ********* ******

      DO 1000 I= 1, NRAYS
```

```
C     ********* ********* ********* ********* ********* ********* ******
C     *** Define surface normal vector and calculate exit temperature
C     ********* ********* ********* ********* ********* ********* ******

        CALL INIT(POINT,ANORM,TEMP)

C     ********* ********* ********* ********* ********* ********* ******
C     *** Determine initial direction of ray, and calculate the flux
C     ***  which it is carrying
C     ********* ********* ********* ********* ********* ********* ******

        CALL DIRECT(ANORM, RAY)
        FLUX = SIGMA0*(TEMP**4)*EPS1*2.*PI*RAD*(2.*X+PI*Y)/NRAYS
        TOTFLX(IHEAT) = TOTFLX(IHEAT) + FLUX

C     ********* ********* ********* ********* ********* ********* ******
C     *** Define the initial strength of the ray
C     ********* ********* ********* ********* ********* ********* ******

        PROP = 1.
 65     NS = -9
        DIST = 1.E29

C     ********* ********* ********* ********* ********* ********* ******
C     *** Check for intersection with first cylinder
C     ********* ********* ********* ********* ********* ********* ******

        CALL INTERC(RAY, POINT, DIST, NS, 1)

C     ********* ********* ********* ********* ********* ********* ******
C     *** Check for intersection with second cylinder
C     ********* ********* ********* ********* ********* ********* ******

        CALL INTERC(RAY, POINT, DIST, NS, 2)

C     ********* ********* ********* ********* ********* ********* ******
C     *** Check for intersection with torus
C     ********* ********* ********* ********* ********* ********* ******

        CALL TORUS(RAY, POINT, DIST, NS)

C     ********* ********* ********* ********* ********* ********* ******
C     *** Check for intersection with any reflector or detector
C     ********* ********* ********* ********* ********* ********* ******

 666    CALL REFL(RAY, POINT, DIST, NS)

C     ********* ********* ********* ********* ********* ********* ******
C     *** No hit or low intensity ray
C     ********* ********* ********* ********* ********* ********* ******
```

```
        IF ((NS.EQ.-9).OR.(PROP.LT..001)) GOTO 810

C     ********* ********* ********* ********* ********* ********* ******
C     *** Find intensity, starting position, and direction of reflected
C     ***  ray
C     ********* ********* ********* ********* ********* ********* ******

        IF (NS.LT.0) THEN
           EMISS=EPS1
        ELSEIF ((NS.GT.0).AND.(NS.LT.NSURF-5)) THEN
           EMISS=EPS2
        ELSEIF ((NS.GE.NSURF-5).AND.(NS.LE.NSURF)) THEN
           EMISS=EPSR
        ENDIF

        Q(IHEAT, NS)= Q(IHEAT,NS) + FLUX*PROP*EMISS

        PROP= PROP*(1-EMISS)
        CALL NRAY(RAY, POINT, ANORM, DIST, NS)
        GOTO 65

 810    IF (MOD(I,NHEAT*NRAYS/25).EQ.0) THEN
           NPER = NPER + 4
           WRITE (6,945) 'The run is ', NPER, ' % complete'
 945       FORMAT (1X,A11,I3,A11)
        ENDIF

 1000 CONTINUE

 111  CONTINUE

      DO I= 1,NHEAT
      TOTFLX(NHEAT+1) = TOTFLX(NHEAT+1) + TOTFLX(I)
      DO J= -3, NSURF
         Q(NHEAT+1,J)= Q(NHEAT+1,J) + Q(I,J)
         Q(I,NSURF+1)= Q(I,NSURF+1) + Q(I,J)
         Q(NHEAT+1,NSURF+1) = Q(NHEAT+1,NSURF+1) + Q(I,J)
      ENDDO
      ENDDO


C   ***O/P interesting values of Q ***
      OPEN (UNIT=10, FILE=FOUT, STATUS='NEW')
      WRITE (10,9998) ((Q(I,J), I=1,NHEAT+1),J=-3,NSURF+1)
      WRITE (10,9998) (TOTFLX(I), I=1,NHEAT+1)
 9998 FORMAT (<NHEAT+1>F11.1)
      STOP
      END
```

Listing 3.2: ITV.CTR

```
// Template for the room temperature/heat flow calculation
// Info for the individual rooms must be entered into the
// appropriate variables, and the values of heat into the
// surfaces must be taken from the MC simulation CDZ 15iii89

// Re-written 15/8/90 by CDZ
// Addition of fn(RADHTC) and introduction of the conductance matrix
// If treated as a procedure it requires that the variables:
// 2nd modification 31/8/90 by CDZ to parameterize the U-value and Ro
// matrices.
// T_EXT = scalar, external temp (deg C)
// T     = 6-r-vector, of surface temps (deg C)
// IRH   = 7-c-vector, of heat i/ps to nodes (W)
// N     = scalar, Air change rate (/hr)
// U     = 6-r-vector, u-values of walls (W/sq m/K)
// emiss = 6-r-vector, emissivities of walls
// Ro    = 3-r-vector, room dimensions (length,width,height) (m)
// to be defined before it is called

// Define necessary functions

deff modulus -c  // [M] = MOD(X,I) performs modulus to base I, on matrix X
deff recip -c    // [M,D] = RECIP(X) performs reciprocal of X e.by.e except
                 // for the diagonal (which is zero), also returns the sum of
                 // the rows into column vector
deff rad_htc -c  // [R_HTC] = RADHTC(DIMS, EMISS, TEMP) radiaive htcs
deff refl -c     // [R] = REFL(X,S) reflects matrix X top/bottom or left/right
deff symm -c     // [M] = SYMM(X) copies top-right to bottom-left to give
                 // symmetric matrix
Ha = [3.0  3.0  4.3  4.3  3.0  3.0 ]; // surface film resis 1-6

A  = [Ro(2)*Ro(3) Ro(1)*Ro(3) Ro(1)*Ro(2) Ro(1)*Ro(2) Ro(1)*Ro(3) Ro(2)*Ro(3)]
V  = Ro(1)*Ro(2)*Ro(3); // Volume of room

// Calculate the radiative htc using function RADHTC
[Hr] = RADHTC(Ro, emiss, T);


// Find Conduction htc
for i=1:6; Hc(1,i) = 1/(1/U(i)-1/Ha(i)); end;

// Define the conductance matrix
G = 0*ones(8);
G(1:6,7) = (Ha*.A)';  // Convective column vector
G(1:6,8) = (Hc*.A)';  // Conductive column vector
G(7,8)   = N*V/3;     // Ventilation htc

G = symm(G);          // make G symmetric

A_6 = ones(1,6).*.A';
```

```
A_6 = A_6';
G(1:6,1:6) = Hr*.A_6; // Radiative interchange matrix
clear A_6;

// Define system of nodal equations to be solved
room = -G(1:7,1:7);
for i=1:7;..
     room(i,i) = sum(G(i,1:8));..
end;

g_ext = g(8,1:7);
irh    = irh + (t_ext*g_ext)';
if cond(room)<100. ,..
    new_t = inv(room)*irh;..
end;

des = 0*ones(6,7);
des(1:6,1:6) = eye(6);
t   = (des*new_t)';
dt  = new_t;
dt(8) = 0;
dt  = dt.*.ones(1,8) - dt'.*.ones(8,1);
q   = dt*.G;
```

Listing 3.3: ROOM_7_N.CTR

```
// This procedure sets up a 14 node simulation of a room being
// heated by flux directed into the six surface nodes and the air point
// node.
// Author CD Ziesler 10/8/90


//Fn definition

deff symm -c      // [M] = SYMM(X) copies top-right to bottom-left to give
                  // symmetric matrix
deff rad_htc -c   // [R_HTC] = RADHTC(ROOM,EMISS,TEMP)

// Physical properties of the surfaces and air
e    = 0.9*ones(1,6);   // Emissivities
rho  = 1500*ones(1,6);  // Densities (kg/m3)
S    = 650*ones(1,6);   // Specific heats (J/kg/K)
Sair = [1200];          // Specific heat of air (J/m3/K)

// Declare scale factors
Tmax = [30] ;           // Maximum temperature excursion (deg C)

// Declarations of useful constant
sig  = [5.67e-8];       // Stefan-Boltzman constant (W/m2/K4)

// Physical parameters of the problem
X    = [32 16 8];       // Length, width, height of the room (m)
V    = X(1)*X(2)*X(3);  // Volume of room (m3)
A    = [X(2)*X(3) X(1)*X(3) X(1)*X(2) X(1)*X(2) X(1)*X(3) X(2)*X(3)];
                        // Areas of the surfaces (m2)
Ha   = [3.0 3.0 4.3 4.3 3.0 3.0];  // convective htc of the surfaces (W/m2/K)
U    = [0.96 0.96 0.26 1.60 0.96 0.96]; // U-values of the surfaces (W/m2/K)
dX   = [0.30 0.30 0.50 0.05 0.30 0.30];// Thicknesses of surfaces (m)
T    = 20*ones(1,6);    // Initial temperatures of the surfaces (K)
N    = [1.0];           // Air change rate (/hr)

// Get info about specific heaters, esp. Qmax
do irh_7_n

// Calculate radiative HTC matrix Hr from geometry and emissivities

[Hr] = RADHTC(x, e, t);

// Calculate conductive HTC vector from U-values and Ha

for i=1:6; Hc(i) = 2/(1/U(i)-1/Ha(i)); end;

// New Conductance matrix

G = 0*ones(14);
```

```
G(1:6,7) = (Ha*.A)';       // column vector of convective conductance (W/K)

G(8:13,14)  = Hc*.A';    // column vector of conductances (W/K)

G(1:6,8:13) = diag(Hc*.A'); // conductance matrix (W/K)

G(7,14) = N*V/3;          // ventilation heat loss (W/K)

A_6    = ones(1,6).*.A'; // form 6,6 matrix for multiplication

G(1:6,1:6) = Hr*.A_6;     // radiative conductance (W/K)
clear A_6;

G = symm(G);              // make G symmetric

C(1:6) = rho*.S*.dX*.A;  //
C(7)   = Sair*V;          // Thermal capacitances (J/K)

C_6  = ones(1,6).*.C([7 1 2 3 4 5 6])';
C_7  = ones(1,7).*.C([7 1 2 3 4 5 6])';

// Define intermediate matrices

for i=1:6;E(i)=sum(G(i,:));end;
E = diag(E) - G(1:6,1:6);

F = -G(1:6,7:13);

J = G(7:13,1:6);
J = J/.C_6;

for i=1:7;K(i)=sum(G(i+6,:));end;
K = -diag(K) + G(7:13,7:13);
K = K/.C_7;

L = 0*ones(7,14);
L(1:7,7:13) = diag(diag(ones(7)/.C_7));
L(1,14)     = 1/C(7);

// Define intermediate matrices for solving simultaneous equations

E_1 = INV(E);

C_R  = 0*ones(7);

C_R(7,1) = 1;

C_R(1:6,1:7) = -E_1*F;

D_R = 0*ones(7,14);
```

```
D_R(1:6,1:6) = E_1;

A_R = K - J*E_1*F;

B_R = L; B_R(1:7,1:6) = J*E_1;

// Define normalization matrices in order to scale problem

Nx = diag(Tmax*ones(1,7));                    // state vector

Nu = diag(Qmax*ones(1,14)); // control vector

Ny = diag(Tmax*ones(1,7));                    // output vector

// Compute normalized system matrices

a_n = Nx\a_r*Nx;
b_n = Nx\b_r*Nu;
c_n = Ny\c_r*Nx;
d_n = Ny\d_r*Nu;
```

Listing 3.4: RADHTC.CTR

```
// [r_htc] = RADHTC (ro, e, t)
// inputs  ro : 3-vector containing length, width, height of room
//         e  : 6-vector      "     emissivities of surfaces 1-6
//         t  :     "         "     temperatures (deg C) of surfaces 1-6
// outputs r_htc : 6x6 radiative htc matrix

deff modulus -c  // [M] = MOD(X,I) performs modulus to base I, on matrix X
deff fpara -c    //
deff fperp -c    //
deff refl -c     // [R] = REFL(X,S) reflects matrix X top/bottom or left/right
deff symm -c     // [M] = SYMM(X) copies top-right to bottom-left to give
                 // symmetric matrix
deff rad_sym -c  // [M] = RAD_SYM(F,A) performs view factor symmetry operation
                 //  ie F(j,i) = F(i,j)*A(i)/A(j)

sig      = [5.67e-8]; // Stefan-Boltzmann constant (W/sq m.K**4)

A  = [Ro(2)*Ro(3) Ro(1)*Ro(3) Ro(1)*Ro(2) Ro(1)*Ro(2) Ro(1)*Ro(3) Ro(2)*Ro(3)]
T  = T + (273.16).*.ones(1,6);    // Converts temps input in deg C to K

//Compute the required Fij values using the macros Fpara and Fperp
f(1,2) = FPERP([ Ro(1)/Ro(3)  Ro(2)/Ro(3)]);
f(1,3) = FPERP([ Ro(1)/Ro(2)  Ro(3)/Ro(2)]);
f(2,3) = FPERP([ Ro(2)/Ro(1)  Ro(3)/Ro(1)]);
f(1,6) = FPARA([ Ro(2)/Ro(1)  Ro(3)/Ro(1)]);
f(2,5) = FPARA([ Ro(3)/Ro(2)  Ro(1)/Ro(2)]);
f(3,4) = FPARA([ Ro(1)/Ro(3)  Ro(2)/Ro(3)]);


f(2,1) = f(1,2)*a(1)/a(2); f(3,1) = f(1,3)*a(1)/a(3); f(3,2) = f(2,3)*a(2)/a(3

ft = [  0       f(1,2)  f(1,3)  f(1,3)  f(1,2)  f(1,6);
        f(2,1)  0       f(2,3)  f(2,3)  f(2,5)  f(2,1);
        f(3,1)  f(3,2)  0       f(3,4)  f(3,2)  f(3,1);
        f(3,1)  f(3,2)  f(3,4)  0       f(3,2)  f(3,1);
        f(2,1)  f(2,5)  f(2,3)  f(2,3)  0       f(2,1);
        f(1,6)  f(1,2)  f(1,3)  f(1,3)  f(1,2)  0    ];

// This section calculates the linearised radiant htcs from the view factors
// areas, emissivities, and temperatures available.  The algorithm is from
// Clarke "Energy Simulation in Building Design", p193.

fir  = (0.)*ones(6);
sec  = (0.)*ones(6);

for i=1:6;..
    for j=1:6;..
        if i<>j,..
           fir(j,i) = e(i)*e(j)*ft(i,j)*(t(i)*t(i) + t(j)*t(j))* ..
```

```
                    (t(i)+t(j)) / ..
                    (1 - (1-e(i))*(1-e(j))*ft(i,j)*ft(i,j)*a(i)/a(j));..
               dum(j,i) = e(i)*e(j)*a(j)*(t(i)*t(i) + t(j)*t(j))* ..
                     (t(i)+t(j));..
               for k=1:6;..
                  sec(j,i) = sec(j,i) + ( ( (1-e(k)) * ft(i,k) * ft(j,k)) / ..
                           (a(k) * (1- ((1-e(i))*(1-e(j))*(1-e(k)) ..
                           *ft(i,k)*ft(k,j)*ft(j,i)) )) );..
end;end;end;end;

r_htc = sig*(fir + dum*.sec);

RETURN
```

Chapter 4

Listing 4.1: SIMULATE.CTR

NOTE: Because CTRL-C is designed for interactive use much of the data input is performed in this interactive mode. Thus, commonly altered parameters, such as heater outputs or building characteristics can either be entered via the keyboard, or alternatively may be saved in .CTR files and invoked when required. This latter method is the one used by *SIMULATE*.

```
// This procedure runs the simulation of a room being
// heated by flux directed into the six surface nodes and the air point
// node.
// It calls the procedures 1) ROOM_PAR, defines the room Space-State
//                            Matrices
//                         2) IRH_PAR,  defines the heater Space-State
//                            Matrices
//                         3) performs room calculations
//                         4) CNR_PAR, defines controller Space-State
//                            Matrices and connects system up
// Author: CD Ziesler
// Date  : 20/9/90

//Function definition

deff symm -c      // [M] = SYMM(X) copies top-right to bottom-left to give
                  // symmetric matrix
deff rad_htc -c   // [R_HTC] = RADHTC(ROOM,EMISS,TEMP)

// Define building parameters
do room_par

// Physical properties of the air
Sair = [1200];          // Specific heat of air (J/m3/K)

// Declare scale factors
Tmax = [30] ;           // Maximum temperature excursion (deg C)

// Declarations of useful constant
sig  = [5.67e-8];       // Stefan-Boltzman constant (W/m2/K4)

// Calculate useful building quantities
V    = X(1)*X(2)*X(3);  // Volume of room (m3)
A    = [X(2)*X(3) X(1)*X(3) X(1)*X(2) X(1)*X(2) X(1)*X(3) X(2)*X(3)];
                        // Areas of the surfaces (m2)

// Get info about specific heaters, esp. Qmax
do irh_par
```

```
// Calculate radiative HTC matrix Hr from geometry and emissivities

[Hr] = RADHTC(x, e, t);

// Calculate conductive HTC vector from U-values and Ha

for i=1:6; Hc(i) = 2/(1/U(i)-1/Ha(i)); end;

// New Conductance matrix

G = 0*ones(14);

G(1:6,7) = (Ha*.A)';       // column vector of convective conductance (W/K)

G(8:13,14)  = Hc*.A';    // column vector of conductances (W/K)

G(1:6,8:13) = diag(Hc*.A'); // conductance matrix (W/K)

G(7,14) = N*V/3;          // ventilation heat loss (W/K)

A_6    = ones(1,6).*.A'; // form 6,6 matrix for multiplication

G(1:6,1:6) = Hr*.A_6;     // radiative conductance (W/K)
clear A_6;

G = symm(G);              // make G symmetric

C(1:6) = rho*.S*.dX*.A;  //
C(7)   = Sair*V;          // Thermal capacitances (J/K)

C_6  = ones(1,6).*.C([7 1 2 3 4 5 6])';
C_7  = ones(1,7).*.C([7 1 2 3 4 5 6])';

// Define intermediate matrices

for i=1:6;E(i)=sum(G(i,:));end;
E = diag(E) - G(1:6,1:6);

F = -G(1:6,7:13);

J = G(7:13,1:6);
J = J/.C_6;

for i=1:7;K(i)=sum(G(i+6,:));end;
K = -diag(K) + G(7:13,7:13);
K = K/.C_7;

L = 0*ones(7,14);
L(1:7,7:13) = diag(diag(ones(7)/.C_7));
L(1,14)     = 1/C(7);
```

```
// Define intermediate matrices for solving simultaneous equations

E_1 = INV(E);

C_R  = 0*ones(7);

C_R(7,1) = 1;

C_R(1:6,1:7) = -E_1*F;

D_R = 0*ones(7,14);

D_R(1:6,1:6) = E_1;

A_R = K - J*E_1*F;

B_R = L; B_R(1:7,1:6) = J*E_1;

// Define normalization matrices in order to scale problem

Nx = diag(Tmax*ones(1,7));                    // state vector

Nu = diag(Qmax*ones(1,14)); // control vector

Ny = diag(Tmax*ones(1,7));                    // output vector

// Compute normalized system matrices

a_n = Nx\a_r*Nx;
b_n = Nx\b_r*Nu;
c_n = Ny\c_r*Nx;
d_n = Ny\d_r*Nu;

do cnr_par;

// Return to Interactive Mode
```

Subsidiary Procedures: *room_par, irh_par, cnr_par*

These examples are particular files from a larger set of files containing a variety of room, heater, and controller data.

ROOM_PAR.CTR

```
// File MED_M.ROM defines medium thermal weight medium-sized building
// Room factors parameters

e    = 0.9*ones(1,6);   // Emissivities
rho  = [1500 1500 1200  960 1500 1500];  // Densities (kg/m3)
S    = [ 650  650  653  950  650  650];   // Specific heats (J/kg/K)
X    = [30 10 8];       // Length, width, height of the room (m)
Ha   = [3.0 3.0 4.3 4.3 3.0 3.0];  // convective htc of the surfaces (W/m2/K)
U    = [1.11 1.11 0.264 1.59 1.11 1.11]; // U-values of the surfaces (W/m2/K)
dX   = [0.25 0.25 1.35 0.075 0.25 0.25];// Thicknesses of surfaces (m)
T    = 10*ones(1,6);   // Initial temperatures of the surfaces (K)
N    = [1.0];           // Air change rate (/hr)
```

IRH_PAR.CTR

```
// FILE             = IRH_1_M_1.CTR
// HEATER           = IRH_1
// TIME-CONSTANT    = 600 s
// No of CONTROLS   = 1
// MOUNTING HEIGHT = 5 m

t_htr = 600;      // heater time constant (s)
f_htr = 1/t_htr; // half power frequency for heater

q_htr = [
0.265180E+04;
0.896360E+04;
0.411563E+05;
0.247900E+04;
0.530890E+04;
0.240780E+04;
0.734010E+04];

q_htr = q_htr';

Qmax    = sum(q_htr);
q_htr_n = (1/Qmax)*q_htr;              // normalize o/p

a_htr = -diag(f_htr*ones(1,7));
b_htr =  q_htr_n';      // weighting factor for the heaters o/p to nodes
c_htr = -a_htr;
[n_o,n_s]=size(c_htr);
[n_s,n_i]=size(b_htr);
d_htr =  0*ones(n_o,n_i);
```

CNR_PAR.CTR

```
// Procedure CNR_PAR.CTR which defines the SSMs for the
//  controller for the 6 surface room, and then puts together the whole
//  system ,ie heaters & external => building => controller
//  NB there will be several files with extensions .CNR which will
//  be different versions of this file.
//  Caveat: assumes that Qmax is available ie IRH_PAR.CTR has been executed

t_ctr = 250; // time constant of the controller (s)
f_ctr = 1/t_ctr;  // half-power freqency of the controller (Hz)

a_ctr = -diag(f_ctr*ones(1,7));
b_ctr = 2/3*diag(1/sum(A)*A);   // gives o/p of T_subjective
b_ctr(7,7) = 1/3;               // ie 1/3 T_a + 2/3 T_mrt
c_ctr = -diag(a_ctr)';
d_ctr =  0*ones(1,7);

Ey    = Qmax*diag(ones(1,7));   // normalization matrix for external o/p
Eu    = [Tmax];                 //     "            "    "     "     i/p

a_ext = 0*ones(7);
b_ext = 0*ones(7,1);
c_ext = a_ext;

d_ext = G(14,7:13)';
d_ext = d_ext([ 2 3 4 5 6 7 1]);
d_ext_n = Ey\d_ext*Eu;

[a_p,b_p,c_p,d_p] = paral(a_htr,b_htr,c_htr,d_htr,a_ext,b_ext,c_ext,d_ext_n);
[a_s,b_s,c_s,d_s] = series(a_p,b_p,c_p,d_p,a_n,b_n,c_n,d_n);
[a_c,b_c,c_c,d_c] = series(a_s,b_s,c_s,d_s,a_ctr,b_ctr,c_ctr,d_ctr);
```

Listing 4.2: PCL.CTR

```
// [y_r,x_r,u_r] = PCL(a,b,c,d,u_e,t,tl,tu,x0)
// inputs a    : (ns,ns) matrix
//        b    : (ns,nu) matrix
//        c    : (no,ns) matrix
//        d    : (no,nu) matrix
//        u_e  : r-vector containing the external temp. sequence
//        t    : r-vector containing time sequence
//        tl   : scalar, low temperature bound (deg C)
//        tu   : scalar, high      "       "      "
//        x0   : ns-c-vector of initial state
// outputs y_r : (no,imax) matrix, history of system o/ps
//        x_r  : (ns,imax) matrix, history of state variables
//        u_r  : (nu,imax) matrix, history of control variables
// Procedure Pseudo-Closed Loop (PCL) for 7 "surfaces" with normalization
// Author: CDZiesler  Date: 15/8/90
//  This procedure uses the library function "simu" repeatedly to generate
//  a time history of state,control, and output variables.  The switching
//  algorithm is in the first instance very simple.

[no,nu] = size(d);
[dum,imax] = size(t);
tmax = 30;
n_htr = nu-1;
tl_n = tl/tmax;
tu_n = tu/tmax;

i= 1;
u    = 0*ones(nu,2);
y    = 0*ones(no,2);

u(1:n_htr,1) = ones(n_htr,1);
u_r(1:n_htr,i) = u(1:n_htr,1);u_r(nu,i)= u_e(1,1); // record initial control
x_r(:,i) = x0;
for i=1:imax-1, ..
        u(:,1) = u_r(:,i);..
        u_i = u(1:n_htr,1);..
        [u_f] = switch(y(1,1),tu_n,tl_n,n_htr,u_i);..
        u(1:n_htr,2) = u_f;u(nu,2)=u_e(1,i+1);..
        ts(1:2)  = t(1,i:i+1);..  // specify integration limits
        simu('ic', x_r(:,i));..             //initialize system
        [y,x]    = simu(a,b,c,d,u,ts);..
        u_r(:,i+1) = u(:,2);..      //record control history
        y_r(:,i+1) = y(:,2);..           //  "    output    "
        x_r(:,i+1) = x(:,2);..           //  "    state     "
end;
RETURN
```

MSS.CTR: Procedure SWITCH

Software implementation of the Multiple Switching Strategy.

```
// [u_f] = SWITCH(t,tl,tu,n_htr, u_i)
// inputs t    : scalar, normalized current temperature
//        tl   : scalar, normalized low temperature bound
//        tu   : scalar, normalized high      "       "
//        n_htr: scalar, number of heaters
//        u_i  : c-vector with current switch states
// output u_f  : c-vector with new switch states
// Procedure SWITCH to be used in PCL
// Author: CDZiesler  Date: 28/8/90
// This subroutine allows the use of multi-stage switching for a room
// with a number of heaters

s = tu:(tl-tu)/n_htr:tl;  // divide up band into n_htr equal segments

for i=1:n_htr;..
    if t > s(i), u_f(i) = 0;end;..   // switch i th heater OFF
    if t < s(i+1), u_f(i) = 1;end;.. // switch i th heater ON
    if t > s(i+1),..
       if t < s(i),  u_f(i) = u_i(i); end;.. // leave heater i alone
    end;..
end;

RETURN
```

Chapter 6

Listing 6.1: IMAGE.FOR

```
C ***************************************************************
C ***** CONVersion program written to produce temperature data,*
C *****  based on G. Wood's program to convert the data files  *
C ***** obtained from the BMC/AGEMA thermography system into   *
C ***** temperature files.                                     *
C *************** Ammended and developed by CDZ 16xi88   ******
C ***************************************************************

      PROGRAM CONVERT

      INTEGER KEYS, VDU
      PARAMETER (KEYS= 5, VDU= 6, INFL=10)

      INTEGER IULEV, TRANGE, FDIST, FSTOP, LANGL
      INTEGER filt, range
      INTEGER BINFL(128,65)
      REAL*4 TMPFL(128,64)
      REAL*4 TMP(128,640)
      REAL*4 Z(128)
      DIMENSION KOLOR(10), ZCL(9)
      character*128 chbin
      DATA KOLOR/12,4,11,5,10,3,9,7,8,2/
C       DATA ZCL/128.,160.,192.,224.,232.,240.,248./
      DATA NINT/10/
      DATA TMAX,TMIN/0.0,9999.9/

      WRITE (VDU,*) 'Input number of files'
      READ (KEYS,*) NFILE

      DO IFILE= 0, NFILE-1
      CALL FILE(INFL, range, filt)
      CALL READ_BIN(INFL, BINFL, chbin)
      CLOSE(10)
      CALL DATA_LIN(chbin, INFL, IULEV, TRANGE, FDIST, FSTOP,
     $              LANGL, EMISS, AIRT, AMBT, TAU)
      CALL DATA_EDT(IULEV, TRANGE, FDIST, FSTOP, LANGL, EMISS,
     $              AIRT, AMRT)
      CALL DEGREE(BINFL, TMPFL, FLMIN, FLMAX,
     $              IULEV, TRANGE, FDIST, FSTOP, range, filt,
     $              LANGL, EMISS, AIRT, AMBT, TAU)

      TMAX= AMAX1(FLMAX,TMAX)
      TMIN= AMIN1(FLMIN,TMIN)

      DO I=1, 128
         DO J=1,64
            TMP(I,J+IFILE*64)= TMPFL(I,J)
```

```
         ENDDO
      ENDDO
      ENDDO

C  **Find max and min of temp data and use to assign colour ranges **


      DT= (TMAX-TMIN)/NINT
      DO I=1, NINT-1
         ZCL(I)= TMIN+ I*DT
      ENDDO

      WRITE (VDU,*) 'input xorigin,yorigin  (mm)'
      READ (KEYS,*) XORI, YORI
      WRITE (VDU,*) 'Specify grid size (X,Y) (mm)'
      READ (KEYS,*) XSIZ, YSIZ

      CALL GROUTE(0)
      CALL GOPEN


      CALL GIMORI(XORI,YORI)
      CALL GIMSIZ(XSIZ, YSIZ)

      CALL rclass(ZCL,NINT-1,0)
      CALL rshade(KOLOR,NINT)

      CALL GIMAGE(Z,128,0)

      DO J=1,64*NFILE
         DO I=1,128
            Z(I) = TMP(I,J)
         ENDDO
         CALL GIMAGE(Z,128,1)
         CALL GIMAGE(Z,128,1)
      ENDDO

      CALL GCOSCL(XORI-40.,YORI)
      CALL GIMAGE(Z,128,9999)
      CALL GCLOSE

      STOP
      END
```

```
C ***************************************
C Get all the image data from the i/p file
C ***************************************

      SUBROUTINE READ_BIN(IN, BIN, chbin)
      CHARACTER*128 CHBIN
      character*600 line
      INTEGER BIN(128,65)
      integer data(10000)

      id=0
      do i=1,17
         read (in,'(q,a)') l,line
         do j=1,l
            id=id+1
            data(id)=ichar(line(j:j))
         enddo
      enddo

      do iy= 1,64
         do ix= 1,128
            bin(ix,iy) = data(128*(iy-1)+ix)
         enddo
      enddo

      do i=33,1
         chbin((i-32):(i-32))= line(i:i)
      enddo


      RETURN
      END

C ***************************************
C Open the FLOW i/p data file ************
C ***************************************

      SUBROUTINE FILE(INFL, range, filt)

      INTEGER KEYS, VDU
      PARAMETER (KEYS=5, VDU=6)

      INTEGER range, filt
      CHARACTER NAME*40, ANSWER*1
      CHARACTER CDATE*9, CTIME*8

      INNF = 3
      WRITE (VDU, '(/////////////////////////)')
      WRITE (VDU, '(1H, 80(1H*))')
      WRITE (VDU, '( ''Program to convert raw thermographic data '',
     $               ''files to temperatures'')')
      WRITE (VDU, '( /''Original program by Graham Wood of MRS'')')
```

```
      WRITE (VDU, '( /''Modified by Chris Ziesler of Aston Uni'')')
      CALL TIME(CTIME)
      CALL DATE(CDATE)
      WRITE (VDU, '(1H,A9, '' '', A8)') CDATE, CTIME
      WRITE (VDU, '(1H, 80(1H*))')
      WRITE (VDU, '(/''(INCLUDE THE EXTENSION ON THE FILE NAME)'')')

C Open the file to read data
      WRITE (VDU, '(/''Enter the data file name: )'', $)')
      READ (KEYS, '(A40)') NAME
      OPEN (UNIT=INFL, FILE=NAME, STATUS='OLD', readonly)


      WRITE (VDU, '(/''Was a flame filter used (1=no, 2=yes):'')')
      READ (KEYS, *) filt
      WRITE (VDU, '(/''Was the range 0-100(=1), 0-1k(=2):'')')
      READ (KEYS, *) range

      RETURN
      END

C ***************************************
C * Read and convert the data line *******
C ***************************************

      SUBROUTINE DATA_LIN(chbin, IN, IULEV, TRANGE, FDIST, FSTOP,
     $                    LANGL, EMISS, AIRT, AMBT, TAU)
      CHARACTER CHBIN*128

      CHARACTER*2 CIULEV, CTRANGE, CFDIST, CFSTOP, CLANGL
      INTEGER IULEV, TRANGE, FDIST, FSTOP, LANGL

      CHARACTER*4 CEMISS, CAIRT, CAMBT

      TAU = 0.88 ! Default value of transmissivity

C  Get the last line of the .BIN file, which should be the data line
      CIULEV = CHBIN(1:2)
      CTRANGE= CHBIN(3:4)
      CFDIST = CHBIN(5:6)
      CFSTOP = CHBIN(7:8)
      CLANGL = CHBIN(9:10)

      CALL CVI(CIULEV, IULEV)
      CALL CVI(CTRANGE, TRANGE)
      CALL CVI(CFDIST, FDIST)
      CALL CVI(CFSTOP, FSTOP)
      CALL CVI(CLANGL, LANGL)

      CEMISS = CHBIN(13:16)
      CAIRT  = CHBIN(17:20)
      CAMBT  = CHBIN(21:24)
```

```
      CALL CVS(CEMISS, EMISS)
      CALL CVS(CAIRT, AIRT)
      CALL CVS(CAMBT, AMBT)

      AIRT = AIRT + 273.15
      AMBT = AMBT + 273.15

      RETURN
      END


C ***************************************
C * Convert 2 characters into an integer *
C ***************************************

      SUBROUTINE CVI(CHI, I)
      CHARACTER*2 CHI

      LSBYTE = ICHAR(CHI(1:1))
      MSBYTE = ICHAR(CHI(2:2))

      I = 256*MSBYTE + LSBYTE

      RETURN
      END

C ***************************************
C * Converts 4 chr$ into REAL*4    *******
C ***************************************

      SUBROUTINE CVS(CHR, R)
      CHARACTER*4 CHR
      INTEGER SMANT, SEXP
      INTEGER BYTE1, BYTE2, BYTE3, BYTE4

      BYTE1 = ICHAR(CHR(1:1))
      BYTE2 = ICHAR(CHR(2:2))
      BYTE3 = ICHAR(CHR(3:3))
      BYTE4 = ICHAR(CHR(4:4))

      SMANT = 1
      IF (BYTE2.GE.128) THEN
         SMANT=-1
         BYTE2 = BYTE2 - 128
      END IF
      SEXP = 1
      IF (BYTE1.GE.128) THEN
         SEXP = -1
         BYTE1 = BYTE1 -128
      END IF
```

```
      R = (65536. * BYTE2) + (256.*BYTE3) + BYTE4
      R = R / (2. **23)
      R = (SMANT*R) * (2.**(SEXP*BYTE1))
      RETURN
      END

C ***************************************
C * Convert binary data to temps   *******
C ***************************************

      SUBROUTINE DEGREE(BIN, TEMPS, TMIN, TMAX,
     $             IULEV, TRANGE, FDIST, FSTOP, range, filt,
     $              LANGL,  EMISS, AIRT, AMBT, TAU)

      INTEGER KEYS, VDU
      PARAMETER (KEYS= 5, VDU= 6)

      LOGICAL MFLAG
      CHARACTER BUFFER*20
      INTEGER IULEV, TRANGE, FDIST, FSTOP, LANGL, lens
      INTEGER range, filt

      REAL*4 TEMPS(128,64), Acoef(3,2,2,0:7), Bcoef(3,2,2,0:7)
      INTEGER BIN(128,64)

      data (Acoef(1,1,2,i),i=0,7) / 673215, 381375, 227043, 138585,
     $                               90780,  66162,  41304,  30677/
      data (Bcoef(1,1,2,i),i=0,7) / 3054.36, 3065.96, 3089.72, 3108.40,
     $                              3181.59, 3242.98, 3270.72, 3384.16/
      data (Acoef(1,2,2,i),i=0,7) /  71972,  47631,  28331,  15688,
     $                                8689,   5758,   3407,   2192/
      data (Bcoef(1,2,2,i),i=0,7) / 3572.30, 3668.70, 3722.53, 3735.72,
     $                              3732.42, 3728.03, 3714.84, 3750.00/
      data (Acoef(2,1,2,i),i=0,7) / 815240.8, 400485.3, 244391.9,
     $                    144609.4, 96198.9, 67478.9, 37766.0, 54321.4/
      data (Bcoef(2,1,2,i),i=0,7) / 3106.4, 3063.7, 3075.4, 3115.6,
     $                              3195.8, 3242.2, 3202.7, 3739.4/
      data (Acoef(2,2,2,i),i=0,7) / 98390.0, 54629.5, 48281.6, 16183.4,
     $                               7961.9,  6208.8,  3613.7,  1210.2/
      data (Bcoef(2,2,2,i),i=0,7) / 3745.2, 3764.8, 4091.4, 3767.2,
     $                              3617.3, 3778.1, 3764.8, 3699.2/
      data (Acoef(3,1,1,i),i=0,2) / 356997, 274763, 155209 /
      data (Bcoef(3,1,1,i),i=0,2) / 2851.9, 2960.5, 2962.5 /
      data (Acoef(3,1,2,i),i=0,7) / 381070, 202907, 152807, 98949,
     $                               84319,  51877,  32966, 20293/
      data (Bcoef(3,1,2,i),i=0,7) / 2877.4, 2866.7, 2972.6, 3032.9,
     $                              3279.6, 3247.1, 3312.8, 3355.6/


      IF (LANGL.EQ.7) THEN
          lens= 1
      ELSEIF (LANGL.EQ.20) THEN
```

```
          lens= 2
      ELSEIF (LANGL.EQ.40) THEN
          lens= 3
      ELSE
          STOP 'ERROR : no data for requested lens'
      ENDIF

      A = Acoef(lens, filt, range, fstop)
      B = Bcoef(lens, filt, range, fstop)

      FIA = A/ (EXP(B/AIRT) - 1)

      TOTAL = 0.0
      NCOUNT = 0
      MFLAG = .TRUE.

      DO 1000 IY= 1, 64
         DO 2000 IX= 1,128
            FIO = (REAL(IULEV)/10.)
     $            + TRANGE *(127.5 - REAL(BIN(IX,IY)))/255.
            FIO = 1. * FIO/(TAU*EMISS)
     $            - FIA*(1./(TAU*EMISS) - 1.)
            TEMP = B / LOG(A/FIO + 1.)
            IF (BIN(IX,IY).NE.255) THEN
               IF (MFLAG) THEN
                  TMAX = TEMP
                  TMIN = TEMP
                  MFLAG = .FALSE.
               ELSE
                  IF (TEMP.GT.TMAX) TMAX=TEMP
                  IF (TEMP.LT.TMIN) TMIN=TEMP
               ENDIF
               NCOUNT = NCOUNT + 1
               TOTAL = TOTAL + (TEMP-273.15)
            ENDIF
            TEMPS(IX,IY) = TEMP
 2000    CONTINUE
 1000 CONTINUE
      RETURN
      END

C
```

```
C     ********* ********* ********* ********* ********* ********* ******
C     ********* SUBROUTINE   DATA_EDT                               ******
C     ********* ********* ********* ********* ********* ********* ******
C
      SUBROUTINE DATA_EDT(IULEV, TRANGE, FDIST, FSTOP, LANGL, EMISS,
     $                    AIRT, AMBT)
C
      IMPLICIT NONE
      INTEGER IULEV, TRANGE, FDIST, FSTOP, LANGL
      REAL*4 EMISS, AIRT, AMBT
      INTEGER NCHOICE, EXIT
C
      PARAMETER (EXIT=9)
      NCHOICE = 1
      DO  WHILE (.NOT.((NCHOICE.EQ.EXIT)))
          CALL DATA_DIS(IULEV, TRANGE, FDIST, FSTOP, LANGL, EMISS,
     $                    AIRT, AMBT, NCHOICE)
          GOTO (100, 101, 102, 103, 104, 105, 106, 107,
     $          199) NCHOICE
          WRITE (6,*) 'Out of range, try again'
          GOTO 199
 100      WRITE (6,*) 'New value of thermal level : '
          READ (5,*) IULEV
          GOTO 199
 101      WRITE (6,*) 'New value of thermal range : '
          READ (5,*) TRANGE
          GOTO 199
 102      WRITE (6,*) 'New value of object dist   : '
          READ (5,*) FDIST
          GOTO 199
 103      WRITE (6,*) 'New value of f-stop        : '
          READ (5,*) FSTOP
          GOTO 199
 104      WRITE (6,*) 'New value of lens angle    : '
          READ (5,*) LANGL
          GOTO 199
 105      WRITE (6,*) 'New value of emissivity    : '
          READ (5,*) EMISS
          GOTO 199
 106      WRITE (6,*) 'New value of air temp      : '
          READ (5,*) AIRT
          GOTO 199
 107      WRITE (6,*) 'New value of ambient temp  : '
          READ (5,*) AMBT
 199  END DO
C
C
 200  RETURN
      END
C
```

```
C     ********* ********* ********* ********* ********* ********* ******
C     ********* SUBROUTINE   DATA_DIS                                ******
C     ********* ********* ********* ********* ********* ********* ******
C
      SUBROUTINE DATA_DIS(IULEV, TRANGE, FDIST, FSTOP, LANGL, EMISS,
     $                    AIRT, AMBT, NC)
C
      IMPLICIT NONE
      INTEGER IULEV, TRANGE, FDIST, FSTOP, LANGL
      REAL*4 EMISS, AIRT, AMBT
      INTEGER NCHOICE, EXIT
C
      INTEGER NC
C
      WRITE (6,*) '_____________________________________________________'
      WRITE (6,*) '********************ALTER*************************'
      WRITE (6,*) '******************PARAMETER***********************'
      WRITE (6,*) '*********************MENU*************************'
      WRITE (6,*) '-----------------------------------------------------'
      WRITE (6,*) '     Variable                :       Current Value '
      WRITE (6,*) '-----------------------------------------------------'
      WRITE (6,902) '1) Thermal level               :', IULEV
      WRITE (6,902) '2) Thermal range               :', TRANGE
      WRITE (6,902) '3) Object Dist                 :', FDIST
      WRITE (6,902) '4) F-stop                      :', FSTOP
      WRITE (6,902) '5) Lens Angle                  :', LANGL
      WRITE (6,901) '6) Emissivity                  :', EMISS
      WRITE (6,901) '7) Air temp                    :', AIRT
      WRITE (6,901) '8) Ambient temp                :', AMBT
      WRITE (6,*)   '9) Return to main              :'
      WRITE (6,*) '_____________________________________________________'
      WRITE (6,*) '   Which option ?'
      READ (5,*) NC
 901  FORMAT (X, A, F8.2)
 902  FORMAT (X, A, I6)
C
      RETURN
      END
```

Listing 6.2: TIME.FOR

Many of the subroutines in this program are the same as IMAGE.FOR they have been excluded from this listing.

```
C ************************************************************
C ***** Conversion program written to produce temperature data,*
C *****  based on G. Wood's prog to convert the data files     *
C ***** obtained from the BMC/AGEMA thermography system into   *
C ***** temperature files.                                     *
C *************** Ammended and developed by CDZ 16xi88   ******
C ************************************************************

      PROGRAM TIME

      INTEGER KEYS, VDU, OUT
      PARAMETER (KEYS= 5, VDU= 6, INFL=10, OUT=11)

      INTEGER IULEV, TRANGE, FDIST, FSTOP, LANGL
      INTEGER filt, range
      INTEGER BINFL(128,65)

      REAL*4 TMP(128,64)
      REAL*4 TMIN, TMAX, TMEAN, TMEAN4
      REAL*4 SIGMA
      character*128 chbin
      CHARACTER*40 PREFIX, NAME, FOUT
      COMMON /temp/ TMP
      DATA SIGMA/5.67E-8/

      WRITE (VDU,*) 'Input number of files'
      READ (KEYS,*) NFILE

      CALL FILE(range, filt, PREFIX)

      OPEN(UNIT=OUT, FILE=PREFIX,STATUS='NEW')
      WRITE (OUT, *)
     $ 'mT4 (K)      Tmax (K)   Tmean(K)   sig.e.T4 (W/sq m)'
      WRITE (OUT, *)
     $ '_____________________________________________________'

      DO IFILE= 0, NFILE-1
        CALL ADD_SUF(PREFIX, NAME, IFILE)
        OPEN (UNIT=INFL, FILE=NAME, STATUS='OLD', readonly)
        CALL READ_BIN(INFL, BINFL, chbin)
        CLOSE(10)
        CALL DATA_LIN(chbin, INFL, IULEV, TRANGE, FDIST, FSTOP,
     $              LANGL, EMISS, AIRT, AMBT, TAU)
        IF (TRANGE.GT.1000) TRANGE = 1000
        CALL DEGREE(BINFL, TMP, TMIN, TMAX,
     $              IULEV, TRANGE, FDIST, FSTOP, range, filt,
     $              LANGL, EMISS, AIRT, AMBT, TAU)
```

```
        CALL F_ROOT(TMEAN, TMEAN4)
        WRITE (OUT,99) TMEAN4, TMAX, TMEAN,
     $                 SIGMA*EMISS*TMEAN4**4

      ENDDO
      WRITE (OUT,*) 'Tmin for these files was = ', TMIN
      CLOSE(OUT)
 99   FORMAT (1X, F9.1, 2X, F9.1, 2X, F9.1, 2X, E10.3)

      STOP
      END

C ***************************************
C SUBR F_ROOT
C Computes the fourth root mean and the mean of tmps
C ***************************************

      SUBROUTINE F_ROOT(MEAN,MEAN4)

      IMPLICIT NONE
      REAL*4 MEAN, MEAN4

      INTEGER I,J,SUM
      REAL*4 T,ST,ST4

      REAL*4 TMP(128,64)
      COMMON /temp/ TMP

      ST  = 0.
      ST4 = 0.
      SUM = 0

      DO J=1,64
         DO I= 1,128
               T   = TMP(I,J)
               ST  = T + ST
               ST4 = T*T*T*T + ST4
               SUM = SUM + 1
      ENDDO
      ENDDO

      MEAN4  = ST4/SUM
      MEAN4  = SQRT(SQRT(MEAN4))
      MEAN   = ST/SUM

      RETURN
      END
```

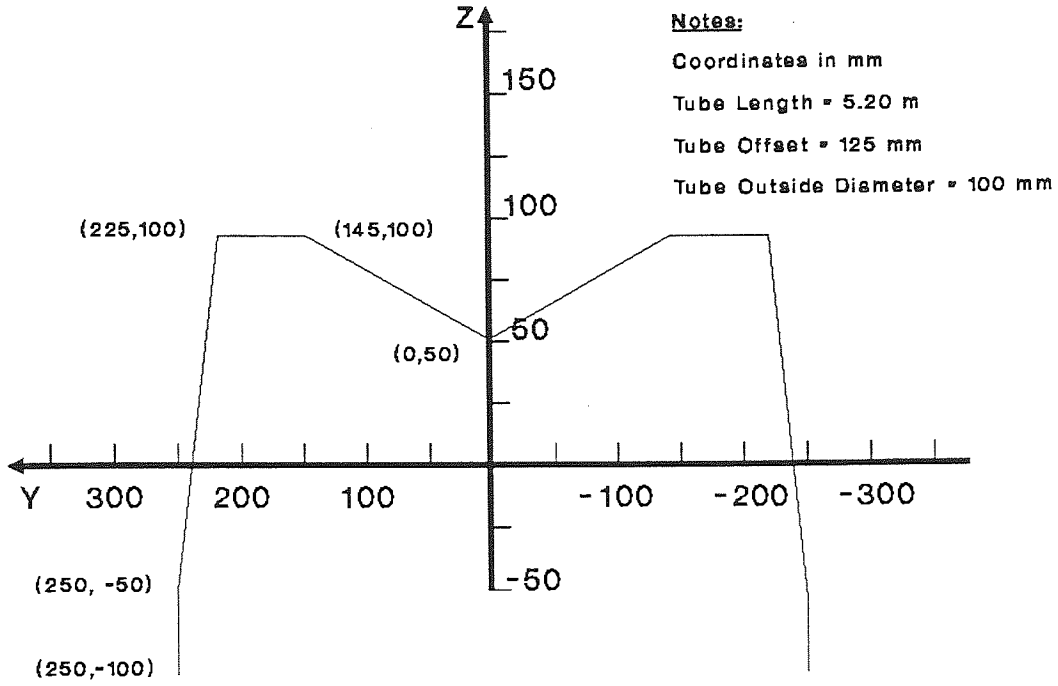
Appendix B: Heater Cross-Sections

The cross-sections that follow form the basis of all the modelling work in the thesis, in the sense that it was from this data that the Monte Carlo model made its predictions. Of the eight heaters outlined the information for the heaters IRH_3, IRH_4, IRH_5, and IRH_6 came from "A Comparison of Four Gas-Fired Radiant Heaters" by Dr J.K. Maund. Information on the other heaters was gathered from heaters available in the Chemical Engineering Department.

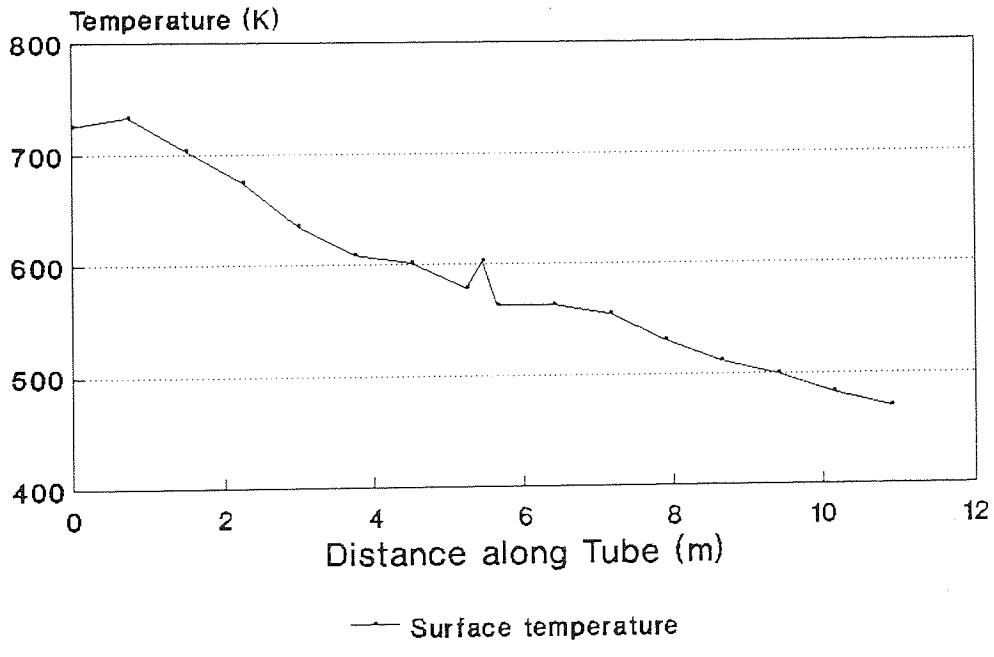
The model derived its input from files in which both the heater and detector information were stored. The annotation on each diagram adds the other three essential pieces of information which the model needs, the length of the tubes, the offset of the tubes from the origin, and the outside diameter of the tubes. The tube offset is the distance from the origin to the centre of the tube.

The tube temperatures, which are also shown, were all measured with a surface thermocouple, some by Dr J.K. Maund and some by the author. These were the temperatures used in the modelling work. Most often the temperatures of the top of the tubes was estimated, as no reliable data was available. Often these estimates were called into question later as better data became available. It was usually found that they were too low, which resulted in efficiencies and irradiances which were an underestimate.

Fig.B.1: Cross-Section of IRH_1

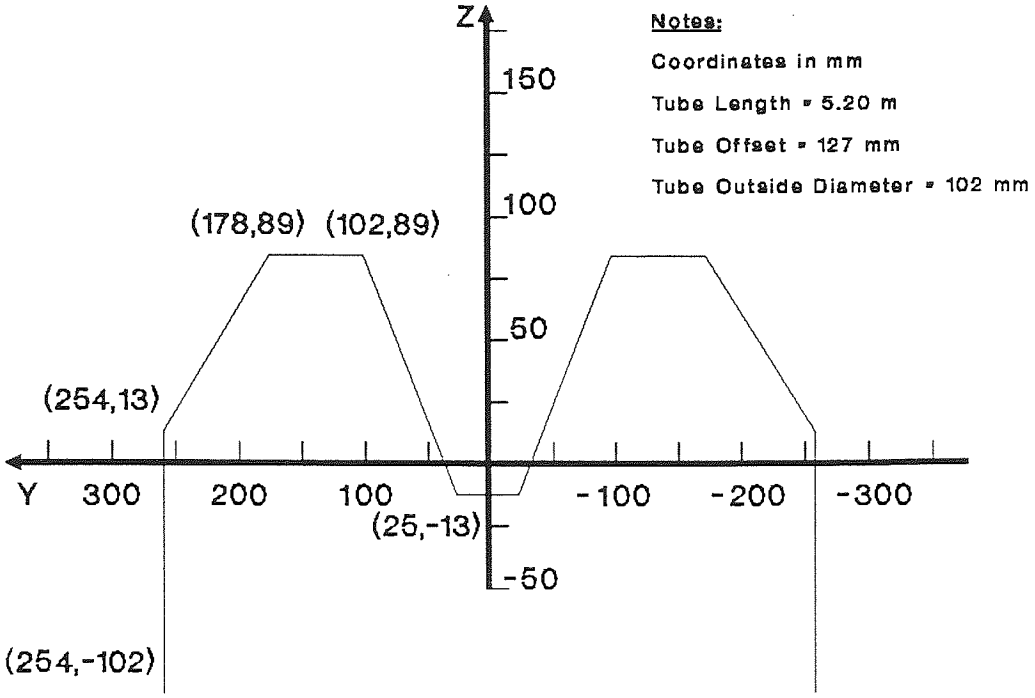


IRH_1: Tube Temperatures



Top and Bottom Temperature the same

Fig.B.2: Cross-Section of IRH_2



IRH_2: Tube Temperatures

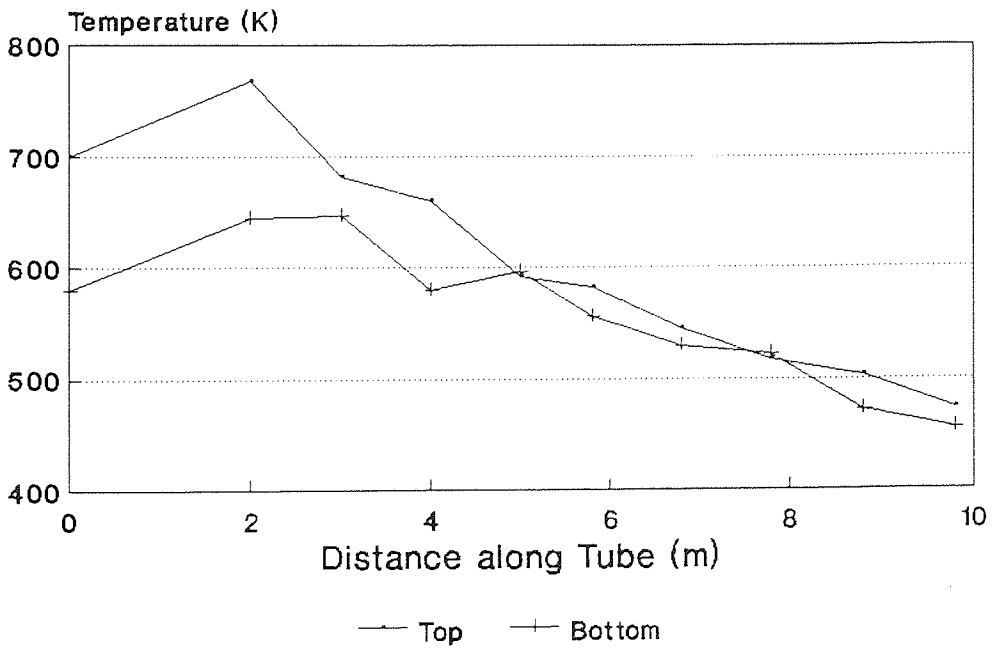
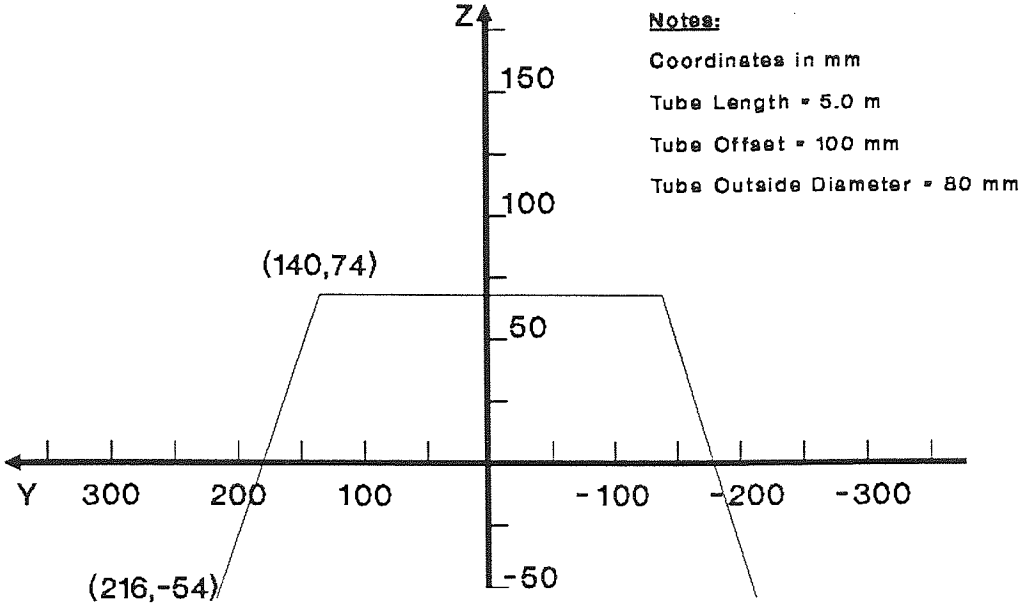
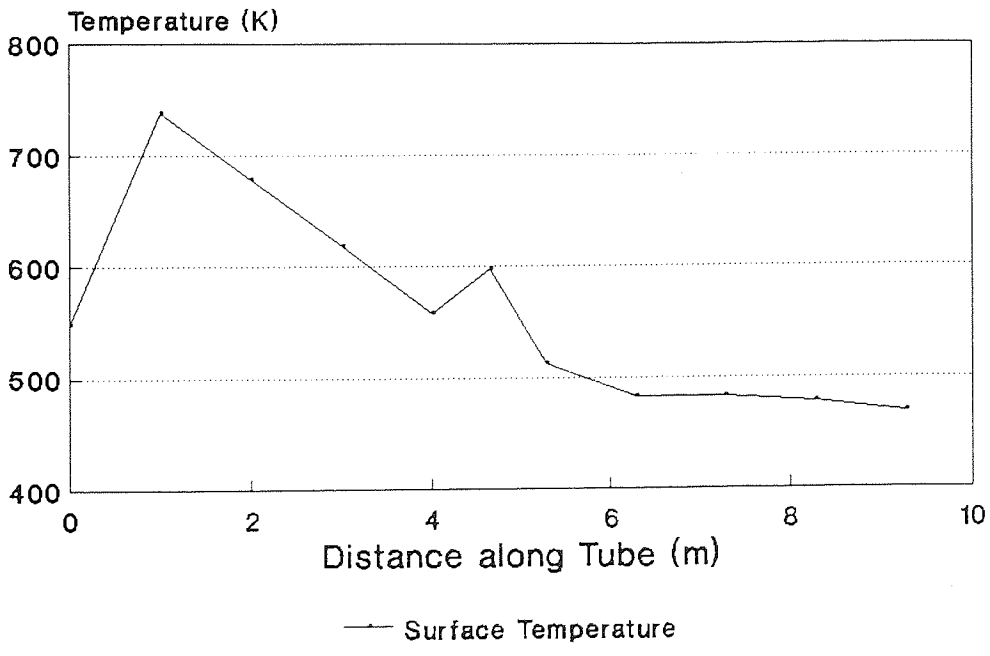


Fig.B.3: Cross-Section of IRH_3

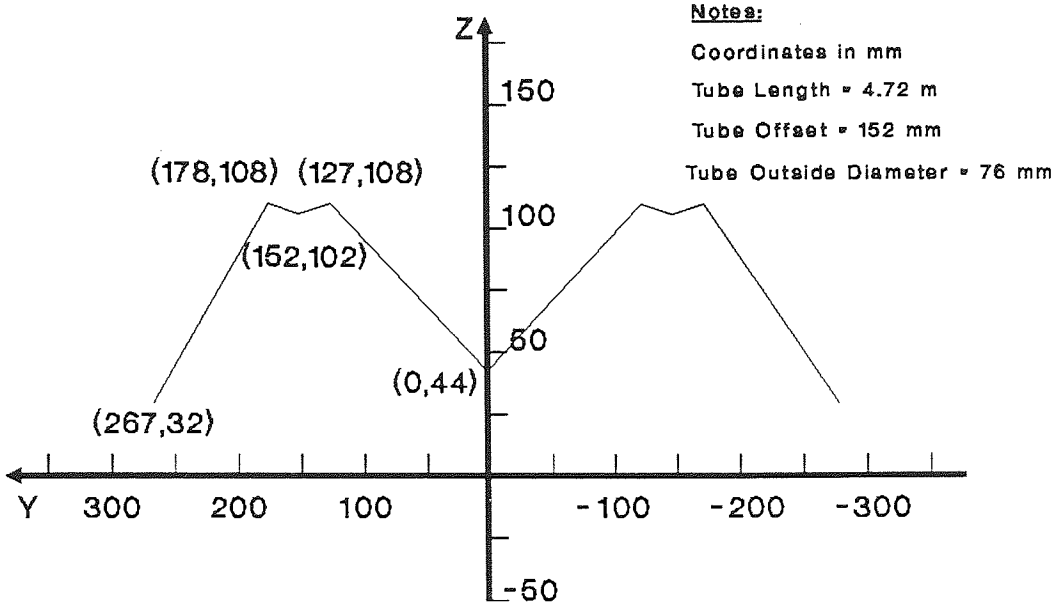


IRH_3: Tube Temperatures

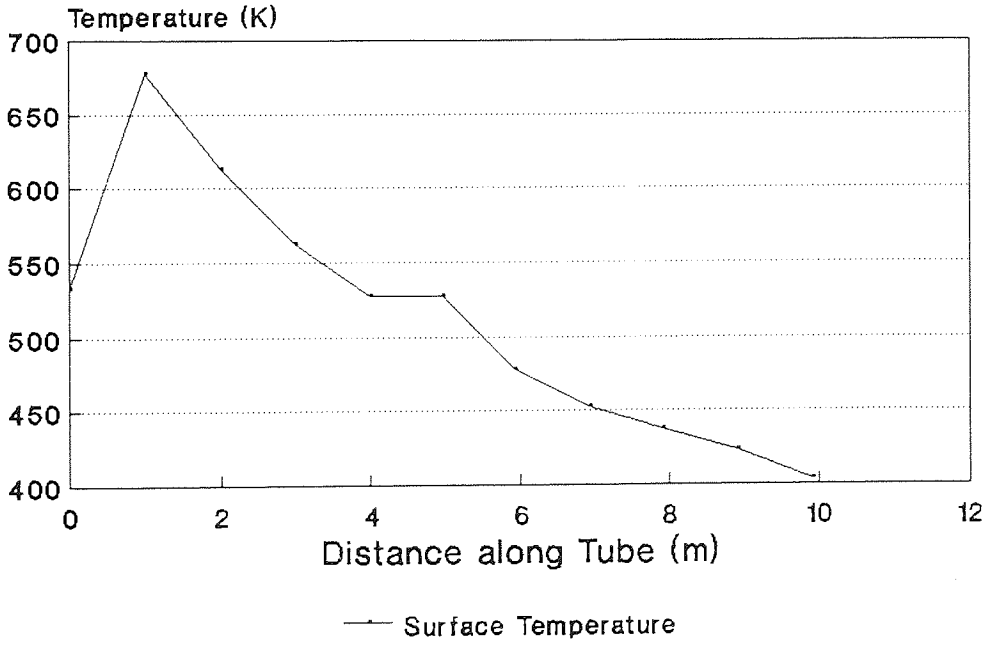


Top and Bottom Temperatures the same

Fig.B.4: Cross-Section of IRH_4

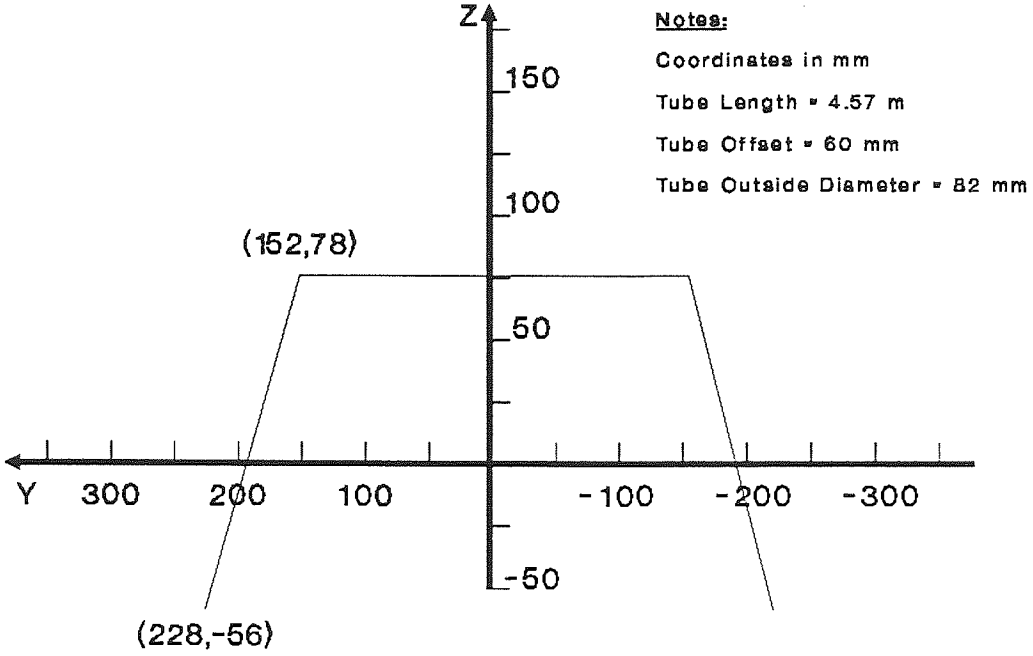


IRH_4: Tube Temperatures

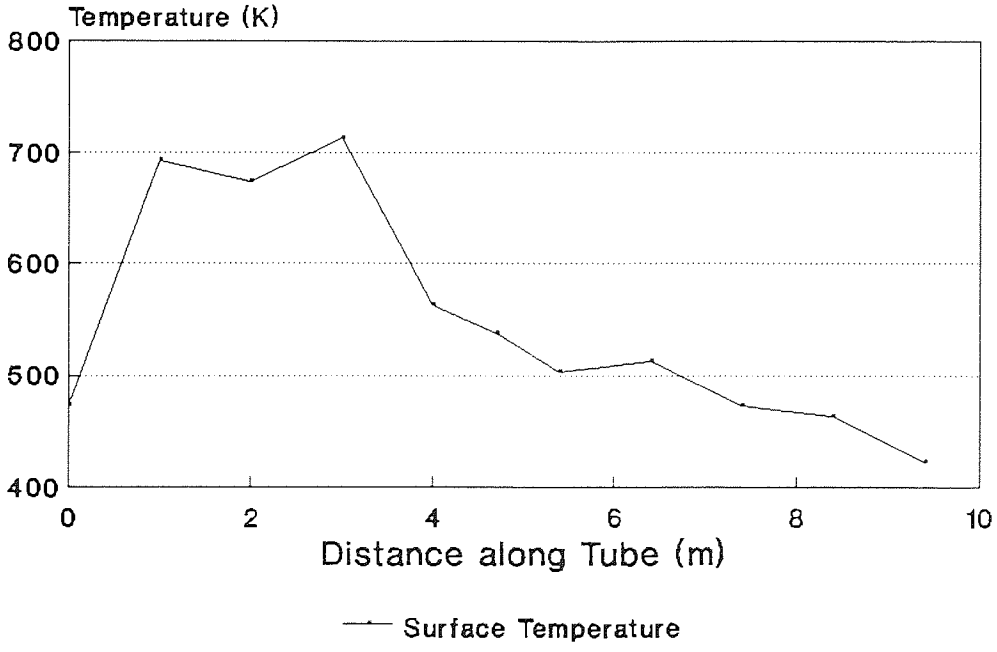


Top and Bottom Temperatures the same

Fig.B.5: Cross-Section of IRH_5

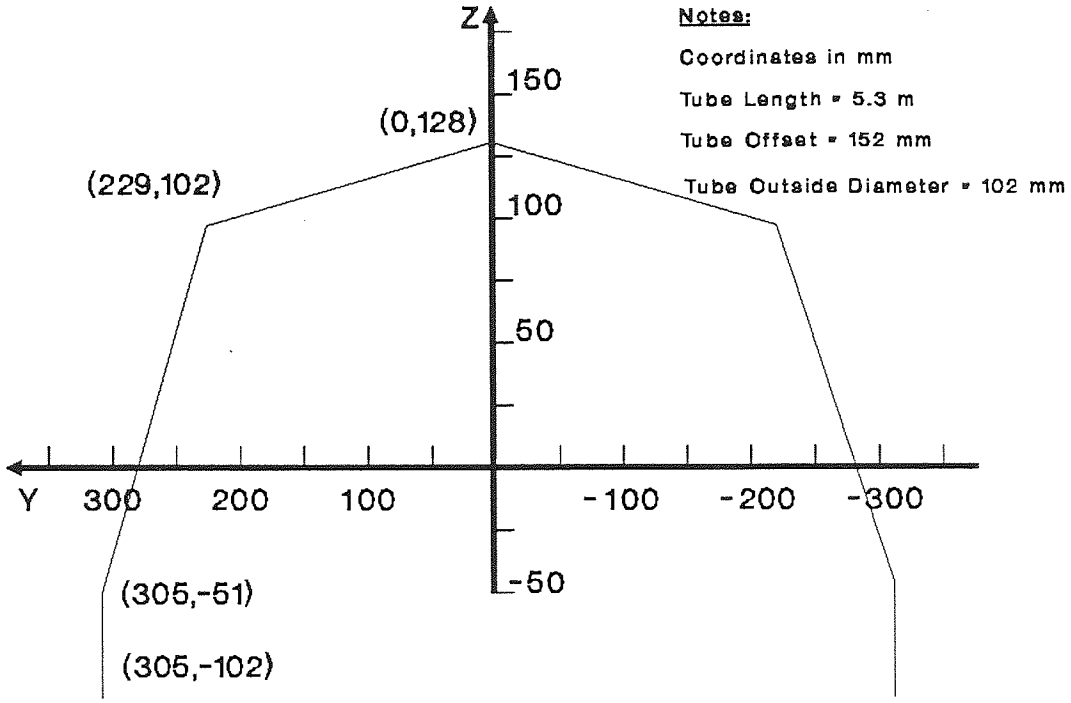


IRH_5: Tube Temperatures

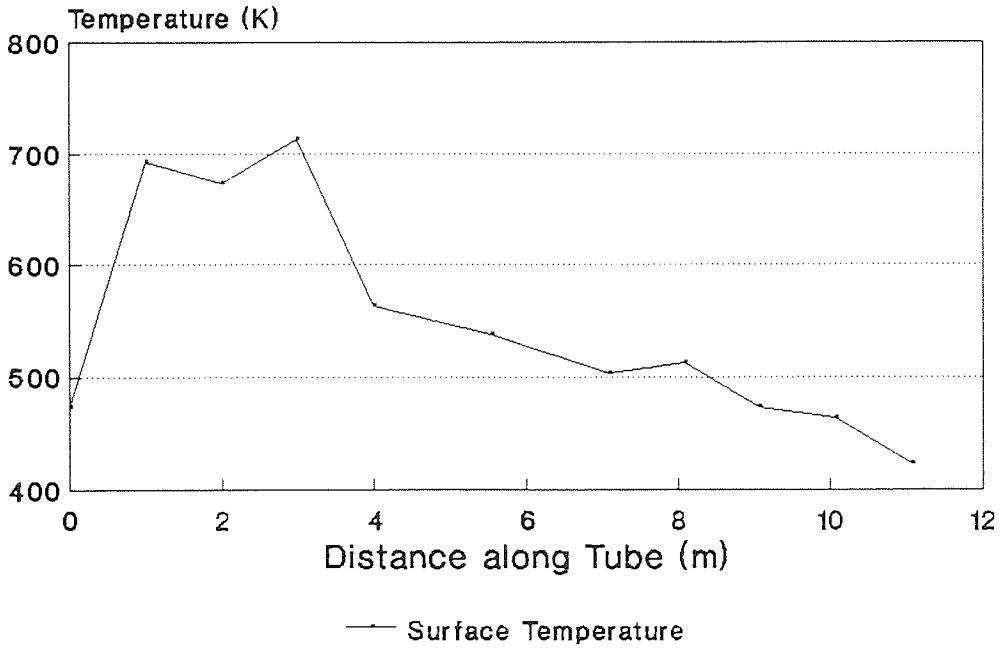


Top and Bottom Temperatures the same

Fig.B.6: Cross-Section of IRH_6

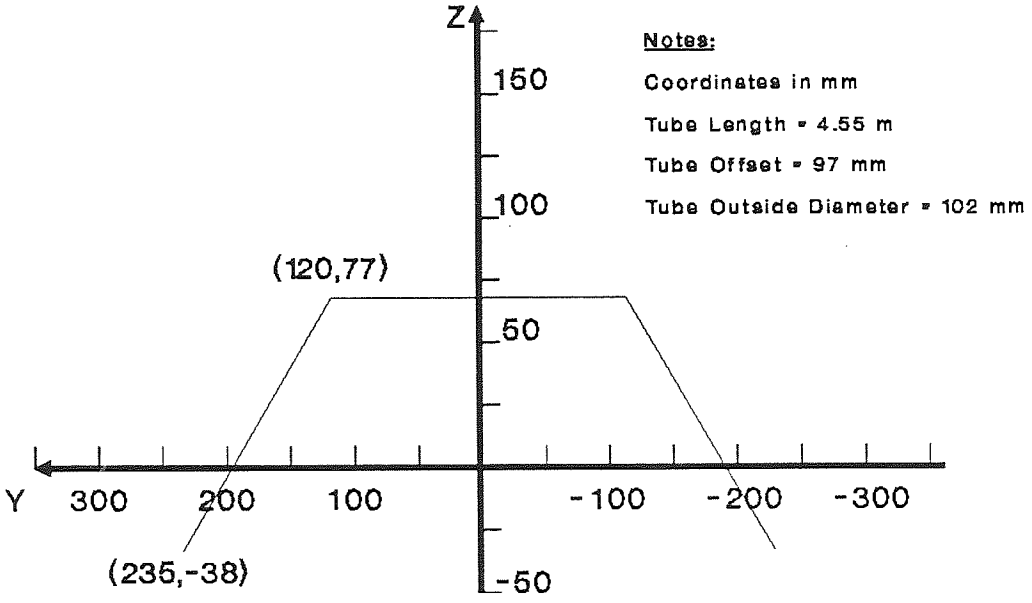


IRH_6: Tube Temperatures



Top and Bottom Temperatures the same

Fig.B.7: Cross-Section of IRH_7



IRH_7: Tube Temperature

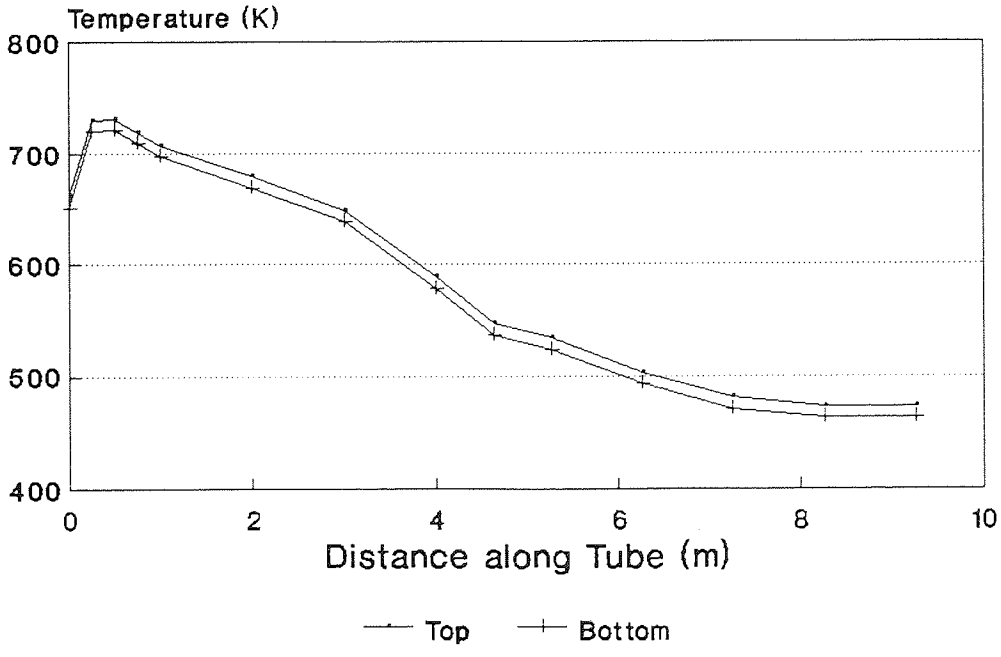
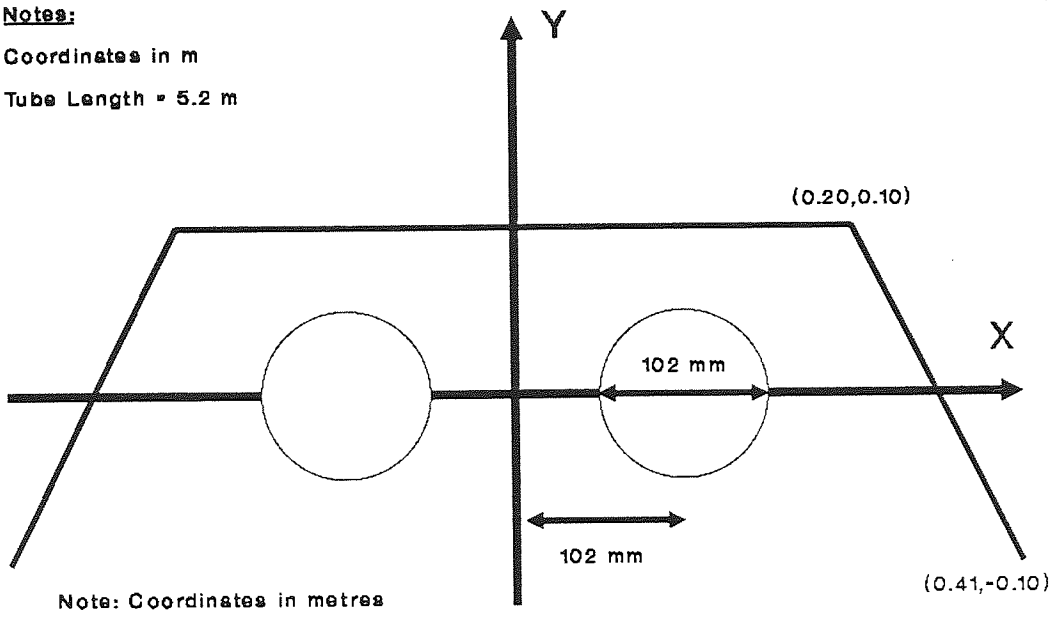
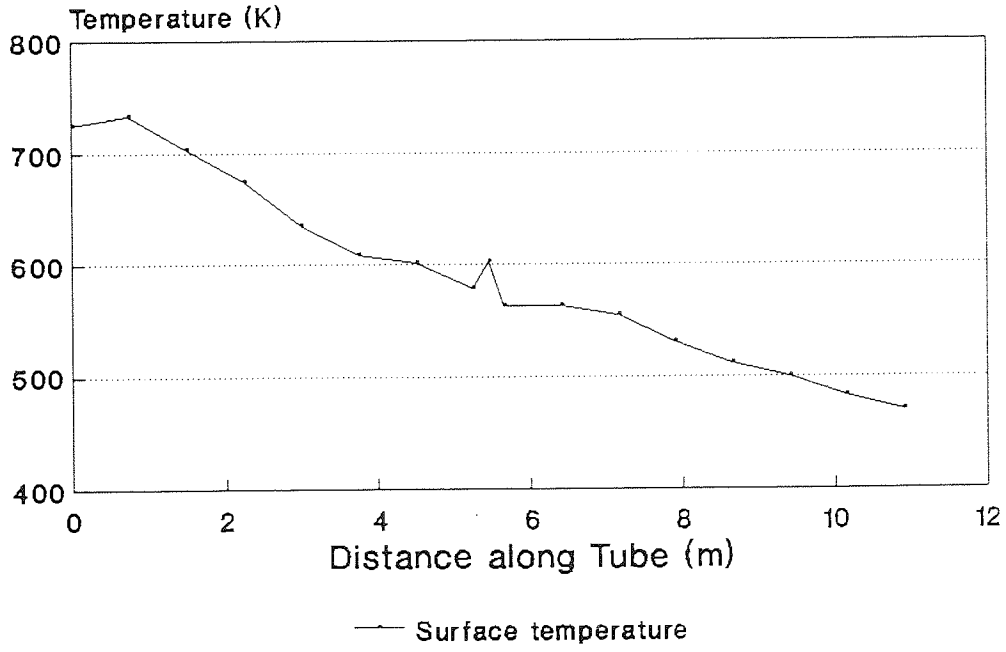


Fig.B.8: Cross-Section of IRH_8



IRH_8: Tube Temperatures



Top and Bottom Temperature the same

Appendix C: Details of the Thermography Survey

Synopsis of the Thermographic Analysis of the Chemical Engineering Workshop performed between Thursday 7th and Tuesday 12th December 1989, including details of the ensuing image captures.

Video Tapes:

There were FIVE video tapes used to record the digitized images. They were entitled:

WORKSHOP Tape 0 : IRH_6 and IRH_7 Steady State	(AGFA E120)
WORKSHOP Tape 1 : IRH_7 and IRH_6 Steady State	(SONY E180)
WORKSHOP Tape 2 : Walls and IRH_7 Dynamic	(SONY E180)
WORKSHOP Tape 3 : IRH_6 Dynamic	(SONY E180)
PILOT PLANT Tape 1 : IRH_1 Dynamic	(SONY E180)

WORKSHOP Tape 0,1 : Recorded from 3000 (VTR counter) onwards by Cheryl Harper.

WORKSHOP Tape 2,3 and PILOT PLANT 1 : This has images of the walls heating up and also the ceiling and floor which were taken by Cheryl Harper and CDZ on Saturday 9th December. Then all the dynamic images were taken by CDZ on the Monday and Tuesday.

WORKSHOP Tape 0 :

SUBJECT : IRH_6 at Steady State
LOCATION : Workshop

THERMOVISION SETTINGS: Lens = 20°
Aperture = f5.1
Distance = 4.0m
Filter = See comments

OTHER PARAMETERS : Gas Pressure = -
Ambient Temperature = - °C

Time	Thermal Range	Thermal Level	VTR Counter	Comment
14:38	1000	686	3001-3010	NOF: section 1 of heater
	50	34	3010-3021	FLM: section 1
	1000	686	3030-3040	NOF: section 2
	50	34	3021-3030	FLM: section 2
	1000	456	3040-3050	NOF: section 3
	20	14	3050-3061	FLM: section 3
	500	438	3070-3081	NOF: section 4 (U-bend)
	20	16	3061-3070	FLM: section 4

SUBJECT : IRH_7 at Steady State
LOCATION : Workshop

THERMOVISION SETTINGS: Lens = 20°
Aperture = f5.1
Distance = 4.0m
Filter = NOF

OTHER PARAMETERS : Gas Pressure = - "w.g.
Ambient Temperature = - °C

Time	Thermal Range	Thermal Level	VTR Counter	Comment
15:35	500	322	3116-3147	U-bend section
	1000	650	3147-3160	Middle section
	1000	810	3160-3175	f7.2 to capture burner tube
	500	300	3175-3190	f5.1 to capture return tube

SUBJECT : IRH_6 Dynamic Images
LOCATION : Workshop

THERMOVISION SETTINGS: Lens = 20°
Aperture = f3.6
Distance = 4.0m
Filter = NOF

OTHER PARAMETERS : Gas Pressure = - "w.g.
Ambient Temperature = - °C

Time	Thermal Range	Thermal Level	VTR Counter	Comment
16:02	500	438	13	Start heater and VCR
16:20			1000	Switch off heater
16:25			1287	Switch off VCR

SUBJECT : IRH_7 Dynamic
LOCATION : Workshop

THERMOVISION SETTINGS: Lens = 20°
Aperture = f7.2
Distance = 4.0m
Filter = NOF

OTHER PARAMETERS : Gas Pressure = - "w.g.
Ambient Temperature = - °C

Time	Thermal Range	Thermal Level	VTR Counter	Comment
16:41:20	500	322	1300	Start VCR tube at c. 70 °C
17:18:30			2770	Switch heater off
17:30:40			3172	Switch off VCR

SUBJECT : Walls, floor, and ceiling
LOCATION : Workshop
DATE : 9/12/89

THERMOVISION SETTINGS: Lens = 40°
Aperture = f1.8
Distance = 10. m
Filter = NOF

Time	Thermal Range	Thermal Level	VTR Counter	Comment
09:52 10:27	5	20	10- 1817	Position C4; Wall 2 heating ISO-1 = 25C, ISO-2 = 20C.
10:40:30	5	20	1830- 1840	Position C4; Wall 2 ISO-1 = 23C, ISO-2 = 17C.
10:48:45	5	20	1850- 1860	Position A3; Wall 3 ISO-1 = 25C, ISO-2 = 17C.
10:55	5	20	1870- 1882	Position B5; Wall 4 ISO-1 = 24C, ISO-2 = 17C.
11:06	5	22	1891- 1902	Camera on balcony looking at floor
11:13:15	5	20	1910- 1922	Position C5, pointing away from door, Ceiling.
11:27:15	10	22	1940- 1950	Position C1, Ceiling
11:32:15	5	20	1960- 1970	Position A2, Wall 2 ISO-1 = 24, ISO-2 = 17C.
11:34:45	5	20	1990- 2010	Position A3, Wall 3 ISO-1 = 23.5C, ISO-2 = 17.5C.
11:37:00	5	20	2020- 2040	Position B5, Wall 4 ISO-1 = 23C, ISO-2 = 17C.

WORKSHOP Tape 2: WALLS and PLAQUES

Subject : PLAQ_5
Location : Pilot Plant

THERMOVISION SETTINGS: Lens = 20°
Aperture = f20
Distance = 4.00m
Filter = NOF

OTHER PARAMETERS : Gas pressure 7.1" w.g.
Ambient Temperature = 15.4 °C

Time	Thermal Range	Thermal Level	VTR Counter	Comment
11:50	500	950	2040:2050	Steady State
11:50	500	950	2060:2070	Cooling
			2070:2080	and Heating

SUBJECT : PLAQ_5 Heater
LOCATION : Pilot Plant

THERMOVISION SETTINGS: Lens = 20°
Aperture = f5.1
Distance = 4.00m
Filter = FLM

OTHER PARAMETERS : Gas Pressure = 7.1"w.g.
Ambient Temperature = 16.0 °C

Time	Thermal Range	Thermal Level	VTR Counter	Comment
11:55	100	294	2120:2135	Steady State
	200	280	2150:2160	Cooling
			2160:2225	and heating
	200	280	2230:2240	One burner; steady state

SUBJECT : PLAQ_6 Heater
LOCATION : Pilot Plant

THERMOVISION SETTINGS: Lens = 20°
Aperture = f3.6
Distance = 2.00m
Filter = FLM

OTHER PARAMETERS : Gas Pressure = 7.1"w.g.
Ambient Temperature = 15.5 °C
Relative Humidity = 47.4 %

Time	Thermal Range	Thermal Level	VTR Counter	Comment
12:10	200	420	2250:2260	Steady state with Recuperator
12:13	200	440	2260:2270	" " " "
	500	466	2280:2310	Cooling with recuperator
	500	466	2315:2410	Heating with recuperator
12:20	200	420	2420:2435	Steady state recuperator
			2435:2455	switching OFF->ON
12:25	200	480	2460:2466	Steady state recuperator
			2466:2510	switching ON->OFF

SUBJECT : IRH_7 Burner Section
LOCATION : Workshop
THERMOVISION SETTINGS: Lens = 20 °
Aperture = f7.2
Distance = 3.38m
Filter = NOF

OTHER PARAMETERS : Ambient Temperature = 16.7 °C

Time	Thermal Range	Thermal Level	VTR Counter	Comment
15:05:45	1000	550	2520	Start VTR
15:06:00				Switch OFF heater
15:16:00			c.2800	Switch ON heater
15:29:05				Stop VTR

SUBJECT : IRH_7 first middle section
LOCATION : Workshop

THERMOVISION SETTINGS: Lens = 20 °

Aperture = f5.1
Distance = 3.38m
Filter = NOF

OTHER PARAMETERS : Ambient Temperature = °C

Time	Thermal Range	Thermal Level	VTR Counter	Comment
16:18:30	1000	572	3280	START VTR
16:18:45				Switch OFF heater
16:28:45			c3590	Switch ON heater
16:50:00			4201	STOP VTR

SUBJECT : IRH_7 second middle section
LOCATION : Workshop

THERMOVISION SETTINGS: Lens = 20 °
Aperture = f2.5
Distance = 3.38m
Filter = NOF

OTHER PARAMETERS : Ambient Temperature = 17.9 °C

Time	Thermal Range	Thermal Level	VTR Counter	Comment
11:22:00	1000	832	5220	START VTR
11:22:15				Switch OFF heater
11:32:15			5460	Switch ON heater
11:50:15			5864	STOP VTR

SUBJECT : IRH_7 U-bend section
LOCATION : Workshop

THERMOVISION SETTINGS: Lens = 20 °
Aperture = f1.8
Distance = 3.38m

Filter = NOF

OTHER PARAMETERS : Ambient Temperature = 16.6 °C

Time	Thermal Range	Thermal Level	VTR Counter	Comment
10:44:00	1000	876	4500	START VTR
10:44:15			c4510	Switch OFF heater
10:54:15			4760	Switch ON heater
11:12:30				STOP VTR

WORKSHOP Tape 3: IRH_6 DYNAMIC

SUBJECT : IRH_6 - Section A
LOCATION : Workshop

THERMOVISION SETTINGS: Lens = 20 °
Aperture = f2.5
Distance = 3.38m
Filter = NOF

OTHER PARAMETERS : Ambient Temperature = °C

Time	Thermal Range	Thermal Level	VTR Counter	Comment
12:07:00	1000	784	0000	START VTR
12:07:15				Switch OFF heater
12:13:15			0342	Switch ON heater
12:26:30			1061	STOP VTR

SUBJECT : IRH_6 - Section B
LOCATION : Workshop

THERMOVISION SETTINGS: Lens = 20 °
Aperture = f2.5
Distance = 3.38m
Filter = NOF

OTHER PARAMETERS : Ambient Temperature = °C

Time	Thermal Range	Thermal Level	VTR Counter	Comment
12:35:00	1000	818	1100	START VTR
12:35:15				Switch OFF heater
12:41:15			1395	Switch ON heater
12:54:30			1958	STOP VTR

SUBJECT : IRH_6 - Section B cool tube
LOCATION : Workshop

THERMOVISION SETTINGS: Lens = 20 °
Aperture = f1.8
Distance = 3.38m
Filter = NOF

OTHER PARAMETERS : Ambient Temperature = °C

Time	Thermal Range	Thermal Level	VTR Counter	Comment
13:40:00				START VTR
13:40:15				Switch OFF heater
13:46:15				Switch ON heater
13:59:30				STOP VTR

SUBJECT : IRH_6 - Section C cool
LOCATION : Workshop

THERMOVISION SETTINGS: Lens = 20 °
Aperture = f2.5
Distance = 3.38m
Filter = NOF

OTHER PARAMETERS : Ambient Temperature = °C

Time	Thermal Range	Thermal Level	VTR Counter	Comment
15:07:00	200	308	2800	START VTR
15:07:15				Switch OFF heater
15:13:15			3010	Switch ON heater
15:26:15			3414	STOP VTR

SUBJECT : IRH_6 - Section C hot
LOCATION : Workshop

THERMOVISION SETTINGS: Lens = 20 °

Aperture = f5.1
Distance = 3.38m
Filter = NOF

OTHER PARAMETERS : Ambient Temperature = 20.1 °C

Time	Thermal Range	Thermal Level	VTR Counter	Comment
15:33:00	1000	700	3450	START VTR
15:33:15				Switch OFF heater
15:39:15			3640	Switch ON heater
15:52:15			4009	STOP VTR

SUBJECT : IRH_6 - Section D hot
LOCATION : Workshop

THERMOVISION SETTINGS: Lens = 20 °
Aperture = f5.1
Distance = 3.38m
Filter = NOF

OTHER PARAMETERS : Ambient Temperature = 20.2 °C

Time	Thermal Range	Thermal Level	VTR Counter	Comment
15:56:00	1000	646	4050	START VTR
15:56:15				Switch OFF heater
16:02:15			4220	Switch ON heater
16:15:15			4546	STOP VTR

SUBJECT : IRH_6 - Section D cool
LOCATION : Workshop

THERMOVISION SETTINGS: Lens = 20 °
Aperture = f2.5
Distance = 3.38m

Filter = NOF

OTHER PARAMETERS : Ambient Temperature = 20.7 °C

Time	Thermal Range	Thermal Level	VTR Counter	Comment
16:21:00	500	456	4600	START VTR
16:21:15				Switch OFF heater
16:27:15			4755	Switch ON heater
16:40:30			5081	STOP VTR

PILOT PLANT Tape 1: IRH_1 Dynamic

SUBJECT : IRH_1 - Section A
LOCATION : Workshop

THERMOVISION SETTINGS: Lens = 40 °
Aperture = f5.1
Distance = 1.42m
Filter = NOF

OTHER PARAMETERS : Ambient Temperature = 18.6 °C
Relative Humidity = 47.7 %

Time	Thermal Range	Thermal Level	VTR Counter	Comment
19:52:00	1000	786	5200	START VTR
19:52:15				Switch OFF heater
19:58:15			5346	Switch ON heater
20:11:30			5647	STOP VTR

SUBJECT : IRH_1 - Section B
LOCATION : Pilot Plant

THERMOVISION SETTINGS: Lens = 40 °
Aperture = f3.6
Distance = 1.42m
Filter = NOF

OTHER PARAMETERS : Ambient Temperature = 20.2 °C
Relative Humidity = 37.5 %

Time	Thermal Range	Thermal Level	VTR Counter	Comment
17:27:00	1000	834	1150	START VTR
17:27:15				Switch OFF heater
17:33:15			1444	Switch ON heater
17:46:30			1999	STOP VTR

SUBJECT : IRH_1 - Section C
LOCATION : Pilot Plant

THERMOVISION SETTINGS: Lens = 40 °
Aperture = f5.1
Distance = 1.42m
Filter = NOF

OTHER PARAMETERS : Ambient Temperature = 17.1 °C
Relative Humidity = 52.8 %

Time	Thermal Range	Thermal Level	VTR Counter	Comment
17:52:00	1000	770	2050	START VTR
17:52:30				Switch OFF heater
17:58:30			2300	Switch ON heater
18:11:45			2772	STOP VTR

SUBJECT : IRH_1 - Section D hot
LOCATION : Pilot Plant

THERMOVISION SETTINGS: Lens = 40 °
Aperture = f7.2
Distance = 1.42m
Filter = NOF

OTHER PARAMETERS : Ambient Temperature = 17.8 °C
Relative Humidity = 50.9 %

Time	Thermal Range	Thermal Level	VTR Counter	Comment
18:16:00	1000	896	2800	START VTR
18:16:15				Switch OFF heater
18:22:15			3010	Switch ON heater
18:35:45			3434	STOP VTR

SUBJECT : IRH_1 - Section E hot
LOCATION : Pilot Plant

THERMOVISION SETTINGS: Lens = 40 °
Aperture = f10.0
Distance = 1.42m
Filter = NOF

OTHER PARAMETERS : Ambient Temperature = 17.6 °C
Relative Humidity = 50.1 %

Time	Thermal Range	Thermal Level	VTR Counter	Comment
18:41:00	1000	616	3500	START VTR
18:41:15				Switch OFF heater
18:47:15			3686	Switch ON heater
19:00:15			4061	STOP VTR

SUBJECT : IRH_1 - Section E cool
LOCATION : Pilot Plant

THERMOVISION SETTINGS: Lens = 40 °

Aperture = f2.5
Distance = 1.42m
Filter = NOF

OTHER PARAMETERS : Ambient Temperature = 18.6 °C
Relative Humidity = 46.9 %

Time	Thermal Range	Thermal Level	VTR Counter	Comment
19:03:00	1000	758	4100	START VTR
19:03:15				Switch OFF heater
19:09:15			4270	Switch ON heater
19:21:45			4596	STOP VTR

SUBJECT : IRH_1 - Section D cool
LOCATION : Pilot Plant

THERMOVISION SETTINGS: Lens = 40 °
Aperture = f2.5
Distance = 1.42m
Filter = NOF

OTHER PARAMETERS : Ambient Temperature = 18.1 °C
Relative Humidity = 47.7 %

Time	Thermal Range	Thermal Level	VTR Counter	Comment
19:29:00	1000	828	4650	START VTR
19:29:15				Switch OFF heater
19:35:15			4807	Switch ON heater
19:48:30			5128	STOP VTR

REFERENCES:

Chapter 1: An Introduction to Radiant Heating

British Gas, School of Fuel Management. Combustion Engineering. Solihull, 1983. p544.

British Standards Institute. B.S. 4727:1971, Part 4, Group 01, "Radiation and Photometry". Glossary of electrotechnical, power, telecommunication, electronics, lighting and colour terms. BSI, 1971.

Building Services Information and Research Association. "Product Profile: Industrial Heating (2)". Statistics Bulletin, 14, number 4, pp i-iv, December 1989.

The Chartered Institute of Building Service Engineers (CIBSE). The CIBSE Guide: Volume A5, Thermal Response of Buildings. London: CIBSE, 1986. p A5-4.

McIntyre, D.A. "The Thermal Radiation Field". Build. Sci., 9, pp 247-262. Pergammon Press, 1974.

Chapter 2: The Monte Carlo Model

Butler, R. "A Mathematical Model of an Overhead Radiant Heater", British Gas Internal Report WH/T/R&D/82/202, Watson House, 1982.

British Standards Institute. B.S. 3561: 1962, Appendix K, "Determination of Radiant Heat Distribution and Efficiency of Overhead Heaters". London: B.S.I., 1962.

Fanger, P.O. Thermal Comfort. New York: McGraw-Hill, 1972. p151.

International Standards Organisation. I.S.O. 7730-1984(E), Annex D, "Computer Program for calculation of predicted mean vote (PMV) and predicted percentage of dissatisfied (PPD)." I.S.O., 1984.

Maund, J.K. "A Comparison of Four Gas-Fired Radiant Heaters", Aston University, 1979. Fig 4, pp 23-24.

Maund, J.K. and Ziesler, C.D. "An Experimental Determination of the Radiant Output of a Schwank 2102 Plaque Heater", Aston University Internal Report, 1988.

Numerical Algorithms Group. NAG FORTRAN library manual Mark 13. Section G05CAF. NAG: Oxford, July 1988.

Numerical Algorithms Group. Op. Cit. Section C02AEF. NAG: Oxford. July 1988.

Schreider, Y. Methods of Statistical Testing: The Monte Carlo Method. Elesvier Publishing, 1964.

Smith, Green, and Klem. "Empirical Tests on A Pseudo-Random Number Generator." J. Assoc. Comp. Mach. 6, p527, 1959.

Ulam, S. and Metropolis, M. "The Monte Carlo Method". J. Amer. Stat. Assoc., 44, number 247, pp 335-341.

Ward, J. and Tucker, R. "Use of the Monte Carlo Technique for the Determination of Radiation Exchange Areas in 'Long-Furnace' Models", paper presented at the 8th International Heat Exchange Conference, San Fransisco, 1986.

Ziesler, C.D. "First Year Report: An Evaluation of Industrial Radiant Heaters", Aston University Chemical Engineering Dept., 1988.

Chapter 3: Room Simulations

Bode, H.W. Network Analysis and Feedback Amplifier Design. New York: Van Nostrand, 1954. Section 1.2, "Branch Equations for a Passive Circuit", pp1-4.

Carslaw, H.S., and Jaeger, J.C. Conduction of Heat in Solids, 2nd Edition. London: Oxford University Press, 1959. Section 12.8-V, "The Slab of any Number of Layers", p326.

Chan, Shu-Park. Introductory Topological Analysis of Electrical Networks, H.R.W. Series in Electrical Engineering, Electronics and Systems. New York: Holt, Rinehart, and Wilson, 1969. Section 4.5, "The Node Equations", pp122-139.

Chirlian, P.M. Basic Network Theory, McGraw-Hill Electrical and Electronic Engineering Series. New York: McGraw-Hill Book Company, 1969. Section 2.4, "Simple Equivalent Circuits", pp79-92.

CIBSE. "Thermal Properties of Building Structures". Volume A3 of CIBSE Guide. London: CIBSE, 1986.

Clarke, J.A. Energy Simulation in Building Design. Bristol: Adam Hilger Ltd., 1985. Section 2.1, "Response Function Methods", pp 24-28.

Edminster, J.A. Electric Circuits, 2nd Edition, Schaum's Outlines Series in Engineering. New York: McGraw-Hill, 1983. Section 4, pp48-62.

Elgerd, O.I. Control Systems Theory, McGraw-Hill Electronic and Electrical Engineering Series. New York: McGraw-Hill, 1967. Section 7.1, "Performance Specifications", pp175-183.

Elgerd, O.I. Op. Cit. Section 3, "Concepts of State and State Variables", pp29-58.

Elgerd, O.I. Op. Cit. Section 3.2, "Reduction of Differential Equations to their Normal Form", pp31-33.

Elgerd, O.I. Op. Cit. Section 6.4.4, "The Nyquist Stability Criterion", pp151-155.

Fisk, D.J. Thermal Control of Buildings. London: Applied Science Publishers, 1981. Section 3.7.1, "Thermal Impedance of a Wall", pp66-70.

Franklin, G.F., Powell, J.D., and Abbas Emani-Naeini. Feedback Control of Dynamic Systems. Reading, Mass.: Addison-Wesley, 1986. Section 6, "State Space Design", pp331-440.

Gebhart, B. Heat Transfer, 2nd Edition. New York: McGraw-Hill Publishing Company Ltd., 1971. Section 13.8, "Network Analogues for Unsteady State", pp543-553.

Gray, W.A. and Muller, R. Engineering Calculations in Radiative Heat Transfer. Oxford: Pergamon Press, 1974.

Jacobs, O.L.R. Introduction to Control Theory. Oxford: Clarendon Press, 1974.

Kochenburger, R.J. "A Frequency Response Method for Analyzing and Synthesizing Contactor Servomechanisms," *Trans. AIEE,* **69**: pp 270-283, 1950.

Paschkiss, V. and Heisler, M.P. "Analogue Computing," *J.App.Phys.*, 17, pp246-249, 1944.

Porch, H.R. and Mo, T. "System Identification: Automated Plant Modelling", *MRS Relay*, British Gas R&T Division, June 1990, pp4-7.

Shampine, L.F. and Gordon, M.K. Computer Solution of Ordinary Differential Equations. San Francisco: W.H.Freeman and Company, 1975.

System Control Technology. Ctrl-C User Guide: Version 4.1. Birmingham: Aston University Computing Service, 1988.

Whitman, D.J., and Ziesler, C.D. "Mathematical Modelling of Radiant Heaters." *MRS Relay*, October 1989, pp7-10.

Ziesler, C.D., and J.K. Maund. "An Introduction to the Sizing of Radiant Heating Systems." Applied Energy Research. Proc. of the Institute of Energy Conf., Swansea, UK, 5-7 September 1989. Bristol: Adam Hilger, 1989. pp47-58.

Chapter 4: Control of Radiant Heating Systems

Clarke, J.A. Energy Simulation in Building Design. Bristol: Adam Hilger Ltd., 1985. Appendix B: "Thermophysical properties of building materials", pp336-338.

Koussouris S.G. "Development of a Binary Digital Flow Meter" Aston University M.Sc. thesis, 1984.

Chapter 5: Radiometry

Hathaway, M.J. and Page, M.W. "Determination of Gross Flue Losses on Oil Gas-Fired Plant". British Gas Report N 376, March 1982.

Lynn, P.A. An Introduction to the Analysis and Processing of Signals, 2nd Edition. London: Macmillan Publishers Ltd., 1982. Section 9.4.4, "Filters with Finite Impulse Response", pp196-206.

Maund, J.K. "Report on Four Gas-Fired Radiant Heaters", Aston University, 1979. Table 9, p39.

Maund, J.K. US patent number 4702618, 27 October 1987.

McIntyre, D.A. "Radiant Heating with Quartz Linear Lamps; Calculation Methods". Electricity Council Research Centre, Document ECRC/M2183, 1987. Section 5, "Transmission through the Atmosphere", pp11-15.

Millward, S. Private Communication. October 1988.

RS Components Ltd. RS Data Set: Sheet number 5055. November 1990.

Chapter 6: Thermometry

Biodata Ltd. "Microlink: The Hardware Users Manual".

Grant M.P. "A Procedure for using the MRS Thermal Imaging Equipment". Midlands' Research Station internal report, N 5248, 1989.

RS Components Ltd. RS Data Set. Sheet number 8638. November 1990

Chapter 7: Final Perspectives

British Standards Institute. B.S. 3561: 1962, Appendix K, "Determination of Radiant Heat Distribution and Efficiency of Overhead Heaters". London: B.S.I., 1962.

Hanrahan, P. "A Survey of Ray-Surface Intersection Algorithms", pp 79-120, of An Introduction to Ray-Tracing. Ed. Glassner, A.S. London: Academic Press, 1989.

Glassner, A.S. An Introduction to Ray-Tracing. London: Academic Press, 1989.